The DELTA project has received funding from the EU's Horizon 2020 research and innovation programme under grant agreement No 773960

DELTA

Project Acronym:	**DELTA**
Project Full Title:	**Future tamper-proof Demand rEsponse framework through seLf-configured, self-opTimized and collAborative virtual distributed energy nodes**
Grant Agreement:	**773960**
Project Duration:	**36 months (01/05/2018 – 30/04/2021)**

DELIVERABLE D1.6
DELTA Overall Framework Architecture v2

Work Package	**WP1 – DELTA Requirements & System Architecture Definition**
Task	**T1.2 – Architectural Design, Functional & Technical Specification**
Document Status:	**Final**
File Name:	**DELTA_D1.6_Final**
Due Date:	**30.04.2020**
Submission Date:	**May 2020**
Lead Beneficiary:	**UCY**

Dissemination Level	
Public	X
Confidential, only for members of the Consortium (including the Commission Services)	

Authors List

Leading Author				
First Name		Last Name	Beneficiary	Contact e-mail
Kyriaki		Psara	UCY	psara.kyriaki@ucy.ac.cy
Co-Author(s)				
#	First Name	Last Name	Beneficiary	Contact e-mail
1	Andrea	Cimmino	UPM	cimmino@fi.upm.es
2	Alba	Fernandez Izquierdo	UPM	albafernandez@fi.upm.es
3	Christos	Patsonakis	CERTH	cpatsonakis@iti.gr
4	Ioannis	Moschos	CERTH	imoschos@iti.gr
5	Apostolos	Tsolakis	CERTH	tsolakis@iti.gr
6	Dimos	Ioannidis	CERTH	djoannid@iti.gr
7	Venizelos	Venizelou	UCY	venizelou.venizelos@ucy.ac.c
8	Ioannis	Aristidou	UCY	aristidou.ioannis@ucy.ac.cy

Reviewers List

Reviewers			
First Name	Last Name	Beneficiary	Contact e-mail
Phivos	Therapontos	EAC	PTherapo@Eac.com.cy
Dan	Clarke	KiWi	dclarke@kiwipowered.com

Legal Disclaimer

The DELTA has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 773960. 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 objective of "T1.2 Architectural Design, Functional & Technical Specification" is to deal with the delivery of a complete set of system requirements, dealing with both functional and non-functional (social, environmental, economic, privacy, security and performance) system requirements. Following the latter, the task also supports the knowledge transferred from requirements into concrete and interconnected components that will define the DELTA Architecture.

The initial architectural framework and its components' functionalities, derived during the first reporting period, have been presented in the first version of this deliverable. In this updated and final version of the report, Section 3.3 has been updated to align with the introduction of an additional module that has been added to the DELTA Architecture. The update accommodates the separation of the "Grid State Simulation" component as an individual module that communicates with the DELTA Aggregator/Energy Retailer through the Grid Simulation Engine CIM. In Section 4, two separate tables have been created for the functional and non-functional requirements respectively, where i) 14 new and 3 updated functional requirements, and ii) 5 new and 2 updated non-functional requirements have been provided. A new field labelled "status" has been added to separate the newly introduced or updated requirements. Additionally, the DELTA Architectural Layers were updated in Section 5.1, to encompass the addition of this module. Subsequently, Section 5.6 has been added in order to describe the "Grid State Simulation" as an individual layer. Moreover, updated descriptions and diagrams of the individual components, that depict the newly introduced functionalities, are included in Section 5.7 of this final version. Changes in the communication between the components also affects the information flow throughout the platform hence the information flow diagram in Section 6.1 that has been updated to comply with the current implementation.

Methodology

The first part of the deliverable presents the methodology used to achieve and document the architecture. To develop and describe the DELTA architecture the international standard IEEE 42010 'Systems and software engineering — Architecture description'[1], which defines core elements like viewpoint and view, is applied. In order to implement and execute this methodology, the approach introduced by Rozanski & Woods[2] is followed. Chapter 2 introduces the methodology and its application in the Framework Design Phase. It implies a process based on a set of relevant architecture viewpoints. For DELTA three functional viewpoints have been defined, namely functional view, deployment view and information view.

The conceptual architecture of the DELTA platform in Chapter 3. This is an overview of the DELTA platform, describing the major building blocks of the system in the form of software modules and dependencies. The main identified architecture components are:

- **DELTA Aggregator/ Energy Retailer:** An extended version of the existing aggregator named the DELTA Aggregator is designed and implemented on existing energy assets while also allowing communication and interaction with the DELTA Virtual Nodes.

- **DELTA Grid State Simulation:** This component in responsible from running simulations and monitoring the grid state in order to identify any deviations from the normal behaviour of the grid.

- **DELTA Virtual Node:** The DVN is a cluster of customers that was formulated based on key common characteristics among the customers as defined by the. When a customer alters one of these parameters, he/she will be automatically reassigned to another cluster.

- **Innovative Customer Engagement Tools:** Three main customer services will be introduced by the DELTA solution and all three of them are interlaced through common user database and information exchange.

- **Delta Fog Enabled Agent:** Within the DELTA framework each customer will be equipped with an intelligent device that will provide real-time information regarding energy related data. The local intelligence within the FEID will allow the use of historical information to be automatically taken into consideration when applying a DR signal.

- **DELTA Repository:** The proposed DELTA architecture will include a Repository, in order to allow access to aggregated information from the aggregation-level business applications.

- **DELTA Cyber Security Services:** Infrastructure planning solutions will be based upon the DELTA Information Model principles conforming to the security requirements outlined in recent standards. This methodology will enable a decentralized, planned and safety-oriented management of the entire system's life cycle. The proposed decentralized architecture of the project ensures secure handling of virtual nodes providing privacy and security services using blockchain.

In line with the above methodology, this reports documents requirements, viewpoints and use case analyses.

Requirements

In Chapter 4, the Volere methodology is used to document the resulting functional and non-functional system requirements. System requirements influence the architectural design process in that they frame the architectural problem and explicitly represent the stakeholders' needs and desires. Functional and non-functional requirements have been carefully selected in order to ensure that they make sense in the context of the final outcome of the project and conveyed to all the team members working on it. During the architecture design many possible scenarios for system usage have been analysed and a risk/cost perspective adopted in a review.

Viewpoints

Chapter 5 defines the functional view which describes the functional elements, their functionality, their responsibilities, and their primary interactions with other elements. In the first phase, the partners have described in detail their main components and their expertise requested by DELTA. The second phase built on the requirements outlined in D1.5, leading to the DELTA architecture. In the scope of this viewpoint a high-level specification of each identified software module is provided.

Chapter 6 defines the deployment view which describes how and where the system will be deployed, which physical components are needed, what are the dependencies, hardware requirements and physical constraints. The information view, described in Chapter 6, defines the application domain models and the data flow as well as data distribution.

Chapter 7 presents a high-level deployment view describing the hardware and/or software environments in which any final versions of components will be deployed.

Use Case Analysis

Finally, in Chapter 8, several use cases have been instantiated through sequence diagrams. The purpose of these sequence diagrams is to clarify how the DELTA platform will work and which components are relevant to achieve different tasks. In the second phase, applications and platform services have been preliminary tested by means of UML sequence diagrams by taking as input uses cases defined within D1.5.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

Term	Description
VPP	Virtual Power Plant
UML	Unified Modelling Language
SOA	Service-oriented Architecture
KPI	Key Performance Indicator
INEA	Innovation and Networks Executive Agency
EC	European Commission
FEID	Fog Enabled Intelligent Device
DR	Demand Response
DSS	Decision Support System
DVN	DELTA Virtual Node
DVNP	DELTA Virtual Node Platform
F	Functional requirement
NF	Non-functional requirements
IEEE	Institute of Electrical and Electronics Engineers

1. Introduction

1.1 Scope and objectives of the deliverable

Within the scope of the DELTA work package (WP1) structure this deliverable has been produced as Task 1.2 -Architectural Design, Functional & Technical Specification. It is responsible for analysing the relevant state of the art as well as specifying the DELTA system architecture. This deliverable "D1.6: DELTA Overall Framework Architecture v2" is the final version of the DELTA platform. The DELTA Architecture and the Architectural Layers in Chapter 3, the DELTA requirements in Chapter 4, as well as the overall information flow diagram in Chapter 6, along with the individual component diagrams in Chapter 5 have been updated to depict the new dynamics that comply with the current implementation.

The goal of this deliverable is to provide a high-level overview of the DELTA software architecture, summarizing the technical and functional design. It deals with the delivery of a complete set of system requirements, addressing both functional and non-functional (social, environmental, economic, privacy, security and performance) system requirements.

Following the latter, the deliverable also supports the knowledge transferred from requirements into concrete and interconnected components that define the DELTA Architecture. It describes the basic functionality of the DELTA platform and introduces functional descriptions of each component. The architectural description includes aspects related to the identification of the major system components, how they should interact and how their external interfaces should be defined. Starting from delivering a general overview of the DELTA system design, every component of the DELTA framework is described presenting a high-level design of the system's individual components, including construction components, detailed description of the functionalities, as well as communicational requirements; and finally revealing in full detail functional and technical specifications.

Furthermore, within this deliverable there is an open specification for establishing interfaces & information flows between different actors/stakeholders, leveraging: (1) outcome of the W3C Web of Things on describing security constraints and protocol bindings for supporting the required interaction patterns; (2) communication protocol and syntactic interoperability layer provided by OpenADR specification. The different components of the DELTA framework make use of the DELTA ontology so as to semantically annotate and enrich information flows that establish with their corresponding interlocutors, so that overall semantic interoperability within DELTA ecosystem is achieved.

1.2 Structure of the deliverable

The purpose of this deliverable is to describe the updated and final version of the DELTA Conceptual Architecture. The goal of the conceptual architecture is to provide a holistic view on the DELTA system architecture, its building blocks, components, interdependencies among components and related constraints such as development methodology. To address these requirements, this report is structured as follows:

Chapter 2 introduces the methodology used to define and document the architecture that has been defined. The documentation of the architecture has been based on the standard IEEE 1471 "Recommended Practice for Architectural Description for Software-Intensive Systems". It implies a process based on a set of relevant architecture viewpoints. For DELTA three functional viewpoints have been defined, namely functional view, deployment view and information view.

Chapter 3 gives an overview of the conceptual architecture – a high-level description of the DELTA system architecture – introducing the functional components of the architectural layers. This chapter has been updated due to important transforming decisions being made during the implementation of some of the components. We therefore provide a high-level description of the aforementioned components.

Chapter 4 provides a complete set of system requirements, addressing both functional and non-functional (social, environmental, economic, privacy, security and performance) system requirements.

Chapter 5 is the Functional View, providing the high-level specification of each component, its functionality, and their interactions. The implementation of some of the components allowed us to review the input/output dependencies between components and therefore updated some of the component diagrams.

Chapter 6 The information view documents information management including storage and distribution within the system. Its aim is to provide a unique and consistent interpretation of the lifecycles of the information objects handled by the infrastructure. Similarly to components diagrams the information flow diagram has also been affected by the new dynamics among the components. Therefore, an updated version of the diagram is provided in this chapter.

Chapter 7 is the Deployment View, which provides an overview of the hardware requirements by describing how and where the system will be deployed, which physical components are needed, what are the dependencies, hardware requirements and physical constraints.

Chapter 8 Applications and platform services have been preliminary instantiated by means of UML sequence diagrams by taking as input uses cases defined within Deliverable D1.5. The purpose of these sequence diagrams is to clarify how the DELTA platform will work and which components are relevant to achieve different tasks.

Chapter 9 sums up the main conclusions and findings of this deliverable, and the next steps for the subsequent deliverables.

1.3 Relation to Other Tasks and Deliverables

Having completed and updated the previous step T1.1. (User and Business Requirements Definition) in WP1, providing a set of user and business requirements, this deliverable defines the system architecture, preparing for prototypal implementation to be carried out by the technical work packages. T1.1. has provided an analysis study that determines the User and Business Requirements of the proposed DELTA framework. Various identified stakeholders have been engaged in the design process to meet their requirements and ensure the results in terms of usability and accessibility of the DELTA project. The analysis captured the user and business requirements of existing infrastructure with special regard to business capabilities, limitation and necessities that have been used to define the incentive DELTA architecture in T1.2, towards fully exploiting the project's objectives and innovative features. Building upon the user and business requirements and the feedback received from identified engaged stakeholders (in the form of questionnaires, interviews, etc.), a wide range of application scenarios and use cases has been defined, through which the innovative technologies and services of DELTA will be extensively evaluated

This deliverable is the final agreement on the DELTA software architecture and development framework the following deliverable D1.7 will elaborate on specific aspects of the proposed architecture in detail. D1.7 –DELTA Information Model will describe the Common Information Model (CIM), which is referred to as the Information View in the Rozanski & Woods methodology. This includes a description of the different kinds of data and data formats consumed and/or produced by the different modules. Further, D1.7 will elaborate on how the semantic mapping between these different data models is implemented in the CIM. The goal of this task is to provide the DELTA ontology network (i.e., Common Information Model) to be used throughout the different components of the DELTA framework and the different actors that will consume the information. The DELTA ontology will serve as a shared vocabulary and model that replaces the missing blocks of information between different standards of the Demand Response domain.

2. Design Approach & Methodology

This section details the approach and methodology that has been followed to arrive at the framework architecture presented herein. The architectural descriptions detailed in this report comprise the second consolidation of dependencies, input/output flows and specifications of the full set of architectural components. Throughout the course of the DELTA project, an iterative approach has been used to arrive at the final system architecture and the architecture documentation elaborated with detailed and specific descriptions.

An overview of the approach used to arrive at the DELTA system architecture description is presented in Figure 1.

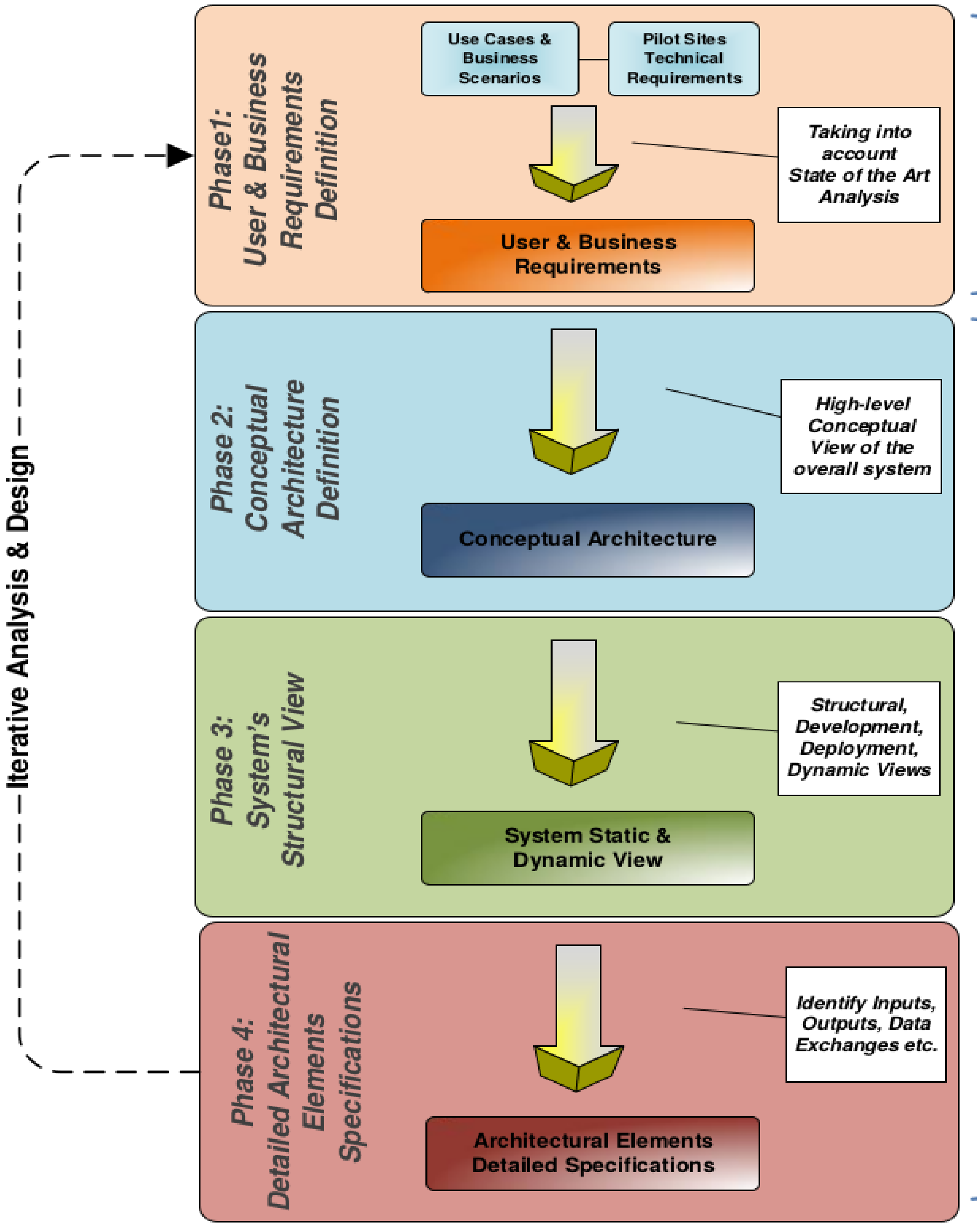

Figure 1. Design Approach for System Architecture

2.1 Fundamental Approach

Close attention is paid here to the international standard IEEE 42010 ‘Systems and software engineering — Architecture description’ [1]. The document provides guidelines for the description of system architectures for complex systems. The goal of the conceptual architecture is to provide a holistic view on the system architecture, its building blocks, components, interdependencies among components and related constraints such as development methodology. This section introduces the methodology and its application in the Framework Design Phase. In order to implement and execute this methodology, the process introduced by Rozanski & Woods [2], which is closely aligned to the IEEE 42010 standard, is followed.

2.2 Design Principles

The DELTA architecture will 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 will be as technology independent as possible, based on existing standards and incorporate (when feasible) the use of generic and standardized solutions for which several key technologies (open source, commercial, etc.) are available.

A set of key design principles have been defined and specified to ensure that the architecture design minimizes costs and maintenance requirements; and promotes extendibility, modularity and maintainability. These design principles are follows:

- **Minimised Upfront Design**
  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.
- **Separation of Concerns**
  The overall system should be divided into distinct features with as little overlap in functionality as possible. The ultimate goal of this principle is, on the one hand, to minimize interaction points and, on the other hand, to ensure increased cohesion and low coupling.
- **Single Responsibility**
  Each architectural element shall be responsible for only a specific feature or functionality, or even aggregation of cohesive functionality.
- **Least Knowledge**
  An architectural element should not directly have access to the internal details of other architectural elements.
- **Don’t Repeat Yourself (DRY)**
  Avoid repeating the same functionality or intent in more than one architectural element 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.

2.3 The Software Architecture Design Process

Rozanski & Woods[2] define the architectural design process as the decompositions of a system into different components and their interactions to satisfy system requirements. During this process stakeholder needs and concerns are captured along with the corresponding architecture that is clearly and unambiguously described. This results in the architectural description and is a crucial step in the project because provides a clear understanding of the problem and affects the quality of the implementation. A broad set of principles must be considered, and stakeholder communication should be maintained to ensure continuous progress towards addressing their concerns and incorporating their requirements. Complex projects require a flexible design process that is able to adapt quickly to changes in the requirements and environments.

During the design process system elements and their relationships must be identified by constructing the foundation of the system, decomposing the system into its main components and validating the overall architecture. Following the establishment of the architecture, evaluation of the architecture design by assessment of the quality attributes in comparison to the initial estimates should take place. The final step is to transform the architecture design until it completely satisfies the quality attribute requirements. This step includes the selection of design solutions that improve the quality of the attributes while preserving the original functionality.

2.3.1 Architecture Definition Activities Overview

The purpose of the design process is to identify design elements and their relationships. Therefore, during the process there are many activities that need to be accomplished including the construction of the foundation of the system, the decomposition of the system into its main components and the validation of the overall architecture.

The scope and context of the DELTA architecture was realised through involvement with stakeholders and is documented by the user and business requirements analysis in D1.5. The stakeholders were included to discuss their architecturally significant needs and desires and capture quality properties. The processes leading to the architectural description are depicted in Figure 2.

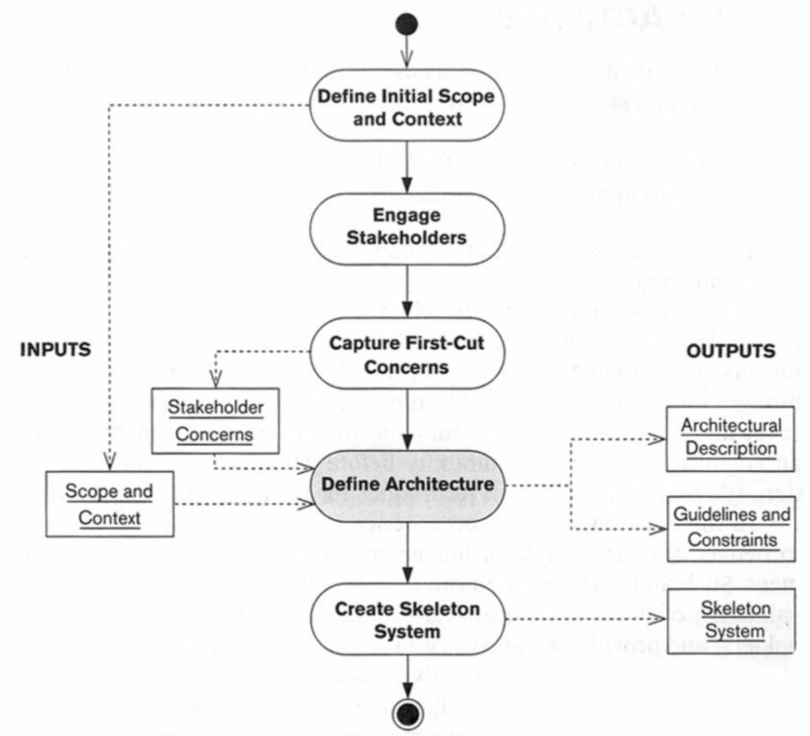

Figure 2. Architecture Definition Activities [2]

Architecture design requires critical supporting activities such as management, communication, reasoning, decision making, and documentation. Strong communication and management among the stakeholders is required in order to identify the knowledge, system requirements and design contexts. This can be achieved by various bilateral and internal meetings to resolve dependency and specification issues of the components. Based on the derived knowledge the next activity was gathering decision problems and associating them with proposed solutions. This process entails evaluation of architecture requirements and decisions as well as conducting architecture analysis. The information collected during the first two activities is documented based on several static and dynamic descriptions.

2.3.2 Architecture Definition Activities To Date

Towards the architecture definition, the following workshops/meetups were organized, in order to iteratively define the overall architecture of DELTA and all of its views:

- Online discussion sessions in WP1 bi-weekly meetings
- Physical architecture technical workshop at UCY on 5th March 2019
- Physical meeting and architectural requirements discussion at KiWi on 17th April 2019

2.4 System Requirements

The DELTA deliverable D1.5 documents the final set of system Business Cases, Use Cases and User Requirements. A business case captures the reasoning that facilitates a decision to start a project. A use case is a list of event steps typically defining the interactions between a system and an actor in order to accomplish a specific goal. User requirements, often referred to as user needs, describe interactions of the user with the system, such as what activities that users must be able to perform.

For the DELTA deliverable D1.5, existing processes used by some of the consortium partners utilising modern and agile frameworks to document and manage user requirements were followed in documentation and requirements gathering. This has led to a combination of online templates derived from the enhanced Volere Methodology and online tools aiming to make to set of user requirements standardised, trackable and prioritized. As such, a JIRA framework was deployed by CERTH to allow user requirement management. This formalized process allows keeping track of the requirements during system development and to quickly adapt to changing or upcoming requirements. Therefore, for the DELTA deliverable D1.6 the Volere methodology is used to document the resulting functional and non-functional system requirements.

System requirements influence the architectural design process in that they frame the architectural problem and explicitly represent the stakeholders' needs and desires. Functional requirements define how a system, or its components should function, i.e. the specific behaviour between inputs and outputs. Non-functional requirements describe criteria that can be used to judge the functions of a system, also known as quality attributes. Non-functional requirements might be further subcategorised to: look and feel, usability and humanity, operational and environmental, maintainability and support, cultural and political, security, performance, and legal.

Functional and non-functional requirements need to be carefully selected in order to ensure that they make sense in the context of the final outcome of the project and conveyed to all the team members working on it. During the architecture design many possible scenarios for system use are considered and the associated requirements reviewed from a risk/cost perspective. The best way to write system requirements is through use cases from deliverable D1.5. The use cases help us to trigger requirements from each event/use case and ensure their completeness while considering how the users will interact with the entire system.

Requirements should be testable, consistent, unambiguous and rational; and should always keep the various actors in mind. The following table indicates how to define requirements based on the Volere methodology.

Table 1. Requirements Documentation Format

ID	Unique ID
Description	A one sentence statement of the intention of the requirement
Type	Functional requirement or non- functional requirement
Rationale	A justification of the requirement
Originator	The person who raised this requirement
Fit Criterion	A measurement of the requirement such that it is possible to test if the solution matches the original requirement
Priority	A rating of the customer value (Must Have, Should Have, Could Have)
Use case	The associated use case as defined on D1.5
Status	Since this is an updated report of the requirements with the status field we indicate if this requirement is derived from the initial extraction of requirements or the updated version

2.5 Viewpoints

The development and description of the DELTA architecture has been based on the standard IEEE 1471 "Recommended Practice for Architectural Description for Software-Intensive Systems". This recommended practice addresses how to describe the architecture of software-intensive systems. It describes a system based on a set of relevant architectural views and their corresponding viewpoints. A *view* is a representation of a system from the perspective of a related concern held by one or more of its stakeholders. A *viewpoint* is a pattern or template for constructing individual views. It establishes the guidelines, principles, and template models for the construction and analysis of a particular view.

For the DELTA project three viewpoints have been defined, namely, the functional view, the deployment view, and the information view. Each architectural viewpoint is determined by its viewpoint name, the stakeholders addressed by the viewpoint, the architectural concerns "framed" by the viewpoint, the viewpoint language (including notations, model, or product types), and the source (author, date, or reference to other documents). Eventually all system diagrams will be modelled in the Modelio tool [2] to ease automated refreshing of architectural diagrams when there are updates to components or relationships.

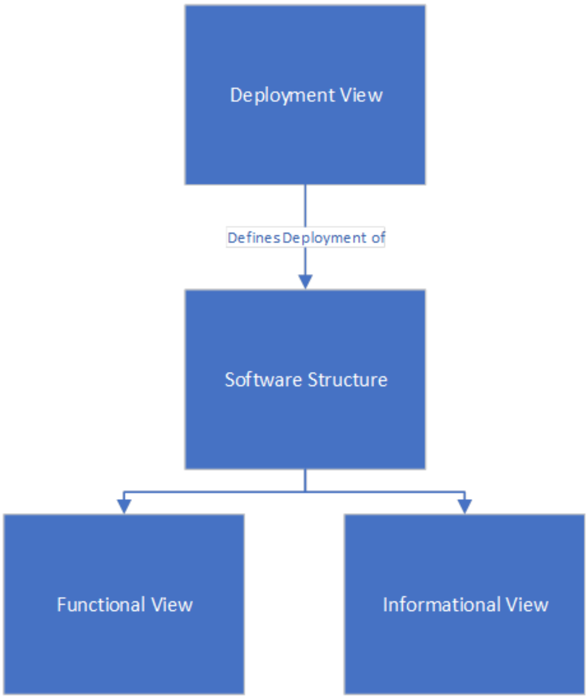

Figure 3. Viewpoint Dependencies

2.5.1 Functional View

The Functional View of the system describes the architectural components that deliver the system functionality. These components are represented as functional elements based on their responsibilities and their primary interactions with other elements. A functional model does not rely on operations that may occur during runtime since it can only express time-free and sequential execution semantics. This is usually the most important viewpoint, as it reflects the quality properties of the system and influences the performance, the maintainability and the extensibility of the system.

The functional view uses component diagrams to document how the system will perform the required functions as specified by the functional and non-functional requirements in section 4. Component diagrams are essentially UML class diagrams that focus on system components. The view typically contains functional elements, interfaces, connectors and external entities. Functional elements constitute the well-defined components of the system that have particular responsibilities and expose well-defined interfaces that allow them to communicate with other elements. Interfaces specify how the functions of a component can be accessed by other components. External entities are connectors which are essentially other entities that communicate with the system indicating the dependencies to other systems or components. The functional view consists of two sections the Platform overview and the Component Overview.

Platform overview Component Diagram
The platform has been designed on top of several layers which consist of different components that interact. This task should provide the overall functional architecture of the platform in the form of a component diagram. The architecture has been designed on the on the basis of the requirements, business requirements, use cases identified in WP1 and reported in deliverable D1.5.

Component Overview
The component overview provides the details and descriptions of the main components and their subcomponents. A description for each component that comprises of a definition of the components and their subcomponents, the interactions between them, and the data interfaces involved; is given in this document. To describe each identified system component and its interaction, a high-level

specification in the form of component diagrams is provided. This section also includes a table describing all the interfaces that interact with the specified component. The outline for the component overview description is given below.

Description:

Some sentences about the purpose of the component and its core.

Component Diagram:

Description of the individual component and the integrated component diagrams highlighting cross-component dependencies with component diagrams, such as:

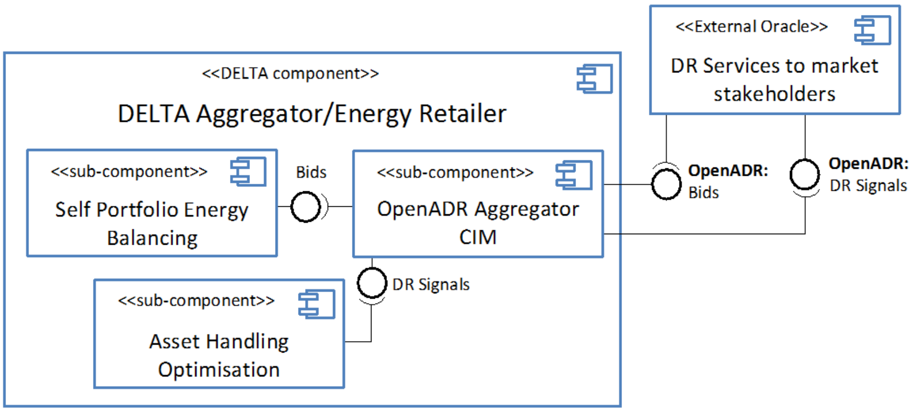

Figure 4. Component Diagram Example

Description of interfaces:

Table 2. Description of Interfaces Outline Example

Interface	R/W	Description
Name	Read or write	A few sentences of the intention of the interface

2.5.2 Deployment View

The Deployment view documents the physical environment into which the system will be deployed and the dependencies the system has on its environment. Specifically, it captures (i) the hardware/software environment of the system (e.g. general-purpose hardware to execute the main functional elements of the system, storage hardware to support databases, hardware that allows users to access the system, network elements required to meet certain quality properties such as firewalls for security, etc.) (ii) the associated technical environment requirements (e.g. the type of operating system run on the devices) and (iii) a mapping of the components to the runtime environment . The technical infrastructure used to execute the system is described by infrastructure elements like geographical locations, environments, computers, processors, channels and net topologies.

Deployment Diagram

The static deployment view of a system (topology of the hardware) comprises of the system's physical aspects and is modelled by a deployment diagram. The deployment diagram is a structure diagram that shows the architecture of the system as a distribution (deployment) of software artefacts to deployment targets. It illustrates how and where the system is to be deployed as realized by the configuration of run-time processing nodes and the components that live on them.

Deployment diagrams are related to component diagrams with the main difference that component diagrams are used to describe the components whereas deployment diagrams show how these components are deployed in hardware. Also note that the hardware components in the deployment diagram should be designed efficiently and in a cost-effective manner in order to meet the business requirements from Deliverable D1.5.

Deployment diagrams are a special kind of UML class diagrams, where nodes represent hardware devices, processors, and software artefacts. Embedded or nested nodes represent the internal construction of sub-nodes. In a graphical representation the nodes appear as three-dimensional boxes, and the artefacts allocated to them appear as rectangles within the boxes. Components are indicated by the standard component icons. Deployment relationships indicate the deployment of artefacts, and manifest relationships revealing the physical implementation of components.

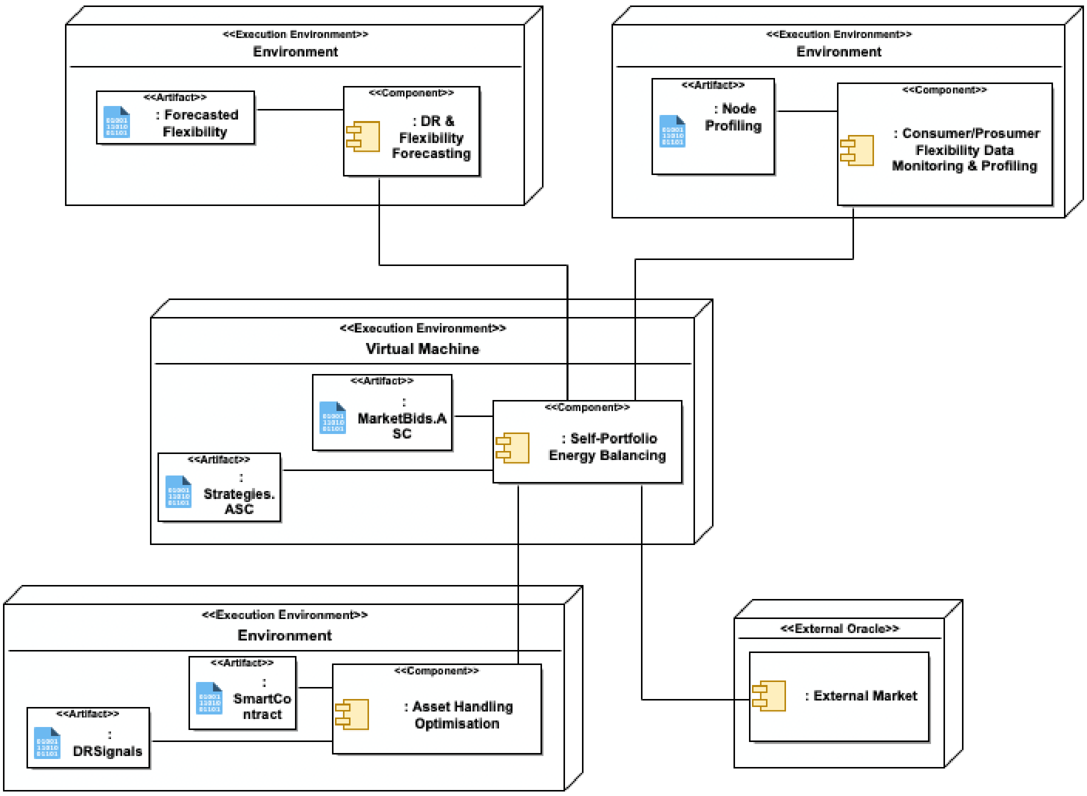

Figure 5. Deployment Diagram

Components Hardware Requirements

Another part of the deployment view is the documentation of the required deployment environment for the DELTA project. In this task the physical architecture of the system is specified, to inform the mapping between that and the logical operations. This provides an overview covering the known hardware requirements of the software modules and used tools. The table below lists the preliminary hardware requirements concerning the main DELTA components.

Table 3. Component Deployment View Information Outline Example

Component Name	Responsible Partner	Associated Task	Supporting Partners	Deployment Platform	Hardware requirements
A Component is a modular part of a system whose behaviour is defined by its provided and required interfaces	Lead beneficiary	Action Tasks	Support Beneficiaries	It could be a device or an execution environment.	Properties or guiding parameters that must be defined for deployment to occur

Existing Software and Hardware Requirements

The deployment view of the platform might depend highly on already existing software and hardware. A brief description of the already existing components is required along with their associated requirements

2.5.3 Information View

The ultimate purpose of virtually any infrastructure is to share and manipulate information. Thus, this viewpoint develops a complete but abstract view of the static data structure and the dynamic information flow in the examined system. Specifically, the information view documents information management including storage and distribution within the system. Similarly to a high-level ontology, the information view aims to provide a unique and consistent interpretation of the lifecycles of the information objects handled by the infrastructure. The objective of this analysis is to answer the big questions around structure, content, ownership, and data migration.

The Information View includes a description of the different kinds of data and data formats consumed and produced by the different components and the semantic mapping between them. The information view specifies the type of the data objects and the relationships between them. It also defines the constraints on information objects and the rules governing those lifecycles. The focus lies on the data itself, without considering any platform-specific or implementation details. It is independent from the computational functions and interfaces that manipulate the data, or the nature of technology used for storage.

Models within the Information viewpoint provide a means of portraying the information requirements, rules, and constraints of the business activities. The information viewpoint language is defined in terms of two schemas. The first one being the Data model which is a static schema that indicates the state of one or more objects at some point in time. The second one being the information flow which is the dynamic schema that allows state changes in one or more data objects. After constructing a static schema consisting of different objects, the dynamic schema that carries out the business logic between these data objects is created. Both schemas are derived from understanding the business, and system requirements as well as identifying the domain components and their relationships.

Data Model
The static schema used here is called the data model which is a conceptual model of the domain that incorporates data. On a basic level, data modelling can be understood as an object-oriented abstraction that envisions the collection of all of the objects in that system that contain data, and a representation

of how that data behaves and interacts with each other. It is an abstract model that organizes elements of data and standardizes how they relate to one another and to properties of the real-world entities

To visualize domain models, the set of conceptual classes are identified as a UML class diagram. However, no operations are defined in Data models but only data objects, entities, associations between them, and attributes of entities. An entity is a person, place, thing, or event that must be represented in the database. Associations between the entities are modelled next to show which entities will interact with the others. Attributes are the individual pieces of information you want to store about the entities. These become the columns in the database. The following example indicates how a Data model should be illustrated.

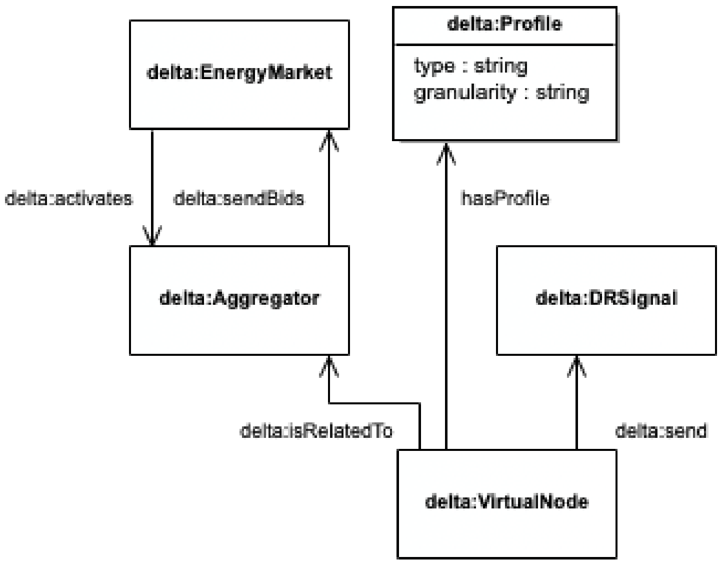

Figure 6. Data Model Illustration Example

Information flow Diagram

Information flow diagrams represent how information is exchanged (or "flows") among the main components of the DELTA platform. Their main purpose is to describe the circulation of information within systems so that sources that send and receive information can be displayed and analysed in different situations. Successful information flow diagrams should highlight gaps that need improvement, display inefficiencies in information, highlight risks such as data confidentiality, display insecure mediums, and they should also provide clarity about who should receive which information when, where and how.

Information flow diagrams are essentially UML sequence diagrams which show exchange of information between system entities at some high levels of abstraction. Construction of an information flow diagram requires the knowledge of different information sources and the connections between them. The sources and targets of information flow can be the following: actor, use case, node, artefact, class, component, port, property, interface, package, activity node, activity partition, or instance specification. A dashed line with an open arrow pointing away from the source to the target is used to represent information flow. The keyword "flow" may be written above or below the dashed line. Information items represent the abstraction of data and act as information flow connectors, representing the flow of transfer of information from source to target.

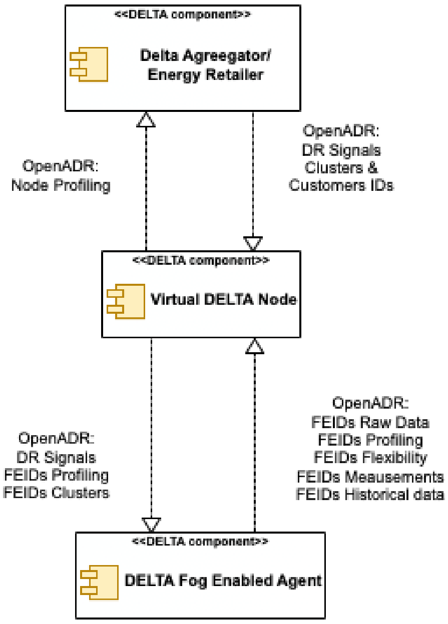

Figure 7. Information Flow Diagram Example

2.6 Perspectives

According to Rozanski & Woods[2] an architectural perspective is a collection of activities to ensure that a system exhibits particular quality properties that are a common concern to many architectural views. These concerns are normally driven by the need to ensure that quality properties are not forgotten during architecture design rather than to provide a particular functionality of the system. However, these quality properties are critical for a successful and careful consideration is essential alongside viewpoint analysis

Some of the proposed quality properties are performance, availability, security, location, maintenance, privacy, regulation etc. Such architectural perspectives need to be carefully chosen according to the DELTA requirements. The following perspectives are identified by the architecture definition process of the DELTA project:

Performance and Scalability is critical for the DELTA project since the system should be able to handle increased processing volumes by predictably executing within its mandated performance. The performance of the system should be measurable with qualitative and quantitative indicators concerning the customer experience.

Availability & Resilience is essential for the DELTA project as the system is expected to respond to real time information and operate accordingly.

Security is an important aspect of the DELTA architecture in order to restrict access to data or functionality to the unsuitable users. Confidentiality and integrity of the information transmitted should be guaranteed to proactively ensure the secure exchange of information between DELTA stakeholders.

Maintenance, Privacy & Usability are other concerns for the DELTA solution hence these architectural perspectives Should be born in mind throughout the development of the project.

Conventional views and viewpoint 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 designed system, 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 DELTA, their dependencies and the respective architectural views, the architectural perspectives are also considered, which are analogous to a viewpoint. In this report, several quality properties are addressed for all architectural elements of the system, as these are outlined in Table 4:

Table 4. Overview of DELTA Architectural Perspectives

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.
Additional Perspectives to cope with DELTA’s non-functional requirements	
Maintainability	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 and 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.

The importance of each of the aforementioned perspectives in regard to the views of the DELTA project may vary and the benefits of addressing them 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 these in the architecture definition process, the aforementioned perspectives will aid later decision making (implementation, deployment and operational phases). Within DELTA, a table will be provided 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 DELTA’s architectural model will meet the functional and non-functional requirements, the aforementioned proposed perspectives should be taken into account. These perspectives could be modified or enriched by partners according to characteristics of the components.

2.7 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 static viewpoints for the architecture will be presented in Chapters 5 and 6 herein, dealing with the functional and information views.
- The dynamic structures of a system illustrate how it actually operates during its utilization, depending on the various scenarios of use and use cases defined, including the way each component acts within them. Thus, the system's dynamic model and structures define its runtime architectural elements and their interactions due to internal or external stimulus. The internal interactions refer to information flows among architectural elements and their parallel or sequential execution of internal tasks, including the potential expression of the effect they may have on the information. The dynamic viewpoints for the architecture are presented in the form of Use Case analyses with accompanying sequence diagrams

2.8 Service-oriented Architecture (SOA)

The DELTA system consists of various services from different components that need to communicate with each other across different platforms, programming languages, execution environments, and development methods. This might lead to interoperability and integration problems among components that have been built on heterogeneous frameworks. Service-oriented Architecture (SOA) defines a set of design principles that are independent of products, vendors and technologies. For the DELTA project, the major SOA principles that will guide the architecture design process are the following:

Service contract

Communication among services follows defined service description documents that describe the technical interfaces of services also known as service contracts. A technical service contract specifies an API of the service's functionality.

Loose coupling

Services have the ability to remain independent of the implementation of other services. The facilitated dependencies between services are realized by the implementation of well-defined interfaces which allow transmission of information without breaking the service contract.

Reusability

Services should be designed to provide reuse of functionality to significantly reduce the time spent during the development process.

Service abstraction

The service contract defines the interaction between services by hiding as much of the underlying details as possible. Loosely coupled relationships invoke services by requiring no other information or knowledge of implementation details.

3. Conceptual Architecture

This chapter provides an overview of DELTA's conceptual architecture introducing the major layers and sub-layers of the DELTA platform along with the included architectural components. Distinctions between the different layers and sub-layers are highlighted.

DELTA proposes a DR management platform that aims to establish a more easily manageable and computationally efficient DR solution by adopting and integrating multiple strategies and policies. DELTA will also deliver a fully autonomous architectural design, enabling prosumers to escape the hassle of responding to complex price/incentive-based signals, while facilitating them with a social collaboration platform and enhanced DR visualisation. Furthermore, DELTA will propose and implement self-learning energy matchmaking algorithms to enable aggregation, segmentation and coordination of several supply and demand clusters. DELTA will set the milestone for data security in future DR applications by not only implementing block-chain methods and authentication mechanisms, but also by making use of Smart Contracts which would secure the Aggregators-to-Prosumers transactions.

DELTA's vision is to develop, validate and deliver a novel framework in designing, developing and integrating Demand Response (DR) services by allowing Aggregators to exploit the flexibility of small/medium customers (consumers, producers, and prosumers). 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 DR programs, Virtual Nodes, fog-enabled intelligent devices, machine and self-learning algorithms, multi-agent systems, award schemes, social collaboration platforms, permissioned blockchain and smart contracts. The conceptual architecture of the DELTA platform is depicted in Figure 8. DELTA Updated Conceptual Architecture.

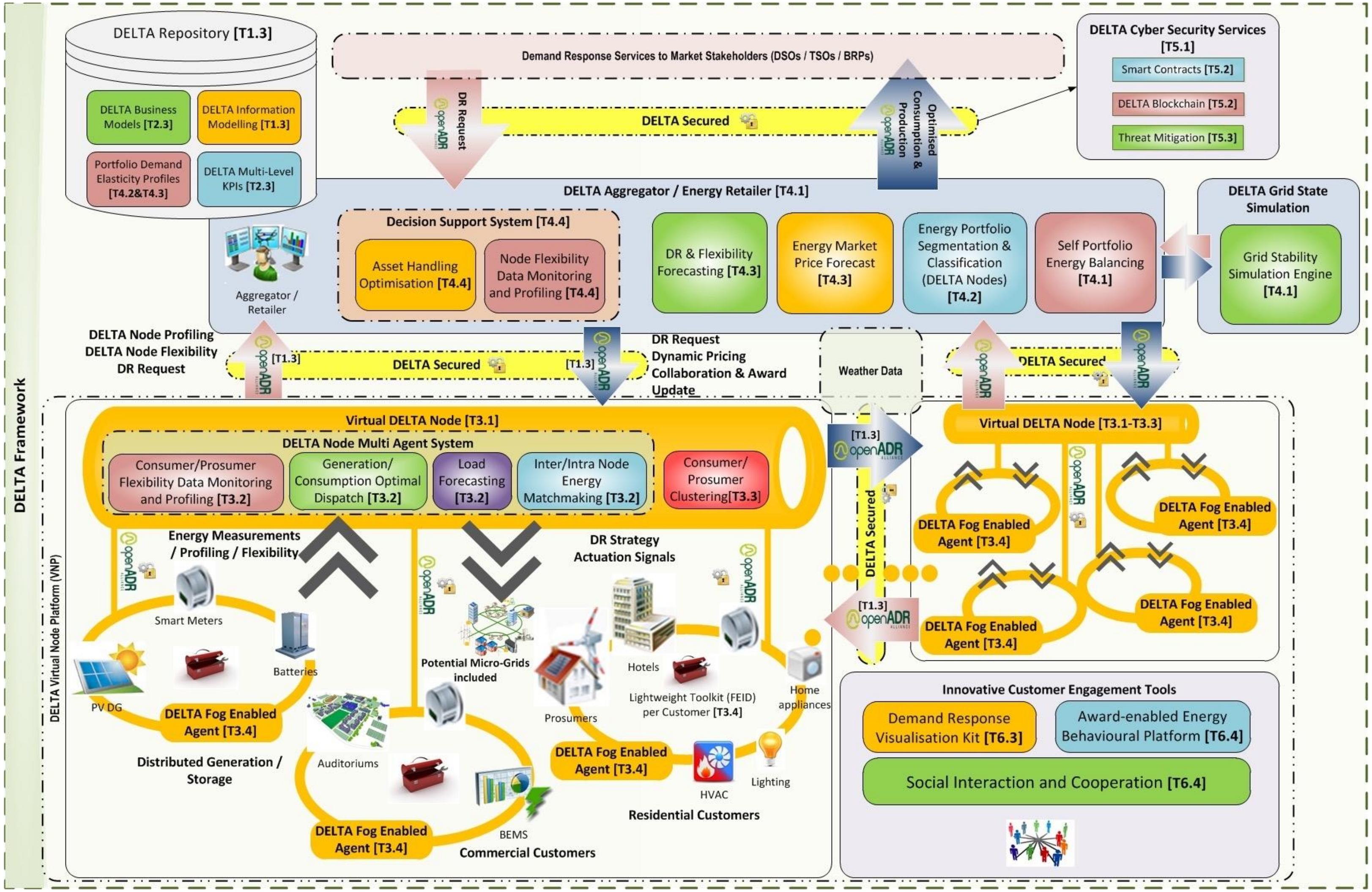

Figure 8. DELTA Updated Conceptual Architecture

The main layers and sub-layers of the DELTA platform are described below.

3.1 DELTA Fog Enabled Agent

Fog computing is a decentralised computing architecture in which storage and computational resources are placed logically between the local network and the cloud to increase the efficiency of operations on and performance of the network. Furthermore, by segmenting network traffic there are security advantages associated with fog computing over conventional cloud computing.

Each customer will be equipped with a fog enabled agent that will provide real-time information, through fog computing, regarding energy related data such as consumption, generation, emissions, available flexibility, etc. based on a strategy (i.e. time interval) imposed to the specific Node by the Aggregator. The agent will be connected directly (through the main circuits) or indirectly (through the BMS) to the customer loads providing accurate, fast and unobtrusive required information, while also assessing environmental and occupancy information where it is available. The local intelligence within the agent, managed by a lightweight toolkit, will allow the use of historical information and simple models to define real-time flexibility and load control based on the incoming DR signal.

3.2 DELTA Virtual Node

The DELTA Virtual Node (DVN) is a cluster of customers (small, medium or large consumers, producers or prosumers) that was formulated based on key common characteristics among the customers, always aligned with the guidelines defined by the Aggregator for each Node. The DVN will be equipped with the following components:

3.2.1 Consumer/Prosumer Flexibility Data Monitoring and Profiling

This component provides a real-time overview of the assets assigned to a specific Virtual DELTA Node, i.e., the DELTA Fog Enabled Agents that a certain Virtual DELTA Node is in charge of managing. The values that this component monitors are namely the data gathered by the different devices that DELTA Fog Enabled Agent handles, as well as, the computed Flexibility related to those devices.

3.2.2 Generation/Consumption Optimal Dispatch

This component aims at establishing energy decision that the DELTA Fog Enabled Agents must fulfil. This component takes from the DELTA Aggregator / Energy Retailer the supplied DR Signals, computes the optimal course that the underneath DELTA Fog Enabled Agents should take, and emits DR Signals to them.

3.2.3 Load Forecasting

This component is vital in handling the DELTA Fog Enabled Agents that a Virtual DELTA Node is in charge of. The forecasted values are a paramount element to establish the right course of actions when DR Signals are sent from the DELTA Aggregator/Energy Retailer. These forecasted values allow maximizing the availability of assets in order to fulfil the energy promises established between the Demand Response Services to Market Stakeholders and the DELTA Aggregator/Energy Retailer. Moreover this activity is the basis for the flexibility forecast.

3.2.4 Inter/Intra Node Energy Matchmaking

This component manages the DELTA Fog Enabled Agents of a certain Virtual DELTA Node. By analysing the profiles of the underlying DELTA Fog Enabled Agents and the current clusters, it aims at reassigning the assets of its Virtual DELTA Node by sending DR Signals to the underlying DELTA Fog Enabled Agents or to other Virtual DELTA Node present in the DELTA Platform while servicing DR schemes originating from the DELTA Aggregator.

3.2.5 Consumer/Prosumer Clustering

This component implements border line techniques to cluster and group in segments the different DELTA Fog Enabled Agents allocated in the same DELTA Virtual Node. The goal of this clustering is to assign each DELTA Fog Enabled Agent to a different Virtual DELTA Node relying on some specified energy profile and social requirements established by the DELTA Aggregator / Energy Retailer. As a result, the clustering will produce suitable configurations to meet the energetic needs required. Techniques that this component implement will be fed by means of the data retrieve through the sub-component Energy Portfolio Segmentation and Classification in the DELTA Aggregator / Energy Retailer that establishes the DVN Clusters. In addition, data from the Consumer/Prosumer Flexibility Data Monitoring and Profiling and incoming DR signals are used to compute the customer clusters.

3.2.6 Blockchain and Smart Contract Tool

This component provides the necessary means to interface with DELTA's Blockchain. For instance, this component allows the retrieval of membership information from DELTA's Blockchain, issuing and monitoring the status of energy-related transactions, as well as instantiating and interacting with smart contracts deployed on DELTA's Blockchain, which is a vital part of participating in, e.g., an explicit DR scheme.

3.3 DELTA Grid State Simulation

The requirements of the GSSE component necessitated the utilisation of the third-party/licensed software, DIgSILENT, that enabled the identification of accurate grid constraint deviations. To this end, the component is developed as a stand-alone module that communicates with the DELTA Aggregator/Energy Retailer through the Grid Simulation Engine CIM. The DELTA Grid State Simulation consists of the GSSE CIM, the Grid Stability Simulation Engine, and Local repository.

3.3.1 Grid Stability Simulation Engine

This component will monitor the entire portfolio in terms of stability (power, voltage, etc.) and will run various simulation scenarios feeding the DELTA Aggregator's DSS with potential risks, including assets that are in the Aggregator's portfolio. Since not all assets are expected to be included in the platform (which will run self-balancing algorithms as well), the DELTA Aggregator will be able to deal with issues that refer to the entire distribution network under its supervision.

3.4 DELTA Aggregator

Even though the concept of the DELTA solution can be applied to existing Aggregator schemas (where each Node will be perceived as a large scale prosumer), an extended version termed the DELTA Aggregator is also defined towards further exploiting the envisioned ICT framework. The DELTA Aggregator will be equipped with the following components:

3.4.1 Energy Market Price Forecast

This component is activated when the Aggregator needs to exploit their participation in the wholesale markets (if such scheme is supported by the market regulatory frameworks). Typical factors that influence electricity prices will be reviewed (such as season/day, weather, fuel prices, demand elasticity etc.) and price forecasting techniques will be employed (regression techniques/SVM/neural networks etc.). However, for use cases involving bilateral agreements and participation in e.g. capacity markets, the market price will be perceived as a known-constant value.

3.4.2 DR & Flexibility Forecasting

This component is of crucial importance in the DELTA architecture, for the Aggregator to conform to their balance responsibility. The Aggregator needs to forecast the energy consumption profiles of the prosumer Nodes lying in their portfolio, before and after a DR request is activated. Similar methods as the ones used for price forecasting can be applied in this case. In order to provide accurate and state of the art forecasting results, a simulation suite will be running all possible scenarios utilising the developed testbed for more effective and robust computational efficiency. This activity is achieved with the inputs from the load forecast component and the respective training data of the DELTA Fog Enabled Agents.

3.4.3 Node Flexibility Data Monitoring and Profiling

The Aggregator will need to supervise each Nodes' flexibility and contextual data, for activating profiling mechanisms for each Node and decide upon their DR request strategies.

3.4.4 Asset Handling Optimization

The DELTA Aggregator will integrate with existing large customers and generation facilities. For that reason a supervisory asset system will be responsible in optimising the mechanisms that will define how each and every element of the portfolio will be handled and which DR strategy should be used in each case towards gaining the most in terms of energy distribution.

3.4.5 Self-Portfolio Energy Balancing

This component will be employed in real-time operation regardless of incoming DR signals (either from the DSO/TSO or the DVNP) as to ensure stability and optimal distribution of electrical energy within the DELTA energy network. An autonomous tool will constantly monitor the portfolio in terms of flexibility and stability and will provide feedback to the DSS for creating DR signals for scenarios that the self-balancing at Node level is not enough or large customer not included in the DVNP present challenges that need to be addressed.

3.4.6 Energy Portfolio Segmentation & Classification

This component analyses all the information provided by the overall energy portfolio, the existing infrastructure and creates guidelines/strategies that will define the way that each Node will create the customer clustering. Moreover, these strategies deployed to each available Node will also include information about reporting intervals, pricing ranges, DR potential strategies, as well as other related and essential restrictions/suggestions that will facilitate the DR communication and maximise the accessibility to Distributed Energy Resources.

3.5 Innovative Customer Engagement Tools

Three main customer services will be introduced by the DELTA solution and all three of them are interlaced through common user database and information exchange:

3.5.1 Demand Response Visualization Kit

A real-time context-aware DR Visualisation Kit that will provide customers with a web-based tool that will visualise real-time and historical energy information in an easy-to-understand.

3.5.2 Award-enabled Energy Behavioural Platform

An Award-enabled energy behavioural service that will promote healthy competition and fun activities based on game mechanics motivating customers to understand, be engaged, and reach Demand Response objectives in a playful manner.

3.5.3 Social Interaction and Cooperation

This platform will enable discussion and knowledge diffusion among consumers/prosumers that improve DR strategies.

3.6 DELTA Repository

The proposed DELTA architecture will include a Repository, in order to allow access to aggregated information from the aggregation-level business applications. This will consist of:

3.6.1 DELTA Business Models

Used to reveal DR strategies, financial targets and other business-oriented parameters related to the operational status of the DELTA Aggregator and Virtual Node Platform.

3.6.2 DELTA Information Modelling

Energy oriented DELTA Information Model, comprising of the ontologies/vocabularies for the DELTA framework.

3.6.3 Portfolio Demand Elasticity Profiles

Energy Portfolio Demand Elasticity Profiles, allowing the business applications addressing specific personalised DR strategies.

3.6.4 Delta Multi-Level KPIs

DELTA Multi-level KPIs are used for assessing the energy performance in all the DELTA layers.

3.7 DELTA Cyber Security Services

Infrastructure planning solutions will be based upon the DELTA Information Model principles conforming to the security requirements outlined in modern standards. This methodology will enable a centralized, planned, and safety-oriented management of the entire system's life cycle.

3.7.1 Smart Contracts

Smart contracts are computer programs written on the Blockchain, containing certain conditions and agreements. Smart Contracts ensure secure energy information exchange within the DELTA energy network. Furthermore, they provide the necessary automation and business logic to the system. This is accomplished by enabling data authenticity, immutable logging, traceability and permission based access for the stakeholders through cryptographic techniques as digital signatures and certificates. In addition, the energy information data flow between the Blockchain actors and the Smart Contracts is mediated through the OpenADR security standards and state of the art privacy and policy algorithms.

3.7.2 DELTA Blockchain

The Blockchain technology utilized to support the DELTA network facilitates secure online transactions on a distributed ledger system. This allows provenance of data's authenticity, auditing, traceability and verification of the transactions. The private permissioned architecture aims to control not only the participation to the network, but also the kind and level of permitted actions each of the participants can perform. With such architecture and policies the security, speed, scalability and robustness of the system are guaranteed.

3.7.3 Threat Mitigation

The project will adopt concrete and dynamic security measures to foresee possible threats and follow certain actions, depending whether it can provide a solution or not. Threat detection will be based on logging normal activity patterns compared to new activity interactions. Abnormal activity can be a massive amount of transactions send at the same time to imitate a DoS attack, or invalid transactions send on purpose to the network to discover possible exploits, or transactions coming from unknown addresses. The Threat Mitigation component will try to block these actions, and if it is not able to fix the problem will send an information message to predefined stakeholders to let them know of its findings.

4. Functional & Non-functional Requirements

In this section the functional and non-functional requirements are document with associated components and use cases.

Table 5. Functional requirements descriptions and associations

ID	Description	Type	Rationale	Originator	Fit Criterion	Priority	Associated Components	Associated Use Cases	Status
1	Run DR scenarios based on potential DSO needs for grid balancing/stabilisation simulating issues that could occur based on current status	F	To increase revenue and support grid stability	UCY	Quantification of DR-related grid constraints	Must have	Grid State Simulation: Grid Stability Simulation Engine	BS3-UC1	Initial
2	Provide feedback of potential risks and physical constrains	F	To demonstrate operation and allow adaptation	UCY	Provision of real-time data and future operation	Must have	Grid State Simulation: Grid Stability Simulation Engine	BS3-UC1	Initial
3	Accurate estimation of the required flexibility	F	To check if possible flexibility provision can lead to potential grid violation	UCY	Restoration of the normal operation of the line	Must have	Grid State Simulation: Grid Stability Simulation Engine	BS1-UC1	New

ID	Description	Type	Rationale	Originator	Fit Criterion	Priority	Associated Components	Associated Use Cases	Status
4	Accurate specification of the geographical location of the grid violation	F	To meet the required flexibility by the assets associated to the violation	UCY	Association of grid constraints with the corresponding assets	Must have	Grid State Simulation: Grid Stability Simulation Engine	BS3-UC2	New
5	Predict upcoming DR event based on dynamic analysis on Day ahead energy profiles	F	To update the grid violations with new predictions based on data over the day	UCY	Provision of forecasted data for the next 24 hours for every pre-specified time step	Must have	Grid State Simulation: Grid Stability Simulation Engine	BS1-UC1	New
6	Secure communication within components that belong in different environments	F	To ensure secure exchange of data between Grid State simulation and Aggregator	UCY	Use of a common information model as an intermediate between Grid State simulation and Aggregator	Must have	Grid State Simulation: Grid Stability Simulation Engine	ALL	New
7	Development of an accurate grid topology	F	To ensure that the grid under evaluation is as close to the actual grid as possible	UCY	DSOs will make available pertinent information about the grid topology	Must have	Grid State Simulation: Grid Stability Simulation Engine	BS3-UC2	New
8	Pre-specified constraints and time steps	F	The platform should allow the aggregator to prespecify the time steps in which the data will be taken as well as the constrains on line loading and voltage	UCY	The platform is able to run simulations on different values of the main parameters	Must have	Grid State Simulation: Grid Stability Simulation Engine	BS3-UC1	New

ID	Description	Type	Rationale	Originator	Fit Criterion	Priority	Associated Components	Associated Use Cases	Status
9	Generate Bids that are sent to the external oracle Demand Response Services for Market Stakeholders	F	To increase revenue and support grid stability targeting a balanced state where the use available flexibility is maximized.	UCY	Calculation and offering of flexibility	Must have	Aggregator: Self Portfolio Energy Balancing	BS3-UC1	Initial
10	Produce a market settlement that specifies a certain behaviour that the DELTA Aggregator / Energy Retailer should consider behaving	F	To decrease expenditure	UCY	Provable demonstration that strategies employed avoided cost and/or generated revenue	Must have	Aggregator: Self Portfolio Energy Balancing	BS1-UC2	Initial
11	Constantly monitor the portfolio's composition and capabilities in terms of stability and flexibility	F	To ensure that flexibility of distributed assets can be aggregated as a single unit to sell services	UCY	Single control requests communicate appropriately	Must have	Aggregator: Self Portfolio Energy Balancing	BS3 – UC1	Updated
12	Cover as much possible of the total requested flexibility	F	To optimally access the energy flexibility market services	UCY	Provided flexibility should have lose to zero deviation from requested flexibility	Must have	Aggregator: Self Portfolio Energy Balancing	BS3 – UC1	New

ID	Description	Type	Rationale	Originator	Fit Criterion	Priority	Associated Components	Associated Use Cases	Status
13	Ensure fair participation based on historical participation data	F	To ensure that flexibility requests are equally distributed to all customers under the Aggregator's portfolio based on previous DR activations	UCY	Fairness index is under a specific percentage	Must have	Aggregator: Self Portfolio Energy Balancing	BS3 – UC1	New
14	Ensure no deviation between committed and delivered flexibility	F	To avoid penalty fees for non or insufficient delivery of the reserved flexibility	UCY	Reliability index is over a specific percentage	Must have	Aggregator: Self Portfolio Energy Balancing	BS3 - UC1	New
15	Prioritize flexibility combinations based on the most profitable options for the Aggregator	F	To ensure maximisation of the aggregator's revenue	UCY	Generate a sorted list of all the combinations of assets that could participate in a request	Must have	Aggregator: Self Portfolio Energy Balancing	BS3 – UC1	New
16	Consideration of the penalty fee	F	To ensure maximisation of the aggregator's revenue even when including penalties	UCY	Consideration of penalty fees for non or insufficient delivery of the reserved flexibility when prioritizing combinations	Must have	Aggregator: Self Portfolio Energy Balancing	BS3 – UC1	New

ID	Description	Type	Rationale	Originator	Fit Criterion	Priority	Associated Components	Associated Use Cases	Status
17	Frequent updating of the fairness and reliability indices used for the optimisation	F	To avoid considering of unreliable assets or frequently used assets	UCY	Update the indices for the associated DVNs after the execution of the latest DR request	Must have	Aggregator: Self Portfolio Energy Balancing	BS3 – UC1	New
18	Store rewards earned by the end-user in the Award-Enable Energy behavioural platform	F	To be consumed later	E7	The awards are received from the Award-Enable Energy behavioural platform	Must have	Innovative Customer Engagement Tools: Award-enabled Energy Behavioural Platform	BS2-UC3	Initial
19	Store KPI's (Key Performance Indicators) received by the DELTA designers	F	To measure the performance and the behaviour of the different customers	E7	The indicators should be based on those collected by European projects and initiatives	Must have	Repository	BS3-UC1	Initial
20	Analyse the incoming signals	F	To process incoming DR signals sent by the external oracle 'DR Services to Market Stakeholders'	CERTH	Should extract the required actions to fulfil the DR signal	Must have	DR Visualization Kit	BS1-UC1, BS2-UC3	Initial

ID	Description	Type	Rationale	Originator	Fit Criterion	Priority	Associated Components	Associated Use Cases	Status
21	Produce a web based tool with demand response visualizations along with other visual analytics information	F	To provide to the end-users an overview of the real time data related to their physical devices	CERTH		Must have	Customer Engagement DR Visualization Kit	BS1-UC1, BS2-UC3	Initial
22	Create an award-enabled energy behavioural service	F	To promote healthy competition and fun activities to end-users based on game mechanics	CERTH	The service should produce awards that could be used as a characteristic for evaluating customer engagement to DR strategies when creating the Node clusters	Must have	Customer Engagement Award-enabled Energy Behavioural Platform	BS2-UC3	Initial
23	Procurements of as accurate as possible weather forecast from online APIs to support both consumption and generation forecasts, as well as flexibility estimation	F	Both consumption and generation are affected by external weather conditions	CERTH	To improve the forecasting results of the respective algorithms the use of weather conditions provide significant added value.	Must Have	Load/Generation Forecasting, Flexibility Forecasting, Flexibility Forecasting, Energy Price Forecasting, DR Forecasting	BSC1-UC1	New
24	Provide real-time information regarding energy related observed values	F	To forward to the upper layers the real time energy related data	CERTH	Historical data, status signal tracked by the FEID per Customer	Must have	FEID	BS1-UC1-2, BS2-UC3-3_1-3_2-3_3, BS4-UC1-2	Initial

ID	Description	Type	Rationale	Originator	Fit Criterion	Priority	Associated Components	Associated Use Cases	Status
25	Generate forecasted energy related values	F	To calculate and forward to the upper layer the forecasted data	CERTH	Forecasted data related to consumption, generation, flexibility	Must have	FEID	BS1-UC2 BS2-UC3 BS3-UC1 BS3-UC2	New
26	The FEID Agent will digitally sign smart contracts on behalf of the prosumer	F	To secure transactions and prove authenticity	CERTH	Digitally Sign smart contracts for end-to-end security	Must have	FEID	BS2-UC1	Initial
27	The FEID Agent will take according actions depending on the smart contract's agreement	F	To enforce actions in order to fulfil the energy scenario	CERTH	The FEID must be able to take actions in order to deliver the energy it agreed with the DVN	Must have	FEID	BS2-UC3_1, BS2-UC3_2, BS2_UC3_3	Initial
28	The FEID will return details about the cluster and the DVN it belongs to	F	To send information about the FEIDs properties	CERTH	The FEID must respond with its detailed information apart from the energy meters and assets	Must have	FEID	BS2-UC1	Initial
29	FEID should be able to control attached / paired devices	F	To provide control actions to the devices	CERTH	Direct control attached devices	Must have	FEID	BS2-UC3_1, BS2-UC3_2, BS2_UC3_3, BS4-UC1, BS4-UC2	Initial

ID	Description	Type	Rationale	Originator	Fit Criterion	Priority	Associated Components	Associated Use Cases	Status
30	FEID should provide hardware security	F	To provide secure environment and communications	CERTH	Pentest the FEID	Must have	FEID	BS4-UC1, BS4-UC2	Initial
31	FEID should communicate with attached / paired devices	F	To get data and control the devices	CERTH	Sample read and write data or control device	Must have	FEID	BS2-UC3_1, BS2-UC3_2, BS2_UC3_3, BS4-UC1, BS4-UC2	Initial
32	Secure remote updates for the firmware and software of the FEID	F	To provide a secure mechanism for future updates	CERTH	Roll a test update for firmware and software	Should have	FEID	BS4-UC1, BS4-UC2	Initial
33	The FEID will return information about customer's profile	F	To send information about the aggregated consumption, generation and storage capacity	CERTH	The FEID must be able to adapt when a device is removed or added and inform the DVN about the updated capacities	Must have	FEID	BS1-UC1, BS2-UC1, BS2-UC3	Updated
34	Smart Contracts will return a status regarding their completion and results	F	To keep track of successful or unsuccessful agreements	CERTH	Prosumers credibility must be modified regarding their successful and unsuccessful delivery of energy services to the DVN	Must have	FEID (Blockchain Client), Cyber Security Services (Blockchain Nodes)	BS2-UC3	Initial

ID	Description	Type	Rationale	Originator	Fit Criterion	Priority	Associated Components	Associated Use Cases	Status
35	The FEID's blockchain client will verify its transactions	F	To ensure security	CERTH	The BC clients must be able to verify the transactions written on the ledger	Must have	FEID (Blockchain Client), Cyber Security Services (Blockchain Nodes)	BS4-UC1	Initial
36	Provide a collaboration platform that offers a large portfolio of useful activities, data and features.	F	To allow end-users to interact among them and the platform	CERTH	The platform should support discussion and knowledge diffusion, Q&A, chatting content posting, timeline of customer activities, social connections e.t.c	Must have	Customer Engagement Social Interaction and Cooperation	BS2-UC3	Initial
37	Search database for previously asked questions and inserted data from other users	F	To gain access to data users will be interested in	CERTH	The Innovative customer engagement tools must provide information	Must have	Social Interaction and Cooperation	BS2-UC3	Initial
38	Accumulate and evaluate in close to real-time the excess or shortage of energy inside the Aggregator's portfolio	F	To provide accurate and close to real-time evaluation inside the Aggregator's portfolio	CERTH	Achieve close to real-time control inside the Aggregator's portfolio	Must have	Asset Handling Optimization	BS1-UC1, BS2-UC2, BS2-UC3_1, BS2-UC3_2, BS2-UC3_3	Initial
39	Identify the new requests' limits taking into account results from the grid stability engine	F	To provide accurate information for the new DR requests	CERTH	Ensure proper formation of DR requests	Must have	Asset Handling Optimization	BS1-UC1, BS2-UC2, BS2-UC3_1, BS2-UC3_2, BS2-UC3_3	Initial

ID	Description	Type	Rationale	Originator	Fit Criterion	Priority	Associated Components	Associated Use Cases	Status
40	Storing the transactions in a blockchain	F	To ensure transparency	NTNU	Audit and verification of transactions	Must have	DELTA Blockchain	BS2-UC3, BS4-UC1, BS4-UC2	Initial
41	Identify and solve potential security risks and threats	F	To fix issues or prevent, when possible, potential attacks	NTNU	All the violations should be fixed or prevented	Must have	Cyber Security Services	BS4-UC1, BS4-UC2	Initial
42	Forward the agreements (smart contracts) generated in the platform to the blockchain	F	To allow the DELTA blockchain to receive and send information to the rest of the platform	NTNU	The blockchain has received the appropriate smart contracts	Must have	DELTA Cyber Security Services: Smart Contracts/ DELTA Blockchain	BS2-UC3_1, BS2-UC3_2, BS2_UC3_3	Initial
43	Explore transactions on the blockchain	F	To inquire information	NTNU/ CERTH	The BC clients and services must be able to request information from the blockchain regarding completed transactions	Should have	Cyber Security Services (Blockchain Nodes)	BS2-UC3	Initial
44	The cyber security services will send a notification when a threat is discovered or mitigated	F	To enforce security and confront threats	NTNU/ CERTH	Threat mitigation involves the discovery of threats, fixing actions and notifications to the responsible stakeholders	Should have	Cyber Security Services	BS4-UC1, BS4-UC2	Initial

ID	Description	Type	Rationale	Originator	Fit Criterion	Priority	Associated Components	Associated Use Cases	Status
45	Review the typical factors that influence electricity prices	F	To model how the market behaves	JRC	The price profile can be used for price forecasting techniques	Must have	Energy Market Price Forecast	BS1-UC2	Initial
46	Employ price forecasting techniques	F	Predict the energy market prices	JRC	Accurate predictions depending on the participation into the wholesale markets	Must have	Energy Market Price Forecast	BS1-UC2	Initial
47	Predict the portfolio's available flexibility capacity range	F	To allow provide information for the planning of DR events	JRC/ CERTH	Allow accurate predictions of the state of the assets	Must have	DR & Flexibility Forecasting	BS1-UC1, BS1-UC2	Initial
48	Predict occasions that may lead to DR signals' activation in a day-ahead framework	F	To provide possible time intervals for the application of DR signals	JRC/ CERTH	Ensure the effective planning of DR signals	Must have	DR & Flexibility Forecasting	BS1-UC1, BS1-UC2	Initial
49	Establish the energy and/or socio-economic requirements that the DELTA aggregator should take into account for its portfolio management	F	To enable the initial allocation of the DVNs underneath the DELTA aggregator	HIT	The allocated DVNs meet the aggregators requirements	Must Have	DVN, Energy Portfolio Segmentation and classification	BS3-UC1	Initial

ID	Description	Type	Rationale	Originator	Fit Criterion	Priority	Associated Components	Associated Use Cases	Status
50	React and rearrange the DVNs, based on specific clustering indicators	F	To cluster the customers in each DELTA Virtual Node.	HIT	React and rearrange following the orders of the overall Energy Portfolio	Must have	DVN, Consumer/Prosumer Clustering	BS3-UC1	Initial
51	Provide real-time overview of the assets assigned to a specific DVN	F	To allow the Aggregator to supervise each node's flexibility and contextual data	HIT	Produce node profiling for each node that follows the DELTA data model specification	Must Have	Node Flexibility Data Monitoring and Profiling	BS2-UC3_1, BS2-UC3_2, BS2_UC3_3	Initial
52	Compute the DR signals that should be sent to the DELTA Fog Enabled Agents	F	To establish the optimal DR signals to be sent to the DELTA Fog Enabled Agent must fulfill	HIT	DR signals sent to the DELTA Fog Enabled Agent should be translated from the DR signal received form the DELTA aggregator	Must have	Generation/Consumption Optimal Dispatch	BS1-UC1, BS1-UC2, BS2-UC3	Initial
53	Generate smart contracts between the DVN and the DELTA Fog Enabled Agents	F	To establish the compromise	HIT	Smart contracts generated appropriately	Must have	DELTA Smart Contracts	BS4-UC1	Initial

ID	Description	Type	Rationale	Originator	Fit Criterion	Priority	Associated Components	Associated Use Cases	Status
54	To forecast load values of the DELTA Fog Enabled Agents which are used to evaluate the DVNs' flexibility	F	To maximize the availability of the assets in order to fulfil the energy promises established by the DR	HIT	Load forecasting establishes the right course of actions when DR Signals are sent from the DELTA Aggregator/Energy Retailer	Must have	Load Forecasting	BS2-UC3_1, BS2-UC3_2, BS2_UC3_3	Initial
55	Analyses the FEIDs profiling of the underneath DELTA Fog Enabled Agent	F	To provide real-time automated monitoring and control of buildings	HIT	Coordinated management of a building's assets in an energy efficient manner	Must have	Consumer/Prosumer Data Monitoring and Profiling	BS1-UC1, BS2-UC3_1, BS2-UC3_2, BS2_UC3_3	Initial
56	The DVN will monitor the SC's execution and act if the FEID is unable to deliver the energy it was requested by automatically employing/triggering the Inter/Intra Matchmakings process	F	DVNs must take actions when an energy service delivery is about to fail	HIT/ CERTH	Failure of delivery will cost credibility to the prosumers and DVN's and brings penalties for the prosumers	Must have	DVN, FEID, Inter/Intra Matchmaking	BS2-UC3	Updated
57	Implement border line techniques to cluster and group in segments the different DELTA Fog Enabled Agents in DVNs	F	To assign each DELTA Fog Enabled Agent to a different DVN	HIT/ CERTH	Suitable configuration to meet the energetic needs required by the DELTA Aggregator	Must have	Energy Portfolio Segmentation & Classification	BS3-UC1	Initial

ID	Description	Type	Rationale	Originator	Fit Criterion	Priority	Associated Components	Associated Use Cases	Status
58	Auto generate smart contracts between Aggregator and DVNs	F	The agreement between Aggregators and DVNs must be shield with a smart contract containing the terms	HIT/ CERTH	The agreement between DVNs and Aggregator	Must have	Aggregator, DVN	BS2-UC1	Initial
59	Aggregator will digitally sign the smart contract agreement with the DVN	F	To enforce security and non-repudiation and prove authenticity	HIT/ CERTH	The SC agreement must be digitally signed from both sides	Must have	Aggregator, DVN	BS2-UC1	Initial
60	Facilitate the self-balancing process, so as to prevent the loss of energy or stability within the portfolio	F	To be able to control the balance of energy or stability inside the Node	HIT/ CERTH	Ensure balance of energy or stability within the portfolio	Must have	Inter/Intra Node Energy Matchmaking	BS2-UC3_2, BS2-UC3_3, BS3-UC1	Initial
61	Accumulate and evaluate in close to real-time the excess or shortage of energy inside the Node's portfolio	F	To provide accurate and close to real-time evaluation inside the Node	HIT/ CERTH	Achieve close to real-time control inside the Node	Must have	Inter/Intra Node Energy Matchmaking	BS2-UC3_2, BS2-UC3_3, BS3-UC1	Initial
62	Request/offer energy from adjacent Nodes when intra-Node energy matchmaking is not possible	F	To provide effective collaboration among the Nodes	HIT/ CERTH	Achieve coordination among the Nodes	Must have	Inter/Intra Node Energy Matchmaking	BS2-UC3_2, BS2-UC3_3, BS3-UC1	Initial

63	Send an “insufficient resources” signal to the Aggregator in case of not sustained balance	F	To allow the communication with the Aggregator	HIT/ CERTH	Ensure information transmission for the state of the Node	Must have	Inter/Intra Node Energy Matchmaking	BS2-UC3_2, BS2-UC3_3, BS3-UC1	Initial

Table 6. Non-functional requirements descriptions and associations

ID	Description	Type	Rationale	Originator	Fit Criterion	Priority	Associated Components	Associated Use Cases	Status
1	Fast, automatic and accurate identification of grid violations	NF	To restore the steady state promptly	UCY	Setting of bounds for real time and automatic control capabilities	Could have	Grid State Simulation: Grid Stability Simulation Engine	BS3-UC1	New
2	Graphical representation of the Grid Violation location	NF	To allow interactive modelling capabilities	UCY	Representation of the specific asset as close to the actual grid topology as possible	Could have	Grid State Simulation: Grid Stability Simulation Engine	BS3-UC2	New
3	High granularity energy profile data	NF	To range from short term to long term calculations	UCY	Adjustment of the time step in which data are received	Must have	Grid State Simulation: Grid Stability Simulation Engine	BS3-UC1	New
4	Request feedback from occupant regarding its comfort	NF	To ensure that occupants are not negatively affected	UCY	Feedback on occupant’s comfort	Could have	Aggregator: Self Portfolio Energy Balancing/	BS2-UC3	Initial

ID	Description	Type	Rationale	Originator	Fit Criterion	Priority	Associated Components	Associated Use Cases	Status
5	Fast response to a DR depending on the markets time limits	NF	To increase market participation	UCY	Setting of bounds for real time response depending on the market type	Must have	Aggregator: Self Portfolio Energy Balancing	BS2 – UC3	New
6	Respond in real time operation regardless of the incoming DR signals	NF	To reduce investment of time and expertise	UCY	Setting of bound for automated operation and demonstration of automated functioning	Could have	Aggregator: Self Portfolio Energy Balancing	BS2 – UC3	New
7	Conforms to security requirements	NF	To enable a centralized, planned, and safety-oriented management of the entire system's life-cycle	NTNU	The security requirements are outlined by recent standards	Must have	Threat Mitigation – requirement for each component	BS4-UC1, BS4-UC2	Initial
8	Automatically reassign a customer to another cluster/Node when one of the parameters changes	NF	To dynamically update the DVN	HIT	The DVN should have uniform characteristics among the customers	Must have	DVN: Consumer/Prosumer Clustering & Inter/Intra Node Energy Matchmaking	BS3-UC1	Initial

ID	Description	Type	Rationale	Originator	Fit Criterion	Priority	Associated Components	Associated Use Cases	Status
9	Each DVN will report in a predefined interval set by the Aggregator a status report	NF	To provide the Aggregator with energy related information	HIT	The report does not exceed the predefined interval	Must have	DVN	BS2-UC1, BS2-UC2, BS3-UC2, BS4-UC2	Initial
10	Integrate and interoperate in existing energy networks	NF	To relieve Aggregator	CERTH /ITI, EAC/UCY, HIT/KIWI	Tested through iterative scenarios both at lab and real case environment	Must have	All components	All UCs	Initial
11	Ethernet and/or WiFi interface will be implemented in FEID	NF	To be able to connect to LAN networks	CERTH	FEID should be discovered from other devices on the network	Must have	FEID	BS4-UC1, BS4-UC2	Initial
12	Serial interfaces will be implemented in FEID (RS-232, RS-485, SPI, I2C, USB)	NF	To be able to connect and communicate with serial devices	CERTH	Sample read, write and control a serial device	Should have	FEID	BS4-UC1, BS4-UC2	Updated
13	Wireless interfaces will be implemented in FEID (Bluetooth, EnOcean)	NF	To be able to connect and communicate with wireless devices	CERTH	Sample read, write and control a wireless device	Should have	FEID	BS4-UC1, BS4-UC2	Initial

ID	Description	Type	Rationale	Originator	Fit Criterion	Priority	Associated Components	Associated Use Cases	Status
14	Digital interface will be implemented in FEID (Relay control)	NF	To be able to control digital devices	CERTH	Sample read, write and control a digital device	Should have	FEID	BS4-UC1, BS4-UC2	Initial
15	Trusted Platform Module (TPM) will be implemented in FEID	NF	To provide the system with hardware security and crypto functions	CERTH	Defend against software / hardware attacks	Must have	FEID	BS4-UC1, BS4-UC2	Updated

5. Functional View

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 Oracles** are connectors (described as dependencies) that represent other systems, software programs, hardware devices or any other entity that communicates with the system.

5.1 DELTA platform overview

The DELTA platform has been designed on top of several layers as depicted in Figure 9. The different components that comprise the layers and their interactions are the outcome of analysing the requirements established by the use cases and the business requirements, which were reported in the Deliverable D1.5.

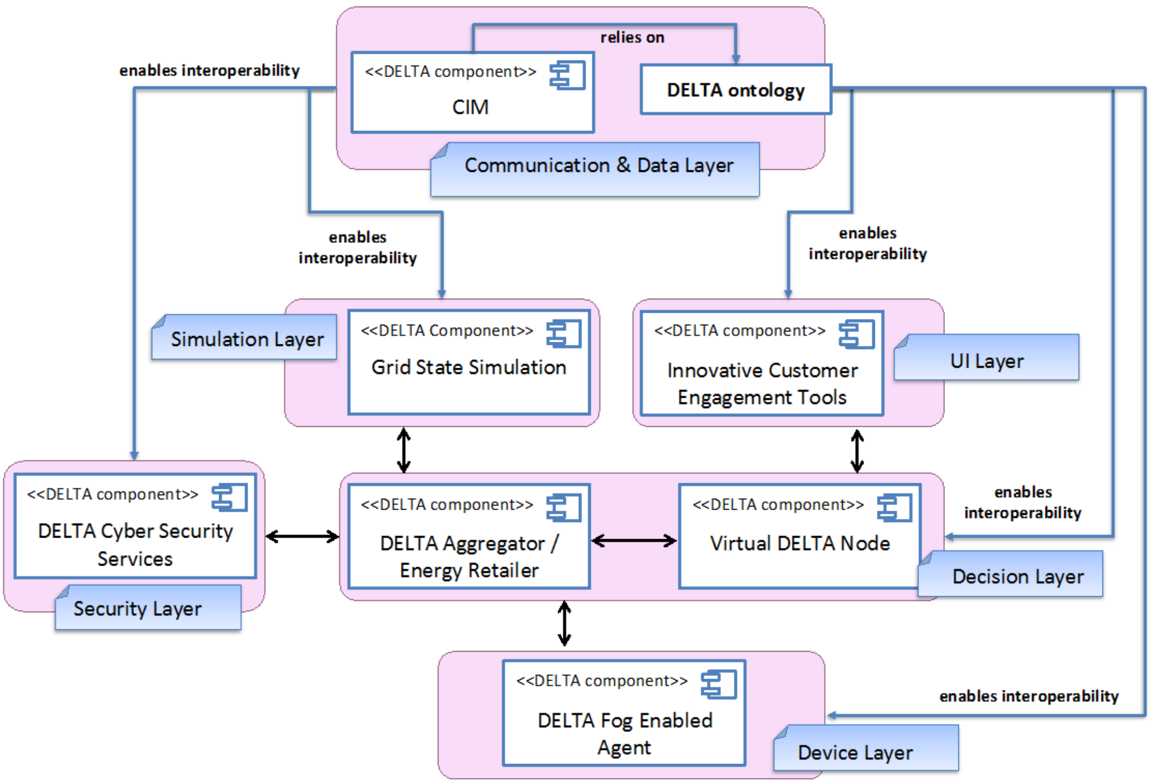

Figure 9. DELTA Architecture Layers

The layers shown in Figure 9 are meant to fulfil the following goals:

- The **UI Layer** consists of a main DELTA component, i.e., Innovative Customer Engagement Tools. It aims at providing end-users mechanisms and services to interact with the DELTA Platform. On the one hand this component is related to the rewards given to end-users based on their performance in the DELTA platform, and, on the other hand, this component allows end-users to visualize an overview of the energy status of their devices and the demand response signals related to them.

- The **Market Layer** consists of an external oracle; in contrast to the rest of the DELTA components, this oracle is out of DELTA's scope. The external oracle represents how the market interacts with the DELTA platform by generating demand response signals that potential DSOs, TSOs, or BRPs could issue to the DELTA platform.

- The **Decision Layer** consists of two main DELTA components, i.e., the DELTA Aggregator / Energy Retailer and the Virtual DELTA Node. It aims at processing the demand response signals provided by potential DSOs, TSOs, or BRPs from the market layer and taking decisions regarding the energy consumption and generation of lower layers, i.e., the Physical Layer. On the one hand, this layer controls, through the Physical Layer, the devices involved in the DELTA platform (by means of control signals) and, on the other hand, by taking the information generated in the platform (forecasted values, current energy consumption, or market energy price) into account, it will take the best decision-course to handle the signals submitted by the DSOs, TSOs, and the BRPs.

- The **Physical Layer** consists of one main DELTA component, i.e., the DELTA Fog Enabled Agent. It aims at controlling the devices involved in the DELTA platform, based on the control signals received by upper layers, e.g., the Decision Layer.

- The **Communication & Data Layer** consists of one main DELTA component, i.e., CIM. On the one hand, this component allows the different main DELTA components to exchange data following the demand response standard OpenADR. On the other hand, this layer specifies the model that data must follow in the whole platform. Some models are international standards such as OpenADR or SAREF, more detailed information about the data models, thee OpenADR standard, and the CIM is given in deliverable D1.7.

- The **Grid State Simulation Layer** consists of the Grid Stability Simulation Engine. This component analyses potential issues at the physical level, concerning voltage line loading fluctuations and offers predictions of physical constraints that may affect the marketplace. The grid stability engine considers the aggregated profiling of a specific *DELTA Aggregator / Energy Retailer* and real-time market prices; then it puts such information against historic conditions and identifies potential physical constraints.

- The **Security Layer** consists of one main DELTA component, i.e., DELTA Cyber Security Services. This component aims at securing the DELTA platform and, at the same time, storing the transactions produced as result of the interaction of different DELTA components. These transactions are stored into a Blockchain, along with Smart Contracts that will be computed as a result of the received transactions.

The following sub-sections provide details and descriptions of the main DELTA components and their sub-components. The descriptions comprise a definition of such components and their sub-components, the interactions between them, and the data interfaces involved; which have being specified and detailed in the DELTA Deliverable D1.7. Consider that the data exchanged in the platform between main components is namely made through the OpenADR standard; on the contrary, communication between two or more sub-components of the same DELTA component follows the REST standard. Several sub-components of the CIM are in charge of implementing the OpenADR technical requirements for the data exchange and data representation.

5.2 Common Information Model (CIM)

The main DELTA component *CIM* consists of several sub-components, namely: *OpenADR FEID CIM, DELTA Repository CIM, Secure Services CIM, OpenADR Aggregator CIM,* and *OpenADR*

Virtual Node CIM. This main DELTA component has several goals, on the one hand it defines the DELTA models that all the data within the platform must follow, on the other hand this component aims at allowing the exchange of information in the DELTA platform. The exchange of data between main DELTA components is performed through the different sub-components that conform the *CIM*, due to architectural decisions, such exchange of information is done following the OpenADR standard. As a result, the sub-components of the *CIM* implement the communication requirements established in the OpenADR profile b. This main DELTA component is depicted in Figure 10 below.

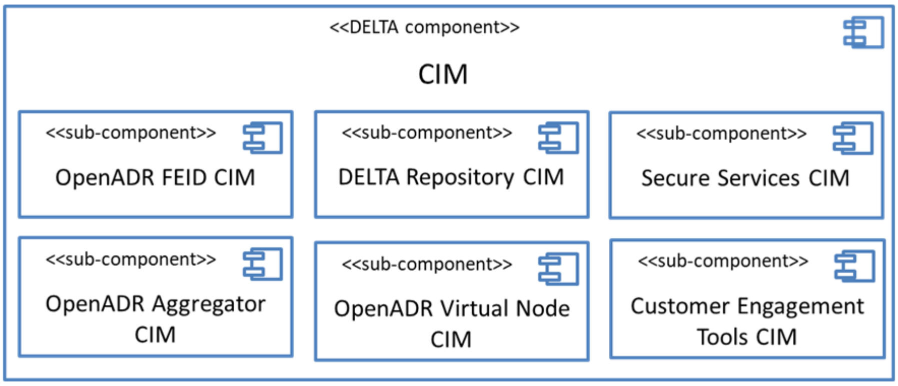

Figure 10. CIM Overview

The sub-components of the *CIM* are used by specific main DELTA components in the architecture to communicate with the rest of the platform, due to this reason and for the sake of simplicity, in the following sub-sections each sub-component of the CIM is explained; nevertheless in following sections these sub-components will appear inside other DELTA main components to simplify the understanding of the data flow.

Notice that the standard OpenADR is implemented by the sub-components of the *CIM*. OpenADR establishes some technological restrictions that must be addressed in the designing of the platform, and its implementation. The higher constraint of this standard applies to the exchange of data, which is done through a P2P network following an XMPP/HTTP protocol. This is the pillar reason to implement the different *CIM* sub-components and distribute them among the main DELTA components so they can interact with each other following the Peer-to-Peer approach. In the case of the DELTA Platform it was chosen to implement the OpenADR standard profile b.

The DELTA data model is a tailored-developed ontology for the DELTA project that describes concepts related to energy market, smart grids, demand response platforms, and the data exchanged among components and sub-components. It will reuse W3C and ETSI standards, such as SAREF ontology, whenever possible. DELTA ontology is also aligned with OpenADR standard. The ontology is available at http://delta.iot.linkeddata.es/index.html.

5.2.1 DELTA Repository CIM

This sub-component of the main DELTA component *CIM* aims at storing global information required by the DELTA platform. On the one hand the *Rewards* earned by end-user in the *Award-enable Energy Behavioural Platform* sub-component are stored for their later consumption. On the other hand, this sub-component has several *KPIs* stored by the DELTA designers and some *Customers Information*; both are required for the well functioning of some DELTA main components. The *KPIs* are key performance indicators that will be used to measure the performance and the behaviour of the different customers that belong to the DELTA platform. The interactions of the *DELTA Repository CIM* sub-component are depicted in Figure 11 below.

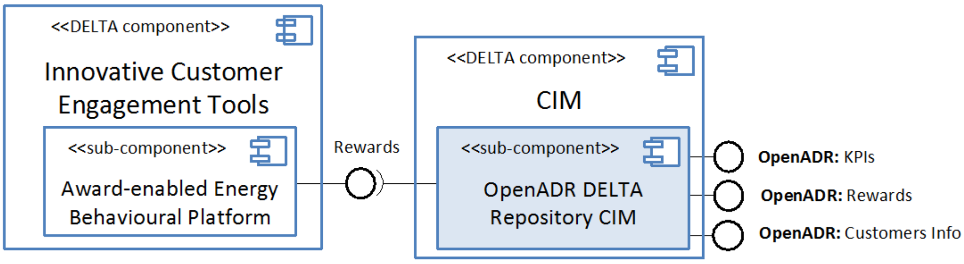

Figure 11. DELTA Repository CIM

As a result, the *DELTA Repository CIM* has the interfaces described in Table 7 to interact with it. Consider that the data models involved in the data exchange through such interfaces are described in Deliverable D1.7.

Table 7. DELTA Repository CIM API

Interface	R/W	Description
OpenADR: KPIs	R/W	This interface exposes all the stored KPIs, nevertheless this interface could be used to write new KPIs in the platform. The KPIs follow the data model specified in DELTA, interacting with this interface requires following the OpenADR specification for communication
OpenADR: Customer Info	R/W	This interface exposes information of customers, at the same time this interface allows writing new information for such customers. The information of the customers must follow the data model specified in DELTA. Interacting with this interface requires following the OpenADR specification for communication
OpenADR: Rewards	R	This interface exposes the rewards earned in the *Award-enable Energy Behavioural Platform*. The rewards follow the DELTA data model specification. In order to read the rewards the OpenADR specification for communication must be followed
Rewards	W	This interface allows only the *Award-enable Energy Behavioural Platform* to write new rewards in order to store them in the DELTA platform. The rewards provided must follow the DELTA data model specification

5.2.2 Secure Services CIM

This sub-component of the main DELTA component *CIM* aims at allowing the storage of all the *Transactions* generated in the platform. The *Secure Services CIM* is not responsible to store this data by itself but to forward it to the proper component, i.e., the *DELTA Blockchain*. The goal of this component is to allow the *DELTA Blockchain* to receive such *Transactions* and at the same time expose the *Smart Contracts* to the rest of the platform; which are computed by the *DELTA Blockchain sub-component*. The interactions of the *Secure Services CIM* sub-component are depicted in Figure 12 below.

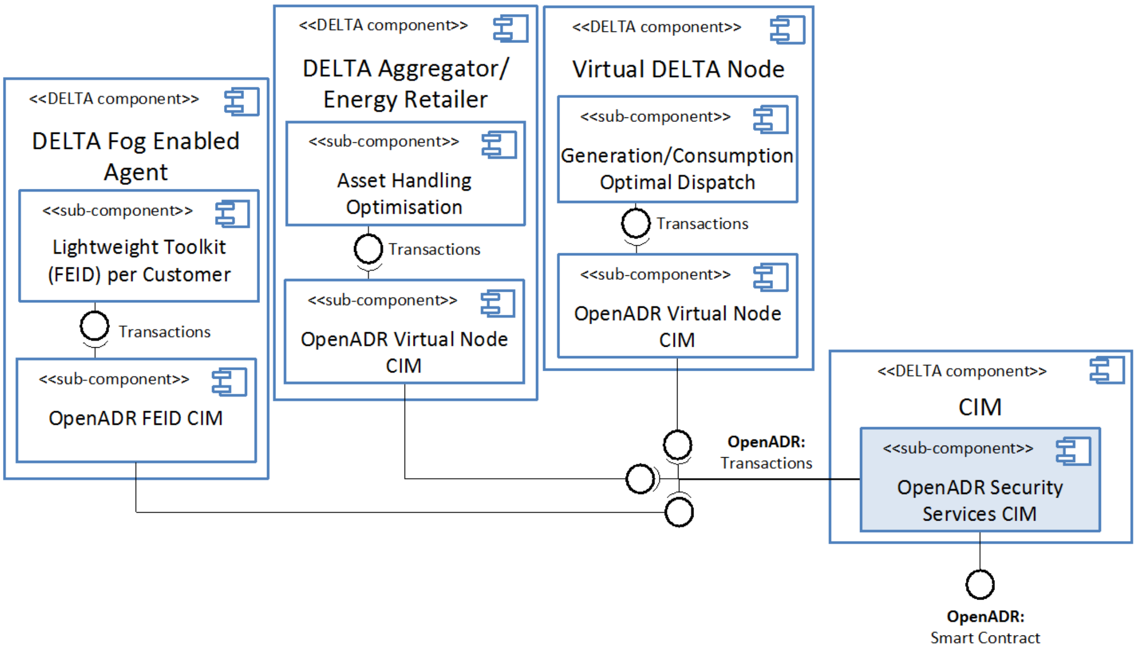

Figure 12. Security Services CIM

The *Security Services CIM* sub-component, as depicted in Figure 12, has only one interface detailed in Table 8.

Table 8. Security Services API

Interface	R/W	Description
OpenADR: Transactions	W	This interface receives the transactions made by the *DELTA Fog Enabled Agent*, *Virtual DELTA Nodes*, and *DELTA Aggregator / Energy Retailer*. These transactions must follow the DELTA data model, and must be sent using the specification of OpenADR for communication.
OpenADR: Smart Contract	R	This interface exposes the smart contracts computed by the *DELTA Blockchain.* Exposed contracts must be accessed following the specification of OpenADR for communication

5.2.3 OpenADR FEID CIM

This sub-component of the *CIM* main DELTA component aims at enabling the communication between the *Virtual DELTA Node* and *the DELTA Fog Enabled Agent*. The goal of the *OpenADR FEID CIM* is to act as a communication bridge that implements the OpenADR standard for communication. The interactions of the *OpenADR FEID CIM* sub-component are depicted in Figure 13 below.

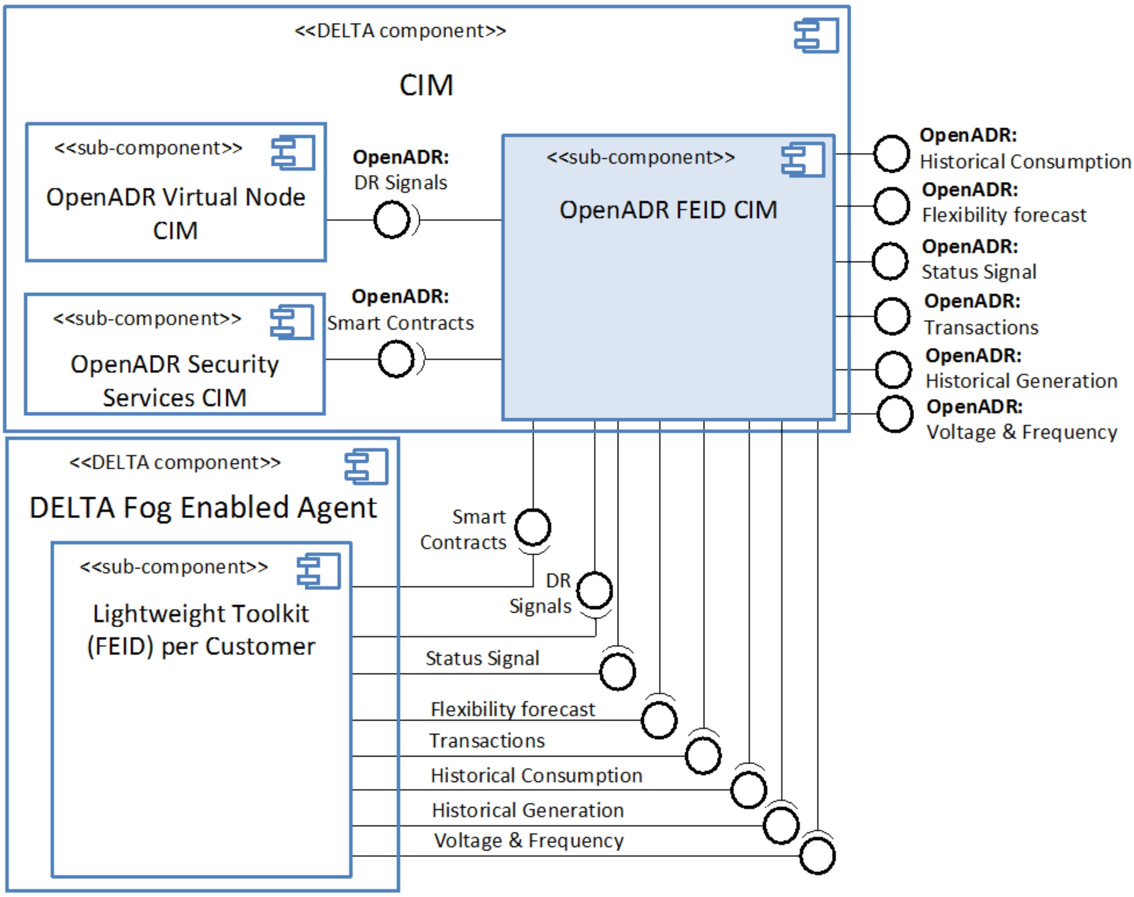

Figure 13. OpenADR FEID CIM

The *OpenADR FEID CIM* has several interfaces that are described in Table 9, the data involved in the interfaces has being explained in Deliverable D1.7.

Table 9. OpenADR FEID CIM API

Interface	R/W	Description
OpenADR: Smart Contracts	W	This interface is meant to receive smart contracts from the *DELTA Blockchain* through the *Security Services CIM*. The contracts must be sent following the OpenADR standard for communication, and modelled using the DELTA data models
OpenADR: DR Signal	W	This interface is meant to receive demand response signals that; which must follow the data models defined DELTA and the communication protocols specified by OpenADR
Flexibility Forecast	W	This interface receives the forecasted flexibility of a given *DELTA Fog Enabled Agent*, the forecasted flexibility follows the DELTA data model specification
Transactions	W	This interface receives the transactions made by a *DELTA Fog Enabled Agent* as a result of interacting with a *Virtual DELTA Node*. The transactions follow the DELTA data model specification
Historical Generation	W	This interface receives the historical generation of a *DELTA Fog Enabled Agent*, the historical generation follows the DELTA data model specification

Historical Consumption	W	This interface receives the historical consumption of a *DELTA Fog Enabled Agent*, the historical consumption follows the DELTA data model specification
Voltage & Frequency		This interface receives the voltage and frequency detected in the *DELTA Fog Enabled Agent*, these measurements follow the DELTA data model specification
Status Signal	W	This interface receives the a signal that describes the real-time status of a *DELTA Fog Enabled Agent*, the status signal follows the DELTA data model specification
OpenADR: Flexibility Forecast	R	This interface exposes the forecasted flexibility received in this component by means of its homonym non-OpenADR interface. This interface adapts the communication protocols to exchange data in order to follow the OpenADR standard
OpenADR: Historical Generation	R	This interface exposes the historical generation received in this component by means of its homonym non-OpenADR interface. This interface adapts the communication protocols to exchange data in order to follow the OpenADR standard
OpenADR: Historical Consumption	R	This interface exposes the historical consumption received in this component by means of its homonym non-OpenADR interface. This interface adapts the communication protocols to exchange data in order to follow the OpenADR standard
OpenADR: Status Signal	R	This interface exposes the status signals received in this component by means of its homonym non-OpenADR interface. This interface adapts the communication protocols to exchange data in order to follow the OpenADR standard
OpenADR: Voltage & Frequency	R	This interface exposes the voltage and frequency measurements received in this component by means of its homonym non-OpenADR interface. This interface adapts the communication protocols to exchange data in order to follow the OpenADR standard
OpenADR: Transactions	R	This interface exposes the transactions received in this component by means of its homonym non-OpenADR interface. This interface adapts the communication protocols to exchange data in order to follow the OpenADR standard
Smart Contract	R	This interface exposes the *Smart Contracts* received to this component from its homonym OpenADR interface. This interface exposes this information locally to a *DELTA Fog Enabled Agent;* without requiring the OpenADR communication protocol
DR Signal	R	This interface exposes the demand response signals received to this component from its homonym OpenADR interface. This interface exposes this information locally to a *DELTA Fog Enabled Agent;* without requiring the OpenADR communication protocol

5.2.4 OpenADR Virtual Node CIM

This sub-component of the *CIM* main DELTA component aims at enabling the communication between the *Virtual DELTA Node* and the *DELTA Aggregator/Energy Retailer and/or* the *DELTA Fog Enabled Agent*. The goal of the *OpenADR Virtual Node CIM* is to act as a communication bridge that implements the OpenADR standard for communication. The interactions of the *OpenADR Virtual Node CIM* sub-component are depicted in Figure 14 below.

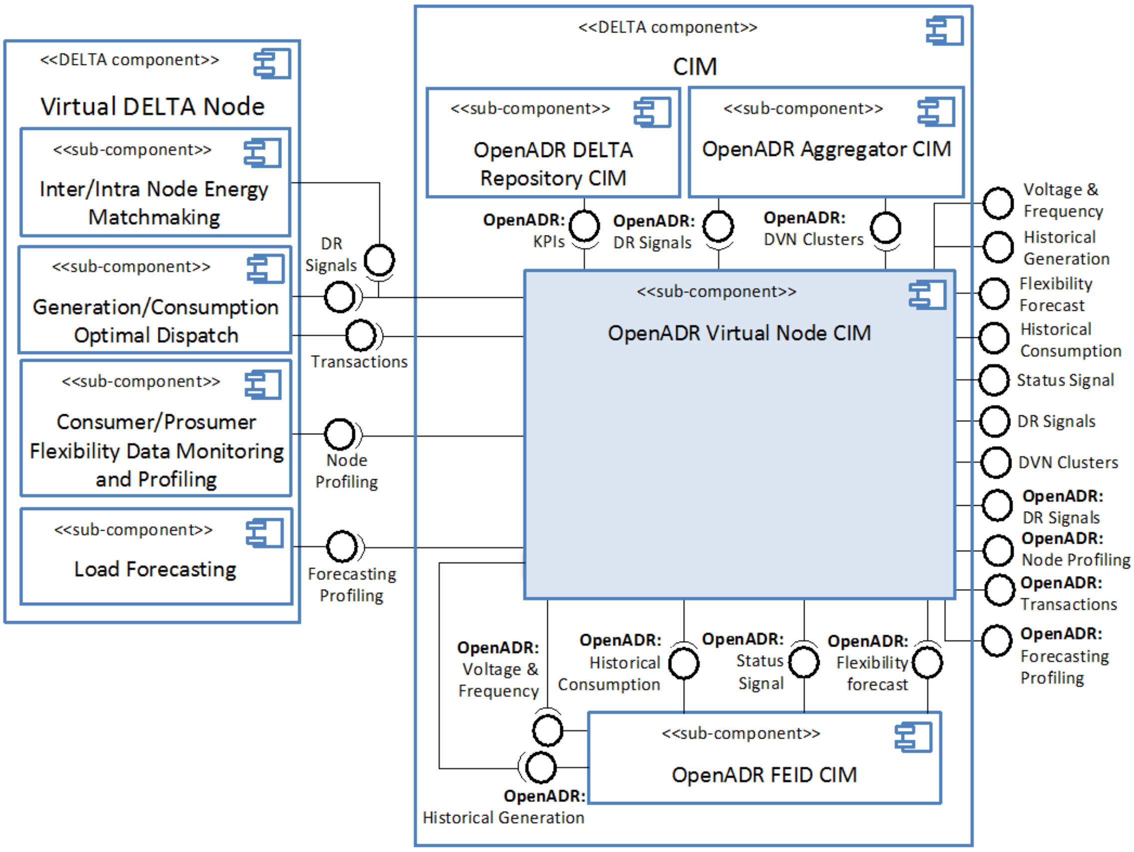

Figure 14. OpenADR Virtual Node CIM

The *OpenADR Virtual Node CIM* has several interfaces that are described in Table 10, the data involved in the interfaces has being explained in Deliverable D1.7.

Table 10. OpenADR Virtual DELTA CIM API

Interface	R/W	Description
OpenADR: Historical Generation	W	This interface receives the historical generation from the *OpenADR FEID CIM*. The data received follows the communication protocol established by the OpenADR standard and the DELTA data model
OpenADR: Voltage & Frequency	W	This interface receives the voltage and frequency measurements from the *OpenADR FEID CIM*. The data received follows the communication protocol established by the OpenADR standard and the DELTA data model
OpenADR: Historical Consumption	W	This interface receives the historical consumption from the *OpenADR FEID CIM*. The data received follows the communication protocol established by the OpenADR standard and the DELTA data model
OpenADR: Status Signal	W	This interface receives the status signals from the *OpenADR FEID CIM*. The data received follows the communication protocol established by the OpenADR

		standard and the DELTA data model
OpenADR: Flexibility forecast	W	This interface receives the flexibility forecast from the *OpenADR FEID CIM*. The data received follows the communication protocol established by the OpenADR standard and the DELTA data model
OpenADR: DR Signals	W	This interface receives the demand response signals from the *OpenADR Aggregator CIM*. The data received follows the communication protocol established by the OpenADR standard and the DELTA data model
OpenADR: DVN Clusters	W	This interface receives the DVN Clusters from the *OpenADR Aggregator CIM*. The data received follows the communication protocol established by the OpenADR standard and the DELTA data model
OpenADR: KPIs	W	This interface receives the KPIs from the *OpenADR Repository CIM*. The data received follows the communication protocol established by the OpenADR standard and the DELTA data model
Transactions	W	This interface receives the transactions that a *Virtual DELTA Node* has performed with one or more *DELTA Fog Enabled Agent*, or one or more *DELTA Aggregator / Energy Retailer*, or one or more *Virtual DELTA Node*. These transactions follow the DELTA data model
DR Signals	W	This interface receives demand response signals that a *Virtual DELTA Node* may compute, such signals must follow the DELTA data model
Node Profiling	W	This interface receives the Node Profiling. The profiling follows the DELTA data models
Flexibility Forecast	R	This interface exposes the flexibility forecast received in this component by means of its homonym OpenADR interface. This interface exposes locally to a *Virtual DELTA Node* the flexibility and load monitoring sent by a *DELTA Fog Enabled Agent*; which must follow the DELTA data model
Historical Generation	R	This interface exposes the historical generation received in this component by means of its homonym OpenADR interface. This interface exposes locally to a *Virtual DELTA Node* the historical data sent by a *DELTA Fog Enabled Agent*; which must follow the DELTA data model
Historical Consumption	R	This interface exposes the historical consumption received in this component by means of its homonym OpenADR interface. This interface exposes locally to a *Virtual DELTA Node* the historical data sent by a *DELTA Fog Enabled Agent*; which must follow the DELTA data model
Voltage & Frequency	R	This interface exposes the voltage and frequency received in this component by means of its homonym OpenADR interface. This interface exposes locally to a *Virtual DELTA Node* these measurements sent by one or more *DELTA Fog Enabled Agents*; which must follow the DELTA data model
Status Signal	R	This interface exposes the status signal received in this

		component by means of its homonym OpenADR interface. This interface exposes locally to a *Virtual DELTA Node* the status signal sent by a *DELTA Fog Enabled Agent*; which must follow the DELTA data model
DR Signals	R	This interface exposes locally to a *Virtual DELTA Node* the demand response signals generated by the same *Virtual DELTA Node*; such signals must follow the DELTA data model
DVN Clusters	R	This interface exposes locally to a *Virtual DELTA Node* the DVN clusters computed by the same *Virtual DELTA Node*; such clusters must follow the DELTA data model
OpenADR: DR Signal	R	This interface exposes demand response signals computed by a *Virtual DELTA Node.* These signals follow the DELTA data model and the OpenADR communication protocol specification
OpenADR: Transactions	R	This interface exposes the transactions received in this component by means of its homonym non-OpenADR interface. The exposed transactions follow the OpenADR communication protocol and the DELTA data model specification
OpenADR: Forecasting Profiling	R	This interface exposes the forecasted profiling received in this component by means of its homonym non-OpenADR interface. The exposed forecasted profiling follows the OpenADR communication protocol and the DELTA data model specification
OpenADR: Node Profiling	R	This interface exposes the node profiling received in this component by means of its homonym non-OpenADR interface. The exposed profiling follows the OpenADR communication protocol and the DELTA data model specification

5.2.5 OpenADR Aggregator CIM

This sub-component of the *CIM* main DELTA component aims at enabling the communication of the *DELTA Aggregator/Energy Retailer* between the *Virtual DELTA Node* and the external oracle *DR Services to Market Stakeholders*. The goal of the *OpenADR Aggregator CIM* is to act as a communication bridge that implements the OpenADR standard. The interactions of the *OpenADR Aggregator CIM* sub-component are depicted in the Figure 15 below.

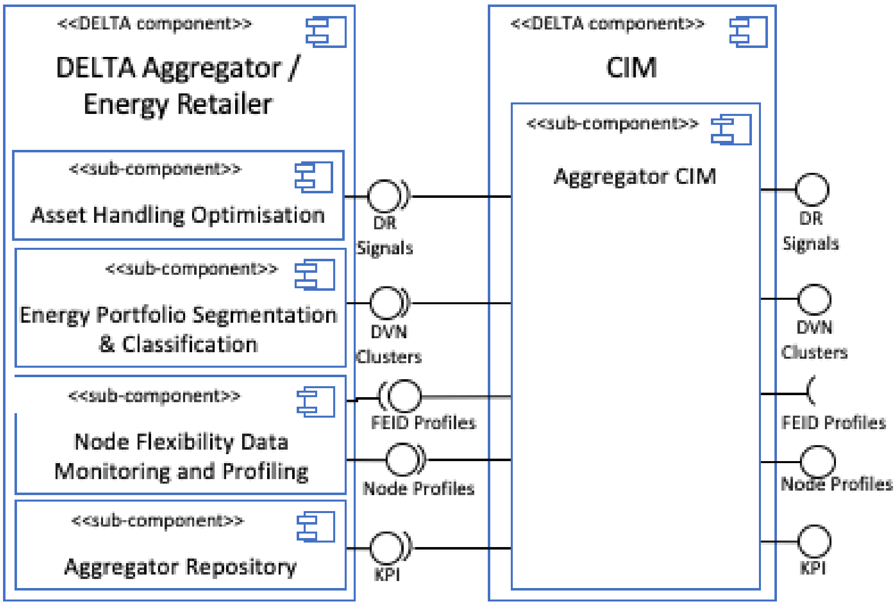

Figure 15. OpenADR Aggregator CIM

The *OpenADR Aggregator CIM* has several interfaces that are described in Table 11, the data involved in the interfaces has being explained in Deliverable D1.7.

Table 11. OpenADR Aggregator CIM API

Interface	R/W	Description
Node Profiles	W	This interface receives the node profiling from the *DELTA Aggregator / Energy Retailer*. The profiling follows the DELTA data model
DR Signals	W	This interface receives demand response signals. The provider of these signals is the market layer, i.e., the external oracle
DVN Clusters	W	This interface receives a set of clusters from the *DELTA Aggregator / Energy Retailer*. The clusters follow the Delta data model
DR Signals	W	This interface receives demand response signals from the *DELTA Aggregator / Energy Retailer*. These signals follow the DELTA data model
FEID profiles	W	This interface receives the FEID profiles from the *DELTA Aggregator / Energy Retailer*. The profiling follows the DELTA data model
Node Profiles	R	This interface exposes the node profiling received in this component. This interface exposes locally to a *DELTA Aggregator / Energy Retailer* the profiling; which must follow the DELTA data model
DR Signals	R	This interface exposes the demand response signals received in this component. This interface exposes locally to a *DELTA Aggregator / Energy Retailer* the signals sent by a potential DSO from the market layer; which must follow the DELTA data model
DR Signals	R	This interface exposes the demand response signals received in this component by means of its homonym

		non-OpenADR interface. The signals are computed by a *DELTA Aggregator / Energy Retailer* and sent to the *Virtual DELTA Nodes* through this interface following the OpenADR communication protocol
DVN Clusters	R	This interface exposes the clusters received in this component. The clusters are computed by a *DELTA Aggregator / Energy Retailer* and sent to the *Virtual DELTA Nodes* through this interface
FEID profiles	R	This interface exposes the aggregated profiling computed by a *DELTA Aggregator / Energy Retailer*, and received in this component. The profiling follow the DELTA data model.

5.3 Innovative Customer Engagement Tools

The main DELTA component *Innovative Customer Engagement Tools* consists of several sub-components, namely: *Demand Response Visualization Kit*, *Award-enabled Energy Behavioural Platform,* and the *Social Interaction and Cooperation*. This main component is the end-user access to the DELTA platform. Figure 16 below shows the different sub-components that this main DELTA component consist of; in following sub-sections each of these sub-components are explained in detail.

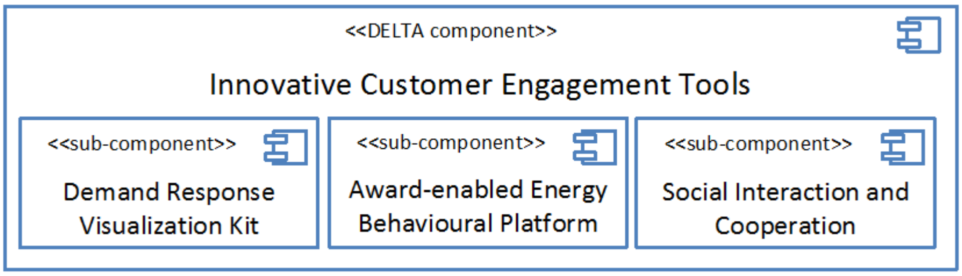

Figure 16. Innovative Customer Engagement Tools Overview

5.3.1 Demand Response Visualisation Kit

This sub-component of the main DELTA component *Innovative Customer Engagement Tools* aims at providing to the end-users an overview of the real-time data related to their physical devices. This sub-component is a web-base user interface that shows the demand response visualizations along with other visual analytics information. The *Demand Response Visualisation Kit* retrieves information from the *DELTA Aggregator / Energy Retailers* and the *Virtual DELTA Nodes* that are deployed in the DELTA platform.

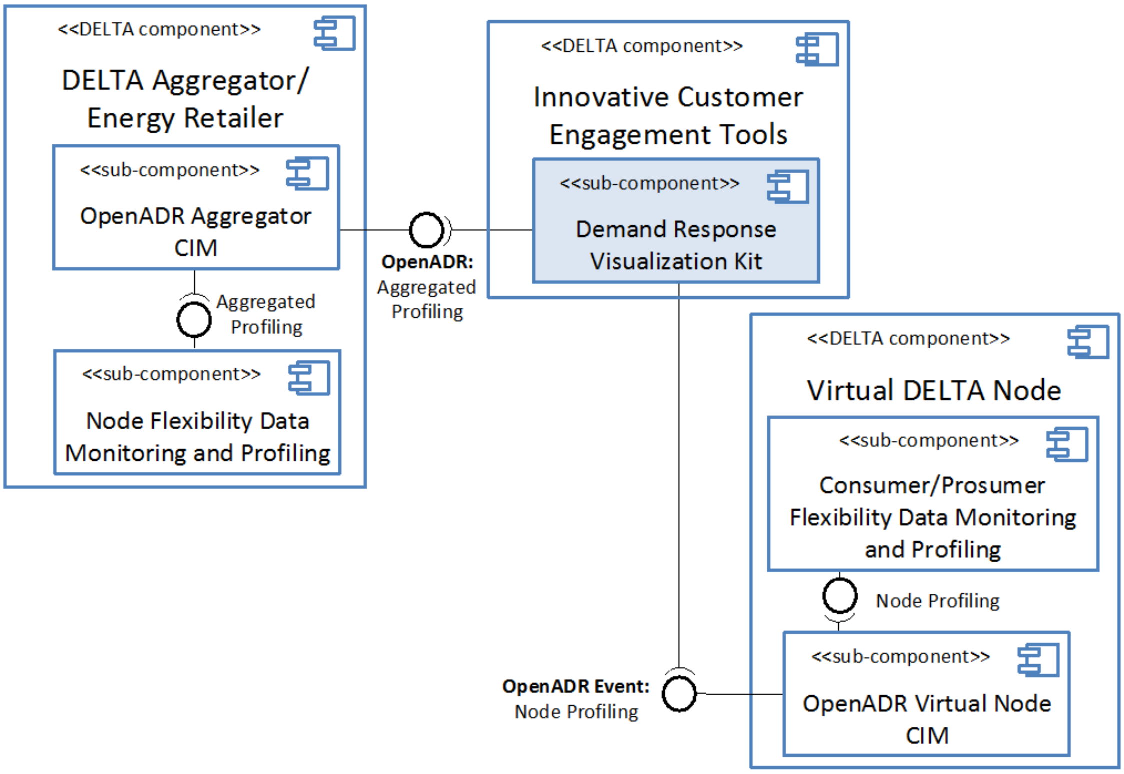

Figure 17. Demand Response Visualization Kit

The *Historical Data* and the *Real-time Visualization Data* with which the *Demand Response Visualization Kit* if fed comprises: real-time and historical energy consumption, energy generation, energy flexibility, emissions, demand response strategies, demand response cost estimations, demand response energy and emissions estimations, and finally financial, environmental and stability information; Figure 17 depicts this interaction. The DELTA components involved in the *Real-time Visualization Data* are the ones containing data related to the end-user that owns the visualization, namely the *DELTA Aggregator/Energy Retailer*, the *Virtual DELTA Node*, and the *DELTA Fog Enabled Agent*. Due to the DELTA architecture the information of the *DELTA Fog Enabled Agent* can be retrieved at the level of the Virtual *DELTA Node*. Due to this reason the *Demand Response Visualization Kit* retrieves the required information from the *DELTA Aggregator/Energy Retailer* through their *OpenADR Aggregator CIM* that exposes the *Aggregated Profiling*, and from the *Virtual DELTA Node* though the *OpenADR Virtual Node CIM* that exposes the *Node Profiling*.

The *Demand Response Visualization Kit* has an interface that is explained in Table 12, this interface gathers data and does not expose any since, as we have already explained, it aims at providing an overview of the data in the platform.

Table 12. Demand Response Visualization Kit API

Interface	R/W	Description
OpenADR: Aggregated Profiling	W	This interface receives the aggregated profiling from the *OpenADR Aggregator CIM* following the OpenADR communication protocol
OpenADR: Node Profiling	W	This interface receives the node profiling from the *OpenADR Aggregator CIM* following the OpenADR communication protocol

5.3.2 Award-enabled Energy Behavioural Platform

This sub-component of the *Innovative Customer Engagement Tools* main DELTA component aims at offering to end-users a game in the context of the energy demand response environment. This sub-component grants end-users with rewards based on their participation in the game. The rewards granted are considered by other DELTA components and sub-components to bestow special treatments in terms of demand response to the devices belonging to the awardee end-users.

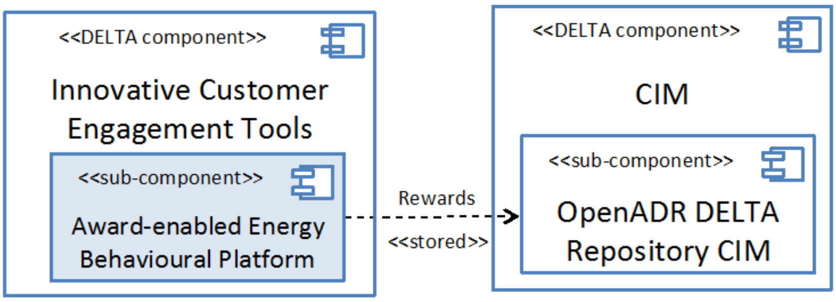

Figure 18. Award-enabled Energy Behavioural Platform

Figure 18 depicts the interaction of the *Award-enable Energy behavioural Platform* with the rest of the platform. When an end-user is grated with a reward, such event is stored in *DELTA Repository* so other DELTA components and sub-components can rely on such information to work properly, specifically the *Inter/Intra Node Energy Matchmaking* sub-component and the *Energy Portfolio Segmentation & Classification* sub-component belonging to the *DELTA Aggregator/Energy Retailer* and the *Virtual DELTA Node*, respectively. The granted rewards are sent to the *DELTA Repository CIM* where they are stored for their later consumption.

The *Award-enable Energy behavioural Platform* does not have a specific API interface since, as described before, when an award is granted it is immediately sent to the *DELTA CIM Repository*. Nevertheless the granted rewards are queriable and can be fetched using the *DELTA CIM Repository*, as explained in the section of this document dedicated to this DELTA component.

5.3.3 Social Interaction and Cooperation

This sub-component of the *Innovative Customer Engagement Tools* main DELTA component aims at providing end-users with a platform that offers a large portfolio of useful data and features, namely: best tailored-practices to follow, suggestions and incentives, a social network, activities related to the DELTA platform, Q&A, chats, notifications of social activities, and content posting. This sub-component allows end-users to interact among them and the platform, but is also related to the other two sub-components *Award-enable Energy behavioural Platform* and the *Demand Response Visualisation Kit* allowing users to check their rewards and visualize the information related to their devices involved in the DELTA platform.

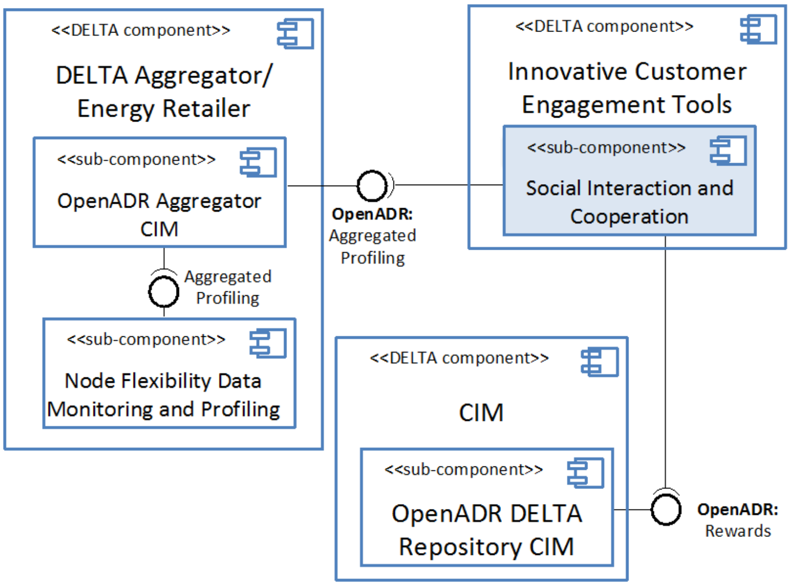

Figure 19. Social Interaction and Cooperation

Figure 19 shows how the *Innovative Customer Engagement Tools* is informed using data fetched from the DELTA platform, namely the data related to the *Award-enable Energy behavioural Platform* and the *Demand Response Visualisation Kit, i.e.*, the *Rewards* and the *Real-time Visualization Data*. Notice that all the fetched data is retrieved through the *OpenADR Aggregator CIM*, which implements the OpenADR protocol for communicating.

The *Innovative Customer Engagement Tools* does not have a specific API interface to expose data since this sub-component is meant for end-users, and thus, it does not generate relevant information for the DELTA platform, only for other end-users. Nevertheless, this sub-component has several interfaces to gather data described in Table 13

Table 13. Social Interaction and Cooperation API

9. Interface	R/W	Description
Rewards	W	This interface receives rewards following the communication protocol of OpenADR. The profiling must follow the DELTA data model
OpenADR: Aggregated Profiling	W	This interface receives aggregated profiling following the communication protocol of OpenADR. The profiling must follow the DELTA data model

5.4 Demand Response Services to Market Stakeholders

The component *Demand Response Services to Market Stakeholders* consists of an external oracle. This component represents the potential DSOs, TSOs or BRPs that would be able to interact with the DELTA platform by sending demand response signals to the main component *DELTA Aggregator/Energy Retailers*; which will process these requests and will respond with computed *Bids*. The interaction among these components will be transparent thanks to the OpenADR standard; nevertheless the transactions made between them will not be written on the *DELTA Blockchain*. Figure 20 below shows the interaction of this external oracle with the *DELTA Aggregator/Energy Retailers* component in the DELTA platform.

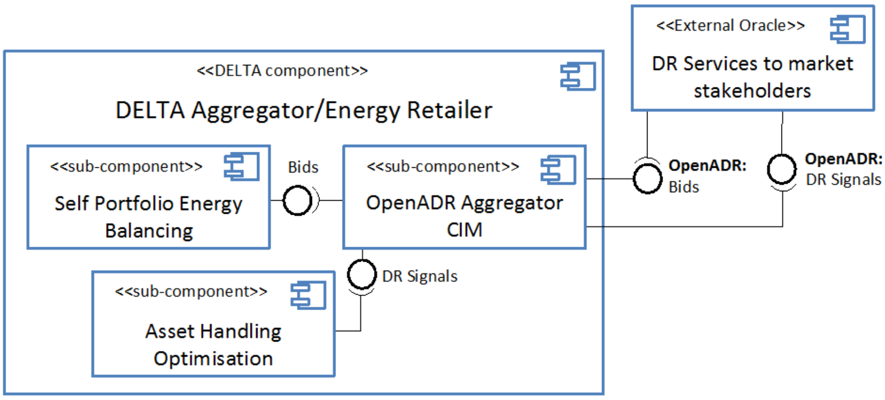

Figure 20. DR Services to market stakeholders

The external oracle *Demand Response Services to Market Stakeholders* does not have a specific API since is out of the scope of the DELTA platform, nevertheless, it submits demand response request to the DELTA platform relaying on the OpenADR standard. The communication between the *Demand Response Services to Market Stakeholders* and the *DELTA Aggregator/Energy Retailer* is done through the *OpenADR Aggregator CIM*, to ensure that the transactions among such components follow the OpenADR standard.

5.5 DELTA Cyber Security Services

The main DELTA component *DELTA Cyber Security Services* consists of two sub-components, namely: the *DELTA Blockchain*, and the *Energy Data Threat Mitigation*. On one hand, this main component aims at keeping track of the transactions made in DELTA in order to ensure the transparency of the DELTA platform; on the other hand, this component aims at actively monitoring the DELTA platform to identify and solve potential security risks and threats. Figure 21 below shows the sub-components that conform the *DELTA Cyber Security Services* component.

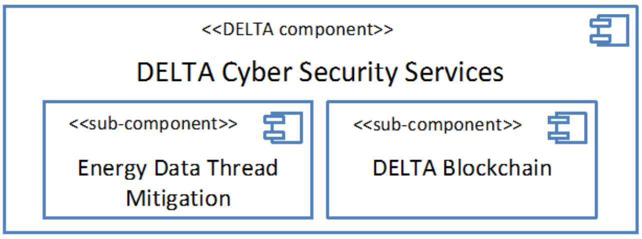

Figure 21. DELTA Cyber Security Services Overview

The main DELTA component *DELTA Cyber Security Services* is not mandatory to have an API since it intends to keep track of normal operations and detect any abnormalities. This component monitors the *Transactions* submitted by the *DELTA Aggregator/Energy Retailers*, the *Virtual DELTA Nodes*, and the *DELTA Fog Enabled Agents*. The *DELTA Blockchain* stores those transactions on a private blockchain. Even though this component might not have an API, the blockchain ledger is available to all stakeholders through their lightweight client applications and can be queried. Thus, the *Transactions* created on the DELTA platform are transparent to the network's participants and can be retrieved, as explained in the *CIM* section. In addition this component is responsible to create and deploy dynamic *Smart Contracts*, which will derive from the DR calls and *local policies and agreements*.

5.5.1 Energy Data Threat Mitigation

This sub-component of the *DELTA Cyber Security Service* main DELTA component aims at monitoring two main DELTA components, namely: *DELTA Aggregator/Energy Retailer* and *Virtual DELTA Node*. As depicted in Figure 22 below, the constant monitor aims at identifying and preventing security breaches, or in case of a security violation starting the necessary actions to fix the issue. In absence of such actions, an alert must be send. In addition, this sub-component will prevent, when possible, potential attacks to the DELTA Platform.

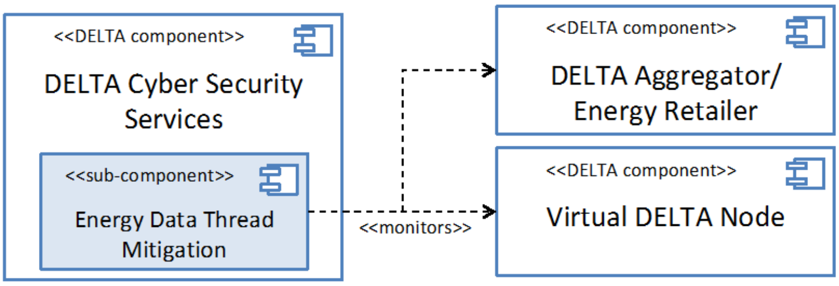

Figure 22. Energy Data Thread Mitigation

The role of this sub-component is passive and therefore it has no API from which data can be retrieved. However, it must be capable to produce alerts and notify the appropriate component's stakeholders to take actions.

5.5.2 DELTA Blockchain

This sub-component of the *DELTA Cyber Security Service* main DELTA component aims at storing the *Transactions* made between the main DELTA components, namely: *DELTA Aggregator/Energy Retailer*, *Virtual DELTA Node*, and *DELTA Fog Enabled Agent*. The *Transactions* made among these components, which are specified and explained in the DELTA Deliverable D1.7, are stored on the blockchain through this sub-component. As shown in Figure 23 below, these *Transactions* are mediated by the *CIM* for every DELTA component that is able to submit *Transactions*. In addition, new *Smart Contracts* are dynamically formed and deployed on the DELTA Blockchain, deriving from DELTA components' interactions. These *Smart Contracts* are stored on the Blockchain and made instantly available to the stakeholders to call and initiate, following CIM compliant messages. Any intermediate and/or final results produced by *Smart Contracts* are also logged on the blockchain and made available to be queried through the platform.

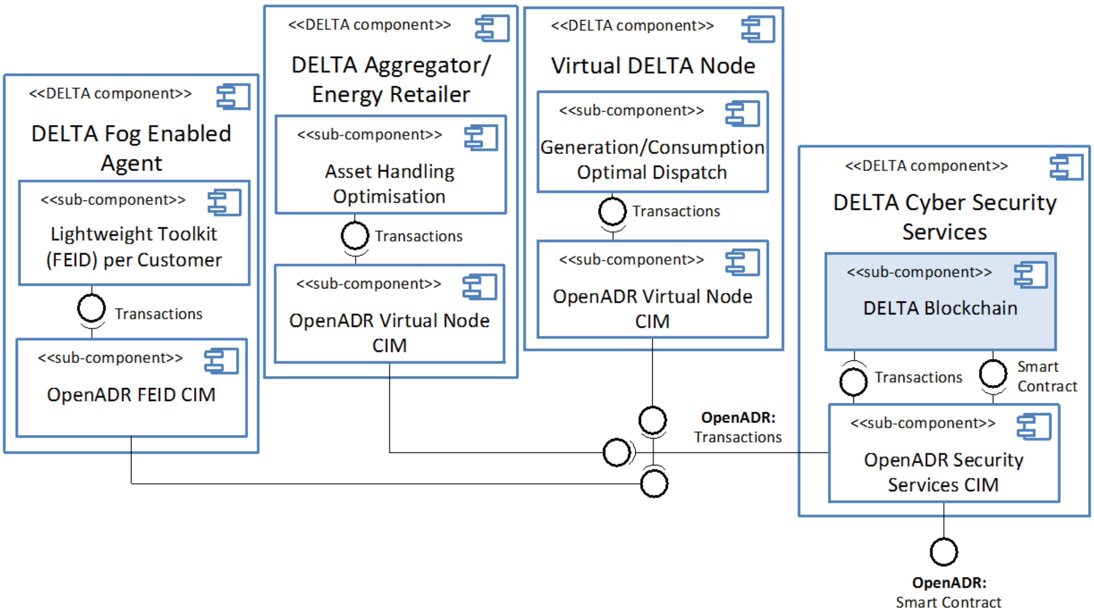

Figure 23. DELTA Blockchain

The *DELTA Blockchain* has two interfaces that allow components with permissions to interact with it. All communications are tunnelled through the *OpenADR Security Services CIM*, due to the OpenADR technology security advantages. Table 14 describes the *DELTA Blockchain* interfaces.

Table 14. DELTA Blockchain API

Interface	R/W	Description
Transactions	W	This interface receives the transactions submitted by other main DELTA components; the transactions' DR commands follow the DELTA data model
Smart Contract	R	This interface exposes the smart contracts formed by the *DELTA Blockchain*. The contracts' DR information follows the DELTA data model

5.6 Grid State Simulation

This component consists of the components needed for Grid state simulation. The goal of this component is to run simulations of the grid state and analyse deviations that might impact the normal behaviour of the grid. To fulfil its purpose the *DELTA Grid State Simulation* is built upon several sub-components depicted in Figure 25. namely*: GSSE CIM, Grid Stability Simulation Engine, and Local Repository*.

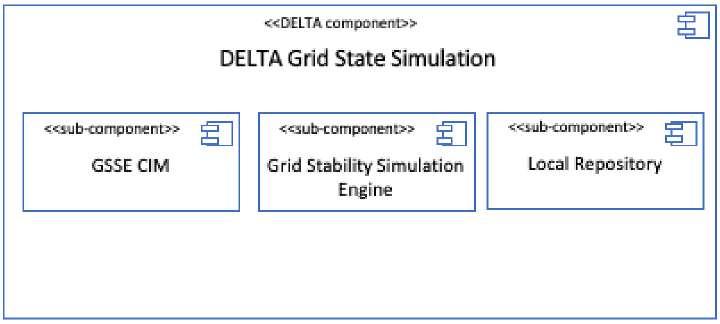

Figure 24. DELTA Grid State Simulation

5.6.1 Grid Stability Simulation Engine

This component analyses potential issues at the physical level, concerning voltage and line loading fluctuations and offers predictions of physical constraints that may affect the marketplace. The extent to which the component is capable of accurate predictions is dependent on the amount of data available. The grid stability engine considers the aggregated profiling of a specific *DELTA Aggregator / Energy Retailer* and real-time market prices; then it puts such information against historic conditions and identifies potential physical constraints. Figure 25 below depicts the *Grid Stability Simulation Engine.*

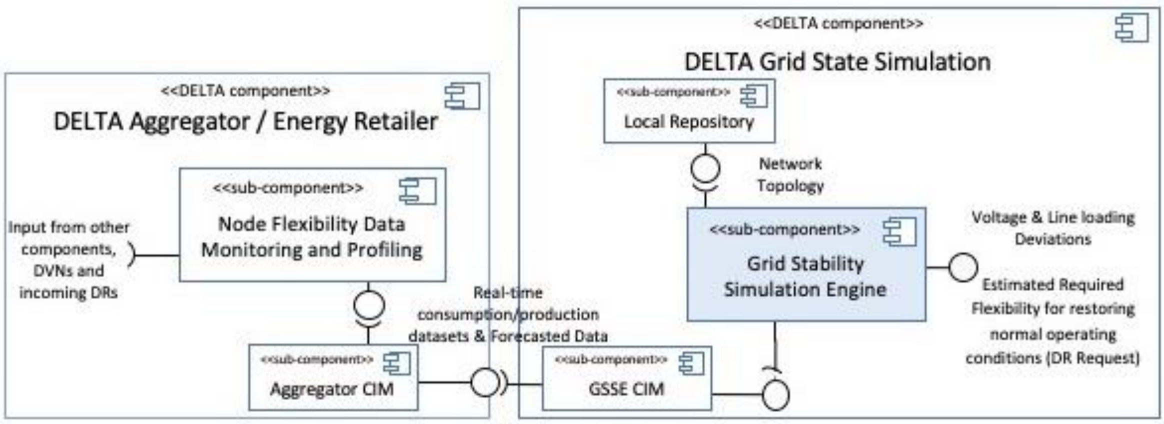

Figure 25. Grid Stability Simulation Engine

The *Grid Stability Simulation Engine* has namely four interfaces that are described in Table 15, the data involved in the interfaces has being explained in Deliverable D1.7.

Table 15. Grid Stability Simulation Engine API

Interface	R/W	Description
Real - time consumption/production datasets and Forecasted data	W	This interface allows the *Grid Stability Simulation Engine* to receive real and forecasted data from the *Node Flexibility Data Monitoring and Profiling* through the respective CIM models. This data follows the DELTA data model specification
Network topology	W	This interface allows the *Grid Stability Simulation Engine* to receive the network topology from the *Local repository*. This data follows the DELTA data model specification
Voltage & Line Loading Deviations	R	This interface exposes the voltage and line loading deviations computed by the *Grid Stability Simulation Engine*. The voltage and line loading constraints follow the DELTA data model
Estimated required flexibility	R	This interface exposes the estimated required flexibility computed by the *Grid Stability Simulation Engine.* The estimated required flexibility follow the DELTA data model

5.7 DELTA Aggregator / Energy Retailer

This component represents real-world Energy Aggregators and Energy Retailers. On the one hand the goal of this component is to assist such entities in their decision-making mechanisms over their portfolios of prosumers. On the other hand, this component gathers the relevant information of its *Virtual DELTA Nodes* underneath. To fulfil its purpose the *DELTA Aggregator / Energy Retailer* is built upon several sub-components depicted in Figure 26 below, namely*: Decision Support System, DR & Flexibility Forecasting, Energy Market Price forecast*, *Energy Portfolio Segmentation & Classification, and Self Portfolio Energy Balancing*. In following sub-sections each of this sub-components is explained in detail.

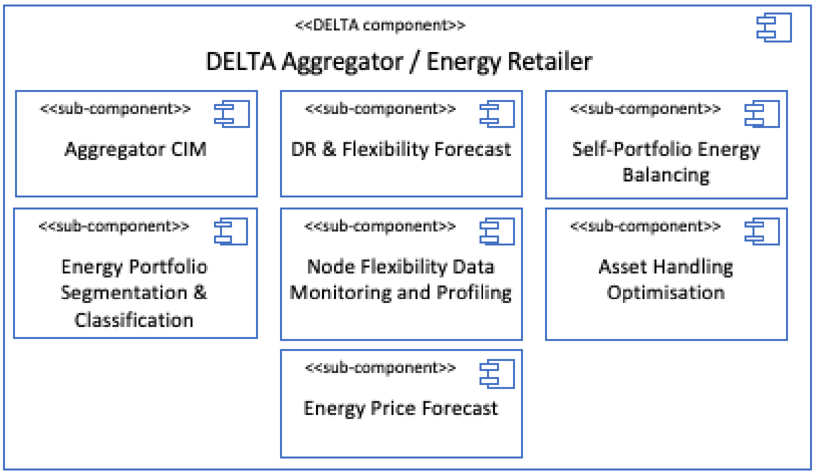

Figure 26. DELTA Aggregator / Energy Retailer Overview

5.7.1 Self Portfolio Energy Balancing

The Self Portfolio Energy Balancing aims at considering the energy market behaviour into account to produce assessments in the *DELTA Aggregator / Energy Retailer* portfolio. Its goal is twofold, on the one hand it generates *Bids* that are sent the external oracle *Demand Response Services to Market Stakeholders* as answer of a previous submitted demand response signal. On the other hand, this sub-component aims at identifying strategies for market operation that specifies a certain behaviour that the *DELTA Aggregator / Energy Retailer* should consider to take, i.e., adapting its assets to the market status.

To achieve its goal the Self Portfolio Energy Balancing relies in the output of the *DR & Flexibility Forecasting* through the interface *Forecasted Flexibility*, the output of the *Energy Market Price Forecast* thorough the interface *Energy Price Profile*, and the *Aggregator Profiling* provided by the *Node Flexibility Data Monitoring and Profiling.* Figure 27 below depicts these interactions.

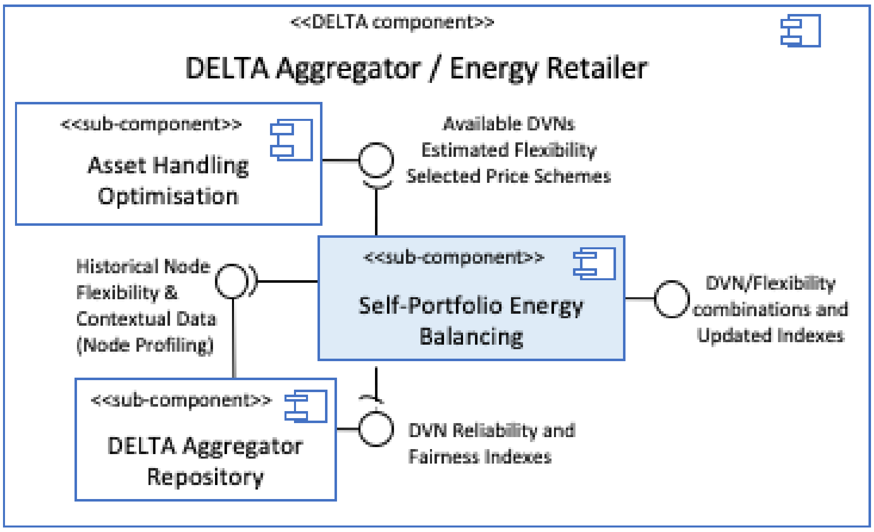

Figure 27. Self Portfolio Energy Balancing

The interfaces the *Self Portfolio Energy Balancing* are described in Table 16. The data handled by the interfaces is described in detail in the DELTA deliverable D1.7.

Table 16. Self Portfolio Energy Balancing API

Interface	R/W	Description
Available DVNs, Estimated Flexibility, Selected Price Schemes	W	This interface receives the Available DVNs, Estimated Flexibility, Selected Price Schemes provided locally by the *Asset Handling Optimisation*. The data follow the DELTA data model specification
Historical Node Flexibility & contextual Data (Node Profiling)	W	This interface receives the node profiling provided locally by the *DELTA Aggregator Repository*. The node profiling follows the DELTA data model
DVN Reliability and Fairness Indexes	W	This interface receives the reliability and fairness indexes located on the *DELTA Aggregator Repository*. The indexes follow the DELTA data model

5.7.2 Energy Portfolio Segmentation & Classification

The *Energy Portfolio Segmentation & Classification* aims at establishing the allocation of the *Virtual DELTA Nodes* underneath the *DELTA Aggregator / Energy Retailer*. The cluster is based on an algorithm that computes the energetic requirements that a *DELTA Aggregator / Energy Retailer* must fulfil, and, in base of those, it allocates the required *Virtual DELTA Nodes* as assets in order to meet such requirements. The *Virtual DELTA Nodes* will re-act and re-arrange themselves autonomously based on indicators, following the orders of the *Energy Portfolio Segmentation & Classification.*

The clustering algorithm runs taking into consideration the following events: i) when there is a change at the Aggregator’s portfolio (addition of new customers and removal of customers that leave its portfolio (allocation of smart contracts etc.); ii) when the Aggregator changes its business model (e.g. decides that self-balancing is more beneficial than DR services provided to the market); iii) after several (to-be defined) attempts of inter-matchmaking between DVNs.

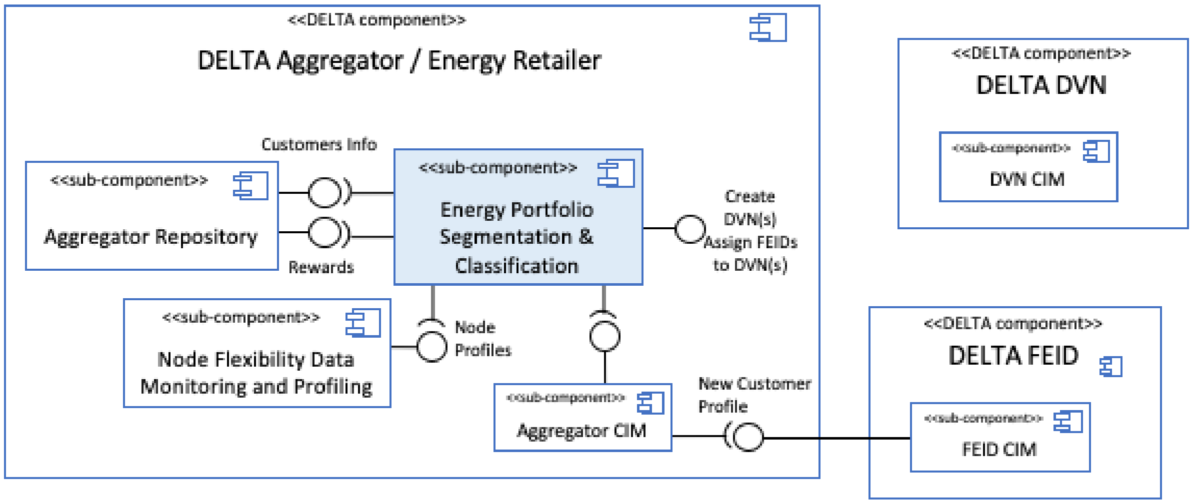

Figure 28. Energy Portfolio Segmentation & Classification

The *Energy Portfolio Segmentation & Classification* counts with the interfaces specified in Table 17, the data involved in the data exchange follows the models specified in Deliverable D1.7.

Table 17. Energy Portfolio Segmentation & Classification API

Interface	R/W	Description
Rewards	W	This interface receives the rewards stored in the *Aggregator Repository*, such rewards follow the DELTA data model specification
Node Profiles	W	This interface receives the aggregated profiling from the Node Flexibility Data Monitoring and Profiling, which follows the DELTA data model specification
Customers Info	W	This interface receives the information of customers stored in the *Aggregator Repository*, which follow the DELTA data model specification
New Customer Profile	W	This interface receives the information of new customers, which follow the DELTA data model specification
Create DVN(s) assign FEIDS to DVN(s)	R	This interface exposes groups of *DELTA Virtual Nodes* that a certain *DELTA Aggregator / Energy Retailer* is in charger of. The clusters follow the DELTA data model specification

5.7.3 Energy Market Price Forecast

The *Energy Market Price Forecast* aims at providing decision support on where to allocate existing flexibility according to energy market prices, in order to maximize the aggregators revenue. The goal of this sub-component is to allow the DELTA platform to include this kind of information in its decision-taking process. The component will analyse likely price signals in the day ahead and intraday markets. The interactions required to compute this forecast are depicted in Figure 29 below.

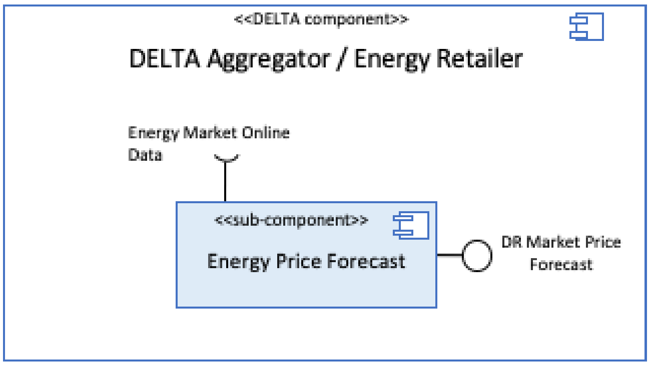

Figure 29. Energy Market Price Forecast

The *Energy Market Price Forecast* counts with the interfaces specified in Table 18, the data involved in the data exchange follows the models specified in Deliverable D1.7.

Table 18. Energy Market Price Forecast API

Interface	R/W	Description
Energy Market Online Data	W	This interface receives the energy market online data provided by an external oracle. The data follow the DELTA data model specification
DR Market Price Forecast	R	This interface exposes the DR market price forecast computed by the *Energy Market Price Forecast*. This price profile follows the DELTA data model.

5.7.4 DR & Flexibility Forecasting

The *DR & Flexibility Forecasting* sub-component is one of the pillars in the DELTA architecture. It allows conforming the balance responsibility that a *DELTA Aggregator / Energy Retailer* may count with. In addition allows to maximize the benefits of applying DR strategies successfully in the underneath DELTA components, i.e., *Virtual DELTA Nodes* and *DELTA Fog Enabled Agents*. Figure 30 below depicts the required interaction that this sub-component demands to gather the data necessary to compute the forecasted flexibility.

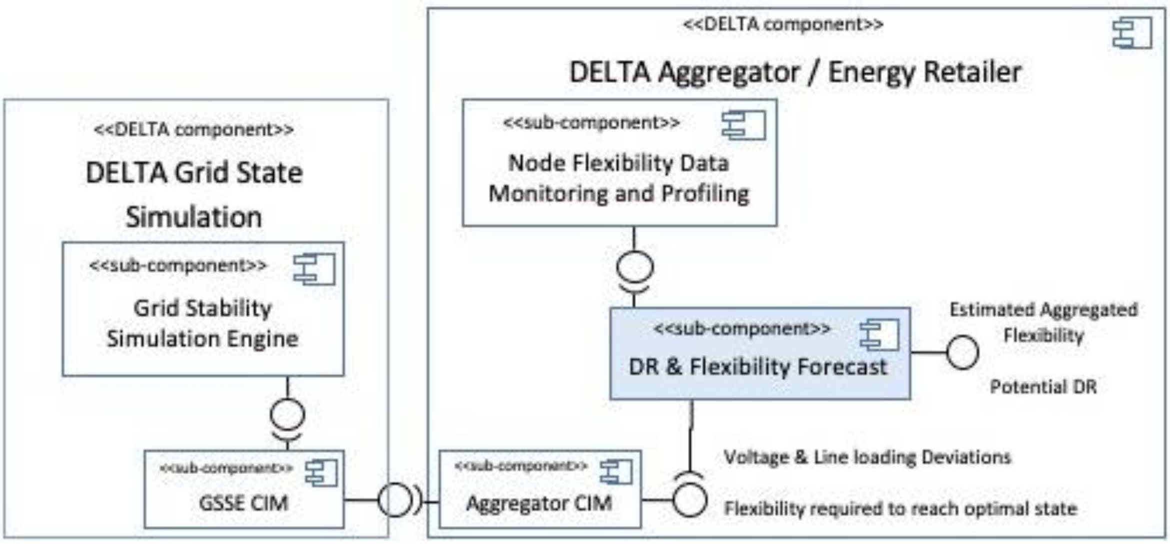

Figure 30. DR & Flexibility Forecasting

The *DR & Flexibility Forecasting* relies on the interfaces explained in Table 19, the data involved in such interfaces is described in Deliverable D1.7.

Table 19. DR & Flexibility Forecasting API

Interface	R/W	Description
Voltage & Line Loading Deviations and Flexibility required to reach optimal state	W	This interface receives the information about voltage & line loading deviations and flexibility required to reach optimal state from the *Grid Stability Simulation through the corresponding CIM components*. The information follows the DELTA data model
Aggregated Profiling	W	This interface receives the aggregated profiling from the *Node Flexibility Data Monitoring and Profiling component*, which summarizes the behaviour of the underneath layers and forecasts the flexibility of the costumer's flexible loads. The profiling follows the DELTA data model
Estimated Aggregated Flexibility, and Potential DR	R	This interface exposes the computed estimated aggregated flexibility, and potential DR, which follows the DELTA data model

5.7.5 Asset Handling Optimisation

The *Asset Handling Optimisation* aims at processing incoming DR signals sent by the external oracle *DR Services to Market Stakeholders*. To fulfil its purpose this sub-component analyses the incoming signal, and extracts the required actions to fulfil the DR signal by combining the forecasted assets flexibility. This may entail re-distribute the assets of a certain *DELTA Aggregator / Energy Retailer*, or create DR signals to forward to the underneath *Virtual DELTA Nodes*. In addition, this component generates *Transactions* that reflect the interactions done between the *DELTA Aggregator / Energy Retailer* and the *Virtual DELTA Nodes*. Figure 31 below depicts the interactions required by the *Asset Handling Optimisation* and the rest of the DELTA components to implement this functionality. Notice that this component takes the market behaviour into account when deciding the DR signals that must be sent to the *Virtual DELTA Nodes*.

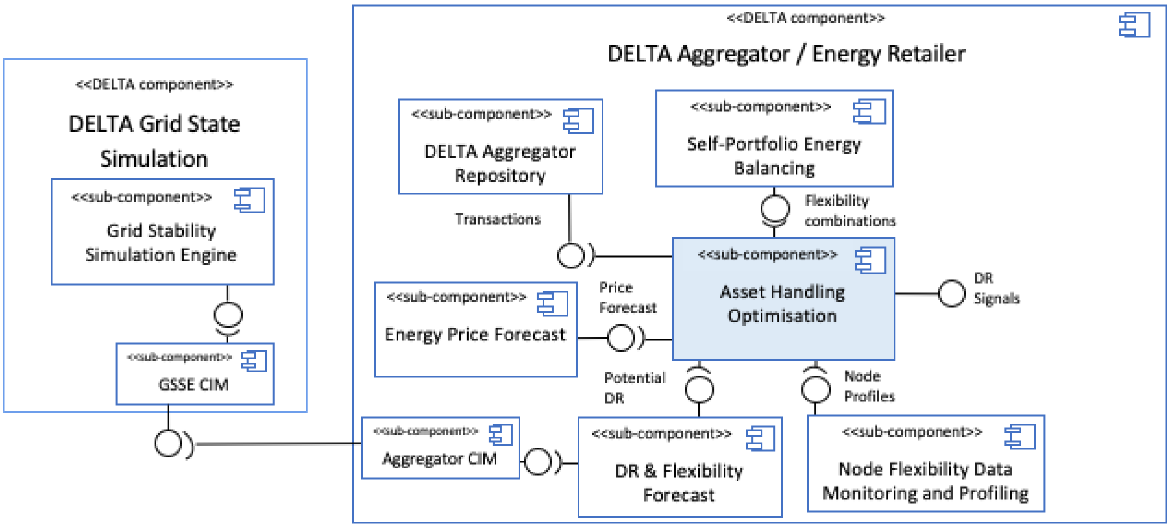

Figure 31. Asset Handling Optimisation

The *Asset Handling Optimisation* relies on the interfaces shown in Table 20, where all the data exchange follows the specifications reported in Deliverable D1.7.

Table 20. Asset Handling Optimisation API

Interface	R/W	Description
Potential DR	W	This interface receives the demand response signals provided by the *GSSE*, which are forwarded by *DR & Flexibility Forecast component*. The demand response signals follow the DELTA data model
Price Forecast	W	This interface receives the price forecast computed by the Energy Market Price forecast, so the energy market price can be taken into consideration for the decision-making. The price forecast follows the DELTA data model
Node Profiles	W	This interface receives the aggregated profiling computed by the *Node Flexibility Data Monitoring and Profiling*. The profiling follows the DELTA data model
Flexibility combinations	W	This interface receives the flexibility combinations computed by the *Self Portfolio Energy Balancing*, so it can be taken into consideration for the decision-making. The market settlement follows the DELTA data model
Transactions	W	This interface receives the computed transactions stored in the *DELTA Aggregator Repository*. The transactions follow the DELTA data model
DR Signals	R	This interface exposes the computed demand response signals that will be forwarded by the *OpenADR Aggregator CIM* to the underneath levels (by prior adapting them to the OpenADR communication protocol)

5.7.6 Node Flexibility Data Monitoring and Profiling

The *Node Flexibility data Monitoring and Profiling* aims at gathering information of the underneath DELTA components that a certain *DELTA Aggregator / Energy Retailer* is in charge of, i.e., the *Virtual DELTA Nodes* that *DELTA Aggregator / Energy Retailer* handles. The scope of this sub-component is to allow the aggregator to have a clear real-time overview of its assets. Figure 32 below depicts this component.

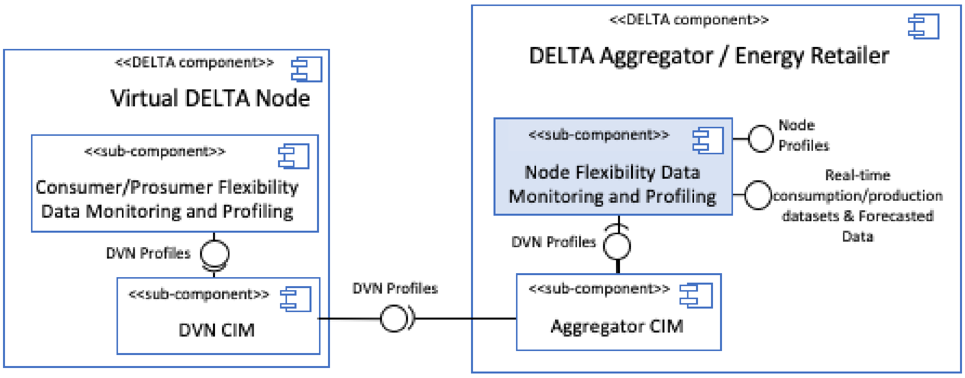

Figure 32. Node Flexibility Data Monitoring and Profiling

The *Node Flexibility Data Monitoring and Profiling* relies on the interfaces described in Table 21, the data involved in each of the interfaces follow the data models specified in Deliverable D1.7.

Table 21. Node Flexibility Data Monitoring and Profiling API

Interface	R/W	Description
DVN Profiles	W	This interface receives the DVN profiling provided by the *Virtual DELTA Nodes* underneath, which are forwarded by

		the *OpenADR Aggregator CIM*. The Node Profiling follow the DELTA data model
Real-time consumption/production datasets & Forecasted Data	R	This interface exposes the computed Real-time consumption/production datasets & Forecasted Data that offer an overview of the status of *Virtual DELTA Nodes* that a certain aggregator handles. The profiling and forecasted data follows the DELTA data model
Node Profiles	R	This interface exposes the computed node profiling that offers an overview of the status of *Virtual DELTA Nodes* that a certain aggregator handles. The aggregated profiling follows the DELTA data model

5.8 Virtual DELTA Node

The *Virtual DELTA Node* is a pillar component in the DELTA architecture. This component aims at managing sets of *DELTA Fog Enabled Agents* based on commands established by the *DELTA Aggregator / Energy Retailer*. The *Virtual DELTA Node* aims at matching the energy demand with a corresponding supply. The main DELTA component *Virtual DELTA Node* consist of several sub-components as depicted in Figure 33 below, namely: *Generation/Consumption Optimal Dispatch*, *Consumer/Prosumer Flexibility Data Monitoring and Profiling*, *Consumer/Prosumer Clustering*, *Inter/Intra Node Energy Matchmaking*, and *Load Forecasting*. In following sub-sections each of this sub-components is explained in detail.

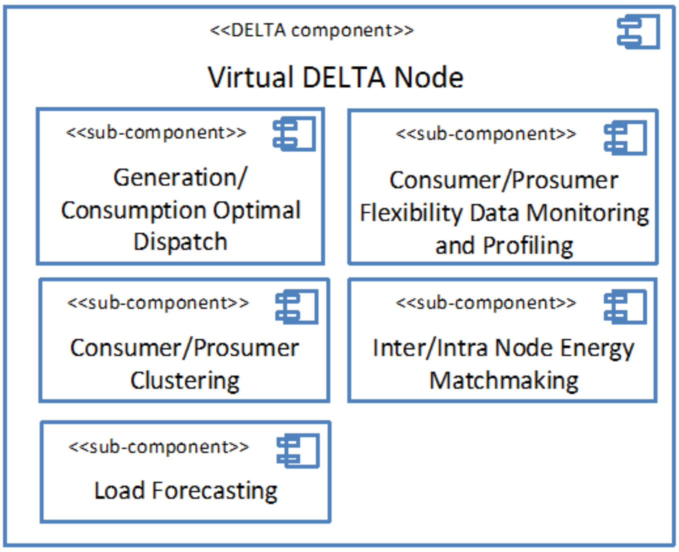

Figure 33. Virtual DELTA Node Overview

5.8.1 Consumer/Prosumer Clustering

This component implements border line techniques to cluster and group in segments the different *DELTA Fog Enabled Agents* allocated in the same *Virtual DELTA Node*. The goal of this clustering is to assign each *DELTA Fog Enabled Agent* to a different *Virtual DELTA Node* relaying on some energetic requirements established by the *DELTA Aggregator / Energy Retailer*. As a result, the clustering will produce suitable configurations to meet the energetic needs required.

Techniques that this component implement will be fed by means of the data retrieve through the sub-component *Energy Portfolio Segmentation and Classification* in the *DELTA Aggregator / Energy Retailer* that establishes the *DVN Clusters*, which are received thanks to the *OpenADR Virtual Node CIM*. In addition, *the Consumer/Prosumer Flexibility Data Monitoring and Profiling* provides the *Node Profiling*, which together with the DR Signals of the *Intra/Inter Node Energy Matchmaking* are used to compute the clusters. Figure 34 below depicts these interactions required by the *Consumer/Prosumer Clustering.*

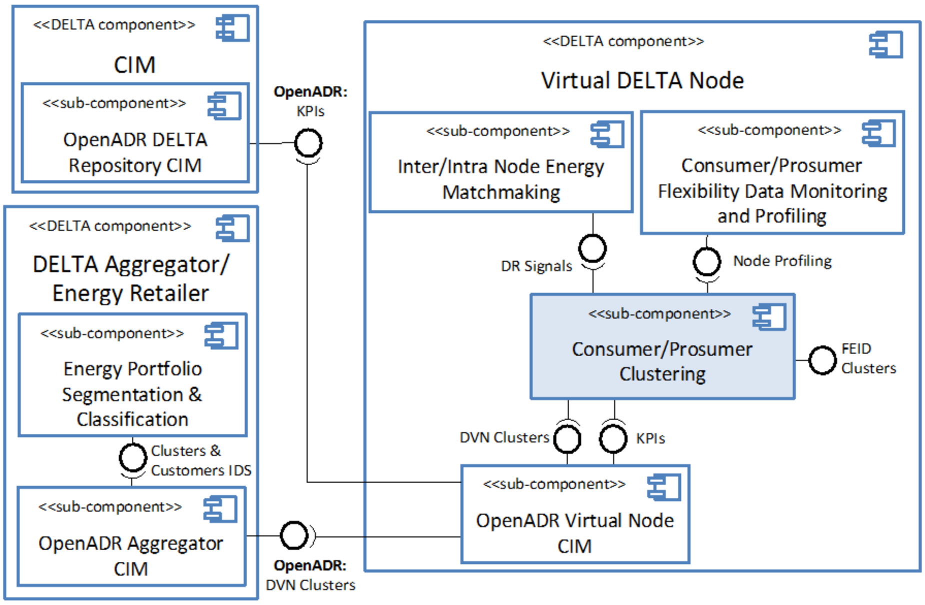

Figure 34. Consumer/Prosumer Clustering

The Consumer/Prosumer Clustering relies on the interfaces reported in Table 22. The data involved in each of the interfaces has being explained and defined in Deliverable D1.7.

Table 22. Consumer/Prosumer Clustering API

Interface	R/W	Description
DR Signals	W	This interface receives the demand response signals, which follow the DELTA data model
Node Profiling	W	This interface receives the node profiling, which follow the DELTA data model
KPIs	W	This interface receives the KPIs, which follow the DELTA data model. The KPIs are fetched from the *DELTA Repository CIM*
DVN Clusters	W	This interface receives the clusters of the nodes, which follow the DELTA data model
FEIDs Clustering	R	This interface exposes locally the clustering of the *DELTA Fog Enabled Agent* computed by the *Consumer/Prosumer Clustering*. The clusters follow the DELTA data model

5.8.2 Consumer/Prosumer Flexibility Data Monitoring and Profiling

The *Consumer/Prosumer Flexibility Data Monitoring and Profiling* aims at providing a real-time overview of the assets assigned to a specific *Virtual DELTA Node*, i.e., the *DELTA Fog Enabled Agents* that a certain *Virtual DELTA Node* is in charge of managing. The values that this component monitors are namely the data gathered by the different devices that *DELTA Fog Enabled Agent* handles, as well as, the computed flexibility forecast related to those devices. Figure 35 below depicts the *Consumer/Prosumer Flexibility Data Monitoring and Profiling.*

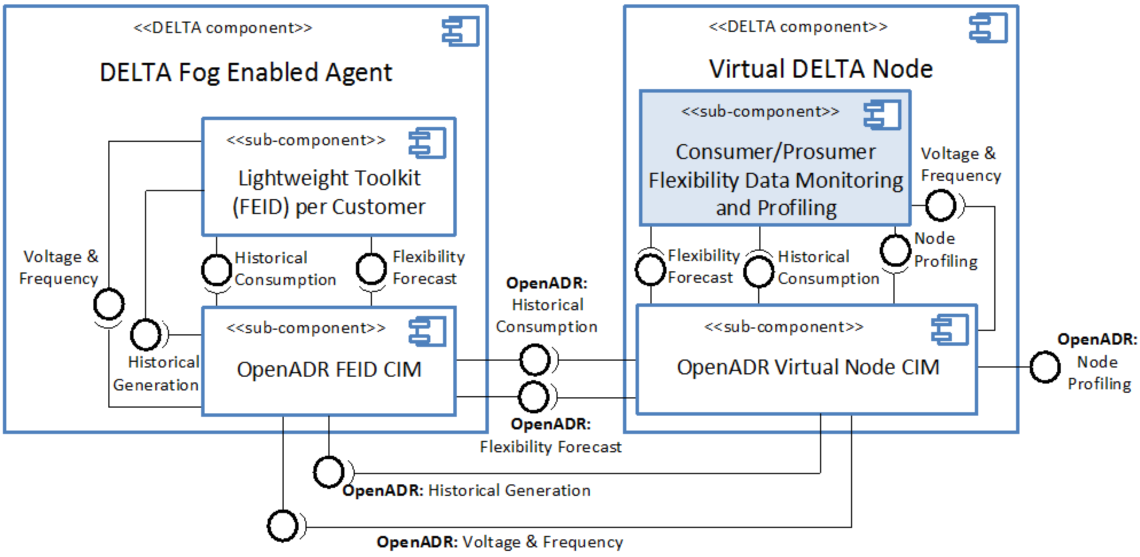

Figure 35. Consumer/Prosumer Flexibility Data Monitoring and Profiling

The *Consumer/Prosumer Flexibility Data Monitoring and Profiling* relies on the interfaces reported in Table 23. The data involved in each of the interfaces has being explained and defined in Deliverable D1.7.

Table 23. Consumer/Prosumer Flexibility Data Monitoring and Profiling API

Interface	R/W	Description
Historical Consumption	W	This interface receives the historical consumption gathered from the *DELTA Fog Enabled Agents* underneath a certain *Virtual DELTA Node*. The historical data is received thanks to the *OpenADR Virtual Node CIM*. This historical data follows the DELA data model
Historical Generation	W	This interface receives the historical generation gathered from the *DELTA Fog Enabled Agents* underneath a certain *Virtual DELTA Node*. The historical data is received thanks to the *OpenADR Virtual Node CIM*. This historical data follows the DELA data model
Voltage & Frequency	W	This interface receives the voltage and frequency measurements gathered from one or more *DELTA Fog Enabled Agents* underneath a certain *Virtual DELTA Node*. These measurements are received thanks to the *OpenADR Virtual Node CIM*. This historical data follows the DELA data model
Flexibility Forecast	W	This interface receives the flexibility forecast computed from the *DELTA Fog Enabled Agents* underneath a certain *Virtual DELTA Node*. The forecasted data is received thanks to the *OpenADR Virtual Node CIM*. This forecasted data follows the DELA data model
Node Profiling	R	This interface exposes the computed node profiling, which follows the DELTA data model.

5.8.3 Generation/Consumption Optimal Dispatch

The *Generation/Consumption Optimal Dispatch* aims at establishing energy decision that the *DELTA Fog Enabled Agents* must fulfil. This component takes from the *DELTA Aggregator / Energy Retailer* the supplied DR Signals, computes the optimal course that the underneath *DELTA Fog Enabled Agents* should take, and emits DR Signals to them. In addition, this component is responsible of generating *Transactions* reflecting the interactions between the *Virtual DELTA Nodes* and the *DELTA*

Fog Enabled Agent. These *Transactions* are stored in the *DELTA Blockchain*. Figure 36 below depicts the *Generation/Consumption Optimal Dispatch*.

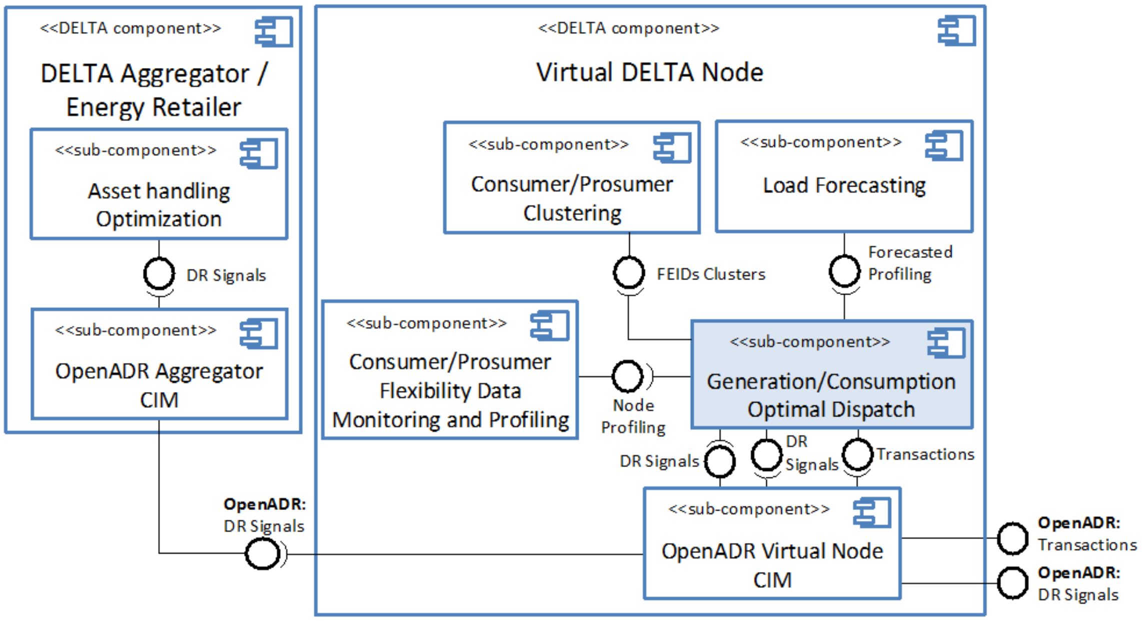

Figure 36. Generation/Consumption Optimal Dispatch

The *Generation/Consumption Optimal Dispatch* relies on the interfaces reported in Table 24. The data involved in each of the interfaces has being explained and defined in Deliverable D1.7.

Table 24. Generation/Consumption Optimal Dispatch API

Interface	R/W	Description
Node Profiling	W	This interface receives the computed node profiling, which follows the DELTA data model
FEIDs Clusters	W	This interface receives clusters for the *DELTA Fog Enabled Agents*, which follow the DELTA data model
Forecasted Energy Profiling	W	This interface receives forecasted energy profiling, which follows the DELTA data model
DR Signals	W	This interface receives the demand response signals computed at the *DELTA Aggregator/Energy Retailer* level. The demand response signals are forwarded to this component thanks to the *OpenADR Virtual Node CIM* following the OpenADR communication protocol. The received demand response signals follow the DELTA data model
DR Signals	R	This interface exposes the computed demand response signals, which follow the DELTA data model. These signals are forwarded to the *DELTA Fog Enabled Agents* that a certain *Virtual DELTA Node* handles following the OpenADR communication protocol. The demand response signals are adapted to this protocol thanks to the *OpenADR Virtual Node CIM.*
Transactions	R	This interface exposes the transactions between a *Virtual DELTA Node* and a *Fog Enabled Agent, or another* Virtual DELTA Node. The transactions follow the DELTA data model.

5.8.4 *Load Forecasting*

The *Load Forecasting* sub-component is a pillar element to handle the *DELTA Fog Enabled Agents* that a *Virtual DELTA Node* is in charge of. The forecasted values are a paramour element to establish the right course of actions when DR Signals are sent from the *DELTA Aggregator/Energy Retailer*, It provides the basis for flexibility forecast of the expected loads. This forecasted values allows maximizing the availability of the assets in order to fulfil the energy promises established between the *Demand Response Services to Market Stakeholders* and the *DELTA Aggregator/Energy Retailer*. Figure 37 below depicts the *Load Forecasting*.

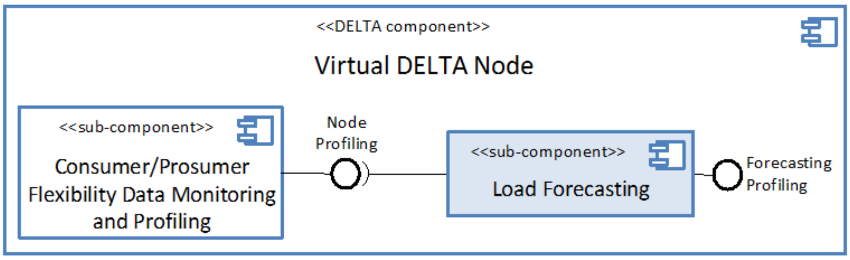

Figure 37. Load Forecasting

The Load Forecasting relies on the interfaces reported in Table 25. The data involved in each of the interfaces has being explained and defined in Deliverable D1.7.

Table 25. Load Forecasting API

Interface	R/W	Description
Node Profiling	W	This interface receives the node profiling of a *Virtual DELTA Node*, which summarizes the data of the underneath *DELTA Fog Enabled Agents*. The profiling follows the DELTA data model
Forecasting Profiling	R	This interface exposes the forecasting profiling computed by the *Load Forecasting*. The forecasting profiling follows the DELTA data model

5.8.5 *Inter/Intra Node Energy Matchmaking*

The *Inter/Intra Node Energy Matchmaking* aims at managing the *DELTA Fog Enabled Agents* of a certain *Virtual DELTA Node*. Analysing the FEIDs Profiling of the underneath *DELTA Fog Enabled Agents*, and the current clusters it aims at reassigning the assets of its *Virtual DELTA Node* by sending DR Signals to the underneath *DELTA Fog Enabled Agents* or to other *Virtual DELTA Node* present in the DELTA Platform. Figure 38 below depicts the *Inter/Intra Node Energy Matchmaking*.

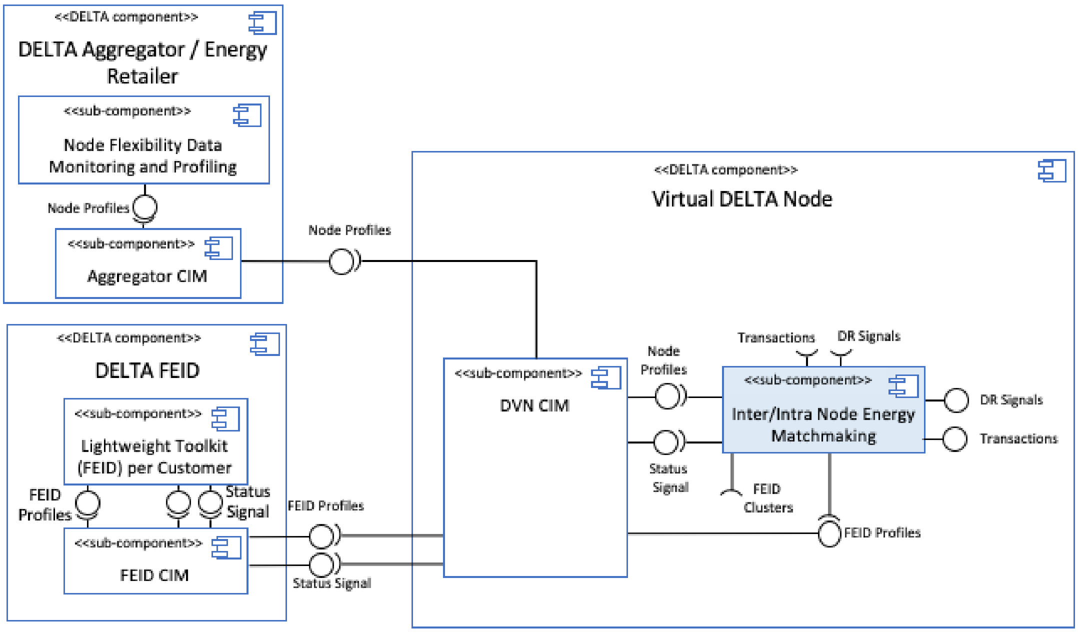

Figure 38. Inter/Intra Node Energy Matchmaking

The *Inter/Intra Node Energy Matchmaking* relies on the interfaces reported in Table 26. The data involved in each of the interfaces has being explained and defined in Deliverable D1.7.

Table 26. Inter/Intra Node Energy Matchmaking API

Interface	R/W	Description
Status Signal	W	This interface receives the status signals of the FEIDs related to a specific *Virtual DELTA Node*; which is obtained thanks to the *DELTA Virtual Node CIM* following the OpenADR communication protocol. The status signals follow the DELTA data model
FEIDS Clusters	W	This interface receives the clusters of the FEIDs computed by the *Consumer/Prosumer Clustering*. These clusters follow the DELTA data model
FEIDS Profiles	W	This interface receives profiling of the FEIDS, computed by the *DELTA Aggregator / Energy Retailer*. The profiling follows the DELTA data model
Node profiles	W	This interface receives profiling of the nodes, computed by the *DELTA Aggregator / Energy Retailer*. The profiling follows the DELTA data model
Transactions	W	This interface receives the computed transactions. The transactions follow the DELTA data model
DR Signals	W	This interface recieves the computed demand response signals, which follow the DELTA data model. The demand response signals follow the DELTA data model specification
Transactions	R	This interface exposes the computed transactions. The transactions follow the DELTA data model
DR Signals	R	This interface exposes the computed demand response signals, which follow the DELTA data model. These signals are forwarded to *Fog Enabled Agents* that a certain *Virtual DELTA Node* handles or to other *Virtual DELTA*

		Nodes. In both cases the communication follows the OpenADR communication protocol. The demand response signals are adapted to this protocol thanks to the *OpenADR Virtual Node CIM*. The demand response signals follow the DELTA data model specification

5.9 DELTA FOG Enabled Agent

The *DELTA Fog Enabled Agent* is a main DELTA component whose goal is twofold, on the one hand this component handles a set of devices by monitoring them and providing real time measurements such as the consumption/generation or the flexibility forecast. On the other hand, this component is responsible of adjusting these devices to meet a set of energy requirements established by the *DELTA Aggregator / Energy Retailer*, which are forwarded to the *DELTA Fog Enabled Agent* through a *Virtual DELTA Node.*

5.9.1 DELTA Fog Enabled Agent

The *DELTA Fog Enabled Agent* consists of only one component, i.e., the *Lightweight Toolkit (FEID) per Customer*. This sub-component is in charge of processing incoming DR Signals and, at the same time, forwarding to upper layers the observed values and computed metrics. Figure 39 below depicts the *Lightweight Toolkit (FEID) per Customer.*

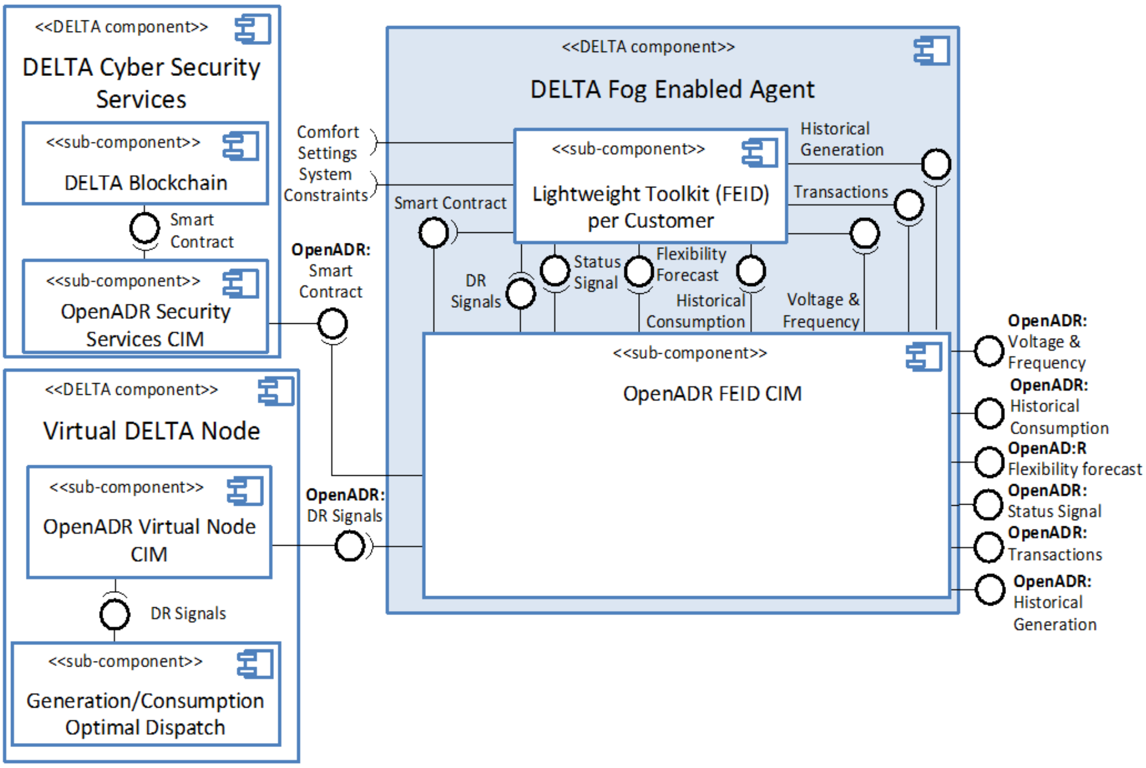

Figure 39. DELTA Fog Enabled Agent

The *Lightweight Toolkit (FEID) per Customer* relies on the interfaces reported in Table 27. The data involved in each of the interfaces has being explained and defined in Deliverable D1.7.

Table 27. DELTA Fog Enabled Agent API

Interface	R/W	Description
DR Signals	W	This interface receives the demand response signals computed at the *Virtual DELTA Node* level. The demand response signals are forwarded to this component thanks to the *OpenADR FEID CIM*. The received signals follow the DELTA data model
System Constraints	W	This interface receives some system constrains specified by the owner of the *Lightweight Toolkit (FEID) per Customer*. The constraints modify the regular behaviour of this asset. The constraints are expressed following the DELTA data model
Comfort Settings	W	This interface receives some comfort settings specified by the owner of the *Lightweight Toolkit (FEID) per Customer*. These settings modify the regular behaviour of this asset. The settings are expressed following the DELTA data model
Smart Contracts	W	This interface receives the smart contracts from the *DELTA Blockchain* sub-component. The contracts are forwarded to this component thanks to the *OpenADR FEID CIM*. The received contracts follow the DELTA data model
Historical Generation	R	This interface exposes the historical generation stored by the *Lightweight Toolkit (FEID) per Customer*. This historical data is delivered to the *Virtual DELTA Node* to through the *OpenADR FEID CIM;* following the OpenADR communication protocol and DELTA data model
Historical Consumption	R	This interface exposes the historical generation stored by the *Lightweight Toolkit (FEID) per Customer*. This historical data is delivered to the *Virtual DELTA Node* to through the *OpenADR FEID CIM;* following the OpenADR communication protocol and DELTA data model
Voltage & Frequency	R	This interface exposes the voltage and frequency measurements provided by the *Lightweight Toolkit (FEID) per Customer*. These measurements are delivered to the *Virtual DELTA Node* to through the *OpenADR FEID CIM;* following the OpenADR communication protocol and DELTA data model
Forecasted Flexibility	R	This interface exposes the forecasted flexibility computed by the *Lightweight Toolkit (FEID) per Customer*. This forecasted data is delivered to the *Virtual DELTA Node* to through the *OpenADR FEID CIM;* following the OpenADR communication protocol and DELTA data model
Status Signal	R	This interface exposes the status signal tracked by the *Lightweight Toolkit (FEID) per Customer*. This signal is delivered to the *Virtual DELTA Node* to through the *OpenADR FEID CIM;* following the OpenADR communication protocol and DELTA data model
Transactions	R	This interface exposes the transactions computed by the *Lightweight Toolkit (FEID) per Customer*. The transactions are forwarded to the DELTA Blockchain through the *OpenADR FEID CIM;* following the OpenADR communication protocol and DELTA data model

6. Information View

The Information View displays an overview of the model specified in the DELTA platform to express the data within. The DELTA platform counts with a model that is based namely in three components, the OpenADR standard, the W3C SAREF standard, and a tailored-designed data model.

The OpenADR standard includes some data model to express DR signals, among other concepts used in the DELTA Platform. Nevertheless, it lacks of data models to express information such as Smart Contracts, or the Clusters & Customer IDs. Due to this reason we also incorporated the SAREF Ontology, which covers some of these needs. Finally, some concepts engaged in the platform where not modelled yet by any standard nor suitable public data model; for instance the Smart Contracts. This lack of data models led us to develop a tailored-designed ontology to cover such concepts.

The current specification of the DELTA data model is depicted in Figure 40. As it can be observed, the main DELTA components, e.g., *Virtual DELTA Node*, as well as, the data produced, exchanged, and consumed, e.g., Rewards, has being already included in our model

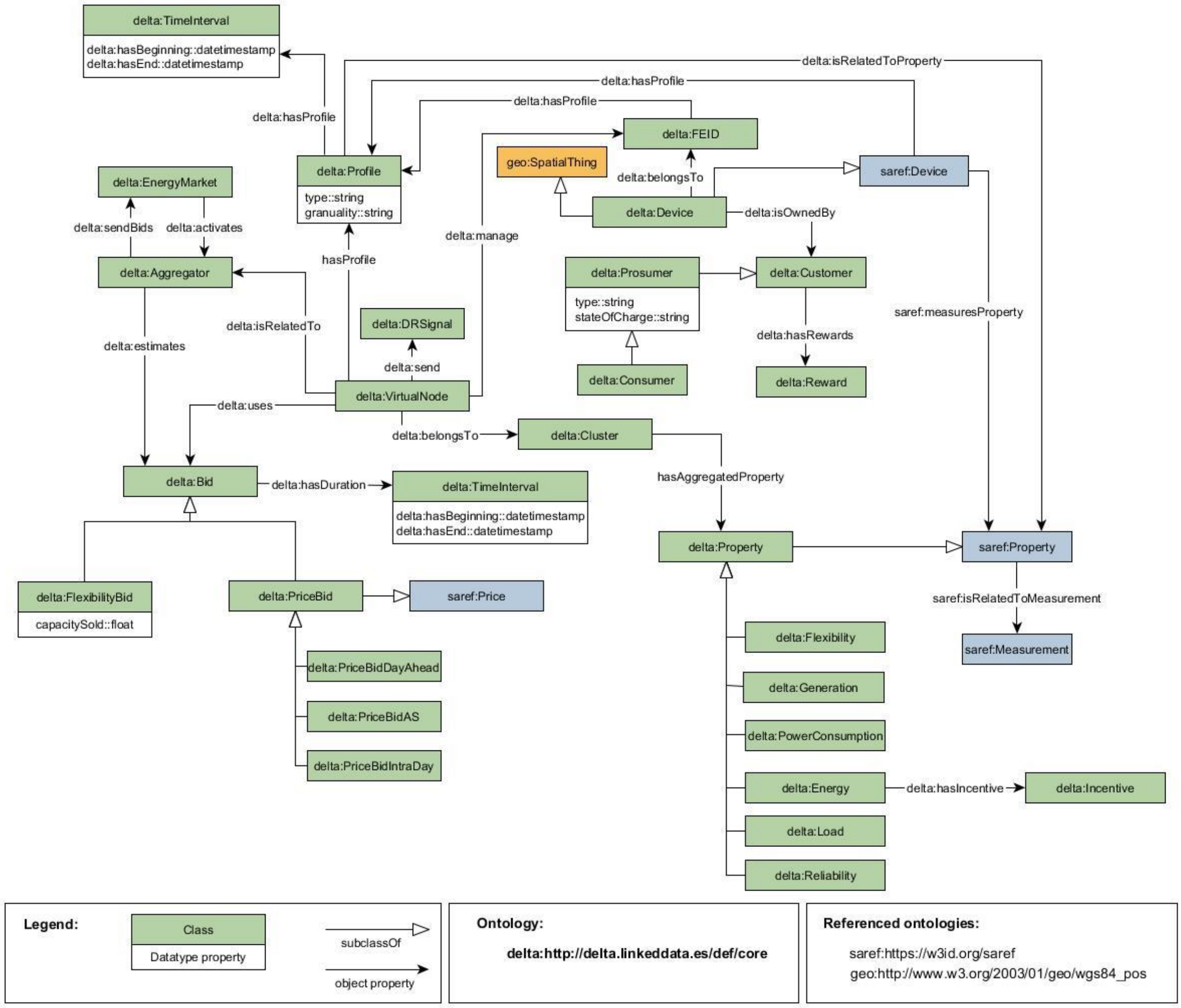

Figure 40. DELTA Data Model

6.1 Information Flow

The Function View section already provided an overview of the data flow, however, in this section we aim at showing the interaction of the different main DELTA components. Figure 41 depicts the data involved in the exchange between such components.

Bear in mind that all the OpenADR data exchanges must implement the OpenADR communication protocol standards, which in DELTA, are handled by the CIM sub-components.

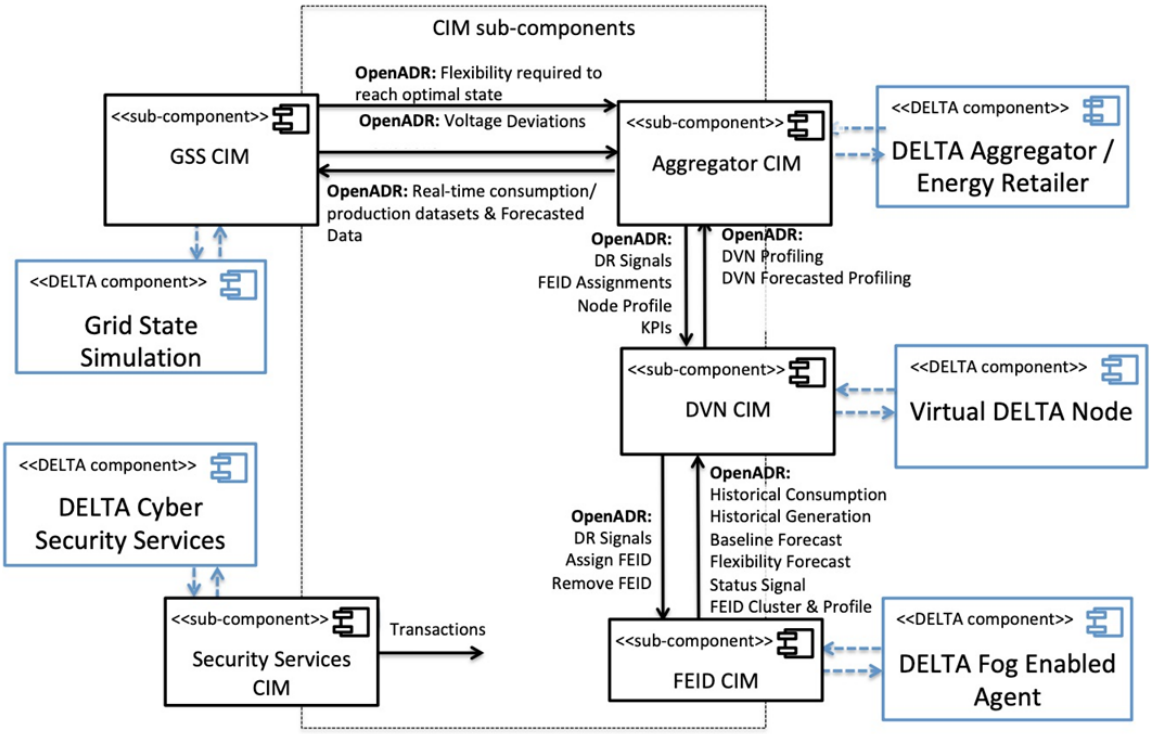

Figure 41. Information Flow Overview

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. The deployment view will help inform the installation requirements at the pilot sites.

The deployment view will be developed throughout the course of the project. Herein, the deployment view focuses on aspects of components that have been developed or are in development to inform the deployment view, hence an overview of the deployment environment of the DELTA platform is developed. UML Deployment Diagrams are presented here to elaborate the deployment.

The deployment information for each of the components is summarised by the following headings/descriptors:

- **Component:**
  A modular part of a system whose behaviour is defined by its provided and required interfaces
- **Owner**
  Lead beneficiary
- **Support**
  Supporting beneficiaries
- **Hardware/Software Object**
  It could be a device or an execution environment. Devices are computing resources with processing capabilities and the ability to execute programs. Some examples of device nodes include PCs, laptops, and mobile phones.
  An execution environment is any computer system that resides within a device. It could be an operating system, a virtual machine, or another servlet container. Computational resource upon which artefacts may be deployed for execution
- **Hardware/Software Requirements**
  Properties or guiding parameters that must be defined for deployment to occur
- **Existing Hardware/Software Objects**
  Existing device or execution environment
- **Existing Hardware/Software Requirements**
  Third-party requirements
- **Software Artefacts**
  A product in the physical world that is used or produced by the software process or by deployment and operation of the system. It could be a text document, source file, script, binary executable file, archive file, database table.
- **Interaction**
  What and who will connect to or interact with the component

Table 28. High Level Deployment View Information

Component	Owner	Support	Hardware/Software object	Hardware/Software requirements	Existing Hardware /Software objects	Existing Hardware/Software requirements	Software artefacts	Interaction
Grid Stability Simulation Engine	UCY	EAC/UPM/HIT/JRC/KIWI	Windows VM deployed on Windows server 2012 on UCY server at the UCY campus in Nicosia	4 Processor Cores, 16 GB RAM, GPU required for fast processing of state simulations. Likely will coded using CUDA libraries thus GPU must be NVIDIA, to reduce time for simulations a GTX1080 minimum. 1TB RAID Array Storage	PC	5th Generation Intel i5, 16 GB RAM, GTX 1080 GPU	ascii text files/historical data/physical limitations probabilities	Node Flexibility Data Monitoring & Profiling/Energy Market Price Forecast
Self Portfolio Energy Balancing	UCY	UPM/HIT/JRC/KIWI	Windows VM deployed on Windows server 2012 on UCY server at the UCY campus in Nicosia	2 Processor Cores, 8 GB RAM. 0.5TB RAID Array Storage	PC	5th Generation Intel i5, 16 GB RAM, GTX 1080 GPU	ascii text files/strategries and market bids	External Market/Asset Handling
Flexibility and DR Forecasting	JRC	CERTH/HIT/UCY	Windows deployed on JRC server	2 processor cores, 2.2 GHz processor, 32 GB RAM	PC	Intel (R) Xeon (R) Silver 4114 CPU, 2.2 GHz, 2 core processors,	close to real-time data display through OpenADR, csv files	Load Forecast, Historical data

						Windows 10, RAM 64 GB	(Python) possible with aggregated and disaggregated information	
Energy Market Price Forecast	JRC	CERTH/ HIT/UC Y	Windows deployed on JRC server	2 processor cores, 2.2 GHz processor, 8 GB RAM	PC	Intel (R) Xeon (R) Silver 4114 CPU, 2.2 GHz, 2 core processors, Windows 10, RAM 64 GB	optimization software, i.e. GAMs, excel files	Grid Stability Simulation Engine, Historical data
Fog Enabled Intelligent Device	CERT H	CERTH	Linux	2 processor cores, 1 GHz processor, 1GB RAM	Raspberry	Broadcom BCM2837B0 , Cortex-A53 (ARMv8) 64-bit 1.4GHz	csv file, script, database table	Platform services, Blockchain
DELTA Reposi-tory	UPM	CERTH/ HIT/JRC /NTNU						
Consumer/Prosumer Flexibility Data Monitoring and Profiling	HIT	CERTH/ UPM	Windows/Linux	2 Processor Cores i5 or i7, 16 GB RAM. 0.5TB	PC	Existing PC	Visual Studio (C++), Spider (Python) netbeans	
Generation/Consumption Optimal Dispatch	HIT	CERTH/ UPM	Windows/Linux	2 Processor Cores i5 or i7, 16 GB RAM. 0.5TB	PC	Existing PC	Visual Studio (C++), Spider (Python) netbeans	

Load Forecasting	HIT	CERTH/ UPM	Windows/Linux	2 Processor Cores i5 or i7, 8 GB RAM. 0.5TB	PC	Existing PC	Visual Studio (C++), Spider (Python) netbeans	
Inter/Intra Node Energy Matchmaking	HIT	CERTH/ UPM	Windows/Linux	4 Processor Cores >2.4GHz, 16GB RAM	PC	-	Visual Studio (C++), Spider (Python) netbeans	
Consumer/Prosumer Clustering	CERTH	HIT	Windows/Linux	4 Processor Cores >2.4GHz, 16GB RAM	PC	-	Visual Studio (C++), Spider (Python) netbeans	
Asset Handling Optimization	CERTH	HIT/UCY/JRC	Windows/Linux	2 Processor Cores i5 or i7, 16 GB RAM. 0.5TB	PC	Existing PC	Visual Studio (C++), Spider (Python) netbeans	
Node Flexibility Data Monitoring and Profiling	CERTH	HIT/UCY/JRC	Windows/Linux	2 Processor Cores i5 or i7, 16 GB RAM. 0.5TB	PC	Existing PC	Visual Studio (C++), Spider (Python) netbeans	
Energy Portfolio Segmentation & Classification (DELTA Nodes)	HIT	CERTH/ UCY/JRC	Windows/Linux	4 Processor Cores >3GHz, 32GB RAM	PC	-	Visual Studio (C++), Spider (Python) netbeans	
Demand Response Visualization Kit	CERTH	HIT	Windows/Linux	2 Processor Cores i5 or i7, 16 GB RAM. 0.5TB	PC	Existing PC	Spider (Python)	

Award-enabled Energy Behavioral Platform	CERTH	EAC/UCY/KiWi/CARR	Windows/Linux	2 Processor Cores i5 or i7, 16 GB RAM. 0.5TB	PC	Existing PC	Spider (Python)	
Social Interaction and Cooperation	CERTH	EAC/UCY/KiWi/CARR	Windows/Linux	2 Processor Cores i5 or i7, 16 GB RAM. 0.5TB	PC	Existing PC	Spider (Python)	

8. Dynamic View (Use Case Analyses)

The dynamic view analysis of the system provides insights and defines how the system actually works within the runtime environment and how it performs in response to external (or internal) signals. 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 DELTA use cases were defined and analysed in deliverable "D1.5 DELTA Requirements, Business Scenarios and Use Cases V1". In the context of the WP1 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.

8.1 DELTA Business Scenario 1 – Provision of high efficiency Demand Response services through the user of Delta Virtual Node Platform and associated services layer

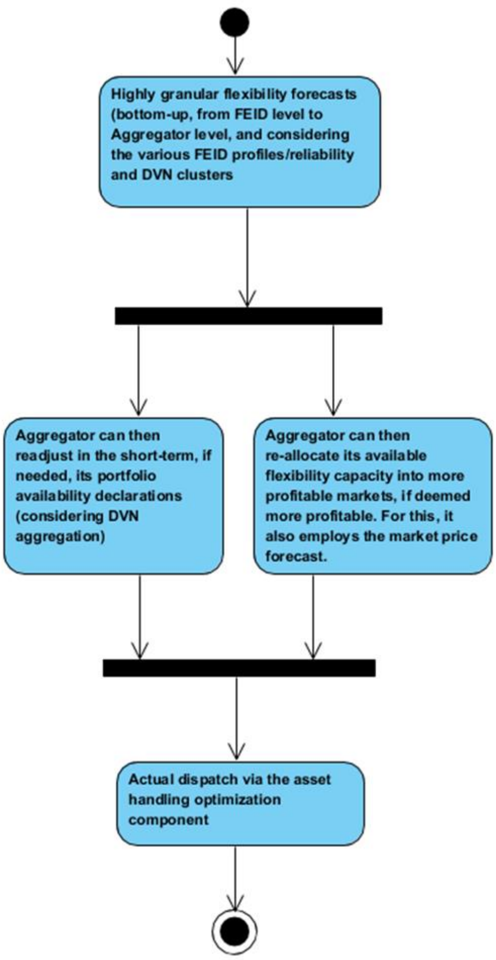

Figure 42 BS1 Activity Diagram

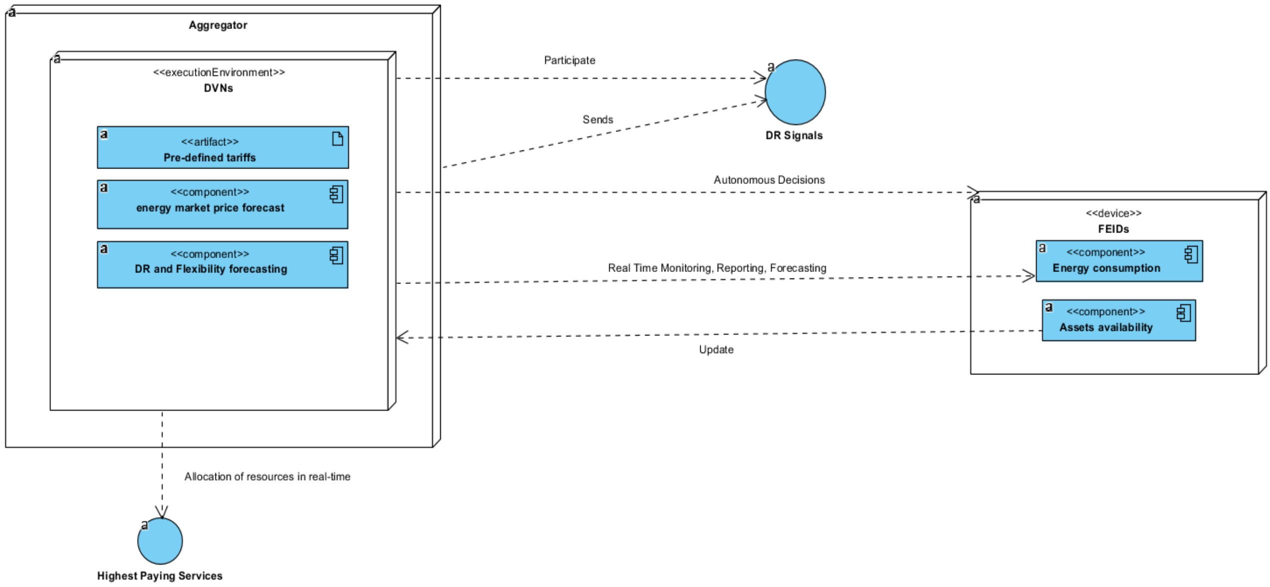

Figure 43 BS1 Deployment Diagram

8.1.1 DELTA BS1 – UC1: Flexibility forecast to improve assets availability declaration and maximize Demand Response revenues

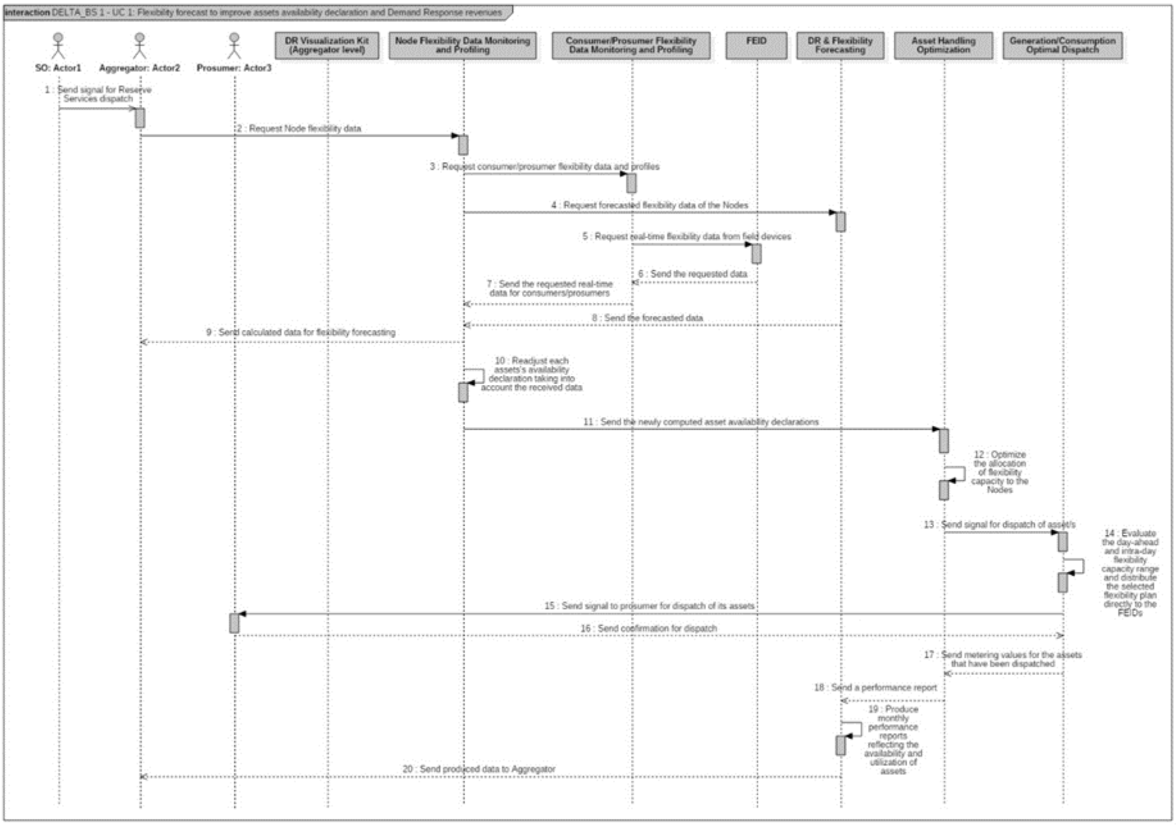

Figure 44. BS1 – UC1 Sequence Diagram

8.1.2 DELTA BS1 – UC2: Improving Demand Side Response (DSR) revenues by trading flexibility in the Imbalance market based on Energy Market Price Forecasts

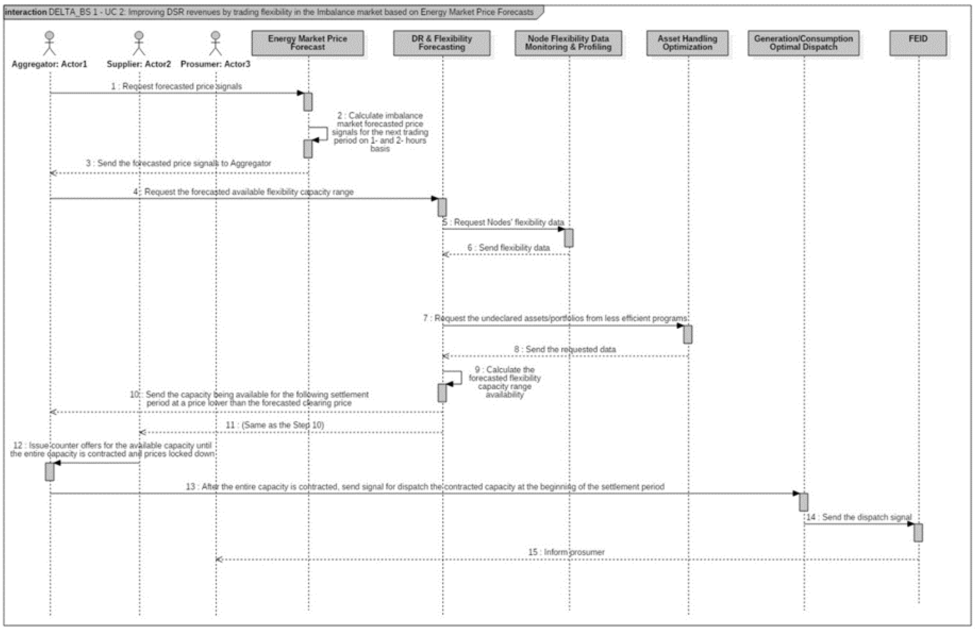

Figure 45. BS1 – UC2 Sequence Diagram

8.2 DELTA Business Scenario 2 – Secure, automated Demand Response services via blockchain enabled smart contracts

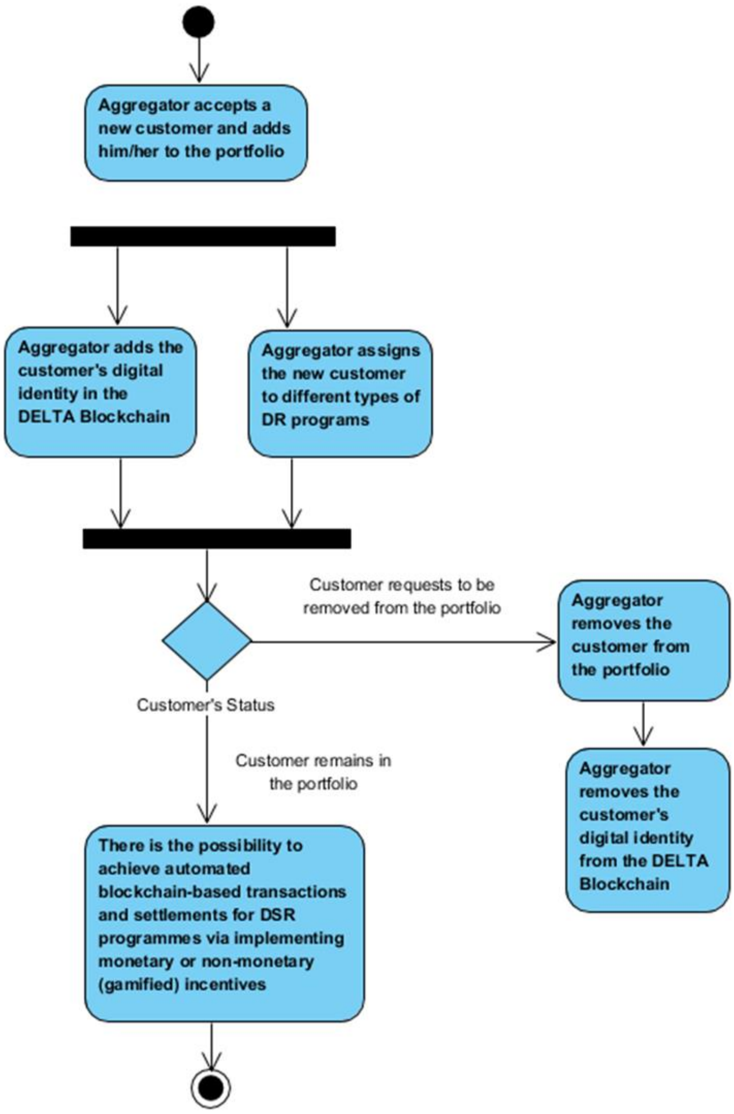

Figure 46 BS2 Activity Diagram

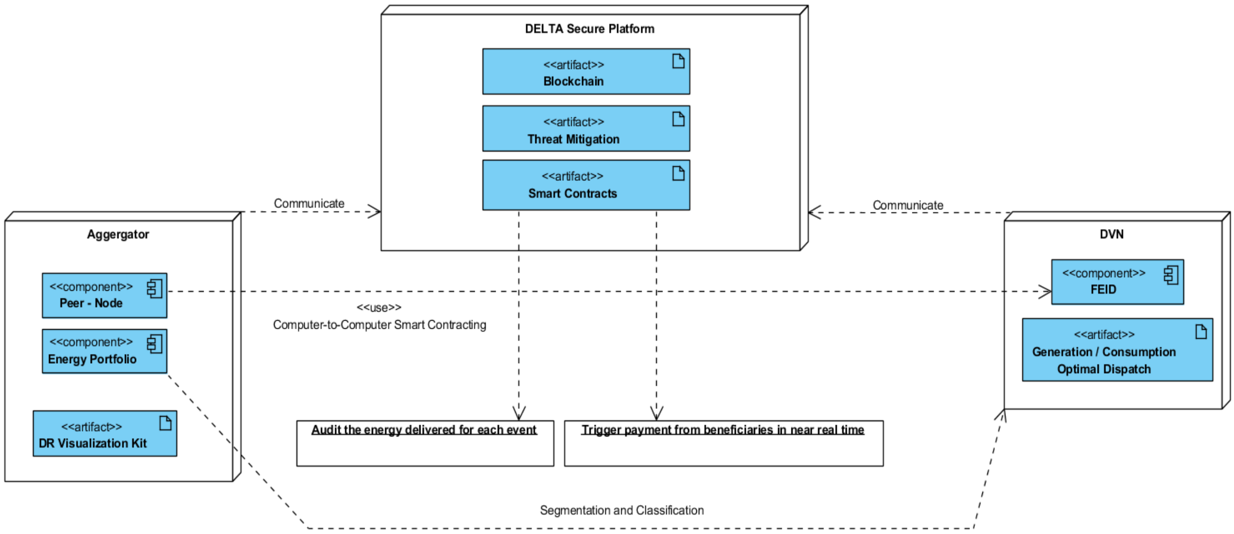

Figure 47 BS2 Deployment Diagram

8.2.1 DELTA BS2 – UC1: Customer Admission to the Aggregator Portfolio

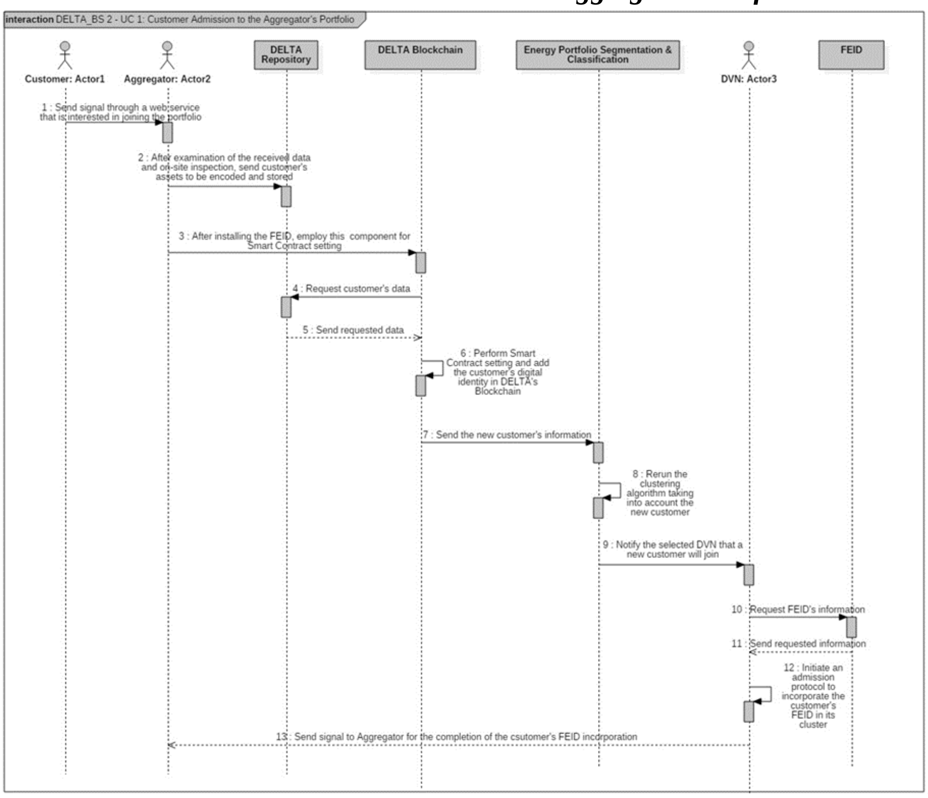

Figure 48. BS2 – UC1 Sequence Diagram

8.2.2 BS2 – UC2: Customer Renunciation from the Aggregator's Portfolio

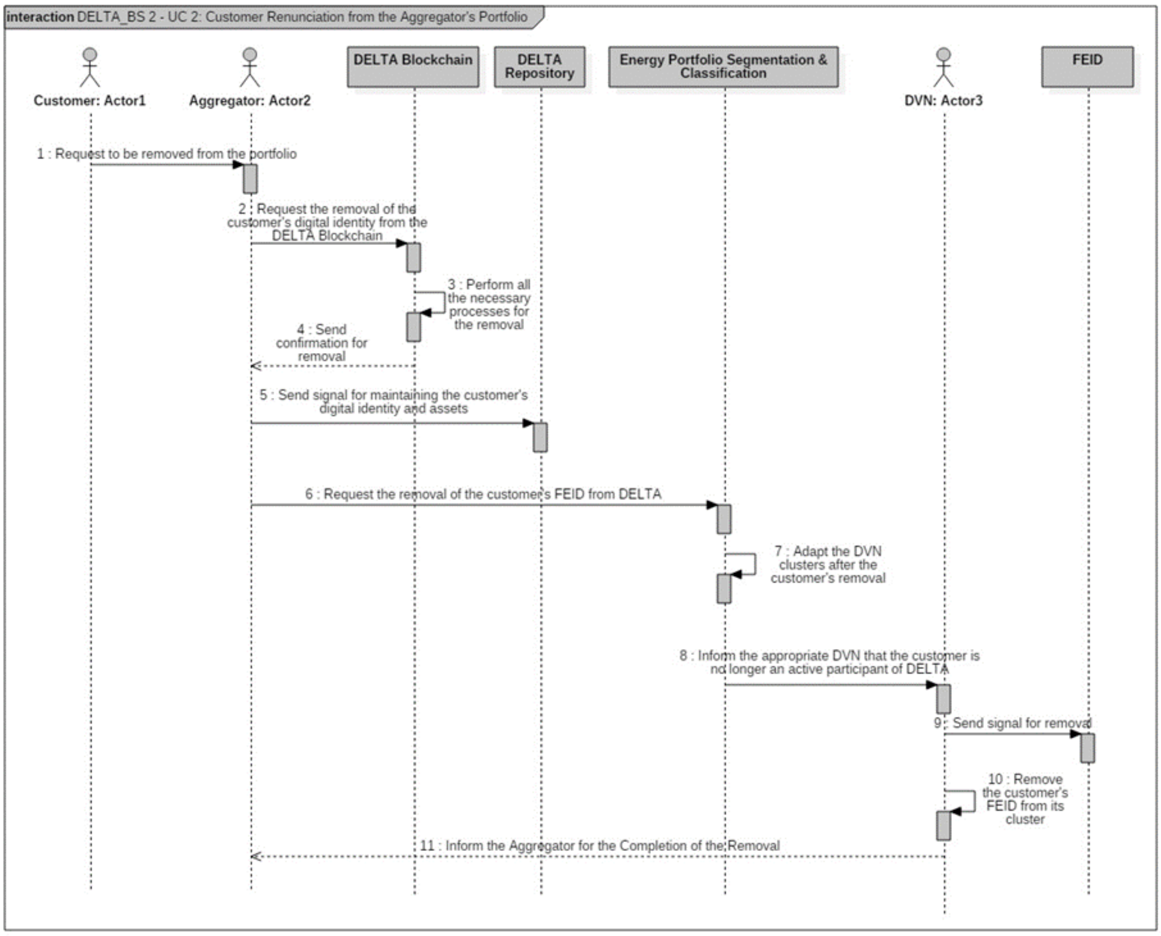

Figure 49. BS2 – UC2 Sequence Diagram

8.2.3 DELTA BS2 – UC3: Automated Demand Side Response settlements through blockchain

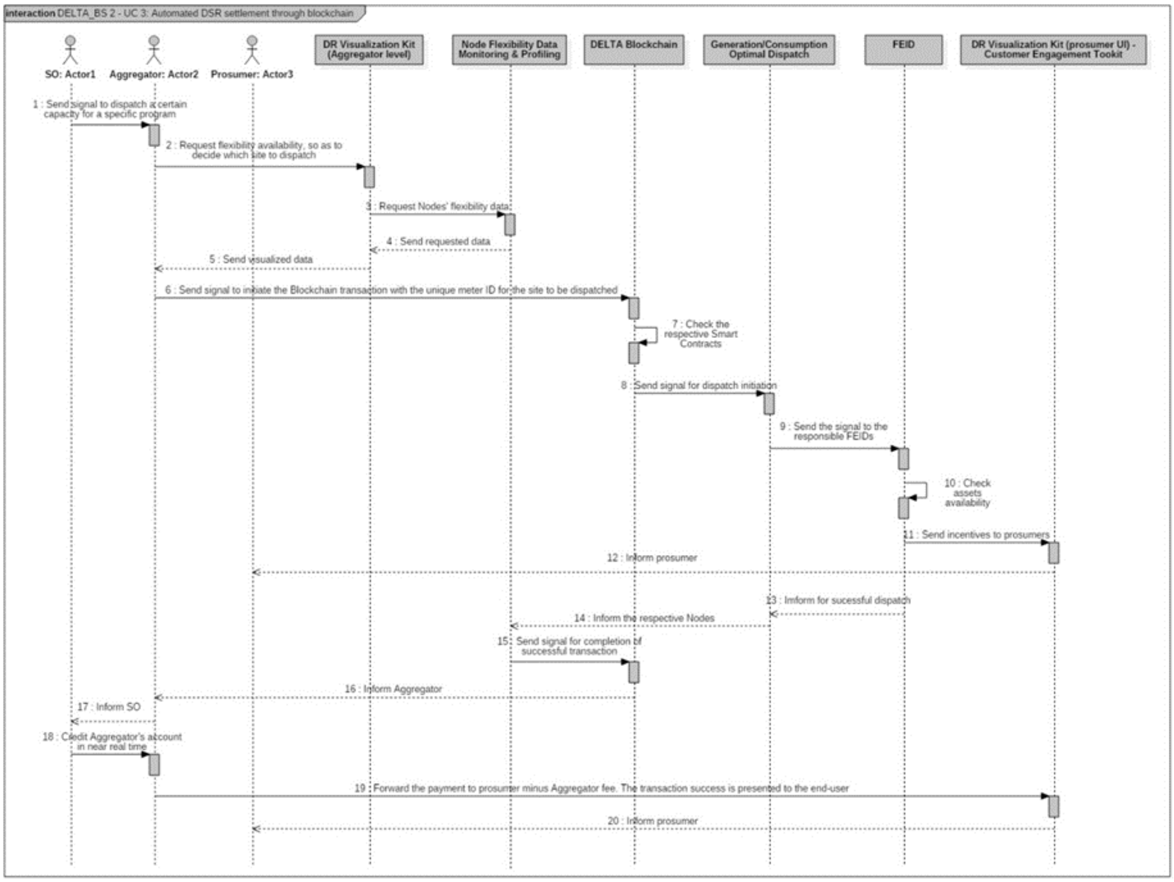

Figure 50. BS2 – UC3 Sequence Diagram

8.2.4 DELTA Indicative Sub-UC1 of BS2 – UC3_SmartHome Use Case 1: Incentive-based Demand Response signal activation involving one Fog Enabled Intelligent Device and one DELTA Virtual Node

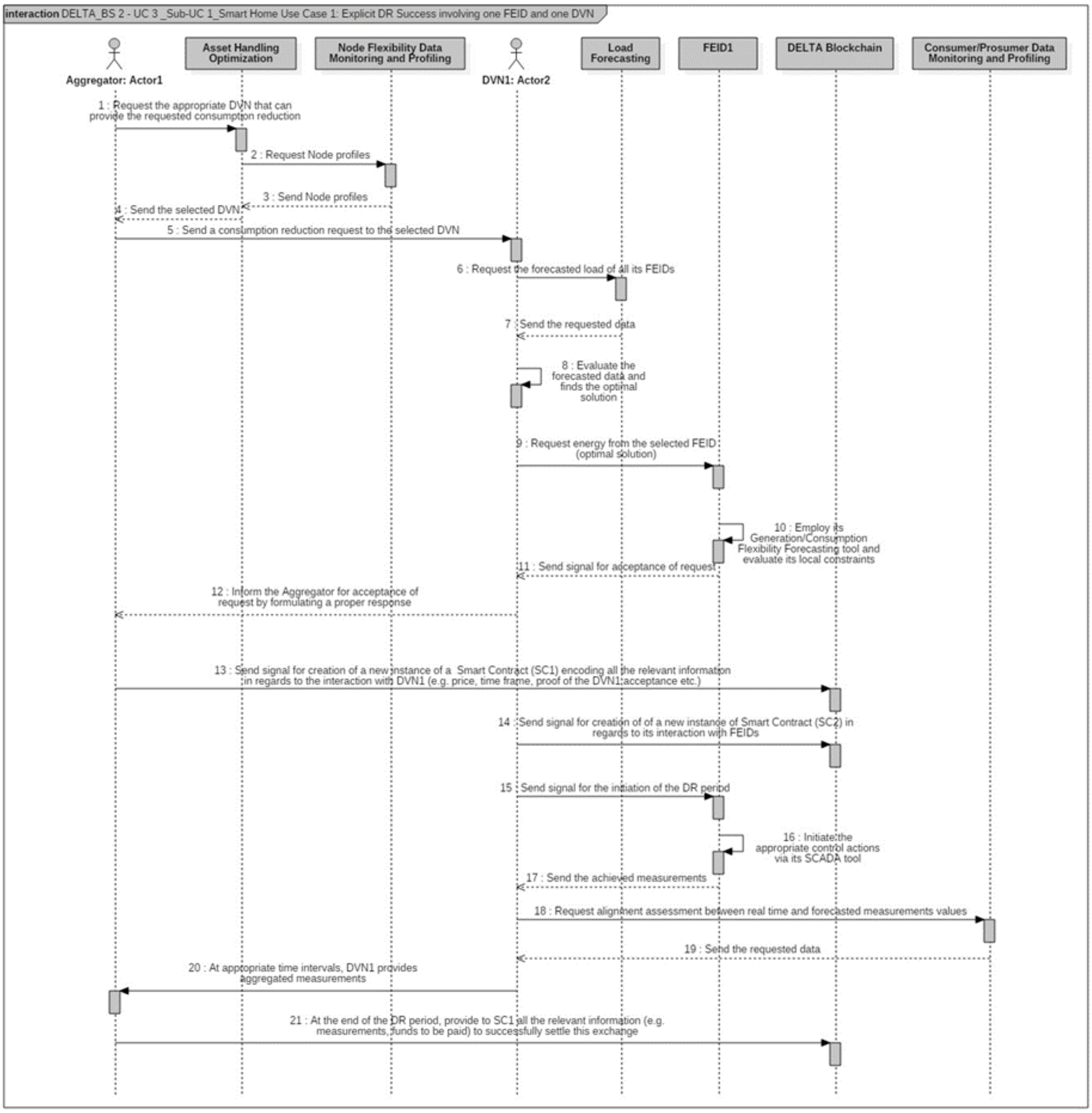

Figure 51. BS2 – UC3_1 Sequence Diagram

8.2.5 DELTA Indicative Sub-UC2 of BS2 – UC3_SmartHome Use Case 2: Intra Delta Virtual Node Allocation – Incentive-based Demand Response signal activation involving two Fog Enabled Intelligent Devices in the same Delta Virtual Node

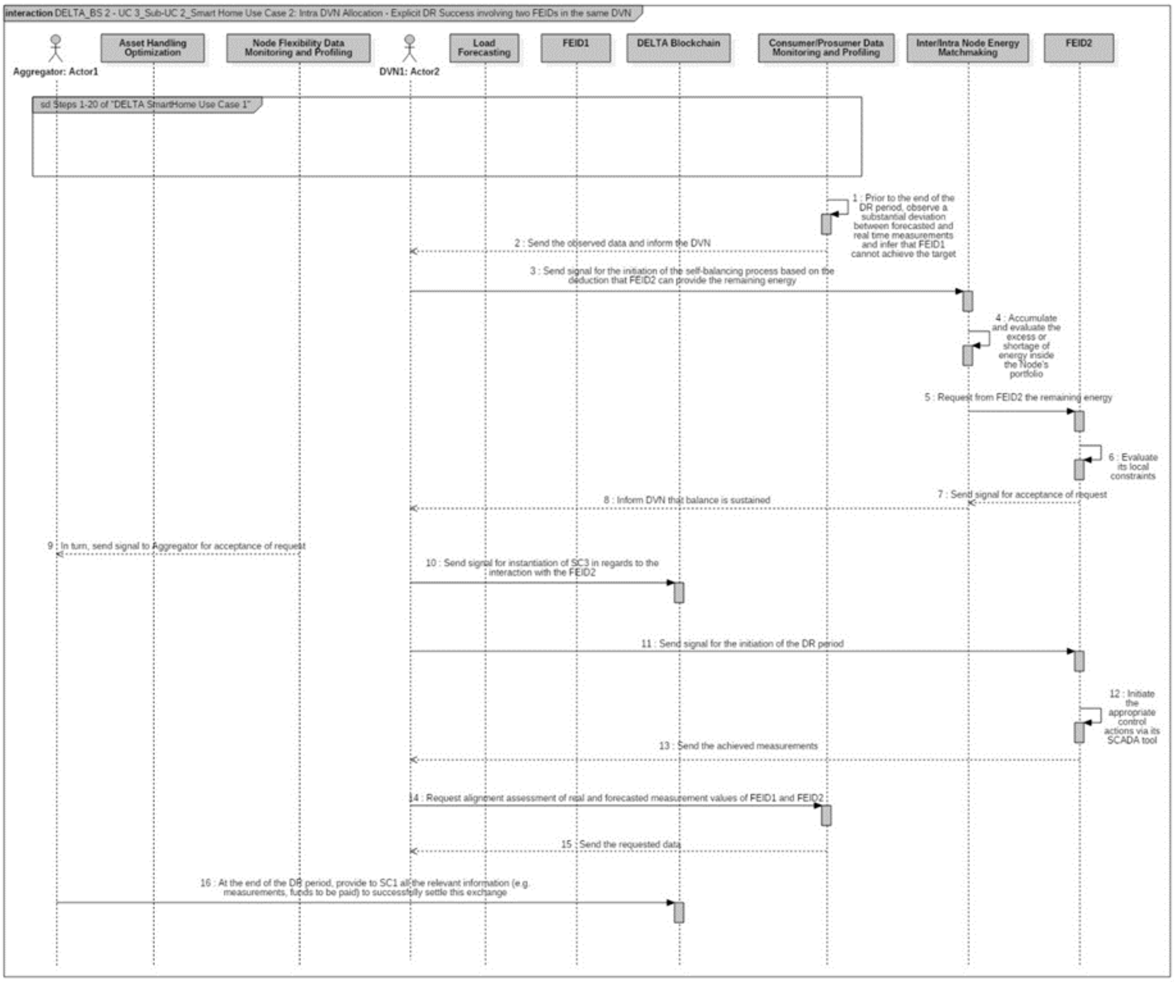

Figure 52. BS2 – UC3_2 Sequence Diagram

8.2.6 DELTA Indicative Sub-UC3 of BS2 – UC3_Smart Home Use Case 3: Inter DELTA Virtual Node Allocation – Explicit Demand Response Success Involving two Fog Enabled Intelligent Devices in two separate DELTA Virtual Node

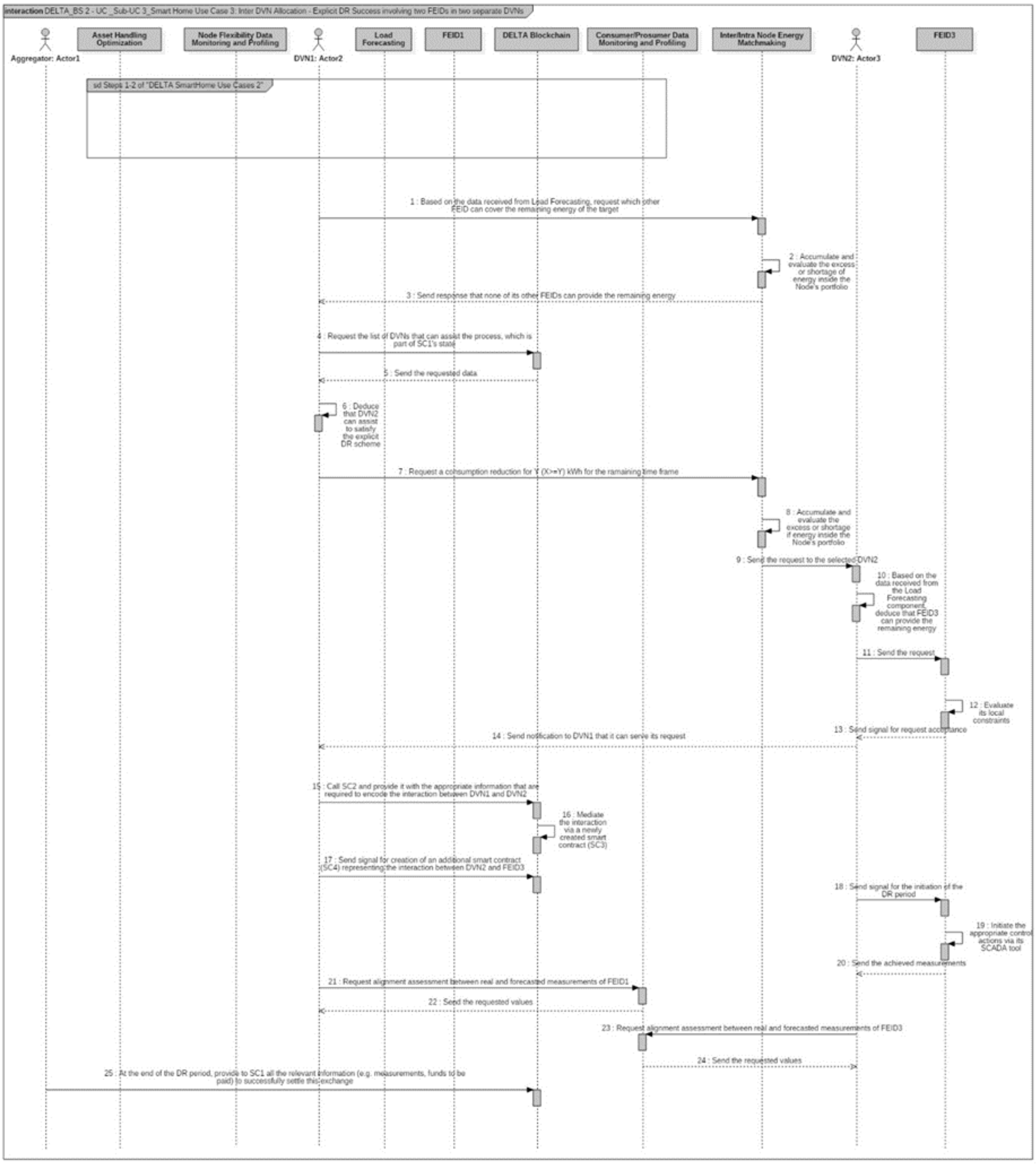

Figure 53. BS2 – UC3_3 Sequence Diagram

8.3 DELTA Business Scenario 3 – Optimal self-portfolio management via DVN and DR services

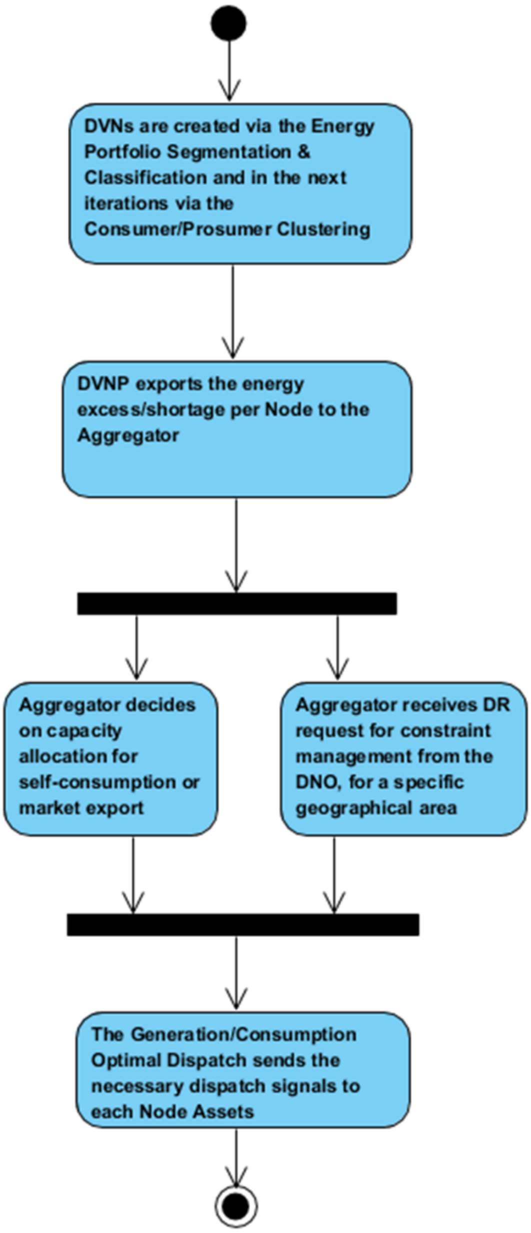

Figure 54 BS3 Activity Diagram

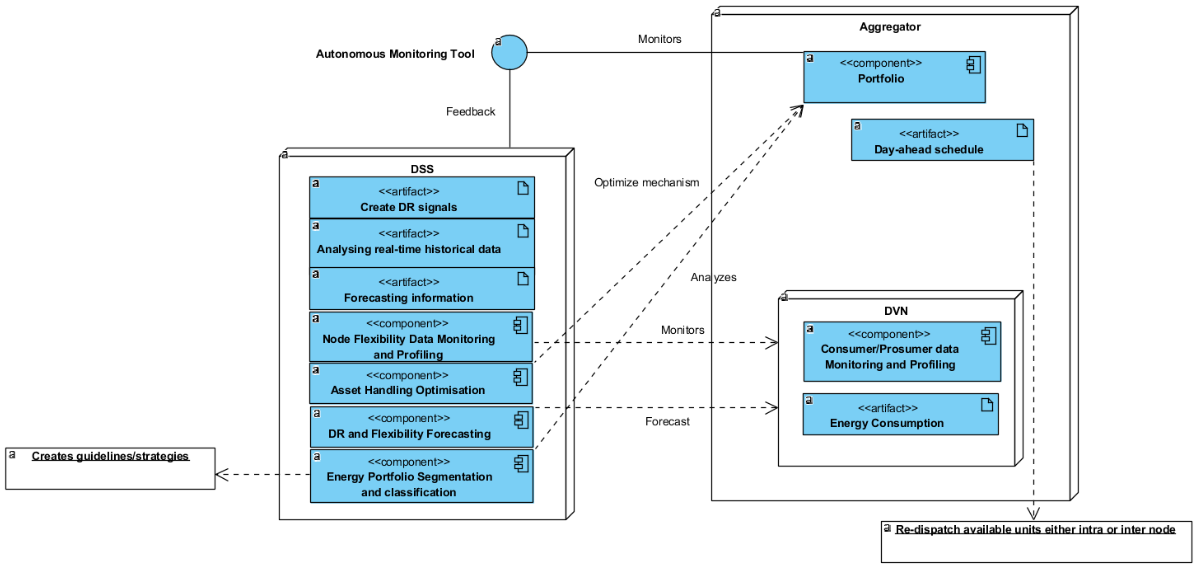

Figure 55 BS3 Deployment Diagram

8.3.1 *DELTA BS3 – UC1: Optimize prosumer Renewable Energy Systems self-consumption and increase flexibility*

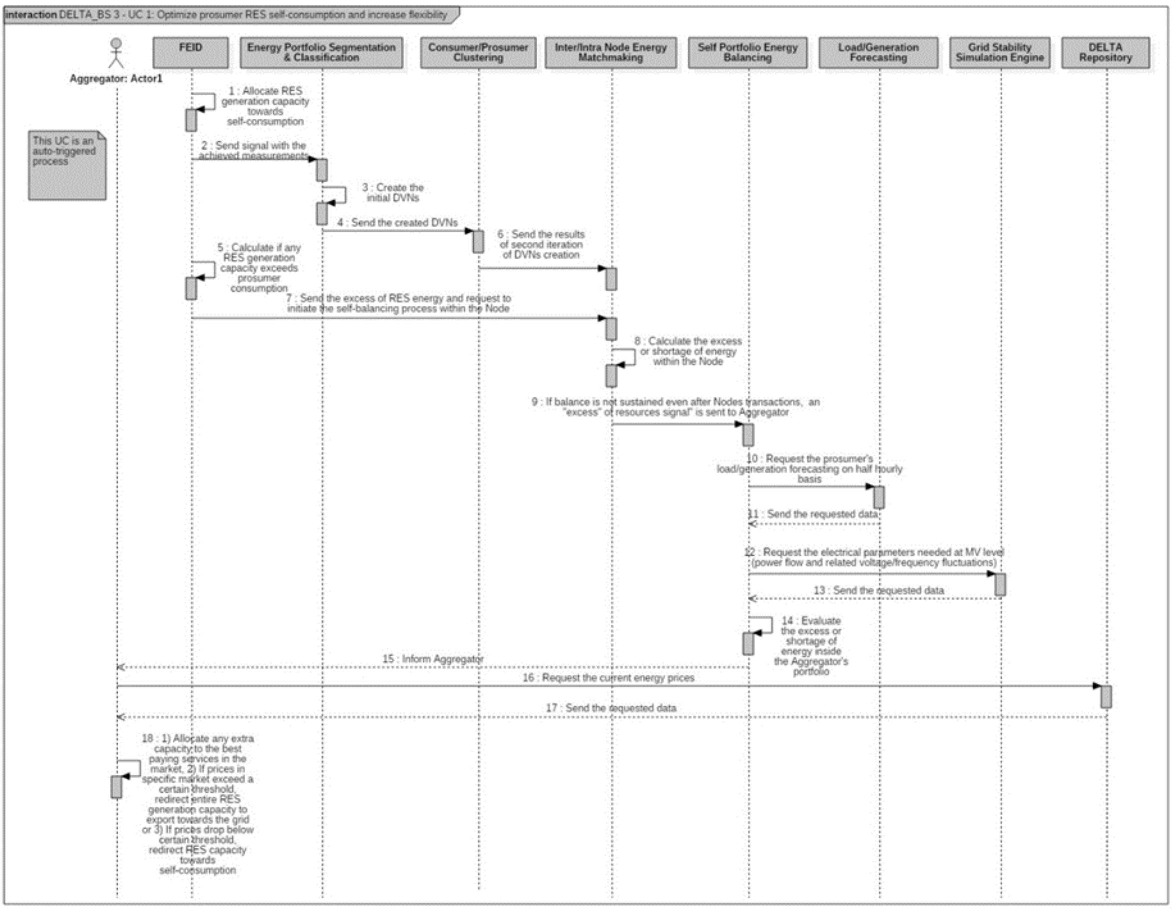

Figure 56. BS3 – UC1 Sequence Diagram

8.3.2 DELTA BS3 – UC2: Providing localized flexibility portfolios for Distribution Network Operator to manage network constraints

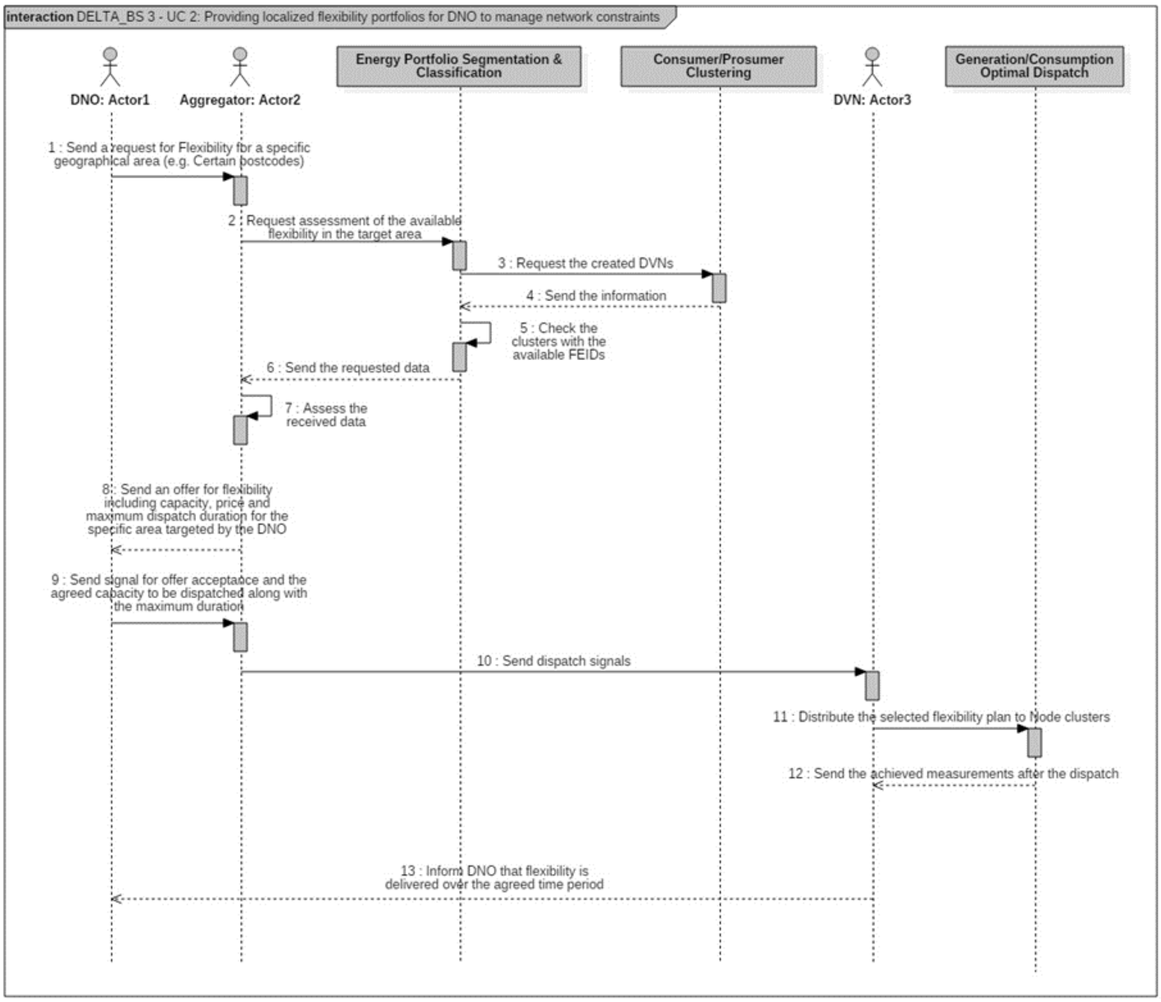

Figure 57. BS3 – UC2 Sequence Diagram

8.4 DELTA Business Scenario 4 – Secure, real time asset metering and control through FEID and DELTA Virtual Nodes

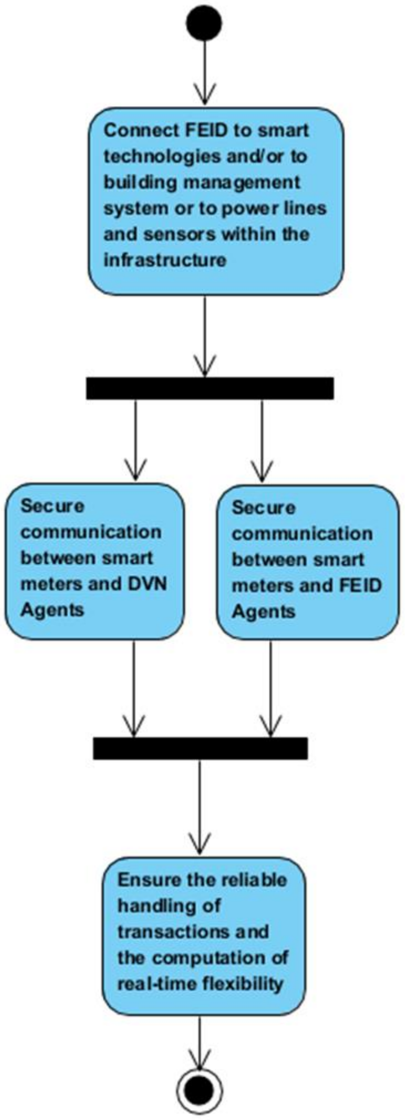

Figure 58 BS4 Activity Diagram

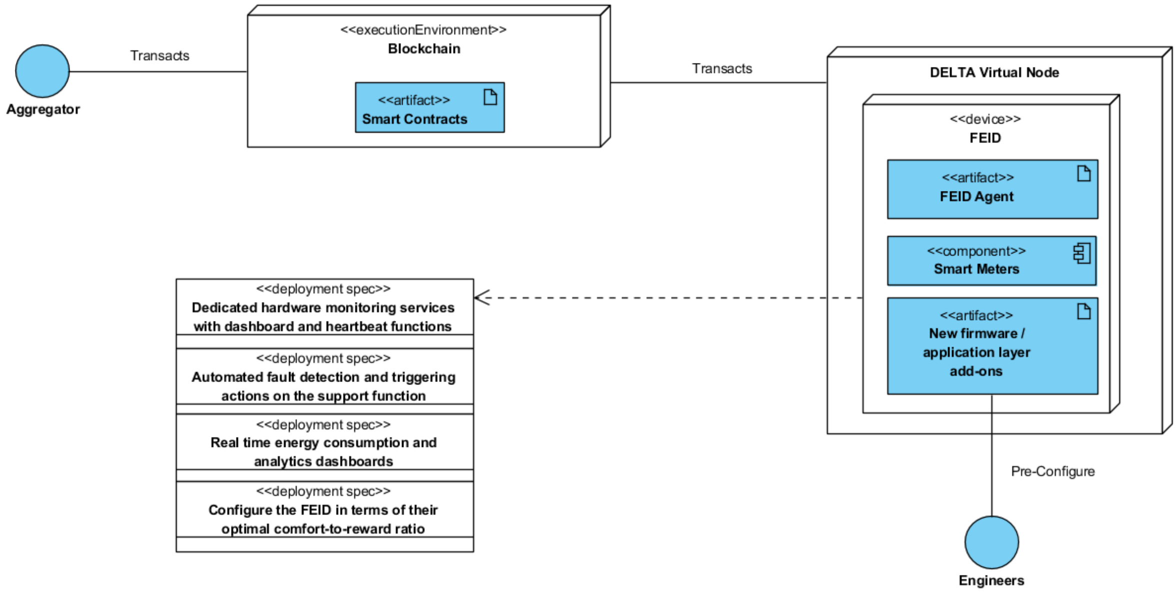

Figure 59 BS4 Deployment Diagram

8.4.1 DELTA BS4 – UC1: Securing communication between Smart Meters and DELTA Virtual Node Agents

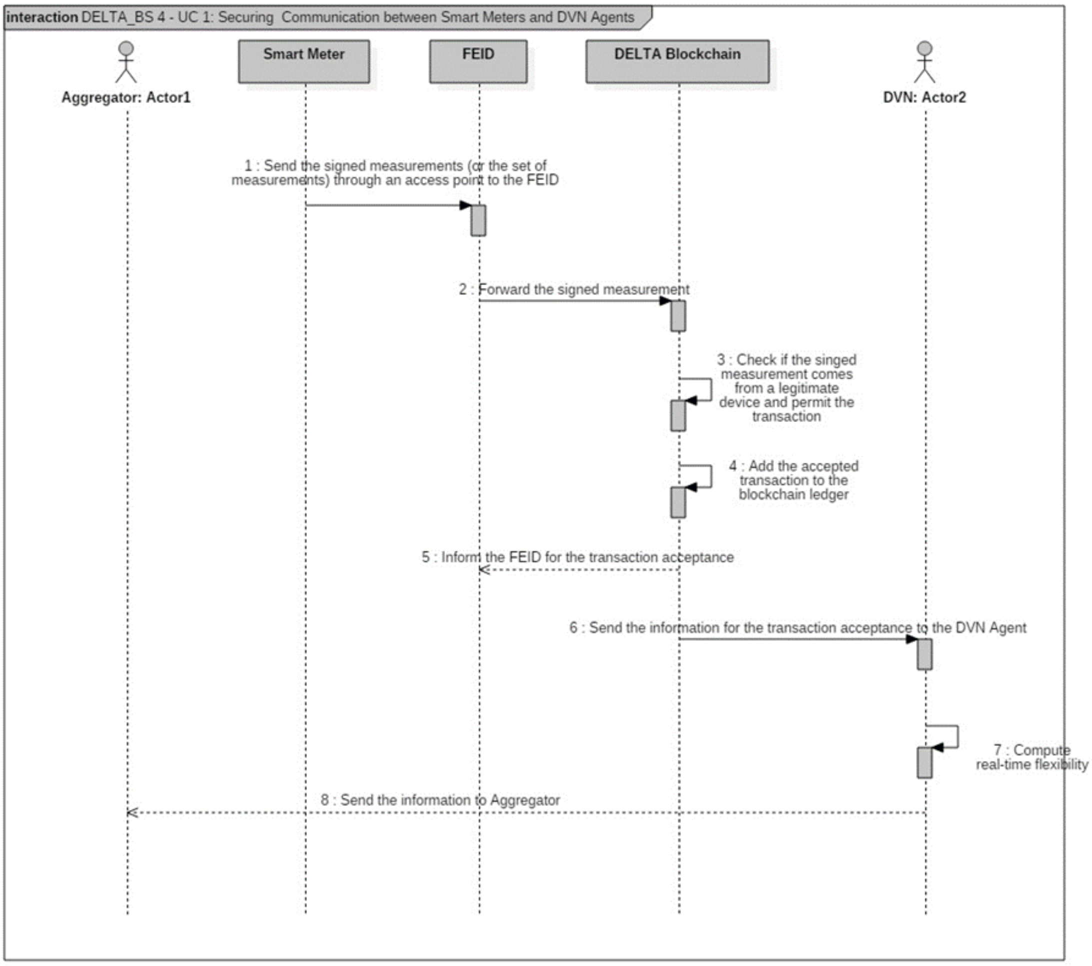

Figure 60. BS4 – UC1 Sequence Diagram

8.4.2 DELTA BS4 – UC2: Securing communication between Smart Meters and Fog Enabled Intelligent Device

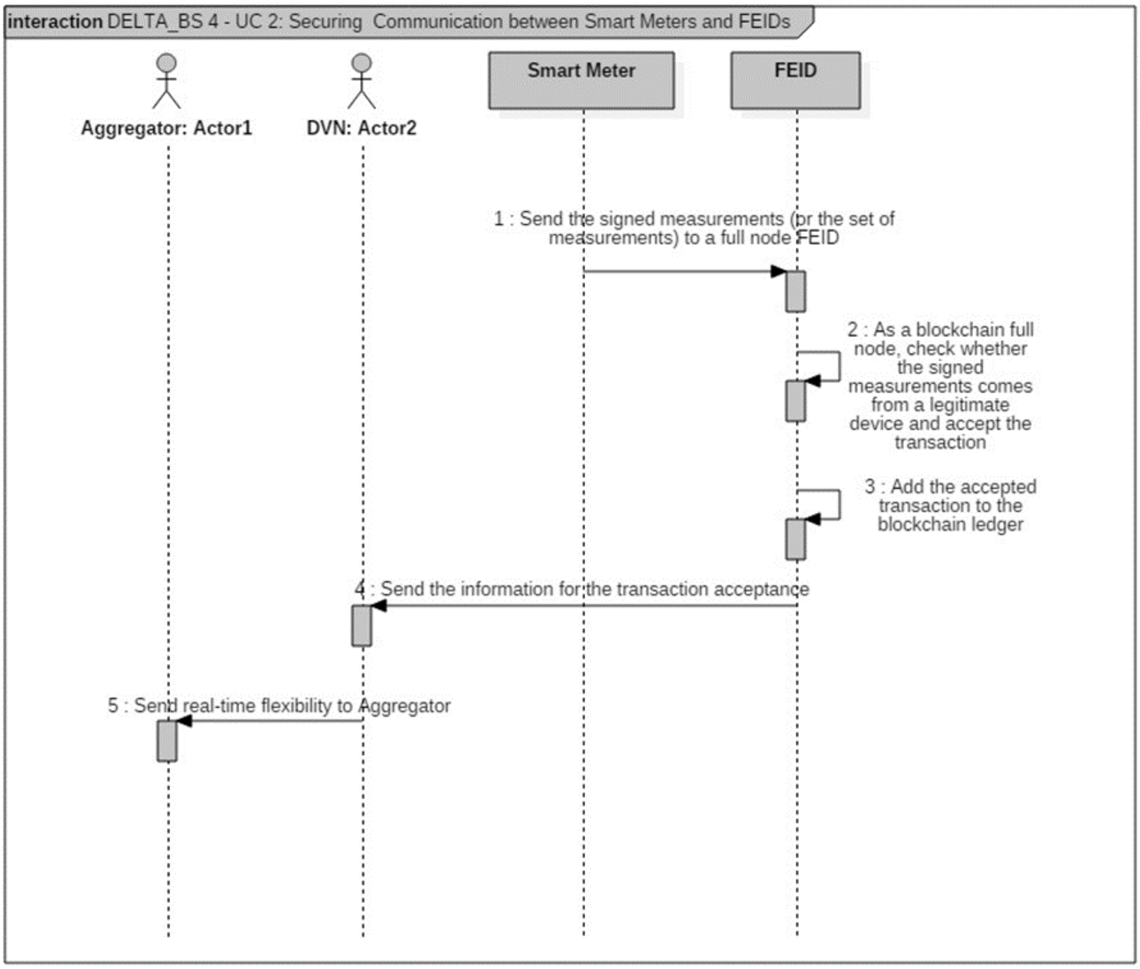

Figure 61. BS4 – UC2 Sequence Diagram

9. Conclusions

This report presents the final DELTA conceptual architecture, describing the system's main building blocks and giving a comprehensive overview of all modules, their high-level functionality and interdependencies.

The methodology that has applied to the Framework Design phase has been described. This provides a well-defined process and structure for describing the DELTA architecture. Following IEEE 1471 and Rozanski & Woods methodology, different viewpoints of the system architecture have been presented, including:

- The Functional View describing the system's functional elements, their responsibilities and primary interactions with other elements.
- The Information view, described in Chapter 6, defines the application domain models and the data flow as well as data distribution.
- The Deployment View, describing the modules' and existing software's hardware requirements.
- The Dynamic View (Use Case Analyses) presents the operations of components, their functionalities and interactions in the runtime environment.

The system requirements that frame the architectural problem and explicitly represent the stakeholders' needs and desires have been described. Functional and non-functional requirements have been carefully selected in order to ensure that they make sense in the context of the outcome of the project and conveyed to all the team members working on it.

As a result of applying this methodology to the DELTA system architecture definition process, the main building blocks of the system have been clearly identified and broken down into manageable modules, with clear responsibilities. In addition, missing components/subcomponents and corresponding functionalities within the original conceptual architecture were identified. Each responsible partner presented the components' internal architecture, functionalities, and interaction with other main components.

Following this approach, the DELTA architecture was defined and documented in four different views (namely: functional view, information view, deployment view and dynamic view) with close attention paid to the standard IEEE 1471 "Recommended Practice for Architectural Description for Software-Intensive Systems". There has been focus given to each component's role and their interaction. In particular, use cases identified within the DELTA deliverable D1.5, were selected and the related sequence diagrams developed by considering the identified components in a more architecturally framed context. This allowed further expressing the services dealt with by each component and the interactions between the components required to satisfy use cases.

10. References

[1] ISO/IEC/IEEE, "INTERNATIONAL STANDARD ISO / IEC / IEEE Systems and software engineering — agile environment," *ISO/IEC/IEEE 26515 First Ed. 2011-12-01; Corrected version 2012-03-15*, vol. 2012, 2011.

[2] E. Rozanski, N. and Woods, *Software systems architecture: working with stakeholders using viewpoints and perspectives*. Addison-Wesley, 2011.