The eDREAM project is co-founded by the EU's Horizon 2020 innovation programme
under grant agreement No 774478 Ref. Ares(2020)3432737 - 30/06/2020

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

DELIVERABLE: D2.10 Requirement-Driven System Development V3

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

Imprint

D2.5 Requirement-Driven System Development V3 (Month 30)

Contractual Date of Delivery to the EC:	30.06.2020
Actual Date of Delivery to the EC:	30.06.2020
Author(s):	Paschalis A. Gkaidatzis (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), Victoria Murcia (KIWI), Javier Gomez (ATOS), Lourdes Gallego (ATOS), Ugo Stecchi (ATOS), Juan Sancho (ATOS), Tommaso Bragatto (ASM), Francesca Santori (ASM), Claudia Pop (TUC), Marcel Antal (TUC), Francesco Bellesini (EMOT), Benjamin Hunter (TU), Fathi Abugchem (TU), Charalampos Psarros (TU), Vladimir Vukovic (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 – System Requirements, Functional and technical specifications
Confidentiality:	Public
Version:	1.0

Legal Disclaimer

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

Copyright

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

Executive Summary

This deliverable D2.10 describes the final reviewed version of the eDREAM architecture (initial version reported in D2.4 [1], and second revised version D2.5 [2]) due to be submitted at M30 (third iteration).

The aim of this deliverable is to present the final refinements 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) [3] and Use cases (D2.2) [4] and their updated versions, i.e. D2.6 [5], D2.8 [6] and D2.7 [7], D2.9 [8], respectively, and depicting the iterative analysis process for specifications' verification. The technical specifications were reviewed though 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 [1] and the second version, that is D2.5 [2]. This is a high-level view of the overall architecture, describing the three major layers of the eDREAM platform, namely, 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 UC list have been finalized in D2.9 "Use Case analysis and application scenarios description V3" and it is reported in this document for the sake of completeness [8].

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.

This report presents the final version of the eDREAM architecture definition, after the architectural components and their detailed specifications have been developed. Moreover, all components and

subsystems have been developed and deployed and all modules have been integrated to the eDREAM platform.

Since the previous versions of this deliverable, the functionalities of several tools have been merged or incorporated by other ones, namely, the functionalities of VPP and active Microgrid Flexibility Profiling have been incorporated by the VPP generation, modelling and forecasting and baseline flexibility estimation. This has also affected the High-level Use Cases 2 and 3, i.e. HL-UC03_LL-UC02 and HL-UC03_LL-UC03 and the changes can be seen in both this deliverable and also Deliverable D2.9, where the Use Case are described in more detail [8]. With respect to the deployment view of the Terni Pilot, the charging stations installed are now described in more detail and technical specifications are provided. Additionally, the non-functional requirements of the involved tools are collected, in order to be utilized for the validation phase of the project and more specifically, within the activities of WP7 and T7.2.

Table of Contents

Executive Summary ... 3
List of Figures ... 6
List of Tables ... 8
List of Acronyms and Abbreviations ... 11
1 Introduction ... 13
1.1 Scope and objectives of the deliverable and relevance in the eDREAM framework ... 13
1.2 Structure of the deliverable ... 14
2 Methodology ... 16
2.1 System Architecture Concepts and Design Fundamentals ... 17
2.1.1 Design Principles ... 17
2.1.2 Static and Dynamic Structures ... 18
2.1.3 End-users and Stakeholder requirements' perspective ... 18
2.1.4 Architectural Views ... 18
2.1.5 Architectural Elements Perspectives ... 19
3 Conceptual Architecture ... 21
4 Structural – Functional View ... 25
4.1 Overall Structural View of eDREAM architecture ... 25
4.2 Field Middleware ... 27
4.2.1 IoT Devices ... 27
4.2.2 Semantic Context Broker ... 28
4.3 Techniques for DR and Energy Flexibility Assessment ... 29
4.3.1 Electricity Consumption/Production Forecasting ... 29
4.3.2 PV/RES Degradation & Trend Analysis ... 30
4.3.3 Baseline Flexibility Estimation ... 31
4.3.4 Virtual Power Plants Generation Modelling & Forecasting ... 31
4.3.5 Multi-building DR characterization through thermal, optical and LIDAR information fusion ... 32
4.4 Next generation DR Services for Aggregators and Customers ... 33
4.4.1 Load Profiling ... 33
4.4.2 Big Data Clustering at Multiple Scales ... 33
4.4.3 Customers Segmentation ... 34
4.4.4 :VPP & DR Services Optimization Engine ... 35
4.5 Blockchain-enabled Decentralized Network Control Optimization and DR Verification ... 36
4.5.1 Distributed Ledger ... 36
4.5.2 Blockchain-driven control for LV networks (flexibility management) ... 37
4.5.3 Blockchain-driven Energy Market ... 38
4.5.4 Closed loop DR Verification Engine ... 39
4.6 Multi-level and Multi-factor Visualization Framework (Front-End) ... 40

4.6.1 HMIs ... 41
4.6.2 Suites ... 41
5 Dynamic View ... 45
5.1 High Level UC 01: Prosumers DR flexibility aggregation via smart contract ... 45
5.1.1 HL-UC01_LL-UC01: Prosumers enrolment in demand response programs ... 46
5.1.2 HL-UC01_LL-UC02: Contract Setting ... 49
5.1.3 HL-UC01_LL-UC03: Potential energy flexibility evaluation ... 53
5.1.4 HL-UC01_LL-UC04: Energy demand/production forecasting for day-ahead trading of flexibility ... 55
5.1.5 HL-UC01_LL-UC05: Flexibility request ... 59
5.1.6 HL-UC01_LL-UC06: Flexibility offering ... 63
5.1.7 HL-UC01_LL-UC07: Flexibility acceptance ... 68
5.1.8 HL-UC01_LL-UC08: Flexibility provisioning ... 71
5.2 High Level UC 02: Peer-to-peer local energy trading market ... 74
5.2.1 HL-UC02_LL-UC01: Prosumers registration with the energy trading platform ... 74
5.2.2 HL-UC02_LL-UC02: Prosumers bids/offers submission ... 77
5.2.3 HL-UC02_LL-UC03: Energy clearing price determination ... 80
5.2.4 HL-UC02_LL-UC04: Transactions validation and financial settlement ... 82
5.3 High Level UC: VPP in Energy Community ... 85
5.3.1 HL-UC03_LL-UC01: Prosumers Profiling and Clusterization ... 85
5.3.2 HL-UC03_LL-UC02: VPP Capability Evaluation for Reserve services and for Frequency services ... 88
5.3.3 HL-UC03_LL-UC03: VPP Export Evaluation for Wholesale market (Intraday trading) and for Imbalance market ... 92
6 Deployment View ... 96
6.1 Active Micro-Grid – ASM Terni ... 96
6.2 Community-based Virtual Power Plants – KiWi Power ... 103
6.3 Overall Deployment Architecture ... 104
6.4 Field Devices ... 108
7 Architectural Components Detailed Specifications ... 116
8 Conclusion ... 169
References ... 170
Annex I: Functional & Non-Functional Requirements ... 173
Annex II: Architectural Specifications Templates ... 187
Annex III: eDREAM Sensors/Gateways/Infrastructure Specifications Template ... 190
Annex IV: Field Devices APIs ... 193
Annex V: Architectural Components APIs ... 205

List of Figures

Figure 1: Architecture Design Approach and Workflow 16
Figure 2: Architectural View 4+1 model 19
Figure 3: eDREAM Conceptual Architecture 21
Figure 4: eDREAM overall structural view 26
Figure 5: Field Data Aggregation layer 27
Figure 6: Electricity Consumption/Production Forecasting 30
Figure 7: PV/RES Degradation & Trend Analysis 31
Figure 8: Baseline Flexibility Estimation 31
Figure 9: Virtual Power Plants Generation Modelling & Forecasting 32
Figure 10: Multi-building DR Characterization through optical, thermal and LIDAR information fusion 33
Figure 11: Load Profiling 33
Figure 12: Big Data Clustering at Multiple Scales 34
Figure 13: Customer Segmentation 35
Figure 14: VPP and DR Services Optimization Engine 36
Figure 15: Blockchain-driven control for LV networks 38
Figure 16: Secured Blockchain-driven Energy Market 39
Figure 17: Closed-loop DR Verification Engine 40
Figure 18: DSS & DR Strategies Optimization 42
Figure 19: DR Aerial Survey Toolkit 43
Figure 20: Forecasting Tool 43
Figure 21: Graph-based Analytics 44
Figure 22: Internal and external connections in the Terni Micro-Grid 96
Figure 23: Unbundled Smart Meter Concept 97
Figure 24: High-level architecture for the communication between SM and SMX 98
Figure 25: Emotion SpotLink EVO charging stations 99
Figure 26: Renault ZOE customized by Emotion 100
Figure 27: Nissan LEAF 101
Figure 28: EMOT network topology 102
Figure 29: Residential estate in Greenwich for testing residential demand response scenarios 103
Figure 30: eDREAM Deployment View Architecture 105
Figure 31: IoT Agents' Concept and Connection 106
Figure 32: Blockchain-based Infrastructure 108
Figure 33: Testbed Topology for the Distributed Ledger Technology 108

List of Tables

Table 1: Main objectives of the three versions of the deliverable for System Requirements and Technical Specifications Definition ... 13
Table 2: Quality properties and perspectives for architectural elements ... 19
Table 3: List of identified architectural components, assigned tasks and partners responsibilities ... 23
Table 4: HL-UC01_LL-UC01: Prosumers enrolment in demand response programs ... 46
Table 5: HL-UC01_LL-UC02: Contract Setting ... 49
Table 6: HL-UC01_LL-UC03: Potential energy flexibility evaluation ... 53
Table 7: HL-UC01_LL-UC04: Energy demand/production forecasting for day-ahead trading of flexibility ... 55
Table 8: HL-UC01_LL-UC05: Flexibility request ... 59
Table 9: HL-UC01_LL-UC06: Flexibility offering ... 63
Table 10: HL-UC01_LL-UC07: Flexibility acceptance ... 68
Table 11: HL-UC01_LL-UC08: Flexibility provisioning ... 71
Table 12: HL-UC02_LL-UC01: Prosumers registration with the energy trading platform ... 74
Table 13: HL-UC02_LL-UC02: Prosumers bids/offer submission ... 77
Table 14: HL-UC02_LL-UC03: Energy clearing price determination ... 80
Table 15: HL-UC02_LL-UC04: Transactions validation and financial settlement ... 82
Table 16: HL-UC03_LL-UC01: Prosumers Profiling and Clusterization ... 85
Table 17: VPP Capability Evaluation for Reserve services and for Frequency services ... 88
Table 18: HL-UC03_LL-UC03: VPP Export Evaluation for Wholesale market (Intraday trading) and for Imbalance market ... 92
Table 19: Smart Metrology Meter (SMM) ... 109
Table 20: Smart Meter Extension (SMX) ... 110
Table 21: EVO Emotion ... 111
Table 22: OBD Emotion ... 112
Table 23: Fruit KiWi Power ... 114
Table 24: Micro-grid Monitor ... 116
Table 25: EVSEs and EV fleet monitoring ... 117
Table 26: Electricity Consumption/Production Forecasting ... 118
Table 27: Virtual Power Plants Generation Modelling & Forecasting ... 123
Table 28: Baseline Flexibility Estimation ... 128
Table 29: PV/RES Degradation & Trend Analysis ... 130
Table 30: Multi-building DR characterization through optimal, thermal and LIDAR information fusion ... 133
Table 31: Load Profiling ... 135
Table 32: Big Data Clustering at Multiple Scales ... 137
Table 33: Customer Segmentation ... 139
Table 34: VPP and DR Services Optimization Engine ... 141
Table 35: Distributed Ledger ... 144
Table 36: Blockchain-driven control for LV networks (flexibility management) ... 146
Table 37: Secured Blockchain-driven Energy Market ... 150
Table 38: Closed loop DR Verification Engine ... 154
Table 39: DSS (Decision Support System) & DR Strategies Optimization ... 157
Table 40: DR Aerial Survey Toolkit ... 159
Table 41: Graph-Based Analytics ... 161
Table 42: Forecasting Tool ... 164
Table 43: HMI ... 166
Table 44: Electricity Consumption/Production Forecasting – FRs ... 173
Table 45: Virtual Power Plants Generation Modelling & Forecasting - FRs ... 173
Table 46: PV/RES Degradation & Trend Analysis - FRs ... 174
Table 47: Baseline Flexibility Estimation - FRs ... 174
Table 48: Multi-building DR characterization through thermal, optical and LIDAR information fusion - FRs ... 174
Table 49: Load Profiling - FRs ... 175

Table 50: Big Data Clustering at Multiple Scales - FRs 175
Table 51: Customer Segmentation - FRs 175
Table 52: VPP and DR Services Optimization Engine - FRs........ 176
Table 53: Distributed Ledger - FRs 176
Table 54: Blockchain-driven control for LV networks - FRs 176
Table 55: Secured Blockchain-driven Energy Market - FRs 177
Table 56: Closed loop DR Verification Engine - FRs........ 177
Table 57: Graph-based Analytics - FRs 177
Table 58: Decision Support System & DR Strategies Optimization – FRs........ 178
Table 59: DR Aerial Survey Toolkit – FRs 178
Table 60: Forecasting Tool – FRs 178
Table 61: HMI - FRs 179
Table 62: EVSEs and EV fleet monitoring – FRs........ 179
Table 63: Electric meters, edge, and field device electric measures – FRs 179
Table 64: Electricity Consumption/Production Forecasting – NFRs........ 180
Table 65: Virtual Power Plants Generation Modelling & Forecasting - NFRs........ 180
Table 66: PV/RES Degradation & Trend Analysis – NFRs 180
Table 67: Baseline Flexibility Estimation - NFRs........ 180
Table 68: Multi-building DR Characterization through thermal, optical and LIDAR information fusion – NFRs.... 180
Table 69. Load Profiling - NFRs........ 181
Table 70: Big Data Clustering at Multiple Scales - NFRs........ 181
Table 71: Customer Segmentation – NFRs........ 182
Table 72: VPP and DR Services Optimization engine - NFRs 182
Table 73: Distributed Ledger – NFRs........ 182
Table 74: Blockchain-driven Control for LV networks - NFRs 183
Table 75: Secured Blockchain-driven Energy Market – NFRs........ 183
Table 76: Closed loop DR Verification Engine – NFRs 183
Table 77: EVSEs and EV fleet monitoring – NFRs 183
Table 78: Graph-based Analytics - NFRs 184
Table 79: HMIs – NFRs 184
Table 80: DSS (Decision Support System) & DR Strategies Optimization – NFRs........ 185
Table 81: DR Aerial Survey Toolkit - NFRs 185
Table 82: Forecasting Tool – NFRs........ 186
Table 83: Functional and Non-Functional Requirements Template........ 187
Table 84: Architectural Components Detailed Specifications Template 187
Table 85: Smart Meters API 193
Table 86: Charging Station API 199
Table 87: Charging Session API 200
Table 88: Remote Start API 202
Table 89: Remote Stop API........ 202
Table 90: Remote Power Output Setup API 203
Table 91: Electric Vehicle API 203
Table 92: Electricity Consumption Forecasting - API........ 205
Table 93: Electricity Production Forecasting API 207
Table 94: Energy Flexibility Forecasting API 209
Table 95: Virtual Power Plants Generation Modelling & Forecasting - API 211
Table 96: Baseline Flexibility Estimation - API........ 217
Table 97: PV/RES Degradation & Trend Analysis - API 218
Table 98: Load Profiling - API 219
Table 99: Big Data Clustering at Multiple Scales - API........ 220
Table 100: Customer Segmentation - API 221
Table 101: VPP and DR Services Optimization Engine - API........ 221
Table 102: Distributed Ledger API........ 222
Table 103: Secured Blockchain-driven Energy Market - API 225

Table 104: Blockchain-driven control for LV networks - API 226

List of Acronyms and Abbreviations

ANN	Artificial Neural Networks
BRs	Business Requirements
CBL	Customer Baseline Load
CHP	Combined Heat & Power
CIM	Common Information Model
DB	Database
DEG	Distributed Energy Generators
DER	Distributed Energy Resources
DLT	Distributed Ledger Technology
DR	Demand Response
DSO	Distribution System Operator
DSS	Decision Support System
eDREAM	enabling new Demand Response Advanced, Market oriented and secure technologies, solutions and business models
ESS	Energy Storage Systems
EV	Electric Vehicle
EVSEs	Electric Vehicle Supply Equipment
FD	Field Data
FDA	Flexible Demand Assets
HD	High Definition
HL-UC	High Level Use Case
HMI	Human Machine Interface
IoT	Internet of Things

KPI	Key Performance Indicator
LIDAR	Light Detection and Ranging
LL-UC	Low Level Use Case
MF	Macro-Functionality
PD	Participatory Design
ToU	Time-of-Use
UAV	Unmanned Aerial Vehicle
UI	User Interface
UPS	Uninterruptible Power Supply
URs	User Requirements
VPP	Virtual Power Plant
WP	Work Package

1 Introduction

The purpose of this deliverable, as the technical output of the project, is to present the third and final 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 and a half year of the project and can be considered as a key input for the upcoming task in WP2 concerning requirement tracking (T2.5), 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.

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

The purpose of this deliverable is to describe the third and final 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 mainly focuses 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 are addressed:

- Conceptual Architecture Design Process: within this part an overall view of the eDREAM architecture is presented comprising of 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.

Table 1: Main objectives of the three versions of the deliverable for System Requirements and Technical Specifications Definition

Deliverable	Objectives
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.

1.2 Structure of the deliverable

D 2.10 "Requirement-Driven System Development V3" consists of eight chapters, in which the third 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 has been 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.

2 Methodology

This section presents the approach and methodology that have been followed by Task 2.4 to define the various versions of the architecture. Task 2.4 started on M3 and it has run continuously until M30. The first version of the architecture comprises of 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 M30:

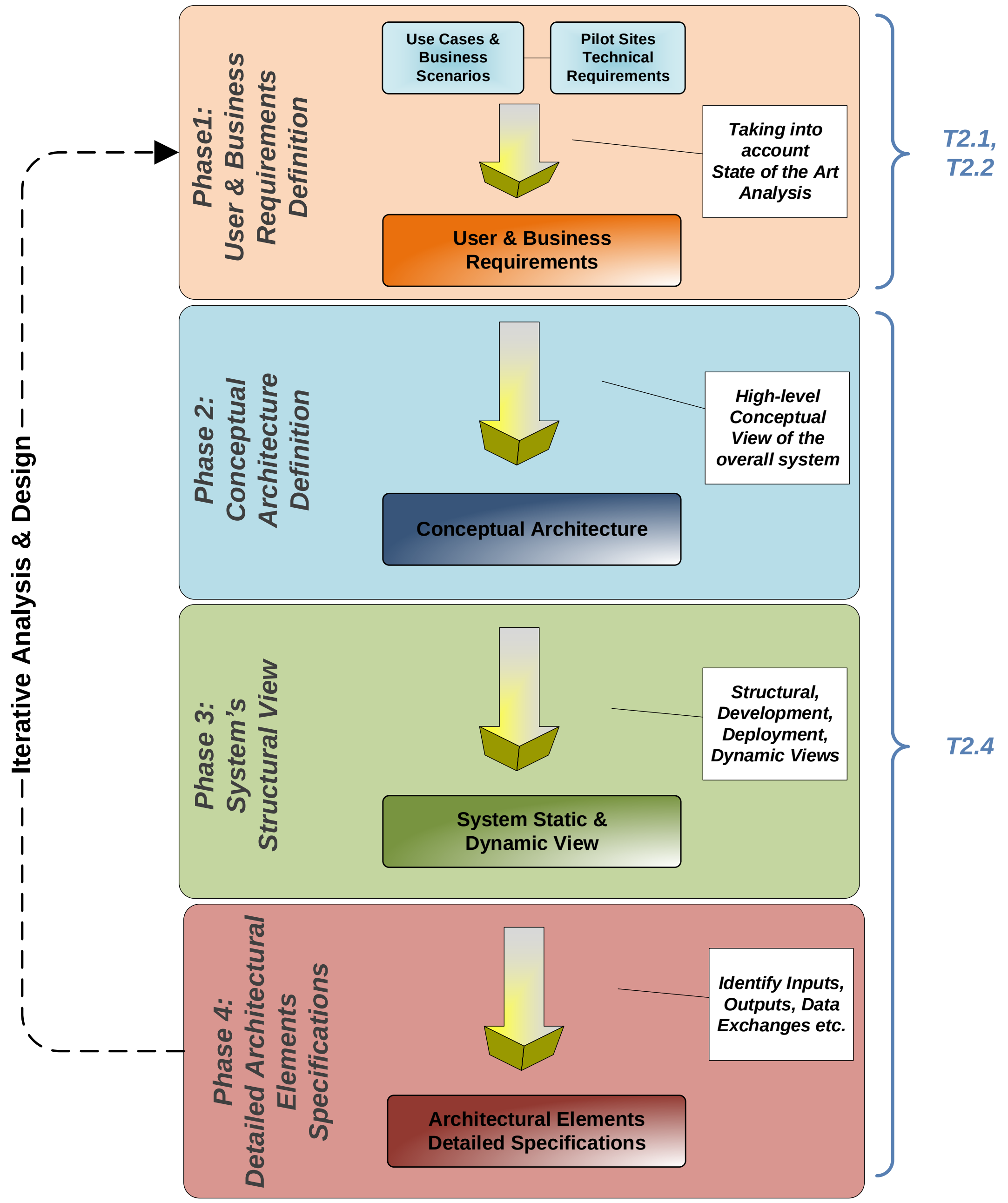


Figure 1: Architecture Design Approach and Workflow

2.1 System Architecture Concepts and Design Fundamentals

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 with 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.

2.1.1 Design Principles

Following basic design principles, the architecture is open and modular, so that all vendors, suppliers, and potential users are 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 architectural designers and developers shall avoid large designing and potential implementation of components at premature stages.

2.1.2 Static and Dynamic Structures

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 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.

2.1.3 End-users and Stakeholder requirements' perspective

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.

2.1.4 Architectural Views

In the context of eDREAM, the 4+1 architectural view model has been used, to present the concurrent views.

The concept is depicted below:

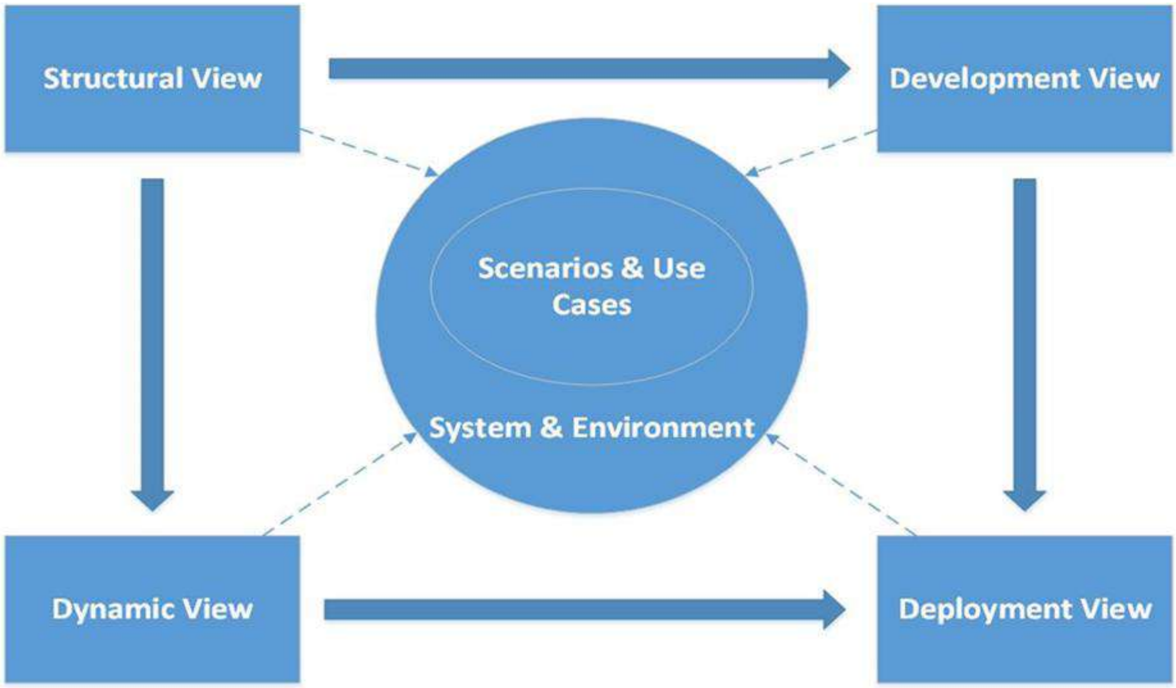


Figure 2: Architectural View 4+1 model

2.1.5 Architectural Elements Perspectives

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:

Table 2: Quality properties and perspectives for architectural elements

Perspective	***Desired Quality***
General Purpose	
Performance and Scalability	*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.*

Availability and Resilience	*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.*
Security	*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.*
Evolution	*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.*
Additional Perspectives to cope with eDREAM non-functional requirements	
Maintenance	*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.*
Privacy & Regulation	*The ability of the system and its architectural elements to conform to national and international laws, policies and other rules and standards.*
Usability	*The ease with which key stakeholders of a system are capable to work effectively and to interact with it in a user-friendly way.*

For each of the aforementioned perspectives, the importance of 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, by addressing in the architecture definition process the importance of the aforementioned perspectives has further helped to the later decision making (implementation, deployment and operational phases). Within eDREAM, a table is 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 have met the functional and non-functional requirements, the above proposed perspectives have been taken into account. These perspectives could be modified or enriched by partners according to characteristics of the components.

3 Conceptual Architecture

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 are 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:

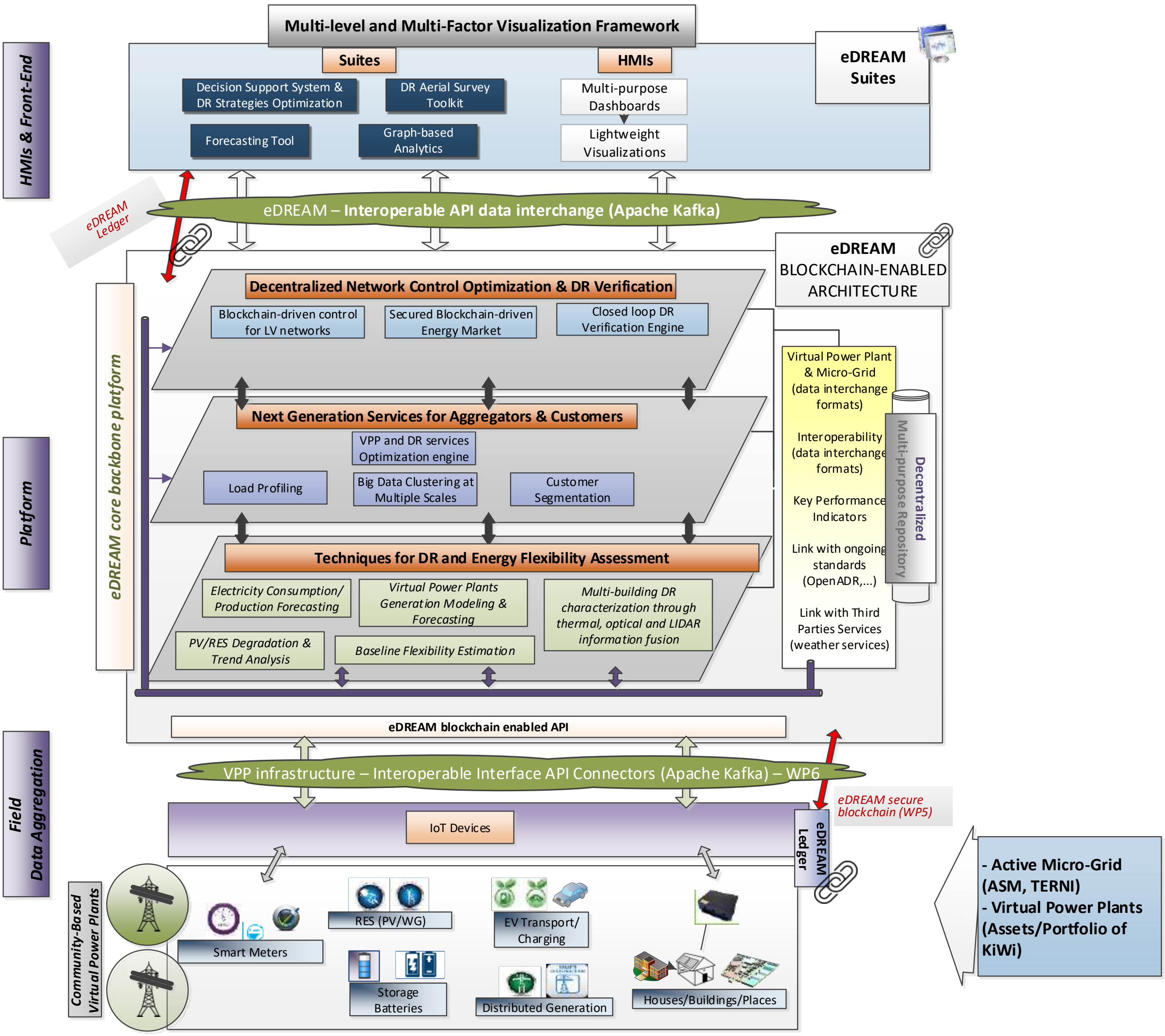


Figure 3: eDREAM Conceptual Architecture

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 forwards 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 is 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** that 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, each of them is assigned to a project WP; notably, they are the following::
    - ***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 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 are investigated aiming at creating optimal coalitions and providing more reliable aggregated supply. The constitution of these coalitions also enables small prosumers to participate in DR programs, such as households of 1kW capacity generation. Furthermore, novel techniques based on aerial survey techniques are 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 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) are investigated. A Big Data Analytics Engine is researched and developed for analysing large streams (including micro batch level) collected from customers. Furthermore, big data clustering techniques at

multiple scales are investigated towards creating customers' clusters with specific load and generation profile patterns.

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

  This sub-layer mostly focuses 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 are performed in a fully decentralized manner. In addition, Proof-of-Stake consensus based algorithms for closed-loop DR programs execution, verification and financial settlement are examined. Finally, the delivery of a Graph-based Analytics platform is going to 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 analysed 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 stores and maintains 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.

Table 3: List of identified architectural components, assigned tasks and partners responsibilities

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 Modelling & 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
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.

4 Structural – Functional View

4.1 Overall Structural View of eDREAM architecture

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 final definition of the modules interfaces and APIs have been completed and presented in the following sections.

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

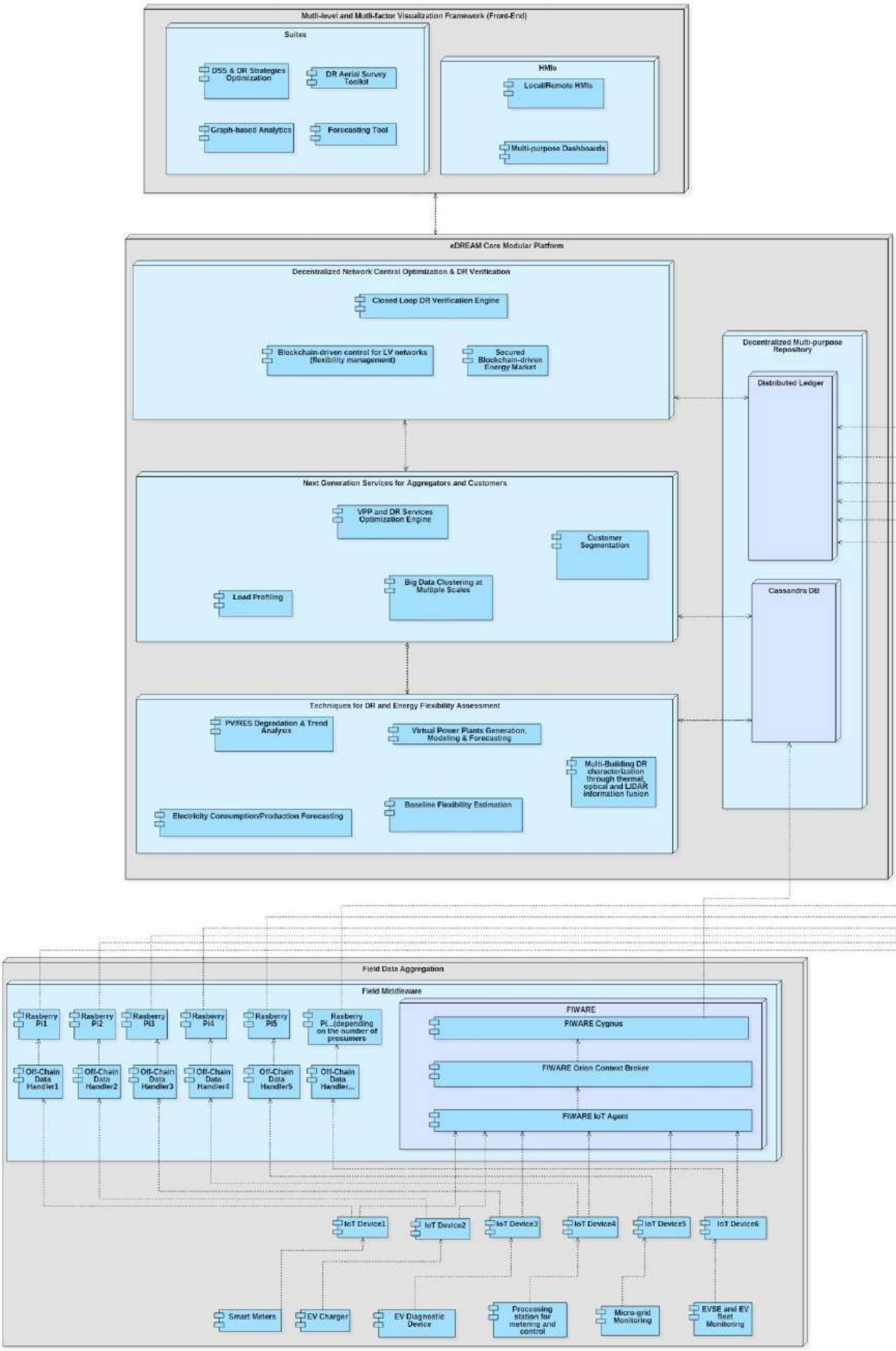


Figure 4: eDREAM overall structural view

4.2 Field Middleware

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 pilot sites will be implemented and the communication paths with the Repository of the project.

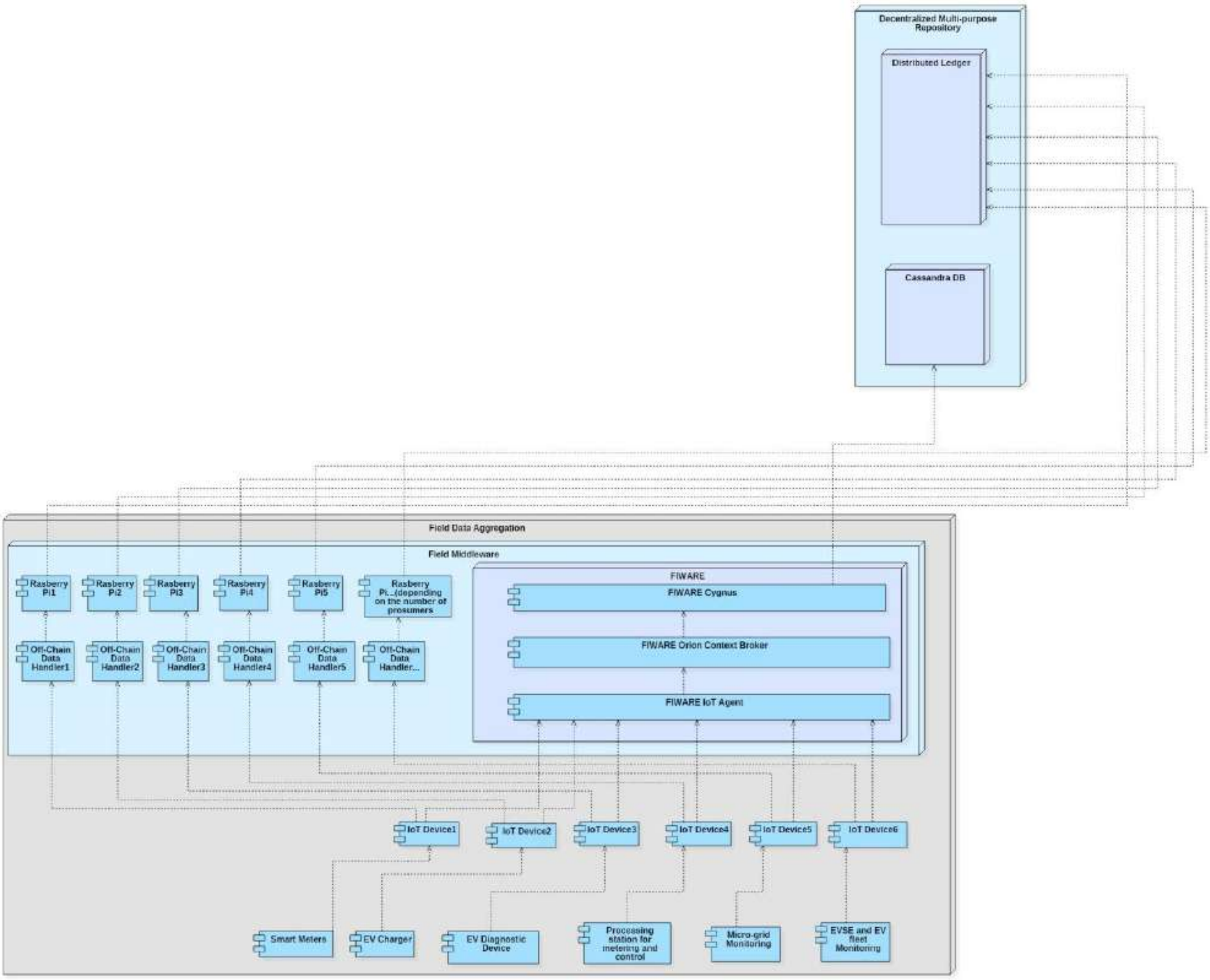


Figure 5: Field Data Aggregation layer

4.2.1 IoT Devices

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 also describes 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 Machine2Machnine (M2M) communication standards.

4.2.2 Semantic Context Broker

Description – Main Functionalities

The **Semantic Context Broker** is 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 modelling 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 which 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 uses context data from the Decentralized Repository according to respective standards (e.g. OpenADR, CIM etc.) in order to enable enhanced data structures.

4.3 Techniques for DR and Energy Flexibility Assessment

4.3.1 Electricity Consumption/Production Forecasting

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 deliverables D3.1 [9] and D3.5 [10].

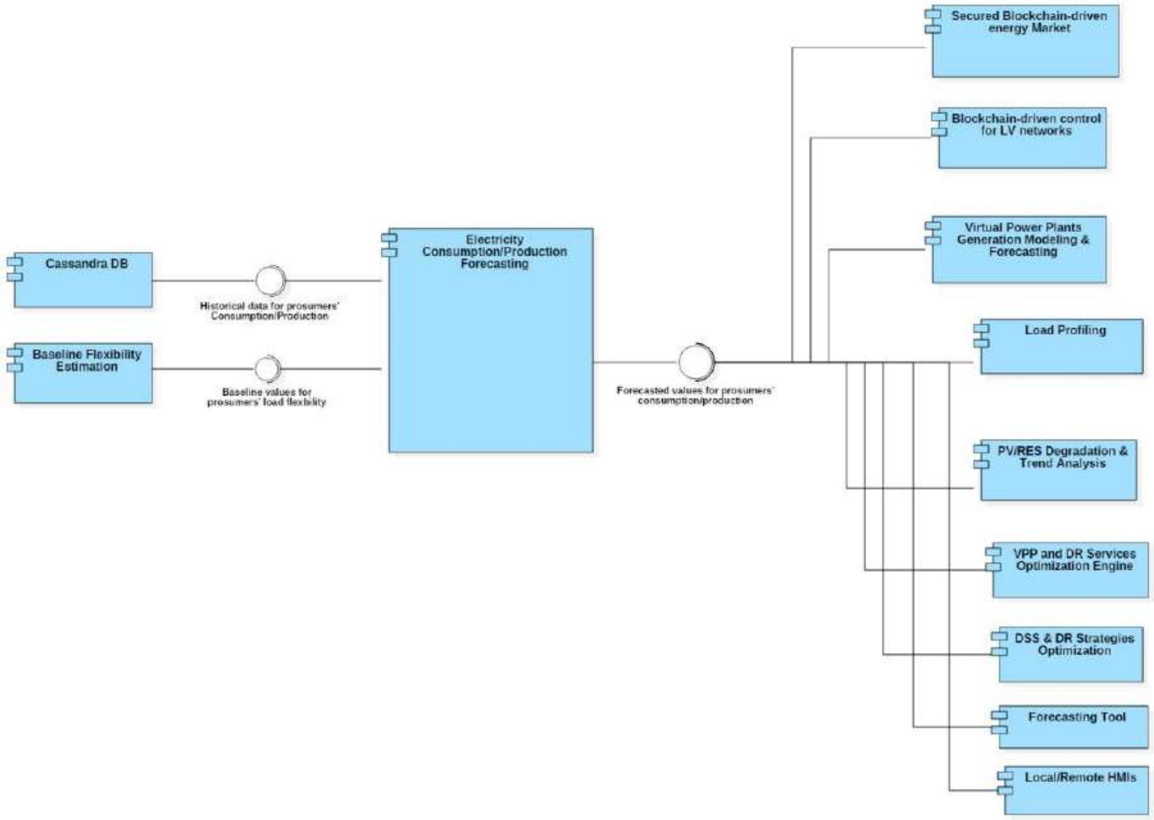


Figure 6: Electricity Consumption/Production Forecasting

4.3.2 PV/RES Degradation & Trend Analysis

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 detects the uncertain changes in production, such as due to weather conditions. This component is further described in the deliverable D4.5 [11].

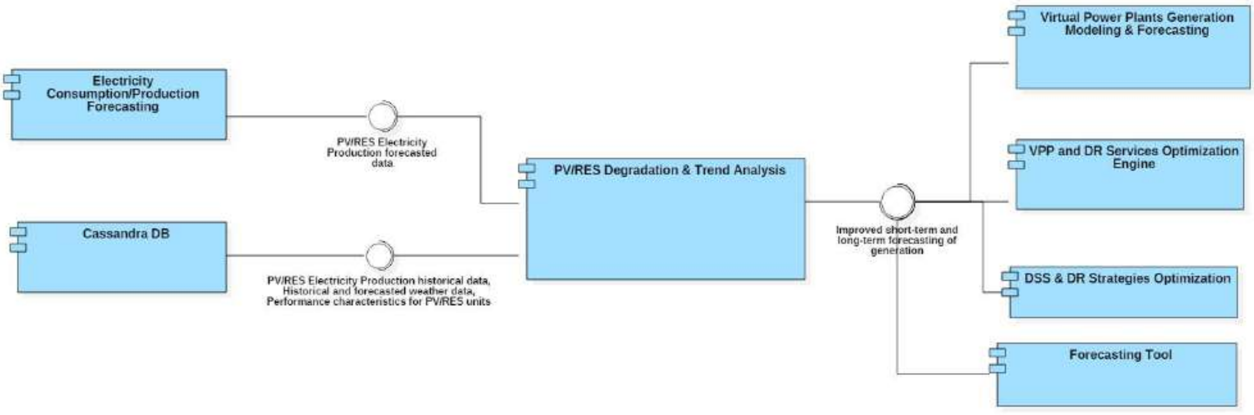


Figure 7: PV/RES Degradation & Trend Analysis

4.3.3 Baseline Flexibility Estimation

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 deliverables D3.2 [12] and D3.6 [13].

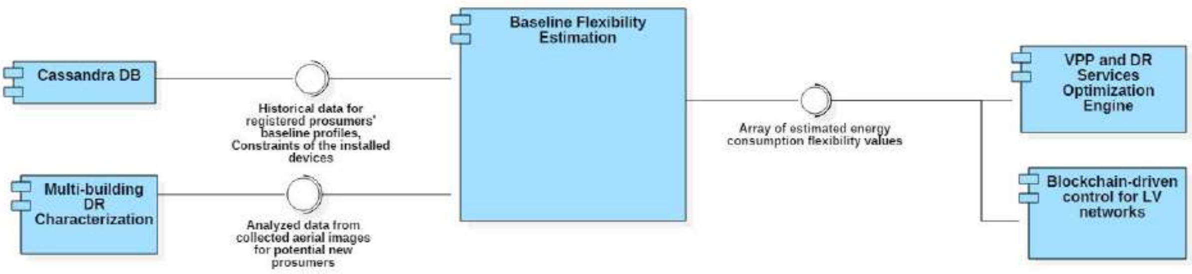


Figure 8: Baseline Flexibility Estimation

4.3.4 Virtual Power Plants Generation Modelling & Forecasting

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 provides 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 (ESS), such as batteries, UPS;
- Flexible Energy Demand Assets (FDA).

In the context of the eDREAM use cases, the VPPs can operate in different types of energy markets targeting the delivery of different types of services. More specifically, the following types of markets have been considered:

- Ancillary services market;
- Wholesale electricity market;
- Balancing market.

The models that have been developed and the detailed description of the services that can be provided by the VPPs are included in the deliverables D3.3 [14] and D3.7 [15].

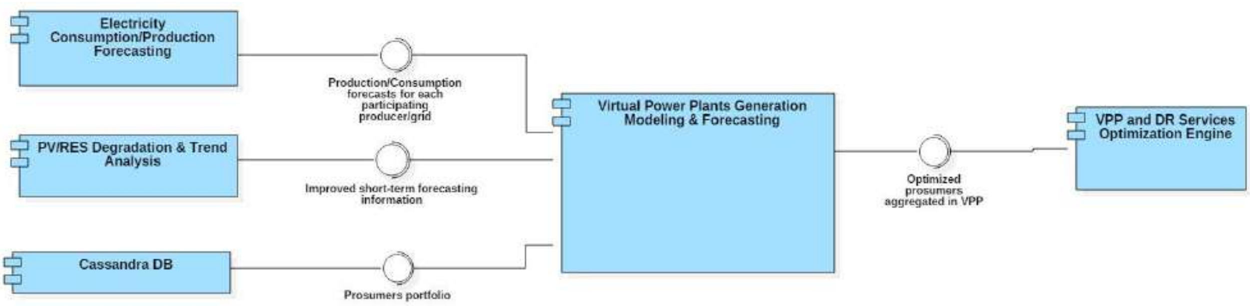


Figure 9: Virtual Power Plants Generation Modelling & Forecasting

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

Main Description – Functionalities

The functionality of this component is related to the estimation of the demand response potential over a wide area of building assets based on the energy demand and generation profile assessment and the overall energy performance of the buildings through optical, thermal and LIDAR images. Appropriate drones (UAVs) are to be equipped with HD optical, thermal imaging and LIDAR scanners. The drones are programmed to fly a fixed path over a specified asset area at different times, in order to collect high-resolution, visible spectrum colour images and infrared thermal images, recorded by the UAVs for the specific building(s) and at a specific date and time. The collected data are used as input to an image processing component, whose goal is to provide an estimate of the building's thermal leakage levels. Thermal leakage for an audit at any given date and time is computed by the image processing system with respect to the leakage levels of an audit that is used as the baseline. Thermal leakage levels are correlated with energy demand requirements, so their estimate becomes useful in the context of DR estimation. The methodology to be followed, the technical specifications for the development and the process for data analysis and DR estimation have been described in the deliverable D3.4 [16] and even further in D3.8 [17].

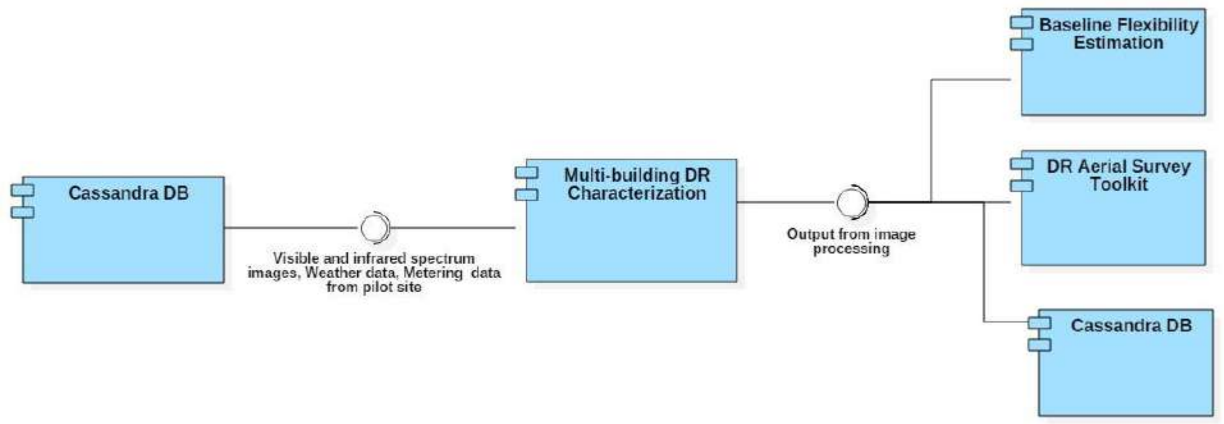


Figure 10: Multi-building DR Characterization through optical, thermal and LIDAR information fusion

4.4 Next generation DR Services for Aggregators and Customers

4.4.1 Load Profiling

Description – Main Functionalities

This component is responsible for detecting load profile patterns and extracting prosumer's load profile based on historical data about load consumption. The identified profile patterns provide necessary information for the grouping of customers, in order to facilitate the selection of the suitable DR program to be applied (e.g. price-based programs, incentives, ToU etc.). More specifically, the historical data of prosumers' consumption is divided into working days and holydays and into different seasons. Afterwards, data from different categories (seasonal and working/holidays) are aggregated in order to extract common profiles. Finally, comparison of different profiles calculated across the considered seasons and days (working/holidays) categorization is performed. The algorithmic process for this component and the corresponding results are described in the deliverables D4.2 [18] and D4.6 [19].

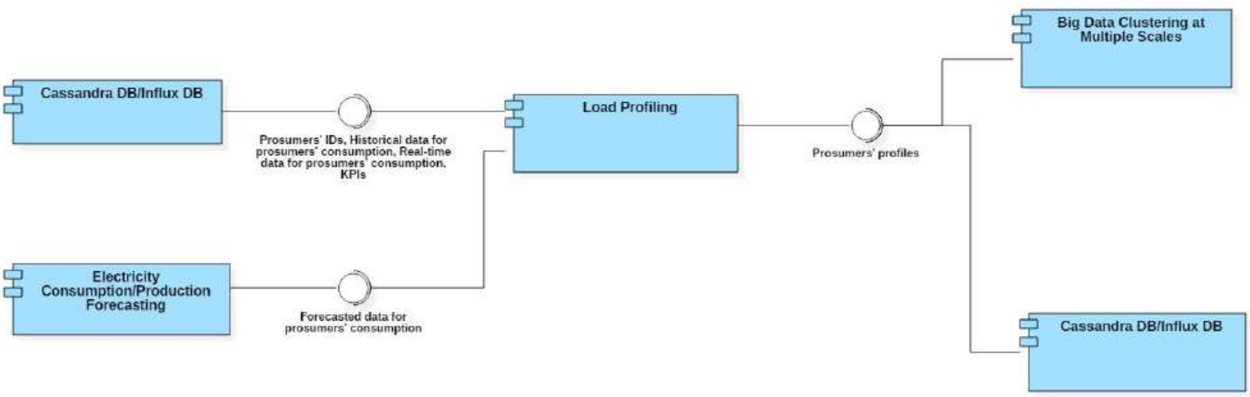


Figure 11: Load Profiling

4.4.2 Big Data Clustering at Multiple Scales

Description – Main Functionalities

The functionality of this component is to provide several well-defined and separate clusters of prosumers. This is a scalable procedure for extracting several 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).

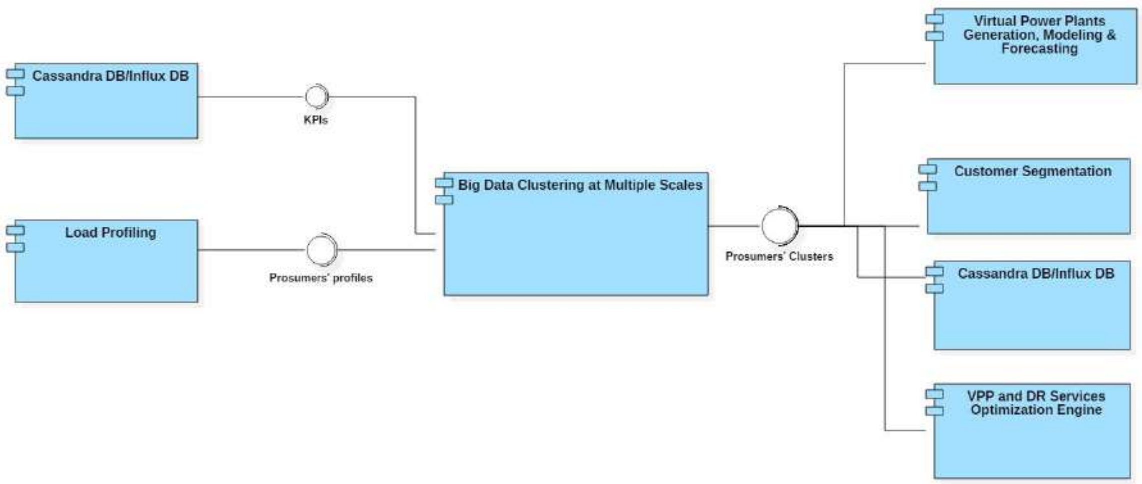


Figure 12: Big Data Clustering at Multiple Scales

4.4.3 Customers Segmentation

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 is 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 deliverables D4.2 [18] and D4.6 [19].

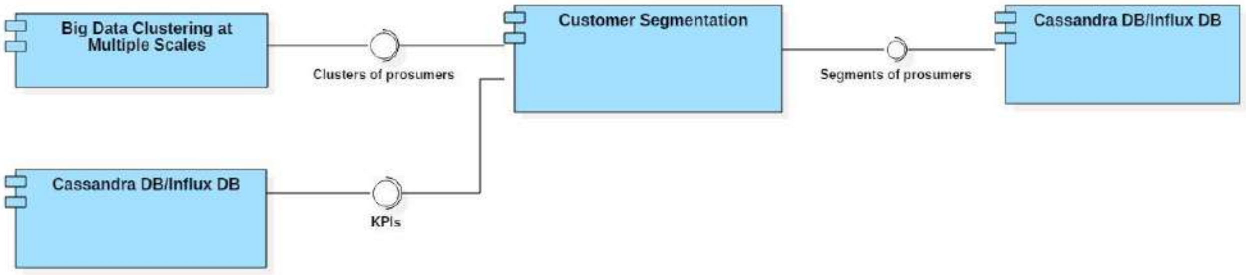


Figure 13: Customer Segmentation

4.4.4 VPP & DR Services Optimization Engine

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 helps 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 deliverables D4.1 [20] and D4.5 [11].

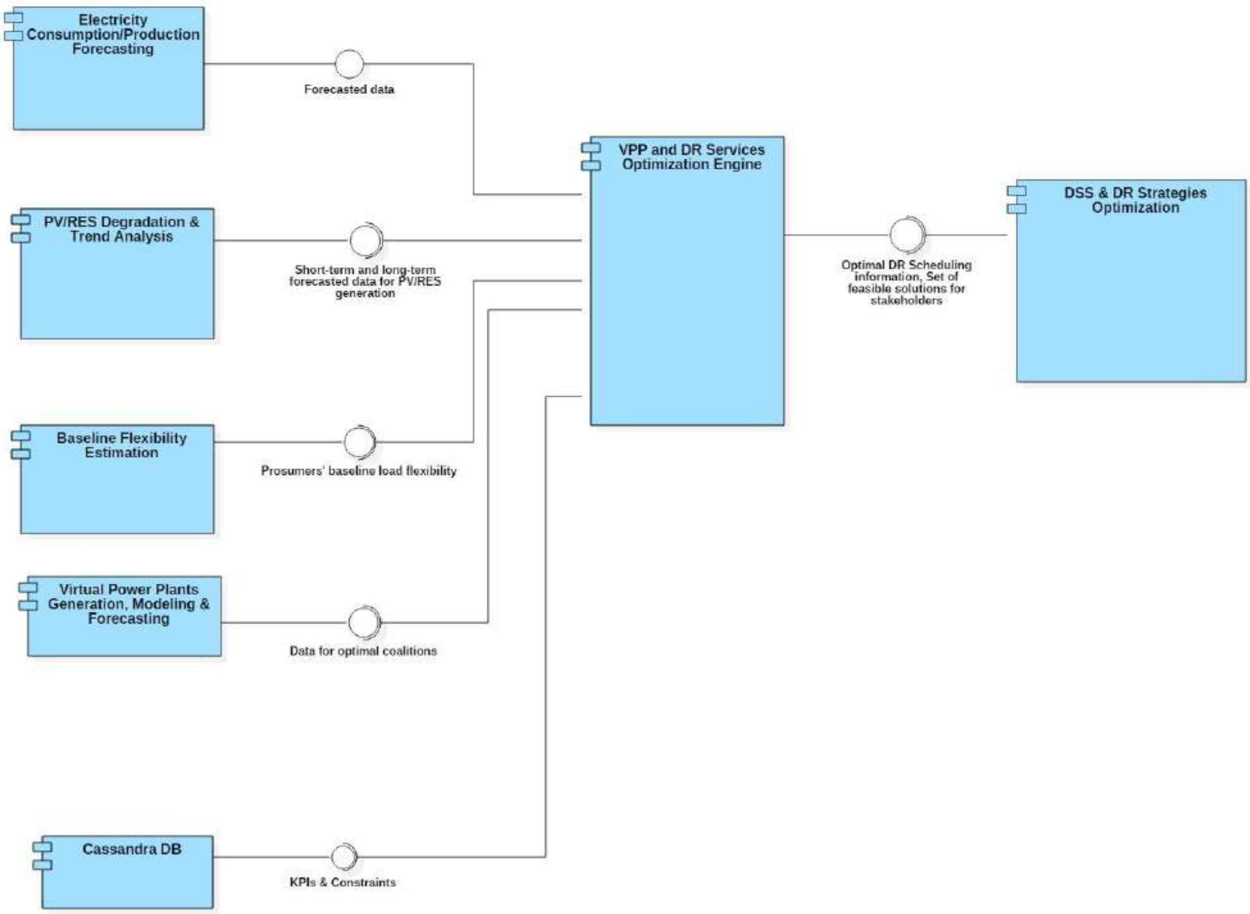


Figure 14: VPP and DR Services Optimization Engine

4.5 Blockchain-enabled Decentralized Network Control Optimization and DR Verification

4.5.1 Distributed Ledger

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 are 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 maintains a copy of the ledger which is 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 deliverables D5.1 [21] and D5.4 [22].

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.

4.5.2 Blockchain-driven control for LV networks (flexibility management)

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) are 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 deliverables D5.2 [23] and D5.5 [24].

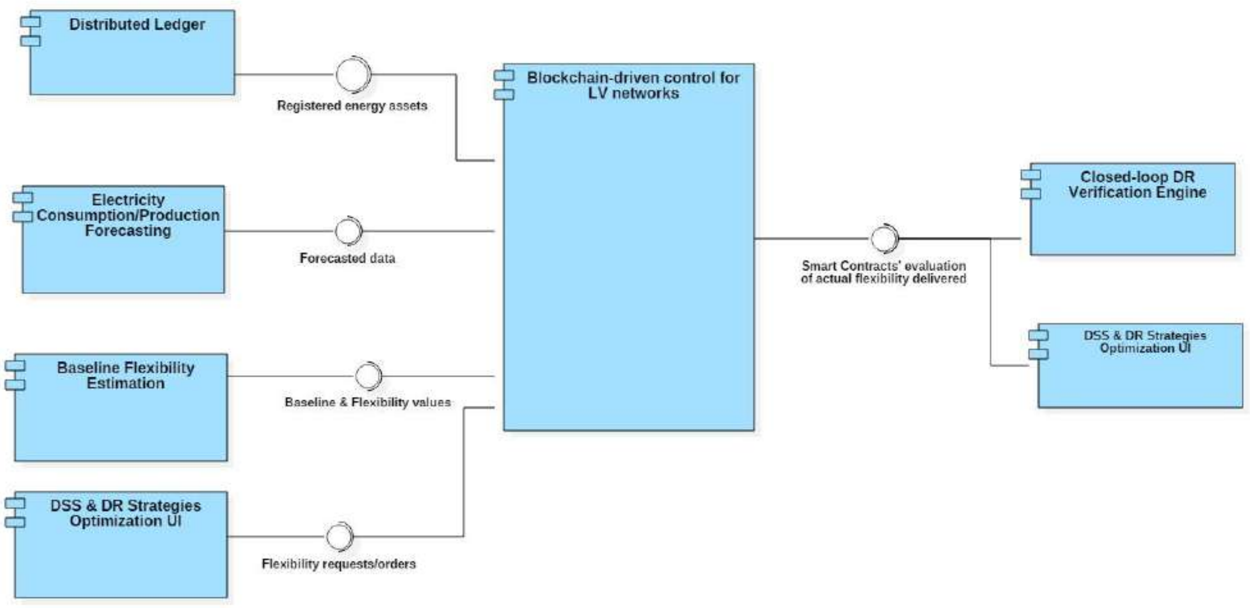


Figure 15: Blockchain-driven control for LV networks

4.5.3 Blockchain-driven Energy Market

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 are 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 is to 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 implementation of this component is reported in the deliverables D5.2 [23] and D5.5 [24]. 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.

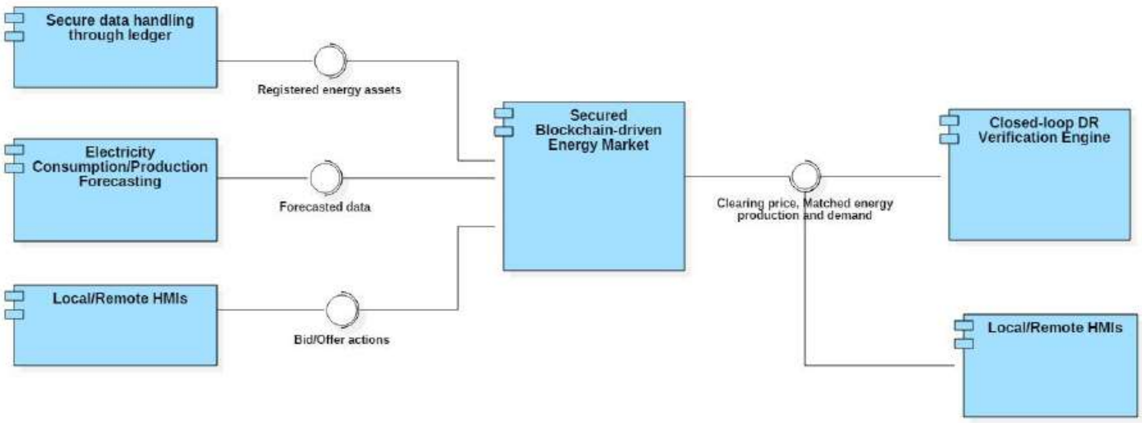


Figure 16: Secured Blockchain-driven Energy Market

4.5.4 Closed loop DR Verification Engine

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 are billed or remunerated accordingly. The verification process is 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 is 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 have been examined. Distributed consensus algorithms are 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 are specialized. The process randomization is 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.

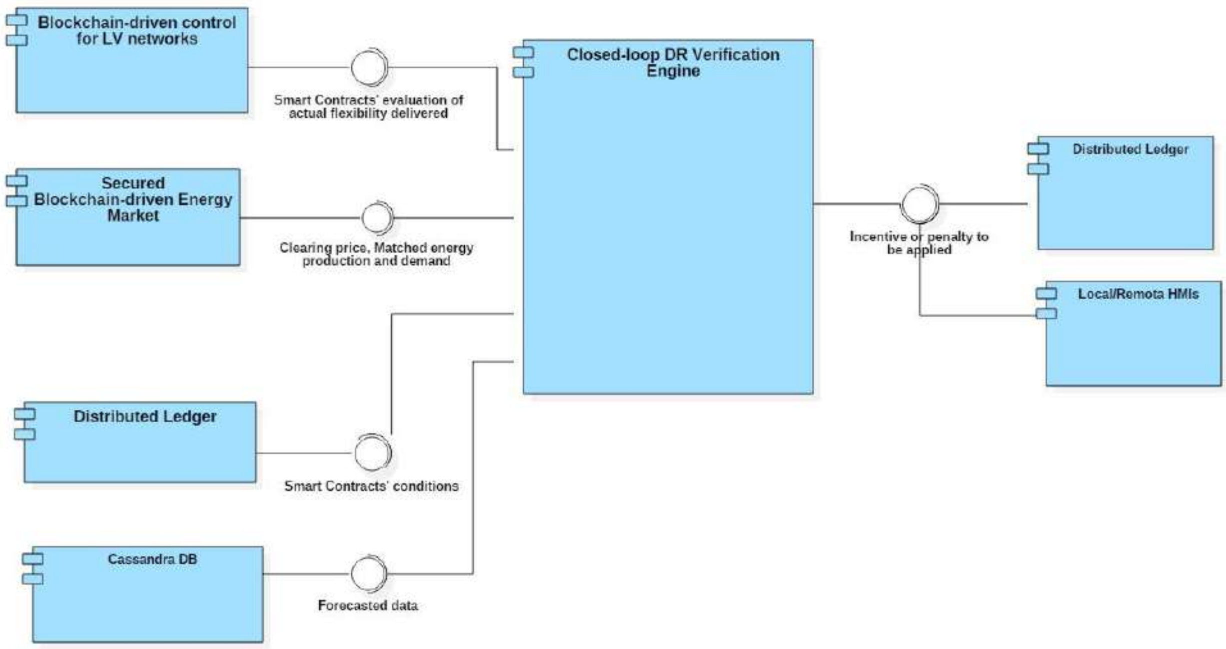


Figure 17: Closed-loop DR Verification Engine

4.6 Multi-level and Multi-factor Visualization Framework (Front-End)

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 are to 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:

- HMIs
  - 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

4.6.1 HMIs

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 are to 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 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;
- Configuration of the desired high-level strategies.

More specifically, the HMIs can be separated into:

Multi-purpose Dashboards:
These components have been 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.

4.6.2 Suites

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 and enables the system stakeholders to maximize the benefits of the applied DR strategies. The suite visualizes optimal DR scheduling for the appropriate management of the flexible resources and provide the stakeholders with a set of feasible solutions, in order to support them in the decision making process. This component has been developed based on the information received from the VPP & DR Services Optimization Engine considering also other business and environmental factors that should be visualized.

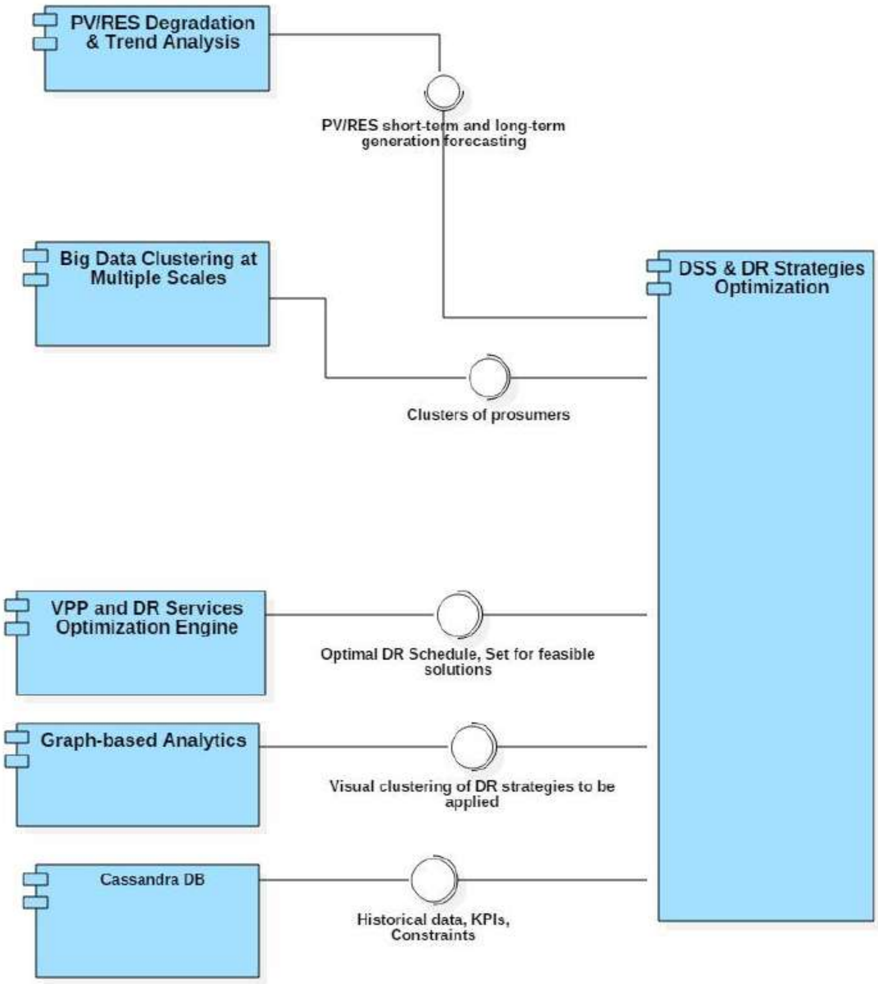


Figure 18: DSS & DR Strategies Optimization

- **DR Aerial Survey Toolkit:** This tool communicates with the Multi-building DR characterization component, in order to receive the analysed data from the collected images and provide insights to aggregators for the new potential customers' capability. This suite visualizes the outputs of the image processing, in order to provide the aggregator an estimation of the energy profile of the potential new prosumers. The methodology for the image processing and the expected results are presented in the deliverable D3.4 [16].

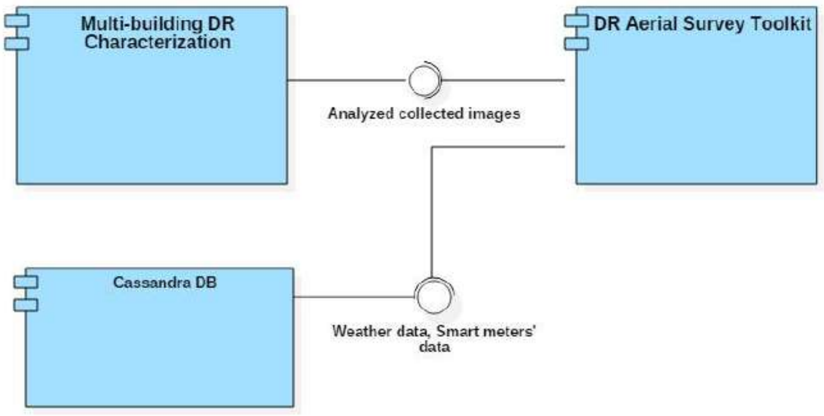


Figure 19: DR Aerial Survey Toolkit

- **Forecasting Tool:** This tool receives signals and information from the Electricity Consumption/Production component, in order 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 is based on the work done in deliverables D3.1 [9] and D3.5 [10].

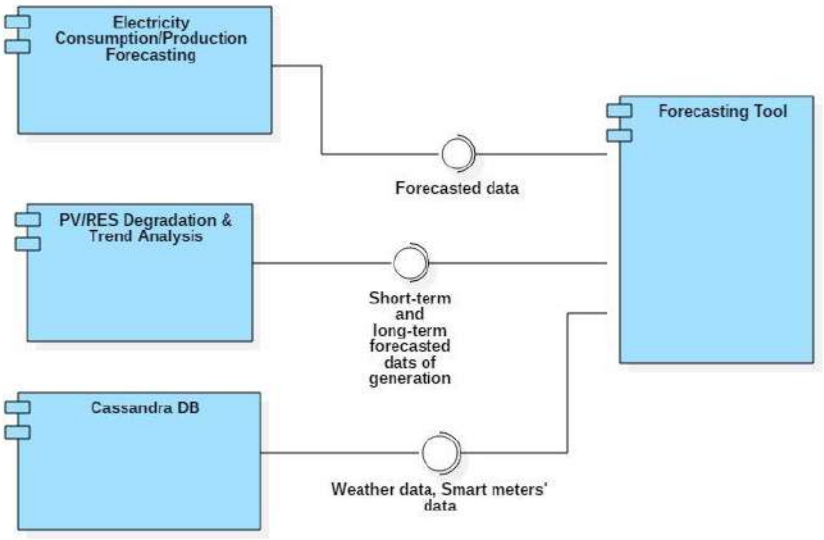


Figure 20: Forecasting Tool

- **Graph-based Analytics:** This suite corresponds to the development of a Graph Analytics platform, which enables 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 is 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 suite are reported in the deliverables D4.3 [25] and D4.4 [26] and their second versions, i.e. D4.7 [27] and D4.8 [28], respectively. 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.

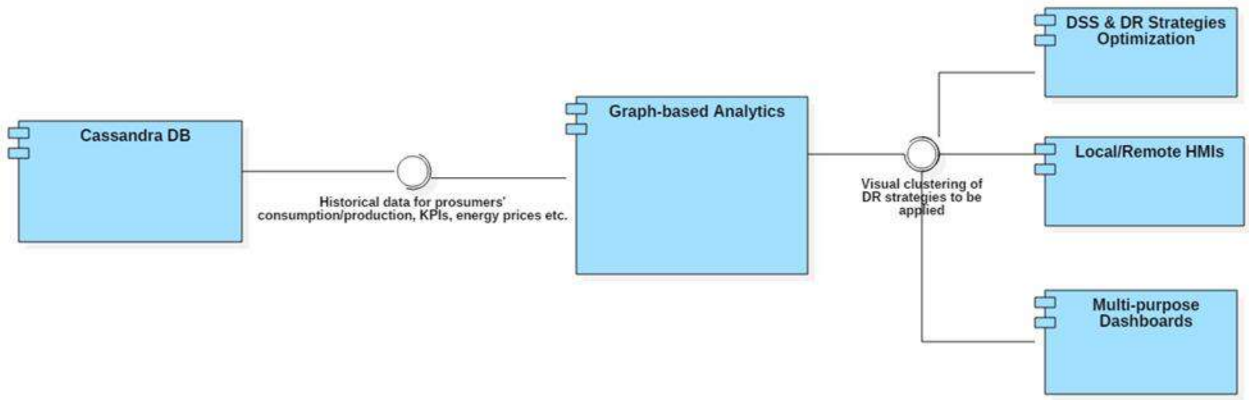


Figure 21: Graph-based Analytics

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 defines 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.

5 Dynamic View

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 analysed in deliverable "D2.2 Use Case Analysis and application scenarios description V1" [4]; in addition they were further refined in D2.5 [2] as well as in D2.7 [7], the final version of the UC has been produced during Task 2.2 and shared in the related deliverable, namely, D2.9 "D2.9 Use Case Analysis and application scenarios description V3" [8]. Use Cases are reported also in this document for the sake of completeness. 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** [29], [30], [31] 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 [1], where the sequence diagrams have been introduced. In this version, the final 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.

5.1 High Level UC 01: Prosumers DR flexibility aggregation via smart contract

The defined UC 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 day-ahead or intraday potential issues on the grid (e.g. congestion and reverse flow), a flexibility request is sent to the aggregator through the marketplace. Thanks to a preliminary assessment of his customers' portfolio, the aggregator is able to evaluate or keep updated the general flexibility capacity of his prosumers through novel techniques and technologies. 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 for the DR event 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.

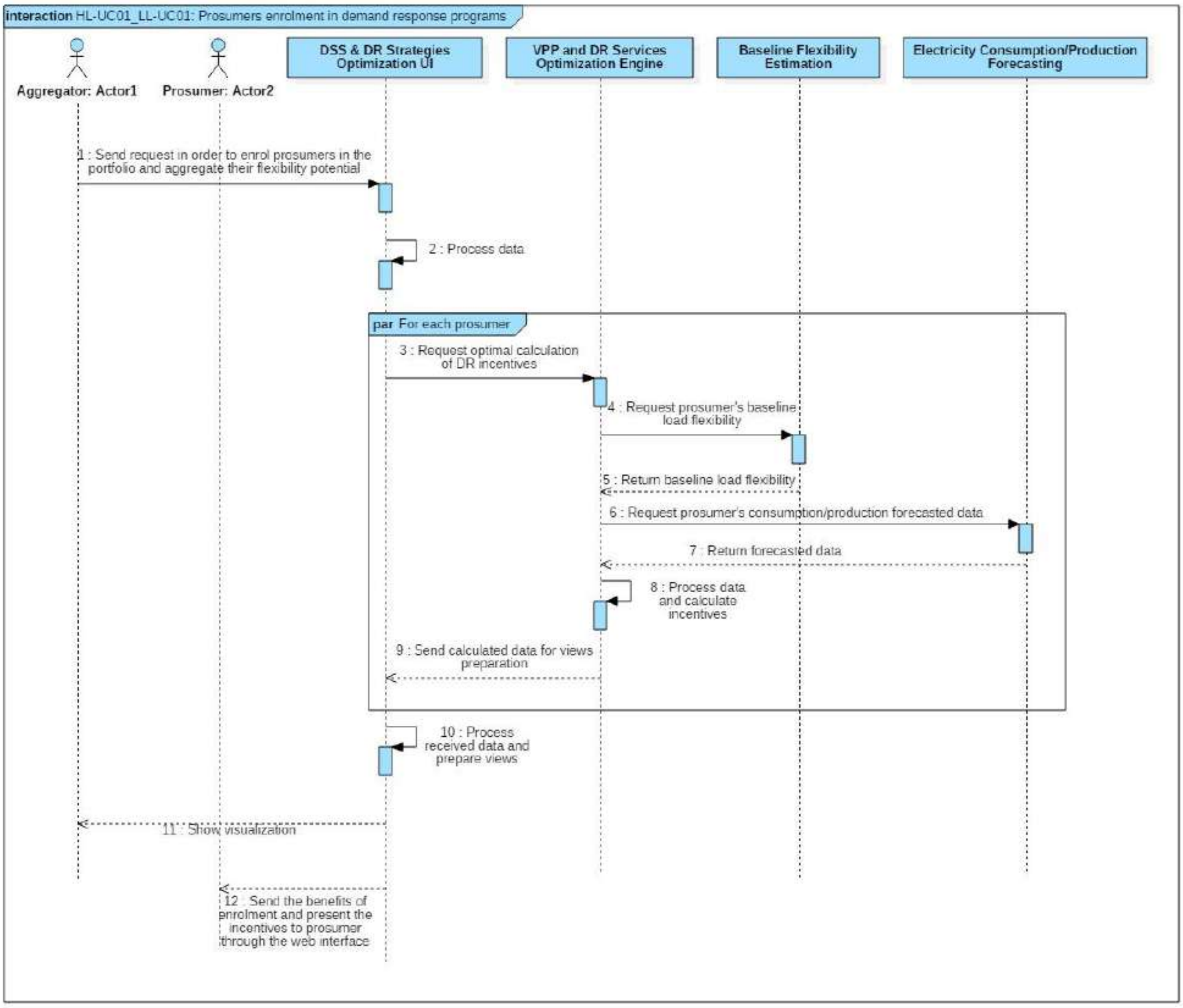
5.1.1 HL-UC01_LL-UC01: Prosumers enrolment in demand response programs

Table 4: HL-UC01_LL-UC01: Prosumers enrolment in demand response programs

Generic Description	
UC Name	**HL-UC01_LL-UC01: Prosumers enrolment in demand response programs**
Version	V0.5
Authors	E@W, TUC
Last Update	1st Version in D2.2 2nd Version in D2.4 3rd Version in D2.5 4th Version in D2.7
Brief Description	Aggregator negotiates with his/her customers (prosumers) showing the benefits through interactive multi-purpose visualization tool for user interaction. 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 an interactive visualization interface, potential incentives are sent to prosumers. These incentives are formulated mainly by taking into account the "Customer Baseline Load (CBL)" through the component of Baseline Flexibility Estimation and forecasted data about consumption and production.
Assumptions and Pre-Conditions	A web interface is available for the Aggregator. The interconnection of the necessary components has been established, thus the decision support system can have access to prosumer's baseline load data.
Goal (Successful End Condition)	Establish a mechanism for enrolling prosumers and make available their flexibility.
Post-Conditions	Prosumers are enrolled.
Involved Actors	Aggregator, Prosumers
UC Initiation	The aggregator needs to enrol prosumers for flexibility availability.
Main Flow	Begin

	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. 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 receives the information about the incentives for the prosumers 12. The aggregator, though the web interface of DSS (Decision Support System) & DR Strategies Optimization, sends the incentives to prosumer.
Alternative Courses	-
Relationships with other UCs	HL-UC01_LL-UC02, HL-UC01_LL-UC03
Architectural Elements / Services Involved	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	
Relevance to eDREAM WPs *Main Tasks Involved* *Main Technical Partners Involved*	WP3 & WP4 T3.1, T3.2, T4.1, T4.3 & T4.4 CERTH, TU, ATOS, TUC, E@W & EMOT
Notes (Optional)	-
UML Sequence Diagram	



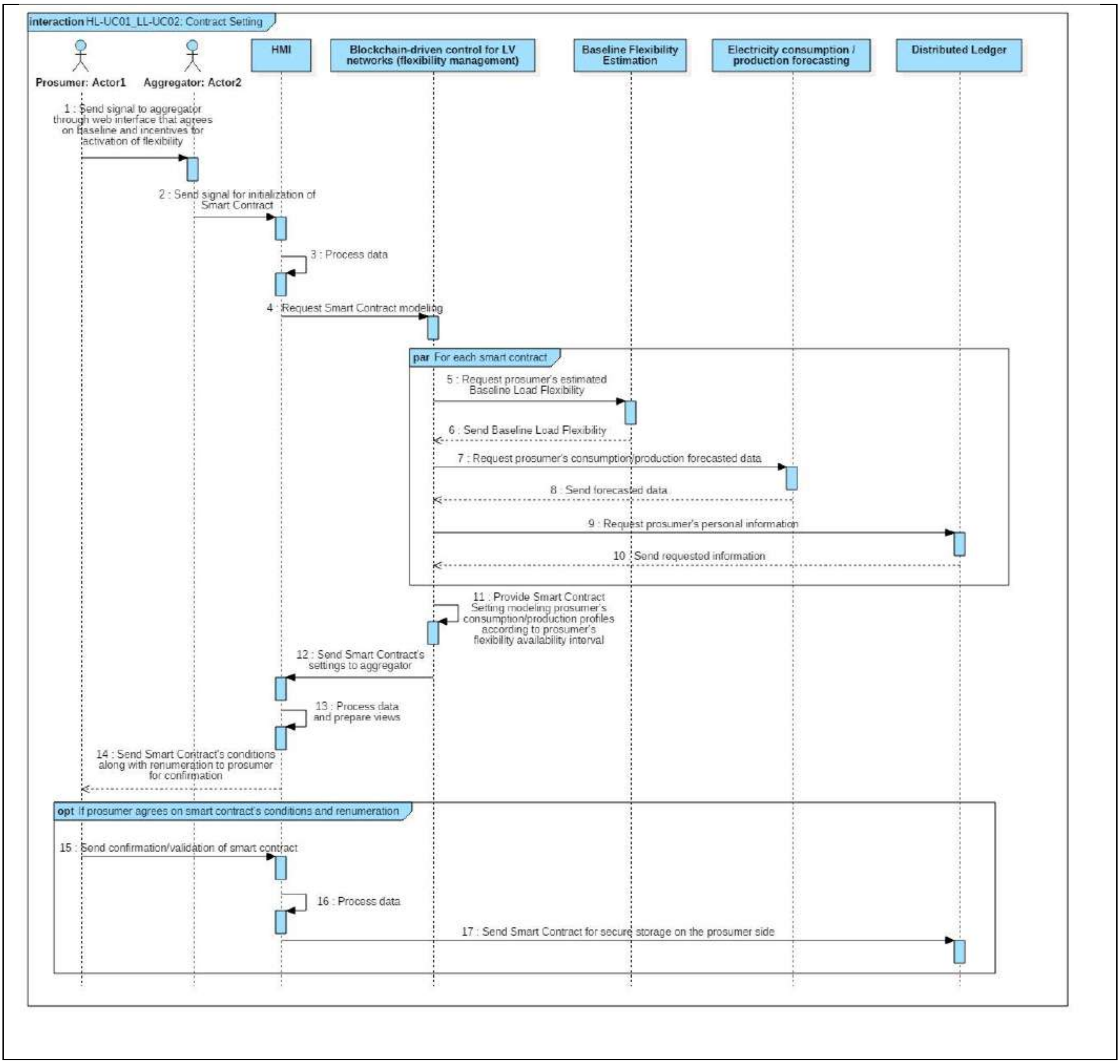
5.1.2 HL-UC01_LL-UC02: Contract Setting

Table 5: HL-UC01_LL-UC02: Contract Setting

Generic Description	
UC 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 3rd Version in D2.5 4th Version in D2.7
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).
Assumptions and Pre-Conditions	A web interface should be available for the prosumer. The smart metering energy devices can properly communicate with the eDREAM platform
Goal (Successful End Condition)	The aim of this UC is the setting of transactions' conditions between the aggregator and prosumers.
Post-Conditions	Prosumers activate their flexibility availability interval in case of request.
Involved Actors	Aggregator, Prosumers
UC Initiation	Aggregator and prosumers agree on baseline and incentives for flexibility activation.
Main Flow	Begin 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 4. The HMI sends request to the Baseline Flexibility Estimation to receive the prosumers estimated baseline loads. 5. The HMI receives the prosumers estimated baseline loads. 6. The HMI sends request to the Electricity Consumption/Production forecasting to receive the prosumers consumption/production forecasted data. 7. The HMI receives the prosumers consumption/production forecasted data. 8. The HMI processes the aggregator request 9. The HMI sends the request to the component Blockchain-driven control for LV networks (flexibility management), in order to model the smart contract. 10. The prosumer's estimated baseline load flexibility from the component Baseline Flexibility Estimation and the prosumer's consumption/production forecasted data from the Electricity Consumption/Production Forecasting component are injected in the Blockchain-driven control for LV networks smart contract 11. The component Blockchain-driven control for LV networks sets the conditions for smart contract modelling. 12. The HMI receives information about the smart contracts modelled by the Blockchain-driven control. 13. The HMI processes the received data and prepares views for the aggregator. 14. The aggregator sends the smart contract's conditions along with the remuneration to prosumer for confirmation through web interface of the HMI. 15. The prosumer sends the confirmation/validation of smart contract to aggregator's HMI through web interface. 16. The HMI receives and processes the input data.

	17. The HMI sends smart contracts for secure storage on the blockchain distributed ledger
Alternative Courses	Some prosumers do not agree on baseline and incentives for activation of flexibility (initialization of Smart Contract).
Relationships with other UCs	HL-UC01_LL-UC01
Architectural Elements / Services Involved	HMI; Blockchain-driven control for LV networks; Baseline Flexibility Estimation; Electricity consumption/production forecasting; Distributed Ledger;
Specific Description	
Relevance to eDREAM WPs *Main Tasks Involved* *Main Technical Partners Involved*	WP3, WP4 & WP5 T3.1, T3.2, T4.3, T5.1 & T5.2 TUC, ENG, E@W, CERTH, TU, ATOS
Notes (Optional)	-
UML Sequence Diagram	

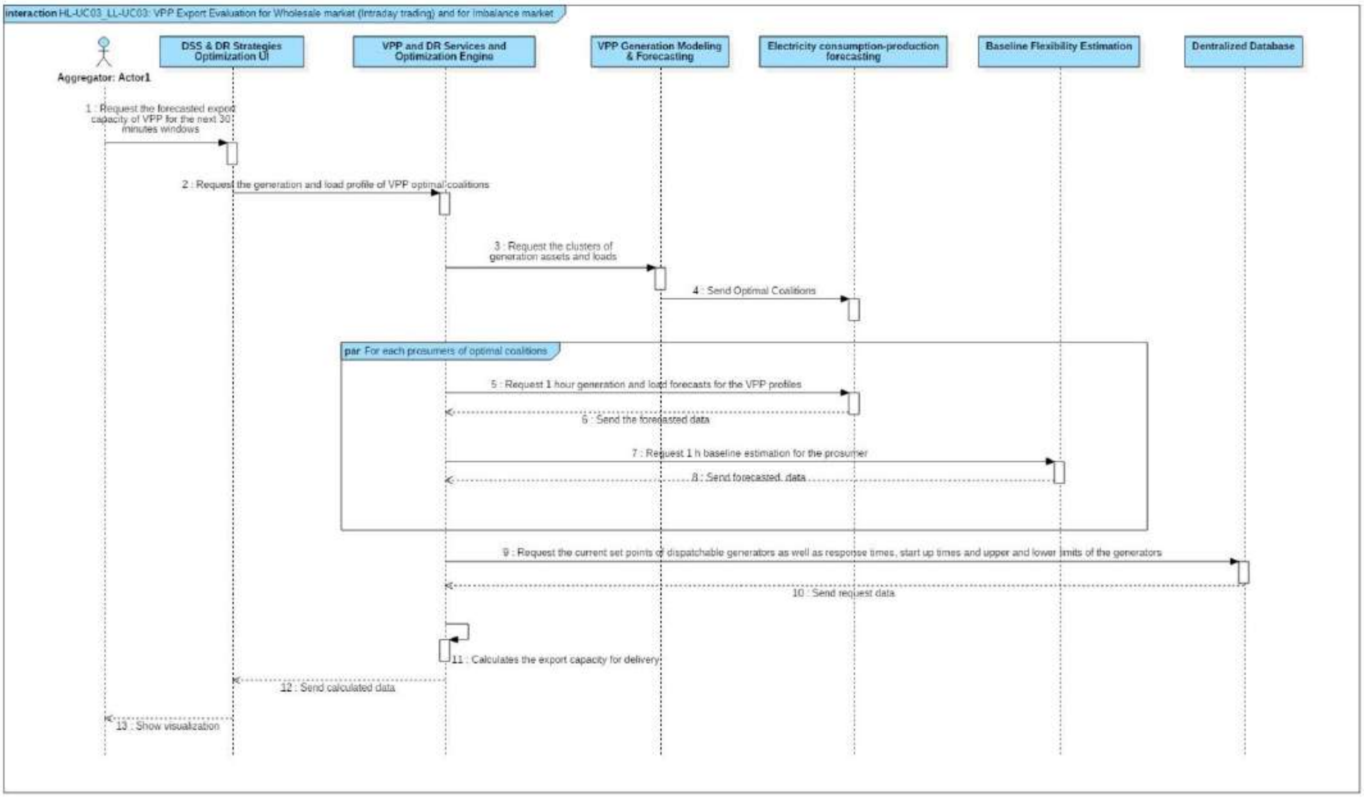

interaction HL-UC01_LL-UC02: Contract Setting
Prosumer: Actor1
Aggregator: Actor2
HMI
Blockchain-driven control for LV networks (flexibility management)
Baseline Flexibility Estimation
Electricity consumption / production forecasting
Distributed Ledger
1 : Send signal to aggregator through web interface that agrees on baseline and incentives for activation of flexibility
2 : Send signal for initialization of Smart Contract
3 : Process data
4 : Request Smart Contract modeling
par For each smart contract
5 : Request prosumer's estimated Baseline Load Flexibility
6 : Send Baseline Load Flexibility
7 : Request prosumer's consumption/production forecasted data
8 : Send forecasted data
9 : Request prosumer's personal information
10 : Send requested information
11 : Provide Smart Contract Setting modeling prosumer's consumption/production profiles according to prosumer's flexibility availability interval
12 : Send Smart Contract's settings to aggregator
13 : Process data and prepare views
14 : Send Smart Contract's conditions along with renumeration to prosumer for confirmation
opt If prosumer agrees on smart contract's conditions and renumeration
15 : Send confirmation/validation of smart contract
16 : Process data
17 : Send Smart Contract for secure storage on the prosumer side

5.1.3 HL-UC01_LL-UC03: Potential energy flexibility evaluation

Table 6: HL-UC01_LL-UC03: Potential energy flexibility evaluation

Generic Description	
UC Name	**HL-UC01_LL-UC03: Potential energy flexibility evaluation**
Version	V0.5
Authors	ENG, TUC, E@W
Last Update	1st Version in D2.2 2nd Version in D2.4 3rd Version in D2.5 4th Version in D2.7
Brief Description	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.
Assumptions and Pre-Conditions	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 appropriately with the eDREAM platform.
Goal (Successful End Condition)	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.
Post-Conditions	The aaggregator identifies new qualified prosumers. Concerning the registered prosumers, the aggregator receives a quick estimation of their flexibility after a long period of time.
Involved Actors	Aggregator, Prosumers
UC Initiation	The aggregator sends request for flexibility evaluation.
Main Flow	Begin 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. 4. The component Multi-building DR characterization collects optical, thermal and/or LIDAR images over the specified area and analyses them. 5. The Multi-building DR characterization sends the analysed 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 through user interface.
Alternative Courses	Other methods are used for the identification of new prosumers. The flexibility evaluation of registered prosumers is only performed through metering devices and suitable calculations.
Relationships with other UCs	HL-UC01_LL-UC01
Architectural Elements / Services Involved	DR Aerial Survey Toolkit UI; Multi-building DR characterization through thermal, optical and LIDAR information fusion; Baseline Flexibility Estimation;
Specific Description	

Relevance to eDREAM WPs	WP3
Main Tasks Involved	T3.2 & T3.4
Main Technical Partners Involved	TU, CERTH, KIWI, E@W, TUC
Notes (Optional)	-
UML Sequence Diagram	
	

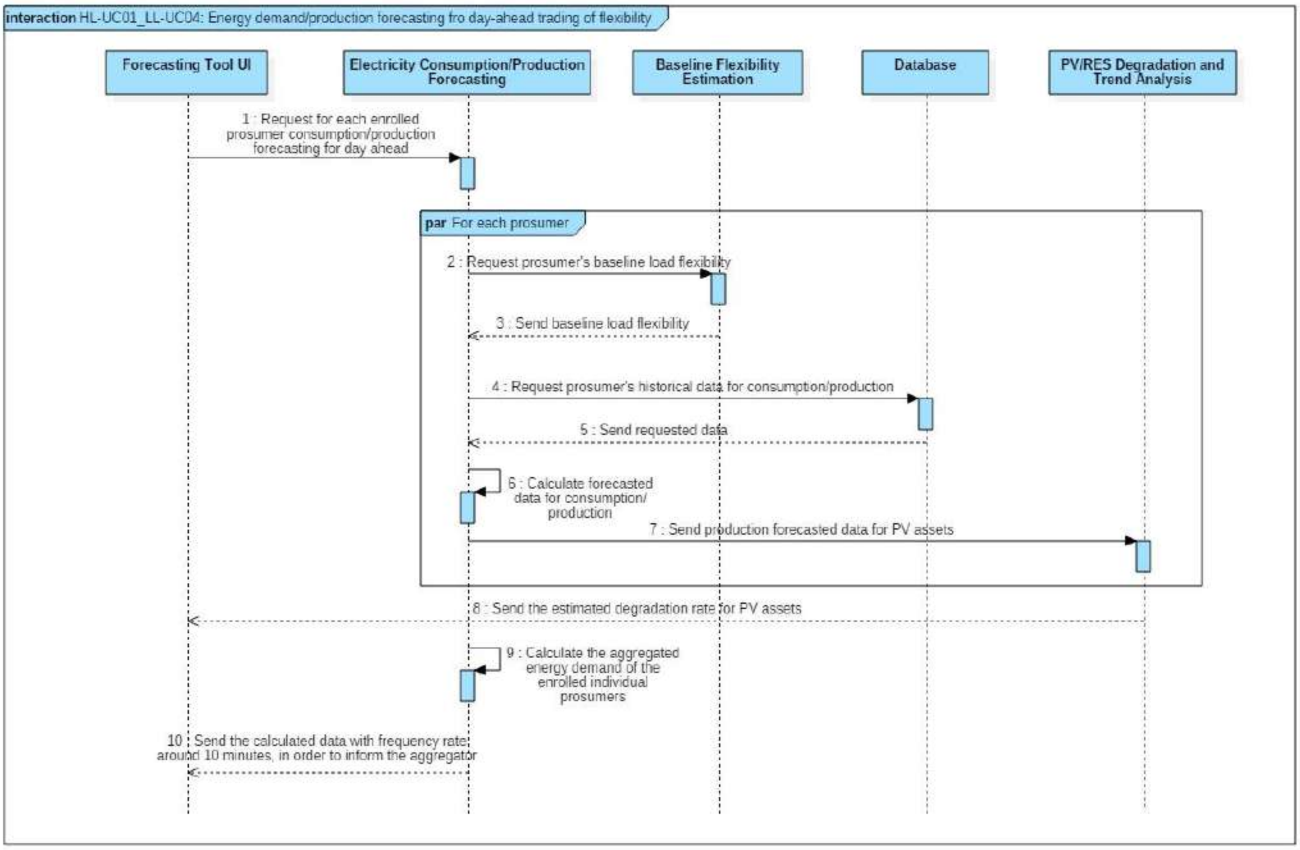
5.1.4 HL-UC01_LL-UC04: Energy demand/production forecasting for day-ahead trading of flexibility

Table 7: HL-UC01_LL-UC04: Energy demand/production forecasting for day-ahead trading of flexibility

Generic Description	
UC Name	**HL-UC01_LL-UC04: Energy demand/production forecasting for day-ahead trading of flexibility**
Version	V0.5
Authors	ENG, TUC, E@W

Last Update	1st Version in D2.2 2nd Version in D2.4 3rd Version in D2.5 4th Version in D2.7
Brief Description	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 of 1 hour for the day-ahead timeframe, or 30 minutes for the intraday timeframe.
Assumptions and Pre-Conditions	Prosumers are enrolled with the aggregator. Prosumer's historical data about consumption/production are available.
Goal (Successful End Condition)	The aim of this UC is to obtain forecasted values for production/consumption at individual prosumer level.
Post-Conditions	The Aggregator can inform the DSO about the total forecasted energy demand of all the registered prosumers.
Involved Actors	Aggregator, Prosumers, DSO
UC Initiation	The UC is automatically initiated as Forecasting Tool UI have to present consumption/production forecasting with frequency rate around 10 minutes.
Main Flow	Begin 1. The Forecasting Tool UI requests for each prosumer enrolled consumption/production forecasted data for day ahead with a frequency rate of 1 hour/30 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.

	6. The Electricity Consumption/Production Forecasting calculates consumption/production forecasted data for each prosumer. 7. The Electricity Consumption/Production Forecasting sends production forecasted data for PV assets to PV Degradation and Trend Analysis. 8. The PV Degradation and Trend Analysis sends the estimated degradation rate for PV assets to the Forecasting tool UI. 9. The Electricity Consumption/Production Forecasting calculates the aggregated energy demand of the enrolled individual prosumers. 10. The Electricity Consumption/Production Forecasting sends the calculated data with frequency rate of 1 hour/ 30 minutes, in order to inform the aggregator. 11. The aggregator visualizes the data through the Forecasting Tool UI
Alternative Courses	-
Relationships with other UCs	HL-UC01_LL-UC05
Architectural Elements / Services Involved	Forecasting Tool UI; Electricity Consumption/Production Forecasting; Baseline Flexibility Estimation; Decentralized Repository; PV/RES Degradation and Trend Analysis;
Specific Description	
Relevance to eDREAM WPs *Main Tasks Involved* *Main Technical Partners Involved*	WP3 & WP4 T3.1, T3.2 & T4.1 TUC, TU, E@W, CERTH, ENG
Notes (Optional)	-
UML Sequence Diagram	


interaction HL-UC01_LL-UC04: Energy demand/production forecasting fro day-ahead trading of flexibility
Forecasting Tool UI
Electricity Consumption/Production Forecasting
Baseline Flexibility Estimation
Database
PV/RES Degradation and Trend Analysis
1 : Request for each enrolled prosumer consumption/production forecasting for day ahead
par For each prosumer
2 : Request prosumer's baseline load flexibility
3 : Send baseline load flexibility
4 : Request prosumer's historical data for consumption/production
5 : Send requested data
6 : Calculate forecasted data for consumption/ production
7 : Send production forecasted data for PV assets
8 : Send the estimated degradation rate for PV assets
9 : Calculate the aggregated energy demand of the enrolled individual prosumers
10 : Send the calculated data with frequency rate around 10 minutes, in order to inform the aggregator

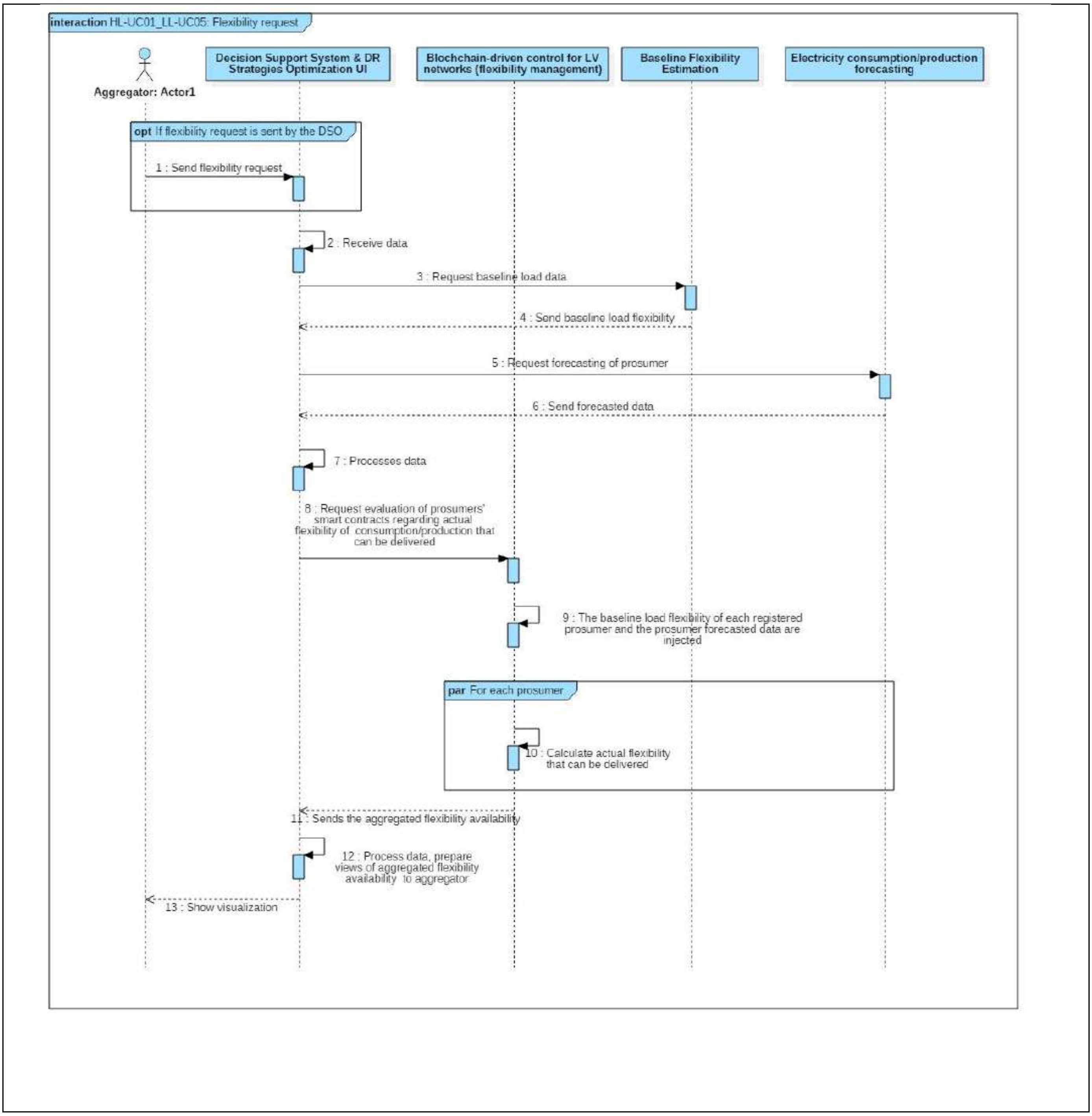
5.1.5 HL-UC01_LL-UC05: Flexibility request

Table 8: HL-UC01_LL-UC05: Flexibility request

Generic Description	
UC 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 3rd Version in D2.5 4th Version in D2.7
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.
Assumptions and Pre-Conditions	Congestion point/s has/have been forecasted by the DSO. The Aggregator is properly informed by the DSO. The Aggregator's UI is connected with eDREAM core platform.
Goal (Successful End Condition)	Through this UC, the Aggregator aims to define the actual aggregated flexibility that can deliver to DSO.
Post-Conditions	Aggregator's total flexibility offer is accepted by the DSO.
Involved Actors	Aggregator, DSO
UC Initiation	The UC is initiated by the Aggregator in case of previous request by the DSO (forecasted congestion point/s).
Main Flow	Begin 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 the input data.

	3. The Decision Support System & DR Strategies Optimization sends request to the Baseline Flexibility Estimation to receive the prosumers estimated baseline loads. 4. The Decision Support System & DR Strategies Optimization receives the prosumers estimated baseline loads. 5. The Decision Support System & DR Strategies Optimization sends request to the Electricity Consumption/Production forecasting to receive the prosumers consumption/production forecasted data. 6. The Decision Support System & DR Strategies Optimization receives the prosumers consumption/production forecasted data. 7. The Decision Support System & DR Strategies Optimization processes the input data. 8. 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). 9. The baseline load flexibility of each registered prosumer and the prosumer's consumption/production forecasted data from the component Electricity consumption/production forecasting are injected in the smart contracts of the Blockchain-driven control for LV networks component. 10. The Blockchain-driven control for LV networks calculates the actual aggregated flexibility that can be delivered. 11. The Decision Support System & DR Strategies Optimization receives the aggregated flexibility availability from the Blockchain-driven control for LV networks smart contracts 12. The Decision Support System & DR Strategies Optimization processes data and prepares views of total flexibility availability to aggregator. 13. The DSS (Decision Support System) & DR Strategies Optimization UI shows the data to the Aggregator
Alternative Courses	DSO applies traditional methods of peak loads reduction.
Relationships with other UCs	HL-UC01_LL-UC04, HL-UC01_LL-UC06

Architectural Elements / Services Involved	Decision Support System & DR Strategies Optimization UI; Blockchain-driven control for LV networks (flexibility management); Baseline Flexibility Estimation; Electricity Consumption/Production Forecasting;
Specific Description	
Relevance to eDREAM WPs *Main Tasks Involved* *Main Technical Partners Involved*	WP3, WP4 & WP5 T3.1, T3.2, T4.1, T4.3 & T5.2 TUC, TU, E@W, CERTH, ENG, ATOS
Notes (Optional)	-
UML Sequence Diagram	


interaction HL-UC01_LL-UC05: Flexibility request
Aggregator: Actor1
Decision Support System & DR Strategies Optimization UI
Blochchain-driven control for LV networks (flexibility management)
Baseline Flexibility Estimation
Electricity consumption/production forecasting
opt If flexibility request is sent by the DSO
1 : Send flexibility request
2 : Receive data
3 : Request baseline load data
4 : Send baseline load flexibility
5 : Request forecasting of prosumer
6 : Send forecasted data
7 : Processes data
8 : Request evaluation of prosumers' smart contracts regarding actual flexibility of consumption/production that can be delivered
9 : The baseline load flexibility of each registered prosumer and the prosumer forecasted data are injected
par For each prosumer
10 : Calculate actual flexibility that can be delivered
11 : Sends the aggregated flexibility availability
12 : Process data, prepare views of aggregated flexibility availability to aggregator
13 : Show visualization

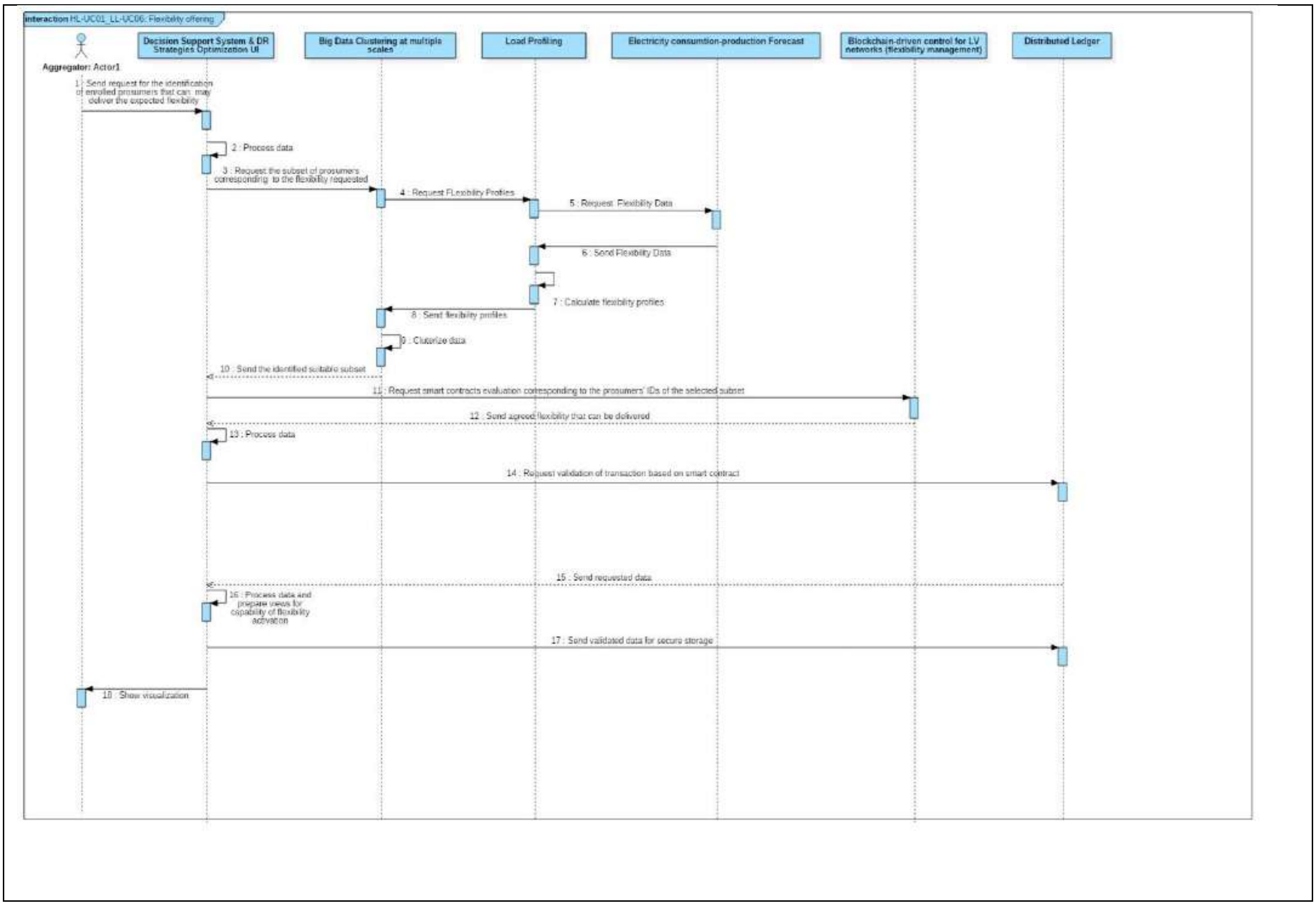
5.1.6 HL-UC01_LL-UC06: Flexibility offering

Table 9: HL-UC01_LL-UC06: Flexibility offering

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
Brief Description	Based on the received flexibility request, the Aggregator inquires its enrolled prosumers to identify the subset which may deliver the expected flexibility.
Assumptions and Pre-Conditions	The Aggregator had received flexibility request by the DSO. The Smart Contracts' conditions are available and accessible.
Goal (Successful End Condition)	Through this UC, the Aggregator aims to define the subset of enrolled prosumers that may deliver the expected flexibility.
Post-Conditions	The Aggregator is ready for delivering flexibility in case the DSO sends the flexibility order.
Involved Actors	Aggregator, Prosumers
Use Case Initiation	The UC is initiated by the Aggregator in order to identify the suitable subset of prosumers for expected flexibility delivery.
Main Flow	Begin 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 Load Profiling requests flexibility data from the Electricity Consumption/Production Forecasting component. 6. The Electricity Consumption/Production Forecasting sends the requested data. 7. The Load Profiling calculates flexibility profiles according to received petition. 8. The Load Profiling sends customers' flexibility profiles to the Big Data Clustering. 9. The Big Data Clustering clusterizes the received profiles considering the requested criteria. 10. The Big Data Clustering at multiple scale sends the identified subset to the Decision Support System & DR Strategies Optimization UI. 11. The Decision Support System & DR Strategies Optimization UI requests smart contracts evaluation of the prosumers belonging to the identified subsets. 12. The Blockchain-driven control for LV networks returns agreed flexibility that can be delivered. 13. The Decision Support System & DR Strategies Optimization UI processes the received data. 14. The Decision Support System & DR Strategies Optimization UI sends request for validation of transaction to the Distributed Ledger smart contracts. 15. The Distributed Ledger sends the received data to the Decision Support System & DR Strategies Optimization. 16. The Decision Support System & DR Strategies Optimization UI processes received data and prepares views for capability of flexibility activation to the aggregator. 17. The Decision Support System & DR Strategies Optimization UI sends the validated data for secure storage to the component Distributed Ledger.

	18. Show visualization
Alternative Cources	The identified prosumers do not agree to deliver their flexibility.
Relationships with other Use Cases	HL-UC01_LL-UC05, HL-UC01_LL-UC07
Architectural Elements / Services Involved	Decision Support System & DR Strategies Optimization UI; Big Data Clustering at multiple scales; Load Profiling; Electricity consumption/production forecast; Blockchain-driven control for LV networks (flexibility management); Closed loop DR Verification Engine; Distributed Ledger;
Specific Description	
Relevance to eDREAM WPs *Main Tasks Involved* *Main Technical Partners Involved*	WP4 & WP5 T4.1, T4.2, T3.1, T5.1, T5.2 & T5.3 TU, ENG, E@W, TUC, EMOT
Notes (Optional)	-
UML Sequence Diagram	

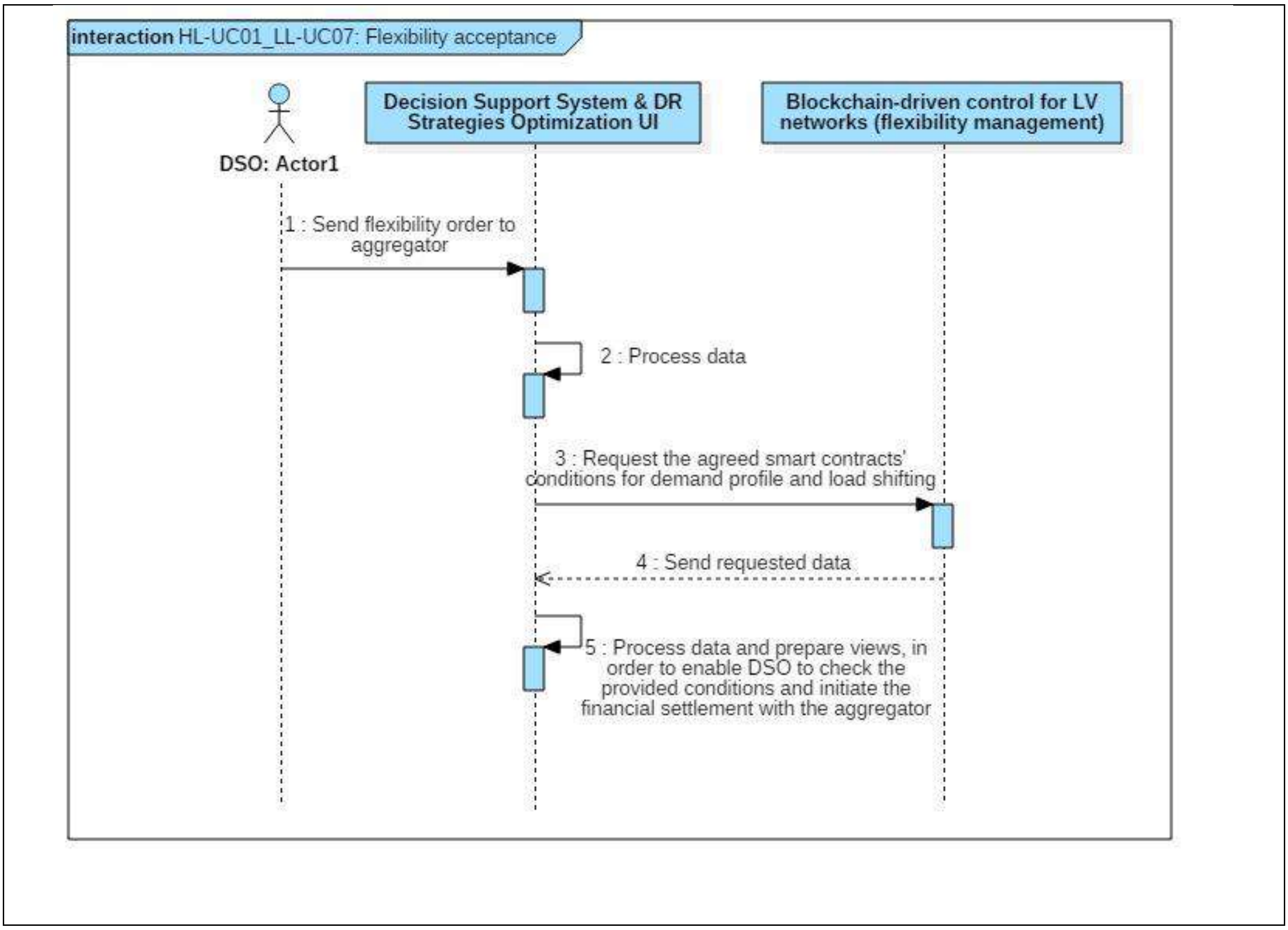

interaction HL-UC01_LL-UC06: Flexibility offering
Aggregator: Actor1
Decision Support System & DR Strategies Optimization UI
Big Data Clustering at multiple scales
Load Profiling
Electricity consumtion-production Forecast
Blockchain-driven control for LV networks (flexibility management)
Distributed Ledger
1 : Send request for the identification of enrolled prosumers that can may deliver the expected flexibility
2 : Process data
3 : Request the subset of prosumers corresponding to the flexibility requested
4 : Request FLexibility Profiles
5 : Request Flexibility Data
6 : Send Flexibility Data
7 : Calculate flexibility profiles
8 : Send flexibility profiles
9 : Cluterize data
10 : Send the identified suitable subset
11 : Request smart contracts evaluation corresponding to the prosumers' IDs of the selected subset
12 : Send agreed flexibility that can be delivered
13 : Process data
14 : Request validation of transaction based on smart contract
15 : Send requested data
16 : Process data and prepare views for capability of flexibility activation
17 : Send validated data for secure storage
18 : Show visualization

5.1.7 HL-UC01_LL-UC07: Flexibility acceptance

Table 10: HL-UC01_LL-UC07: Flexibility acceptance

Generic Description	
UC 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 3rd Version in D2.5 4th Version in D2.7
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.
Goal (Successful End Condition)	The DSO aims to inform properly the Aggregator that his/her flexibility offer/s are accepted.
Post-Conditions	The Aggregator is ready for initiating the flexibility provisioning.
Involved Actors	DSO, Aggregator
UC Initiation	The Aggregator's flexibility offers are successfully accepted by the DSO.
Main Flow	Begin 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 Blockchain-driven control for LV networks component the demanded profile and flexibility requests to be delivered by each prosumer.

	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.
Alternative Courses	The Aggregator's flexibility offer/s are rejected by the DSO.
Relationships with other UCs	HL-UC01_LL-UC06, HL-UC01_LL-UC08
Architectural Elements / Services Involved	Decision Support System & DR Strategies Optimization UI; Blockchain-driven control for LV networks (flexibility management);
Specific Description	
Relevance to eDREAM WPs *Main Tasks Involved* *Main Technical Partners Involved*	WP4 & WP5 T4.1, T4.3, T4.4 & T5.2 TUC, TU, CERTH, ENG, ATOS, E@W, EMOT
Notes (Optional)	-
UML Sequence Diagram	


interaction HL-UC01_LL-UC07: Flexibility acceptance
DSO: Actor1
Decision Support System & DR Strategies Optimization UI
Blockchain-driven control for LV networks (flexibility management)
1 : Send flexibility order to aggregator
2 : Process data
3 : Request the agreed smart contracts' conditions for demand profile and load shifting
4 : Send requested data
5 : Process data and prepare views, in order to enable DSO to check the provided conditions and initiate the financial settlement with the aggregator

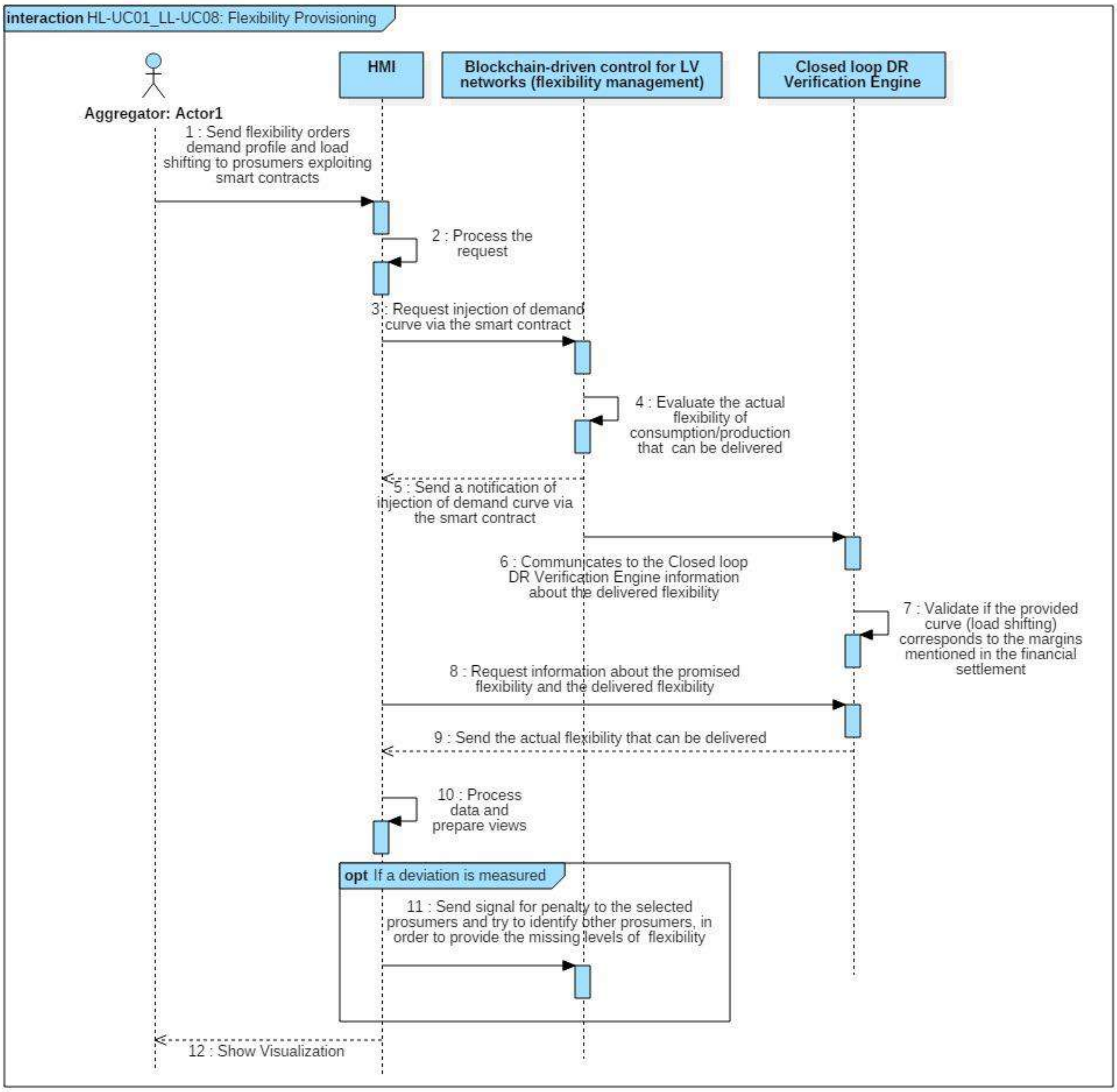
5.1.8 HL-UC01_LL-UC08: Flexibility provisioning

Table 11: HL-UC01_LL-UC08: Flexibility provisioning

Generic Description	
UC 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 3rd Version in D2.5 4th Version in D2.7
Brief Description	Aggregator sends the flexibility orders to his/her prosumers (injection of demand/supply curve via a smart contract), in order to adjust the load/generation of his/her clients and fulfill 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).
Assumptions and Pre-Conditions	The Aggregator received a flexibility order by the DSO.
Goal (Successful End Condition)	The aim of this UC is to deliver the appropriate agreed flexibility to the DSO.
Post-Conditions	The Aggregator provides the appropriate agreed flexibility to the DSO leveraging on its prosumers.
Involved Actors	Aggregator, Prosumers
UC Initiation	The Aggregator receives a flexibility offer by the DSO.
Main Flow	Begin 1. The Aggregator sends flexibility orders (demand profile and load shifting) to his/her prosumers exploiting the smart contracts. 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 evaluates the actual flexibility of consumption/production that is delivered. Ex 6 5. The HMI receives notification of injection of demand curve via the smart contract from the component Blockchain-driven control for LV networks. 6. The Blockchain-driven control for LV networks communicates to the Closed loop DR Verification Engine information about the delivered flexibility. 7. The Closed loop DR Verification Engine validates if the provided curve corresponds to the flexibility request demanded from the prosumer. The financial settlement is enforced according to the determined deviations. 8. The HMI can request information about the promised flexibility and the delivered flexibility from the Closed Loop DR Verification Engine components. 9. The HMI receives information about the promised flexibility and the delivered flexibility from the Closed Loop DR Verification Engine components. 10. The HMI processes input data and prepare views. 11. If a deviation is measured, 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. 12. The HMI shows visualization.
Alternative Courses	The prosumers do not deliver the ordered amount of flexibility to the aggregator.
Relationships with other UCs	HL-UC01_LL-UC07
Architectural Elements / Services Involved	HMI; Blockchain-driven control for LV networks (flexibility management); Distributed Ledger; Closed loop DR Verification Engine;
Specific Description	

Relevance to eDREAM WPs *Main Tasks Involved* *Main Technical Partners Involved*	WP4 & WP5 T4.3, T5.1, T5.2 & T5.3 TUC, ENG, E@W, EMOT
Notes (Optional)	-
UML Sequence Diagram	



5.2 High Level UC 02: Peer-to-peer local energy trading market

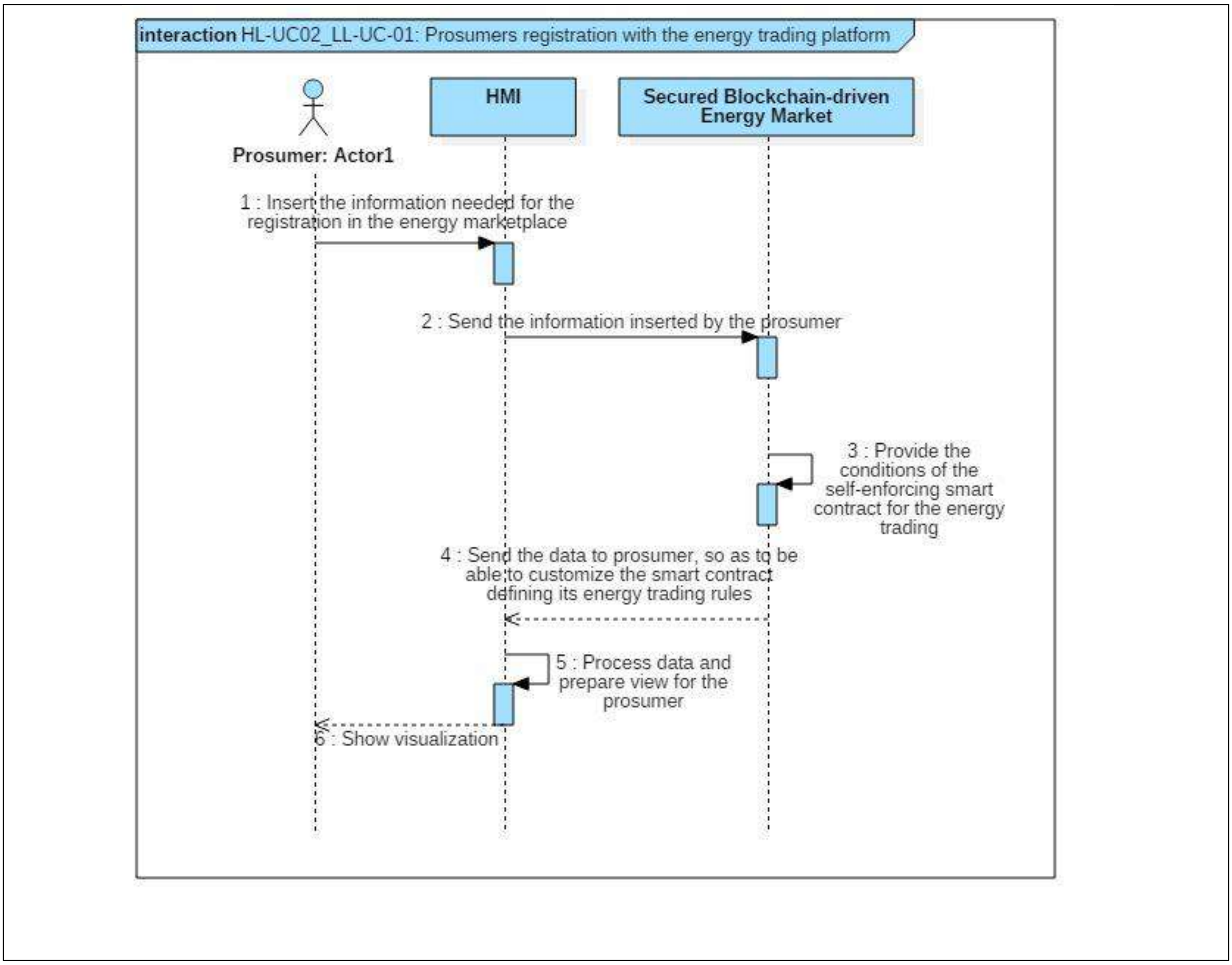
In this scenario, the eDREAM project considers a blockchain based mechanism for decentralized energy trading (price-driven) which enables prosumers to buy or sell energy directly by means of peer-to-peer energy transactions. Prosumers that are willing to buy or sell energy should register with the blockchain based energy trading platform.

5.2.1 HL-UC02_LL-UC01: Prosumers registration with the energy trading platform

Table 12: HL-UC02_LL-UC01: Prosumers registration with the energy trading platform

Generic Description	
UC Name	**HL-UC02_LL-UC01: Prosumers registration with the energy trading platform**
Version	V0.3
Authors	TUC
Last Update	1st Version in D2.2 2nd Version in D2.4 3rd Version in D2.5 4th Version in D2.7
Brief Description	Prosumers register with the peer to peer energy market providing their information that will be validated through self-enforcing smart contracts. 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.
Assumptions and Pre-Conditions	Data about the prosumer identification and energy demand/production is available. The energy trading platform is operational.
Goal (Successful End Condition)	The aim of this UC is to establish a mechanism for enrolling prosumers with the energy trading market.
Post-Conditions	The prosumer is registered with the energy trading market.
Involved Actors	Prosumers
UC Initiation	Prosumer is willing to trade energy with the blockchain based energy market.
Main Flow	Begin

	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 provides the conditions of the self-enforcing smart contract for the energy trading. 4. The HMI requests data from the Secured Blockchain Driven Energy Market, in order to customize the smart contract conditions. 5. The HMI processes the received data and prepare views; 6. The prosumer sees the data through the HMI
Alternative Courses	Prosumers are not willing to buy or sell energy in a decentralized (blockchain based) market.
Relationships with other UCs	HL-UC02_LL-UC02
Architectural Elements / Services Involved	HMI; Secured Blockchain-driven energy market; Distributed Ledger; Closed loop DR Verification Engine;
Specific Description	
Relevance to eDREAM WPs *Main Tasks Involved* *Main Technical Partners Involved*	WP4 & WP5 T4.3, T5.1, T5.2 & T5.3 TUC, ENG, E@W, EMOT
Notes (Optional)	-
UML Sequence Diagram	

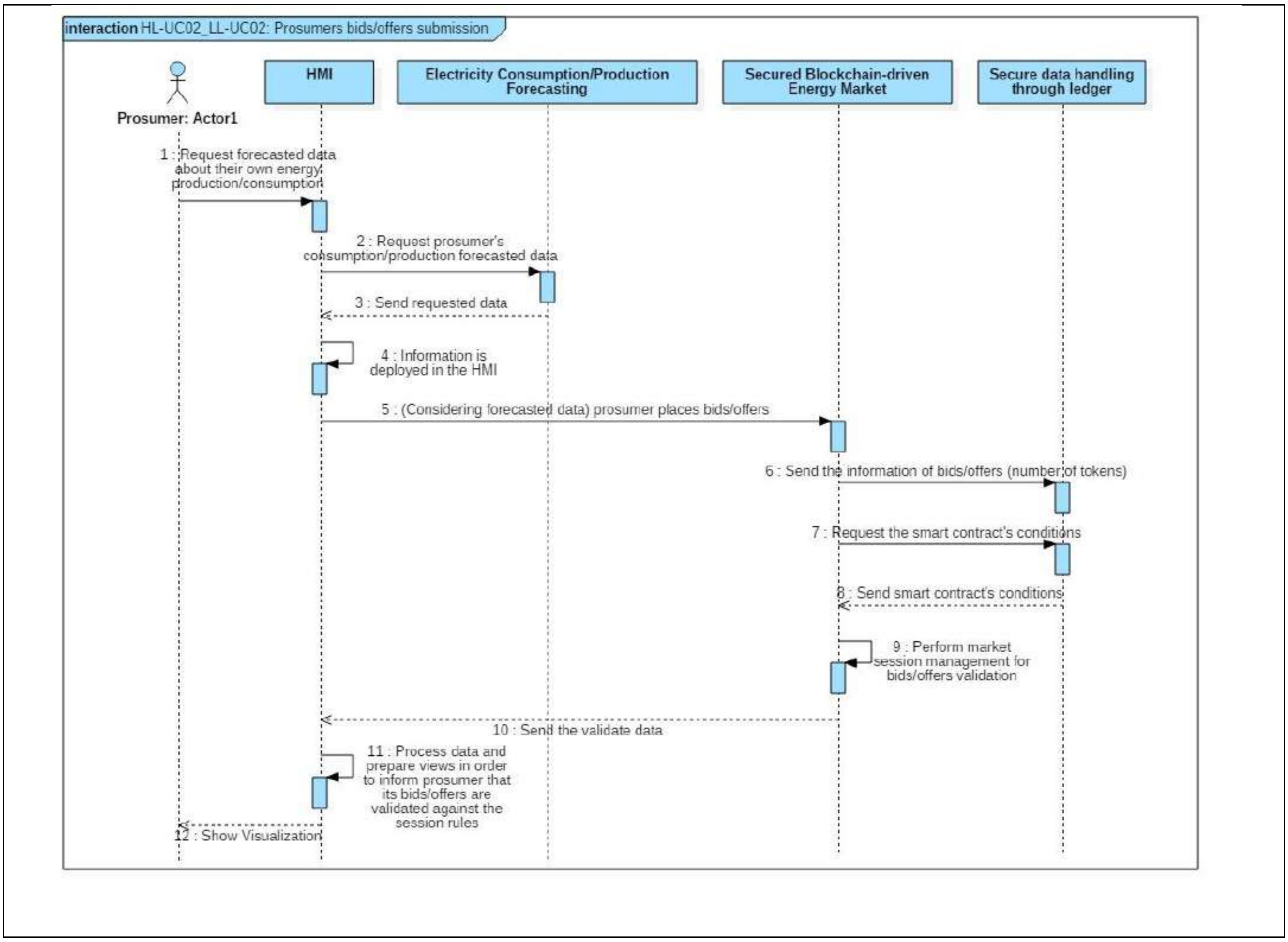

interaction HL-UC02_LL-UC-01: Prosumers registration with the energy trading platform
HMI
Secured Blockchain-driven Energy Market
Prosumer: Actor1
1 : Insert the information needed for the registration in the energy marketplace
2 : Send the information inserted by the prosumer
3 : Provide the conditions of the self-enforcing smart contract for the energy trading
4 : Send the data to prosumer, so as to be able to customize the smart contract defining its energy trading rules
5 : Process data and prepare view for the prosumer
6 : Show visualization

5.2.2 HL-UC02_LL-UC02: Prosumers bids/offers submission

Table 13: HL-UC02_LL-UC02: Prosumers bids/offer submission

Generic Description	
UC 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 3rd Version in D2.5 4th Version in D2.7
Brief Description	Prosumers submit bids/offers associating the number of tokens equivalent with their amount of energy (e.g. 1 token unit = 1 Wh). Prosumers use the energy demand and generation forecasted data to construct the bids/offers for the next market session.
Assumptions and Pre-Conditions	The prosumer is registered with the energy trading market.
Goal (Successful End Condition)	The aim of this UC is to empower the prosumers to submit their bids/offers of energy.
Post-Conditions	The bids/offers are submitted.
Involved Actors	Prosumers
UC Initiation	The prosumer would like to submit energy bids/offers with the blockchain based energy market.
Main Flow	Begin 1. Prosumer get forecasted data about its 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 smart contracts will act as escrows for the financial deposits registered by the prosumers upon registering orders in the market 7. The Secured Blockchain-driven Energy Market performs market session management actions for bids/offers validation. 8. The HMI can request the necessary information and validated data from the Secured Blockchain-driven Energy Market 9. The HMI processes the received data and prepares views, to inform prosumer about the list of validated bids/offers. 10. The prosumer sees the data through the HMI
Alternative Courses	-
Relationships with other UCs	HL-UC02_LL-UC01, HL-UC02_LL-UC03
Architectural Elements / Services Involved	HMI; Electricity Consumption/Production Forecasting; Secured Blockchain-driven energy market; Distributed Ledger;
Specific Description	
Relevance to eDREAM WPs *Main Tasks Involved* *Main Technical Partners Involved*	WP3, WP4 & WP5 T3.1, T4.1, T5.1 & T5.2 TUC, ENG, E@W, EMOT, CERTH, TU
Notes (Optional)	-
UML Sequence Diagram	

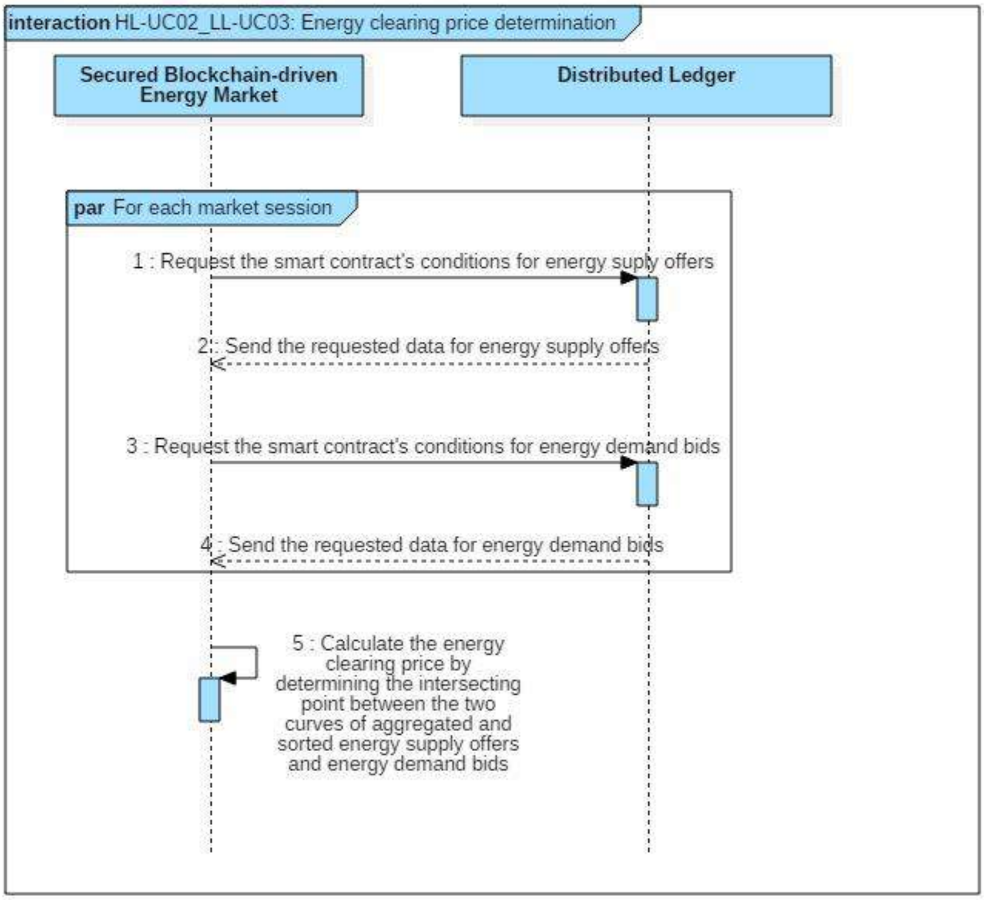

interaction HL-UC02_LL-UC02: Prosumers bids/offers submission
Prosumer: Actor1
HMI
Electricity Consumption/Production Forecasting
Secured Blockchain-driven Energy Market
Secure data handling through ledger
1 : Request forecasted data about their own energy production/consumption
2 : Request prosumer's consumption/production forecasted data
3 : Send requested data
4 : Information is deployed in the HMI
5 : (Considering forecasted data) prosumer places bids/offers
6 : Send the information of bids/offers (number of tokens)
7 : Request the smart contract's conditions
8 : Send smart contract's conditions
9 : Perform market session management for bids/offers validation
10 : Send the validate data
11 : Process data and prepare views in order to inform prosumer that its bids/offers are validated against the session rules
12 : Show Visualization

5.2.3 HL-UC02_LL-UC03: Energy clearing price determination

Table 14: HL-UC02_LL-UC03: Energy clearing price determination

Generic Description	
UC 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 3rd Version in D2.5 4th Version in D2.7
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.
Assumptions and Pre-Conditions	Market session is opened and bids/offers have been submitted.
Goal (Successful End Condition)	The aim of this UC is to determine the energy trading price per market session.
Post-Conditions	The energy trading price is determined.
Involved Actors	-
UC Initiation	End of market session.
Main Flow	Begin 1. The Secured Blockchain-driven Energy Market gathers all the energy supply offers registered by the prosumers during an active market session. 2. The Secured Blockchain-driven Energy Market gathers all the energy demand bids registered by the prosumers during an active market session. 3. 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).

	4. The prosumer smart contracts are notified about the matched orders
Alternative Courses	Other greedy algorithms for clearing price calculations can be used (i.e. graph based).
Relationships with other UCs	HL-UC02_LL-UC02, HL-UC02_LL-UC04
Architectural Elements / Services Involved	Secured Blockchain-driven energy market; Distributed Ledger;
Specific Description	
Relevance to eDREAM WPs *Main Tasks Involved* *Main Technical Partners Involved*	WP5 T5.1 & T5.2 TUC, ENG, E@W, EMOT, CERTH, TU
Notes (Optional)	-
UML Sequence Diagram	

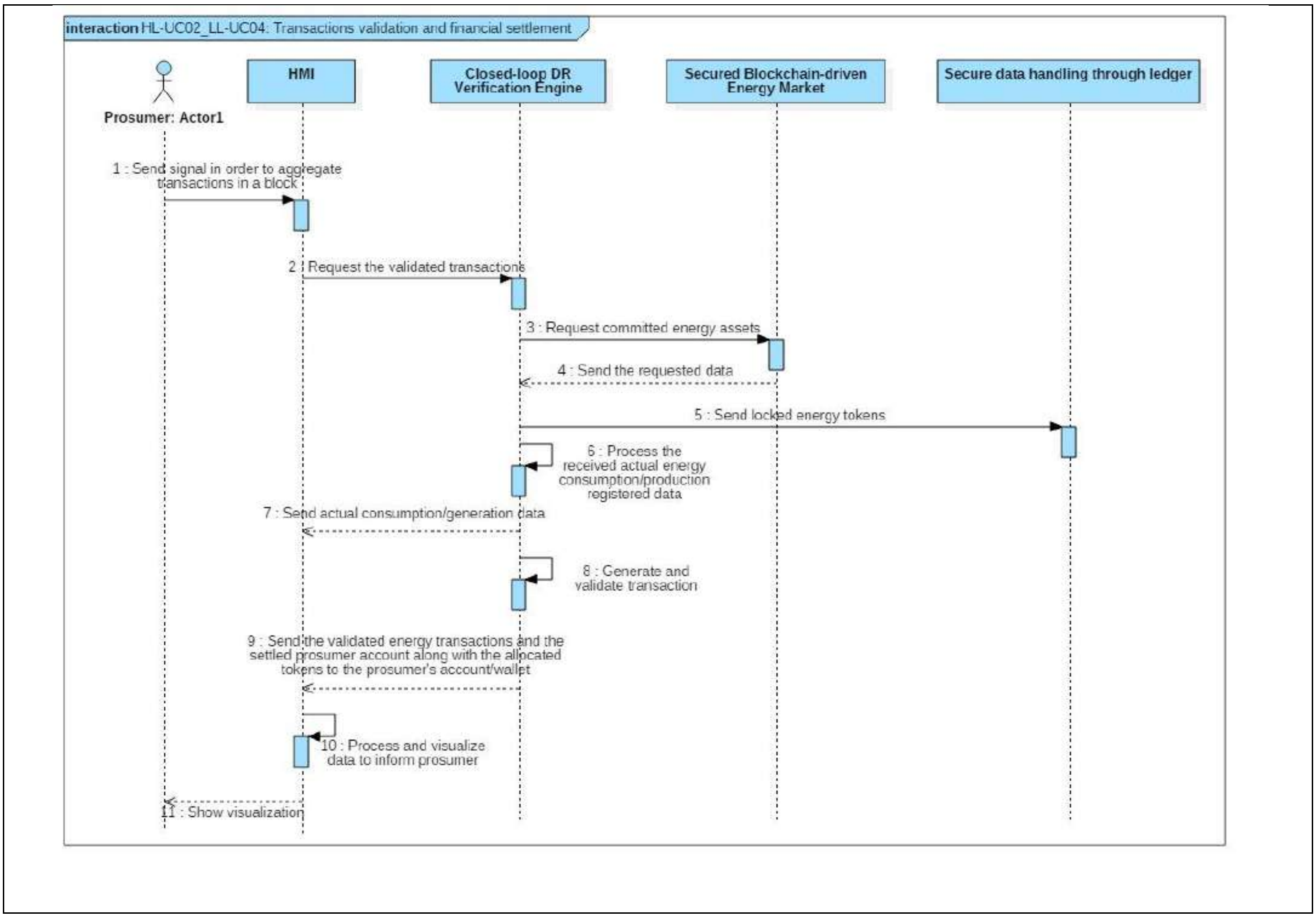


5.2.4 HL-UC02_LL-UC04: Transactions validation and financial settlement

Table 15: HL-UC02_LL-UC04: Transactions validation and financial settlement

Generic Description	
UC 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 3rd Version in D2.5 4th Version in D2.7
Brief Description	The energy transactions are validated and the prosumer accounts settled allocating tokens to the prosumers accounts/wallets.
Assumptions and Pre-Conditions	Market session is ended.
Goal (Successful End Condition)	The aim of this UC is to allocate tokens to the prosumers' accounts/wallets.
Post-Conditions	Tokens are allocated to prosumers.
Involved Actors	Prosumers
UC Initiation	After the end of market session, 1 prosumer is selected as block miner.
Main Flow	Begin 1. The prosumer registers in near-real time the monitored energy assets in the Secured Blockchain-driven Energy Market or, alternatively, through automatic enforce whenever a new monitored value is registered on-chain, thus validating the correct functionality. 2. The Secured Blockchain-driven Energy Market sends the committed energy assets to the Closed-loop DR Verification Engine. 3. The Closed-loop DR Verification Engine processes the received actual energy consumption/production data, and enforces the financial settlement by evaluating the energy assets actually delivered against the energy assets promised through the market orders.

	4. The Closed-loop DR Verification Engine generates and validates the payment transfer between the market participants. 5. The HMI can request information about the actual delivered energy, settlement details and validate transactions from the Closed-loop DR Verification Engine. 6. The HMI processes and visualizes the received data, in order to inform prosumer; 7. The prosumer sees the data through the HMI
Alternative Courses	-
Relationships with other UCs	HL-UC02_LL-UC02, HL-UC02_LL-UC03
Architectural Elements / Services Involved	HMI; Closed-loop DR Verification Engine; Secured Blockchain-driven energy market; Distributed Ledger;
Specific Description	
Relevance to eDREAM WPs *Main Tasks Involved* *Main Technical Partners Involved*	WP4 & WP5 T4.3, T5.1, T5.2 & T5.3 TUC, ENG, E@W, EMOT,
Notes (Optional)	-
UML Sequence Diagram	


interaction HL-UC02_LL-UC04: Transactions validation and financial settlement
Prosumer: Actor1
HMI
Closed-loop DR Verification Engine
Secured Blockchain-driven Energy Market
Secure data handling through ledger
1 : Send signal in order to aggregate transactions in a block
2 : Request the validated transactions
3 : Request committed energy assets
4 : Send the requested data
5 : Send locked energy tokens
6 : Process the received actual energy consumption/production registered data
7 : Send actual consumption/generation data
8 : Generate and validate transaction
9 : Send the validated energy transactions and the settled prosumer account along with the allocated tokens to the prosumer's account/wallet
10 : Process and visualize data to inform prosumer
11 : Show visualization

5.3 High Level UC: VPP in Energy Community

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.

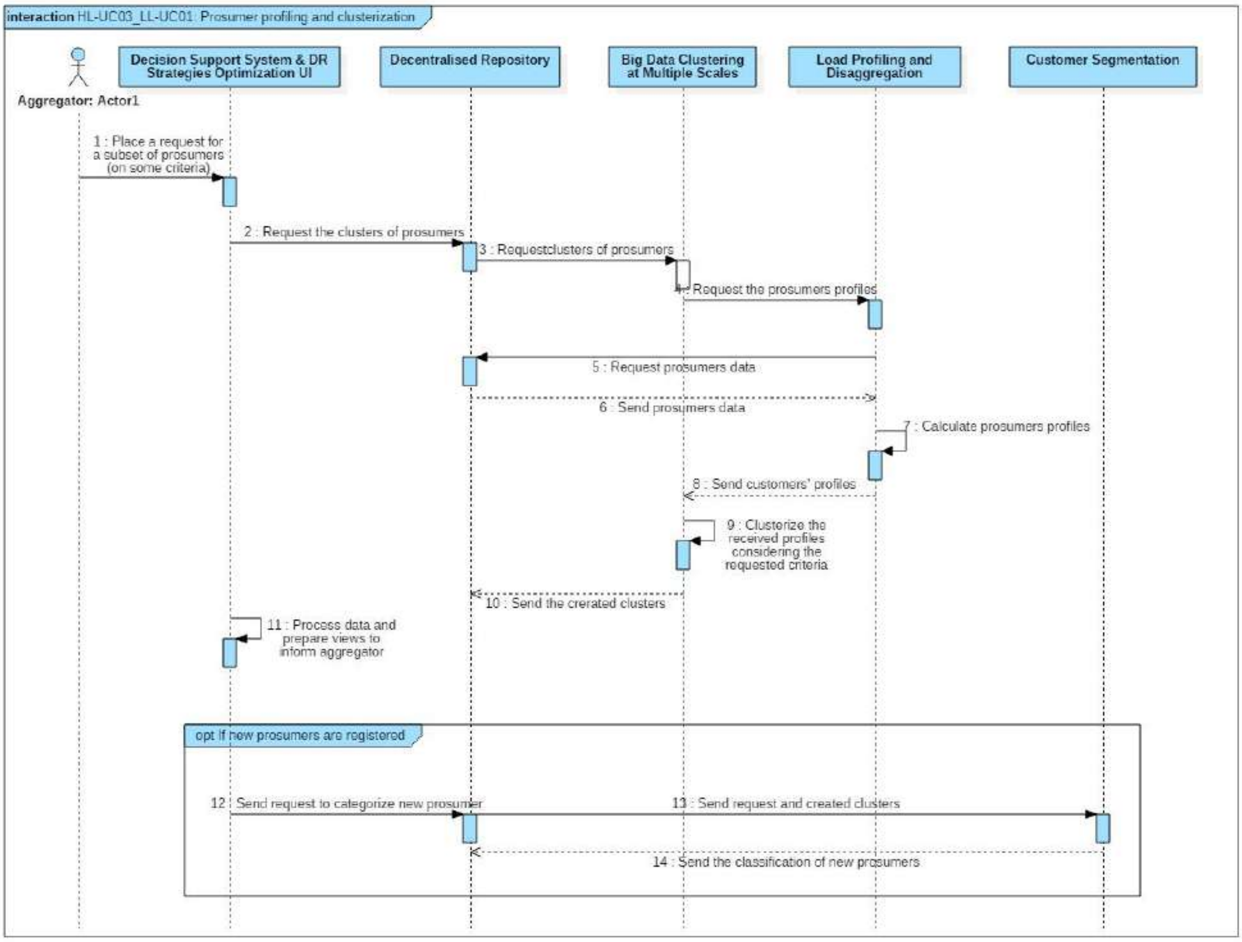
5.3.1 HL-UC03_LL-UC01: Prosumers Profiling and Clusterization

Table 16: HL-UC03_LL-UC01: Prosumers Profiling and Clusterization

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 to 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	Begin 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 Decentralized Repository. 3. The database requests the clusters of prosumers from the Big Data Clustering component.

	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 sends customers' profiles to the Big Data Clustering. 9. The Big Data Clustering clusterizes the received profiles considering the requested criteria. 10. The Big Data Clustering sends the created clusters to the database. 11. The database sends the received clusters to the Graph-based Analytics, in order to be visualized. 12. The Graph-based Analytics processes the received data and informs the Aggregator. 13. 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. 14. The database sends the received request and the created clusters to the Customer Segmentation. 15. The Customer Segmentation sends the classification of new prosumers to the database.
Alternative Cources	-
Relationships with other Use Cases	HL-UC03_LL-UC02, HL-UC03_LL-UC03
Architectural Elements / Services Involved	Graph-based Analytics; Decentralized Repository; Big Data Clustering at multiple scales; Load Profiling; Customer Segmentation;

Specific Description	
Relevance to eDREAM WPs *Main Tasks Involved* *Main Technical Partners Involved*	WP3 & WP4 T3.2, T3.4, T4.2, T4.3 TU, ATOS, CERTH, TUC, E@W
Notes (Optional)	-
UML Sequence Diagram	



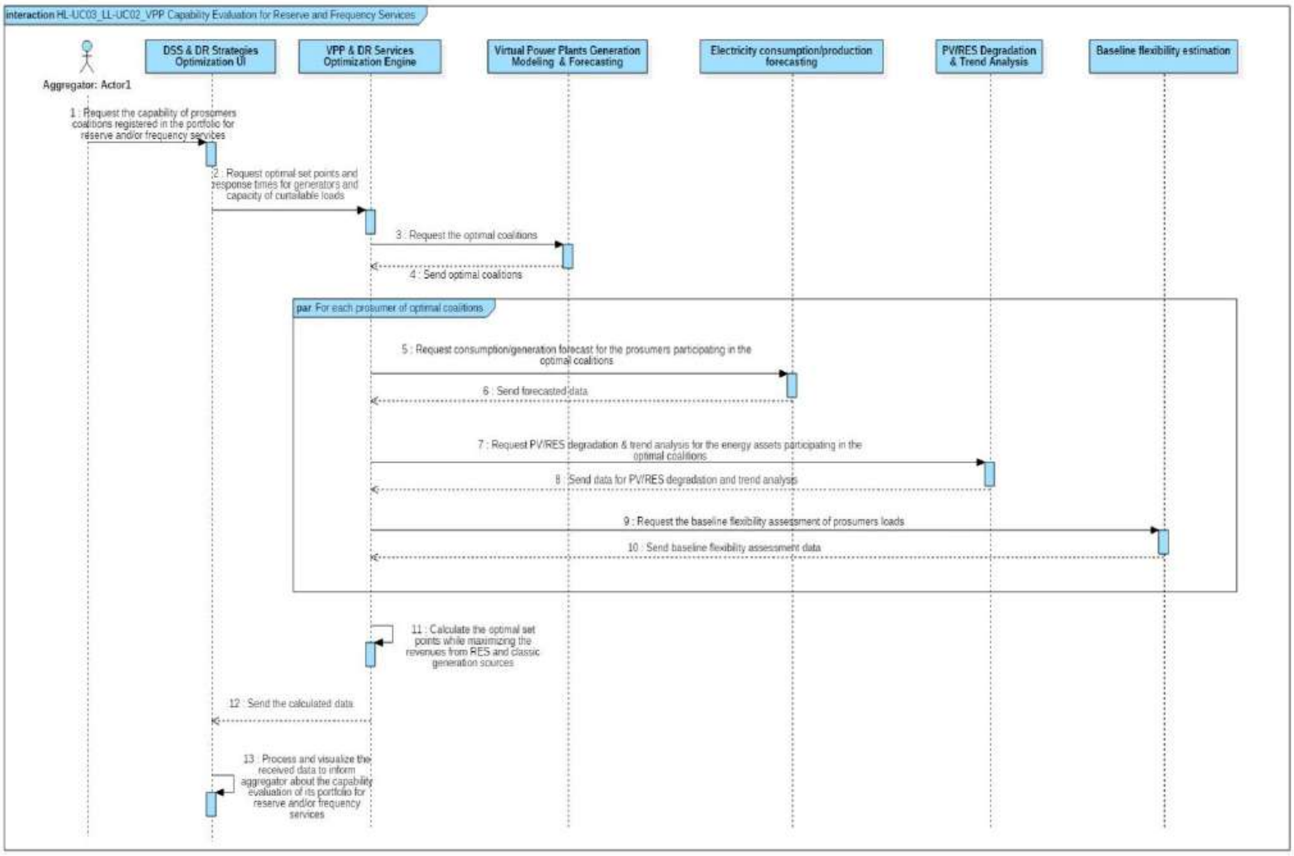
5.3.2 HL-UC03_LL-UC02: VPP Capability Evaluation for Reserve services and for Frequency services

Table 17: VPP Capability Evaluation for Reserve services and for Frequency services

Generic Description	
UC 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
Last Update	1st Version in D2.2 2nd Version in D2.4 3rd Version in D2.5 4th Version in D2.7
Brief Description	Aggregator estimates the capability of its portfolio of prosumers for ancillary services market participation: • ***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.
Assumptions and Pre-Conditions	Aggregator has access to a large number of prosumers, participating in Energy Community programmes with different RES and classic generation sources.
Goal (Successful End Condition)	Maximize utilization and revenues from RES and classic generation sources through accessing Reserve and Frequency markets as an aggregated portfolio.
Post-Conditions	The Aggregator knows the capability of its portfolio considering optimal set points and response times of dispatchable generators and capacity of curtailable loads.
Involved Actors	Aggregator, Prosumers

UC Initiation	The Aggregator needs to know the capability, in order to provide ancillary services, such as reserve services or frequency services.
Main Flow	Begin 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 request optimal set points and response times for generators and capacity of curtailable loads from the VPP and DR Services Optimization Engine. 3. The VPP and DR Services Optimization Engine requests the optimal coalitions from the Virtual Power Plants Generation Modelling & Forecasting. 4. The Virtual Power Plants Generation Modelling & 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.
Alternative Courses	The Aggregator is not able to achieve the capability required for implementation of ancillary services.
Relationships with other UCs	HL-UC03_LL-UC01
Architectural Elements / Services Involved	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 *Main Tasks Involved* *Main Technical Partners Involved*	WP3 & WP4 T3.1, T3.2, T3.3, T3.4, T4.1, T4.2, T4.3 TUC, ENG, ASM, EMOT, CERTH, TU
Notes (Optional)	-
UML Sequence Diagram	


interaction HL-UC03_LL-UC02_VPP Capability Evaluation for Reserve and Frequency Services
Aggregator: Actor1
DSS & DR Strategies Optimization UI
VPP & DR Services Optimization Engine
Virtual Power Plants Generation Modeling & Forecasting
Electricity consumption/production forecasting
PV/RES Degradation & Trend Analysis
Baseline flexibility estimation
1 : Request the capability of prosumers coalitions registered in the portfolio for reserve and/or frequency servibes
2 : Request optimal set points and response times for generators and capacity of curtailable loads
3 : Request the optimal coalitions
4 : Send optimal coalitions
par For each prosumer of optimal coalitions
5 : Request consumption/generation forecast for the prosumers participating in the optimal coalitions
6 : Send forecasted data
7 : Request PV/RES degradation & trend analysis for the energy assets participating in the optimal coalitions
8 : Send data for PV/RES degradation and trend analysis
9 : Request the baseline flexibility assessment of prosumers loads
10 : Send baseline flexibility assessment data
11 : Calculate the optimal set points while maximizing the revenues from RES and classic generation sources
12 : Send the calculated data
13 : Process and visualize the received data to inform aggregator about the capability evaluation of its portfolio for reserve and/or frequency services

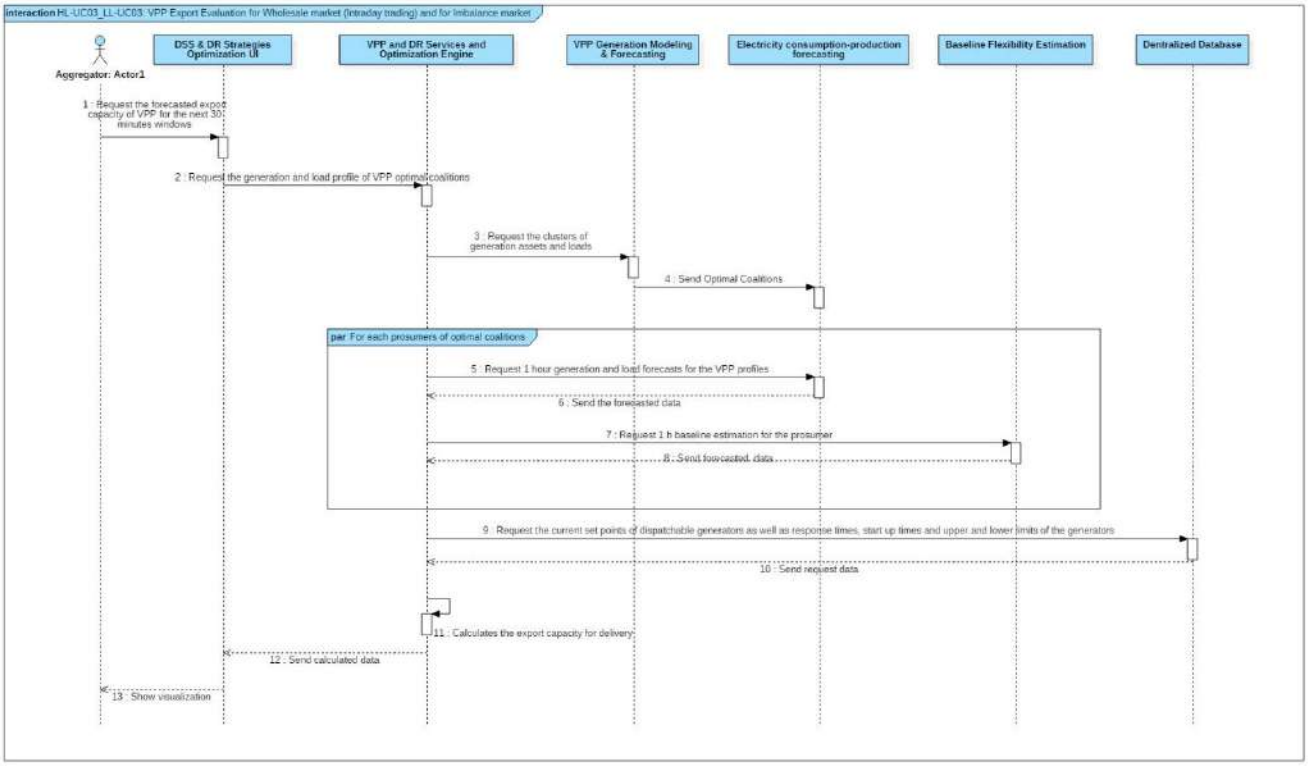
5.3.3 HL-UC03_LL-UC03: VPP Export Evaluation for Wholesale market (Intraday trading) and for Imbalance market

Table 18: HL-UC03_LL-UC03: VPP Export Evaluation for Wholesale market (Intraday trading) and for Imbalance market

Generic Description	
UC Name	**HL-UC03_LL-UC03: VPP Export Evaluation for Wholesale market (Intraday trading) and for Imbalance market**
Version	V0.4
Authors	ENG, E@W, ASM, KIWI
Last Update	1st Version in D2.2 2nd Version in D2.4 3rd Version in D2.5 4th Version in D2.7
Brief Description	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 needs 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: • 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.
Assumptions and Pre-Conditions	Aggregator has access to a large number of prosumers, participating in Energy Community programs 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.
Goal (Successful End Condition)	Maximize utilization and revenues from RES and classic generation sources through accessing Wholesale and Imbalance markets as an aggregated portfolio.
Post-Conditions	The Aggregator knows the forecasted export capacity of the VPPs for the next 30 minutes window.
Involved Actors	Aggregator, Prosumers
UC Initiation	The Aggregator needs to know the forecasted capacity, in order to implement trading services, such as intraday trading and trading in the imbalance market.
Main Flow	Begin 1. The Aggregator requests the forecasted export capacity of VPP for the next one-hour 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 request the VPP optimal coalitions from the Virtual Power Plants Generation Modeling & Forecasting. 4. The Virtual Power Plants Generation Modeling & Forecasting sends the requested data. For each prosumer of optimal coalitions, the steps 5-8 are repeated. 5. The VPP and DR Services Optimization Engine requests next 30 minutes consumption/production forecast for the prosumer 6. The Electricity Consumption/Production Forecasting sends the forecasted data. 7. The VPP and DR Services Optimization Engine requests next 30 minutes baseline estimate for the prosumer. 8. The Baseline Estimation sends the requested data.

	9. The VPP and DR Services Optimization Engine requests the current set points of dispatchable generators as well as response times, start-up times and upper/lower limits of the generators. 10. Decentralized Repository (Database) sends the requested data 11. The VPP and DR Services Optimization Engine calculates the export capacity for delivery. 12. The VPP and DR Services Optimization Engine sends the calculated data to the DSS & DR Strategies Optimization UI, in order to inform aggregator; 13. The Aggregator sees the data through the DSS & DR Strategies Optimization UI
Alternative Courses	The Aggregator is not able to achieve the capability required for implementation of trading services.
Relationships with other UCs	HL-UC03_LL-UC01
Architectural Elements / Services Involved	DSS & DR Strategies Optimization UI; VPP and DR Services Optimization Engine; Virtual Power Plants Generation Modeling & Forecasting; Electricity Consumption/Production Forecasting;
Specific Description	
Relevance to eDREAM WPs *Main Tasks Involved* *Main Technical Partners Involved*	WP3 & WP4 T3.1, T3.3, T4.1, T4.2, T4.3 TUC, ATOS, CERTH, TU
Notes (Optional)	-
UML Sequence Diagram	


interaction HL-UC03_LL-UC03: VPP Export Evaluation for Wholesale market (Intraday trading) and for Imbalance market
Aggregator: Actor1
DSS & DR Strategies Optimization UI
VPP and DR Services and Optimization Engine
VPP Generation Modeling & Forecasting
Electricity consumption-production forecasting
Baseline Flexibility Estimation
Dentralized Database
1 : Request the forecasted export capacity of VPP for the next 30 minutes windows
2 : Request the generation and load profile of VPP optimal coalitions
3 : Request the clusters of generation assets and loads
4 : Send Optimal Coalitions
par For each prosumers of optimal coalitions
5 : Request 1 hour generation and load forecasts for the VPP profiles
6 : Send the forecasted data
7 : Request 1 h baseline estimation for the prosumer
8 : Send forecasted data
9 : Request the current set points of dispatchable generators as well as response times, start up times and upper and lower limits of the generators
10 : Send request data
11 : Calculates the export capacity for delivery
12 : Send calculated data
13 : Show visualization

6 Deployment View

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.

6.1 Active Micro-Grid – ASM Terni

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

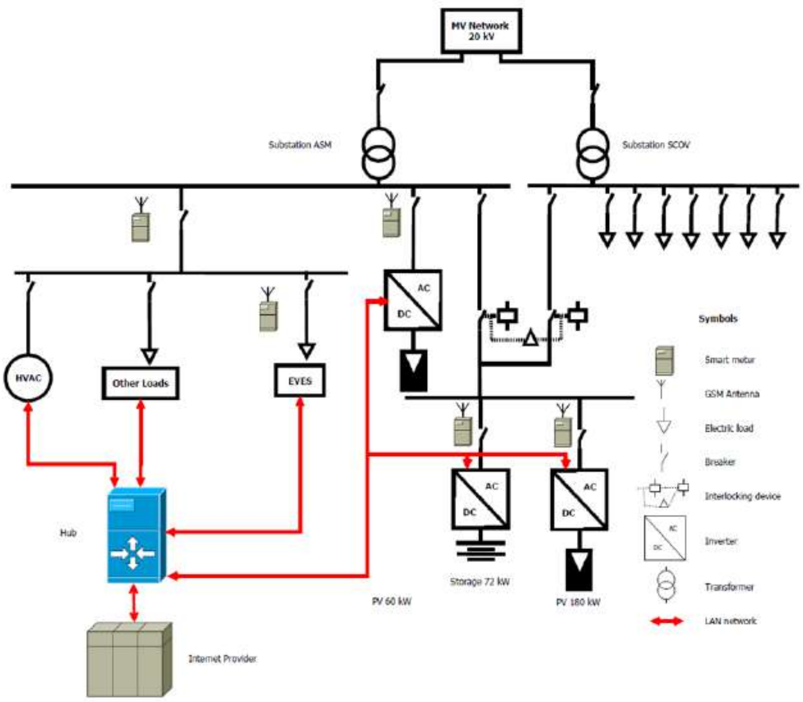


Figure 22: Internal and external connections in the Terni Micro-Grid

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

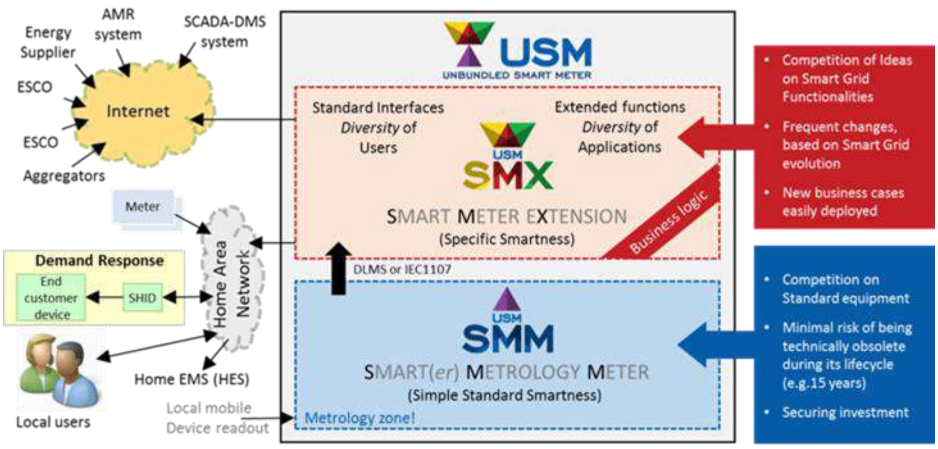


Figure 23: Unbundled Smart Meter Concept

A shown in Figure 23 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.

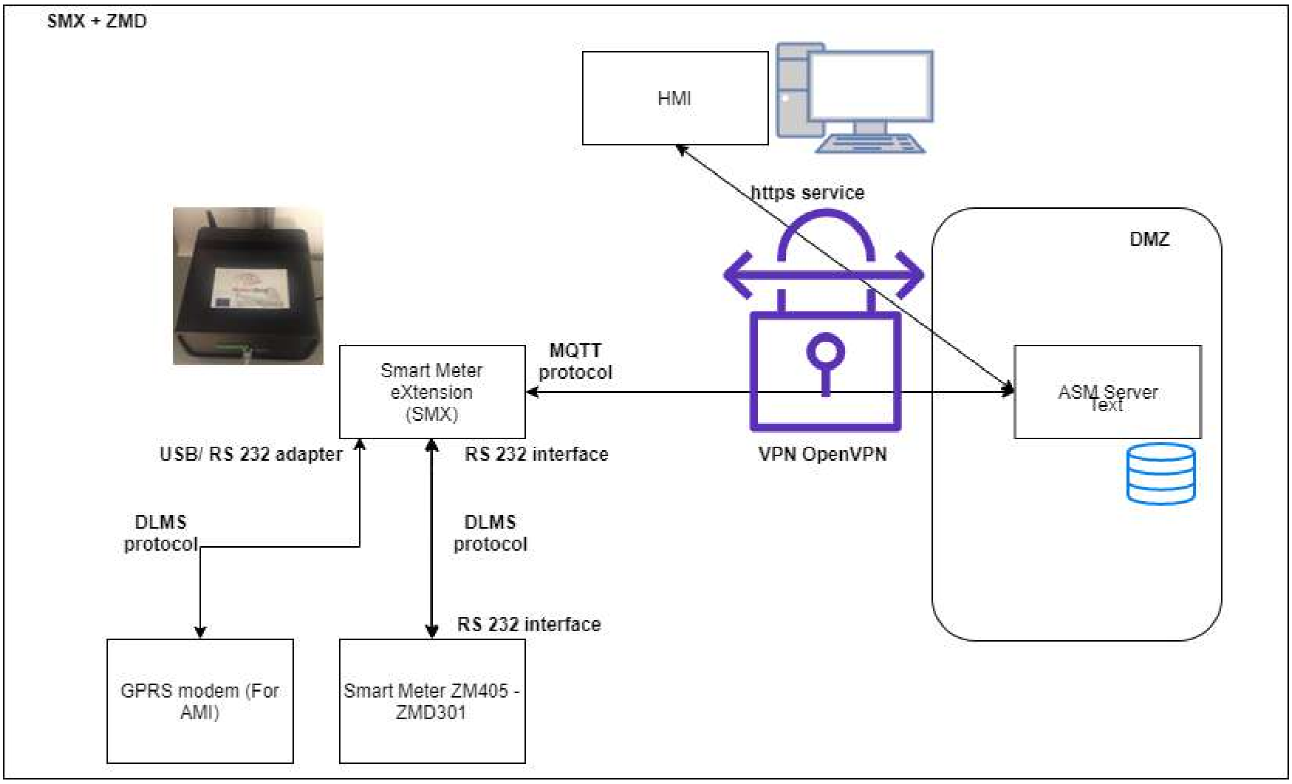


Figure 24: High-level architecture for the communication between SM and SMX

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

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

Figure 25: Emotion SpotLink EVO charging stations

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

Renault ZOE, acronym of ZerO Emissions, is a five-door supermini electric car produced by the French manufacturer Renault. The ZOE is powered by a 22 kWh lithium-ion battery pack weighing 275 kg, driving a 65 kW (87 bhp; 88 PS) synchronous electric motor supplied by Continental (the Q210). Maximum torque is 220 N·m (162 lb-ft) with a top speed of 135 km/h (84 mph). The NEDC cycle range is 210 km (130 mi). Renault estimates that in suburban use, the ZOE can achieve around 100 km (62 mi) in cold weather and 150 km (93 mi) in temperate conditions. The car features a charging system called "Caméléon" (Chameleon) charger that allows the ZOE to be charged at any level of power, taking between 30 minutes and nine hours [32].

Figure 26: Renault ZOE customized by Emotion

The Nissan LEAF, acronym of Leading, Environmentally Friendly, Affordable, Family car, is an electric propulsion car introduced by Nissan on the markets in December 2010. It is equipped with an 80 kW (109 hp) synchronous AC electric motor. The first version equipped with a lithium-ion battery, consisting of 48 modules and each of them contains 4 cells for a total of 192, with a capacity of 24 kWh and an autonomy of 199 km NEDC cycle. Since 2016, a 30 kWh Lithium-ion battery with an operating cycle of 250 km NEDC is available as an accessory. Nissan LEAF recharges in alternating current or in direct current. In AC, LEAF uses an on-board charger with a maximum 7.4 kW (32A maximum current, 230V, single-phase) with the Type 2 socket. In DC it uses the CHAdeMO standard up to 50 kW of power. Charging times vary from 5/6 hours to about 7 kW up to 1 hour with direct current charging [33].

Figure 27: Nissan LEAF

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

• CPU: 2 core 3.1 GHz;

• HDD: 50 GB;

• RAM: 4 GB;

• S.O.: Ubuntu 16.04 LTS.

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

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

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

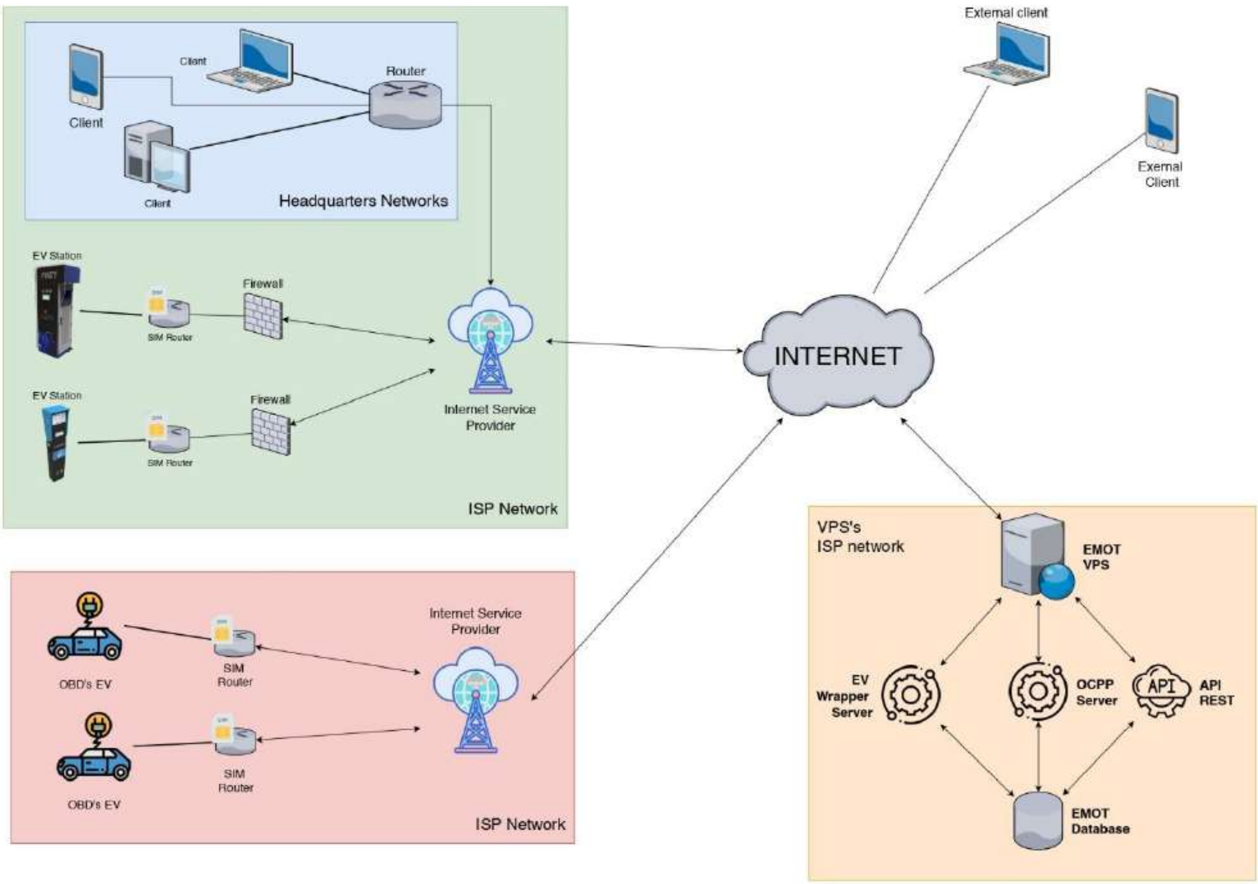


Figure 28: EMOT network topology

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.

6.2 Community-based Virtual Power Plants – KiWi Power

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

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

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

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

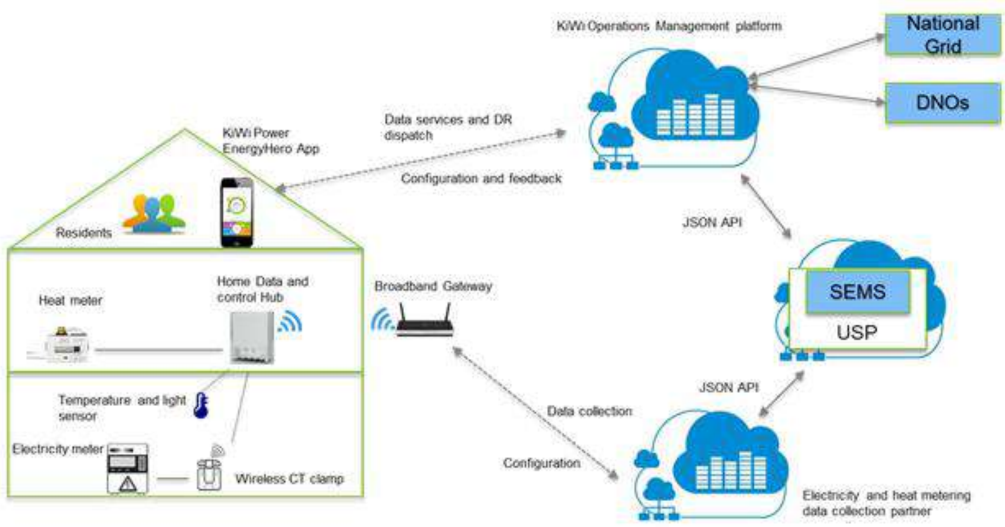


Figure 29: Residential estate in Greenwich for testing residential demand response scenarios

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

6.3 Overall Deployment Architecture

This section presents an updated version of the deployment view of the eDREAM platform indicating the interactions with the physical world. The two pilot sites have different technologies of smart meters that will provide the eDREAM platform with the necessary real time measurements for the testing of the architectural components and the tools. These devices are connected to appropriate IoT devices, which will forward the information to the FIWARE Orion Context Broker and to the Off-Chain Data Handlers respectively for the blockchain related information. As a next step, the Context Broker creates different context elements of the received information and manages them. The deployment view of the eDREAM platform is presented in the following Figure 30:

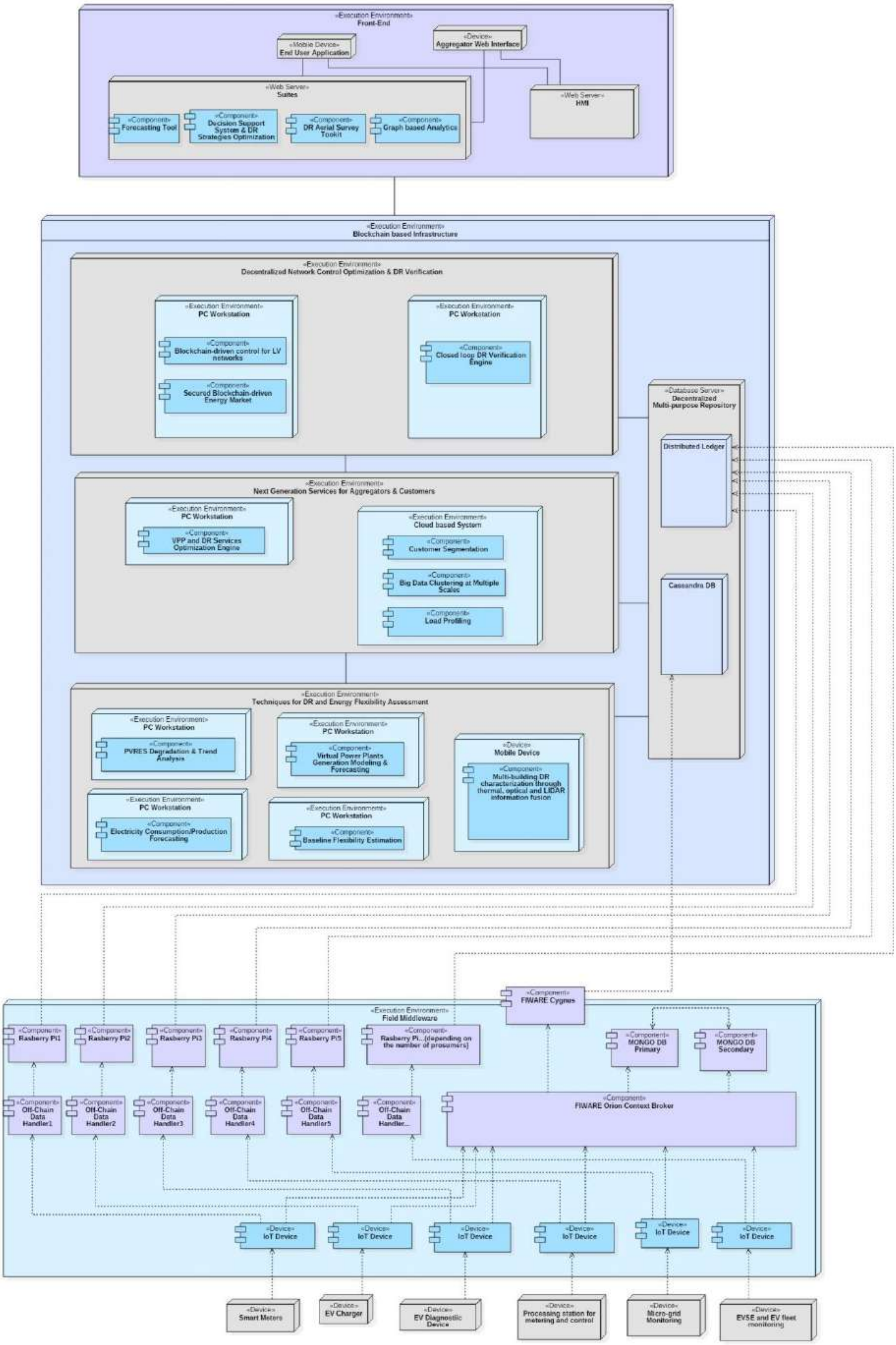


Figure 30: eDREAM Deployment View Architecture

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:

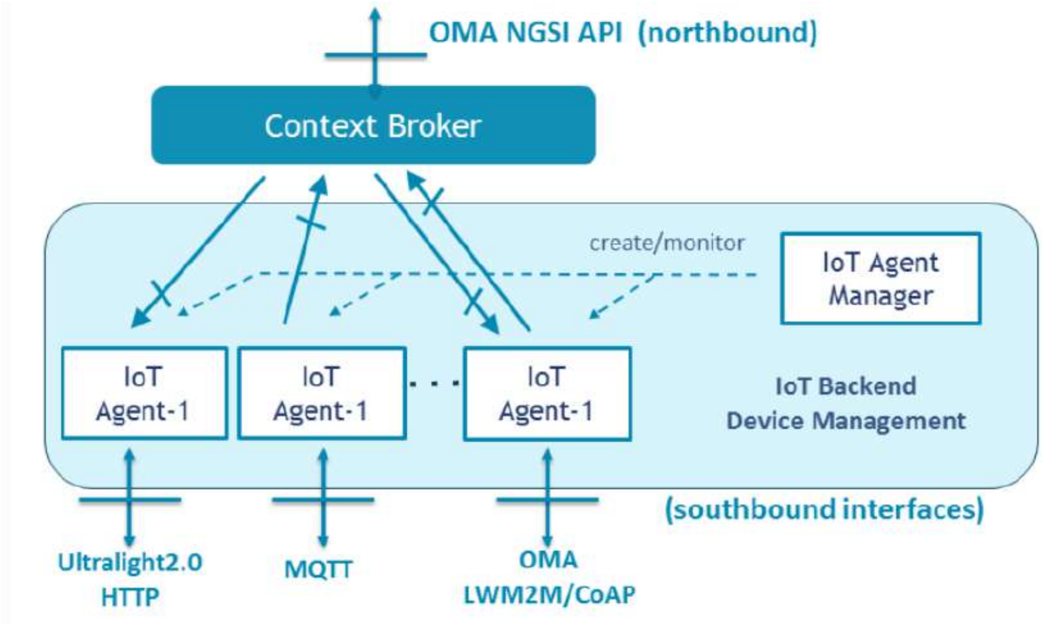


Figure 31: IoT Agents' Concept and Connection

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

- The accomplishment of a secure API gateway can significantly reduce coding efforts and make the applications much more efficient, while decreasing errors. A microservices API gateway behaves like any other API gateway. This component provides a front end layer used to access the below microservices. This gateway creates a single interface for a single application. This means that it can create multiple APIs, one for each platform (e.g. mobile applications, browsers, serve-side

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

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 Kong[4] and Keycloak[5]. For both technologies, the Postgresql[6] can be used as a backend database.

- All the microservices shall be able to communicate with each other through rest APIs or AMQP/MQTT messages. A possible solution for message broking is the RabbitMQ[7] that can be easily developed over python or Java using Pika or Java AMQP-client.
- Finally, one more step for the appropriate delivery of microservices is the use of docker[8] 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 modelling, 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/

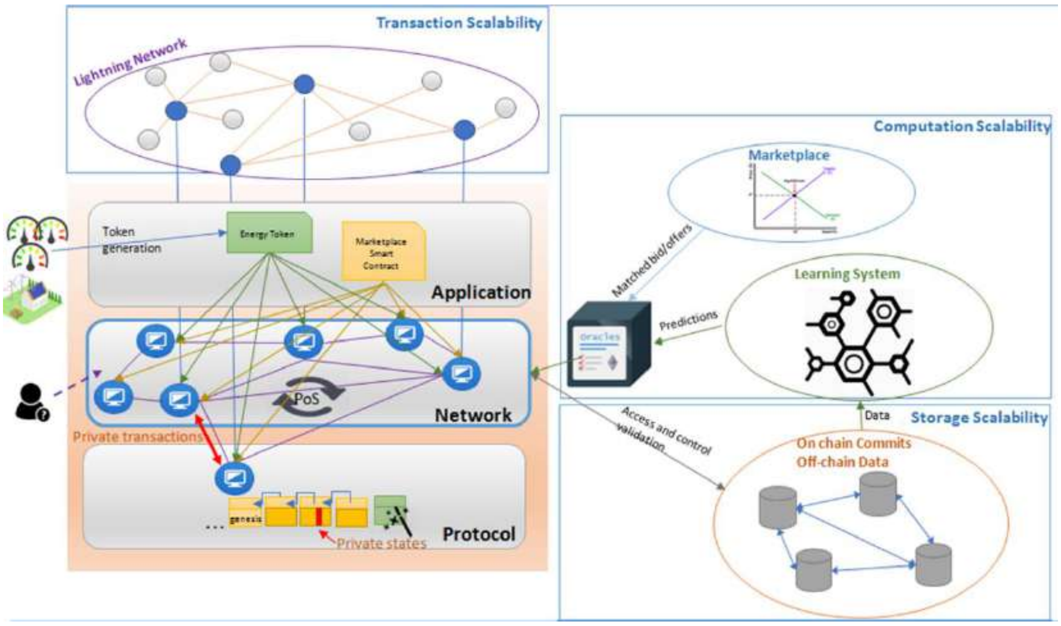


Figure 32: Blockchain-based Infrastructure

For the testing and implementation of the secure distributed ledger technology, the Parity Ethereum Client[9] 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:

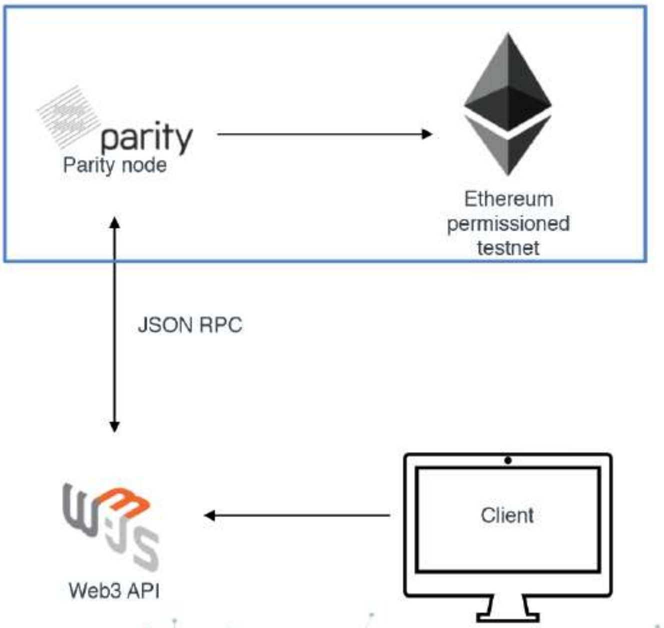


Figure 33: Testbed Topology for the Distributed Ledger Technology

6.4 Field Devices

In this section, the field devices, that are going to be used in the two pilot sites for taking real time measurements, are presented along with their functionalities, technical characteristics and installation

[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).

Table 19: Smart Metrology Meter (SMM)

Device: Smart Metrology Meter	
Name	Smart Metrology Meter (SMM)
Short Description	Wally-A is used in the pilot as SMM, it is used for monitoring the block of energy units within the pilot a as well as it can provides power analysis. 
Measurement	Electrical values: § Voltages and Currents § Active, Reactive and Apparent power § Active and Reactive (4 quadrants) Energy § Power Factor § Frequency § Flicker (Pst e Plt) § Voltages and Currents Harmonics and Interharmonics (up to 50° order) § Voltage Unbalance § Voltage Dips and Swells § Voltage Interruptions (short and long) § Rapid Voltage Changes § Waveforms (window records with programmable Pre and Post-Triggers
Digital/Analog Signals	Output 4 – 20 mA
Functionality	Wally-A is a metering device with high level class of precision, certificated according 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	Voltage: 80÷275 Vac/dc - 50/60Hz Power consumptions: 20 VA

	Battery: 12 V 0.8Ah Pb sealed Battery operating time: 30 minutes, self-limited
Data Connections	Modem type: Internal, Quadband 850/900/1800/1900 MHz Networking: GSM and GPRS/UMTS Internal antenna Optional external antenna SIM holder accessible
Data Format	.csv, .json, .xml, .pqdif.
Data Size	In conjunction with SMX average is 1 MB / day, 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	-

Table 20: Smart Meter Extension (SMX)

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

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

Table 21: EVO Emotion

Device: EVO Emotion	
Name	EVO Emotion
Short Description	Electric Vehicle Supply Equipment 

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

Table 22: OBD Emotion

Device: OBD Emotion	
Name	*OBD Emotion*
Short Description	*EV on-board diagnostic device*

	
Measurement	Battery State-of-Charge (%); Residual Autonomy (Km); Needed time to Full Charge (m); Geolocation (geographic coordinates); Doors Car State (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*

Table 23: Fruit KiWi Power

Device: Fruit	
Name	Fruit KiWi Power
Short Description	Low-cost processing station designed for Demand Response and Battery Management. The Fruit provides a wide range of metering and control functions and is integrated with the KiWi Operations Management Platform to allow easy set-up and monitoring for any application. It provides flexible interfacing options including GPIO, RS232, RS485 and relay terminals and supports Modbus/TCP and Modbus/RTU. 
Measurement	Supply voltage (V); Current consumption (at 12V) (mA); Current consumption (at 24V) (mA); Pulse input voltage (V); Pulse measurement frequency (Hz); Maximum number of Segments; Relay terminal voltage (V); Relay terminal current (AC) (A); Relay switching power (VA).
Digital/Analog Signals	-
Functionality	Enabling centrally-dispatched Demand Response 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;

	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	-

7 Architectural Components Detailed Specifications

Table 24: Micro-grid Monitor

Name of Existing Component/Service	***Micro-grid Monitor***	
Type	Service	
Functionality	Data gathering from the energy units	
***Output Connections & Interfaces:* To which architectural component it is offering services**	• 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	
***Input Connections & Interfaces:* which other components are using it**	Field data aggregation	
Functional Requirements	FD_BR01_UR01_FR01	
Non-Functional Requirements	-	
Input Parameters		
Description	*Modelling format*	*Data Received From*
§ 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
Output Parameters		
Description	*Modelling format*	*Data Sent To*
§ 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	-	

Table 25: EVSEs and EV fleet monitoring

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	• *Electricity Consumption/Production Forecasting* • *Baseline Flexibility Estimation* • *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	-

Input Parameters		
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*
Output Parameters		
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	-	

Table 26: Electricity Consumption/Production Forecasting

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	*Energy demand and production forecasting for Day-ahead, intra-day, one hour ahead time frames* *Flexibility forecasting for day-ahead and intr-day time frames*
Input Connections & Interfaces: From which components it receives input	• *Cassandra DB* • *Baseline Flexibility Estimation*
Output Connections & Interfaces: To which components it sends the results	• 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* REST API: <host>:<port>/edream-ebb/
Functional Requirements	MF01_BR02_UR01_FR01, MF01_BR02_UR02_FR02 MF01_BR02_UR03_FR03, MF01_BR02_UR04_FR04 MF01_BR02_UR05_FR05, MF02_BR07_UR01_FR06 MF02_BR07_UR02_FR07, MF02_BR07_UR03_FR08
Non-Functional Requirements	MF01_BR02_UR01_NFR01

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, "deviceMeasurementId": "18bb6e98-8429-4162-8236-cc4f231bb2a4", "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" }, ...]]	**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, "deviceMeasurementId": "18bb6e98-8429-4162-8236-cc4f231bb2a4", "deviceId": "4836bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "property": "Baseline Consumption" }, { "timestamp": 1537803600000,	**Range** – **Frequency**:	*Baseline Flexibility Estimation*

			"value": 908.483, "deviceMeasurementId": "18bb6e98-8429-4162-8236-cc4f231bb2a4", "deviceId": "4836bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "property": "Baseline Consumption" } ...]		
Output Parameters					
Attribute /Para-meter	**Short Description**	**Data Type**	**Data Format**	**Value Range & Frequency**	**Data Sent To**

Forecasted values	The module will provide an array containing the device and the measurement the forecast is referring to, together with the forecasted values		{
			"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"
			}
				Range – depending on the forecasted property
				Frequency: every 30 min, 1 hour
					• 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*

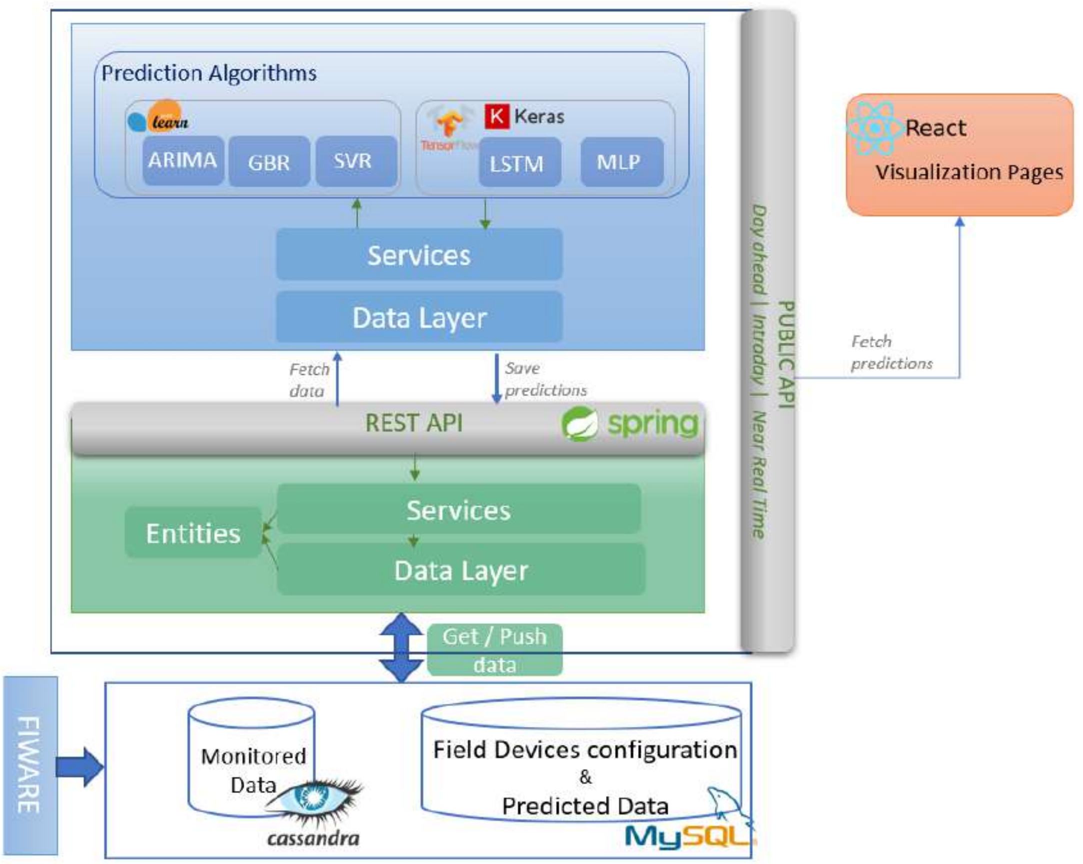
Software Requirements/Development Language	• 64 bit Linux / Windows OS • JRE (Java Runtime Environment) version 8 or above • Internet connection for package management • *nVIDIA CUDA, Keras, TensorFlow / Python3,* • Cassandra, MySql, RabbitMQ
Hardware Requirements	• Intel/AMD processor with at least 4 cores and 2.5GHz base frequency • 8 GB RAM • HDD/SSD with at least 100GB free space • Internet connection for package management • *NVIDIA GeForce GTX 1050 (equivalent or above)*
Communications	REST API , HTTP
Status of the development of the component	Initial component prototype implemented
High-level Class Diagram	

Table 27: Virtual Power Plants Generation Modelling & Forecasting

Name of New Component/Service:	Virtual Power Plants Generation Modelling & 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* *- trade energy for increasing profit;* *- offer capacity bidding service;* *- participate to demand response programs* *- offer reactive power control services*
Input Connections & Interfaces: From which components it receives input	• Electricity Consumption/*Production* Forecasting • PV/RES Degradation & Trend Analysis • *Cassandra DB*
Output Connections & Interfaces: To which components it sends the results	• VPP *and* DR Services Optimization *engine* REST API: <host>:<port>/edream-vpp/
Functional Requirements	MF02_BR03_UR01_FR01, MF02_BR03_UR02_FR02 MF02_BR03_UR03_FR03
Non-Functional Requirements	MF02_BR03_UR01_NFR01

Input Parameters					
Attribute/*Parameter*	**Short Description**	**Data Type**	**Data Format**	**Value Range & Frequency**	**Data Received From**
Production /Consumption forecasts for each participating producer/grid	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" }, ...], "entityDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "profileGranularityMinutes": 60 , "predictionGranularity": "DAYAHEAD", "property": "ENERGY CONSUMPTION" }	**Range** – depending on the forecasted property **Frequency**: every 30 min, 1 hour	• Electricity Production/Consumption Forecasting – Prosumer Level/Grid Level
Improved short-term forecasting information	Information about the device degradation	JSON	{ "value": 100, "deviceMeasurementId": "12bb6e98-8429-4162-8236-cc4f231bb2a4", "deviceId": "4236bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "property": "Degradation-Trend" }	**Range** – **Frequency**: every 30 min, 1 hour	• PV/RES Degradation & Trend Analysis
Requested Goal	Choose the goal to be followed by the prosumer oprimization process		"goal": { type" : "ENERGY TRADING", "priceSignal": [{"value": 157167, "timestamp": "2018-09-08T00:00:00"},	**Range** – **Frequency**: on request	• HDMI

			{"value": 157234, "timestamp": "2018-09-08T01:00:00"}, ...] }		

Prosumers portofolio	The available prosumers portfolio		{
			"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,
				Range – Frequency:	• *Cassandra DB*

			"max": 1.2,
			
			"entityDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2",
			
			"deviceMeasurementId":"c0a735f0-b6fe-47e6-b951-79910cd0e822",
			
			"property": "Degradation-Trend"
			}
			
			"prosumerDetails": {
			
			"specification": {...}
			
			"type": "DEG"
			}
			}, ...],
			}		

Output Parameters					
Attribute /Para-meter	**Short Description**	**Data Type**	**Data Format**	**Value Range & Frequency**	**Data Sent To**

Optimized prosumers aggregated in VPP	An array containing all the prosumers that are part of the optimized VPP	JSON	{
			"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"}, ...
				Range – **Frequency**: on request	• VPP and active micro-grid flexibility
					• VPP & DR Services Optimization Engine

			]}, ...] "totalEnergyTraded": [{"value": 157160, "timestamp": "2018-09-08T00:00:00"}, {"value": 157230, "timestamp": "2018-09-08T01:00:00"}, ...]		

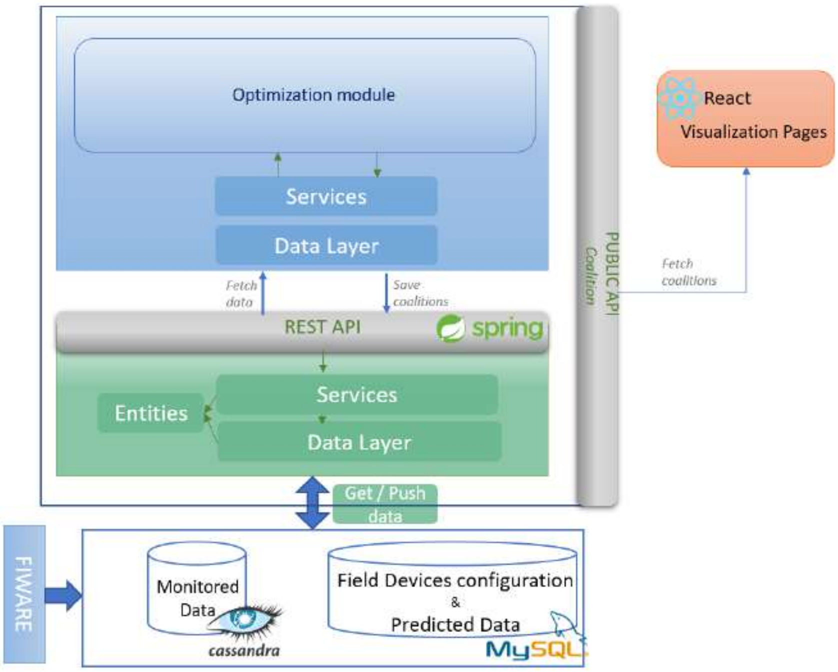
Software Requirements/Development Language	• 64 bit Linux / Windows OS • JRE (Java Runtime Environment) version 8 or above • Internet connection for package management
Hardware Requirements	• Intel/AMD processor with at least 4 cores and 2.5GHz base frequency • 8 GB RAM • HDD/SSD with at least 100GB free space • Internet connection for package management
Communications	REST API
Status of the development of the component	Initial component prototype implemented
High-level Class Diagram	

Table 28: Baseline Flexibility Estimation

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.*		
***Input Connections & Interfaces:* From which components it receives input**			• *Cassandra DB* • *Multi-building DR Characterization*		
***Output Connections & Interfaces:* To which components it sends the results**			• *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			*MF01_BR03_UR01_NFR01*		
Input Parameters					
Attribute/Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Received From***
Prosumer baseline profiles	*A profile for prosumer baseline*	*Prosumer ID (int), DR Programme ID*	*CSV, JSON or XML*	*Non-negative, granulrity*	• *Cassandra DB*

	flexibity estimation	*(int) starttimstamp (string), endtimestamp (string), baseline energy flexibility (float), monetary cost (float), comfort cost (float)*		*is 15 minutes*	• *Multi-Building DR Characterization*
Constraints of the installed devices					• *Cassandra DB*
Output Parameters					
Attribute/Para-meter	***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)-Starttimestap (string)-endtimestamp (string)-Energy consumption flexibility array (float[])-monetary cost during starttime and endtime (float []), comfort cost during starttime and*	*CSV, JSON or XML*	*Greater than 0, every 15 minutes*	• *VPP and DR Services Optimization Engine* • *Blockchain-driven control for LV networks*

		end time (float[])			

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.* *In case it needs any special sensor that is included in the sensor specification, it can be included also here as a reference.*
Communications	*Ethernet or WiFi Connectivity*
Status of the development of the component	Initial component prototype implemented

Table 29: PV/RES Degradation & Trend Analysis

Name of New Component/Service:	***PV/RES Degradation and Trend Analysis***
Type:	*Component*
Functionality:	*Calculates the degradation rate at which PV/RES modules lose their performance over time* *Improves the short-term forecasting of generation of the PV/RES units based on near real-time trend analysis algorithm*
Provided Services	*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*
***Input Connections & Interfaces:* From which components it receives input**	• *Electricity consumption/production forecasting* • *Cassandra DB*

***Output Connections & Interfaces:** To which components it sends the results*	• *Virtual Power Plants Generation Modelling and Forecasting* • *VPP and DR Services Optimization Engine* • *DSS & DR Strategies Optimization* • *Forecasting Tool*
Functional Requirements	*MF01_BR04_UR01_FR01, MF01_BR04_UR02_FR02, MF01_BR04_FR03, MF01_BR04_FR04*
Non-Functional Requirements	*MF01_BR04_UR01_NFR01, MF01_BR04_UR02_NFR02, MF01_BR04_UR03_NFR03*

Input Parameters

Attribute/Para-meter	***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 panel in month/year*	*Reated power in watts, reference irradiance (1000w/m²)* *date (yy-mm)*	*CSV/JSON*	*15 min*	
Output Parameters					
Attribute/Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Sent To***
Improved short-term and long-term forecasting of generation	*Improving the forecasting of generation of PV*	*Float*	*CSV/JSON*	*>=0* *15 min*	• *VPP Generation Modelling and Forecasting* • *VPP and DR Services Optimization Engine* • *DSS & DR Strategies Optimization* • *Forecasting Tool*
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.*		
Communications			*Ethernet or WiFi Connectivity*		

Status of the development of the component	Initial component prototype implemented

Table 30: Multi-building DR characterization through optimal, thermal and LIDAR information fusion

Name of New Component/Service:	***Multi-building DR characterization through thermal, optical and LIDAR information fusion***
Type:	*Component*
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*
***Input Connections & Interfaces:* From which components it receives input**	• Drone camera mount • LiDAR Ethernet port • Skyport API • Cassandra DB
***Output Connections & Interfaces:* To which components it sends the results**	• 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	*MF01_BR01_UR01_NFR01*
Input Parameters	

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 images taken during the aerial survey	bitmap	.jpeg	8 bit per channel RGBA	Drone camera
Smart Meter data (KiWi)					Cassandra DB
Weather Data	Data from weather station.	Luminosity, temperature, humidity, rain, wind speed	.csv	15min frequency	
Output Parameters					
Attribute/Parameter	**Short Description**	**Data Type**	**Data Format**	**Value Range & Frequency**	**Data Sent To**
3d model	3d model made from pics taken during aerial survey.	geometry	.obj	n/a	3d workstation system
Output from image processing (CERTH)	Building thermal signature	Units of thermal leakage levels with respect to baseline	.csv	n/a, once per scan	Cassandra DB DR Aerial Survey Toolkit

					Baseline Flexibility Estimation

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	Initial component prototype implemented

Table 31: Load Profiling

Name of New Component/Service:	***Load Profiling***
Type:	*Component*
Functionality:	*A non-intrusive appliance load analysis technique*
Provided Services	• *Creates different profiles based on loads' measurements and forecast* • *Serves profiles to other components*
Input Connections & Interfaces: From which components it receives input	• *Electricity Consumption/Generation Forecast* • *Cassandra DB /InfluxDB*
Output Connections & Interfaces: To which components it sends the results	• *Big Data Clustering at Multiple Scales* • *Cassandra 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*

Input Parameters

Attribute/Parameter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Received From***
Prosumer ID	*Prosumer Identifier*	*String*	*csv*	*E. Abs >= 0 Freq = 15 min*	*Cassandra DB*
Historical data	*Historical data for load consumption*	*Float*	*csv*	*E. Abs >= 0 Freq = 15 min*	
Real-time data	*Data from field devices*	*Float*	*csv*	*E. Abs >= 0 Freq = 15 min*	
KPIs	*Defines indicators*	*String*	*csv*	-	
Forecasted data for prosumers' consumption					*Electricity Consumption/Production Forecasting*

Output Parameters

Attribute/Parameter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Sent To***
Prosumers profiles	*Pattern related to*	*Float*	*csv*	*E. Abs >= 0 Freq = 15 min*	*Big Data Clustering at Multiple Scales*

	specific features				*Cassandra DB/InfluxDB*

Software Requirements/Development Language	*TCP/IP connectivity* *MQTT/AMPQ, HTTP as transport protocols*
Hardware Requirements	*Cloud-based system*
Communications	*Input data should be provided either by csv files or by API requests.* *Output data shall be offered by API requests or by a context brokering service.*
Status of the development of the component	Initial component prototype implemented

Table 32: Big Data Clustering at Multiple Scales

Name of New Component/Service:	***Big Data Clustering at Multiple Scales***
Type:	*Component*
Functionality:	*Analytical component for clusterization of energy customers.*
Provided Services	• *Creates clusters based on load profiles* • *Serves clusters to other components*
Input Connections & Interfaces: From which components it receives input	• *Load Profiling* • *Cassandra DB/InfluxDB*
Output Connections & Interfaces: To which components it sends the results	• *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_UR02_NFR02, MF02_BR01_UR03_NFR03, MF02_BR01_UR04_NFR04, MF02_BR01_UR05_NFR05, MF02_BR01_UR06_NFR06*			
Input Parameters					
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*	• *Load Profiling*
KPIs	*DR related indiators*	*String*	*csv*	-	*Cassandra DB/Influx DB*
Output Parameters					
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*	• *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	*Input data should be provided either by csv files or by API requests.* *Output data shall be offered by API requests or by a context brokering service.*
Status of the development of the component	Initial component prototype implemented

Table 33: Customer Segmentation

Name of New Component/Service:	***Customers Segmentation***
Type:	*Component*
Functionality:	*Big data tool for clusterization of energy customers*
Provided Services	• *Classifies users based on the clusters* • *Serves classification to other components*
Input Connections & Interfaces: From which components it receives input	• *Big Data Clustering at multiple scales* • *Cassandra DB/InfluxDB*
Output Connections & Interfaces: To which components it sends the results	• *Cassandra DB/InfluxDB*
Functional Requirements	*MF02_BR01_FR01, MF02_BR01_FR02*

			MF02_BR02_FR01, MF02_BR02_FR02 *MF02_BR02_FR03, MF02_BR02_FR04* *MF02_BR02_FR05*		
Non-Functional Requirements			*MF02_BR02_UR09_NFR01, MF02_BR01_NFR01* *MF02_BR01_NFR02, MF02_BR02_UR09_NFR01*		
Input Parameters					
Attribute/Parameter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Received From***
Clusters of prosumers	*Groups of prosumers with same features*	*Float*	*csv*	*E. Abs >= 0 Freq = 15 min*	*Big Data Clustering at Multiple Scales*
KPIs	*Behavioral Indicators*	*String*	*csv*	*-*	*Cassandra DB/InfluxDB*
Output Parameters					
Attribute/Parameter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Sent To***
Segments of prosumers	*Sub-groups of prosumers with specific attributes values*	*Float*	*csv*	*E. Abs >= 0 Freq = 15 min*	*Cassandra DB/InfluxDB*
Software Requirements/Development Language			*TCP/IP connectivity* *MQTT/AMPQ, HTTP as transport protocols*		
Hardware Requirements			*Cloud-based system*		

Communications	*Input data should be provided either by csv files or by API requests.* *Output data shall be offered by API requests or by a context brokering service.*
Status of the development of the component	Initial component prototype implemented

Table 34: VPP and DR Services Optimization Engine

Name of New Component/Service:	***VPP and DR Services Optimization Engine***
Type:	*Component*
Functionality:	*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)*
Provided Services	*DR Optimization and decision support system*
Input Connections & Interfaces: From which components it receives input	• *Virtual Power Plants Generation, Modelling & Forecasting* • *Baseline Flexibility Estimation* • *Electricity consumption/Production Forecasting* • *PV/RES Degradation & Trend Analysis* • *Cassandra DB*
Output Connections & Interfaces: To which components it sends the results	• *DSS & DR Strategies Optimization*

Functional Requirements			*MF02_BR04_UR01_FR01, MF02_BR04_FR02* *MF02_BR04_FR03, MF02_BR04_FR04* *MF02_BR04_FR05, MF02_BR04_FR06* *MF02_BR04_FR07, MF02_BR04_FR08* *MF02_BR04_FR09, MF02_BR04_FR010*		
Non-Functional Requirements			*MF02_BR04_UR01_NFR01, MF02_BR04_UR02_NFR02*		
Input Parameters					
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)-Starttimestap (string)-endtimestamp (string)-Energy consumption flexibility array (float[])-monetary cost during starttime and endtime (float []), comfort cost during starttime and*	*XML,JSON or CSV*	*Greater than 0, every 15 min*	*Baseline Flexibility Estimation*

		end time (float[])			
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*
Output Parameters					
Attribute/Para-meter	***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 (flaot[])*	*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 (string) – array*	*XML,JSON or CSV*	*Greater than 0, every 15 min*	

		of values (flaot[])			

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.* *In case it needs any special sensor that is included in the sensor specification, it can be included also here as a reference.*
Communications	*Ethernet or WiFi Connectivity*
Status of the development of the component	Initial component prototype implemented

Table 35: Distributed Ledger

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*
Input Connections & Interfaces: From which components it receives input	• *Field Data Aggregation* *REST API*
Output Connections & Interfaces: 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				
Input Parameters					
Attribute/Para-meter	***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		
Output Parameters					
Attribute/Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Sent To***
Same as above (it will be a storage component, so it is expected to provide the same data entered)					
Software Requirements/Development Language	*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*				

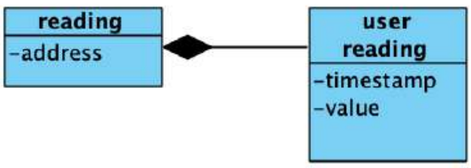
	infrastructure is based on clients written in Rust programming language
Hardware Requirements	*the component runs on a blockchain infrastructure*
Communications	*http; rpc;*
Status of the development of the component	Initial component prototype implemented
High-level Class Diagram	

Table 36: Blockchain-driven control for LV networks (flexibility management)

Name of New Component/Service:	Blockchain-driven control for LV networks (flexibility management)
Type:	Component
Functionality:	Provide a set of smart contracts modelling and evaluating the production/consumption profiles with respect to the flexibility orders and DR signals
Provided Services	*Flexibility tracking and control;* *Flexibility request disaggregation;*

Input Connections & Interfaces: From which components it receives input	• *Distributed Ledger* • Electricity Consumption/*Production* Forecasting • Baseline Flexibility Estimation • *DSS & DR Strategies Optimization* **REST API**: <host>:<port>/edream-flexibility-market
Output Connections & Interfaces: To which components it sends the results	• Closed Loop DR Verification Engine • *DSS & DR Strategies Optimization*
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_BR02_UR01_NFR01, MF03_BR02_UR02_NFR02, MF03_BR02_UR03_NFR03,

Input Parameters

Attribute /Para-meter	**Short Description**	**Data Type**	**Data Format**	**Value Range & Frequency**	**Data Received From**

Monito-red Values	An array containing all the monitored values for a period	JSON	[{
			"timestamp": 1537803600000,
			"value": 908.483,
			"deviceMeasurementId": "18bb6e98-8429-4162-8236-cc4f231bb2a4",
			"deviceId": "4836bee7-bc42-48a9-9e4e-aa1ecc68e6d8",
			"property": "Power Consumption"
			}, {
				Range – depending on the monitored device
				Frequency: every5 min
					Distributed Ledger

			"timestamp": 1537341600000, "value": 910.54, "deviceMeasurementId" : "18bb6e98-8429-4162-8236-cc4f231bb2a4", "deviceId": "4836bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "property": "Power Consumption" }, ...]		
Registe-red Energy Assets	The energy asset registered as token in the chain	TOKEN	-	**Range** – **Frequency**:	
Baseline & *Flexibili-ty* values		JSON	[{ "timestamp": 1537803600000, "value": 905.5, "deviceMeasurementId": "12bb6e98-8429-4162-8236-cc4f231bb2a4", "deviceId": "4836bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "granularity": "DAYAHEAD", "property": "Baseline Flexibility" },...]	**Range** – depending on the forecasted property **Frequency**: every 30 min, 1 hour	Baseline Flexibility Estimation
Predic-ted Values	An array containing the device and the measuremen t the forecast is referring to, together with the forecasted value	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	Electricity Production/*Con-sumption* Forecasting

			"entityDeviceId": "ab7d658d-84cd-4662-b573-74db92a297f2", "deviceMeasurementId": "c0a735f0-b6fe-47e6-b951-79910cd0e822", "profileGranularityMinutes": 60 , "predictionGranularity": "DAYAHEAD", "property": "ENERGY CONSUMPTION" }		
Flexibility offers/ orders					*DSS & DR Strategies Optimization*
Output Parameters					
Attribute /Parameter	**Short Description**	**Data Type**	**Data Format**	**Value Range & Frequency**	**Data Sent To**
Smart Contracts *evaluation* of actual *profile* delivered	Evaluate the energy flexibility delivered	-	Evaluation inside the contract. The result is the updated state of the contract	-	Closed Loop DR Verification Engine *DSS & DR Strategies Optimization*

Software Requirements/Development Language	• 64 bit Linux / Windows OS • Internet connection for package management • Ethereum – Parity , Solidity • NodeJS, MySql
Hardware Requirements	• Intel/AMD processor with at least 4 cores and 2.5GHz base frequency • 8 GB RAM

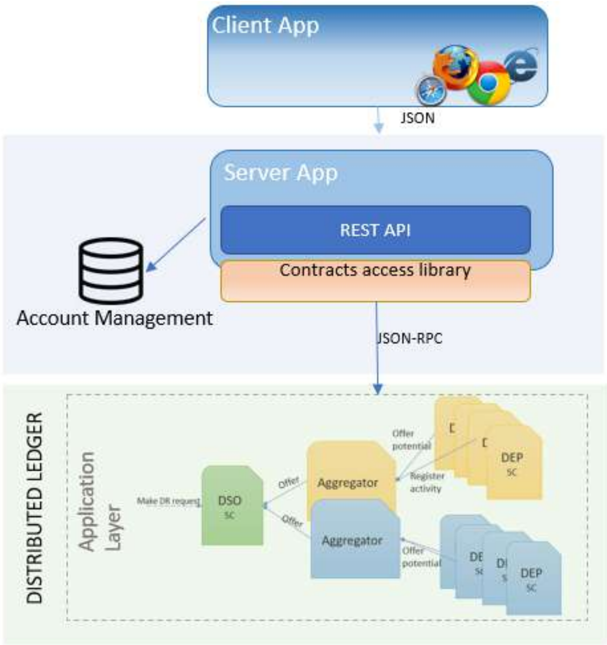
	• HDD/SSD with at least 100GB free space • Internet connection for package management
Communications	REST API, JSON RPC
Status of the development of the component	Initial component prototype implemented
High-level Class Diagram	

Table 37: Secured Blockchain-driven Energy Market

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

Provided Services	*Peer to peer energy trading;* *Energy bids/offers matching*
Input Connections & Interfaces: From which components it receives input	• *Distributed Ledger* • Electricity Consumption/*Production* Forecasting • *Local/Remote HMIs*
Output Connections & Interfaces: To which components it sends the results	• Closed Loop DR Verification Engine • *Local/Remote HMIs* 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, MF03_BR04_UR02_NFR02, MF03_BR04_UR03_NFR03

Input Parameters

Attribute /Para-meter	**Short Description**	**Data Type**	**Data Format**	**Value Range & Frequency**	**Data Received From**
Registe-red Energy Assets	The energy asset registered as token in the chain	TOKEN	-	**Range** – **Frequency**:	*Distributed Ledger*

Bid/Offer Actions	Place orders on the market contract	JSON	{ "orderSide": "BID", "prosumerAddress":"0xAA21803000499f1b58C67F4DA7083AFA2ee37090", "timestamp":"123453123", "tokenId":10, "metadata": { "startTimeToken": 0, "endTimeToken": 1, "energyType":"GREEN", "producer":"0xAA21803000499f1b58C67F4DA7083AFA2ee37090"} "quantity":12321 "price":35, }	**Range:** **Frequency**: 30-60 min	HMI
Predic-ted Values	An array containing the device and the measurement the forecast is referring to, together with the forecasted value	JSON	{ "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" }	**Range** – depending on the forecasted property **Frequency**: every 30 min, 1 hour	*Electricity Production/Consumption Forecasting*
<u>Output Parameters</u>					
Attribute /Para-meter	**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			• 64 bit Linux / Windows OS • Internet connection for package management • Ethereum – Parity , Solidity • NodeJS, MySql		
Hardware Requirements			• 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		

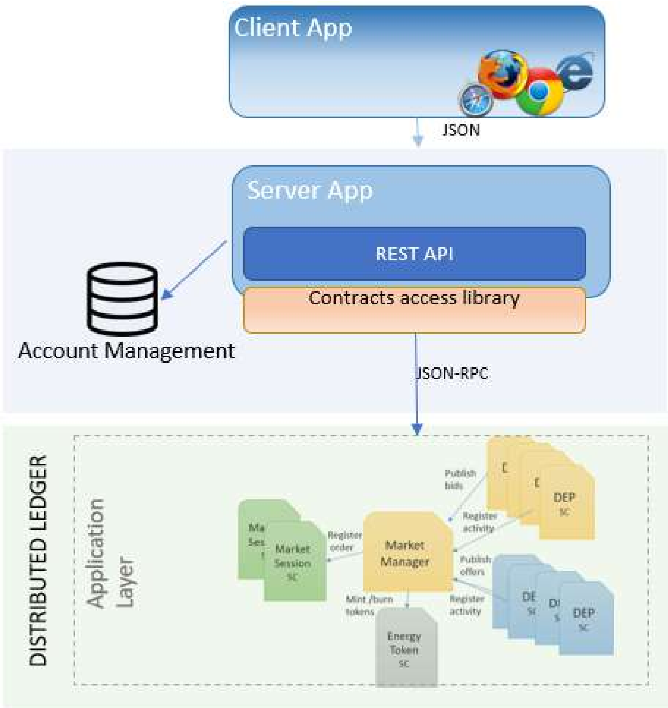
Communications	REST API, JSON RPC
Status of the development of the component	Initial component prototype implemented
High-level Class Diagram	

Table 38: Closed loop DR Verification Engine

Name of New Component/Service:	***Closed loop DR Verification Engine***
Type:	Component

Functionality:	This component will be used to validate transactions previously agreed on the marketplace. 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.
Provided Services	*Verification, Financial settlement (incentives).*
Input Connections & Interfaces: From which components it receives input	• *Distributed* Ledger • *Blockchain-driven control for LV networks* • *Blockchain-driven Energy Market* • *Cassandra DB* Rest API
Output Connections & Interfaces: To which components it sends the results	• *Distributed Ledger* • *Local/Remote HMIs* Rest API
Functional Requirements	MF03_BR02_UR01_FR01, MF03_BR02_UR02_FR02 MF03_BR02_UR03_FR03
Non-Functional Requirements	*MF03_BR02_UR01_NFR01, MF03_BR02_UR02_NFR02,*
Input Parameters	

Attribute/Para-meter	Short Description	Data Type	Data Format	Value Range & Frequency	Data Received From
forecast	Forecasted value at target timestamp	number	int	Each of the monitored timestamp	*Cassandra DB*
request	Requested value at target timestamp	number	int		*Blockchain-driven control for LV networks (flexibility management)*
reading	measured value at target timestamp	number	int		*Distributed Ledger*
range	Tolerance range for the request	number	uint	Fixed for the contract duration	*Distributed Ledger*
multiplier	incentive/penalty multiplier	number	float		
Clearing price, Matched energy produc-tion and demand					
Output Parameters					
Attribute/Para-meter	**Short Description**	**Data Type**	**Data Format**	**Value Range & Frequency**	**Data Sent To**
incentive/penalty	Incentive or penalty to be applied for each timestamp	number	int	Each of the monitored timestamp	*Distributed Ledger Local/Remote HMIs*

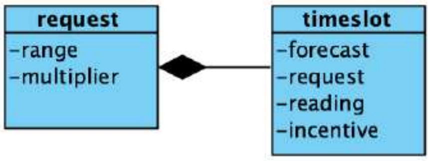
Software Requirements/Development Language	Solidity (for the smart contract development)
Hardware Requirements	*the component runs on a blockchain infrastructure*
Communications	http; rpc
Status of the development of the component	Initial component prototype implemented
High-level Class Diagram	

Table 39: DSS (Decision Support System) & DR Strategies Optimization

Name of New Component/Service:	***DSS (Decision Support System) & DR Strategies Optimization***
Type:	*Component*
Functionality:	*An interface for analyzing and preparing energy trend and flow information for display by a HMI. Interface for inputting variables for optimizing DR services.*
Provided Services	*Provide a set of feasible solutions to stakeholders*

Input Connections & Interfaces: ***From which components it receives input***	• *VPP and DR Services Optimization Engine* • *PV/RES Degradation and Trend Analysis* • *Baseline Flexibility Estimation* • *Big Data Clustering at multiple scales* • *Cassandra DB*
Output Connections & Interfaces: ***To which components it sends the results***	• *End-user's web interface*
Functional Requirements	*MF02_BR04_UR01_FR01, MF02_BR04_FR02* *MF02_BR04_FR03, MF02_BR04_FR04*
Non-Functional Requirements	*MF02_BR04_NFR01, MF02_BR04_NFR02, MF02_BR04_NFR03, MF02_BR04_NFR04, MF02_BR04_NFR05, MF02_BR04_NFR06*

Input Parameters

Attribute/Para-meter	***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*

Output Parameters					
Attribute/Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Sent To***
Set of feasible solutions	*Optimzed DR profile*	*Starttime (string) – end time (string)- values of optimized DR profile (float)*	*XML,CSV,JSON*	*Greater than 0, 15 min*	*End-user's web interface*

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.* *In case it needs any special sensor that is included in the sensor specification, it can be included also here as a reference.*
Communications	*Ethernet or WiFi Connectivity*
Status of the development of the component	Initial component prototype implemented

Table 40: DR Aerial Survey Toolkit

Name of New Component/Service:	***DR Aerial Survey Toolkit***
Type:	*Component*
Functionality:	*Estimate the demand response potential over a wide area of building assets based on the energy demand profile assessment and the overall energy*

<table>
<tr><td colspan="3"></td><td colspan="3">performance of the buildings through optical, thermal and LIDAR images</td></tr>
<tr><td colspan="3">Provided Services</td><td colspan="3">Provide data for energy assessment of buildings and specific zones</td></tr>
<tr><td colspan="3"><u>Input Connections & Interfaces:</u> From which components it receives input</td><td colspan="3">• Drone camera mount
• LiDAR Ethernet port
• Skyport API
• Multi-building DR characterization
• Cassandra DB</td></tr>
<tr><td colspan="3"><u>Output Connections & Interfaces:</u> To which components it sends the results</td><td colspan="3">• End-user’s web interface</td></tr>
<tr><td colspan="3">Functional Requirements</td><td colspan="3">MF01_BR01_UR01_FR01, MF01_BR01_UR01_FR02
MF01_BR01_FR03, MF01_BR01_FR04
MF01_BR01_FR05, MF01_BR01_FR06
MF01_BR01_FR07</td></tr>
<tr><td colspan="3">Non-Functional Requirements</td><td colspan="3">MF01_BR02_UR01_NFR01</td></tr>
<tr><td colspan="6"><u>Input Parameters</u></td></tr>
<tr><td>Attribute/Para-meter</td><td>Short Description</td><td>Data Type</td><td>Data Format</td><td>Value Range & Frequency</td><td>Data Received From</td></tr>
<tr><td>Aggregated and processed data from aerial surveys</td><td></td><td></td><td></td><td></td><td>• Cameras
• Multi-building DR Characterization</td></tr>
<tr><td>Weather data, Smart meters’ data</td><td></td><td></td><td></td><td></td><td>• Cassandra DB</td></tr>
<tr><td colspan="6"><u>Output Parameters</u></td></tr>
</table>

Attribute/Parameter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Sent To***
Estimated DR potential for groups of assets or buildings					• *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			Initial component prototype implemented		

Table 41: Graph-Based Analytics

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*
***Input Connections & Interfaces:* From which components it receives input**	• *Cassandra DB*

***Output Connections & Interfaces:* To which components it sends the results**	• *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	*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*

Input Parameters

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.	*String*	*XML, JSON*	-	

profits/loses, congestion improvement achieved etc.)				
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*	*-*	
Output Parameters				
Attribute/Parameter ***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Sent To***
Data model for optimal selection selection of the portfolio in spatial-temporal and operation views	*Float*	*XML, JSON*	*-*	*HMI*
Specific characteristics of the selected portfolio (spatial-temporal-operational) along with relevant KPIs	*String*	*XML, JSON*		
Improved DR activation signals	*Float*	*XML, JSON*		

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.*
Communications	*Ethernet or WiFi Connectivity* *Web Interface*
Status of the development of the component	Initial component prototype implemented

Table 42: Forecasting Tool

Name of New Component/Service:	***Forecasting Tool***				
Type:	*Component*				
Functionality:	*Forecasting Tool with interactive visualization framework enabling the agregator to be informed with a standard pre-defined frequency rate about 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	• *Electricity consumption/Production Forecasting* • *PV/RES Degradation & Trend Analysis* • *Cassandra DB*				
Output Connections & Interfaces: To which components it sends the results	• *End-user's web interface*				
Functional Requirements	*MF01_BR02_FR01, MF01_BR02_FR02* *MF01_BR02_FR03, MF01_BR02_FR04*				
Non-Functional Requirements	*MF01_BR02_UR01_NFR01, MF01_BR02_UR02_NFR02, MF01_BR02_UR03_NFR03,*				
Input Parameters					
Attribute/Parameter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Received From***

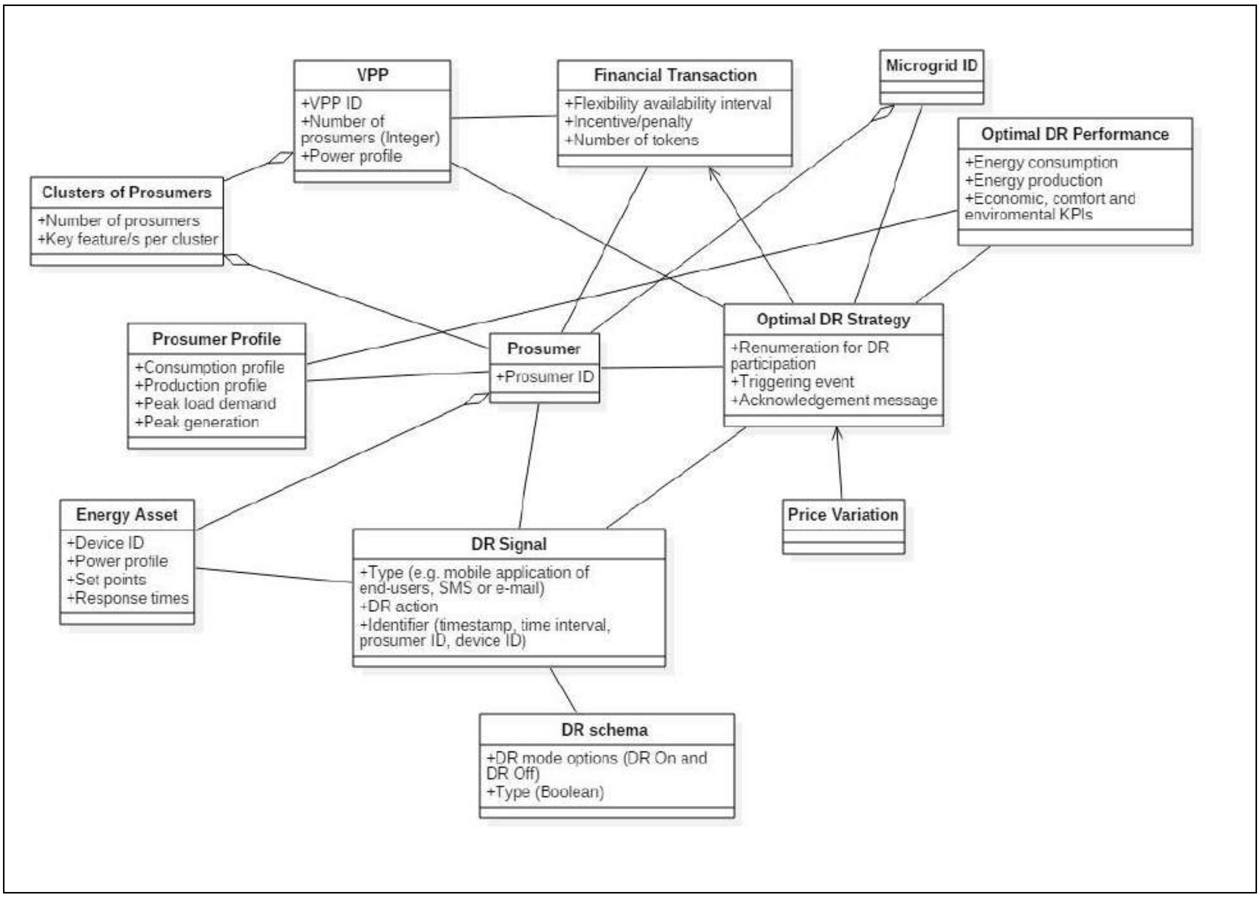
Forecasted consumption/production prosumers data	*Near real time forecasted values for intraday and day-ahead planning*	*Float*	*XML,JSON*	*Every 10 min (5min or 1 hour are also possible)*	*Electricity consumption/Production Forecasting*
Short-term and long-term generation forecasting					*PV/RES Degradation & Trend Analysis*
Weather data, Smart meters' data					*Cassandra DB*
Output Parameters					
Attribute/Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Sent To***
Visualizations of the input data					
Software Requirements/Development Language			*The software requirement is python libraries and runtime environment. Python will be used as programming language.*		
Hardware Requirements			*A Windows/Linux based PC with administrator right and credentials.* *Web Server*		
Communications			*Ethernet or WiFi Connectivity* *Web Interface*		
Status of the development of the component			Initial component prototype implemented		

Table 43: HMI

Name of New Component/Service:	***HMI***
Type:	*Component*
Functionality:	*An interactive multi-level and multi-factor visualization framework enabling the end-user to interact with the components of the core platform.*
Input Connections & Interfaces: From which components it receives input	• *Core platform's components* • *End-user web interface*
Output Connections & Interfaces: To which components it sends the results	• *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*

Input Parameters					
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.					
Output Parameters					
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	*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.* *Web Server*
Communications	*Ethernet or WiFi Connectivity* *Web Interface*
Status of the development of the component	Initial component prototype implemented
High-level Class Diagram	


VPP
+VPP ID
+Number of prosumers (Integer)
+Power profile
Financial Transaction
+Flexibility availability interval
+Incentive/penalty
+Number of tokens
Microgrid ID
Optimal DR Performance
+Energy consumption
+Energy production
+Economic, comfort and enviromental KPIs
Clusters of Prosumers
+Number of prosumers
+Key feature/s per cluster
Prosumer Profile
+Consumption profile
+Production profile
+Peak load demand
+Peak generation
Prosumer
+Prosumer ID
Optimal DR Strategy
+Renumeration for DR participation
+Triggering event
+Acknowledgement message
Energy Asset
+Device ID
+Power profile
+Set points
+Response times
DR Signal
+Type (e.g. mobile application of end-users, SMS or e-mail)
+DR action
+Identifier (timestamp, time interval, prosumer ID, device ID)
Price Variation
DR schema
+DR mode options (DR On and DR Off)
+Type (Boolean)

8 Conclusion

This report has presented the updated version architectural design of the eDREAM system along with the respective functional and technical specifications and interfaces. The updates have emerged from the work done during the technical WPs 3, 4, 5 and 6. No major modifications are noticed to the overall eDREAM system architecture. The involvement of the developers to the architecture refinement process has been significant and resulted in a more coherent architecture definition and its architectural elements, that also has encapsulated the view of developers.

Since the previous versions of this deliverable, the functionalities of several tools have been merged or incorporated by other ones, namely, the functionalities of VPP and active Microgrid Flexibility Profiling have been incorporated by the VPP generation, modelling and forecasting and baseline flexibility estimation. This has also affected the High-level Use Cases 2 and 3, i.e. HL-UC03_LL-UC02 and HL-UC03_LL-UC03 and the changes can be seen in both this deliverable and also Deliverable D2.9, where the Use Case are described in more detail [8]. With respect to the deployment view of the Terni Pilot, the charging stations installed are now described in more detail and technical specifications are provided. Additionally, the non-functional requirements of the involved tools are collected, in order to be utilized for the validation phase of the project and more specifically, within the activities of WP7 and T7.2.

References

[1]. [eDREAM D2.4] – eDREAM Deliverable D2.4 – Requirement-Driven System Development V1. December 2018.
[2]. [eDREAM D2.5] – eDREAM Deliverable D2.5 – Requirement-Driven System Development V2. June 2019.
[3]. [eDREAM D2.1] – eDREAM Deliverable D2.1 – User group definitions, end-user needs, requirement analysis and deployment guidelines V1. August 2018.
[4]. [eDREAM D2.2] – eDREAM Deliverable D2.2 – Use Case analysis and application scenarios description V1, August 2018.
[5]. [eDREAM D2.6] – eDREAM Deliverable D2.6 – User group definitions, end-user needs, requirement analysis and deployment guidelines V2. August 2019
[6]. [eDREAM D2.8] – eDREAM Deliverable D2.6 – User group definitions, end-user needs, requirement analysis and deployment guidelines V3. June 2020
[7]. [eDREAM D2.7] – eDREAM Deliverable D2.2 – Use Case analysis and application scenarios description V2, August 2019
[8]. [eDREAM D2.9] – eDREAM Deliverable D2.2 – Use Case analysis and application scenarios description V3, June 2020.
[9]. [eDREAM 3.1] – eDREAM Deliverable D3.1 – Electricity production/consumption forecasting techniques and tool V1. April 2019.
[10]. [eDREAM 3.5] – eDREAM Deliverable D3.5 – Electricity production/consumption forecasting techniques and tool V2. February 2020.
[11]. [eDREAM 4.5] – eDREAM Deliverable D4.5 – Specification for Improved Decision Making and DR optimization toolsets V2. April 2020.
[12]. [eDREAM 3.2] – eDREAM Deliverable D3.2 – Recommendations for baseline load calculations in DR programs V1. April 2019.
[13]. [eDREAM 3.6] – eDREAM Deliverable D3.6 – Recommendations for baseline load calculations in DR programs V2. April 2020.
[14]. [eDREAM 3.3] – eDREAM Deliverable D3.3 – Consumption flexibility models and aggregation techniques V1. May 2019.
[15]. [eDREAM 3.7] – eDREAM Deliverable D3.7 – Consumption flexibility models and aggregation techniques V2. May 2020.
[16]. [eDREAM 3.4] – eDREAM Deliverable D3.4 – Arial 3D models and simulation procedures for DR estimation V1. May 2019.
[17]. [eDREAM 3.8] – eDREAM Deliverable D3.8 – Arial 3D models and simulation procedures for DR estimation V2. August 2020.
[18]. [eDREAM D4.2] – eDREAM Deliverable D4.2 – Load profiles and customer clusters V1. May 2019.
[19]. [eDREAM D4.6] – eDREAM Deliverable D4.6 – Load profiles and customer clusters V2. May 2020.
[20]. [eDREAM 4.1] – eDREAM Deliverable D4.1 – Specification for Improved Decision Making and DR Optimization toolset V1. April 2019.
[21]. [eDREAM 5.1] – eDREAM Deliverable D5.1 – Blockchain platform for secure and distributed management of DR programs V1. May 2019.
[22]. [eDREAM 5.4] – eDREAM Deliverable D5.4 – Blockchain platform for secure and distributed management of DR programs V2. May 2020.

[23]. [eDREAM 5.2] – eDREAM Deliverable D5.2 – Self-enforcing smart contract for DR tracking and control V1. May 2019.
[24]. [eDREAM 5.5] – eDREAM Deliverable D5.5 – Self-enforcing smart contract for DR tracking and control V2. May 2020.
[25]. [eDREAM D4.3] – eDREAM Deliverable D4.3 – Multi-factor automated DR Optimization & Scheduling toolkit V1. May 2019.
[26]. [eDREAM D4.4] – eDREAM Deliverable D4.4 – Interactive Visualization framework for improving DR strategies V1. April 2019.
[27]. [eDREAM D4.7] – eDREAM Deliverable D4.7 – Multi-factor automated DR Optimization & Scheduling toolkit V2. May 2020.
[28]. [eDREAM D4.8] – eDREAM Deliverable D4.8 – Interactive Visualization framework for improving DR strategies V1. April 2020.
[29]. Agile Modelling, UML2 Use Case Diagrams, available under http://www.agilemodeling.com/artifacts/useCaseDiagram.htm (state on September 2013).
[30]. 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).
[31]. StarUML ,Modelling with Class Diagram, available under http://staruml.sourceforge.net/docs/user-guide(en)/ch05_2.htm
[32]. https://www.car.info/en-se/renault/zoe/zoe-2018-16314878/specs
[33]. https://www.car.info/en-se/nissan/leaf/leaf-86616/specs

Annex I: Functional & Non-Functional Requirements

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

Functional Requirements

Table 44: Electricity Consumption/Production Forecasting – FRs

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

Table 45: Virtual Power Plants Generation Modelling & Forecasting - FRs

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

Table 46: PV/RES Degradation & Trend Analysis - FRs

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

Table 47: Baseline Flexibility Estimation - FRs

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

Table 48: Multi-building DR characterization through thermal, optical and LIDAR information fusion - FRs

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)

Table 49: Load Profiling - FRs

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

Table 50: Big Data Clustering at Multiple Scales - FRs

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

Table 51: Customer Segmentation - FRs

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

Table 52: VPP and DR Services Optimization Engine - FRs

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

Table 53: Distributed Ledger - FRs

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

Table 54: Blockchain-driven control for LV networks - FRs

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

Table 55: Secured Blockchain-driven Energy Market - FRs

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

Table 56: Closed loop DR Verification Engine - FRs

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

Table 57: Graph-based Analytics - FRs

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

Table 58: Decision Support System & DR Strategies Optimization – FRs

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

Table 59: DR Aerial Survey Toolkit – FRs

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

Table 60: Forecasting Tool – FRs

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

Table 61: HMI - FRs

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

Table 62: EVSEs and EV fleet monitoring – FRs

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

Table 63: Electric meters, edge, and field device electric measures – FRs

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

Table 64: Electricity Consumption/Production Forecasting – NFRs

Component: Electricity Consumption/Production Forecasting	
Non-Functional Requirement ID	***Description***
MF01_BR02_UR01_NFR01	Scale with amount of historical data and number of prosumers

Table 65: Virtual Power Plants Generation Modelling & Forecasting - NFRs

Component: Virtual Power Plants Generation Modelling & Forecasting	
Non-Functional Requirement ID	***Description***
MF02_BR03_UR01_NFR01	Scale with the number of prosumers considered in the optimization

Table 66: PV/RES Degradation & Trend Analysis – NFRs

Component: PV/RES Degradation & Trend Analysis	
Non-Functional Requirement ID	***Description***
MF01_BR04_UR01_NFR01	Scale with the amount of the historical data of the PV system considered in the estimation
MF01_BR04_UR02_NFR02	The component should be able to be integrated with a user interface and able to request data from other components of the eDREAM platform
MF01_BR04_UR03_NFR03	It should be possible to package up the component into a portable, self-sufficient package, which can run on a different hosts

Table 67: Baseline Flexibility Estimation - NFRs

Component: Baseline Flexibility Estimation	
Non-Functional Requirement ID	***Description***
MF01_BR03_UR01_NFR01	The system must be able to process data gathered from customer electricity consumption, in order to forecast a consumption or a next day. in important provide the consumption data from the past days in 30 minutes intervals

Table 68: Multi-building DR Characterization through thermal, optical and LIDAR information fusion – NFRs

Component: Multi-building DR Characterization through thermal, optical and LIDAR information fusion	
Non-Functional Requirement ID	***Description***
MF01_BR01_UR01_NFR01	Scale with the number of prosumers considered in the optimization

Table 69. Load Profiling - NFRs

Component: Load Profiling	
Non-Functional Requirement ID	***Description***
MF02_BR02_UR09_NFR01	The system should provide ingestion mechanisms to collect data at different ingestion rates.
MF02_BR01_UR02_NFR01	The system must 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, pre-processed, other modules outputs, devices etc.).
MF02_BR01_NFR02	The systems shall include modules that implement near real-time data processing techniques, ensuring response within specified time constraints. This requirement indicates that there are no substantial delays and quantification of the system responsiveness will depend on the specific use-case and context.

Table 70: Big Data Clustering at Multiple Scales - NFRs

Component: Big Data Clustering at Multiple Scales	
Non-Functional Requirement ID	***Description***
MF02_BR01_UR01_NFR01	The system must be able to execute different clusterization processes in parallel, useful for the evaluation of the available flexibility capacity for different entities (generators, loads, EVs).
MF02_BR01_UR02_NFR02	It should be possible to decompose the solution in different micro-services, so to better run different processes in several machines.
MF02_BR01_UR03_NFR03	Information types produced, consumed and transformed shall be documented in an information model which shall also include the relationships between information types.
MF02_BR01_UR04_NFR04	The system shall be able to process data at different levels in order to integrate external processes or modules with computing resources. This requirement will allow the aggregation of concurrent data exchanges with big number of sources or devices.
MF02_BR01_UR05_NFR05	The system shall be able to exchange data with a great number of devices and, at the same time, preserving its computational capacity. This requirement will need proper modular and distributed features.
MF02_BR01_UR06_NFR06	Information model: 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, submit and generation times, ownership.

Table 71: Customer Segmentation – NFRs

Component: Customer Segmentation	
Non-Functional Requirement ID	***Description***
MF02_BR02_UR09_NFR01	The systems shall include modules that implement near real-time data processing techniques, ensuring response within specified time constraints. This requirement indicates that there are no substantial delays and quantification of the system responsiveness will depend on the specific use-case and context.
MF02_BR01_NFR01	The system shall include modules that implement batch data processing techniques in order to ingest historical data streams that will further allow extracting know-how and derived information from eDREAM resources.
MF02_BR01_NFR02	The system must be able to ingest data from devices and services that represent data using different information models.
MF02_BR02_UR09_NFR02	The system should provide ingestion mechanisms to collect data at different ingestion rates.
MF02_BR02_UR09_NFR03	The system must 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, pre-processed, other modules outputs, devices etc.).

Table 72: VPP and DR Services Optimization engine - NFRs

Component: Customer Segmentation	
Non-Functional Requirement ID	***Description***
MF02_BR04_UR01_NFR01	The component should be able to be integrated with a user interface and able to request data from other components of the eDREAM platform
MF02_BR04_UR02_NFR02	It should be possible to package up the component into a portable, self-sufficient package, which can run on a different hosts.

Table 73: Distributed Ledger – NFRs

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	The application shall grant the access only to authorized users and grant the non-repudiability.

Table 74: Blockchain-driven Control for LV networks - NFRs

Component: Blockchain-driven control for LV networks	
Non-Functional Requirement ID	***Description***
MF03_BR02_UR01_NFR01	Ensure scalability with high number of energy transactions
MF03_BR02_UR01_NFR02	Store transactions in a secured and tamperproof manner
MF03_BR02_UR01_NFR03	The application shall grant the access only to authorized users and grant the non-repudiability.

Table 75: Secured Blockchain-driven Energy Market – NFRs

Component: Secured Blockchain-driven Energy Market	
Non-Functional Requirement ID	***Description***
MF03_BR04_UR01_NFR01	Ensure scalability with high number of energy transactions
MF03_BR04_UR02_NFR02	Store transactions in a secured and tamperproof manner
MF03_BR04_UR03_NFR03	The application shall grant the access only to authorized users and grant the non-repudiability.

Table 76: Closed loop DR Verification Engine – NFRs

Component: Closed loop DR Verification Engine	
Non-Functional Requirement ID	***Description***
MF03_BR02_UR01_NFR01	Store transactions in a secured and tamperproof manner
MF03_BR02_UR02_NFR02	The application shall grant the access only to authorized users and grant the non-repudiability.

Table 77: EVSEs and EV fleet monitoring – NFRs

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

Table 78: Graph-based Analytics - NFRs

Component: Graph-based Analytics	
Non-Functional Requirement ID	***Description***
MF01_BR05_BR10_MF03_BR05_NFR01	The User Interface shall have a user friendly look, fully customized to the needs of different stakeholders
MF01_BR05_BR10_MF03_BR05_NFR02	The User Interface shall provide a user interface offering maps Visualization for a more concrete analysis
MF01_BR05_BR10_MF03_BR05_NFR03	The User Interface shall be able to allow an easy discoverability of the actions available
MF01_BR05_BR10_MF03_BR05_NFR04	The User Interface shall be tailored to the end user needs
MF01_BR05_BR10_MF03_BR05_NFR05	The messages provided by the system must be clear and easy to understand
MF01_BR05_BR10_MF03_BR05_NFR06	The User Interface must be simple and intuitive
MF01_BR05_BR10_MF03_BR05_NFR07	End users will have multiple interfaces to the whole system

Table 79: HMIs – NFRs

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

Table 80: DSS (Decision Support System) & DR Strategies Optimization – NFRs

Component: DSS (Decision Support System) & DR Strategies Optimization	
Non-Functional Requirement ID	***Description***
MF02_BR04_NFR01	The system should be able to be integrated with an interface and communicate with other DSS related components.
MF02_BR04_NFR02	The system should have the lowest possible response times between various operational streams, ensuring its fast and uninterrupted normal operation.
MF02_BR04 _NFR03	The mean time value between failures should be the maximum possible. The duration of the failures should be the lowest possible.
MF02_BR04_NFR04	The system should be able to scale up to a great number of assets in case of a prosumer, a large number of prosumers/consumers in case of Aggregator and buses of a distribution network in case of a DSO. In all of the aforementioned cases the system should function without any suffering on its overall operation.
MF02_BR04_NFR05	The user data should be confidential ensuring the users' privacy and anonymity, aligning with GDPR regulations.
MF02_BR04_NFR06	The system design (HMI) should function aiming to provide high quality ease of use, smooth navigation and a fine-grained user interaction experience with respect to all the involved users. This involves a User Interface that provides appropriate messages (feed-back) to the navigator, while remaining intuitive and easy to comprehend, pertaining its simplicity in its design.

Table 81: DR Aerial Survey Toolkit - NFRs

Component: DR Aerial Survey Toolkit	
Non-Functional Requirement ID	***Description***
MF01_BR01_UR01_NFR01	The drone surveyed material's data should be readable by the visualization component

Table 82: Forecasting Tool – NFRs

Component: Forecasting Tool	
Non-Functional Requirement ID	***Description***
MF01_BR02_UR01_NFR01	Scale with amount of historical data and number of prosumers
MF01_BR02_UR02_NFR02	The end-users should be able to flow through the forecasting tool without being interrupted
MF01_BR02_UR02_NFR03	Tool front end must be simple and intuitive

Annex II: Architectural Specifications Templates

Table 83: Functional and Non-Functional Requirements Template

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

Table 84: Architectural Components Detailed Specifications Template

Name of New Component/Service:	*<please write here the name of the architectural element e.g. Baseline Flexibility Estimation>*
Type:	*<Component, Software, Device etc.>*
Functionality:	*<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.>*
Provided Services	*<please mention the services provided by the component>*
***Input Connections & Interfaces:* From which components it receives input**	*<please write the components from which it receives input (input dependencies) and mention also the available connection interfaces e.g. API etc.>*
***Output Connections & Interfaces:* To which components it sends the results**	*<please write the components to which it sends the results (output dependencies) and mention also the available interfaces e.g. API etc.>*
Functional Requirements	*<write the functional requirements that the module satisfies, mention the respective IDs from the relevant template>*

Non-Functional Requirements				*<write the non-functional requirements that the module satisfies, mention the respective IDs from the relevant template>*	
Input Parameters					
Attribute/Para-meter	***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>*
Output Parameters					
Attribute/Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Sent To***

<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>*
Software Requirements/Development Language	*<specify any software requirements related to the architectural element, explain the Programming Language that is used during the development of the component>*				
Hardware Requirements	*<specify what hardware requirements are of the module, give specifications about the hardware requirements which are necessary for the best functionality of the component>* *In case it needs any special sensor that is included in the sensor specification, it can be included also here as a reference.*				
Communications	*<address specific communication requirements either for data input or for data output>*				
Status of the development of the component	*<specify if the component is "already developed" or "partially developed" or "to be developed from scratch">*				

Annex III: eDREAM Sensors/Gateways/Infrastructure Specifications Template

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

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

Annex IV: Field Devices APIs

Table 85: Smart Meters API

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/RmsFormated Voltage L1-N/value", "LD01/1-1-32-7-0-255/-2"], ["wallya1/RmsFormated Voltage L2-N/value", "LD01/1-1-52-7-0-255/-2"], ["wallya1/RmsFormated 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/RmsFormated Current L1-N/value", "LD01/1-1-31-7-0-255/-2"], ["wallya1/RmsFormated Current L2-N/value", "LD01/1-1-51-7-0-255/-2"], ["wallya1/RmsFormated 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/ActiveFormated Power Phase L1/value", "LD01/1-1-36-7-0-255/-2"], ["wallya1/ActiveFormated Power Phase L2/value", "LD01/1-1-56-7-0-255/-2"], ["wallya1/ActiveFormated Power Phase L3/value", "LD01/1-1-76-7-0-255/-2"], ["wallya1/ActiveFormated 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"], ["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/FundamentalFormated Reactive Power Phase L1/value", "LD01/1-1-23-7-0-255/-2"], ["wallya1/FundamentalFormated Reactive Power Phase L2/value", "LD01/1-1-43-7-0-255/-2"], ["wallya1/FundamentalFormated Reactive Power Phase L3/value", "LD01/1-1-63-7-0-255/-2"], ["wallya1/FundamentalFormated 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"]
Allowed HTTP Methods	N/A

Class type of GET response	{
	".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"
	},
	"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": {
	"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"
	},
	"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"
	},
	"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": {
	"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"
	}
	}

Table 86: Charging Station API

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	https://panel.spot-link.it/public/api/chargeboxes/{"chargeboxID":"24"}
	
	{
	"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"
	}

Table 87: Charging Session API

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	NULL: Charging Session List {"dateFrom":"Initialdate", "dateTo":"Finaldate"} : Charging Session related to a time interval {"chargeID":"sessionID"}: Single charging session info

	{"chargeboxID":"stationID"}: list of charging session of a single charging station {"chargeboxID":"stationID", "dateFrom":"Initialdate", "dateTo":"Finaldate"}: list of charging session of a single charging station in a time interval {"userID":"ID"}: list of charging session of a single EV user {"userID":"ID", "dateFrom":"Initialdate", "dateTo":"Finaldate"}: list of charging session of a single EV user in a time interval 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.
Allowed HTTP Methods	GET
Class type of GET response	https://panel.spot-link.it/public/api/historyCharges/{"chargeboxID":"24", "dateFrom":"2019-06-14 09:00:00", "dateTo":"2019-06-17 13:00:00"} { "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", "socketID": "38", "chargeboxID": "24", "nomePresa": "presa B" }] }

Table 88: Remote Start API

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	https://panel.spot-link.it/public/api/startChargebox/{"chargeboxID":"24", "socketID":"37"} "START_OK"

Table 89: Remote Stop API

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	https://panel.spot-link.it/public/api/stopChargebox/{"chargeboxID":"24", "socketID":"37"} "STOP_OK"

Table 90: Remote Power Output Setup API

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"

Table 91: Electric Vehicle API

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 • 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 =====> -LPeIpMRcbj6TZQCXE7c
Allowed HTTP Methods	GET
Class type of GET response	https://panel.spot-link.it/public/api/ev/{"vehicleID:"-LRGfW9NO0SNzKtnbxZ8"} {"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"}

Annex V: Architectural Components APIs

Table 92: Electricity Consumption Forecasting - API

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	*prosumerDeviceId*: the id of the prosumer's device that the energy predictions are requested for. *starttime:* 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 *starttime* is considered. *endtime:* 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 *endtime* is considered.
Allowed HTTP Methods	GET

Class type of GET response	Energy Profile
	{
	"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 CONSUMPTION"
	}

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	*prosumerDeviceId*: the id of the prosumer’s device that the energy predictions are requested for. *starttime:* 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 *starttime* is considered. *endtime:* 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 *endtime* is considered.
Allowed HTTP Methods	GET

Class type of GET response	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"
	}

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	*prosumerDeviceId*: the id of the prosumer’s device that the energy predictions are requested for. *starttime:* 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 *starttime* is considered. *endtime:* 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 *endtime* is considered.
Allowed HTTP Methods	GET
Class type of GET response	Energy Profile

	{
	"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"
	}

Table 93: Electricity Production Forecasting API

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	*prosumerDeviceId*: the id of the prosumer's device that the energy predictions are requested for. *starttime:* 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 *starttime* is considered. *endtime:* 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 *endtime* is considered.
Allowed HTTP Methods	GET
Class type of GET response	Energy Profile { "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 " }
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	*prosumerDeviceId*: the id of the prosumer's device that the energy predictions are requested for. *starttime:* 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 *starttime* is considered. *endtime:* 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 *endtime* is considered.
Allowed HTTP Methods	GET

Class type of GET response	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 PRODUCTION "
	}

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	*prosumerDeviceId*: the id of the prosumer's device that the energy predictions are requested for *starttime:* 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 *starttime* is considered. *endtime:* 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 *endtime* is considered.
Allowed HTTP Methods	GET

Class type of GET response	Energy Profile
	{
	"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"
	}

Table 94: Energy Flexibility Forecasting API

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	*prosumerDeviceId*: the id of the prosumer's device that the flexibility predictions are requested for. *starttime:* 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 *starttime* is considered. *endtime:* 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 *endtime* is considered.
Allowed HTTP Methods	GET

Class type of GET response	Flexibility Profile
	{
	"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"
	}

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	*prosumerDeviceId*: the id of the prosumer's device that the predictions are requested for. *starttime:* 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 *starttime* is considered. *endtime:* 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 an intra-day interval, then the end of the intra-day interval containing the *endtime* is considered.
Allowed HTTP Methods	GET
Class type of GET response	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" }

Table 95: Virtual Power Plants Generation Modelling & Forecasting - API

Virtual Power Plants Generation Modelling & 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	{
	"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"
	}
	}, ...],
	
	"goal": {
	type" : "ENERGY TRADING",
	"priceSignal": [
	{"value": 157167,
	"timestamp": "2018-09-08T00:00:00"
	},
	{"value": 157234,
	"timestamp": "2018-09-08T01:00:00"
	}, ...]
	}
	}

Response	{
	"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"
	}, ...
	]
	}

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

<table>
<tr><td>Request Body</td><td>{
"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"
}
}, ...],

"goal": {
type" : "CAPACITY BIDDING",
"priceSignal": [
{"value": 135000,
"timestamp": "2018-09-08T00:00:00"
},
{"value": 157000,
"timestamp": "2018-09-08T01:00:00"
}, ...]
}
}</td></tr>
<tr><td>Response</td><td>{
"coalitionId": "4726bee7-bc42-48a9-9e4e-aa1ecc68e6f1",
"selectedProsumers":[{</td></tr>
</table>

	"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"
	}, ...
	]
	}

Information Exchanged	***Construct VVP 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	{
	"prosumers": [{
	"prosumerId": "4836bee7-bc42-48a9-9e4e-aa1ecc68e6d8",
	"predictedProfile": { [
	{"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"
	}, ...]
	}
	}

Response	{
	"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": 54323,
	"timestamp": "2018-09-08T01:00:00"
	}, ...
	]
	}, ...]
	"totalEnergySpinned": [
	{"value": 150000,
	"timestamp": "2018-09-08T00:00:00"
	},
	{"value": 136000,
	"timestamp": "2018-09-08T01:00:00"
	}, ...

	] }
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	{
	"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"
	}
	}, ...],
	
	"goal": {
	"type": "DEMAND RESPONSE",
	"priceSignal": [
	{"value": 150000,
	"timestamp": "2018-09-08T00:00:00"
	},

	{"value": 112000, "timestamp": "2018-09-08T01:00:00" }, ...] } }

Response	{
	"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"
	}, ...
	]
	}

Table 96: Baseline Flexibility Estimation - API

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	prosumerDeviceId: the id of the prosumer's device that the baseline flexibility estimation is requested for starttime: the timestamp representing the start limit of the timeframe for which the improved forecasting is requested endtime: the timestamp representing the end limit of the timeframe for which the improved forecasting is requested
Allowed HTTP Methods	GET

Class type of GET response	Baseline Flexibility Estimation { "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"}…]

Table 97: PV/RES Degradation & Trend Analysis - API

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	prosumerDeviceId: the id of the prosumer's device that the Improved short-term forecasting of generation is requested for starttime: the timestamp representing the start limit of the timeframe for which the improved forecasting is requested endtime: the timestamp representing the end limit of the timeframe for which the improved forecasting is requested
Allowed HTTP Methods	GET
Class type of GET response	Improved short term forecasting { "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"}…]
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	prosumerDeviceId: the id of the prosumer's device that the Improved long-term forecasting of generation is requested for starttime: the timestamp representing the start limit of the timeframe for which the improved forecasting is requested endtime: the timestamp representing the end limit of the timeframe for which the improved forecasting is requested
Allowed HTTP Methods	GET
Class type of GET response	Improved long term forecasting { "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"}...]

Table 98: Load Profiling - API

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
	{
	"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" …] }

Table 99: Big Data Clustering at Multiple Scales - API

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
	{
	"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" …
	]
	}

Table 100: Customer Segmentation - API

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	/segments/daily/{prosumer_id} /segments/weekly/{prosumer_id} /segments/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 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" …
	]
	}

Table 101: VPP and DR Services Optimization Engine - API

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	coalitionId: the id of the VPP which containning the prosumers’ devices that are part of the optimization

	starttime: the timestamp representing the start limit of the timeframe for which the improved forecasting is requested endtime: the timestamp representing the end limit of the timeframe for which the improved forecasting is requested
Allowed HTTP Methods	GET

Class type of GET response	Optimized schduling
	{
	[
	"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"}…
	]
	"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"}…
	]
	…
	]
	}

Table 102: Distributed Ledger API

Subscription - API	
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	{ "SubscriptionId":"<id>", "Originator":"<host>", "ContextResponses":[{

	"ContextElement":{
	"type":"<type (String)>",
	"id":"<id>",
	"attributes":[
	{
	"name":"P",
	"Type":"W",
	"value":"<int>",
	"Metadatas":[
	{
	"name":"Timeinstant",
	"type":"ISO8601",
	"value":"<date>"
	}
	]
	}
	]
	},
	"statusCode":{
	"Code":"200",
	"reasonPhrase":"OK"
	}
	}
	]
	}

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	{
	"data":{
	"meter":{
	"serial_number":"SN",
	"manufacturer":"MANUFACTURER"
	}
	}
	}

Allowed HTTP Methods	POST
Class type of GET response	{ “Id":"<id>", }
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", }

Table 103: Secured Blockchain-driven Energy Market - API

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	{"orderIds": ["0x9a37ac6ecdc856ea6e87698d889217803b82965e52f4af852dfcaf08bdd996b5", ..]}
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	{
	"orderSide": "BID",
	"prosumerAddress":"0xAA21803000499f1b58C67F4DA7083AFA2ee37090",
	"timestamp":"123453123",
	"tokenId":10,
	"metadata": { "startTimeToken": 0,
	"endTimeToken": 1,
	"energyType":"GREEN",
	
	"producer":"0xAA21803000499f1b58C67F4DA7083AFA2ee37090";
	}
	"quantity":12321
	"price":35,
	}

Response	{ "orderId": "0x9a37ac6ecdc856ea6e87698d889217803b82965e52f4af852dfcaf08bdd996b5"}
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	[{ "id": "0x309911d2e38ff1649f7b6d39ef0e3fb33c87ae5d4fda6fa065ec9821c77fe2e3", "buyOrderId": "0x1c817f45dd2349c07b100ac0c9204d1652cea73006c0b000c9c39c4c40710d78", "sellOrderId": "0x95c8b70fc368ad3a3d7544715461d189616114cca00cabb97d2665a84e5d8b54", "prosumerBuyingAddress":"0x04fb94f5e2555d1e860462060337aa62ec6e919d", "prosumerSellingAddress":"0xAA21803000499f1b58C67F4DA7083AFA2ee37090", "timestamp":"123453123", "tokenId":10, "quantity":12321 "price":35, }, ...]
Response	

Table 104: Blockchain-driven control for LV networks - API

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	{ "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]

	}
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	{
	"reward":
	[20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20],
	"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]
	}

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	[{
	"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]}
	…
	]

Response	{
	"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"
	}

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]