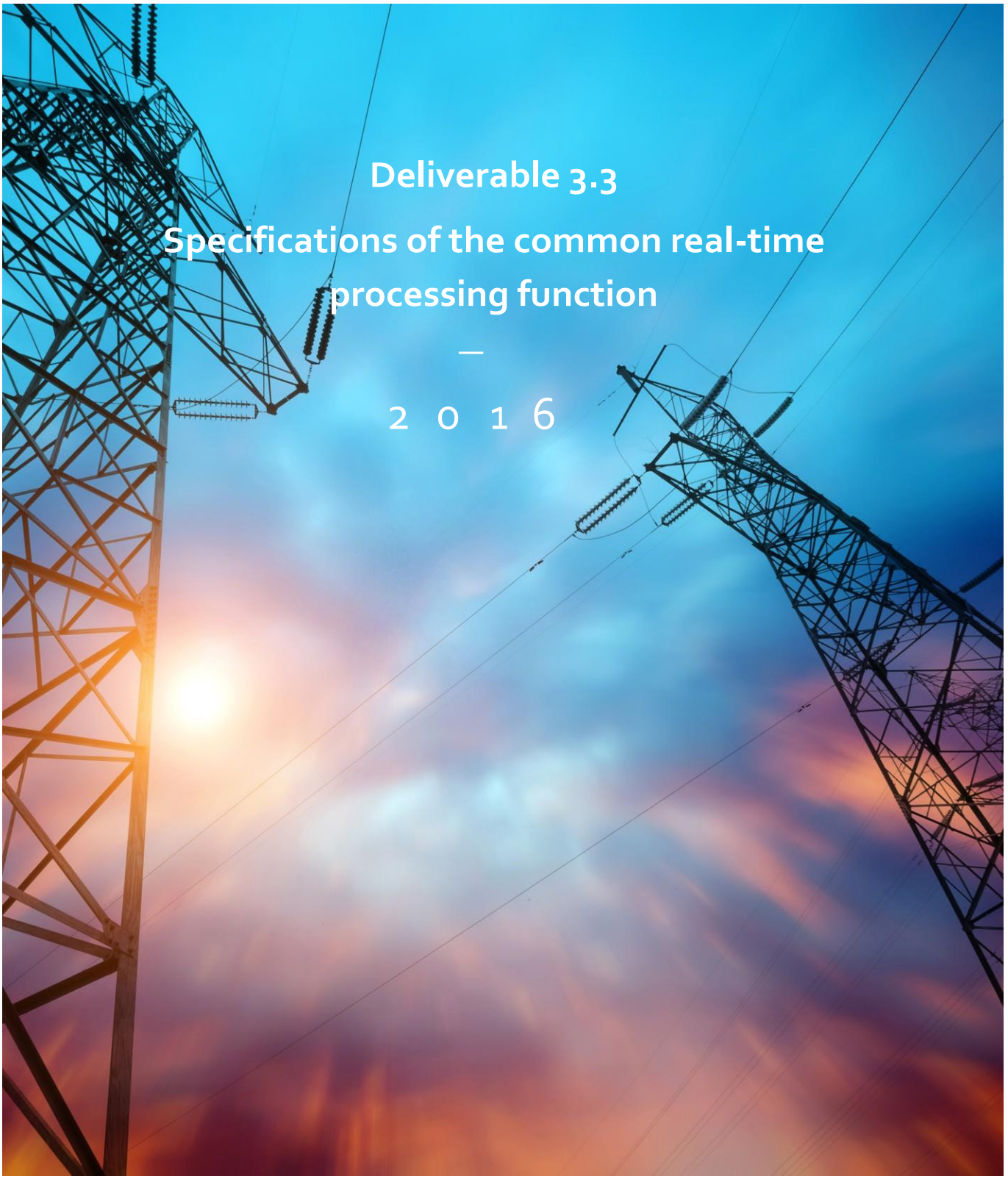
FutureFlow



Deliverable 3.3

Specifications of the common real-time processing function

—

2016

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n° 691777

Document Information

Programme	Horizon 2020 – Cooperation / Energy
Project acronym	FutureFlow
GA number	691777
Deliverable	D3.3
WP/Task related	W3 / T3.3
Type	Report
Confidential:	Members of consortium
Date of delivery	06.06.2017
Status and Version	Version 1
Number of pages	31
Document Responsible	ELES
Authors	Matjaž Dolinar – ELES
	Marko Mihorko - ELES
	David Gerbec - ELES
	Darko Kramar - ELES
Reviewers	EIMV
	SAP

Version	Date	Author(s)	Notes	Status
1	13.12.2016	ELES	Draft	

FutureFlow

Table of Contents

Document Information 2

Table of Contents 4

List of Figures 5

List of Tables 6

Table of acronyms 7

Glossary 8

The aim of the FutureFlow Project 9

Project Partners 10

Executive summary 11

1 Introduction 12

1.1 Outline 12
1.2 Relation to other work packages 12

2 Demonstration architecture overview 13

3 Functional requirements 15

3.1 Demo site 15
3.2 Development environment 17
3.3 Functional description of development environment supporting building blocks 21

4 Demo site implementation overview 25

4.1 Information layer 25
4.2 Communication layer 26

5 Conclusions 28

6 References 29

7 Appendix 1: Control Option1, Option2 structures 30

8 Appendix 2: LFC extension platform integration/function 31

List of Figures

Figure 1: FutureFlow demonstration architecture. ..... **Fehler! Textmarke nicht definiert.**
Figure 2: Demo site I/O connections. 15
Figure 3: LFC extensions for all control zones, integration within development environment. 18
Figure 4: LFC extension I/O signals. 19
Figure 8: Information architecture of demo site. 26
Figure 9: Communication architecture of demo site. 27
Figure 10: Control Option1 LFC controller structure. 30
Figure 11: Control Option2 LFC controller structure. 30

List of Tables

Table 1 : FutureFlow Demo site interfaces and protocols. 16
Table 2 : LFC controller I/O signal specification, generic. 20
Table 4: Unit response model signal specification, generic. 20
Table 4 : Historical signal instantiation function. 21
Table 5 : Signal resampling function. 22
Table 6 : Matlab to OPC DA conversion function, generic. 22
Table 7 : OPC DA to Matlab conversion function, generic. 22
Table 8 : Matlab to MQTT conversion function, generic. 23
Table 9 : MQTT to Matlab conversion function, generic. 23
Table 11: Matlab to IEC104 conversion function, generic....23
Table 12: IEC104 to Matlab conversion function, generic....24

Table of acronyms

Acronym	Meaning
ACE	Area Control Error
aFRR	Automatic Frequency Restoration Reserve
API	Application program interface
DG	Distributed generation
DR	Demand response
EMS	Energy Management System
FCR	Frequency Containment Reserve
FF	FutureFlow
GW	Gateway
HMI	Human machine interface
HTTPS	Hypertext Transfer Protocol Secure (HTTP over TLS)
ICCP	Inter-Control Center Communications Protocol (TASE.2)
ICT	Information and Communication technology
IEC 104	Protocol IEC 60870-5-104
LFC	Load Frequency Control
mFRR	Manual Frequency Restoration Reserves
MQTT	MQ telemetry protocol
OPC	OLE for process control
REST	Representational State Transfer
TSO	Transmission system operator
UI	User Interface
UML	Unified modelling language
VPN	Virtual Private Network
VPP	Virtual Power Plant

Glossary

Refer to ENTSO-E glossary, https://www.entsoe.eu/data/data-portal/glossary/Pages/home.aspx.

The aim of the FutureFlow Project

Four European TSOs of Central-Eastern Europe (Austria, Hungary, Romania, Slovenia), associated with power system experts, electricity retailers, IT providers and renewable electricity providers, propose to design a unique regional cooperation scheme: it aims at opening Balancing and Redispatching markets to new sources of flexibility and supporting such sources to act on such markets competitively. By means of a prototype aggregation solution and renewable generation forecasting techniques, flexibility providers – distributed generators (DG) and commercial and industrial (C&I) consumers providing demand response (DR) – are enabled, to provide competitive offers for Frequency Restoration Reserve (including secondary control activated with a response time between 30 seconds and 15 minutes). Retailers act as flexibility aggregators and pool the resource in order to provide the products required by the TSO. A comprehensive techno-economic model for the cross border integration of such services involves a common activation function (CAF) tailored to deal with congested borders and optimized to overcome critical intra-regional barriers. The resulting CAF is implemented into a prototype Regional Balancing and Redispatching Platform, securely integrated within the four TSOs' IT systems: this makes research activities about cross-border integration flexible while linking with the aggregation solution. Use cases of growing complexity are pilot-tested, going from the involvement of DR and DG into national balancing markets to cross border competition between flexibility providers. Based on past experience with tertiary reserve, participating C&I consumers and DG are expected to provide close to 40 MW of secondary reserve. Impact analyses of the pilot tests together with dissemination activities towards all the stakeholders of the electricity value chain will recommend business models and deployment roadmaps for the most promising use cases, which, in turn, contribute to the practical implementation of the European Balancing Target Model by 2020.

Project Partners

No	Name	Short name	Country
1	ELES DOO SISTEMSKI OPERATOR PRENOSNEGA ELEKTROENERGETSKEGA OMREZJA	ELES, d.o.o.	Slovenia
2	AUSTRIAN POWER GRID AG	APG	Austria
3	MAVIR MAGYAR VILLAMOSENERGIA-IPARI ATVITELI RENDSZERIRANYITO ZARTKORUEN MUKODO RESZVENYTARSASAG	MAVIR ZRT	Hungary
4	COMPANIA NATIONALA DE TRANSPORT ALENERGIEI ELECTRICE TRANSELECTRICA SA	TRANS	Romania
5	ELEKTROINSTITUT MILAN VIDMAR	EIMV	Slovenia
6	ELEKTROENERGETSKI KOORDINACIONI CENTAR DOO	EKC	Serbia
7	ELEKTRO ENERGIJA, PODJETJE ZA PRODAJO ELEKTRIKE IN DRUGIH ENERGENTOV, SVETOVANJE IN STORITVE, D.O.O.	EE	Slovenia
8	GEN-I, TRGOVANJE IN PRODAJA ELEKTRICNE ENERGIJE, D.O.O.	GEN-I, d.o.o.	Slovenia
9	SAP SE	SAP SE	Germany
10	CYBERGRID GMBH	CYBERGRID	Austria
11	GEMALTO SA	GTO	France
12	3E NV	3E	Belgium

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n° 691777

Executive summary

This document defines specifications of the common real-time processing function.

The role of common real time processing function is to provide a tool which can enable expanding or simulating each TSO operating environment and reproducing it in the pilot test process IT system, to continuously feed the pilot test process IT system with real time data and provide the link between the TSO Cloud and real time environment of the TSOs.

Although initially predicted to be a part of TSO Cloud, the decision to implement it in a more flexible development environment was taken by the project members. The main functionalities will remain the same, however external platform will also be able to perform other functions, such as connection to pilot field units (BSPs), user friendly prototyping of local LFCs and unit responses as well as an ability to perform offline simulations.

The core functionality of this Demo site and development environment is to host fully functional load-frequency control (LFC) and modelled response of aFRR units for each TSO. Development environment is realised in Matlab/Simulink that implements control algorithms (PI and filtering) for each TSO in a separate, fully parametrizable building block. Real time data from SCADA/EMS enables real time realisation of pilot tests. Use of historical data streams or a mix of real time and historical is possible during functional test phases and selected pilot runs.

Demo site is being designed as a flexible prototyping platform, enabling efficient user interaction both on IT/communication as well as on LFC simulation and control layer.

1 Introduction

1.1 Outline

This document defines specifications of the common real-time processing function.

The role of common real time processing function is to enable expanding or simulating each TSO operating environment and reproducing it in the pilot test process IT system, to continuously feed the pilot test process IT system with real time data and provide the link between the TSO Cloud and real time environment of the TSOs.

Although initially predicted to be a part of TSO Cloud, the decision to implement it in a more flexible development environment was taken by the project members. The main functionalities will remain the same, however this external platform, hereinafter named "Demo site" will also be able to perform other functions, such as connection to pilot field units (BSPs), user friendly prototyping of local LFCs and its unit responses and ability to perform offline simulations. The advantage of having Demo site separated from the TSO Cloud is also the fact that Demo site will essentially mimic currently non-existing functionalities of local LFCs and provide signals to the TSO Cloud in the same way as LFCs in the future would; therefore, such system configuration will enable more thorough and realistic tests of the Cloud functions performance.

To simplify data exchange between partners during the pilot tests and avoid real-time system modifications as much as possible, it was decided to implement the Demo site within one of the participating TSO's environment. That TSO (ELES) will communicate with other participating TSOs via existing communication paths and will act as a host of the Common real-time processing function for all other TSOs.

Functional description of common real-time processing function within Demo site is organised using top down approach, covering high level requirements at the beginning and more details towards the end. Some implementation solutions are proposed, e.g. simulation packages, gateways, message queue servers.

Document is the basis for implementation partner to realise currently identified required functionalities on the Demo site.

1.2 Relation to other work packages

Deliverable 3.3 (D3.3) is related to WP1, deliverables D1.1, D1.2 and D1.3 dealing with project concept and main system overview. It is also related to WP2, Task 2.1 "Specification of the DR & DG flexibility aggregation platform" that specifies aFRR aggregation platform residing within BSP and its communication with flexibility provider. Detailed API view of CAF interfacing is defined in WP3, Task 3.1 "Specifications of the prototype regional balancing and redispatching platform with common activation function for FRR".

2 Demonstration architecture overview

FutureFlow demonstration architecture is used for running pilot tests from all control zones. It will be implemented within a secure TSO environment as a hosting site (ELES demo site) for all other TSOs (APG, MAVIR, TRANS) and BSPs in FutureFlow (FF BSPs). ELES Demo Site is planned to be connecting all participating TSO to FutureFlow Cloud Platform (**Fehler! Verweisquelle konnte nicht gefunden werden.**). A high level overview of main building blocks of the planned architecture is described below.

2.1.1 Architecture building blocks

The demonstration architecture (Note: this is the evolved architecture model from D1.3, with additional details.) consists of these main system building blocks:

1. TSOs and FutureFlow Demo site platform. The central part of the Demo site is the integrated Development environment containing main common real time processing functionalities (LFC extensions, modelled aFRR unit responses, data management functions). The demo site is hosted at ELES, connected to its SCADA/EMS via OPC-DA and through ELES to other TSOs via existing TSO-TSO communication paths. The Cross-border capacity management information (CZC) can separately be exchanged between TSOs and the FutureFlow Cloud via a secure Internet link (green line). Alternative exchange of CZC is also possible through Demo site to FF Cloud platform via Web services. The web human Interface, enabling TSOs the management of the FF Cloud platform applications (e.g. for HMI bid management interaction) can be accessed via secured Internet link (purple line).,
2. FutureFlow balancing service providers (FF BSPs), which are introduced through WP2 into the project, being located in each TSO control area (SI, AT, HU, RO) are managing their own DR/DG flexibilities pools. In the demonstration architecture, all FF BSPs communicate through the Demo site instead through their local TSO as in target solution. During the pilot tests, each FF BSP is planned to be connected with a secure link over public Internet paths (e.g. VPN or Internet), with separate data flows for measurement and control (blue line) and bidding data (yellow line).
3. The FutureFlow Cloud (based on SAP HANA Cloud Platform) implements common functionalities and connects to Demo site using various communication protocols. The FF Cloud platform has three main interfaces: real-time data interface, web services interface and human web interface.

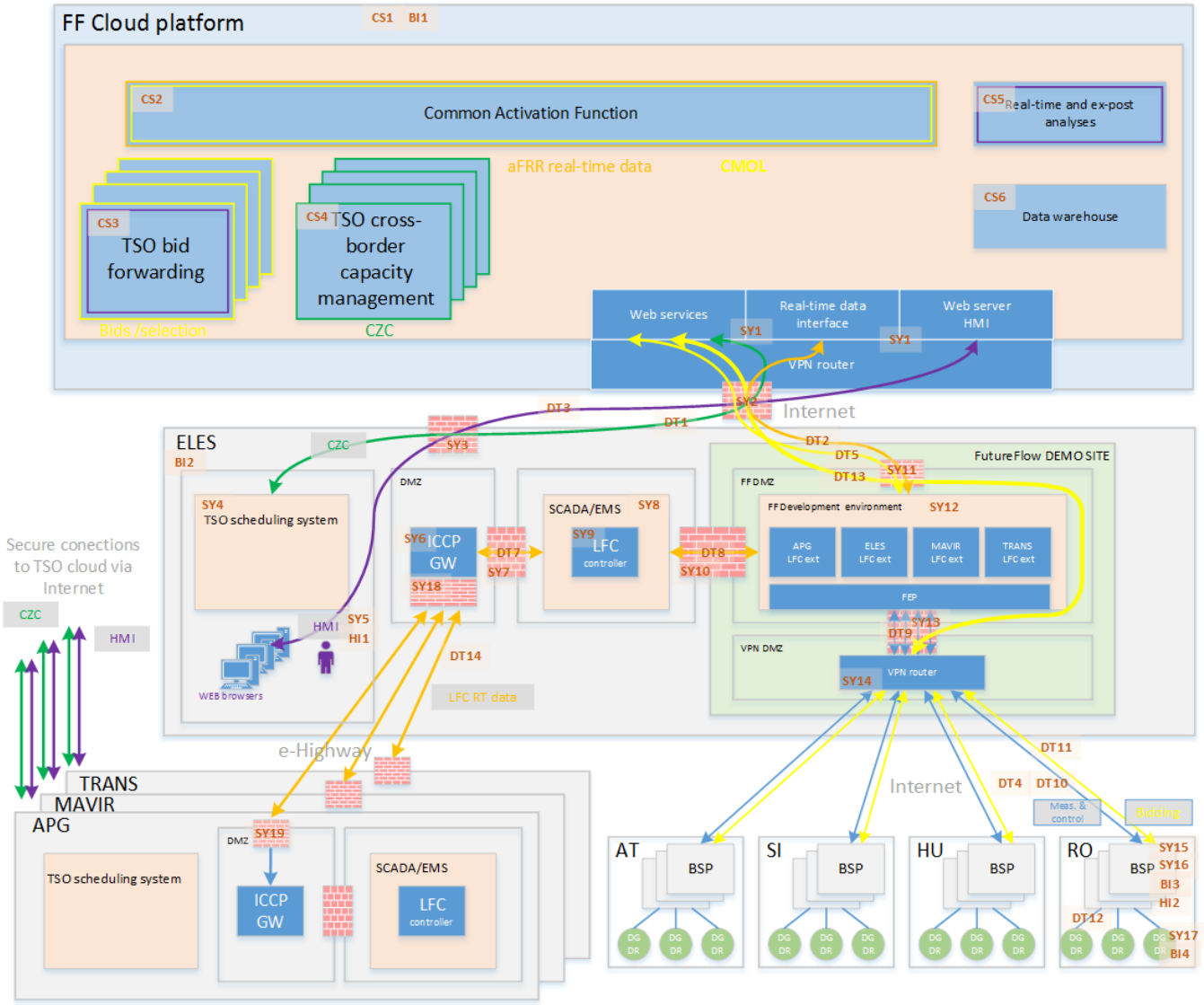


Figure 1: FutureFlow demonstration architecture.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n° 691777

3 Functional requirements

This chapter defines the FutureFlow Demo Site functional requirements. As one of the Demo Site functions, the aim is to contain all the functionalities of this common real time processing function, that is to expand or simulate each TSO operating environment and reproduce it in the pilot test process IT system, to continuously feed the pilot test process IT system with real time data and to provide the link between the FutureFlow Cloud and real time environment of the TSOs through the existing data communication channels between the TSOs. In addition to common real-time processing function, Demo site also has to be able to perform other functions, such as connection to pilot field units (BSPs involved in the project) during the pilot tests, user friendly prototyping of local LFC extension environments and its aFRR unit responses as well as an ability to perform offline and online simulations by the TSOs. Solutions described or planned in this document shall not construe a legally bind for use or apply by any member party; therefore, these solutions are only for R&D purposes.

Description follows top down approach, gradually exposing more detailed structure of systems building blocks as the document proceeds.

3.1 Demo site

Demo site is the top level architecture environment hosting development environment and other supporting functions, e.g. exposed communication interfaces, VPN routers, firewalls, etc. A block diagram depicting I/O signal connections is shown in Figure 2. User interaction is denoted in violet, emphasizing involvement of project partners during development, testing and operation of the solution.

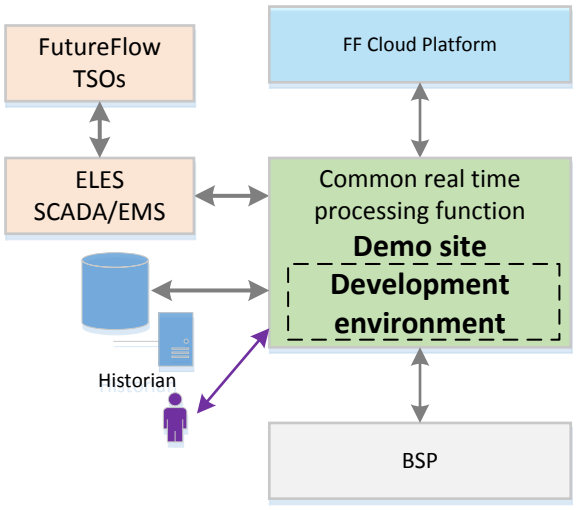


Figure 2: Demo site I/O connections.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n° 691777

3.1.1 Interaction of Demo site with other FutureFlow components

Common entry point and data exchange of Demo site with TSOs is through ELES ICCP/OPC DA gateway that connects SCADA/EMS of hosting TSO (ELES). This gateway transports data streams from ELES and other TSOs connected to the development environment through ELES. Incoming stream from partner TSOs is via TSO-TSO communication path and is handled by ELES in coordination with partner TSOs, meaning that there is no direct connection of other TSOs to the Demo site.

FF Cloud platform interaction is via a set of one way message queues (MQTT), (separate for each partner TSO to simulate connection of individual TSOs as in target solution). MQTT architecture is deployed with MQTT broker (server) implemented by the Demo Site and MQTT client implemented by the FF Cloud. The preferred implementation of MQTT server (in this architecture) is Mosquitto [Mosq 2016]. There is a single MQTT server for the whole demo system.

Alternatively, a solution with the MQTT broker implemented in the FF Cloud and with FF Demo Site connecting as client is also possible and the implementation decision is to be decided during subsequent development phases. TLS protocol security must be enabled for all MQTT transfers (at router, firewall).

All data streams are combined into single VPN router and firewall that established a communication channel via Internet to the FF Cloud.

External connection to FF BSP (aFRR aggregation platforms) is again via VPN router and firewall to Internet. These connections are to all BSPs in all control zones, each to specific IP address. Protocol interface is either IEC 61870-5-104 or MQTT.

In addition to exchange of real time data, Demo site also has interfaces to exchange other types of data, i.e. bid data, local merit order list (MOL) data, historical data, etc. Exchange of this kind of data is exposed either via Web services interfaces or MQTT (to be decided during development, in connection with the FF Cloud). HMI interfaces are implemented via Web Server interfaces.

Interface types and protocols are described in Table 1.

Table 1 : FutureFlow Demo site interfaces and protocols.

Name / counterparties	Data interface type (purpose)	Protocol
TSO -> TSO TSO (ELES) -> FF Demo site	Real time (measurements, control)	• IEC 61870-6-503 (ICCP, TASE.2) • OPC DA • IEC 60870-5-104 • IEC 60870-5-101 (option)
FF Cloud -	Historical data stored in cloud data warehouse	• Web services: HTTP/HTTPS, REST (JSON)
HMI TSO -> Demo site	Web interface (webpage) (HMI, configuration files transfer, development environment changing	• HTTPS (HTTP over TLS)

Name / counterparties	Data interface type (purpose)	Protocol
	configuration parameters, bid editing)	
HMI FF BSP -> FF Demo Site	Web server (HMI, FF BSPs bid editing)	• HTTPS (HTTP over TLS)
FF BSP -> FF Demo Site	Real time (measurements, control of BSP)	• IEC 60870-5-104 • MQTT (option)
FF BSP -> FF Demo Site	Web services (Bidding data)	• HTTPS (HTTP over TLS)
FF Demo Site -> FF Cloud	Real time (measurements, control)	• MQTT
FF Demo Site -> FF Cloud	Web services (Bidding data, CZC)	• HTTPS (HTTP over TLS)
FF Cloud -> FF Demo Site	Balancing bid activations, effective & used CZC, correction signal, local merit order list	• MQTT

Time synchronization in specific building blocks is referenced to TSO hosted environment (ELES). Preferable timing source is IP network derived from NTPv2 (or higher) protocol. Time stamp format is according to ISO 8601 [ISO8601 2004] and RFC3339 [RFC3339 2002] and is the same as emitted by method Date.prototype.toJSON() [DtJ 2016].

3.2 Development environment

Development environment is the core component of Demo site, containing prototyping environment for simulation and real-time operation of LFC extensions, unit response modelling and various support functions like historical databases, protocol converters and real-time signal generators (Figure 3).

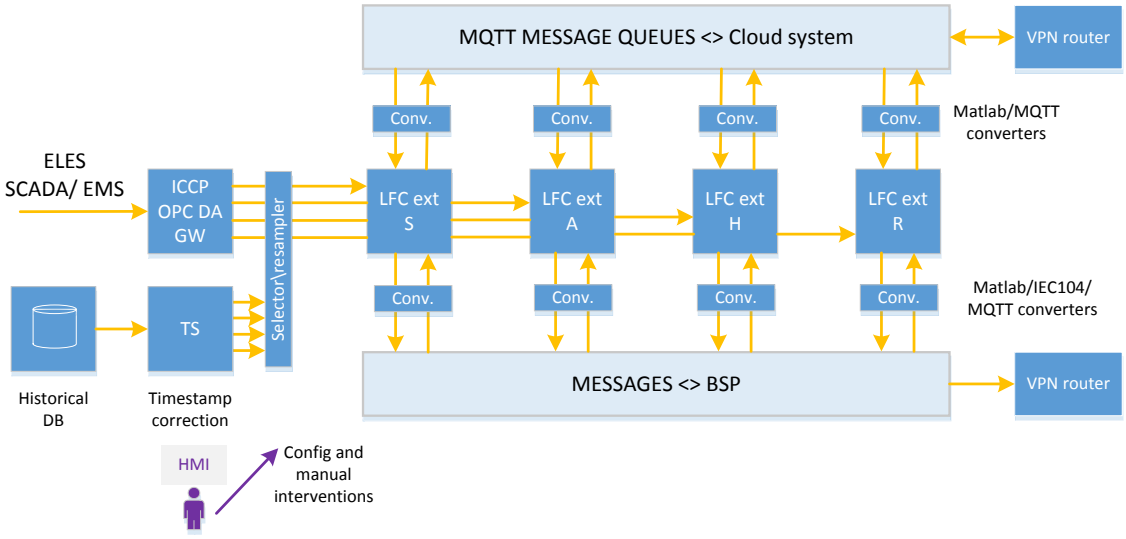


Figure 3: LFC extensions for all control zones, integration within development environment.

Main components of development environment are four instances of LFC extensions, representing a modelled version of four participating TSOs LFCs and aFRR unit modelled responses. All LFC extensions have the same basic architecture (PI controller, digital filters), but can have different transfer functions, since they are fully parametrized via HMI.

3.2.1 Modes of operation

Development environment is designed to support three modes of operation. Mode of operation depends on the nature of data used: real time or historical. These modes are:

1. Real time. All signals are real time. Real time operation can be activated only after development environment has been functionally and operationally fully validated and verified.
2. Hybrid. At least one signal is taken from historical measurements. It is active during validation tests of its building blocks. It can also be used during use case testing if any of the signals from TSO cannot be supplied in real time.
3. Simulation. All signals are taken from historical measurements. It is active during development phase of aFRR balancing energy exchange process (simulations of different aFRR balancing energy implementation options) and testing of IT environment itself.

Additionally, development environment supports different types of operation in line with pilot test use cases as defined in WP4, for example operation of single LFC extension or parallel operation of all LFC extensions.

3.2.2 HMI functionality

User interaction via HMI is vital for interactive and smooth operation of development environment. Manual user interventions are possible on:

a) Simulation models (Matlab/Simulink). Complete model design of LFC extension

and unit responses, signals selection (e.g. real time, historical, parametrisation of control loop elements).

b) Historical signals. Editing and manual generation.

c) LFC extension parametrisation. This includes configuration of LFC extension controller (PI, constants) and selection of real time or historical signals for operation.

d) Local MOL and activation schedule list. Editing, definition of final activation schedules.

In addition to HMI requirements above, parametrization of various IT platform parameters is requested via HMI.

3.2.3 Functional description of LFC extension operation and signals

LFC extension is comprised of two parts: LFC that models, mimics performance of TSO's local controller and model of that TSO unit responses. If local controller doesn't support some functionalities needed for FutureFlow designed aFRR balancing energy exchange, those functionalities have to be added to the LFC extension. However, prior to those modification, each aFRR controller and modelled aFRR unit responses have to be tested using actual configuration, to make sure that aFRR unit modelled responses are adequate and validated.

Basic PI controller structure, its integration into LFC extension environment and signal routing is shown in Appendices 1 and 2. LFC extension environment input-output signals are shown in Figure 4.

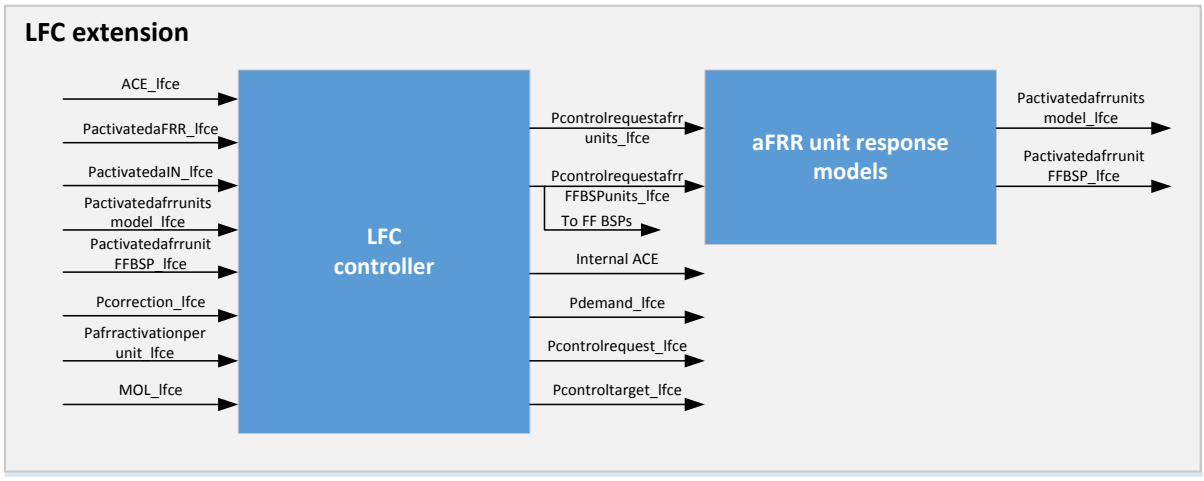


Figure 4: LFC extension I/O signals.

Since target integration model for aFRR cross-border balancing energy exchange has two alternative solutions (for additional information see D1.2), LFC extension has to support both integration modes.

Each LFC extension operates completely independently and is attached to its own set of I/O signals. A generic specification of LFC extension signals is defined in *Table 2* (for LFC controller) and *Table 3* (for aFRR unit response models).

Table 2 : LFC controller I/O signal specification, generic.

Item	Description
Inputs	• ACE_lfce; from TSO SCADA; ACE from the TSO (ACE before any inbalance netting or aFRR optimization) • PactivatedaFRR_lfce; from TSO SCADA; actual activated aFRR • PactivatedaIN_lfce; from TSO SCADA; actual exchanges due to imbalance nettings • Pactivatedafrrunitsmodel_lfce; from Unit response model; simulated responses of aFRR units from classical BSPs • PactivatedafrrunitFFBSP_lfce; measured/calculated response from FF BSPs; measured or simulated responses of aFRR units from FF BSPs • Pcorrection_lfce; from cloud platform (for LFC implementation options 1 and 2);cross-border aFRR exchange • Pafrractivationperunit_lfce; from cloud platform (for LFC implementation option 2 only); list of activated bids and activated power of each bid • MOL_lfce; from cloud system platform or from internal database; local merit order list
Outputs	• Pcontrolrequestafrrunits_lfce; to modelled units; control request for aFRR units for classical BSPs • PcontrolrequestafrrFFBSPunits_lfce; to FF BSPs; control request for aFRR units for FF BSPs • Pdemand_lfce; to cloud platform (for implementation option 1); within LFC extension's calculated Pdemand • Pcontrolrequest_lfce; to cloud platform (for implementation option 2/1); control request signal calculated by LFC extension • Pcontroltarget_lfce; to cloud platform (for implementation option 2/2) ; control target signal calculated by LFC extension • InternalACE;to storage; Internally calculated ACE for post analyses
Action	LFC extension contains aFRR controller algorithm. Preferred implementation is Matlab/Simulink, as a standalone application derived from deployable mex and S functions compiled into C (compile mcc cpplib), Java (compile mcc -W java) or Python (compile mcc -W python) code. Target is either production server (or VM) or embedded board. Generic specification has all signals, for both LFC integration options - Option 1 and Option2 type controls. This generic specification is adapted to separate LFC controller for each TSO.
Time period	All signals have time period 2 s. Exceptions: MOL (MOL _lfce), received only when updated or automatic refresh after predefined amount of time.
Time synchronisation	All signals are sampled at the same time instance, i.e. must have the same time stamp. Exceptions: MOL is versioned XML document, highest version is the actual one
DWH interaction	All signals are stored in the internal DB. Signals needed for post analysis are stored to DWH (offline bulk upload).

Table 3: Unit response model signal specification, generic.

Item	Description
Inputs	• PcontrolrequestafrrunitS_lfce; from LFC controller; control request for aFRR units from classical BSPs • PcontrolrequestafrrFFBSPunits_lfce; from LFC controller (optional); control request for aFRR units from FF BSPs (in case actual activation of FF BSPs is performed, this signal is ommited)
Outputs	• PactivatedafrrunitsmodeL_lfce; from Unit response model; simulated response of aFRR units from classical BSPs • PactivatedafrrunitFFBSP_lfce; measured/calculated response from FF BSPs; measures or simulated response of aFRR units from FF BSPs
Action	LFC controller extension contains aFRR controller algorithm. Preferred implementation is Matlab/Simulink, as a standalone application derived from deployable mex and S functions compiled into C (compile mcc cpplib), Java (compile mcc -W java) or Python (compile mcc -W python) code. Target is either production server (or VM) or embedded board. This generic specification is adapted to separate unit response models for each TSO.
Time period	All signals have time period 2 s.
Time synchronisation	All signals are sampled at the same time instance, i.e. must have the same time stamp.
DWH interaction	All signals are stored in the internal DB. Signals needed for post analysis are stored to DWH (offline bulk upload)

Note: To/from descriptions are described as originating sources or final targets. Signals are transformed in between.

3.3 Functional description of development environment supporting building blocks

Several smaller supporting building blocks are required for more flexible operation of the development environment, like managing real time and historical signals, converting between OPC, MQTT, IEC 60870-5104 (IEC 104) and Matlab formats. These are partly implemented in Matlab (toolboxes) and partly as standalone programs (microservices, REST API based) within Linux and Windows environment.

3.3.1 Real time and historical signals

Real time signals are processed in LFC extension blocks during real time mode of operation. All real time signals entering LFC extensions must have the same time period, e.g. 2 s. Resampling is perfomed on-site as described in 3.3.2.

When historical signals are used as the input to LFC extension, historical data have to be re-timestamped to actual time (instantiation function defined in *Table 4*), because FF Cloud only operates in real time mode (historical mode not applicable, thus it is also not possible to run simulations and real-time concurrently). Instantiated new historical signals are saved under different name as actual real-time signals and are managed within FF Demo Site.

Table 4 : Historical signal instantiation function.

Item	Description

Inputs	Any historical signal (either two second based real time signal or BID/CZC document.
Outputs	Time shifted input signal/document.
Action	Signals are timestamped to real-time online.
Time period	The same as input signal.
Time synchronisation	NA
DWH interaction	No. All signals are stored in Demo site internal DB. Signals needed for post analysis are stored to DWH (offline bulk upload)

3.3.2 Signal resampling

Real-time signals or historical signals are often not equally spaced, i.e. do not have an exact 2 s period. This is either due to type of data, delays in real-time communication or due to different LFC cycles of TSOs (for example 4 s instead of 2 s LFC cycle). To ensure smooth and reliable operation of LFC extension, all signals are routed through signal resampling function (essentially zero order hold function) that every two seconds takes last available input value and sends it to the output (*Table 5*).

Table 5 : Signal resampling function.

Item	Description
Inputs	Any signal with > 2 s time period.
Outputs	Any signal with 2 second period
Action	Last valid input value hold and sent to the output with 2 s period.
Time period	Output signal has time period of 2s.
Time synchronisation	NA
DWH interaction	/

3.3.3 Matlab to OPC DA conversion

All signals that interact between LFC extension block and TSO SCADA/EMS are converted with Matlab to OPC DA converter (*Table 6*).

Table 6 : Matlab to OPC DA conversion function, generic.

Item	Description
Inputs	Any signal that comes from Matlab.
Outputs	OPC DA
Action	Matlab data structuresare converted to OPC DA protocol (OPC Toolbox).
Time period	The same as input signal.
Time synchronisation	NA
DWH interaction	/

3.3.4 OPC DA to Matlab conversion

All signals that interact between TSO SCADA/EMS and LFC extension block through ELES are converted with OPC DA to Matlab converter (*Table 7*).

Table 7 : OPC DA to Matlab conversion function, generic.

Item	Description
Inputs	Any signal that comes from SCADA/EMS via OPC DA.
Outputs	Matlab data structures.
Action	Data elements from OPC DA protocol are converted to Matlab data structures (OPC Toolbox).
Time period	The same as input signal.
Time synchronisation	NA
DWH interaction	/

3.3.5 Matlab to MQTT conversion

All signals that interact between LFC extension block and cloud platform are converted with Matlab to MQTT converter (*Table 8*).

Table 8 : Matlab to MQTT conversion function, generic.

Item	Description
Inputs	Any Matlab signal that interacts with cloud.
Outputs	TCP/IP stream to MQTT converter.
Action	Matlab data structures are converted to MQTT, obeying JSON vector format. A vector of time stamps corresponding to each data element of input signal is also generated.
Time period	The same as input signal.
Time synchronisation	NA
DWH interaction	/

3.3.6 MQTT to Matlab conversion

All signals that interact between cloud platform and LFC extension block are converted with MQTT to Matlab converter (*Table 9*).

Table 9 : MQTT to Matlab conversion function, generic.

Item	Description
Inputs	TCP/IP stream from MQTT converter.
Outputs	Matlab data structures.
Action	Data elements from MQTT protocols are converted to Matlab data structures. A vector of time stamps corresponding to each data element of input signal is also generated.
Time period	The same as input signal.
Time synchronisation	NA
DWH interaction	/

3.3.7 Matlab to IEC 104 conversion

All signals that interact between LFC extension block and BSPs are converted with Matlab to OPC DA and further to OPC/IEC104 converter (*Table 10*).

Table 10: Matlab to IEC 104 conversion function, generic.

Item	Description
Inputs	Any Matlab signal that interacts with BSPs.
Outputs	OPC DA.
Action	Matlab data structures are converted to OPC/IEC104 data.
Time period	The same as input signal.
Time synchronisation	NA
DWH interaction	/

3.3.8 IEC104 to Matlab conversion

All signals that interact between BSPs platform and LFC extension block are converted with IEC 104/OPC to OPC Matlab converter (*Table 11*).

Table 11: IEC 104 to Matlab conversion function, generic.

Item	Description
Inputs	Any IEC 104 signal that interacts with Matlab.
Outputs	Matlab data structures.
Action	IEC 104 data are converted to OPC and further to Matlab data structures.
Time period	The same as input signal.
Time synchronisation	NA
DWH interaction	/

4 Demo site implementation overview

This chapter describes Demo site from implementation point of view, covering information technologies and solutions, and communication networking technologies and solutions. Both comprise the implementation of the whole system. It is situated in secure ELES environment, with physically restricted access and secured communication links via DMZ.

4.1 Information layer

Information layer comprises several software and hardware solutions that take care of algorithm signal processing, data transport and data transformation (Figure 5).

Real time LFC data streams from ELES and other TSOs enter demo site via ICCP/OPC gateway that translates originating ICCP protocol to OPC DA. This is implemented on existing Sisco AX-S4ICCP server.

Algorithm processing of all LFC controller extensions is implemented as a Matlab/Simulink model running on a dedicated Windows server. It has to support connection to OPC DA (OPC Toolbox). Connecting to external MQTT client or broker is realised with additional code using raw TCPsockets. User interacts through classical Matlab user interface.

Real time exchange of LFC extension signals to cloud platform is via MQTT gateway, with separate logical queues for each TSO's LFC extension block. This code runs on Linux server.

BSP connects measurement and control data via IEC 104/OPC gateway (Matrikon) [Matr 2016] that runs on a dedicated Windows server. Alternative possibility is to connect the same data to MQTT broker (Mosquitto [Mosq 2016]) if BSP can communicate natively with MQTT.

Matlab environment connects to Linux server that realises also CZC file exchange and foremost links Matlab signals, control setpoints for BSP, towards OPC/IEC 104 gateway. The same server hosts also database for historical signals used during Matlab runs (some are sources from ELES, others are derived from past real time data flows with different time stamps and stored locally on Demo site), lists of local MOL and acts as a general historian of data exchanges. HMI user interface is classical Linux console.

BSP bids are accepted over REST web services that are implemented on an Apache Tomcat server [Apac 2016]. The bids are stored in local database, and are as well forwarded to web service client towards FutureFlow Cloud platform.

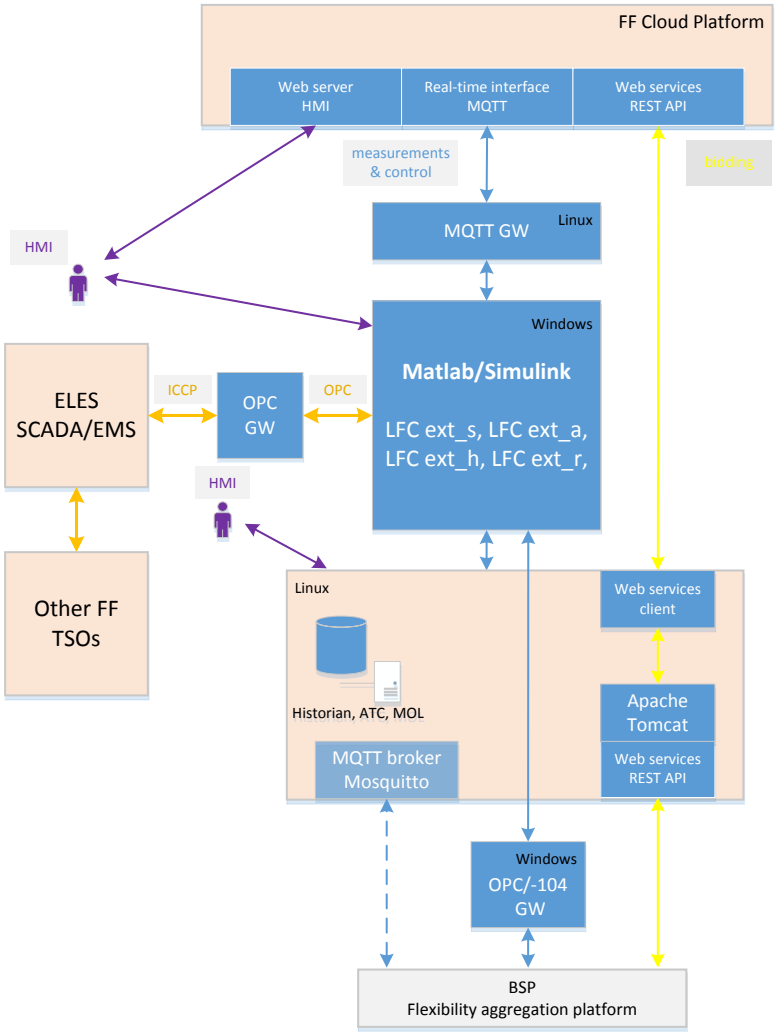


Figure 5: Information architecture of demo site.

4.2 Communications layer

Communications layer realises physical interconnection of IP paths with required security solutions (Figure 6). Demo site resides in existing DMZ secure TSO (ELES) environment. All Internet connections are secured with IPSec VPN or TLS connection. IPSec tunnel realises per site secure information tunnelling, whereas TLS offers per application type secure encrypted access. TLS Ver1.2 with SHA-2 hashing is mandatory [RFC5246 2008]. For security reasons MQTT must run always on TLS enabled routers/firewalls.

Access to data exchange with TSO measurements (ICCP/OPC gateway) is via TSO internal IP addresses (unrouted) and firewall. Only TCP ports needed for OPC are opened by default and restriction on other opened ports is enforced.

BSP access to demo site is over Internet. A dedicated DMZ and firewall and public address IP entry point is created with restricted port access. The following TCP ports are

opened by default: 2404 for IEC 104, 8080 for web services running on Tomcat server and optionally 1883, 8883 for MQTT (only if implemented). Some REST services may require additional opened ports, that can be pushed beyond port number 8080.

Cloud platform connection to demo site is over Internet. A dedicated DMZ and firewall and public address IP entry point is created with restricted port access. The following TCP ports are opened by default: 8080 for web services, 1883, 8883 for MQTT.

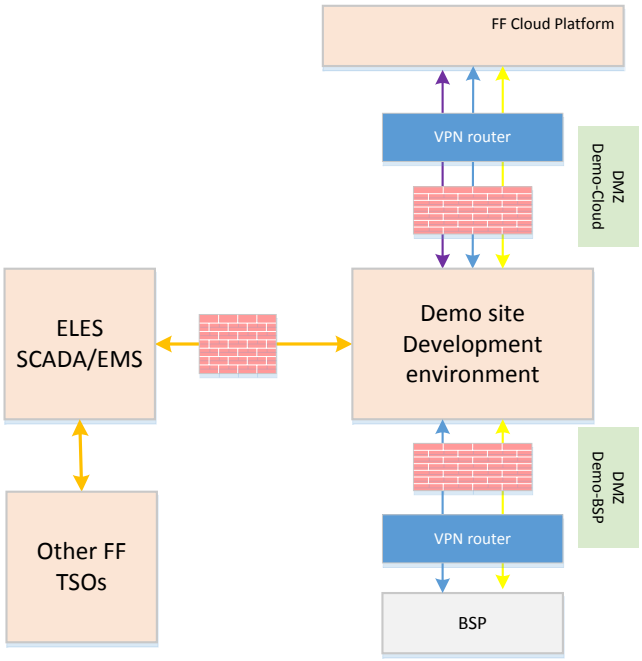


Figure 6: Communication architecture of demo site.

5 Conclusions

Demo site and development environment implement the functionalities of common real time processing function in relation to the aFRR balancing process.

The core functionality of Demo site and development environment is to host fully functional LFC (LFC extension) and modelled response of aFRR units for each TSO. Development environment is realised in Matlab/Simulink that implements control algorithms (PI and filtering) for each TSO in a separate, fully parametrizable building block. Real time data from SCADA/EMS enables real time realisation of pilot tests. Use of historical data streams or a mix of real time and historical is possible during functional test phases and selected pilot runs.

FF BSP measurements, control and bids are connected via IEC-104 and Web services, respectively to the same demo site. Data exchange of real-time signals (outputs, like P-demand, or inputs assimilated to responses, like correction signal, bid activations, local merit order list, effective and used cross-zonal capacities) with FF Cloud platform is done via MQTT messaging protocol.

Communication security measures are taken into consideration on Internet connected BSP and cloud entities, whereas TSO integration is already within existing DMZ.

Demo site is being designed as a flexible prototyping platform, enabling efficient user interaction both on IT/communication as well as on LFC simulation and control layer. Data processed or generated in Demo site is stored in FF main data warehouse located within FF Cloud. Data access rights are defined in a Data Management Plan deliverable (D 7.4).

6 References

[Apac 2016] http://tomcat.apache.org/, 2016.

[DtJ 2016] Mozzila, https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date/toJSON , 2016.

[ECA 2011] Capacity Allocation and Nomination System (ECAN), ENTSO-E, 2011.

[ERR 2013] Reserve Resource Process ERRP Version 5 Release 0, ENTSO-E, 2013.

[ESS 2012] Scheduling System ESS - Version 4 Release 1, ENTSO-E, 2012.

[IEC301 2013] IEC 61970-301 Energy management system application program interface (EMS-API) - Part 301: Common information model (CIM) base, IEC, 2013.

[IEC559 2015] IEC 62559-2 Use case methodology - Part 2: Definition of the templates for use cases, actor list and requirements list, IEC, 2015.

[IEC870 2014] IEC 60870-6-503 Telecontrol equipment and systems - Part 6-503: Telecontrol protocols compatible with ISO standards and ITU-T recommendations - TASE.2, IEC, 2014.

[ISO8601 2004] Data elements and interchange formats, Information interchange, Representation of dates and times, ISO, 2004.

[ISO922 2016] ISO/IEC 20922 Information technology, Message Queuing Telemetry Transport (MQTT) v3.1.1, ISO, 2016.

[JSON 2013] The JSON Data Interchange Format, ECMA, 2013.

[Matr 2016] http://www.matrikonopc.com/drivers/driver-types.aspx, 2016.

[Mosq 2016] https://mosquitto.org/, 2016.

[MQTT 2014] MQTT Version 3.1.1, OASIS, 2014.

[RAML 2016] The simplest way to design APIs, http://raml.org/.

[RFC3339 2002] RFC3339, Date and Time on the Internet: Timestamps, 2002

[RFC5246 2008] RFC5246, The Transport Layer Security (TLS) Protocol Version 1.2, 2008.

[SOAP 2007] SOAP Version 1.2 Part 1: Messaging Framework, W3C, 2007.

7 Appendix 1: Control Option1, Option2 structures

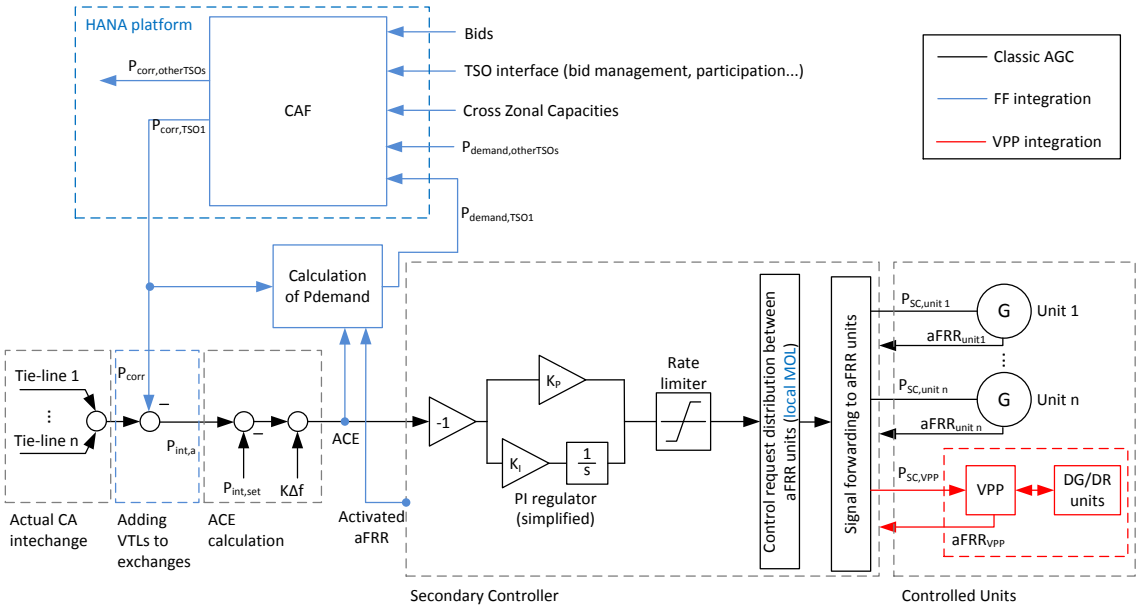


Figure 7: Control Option1 LFC controller structure.

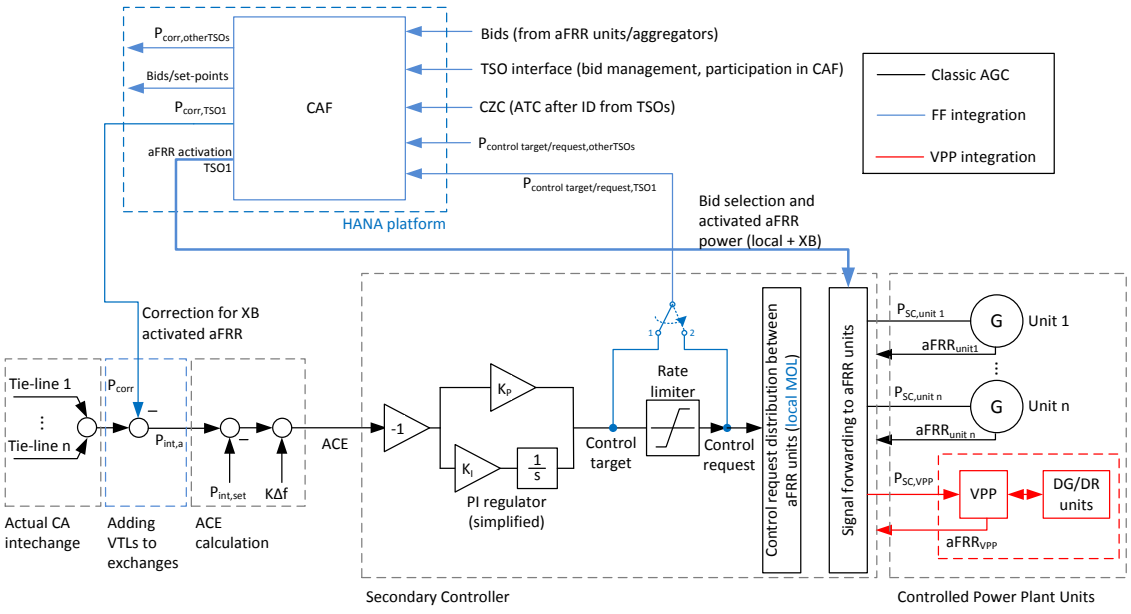


Figure 8: Control Option2 LFC controller structure.

8 Appendix 2: LFC extension platform integration/function