

Title:	Document Version:
D13.1 WiseGRID Cockpit Design	1.0

Project Number:	Project Acronym:	Project Title:
H2020-731205	WiseGRID	Wide scale demonstration of Integrated Solutions for European Smart GRID

Contractual Delivery Date:	Actual Delivery Date:	Deliverable Type*-Security*:
M18 (April 2018)	M18 (April 2018)	R-PU

*Type: P: Prototype; R: Report; D: Demonstrator; O: Other.

**Security Class: PU: Public; PP: Restricted to other programme participants (including the Commission); RE: Restricted to a group defined by the consortium (including the Commission); CO: Confidential, only for members of the consortium (including the Commission).

Responsible:	Organisation:	Contributing WP:
Alberto Zambrano	ETRA	WP13
Álvaro Nofuentes	ETRA	WP13

Authors (organization):
Diego García-Casarrubios Gálvez (ETRA), Álvaro Nofuentes (ETRA), Alberto Zambrano (ETRA), Foivos Palaiogiannis (ICCS), Ioannis Vlachos (ICCS), Georgina Asimakopoulou (ICCS), Panos Karafotis (ICCS), Stela Sarri (ICCS), Leandro Lombardo (ENG), Giusi Caruso (ENG), Mihai Sanduleac (CRE), Catalin Chimirel (CRE), Paul Lacatus (CRE), Julio César Díaz (ITE), Juan Pablo González (ITE), Amparo Mocholí (ITE), Javier Monreal (ITE)

Abstract:
This document reports the work performed within Task 13.1 “WiseGRID Cockpit Design” in terms of functionalities offered and design of the solution for small DSOs and microgrid operators. This framework takes as starting point the work done in the requirement definition, use case identification, and preliminary architectural design.

Keywords:
Smart Grid, DSO, Cockpit, Distribution Grid, MV, LV

Revision History

Revision	Date	Description	Author (Organisation)
V0.0	01.03.2018	New document	Álvaro Nofuentes (ETRA)
V0.1	14.03.2018	First round of contributions	ETRA, ICCS, ENG, ITE
V0.2	29.03.2018	Second round of contributions	ETRA, ICCS, ENG, ITE, CRE
V0.3	13.04.2018	Last contributions	ETRA, ICCS
V0.4	16.04.2018	Version ready for peer review	Alberto Zambrano (ETRA) and Álvaro Nofuentes (ETRA)
V0.5	23.04.2018	Integration of feedback received	ETRA, ITE, CRE
V1.0	25.04.2018	Final refinement and ready for submission	Álvaro Nofuentes (ETRA)

INDEX

EXECUTIVE SUMMARY	8
1 INTRODUCTION	10
1.1 PURPOSE OF THE DOCUMENT.....	10
1.2 SCOPE OF THE DOCUMENT	10
1.3 STRUCTURE OF THE DOCUMENT	10
2 BACKGROUND	11
2.1 Overview	11
2.2 Relevant actors	11
2.3 Requirements	13
2.4 High Level use cases	16
2.5 Primary use cases.....	16
2.6 Architecture.....	17
3 WISEGRID COCKPIT MODULES	20
3.1 Internal Service Bus.....	20
3.1.1 Context	20
3.1.2 Messaging mechanism	21
3.1.3 Internal Enterprise Service Bus.....	22
3.1.4 RabbitMQ	22
3.1.5 Common communication patterns enabled by the ESB.....	23
3.1.6 Integration with modules	23
3.2 Real-Time Monitoring.....	24
3.2.1 Unbundled Smart Meters Wrapper (SMX Wrapper).....	24
3.2.2 AMI Wrapper	25
3.2.3 SCADA Wrapper	26
3.2.4 Real-time monitor	28
3.2.5 KPI engine	29
3.3 Off-Line processes	30
3.3.1 Topology processor	30
3.3.2 GIS provider	35
3.3.3 Power Quality	44
3.3.4 Grid planning assistant	47
3.3.5 Energy Management System for Non-Interconnected Systems	49

3.3.6 Load Demand and Peak Prediction.....	53
3.4 Data analysis, operation and control.....	55
3.4.1 Threshold monitor.....	55
3.4.2 Outlier detector	56
3.4.3 Congestion forecast.....	57
3.4.4 Grid fault manager.....	60
3.4.5 Ancillary services market hub.....	67
3.4.6 Demand and production forecast services.....	69
3.4.7 Maintenance manager	75
3.4.8 Power flow calculator/state estimator.....	78
3.5 User Interface	81
3.5.1 User profiles.....	82
3.5.2 Sections.....	86
4 LINK WITH OTHER WISEGRID APPLICATIONS	104
5 DPIA CONSIDERATIONS.....	105
6 CONCLUSIONS	106
7 REFERENCES AND ACRONYMS.....	107
7.1 References	107
7.2 Acronyms.....	108
8 ANNEXES	110
8.1 Swagger specifications of the Maintenance Manager	110

List of figures

Figure 1 – WiseGRID Cockpit	11
Figure 2 – Overview of offline and commissioning modules	17
Figure 3 – Overview of data ingestion modules	18
Figure 4 – Overview of data analysis, operation and control.....	19
Figure 5 – Monolithic and microservice architecture comparison [1]	21
Figure 6 – Example of different communication patterns [2]	22
Figure 7 – RabbitMQ [3]	23
Figure 8 – Publication of data from USM into the WiseGRID IOP	25
Figure 9 – Translation to CIM Common Data Model.....	25
Figure 10 – Approach for integration of AMI data	26
Figure 11 – Overview of generic approach for SCADA integration	26
Figure 12 – Overview of integration with ENERCOOP's SCADA	27
Figure 13 – Overview of the structure of the RT monitor	29
Figure 14 – Structure of the KPI calculator modules.....	30
Figure 15 – Stencil of the Topology processor module.....	31
Figure 16 – Sample grid with component details.....	31
Figure 17 – Topology tool structure	32
Figure 18 – Formats provided by Data Transformation Service (DTS)	33
Figure 19 – Sample of the Intermediate Format (IMF).	34
Figure 20 – Screenshot of ENERCOOP's GIS database	43
Figure 21 – ENERCOOP's GIS provider structure.....	43
Figure 22 – ASM's GIS provider structure	44
Figure 23 – Input data (RMS values) from database.	45
Figure 24 - Input data from user.	45
Figure 25 – Voltage dips classification.....	46
Figure 26 – Components of the Grid Planning Assistant.....	47
Figure 27 – Screenshot of the simulator for optimizing Multi-Objective Energy Systems [12], basis of the Grid planning assistant	48
Figure 28 – EMS for non-interconnected islands, sample of the Web based UI.....	50
Figure 29 – EMS for non-interconnected islands, sample of the Thermal Generators.....	51
Figure 30 – EMS for non-interconnected islands, sample of historical data.....	52
Figure 31 – EMS for non-interconnected islands, sample of one output file with detailed results of the Unit Commitment module.	53
Figure 32 – Sample of the input data in JSON format for the Load Demand and Peak Prediction Module.	54
Figure 36 – Data flows subscribed by the Threshold Monitor module	55

Figure 33 – Data flows subscribed by the Outlier Detector module	56
Figure 34 – The workflow of the CMAP module responsible for constraints management.....	58
Figure 35 – Time flow	59
Figure 37 – Example of a simple BPMN workflow.....	60
Figure 38 – Integration of alerting modules with the Failure Treatment Module.....	62
Figure 39 – Functional diagram	63
Figure 40 – Flexibility request.....	68
Figure 41 – Flexibility offer reception.....	68
Figure 42 – Flexibility offer selection and flexibility order	69
Figure 43 – Forecast server structure integration schema	70
Figure 44 – Screenshot of the Maintenance Module Web Interface.....	76
Figure 45 – Screenshot of the Maintenance Module Android App	76
Figure 46 – Sample of the input file of 3PPF - Defining the scenario data.....	79
Figure 47 – Sample of input file of 3PPF – Defining the topological data (branch data).	80
Figure 48 – Sample of input file of SE – Defining the connectivity of network buses, as well as the type, configuration, and length of the lines.	81
Figure 49 – Sample of input file of SE – Defining the line parameters (resistance, reactance, susceptance).	81
Figure 50 – Dashboard.....	86
Figure 51 – Dashboard (maintenance)	87
Figure 52 – Map (status view)	88
Figure 53 – Map (topology view).....	89
Figure 54 – Single-line diagram	90
Figure 55 – AMR (general information).....	91
Figure 56 – AMR (associated elements)	92
Figure 57 – AMR (measures)	93
Figure 58 – Grid maintenance (phone calls).....	94
Figure 59 – Grid maintenance (phone call registration)	95
Figure 60 – Grid Maintenance (incidents)	96
Figure 61 – Grid maintenance (incident details, general info.)	96
Figure 62 – Grid maintenance (incident details, calls)	97
Figure 63 – Grid maintenance (incident details, works).....	98
Figure 64 – Grid maintenance (new incident registration)	99
Figure 65 – Summary of works	100
Figure 66 – Work details	100
Figure 67 – Overview of field crew	101
Figure 68 – Overview of Demand response campaigns	102

Figure 69 – Demand Response campaign details.....	103
Figure 70 – Self-healing strategies	104
Figure 71 – WG Cockpit interactions.....	105

List of Tables

Table 1 – WG Cockpit actor’s inventory	11
Table 2 – WiseGRID Cockpit requirements	13
Table 3 – Key aspects topology	35
Table 4 – Medium voltage lines GIS model	35
Table 5 – Low voltage lines GIS model	36
Table 6 – Substations GIS model	37
Table 7 – Usage point location GIS model.....	39
Table 8 – Usage point GIS model.....	40
Table 9 – RES GIS model	42
Table 10 – Datasheet for power quality analysis	45
Table 11 – Key aspects power quality module	47
Table 12 – Inputs required by the Grid Planning Assistant	48
Table 13 – Outcomes of the Grid Planning Assistant	49
Table 14 – Key aspects.....	50
Table 15 – Load demand and peak prediction output	54
Table 16 – Inputs required	64
Table 17 – Outputs	64
Table 18 – Inputs	64
Table 19 – Outputs	65
Table 20 – Operational parameters.....	66
Table 21 – User associated parameter	66
Table 22 – Demand forecast API, error codes.....	71
Table 23 – Production forecast API, error codes.....	73
Table 24 – Summary of methods.....	77
Table 25 – Workflow	77
Table 26 – Data formats	80
Table 27 – Data formats	81
Table 28 – User profiles considered for WiseGRID Cockpit	83
Table 29 – Threat and feared events identification for WG Cockpit.....	105
Table 30 – Acronyms list.....	108

EXECUTIVE SUMMARY

The current and future distribution networks have to face the fact that there is no more a unidirectional power flow in their lines. The increasing of the installed power within Distributed Energy Resources provokes the appearance of potential problems in the grid (overvoltage, congestion, imbalance...). It is also noteworthy that the deployment of more and more EVs on European roads will mean an increase in the electricity demand as never seen before.

Moreover, the increasing share of Renewable Energy Sources in the European Power Systems implies the need of more accurate forecasting tools to better calculate the energy that these sources will deliver. The intermittence and uncertainty of the weather phenomena which mobilise these energy sources means an increase of the uncertainties for network operation purposes.

In order to avoid these potential problems, some strategies are being developed during the last years. These strategies are mainly focused not only on reducing the operating problems of the DER and RES but even in taking advantage of these kind of sources and use them to improve safety and secure operation of the distribution networks. One of the strategies with highest potential is the performing of Demand Response (DR) campaigns. Thanks to the modulation of the energy consumption of small and medium users, it is possible to produce beneficial services to the distribution grid. Also the increase of storage solutions such as domestic or industrial batteries and the batteries of the EVs (thanks to V2G activities), can be used to improve the quality of the MV and LV grids.

These activities must go hand in hand with the development of new technologies such as smart meters, power electronics, big data analytics, or the establishment of new algorithms that helps to assure the reliability of the grid.

WiseGRID directly tackles these facts with the development of WG Cockpit, a tool addressed to small DSOs and microgrid operators. This product will help them to better control and monitor their MV and LV networks without forgetting the decarbonization goals of the European Commission and also taking advantage of the technological and conceptual improvements above mentioned.

The design of WG Cockpit has been done taking into account the specifications, the requirements and the expected test scenarios envisaged in previous stages of the project. The architecture and the modules reflected in this deliverable assure the performance of the expected characteristics of the tool and their synergic work with the other tools of the WiseGRID framework. In addition, the design of the communication of the internal modules (based on RabbitMQ) has been done in order to assure the proper interaction between the different functionalities of the tool.

In order to better explain the modules of the tools, they have been clustered into different categories: Real-Time Monitoring, Off-line processes, Problem detection and reaction and User interface.

Real-Time monitoring

This section deals with the modules which collect data from different meter sources (such as Unbundled Smart Meters, AMIs or SCADAs). Inside this section monitors all these data sources and provide the necessary KPI calculations in order to correctly assess the performance of the tool.

Off-Line processes

In this cluster are included the tools which do not need permanently Real-Time data to work. These are the modules dealing with the topology and geographical information of the monitored grid, a power quality assessment of the distribution lines and an assistant for planning future grid improvements.

Problem detection and reaction

These modules are focused on the control of the distribution grid. These modules will detect the possible problems of the network (incidents) and will establish the best strategies that will solve those incidents.

User Interface

The WG Cockpit User Interface will be the interaction point with the DSO operator. It must therefore show all relevant information compiled by all other modules present in the architecture of the application.



1 INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

The present document aims to describe and explain the results acquired from the Task 13.1 “WiseGRID Cockpit Design”. This task mainly deals with the design of the modules and functionalities of the WG Cockpit tool which is going to be used by small DSOs and Microgrid operators. Those specifications are explained with technical and accurate language but being comprehensive to several types of readers.

The functionalities and capabilities of this tool have been developed following the requirements and the use cases that have been set in the D2.1 “WiseGRID requirements, use cases and Pilot Sites analysis” as it is possible to check within the document.

Moreover, the authors of this document have provided a univocal view of the tool but considering the relationships between it and the other WiseGRID tools having in mind that they share common modules and have internal dependencies. To reach this objective, the authors have shared and collected information from other deliverables in order to avoid readers to be aware of the work done in other WPs to totally understand this product.

However, the information included in this deliverable could be enhanced later during the development phase and those improvements will be reported in the deliverable D13.2 “WiseGRID Cockpit implementation and lab-testing”.

1.2 SCOPE OF THE DOCUMENT

The extent of this deliverable is focused on the needs that the WG Cockpit has to address in order to provide valuable functionalities to its users. The description of the different modules and the design activities have been addressed within the document leaving the deployment and implementation issues for upcoming deliverables which have to deal with those tasks.

The design provided in this document could be seen as the starting point for the further implementation of the WG Cockpit tool, by taking into account all the foundation documents provided previously in WiseGRID.

1.3 STRUCTURE OF THE DOCUMENT

Apart from this introductory section, the current document is structured as follows:

Firstly, it is explained the background needed to collate and understand the previous work done in other WPs related to WG Cockpit. For that purpose, the authors have included the required information from other deliverables. This section provides the basis for the future sections of the document.

Following that section, the document presents the overall modular structure and describes the different modules of the WG Cockpit tool and how these modules address the required functionalities of the product. In order to give to the reader a high-level view of the tool, the modules have been clustered into different categories and thus a better understanding of the tool is achieved. This section could be seen as the core part of the document and the most technical part of it.

Once the deliverable has explained WG Cockpit as a single product, the authors give to the reader a high-level view of the position of the tool inside the WiseGRID framework and some considerations about data protection.

Finally, the document ends with a brief summary and the exposition of the main conclusions that can be extracted from the whole document.

2 BACKGROUND

2.1 OVERVIEW

As explained in the Deliverable 2.1, the WG Cockpit is the WiseGRID technological product targeting small DSOs and micro grid operators, allowing them to control, manage and monitor their own network, using flexibility to improve stability and security of their network. Taking into account the goals of the project, the features to be implemented within WG Cockpit consider a scenario of increasing share of distributed energy resources (DER) and services provided by communities of prosumers (aggregated in the form of VPPs or co-operatives) in order to achieve higher participation and environmental, social and economic benefits.

The main purpose of the WG Cockpit is to enable grid operators to manage the fundamental changes that distribution grids are facing nowadays, some remarkable ones of those being the transition towards a grid with high penetration of distributed renewable energy resources and the presence of additional significant loads coming from EVs among others. In addition, this particular outcome of the project aims at approaching the benefits that new technologies (such as big data or unbundled smart meters) and algorithms (such as state estimation or fault detection) bring to the operation of the grid. Finally, since one of the objectives of the project is the empowerment of the citizens in the energy field, the WG Cockpit will also demonstrate how that empowerment can be beneficial for several actors – including DSOs –, and how the whole ecosystem of actors can contribute to reach an environmentally and economically sustainable energy system.



Figure 1 – WiseGRID Cockpit

2.2 RELEVANT ACTORS

During the first year of the project, the consortium has identified who are the main actors that participates in the WiseGRID framework. In this subsection will be shown and explained who are the actors (whatever the type: organization, system, person or device) that are included in the WG Cockpit framework. The consortium has identified 23 actors that can be directly linked with the WG Cockpit.

Table 1 – WG Cockpit actor's inventory

Actor name	Description	Actor type
Aggregator	Accumulates flexibility from Prosumers and Consumers and sells it to the Supplier, the DSO or the TSO.	Organization
AMI	<i>Advanced Metering Infrastructure</i> . A set of systems that monitor, collect and analyse electricity consumption, and have two-way communication capabilities.	System
Balance Responsible Party	A party that has a contract proving financial security and identifying balance responsibility with the Market Operator entitling the party to operate in the market. [The meaning of the word "balance" in this context signifies that the	Organization

Actor name	Description	Actor type
	quantity contracted to provide or to consume must be equal to the quantity really provided or consumed.]	
Battery Operator	An entity responsible for operating a set of Storage Units connected to the electricity grid.	Organization
Data Provider	Independent entity responsible for undertaking and coordinating the information exchange and translation of the data of various sources into a common data model.	Organization
Distributed Energy Resource	Any type of generation units, storage units and load flexibility resources connected to the distribution network.	System
DMS	<i>Distribution Management System</i> . A system that monitors, controls and analyses in real-time or near real-time the electricity distribution system.	System
DSO	<i>Distribution System Operator</i> . The entity responsible for: the distribution network planning and development; the safe and secure operation and management of the distribution system; for data management associated with the use of the distribution system; for procurement of flexibility services.	Organization
Electronic Meter	A physical device containing one or more registers.	Device
Energy Management System	A system that monitors, controls and optimizes the operation of the energy system under supervision.	System
EVSE Operator	<i>Electric Vehicle Supply Equipment Operator</i> . The entity responsible for managing and operating the EV charging infrastructure.	Organization
Forecast Provider	The organization that provides, upon demand, forecasts regarding certain variables (e.g. electricity demand, RES production, weather conditions, etc.)	Organization
GIS	<i>Geographical Information System</i> .	System
Harbour Operator	An organization that manages and operates the harbour infrastructure.	Organization
Market Operator	The unique power exchange of trades for the actual delivery of energy that receives the bids from the Balance Responsible Parties that have a contract to bid. Determines the market energy price taking into account the technical constraints from the Transmission System Operator.	Organization
PDC	<i>Phasor Data Concentrator</i> . Receives and time-synchronizes phasor data from multiple phasor measurement units (PMUs) to produce a real-time, time-aligned output data stream.	Device
Producer	An entity connected to the grid that injects electricity to the grid.	Person
RES Unit	<i>Renewable Energy Source Unit</i> . A type of Producer that transforms energy from renewable energy sources (e.g. sun, wind, etc.) to electricity and injects it to the grid.	Device
SCADA	<i>Supervisory Control And Data Acquisition system</i>	Device
Smart Meter	An Electronic Meter with two-way communication capabilities.	Device
Supplier	Supplies and invoices energy to its customers.	Organization
TSO	<i>Transmission System Operator</i> . A party that is responsible for a stable power system operation (including the organisation of physical balance) through a transmission grid in a geographical area. The System Operator will also	Organization

Actor name	Description	Actor type
	determine and be responsible for cross border capacity and exchanges. If necessary, he may reduce allocated capacity to ensure operational stability.	
VPP Operator	<i>Virtual Power Plant Operator.</i>	Organization

2.3 REQUIREMENTS

In order to be sure that the features and characteristics of the WiseGRID technological solutions meet the project goals, the consortium created a list of requirements that each partner agreed. The complete list of requirements can be found in Deliverable 2.1, however, in this subsection will be shown the ones related with the WG Cockpit.

Table 2 – WiseGRID Cockpit requirements

Requirement ID	Description	Type
AMG_001	Bidirectional communication between DSO and involved energy stakeholders is established.	Functional and data requirements
AMG_002	Contractual agreements between the DSO and the rest of the actors are in place.	Legal requirements
AMG_003	The proprietor of the unit (if different from the DSO) has signed a contract with the DSO that describes the conditions that apply regarding the load control actions.	Legal requirements
AMG_004	Reliable calculation of customer response to control signals.	Functional and data requirements
AMG_005	Characteristics of load control actions (timeframe, triggering events) should be well-established.	Legal requirements
AMG_006	The resource operator has signed a contract with the DSO that describes the conditions that apply regarding the load control actions.	Legal requirements
AMG_007	The Prosumers, EVs and other local energy resources have signed a contract with the resource operator for load control actions.	Legal requirements
AMG_008	Actor's portfolio (characteristics) is known.	Functional and data requirements
AMG_009	The optimization algorithm is robust.	Functional and data requirements
AMG_010	The optimization algorithm produces reliable results.	Functional and data requirements
AMG_011	Adequate amount of pieces of equipment for monitoring the distribution network is installed.	Functional and data requirements
AMG_012	Regulation allows islanded operation of part of the distribution network.	Legal requirements
WCP_001	Smart meters shall be installed in enough ratio to allow proper measurements of the grid buses.	Operational requirements
WCP_002	Smart meters shall be installed in participant prosumers to allow proper consumption/production/power quality measurements.	Operational requirements
WCP_003	WiseGRID Cockpit must visualize all collected measurements of the grid.	Functional and data requirements
WCP_004	WiseGRID Cockpit must visualize KPIs of Power Quality-related metrics.	Functional and data requirements

Requirement ID	Description	Type
WCP_005	DER attached to the grid must include control interfaces.	Operational requirements
WCP_006	Ancillary services market must offer an open interface to post requests.	Operational requirements
WCP_007	VPPs and Storage as a Service (including DR, RES, EVs and batteries) must offer an open interface to post requests.	Operational requirements
WCP_008	WG Cockpit must provide an interface to inform fault detections to operators.	Functional and data requirements
WCP_009	WG Cockpit must provide operators with grid reconfiguration proposals in the event of (expected) faults.	Functional and data requirements
WCP_010	WG Cockpit must be able to automatically apply optimized grid reconfiguration.	Functional and data requirements
WCP_011	Switches available in the feeders must be controllable.	Operational requirements
WCP_012	The platform will be able to manage grid imbalance.	Functional and data requirements
WCP_013	WG Cockpit must perform grid state estimation.	Functional and data requirements
WCP_014	WG Cockpit will monitor the distribution network.	The scope of the product
WCP_015	WG Cockpit will improve the flexibility of the distribution network.	The scope of the product
WCP_016	WG Cockpit will enhance RES integration in the distribution network.	The scope of the product
WCP_017	WG Cockpit will detect faults in the distribution network.	The scope of the product
WCP_018	WG Cockpit will self-configure the distribution network.	The scope of the product
WCP_019	WG Cockpit will manage remotely the switching elements in the distribution network.	The scope of the product
WCP_020	WiseGRID should use open standards, if possible and define interfaces for all interconnected systems under WiseGRID Cockpit app.	The scope of the product
WCP_021	WG Cockpit will enable a connection bus between different various operational facilities and applications of the DSOs.	The scope of the product
WCP_022	WG Cockpit should be able to provide suggested solutions as best practises to faults in case of repetition, or in cases of grid reconfiguration.	Functional and data requirements
WCP_023	The topology of the grid at pilot sites must be known in advance.	Operational requirements
WCP_024	The developed fault location algorithms should work effectively on both underground and overhead networks.	Functional and data requirements
WCP_025	Real-Time or near Real-Time update of asset database/GIS.	Functional and data requirements
WCP_026	Bidirectional communication between installed devices and central control unit.	Functional and data requirements
WCP_027	Data transfer between different applications based on a common data model.	The scope of the product

Requirement ID	Description	Type
WCP_028	It should perform/provide the results of load flow analysis.	Operational requirements
WCP_029	It should provide information on the status of the asset and their maintenance to the DSO.	Functional and data requirements
WCP_030	WG Cockpit shall provide clear and effective visualization of the topology, so that the DSO can easily spot the whole affected area in case a fault happens at the grid.	Functional and data requirements
WCP_031	WG Cockpit must have an open interface with GIS (through IOP).	Functional and data requirements
WCP_032	WG Cockpit must support communication with DMS, through IOP.	Functional and data requirements
WCP_033	WG Cockpit should allow DSOs to generate curtailment warnings.	Functional and data requirements
WCP_034	WG Cockpit allows the active voltage control and the management of switching components in the network.	The scope of the product
WCP_035	WG Cockpit provides support for grid automation based on Unbundled Smart Meters measurements.	The scope of the product
WCP_036	WG Cockpit has to support grid operators to better manage and control the energy network.	The scope of the product
WCP_037	WG Cockpit provides SCADA support and DER dispatch services, for Smart Grid using real-time data (at the consumer level) and direct control of DERs.	The scope of the product
WCP_038	WG Cockpit provides energy quality monitoring service (e.g. voltage levels and voltage asymmetries, current asymmetries in low voltage, time of voltage, and failures at customer levels).	The scope of the product
WCP_039	WG Cockpit shows energy grid load, RES forecast, state estimation and load flow.	Functional and data requirements
WCP_040	WG Cockpit will support grid energy storage system (g-ESS) optimisation for grid capacity management. In this context, g-ESS will be DSO asset and shall be combined with SaaS (Storage as a Service) offered by market aggregators.	Functional and data requirements
WCP_041	WG Cockpit will support grid survey and will highlight critical nodes in terms of voltage and critical sections in terms of power flow.	Functional and data requirements
WCP_042	WG Cockpit will have suitable interface to energy market platform.	Functional and data requirements
WCP_043	WG Cockpit will have suitable interface to balancing market platform.	Functional and data requirements
WCP_044	WG Cockpit should perform/provide the results of short circuit calculations.	Functional and data requirements
WCP_045	WG Cockpit shall provide zoning of the grid with clear allocation of each RES, storage and/or dispatchable consumer. This would manage also the visualization of the topology, so that the DSO can easily evaluate actors implication within various operating	Functional and data requirements
WCP_046	WG Cockpit will have a refreshing period for data update to suit various functions availability.	Functional and data requirements

2.4 HIGH LEVEL USE CASES

In order to facilitate the development and the assessment of the WiseGRID solutions, the demonstrations will be conducted following 7 High Level Use Cases. The WG Cockpit is linked to all these HL-UCs and their descriptions are shown hereafter.

- HL-UC 1: DISTRIBUTED RES INTEGRATION IN THE GRID

To integrate and demonstrate the largest possible share of intermittent decentralized RESs, showing the services that will provide stable and secure grids in these circumstances, including avoiding curtailment.

- HL-UC 2: DECENTRALIZED GRID CONTROL AUTOMATION

To provide an intelligent distributed control for DSO in order to detect faults, self-protect and self-reconfigure the network in a robust way to restore the power system without the intervention of a central intelligence (self-healing)

- HL-UC 3: E-MOBILITY INTEGRATION IN THE GRID WITH V2G

E-mobility and electric transport systems integration into the network with the implementation of the V2G technology, in order to provide services to the grid, such as storage capacity.

- HL-UC-4: BATTERY STORAGE INTEGRATION AT SUBSTATION AND PROSUMER LEVEL

Use of storage systems integrated into the utilities and substations level in order to provide services to the grid, such as storage capacity.

- HL-UC 5: COGENERATION INTEGRATION IN PUBLIC BUILDINGS/HOUSING

Cogeneration integration in a public building or collective housing, using hot water thermal storage units. Heat can be optimized and can also be used power generated to heat.

- HL-UC 6: VPP TECHNICAL AND ECONOMIC FEASIBILITY

Technical and economic feasibility of the VPP concept using aggregated resources to achieving a higher efficiency; i.e., to deliver peak load electricity or load-aware power generation at short notice.

- HL-UC 7: CITIZENS EMPOWERMENT IN ENERGY MARKET AND REDUCTION OF ENERGY POVERTY

This HL-UC aims to bring the energy prosumers – residential and tertiary – closer to the energy markets in order to reduce their energy costs and improve market integration.

2.5 PRIMARY USE CASES

Simultaneously to the requirements' definition, a set of Primary Use Cases were defined in order to envisage which will be the main actions and scenarios to be tested in the project demonstration. The ones in which the WG Cockpit is the main tool are:

- HL-UC 1_PUC_1_Network monitoring
- HL-UC 1_PUC_2_Control strategies for reducing RES curtailment
- HL-UC 1_PUC_3_Voltage support and congestion management
- HL-UC 1_PUC_4_Grid planning analysis
- HL-UC 2_PUC_1_Distribution network real-time monitoring
- HL-UC 2_PUC_2_Real-time distribution system awareness
- HL-UC 2_PUC_3_Grid control
- HL-UC 3_PUC_3_EV charging management
- HL-UC 3_PUC_4_Interaction with the energy infrastructure

In Deliverable 2.1 it is possible to find further information about these Primary Use Cases.

2.6 ARCHITECTURE

Based on the requirements and list of Use Cases to be realized by the WiseGRID Cockpit application, the following modules have been defined:

Offline processes

A number of modules have been identified whose functionalities are needed either during the commissioning of the system, or which provide functionalities that do not rely in the treatment of data in real time. Those modules include:

- *Topology processor*, the software that translates the topological representation of the distribution grid and the technical characteristics of its elements into the Common Data Model that can be managed by WG Cockpit
- *GIS provider*, the software that translates GIS data from the original format managed by the DSO into the Common Data Model accepted by WG Cockpit
- *Grid planning assistant*, a module allowing the DSO to perform simulation of different scenarios and assess the possible issues met when considering different patterns for demand, production and flexibility in the distribution grid
- *Thermal and RES Planning*, a tool used for network planning, which analyses the capacity of an energy system in RES, and how this is influenced if thermal units are also considered. This tool provides investment decision support manager in the long-term period to the DSO

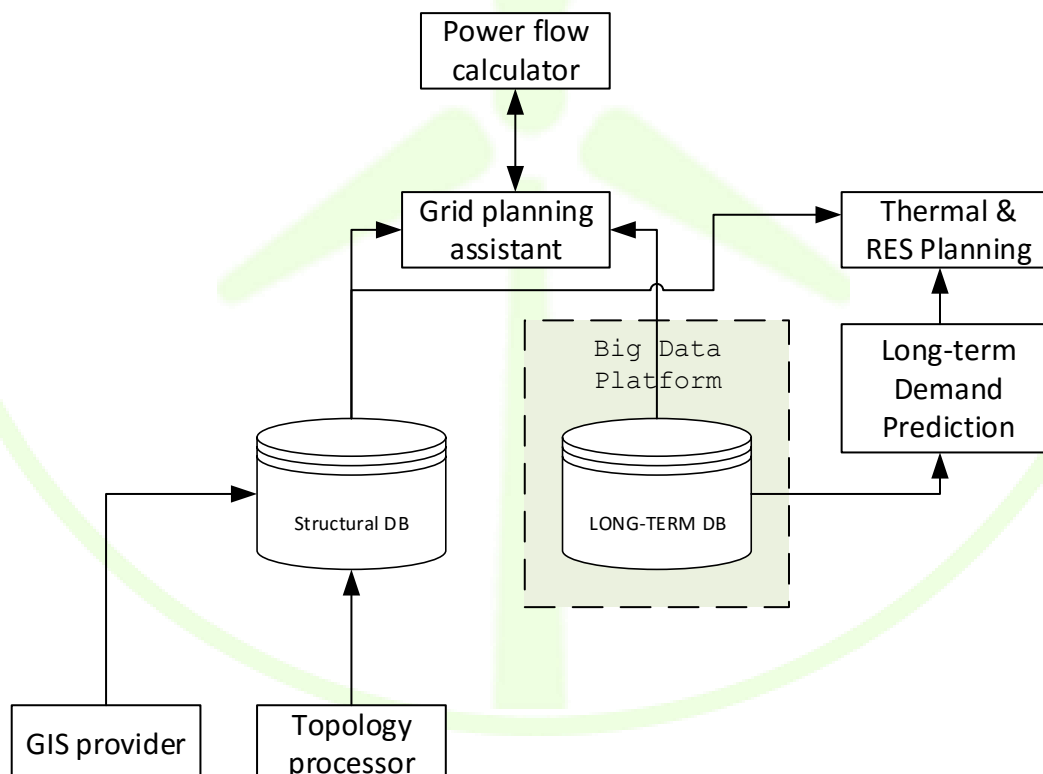


Figure 2 – Overview of offline and commissioning modules

Data ingestion

The first step considered in the design of the real-time features of the application is the data ingestion. The procedure followed is common to other applications in the project, and implies the following steps:

1. Publication of data from Wrappers to the WG IOP Message Broker. Following the principle taken in the overall project, data sources publish data to the Interoperable Platform, allowing different application with the corresponding permissions to access to those data flows
2. Subscription to data flows of interest. In the case of the WiseGRID Cockpit, the main data flows considered are those available by the DSO to monitor the distribution grid. Within the project, 3 different elements are considered: Advanced Metering Infrastructure systems, SCADA systems and Unbundled Smart Meters (SMX) devices. All three data sources will provide energy readings and electrical measurements on different points of the grid. Necessary software adaptations will be developed to allow those elements to publish data that can be used by the WiseGRID Cockpit to monitor the grid. This subscription to these data flows is performed by the *RT monitor* module
3. Store data for further analysis. The *RT monitor* module is in charge of populating both the *Operation* and the *Long-term DB* for further analysis
4. Analysis to increase observability of the grid. In the particular case of the WiseGRID Cockpit, Power Flow and State Estimation algorithms will be executed periodically to get an overview of the whole distribution network based on the available monitoring points.

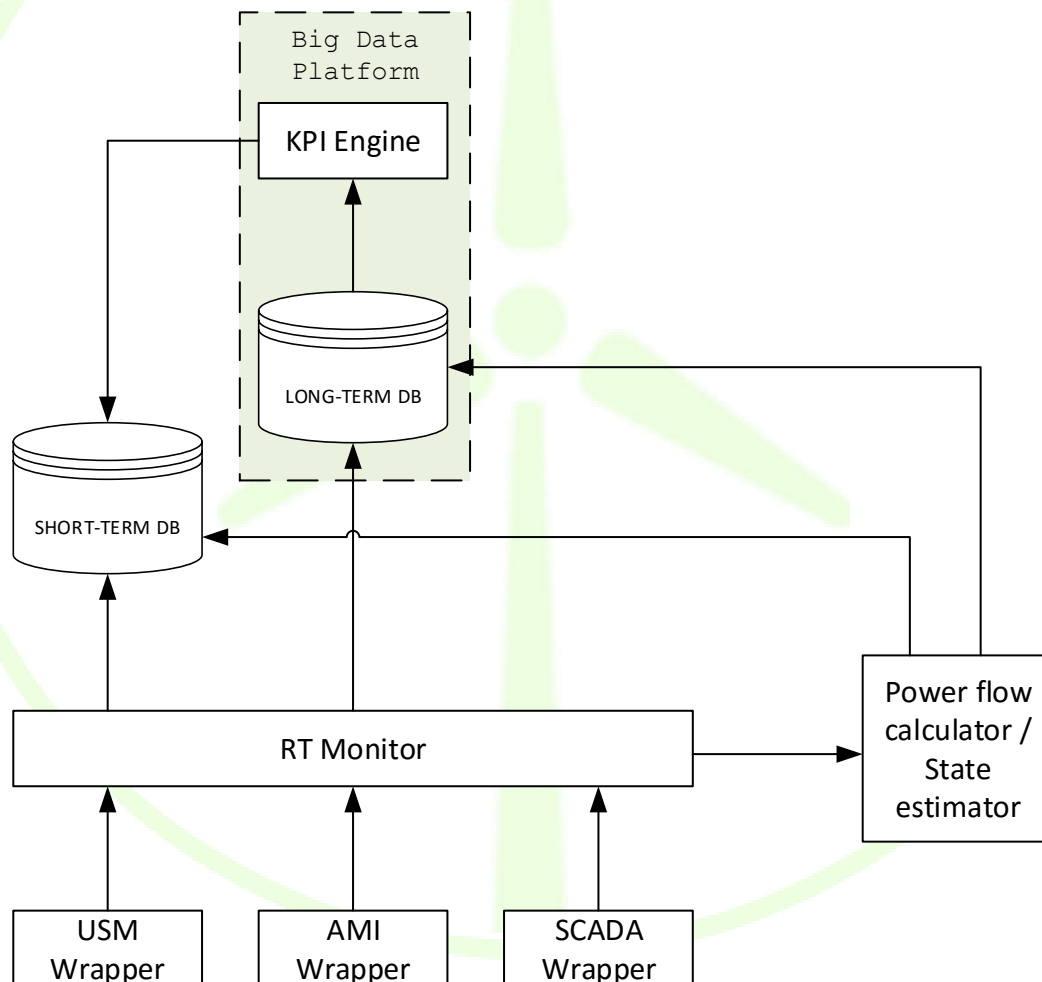


Figure 3 – Overview of data ingestion modules

Data analysis, operation and control

Under this group, different modules have been defined in order to process the raw data coming from the different data sources in order to get the relevant information out of those. The objective of those can be

either informative (obtain information of interest to the DSO operator), or action triggering (i.e. monitoring certain aspects of the grid and triggering specific workflows when certain situations are met). These modules include:

- *KPI engine* module, in charge of extracting different indicators and patterns from the raw data, focusing in energy demand of the prosumers supplied by the distribution grid and the frequency and reaction time of the DSO operator upon the detection of certain issues in the grid.
- *Demand and production forecast* module, providing forecasts for the energy demand of the customers of the DSO.
- *Threshold monitor* module, analysing the real-time flow of data to trigger alerts whenever operational conditions are exceeded.
- *Outlier detector* module, analysing the real-time flow of data to trigger alerts when certain parameters measured in the grid deviate from their statistical trend.
- *Congestion forecast* module, periodically executing an analysis algorithm that estimates the probability of congestion issues in the future.
- *Failure treatment* module, allowing the DSO operator to define custom workflows that will be automatically executed whenever a certain preconfigured condition is met and identified by the other modules of the WG Cockpit.
- *Unplanned Outage Treatment/FLISR* module, which will automatically calculate the optimum operations to recover from an outage in the distribution grid to get the minimum possible impact, also considering the implementation of automatic restoration of the supply.
- *Maintenance Manager* module, assisting the DSO operator to properly perform the preventive and corrective maintenance of the elements of the grid.

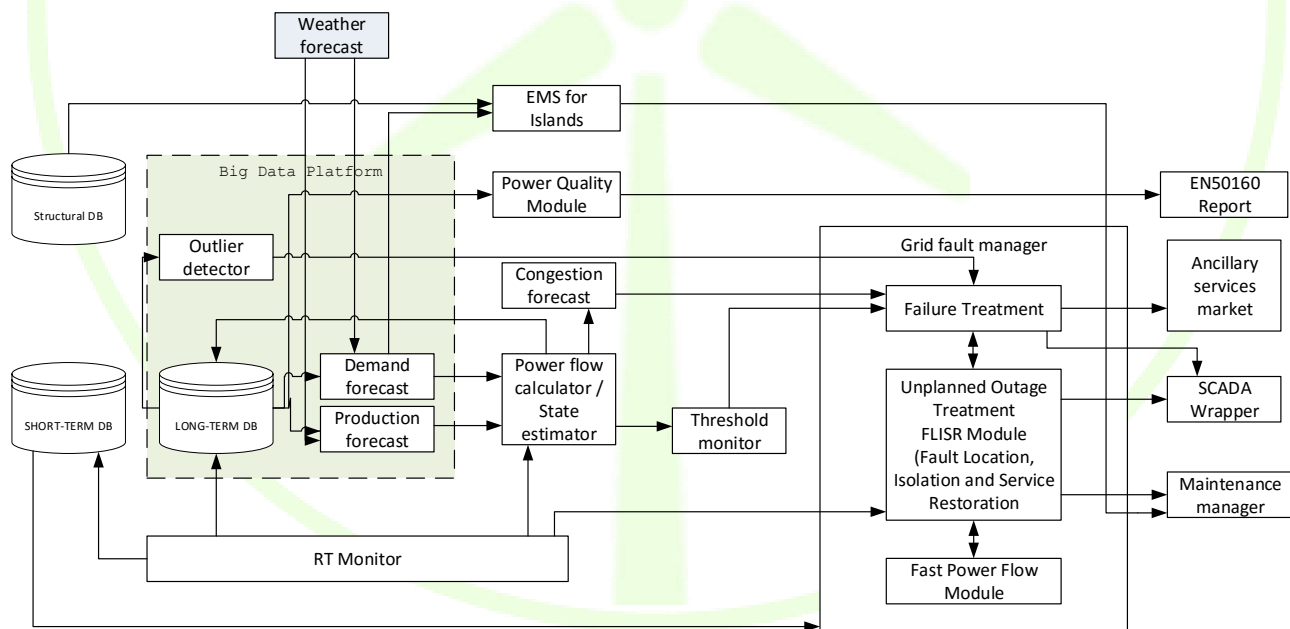


Figure 4 – Overview of data analysis, operation and control

Interaction with other applications

Within the WiseGRID project, a special focus is set in demonstrate how different kind of technologies can be beneficial to support the operation of the Smart Grid. These technologies will be addressed by different ap-

plications within the project, and will include electric vehicles, storage systems and other controllable demand assets. In this context, WiseGRID Cockpit, will interact with the different applications of the project addressing aggregators (WiseEVP, WG StaaS/VPP and WiseCOOP) through the *Ancillary Services Market*, in order to request support from those for assisting the correct operation of the distribution grid when required. Each application will evaluate up to which extent the corresponding technologies can assist the DSO by considering both the needs of the DSO, the business requirements of the aggregator and the particularities of the technologies and customers being aggregated.

Horizontal and support functionalities

Different modules will be used indirectly by the WiseGRID Cockpit application. Summarizing, these modules are data providers that offer information needed by other modules of the application to fulfil their duties, which are reused among different applications developed within the project. The list includes the *Weather Forecast* – whose information will assist the forecast modules - , *Energy mix provider* – whose data is used to assess the environmental impact of the used energy – and the *Big Data platform* that will support the long-term storage and analysis. Finally, the *WG Cockpit User Interface* is included in this category, providing web-based access to the information and functionalities provided by the other modules.

3 WISEGRID COCKPIT MODULES

3.1 INTERNAL SERVICE BUS

The architectural approach taken in the design of the applications of the project can benefit from the integration within the applications of the corresponding private internal service buses to handle the communication flows between the modules. This section presents the context leading to such decision and describes the adopted technology.

3.1.1 Context

3.1.1.1 The micro-services approach

As presented in the architectural view of WiseCOOP, the application itself is actually composed by a set of specialized software modules, each of those dealing with its own particular task. This design approach has the following main principles:

- A service-oriented architecture is composed of loosely coupled elements that have bounded contexts.
- Services can be updated independently without affecting the others.
- Database coupling is avoided as well, thus preventing side effects produced by parallel access (and possible modification) of different modules to the same data source.
- Services communicate with each other strictly through APIs. They don't share data structures, database schemas, or other internal representations of objects.
- Components are stateless: they don't store any information related to previous requests. An incoming request can be sent to any instance of the service.

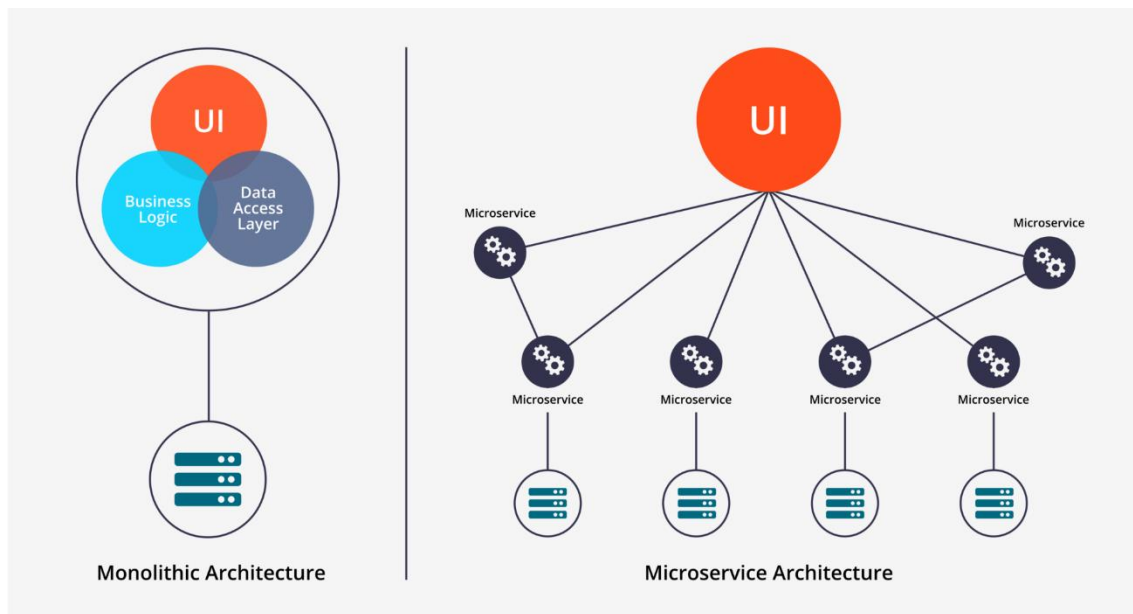


Figure 5 – Monolithic and microservice architecture comparison [1]

Dividing applications in a set of micro-services presents several advantages when compared to the monolithic approach, the most important ones being:

- Scalability: each one of the modules can be replicated in different machines in order to increase the throughput as required
- Graceful degradation: if one of the modules fails to perform its work, functions of the system not affected by the failure will continue working. Single points of failure are avoided as far as possible

3.1.2 Messaging mechanism

In order to handle communications between the different micro-services, a reliable messaging mechanism is needed to appropriately deliver the data flows established between the different modules. The following properties have been taken into account when selecting the proper platform to handle messaging:

- Easy setup: approaches allowing simple configuration of the micro-services are preferred
- Monitoring of communications: systems allowing to monitor communications are desirable, since those can significantly help identifying runtime problems, bottlenecks, etc.
- Asynchronous communication: overall system performance, scalability and maintainability can benefit from an asynchronous communication pattern between the different modules
- Durability of the messages: messaging approaches are seek that offer persistence of the messages in those cases when receiver is not able to process those (e.g. the process is down), and leverage the data source from handling destination unavailability. This feature simplifies the design this task and making it in order to successfully implement an asynchronous system, the messaging platform needs to offer the ability to persist undelivered messages until the receiver of those is ready to process them
- Policies for access control and filtering
- Scalability: one of the advantages of using a micro-service oriented approach is its possibilities to easily scale when workload limit is reached. The communication mechanism employed shall facilitate this task, and be able to scale together with the system as well
- Flexible delivery patterns: in order to optimize communications, the messaging mechanism shall allow different patterns, such as publish/subscribe, one-to-many delivery, etc.

- Support of different protocols: by supporting different communication protocols, flexibility is given to the system and to the design of the micro-services, even allowing that software modules using different protocols can communicate with each other

3.1.3 Internal Enterprise Service Bus

Taking into account the requirements just presented, an internal service bus has been selected as the most appropriate approach to handle the communications between the different modules within WiseGRID Cockpit.

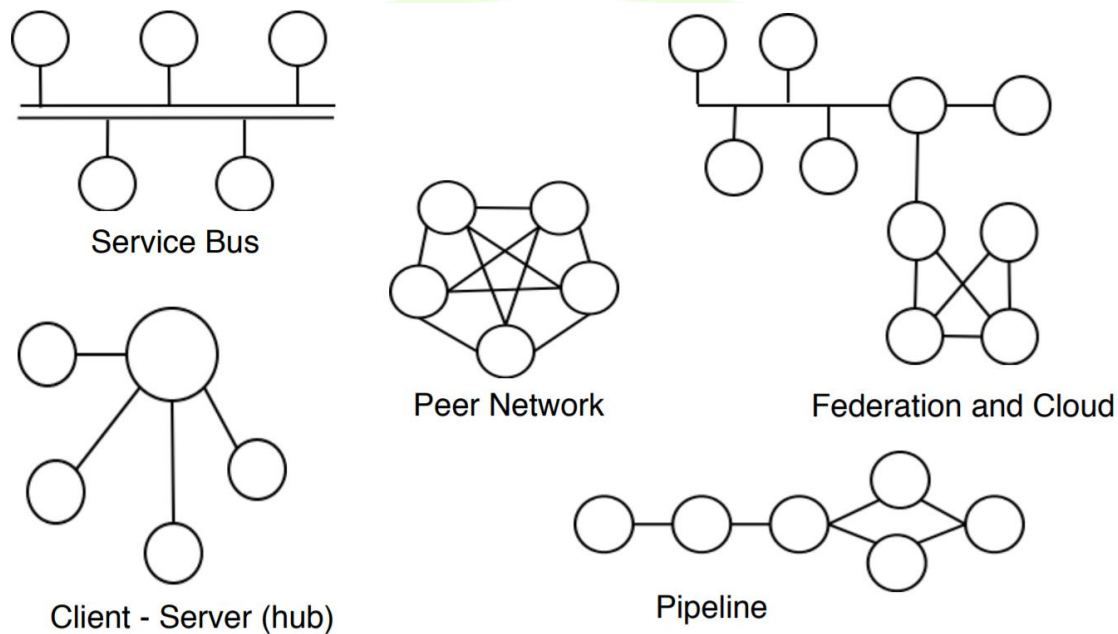


Figure 6 – Example of different communication patterns [2]

The use of an internal service bus presents the following advantages:

- Simplifies configuration of the modules: in a system with a relatively high number of modules/micro-services that interact among each other, direct communication may lead to complexity for maintaining overall configuration. By centralizing communications through a central system, configuration is simplified.
- Enables monitoring of communications between the different modules, making it easier to detect failures or bottlenecks in the system.
- Decouples data sources from data sinks, adding versatility to the design of the data flows.

3.1.4 RabbitMQ

RabbitMQ has been selected as the platform to implement an enterprise service bus within the application, since it features the set of desired characteristics exposed in the previous point. It also supports integration mechanisms (such as shovels – automatic delivery of messages to remote messaging platforms - and federation – automatic aggregation of messages from different sources into a single stream -) with different instances of RabbitMQ, thus enabling further integration possibilities with systems also using RabbitMQ, as the WiseGRID IOP message broker.

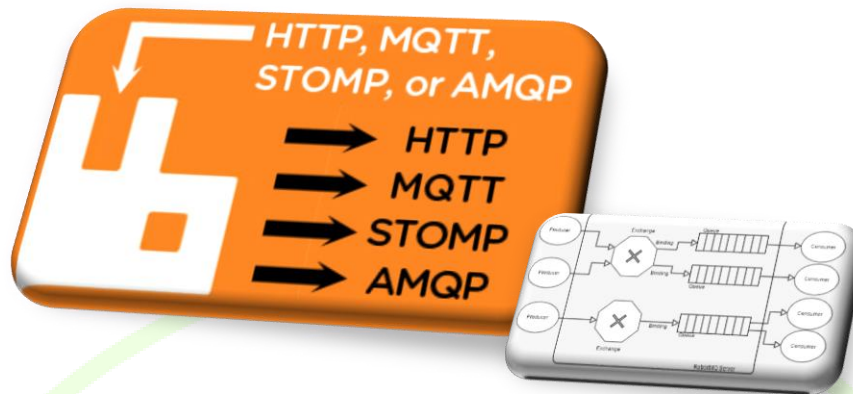


Figure 7 – RabbitMQ [3]

3.1.5 Common communication patterns enabled by the ESB

3.1.5.1 Publish-subscribe

Publish-subscribe is a pattern for asynchronous communication particularly suited for one-to-many, many-to-one and many-to-many communication flows scenarios, very typical in IoT applications.

This pattern defines three core elements:

- **Publisher:** is an element producing data that needs to be processed by a third party module.
- **Consumer:** is an element that is interested in processing data from different publishers; in order to do that, the consumer “subscribes” to certain topics which are published by other entities;
- **Broker:** the central element to the communication, which matches the information published by publishers with the information requirements of the consumers (subscribers), and delivers the messages appropriately.

3.1.5.2 Asynchronous RPC

Asynchronous RPC is a pattern enabling Remote Procedure Calls (one-to-one flows) while offering the advantages of asynchronous architectures. It defines three core elements:

- **Client:** needs to consume a service offered by a server, without blocking until the response is received.
- **Server:** offers a service, processing the incoming requests in order to produce a response.
- **Broker:** the central element that upon reception of a RPC requests, determines the server that is capable of fulfilling it, and stores it in work queue for that server. Server is continuously checking if there exist pending requests in that queue, and processing them. If server is stopped or under heavy load, the broker maintains the elements in the queue to be processed as soon as the server is available.

3.1.6 Integration with modules

In order to successfully integrate the modules with the internal service bus, those need to specify the following properties for each one of the communication flows they implement:

- **Protocol to be used:** the module needs to select between HTTP, MQTT or AMQP. RabbitMQ supports all three and provides interoperability among them. The main characteristics are:
 - **MQTT:** MQTT (MQ Telemetry Transport or Message Queuing Telemetry Transport) is an ISO standard (ISO/IEC PRF 20922) publish-subscribe-based messaging protocol. It works on top of the TCP/IP protocol. It is designed for connections with remote locations where a "small

code footprint" is required or the network bandwidth is limited, therefore very popular in the field of IoT sensors. It uses publish-subscribe messaging pattern.

- AMQP: The Advanced Message Queuing Protocol (AMQP) is an open standard application layer protocol for message-oriented middleware. The defining features of AMQP are message orientation, queuing, routing (including point-to-point and publish-and-subscribe), reliability and security.
- HTTP: The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, and hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web, and therefore also supported due to its availability in all kind of technologies.
- Details depending on the selected protocol:
 - MQTT topic where data will be published (publish/subscribe pattern). RabbitMQ provides seamless interaction between MQTT and AMQP clients, by mapping the concepts of MQTT topics to AMQP Routing Keys
 - AMQP queue where the module listens for requests (RPC pattern). If this is the case, further properties may be declared, such as "reply_to" (sender specifies the name of the queue where the response is expected) and "correlation_id" (sender specifies a unique id to the message that must be also included in the response)
 - Message payload: message payload supports any kind of content. Most modules will actually use JSON messages.

3.2 REAL-TIME MONITORING

3.2.1 Unbundled Smart Meters Wrapper (SMX Wrapper)

The WiseGRID Cockpit application will interact with the required field data sources using the WiseGRID IOP. One of the main data sources that will be used are the Unbundled Smart Meters (USMs). These smart meters are equipped with WiFi, 3G and/or Ethernet, and are capable of communicating data to the application with low-period frequency (e.g. from 1 to 10 seconds) using regular internet connections. These devices will directly communicate with the WG IOP, thus making these measurements available to any application of the project (including WG Cockpit) requiring it and having the right to access it. This information will be published using the CIM format (IEC61968), encapsulated in MQTT payloads.

The details of the communication of those devices with the ecosystem of applications are detailed in D4.2 [4]. This section describes the relevant facts for the design of the WiseGRID Cockpit.

3.2.1.1 Unbundled Smart Meters concept

Unbundled Smart Meters concept has been previously developed in the NobelGrid project [5]. The Unbundled Smart Meter is a systemization where smart meter functionalities are grouped in two separate components – Smart Metrology Meter (SMM) for metrological and hard real-time functions, and Smart Meter extension (SMX) to provide flexibility by allowing the deployment of further functionalities and incorporating new communication channels. SMXs are able to communicate with smart meters of different vendors and provide energy readings and real-time measurements in a unified format.

3.2.1.2 Wrapper features

The wrapper module is based on the software SMXCore [6], which runs on the SMX device and provides the interface to the WG IOP.

The wrapper interacts with the WG IOP using MQTT. SMX regularly publishes DLMS/COSEM-based energy and real-time measurements readings and subscribes to topics where configuration updates shall be received.

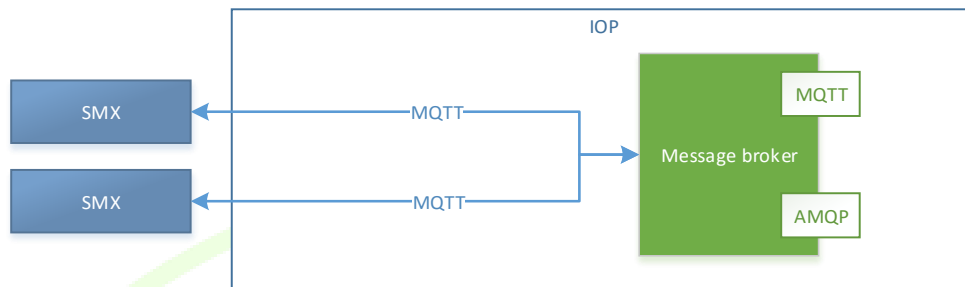


Figure 8 – Publication of data from USM into the WiseGRID IOP

In order to comply with the CIM common data model adopted by the project, the WiseGRID IOP incorporates a module in charge of translating those measurements on the fly into CIM MeterReading messages.

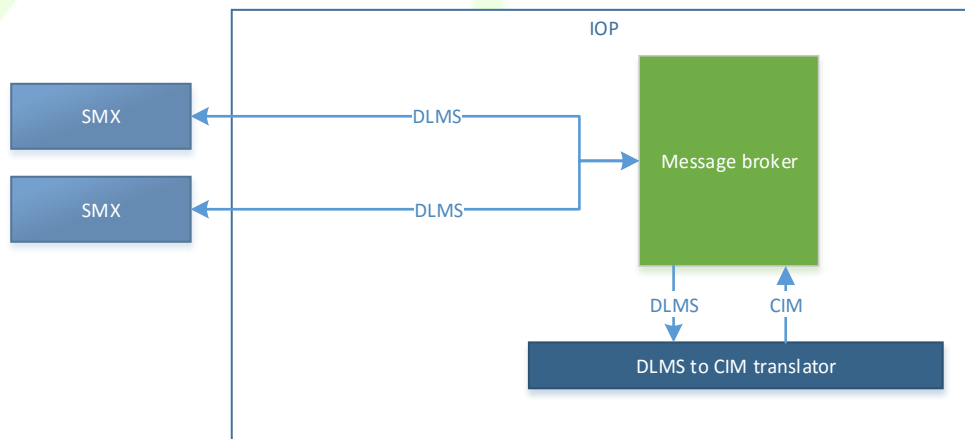


Figure 9 – Translation to CIM Common Data Model

Once the data flow is published in the WiseGRID IOP, the Real-time monitor of the WiseGRID Cockpit (Section 3.2) will subscribe to it and will feed the corresponding databases, keeping track of all data provided by those devices.

3.2.2 AMI Wrapper

The WiseGRID Cockpit application will interact with the required field data sources using the WiseGRID IOP. The second main data source that will be used are the corresponding Advanced Metering Infrastructure systems of the pilot sites. These systems are in charge of regularly collecting data of the already deployed smart meters, usually communicating with those using PLC (Power Line Communication) channels. A specific wrapper will be implemented for each one of the pilot sites that already has an AMI system for the deployed smart meters. The wrapper will be in charge of retrieving measurements collected by the AMI system and publish those to the WG IOP, thus making them available to the different applications of the WiseGRID project. This information will be published using the CIM format (IEC61968), encapsulated in MQTT payloads.

The details of the communication of those devices with the ecosystem of applications are detailed in D4.2 [4]. This section describes the relevant facts for the design of the WG Cockpit.

3.2.2.1 Wrapper features

AMI systems typically collect data from the deployed smart meters on a regular basis (e.g. 24 hour demand curves retrieved once per day). In order to integrate this data with the ecosystem of WiseGRID applications,

it will be published following the exact same mechanisms (MQTT publications) and data model (CIM MeterReadings) as defined for the Unbundled Smart Meters, thus making the actual source of information transparent to the application.

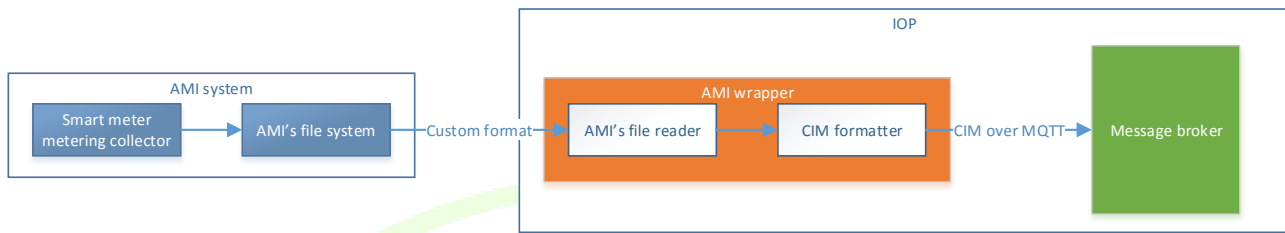


Figure 10 – Approach for integration of AMI data

Since source data will be provided in a custom format (depending on the vendor of the deployed AMI system), different AMI wrappers will need to be developed for each one of the pilot sites. Further details can be found in D4.2.

Once the data flow is published in the WG IOP, the Real-time monitor of the WiseGRID Cockpit (section 3.2) will subscribe to it and will feed the corresponding databases, keeping track of all data provided by those devices.

3.2.3 SCADA Wrapper

In the context of the WiseGRID project, SCADA systems monitor the status of the different elements in the distribution grid under control of the DSO. The information that is particularly relevant to the WiseGRID Cockpit comprises:

- Electric measurements from substations: electric parameters of the buses of the grid are typically measured by equipment located in the substations. These measurements will be published to the WiseGRID IOP using the already described CIM MeterReading messages. This way, measurements will be transparently integrated in the WG Cockpit, exactly as they were provided by USM devices or extracted from the AMI systems.
- Status of safety elements (switches): algorithms to be executed within the WiseGRID Cockpit need to be aware of the current status and changes in those elements. The status will be read from the corresponding SCADA and published to the WG IOP using CIM Switch messages.

```

{
  mRID: "SW00001"
  normalOpen: true
}

```

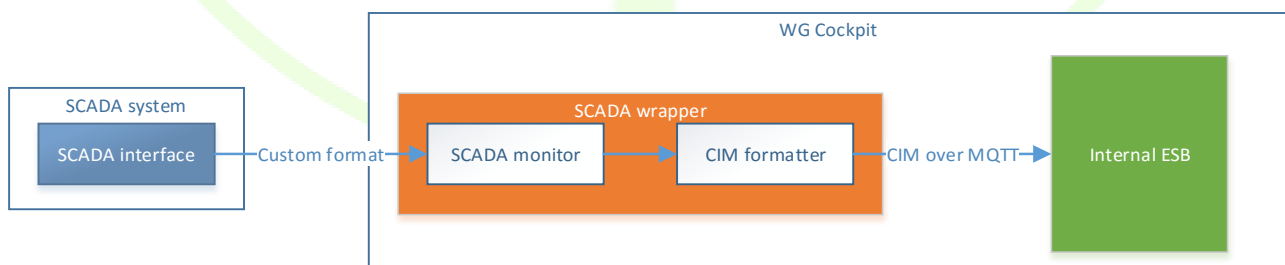


Figure 11 – Overview of generic approach for SCADA integration

Different SCADA wrappers will need to be developed for each one of the pilot sites.

3.2.3.1 Required SCADA Wrappers per pilot site

3.2.3.1.1 Crevillent

ENERCOOP currently works with a SCADA system provided by SITEL [7]. At the moment of writing, this SCADA is being upgraded to its latest version, which will include a set of web-services allowing to extract data from the SCADA. The approach that will be taken by this wrapper will be, therefore:

1. Execution of the SCADA Wrapper will be programmed with a period as short as possible (e.g. 10 seconds)
2. Execution will consist of a set of calls to the SCADA Web-Service interface in order to get data from all relevant items. For each execution:
 - a. A request will be triggered to the SCADA
 - b. Response will be received and processed to the corresponding CIM message
 - c. CIM message will be published to the internal ESB using MQTT protocol

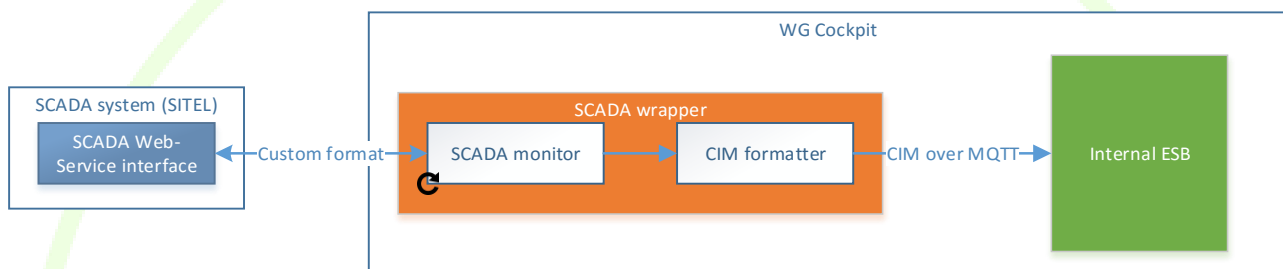


Figure 12 – Overview of integration with ENERCOOP’s SCADA

3.2.3.1.2 Terni

There will be no integration of Terni’s SCADA during the deployment phase. The data will be provided by SMX devices.

3.2.3.1.3 Mesogia

For the Mesogia pilot site, the data from the test SCADA system of the area (that will be an exact replica of the actual systems) will be available and collected offline. The procedure to be followed for the collection of the data described below.

- Data from the Mesogia SCADA system are exported manually every week in the form of CSV files.
- CSV file are then compressed and uploaded to an FTP server in a specific directory with a pre-defined formatting of the CSV file.
- A micro-service is triggered automatically upon the upload of the CSV files.
- CSV files are decompressed and data checking and cleansing, as well as missing values analysis and correction is performed.
- Data from each diesel engine are then used to construct the respective JSON file to be send automatically to the WG IOP

To provide the WiseGRID IOP with simulated “real-time” data of the SCADA operational database, a sliding window approach on historical data will be used, where current operational data will be simulated using the historical data of the previous week on the same date and time as with the current moment.

3.2.3.1.4 Kythnos

For the Kythnos pilot site, the data from the production station of the island will be available and collected offline from the local SCADA system in the generation power plant in Kythnos, mainly related to the production profiles of the diesel generators in operation in the island. The procedure to be followed for the collection of the data is similar than the one for Mesogia.

- Data from the Kythnos power plant station operational SCADA system are exported manually every week in the form of CSV files.
- CSV file are then compressed and uploaded to an FTP server in a specific directory with a pre-defined formatting of the CSV file.
- A micro-service is triggered automatically upon the upload of the CSV files.
- CSV files are decompressed and data checking and cleansing, as well as missing values analysis and correction is performed.
- Data from each diesel engine are then used to construct the respective JSON file to be send automatically to the WG IOP

3.2.4 Real-time monitor

The *Real Time monitor* is a horizontal module that will handle the data ingestion for most of the applications of the project. It has been design in order to fulfill the requirements for data ingestion accordingly to the requirements and the architecture of communications proposed for the applications. Its main features are:

Support for ESB communications

The module supports the main communication protocols proposed for the applications of the project: MQTT and AMQP. Use of these communication protocols show several advantages, as described in detail in D4.2 [8]. The RT monitor will therefore support the communication mechanisms deemed appropriate to fulfill the requirements of the applications of the project.

Configurable subscription to data flows

The common approach for the data ingestion in the project, is that field devices regularly publish their status to the WG IOP by using the proper Common Data Model. This task is accomplished by the different Wrappers developed within the project in order to integrate a variety of data sources. Publications to the WG IOP are either performed using MQTT or AMQP protocols.

The RT monitor takes advantage of this design to allow the application to subscribe to the proper data flows of the W IOP Message Broker, thus being able to retrieve data as this gets published by the field devices.

Maintenance of databases

The module automatically maintains the necessary data within the two different MongoDB databases proposed per application: the operational database and the long-term (big data) database. Data is stored in different collections accordingly to its source (smart meters, batteries, sensors...)

- Operational DB keeps one document per device, stating last known status. Its main purpose is to hold a picture of the current status of the devices controlled by the application.
- Long-term DB keeps a history of changes per device, each change hold by a document. Its main purpose is serve as data source for KPI calculation and analysis.

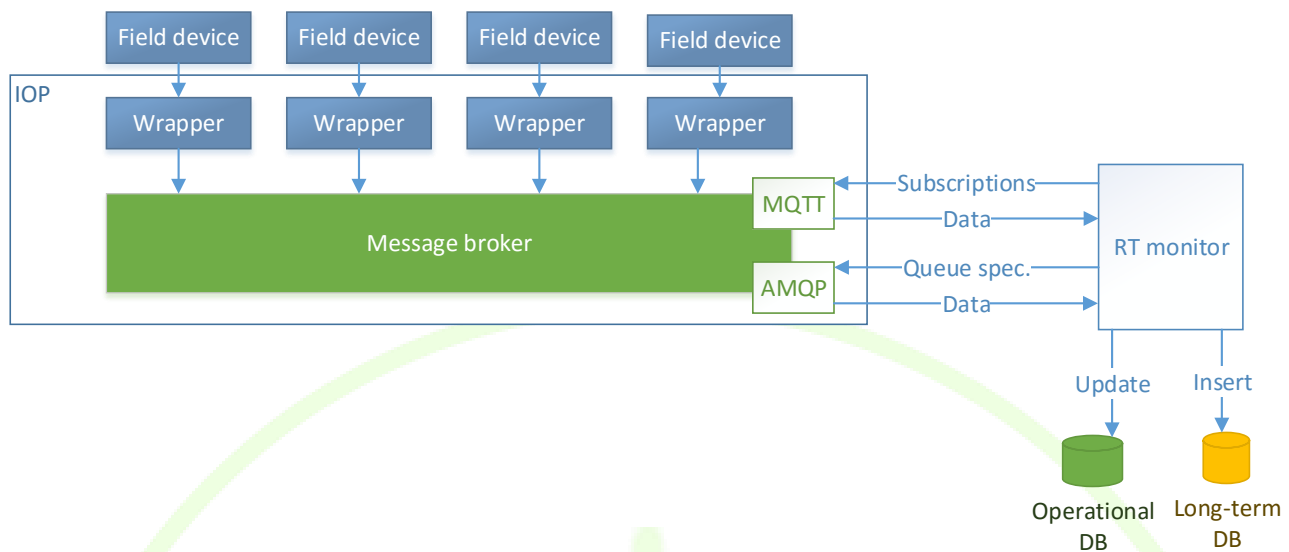


Figure 13 – Overview of the structure of the RT monitor

3.2.5 KPI engine

The KPI engine is a module that takes advantage of the big data platform features to calculate relevant indicators for the application. In the context of WiseGRID Cockpit, it will be in charge on the regular calculation of the following KPIs relevant to DSO, as defined in D2.1 [9]:

- Temporal aggregation (daily, monthly) of demand and production readings in the grid.
- Equivalent CO2 emissions of energy supplied within the distribution grid.
- Energy mix for the distribution grid (taking into account monitored production).
- System Average Interruption Duration Index.
- System Average Interruption Frequency Index.
- Customer Average Interruption Duration Index.
- Customer Average Interruption Frequency Index.
- LV faults clearance index.
- Voltage variation.
- Frequency deviation.

All these KPIs will be based on the following inputs, which will be made available by modules of the WiseGRID IOP or the WiseCORP internal modules:

- Demand and production data of the customers supplied by the DSO, as provided by the corresponding USM and AMI Wrappers, or SCADA Wrapper.
- Composition of the energy mix and equivalent CO2 emissions of supplied energy, as provided by the energy mix provider.
- Electric measurements of the grid, provided by USM or SCADA Wrappers.
- Detail of incidences managed by the Maintenance Module.

The calculation of each one of those indicators will be implemented as an independent Spark processing module that will take the data available in the long-term database as input. Those modules will be regularly executed to calculate the required indicators accordingly to the new data available on database, and will deliver results to three possible sinks, depending on the KPI being calculated: the operational database of the application (for KPIs representing current status), long-term database (for KPIs indicating aggregations or trends) and the internal ESB of the application (for KPIs triggering further actions on other modules).

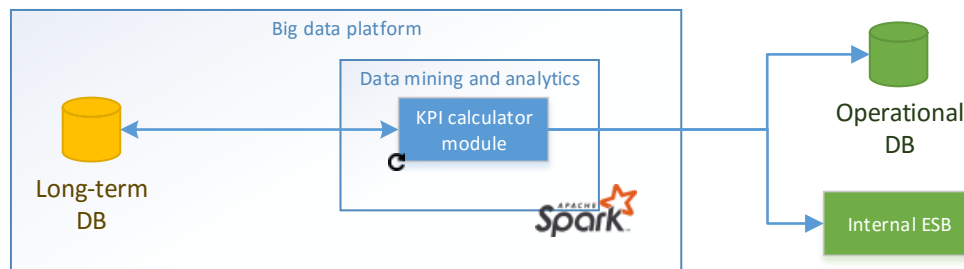


Figure 14 – Structure of the KPI calculator modules

The foreseen technique to be used for the calculation of the KPIs is **Map/Reduce**. This is a common technique when performing aggregations of big data sets allowing parallel distributed computation, thus being able to take advantage of the availability of data mining infrastructure in a cluster, configuration allowed by Spark. Processing is based in combinations of the two operations: *Map* and *Reduce*.

Map operation is used to transform the original (raw) data into a different format, more delimited to the calculation that needs to be performed. With this operation, irrelevant data for the aggregation can be removed from the original document, releasing a new version of the document that just contains the necessary data for the aggregation (e.g. id of the data source, timestamp and value for energy consumption).

Reduce operation is used to aggregate the information contained within a set of documents into a single one. This operation will be used to performed several aggregations, which may be used simultaneously:

- Temporal aggregation: data produced by a single source (e.g. a single smart meter) is aggregated using its timestamp value (e.g. all values falling in the same day or month), to produce a timely aggregated value.
- Source aggregation: data produced by several different sources (e.g. all smart meters whose owners have a common contract type) are aggregated to produce a single value per timestamp (e.g. to get the aggregated quarterly demand of the whole portfolio of prosumers).

The objective of the KPI calculation modules using this technique is to preprocess information that will be further on displayed by the WiseGRID Cockpit application and used as part of the evaluation of the tool.

3.3 OFF-LINE PROCESSES

3.3.1 Topology processor

3.3.1.1 Overview

The Topology processor is a set of tools that allows the user to easily design the electrical grid using predefined electrical components, provide information for each individual object in the grid and finally transform and forward the resulting grid in a data format that WiseGRID services can use and exploit.

The Topology processor relies on three different components. These are:

- Microsoft Visio as a diagram and visual editor.
- A stencil that provides the user with a predefined set of electrical components.
- A transformation and data forwarding application (Topology Tool).

The Topology processor uses the popular diagram software Microsoft Visio© to provide users with a familiar and intuitive design environment that allows them to create and modify the electrical grid topology and component properties in visual manner.

3.3.1.2 Detailed Description

The topology is created by the user by adding and connecting the different components that are provided by

a stencil that is shipped with Topology processor. The Stencil includes a set of predefined electrical grid components that are currently supported. Additional components can be added easily in case any such need arises.

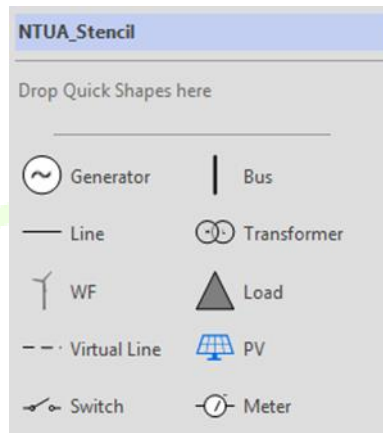


Figure 15 – Stencil of the Topology processor module.

The currently supported electrical grid components include:

- Thermal Generators (Generator): conventional generators using fossil fuels to produce energy
- Wind Generators or Farms (WF)
- Photovoltaic Generators (PV)
- Transformers (Transformer)
- Loads or group of Loads (Load)
- Switches (Switch)
- Lines (Line)
- Meters (Meter) to denote a metering point on a component
- Virtual Lines (Virtual Line) which is a zero resistance/reactance connector between electrical grid components (used primarily in substations)

Each electrical grid component has a predefined set of properties that the user fills in in order to provide more information about the component itself and its position in the grid.

The figures below show sample grids and some of the component details.

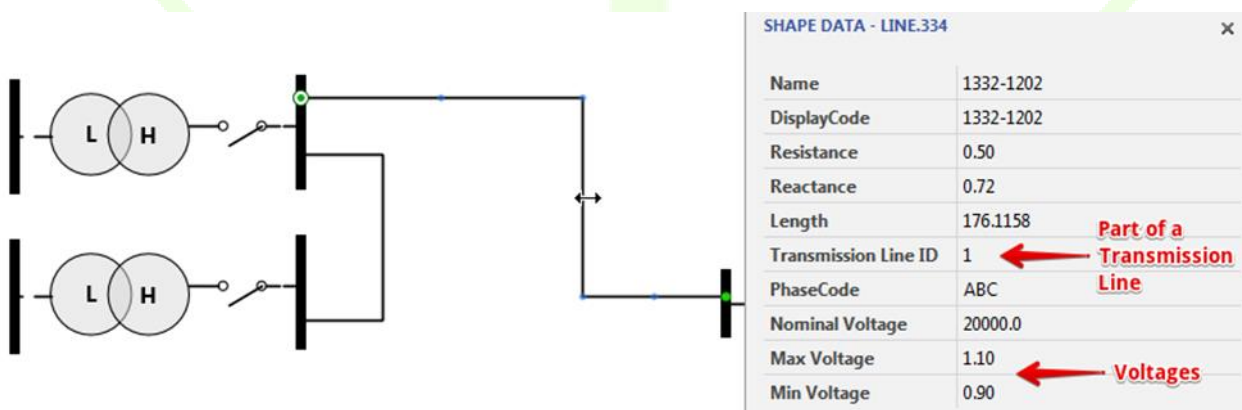


Figure 16 – Sample grid with component details.

Once the grid is finalized, the file is saved (vsdx format) in the local system. This file will be used by the Topology Tool to extract the topology of the grid, component properties, transform the data and finally forward them to the Internal Service Bus.

The Topology Tool is acting as an intermediate layer between the Visio diagrams (OpenXML format) used to map the available Visio grid components to intermediate format (IMF) objects based on a predefined set of rules.

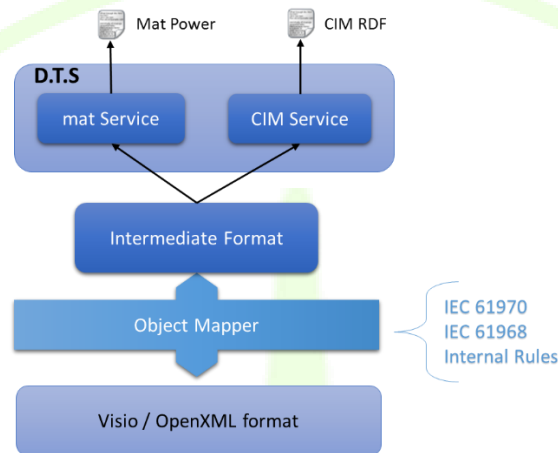


Figure 17 – Topology tool structure

The intermediate format (IMF) is only used in the context of the ISB and carries information regarding the connectivity, diagram information, GPS data and extended properties (properties not available in the IEC 61970 but useful in other cases). The IMF is then used by Data Transformation Services (DTS) to change the topology information into other formats (e.g. mat Power, RDF).

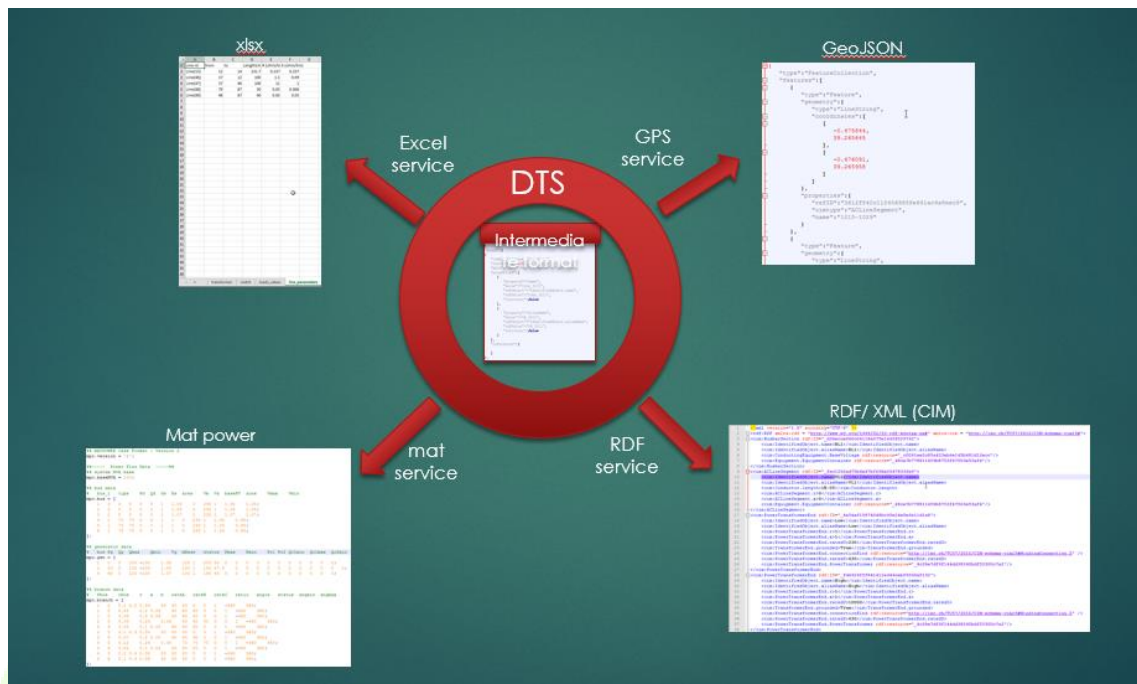


Figure 18 – Formats provided by Data Transformation Service (DTS)

A small sample of the intermediate format can be seen in following figure.

```
{
  "graph": [
    {
      "cimType": "Terminal",
      "resourceID": 3210,
      "mRID": "4c80851c258246b488160a6c7afe711f",
      "properties": [
        {
          "property": "name",
          "value": null,
          "rdfObject": "IdentifiedObject.name",
          "rdfValue": null,
          "isLiteral": false
        },
        {
          "property": "aliasName",
          "value": "CN_3214",
          "rdfObject": "IdentifiedObject.aliasName",
          "rdfValue": "CN_3214",
          "isLiteral": false
        }
      ],
      "references": [
        {
          "cimObject": "ConductingEquipment",
          "rdfObject": "Terminal.ConductingEquipment",
          "resourceID": 2017
        },
        {
          "cimObject": "ConnectivityNode",
          "rdfObject": "Terminal.ConnectivityNode",
          "resourceID": 3211
        }
      ]
    },
    {
      "cimType": "ConnectivityNode",
      "resourceID": 3214,
      "mRID": "4c31629d5c094171b733a15b3c774088",
      "properties": [
        {
          "property": "name",
          "value": "Conn_3214",
          "rdfObject": "IdentifiedObject.name",
          "rdfValue": "Conn_3214",
          "isLiteral": false
        },
        {
          "property": "aliasName",
          "value": "CN_3214",
          "rdfObject": "IdentifiedObject.aliasName",
          "rdfValue": "CN_3214",
          "isLiteral": false
        }
      ],
      "references": [
        {
          "cimObject": "Terminal",
          "rdfObject": "ConnectivityNode.Terminal",
          "resourceID": 3210
        }
      ]
    }
  ],
  "diagram": [
    {
      "Height": -13.246000000000095,
      "Width": 5.0,
      "Angle": 0.0,
      "Top": 269.553000000000005,
      "Left": 110.49999999999999,
      "MasterShape": "Line",
      "ShapeCode": "Line",
      "BeginX": 109.49999999999999,
      "BeginY": 276.176000000000022,
      "EndX": 111.5,
      "EndY": 262.930000000000012,
      "internalID": 48
    }
  ],
  "ext": [
    {
      "internalID": 48,
      "property": "SLat",
      "value": 39.249405
    }
  ]
}
```

Figure 19 – Sample of the Intermediate Format (IMF).

The following table summarizes some key aspects of the module.

Table 3 – Key aspects topology

Key aspect	Format
Input User Interface	Microsoft Visio©
Input Format	Vsdx format
Output Format	.xlsx, .mat, GEOJson, RDF/XML

3.3.2 GIS provider

The GIS provider module comprises all necessary software needed to retrieve GIS data from the databases available on the different pilot sites in order to convert those to the GIS Common Data Model proposed for the WG Cockpit tool, thus making it possible to geographically represent all elements in the User Interface and establish relationships among them that facilitate the usage to the DSO operator. Those representations are based on CIM Common Data Model and GeoJSON, extended with further information useful for the implementation of the WG Cockpit.

The following sections describe in detail this Common Data Model, serving as a reference to for the transformation work to be done for each one of the pilot sites demonstrating the WG Cockpit. The minimum amount of information required by the application is shown. By using JSON structures, extra CIM-based fields with additional information deemed necessary can be added to the model.

3.3.2.1 Data model

The grid elements required to be modelled with the GIS format include lines (medium and low voltage), substations, usage point location, usage point and RES.

3.3.2.2 Lines

3.3.2.2.1 Medium voltage lines

The information deemed necessary to model the medium voltage lines include its ID, its geometry and the list of substations that are connected to the line.

Table 4 – Medium voltage lines GIS model

Item	Description
mRID	Id of the line (free text)
Geometry	GeoJSON representation of the line.
Relations / substations	Id of the substation(s) and RES the line is connected to
type	MV (medium voltage)
name	Name of the line
site	Pilot site identifier

Exemplary document:

```
{
  "type" : "MV",
  "site" : "Crevillent",
  "mRID" : 4,
```

```

"name" : "LB",
"geometry" : {
  "type" : "MultiLineString",
  "coordinates" : [
    [
      [
        -0.80259189413784,
        38.2468038749651
      ],
      [
        -0.802568909933584,
        38.2467950725124
      ],
      ...
    ]
  ]
}

```

3.3.2.2.2 Low voltage lines

The information deemed necessary to model the medium voltage lines include its ID, its geometry and the list of substations that are connected to the line (usually just one).

Table 5 – Low voltage lines GIS model

Item	Description
mRID	Id of the line (free text)
Geometry	GeoJSON representation of the line.
Relations / substations	Id of the substation(s), usage point location and RES the line is connected to
type	LV (low voltage)
site	Pilot site id

Exemplary document:

```

{
  "site" : "Crevillent",
  "mRID" : "101",
  "type" : "LV",
  "geometry" : {
    "type" : "MultiLineString",
    "coordinates" : [

```

```
[
  [
    -0.81256855532408,
    38.2458319451761
  ],
  [
    -0.812566077538753,
    38.2458357515238
  ],
  ...
]
},
"relations" : {
  "substations" : [
    "1"
  ]
},
}
```

3.3.2.3 Substations

The information deemed necessary to model the substations include its ID and its geometry. A generic *Properties* placeholder has been introduced to store all other information available at source GIS databases that may be considered relevant for future use. The relationships to the buses of the substations need to be introduced as well, in the form of a link to the corresponding element in the topology model.

Table 6 – Substations GIS model

Item	Description
mRID	Id of the substation (free text)
Geometry	GeoJSON representation of the line.
Properties	Placeholder for further properties of interest
type	Substation
site	Pilot site id
cim	Link to the substation element in the topology model, as provided by the Topology Processor Tool (see section 3.3.1)
busesHigh	Link to the related high voltage buses in the topology model, as provided by the Topology Processor Tool (see section 3.3.1)
busesLow	Link to the related low voltage buses in the topology model, as provided by the Topology Processor Tool (see section 3.3.1)

Exemplary document:

```
{
  "_id" : "50",
  "type" : "Substation",
  "properties" : {
    "id" : 50,
    "id_subtipus" : 1,
    "name" : "091",
    "rotation" : -38.109305622704,
    "texte" : "091\nLA VENTA"
  },
  "geometry" : {
    "type" : "Point",
    "coordinates" : [
      -0.779318651317007,
      38.2758616316822
    ]
  },
  "site" : "Crevillent",
  "mRID" : "50",
  "busesLow" : [
    {
      "id" : "http://crevillent.wise-
grid.eu/#_6ad79c2800234691b28f7eaf35891890"
    }
  ],
  "busesHigh" : [
    {
      "id" : "http://crevillent.wise-
grid.eu/#_eaa9e983b7e446f3851763a264820664"
    },
    {
      "id" : "http://crevillent.wise-
grid.eu/#_a5d1b1aea9744c8396c68734c1929cf3"
    }
  ]
}
```

3.3.2.4 Usage point locations

The data model for the definition of the usage point locations includes all elements of its CIM definition. The following table defines the minimum necessary information to manage those correctly within the WG Cockpit.

Table 7 – Usage point location GIS model

Item	Description
mRID	Id of the usage point location (free text)
direction	Address of the usage point location
type	Type of usage point location (free text)
site	Pilot site id
Name	Name of the usage point location
geometry	GeoJSON representation of the usage point location
Relations / LV	Id of the low voltage line connected to this usage point location
Relations / MV	Id of the medium voltage line connected to this usage point location
Relations / Substation	Id of the substation supplying this usage point location

Exemplary document:

```
{
  "_id" : "6",
  "mRID" : "6",
  "accessMethod" : "",
  "direction" : "PTDA.SAN ANT.FLORIDA 0 CREVILLEN",
  "geoInfoReference" : null,
  "remark" : "",
  "siteAccessProblem" : null,
  "type" : "CONTA-EP",
  "ConfigurationEvents" : null,
  "CoordinateSystem" : null,
  "electronicAddress" : null,
  "mainAddress" : null,
  "Names" : {
    "name" : "PTDA.SAN ANT.FLORIDA 0 CREVILLEN",
    "NameType" : null
  },
  "phone1" : null,
  "phone2" : null,
  "PositionPoints" : null,
  "secondaryAddress" : null,
}
```

```

"status" : null,
"shape" : "pod",
"geometry" : {
  "type" : "Point",
  "coordinates" : [
    -0.815703126295567,
    38.2456671918958
  ]
},
"name" : "PTDA.SAN ANT.FLORIDA 0 CREVILLEN",
"site" : "Crevillent",
"relations" : {
  "LV" : null,
  "substation" : null
},
}

```

3.3.2.5 Usage point

The data model for the definition of the usage points includes all elements of its CIM definition. The following table defines the minimum necessary information to manage those correctly within the WG Cockpit.

Table 8 – Usage point GIS model

Item	Description
mRID	Id of the usage point (free text)
Name	Name of the usage point
geometry	GeoJSON representation of the usage point
Relations / Usage point location	Id of the usage point location connected to this usage point
Relations / LV	Id of the low voltage line connected to this usage point
Relations / Substation	Id of the substation supplying this usage point

Exemplary document:

```

{
  "_id" : "1",
  "relations" : {
    "usagePointLocation" : "5557",
    "substation" : null,
    "LV" : null
  },
  "mRID" : "ES007300000221XXXXXX",
  "amiBillingReady" : 4,
}

```



```

"amiBillingReadySpecified" : true,
"checkBilling" : false,
"checkBillingSpecified" : false,
"connectionState" : 1,
"connectionStateSpecified" : true,
"estimatedLoad" : 0.0,
"estimatedLoadSpecified" : false,
"grounded" : false,
"groundedSpecified" : false,
"isSdp" : true,
"isSdpSpecified" : true,
"isVirtual" : false,
"isVirtualSpecified" : true,
"minimalUsageExpected" : true,
"minimalUsageExpectedSpecified" : true,
"nominalServiceVoltage" : 220.0,
"nominalServiceVoltageSpecified" : true,
"outageRegion" : "33,3",
"phaseCode" : 0,
"phaseCodeSpecified" : false,
"ratedCurrent" : 0.0,
"ratedCurrentSpecified" : false,
"ratedPower" : 6928.0,
"ratedPowerSpecified" : true,
"readCycle" : null,
"readRoute" : null,
"serviceDeliveryRemark" : null,
"servicePriority" : null,
"ConfigurationEvents" : null,
"Equipments" : {
  "mRID" : null,
  "Names" : null
},
"MetrologyRequirements" : null,
"Names" : {
  "name" : "PTDA. CACHAPETS 3 (CREVILLEN)",
  "NameType" : null
},
"PricingStructures" : null,

```

```

    "ServiceCategory" : null,
    "ServiceLocation" : null,
    "ServiceMultipliers" : null,
    "ServiceSupplier" : null,
    "UsagePointLocation" : null,
    "name" : "PTDA. CACHAPETS 3 (CREVILLENT)",
    "location" : "5557",
    "site" : "Crevillent"
  }

```

3.3.2.6 Renewable Energy Sources

The GIS common data model for RES includes information about its id, location, type (PV, Wind) and capacity.

Table 9 – RES GIS model

Item	Description
mRID	Id of the RES (free text)
geometry	GeoJSON representation of the RES
subtype	Type of RES (pv panel, wind mill)
capacity	Production capacity of the RES

Exemplary document:

```

{
  "_id" : "renew2",
  "mRID" : "renew2",
  "geometry" : {
    "type" : "Point",
    "coordinates" : [
      -0.472813040013983,
      39.2615159843937
    ]
  },
  "shape" : "pvpanel",
  "site" : "Crevillent",
  "state" : {},
  "relations" : {},
  "type" : "Renewable",
  "subtype" : "PV Panel",
  "capacity" : 50.0
}

```

3.3.2.7 Required GIS data sources adapter per pilot site

3.3.2.7.1 Crevillent

ENERCOOP currently manages a PostGIS database [10], an extension of PostgreSQL [11] for managing spatial elements. All elements under responsibility of the DSO are modelled in this database, including not only geospatial data, but also all details relevant to each of the elements (e.g. relationships between Usage Points and Usage Point Locations, or between lines and substations).

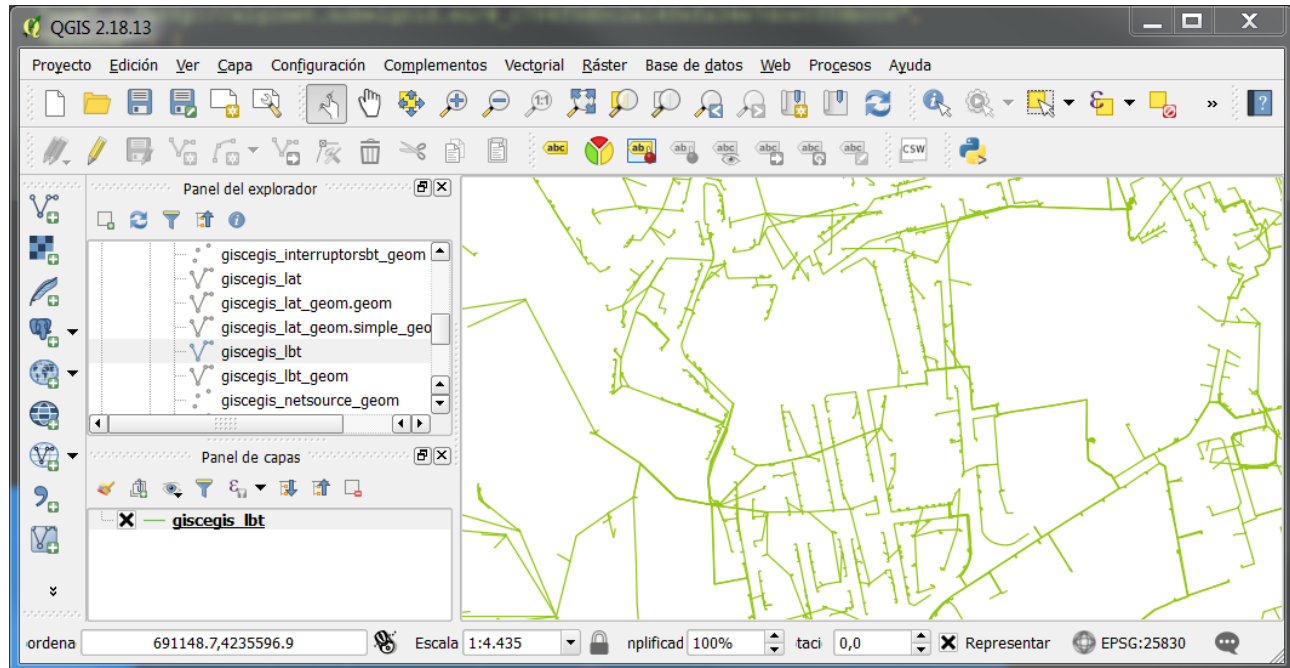


Figure 20 – Screenshot of ENERCOOP's GIS database

The process to incorporate this information into the WiseGRID Cockpit is performed offline and comprises the following tasks:

1. Connect to ENERCOOP's GIS database. Only read-access permission is required.
2. Extract the information from the relevant tables for all elements considered in the GIS Common Data Model of WiseGRID Cockpit
3. Transform available data into the Common Data Model
4. Write transformed information into the Operational Database of WiseGRID Cockpit

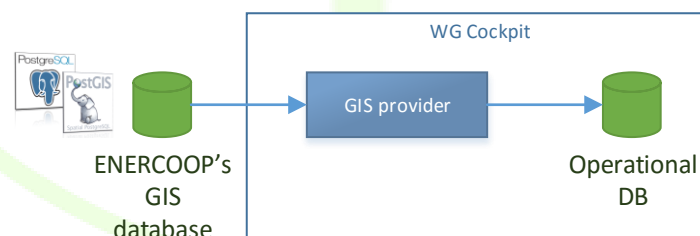


Figure 21 – ENERCOOP's GIS provider structure

3.3.2.7.2 Terni, Mesogia and Kythnos

For these pilot sites, the Topology Processor tool (see section 3.3.1) used to model the electric details of the grid is also used to include extended information, including the relevant GIS data (GeoJSON representation of the elements of the topology that can be modelled with the tool). The process to incorporate this

information into the WG Cockpit is performed offline and comprises the following tasks:

1. Read the CIM topology model of the pilot site, exported from the Topology Processor tool.
2. Extract the relevant GIS data (GeoJSON representation of the relevant elements – lines, substations, production units...)
3. Transform this data into the Common Data Model for the GIS data managed by WG Cockpit
4. Write transformed information into the Operational Database of WG Cockpit.

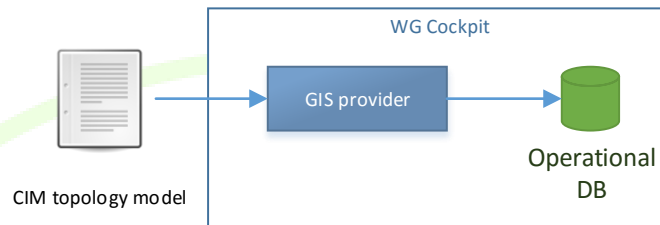


Figure 22 – ASM's GIS provider structure

3.3.3 Power Quality

3.3.3.1 Overview

The Power Quality module is an offline module, the scope of which is to process and evaluate the power quality for a set of measurements for a specific time interval. The outcome of this process is a report which assesses an exact or low error practical compliance of the measurements to the standard EN 50160, by using specialized equipment or cost effective smart meter acquisition and algorithms based on statistical approach to reach an appropriate view of the power quality level.

The theoretical background and the algorithmic framework of the module have been described at the deliverable D12.2. In brief, the module can analyse two different kind of data, depending on what is available: either the voltage waveforms directly, if the module is used at a lower level architecture, or the smart meter collected data, which include magnitude and angle of every harmonic, for cases where the module is used at higher level architectures like in WG Cockpit.

3.3.3.2 Detailed Description

The Power Quality module, implemented in the WG Cockpit architecture, retrieves data which correspond to a specific time interval from the long-term database.

The data fill the appropriate sheet of the input file, which is a .xlsx document (or other easy processable data format such as csv or JSON), depending on the type of the data (RMS values or voltage time series).

	A	B	C	D	E	F	G	H
1	t (ms)	Voltage			Current			
2		Va (V)	Vb (V)	Vc (V)	Ia (A)	Ib (A)	Ic (A)	
3	0	0	-95,23	95,23	0	-3,01	4,01	
4	0,02	0,01	-95,26	95,25	0,01	-3,02	4,02	
5	0,04	0,05	-95,27	95,28	0,03	-3,05	4,05	
6	0,06	0,12	-95,29	95,29	0,08	-3,09	4,09	
7	0,08	0,18	-95,31	95,32	0,11	-3,11	4,11	
8	0,1	0,25	-95,33	95,33	0,13	-3,14	4,14	
9	0,12	0,29	-95,35	95,35	0,14	-3,15	4,15	
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								

Figure 23 – Input data (RMS values) from database.

At the same document, the necessary parameters need to be configured by the user through an appropriate user interface (WG Cockpit UI) or directly on the file.

A	B	C	D	E	F	G	H	I
f (Hz)	Number of Harmonics	First option: RMS values given		Second Option: Waveform construction		Voltage dips classification		
		Total examination time (h)	Sample rate (ms)	Total examination time (h)	Sample rate (ms)	Number of voltage dips	Voltage limits (%)	Time limits (ms)
50	25	X	Y	X	Y	N	90	10
							80	200
							70	500
							40	1000
							5	5000
							0	60000

Figure 24 - Input data from user.

It should be noted that the executable file is the same, regardless of the type of the input data.

The outcome of the data process is a report which contains:

- a datasheet, including the harmonic and unbalance analysis of the distribution network, as well as the classification of voltage dips,

Table 10 – Datasheet for power quality analysis.

Harmonics			
Power Quality Indices	Value (95% of 10 min mean r.m.s.)	Limits according to EN 50160	Error
Relative Harmonic Voltage			
U ₂	1,7 %	2,0 %	NO
U ₃	5,4 %	5,0 %	YES
U ₄	0,8 %	1,0 %	NO

U ₅	7,1 %	6,0 %	YES		
...		
U _h		
THD	6,4 %	8 %	NO		
Unbalance					
Power Quality Indices	Value (95% of 10 min mean r.m.s.)	Limits according to EN 50160	Error		
VUF	1,2 %	2 – 3 %	NO		
Voltage dips					
Voltage dips magnitude u (%)	Duration t (ms)				
	10≤t≤200	200≤t≤500	500≤t≤1000	1000≤t≤5000	5000≤t≤60000
90>u≥80	8	7	7	5	6
80>u≥70	9	9	2	2	3
70>u≥40	6	5	5	3	1
40>u≥5	3	3	3	1	1
5>u	5	1	1	0	0

- A graphical representation of voltage dips classification,

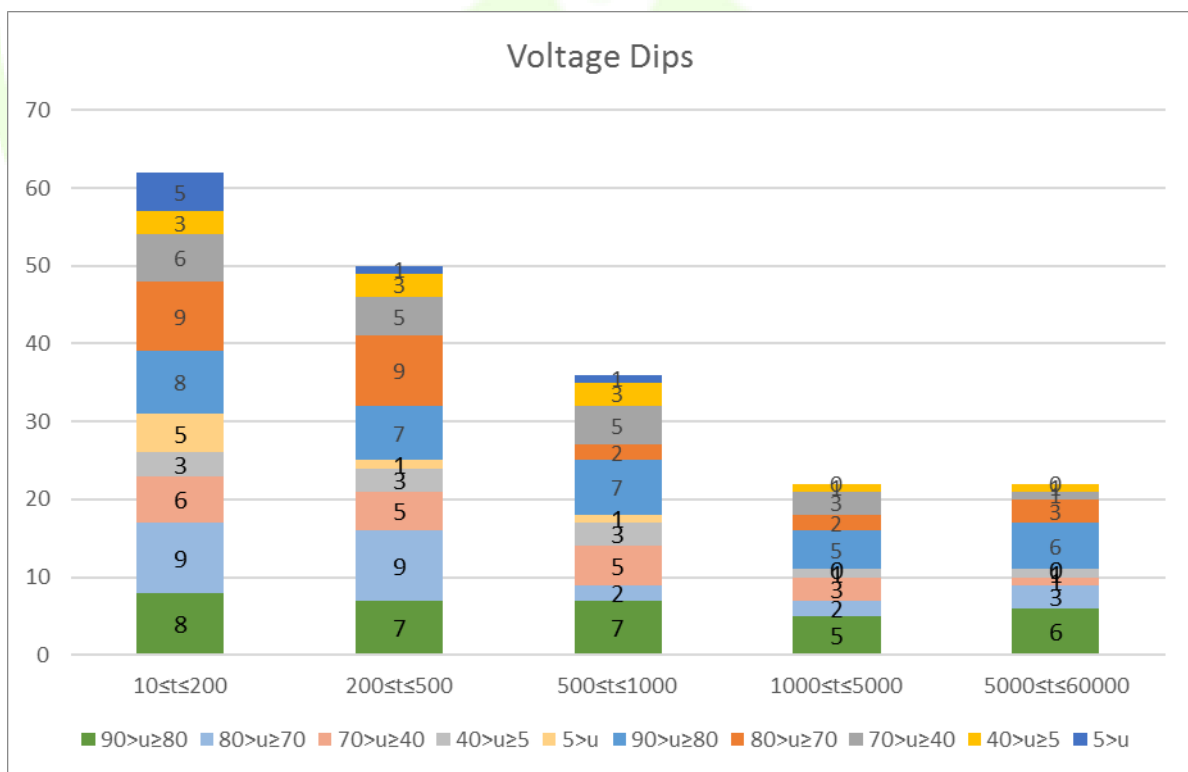


Figure 25 – Voltage dips classification.

Both of the previous graphical representations are stored in an .xlsx document or in files based on another easy processable format (e.g. csv files or structured JSON, to be further defined), which is the output file.

The following table summarizes some key aspects of the module.

Table 11 – Key aspects power quality module

Key aspect	Format
Input User Interface	WG Cockpit UI or Microsoft Excel
Input Format	.xlsx
Output Format	.xlsx

3.3.4 Grid planning assistant

The main purpose of the Grid Planning Assistant is allowing the distribution operator to simulate *what if* scenarios, in order to assess the effect of different load, generation and flexibility patterns to the grid. This module integrates the results of the tasks T4.1 and T4.2 within the WiseGRID Cockpit application.

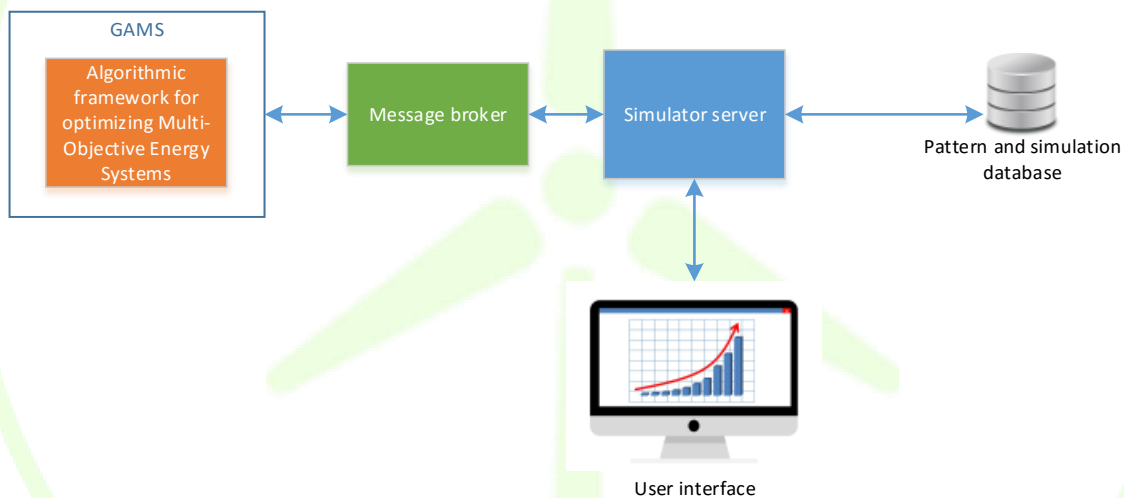


Figure 26 – Components of the Grid Planning Assistant

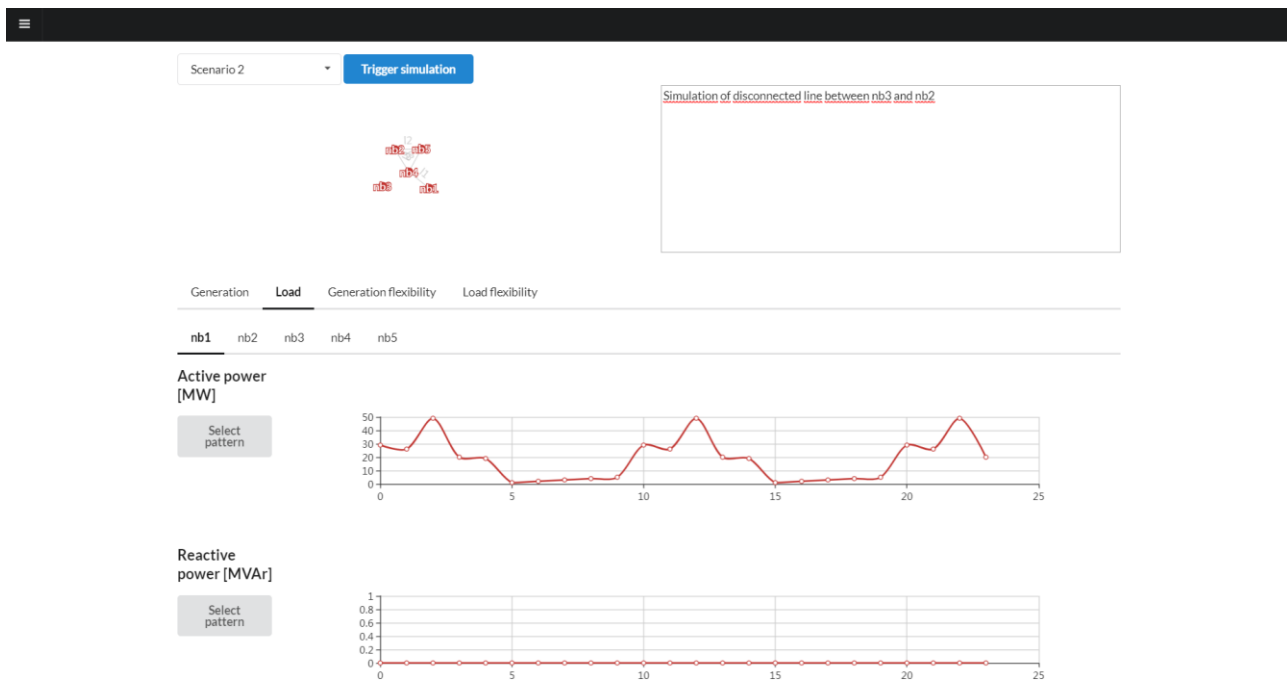


Figure 27 – Screenshot of the simulator for optimizing Multi-Objective Energy Systems [12], basis of the Grid planning assistant

The main inputs of the module are summarized in the table below.

Table 12 – Inputs required by the Grid Planning Assistant

Input	Description
Grid topology	CIM-based model of the topology including all relevant information for the simulation (buses, lines, connection matrix, accepted voltage limits)
Load pattern per bus	For each one of the buses of the grid, the operator may select a load pattern to be considered in the simulation. The integration will consider using directly historic data of the WiseGRID Cockpit to feed the simulation. In addition, data can be manually introduced, and it will be possible to add different load patterns to the base load curve (e.g. reference load of a number of domestic electric vehicles)
Generation pattern per bus	For each one of the buses of the grid, the operator may select a generation pattern to be considered in the simulation. The integration will consider using directly historic data of the WiseGRID Cockpit to feed the simulation. In addition, data can be manually introduced, and different load patterns can be added to the base generation curve (e.g. reference generation of a number of domestic PV panels)
Load flexibility availability per bus	Load flexibility availability per bus: existence of load flexibility (as such offered by other WiseGRID tools) can be added to the model to be simulated (e.g. flexibility offered by big facilities via WiseCOOP/WiseCORP)
Generation flexibility availability per bus	Existence of generation flexibility (controllable generation) can be added to the model to be simulated (e.g. flexibility offered by a VPP)

The main outcomes of the module are summarized in the table below:

Table 13 – Outcomes of the Grid Planning Assistant

Outcome	Description
Voltage levels per bus	The simulation provides estimation of the evolution of the voltage levels per bus, enabling the identification of possible problems due to excessive voltage deviations in the simulated scenario
Power flows	The simulation also provides estimation of the evolution of the power flows between the different nodes of the grid, enabling the identification of possible problems due to congestion in the simulated scenario

3.3.5 Energy Management System for Non-Interconnected Systems

3.3.5.1 Overview

The purpose of this web-based software module is to optimize the overall performance of non-interconnected systems in liberalized market environments by increasing the RES penetration. More specifically, modules are developed in the scope of the EMS for the collection and analysis of renewable, electrical and operating data.

This module is necessary in cases where the DSO is responsible for the power generation of a non-interconnected system like, for example, an island. In such a case, DSO needs to manage the energy production of the thermal and RES units minimizing the costs and matching the energy demand. In WiseGRID this module will be particularly applicable in the pilot site of Kythnos, which is a non-interconnected system (not connected to the mainland grid of Greece) and where HEDNO is the only responsible for the management of the energy production.

The system comprises the following systems:

- **Energy Load Forecast:** based on historical data regarding the energy demand and meteorological data, as well as on measured data of the electric system, the system forecasts the energy demand during the next hours.
- **RES Production Forecast:** based on historical data regarding the production and meteorological data of the region where each station is located (cloud coverage, wind speed and direction, temperature, etc.), the system forecasts the RES production during the next hours.
- **Unit Commitment and Economic Dispatch of Conventional Production Units:** based on the load and RES forecasts, the system solves a mathematical programming problem, the output of which is the optimal operation of the conventional production units that ensures the cost effective and secure operation of the autonomous system, while minimizing the RES production curtailment.

3.3.5.2 Detailed Description

For their operation the modules are using a database for storing the required data as well the results. The data in the database are organised in the following types:

- **Static Data:** Contains the technical characteristics of the thermal generators and the RES units
- **Online Data:** Data coming from the SCADA, usually updated every minute
- **Historical Data:** recording of the production of various units
- **Results:** Results of the algorithms

The interfaces to the DB can easily be replaced by interfaces using JSON format.

The following table summarizes some key aspects of the module.

Table 14 – Key aspects

Input User Interface	WG Cockpit UI or Web Page
Input Format	ODBC/ JSON
Output Format	ODBC/ xlsx/.csv

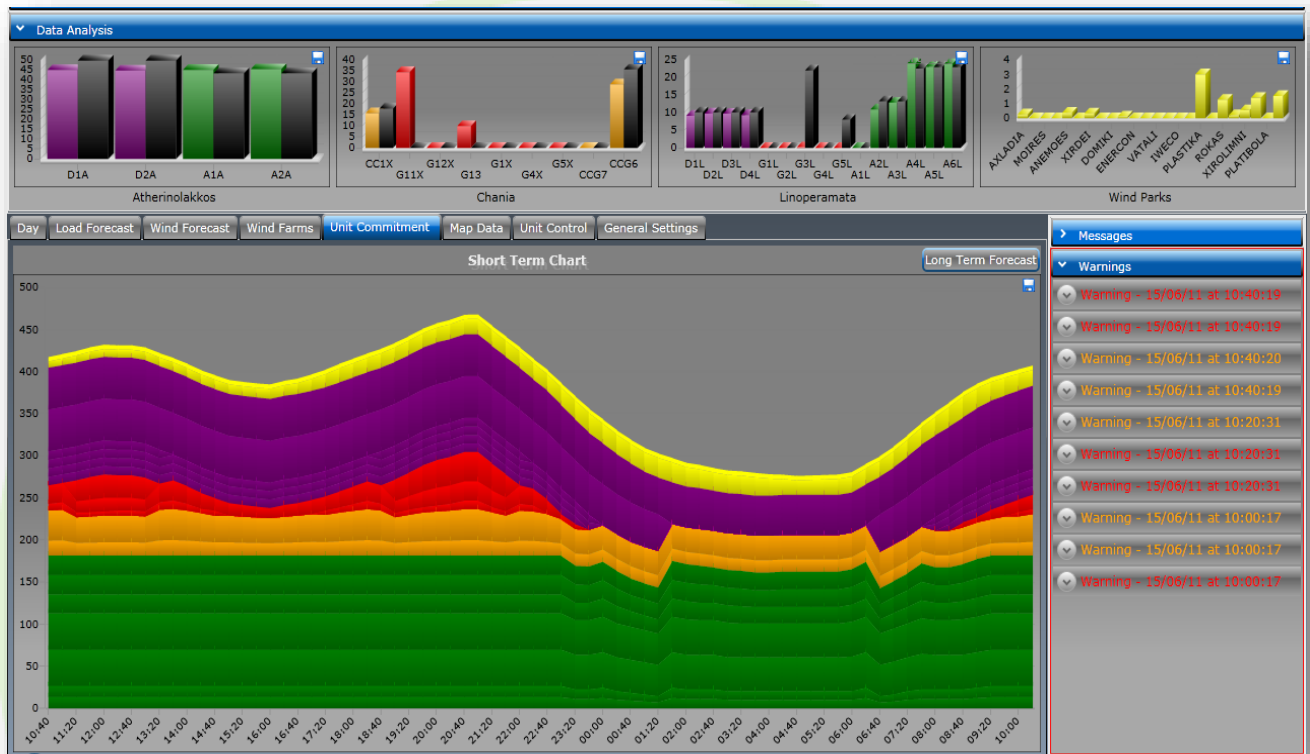


Figure 28 – EMS for non-interconnected islands, sample of the Web based UI

Script for SelectTopNRows command from SSMS

```

SELECT TOP (1000) [UnitId]
, [PowerStationId]
, [Busid]
, [UnitName]
, [TrainCode]
, [DisplayCode]
, [TypeofUnit]
, [Class]
, [RatedPower]
, [MinActivePower]
, [MaxActivePower]
, [MinReactivePower]
, [MaxReactivePower]

```

	UnitId	PowerStationId	Busid	UnitName	TrainCode	DisplayCode	TypeofUnit	Class	RatedPower	MinActivePower	MaxActivePower	MinReactivePower	MaxReactivePower	UpperOverloadin
1	13	1	17	GAS1X	GAS1X	G1X	2	25	16.2	3	11	2.5	11	14
2	14	1	441	GAS4X	GAS4X	G4X	2	24	28	5	27	-5	22.5	18.8
3	15	1	24	GAS5X	GAS5X	G5X	2	23	30	5	27	-14.1	17	28.1
4	16	1	20	CCST	CCST	CC1X	4	12	38	7	38	-7.5	25	42
5	18	1	50	GAS11X	GAS11X	G11X	2	21	59	8	52	-5	30	58
6	19	1	51	GAS12X	GAS12X	G12X	2	19	59	8	52	-5	30	58
7	9	2	28	DIESEL1	DIESEL1	D1L	3	14	12.28	3	10.1	-4.5	9.2	11.8
8	10	2	29	DIESEL2	DIESEL2	D2L	3	12	12.28	3	10.1	-4.5	9.2	11.8
9	11	2	30	DIESEL3	DIESEL3	D3L	3	13	12.28	3	10.1	-4.5	9.2	11.8
10	12	2	31	DIESEL4	DIESEL4	D4L	3	18	12.28	3	10.1	-4.5	9.2	11.8
11	4	2	25	STEAM4L	STEAM4L	A4L	1	3	25	12	22.5	-2	18.75	25
12	5	2	26	STEAM5L	STEAM5L	A5L	1	4	25	18	23	-10	18.75	25
13	6	2	27	STEAM6L	STEAM6L	A6L	1	5	25	18	23	-10	18.75	25
14	7	2	9999990	GAS1L	GAS1L	G1L	2	23	16.25	3	15	-1	13	15
15	8	2	33	GAS2L	GAS2L	G2L	2	24	16.25	3	12.8	-1	9	15
16	2	2	35	STEAM2L	STEAM2L	A2L	1	8	15	7	13.2	-5.2	11.3	15
17	3	2	36	STEAM3L	STEAM3L	A3L	1	9	15	7	13.2	-5.2	11.3	15
18	1	2	9999991	STEAM1L	STEAM1L	A1L	1	26	0	0	0	-1	3	6.2
19	17	3	84	STEAM1 ATH	STEAM1 ATH	A1A	1	1	46.5	20	43.2	-9	36	25
20	26	3	85	STEAM2 ATH	STEAM2 ATH	A2A	1	2	46.5	20	43.2	-9	36	25
21	20	2	53	GAS4L	GAS4L	G4L	2	20	14.72	3	12.8	-5	10	12
22	21	2	52	GAS3L	GAS3L	G3L	2	16	43.3	5	40	-5	26	15
23	23	1	9999992	GAS13X	GAS13X	G13	2	22	28	5	27.55	-5	35	58
24	24	3	80	dath1	d1	D1A	3	11	51.12	35	40	-9	37.25	58
25	25	3	81	dath2	d2	D2A	3	10	51.12	35	40	-9	37.25	58
26	22	2	442	GAS5L	GAS5L	G5L	2	17	27.95	5	27	-5	22.5	12
27	28	1	21	GAS7X	GAS7X	CCG7	4	7	38	15	38	-7.5	26	42
28	27	1	22	GAS6X	GAS6X	CCG6	4	6	38	15	38	-7.5	26	42

Figure 29 – EMS for non-interconnected islands, sample of the Thermal Generators.

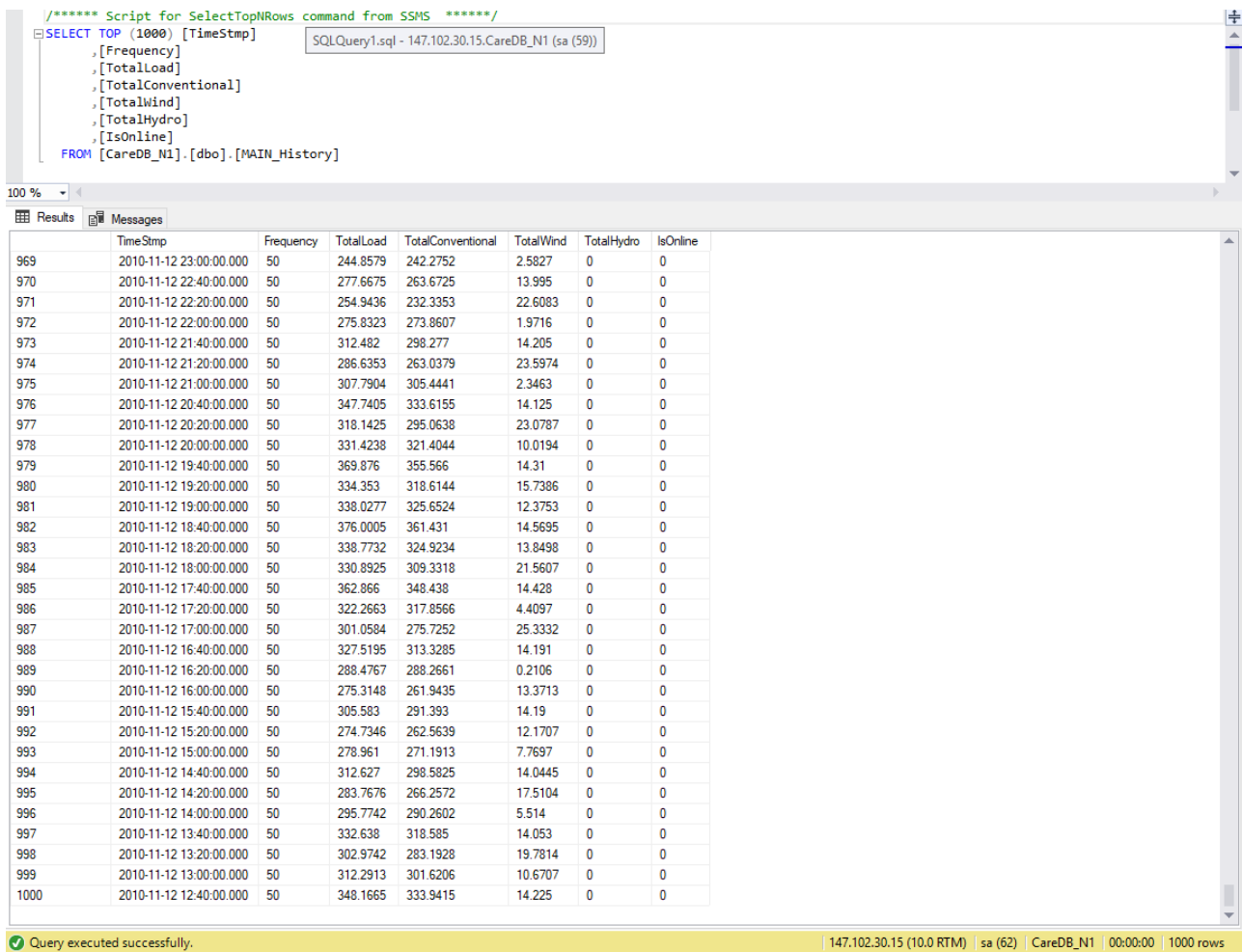


Figure 30 – EMS for non-interconnected islands, sample of historical data.

```

/***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [UC_Time_Index]
, [UC_unitid]
, [UC_model]
, [UC_status]
, [UC_gen]
, [Busid]
FROM [CareDB_N1].[dbo].[LTUC_Conventional_Output]

```

	UC_Time_Index	UC_unitid	UC_model	UC_status	UC_gen	Busid
1	1	2	1	1	11.2	0
2	1	3	1	1	11.2	0
3	1	1	1	0	0	0
4	1	17	1	1	40	0
5	1	26	1	1	40	0
6	1	4	1	0	0	0
7	1	5	1	1	21	0
8	1	6	1	1	21	0
9	1	7	1	0	0	0
10	1	8	1	0	0	0
11	1	22	1	0	0	0
12	1	20	1	0	0	0
13	1	21	1	0	0	0
14	1	23	1	0	0	0
15	1	13	1	0	0	0
16	1	14	1	0	0	0
17	1	15	1	0	0	0
18	1	18	1	0	0	0
19	1	19	1	0	0	0
20	1	9	1	0	0	0
21	1	10	1	0	0	0
22	1	11	1	0	0	0
23	1	12	1	0	0	0
24	1	24	1	1	38	0
25	1	25	1	1	38	0
26	1	16	1	1	17	0
27	1	28	1	0	0	0
28	1	27	1	1	36	0
29	12	2	1	1	11.2	0
30	12	3	1	1	11.2	0
31	12	1	1	0	0	0
32	12	17	1	1	41	0
33	12	26	1	1	41.2	0
34	12	4	1	0	0	0

Query executed successfully. 147.102.30.15 (10.0 RTM) sa (63) CareDB_N1 00:00:00 1000 rows

Figure 31 – EMS for non-interconnected islands, sample of one output file with detailed results of the Unit Commitment module.

3.3.6 Load Demand and Peak Prediction

3.3.6.1 Overview

The Load Demand and Peak Prediction module provides a long-term prediction for the load demand and peak over a predefined time horizon. The change of demand over the time horizon under study is determined on the basis of estimates of historical data and according to the user-defined parameters. In brief, the user can select between some variations of linear interpolation methods to solve the problem. More information on the available options for the user can be found at deliverable D12.2, paragraph 3.1.8, where this module will be used as part of the Investment Decision Support Manager Module.

3.3.6.2 Detailed Description

The module based on the estimate of the annual electricity demand and the load factor, gives the peak for each year of the study horizon using simple or modified linear interpolation methods. For that it requires, on the one hand, historical values of the energy demand and peak power of the previous years and, on the other, the definition of the software parameters and of other required values. For example, the user is required to choose the Load Factor that will account for the estimated demand (either as the load factor of the last reporting year or as the average of the last five reporting years). Even more, the user can determine a rate of peak load increase that applies only to the first year of the estimate.

The user-defined input data can be given either through the WG Cockpit modified appropriately for this purpose or through a separate web page. However, the historical input data need to be retrieved from the long-term, or a similar, database the input format can be JSON. An example of that is presented at Figure 32.

The output of the module can be provided in .xlsx files.

Table 15 – Load demand and peak prediction output

Input User Interface	WG Cockpit UI or Web Page
Input Format	JSON
Output Format	ODBC/ xlsx/ .csv

```
{
  "estimated": [ ],
  "past": {
    "values": [
      {
        "Year": 2005,
        "Demand": 96300,
        "Peak": 32.7,
        "LF": 0.336181978132462
      },
      {
        "Year": 2006,
        "Demand": 103136,
        "Peak": 34,
        "LF": 0.34627988181574
      },
      {
        "Year": 2007,
        "Demand": 110988,
        "Peak": 33,
        "LF": 0.38393524283935243
      },
      {
        "Year": 2008,
        "Demand": 112520,
        "Peak": 34.1,
        "LF": 0.3766788521538853
      },
      {
        "Year": 2009,
        "Demand": 117161,
        "Peak": 31.7,
        "LF": 0.42190988577272664
      },
      {
        "Year": 2010,
        "Demand": 117957,
        "Peak": 36.4,
        "LF": 0.36992887249736567
      },
      {
        "Year": 2011,
        "Demand": 120057,
        "Peak": 33.55,
        "LF": 0.40849886695384113
      }
    ]
  }
}
```

Figure 32 – Sample of the input data in JSON format for the Load Demand and Peak Prediction Module.

3.4 DATA ANALYSIS, OPERATION AND CONTROL

3.4.1 Threshold monitor

The purpose of the threshold monitor is to provide alerts upon the detection of an excess of power line or deviation of voltage on the monitored buses of the distribution grid, for the DSO operator to take the corresponding actions.

With this objective, the module will be subscribed to the data flows initiated by the Unbundled Smart Meter and SCADA wrappers (since those are the wrappers capable of providing real-time data by measurements from different nodes of the grid). Similarly to other modules, these data will be retrieved from the WiseGRID IOP and the internal ESB by the Real-time monitor module.

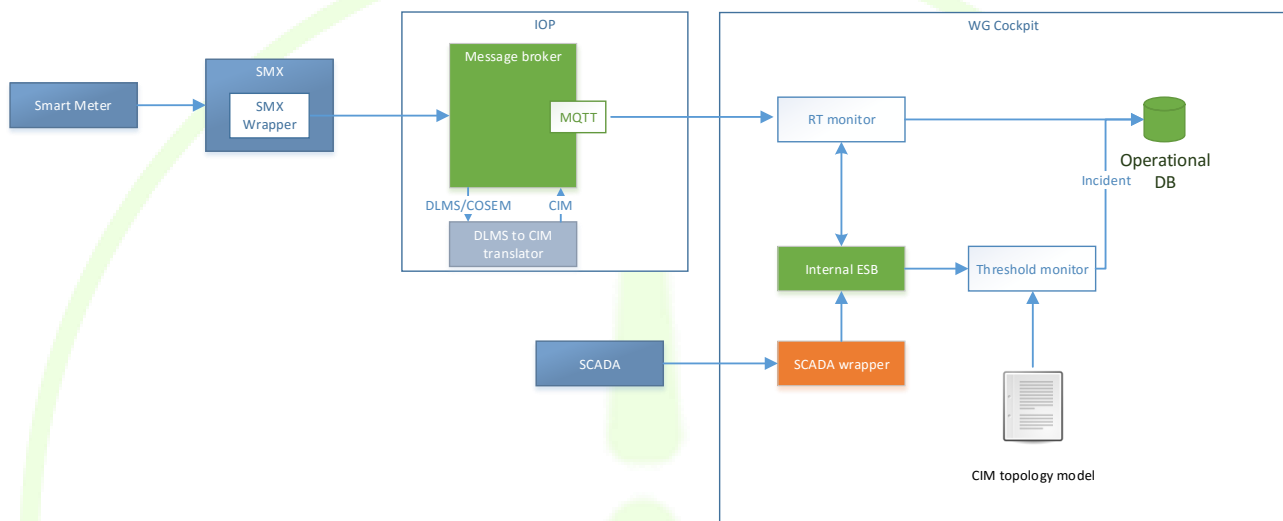


Figure 33 – Data flows subscribed by the Threshold Monitor module

Upon reception of new data (basically, voltage magnitudes measured at buses and apparent power magnitudes measured at lines or transformers), received values will be compared against the corresponding accepted thresholds (as detailed in the topology model of the distribution grid). Whenever a threshold gets exceeded, an incident will be notified to the Failure Treatment module, which will be in charge of triggering the corresponding actions. These notifications will be performed by inserting the proper incident document in the operational database. Information to be detailed in the incident includes:

- Identification of the monitored element (smart meter or SCADA sensor)
- Magnitude whose threshold has been surpassed (power or voltage)
- Actual measured value
- Upper threshold for the magnitude
- Lower threshold for the magnitude

```

{
  "system" : "threshold_monitor",
  "type" : "threshold_surpass",
  "details" : {
    "id" : "BBB5979",
    "magnitude" : "power",
    "value" : 1500,
    "threshold_upper" : 1450,
  }
}
  
```

```

    "threshold_lower" : 0
  }
}

```

3.4.2 Outlier detector

The purpose of the outlier detector is to implement a *control chart* over the voltage and frequency magnitudes that are measured at different sections (nodes) of the distribution grid.

The module will be subscribed to the data flows initiated by the Unbundled Smart Meter and SCADA wrappers (since those are the wrappers capable of providing real-time data by measurements from different nodes of the grid). These data will be retrieved from the WiseGRID IOP and the internal ESB by the Real-time monitor module, whose main duty is storing it in the corresponding databases. In order to allow the proper operation of the modules analysing those flows, at least a selection of data will be republished to the internal ESB to make it available for other modules as well.

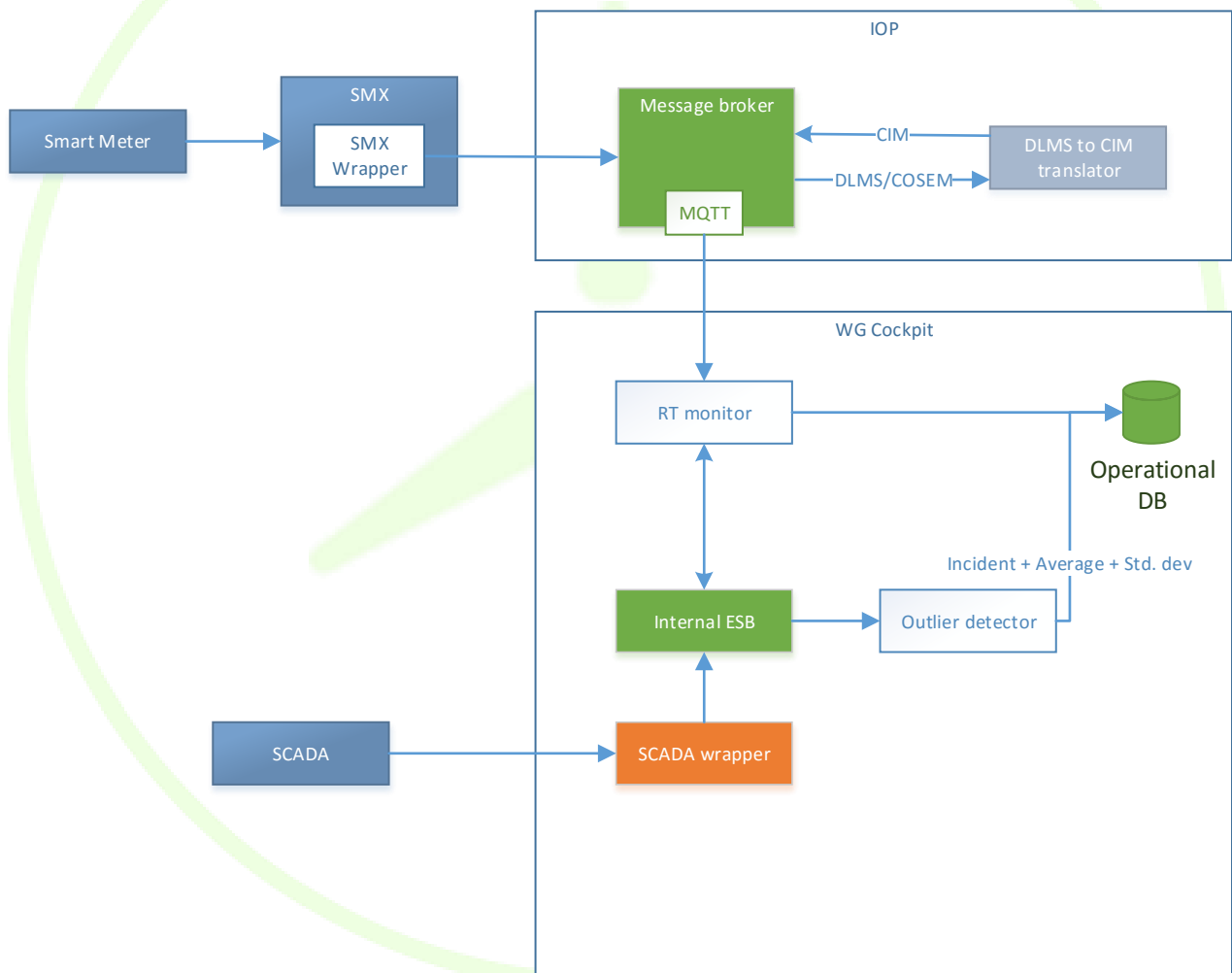


Figure 34 – Data flows subscribed by the Outlier Detector module

Upon reception of new data (basically, triplets of: bus, voltage magnitude and frequency magnitude), received values will be used to update the parameters of the control chart (average and standard deviation), and compared against the corresponding alert limits ($\pm 3\sigma$, which holds the 99.73% of samples in a normal distribution). Whenever an alert gets exceeded, an incident will be notified to the Failure Treatment module,

which will be in charge of triggering the corresponding actions. This notifications will be performed by inserting the proper incident document in the operational database. Information to be detailed in the incident includes:

- Identification of the monitored element (smart meter or SCADA sensor)
- Magnitude whose threshold has been surpassed (frequency or voltage)
- Actual measured value
- Calculated average value
- Upper threshold for the magnitude ($+3\sigma$)
- Lower threshold for the magnitude (-3σ)

```
{
  "system" : "outlier_detector ",
  "type" : "alert",
  "details" : {
    "id" : "BBB5979",
    "magnitude" : "voltage",
    "value" : 210.01,
    "average" : 226.84,
    "threshold_upper" : 239.22,
    "threshold_lower" : 214.46
  }
}
```

3.4.3 Congestion forecast

Nowadays, the growing demand of electricity – which will boost as the penetration of electric vehicles gets higher – and the installation of new elements in the distribution grid, such as distributed energy resources, may lead to scenarios where the currently deployed assets of the grid are not properly dimensioned to handle the resulting power flows. The first step towards implementing and activating mechanisms to handle this kind of issues is being able to forecast those situations in advance. With this purpose, the WiseGRID Cockpit will deploy this *Congestion forecast* module. This module is focused on steady state calculations for various input data available to the DSO operator, with the objective of producing early alerts of possible congestion issues taking place in the short-term future.

The **Constrains MAnagement Application** (CMAP) module will need input data files and will provide output data files with a list of constraints.

The two lists are for:

- The case with basic topology (the topology on normal grid status projected on forecasted production and consumption).
- The case with improved topology, where some predefined changes in topology are considered for the power flow as forecasted.

Input data files for the load flow calculation (Data collection in the Figure below):

- Topology and parameters file from WG COCKPIT (considered in figure below also as "normal grid status")
- Forecast of consumption in all nodes ($P_c(t)$, $Q_c(t)$ series).

- Forecast of all production, including on renewables ($P_g(t)$, $U(t)$ series);

Output data files from the Load flow calculation:

- P , Q , U , θ in all nodes.
- P , Q , I on each line and transformer.

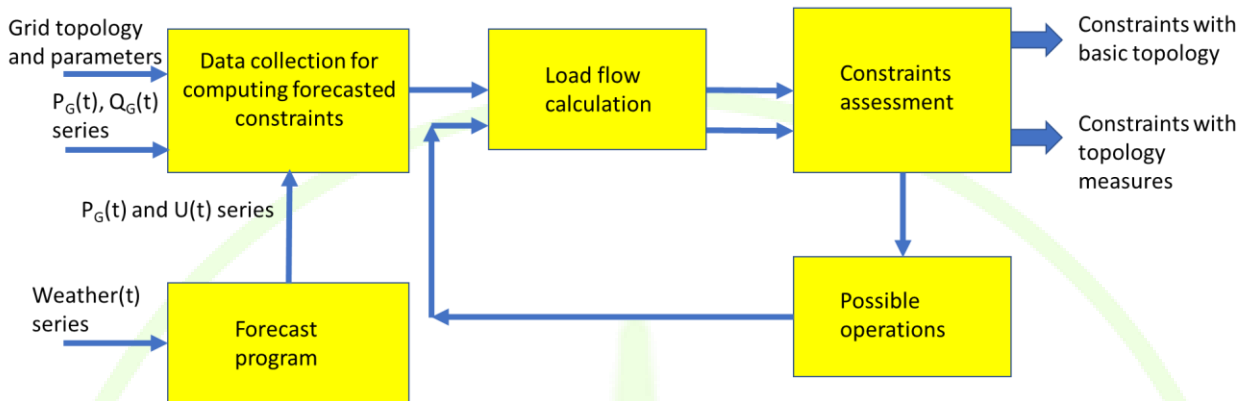


Figure 35 – The workflow of the CMAP module responsible for constraints management

The horizon of the future depends on the horizon for the forecast. It is considered as standard horizon a 24 hours timeframe, in steps of one hour. Shorter periods can be also assessed.

For a specific timeframe of K hours (e.g. $K=24$), as the project receives forecasts based on one hour steps, it means that the set of calculations considers K different steady state power flows regimes. For instance, if the system receives at hour 22:35 a forecast for 24 hours, meaning at hour 01:00, 02:00... 24:00 of next day, a corresponding number of 24 steady state regimes (load-flows) will be generated for each one hour. The forecast should be received with more than one hour before first assessed congestion period.

This procedure can be described formally as follows:

- At $t = T_0$, the constraints management application (CMAP) asks to WG COCKPIT for the grid status by writing in a specific subdirectory a CMAPtoWG COCKPIT_req.txt file; after a certain period WG COCKPIT writes a WG COCKPITtoCMAP_ans.txt file with all grid status data written in the text (format to be refined); the request is honoured in $\Delta t_1 < \Delta t_{1WG Cockpit_timeout}$, which means no later than $\Delta t_{1WG Cockpit_timeout} = 1$ to 2 minutes after the request (timeout).
- CMAP selects consumption and production points and sends to the forecast program (FOP) a request for a forecast, by writing in a specific subdirectory a CMPAtoFOL_req.txt file; after a certain period FOP writes a FOPTtoCMAP_ans.txt file with all forecast data (generation and consumption) written in the text (format to be refined); the request is honoured in $\Delta t_2 < \Delta t_{2fop_timeout}$, which means no later than $\Delta t_{1fop_timeout} = 1$ to 2 minutes after the request (timeout is 1 to 2 minutes).
- CMAP prepares an input file for the load flow calculation application (LFC) and sends to LFC a request for a load-flow calculation, by writing in a specific subdirectory a CMPAtoFOL_req.txt file; after a certain period LFC writes a LFCTtoCMAP_ans.txt file (or specific results of LFC, which will be converted in a LFCTtoCMAP_ans.txt file (formats to be refined); the request is honoured in $\Delta t_3 < \Delta t_{3lfc_timeout}$, which means no later than $\Delta t_{1lfc_timeout} = 30$ to 45 seconds after the request (timeout is 30 to 45 seconds); the load-flow is made for each time interval of the forecast (max 24 runs).
- CMAP sends the results to the Constraints manager (CnsMng) the output file.
- CnsMng selects the nodes and lines where constraints are violated (U outside boundaries and/or current I greater than a limit) and give a report as a file with the name CnsMngtoCMPA_ans.txt;

- CMAP analyses if there are constraints, and if so, a set of usual operations are simulated by CMAP, by producing a set of additional scenarios CMPAtoFOL_req.txt, which are sent to LF and then to CnsMng. For each scenario CMPA receives a report in a CnsMngtoCMPA_ans.txt file.
- CMAP makes a final report showing the nodes and lines where there are constraints, within a file CMAP_ans.txt, which is written in a specific subdirectory, to be read by WG Cockpit.

The following time-flow applies to the whole process:

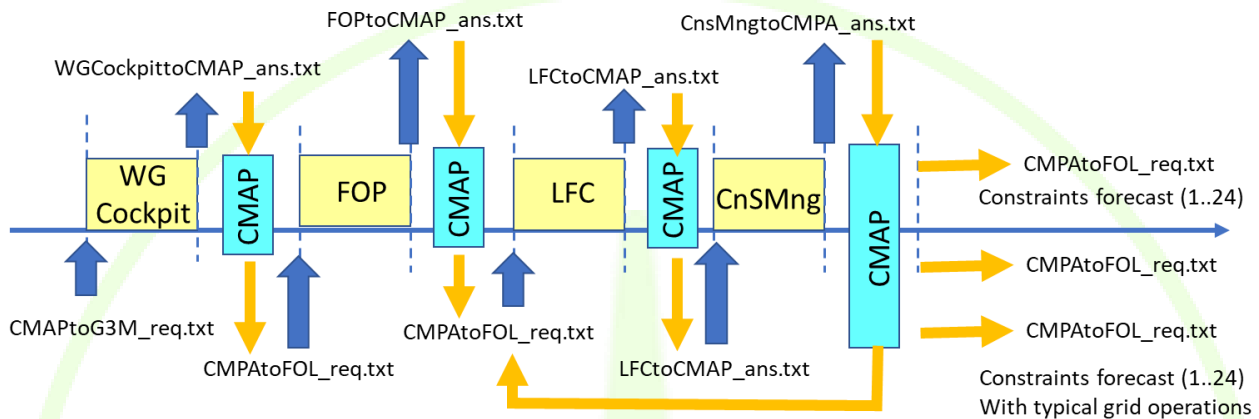


Figure 36 – Time flow

For a forecast of 24 hours, a complete process, without grid operations will take maximum:

$$T_{\max} = 2 + 2 + 0.75 \times 24 = 22 \text{ minutes}$$

Where 0.75 means 45 seconds, or 3/4 minutes.

For each additional grid operation, it is used additional time

$$T_{\max_go} = 18 \text{ minutes}$$

It is advisable to have at least two different grid operations (usual operations of the grid operator) parametrized in the CMAP, such that in a full hour the process can have:

$$T_{\text{tot_max}} = T_{\max} + 2 \times T_{\max_go} = 22 + 2 \times 18 = 58 \text{ minutes}$$

For more grid operations to be tested it is needed a quicker LFC run. For reasons of stable interconnection, for a medium size MV or LV grid it is possible to go down to 30 seconds per LFC run + CMAP assessment and files generation for process control. Moreover, the maximum time of interaction with WG COCKPIT and FOP may also be reduced to one minute.

A minimum time for constraints plus four different grid simulated operations may give:

$$T_{\max} = 1 + 1 + 0.5 \times 24 = 14 \text{ minutes}$$

$$T_{\max_go} = 0.5 \times 24 = 12 \text{ minutes}$$

$$T_{\text{tot_max}} = T_{\max} + 4 \times T_{\max_go} = 14 + 4 \times 12 = 62 \text{ minutes}$$

which is more than one hour.

This finding shows that a robust operation requests that the whole process should be started in best case every two hours, to allow up to eight different grid simulated operations, which is enough complex to help the grid operator with valuable support.

Moreover, the entire set of calculations may first start with the (emerging) next hour and all its related possible different grid operations. This may give answers for the next hour in:

$$T_{\text{tot_max_1h_low}} = 1 + 1 + 0.5 \times 8 = 6 \text{ minutes}$$

In fast mode (all the timings at the lowest values) or in case of higher values in:

$$T_{\text{tot_max_1h_high}} = 2 + 2 + 0.75 \times 8 = 10 \text{ minutes}$$

This time of reaction is fast enough to give in-time data to the grid operator in WG COCKPIT platform.

The grid simulated operations will be in of the following types:

- Change the place for sectioning an MV loop;
- Modify tap changer to a transformer
- Change the voltage set-point to a PV production plant which is controlling the voltage through reactive power.

The chosen operations are based on grid operator experience.

3.4.4 Grid fault manager

3.4.4.1 Failure treatment

WG Cockpit will integrate a component for incident management allowing creating and monitoring incidents notified by other modules of the platform. This module will allow generalization and customization of the process to be performed upon an incident by using a BPMN engine as its core.

3.4.4.1.1 BPMN

Business Process Modelling Notation (BPMN) is an international standard for business process modelling. It enables the graphical representation of a workflow related with a business process of any kind.

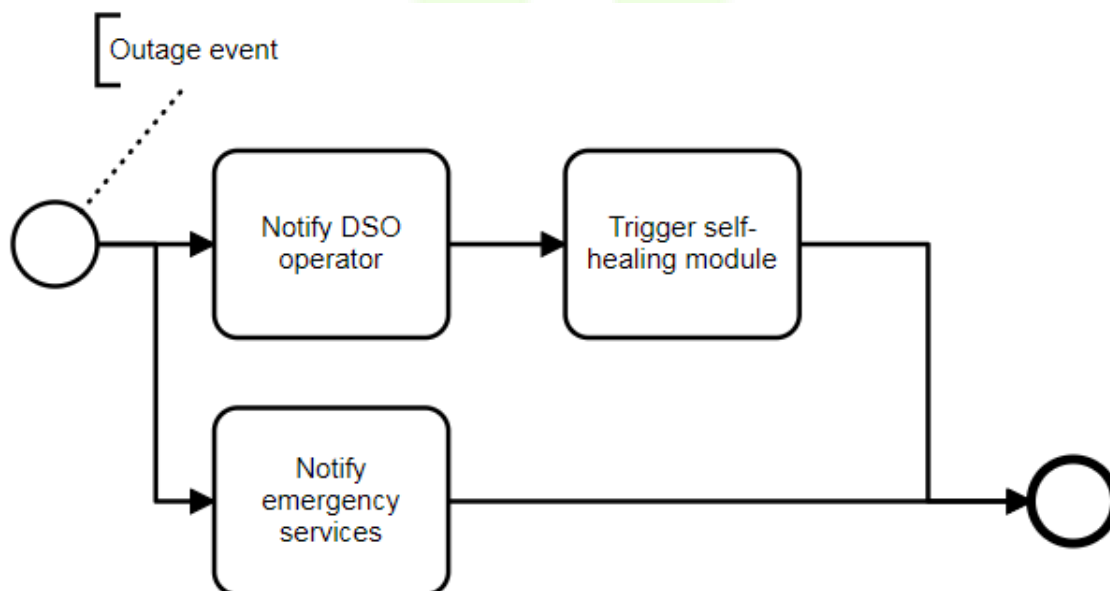


Figure 37 – Example of a simple BPMN workflow

Within the failure treatment module, it provides a consistent, versatile mechanism to define the actions that need to be performed upon detection of an incident. This approach presents very clear advantages:

- Versatility to define actions to be taken either within the scope of the WiseGRID Cockpit tool, as well as outside the scope (manual actions or automatic notification to third party systems)

- Freedom to the DSO operators to define new workflows or modify existing ones without requiring changes in the code of the application

Some examples of the capabilities of this approach may include:

- Notify incidents by email
- Request human operator to perform manual tasks (e.g. notify Public Safety Units)
- Request human operator to provide extra information to be registered together with the incident
- Automatically notify other modules of the platform (e.g. register new maintenance tasks, trigger Demand-Response campaigns, trigger self-healing module)

Flow modelling

The DSO operator has freedom to define the workflow to be executed for each type of incident. Versatility is provided by the BPMN core, enabling the definition of workflows with:

- Conditional paths: some of the tasks will be executed only if the defined condition is met
- Parallel paths: tasks without dependencies among each other can be executed in parallel
- Wait for previous tasks to be completed: the workflow may be interrupted until the required previous tasks have been completed
- Triggering new workflows depending on the result of previous ones

3.4.4.1.2 Main actions enabled by the BPMN core

Scripting

Custom code can be executed as one of the steps. This may be used to automatically process information related to the incident (e.g. retrieve further details from database if required)

Requiring actions from human operator

At some cases, it may be required the manual intervention of the operator to provide further details of the incident, or even to take actions that cannot be automated (e.g. inform affected customer by phone). The BPMN engine allows the definition and registration of such tasks

Notification via email

HTTP POST

This is the most versatile action, since it provides a generic mechanism to send information to external software systems. Examples of further actions enabled by this feature include:

- Triggering messages to the internal ESB of WiseGRID Cockpit to be processed by other module
- Creating new maintenance tasks using the REST API of the Maintenance Manager
- Send information to any third party system offering a REST API
- Open to new public notification systems (e.g. public messaging Apps)

3.4.4.1.3 Integration within WiseGRID Cockpit architecture

This component will take benefit of the DDP protocol [13] to monitor changes on a specific incidents collection of the operational database, automatically being notified of the appearances of new incidents. Upon registration of a new incident, all necessary details will be pushed to the corresponding working flow of the BPMN engine, depending on the type of incident.

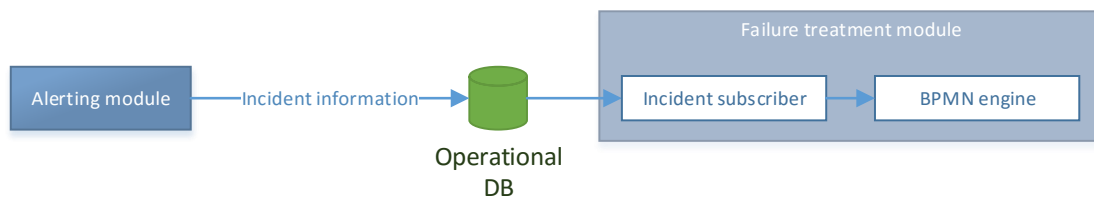


Figure 38 – Integration of alerting modules with the Failure Treatment Module

The incidents collection will hold the documents with the details of the notified incidents. The absolute minimum information required is **type**, since this property will be used to trigger the corresponding workflow in the BPMN engine. Further properties can be used to model the data related to the incident, and will depend on the actual incidents being modelled. An example is presented below:

```

{
  "system" : "congestion_forecast",
  "type" : "congestion",
  "details" : {
    "start" : "2018-01-01T12:00:00.000Z",
    "end" : "2018-01-01T14:00:00.000Z",
    "congestion_point" : "SUB0001",
    "required_flex" : 20000
  }
}
  
```

3.4.4.2 FLISR module and Fast power flow module

It is known that for a system to be highly reliable and fault tolerant, in addition to multiple redundant paths, it is paramount to have smart strategies such as fault location, isolation and restoration (FLISR) to manage redundancy. In this regard, the purpose of this module is to automate the logic for optimal service restoration of the distribution system according to a set of operational criteria after the location of a fault on the network. Besides, this module can be also used whenever required as an optimal network reconfiguration tool suggesting an optimized network topology. This module leverages a fast power flow algorithm specifically designed to perform a faster evaluation of operational criteria in radial networks.

3.4.4.2.1 Overview

As explained in Deliverable 12.2 “RESCO services and Advanced models for smartening the distribution grid”, this module comprises two stages:

- At the first stage, the submodule named “Fault location” provides the following functionalities:
 - a) Verifies the occurrence of faults (functionality F1) and
 - b) provides the faulted line segments in the network (functionality F2)
 by combining the status information of the directional fault pass indicators (DFPIs), located in some points of the network, with the fault distance information coming from the fault distance relays, located generally at the substation.
- At the second stage, the submodule named “Service Restoration”, suggests an optimal reconfiguration of the topology of the network necessary to
 - a) isolate the fault (functionality F3) and
 - b) restore the service in the affected area (functionality F4).

Specifically, the algorithm provides the switching scheme indicating the recommended status of

switching points or switching elements of the MV network for optimal operation according to predefined restoration criteria. It also provides the values obtained with respect to these optimization criteria, and the sequence of switching operations required to re-configure the network from the current (faulty) state to the optimized one.

This second submodule may also be used as an optimizer of the network configuration anytime, lowering energy losses and meeting current constraints when no faults are present in the network.

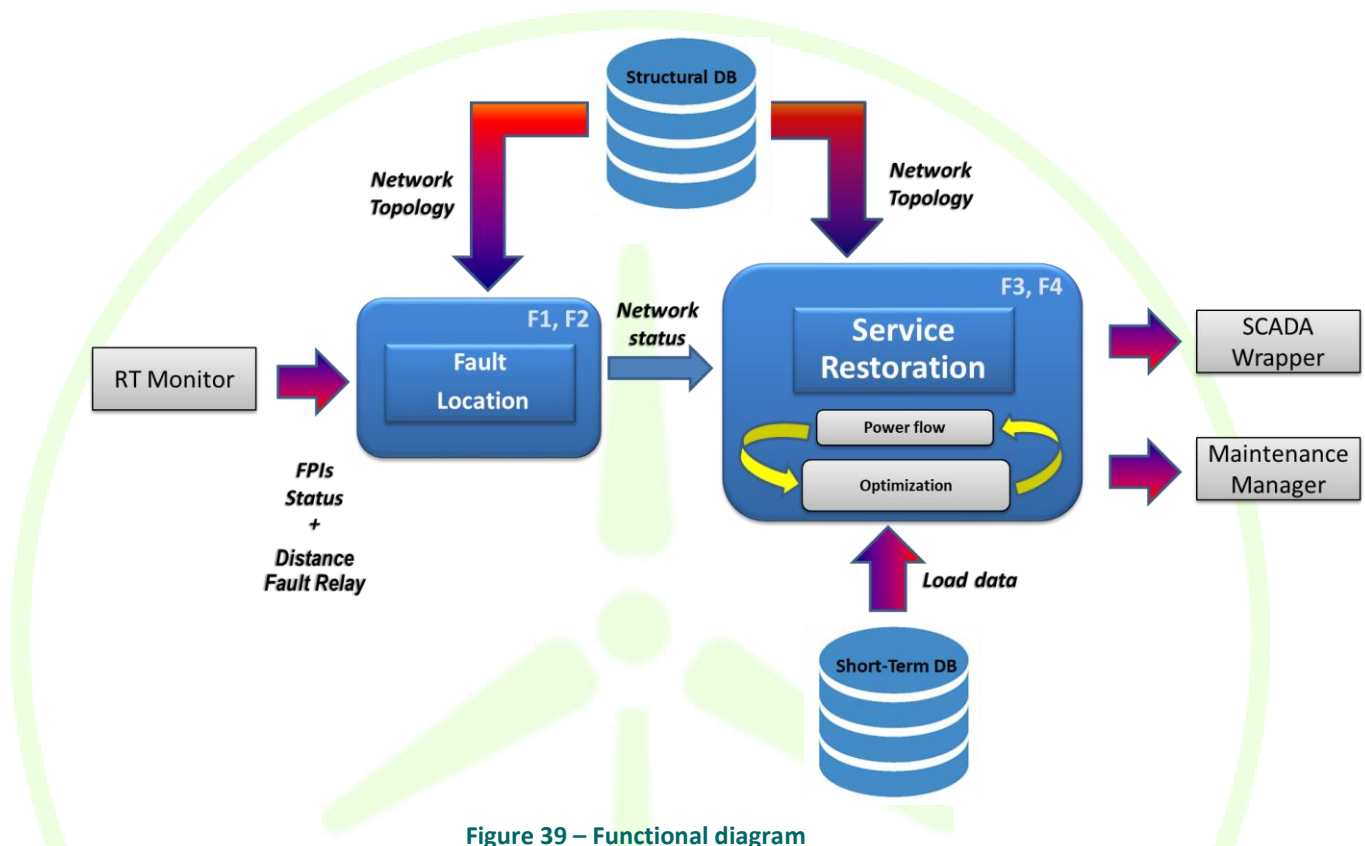


Figure 39 – Functional diagram

This module will be executed whenever a fault is detected in the distribution grid (e.g. when the safety elements in the substations get triggered), in order to calculate the optimum operation of reconfiguration to be performed in order to restore the service as fast as possible while minimizing the impact.

3.4.4.2.2 Composing submodules

The main functionalities and characteristics of these submodules has been explained in D12.2 so in this section there will be only explained their implementation in the WG Cockpit.

3.4.4.2.2.1 Fault Location submodule

Inputs

The required data inputs are gathered on the following table:

Table 16 – Inputs required

Data	Description	Type	Use	Origin
Network topology and parameters	Connectivity of network nodes, branches and switches	<i>to be defined,</i>	Configuration	Structural DB
	Location of fault pass indicators			
	Line segments impedance			
Switching status	Current status of switches	Vector of boolean values	Running	RT monitor
Fault information	Status and directionality of FPIs	vector of integer values		RT monitor
	Distance (ohms) from fault distance relays ¹	Vector of float values		RT monitor

Outputs

The outputs of localization algorithm are summarized on the following table:

Table 17 – Outputs

Data	Description	Type	Destination
Faulted line segments	Faulted network line segment(s) ²	Integer value(s)	Service Restoration module

3.4.4.2.2.2 Service Restoration Submodule

Inputs

The required data inputs are gathered on the following table:

Table 18 – Inputs

Data	Description	Type	Use	Origin
Network topology and parameters	Connectivity of network nodes, branches and switches.	<i>to be defined</i>	Configuration	Structural DB
	Location of loads			
	Location of power sources (head of feeders, distributed generators,...)			

¹ Fault distance relay is assumed to be located at the head of the substation output feeders.

² Depending on the actual DFPI and distance relays deployment, there is the possibility that the algorithm proposes several candidates for faulted line segment.

	Nominal power of power sources			
	Type ³ of switching points			
	Line segments impedance			
	Line segments current capacity			
Network operational state	Current status of switches	Vector of boolean values	Running	RT monitor
	Current load and distributed generators power	Vector of float values		RT monitor
	Faulted line segments	Vector of integer values		Fault location submodule

Outputs

The outputs of the Service Restoration Submodule are summarized on the following table:

Table 19 – Outputs

Data	Description	Type	Destination
Connection scheme	Optimal connection scheme suggesting switching points status to be changed after network restoration	Integer vector/bool vector	SCADA Wrapper / Maintenance manager
	Switching sequence from current (faulty) topology to of the optimal connection scheme	Integer vector	SCADA Wrapper / Maintenance manager
FITNESS	Final value (optimal) of the fitness function	Float value	SCADA Wrapper / Maintenance manager
Criteria values	Final values with respect to every optimization criteria: <ul style="list-style-type: none"> • Non restored power (MW) • Power losses (MW) • Voltage deviation (%) • Number of switches to be operated • Current capacity of branches (%) • Load imbalance (%). 	Float vector	SCADA Wrapper / Maintenance manager

³ Operated either manually or remotely.

3.4.4.2.3 Configuration

Service Restoration Submodule

This sub-module requires some configuration parameters. Operational ones must be specified each time. User associated parameters are specific of each user:

Table 20 – Operational parameters

Operational parameters	Description	Type
Criteria weights	Weights assigned to each criterion of the fitness function ⁴ : <ul style="list-style-type: none"> • Power losses (MW) • Voltage deviation (%) • Number of switches to be operated • Current capacity of branches (%) • Load imbalance (%). Sum of all weights must be 1.	Float vector
Voltage limit	Maximum deviation from nominal voltage admitted. Default: 7%.	Real value
Voltage violations	Quantity of node voltage violations allowed for a solution to be considered a feasible solution	Integer value
Remote	Indicates whether reconfiguration is done with all the switches or only with remotely controlled ones	Boolean value

Table 21 – User associated parameter

User associated parameter	Description	Type
GA parameters	Configuration of algorithm parameters (operators such as selection, mutation, crossover, ...) related to the GA genetic algorithm ⁵	Enumerate

3.4.4.2.4 Formal description of API

Subscribed to: FLISR (Fault Location Isolation Service Restoration)

⁴ The weight for the criterion of Power Not Supplied (PNS) is fixed to 1, and only the remaining criteria are allowed to be weighted.

⁵ Advanced configuration, recommended only for expert users.

Payload:

client: identifier of the client that ask for FLISR

Configuration:

criteria_weights: array of weights assigned to each criteria of the fitness function (excepting Power Not Supplied).

voltage_limit: Maximum deviation from nominal voltage allowed.

voltage_violations: total quantity of node voltage violations allowed.

remote: boolean indicating reconfiguration with all switches (false) or only with remotely controlled ones (true) .

Example:

```
[{
  "client_id" : 2,
  "configuration" : [{
    "criteria_weights": [ 0.2, 0.2, 0.2, 0.2, 0.2],
    "voltage_limit": 0.07,
    "voltage_violations" : 5,
    "remote" : true,
  }]
}]
```

Response payload:

```
[{
  "client_id" : 2,
  "connection_scheme" : [{10, OFF}, {5, ON},{7, ON}, {1, OFF}]
  "FITness" : [-4.564]
  "criteria_values" : [3.2, 0.08, 0.12, 4, 0, 0.56 ]
}]
```

3.4.5 Ancillary services market hub

This interface defines the interaction between the DSO (WG Cockpit) and the rest of the applications which are in position of providing ancillary services (congestion and voltage support, related to explicit demand response campaigns as discussed in WP10 [14]). This interface is based on the USEF specification [15]. The WiseGRID project will implement a market-based approach including several aggregators in the loop for delivering Ancillary Services to the DSO.

USEF already defines the messages to be exchanged in the form of XSD files (XML messages). Those messages will be exchanged through the WiseGRID IOP, and are therefore defined in detail in the corresponding deliverable D4.2 [16]. This section provides an overview of the interactions that will happen in the context of the WiseGRID Cockpit.

3.4.5.1 Detection of congestion problems

The different WiseGRID Cockpit modules collaborate to allow DSO to assure smooth operation of the grid, assisting the operator upon the occurrence or forecast of different problems. One of those modules is the *Congestion Forecast* module, in charge of analysing historical and current data from the different sensors of the grid in order to estimate the power flows in the short-term future and provide alerts of possible future congestions.

When such an alert is produced, the *Failure Treatment* submodule will receive it and trigger the actions configured by the DSO operator. One of those possible actions to deal with the congestion that is enabled by the WiseGRID project implies buying flexibility from an Ancillary Service Market, where several aggregators (targeted by different application of the project) will participate.

3.4.5.2 WiseGRID Ancillary Service Market with multi-actor interaction

If the DSO operator decides to activate the participation in the Ancillary Service Market, the Failure Treatment module will post all the information about the congestion forecast to the *Ancillary Services Market Hub* module. This information is performed by posting a message to a dedicated queue of the internal ESB, with the necessary information to compose the necessary Flexibility Requests:

- Identification of the Congestion Point.
- Time period where congestion is identified.
- Required flexibility per Program Time Unit (PTU).

Once the Ancillary Services Market Hub receives the information of the Congestion Forecast, the interaction with the other applications able to provide flexibility is started. This interaction happens by posting and reading messages from the WG IOP, as specified in D4.2 [16]. The summary of these interactions is:

1. Upon detection of a problem, the Ancillary Services Market Hub detects a problem in the grid and publishes a request. This request needs to reach each of the aggregators participating in the platform.

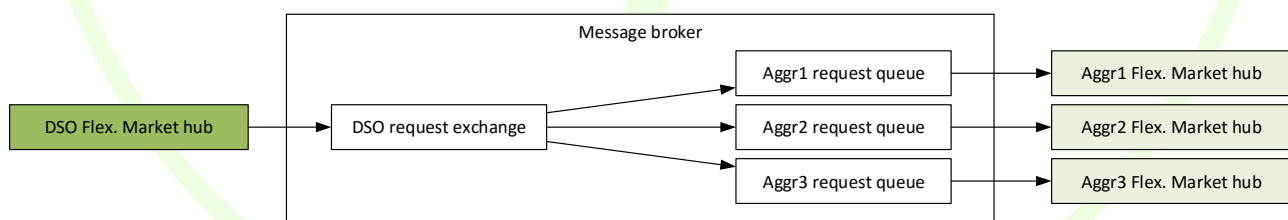


Figure 40 – Flexibility request

2. The aggregators can post an offer, describing up to which extent they can support the DSO, and the corresponding price.

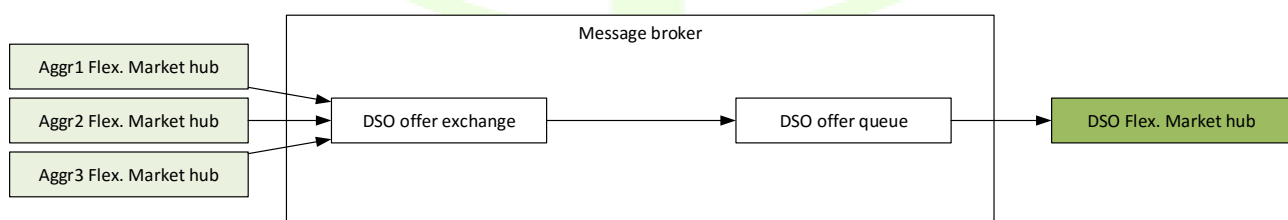


Figure 41 – Flexibility offer reception

3. DSO selects the set of offers to be accepted, and sends the order to the corresponding aggregators.

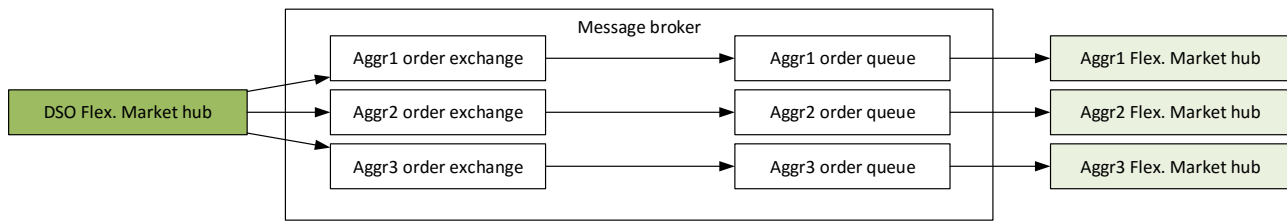


Figure 42 – Flexibility offer selection and flexibility order

This implies the development of 2 generic modules implementing the logic and allowing applications to interact in the ancillary services market:

1. DSO Flex. Market hub: module implementing the DSO-side logic (to be used in the WG Cockpit)
2. Aggr. Flex. Market hub: module implanting the aggregator-side logic (to be used in the WG StaaS/VPP, WiseCOOP and WiseEVP)

3.4.5.3 Selection of offers

After step number 2 in the process described below, the WiseGRID Cockpit will be in possession of all offers provided by the different aggregators participating in the Ancillary Services Market. Selection of offers will be performed as follows, for each one of the PTUs where congestion was identified:

1. Different groups of offers will be calculated (combination over N offers starting with selection of 1 element per group and finishing with selection of a group with all N offers). Thus, the number of possible combinations to be evaluated will be:

$$combinations = \sum_{j=1}^N \frac{N!}{j! (N-j)!}$$

For example, upon reception of 5 offers, 31 different combinations will be evaluated.

2. For each combination, the combined flexibility and price will be calculated by simply adding the flexibility and price of each offer composing the combination.
3. Offers not reaching the minimum required flexibility (as posted in the flexibility request) are dismissed.
4. Remaining combinations are ordered accordingly to its combined price (lower price first). The lowest price combination gets selected. At this point, pricing limitations may be applied.
5. Orders will be posted to the aggregators taking part of the selected combination.

3.4.6 Demand and production forecast services

The demand and production forecast provider is a wrapper of the forecast algorithms for an easy integration with the Cockpit tool.

The forecasting module developed in the WiseGRID project could be seen as a function that forecasts the next desired values taking into consideration historical data and additional data such as a calendar (with working days, holidays and weekends), exogenous variables (weather) and some client parameters.

Figure 43 contains the structure defined to integrate demand forecast and renewable forecast services into the Cockpit developments. The forecast server will communicate with clients through the central platform server, specifically, by means of the RabbitMQ services in the central control platform. Thus, the forecast server will include an AMQP client with two main objectives: route client request to the RPC forecasting functions and call for the RPC function which obtains the weather forecast.

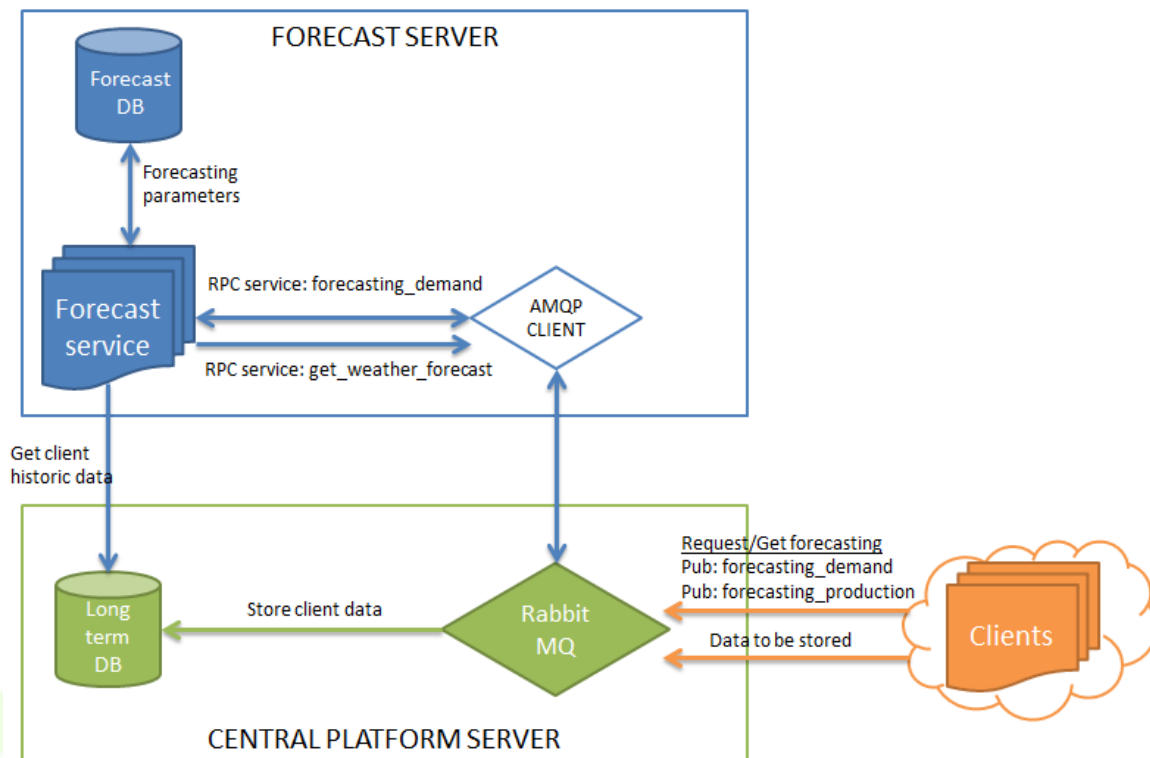


Figure 43 – Forecast server structure integration schema

Based on this architecture, the implemented developments will support the next general procedure.

1. The Cockpit tool (called client or tenant) requests the forecasting profile calling a Remote Procedure Call function with some required parameters (more details in following sections).
2. The forecast service will receive the forecasting request, by means the AMQP client, and will perform the forecasting based on historical data (long term data base) and weather forecast if proceed.
3. Once the forecasting service has completed the forecast request, taking into consideration parameters included in the request message, the forecast service will return as a response the results to that specific client.

The forecast database is in charge of storing every model and parameter directly needed for the training of the algorithm model and for the forecast. The Long Term database is needed for storing all the historical data about client consumptions and plant production as well as historical weather values to be taken into account for the forecasts. The characteristics of the historical data will affect the forecast output requirements. As an example, the forecasting algorithm will be not able to obtain a forecast with fifteen minutes detail if the historical data does not have at least the same time granularity.

Concrete to the input arguments, for each forecast to be generated, this module will take into consideration next inputs from the client:

- The client identifier. One client identifier is associated to an aggregation thus the same client application can request a forecast under different client identifiers.
- The period of time between two consecutive forecasting values.
- The total window horizon for the forecast output.

The algorithm has also a high number of internal parameters that will be automatically optimized to obtain the best prediction, such as: number of past days for training, number of past hours for prediction and variables related with internal supervised learning and optimization algorithms.

Furthermore in the Long Term database should be available information about historical data and calendars. In addition, it is mandatory to have next information in the long term data base:

- The aggregation of Supply Points (concerning consumers and producers) associated to the same aggregation identifier.
- In the case of producers, the energy resource type of each Supply Points, since the possible renewable resource considered in the scope of the project are photovoltaic and wind power. Thus, a production forecast request can consider a mix of different renewable resources.

Finally, production forecast service request **weather forecast** to the service developed in the scope of the project. Specifically, the detailed information required from the weather forecast to the production forecast service is detailed below.

- Day and hour (type: day [dd-mm-yyyy hh:mm]). Day of the year and hour.
- Wind speed (type: int8). Meters per second.
- Wind direction: SSO, NNO, etc.
- Average temperature forecasted in Celsius degrees (type: int8)
- Maximum temperature forecasted in Celsius degrees (type: int8)
- Minimum temperature forecasted in Celsius degrees (type: int8)
- Solar radiation (type: int8). W/m²

3.4.6.1 Demand Forecast API

Provides load demand forecast for a client/tenant who ask for a prediction.

AMQP queue: "forecasting_demand"

Message properties

- reply_to: name of the queue where response will be delivered
- correlation_id: free text for query/response correlation (RPC pattern <https://www.rabbitmq.com/tutorials/tutorial-six-python.html>)
- Payload:
 - client_id: client identifier in the WiseGRID database.
 - Horizon: number of days the client wants to predict, starting from current day. From 1 to 7.
 - Period: Time period between forecast samples. 15 min / 60 min. Default 60 minutes.

Response properties

- errCode: Error code regarding possible exceptions.
- Forecast: Desired prediction formed by key value pair ("Timestamp" : Value)
 - Timestamp: UNIX time seconds (UTC).
 - Value: predicted value for the specified timestamp.
- Units: Value units.

Table 22 – Demand forecast API, error codes

Error Code	Meaning
0	No error
1	Invalid client identifier
2	No enough historical data to perform forecasts
3	Untrained forecast model
4	Unfound forecast model

5	Invalid parameter horizon
6	Invalid parameter period
7	Unauthorized access to forecast model
255	Unhandled error

Message payload examples

```
{
  "Client_id": 1,
  "Horizon": 1,
  "Period": 60
}
```

Response example:

```
{
  "errCode":0,
  "forecast":
  {
    "1507154400":22.544,
    "1507158000":21.438,
    "1507161600":12.242,
    "1507165200":12.116,
    "1507168800":10.985,
    "1507172400":12.235,
    "1507176000":9.152,
    "1507179600":58.837,
    "1507183200":65.365,
    "1507186800":22.05,
    "1507190400":38.03,
    "1507194000":8.861,
    "1507197600":1.071,
    "1507201200":15.919,
    "1507204800":20.187,
    "1507208400":16.721,
    "1507212000":9.775,
    "1507215600":2.027,
    "1507219200":4.288,
    "1507222800":3.249,
    "1507226400":6.186,
```



```

    "1507230000":2.068,
    "1507233600":3.909,
    "1507237200":2.478
  },
  "units":"kW"
}

```

3.4.6.2 Production Forecast API

Provides load demand forecast for a client/tenant who ask for a prediction.

AMQP queue: “forecasting_production”

Message properties

- reply_to: name of the queue where response will be delivered
- correlation_id: free text for query/response correlation (RPC pattern <https://www.rabbitmq.com/tutorials/tutorial-six-python.html>)
- Payload⁶:
 - client_id: client identifier in the WiseGRID database.
 - Horizon: number of days the client wants to predict, starting from current day. From 1 to 7.
 - Period: Time period between forecast samples. 15 min / 60 min. Default 60 minutes.

Response properties

- errCode: Error code regarding possible exceptions.
- Forecast: Desired prediction formed by key value pair (“Timestamp” : Value)
 - Timestamp: UNIX time seconds (UTC).
 - Value: predicted value for the specified timestamp.
- Units: Value units.

Table 23 – Production forecast API, error codes

Error Code	Meaning
0	No error
1	Invalid client identifier
2	No enough historical data to perform forecasts
3	Untrained forecast model
4	Unfound forecast model
5	Invalid parameter horizon
6	Invalid parameter period
7	Unauthorized access to forecast model
255	Unhandled error

⁶ Note that for a production forecast, a weather forecast for a specific location is needed. Clients do not need to specify its location due that this information should be available into the database.

Message payload examples

```
{  "Client_id": 1,
    "Horizon": 1,
    "Period": 60,

}
```

Response example:

```
{
  "errCode":0,
  "forecast":
  {
    "1507154400":22.544,
    "1507158000":21.438,
    "1507161600":12.242,
    "1507165200":12.116,
    "1507168800":10.985,
    "1507172400":12.235,
    "1507176000":9.152,
    "1507179600":58.837,
    "1507183200":65.365,
    "1507186800":22.05,
    "1507190400":38.03,
    "1507194000":8.861,
    "1507197600":1.071,
    "1507201200":15.919,
    "1507204800":20.187,
    "1507208400":16.721,
    "1507212000":9.775,
    "1507215600":2.027,
    "1507219200":4.288,
    "1507222800":3.249,
    "1507226400":6.186,
    "1507230000":2.068,
    "1507233600":3.909,
    "1507237200":2.478
  },
  "units":"kW"
}
```

}

3.4.7 Maintenance manager

The maintenance manager is an independent module focused on integral maintenance. The module can be hosted in the cloud (SaaS), and can be interfaced using a REST web service. It also offers web-based access to show the managed information and perform actions, and an Android App for the crew.

The module will be integrated within the WiseGRID Cockpit with the objective of providing the DSO an automated and formal platform to deal with any incident requiring manual actions and coordination of personnel from their side.

3.4.7.1 Data model

3.4.7.1.1 Incidents

3.4.7.1.1.1 Type

An important issue of Maintenance Management Systems is the kind of activities able to handle to fix or prevent incidents. These incidents or events typically are classified into two groups:

- Corrective actions: those that refer to an active problem or fault. Given a specific fault, the organization usually has to solve it within a certain amount of time (agreed with the client)
- Preventive actions those that are taken in order to prevent future problems. These actions are usually defined by the organization, based on their know-how, manufacturer recommendations, previous experiences or legal requirements (e.g. checking the components of certain devices). These actions are usually planned on a periodic basis.

3.4.7.1.1.2 Lifecycle

An incident goes through 4 different status during its lifecycle

- Notified (N): the maintenance manager has registered the incident, but it has not been attended yet
- Registered (R): the maintenance operator confirms the reception of the incident, but does not perform any action yet
- Initiated (I): the maintenance operator is working on the incident
- Finished (F): the maintenance operator has finished working on the incident

3.4.7.1.2 Tasks

A task is any action performed on an incident. These actions reflect evolution of the status, or any information provided by the crew working on the incident (free text or photographs).

3.4.7.1.3 Crew

The maintenance manager holds a list of personnel that can be assigned to work in the incidents. Each incident can be assigned to a single worker.

3.4.7.2 Internal modules

The maintenance management module internally implements the following modules:

- Maintenance database: stores all data and handles the lifecycle of the incidents
- Maintenance management UI: web interface where all functionality of the module is exposed
- Android APP: allows personnel assigned to maintenance to interact with the maintenance management module, check list of assigned incidents and update the status of those

- REST API: exposes some of the functionalities of the maintenance manager module to allow integration of third party applications

3.4.7.2.1 Web interface

The web interface is the main working interface for the personnel coordinating the maintenance in the DSO. From this interface, a general overview of the status of all assets, open incidents, planning of maintenance and evolution of the works is presented. In addition, reports can be generated to gain insight into the efficiency of the works, the times usually required to solve certain type of problems, or the distribution of the incidents according to its type, for instance.

Correctivo > Consultas

Filtros Acciones Informes

Estados: N R I F Imp

Ver	PT	Equipo	Nombre	NP	Avería Notificada	Est	Fecha Aviso	Zona	Imp.	Pri.	Restante
<input type="checkbox"/>	2012-PKG-20	C.100.30.2-4	Cajero Automático 4 - Planta -1 Acceso C/ Burgense	PKG	No comunica	F	14/08/12 11:59	PKG	NO	0	---
<input type="checkbox"/>	2012-PKG-21	C.100.40.2-2	Abrepuertas 2 - Acceso Dr. Fleming	E	Fallo lector ticket	F	14/08/12 21:43	PKG	NO	0	---
<input type="checkbox"/>	2012-PKG-22	C.100.30.2-3	Cajero Automático 3 - Planta -1 Acceso C/ San Pablo	PKG	Fallo bajante moneda	F	15/08/12 16:33	PKG	NO	0	---
<input type="checkbox"/>	2012-PKG-23	C.100.40.2-2	Abrepuertas 2 - Acceso Dr. Fleming	E	Fallo lector tarjeta	F	16/08/12 19:13	PKG	NO	0	---
<input type="checkbox"/>	2012-PKG-24	C.100.30.2-2	Cajero Automático 2 - Planta -1 Acceso CENIEH	E	Fallo puerta cajero	F	16/08/12 19:43	PKG	NO	0	---
<input type="checkbox"/>	2012-PKG-25	C.100.10.2-1	Expendedor Entrada C/Burgense	PKG	Fallo impresora	F	21/08/12 17:02	PKG	NO	0	---
<input type="checkbox"/>	2012-MIR-12	MC.5.1.2-101	Pilonas Plaza España	T	Sube muy lenta	I	22/08/12 16:10	MIR	NO	0	---
<input type="checkbox"/>	2012-MIR-13	MC.5.1.2-101	Pilonas Plaza España	T	Fallo Leds	I	22/08/12 16:13	MIR	NO	0	---
<input type="checkbox"/>	2012-PKG-26	C.100.10.2-1	Expendedor Entrada C/Burgense	PKG	Fallo barrera	F	27/08/12 09:43	PKG	NO	0	---
<input type="checkbox"/>	2012-PKG-27	C.100.10.2-1	Expendedor Entrada C/Burgense	PKG	Fallo barrera	F	27/08/12 19:49	PKG	NO	0	---
<input type="checkbox"/>	2012-MIR-14	MC.2.2.2-1	C/ Logroño - C/ Estación	E	Modificación, mejora, reforma	F	31/08/12 07:57	MIR	NO	0	---
<input type="checkbox"/>	2012-MIR-15	MC.2.2.2-1	C/ Logroño - C/ Estación	E	Lámpara mal funcionamiento/estado	I	31/08/12 08:01	MIR	NO	0	---
<input type="checkbox"/>	2012-MIR-16	MC.2.2.2-2	C/ Logroño - C/ Arenal	E	Modificación, mejora, reforma	F	31/08/12 08:05	MIR	NO	0	---
<input type="checkbox"/>	2012-MIR-17	MC.2.2.2-2	C/ Logroño - C/ Arenal	E	Avisador acústico	F	31/08/12 08:07	MIR	NO	0	---
<input type="checkbox"/>	2012-MIR-18	MC.2.2.2-3	C/ Logroño - C/ República Argentina	E	Modificación, mejora, reforma	F	31/08/12 08:10	MIR	NO	0	---
<input type="checkbox"/>	2012-MIR-19	MC.2.2.2-3	C/ Logroño - C/ República Argentina	E	Avisador acústico	I	31/08/12 08:13	MIR	NO	0	88d23h56
<input type="checkbox"/>	2012-MIR-20	MC.2.2.2-4	C/ Ronda del Ferrocarril - C/ Arenal	E	Modificación, mejora, reforma	F	31/08/12 08:18	MIR	NO	0	---
<input type="checkbox"/>	2012-MIR-21	MC.2.2.2-4	C/ Ronda del Ferrocarril - C/ Arenal	E	Lámpara fundida	F	31/08/12 08:20	MIR	NO	0	---
<input type="checkbox"/>	2012-MIR-22	MC.2.2.2-4	C/ Ronda del Ferrocarril - C/ Arenal	E	Avisador acústico	F	31/08/12 08:22	MIR	NO	0	---
<input type="checkbox"/>	2012-MIR-23	MC.2.2.2-5	C/ Ronda del Ferrocarril - C/ Estación - C/ Vitoria	E	Modificación, mejora, reforma	F	31/08/12 08:24	MIR	NO	0	---
<input type="checkbox"/>	2012-MIR-24	MC.2.2.2-5	C/ Ronda del Ferrocarril - C/ Estación - C/ Vitoria	E	Avisador acústico	F	31/08/12 08:29	MIR	NO	0	---
<input type="checkbox"/>	2012-MIR-25	MC.2.2.2-6	C/ Estación - C/ San Agustín	E	Modificación, mejora, reforma	F	31/08/12 08:33	MIR	NO	0	---
<input type="checkbox"/>	2012-MIR-26	MC.2.2.2-7	C/ Real Alcaide - C/ Leopoldo Lewin	E	Avisador acústico	I	31/08/12 08:42	MIR	NO	0	---
<input type="checkbox"/>	2012-MIR-27	MC.2.2.2-9	Gta. C/ San Francisco - Ctra. Fuente Caliente P.P	E	Lámpara mal funcionamiento/estado	I	31/08/12 09:15	MIR	NO	0	---
<input type="checkbox"/>	2012-MIR-28	MC.2.2.2-5	C/ Ronda del Ferrocarril - C/ Estación - C/ Vitoria	E	Lámpara fundida	I	31/08/12 09:46	MIR	NO	0	88d02h23

1 2

Página 2 de 2

59 Averías listadas

Figure 44 – Screenshot of the Maintenance Module Web Interface

3.4.7.2.2 Android app

The Android App is the main working interface for the field personnel in charge of working in the incidences. This application is designed as a tool that allows maintenance technicians to have simply updated information in real time in relation to their work and communicate the progress of the work done to the rest of the organization, facilitating the exchange of information among the different agents of the system during their working day.

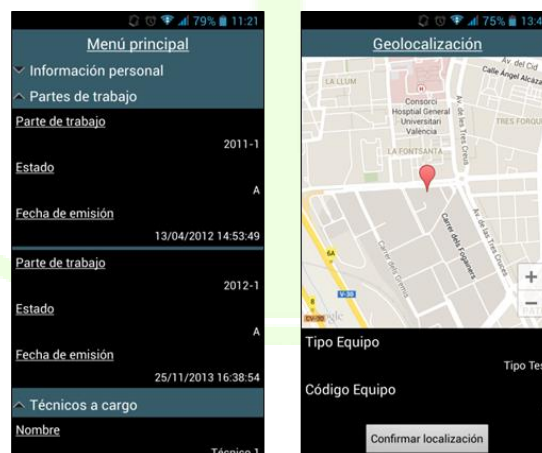


Figure 45 – Screenshot of the Maintenance Module Android App

3.4.7.2.3 REST API

The REST API is documented using swagger. This API will be used for the integration of the module within the WiseGRID Cockpit architecture. A summary of the methods is presented hereby.

Table 24 – Summary of methods

METHOD	PATH	INPUT	OUTPUT
GET	/api/incident		List registered incidents
POST	/api/incident	New incident details	Details of new registered incident
GET	/api/incident/{id}	Id of incident	Details of incident
GET	/api/incident/{id}/tasks	Id of incident	List of registered tasks related to the incident
GET	/api/crew		List of workers
GET	/api/crew/{id}	Id of worker	Details of workers
GET	/api/crew/{id}/incidents	Id of worker	List of incidents assigned to worker
GET	/api/crew/{id}/tasks	Id of worker	List of tasks performed by worker
GET	/api/device		List of registered devices
POST	/api/device	Device details	Details of new registered device
GET	/api/device/{type}	Type of device	List of registered devices for the given type
GET	/api/device/{type}/{id}	Type and id of device	Details of device
GET	/api/devicetypes		List of registered device types

3.4.7.3 Workflow

This section describes a possible workflow for the integration of the maintenance management module within a system:

Table 25 – Workflow

ORDER	ACTION	INTERFACE	DESCRIPTION
1	Register asset types	Maintenance Management UI	The organization registers the kind of assets they maintain (e.g. lines, substations...)
2	Register incident types per asset	Maintenance Management UI	Incidents in the maintenance management module are typified. Each asset type may be only assigned certain types of incidents
3	Register personnel information	Maintenance Management UI	The organization registers the personnel assigned to maintenance. An account for the smartphone app is created for each individual upon registration
4	Register assets	Maintenance Management UI / REST API	Actual instances of the assets under maintenance are registered in the system. An instance is identified by its asset type and its numeric id
5	Register new incidents	Maintenance Management UI / REST	Upon detection of a problem with an asset, a new incident

		API / Mobile App	can be registered within the maintenance management module
6	Assign personnel to an incident	Maintenance Management UI	The maintenance manager may assign personnel to deal with the incident
7	Work on incident	Maintenance Management UI / Mobile App	Personnel assigned to the incident receives the information and performs updates on the status from the mobile app. Similar work can be performed by the maintenance manager using the web UI
8	Close incident	Maintenance Management UI / Mobile App	Personnel assigned to the incident can close the incident from the mobile app when work is finished. Similar work can be performed by the maintenance manager using the web UI
9	Monitor incidents and personnel	Maintenance Management UI / REST API	Maintenance manager may monitor the status of the personnel and the incidents using the web UI. Some functionalities for monitoring are also exposed to the REST API

3.4.8 Power flow calculator/state estimator

3.4.8.1 Overview

The calculation of the power flows and the bus voltage levels of the distribution grid is an essential and fundamental part of the WiseGRID Cockpit and will serve both for the monitoring and for the planning operations of the tool. For that purpose, two separate families of algorithms have been implemented and are functioning in the tool, i.e. the Three Phase Power Flow (3PPF) algorithm and the State Estimation (SE). The necessity and functionality of both families of algorithms have been described at previous paragraphs of this deliverable and at deliverable D12.2 as well.

Essentially, the 3PPF and the SE have the same scope, that is to calculate the voltages and/or the currents of a given network topology under specific (quasi) steady state conditions. The input data of the 3PPF are the injected active and reactive power at load buses, the injected active power and voltage at generation buses and the voltage phasor at the slack bus, whereas in the case of SE, the input data may also consist of different types of measurements, such as active and reactive power flows, branch current (magnitudes or phasors), and injected currents (magnitudes or phasors). In SE, there is a distinction between: 1) monitored buses (equipped with a real measurement device), 2) non-monitored buses and 3) zero-injection buses (when there is no load/generator connected to them). The main difference between the 3PPF and the SE lies in the fact that the former requires complete knowledge of all the input parameters mentioned while the latter, using estimation theory and statistical analysis, can give results of good accuracy in cases of missing, wrong, inaccurate or noisy measurements. Therefore, the 3PPF algorithm suits better to the planning operations of

the tool, where different operating scenarios could be tested and evaluated by the operator, while the SE is ideal for the real-time monitoring operations.

3.4.8.2 Detailed Description

3.4.8.2.1 Three Phase Power Flow (3PPF) module

The 3PPF is based on a backward/forward sweep method, as it has already been presented at a previous section, in order to solve the power flow problem for a three-phase unbalanced distribution network. In general, the input data can be divided into two types:

- Network topology data.

The algorithm requires a great variety of network topology data in order to accurately calculate the electrical parameters of the grid. For example, since the LV grid is essentially unbalanced, the geometrical configuration of the overhead or underground lines needs to be defined, in order to appropriately calculate the mutual inductances.

- Measurement or scenario data.

These are the data coming from the measurements or, more likely, defined by the specific scenario. They refer to the bus data (active power, reactive power or voltage depending on the bus type), and the switch status.

As already been said, the algorithm provides the bus voltages and the power and currents for the nodes and branches.

The algorithm is an executable which reads the data from an .xlsx file and writes them also at an .xlsx file.

	A	B	C	D	E	F	G	H	I	J	K
	bus I	ground_id	connection_id	Type	P_ph.A (kW)	P_ph.B (kW)	P_ph.C (kW)	Q_ph.A (kvar)	Q_ph.B (kvar)	Q_ph.C (kvar)	Vnom(kV)
1	650	1	0	0	0,00	0,00	0,00	0,00	0,00	0,00	4,16
2	632	1	0	0	0,00	0,00	0,00	0,00	0,00	0,00	4,16
3	633	1	0	0	0,00	0,00	0,00	0,00	0,00	0,00	4,16
4	634	1	1	1	160,00	120,00	120,00	110,00	90,00	90,00	0,48
5	645	1	1	1	0,00	170,00	0,00	0,00	125,00	0,00	4,16
6	646	1	2	3	0,00	230,00	0,00	0,00	132,00	0,00	4,16
7	671	1	2	1	385,00	385,00	385,00	220,00	220,00	220,00	4,16
8	692	1	2	2	0,00	0,00	170,00	0,00	0,00	151,00	4,16
9	675	1	1	1	485,00	68,00	290,00	190,00	60,00	212,00	4,16
10	684	1	0	0	0,00	0,00	0,00	0,00	0,00	0,00	4,16
11	611	1	1	2	0,00	0,00	170,00	0,00	0,00	80,00	4,16
12	652	1	1	3	128,00	0,00	0,00	86,00	0,00	0,00	4,16
13	680	1	0	0	0,00	0,00	0,00	0,00	0,00	0,00	4,16
14	1	1	1	1	11,33	44,00	78,00	6,67	25,33	45,33	4,16
15	2	1	1	1	5,67	22,00	39,00	3,33	12,67	22,67	4,16
16	700	1	0	0	0,00	0,00	0,00	0,00	0,00	0,00	4,16
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											

Figure 46 – Sample of the input file of 3PPF - Defining the scenario data.

	B	C	D	E	F	G	H	I	J	K	L
1	t_bus	Status	type_id	Phase Conductor type_id	Neutral Conductor type_id	length (m)	Spacing_id	A	B	C	N
2	700	1	0								
3	632	1	1	5	2	609,600	500	2	1	3	1
4	633	1	1	2	2	152,400	500	2	3	1	1
5	634	1	0								
6	645	1	1	1	1	152,400	505	0	2	1	1
7	646	1	1	1	1	91,440	505	0	2	1	1
8	1	1	1	5	2	152,400	500	2	1	3	1
9	2	1	1	5	2	457,200	500	2	1	3	1
10	671	1	0								
11	692	1	0								
12	675	1	3	301	6	152,400	515	1	2	3	0
13	684	1	1	1	1	91,440	505	1	0	2	1
14	611	1	1	1	1	91,440	510	0	0	1	1
15	652	1	4	401	3	243,840	520	1	0	0	1
16	680	1	1	5	2	304,800	500	2	1	3	1
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											

Figure 47 – Sample of input file of 3PPF – Defining the topological data (branch data).

Table 26 – Data formats

Input User Interface	MS Excel
Input Format	.xlsx
Output Format	.xlsx

3.4.8.2.2 State Estimator

The SE algorithm can calculate the system (quasi) steady state values using two different methods, i.e. the Weighted Least Squares (linear and non-linear systems), and Kalman Filter (Discrete Kalman Filter for linear systems, Extended and Iterated Extended Kalman Filter for non-linear systems). The SE uses the theory of statistical analysis and solves an optimization problem, where the goal is to use available input data and estimate a system state that shall be as close as possible to the true system state, by minimizing thus the SE error. SE can be applied to medium voltage unbalanced systems.

The input data types are:

- Network topology data (i.e. the network admittance matrix).
The nodes and their interconnections, the line parameters and their configuration, other grid parameters. This data are retrieved from the structural database.
- The real-time or near real-time measurements.
The measurements of the loads (active and reactive power), power flows, injected and branch currents (magnitude or phasors) and the bus voltages (magnitudes or phasors). There are also the so-called zero-injection buses, where no load/generator is connected and the so-called “pseudomeasurements”, that may consist of historical measurements, forecasts or typical profiles.

The distinction between real-time and near real-time measurements depends on the type of used measurement devices and, more particularly, on their reporting rate.

- The accuracy of measurement devices and sensors (i.e. the measurement noise).

The accuracy is expressed in terms of standard deviation and shall be known, in order to calculate the value of the measurement noise covariance matrix of the SE.

The outputs are essentially the same as the ones of 3PPF.

The algorithm has been developed at Mathworks MATLAB software but it runs as an executable. The input data are retrieved from a MS Excel file. The **only data retrieved** from a text file are the node interconnection (start node – end node) and the line lengths. The output data are exported to a MS Excel file.

line	start node	end node	line type (overhead/cable)	code (configuration)	length (ft)
SB_01	01	02	overhead	13_IEEE_TESTsb	5280
01_02	02	03	overhead	conf602	500
02_03	03	04	overhead	conf602	2000
03_04	04	05	overhead	conf602	300
02_05	03	06	overhead	conf602	500
05_06	06	07	overhead	conf602	300
02_07	03	08	overhead	conf602	2000
07_08	08	09	overhead	conf602	300
08_09	09	10	overhead	conf602	500
07_10	08	11	overhead	conf602	300
10_11	11	12	overhead	conf602	300
10_13	11	14	overhead	conf602	800
07_12	08	13	overhead	conf602	1000

Figure 48 – Sample of input file of SE – Defining the connectivity of network buses, as well as the type, configuration, and length of the lines.

A	B	C	D	E	F	G	H	I	J	K	L
code (configuration)	R (Ohm/mi)				X (Ohm/mi)				B (uS/mi)		
conf300	1.33680E+00	2.10100E-01	2.13000E-01		1.33430E+00	5.77900E-01	5.01500E-01		5.33500E+00	-1.53130E+00	-9.94300E-01
	2.10100E-01	1.32380E+00	2.06600E-01		5.77900E-01	1.35690E+00	4.59100E-01		-1.53130E+00	5.09790E+00	-6.21200E-01
	2.13000E-01	2.06600E-01	1.32940E+00		5.01500E-01	4.59100E-01	1.34710E+00		-9.94300E-01	-6.21200E-01	4.88800E+00
13_IEEE_TESTsb	7.46E-02	0.00000E+00	0.00000E+00		7.46E-01	0.00000E+00	0.00000E+00		0.00000E+00	0.00000E+00	0.00000E+00
	0.00000E+00	7.46E-02	0.00000E+00		0.00000E+00	7.46278E-01	0.00000E+00		0.00000E+00	0.00000E+00	0.00000E+00
	0.00000E+00	0.00000E+00	7.46E-02		0.00000E+00	0.00000E+00	7.46278E-01		0.00000E+00	0.00000E+00	0.00000E+00
conf602	7.52600E-01	1.58000E-01	1.56000E-01		1.18140E+00	4.23600E-01	5.01700E-01		5.70E+00	-1.08E+00	-1.69E+00
	1.58000E-01	7.47500E-01	1.53500E-01		4.23600E-01	1.19830E+00	3.84900E-01		-1.08E+00	5.18E+00	-6.59E-01
	1.56000E-01	1.53500E-01	7.43600E-01		5.01700E-01	3.84900E-01	1.21120E+00		-1.69E+00	-6.59E-01	5.42E+00

Figure 49 – Sample of input file of SE – Defining the line parameters (resistance, reactance, susceptance).

Table 27 – Data formats

Input User Interface	MS Excel and text Editor
Input Format	.xlsx and .txt
Output Format	.xlsx

3.5 USER INTERFACE

The WiseGRID Cockpit User Interface will be the interaction point with the DSO operator. It must therefore show all relevant information compiled by all other modules present in the architecture of the application. Final objective is to offer a comfortable interface where all relevant information (current status, forecasts, monitoring data, incidents, demand response campaigns...) can be accessed in a comprehensive and simple manner.

3.5.1 User profiles

Considering that the WiseGRID Cockpit tool addresses the DSOs and will be used by DSO's users with different tasks, fields and thus needs, the tool will be adopted by providing access to interfaces for the different roles.

A first approach general/main roles has been designed, enlisting the features needed by the different kinds of personnel.

Three different user profiles are considered:

- Local personnel, requiring fine-grained details of the relevant technical aspects of the grid, for exploitation and planning purposes, potentially focusing in a specific part of the grid. This personnel is this covering a wide geographical area, and makes it necessary to provide information on all geographical area.
- Regional personnel, also requiring fine-grained details of the relevant technical aspects of the grid, this case accessing a general view of a bigger area, for grid operation and planning purposes.
- General personnel, requiring summarized overview of the status of the whole distribution grid

The following table summarizes the different interfaces of the application that will be accessible by each one of those roles:

Table 28 – User profiles considered for WiseGRID Cockpit

INFO	LOCAL (area level)		REGIONAL		GENERAL		
	Grid exploitation	Studies - constructions	Grid operation center	Studies - constructions	Simple	Operation management	Network users
1. Dashboard – Overall demand, production graph	X <i>(adapted to the area controlled)</i>	X <i>(historical data adapted to the area controlled)</i>	X <i>(adapted to the region controlled)</i>	X <i>(historical data adapted to the region controlled)</i>			X <i>(historical data aggregated per region and whole distribution network for prediction of energy demand purposes)</i>
1. Dashboard - Weather Report	X <i>(adapted to the area controlled)</i>		X <i>(adapted to the region controlled)</i>		X	X	X
1. Dashboard – Active incidents	X <i>(adapted to the area controlled)</i>		X <i>(adapted to the region controlled)</i>				
1. Dashboard – Active works							
1. Dashboard – Active demand response campaigns							X
2. Map - incidents	X <i>(adapted to the area controlled)</i>		X <i>(adapted to the region controlled)</i>				

2. Map – Topology view (with reconfiguration possibilities)	X <i>(for healing strategies and to manage faults situations - adapted to the area controlled)</i>	X <i>(for planning purposes, network studies and constructions on LV level - adapted to the area controlled)</i>	X <i>(for healing strategies and to manage faults situations - adapted to the region controlled)</i>	X <i>(for planning purposes, network studies and constructions on MV level - adapted to the region controlled)</i>			
2. Map – State view	X <i>(for assets connected to LV and MV level - adapted to the area controlled)</i>		X <i>(for assets connected to MV level - adapted to the region controlled)</i>				
3. Single line - diagram	X <i>(view - for elements connected to LV and MV level - adapted to the area controlled)</i>	X <i>(update and edit - for planning purposes, network studies and constructions on LV level - adapted to the area controlled)</i>	X <i>(view- for elements connected to MV level - adapted to the region controlled)</i>	X <i>(update and edit - for planning purposes, network studies and constructions on MV level - adapted to the region controlled)</i>			
4. AMR	X <i>(status info in order to handle fault occasions - adapted to the area controlled)</i>	X <i>(AMR registration in the system - adapted to the area controlled)</i>	X <i>(status info in order to handle fault occasions for MV and important for the country customers- - adapted to the region controlled)</i>		X <i>(AMR number)</i>		X <i>(metering data for each AMR and user data for load forecast, billing and power theft detection purposes)</i>
5. Grid maintenance – incidents	X <i>(for the incidents at LV & MV level -</i>		X <i>(for all the incidents of distribution network -</i>		X <i>(MAJOR incidents should be reported</i>	X <i>(a combination of them</i>	

	<i>adapted to the area controlled)</i>		<i>adapted to the region controlled)</i>		as soon as possible)	would be useful for them and in an aggregated form)	
6. Works	X		X				
7. Demand response campaigns	X – view all the information except for price		X – view all the information except for price				X
8. Self-healing strategies	X		X				

3.5.2 Sections

3.5.2.1 Dashboard

The dashboard present a chart with the last hour overall demand and production measured on the grid. A view is also included to indicate the corresponding demand and production forecasts for the current day.

Additionally, an overview of the future congestion forecasts as provided by the Congestion Forecast module, together with their details (time period, location and power excess) will be shown.

Finally, also the current weather report for the controlled area will be provided.

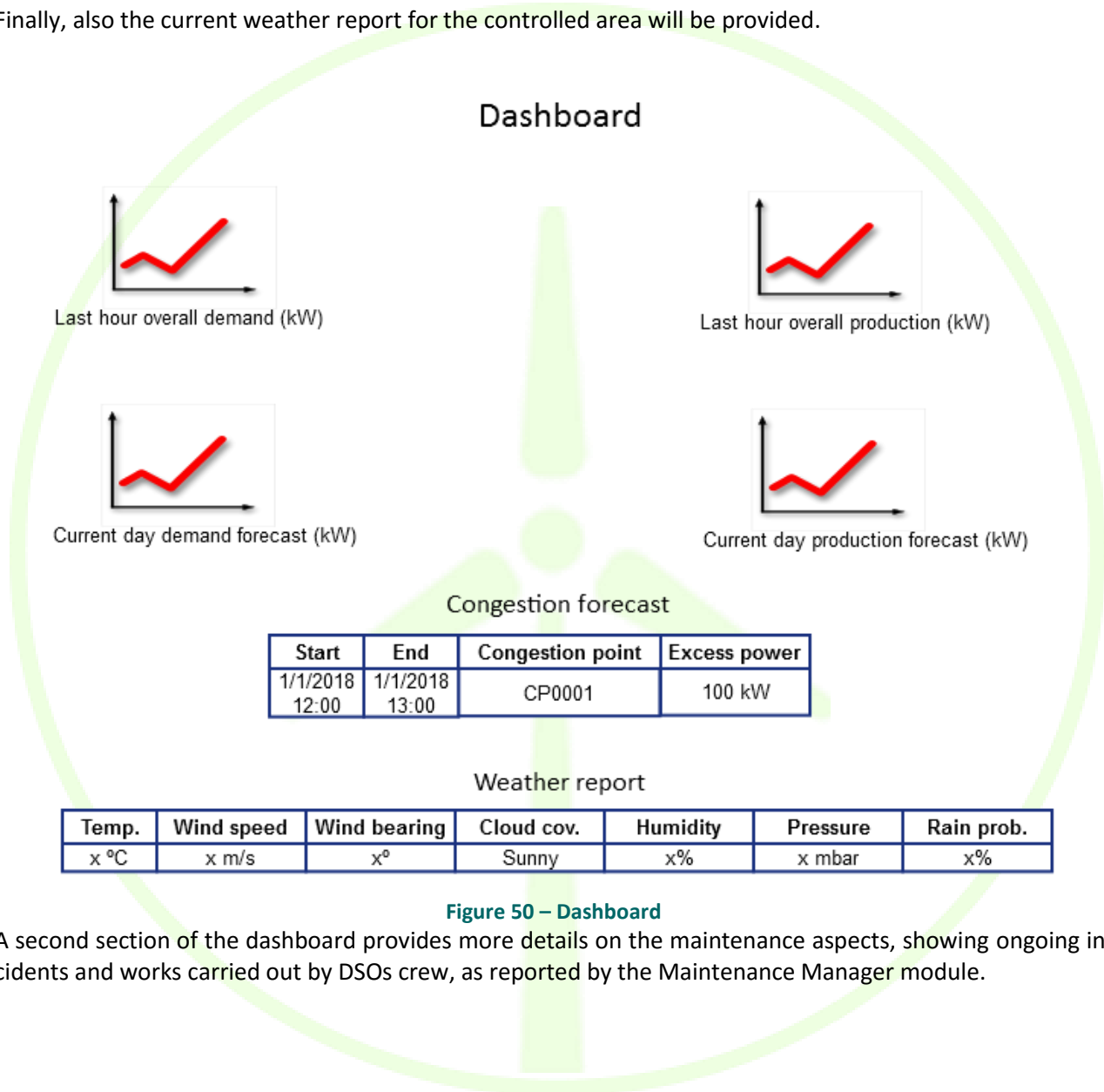


Figure 50 – Dashboard

A second section of the dashboard provides more details on the maintenance aspects, showing ongoing incidents and works carried out by DSOs crew, as reported by the Maintenance Manager module.

Dashboard (maintenance)

Active incidents

Tags	Type	Priority	State	Start date	Creator	Asset
outage, failure	Outage	High	Registered	1/1/2017 00:00	automatically identified	Substation 1001
failure, voltage	Electrical problem	Medium	Registered	1/1/2017 00:00	automatically identified	Substation 1002

Active works

ID	Type	Priority	State	Start date	Incident	Asset	Details
1	Corrective	High	Started	1/1/2017 06:00	#01234567	Substation 1001	
2	Corrective	Medium	Started	1/1/2017 06:00	#01234568	Substation 1002	

Figure 51 – Dashboard (maintenance)

3.5.2.2 Map

A map will show all the elements managed by the DSO (lines, substations, RES and usage point locations). Different visualizations are foreseen, which will change the colour legend:

- Status view: simple view highlighting elements with active incidents. The colour legend will indicate the different possible status: ok – green -, under maintenance – yellow -, with ongoing incident – red -, or unknown – grey. Hovering over the marker of an active incident will show a summary.

Status view - elements are colored accordingly to their state (green=ok, yellow=under maintenance, red=with incident, grey=unknown)
Clicking on elements navigates to the detailed view

Map - Status view

MV/LV Lines
Substations
Usage points
RES

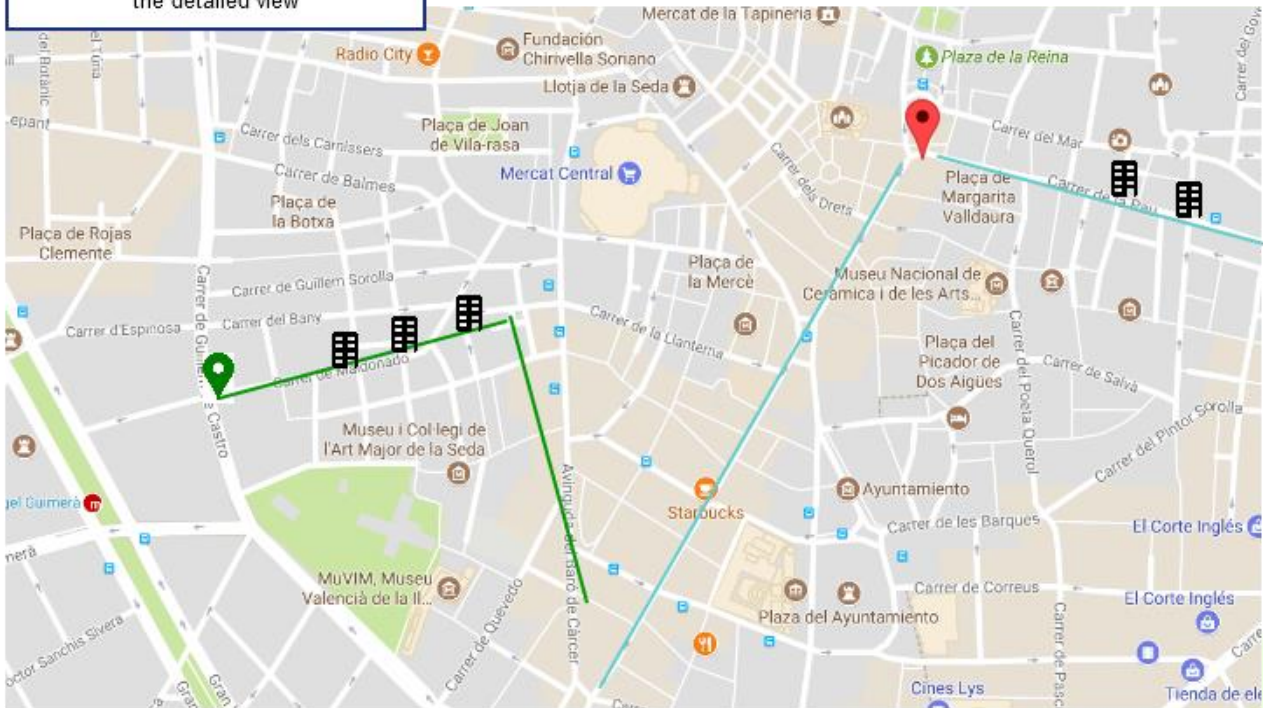


Figure 52 – Map (status view)

- Topology view: view where the colour code represents dependencies among the different elements (e.g. lines fed by the same substation get the same colour code)

Topology view - elements
related by colors
Clicking on elements navigates
to the detailed view

Map - Topology view

MV/LV Lines
Substations
Usage points
RES

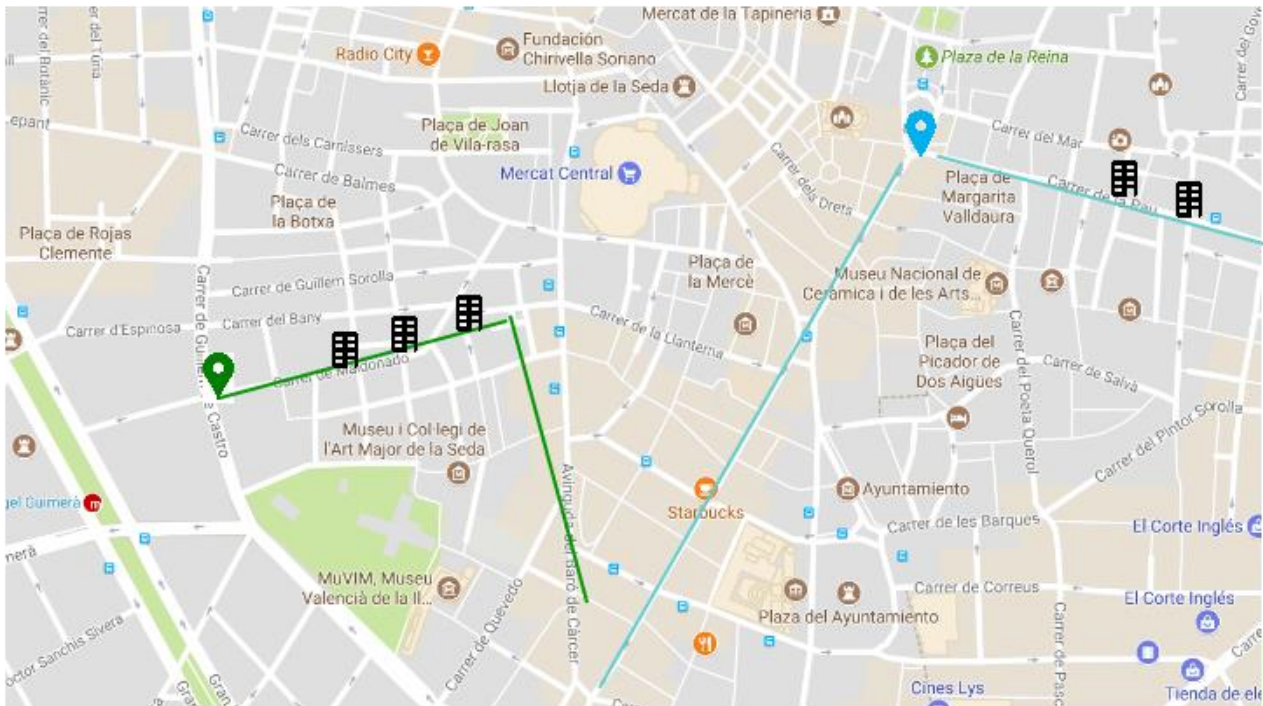


Figure 53 – Map (topology view)

3.5.2.3 Single-line diagram

The objective of this view is to provide a different representation of the status of the monitored elements over a single-line diagram representation of the distribution grid. The colour legend will indicate the different possible status: ok – green -, under maintenance – yellow -, with active incident – red -, or unknown – white. Hovering over the marker of an active incident will show a summary

Single line diagram. Color indicates state of element (green=ok, yellow= under maintenance, white=unknown, red=incident)

Single-line diagram

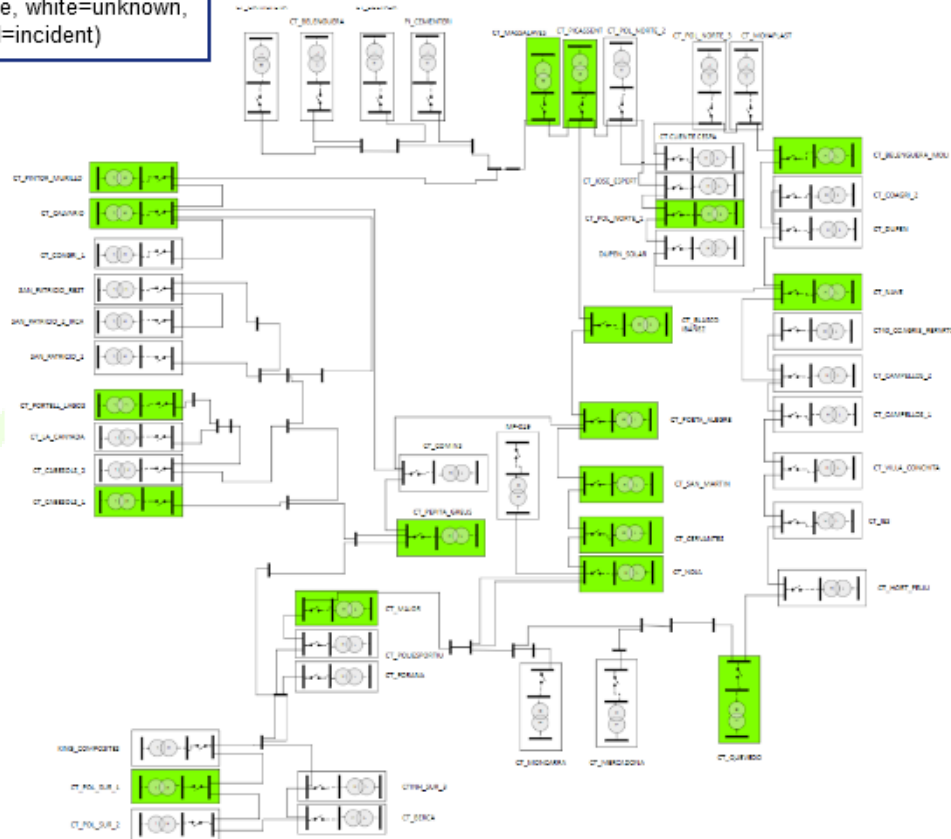


Figure 54 – Single-line diagram

3.5.2.4 AMR

This section is a compendium of interfaces offering an overview of the status of the different data sources managed by the WiseGRID Cockpit that provide Meter Readings: Unbundled Smart Meters, AMR system and the data being collected from SCADA. It will include 3 subsections:

- General information, displaying static information associated to each meter or SCADA sensor

AMR

Search meter...

Static info. per smart meter

General info

Associated
element

Measures

Current status OK

Asset info.

Model	SM Model v1.0
Phase count	3
Rated current	xA
Rated voltage	xV
...	...

Figure 55 – AMR (general information)

- Associated element, displaying information of the asset being measured by the corresponding sensor (substation bus or usage point). From this view, the measurements can be linked to a particular asset if they were not linked yet

AMR

Search meter...

Smart meters may be associated to substations or usage points.
This view provides summary information of the corresponding element

General info

Associated element

Measures


Substation " "

Current status OK

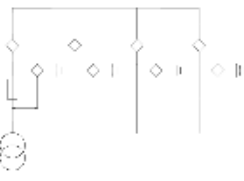
General information

Name	SM Model v1.0
Code	3
Net type	LV
Connection code	S
Description	---
Address	---

Map



Schematic



Topology

- Substations (1)
 - 1065
 - ACI lineSegments (4)
 - BusbarSections (3)
 - 1065
 - BaseVoltages (1)
 - IdentifiedObject.aliasName: 1065
 - Terminals (1)
 - VoltageLimitSets (1)
 - 1065
 - 1065
 - IdentifiedObject.aliasName: 1065
 - Power Transformers (1)
 - Substations (2)
 - Switches (1)
 - VoltageLevels (2)

Figure 56 – AMR (associated elements)

- Measures, providing access to the history of raw measures of the selected element

AMR

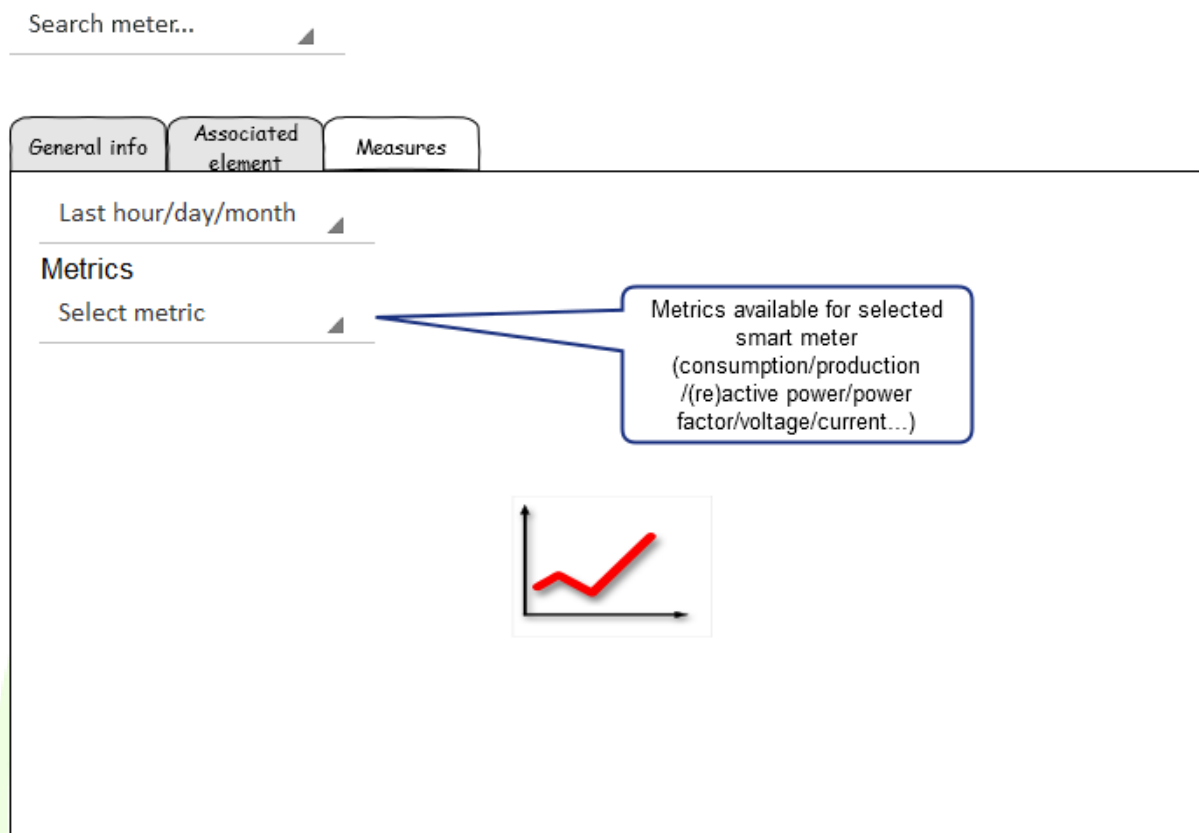


Figure 57 – AMR (measures)

3.5.2.5 Grid maintenance

This section contains several dialogs related with the management of the grid maintenance. Its main purpose is to provide the capability to create new activities and incidents in the Maintenance Manager module from the main user interface of the WiseGRID Cockpit, and provide an overview of the active incident, maintenance tasks and works of the personnel.

3.5.2.5.1 Phone calls

This dialog facilitates the registration of the phone calls received at DSOs customer desk, allowing those to record new incidents accordingly to the information received.

Grid maintenance - phone calls

Date	Phone number	In/Out	Asset	Incident	Priority	Description	Caller	Details	Answer
1/1/2020 00:00	+34xxxxxxxx	In	Usage point xxxx	6354878xxxxxx	High	Outage at home	Name	...	Fixed
1/1/2020 00:00	+34xxxxxxxx	In	Substation 10001	6354878xxxxxx	High	Routine maintenance	Op 1	...	
...

Register phone calls that will be afterwards associated to incidents

Figure 58 – Grid maintenance (phone calls)

Grid maintenance - phone calls

New phone call

In/Out ▼

Phone number

Caller

Asset

Date

Priority ▼

Description

Details

Answer

Save call

Register phone calls that will be afterwards associated to incidents



Figure 59 – Grid maintenance (phone call registration)

3.5.2.5.2 Incidents

This section provides a detailed overview of active and history of incidents, and a summary of related information as provided by the Maintenance Management module (related asset, related phone calls, related works, personnel assigned to those works...)

Grid maintenance - incidents

Tags	Type	Priority	State	Start date	Est. end	End date	Creator	Asset	Details
outage, maintenance	Outage (MAINTENANCE)	High	Registered	1/1/2017 00:00	1/1/2017 00:00	1/1/2017 00:00	automatically identified	Substation 1001	Substation 1001
failure, voltage	Electrical problem	Medium	Registered	1/1/2017 00:00	1/1/2017 00:00	1/1/2017 00:00	automatically identified	Substation 1002	Substation 1002

Clicking on an incident shows details

Figure 60 – Grid Maintenance (incidents)

Grid maintenance - incident details

Incident #645654xxxxx

General info

Calls

Works

Type

Priority

State

Creator

Start

End

Tags

Asset

Details




Figure 61 – Grid maintenance (incident details, general info.)

Grid maintenance - incident details

Incident #645654xxxxx

Detail of calls
associated to this
incident

General info								
Calls								
Works								
Date	Phone number	In/Out	Incident	Priority	Description	Caller	Details	Answer
1/1/2020 00:00	+34xxxxxxxxxx	In	6354878xxxxxx	High	Outage at home	Name	...	Fixed

Figure 62 – Grid maintenance (incident details, calls)

Grid maintenance - incident details

Incident #645654xxxxx

Detail of works
associated to this
incident

General info
Calls
Works

ID	Type	Priority	State	Started	Finished	Details
C 2017-001	Corrective	High	Initiated	01/11/2017 12:00		...

Figure 63 – Grid maintenance (incident details, works)

From this section, it is also possible to register new incidents in the Maintenance Management module.

Grid maintenance - incidents

[Register new incidents](#)

New incident

Type	<input type="text" value="Outage"/>	Priority	<input type="text" value="High"/>
State	<input type="text" value="Registered"/>	Creator	<input type="text" value="text"/>
Date	<input type="text" value="dd/mm/yyyy hh:mm"/>		
Tags	<input type="text" value="text"/>		
Asset	<input type="text" value="Substation 1001"/>		

Details


[Save incident](#)

Figure 64 – Grid maintenance (new incident registration)

3.5.2.5.3 Works

This section gives an overview of the different scheduled works (preventive and corrective maintenance), current status and the personnel assigned to those works. Works are assigned to incidents (i.e. one incident may require several different works to be solved). This information is, in fact, a summary of the information handled by the Maintenance Management module. Maintenance Management module is in charge of handling priorities of works and assignment of those to the corresponding personnel.

Works

Overview of the works, managed by the Maintenance Service.
Clicking on a row shows details

ID	Type	Priority	State	Started	Est. finish	Finished	Incident	Asset	Details
C 2017-001	Corrective	High	Initiated	01/11/2017 12:00	01/11/2017 15:00		#645654xx	Substation 1001	...
C 2017-002	Preventive	Low	Initiated	01/11/2017 12:00	01/11/2017 15:00			Substation 1002	Regular review

Figure 65 – Summary of works

Clicking on each one of the works will show a new screen with details:

Works

Work C 2017-001

Details of work, managed by Maintenance Service

Type	Priority	State	Started	Finished	Incident	Asset	Details
Corrective	High	Initiated	01/11/2017 12:00		#645654xx	Substation 1001	...

Assigned worker

ID	Name
T0001	...

Assign to different worker



History

	Date	Issuer	Details	Message
1	01/11/2017 12:00	T0001	Work initiated	
1	01/11/2017 13:00	T0001	Work postponed	Need to change asset x

Figure 66 – Work details

3.5.2.5.4 Field personnel

This section provides a summary of the current workload of the field personnel, showing a summary of the works assigned to each member of the team. This information is provided by the Maintenance Management module.

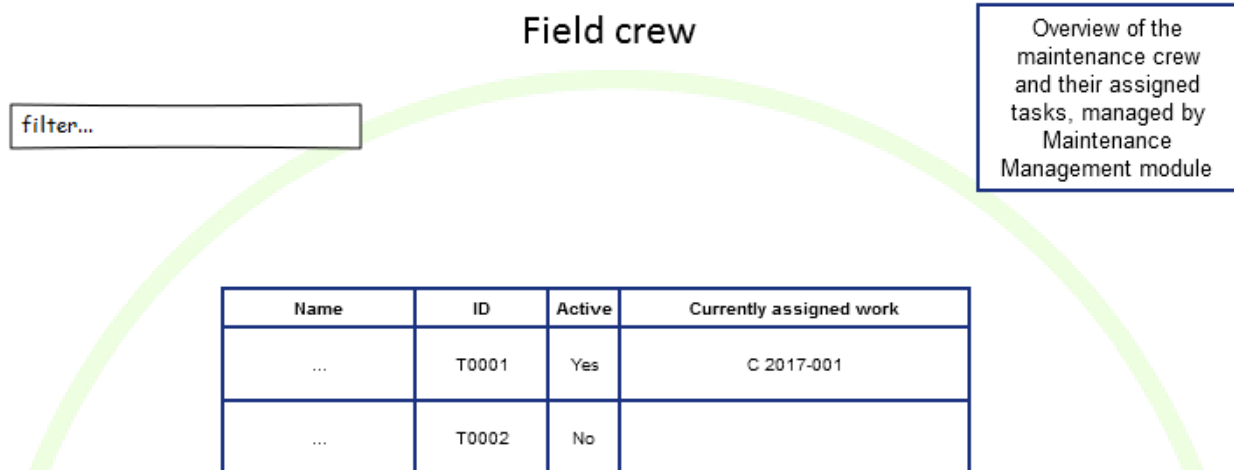


Figure 67 – Overview of field crew

3.5.2.6 Demand response campaigns

This section gives the DSO operator an overview of the demand response campaigns that have been initiated by the Failure Treatment module. Each demand response campaign is defined by a time period (PTU, usually 15 minutes), a congestion point (the location in the grid where the congestion was forecasted), the requested flexibility for that period, and its status (*request* sent by DSO, *offers* received from aggregators, *orders* set by DSO, on-going and finalized)

Demand response campaigns

Forecasted congestions

Point	Start	End	Flexibility req	Status
Substation 10001 - Bus 1	1/1/2017 18:00	1/1/2017 18:30	30 kW	Offers received
Substation 10002 - Bus 1	1/1/2017 18:00	1/1/2017 18:30	30 kW	Offers accepted
Substation 10003 - Bus 1	1/1/2017 18:00	1/1/2017 18:30	30 kW	-

Status: request sent / offers received / offers accepted

Finished campaigns

Point	Start	End	Flexibility req	Price	Status	Actual excess
Substation 10001 - Bus 1	1/1/2017 12:00	1/1/2017 12:30	30 kW	x €	Finalized	0 kW
Substation 10002 - Bus 1	1/1/2017 12:00	1/1/2017 12:30	30 kW		No offers received	0 kW

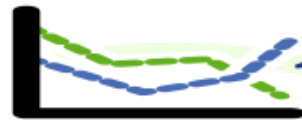
Clicking on the items provides details on the campaign

Status: finalized / no offers received / no offers accepted

Figure 68 – Overview of Demand response campaigns

Demand response campaign #DR654897xxxx

Finalized	Final price x €	Provided flex. 30 kW
-----------	--------------------	-------------------------



Forecasted vs Actual power

Visual insight of effect of DR campaign

History

	Status	Party	Date	Flexibility	Duration (PTUs)	Price
1	Request sent	DSO	1/1/2017 06:00	30 kW	2	
2	Offer received	EV Fleet operator	1/1/2017 06:01	30 kW	2	250€
2	Offer received	VPP	1/1/2017 06:02	10 kW	2	300€
3	Offer accepted	EV Fleet operator	1/1/2017 06:05	30 kW	2	250€
4	Started		1/1/2017 18:00	30 kW	2	
5	Finalized		1/1/2017 18:30	30 kW	2	

Figure 69 – Demand Response campaign details

3.5.2.7 Self-healing

This dialog shows the list of incidents that have been addressed by the FLISR module (described in section 3.4.4.2). For each of those, the details of the steps automatically performed are shown. In those cases where no automatic control is available, those steps are provided so the DSO maintenance crew can manually perform them.

The dialog will be divided in two sections:

- Upper half of the screen will display a list of events (filtering is allowed). Selecting any one opens the lower half of the screen.
- Lower half of the screen will display the steps to be performed for the selected event. Clicking on “Operate” performs the step if system is in manual mode (in automatic mode steps are performed sequentially without human intervention). When the step involves a grid asset which cannot be operated remotely, clicking generates a work order to the Maintenance Management module. Status gives feedback on the proper execution of the command. Clicking on any of the display check boxes displays in a separate screen the map or the single line diagram after the current step, superimposed to the actual state of the grid.

SELF-HEALING STRATEGIES

SELECT THE HEALING OPERATION MODE:
☒ **AUTOMATIC**
☐ **MANUAL**

DETECTED EVENTS

View Steps	id Fault	Tags	Priority	State	Start date	End date	Creator	Asset	Details
<input type="checkbox"/>	#0009	Low Voltage	Medium	Resolved	24/12/16 08:34	----	Automatically identified	Substation 1012	---
<input type="checkbox"/>	#0004	Outage Failure	High	Unresolved	04/10/16 07:35	----	Automatically identified	Substation 1104	---
<input type="checkbox"/>	#0003	High Frequency	High	Resolved	12/12/16 14:35	----	Automatically identified	Substation 1405	---
<input checked="" type="checkbox"/>	#0012	Outage Failure	High	Unresolved	1/1/17 04:35	----	Automatically identified	Substation 1001	---
<input type="checkbox"/>	#0010	Low voltage	Low	Registered	1/1/17 11:35	----	Automatically identified	Substation 1001	---

STEPS (TO BE) PERFORMED [FAULT id = #0012]

STEPS	DESCRIPTION	ASSET OPERATION	OPERATE	STATUS	DISPLAY MAP	DISPLAY SINGLE LINE DIAGRAM
01:	Disconnect asset 1005	Remote	✓	Done	<input type="checkbox"/>	<input type="checkbox"/>
02:	Disconnect asset 1014	Remote	✓	Done	<input type="checkbox"/>	<input checked="" type="checkbox"/>
03:	Connect asset 1056	Local		Waiting	<input type="checkbox"/>	<input type="checkbox"/>

Single-line diagram. Color indicates state of element: ok, white=unknown, red=incident

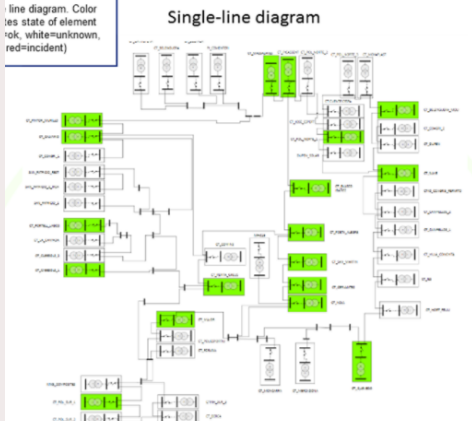


Figure 70 – Self-healing strategies

4 LINK WITH OTHER WISEGRID APPLICATIONS

WG Cockpit is the tool dealing with the activities of the DSO and is in charge of assuring the quality of the electricity grid within the WiseGRID project. For this reason, WG Cockpit is the tool responsible for triggering DR campaigns and transmit its requirements to the proper WiseGRID tools. The tools which are linked to WG Cockpit in this DR framework are: WiseCOOP, WG STaaS/VPP and WiseEVP.

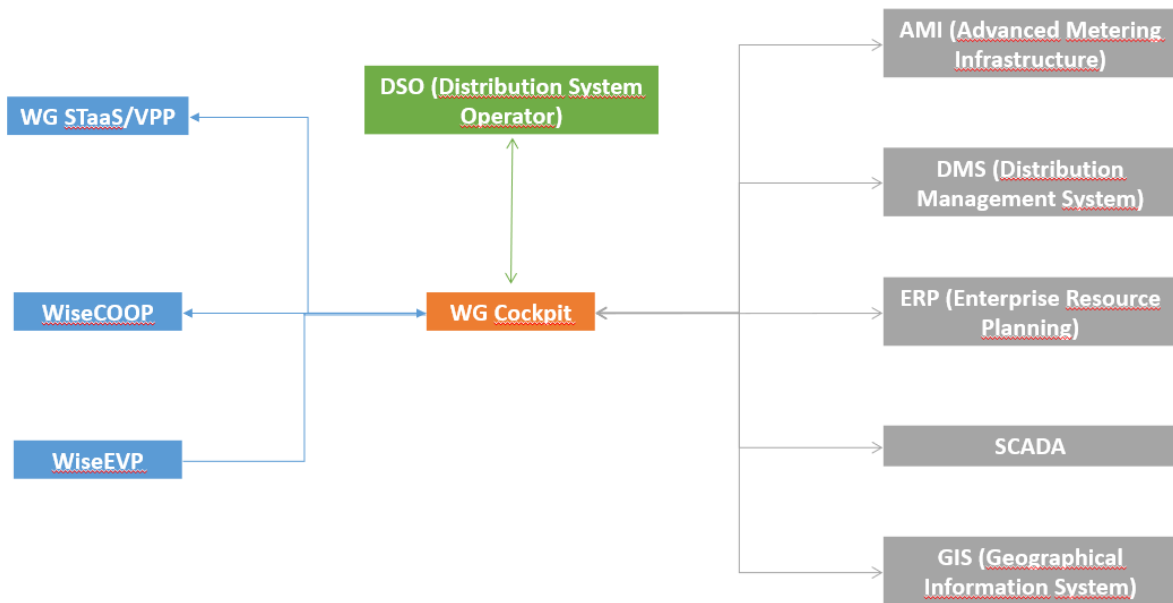


Figure 71 – WG Cockpit interactions

When WG Cockpit considers that it is necessary to start a DR campaign in order to solve a problem in the Distribution grid, WG Cockpit sends a request to the tools that can provide flexibility (the ones above mentioned) and waits for their offers. The offers contain the amount of flexibility that each tool can provide to WG Cockpit and the prize of this flexibility. Then, WG Cockpit chooses which is the most efficient offer and transmit its decision to the involved tools (as mentioned before, these are WG STaaS/VPP, WiseCOOP and WiseEVP).

WG Cockpit's interactions with WiseCOOP, WiseEVP and WG STaaS/VPP are of course performed through WG IOP.

5 DPIA CONSIDERATIONS

WG Cockpit is a specific tool within the Project and therefore it becomes subject of an assessment to evaluate risks level in relation to data privacy conformity. The evaluation (assessment was performed under centralised procedure within D3.1 and was recently updated within deliverable D3.2. Relevant threats and events were assessed and conclusions are that no significant risks could affect WG Cockpit tool from the perspective of personal data protection. However, there are general and tool specific recommendations resulting from the updated DPIA that are considered within the design phase of the Project.

The synthesis of threats and events identified within the assessment for WG Cockpit is in the following table:

Table 29 – Threat and feared events identification for WG Cockpit

Feared events	Threat ID	Threat name	Brief explanation why relevant
Breach of use of personal data	NLPDP	Non legally based personal data processing	Processing of personal data is not based on consent, a contract, legal obligation, or other relevant legal ground

Feared events	Threat ID	Threat name	Brief explanation why relevant
Change in processing: it deviates from what was originally planned (diversion of the purpose, excessive or unfair collection)	II	Incomplete information	The information provided to the data subject on the purpose and use of data is not complete
Disappearance of personal data: they are not or no longer available	ED	Eavesdropping of computer channels	Interception of Ethernet traffic; acquisition of data sent over a Wi-Fi network, etc.
	Pobj	Prevention of objections	Data subjects have the right to object to the processing of data. If they want to execute this right it must be (technically) possible
Diverting of personal data to other users: they are	DoS	Denial of service	Denial of service will lead to unavailability of computing system
Unavailability of legal processes: they do not or no longer exist or work	SA	Software alteration	Errors during updates, configuration or maintenance; infection by malicious code; replacement of components, etc.

Even within application “personal data” (like metering data) is mentioned, so we considered it in DPIA. However, Metering data is not connected to individual persons, so their risks associated likelihood is very low. There is no need for specific additional controls as long as the risk is always below acceptable level.

However, due to complete integration of WG Cockpit within WiseGRID, the general controls should be applied for safer approach:

1. Take written consent from all customers involved in the usage of designated tools (the form of the consent will be developed in due time, before launching the implementation on pilot sites).
2. Data will be anonymized as soon as affordable within the process, with no influence in the functionality.
3. WiseGRID Cockpit does not consider a “personal data breach reaction plan” and “reporting protocol”. This should be reevaluated. In any case, the tool shall consider “Reducing hardware vulnerabilities” as a control.

6 CONCLUSIONS

In this deliverable, the work performed in the framework of task T13.1 has been presented.

Task 13.1 *WiseGRID Cockpit Design* focuses in the analysis of the different requirements and use cases defined by the WP2 that need to be addressed by the WiseGRID Cockpit application, the identification of the necessary modules to fulfil those and the specification of the functions to be implemented within each one of these modules. In addition, interfaces with the target users of the applications have been designed following several iterations while keeping potential end users within the consortium involved in the process.

This deliverable additionally presents or references the formal specification of the APIs to be implemented by each one of the modules. This will allow the parallel development of the different modules that are responsibility of different teams within the consortium during the implementation phase, and will serve as a basis for future module independent unitary testing and first assessment of the integration task to be performed as part of the development phase.

Upcoming work in task T13.2 includes the development of the different modules described in this deliverable. These modules will be developed independently and will implement the documented APIs described in this

report. Task T13.2 will finalise with the integration of the different modules that compose the complete WiseGRID Cockpit application.

Finally, under task T13.3 different functional tests over the applications – once all independent modules have been successfully integrated together – will be carried out in a lab-testing environment, thus ensuring the correct operation of the WiseGRID Cockpit application under controlled environment before proceeding with the deployment of the application in the different pilot sites. Implementation and lab-testing will be finalised in month M21 and reported accordingly in the follow-up deliverable D13.2 *WiseGRID Cockpit implementation and lab-testing*.

7 REFERENCES AND ACRONYMS

7.1 REFERENCES

- [1] "<https://hackernoon.com/microservices-are-hard-an-invaluable-guide-to-microservices-2d06bd7bcf5d>," [Online].
- [2] "<http://www.rabbitmq.com/resources/google-tech-talk-final/alexis-google-rabbitmq-talk.pdf>," [Online].
- [3] "<https://www.rabbitmq.com/>," [Online].
- [4] "D4.2 WiseGRID interoperable Integrated Process".
- [5] "<http://nobelgrid.eu/d4-1/>," [Online].
- [6] "https://github.com/SMXCore/SMXCore_NG," [Online].
- [7] "<http://www.sitel-sa.com/>," [Online].
- [8] "D4.2 WiseGRID interoperable Integrated Process (IOP)".
- [9] "D2.1 WiseGRID requirements, Use cases and pilot sites analysis".
- [10] "<https://postgis.net/>," [Online].
- [11] "<https://www.postgresql.org/>," [Online].
- [12] "D4.2 WiseGRID interoperable Integrated Process (WG IOP)".
- [13] "<https://github.com/meteor/meteor/blob/devel/packages/ddp/DDP.md>," [Online].
- [14] "D10.2 WiseGRID Flexibility-based DR Optimization Framework Specifications".
- [15] "<https://www.usef.energy/>," [Online].
- [16] "D4.2 WiseGRID interoperable Integrated Process (WG IOP)".

7.2 ACRONYMS

Table 30 – Acronyms list

Acronyms List	
AMI	Advanced Metering Infrastructure
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
CIM	Common Information Module
DB	Data Base
DER	Distributed Energy Resource
DMS	Distribution Management System
DPIA	Data Protection Impact Assessment
DR	Demand Response
DSO	Distribution System Operator
ESB	Enterprise Service Bus
EV	Electric Vehicle
FLISR	Fault Location Isolation and Service Restoration
FTP	File Transfer Protocol
GIS	Geographic Information System
HL-UC	High Level Use Case
HTTP	Hypertext Transfer Protocol
ID	Identification
IMF	Intermediate Format
IoT	Internet of Things
KPI	Key Performance Indicator
LV	Low Voltage
MV	Medium Voltage
PLC	Power Line Communication
PV	Photovoltaic
RES	Renewable Energy Source
RPC	Remote Procedure Calls
RT	Real-Time
SCADA	Supervisory Control And Data Acquisition
SMM	Smart Metrology Meter
SMX	Smart Meter eXtension
TSO	Transmission System Operator
USM	Unbundled Smart Meter

V2G	Vehicle to Grid
VPP	Virtual Power Plant
WG	WiseGRID



8 ANNEXES

8.1 SWAGGER SPECIFICATIONS OF THE MAINTENANCE MANAGER

```
{
  "swagger": "2.0",
  "info": {
    "description": "Maintenance Manager",
    "version": "1.2",
    "title": "maintenance_manager"
  },
  "tags": [{
    "name": "incident",
    "description": "Methods related to incidents"
  },
  {
    "name": "crew",
    "description": "Methods related to crew"
  },
  {
    "name": "device",
    "description": "Methods related to devices"
  }
  ],
  "paths": {
    "/api/incident": {
      "get": {
        "tags": ["incident"],
        "summary": "Get list of incidents",
        "produces": ["application/json"],
        "responses": {
          "200": {
            "description": "List of incidents (with details)",
            "schema": {
              "type": "array",
              "items": {
                "$ref": "#/definitions/Incident"
              }
            }
          }
        }
      },
      "post": {
        "tags": ["incident"],
        "summary": "Create new incident",
        "produces": ["application/json"],
        "parameters": [{
          "in": "body",
          "name": "body",
          "description": "New incident details",

```

```

        "required": true,
        "schema": {
            "$ref": "#/definitions/NewIncident"
        }
    }],
    "responses": {
        "200": {
            "description": "Details of created incident",
            "schema": {
                "$ref": "#/definitions/Incident"
            }
        }
    }
},
"/api/incident/{id}": {
    "get": {
        "tags": ["incident"],
        "summary": "Get incident data",
        "produces": ["application/json"],
        "parameters": [{
            "in": "path",
            "name": "id",
            "description": "Incident id",
            "required": true,
            "type": "string"
        }],
        "responses": {
            "200": {
                "description": "Incident details",
                "schema": {
                    "$ref": "#/definitions/Incident"
                }
            }
        }
    }
},
"/api/incident/{id}/tasks": {
    "get": {
        "tags": ["incident"],
        "summary": "Get incident tasks",
        "produces": ["application/json"],
        "parameters": [{
            "in": "path",
            "name": "id",
            "description": "Incident id",
            "required": true,
            "type": "string"

```

```

    }],
    "responses": {
      "200": {
        "description": "List of related tasks",
        "schema": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/Task"
          }
        }
      }
    }
  },
  "/api/device": {
    "get": {
      "tags": ["device"],
      "summary": "Get list of devices",
      "produces": ["application/json"],
      "parameters": [{
        "in": "query",
        "name": "type",
        "description": "Device type (for filtering results)",
        "required": false,
        "type": "string"
      },
      {
        "in": "query",
        "name": "id",
        "description": "Device id (for filtering results)",
        "required": false,
        "type": "string"
      }
    ],
    "responses": {
      "200": {
        "description": "List of devices",
        "schema": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/Device"
          }
        }
      }
    }
  },
  "post": {
    "tags": ["device"],
    "summary": "Register a new device",

```



```

"produces": ["application/json"],
"parameters": [{
  "in": "body",
  "name": "body",
  "description": "New device details",
  "required": true,
  "schema": {
    "$ref": "#/definitions/NewDevice"
  }
}],
"responses": {
  "200": {
    "description": "Inserted device details",
    "schema": {
      "$ref": "#/definitions/Device"
    }
  }
},
"/api/device/{type}": {
  "get": {
    "tags": ["device"],
    "summary": "Get list devices for a given type",
    "produces": ["application/json"],
    "parameters": [{
      "in": "path",
      "name": "type",
      "description": "Device type",
      "required": true,
      "type": "string"
    }],
    "responses": {
      "200": {
        "description": "List of devices of a given type",
        "schema": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/Device"
          }
        }
      }
    }
  }
},
"/api/device/{type}/{id}": {
  "get": {
    "tags": ["device"],

```

```

"summary": "Get device details",
"produces": ["application/json"],
"parameters": [{
  "in": "path",
  "name": "type",
  "description": "Device type",
  "required": true,
  "type": "string"
},{
  "in": "path",
  "name": "id",
  "description": "Device id",
  "required": true,
  "type": "string"
}],
"responses": {
  "200": {
    "description": "List of devices",
    "schema": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/Device"
      }
    }
  }
}
},
"/api/devicetypes": {
  "get": {
    "tags": ["device"],
    "summary": "Get list of defined device types",
    "produces": ["application/json"],
    "responses": {
      "200": {
        "description": "List of defined device types",
        "schema": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/DeviceType"
          }
        }
      }
    }
  }
}
},
"/api/crew": {
  "get": {

```

```

"tags": ["crew"],
"summary": "Get list of workers",
"produces": ["application/json"],
"responses": {
  "200": {
    "description": "List of workers",
    "schema": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/Worker"
      }
    }
  }
},
"/api/crew/{id}": {
  "get": {
    "tags": ["crew"],
    "summary": "Get details of a worker",
    "produces": ["application/json"],
    "parameters": [{
      "in": "path",
      "name": "id",
      "description": "Worker id",
      "required": true,
      "type": "string"
    }],
    "responses": {
      "200": {
        "description": "Worker details",
        "schema": {
          "$ref": "#/definitions/WorkerDetails"
        }
      }
    }
  }
},
"/api/crew/{id}/incidents": {
  "get": {
    "tags": ["crew"],
    "summary": "Get list of incidents assigned to a worker",
    "produces": ["application/json"],
    "parameters": [{
      "in": "path",
      "name": "id",
      "description": "Worker id",
      "required": true,

```

```

    "type": "string"
  }],
  "responses": {
    "200": {
      "description": "List of UIDs of assigned incidents",
      "schema": {
        "type": "array",
        "items": {
          "type": "string"
        }
      }
    }
  }
},
"/api/crew/{id}/tasks": {
  "get": {
    "tags": ["crew"],
    "summary": "Get list of tasks posted by a worker",
    "produces": ["application/json"],
    "parameters": [{
      "in": "path",
      "name": "id",
      "description": "Worker id",
      "required": true,
      "type": "string"
    }],
    "responses": {
      "200": {
        "description": "Worker details",
        "schema": {
          "$ref": "#/definitions/WorkerTask"
        }
      }
    }
  }
},
"/api/notification/{deviceType}": {
  "get": {
    "tags": ["incident"],
    "summary": "Get list of possible incident notifications for a device type",
    "produces": ["application/json"],
    "parameters": [{
      "in": "path",
      "name": "deviceType",
      "description": "Device type",
      "required": true,
      "type": "string"
    }],

```

```

    }],
    "responses": {
      "200": {
        "description": "List of possible notifications",
        "schema": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/Notification"
          }
        }
      }
    }
  },
  "definitions": {
    "Incident": {
      "type": "object",
      "properties": {
        "mRID": {
          "type": "string",
          "description": "UID of the incident"
        },
        "workId": {
          "type": "integer",
          "description": "Numeric ID"
        },
        "typeId": {
          "type": "integer",
          "description": "0=corrective, 1=preventive"
        },
        "type": {
          "type": "string",
          "description": "Type description"
        },
        "priority": {
          "type": "integer",
          "description": "Priority 0-100"
        },
        "stateId": {
          "type": "string",
          "description": "State code R=registered, N=notified, F=finished"
        },
        "state": {
          "type": "string",
          "description": "State description"
        },
        "startDate": {

```

```

    "type": "string",
    "description": "Start data"
  },
  "endDate": {
    "type": "string",
    "description": "End date"
  },
  "workerId": {
    "type": "string",
    "description": "Assigned worker id"
  },
  "workerName": {
    "type": "string",
    "description": "Assigned worker name"
  },
  "notificationDate": {
    "type": "string",
    "description": "Notification date"
  },
  "updateData": {
    "type": "string",
    "description": "Last update date"
  },
  "communicationDate": {
    "type": "string",
    "description": "Communication date TODO"
  },
  "deadlineDate": {
    "type": "string",
    "description": "Deadline"
  },
  "creatorId": {
    "type": "string",
    "description": "ID of the notifier of the incident"
  },
  "creator": {
    "type": "string",
    "description": "Description of the notifier of the incident (e.g. system, customer...)"
  },
  "assetId": {
    "type": "string",
    "description": "ID of asset"
  },
  "asset": {
    "type": "string",
    "description": "Name of asset"
  },
  "assetType": {

```

```

        "type": "string",
        "description": "Type of asset"
    },
    "incident": {
        "type": "string",
        "description": "Description of the incident"
    },
    "details": {
        "type": "string",
        "description": "Free text with further details"
    }
  },
  "NewIncident": {
    "type": "object",
    "properties": {
      "deviceType": {
        "type": "string",
        "description": "Device type code"
      },
      "deviceId": {
        "type": "integer",
        "description": "Device ID"
      },
      "typeId": {
        "type": "integer",
        "description": "0=corrective, 1=preventive"
      },
      "notification": {
        "type": "string",
        "description": "Notification code"
      }
    }
  },
  "Task": {
    "type": "object",
    "properties": {
      "sortNum": {
        "type": "integer",
        "description": "Order of the task"
      },
      "date": {
        "type": "string",
        "description": "Date of task registration"
      },
      "status": {
        "type": "string",
        "description": "Status code of the incident after task N=notified R=registered F=finished"
      }
    }
  }
}

```

```

    },
    "worker": {
      "type": "string",
      "description": "Name of the worker that performed the task"
    },
    "details": {
      "type": "string",
      "description": "Data associated to the message field"
    },
    "message": {
      "type": "string",
      "description": "Free text describing the task, including placeholders for details"
    }
  }
},
"Worker": {
  "type": "object",
  "properties": {
    "mRID": {
      "type": "string",
      "description": "UID of the worker"
    },
    "name": {
      "type": "string",
      "description": "Name of the worker"
    },
    "idCard": {
      "type": "string",
      "description": "ID of the worker (company code)"
    },
    "contractStartDate": {
      "type": "string",
      "description": "Initial date of contract"
    },
    "contractEndDate": {
      "type": "string",
      "description": "Final date of contract"
    },
    "work": {
      "type": "string",
      "description": "UID of the currently assigned incident (first one of the list)"
    }
  }
},
"WorkerDetails": {
  "allOf": [
    "$ref": "#/definitions/Worker",
    {

```



```

    "type": "object",
    "properties": {
      "mainAddress": {
        "type": "object",
        "properties": {
          "streetDetail": {
            "type": "object",
            "properties": {
              "name": {
                "type" : "string",
                "description": "Main address street"
              }
            }
          },
          "townDetail": {
            "type": "object",
            "properties": {
              "name": {
                "type" : "string",
                "description": "Main address town"
              },
              "code": {
                "type" : "string",
                "description": "Main address ZIP code"
              },
              "country": {
                "type" : "string",
                "description": "Main address country"
              }
            }
          }
        }
      },
      "WorkerTask": {
        "allOf": [
          {
            "type": "object",
            "properties": {
              "mRID": {
                "type" : "string",
                "description": "UID of the associated incident"
              }
            }
          }
        ]
      }
    }
  },

```

```

    "$ref": "#/definitions/Task"
  ]
},
"NewDevice": {
  "type": "object",
  "properties": {
    "type": {
      "type": "string",
      "description": "Type of device"
    },
    "description": {
      "type": "string",
      "description": "Description of device"
    },
    "latitude": {
      "type": "number",
      "description": "Location of device (latitude)"
    },
    "longitude": {
      "type": "number",
      "description": "Location of device (longitude)"
    }
  }
},
"Device": {
  "type": "object",
  "properties": {
    "type": {
      "type": "string",
      "description": "Type of device"
    },
    "id": {
      "type": "number",
      "description": "Id of device"
    },
    "description": {
      "type": "string",
      "description": "Description of device"
    },
    "latitude": {
      "type": "number",
      "description": "Location of device (latitude)"
    },
    "longitude": {
      "type": "number",
      "description": "Location of device (longitude)"
    }
  }
}

```

```

    },
    "DeviceType": {
      "type": "object",
      "properties": {
        "type": {
          "type": "string",
          "description": "Type of device"
        },
        "description": {
          "type": "string",
          "description": "Description of device"
        }
      }
    },
    "Notification": {
      "type": "object",
      "properties": {
        "notification": {
          "type": "string",
          "description": "Notification code"
        },
        "typeID": {
          "type": "integer",
          "description": "0=corrective, 1=preventive"
        },
        "description": {
          "type": "string",
          "description": "Description of the notification"
        }
      }
    }
  }
}

```