edgeFLEX

D4.1

Description of edgeFLEX platform design

The research leading to these results has received funding from the European Union's Horizon 2020 Research and Innovation Programme, under Grant Agreement no 883710.

Project Name	edgeFLEX
Contractual Delivery Date:	31.03.2021
Actual Delivery Date:	30.03.2021
Author(s):	David Ryan (WIT)
Workpackage:	WP4 – Platform and Services for Dynamically Controlled VPP Solutions
Security:	P
Nature:	R
Version:	V1.0
Total number of pages:	75

Abstract

The goal of the edgeFLEX project is to advance the role of the VPP with the use of advanced grid management techniques, effective optimisation, flexibility provisioning and trading combined with enabling solutions such as Service Level Agreement Monitoring tools, edgePMU devices and 5G capabilities.

This report describes the platform developed in the project enabling the development, testing and deployment of these services and techniques in both field and laboratory trials. It also describes the backbone services allowing the services to be linked and deployed singularly or in groups to enable the development of use cases enabling new ways to interact with and utilise the capability of a VPP.

Keyword list

edgeFLEX Implementation, use cases, Virtual Power Plant, Optimisation, Flexibility Provisioning, Control, enhanced measuring, grid management, 5G, Automatic SLA Monitoring

Executive Summary

With advances in grid management, algorithms that enable the optimisation of energy systems, such as in VPPs, and emerging systems facilitating the trading of flexibility on the market, comes the need to explore and develop use cases utilising edge cloud technologies, such as 5G, edgePMUs and brokerage to advance the role of the VPP.

These advances will enable the VPP to offer new services to system operators for grid stability, like voltage and frequency control. They will also allow the system operators, with the use of flexibility trading and dynamic service level agreement monitoring, to engage with VPP operators in a structured and mutually beneficial way to realise use cases requiring cross sector support.

The edgeFLEX project has the aim of defining and developing use cases through the research and development of new techniques and technologies and using them to build services which can be linked and deployed in both laboratory and field trials. These services form the edgeFLEX platform and, as a platform, it has the goal to provide these services and supplementary services, like advanced measuring techniques, communications, data persistence and visualisation enabling VPP operators and grid operators, such as DSOs and TSOs, to operate in a smarter and more aligned way.

This report describes the platform and details how the ICT and functional requirements of the services developed in WP1, WP2 and WP3 were gathered, how the backbone services facilitate the linkages of the services and how the deployment is designed to allow the platform to be flexible in its deployment. To develop the services and the platform components with the trial sites in mind, it was important to create methods to simulate, as closely as possible, a trial site implementation and this report details this on a service-by-service basis. The edgeFLEX platform contains several enabler services, the 5G API which levers the advanced communications capabilities of 5G, the edgePMU which will provide telemetry levering edge cloud technologies and the SLA Monitoring tool which serves the purpose of brokering cross sector actor use cases.

This report also outlines a use case that will be developed in phase two of the project where the grid management and the flexibility provisioning elements of the project come together via the SLA Monitoring tool in an automated way to either request flexibility on the local flexibility trading market or directly use DSO owned assets to perform voltage control and thus stabilise the network given differing timescales.

The early stages of the edgeFLEX project and the definition of the platform is centred on the definition and creation of the key components needed to realise the use cases in the trials and therefore the architectures presented in this report have a focus on the trials, which at the time of writing, do not have 5G capabilities. This report details the starting point of the investigation of 5G and the edge cloud with respect the edgeFLEX platform. Also, in the coming months the aim is to create a more comprehensive view of the edgeFLEX platform. This view will take into consideration the views created for the trials plus a wider view of the architecture. This wider view will encompass the use cases that can be facilitated and trialled at present and use cases that can expand the role of the VPP further in future energy systems.

This work will be expanded in the coming months and throughout the lifetime of the project with details and reporting carried out in later demonstrations, laboratory trials and deliverables.

Authors

Partner	Name	e-mail
WIT		
	David Ryan	dryan@tssg.org
	Darren Leniston	dleniston@tssg.org
	Gearoid Murphy	gmurphy@tssg.org
RWTH		
	Gianluca Lipari	glipari@eonerc.rwth-aachen.de
	Edoardo De Din	ededin@eonerc.rwth-aachen.de
	Diala Nouti	dnouti@eonerc.rwth-aachen.de
	Manuel Pitz	manuel.pitz@eonerc.rwth-aachen.de
INEA		
	Zoran Marinsek	zoran.marinsek@inea.si
	Sašo Brus	saso.brus@inea.si
	Uroš Glavina	uros.glavina@inea.si
EDD		
	Robert Farac	robert.farac@ericsson.com
	Felix Maier	felix.maier@ericsson.com
	Tuana Ensezgin	tuana.ensezgin@ericsson.com
ALPIQ		
	Ralfe Poisson	ralfe.poisson@alpiq.com
	Alexandre Juncker	alexandre.juncker@alpiq.com

Table of Contents

1. Introduction

The edgeFLEX project is divided into two phases, Phase 1 which is from month 1 to month 12 of the project and Phase 2 which extends from month 1 to month 36, and this report describes the work carried out in the first phase of the platform in terms of defining the requirements, the components, architecture, and the delivery of the platform.

The aim of the work in Phase 1 from a platform perspective was to have a Minimum Viable Product (MVP) developed to a level where the services developed in WP1 and WP2 were integrated with a set of supporting services that enable them to operate in a way that would mimic a deployment like that of a field deployment. The end product of this aim is a subset of the edgeFLEX platform which will, throughout Phase 2, go through a series of iterative improvements via the edgeFLEX improvement model in Figure 1 which involves the gathering of feedback from the trial site deployments and lab trials to bring the platform and services from an MVP state to a level of maturity that may be exploited in a commercial setting by project partners at the end or shortly after the project is complete.

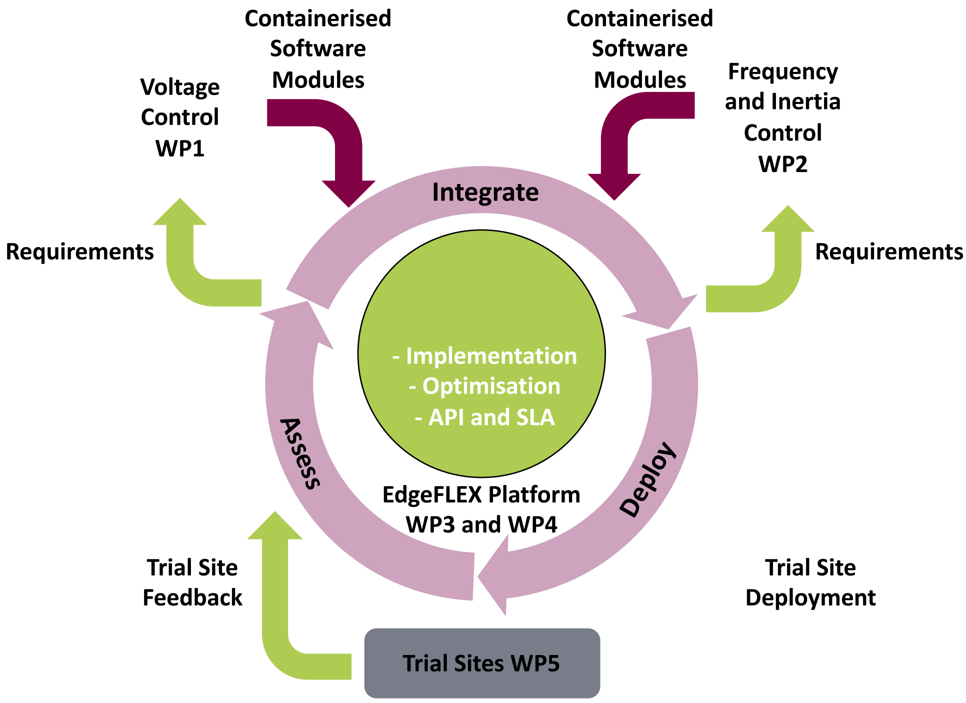


Figure 1 edgeFLEX Improvement Model

The architectures presented regarding the platform are very trial site focused as the initial stages of the work were focused on gathering the requirements and components that will be used in phase 2 to facilitate the trials. The next steps in defining a project wide architecture will take these components and align them with the goals of the project in terms of advancing the role of the VPP beyond the life of the project utilising the edgeCloud, the brokerage and the edgePMU to further provide platforms for new business models and exploitation avenues.

1.1 Related Work

This report is based on work carried out in the first phase of WP4, in terms of defining the functional and non-functional requirements to develop a system architecture that will facilitate the control services developed in WP1 and WP2 (Task 4.1), develop a set of interfaces and a method of integration (Task 4.2) that will allow for the implementation of the control services and the 5G functionalities from WP3 (Task 3.2) in both lab and field trial tests. To not only limit the platform to the integration of the control services, this report will also describe how the other services, like the edgePMU (WP3, Task 3.3), Flexibility Trading and the interaction with the GOFLEX platform (Task 4.5), the SLA Monitoring Tool (Task 4.4) and VPP Optimisation (WP3, Task 3.4) will form the edgeFLEX platform that is designed to facilitate, not only use cases that centre on the control services but ones that can trade flexibility. Figure 2 illustrates how the tasks in each of the Work Packages relate to the platform and how these feed into the trials that are being developed in the tasks in WP5.

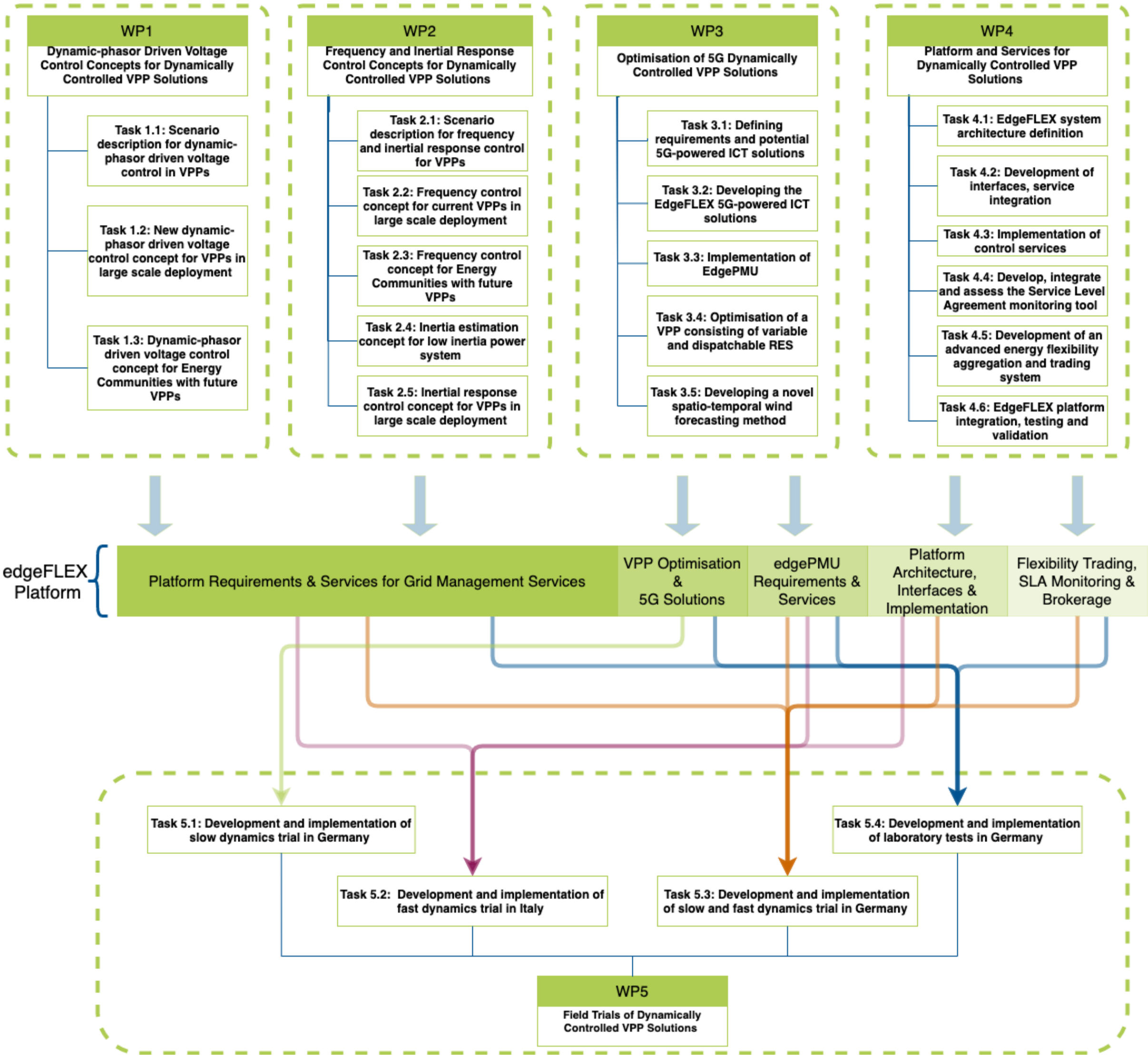


Figure 2 edgeFLEX Tasks and the Platform

The platform developed in WP4 and described in this report is considerate of the trial site implementation and is very much designed to be a staging point for the Research and Innovation Concepts prior to the trials.

1.2 Objectives of the Report

The main objective of this report is to describe the edgeFLEX Platform that will enable the use cases, both those developed in the research concepts and those that we discover during our research in the project and beyond.

1.3 Outline of the Report

This report describes how we gathered and derived the functional and non-functional requirements for the platform (Section 2), developed the interfaces, developed service integration mechanisms and supplementary services (Section 3.2) that enable the monitoring and running of the services in both laboratory and field trial scenarios. Key to the platform and its relevance to the field trials is the method of deploying and linking the platform components, which is also outlined (Section 3.1). This report will also cover a brief overview of each service and how the platform facilitates their integration both architecturally and technically (Section 3.4). It will also describe the enabler services, which serve the purpose of facilitating communications, enhanced measuring, and cross sector actor brokerage (Section 3.3). The architectures for VPP Optimisation (Section 3.5) and Flexibility Trading (Section 3.6) are also detailed. Given that the edgeFLEX platform has a goal of facilitating cross sector use cases, Section 4 details how the platform is developing a use case that automatically engages the Flexibility Market if a certain set of conditions are met on the Grid Management side via the edgeFLEX SLA monitoring tool.

1.4 How to Read this Document

This document is the first description of the edgeFLEX platform and therefore there are no prior iterations of it. There are, however, concepts in here that will be summarised but a more comprehensive explanation and the background research to them will be contained in other deliverables, these include:

- D1.1: Scenario description for dynamic-phasor driven voltage control for VPPs and D1.2: Dynamic-phasor driven voltage control concept for current VPPs in large scale deployment which provides the background to the Voltage Control service detailed in section 3.4.1.
- ***D2.2: Frequency control concept for current VPPs in large scale deployment*** which provides context and background for the Frequency Control service detailed in section 3.4.3
- ***D2.4: Inertia estimation concept for low inertia power system*** which provides the background to the Inertia Estimation service detailed in section 3.4.2.
- ***D3.2: Report on VPP optimisation*** that provides context for the VPP Optimisation services detailed in section 3.5

All other concepts that are not in a deliverable from another work package will be explained in a comprehensive way or linkages to prior research from other projects will be cited and referenced where appropriate.

This deliverable very much captures the work carried out in Phase 1 of the project which is centred on the architecture and description of the platform in relation to the trials. In Phase 1 also, the focus is on getting the basic requirements gathered and the software built around those requirements so that they can be refactored and improved upon in Phase 2 and therefore the subsequent deliverables, ***D4.2: Description of edgeFLEX MVP(M18)*** and ***D4.3 Description of internal interfaces for control services(M24)***. These deliverables, in particular D4.2 will contain a more comprehensive architecture and one that is cognisant of the big picture issues the edgeFLEX project aims to address.

2. Platform Requirements

The edgeFLEX platform consists of two main groupings of concepts and services, Grid Management and Market. The tools and concepts on both sides differ in terms of the way they are represented, how they are used and for what purpose they are used. During the initial stages of determining the components required and the overall architecture of the edgeFLEX platform it became clear that the architecture was layered with different components and concepts have more relevance to one layer over the other. This is evident when attempting to align the grid services with the market interaction where the grid services, from the outset, find more relevance in the layers representing the communications and direct implementation whereas the market functions are more founded in the business impact layers. To align both it was important to view each concept in its own context first and then look at them in the context of the overall platform. This was carried out initially using three tools, a **requirements survey**, that looked at the needs of each service in terms of how it would operate in the platform, the classification of the services into grid management and Flexibility Trading and concepts using a **classification matrix** and a **solution architecture definition workshop** based on the TOGAF [1] approach that allowed to look at the services and use cases in a layered way.

2.1 Requirements Survey

Given that the partners participating in edgeFLEX from a technical perspective have a grounding in ICT, Software Engineering and Electrical Engineering and to build the edgeFLEX architecture it is important for each field of expertise to understand what the platform needs to support to facilitate the effective running of the services. To discover these needs as early as possible a survey containing a set of targeted questions was sent out to the service owners in month one. This exercise was geared, not only to find what we need to build to host and deploy the services, but also to aid the service owners discover aspects of their service that they may need to rework to take it from a laboratory setting into a deployable platform. The questions contained in the surveys centred on topics such as:

- the types and sizes of input and output data to and from the service.
- the optimal latency required for the service to run.
- the potential security concerns associated with running the service.
- the software language and operating system needed to run the service.
- does the service require persistence?
- communications requirements
- and where does the service owner envisage the service running.

The answers to these questions helped us understand the services from an ICT perspective and enabled us to define a set of basic functional and non-functional requirements needed to build a generic software component to host the services. From the workshops described in the section on Solution Architecture it was discovered that a docker hosted microservice architecture was the most effective way to host the and instantiate the services

2.2 Classification Matrix

Given the wide range of scope of the services being developed and trialled in edgeFLEX it was important to explore the context of the component in terms of who would use it and who would provide it with the goal of identifying synergies between the components.

[1] https://www.opengroup.org/togaf

In edgeFLEX there are broadly three types of service:

- Grid Management, which contains non-commercial/public function activities: grid infrastructure – management, operation, and maintenance of the grid. In the case of edgeFLEX some of these tools and services were outputs of such projects as RE-SERVE, SOGNO & PLATONE.
- Market Interaction, which consists of commercial activities like trading, production, storage, consumption. In the case of edgeFLEX it will engage the further development and innovation of automated flexibility trading platform developed by GOFLEX project and of ALPIQ trading of flexible products in intraday and day ahead markets.
- Enabling Services, that focus on allowing the services interact with each other, with the grid assets and with the communications network.

In the early stages of the project, it was important to understand what services were applicable to what grouping but more importantly what services were applicable to what function or actor in the grid and Figure 3 is an aggregation of an excel sheet distributed to the service owners at the early stages of the project. In this matrix we considered the structure of the grid system and the possible use cases, the complete set of tools and services structures into the following types:

Grid Management Services provided:

- By DSO for DSO
- By TSO for DSO
- By DSO for TSO

Flexibility trading services provided by commercial players:

- For DSO
- For TSO
- In Local Energy Communities (LEC) markets
- In Market Balance Area (MBA) markets
- Reactive energy trading (for DSO)

While Reactive Energy Trading is listed as a service it is also a concrete objective in edgeFLEX and one being addressed as part of Task 4.5 in Work Package 4.

Service	Grid Management				Flexibility Trading			
	by DSO for DSO	by TSO for DSO	by DSO for TSO	for DSO	for TSO	in LEC Markets	in MBA Markets	for Reactive Energy Trading
				Services Provided By Commercial Players				
Frequency Control	YES	YES						
Inertia Estimation	YES							
Dynamic Phasor Driven Voltage Control	YES							
Dynamic SLA Monitoring Tool	YES					YES		YES
edgePMU	YES		YES	YES				
5G API	YES	YES	YES	YES	YES	YES	YES	YES
Spatio-Temporal Wind Forecasting						YES	YES	
VPP Optimisation						YES	YES	
Stochastic Intraday Market Modelling						YES	YES	
FlexOffer				YES	YES	YES	YES	YES
FlexOffer Agent				YES		YES		YES
Automated Flexibility Trading Platform				YES		YES	YES	YES
Industrial Energy Management System				YES	YES	YES	YES	YES
Advanced Energy Flexibility and Trading System						YES	YES	YES
Reactive Energy Trading				YES				YES

Figure 3 edgeFLEX Matrix

The architecture matrix is the base for a “reference architecture” being developed in the project, which superimposes all the potential use cases with their reduced actual architectures into one imaginary comprehensive case. The building blocks and the relationships in different “architectural slices” that appear in several of them are the base for optimised development, design, and deployment of services in various use cases.

2.3 Solution Architecture Workshops

2.3.1 Architecture Approach

For the construction of the Architecture, the decision was taken to make use of the TOGAF 9 [1] (The Open Group Architecture Framework) and ArchiMate 3 [2] standards from The Open Group [3]. The TOGAF standard is quite extensive and requires tailoring for the purpose of specific architecture work. Within the scope of the edgeFLEX project, the terminology, architecture layers (or domains), methodology (ADM) and artefacts are utilised. To ensure that the models and diagrams are clear and easy to interpret, the ArchiMate modelling language has been utilised for the creation of the TOGAF artefacts. This standard allows a common set of symbols and connectors with clear meanings and definitions.

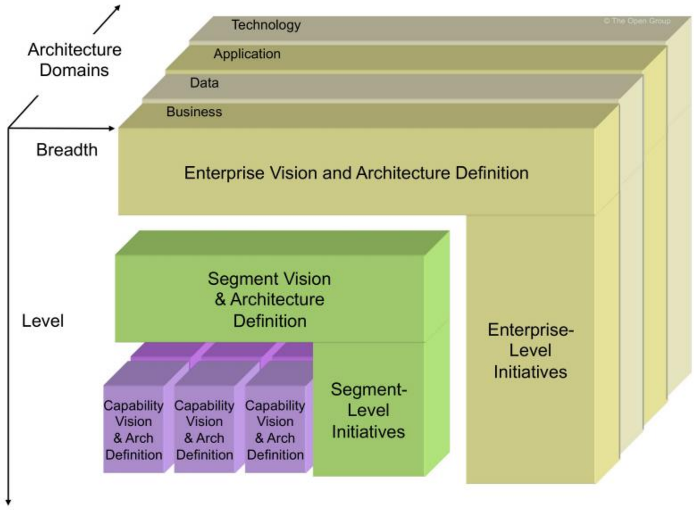


Figure 4 TOGAF Architecture layers

Figure 4 illustrates the architecture layers, or domains and in the definitions of the solution architectures for edgeFLEX the following are utilised:

1. Business Architecture: This includes artefacts reflecting stakeholder analysis, motivations, roles, actors and business processes.
2. Application & Data Architecture: This layer contains artefacts which model the composition and interaction of application components, as well as the data objects which move between them.
3. Technology Architecture: Within this layer, artefacts model the physical technology components which are then mapped to the logical application components.

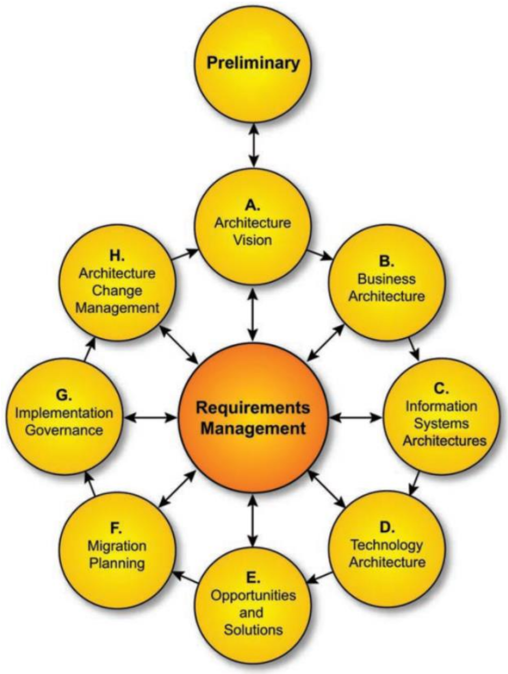


Figure 5 TOGAF Architecture Development Methodology (ADM)

TOGAF define a structured methodology for developing architecture. This methodology, shown in Figure 5, starts with defining a vision, which was provided by the edgeFLEX initialisation documentation, followed by the sequential development of the business, application and finally technology architectures. Following the development of the architectures, the rest of the methodology then provides guidance as to how the architecture is split into work packages, and then planned for implementation (perhaps with intermediate steps using transitional architectures) and then governed through conformity reviews before final delivery.

The TOGAF/ArchiMate artefacts, which has a specific notation that is detailed in Figure 6 and in the workshops that helped define the edgeFLEX architectures the following were used:

- Solution Concept Diagram: Contains the architecture vision, which combines elements across multiple layers, to depict the high-level model describing the main idea of the system.
- Application Communication Diagram: An Application & Data layer artefact which models the structure of application components and data flows between the application components.
- Implementation Diagram: A Technology layer artefact which models the environment, computational components and technical interfaces which support the application components.

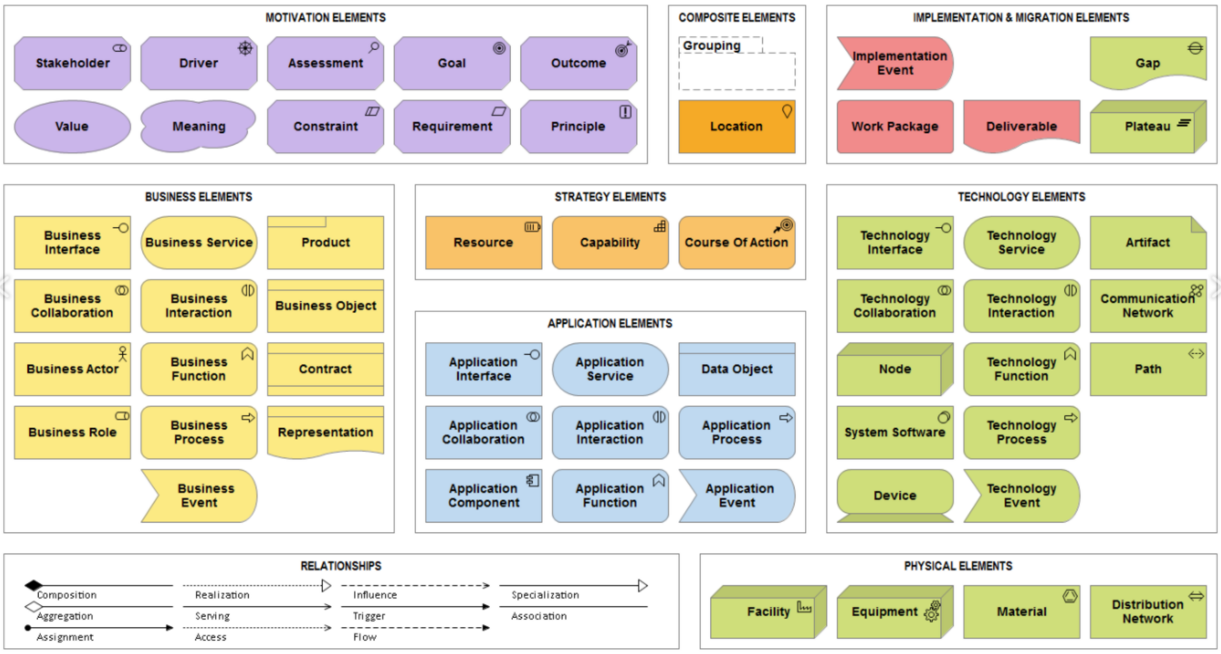


Figure 6 ArchiMate 3.0 notation

2.3.2 Solution Architecture Methodology Application in edgeFLEX

The TOGAF methodology was utilised in the initial phases of planning the edgeFLEX platform, by defining solution architectures for each of the services (Voltage Control, Frequency Control and Inertia Estimation) and how these interact with the grid side, VPP and the edgeFLEX platform. This was done with input from WP1-4 partners through several solution workshops. Each workshop focused on one of the control services to determine their key components and the backbone edgeFLEX services they will utilise (e.g., data persistence, visualisation, and data bus). From these workshops three architectures were created for each control service; a solution concept architecture outlining the overall picture of how the edgeFLEX platform, VPP and Grid side interact in the context of that service.

Secondly, a more detailed application communication architecture, providing a lower-level picture of the control service interaction with components such as data persistence, data busses, measurement devices and communication interfaces (REST and MQTT) were created.

Finally, a business architecture, outlining the benefits of each service to specific actors was created. The architectures created as an output of these workshops brought clarity in how the components and actors involved in the edgeFLEX solution interact, operate and communicate, providing a strong starting point to being development on the edgeFLEX platform and bringing an understanding between each partner involved in the development tasks. The singular architecture diagrams for the Voltage Control, Inertia Control and Frequency Control can be viewed in annex A.1.1.2, A.1.2.2 and A.1.3.2 respectively.

3. edgeFLEX Platform Description

The edgeFLEX platform is a suite of services, focused on Grid Management, VPP Optimisation and Flexibility Trading, that can be utilised in isolation or as a linked set of services to help sector actors, like DSOs and VPP operators, realise use cases that range from single focused use cases, like performing Voltage Control, to more complex ones, like performing interactions with external market platforms via the creation of flexibility offers.

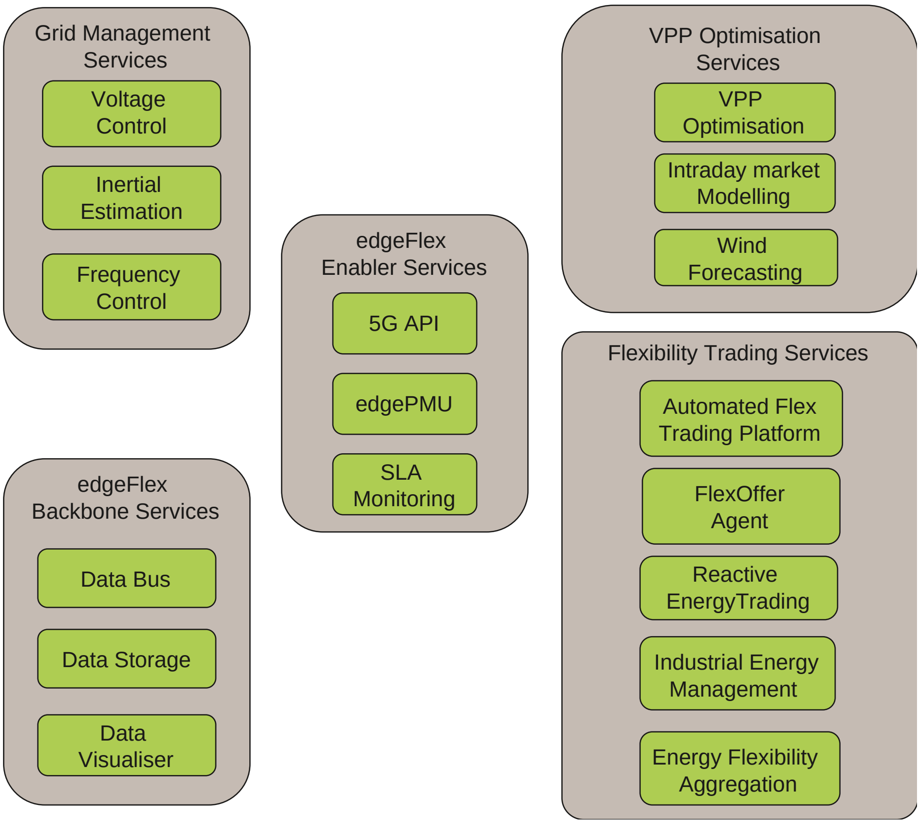


Figure 7 edgeFLEX Platform Components

These services can be categorised into 5 specific categories, Grid Management, Flexibility Trading, VPP Optimisation, Backbone Services and Enabler Services. Figure 7 shows the research and innovation concepts developed edgeFLEX categorised in this way.

The design of the edgeFLEX platform is centred on allowing components to be instantiated in isolation, for example Voltage Control, without monitoring, the database service or visualisation by configuring it to receive its data from data busses already instantiated in the DSO system. This would allow the DSO to integrate a service like Voltage Control into their system without the overhead of deploying the backbone services and instead leveraging the components already existing in their system. Similarly, the platform can be used to realise use cases that require services that can be in multiples of the categories above, like that detailed in section 4 where we demonstrate a use case that utilises Voltage Control, the FlexOffer Agent, Reactive Energy Trading, the edgeFLEX backbone services, the edgePMU and the SLA Monitoring Tool.

The following sections describe the categories under which each service falls, the low-level implementation of each service in terms of the other services it interacts with and how from a communications and ICT perspective.

3.1 edgeFLEX Deployment Mechanism

For the edgeFLEX platform to serve the purpose of facilitating a diverse range of use cases that can be deployed in isolation or as a group it is important that each service, where possible, can be configured when being instantiated to act a certain way or connect to a certain data source or be accompanied with some other services. To facilitate this, a microservices architecture [4] was chosen, which at its core has the philosophy that each service should be loosely coupled, independently deployable and highly maintainable. A key concept that allows this architecture to be built is containerisation which allows the code or application to be ran on a software-based operating system with the requirements in terms of networking and programming languages needed preinstalled within the container. This means that the developer or researcher can write the applications as they are intended to be written independent of the operating system of the host machine. In edgeFLEX we choose Docker [5] as the containerisation solution and we heavily use docker-compose [6] to orchestrate, configure and link the containers. The Service Configuration File, as seen is in Figure 8, is an important component as it allows the developer and user of the service to deploy the service in a way that best suits the use case they are running. This file is injected by docker-compose into the service at run time and once the service is up the configuration file can be deleted as the service is then configured. By using a highly flexible method of configuration for the service the edgeFLEX platform can dynamically switch from using the edgeFLEX data bus or persistence to use pre-existing data sources within the systems already deployed by the system operator.

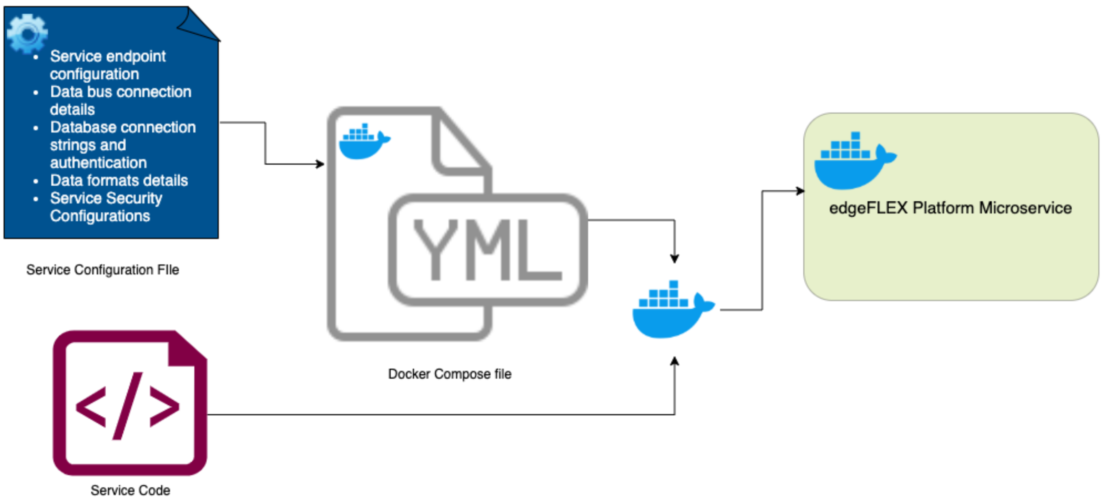


Figure 8 Service Deployment with Docker

While Docker Compose takes care of the networking and configuration of the services, they are deployed using Bash [7], a scripting utility that allows commands to be run on the host machines. This script and how it is run is parameterised, and these parameters indicate what services the system is deploying. This allows the platform to be deployed in whatever mode the use case requires with whatever backbone services, enablers, flexibility or VPP optimisation services needed. An example of the deployment would follow that illustrated in Figure 9.

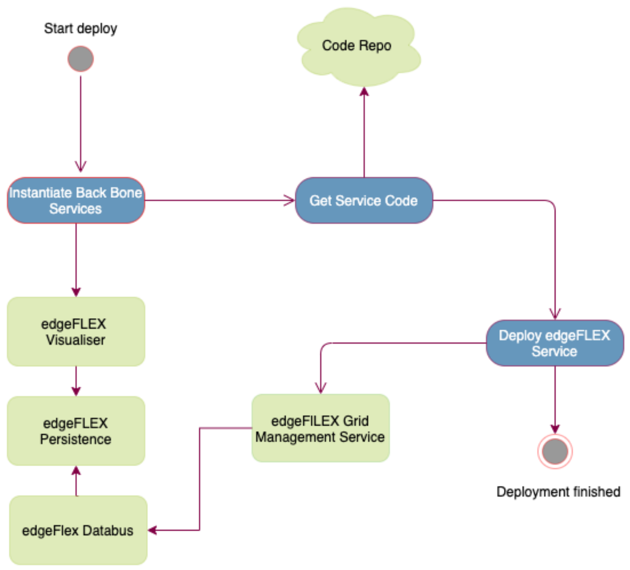


Figure 9 Basic Deployment Flow

Core to the edgeFLEX platform is that the code repositories that hold the service code are not stored in one repository and are pulled in by the platform when the platform is being deployed. This ensures that the electrical engineering research can take place in tandem with the trial site and lab deployments to enable a strong continuous improvement pipeline, where the feedback from the trial sites can be received by the electrical engineering researchers and the refactoring or improved service can be deployed in a dynamic way via the platform deployment.

3.2 edgeFLEX Backbone Services

To facilitate the running of the edgeFLEX Grid Management services and to trial the concepts presented it is necessary for them to receive the data, be able to store it and visualise it so that the services and their impact on the grid can be monitored. In the edgeFLEX platform these requirements are handled by the backbone services. Taken as a set of components from the SOGNO reference architecture and used to facilitate the grid resilience services in the trials in terms of providing the data interfaces for the on-pole sensors and the storage and visualisation for post processing and monitoring. These are key services as they provide an interface for measuring units, like the edgePMUs, to send readings into the platform, a method of persistence in the platform, where the readings are stored for scheduled analysis by some services, and a visualisation service that allows the outputs of the platform be monitored. The flexibility of this platform allows these services to be instantiated with the other services or they can be already hosted within the trial site providers DMS (Distribution Management System).

3.2.1 edgeFLEX Data Bus

The edgeFLEX data bus service is a component of the platform that can enable the receipt of data from the field devices deployed as part of the edgeFLEX field trials and the sending of the outputs of the grid management services to the grid devices so that the set points can be applied. It is based off the open source DockerMQTT [8]application (an output of the RESERVE project) and provides an SSL secured and password protected connection to an MQTT broker for the receipt of data from the services and field devices. MQTT is a lightweight publish/subscribe messaging transport protocol and is widely used in Internet of Things applications. The implementation used in the edgeFLEX platform levers the Eclipse Mosquito [9] opensource software that exposes the publish and subscribe clients as methods to communicate with an MQTT server.

While the edgeFLEX platform, when deployed and instantiated in full, the data bus is part of the deployment, the flexibility of the platform facilitates the services, via configuration, connect to a data bus owned and already existing in the within system operator's grid or flexibility management system.

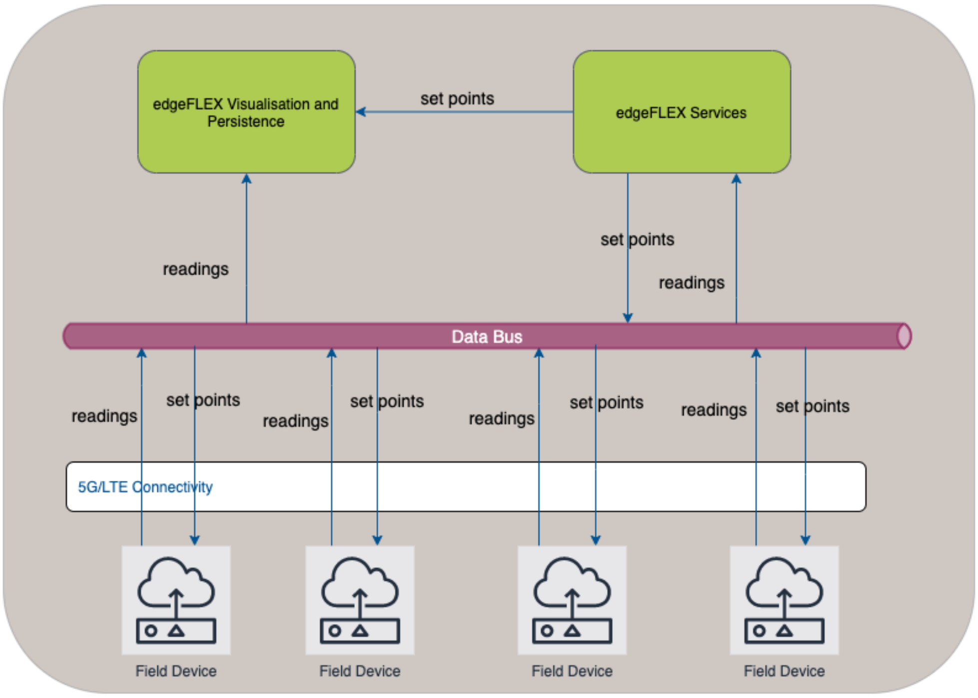


Figure 10 edgeFLEX Data Bus connection to services

3.2.2 edgeFLEX Persistence

Data Persistence in platforms are used for several reasons, to allow for data aggregation, postprocessing of the data, historical data analysis, visualisation, and monitoring. For the implementation of Voltage Control and Frequency Control services are using this service purely to facilitate monitoring and historical analysis but for the case of Inertia Estimation, the persistence service facilitates the need to post process the stored data for a period to estimate the inertia.

From the requirements gathered in the early stages of the project it was identified that the data being received from the devices to be used as input to the services was most suited to be stored as time series data. To perform this task in the edgeFLEX platform we used a database technology called InfluxDB which is a standalone timeseries database developed by InfluxData [10]. It has a docker implementation also, which comes with the prebuilt server and storage mechanisms to allow for the easy deployment of the persistence later of the platform. Core to choosing on a database technology is assessing its suitability in terms of how other services can interact with it and from the assessing requirements the availability of a python client [11] to provide this interaction was crucial.

3.2.3 edgeFLEX Visualisation

In platforms, such as edgeFLEX, a means to visualise the performance and the outputs of the services and the concepts are important. This allows the researchers and all other stakeholders to view the data being sent up from the field and the outputs of the algorithms. In the edgeFLEX platform we have developed a service, based on Grafana [12], which is instantiated and configured to communicate with the persistence layer and detect and automatically generate dashboards based on the data persisted. This is carried out in a two-step process, where the service is provisioned in terms of instantiating the service and making the connection with the persistence service and the dashboard builder phase where the contents of the database are analysed, and the dashboards are built with certain fields ignored as per a predefined list. This auto generation of dashboards is a key feature of the visualisation service as it provides basic visualisation of the service inputs and outputs without intervention of the system operator. On top of the auto generation, this service allows for the dashboards to be edited, extended, and enhanced if the use case so requires.

3.2.4 edgeFLEX Platform Cloud Architecture

For the first iteration of the edgeFLEX Platform it is necessary to provide a cloud deployment that will enable the running of the trials in a centralised way. This means that the services, the backbone services, and the SLA Monitoring Tool would be deployed centrally, on servers in RWTH or WIT and the edgePMUs or other trial site devices would send and receive data via the edgeFLEX Data Bus. It is important that this deployment and cloud architecture is done in a secure and modular way so that it can provide adequate flexibility in terms of deploying it in different scenarios, where some of the components, for example the data bus, might already exist at a trial site. The edgeFLEX platform in general designed in a way that all the services can be deployed in isolation so at its core is modularity and with modularity comes the need to have a robust and secure method of communications between the components. This is tested and inbuilt in the edgeFLEX platform cloud architecture, illustrated in Figure 11, where the services are grouped in a way that they can be deployed in Virtual Machines either inside or outside the cloud platform. Each connection to the outward facing services, like the data bus, the SLA Broker, the persistence, and the visualisation services are secured by authentication and encrypted either with SSL or HTTPS.

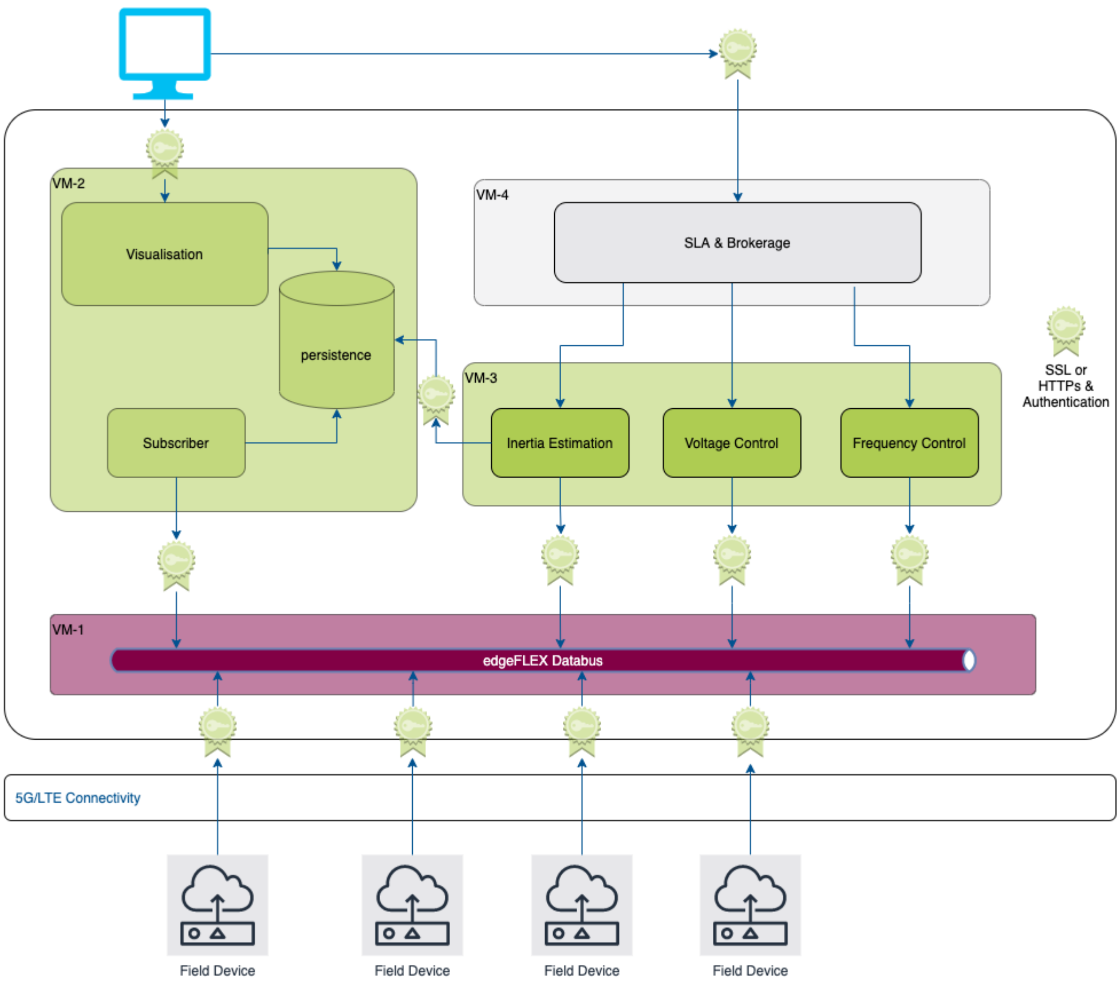


Figure 11 edgeFLEX Platform Cloud Architecture

3.3 edgeFLEX Enablers

3.3.1 Service level Agreement Monitoring Tool

The edgeFLEX SLA monitoring and brokerage solution is intended to be an enabler with the edgeFLEX platform, that allows the Grid Management, VPP Optimisation and Flexibility Trading services of the platform to interact so that the services can be used to realise more sophisticated use cases, which need brokered cross-sector actor support in terms of asset configuration and constraint details, external asset interaction, data acquisition, external platform interaction and flexibility trading.

The SLA monitoring tool sits between the service components of the edgeFLEX platform and the grid management and energy market to facilitate flexibility and fast observability. To enable this functionality, the SLA monitoring tool will interact with several actors: the VPP, the DSO, the edgeFLEX platform ancillary services and the energy market through the GoFLEX platform. Within this interaction, flexibility is handled via semi-standardised FlexOffers through the GoFLEX platform to the DSO, while the control services (e.g., Voltage Control) are hosted on the edgeFLEX platform and act to maintain grid stability in the event of imbalance, enabled by measurement data supplied by edgePMUs deployed on the grid. While the structure and contents of FlexOffer payloads will be finalised in the next phase of the project, samples of such data structures have been provided by INEA based on their solution and will serve as a reference for those used in the edgeFLEX platform. Such FlexOffers are used to report on pre-defined units of energy flexibility and the resulting desired activations of this flexibility The FlexOffer request and responses are in a JavaScript Object Notation (JSON format), providing a more human readable representation and efficient parsing of data [13]. A sample of FlexOffer request and response, including a detailed breakdown of the parameters can be found in annex item 0.

As VPPs will consist of several assets, it may be the case that not all these assets will participate in control enabled by the ancillary services or flexibility, or only up to a certain extent. The market side will require awareness of this level of participation and access to data points from available nodes for flexibility and fast observability, the control services will also need awareness of participation levels of assets in terms of control, finally, the grid side will require information relating to asset ownership of the VPP.

To meet the needs of the market-side, control services and the grid, the SLA monitoring tool will host the essential system data of the VPP, including those pertaining to the assets owned and the level of participation each asset will have in making use of flexibility and ancillary services from the edgeFLEX platform.

The SLA monitoring tool will interact with other systems and services through open MQTT and HTTP REST interfaces that will enable the input and output of requested data and information, such as for the Voltage Control service as described in section 3.4.1

As outlined in the architecture in Figure 12, the SLA monitoring tool will interact via a REST interface to supply requested VPP system configuration data through API endpoints.

The SLA monitoring tool will also provide device configurations and data points to the GoFLEX platform to subscribe to the relevant nodes allowing for awareness of the level of participation assets will have regarding flexibility. Interaction with outside entities such as the GoFLEX platform will be enabled through the IoT Data Platform via an MQTT interface.

To achieve this functionality, the SLA monitoring tool will leverage the concept of *Policy Based Network Management* (PBNM), a methodology that sees all in terms of policies, events and managed objects [14]. Utilising PBNM in the SLA monitoring tool, SLAs will take the form of policies, when an event is sensed by the SLA monitoring tool a decision is made based on the relevant policy/policies defined on the system, finally the decision is enforced or rejected. To enable these interactions, the SLA monitoring tool will consist of several PBNM related software services for storing policies (Policy Repository), defining decisions (Policy Decision Point) and enforcing decisions deduced (Policy Enforcement Point).

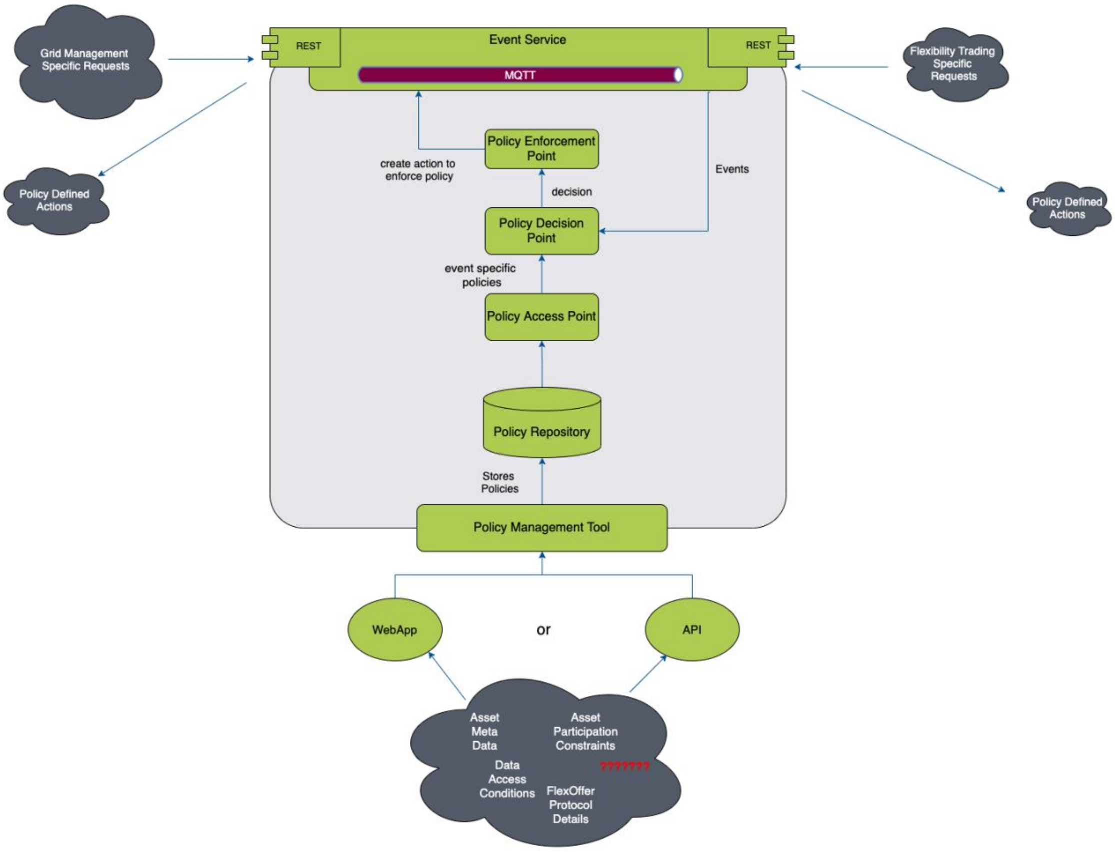


Figure 12 Component Interaction Within SLA Monitoring Tool

In the context of edgeFLEX and energy flexibility, policies will be defined based on the needs of the FMAN (Flexibility Management) and GMAN (Grid Management) using a Graphical User Interface (GUI) and/or API. Such policies may be related to specific locations, events, or assets. The SLA monitoring tool can then receive and enact on the requests related to grid management or flexibility through REST and MQTT interfaces, for example returning asset participation in flexibility to the market. In phase 1 of edgeFLEX, an initial version of the core components of the SLA monitoring tool have been developed and deployed. This early version of the brokerage contains the open interfaces over MQTT and HTTP REST, interaction between each component in terms of retrieving policies. These components have been built using the GoLang language, utilised for its scalability, suite of modules and most importantly its speed, which is of vital importance to the function of the SLA monitoring tool [15]. A graphical user interface has also been implemented via Python/Django, allowing the addition of policies for both FMAN and GMAN, which are stored in a policy repository PostgreSQL database. Figure 13 displays the SLA Monitoring tool GUI for adding policies.

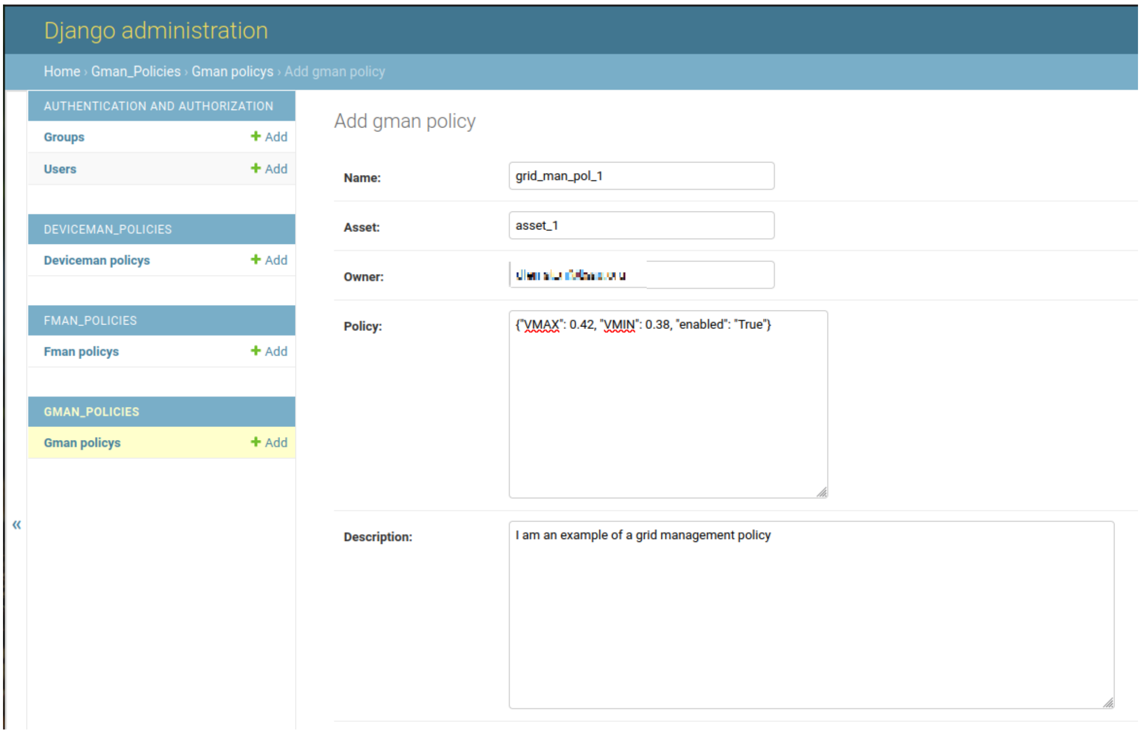


Figure 13 SLA Monitoring Tool GUI

3.3.2 edgePMU

This section will describe the current version of the edgePMU in hardware as well as in software and will put the approach in context of this project.

The general approach of the edgePMU, contrary to classical Phasor Measurement Unit (PMU) design, is to split the data acquisition and the phasor estimation algorithm into two separate tasks and deploy them on separate computational units to gain flexibility for the algorithm. The data is acquired on a low-cost computational device and then transmitted over a reliable and fast communication interface, either wired or wireless, to a cloud/edge-cloud environment where the data processing is performed, and the final phasor is calculated. Because this approach transmits a continuous stream of raw samples higher bandwidth is needed with respect to a normal PMU. In the scope of the edgeFLEX project 5G mobile communication will be used and evaluated. A simplified block diagram can be seen in Figure 14. This figure shows on the left the acquisition unit (field device) and on the right the algorithm running in a docker container (cloud service) which is the intended implementation in the edgeFLEX project.

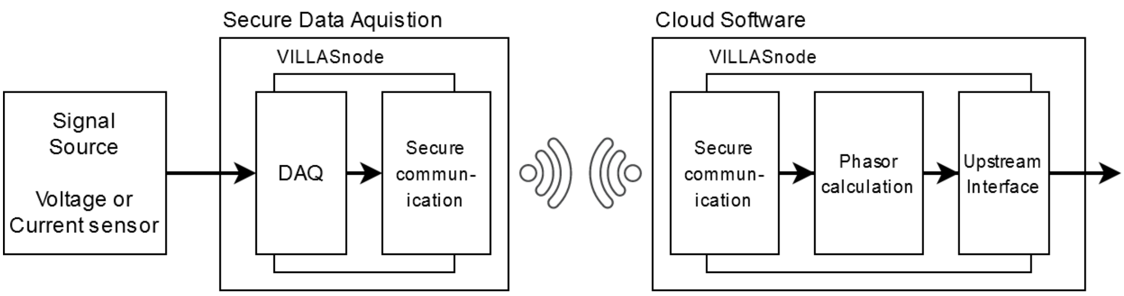


Figure 14 edgePMU basic data flow

The edgePMU is an enabler for other services that are developed in this project. The edgePMU, in fact, provides the edgeFLEX grid management services with measurements of amplitude, frequency, phase and change of rate of frequency (ROCOF) of grid voltages and eventually currents. The edgePMU therefore is the link between real world and service platform. During this project multiple field devices will be deployed and will then provide live data to the platform. Depending on the service needs these field devices either can measure voltage or current on multiple channels.

The current hardware setup consists of a Raspberry PI 3 and a MCC USB-201 [16] USB analogue to digital converter. The hardware is a modified version of the Low-Cost PMU [17] hardware which was developed in the H2020 project SUCCESS [18]. On the cloud side docker containers are used in combination with docker-compose. With the current software and hardware revision the following KPIs can be achieved.

KPI	**Value**
Sample rate per channel	10 kSps
Number of sampled channels	7
Samples per packet	100
Analog resolution	12 Bit
Analog input voltage	±10 V
Transmission protocol	UDP (VILLAS Binary [19])

The field device and the cloud service both run a software called VILLASnode [20]. This software interfaces with the analogue to digital converter and handles the time tagging of the samples. Then the samples are packed in UDP packages and transmitted to the cloud side. The number of samples in one packet can vary and is a trade-off between transmission overhead and payload. It is found that 100 samples per packet are a good trade-off. With the current sample rate this equals 10ms or half a power line cycle (European grid with 50 Hz). The samples are then received by the cloud service and processed in the VILLASnode hook for the Fourier-Transformation. Then the phasor is extracted and forwarded via MQTT. The mechanism for forwarding can be modified to fit the service needs. Currently plain text MQTT is supported. During the project more formats will be implemented to fit the different service needs.

3.3.3 5G API

In this section, an early product prototype of the 5G Application Programming Interface (API) from Ericsson will be introduced, which makes it easier for power system operators and their use cases to use 5G and benefit from its functionalities. In Section 3.3.3.1, the 5G API will be introduced, its key functionalities will be given and the use of the 5G API in the energy sector will be explained. Then, in Section 3.3.3.2, the purpose of using a 5G API in edgeFLEX project will be discussed and the relevant and required functionalities of the 5G API for edgeFLEX use cases will be given.

3.3.3.1 Introduction to the 5G API and its use in the energy sector

5G was designed to support high data rates, high reliability, and low latency, and to connect many devices and objects with each other [21]. Thus, one of the most significant differences between 5G and other wireless communications technologies is that 5G is not only enhancing the mobile broadband, but also providing flexibility and cost efficiency to deploy new services through its capabilities such as edge cloud and virtualization. In addition, 5G can support machine-type communication and Internet of Things (IoT) applications in many verticals in the context of providing connectivity between many devices [22]. One of these verticals that can benefit from 5G is the energy sector.

As the role of 5G is increasing in many verticals, it becomes more and more important to provide a simple way to use 5G communications. For instance, in the energy sector, power system operators have limited to no knowledge about 5G or are not experienced to use 5G in their systems. Therefore, there is a need to provide an API which is straightforward, does not require prior communications knowledge and addresses the requirements of the power system operators [23]. This API is called 5G API which exposes the capabilities of 5G to power grid services.

The 5G API is an interface that manages connectivity for the data communication for power system operators as well as makes the data management easier for them. For example, power system operators will be able to monitor remotely the status of their sensors, actuators, and data flows. The primary role of the 5G API is to manage the user plane of a 5G network that performs the transmission of data between field devices and power grid services (e.g., connections being established, monitored, changed, terminated etc.). Hence, the 5G API can dynamically manage the connections with required transmission characteristics.

Figure 15 illustrates the high-level communication layer architecture of the overall system from both device-side and network-side. The connections in the figure are used for transmitting data between field devices and energy applications via the 5G network.

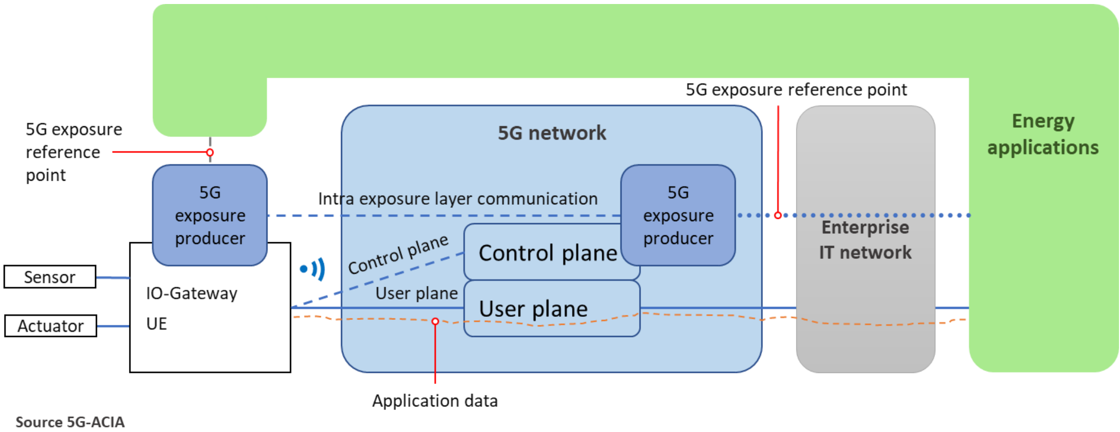


Figure 15 High-level communication layer architecture [10]

Here are the definitions of the components illustrated in Figure 15:

- The **User Equipment (UE),** which provides the connectivity to the 5G network.
- **Sensors and actuators**, which are in the field. Sensors are used to collect data from the grid and actuators are used to run the control commands coming from the energy application. They can either be integrated into the UE or an **Input-Output Gateway** function (IO-Gateway) can connect the external sensors and actuators to the UE.
- The **energy application**, that consumes services of the 5G exposure interfaces and is deployed on a compute and store platform.
- **5G exposure producer**, that implements a service exposed to users of the 5G system.
- **The enterprise IT network,** that is a communications infrastructure on the enterprise premises used for non-real-time resource planning and supervision energy applications.
- The **control plane**, that is employed to dynamically manage the user plane data connections in the 5G system with required transmission characteristics.
- The **user plane**, also called the data plane, that carries the network user traffic.

The 5G API can provide several device management functionalities. Some of the key functionalities are listed below [22]:

- **Device provisioning and onboarding:** The 5G API can support integration and configuration of a device into a 5G system by provisioning the relevant UE information (e.g. UE IDs, user subscriptions) to the 5G network to accept device connection when the device is activated.

- **Device connectivity management:** When the field device is activated, the 5G API can support the provisioning of device-to-service connections with requested quality of service (QoS) characteristics. However, the 5G API can also change the connectivity parameters of the device according to the new requirements. For example, the QoS characteristics can be modified, additional connections can be established, or the connection can be terminated.

- **Device connectivity monitoring:** When the field device is activated and already onboarded to the 5G network, the 5G API periodically monitors the status of the device. In addition, the 5G API can notify the relevant service with the detailed device status and connectivity-related data of the device. Some of these detailed data are the information about established connections of the device, QoS characteristics of the connections, the status of the certificates, current reference signals received power value etc. Furthermore, instead of monitoring the device status periodically, it is also possible for the 5G API to provide event-triggered monitoring. In that case, the 5G API can notify the relevant service if there was a failure or violation occurred.

- **Device group management:** When many field devices are activated, the 5G API can enable the creation, modification, and removal of the device groups. For example, device groups can be created in case the devices would serve for the same purposes or have the same QoS characteristics. Once a device group is created or a device is added/removed from the device group, the 5G API can notify relevant service about the modifications in the device group.

3.3.3.2 The use of the 5G API in edgeFLEX project

In the edgeFLEX project, EDD will further develop and adapt the 5G API prototype, which was implemented and tested with the use cases from other verticals by Ericsson. The objective of bringing a 5G API prototype into edgeFLEX project is to evaluate and improve the usefulness of the 5G API in the energy sector, especially for the use cases defined in edgeFLEX.

As the early product prototype was specifically implemented for the use cases from other verticals, EDD will further develop the 5G API prototype to address and fulfil the requirements of edgeFLEX use cases such as voltage control supported by the edgePMU concept. The 5G API prototype will be tested on 5G mobile network in the laboratory performing the selected edgeFLEX use cases. The details of the 5G API implementation and the functionalities which edgeFLEX use cases require from the 5G API will be reported in D3.1, while the results of the laboratory tests with the 5G API will be reported in D5.5.

In the previous section, a general overview of the 5G API functionalities was given. The 5G API functionalities that are most relevant for the selected edgeFLEX use cases in particular the use cases supported by edgePMU concept were identified. These functionalities are listed below:

- **Detecting loss of communication:** The power system operator needs to monitor the status of their connected devices in the field. Therefore, in case there is a connectivity problem with a field device, which means the device is no longer connected to the 5G network, then the 5G API can inform the power grid service in the edge cloud about the disconnection. The power grid service in the edge cloud can either subscribe to receive this disconnection message, or it can request the information on-demand.

- **Detecting packet loss:** The power system operator needs to monitor the quality and reliability of data transmissions between field devices and the power grid service in the edge cloud. Therefore, in case of a packet loss, the 5G API can inform the power grid service in the edge cloud about the potential QoS violation. The power grid service in the edge cloud can either subscribe to receive this QoS violation message, or it can request the information on-demand.

- **Creating, modifying and removing of device groups:** The power system operator can define device groups belonging to the power grid service. Thus, onboarding and monitoring of multiple devices can be performed easier as device group management. A device can belong to multiple groups concurrently and it can join or leave a group in accordance with, for instance, device location.

- **Monitoring of latency:** The power system operator needs to monitor the latency values of each data transmission between field devices and the power grid service in the edge cloud. Therefore, 5G API can inform the power grid service in the edge cloud about the latency values of the messages.

3.4 Grid Management Services Architecture

3.4.1 Voltage Control

The Voltage control service developed in the edgeFLEX architecture comprehend the algorithm that has been extensively described in D1.2, supported by simulation results. The algorithm receives voltage measurements from the nodes of the grid and calculates control signals for the distributed generators (DGs) if the voltage locally exceeds the limits defined by the grid codes or by the DSO.

3.4.1.1 Architecture

The architecture of the voltage control as service described in Figure 16 is characterised by three components.

- The **voltage control,** a docker container where the algorithm receives the measurements and calculates the power set-points. The algorithm has been implemented in a centralized solution, given that it allows a better integration in the platform. The communication interface for the service is performed by the dmu [24], a tool developed in RWTH internally to facilitate the integration of communication interfaces in python-based codes.
- The **edgeFLEX Backbone**, which contains the interface for the voltage control with the rest of the platform by means of the edgeFLEX Databus. From the Databus the control algorithm receives the measurements of the voltage and of the available power of the DGs. The service also interacts with SLA monitoring tool, described in section 3.3.1 via REST API, interfacing with the Databus.
- The **Network,** including all the communication aspects and interfaces to gather and transmit data to the assets.

The overall service is stored in a git repository, which it gives the possibility to update the grid data using the git platform.

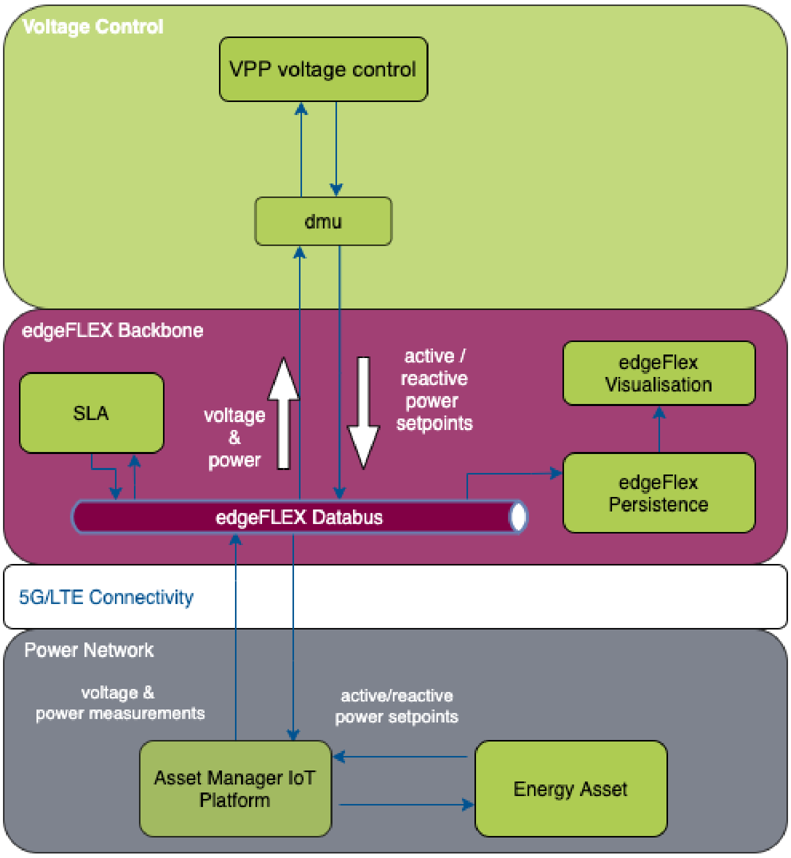


Figure 16 Voltage Control High Level architecture

3.4.1.2 Communications & Interfaces

The main communication interfaces used in the voltage control service are the MQTT and REST API. In the algorithm, these interfaces are realized by means of the dmu [24].

3.4.1.3 Simulation

The simulation of the grid is performed by means of an additional service “edgeFLEX- powerflow service” developed for testing purpose. This service implements a powerflow calculation based on the python-based powerflow solver PYPOWER [25]. The results of each simulation time-step are forwarded to the edgeFLEX data bus via MQTT using the dmu. The powerflow receives at each iteration the result of the voltage control algorithm and based on these values performs the simulation of the grid. The new values of the variables of the grid are sent to the voltage control by means of the edgeFLEX architecture.

The use of the powerflow simulation allows the proper test of the voltage control algorithm, it verifies the ability of the control strategy to counteract undesired voltage violations, and it can be used to eventually test the calculated set-points before applying on the field.

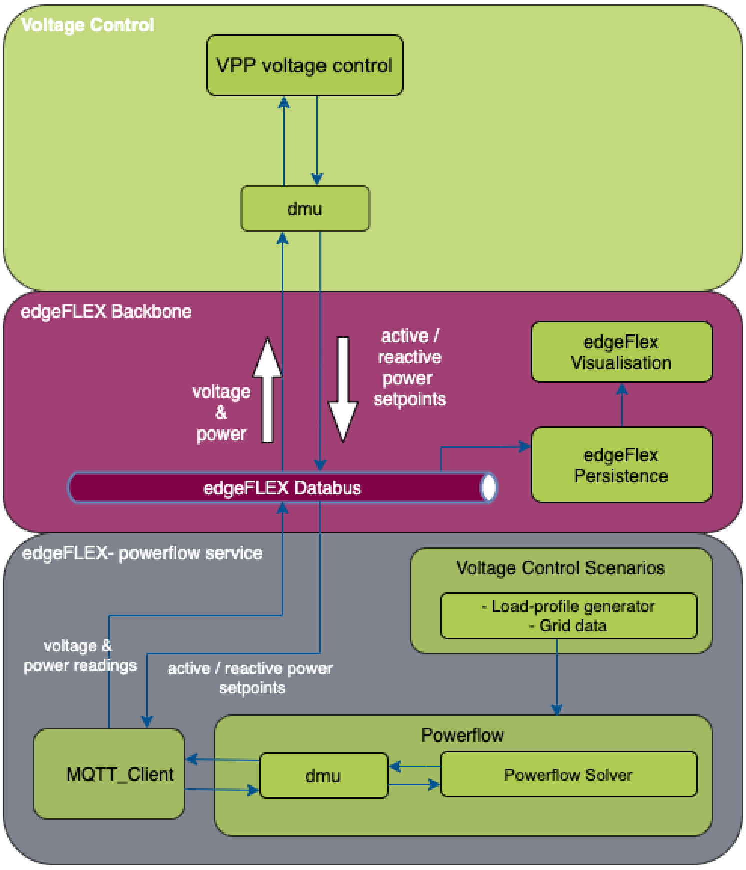


Figure 17 Voltage Control Platform Simulation Mechanism

3.4.2 Inertia Estimation

The inertia estimation service developed within edgeFLEX serves as a monitoring tool for system operators. The inertia estimation is achieved by collecting frequency and active power measurements of the generating units providing inertia in the network and then performing parametric regression based on post-processing of these measurements. The estimation method requires a certain level of variation of the system frequency, which is not provided during normal operating conditions. Hence, the estimation method is meant to be triggered by rescheduling events of generator units and thus it operates in the timescale of tens of minutes up to an hour.

3.4.2.1 Architecture

Similar to the other services within edgeFLEX, the inertia estimation service is integrated to the edgeFLEX platform in a centralised manner. The inertia estimation service within the context of edgeFLEX is composed of three main layers:

- The **Inertia Estimation** service: the dockerised service reads the measurement data stored in the persistence database to be post processed by the estimation algorithm.
- The **edgeFLEX Backbone**: hosts the edgeFLEX persistence service which plays a key role in the inertia estimation service. The live stream measurements received by the edgFLEX platform are stored in the persistence database to facilitate the post processing of the stored data by the inertia estimation algorithm.
- The **Network**: the measurements of the energy assets in the network enter the edgeFLEX platform through the edgeFLEX databus.

Subsequently, the output of the estimation algorithm i.e. the estimated value of inertia is stored in the persistence database.

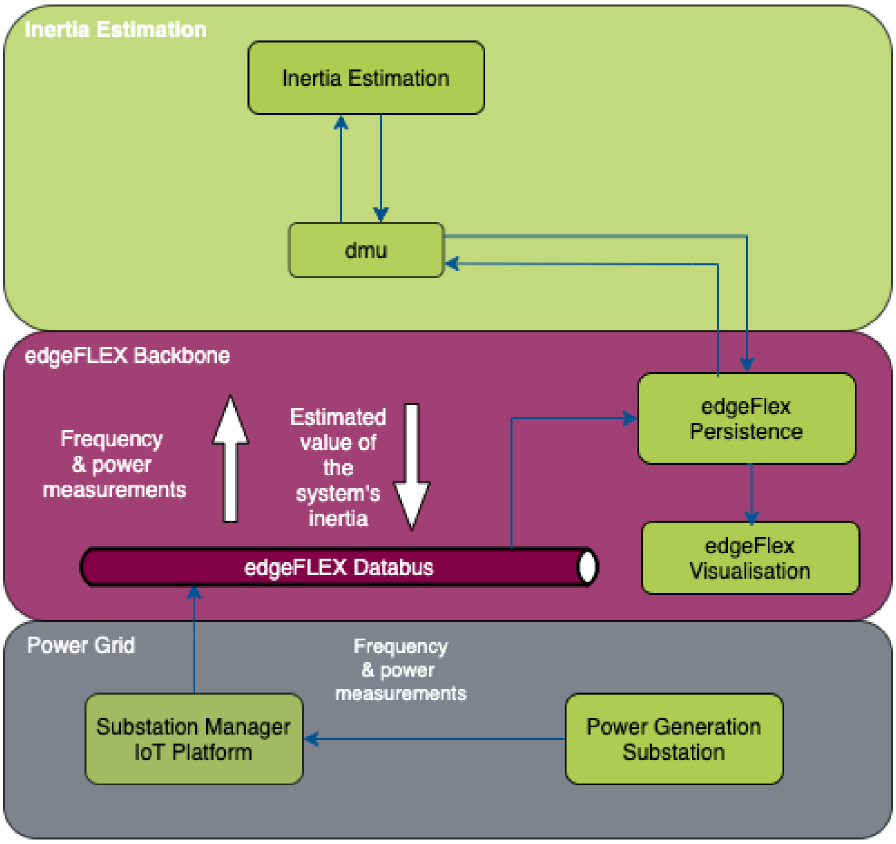


Figure 18 Inertia Estimation High Level Architecture

3.4.2.2 Communications & Interfaces

The MQTT protocol is used to interface the different parts of the inertia estimation service by means of the dmu [24].

It is also worth mentioning, that we used Telegraf [26], a plugin-driven agent, to collect data and store it in the persistence database.

3.4.2.3 Simulation

To validate and test the inertia estimation service implementation before deployment in the field trials, we tested the service integration into the platform through simulation as illustrated in Figure 19.

We used a power systems simulation software to generate simulation measurement data in a CSV file format representing a scenario. Afterwards, by means of an additional service developed for testing purposes, the CSV file data was converted to form a payload to be sent via MQTT using the dmu to the edgeFLEX data bus. Moreover, Telegraf was used to collect the data and store it to the persistence database.

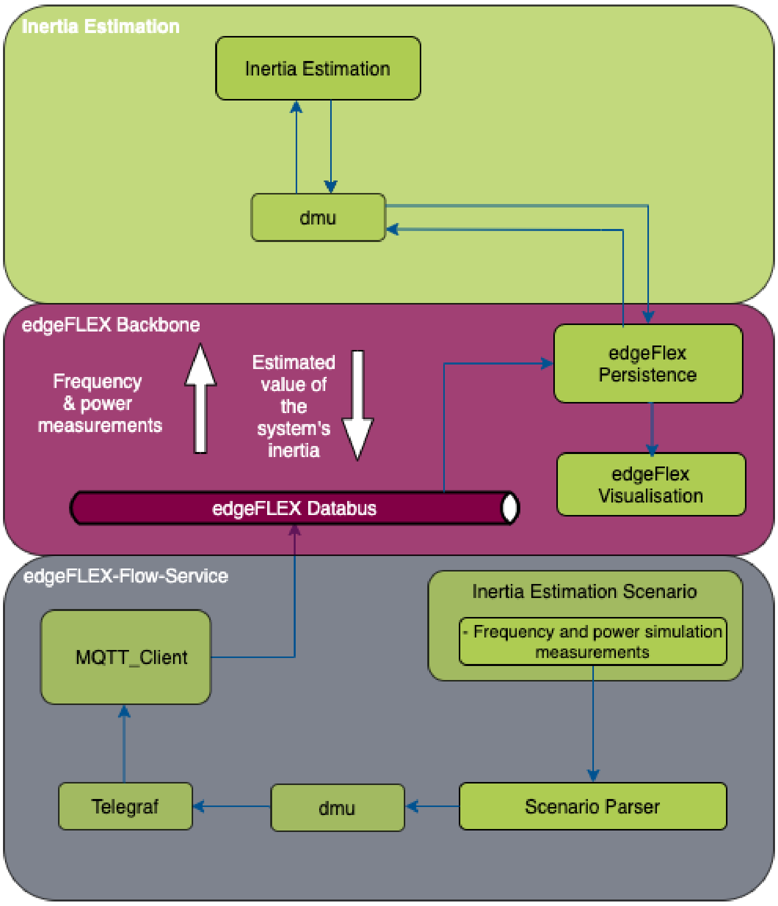


Figure 19 Inertia Estimation Platform Simulation Mechanism

3.4.3 Frequency Control

Frequency Control within the edgeFLEX platform comprises of three algorithms, all performing a different operation for the TSO and VPP.

VPP coordinated frequency control: The controller employs the total active power injected into the transmission grid by the VPP as well as the frequency variation at the point of common connection. It does this to generate a signal that is then transmitted to the distributed energy resources that compose the VPP. The role of this signal is to amplify the sensitivity of fast frequency control with respect to the local frequency deviation by a coefficient that is proportional to the power generated by the VPP.

VPP automatic generation control: The controller employs the reference power signal sent by the Transmission System Operator (TSO) to the VPP as well as the measured active power of the distributed energy resources included in the VPP. It does this to generate a signal which is then used to coordinate the frequency control loops of the VPP resources proportionally to their droops. The purpose of the control is secondary frequency regulation and thus it operates in the time scale of tens of seconds up to tens of minutes.

Frequency regulation metering: The algorithm implements a practical criterion to distinguish between devices that, in transient conditions, modify the frequency from those who do not. To test the frequency regulation of a grid-connected device, the measurement of the frequencies at the neighbouring buses of the device to be monitored the admittances as well that connect the device to the neighbouring buses are employed. With this information, the criterion assesses the frequency regulation of the device based on its corresponding Rate-of-Change-of-Power (RoCoP).

At the early stage of Phase 1 we gathered the requirements for this service and these requirements fed into how we integrate the service with the platform, what interfaces it needed and what we needed to do to simulate it so that we could mimic, as close as possible, a trial site deployment.

3.4.3.1 Architecture

From an architectural perspective it was identified that the algorithms could be deployed in two separate scenarios, one where the algorithms would be tightly coupled to or installed on the controller of the energy asset and another that would be a more centralised implementation where the energy asset or the IoT framework controlling the asset would interact with the algorithm. In the context of the first phase of the Frequency Control and its integration with the edgeFLEX platform we considered the centralised approach as it allowed us to build the software to host the algorithm and interactions with the backbone services. I also facilitated the testing of the software implementation of the algorithms without the need to interact with the trial site assets, which is not the goal of the work in Phase 1. Figure 20 shows the Frequency Control Algorithms in the context of the edgeFLEX Platform. The three algorithms are hosted in the same service and energy asset telemetry, which enters the platform via the data bus, routed to the relevant algorithm.

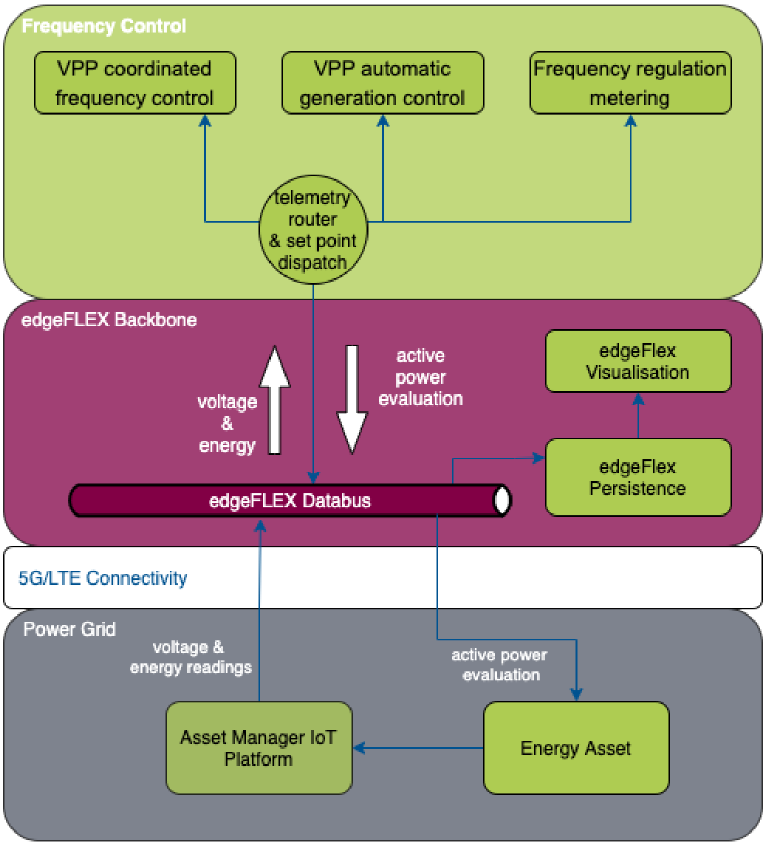


Figure 20 Frequency Control High Level architecture

3.4.3.2 Communications & Interfaces

As stated in section 3.4.3.1 the telemetry enters the system via the data bus and the readings are routed to the relevant algorithm using a key word in the payload. The payloads (detailed in Annex Item A.1.3.3) contain a keyword that matches the intended algorithm for the payload and a scenario that denotes the system the algorithm is being applied for. In Figure 20 there is a component illustrated called **telemetry router and set point dispatch**, and this component takes its configuration at the time of deployment and based on that configuration and the keywords supplied in the telemetry routes the telemetry and the setpoints generated to the required destination. A flow of this interaction as detailed in Figure 21 and it shows how the configurations that are supplied at the time of deployment drive, firstly the connection to the data bus and the topics that it should subscribe to but then the destination of the set point in terms of where it should be sent to. This is designed with portability in mind and is aiming to ensure that the service can be deployed in a wider context than that detailed in the edgeFLEX cloud deployment in Figure 11.

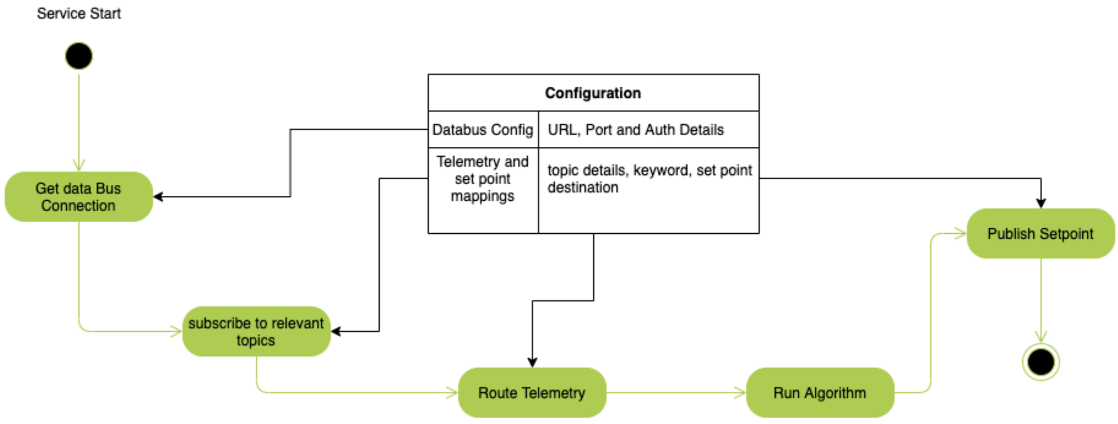


Figure 21 Frequency Control Algorithm example execution flow

3.4.3.3 Simulation

To enable the integration of the Frequency Control algorithms into the platform in a way that is similar in terms of the data that will be sent to it and the other edgeFLEX service interactions it was necessary to build a service to simulate a field asset or field IoT platform. This was carried out using a service that would ingest test data (provided by the partners in WP2 responsible for the Frequency Control algorithms) and a descriptor of the data which is interpreted to form a payload that is sent based on time intervals in the data to the Data Bus via a dispatch mechanism. Figure 22 illustrates the setup of the platform simulation and a key point to note is the similarity to the architecture in Figure 20 where the scenarios and their parser is mimicking the energy asset and the MQTT_client element is mimicking the Asset manager IoT Platform.

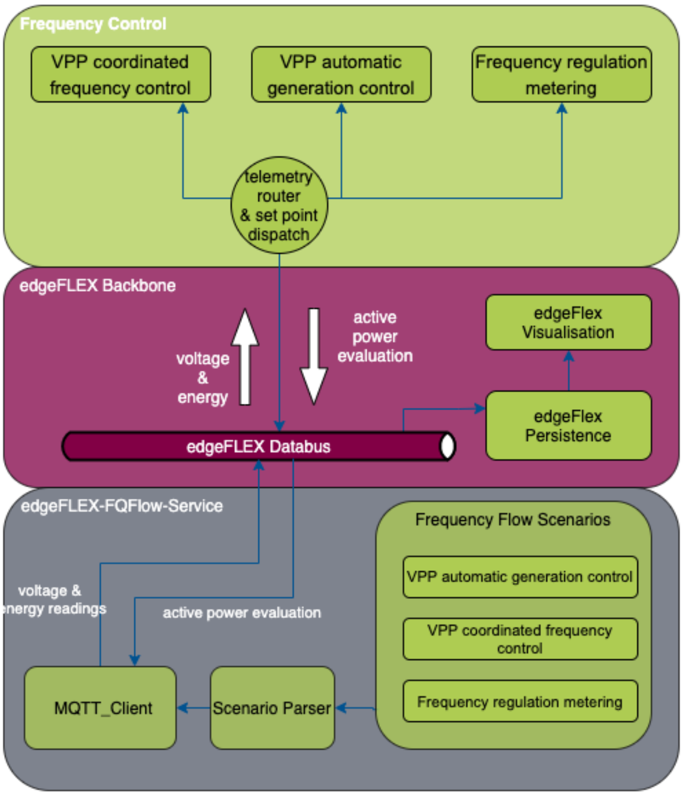


Figure 22 Frequency Control Platform Simulation Mechanism

The scenario is composed of a CSV file like the one in Figure 23 and a descriptor file that allows the scenario interpreter to iterate the data file and from that compose the payload and time the sending of it to match the timing of the simulation. This approach will allow potential new scenarios to be ran in a software-based simulation prior to the instantiation of the service in a real trial site with just the acquisition of a sample of the data and a definition of the fields.

```
1.0025,-0.236074,1.00006
1.0057,-0.097499,1.000138
1.01,-0.059319,1.000244
1.0155,-0.047221,1.000384
1.0226,-0.041106,1.000567
1.0319,-0.057811,1.000803
1.044,-0.098672,1.001103
1.0596,-0.15111,1.001482
1.0796,-0.187404,1.001951
1.0996,-0.193683,1.002415
```

```
{
    "1":{
        "name": "time",
        "measurand": "time"
    },
    "2":{
        "name": "active_power",
        "measurand": "active_power"
    },
    "3":{
        "name": "frequency_meas",
        "measurand": "frequency_meas"
    }
}
```

Figure 23 Frequency Control Simulation Scenario Example

3.5 Virtual Power Plant Optimisation Service Architecture

One of the goals of the edgeFLEX project is to increase the efficiency of VPP operations through optimisation. The aim of VPP optimisation is to prove that the economic outcome if jointly managing intermittent and flexible assets which form a diversified VPP, is improved.

This will be demonstrated using an optimisation algorithm and related software services that will produce signals of action on the market for flexible assets, given as entry data on the market status and predictive status, jointly with the production status and predictions of the intermittent assets. The signals produced by the optimisation service will consist of actions to buy or sell electricity from the assets or to be stored within the flexible part of the VPP.

There are several flows of data to be collected to serve as input for the optimisation algorithms, which are detailed below with the method of collection.

- Production data – Data from assets in the field located in Germany, this data is obtained in real-time by established channels
- Intermittent energies production forecast data – Data resulting from either a power prediction neural network tool developed in Task 3.5 or the Energymeteo [27] data gathered through commercial APIs
- Price data – Day ahead and intraday prices obtained from the European Energy Exchange (EEX) [28]
- Price forecast data – Retrieved from Thomson Reuters [29].
- Alpiq's trading floor trading results on the assets of the VPP – Retrieved from Alpiq's IT systems.

In order to support the VPP optimization algorithm, if the intermittent energies production forecast data is provided within edgeFLEX, then this is another service that needs the additional input data:

- Weather forecasts
- Weather measurement data
- Production forecasts from intermittent production units around the considered site

The optimisation will work in two parts, the aggregation of data; where data from DERs, weather forecasts, system and market state and market signals (information concerning product quality, reputation or intentions of other stakeholders) as input. This data is processed to be used for optimisation however due to the amount of data involved, a simplified model will be used while ensuring an acceptable amount of precision.

Once the data has been aggregated, an ex-post analysis is conducted to determine the effectiveness of the algorithm (based on speed and projected profitability of optimisation).

The next stage of the process is to dis-aggregate the data and use it to generate control actions for the VPP, this heavy workload will use distributed processing involving several layers of parallelisation. After the control actions another ex-post analysis is conducted to determine their effectiveness. Utilising ex-post analysis ensures the speed of the optimisation operation while ensuring it can be iterated an improved after each operation based on the effectiveness of the outputs generated. A detailed view of the optimisation algorithm service and interaction architecture can be found in Annex A.3.

In the context of the edgeFLEX platform, the VPP optimisation solution will interact with the edgeFLEX backbone services to retrieve aggregated data through the edgeFLEX data bus from DERs and weather forecasting sources and store aggregated data for the optimisation algorithm with the edgeFLEX persistence service, the interactions are described in Figure 24.

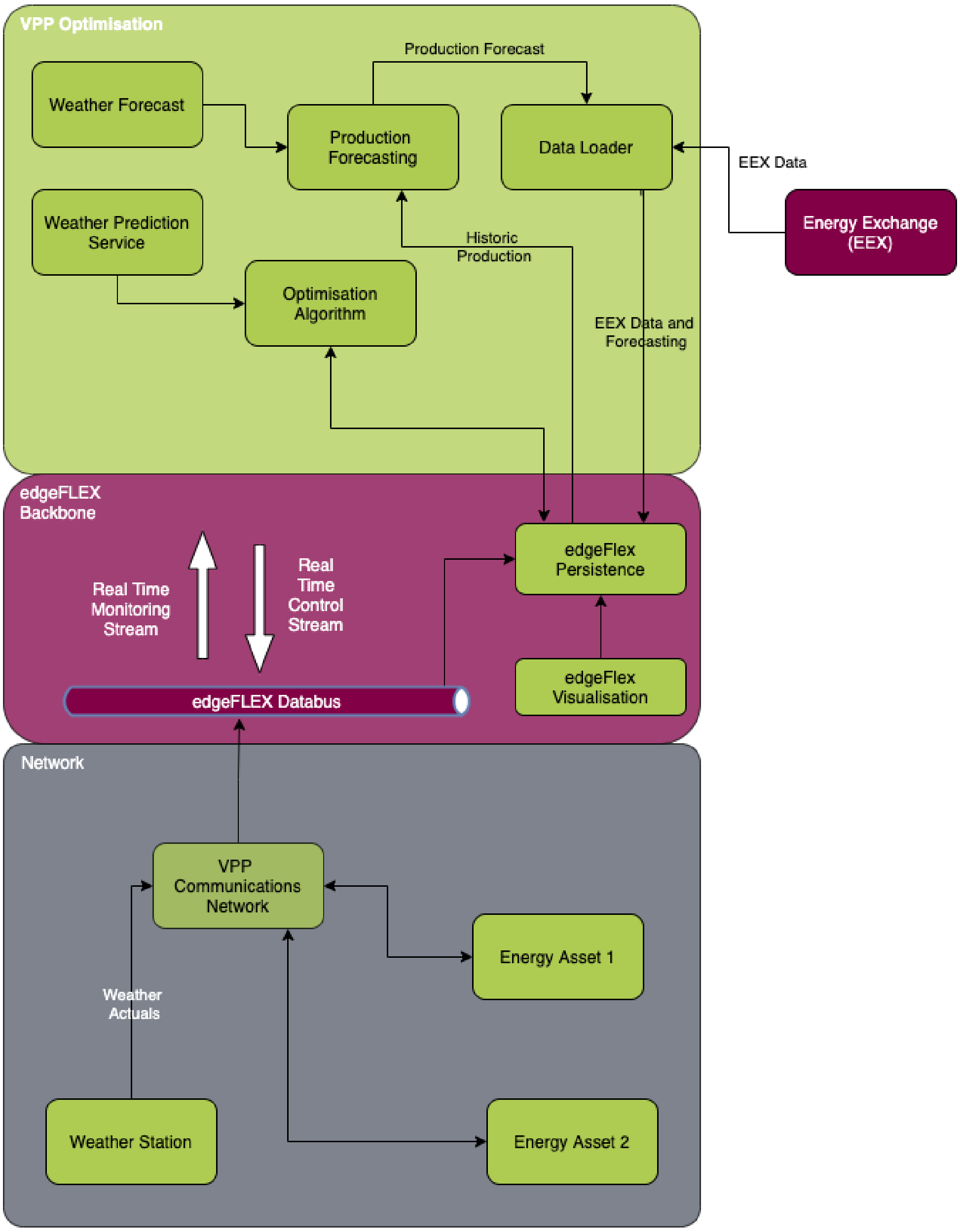


Figure 24 VPP optimisation & edgeFLEX backbone services

Through the next phase of edgeFLEX project this work will focus on the completion of the field trial through the following steps:

- Plug of the algorithms on the real-time data flow
- Incorporation of the characteristics of battery from SWW, installed in Germany
- Create a way to retrieve Alpiq's trading floor trading results on the assets of the VPP
- Create a comparison tool to assess the results of the optimized VPP management compared to the actual Alpiq's management
- Create the interface with trading team for delivering the trade signals to be passed on the trading tools (manually)
- Complete 10 trades according to the optimized strategy, live on the assets of the VPP

3.6 Flexibility Trading Architecture

3.6.1 Architecture of the generic system

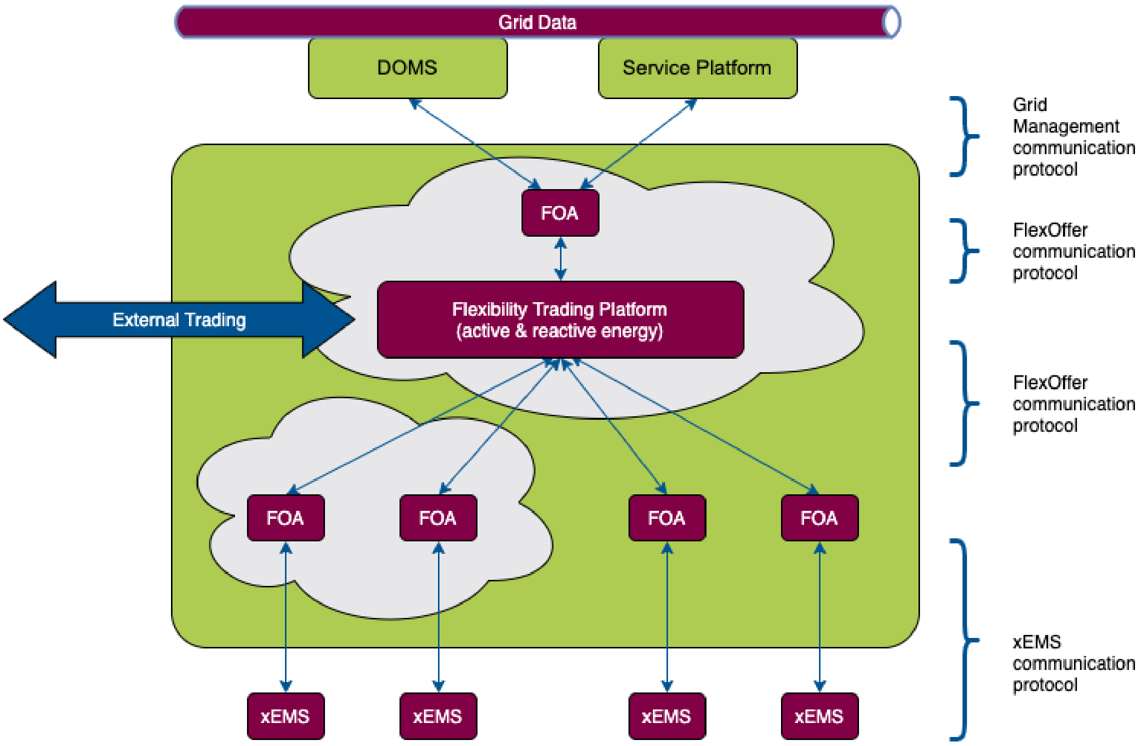


Figure 25 Flexibility trading system architecture

The Flexibility trading system consists of the Virtual Power System (VPS), the prosumers' energy management systems (xEMS) controlling the prosumer's devices (loads and production units) on the offering side, and the Grid systems (Service Platform and DOMS – Distribution Observability and Management System) on the requesting side. The xEMS system is located at the prosumer location. The task of xEMS is regular calculation of the adaptation and reporting it to the FlexOfferAgent (FOA) in the prescribed form. At the reception of the demand, it executes the received schedule by control of the device.

Architecture of the VPS is a two-level system consisting of:

1. FOA (enables connection of xEMS and of Grid Systems) and
2. Flexibility Trading Platform.

The FOA assigns the cost to the received adaptation capacity before forwarding the information to the Flexibility Trading Platform in the form of FlexOffer (FO) and aggregates its received schedules before sending them to the prosumers' xEMSs. The FOA is installed either in cloud form or with physical HW at prosumer location. The user entity (grid operator, organized market,) establishes the communication with the VPS via dedicated (cloud) FOA. Additionally, the Flexibility trading system offers possibility for development of new open interfaces for connection to other trading entities (cluster VPS).

The Flexibility Trading Platform is a cloud application responsible for management of FOs, reception of external requirements and generation of operating schedules.

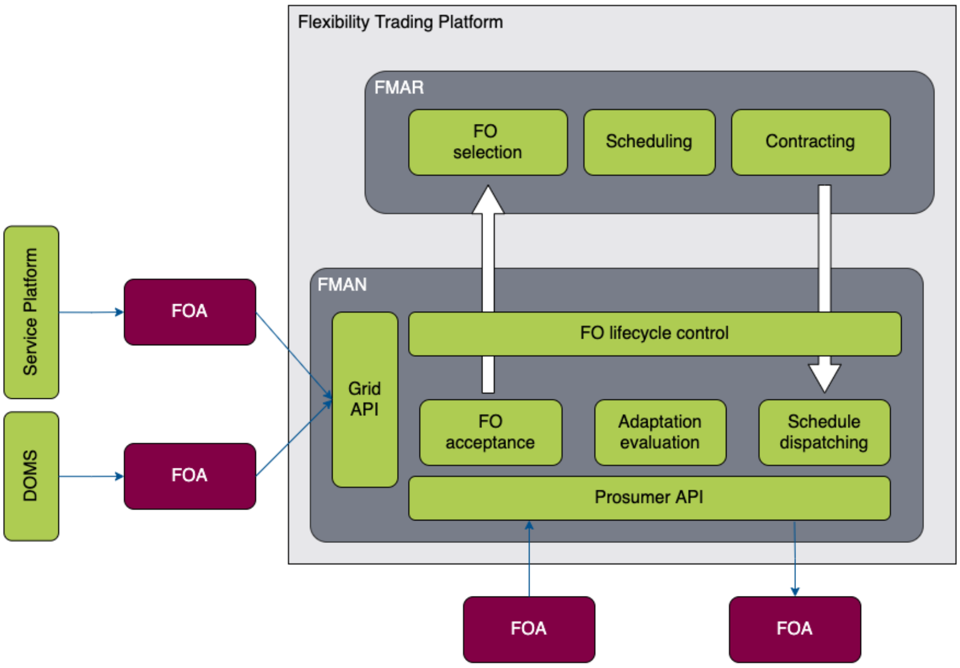


Figure 26 VPS components and processes

The KIBERnet control centre component is responsible for collecting, managing and trading of flexibilities (FO) from participating prosumers. The tasks are shared among flexibility manager (FMAN) component and flexibility market (FMAR) component:

- FlexOffer Manager (FMAN) – allows managing (potentially large) collections of flexible devices in the FO form. It performs the FO acceptance, schedules delivery and executes overall FO lifecycle control. It also provides the side processes such as adaptation evaluation, prosumer reliability and other KPI calculations.
- FlexOffer Market (FMAR) – offers various flexibility trading options, potentially for use by Balance Responsible Parties, micro-grids, and third-party flexibility marketers. It provides functionalities of single-, two-sided pool trading, or many-to-many trading, suitable for different flexibility trading scenarios of demand-supply balancing, congestion management, etc.

The integrated system is provided with a graphical user interface (GUI) that allows Aggregator/micro-grid users/operators to observe and configure flexibility trading processes.

3.6.1.1 FMAN component

The KIBERnet flexibility manager (FMAN) is a software system for handling the FO and contracts between prosumers and the flexibility manager user (e.g., micro-grid or an aggregator). Specifically, it handles the following tasks:

- Acceptance of FlexOffers, which involves checking FO consistency and archiving them. The FOs may be received from FOA or via another FMAN. The acceptance functionality is implemented based on a RESTful HTTP API interface available for external calls from FOAs.
- Maintenance of FlexOffer pool takes the control over the FO life cycle. It rejects unused, improperly formed, and removes the expired and scheduled FOs. FOs are grouped according to the status and location. This functionality is implemented as an internal FMAN service.
- Management of the resulted schedules/contracts includes the reception of schedules from flexibility market (FMAR) and their delivery to the prosumers. The schedule reception is implemented using an HTTP API interface available for internal calls from

FMAR. The delivery to FOA is implemented using an HTTP API interface available for external calls from FOAs.

- The evaluation of the executed prosumer's adaptation is based on the characteristic consumption. The control centre measures the prosumer's total consumption. Based on past measurements the prosumer's most probable consumption for the time period of the intervention is calculated and compared with the measurement. The difference is taken as the adaptation realization.

One of the FMAR tasks is also calculation of the FO energy transfer costs and inclusion of their influence on the FO selection, which is planned for implementation in the final product.

The FMAN also supports GUI for presenting offered flexibilities and assigned schedules. The presented data are hierarchically organized, which enables the geographical filtering of offers/schedules from single prosumer to aggregated information on the area of interest.

3.6.1.2 FMAR component

The KIBERnet flexibility market (FMAR) matches the production and consumption FOs and calculates financial flows of the assigned FOs. FMAR handles the following tasks:

- Preparation and selection of the proper FlexOffers from the pool and triggering the matching process to provide the assignments. The selection of FOs is based on the location of their issuers and corresponding capability to provide a solution for the localized grid problem (i.e., congestion area). The information about the grid location is provided from DSO's FOA in the form of FO.
- Management concurrent interventions. The FMAR component contains a dispatcher to manage several concurrent adaptation requests. The adaptations need to be geographically separated – i.e., adaptations in two sub-areas.
- Matching of production and consumption FlexOffers, leading to the contracts with assigned energy schedules and financial remunerations. Received adaptation offers are aggregated according to the price. Before the selection process the offers are sorted into the price-volume diagram. The entered user's imbalance information provides the requested volume and estimated cut-off price, which triggers the selection algorithm. Among several schedules matching FO constraints, the advanced optimisation algorithm searches for one with larger financial income. The reduction of imbalances is achieved indirectly by setting the proper price to the positive and negative imbalance. The task is handled by an internal process, which is triggered when a proper set of FOs is prepared in the FO selection process.
- Delivery of the assigned schedules to FMAN, which is triggered when the matching process finishes, and the assignments of FOs become available. It invokes FMAN for schedule storage and further processing.

3.6.1.3 Active/reactive power congestion avoidance and balancing

The active and reactive power congestion scenarios are provided by the Voltage control component for the time horizon of several hours on the 1 min granularity and transported to the SLA Broker component as described in section 4.2.

The congestion scenario provides the information about energy amounts with prices and grid limit from the grid assets to the VPS in the form of time series. The time series provides the following information:

- Predicted energy amount, which exceeds the grid limit (energy flow during peak). This information is needed by VPS to assign corresponding internal sources and reduce the exceeds.
- Free capacity between predicted energy and grid limit during off-peak period. This information is needed by VPS to prepare the prosumers for peak operation and achieve better performance, and to control the rebound effect.

- The price for intervention. The price is separated for production and consumption.

The information about active and reactive energy request is transferred separately. On the other hand, the prosumers send their adaptation capacity offer reflecting their load capability. It may be active or partial active and partial reactive. In the latter case they are encouraged to describe their adaptation capacity in the form of apparent energy. Before matching the VPS shall transform the prosumer FlexOffer into the proper unit. The reverse operation shall be provided after matching and before sending the matched schedules to the prosumers.

3.6.1.4 DOMS and Service Platform

Distribution Observability and management system uses the flexibility offered by the prosumers to avoid congestion or balance the energy flows on the grid. To do this, it issues the buying requests, which are shaped into FlexOffers in FlexOffer Agent for buying. The DOMS component and therefore also the buying FlexOffers shall predict and trade both active and reactive energy. The function of FOA in edgeFLEX will be shared between SLA Broker and FOA module of the Flexibility trading platform.

DOMS functionality and architecture is part of the grid management system (DMS). From the viewpoint of Flexibility trading platform, in edgeFLEX the DOMS is a “black box” system, similarly to xEMS systems on the prosumers’ side.

3.6.1.5 Energy management systems

Energy management systems of different categories (with acronym xEMS, x denoting different categories of Energy management systems) are located at consumers, producers and prosumers (in edgeFLEX, all collectively referred to as “prosumers”) and extract flexibilities from either virtual or explicit energy reservoirs in different processes. From the viewpoint of Flexibility trading system, the xEMS is a “black box “system, similarly to DOMS on the DSO side. Communications & Interfaces

3.6.2 Communications & Interfaces

3.6.2.1 FlexOffer

Communication module is responsible for data exchange with FOA and VPS in both directions. This is how the offered and assigned FlexOffers are transmitted between devices. Exchanging data between VPS and FOA is implemented with HTTP RESTful communication. Since this data can be considered private, the communication must be secure. Use of VPN or another encryption method (HTTPS) is recommended.

The FOA acts as a client and VPS has a server role. Messages are written in JSON form.

3.6.2.2 FOA

FOA comes in two (2) different variations:

1. HW + SW solution (whole system, including server runs locally on EMS site),
2. SW solution (server runs in the cloud).

3.6.2.2.1 Communication protocol

For communication we are using HTTP (REST) protocol. We use HTTP inside local network and HTTPS for communication over outside networks (internet). In the document we mostly use notation HTTP, but keep in mind that both approaches are possible, either HTTPS or HTTP, depends on your requirements.

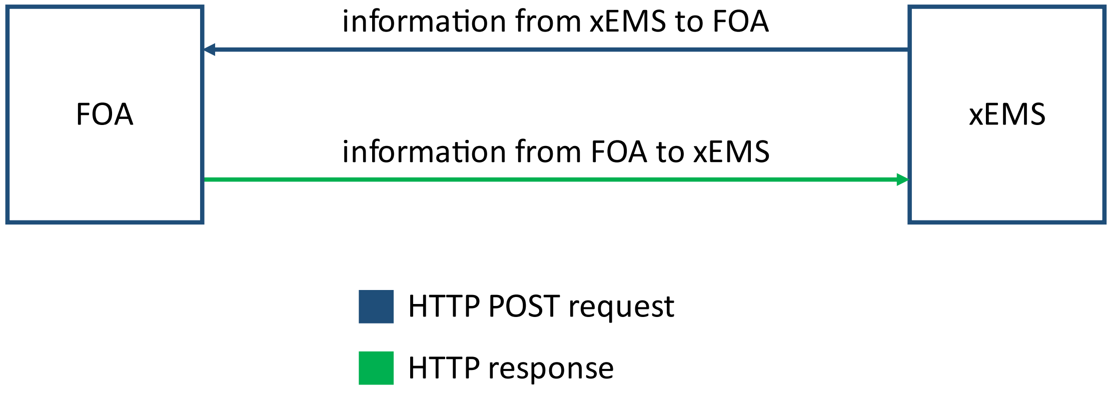


Figure 27 Communication between EMS and FOA

Communication interface consists of HTTP API, which resides on FOA, and HTTP client, which resides on EMS. EMS is the initiator of communication, which periodically, at appropriate intervals, sends HTTP Post Request to HTTP API on FOA. The message contains the relevant information for the FOA packed as payload body of the Request.

In order to get any data from FOA, EMS first needs to send HTTP POST method to FOA, and the required information is in the response message as a packed payload body in the Response.

To achieve adequate response times, the message frequency shall be as high as possible.

Each Post request contains the JSON payload with the following elements:

- Common Header
- Information packets:
  - OperationInfo
  - MeterReadings
  - FlexibilityData
  - DemandSchedule (rejection case only)

The information packets (if several) are packed into single request sent by EMS.

Each Request induces a Response, which contains JSON payload with the following elements:

- Common Header
- Information packets:
  - DemandSchedule
  - FlexibilityData (rejection case only)

The information packets (if several) are packed into a single response from FOA in the form of payload.

4. Virtual Power Plant, Market and Grid Management Unbundled Architecture

4.1 Observability Mechanisms

The objectives of the well managed grid system as addressed within the scope of edgeFLEX project are roughly to maintain quality of supplied energy, avoid congestion and balance the grid.

The tools and services to be developed to this end reside on one hand within grid management system and on the other within the energy management systems of the commercial players – prosumers. In partitioning them between the two segments, the concept of unbundling is followed. The classification is described in section 2.2 of this deliverable.

The categories that can be acted on to attain the objectives need a wide range of functionalities of the distribution grid observability system as part the distribution grid management system. And as an example, that is further developed in the selected combined use case:

- For congestion avoidance and grid balancing through active energy: the main contribution is by trading flexibilities on the market. For this, prediction of operational states and time-critical transients on the grid are of utmost importance
- For controlling voltage excursions, voltage control by real-time grid management system is the primary approach and trading of flexibilities through reactive energy is emerging technology.

For mitigating the observed and predicted transients (congestion avoidance and grid balancing issues), both active and reactive energy flexibility trading need close to real time forecasting of short- and medium-term prediction. On the other hand, the tools and services in grid management system depend on methodologies with which fast close-to-real time response to measured state is possible.

If the functionalities needed to cover both categories are contained in one tools & service package, its complexity outweighs the targeted impact. This becomes even more apparent if extended to other categories of tools and services under development, inertia and frequency control.

The approach taken in flexibility trading in edgeFLEX is to structure the distribution observability and management system (DOMS) into required number of buyers (roles) for different categories. The buyers can act in parallel or combine in one combined buyer, depending on the functionality and scope of services needed. Furthermore, in this approach, adding new categories to flexibility trading is facilitated.

The architecture and functionality at this stage will treat the DOMS for trading the reactive energy flexibilities independently from the DOMS for active energy flexibility trading.

4.2 Voltage Control – FlexOffer Use Case

The use case addresses the mitigation of voltage excursions on the grid by engaging the voltage control functionality of the grid management system and the flexibilities in reactive energy acquired on the flexibility market. The case is unbundled:

- Grid management system engages assets on the grid
- Flexibility trading: assets of the prosumers are engaged and contracted through the Flexibility Trading Platform

The architecture of the combined use case is depicted in Figure 28.

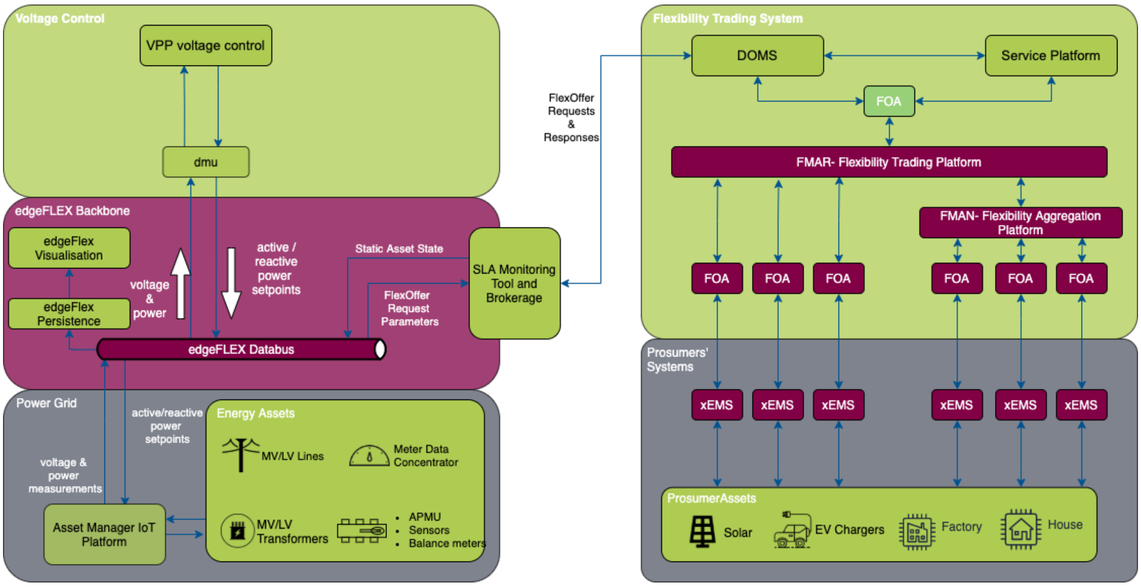


Figure 28 Conceptual architecture Voltage control – Flexibility trading Use case

The figure represents generic architecture case. In actual case to be demonstrated, the architecture will be tailored to both the existing xEMSes in GOFLEX system infrastructure to be engaged, and the scope of the Distribution Management System to be deployed.

The building blocks engaged and the interfaces between them are described in Section 3.4.1 for voltage control and in Section 3.6 for flexibility trading, the interface between the two segments is through SLA Broker, described in section 3.3.1, and the FlexOffer protocol described in Section 3.6.2.1.

The FO protocol has been designed to provide for various characteristics of traded flexibilities based on the needs and functionality of the buying DOMS – Distribution Observability and management system. Consequently, the scope of parameters to be exchanged depends on the request – buying offer for reactive energy that will be formulated by SLA Broker. As a minimum, it must define

- How much reactive energy
- When: time and for how long (time interval)
- Where (location)
- Cut-off price

The functionalities of the two systems – voltage control and reactive energy trading act predominantly in different phases of the process and complement each other. This is indicated in Figure 29 positioning their participation in the sequence of processes in electricity trading cycle as defined in Harmonized electricity Market Roles Model [30] (EbiX [31] view).
The voltage control system acts close to real time directly based on measured state of the transient, whereas the flexibilities are traded based on predicted future state of the grid, relative to the time of trading.

The following positions the goals of grid management versus flexibility trading (Note: colour coded cells indicate the positioning of goals of Flexibility trading and of Grid management services into the sequence of processes in the HEMRM (Harmonised Electricity Role Model) as structured by ebIX.)

ebIX processes (HEMRM)	Flexibility Trading (GOFLEX)	Grid Management
Structure	(*)	
Plan	**	

Trade	**	-
Operate	**	**
Measure	*	**
Settle	(*)	(*)
Bill		

Figure 29 Goals of Grid Management versus Flexibility Trading

The Flexibility trading interval for trading of active energy flexibilities is typically 15 min; for trading reactive energy for voltage excursions, the target is to go to minute interval. The voltage control system acting directly on the grid assets is focused on a shorter time horizon, closer to real time. The strategy for combining the available flexibility of the assets on the grid with the flexibility of the prosumers through trading will be defined by the grid management system and will be based on overall optimum principle, taking into consideration how different combinations of the two would better complement each other.

The use case is planned to be tried out within German trial site for fast and slow processes at SWW and will be further specified and implemented in WP5, T3.5. To the extent that the case will be demonstrated in real environment, the existing GOFLEX system infrastructure will be used. The necessary upgrading and additions will include the upgrading of the participating xEMS systems as described in Section 3.6.1.1 and upgrading and additions to the grid management system for voltage control infrastructure to assure the functionality of the distributed observability and management system (DOMS) for reactive energy flexibility trading.

4.3 Simulation

Given that this use case is in the early stages from a development point of view and given that the Voltage Control, the SLA Monitoring Tool and the edgeFLEX backbone services are integral parts of this use case and needed to be developed first we can only outline our plans here to simulate the interactions for this use case prior to it being trialled. Based on this pour plan is to build this simulation and the interactions between all the relevant systems in a stepwise manner with the SLA Monitoring tool providing the interaction between both sides in isolation first and then fully integrate the use case to form an end-to-end implementation in a laboratory trial. This will allow the initial stages of this development to focus on the interaction in a targeted way so that we can define the system policies that will trigger the request of a Flex Offer from the Voltage Control side and from the FlexOffer side allow the system policy to conform with the FlexOffer protocol detailed in the annex item on page 71.

The steps to build out both sides of the interaction are as follows…

FlexOffer Integration:

- Create Sandbox environment to receive FlexOffer
- Decide on the format of the FlexOffer payload and the Flexibility Management SLA (policy)
- Begin sending requests and testing the response
- Create persistence to track the interaction on the SLA Monitoring Tool side

Voltage Control Interaction:

- Build interaction with the SLA Monitoring Tool for the retrieval of policies
- Define Grid Management SLA (policy) and triggering criteria for FlexOffer creation

- Build logic in the Voltage Control service to implement the SLA
- Test the interaction between the Voltage Control and the SLA in terms of requesting the Flexibility Management SLA

Once the steps detailed above are completed the next steps will be to join both systems together via a mutual set of Grid Management and Flexibility Management SLAs. This will form the first steps to test the proposed use case end to end with sand boxed systems on both sides.

5. 5G and the edge cloud's role in the edgeFLEX Platform

5.1 5G Features

5G is a technology that enables wireless connectivity for a very wide range of services and use cases with new characteristics and requirements from various vertical sectors [32]. Some of these vertical sectors are energy, automotive, transport, manufacturing, healthcare, and public safety. 5G helps society to shape a smarter and more sustainable future, as it can support high bandwidth services, mission critical applications and IoT applications.

3GPP (the 3rd Generation Partnership Project), which is the organization responsible for standardization of cellular technologies, specifies that 5G systems should have the following capabilities [33]:

- 5G systems should support **data rates** exceeding 10 Gbps in specific scenarios such as indoor and dense outdoor environments.
- 5G systems should support an application **end-to-end latency** of 1ms or less, in order to support latency-critical applications.
- 5G should also enable connectivity with **ultra-high reliability** and ultra-high availability. For example, some industrial applications might need to guarantee successful packet delivery within 1 ms with a probability as high as 99.9999 percent.
- 5G devices should be available at very **low cost** and with a **battery life** of several years without recharging.

There are two main deployment schemes to deploy and use 5G technologies: Standalone and Non-standalone. In order to provide capabilities, close to those listed above, early 5G systems were offered as "Non-standalone 5G" which means radio networks of both 5G and LTE use the core network of LTE. On the other hand, "Standalone 5G", which are advanced 5G systems, will use the 5G radio network and the 5G core network supporting the full range of new features. They are expected to be available in the global market within the coming few years. These standalone 5G systems will provide new features, such as network slicing and edge cloud, described below, enabling enhanced support for various use cases.

Network Slicing

The network slicing concept makes it possible to divide the physical infrastructure into virtual networks, so-called network slices. Each slice exists isolated from each other and provides individual quality of service. The quality of service can be associated with security, privacy, latency, reliability and so on [34]. In this way, a network slice can be specifically adjusted to the business purpose it shall serve and the requirements it shall fulfil [35]. Network slicing can be realised in various scenarios: in a private network, in a combination of public and private networks or in a dedicated network for a specific service.

Edge Cloud

Edge cloud is the ability to provide computing and storage resources with adequate connectivity at close proximity to the data sources [36]. The main benefits of using edge cloud are low latency, high bandwidth, and trusted computing and storage.

The transport network is one of the major contributors to the end-to-end latency, as it varies depending on regions, distances and transport solutions used. If a service runs closer to the end user and 5G radio, then latencies caused by transport network can be decreased thanks to the use of edge cloud. In addition, edge cloud helps to reduce the load on devices located at the central data centres by hosting the service at the edge [36].

The edge cloud relies on efficient orchestration logic that enables multiple enterprises to develop, deploy and manage their services. Each of these services can have different characteristics and requirements. When a service or a Virtual Network Function (VNF) is required to be deployed on the edge cloud, the deployment decision can be made based on latency requirement, geolocation, throughput requirement or deployment cost.

5.2 Edge cloud and the edgeFLEX Platform

As explained in the previous section, 5G plays an important role in supporting a wide range of services including the services in energy sector. In the edgeFLEX project, 5G systems, especially together with edge cloud, will enable to host the edgeFLEX services and provide the wireless connectivity between energy or VPP assets and edgeFLEX services considering the communications requirements.

In section 3 where we describe the edgeFLEX platform we do so in the context of the trials as they are at present, where the infrastructure and communications available do not facilitate edge cloud deployments yet. In future deployments and large-scale roll outs edge cloud deployments would lever the features of 5G mentioned in section 5.1 and 5.2. Given that the edgeFLEX platform and its components are built in a modular way and driven by configuration it means that all the components for a given solution do not need to be situated on the same system. This approach to designing a system is particularly adaptable to deploying a component or components to the edge cloud. What this means is that the system can harness the distributed computing power and storage capabilities that edge cloud deployments provide. Plus, at the same time, provide the capabilities of having some common functionalities like for example visualisation or centrally held system meta data could be held centrally or at the respective management systems of the asset owners or managers.

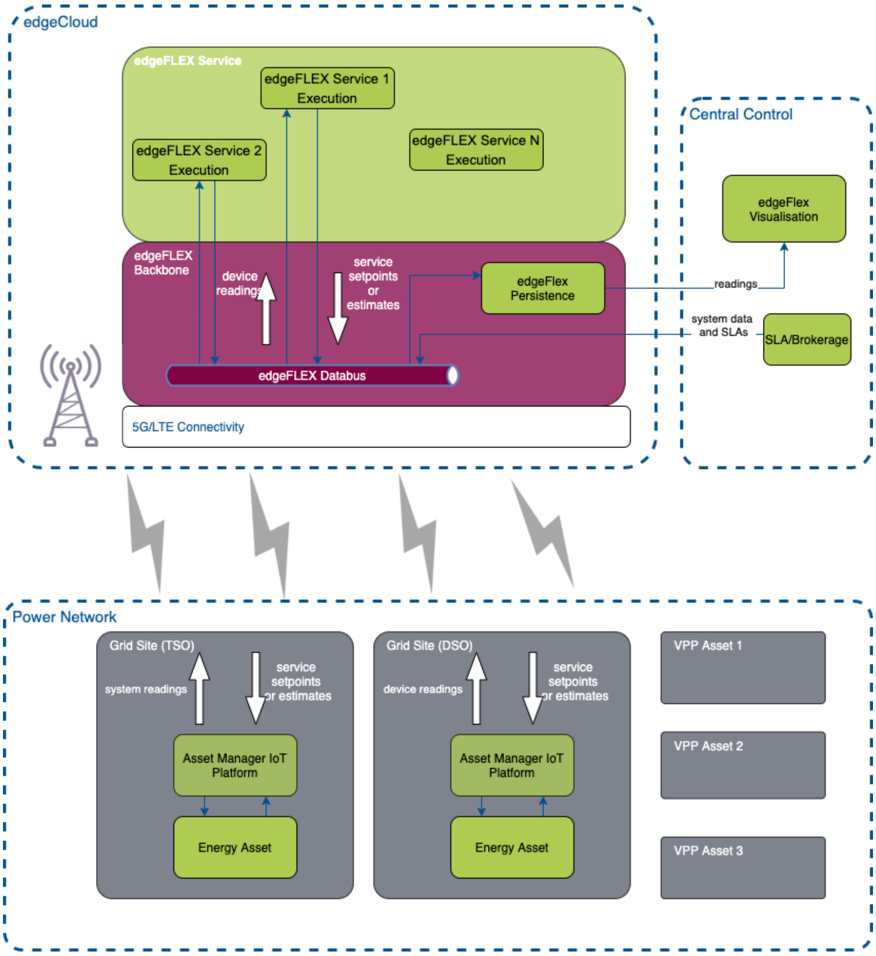


Figure 30 edgeFLEX service deployment example in the edge cloud

In Figure 30 we illustrate a scenario where an edgeFLEX service or services are running on the edge cloud with the visualisation and Brokerage abstracted to a Centrally Controlled. It must be noted that Central Control, in this case, could be either hosted by the DSO, VPP operator or TSO and any cross-actor readings or configurations would be agreed, aggregated, and/or filtered in advance. The edge cloud in this case is depicted as being installed at the base station and the power network entities could be all VPP assets or a mixture of distribution and transmission.

Figure 31 shows how multiple edge cloud instances could potentially from a scheme that would involve only having the necessary components relevant to the control of the site with the relevant data on site and the algorithmic execution close to the asset to ensure that the latency requirements of the service, if relevant, are fulfilled.

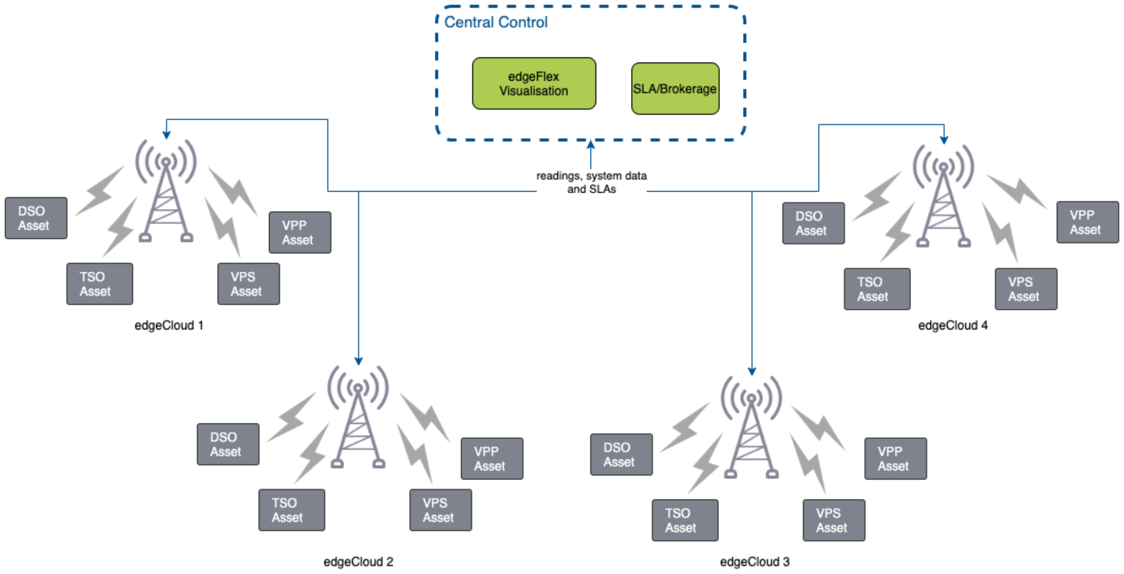


Figure 31 Multiple edge cloud deployment

Conclusion

The goal for Phase 1 of the project from an edgeFLEX platform perspective was to have the requirements gathered, the backbone services identified and the overall picture of the platform, the services and how it can subsume the research and innovation concepts developed with a goal of having them create a meaningful impact when it comes to the trials in WP5. It was found that core to this was an effective requirement gathering process that not only gathered the functional requirements for the services using the surveys, but also using the solution architecture workshops and the identity matrix to determine the linkages between, not only the services but also who they are intended for and why they are needed. These requirements allow us also to take the research from the laboratory and deliver it in a software context with a view to using common software components that can be configured in a relatively dynamic way so as the platform has the adaptability to be deployed in trials with varying architectures and needs. Core to this is the initial simulation of field devices and data providers that allowed for the building of the linkages between the components in a way that will mimic a basic trial site deployment and by doing so prepare the platform for the trial sites in WP5.

Key to the development of this platform to date has been a strong engagement pipeline between the partners in WP1, WP2 and WP3 in terms of giving feedback on the software implementation of their concepts and their timely edits to their packaged code that allowed the integration into the platform to be a fluid process driven by requirements and continuous engagement between the ICT and electrical engineering focused partners of the project. This continuous engagement pipeline allows for the implementation and, therefore, the platform to be built in a way that is cognisant of the needs from an electrical perspective while adhering to best practise from a software and security perspective.

In Month 10 of the project, there was a milestone (MS8) where a subset of the platform was to be delivered and Figure 32 illustrates what was delivered in terms of simulation services, grid management services, enablers, and backbone services. The work on the platform was, however, not only confined to the delivery of this milestone and work has been ongoing in terms of delivering the remainder of the services detailed throughout this report to complete the set of services that will complete the full set of services that constitute the edgeFLEX Platform.

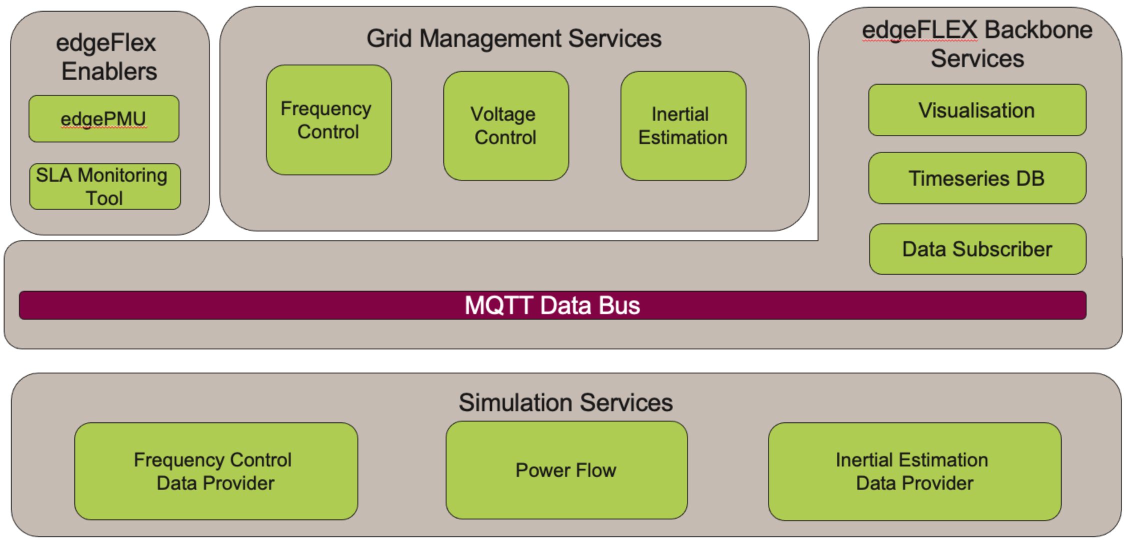


Figure 32 Subset of the edgeFLEX Platform Delevered in MS8

What this means, at this stage of the project is that the architecture of the edgeFLEX platform is very use case focused and in the next stage the architecture will be expanded to provide a more comprehensive architecture with a more prominent emphasis on advancing the VPP using all the services being developed in edgeFLEX and also with 5G playing a more prominent role.

At this stage of the project, however, the platform is described as a Minimum Viable Product (MVP) and one that will require further development to bring it to a production ready state.

Through the edgeFLEX improvement model, where the feedback will be gotten from the trial sites in both the context of the software and the algorithm performance and based on that feedback the services will be refactored and redeployed in the trials for further evaluation.

Activities in Phase 2 of the project will not only look towards the enhancement of the services and the platform but will also aim to look at the role the platform can play beyond the duration of the edgeFLEX project with a particular focus on how edge computing and advancements in 5G can facilitate a more dynamic grid. Given the wide scope of the project and the diverse nature of the services on offer the future work will also look at leveraging the flexibility of the platform and the brokerage tool to realise potential use cases that can be realised in the future with changes than may provide regulatory and market readiness.

6. List of Figures

7. Bibliography

[1] TOGAF. [Online]. Available: https://www.opengroup.org/togaf .

[2] Archimate, “Archimate,” [Online]. Available: https://www.opengroup.org/archimate-forum/archimate-overview .

[3] “Open Group,” [Online]. Available: www.opengroup.org.

[4] C. Richardson, “Microservices Architecture,” 2020. [Online]. Available: https://microservices.io/. [Accessed 02 02 2021].

[5] Docker Inc, “Docker,” 2020. [Online]. Available: https://www.docker.com/. [Accessed 02 02 2021].

[6] Docker Inc, “Overview of Docker Compose,” 2020. [Online]. Available: https://docs.docker.com/compose/. [Accessed 02 02 2021].

[7] GNU, 2020. [Online]. Available: https://www.gnu.org/software/bash/. [Accessed 02 02 2021].

[8] T. S. S. G. -. WIT, “DockerMQTT,” TSSG, 28 10 2018. [Online]. Available: https://github.com/TSSG/DockerMQTT. [Accessed 10 1 2021].

[9] R. A. Light, “Mosquitto: server and client implementation of the MQTT protocol,” *The Journal of Open Source Software,* vol. 2, no. 22, 2017.

[10] InfluxData, “InfluxData,” 2021. [Online]. Available: https://www.influxdata.com/. [Accessed 02 02 2021].

[11] N. Crowley, “Getting Started with Python and InfluxDB,” 2018. [Online]. Available: https://www.influxdata.com/blog/getting-started-python-influxdb/. [Accessed 02 02 2020].

[12] Grafana Labs, “Grafana,” 2021. [Online]. Available: https://grafana.com/. [Accessed 02 02 2021].

[13] M. Soliman, T. Abiodun, T. Hamoud, J. Z. and C. Lung, “Smart Home: Integrating Internet of Things with Web Services and Cloud Computing,” *2013 IEEE 5th International Conference on Cloud Computing Technology and Science,* pp. 317-320, 2013.

[14] G. M. Perez, A. F. G. Skarmeta, S. Zeber, J. Spagnolo and T. Symchych, “Dynamic Policy-Based Network Management for a Secure Coalition Environment,” *IEEE Communications Magazine,* pp. 58-64, 2006.

[15] M. Samaniego and R. Deters, “Hosting Virtual IoT Resources on Edge-Hosts with Blockchain,” in *2016 IEEE International Conference on Computer and Information Technology (CIT),* Nadi, Fiji,, 2016.

[16] MCC Measrurement Computing, “12-Bit Analog and Digital I/O DAQ Devices,” [Online]. Available: https://www.mccdaq.com/PDFs/specs/USB-200-Series-data.pdf. [Accessed 24 March 2021].

[17] A. Angioni, G. Lipari, M. Pau, F. Ponci and A. Monti, “A Low Cost PMU to Monitor Distribution Grids,” in *2017 IEEE International Workshop on Applied Measurements for Power Systems (AMPS),* Liverpool, 2017.

[18] SUCCESS H2020 Project, “success-energy,” [Online]. Available: https://success-energy.eu/. [Accessed 24 March 2021].

[19] Fein Aachen, “VILLASframework: Custom VILLAS Binary,” [Online]. Available: https://villas.fein-aachen.org/doc/node-format-villas-binary.html. [Accessed 24 March 2021].

[20] fein-aachen, “VILLASnode - FEIN Aachen,” [Online]. Available: https://www.fein-aachen.org/projects/villas-node/. [Accessed 24 March 2021].

[21] Ericsson, “5G Explained: What is 5G?,” [Online]. Available: https://www.ericsson.com/en/5g/what-is-5g.

[22] 5GACIA, “Exposure of 5G Capabilities for Connected Industries and Automation Applications,” ZVEI – German Electrical and Electronic Manufacturers’ Association, Frankfurt am Main, 2020.

[23] Ericsson, “Enable innovation with open network exposure,” [Online]. Available: https://www.ericsson.com/en/service-orchestration/network-exposure#. [Accessed 08 February 2021].

[24] “DMU,” [Online]. Available: https://git.rwth-aachen.de/acs/public/automation/modules/dmu.

[25] [Online]. Available: https://pypi.org/project/PYPOWER/.

[26] InfluxData, “https://docs.influxdata.com/telegraf/v1.17/,” [Accessed 19 02 2021]. [Online].

[27] Energy&Meteo Systems, “Solar Power Forecasts | Wind Power Forecasts,” [Online]. Available: https://www.energymeteo.com/products/power_forecasts/wind-solar-power-forecasts.php. [Accessed 2021 March 24].

[28] European Energy Exchange, “Market Data Services | EPEX SPOT,” [Online]. Available: https://www.epexspot.com/en/marketdataservices. [Accessed 24 March 2021].

[29] Thomson Reuters, “Price Forecast Data,” [Online]. Available: https://www.reuters.com/finance/commodities/energy. [Accessed 24 March 2021].

[30] ENTSO-E, “Harmonised Electricity Role Model,” [Online]. Available: https://www.entsoe.eu/digital/cim/role-models/.

[31] ebIX, “Harmonised Electricity Market Model roles,” [Online]. Available: https://www.ebix.org/artikel/role_model.

[32] Ericsson, “5G wireless access: an overview,” [Online]. Available: https://www.ericsson.com/498a10/assets/local/reports-papers/white-papers/whitepaper-5g-wireless-access.pdf. [Accessed 1 March 2021].

[33] 3GPP, “TS 23.251 v15.1.0 Network sharing, Architecture and functional description,” 2018.

[34] Ericsson, “Ericsson Technology Review - Applied Network Slicing Scenarios,” [Online]. Available: https://www.ericsson.com/491069/assets/local/reports-papers/ericsson-technology-review/docs/2021/applied-network-slicing-scenarios-in-5g.pdf. [Accessed 1 March 2021].

[35] Ericsson, “5G RAN Slicing: Capture new business revenues in the 5G era,” [Online]. Available: https://www.ericsson.com/48f7ce/assets/local/networks/documents/5g-ran-slicing-business-review.pdf. [Accessed 1 March 2021].

[36] Ericsson, “Edge computing and 5G,” [Online]. Available: https://www.ericsson.com/49d80b/assets/local/digital-services/trending/edge-computing/edge-computing-5g-report.pdf. [Accessed 1 March 2021].

[37] MQTT, “MQTT: The Standard for IoT Messaging,” [Online]. Available: https://mqtt.org/. [Accessed 02 02 2021].

[38] [Online]. Available: https://git.rwth-aachen.de/acs/public/automation/modules/dmu.

8. List of Abbreviations

ADM	Architecture Domains Methodology
API	Application Programming Interface
CSV	Comma-separated values
DB	Database
DER	Distributed Energy Resource
DG	Distributed Generator
DMS	Data Magament System
DOMS	Distribution Observability and Management System
DSO	Distribution System Operator
EEX	European Energy Exchange
EMS	Energy Management System
ESS	Energy Storage System
FA	FlexAgent
FMAN	Flexibility Management
FMAR	Flexibility Market
FO	FlexOffer
FOA	FlexOffer Agent
GMAN	Grid Management
GUI	Graphical User Interface
HEMRM	Harmonised Electricity Role Model
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICT	Information and Communication Technology
IEEE	Institute of Electrical and Electronics Engineers
KPI	Key Perfomance Indicator
LEC	Local Energy Community
LTE	Long Term Evolution
MBA	Market Balance Area
MQTT	Message Queue Telemetry Transport

MVP	Minimum Viable Product
PBNM	Policy based Network Management
PMU	Phasor Measurement Unit
PV	Photovoltaic
REST	Representational state transfer
ROCOF	Rate of Change of Frequency
ROCOP	Rate of Change of Power
SLA	Service Level Agreement
SSL	Secure Sockets Layer
TOGAF	The Open Group Architecture Framework
TSO	Transmission System Operator
UDP	User Datagram Protocol
UE	User Equipment
VNF	Virtual Network Function
VPN	Virtual Private Network
VPP	Virtual Power Plant
VPS	Virtual Power System
WP	Work Package

ANNEX

A.1 Service Requirements

A.1.1 Voltage Control

A.1.1.1 Service Requirements

Question	Answer	Comments
What input variables will the service need to function?	• Voltage measurements • Active and Reactive Power from generators • Active and reactive power from loads • PMU data	
What will the output of the service be?	Set point of Active and/or Reactive Power	
What is the sampling rate for the service per second?	1-10 sec	Voltage control can be run in the given range of control loops. Shorter control loops are not relevant.
What is the message size?	10-100 kB	Range varies depending on the type of device and information content. Given range includes the whole possible range of devices
What are the latency requirements of the service in milliseconds?	100-500 ms	Given the minimum control loop, total latency can't exceed half of such control loop rate
Does the service need to be instantiated at the trial site?	The service should run as closer as possible to the edge. But it is not geographically distributed.	Local processing would be beneficial for the algorithm.
How many data points on the network does the service need data from to be effective?	The service requires voltage and power measurements of all the nodes where the controllable devices are installed	Such measurements can be either provided by a dense monitoring network on the field or estimated with a State estimator, with a limited number of measurement point on the field
Are there security considerations needed from an algorithm execution point of view?	The service could communicate with external sources to receive update of the grid configuration and eventually send the control output to a visualization tool	Such external sources would be, anyway, own or managed by grid operators, so their security requirements and features have to be taken into consideration
What devices does the algorithm need to communicate with in terms of getting data?	Inverters, Battery Storage System and other controllable loads if present PMUs	Such devices will anyway provide data through a Meter. Controllable devices will also receive setpoint from the control services through the same interface.

Are there any communications security requirements?	Voltage control is a sensitive service, therefore data integrity and security is very important. Data error must be tested	
Is the type of communication (wired, wireless etc) relevant to the running of the algorithm?	The type of communication is not relevant as long as the latency requirements are fulfilled	
Are communications protocols relevant to the algorithm?	Communication protocols are not relevant at the moment	
What will the service need to run in terms of software?	The service needs normal python packages and does not need dedicated storage to operate	
Is time stamping relevant?	Yes, to discard not useful data	However, requirement on timestamping is not too stringent (uncertainty below one second could be acceptable)
Is there a need to store data?	No, the service react only based on the last measurements	
Is there a diagram that shows how the algorithm or service should fit into the power system?	It can be prepared	
Can this service or algorithm be trialled in a lab trial?	Yes, the plan is to first test the algorithm in the lab	First tests will be done in RWTH and UNIBO labs using power system simulators. In second phase of the project 5G lab trial test can be arranged
What components can be provided to the lab site to run the lab test with 5G as a hardware in the loop?	Power system simulations data streams	
Can data stream be provided to the lab trial for the test?	Yes	
Requirements for 5G-API	None at the moment	
Any other comments about the algorithm and service that we may not have captured from an ICT or Software Implementation Perspective?		

A.1.1.2 Solution Architecture Diagrams

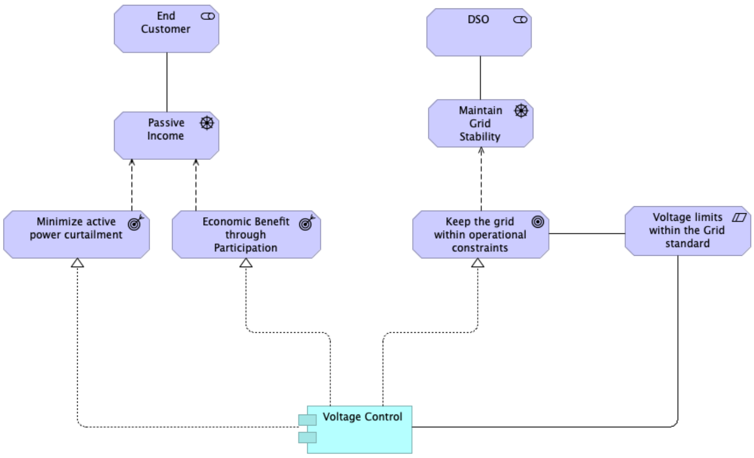


Figure 33 VC Business Architecture

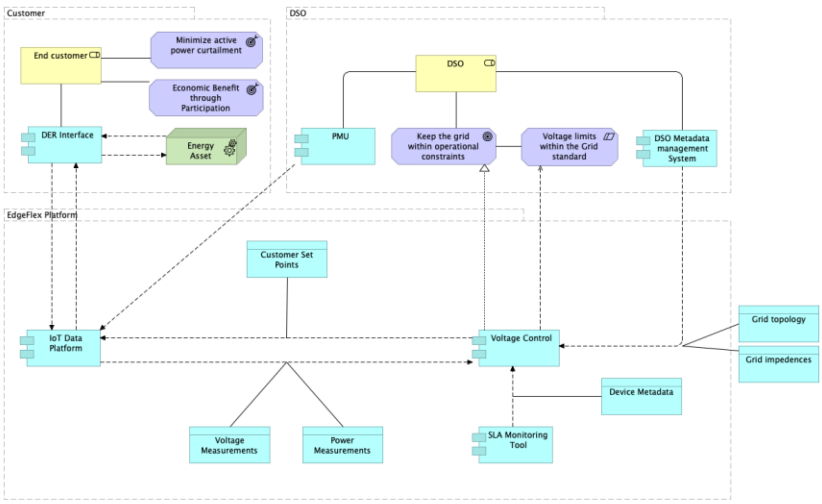


Figure 34 - VC Solution Architecture

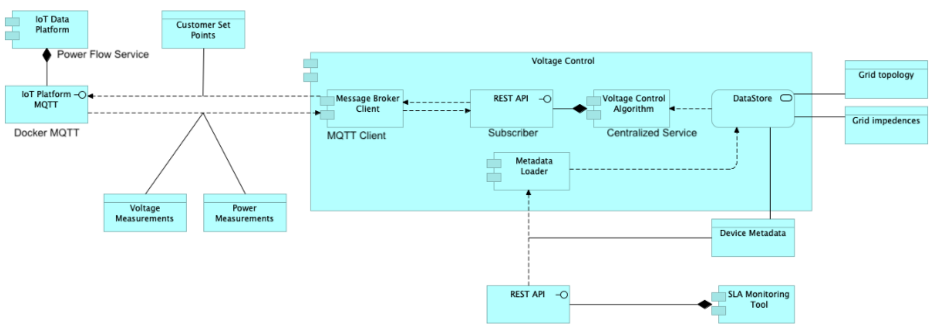


Figure 35 - VC Communications Architecture

A.1.1.3 Payload Structures

Voltage Control				
Direction	**Object**	**Key**	**Type**	**Example Value**

up	Active Power	"active_power"	List of Nodes Values	{"node_2": 0.0, "node_3": 0.0, "node_4": 0.0, …}
up	Active Power ESS	"active_power_ESS"	List of Nodes Values	{"node_6": 0.0, "node_11": 0.0, "node_12": 0.0, …}
up	Reactive Power	"reactive_power"	List of Nodes Values	{"node_2": 0.0, "node_3": 0.0, "node_4": 0.0, …}
up	PV	"pv_input_measurements"	List of Nodes Values	{"node_2": 0.0, "node_3": 0.0, "node_4": 0.0, …}
up	Voltage	"voltage_measurements"	List of Nodes Values	{"node_2": 0.999640163792404, "node_3": 0.9986819931183817, …}
up	Measurements	"test"	Float	0.1
		"voltage_measurements"	List of Nodes Values	{"node_2": 0.999817790062436, "node_3": 0.9993274712231585, …}
		"pv_input_measurements"	List of Nodes Values	{"node_2": 0.0, "node_3": 0.01065, "node_4": 0.005325, …}

A.1.2 Inertia Estimation

A.1.2.1 Service Requirements

Question	Answer	Comments
What input variables will the service need to function?	Frequency and power measurements. If power measurements are not available directly, then voltage and current phasor measurements would be enough to calculate the power.	What will the algorithm or service need to make a decision, will they be amps, frequency, volts, power factor etc?
What will the output of the service be?	Estimated value of inertia	What form will the decision be in, will it be a set point, a range, recommendation or a set of values?
What is the sampling rate of the readings?	1ms-100ms	The minimum reporting rate will highly depend on the implemented algorithm
How often will the service need to run to perform?	tens of minutes up to one hour	
What is the message size?	5 measurement quantities (frequency, voltage amplitude, voltage angle, current amplitude, current angle). Each quantity of type double. Additionally,	The message size indicates size of the message transmitted between sensor and control center. It is enough to say how

	timestamps are also needed for each of the measurements.	many measurement values the message will carry?
What are the latency requirements of the service in milliseconds?	Latency is not an issue, since the algorithm relies on post processing of data.	This can be a range also.
Does the service need to be instantiated at the trial site?	No	How close to the edge does the service need to be to be effective? Is local processing needed? Local processing should be run in edge cloud not more than few kilometers away from the device. Distributed processing needed? Algorithms need to be deployed further away from devices. Communications with several edge clouds needed.
How many data points on the network does the service need data from to be effective?	Limited number - only measurements from generation units that are providing inertia	Is the service just concerned with one device/node/feeder or does it need more visibility?
Are there security considerations needed from an algorithm execution point of view?	no	Does the service need to communicate with external sources for configuration parameters or send monitoring data?
What devices does the algorithm need to communicate with in terms of getting data?	PMUs, Power meter devices in substations	
Are there any communications security requirements?	No	Criticality of receipt of individual messages? Sensitivity to modification of messages (e.g. because of cyber-attacks)? What is maximum allowed data error rate (e.g.10% delayed or loss messages can be allowed that system functioning properly).
Is the type of communication (wired, wireless etc) relevant to the running of the algorithm?	No	
Are communications protocols relevant to the algorithm?	No	E.g.: MMS, 61850, MQTT, AMQP...
What will the service need to run in terms of software?	No	Does the service require any machine learning tools installed for instance or does it need a dedicated data store to operate.
Is time stamping relevant?	Yes	Does the algorithm require timestamps that need to be synchronised across the system.

Is there a need to store data?	Yes	Does the service or algorithm need to store data or need historical data to run?
Is there a diagram that shows how the algorithm or service should fit into the power system?	Yes – figure shown below The algorithm is meant to run by the system operator but can also run at the VPP level in case there is distributed generation providing inertia.	
Can this service or algorithm be trialled in a lab trial?	Yes- can be tested as part of the platform	
What components can be provided to the lab site to run the lab test with 5G as a hardware in the loop?	The algorithm is to be delivered to WP4 for platform and architecture integration. Lab trial should take the platform integration from WP4 and deploy for testing.	E.g. power network simulator, the service software that can be deployed in the test system
Can data stream be provided to the lab trial for the test?	Required input specifications can be provided to generate synthetic data streams	
Requirements for 5G-API	none	Specify what 5G capability is needed for VPP application to improve its performance. E.g. VPP optimisation algorithm needs to gather the info about current utilization of edge cloud computational/storage resources. Based on that info, the application can decide to put additional workload to the local edge cloud or onto neighbouring one, or to migrate workload the edge cloud gets overloaded etc.
Any other comments about the algorithm and service that we may not have captured from an ICT or Software Implementation Perspective?	no	

A.1.2.2 Solution Architecture Diagrams

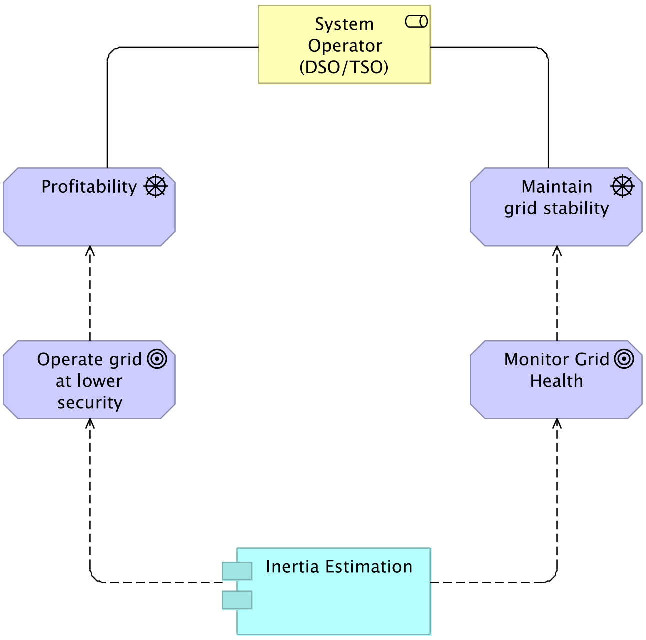


Figure 36 IE Business Architecture

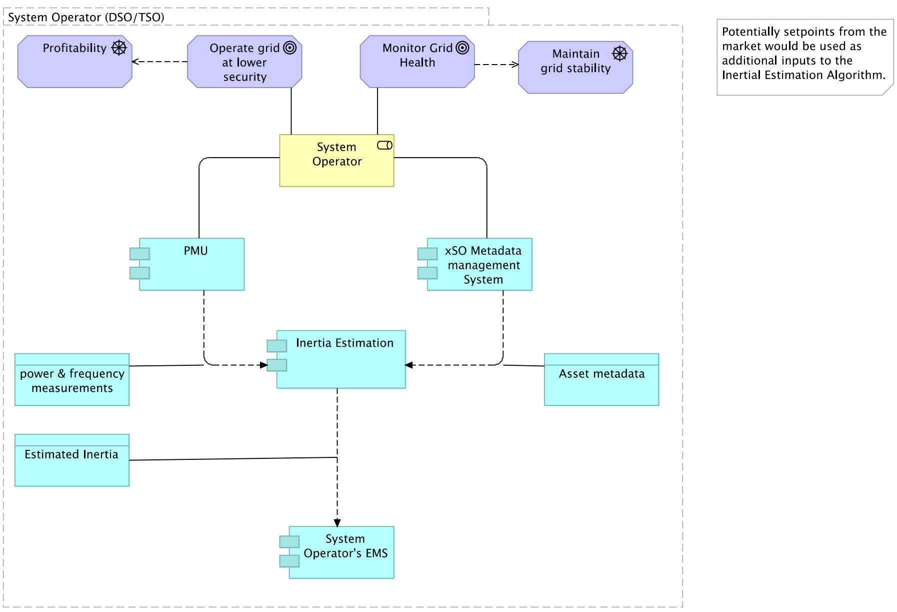


Figure 37 IE Solution Architecture

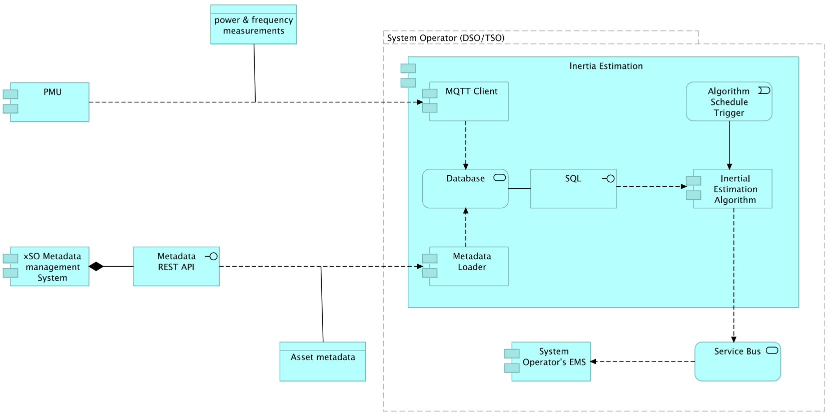


Figure 38 IE Communications Architecture

A.1.2.3 Payload Structures

Inertia Estimation				
Direction	**topic**	**Key**	**Type**	**Example Value**
up	Inertia Estimation	“P1”	Float	90225188.1517576
		“P2”	Float	100278839.04664
		“P3”	Float	95922419.1452441
		“f1”	Float	50.0000503973047
		“f2”	Float	50.0009261526443
		“f3”	Float	50.0009048672263
		“trigger”	Float	0.0
		"timestamp"	bigInt	20
		“test”	Float	0.5

A.1.3 Frequency Control

A.1.3.1 Service Requirements

Question	Answer	Comments
What input variables will the service need to function?	Frequency and active power (or amps+volts) from each units	What will the algorithm or service need to make a decision, will they be amps, frequency, volts, power factor etc?
What will the output of the service be?	Setpoint, specifically, the power reference and frequency reference for each units	What form will the decision be in, will it be a set point, a range, recommendation or a set of values?
What is the sampling rate for the service per second?	Depends on the wind stochastics. A large wind power change, or the	How often will the service need to run to perform or how often will the

	setpoint change from TSO, it need re-sampling	service need readings in order to make a decision?
What is the message size?	Single floating-point value per time sample	The message size indicates size of the message transmitted between sensor and control center. It is enough to say how many measurement values the message will carry?
What are the latency requirements of the service in milliseconds?	Minimum possible for on-line application. Power electronics devices should be at least less than 10 ms.	This can be a range also.
Does the service need to be instantiated at the trial site?	Not sure by now. Probably not	How close to the edge does the service need to be to be effective? Is local processing needed? Local processing should be run in edge cloud not more than few kilometers away from the device. Distributed processing needed? Algorithms need to be deployed further away from devices. Communications with several edge clouds needed.
How many data points on the network does the service need data from to be effective?	Needs information from all units.	Is the service just concerned with one device/node/feeder or does it need more visibility?
Are there security considerations needed from an algorithm execution point of view?	No.	Does the service need to communicate with external sources for configuration parameters or send monitoring data?
What devices does the algorithm need to communicate with in terms of getting data?	PMU, all the units, the optimisation result (VPP setpoint) from WP3	
Are there any communications security requirements?	No	Criticality of receipt of individual messages? Sensitivity to modification of messages (e.g. because of cyber-attacks)? What is maximum allowed data error rate (e.g.10% delayed or loss messages can be allowed that system functioning properly).
Is the type of communication (wired, wireless etc) relevant to the running of the algorithm?	No	
Are communications protocols relevant to the algorithm?	No	E.g.: MMS, 61850, MQTT, AMQP...

What will the service need to run in terms of software?	Simple operator.	Does the service require any machine learning tools installed for instance or does it need a dedicated data store to operate.
Is time stamping relevant?	Yes	Does the algorithm require timestamps that need to be synchronised across the system.
Is there a need to store data?	Yes	Does the service or algorithm need to store data or need historical data to run?
Is there a diagram that shows how the algorithm or service should fit into the power system?	Algorithm is simple mathematical equations.	
Can this service or algorithm be trialled in a lab trial?	Yes	
What components can be provided to the lab site to run the lab test with 5G as a hardware in the loop?	The algorithm need communicate with all the units.	E.g. power network simulator, the service software that can be deployed in the test system
Can data stream be provided to the lab trial for the test?	Yes	
Requirements for 5G-API	Unsure	Specify what 5G capability is needed for VPP application to improve its performance. E.g. VPP optimisation algorithm needs to gather the info about current utilization of edge cloud computational/storage resources. Based on that info, the application can decide to put additional workload to the local edge cloud or onto neighbouring one, or to migrate workload the edge cloud gets overloaded etc.
Any other comments about the algorithm and service that we may not have captured from an ICT or Software Implementation Perspective?	None	

A.1.3.2 Solution Architecture Diagrams

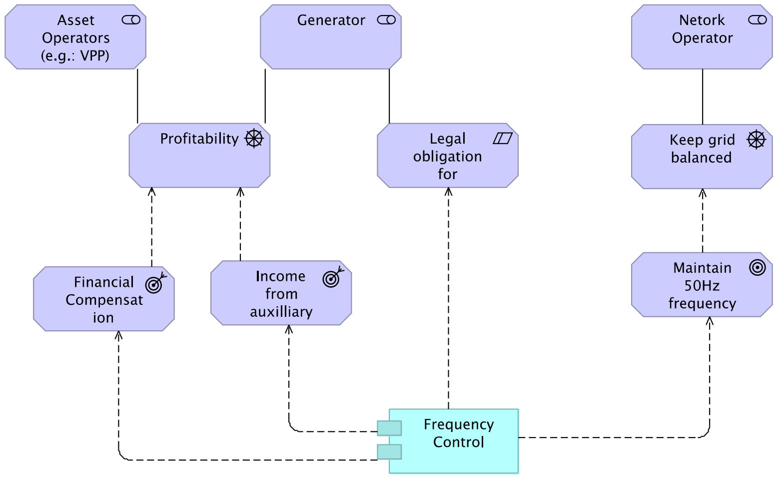


Figure 39 FC Business Solution Architecture

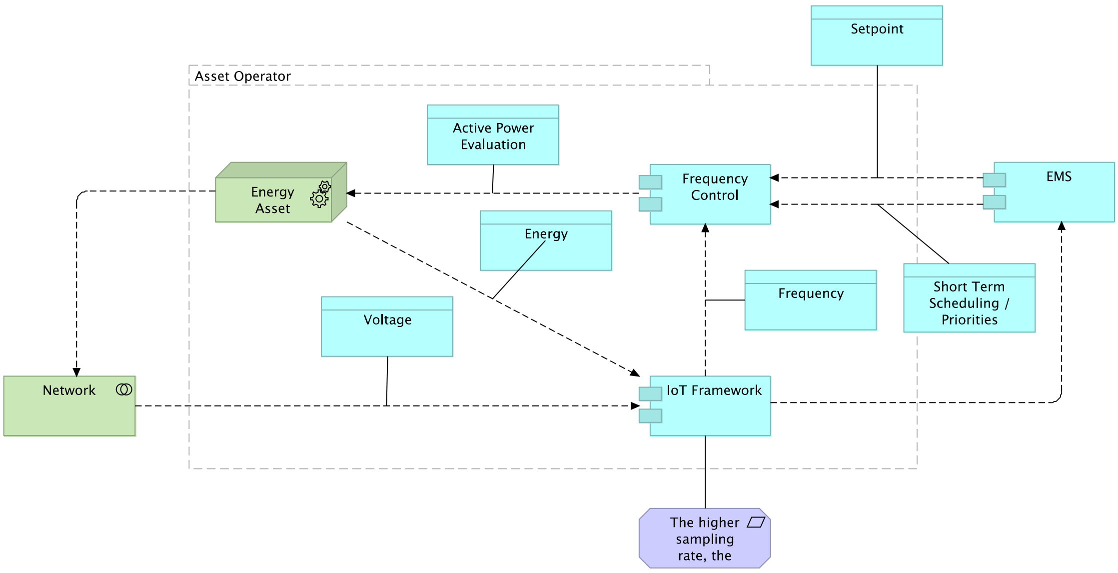


Figure 40 FC Solution Architecture

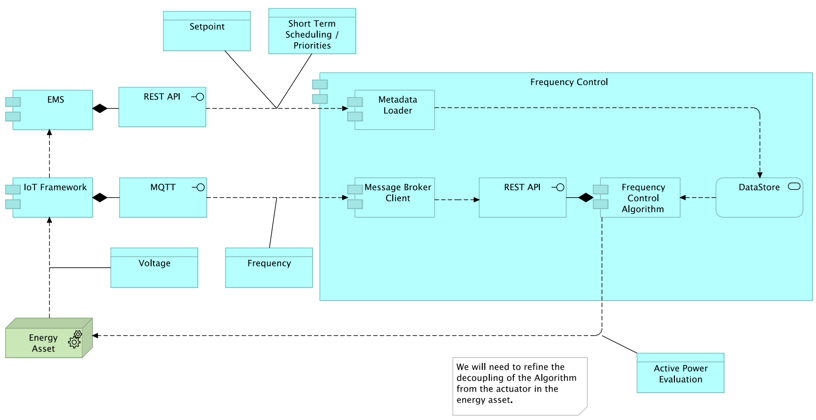


Figure 41 FC Communications Architecture

A.1.3.3 Payload Structures

VPP coordinated frequency control				
Direction	*MQTT Topic*	*Key*	*Type*	*Sample*
up	AGC	“mode”	String	“AGC”
		“scenario”	String	“vpp_agc”
		“meta”	Object	{"droops": [0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5], "gain": 60}
		“time”	bigint	1612990917280
		“active_power”	Array	[0.0188, 0.0188, 0.0188, 0.0188, 0.0188, 0.0188, 0.0188, 0.0188, 0.0188, 0.0188]
		“agc_ref”	float	0.188
up	Vpp agc AGC crl	“mode”	String	“vpp_agc”
		“scernario”	String	“AGC”
		“time”	Int	1612990999861
		“agc_out”	Array	[0.0, 1.2690000000000001, 2.5380000000000003, 3.8070000000000004, …]
		“pwr_ord”	Array of Arrays	[[0.0, 0.6345000000000001, 1.2690000000000001, 1.9035000000000002, 2.5380000000000003, 3.1725000000000003, 3.8070000000000004, 4.4415000000000004, 5.0760000000000005, 5.7105000000000001],..]

VPP automatic generation control				
Direction	*MQTT Topic*	*Key*	*Type*	*Sample*
up	COORD	“mode”	String	“COORD”
		“scenario”	String	“vpp_coordinated_ffc”
		“meta”	Object	{"ki": [0.2, 0.2, 0.2, 0.5], "gain": 60}
		“time”	bigint	1612991880249
		“active_power”	Float	-0.262979

		"frequency_meas"	Float	1.000329
up	COORD ctl out	“mode”	String	“COORD”
		“scenario”	String	“vpp_coordinated_ffc”
		“ctl_out”	Float	-0.007729489439999652
		“pwr_ord”	Array	[-0.0015458978879999306, -0.0015458978879999306, -0.0015458978879999306, -0.003864744719999826]
		“time”	Int	1612993129282

Frequency regulation metering				
Direction	*MQTT Topic*	*Key*	*Type*	*Example Value*
up	ROCOP	“mode”	String	“ROCOP”
		“scenario”	String	"fc_metering_ex1"
		“meta”	Object	{"admitt_ess": [-13.6979785967, 23.4822490229, -9.7842704262], "admitt_tcl": [-13.6979785967, 23.4822490229, -9.7842704262], "fn": 60}
		“time”	bigint	1612992871726
		“freq_ess”	Array	[1.002614, 1.002615, 1.002615]
		“freq_tcl”	Array	[1.004437, 1.004437, 1.004438]
up	FC metering ex1 ROCOP ctl	“mode”	String	“ROCOP”
		“scenario”	String	"fc_metering_ex1"
		"rocop_ess"	Array	[0.0, 0.0, 0.0]
		"rocop_tcl"	Array	[-9.094947017729282e-13, -9.094947017729282e-13, -9.094947017729282e-13]
		“time”	bigint	1612992905857

A.2 Flexibility Trading – FlexOffer Request Details

Table 1 Payload parameters of the request between VPS and FOA

Name	Type	Direction	Description
senderId	String	To VPS	Holds the FA component unique Id
creationTime	String	To VPS	UTC moment of the message creation in the "YYYY-MM-DDTHH:mm:ssZ" format
receiveStatus	Int	To FA	Reception confirmation or rejection
receiveStatusTxt	String	To FA	Details of the reception
heartBeat	JSON	To VPS	Holds the Observation component status information
flexOffer	Array of JSON	To VPS	An array of prosumer FlexOffers
flexOfferSchedule	Array of JSON	To FA	An array of prosumer FlexOffers schedules to FA.
dataReadings	Array of JSON	To FA	An array of time series holding the future adaptation price

Table 2 The structure of the heartbeat message

Attribute		Description	Type
offeredById		Unique ID representing issuer of the information.	String
faTime		Creation time of the message	DateTime
meterReadings		Array of structures of data series	JSON array
operationInfo		Structure describing the operation parameters of the prosumer	JSON array
operationInfo	intervalLength	Duration of the FlexOffer slices/intervals in terms of No. of seconds (default 900)	Integer
	operationState	Describes the availability for adaptation. The value is enumerator: ("0" = not available, "1" = online, "2" = available, "3"=waiting for execution, "4" =in adaptation)	Integer
	operationPower	Momentary consumption or production power of prosumer (sum of powers of flexible, i.e. controlled by xEMS, and non-flexible appliances). Negative value means consumption, positive values means production	JSON array
	accountingPower	The averaged production or consumption power which reflects the values from the accounting meter. Negative value means consumption, positive values means production. In the case there was communication interruption between FA and xEMS the missing values	JSON array

		should be sent in the form of the time series ("timestamp", "value").	
	operationPrognoses	Time series of the future prosumer's operation (sum of powers of flexible, i.e. controlled by xEMS, and non-flexible appliances). Negative values are consumption, positive values are production. It is a series of ("timestamp", "value") pairs.	JSON array

Table 3 The structure of the FlexOffer message

Attribute	Description	Type
Id	FlexOffer (FO) Unique ID	Int
State	FO State (offered / rejected/ assigned/ executed/ invalid/ canceled)	String
stateReason	Reason for FO Rejection (if state == Rejected)	String
numSecondsPerInterval	Number of second in for each FlexOffer interval (default 900)	Int
creationTime	Absolute time the FO is created	Datetime
offeredById	Unique ID representing FO owner	String
locationId	Unique ID representing location of the generated FO in the grid system	String
acceptBeforeTime	Absolute time before which FO must be accepted, i.e., decision regarding the acceptance or rejection of FO must be made	Datetime
acceptBeforeInterval	Interval before which FO must be accepted	Longint
assignmentBeforeTime	Absolute time before which an FO must be scheduled	Datetime
assignmentBeforeInterval	Interval before the scheduled start time of the FO	Longint
totalEnergyConstraint	Total minimum and maximum energy to be assigned for an FO, e.g., a sum of min and max energy for all slices in an FO	JSON
subSotalEnergyConstraint	The upper and lower limit of the storage energy.	JSON
startAfterTime	Absolute time after which a FO must be started	Datetime
startBeforeTime	Absolute time before which a FO must be started	Datetime

flexOfferProfileConstraints	Minimum and Maximum energy, duration, and tariff bounds for each FO slices.	FlexofferSlice
flexOfferTariffConstraint	Minimum and Maximum tariff and its duration for each FO slices.	TariffSlice
defaultSchedule	Default energy consumption and time schedule of a FO	Schedule Slice

Table 4 The structure of the FlexOffer schedule message

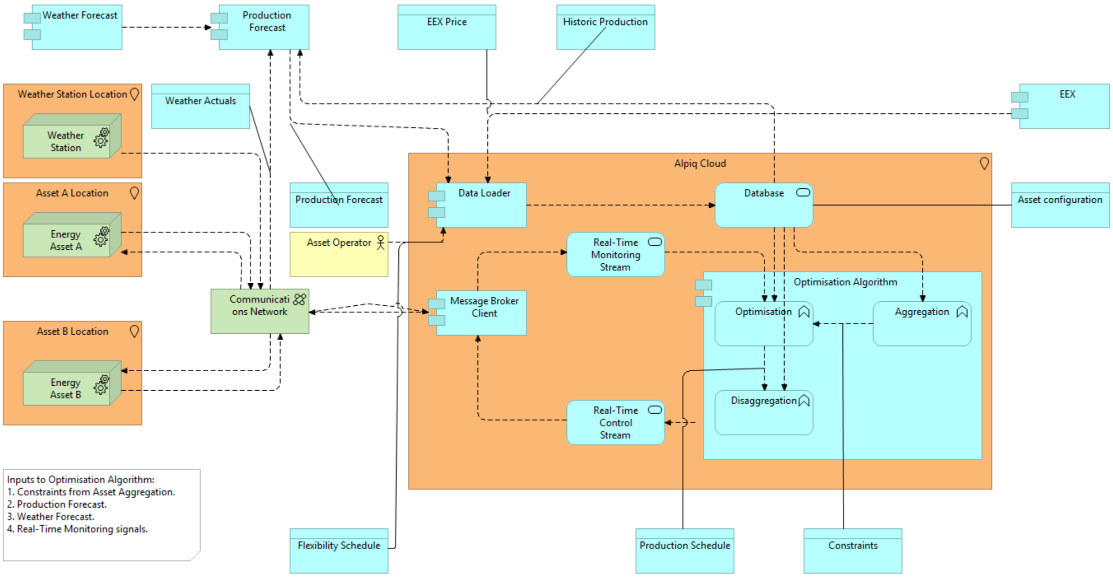
Attribute	Description	Type
id	FlexOffer (FO) Unique ID	Int
state	FO State (initial/ offered/ accepted/ rejected/ assigned/ executed/ invalid/canceled)	String
stateReason	Reason for FO Rejection (if state == Rejected)	String
numSecondsPerInterval	Number of second in for each FlexOffer interval (default 900)	Int
flexOfferSchedule	Default energy consumption and time schedule of a FO	JSON

Table 5 The structure of the adaptation price & amount message

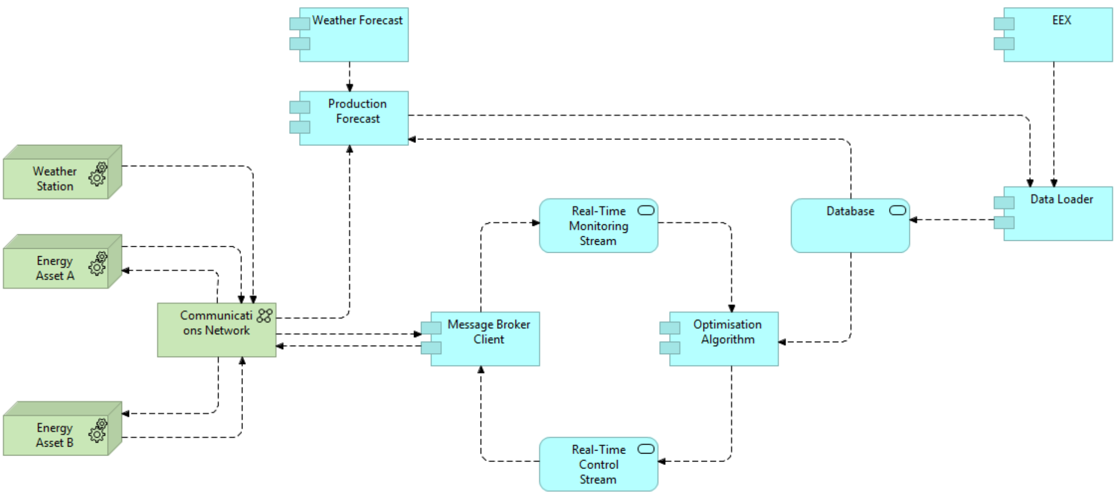
Name	Type	Description
receiveStatus	Int	Reception confirmation or rejection
receiveStatusTxt	String	Details of the reception
dataReadings	Array of JSON	An array of times series data

A.3 Commercial System Service Architecture

A.3.1 Cross-Layer Architecture View



A.3.2 Application Components View



A.3.3 Application Communication Diagram