

**Author: Antigoni Noula (CERTH), Christina Tsita (CERTH),
Dimosthenis Ioannidis (CERTH)**



D2.5 Requirement-Driven System Development V2 (Month 18)

Contractual Date of Delivery to the EC:	30.06.2019
Actual Date of Delivery to the EC:	28.06.2019
Author(s):	Antigoni Noura (CERTH), Christina Tsita (CERTH), Dimosthenis Ioannidis (CERTH)
Participant(s):	Napoleon Bezas (CERTH), Giorgos Sfikas (CERTH), Giuseppe Raveduto (ENG), Vincenzo Croce (ENG), Giuseppe Mastandrea (E@W), Luigi D'Oriano (E@W), Mircea Bukur (KIWI), Dara Kolajo (KIWI), Ugo Stecchi (ATOS), Juan Sancho (ATOS), Javier Gomez (ATOS), Lourdes Gallego (ATOS), Tommaso Bragatto (ASM), Francesca Santori (ASM), Claudia Pop (TUC), Marcel Antal (TUC), Francesco Bellesini (EMOT), Michele Pagliaccia (EMOT), Benjamin Hunter (TU), Fathi Abugchem (TU), Charalampos Psarros (TU), Paul van Schaik (TU).
Project:	enabling new Demand Response Advanced, Market oriented and secure technologies, solutions and business models (eDREAM)
Work package:	Wp2 – User requirements, use cases and system specification
Task:	2.4 – Requirement-Driven System Development V1
Confidentiality:	Public
Version:	1.0

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

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

This deliverable D2.5 describes a reviewed version of the eDREAM architecture (initial version reported in D2.4) due to be submitted at M18 (second iteration). The final version will be released at M30 (third iteration).

The aim of this deliverable is to present the updates on architectural components' dependencies and specifications as an outcome of the progress of development activities within WP3, WP4 and WP5 and the definition of the eDREAM integration and interconnection plan (WP6). The reported updates are important, in order to map the way for a successful integration and ensure the consistency of use cases and user requirements.

In Chapter 2, the methodological approach for system requirements elicitation is presented indicating the relevance of System requirements with User requirements (D2.1) and Use cases (D2.2) and depicting the iterative analysis process for specifications' verification. The technical specifications were reviewed through internal elicitation using appropriate templates, so that the updates could be captured properly.

For the sake of completeness, the Conceptual architecture of the eDREAM system is presented in the beginning of Chapter 3 with some refinements, as in the initial document D2.4. This is a high-level view of the overall architecture, describing the three major layers of the eDREAM platform, the Field Data Aggregation, the Core Backbone Platform and the Visualization Framework. The platform comprises also the Decentralized Repository for secure storage of the data exchanged between the modules of the platform and with external interfaces.

Then, in Chapter 4, the structural view of the system is updated, presenting the different architectural components that deliver the system's functionalities. This view provides the system's decomposition into different components, demonstrating the updated dependencies between them, their interfaces, the data exchanges and their functionalities.

Chapter 5 focuses on the dynamic behaviour of the system, where the actual high-level and low-level use cases are correlated with each architectural component. The way that each component acts within the use cases determines its functional requirements. The updates of components' dependencies are reflected on the structure of the respective UML sequence diagrams causing also some changes in the main flow of the use cases' steps.

The deployment view is described in Chapter 6 defining the physical environment, in which the system is intended to run including hardware requirements (e.g. processing nodes, network interconnections, etc.). An updated version concerning the integration of field devices from pilot sites is presented.

Finally, the Chapter 7 presents in appropriate templates updated version of the detailed technical specifications of the eDREAM core architectural components focusing on the functionalities, inputs/outputs, interfaces and data types. The detailed interfaces of the field devices and the architectural components are presented in Annexes IV and V.

It should be mentioned that this report presents an updated version of the eDREAM architecture definition, but the architectural components and their detailed specifications will remain an open issue until all components and subsystems are developed and deployed and all modules have been integrated to the eDREAM platform. Thus, this document can be considered as a living document.

Executive Summary	4
List of Figures	7
List of Tables	8
List of Acronyms and Abbreviations	10
1 Introduction	11
1.1 Scope and objectives of the deliverable and relevance in the eDREAM framework	11
1.2 Structure of the deliverable	12
2 Methodology	14
2.1 System Architecture Concepts and Design Fundamentals	15
2.1.1 Design Principles	15
2.1.2 Static and Dynamic Structures	16
2.1.3 End-users and Stakeholder requirements' perspective	16
2.1.4 Architectural Views	16
2.1.5 Architectural Elements Perspectives	17
3 Conceptual Architecture	19
4 Structural – Functional View	23
4.1 Overall Structural View of eDREAM architecture	23
4.2 Field Middleware	25
4.2.1 IoT Devices	25
4.2.2 Semantic Context Broker	26
4.3 Techniques for DR and Energy Flexibility Assessment	27
4.3.1 Electricity Consumption/Production Forecasting	27
4.3.2 PV/RES Degradation & Trend Analysis	28
4.3.3 Baseline Flexibility Estimation	29
4.3.4 Virtual Power Plants Generation Modeling & Forecasting	29
4.3.5 Multi-building DR characterization through thermal, optical and LIDAR information fusion	30
4.4 Next generation DR Services for Aggregators and Customers	31
4.4.1 Load Profiling	31
4.4.2 Big Data Clustering at Multiple Scales	31
4.4.3 Customers Segmentation	32
4.4.4 VPP and active Microgrid Flexibility Profiling	33
4.4.5 VPP & DR Services Optimization Engine	33
4.5 Blockchain-enabled Decentralized Network Control Optimization and DR Verification	34
4.5.1 Distributed Ledger	34
4.5.2 Blockchain-driven control for LV networks (flexibility management)	35
4.5.3 Blockchain-driven Energy Market	36
4.5.4 Closed loop DR Verification Engine	37

4.6	Multi-level and Multi-factor Visualization Framework (Front-End)	38
4.6.1	HMIs.....	39
4.6.2	Suites	39
5	Dynamic View	43
5.1	High Level Use Case 01: Prosumers DR flexibility aggregation via smart contract	43
5.1.1	HL-UC01_LL-UC01: Prosumers enrolment in demand response programs.....	43
5.1.2	HL-UC01_LL-UC02: Contract Setting.....	46
5.1.3	HL-UC01_LL-UC03: Potential energy flexibility evaluation	49
5.1.4	HL-UC01_LL-UC04: Energy demand/production forecasting for day-ahead trading of flexibility ..	52
5.1.5	HL-UC01_LL-UC05: Flexibility request	55
5.1.6	HL-UC01_LL-UC06: Flexibility offering.....	58
5.1.7	HL-UC01_LL-UC07: Flexibility acceptance	61
5.1.8	HL-UC01_LL-UC08: Flexibility provisioning.....	63
5.2	High Level Use Case 02: Peer-to-peer local energy trading market.....	66
5.2.1	HL-UC02_LL-UC01: Prosumers registration with the energy trading platform.....	66
5.2.2	HL-UC02_LL-UC02: Prosumers bids/offers submission	69
5.2.3	HL-UC02_LL-UC03: Energy clearing price determination	72
5.2.4	HL-UC02_LL-UC04: Transactions validation and financial settlement.....	74
5.3	High Level Use Case: VPP in Energy Community.....	77
5.3.1	HL-UC03_LL-UC01: Prosumers Profiling and Clusterization	77
5.3.2	HL-UC03_LL-UC02: VPP Capability Evaluation for Reserve services and for Frequency services...	80
5.3.3	HL-UC03_LL-UC03: VPP Export Evaluation for Wholesale market (Intraday trading) and for Imbalance market.....	83
6	Deployment View	87
6.1	Active Micro-Grid – ASM Terni.....	87
6.2	Community-based Virtual Power Plants – KiWi Power	90
6.3	Overall Deployment Architecture	91
6.4	Field Devices	95
7	Architectural Components Detailed Specifications	103
8	Conclusion	160
	References	161
	Annex I: Functional & Non-Functional Requirements	162
	Annex II: Architectural Specifications Templates	172
	Annex III: eDREAM Sensors/Gateways/Infrastructure Specifications Template	175
	Annex IV: Field Devices APIs	177
	Annex V: Architectural Components APIs.....	188

Figure 1 Architecture Design Approach and Workflow	14
Figure 2 Architectural View 4+1 model.....	17
Figure 3 eDREAM Conceptual Architecture	19
Figure 4 eDREAM overall structural view	24
Figure 5 Field Data Aggregation layer	25
Figure 6 Electricity Consumption/Production Forecasting	28
Figure 7 PV/RES Degradation & Trend Analysis	29
Figure 8 Baseline Flexibility Estimation.....	29
Figure 9 Virtual Power Plants Generation Modeling & Forecasting	30
Figure 10 Multi-building DR Characterization through optical, thermal and LIDAR information fusion	31
Figure 11 Load Profiling	31
Figure 12 Big Data Clustering at Multiple Scales	32
Figure 13 Customer Segmentation	33
Figure 14 VPP and active Microgrid Flexibility Profiling	33
Figure 15 VPP and DR Services Optimization Engine	34
Figure 16 Blockchain-driven control for LV networks	36
Figure 17 Secured Blockchain-driven Energy Market	37
Figure 18 Closed-loop DR Verification Engine	38
Figure 19 DSS & DR Strategies Optimization	40
Figure 20 DR Aerial Survey Toolkit.....	41
Figure 21 Forecasting Tool	41
Figure 22 Graph-based Analytics	42
Figure 23 Internal and external connections in the Terni Micro-Grid.....	88
Figure 24 Unbundled Smart Meter Concept.....	89
Figure 25 High-level architecture for the communication between SM and SMX	89
Figure 26 Residential estate in Greenwich for testing residential demand response scenarios	90
Figure 27 eDREAM Deployment View Architecture.....	92
Figure 28 IoT Agents' Concept and Connection.....	93
Figure 29 Blockchain-based Infrastructure	95
Figure 30 Testbed Topology for the Distributed Ledger Technology	95

Table 1 Main objectives of the three versions of the deliverable for System Requirements and Technical Specifications Definition	11
Table 2 Quality properties and perspectives for architectural elements.....	17
Table 3 List of identified architectural components, assigned tasks and partners responsibilities	21
Table 4 HL-UC01_LL-UC01: Prosumers enrolment in demand response programs	43
Table 5 HL-UC01_LL-UC02: Contract Setting	46
Table 6 HL-UC01_LL-UC03: Potential energy flexibility evaluation.....	49
Table 7 HL-UC01_LL-UC04: Energy demand/production forecasting for day-ahead trading of flexibility.....	52
Table 8 HL-UC01_LL-UC05: Flexibility request	55
Table 9 HL-UC01_LL-UC06: Flexibility offering.....	58
Table 10 HL-UC01_LL-UC07: Flexibility acceptance	61
Table 11 HL-UC01_LL-UC08: Flexibility provisioning.....	63
Table 12 HL-UC02_LL-UC01: Prosumers registration with the energy trading platform	66
Table 13 HL-UC02_LL-UC02: Prosumers bids/offer submission.....	69
Table 14 HL-UC02_LL-UC03: Energy clearing price determination.....	72
Table 15 HL-UC02_LL-UC04: Transactions validation and financial settlement.....	74
Table 16 HL-UC03_LL-UC01: Prosumers Profiling and Clusterization	77
Table 17 VPP Capability Evaluation for Reserve services and for Frequency services.....	80
Table 18 HL-UC03_LL-UC03: VPP Export Evaluation for Wholesale market (Intraday trading) and for Imbalance market.....	83
Table 19 Smart Metrology Meter (SMM).....	96
Table 20 Smart Meter Extension (SMX)	97
Table 21 EVO Emotion	98
Table 22 OBD Emotion	99
Table 23 Fruit KiWi Power	101
Table 24 Micro-grid Monitor.....	103
Table 25 EVSEs and EV fleet monitoring.....	104
Table 26 Electricity Consumption/Production Forecasting.....	106
Table 27 Virtual Power Plants Generation Modeling & Forecasting.....	110
Table 28 Baseline Flexibility Estimation	115
Table 29 PV/RES Degradation & Trend Analysis.....	118
Table 30 Multi-building DR characterization through optimal, thermal and LIDAR information fusion.....	120
Table 31 Big Data Analytics Tool	123
Table 32 Load Profiling.....	125
Table 33 Big Data Clustering at Multiple Scales	127
Table 34 Customer Segmentation.....	129
Table 35 VPP and active Microgrid Flexibility Profiling.....	131
Table 36 VPP and DR Services Optimization Engine	133
Table 37 Distributed Ledger	136
Table 38 Blockchain-driven control for LV networks (flexibility management)	138
Table 39 Secured Blockchain-driven Energy Market.....	142
Table 40 Closed loop DR Verification Engine	146
Table 41 DSS (Decision Support System) & DR Strategies Optimization.....	149
Table 42 DR Aerial Survey Toolkit	151
Table 43 Graph-based Analytics.....	153
Table 44 Forecasting Tool.....	155
Table 45 HMI.....	157
Table 46 Electricity Consumption/Production Forecasting – FRs	162
Table 47 Virtual Power Plants Generation Modeling & Forecasting - FRs	162
Table 48 PV/RES Degradation & Trend Analysis - FRs	162
Table 49 Baseline Flexibility Estimation - FRs	163

Table 50 Multi-building DR characterization through thermal, optical and LIDAR information fusion - FRs.....	163
Table 51 Big Data Analytics Engine Tool - FRs	163
Table 52 Load Profiling - FRs	164
Table 53 Big Data Clustering at Multiple Scales - FRs	164
Table 54 Customer Segmentation - FRs	164
Table 55 VPP and active Microgrid Flexibility Profiling - FRs	165
Table 56 VPP and DR Services Optimization Engine - FRs.....	165
Table 57 Distributed Ledger - FRs	166
Table 58 Blockchain-driven control for LV networks - FRs	166
Table 59 Secured Blockchain-driven Energy Market - FRs	166
Table 60 Closed loop DR Verification Engine - FRs.....	167
Table 61 Graph-based Analytics - FRs	167
Table 62 Decision Support System & DR Strategies Optimization – FRs	167
Table 63 DR Aerial Survey Toolkit – FRs	168
Table 64 Forecasting Tool – FRs	168
Table 65 HMI - FRs	168
Table 66 EVSEs and EV fleet monitoring – FRs.....	169
Table 67 Electric meters, edge, and field device electric measures – FRs	169
Table 68 Distributed Ledger – NFRs.....	169
Table 69 Secured Blockchain-driven Energy Market – NFRs.....	169
Table 70 Big Data Clustering at Multiple Scales – NFRs	170
Table 71 Load Profiling & Disaggregation_ Customer Segmentation – FRs.....	170
Table 72 EVSEs and EV fleet monitoring – NFRs	170
Table 73 Big Data Analytics Engine Tools – NFRs	171
Table 74 HMIs – NFRs	171
Table 75 Functional and Non-Functional Requirements Template.....	172
Table 76 Architectural Components Detailed Specifications Template	172
Table 77 Smart Meters API	177
Table 78 Charging Station API.....	183
Table 79 Charging Session API	184
Table 80 Remote Start API	186
Table 81 Remote Stop API.....	186
Table 82 Remote Power Output Setup API.....	187
Table 83 Electric Vehicle API.....	187
Table 84 Electricity Consumption Forecasting - API.....	188
Table 85 Electricity Production Forecasting API.....	190
Table 86 Energy Flexibility Forecasting API.....	192
Table 87 Virtual Power Plants Generation Modeling & Forecasting - API	194
Table 88 Baseline Flexibility Estimation - API.....	200
Table 89 PV/RES Degradation & Trend Analysis - API	201
Table 90 Load Profiling - API	202
Table 91 Big Data Clustering at Multiple Scales - API.....	203
Table 92 Customer Segmentation - API	204
Table 93 VPP and active Microgrid Flexibility Profiling - API.....	204
Table 94 VPP and DR Services Optimization Engine - API.....	205
Table 95 Distributed Ledger API.....	206
Table 96 Secured Blockchain-driven Energy Market - API	208
Table 97 Blockchain-driven control for LV networks - API	210

eDREAM	enabling new Demand Response Advanced, Market oriented and secure technologies, solutions and business models
WP	Work Package
DSO	Distribution System Operator
DR	Demand Response
VPP	Virtual Power Plant
HL-UC	High Level Use Case
LL-UC	Low Level Use Case
MF	Macro-Functionality
BRs	Business Requirements
FD	Field Data
URs	User Requirements
EVSEs	Electric Vehicle Supply Equipment
EV	Electric Vehicle
UI	User Interface

The purpose of this deliverable, as the technical output of the project, is to present the second consolidated version of the system requirements and technical specifications for the eDREAM platform. The deliverable describes the steps and actions performed in Task 2.4 after the first year of the project and can be considered as a key input for the upcoming task in WP2 concerning requirement tracking (T2.5), as well as WP3, WP4, WP5, WP6 and WP7. Throughout the document, the main requirements and specifications of eDREAM platform are described in the scope of addressing the eDREAM objectives and innovation potential.

The purpose of this deliverable is to describe the second version of eDREAM Conceptual Architecture as well as the integrated system specifications along with the functional and non-functional requirements. This document provides a holistic view of the eDREAM overall Architecture, its building blocks, components, interdependencies among components, related constraints, such as development methodology and interfaces for data exchanges.

The concept of the architectural framework will mainly focus on deriving the specifications of the system's key components and their functionalities based upon User Needs and Business Requirements. Following the basic design principles, the following aspects will be addressed:

- Conceptual Architecture Design Process: within this part an overall view of the eDREAM architecture will be presented comprising the components, the interfaces between them and the connections with the external interfaces.
- Functional and Technical Specifications of Architectural Elements/Modules: the objectives of this section are the followings:
 - to provide a high level diagram of dependencies among the different parts of the framework;
 - to describe in detail the constraints of the system elements in terms of hardware and software resources, compatibility with standards, etc..

D 2.4: Requirement-Driven System Development V1 [M12]

1. Definition of the overall approach and methodology for elicitation of System dependencies between architectural components and requirements
2. Definition of the first set of the System dependencies and functionalities through internal elicitation
3. Refinement of the first set of the System dependencies and functionalities based on the first architectural workshop between Consortium partners

4. Refinement of the first set of the System dependencies and functionalities based on the first released document concerning Business and User requirements (D2.1)
5. Consolidation of the System dependencies and functional requirements based on the first consolidated version of use cases (D2.2) and the outcomes of teleconferences concerning use cases refinements & functional analysis.
6. Consolidation of the system technical specifications based on the final stage of internal elicitation

D 2.5: Requirement-Driven System Development V2 [M18]

1. Continuous assessment of the System requirements and technical specifications based on the outcomes of the parallel development activities during WP3, WP4 and WP5
2. Perform parallel activities with WP6 towards the definition of detailed modules interfaces and API for interoperability

D 2.10: Requirement-Driven System Development V3 [M30]

1. Continuous assessment and refinement of the system requirements based on the second consolidated version of Business and User requirements and Use Cases and Application Scenarios.
2. Use of the prototypes to refine the requirements
3. Final version of System requirements and technical specifications

D 2.5 “Requirement-Driven System Development V2” consists of eight chapters, in which the second consolidated version of System requirements, dependencies and technical specifications have been described as follows:

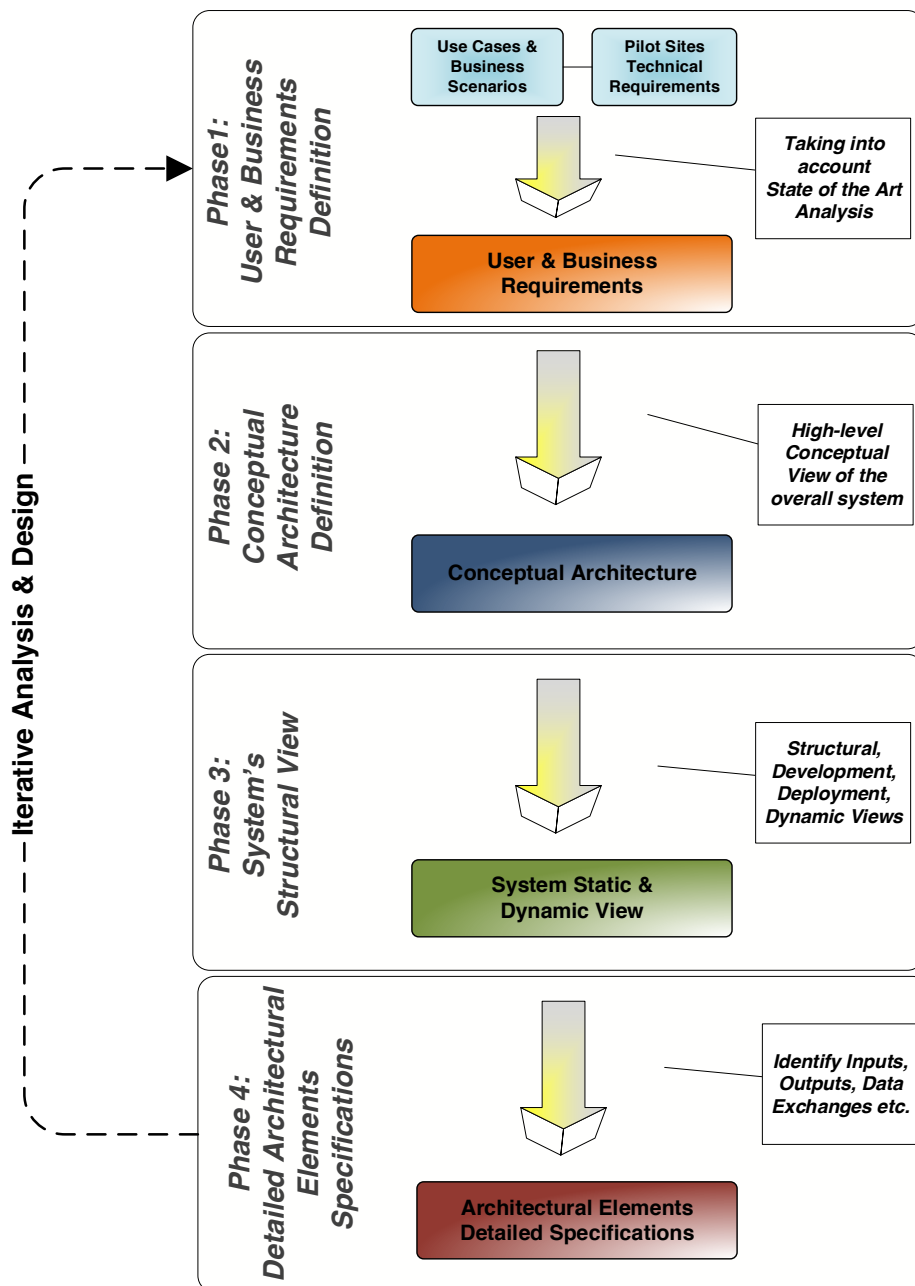
- **Chapter 1** presents the general description of the scope and objectives of the deliverable;
- **Chapter 2** describes the methodology, which will be followed during the architectural design in order to derive the detailed functional and technical specifications of the eDREAM system. It presents the basic architectural design concepts and principles adopted towards the outline of the different phases and the definition of the architectural layers and elements that compose the eDREAM system.
- **Chapter 3** presents the updated conceptual architecture of the eDREAM system through a high-level diagram introducing the 3 main layers comprising the eDREAM system
- **Chapter 4** describes the updated structural view of the eDREAM platform describing the different architectural elements/modules that provide the system’s functionalities. The system’s decomposition into different components is also presented during this section, demonstrating how each component carries out the required functions.

- **Chapter 5** presents an updated analysis of the dynamic behaviour of the eDREAM system through Use Cases and sequence diagrams. This dynamic view defines how the system actually works and what responses it gives to external or internal stimulus.
- **Chapter 6** depicts the updated deployment view of the eDREAM system covering the hardware requirements of the architectural components and tools to be used.
- **Chapter 7** presents the updated version of the system's detailed architectural elements specifications.
- **Chapter 8** provides the conclusions of the overall work.

Finally, the Functional and Non-Functional Requirements of the system components are presented in **Annex I**, the templates used for the internal elicitation of requirements and technical specifications are included in **Annex II** and as already mentioned before the APIs of field devices and architectural components are presented in **Annexes IV** and **V** respectively.

This section presents the approach and methodology that have been followed by Task 2.4 to define the first version of the architecture. Task 2.4 started on M3 and it will run continuously until M30. The first version of the architecture comprises the first consolidation of dependencies, inputs/outputs and specifications of architectural components. The next version is intended to present detailed information concerning the interfaces between the components based on the outcomes of technical work packages WP3, WP4, WP5 and WP6. Finally, the last version is going to present the detailed description of the whole platform in terms of architecture, modules, dataflow, processes, APIs specifications and interoperability issues.

The following figure presents the process for the system requirements elicitation until M12:



The overall architecture of a system is the composition of several architectural system structures, which comprise software elements, the externally visible attributes and properties of those elements along with the relationships and interfaces among them. It describes its different components and the way they interact towards carrying out the required functionality.

The representation of the conceptual architecture and its architectural elements enables communication among all stakeholders that are interested or concerned about the realization of the system. Definition of the overall system structure and orchestration among architectural elements are fundamental parts on the system development process, as architectural design decisions have a profound impact on all the development work that follows as well as on the accomplishment of the system tasks. Finally, all components that comprise the system shall take into account the concerns which are derived during the user and business requirements process with the actual involvement and engagement of the key stakeholders.

Following basic design principles, the architecture shall be open and modular, so that all vendors, suppliers, and potential users will be able to make use of what is in the functional part of the defined architecture. Furthermore, the architecture shall be as much as possible technology independent, based on standards and promote (when it is feasible) the use of generic and standardized solutions for which several key technologies (open source, commercial, etc.) are available.

Based upon the static and dynamic models, a set of **key design principles** have been defined and specified in order to ensure that architecture designers minimize costs, maintenance requirements and promote extendibility, modularity and maintainability. These can be categorized into the following:

- **Separation of concerns**, which outlines that the overall system/application should be divided into distinct features with as little overlap in functionality as possible. The ultimate goal of this principle is, from one hand to minimize interaction points and on the other hand, to ensure increased cohesion and low coupling.
- **Single Responsibility principle**, which outlines that each architectural element (e.g. core component of the system) shall be responsible for only a specific feature or functionality, or even aggregation of cohesive functionality.
- **Principle of Least Knowledge**, which defines that an architectural element (e.g. component or object) should not directly have access to the internal details of other architectural elements (e.g. components or objects).
- **Don't repeat yourself (DRY)**, which refers to the principle of avoiding repeating the same functionality or intent in more than one architectural elements of the system under design. Thus, according to this principle, common functionalities are addressed in more general architectural elements or components, which can be utilized by each separate element in order to "access" or "deliver" the required functionality.
- **Minimize upfront design**, which outlines that the design of more functionalities and methods than the ones needed for the system under design should be avoided. This principle mainly refers to the early stages of the architecture development process, when the design is likely to change over time. Thus, the architecture designers and developers shall avoid large designing and potential implementation of components at premature stages.

The key output of the architectural elements design process is the detailed definition of the conceptual architecture and the components that comprise the system, namely system's structures and its exposable attributes and properties. The system structures are divided in two complementary categories, the static (design-time orchestration) and dynamic (runtime orchestration):

- The **static structures** refer mainly to the design-time of the architectural elements of the system (objects, components) and the way they fit together internally. The static arrangement of the architectural elements depends on the actual context of use and provides information such as associations, relationships, or connectivity among them. For instance, relationships define how data items (either inputs or outputs) are linked to each other. In hardware, the relationships provide the required physical interconnections between the hardware components and the sub-systems comprising the overall system.
- The **dynamic structures** of a system illustrate how it actually operates during its utilization, depending on the various scenarios of use and use cases defined, including the way each component acts within them. Thus, the system's dynamic model and structures define its runtime architectural elements and their interactions due to internal or external stimulus. The internal interactions refer to information flows among architectural elements and their parallel or sequential execution of internal tasks, including the potential expression of the effect they may have on the information.

The eDREAM project adopted a participatory design (PD) process, where all the relevant stakeholder groups could actively participate during the lifetime of the project. This facilitates the coordination between user and business requirements definition and functional requirements and technical specifications definition. This approach is based on iterative cycles concerning capturing of end-user and business needs as a reference point for the overall design, implementation and evaluation process.

In the context of eDREAM, we will use the 4+1 architectural view model, so as to present the concurrent views.

The concept is depicted below:



Conventional views and viewpoints approaches provide meaningful information to the architecture derivation process and in the definition of the various architectural structures. However, to broaden the modularity, reliability and credibility of the system under design it is useful to outline and consider specific quality properties during the final stages of the architecture definition process. Towards defining the architectural elements of eDREAM, their dependencies and the respective architectural vies, the architectural perspectives are also considered, which are analogous to a viewpoint, as they were described in detail for the structural/functional, dynamic development and deployment views. In this report, several quality properties are addressed for all architectural elements of the system, as these are outlined in the following table:

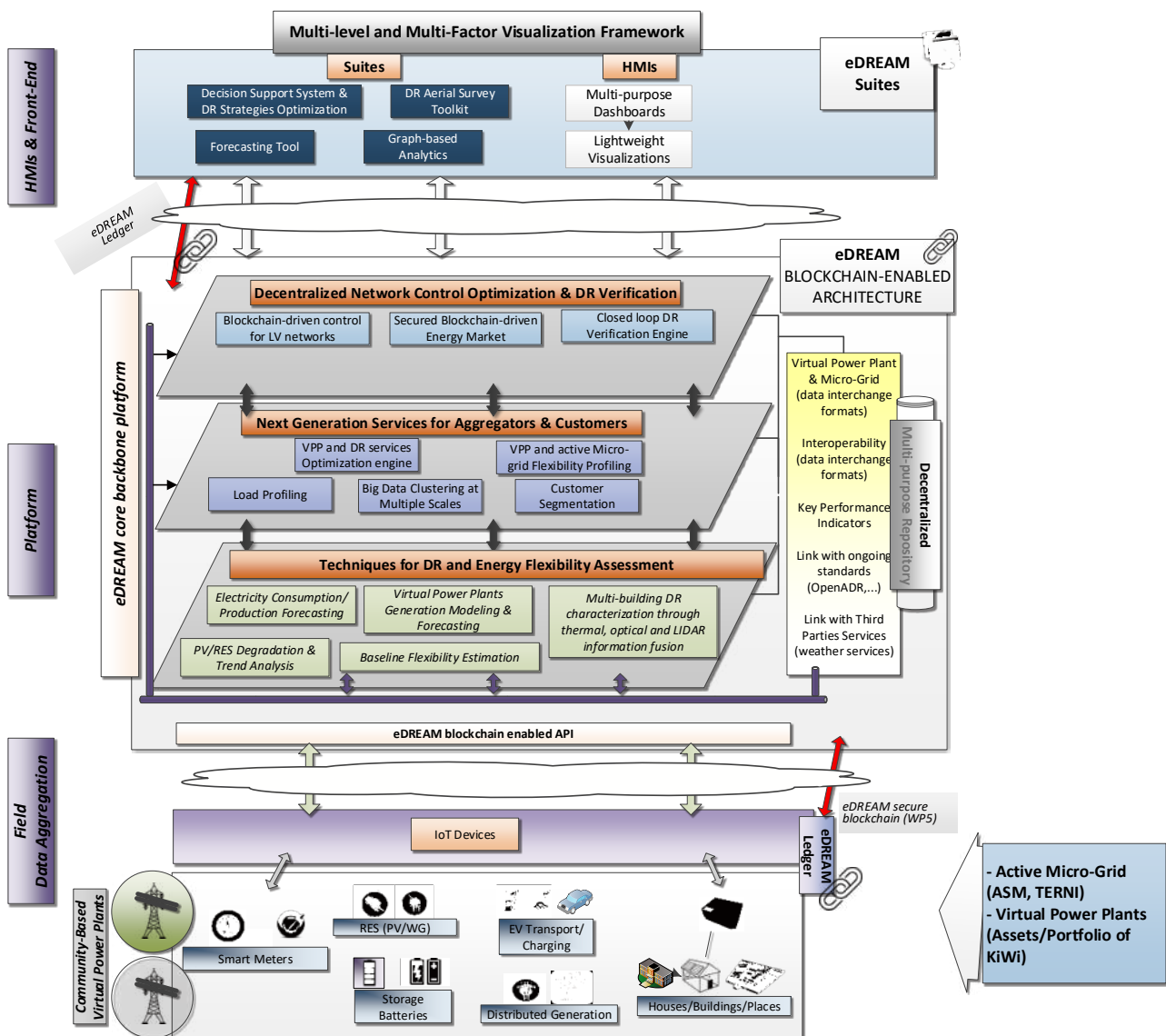
<i>Perspective</i>	<i>Desired Quality</i>
<i>General Purpose</i>	
<i>Performance and Scalability</i>	<i>The ability of the system as a whole including its architectural elements to predictably execute within its mandated performance that cope with system requirements and is able to handle increased processing volumes of information.</i>

<i>Availability and Resilience</i>	<i>The ability of the system as a whole to be fully or partly operational as and when required and to effectively handle failures on all levels (hardware, software) that could potential affect system availability and credibility.</i>
<i>Security</i>	<i>The capacity of the system to reliably and effectively control, monitor and additional audit if the policies defined are met (e.g. what actions on what assets/resources) and to be able to recover from failures in security-related attacks.</i>
<i>Evolution</i>	<i>The capability of the system and its architectural elements to be flexible enough in the case of non-foreseen changes during deployment or installation process.</i>
<i>Additional Perspectives to cope with eDREAM non-functional requirements</i>	
<i>Maintenance</i>	<i>The ability of the system to comply with coding guidelines and standards. Includes also the functionality that needs to be provided to support maintenance and administration of the system during its operational phase.</i>
<i>Privacy & Regulation</i>	<i>The ability of the system and its architectural elements to conform to national and international laws, policies and other rules and standards.</i>
<i>Usability</i>	<i>The ease with which key stakeholders of a system are capable to work effectively and to interact with it in a user-friendly way.</i>

For each of the aforementioned perspectives, the importance on the four views of the eDREAM framework may vary and the benefits of addressing them are essential towards providing a common sense of concerns that shall guide the architectural elements definition process and their later implementation and deployment to the validation and integration phase. In this respect, it is anticipated that by addressing in the architecture definition process the importance of the aforementioned perspectives will further help to the later decision making (implementation, deployment and operational phases). Within eDREAM, a table will be provided for the eDREAM system (for both frameworks) in order to ensure that all concerns and non-functional requirements are addressed and to exhibit what quality properties are considered within the system and which architectural elements contribute towards fulfilling them. In order to ensure that the eDREAM architectural model will meet the functional and non-functional requirements, but the above proposed

perspectives should be taken into account. These perspectives could be modified or enriched by partners according to characteristics of the components.

This chapter provides an overview of the eDREAM conceptual architecture introducing the major layers and sub-layers of the eDREAM platform along with the included architectural components. eDREAM's vision is to develop, validate and deliver a decentralized and secure closed loop Demand Response ecosystem enabling the seamless cooperation between DSOs and aggregators in the scope of maximizing exploitation of the flexibility potential of a large variety of heterogeneous loads and generation assets. During the lifetime of the project, novel functionalities and services will be researched and examined by using the principles of Internet of Things (IoT), the concepts of Demand Response programs and the blockchain-driven technology. The following figure presents the conceptual architecture of the eDREAM platform:



The main layers and sub-layers of the eDREAM platform are described in brief below:

- The first layer of the architecture is the **Field Data Aggregation**, which is the interface with the physical world through smart metering devices and communication interfaces. The use of IoT devices provides access to real-time data from the pilot sites providing the electrical measurements of field devices installed in the Microgrid and the Virtual Power Plants (e.g. VPP data collected in an aggregated manner through KiWi smart gateway). This layer will forward the necessary real-time information to the upper layers of the platform, in order to enable the functional architectural components to perform their analysis and calculations. The information exchange will be based on open communication specifications (based on xsd data schemas and Rest services) that realize the Machine2Machine (M2M) communication through which the data, information and actions are dispatched to the appropriate field device or upper layer of the platform.
- The main layer is the **Core Backbone Platform** which is the fundamental part of the conceptual framework. It includes all the necessary components and mechanisms to support the structure of a decentralized ecosystem for closed-loop DR programs. The aim of this platform is the combination of components in the scope of research and development for providing improved services to the system's stakeholders. This layer includes three hierarchical connected sub-layers of which are together connected with a Decentralized Multi-purpose Repository. These are the followings:
 - **Techniques for DR and Energy Flexibility Assessment (WP3)**
The aim of this sub-layer is the development of innovative electricity consumption and production forecasting mechanisms for registered prosumers in the aggregator's portfolio towards improving the short-term and long-term forecasting of energy demand and generation. The outcomes of these algorithms and techniques will support various programs in the energy market, such as day-ahead, direct trading and coupon-based DR programs. In addition, advanced models for multiple types of distributed generation resources will be investigated with the point of creating optimal coalitions and providing more reliable aggregated supply. The constitution of these coalitions will also enable small prosumers, such as households of 1kW capacity generation to participate in DR programs. Furthermore, novel techniques based on aerial surveys techniques will be designed and developed for pre-evaluation of new customers in DR programs, thus giving to aggregators valuable insights for the improvement of their business plans.
 - **Next Generation Services for Aggregators and Customers (WP4)**
The main purpose of this sub-layer is to provide to the system stakeholders (Aggregators, DSO etc.) all the necessary services, so as to be able to calculate and extract all the necessary key features and parameters for their customers/prosumers at different scales. These parameters are related to load and generation profiling and they will help aggregators towards optimal DR strategies classification and scheduling. Innovative machine learning techniques for load profiling and disaggregation at multiple scales (e.g. micro-grid level, virtual power plants and in lower loads related to Distributed Energy Resources) will be investigated. A Big Data Analytics Engine will be researched and developed for analyzing large streams (including micro batch level) collected from customers. Furthermore, big data clustering techniques at multiple scales will be investigated towards creating customers' clusters with specific load and generation profile patterns.

○ **Decentralized Network Control Optimization & DR Verification (WP5)**

This sub-layer will mostly focus on research and development concepts for decentralized network control and financial transactions. Its main goal is to investigate the use of blockchain platforms in DR modelling, distributed control and validation. A shared and replicated blockchain distributed ledger at grid level is going to be developed and implemented in order to ensure secure and reliable storage of energy transactions and DR flexibility services. In addition, the definition and implementation of self-enforcing smart contracts for tracking and controlling the energy transactions and DR flexibility services in smart energy grids will be performed in a fully decentralized manner. Finally, Proof-of-Stake consensus based algorithms for closed-loop DR programs execution, verification and financial settlement will be examined. In addition, the delivery of a Graph-based Analytics platform will support automated closed-loop DR programs, while providing hypothesis testing framework for multi-factor parameters analysis and DR programs improvement.

- The upper layer **HMIs & Front-end for end-users and operators (WP4 & WP6)** contains accessible and easy-to-use HMIs (e.g. accessible by mobile phone through lightweight visualizations) for end-users and operators that enables vertical collaboration (from the DSO and aggregators to prosumers/consumers) and horizontal collaboration (using virtual topologies, such as the community-based VPPs) within the eDREAM architectural framework. The main purpose of this layer is the visualization of the output data from the Core Backbone Platform which are additionally analyzed and interpreted. Bidirectional data flow is performed between the core platform and the front-end layer, since several decisions of stakeholders are based on the provided results from the components of the core platform.
- The Core Platform is connected with a **Decentralized Multi-purpose Repository (WP5)** which allows data exchanges within eDREAM core framework. This component provides the necessary place for storage and maintenance of data from field devices, data models/profiles for supporting the functions of core components and information from third parties' services (e.g. weather services).

As mentioned above, during the **bottom-up process** of the architecture definition all the technology provider partners were identified. The main purpose of this phase was the identification of the architectural components that should be developed and the corresponding partner/s. During the first round of information collection, a basic template was created and circulated with requested information concerning main functionalities, dependencies, inputs needed and outputs provided. The list of architectural components along with the assigned tasks and associated partners responsibilities is presented in the table 3.

Component	Related Task	Responsible partner	Contributing partners
Electricity Consumption/Production Forecasting	T3.1	TUC & CERTH	TU, ENG
PV/RES Degradation & Trend Analysis	T4.1	TU & CERTH	ENG, SVT

Baseline Flexibility Estimation	T3.2	TU	TUC, E@W, EMOT
Virtual Power Plants Generation Modeling & Forecasting	T3.3	TUC	EMOT, ENG, ASM
Multi-building DR characterization through thermal, optical and LIDAR information fusion	T3.4	TU & CERTH	KIWI
Load Profiling	T4.2	ATOS & CERTH	E@W, KIWI, ASM
VPP and active Micro-grid Flexibility Profiling	T4.1	TU	ENG, CERTH, SVT, ATOS
Big Data Clustering at Multiple Scales	T4.2	ATOS & CERTH	E@W, KIWI, ASM
Customer Segmentation	T4.2	ATOS & CERTH	E@W, KIWI, ASM
VPP and DR Services Optimization engine	T4.1	TU & CERTH	ENG, SVT
Distributed Ledger	T5.1	ENG	TUC, E@W, ASM
Blockchain-driven control for LV networks	T5.2	TUC	ENG, EMOT, ASM
Secured Blockchain-driven Energy Market	T5.2	TUC	ENG, EMOT, ASM
Closed loop DR Verification Engine	T5.3	ENG	TUC, E@W, ASM
Graph-based Analytics	T4.3 & T4.4	CERTH	TU, ATOS, E@W, KIWI, ASM
HMIs	T4.3, T4.4 & T6.2	CERTH	ATOS, TU, E@W, ENG, KIWI, ASM, EMOT
DSS (Decision Support System) & DR Strategies Optimization	T4.1, T4.3, T4.4 & T6.2	TU & CERTH	TU, E@W, ENG, SVT, EMOT
DR Aerial Survey Toolkit	T3.4 & T6.2	TU & CERTH	KIWI, ATOS
Forecasting Tool	T3.1 & T6.2	TUC & CERTH	TU, ENG, ATOS

For all the components, an updated detailed description template is provided in Annex II including the currently known technical specifications. The following chapter provides the updated structural view of the eDREAM architecture and presents the main functionalities and dependencies for each architectural component.

The structural view presents the different architectural elements that deliver the system's functionalities to the end-users. In the context of this view, the individual system's components have been identified and defined along with their high-level dependencies and interfaces in relation to other components. The functional system model includes the following elements:

- **Functional Components** constitute of clearly-defined parts of the system that have specific responsibilities, perform distinct functions and dispose well-defined interfaces that allow them to be connected with other components.
- **Dependencies** are channels, indicating how the functions of a component can be made available to other components. An interface is defined by the inputs, outputs and semantics of the provided operation/interaction.
- **External (third-party) entities** are connectors (described as dependencies) which represent other systems, software programs, hardware devices or any other entity that communicates with the system.

The following sections have been updated in terms of the defined architectural components with their main functionalities and the dependencies from other components. The initial definition of the modules interfaces and APIs have been completed and presented in the following sections. This version will be used as a basis for the final version of the document that is going to be delivered on M30.

The Figure 4 depicts an updated version of the eDREAM overall structural view with the main flows of information.

This layer is the bottom layer of the eDREAM system and refers to the communication interfaces with the field devices. The Field Middleware is the basic interface with the physical world and performs primary information processing based on the received raw data from smart meters and the other field devices. In addition, it provides semantic context interpretation of physical signals according to the identified ontologies and standards. In the following subsections, the main elements and concepts related to the field data layer are presented. The below figure presents how the integration of data from pilots sites will be implemented and the communication paths with the Repository of the project.



Description – Main Functionalities

Each **IoT Device** is a software representation of a field device, such as a smart meter, an EV charger, a diagnostic device or a more complex system like a processing station. An IoT Device exposes a set of operations, for setting status, performing actions or reading current data values. The device representation will also describe the nature of the devices itself and the characteristics of the shared information (e.g. type and unit of measured data). The IoT devices are the first level that ensures proper information transmission

to the upper layers of the architecture. They ensure data harmonization and seamless information exchange relying on open communication specifications and Machine2Machine (M2M) communication standards.

Description – Main Functionalities

The **Semantic Context Broker** will be implemented using the FIWARE Orion Context Broker¹ that is an open source software for creating different context elements based on the received data and manage them. This component provides the capabilities of producing, gathering, publishing and exploiting context information at large scale. Through this software, the context information is represented through values assigned to attributes that characterize the entities relevant to the received measurements. The Context Broker is able to handle context information at large scale by implementing standard REST APIs. One of the most important features of the Context Broker is that it allows to modeling and getting access to context information irrespectively of the source of this information. This component is based on the concepts of the NGSI model for the management of entities, attributes and context information. The functionality of the APIs designed or selected to connect the CMP with the Field Middleware layer are related to reading inputs (registers, variables and parameters), writing outputs (registers, variables and settings), handling alarms and events and manage security features. The most common APIs and standards related to smart grids and Demand Response programs are the following:

- **RESTful API** is a web service designed in accordance with the Representational State Transfer (REST) paradigm. It is not directly linked with any particular platform or technology, although HTTP is the preferred communication protocol due to its widespread use.
- **MQTT** – Message Queuing Telemetry Transport – is an M2M/IoT connectivity protocol. It was designed as an extremely lightweight publish/subscribe messaging mechanism over TCP.
- **OPC** is the interoperability standard for the secure and reliable exchange of data in the industrial automation space. It is platform independent and ensures the seamless flow of information among devices from multiple vendors. The specifications of OPC provide separate definitions for accessing process data, alarms and historical data. This standard specifies the software interface for a server that collects data produced by clients (e.g. field devices, controllers etc.).
- **IEC 61850** is a multi-part standard that defines interoperable information exchanges between intelligent electronic devices from multiple vendors in electrical substations using TCP/IP.
- **OpenADR** – Open Automated Demand Response – an open and standardized way for electricity providers and system operators to communicate DR signals with each other and with their customers using a common language over any existing TCP/IP based communications network.
- **IEEE 2030.5 (SEP 2.0)** is an industry effort to promote the interoperability between metering and home energy management systems, supporting device types like gateway, metering devices, thermostat and load control devices. The standard uses IEC 61968 (CIM) as a “dictionary” and a RESTful architecture.
- **IEC CIM**: It represents the main resources for the management of the electric system.
- **Facility Smart Grid Information Model (FSGIM)**: It defines information that must be exchanged between electricity providers and electricity consumers and guides the evolution of control technologies used to manage loads and generation sources in facilities.
- **Enery@home**: It transforms the home environment in an eco-system of devices that communicate with each other.

¹ <https://github.com/telefonicaid/fiware-orion>

- **NAESB Energy Usage Information Model:** It describes energy usage information. NAESB is also consistent with IEC TC57 CIM and ZigBee Smart Energy Profile 2.0.
- **OASIS Energy Market Information Exchange (eMIX):** It describes the exchange of price and product information for the power and energy markets.
- **ETSI M2M** (evolved to **oneM2M**) representing Machine-to-Machine communications: It is an application agnostic standard containing an overall end to end M2M functional architecture, identifying the functional entities and the related reference points. It can be used for the exchange of data and events between machines involving communications across networks without requiring human intervention.

The use of standard protocols at any level of the architecture is one of the best ways to ensure interoperability that is one of the most important non-functional requirements of the project. This means that the field devices can be easily replaced in case of malfunction, the system can be more easily expanded and more efficient and less expensive devices can be procured. For the integration layer it means that the information is more easily exchanged with existing applications or that the required development is minimal. In addition, the management and security features are more widely understood and up to date.

Along with the standards and communication protocols, the ontologies related to smart grids and Demand Programs have been considered:

- **SAREF4ENER:** It is an extension of SAREF created in collaboration with the EEBus and Energy@Home industry associations to interconnect their (different) data models.
- **MAS2TERING:** It describes the message exchange between the agents for the smart grid, the agents and their behaviours, and the constraints.
- **OEMA Ontology Network:** Ontology network to unify existing heterogeneous ontologies that represent different energy-related data, such as equipment or infrastructure.
- **CIM** ontology for Smart Grids: It is a profile of the IEC Common Information Model for Smart Grids developed by the Cerise-SG project.

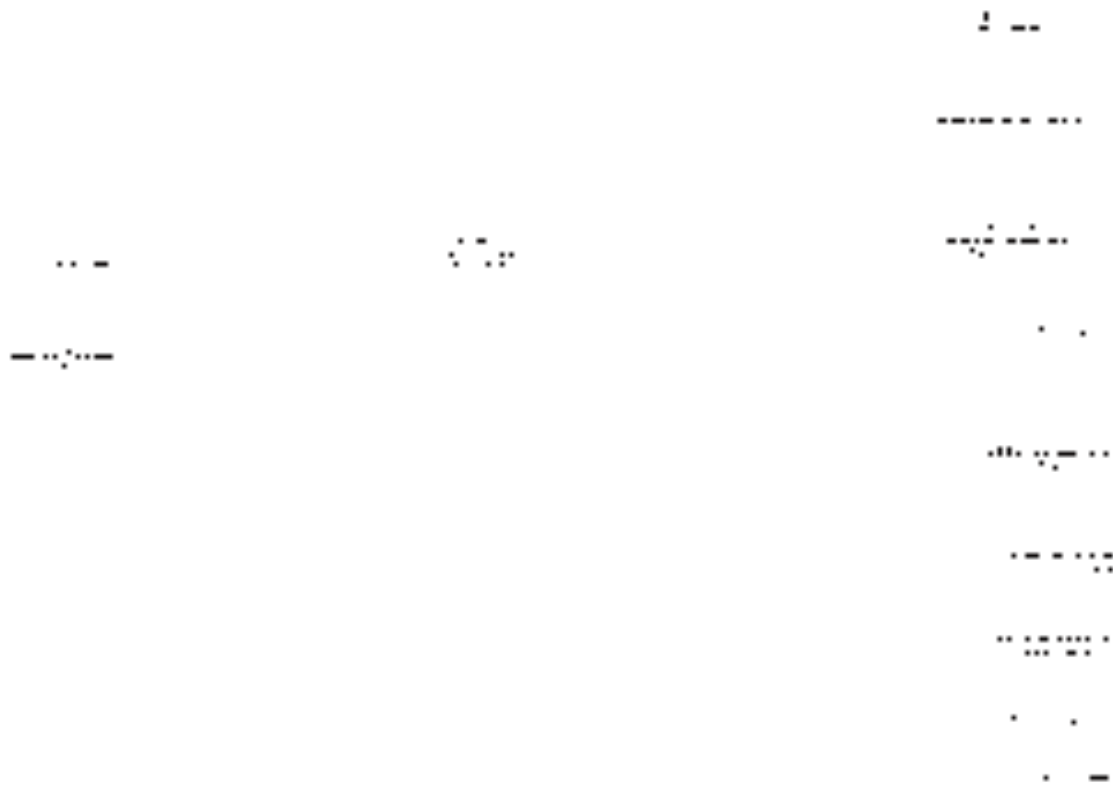
Dependencies to other components

- The two basic dependencies of Field Middleware is that it receives raw/low level data from field devices and translates them to semantically-enhanced outputs that are forwarded to the components of the Core Backbone Platform.
- The Field Middleware will use context data from the Decentralized Repository according to respective standards (e.g. OpenADR, CIM etc.) in order to enable enhanced data structures.

Description – Main Functionalities

This component is responsible for detecting prosumer's energy consumption/production patterns and delivering accurate predictions of energy supply and demand at different levels of granularities (scale/time). The energy consumption / production forecasting methodology, proposed in the context of eDREAM project, is based on the meta-learning model featuring an ensemble prediction architecture that aggregates a set of prediction models and combines their prediction results using a weighted average with dynamically computed weights at run-time as a function of input features. The model is able to learn the values of the meta-parameters represented by the weights used in the ensemble process by dynamically benefiting on the

advantages brought by each individual prediction model adapted to the input data set features. According to the defined scenarios and use cases, three prediction time horizons have been identified: **day-ahead**: energy values for the next 24 hours with a granularity of 1 hour; **intra-day**: energy values for the next 4 hours with granularity of half an hour and **1hour-ahead**: energy values for the next hour with granularity of 1 hour. For the identified time frames, both energy based features determined based on the historical data acquired by prosumers on-site smart meters and contextual features (such as season, week days, calendar days, etc.) have been considered. In addition, a model for energy flexibility forecasting at prosumer level has been developed. The considered energy flexibility is defined as the degree in which the prosumer can modify his/her baseline energy profile either by increasing, or decreasing the load. The methodology that has been followed for the development of this component along with relevant results are described in detail in the deliverable D3.1.



Description – Main Functionalities

The functionality of this component is to calculate the degradation rate (Rd) at which PV systems and other RES modules lose their performance over time. This is a significant information for **long term energy production estimation**. Various methods and techniques are currently being investigated for calculating PV degradation rate, such as regression modelling, normalized and scaled ratings, measurement qualification and filtering, statistical analysis and year-on-year degradation calculation. In addition, it provides improved **short-term forecasting of generation** based on near real-time trend analysis algorithm, which detect the uncertain changes in production, such as due to weather conditions. This component will be further described in the deliverable D4.5.



Description – Main Functionalities

The main functionality of this component is to calculate the flexibility of the **Customer Baseline Load (CBL)**, in order to determine the prosumers' participation in appropriate types of DR programs. In the context of eDREAM use cases, four methods of baseline flexibility calculation have been currently investigated: Comparative day, X of Y medium, Addition adjust and Recursive least square. The descriptions of the models that have been tested and the corresponding results are presented in detail in the deliverable D3.2.



Description – Main Functionalities

The responsibility of this component is to develop models for distributed generation of electricity of various types (e.g. wind-turbines, small hydro, photovoltaics, back-up generators, etc.) in order to create optimal dynamic coalitions (Virtual Power Plants - VPPs), which will provide a more reliable aggregated power supply. The types of VPPs that have been considered are the following:

- Distributed Energy Generators (DEG), such as small-scale wind power plants, photovoltaic units, CHP systems, diesel generators, etc.;
- Energy Storage Systems (EES), such as batteries, UPS;

The functionality of this component is to provide a number of well-defined and separate clusters of prosumers. This will be a scalable procedure for extracting a number of clusters from a large amount of users' load curves, in order to be able to apply this procedure to heterogeneous data sets with a variable number of load curves. This means that the same technique can be applied with different constraints and starting conditions. As presented in the deliverable D4.2, the clustering process can be considered the core process of the big data layer, which is responsible for the extraction of the most valuable information from the received data. The main objectives of this component within the eDREAM project are the following:

- flexibility evaluation, that means helping energy market operators to quickly identify portions of customers with given flexibility potentials;
- portfolio assessment, that is a general assessing of prosumers for proper market participation.

An important step during the process is the Attributes' selection for clustering load patterns and back-up generators. This is important, because it can contribute to the proper creation of the requested clusters according to predefined criteria. Finally, it should be mentioned that the tool supports the appropriate clustering algorithm selection according to the following parameters:

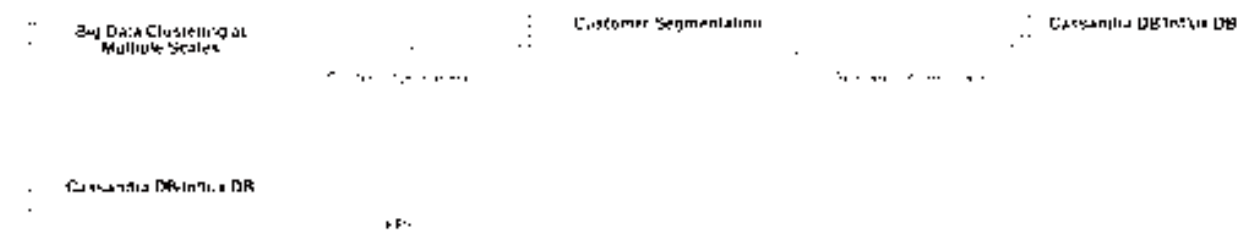
- the type of data used for input;
- the way the algorithm evaluates the similarity between data points;
- the basic theory a clustering algorithm is based on (e.g. fuzzy theory, statistics).



Description – Main Functionalities

This component is responsible for the assignment of the customers to a particular customer group (cluster) by recognizing the customer's load profile pattern. Segmentation of prosumers will be also useful for categorizing the participation of small and medium generation to different energy markets (ancillary services, balance market etc.). Within eDREAM project, Artificial Neural Networks (ANN) have been chosen for the development of the segmentation component. The application of these algorithms provides a series of

advantages that allow to classify the profiles of new clients without the need to group them again, once trained. This process is included in the deliverable D4.2.



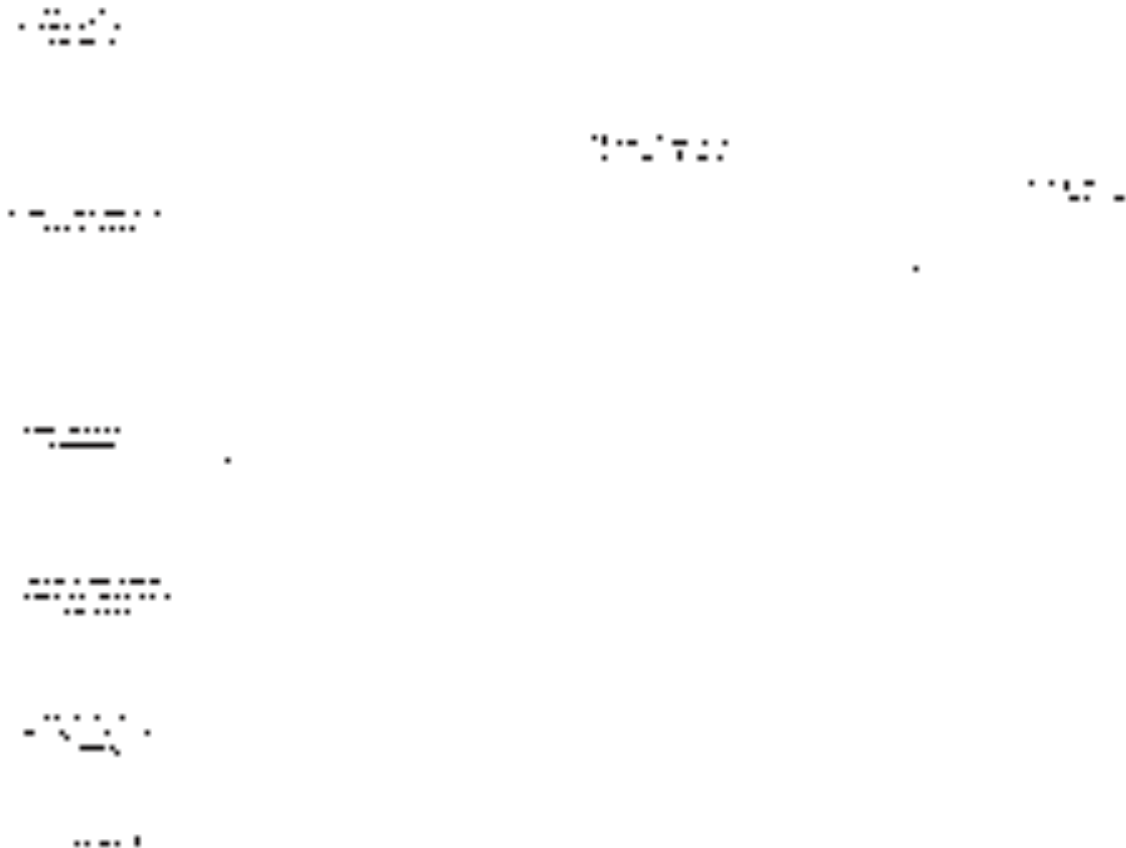
Description – Main Functionalities

The primary functionality of this component is to provide the flexibility margins of the prosumers generation assets. If the prosumers loads are also considered, the overall VPP and Microgrid flexibility (e.g. active assets in microgrids fully or partially connected to the grid, RES generation, other DERs etc.) can be calculated. The prosumers, directly or via enabling aggregators, will be able to offer their production assets as flexibility resources. By using this component, the Aggregator can exploit the VPPs and microgrid flexibility, in order to be able to manage DSOs requests in case of grid instability.



Description – Main Functionalities

The core functionality of this component is the optimization process, in order to provide optimal DR scheduling of the flexible assets (e.g. DERs, Microgrids etc.) that are registered in the Aggregator's portfolio. This will help to the efficient management of flexibility requests from the DSO. During the process, the conventional generators and their attributes should also be considered. The optimization problem can be formulated as a customized unit commitment and economic load dispatch problem considering DR events as flexible resources and taking into account customer constraints and environmental indicators. For the solution of the problem, a mixed integer linear programming solver is proposed. The problem formulation along with the requirements and the relative techniques are presented in the deliverable D4.1.



Description – Main Functionalities

A Distributed Ledger (or Distributed Ledger Technology, DLT) is defined as a distributed, tamper-proof database, not controlled by a single institution. A Blockchain is a distributed ledger, in which data records are grouped in blocks and chained through cryptographic hashes. In eDREAM context, the implementation of a blockchain distributed ledger at the micro-grid level has been proposed, in which all energy monitored data is registered at the level of an individual prosumer (or any other distributed energy resource) and then stored

as immutable energy transactions. The individual energy transactions are aggregated in blocks, which will be replicated in the ledger. The prosumer is modelled as a node of the peer to peer distributed energy network (i.e. a graph of peer nodes) and will maintain a copy of the ledger which will be automatically updated when new energy transactions are being registered. Other energy players, such as the energy aggregators or the DSO that are interested in micro-grid management are also registered as peers. The proposed technology and the first prototype implementation of this component are described in detail in the deliverable D5.1.

The main functionalities of this component can be summarized to the following:

- Enable the other components of the platform to access the stored data;
- Ensure secure storage scalability and transaction speed;
- Enable the storage of energy transactions in a secure and tamper proof manner;
- Grant access to data only to authorized users.

Dependencies to other components

This component can interact with all the components of the platform, as it can provide a secure place for data storage of the information produced and exchanged.

Description – Main Functionalities

This module introduces the concept of self-enforcing smart contracts for modelling and controlling DR flexibility services and Energy transactions in low voltage grids. The aim of this module is to participate in the prevention of future congestion points in the grid by evaluating the flexibility offers from the aggregators, choosing the best offers and tracking the monitored activity. In addition, it should monitor and control the prosumer activity to follow the corresponding promised flexibility and DR agreement. The main functionalities can be summarized to the following:

- Process flexibility requests to aggregators;
- Selection of flexibility offers from aggregators;
- Communicate flexibility requests to prosumers;
- Communicate flexibility availability of prosumer to aggregators;
- Selection of prosumers from portfolio to meet a specific aggregated flexibility;
- Track and control the flexibility delivery from prosumers to aggregators.

More specifically, the prosumers are able to offer and trade their flexibility in terms of loads modulation, indirectly via enabling aggregators or directly with the DSO via direct DR and control of their energy assets. The energy stakeholders (DSO or aggregators) will be able to assess and trace the share of contracted flexibility services, actually activated in real time by the enrolled prosumers through self-enforcing smart contracts enabling both demand-offer matching and decentralized coordinated control. The prototype implementation of this component is reported in the deliverable D5.2.



Description – Main Functionalities

This component provides a market session enforced by smart contracts allowing the registration of demand and offer actions and the computation of the clearing price and the matching actions. The aggregators/producers/prosumers will be able to offer their services by reacting to changes in the price of energy compared to the reference value, thanks to the trading marketplace that will be created using the smart-contracts. The producers/prosumers trade energy in a peer to peer fashion either directly or through an energy aggregator if they are not big enough. The prototype implantation of this component is reported in the deliverable D5.2. In brief, the main functionalities of this component are the following:

- Ensure energy transactions security;
- Track and control the registration and validation of prosumer;
- Publish bid/offer actions by prosumer;
- Perform energy bids/offers matching and clearing price computation.



Description – Main Functionalities

The purpose of this component is the monitoring and verification of the services matched through the platform between DSO and prosumer and DSO and VPP manager (e.g. production/load modulation). Then prosumers and VPP will be billed or remunerated accordingly. The verification process will be also used to penalize actors who don't comply with the agreements. More specifically, it monitors the outputs from the smart contracts (matched) defining prices, penalties and services. The verification process will be used for 1) Billing, 2) User Remuneration and 3) Penalization. Proof-of-Stake algorithms for miming the next valid block and validating associated DR transactions/services in the blockchain will be examined. Distributed consensus algorithms will be implemented to deal with transaction and DR flexibility services validation and financial settlement. The main goal is to implement a novel Closed Loop blockchain-based DR validation towards the direction of increased reliability of the DR system and improved reliability of DSO operation. Concept like Ether to smart grid DR management case and use the total rewarded DR incentives and their age a guarantee of blocks validation process will be specialized. The process randomization will be assured by implementing solutions similar to the one provided PPCoin peer-to-peer cryptocurrency which combines flip coin randomization with the coin age as factor. The main functionalities of this component are the following:

- Validate DR flexibility actually provided (at prosumer level);
- Register energy bids/offers in Marketplace;
- Perform matching of energy demand with energy production;
- Perform mining of new blocks of energy transactions;
- Settle Accounts according to DR Flexibility Validation.



The upper layer of the eDREAM system represents the integrated Visualization Framework that enables enhanced user interaction. This layer receives data from the Cross Backbone Platform layer, that are additionally analysed and interpreted. The main goal of this analysis is to enable all of the stakeholders (DSOs, Aggregators and prosumers) to share a comprehensive view of the provided core platform capabilities, obtaining a more active role during the core platform operation. As a result, bidirectional data flow is carried out between the core platform and the front-end components, since several stakeholders take decisions based on the provided core platform results. These decisions will be returned to the core platform as input values to the Decision Making & Optimization Mechanisms, leading to an iterative process with the main scope to conclude to the optimal solution for the eDREAM system.

The Visualization Framework includes two main groups of components that are:

- HMI
 - Remote HMIs via web applications (mobile interface)
 - Local HMIs (field level, standalone applications)
 - Multi-purpose Dashboards
- Suites
 - DSS (Decision Support System) & DR Strategies Optimization
 - DR Aerial Survey Toolkit
 - Forecasting Tool
 - Graph-based Analytics

Human Machine Interfaces (HMIs) are used as the interface between the different processes and the operators (stakeholders or users). In the context of eDREAM architecture, two types of HMIs can be defined, based on their capabilities concerning parameter configuration and thus, amount of information they handle. Firstly, a number of Multi-purpose Dashboards is used at grid-operator level and secondly, every device can be equipped with a Local HMI and every user/prosumer can be supplied with a Remote HMI (e.g. via a mobile application or a web service). Every HMI, either a dashboard or a local/ remote one, is able to display near real-time operational information tailored to each stakeholder's ability to coordinate and control the entire system. From the point of users, they can supervise the current state of their devices and enter information according to their energy plans. Detailed setting parameters will be provided from the operator, in order to set the constraints for the analysis needed. The role of these components focuses on that the decisions taking from stakeholders or the end users will directly affect the operation of the network. The main functionalities of the HMIs are:

- Real-time monitoring of the entire system;
- Detailed user-friendly analysis based on real-time energy consumption/production and flexibility values will be provided;
- Configuration of the desired high-level strategies.

More specifically, the HMIs can be separated into:

Multi-purpose Dashboards:

These components will be developed in order to deliver all necessary information that represents fully the operation of the grid in a user-friendly graphical way, via a series of statistical diagrams, real-time flow diagrams, and comparative visualisation with historical data etc. to provide a visualization of different aspects related to eDREAM. All these different aspects regarding the grid operation range from field level data (e.g. data from smart meters etc.) to KPIs' evolution related data. Stakeholders can use the information delivered by dashboards to plan their actions regarding the DR strategies planning and the services that can be offered to the grid by evaluating the results achieved by the strategies followed in recent time.

Remote HMIs:

Remote HMIs (such as mobile or web applications) enable specifically the end users to have access related to their assets and the respective constraints. In addition, information about their participation in DR programs are also visualized, such as completed, current and pending DR events, incentives, points gained etc.

Local HMIs: These HMIs visualize the current state of the connected devices at field level and also provide data for their foreseen operation, such as plan the charging time of EVs etc.

HMIs constitute the interface between the core platform components and the eDREAM stakeholders. HMIs can also be dependent to other components of the front-end layer. Based on the information received from the rest front-end components, HMIs can be utilized for visualising available collected data, either historical or real-time.

This group of front-end tools comprises dedicated applications for specific purposes, that are accessible to system stakeholders via web interfaces. These tools present different aspects of scenarios according to some defined parameters and provide useful representations. The components of this group are the following:

- **Decision Support System & DR Strategies Optimization Engine:** This component is in direct communication with the component VPP & DR Services Optimization Engine with the core platform



- **Forecasting Tool:** This tool receives signals and information from the Electricity Consumption/Production component, so as to provide aggregators with accurate forecasted data with a standard pre-defined frequency rate. This data is visualized, in order to enable the aggregator to organize the plan for trading the flexibility. The concept of this suite will be based on the work done in derivable D3.1.



- **Graph-based Analytics:** This suite corresponds to the development of a Graph Analytics platform, which will enable the aggregators to segment the DR strategies to be applied to appropriate DERs registered in their portfolio. Visual clustering and multi-objective analysis techniques have been developed, in order to achieve the efficient correlation between different features, such as incentives to be sent, energy prices KPI factors etc. This suite can be efficiently combined with the DSS & DR Strategies Optimization, in order to improve the framework for the decision making process. The visualization frameworks and the adopted techniques for this suites are reported in the deliverable D4.3 and D4.4. In brief, two operational modes are proposed:
 - **DR automated mode (D4.4):** This framework refers to analysis that should be performed based on predefined criteria. In brief, it can propose matching of VPPs, Microgrids or other type of DERs with specific type of DR program (such as a VPP can participate in a Critical Peak

Pricing program etc.) considering also the operational constraints of DERs and the comfort zones of prosumers.

- **DR hypothesis evaluation mode (D4.3):** This mode provides a set of feasible solutions that can be selected by the aggregators. For each of the solutions DR scheduling strategy is proposed. This is very useful, because it provides the possibility to select a solution depending on the factor you want to maximize or minimize respectively.



At this point, it should be mentioned that the architectural view presented in this chapter provides an overview of the data flow between the architectural components. In addition, in chapter 7 there are detailed specification templates that include the identified data models. This information provides insights for the development of the information view concerning the eDREAM architectural aspects. A basic standard that will define the eDREAM data model is the OpenADR. This standard is the most common one that includes data model to express DR signals and data exchanges.

The dynamic view analysis of the system provides insights and defines how the system actually works within runtime environment and how it performs in response to external (or internal) signal. The interactions between the system's actors and system's components are usually data flows representing the information exchanged in parallel or sequential execution of internal tasks.

The eDREAM use cases were firstly defined and analyzed in deliverable "D2.2 Use Case Analysis and application scenarios description V1". In the context of the WP2 activities, technical teleconferences on use cases/functional analysis were carried out in the scope of identifying all the dependencies between the key architectural components and the data exchanged during the system's functions or procedures. The logic of these complex operations are presented through **Sequence Diagrams** defining the functionalities of each of the key architectural components and the execution flows within each use case. A version revision of the eDREAM use cases has been presented in the deliverable D2.4, where the sequence diagrams have been introduced. In this version, further updates on use cases description and sequence diagrams have been performed due to inconsistencies during the activities of the technical WPs 3, 4 and 5 and the interconnection plan of WP6.

The defined Use Case involves DSOs, aggregators and prosumers. Its main objective is to establish a mechanism for aggregating flexibility and detecting in near real time the amount of flexibility actually provided by each prosumer.

The aggregation of the flexibility potential provided by multiple prosumers and the management of the individual deviations will avoid grid level congestion points, solving potential grid issues. To do so, prosumers are enrolled with aggregators, who knows their flexibility availability through current and forecasted data from power production and load demands.

When the DSO identifies potential issues on the grid (e.g. congestion and reverse flow), a flexibility request is sent to the aggregator through the marketplace. Based on the received flexibility request, the aggregator inquires its enrolled prosumers to identify the subset which may deliver the expected flexibility, creating an offer on the marketplace. The consolidated flexibility request curves are being injected into the prosumers self-enforcing smart contracts by the aggregator, then the deviation among the prosumer actual energy consumption and the expected profile is measured. In case of significant deviations, other prosumers (from the enrolled ones) will be identified, to provide the missing amount of flexibility. The deviating prosumers (if any) will be penalized while the prosumers operating as expected will be rewarded through incentives.

Generic Description	
Use Case Name	HL-UC01_LL-UC01: Prosumers enrolment in demand response programs

<i>Version</i>	V0.5
<i>Authors</i>	E@W, TUC
<i>Last Update</i>	1st Version in D2.2 2nd Version in D2.4 Updated within June 2019
<i>Brief Description</i>	<p>Aggregator negotiate with his/her customers (prosumers) showing the benefits through interactive multi-purpose visualization tool for user interaction.</p> <p>A first step towards the aggregation of flexibility on the Aggregator's side is to inform properly his/her prosumers about the benefits of enrolment in demand response programs. Through interactive visualization interface, potential incentives are sent to prosumers. These incentives are formulated mainly taking into account the "Customer Baseline Load (CBL)" through the component of Baseline Flexibility Estimation and forecasted data about consumption and production.</p>
<i>Assumptions and Pre-Conditions</i>	<p>A web interface is available for the Aggregator.</p> <p>The interconnection of the necessary components has been established, thus the decision support system can have access to prosumer's baseline load data.</p>
<i>Goal (Successful End Condition)</i>	Establish a mechanism for enrolling prosumers and make available their flexibility.
<i>Post-Conditions</i>	Prosumers are enrolled.
<i>Involved Actors</i>	Aggregator, Prosumers
<i>Use Case Initiation</i>	The aggregator needs to enrol prosumers for flexibility availability.
<i>Main Flow</i>	<p>Begin</p> <ol style="list-style-type: none"> 1. The aggregator sends request via the interactive visualization framework, in order to obtain the optimal DR incentives that should be sent to prosumers. 2. The DSS (Decision Support System) & DR Strategies Optimization UI receives the request and processes the settings/preferences. 3. The DSS (Decision Support System) & DR Strategies Optimization UI component requests the optimal calculation of DR incentives from the component VPP and DR Services Optimization Engine.

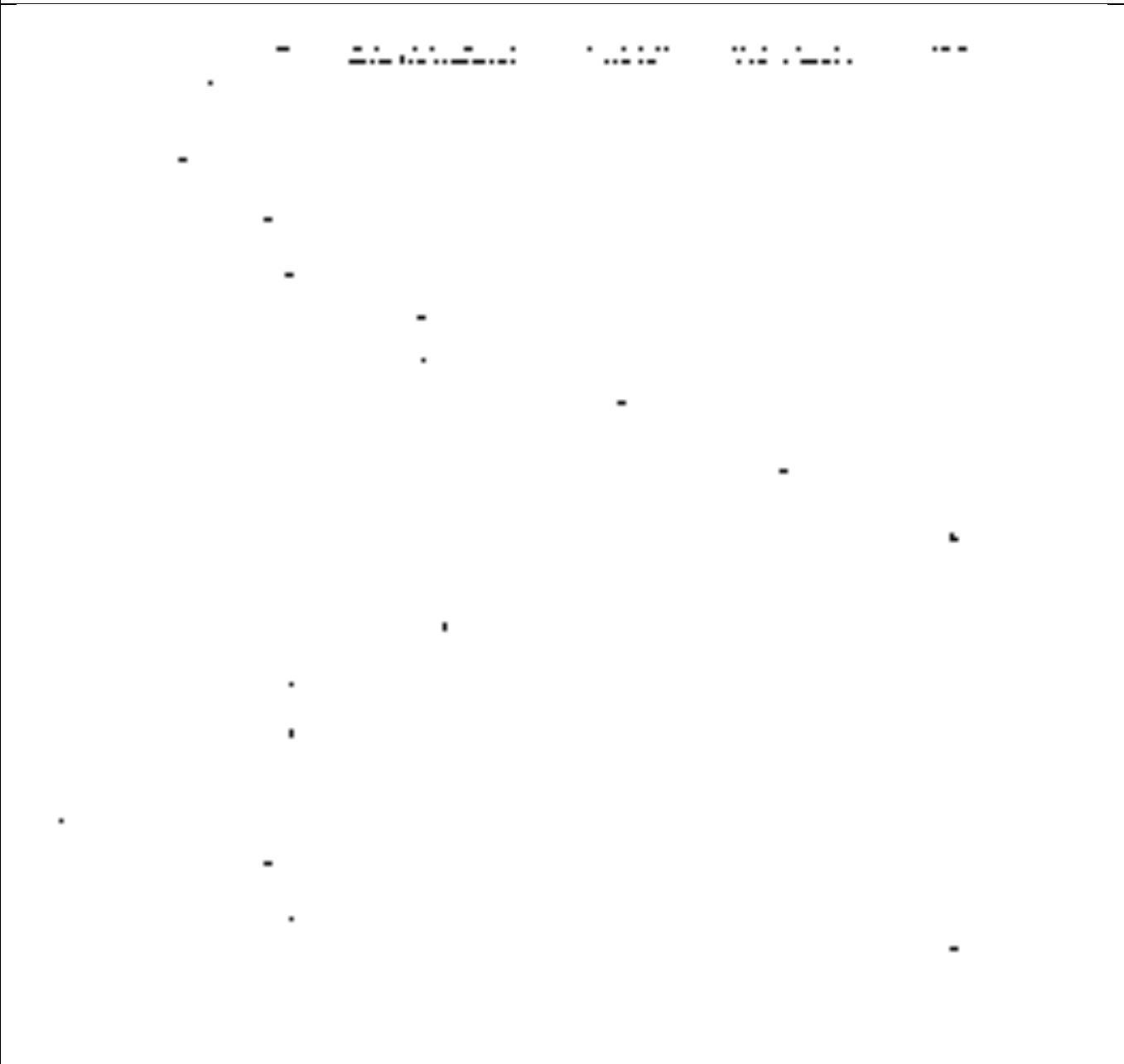
	<ol style="list-style-type: none"> 4. The VPP and DR Services Optimization Engine requests the baseline load flexibility for each prosumer. 5. The Baseline Flexibility Estimation returns the requested data. 6. The VPP and DR Services Optimization Engine requests consumption/production forecasted data for each prosumer. 7. The Electricity Consumption/Production Forecasting returns the requested data. 8. The VPP and DR Services Optimization Engine processes the received data and calculates the incentives. 9. The VPP and DR Services Optimization Engine sends the calculated data to DSS (Decision Support System) & DR Strategies Optimization UI for views preparation. 10. The DSS (Decision Support System) & DR Strategies Optimization UI processes the received data and prepare views. 11. The aggregator, through the web interface of DSS (Decision Support System) & DR Strategies Optimization, sends the incentives to prosumer.
<i>Alternative Courses</i>	-
<i>Relationships with other Use Cases</i>	HL-UC01_LL-UC02, HL-UC01_LL-UC03
<i>Architectural Elements / Services Involved</i>	DSS (Decision Support System) & DR Strategies Optimization UI (Graph-based Analytics considered); VPP and DR Service Optimization Engine; Baseline Flexibility Estimation; Electricity Consumption/Production Forecasting;
Specific Description	
<i>Relevance to eDREAM WPs</i>	WP3 & WP4
<i>Main Tasks Involved</i>	T3.1, T3.2, T4.1, T4.3 & T4.4
<i>Main Technical Partners Involved</i>	CERTH, TU, ATOS, TUC, E@W & EMOT
<i>Notes (Optional)</i>	-
UML Sequence Diagram	



Generic Description	
Use Case Name	HL-UC01_LL-UC02: Contract Setting
Version	V0.5
Authors	E@W, TUC
Last Update	1st Version in D2.2 2nd Version in D2.4 Updated within June 2019
Brief Description	Aggregator and prosumers agree on baseline and incentives for activation of flexibility through the initialization of self-enforcing smart contract in

	which the prosumers provide their energy flexibility availability interval. The self-enforcing smart contract is defined as a distributed mean of transaction and specifies the contracted baseline energy consumption or production levels (curves).
<i>Assumptions and Pre-Conditions</i>	<p>A web interface should be available for the prosumer.</p> <p>The smart metering energy devices can properly communicate with the eDREAM platform</p> <p>The smart contract can be stored in a secured way on the prosumer's side.</p>
<i>Goal (Successful End Condition)</i>	The aim of this UC is the setting of transactions' conditions between the aggregator and prosumers.
<i>Post-Conditions</i>	Prosumers activate their flexibility availability interval in case of request.
<i>Involved Actors</i>	Aggregator, Prosumers
<i>Use Case Initiation</i>	Aggregator and prosumers agree on baseline and incentives for flexibility activation.
<i>Main Flow</i>	<p>Begin</p> <ol style="list-style-type: none"> 1. The prosumer agrees on baseline and incentives and sends signal to aggregator through web interface. 2. The aggregator sends signal for initialization of smart contract through the HMI. 3. The HMI receives the aggregator's request and processes the input data. 4. The HMI sends the request to the component Blockchain-driven control for LV networks (flexibility management), in order to perform calculations for smart contract modeling. 5. The Blockchain-driven control for LV networks requests prosumer's estimated baseline load flexibility from the component Baseline Flexibility Estimation. 6. The Baseline Flexibility Estimation sends the requested data. 7. The Blockchain-driven control for LV networks requests prosumer's consumption/production forecasted data from the component Electricity Consumption/Production Forecasting.

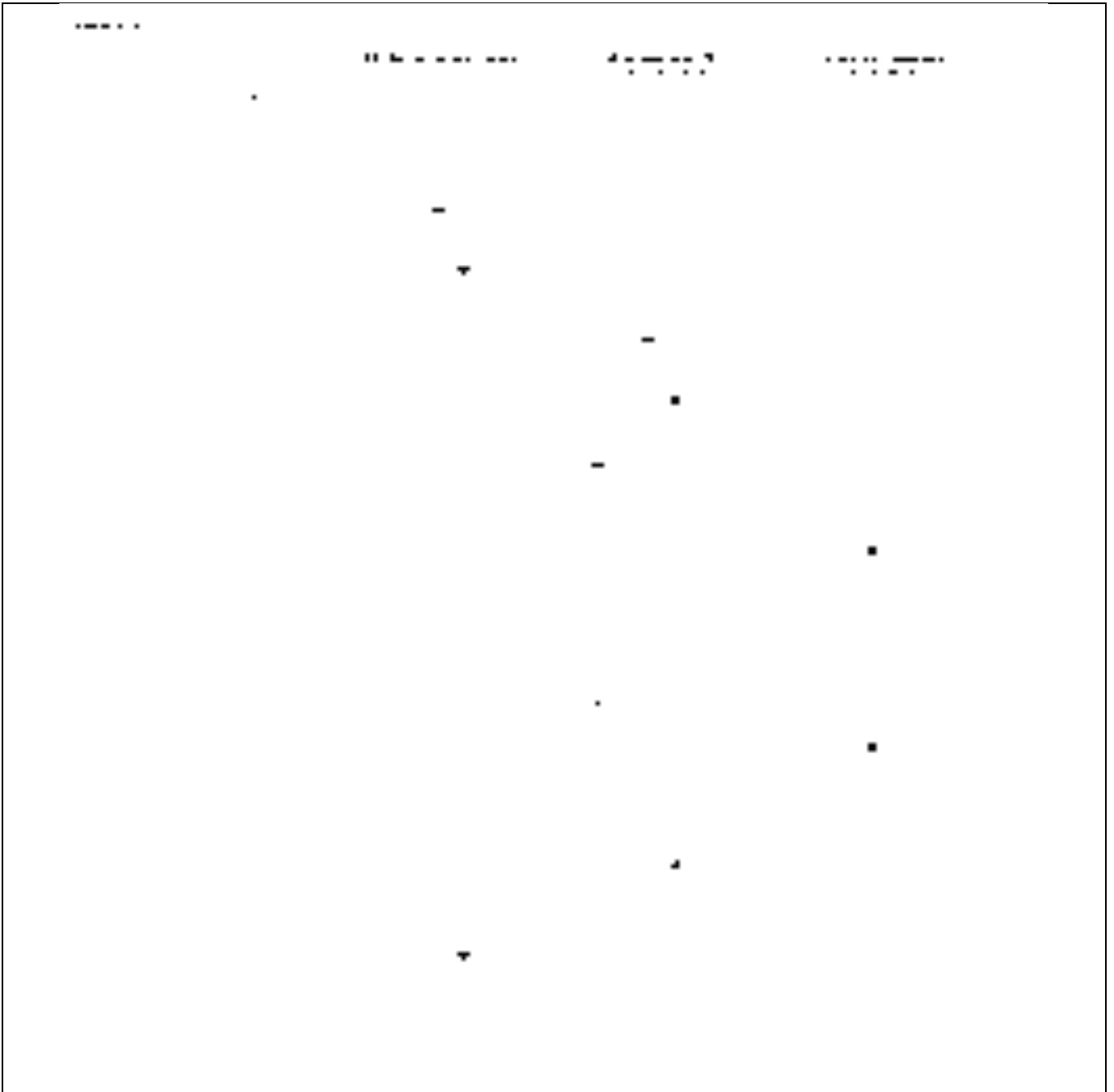
	<p>8. The Electricity Consumption/Production Forecasting returns the requested data.</p> <p>9. The Blockchain-driven control for LV networks requests prosumer's personal information from the Distributed Ledger.</p> <p>10. The Distributed Ledger returns the requested information.</p> <p>11. The component Blockchain-driven control for LV networks sets the conditions for smart contract modeling.</p> <p>12. The Blockchain-driven control sends the modeled smart contract to aggregator through the HMI.</p> <p>13. The HMI processes the received data and prepare views for the aggregator.</p> <p>14. The aggregator sends the smart contract's conditions along with the remuneration to prosumer for confirmation through web interface of the HMI.</p> <p>15. The prosumer sends the confirmation/validation of smart contract to aggregator's HMI through web interface.</p> <p>16. The HMI receives and processes the input data.</p> <p>17. The HMI sends smart contract for secure storage on the prosumer side.</p>
<i>Alternative Courses</i>	Some prosumers do not agree on baseline and incentives for activation of flexibility (initialization of Smart Contract).
<i>Relationships with other Use Cases</i>	HL-UC01_LL-UC01
<i>Architectural Elements / Services Involved</i>	HMI; Blockchain-driven control for LV networks; Baseline Flexibility Estimation; Electricity consumption/production forecasting; Distributed Ledger;
<i>Specific Description</i>	
<i>Relevance to eDREAM WPs</i>	WP3, WP4 & WP5
<i>Main Tasks Involved</i>	T3.1, T3.2, T4.3, T5.1 & T5.2
<i>Main Technical Partners Involved</i>	TUC, ENG, E@W, CERTH, TU, ATOS

Notes (Optional)	-
UML Sequence Diagram	
	

Generic Description	
Use Case Name	HL-UC01_LL-UC03: Potential energy flexibility evaluation

<i>Version</i>	V0.5
<i>Authors</i>	ENG, TUC, E@W
<i>Last Update</i>	1st Version in D2.2 2nd Version in D2.4 Updated within June 2019
<i>Brief Description</i>	Aggregator evaluates the potential energy flexibility guaranteed by prosumers using drones for aerial surveying with optical and thermal imaging and laser scanning to assess the application of Demand Response in a specific zone.
<i>Assumptions and Pre-Conditions</i>	A web interface should be available for the aggregator. The drone should be properly equipped with optical, thermal and laser cameras. The DR Aerial Survey Toolkit interface should communicate in an appropriate way with the eDREAM platform.
<i>Goal (Successful End Condition)</i>	The aim of this UC is to obtain an energy consumption and production baseline flexibility estimation of potential prosumers or already registered prosumers after a long period.
<i>Post-Conditions</i>	The aggregator identifies new qualified prosumers. Concerning the registered prosumers, the aggregator receives a quick estimation of their flexibility after a long period of time.
<i>Involved Actors</i>	Aggregator, Prosumers
<i>Use Case Initiation</i>	The aggregator sends request for flexibility evaluation.
<i>Main Flow</i>	<p>Begin</p> <ol style="list-style-type: none"> 1. The aggregator requests flexibility evaluation of potential prosumers or flexibility verification of registered prosumers (after a long period) through the UI of the DR Aerial Survey Toolkit. 2. The DR Aerial Survey Toolkit UI receives the request and processes the input data. 3. The DR Aerial Survey UI requests flexibility evaluation over a specified area of building assets from the component Multi-building DR characterization.

	<ol style="list-style-type: none"> 4. The component Multi-building DR characterization collects optical, thermal and/or LIDAR images over the specified area and analyzes them. 5. The Multi-building DR characterization sends the analyzed images to Baseline Flexibility Estimation and requests baseline flexibility estimation for each potential new prosumer. 6. The Baseline Flexibility Estimation returns the requested data. 7. In case of registered prosumer, the Multi-building DR characterization requests the prosumer's baseline load flexibility. 8. The Baseline Flexibility Estimation returns the requested data. 9. The Multi-building DR characterization processes the received data and produces data for flexibility evaluation. 10. The Multi-building DR characterization sends the calculated data to DR Aerial Survey Toolkit. 11. The DR Aerial Survey UI processes the received data, prepare view for assessment of DR application potential and present them to aggregator.
<i>Alternative Courses</i>	<p>Other methods are used for the identification of new prosumers.</p> <p>The flexibility evaluation of registered prosumers is only performed through metering devices and suitable calculations.</p>
<i>Relationships with other Use Cases</i>	HL-UC01_LL-UC01
<i>Architectural Elements / Services Involved</i>	<p>DR Aerial Survey Toolkit UI;</p> <p>Multi-building DR characterization through thermal, optical and LIDAR information fusion;</p> <p>Baseline Flexibility Estimation;</p>
Specific Description	
<i>Relevance to eDREAM WPs</i>	WP3
<i>Main Tasks Involved</i>	T3.2 & T3.4
<i>Main Technical Partners Involved</i>	TU, CERTH, KIWI, E@W, TUC
<i>Notes (Optional)</i>	-
UML Sequence Diagram	



Generic Description	
Use Case Name	HL-UC01_LL-UC04: Energy demand/production forecasting for day-ahead trading of flexibility
Version	V0.5

<i>Authors</i>	ENG, TUC, E@W
<i>Last Update</i>	1st Version in D2.2 2nd Version in D2.4 Updated within June 2019
<i>Brief Description</i>	Aggregator receives from each prosumer enrolled the individual energy demand/production values for the next day. Then he/she creates and sends a forecast of the aggregated energy demand of all individual customers to the DSO, who uses it to forecast future congestion points. Forecasting tools have to provide the next-days prediction at different granularity (prosumer, aggregator, DSO). The HMI has to show these values with frequency rate around 10 minutes (5 min or one hour are also possible sample rates).
<i>Assumptions and Pre-Conditions</i>	Prosumers are enrolled with the aggregator. Prosumer's historical data about consumption/production are available.
<i>Goal (Successful End Condition)</i>	The aim of this UC is to obtain forecasted values for production/consumption at individual prosumer level.
<i>Post-Conditions</i>	The Aggregator can inform the DSO about the total forecasted energy demand of all the registered prosumers.
<i>Involved Actors</i>	Aggregator, Prosumers, DSO
<i>Use Case Initiation</i>	The UC is automatically initiated as Forecasting Tool UI have to present consumption/production forecasting with frequency rate around 10 minutes.
<i>Main Flow</i>	<p>Begin</p> <ol style="list-style-type: none"> 1. The Forecasting Tool UI requests for each prosumer enrolled consumption/production forecasted data for day ahead with a frequency rate around 10 minutes from the component Electricity Consumption/Production Forecasting. 2. The component Electricity Consumption/Production Forecasting requests prosumer's baseline load flexibility from the component Baseline Flexibility Estimation. 3. The Baseline Flexibility Estimation returns the requested data. 4. The Electricity Consumption/Production Forecasting requests prosumer's historical data for consumption/production from the Database. 5. The Database returns the requested data.

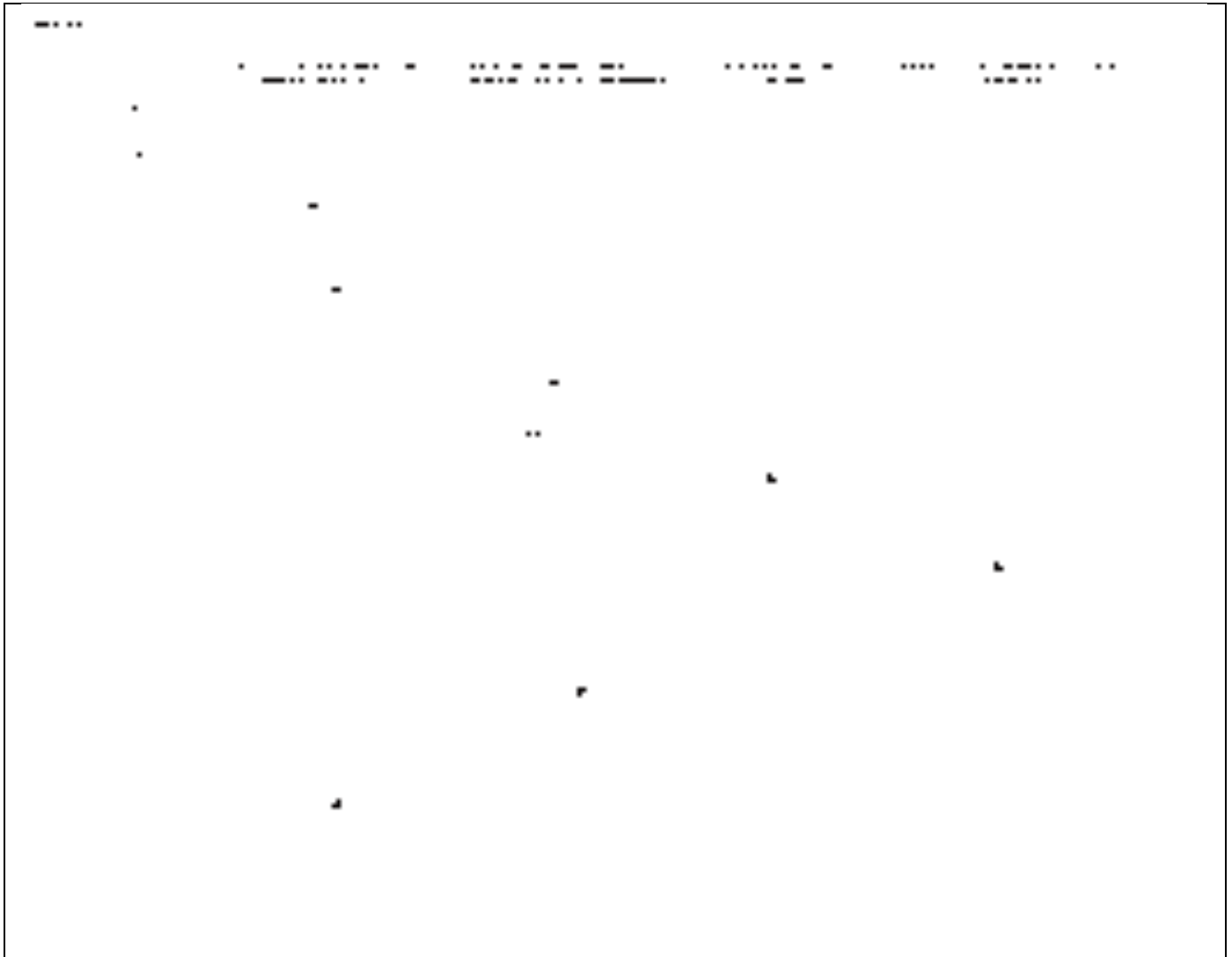
	<p>6. The Electricity Consumption/Production Forecasting calculates consumption/production forecasted data for each prosumer.</p> <p>7. The Electricity Consumption/Production Forecasting sends production forecasted data for PV assets to PV Degradation and Trend Analysis.</p> <p>8. The PV Degradation and Trend Analysis sends the estimated degradation rate for PV assets.</p> <p>9. The Electricity Consumption/Production Forecasting calculates the aggregated energy demand of the enrolled individual prosumers.</p> <p>10. The Electricity Consumption/Production Forecasting sends the calculated data with frequency rate around 10 minutes, in order to inform the aggregator.</p>
<i>Alternative Courses</i>	-
<i>Relationships with other Use Cases</i>	HL-UC01_LL-UC05
<i>Architectural Elements / Services Involved</i>	<p>Forecasting Tool UI;</p> <p>Electricity Consumption/Production Forecasting;</p> <p>Baseline Flexibility Estimation;</p> <p>Database (Cassandra DB);</p> <p>PV/RES Degradation and Trend Analysis;</p>
Specific Description	
<i>Relevance to eDREAM WPs</i>	WP3 & WP4
<i>Main Tasks Involved</i>	T3.1, T3.2 & T4.1
<i>Main Technical Partners Involved</i>	TUC, TU, E@W, CERTH, ENG
<i>Notes (Optional)</i>	-
UML Sequence Diagram	



Generic Description	
Use Case Name	HL-UC01_LL-UC05: Flexibility request
Version	V0.5
Authors	E@W, TUC
Last Update	1st Version in D2.2 2nd Version in D2.4 Updated within June 2019
Brief Description	DSO creates a forecast of the total load on the critical branches of the network (i.e. parts of the grid for which a congestion is expected) and in case congestion is forecasted, he/she sends a flex request to a Flexibility Marketplace (only between DSO and Aggregators) with associated incentives

	(intraday: step is repeated). Then, the Aggregator requests the flexibility availability of the registered prosumers.
<i>Assumptions and Pre-Conditions</i>	<p>Congestion point/s has/have been forecasted by the DSO.</p> <p>The Aggregator is properly informed by the DSO.</p> <p>The Aggregator's UI is connected with eDREAM core platform.</p>
<i>Goal (Successful End Condition)</i>	Through this UC, the Aggregator aims to define the actual aggregated flexibility that can deliver to DSO.
<i>Post-Conditions</i>	Aggregator's total flexibility offer is accepted by the DSO.
<i>Involved Actors</i>	Aggregator, DSO
<i>Use Case Initiation</i>	The UC is initiated by the Aggregator in case of previous request by the DSO (forecasted congestion point/s).
<i>Main Flow</i>	<p>Begin</p> <ol style="list-style-type: none"> 1. The aggregator sends flexibility request through the UI of the component Decision Support System & DR Strategies Optimization in case that a previous request have been sent by the DSO. 2. The Decision Support System & DR Strategies Optimization receives and processes the input data. 3. The Decision Support System & DR Strategies Optimization sends request for evaluation of prosumers' flexibility availability to the component Blockchain-driven control for LV networks (flexibility management). 4. The Blockchain-driven control for LV networks requests baseline load flexibility for each registered prosumer. 5. The Baseline Flexibility Estimation returns the requested data. 6. The Blockchain-driven control for LV networks requests prosumer's consumption/production forecasted data from the component Electricity consumption/production forecasting. 7. The Electricity consumption/production forecasting returns the forecasted data. 8. The Blockchain-driven control for LV networks calculates the actual aggregated flexibility that can be delivered.

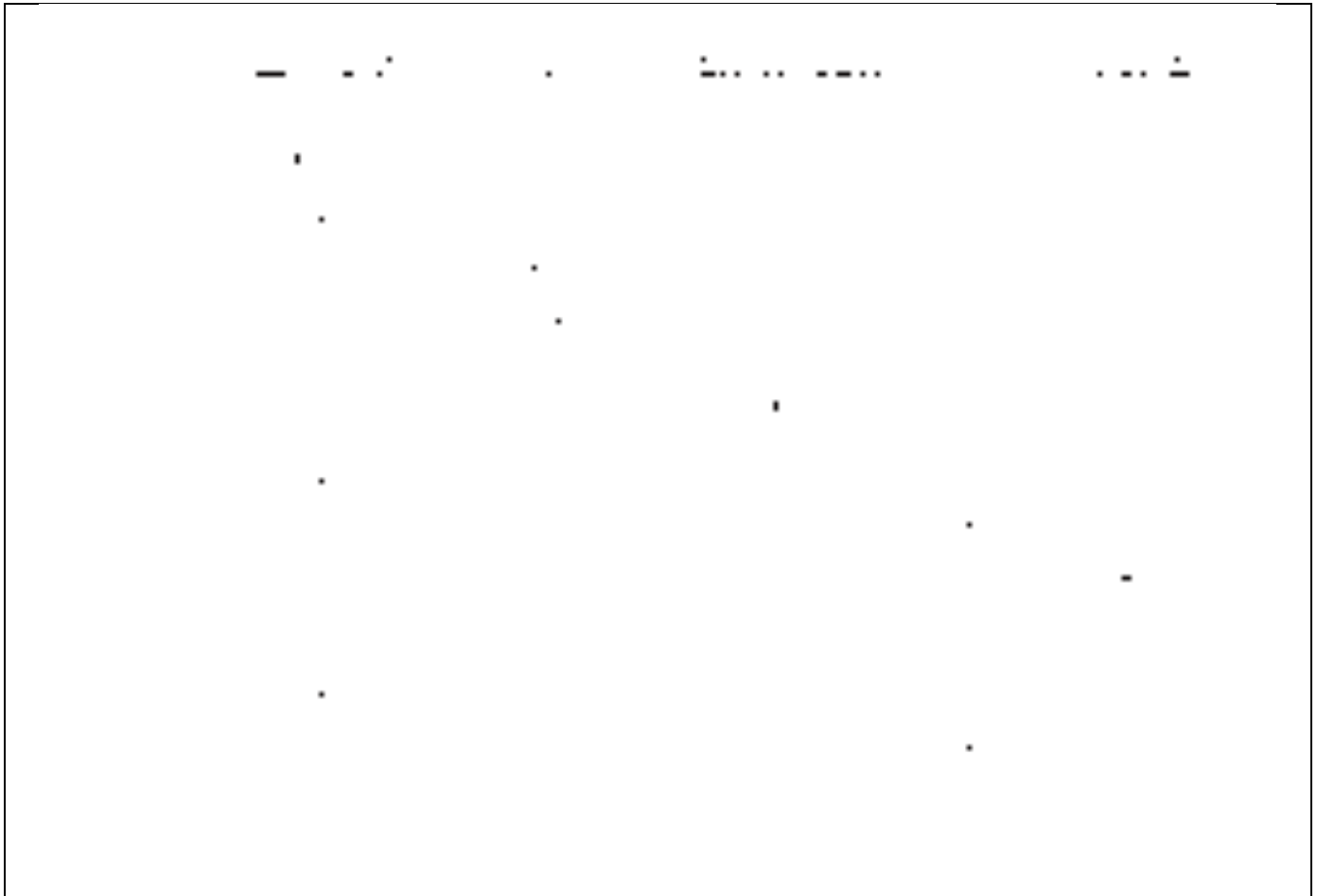
	<p>9. The Blockchain-driven control for LV networks sends the aggregated flexibility availability to the Decision Support System & DR Strategies Optimization.</p> <p>10. The Decision Support System & DR Strategies Optimization processes data and prepares views of total flexibility availability to aggregator.</p>
<i>Alternative Courses</i>	DSO applies traditional methods of peak loads reduction.
<i>Relationships with other Use Cases</i>	HL-UC01_LL-UC04, HL-UC01_LL-UC06
<i>Architectural Elements / Services Involved</i>	Decision Support System & DR Strategies Optimization UI; Blockchain-driven control for LV networks (flexibility management); Baseline Flexibility Estimation; Electricity Consumption/Production Forecasting;
<i>Specific Description</i>	
<i>Relevance to eDREAM WPs</i>	WP3, WP4 & WP5
<i>Main Tasks Involved</i>	T3.1, T3.2, T4.1, T4.3 & T5.2
<i>Main Technical Partners Involved</i>	TUC, TU, E@W, CERTH, ENG, ATOS
<i>Notes (Optional)</i>	-
<i>UML Sequence Diagram</i>	



Generic Description	
Use Case Name	HL-UC01_LL-UC06: Flexibility offering
Version	V0.5
Authors	E@W, TUC
Last Update	1st Version in D2.2 2nd Version in D2.4 Updated within June 2019

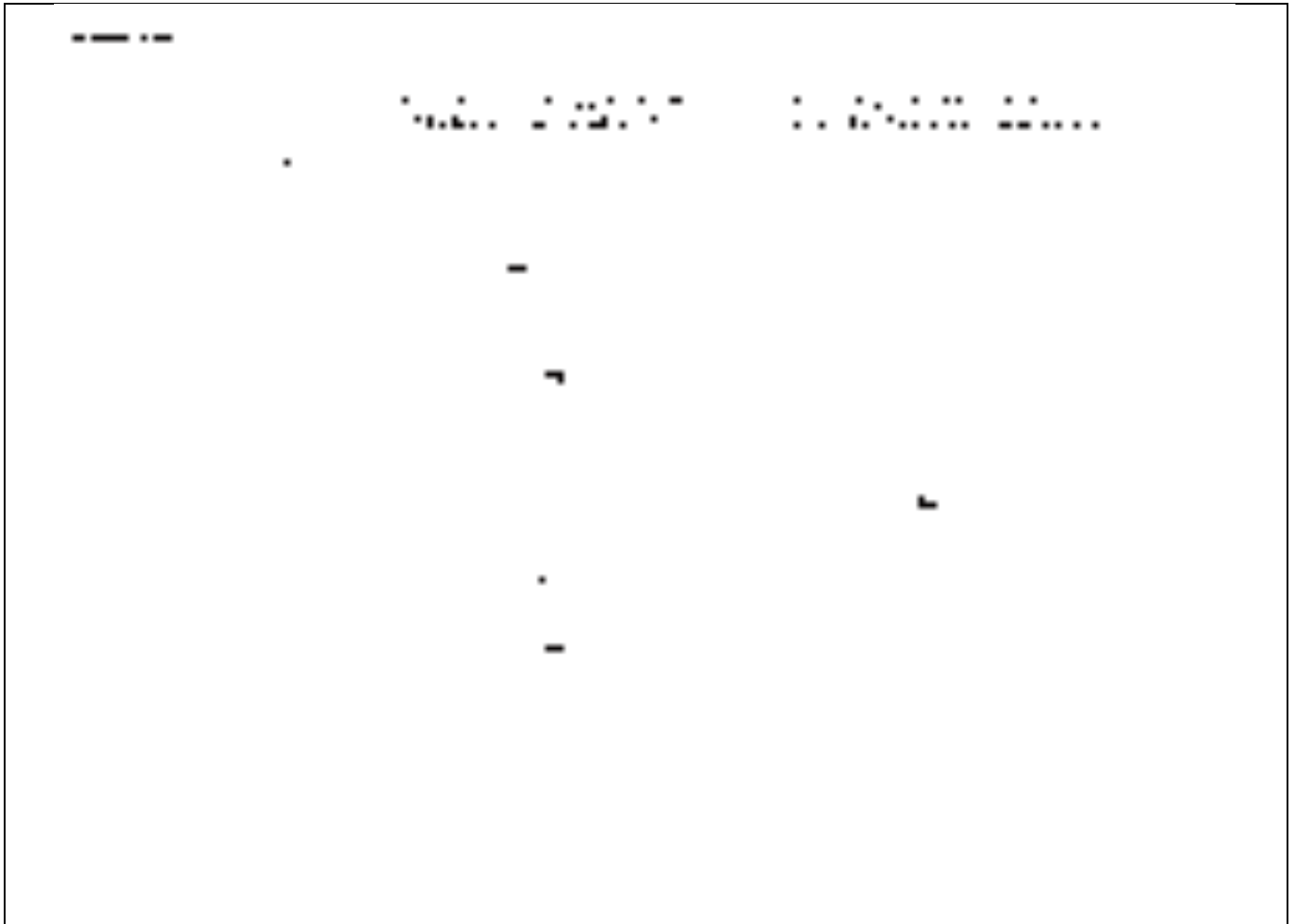
<i>Brief Description</i>	Based on the received flexibility request, the Aggregator inquires its enrolled prosumers to identify the subset which may deliver the expected flexibility.
<i>Assumptions and Pre-Conditions</i>	The Aggregator had received flexibility request by the DSO. The Smart Contracts' conditions are available and accessible.
<i>Goal (Successful End Condition)</i>	Through this UC, the Aggregator aims to define the subset of enrolled prosumers that may deliver the expected flexibility.
<i>Post-Conditions</i>	The Aggregator is ready for delivering flexibility in case the DSO sends the flexibility order.
<i>Involved Actors</i>	Aggregator, Prosumers
<i>Use Case Initiation</i>	The UC is initiated by the Aggregator in order to identify the suitable subset of prosumers for expected flexibility delivery.
<i>Main Flow</i>	<p>Begin</p> <ol style="list-style-type: none"> 1. The aggregator sends request though the UI of Decision Support System & DR Strategies Optimization for identification of the suitable subset of prosumers. 2. The Decision Support System & DR Strategies Optimization UI receives the request and processes the input data. 3. The Decision Support System & DR Strategies Optimization UI requests the subset of prosumers from the component Big Data Clustering at multiple scales. 4. The Big Data Clustering at multiple scales receives the request and indicates the identified subset. 5. The Big Data Clustering at multiple scale sends the identified subset to the Decision Support System & DR Strategies Optimization UI. 6. The Decision Support System & DR Strategies Optimization UI requests smart contracts evaluation of the prosumers belonging to the identified subsets. 7. The Blockchain-driven control for LV networks returns agreed flexibility that can be delivered. 8. The Decision Support System & DR Strategies Optimization UI processes the received data.

	<p>9. The Decision Support System & DR Strategies Optimization UI sends request for validation of transaction to the Distributed Ledger.</p> <p>10. The Distributed Ledger sends the received request to the Closed Loop DR Verification Engine.</p> <p>11. The Closed loop DR Verification Engine returns the requested data to the Distributed Ledger.</p> <p>12. The Distributed Ledger sends the received data to the Decision Support System & DR Strategies Optimization.</p> <p>13. The Decision Support System & DR Strategies Optimization UI processes received data and prepares views for capability of flexibility activation to the aggregator.</p> <p>14. The Decision Support System & DR Strategies Optimization UI sends the validated data for secure storage to the component Distributed Ledger.</p>
<i>Alternative Courses</i>	The identified prosumers do not agree to deliver their flexibility.
<i>Relationships with other Use Cases</i>	HL-UC01_LL-UC05, HL-UC01_LL-UC07
<i>Architectural Elements / Services Involved</i>	Decision Support System & DR Strategies Optimization UI; Big Data Clustering at multiple scales; Blockchain-driven control for LV networks (flexibility management); Closed loop DR Verification Engine; Distributed Ledger;
Specific Description	
<i>Relevance to eDREAM WPs</i>	WP4 & WP5
<i>Main Tasks Involved</i>	T4.1, T4.2, T5.1, T5.2 & T5.3
<i>Main Technical Partners Involved</i>	TU, ENG, E@W, TUC, EMOT
<i>Notes (Optional)</i>	-
UML Sequence Diagram	



Generic Description	
Use Case Name	HL-UC01_LL-UC07: Flexibility acceptance
Version	V0.5
Authors	E@W, TUC, ENG
Last Update	1st Version in D2.2 2nd Version in D2.4 Updated within June 2019
Brief Description	DSO accepts one or multiple flexibility offers and, if so, the DSO sends a flexibility order.
Assumptions and Pre-Conditions	The DSO accepted flexibility offers. Specific flexibility offer/s are selected by the DSO.

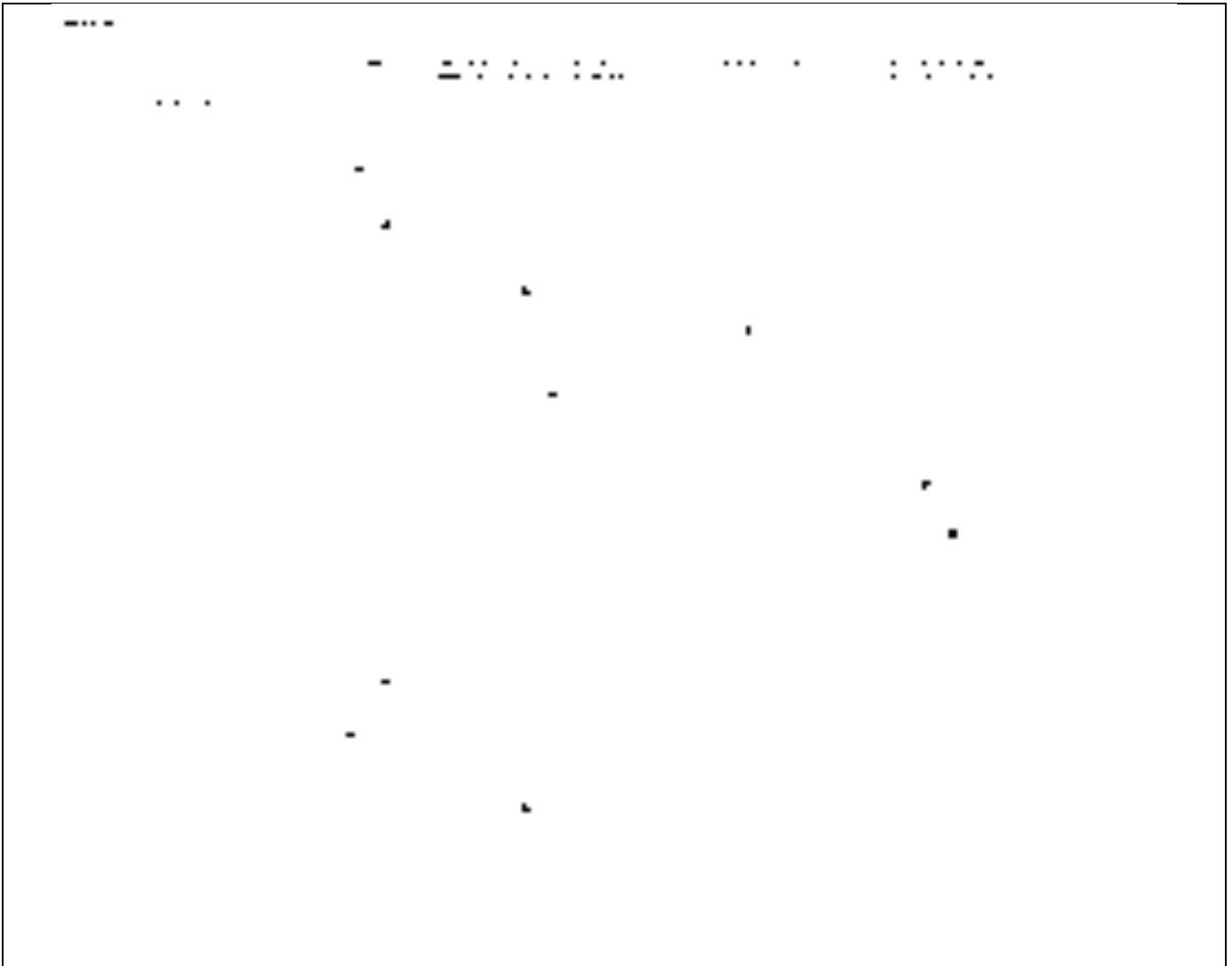
<i>Goal (Successful End Condition)</i>	The DSO aims to inform properly the Aggregator that his/her flexibility offer/s are accepted.
<i>Post-Conditions</i>	The Aggregator is ready for initiating the flexibility provisioning.
<i>Involved Actors</i>	DSO, Aggregator
<i>Use Case Initiation</i>	The Aggregator's flexibility offers are successfully accepted by the DSO.
<i>Main Flow</i>	<p>Begin</p> <ol style="list-style-type: none"> 1. The DSO sends flexibility order to aggregator through the web interface of the component Decision Support System & DR Strategies Optimization. 2. The Decision Support System & DR Strategies Optimization UI receives the request and processes the preferences. 3. The Decision Support System & DR Strategies Optimization UI requests from the component Blockchain-driven control for LV networks the smart contracts' conditions concerning demand profile and load shifting (that is going to be activated). 4. The Blockchain-driven control for LV networks returns the requested data. 5. The Decision Support System & DR Strategies Optimization UI processes data and prepares views, in order to enable DSO to check the provided conditions.
<i>Alternative Courses</i>	The Aggregator's flexibility offer/s are rejected by the DSO.
<i>Relationships with other Use Cases</i>	HL-UC01_LL-UC06, HL-UC01_LL-UC08
<i>Architectural Elements / Services Involved</i>	Decision Support System & DR Strategies Optimization UI; Blockchain-driven control for LV networks (flexibility management);
Specific Description	
<i>Relevance to eDREAM WPs</i>	WP4 & WP5
<i>Main Tasks Involved</i>	T4.1, T4.3, T4.4 & T5.2
<i>Main Technical Partners Involved</i>	TUC, TU, CERTH, ENG, ATOS, E@W, EMOT
<i>Notes (Optional)</i>	-
UML Sequence Diagram	



Generic Description	
Use Case Name	HL-UC01_LL-UC08: Flexibility provisioning
Version	V0.5
Authors	E@W, TUC
Last Update	1st Version in D2.2 2nd Version in D2.4 Updated within June 2019
Brief Description	Aggregator sends the flexibility orders to his/her prosumers (injection of demand curve via a smart contract), in order to adjust the load of his/her clients and fulfil the flexibility need. The prosumers, that followed the provided curve by shifting their load, will receive payment from the

	aggregator for the flexibility provision based on their flexibility contract (settlement).
<i>Assumptions and Pre-Conditions</i>	The Aggregator received a flexibility order by the DSO.
<i>Goal (Successful End Condition)</i>	The aim of this UC is to deliver the appropriate agreed flexibility to the DSO.
<i>Post-Conditions</i>	The Aggregator provides the appropriate agreed flexibility to the DSO leveraging on its prosumers.
<i>Involved Actors</i>	Aggregator, Prosumers
<i>Use Case Initiation</i>	The Aggregator receives a flexibility offer by the DSO.
<i>Main Flow</i>	<p>Begin</p> <ol style="list-style-type: none"> 1. The Aggregator sends flexibility orders (demand profile and load shifting) to his/her prosumers through the HMI. 2. The HMI receives the request and processes the input preferences. 3. The HMI requests injection of demand curve via the smart contract from the component Blockchain-driven control for LV networks (flexibility management). 4. The Blockchain-driven control for LV networks requests the prosumers' registered energy assets from the component Distributed Ledger. 5. The Distributed Ledger sends the requested data. 6. The Blockchain-driven control for LV networks evaluates the actual flexibility of consumption/production that is delivered. 7. The Blockchain-driven control for LV networks requests the actual flexibility that is delivered from the component Closed loop DR Verification Engine. 8. The Closed loop DR Verification Engine validates if the provided curve corresponds to the margins mentioned in the previous financial settlement. 9. The Closed loop DR Verification Engine returns the validated data to the Distributed Ledger for secure storage.

	<p>10. The Closed loop DR Verification Engine sends the actual flexibility that can be delivered to the HMI.</p> <p>11. The HMI processes input data and prepare views.</p> <p>12. If a deviation is measused, the Aggregator though the HMI penalizes the selected prosumers and sends request to the Blockchain-driven control for LV networks, in order to identify other prosumers to provide the missing levels of flexibility.</p>
<i>Alternative Cources</i>	The prosumers don't deliver the ordered amount of flexibility to the aggregator.
<i>Relationships with other Use Cases</i>	HL-UC01_LL-UC07
<i>Architectural Elements / Services Involved</i>	HMI; Blockchain-driven control for LV networks (flexibility management); Distributed Ledger; Closed loop DR Verification Engine;
<i>Specific Description</i>	
<i>Relevance to eDREAM WPs</i>	WP4 & WP5
<i>Main Tasks Involved</i>	T4.3, T5.1, T5.2 & T5.3
<i>Main Technical Partners Involved</i>	TUC, ENG, E@W, EMOT
<i>Notes (Optional)</i>	-
<i>UML Sequence Diagram</i>	



In this scenario, the eDREAM project aims to define a mechanism for decentralized energy trading (price-driven), enabling prosumers to buy or sell energy by means of peer-to-peer transactions either directly or through an energy aggregator if they are not big enough.

Prosumers are willing to buy or sell energy and for this, they are registered with the energy trading platform or with the energy aggregator.

Generic Description	
Use Case Name	HL-UC02_LL-UC01: Prosumers registration with the energy trading platform

<i>Version</i>	V0.3
<i>Authors</i>	TUC
<i>Last Update</i>	1st Version in D2.2 2nd Version in D2.4 Updated within June 2019
<i>Brief Description</i>	Prosumers register on the energy market providing their information that will be validated through self-enforcing smart contract. Prosumers must be able to buy energy tokens needed to transact energy, which will be deposited in their wallets and to customize the smart contract for the definition of their energy trading rules.
<i>Assumptions and Pre-Conditions</i>	Data about the prosumer identification and energy demand/production is available. The energy trading platform is operational.
<i>Goal (Successful End Condition)</i>	The aim of this UC is to establish a mechanism for enrolling prosumers with the energy trading market.
<i>Post-Conditions</i>	The prosumer is registered with the energy trading market.
<i>Involved Actors</i>	Prosumers
<i>Use Case Initiation</i>	Prosumer is willing to trade energy with the blockchain based energy market.
<i>Main Flow</i>	<p>Begin</p> <ol style="list-style-type: none"> 1. Prosumer uses the web interface of the energy market HMI to provide the information needed for registration in the energy marketplace. 2. The HMI sends the information inserted by the prosumer to the component Secured Blockchain-driven Energy Market. 3. The Secured Blockchain-driven Energy Market requests necessary information for setting a self-enforcing smart contract from the Distributed Ledger. 4. The Distributed Ledger returns the requested data. 5. The Secured Blockchain-driven Energy Market provides the conditions of the self-enforcing smart contract for the energy trading.

	<p>6. The Secured Blockchain-driven Energy Market sends the data to prosumer through the HMI, so as to be able to customize the smart contract conditions.</p> <p>7. The Secured Blockchain-driven Energy Market sends the smart contract conditions and request the energy token that should be bought by the prosumer from the Closed Loop DR Verification Engine.</p> <p>8. The Closed Loop DR Verification Engine sends the energy token needed to the HMI, in order to inform the prosumer.</p> <p>9. The HMI processes the received data and prepare views.</p>
<i>Alternative Courses</i>	Prosumers are not willing to buy or sell energy in a decentralized (blockchain based) market.
<i>Relationships with other Use Cases</i>	HL-UC02_LL-UC02
<i>Architectural Elements / Services Involved</i>	HMI; Secured Blockchain-driven energy market; Distributed Ledger; Closed loop DR Verification Engine;
<i>Specific Description</i>	
<i>Relevance to eDREAM WPs</i>	WP4 & WP5
<i>Main Tasks Involved</i>	T4.3, T5.1, T5.2 & T5.3
<i>Main Technical Partners Involved</i>	TUC, ENG, E@W, EMOT
<i>Notes (Optional)</i>	-
<i>UML Sequence Diagram</i>	



Generic Description	
Use Case Name	HL-UC02_LL-UC02: Prosumers bids/offer submission
Version	V0.3
Authors	TUC
Last Update	1st Version in D2.2 2nd Version in D2.4 Updated within June 2019
Brief Description	Prosumers submit bids/offers associating the number of tokens equivalent with their amount of energy (e.g. 1 token = 1 Kw). Prosumers use forecasted data to construct the bids/offers for the next market session.

<i>Assumptions and Pre-Conditions</i>	The prosumer is registered with the energy trading market.
<i>Goal (Successful End Condition)</i>	The aim of this UC is to establish a mechanism for prosumers to submit their bids/offers of energy.
<i>Post-Conditions</i>	The bids/offers are submitted.
<i>Involved Actors</i>	Prosumers
<i>Use Case Initiation</i>	The prosumer would like to submit energy bids/offers with the blockchain based energy market.
<i>Main Flow</i>	<p>Begin</p> <ol style="list-style-type: none"> 1. Prosumer requests forecasted data about his/her own production/consumption through the web interface of the HMI. 2. The HMI requests prosumer's consumption/production forecasted data from the component Electricity Consumption/Production Forecasting. 3. The Electricity Consumption/Production Forecasting returns the requested data. 4. The information is deployed in the HMI. 5. (Considering forecasted data) prosumer places bids/offers in the energy market through the component Secured Blockchain-driven Energy Market. 6. The Secured Blockchain-driven Energy Market sends the information of bids/offers (number of tokens) to the Distributed Ledger. 7. The Secured Blockchain-driven Energy Market requests the smart contract's conditions from the Distributed Ledger. 8. The Distributed Ledger returns the requested data. 9. The Secured Blockchain-driven Energy Market performs market session management for bids/offers validation. 10. The Secured Blockchain-driven Energy Market sends the validated data to the HMI.

	11. The HMI processes the received data and prepares views, in order to inform prosumer for the validation of bids/offers against the session rules.
<i>Alternative Courses</i>	-
<i>Relationships with other Use Cases</i>	HL-UC02_LL-UC01, HL-UC02_LL-UC03
<i>Architectural Elements / Services Involved</i>	HMI; Electricity Consumption/Production Forecasting; Secured Blockchain-driven energy market; Distributed Ledger;
<i>Specific Description</i>	
<i>Relevance to eDREAM WPs</i>	WP3, WP4 & WP5
<i>Main Tasks Involved</i>	T3.1, T4.1, T5.1 & T5.2
<i>Main Technical Partners Involved</i>	TUC, ENG, E@W, EMOT, CERTH, TU
<i>Notes (Optional)</i>	-
<i>UML Sequence Diagram</i>	



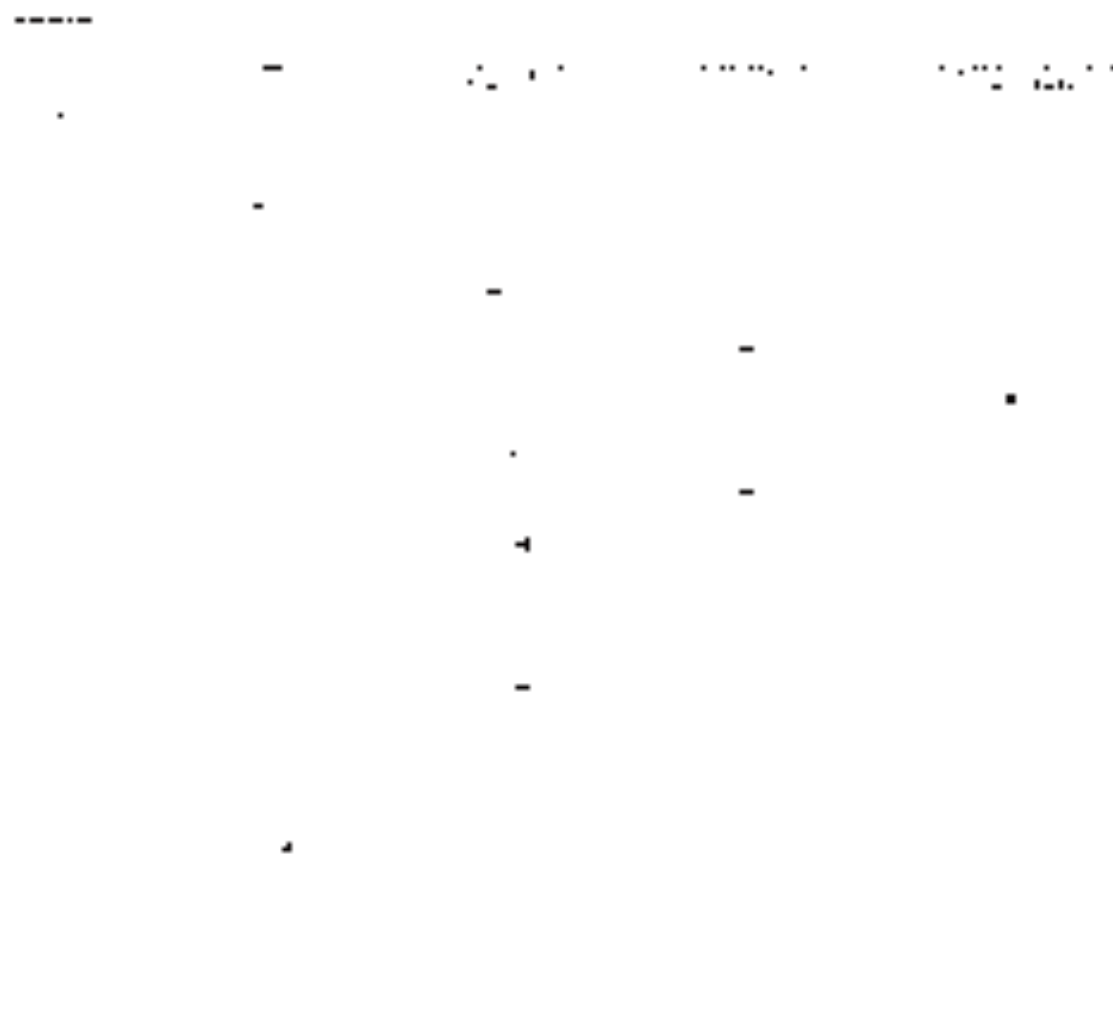
Generic Description	
Use Case Name	HL-UC02_LL-UC03: Energy clearing price determination
Version	V0.3
Authors	TUC
Last Update	1st Version in D2.2 2nd Version in D2.4 Updated within June 2019
Brief Description	The energy trading price is determined by intersecting the two curves obtained aggregating and sorting respectively the energy supply offers and energy demand bids.

<i>Assumptions and Pre-Conditions</i>	Market session is opened and bids/offers have been submitted.
<i>Goal (Successful End Condition)</i>	The aim of this UC is to determine the energy trading price per market session.
<i>Post-Conditions</i>	The energy trading price is determined.
<i>Involved Actors</i>	-
<i>Use Case Initiation</i>	End of market session.
<i>Main Flow</i>	<p>Begin</p> <ol style="list-style-type: none"> 1. The Secured Blockchain-driven Energy Market requests the smart contract's conditions for energy supply offers (for each market session) from the component Distributed Ledger. 2. The Distributed Ledger returns the requested data. 3. The Secured Blockchain-driven Energy Market requests the smart contract's conditions for energy demand bids (for each market session) from the component Distributed Ledger. 4. The Distributed Ledger returns the requested data. – (Steps 1 & 2 are executed in parallel with the steps 3 &4). 5. The Secured Blockchain-driven Energy Market calculates the energy clearing price by determining the intersection point between the two curves (energy supply offers in ascending order and energy demand bids in descending order).
<i>Alternative Courses</i>	-
<i>Relationships with other Use Cases</i>	HL-UC02_LL-UC02, HL-UC02_LL-UC04
<i>Architectural Elements / Services Involved</i>	Secured Blockchain-driven energy market; Distributed Ledger;
Specific Description	
<i>Relevance to eDREAM WPs</i>	WP5
<i>Main Tasks Involved</i>	T5.1 & T5.2
<i>Main Technical Partners Involved</i>	TUC, ENG, E@W, EMOT, CERTH, TU
<i>Notes (Optional)</i>	-

UML Sequence Diagram

Generic Description	
Use Case Name	HL-UC02_LL-UC04: Transactions validation and financial settlement
Version	V0.3
Authors	TUC
Last Update	1st Version in D2.2 2nd Version in D2.4 Updated within June 2019
Brief Description	The energy transactions are validated and the prosumer accounts settled allocating tokens to the prosumers accounts/wallets.

<i>Assumptions and Pre-Conditions</i>	Market session is ended.
<i>Goal (Successful End Condition)</i>	The aim of this UC is to allocate tokens to the prosumers' accounts/wallets.
<i>Post-Conditions</i>	Tokens are allocated to prosumers.
<i>Involved Actors</i>	Prosumers
<i>Use Case Initiation</i>	After the end of market session, 1 prosumer is selected as block miner.
<i>Main Flow</i>	<p>Begin</p> <ol style="list-style-type: none"> 1. The prosumer sends signal, in order to aggregate transactions in a block through the web interface of the HMI. 2. The HMI requests the validated transactions from the component Closed-loop DR Verification Engine. 3. The Closed-loop DR Verification Engine requests the committed energy assets from the component Distributed Ledger. 4. The Distributed Ledger sends the received request to the Secured Blockchain-driven Energy Market. 5. The Secured Blockchain-driven Energy Market sends the committed energy assets to the Closed-loop DR Verification Engine. 6. The Closed-loop DR Verification Engine sends the locked energy tokens to the Distributed Ledger. 7. The Closed-loop DR Verification Engine processes the received actual energy consumption/production data. 8. The Closed-loop DR Verification Engine sends the actual consumption/generation data to the HMI. 9. The Closed-loop DR Verification Engine generates and validates transaction. 10. The Closed-loop DR Verification Engine sends the validated energy transactions and the settled prosumer's account to the HMI. 11. The HMI processes and visualizes the received data, in order to inform prosumer.
<i>Alternative Courses</i>	-

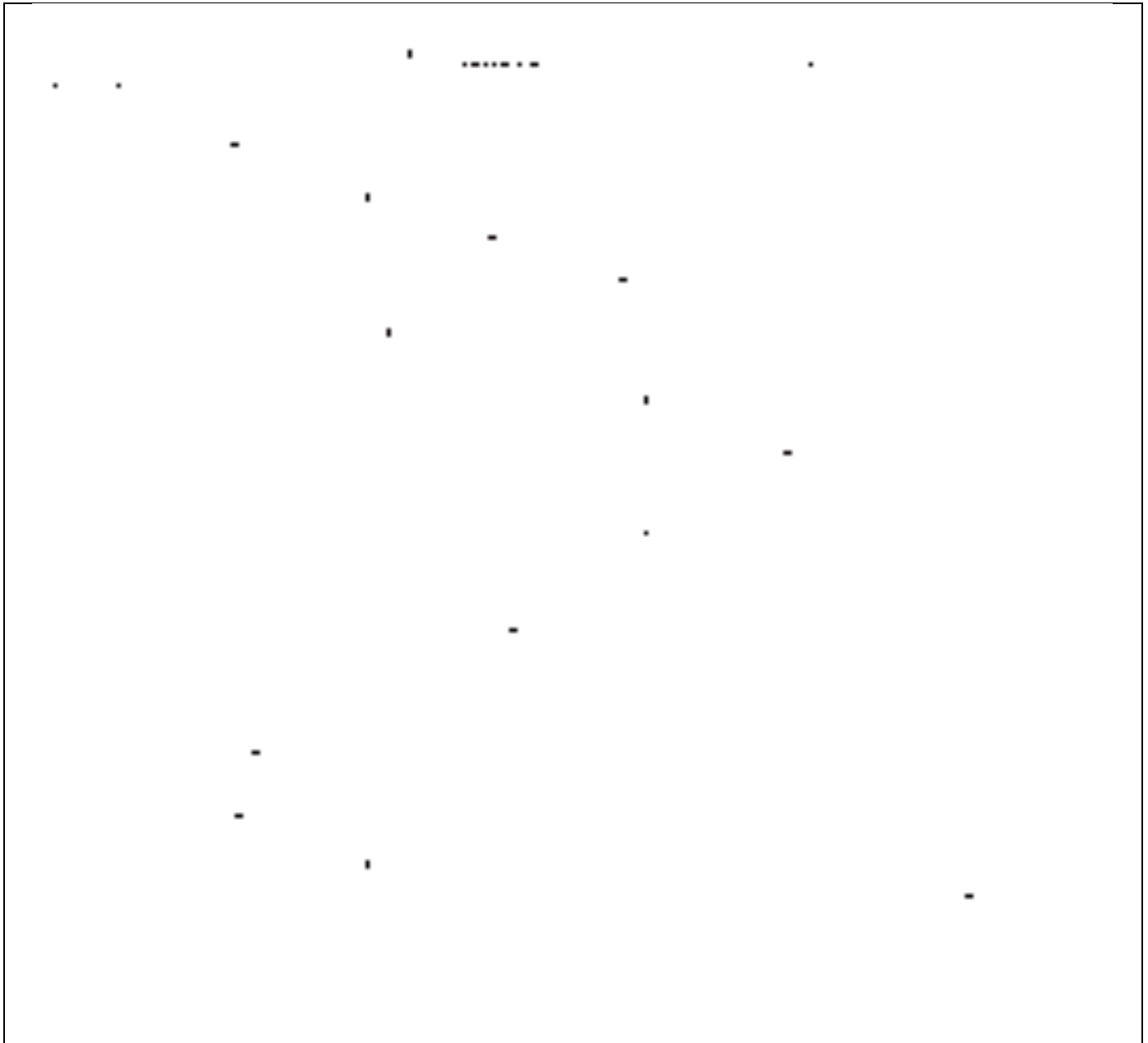
<i>Relationships with other Use Cases</i>	HL-UC02_LL-UC02, HL-UC02_LL-UC03
<i>Architectural Elements / Services Involved</i>	HMI; Closed-loop DR Verification Engine; Secured Blockchain-driven energy market; Distributed Ledger;
Specific Description	
<i>Relevance to eDREAM WPs</i> <i>Main Tasks Involved</i> <i>Main Technical Partners Involved</i>	WP4 & WP5 T4.3, T5.1, T5.2 & T5.3 TUC, ENG, E@W, EMOT
<i>Notes (Optional)</i>	-
UML Sequence Diagram	
	

This scenario is considering the increasing need to optimize output from multiple local generation assets (e.g. wind-turbines, small hydro, photovoltaic and back-up generators) that serve primarily local communities and also have export connections at power distribution network.

Generic Description	
Use Case Name	HL-UC03_LL-UC01: Prosumers Profiling and Clusterization
Version	V0.4
Authors	ATOS, ENG, E@W, ASM
Last Update	1st Version in D2.2 2nd Version in D2.4 Updated within June 2019
Brief Description	The Aggregator requests clusters of prosumers according to some criteria, in order to categorize their participation in ancillary and balance markets.
Assumptions and Pre-Conditions	Prosumers have accepted to participate in Demand Response programs.
Goal (Successful End Condition)	The aim of this UC is enable the Aggregator to manage properly prosumers with similar energy profiles considering also some other business criteria.
Post-Conditions	Prosumers are grouped into clusters.
Involved Actors	Aggregator, Prosumers
Use Case Initiation	The Aggregator requests the clusters of his/her prosumers.
Main Flow	<p>Begin</p> <ol style="list-style-type: none"> 1. The Aggregator requests the subsets of prosumers according to some criteria through the web interface of the Graph-based Analytics. 2. The Graph-based Analytics sends the request to the database (Cassandra DB). 3. The database requests the clusters of prosumers from the Big Data Clustering component.

	<ol style="list-style-type: none"> 4. The Big Data Clustering requests the customers' profiles from the Load Profiling component. 5. The Load Profiling requests historical data for customers' consumption from the database. 6. The database sends the requested data. 7. The Load Profiling calculates customers' profiles based on historical data. 8. The Load Profiling requests consumption forecasted data from the Electricity Consumption/Production Forecasting component. 9. The Electricity Consumption/Production Forecasting sends the requested data. 10. The Load Profiling updates the calculated profiles according to forecasted data. 11. The Load Profiling sends customers' profiles to the Big Data Clustering. 12. The Big Data Clustering clusterizes the received profiles considering the requested criteria. 13. The Big Data Clustering sends the created clusters to the database. 14. The database sends the received clusters to the Graph-based Analytics, in order to be visualized. 15. The Graph-based Analytics processes the received data and informs the Aggregator. 16. If new prosumers are registered, the Aggregator, through the web interface, sends a request to the database to categorize the new prosumers to the created clusters. 17. The database sends the received request and the created clusters to the Customer Segmentation. 18. The Customer Segmentation sends the classification of new prosumers to the database.
<i>Alternative Courses</i>	-


<i>Relationships with other Use Cases</i>	HL-UC03_LL-UC02, HL-UC03_LL-UC03
<i>Architectural Elements / Services Involved</i>	Graph-based Analytics; Database (Cassandra DB); Big Data Clustering at multiple scales; Load Profiling; Electricity Consumption/Production Forecasting; Customer Segmentation;
<i>Specific Description</i>	
<i>Relevance to eDREAM WPs</i>	WP3 & WP4
<i>Main Tasks Involved</i>	T3.1, T3.2, T3.4, T4.2, T4.3
<i>Main Technical Partners Involved</i>	TU, ATOS, CERTH, TUC, E@W
<i>Notes (Optional)</i>	-
<i>UML Sequence Diagram</i>	



Generic Description	
Use Case Name	HL-UC03_LL-UC02: VPP Capability Evaluation for Reserve services and for Frequency services
Version	V0.4
Authors	ATOS, ENG, E@W, ASM, KIWI

<i>Last Update</i>	1st Version in D2.2 2nd Version in D2.4 Updated within June 2019
<i>Brief Description</i>	<p>Aggregator estimates the capability of its portfolio of prosumers for ancillary services market participation:</p> <ul style="list-style-type: none"> • VPP for Reserve services: Aggregator estimates the capability of its portfolio of prosumers maximizing the utilization and revenues from RES and classic generation sources through accessing Reserve markets as an aggregated portfolio. By combining different types of RES, a more stable export, that is easier to predict and to be assigned to specific products in the Reserve market. • VPP for Frequency services: Aggregator estimates the capability of its portfolio of prosumers for frequency services. Generation and turn-down assets which do not meet the response time requested by the Frequency markets are excluded from the portfolio and the qualified assets are assigned to specific services (Dynamic, Static, Enhanced) based on their generation profile.
<i>Assumptions and Pre-Conditions</i>	Aggregator has access to a large number of prosumers, participating in Energy Community programmes with different RES and classic generation sources.
<i>Goal (Successful End Condition)</i>	Maximize utilization and revenues from RES and classic generation sources through accessing Reserve and Frequency markets as an aggregated portfolio.
<i>Post-Conditions</i>	The Aggregator knows the capability of his/her portfolio considering optimal set points and response times of dispatchable generators and capacity of curtailable loads.
<i>Involved Actors</i>	Aggregator, Prosumers
<i>Use Case Initiation</i>	The Aggregator needs to know the capability, in order to provide ancillary services, such as reserve services or frequency services.
<i>Main Flow</i>	<p>Begin</p> <ol style="list-style-type: none"> 1. The Aggregator requests the capability of prosumers coalitions registered in the portfolio for reserve and/or frequency services through the web interface of the DSS & DR Strategies Optimization UI. 2. The DSS & DR Strategies Optimization requests optimal set points and response times for generators and capacity of curtailable loads from the VPP and DR Services Optimization Engine.

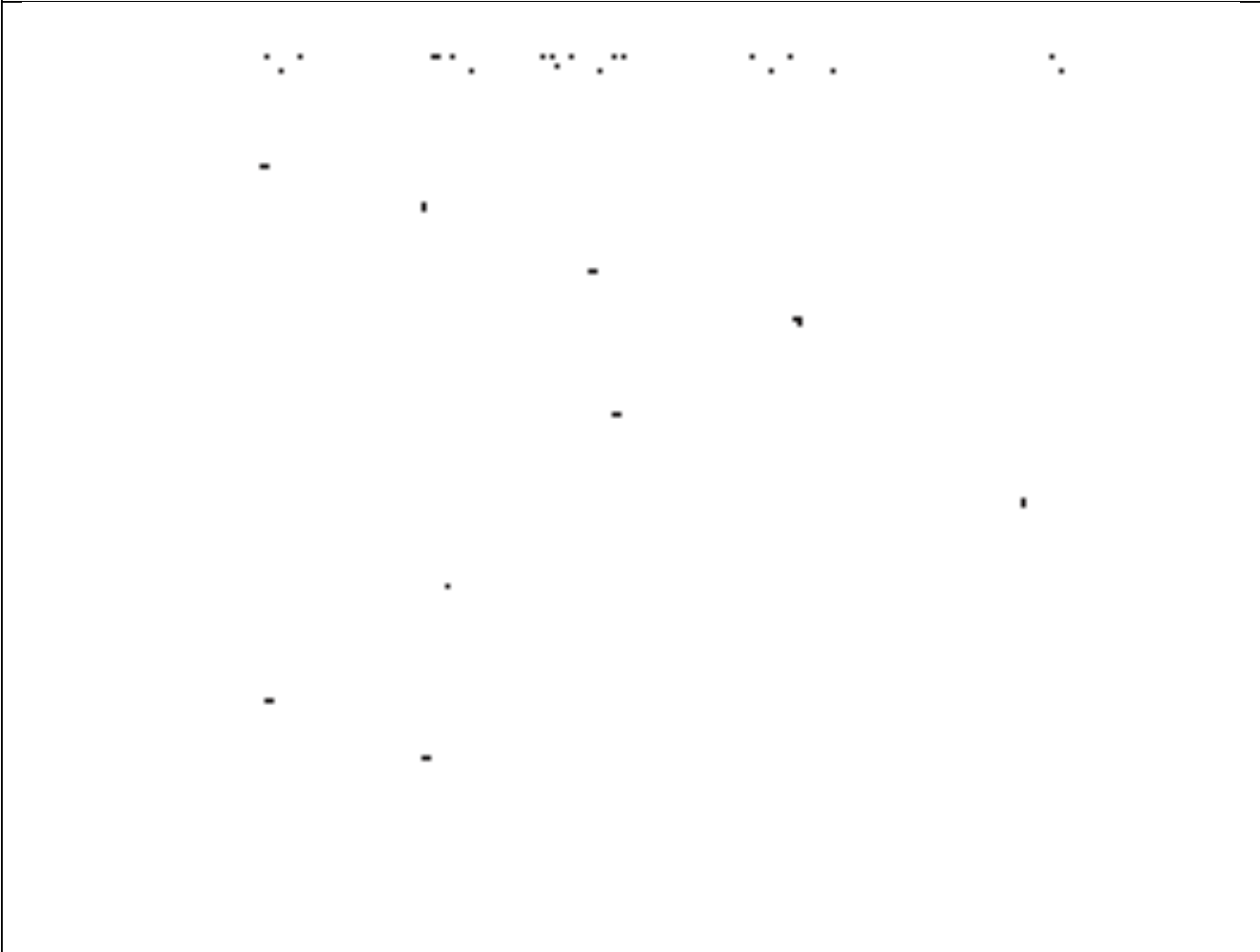
	<ol style="list-style-type: none"> 3. The VPP and DR Services Optimization Engine requests the optimal coalitions from the Virtual Power Plants Generation Modeling & Forecasting. 4. The Virtual Power Plants Generation Modeling & Forecasting sends the optimal coalitions. For each prosumer of optimal coalitions, the steps 5-10 are repeated. 5. The VPP and DR Services Optimization Engine requests consumption/production forecast for the prosumers participating in the optimal coalitions. 6. The Electricity Consumption/Production Forecasting sends the forecasted data. 7. The VPP and DR Services Optimization Engine requests PV/RES degradation rate for the energy assets participating in the optimal coalitions. 8. The PV/RES Degradation & Trend Analysis sends the requested data. 9. The VPP and DR Services Optimization Engine requests baseline flexibility assessment of prosumers' loads. 10. The Baseline Flexibility Estimation sends the requested data. 11. The VPP and DR Services Optimization Engine calculates the optimal set points. 12. The VPP and DR Services Optimization Engine sends the calculated data to the DSS & DR Strategies Optimization. 13. The DSS & DR Strategies Optimization processes the received data and visualizes them, in order to inform the aggregator about the capability of his/her portfolio.
<i>Alternative Courses</i>	The Aggregator is not able to achieve the capability required for implementation of ancillary services.
<i>Relationships with other Use Cases</i>	HL-UC03_LL-UC01
<i>Architectural Elements / Services Involved</i>	DSS & DR Strategies Optimization UI; VPP and DR Services Optimization Engine; Virtual Power Plants Generation Modeling & Forecasting; Electricity Consumption/Production Forecasting;

	PV/RES Degradation & Trend Analysis; Baseline Flexibility Estimation;
Specific Description	
Relevance to eDREAM WPs	WP3 & WP4
Main Tasks Involved	T3.1, T3.2, T3.3, T3.4, T4.1, T4.2, T4.3
Main Technical Partners Involved	TUC, ENG, ASM, EMOT, CERTH
Notes (Optional)	-
UML Sequence Diagram	
	

Generic Description	
Use Case Name	HL-UC03_LL-UC03: VPP Export Evaluation for Wholesale market (Intraday trading) and for Imbalance market

<i>Version</i>	V0.4
<i>Authors</i>	ENG, E@W, ASM, KIWI
<i>Last Update</i>	1st Version in D2.2 2nd Version in D2.4 Updated within June 2019
<i>Brief Description</i>	<p>Aggregator accurately estimates 30 minutes' generation and load forecasts to perform big data analysis in order to profile loads to be shed and to identify set points of dispatchable generators as well as response times from each type of generation asset. Aggregator need to know the VPP export capacity to implement trading services such as intraday trading and imbalance market. These trading services are described in brief below:</p> <ul style="list-style-type: none"> • VPP for Wholesale market (Intraday trading): VPP launches an offer on the wholesale market for the next 30 minutes' slot. The offer is based on the forecasted generation and flexibility availability for a 30 minutes trading window received one hour ahead. At the end of the 30 minutes trading interval, the offer is locked, price is cleared and the VPP received a committed capacity order for the market which is delivered over the next 30 minutes. If clearing price is still above the thresholds, back to step 1 for the next 30 minutes' window. If not, VPP export availability is handed over to other markets. A reference to a price forecasting module is needed and we can assume that the Aggregator already has / it's provided by a third party. • VPP for Imbalance market: VPP launches an offer to its partners trading on the imbalance market to provide capacity under the settlement price for the next 30 minutes' period. At the end of the 30 minutes trading interval, the offer is locked, price is cleared and the VPP received a committed capacity order from its partner which is delivered over the next 30 minutes. If imbalance settlement price forecast is still above the threshold for the next 30 minutes' period, back to step 1 for the next 30 minutes' window. If not, VPP export availability is handed over to other markets.
<i>Assumptions and Pre-Conditions</i>	Aggregator has access to a large number of prosumers, participating in Energy Community programmes with different RES and classic generation sources. The Aggregator has also a supply license or an agreement with a supplier to be allowed to bid in the Wholesale and Imbalance market.
<i>Goal (Successful End Condition)</i>	Maximize utilization and revenues from RES and classic generation sources through accessing Wholesale and Imbalance markets as an aggregated portfolio.
<i>Post-Conditions</i>	The Aggregator knows the forecasted export capacity of the VPPs for the next 30 minutes window.
<i>Involved Actors</i>	Aggregator, Prosumers

<i>Use Case Initiation</i>	The Aggregator needs to know the forecasted capacity, in order to implement trading services, such as intraday trading and trading in the imbalance market.
<i>Main Flow</i>	<p>Begin</p> <ol style="list-style-type: none"> 1. The Aggregator requests the forecasted export capacity of VPP for the next 30 minutes window through the web interface of the DSS & DR Strategies Optimization UI. 2. The DSS & DR Strategies Optimization UI sends the request to the VPP and DR Services Optimization Engine. 3. The VPP and DR Services Optimization Engine requests the generation and load profiles of VPP optimal coalitions from the VPP and active Microgrid Flexibility Profiling. 4. The VPP and active Microgrid Flexibility Profiling requests the VPP optimal coalitions from the Virtual Power Plants Generation Modeling & Forecasting. 5. The Virtual Power Plants Generation Modeling & Forecasting sends the requested data. 6. The VPP and active Microgrid Flexibility Profiling processes the received data and calculates the VPP flexibility profiles. 7. The VPP and active Microgrid Flexibility Profiling sends the VPP coalitions profiles to the VPP and DR Services Optimization Engine. 8. The VPP and DR Services Optimization Engine requests 30 minutes generation and load forecasts for the VPP profiles. 9. The Electricity Consumption/Production Forecasting sends the requested data. 10. The VPP and DR Services Optimization Engine calculates the export capacity for delivery. 11. The VPP and DR Services Optimization Engine sends the calculated data to the DSS & DR Strategies Optimization UI, in order to inform aggregator. 12. If clearing price is above the threshold, the Aggregator requests forecasted export capacity for the next 30 minutes window.
<i>Alternative Courses</i>	The Aggregator is not able to achieve the capability required for implementation of trading services.
<i>Relationships with other Use Cases</i>	HL-UC03_LL-UC01

<i>Architectural Elements / Services Involved</i>	DSS & DR Strategies Optimization UI; VPP and DR Services Optimization Engine; VPP and active Microgrid Flexibility Profiling; Virtual Power Plants Generation Modeling & Forecasting; Electricity Consumption/Production Forecasting;
Specific Description	
<i>Relevance to eDREAM WPs</i>	WP3 & WP4
<i>Main Tasks Involved</i>	T3.1, T3.3, T4.1, T4.2, T4.3
<i>Main Technical Partners Involved</i>	TUC, ATOS, CERTH
<i>Notes (Optional)</i>	-
UML Sequence Diagram	
	

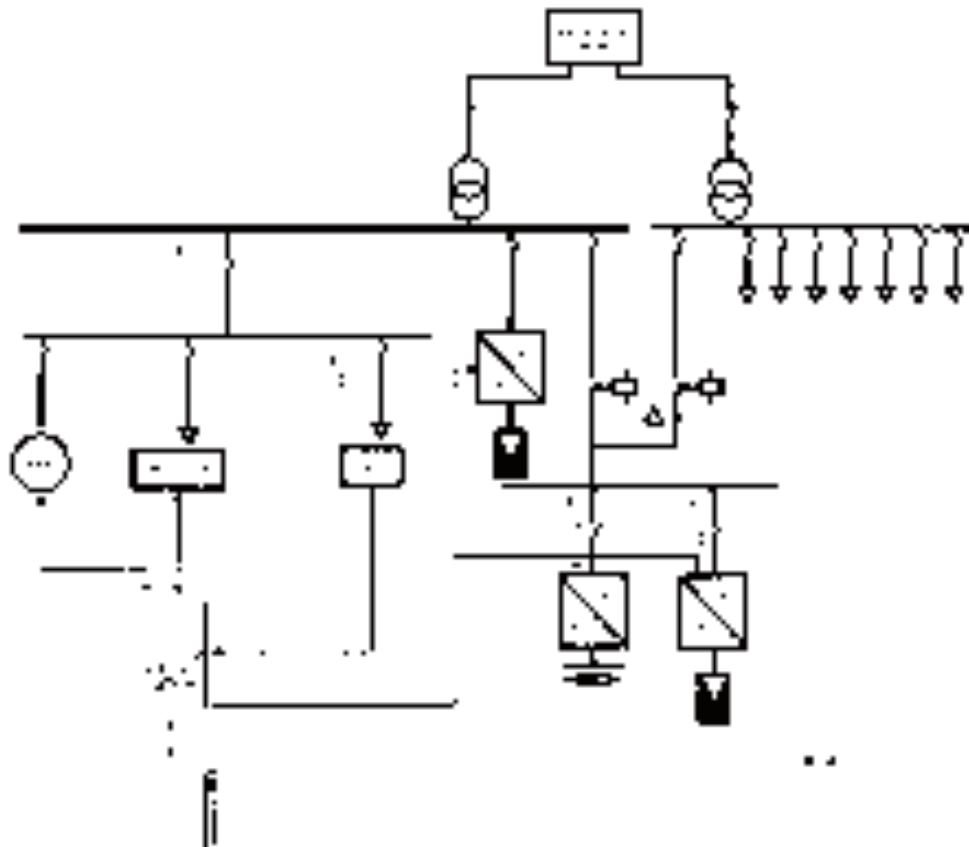
The Deployment View presents aspects of the system that are connected with the realization of the system's components in the physical world. This view defines the physical entities of the environment, in which the system is intended to perform its running processes and operations, including:

- 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, is described in brief in the following two sections.

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:

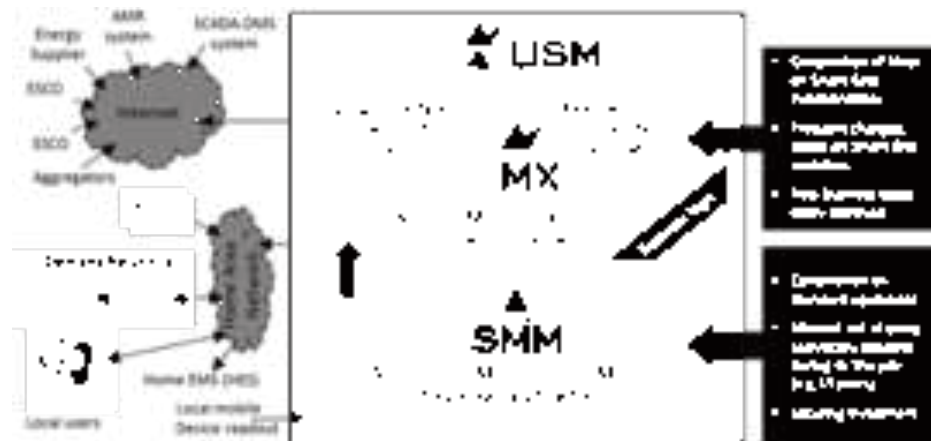


The ASM electric grid is characterized by a large number of distributed renewable energy sources embedded in the medium and low voltage distribution networks. It is worth pointing out that the measurements provided by the ASM electric grid for the year 2016 indicate that the yearly RPF measured in the 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 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.

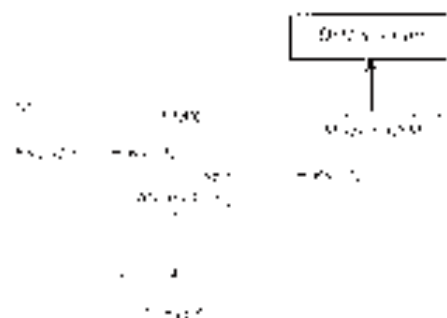
In order to get real time measurements and support real time operations in a Smart Grid environment, the Terni Micro-Grid is equipped with advanced smart meter technologies, such as the following:

- 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.



As shown in the figure 6, 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 following figure.



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, as depicted in the figure 5, the Terni microgrid includes three electric vehicle (EV) charging stations (two SPOTLINK – EVO filling stations and one EV fast charger) and three EVs which are provided by the company EMOTION. Thus, two fundamental devices related to EVs are also included in the metering equipment. These devices are the Electric Vehicle Supply Equipment (EVO Emotion) and the EV on-board diagnostic device (OBD Emotion).

Concerning the 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.

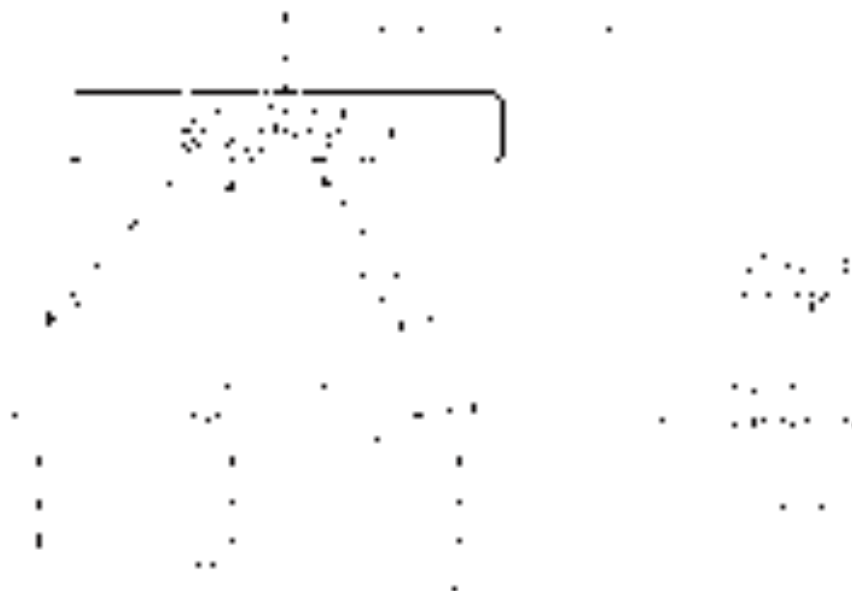
During the application of uses proposed in eDREAM, KiWi can provide specific tools and components or platform as a whole according to the following:

- 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 and indoor air quality monitoring equipment to test residential demand response scenarios.

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 9:

Taking into account the intension 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 Agents² and FIWARE Orion Context Broker³. 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:



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

² <https://github.com/telefonicaid/iotagent-node-lib>

³ <https://github.com/telefonicaid/fiware-orion>

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 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 Kong⁴ and Keycloak⁵. For both technologies, the PostgreSQL⁶ 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 RabbitMQ⁷ 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 docker⁸ 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.

Another important part of the deployment of the eDREAM platform is the infrastructure of the third layer of the core platform that is based on blockchain-driven technology for DR energy transactions modeling, tracking and decentralized control. A high-level deployment diagram of this infrastructure is depicted in the following figure.

⁴ <https://konghq.com/kong-community-edition/>

⁵ <https://www.keycloak.org/>

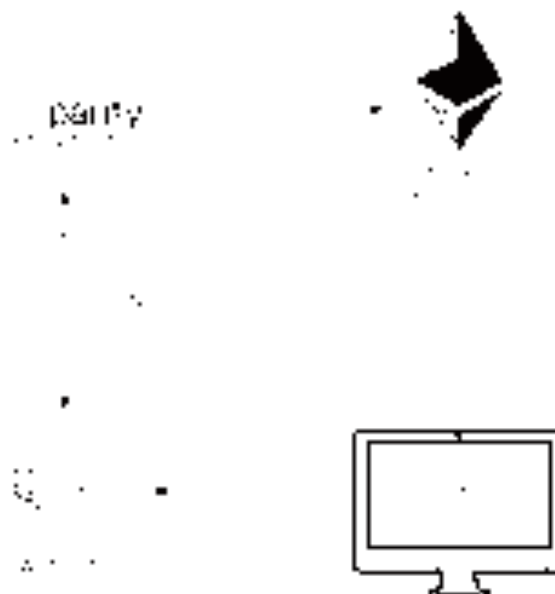
⁶ <https://www.postgresql.org/>

⁷ <https://www.rabbitmq.com/>

⁸ <https://www.docker.com/>, <https://docs.docker.com/compose/>




For the testing and implementation of the secure distributed ledger technology, the Parity Ethereum Client⁹ 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 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

⁹ <https://www.parity.io/ethereum/>


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 (see Annex III).

Device: Smart Metrology Meter	
Name	Smart Metrology Meter (SMM)
Short Description	<p>Wally-A is used in the pilot as SMM, it is used for monitoring the block of energy units within the pilot a as well as it can provides power analysis.</p> 
Measurement	<p>Electrical values:</p> <ul style="list-style-type: none"> § 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 with the standard CEI EN61000-4-30
Physical Characteristics	
Dimensions	Rack 19" Standard – 3U
Weight	Kg 3.2
Material	Plastic and steel
Mounting	Compliant with the rack standard
Hardware Requirements	
Power Requirements	<p>Voltage: 80÷275 Vac/dc - 50/60Hz</p> <p>Power consumptions: 20 VA</p>

	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, otherwise
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	-


Device: Smart Meter Extension	
Name	Smart Meter Extension (SMX)
Short Description	<p>Smart meter extension is a result of Nobel GRID project 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).</p> 
Measurement	All the data collected by the meter
Digital/Analog Signals	Digital signals
Functionality	Data gathering
Physical Characteristics	
Dimensions	15X14X17

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	-


Device: EVO Emotion	
Name	EVO Emotion
Short Description	<p>Electric Vehicle Supply Equipment</p> 

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(triphas)
Data Connections	GSM
Data Format	OCPP
Data Size	Some kBs
Data Availability	Continuous or On Demand
Transmission Frequency	Each 5 seconds
Software Requirements (e.g. API creation)	
Software Required	Yes
Software Details	Software owned by 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 (Opened/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	Each 5 seconds
Software Requirements (e.g. API creation)	
Software Required	Yes

Software Details	Software owned by Emotion
-------------------------	---------------------------

Device: Fruit	
Name	Fruit KiWi Power
Short Description	<p>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.</p> 
Measurement	<p>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).</p>
Digital/Analog Signals	-
Functionality	<p>Enabling centrally-dispatched Demand Response programmes 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;</p>

	DNO constraint management; Remote Telecoms Unit (RTU) for TSO integration.
Physical Characteristics	
Dimensions	114 x 100
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	-

<i>Name of Existing Component/Service</i>	<i>Micro-grid Monitor</i>	
<i>Type</i>	Service	
<i>Functionality</i>	Data gathering from the energy units	
<i><u>Output Connections & Interfaces: To which architectural component it is offering services</u></i>	<ul style="list-style-type: none">• Distributed Ledger• Blockchain-driven control for LV networks (flexibility management)• Electricity Production/Consumption Forecasting• Big Data Clustering at Multiple Scales• Closed loop DR Verification Engine	
<i><u>Input Connections & Interfaces: which other components are using it</u></i>	Field data aggregation	
<i>Functional Requirements</i>	FD_BR01_UR01_FR01	
<i>Non-Functional Requirements</i>	-	
<i><u>Input Parameters</u></i>		
<i>Description</i>	<i>Modelling format</i>	<i>Data Received From</i>
§ 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	.pqdif	Wally-A
<i><u>Output Parameters</u></i>		
<i>Description</i>	<i>Modelling format</i>	<i>Data Sent To</i>
§ Voltages and Currents § Active, Reactive and Apparent power § Active and Reactive (4 quadrants) Energy § Power Factor § Frequency	.pqdif	Wally-A

§ 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		
Communications	FTP protocol	
License	-	
Technology Readiness Level (TRL)	9	
Hardware Requirements	Wally-A CPU: 2 core 3.1 GHz HDD: 50 GB RAM: 4 GB S.O.: Ubuntu 15.04	
Development Language	N/A	
Other Resources Required	-	

Name of Existing Component/Service	EVSEs and EV fleet monitoring
Type	Service
Functionality	To perform DR campaign, Fleet Manager have to constantly monitoring EVSEs and EV fleet
<u>Output Connection & Interfaces:</u> To which architectural component it is offering services	<ul style="list-style-type: none"> Electricity Consumption/Production Forecasting Baseline Flexibility Estimation VPP and active Microgrid Flexibility Profiling Load Profiling VPP and DR Services Optimization Engine Closed Loop DR Verification Engine Blockchain-driven Control for LV Networks DSS & DR Strategies Optimization
<u>Input Connections & Interfaces:</u> which other components are using it	-
Functional Requirements	MF02_BR07_UR01_FR01

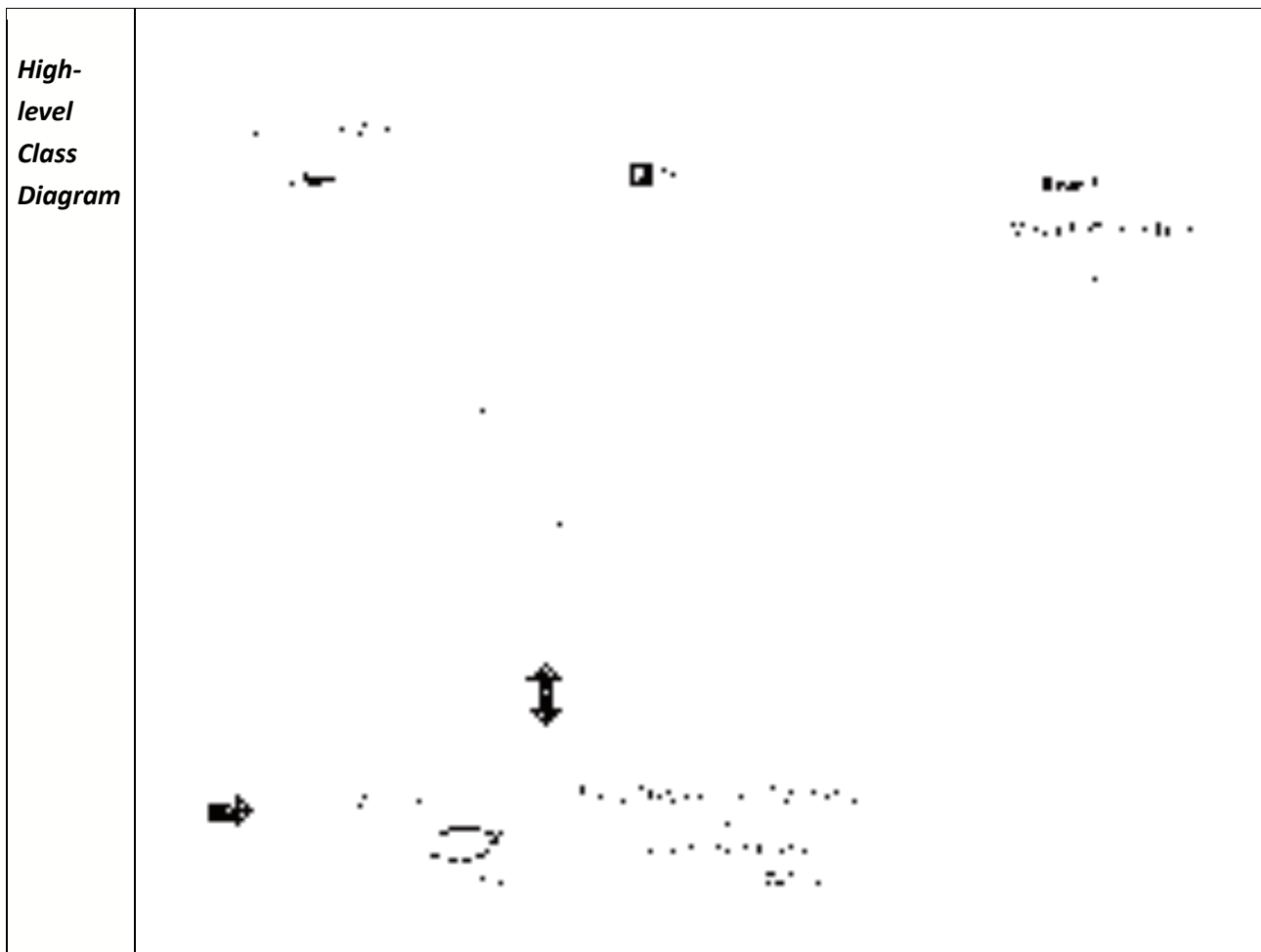
	MF02-BR07-UR02_FR02	
Non-Functional Requirements	-	
<u>Input Parameters</u>		
Description	Modelling format	Data Received From
Battery State-of-Charge (percentage)	JSON	Electric Vehicle
Residual Autonomy (kilometres)	JSON	Electric Vehicle
Needed time to Full Charge (minutes)	JSON	Electric Vehicle
Geolocation (geographic coordinates)	JSON	Electric Vehicle
Doors Car State (Opened/Closed)	JSON	Electric Vehicle
Engine Car State (On/Off)	JSON	Electric Vehicle
Energy data (Power, Voltage, Current)	JSON	Charging Station
Number of plugs in use (0/1/2)	JSON	Charging Station
Alarms (electrical problems, connection problems)	JSON	Charging Station
<u>Output Parameters</u>		
Description	Modelling format	Data Sent To
[] – array of EV real time monitored data on the start-end interval, given the sampling rate	JSON	Electric Vehicle
[] – array of EVSE real time monitored data on the start-end interval, given the sampling rate	JSON	Charging Station
Communications	-	
License	-	
Technology Readiness Level (TRL)	9	
Hardware Requirements	CPU: 2 core 3.1 GHz HDD: 50 GB RAM: 4 GB S.O.: Ubuntu 15.04	
Development Language	Java	
Other Resources Required	-	

Name of New Component/Service:	Electricity consumption/production forecasting
Type:	Component
Functionality:	Built on top of Energy Budget Broker implements several big data analysis and deep learning techniques for accurate predictions of energy supply and demand at different levels of granularities (scale / time)
Provided Services	<p>Energy demand and production forecasting for Day-ahead, intra-day, one hour ahead time frames</p> <p>Flexibility forecasting for day-ahead and intr-day time frames</p>
<u>Input Connections & Interfaces:</u> From which components it receives input	<ul style="list-style-type: none"> • Cassandra DB • Baseline Flexibility Estimation
<u>Output Connections & Interfaces:</u> To which components it sends the results	<ul style="list-style-type: none"> • Virtual Power Plants Generation Modeling & Forecasting • Load Profiling • PV/RES Degradation & Trend Analysis • VPP and DR Services Optimization Engine • Blockchain-driven control for LV networks • Secured Blockchain-driven Energy market • DSS & DR Strategies Optimization • Forecasting Tool • Local/Remote HMIs <p>REST API: <host>:<port>/edream-ebb/</p>
Functional Requirements	<p>MF01_BR02_UR01_FR01, MF01_BR02_UR02_FR02</p> <p>MF01_BR02_UR03_FR03, MF01_BR02_UR04_FR04</p> <p>MF01_BR02_UR05_FR05, MF02_BR07_UR01_FR06</p>

			MF02_BR07_UR02_FR07, MF02_BR07_UR03_FR08		
Non-Functional Requirements			-		
Input Parameters					
Attribute/ Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From
Historical data	A json containing the monitored values for a period	JSON	[{ "timestamp": 1537803600000, "value": 908.483, "deviceMeasurementI d": "18bb6e98-8429-4162- 8236-cc4f231bb2a4", "deviceId": "4836bee7-bc42-48a9- 9e4e-aa1ecc68e6d8", "property": "Power Consumption" }, { "timestamp": 1537341600000, "value": 910.54, "deviceMeasurementI d": "18bb6e98-8429-4162- 8236-cc4f231bb2a4", "deviceId": "4836bee7-bc42-48a9- 9e4e-aa1ecc68e6d8", "property": "Power Consumption" }, ...]]	Range – depending on the monitored device Frequency: every5 min	Cassandra DB
Baseline Values	A json containing the baseline values for a period		[{ "timestamp": 1537803600000, "value": 908.483, "deviceMeasurementI d": "18bb6e98-8429-4162- 8236-cc4f231bb2a4",	Range – Frequency:	Baseline Flexibility Estimation

			<pre> "deviceId": "4836bee7-bc42-48a9- 9e4e-aa1ecc68e6d8", "property": "Baseline Consumption" }, { "timestamp": 1537803600000, "value": 908.483, "deviceMeasurementI d": "18bb6e98-8429-4162- 8236-cc4f231bb2a4", "deviceId": "4836bee7-bc42-48a9- 9e4e-aa1ecc68e6d8", "property": "Baseline Consumption" } ...] </pre>		
<u>Output Parameters</u>					
Attribute /Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Sent To
<i>Forecasted values</i>	The module will provide an array containing the device and the measurement the forecast is referring to, together with the forecasted values		<pre> { "profile": [{ "value": 57167.566, "timestamp": "2018-09-08T00:00:00" }, { "value": 57174.233, "timestamp": "2018-09-08T01:00:00" }, ...], "entityDeviceId": "ab7d658d-84cd-4662- b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951- 79910cd0e822", "profileGranularityMinutes" : 60 , "predictionGranularity": "DAYAHEAD", </pre>	Range – depending on the forecasted property Frequency: every 30 min, 1 hour	<ul style="list-style-type: none"> Virtual Power Plants Generation Modeling & Forecasting Load Profiling PV/RES Degradation & Trend Analysis VPP and DR Services Optimization Engine Blockchain-driven control for LV networks Secured Blockchain-driven Energy market

			"property": "ENERGY CONSUMPTION" }		<ul style="list-style-type: none"> • <i>DSS & DR Strategies Optimization</i> • <i>Forecasting Tool</i> • <i>Local/Remote HMIs</i>
Software Requirements/Development Language			<ul style="list-style-type: none"> • 64 bit Linux / Windows OS • JRE (Java Runtime Environment) version 8 or above • Internet connection for package management • <i>nVIDIA CUDA, Keras, TensorFlow / Python3,</i> • Cassandra, MySQL, RabbitMQ 		
Hardware Requirements			<ul style="list-style-type: none"> • Intel/AMD processor with at least 4 cores and 2.5GHz base frequency • 8 GB RAM • HDD/SSD with at least 100GB free space • Internet connection for package management • <i>NVIDIA GeForce GTX 1050 (equivalent or above)</i> 		
Communications			REST API , HTTP		
Status of the development of the component			Initial component prototype implemented		



Name of New Component/Service:	Virtual Power Plants Generation Modeling & Forecasting
Type:	Component
Functionality:	Provide innovative distributed electricity generation virtual aggregation techniques (prosumers coalitions construction in VPP) based on nature inspired heuristics, potentially taking advantage on the features provided by the block chain-based distributed computing platforms
Provided Services	Generate optimal coalition from a prosumers portfolio to

	<ul style="list-style-type: none"> - trade energy for increasing profit; - offer capacity bidding service; - participate to demand response programs - offer reactive power control services
<u>Input Connections & Interfaces:</u> From which components it receives input	<ul style="list-style-type: none"> • Electricity Consumption/Production Forecasting • PV/RES Degradation & Trend Analysis • Cassandra DB
<u>Output Connections & Interfaces:</u> To which components it sends the results	<ul style="list-style-type: none"> • VPP and active Microgrid flexibility profiling • VPP and DR Services Optimization engine <p>REST API: <host>:<port>/edream-vpp/</p>
Functional Requirements	<p>MF02_BR03_UR01_FR01, MF02_BR03_UR02_FR02</p> <p>MF02_BR03_UR03_FR03</p>
Non-Functional Requirements	-


Input Parameters

Attribute/ Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From
<i>Production/Consumption forecasts for each participating producer/grid</i>	An array containing the device and the measurement the forecast is referring to, together with the forecasted values	JSON	{ "profile": [{"value": 57167.566, "timestamp": "2018-09-08T00:00:00" }, {"value": 57174.233, "timestamp": "2018-09-08T01:00:00" }, ...], }	Range – depending on the forecasted property Frequency: every 30 min, 1 hour	<ul style="list-style-type: none"> • Electricity Production/Consumption Forecasting – Prosumer Level/Grid Level

			<pre>"entityDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "profileGranularityMinutes": 60 , "predictionGranularity": "DAYAHEAD", "property": "ENERGY CONSUMPTION" }</pre>		
<i>Improved short-term forecasting information</i>	Information about the device degradation	JSON	<pre>{ "value": 100, "deviceMeasurementId": "12bb6e98-8429-4162-8236-cc4f231bb2a4", "deviceId": "4236bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "property": "Degradation-Trend" }</pre>	Range – Frequency: every 30 min, 1 hour	<ul style="list-style-type: none"> <i>PV/RES Degradation & Trend Analysis</i>
<i>Requested Goal</i>	Choose the goal to be followed by the prosumer optimization process		<pre>"goal": { type": "ENERGY TRADING", "priceSignal": [{"value": 157167, "timestamp": "2018-09-08T00:00:00"}, {"value": 157234, "timestamp": "2018-09-08T01:00:00"}, ...] }</pre>	Range – Frequency: on request	<ul style="list-style-type: none"> <i>HDMI</i>
<i>Prosumers portfolio</i>	The available prosumers portfolio		<pre>{ "prosumers": [{ "prosumerId": "4836bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "predictedProfile": { [{"value": 57167.566, "timestamp": "2018-09-08T00:00:00" }</pre>	Range – Frequency:	<ul style="list-style-type: none"> <i>Cassandra DB</i>

			<pre> }, {"value": 57174.233, "timestamp": "2018-09-08T01:00:00" }, ...], "entityDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "profileGranularityMinutes": 60 , "predictionGranularity": "DAYAHEAD", "property": "ENERGY PRODUCTION" }, "uncertainty": { "min": 0.8, "max": 1.2, "entityDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "property": "Degradation-Trend" } "prosumerDetails": { "specification": {...} </pre>		
--	--	--	--	--	--

			<pre> "type": "DEG" } },], } </pre>		
<u>Output Parameters</u>					
Attribute /Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Sent To
<i>Optimized prosumers aggregated in VPP</i>	An array containing all the prosumers that are part of the optimized VPP	JSON	<pre> { "coalitionId": "4726bee7- bc42-48a9-9e4e- aa1ecc68e6f1", "selectedProsumers": [{ "prosumerId": "4836bee7- bc42-48a9-9e4e- aa1ecc68e6d8", "prosumerType": "DEG", "tradedEnergy": [{"value": 57160, "timestamp": "2018-09- 08T00:00:00"}, {"value": 57170, "timestamp": "2018-09- 08T01:00:00"}, ...]}, ...] "totalEnergyTraded": [{"value": 157160, "timestamp": "2018- 09-08T00:00:00"}, {"value": 157230, "timestamp": "2018- 09-08T01:00:00"}, ...] </pre>	Range – Frequency: on request	<ul style="list-style-type: none"> VPP and active micro-grid flexibility VPP & DR Services Optimization Engine
Software Requirements/Development Language				<ul style="list-style-type: none"> 64 bit Linux / Windows OS JRE (Java Runtime Environment) version 8 or above Internet connection for package management 	
Hardware Requirements				<ul style="list-style-type: none"> Intel/AMD processor with at least 4 cores and 2.5GHz base frequency 8 GB RAM 	

	<ul style="list-style-type: none"> • HDD/SSD with at least 100GB free space • Internet connection for package management
Communications	REST API
Status of the development of the component	Initial component prototype implemented
High-level Class Diagram	

Name of New Component/Service:	Baseline Flexibility Estimation
Type:	Component
Functionality:	Estimation of the load profile before implementation of DR strategies - Considering the devices installed at the prosumer site, the aim is to compute the flexibility of the prosumer with respect to its normal functionality that is given by the baseline

Provided Services			Supply the load profile patterns identified according to the selected demand response strategy.		
<u>Input Connections & Interfaces:</u> From which components it receives input			<ul style="list-style-type: none">• Cassandra DB• Multi-building DR Characterization		
<u>Output Connections & Interfaces:</u> To which components it sends the results			<ul style="list-style-type: none">• VPP and Active Microgrid Flexibility Profiling• VPP and DR Services Optimization Engine• Blockchain-driven control for LV networks		
Functional Requirements			MF01_BR03_UR01_FR01, MF01_BR03_FR02 MF01_BR03_FR03, MF02_BR09_UR01_FR04 MF02_BR09_UR02_FR05		
Non-Functional Requirements			-		
<u>Input Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From
Prosumer baseline profiles	A profile for prosumer baseline flexibility estimation	Prosumer ID (int), DR Programme ID (int) starttimestamp (string), endtimestamp (string), baseline energy flexibility (float), monetary cost (float), comfort cost (float)	CSV, JSON or XML	Non-negative, granularity is 15 minutes	<ul style="list-style-type: none">• Cassandra DB• Multi-Building DR Characterization

Constraints of the installed devices					<ul style="list-style-type: none"> Cassandra DB
<u>Output Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Sent To
Array of estimated electricity demand baseline flexibility values		Prosumer ID (int)-DR programme ID (int)- Starttimestamp (string)- endtimestamp (string)-Energy consumption flexibility array (float[])-monetary cost during starttime and endtime (float []), comfort cost during starttime and end time (float[])	CSV, JSON or XML	Greater than 0, every 15 minutes	<ul style="list-style-type: none"> VPP and active Microgrid Flexibility Profiling VPP and DR Services Optimization Engine Blockchain-driven control for LV networks
Software Requirements/Development Language			The software requirement is C++ libraries and runtime environment. C++ will be used as programming language.		
Hardware Requirements			A Windows/Linux based PC with administrator right and credentials.		

	<i>In case it needs any special sensor that is included in the sensor specification, it can be included also here as a reference.</i>
Communications	<i>Ethernet or WiFi Connectivity</i>
Status of the development of the component	<i>To be developed from scratch</i>

Name of New Component/Service:	<i>PV/RES Degradation and Trend Analysis</i>
Type:	<i>Component</i>
Functionality:	<p><i>Calculates the degradation rate at which PV/RES modules lose their performance over time</i></p> <p><i>Improves the short-term forecasting of generation of the PV/RES units based on near real-time trend analysis algorithm</i></p>
Provided Services	<i>The output of this component will contribute to the calculation of DR optimization algorithms for short term DR planning (e.g. support day-ahead, direct trading, coupon-based DR programs etc.) and for long-term DR scheduling maximizing the benefits for DSOs, Aggregators and prosumers</i>
<u>Input Connections & Interfaces:</u> From which components it receives input	<ul style="list-style-type: none"> <i>Electricity consumption/production forecasting</i> <i>Cassandra DB</i>
<u>Output Connections & Interfaces:</u> To which components it sends the results	<ul style="list-style-type: none"> <i>Virtual Power Plants Generation Modelling and Forecasting</i> <i>VPP and DR Services Optimization Engine</i> <i>DSS & DR Strategies Optimization</i> <i>Forecasting Tool</i>

Functional Requirements			MF01_BR04_UR01_FR01, MF01_BR04_UR02_FR02, MF01_BR04_FR03, MF01_BR04_FR04		
Non-Functional Requirements			-		
Input Parameters					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From
PV Electricity production forecast	PV Electricity production forecast value	float	CSV/JSON	15 min	Electricity consumption/production forecasting
Weather information	forecast and historical Weather data	location (latitude, longitude) coordinates, Solar irradiation (W/m2), cloudiness (%), date and time (yy-mm-dd-hh-mm-ss)	JSON	15 min	Cassandra DB
PV Electricity production readings	Historical data of Electricity production of the PV	float	CSV/JSON	15 min	
Performance characteristics of the PV unit	Solar PV technical information Age of PV	Reated power in watts, reference	CSV/JSON	15 min	

	<i>panel in month/year</i>	<i>irradiance (1000w/m²)</i> <i>date (yy-mm)</i>			
<u>Output Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Sent To
<i>Improved short-term and long-term forecasting of generation</i>	<i>Improving the forecasting of generation of PV</i>	<i>Float</i>	<i>CSV/JSON</i>	<i>>=0</i> <i>15 min</i>	<ul style="list-style-type: none"> • <i>VPP Generation Modelling and Forecasting</i> • <i>VPP and DR Services Optimization Engine</i> • <i>DSS & DR Strategies Optimization</i> • <i>Forecasting Tool</i>
Software Requirements/Development Language			<i>The software requirement is C++ libraries and runtime environment. C++ will be used as programming language.</i>		
Hardware Requirements			<i>A Windows/Linux based PC with administrator right and credentials.</i>		
Communications			<i>Ethernet or WiFi Connectivity</i>		
Status of the development of the component			<i>Needs to be developed from scratch</i>		

Name of New Component/Service:	<i>Multi-building DR characterization through thermal, optical and LIDAR information fusion</i>
Type:	<i>Component</i>

Functionality:						An interface for gathering, processing and analyzing data gathered by aerial surveying activities, producing usable data for the purposes of estimating DR flexibility
Provided Services						Provide appropriate data and indexes after the image processing for building energy assessment
<u>Input Connections & Interfaces:</u> From which components it receives input						<ul style="list-style-type: none"> • Drone camera mount • LiDAR Ethernet port • Skyport API • Cassandra DB
<u>Output Connections & Interfaces:</u> To which components it sends the results						<ul style="list-style-type: none"> • 3D Workstation • Pix4D software • VeloView Software • Cassandra DB • Baseline Flexibility Estimation • DR Aerial Survey Toolkit
Functional Requirements						MF01_BR01_FR01, MF01_BR01_FR02 MF01_BR01_FR03, MF01_BR01_FR04
Non-Functional Requirements						-
<u>Input Parameters</u>						
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From	
LiDAR point cloud files	Data received from LiDAR	Point cloud	.pcap	n/a, once per scan	Velodyne VLP-16 LiDAR	
Aerial survey both visible and infrared spectrum images	Visible and infrared (thermal imaging) spectrum	bitmap	.jpeg	8 bit per channel RGBA	Drone camera	

	<i>images taken during the aerial survey</i>				
<i>Smart Meter data (KiWi)</i>					<i>Cassandra DB</i>
<i>Weather Data</i>	<i>Data from weather station.</i>	<i>Luminosity, temperature, humidity, rain, wind speed</i>	<i>.csv</i>	<i>15min frequency</i>	
<u>Output Parameters</u>					
<i>Attribute/Parameter</i>	<i>Short Description</i>	<i>Data Type</i>	<i>Data Format</i>	<i>Value Range & Frequency</i>	<i>Data Sent To</i>
<i>3d model</i>	<i>3d model made from pics taken during aerial survey.</i>	<i>geometry</i>	<i>.obj</i>	<i>n/a</i>	<i>3d workstation system</i>
<i>Output from image processing (CERTH)</i>	<i>Building thermal signature</i>	<i>Units of thermal leakage levels with respect to baseline</i>	<i>.csv</i>	<i>n/a, once per scan</i>	<i>Cassandra DB</i> <i>DR Aerial Survey Toolkit</i> <i>Baseline Flexibility Estimation</i>
<i>Software Requirements/Development Language</i>			Velodyne VeloView LiDAR processing software. Pix4D aerial survey software		
<i>Hardware Requirements</i>			Quadcopter Drone. We are using a DJI Matrice 210. Thermal imaging camera: DJI Zenmuse XT2 with 19mm lens, 30Hz refresh rate and 640x512 pixels resolution.		

	LiDAR: Velodyne VLP 16 ('Puck')
Communications	TCP/IP for Wireless Communications
Status of the development of the component	Partially developed.

Name of New Component/Service:			Big Data Analytics Engine Tool		
Type:			<i>Platform</i>		
Functionality:			<i>A set of orchestrated functionalities to provide data lifecycle management for energy domain advanced services</i>		
Provided Services			<i>Data processing</i>		
<u>Input Connections & Interfaces:</u> From which components it receives input			<ul style="list-style-type: none">• <i>Cassandra DB</i>• <i>Load Profiling</i>• <i>Electricity Consumption / Production Forecasting</i>• <i>Baseline Flexibility Estimation</i>		
<u>Output Connections & Interfaces:</u> To which components it sends the results			<ul style="list-style-type: none">• <i>Big Data Clustering at Multiple Scales</i>• <i>Load Profiling</i>• <i>HMI</i>		
Functional Requirements			<i>MF02_BR01_FR01, MF02_BR01_FR02</i> <i>MF02_BR01_FR03, MF02_BR01_FR04</i> <i>MF02_BR01_FR05, MF02_BR01_FR06</i> <i>MF02_BR01_FR07</i>		
Non-Functional Requirements			<i>MF02_BR01_NFR01, MF02_BR01_NFR02</i>		
<u>Input Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From

<i>Smartmeter_id</i>	<i>Unique identifier for the smartmeters</i>	<i>STRING</i>	<i>CSV</i>		<i>ASM</i>
<i>Energy Absorbed</i>	<i>Energy absorbed by the smartmeter</i>	<i>FLOAT</i>	<i>CSV</i>	<i>E. Abs >= 0 Freq = 15 min</i>	<i>ASM</i>
<i>Energy Injected</i>	<i>Energy injected to the grid by the Smartmeter</i>	<i>FLOAT</i>	<i>CSV</i>	<i>Energy Injected >= 0 Freq = 15 min</i>	<i>ASM</i>
<i>Date</i>	<i>Date: DD-MM-YYYY</i>	<i>STRING</i>	<i>CSV</i>		<i>ASM</i>
<u>Output Parameters</u>					
<i>Attribute/Parameter</i>	<i>Short Description</i>	<i>Data Type</i>	<i>Data Format</i>	<i>Value Range & Frequency</i>	<i>Data Sent To</i>
Data processing					<i>Big Data Clustering at multiple scales</i> <i>Load Profiling</i> <i>HMI</i>
<i>Software Requirements/Development Language</i>			<i>TCP/IP connectivity</i> <i>MQTT/AMPQ, HTTP as transport protocols</i>		
<i>Hardware Requirements</i>			<i>Cloud-based system</i>		
<i>Communications</i>			<i>Input data should be provided either by csv files or by API requests.</i>		

	<i>Output data shall be offered by API requests or by a context brokering service.</i>
Status of the development of the component	<i>On early development stage (5%-10%)</i>

Name of New Component/Service:			Load Profiling		
Type:			Component		
Functionality:			A non-intrusive appliance load analysis technique		
Provided Services			<ul style="list-style-type: none">Creates different profiles based on loads' measurements and forecastServes profiles to other components		
<u>Input Connections & Interfaces:</u> From which components it receives input			<ul style="list-style-type: none">Electricity Consumption/Generation ForecastCassandra DB /InfluxDB		
<u>Output Connections & Interfaces:</u> To which components it sends the results			<ul style="list-style-type: none">Big Data Clustering at Multiple ScalesVPP and active Microgrid Flexibility ProfilingCassandra DB/InfluxDB		
Functional Requirements			MF02_BR02_UR02_FR01, MF02_BR02_FR02 MF02_BR02_FR03, MF02_BR02_FR04 MF02_BR02_FR05, MF02_BR02_FR06 MF02_BR01_FR01, MF02_BR01_FR02		
Non-Functional Requirements			MF02_BR02_UR09_NFR01, MF02_BR01_NFR01 MF02_BR01_NFR02		
<u>Input Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From

<i>Prosumer ID</i>	<i>Prosumer Identifier</i>	<i>String</i>	<i>csv</i>	<i>E. Abs >= 0 Freq = 15 min</i>	<i>Cassandra DB</i>
<i>Historical data</i>	<i>Historical data for load consumption</i>	<i>Float</i>	<i>csv</i>	<i>E. Abs >= 0 Freq = 15 min</i>	
<i>Real-time data</i>	<i>Data from field devices</i>	<i>Float</i>	<i>csv</i>	<i>E. Abs >= 0 Freq = 15 min</i>	
<i>KPIs</i>	<i>Defines indicators</i>	<i>String</i>	<i>csv</i>	<i>-</i>	
<i>Forecasted data for prosumers' consumption</i>					<i>Electricity Consumption/Production Forecasting</i>
<u>Output Parameters</u>					
<i>Attribute/Parameter</i>	<i>Short Description</i>	<i>Data Type</i>	<i>Data Format</i>	<i>Value Range & Frequency</i>	<i>Data Sent To</i>
<i>Prosumers profiles</i>	<i>Pattern related to specific features</i>	<i>Float</i>	<i>csv</i>	<i>E. Abs >= 0 Freq = 15 min</i>	<i>Big Data Clustering at Multiple Scales VPP and active Microgrid Flexibility Profiling Cassandra DB/InfluxDB</i>
<i>Software Requirements/Development Language</i>			<i>TCP/IP connectivity MQTT/AMPQ, HTTP as transport protocols</i>		
<i>Hardware Requirements</i>			<i>Cloud-based system</i>		
<i>Communications</i>			<i>Input data should be provided either by csv files or by API requests.</i>		

	<i>Output data shall be offered by API requests or by a context brokering service.</i>
Status of the development of the component	<i>On early stage development (5%-10%)</i>

Name of New Component/Service:	Big Data Clustering at Multiple Scales
Type:	Component
Functionality:	Analytical component for clusterization of energy customers.
Provided Services	<ul style="list-style-type: none"> • Creates clusters based on load profiles • Serves clusters to other components
<u>Input Connections & Interfaces:</u> From which components it receives input	<ul style="list-style-type: none"> • Load Profiling • VPP and active Microgrid Flexibility Profiling • Cassandra DB/InfluxDB
<u>Output Connections & Interfaces:</u> To which components it sends the results	<ul style="list-style-type: none"> • Customers Segmentation • Virtual Power Plant Generation, Modeling & Forecasting • Cassandra DB/InfluxDB • VPP and DR Services Optimization • DSS & DR Strategies Optimization
Functional Requirements	MF02_BR01_FR01, MF02_BR01_FR02 MF02_BR01_FR03, MF02_BR01_FR04 MF02_BR01_FR05
Non-Functional Requirements	MF02_BR01_UR01_NFR01, MF02_BR01_UR01_NFR02 MF02_BR01_UR03_NFR03, MF02_BR01_UR04_NFR04

					MF02_BR01_UR05_NFR05
<u>Input Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From
Prosumers Profiles	Indicating prosumers load consumption and flexibility behavior	Float	csv	E. Abs ≥ 0 Freq = 15 min	<ul style="list-style-type: none"> Load Profiling VPP and Active Microgrid Flexibility Profiling
KPIs	DR related indicators	String	csv	-	Cassandra DB/Influx DB
<u>Output Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Sent To
Prosumers Clusters	Groups of prosumers with related characteristics regarding load consumption	Float	csv	E. Abs ≥ 0 Freq = 15 min	<ul style="list-style-type: none"> Cassandra DB/Influx DB Customers Segmentation Virtual Power Plant Generation, Modeling & Forecasting VPP and DR Services Optimization DSS & DR Strategies Optimization
Software Requirements/Development Language			TCP/IP connectivity MQTT/AMPQ, HTTP as transport protocols		

Hardware Requirements	Cloud-based system
Communications	<p>Input data should be provided either by csv files or by API requests.</p> <p>Output data shall be offered by API requests or by a context brokering service.</p>
Status of the development of the component	On early stage development (5%-10%)

Name of New Component/Service:	Customers Segmentation
Type:	Component
Functionality:	Big data tool for clusterization of energy customers
Provided Services	<ul style="list-style-type: none"> Classifies users based on the clusters Serves classification to other components
<u>Input Connections & Interfaces:</u> From which components it receives input	<ul style="list-style-type: none"> Big Data Clustering at multiple scales Cassandra DB/InfluxDB
<u>Output Connections & Interfaces:</u> To which components it sends the results	<ul style="list-style-type: none"> Cassandra DB/InfluxDB
Functional Requirements	<p>MF02_BR01_FR01, MF02_BR01_FR02</p> <p>MF02_BR02_FR01, MF02_BR02_FR02</p> <p>MF02_BR02_FR03, MF02_BR02_FR04</p> <p>MF02_BR02_FR05</p>
Non-Functional Requirements	<p>MF02_BR02_UR09_NFR01, MF02_BR01_NFR01</p> <p>MF02_BR01_NFR02, MF02_BR02_UR09_NFR01</p>
<u>Input Parameters</u>	

Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From
<i>Clusters of prosumers</i>	<i>Groups of prosumers with same features</i>	<i>Float</i>	<i>csv</i>	<i>E. Abs ≥ 0 Freq = 15 min</i>	<i>Big Data Clustering at Multiple Scales</i>
<i>KPIs</i>	<i>Behavioral Indicators</i>	<i>String</i>	<i>csv</i>	<i>-</i>	<i>Cassandra DB/InfluxDB</i>
<u>Output Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Sent To
<i>Segments of prosumers</i>	<i>Sub-groups of prosumers with specific attributes values</i>	<i>Float</i>	<i>csv</i>	<i>E. Abs ≥ 0 Freq = 15 min</i>	<i>Cassandra DB/InfluxDB</i>
Software Requirements/Development Language			<i>TCP/IP connectivity</i> <i>MQTT/AMQP, HTTP as transport protocols</i>		
Hardware Requirements			<i>Cloud-based system</i>		
Communications			<i>Input data should be provided either by csv files or by API requests.</i> <i>Output data shall be offered by API requests or by a context brokering service.</i>		
Status of the development of the component			<i>On early stage development (5%-10%)</i>		

Name of New Component/Service:	VPP and active Microgrid Flexibility Profiling
Type:	Component
Functionality:	<p>Provide the flexibility margins of the prosumers generation assets</p> <p>Calculate the overall VPP and Microgrid flexibility</p>
Provided Services	Considering the flexibility of the prosumers forming the VPP, the aim is to compute and optimize the flexibility of the micro-grid by following the requested profiles
<u>Input Connections & Interfaces:</u> From which components it receives input	<ul style="list-style-type: none"> • Virtual Power Plant Generation, Modeling and Forecasting • Load Profiling • Baseline Flexibility Estimation • Cassandra DB
<u>Output Connections & Interfaces:</u> To which components it sends the results	<ul style="list-style-type: none"> • VPP and DR Services Optimization Engine • DSS & DR Strategies Optimization • Big Data Clustering at multiple scales
Functional Requirements	<p>MF02_BR02_UR08_FR01, MF02_BR02_UR02_FR02</p> <p>MF02_BR02_UR08_FR03, MF02_BR02_UR04_FR04</p> <p>MF02_BR02_UR05_FR05, MF02_BR02_FR06</p>
Non-Functional Requirements	-
<u>Input Parameters</u>	

Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From
<i>Optimal coalition prosumers aggregated in VPP</i>	<i>prosumer Id</i>	<i>String</i>	<i>CSV/JSON</i>		<i>Virtual Power Plants Generation Modeling & Forecasting</i>
<i>Prosumers baseline load calculation</i>	<i>The estimated consumption and production flexibilities of the prosumer</i>	<i>Float</i>	<i>CSV/JSON</i>	<i>Greater than 0, every 15 min</i>	<i>Baseline Flexibility Estimation</i>
<i>Historical data for consumption/pro-duction profiles</i>					<i>Cassandra DB</i>
<u>Output Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Sent To
<i>Flexibility margins of the prosumers aggregated in VPP</i>	<i>Array of estimated flexibility values</i>	<i>Float</i>	<i>CSV/JSON</i>	<i>Greater than 0, every 15 min</i>	<ul style="list-style-type: none"> • <i>Big Data Clustering at Multiple Scales</i> • <i>VPP and DR Services Optimization Engine</i> • <i>DSS & DR Strategies Optimization</i>
Software Requirements/Development Language			<i>The software requirement is C++ libraries and runtime environment. C++ will be used as programming language.</i>		
Hardware Requirements			<i>A Windows/Linux based PC with administrator right and credentials.</i>		

	<i>In case it needs any special sensor that is included in the sensor specification, it can be included also here as a reference.</i>
Communications	<i>Ethernet or WiFi Connectivity</i>
Status of the development of the component	-

Name of New Component/Service:	VPP and DR Services Optimization Engine
Type:	Component
Functionality:	<i>Generic optimization capability for demand response services (e.g. take into account the load distribution (demand), the supply of energy (generation), the resources associations, customers classification and financial KPIs, so as to improve the DR strategies in VPP level)</i>
Provided Services	<i>DR Optimization and decision support system</i>
<u>Input Connections & Interfaces:</u> From which components it receives input	<ul style="list-style-type: none"> • <i>Virtual Power Plants Generation, Modelling & Forecasting</i> • <i>Baseline Flexibility Estimation</i> • <i>Electricity consumption/Production Forecasting</i> • <i>PV/RES Degradation & Trend Analysis</i> • <i>VPP and active Microgrid Flexibility Profiling</i> • <i>Cassandra DB</i>
<u>Output Connections & Interfaces:</u> To which components it sends the results	<ul style="list-style-type: none"> • <i>DSS & DR Strategies Optimization</i>
<u>Functional Requirements</u>	<i>MF02_BR04_UR01_FR01, MF02_BR04_FR02</i>

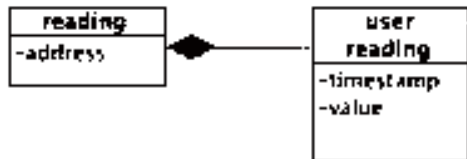
			MF02_BR04_FR03, MF02_BR04_FR04 MF02_BR04_FR05, MF02_BR04_FR06 MF02_BR04_FR07, MF02_BR04_FR08 MF02_BR04_FR09, MF02_BR04_FR010		
<u>Non-Functional Requirements</u>			-		
<u>Input Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From
Optimal coalitions information	Optimiation objective function and constraints	Array of floats	XML,JSON or CSV	Greater than 0, every 15 min	Virtual Power Plants Generation, Modelling & Forecasting
Prosumer Baseline flexibility		Prosumer ID (int)-DR programme ID (int)- Starttimestamp (string)- endtimestamp (string)-Energy consumption flexibility array (float[])- monetary cost during starttime and endtime (float []), comfort cost during starttime and end time (float[])	XML,JSON or CSV	Greater than 0, every 15 min	Baseline Flexibility Estimation

Forecast data		Device ID (int)- timestamp (string) –value (float)	XML,JSON or CSV	Greater than 0, every 15 min	Electricity consumption/Production Forecasting
KPI factors and constraints			XML,JSON or CSV	Greater than 0, every 15 min	Cassandra DB
Short-term and long-term forecasted data for PV/RES generation					PV/RES Degradation & Trend Analysis
VPPs' and Microgrid's flexibility profiles					VPP and active Microgrid Flexibility Profiling
<u>Output Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Sent To
Optimal DR scheduling information		Device ID (int) – DR programme ID (int) –start time (string) – end time (string) – array of values (float[])	XML,JSON or CSV	Greater than 0, every 15 min	DSS & DR Strategies Optimization
Set of feasible solutions for stakeholders and proposed DR scheduling		Device ID (int) – DR programme ID (int) –start time (string) – end time	XML,JSON or CSV	Greater than 0, every 15 min	

		(string) – array of values (float[])			
Software Requirements/Development Language			The software requirement is C++ libraries and runtime environment. C++ will be used as programming language.		
Hardware Requirements			<p>A Windows/Linux based PC with administrator right and credentials.</p> <p>In case it needs any special sensor that is included in the sensor specification, it can be included also here as a reference.</p>		
Communications			Ethernet or WiFi Connectivity		
Status of the development of the component			To be developed from scratch		

Name of New Component/Service:	Distributed Ledger
Type:	Component
Functionality:	Blockchain distributed ledger, used as secure storage and enabling the execution of smart contracts
Provided Services	Storage
<u>Input Connections & Interfaces:</u> From which components it receives input	<ul style="list-style-type: none"> Field Data Aggregation REST API
<u>Output Connections & Interfaces:</u> To which components it sends the results	This component will be required for the near realtime financial settlement and DR verification

				Rest API	
Functional Requirements				MF03_BR01_UR01_FR01, MF03_BR01_UR02_FR02 MF03_BR01_UR03_FR03, MF03_BR01_UR04_FR04	
Non-Functional Requirements				MF03_BR01_UR02_NFR01, MF03_BR01_UR03_NFR02 MF03_BR01_UR04_NFR03	
<u>Input Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From
address	the identifier associated to each of the distributed ledger participant	String	Alphanumeric	About 10 minutes	Field Data Aggregation
timestamp	Timestamp of the reading	Number	uint		
value	Value read from the meter	Number	int		
<u>Output Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Sent To

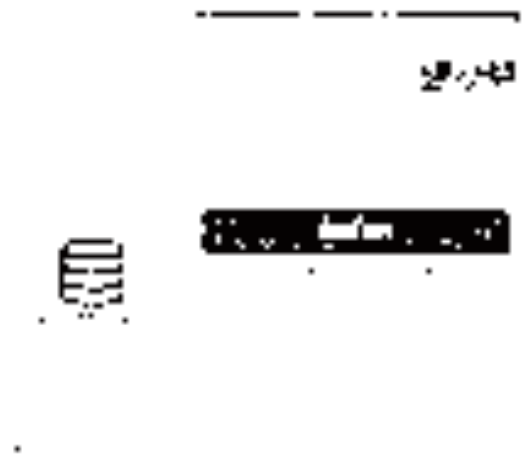
<i>Same as above (it will be a storage component, so it is expected to provide the same data entered)</i>	
Software Requirements/Development Language	<i>The smart contracts used will be written using the solidity language, libraries such as web3js will be used to interact with the contract, the network infrastructure is based on clients written in Rust programming language</i>
Hardware Requirements	<i>the component runs on a blockchain infrastructure</i>
Communications	<i>http; rpc;</i>
Status of the development of the component	<i>To be developed</i>
High-level Class Diagram	 <pre> classDiagram class reading { -address } class "user reading" { -timestamp -value } reading "1" *-- "1" user_reading : composition </pre>

Name of New Component/Service:	<i>Blockchain-driven control for LV networks (flexibility management)</i>
Type:	<i>Component</i>
Functionality:	<i>Provide a set of smart contracts modelling and evaluating the production/consumption profiles with respect to the flexibility orders and DR signals</i>

Provided Services			<i>Flexibility tracking and control;</i> <i>Flexibility request disaggregation;</i>		
<u>Input Connections & Interfaces:</u> From which components it receives input			<ul style="list-style-type: none">• <i>Distributed Ledger</i>• <i>Electricity Consumption/Production Forecasting</i>• <i>Baseline Flexibility Estimation</i>• <i>DSS & DR Strategies Optimization</i> REST API: <host>:<port>/edream-flexibility-market		
<u>Output Connections & Interfaces:</u> To which components it sends the results			<ul style="list-style-type: none">• <i>Closed Loop DR Verification Engine</i>• <i>DSS & DR Strategies Optimization</i>		
Functional Requirements			MF03_BR02_UR01_FR01, MF03_BR02_UR02_FR02 MF03_BR02_UR03_FR03, MF03_BR02_UR04_FR04 MF03_BR03_UR01_FR05, MF03_BR03_UR02_FR06 MF03_BR03_UR03_FR07, MF03_BR03_UR04_FR08		
Non-Functional Requirements					
<u>Input Parameters</u>					
Attribute /Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From
<i>Monitored Values</i>	An array containing all the monitored values for a period	JSON	[{ "timestamp": 1537803600000, "value": 908.483, "deviceMeasurementId" : "18bb6e98-8429-4162- 8236-cc4f231bb2a4",	Range – depending on the monitored device Frequency: every5 min	<i>Distributed Ledger</i>

			<pre> "deviceId": "4836bee7- bc42-48a9-9e4e- aa1ecc68e6d8", "property": "Power Consumption" }, { "timestamp": 1537341600000, "value": 910.54, "deviceMeasurementId" : "18bb6e98-8429-4162- 8236-cc4f231bb2a4", "deviceId": "4836bee7- bc42-48a9-9e4e- aa1ecc68e6d8", "property": "Power Consumption" }, ...] </pre>		
<i>Registered Energy Assets</i>	The energy asset registered as token in the chain	TOKEN	-	Range – Frequency:	
<i>Baseline & Flexibility values</i>		JSON	<pre> [{ "timestamp": 1537803600000, "value": 905.5, "deviceMeasurementId": "12bb6e98-8429-4162-8236- cc4f231bb2a4", "deviceId": "4836bee7- bc42-48a9-9e4e- aa1ecc68e6d8", "granularity": "DAYAHEAD", "property": "Baseline Flexibility" },...] </pre>	Range – depending on the forecasted property Frequency: every 30 min, 1 hour	<i>Baseline Flexibility Estimation</i>
<i>Predicted Values</i>	An array containing the device and the measurement the	JSON	<pre> { "profile": [{"value": 57167.566, "timestamp": "2018-09-08T00:00:00" }, </pre>	Range – depending on the forecasted property Frequency: every 30 min, 1 hour	<i>Electricity Production/Consumption Forecasting</i>

	forecast is referring to, together with the forecasted value		<pre> {"value": 57174.233, "timestamp": "2018-09-08T01:00:00" }, ...], "entityDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "profileGranularityMinutes": 60 , "predictionGranularity": "DAYAHEAD", "property": "ENERGY CONSUMPTION" } </pre>		
<i>Flexibility offers/orders</i>					<i>DSS & DR Strategies Optimization</i>
<u>Output Parameters</u>					
Attribute /Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Sent To
<i>Smart Contracts evaluation of actual profile delivered</i>	Evaluate the energy flexibility delivered	-	Evaluation inside the contract. The result is the updated state of the contract	-	<i>Closed Loop DR Verification Engine</i> <i>DSS & DR Strategies Optimization</i>
Software Requirements/Development Language				<ul style="list-style-type: none"> • 64 bit Linux / Windows OS • Internet connection for package management • Ethereum – Parity , Solidity 	


	<ul style="list-style-type: none"> NodeJS, MySql
Hardware Requirements	<ul style="list-style-type: none"> Intel/AMD processor with at least 4 cores and 2.5GHz base frequency 8 GB RAM HDD/SSD with at least 100GB free space Internet connection for package management
Communications	REST API, JSON RPC
Status of the development of the component	Initial component prototype implemented
High-level Class Diagram	 <pre> sequenceDiagram participant User participant Process User->>Process: activate Process Process->>User: deactivate Process </pre>

Name of New Component/Service:	Secured Blockchain-driven Energy Market
Type:	Component

Functionality:			<i>Provide a secured Energy Marketplace, where each actor can register bid/offer energy market actions. The clearing price and the settlement will be ensured in a secure way using self-enforced smart contracts.</i>		
Provided Services			<i>Peer to peer energy trading;</i> <i>Energy bids/offers matching</i>		
<u>Input Connections & Interfaces: From which components it receives input</u>			<ul style="list-style-type: none">• <i>Distributed Ledger</i>• <i>Electricity Consumption/Production Forecasting</i>• <i>Local/Remote HMIs</i>		
<u>Output Connections & Interfaces: To which components it sends the results</u>			<ul style="list-style-type: none">• <i>Closed Loop DR Verification Engine</i>• <i>Local/Remote HMIs</i> REST API: <host>:<port>/edream-energy-market/		
Functional Requirements			MF03_BR02_UR01_FR01, MF03_BR02_UR02_FR02 MF03_BR02_UR03_FR03, MF03_BR02_UR04_FR04 MF03_BR03_UR01_FR05, MF03_BR03_UR02_FR06 MF03_BR03_UR03_FR07, MF03_BR03_UR04_FR08		
Non-Functional Requirements			MF03_BR04_UR01_NFR01		
<u>Input Parameters</u>					
Attribute /Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From
<i>Registered</i>	The energy asset registered as	TOKEN	-	Range – Frequency:	<i>Distributed Ledger</i>

<i>Energy Assets</i>	token in the chain				
<i>Bid/Offer Actions</i>	Place orders on the market contract	JSON	<pre>{ "orderSide": "BID", "prosumerAddress": "0xAA21803000499f1b58C67F4DA7083AFA2ee37090", "timestamp": "123453123", "tokenId": 10, "metadata": { "startTimeToken": 0, "endTimeToken": 1, "energyType": "GREEN", "producer": "0xAA21803000499f1b58C67F4DA7083AFA2ee37090" } "quantity": 12321 "price": 35, }</pre>	Range: Frequency: 30-60 min	<i>HMI</i>
<i>Predicted Values</i>	An array containing the device and the measurement the forecast is referring to, together with the forecasted value	JSON	<pre>{ "profile": [{ "value": 57167.566, "timestamp": "2018-09-08T00:00:00" }, { "value": 57174.233, "timestamp": "2018-09-08T01:00:00" }, ...], "entityDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "profileGranularityMinutes": 60 , "predictionGranularity": "DAYAHEAD", "property": "ENERGY CONSUMPTION" }</pre>	Range – depending on the forecasted property Frequency: every 30 min, 1 hour	<i>Electricity Production/Consumption Forecasting</i>
<u>Output Parameters</u>					


Attribute /Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Sent To
Clearing Price	The clearing price obtained at the end of the market session	JSON	{ "marketAddress": "0x32fc0bca5134a5cd7469c2fe1f9968f44d1949a8", "marketPrice" : [14, ...] }	Range – Frequency:	Closed Loop DR Verification Engine HMI
Matched energy production and demand	The matching between the sell and buy orders	JSON	[{ "id": "0x309911d2e38ff1649f7b6d39ef0e3fb33c87ae5d4fda6fa065ec9821c77fe2e3", "buyOrderId": "0x1c817f45dd2349c07b100ac0c9204d1652cea73006c0b000c9c39c4c40710d78", "sellOrderId": "0x95c8b70fc368ad3a3d7544715461d189616114cca00cabb97d2665a84e5d8b54", "prosumerBuyingAddress":"0x04fb94f5e2555d1e860462060337aa62ec6e919d", "prosumerSellingAddress":"0xAA21803000499f1b58C67F4DA7083AFA2ee37090", "timestamp":"123453123", "tokenId":10, "quantity":12321 "price":35, }, ...]	Range – Frequency:	
Software Requirements/Development Language			<ul style="list-style-type: none">• 64 bit Linux / Windows OS• Internet connection for package management• Ethereum – Parity , Solidity• NodeJS, MySql		
Hardware Requirements			<ul style="list-style-type: none">• Intel/AMD processor with at least 4 cores and 2.5GHz base frequency		

	<ul style="list-style-type: none"> • 8 GB RAM • HDD/SSD with at least 100GB free space • Internet connection for package management
Communications	REST API, JSON RPC
Status of the development of the component	Initial component prototype implemented
High-level Class Diagram	

Name of New Component/Service:	<i>Closed loop DR Verification Engine</i>
Type:	<i>Component</i>

Functionality:					
<p><i>This component will be used to validate transactions previously agreed on the marketplace.</i></p> <p><i>Provide a closed-loop verification engine that aims to assess the flexibility actually activated by prosumers and aggregators and set the associated financial settlement. New methods of achieving energy transactions validation and consensus.</i></p>					
Provided Services					
Verification, Financial settlement (incentives).					
<u>Input Connections & Interfaces:</u> From which components it receives input					
<ul style="list-style-type: none"> Distributed Ledger Blockchain-driven control for LV networks Blockchain-driven Energy Market Cassandra DB <p>Rest API</p>					
<u>Output Connections & Interfaces:</u> To which components it sends the results					
<ul style="list-style-type: none"> Distributed Ledger Local/Remote HMIs <p>Rest API</p>					
Functional Requirements					
MF03_BR02_UR01_FR01, MF03_BR02_UR02_FR02 MF03_BR02_UR03_FR03					
Non-Functional Requirements					
-					
<u>Input Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From

<i>forecast</i>	Forecasted value at target timestamp	number	int	Each of the monitored timestamp	<i>Cassandra DB</i>
<i>request</i>	Requested value at target timestamp	number	int		<i>Blockchain-driven control for LV networks (flexibility management)</i>
<i>reading</i>	measured value at target timestamp	number	int		<i>Distributed Ledger</i>
<i>range</i>	Tolerance range for the request	number	uint	Fixed for the contract duration	<i>Distributed Ledger</i>
<i>multiplier</i>	incentive/penalty multiplier	number	float		
<i>Clearing price, Matched energy production and demand</i>					
<u>Output Parameters</u>					
<i>Attribute/Parameter</i>	<i>Short Description</i>	<i>Data Type</i>	<i>Data Format</i>	<i>Value Range & Frequency</i>	<i>Data Sent To</i>
<i>incentive/penalty</i>	Incentive or penalty to be applied for each timestamp	number	int	Each of the monitored timestamp	<i>Distributed Ledger Local/Remote HMIs</i>
<i>Software Requirements/Development Language</i>			<i>Solidity (for the smart contract development)</i>		
<i>Hardware Requirements</i>			<i>the component runs on a blockchain infrastructure</i>		

Communications	<i>http; rpc</i>
Status of the development of the component	<i>to be developed</i>
High-level Class Diagram	 <pre> classDiagram class Request { request sensor multiplier } class Decision { forecast request reading uncertainty } Request --> Decision </pre>

Name of New Component/Service:	<i>DSS (Decision Support System) & DR Strategies Optimization</i>
Type:	<i>Component</i>
Functionality:	<i>An interface for analyzing and preparing energy trend and flow information for display by a HMI. Interface for inputting variables for optimizing DR services.</i>
Provided Services	<i>Provide a set of feasible solutions to stakeholders</i>
<u>Input Connections & Interfaces:</u> From which components it receives input	<ul style="list-style-type: none"> • <i>VPP and DR Services Optimization Engine</i> • <i>PV/RES Degradation and Trend Analysis</i> • <i>Baseline Flexibility Estimation</i> • <i>Big Data Clustering at multiple scales</i> • <i>VPP and active Microgrid Flexibility Profiling</i> • <i>Cassandra DB</i>

<u>Output Connections & Interfaces:</u> To which components it sends the results			<ul style="list-style-type: none">End-user’s web interface		
Functional Requirements			MF02_BR04_UR01_FR01, MF02_BR04_FR02 MF02_BR04_FR03, MF02_BR04_FR04		
Non-Functional Requirements			-		
<u>Input Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From
DR programme	ID of DR programme		XML, CSV or JSON	Greater than 0, every 15 min	Cassandra DB
Historical data, KPIs, Constraints					Cassandra DB
Short-term and long-term generation forecasted data					PV/RES Degradation & Trend Analysis
Optimal DR schedule					VPP & DR Services Optimization Engine
Prosumers’ clusters					Big Data Clustering at multiple scales
VPPs’ & Microgrids’ flexibility profiles					VPP and active Microgrid flexibility profiling
<u>Output Parameters</u>					

Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Sent To
<i>Set of feasible solutions</i>	<i>Optimized DR profile</i>	<i>Starttime (string) – end time (string)- values of optimized DR profile (float)</i>	<i>XML,CSV,JSON</i>	<i>Greater than 0, 15 min</i>	<i>End-user's web interface</i>
Software Requirements/Development Language			<i>The software requirement is C++ libraries and runtime environment. C++ will be used as programming language.</i>		
Hardware Requirements			<i>A Windows/Linux based PC with administrator right and credentials.</i> <i>In case it needs any special sensor that is included in the sensor specification, it can be included also here as a reference.</i>		
Communications			<i>Ethernet or WiFi Connectivity</i>		
Status of the development of the component			<i>To be developed from scratch</i>		

Name of New Component/Service:	DR Aerial Survey Toolkit
Type:	<i>Component</i>
Functionality:	<i>Estimate the demand response potential over a wide area of building assets based on the energy demand profile assessment and the overall energy performance of the buildings through optical, thermal and LIDAR images</i>

Provided Services	Provide data for energy assessment of buildings and specific zones				
<u>Input Connections & Interfaces:</u> From which components it receives input	<ul style="list-style-type: none">• Drone camera mount• LiDAR Ethernet port• Skyport API• Multi-building DR characterization• Cassandra DB				
<u>Output Connections & Interfaces:</u> To which components it sends the results	<ul style="list-style-type: none">• End-user’s web interface				
Functional Requirements	MF01_BR01_UR01_FR01, MF01_BR01_UR01_FR02 MF01_BR01_FR03, MF01_BR01_FR04 MF01_BR01_FR05, MF01_BR01_FR06 MF01_BR01_FR07				
Non-Functional Requirements	-				
<u>Input Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From
Aggregated and processed data from aerial surveys					<ul style="list-style-type: none">• Cameras• Multi-building DR Characterization
Weather data, Smart meters’ data					<ul style="list-style-type: none">• Cassandra DB
<u>Output Parameters</u>					

Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Sent To
<i>Estimated DR potential for groups of assets or buildings</i>					<ul style="list-style-type: none"> End-user's web interface
Software Requirements/Development Language			Velodyne VeloView LiDAR processing software. Pix4D aerial survey software		
Hardware Requirements			Quadcopter Drone: We are using a DJI Matrice 210. Thermal imaging camera: DJI Zenmuse XT2 with 19mm lens, 30Hz refresh rate and 640x512 pixels resolution. LiDAR: Velodyne VLP 16 ('Puck')		
Communications			TCP/IP for Wireless Communications		
Status of the development of the component			Partially developed		

Name of New Component/Service:	Graph-based Analytics
Type:	Component
Functionality:	A data and graph analytics engine enabling hypothesis testing for the identification of the optimal parameters for the DR strategies.
Provided Services	Visual clustering and Multi-objective analysis
<u>Input Connections & Interfaces:</u> From which components it receives input	<ul style="list-style-type: none"> Cassandra DB

<u>Output Connections & Interfaces:</u> To which components it sends the results		<ul style="list-style-type: none">• DSS & DR Strategies Optimization• HMI• Multi-purpose Dashboards		
Functional Requirements		MF01_BR05_FR01, MF01_BR05_FR02 MF01_BR05_FR03, MF01_BR05_FR04 MF01_BR05_FR05, MF01_BR05_FR06 MF01_BR05_FR07		
Non-Functional Requirements		-		
<u>Input Parameters</u>				
Attribute/Para-meter Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From
Spatial layout of the grid	Float	XML, JSON	-	Cassandra DB
Parameters for visualization (e.g. time resolution)	Float	XML, JSON	-	
Data models related to segmentation metrics - KPIs (e.g. profits/loses, congestion improvement achieved etc.)	String	XML, JSON	-	
DR activation signals	String	XML, JSON	-	HMI
Input settings/parameters by the HMI	Float	XML, JSON	-	
Parameters of DR signals and mapping to the portfolio	Float	XML, JSON	-	

<u>Output Parameters</u>				
<i>Attribute/Parameter</i> <i>Short Description</i>	<i>Data Type</i>	<i>Data Format</i>	<i>Value Range & Frequency</i>	<i>Data Sent To</i>
<i>Data model for optimal selection selection of the portfolio in spatial-temporal and operation views</i>	<i>Float</i>	<i>XML, JSON</i>	<i>-</i>	<i>HMI</i>
<i>Specific characteristics of the selected portfolio (spatial-temporal-operational) along with relevant KPIs</i>	<i>String</i>	<i>XML, JSON</i>		
<i>Improved DR activation signals</i>	<i>Float</i>	<i>XML, JSON</i>		
<i>Software Requirements/Development Language</i>		<i>The software requirement is C++ libraries and runtime environment. C++ will be used as programming language.</i>		
<i>Hardware Requirements</i>		<i>A Windows/Linux based PC with administrator right and credentials.</i>		
<i>Communications</i>		<i>Ethernet or WiFi Connectivity</i> <i>Web Interface</i>		
<i>Status of the development of the component</i>		<i>On early development stage</i>		


Name of New Component/Service:	Forecasting Tool
Type:	<i>Component</i>
Functionality:	<i>Forecasting Tool with interactive visualization framework enabling the aggregator to be informed with a standard pre-defined frequency rate about</i>

	the prosumers consumption/production forecasted values.				
Provided Services	Provide options to end-users, in order to select different parameters for the forecasting algorithm				
Input Connections & Interfaces: From which components it receives input	<ul style="list-style-type: none">Electricity consumption/Production ForecastingPV/RES Degradation & Trend AnalysisCassandra DB				
Output Connections & Interfaces: To which components it sends the results	<ul style="list-style-type: none">End-user’s web interface				
Functional Requirements	MF01_BR02_FR01, MF01_BR02_FR02 MF01_BR02_FR03, MF01_BR02_FR04				
Non-Functional Requirements	-				
Input Parameters					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From
Forecasted consumption/production prosumers data	Near real time forecasted values for intraday and day-ahead planning	Float	XML,JSON	Every 10 min (5min or 1 hour are also possible)	Electricity consumption/Production Forecasting
Short-term and long-term generation forecasting					PV/RES Degradation & Trend Analysis
Weather data, Smart meters’ data					Cassandra DB

<u>Output Parameters</u>					
<i>Attribute/Para-meter</i>	<i>Short Description</i>	<i>Data Type</i>	<i>Data Format</i>	<i>Value Range & Frequency</i>	<i>Data Sent To</i>
<i>Visualizations of the input data</i>					
<i>Software Requirements/Development Language</i>		<i>The software requirement is python libraries and runtime environment. Python will be used as programming language.</i>			
<i>Hardware Requirements</i>		<i>A Windows/Linux based PC with administrator right and credentials.</i> <i>Web Server</i>			
<i>Communications</i>		<i>Ethernet or WiFi Connectivity</i> <i>Web Interface</i>			
<i>Status of the development of the component</i>		<i>To be developed</i>			

<i>Name of New Component/Service:</i>	<i>HMI</i>
<i>Type:</i>	<i>Component</i>
<i>Functionality:</i>	<i>An interactive multi-level and multi-factor visualization framework enabling the end-user to interact with the components of the core platform.</i>
<i><u>Input Connections & Interfaces:</u> From which components it receives input</i>	<ul style="list-style-type: none"> <i>Core platform's components</i> <i>End-user web interface</i>

<u>Output Connections & Interfaces:</u> To which components it sends the results				<ul style="list-style-type: none">• Core platform’s components• End-user web interface	
Functional Requirements				MF02_BR05_BR10_MF03_BR05_FR01 MF02_BR05_BR10_MF03_BR05_FR02 MF02_BR05_BR10_MF03_BR05_FR03 MF03_BR05_UR01_FR04	
Non-Functional Requirements				MF02_BR05_BR10_MF03_BR05_NFR01 MF02_BR05_BR10_MF03_BR05_NFR02 MF02_BR05_BR10_MF03_BR05_NFR03 MF02_BR05_BR10_MF03_BR05_NFR04 MF02_BR05_BR10_MF03_BR05_NFR05 MF02_BR05_BR10_MF03_BR05_NFR06 MF02_BR05_BR10_MF03_BR05_NFR07	
<u>Input Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From
All the data models exchanged within the core platform and with the end-user web interface.					
<u>Output Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Sent To
All the data models exchanged within the core platform and with the end-user web interface.					

Software Requirements/Development Language	<i>The software requirement is C++ libraries and runtime environment. C++ will be used as programming language.</i>
Hardware Requirements	<i>A Windows/Linux based PC with administrator right and credentials.</i> <i>Web Server</i>
Communications	<i>Ethernet or WiFi Connectivity</i> <i>Web Interface</i>
Status of the development of the component	<i>To be adjusted and evolved</i>
High-level Class Diagram	
	

This report has presented the updated version architectural design of the eDREAM system along with the respective functional and technical specifications and interfaces. The updated have emerged from the work done during the technical WPs 3, 4, 5 and 6. This can be considered as a living document, because as the programming and implementation activities progress, further updates may arise. The fully detailed description of the platform will be presented in the final version that is due to be submitted in month 30.

- [1]. Agile Modeling, UML2 Use Case Diagrams, available under <http://www.agilemodeling.com/artifacts/useCaseDiagram.htm> (state on September 2013).
- [2]. Carnegie Mellon community, UML Use Case Diagrams: Tips and FAQ, available under <http://www.andrew.cmu.edu/course/90-754/umlucdfaq.html> (state on September 2013).
- [3]. StarUML ,Modelling with Class Diagram, available under [http://staruml.sourceforge.net/docs/user-guide\(en\)/ch05_2.htm](http://staruml.sourceforge.net/docs/user-guide(en)/ch05_2.htm)

This section introduces the eDREAM functional & non-functional requirements.

Functional Requirements

Component: Electricity Consumption/Production Forecasting – Prosumer Level	
Functional Requirement ID	Description
MF01_BR02_UR01_FR01	Process Raw Monitored Data
MF01_BR02_UR02_FR02	Prosumer flexibility forecast for specific time interval
MF01_BR02_UR03_FR03	Prosumer energy consumption / production forecast for specific time interval
MF01_BR02_UR04_FR04	Get Weather Forecast Data from external service
MF01_BR02_UR05_FR05	Store prosumer consumption / production forecast view
Component: Electricity Consumption/Production Forecasting – Grid Level	
Functional Requirement ID	Description
MF02_BR07_UR01_FR06	Grid (group of prosumers) flexibility forecasting for specific time interval
MF02_BR07_UR02_FR07	Grid (group of prosumers) energy consumption / production forecast for specific time interval
MF02_BR07_UR03_FR08	Store grid consumption / production forecast view

Component: Virtual Power Plants Generation Modelling and Forecasting	
Functional Requirement ID	Description
MF02_BR03_UR01_FR01	Aggregate and Optimize the production profiles
MF02_BR03_UR02_FR02	Create the optimal coalitions of energy generation sources (VPP)
MF02_BR03_UR03_FR03	Provide list of prosumers coalized in a VPP

Component: PV/RES Degradation and Trend Analysis	
Functional Requirement ID	Description
MF01_BR04_UR01_FR01	Obtain data for field devices' physical parameters and constraints

MF01_BR04_UR02_FR02	Receive forecasted data for weather conditions from Weather APIs
MF01_BR04_FR03	Receive historical data for measurements related to generation assets from Decentralized Repository
MF01_BR04_FR04	Receive historical data for weather conditions

Component: Baseline Flexibility Estimation	
Functional Requirement ID	Description
MF01_BR03_UR01_FR01	Receive historical data for prosumer's baseline load calculations
MF01_BR03_FR02	Receive physical parameters and constraints of the installed field devices
MF01_BR03_FR03	Store prosumer's baseline load flexibility
MF02_BR09_UR01_FR04	Assess flexibility availability of individual prosumers
MF02_BR09_UR02_FR05	Assess flexibility values from Blockchain-driven control for LV networks and Closed loop DR verification engine

Component: Multi-building DR characterization through thermal, optical and LIDAR information fusion	
Functional Requirement ID	Description
MF01_BR01_UR01_FR01	Receive LIDAR point cloud files
MF01_BR01_UR01_FR02	Obtain visible and infrared spectrum images
MF01_BR01_FR03	Receive historical data for weather conditions
MF01_BR01_FR04	Receive actual smart meter monitoring data
MF01_BR01_FR05	Process collected digital images
MF01_BR01_FR06	Perform data analysis on processed data from digital images
MF01_BR01_FR07	Store analyzed data in the Decentralized Repository (Database – Cassandra DB)

Component: Big Data Analytics Engine Tool	
Functional Requirement ID	Description
MF02_BR01_FR01	Near real-time data processing

MF02_BR01_FR02	Batch Processing
MF02_BR01_FR03	Data Harmonization
MF02_BR01_FR04	Data Preprocessing & Postprocessing
MF02_BR01_FR05	Data Gathering
MF02_BR01_FR06	Data Discovery
MF02_BR01_FR07	Data Filtering

Component: Load Profiling & Disaggregation	
Functional Requirement ID	Description
MF02_BR02_UR02_FR01	Receive historical data for prosumers' energy consumption
MF02_BR02_FR03	Detect energy consumption patterns
MF02_BR02_FR04	Store prosumers' load profiles
MF02_BR02_FR05	Produce the energy consumption profile of the prosumer's high-consumption electrical appliances
MF02_BR02_FR06	Store the produced data in the Repository

Component: Big Data Clustering at Multiple Scales	
Functional Requirement ID	Description
MF02_BR01_FR01	Receive prosumers' load profiles
MF02_BR01_FR02	Receive prosumers' generation assets profiles
MF02_BR01_FR03	Receive historical data concerning prosumers' behavior and responsiveness to DR schemes
MF02_BR01_FR04	Expose clusters to other architectural components
MF02_BR01_FR05	Store produced clusters of prosumers

Component: Customers Segmentation	
Functional Requirement ID	Description
MF02_BR02_FR01	Get clusters of prosumers

MF02_BR02_FR02	Receive specific attributes values for the requested segments of previous clusterizations
MF02_BR02_FR03	Receive related KPIs from the Repository
MF02_BR02_FR04	Detect patterns according to the received attributes values
MF02_BR02_FR05	Store segments of prosumers

Component: VPP and active Microgrid flexibility profiling	
Functional Requirement ID	Description
MF02_BR02_UR08_FR01	Receive prosumers' load profiles
MF02_BR02_UR02_FR02	Receive historical data for generation of prosumers' energy assets
MF02_BR02_UR08_FR03	Obtain physical parameters and constraints of field devices in order to calculate the flexibility margins of assets
MF02_BR02_UR04_FR04	Process EV (Battery SoC, residual autonomy etc.)
MF02_BR02_UR05_FR05	Process EVSE data (power, voltage, current etc.)
MF02_BR02_FR06	Identify generation patterns

Component: VPP and DR Services Optimization Engine	
Functional Requirement ID	Description
MF02_BR04_UR01_FR01	Receive analyzed data for the efficacy of the implemented DR strategies from Graph-based Analytics
MF02_BR04_FR02	Receive consumption/production forecasted data
MF02_BR04_FR03	Obtain energy prices from the Decentralized Repository
MF02_BR04_FR04	Receive economic, conform, environmental and business KPIs from the Decentralized Repository
MF02_BR04_FR05	Get potential incentives for the final users from the HMI
MF02_BR04_FR06	Obtain the load profiles of the registered prosumers
MF02_BR04_FR07	Obtain the generation profiles of the participating flexible resources
MF02_BR04_FR08	Identify patterns among the input data
MF02_BR04_FR09	Calculate optimal set points for generators and load curtailment
MF02_BR04_FR010	Produce optimal DR scheduling

Component: Distributed Ledger	
Functional Requirement ID	Description
MF03_BR01_UR01_FR01	Register Energy as Digital Assets
MF03_BR01_UR02_FR02	Transfer Energy as Digital Assets
MF03_BR01_UR03_FR03	Control and Permission Enforcement
MF03_BR01_UR04_FR04	Distributed Ledger Scalability

Component: Blockchain-driven control for LV networks (flexibility management)	
Functional Requirement ID	Description
MF03_BR02_UR01_FR01	Detection of grid level congestion points
MF03_BR02_UR02_FR02	Flexibility requests to aggregators
MF03_BR02_UR03_FR03	Selection of flexibility offers from aggregators
MF03_BR02_UR04_FR04	Track and control the flexibility delivery of aggregators
MF03_BR03_UR01_FR05	Communicate flexibility requests to prosumers
MF03_BR03_UR02_FR06	Communicate flexibility availability of prosumer to aggregators
MF03_BR03_UR03_FR07	Selection of prosumers from portfolio to meet a specific aggregated flexibility
MF03_BR03_UR04_FR08	Track and control the flexibility delivery of prosumers

Component: Secured Blockchain-driven Energy Market	
Functional Requirement ID	Description
MF03_BR04_UR01_FR01	Energy transactions security
MF03_BR04_UR02_FR02	Registration and Validation of Prosumer
MF03_BR04_UR03_FR03	Publish Bid/Offer actions by Prosumer
MF03_BR04_UR04_FR04	Energy Bids/ Offers matching and Clearing Price Computation

Component: Closed loop DR Verification Engine	
Functional Requirement ID	Description
MF03_BR02_UR01_FR01	Validate DR Flexibility actually provided (at prosumer level)
MF03_BR02_UR02_FR02	Mining new blocks of energy transactions
MF03_BR02_UR03_FR03	Settle Accounts according to DR Flexibility Validation

Component: Graph-based Analytics	
Functional Requirement ID	Description
MF01_BR05_FR01	Receive data for the spatial layout of the grid
MF01_BR05_FR02	Receive flexibility data (actual & forecasted) per each prosumer
MF01_BR05_FR03	Obtain DR related KPIs (e.g. profits/loses, congestion improvement achieved etc.)
MF01_BR05_FR04	Get input settings from the Aggregator UI
MF01_BR05_FR05	Receive parameters of DR signals and mapping to the portfolio
MF01_BR05_FR06	Perform data analysis and correlation among the input parameters
MF01_BR05_FR07	Store analyzed data and identified patterns

Component: Decision Support System & DR Strategies Optimization	
Functional Requirement ID	Description
MF02_BR04_UR01_FR01	Receive analytics data for the efficiency of the currently implemented DR strategies from Graph-based Analytics
MF02_BR04_FR02	Receive optimized parameters from the VPP & DR Services Optimization for the loads to be shed and the set points of dispatchable generators
MF02_BR04_FR03	Store optimized parameter in the Decentralized Repository
MF02_BR04_FR04	Communicate with the Operator's application via web interface

Component: DR Aerial Survey Toolkit	
Functional Requirement ID	Description
MF01_BR01_FR01	Receive analyzed data from collected images from the component Multi-building DR characterization through thermal, optical and LIDAR information fusion
MF01_BR01_FR02	Receive baseline load flexibility values for the registered prosumers
MF01_BR01_FR03	Store DR estimated capability of potential customers in the Decentralized Repository
MF01_BR01_FR04	Communicate with the Operator's application via web interface

Component: Forecasting Tool	
Functional Requirement ID	Description
MF01_BR02_FR01	Receive forecasted data from the Electricity Consumption/Production Forecasting with a standard frequency rate around 10min
MF01_BR02_FR02	Receive historical data for prosumer consumption/production
MF01_BR02_FR03	Store the forecasted data to the Decentralized Repository
MF01_BR02_FR04	Communicate with the Operator's application via web interface

Component: HMIs	
Functional Requirement ID	Description
MF02_BR05_BR10_MF03_BR05_FR01	Communicate with the Operator's application via a web interface
MF02_BR05_BR10_MF03_BR05_FR02	Allow parametrization of inputs, conditions and process constraints
MF02_BR05_BR10_MF03_BR05_FR03	Communicate with components of the core platform in order to retrieve and collect data for further analysis and processing
MF03_BR05_UR01_FR04	Allow the prosumers to initialize or edit the parameters used by smart contracts for both energy and flexibility trading

Component: EVSEs and EV fleet monitoring	
Functional Requirement ID	Description
MF02_BR07_UR01_FR01	The system shall be able to process EV data (Battery State-of-Charge, residual Autonomy, minutes to Full Charge, Geolocation, Doors Car State, Engine Car State). It will be enclosed in a wrapper and sent via the MQTT protocol
MF02-BR07-UR02_FR02	The system shall be able to process EVSE data (power, voltage, current, plug status, energy consumption). It will be enclosed in a wrapper and sent via the MQTT protocol

Component: Electric meters, edge, and field device electric measures	
Functional Requirement ID	Description
FD_BR01_UR01_FR01	The system should be able to acquire real time measures from the field with adequate latency, or from existing automatic reading systems. The importance to receive and process data from field devices (directly from equipment or indirectly from others supervision systems) is relevant in micro-grid operation especially in islanding operation, when low inertia occurs. The proper time latency should be identified according for each service to provide by eDREAM platform.

Non-Functional Requirements

Component: Distributed Ledger	
Non-Functional Requirement ID	Description
MF03_BR01_UR02_NFR01	Ensure scalability with high number of energy transactions
MF03_BR01_UR03_NFR02	Store transactions in a secured and tamperproof manner
MF03_BR01_UR04_NFR03	Grant access to data only to authorized users

Component: Secured Blockchain-driven Energy Market	
Non-Functional Requirement ID	Description

MF03_BR04_UR01_NFR01	Store transactions in a secured and tamperproof manner
-----------------------------	--

Component: Big Data Clustering at Multiple Scales	
Non-Functional Requirement ID	Description
MF02_BR01_UR01_NFR01	Be able to operate in a scalable way horizontally or vertically depending on the demands related to data injection, processing and storage
MF02_BR01_UR01_NFR02	Be able to decompose the solution in different microservices, so as to run better during different processes in several machines
MF02_BR01_UR03_NFR03	Be able to document information types produced, consumed and transformed in an information model which shall include the relationships between information types
MF02_BR01_UR04_NFR04	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.).
MF02_BR01_UR05_NFR05	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.)

Component: Load Profiling_Customer Segmentation	
Non-Functional Requirement ID	Description
MF02_BR02_UR09_NFR01	Be able to operate in a scalable way horizontally or vertically depending on the demands related to data injection, processing and storage

Component: EVSEs and EV fleet monitoring	
Non-Functional Requirement ID	Description
MF02_BR07_UR01_NFR01	Connectivity and interoperability between EV and EVSE systems
MF02_BR07_UR02_NFR02	EV data and EVSE data must be collected in real time (or very close to real time).
MF02-BR07-UR04_NFR03	Data accessibility: data coming from EVSEs and the EVs should be consistent, reliable, transparent and accessible to the partners

Component: Big Data Analytics Engine Tool	
Non-Functional Requirement ID	Description
MF02_BR01_NFR01	Scalability
MF02_BR01_NFR02	Recoverability

Component: HMIs	
Non-Functional Requirement ID	Description
MF02_BR05_BR10_MF03_BR05_NFR01	The User Interface shall have a user friendly look, fully customized to the needs of different stakeholders
MF02_BR05_BR10_MF03_BR05_NFR02	The User Interface shall provide a user interface offering maps Visualization for a more concrete analysis
MF02_BR05_BR10_MF03_BR05_NFR03	The User Interface shall be able to allow an easy discoverability of the actions available
MF02_BR05_BR10_MF03_BR05_NFR04	The User Interface shall be tailored to the end user needs
MF02_BR05_BR10_MF03_BR05_NFR05	The messages provided by the system must be clear and easy to understand
MF02_BR05_BR10_MF03_BR05_NFR06	The User Interface must be simple and intuitive
MF02_BR05_BR10_MF03_BR05_NFR07	End users will have multiple interfaces to the whole system

Name of Requirement	<i><provide the name of the requirement></i>
Requirement ID	<i><provide the reference ID of the requirement></i>
Requirement Type	<i><Functional or Non-Functional></i>
Description	<i><provide a short description></i>

Name of New Component/Service:	<i><please write here the name of the architectural element e.g. Baseline Flexibility Estimation></i>
Type:	<i><Component, Software, Device etc.></i>
Functionality:	<i><please write here in free text a short description of the operation of this module/component. A list of functions and operations will be an added value.></i>
Provided Services	<i><please mention the services provided by the component></i>
<u>Input Connections & Interfaces:</u> From which components it receives input	<i><please write the components from which it receives input (input dependencies) and mention also the available connection interfaces e.g. API etc.></i>
<u>Output Connections & Interfaces:</u> To which components it sends the results	<i><please write the components to which it sends the results (output dependencies) and mention also the available interfaces e.g. API etc.></i>
Functional Requirements	<i><write the functional requirements that the module satisfies, mention the respective IDs from the relevant template></i>

Non-Functional Requirements			<write the non-functional requirements that the module satisfies, mention the respective IDs from the relevant template>		
<u>Input Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From
<please mention the input parameters. Each row corresponds to one parameter>	<mention a short description of the input parameter if necessary>	<please mention the data type of this parameter (e.g. (e.g. int, string, etc. or complex type, e.g. list, object, etc.)	<e.g. XML, JSON etc.>	<indicate measurement unit and range of values for this attribute/parameter and frequency-sample rate>	<please mention the source component or module that provides input data to this parameter>
<u>Output Parameters</u>					
Attribute/Parameter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Sent To

<i><please mention the input parameters. Each row corresponds to one parameter></i>	<i><mention a short description of the input parameter if necessary></i>	<i><please mention the data type of this parameter (e.g. (e.g. int, string, etc. or complex type, e.g. list, object, etc.)</i>	<i><e.g. XML, JSON etc.></i>	<i><indicate measurement unit and range of values for this attribute/parameter and frequency-sample rate></i>	<i><please mention the source component or module that provides input data to this parameter></i>
Software Requirements/Development Language			<i><specify any software requirements related to the architectural element, explain the Programming Language that is used during the development of the component></i>		
Hardware Requirements			<i><specify what hardware requirements are of the module, give specifications about the hardware requirements which are necessary for the best functionality of the component></i> <i>In case it needs any special sensor that is included in the sensor specification, it can be included also here as a reference.</i>		
Communications			<i><address specific communication requirements either for data input or for data output></i>		
Status of the development of the component			<i><specify if the component is “already developed” or “partially developed” or “to be developed from scratch”></i>		

Device/Gateway/Infrastructure Description and Functionality	
Name	<provide the name of the sensor>
Short Description	<provide a brief statement of the sensor, mentioning its WP/Task Number within the overall architecture>
Measurement	<provide description of the sensor measurement (directly, how, any restrictions)>
Digital/Analog Signals	<describe the signalling mode (analog, TTL, CMOS, etc.), if applicable for the sensor>
Functionality	<describe how the sensor functions within the eDREAM system architecture>
Physical Characteristics	
Dimensions	<L x W x H in mm>
Weight	<total weight of sensor in kg>
Material	<materials used for its construction>
Mounting	<how is sensor attached>
Operational Characteristics	
Measurement Range	<minimum to maximum values that can be measured by the sensor (e.g. -40 to +80 °C)>
Measurement Resolution	<level of measurement (e.g. to 0.01°C)>
Accuracy	<accuracy of the measurement (e.g. ±x% of actual reading)>

Zero Error	<i><amount required to pre-calibrate sensor and/or adjust readings by (e.g. $\pm 0.05^{\circ}\text{C}$)></i>
Humidity	<i><minimum to maximum humidity levels in %: range in which the sensor can operate></i>
Pressure	<i><minimum to maximum pressure levels in $\text{Pa}/\text{kgm}^{-3}/\text{N}$ etc.: range in which the sensor can operate></i>
Lifetime	<i><specify approximate lifetime under standard operating conditions></i>
Hardware Requirements	
Power Requirements	<i><specify electrical power supply required for sensor to operate without disruption></i>
Data Connections	<i><specify the communication networks and protocols involved e.g. USB, GSM, WiFi, Bluetooth etc.></i>
Data Format	<i><specify the output format of the sensor></i>
Data Rate	<i><specify at what rate data is read/extracted/logged></i>
Data Availability	<i><specify whether data stream is continuous, periodic, on demand etc.></i>
Transmission Frequency	<i><specify the power of the data stream, e.g. X mW, if applicable></i>
Software Requirements (e.g. API creation)	
Software Required	<i><yes/no></i>
Software Details	<i><provide details of software required for proper sensor function></i>

Note	<write any important note related to the sensor>
-------------	--

Real time data from Smart Meters	
Information Exchanged	Real time data (power, energy, frequency, voltage ...)
Description	Through this interface, eDREAM actors or modules may retrieve real time energy consumption for a specific device.
End-point URL	"publish": "BBB6038/SMX/", "subscribe": "MQTT/config/BBB6038/SMX/", "internal": ""
Parameters	["wallya1/RmsFormatted Voltage L1-N/value", "LD01/1-1-32-7-0-255/-2"], ["wallya1/RmsFormatted Voltage L2-N/value", "LD01/1-1-52-7-0-255/-2"], ["wallya1/RmsFormatted Voltage L3-N/value", "LD01/1-1-72-7-0-255/-2"], ["wallya1/Rms Voltage L1-N/value", "LD01/1-1-32-7-0-255/-9"], ["wallya1/Rms Voltage L2-N/value", "LD01/1-1-52-7-0-255/-9"], ["wallya1/Rms Voltage L3-N/value", "LD01/1-1-72-7-0-255/-9"], ["wallya1/RmsFormatted Current L1-N/value", "LD01/1-1-31-7-0-255/-2"], ["wallya1/RmsFormatted Current L2-N/value", "LD01/1-1-51-7-0-255/-2"], ["wallya1/RmsFormatted Current L3-N/value", "LD01/1-1-71-7-0-255/-2"], ["wallya1/Rms Current L1-N/value", "LD01/1-1-31-7-0-255/-9"], ["wallya1/Rms Current L2-N/value", "LD01/1-1-51-7-0-255/-9"], ["wallya1/Rms Current L3-N/value", "LD01/1-1-71-7-0-255/-9"], ["wallya1/ActiveFormatted Power Phase L1/value", "LD01/1-1-36-7-0-255/-2"], ["wallya1/ActiveFormatted Power Phase L2/value", "LD01/1-1-56-7-0-255/-2"], ["wallya1/ActiveFormatted Power Phase L3/value", "LD01/1-1-76-7-0-255/-2"], ["wallya1/ActiveFormatted Power 3-Phase/value", "LD01/1-1-16-7-0-255/-2"], ["wallya1/Active Power Phase L1/value", "LD01/1-1-36-7-0-255/-9"],

	<pre>["wallya1/Active Power Phase L2/value", "LD01/1-1-56-7-0-255/-9"], ["wallya1/Active Power Phase L3/value", "LD01/1-1-76-7-0-255/-9"], ["wallya1/Active Power 3-Phase/value", "LD01/1-1-16-7-0-255/-9"], ["wallya1/FundamentalFormatted Reactive Power Phase L1/value", "LD01/1-1-23-7-0-255/-2"], ["wallya1/FundamentalFormatted Reactive Power Phase L2/value", "LD01/1-1-43-7-0-255/-2"], ["wallya1/FundamentalFormatted Reactive Power Phase L3/value", "LD01/1-1-63-7-0-255/-2"], ["wallya1/FundamentalFormatted Reactive Power 3-Phase/value", "LD01/1-1-3-7-0-255/-2"], ["wallya1/Fundamental Reactive Power Phase L1/value", "LD01/1-1-23-7-0-255/-9"], ["wallya1/Fundamental Reactive Power Phase L2/value", "LD01/1-1-43-7-0-255/-9"], ["wallya1/Fundamental Reactive Power Phase L3/value", "LD01/1-1-63-7-0-255/-9"], ["wallya1/Fundamental Reactive Power 3-Phase/value", "LD01/1-1-3-7-0-255/-9"], ["wallya1/Delivered Active Energy/value", "LD01/1-1-2-8-0-255/-2"], ["wallya1/Received Active Energy/value", "LD01/1-1-1-8-0-255/-2"], ["wallya1/Frequency/value", "LD01/1-1-14-7-0-255/-2"]]</pre>
Allowed HTTP Methods	N/A
Class type of GET response	<pre>{ ".REQUEST_TIME": "2019-06-07 23:44:39", ".SERVER_TIME": "2019-06-07 21:44:38", "Active Power 3-Phase": { "unit": "W", "value": "988.267" }, "Active Power Phase L1": { "unit": "W", "value": "382.259" }, "Active Power Phase L2": { "unit": "W", "value": "129.896" }, "Active Power Phase L3": { "unit": "W", "value": "476.111" }, "Apparent Energy": { "unit": "kVAh", "value": "043132.0459" }, "Apparent Power 3-Phase": { "unit": "VA", "value": "004.280k" }, }</pre>

	<pre> "Apparent Power Phase L1": { "unit": "VA", "value": "001.621k" }, "Apparent Power Phase L2": { "unit": "VA", "value": "001.326k" }, "Apparent Power Phase L3": { "unit": "VA", "value": "001.334k" }, "Current U0/U1": { "unit": "%", "value": "000.384" }, "Current U2/U1": { "unit": "%", "value": "014.369" }, "Currents Sequence": { "value": "L1-L2-L3" }, "Delivered Active Energy": { "unit": "kWh", "value": "002573.4636" }, "Digital Inputs (0:OFF 1:ON)": { "value": "000000000000" }, "Flicker Inst Max L1-N": { "value": "000.072" }, "Flicker Inst Max L2-N": { "value": "000.075" }, "Flicker Inst Max L3-N": { "value": "000.068" }, "Flicker Plt L1-N": { "value": "000.255" }, "Flicker Plt L2-N": { "value": "000.253" }, "Flicker Plt L3-N": { "value": "000.289" }, "Flicker Pst L1-N": { "value": "000.118" }, "Flicker Pst L2-N": { "value": "000.119" }, "Flicker Pst L3-N": { "value": "000.119" }, "Frequency": { </pre>
--	--

	<pre> "unit": "Hz", "value": "050.049" }, "Fundamental PF 3-Phase": { "value": "0.254" }, "Fundamental Reactive Power 3-Phase": { "unit": "VAr", "value": "004.126k" }, "Fundamental Reactive Power Phase L1": { "unit": "VAr", "value": "001.572k" }, "Fundamental Reactive Power Phase L2": { "unit": "VAr", "value": "001.317k" }, "Fundamental Reactive Power Phase L3": { "unit": "VAr", "value": "001.237k" }, "InstFrequency": { "unit": "Hz", "value": "050.054" }, "Non-Active Power 3-Phase": { "unit": "VAr", "value": "004.140k" }, "Non-Active Power Phase L1": { "unit": "VAr", "value": "001.575k" }, "Non-Active Power Phase L2": { "unit": "VAr", "value": "001.320k" }, "Non-Active Power Phase L3": { "unit": "VAr", "value": "001.246k" }, "Non-Fundamental Power 3-Phase": { "unit": "VA", "value": "327.205" }, "Non-Fundamental Power Phase L1": { "unit": "VA", "value": "093.997" }, "Non-Fundamental Power Phase L2": { "unit": "VA", "value": "082.665" }, "Non-Fundamental Power Phase L3": { "unit": "VA", "value": "150.542" }, }, </pre>
--	---

	<pre> "OverDeviation L1": { "unit": "%", "value": "003.756" }, "OverDeviation L2": { "unit": "%", "value": "003.368" }, "OverDeviation L3": { "unit": "%", "value": "003.257" }, "OverDeviation L4": { "unit": "%", "value": "000.000" }, "PF 3-Phase": { "value": "0.231" }, "Q1 Reactive Energy": { "unit": "kVArh", "value": "029885.0854" }, "Q2 Reactive Energy": { "unit": "kVArh", "value": "000065.7651" }, "Q3 Reactive Energy": { "unit": "kVArh", "value": "011097.4121" }, "Q4 Reactive Energy": { "unit": "kVArh", "value": "000010.0820" }, "Received Active Energy": { "unit": "kWh", "value": "007302.4362" }, "Rms Current L1-N": { "unit": "A", "value": "006.799" }, "Rms Current L2-N": { "unit": "A", "value": "005.583" }, "Rms Current L3-N": { "unit": "A", "value": "005.620" }, "Rms Current L4-N": { "unit": "A", "value": "000.000" }, "Rms Voltage L1-L2": { "unit": "V", "value": "412.710" </pre>
--	--

	<pre> }, "Rms Voltage L1-N": { "unit": "V", "value": "238.638" }, "Rms Voltage L2-L3": { "unit": "V", "value": "411.242" }, "Rms Voltage L2-N": { "unit": "V", "value": "237.746" }, "Rms Voltage L3-L1": { "unit": "V", "value": "412.432" }, "Rms Voltage L3-N": { "unit": "V", "value": "237.492" }, "Rms Voltage L4-N": { "unit": "V", "value": "000.000" }, "THDI L1": { "unit": "%", "value": "006.401" }, "THDI L2": { "unit": "%", "value": "006.130" }, "THDI L3": { "unit": "%", "value": "011.435" }, "THDI L4": { "unit": "%", "value": "n/a" }, "THDV L1-N": { "unit": "%", "value": "002.090" }, "THDV L2-N": { "unit": "%", "value": "002.229" }, "THDV L3-N": { "unit": "%", "value": "002.319" }, "THDV L4-N": { "unit": "%", "value": "n/a" }, "UnderDeviation L1": { </pre>
--	--

	<pre> "unit": "%", "value": "000.000" }, "UnderDeviation L2": { "unit": "%", "value": "000.000" }, "UnderDeviation L3": { "unit": "%", "value": "000.000" }, "UnderDeviation L4": { "unit": "%", "value": "100.000" }, "Voltage U0/U1": { "unit": "%", "value": "000.075" }, "Voltage U2/U1": { "unit": "%", "value": "000.220" }, "Voltages Sequence": { "value": "L1-L2-L3" } } } </pre>
--	--

Charging Station API	
Information Exchanged	Charging station details and status
Description	Through this interface, eDREAM actors or modules may retrieve details and real-time status about each charging station.
End-point URL	https://panel.spot-link.it/public/api/chargeboxes/
Parameters	NULL: all charging station info {"chargeboxID":"ID"}: single charging station info Terni Pilot Site Charging Stations ID: SpotLink EVO ID =====> 24 (Sockets ID: 37 – 38) SpotLink EVO ID =====> 25 (Sockets ID: 39 – 40) FAST ID =====> 80 (Sockets ID: 126 -127)
Allowed HTTP Methods	GET

Class type of GET response	<a chargeboxid\":\"24\"}"="" href="https://panel.spot-link.it/public/api/chargeboxes/{\">https://panel.spot-link.it/public/api/chargeboxes/{\"chargeboxID\":\"24\"} <pre> { \"chargeboxID\": \"24\", \"address\": \"ASM Terni, Strada di Maratta Bassa, TR\", \"latitude\": \"42.5673558\", \"longitude\": \"12.6070454\", \"maxPwrAC\": \"32\", \"maxPwrDC\": \"0\", \"drStatus\": \"0\", \"tSocketA\": \"type 2\", \"stSocketA\": \"alarm\", \"tSocketB\": \"type 2\", \"stSocketB\": \"alarm\" } </pre>
-----------------------------------	--

Charging Session API	
Information Exchanged	Charging sessions info
Description	Through this interface, eDREAM actors or modules may retrieve details about real-time and historical charging sessions.
End-point URL	https://panel.spot-link.it/public/api/historyCharges/
Parameters	<p>NULL: Charging Session List</p> <p>{\"dateFrom\":\"Initialdate\", \"dateTo\":\"Finaldate\"} : Charging Session related to a time interval</p> <p>{\"chargeID\":\"sessionID\"}: Single charging session info</p> <p>{\"chargeboxID\":\"stationID\"}: list of charging session of a single charging station</p> <p>{\"chargeboxID\":\"stationID\", \"dateFrom\":\"Initialdate\", \"dateTo\":\"Finaldate\"}: list of charging session of a single charging station in a time interval</p> <p>{\"userID\":\"ID\"}: list of charging session of a single EV user</p>

	<p>{"userID":"ID", "dateFrom":"Initialdate", "dateTo":"Finaldate"}: list of charging session of a single EV user in a time interval</p> <p>The "date from" and "date to" format must correspond to: YYYY-MM-DD HH: MM: SS. If you want to take the whole day do not enter HH: MM: SS.</p>
Allowed HTTP Methods	GET
Class type of GET response	<p><a 09:00:00\",="" 13:00:00\"}"="" \"datefrom\":\"2019-06-14="" \"dateto\":\"2019-06-17="" chargeboxid\":\"24\",="" href="https://panel.spot-link.it/public/api/historyCharges/{\">https://panel.spot-link.it/public/api/historyCharges/{\"chargeboxID\":\"24\", \"dateFrom\":\"2019-06-14 09:00:00\", \"dateTo\":\"2019-06-17 13:00:00\"}</p> <pre>{ "numCharge": "2", "recharges": [{ "chargeID": "5806", "dataStart": "2019-06-17 06:51:14", "dataStop": "2019-06-17 08:01:54", "kwTot": 11.02, "importoTot": "3.97", "address": "ASM Terni, Strada di Maratta Bassa, TR", "idUser": "1403", "socketID": "37", "chargeboxID": "24", "nomePresa": "presa A" }, { "chargeID": "5789", "dataStart": "2019-06-14 10:37:30", "dataStop": "2019-06-14 11:16:59", "kwTot": 14.17, "importoTot": "5.1", "address": "ASM Terni, Strada di Maratta Bassa, TR", "idUser": "1403",</pre>

	<pre> "socketID": "38", "chargeboxID": "24", "nomePresa": "presa B" }] }</pre>
--	--

Remote Start API	
Information Exchanged	Start charging session command
Description	Through this interface, eDREAM actors or modules may start a charging session remotely
End-point URL	https://panel.spot-link.it/public/api/startChargebox/
Parameters	{"chargeboxID":"ID", "socketID":"ID"}
Allowed HTTP Methods	GET
Class type of GET response	<a \"socketid\":\"37\"}"="" chargeboxid\":\"24\",="" href="https://panel.spot-link.it/public/api/startChargebox/{\">https://panel.spot-link.it/public/api/startChargebox/{\"chargeboxID\":\"24\", \"socketID\":\"37\"} "START_OK"

Remote Stop API	
Information Exchanged	Stop charging session command
Description	Through this interface, eDREAM actors or modules may stop a charging session remotely
End-point URL	https://panel.spot-link.it/public/api/stopChargebox/
Parameters	{"chargeboxID":"ID", "socketID":"ID"}
Allowed HTTP Methods	GET
Class type of GET response	<a \"socketid\":\"37\"}"="" chargeboxid\":\"24\",="" href="https://panel.spot-link.it/public/api/stopChargebox/{\">https://panel.spot-link.it/public/api/stopChargebox/{\"chargeboxID\":\"24\", \"socketID\":\"37\"} "STOP_OK"

Remote Power Output Setup API	
Information Exchanged	Power output set-point command
Description	Through this interface, eDREAM actors or modules may modify charging station power output remotely
End-point URL	https://panel.spot-link.it/public/api/setPower/
Parameters	{"chargeboxID":"ID", "powerValue":"Ampere value"} Ampere value range is: 7A – 64A (integer number)
Allowed HTTP Methods	GET
Class type of GET response	https://panel.spot-link.it/public/api/setPower/{"chargeboxID":"24", "powerValue":"64"} "SET_OK"

Electric Vehicle API	
Information Exchanged	Electric vehicle details and status
Description	Through this interface, eDREAM actors or modules may retrieve details and real-time status about each electric vehicle.
End-point URL	https://panel.spot-link.it/public/api/ev/
Parameters	NULL: all electric vehicle info {"vehicleID":"ID"}: single electric vehicle info Terni Pilot Site EVs ID <ul style="list-style-type: none"> Renault ZOE FE132DG ID =====> -LRGfW9NO0SNzKtnbxZ8 Renault ZOE FE133DG ID =====> -LRGgC7MsyqieBj_-1aq Nissan Leaf 2011 FE454KF ID ==> -LRGpD8y9_U5tg2HmRSH Renault ZOE EY681VS ID =====> -LU9oyrvdmwDCqJO8fQD Nissan Leaf 2011 FG790NP ID => -LPRRuPLegLUMMzdV-sk Renault ZOE FH380BL ID =====> -LPelpMRcbj6TZQCXE7c
Allowed HTTP Methods	GET

Class type of GET response	<p>https://panel.spot-link.it/public/api/ev/{\"vehicleID\":\"-LRGfW9NO0SNzKtnbxZ8\"}</p> <pre>{\"vehicleID\":\"1\", \"model\":\"Renault ZOE 22\", \"connector\":\"type2\", \"batteryKw\":\"22\", \"batteryPower\":\"22\", \"licensePlate\":\"FM743BA\", \"status\":\"disconnected\", \"timestamp\":\"2019-06-24 08:17:08\", \"autonomyKm\":\"1023\", \"speed\":\"0\", \"batteryPerc\":\"0\", \"latitude\":\"42.5685485833333\", \"longitude\":\"12.6074580166667\", \"ready\":false, \"doorsLocked\":\"yes\", \"frontDX\":\"close\", \"frontSX\":\"close\", \"rearDX\":\"close\", \"rearSX\":\"close\", \"carTrunk\":\"close\"}</pre>
-----------------------------------	---

Energy Consumption Predictions – REST API	
Information Exchanged	Day-ahead Consumption Energy Predictions
Description	Through this interface, eDREAM actors or modules may retrieve day-ahead energy consumption predictions for a specific prosumer device.
End-point URL	/predictions/consumption/dayahead/{prosumerDeviceId}/{starttime}/{endtime}
Parameters	<p><i>prosumerDeviceId</i>: the id of the prosumer's device that the energy predictions are requested for.</p> <p><i>starttime</i>: the timestamp representing the start limit of the timeframe for which the energy predictions are requested. If the input timestamp does not correspond to a start of a day-ahead interval (start of day), then the start of the day-ahead interval containing the <i>starttime</i> is considered.</p> <p><i>endtime</i>: the timestamp representing the end limit of the timeframe for which the energy predictions are requested. If the input timestamp does not correspond to an end of a day-ahead interval (end of day), then the end of the day-ahead interval containing the <i>endtime</i> is considered.</p>
Allowed HTTP Methods	GET
Class type of GET response	<p>Energy Profile</p> <pre>{ \"profile\": [{\"value\": 57167.566, \"timestamp\": \"2018-09-08T00:00:00\" }, {\"value\": 57174.233, \"timestamp\": \"2018-09-08T01:00:00\" }, ...], \"prosumerDeviceId\": \"ab7d658d-84cd-4662-b573-74db92a297f2\",</pre>

	<pre>"deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "profileGranularityMinutes": 60, "predictionGranularity": "DAYAHEAD", "property": "ENERGY CONSUMPTION" }</pre>
Information Exchanged	Intra Day Consumption Energy Predictions
Description	Through this interface, actors or other modules may retrieve intra-day time energy predictions for a specific prosumer.
End-point URL	/predictions/consumption/intraday/{prosumerDeviceId}/{starttime}/{endtime}
Parameters	<p><i>prosumerDeviceId</i>: the id of the prosumer's device that the energy predictions are requested for.</p> <p><i>starttime</i>: the timestamp representing the start limit of the timeframe for which the energy predictions are requested. If the input timestamp does not correspond to a start of an intra-day interval, then the start of the intra-day interval containing the <i>starttime</i> is considered.</p> <p><i>endtime</i>: the timestamp representing the end limit of the timeframe for which the energy predictions are requested. If the input timestamp does not correspond to an end of an intra-day interval, then the end of the intra-day interval containing the <i>endtime</i> is considered.</p>
Allowed HTTP Methods	GET
Class type of GET response	<pre>Energy Profile { "profile": [{ "value": 28836.5, "timestamp": "2018-09-08T00:00:00" }, { "value": 28331.06, "timestamp": "2018-09-08T00:30:00" }, ...], "prosumerDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "profileGranularityMinutes": 30, "predictionGranularity": "INTRA-DAY", "property": "ENERGY CONSUMPTION" }</pre>
Information Exchanged	1hour-ahead Consumption Energy Predictions
Description	Through this interface, actors or other modules may retrieve 1hour-ahead energy predictions for a specific prosumer.
End-point URL	/predictions/consumption/1hour-ahead/{prosumerDeviceId}/{starttime}/{endtime}

Parameters	<p><i>prosumerDeviceId</i>: the id of the prosumer's device that the energy predictions are requested for.</p> <p><i>starttime</i>: the timestamp representing the start limit of the timeframe for which the energy predictions are requested. If the input timestamp does not correspond to a start of a 1hour-ahead interval, then the start of the 1hour-ahead interval containing the <i>starttime</i> is considered.</p> <p><i>endtime</i>: the timestamp representing the end limit of the timeframe for which the energy predictions are requested. If the input timestamp does not correspond to an end of a 1hour-ahead interval, then the end of the 1hour-ahead interval containing the <i>endtime</i> is considered.</p>
Allowed HTTP Methods	GET
Class type of GET response	<p>Energy Profile</p> <pre>{ "profile": [{ "value": 9691.81, "timestamp": "2018-09-08T00:00:00" }, { "value": 9606.23, "timestamp": "2018-09-08T00:10:00" }, ...], "prosumerDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "profileGranularityMinutes": 60 , "predictionGranularity": "1HOUR-AHEAD", "property": "ENERGY CONSUMPTION" }</pre>

Energy Production Predictions – REST API	
Information Exchanged	Day-ahead Production Energy Predictions
Description	Through this interface, actors or other modules may retrieve day-ahead production energy predictions for a specific prosumer device.
End-point URL	/predictions/production/dayahead/{prosumerDeviceId}/{starttime}/{endtime}
Parameters	<p><i>prosumerDeviceId</i>: the id of the prosumer's device that the energy predictions are requested for.</p> <p><i>starttime</i>: the timestamp representing the start limit of the timeframe for which the energy predictions are requested. If the input timestamp does not correspond to a start of a day-ahead interval (start of day), then the start of the day-ahead interval containing the <i>starttime</i> is considered.</p> <p><i>endtime</i>: the timestamp representing the end limit of the timeframe for which the energy predictions are requested. If the input timestamp does not</p>

	correspond to an end of a day-ahead interval (end of day), then the end of the day-ahead interval containing the <i>endtime</i> is considered.
Allowed HTTP Methods	GET
Class type of GET response	Energy Profile <pre>{ "profile": [{"value": 57167.566, "timestamp": "2018-09-08T00:00:00"}, }, {"value": 57174.233, "timestamp": "2018-09-08T01:00:00"}, }, ...], "prosumerDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "profileGranularityMinutes": 60 , "predictionGranularity": "DAYAHEAD", "property": "ENERGY PRODUCTION " }</pre>
Information Exchanged	Intra Production Energy Predictions
Description	Through this interface, actors or other modules may retrieve intra-day production energy predictions for a specific prosumer.
End-point URL	/predictions/production/intraday/{prosumerDeviceId}/{starttime}/{endtime}
Parameters	<p><i>prosumerDeviceId</i>: the id of the prosumer's device that the energy predictions are requested for.</p> <p><i>starttime</i>: the timestamp representing the start limit of the timeframe for which the energy predictions are requested. If the input timestamp does not correspond to a start of an intra-day interval, then the start of the intra-day interval containing the <i>starttime</i> is considered.</p> <p><i>endtime</i>: the timestamp representing the end limit of the timeframe for which the energy predictions are requested. If the input timestamp does not correspond to an end of an intra-day interval, then the end of the intra-day interval containing the <i>endtime</i> is considered.</p>
Allowed HTTP Methods	GET
Class type of GET response	Energy Profile <pre>{ "profile": [{"value": 28836.5, "timestamp": "2018-09-08T00:00:00"}, }, {"value": 28331.06, "timestamp": "2018-09-08T00:30:00"}] }</pre>

	<pre> }, ...], "prosumerDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "profileGranularityMinutes": 30 , "predictionGranularity": "INTRA-DAY", "property": "ENERGY PRODUCTION " } </pre>
Information Exchanged	1hour-ahead Production Energy Predictions
Description	Through this interface, actors or other modules may retrieve 1hour-ahead production energy predictions for a specific prosumer.
End-point URL	/predictions/production/1hour-ahead/{prosumerDeviceId}/{starttime}/{endtime}
Parameters	<p><i>prosumerDeviceId</i>: the id of the prosumer's device that the energy predictions are requested for</p> <p><i>starttime</i>: the timestamp representing the start limit of the timeframe for which the energy predictions are requested. If the input timestamp does not correspond to a start of a 1hour-ahead interval, then the start of the 1hour-ahead interval containing the <i>starttime</i> is considered.</p> <p><i>endtime</i>: the timestamp representing the end limit of the timeframe for which the energy predictions are requested. If the input timestamp does not correspond to an end of a 1hour-ahead interval, then the end of the 1hour-ahead interval containing the <i>endtime</i> is considered.</p>
Allowed HTTP Methods	GET
Class type of GET response	<p>Energy Profile</p> <pre> { "profile": [{ "value": 9691.81, "timestamp": "2018-09-08T00:00:00" }, { "value": 9606.23, "timestamp": "2018-09-08T00:10:00" }, ...], "prosumerDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "profileGranularityMinutes": 60 , "predictionGranularity": "1HOUR-AHEAD", "property": "ENERGY PRODUCTION" } </pre>

Energy Flexibility Predictions – REST API	
Information Exchanged	Day-ahead Flexibility Predictions

Description	Through this interface, actors or other modules may retrieve day-ahead flexibility predictions for a specific prosumer.
End-point URL	/predictions/flexibility/dayahead/{prosumerDeviceId}/{starttime}/{endtime}
Parameters	<p><i>prosumerDeviceId</i>: the id of the prosumer's device that the flexibility predictions are requested for.</p> <p><i>starttime</i>: the timestamp representing the start limit of the timeframe for which the flexibility predictions are requested. If the input timestamp does not correspond to a start of a day-ahead interval (start of day), then the start of the day-ahead interval containing the <i>starttime</i> is considered.</p> <p><i>endtime</i>: the timestamp representing the end limit of the timeframe for which the flexibility predictions are requested. If the input timestamp does not correspond to an end of a day-ahead interval (end of day), then the end of the day-ahead interval containing the <i>endtime</i> is considered.</p>
Allowed HTTP Methods	GET
Class type of GET response	<p>Flexibility Profile</p> <pre>{ "profile": [{ "value": 57167.566, "timestamp": "2018-09-08T00:00:00" }, { "value": 57174.233, "timestamp": "2018-09-08T01:00:00" }, ...], "prosumerDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "profileGranularityMinutes": 60 , "predictionGranularity": "DAYAHEAD", "property": "ENERGY FLEXIBILITY" }</pre>
Information Exchanged	Intra Flexibility Predictions
Description	Through this interface, actors or other modules may retrieve intra-day flexibility predictions for a specific prosumer.
End-point URL	/predictions/flexibility/intraday/{prosumerDeviceId}/{starttime}/{endtime}
Parameters	<p><i>prosumerDeviceId</i>: the id of the prosumer's device that the predictions are requested for.</p> <p><i>starttime</i>: the timestamp representing the start limit of the timeframe for which the flexibility predictions are requested. If the input timestamp does not correspond to a start of an intra-day interval, then the start of the intra-day interval containing the <i>starttime</i> is considered.</p> <p><i>endtime</i>: the timestamp representing the end limit of the timeframe for which the flexibility predictions are requested. If the input timestamp does not</p>

	correspond to an end of an intra-day interval, then the end of the intra-day interval containing the <i>endtime</i> is considered.
Allowed HTTP Methods	GET
Class type of GET response	<pre> Flexibility Profile { "profile": [{ "value": 28836.5, "timestamp": "2018-09-08T00:00:00" }, { "value": 28331.06, "timestamp": "2018-09-08T00:30:00" }, ...], "prosumerDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "profileGranularityMinutes": 30 , "predictionGranularity": "INTRA-DAY", "property": "ENERGY FLEXIBILITY" } </pre>

Virtual Power Plants Generation Modeling & Forecasting – REST API	
Information Exchanged	Construct VPP for energy trading
Description	Through this interface, actors or other modules may post requests for the construction of a coalition of prosumers able to buy/sell a specific amount of energy from the energy marketplace also considering the energy price signal.
End-point URL	/vpp/energy-trading
Allowed HTTP Methods	POST
Request Body	<pre> { "prosumers": [{ "prosumerId": "4836bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "predictedProfile": { [{ "value": 57167.566, "timestamp": "2018-09-08T00:00:00" }, { "value": 57174.233, "timestamp": "2018-09-08T01:00:00" }, ...], "entityDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", }] } </pre>

	<pre> "profileGranularityMinutes": 60, "predictionGranularity": "DAYAHEAD", "property": "ENERGY PRODUCTION" }, "uncertainty": { "min": 0.8, "max": 1.2, "entityDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "property": "Degradation-Trend" } "prosumerDetails": { "specification": {...} "type": "DEG" } },], "goal": { type : "ENERGY TRADING", "priceSignal": [{"value": 157167, "timestamp": "2018-09-08T00:00:00" }, {"value": 157234, "timestamp": "2018-09-08T01:00:00" }, ...] } } </pre>
Response	<pre> { "coalitionId": "4726bee7-bc42-48a9-9e4e-aa1ecc68e6f1", "selectedProsumers": [{ "prosumerId": "4836bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "prosumerType": "DEG", "tradedEnergy": [{"value": 57160, "timestamp": "2018-09-08T00:00:00" }, {"value": 57170, "timestamp": "2018-09-08T01:00:00" }, ...] }, ...] "totalEnergyTraded": [{"value": 157160, "timestamp": "2018-09-08T00:00:00" }, ...] } </pre>

	<pre> {"value": 157230, "timestamp": "2018-09-08T01:00:00" }, ...] } </pre>
Information Exchanged	Construct VPP for capacity bidding
Description	Through this interface, actors or other modules may post requests for the construction of a coalition of prosumers able to provide a replacement capacity on short notice.
End-point URL	/vpp/capacity-bidding
Allowed HTTP Methods	POST
Request Body	<pre> { "prosumers": [{ "prosumerId": "4836bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "predictedProfile": { [{ "value": 57167.566, "timestamp": "2018-09-08T00:00:00" }, { "value": 57174.233, "timestamp": "2018-09-08T01:00:00" }, ...], "entityDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "profileGranularityMinutes": 60, "predictionGranularity": "DAYAHEAD", "property": "ENERGY PRODUCTION" }, "uncertainty": { "min": 0.8, "max": 1.2, "entityDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "property": "Degradation-Trend" } "prosumerDetails": { "specification": {...} "type": "DEG" } },], </pre>

	<pre> "goal": { type": "CAPACITY BIDDING", "priceSignal": [{"value": 135000, "timestamp": "2018-09-08T00:00:00" }, {"value": 157000, "timestamp": "2018-09-08T01:00:00" }, ...] } } </pre>
Response	<pre> { "coalitionId": "4726bee7-bc42-48a9-9e4e-aa1ecc68e6f1", "selectedProsumers": [{ "prosumerId": "4836bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "prosumerType": "DEG", "biddedEnergy": [{ "value": 54412, "timestamp": "2018-09-08T00:00:00" }, { "value": 57123, "timestamp": "2018-09-08T01:00:00" }, ...] }, ...] "totalEnergyBidded": [{"value": 136000, "timestamp": "2018-09-08T00:00:00" }, {"value": 150000, "timestamp": "2018-09-08T01:00:00" }, ...] } </pre>
Information Exchanged	Construct VPP coalition for providing spinning reserve
Description	Through this interface, actors or other modules may request the dynamic construction of a VPP coalition of prosumers able to provide spinning reserve service on demand by activating or deactivating un-used capacity which can modify the reactive power.
End-point URL	/vpp/spinning-reserve
Allowed HTTP Methods	POST
Request Body	<pre> { "prosumers": [{ "prosumerId": "4836bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "predictedProfile": { [</pre>

	<pre> {"value": 57167.566, "timestamp": "2018-09-08T00:00:00" }, {"value": 54334.233, "timestamp": "2018-09-08T01:00:00" }, ...], "entityDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "profileGranularityMinutes": 60 , "predictionGranularity": "DAYAHEAD", "property": "ENERGY PRODUCTION" }, "uncertainty": { "min": 0.8, "max": 1.2, "entityDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "property": "Degradation-Trend" } "prosumerDetails": { "specification": {...} "type": "DEG" } },], "goal": { "type": "SPINNING RESERVE", "priceSignal": [{"value": 150000, "timestamp": "2018-09-08T00:00:00" }, {"value": 167000, "timestamp": "2018-09-08T01:00:00" }, ...] } } </pre>
Response	<pre> { "coalitionId": "4726bee7-bc42-48a9-9e4e-aa1ecc68e6f1", "selectedProsumers": [{ "prosumerId": "4836bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "prosumerType": "DEG", "biddedEnergy": [{"value": 57312, "timestamp": "2018-09-08T00:00:00" } }] } </pre>

	<pre> }, {"value": 54323, "timestamp": "2018-09-08T01:00:00" }, ...] }, ...] "totalEnergySpinned": [{"value": 150000, "timestamp": "2018-09-08T00:00:00" }, {"value": 136000, "timestamp": "2018-09-08T01:00:00" }, ...] } </pre>
Information Exchanged	Construct VVP coalition for demand response
Description	Through this interface, actors or other modules request the construction of a coalition of prosumers in VPP able to provide a requested target generation on demand.
End-point URL	/vpp/demand-response
Allowed HTTP Methods	POST
Request Body	<pre> { "prosumers": [{ "prosumerId": "4836bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "predictedProfile": [{ {"value": 57167.566, "timestamp": "2018-09-08T00:00:00" }, {"value": 57174.233, "timestamp": "2018-09-08T01:00:00" }, ...], "entityDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "profileGranularityMinutes": 60, "predictionGranularity": "DAYAHEAD", "property": "ENERGY PRODUCTION" }, "uncertainty": { "min": 0.8, "max": 1.2, "entityDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", </pre>

	<pre> "property": "Degradation-Trend" } "prosumerDetails": { "specification": {...} "type": "DEG" } },], "goal": { "type": "DEMAND RESPONSE", "priceSignal": [{"value": 150000, "timestamp": "2018-09-08T00:00:00" }, {"value": 112000, "timestamp": "2018-09-08T01:00:00" }, ...] } } } </pre>
Response	<pre> { "coalitionId": "4726bee7-bc42-48a9-9e4e-aa1ecc68e6f1", "selectedProsumers": [{ "prosumerId": "4836bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "prosumerType": "DEG", "biddedEnergy": [{"value": 57312, "timestamp": "2018-09-08T00:00:00" }, {"value": 53223, "timestamp": "2018-09-08T01:00:00" }, ...] }, ...] "totalEnergyMatched": [{"value": 150000, "timestamp": "2018-09-08T00:00:00" }, {"value": 140000, "timestamp": "2018-09-08T01:00:00" }, ...] } </pre>

Baseline Flexibility Estimation – REST API	
Information Exchanged	Day-ahead Consumption and production Flexibility Estimation

Description	Through this interface, eDREAM actors or modules may retrieve day-ahead energy consumption and production flexibilities for a specific prosumer device.
End-point URL	/edream/baselineFlexibilityEstimation/{prosumerDeviceId}/{starttime}/{endtime}
Parameters	<p>prosumerDeviceId: the id of the prosumer's device that the baseline flexibility estimation is requested for</p> <p>starttime: the timestamp representing the start limit of the timeframe for which the improved forecasting is requested</p> <p>endtime: the timestamp representing the end limit of the timeframe for which the improved forecasting is requested</p>
Allowed HTTP Methods	GET
Class type of GET response	<p>Baseline Flexibility Estimation</p> <pre>{ "prosumerDeviceId ": "ab7d658d-84cd-4662-b573-74db92a297f2", "property": " consumption " Flexibility values:[{"value": 57167.566,"timestamp": "2018-09-08T00:00:00"}, {"value": 57174.233, timestamp": "2018-09-08T01:00:00"}...] }</pre>

Short-term PV Degradation and trend analysis - REST API	
Information Exchanged	Improved short-term forecasting of generation
Description	Through this interface, eDREAM actors or modules may retrieve Improved short-term forecasting of generation for a specific prosumer device
End-point URL	/edream/improvedForecasting/shortTerm/{prosumerDeviceId}/{starttime}/{endtime}
Parameters	<p>prosumerDeviceId: the id of the prosumer's device that the Improved short-term forecasting of generation is requested for</p> <p>starttime: the timestamp representing the start limit of the timeframe for which the improved forecasting is requested</p> <p>endtime: the timestamp representing the end limit of the timeframe for which the improved forecasting is requested</p>
Allowed HTTP Methods	GET

Class type of GET response	<p>Improved short term forecasting</p> <pre>{ "prosumerDeviceId ": "ab7d658d-84cd-4662-b573-74db92a297f2", "property": "short term" Improved values:[{"value": 57167.566,"timestamp": "2018-09-08T00:00:00"}, {"value": 57174.233, timestamp": "2018-09-08T01:00:00"}...]</pre>
Long-term PV Degradation and trend analysis - REST API	
Information Exchanged	Improved long-term forecasting of generation
Description	Through this interface, eDREAM actors or modules may retrieve Improved long-term forecasting of generation for a specific prosumer device
End-point URL	/edream/improvedForecasting/longTerm/{prosumerDeviceId}/{starttime}/{endtime}
Parameters	<p>prosumerDeviceId: the id of the prosumer's device that the Improved long-term forecasting of generation is requested for</p> <p>starttime: the timestamp representing the start limit of the timeframe for which the improved forecasting is requested</p> <p>endtime: the timestamp representing the end limit of the timeframe for which the improved forecasting is requested</p>
Allowed HTTP Methods	GET
Class type of GET response	<p>Improved long term forecasting</p> <pre>{ "prosumerDeviceId ": "ab7d658d-84cd-4662-b573-74db92a297f2", "property": " long term" Improved values:[{"value": 57167.566,"timestamp": "2018-09-08T00:00:00"}, {"value": 57174.233, timestamp": "2018-09-08T01:00:00"}...]</pre>

Load Profiling - API	
Information Exchanged	Daily, weekly or monthly load profiles
Description	Through this interface, eDREAM actors or modules can request the load profiles of the selected customers.

End-point URL	/profiles/daily/{prosumer_id} /profiles/weekly/{prosumer_id} /profiles/monthly/{prosumer_id}
Parameters	start_time: string(ISO8601) end_time: string(ISO8601) type: [json, csv]
Allowed HTTP Methods	GET, POST, PUT, PATCH, DELETE
Class type of GET response	e.g. Energy Profile <pre>{ "profile":{ Profile_id: "string", Prosumer_id: "string", Profiling: [{"value": 57167.566, "timestamp": "2018-09-08T00:00:00"}, {"value": 57174.233, "timestamp": "2018-09-08T01:00:00" ...] } }</pre>

Clustering - API	
Information Exchanged	Clustering profiles based on the data provided
Description	Through this interface, eDREAM actors or modules may request different types of clusterization based on their needs
End-point URL	/clusters/daily/{profile_id} /clusters/weekly/{profile_id} /clusters/monthly/{profile_id}
Parameters	start_time: string(ISO8601) end_time: string(ISO8601) type: [json, csv]
Allowed HTTP Methods	GET, POST, PUT, PATCH, DELETE
Class type of GET response	e.g. Energy Cluster <pre>{</pre>

	<pre> "cluster":{ Cluster_id: "string", Profile_id: "string", Clustering: [{"value": 57167.566, "timestamp": "2018-09-08T00:00:00"}, {"value": 57174.233, "timestamp": "2018-09-08T01:00:00" ...] } </pre>
--	--

Customer Segmentation - API	
Information Exchanged	Classification of the selected user into one of the previously defined clusters
Description	Through this interface, eDREAM actors or modules may request the classification of a user or a subset of them into the defined clusters.
End-point URL	<pre> /segments/daily/{prosumer_id} /segments/weekly/{prosumer_id} /segments/monthly/{prosumer_id} </pre>
Parameters	<pre> start_time: string(ISO8601) end_time: string(ISO8601) type: [json, csv] </pre>
Allowed HTTP Methods	GET, POST, PUT, PATCH, DELETE
Class type of GET response	<pre> e.g. Energy Cluster { "segment":{ Segment_id: "string", Prosumer_id: "string", Segmentation: [{"value": 57167.566, "timestamp": "2018-09-08T00:00:00"}, {"value": 57174.233, "timestamp": "2018-09-08T01:00:00" ...] } } </pre>

Flexibility Profiling - REST API	
Information Exchanged	Flexibility margins of the prosumers' generation asset

Description	Through this interface, eDREAM actors or modules may retrieve flexibility margins of a specific prosumer's generation asset
End-point URL	/edream/flexibility margins/{prosumerDeviceId}/{starttime}/{endtime}
Parameters	<p>prosumerDeviceId: the id of the prosumer's device that the Improved long-term forecasting of generation is requested for</p> <p>starttime: the timestamp representing the start limit of the timeframe for which the improved forecasting is requested</p> <p>endtime: the timestamp representing the end limit of the timeframe for which the improved forecasting is requested</p>
Allowed HTTP Methods	GET
Class type of GET response	Flexibility margins <pre>{ "prosumerDeviceId ": "ab7d658d-84cd-4662-b573-74db92a297f2", margins:[{"value": 57167.566,"timestamp": "2018-09-08T00:00:00"}, {"value": 57174.233, "timestamp": "2018-09-08T01:00:00"}...] }</pre>

VPP and DR services Optimization Engine – REST API	
Information Exchanged	Day-ahead unit commitment and economic load dispatch scheduling
Description	Through this interface, eDREAM actors or modules may retrieve day-ahead optimal unit commitment and economic load dispatch scheduling of the prosumer's device connected to a specific VPP.
End-point URL	/edream/optimizationEngine/UCELD/{coalitionId}/{starttime}/{endtime}
Parameters	<p>coalitionId: the id of the VPP which containing the prosumers' devices that are part of the optimization</p> <p>starttime: the timestamp representing the start limit of the timeframe for which the improved forecasting is requested</p> <p>endtime: the timestamp representing the end limit of the timeframe for which the improved forecasting is requested</p>
Allowed HTTP Methods	GET
Class type of GET response	Optimized scheduling <pre>{ ["prosumerDeviceId ": "ab7d658d-84cd-4662-b573-74db92a297f2",</pre>

	<pre> scheduling [{"value": 57167.566,"timestamp": "2018-09-08T00:00:00"}, {"value": 57174.233, timestamp": "2018-09-08T01:00:00"}...] "prosumerDeviceId ": "ab7d658d-84cd-4662-b573-74db92a297f2", scheduling [{"value": 57167.566,"timestamp": "2018-09-08T00:00:00"}, {"value": 57174.233, timestamp": "2018-09-08T01:00:00"}...] ...] } </pre>
--	---

S	
Information Exchanged	Context Information
Description	Through this Interface context information regarding the smart meters is provided in near real-time each time there is a context update.
End-point URL	http://<host>:1026/v2/entities/<entity id>
Parameters	-
Allowed HTTP Methods	GET
Class type of GET response	<pre> { "SubscriptionId":"<id>", "Originator":"<host>", "ContextResponses":[{ "ContextElement":{ "type":"<type (String)>", "id":"<id>", "attributes":[{ "name":"P", "Type":"W", "value":"<int>", "Metadatas":[{ "name":"Timeinstant", </pre>

	<pre> "type": "ISO8601", "value": "<date>" }] }] }, "statusCode": { "Code": "200", "reasonPhrase": "OK" } }] } </pre>
Smart meter digital asset definition	
Information Exchanged	Smart meter attributes
Description	translate NGSI data in the appropriate format to be sent to BigchainDB.
End-point URL	http://<host>/meters/register/
Parameters	<pre> { "data": { "meter": { "serial_number": "SN", "manufacturer": "MANUFACTURER" } } } </pre>
Allowed HTTP Methods	POST
Class type of GET response	<pre> { "id": "<id>", } </pre>
Smart meter transaction	
Information Exchanged	Smart meter values
Description	In order to track the measures provided by the smart meter, new transactions containing the updated values will be appended to the blockchain using a transfer transaction, via this interface.

End-point URL	http://<host>:1026/v2/transaction/add/
Parameters	[{ "metadata":{ "recvTime":"ISO DATE", "attrName":"P", "attrType":"W", "attrValue":"NUMBER" }, "id":"ID" }]
Allowed HTTP Methods	POST
Class type of GET response	{ "status":"OK", }

Peer-to-peer Energy-trading related REST API	
Information Exchanged	Place Orders Based on Estimated Energy Profile
Description	Through this interface, prosumers may place several orders based on the estimated energy profile and prices provided for the next period of time.
End-point URL	/market/order/{market_session}/{account}
Parameters	market_session: the market session where the order is to be published account: the account identification information about the prosumer that publishes the orders
Allowed HTTP Methods	POST
Request Body	"prosumerTradingEnergy": { "estimatedEnergy": [20,40,-30,40,50,70,30,45,45,-23,-56, 34,76,34,34,65,34,23,76, 34,54,45,23,34], "tradingPrices": [20,40,30,40,50,70,30,45,45,23,56,34,76,34,34,65,34,23,76,34, 54,45,23,34], "profileStartHor": 0, "energyType": "GREEN", "marketSessionType":"DAYAHEAD" }

Response	<pre>{ "orderIds": ["0x9a37ac6ecdc856ea6e87698d889217803b82965e52f4af852dfcaf08bdd996b5", ...] }</pre>
Information Exchanged	Place Energy Order (BID or OFFER)
Description	Through this interface, prosumers may place individual orders (bids or offers) on the open sessions on chain.
End-point URL	/market/order/{market_session}/{ account }
Parameters	market_session: the type of the market session where the order is to be published account: the account identification information about the prosumer that publishes the order
Allowed HTTP Methods	POST
Request Body	<pre>{ "orderSide": "BID", "prosumerAddress": "0xAA21803000499f1b58C67F4DA7083AFA2ee37090", "timestamp": "123453123", "tokenId": 10, "metadata": { "startTimeToken": 0, "endTimeToken": 1, "energyType": "GREEN", }, "producer": "0xAA21803000499f1b58C67F4DA7083AFA2ee37090"; }</pre>
Response	<pre>{ "orderId": "0x9a37ac6ecdc856ea6e87698d889217803b82965e52f4af852dfcaf08bdd996b5" }</pre>
Information Exchanged	Register Matched Orders (BIDs and OFFERs)
Description	Through this interface, the oracle can publish the matching orders to be settled.
End-point URL	/energy/market/trades/{market_session_contract}
Parameters	market_session_contract: the address that identifies the market session on which the trades are to be published
Allowed HTTP Methods	POST
Request Body	<pre>[{ "id": "0x309911d2e38ff1649f7b6d39ef0e3fb33c87ae5d4fda6fa065ec9821c77fe2e3", "buyOrderId": "0x1c817f45dd2349c07b100ac0c9204d1652cea73006c0b000c9c39c4c40710d78", "sellOrderId": "0x95c8b70fc368ad3a3d7544715461d189616114cca00cabb97d2665a84e5d8b54", "prosumerBuyingAddress": "0x04fb94f5e2555d1e860462060337aa62ec6e919d", "prosumerSellingAddress": "0xAA21803000499f1b58C67F4DA7083AFA2ee37090", "timestamp": "123453123", }</pre>

	<pre>"tokenId":10, "quantity":12321 "price":35, }, ...]</pre>
Response	

Flexibility Services Management-related REST API	
Information Exchanged	Register Prosumer Flexibility Potential
Description	Through this interface, the prosumer can place his flexibility potential for the following period of time.
End-point URL	/prosumer/potential/{account}
Parameters	account: the account identification information of the prosumer that publishes the request
Allowed HTTP Methods	POST
Request Body	<pre>{ "baseline": [20,40,30,40,50,70,30,45,45,23,56,34,76,34,34,65,34,23,76,34,54,45,23,34], "flexibilityBelow": [10,30,20,30,40,60,20,35,35,13,46,24,66,24,24,55,24,13,66,24,44,35,13,24], "flexibilityAbove": [30,50,40,50,60,80,40,55,55,33,66,44,86,44,44,75,44,33,86,44,64,55,33,44] }</pre>
Response	
Information Exchanged	Register DSO Request for Flexibility
Description	Through this interface, the DSO account can place a flexibility request for the following period of time.
End-point URL	/dso/flexibility-request/{account}
Parameters	account: the account identification information of the DSO that publishes the request
Allowed HTTP Methods	POST
Request Body	<pre>{ "reward": [20,20], "flexibilityRequest": [50733, 62244, 61116, 72334, 48948, 63998, 75964, 41910, 73644, 106410, 67629, 125160, 145908, 71520, 176958, 82218, 121033, 115099, 79313, 42571, 101089, 46595, 77595, 37843] }</pre>
Response	
Information Exchanged	Register Aggregator Flexibility Offer

Description	Through this interface, the Aggregator account can place the optimal subset of the prosumers that can match the profile requested by the DSO. Based on the provided information, the aggregated profile is placed as a counter-offer to the DSO's request
End-point URL	/aggregator/optimal-prosumer-subset/{account}
Parameters	account: the account identification information of the aggregator that makes the request
Allowed HTTP Methods	POST
Request Body	<pre>[{ "prosumerAddress": "Prosumer 1", "flexibilityOrder": [20,40,30,40,50,70,30,45,45,23,56,34,76,34,34,65,34,23,76,34,54,45,23]}, { "prosumerAddress": "Prosumer 1", "flexibilityOrder": [20,40,30,40,50,70,30,45,45,23,56,34,76,34,34,65,34,23,76,34,54,45,23]} ...]</pre>
Response	<pre>{ "flexibilityAggregated": [50733, 62244, 61116, 72334, 48948, 63998, 75964, 41910, 73644, 106410, 67629, 125160, 145908, 71520, 176958, 82218, 121033, 115099, 79313, 42571, 101089, 46595, 77595, 37843] "evaluationScore": "1000", "riskFactor": "2" }</pre>
Information Exchanged	Get Prosumer's Flexibility
Description	Through this interface, the prosumer interrogates the flexibility request set by the aggregator for him to follow.
End-point URL	prosumer/flexibility-order/{account}
Parameters	account: the account identification information of the prosumer that makes the request
Allowed HTTP Methods	GET
Request Body	["prosumerAddress": "Prosumer 1"]
Response	[20,40,30,40,50,70,30,45,45,23,56,34,76,34,34,65,34,23,76,34,54,45,23,34]