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

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

DELIVERABLE: D2.4 Requirement-Driven System Development V1

Author: Antigoni Noula (CERTH),

Dimosthenis Ioannidis (CERTH)

Imprint

Requirement-Driven System Development V1, 12 2018

Contractual Date of Delivery to the EC:	31.12.2018
Actual Date of Delivery to the EC:	31.12.2018

Author(s):	Antigoni Noula (CERTH), Dimosthenis Ioannidis (CERTH)
Participant(s):	Giuseppe Raveduto (ENG), Vincenzo Croce (ENG), Giuseppe Mastandrea (E@W), Luigi D'Oriano (E@W), Mircea Bukur (KIWI), Ugo Stecchi (ATOS), Juan Sancho (ATOS), Alessio Cavadenti (ASM), Tudor Cioara (TUC), Ionut Anghel (TUC), Claudia Pop (TUC), Marcel Antal (TUC), Francesco Bellesini (EMOT), Michele Pagliaccia (EMOT), Benjamin Hunter (TU), Muneeb Dawood (TU), Charalampos Psarros (TU), Paul van Schaik (TU), Napoleon Bezas (CERTH), Christina Tsita (CERTH), Giorgos Sfikas (CERTH).

Project:	enabling new Demand Response Advanced, Market oriented and secure technologies, solutions and business models (eDREAM)
Work package:	Wp2 – User requirements, use cases and system specification
Task:	2.4 – Requirement-Driven System Development V1
Confidentiality:	public
Version:	1.0

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

Executive Summary

The deliverable D2.4 related to the Task 2.4 and entitled "Requirement-Driven System Development V1" is the fourth of WP2 reports describing the first set of stakeholders' requirements.

The aim of this deliverable is the definition of System requirements and technical specifications which a crucial and fundamental step for the successful design of the eDREAM system one of the main objectives of WP2 (User requirements, use cases and system specification) activities.

Initially, the methodological approach for system requirements elicitation is presented indicating the relevance of System requirements with User requirements (D2.1) and Use cases (D2.2). The technical specifications were extracted though internal elicitation using appropriate templates.

The Conceptual architecture of the eDREAM system is introduced in the beginning of Chapter 3. This is a high-level view on the overall architecture, describing the three major layers of the eDREAM platform, the Field Data Aggregation, the Core Backbone Platform and the Visualization Framework. The platform comprises also the Decentralized Repository for secure storage of the data exchanged between the modules of the platform and with external interfaces. Then, the structural view of the system is defined, presenting the different architectural components that deliver the system's functionalities. This view provides the system's decomposition into different components, demonstrating the dependencies between them and their functionalities.

Chapter 4 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 interaction between the components in each use case are represented with UML use case and sequence diagrams.

In Chapter 5, the development view is presented, describing how the architecture's development requirements. Here, components' technical requirements, programming technologies and use of existing tools are presented.

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.).

The Chapter 7 presents in total all the possible data flows within the eDREAM platform and the data models identified.

Finally, the Chapter 9 includes the detailed technical specifications of the core architectural elements focusing on the functionalities, inputs/outputs, interfaces and data types.

It should be mentioned that this report is going to be delivered according to eDREAM DoA in Month 12, but the architectural components and their detailed specifications will remain an open issue until all components and subsystems are developed and deployed and all modules have been integrated to the eDREAM platform. Thus, this document can be considered as a living document.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

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

1 Introduction

The purpose of this deliverable, as the technical output of the project, is to present the first 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 during the first 12 of the project and can be considered as a key input for the upcoming task in WP2 concerning requirement tracking (T2.5), as well as WP3, WP4, WP5, WP6 and WP7. Throughout the document, the main requirements and specifications of eDREAM platform are described in the scope of addressing the eDREAM objectives and innovation potential.

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

The purpose of this deliverable is to describe the first 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 and related constraints such as development methodology.

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

- Conceptual Architecture Design Process: within this part an overall view of the eDREAM architecture will be presented comprising the components, the interfaces between them and the connections with the external interfaces.

- Functional and Technical Specifications of Architectural Elements/Modules: the objectives of this section are the followings:

    - to provide a high level diagram of dependencies among the different parts of the framework;

    - to describe in detail the constraints of the system elements in terms of hardware and software resources, compatibility with standards, etc;

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 2.4 “Requirement-Driven System Development” consists of eight chapters in which the first consolidated version of System requirements, depencencies and technical specifications have been described as follows:

- **Chapter 1** presents the general description of the scope and objectives of the deliverable;
- **Chapter 2** describes the methodology which will be followed during the architectural design in order to derive the detailed functional and technical specifications of the eDREAM system. It presents the basic architectural design concepts and principles adopted towards the outline of the different phases and the definition of the architectural layers and elements that compose the eDREAM system.
- **Chapter 3** presents the conceptual architecture of the eDREAM system through a high-level diagram introducting the 3 main layers comprising the eDREAM system and analyses the 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 4** presents an 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 5** provides the development view of the eDREAM system. It displays an overview of the components’ software requirements and development languages.

- **Chapter 6** depicts the deployment view of the eDREAM system covering the hardware requirements of the architectural components and tools used.
- **Chapter 7** presents 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** and the templates used for the internal elicitation of requirements and technical specifications are included in **Annex II**.

2 Methodology

This section presents the approach and methodology that have been followed by Task 4.2 to define the first version of the architecture. Task 2.4 started on M3 and it will run continuously until M30. The first version of the architecture comprises the first consolidation of dependencies, inputs/outputs and specifications of architectural components. The next version is intended to present detailed information concerning the interfaces between the compopnnents 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 elecitation until M12:

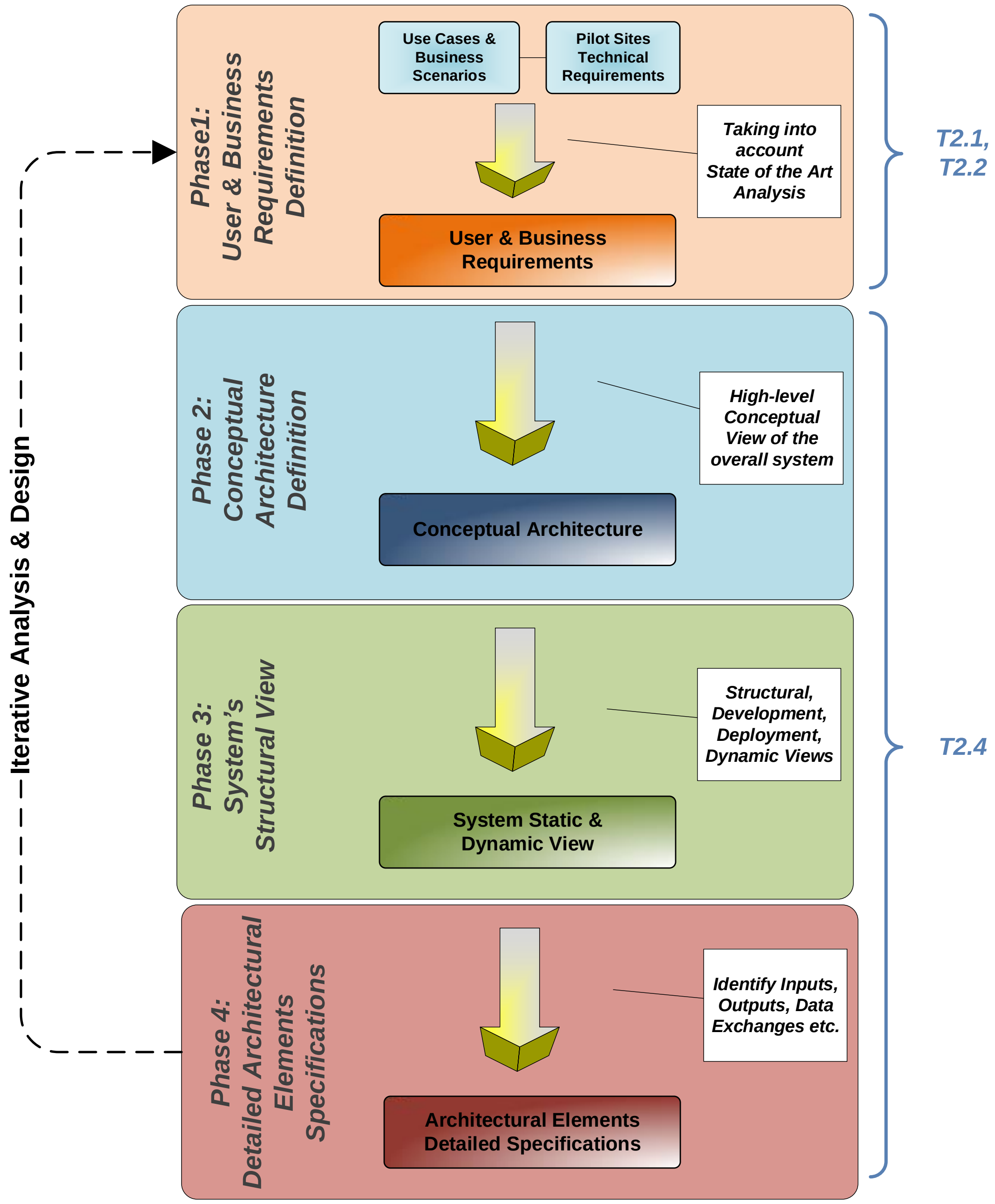


Figure 1Architecture Design Approach and Workflow

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 towards carrying out the required functionality.

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

2.1.1 Design Principles

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

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

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

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 actually operates during its utilization, depending on the various scenarios of use and use cases defined, including the way each component acts within them. Thus, the system's dynamic model and structures define its runtime architectural elements and their interactions due to internal or external stimulus. The internal interactions refer to information flows among architectural elements and their parallel or sequential execution of internal tasks, including the potential expression of the effect they may have on the information.

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

The concept is depicted below:

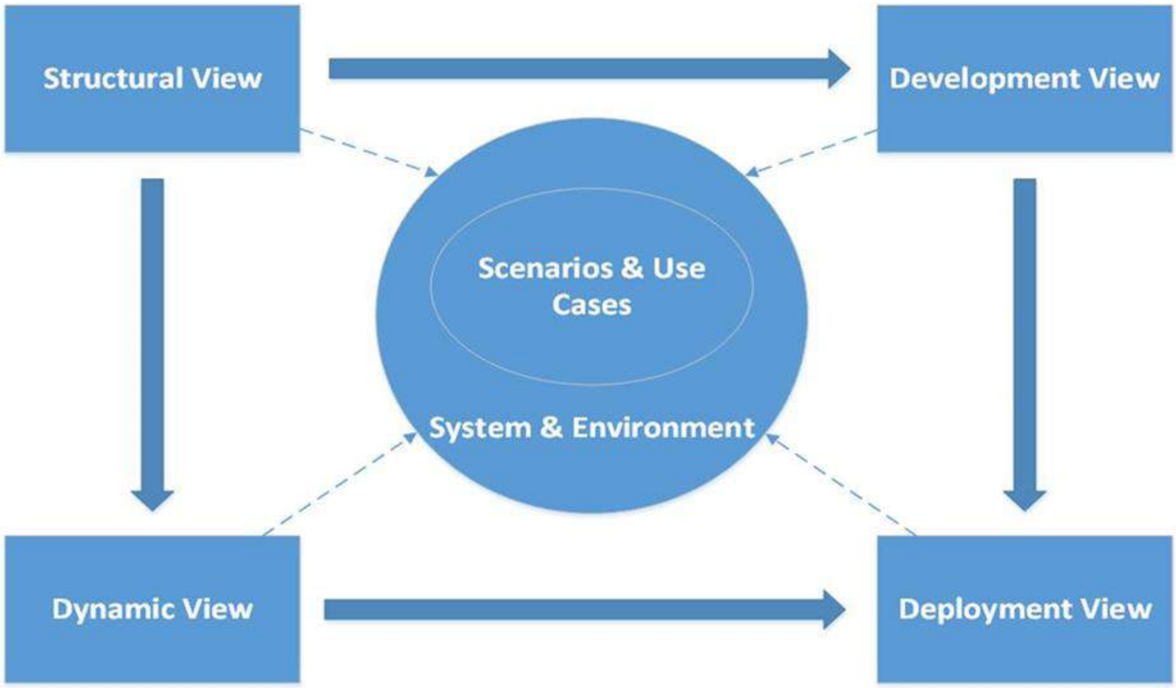


Figure 2 Architectural View 4+1 model

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

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 will be researched and examined by using the principles of Internet of Things (IoT), the concepts of Demand Response programs and the blockchain-driven technology. The following figure presents the conceptual architecture of the eDREAM platform:

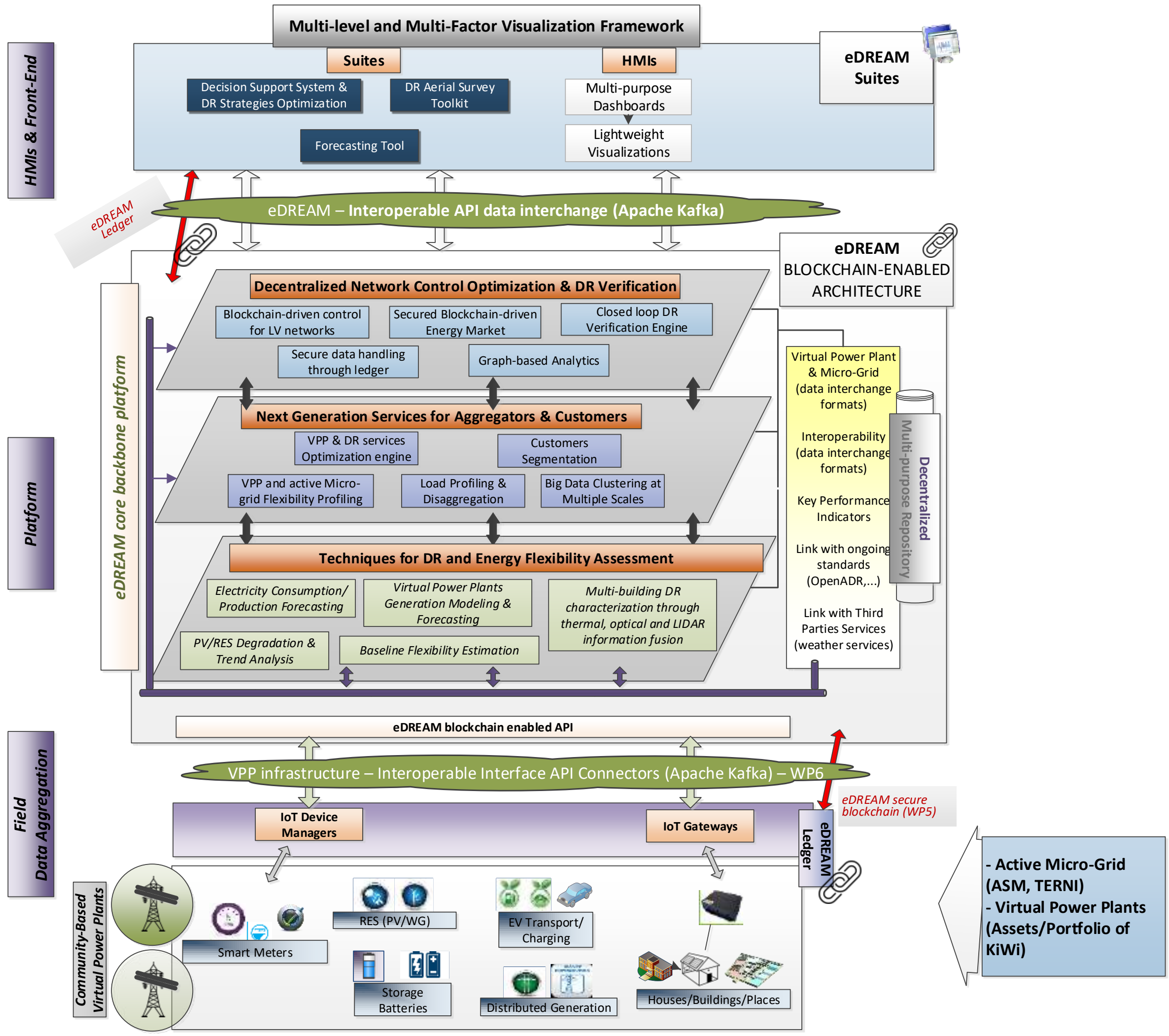


Figure 3 eDREAM Conceptual Architecture

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

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

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

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

- The upper layer **HMIs & Front-end for end-users and operators (WP4 & WP6)** contains accessible and easy-to-use HMIs (e.g. accessible by mobile phone through lightweight visualizations) for end-users and operators that enables vertical collaboration (from the DSO and aggregators to prosumers/consumers) and horizontal collaboration (using virtual topologies, such as the community-based VPPs) within the eDREAM architectural framework. The main purpose of this layer is the visualization of the output data from the Core Backbone Platform which are additionally analyzed and interpreted. Bidirectional data flow is performed between the core platform and the front-end layer, since several decisions of stakeholders are based on the provided results from the components of the core platform.

- The Core Platform is connected with a **Decentralized Multi-purpose Repository (WP5)** which allows data exchanges within eDREAM core framework. This component provides the necessary place for storage and maintenance of data from field devices, data models/profiles for supporting the functions of core components and information from third parties services (e.g. weather services).

As mentioned above, during the **bottom-up process** of the architecture definition all the technology providers 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 4.

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

Component	Related Task	Responsible partner	Contributing partners
Micro-grid monitoring	-	ASM	

EVSEs and EV fleet monitoring	-	EMOT	
IoT Device Manager	T6.2 & T6.3	KIWI & ATOS	TU, CERTH, ENG, EMOT, SVT
IoT Gateway	T6.2 & T6.3	KIWI & ATOS	TU, CERTH, ENG, EMOT, SVT
Electricity Consumption/Production Forecasting	T3.1	TUC	TU, ENG, CERTH
PV/RES Degradation & Trend Analysis	T4.1	TU	ENG, CERTH, SVT
Baseline Flexibility Estimation	T3.2	TU	TUC, E@W, EMOT
Virtual Power Plants Generation Modeling & Forecasting	T3.3	TUC	EMOT, ENG, ASM
Multi-building DR characterization through thermal, optical and LIDAR information fusion	T3.4	TU	CERTH, KIWI
Load Profiling & Disaggregation	T4.2	ATOS	CERTH, E@W, KIWI, ASM
VPP and active Micro-grid Flexibility Profiling	T4.1	TU	ENG, CERTH, SVT, ATOS
Big Data Clustering at Multiple Scales	T4.2	ATOS	CERTH, E@W, KIWI, ASM
Customers Segmentation	T4.2	ATOS	CERTH, E@W, KIWI, ASM
VPP & DR Services Optimization Engine	T4.1	TU	ENG, CERTH, SVT
Secure data handling through 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
Decision Support System & DR Strategies Optimization	T4.1, 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	TU, ENG, CERTH, ATOS

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

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 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 will introduce the defined architectural components with their main functionalities and the dependencies from other components. This process will provide the necessary information for the definition of the detailed modules interfaces and APIs in the next version of the document that is going to be delivered on M18.
The Figure 3 depicts the eDREAM overall structural view with all the identified high-level dependencies.

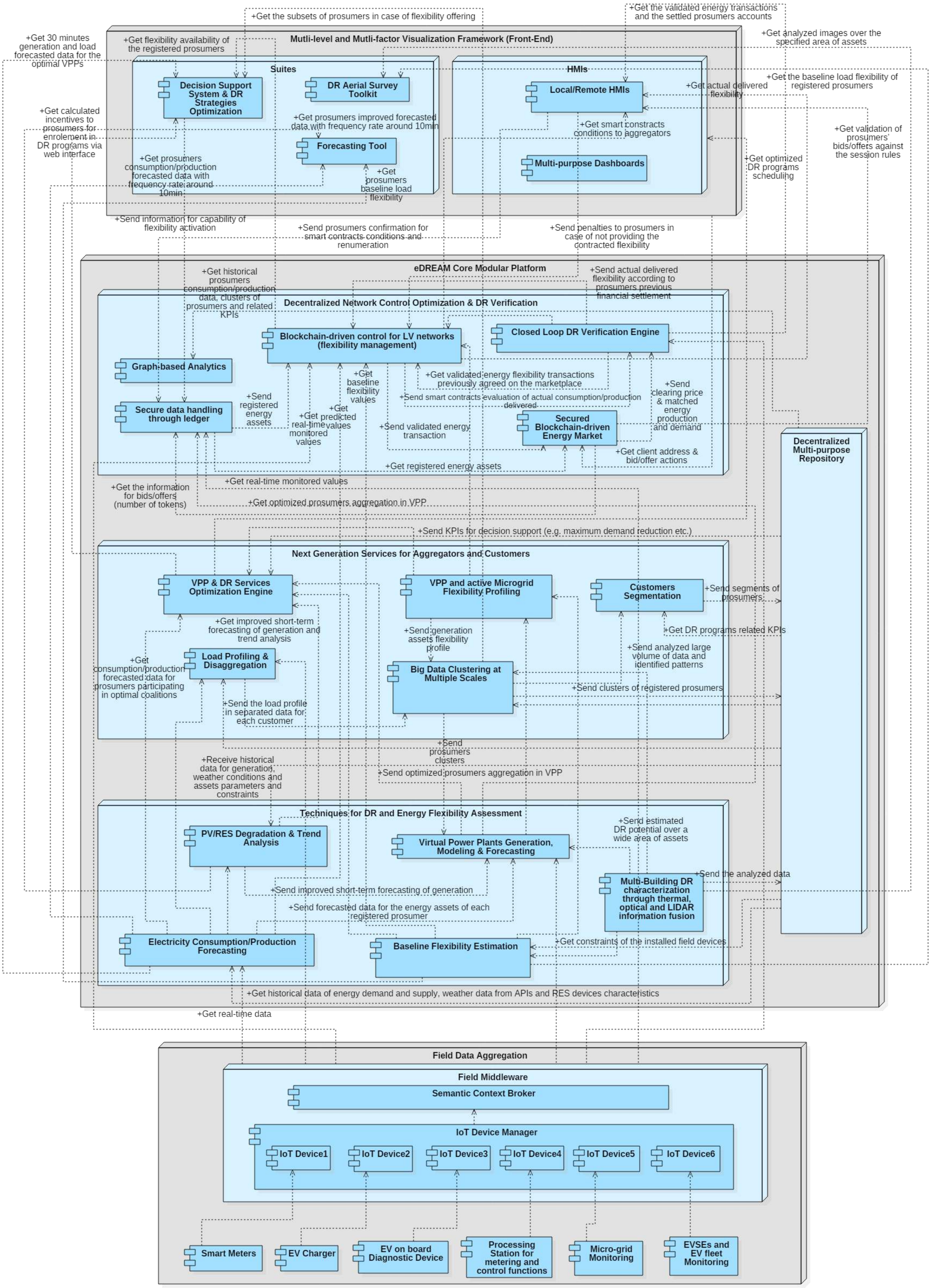


Figure 4 eDREAM overall structural view

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.

Internal Components

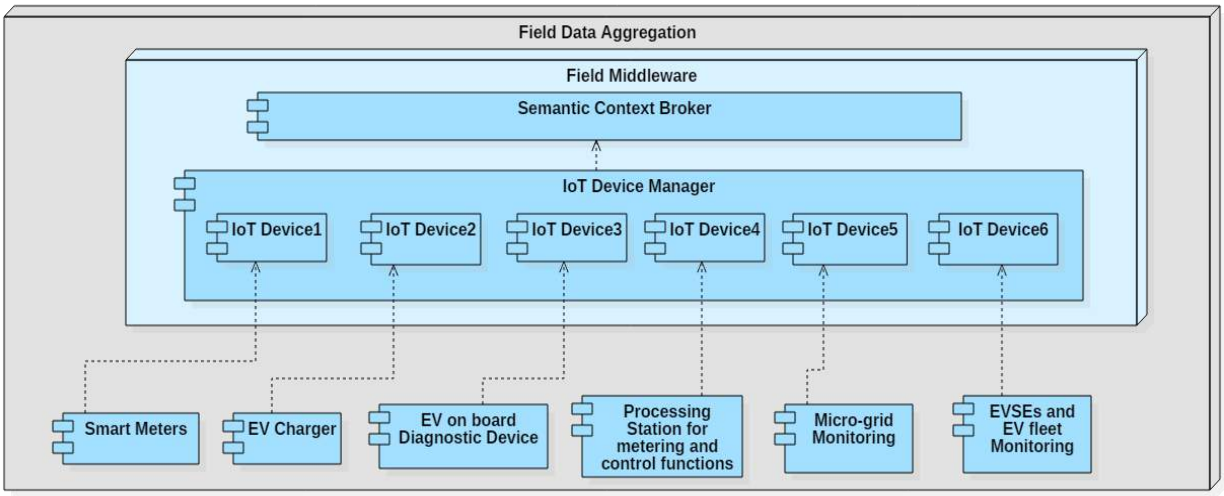


Figure 5 Field Middleware

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 will also describe the nature of the devices itself and the characteristics of the shared information (e.g. type and unit of measured data). The IoT devices are the first level that ensures proper information transmission to the upper layers of the architecture. They ensure data harmonization and seamless information exchange relying on open communication specifications and Machine2Machnine (M2M) communication standards.

4.2.2 IoT Device Manager

Description – Main Functionalities

The **IoT Device Manager** will enable the establishment of a virtual network of heterogeneous physical devices through a common interface and technology-agnostic access to the lower layer of field devices. The virtual representation of the field devices will ensure their seamless integration with the other components of eDREAM platform. The IoT Device Manager facilitates the transition from the deployment of vertically-oriented closed systems towards open systems and platforms that support multiple applications. This component considers the compliance with current and upcoming communication standards and ontologies.

4.2.3 Semantic Context Broker

Description – Main Functionalities

The **Semantic Context Broker** will be possibly implemented using the FIWARE Orion Context Broker[1] that is an open source software for creating different context elements based on the received data and manage them. This component provides the capabilities of producing, gathering, publishing and exploiting context information at large scale. Through this software, the context information is represented through values assigned to attributes that characterize the entities relevant to the received measurements. The Context Broker is able to handle context information at large scale by implementing standard REST APIs. One of the most important features of the Context Broker is that it allows to modeling and getting access to context information irrespectively of the source of this information. This component is based on the concepts of the NGSI model for the management of entities, attributes and context information. The functionality of the APIs designed or selected to connect the CMP with the Field Middleware layer are related to reading inputs (registers, variables and parameters), writing outputs (registers, variables and settings), handling alarms and events and manage security features. The most common APIs and standards related to smart grids and Demand Response programs are the following:

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

[1] https://github.com/telefonicaid/fiware-orion

- **ETSI M2M** (evolved to **oneM2M**) representing Machine-to-Machine communications: It is an application agnostic standard containing an overall end to end M2M functional architecture, identifying the functional entities and the related reference points. It can be used for the exchange of data and events between machines involving communications across networks without requiring human intervention.

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

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

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

Dependencies to other components

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

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). Advanced data analytics methods and tools will be used to enable accurate predictions of energy supply and energy demand based on time series analysis and deep learning techniques. The electricity DR prediction component will rely on the prediction platform implemented in GEYSER project, allowing for powerful data analysis and a lightweight complex event detection module offering online analysis of data streams for real-time detection of events. The prediction platform will provide the following functionalities:

- Enable creation of various prediction models on stream multimodal data, data enrichment, automatic evaluation of series of prediction models for selected source, optimization of various prediction methods;
- Provide prediction of energy consumption and energy demand for selected distributed energy sources.

For electricity production and consumption, GEYSER tools for prediction techniques and tools, based on time series analysis and deep neural networks, will be used and extended to determine future energy consumption trends (aggregated and at each DC component level), frequent patterns etc. Data regarding the historical energy consumption values both at distributed energy resources and smart grid levels will be modelled by means of time series and then processed using neural networks specific methods, for translating the past values into estimates of the future. After that the constructed models will be used to extract information regarding the energy consumption short and long time trends taking into account other related patterns (e.g. seasonal etc.). The big data sets for DERs monitored data and constant sample rates for smart energy meters will be dispatched to both a Batch layer (Apache Spark for managing data sets) and a speed layer (Apache Storm managing real stream of data). This will provide balanced latency, throughput, and fault-tolerance as result of using at the same time batch processing to generate comprehensive and accurate views of data sets, and real-time stream processing to provide views of online data. The collected data will be stored as time series in highly scalable, high-performance distributed NoSQL database (e.g. Cassandra) designed to handle large amounts of data. This innovative approach of using deep learning models on top of big data analysis tools have several advantages: better training speed and smaller error rate for hyper parameters for neural network training and the application of deep neural networks onto big data. Thus, this component will provide the aggregators insights for day-ahead and long term DR planning. In addition, it can provide valuable information to DSOs and system operators for short-term prediction of grid's congestion points and long-term spatial grid design. Figure 5 shows the high level architectural dependencies of this component indicating the connections and the information exchange with other components.

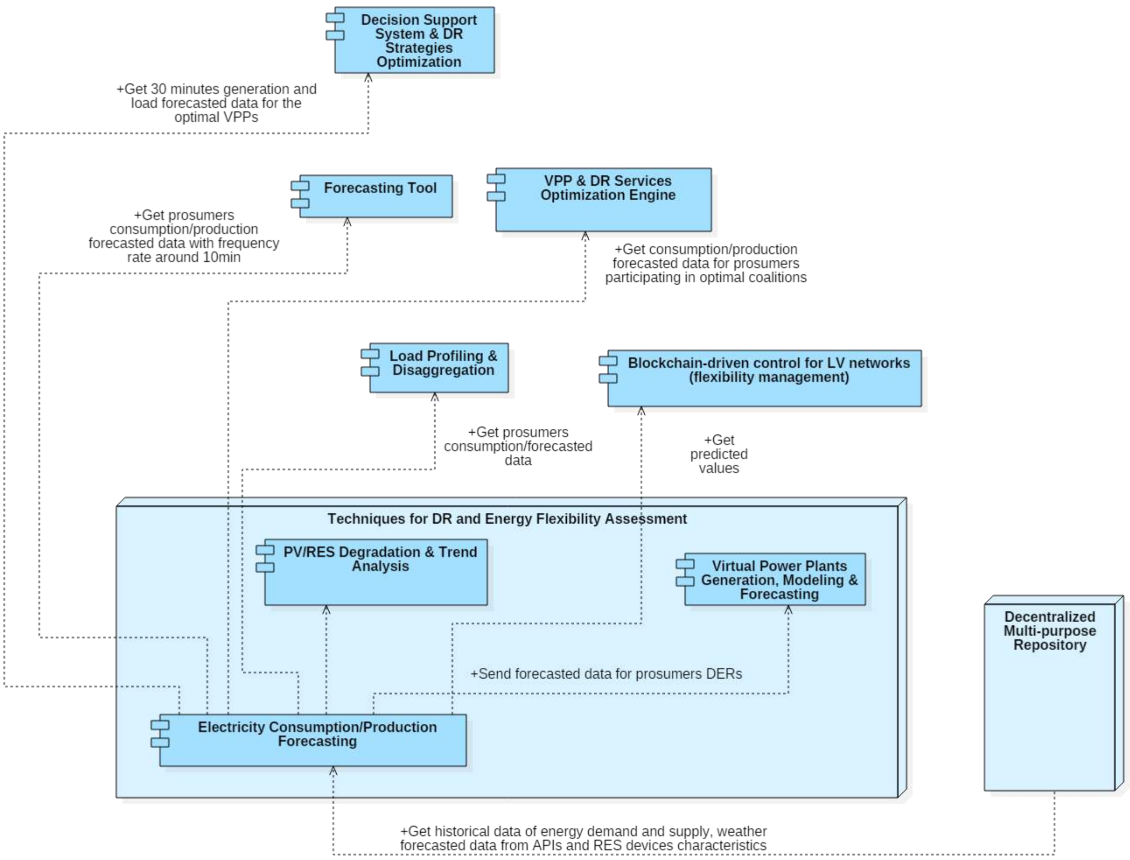


Figure 6 Electricity Consumption/Production Forecasting Component

Dependencies to other components

- The component retrieves the necessary input data in order to perform its calculations from the **Decentralized Multi-purpose Repository**. This data includes historical data of prosumer energy demand and supply, weather forecasted data from respective weather APIs and parameters and constraints for RES devices.
- The component **Virtual Power Plants Generation, Modeling & Forecasting** receives forecasted data for the production of prosumers DERs in order to identify the optimal coalitions.
- The **PV/RES Degradation & Trend Analysis** receives the produced forecasted data and calculates improved short-term forecasting of generation.
- The component **Blockchain-driven control for LV networks (flexibility management)** receives the predicted values of consumption/production for registered prosumers in order to identify the flexibility that can be delivered in case of a flexibility request.
- **Load Profiling & Disaggregation** receives the forecasted data for prosumer consumption in order to perform accurate calculations and produce the prosumer load profile and the disaggregation of consumption to different devices.
- The **Forecasting Tool** that belongs to the front-end part and is connected with the Aggregator’s UI receives receives prosumers consumption/production forecasted data with a standard frequency rate around 10min (the frequency rate can be also 5min or 1 hour).
- The **VPP & DR Services Optimization Engine** receives forecasted data in order to identify the loads to be shed and to identify the optimal set points of dispatchable generators in case that a VPP provides ancillary services (e.g. Frequency Services etc.) to the energy market or participates in different energy markets (e.g. Wholesale Market via intra-day trading services etc.).
- The **Decision Support System & DR Strategies Optimization** belongs to the front-end part and exchanges basic information with the back-end component VPP & DR Services Optimization Engine for the calculations related to the participation of VPPs in the different energy markets and the related services.

4.3.2 PV/RES Degradation & Trend Analysis

Description – Main Functionalities

The functionality of this component it 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. In addition, it provides improved short-term forecasting of generation based on near real-time trend analysis algorithm. 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. The high-level dependencies of this component are depicted in Figure 6.

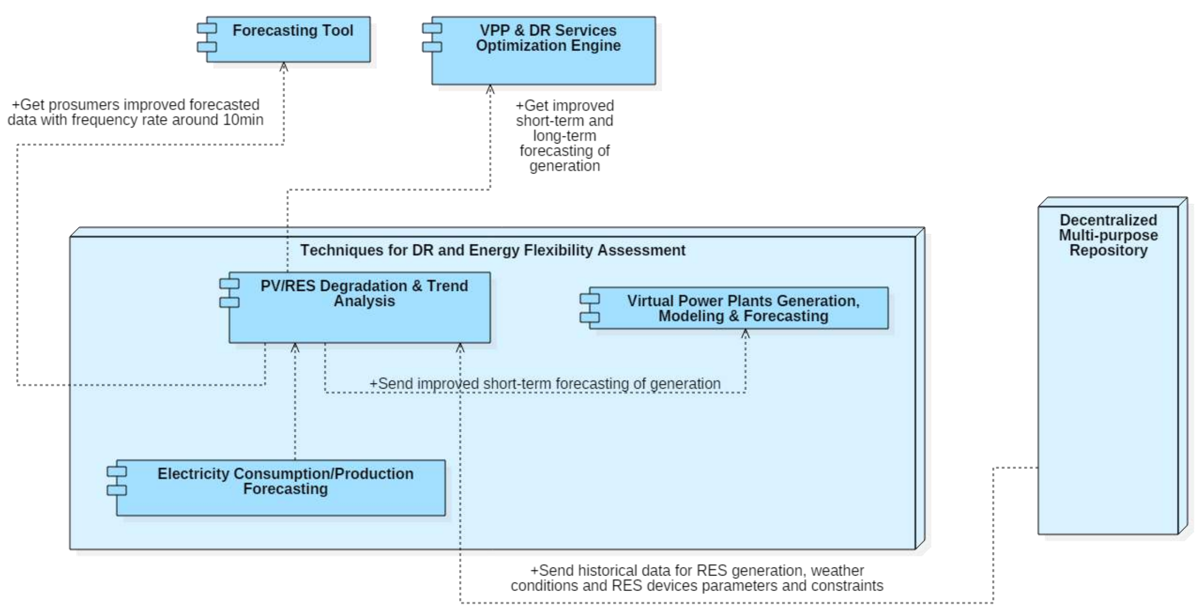


Figure 7 PV/RES Degradation & Trend Analysis component

Dependencies to other components

- Receive the output of **Electricity Consumption/Production Forecasting** and produce improved forecasted data.
- Towards performing the necessary calculations, it receives historical data for RES generation, weather conditions and RES devices parameters and constraints from the **Decentralized Multi-purpose Repository**.
- This component sends improved short-term and long-term forecasting of generation to the component **Virtual Power Plants Generation Modeling & Forecasting** in order to create the optimal coalitions of generation assets.
- The front-end component **Forecasting Tool** receives the improved forecasted data concerning prosumers generation with frequency rate around 10min.
- This component sends improved forecasting of generation to the component **VPP & DR Services Optimization Engine** towards improving the short-term and long-term DR program planning and scheduling.

4.3.3 Baseline Flexibility Estimation

Description – Main Functionalities

This component is responsible for calculating the flexibility of prosumer baseline load curve taking into account smart metering data/energy demand profiles for the participation in various DR programs (resource adequacy/capacity, economic/energy tariff, balancing/ancillary services etc.). Comparison of baseline load calculation methods for participation in various DR programs will be performed for all eDREAM demonstration/pilot sites subject to their local weather conditions (historic monitoring records or typical meteorological year). The examined calculation methods will be based on moment of dispatch, average and weighted average over X of Y most recent like-days with the highest load preceding an event (e.g. highest 5 of 10 previous admissible days). Consolidated Edison of New York and the NYISO (New York Independent System Operator) Emergency Demand Response Program Manual (12/02/2010 version 6.2 Section 5.2) will be taken into account together with the currently employed methodologies by the EU member states to calculate a Customer Baseline Load (“CBL”) for Customers/Aggregators enrolled in DR programs.

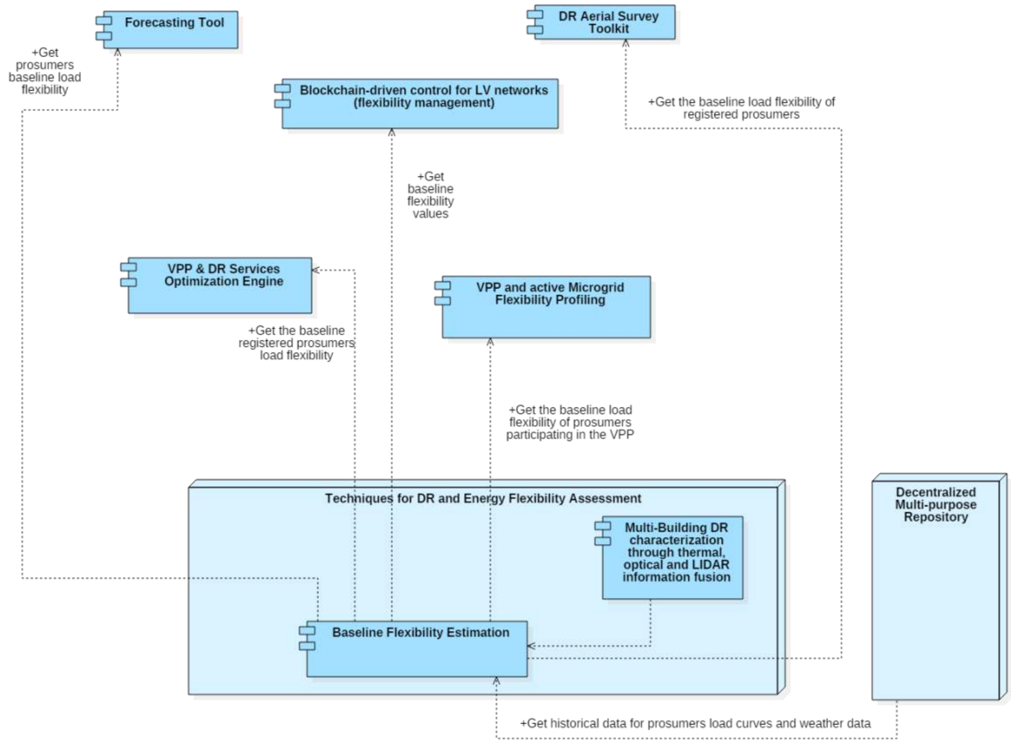


Figure 8 Baseline Flexibility Estimation component

Dependencies to other components

- The component receives historical data for registered prosumers load curves and weather data from **Decentralized Multi-purpose Repository** in order to perform the necessary calculations.
- It can also receive analysed images and data from the component **Multi-building DR characterization through thermal, optical and LIDAR information fusion** in order to evaluate the baseline load flexibility of new potential prosumers.
- The component **VPP and active Microgrid Flexibility profiling** can receive the baseline load flexibility of prosumers participating in the VPP in order to calculate the VPP flexibility profiling.
- The **VPP & DR Strategies Optimization Engine** receives the baseline load flexibility of registered prosumers that is going to be used in the planning of DR programs.
- The component **Blockchain-driven control for LV networks (flexibility management)** can receive the baseline flexibility values of registered prosumers in order to evaluate the actual flexibility that can be delivered in case of a flexibility order sent by the DSO.
- The front-end component **Forecasting Tool** receives the prosumers baseline load flexibility, so as to make a comparison with the displayed forecasted consumption.
- The **DR Aerial Survey Toolkit** that belongs to the front-end part can receive the baseline load flexibility of registered prosumers in case of evaluation of their flexibility after a long time period with aerial survey. It can also receive the estimated baseline load flexibility of new potential prosumers that results from calculations based on analyzed collected images.

4.3.4 Virtual Power Plants Generation Modeling & Forecasting

Description – Main Functionalities

The responsibility of this component is to develop models for distributed generation of electricity of multiple types (i.e. wind-turbines, small hydro, photovoltaics, back-up generators, etc.) in the scope of creating optimal coalitions (Virtual Power Plants) which will provide a more reliable aggregated power supply. The rationale behind creating such coalitions is that a mix of different energy generation resources which have different energy generation models and scale may be interested to cooperate in a convenient way, with a view to achieve pre-defined smart grid sustainability objectives. At the same time, this will increase the participation in DR programs of small renewable energy producers existing studies showing that this could be supported by any household which has as little as 1 kW capacity generation. The local co-generation side-effects will be studied and modelled with the goal of determining the potential surplus/deficit of locally generated electrical energy, generation peaks and valleys compensations, and energy production aggregation potential. The ultimate aim is to provide an innovative fully distributed controlled electricity generation virtual aggregation techniques based on nature inspired heuristics and taking advantage on the features provided by the blockchain-based distributed computing platforms. In this approach the energy generation resources may act as independent agents which participate and cooperate to provide reliable aggregated power supply and improved efficiency of smart energy grids.

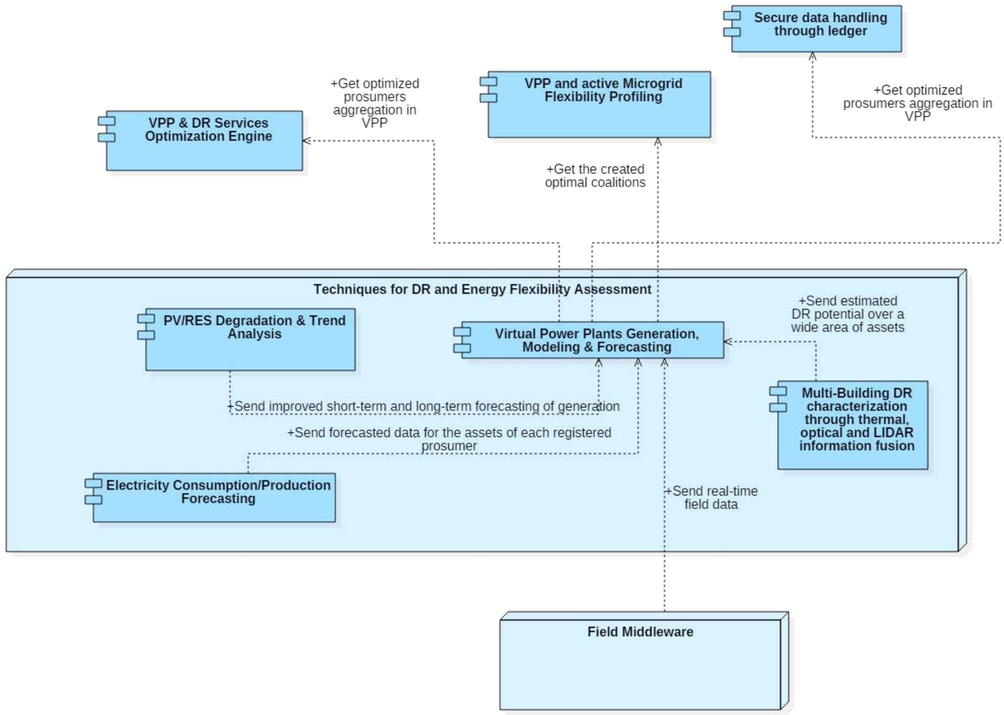


Figure 9 Virtual Power Plants Generation Modeling & Forecasting component

Dependencies to other components

- The component receives the forecasted consumption/production data for the assets for each registered prosumer in the portfolio from the component **Electricity Consumption/Production Forecasting**.
- The **PV/RES Degradation & Trend Analysis** sends to the component improved short-term and long-term forecasting of generation.
- It receives processed real-time data from the **Field Middleware**.
- It receives analysed data for DR potential over a specified area of assets from the component **Multi-building DR characterization through thermal, optical and LIDAR information fusion**.

- The component sends the created optimal coalitions to the **VPP and active Microgrid Flexibility Profiling**, in order to calculate the flexibility that can be delivered from the created coalitions.
- The **VPP & DR Services Optimization Engine** receives the optimal coalitions, so as to define their participation in ancillary energy market services and to identify the loads that can be shed and the dispatchable generators.
- The component send the information about the created optimal coalitions to the **Secure data handling through ledger** for secure storage.

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 will be equipped with HD optical, thermal imaging and LIDAR scanners. The drone(s) will be programmed to fly a fixed path over a specified asset area at different times in order to:

- Identify peak and minimum energy demand (aggregated and individual) requirements;
- Provide information about the presence of energy intensive plant items of the building (e.g. HVAC, CHP etc.) and the heat-loss parameters;
- Provide information about the orientation and shading of the building;
- Detect data for RES generation.

The collected digital images will be live streamed into a mobile workstation for further data analyses. Data analyses will derive estimates of assets' energy demand (both aggregated and individual extremes) based on the recorded heat dissipation maps/thermograms, using a specific software. 3D model(s) of the surveyed area will be also created. The created model will be used to simulate assets' energy performance in comparison to the results obtained from thermal image analyses. Individual assets with the highest demand response potential (highest absolute difference between peak and minimum energy demand/losses) will be identified in addition to the estimates of the overall DR potential of the surveyed area. The estimates based on a particular survey time period will be extrapolated over the whole year, taking into account potential measurement uncertainties and variability in environmental conditions from historic typical meteorological year (tmy) data files and/or weather monitoring records. The identified estimates will be compared to the current industry standard procedures based on detailed technical data records of DR assets, as well as to the actual smart meter monitoring data provided by KiWi power. As a final result, the development process of this component will provide accuracy comparison and recommendations between various aerial survey techniques (optical, thermal imagining, LIDAR) and modelling methods (photogrammetry, thermography, point cloud), subject to flight plan parameters (e.g. pass density, flight time, height, speed, resolution, survey area, view angle).

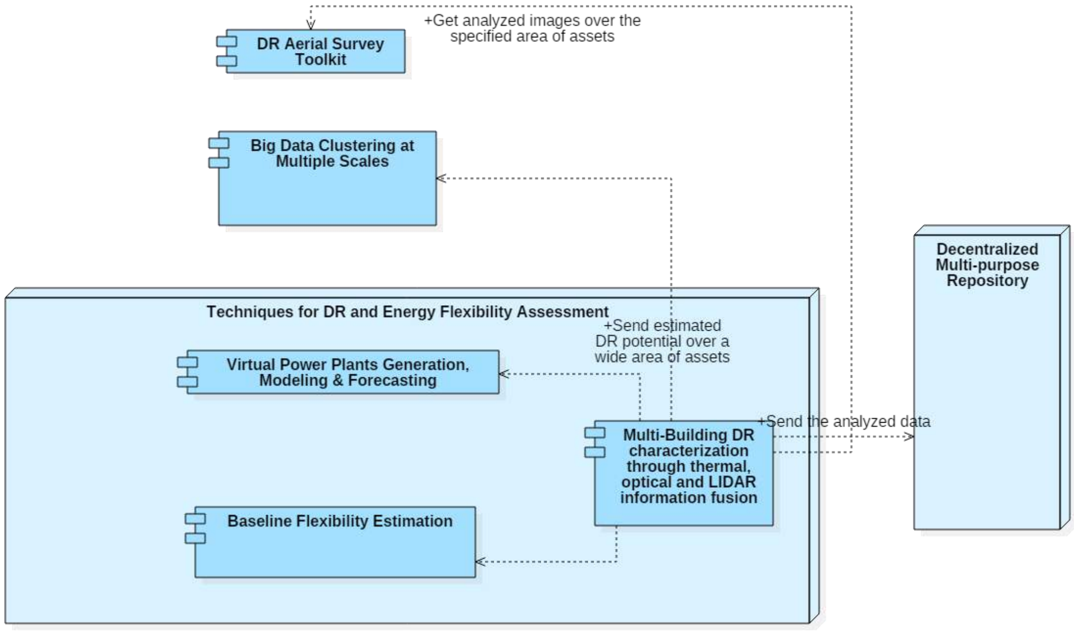


Figure 10 Multi-building DR characterization through thermal, optical and LIDAR information fusion component

Dependencies to other components

- The connections with the components **Virtual Power Plants Generation, Modeling & Forecasting** and **Baseline Flexibility Estimation** were explained in the above respective sections.
- The component sends analysed data for the estimated DR potential over a specific area of assets to the component **Big Data Clustering at Multiple Scales** in order to identify patterns on the data of new potential customers and create additional clusters if needed.
- The component sends the output data for secure storage in the **Decentralized Multi-purpose Repository**.
- The front-end component **DR Aerial Survey Tookit** receives the analysed images over the specified area of assets in order to inform aggregator for the DR potential.

4.4 Next generation DR Services for Aggregators and Customers

4.4.1 Load Profiling & Disaggregation

Description – Main Functionalities

This component is responsible for detecting load profile patterns and extracting prosumer load profile based on historical data about load consumption. It will also provide the possibility of near real time load profiling using real time data from field devices. The identified profile patterns will provide necessary information for the grouping of customers, so as to facilitate the selection of the suitable DR program to be applied (e.g. price-based programs, incentives, ToU etc.). An additional feature is that it will calculate an estimation of load disaggregation, so as to give information for the load consumption of the prosumer different appliances.

Dependencies to other components

- The component will receive historical data for the prosumer load consumption curves and related KPIs from the **Decentralized Multi-purpose Repository**.
- It will also get real time data from the **Field Middleware** layer in order to perform near real time load profiling.
- The component **Electricity Consumption/Production Forecasting** can send forecasted data for prosumer consumption to the component, so as to improve the calculated load profiles.
- The component sends the calculated load profiles to the component **Big Data Clustering at Multiple Scales** that will use the profiles to create groups – clusters of customers.

4.4.2 Big Data Clustering at Multiple Scales

Description – Main Functionalities

The functionality of this component is to provide a number of well defined and separate clusters of prosumers. The number and the mutual distance of clusters will respect the threshold derived from proper KPIs (technology, generation curve, time response, frequency, voltage, reactive capacity modulation etc.) to be defined. This will be a scalable procedure for extracting a number of clusters from a large amount of users load curves in order to be able to apply this procedure to heterogenous data sets with a variable number of load curves. This means that the same technique can be applied with different constraints and starting conditions.

Dependencies to other components

- The component will receive the prosumers load profiles from the component **Load Profiling & Disaggregation**.
- The **VPP and active Microgrid Flexibility Profiling** can send the flexibility margins of the prosumers generation assets.
- The component will receive related KPIs from the **Decentralized Repository** and it will send the created clusters for secure storage after the calculation.
- The **Multi-building DR characterization through thermal, optical and LIDAR information fusion** can send the analysed images to the component in order to detect similarities between the existing clusters and the potential prosumers estimated profile.

- The component can send clusters of prosumer to the **Virtual Power Plants Generation, Modeling and Forecasting**, so as to create the optimal coalitions.
- The **Customers Segmentation** receives the created clusters, so as to be able to assign prosumers to specific clusters.

4.4.3 Customers Segmentation

Description – Main Functionalities

This component is responsible for the assignment of the customers to a particular customer group by recognizing the customer's load profile pattern. Segmentation of prosumers will be also useful for catagorizing the participation of small and medium generation to different energy markets (ancillary services, balance market etc.).

Dependencies to other components

- The component receives the created clusters of prosumers from the component **Big Data Clustering at Multiple Scales**.
- The component gets related KPIs from the **Decentralized Repository** and after the calculation, it sends the customers segments for secure storage.

4.4.4 VPP and active Microgrid Flexibility Profiling

Description – Main Functionalities

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

Dependencies to other components

- The component receives the optimal coalitions from the component **Virtual Power Plants Generation, Modeling & Forecasting**.
- It also receives the prosumers baseline load calculation from the component **Baseline Flexibility Estimation**.
- The **Big Data Clustering at Multiple Scales** can consider as input the generation assets flexibility profile deriving from the component.
- The component provides its output to the **VPP & DR Services Optimization Engine**, so as to be considered as input parameter during the optimal DR scheduling.
- It also provides input to the component **Blockchain-driven control for LV networks (flexibility management)**.

4.4.5 VPP & DR Services Optimization Engine

Description – Main Functionalities

This component will deliver the functionalities of advanced DR optimization mechanisms that will meet the expectations of all stakeholders (aggregators and services offered in their customers as well as in DSOs), in particular dealing with situations where events are unscheduled and there are a lot of uncertainties in respect to the effective multi-factor load dispatch. Going beyond traditional reduction of peak loads in buildings, a multi-factor decision support system will be implemented that is able to take into account the participation of flexible resources (e.g. active assets in micro-grids fully or partially connected to the grid, RES generation, other DERs) towards optimizing from one side the energy plans (Aggregators/DSOs) and on the other side to maximize expectations for the consumers/prosumers (mainly coupling economic, comfort and environmental indicators). The engine will be able to consider during its optimization process the consumption and production forecasting, will improve forecasting with trend analysis algorithms (e.g. slope statistic profiles), the average baseline calculated in a contextual manner (e.g. for a given district area, a virtual power plant, micro-grid node and in combination of the latter), as well as potential incentives for the final users (in terms of automated rewards that ensure lower prices for the customers given their existing pricing contracts). The toolkit will provide a set of feasible solutions for the stakeholders (e.g. Aggregators), along with the proposed scheduling of the demand response strategies across multiple portfolio (supporting a range of portfolio assets including virtual power plants). The decision support system will have a number of KPIs that will take into account for selecting automatically the optimal scheduling such as smooth reduction, maximum demand reduction, maximum economic benefits, as well as other KPIs that will come in collaboration with the end-users (KIWI, ASM). The optimal scheduling approach will be based upon graph partition techniques (i.e. extended Branch and Bound, minimum and random spanning trees). In the case of VPP, loads and generators profiles together with flexibility assessment and consumption/generation forecast are used to obtain optimal set points for generators and load curtailment according to the RES drop Energy Community scenario. It will provide services related to requests/offers matching system that will be able to interact with prosumers' smart contracts and aggregate flexibility service offers. In addition, the toolkit will be enhanced with the integration of multi-objective optimization and interactive visualization mechanisms.

Dependencies to other components

- The component receives forecasted data from the component **Electricity Consumption/Production Forecasting.**
- The component will improve the forecasted data with the inputs received from **PV Degradation & Trend Analysis** and **Baseline Flexibility Estimation**.
- It will receive DR related KPIs from the **Decentralized Repository**, in order to consider them during decision support process.
- It receives the optimal coalitions from the component **Virtual Power Plants Generation, Modeling & Forecasting**.
- It also receives the flexibility profile of VPPs from the **VPP and active Microgrid Flexibility Profiling**.
- The component will send the calculated results to the front-end components **Decision Support System & DR Strategies Optimization** and to other related **HMIs** in order to inform properly the system stakeholders.

4.5 Blockchain-enabled Decentralized Network Control Optimization and DR Verification

4.5.1 Secure data handling through ledger

Description – Main Functionalities

The responsibility of this component, named also as blockchain distributed ledger, is to store energy transactions and DR flexibility services at a microgrid level in a secure and trustful manner. State of the art approaches in storing, sharing and replicating data in a secure and reliable manner cross the nodes of a distributed system will be investigated. Cryptographically authenticated data structure like Merkle-Patricia trees will be used to store the data feed by the smart energy metering devices using blockchains. Control and permission mechanisms will be implemented for verifying the participation of new energy prosumption resources to the distributed network for DR programs management with view of creating permissioned implementation of blockchain platform supporting data privacy. The ledger should be characterized by scalability and the usage of blockchain based distributed data bases such as BigchainDB will be assessed. 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 detection and 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:

- ➢ Detection of grid level congestion points;
- ➢ Flexibility requests to aggregators;
- ➢ Selection of flexibility offers from aggregators;
- ➢ Track and control the flexibility delivery of 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 of prosumers.

Dependencies to other components

- The component receives the registered energy assets from the **Secure data handling through ledger**.
- It receives real time data, forecasted values and baseline load flexibility from the **Field Middleware**, **Electricity Consumption/Production Forecasting** and **Baseline Flexibility Estimation** in order to perform the necessary calculations.
- It sends information for the smart contracts evaluation of actual consumption/production delivered to the component **Closed loop DR Verification Engine** and it receives in turn the validated energy transactions.
- The component sends to the **Decision Support System & DR Strategies Optimization** the flexibility availability of the registered prosumers.

4.5.3 Secured 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 will be able to offer their services by reacting to changes in the price of energy compared to the reference value, thanks to the trading marketplace that will be created using the smart-contracts. The producers/prosumers trade energy in a peer to peer fashion either directly or through an energy aggregator if they are not big enough. The 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.

Dependencies to other components

- The component receives the registered energy assets from the **Secure data handling through ledger**.
- It also receives the bid/offer actions and the client address from the **HMIs.**
- It sends the clearing pricing and the matched energy production and demand from the component **Closed loop DR Verification Engine**.

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

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

Dependencies to other components

- The component sends the validated energy flexibility transactions previously agreed on the market place to the **Blockchain-driven control for LV networks**.
- It send the validated energy transactions and the settled prosumers accounts to the **HMIs**.

4.5.5 Graph-based Analytics

Description – Main Functionalities

This component offers graphical representation of analysed output data from clustering and segmentation techniques and DR strategies/programs. It provides visualizations related to all the available time series coming either from energy (e.g. transactions) or finance (e.g. settlements). This tool supports the analysis of large volumes of prosumer energy consumption/production data. It allows the detection of spatiotemporal patterns that are important for DR strategies optimization. More specifically, the tool allows the user to compare high-level spatio-temporal signals (e.g. segmentation, clustering) and low-level activity (e.g. DR strategies/programs), correlate them with impact to grid stability and link them to business and comfort aspects, in order to validate and possibly initiate changes in the DR planning. The tool allows the analyst to process data in different temporal resolutions and with different temporal filters, assisting in detection of patterns that may be difficult to detect otherwise. Furthermore, the component supports design and DR programs management decisions by visualizing the results of simulations on DR programs application under different conditions.

<u>Main Functionalities</u>

- Obtain and pre-process large volume of historical data.
- Have access to real time and near real time from smart meters and monitoring devices.
- Have access to KPIs from Decentralized Repository.
- Receive data from clustering and segmentation components.

Dependencies to other components

The component can interact with other components from the core platform, the Decentralized Repository and with components of the front-end part.

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 will be returned to the core platform as input values to the Decision Making & Optimization Mechanisms, leading to an iterative process with the main scope to conclude to the optimal solution for the eDREAM system.

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

- HMIs
  - Remote HMIs via web applications
  - Multi-purpose Dashboards
- Suites
  - Decision Support System & DR Strategies Optimization
  - DR Aerial Survey Toolkit
  - Forecasting Tool

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 will be provided from the operator in order to set the constraints for the analysis needed. The role of these components is the decisions taking from stakeholders or the end users will directly affect the operation of the network. The main functionalities of the HMIs are:

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

More specifically, the component of the HMIs are:

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

Remote HMIs:
Remote HMIs (such as mobile or web applications) concern an easy to use multi-level and multi-factor visualization framework for aggregators as well as for prosumers (mainly PV and wind power owners) towards improving their portfolio management. An enhanced graph analytics platform will be designed targeting the challenges of Demand Response sector, by introducing novel techniques for initial graph formulation and applying innovative visual clustering (e.g. k-partite techniques to segment applied DR strategies at different DERs and their correlation with other critical KPIs such as profits/loses, congestion improvement achieved, etc.) and multi-objective analysis, so as to analyse in the temporal domain patterns multiple aspects at once (e.g. generate distance graphs among different virtual power plants taking into account applied DR strategies, incentives used and finding optimal trade-offs among multiple KPIs) providing relevant stakeholders a more complete insight for long-term improvement of the applied DR strategies. The visualisation framework will support the latest generation augmented reality (AR) tools (e.g. Microsoft Hololens). The HMIs will also be enhanced with the following functionality in order to include more parameters in the optimal DR planning:

- **Multi-Objective Optimization:** A hypothesis/objective formulation framework will be incorporated to the visualization component, allowing the operators (e.g. DSO/Aggregator and prosumers) to dynamically formulate and validate DR strategies in sub-sets of available historical data. The component will use the algorithms the forecasting algorithms for consumption/production in order to analyse the hypothesis cases formulated by the end-users through the Demand Response Graph Analytics toolkit (DR-GAT). Integration with the external generic optimisation tools (e.g. GenOpt, Dakota) will provide access to the advanced optimisation algorithms which will be applied to achieve optimal DR performance. Application of tailored multi-objective optimisation algorithms will be tested vs. surrogate-based minimisation. The application will be able to recommend the customers, based on their location, additional services they can experience for further smartening/revenue generation of their energy assets and participating DR programs.

HMIs constitute the interface between the core platform components and the eDREAM stakeholders. HMIs can also be dependent to other components of this 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 maximize the benefits of the applied DR strategies.

- **DR Aerial Survey Toolkit:** This tool communicates with the Multi-building DR characterization component, so as to receive the analysed data from the collected images and provide insights to aggregators for the new potential customers capability.
- **Forecasting Tool:** This tool receives signals and information from the Electricity Consumption/Production component, so as to provide aggregators with accurate forecasted data with a standard pre-defined frequency rate.

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 analyzed in deliverable "D2.2 Use Case Analysis and application scenarios description V1". In the context of the WP2 activities, technical teleconferences on use cases/functional analysis were carried out in the scope of identifying all the dependencies between the key architectural components and the data exchanged during the system's functions or procedures. The logic of these complex operations are presented through **Sequence Diagrams** defining the functionalities of each of the key architectural components and the execution flows within each use case.

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

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

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

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

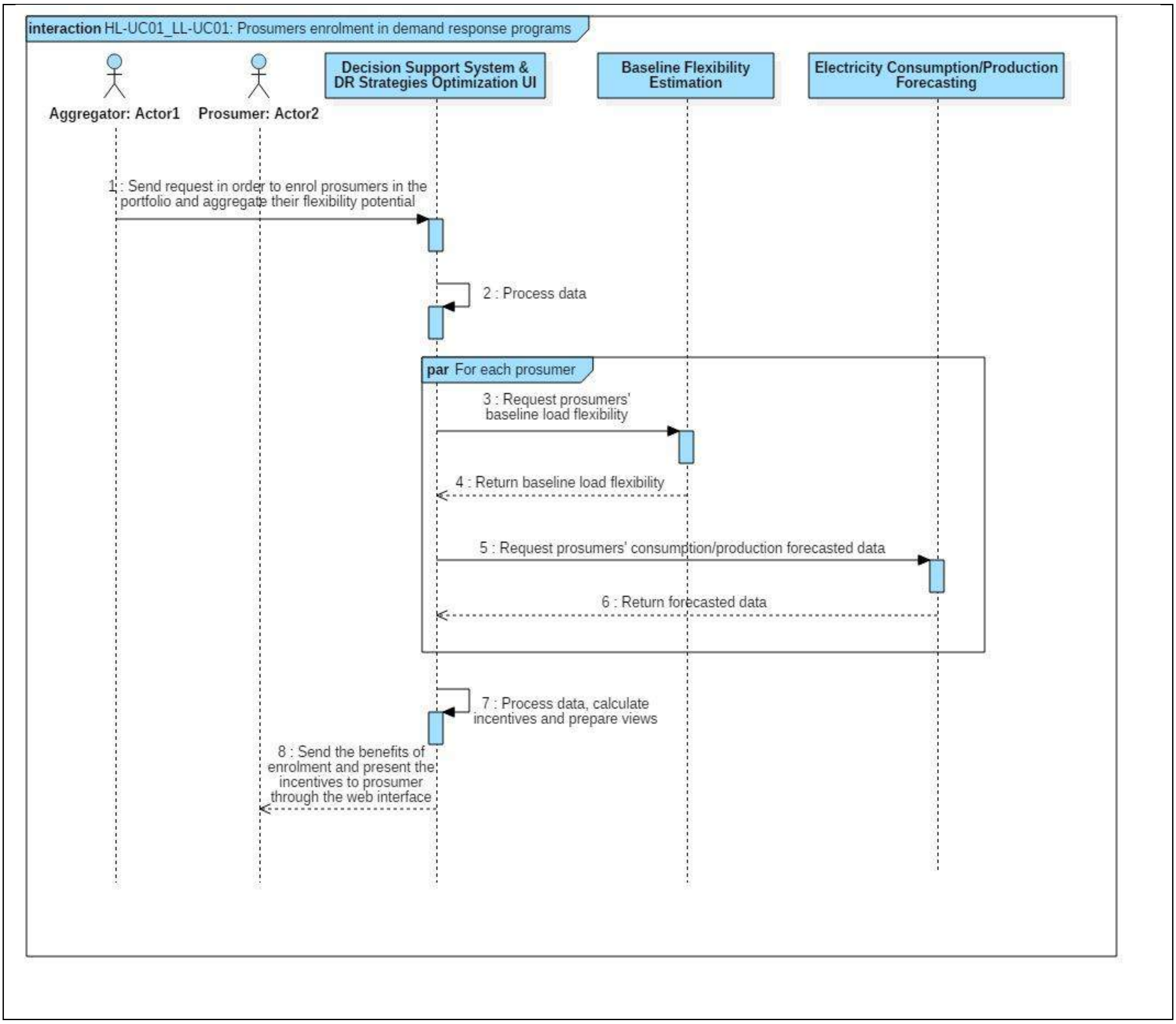
5.1.1 HL-UC01_LL-UC01: Prosumers enrolment in demand response programs

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

Generic Description

Use Case Name	**HL-UC01_LL-UC01: Prosumers enrolment in demand response programs**
Version	V0.4
Authors	E@W, TUC
Last Update	1st Version with D2.2 Update within December 2018
Brief Description	Aggregator negotiate with its 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 its prosumers about the benefits of enrolment in demand response programs. Through interactive visualization interface, potential incentives are sent to prosumers. These incentives are formulated mainly taking into account the "Customer Baseline Load (CBL)" through the component of Baseline Flexibility Estimation and forecasted data about consumption and production.
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	Aggregators and prosumers are enrolled.
Involved Actors	Aggregator, Prosumer
Use Case 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 baseline load flexibility of prosumers. 2. The Decision Support System & DR Strategies Optimization UI receives the request and process the settings/preferences. 3. The Decision Support System & DR Strategies Optimization UI component requests the baseline load flexibility for each prosumer from the component Baseline Load Flexibility.

	4. The Baseline Load Flexibility returns the requested data. 5. The Decision Support System & DR Strategies Optimization UI requests forecasted data or consumption/production for each prosumer from the component Electricity Consumption/Production Forecasting. 6. The Electricity Consumption/Production Forecasting returns the requested data. 7. The Decision Support System & DR Strategies Optimization UI process the received data, calculate incentives and prepare views. 8. The aggregator though the web interface of Decision Support System & DR Strategies Optimization send incentives to prosumer.
Alternative Cources	-
Relationships with other Use Cases	
Architectural Elements / Services Involved	Decision Support System & DR Strategies Optimization UI; 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
Addressed requirements of the system	
Notes (Optional)	-
UML Sequence Diagram	



5.1.2 HL-UC01_LL-UC02: Contract Setting

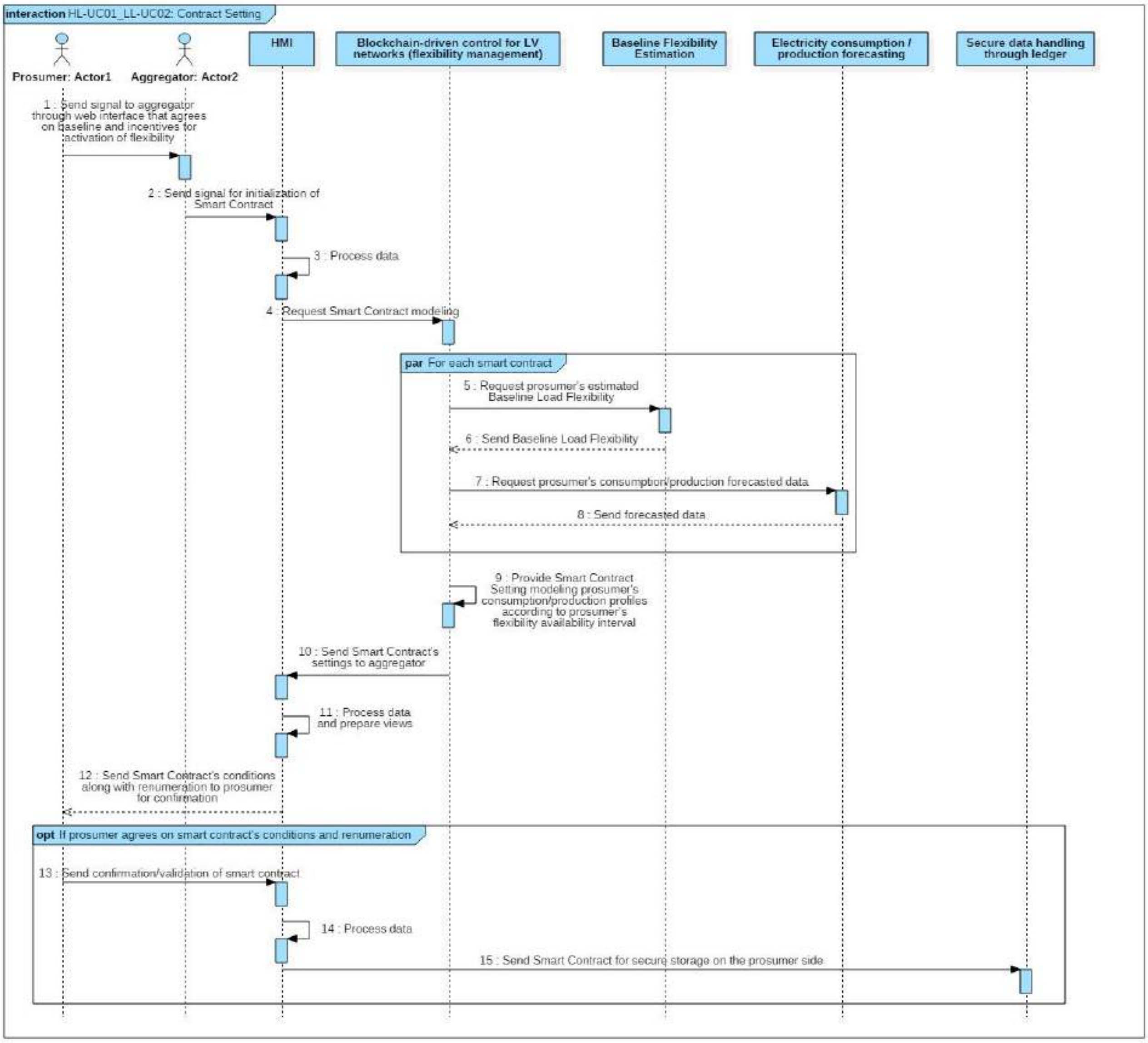
Table 5 HL-UC01_LL-UC02: Contract Setting

Generic Description	
Use Case Name	**HL-UC01_LL-UC02: Contract Setting**
Version	V0.4
Authors	E@W, TUC
Last Update	1st Version with D2.2 Update within December 2018

Brief Description	Aggregator and prosumers agree on baseline and incentives for activation of flexibility through the initialization of the 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 (curve).
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 The smart contract can be stored in a secured way on the prosumer's side.
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 availabaility interval in case of request.
Involved Actors	Aggregator, Prosumer
Use Case Initiation	Aggregator and prosumers agree on baseline and incentives for flexibility activation.
Main Flow	Begin 1. The prosumer agrees on baseline and incentives and send signal to aggregator through web interface. 2. The aggregator sends signal for initialization of smart contract through the HMI. 3. The HMI receives the aggregator's request and processes the input data. 4. The HMI request from the component Blockchain-driven control for LV networks (flexibility management) in order to perform calculations for smart contract modeling. 5. The Blockchain-driven control for LV networks requests prosumer's estimated baseline load flexibility from the component Baseline Flexibility Estimation. 6. The Baseline Flexibility Estimation sends the requested data. 7. The Blockchain-driven control for LV networks requests prosumer's consumption/production forecasted data from the component

	Electricity consumption/production forecasting. 8. The Electricity consumption/production forecasting returns the requested data. 9. The component Blockchain-driven control for LV networks sets the conditions for smart contract modeling. 10. The Blockchain-driven control sends the modeled smart contract to aggregator through the HMI. 11. The HMI processes the received data and prepare views for the aggregator. 12. The aggregator sends the smart contract's conditions along with the renumeration to prosumer for confirmation through web interface of HMI. 13. The prosumer sends the confirmation/validation of smart contract to aggregator's HMI through web interface. 14. The HMI receives and processes the input data. 15. The HMI sends smart contract for secure storage on the prosumer side.
Alternative Cources	Some prosumers do not agree on baseline and incentives for activation of flexibility (initialization of Smart Cotract).
Relationships with other Use Cases	
Architectural Elements / Services Involved	HMI; Blockchain-driven control for LV networks; Baseline Flexibility Estimation; Electricity consumption/production forecasting; Secure data handling through 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
Addressed requirements of the system	

Notes (Optional)	-
UML Sequence Diagram	



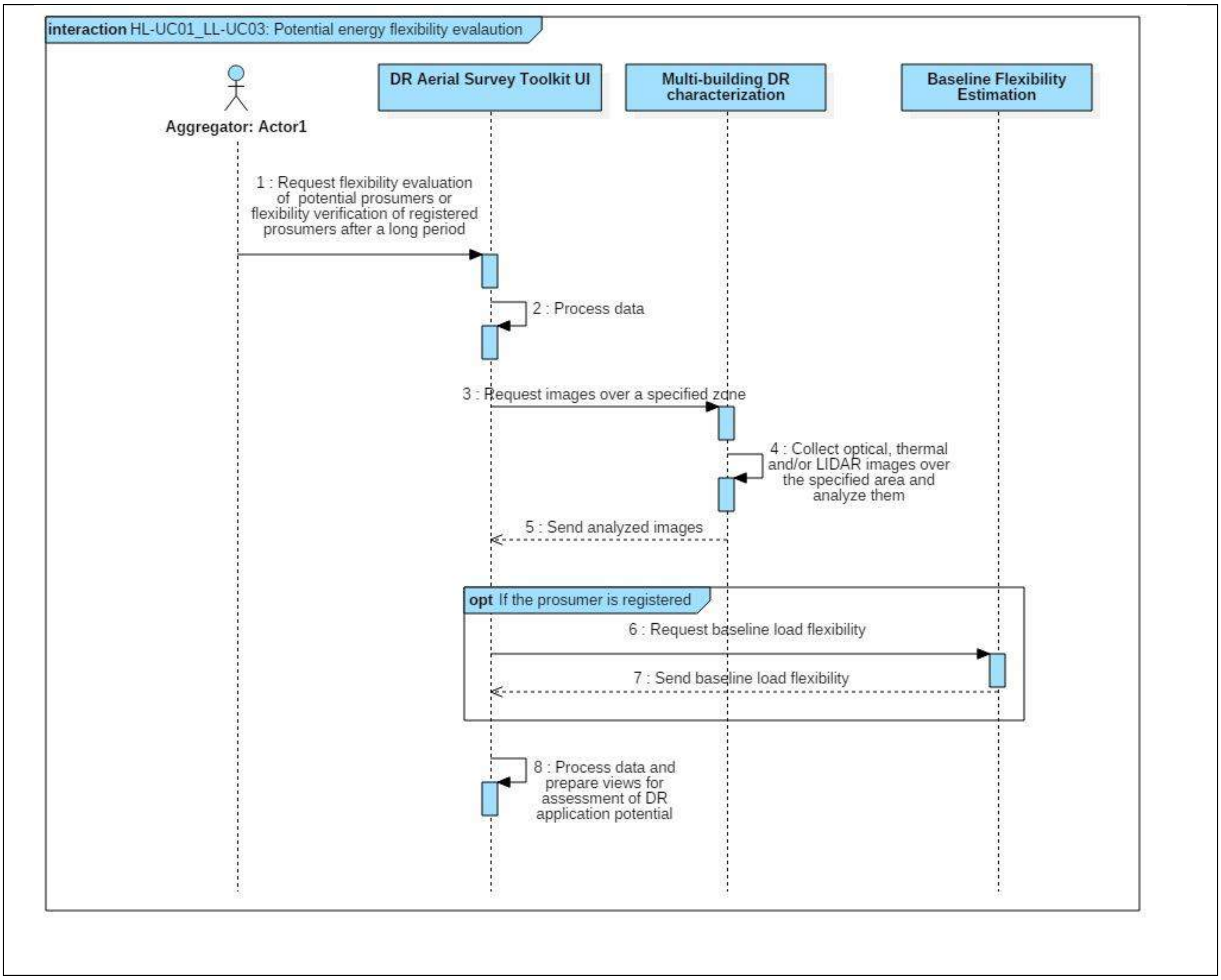
5.1.3 HL-UC01_LL-UC03: Potential energy flexibility evaluation

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

Generic Description	
Use Case Name	**HL-UC01_LL-UC03: Potential energy flexibility evaluation**
Version	V0.4

Authors	ENG, TUC, E@W
Last Update	1st Version with D2.2 Update within December 2018
Brief Description	Aggregator evaluates the potential energy flexibility guaranteed by prosumers using drones for aerial surveying in combination with 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 in an appropriate way 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 aggregator identifies new qualified prosumers. Concerning the resistered prosumers, the aggregator receives a quick estimation of their flexibility after a long period of time.
Involved Actors	Aggregator, Prosumer
Use Case 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 DR Aerial Survey. 2. The DR Aerial Survey UI receives the request and processes the input data. 3. The DR Aerial Survey UI requests optical, thermal and/or LIDAR images over a specified area of building assets. 4. The component Multi-building DR characterization collects the requested images and analyzes them. 5. The Multi-building DR characterization returns the analyzed images.

	6. In case of registered prosumer, the DR Aerial Survey UI requests the prosumer's baseline load flexibility. 7. The Baseline Flexibility Estimation returns the requested data. 8. The DR Aerial Survey UI processes the received data, prepare view for assessment of DR application potential and present them to aggregator.
Alternative Cources	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 Use Cases	
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 *Main Tasks Involved* *Main Technical Partners Involved*	WP3 T3.2 & T3.4 TU, CERTH, KIWI, E@W, TUC
Addressed requirements of the system	
Notes (Optional)	-
UML Sequence Diagram	



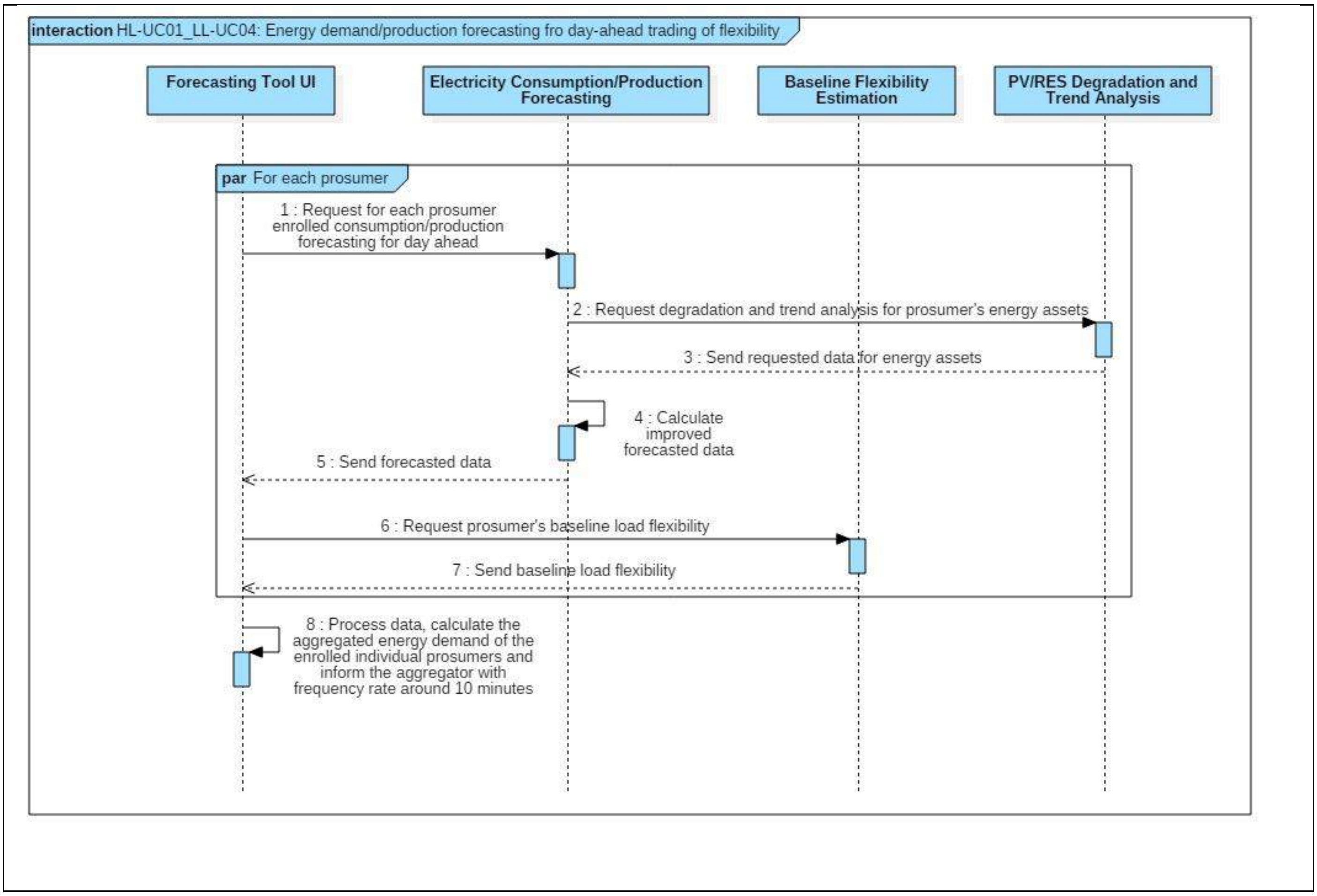
5.1.4 HL-UC01_LL-UC04: Energy demand/production forecasting for day-ahead trading of flexibility

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

Generic Description	
Use Case Name	**HL-UC01_LL-UC04: Energy demand/production forecasting for day-ahead trading of flexibility**
Version	V0.4
Authors	ENG, TUC, E@W
Last Update	1st Version with D2.2 Update within December 2018
Brief Description	Aggregator receives from each prosumer enrolled the individual energy

	demand/production values for the next day and create and send a forecast of the aggregated energy demand of all of their individual clients to the DSO which use it to forecast future congestion points. Forecasting tools have to provide the next-days prediction at different granularity (prosumer, aggregator, DSO). The HMI have to show these values with frequency rate around 10 minutes (5 min or one hour are also possible sample rates).
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, Prosumer, DSO
Use Case 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 around 10 minutes from the component Electricity consumption/production forecasting. 2. The component electricity consumption/production forecasting requests degradation and trend analysis data from the component PV/RES Degradation and Trend Analysis. 3. The PV/RES Degradation and Trend Analysis returns the requested data. 4. The Electricity consumption/production forecasting calculates improved forecasted data; 5. The Electricity consumption/production forecasting returns the forecasted data to Forecasting Tool UI. 6. The Forecasting Tool UI requests prosumer's baseline load flexibility from the component Baseline Flexibility Estimation. 7. The Baseline Flexibility Estimation returns the requested data. 8. The Forecasting Tool UI processes receivd data, calculates aggregated

	energy demand and inform aggregator every 10 minutes.
Alternative Cources	-
Relationships with other Use Cases	
Architectural Elements / Services Involved	Forecasting Tool UI; Electricity Consumption/Production Forecasting; Baseline Flexibility Estimation; 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
Addressed requirements of the system	
Notes (Optional)	-
UML Sequence Diagram	



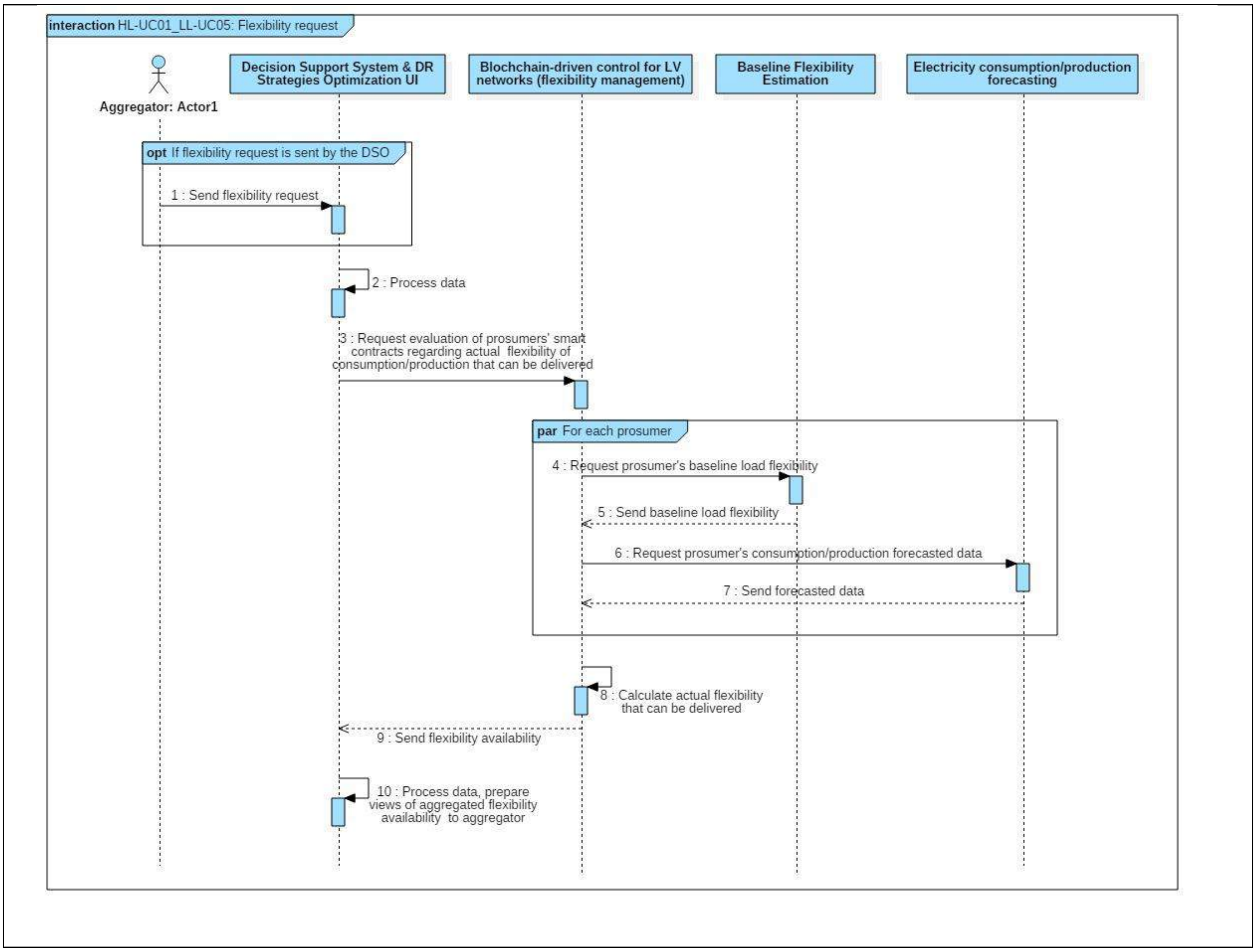
5.1.5 HL-UC01_LL-UC05: Flexibility request

Table 8 HL-UC01_LL-UC05: Flexibility request

Generic Description	
Use Case Name	**HL-UC01_LL-UC05: Flexibility request**
Version	V0.4
Authors	E@W, TUC
Last Update	1st Version with D2.2 Update within December 2018
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 forecast, 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
Use Case 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 send flexibility request through the UI of the component Decision Support System & DR Strategies Optimization in case that a previous request had been sent by the DSO. 2. The Decision Support System & DR Strategies Optimization receives and process the input data. 3. The Decision Support System & DR Strategies Optimization send requests for evaluation of prosumers' flexibility availability to the component Blockchain-driven control for LV networks (flexibility management). 4. The Blockchain-driven control for LV networks requests baseline load flexibility for each registered prosumer. 5. The Baseline Flexibility Estimation returns the requested data. 6. The Blockchain-driven control for LV networks requests prosumer's consumption/production forecasted data from the component Electricity consumption/production forecasting. 7. The Electricity consumption/production forecasting returns the forecasted data. 8. The Blockchain-driven control for LV networks calculates the aggregated flexibility that can be delivered. 9. The Blockchain-driven control for LV networks sends the aggregated flexibility availability to the Forecasting Tool UI.

	10. The Forecasting Tool UI process data and prepare views of total flexibilty availability to aggregator.
Alternative Cources	DSO applies traditional methods of peak loads reduction.
Relationships with other Use Cases	
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
Addressed requirements of the system	
Notes (Optional)	-
UML Sequence Diagram	



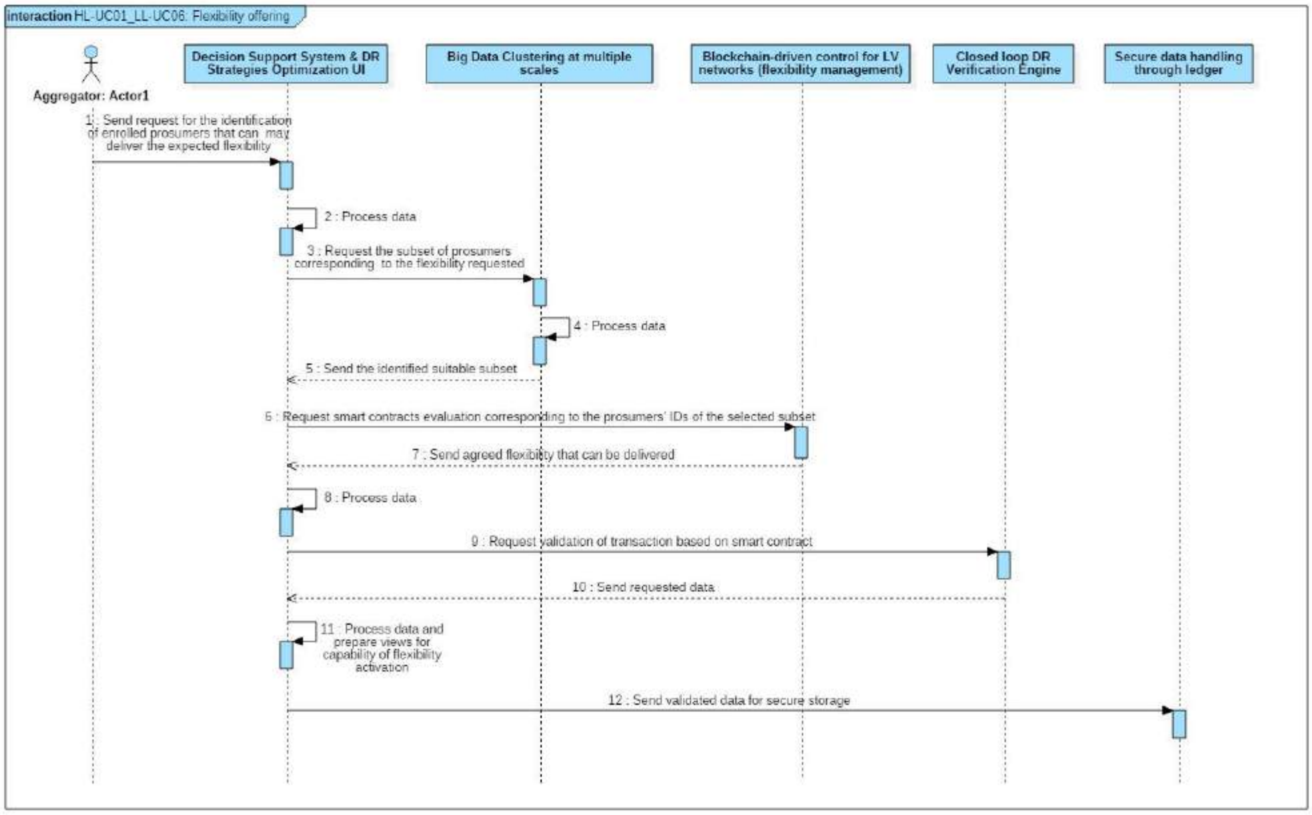
5.1.6 HL-UC01_LL-UC06: Flexibility offering

Table 9 HL-UC01_LL-UC06: Flexibility offering

Generic Description	
Use Case Name	**HL-UC01_LL-UC06: Flexibility offering**
Version	V0.4
Authors	E@W, TUC
Last Update	1st Version with D2.2 Update within December 2018
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 Big Data Clustering at multiple scale sends the identified subset to the Decision Support System & DR Strategies Optimization UI. 6. The Decision Support System & DR Strategies Optimization UI requests smart contracts evaluation of the identifies subset. 7. The Blockchain-driven control for LV networks returns agreed flexibility that can be delivered. 8. The Decision Support System & DR Strategies Optimization UI processes the received data. 9. The Decision Support System & DR Strategies Optimization UI requests validation of transaction from the component Closed loop DR Verification Engine.

	10. The Closed loop DR Verification Engine returns the requested data. 11. The Decision Support System & DR Strategies Optimization UI processes received data and prepares views for capability of flexibility activation to aggregator. 12. The Decision Support System & DR Strategies Optimization UI sends the validated data for secure storage to the component Secure data handling through ledger.
Alternative Cources	The identified prosumers do not agree to deliver their flexibility.
Relationships with other Use Cases	
Architectural Elements / Services Involved	Decision Support System & DR Strategies Optimization UI; Big Data Clustering at multiple scales; Blockchain-driven control for LV networks (flexibility management); Closed loop DR Verification Engine; Secure data handling through ledger;
Specific Description	
Relevance to eDREAM WPs *Main Tasks Involved* *Main Technical Partners Involved*	WP4 & WP5 T4.1, T4.2, T5.1, T5.2 & T5.3 TU, ENG, E@W, TUC, EMOT
Addressed requirements of the system	
Notes (Optional)	-
UML Sequence Diagram	



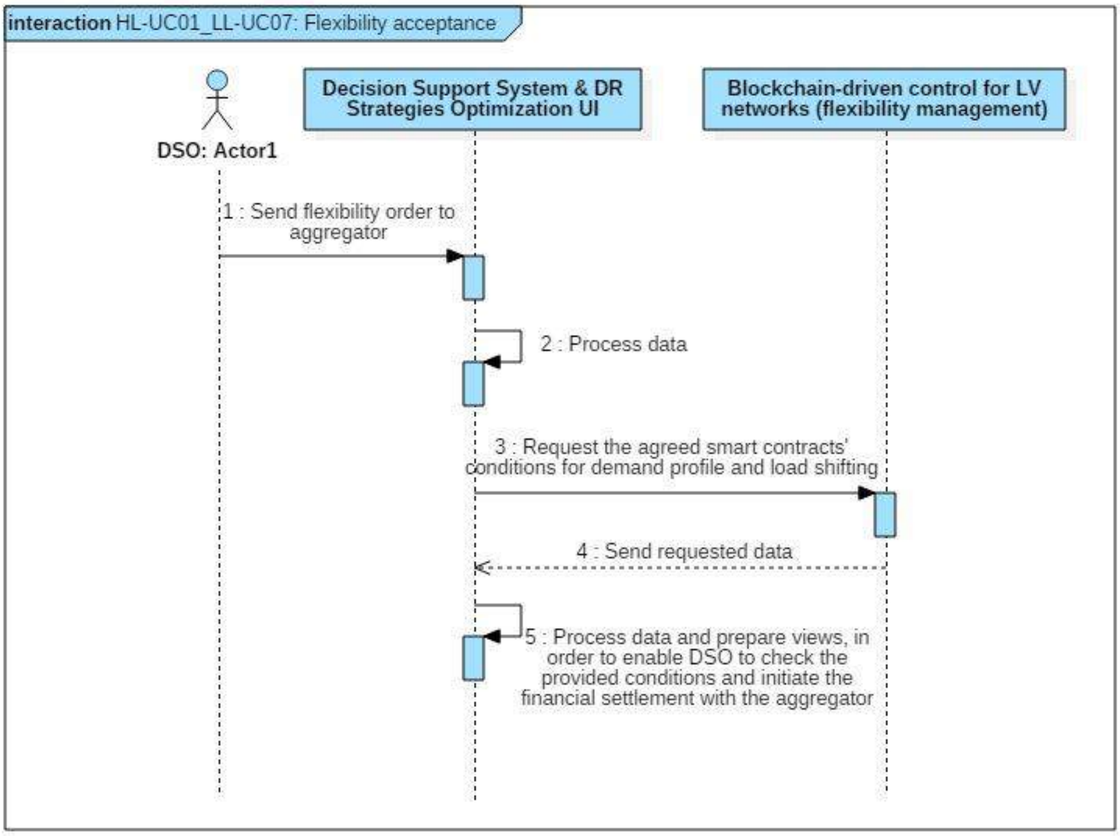
5.1.7 HL-UC01_LL-UC07: Flexibility acceptance

Table 10 HL-UC01_LL-UC07: Flexibility acceptance

Generic Description	
Use Case Name	**HL-UC01_LL-UC07: Flexibility acceptance**
Version	V0.4
Authors	E@W, TUC, ENG
Last Update	1st Version with D2.2 Update within December 2018
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 its flexibility offer/s are accepted.
Post-Conditions	The Aggregator is ready for initiating the flexibility provisioning.
Involved Actors	DSO, Aggregator
Use Case 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 UI of the component Decision Support System & DR Strategies Optimization. 2. The Decision Support System & DR Strategies Optimization UI receives the request and processes the preferences. 3. The Decision Support System & DR Strategies Optimization UI requests from the component Blockchain-driven control for LV networks the smart contracts conditions concerning demand profile and load shifting (that is going to be activated). 4. The Blockchain-driven control for LV networks returns the requested data. 5. The Decision Support System & DR Strategies Optimization UI processes data, prepare views for the conditions to be provided in order to initiate the financial settlement with the aggregator.
Alternative Cources	The Aggregator's flexibility offer/s are rejected by the DSO.
Relationships with other Use Cases	
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
Addressed requirements of the system	

Notes (Optional)	-
UML Sequence Diagram	



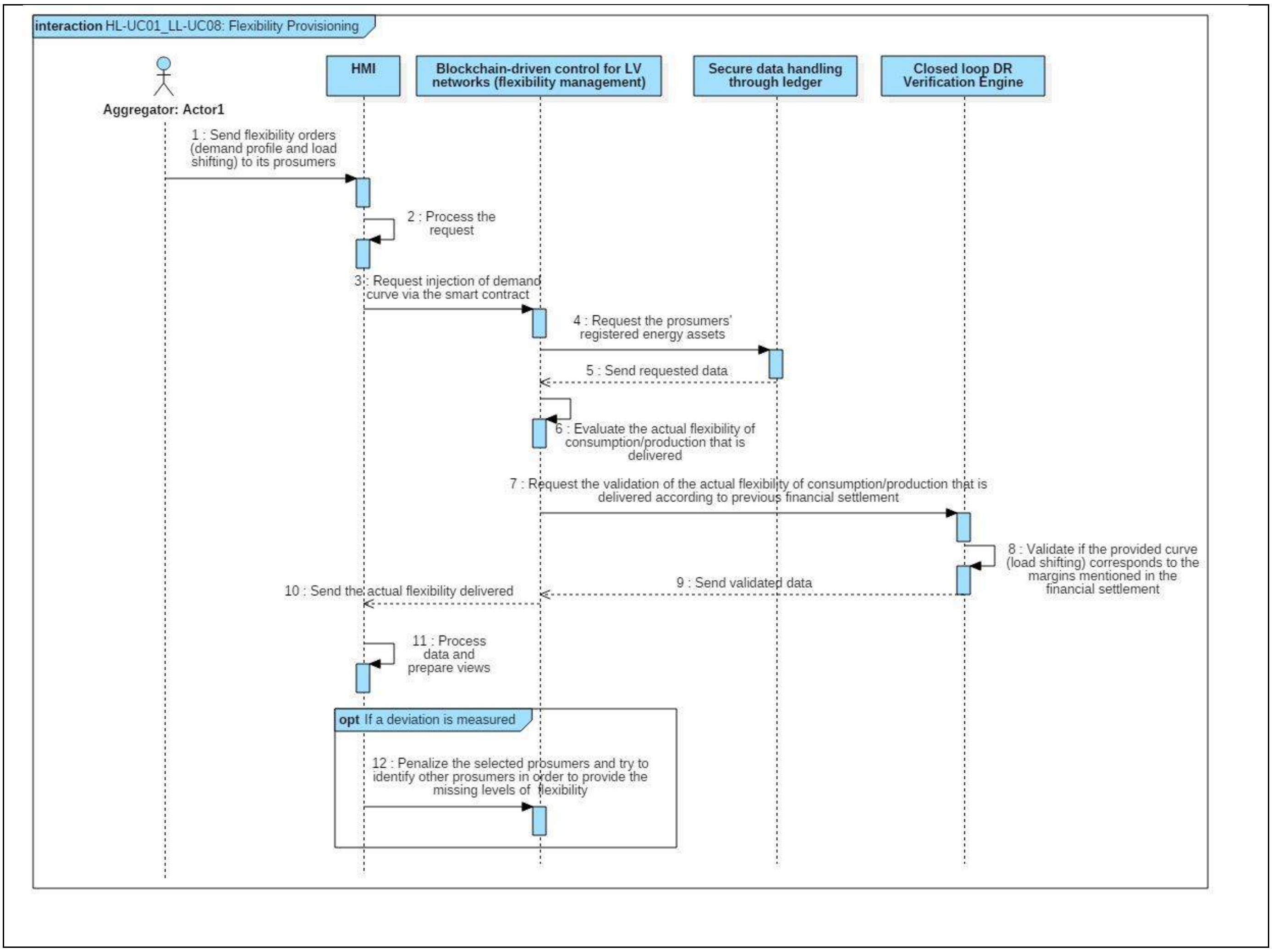
5.1.8 HL-UC01_LL-UC08: Flexibility provisioning

Table 11 HL-UC01_LL-UC08: Flexibility provisioning

Generic Description	
Use Case Name	**HL-UC01_LL-UC08: Flexibility provisioning**
Version	V0.4
Authors	E@W, TUC
Last Update	1st Version with D2.2 Update within December 2018
Brief Description	Aggregator sends the flexibility orders to its prosumers (injection of demand curve via a smart contract) in order to adjusts the load of its clients to fulfil the flexibility need. The prosumers that followed the provided curve by

	shifting their load will receive payment from the aggregator for the flexibility provision based on their flexibility contract (settlement).
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
Use Case Initiation	The Aggregator receives a flexibility offer by the DSO.
Main Flow	Begin 1. The Aggregator send flexibility orders (demand profile and load shifting) to its prosumers through the HMI. 2. The HMI receives the request and processes the input preferences. 3. The HMI requests injection of demand curve via the smart contract from the component Blockchain-driven control for LV networks (flexibility management). 4. The Blockchain-driven control for LV networks requests the prosumers' registered energy assets from the component Secure data handling through ledger. 5. The Secure data handling through ledger sends the requested data. 6. The Blockchain-driven control for LV networks evaluates the actual flexibility of consumption/production that is delivered. 7. The Blockchain-driven control for LV networks requests the actual flexibility that is delivered from the component Closed loop DR Verification Engine. 8. The Closed loop DR Verification Engine validates if the provided curve corresponds to the margins mentioned in the financial settlment. 9. The Closed loop DR Verification Engine returns the validated data to the Blockchain-driven control for LV networks. 10. The Blockchain-driven control for LV networks sends the actual

	flexibility delivered to the HMI. 11. The HMI processes input data and prepare views. 12. If a deviation is measusred 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.
Alternative Cources	The prosumers don't deliver the ordered amount of flexibility to the aggregator.
Relationships with other Use Cases	
Architectural Elements / Services Involved	HMI; Blockchain-driven control for LV networks (flexibility management); Secure data handling through 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
Addressed requirements of the system	
Notes (Optional)	-
UML Sequence Diagram	



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

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

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

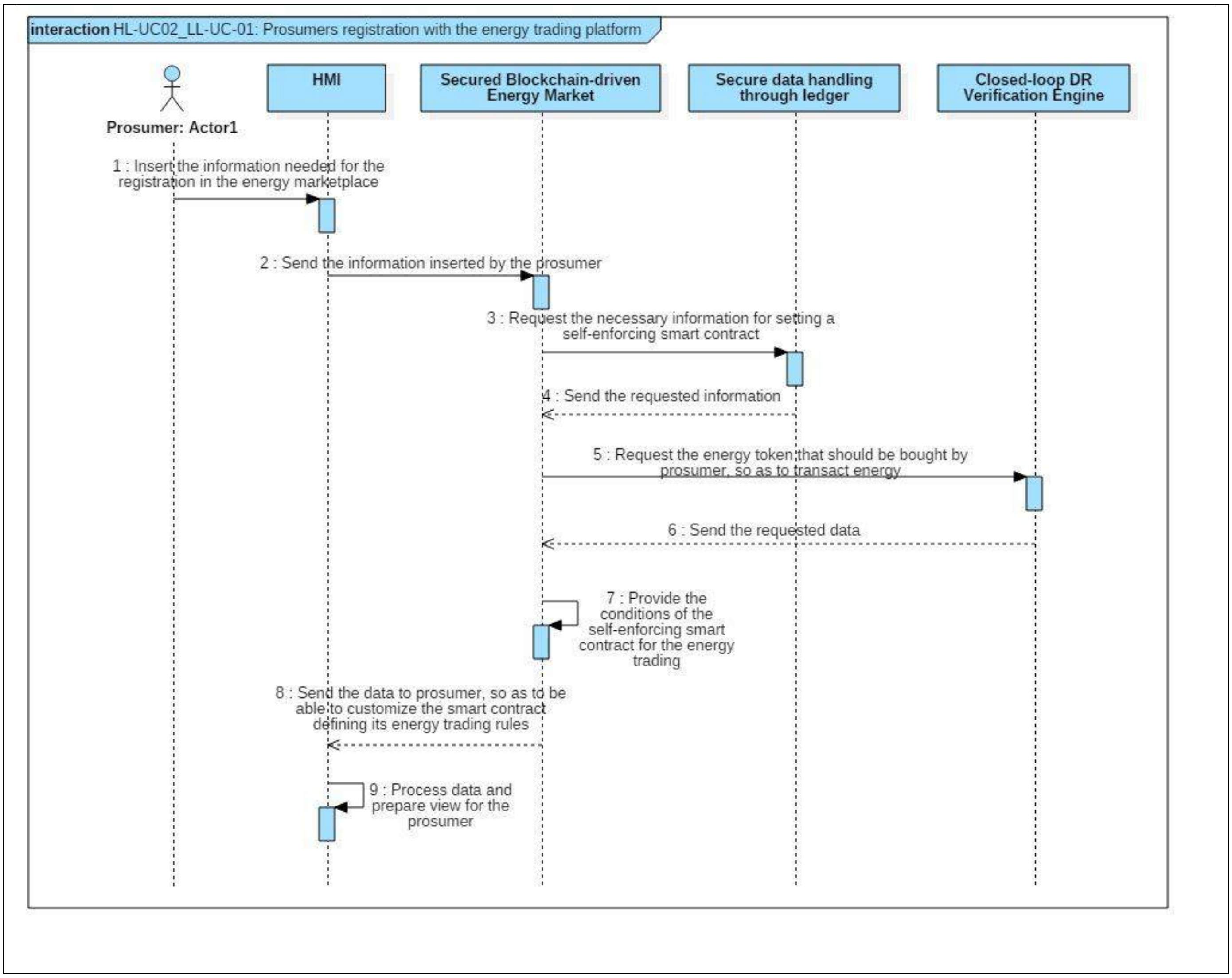
5.2.1 HL-UC02_LL-UC01: Prosumers registration with the energy trading platform

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

Generic Description	
Use Case Name	**HL-UC02_LL-UC01: Prosumers registration with the energy trading platform**
Version	V0.2

Authors	TUC
Last Update	1st Version with D2.2 Update within December 2018
Brief Description	Prosumers register on the energy market providing their information that will be validated through self-enforcing smart contract. Prosumers must be able to buy energy tokens needed to transacting energy which will be deposit 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
Use Case 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 requests necessary information for setting a self-enforcing smart contract from the Secure data handling through ledger. 4. The Secure data handling through ledger returns the requested data. 5. The Secured Blockchain-driven Energy Market requests the energy token that should be bought by the prosumer in order to transact energy from the component Closed-loop DR Verification Engine. 6. The Closed-loop DR Verification Engine returns the requested

	data. 7. The Secured Blockchain-driven Energy Market processes the received data and provides the conditions of the self-enforcing smart contract. 8. The Secured Blockchain-driven Energy Market sends the data about the smart contract's conditions to the HMI. 9. The HMI processes the received data and prepare views, in order to enable the prosumer to customize the self-enforcing smart contract's conditions concerning the energy trading rules.
Alternative Cources	Prosumers are not willing to buy or sell energy in a decentralized (blockchain based) market.
Relationships with other Use Cases	
Architectural Elements / Services Involved	HMI; Secured Blockchain-driven energy market; Secure data handling through 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
Addressed requirements of the system	
Notes (Optional)	-
UML Sequence Diagram	



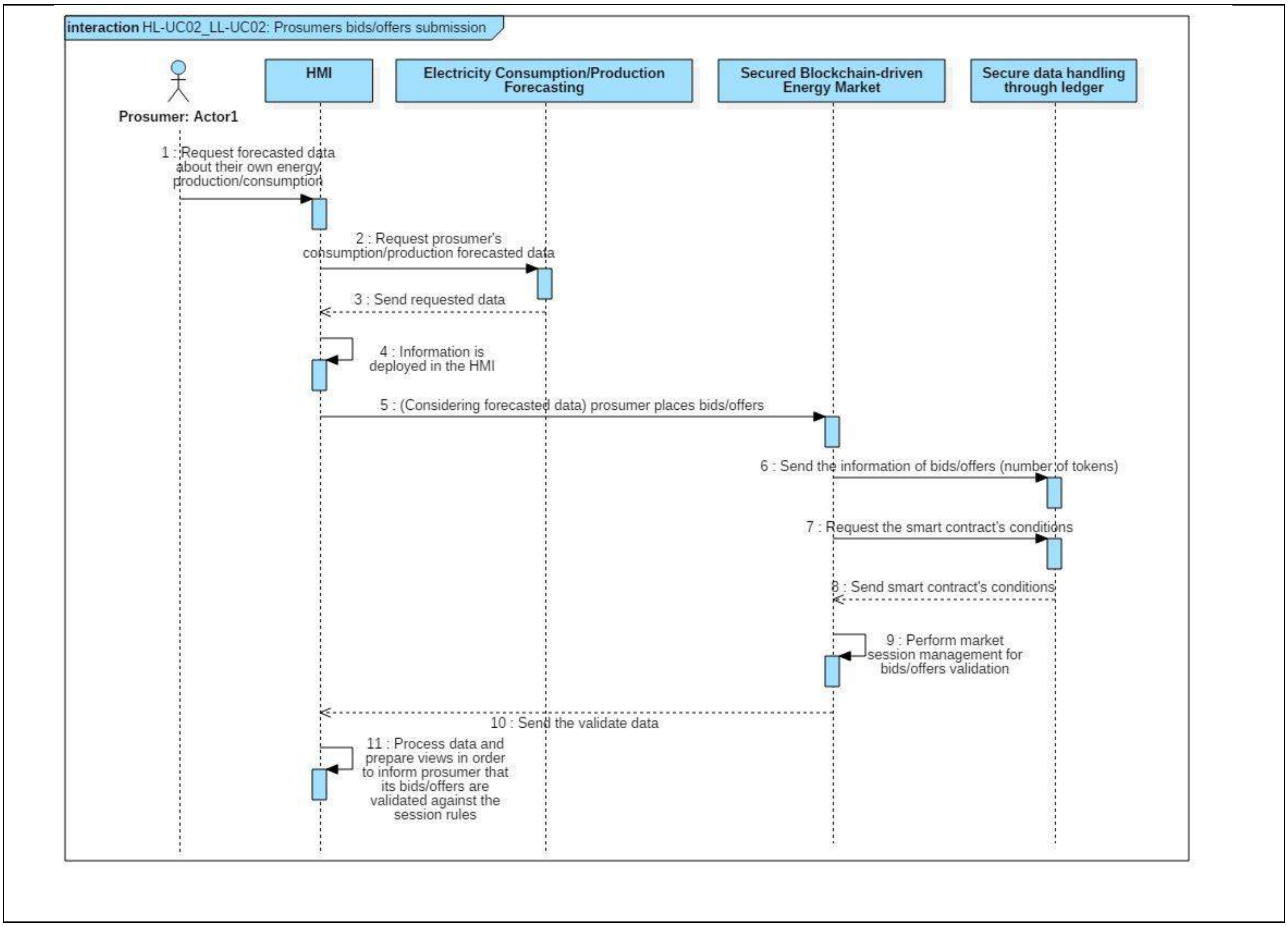
5.2.2 HL-UC02_LL-UC02: Prosumers bids/offers submission

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

Generic Description	
Use Case Name	**HL-UC02_LL-UC02: Prosumers bids/offer submission**
Version	V0.2
Authors	TUC
Last Update	1st Version with D2.2 Update within December 2018
Brief Description	Prosumers submit bids/offers associating them the number of tokens equivalent with their amount of energy (e.g. 1 token = 1 Kw). Prosumers use forecasted data to construct the bids/offers for the next market

	session.
Assumptions and Pre-Conditions	The prosumer is registered with the energy trading market.
Goal (Successful End Condition)	The aim of this UC is to establish a mechanism for prosumers to submit their bids/offers of energy.
Post-Conditions	The bids/offers are submitted.
Involved Actors	Prosumers
Use Case Initiation	The prosumer would like to submit energy bids/offers with the blockchain based energy market.
Main Flow	Begin 1. Prosumer requests 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 returs the requested data. 4. The information is deployed in the HMI. 5. (Considering forecasted data) prosumer places bids/offers in the energy market through the component Secured Blockchain-driven Energy Market. 6. The Secured Blockchain-driven Energy Market sends the information of bids/offers (number of tokens) to the Secure data handling through ledger. 7. The Secured Blockchain-driven Energy Market requests the smart contract's conditions from the Secure data handling through ledger. 8. The Secure data handling through ledger returns the requested data. 9. The Secured Blockchain-driven Energy Market performs market session management for bids/offers validation. 10. The Secured Blockchain-driven Energy Market sends the validated

	data to the HMI. 11. The HMI processes the received data and prepares views in order to inform prosumer for the validation of bids/offers against the session rules.
Alternative Cources	-
Relationships with other Use Cases	
Architectural Elements / Services Involved	HMI; Electricity Consumption/Production Forecasting; Secured Blockchain-driven energy market; Secure data handling through 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
Addressed requirements of the system	
Notes (Optional)	-
UML Sequence Diagram	



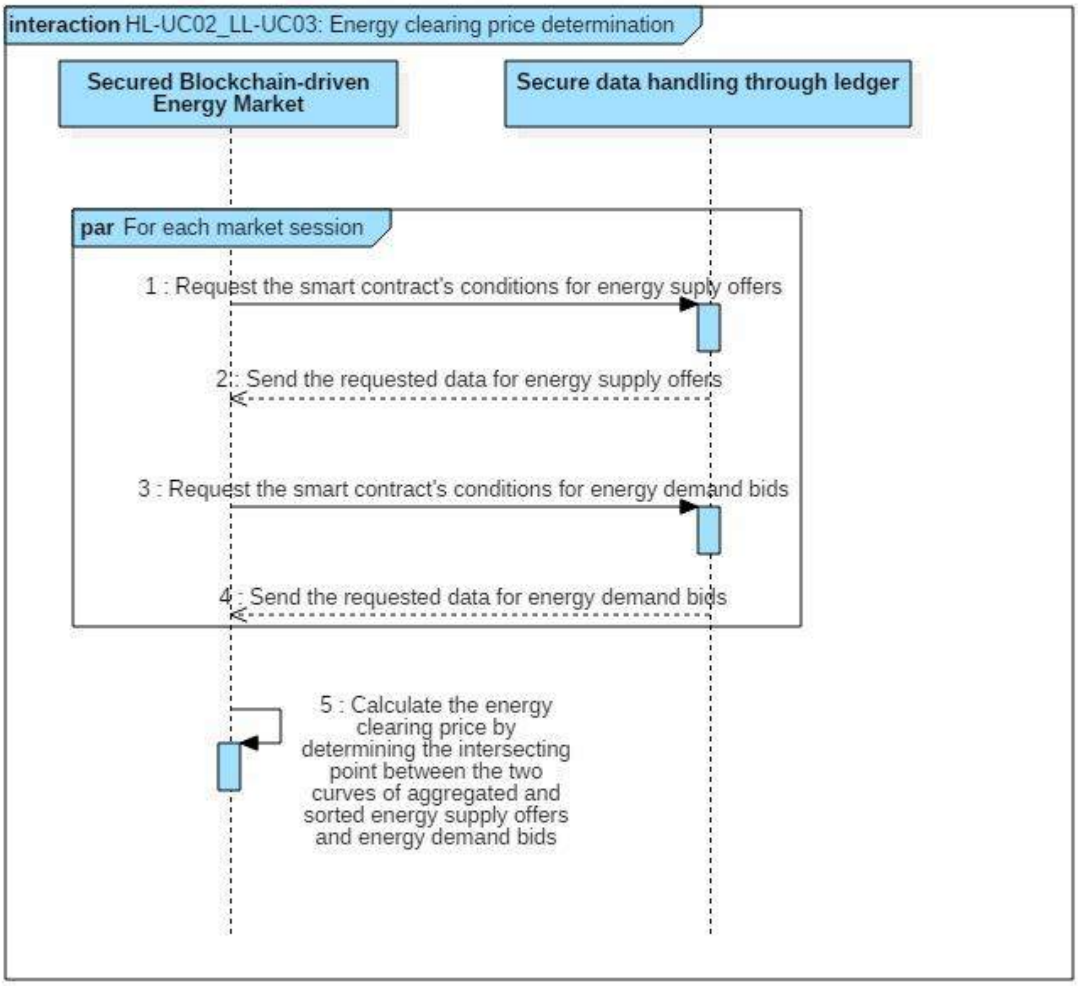
5.2.3 HL-UC02_LL-UC03: Energy clearing price determination

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

Generic Description	
Use Case Name	**HL-UC02_LL-UC03: Energy clearing price determination**
Version	V0.2
Authors	TUC
Last Update	1st Version with D2.2 Update within December 2018
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 had 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	-
Use Case Initiation	End of market session.
Main Flow	Begin 1. The Secured Blockchain-driven Energy Market requests the smart contract's conditions for energy supply offers (for each market session) from the component Secure data handling through ledger. 2. The Secure data handling through ledger returns the requested data. 3. The Secured Blockchain-driven Energy Market requests the smart contract's conditions for energy demand bids (for each market session) from the component Secure data handling through ledger. 4. The Secure data handling through ledger returns the requested data. – (Steps 1 & 2 are executed in parallel with the steps 3 &4). 5. The Secured Blockchain-driven Energy Market calcuates 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).
Alternative Cources	-
Relationships with other Use Cases	
Architectural Elements / Services Involved	Secured Blockchain-driven energy market; Secure data handling through 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

Addressed requirements of the system	
Notes (Optional)	-
UML Sequence Diagram	



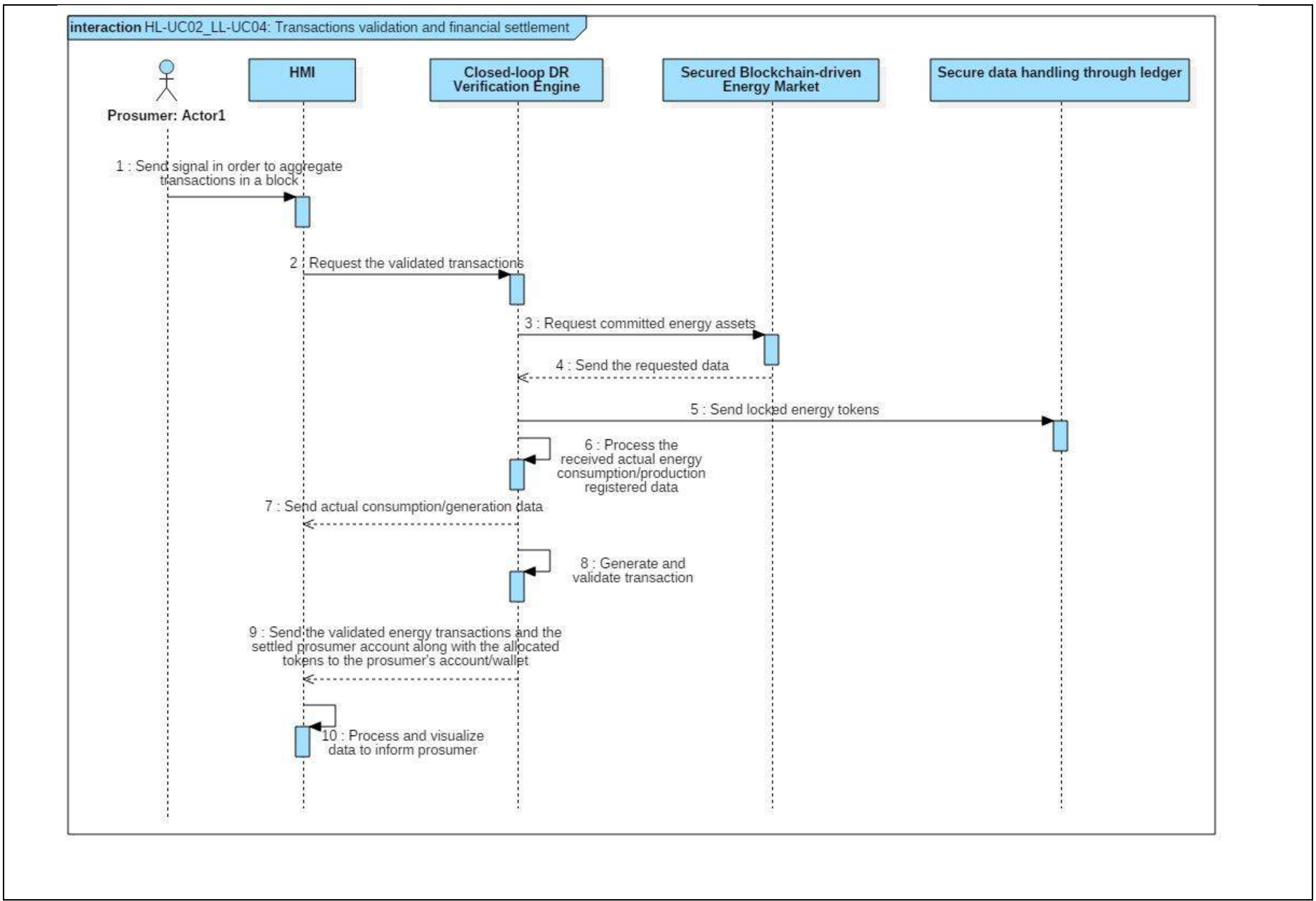
5.2.4 HL-UC02_LL-UC04: Transactions validation and financial settlement

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

Generic Description	
Use Case Name	**HL-UC02_LL-UC04: Transactions validation and financial settlement**
Version	V0.2

Authors	TUC
Last Update	1st Version with D2.2 Update within December 2018
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	Prosumer
Use Case Initiation	After the end of market session, 1 prosumer is selected as block miner.
Main Flow	Begin 1. The prosumer sends signal in order to aggregate transactions in a block through the web interface of the HMI. 2. The HMI requests the validated transactions from the component Closed-loop DR Verification Engine. 3. The Closed-loop DR Verification Engine requests the committed energy assets from the component Secured Blockchain-driven Energy Market. 4. The Secured Blockchain-driven Energy Market returns the requested data. 5. The Closed-loop DR Verification Engine sends locked energy tokens to the Secure data handling through ledger. 6. The Closed-loop DR Verification Engine processes the received actual energy consumption/production data. 7. The Closed-loop DR Verification Engine sends the actual consumption/generation data to the HMI. 8. The Closed-loop DR Verification Engine generates and validates transaction. 9. The Closed-loop DR Verification Engine sends the validated energy

	transactions and the settled prosumer's account to the HMI. 10. The HMI processes and visualizes the received data in order to inform prosumer.
Alternative Cources	-
Relationships with other Use Cases	
Architectural Elements / Services Involved	HMI; Closed-loop DR Verification Engine; Secured Blockchain-driven energy market; Secure data handling through 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
Addressed requirements of the system	
Notes (Optional)	-
UML Sequence Diagram	



5.3 High Level Use Case: 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

(Main Technical Partners involved: TU, ATOS, CERTH, TUC, E@W – Main Tasks Involved: T3.1, T3.2, T3.4, T4.2, T4.3) Aggregator receives data from Field Devices and calculates profiles of its customers updating these profiles with forecasted and real time data, and clustering them in order to categorize their participation in ancillary and balance markets.

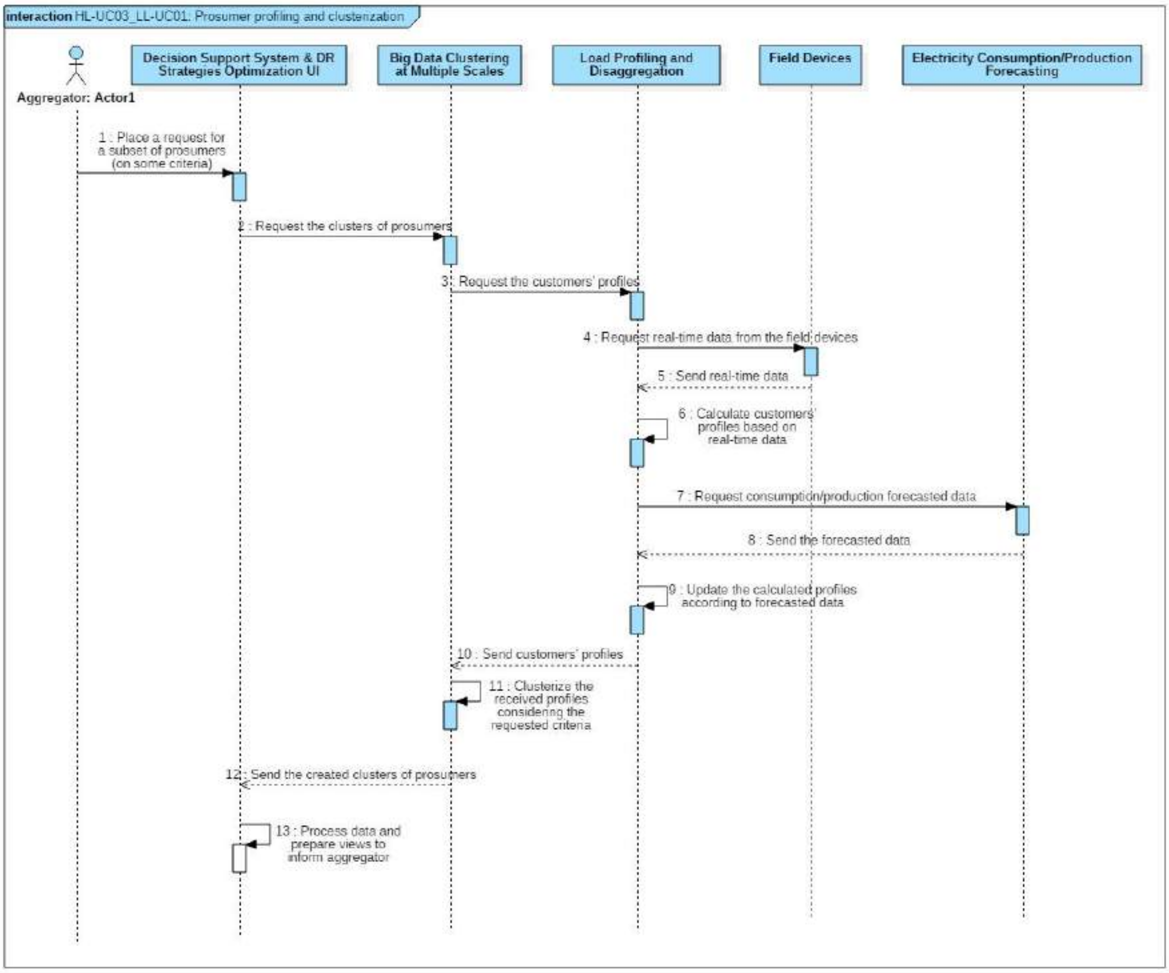


Figure 11 HL-UC03_LL-UC01: Prosumers Profiling and Clusterization

5.3.2 HL-UC03_LL-UC02: VPP capability evaluation

(Main Technical Partners involved: TUC, ENG, ASM, EMOT, CERTH – Main Tasks Involved: T3.1, T3.2, T3.3, T3.4, T4.1, T4.2, T4.3) Aggregator estimates the capability of its portfolio of prosumers for ancillary services market participation.

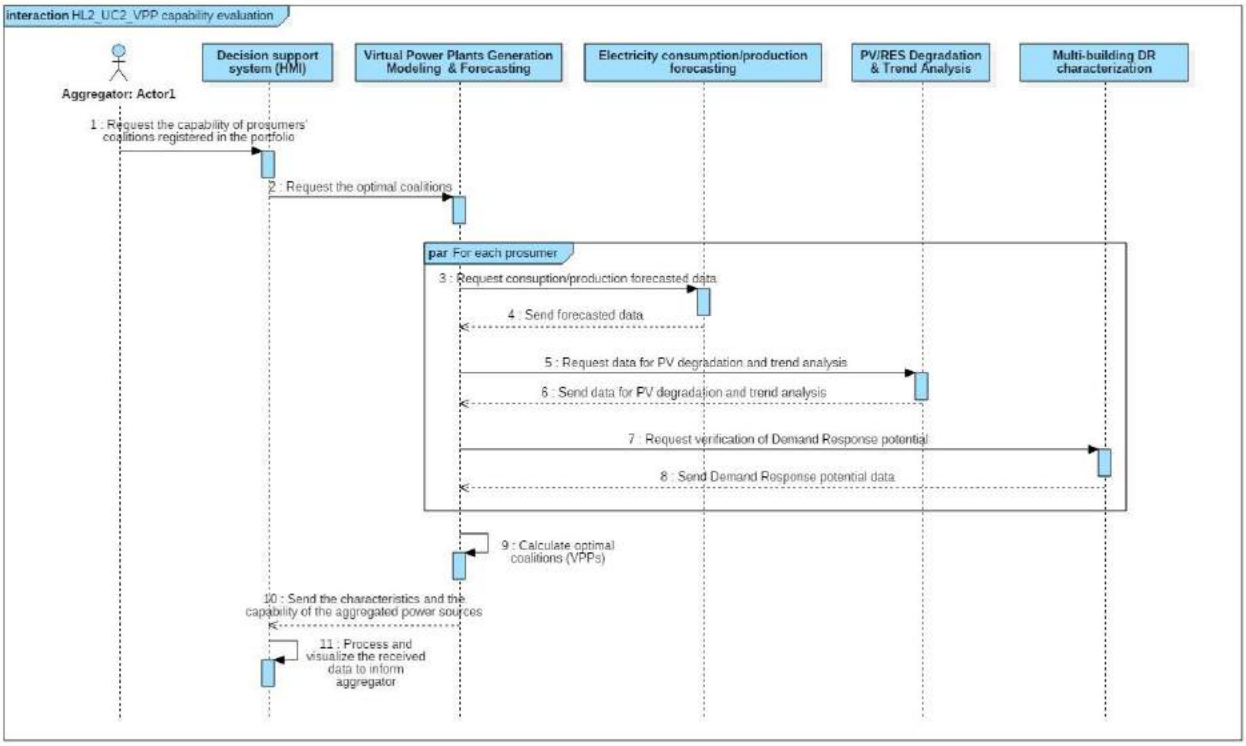


Figure 12 HL-UC03_LL-UC02: VPP capability evaluation

5.3.3 HL-UC03_LL-UC03: VPP for Reserve services

(Main Technical Partners involved: TU, TUC, ENG, CERTH, ATOS – Main Tasks Involved: T3.1, T3.2, T3.3, T4.1, T4.2, T4.3) Aggregator estimates the capability of its portfolio of prosumers maximising the utilisation and revenues from RES and classic generation sources through accessing Reserve markets as an aggregated portfolio. If there is the need to pick just one product within this market, I would sugest STOR as is one of the longest running programmes in the market. Data is near real time (minute by minute) and exposed to the System operator via a dedicated interface – a STOR terminal, essentially a purpose build computer that forwards metering data from our Operations Platform to National grid in real time. National Grid just anounced it's intention to remove the requirement for this type of interface so we can transfer data through a dedicated API, however it may take a while until we get the specificaiton and the process in place. By combining different types of RES we aim to create a more stable (linear) export that is easier to predict and asign to specific products in the Reserve market. Baseline Flexibility Estimation could be useful to estimate the curtailable capacity of loads. In this case it will need access to field device and historical data. Correct. This can also be used in conjunction with the RES portofolio to further.

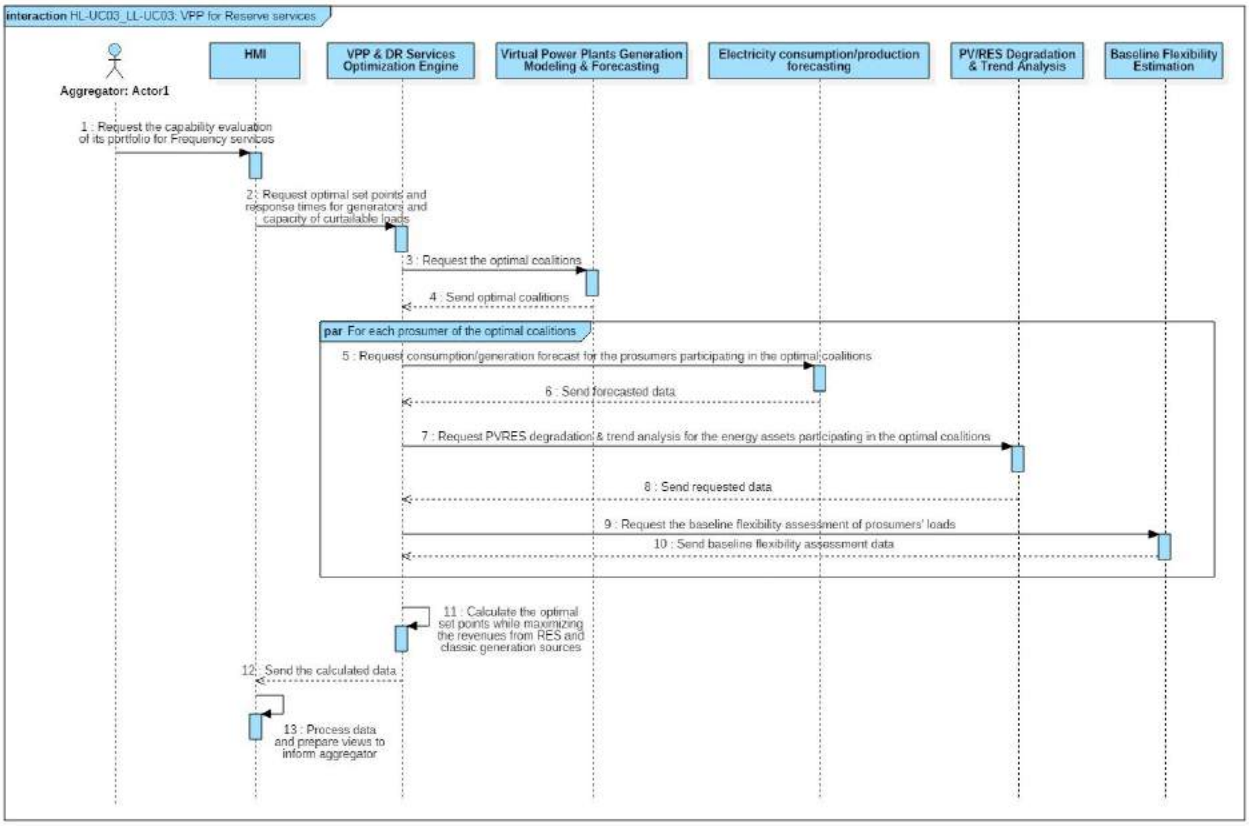


Figure 13 HL-UC03_LL-UC03: VPP for Reserve Services

5.3.4 HL-UC03_LL-UC04: VPP for Frequency services

Main Technical Partners involved: TU, TUC, ENG, CERTH, ATOS – Main Tasks Involved: T3.1, T3.2, T3.3, T4.1, T4.2, T4.3). 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. VPP generation modeling and forecasting, VPP & customer segmenmtation, Load profiling and disagregation, Baseline Flexibility estimation - all should produce outputs feeding back into Microgrid flexibility profiling. It may be also that the output from some modules it's an input for other modules.

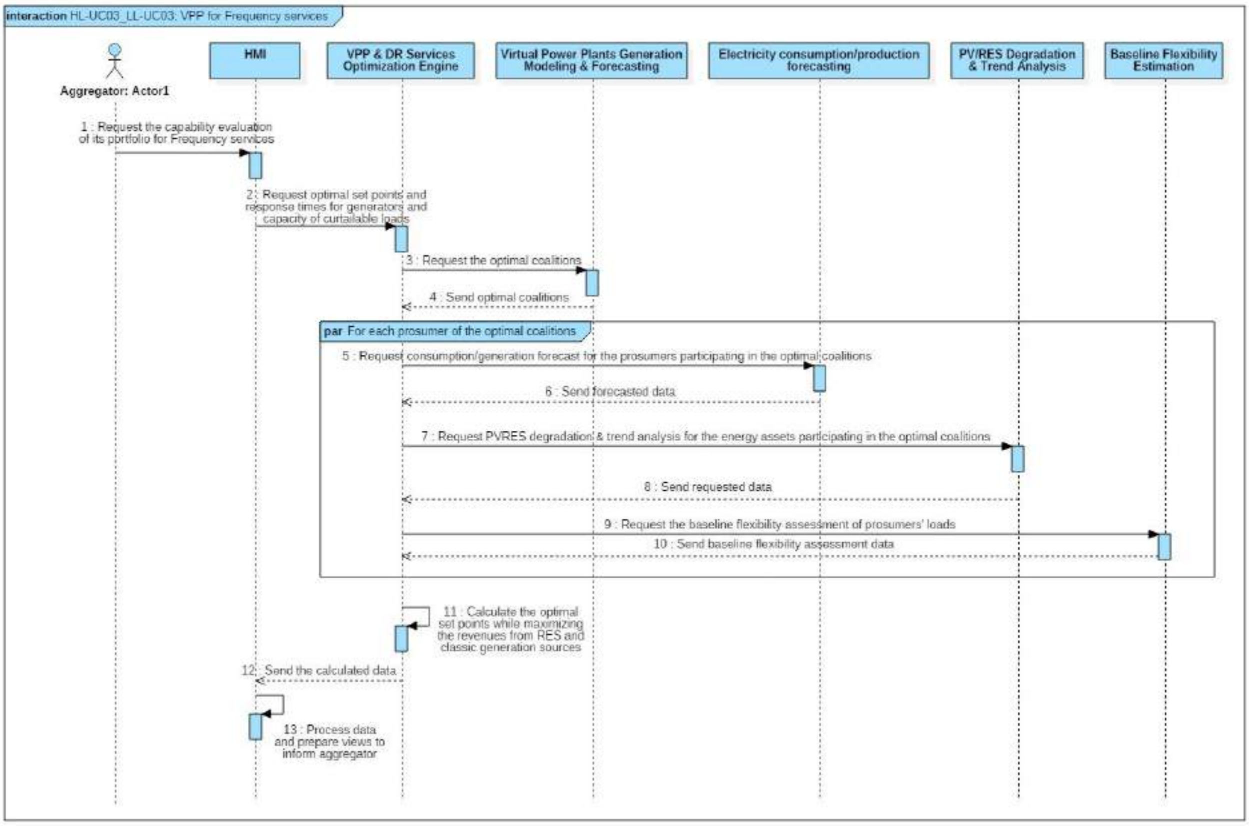


Figure 14 HL-UC04_LL-UC04: VPP for Frequency Services

5.3.5 HL-UC03_LL-UC05: VPP export evaluation

(Main Technical Partners involved: TUC, ATOS, CERTH – Main Tasks Involved: T3.1, T3.3, T4.1, T4.2, T4.3). Aggregator accurately estimates 30 minutes generation and load forecasts to perform big data analysis in order to profile loads to be shed and to identify set points of dispatchable generators as well as response times from each type of generation asset. Aggregator need to know the VPP export capacity to implement trading services such as intraday trading and imbalance market.

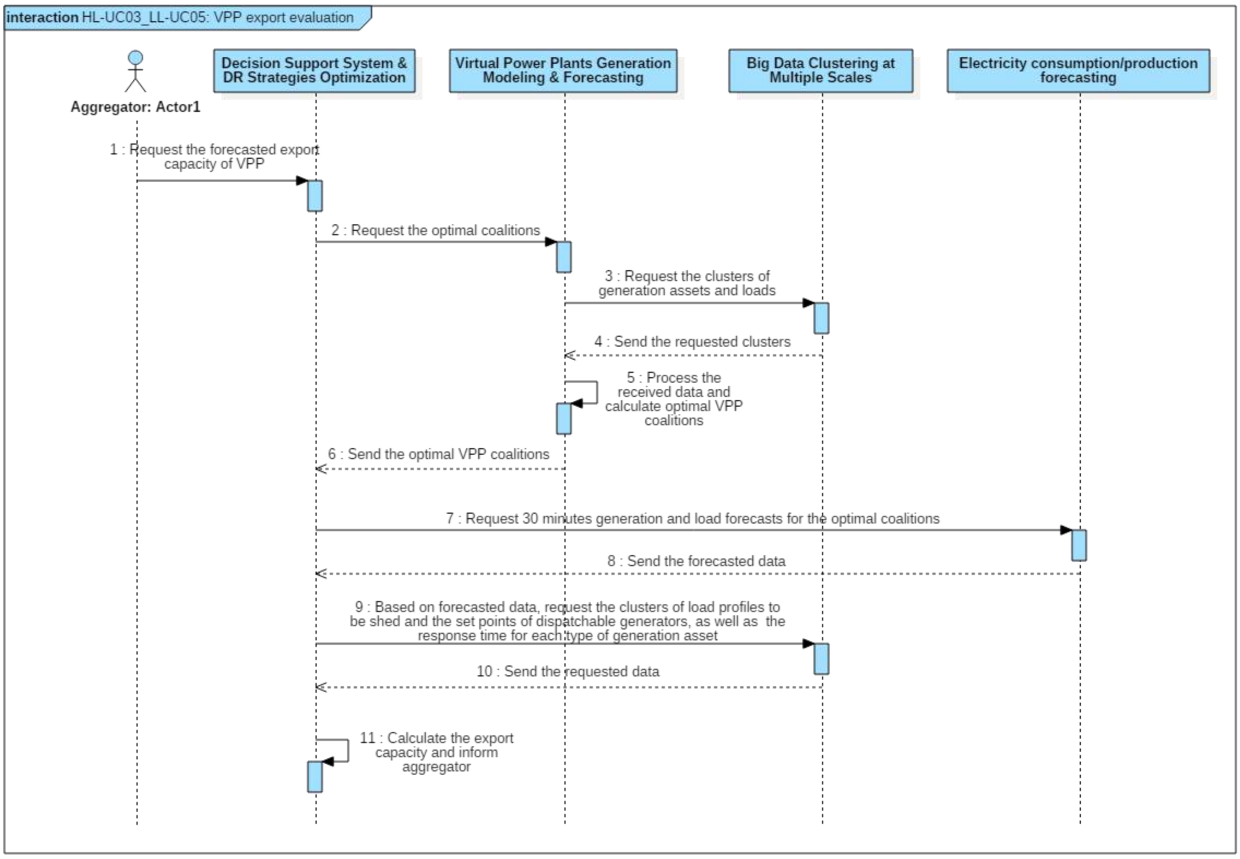


Figure 15 HL-UC03_LL-UC05: VPP export evaluation

5.3.6 HL-UC03_LL-UC06: VPP for Wholesale Market – Intraday trading

(Main Technical Partners involved: TUC, TU, CERTH, ATOS – Main Tasks Involved: T3.1, T3.2, T3.3, T4.1, T4.2, T4.3). 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.

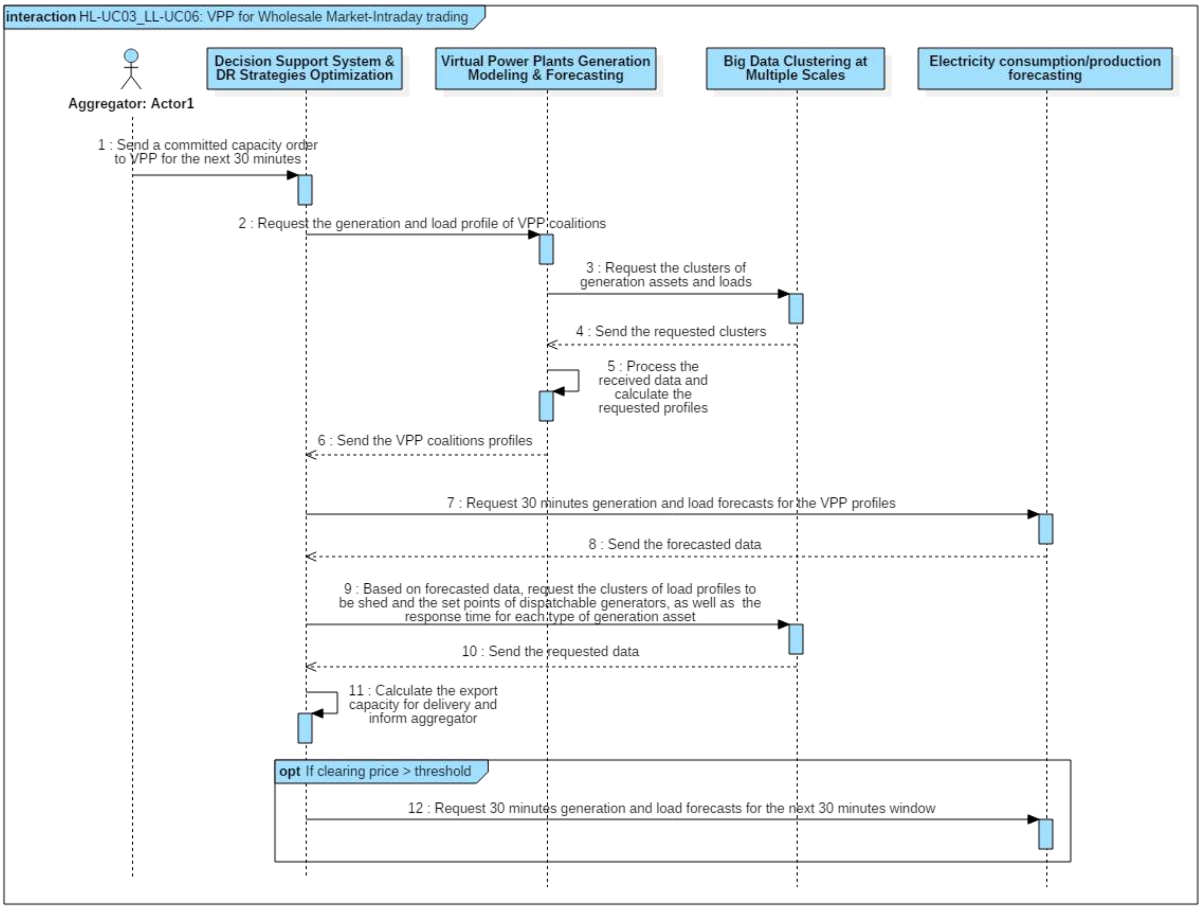


Figure 16 HL-UC03_LL-UC06: VPP for Wholesale Market – Intraday trading

5.3.7 HL-UC03_LL-UC07: VPP for Imbalance market

(Main Technical Partners involved: TUC, TU, CERTH, ATOS – Main Tasks Involved: T3.1, T3.2, T3.3, T4.1, T4.2, T4.3). 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.

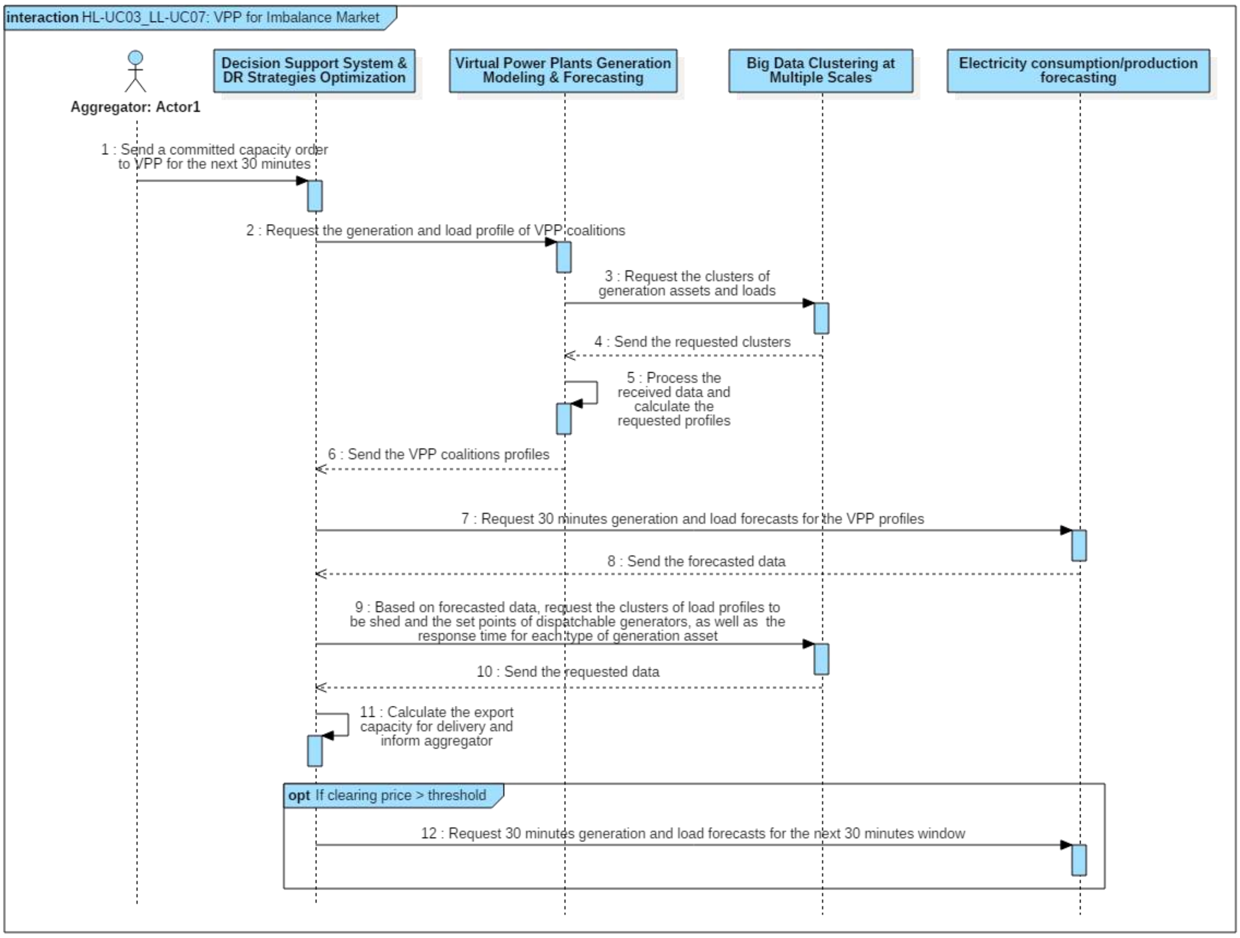


Figure 17 HL-UC03_LL-UC07: VPP for Imbalance market

6 Development View

Within this section we will provide the development view of the eDREAM system, covering aspects such as use of existing software (along with respective legal and accessibility issues), components' technical requirements and dependencies, as well as programming languages/technologies that may be used by each component.

Each partner after analysing the needs, requirements and functionalities of the component under his responsibility, has defined the development solution to be followed taking also into consideration interoperability issues concerning the communications and interactions with the other eDREAM components. Software licences and availability of tools has been also taken into account for the final decision.

In the following section, we present the existing software packages/platforms that will be used during implementation describing also their specific role in eDREAM and provide the development technologies and programming tools that will be utilized by each system component.

6.1 Existing Software

The below table presents the existing tools that can be used during the eDREAM activities.

Table 16 Existing Tools for the eDREAM system

eDREAM Tool Name	***Partner***	***Related Architectural Component***	***Explanation for re-use into eDREAM new Components***
Energy Budget Broker	*TUC*	***Electricity Consumption / Production Forecasting***	*TUC had developed a technique for assessing the current and forecasted energy budget of data centres in the context of FP7 GEYSER project. It addresses the case of data centres as large scale prosumers. We will investigate their potential adaptation for the case of small scale prosumers energy forecasting considering new specific aspects of high variance, uncertainty, of energy production/consumption, flexibility availability forecasting, etc.*
Multi-Criteria Energy Efficiency Optimizer	*TUC*	***Virtual Power Plants Generation Modelling & Forecasting***	*We had defined multi-criteria optimization techniques for data centres integration in smart grids. They may serve as starting point for the construction of optimal coalitions of energy generation sources our goal being to adapt and use them in combination with new bio-inspired heuristics allowing us to consider hybrid version between gradient descent and population based optimization*
Blockchain Services for Electronic Registration, Transacting and Processing of Assets	*TUC*	***Blockchain-driven control for LV networks***	*In the context of a bilateral contract we had defined services for blockchain based digitalization of assets. They will be specialized for the case of energy case*

			allowing us treat energy as a digital asset.
Graph-based analytics for Closed Loop DR optimal scheduling and Hypothesis Testing	*CERTH*	***Graph-based Analytics HMIs***	*CERTH had developed a visualization platform in the context of FP7 NEMESIS project. This tool will be investigated towards Demand Response programs and hypothesis testing formulation.*
Customer/Prosumers/Virtual Power Plants Clustering & Segmentation suite	*ATOS & CERTH*	***Next generation services for aggregators and customers*** *(Big Data layer)*	*CERTH had developed a clustering and segmentation technique in the context of FP7 NEMESIS project. This tool is going to be investigated concerning the clustering and segmentation of prosumers in smart grids according to their profile pattern.*
Market Session Manager (MSM) and Data Storage	*ENG*	***Secured Blockchain-driven Energy Market*** *Still to be clarified if this component can be reused or if this part will be done directly via Smart contracts (T5.2)*	*ENG has developed the Market Session Manager and Data Storage in the context of FP7 GEYSER project.* *The components are responsible for managing the market sessions and store and retrieve data related to session management.* *The knowledge acquired in the development of the tool will contribute to the realization of the decentralized marketplace for flexibility exchange, reflecting the bids/offers collection process and the clearing price algorithm into the smart contracts definition.*
Demand Response Manager	*ENG*		*Demand Response Manager (DRM) is a component supporting the interaction of DSO towards the Smart Grids developed by ENG in the context of GEYSER project. It was developed considering large scale prosumers (data centers).* *In eDREAM the potential adaptation for prosumers like domestic and SME will be investigated.*
Energy Management System (EMS)	*TU*	***Field Data Aggregation***	*The DR baseline flexibility will be estimated by this tool. This will be based on prosumer energy generation/consumption pattern and willingness to take part in a DR program as specified in the contract with Aggregator or DSO.* *DR and energy flexibility will be estimated based on the length of time and the amount of energy which can be committed to a specific DR programme.* *Moreover, monetary cost/benefit and*

			prosumer comfort will also be considered.
DER Flexibility Engine	*CERTH*	***Next generation services for aggregators and customers***	*This engine has been developed during the FP7 Inertia project. Its main functionality is to estimate the flexibility margins of generation assets. This tool will be investigated for the VPP and Microgrid flexibility profiling.*

7 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.

7.1 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:

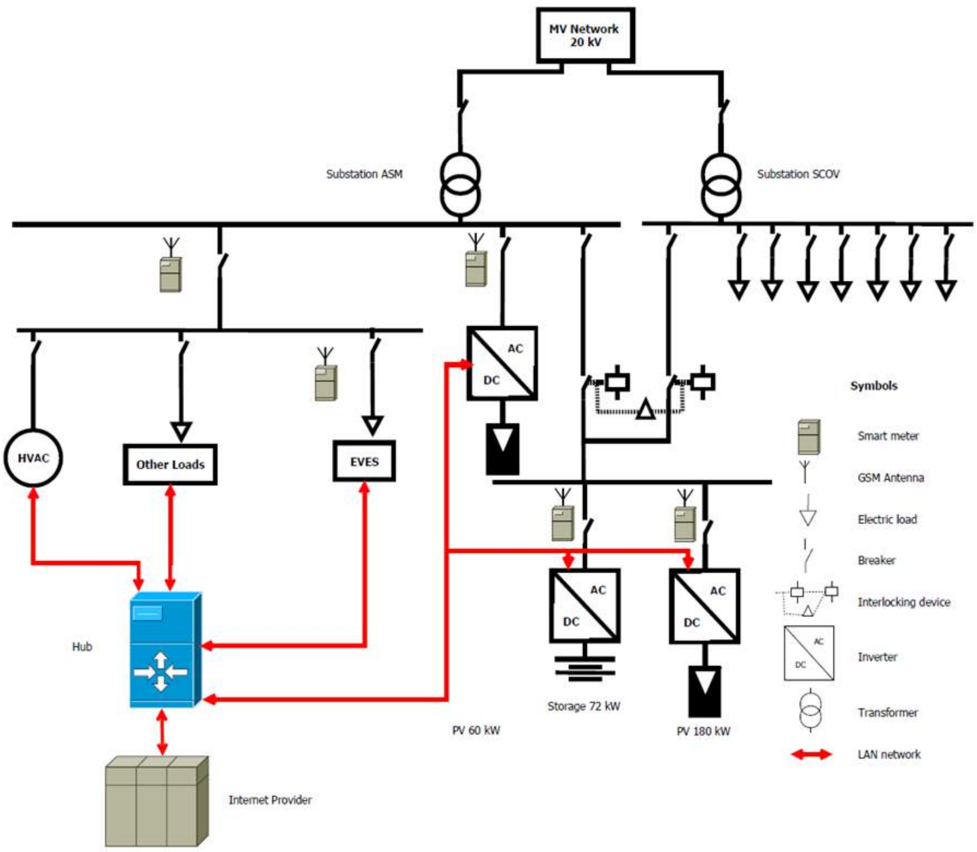


Figure 18 Internal and external connections in the Terni Micro-Grid

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.

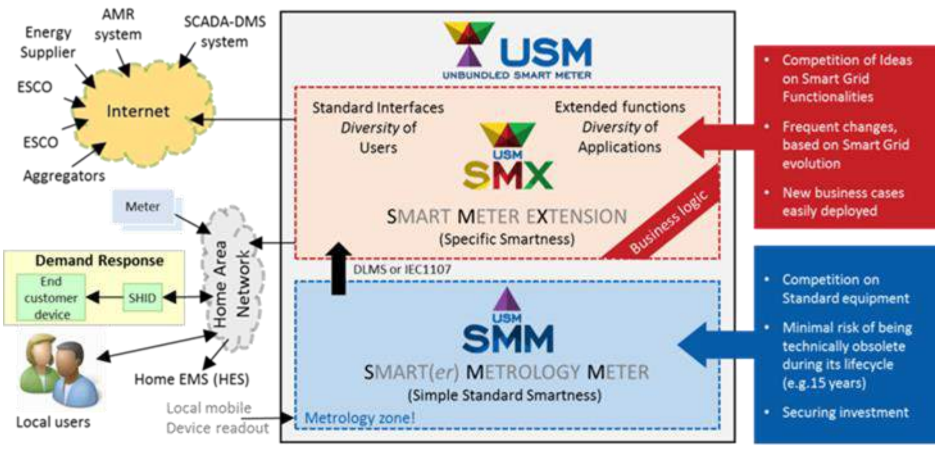


Figure 19 Unbundled Smart Meter Concept

A shown in the figure 6, the SMX is responsible for all communication, to both local network (HAN) and public IP-network (internet), as well as running necessary basic applications and third party applications. The SMs are able to communicate with the SMX for the everyday operations with the distribution power network. They are used by ASM for energy transfer measurements (absorption/consumption) in the context of billing purposes through a DSO's server. This communication framework is depicted in the follow figure.

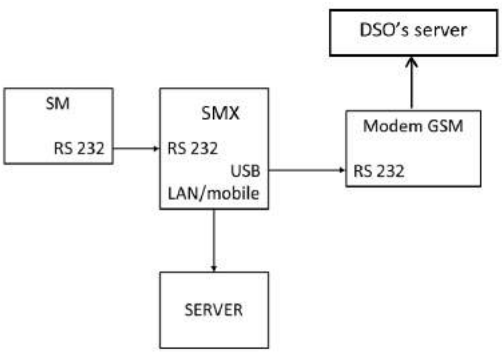


Figure 20 High-level architecture for the communication between SM and SMX

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

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

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

7.2 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.

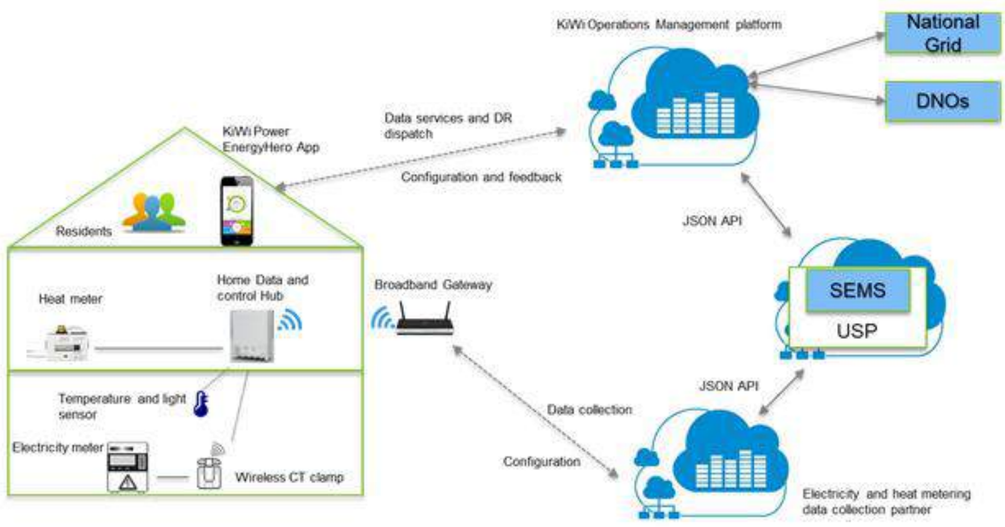


Figure 21 Residential estate in Greenwich for testing residential demand response scenarios

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.

7.3 Overall Deployment Architecture

This section presents a preliminary deployment view of the eDREAM components 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 the Field Middleware Device Manager through corresponding gateways, which will forward the information from the field devices of the pilot sites to the Middleware. The Device Manager, that is running in a dedicated PC named IoT Gateway, is equipped with all the necessary drivers of the field equipment, so as to understand and interpret the signals from the different metering devices. As a next step, the Context Broker creates different context elements of the received information and manage them. The deployment view of the eDREAM platform is presented in the following figure 9:

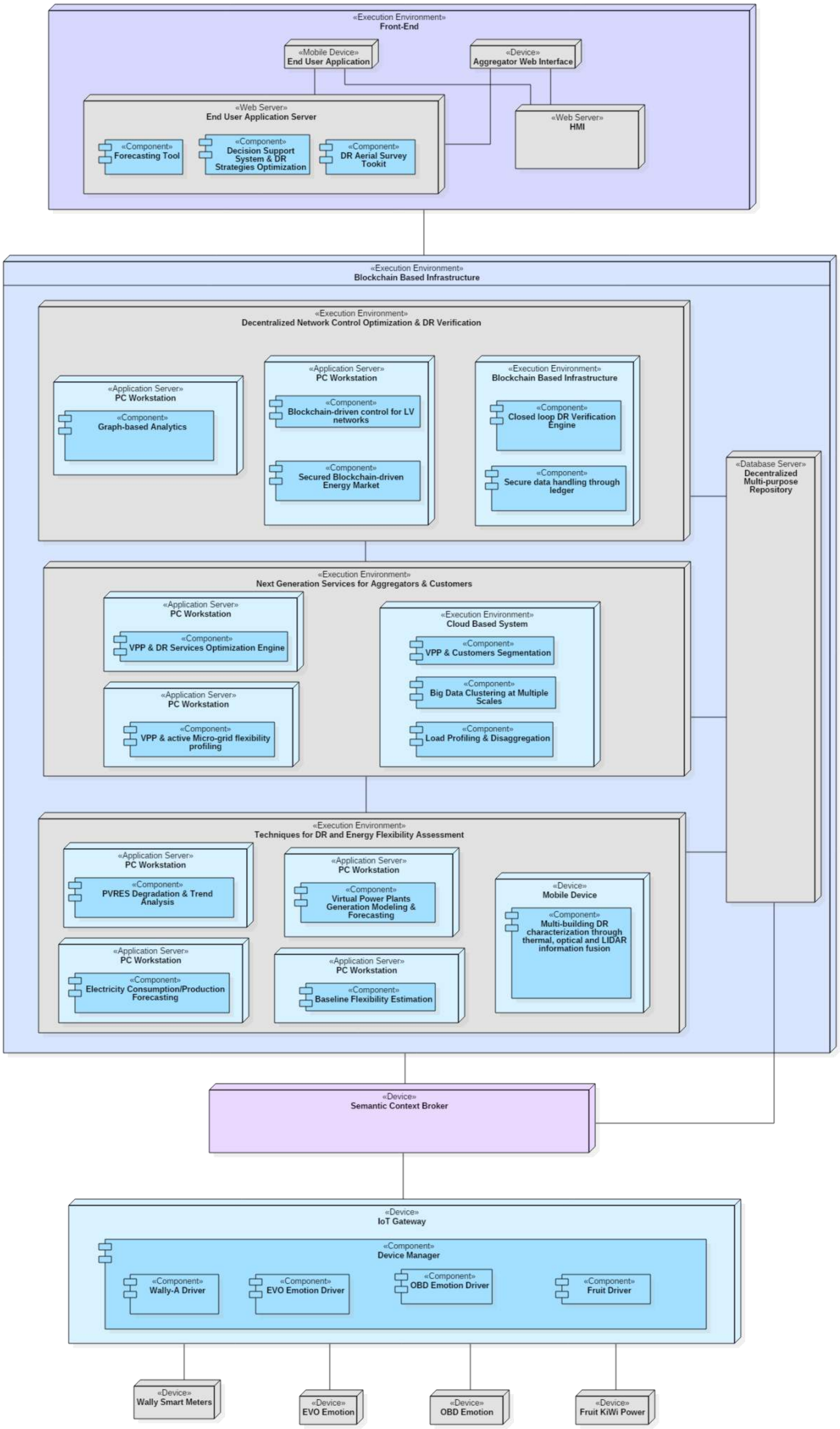


Figure 22 eDREAM Deployment View Architecture

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 Agents[2] and FIWARE Orion Context Broker[3]. The use of IoT Agents allows the integration and interaction of heterogenous 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 heterogenous 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:

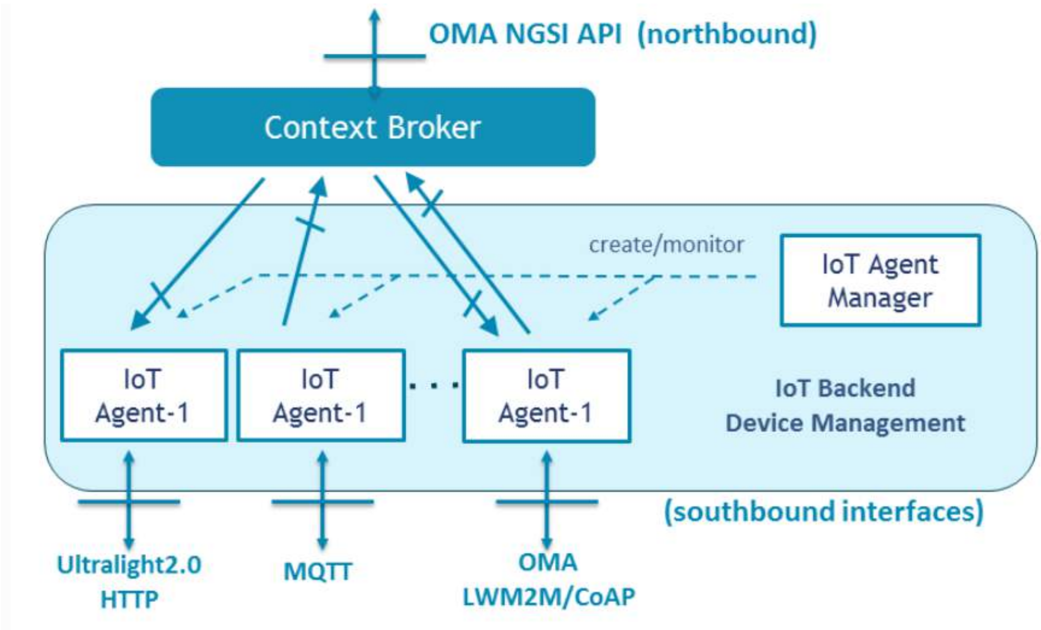


Figure 23 IoT Agents' Concept and Connection

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

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

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

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 modeling, tracking and decentralized control. A high-level deployment diagram of this infrastructure is depicted in the following figure.

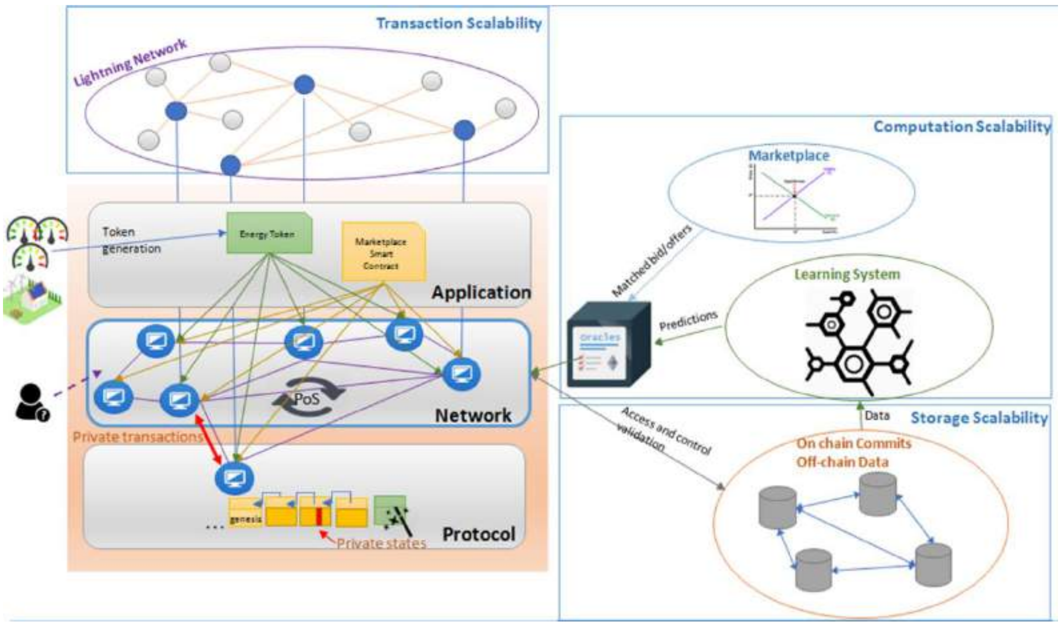


Figure 24 Blockchain-based Infrastructure

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

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

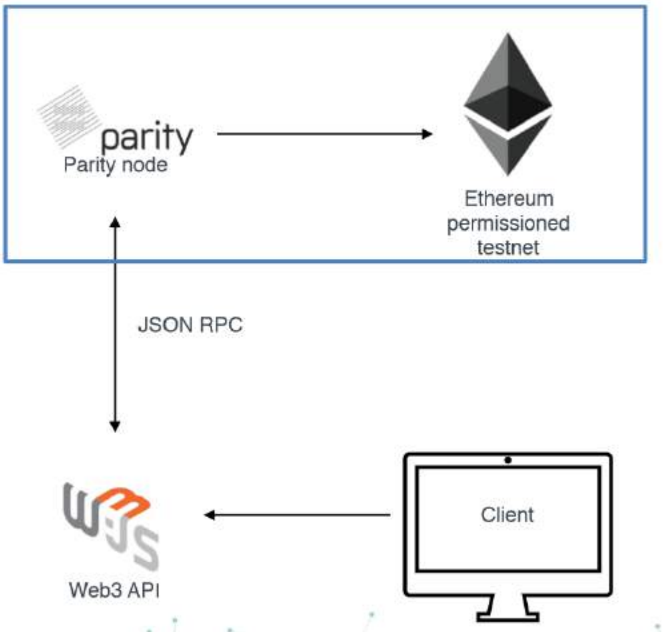


Figure 25 Testbed Topology for the Distributed Ledger Technology

7.4 Field Devices

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

Table 17 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. 

[9] https://www.parity.io/ethereum/

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 18 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 19 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 20 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 21 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	-

8 Architectural Components Detailed Specifications

Table 22 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***	• Secure data handling through ledger • Blockchain-driven control for LV networks (flexibility management) • Electricity Production/Consumption Forecasting • Big Data Clustering at Multiple Ccales • 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 23 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* • *VPP and Active Micro-grid Flexibility Profiling* • *Load Profiling & Disaggregation* • *VPP & DR Services Optimization Engine* • *Closed Loop DR Verification Engine* • *Blockchain-driven Control for LV Networks* • *DR Strategies Optimization (Front-End)*
<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 (kilometers)	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 24 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)*
Input Connections & Interfaces: From which components it receives input	• *Decentralized Multi-purpose Repository* • *Weather Forecast APIs*
Output Connections & Interfaces: To which components it sends the results	• *Virtual Power Plants Generation Modelling and Forecasting* • *Load Profiling and Disaggregation* • *PV/RES Degradation & Trend Analysis* • *VPP & DR Services Optimization Engine* **REST API**: <host>:<port>/edream-ebb/prediction/ /{granurality}/find/{microgridId}/{start}/{end}
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	-

Input Parameters					
Attribute /Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Received From***
Historical data	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	*Decentralized Multi-Purpose Repository*
Weather Forecast	Information about the weather prognostic for the following period	JSON	[{ "timestamp": 1537803600000, "temperature": 30, "radiationShortWave": 10, "windSpeed": 10, "airDensity": 10 }, ...]	**Frequency**: every 30 min	*Weather Forecast APIs*
Renewable Devices Related Characteristics	Provide all the technical information about the devices installed on site	JSON	{ "value":3, "deviceMeasurementId": "23bb6e98-8429-4162-8236-cc4f231bb2a4", "deviceId": "1236bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "property": "Blade Length" }	**Range** – depending on the property	• *Field Data Aggregation* • *Decentralized Multi-Purpose Repository*
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 value		[{ "timestamp": 1537803600000, "value": 905.5, "deviceMeasurementId": "18bb6e98-8429-4162-8236-cc4f231bb2a4", "deviceId": "4836bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "granularity": "DAYAHEAD", "property": "Power Consumption" },...]	**Range** – depending on the forecasted property **Frequency**: every 30 min, 1 hour	· *Virtual Power Plants Generation Modelling and Forecasting* • *Load Profiling and Disaggregation* • *PV/RES Degradation & Trend Analysis* • *VPP & DR Services Optimization Engine*

Software Requirements/Development Language	*nVIDIA CUDA, Keras, TensorFlow / Python3, Java,* Cassandra, MySql, RabbitMQ
Hardware Requirements	*NVIDIA GeForce GTX 10XX/ …* *LINUX Server*
Communications	REST API , HTTP
Status of the development of the component	“partially developed”

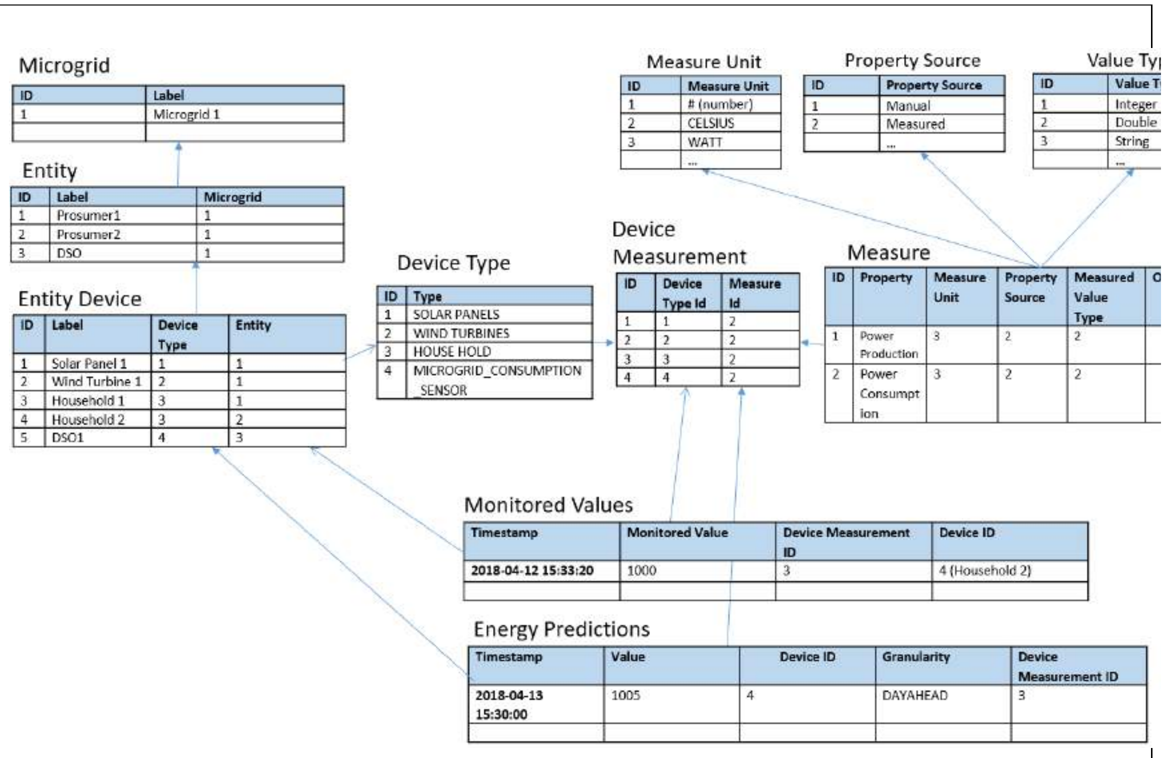
High-level Class Diagram	

Table 25 Virtual Power Plants Generation Modeling & Forecasting

Name of New Component/Service:	***Virtual Power Plants Generation Modeling & Forecasting***
Type:	*Component*
Functionality:	*Provide and evaluate energy co-generation optimization algorithms regarding the creation of optimal coalitions, Virtual Power Plants, with the aim of providing more reliable aggregated power sources*

***Input Connections & Interfaces:* From which components it receives input**	• *Electricity Production/Consumption Forecasting* • *Multi-Building DR characterization through thermal, optical and LIDAR information fusion* • *PV/RES Degradation & Trend Analysis* • *Field Data Aggregation*
***Output Connections & Interfaces:* To which components it sends the results**	• *VPP and active micro-grid flexibility* • *VPP & DR Services Optimization Engine* • *Secure Data Handling Through Ledger* **REST API**: <host>:<port>/edream-vpp/optimize
Functional Requirements	*MF02_BR03_UR01_FR01, MF02_BR03_UR02_FR02* *MF02_BR03_UR03_FR03*
Non-Functional Requirements	-

Input Parameters

Attribute/ Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Received From***
Production /Consumption forecasts for each participati	An array containing the device and the measurement the forecast is referring to,	JSON	[{ "timestamp": 1537803600000, "value": 905.5, "deviceMeasurementId": "18bb6e98-8429-4162-8236-cc4f231bb2a4", "deviceId": "4836bee7-	**Range** – depending on the forecasted property **Frequency**: every 30 min, 1 hour	• *Electricity Production/Consumption Forecasting – Prosumer Level/Grid Level*

ng producer/grid	together with the forecasted value		bc42-48a9-9e4e-aa1ecc68e6d8", "granularity": "DAYAHEAD", "property": "Power Consumption" },...]		
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*
DR specifications for Buildings	Information about the DR potential of the buildings	JSON	{ "value": 2000, "deviceMeasurementId": "14bb6e98-8429-4162-8236-cc4f231bb2a4", "deviceId": "4836bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "property": "DR Potential" }	**Range** – **Frequency**:	• *Multi-Building DR characterization through thermal, optical and LIDAR information fusion*
Requested Goal	Choose the goal to be followed by the prosumer oprimization process		{ "goal": "RESERVE", "signal" : [{ "timestamp": 1537341600000, "value": 2000}, { "timestamp": 1537341600000, "value": 2000 },...], "typeOfGeneration": [{ "value": 0.30, "deviceTypeId": "43bb6e98-8429-4162-8236-cc4f231bb2a4",}, {	**Range** – **Frequency**: on request	• *HDMI*

			"value": 0.20, "deviceTypeId": "41ba6e98-8429-4162-8236-cc4f231bb2a4"}, ...]}		
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	{ "prosumerIds" : ["2136bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "3436bee7-bc42-48a9-9e4e-aa1ecc68e6d8", ...] }	**Range** – **Frequency**: on request	• *VPP and active micro-grid flexibility* • *VPP & DR Services Optimization Engine* • *Secure Data Handling Through Ledger*

Software Requirements/Development Language	JAVA
Hardware Requirements	*LINUX Server*
Communications	REST API
Status of the development of the component	“partially developed”
High-level Class Diagram	-

Table 26 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*
Input Connections & Interfaces: From which components it receives input	• *Field Data aggregation* • *Secured Data Handling Through ledger* • *Electricity Production/Consumption Forecasting* • *Closed Loop DR Verification Engine* • *Baseline Flexibility Estimation*
Output Connections & Interfaces: To which components it sends the results	• *Closed Loop DR Verification Engine*
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	-
Input Parameters	

Attribute /Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Received From***
Monitored 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	• *Field Data aggregation*
Registered Energy Assets	The energy asset registered as token in the chain	TOKEN	-	**Range** – **Frequency**:	• *Secured Data Handling Through ledger*
Baseline & Flexibility 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*
Validated energy flexibility transactions	Validation of the actual flexibility delivered according to the settlement	-	-	**Range** – **Frequency**:	• *Closed Loop DR Verification Engine*

Predicted Values	An array containing the device and the measurement the forecast is referring to, together with the forecasted value	JSON	[{ "timestamp": 1537803600000, "value": 905.5, "deviceMeasurementId": "18bb6e98-8429-4162-8236-cc4f231bb2a4", "deviceId": "4836bee7-bc42-48a9-9e4e-aa1ecc68e6d8", "granularity": "DAYAHEAD", "property": "Power Consumption" },...]	**Range** – depending on the forecasted property **Frequency**: every 30 min, 1 hour	• *Electricity Production/Consumption Forecasting*
Output Parameters					
Attribute /Parameter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Sent To***
Smart Contracts evaluation of actual consumption/production delivered	Evaluate the energy flexibility delivered and penalized according to the deviations	-	Evaluation inside the contract. The result is an updated account	-	• *Closed Loop DR Verification Engine*

Software Requirements/Development Language	Ethereum, Solidity, NodeJS
Hardware Requirements	*NVIDIA GeForce GTX 10XX/ …* *LINUX Server*
Communications	REST API, JSON RPC

Status of the development of the component	to be developed from scratch
High-level Class Diagram	-

Table 27 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.*
Input Connections & Interfaces: From which components it receives input	• *Secured Data Handling Through ledger* • *Electricity Production/Consumption Forecasting* • *Closed Loop DR Verification Engine* • *HMI*
Output Connections & Interfaces: To which components it sends the results	• *Closed Loop DR Verification Engine* **REST API**: <host>:<port>/edream-market-api/register-order/…
Functional Requirements	MF03_BR02_UR01_FR01, MF03_BR02_UR02_FR02 MF03_BR02_UR03_FR03, MF03_BR02_UR04_FR04 MF03_BR03_UR01_FR05, MF03_BR03_UR02_FR06 MF03_BR03_UR03_FR07, MF03_BR03_UR04_FR08

Non-Functional Requirements			MF03_BR04_UR01_NFR01		
			<u>Input Parameters</u>		
Attribute /Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Received From***
Registered Energy Assets	The energy asset registered as token in the chain	TOKEN	-	**Range** – **Frequency**:	*Secure Data handling through ledger*

Bid/Offer Actions	Place orders on the market contract	JSON	{
			 "timestamp": 1537341600000,
			 "addressToBeCredited": "0x11fc0bca5134a5cd7469c2fe1f9968f44d1949a8",
			 "addressBeneficiary": "0x11fc0bca5134a5cd7469c2fe1f9968f44d1949a8",
			 "timelimit": 1537341600000,
			 "quantity": 132000,
			 "price": 12,
			 "orderSide": "BUY",
			 "orderType": "MARKET",
			 "orderAvailability": "OPEN",
			 }
				Range:
				 Frequency: 30-60 min
					HMI
					
					 Electricity Production/ Consumption Forecasting

Validated Energy Transactions	Validation of the actual energy delivered according to the settlement	-	-	**Range** – **Frequency**:	*Closed Loop DR Verification Engine*
			<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*

Matched energy production and demand	The matching between the sell and buy orders	JSON	{
			"marketAddress": "0x32fc0bca5134a5cd7469c2fe1f9968f44d1949a8",
			"trades" : [
			{
			"id": 2,
			"buyOrderId": 2,
			"sellOrderId" : 3,
			"symbolQuantity": 132000,
			"price": 12,
			"symbol": "ENERGY"
			"symbolAddress": "0x34fc2bca5134a5cd7469c2fe1f9968f44d1949a8",
			"symbolAccountToBeCredited": "0x11fc0bca5134a5cd7469c2fe1f9968f44d1949a8",
			"ethAccountBeneficiary" : "0x12fc0bca5134a5cd7469c2fe1f9968f44d1949a8",
			"ethAccountToBeCredited": "0x11fc0bca5134a5cd7469c2fe1f9968f44d1949a8",
			"symbolAccountBeneficiary" : "0x12fc0bca5134a5cd7469c2fe1f9968f44d1949a8",
			"timestamp": 1537341600000,
			"timelimit": 1537341600000,
			}, ...]
			}
				Range – Frequency:	• *Closed Loop DR Verification Engine*

Software Requirements/Development Language	Ethereum, Solidity, NodeJS

Hardware Requirements	*NVIDIA GeForce GTX 10XX/ …* *LINUX Server*
Communications	REST API, JSON RPC
Status of the development of the component	Partially developed
High-level Class Diagram	-

Table 28 Secure data handling through ledger

Name of New Component/Service:	***Secure data handling through ledger***
Type:	*Component*
Functionality:	*Blockchain distributed ledger, used as secure storage and enabling the execution of smart contracts*
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/Parameter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Received From***
address	the identifier associated to each of the distributed ledger participant	String	Alphanumeric	About 10 minutes	field data aggregation
timestamp	Timestamp of the reading	Number	uint		
value	Value read from the meter	Number	int		
Output Parameters					
Attribute/Parameter	***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*		

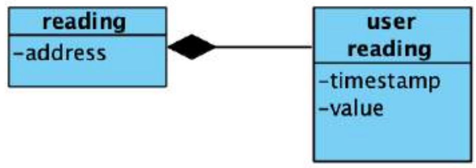
Communications	*http; rpc;*
Status of the development of the component	*To be developed*
High-level Class Diagram	

Table 29 Closed loop DR Verification Engine

Name of New Component/Service:	***Closed loop DR Verification Engine (Financial Settlement through eDREAM ledger + DR Verification and Finacial Settlement)***
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.*
***Input Connections & Interfaces:* From which components it receives input**	• *eDREAM Ledger* • *Smart meters – Field Data Aggregation* • *Secured Data Handling through Ledger* *Rest API*

Output Connections & Interfaces: ***To which components it sends the results***	• *End-user dashboard* • *Blockchain driven control for LV networks* *Rest API*
Functional Requirements	MF03_BR02_UR01_FR01, MF03_BR02_UR02_FR02 MF03_BR02_UR03_FR03
Non-Functional Requirements	-

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	Electricity consumption/production forecast
request	Requested value at target timestamp	number	int		Blockchain-driven control for LV networks (flexibility management)
reading	measured value at target timestamp	number	int		Field data aggregation
range	Tolerance range for the request	number	uint	Fixed for the contract duration	Defined in the smart contract
multiplier	incentive/penalty multiplier	number	float		

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	Blockchain-driven control for LV networks (flexibility management)

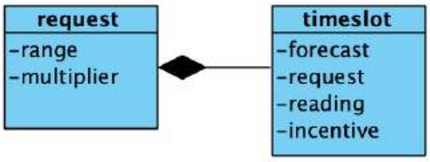
Software Requirements/Development Language	*Solidity (for the smart contract development)*
Hardware Requirements	*the component runs on a blockchain infrastructure*
Communications	*http; rpc*
Status of the development of the component	*to be developed*
High-level Class Diagram	

Table 30 Big Data Analytics Engine Tool

Name of New Component/Service:	***Big Data Analytics Engine Tool***
Type:	*Platform*

Functionality:	*A set of orchestrated functionalities to provide data lifecycle management for energy domain advanced services*
***Input Connections & Interfaces:* From which components it receives input**	• *Field Data Aggregation* • *Load Profiling and Disaggregation* • *Electricity Consumption / Production Forecasting* • *Multi-building DR characterization* • *Baseline Flexibility Estimation*
***Output Connections & Interfaces:* To which components it sends the results**	• *Big Data Clustering at Multiple Scales* • *Load Profiling and Disaggregation* • *HMI*
Functional Requirements	*MF02_BR01_FR01, MF02_BR01_FR02* *MF02_BR01_FR03, MF02_BR01_FR04* *MF02_BR01_FR05, MF02_BR01_FR06* *MF02_BR01_FR07*
Non-Functional Requirements	*MF02_BR01_NFR01, MF02_BR01_NFR02*

Input Parameters

Attribute/Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Received From***
Smartmeter_id	*Unique identifier for the smartmeters*	*STRING*	*CSV*		*ASM*
Energy Absorbed	*Energy absorbed by the smartmeter*	*FLOAT*	*CSV*	*E. Abs >= 0* *Freq = 15 min*	*ASM*

Energy Injected	*Energy injerted to the grid by the Smartmeter*	*FLOAT*	*CSV*	*Energy Injected >= 0 Freq = 15 min*	*ASM*
Date	*Date:* *DD-MM-YYYY*	*STRING*	*CSV*		*ASM*
Output Parameters					
Attribute/Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Sent To***
(attributes to be defined according to Use cases)					
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	*On early development stage (5%-10%)*				

Table 31 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 (core component embedded in Big Data Analytics Engine tool)*
***Input Connections & Interfaces:* From which components it receives input**	• *Field Data Aggregation* • *Load Profiling and Disaggregation* • *Electricity Consumption / Production Forecasting* • *Multi-building DR characterization? (please check comment in email)* • *Baseline Flexibility Estimation*
***Output Connections & Interfaces:* To which components it sends the results**	• *Customers Segmentation component* • *HMI* • *VPP and Active microgrid Flexibility Profiling*
Functional Requirements	*MF02_BR01_FR01, MF02_BR01_FR02* *MF02_BR01_FR03, MF02_BR01_FR04* *MF02_BR01_FR05*
Non-Functional Requirements	*MF02_BR01_UR01_NFR01, MF02_BR01_UR01_NFR02* *MF02_BR01_UR03_NFR03, MF02_BR01_UR04_NFR04* *MF02_BR01_UR05_NFR05*

Input Parameters

Attribute/Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Received From***
Prosumers Profiles	*Indicating prosumers load consumption behavior*	*Float*	*csv*	*E. Abs >= 0 Freq = 15 min*	*Load Profiling & Disaggregation*
KPIs	*DR related*	*String*	*csv*	*-*	*Decentralized Repository*

	indiators				
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*	• Decentralized Repository • Customers Segmentation
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	*On early stage development (5%-10%)*				

Table 32 Load Profiling & Disaggregation

Name of New Component/Service:	***Load Profiling and Disaggregation***
Type:	*Component*
Functionality:	*A non-intrusive appliance load analysis technique with the goal of disaggregating large composite loads*

***Input Connections & Interfaces:** From which components it receives input*	• *Big Data Analytics Engine* • *Field Data Aggregation* • *Electricity Consumption/Generation Forecast*
***Output Connections & Interfaces:** To which components it sends the results*	• *Big Data Clustering at Multiple Scales component* • *VPP and Active Microgrid Flexibility Profiling Component*
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*	*Decentralized Repository*
Historical data	*Historical data for load consumption*	*Float*	*csv*	*E. Abs >= 0 Freq = 15 min*	*Decentralized Repository*
Real-time data	*Data from field devices*	*Float*	*csv*	*E. Abs >= 0 Freq = 15 min*	*Field Devices*
KPIs	*Defines indicators*	*String*	*csv*	*-*	*Decentralized Repository*

Output Parameters

Attribute/Para-meter	*Short Description*	*Data Type*	*Data Format*	*Value Range & Frequency*	*Data Sent To*
Prosumers profiles	*Pattern related to specific features*	*Float*	*csv*	*E. Abs >= 0 Freq = 15 min*	*Big Data Clustering at Multiple Scales*
Device Estimated Consumption	*Individual load consumption per device*	*Float*	*csv*	*E. Abs >= 0 Freq = 15 min*	*Decentralized Repository*

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	*On early stage development (5%-10%)*

Name of New Component/Service:	***Customers Segmentation***
Type:	*Component*
Functionality:	*Big data tool for clusterization of energy customers*
***Input Connections & Interfaces:* From which components it receives input**	• *Big Data Clustering at multiple scales component* • *Decentralized Repository (behavioral KPIs)*
***Output Connections & Interfaces:* To which**	• *VPP and Active Microgrid Flexibility Profiling component*

components it sends the results					
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/Para-meter	***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*	*-*	*Decentralized Repository*
Output Parameters					
Attribute/Para-meter	***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*	*Decentralized Repository*
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	*On early stage development (5%-10%)*

Table 33 PV/RES Degradation and Trend Analysis

Name of New Component/Service:	***PV/RES Degradation and Trend Analysis***
Type:	*Component*
Functionality:	*Improve the short-term forecasting of generation (e.g. support day-ahead, direct trading, coupon-based DR programs etc.) based on PV degradation and trend analysis and forecasting*
***Input Connections & Interfaces:* From which components it receives input**	• *Field Data Aggregation* • *Weather Forecast APIs* • *Decentralized Repository (Renewable Device Related Characteristics, Historical data for past trading activity etc.)* • *Electricity consumption/production forecasting*
***Output Connections & Interfaces:* To which components it sends the results**	• *Virtual Power Plants Generation Modelling and Forecasting*
Functional Requirements	*MF01_BR04_UR01_FR01, MF01_BR04_UR02_FR02,*

			MF01_BR04_FR03, MF01_BR04_FR04		
Non-Functional Requirements			-		
Input Parameters					
Attribute/Parameter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Received From***
Weather information	*Weather information such as solar irradiation, cloudiness etc*	*Timestamp (string),Weather attribute (string, value (float))*	*Time series format, location (latitude, longitude) coordinates, Solar irradiation (W/m2), cloudiness (%),* *date and time (yy-mm-dd-hh-mm-ss)* *CSV, XML or JSON*	*15 min*	*Online Weather provider/KRW*
Age of pv panel	*Age of PV panel in month/year*		*Float number*	*Greater than 0 and freq is 15 min*	*Solar PV model characteristics from datasheet*
Output Parameters					
Attribute/Parameter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Sent To***
Solar PV generation		*Floating point number*	*ID-timstamp-*	*>=0*	

		W/m2	*value*	*15 min*	

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	*Partially developed. Solar PV model is already developed. Trend analysis algorithm needs to be developed from scratch.*

Table 34 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*
<u>Input Connections & Interfaces:</u> From which components it receives input	• *Field Data Aggregation* • *Decentralized Multi-purpose Repository* • *Multi-Building DR Characterization* • *JSON/XML/CSV based information*

	exchange
<u>Output Connections & Interfaces:</u> To which components it sends the results	• *VPP and Active Micro-grid Flexibility Profiling* • *VPP & DR Services Optimization Engine* • *Blockchain-driven control for LV networks*
Functional Requirements	*MF01_BR03_UR01_FR01, MF01_BR03_FR02* *MF01_BR03_FR03, MF02_BR09_UR01_FR04* *MF02_BR09_UR02_FR05*
Non-Functional Requirements	-

<u>Input Parameters</u>

Attribute/Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Received From***
Prosumer baseline profiles	*A profile for prosumer baseline flexibity estimation*	*Prosumer ID (int), DR Programme ID (int) starttimstamp (string), endtimestamp (string), baseline energy flexibility (float), monetary cost (float), comfort cost (float)*	*CSV, JSON or XML*	*Non-negative, granlurity is 15 minutes*	• *Field Data Aggregation* • *Decentralized Repository* • *Multi-Building DR Characterization*
Constraints of the installed devices					• *Decentralized Repository*

Output Parameters					
Attribute/Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Sent To***
Array of estimated energy consumption 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 end time (float[])*	*CSV, JSON or XML*	*Greater than 0, every 15 minutes*	• *VPP and Active Micro-grid Flexibility Profiling* • *VPP & DR Services Optimization Engine* • *Blockchain-driven control for LV networks*
Array of estimated energy production flexibility values		*Prosumer ID (int)-DR programme ID (int)-Starttimestap (string)-endtimestamp (string)-Energy consumption flexibility array (float[])-monetary cost*	*Prosumer ID-DR programme ID-Starttimestap-endtimestamp -Energy production flexibility array-monetary cost array during starttime and*	*Greater than 0, every 15 minutes*	• *VPP and Active Micro-grid Flexibility Profiling* • *VPP & DR Services Optimization Engine* • *Blockchain-driven control for LV networks*

		during starttime and endtime (float []), comfort cost during starttime and end time (float[])	*endtime, comfort cost array during starttime and end time*		

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	*To be developed from scratch*

Table 35 VPP and active Microgrid Flexibility Profiling

Name of New Component/Service:	***VPP and active Microgrid Flexibility Profiling***
Type:	*Component*
Functionality:	*Considering the flexibility of the prosumers forming the VPP, the aim is to compute and optimize the flexibility of the micro-grid by following the requested profiles*
***Input Connections & Interfaces:* From which components it receives input**	• *Virtual Power Plant Generation, Modeling and Forecasting* • *Load Profiling and Disaggregation* • *VPP Customers' Segmentation*

Output Connections & Interfaces: ***To which components it sends the results***	• *Blockchain-driven control of LV Networks*
Functional Requirements	*MF02_BR02_UR08_FR01, MF02_BR02_UR02_FR02* *MF02_BR02_UR08_FR03, MF02_BR02_UR04_FR04* *MF02_BR02_UR05_FR05, MF02_BR02_FR06*
Non-Functional Requirements	-

Input Parameters

Attribute/Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Received From***
Micro-grid baseline		*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 end time (float[])*	*CSV, XML or JSON*	*Greater than 0, every 15 min*	

The flexibility of each prosumer		*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 end time (float[])*	*CSV, XML or JSON*	*Greater than 0, every 15 min*	
Output Parameters					
Attribute/Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Sent To***
Array of estimated flexibility values		*Prosumer ID (int)-DR programme ID (int)-Starttimestap (string)-endtimestamp (string)-Energy consumption flexibility array (float[])-monetary cost*	*CSV, XML or JSON*	*Greater than 0, every 15 min*	*Blockchain-driven control of LV Networks*

		during starttime and endtime (float []), comfort cost during starttime and 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	-

Table 36 VPP & DR Services Optimization Engine

Name of New Component/Service:	***VPP & 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)*
<u>Input Connections & Interfaces:</u> From which components it receives input	• *Virtual Power Plants Generation, Modelling & Forecasting* • *Baseline Flexibility Estimation* • *Electricity consumption/Production*

	Forecasting • Decentralized repository
***Output Connections & Interfaces:* To which components it sends the results**	• VPP and active Micro-grid Flexibility Profiling • Techniques for DR and Energy Flexibility Assessment • Decentralized Network Control Optimization & DR Verification
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	-

Input Parameters

Attribute/Para-meter	***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*	*XML,JSON or CSV*	*Greater than 0, every 15 min*	*Baseline Flexibility Estimation*

		flexibility array (float[])- monetary cost during starttime and endtime (float []), comfort cost during starttime and 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*	*Decentralized Repository*
Output Parameters					
Attribute/Parameter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Sent To***
Optimal DR scheduling information		*Device ID (int) – DR programme ID (int) –start time (string) – end time (string) – array of values (flaot[])*	*XML,JSON or CSV*	*Greater than 0, every 15 min*	
Set of feasible		*Device ID (int)*	*XML,JSON*	*Greater*	

solutions for stakeholders and proposed DR scheduling		*– DR programme ID (int) –start time (string) – end time (string) – array of values (flaot[])*	*or CSV*	*than 0, every 15 min*	
Improved forecasting data with trend analysis algorithms		*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*	
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			*To be developed from scratch*		

Table 37 Decision Support System & DR Strategies Optimization (Front-End)

Name of New Component/Service:	***Decision Support System & DR Strategies Optimization (Front-End)***

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.*		
Input Connections & Interfaces: ***From which components it receives input***			• *VPP & DR Services Optimization Engine* • *Electricity Consumption/Production Forecasting* • *Blockchain-driven control for LV networks*		
Output Connections & Interfaces: ***To which components it sends the results***			• *VPP & DR Services Optimization Engine* • *End-user web interface*		
Functional Requirements			*MF02_BR04_UR01_FR01, MF02_BR04_FR02* *MF02_BR04_FR03, MF02_BR04_FR04*		
Non-Functional Requirements			*-*		
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*	
DR profiles					
Output Parameters					
Attribute/Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Sent To***

	Optimzed DR profile	*Starttime (string) – end time (string)- values of optimized DR profile (float)*	*XML,CSV,JSON*	*Greater than 0, 15 min*	

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	*To be developed from scratch*

Table 38 Multi-building DR characterization through thermal, optical and LIDAR information fusion

Name of New Component/Service:	***Multi-building DR characterization through thermal, optical and LIDAR information fusion***
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 performance of the buildings through optical, thermal and LIDAR images*
***Input Connections & Interfaces:* From which components it receives input**	• *DR Aerial Survey Toolkit* • *Drone camera mount.* • *LiDAR Ethernet port.*

	• *Skyport API.*
<u>Output Connections & Interfaces:</u> To which components it sends the results	• *Virtual Power Plants Generation Modeling & Forecasting* • *Big Data Clustering at multiple scales* • *Baseline flexibility estimation*
Functional Requirements	*MF01_BR01_UR01_FR01, MF01_BR01_UR01_FR02* *MF01_BR01_FR03, MF01_BR01_FR04* *MF01_BR01_FR05, MF01_BR01_FR06* *MF01_BR01_FR07*
Non-Functional Requirements	-

<u>Input Parameters</u>					
Attribute/Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Received From***
Aggregated and processed data from aerial surveys					• *DR Aerial Survey Toolkit* • *Cameras*
<u>Output Parameters</u>					
Attribute/Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Sent To***
Estimated DR potential for groups of assets or buildings					• *Virtual Power Plants Generation Modeling & Forecasting* • *Big data Clustering at Multiple Scales* • *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	Partially developed

Table 39 DR Aerial Survey Toolkit (Front-End)

Name of New Component/Service:	***DR Aerial Survey Toolkit (Front-End)***
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*
***Input Connections & Interfaces:* From which components it receives input**	Drone camera mount. LiDAR Ethernet port. Skyport API.
***Output Connections & Interfaces:* To which components it sends the results**	3D Workstation. Pix4D software, VeloView Software.
Functional Requirements	*MF01_BR01_FR01, MF01_BR01_FR02*

			MF01_BR01_FR03, MF01_BR01_FR04		
Non-Functional Requirements			-		
Input Parameters					
Attribute/Para-meter	***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)					
Weather Data	*Data from weather station.*	*Luminosity, temperature, humidity, rain, wind speed*	*.csv*	*15min frequency*	*Weather station*
Output Parameters					
Attribute/Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Sent To***
3d model	*3d model made from pics taken during aerial*	*geometry*	*.obj*	*n/a*	*3d workstation system*

	survey.				
Output from image processing (CERTH)	*Building thermal signature*	*Units of thermal leakage levels with respect to baseline*	*.csv*	*n/a, once per scan*	*Decision Support System, Graph analytics platform*

Software Requirements/Development Language	Velodyne VeloView LiDAR processing software. Pix4D aerial survey software
Hardware Requirements	Quadcopter Drone. We are using a DJI Matrice 210. Thermal imaging camera: DJI Zenmuse XT2 with 19mm lens, 30Hz refresh rate and 640x512 pixels resolution. LiDAR: Velodyne VLP 16 ('Puck')
Communications	TCP/IP for Wireless Communications
Status of the development of the component	Partially developed.

Table 40 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.*
***Input Connections & Interfaces:* From which components it receives input**	• *Field Data Aggregation* • *Decentralized Repository* • *HMI*

Output Connections & Interfaces: ***To which components it sends the results***	• *Closed Loop DR Verification Engine* • *HMI* • *Decentralized Repository*			
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	-			
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*	-	*Field Data Aggregation*
Parameters for visualization (e.g. time resolution)	*Float*	*XML, JSON*	-	*Decentralized Repository*
Data models related to segmentation metrics - KPIs (e.g. profits/loses, congestion improvement achieved etc.)	*String*	*XML, JSON*	-	*Decentralized Repository*
DR activation signals	*String*	*XML, JSON*	-	*HMI*
Input settings/parameters by the HMI	*Float*	*XML, JSON*	-	*HMI*
Parameters of DR signals and mapping to the portfolio	*Float*	*XML, JSON*	-	*HMI*
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*		• *HMI* • *Decentralized Repository*
Improved DR activation signals	*Float*	*XML, JSON*		*HMI*

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	*To be adjusted and evolved*

Table 41 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.*

<u>Input Connections & Interfaces:</u> From which components it receives input	• *Core platform's components* • *End-user web interface*
<u>Output Connections & Interfaces:</u> 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*

<u>Input Parameters</u>					
Attribute/Para-meter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Received From***
All the data models exchanged within the core platform and with the end-user web interface.					

<u>Output Parameters</u>					
Attribute/Para-	***Short***	***Data***	***Data Format***	***Value Range***	***Data Sent To***

<table>
<tr><th>meter</th><th>Description</th><th>Type</th><th></th><th>& Frequency</th><th></th></tr>
<tr><td colspan="6">All the data models exchanged within the core platform and with the end-user web interface.</td></tr>
<tr><td colspan="3">Software Requirements/Development Language</td><td colspan="3">The software requirement is C++ libraries and runtime environment. C++ will be used as programming language.</td></tr>
<tr><td colspan="3">Hardware Requirements</td><td colspan="3">A Windows/Linux based PC with administrator right and credentials.

Web Server</td></tr>
<tr><td colspan="3">Communications</td><td colspan="3">Ethernet or WiFi Connectivity

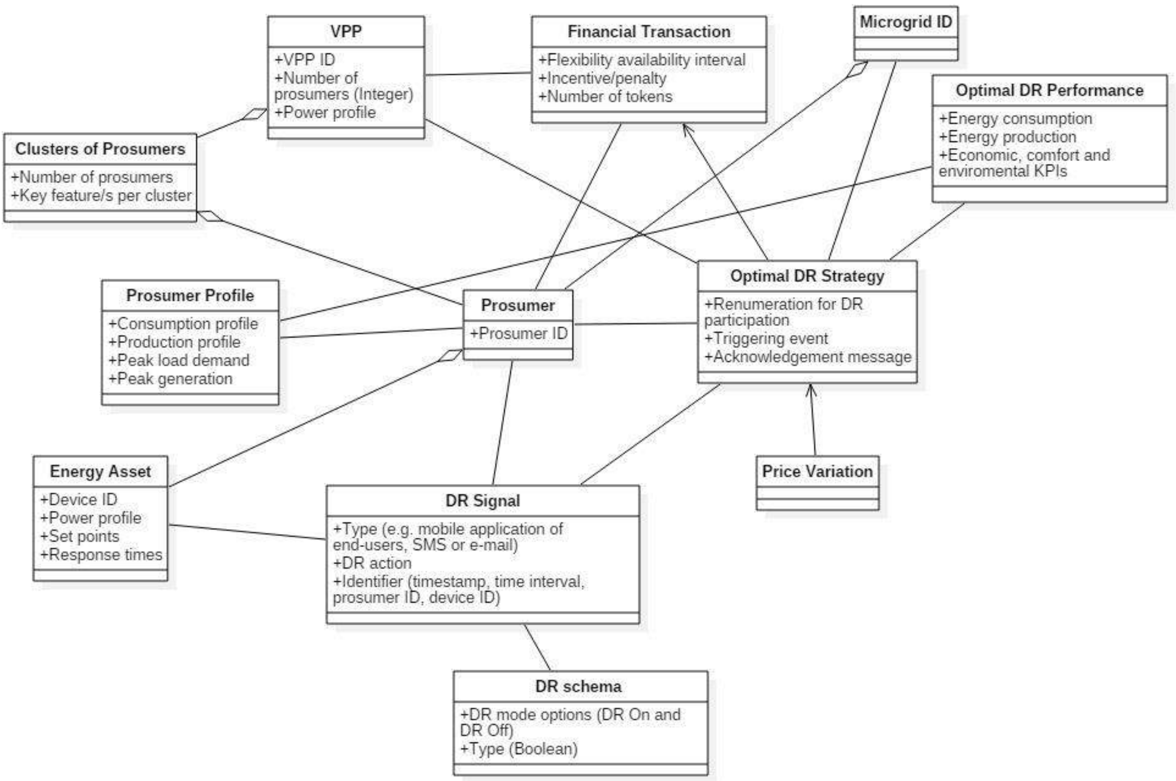
Web Interface</td></tr>
<tr><td colspan="3">Status of the development of the component</td><td colspan="3">To be adjusted and evolved</td></tr>
<tr><td colspan="6">High-level Class Diagram</td></tr>
<tr><td colspan="6">
</td></tr>
</table>

Table 42 Forecasting Tool (Front-End)

Name of New Component/Service:	***Forecasting Tool (Front-End)***				
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.*				
Input Connections & Interfaces: From which components it receives input	• *Baseline Flexibility Estimation* • *Electricity consumption/Production Forecasting*				
Output Connections & Interfaces: To which components it sends the results	• *End-User Web Interface*				
Functional Requirements	*MF01_BR02_FR01, MF01_BR02_FR02* *MF01_BR02_FR03, MF01_BR02_FR04*				
Non-Functional Requirements	-				
Input Parameters					
Attribute/Parameter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Received From***
Forecasted consumption/production prosumers data	*Near real time forecasted values for intraday and day-ahead planning*	*Float*	*XML,JSON*	*Every 10 min (5min or 1 hour are also possible)*	*Electricity consumption/Production Forecasting*

Baseline load flexibility	*Baseline load fluctuations for enabling DR program application*	*Float*	*XML,JSON*	*-*	*Baseline Flexibility Estimation*
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			*To be developed from scratch*		

9 Conclusion

This report has presented the architectural design of the eDREAM system along with the respective functional and technical specifications. Initially, the methodology followed towards the detailed definition of the architectural elements that compose the eDREAM system architecture was described. Then, the eDREAM conceptual architecture, as well as the structural, development, deployment and dynamic view of the system were analysed.

The architectural views and perspectives presented in this deliverable will further drive the design and implementations during the project lifetime. The UML diagrams illustrate the core components of the eDREAM system, the key actors as well as the way they interact with the system. Moreover, the detailed description of the architectural elements provide a comprehensive view of the eDREAM components focusing mainly on its major architectural elements and providing high-level use-case and sequence diagrams, which allow to reason about and describe the dynamic behaviour of the system. The analysis presented for each architectural element, which encompasses data inputs and outputs, interrelation among system entities and correlation with respective use cases and key system requirements will enable during the implementation phase component developers and integrators to communicate about architectural issues in a more efficient and effective way.

Although no major modifications are expected to the overall eDREAM system architecture and its major components, this report can be considered as a living document that will address minor refinements that might be necessary in case of new unforeseen limitations that will come up during the implementation phase. It is worth mentioning that in the definition of the architectural elements process, all major component developer responsible partners were involved. The involvement of the developers to the architecture refinement process was significant because it resulted in a more coherent architecture definition (and its architectural elements), which also encapsulated the view of developers.

Summarizing, this deliverable provides a sufficient basis for the technical developments of the eDREAM system that will take place in WPs 3-6, whereas the actual architectural elements of each framework will be implemented and validated (WP7).

References

[1]. Agile Modeling, UML2 Use Case Diagrams, available under http://www.agilemodeling.com/artifacts/useCaseDiagram.htm (state on September 2013).

[2]. Carnegie Mellon community, UML Use Case Diagrams: Tips and FAQ, available under http://www.andrew.cmu.edu/course/90-754/umlucdfaq.html (state on September 2013).

[3]. StarUML ,Modelling with Class Diagram, available under http://staruml.sourceforge.net/docs/user-guide(en)/ch05_2.htm

Annex I: Functional & Non-Functional Requirements

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

Functional Requirements

Table 43 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 energy consumption / production patterns detection
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 energy consumption / production patterns detection
MF02_BR07_UR02_FR07	Grid energy consumption / production forecast for specific time interval
MF02_BR07_UR03_FR08	Store grid consumption / production forecast view

Table 44 PV/RES Degradation and 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 45 Virtual Power Plants Generation Modelling and 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 Inventory of field devices

Table 46 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 47 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

Table 48 Load Profiling and Disaggregation – FRs

Component: Load Profiling & Disaggregation	
Functional Requirement ID	***Description***
MF02_BR02_UR02_FR01	Receive historical data for prosumers' energy consumption
MF02_BR02_FR02	Receive historical weather data (e.g. temperature)
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 Recentralized Repository

Table 49 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	Obtain load profiles and generation profiles patterns
MF02_BR01_FR04	Receive historical data concerning prosumers' behavior and responsiveness to DR schemes
MF02_BR01_FR05	Store produced clusters of prosumers

Table 50 Customers 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 Decentralized Repository
MF02_BR02_FR04	Detect patterns according to the received attributes values
MF02_BR02_FR05	Store segments of prosumers

Table 51 VPP and active Micro-grid flexibility profiling – FRs

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

Table 52 VPP & DR Services Optimization Engine – FRs

Component: VPP & 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 Graph-based Analytics

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 54 Secure data handling through ledger – FRs

Component: Secure data handling through 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 55 Blockchain-driven control for LV networks (flexibility management) – 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 56 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 57 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 58 HMIs – 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 59 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 60 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 61 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 62 Big Data Analytics Engine Tool – FRs

Component: Big Data Analytics Engine Tool	
Functional Requirement ID	***Description***
MF02_BR01_FR01	Near real-time data processing
MF02_BR01_FR02	Batch Processing
MF02_BR01_FR03	Data Harmonization
MF02_BR01_FR04	Data Preprocessing & Postprocessing
MF02_BR01_FR05	Data Gathering
MF02_BR01_FR06	Data Discovery
MF02_BR01_FR07	Data Filtering

Table 63 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 64 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 65 Secure data handling through ledger – NFRs

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

Table 66 Secured Blockchain-driven Energy Market – NFRs

Component: Secured Blockchain-driven Energy Market	
Non-Functional Requirement ID	***Description***
MF03_BR04_UR01_NFR01	Store transactions in a secured and tamperproof manner

Table 67 Big Data Clustering at Multiple Scales – NFRs

Component: Big Data Clustering at Multiple Scales	
Non-Functional Requirement ID	***Description***
MF02_BR01_UR01_NFR01	Be able to operate in a scalable way horizontally or vertically depending on the demands related to data injection, processing and storage
MF02_BR01_UR01_NFR02	Be able to decompose the solution in different microservices, so as to run better during different processes in several machines
MF02_BR01_UR03_NFR03	Be able to document information types produced, consumed and transformed in an information model which shall include the relationships between information types
MF02_BR01_UR04_NFR04	Be able to process data gathered from different sources in order to achieve flexibility profiling. It is crucial for such calculation to ensure the capacity to provide data coming from differed database and data lake (batch, preprocessed, other modules outputs, devices etc.).
MF02_BR01_UR05_NFR05	Information models that govern the data exchanged with the different types of devices and managed or stored by the modules will consider context data or metadata (e.g. location, accuracy, ownership etc.)

Table 68 Load Profiling & Disaggregation & Customer Segmentation – FRs

Component: Load Profiling & Disaggregation & Customer Segmentation

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

Table 69 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 70 Big Data Analytics Engine Tools – NFRs

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

Table 71 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

Annex II: Architectural Specifications Templates

Table 72 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 73 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.>
***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/Parameter	***Short Description***	***Data Type***	***Data Format***	***Value Range & Frequency***	***Data Received From***
<please mention the input parameters. Each row corresponds to one parameter>	*<mention a short description of the input parameter if necessary>*	*<please mention the data type of this parameter (e.g. (e.g. int, string, etc. or complex type, e.g. list, object, etc.)*	*<e.g. XML, JSON etc.>*	*<indicate measurement unit and range of values for this attribute/parameter and frequency-sample rate>*	*<please mention the source component or module that provides input data to this parameter>*
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 ^{o}C)>*
Measurement Resolution	*<level of measurement (e.g. to 0.01^{o}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>*