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 D6.3
DELTA Integrated Prototype Framework v1

Work Package	WP6 – Work package title DELTA Integration & Added-Value Services
Tasks	T6.1 – Planning and Integration of individual components and overall DELTA Framework
Document Status:	Final Version for Peer Review
File Name:	DELTA_D6.3_PeerReview
Due Date:	30 April 2020
Submission Date:	May 2020
Lead Beneficiary:	CERTH

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**
Apostolos	Tsolakis	CERTH	tsolakis@iti.gr

Co-Author(s)				
#	**First Name**	**Last Name**	**Beneficiary**	**Contact e-mail**
1	Christos	Patsonakis	CERTH	cpatsonakis@iti.gr
2	Kostantinos	Kostopoulos	CERTH	kostopoulos@iti.gr
3	Ioannis	Koskinas	CERTH	jkosk@iti.gr
4	Andrea	Cimmino	UPM	cimmino@fi.upm.es
5	Juan	Cano-Benitez	UPM	jcano@fi.upm.es
6	Raúl	García-Castro	UPM	rgarcia@fi.upm.es
7	Alexandre	Lucas	JRC	Alexandre.LUCAS@ext.ec.europa.eu
8	George	Karagiannopoulos	HIT	g.karagiannopoulos@hit-innovations.com
9	George	Spathoulas	NTNU	georgios.spathoulas@ntnu.no
10	Venizelos	Venizelou	UCY	venizelou.venizelos@ucy.ac.cy
11	Alexis	Frangoullides	UCY	frangoullides.alexis@ucy.ac.cy
12	Phivos	Therapontos	EAC	ptherapo@eac.com.cy

Reviewers List

Reviewers			
First Name	**Last Name**	**Beneficiary**	**Contact e-mail**
Christof	Amann	e7	christof.amann@e-sieben.at
Alessio	Baiocco	NTNU	alessio.baiocco@ntnu.no

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

© DELTA. Copies of this publication – also of extracts thereof – may only be made with reference to the publisher.

Executive Summary

This report aims to provide detailed information regarding the implementation and integration status of the DELTA architecture components, both individually and in regards to the overall DELTA framework. The integration process follows an iterative procedure in which each component is thoroughly tested and evaluated based on expected outcomes and initial requirements towards presenting the optimal result. To support this, detailed information on each component's basic functionality, communication interfaces, implemented APIs, and User Interfaces is presented within this report. Furthermore, problems encountered during both development and integration are presented, to demonstrate what issues occurred and how they were countered on each step.

The DELTA integrated framework comprises of several components. Each component is described in details. Most of the DELTA components have been developed and are currently undergoing testing and integration to the overall framework, while also following scale up procedures to ensure viability in real-life conditions. Further development is required for certain components whereas intensive integration activities are foreseen in the following months towards presenting the complete framework for pilot deployment.

As even after pilot deployment further improvements are expected for the integrated DELTA framework, the development and integration processes will follow until the end of the project with a close to final version being documented in D6.4 in M32.

Table of Contents

1. Introduction ... 10

1.1 Scope and objectives of the deliverable ... 10
1.2 Structure of the deliverable ... 11
1.3 Relation to Other Tasks and Deliverables ... 11

2. Implementation and Integration status of Architecture Components ... 12

2.1 DELTA Customer ... 12
2.1.1 Fog-enabled Intelligent Device (FEID) ... 12
2.1.2 Simulated FEIDs ... 15
2.2 DELTA Virtual Node ... 17
2.2.1 Consumer/Prosumer Flexibility Data Monitoring and Profiling ... 17
2.2.2 Generation/Consumption Optimal Dispatch ... 18
2.2.3 Load Forecasting ... 19
2.2.4 Inter/Intra Node Energy Matchmaking ... 20
2.2.5 Consumer/Prosumer Energy/Social Clustering ... 21
2.3 DELTA Aggregator ... 22
2.3.1 Energy Market Price Forecast ... 22
2.3.2 DR & Flexibility Forecasting ... 24
2.3.3 Node Flexibility Data Monitoring and Profiling ... 25
2.3.4 Asset Handling Optimization ... 26
2.3.5 Self-Portfolio Energy Balancing ... 27
2.3.6 Energy Portfolio Segmentation & Classification ... 29
2.4 DELTA Grid State Simulation ... 30
2.4.1 Grid Stability Simulation Engine ... 30
2.5 Innovative Customer Engagement Tools ... 37
2.5.1 DR Visualisation Kit ... 37
2.5.2 Award–enabled Energy Behavioural Platform ... 38
2.5.3 Social Interaction and Cooperation Platform ... 40
2.6 Common Information Modelling ... 41
2.6.1 FEID CIM ... 41
2.6.2 DELTA Virtual Node CIM ... 42
2.6.3 Aggregator CIM ... 43
2.6.4 Security Services CIM ... 44
2.7 Cybersecurity Services ... 45
2.7.1 DELTA Blockchain ... 45
2.7.2 Smart Contracts ... 46
2.7.3 Smart Contract Gateway ... 47
DELTA Smart Contracts ... 47
2.7.4 Threat Mitigation ... 48

3. DELTA Components User Interfaces ... 50

3.1 DELTA Customer ... 50
3.1.1 Fog-enabled Intelligent Device ... 50
3.2 DELTA Virtual Node ... 58
3.2.1 Consumer/Prosumer Flexibility Data Monitoring and Profiling ... 58
3.2.2 Generation/Consumption Optimal Dispatch ... 58

3.2.3 Load Forecasting 60
3.2.4 Inter/Intra Node Energy Matchmaking 61
3.2.5 Consumer/Prosumer Energy/Social Clustering 61
3.3 DELTA Aggregator 62
3.3.1 Energy Market Price Forecast 62
3.3.2 DR & Flexibility Forecasting 62
3.3.3 Node Flexibility Data Monitoring and Profiling 62
3.3.4 Asset Handling Optimization 62
3.3.5 Self-Portfolio Energy Balancing 62
3.3.6 Grid Stability Simulation Engine 63
3.3.7 Energy Portfolio Segmentation & Classification 65
3.4 Innovative Customer Engagement Tools 66
3.4.1 DR Visualisation Kit 66
3.4.2 Award –enabled Energy Behavioural Platform 69
3.4.3 Social Interaction and Cooperation Platform 72
3.5 Common Information Modelling 72
3.6 Cybersecurity Services 74
3.7 DR Emulator 76

4. Conclusions 77

References 78

List of Figures

Figure 1 Conceptual architecture of the overall DELTA framework 10

Figure 2: Two core functionalities of SPEB 28

Figure 3: GSSE: Methodology for Overvoltage identification and restoration 32

Figure 4: GSSE: Methodology for Undervoltage identification and restoration. 33

Figure 5: GSSE: Methodology for Overload (due to increased consumption) identification and restoration. 34

Figure 6: GSSE: Methodology for Overload (due to increased production) identification and restoration. 35

Figure 7: FEID Wi-Fi access point 50

Figure 8: Installer UI starting screen 50

Figure 9: FEID detection and information about registered 51

Figure 10: FEID connection test 51

Figure 11: FEID devices list 52

Figure 12: Add Building Management System 52

Figure 13: Add direct control device ... 53

Figure 14: Device Type ... 53

Figure 15: Completed device list ... 54

Figure 16: Customer preferences ... 55

Figure 17: Customer general information ... 56

Figure 18: Customer UI, login page ... 56

Figure 19: Customer UI, main dashboard ... 57

Figure 20: Customer UI, user preferences ... 57

Figure 21: Customer UI, device management ... 57

Figure 22: Customer UI, historical DR messages ... 58

Figure 23: DVN Optimal Dispatch (OptiDVN) UI – Aggregator's view ... 59

Figure 24: DVN Optimal Dispatch (OptiDVN) UI – Load Dispatch Scenario – Explicit DR Users ... 59

Figure 25: DVN Optimal Dispatch (OptiDVN) UI – Time of Use Scenario – Explicit DR Users 60

Figure 26: Baseline Load Forecasting UI - Parameter Definition ... 60

Figure 27: Baseline Load Forecasting UI – Results ... 61

Figure 28: DELTA DVN Energy/Social Clustering UI ... 61

Figure 29: Aggregator Asset Handling Optimization UI – Parameterisation ... 62

Figure 30: Detailed model of the UCY campus power network. ... 63

Figure 31: Power Flow Analysis at the time of violation. ... 64

Figure 32: Quasi-Dynamic analysis – Feeder Active Power Flow. ... 65

Figure 33: DVNP UI – Available DVNs and their characteristics ... 65

Figure 34: Radial Tree representing both the Segmentation and the Clustering steps. ... 65

Figure 35: Customer General Information ... 66

Figure 36: Historical Generation and Consumption ... 67

Figure 37: Incoming and Outgoing DR Events ... 67

Figure 38: Customer Rewards and progress per game ... 68

Figure 39: Customer's DR Signals ... 68

Figure 40: FEID Energy Profile ... 68

Figure 41: Game Engine UI provided to the Aggregator ... 69

Figure 42: Manage Game Actions ... 69

Figure 43: Manage Game Awards ... 70

Figure 44: Manage Game Rules ... 70

Figure 45: Manage Game View current Leader board ... 71

Figure 46: Customer Gaming History ... 71

Figure 47: Collaboration Forum Aggregator's view ... 72

Figure 48: Collaboration Forum Customer's view ... 72

Figure 49 CIM UI – Main Dashboard – Connection to a CIM instance ... 73

Figure 50: CIM UI - Bridging service ... 73

Figure 51: CIM UI – Knowledge Graph service – SPARQL querying interface ... 74

Figure 52: DELTA Cybersecurity services – The Fabric Explorer ... 74

Figure 53: Smart contract visualization for a specific DR event from the Aggregator's point of view. ... 75

Figure 54: Smart contract visualization tool for DELTA's DR events from the customer's point of view ... 75

Figure 55: Web interface dashboard for monitoring identities (Aggregator's CA) ... 76

Figure 56: DR Emulator as an Aggregator Tool ... 76

List of Acronyms and Abbreviations

Term	Description
BMS	Building Management System
DR	Demand Response
DVN	DELTA Virtual Node
FEID	Fog Enabled Intelligent Device
GSSE	Grid Stability Simulation Engine
SPEB	Self-Portfolio Energy Balancing
DoS	Denial of Service
RDF	Resource Description Framework

1. Introduction

1.1 Scope and objectives of the deliverable

The main objective of this deliverable is to present the integrated DELTA framework, introducing the technical characteristics of the communication among different components and sub-components, as depicted in D1.6 “DELTA Overall Framework Architecture v2” and D1.7 “DELTA Information Model v2”. Towards this direction, the report introduces in detail the development and integration status of all DELTA technical entities for the seamless interoperation process among each other.

As integration is an iterative process, this deliverable documents the current status (M24) covering component by component both the back-end and front-end aspects.

The following figure depicts the high-level technical architecture of the DELTA framework as presented in D1.6, including the information flow.

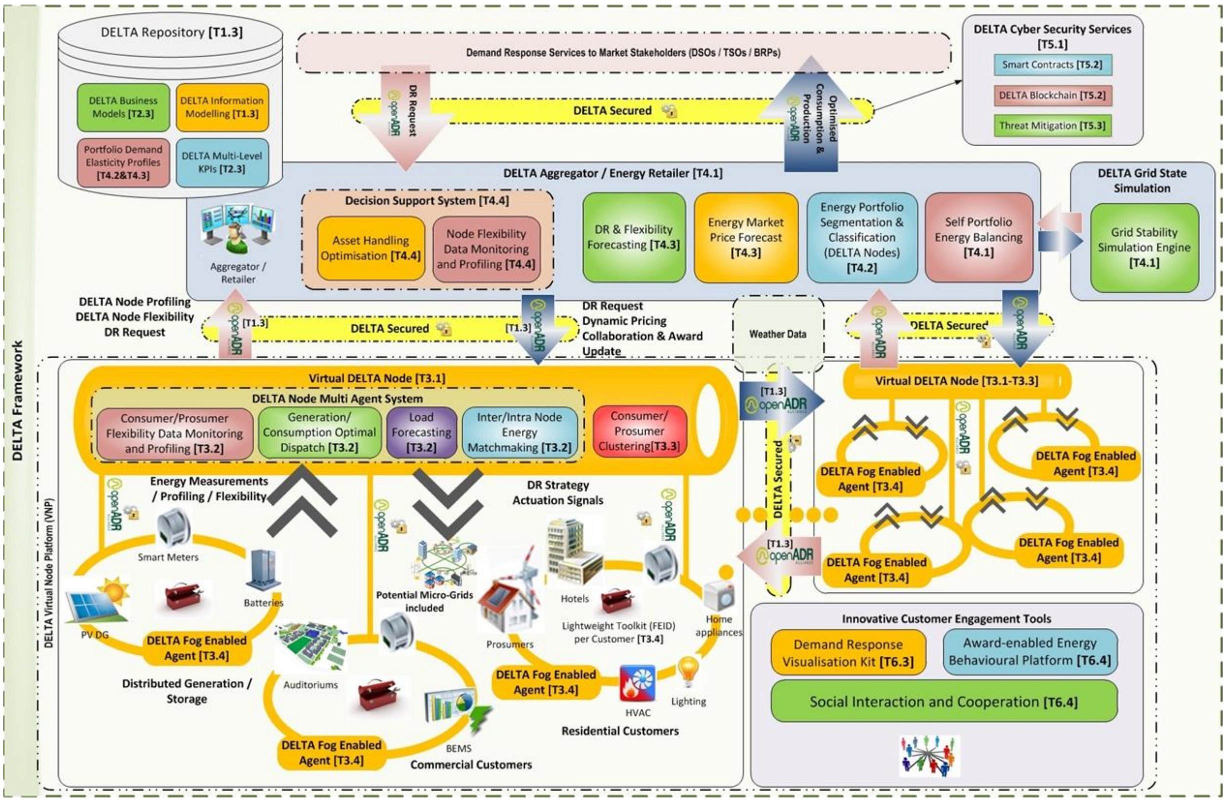

Figure 1 Conceptual architecture of the overall DELTA framework

The main components of the DELTA architecture that the present deliverable focuses on are as follows, divided in distinct layers, three vertical and two horizontal:

- DELTA Customer
  - Fog-Enabled Intelligent Device (FEID)
- DELTA Virtual Node (DVN)
  - Consumer/Prosumer Flexibility Data Monitoring and Profiling

  - Generation/Consumption Optimal Dispatch
  - Load/Generation Forecasting
  - Inter/Intra Node Energy Matchmaking
  - Consumer/Prosumer Clustering
- DELTA Aggregator
  - Decision Support System
    - Node Flexibility Data Monitoring and Profiling
    - Asset Handling Optimisation
  - Energy Market Price Forecast
  - Demand Response (DR) & Flexibility Forecast
  - Self-portfolio Energy Balancing (SPEB)
  - Energy Portfolio Segmentation and Classification
- DELTA Grid State Simulation
  - Grid Stability Simulation Engine (GSSE)
- Innovative Customer Engagement Tools
  - DR Visualisation Kit
  - Award-enabled Energy Behavioural Platform
  - Social Interaction and Cooperation

1.2 Structure of the deliverable

The deliverable following this section is structured as follows:

Section 2 presents the current development and integration status of each component;

Section 3 complements the integration of the supporting and functional User Interfaces (UIs); and

Section 4 concludes the report.

1.3 Relation to Other Tasks and Deliverables

This report is directly linked with all technical activities of WP3, WP4, WP5 and WP6, and presents how these interlink together forming the overall DELTA integrated framework, having WP1 requirements as guideline for the integration process.

2. Implementation and Integration status of Architecture Components

For every DELTA component that was defined in D1.2 (updated in D1.6), this section will introduce again, for the report to be self-contained, a brief functional description, how it is connected with the other DELTA components (if it is connected), the development and integration status into the whole integrated platform, and finally any problems encountered during the development and integration of the components.

Towards elaborating in an organised manner, and following the Architecture layout, the components are presented in three main layers: the customer, the virtual node and the aggregator.

2.1 DELTA Customer

This layer comprises of DELTA hardware and software components that are deployed at the customer's side and are built on the concept of fog computing. The DELTA customers can be any type of current Aggregators' customers, including small and medium size customers, consumers, producers, and prosumers.

2.1.1 Fog-enabled Intelligent Device (FEID)

One of the core components of DELTA from the customer's perspective is the Fog-enabled Intelligent Device (FEID). This physical device contains a lightweight toolkit that provides local intelligence and empowers the end-user with state-of-the-art real-time analysis and forecasting capabilities, while being also responsible for customer real-time monitoring and control.

2.1.1.1 Functional Description

As referred previously, FEID contains a lightweight toolkit that allows FEID to perform multiple functions. The supported functions from FEID are listed below:

- Real-time measurements
- Day-ahead forecasting
- Control of Loads/ DERs
- Demand-Response application
- User preferences

Real-time measurements

FEID is installed and integrated into smart or otherwise customers and communicates with existing smart technologies. It acquires real-time energy related data (e.g. active/reactive power, energy etc.) from multiple devices such as:

- Smart meters
- DER inverters/converters/chargers
- Building Management Systems (BMS)

All the data are collected from these devices in one minute time interval and are stored in a local time series database. The stored data are used both for the training of machine learning algorithms and for visualization purposes. In addition, FEID sends the aggregated energy related measurements, such as consumption, generation and flexibility to DELTA Virtual Node (DVN) in almost real-time. More specifically, the reporting frequency to DVN is one minute.

As FEID can be connected with a lot of different smart meters/sensors produced by different manufactures, it can support multiple communication protocols. The currently supported protocols are listed below. As will be presented in detail in D3.4 "Fog-enabled Intelligent Devices" the FEID has been designed supporting add-ons that enable communication capabilities with other protocols following ad-hoc requirements.

- Modbus TCP/ IP & RTU
- EnOcean
- Bluetooth
- LoRa

<u>Day-ahead forecasting</u>
FEID contains a variety of forecasting tools that are run on it daily. All the incorporated forecasting tools are able to "learn" from previous experience in order to correct next computational iteration in order to provide more accurate information to the DELTA Virtual Node.

- **Load Consumption:** The day ahead load forecast tool is a multi-step time series machine learning forecasting algorithm. The models that are used in this tool are based on FEID historical consumption values while taking into account a variety of features, like month, day of week, time of the day and their correlation.

- **Load flexibility:** The day ahead load flexibility tool calculates based on historical consumption values the available power that can be utilized in order to either achieve an optimal operation of the system or service incoming demand response schemes from the DELTA virtual node.

- **PV Generation (if existent)**: The day ahead PV generation tool combines a physical model that calculates the actual generation based on numerical weather forecast and PV plant technical specifications, and state-of-the-art machine learning algorithms that correct the error introduced by the limited accuracy of the online weather forecasting tools in the specific location of interested and using historical generation data.

<u>Control of Loads/DERs</u>
FEID instead of monitoring the smart devices in a specific infrastructure can also control them. The way of control depends entirely on the type of the installed devices. More specifically, it can directly control the smart devices and relays otherwise if a building management system (BMS) is already installed in the infrastructure; it controls the loads and the other assets in collaboration with it.

FEID incorporates a smart decision support system that automatically adjust the consumption/ production of all the assets in order achieve an optimal operation for the entire infrastructure taking into account the user predefined preferences. Nevertheless, the user can always intervene in FEID control and change the operation of specific devices.

DR Application

FEID is also capable of receiving and applying implicit or explicit Demand Response schemes coming from the upper layer. More specifically, it applies the DR set-points dispatched by OptiDVN (DVN Optimal Dispatch) module.

As a Demand Response message has arrived the FEID either responses automatically after performing a decision algorithm or informs the user in order to decide upon this request. Simultaneously the transaction is published in the blockchain system.

FEID is responsible for controlling all the devices that have been defined as controllable by the user in a time interval lower than one minute in order to achieve the target that was defined by DVN.

User preferences

The user can set up and change their preferences regarding the room temperature, room luminance and the devices that should be considered as uncontrollable. These preferences are always respected by FEID decision support system.

2.1.1.2 Connection with other Components and Interfaces

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
DELTA User Interface	TCP/IP	RESTful Services	JSON	
DELTA Installer App	TCP/IP	RESTful Services	JSON	
DELTA Virtual Node	TCP/IP	RESTful Services	JSON-LD	DELTA Ontology
DELTA Aggregator	TCP/IP	RESTful Services	JSON-LD	DELTA Ontology
Smart Meters	TCP/IP or RTU	Modbus	Raw	
Smart Devices	TCP/IP or Serial	Multiple Protocols	Depending on the Protocol	

2.1.1.3 Development Status

Development Status	□ Final √ Under development
Programming Language	Python
Progress up to date	A first version of all functionalities has been developed and has been deployed for evaluation. The final developments and experimental results will be documented in deliverable D3.4
Pending Development Actions	Further refinement of individual software performance / accuracy. More generic implementation to the communication with multiple hardware devices (i.e. smart meters, inverters, etc.), as well as BMS

2.1.1.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	RESTful Services
Progress up to date	Completed tests regarding hardware communication with all layers.
Pending Integration Actions	Extend connection to BMS, converter all intra DELTA components messages to JSON-LD and ensure communication through the CIM, extended tests related to Demand Response schemes

2.1.1.5 Hardware/Software Problems Encountered

Problem Type	*Problem Description*	*Problem Cause*	*Countermeasure*
Data Loss due to linear programming	During the day-ahead schedule were all forecasting algorithms are executed, the were significant delays in data retrieval	Each forecasting algorithms requires a few seconds to present results. Initially the scheduler implemented was executing each algorithm following a linear approach which added the time delays, presenting a prolonged execution time	The scheduler was changed to a parallel approach running all forecasting components simultaneously.
Data Loss due to smart meter protocol restrictions	Data loss has been observed during specific timeframes within the day from multiple smart meters directly connected to the FEID through Modbus RTU	The problem was caused by the Modbus communication library used to extract data from both devices at the same time	A different library was selected that allowed in parallel communication to more than one devices at the same time.

2.1.2 Simulated FEIDs

2.1.2.1 Functional Description

To be able to test both individual and integrated components it is imperative to perform large scale testing. Towards that direction a simulation engine has been developed to simulate the performance of multiple FEIDs. The engine deploys a FEID just like the real case and then generates data and handles DR following the same principles as the actual thing, but with a predefined reliability.

2.1.2.2 Connection with other Components and Interfaces

Component	Connection Type	API Protocol	Data Type	Comments
DELTA User Interface	TCP/IP	RESTful Services	JSON	-
DELTA Virtual Node	TCP/IP	RESTful Services	JSON-LD	DELTA Ontology
DELTA Aggregator	TCP/IP	RESTful Services	JSON-LD	DELTA Ontology
Simulated FEID Local Repository	TCP/IP	RESTful Services	PostgreSQL queries	Retrieving customer models' data

2.1.2.3 Development Status

Development Status	□ Final √ Under development
Programming Language	Python
Progress up to date	First version of simulated FEIDs has been delivered yet some problems have been encountered
Pending Development Actions	Further development is needed to support robust functionality of the simulation engine

2.1.2.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	Python package
Progress up to date	Complete integration with custom data models
Pending Integration Actions	Integration with the DELTA CIM

2.1.2.5 Hardware/Software Problems Encountered

Problem Type	Problem Description	Problem Cause	Countermeasure
Not possible to scale up	Simulated FEIDs are not reporting correct values after a number and there seems to be a memory issue for each simulated FEID	Pending further investigation	Detailed debugging and further development.

2.2 DELTA Virtual Node

2.2.1 Consumer/Prosumer Flexibility Data Monitoring and Profiling

2.2.1.1 Functional Description

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, while also supporting communication for data exchange with other DVNs or the Aggregator upon request. The same component is also responsible for storing data locally to the DVN repository.

2.2.1.2 Connection with other Components and Interfaces

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
FEID	TCP/IP	CIM	JSON-LD	DELTA Ontology
DVN	TCP/IP	CIM	JSON-LD	DELTA Ontology
Aggregator	TCP/IP	CIM	JSON-LD	DELTA Ontology
DVN Repository	TCP/IP	RESTful Services	PostgreSQL queries	Based on the local DB format

2.2.1.3 Development Status

Development Status	√ Final □ Under development
Programming Language	Python
Progress up to date	The final developments and experimental results will be documented in deliverable D3.2
Pending Development Actions	None.

2.2.1.4 Integration Status

Integration Status	√ Final □ Under development
Format for Integration	Python code within the DVN MAS
Progress up to date	Custom integration completed
Pending Integration Actions	Integration with the DELTA CIM

2.2.1.5 Hardware/Software Problems Encountered

None.

2.2.2 Generation/Consumption Optimal Dispatch

2.2.2.1 Functional Description

The Generation/Consumption Optimal Dispatch (OptiDVN) aims at establishing energy decisions that the FEIDs selected must fulfil. This component takes from the DELTA Aggregator/Energy Retailer the supplied DR Signals, computes the optimal course that the underneath DELTA FEIDs should take, solving the unit commitment problem and emits DR Signals to them.

2.2.2.2 Connection with other Components and Interfaces

Component	***Connection Type***	***API Protocol***	***Data Type***	***Comments***
DVN Multi Agent System	Python script	-	JSON	The DVN MAS provides the OptiDVN with the DR signal and the available FEIDs (after assessing the clustering results) and retrieving the outcome sends the resulting DRs to the FEIDs through the CIM
DVN Local Repository	TCP/IP	RESTful Services	PostgreSQL queries	Based on the local DB format for reading forecasted data. Also stores new results to be available for other components.

2.2.2.3 Development Status

Development Status	□ Final √ Under development
Programming Language	Python
Progress up to date	Close to final version available, deployed and tested in real-life conditions
Pending Development Actions	Important extensions have been identified and are needed to further improve the optimization results. A wider service of DR signals is also required

2.2.2.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	Python Script
Progress up to date	Complete integration of the current version available
Pending Integration Actions	Further integration is required with the MAS to support the clustering results and also the new extensions identified – such as computing energy and not power related DRs, servicing more OpenADR signal types, and more.

2.2.2.5 Hardware/Software Problems Encountered

Problem Type	***Problem Description***	***Problem Cause***	***Countermeasure***
Long computation time	Scaling up, the computation time exceeds the acceptable limits	One of the main problems identified are the data structures used, including some issues in the coding itself	Code optimisation has been performed to support more acceptable execution times. Further testing is required.

2.2.3 Load Forecasting

2.2.3.1 Functional Description

The Load Forecasting sub-component is a pillar element to handle the FEIDs that a DVN is in charge of. It provides the basis for flexibility forecast of the expected loads. For the DVN layer this tool has some key functionalities: Aggregate FEID forecasts, forecast directly aggregated DVN measurements, compute a weighted comparison and provide the optimal results as output for the DVN forecasting.

2.2.3.2 Connection with other Components and Interfaces

Component	***Connection Type***	***API Protocol***	***Data Type***	***Comments***
DVN Local Repository	TCP/IP	RESTful Services	PostgreSQL queries	Based on the local DB format for reading historical data for re-training and executing. Also stores new results to be available for other components.
Consumer/Prosumer Flexibility Data Monitoring and Profiling	Python package	-	Data Array	The profiling calls the Load Forecasting for retrieving the forecasts and feed them to the Aggregator or other DVNs

2.2.3.3 Development Status

Development Status	□ Final √ Under development
Programming Language	Python
Progress up to date	Load Forecasting at DVN level (presenting both positive and negative values for covering consumption and generation has been developed, deployed and tested.
Pending Development Actions	Further testing and improvements are required to improve the accuracy of the final outcome. Adaptive re-training of both the models and the weighted comparison is also required.

2.2.3.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	Python Package
Progress up to date	Current version has been already implemented in order to be evaluated
Pending Integration Actions	Integrate future component versions

2.2.3.5 Hardware/Software Problems Encountered

None.

2.2.4 Inter/Intra Node Energy Matchmaking

2.2.4.1 Functional Description

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

2.2.4.2 Connection with other Components and Interfaces

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
DVN MAS	Python script	-	JSON	The DVN MAS provides the Matchmaking with the shortage / failure data and the remaining available FEIDs (after assessing the clustering results) and retrieving the outcome sends the resulting DRs to the FEIDs/DVNs through the CIM
DVN Local Repository	TCP/IP	RESTful Services	PostgreSQL queries	Based on the local DB format for reading forecasted data. Also stores new results to be available for other components.

2.2.4.3 Development Status

Development Status	□ Final √ Under development
Programming Language	Python
Progress up to date	Both inter and intra matchmaking approaches have been developed and been provided for deployment and testing
Pending Development Actions	Further development is expected after preliminary large scale integration testing

2.2.4.4 Integration Status

Integration Status	√ Final □ Under development
Format for Integration	Python Package
Progress up to date	The package has been provided for integration. It is currently under intensive deployment within the DVN MAS for testing.
Pending Integration Actions	To accurately evaluate the performance of this component a larger scale deployment is required. Upon integration completion of both the Simulated FEIDs and the Clustering sub-components, the evaluation will be performed running multiple scenarios

2.2.4.5 Hardware/Software Problems Encountered

None.

2.2.5 Consumer/Prosumer Energy/Social Clustering

2.2.5.1 Functional Description

Consumer/Prosumer Energy/Social Clustering engine endeavours to identify groups of FEIDs that have common Energy/Social behaviour. This grouping can be harnessed from other DVN's components in order to distribute the DR signals with an optimized way. In more detail, the Clustering tool is a dynamic process that formulates groups of FEIDs per hour according to a combination of Flexibility and reliability Behaviour. In addition, Clustering engine tries to adapt the Algorithms Parameters according to the efficiency of the algorithm in a way that it can provide more informative insight to other DVN's components. Finally, the engine extracts descriptive features for each cluster in order to offer assistive services towards the Optimal Dispatch (OptiDVN) engine.

2.2.5.2 Connection with other Components and Interfaces

Component	***Connection Type***	***API Protocol***	***Data Type***	***Comments***
DVN	Package	-	JSON	Clustering Results are saved in DVN's Database and are used as input to the Optimal Dispatch and Matchmaking sub-components

2.2.5.3 Development Status

Development Status	□ Final √ Under development
Programming Language	Python
Progress up to date	Clustering based on energy features has been completed providing both temporal and spatial clusters
Pending Development Actions	Social clustering is currently being implement. The analysis will cover the customers' engagement status and will create clusters that will better respond to different engagement strategies.

2.2.5.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	Python Package
Progress up to date	Integration has just initiated
Pending Integration Actions	Complete integration of both clustering approaches.

2.2.5.5 Hardware/Software Problems Encountered

None.

2.3 DELTA Aggregator

2.3.1 Energy Market Price Forecast

2.3.1.1 Functional Description

The developed model considers the Elexon balancing market in the UK and aims at predicting with fair accuracy the balance energy price for the present day. The output will be 48 price predictions, each corresponding to settlement periods of half an hour. The accuracy of the regressions is not crucial as an absolute value, since the physical notification from the parties/aggregators to the market, do not contain any price indication. More importantly is the direction of the price, either if it has a tendency of going up or down in the following settlement periods. This will provide an indication of when is the right time to assign the assets in the portfolio. The model uses the well-known XGBoost machine learning algorithm and considers 15 forecasted variables: Settlement Period, 12h LoLP, 12h DRM, 8h LoLP, 8h DRM, 4h LoLP, 4h DRM, 2h LoLP, 2h DRM, 1h LoLP, 1h DRM, Production, Solar generation; Wind generation, Transmission System demand forecast. The model reads directly from different URLs the required data. Then, when ran, it is able to provide a list with 48 price predictions. The model is able to cope with missing values. Since some variables such as the LOLP 1h will only provide a value 1 hour ahead of gate closure, the highest accuracy will be achieved when the model is ran at this hour (closer to the gate closure).

Example of the model output:

The expected balancing energy price for the Day per SP is: *[10.235224 41.071392 21.586876 39.62035 42.84652 40.751232 41.83968 30.06947 26.314615 31.536865 25.5027 37.781624 27.479837 42.756287 43.752533 28.62737 52.44014 55.73102 54.122536 46.675137 33.941055 31.47266 31.105272 32.734436 35.908123 35.888165 35.907192 39.41408 32.981533 33.860718 32.714203 38.827602 42.111267 38.20441 41.272614 37.914246 52.128315 39.057163 63.512775 59.373005 56.56427 55.67102 36.085934 53.00347 50.041164 31.35972 29.303764 37.424706] £/MWh.*

2.3.1.2 Connection with other Components and Interfaces

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
Internet (ex: Elexon page)	TCP/IP	HTML	HTML	Scrapping of data done in the python script
Decision support System	Python script	-	Data Array	The script can be integrated into another tool as desired, or other formats such as a CSV file for example

2.3.1.3 Development Status

Development Status	□ Final √ Under development
Programming Language	Python
Progress up to date	The script, is capable of training, testing, retrieving data from the internet and generate an output o price. The complete cycle is completed.
Pending Development Actions	Net Imbalance volume is being considered as an average value. Its forecasted value still to be integrated. The script needs to be put in a user friendly format

2.3.1.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	Python script
Progress up to date	Python script has just been made available and integration has started. Making script user friendly
Pending Integration Actions	Net imbalance volume needs to be forecasted in order to provide the model with high accuracy

2.3.1.5 Hardware/Software Problems Encountered

None.

2.3.2 DR & Flexibility Forecasting

2.3.2.1 Functional Description

This sub-component 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. By evaluating input from the GSSE, as well as information provided from both the DVNs through the Node Flexibility Data Monitoring and Profiling, as well as other Aggregator's components, such as the Decision Support System (DSS), it provides an aggregated estimation of the available flexibility for any given timeslot as well as a probability of incoming or generally needed DR requests.

2.3.2.2 Connection with other Components and Interfaces

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
Node Flexibility Data Monitoring and Profiling	Internal Endpoint	RESTfull Services	JSON	Retrieval of both DVN forecasts and stored historical information from the DELTA repository.
GSSE	CIM	RESTfull Services	JSON-LD	-

2.3.2.3 Development Status

Development Status	□ Final √ Under development
Programming Language	Python
Progress up to date	Aggregation of DVN information has been completed. Initial analysis of historical DR signals has been performed
Pending Development Actions	Further analysis on historical data is required for estimating DR probability on top of GSSE input.

2.3.2.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	Python code – integration to the DSS
Progress up to date	Flexibility Aggregation has been integrated
Pending Integration Actions	Complete integration with the GSSE and the respective DR forecasting analysis.

2.3.2.5 Hardware/Software Problems Encountered

None.

2.3.3 Node Flexibility Data Monitoring and Profiling

2.3.3.1 Functional Description

Node Flexibility Data Monitoring and Profiling (NFDM&P) component is responsible to inspect, record and transmit to the upper layer: real time, historical and forecasted measurements about: Consumption, Flexibility, Generation and Profiling data flow of each DVN. It consists the intermediate component between the Aggregator and the DVNs through their CIM communication, in compliance with OpenADR communication protocol. In that way, NFDM&P provides data towards Asset Handling Optimization, GSSE and SPEB component in order to act cooperatively and plan efficient Demand Response strategies.

2.3.3.2 Connection with other Components and Interfaces

Component	Connection Type	API Protocol	Data Type	Comments
DVN	TCP/IP	CIM	JSON-LD	Real Time, Historical and Forecasted Data are transferred through the CIM in accordance with the OpenADR compliance.
AHO	Python Package	-	JSON	Both Sub-Components as parts of the Aggregator Entity.
GSSE	TCP/IP	CIM	JSON-LD	GSSE receives useful information about forecasted measurements
SPEB	Package	-	JSON	Both Sub-Components as parts of the Aggregator Entity

2.3.3.3 Development Status

Development Status	√ Final □ Under development
Programming Language	Python
Progress up to date	All functionalities required up to M24 have already been implemented, deployed and tested
Pending Development Actions	Depending on future integration with other components.

2.3.3.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	Python Script
Progress up to date	Most available components up to M24 have been integrated. All communications within the Aggregator layer have been tested and validated.
Pending Integration Actions	Pending integration with the DELTA CIM

2.3.3.5 Hardware/Software Problems Encountered

None.

2.3.4 Asset Handling Optimization

2.3.4.1 Functional Description

Asset Handling Optimization (AHO) consists a core module of Aggregator's Decision Support System. It combines information from several aggregator's subcomponents as well as information received from lower Delta layers in order to define how to handle each Portfolio element in the optimally efficient way via dynamic DR strategies.

2.3.4.2 Connection with other Components and Interfaces

Component	Connection Type	API Protocol	Data Type	Comments
Node Flexibility Data Monitoring and Profiling	TCP/IP	RESTful Services	JSON	Sends Data about Historical Consumption, Generation, Flexibility per DVN
Self Portfolio Energy Balancing	TCP/IP	RESTful Services	JSON	Selects DVNs that will join a DR
Grid Stability Simulation Engine	TCP/IP	CIM	OpenADR	Informs about Grid instabilities
DR & Flexibility Forecasting	TCP/IP	RESTful Services	JSON	Sends forecasting flexibility values

2.3.4.3 Development Status

Development Status	□ Final √ Under development
Programming Language	Python
Progress up to date	Distribution of incoming DR requests both explicit and implicit based on Node available flexibility.
Pending Development Actions	Develop a DR strategy that will exploit forecasted Markets Pricing within the Imbalance market context.

2.3.4.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	Python script
Progress up to date	Communication with most components has been completed and tested in real-life conditions
Pending Integration Actions	Final integration with remaining components, update integration status with final versions, deploy final version for lab testing. Integrate CIM for communication with other DELTA components/layers

2.3.4.5 Hardware/Software Problems Encountered

None.

2.3.5 Self-Portfolio Energy Balancing

2.3.5.1 Functional Description

Self – Portfolio Energy Balancing (SPEB) component is part of the DELTA Aggregator platform and aims at prioritizing the possible combinations of market settlements that specify a certain behavior that the DELTA Aggregator/Energy Retailer should consider behaving. This output entails every asset that is available to offer flexibility along with the associated market in order to deliver either a specific level of power from generators or a reduction of demand.

Essentially, the two core functionalities of this component are:

1. To generate all combinations of available DVNs, that can meet the total requested flexibility, considering the flexibility compensation price (€/kWh) as well as a penalty fee for non or insufficient delivery of the reserved flexibility. The flexibility compensation price as well as penalty fees vary across the customers based on the available flexibility volume and the respective participation market. The derived combinations are prioritized based on the most profitable options for the Aggregator as well as two other major indices, reliability (RI) and fairness (FI). Reliability Index (%) is defined as the ratio of the amount of provided flexibility over the total requested one, while Fairness Index (%) ensures that flexibility requests are equally distributed to all customers under the Aggregator's portfolio based on previous DR activations.
2. The second functionality updates the two factors, RI and FI, for the DVNs that actually participated in the latest DR request.

The two core functionalities of the SPEB component are depicted in the following figure.

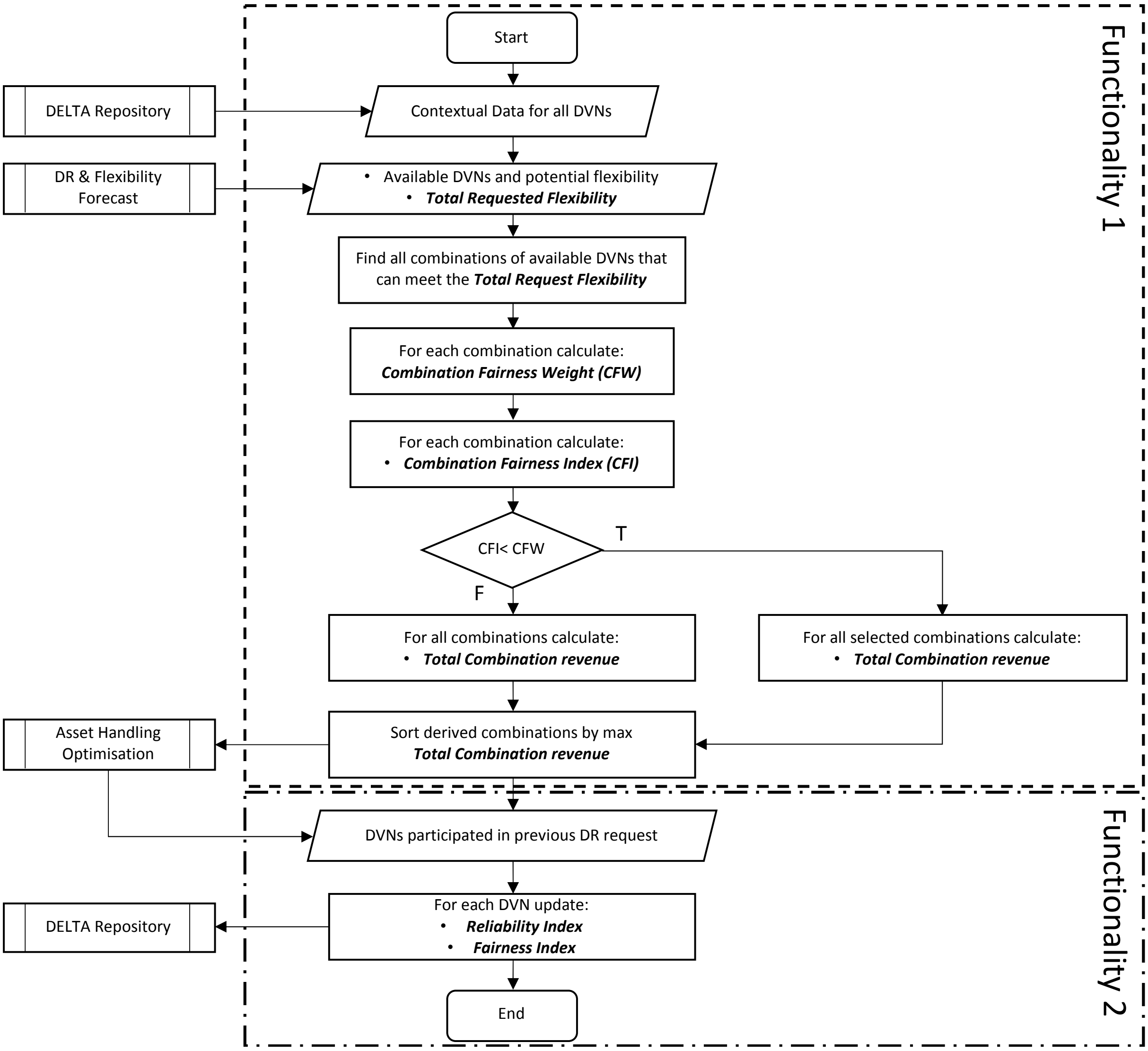

Figure 2: Two core functionalities of SPEB

All possible combinations of all available DVNs that can meet the Total Requested Flexibility are derived based on the data provided by the "DR & Flexibility Forecast" component. The combinations are then limited to the ones with the fewer participations based on a comparison between the Combination Fairness Index (sum of all FI in a combination) and the Combination Fairness Weight (sum of all DVN weights).

2.3.5.2 Connection with other Components and Interfaces

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
DR & Flexibility Forecast	Input Function Variables	-	List	Input Data include: • available assets and estimated provided flexibility Total requested flexibility and respective market
Repository	CIM ()	-	List	Input Data for each asset include:

				• Compensation flexibility price • Penalty fee • Participation Market • Reliability Index Fairness Index
Asset Handling Optimization	Input Function Variables	-	List	Output: Prioritized combinations of all applicable assets
Repository	CIM ()	-	List	Output: • Updated Reliability Index Updated Fairness Index

2.3.5.3 Development Status

Development Status	√ Final □ Under development
Programming Language	Python
Progress up to date	The final developments and experimental results have been documented in deliverable D4.1.
Pending Development Actions	Refinement based on integration results if needed.

2.3.5.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	Python script to executed by the DELTA DSS when needed
Progress up to date	Preliminary integration successful.
Pending Integration Actions	Integrate and test under real-life conditions the final version provided.

2.3.5.5 Hardware/Software Problems Encountered

None.

2.3.6 Energy Portfolio Segmentation & Classification

2.3.6.1 Functional Description

Energy Portfolio Segmentation & Classification engine is responsible to distribute the assets (FEIDs) between the available DVNS in a dynamic and coherent way in order to generate robust and independent entities that they will facilitate an optimized DR strategy. Segmentation engine, except from distributing the available assets, it is responsible as well for the assignment of a new FEID in a DVN group according to their fundamental contract characteristics. This temporal reallocation of FEIDs among the DVNs focuses on maintaining a relative balance and stability in the DVNs’ functionality, maximizing the DR strategies efficiency.

2.3.6.2 Connection with other Components and Interfaces

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
DVNs	TCP/IP	RESTfull Services (through CIM)	JSON-LD	Assign feids to DVNS in a progressive way and inform DVNs about the new entities.
FEIDs	TCP/IP	RESTfull Services (through CIM)	JSON-LD	Receive information about a new Feid and assign it to a DVN.
GSSE	TCP/IP	RESTfull Services (through CIM)	JSON-LD	Receive indications about instabilities from GSSE.
NFDM&P	TCP/IP	RESTfull Services	JSON	Receive information about the DVN profiles.

2.3.6.3 Development Status

Development Status	□ Final √ Under development
Programming Language	Python
Progress up to date	Fully functional version available for both Segmentation and Classification purposes
Pending Development Actions	Further features are to be evaluated for improving the segmentation performance. Re-triggering this component and the dynamic re-construction of the Virtual Network

2.3.6.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	Python Package
Progress up to date	Complete integration of the current Version
Pending Integration Actions	Integration of the final version when it becomes available.

2.3.6.5 Hardware/Software Problems Encountered

None.

2.4 DELTA Grid State Simulation

2.4.1 Grid Stability Simulation Engine

2.4.1.1 Functional Description

Future Aggregators should be able to monitor the entire distribution network under its supervision and identify conditions that may affect his bidding and operation strategies. The objective of the Grid Stability Simulation Engine (GSSE) component is to quantify DR-related grid constraints and identify potential risks of assets connected to power network that is under the Aggregator's portfolio. Essentially, GSSE considers the balanced aggregated profile for real-time as well as Day-Ahead markets to identify potential physical constraints through load flow calculations that are used to analyse power systems under steady-state non-faulted (short-circuit-free) conditions. The load flow simulator used here is the Powerfactory - DIgSILENT, which is a leading power system analysis software application for analysing generation, transmission and distribution systems.

The grid topology is based on data and information provided by the local Distribution Systems Operator (DSO) as it is anticipated in their new role as neutral market facilitators, DSOs will make available pertinent information to the competent players [1]-[3].

The two core functionalities of GSSE are:

- Participation in Flexibility markets: To check if possible flexibility provision can lead to potential grid violation.
- Pre-DR signal: Predict an upcoming DR event (day-ahead) that can occur due to a specific grid constraint violation: The required flexibility must be met by the assets associated to the violation (based on their location and connection to the grid).

In both cases the required flexibility that will maintain the stability of the investigate grid is estimated by the GSSE component. The required flexibility volume is fed to the "DR & Flexibility Forecasting" component, while the violation levels as well as a graphical representation of their location in the grid are outputted to the "Asset Handling Optimization".

The following figures (Figure 3 to Figure 6) illustrate the methodology followed for identifying and restoring unbalanced grid conditions.

In Study Cases 1 and 2 there is a predefined upper and lower voltage limit (Grid rules) where the voltage of the equipment should not surpass at any time. In case that the voltage is higher than the upper or below than the lower limit then an overvoltage or undervoltage event occurs, respectively. In order to evaluate the amount of flexibility needed to restore the nominal conditions, the GSSE calculates the ratios of $\delta V/\delta P$ and $\delta V/\delta Q$ (deviation of voltage in relation to active and reactive power) at the violated busbar. An overvoltage event can be overcome by increasing the active power, or equally requesting upwards flexibility. Correspondingly, an undervoltage event can be regulated through active power reduction.

Study Cases 3.1 and 3.2 summarize the actions followed when the loading (of any equipment) exceeds a predefined loading limit, thus an overload event occurs. GSSE estimates the amount of flexibility required to mitigate the overload, by calculating the percentage difference between the nominal equipment rating and the expected loading. The study case 3.1 involves the methodology followed when overload occurs due to increased load and therefore can be mitigated by reducing the active power (downwards flexibility provision), while study case 3.2 involves the actions followed when increased generation leads to an overload event and therefore the active power must be increased (upwards flexibility provision).

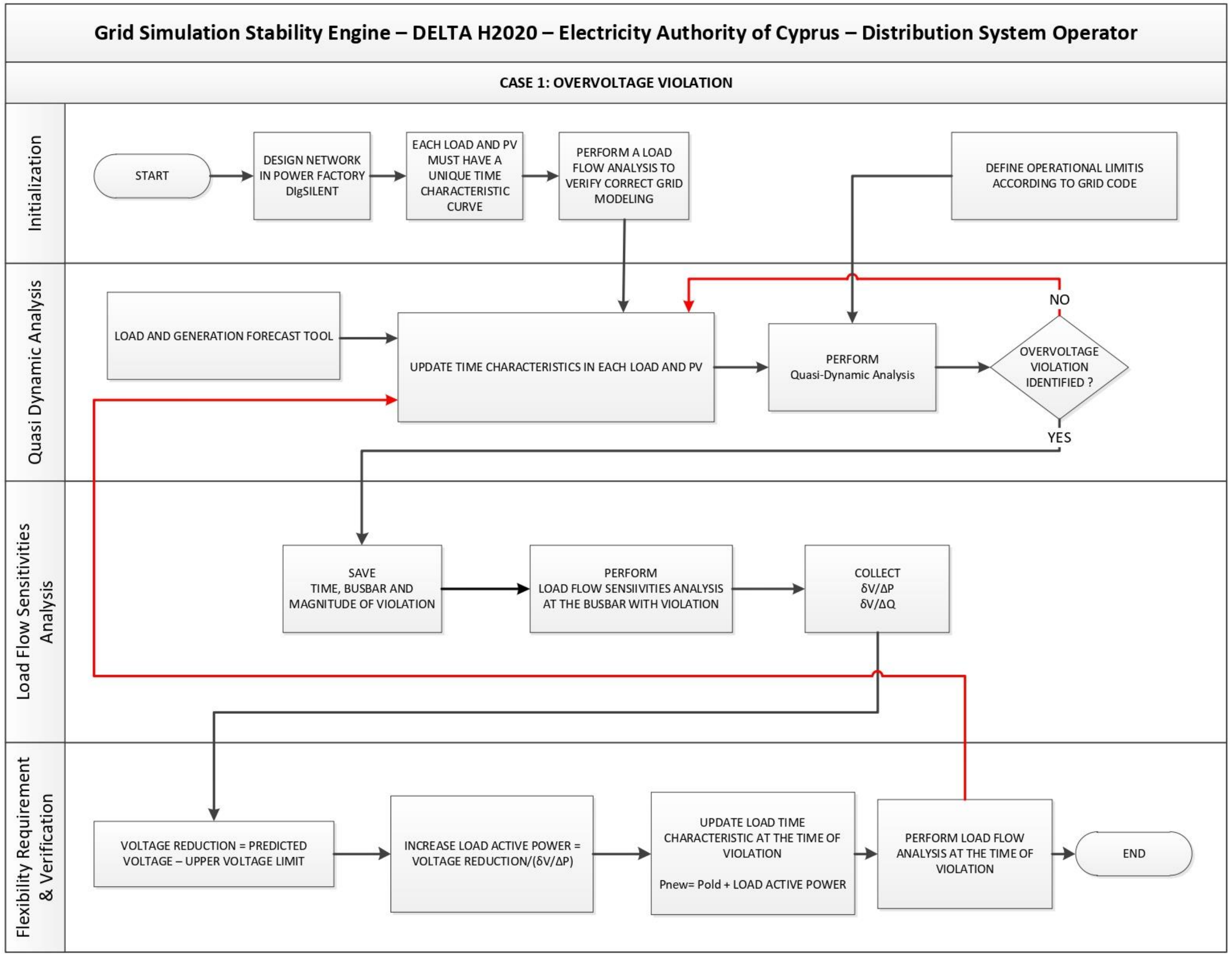

Figure 3: GSSE: Methodology for Overvoltage identification and restoration

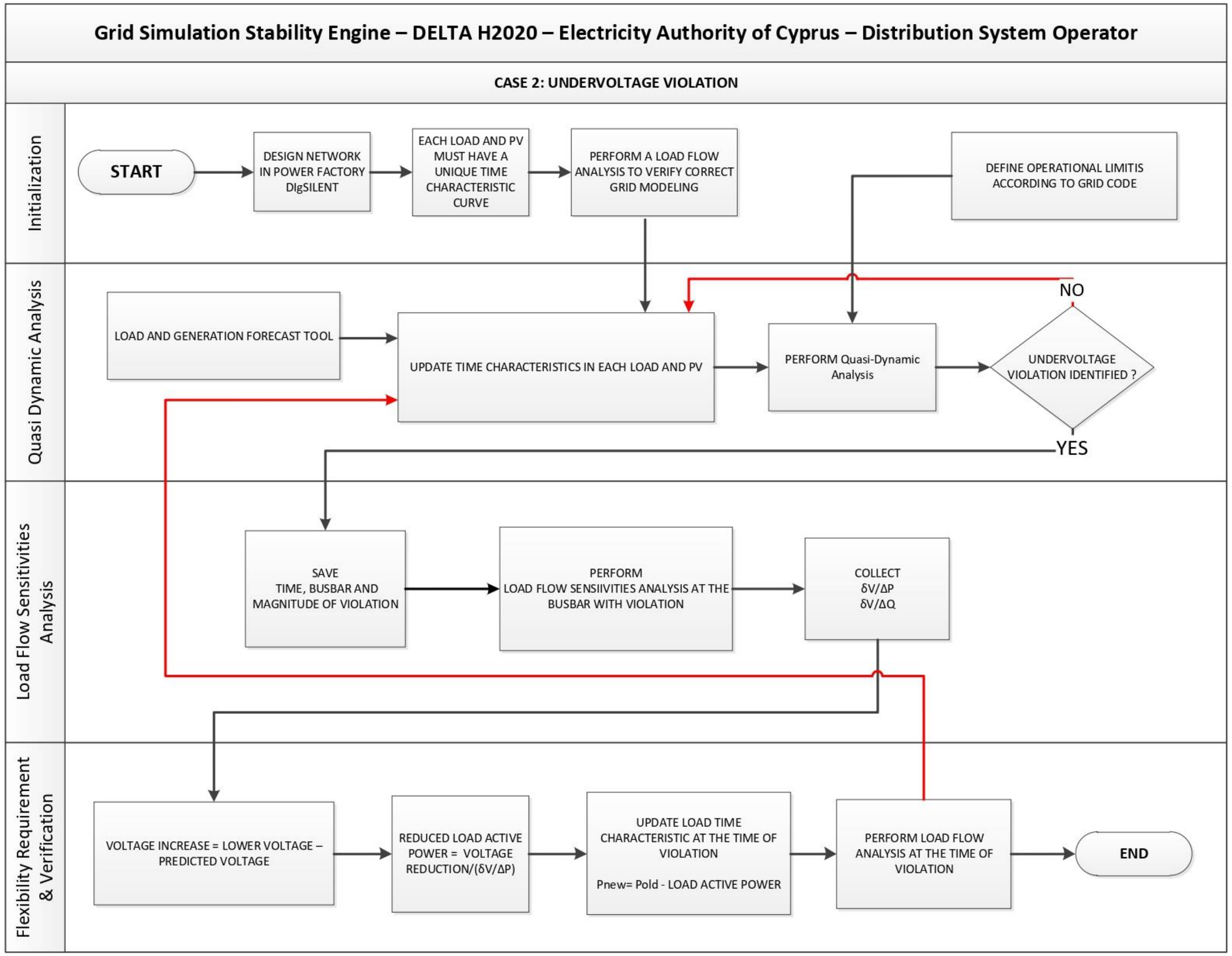

Figure 4: GSSE: Methodology for Undervoltage identification and restoration.

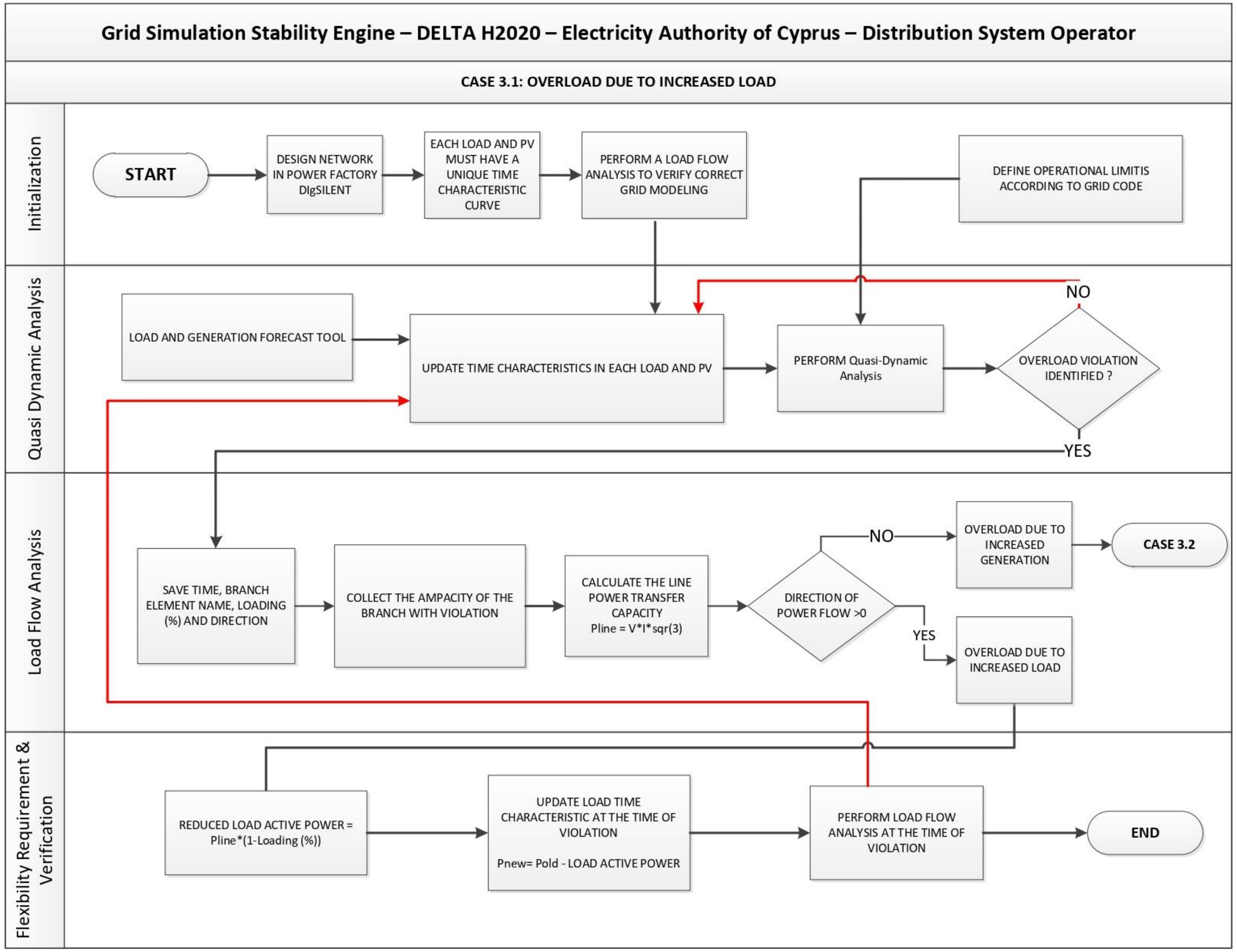

Figure 5: GSSE: Methodology for Overload (due to increased consumption) identification and restoration.

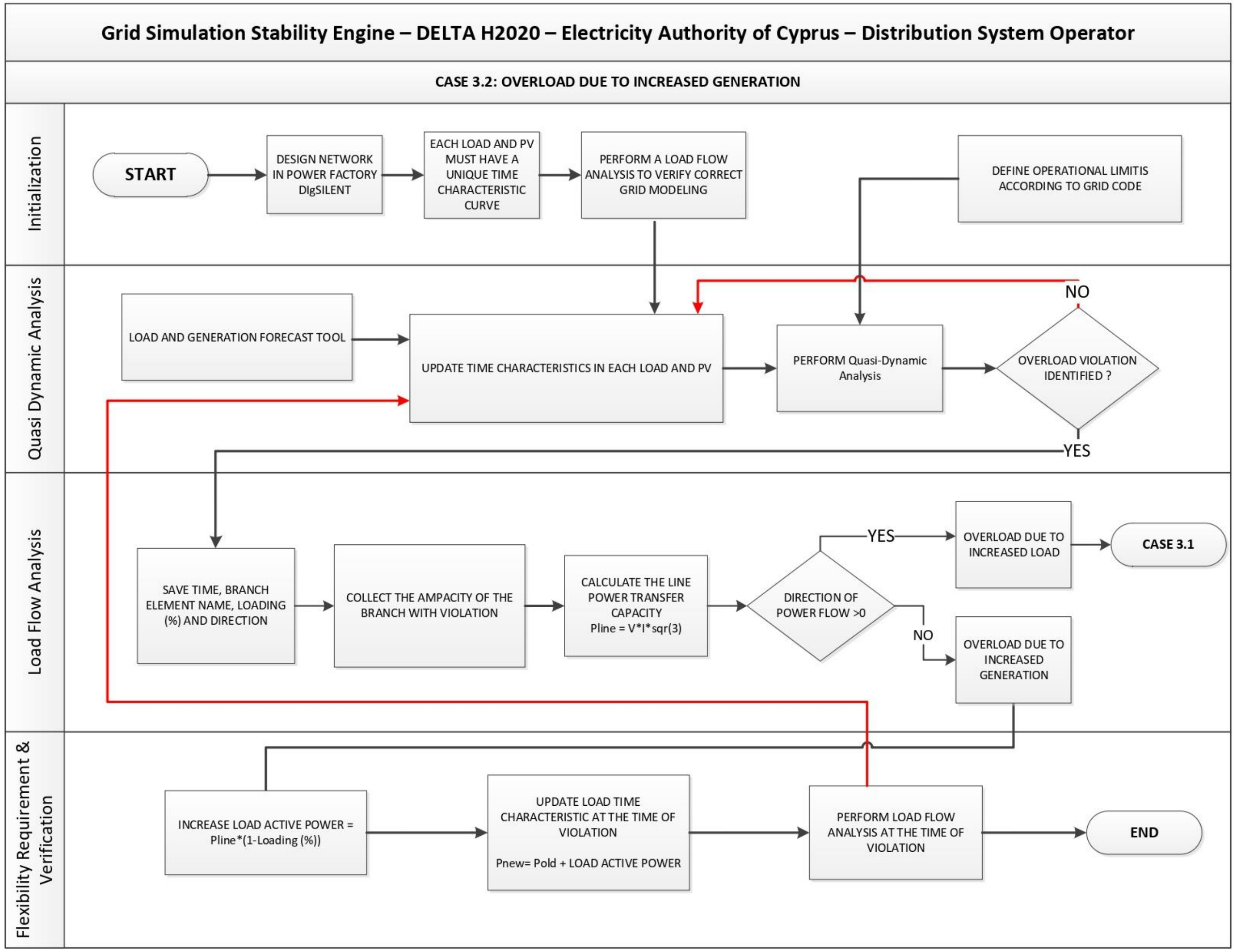

Figure 6: GSSE: Methodology for Overload (due to increased production) identification and restoration.

2.4.1.2 Connection with other Components and Interfaces

Component	Connection Type	API Protocol	Data Type	Comments
Node Flexibility Data Monitoring and Profiling	TCP/IP	RESTful Services	JSON-LD	Energy Profiles Communication will be handled through CIM
Asset Handling Optimization	TCP/IP	RESTful Services	JSON-LD	Voltage and Line Loading violations Violation Location (.wmf) Communication will be handled through CIM
DR & Flexibility Forecasting	TCP/IP	RESTful Services	JSON-LD	Estimated Required Flexibility Communication will be handled through CIM

2.4.1.3 Development Status

Development Status	√ Final □ Under development
Programming Language	Python, PowerFactory – DIgSILENT (Licensed Software)
Progress up to date	The final developments and experimental results have been documented in deliverable D4.1
Pending Development Actions	Refinement based on integration results if needed.

2.4.1.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	RESTfull Services through CIM
Progress up to date	An initial endpoint and data format have been set but further integration has been hindered by the coronavirus outbreak.
Pending Integration Actions	Full integration between the DELTA Aggregator and the DELTA Simulation State Estimation components.

2.4.1.5 Hardware/Software Problems Encountered

Problem Type	*Problem Description*	*Problem Cause*	*Countermeasure*
Fast, automatic and accurate identification of grid violations and restoring conditions.	For fast and accurate Grid state estimation as well as identification of possible grid violations a Quasi Dynamic Analysis on the investigated power network is required.	Power Network Simulators run manually without offering all functionalities necessary to automatically identify areas of grid constraints violation and how the steady state can be restored.	A Python script was developed that establishes real time and automatic control capabilities over a licensed software through an API. To this end, target areas in the investigated power network can be simulated, thus enabling identification of potential grid violations (type, location, time) and restoring conditions.

2.5 Innovative Customer Engagement Tools

2.5.1 DR Visualisation Kit

2.5.1.1 Functional Description

The DR Visualisation Kit's aim is to visualize real-time and historical energy information including consumption, generation, appropriate DR strategies for the Virtual Node, as well as financial and environmental data. It consists of two levels Aggregator level and Customer level.

On the Aggregator level DR Visualisation Kit provides a visualisation of the following data: Customers information, Historical Consumption, Historical Generation, Forecasted Flexibility, DR Signals, Bids, Rewards, Energy Price Profiling, DVN Clusters, Node Profiling and Aggregated Profiling.

Likewise on the Customer level DR Visualisation Kit is responsible for visualising: Rewards, DR Signals and FEID Energy Profile (as can been seen in Section 3.1).

2.5.1.2 Connection with other Components and Interfaces

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
Aggregator (in general)	TCP/IP	RESTful Services	JSON	Web API calls directly to the aggregator server, as the UI server is hosted on the same machine
FEID (CIM)	TCP/IP	RESTful Services	JSON-LD	Web API calls pass through CIM and the relevant P2P network

2.5.1.3 Development Status

Development Status	□ Final √ Under development
Programming Language	AngularJS and nodejs
Progress up to date	Most functionalities have already been developed and tested.
Pending Development Actions	Periodically retrieve up to date data. Display notifications for upcoming DR Signals and their status and outcome.

2.5.1.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	Web-based interface
Progress up to date	Current version has been already implemented in order to be evaluated
Pending Integration Actions	Integrate future component versions

2.5.1.5 Hardware/Software Problems Encountered

None.

2.5.2 Award–enabled Energy Behavioural Platform

2.5.2.1 Functional Description

Award-enabled Energy Behavioural Platform aims in promoting healthy competition amongst customers and mainly reaches Demand Response objectives in a playful manner based on game mechanics. Based on customer actions and rules defined in the game engine customer receives rewards that are saved for later use.

2.5.2.2 Connection with other Components and Interfaces

Component	Connection Type	API Protocol	Data Type	Comments
Aggregator (in general)	TCP/IP	RESTful Services	JSON	Web API calls directly to the aggregator server, as the UI server is hosted on the same machine
FEID (CIM)	TCP/IP	CIM	JSON-LD	Following the DELTA Ontology each FEID will be informed about the gamification results (besides smart contracts)

2.5.2.3 Development Status

Development Status	□ Final √ Under development
Programming Language	Nodejs
Progress up to date	The overall gamification platform has been implemented, and three main games have been developed.
Pending Development Actions	Further gamified services are envisioned and the existing games require extensive testing.

2.5.2.4 Integration Status

Integration Status	√ Final □ Under development
Format for Integration	Nodejs REST API
Progress up to date	The most recent gamification platform is currently being integrated to the Aggregator backend.
Pending Integration Actions	Extensive testing of basic functionalities as well as game creation and management are required.

2.5.2.5 Hardware/Software Problems Encountered

Problem Type	Problem Description	Problem Cause	Countermeasure
Initial framework aborted	The Initial gamification framework selected was based on the Liferay framework which introduced multiple problems to the overall functionality	The liferay implementation in general, mainly due to dependencies restrictions.	The liferay framework has been aborted and the gamification engine was transferred to a nodejs based implementation.

2.5.3 Social Interaction and Cooperation Platform

2.5.3.1 Functional Description

The Social Interaction and Cooperation Platform 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.

2.5.3.2 Connection with other Components and Interfaces

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
DR Visualisation Kit	TCP/IP	RESTful Services	JSON	Web API calls
Award-enable Energy behavioural Platform	TCP/IP	RESTful services	JSON	Web API calls
Aggregator Local Repository	TCP/IP	RESTful services	PostgreSQL queries	Retrieve and store

2.5.3.3 Development Status

Development Status	√ Final □ Under development
Programming Language	Angular
Progress up to date	Chat, Forum, and other basic functionalities available.
Pending Development Actions	Investigate ways to enhance communication and ease cooperation and implement them.

2.5.3.4 Integration Status

Integration Status	√ Final □ Under development
Format for Integration	Web-based interface
Progress up to date	Current version has been already implemented in order to be evaluated
Pending Integration Actions	Integrate future versions.

2.5.3.5 Hardware/Software Problems Encountered

None.

2.6 Common Information Modelling

The Common Information Model is the DELTA component that implements the Semantic Interoperability approach endowed for the DELTA project. It enables other components to exchange data transparently, as it has been reported in previous deliverables (D1.2 and 1.3). However, since different components have different data requirements and data exchange necessities, in DELTA we distinguish four different CIMs: FEID CIM, Virtual DELTA Node CIM, DELTA Aggregator CIM, and Security Services CIM.

In the following sub-sections details about each of those CIMs, and how they integrate their components to the rest of the platform, will be provided.

2.6.1 FEID CIM

2.6.1.1 Functional Description

The FEID CIM is in charge of allowing the FEID to exchange data with other components. More specifically, it enables the sub-services and interfaces of the FEID to receive data, or send data, to a set of allowed components from the platform. The interfaces of the FEID that are connected to the rest of the platform through the CIM are reported in D1.6. However, in general, the FEIDs only exchange data with the DVNs and Aggregators through their CIMs. Also, all the data exchanged is modelled with the DELTA Ontology.

The FEID is one of the extension points of the platform. The FEID CIM does not only connect FEIDs to the DELTA platform, but potentially, is also able to integrate non-DELTA compliant components that could work as a FEID. For those cases, the CIM is able to translate from heterogeneous formats and models used by those components into the DELTA format (JSON-LD) and model (DELTA ontology).

2.6.1.2 Connection with other Components and Interfaces

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
FEID	TCP/IP	RESTful Services	JSON-LD	Connected to all local FEID endpoints for data exchange with other layers.
DVN CIM	TCP/IP	XMPP	XML	-
Aggregator CIM	TCP/IP	XMPP	XML	-

2.6.1.3 Development Status

Development Status	√ Final □ Under development
Programming Language	Java
Progress up to date	The final developments and experimental results have been documented in deliverables D1.7, D3.1.
Pending Development Actions	None.

2.6.1.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	Java Library
Progress up to date	Deployment of the CIM the local infrastructure of the FEIDs.
Pending Integration Actions	Correctly setting up the CIM.

2.6.1.5 Hardware/Software Problems Encountered

None.

2.6.2 DELTA Virtual Node CIM

2.6.2.1 Functional Description

The Virtual DELTA Node CIM is in charge of allowing the Delta Virtual Nodes (DVN) to exchange data with other components. More specifically, it enables the sub-services and interfaces of the DVN to receive data, or send data, to a set of allowed components from the platform. The interfaces from a DVN connected to the rest of the platform are described in D1.7. All data been exchanged must follow the DELTA Semantic Interoperability requirements, which are: express all data in JSON-LD, and model the data with the DELTA Ontology.

Notice that the DVNs are the only components in DELTA that actually exchange data among them, besides exchanging data with the FEIDs and the Aggregator.

2.6.2.2 Connection with other Components and Interfaces

Component	***Connection Type***	***API Protocol***	***Data Type***	***Comments***
FEID CIM	TCP/IP	XMPP	XMPP	
DVN CIM	TCP/IP	XMPP	XML	
Aggregator CIM	TCP/IP	XMPP	XML	
DVN	TCP/IP	REST API	JSON-LD	

2.6.2.3 Development Status

Development Status	√ Final □ Under development
Programming Language	Java
Progress up to date	The final developments and experimental results have been documented in deliverables D1.7 and D3.1.
Pending Development Actions	None.

2.6.2.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	Java Library
Progress up to date	Deployment of the CIM the local infrastructure of the DVNs
Pending Integration Actions	Correctly setting up the CIM

2.6.2.5 Hardware/Software Problems Encountered

None.

2.6.3 Aggregator CIM

2.6.3.1 Functional Description

The Aggregator CIM is in charge of allowing the Aggregator to exchange data with other components. More specifically, it enables the sub-services and interfaces of the Aggregator to receive data, or send data, to a set of allowed components from the platform. The Aggregator CIM allows the Aggregator sub-components to exchange data with the rest of the platform as specified in D1.7. Similarly than for the other CIMs, all communication been exchanged through the CIM has to be expressed in JSON-LD and modelled with the DELTA ontology.

2.6.3.2 Connection with other Components and Interfaces

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
Aggregator	TCP/IP	REST API	JSON-LD	
DVNs CIM	TCP/IP	XMPP	XML	
FEID CIM	TCP/IP	XMPP	XML	

2.6.3.3 Development Status

Development Status	√ Final □ Under development
Programming Language	Java
Progress up to date	The final developments and experimental results have been documented in deliverable D1.7, D4.1.
Pending Development Actions	None.

2.6.3.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	Java Library
Progress up to date	Deployment of the CIM the local infrastructure of the Aggregator.
Pending Integration Actions	Correctly setting up the CIM

2.6.3.5 Hardware/Software Problems Encountered

None.

2.6.4 Security Services CIM

2.6.4.1 Functional Description

The Security Services CIM is in charge of allowing any component to query data from the DELTA Blockchain, as reported in D5.2. However, up to the moment there is no component that requires the data from the Blockchain in order to work. Nevertheless, the CIM enables consuming this data in case in the future a new component would require this feature.

The data stored in the Blockchain is not Resource Description Framework (RDF), and thus, the Security Services CIM performs a translation on the fly from non-RDF data to RDF in order to answer SPARQL queries.

2.6.4.2 Connection with other Components and Interfaces

Component	***Connection Type***	***API Protocol***	***Data Type***	***Comments***
Blockchain	TCP/IP	REST API	JSON	
DVN CIM	TCP/IP	XMPP	XML	
Aggregator CIM	TCP/IP	XMPP	XML	
FEID CIM	TCP/IP	XMPP	XML	

2.6.4.3 Development Status

Development Status	□ Final √ Under development
Programming Language	Java
Progress up to date	The final developments and experimental results have been documented in deliverable D1.7 and D5.2
Pending Development Actions	Develop the integration module to connect the CIM to the Blockchain.

2.6.4.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	Java Library
Progress up to date	Deployment of the CIM the local infrastructure of the Security Services
Pending Integration Actions	Correctly setting up the CIM

2.6.4.5 Hardware/Software Problems Encountered

None.

2.7 Cybersecurity Services

2.7.1 DELTA Blockchain

2.7.1.1 Functional Description

The DELTA blockchain network is critical for the operation of the DELTA platform. It will mainly build the infrastructure on which smart contracts are going to be deployed. It should also communicate with other DELTA components, in order to facilitate the interaction of such components with the DELTA blockchain smart contracts.

2.7.1.2 Connection with other Components and Interfaces

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
FEID	TCP/IP	RESTful Services	Certificate	Certificate is returned to FEID
FEIDs, peers, ordering services and Aggregator's components.	TCP/IP	RESTful Services	CRL	Updated CRLs are returned to nodes
FEIDs, peers, ordering services and Aggregator's components.	TCP/IP	RESTful Services	Certificate	New certificates are distributed to all nodes

2.7.1.3 Development Status

Development Status	√ Final □ Under development
Programming Language	Hyperledger Fabric / Docker
Progress up to date	The final developments and experimental results have been documented in deliverable D5.2
Pending Development Actions	None.

2.7.1.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	The blockchain network is going to be integrated with the rest of components, in the sense that every component can communicate and get the required response for the network.
Progress up to date	Initial tests done with mock-up components
Pending Integration Actions	More elaborate tests will be done with actual implementation of the DELTA components

2.7.1.5 Hardware/Software Problems Encountered

None.

2.7.2 Smart Contracts

2.7.2.1 Functional Description

One of DELTA's main innovations is the design and implementation of an automated, verifiable, secure and decentralized framework that governs the transactions among the actors in its ecosystem. To achieve this, DELTA harnesses the power and expressiveness of blockchain-based smart contracts. This modern, ingenious construct allows us to define, in executable code, the data, rules and processes involved in business models and use cases. Consequently, DELTA brings forth a new era of contractual agreements between small/medium customers and aggregators.

2.7.2.2 Connection with other Components and Interfaces

Component	***Connection Type***	***API Protocol***	***Data Type***	***Comments***
Smart Contract Gateway	TCP/IP	gRPC	JSON	See D5.2 for more details.

2.7.2.3 Development Status

Development Status	√ Final □ Under development
Programming Language	Python
Progress up to date	A first complete smart contract that covers most of the DR envisioned transactions has been developed, deployed and tested.
Pending Development Actions	Further tests are required. More OpenADR scenarios need to be evaluated and potentially lead to new contracts

2.7.2.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	Python Scripts
Progress up to date	The current smart contact has been fully integrated and can dynamically be executed to support the envisioned transactions
Pending Integration Actions	Further testing is required for both scale up scenarios and multiple DR events to evaluate robustness.

2.7.2.5 Hardware/Software Problems Encountered

None.

2.7.3 Smart Contract Gateway

2.7.3.1 Functional Description

This component consists the main decentralised application (dapp)[1] developed in DELTA, and enables easy interaction of other DELTA components with the smart contracts.

2.7.3.2 Connection with other Components and Interfaces

Component	*Connection Type*	*API Protocol*	*Data Type*	*Comments*
Aggregator	TCP/IP	RESTful Services	JSON-LD	
DVN	TCP/IP	RESTful Services	JSON-LD	
FEID	TCP/IP	RESTful Services	JSON-LD	
DELTA Smart Contracts	TCP/IP	RESTful Services (gRPC)	JSON	See D5.2 for more details.

[1] http://blockchainhub.net/decentralized-applications-dapps/

2.7.3.3 Development Status

Development Status	√ Final □ Under development
Programming Language	GO
Progress up to date	The smart contract gateway has been developed, deployed and tested.
Pending Development Actions	None.

2.7.3.4 Integration Status

Integration Status	□ Final √ Under development
Format for Integration	Python Scripts
Progress up to date	The smart contract gateway has been integrated towards providing easy access to any DELTA smart contract to other layers.
Pending Integration Actions	Further testing upon integration is required.

2.7.3.5 Hardware/Software Problems Encountered

None.

2.7.4 Threat Mitigation

2.7.4.1 Functional Description

Threat detection is 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 Denial of Service (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.

2.7.4.2 Connection with other Components and Interfaces

Implementation of this component is still at the early stages, so the designed connection can be extracted from D1.6.

2.7.4.3 Development Status

Development Status	□ Final √ Under development
Programming Language	-
Progress up to date	Currently undergoing early stage implementation
Pending Development Actions	Complete implementation of the first functional version

2.7.4.4 Integration Status

Integration Status	◻ Final ◻ Under development
Format for Integration	-
Progress up to date	Not started
Pending Integration Actions	Complete integration of the first functional version

2.7.4.5 Hardware/Software Problems Encountered

None.

3. DELTA Components User Interfaces

3.1 DELTA Customer

3.1.1 Fog-enabled Intelligent Device

The experienced technician that will install FEID in a customer infrastructure should follow a specific installation procedure. The first time that FEID is plugged in power, creates a Wi-Fi access point. The technician though a portable device (e.g. mobile phone, tablet) searches for the available networks and connects his device to FEID's access point, Figure 7.

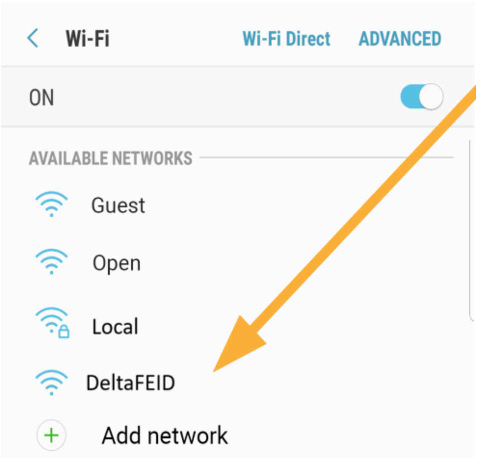

Figure 7: FEID Wi-Fi access point

Then the technician starts the installer application, Figure 8. In the screen appears the connected FEID that is not yet registered, Figure 9.

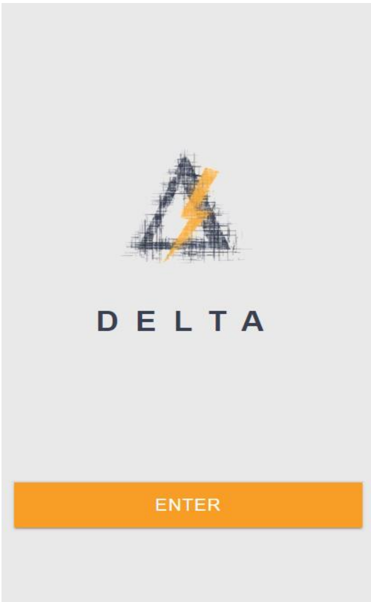

Figure 8: Installer UI starting screen

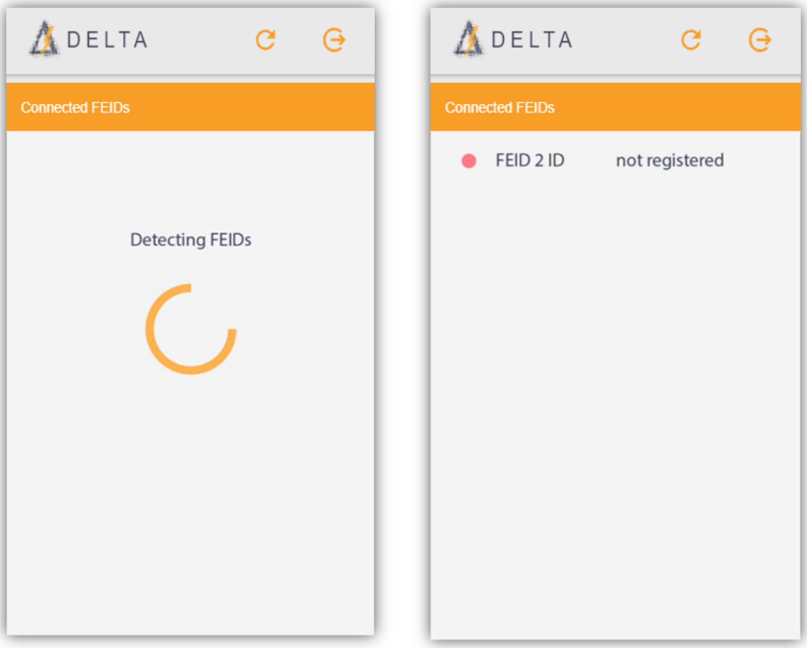

Figure 9: FEID detection and information about registered

In the next screen the connection with the FEID can be tested, Figure 10.

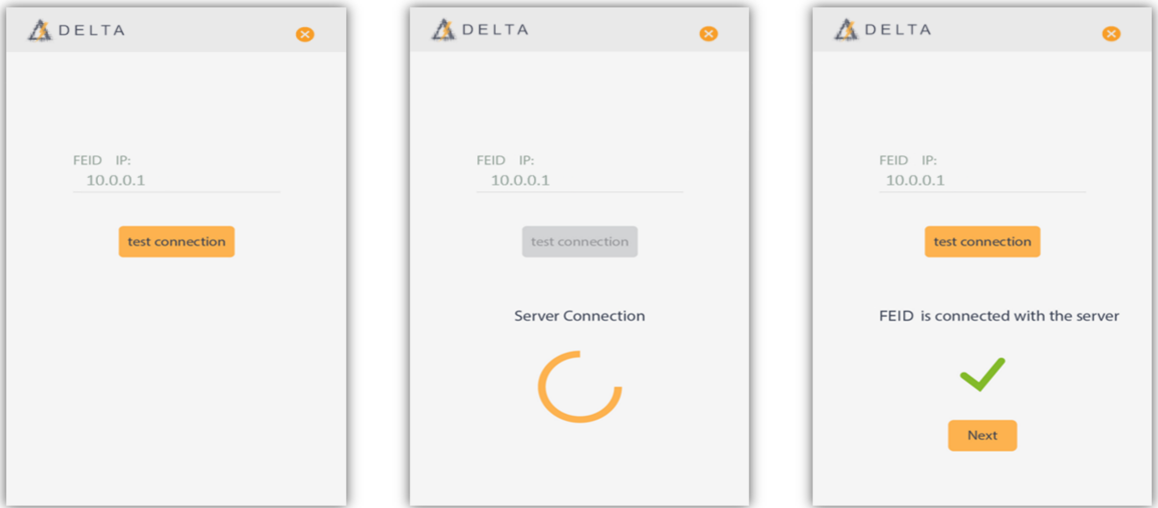

Figure 10: FEID connection test

After the connection testing, the list with all connected devices appears. Initially the list is empty and the technician should add all the devices that will be monitored and controlled by FEID, Figure 11.

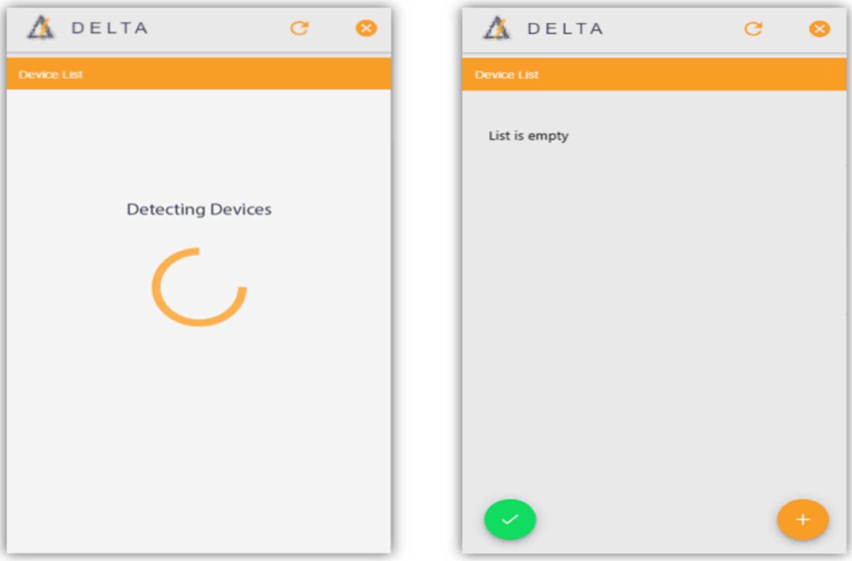

Figure 11: FEID devices list

When adding a device, the technician can choose between adding a connection with the BMS or directly connection with a smart device, Figure 12.

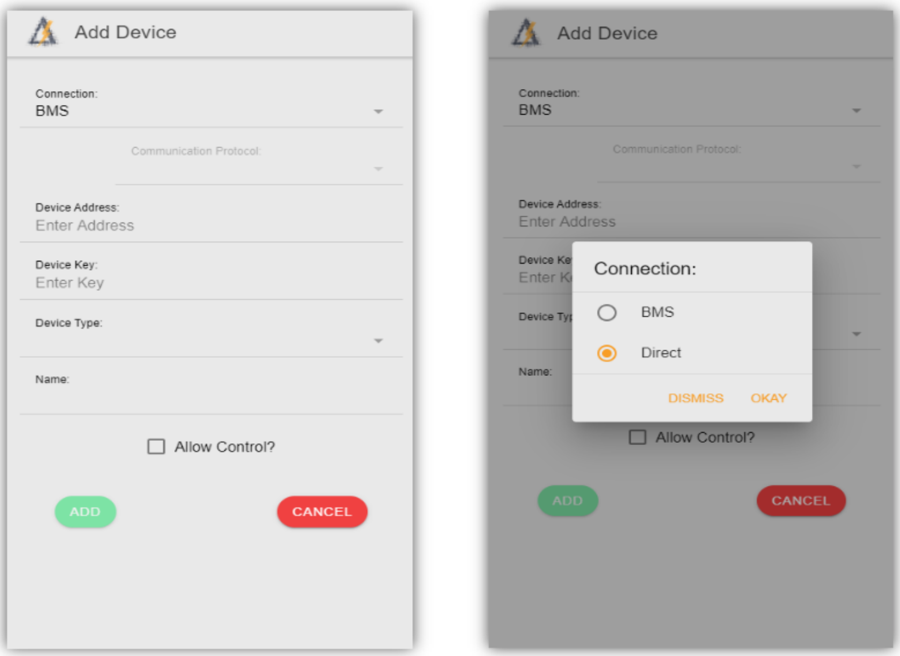

Figure 12: Add Building Management System

If direct connection is selected, then the communication protocol with this device should be chosen. In case relay protocol is selected, then the technician has to define in which COM port of the FEID this device will be connected, Figure 13.

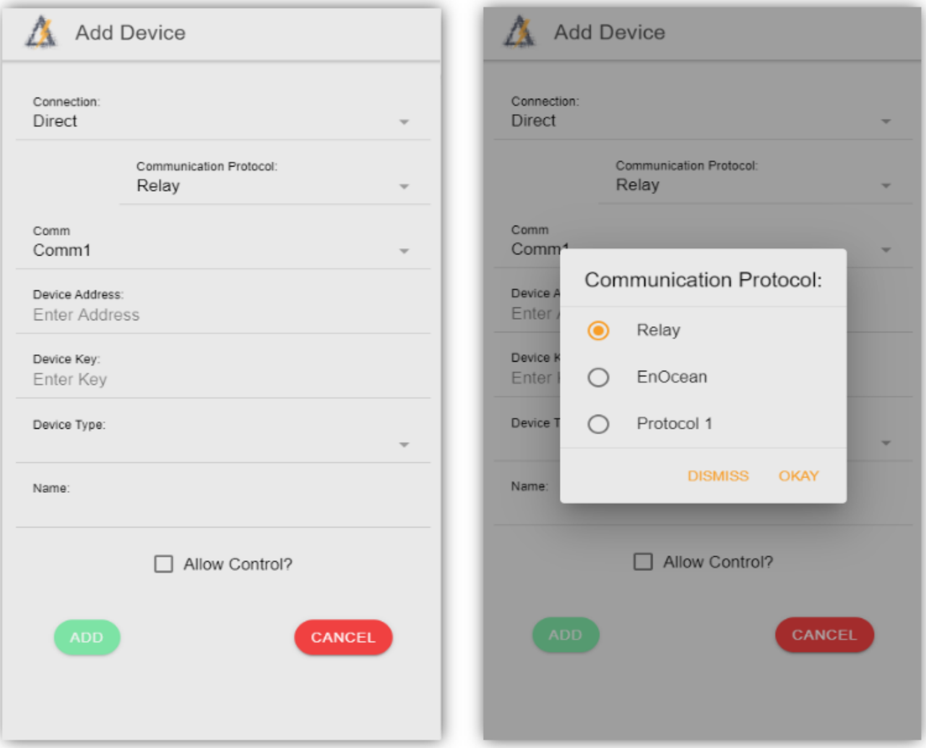

Figure 13: Add direct control device

Finally the device type as well as its consumption/storage/generation capacity (as provided by the customer and verified by the installer) has to be selected and whether this device will be controllable or not, Figure 14.

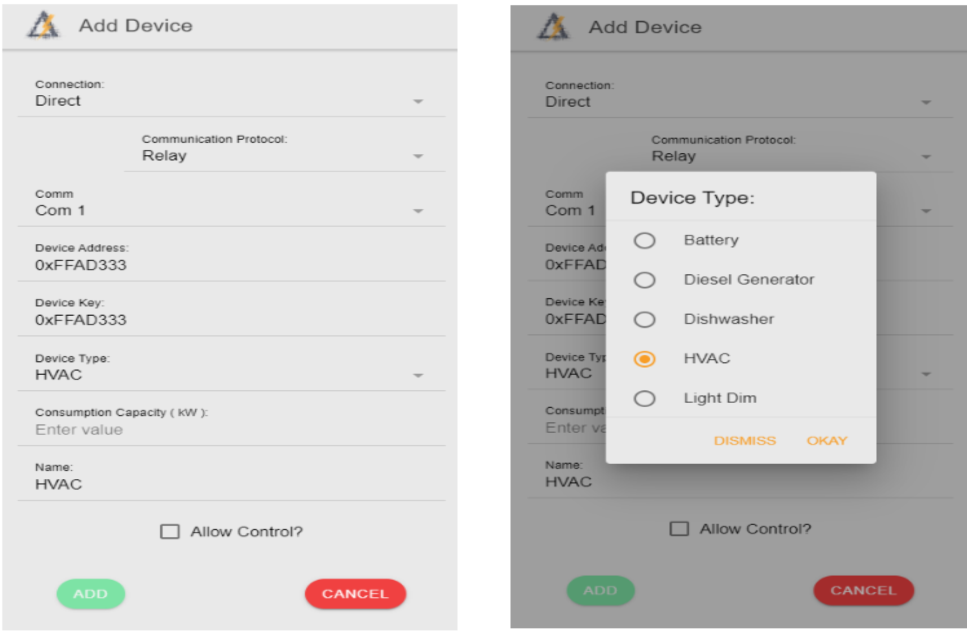

Figure 14: Device Type

The device is added to the general list and the same procedure repeats until all devices have been registered in FEID, Figure 15.

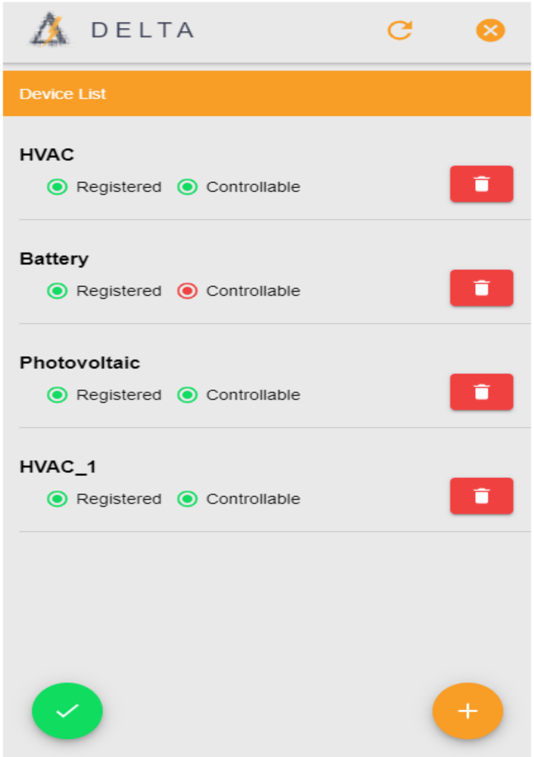

Figure 15: Completed device list

The technician along with customer defines the time-slots during which the customer is available for receiving Demand Response schemes and also defines their comfort preferences (due to complexity, only once per building and not per zone). This information can also be changed later by customer from their user interface, Figure 16.

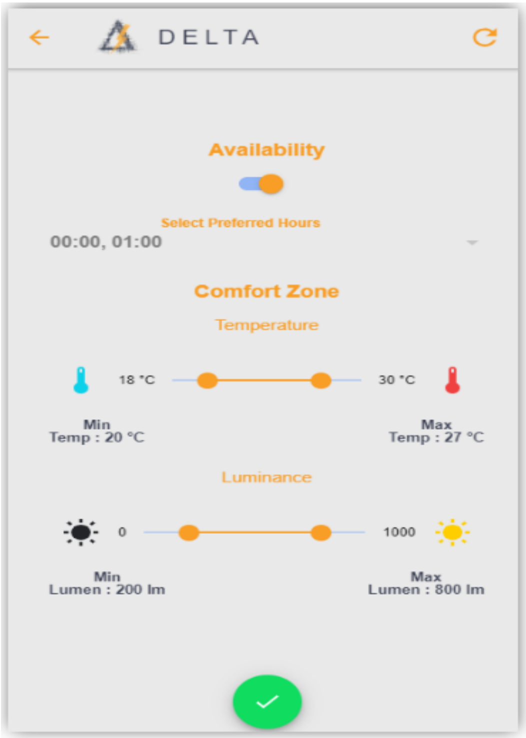

Figure 16: Customer preferences

Finally, the technician sets the information related to the customer as they have been provided by the aggregator, Figure 17.

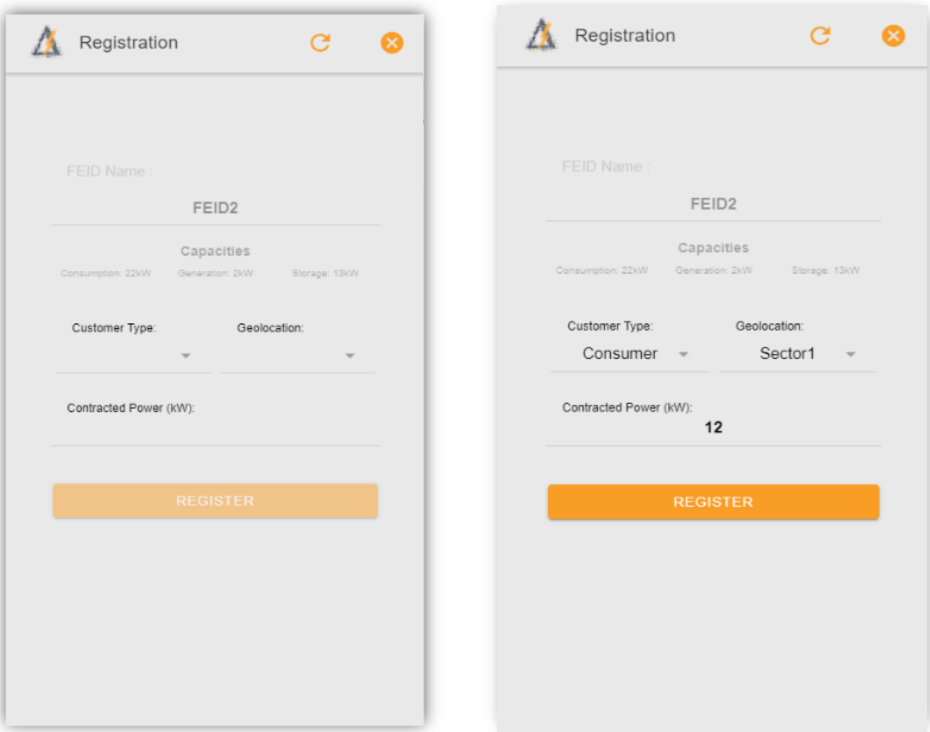

Figure 17: Customer general information

FEID also supports a user interface for the customer. The login page appears in Figure 18.

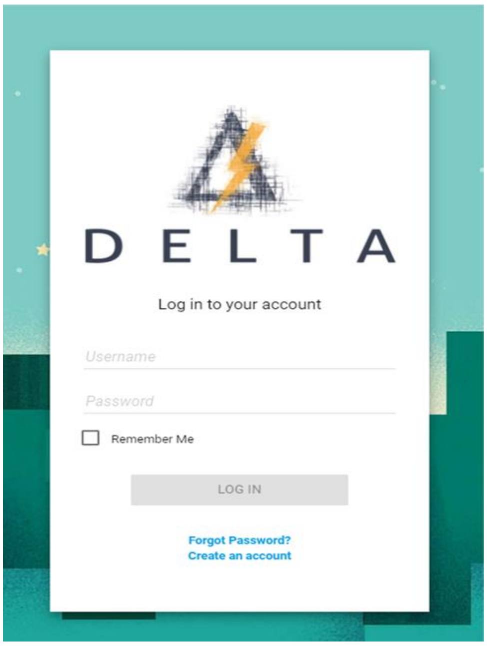

Figure 18: Customer UI, login page

The main dashboard of the customer UI appears in Figure 19. Here the user has a general overview of the current status of their infrastructure.

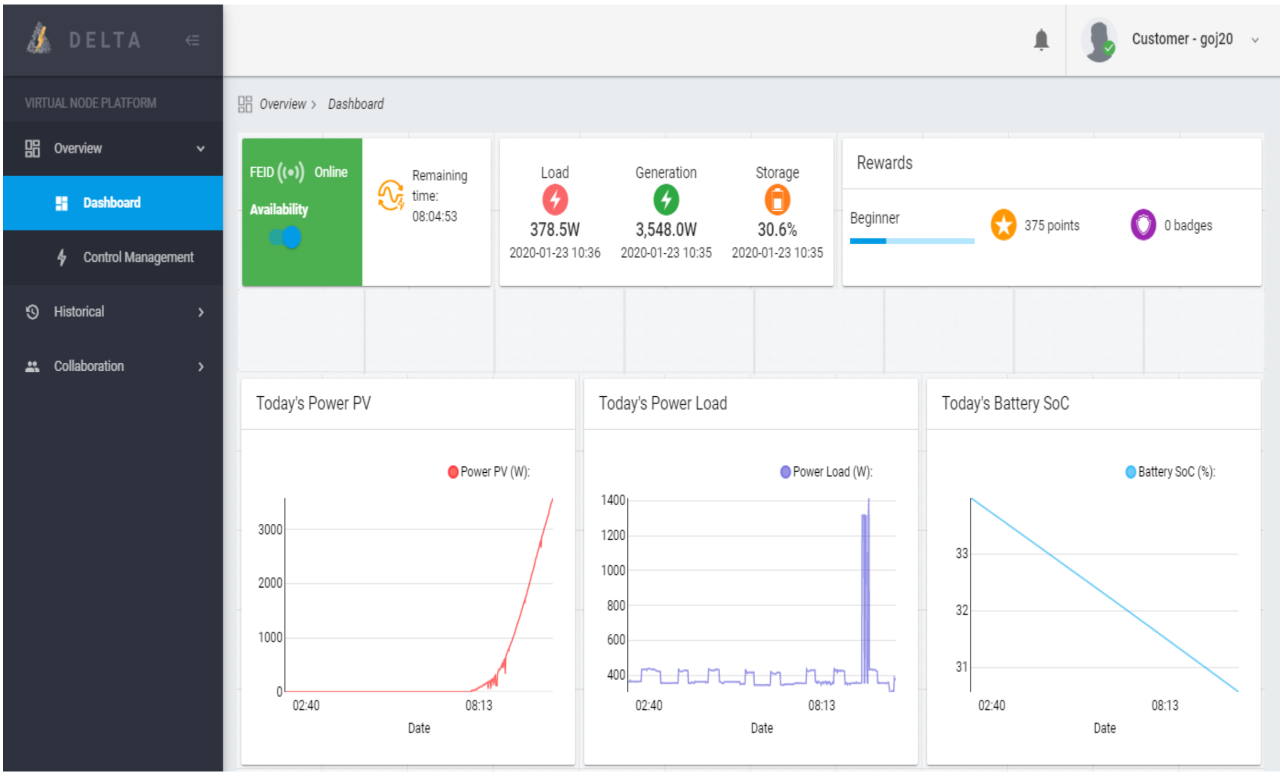

Figure 19: Customer UI, main dashboard

In this dashboard user can change their preferences regarding the timeslots that they are available for receiving demand response messages, the device that are referred as controllable and preferences related to their comfort, Figure 20.

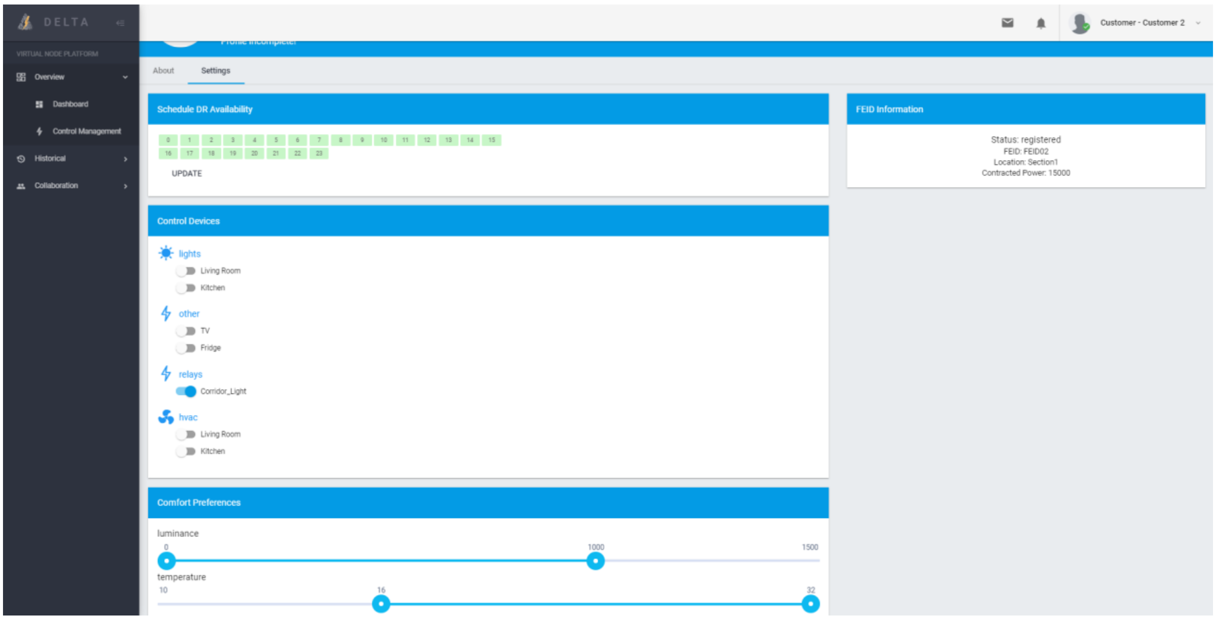

Figure 20: Customer UI, user preferences

From this dashboard the user can monitor and control each device separately, Figure 21.

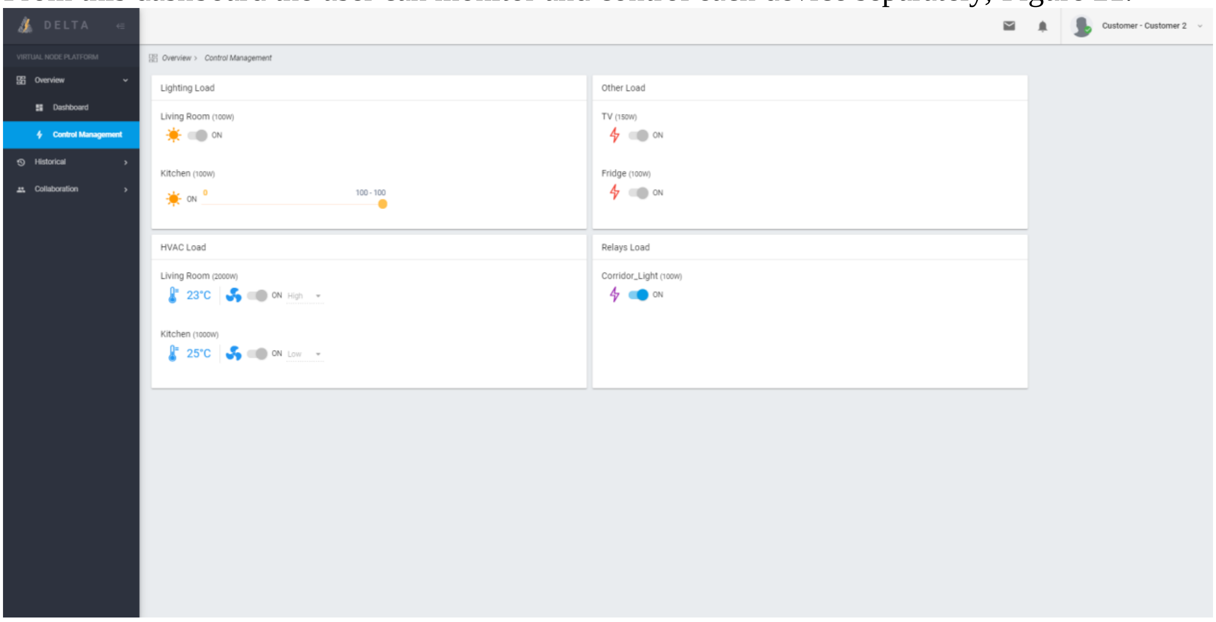

Figure 21: Customer UI, device management

This dashboard contains a list of all the historical demand response messages that have been arrived to this FEID, as shown in Figure 22. More information about the DR Visualisation Kit as well as other innovative customer engagement tools are presented in Section 3.4.

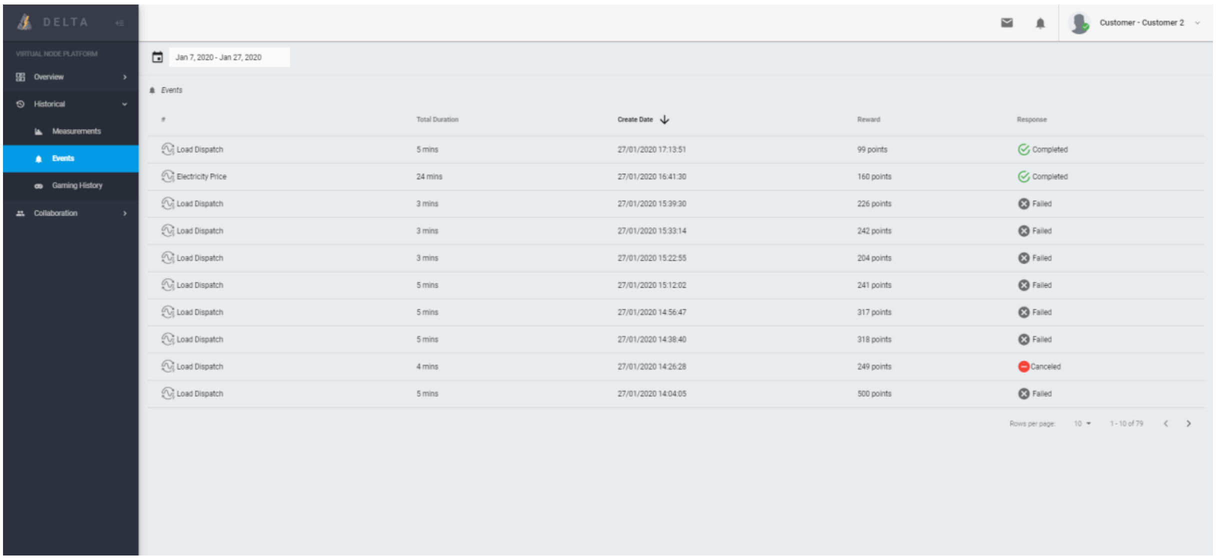

Figure 22: Customer UI, historical DR messages

3.2 DELTA Virtual Node

3.2.1 Consumer/Prosumer Flexibility Data Monitoring and Profiling

Beyond the information provided to the Aggregator per node (see Section 3.4), no dedicated UI has been developed to demonstrate the information provided by this tool within the DVNP.

3.2.2 Generation/Consumption Optimal Dispatch

To facilitate knowledge transfer as well as better understanding of how the DELTA tools work, a web interface has been delivered to execute various scenarios in regards to the current version of the Optimal Dispatch sub-components. Two different indicative examples are presented in the following figures.

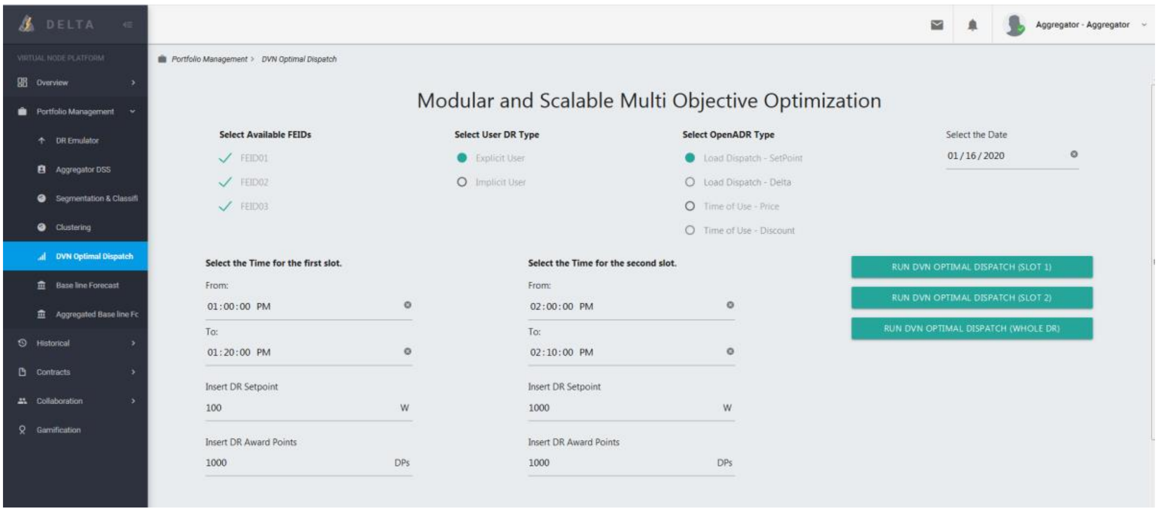

Figure 23: DVN Optimal Dispatch (OptiDVN) UI – Aggregator's view

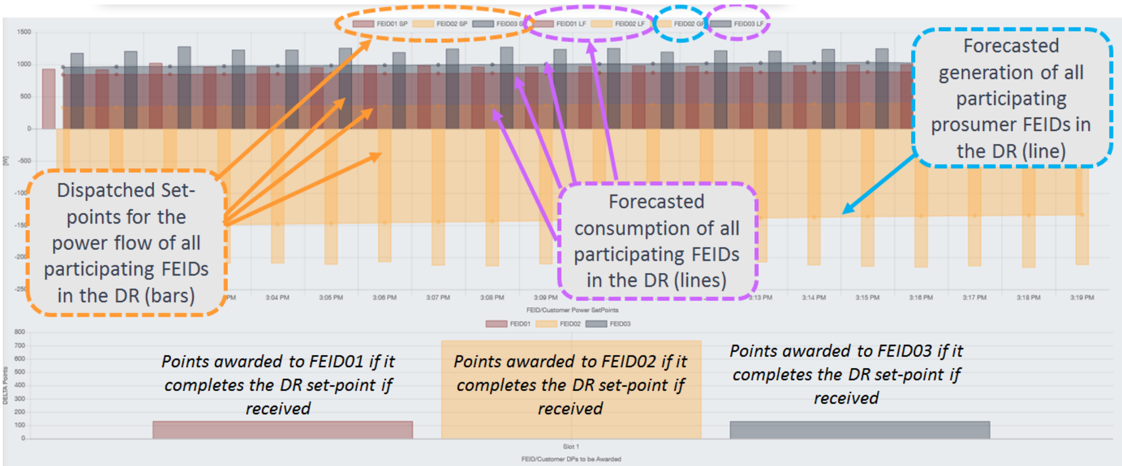

Figure 24: DVN Optimal Dispatch (OptiDVN) UI – Load Dispatch Scenario – Explicit DR Users

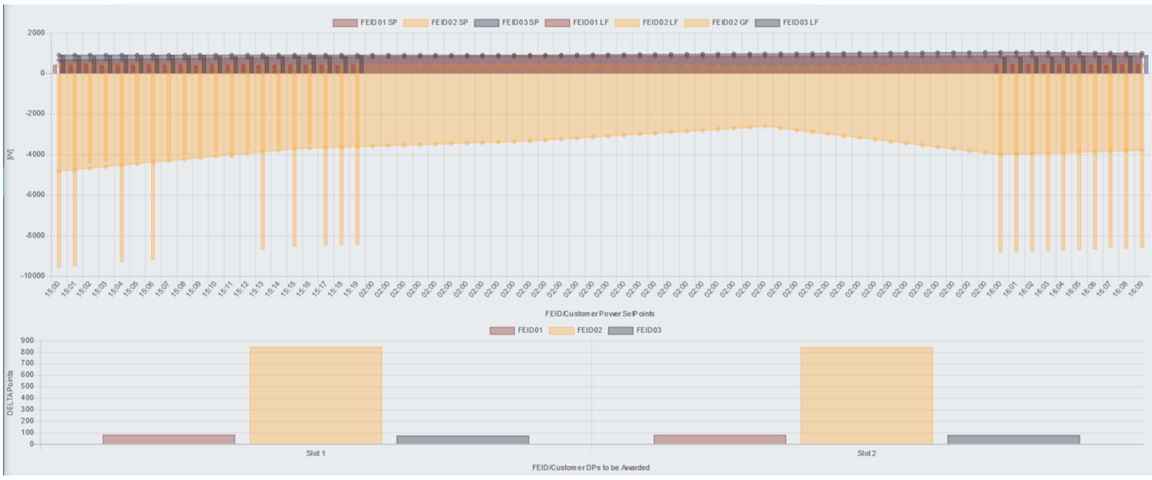

Figure 25: DVN Optimal Dispatch (OptiDVN) UI – Time of Use Scenario – Explicit DR Users

3.2.3 Load Forecasting

Similarly with the above components, a UI has been created towards evaluating the performance of the Load forecasting from the Aggregator's perspective and for both the DVN and the Aggregator layers. Indicative examples are presented in the Figures

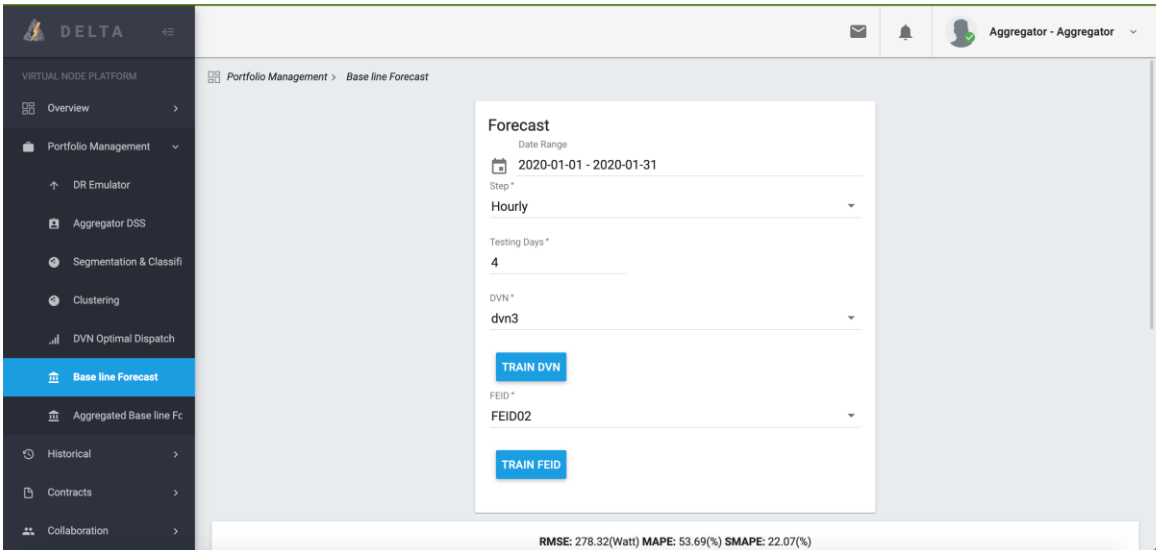

Figure 26: Baseline Load Forecasting UI - Parameter Definition

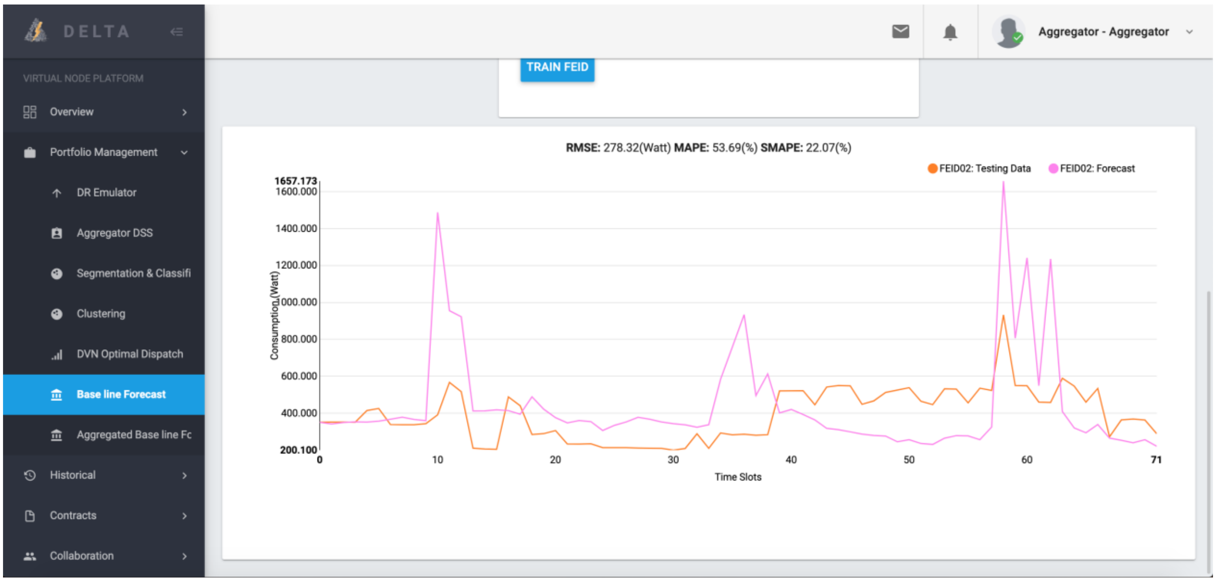

Figure 27: Baseline Load Forecasting UI – Results

3.2.4 Inter/Intra Node Energy Matchmaking

There is no specific UI until the examined period. Based on the implementation needs, a dedicated UI may be developed for this component. The results however can be visualised by the DR visualisation Kit.

3.2.5 Consumer/Prosumer Energy/Social Clustering

Once again, a UI has been delivered to allow better understanding and evaluation of this component. Further development is required to fully present a user friendly environment for engaged end-users.

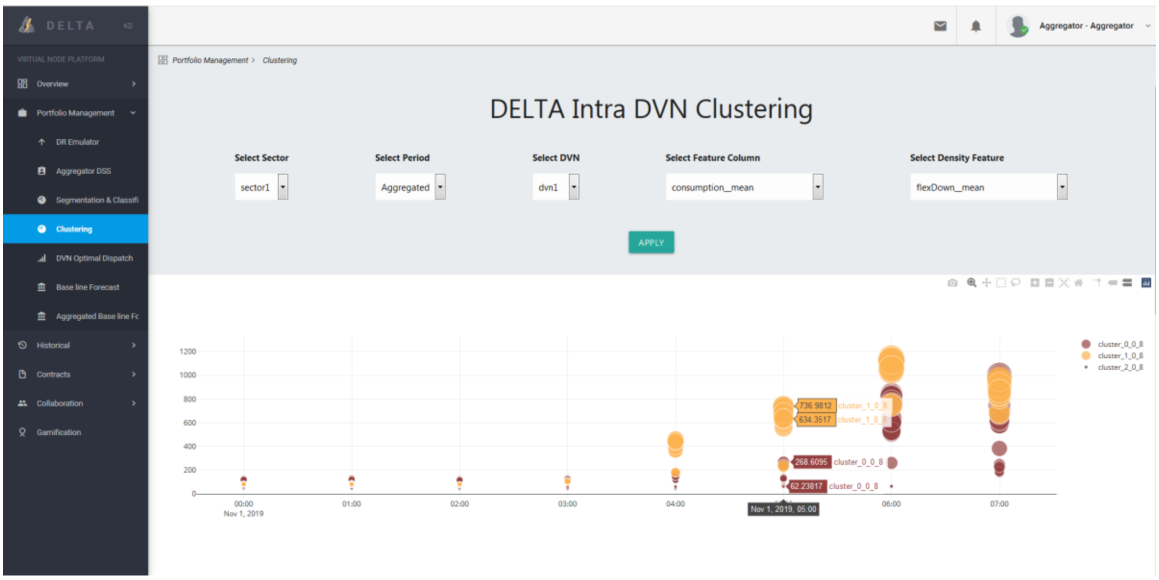

Figure 28: DELTA DVN Energy/Social Clustering UI

3.3 DELTA Aggregator

3.3.1 Energy Market Price Forecast

There is no specific UI until the examined period. There is a stand-alone UI which has been used during implementation, as seen in the figure that follows, but up to M24 this hasn't been integrated to the DVNP. It is within next actions to provide detailed representation of this component including the imbalance market actual results.

3.3.2 DR & Flexibility Forecasting

There is no specific UI until the examined period for this tool. Upon completion of the related sub-components a dedicated UI has already been designed. At this point, as already demonstrated in other views, the aggregation of the DVN flexibility results is provided to the Aggregator.

3.3.3 Node Flexibility Data Monitoring and Profiling

There is no specific UI. Monitoring is available through the overall DVNP UI.

3.3.4 Asset Handling Optimization

Once again, towards facilitating understanding of how the DELTA tools work, a web interface has been delivered to execute various scenarios in regards to the current version of the Asset Handling Optimization component.

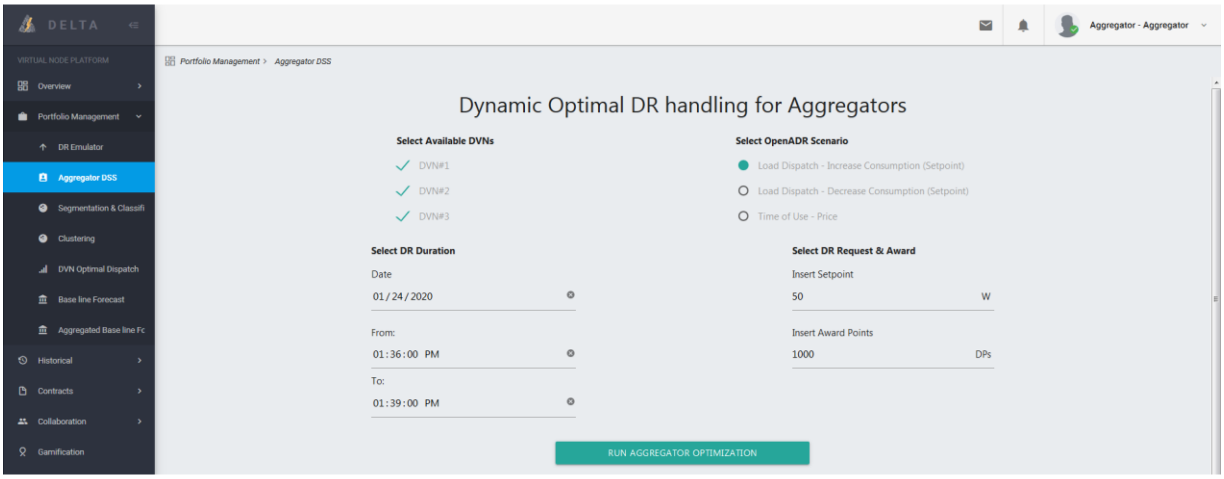

Figure 29: Aggregator Asset Handling Optimization UI – Parameterisation

3.3.5 Self-Portfolio Energy Balancing

There is no specific UI until the examined period for this tool. The SPEB results will be demonstrated through the DSS UI.

3.3.6 Grid Stability Simulation Engine

Currently the only visual representation of the GSSE results is through the UI of the DIgSILENT software as seen in the figures below. Upon completion of the integration process, the UI of the Aggregator will also include a dedicated section for the GSSE results, towards facilitating the information flow and understanding of the GSSE outcomes.

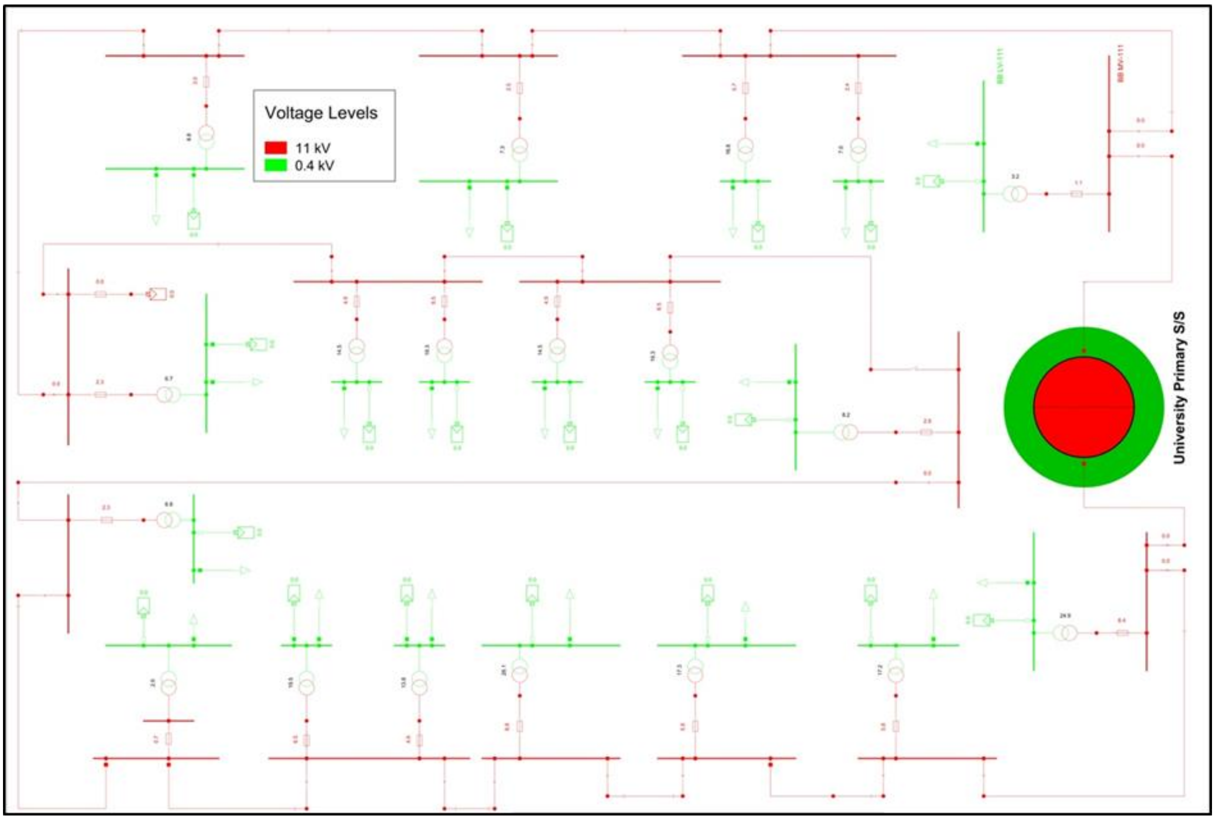

Figure 30: Detailed model of the UCY campus power network.

Figure 31: Power Flow Analysis at the time of violation.

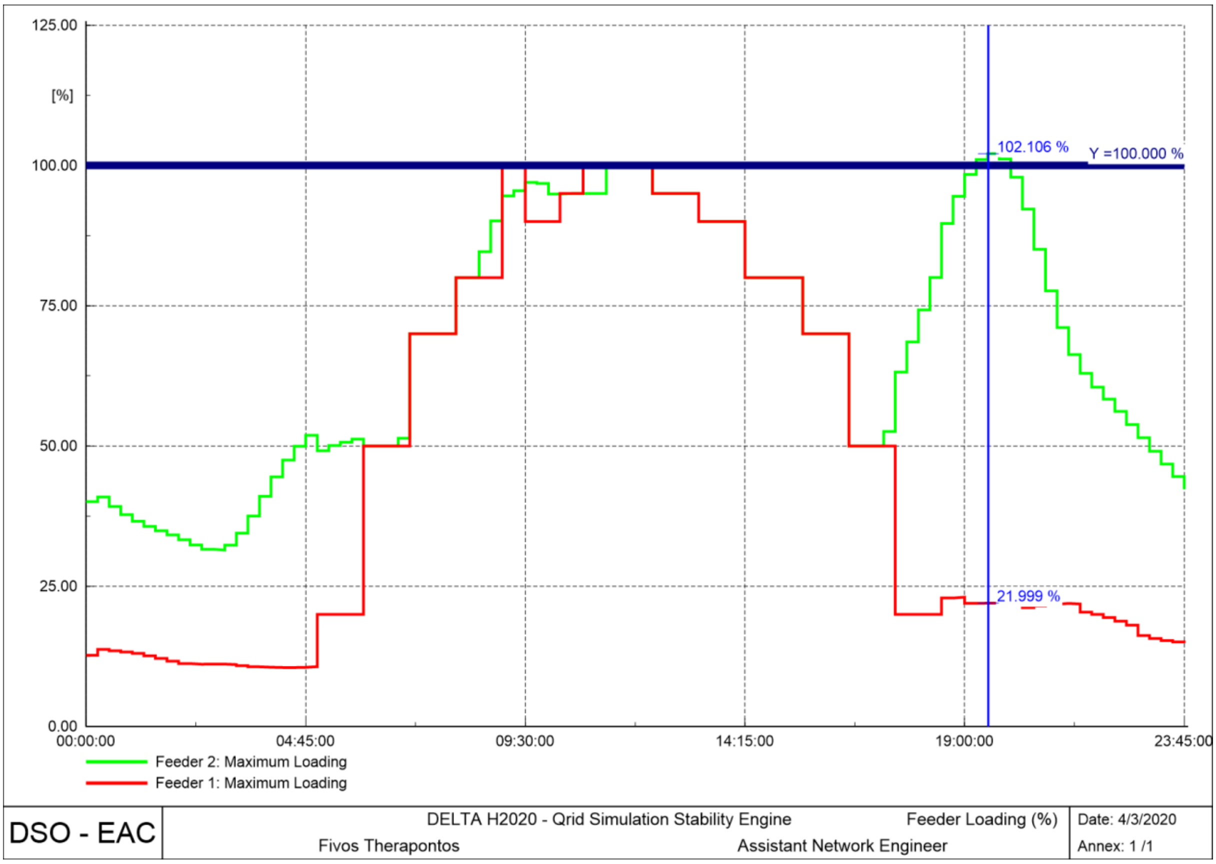

Figure 32: Quasi-Dynamic analysis – Feeder Active Power Flow.

3.3.7 Energy Portfolio Segmentation & Classification

Following integration steps of both the Segmentation and Clustering sub-components there are two different visual representations of the resulting information. Through the general DVNP UI, through which the Aggregator can navigate within it's portfolio down to the FEID level, always going through the various segments/clusters (DVNs and clusters of FEIDs), and through a dedicated radial representation that aims to facilitate identification of potential positive (i.e. available flexibility in an expected high price imbalance market slot) and negative (i.e. low reliability customers) situations. Both representations are presenting in the following figures.

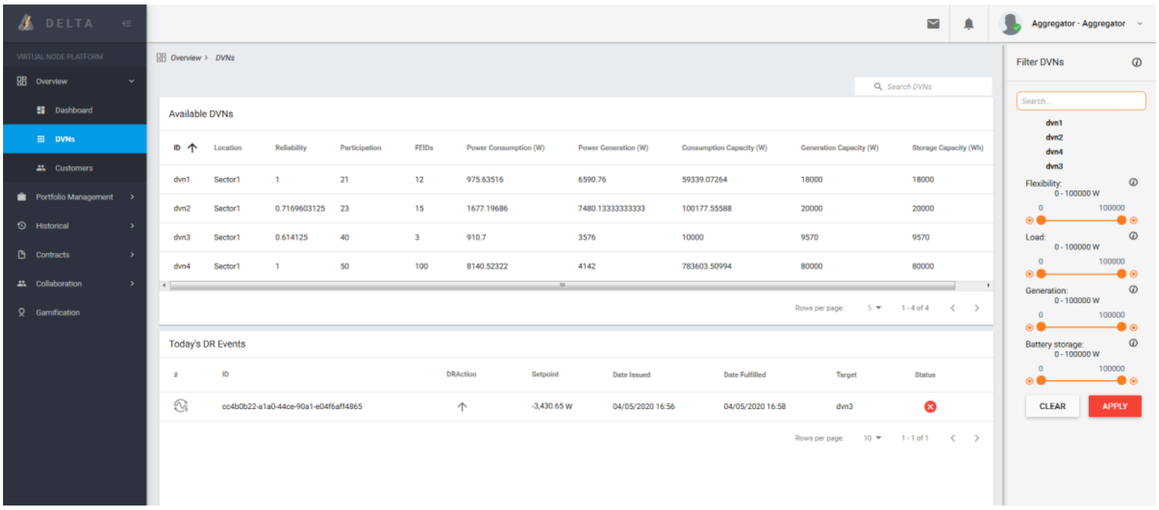

Figure 33: DVNP UI – Available DVNs and their characteristics

Figure 34: Radial Tree representing both the Segmentation and the Clustering steps.

3.4 Innovative Customer Engagement Tools

3.4.1 DR Visualisation Kit

The DR Visualisation Kit consists of two levels, Aggregator and Customer, and as previously explained in 3.4 Innovative Customer Engagement Tools each has is own view in the UI.

<u>Aggregator level:</u>

In Aggregator level the provided information is Customers information, Historical Consumption, Historical Generation, Forecasted Flexibility, DR Signals, Bids, Rewards, Energy price Profiling, DVN Clusters, Node Profiling and Aggregated Profiling. The following images demonstrate the above.

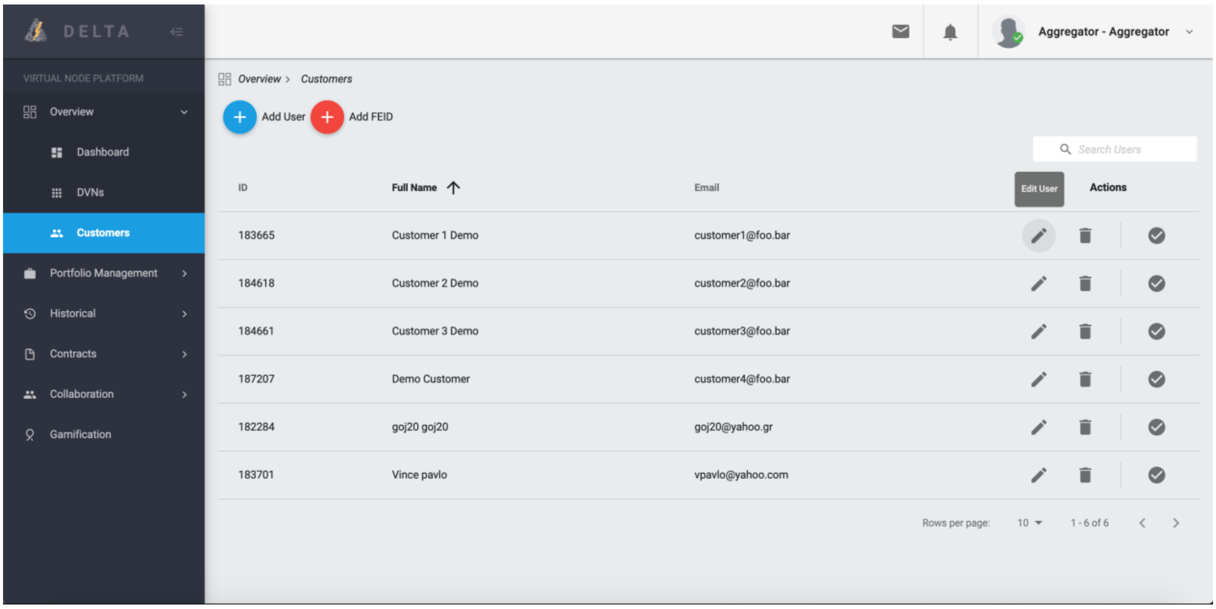

Figure 35: Customer General Information

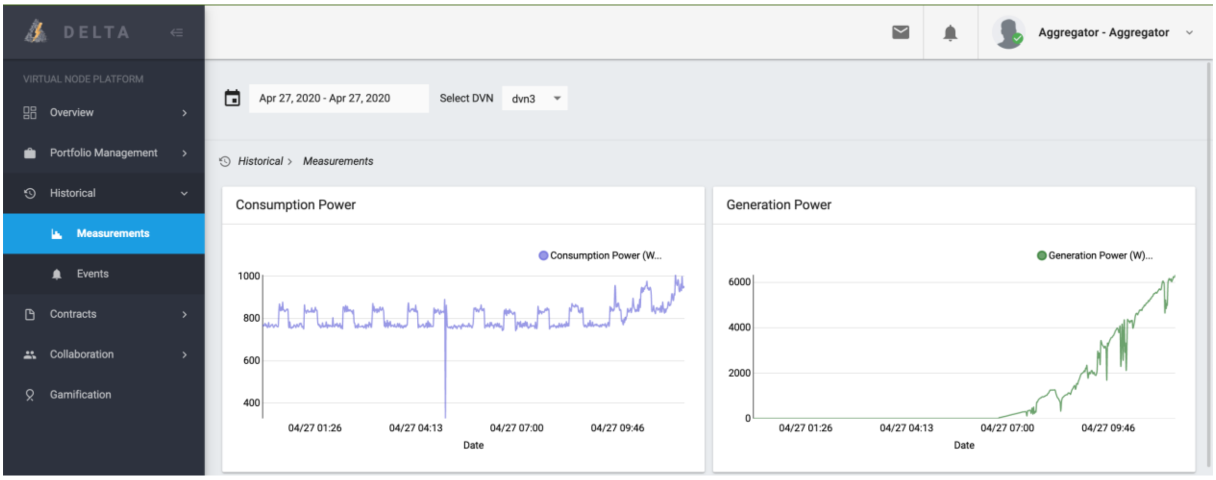

Figure 36: Historical Generation and Consumption

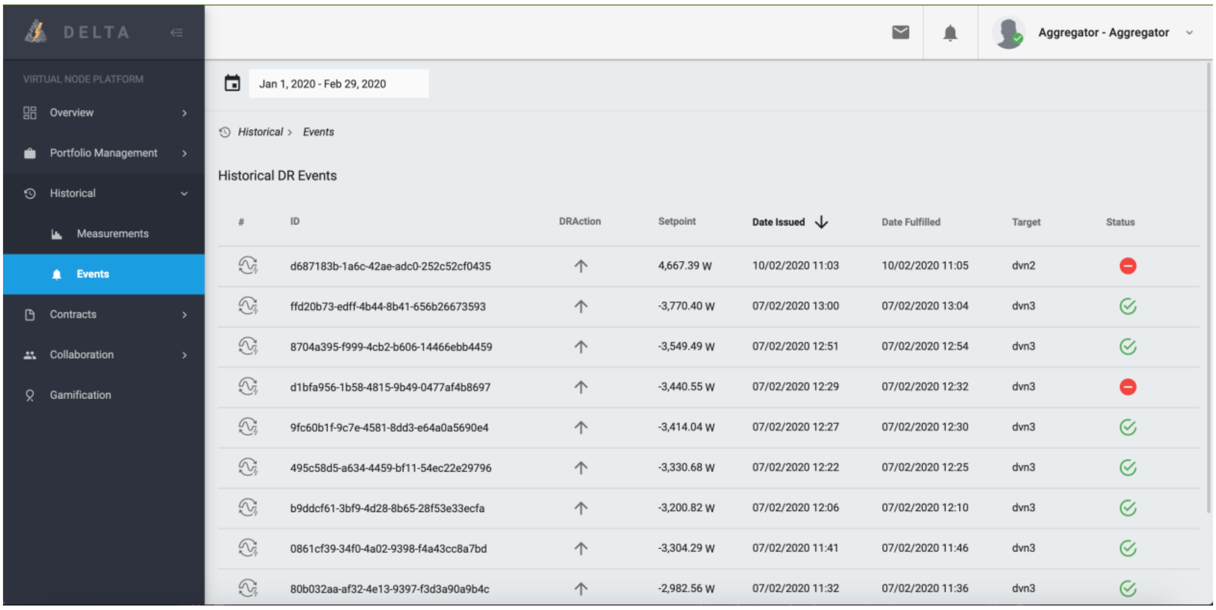

Figure 37: Incoming and Outgoing DR Events

<u>Customer level:</u>

In Customer level the provided information is Rewards, DR Signals and FEID Energy Profile. The images below demonstrate the above.

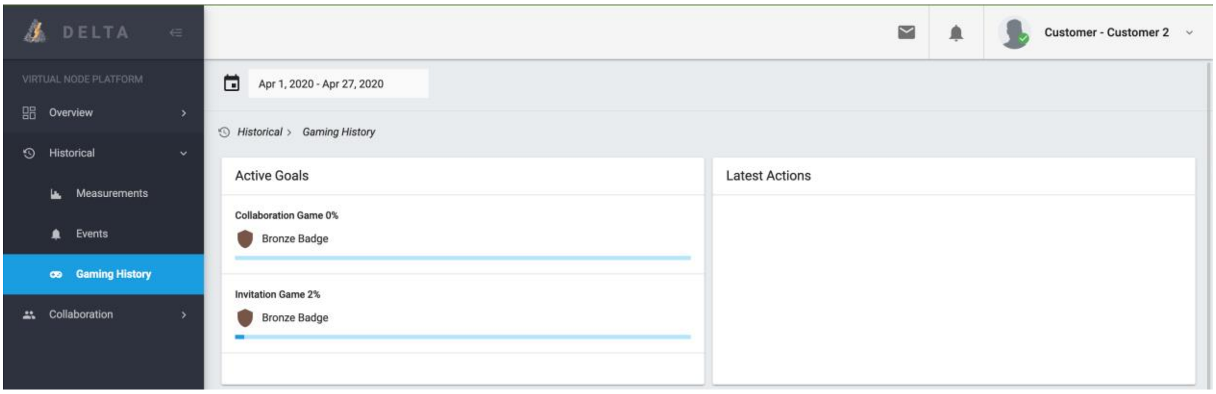

Figure 38: Customer Rewards and progress per game

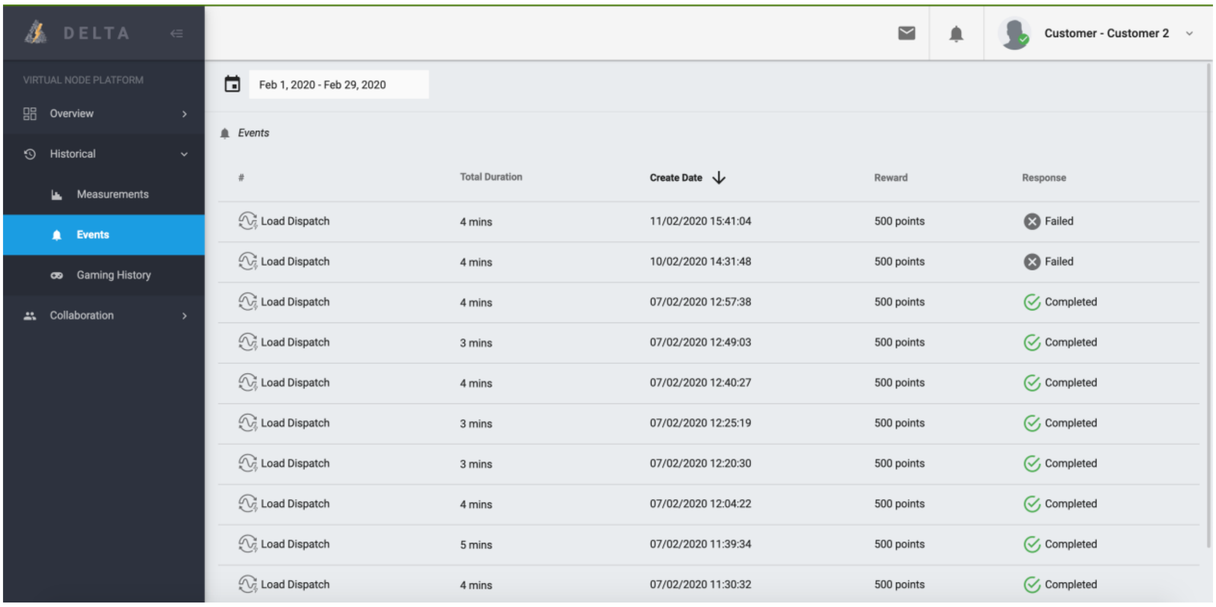

Figure 39: Customer's DR Signals

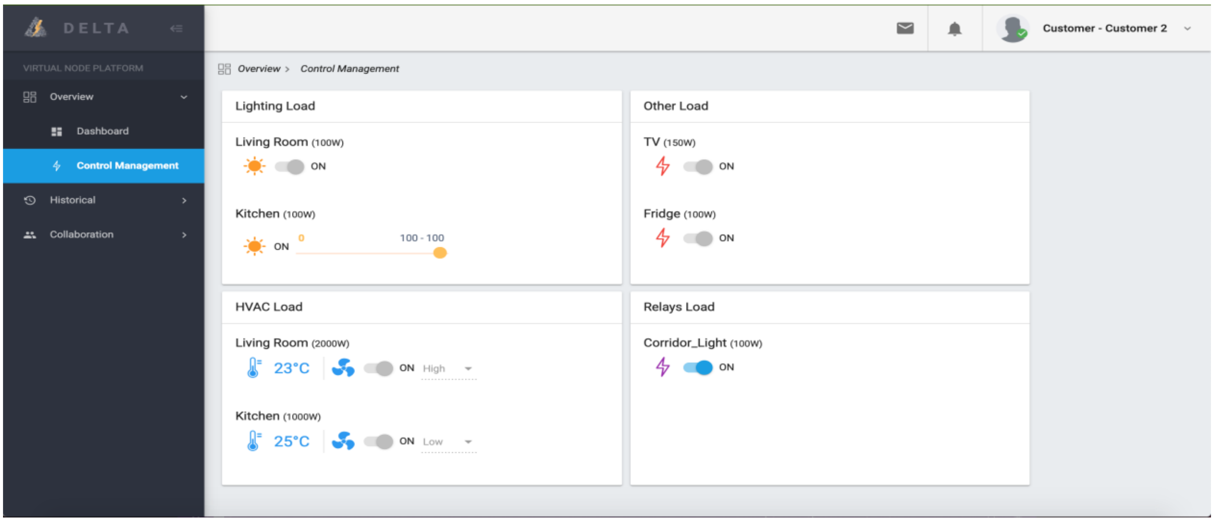

Figure 40: FEID Energy Profile

3.4.2 *Award –enabled Energy Behavioural Platform*

Following a similar approach, different functionalities are provided for the two distinct roles within the DELTA platform. The Aggregator has the capability to create, edit, manage, and overview the gamified services provided to the portfolio, whereas the customer is mainly able to participate and compete in the available to him/her games.

Aggregator UI

Aggregator can create, edit and delete games as also configure game.

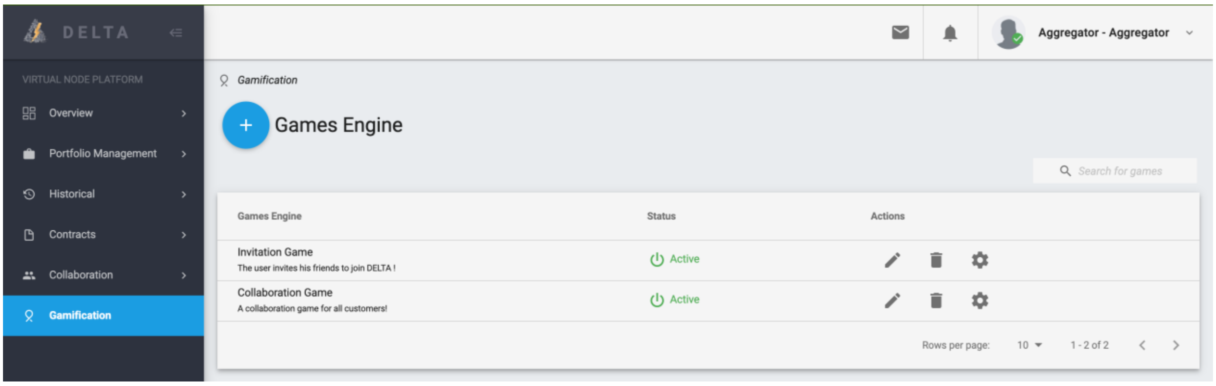

Figure 41: Game Engine UI provided to the Aggregator

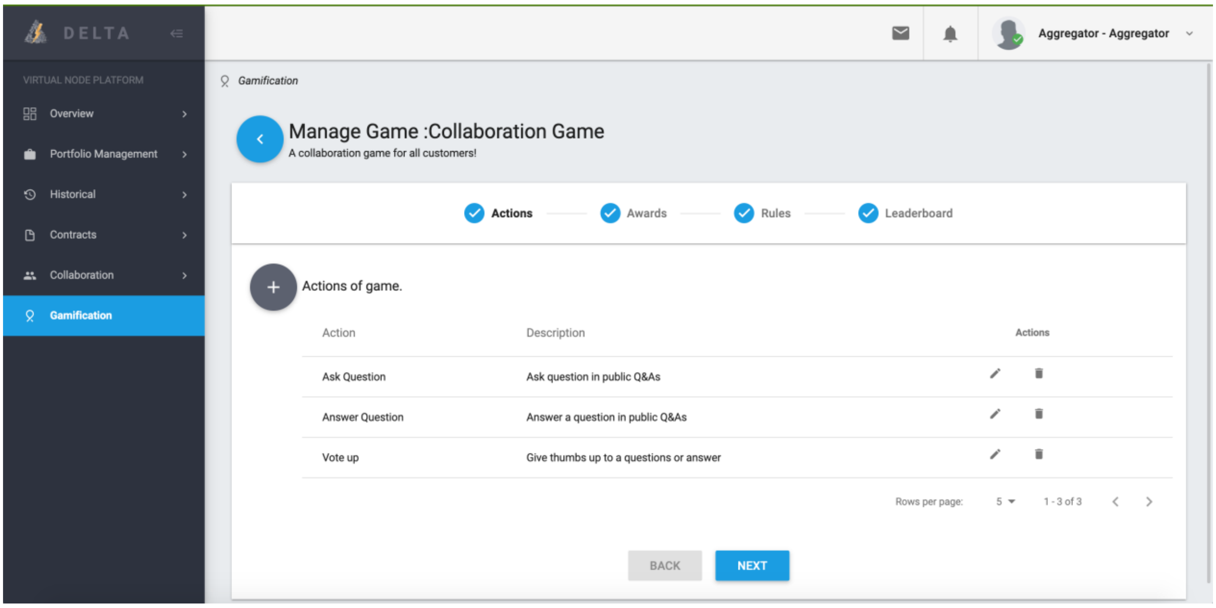

Figure 42: Manage Game Actions

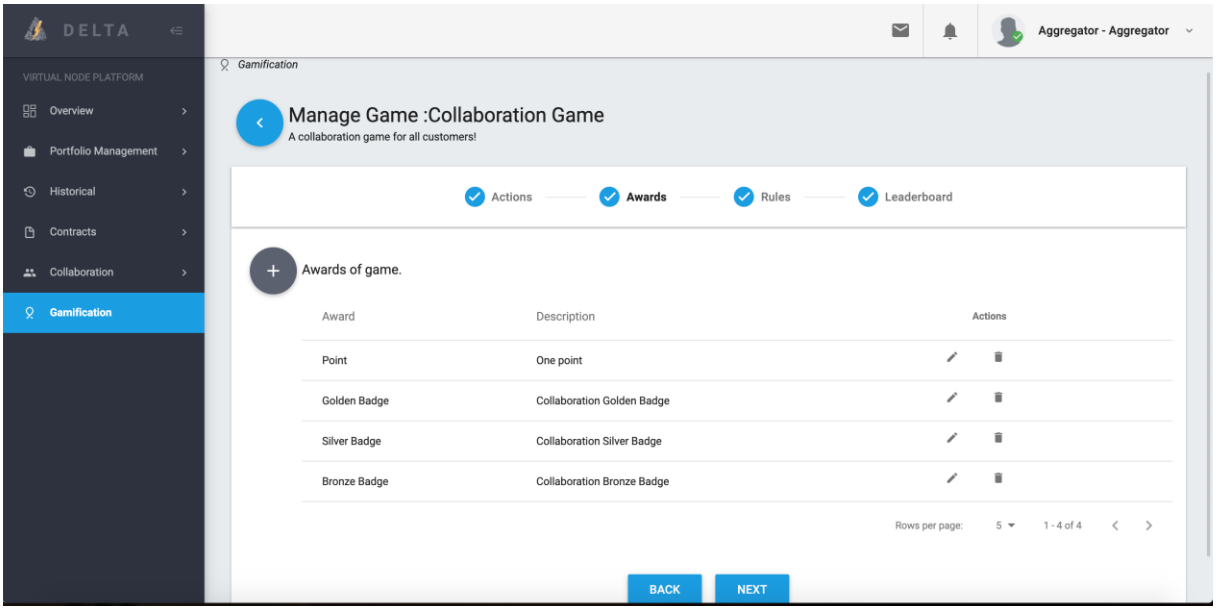

Figure 43: Manage Game Awards

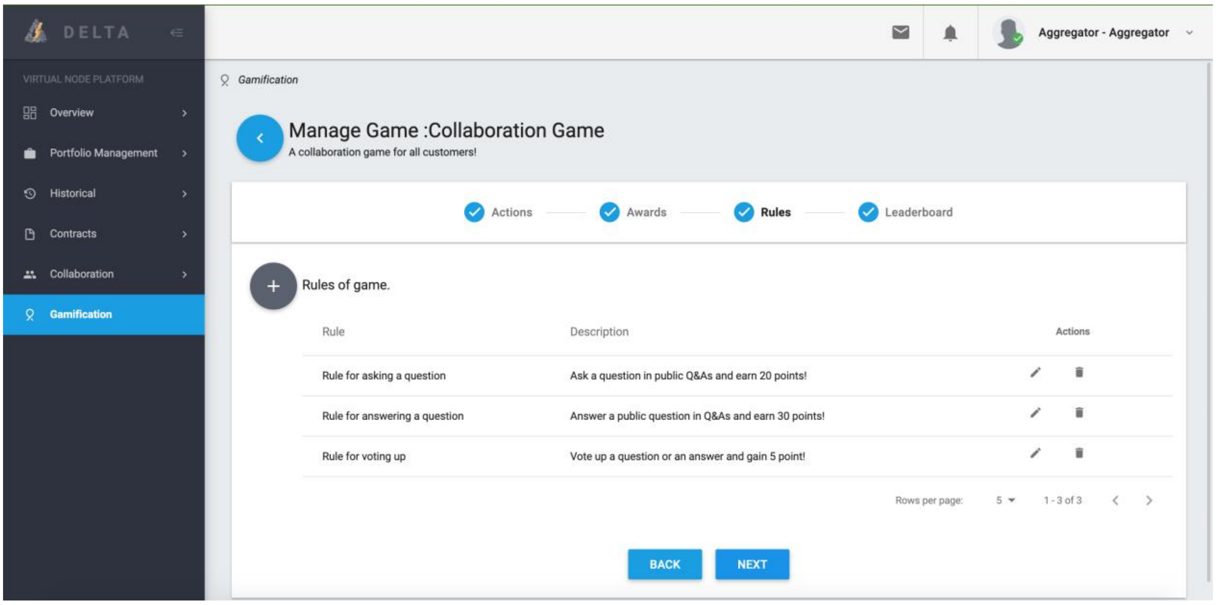

Figure 44: Manage Game Rules

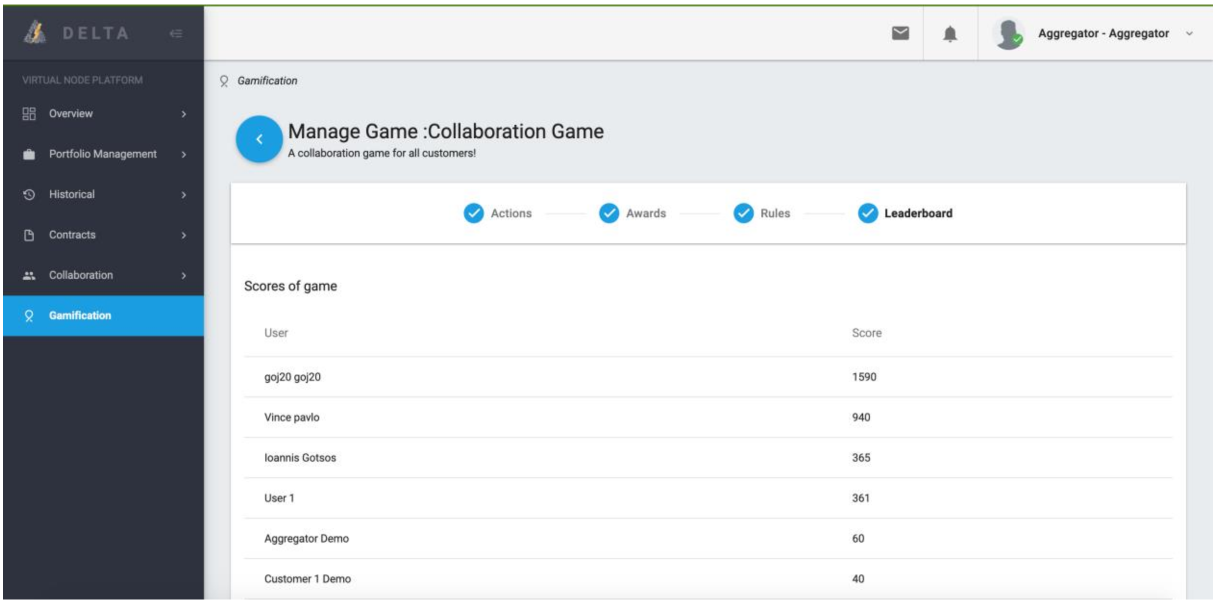

Figure 45: Manage Game View current Leader board

Customer UI

Customer can participate in games by completing or not the actions defined by the Aggregator for the game, and also keep track of game progress.

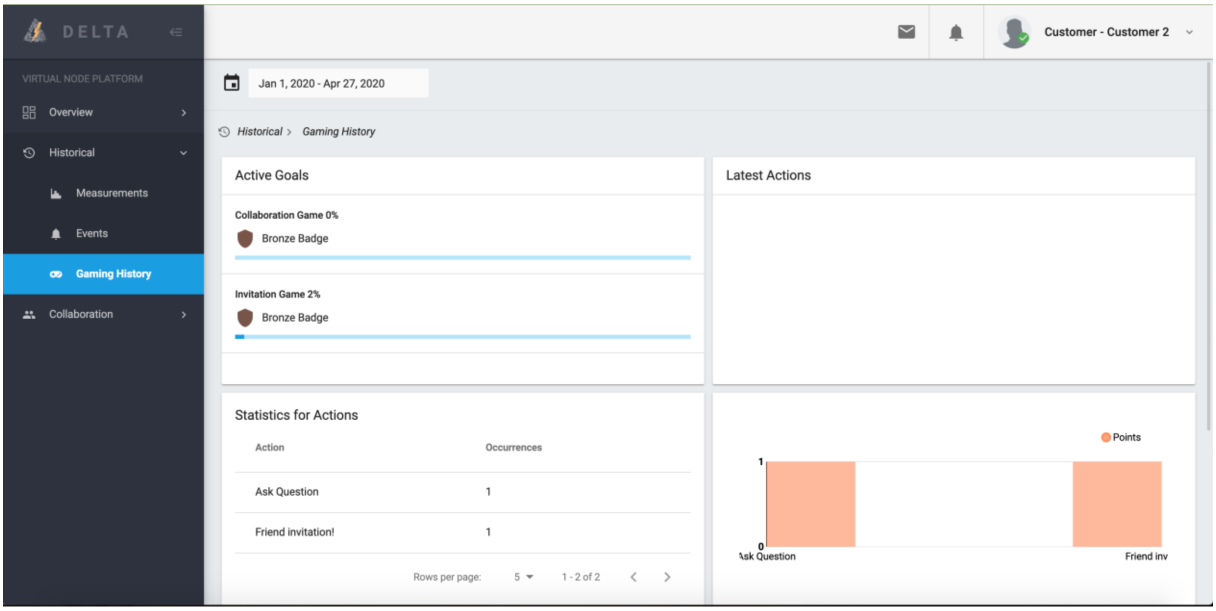

Figure 46: Customer Gaming History

3.4.3 Social Interaction and Cooperation Platform

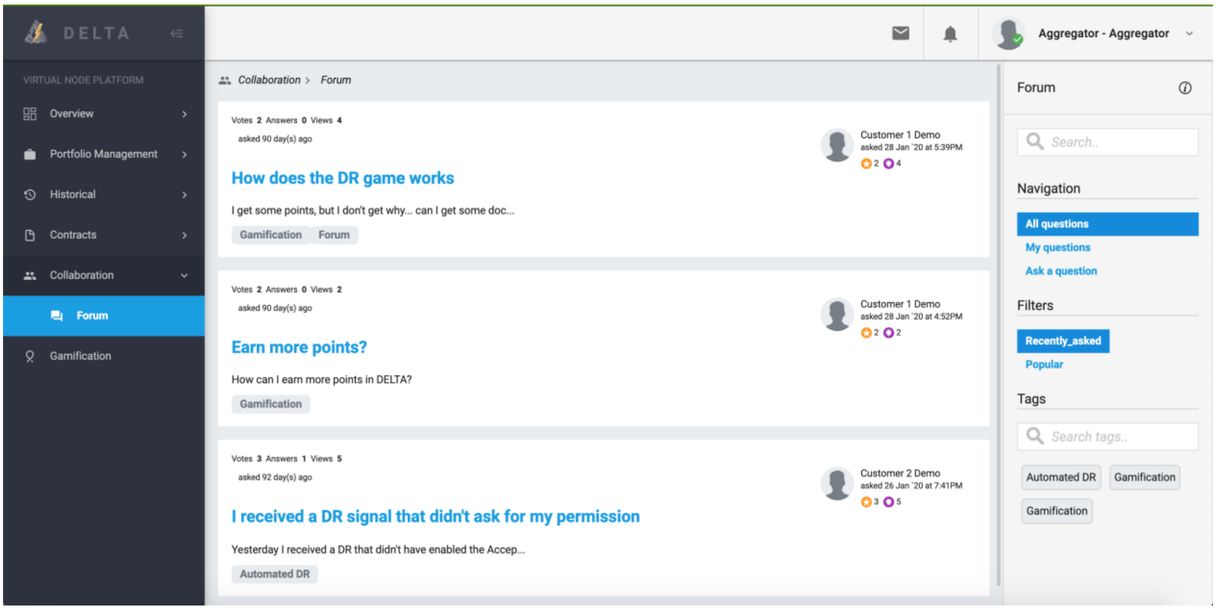

Figure 47: Collaboration Forum Aggregator's view

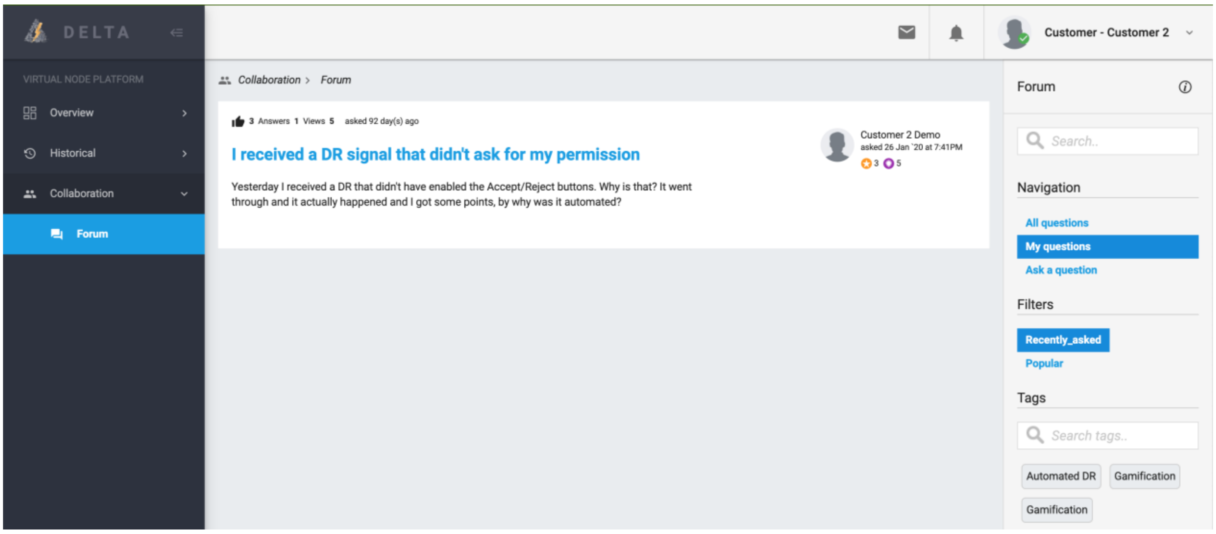

Figure 48: Collaboration Forum Customer's view

3.5 Common Information Modelling

Again for facilitating the integration purposes as well as configuration of each CIM instance, a UI has been provided. Some indicative examples of the features provided are demonstrated in the figures below.

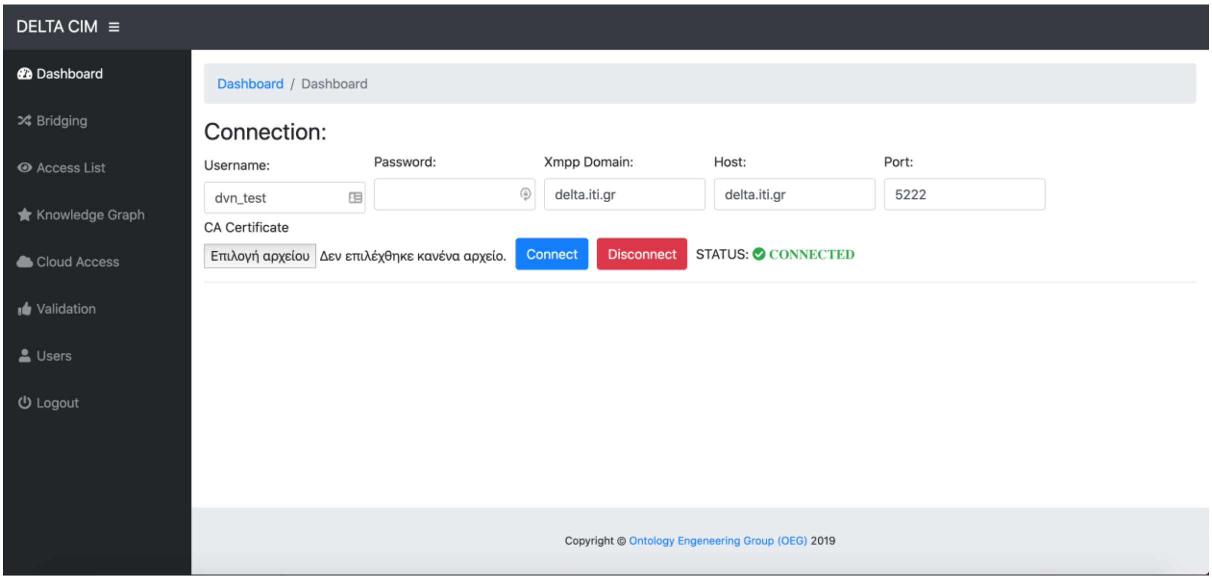

Figure 49 CIM UI – Main Dashboard – Connection to a CIM instance

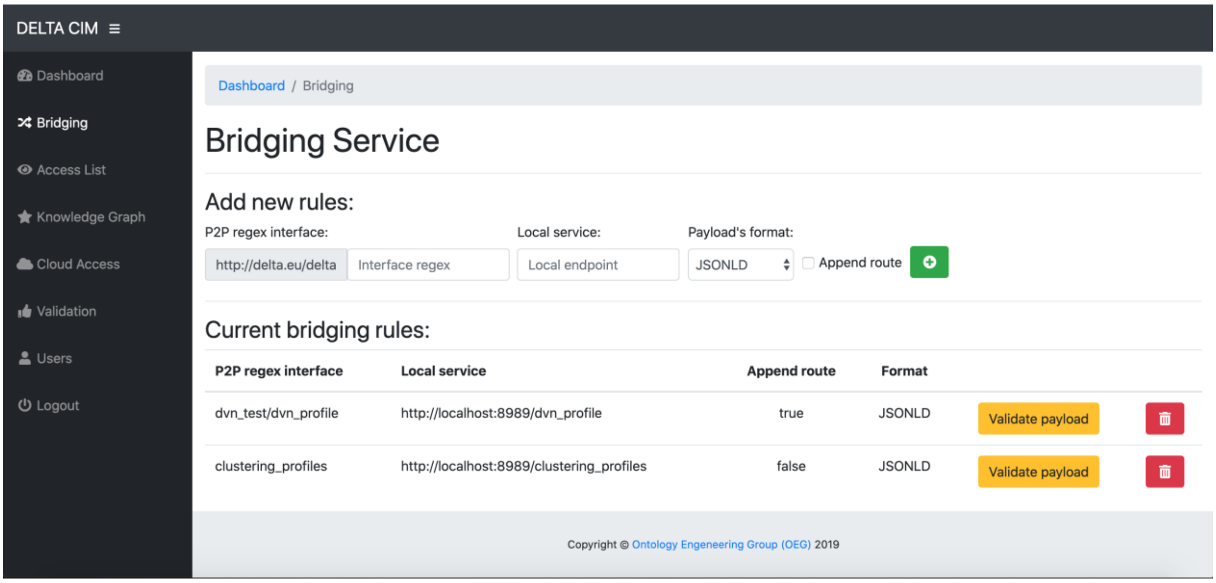

Figure 50: CIM UI - Bridging service

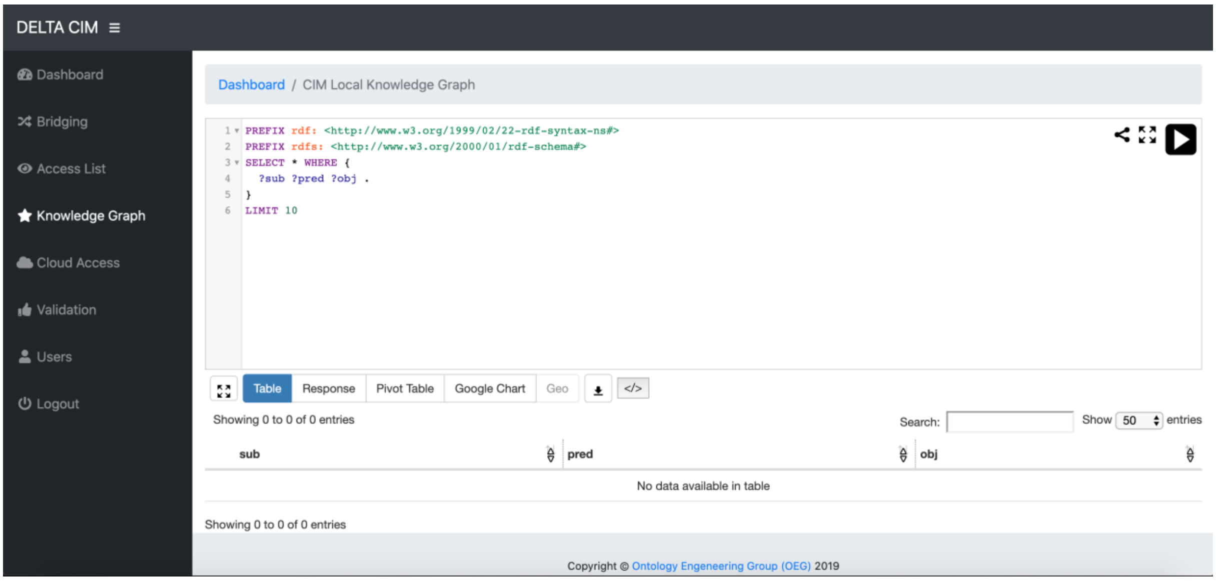

Figure 51: CIM UI – Knowledge Graph service – SPARQL querying interface

3.6 Cybersecurity Services

There are different visualisation approaches for the DELTA cybersecurity services. Starting with the available Fabric UI, the blockchain infrastructure can be fully visualised and monitored in real-time operation. This UI has already been integrated to the DVNP offering the required level of monitoring directly from the Fabric explorer. In general more details about the DELTA cybersecurity services can be found in D5.2.

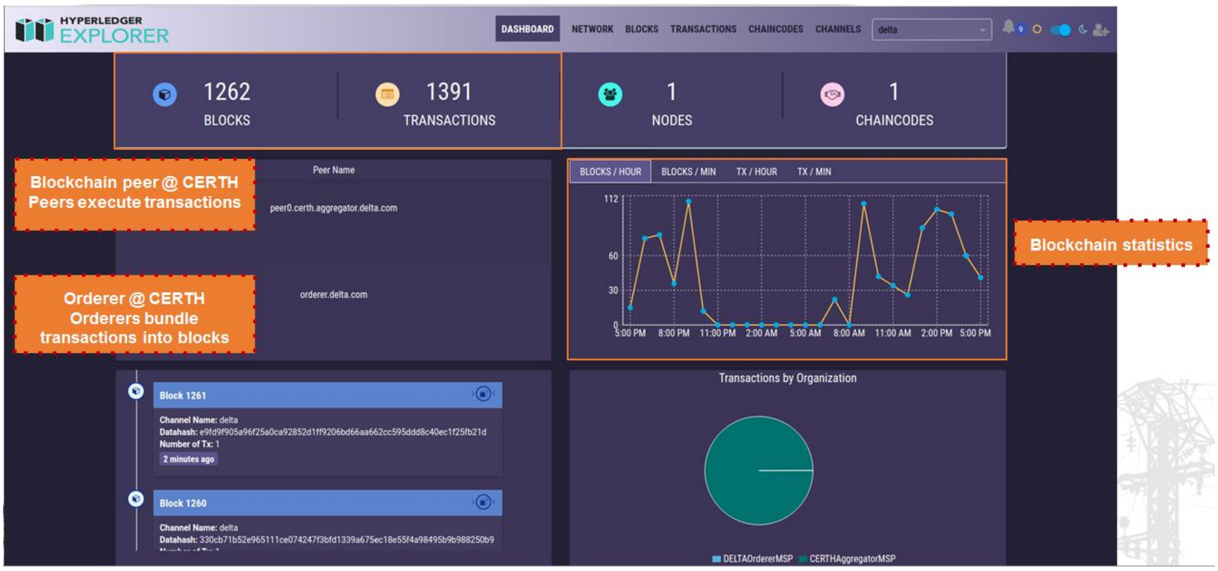

Figure 52: DELTA Cybersecurity services – The Fabric Explorer

On top of that, within DELTA, specialised dApps have been implemented towards providing custom context-enriched UIs to both the Aggregator and the Customer. These UIs are still under development and require further improvements.

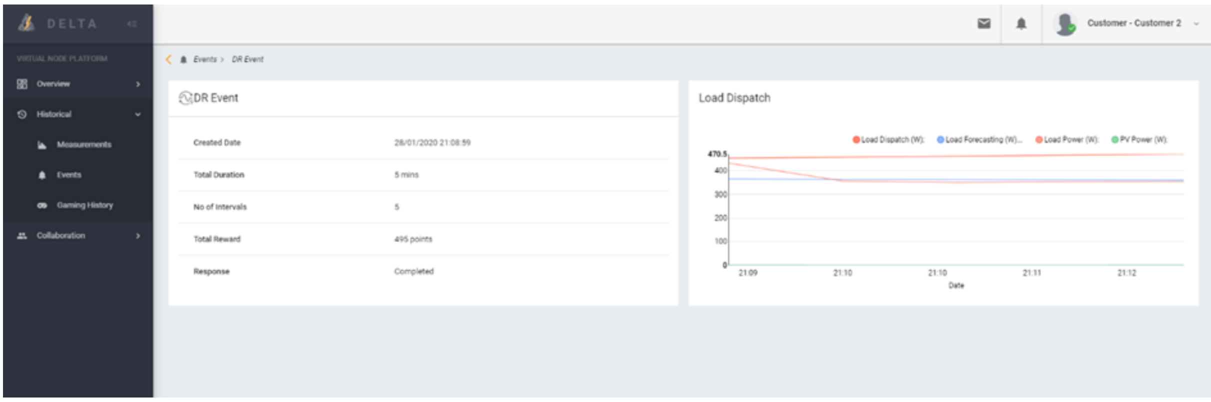

Figure 53: Smart contract visualization for a specific DR event from the Aggregator's point of view.

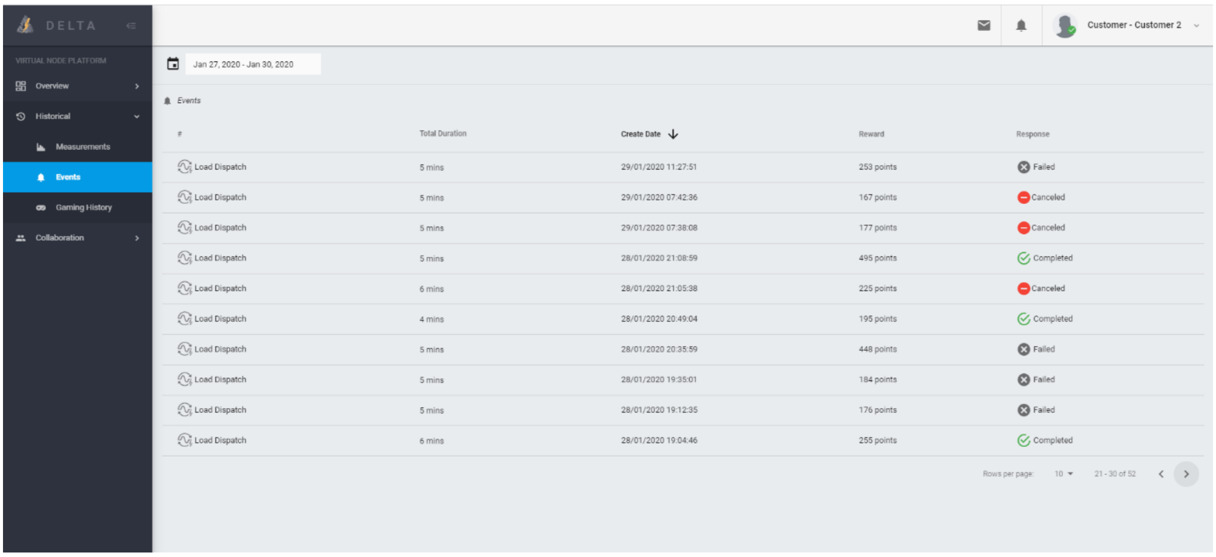

Figure 54: Smart contract visualization tool for DELTA's DR events from the customer's point of view

For the certification authority (CA) a third UI has been developed towards allowing a more elaborate monitoring of the DELTA certificates within the various DELTA entities. Examples of this UI are as follows.

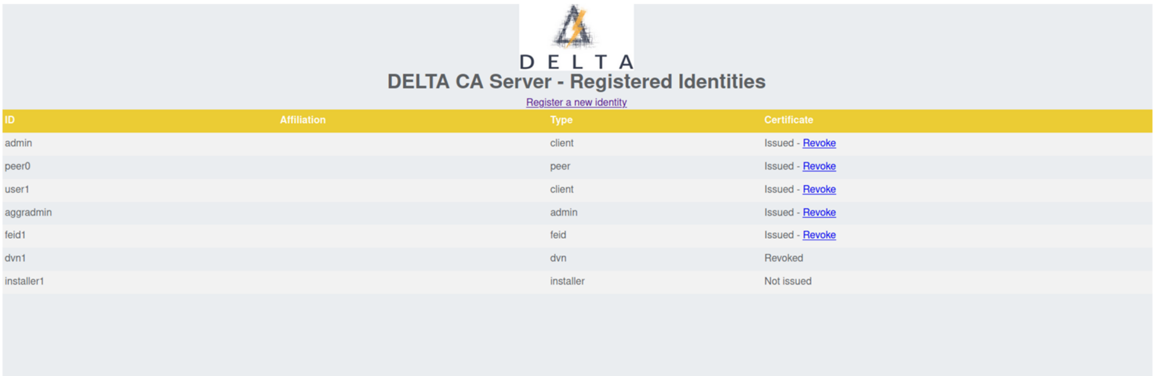

Figure 55: Web interface dashboard for monitoring identities (Aggregator's CA)

3.7 DR Emulator

In order to be able to emulate incoming DR signals/events from higher levels than the Aggregator a simple OpenADR compliant interface has been implementing for evaluation purposes. The interface has been including in the Aggregator's tools on the respective web platform.

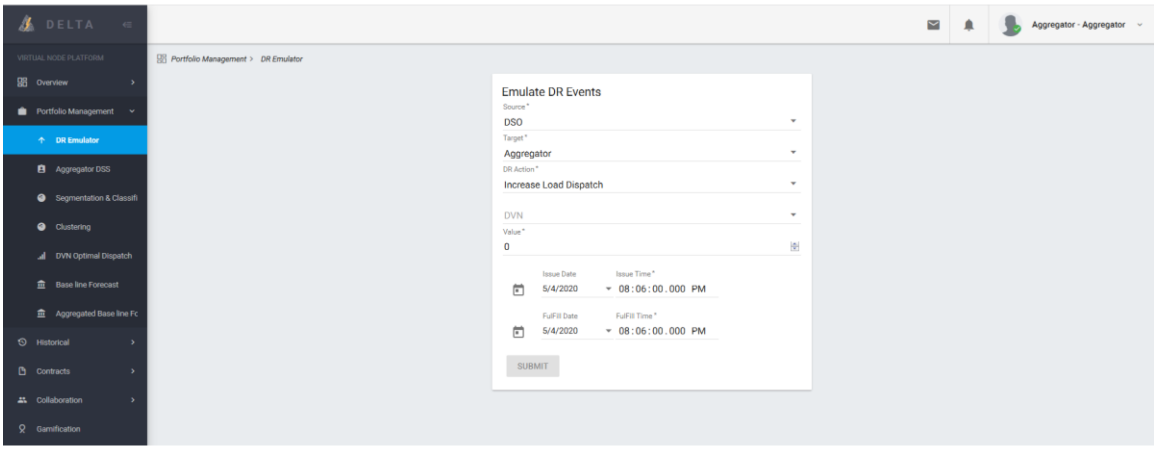

Figure 56: DR Emulator as an Aggregator Tool

4. Conclusions

In this deliverable, the detail development and integration status of all DELTA components has been presented. The DELTA framework comprises by a number of different components. All these components exchange information and data among each other through various different pathways, mainly directly when presenting internal subcomponents, or through the CIM when discussing about higher level components. All different DELTA components have their internal local repository and store retrieve adaptively data based on individual requirements. Most DELTA components, in their current form, have been developed and integrated to the DELTA overall framework.

The implementation and integration details of each component, as well as their User Interfaces (UIs), have been extensively described in this report. Following the iterative integration process adopted so far, the integration of each new version or new component is expected to be completed without any major issues on time for the pilot deployment. More information about individual components can be found in the respective deliverables, whereas the final integrated version of the DELTA framework will be documented in D6.4 in M32.

The final version will also include any changes that may occur in the architecture design based on the deployment results that may affect some of the functionalities described up to this point in D1.6.

References

[1] G. Prettico, M. G. Flammini, N. Andreadou, S. Vitiello, G. Fulli, and M. Masera, “Distribution System Operators observatory 2018,” 2019.
[2] “USEF : The Framework Explained,” 2015.
[3] H. de Heer and W. van den Reek, “White Paper: Flexibility Platforms,” 2018.