

Title:	Document Version:
D7.1 WiseCOOP and WiseCORP Apps 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	WP7
Álvaro Nofuentes	ETRA	WP7

Authors (organization):
Álvaro Nofuentes (ETRA), Alberto Zambrano (ETRA), Germán Martínez (ETRA), Julio César Díaz (ITE), Amparo Mocholí (ITE), Antonis Papanikolaou (HYP), Benjamin Kraft (VS), Leandro Lombardo (ENG), Alexandre Lapedra (BYES), Catalin Chimirel (CRE), Vincent Dierickx (ECO-EID), Ine Swennen (ECO-EID), Foivos Palaogiannis (ICCS)

Abstract:
This document reports the work performed within Task 7.1 “WiseCOOP and WiseCORP Apps Design” in terms of functionalities offered and design of the solution for cooperatives and facility managers respectively. This framework takes as starting point the work done in the requirement definition, use case identification and preliminary architectural design.

Keywords:
Energy efficiency, Building, Cooperative, Facility manager, ESCO, Retailer, DR, Flexibility

Revision History

Revision	Date	Description	Author (Organisation)
V0.0	01.03.2018	New document	Álvaro Nofuentes (ETRA)
V0.1	20.03.2018	First round of contributions	ETRA, ICCS, ECO, ENG, ITE, HYP
V0.2	04.04.2018	Second round of contributions	ETRA, ITE, BYES
V0.3	16.04.2018	Last contributions and version ready for peer review	ETRA, BYES, CRE, VS
V0.4	24.04.2018	Integration of feedback received	ETRA, ICCS, ECO
V1.0	25.04.2018	Final refinement and ready for submission	Álvaro Nofuentes (ETRA)

INDEX

EXECUTIVE SUMMARY	9
1 INTRODUCTION	11
1.1 PURPOSE OF THE DOCUMENT.....	11
1.2 SCOPE OF THE DOCUMENT	11
1.3 STRUCTURE OF THE DOCUMENT	11
2 BACKGROUND	12
2.1 Overview	12
2.1.1 WiseCOOP.....	12
2.1.2 WiseCORP	12
2.2 Relevant actors	13
2.2.1 WiseCOOP.....	13
2.2.2 WiseCORP	14
2.3 Requirements	16
2.3.1 WiseCOOP.....	16
2.3.2 WiseCORP	17
2.4 High Level use cases	18
2.5 Primary use cases	19
2.5.1 WiseCOOP.....	19
2.5.2 WiseCORP	19
2.6 Architecture	19
2.6.1 WiseCOOP.....	19
2.6.2 WiseCORP	21
3 WISECOOP MODULES	23
3.1 Internal Service Bus.....	23
3.1.1 Context	23
3.1.2 Messaging mechanism	24
3.1.3 Internal Enterprise Service Bus.....	25
3.1.4 RabbitMQ	25
3.1.5 Common communication patterns enabled by the ESB.....	26
3.1.6 Integration with modules	26
3.2 Real-Time Monitoring.....	27

3.2.1	SMX Wrapper	27
3.2.2	AMI Wrapper	28
3.2.3	Real-time monitor	29
3.3	Portfolio Analytics.....	30
3.3.1	KPI Engine	30
3.3.2	Demand and Production forecast services	32
3.4	Demand Response.....	38
3.4.1	Relevant DR strategies.....	38
3.4.2	Architecture overview for implicit DR modules	39
3.4.3	Architectural overview for explicit Demand Response modules	41
3.5	User Interface	42
3.5.1	Dashboard	42
3.5.2	Prosumer detail	43
3.5.3	Geographical demand heat map	44
3.5.4	Customer profiling.....	45
3.5.5	Tariff designer.....	47
3.5.6	Tariff comparison.....	48
3.5.7	Energy trade assistant	49
3.5.8	Demand response campaigns.....	50
4	WISECORP MODULES	52
4.1	Internal Service Bus.....	52
4.2	Building Assets dispatcher and Integration.....	52
4.2.1	Asset dispatcher	54
4.2.2	Asset wrappers	55
4.3	Real-Time monitoring.....	57
4.3.1	Real-time monitor	57
4.3.2	KPI engine	57
4.4	Optimization and Analytics.....	59
4.4.1	Demand and Production forecast services.....	59
4.4.2	Comfort-based demand flexibility forecast module (DR framework)	59
4.4.3	Energy usage optimizer	61
4.5	User Interface	75
4.5.1	Dashboard	75
4.5.2	Map.....	78
4.5.3	Building detail	80

4.5.4 Demand overview.....	83
4.5.5 Production overview	84
4.5.6 Forecasting	86
4.5.7 Automation.....	88
4.5.8 Tariff comparator	89
4.5.9 Demand response.....	90
5 LINK WITH OTHER WISEGRID APPLICATIONS.....	91
5.1 WiseCOOP	91
5.2 WiseCORP.....	91
6 DPIA CONSIDERATIONS	92
7 CONCLUSIONS	93
8 REFERENCES AND ACRONYMS	94
8.1 References	94
8.2 Acronyms.....	94
9 ANNEXES.....	96
9.1 Message specifications for building assets monitoring and control	96
9.1.1 Sensor and Devices Monitoring and Control Interfaces specification	96
9.1.2 Interface with Asset Dispatcher	101
9.1.3 Configuration Parameters (Commissioning time)	103
9.2 Energy usage optimizer equations	107
9.2.1 General Energy Storage.....	107
9.2.2 HVAC.....	110
9.2.3 Global Optimization.....	111

List of Figures

Figure 1 – WiseCOOP.....	12
Figure 2 – WiseCORP	13
Figure 3 – Overview of interactions among the modules of the WiseCOOP application	21
Figure 4 – Overview of interactions among the modules of the WiseCOOP application	23
Figure 5 – Monolithic and micro-service architecture comparison [1]	24
Figure 6 – Example of different communication patterns [2]	25
Figure 7 – RabbitMQ [3]	26
Figure 8 – SMX Wrapper in the context of WiseCOOP.....	28
Figure 9 – AMI Wrapper in the context of WiseCOOP	29
Figure 10 – Overview of the structure of the RT monitor	30
Figure 11 – Structure of the KPI calculator modules.....	31
Figure 12 – Forecast server structure integration schema	33
Figure 13 – Implicit Demand Response Component Architecture	40
Figure 14 – Explicit Demand Response Component Architecture.....	41
Figure 15 – Dashboard mock-up	43
Figure 16 – Prosumer monitoring mock-up	44
Figure 17 – Current demand heat map mock-up	45
Figure 18 – Customer profile mock-up.....	46
Figure 19 – Customer profile mock-up.....	46
Figure 20 – Customer profile mock-up.....	47
Figure 21 – Tariff designer & editor mock-up	48
Figure 22 – Tariff comparer mock-up.....	49
Figure 23 – Energy trade dashboard mock-up	50
Figure 24 – Implicit DR - Dynamic tariff.....	51
Figure 25 – Explicit DR campaigns info mock up	52
Figure 26 – Building assets integration architecture.....	53
Figure 27 – BMS wrapper architecture	53
Figure 28 – SMX communication.....	54
Figure 29 – Overview of interaction among modules with the asset dispatcher	54
Figure 30 – Overview of communication between batteries and WG StaaS/VPP via WG IOP	55
Figure 31 – Structure of the KPI calculator modules.....	58
Figure 32 – Context-Aware Engine Conceptual Architecture.....	60
Figure 33 – General Optimization Process	62
Figure 34 – Energy model.....	64

Figure 35 – HVAC Model.....	66
Figure 36 – Global optimization Model.....	68
Figure 37 – Energy process Summary.....	71
Figure 38 – Input energy.....	72
Figure 39 – Stored energy	73
Figure 40 – Supplied energy	73
Figure 41 – Used energy	74
Figure 42 – Self-stored energy.....	74
Figure 43 – Consumed energy	75
Figure 44 – Dashboard mock-up	76
Figure 45 – Dashboard mock-up	77
Figure 46 – Map mock-up.....	78
Figure 47 – Map mock-up.....	79
Figure 48 – Map mock-up.....	80
Figure 49 – Building detail mock-up	81
Figure 50 – Building detail – monitoring mock-up	82
Figure 51 – Building detail – Demand distribution.....	83
Figure 52 – Consumption overview mock-up.....	84
Figure 53 – Production overview mock-up	85
Figure 54 – Forecasting mock-up	86
Figure 55 – Usage calendar mock-up	87
Figure 56 – Forecast analysis mock-up.....	88
Figure 57 – Automation mockup.....	88
Figure 58 – Tariff comparator mock-up.....	89
Figure 59 – Implicit DR mock-up.....	90
Figure 60– Explicit DR mock-up	90
Figure 61 – WiseCOOP interactions	91
Figure 62 – WiseCORP interactions.....	92

List of Tables

Table 1 – WiseCOOP actors' inventory	13
Table 2 – WiseCORP actors' inventory	14
Table 3 – WiseCOOP requirements	16
Table 4 – WiseCORP requirements.....	17
Table 5 – Demand forecast interface, error codes.....	35
Table 6 – Production forecast interface, error codes.....	37
Table 7 – Implicit Demand Response Strategy	38
Table 8 – Explicit Demand Response Strategy	39
Table 9 – HVAC Model.....	56
Table 10 – Gas meter model	57
Table 11 – CHP model.....	57
Table 12 – Information required for optimization	63
Table 13 – Information generated by the optimization	63
Table 14 – Functions of the Control System.....	64
Table 15 – General Energy Storage model definitions	65
Table 16 – HVAC Model definitions.....	66
Table 17 – Global Optimization model variables.....	68
Table 18 – Global Optimization definitions	69
Table 19 – Threat and feared events identification for WiseCOOP	92
Table 20 – Threat and feared events identification for WiseCORP.....	93
Table 21 – Acronyms list.....	94
Table 22 – Global Optimization Model, equalities matrix expression	116

EXECUTIVE SUMMARY

Achieving higher energy efficiency in buildings is one of the major issues that the European Commission wants to tackle in the Horizon 2020 programme. This objective can be reached thanks to the optimization of the energy demands or by means of smarter ICT strategies (such as DR activities).

This is also linked to the need of integrating increasing shares of RES installed close to the final users' premises. This necessity will enable prosumers to also operate and behave as producers, thus implementing additional flexibility into the system and increased challenges for medium and big users such as system operators and regulators.

WiseGRID will facilitate and promote active participation of consumers and prosumers by means of new market structures and innovative ICT services through the implementation of the WiseCOOP and WiseCORP tools.

WiseCOOP is the WiseGRID technological solution targeting aggregators of consumers and prosumers (particularly focused on domestic and small businesses), supporting them in their roles of energy retailers, local communities and cooperatives. The main goal of the solution is helping consumers and prosumers to work together in order to achieve better energy deals while relieving them from administrative procedures and cumbersome research.

WiseCORP is the WiseGRID technological solution targeting businesses, industries, ESCOs and public facility consumers and prosumers, with the objective of providing them the necessary mechanisms to become smarter energy players. By means of energy usage monitoring and analysis, proper information can be given to facility managers helping them to reduce energy costs and environmental impact.

The design of WiseCOOP and WiseCORP 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 tools 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 tools.

In order to better explain the modules of the tools, they have been clustered into different categories.

Firstly, WiseCOOP modules have been divided in:

Real-Time Monitoring

This section deals with the modules which collect energy readings from different meter sources (such as Unbundled Smart Meters or AMIs). This section also describes the approach and techniques to be used in order to provide the necessary KPI calculations of interest to the targeted users of the application.

Portfolio Analytics

The modules clustered in this section will provide to the aggregator the required functionalities for managing its portfolio of clients.

Demand Response

Here the required functionalities for the integration of the tool in a DR framework are explained.

User Interface

The User Interface is designed in order to give the appropriate information to its users, providing an insight of the behaviour of the portfolio in terms of energy usage and helping them to understand the characteristics of the aggregated prosumers.

On the other hand, WiseCORP modules have been divided in:

Building assets integration

This cluster deals with the wrappers that will collect and manage the information of the different building energy assets.

Real-Time monitoring

This section deals with the modules which collect data from different meter sources. Specifically, this section presents the approach taken by WiseCORP application in order to ingest all data required to perform its operation from a variety of assets and locations.

Optimization and Analytics

The modules here included will work with the optimized strategies related to the energy production and consumption of the building without forgetting users' comfort preferences.

User Interface

WiseCORP's UI is designed under the assumption that a single facility manager may be able to monitor one or more buildings/facilities. Each one of those buildings may also include different areas (sub-metering).



1 INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

The present document aims to describe and explain the results acquired from the Task 7.1 “WiseCOOP and WiseCORP Apps Design”. This task mainly deals with the design of the modules and functionalities of WiseCOOP (tool which is going to be used by small retailers and cooperatives) and WiseCORP (tool which is going to be used by facility managers and ESCOs). 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 tools but considering the relationships between each of them and the other WiseGRID tools having in mind that all of them 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 D7.2 “WiseCOOP and WiseCORP implementation and lab-testing”.

1.2 SCOPE OF THE DOCUMENT

The extent of this deliverable is focused in the needs that WiseCOOP and WiseCORP have to address in order to provide valuable functionalities to their 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 WiseCOOP and WiseCORP 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 understand the previous work carried out in other WPs related to WiseCOOP and WiseCORP. 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 describes the different modules of the WiseCOOP tool and how these modules address the required functionalities of the product. Straightaway, there is a similar section for the WiseCORP tool.

In order to give to the reader a high-level view of the tools, the modules have been clustered into different categories and thus a better understanding of the tools is achieved.

These two sections could be seen as the core of the document and the most technical part of it.

Once the deliverable has explained WiseCOOP and WiseCORP as single products, the authors give to the reader an overview of the position of the tools 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

This deliverable should be focused on the WiseCOOP and WiseCORP tools so this section will cover both tools simultaneously.

2.1 OVERVIEW

2.1.1 WiseCOOP

According to the description provided in deliverable D2.1, WiseCOOP is the WiseGRID technological solution targeting aggregators of consumers and prosumers - particularly focused on domestic and small businesses -, supporting them in their roles of energy retailers, local communities and cooperatives – which may have different objectives.

The main goal of the solution is helping consumers and prosumers to work together in order to achieve better energy deals while relieving them from administrative procedures and cumbersome research. In the particular scenario of increasing share of distributed renewable energy resources, this goal can be achieved by pursuing several objectives:

- Net metering: supporting the operation of communities of prosumers that invest in renewable energy sources aiming at reducing their environmental impact.
- Member profiling: clusters of consumers and prosumers with common energy usage patterns may be identified, allowing the aggregator to negotiate special terms (as for instance energy tariffs) particularly beneficial for those groups.
- Demand forecasting: by allowing the retailer to forecast the demand of its customers, optimized purchase of energy at the wholesale market is enabled.
- Tariff comparison: by offering members a tool for comparing their particular consumption with different available tariffs, those will have access to very valuable information to reduce their energy bills.
- Implicit price-based DR towards modulating the overall demand of the group to achieve a common objective (as, for instance, maximize usage of renewable energy sources produced within the group).
- Providing clear information to members to raise awareness on efficient energy usage and environmental awareness.



Figure 1 – WiseCOOP

2.1.2 WiseCORP

According to the description provided in deliverable D2.1, WiseCORP is the WiseGRID technological solution targeting businesses, industries, ESCOs and public facility consumers and prosumers, with the objective of providing to them the necessary mechanisms to become smarter energy players. By means of energy usage

monitoring and analysis, proper information can be given to facility managers helping them to reduce energy costs and environmental impact.

A key factor towards achieving these objectives is a proper retrieval and analysis of energy usage data, and visualization of meaningful information extracted from it. This information may include:

- Detailed visualization of energy demand at different areas of the building, helping facility managers to identify opportunities for enhancing energy efficiency;
- Energy tariff comparison, enabling a direct economic cost reduction by shifting to a more adequate tariff;
- Energy demand forecast, enabling medium to long term cost estimations and supporting operative decisions about the usage of the facilities;
- Demand flexibility estimation, allowing the execution of optimization algorithms that will – either automatically or by providing advices – shift demand in order to minimize economic costs – by maximizing self-consumption or moving demand to off-peak periods – or minimize environmental impact – by shifting demand to periods where green energy is available.



Figure 2 – WiseCORP

2.2 RELEVANT ACTORS

During the first year of the project, the consortium has identified who are the main actors that participate 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 WiseCOOP and WiseCORP frameworks.

2.2.1 WiseCOOP

The consortium has identified 17 actors that can be directly linked with the WiseCOOP.

Table 1 – WiseCOOP actors' 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 analyze electricity consumption, and have two-way communication capabilities.	System
Consumer	An entity connected to the grid, that consumes energy, i.e. a Prosumer without any production capabilities.	Person

Actor name	Description	Actor type
Distributed Energy Resource	Any type of generation units, storage units and load flexibility resources connected to the distribution network.	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
ESCO	<i>Energy Service COmpany</i> . Offers auxiliary energy-related services to Prosumers.	Organization
Forecast Provider	The organization that provides, upon demand, forecasts regarding certain variables (e.g. electricity demand, RES production, weather conditions, etc.)	Organization
HVAC	<i>Heating, ventilation and air conditioning</i> . An HVAC system maintains desired environmental conditions in a space.	System
Inverter	A power electronic device that converts DC electricity to AC and vice versa.	Device
Load Controller	A device that communicates with on-site electricity loads and has capabilities of sending control signals for increasing/decreasing the electricity demand.	Device
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
Producer	An entity connected to the grid that injects electricity to the grid.	Person
Prosumer	An entity that consumes and produces energy. There is no distinction between residential end-users, small and medium-sized enterprises or industrial users.	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
Smart Meter	An Electronic Meter with two-way communication capabilities.	Device
Supplier	Supplies and invoices energy to its customers.	Organization

2.2.2 WiseCORP

The consortium has identified 24 actors that can be directly linked with the WiseCORP.

Table 2 – WiseCORP actors' 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 analyze electricity consumption, and have two-way communication capabilities.	System
Building Management System	An automated system that monitors and controls the equipment of a building (ventilation, lighting, electricity infrastructure, etc.)	System

Actor name	Description	Actor type
CHP	<i>Combined Heat and Power</i> . A system that simultaneously generates electricity and useful thermal energy in one process from a single source of energy.	Device
Consumer	An entity connected to the grid, which consumes energy, i.e. a Prosumer without any production capabilities.	Person
Distributed Energy Resource	Any type of generation units, storage units and load flexibility resources connected to the distribution network.	System
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
ESCO	<i>Energy Service Company</i> . Offers auxiliary energy-related services to Prosumers.	Organization
Facility Manager	An entity responsible for the management of one or more buildings or other facilities in general.	Organization
Forecast Provider	The organization that provides, upon demand, forecasts regarding certain variables (e.g. electricity demand, RES production, weather conditions, etc.)	Organization
Gas Distribution Company	The organization responsible for the distribution of natural gas to final consumers.	Organization
Gas Meter	A device that measures and records the amount of gas (natural gas) consumed in residential, commercial and industrial buildings.	Device
HVAC	<i>Heating, ventilation and air conditioning</i> . An HVAC system maintains desired environmental conditions in a space.	System
Inverter	A power electronic device that converts DC electricity to AC and vice versa.	Device
Load Controller	A device that communicates with on-site electricity loads and has capabilities of sending control signals for increasing/decreasing the electricity demand.	Device
P2G Unit	<i>Power to Gas Unit</i> . A unit that converts electrical power to a gas fuel.	Device
Producer	An entity connected to the grid that injects electricity to the grid.	Person
Prosumer	An entity that consumes and produces energy. There is no distinction between residential end-users, small and medium-sized enterprises or industrial users.	Person
Public Authority	Governmental organization that administrates the public life on the level of a municipality.	Organization
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
Sensor	A device that monitors and processes specific input from the physical environment (e.g. light, heat, motion, etc.).	Device
Smart Meter	An Electronic Meter with two-way communication capabilities.	Device
Supplier	Supplies and invoices energy to its customers.	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” WiseGRID requirements, Use cases and pilot sites analysis”, however, in this subsection will be shown the ones related with the WiseCOOP and WiseCORP.

2.3.1 WiseCOOP

The requirements that WiseCOOP has to meet are:

Table 3 – WiseCOOP requirements

Requirement ID	Description	Type
COP_002	Intelligent decision algorithms to solve local imbalance between local production and local consumption	Functional and data requirements
COP_003	Users of the platform will have access to consumption measurements of the aggregated prosumers.	Functional and data requirements
COP_004	Users of the platform will have access to production measurements of the aggregated prosumers.	Functional and data requirements
COP_005	Users of the platform will have access to battery status of the aggregated prosumers.	Functional and data requirements
COP_006	WiseCOOP shall be able to cluster prosumers accordingly to their consumption pattern.	Functional and data requirements
COP_007	WiseCOOP shall be able to cluster prosumers accordingly to their production pattern.	Functional and data requirements
COP_008	WiseCOOP shall be able to assist the selection of the proper tariffs for each cluster.	Functional and data requirements
COP_009	WiseCOOP shall be able to aggregate the cooperatives' partners' available flexibility.	Functional and data requirements
COP_011	RES generation should never be limited (full priority) in times of potential grid congestion or local imbalance. Appropriate scenarios should be evoked to avoid grid problems.	Operational requirements
COP_013	Users of the platform shall be able to participate in DR programs as a whole.	Functional and data requirements
COP_014	WiseCOOP shall offer billing functionalities for cooperatives with a retailer role.	Functional and data requirements
COP_015	WiseCOOP shall be able to geographically map the clustered consumers to grid lines.	Functional and data requirements
COP_015	WiseCOOP shall be able to geographically map the clustered prosumers to grid lines.	Functional and data requirements
COP_017	WiseCOOP shall be able to have access to measurements of the interconnected entities and transform them accordingly.	Functional and data requirements
COP_018	Platform will require permission from prosumers to access and operate with their data.	Legal requirements
COP_020	Permission from domestic battery owners will be required in order to aggregate those batteries under Storage as a Service services.	Legal requirements
COP_021	WiseCOOP app could support gamification initiatives among the cooperatives members.	Functional and data requirements

Requirement ID	Description	Type
COP_022	WiseCOOP app should provide visualization of the KPIs of the energy management strategies and DR campaigns of the cooperative.	Functional and data requirements
COP_023	WiseCOOP demand shifting mechanisms will focus in implicit (price-based) demand-response.	The scope of the work
COP_024	An information channel from aggregator to portfolio members will be available.	The scope of the work

2.3.2 WiseCORP

The requirements that WiseCORP has to meet are:

Table 4 – WiseCORP requirements

Requirement ID	Description	Type
CRP_001	Platform users' shall have access to measured data of their related facilities.	Functional and data requirements
CRP_002	Manageable loads must include a control interface.	Functional and data requirements
CRP_003	Energy tariffs must be available.	Functional and data requirements
CRP_004	Energy price at wholesale electricity market must be available.	Functional and data requirements
CRP_005	Platform will have access to energy measurements (consumption and production) of the platform users' related facilities.	Functional and data requirements
CRP_006	WiseCORP shall show consumption and production evolution charts.	Functional and data requirements
CRP_007	WiseCORP shall show consumption and production KPIs.	Functional and data requirements
CRP_008	Platform users should get informed about the economic cost of the energy consumed in their facilities.	Functional and data requirements
CRP_009	Platform users' shall be able to watch the energy mix and CO ₂ emissions associated to their consumption.	Functional and data requirements
CRP_010	WiseCORP will be able to control smart devices (HVAC, lighting, industrial processes) for energy optimization purposes.	Functional and data requirements
CRP_011	WiseCORP will be able to minimize economic costs by using flexibility.	Functional and data requirements
CRP_012	WiseCORP will be able to minimize CO ₂ emissions by using flexibility.	Functional and data requirements
CRP_013	WiseCORP will be able to trade flexibility in the market.	Functional and data requirements
CRP_014	WiseCORP will allow end-users to choose usage of their flexibility (optimize energy demand, costs or CO ₂ emissions, or offer it in the ancillary services market).	Functional and data requirements
CRP_015	WiseCORP will exchange data with operational facilities.	Functional and data requirements
CRP_016	WiseCORP will need communication networks to incorporate various entities in data exchange.	Relevant facts and assumptions

Requirement ID	Description	Type
CRP_017	WiseCORP will need synchronization mechanism.	Functional and data requirements
CRP_018	WiseCORP app users shall have a secure protocol for data exchange and defined security rules for operational facilities integration to platform.	Functional and data requirements
CRP_019	One user of WiseCORP (i.e. a company or organisation) may be able to monitor and control several facilities from the platform.	The scope of the product
CRP_020	WiseCORP shall calculate demand and generation forecasts.	The scope of the product
CRP_021	WiseCORP will facilitate the participation of companies in demand-response programmes.	The scope of the product
CRP_022	WiseCORP shall allow its users to set constraints to the remote control of the smart appliances.	Operational requirements
CRP_023	Users of the platform (companies, municipalities, etc.) will explicitly be informed (upon registration) about the data required and how it will be managed.	Legal requirements
CRP_024	End-users of the platform (organisations, municipalities, etc.) will grant permissions to the platform to access and operate with their data.	Legal requirements
CRP_025	WiseCORP app should provide information on energy production and storage equipment status and their maintenance details.	Functional and data requirements
CRP_026	WiseCORP app should offer access to historical data of consumption, production of the facilities and KPIs of different management strategies in them.	Functional and data requirements
CRP_027	WiseCORP app could include a map with all the facilities and the relevant assets for better visualization to facilitate the managers/users.	Operational requirements
CRP_028	WiseCORP app shall provide visualization of the energy transferred to and from the EV fleet of the company, if it exists and uses its infrastructure as charging station.	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.

WiseCOOP is linked to 4 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 4: BATTERY STORAGE INTEGRATION AT SUBSTATION AND PROSUMER LEVEL

HL-UC 4 mainly contributes to the deployment, integration and demonstration of energy storage systems in the distribution grid. Furthermore, techniques for a smarter distribution grid as well as innovative and advanced demand-response mechanisms are tackled.

- 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.

The WiseCORP is linked to 3 HL-UCs and their descriptions are shown hereafter.

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

HL-UC 4 mainly contributes to the deployment, integration and demonstration of energy storage systems in the distribution grid. Furthermore, techniques for a smarter distribution grid as well as innovative and advanced demand-response mechanisms are tackled.

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

Integration of cogeneration in a public building or collective housing, in order to study the capabilities of this technology into the WiseGRID framework.

- 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.

2.5.1 WiseCOOP

The Primary Use Case in which the WiseCOOP is the main tool is:

- HL-UC 7_PUC_2_Dynamic aggregation of distributed energy assets and active participation into energy market

In deliverable D2.1 it is possible to find further information about this Primary Use Case.

2.5.2 WiseCORP

The Primary Use Cases in which the WiseCORP is the main tool are:

- HL-UC 4_PUC_1_Batteries management at prosumer level
- HL-UC 5_PUC_1_Thermal monitoring
- HL-UC 5_PUC_2_Cogeneration and HVAC management
- HL-UC 5_PUC_3_Comfort-based demand flexibility models
- HL-UC 5_PUC_4_Cogeneration and HVAC optimisation
- HL-UC 7_PUC_1_Dynamic management of demand side assets in tertiary sector

In deliverable D2.1 it is possible to find further information about these Primary Use Cases.

2.6 ARCHITECTURE

2.6.1 WiseCOOP

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

Data ingestion

The first step considered in the design 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 WiseGRID 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 WiseCOOP, these data flows are the ones containing energy readings from the members of the portfolio. There are two types of data sources for energy readings that will be integrated in the project: SMXs and AMIs. This subscription 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

Data analysis

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. These modules include:

- *KPI engine* module, in charge of extracting different indicators and patterns from the raw data, including profiling of the portfolio members according to different features of their energy demand and production
- *Demand and production forecast* module, providing forecasts for different aggregations of members of the portfolio

Operation and control

Under this group, different modules have been defined implementing specific tasks in order to fulfil the different functional requirements of the application. These modules comprise of the:

- *Tariff designer* module, allowing the users of the application to define energy tariffs that can be later on used for simulated bills or optimization
- *Tariff comparer* module, allowing the simulation of energy bills for the members of the portfolio under different pricing schemes
- *Billing management* module, closely related to the tariff comparer module. Allows the generation of bills for the members of the portfolio, according to the tariffs those have contracted
- *DR framework* module, implementing the complete set of functionalities required to enable the design and participation of both implicit and explicit demand response campaigns that will be tested in the project

Interaction with other applications

WiseCOOP will interact with other applications of the project, mainly with the following objectives:

- WiseCORP, WG STaaS/VPP, WiseEVP and WiseHOME will interact with WiseCORP during the participation in implicit and/or explicit demand response campaigns. Since implicit demand response campaigns is articulated by the creation of a dynamic price tariff, participation in implicit demand response campaign is extensible to any other application using tariff pricing as an input considered in its internal optimizations.
- WiseGRID Cockpit, the tool targeting DSO operators, will interact with WiseCOOP through the *Ancillary Services Market*, in order to request support for assisting the correct operation of the distribution grid when required. WiseCOOP participation in these market will be realized by triggering the corresponding explicit Demand Response campaigns on members of the portfolio with capability to participate in those, in order to cover the flexibility required by the DSO

Horizontal and support functionalities

Different modules will be used indirectly by the WiseCOOP 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*,

```

graph TD
    SD[Structural DB] --> KE[KPI engine]
    KE --> OPDB[(OPERATION DB)]
    OPDB --> RTM[RT monitor]
    SMX[SMX Wrapper] --> RTM
    AMI[AMI Wrapper] --> RTM
    RTM --> LTM[(LONG-TERM DB)]
    LTM --> PF[Production forecast]
    LTM --> DF[Demand forecast]
    LTM --> DRF[DR framework]
    LTM --> WGC[WG Cockpit]
    LTM --> WSM[Ancillary services market]
    LTM --> WSC[WiseCORP]
    LTM --> WSH[WiseHOME]
    WFP[Weather forecast provider] --> PF
    WM[Wholesale market] --> DF
    TP[Tariff provider] --> DRF
    WCG[WiseCOOP GUI] --> DRF
    WSM --> DRF
    WGC --> WSM
    BIL[Billing management] --> DRF
    TDC[Tariff designer & comparer] --> DRF
    RTM --> DRF
    PF --> DRF
    DF --> DRF
    subgraph BDP [Big data processing]
        KE
        PF
        DF
    end
    
```

2.6.2 WiseCORP

Data ingestion

1. Publication of data from *Wrappers* to the *WiseGRID 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 WiseCORP, these data flows include two main types: energy readings of the building (both demand and production, possibly including sub-metering), and status from sensors and controllable assets within the building. This subscription 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 with the data received from the different sources

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. These modules include:

- *KPI engine* module, in charge of extracting different indicators and patterns from the raw data, mainly related to the energy demand distribution in time to support the Facility Manager in the analysis of further actions to reduce demand, energy costs or to promote self-consumption in the facilities
- *Demand and production forecast* module, providing forecasts for the buildings monitored by the tool

Operation and control

Under this group, different modules have been defined implementing specific tasks in order to fulfil the different functional requirements of the application. These modules comprise:

- *Tariff advisor* module, allowing the facility managers to simulate their bills with different tariffs, thus comparing with historical data which tariffs are more beneficial
- *Energy Usage Optimizer* module, which will elaborate the schedule of the different controllable assets of the building according to their energy requirements model, in order to optimize the objective of the facility manager – mainly reduce economic costs or environmental impact
- *Asset Dispatcher* module, dealing with the communication with the different controllable assets and ensuring those follow the calculated schedule
- *DR framework* module, implementing the complete set of functionalities required to enable the participation of the facility in explicit demand response campaigns that will be tested in the project, keeping the occupants' comfort in the core

Interaction with other applications

The main interaction that will be implemented within WiseCORP, will be with the WiseCOOP application. WiseCOOP participates in the Ancillary Services Market of the DSO to offer the flexibility of the members of its portfolio to support the DSO maintain the quality of the supply in the distribution grid. The aggregator (using WiseCOOP) must therefore be able to select and send signals to the aggregated members to request them to shift their demand accordingly to meet the agreement with the DSO. The prosumers with more potential to offer significant modulation of their demand are those with bigger energy requirements, those targeted by the WiseCORP application.

Horizontal and support functionalities

Different modules will be used indirectly by the WiseCORP 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*, the *Tariff Provider* module, as well as the *Big Data platform* that will support the long-term storage and analysis. In addition, the *WiseCORP User Interface* is included in this category, providing web-based access to the information and functionalities provided by the other modules. Additionally, notification mechanisms (such as email, Telegram or Twitter) will be implemented in order to notify the facility manager of the significant events occurred in the system (e.g. triggering of explicit demand response campaigns).

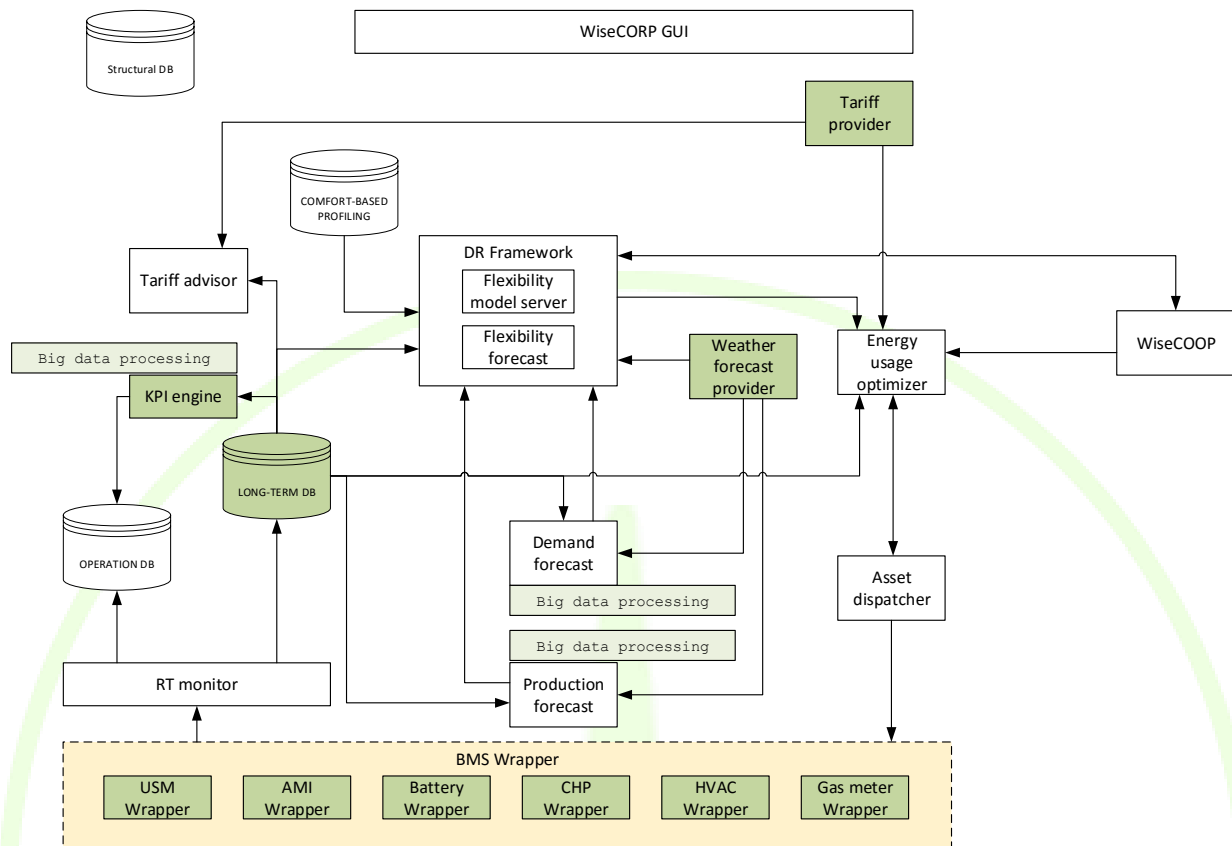


Figure 4 – Overview of interactions among the modules of the WiseCOOP application

3 WISECOOP 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 Application Programming Interfaces (APIs). They do not share data structures, database schemas, or other internal representations of objects.

- Components are stateless: they do not store any information related to previous requests. An incoming request can be sent to any instance of the service.

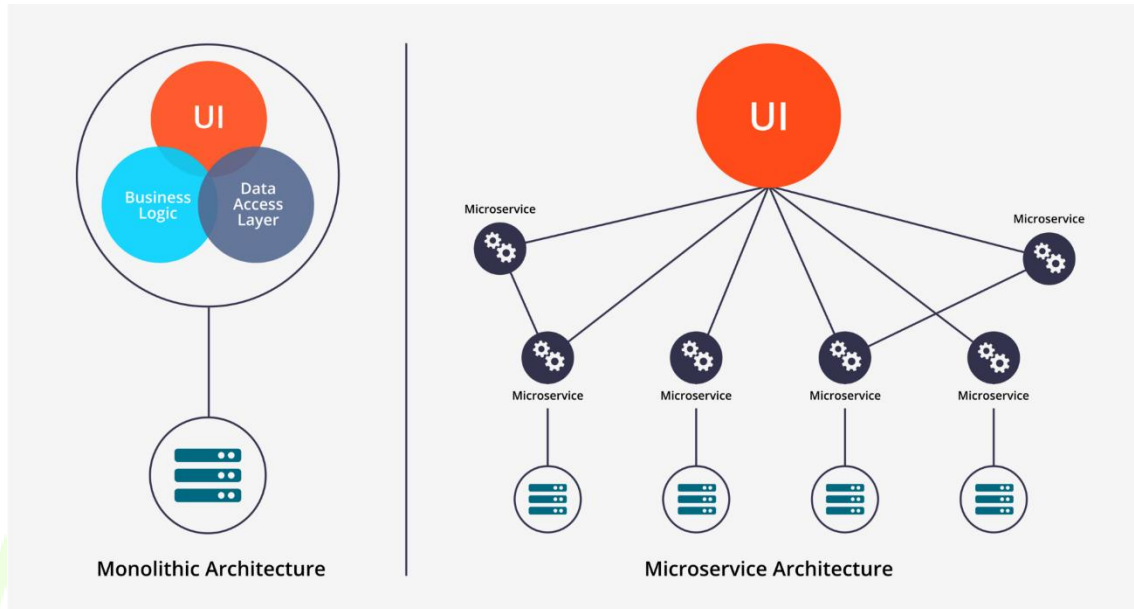


Figure 5 – Monolithic and micro-service architecture comparison [1]

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 monitoring of 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 seek to offer persistence of the messages in those cases when receiver is not able to process those (e.g. the process is down), and leverage on 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: the messaging platform must be open to different actors, and therefore it is necessary to handle policies for ensuring that each actor can only access to the information it has been granted access.

- 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 WiseCOOP.

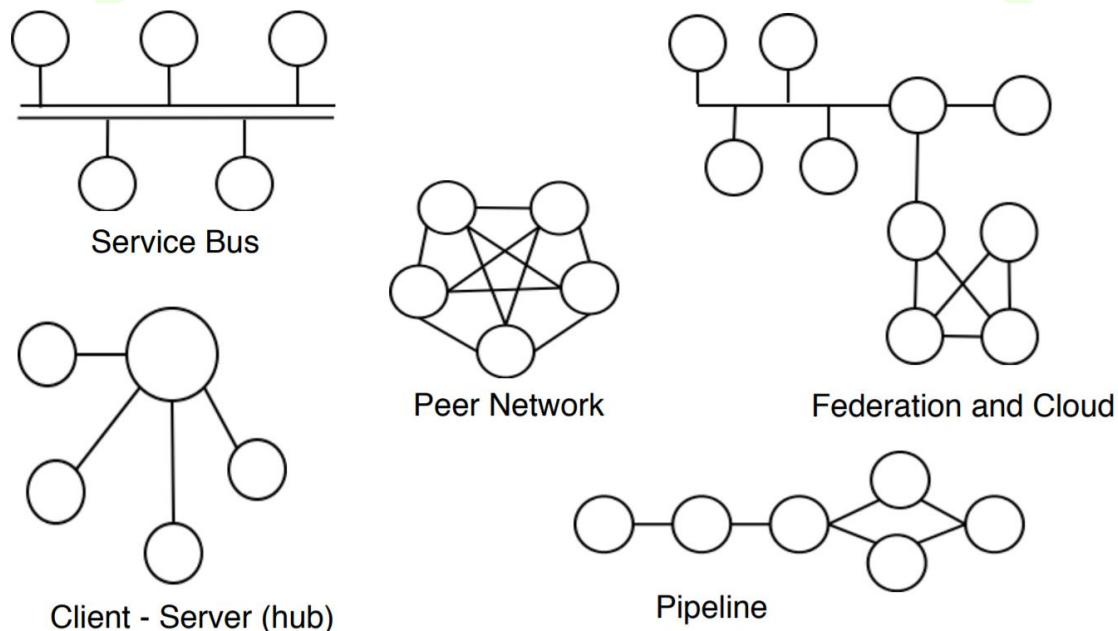


Figure 6 – Example of different communication patterns [2]

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 ESB 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.

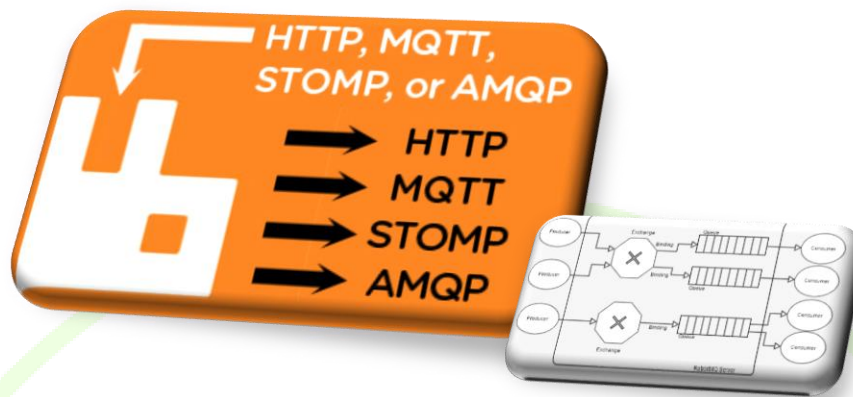


Figure 7 – RabbitMQ [3]

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.
- **Broker:** the central element to the communication, which matches the information published by publishers with the information requirements of the consumers, 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 exists 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 SMX Wrapper

SMXs and SLAMs are products of the H2020 project NobelGRID [4] that are also incorporated into the WiseGRID project. While SLAMs include a smart meter, both devices offer capability to install third party applications in order to offer extra functionalities on top of the smart meter readings. In the WiseGRID project, those devices are used to obtain energy readouts that are sent to the IOP to be processed by one or many applications as required.

This chapter presents the details of the wrapper in the context of the WiseCOOP application. Further details of this wrapper are documented in D4.2 [5].

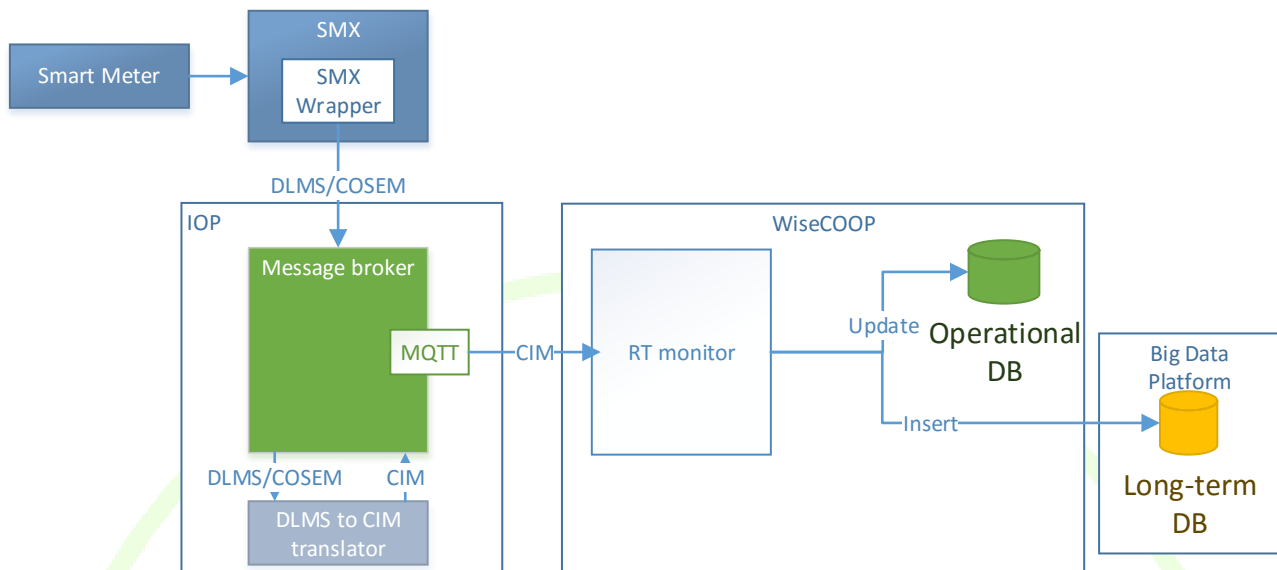


Figure 8 – SMX Wrapper in the context of WiseCOOP

SMX produces data in the form of DLMS/COSEM over MQTT (OBIS codes). This format is convenient for transmission (small size), but does not conform to the standard data model handled at application level (CIM).

In order to overcome this issue, the module *DLMS to CIM translator* will be used. This module of the WiseGRID IOP will perform three operations:

- Bundle all readouts referred to the same timestamp within a single message.
- Format the message to the CIM MeterReading common data model.
- Emit the result back to the IOP using MQTT under a different topic.

In Deliverable 4.2 “WiseGRID interoperable Integrated Process (WG IOP) is presented an example of the code of this module.

3.2.2 AMI Wrapper

Some of the pilot sites of the project have already deployed smart meters to a significant portion of their customers, which are being read by AMI systems. The project will therefore take advantage of the existence of such systems to demonstrate how the information collected by those can be published to the WiseGRID IOP and incorporated to the ecosystem of applications.

This chapter presents the details of the wrapper in the context of the WiseCOOP application. Further details of this wrapper are documented in D4.2 [5].

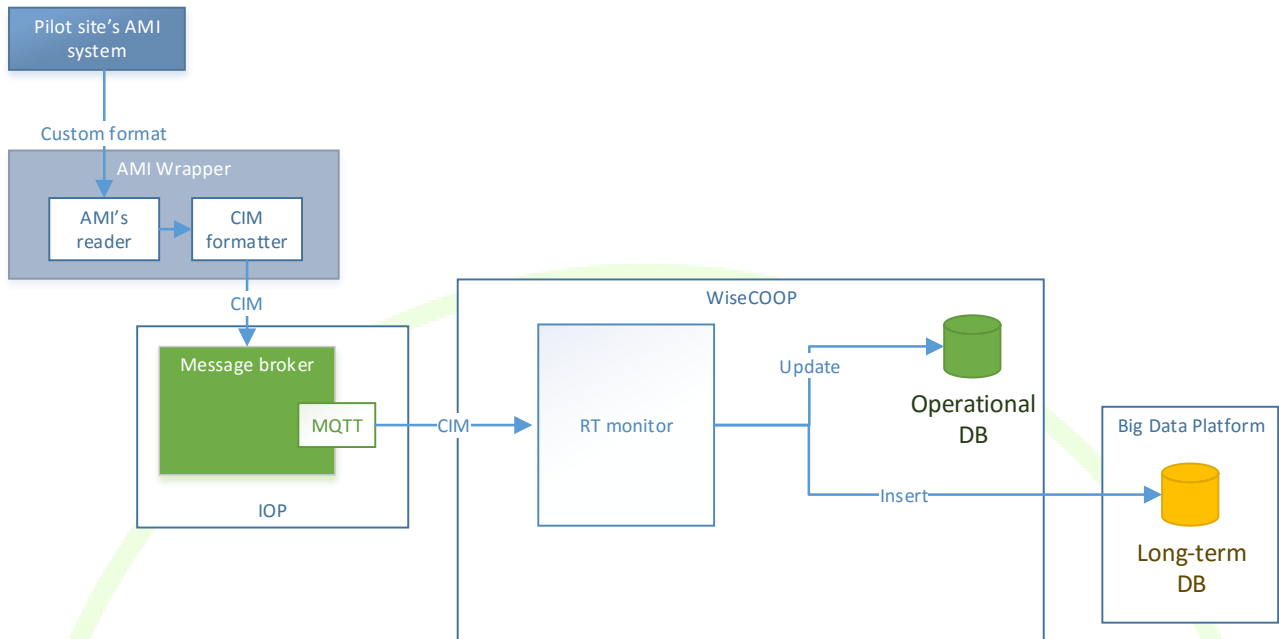


Figure 9 – AMI Wrapper in the context of WiseCOOP

The AMI Wrapper is the software module in charge of retrieving data from the corresponding AMI systems of the different pilot sites and makes it available to the applications of the project. In order to perform this task, the following steps are implemented:

1. Data from the AMI system is accessed. The way this access is performed depends on the vendor of the AMI system being integrated. The format of this data is normally proprietary, as defined by the vendor of the AMI system.
2. Data is transformed into messages that conform to the Common Data Model of the WiseGRID applications for energy readings: the CIM MeterReading format. For this, proper integration of the custom vendor formats needs to be developed.
3. CIM MeterReading messages are published into the WiseGRID IOP. The WiseGRID IOP will deliver this data to the appropriate applications, WiseCOOP being one of them. Data reaching the applications is always encoded using the CIM Common Data Model, independently of its source.
4. The Real-Time monitor of WiseCOOP is subscribed to this data flow, and responsible of processing and storing the required information.

3.2.3 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 [5]. 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 WiseGRID 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 WiseGRID 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 WiseGRID 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 to serve as a data source for KPI calculation and analysis.

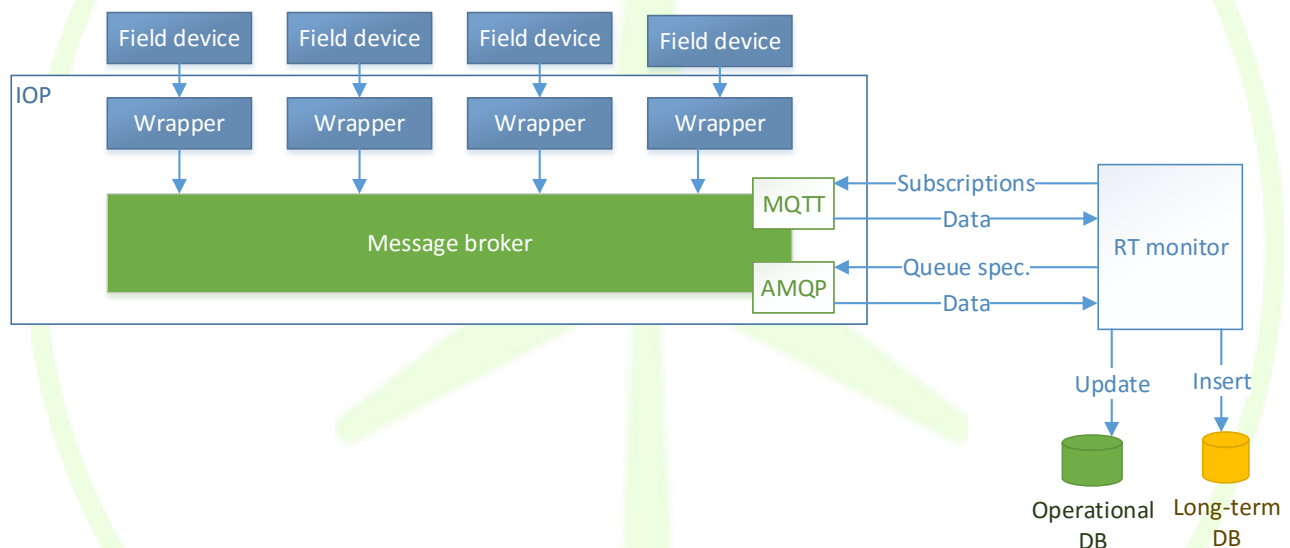


Figure 10 – Overview of the structure of the RT monitor

3.3 PORTFOLIO ANALYTICS

3.3.1 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 WiseCOOP, it will be in charge of the following analysis:

- Profiling of prosumer portfolio according to demand, economic cost and environmental impact patterns.
- Breakdown of portfolio demand according to contract type (domestic, industrial, etc.).

Since WiseCOOP will be also responsible for providing the required data to the WiseHOME application – the tool for end-users part of the portfolio of an aggregator (retailer or cooperative of prosumers) – some extra

indicators are required to be provided to individual prosumers:

- Cumulative daily consumption in real-time.
- Cumulative weekly consumption in real-time.
- Cumulative monthly consumption in real-time.
- Cumulative yearly consumption in real-time.
- Cumulative energy consumption for the same calendar day of previous year.
- Cumulative weekly energy consumption for the same week of previous year until corresponding day.
- Cumulative energy consumption for the same month of previous year until corresponding day.
- Cumulative energy consumption during the same previous year until corresponding day.
- Cumulative energy consumption for the same day (24h) of previous year.
- Cumulative energy consumption for the whole week (7 full days) of previous year.
- Cumulative energy consumption for the full month of previous year.
- Cumulative energy consumption during the entire previous year.

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

- Demand and production data of the members of the portfolio, as provided by the corresponding USM and AMI Wrappers.
- Composition of the energy mix and equivalent CO2 emissions of supplied energy, as provided by the energy mix provider.
- Type of prosumer: each one of the prosumers of the portfolio will be typified accordingly to the contracted tariff.

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).

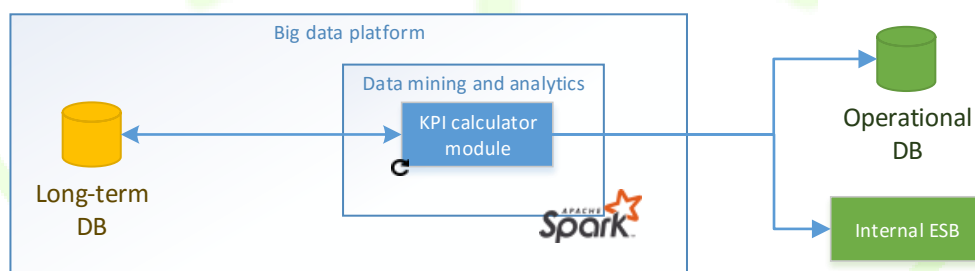


Figure 11 – Structure of the KPI calculator modules

The foreseen techniques to be used for the calculation of the KPIs are described hereunder.

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 WiseCOOP or WiseHOME application to provide insights in the behavior of the portfolio and assist in the decision support.

Machine learning – Clustering

Spark's Machine Learning Library [6] provides the implementation of a set of Machine Learning Algorithms that can take advantage of the features offered by the Spark framework, mainly its ability to parallelize and distribute computation of data across different machines under a cluster configuration.

In particular, within the context of WiseCOOP the Clustering algorithms offered by Spark will be used to perform analysis of the behavior of the portfolio, with regards to the economic and environmental impact of their demand. This will help the operator of WiseCOOP to gain knowledge on how the different segments of the portfolio behave accordingly to this parameters (how the demand of each member is usually distributed, how members of the portfolio react to DR campaigns or green energy usage awareness campaigns, etc.). As a result, the retailer/aggregator can use this information to properly design further awareness strategies, or accommodate tariffs to the actual needs of the prosumers.

3.3.2 Demand and Production forecast services

The demand and production forecast provider is a wrapper of the forecast algorithms with an easy integration with the WiseCOOP 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 12 contains the structure defined to integrate demand forecast and renewable forecast services into the WiseCOOP 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.

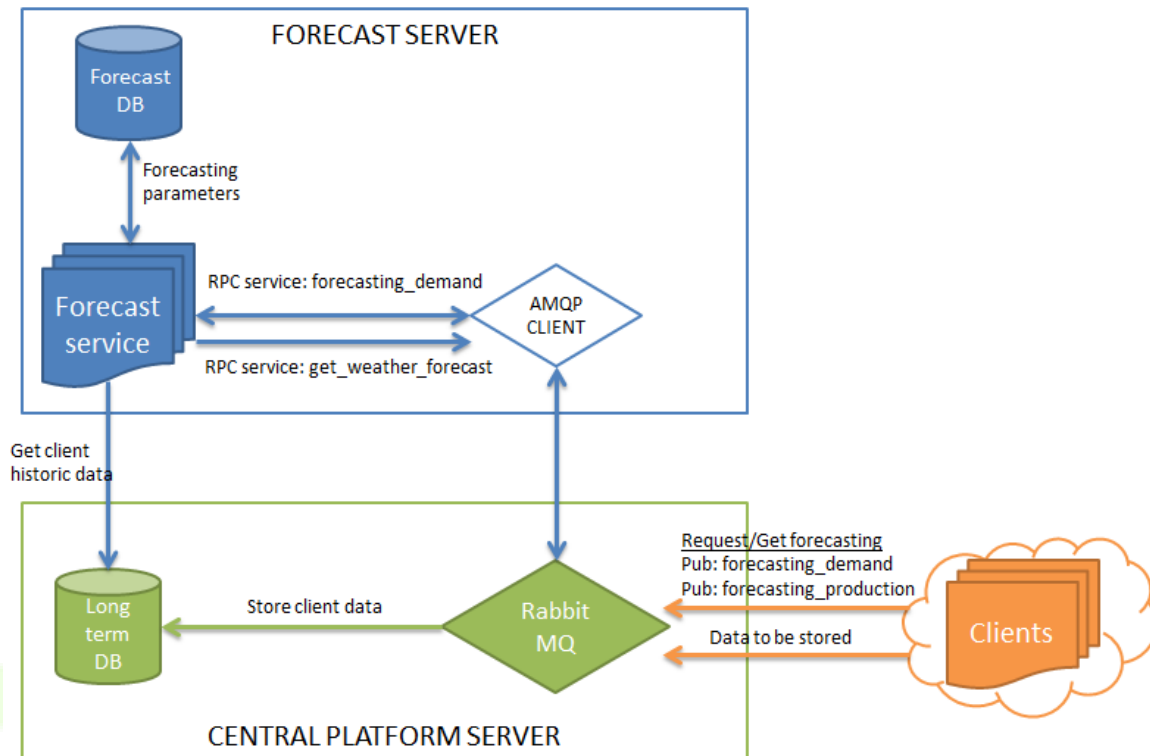


Figure 12 – Forecast server structure integration schema

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

1. The WiseCOOP 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 from the AMQP client, and will perform the forecasting based on historical data (stored in the Long Term DB) and weather forecast if proceeds.
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 DB 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 Point, 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 energy resources.

Finally, production forecast service requests 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

Concerning the forecasting algorithm integrated in the forecasting service, it has been implemented under CNEA (Cascaded Neuro-Evolutionary Algorithm) forecasting model premises that are based on neural networks algorithms. The CNEA forecasting model consists on some SVM (Support Vector Machine) in cascade with an optimal chose of input models, number of neurones and evolutionary training process. It is possible to find further information of this algorithm in deliverable 4.2 "WiseGRID interoperable Integrated Process (WG IOP)" and deliverable 12.2. "RESCO services and advanced models for smartening the Distribution Grid".

3.3.2.1 Demand Forecast Interface

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>)
- Payload¹:
 - 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.

¹ 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 in the database.

- 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 5 – Demand forecast interface, 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"
}

```

3.3.2.2 Production Forecast Interface

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 6 – Production forecast interface, 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 DEMAND RESPONSE

The purpose of this chapter is to provide an overview of the approach implemented for WiseGRID demand response optimization by providing an overview of the supported Demand Response mechanisms and the high-level architecture of the software modules of WiseCOOP that enable the respective campaign management.

3.4.1 Relevant DR strategies

Two different types of DR strategies are defined in this project, covering in that way the alternative demand response business cases examined in the project; i.e. implicit and explicit demand response. The high-level description of the implicit and explicit DR strategy is presented in the following tables:

Table 7 – Implicit Demand Response Strategy

Strategy Id	Implicit Demand Response
Strategy Description	The main idea behind this functionality is to provide retailers an IT tool for managing load imbalances through novel dynamic pricing schemes. This strategy allows energy market participants to define fine-grained billing strategies for a portfolio of consumers based on their specific operational profiles (energy consumption during the night/day, energy consumption based on the day of the week, etc.)
Metrics	Energy consumption (either aggregated or daily load profile information) and forecasted energy consumption (either aggregated or daily load profile information), billing price levels, billing period etc.
Workflow	<ol style="list-style-type: none"> 1. Day-ahead estimation of portfolio energy imbalance (DQ) time-series for a predefined period (target day) and for predefined intervals (e.g. hourly) given a generation and demand forecast; 2. Calculation of elasticity of demand for each asset for a given set of billing prices on a time-interval basis (e.g. hourly);

3. Optimisation of portfolio elasticity based on pricing levels; the pricing scheme is, thereafter, broadcasted to all assets in the portfolio;
4. Definition of optimal dynamic price (24-hour time-series) that alleviates retailer portfolio imbalance.

Table 8 – Explicit Demand Response Strategy

Strategy Id	Explicit Demand Response
Strategy Description	The main idea behind this functionality is to provide aggregators an IT tool for delivering demand profile modifications in near real-time through explicit demand response upon demand of the DSO. This strategy allows energy market participants to cover load imbalances or DQ requests by the DSO, in the short-term future (2 hours ahead) by dispatching device-specific control requests to buildings based on their specific operational profiles (current and near-future energy consumption of devices, devices' current status, indoor ambient conditions of buildings, consumer preferences, etc.)
Metrics	Energy Consumption (either aggregated information or daily load profile) following a normalization process through feature extraction analysis.
Workflow	<ol style="list-style-type: none"> 1. Total requested flexibility (DQ) time-series for a predefined period (2 hours ahead) and for predefined intervals (e.g. hourly); 2. Collection of demand flexibility for each building/device for a set of setpoints on a time-interval basis (e.g. hourly or 15-minute intervals); 3. Filtering of energy assets based on particular restrictions imposed during the DSO request (e.g. location of connection point) or other metrics related to the past performance of the assets or SLA restrictions; 4. Ranking of assets based on: <ol style="list-style-type: none"> a. their flexibility potential, b. number of DR triggers based on historical DR data, c. DR responsiveness of each asset based on historical DR data; 5. Clustering analysis for the definition of groups of assets with similar characteristics. Assets are clustered in groups which reflect groups with high, medium and low priority in a potential DR request; 6. Optimization is then performed in order to select the assets that cover the specific explicit DR request at hand, based on the aforementioned asset ranking.

3.4.2 Architecture overview for implicit DR modules

The following figure depicts a high-level architectural view of the components included in the implicit demand response module. The retailer is the main actor using this view of the DR architecture in order to balance the energy demand of its user portfolio with the available locally generated energy as well as that

purchased in the wholesale market.

The purpose of this module of WiseCOOP is to determine the appropriate price scheme for day-ahead application at portfolio level. This price scheme is, thereafter, broadcasted to all interested tools (i.e. WiseCORP, WG StaaS/VPP, WiseHOME and WiseEVP).

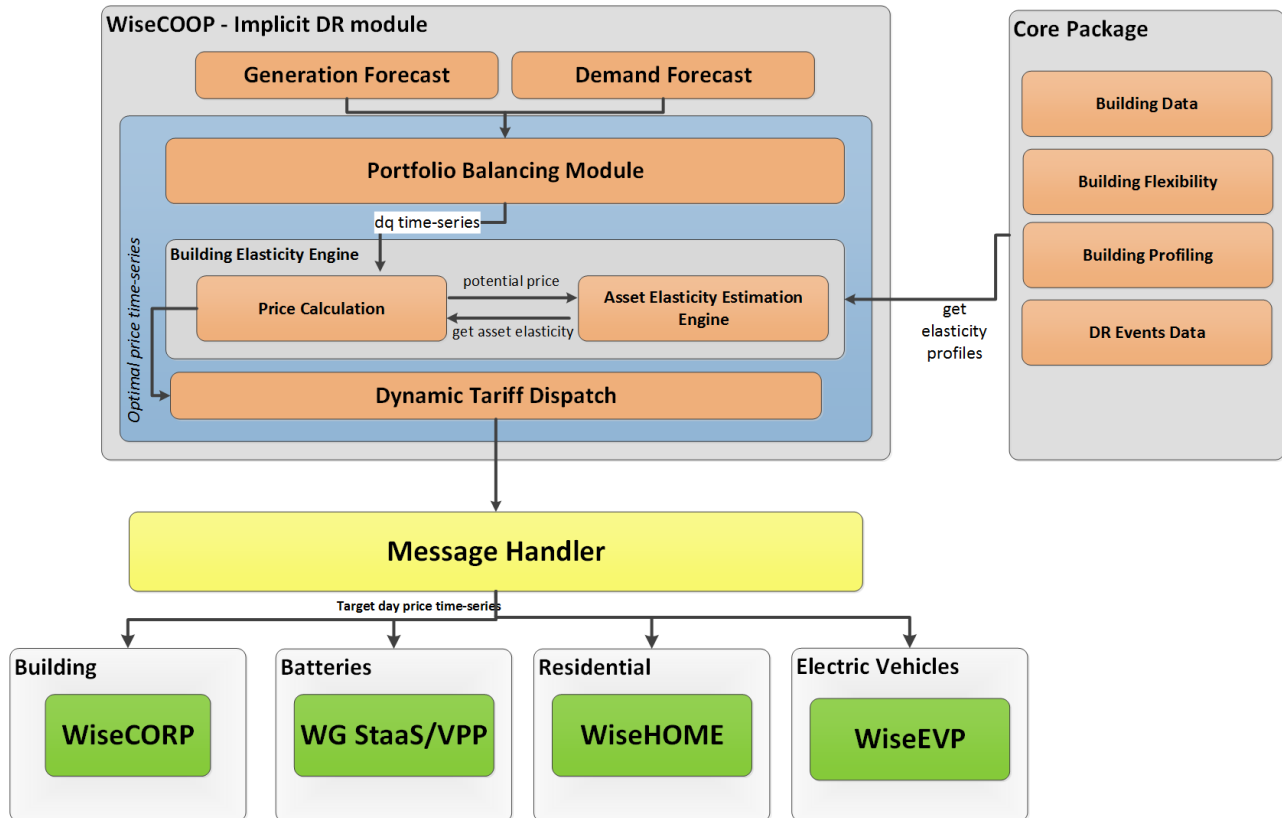


Figure 13 – Implicit Demand Response Component Architecture

The implicit DR module of WiseCOOP entails the following main functionalities:

- **Portfolio balancing module:** this module estimates the necessary demand modification that is required during the following day in order to ensure the balance of the retailer's portfolio. It uses the production (locally by retailer's generation assets) and the demand forecast in order to estimate the moments in time and the respective quantity of imbalance. This information is essentially equivalent to specific requests for demand profile modification using dynamic tariff schemes.
- **Asset elasticity estimation engine:** the purpose of this component is to estimate the total portfolio demand of the retailer based on a price level it receives from the price calculation component through estimation and aggregation of the building level demand. It leverages building price elasticity models to evaluate the potential demand modification at the building level and for the entire portfolio.
- **Price calculation:** this component keeps track of the price optimization process by exploring alternative pricing levels throughout the time slots of the target day and by invoking the component above to quantify the demand alteration. The outcome is the price time-series for the target day (following day) that optimally alleviates the imbalance calculated up front.
- **Dynamic tariff dispatch:** the purpose of this component is to communicate the calculated dynamic price time-series via the WG IOP so that the other WiseGRID products receive it for their internal purposes.

3.4.3 Architectural overview for explicit Demand Response modules

Respectively, the following figure depicts the modules for the implementation of the necessary functionality for the management of explicit demand response campaigns. In this case, requests are received from the distribution system operator or other market actor who requires provision of demand flexibility. This functionality is provided through the WiseCOOP tool, which initially analyses the DR signal and consequently ranks the available assets on a multi-criteria basis. Thereafter, a selection of assets that collectively meet the needs of the DR request receive a request to provide upward or downward flexibility. Each building is then responsible to translate this flexibility request into control signals to devices through a second level of optimization.

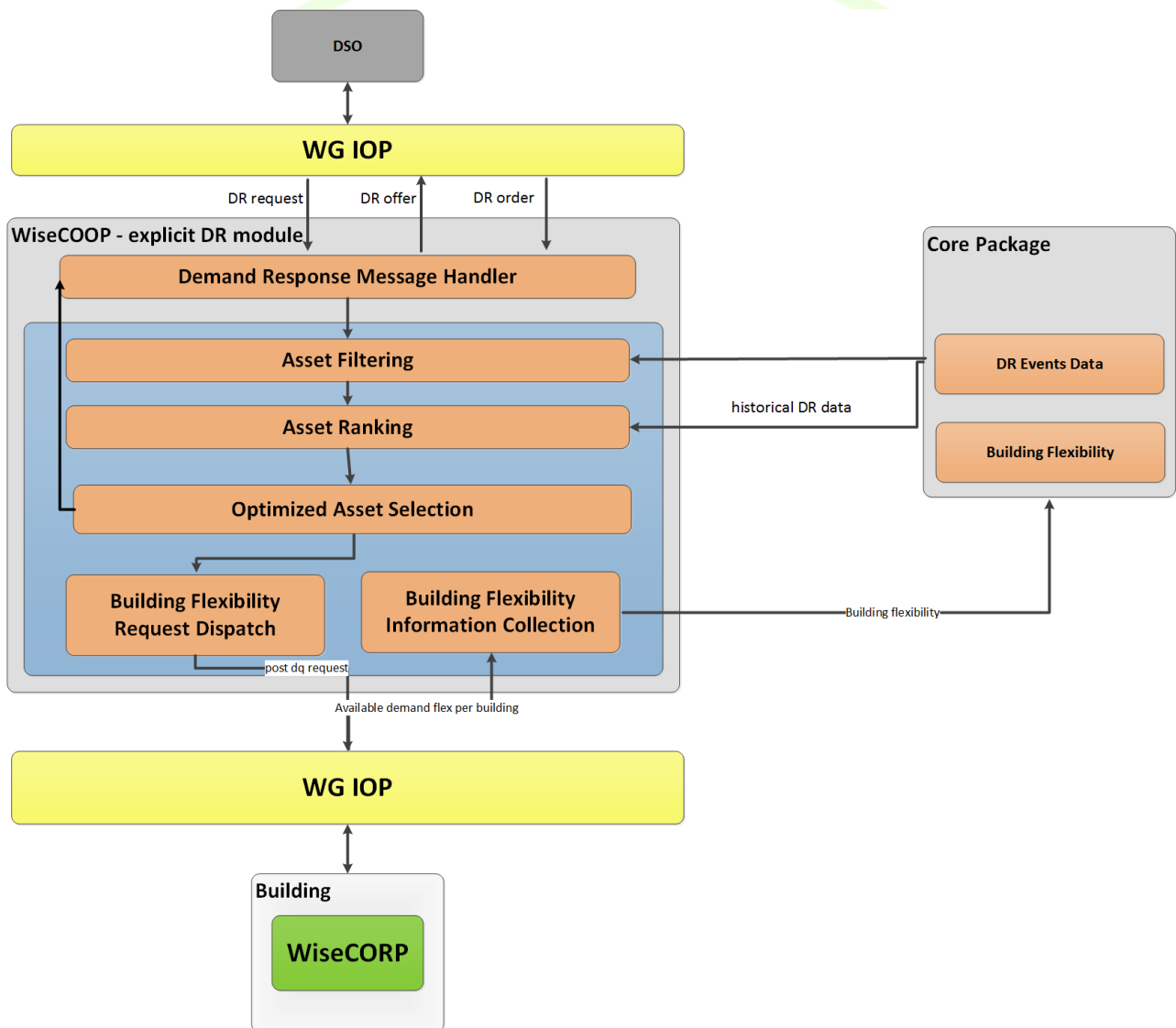


Figure 14 – Explicit Demand Response Component Architecture

A short description of the common core components of the DR framework is provided:

- **Demand Response Message Handler:** this component is responsible for the interactions with the IOP in order to ensure proper information exchange. It will also perform the hand-shaking with the DSO according to the USEF specifications to facilitate the negotiations between the DSO and all the actors who are willing to provide the requested flexibility.
- **Asset Filtering:** the purpose of this module is to eliminate any asset that is not eligible to provide flexibility for the specific DR request. There may be several reasons for this action, e.g. the location of its connection point on the grid, the maximum invocation number may have been reached, the asset may have declared itself unavailable due to maintenance, etc.
- **Asset Ranking:** this component aims to rank the available assets/buildings according to optimization criteria of interest to the aggregator. These may include the amount of flexibility offered, the flexibility provisions reliability of the asset, etc.
- **Optimised Asset Selection:** the role of this component is to select the assets from the list that will comprise the asset subset that will be called to offer flexibility. Furthermore, this component will also define how much flexibility per asset should be delivered in order to optimise some objective function.
- **Building Flexibility Request Dispatch:** this component is responsible for dispatching the flexibility requests to the specific assets that have been selected to participate in the specific explicit DR campaign. It will inform the buildings about the timing and exact amount of demand modification expected to fulfil the DSO request.
- **Building Flexibility Information Collection:** this component collects the available flexibility from the various assets/buildings that have a commercial agreement with the WiseCOOP user. These flexibilities are the starting points for the explicit DR module in order to perform all the aforementioned functionalities.

3.5 USER INTERFACE

WiseCOOP target users are aggregators of consumers and prosumers, including different roles such as retailers or cooperative of consumer and prosumers. The User Interface is designed in order to give the appropriate information to those kind of users, providing an insight of the behaviour of the portfolio in terms of energy usage and helping them to understand the characteristics of the aggregated prosumers. The UI also assists these organizations in the design and analysis of new tariffs, and gives the relevant information about the outputs of the Demand Response Optimization Framework.

3.5.1 Dashboard

Dashboard shows a selection of KPIs (calculated over last 30 days). Those include demand and production indicators per customer type, as well as the distribution of the energy supplied according to the type of consumer or prosumer to whom it has been supplied.

Dashboard

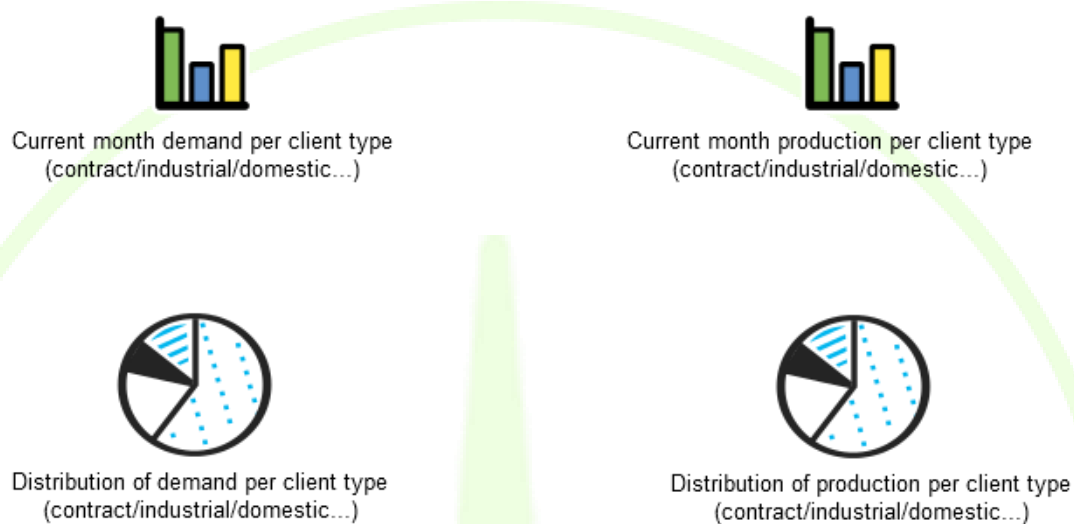


Figure 15 – Dashboard mock-up

3.5.2 Prosumer detail

This section provides access to raw data of individual customers. The form allows the selection of an individual contract of a prosumer (linked to a single smart meter) and a data range, in order to display its production and demand curves.

Prosumer Monitoring

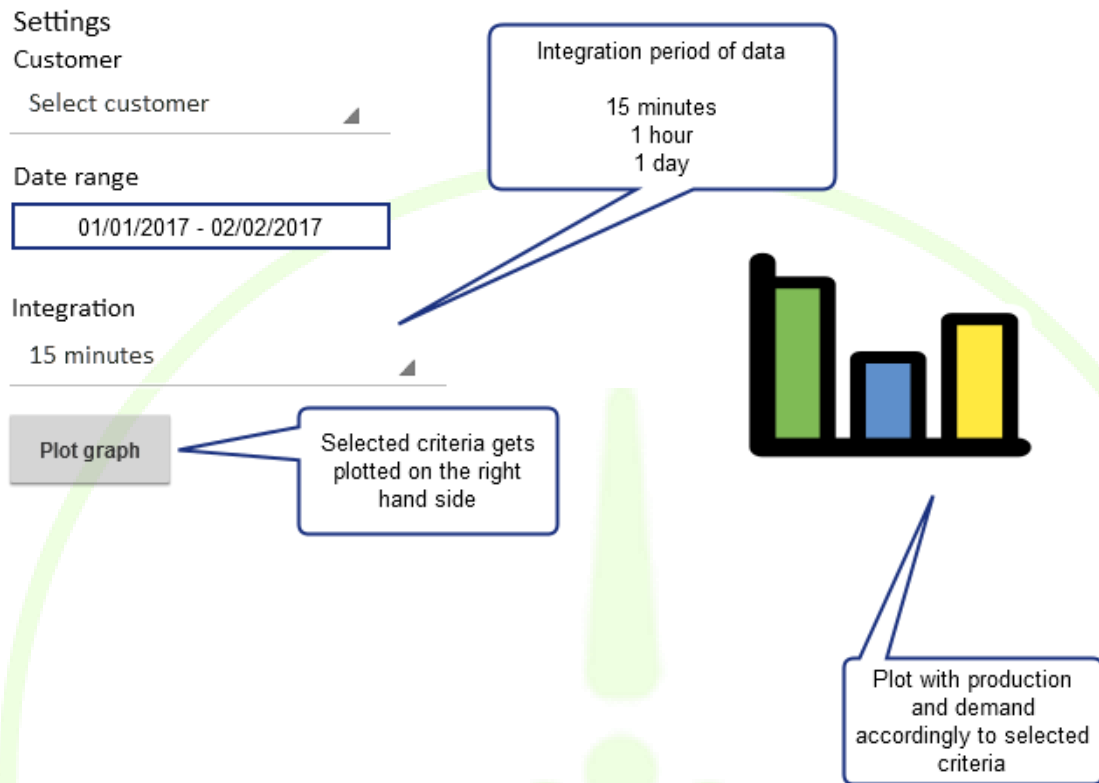


Figure 16 – Prosumer monitoring mock-up

3.5.3 Geographical demand heat map

This visualization depicts the geographical distribution of the last 30 days demand and production of the customers/members of the aggregator, in order to give an insight on the existence of localised peaks and gain information about the geographical location of the portfolio. Data may be filtered per customer type (contracted tariff) or profile (accordingly to the results of the portfolio profiler).

Current demand heatmap

- ☒ Industrial
- ☒ Domestic

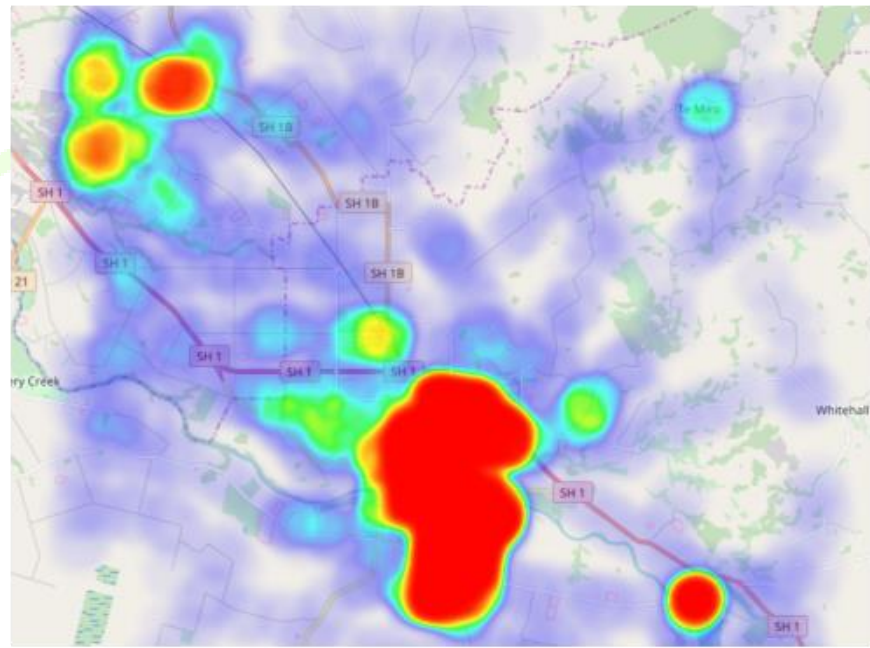


Figure 17 – Current demand heat map mock-up

3.5.4 Customer profiling

This section shows the output of the customer profiler module. This module is one of the KPI indicator calculation modules, and is in charge of analysing the behaviour of the portfolio accordingly to the usage they make of the energy (temporal distribution of demand and production, and associated economic and environmental impact).

For each classification, the results include the number of identified clusters, the details of the cluster centroid (average values for the considered properties), and the amount and percentage of customers included in that cluster. This visualization is of great value to get an insight on the different behaviour patterns identified and how the aggregated prosumers are distributed among those.

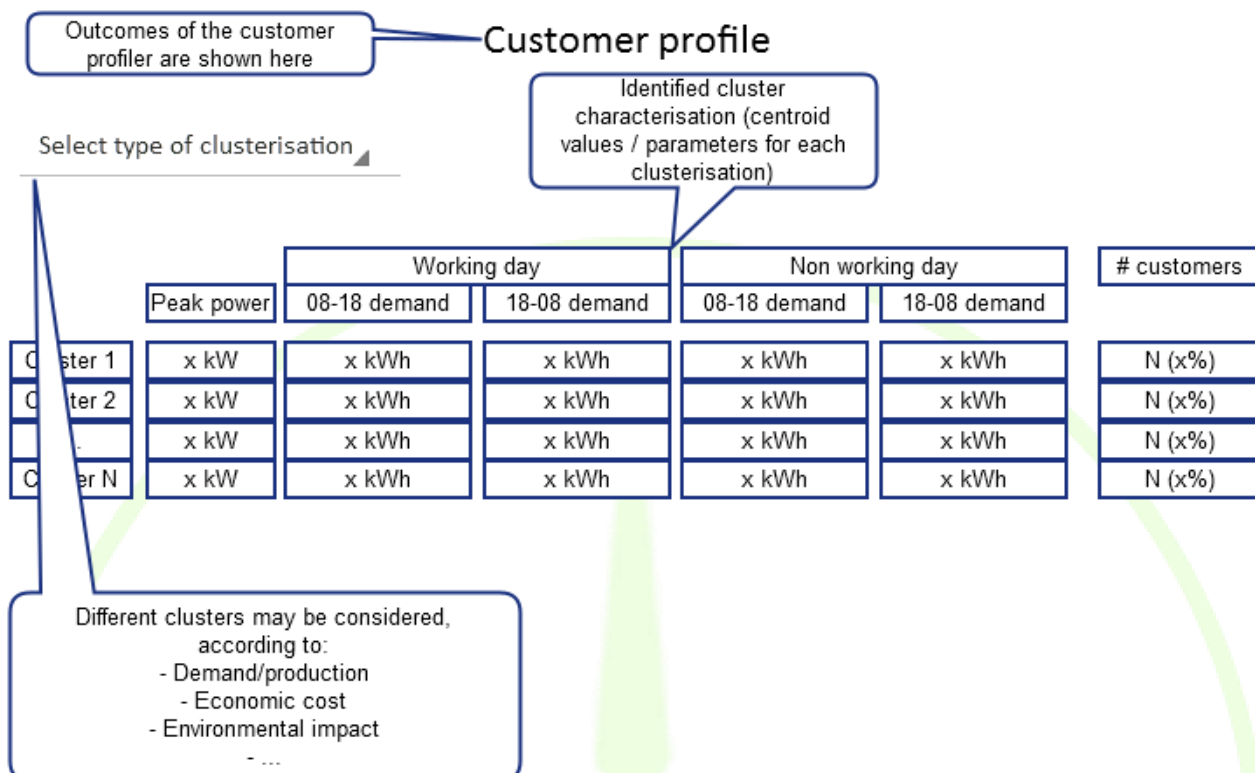


Figure 18 – Customer profile mock-up

A second view of the customer profiling section will allow the comparison of the behaviour of a single customer with the average members of its cluster. Two comparisons are included: feature-based and time-based.

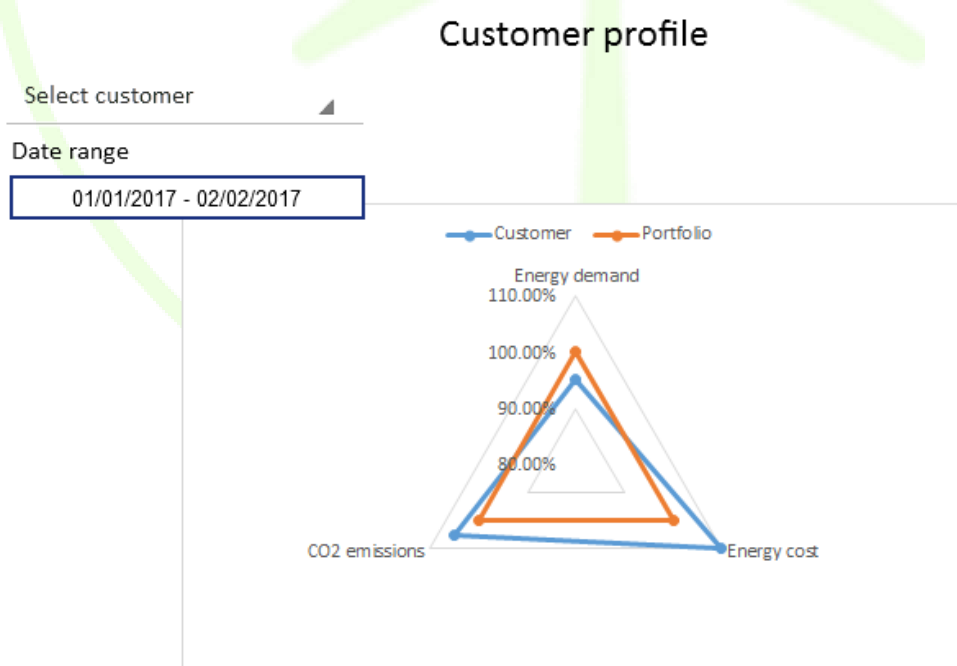


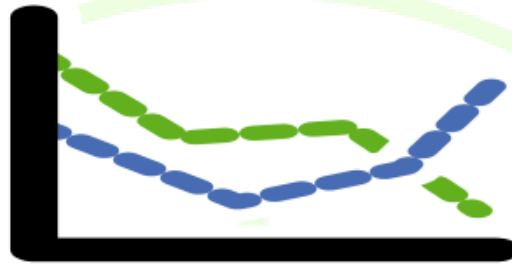
Figure 19 – Customer profile mock-up

Customer profile

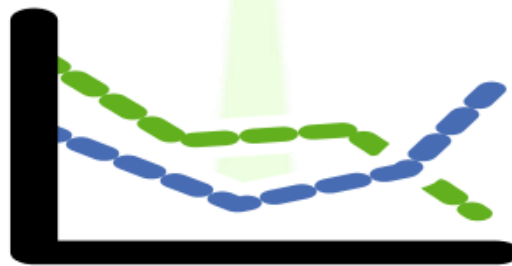
Select customer

Date range

01/01/2017 - 02/02/2017



Period demand comparison (customer vs portfolio average comparison)



Period production comparison (customer vs portfolio average comparison)

Figure 20 – Customer profile mock-up

3.5.5 Tariff designer

This section allows the retailer to define tariffs that can be later on used to simulate bills of individual customers (or groups of customers). This feature can be beneficial within the following contexts:

- Retailers can evaluate competitiveness of the tariffs they offer, and design new tariffs accordingly to the insights gained through the analysis of the portfolio.
- Cooperatives of consumers and prosumers can introduce tariff data that can be later on used to produce simulated bills and compare the costs they would face if different tariffs are contracted.

The tariffs are inserted accordingly to the tariff data model documented in D4.2 [5].

Tariff designer & editor

+

Select tariff ▲

Name Tariff xyz

Retailer Retailer x

Periods

+

Period 1

from: 00:00

to: 06:00

☐ Saturdays

☐ Sundays

☐ Holidays

☐ Special

☒ DST

Period 2

from: 06:00

to: 24:00

☐ Saturdays

☐ Sundays

☐ Holidays

☐ Special

☒ DST

Allows to populate tariff database with own and third party tariffs

Prices

	Energy	Power
Period 1	0.09 €/kWh	38.04 €/kW
Period 2	0.15 €/kWh	38.04 €/kW

Figure 21 – Tariff designer & editor mock-up

3.5.6 Tariff comparison

Based on the tariff database populated with tariff designer, this section allows the generation of virtual bills for a customer or group of customers (depending on the contract or the groups identified by the customer profiler), comparing those with the actual bill.

Tariff comparer

Select customer / group ▼

Select month: October 2017

Filter: ☐ Flat-rate ☒ Green energy ☒ Period

Priority: ☐ Energy cost ☒ Global cost

Process

Compares history of demand with available tariffs

Filter for considering only certain types of tariffs

Compare only cost of energy, or total costs associated to tariff

Measured values	Best energy plans		Cost (€)		Savings (€)
	Name	Retailer	Global	Energy	Energy
Peak power: x kW Demand: x kWh Cost (current tariff): 3500 €	Tariff A	Retailer A	3000€	1600€	500€
	Tariff B	Retailer B	3100€	1700€	400€
	Tariff C	Retailer C	3600€	2000€	-100€

Figure 22 – Tariff comparer mock-up

3.5.7 Energy trade assistant

The purpose of this section is showing relevant information from the wholesale energy market to facilitate the operation of the aggregator/retailer. Upon selection of a date range, it will show the following information:

- Wholesale market prices
- Energy mix
- Day-ahead demand forecasted for the portfolio, compared with the actual demand as reported by the smart meters
- Day-ahead production forecasted for the portfolio, compared with the actual production as reported by the smart meters
- Economic cost and environmental impact indicators associated to the actual demand
- Indicators of deviation identified between demand forecast and actual demand
- Indicators of deviation identified between production forecast and actual production

Energy trade dashboard

Date range

01/01/2017 - 02/02/2017

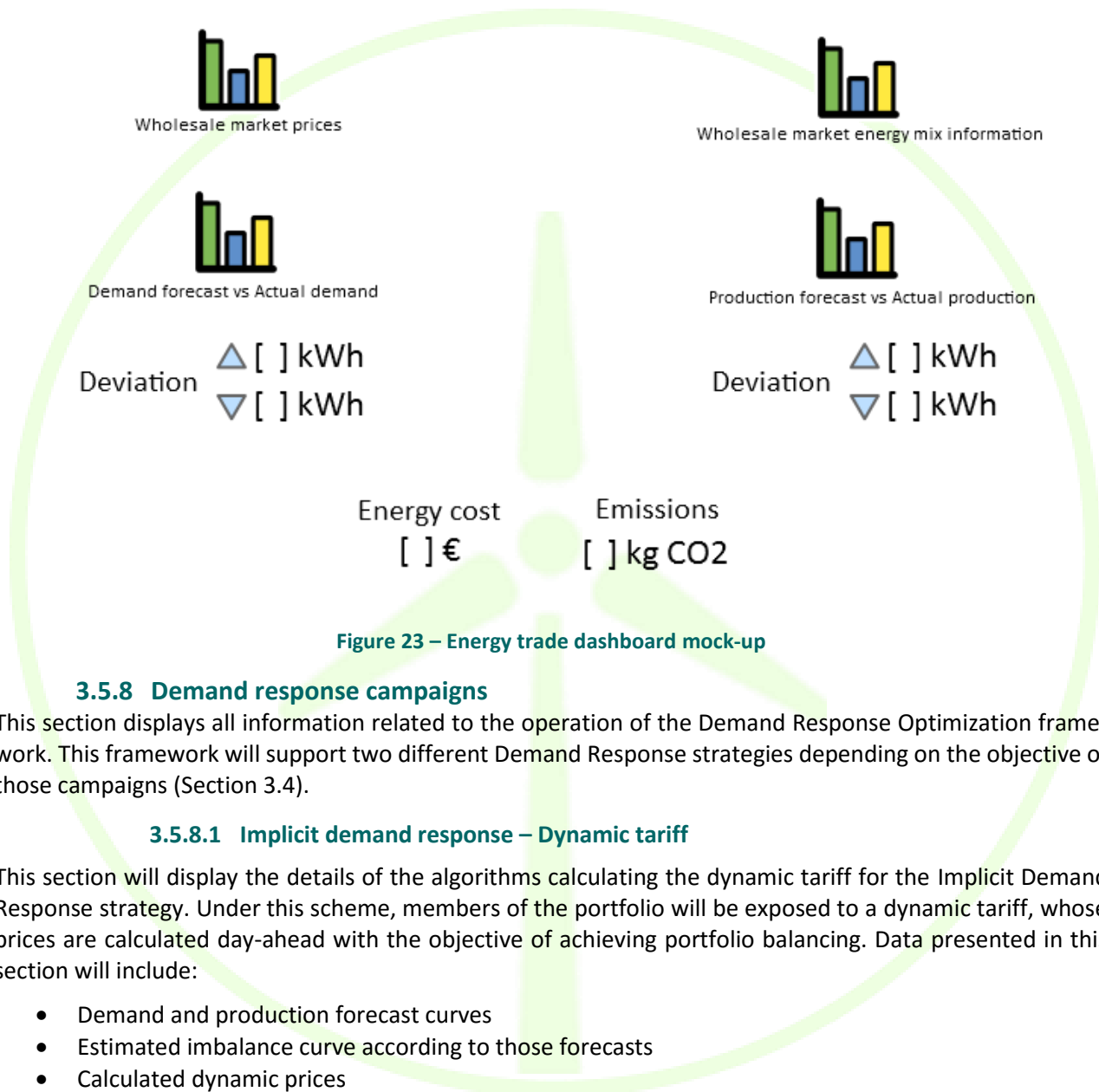


Figure 23 – Energy trade dashboard mock-up

3.5.8 Demand response campaigns

This section displays all information related to the operation of the Demand Response Optimization framework. This framework will support two different Demand Response strategies depending on the objective of those campaigns (Section 3.4).

3.5.8.1 Implicit demand response – Dynamic tariff

This section will display the details of the algorithms calculating the dynamic tariff for the Implicit Demand Response strategy. Under this scheme, members of the portfolio will be exposed to a dynamic tariff, whose prices are calculated day-ahead with the objective of achieving portfolio balancing. Data presented in this section will include:

- Demand and production forecast curves
- Estimated imbalance curve according to those forecasts
- Calculated dynamic prices

Implicit Demand Response - Dynamic tariff

Date range

01/01/2017 - 02/02/2017

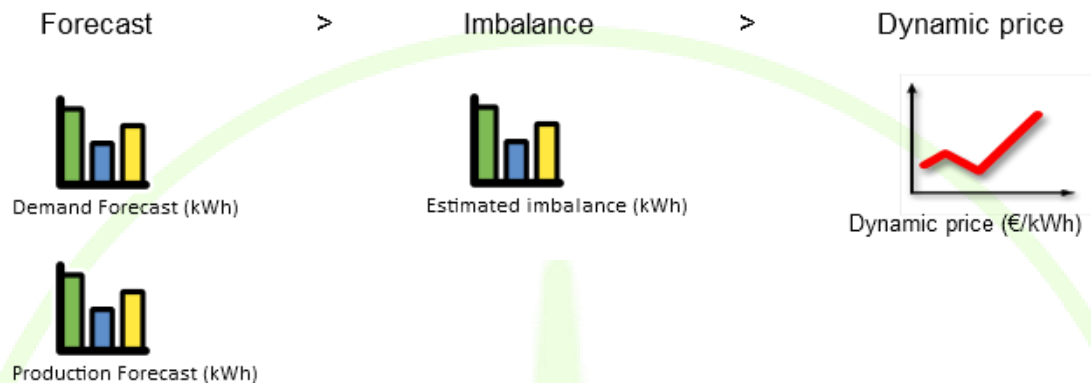


Figure 24 – Implicit DR - Dynamic tariff

3.5.8.2 Explicit demand response

This section will include the details of the Explicit Demand Response campaigns where the portfolio of WiseCOOP is participating. These kinds of campaigns are initiated by the DSO upon detection of a congestion problem on the grid. The Demand Response Optimization Framework is responsible of analysing the available flexibility and composing the offers to answer to the DSO. The information presented in this section includes:

- List of active explicit demand response campaigns indicating:
 - *Identification*
 - *Time slot* where flexibility has been requested
 - Requested *power reduction*
 - *Status* of the campaign: Request received from DSO, Offer sent to DSO or Offer accepted
 - Price: price of the offer composed by the WiseCOOP Demand Response Optimization Framework
- List of finalised explicit demand response campaigns indicating:
 - *Identification*
 - *Time slot* where flexibility has been requested
 - Requested *power reduction*
 - *Status* of the campaign: Offer rejected or Finalized
 - Price: price of the offer composed by the WiseCOOP Demand Response Optimization Framework

Explicit Demand response campaigns info.

Regulation area Downtown ▲

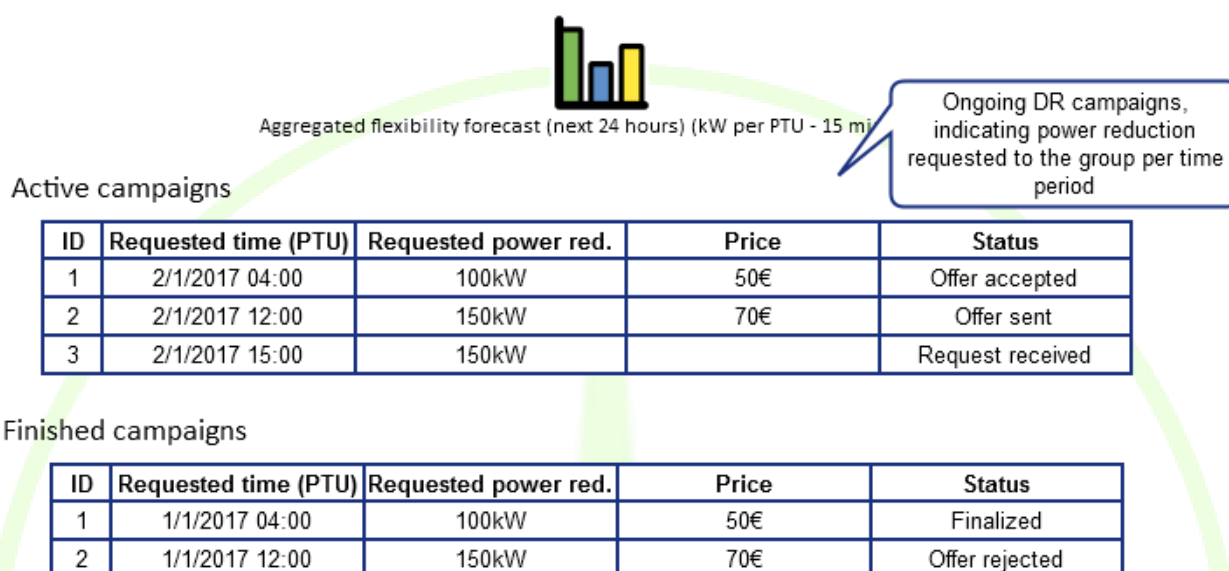


Figure 25 – Explicit DR campaigns info mock up

4 WISECORP MODULES

4.1 INTERNAL SERVICE BUS

As the Internal Service Bus is a common module in WiseGRID, it is possible to find the information concerning this module in section 3.1.

4.2 BUILDING ASSETS DISPATCHER AND INTEGRATION

The building assets integration objective is interfacing the different building equipment with the rest of the application modules. Each wrapper communicates with the different equipment and will give the needed information to the RT monitor. The data exchange table is defined on the following sections of this deliverable. On the other hand, each controllable equipment of the buildings also receives commands from the asset dispatcher, enabling then to control the building assets as required by the application.

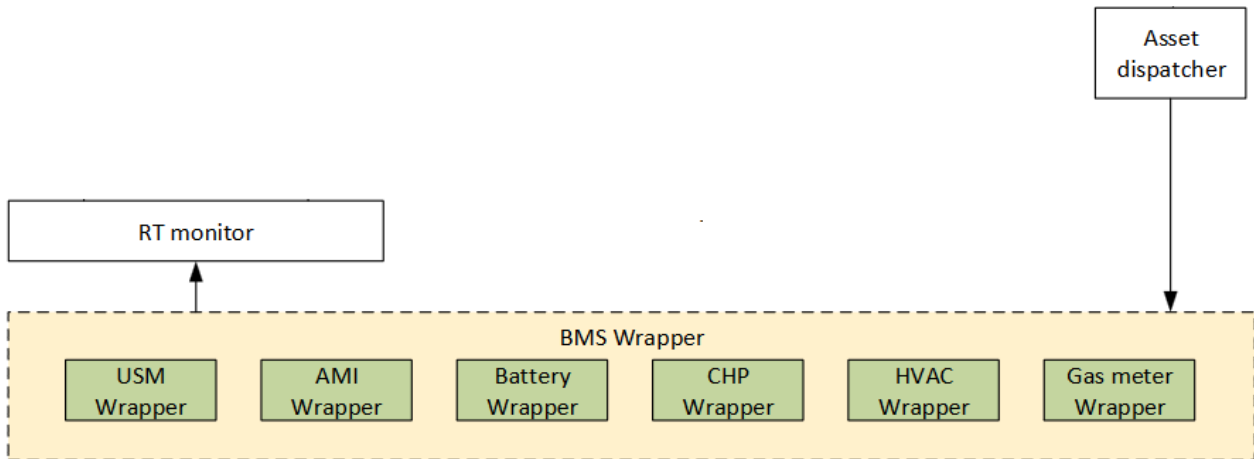


Figure 26 – Building assets integration architecture

The communication between the BMS wrapper and the rest of the application needs to be standardized to have the best simplified programming on the rest of the application modules. To keep a standardized method, the MQTT communication has been chosen for the BMS wrapper. All kind of communications that can be encountered, such as Modbus TCP, OPC UA or BACnet are “translated” into MQTT communication. This is reached by introducing a common gateway linked to the WiseCORP application:

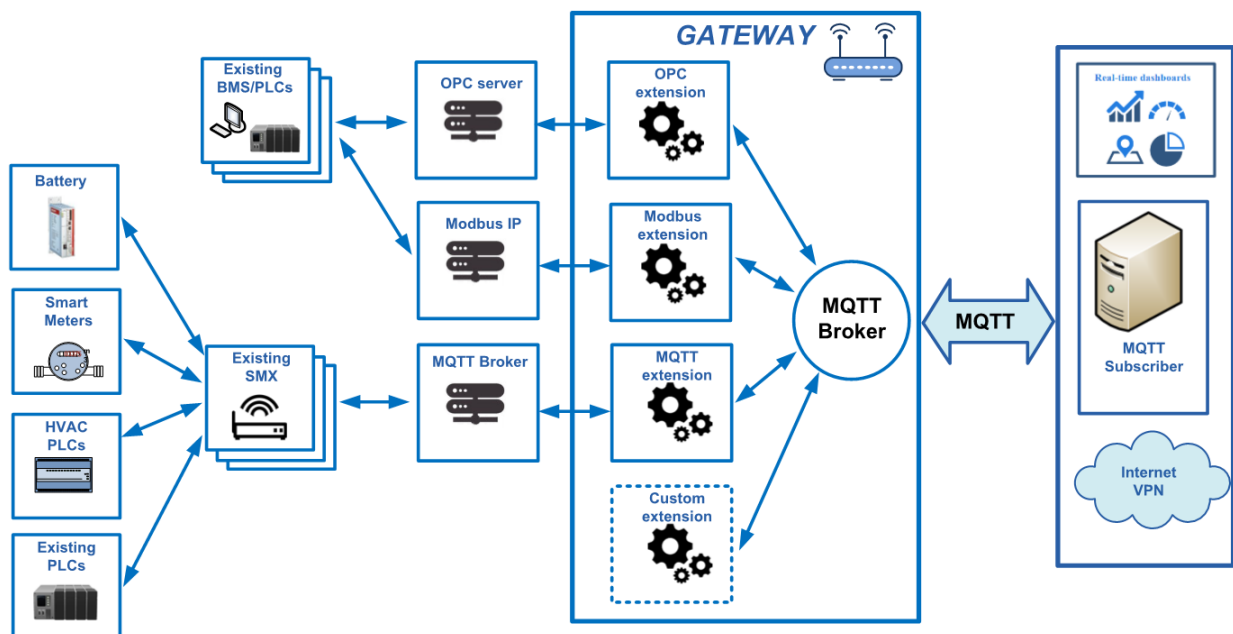


Figure 27 – BMS wrapper architecture

For some pilot sites on the WiseGRID project, SMX (outcome of the Nobelgrid project) are available and already linked to some equipment. On the WiseGRID project, this equipment is reused since MQTT communication are already developed. On a more general way, the SMX are reused for new equipment if it is technically possible.

Those SMX equipment contains natively MQTT broker running on port 1883 with RabbitMQ. A digital input, RS232 serial port, RS485 serial port and Ethernet connection are available on the equipment allowing integration of equipment controllable by any of those media:



Figure 28 – SMX communication

4.2.1 Asset dispatcher

The Asset Dispatcher module is a software module in charge of executing the appropriate schedule of the controllable assets, by continuously comparing the actual setpoint executed by the assets with the planned one. The corresponding commands to configure the appropriate setpoints are triggered whenever deemed required, upon detection of a mismatch between the schedule and the status.

This mechanism therefore builds the bridge between the modules composing the optimum schedule for the different assets of the building (the energy usage optimizer, which builds a baseline schedule by minimizing the cost or environmental impact of the energy to be consumed, and the DR framework, updating the baseline schedule in order to meet the constraints imposed by the demand response campaigns in which the building assets participate).

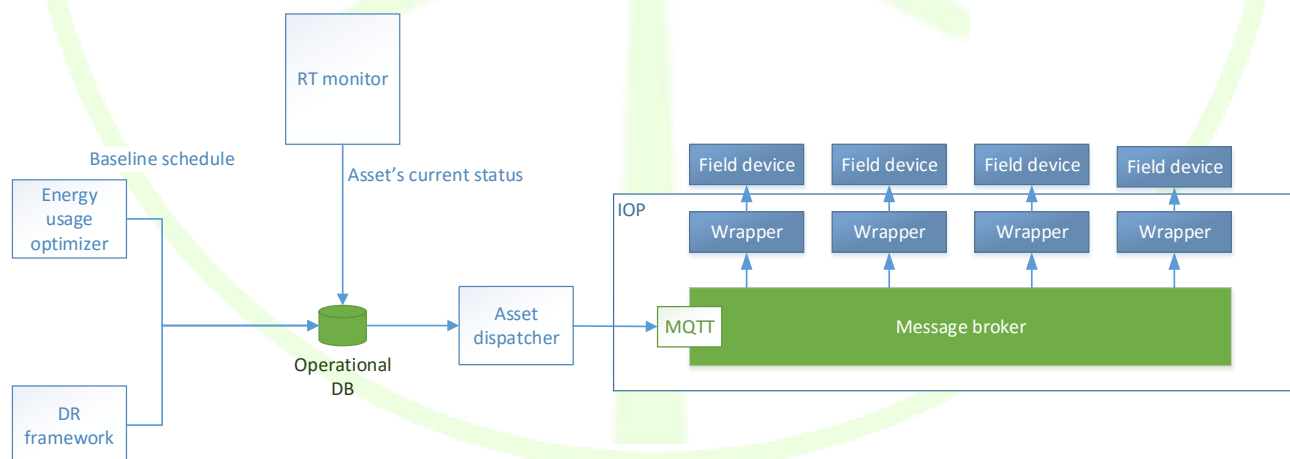


Figure 29 – Overview of interaction among modules with the asset dispatcher

The data flows and components involved in the operation of the Asset dispatcher are as follows:

- The Energy Usage Optimizer periodically calculates the next hours optimum schedule, accordingly to its optimization function. This schedule is composed by an array of setpoints per asset, indicating the setpoint to be executed in each one of the controllable assets at a specific time period. This schedule is stored in the Operational Database.

- The DR framework, which may update the baseline schedule in order to meet further requirements introduced by the participation of the assets in demand response campaigns. These updates get reflected in the operational database.
- The Real-time monitor module, which keeps track of the actual status of the controllable assets also in the operational database.
- The Asset Dispatcher compares for each time slot the current and the scheduled setpoints of each controllable asset, triggering the proper commands when those do not match, thus assuring that the controllable assets follow the corresponding schedule. These commands are triggered through the WiseGRID IOP platform, which will distribute them to the corresponding wrappers. The specification of those messages is documented in section 9.1.
- Upon a reception of a setpoint command, asset wrappers implement the necessary logic to configure the assets accordingly to the information received.

4.2.2 Asset wrappers

4.2.2.1 SMX wrapper

Information on integration of SMX devices for energy metering is presented in Section 3.2.1.

4.2.2.2 AMI wrapper

Information on integration of AMI systems for energy metering is presented in Section 3.2.2.

4.2.2.3 Battery wrapper

In order to translate vendor-specific protocols to a defined common data model that has been designed by the partners battery wrappers are implemented. This way any battery deployed in WiseGRID can be monitored and controlled in the same way, no matter which vendor it belongs to. The specification of the mentioned data model can be found in the deliverable D6.2. Whereas AMP and VARTA implemented the Battery Wrapper directly on their storage devices, BYES included it in the SMX which is also used for the smart metering. The overall structure of the communication paths is shown in the following figure.

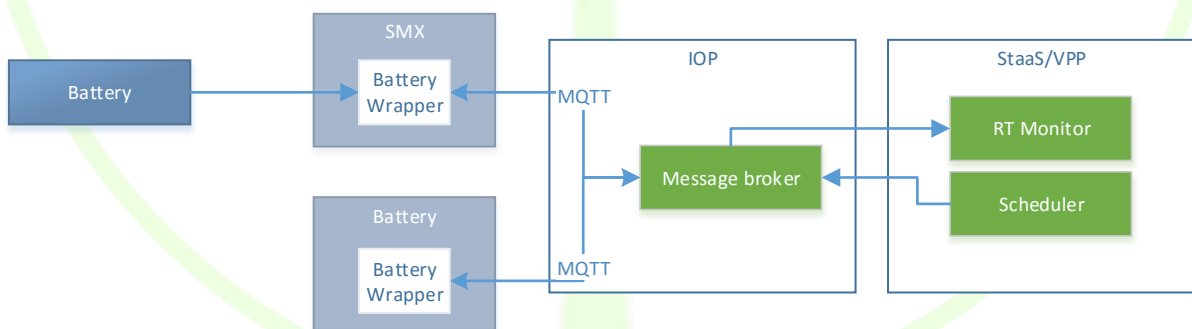


Figure 30 – Overview of communication between batteries and WG StaaS/VPP via WG IOP

Details regarding the implemented protocol and the vendor specific implementation of the MQTT clients and battery wrappers can be found in D4.2 and D6.2

4.2.2.4 HVAC wrapper

The HVAC system is a high source of flexibility on a building. The HVAC wrapper models the consumption of the system as a function of heat demand. The heat demand is therefore calculated, addressing the thermal model of each zone examined. The Thermal model contains the thermal characteristics of a building area together with the heat gains produced by the context environment in the area and the desired temperature set point.

Configuration parameter are fixed parameter and output parameter are dynamic parameter.

Table 9 – HVAC Model

Parameter	Description	Units	Type
Configuration parameters			
ID	ID of the HVAC		Float
operation mode	Operation mode of the HVAC system: cooling, heating.	--	String
cooling capacity	Cooling capacity of the HVAC system	W	float
heating capacity	Heating capacity of the HVAC system	W	float
cooling efficiency	Cooling efficiency of the HVAC system (EER)	W_i/W_e	float
heating efficiency	Heating efficiency of the HVAC system (COP)	W_i/W_e	float
Output parameters			
status	Status/Mode/Set-point	--	Complex
heat demand	Heat demand of the thermal zone. The heat demand is calculated in the Thermal Zone model	--	Complex
interior temperature	Internal temperature in the thermal zone at the initial time	°C	float
Power	Estimated power consumption of the HVAC system	W	float
final temperature	Estimated final temperature in the thermal zone	°C	float
Input parameters			
Set Point	Temperature desired	°C	float

The data model, defined by the assets dispatcher is shared in the MQTT protocol (see section 9.1). From the device itself to the gateway, the communication is specific in each test site and an adaptation is done depending on the need of the test site.

4.2.2.5 Gas meter wrapper

In complement to the CHP, a gas meter can be integrated to follow the gas consumption of the test site. Depending on the device installed, the frequency of the data acquisition can be different from an equipment to another. Each gas meter will provide the counter value of the accumulated gas consumption since installation. This means that, in order to have the consumption on a certain period, the delta between two figures needs to be calculated by the system.

Table 10 – Gas meter model

Parameter	Description	Units	Type
Output parameters			
ID	ID of the meter		Float
Frequency of metering data	Frequency of the data acquisition		float
Gas consumption	Gas consumption at the particular meter	m ³	float

This data model is shared using MQTT protocol to the rest of the application.

4.2.2.6 CHP wrapper

The CHP devices are a source of energy on a building and so can be part of the flexibility possibilities of the building by adjusting its output power.

The same way than the HVAC wrapper, configuration parameters are fixed parameter and output parameter are dynamic parameter.

Table 11 – CHP model

Parameter	Description	Units	Type
Configuration parameters			
ID	ID of the CHP		Float
operation mode	Operation mode of the HVAC system: cooling, heating.	--	String
Nominal Power	Heating efficiency of the HVAC system (COP)	W _t /W _e	float
Output parameters			
status	Status/Mode/Set-point	--	Complex
Power	Power production of the CHP	W	float
Potential increase	Estimated power increase of the CHP	W	float
Input parameter			
Power	Required power for the CHP	W	Float

All these information is shared with WiseCORP application through MQTT communications (see section 9.1).

4.3 REAL-TIME MONITORING

4.3.1 Real-time monitor

As the Real-time monitor is a common module in WiseGRID, it is possible to find the information concerning this module in section Section 3.2.3.

4.3.2 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 WiseCORP, it will be in charge of the following analysis:

- Ratio of self-consumption

- Economic indicators over energy supplied by retailer
- Environmental indicators (equivalent CO₂) over consumed energy
- Contribution of individual buildings to overall consumption of the building portfolio
- Evolution of the energy demand of the building
- Contribution of individual buildings to overall production of the building portfolio
- Evolution of the energy production of the building
- Breakdown of the demand accordingly to the supply source (self-consumption, supplied energy from RES, supplied energy from non-RES)
- Analysis of active power demand
- Breakdown of the demand accordingly to the tariff period (peak, super-peak, off-peak)
- Building energy efficiency indicators (demand per square meter, demand per employee, demand per working hour)

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

- Demand and production data of the different buildings, as provided by the corresponding USM and AMI Wrappers
- Electric measurements of the buildings (active/reactive power)
- Composition of the energy mix and equivalent CO₂ emissions of supplied energy, as provided by the energy mix provider
- Contracted tariff per building

As also explained in section 3.3.1, 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).

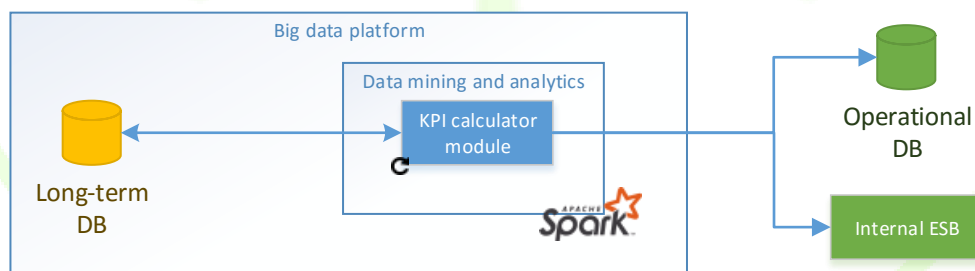


Figure 31 – Structure of the KPI calculator modules

The main foreseen technique to be used for the calculation of the KPIs is *Map/Reduce*.

Map/Reduce

As also explained in section 3.3.1, 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 perform 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 within the same building) 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 WiseCORP UI to provide insights in the behavior of the building in energy terms and assist in the decision support.

4.4 OPTIMIZATION AND ANALYTICS

4.4.1 Demand and Production forecast services

The Demand and Production forecast is a common module of WiseCOOP and WiseCORP (also is used in WG Cockpit) so the required information can be found in section 3.3.2.

4.4.2 Comfort-based demand flexibility forecast module (DR framework)

This section outlines the software component that will be responsible for forecasting the available short-term demand flexibility based on information about the context and building occupant comfort preferences so as to facilitate the implementation of human-centric demand response campaigns.

The high-level conceptual architecture of the flexibility forecast component and the different modules it comprises are presented in Figure 32.

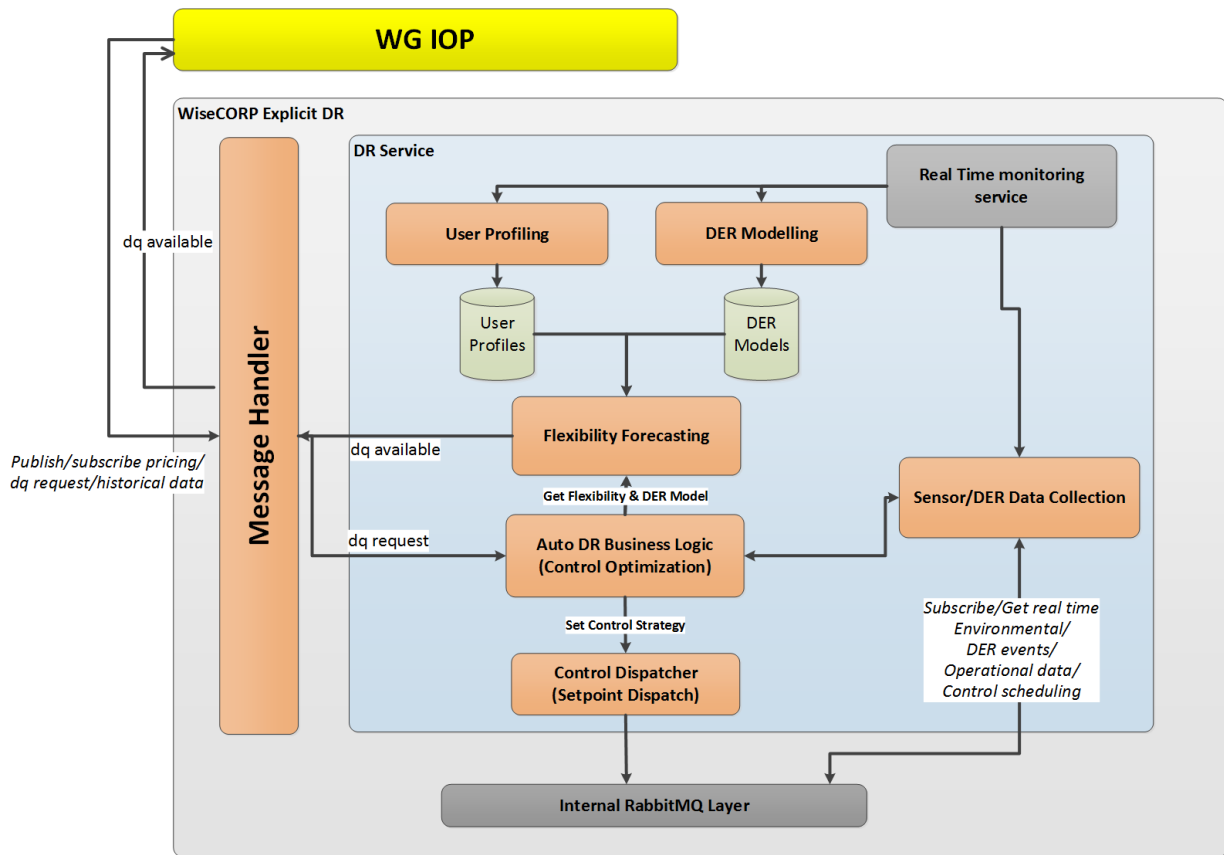


Figure 32 – Context-Aware Engine Conceptual Architecture

- **User profiling:** this module is responsible for creating the comfort/ discomfort profile of the relevant building occupants. These profiles essentially encapsulate the information regarding the available “flexibility” in terms of indoor environmental conditions (e.g. temperature, lighting) that is acceptable by the user and which can be used to change the energy demand profile of the building.
- **DER Modelling:** this module essentially links the variation in terms of environmental conditions with the energy demand profile of the respective building device (e.g. the link between HVAC energy consumption and indoor temperature). It is evident that each device/DER system must be modelled individually and typically is associated to specific elements of the indoor conditions.
- **Flexibility forecasting:** given the tolerances of the building occupant regarding changes in indoor conditions and the impact they can make on the energy consumption at the building level, the flexibility forecasting module estimates the available energy demand flexibility that can be offered to an aggregator.
- **AutoDR Business Logic:** this component is responsible for delivering a requested amount of demand flexibility under the optimal circumstances. The requested amount of flexibility may be smaller than the maximum building-level flexibility, so this module decides on the prioritization between devices and their setpoints that will be engaged in delivering flexibility. This selection entails an optimization in order to optimally satisfy criteria, such as user comfort, energy cost, etc. as well as constraints such as operational limitations/boundaries of the various devices.
- **Control Dispatcher:** the dispatcher essentially translates the decision of the previous module into specific setpoints per device that will be communicated via the asset dispatcher WiseCORP component to the physical devices.

A detailed description of all the components involved in facilitating the participation of WiseCORP in demand

response campaigns can be found in deliverable D10.2 “Demand Response Framework Specifications”. This document provides elaborate explanations of all involved models, software modules as well as specifications for information exchange among the relevant products, tools and modules.

4.4.3 Energy usage optimizer

The Facility Energy Optimizer is the module within WiseCORP that is in charge of periodically calculating the optimum setpoints for the set of controllable assets. In order to accomplish this task, different inputs are required, namely:

- Cost function for the optimization: curve of unitary costs associated to the energy supplied. These costs can be associated to economic or environmental impact terms, or a weighted combination of both
- Models for the controllable assets: the module works with generic device models which can be instantiated accordingly to the characteristics of the different assets. In the context of WiseCORP, focus is put on modelling storage devices and HVACs, but the model could still be applied to other kind of assets such as electric vehicles
- Initial status of the controllable assets: this requirement mainly applies to storage devices. The module needs to know the initial status (i.e. the state of charge) in order to correctly apply the constraints on the calculation of the optimum schedule (e.g. the amount of energy that can be supplied or taken by the storage device at a given time)
- Energy requirements per controllable loads: for each controllable load, the required energy to perform its task (divided in fixed and variable demand) needs to be defined

The output of the module is the optimum distribution of the energy to be consumed by each one of the controllable assets during the following considered time horizon (next 24 hours). The schedule is optimum with regards to the cost function: it assures that the total cost of the supplied energy, as defined by the cost function, is the minimum achievable.

4.4.3.1 General approach

The optimization in the energy field consists in the minimization of the energy usage and other variables that depends on it, like cost or environmental impact according to the source of the consumed energy.

It is possible to optimize the energy taking basic decisions as to a suitable temperature in air conditioning systems but for a general energy optimization is needed one or two of these actions:

- Shift the energy usage when possible.
- Store energy for later use.

The figure below shows the general process of energy optimization.

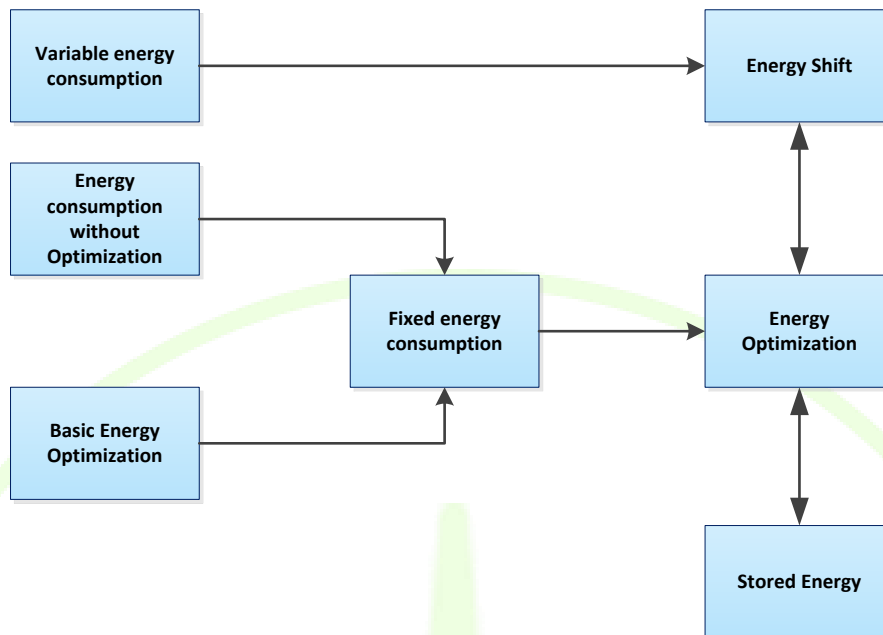


Figure 33 – General Optimization Process

- There is energy consumption that cannot be optimized due to many reasons like lack of information, control, etc. This energy contributes to the fixed energy consumption.
- There is energy consumption that can be optimized with basic techniques like HVAC settings.
- There is variable energy consumption like washing machine that is suitable for energy shift.
- The ability to store energy can be useful to shift energy usage to optimum periods.

In ANNEX 9.2 Energy usage optimizer equations are shown the equations needed to understand this module.

4.4.3.2 Theoretical Foundations

A mathematical optimization problem, or just optimization problem, has the form:

$$\text{Minimize } f_0(x)$$

$$\text{Subject to } f_i(x) \leq b_i, i = 1, \dots, m.$$

Here the vector $x = (x_1, \dots, x_n)$ is the optimization variable of the problem, the function $f_0: \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function, the functions $f_i: \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$, are the (inequality) constraint functions, and the constants b_1, \dots, b_m are the limits, or bounds, for the constraints. A vector x^* is called optimal, or a solution of the problem, if it has the smallest objective value among all vectors that satisfy the constraints: for any z with $f_1(z) \leq b_1, \dots, f_m(z) \leq b_m$, we have

$$f_0(z) \geq f_0(x^*).$$

We generally consider families or classes of optimization problems, characterized by particular forms of the objective and constraint functions. As an important example, the optimization problem is called a linear program if the objective and constraint functions f_0, \dots, f_m are linear, i.e., satisfy

$$f_i(\alpha x + \beta y) = \alpha f_i(x) + \beta f_i(y)$$

for all $x, y \in \mathbb{R}^n$ and all $\alpha, \beta \in \mathbb{R}$. If the optimization problem is not linear, it is called a nonlinear program.

4.4.3.3 Energy Optimization

In the field of energy the optimization consists in the determination of the minimum cost of energy use during a time interval. So the optimization function is:

$$f = \int_{Time\ Interval} f_t(t) \cdot dt$$

The functions f_t depends of several variables and parameters. In an HVAC system the energy behaviour depends on the external temperature, the settings of the internal temperature, the humidity, the buildings characteristics, the solar radiation, etc. This dependencies are no linear and in many cases too complex.

For the simplification of the problem, it is possible to suppose that during a little time step the behaviour of the system is constant and the evolution occurs between time steps. This permits a discretization of the problem that can be expressed as:

$$f = \sum_t^{Time\ Interval} f_t$$

The functions f_t that defines the optimum value depends on several variables like the input energy, the stores energy, the output energy, etc. These variables have dependences between them in the same time step and in different time steps. Actions on the energy process affect the current and the future status. So the energy optimization must be determined for the full time interval. The dependences between variables in distinct time steps are expressed by difference equations systems that are embedded in the constraints.

The time interval for the optimization or the optimization time horizon is a basic aspect of the optimization process. The time horizon must be fixed in a way that does not distort the optimization process. It should especially be noted that the time horizon covers a periodic time interval like one day or one week.

It is also possible to assume that the system behaviour is linear as can be seen in the definition of each sub-system.

4.4.3.4 Interaction with Control Systems

Optimization requires receiving information from the control system and to act on controlled items.

Optimization requires as input the information indicated in the following table:

Table 12 – Information required for optimization

Forecast of the available energy of each time step of the optimization time interval
Forecast of the energy cost (€/Kwh, CO ₂ Tons/Kwh etc.) for each time step of the optimization time interval.
Forecast of the Fixed consumption for each time step of the optimization time interval.
Data that characterizes the energy behaviour of each device: Input power, output power, storage capacity, initial stored energy, energy performance, etc.
Forecast of the variable energy required for each device.

The optimization process generates as a result the information indicated in the following table.

Table 13 – Information generated by the optimization

Total cost of the energy.
Supplied energy for each device and for time step of the optimization time interval.
Input energy for each device and for time step of the optimization time interval.
Stored energy for each device and for time step of the optimization time interval.

Variable Consumption for each device and for time step of the optimization time interval.

The control system must have the functions indicated in the following table.

Table 14 – Functions of the Control System

Provide the information specified in Table 12.
Receive the information in Table 13.
Select the factors to be considered in the optimization and the parameters that characterize it. In this case to optimize the monetary cost must provide the energy cost for each time step. For example, to optimize CO ₂ emissions the system must provide CO ₂ emissions per kWh and time step.
To cover the Fixed consumption, the control system must be able to provide energy to each device. In general, this is accomplished when the device is connected to the electricity network and operates as intended.
For energy storage the control system must regulate the input energy to the accumulator.
To extract energy from an accumulator, the control system must control the timing of energy supply.
To regulate the variable consumption, the system must decide at every time step when the device consumes and when does not.

4.4.3.5 Specific devices optimization

4.4.3.5.1 General Energy Storage

Introduction

This is a general device model which includes fixed and variable consumption, direct energy supply and energy storage. Fixed consumption is associated to a scheduled consumption without possibility of rescheduling. Variable consumption is associated to an amount of required energy which consumption can be shifted with optimization purposes.

Model description

The figure below shows a scheme of the model.

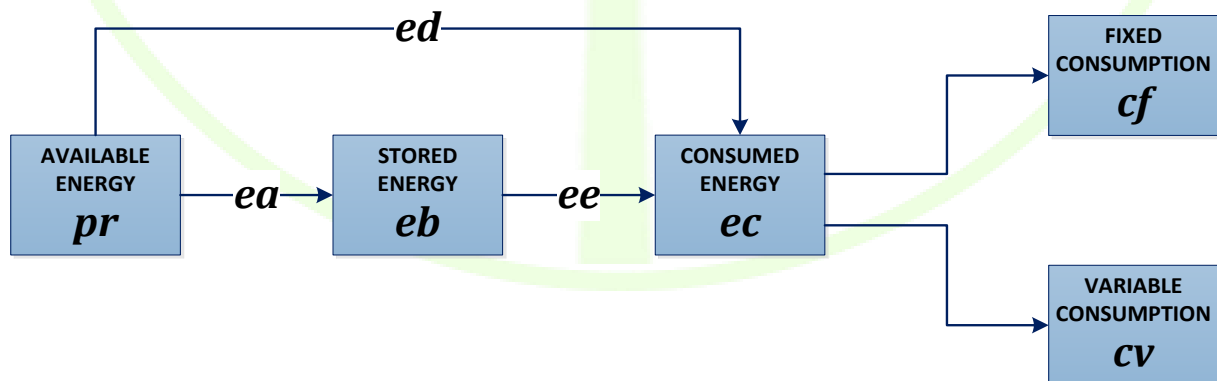


Figure 34 – Energy model

There are an Available energy (pr) that can be supplied directly to the system (ed) or stored (eb). The Stored energy can be supplied in any moment (ee). The consumed energy (ec) covers the fixed consumption and the variable consumption.

There are several limits for the energy: Maximum energy available, maximum input energy, maximum output energy, maximum storage capacity, etc.

Table 15 – General Energy Storage model definitions

cf_i	Fixed Consumption
c_i	Unitary cost of energy for time step i
cv_i	Variable consumption for time step i
dt_i	Time step duration.
Ea_i	Stored energy in time step i
$eamax$	Maximum storage capacity.
Eb_i	Stored energy in time step i .
ec_i	Consumed energy in time step i : $ec_i = ed_i + ee_i$
ed_i	Direct supplied energy in time step i
ee_i	Extracted energy in time step i
eme_i	Maximum extracted energy in time step i .
$emma_i$	Maximum stored energy in in time step i due the limitations on input energy and available energy.
$Ereq$	Required Consumption.
F	Optimization function.
N	Number of time steps.
Pma	Maximum input power.
Pme	Maximum output power.
Pr_i	Available energy in time step i .
rnd	Energetic performance [0,1].

4.4.3.5.2 HVAC

Model description

This model defines the behaviour of HVAC systems with several devices. These devices can be air conditioners, water heaters or coolers. The devices have an energy accumulator for self-use.

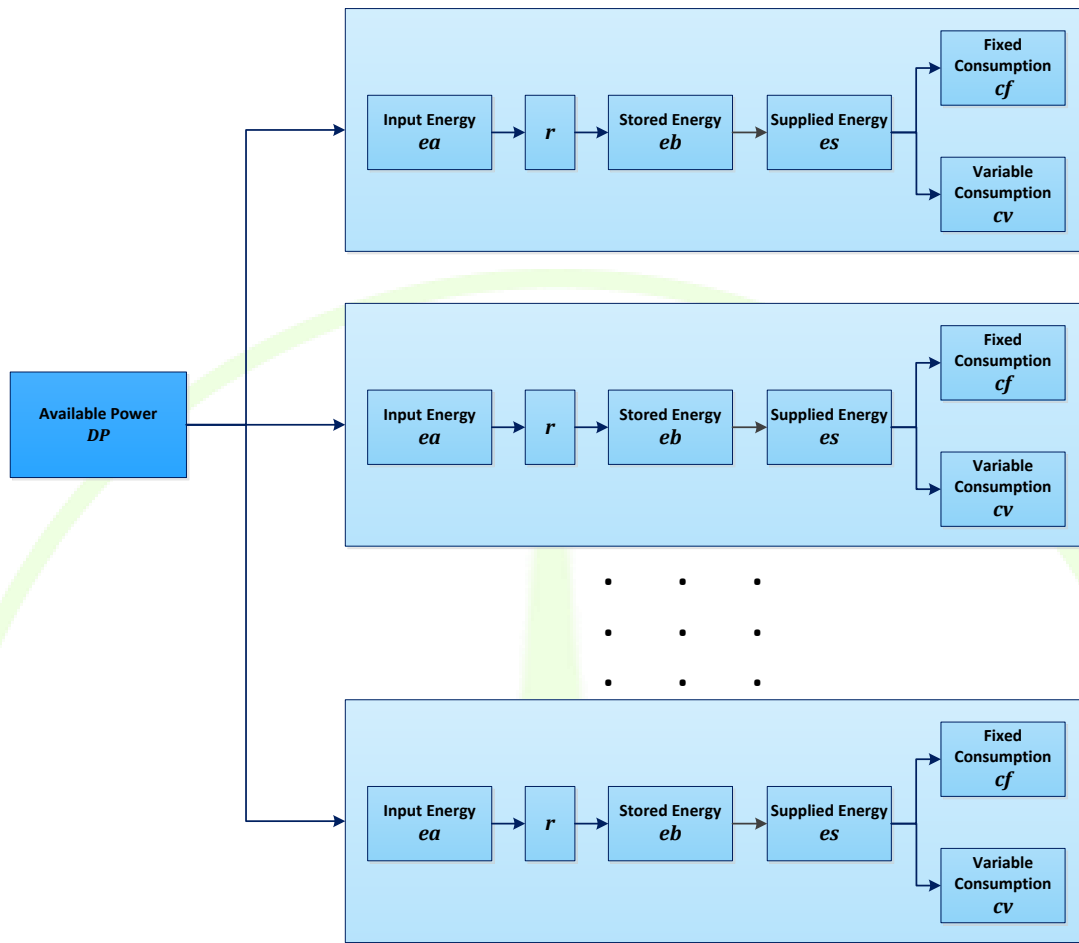


Figure 35 – HVAC Model

Table 16 – HVAC Model definitions

c_j	Energy cost in time step j .
cf_i	Fixed consumption of device i in a time step.
Ct_{ij}	Total consumption (fixed+variable) of device i in time step j .
cv_{ij}	Variable consumption of device i in time step j .
ea_{ij}	Input energy of device i in time step j .
eat_j	Input energy in time step j for all devices.
$eb_{si,j}$	Stored energy of device i in time step j .
$eb0_i$	Initial stored energy of device i .
$ebmax_i$	Maximum Stored energy of device i
ebt_j	Stored energy in time step j for all devices.
Ed_j	Available energy for all devices in time step j
ee_i	Maximum Input energy due to the input power limitation and the available energy.
Eep_i	Maximum Input energy due to the input power limitation.
Ees_i	Maximum supplied energy due to the maximum supplied power.

Es_{ij}	Supplied energy of device i in time step j .
$esot_j$	Excess energy in time step j for all devices.
Est_j	Supplied energy in time step j for all devices.
I	Device index.
J	Time step index.
K	Variable type index: $ea: k=1, eb: k=2, r: k=3$.
Ni	Number of devices.
Nj	Number of time steps.
Nk	Number of variable types. $Nk=3$.
Pd_j	Available power for all devices in time step j
pe_i	Maximum Input power of device i .
pfi	Fixed power of device i .
psi	Maximum supplied power of device i .
r_i	Performance of device i .

4.4.3.6 Global Optimization

Approach

A global optimization of all energy management devices is performed. The process optimizes all devices at once, establishing what every device should be done. Optimization is evaluated assuming a determined time horizon and lasting configurable intervals.

The model considers the following:

- The energy available at each time step.
- The energy storage devices commonly used (batteries in the electric case).
- Electric Vehicles.
- Devices with fixed energy.
- Devices with varying energy consumption.
- HVAC devices with energy storage for own use including heaters and refrigerators.
- The costs associated with energy use.

The energy behaviour of these elements is defined by a model. The model is defined by a set of variables.

Model description

The model is shown in the following figure.

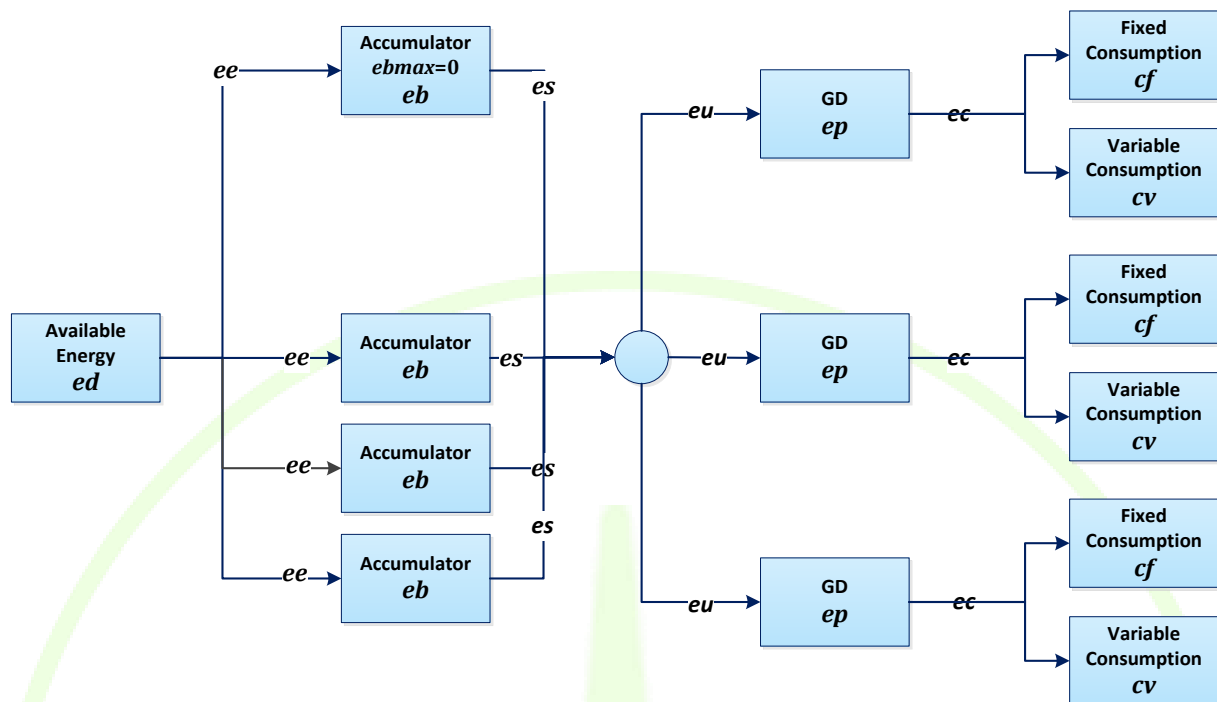


Figure 36 – Global optimization Model

There are energy storage devices capable of supplying this energy to the remaining systems. The direct supply of energy is modelled considering a battery with null capacity of accumulation.

The other devices are generic and according to their characteristics can be energy consumer devices (fixed and variable), electric vehicles and thermal equipment (HVAC, heaters, coolers, refrigerators) that store energy for their own use.

A fixed consumer follows a specific time schedule. A variable consumer may demand energy at any time within the time horizon.

The variables that define the model are indicated in the following table.

Table 17 – Global Optimization model variables

Variable	Description	Comments
eb	Stored energy	Energy stored in the generic accumulator.
Ec	Consumed energy	Energy consumed by a generic device. In the case of the electric vehicle it corresponds to the energy stored in the battery.
Ee	Input energy	Input energy to the system. It may be supplied directly or used to store on a generic accumulator.
Ep	Self-stored energy	Energy stored in a generic device for use by the device itself.
Es	Supplied energy	Energy supplied directly or throw the generic accumulator.
Eu	Used energy	Used energy by a generic device.

The other model elements are part of the restrictions and are not part of the variables that must be optimized.

The following table shows the definition of the variables used in the model.

Table 18 – Global Optimization definitions

Concept	Definition
c_j	Cost of energy during time step j .
$69si_{i,j}$	Device i fixed consumption during time step j .
cr_i	Required consumption for device i (variable consumption). For the electrical vehicle is the energy that must be stored in its battery.
$Ct_{i,j}$	Device i total consumption during time step j : $ct_{i,j} = cf_{i,j} + cv_{i,j}$
ctd_i	Total consumption of device i for all the time steps (fixed consumption + required consumption)
ctt	Total consumption: $ctt = \sum_{i=1}^{n_i} \sum_{j=1}^{n_j} ct_{i,j}$.
$Cv_{i,j}$	Device i variable consumption during time step j .
dt	Duration of the time step.
$69si_{i,j}$	Stored energy for device i during time step j .
$ebmax_i$	Maximum stored energy of device i .
$69si_{i,j}$	Consumed energy for device i during time step j .
ed_j	Available energy during time step j .
$ee_{i,j}$	Input energy for device i during time step j .
emc_i	Maximum consumption energy of device i due to output power limit.
Emp_i	Maximum input energy of device i due to input power limit.
Ems_i	Maximum output energy device i due to output power limit.
Emu_i	Maximum input energy for self-use of device i due to input power limit.
$Ep_{i,j}$	Self-stored energy for device i during time step j .
$epmax_i$	Maximum self-stored energy of device i .
$es_{i,j}$	Supplied energy for device i during time step j .
$eu_{i,j}$	Used energy for device i during time step j .
i	Device index.
J	Time step index.
Ni	Number of devices.
Nj	Number of time steps.
P_j	Available power during time step j .
pc_i	Maximum consumption power in device i .
pe_i	Maximum input power in device i .
pf_i	Fixed consumption in device i .
ps_i	Maximum output power in device i .
pu_i	Maximum input power for self-use in device i .
r_i	Performance of the accumulator in device i .
$tfin_i$	Final time step of availability of device i
$tini_i$	Initial time step of availability of device i

4.4.3.7 Optimization Algorithms

The optimization models described above are linear model so the optimization algorithms can be based on linear programming techniques.

Traditional optimization techniques of linear systems have been based on the simplex method developed in the 1940s. In the 1980s it was discovered that many of the large linear programs can be solved efficiently by formulating and nonlinear problems and solving them with various modifications nonlinear algorithms such as Newton's method. A characteristic of these methods was that requiring that all iterates to satisfy the inequality constraints on the problem strictly, so soon it became known as interior point methods. In early 1990, a class-primal-dual methods had distinguished themselves as the most efficient and practical approach proved a strong contender for the simplex method in big problems.

These techniques can easily solve problems with hundreds of variables and thousands of constraints on a small desktop computer, in a matter of seconds.

4.4.3.8 Example: Office Building

4.4.3.8.1 Description

This example optimizes the consumption of a building with high energy consumption. Facilities include HVAC with energy storage, hot water supply, Lighting, variable and fixed consumption. The building has electrical storage capacity for general use.

4.4.3.8.2 Process Summary

The summary of the energy management is:

Available Energy:	13670.00 (Kwh)
Input Energy:	1577.87 (Kwh)
Supplied Energy:	1572.82 (Kwh)
Used Energy:	1572.82 (Kwh)
Consumed Energy:	1562.40 (Kwh)
Total fixed consumption:	1262.40 (Kwh)
Total required consumption:	300.00 (Kwh)
Total consumption:	1562.40 (Kwh)
Diff. Available-Input:	12092.13 (Kwh)
Diff. Consumed-Total consumption:	0.00 (Kwh)
Total Cost:	177.79 (€)
Mean Cost:	0.11 (€/Kwh)

4.4.3.8.3 Summary by time step

The figure below shows the time evolution for the energy process of all the devices.

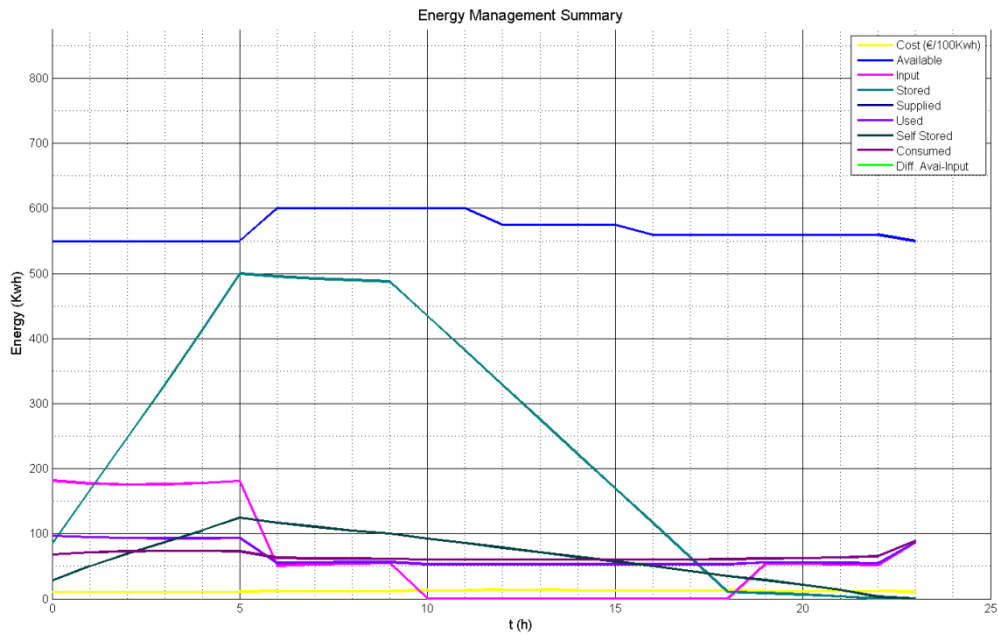


Figure 37 – Energy process Summary

The following information summarizes the energy process for all devices during each time step.

J	th	c	ed	eet	ebt	est	eut	ept	ect	cvt
1	0.00	0.11	550.00	181.58	84.01	96.72	96.72	27.30	68.17	15.57
2	1.00	0.11	550.00	177.14	165.82	94.51	94.51	49.55	71.01	18.41
3	2.00	0.11	550.00	175.62	247.27	93.35	93.35	68.90	72.75	20.15
4	3.00	0.11	550.00	175.93	329.47	92.90	92.90	87.02	73.53	20.93
5	4.00	0.11	550.00	177.71	413.38	92.95	92.95	105.16	73.57	20.97
6	5.00	0.11	550.00	181.09	500.00	93.59	93.59	124.69	72.81	20.21
7	6.00	0.12	600.00	50.73	495.50	55.23	55.23	116.88	62.91	10.31
8	7.00	0.12	600.00	52.38	492.16	55.73	55.73	110.41	62.04	9.44
9	8.00	0.12	600.00	53.56	489.67	56.04	56.04	104.73	61.55	8.95
10	9.00	0.12	600.00	54.47	487.63	56.51	56.51	99.83	61.22	8.62
11	10.00	0.13	600.00	0.00	434.59	53.04	53.04	92.56	60.25	7.65
12	11.00	0.13	600.00	0.00	381.56	53.03	53.03	85.44	60.10	7.50
13	12.00	0.14	575.00	0.00	328.55	53.01	53.01	78.38	60.02	7.42
14	13.00	0.14	575.00	0.00	275.54	53.01	53.01	71.35	59.99	7.39
15	14.00	0.13	575.00	0.00	222.52	53.02	53.02	64.30	60.01	7.41
16	15.00	0.13	575.00	0.00	169.51	53.01	53.01	57.17	60.08	7.48
17	16.00	0.13	560.00	0.00	116.51	53.00	53.00	49.91	60.21	7.61
18	17.00	0.13	560.00	0.00	63.52	52.99	52.99	42.44	60.40	7.80
19	18.00	0.13	560.00	0.00	10.53	52.99	52.99	34.69	60.68	8.08
20	19.00	0.12	560.00	53.99	8.65	55.88	55.88	28.57	61.84	9.24
21	20.00	0.12	560.00	53.33	6.46	55.52	55.52	21.55	62.39	9.79
22	21.00	0.12	560.00	52.40	3.69	55.18	55.18	13.36	63.23	10.63
23	22.00	0.12	560.00	51.05	0.00	54.74	54.74	2.93	65.07	12.47
24	23.00	0.11	550.00	86.88	0.00	86.88	86.88	0.00	88.56	35.96

4.4.3.8.4 Summary by device

The following information summarizes the energy process for each device during all time steps.

I	id	ee	eb	es	eu	ep	ec	cf	cv	cr
1	1000001	1072.82	0.00	1072.82	0.00	0.00	0.00	0.00	0.00	0.00
2	2000001	505.05	0.00	500.00	0.00	0.00	0.00	0.00	0.00	0.00
3	3000001	0.00	0.00	0.00	208.42	0.00	198.00	48.00	150.00	150.00
4	3000002	0.00	0.00	0.00	52.40	0.00	52.40	2.40	50.00	50.00
5	4000001	0.00	0.00	0.00	100.00	0.00	100.00	0.00	100.00	100.00
6	4000002	0.00	0.00	0.00	240.00	0.00	240.00	240.00	0.00	0.00
7	4000003	0.00	0.00	0.00	972.00	0.00	972.00	972.00	0.00	0.00

4.4.3.8.5 Full Process Information

The figure below shows the evolution of the input energy.

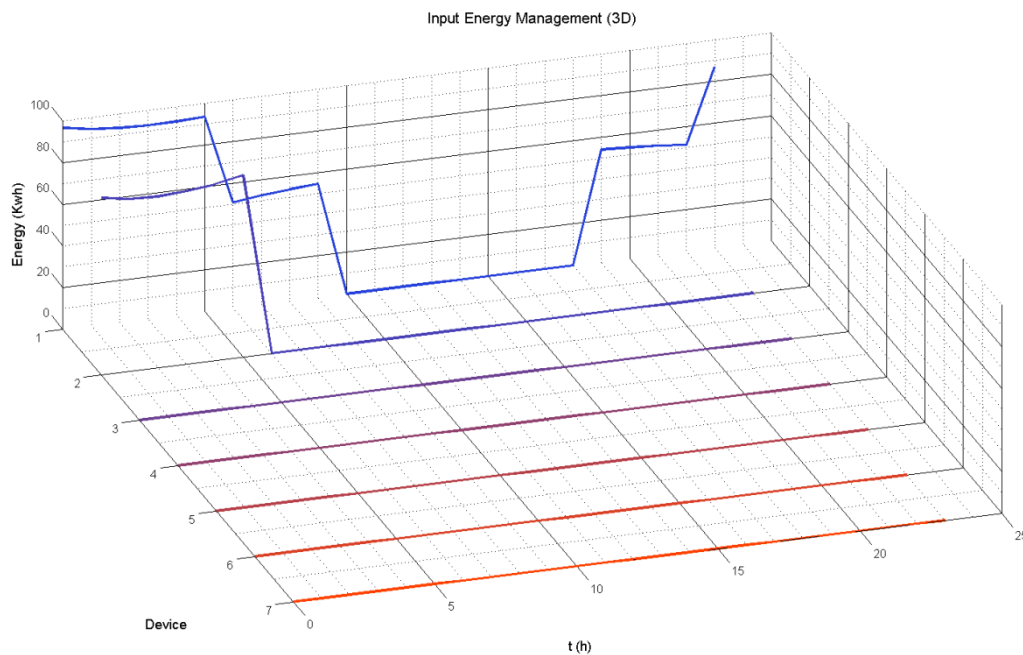


Figure 38 – Input energy

The figure below shows the stored energy for the devices.

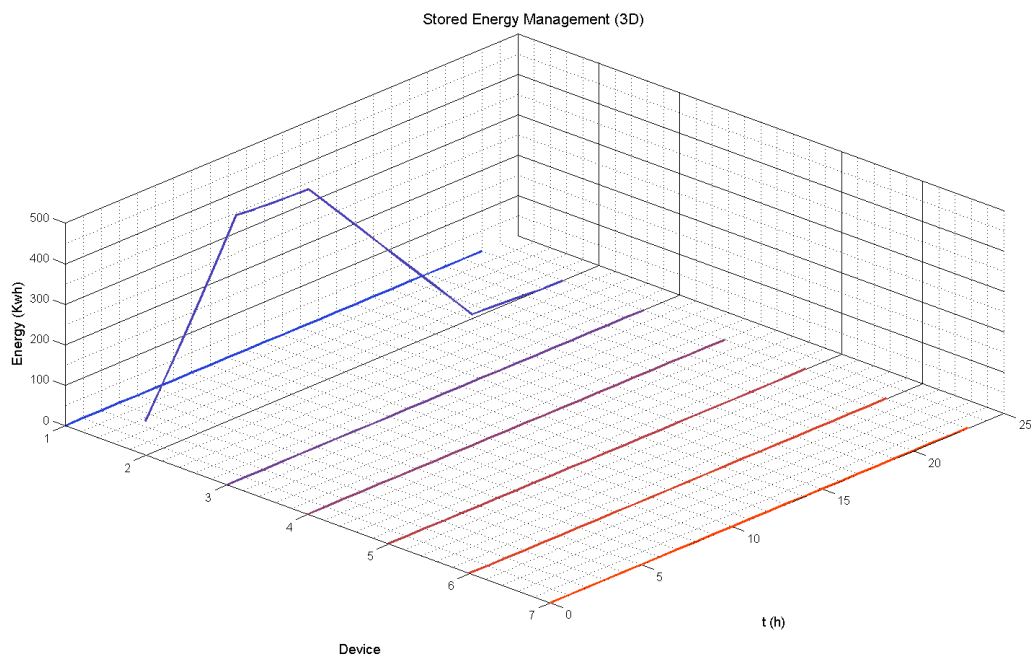


Figure 39 – Stored energy

The figure below shows the supplied energy for all the devices.

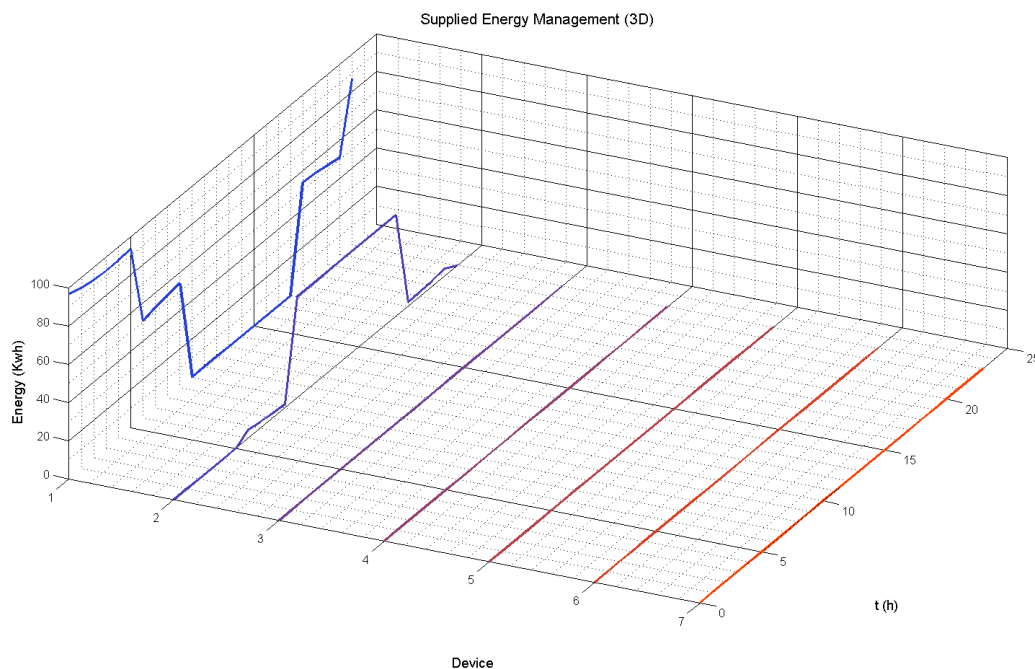


Figure 40 – Supplied energy

The figure below shows the used energy for the devices.

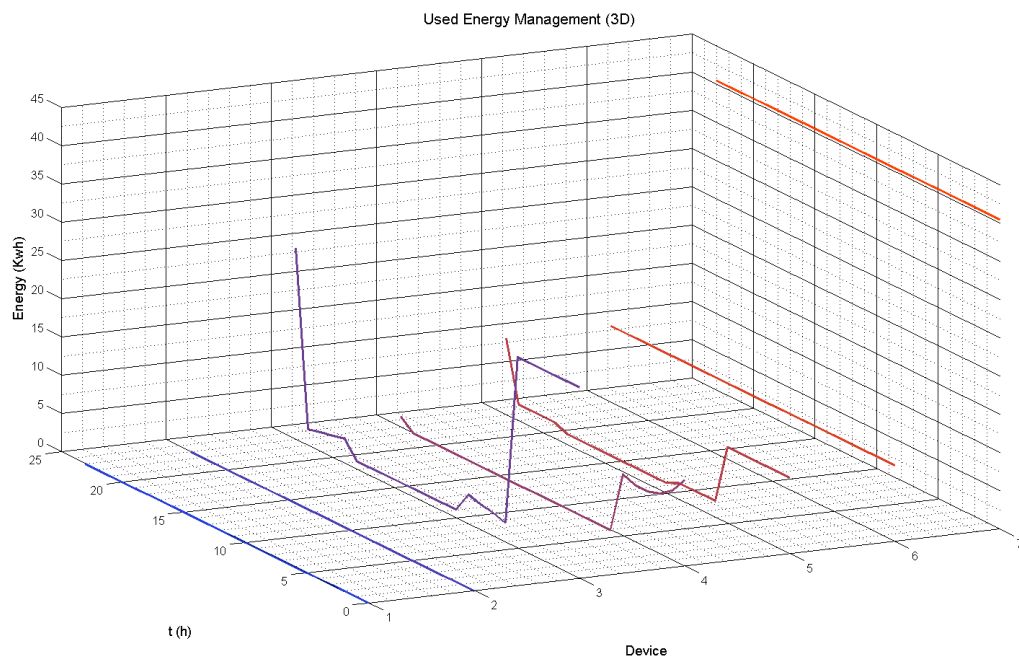


Figure 41 – Used energy

The figure below shows the self-stored energy for the devices.

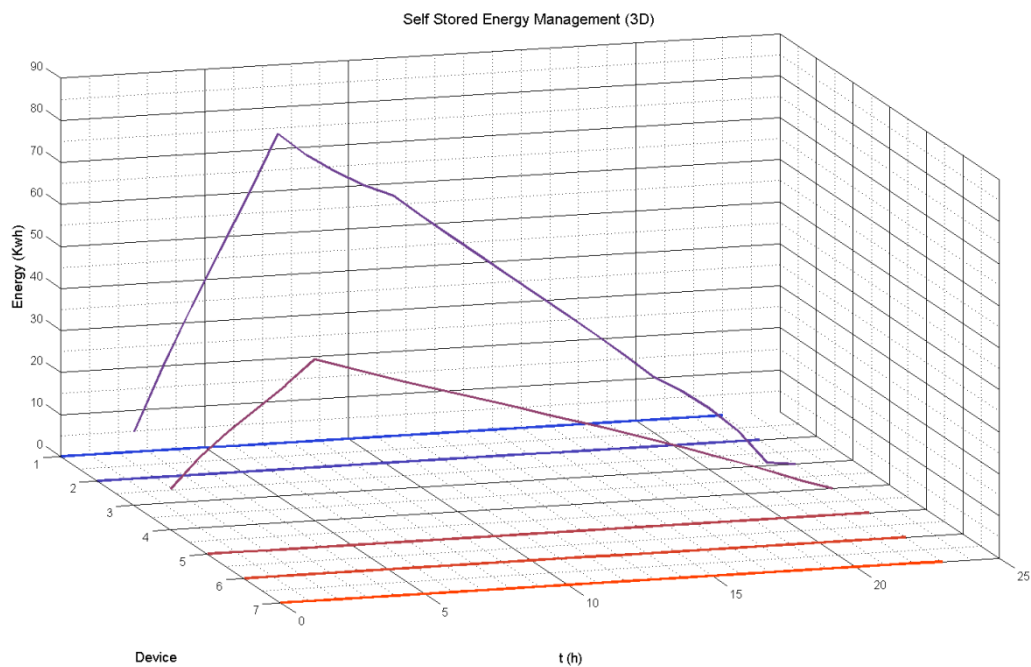


Figure 42 – Self-stored energy

The figure below shows the consumed energy for the devices.

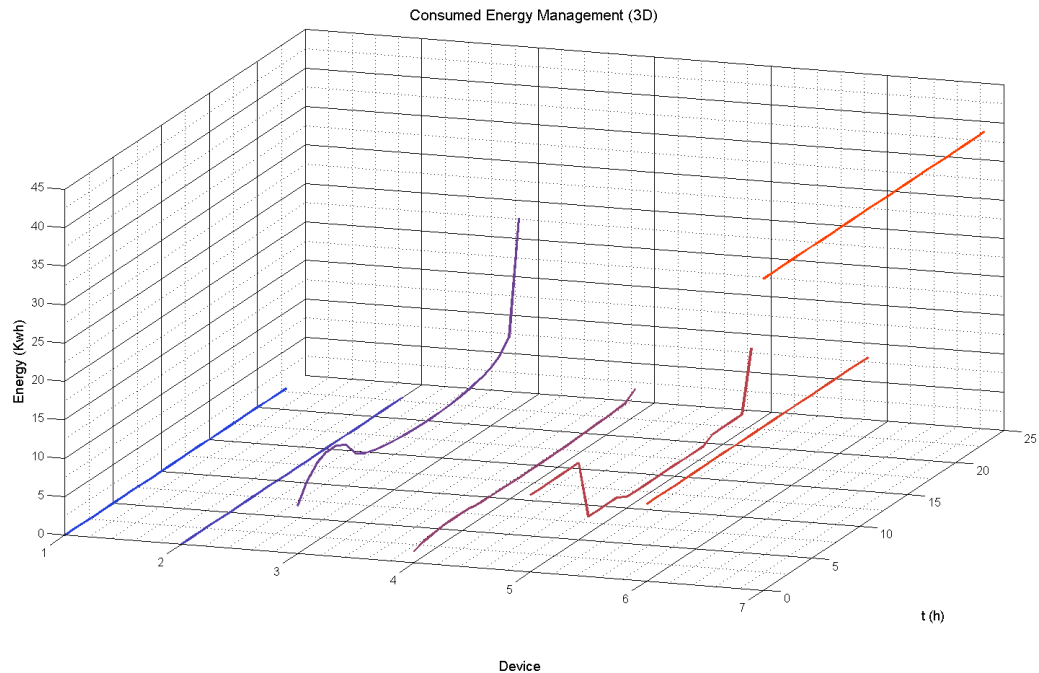


Figure 43 – Consumed energy

4.5 USER INTERFACE

WiseCORP target users are Facility managers. UI is designed under the assumption that a single facility manager may be able to monitor one or more buildings/facilities. Each one of those buildings may also include different areas (sub-metering).

4.5.1 Dashboard

The dashboard provides an overview of all monitored facilities, with a selection of KPIs (self-consumption ratio, economic savings, reduction of environmental impact, etc.)

A user can visualize several buildings under its control

Dashboard

Overview of all monitored buildings, with current KPIs (evaluated over last 30 days)

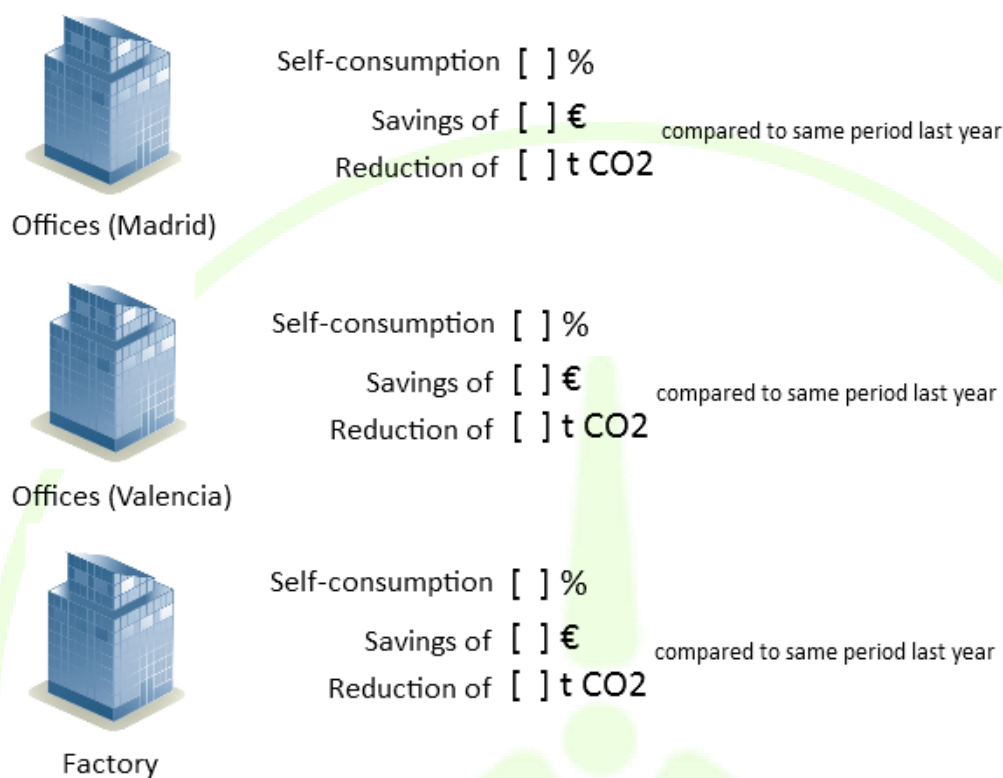


Figure 44 – Dashboard mock-up

A second section of the dashboard provides metrics of energy consumption/production per building/area, comparison of those with reference metrics (e.g. averaged values of past days), and indicators of availability of data (i.e. ratio of sensors that are successfully sending data, which gives an indication of the quality of the data)

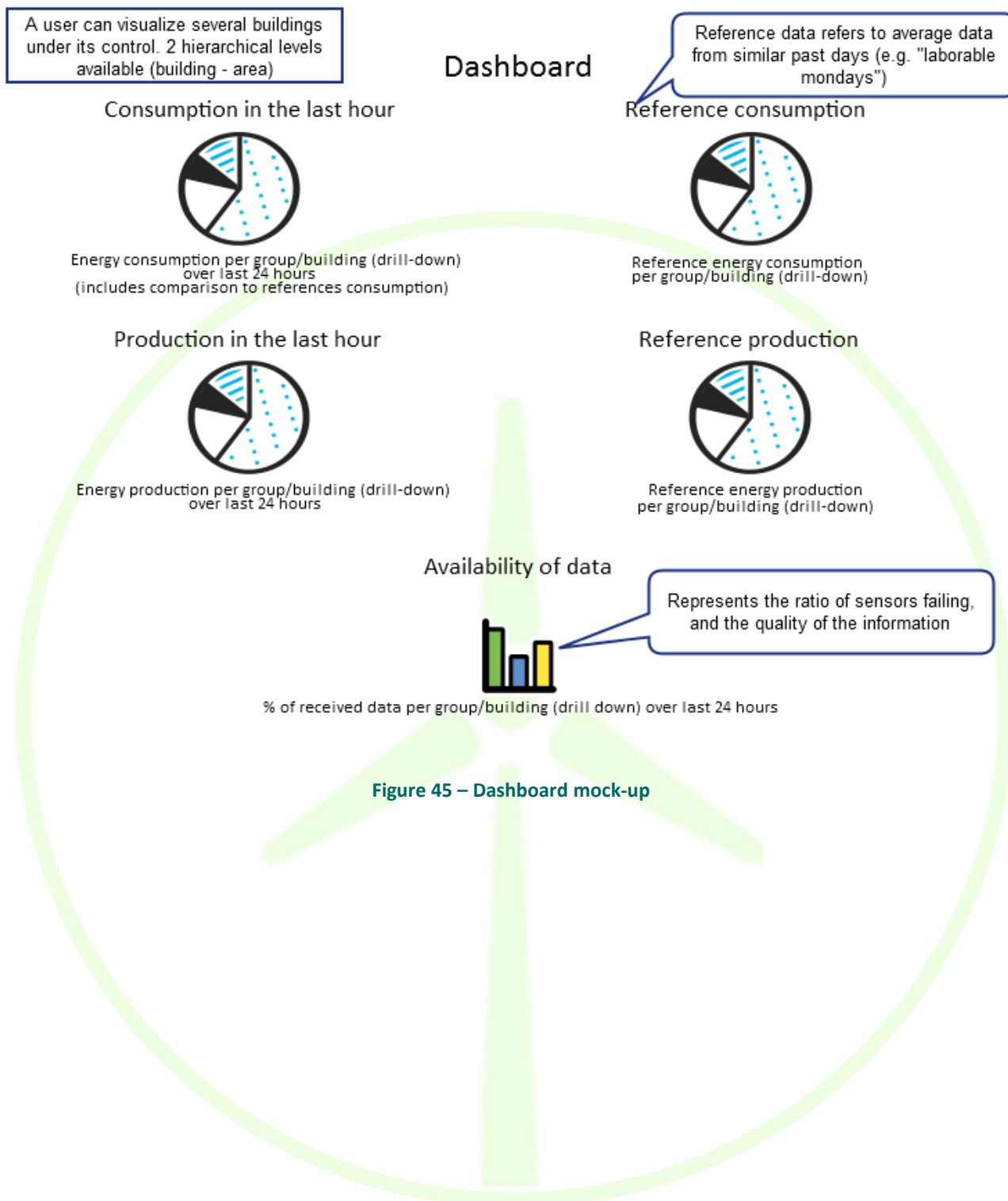


Figure 45 – Dashboard mock-up

4.5.2 Map

All monitored buildings are represented in a map. Basic information is shown upon selection.

Map

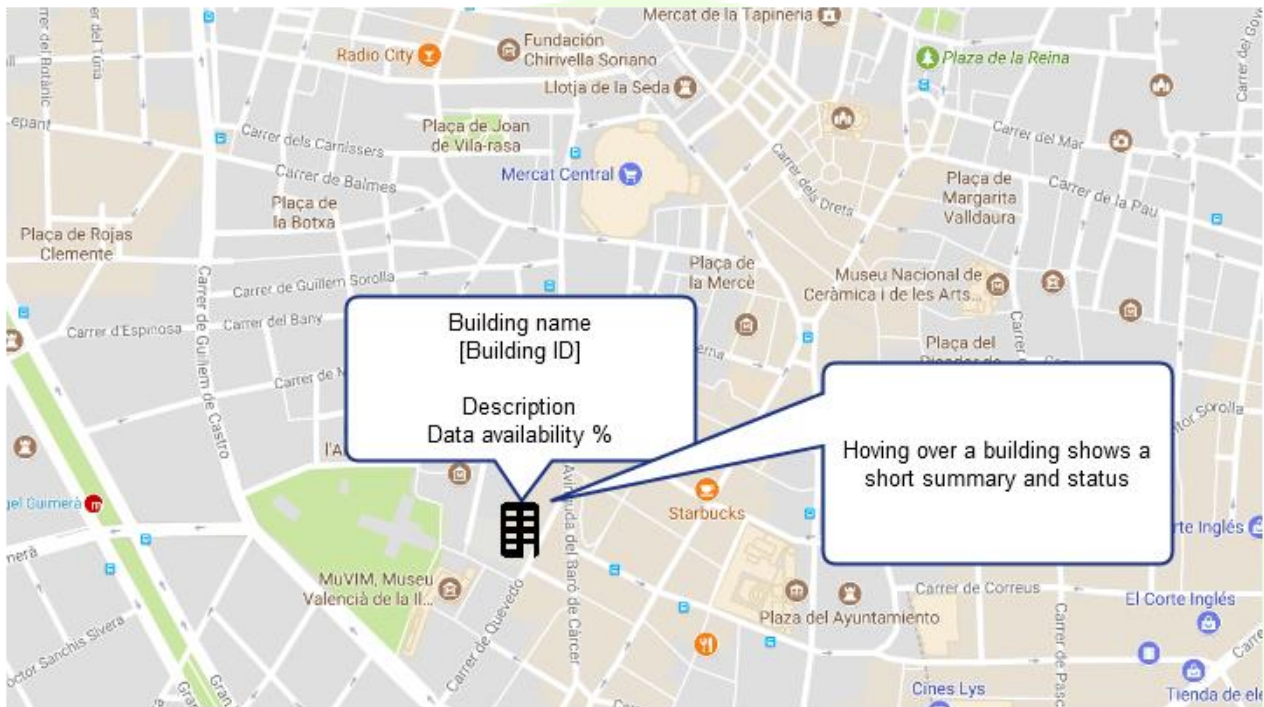


Figure 46 – Map mock-up

After clicking on a building, the different available sensors are depicted, and basic information about those is presented.

Map

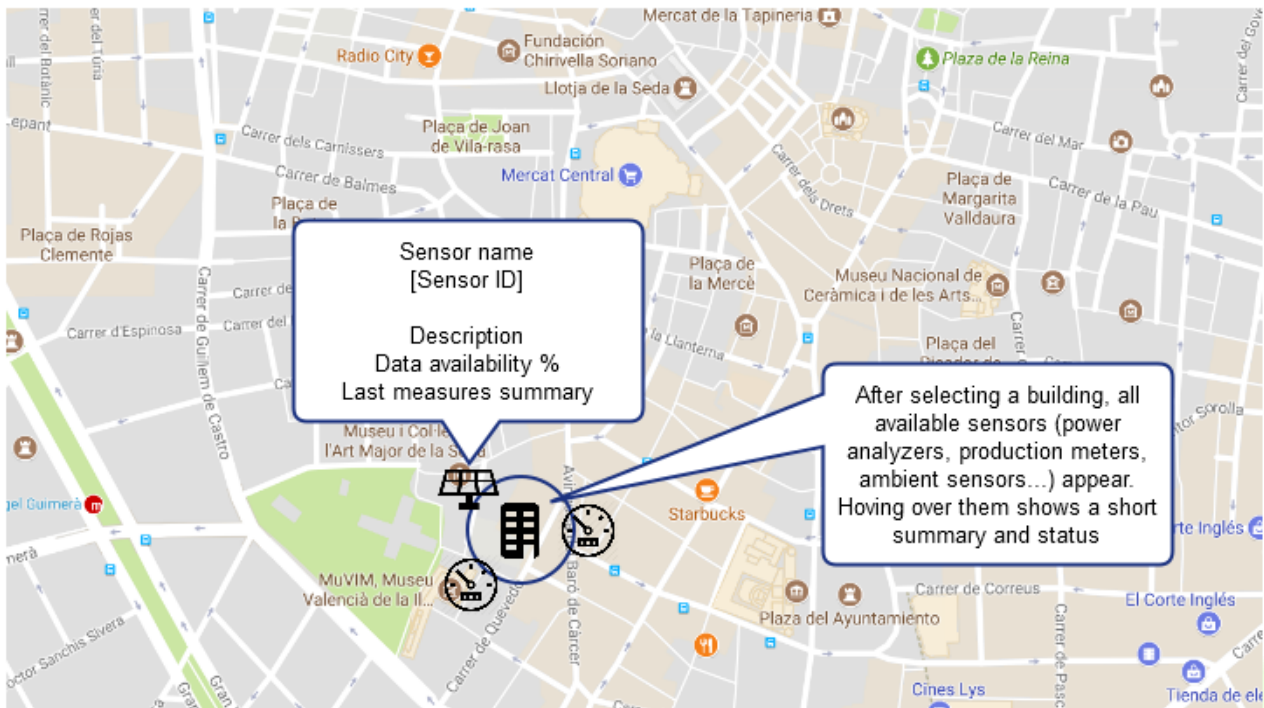


Figure 47 – Map mock-up

After clicking on one of the sensors, a dialog is shown allowing to visualize the data reported by that sensor over the last 24 hours or month.

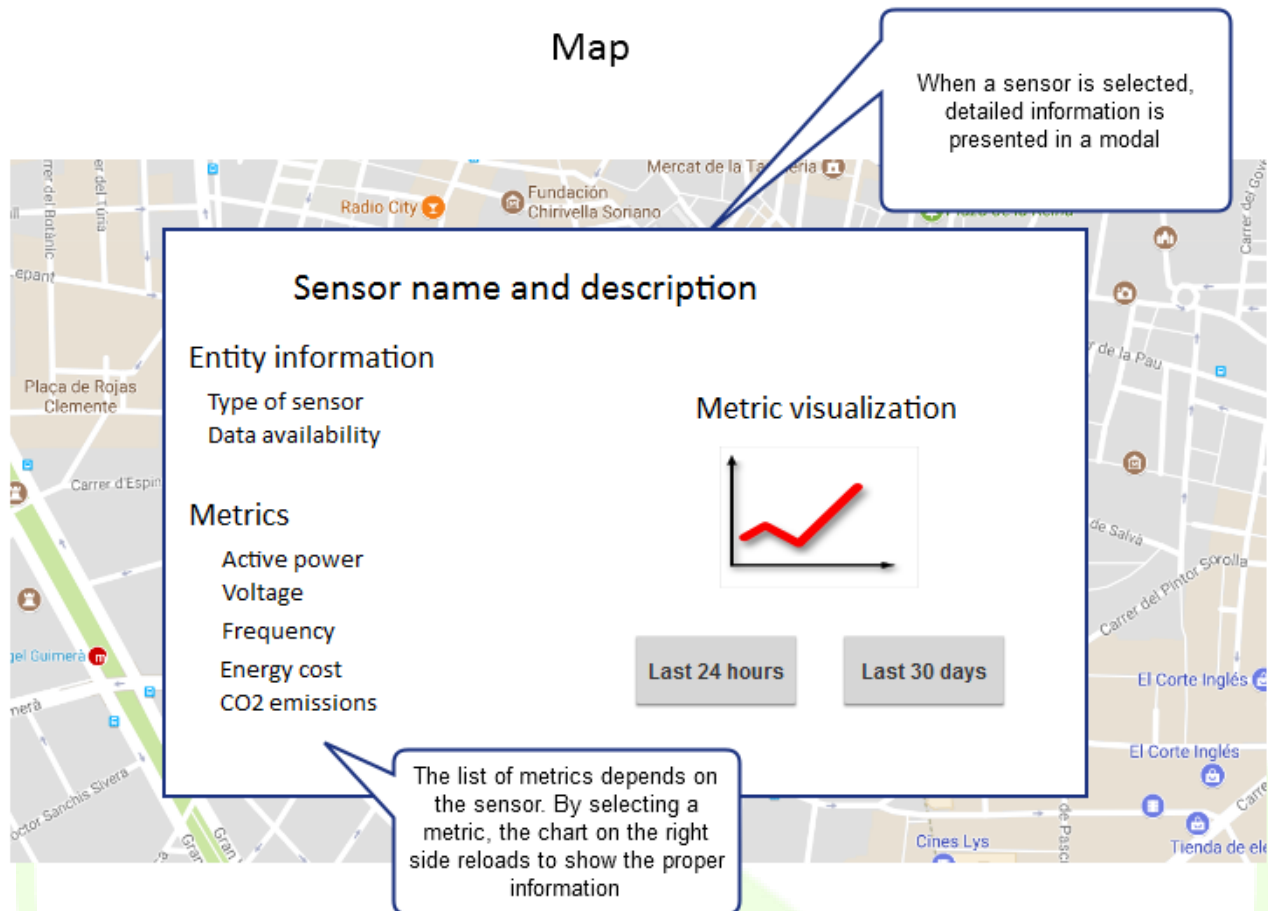


Figure 48 – Map mock-up

4.5.3 Building detail

4.5.3.1 Monthly indicators

This section provides details of the energy usage per building, including the following monthly indicators:

- Total demand
- Self-consumption ratio
- Economic cost
- Environmental impact
- Evolution compared with last year
- Energy source distribution
- Histogram of measured active power
- Distribution of demand under time of use tariff period slots

Building detail

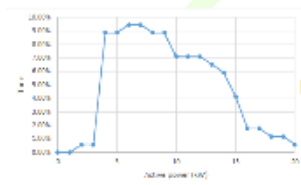
Search building...

Select month

Building name



Energy source
(self-consumed /
external - renewable sources /
external - exhaustible sources)



Power (x) vs %Time (y) vs Tariff Power Limits



Energy consumed per tariff period (peak/(super)off-peak)

Assisting facility manager
to assess feasibility of
(manual) reduction of peak
power and change in tariff

Include comparison to
previous similar periods
as well

All those metrics referred to a single
month



(picture placeholder)

Self-consumption

[] %

Energy cost

[] €

Emissions

[] t CO2

Savings of

[] €

Reduction of

[] t CO2

compared to same month last year

Supply Demand Indicators

[] kWh / m2

[] kWh / employee

[] kWh / working hour

Figure 49 – Building detail mock-up

4.5.3.2 Access to raw data

A second subsection provides access to raw data of the different sensors installed in the building.

Building detail - Monitoring

Settings

Assets

Select (list of) sensors

Date range

01/01/2017 - 02/02/2017

Metric

Select (list of) available metrics

Integration

15 minutes

Plot graph

Selected criteria gets plotted on the right hand side

Integration period of data

15 minutes
1 hour
1 day

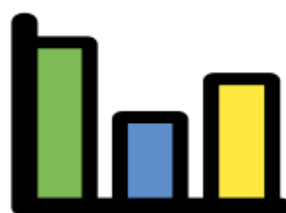


Figure 50 – Building detail – monitoring mock-up

4.5.3.3 Temporal distribution of demand

A third subsection provides a graphical overview of the temporal distribution of the demand.

Building detail - Demand distribution

Search building...

Select week

Color map of distribution of demand

	M	Tu	W	Th	F	Sa	Su
00:00 - 01:00							
01:00 - 02:00							
03:00 - 04:00							
04:00 - 05:00							
05:00 - 06:00							
06:00 - 07:00							
...							
23:00 - 24:00							



















Figure 51 – Building detail – Demand distribution

4.5.4 Demand overview

This section provides summary graphs with the demand of a building or group of buildings, allowing at a glance to get the relevant information about the distribution of the demand of the last 48 hours and the last 30 days.

Consumption overview

Select building or group... ▲

Entity	Energy consumption (kWh)		Energy cost (€)		CO2 emissions (t)	
	Last 48 hours	Last 30 days	Last 48 hours	Last 30 days	Last 48 hours	Last 30 days
Area 1	 Total: x kWh	 Total: x kWh	 Total: x €	 Total: x €	 Total: x t	 Total: x t
Area 2	 Total: x kWh	 Total: x kWh	 Total: x €	 Total: x €	 Total: x t	 Total: x t
Area 3	 Total: x kWh	 Total: x kWh	 Total: x €	 Total: x €	 Total: x t	 Total: x t

Each cell represents a summary of that period, showing a small chart and the total amount







Figure 52 – Consumption overview mock-up

4.5.5 Production overview

This section provides similar summary graphs with the production of a building or group of buildings, allowing at a glance to get the relevant information about the distribution of the demand of the last 48 hours and the last 30 days.

Production overview

Select building or group... ▲

Entity	Energy production (kWh)	
	Last 48 hours	Last 30 days
PV 1 Nom. power: x kW	 Total: x kWh	 Total: x kWh
PV 2 Nom. power: x kW	 Total: x kWh	 Total: x kWh
uWind 1 Nom. power: x kW	 Total: x kWh	 Total: x kWh

Each cell represents a summary of that period, showing a small chart and the total amount




Figure 53 – Production overview mock-up

4.5.6 Forecasting

This section provides similar summary graphs with the demand & production forecasts of a building or group of buildings.

Forecasting

Select building or group... ▲

	Energy consumption forecast(kWh)	Energy production forecast(kWh)
Entity	Next 48 hours	Next 48 hours
PV 1		 Total: x kWh
PV 2		 Total: x kWh
Area 1	 Total: x kWh	

Each cell represents a summary of that period, showing a small chart and the total amount

Figure 54 – Forecasting mock-up

4.5.6.1 Usage calendar

In order to allow the forecast algorithms to properly analyse the demand and production data, the usage calendar of the buildings needs to be given. This basically summarizes in providing a calendar with the working/non-working days for each building.

Usage calendar

Search building... ▲

Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi
					Dimanche

Building manager can introduce the days the building is closed, an input to the forecast algorithms

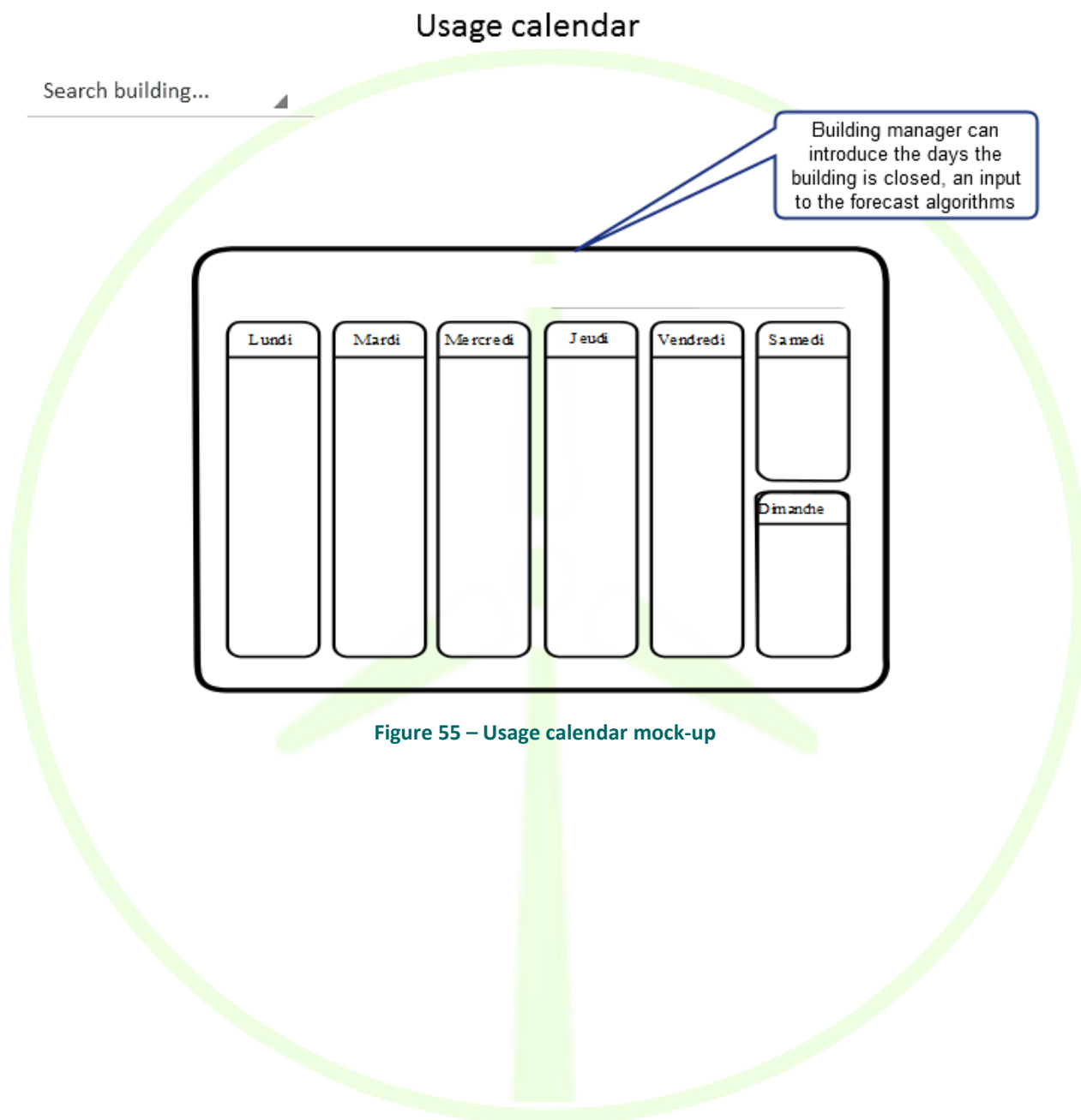


Figure 55 – Usage calendar mock-up

4.5.6.2 Forecast analysis

In order to evaluate the validity of the demand and production forecasts, this section will present the curves of the forecasted and actual measured values, allowing the evaluation of the accuracy of the forecasts.

Forecast analysis

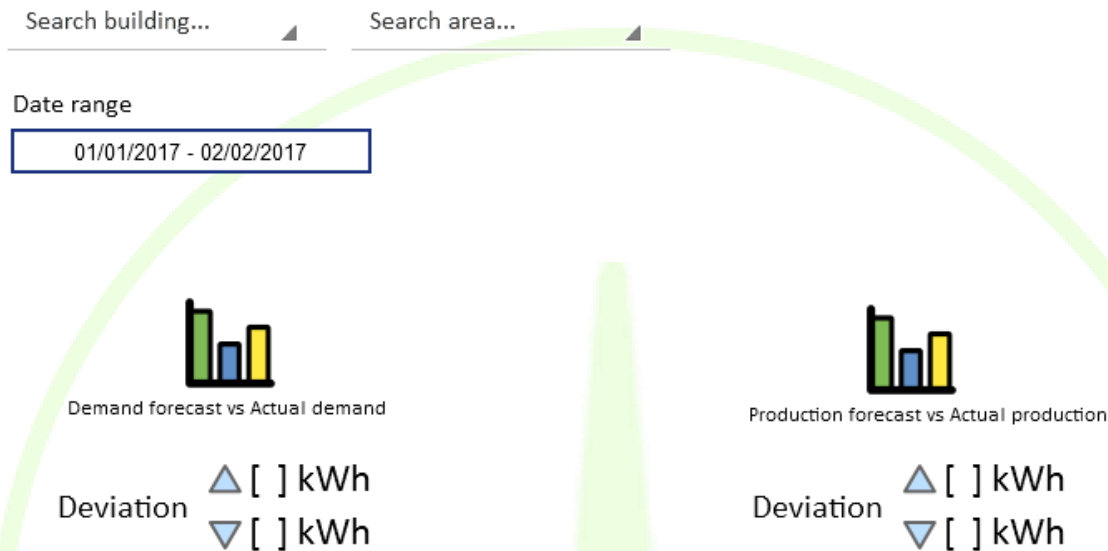


Figure 56 – Forecast analysis mock-up

4.5.7 Automation

This section visualizes the status of the controllable assets that are managed by WiseCORP (HVACs, lighting, batteries...). For each of them, a timeline is shown with the history of setpoints, and the scheduled setpoints. The actual scheduling is provided by the asset dispatcher module, and composed with the outputs of the *energy usage optimizer* module (which will calculate the day-ahead optimum schedule) and the Demand Response Optimization framework (which will calculate intra-day changes to deal with scheduled Demand-Response campaigns).

Automation

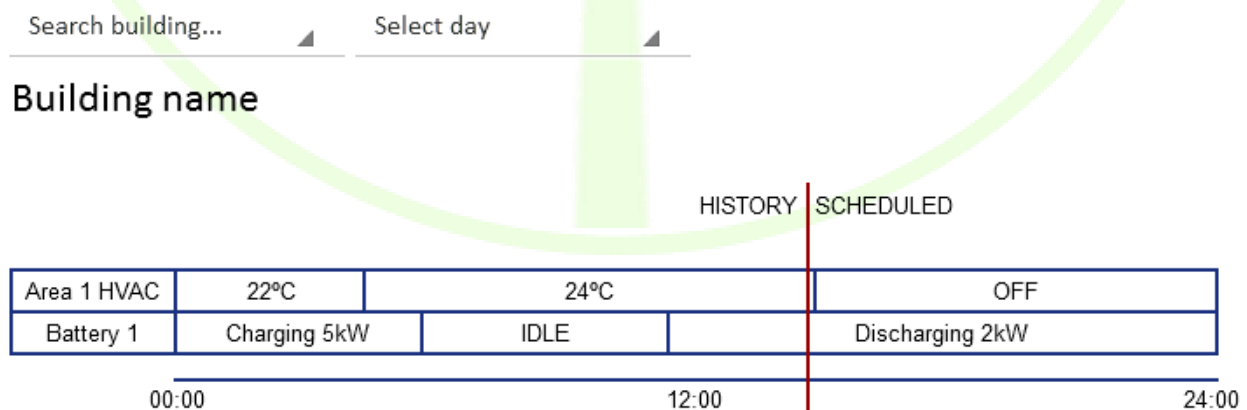
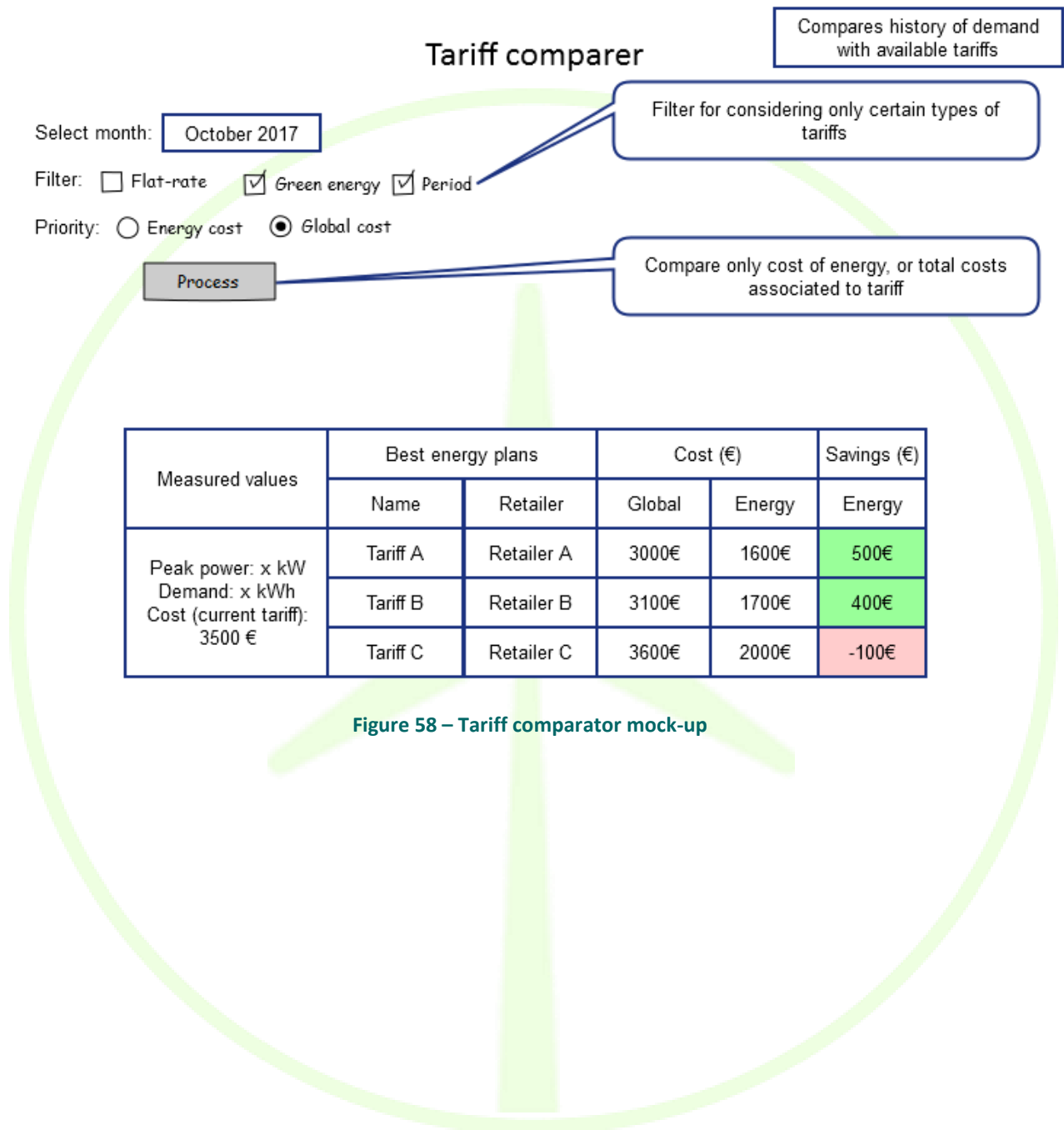


Figure 57 – Automation mockup

4.5.8 Tariff comparator

This section allows to simulate bills accordingly to different tariffs contained in a database. Required data is monthly demand (as provided by the corresponding sensors), and details of the currently contracted tariff (for comparison).



4.5.9 Demand response

Two different screens will give details of the ongoing operations related to the participation of the controllable assets of the building in Demand Response Campaigns.

4.5.9.1 Implicit Demand Response – Dynamic Price

This section will display the current day dynamic price, if the building is participating in implicit demand response campaigns by contracting a dynamic tariff from the retailer (as detailed in the WiseCOOP sections). This price is taken into account by the Energy Usage Optimizer module to calculate the optimum schedule day ahead. The section will also display the calculated optimum schedule.

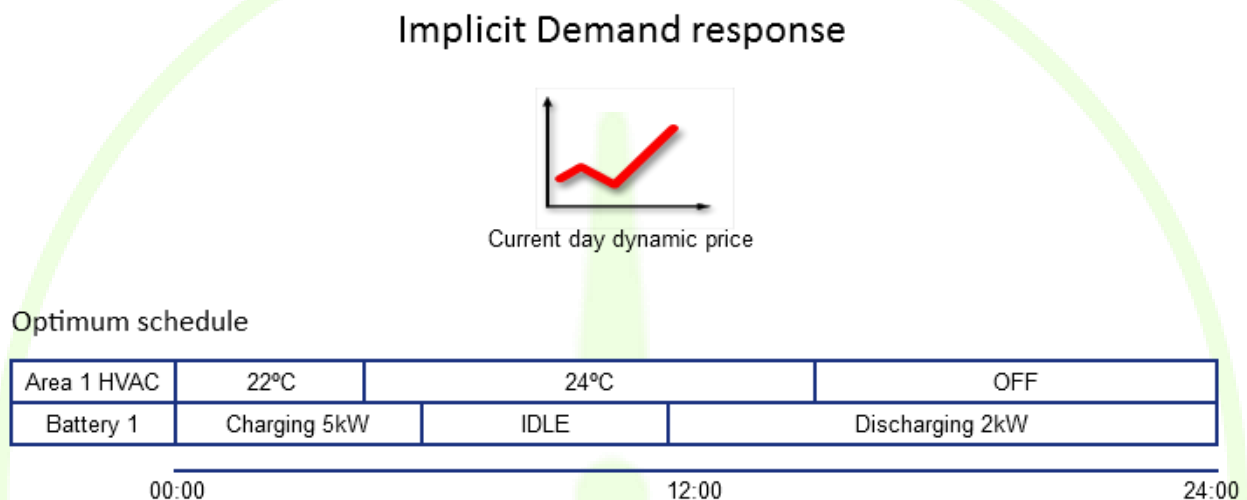


Figure 59 – Implicit DR mock-up

4.5.9.2 Explicit Demand Response

This section displays the modification of the original schedule of the controllable assets, as performed by the Demand Response Optimization framework in order to deliver the amount of flexibility required by an explicit demand response campaign.

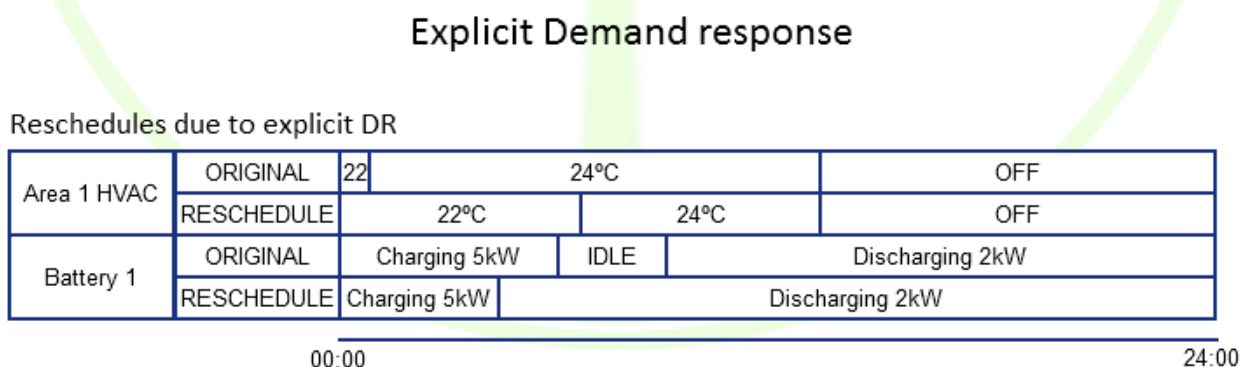


Figure 60– Explicit DR mock-up

5 LINK WITH OTHER WISEGRID APPLICATIONS

5.1 WISECOOP

WiseCOOP is the core tool within the DR framework outlined in the WiseGRID project. Because of this, this product has direct links with another five applications: WG Cockpit, WiseCORP, WG StaaS/VPP, WiseEVP and WiseHOME. Of course, all those interactions are performed through WG IOP. Nevertheless, this interaction with WG IOP will not be further explained in this section as it is not necessary to understand the place of WiseCOOP within the WiseGRID architecture.

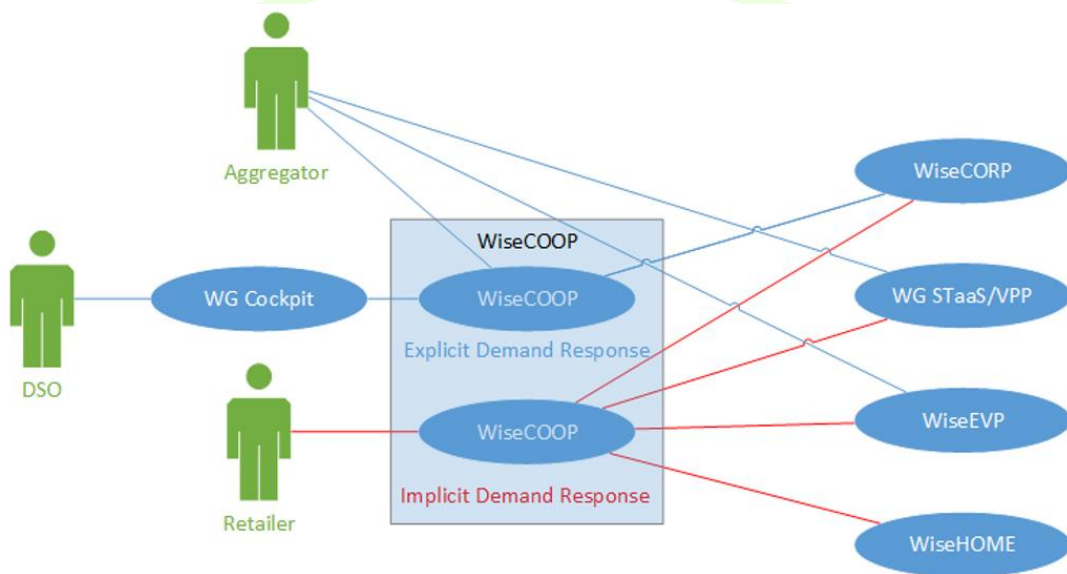


Figure 61 – WiseCOOP interactions

As it can be seen in the previous picture, to better explain the interaction between WiseGRID and the other tools, it is necessary to differentiate the role of WiseCOOP in the explicit DR context and in the implicit DR context.

In an explicit DR situation, the DSO (through WG Cockpit) sends to the different aggregators a DR request in order to solve a problem in the MV or LV grid. WiseCOOP is one of the tools that can be used by these aggregators (the others are the WiseEVP and the WG StaaS/VPP). WiseCOOP answers this WG Cockpit request with an offer (in terms of price and flexibility available). To make this response, WiseCOOP has to previously collect data from WiseCORP (for example building flexibility information).

In case the DSO accepts the offer of the aggregator using WiseCOOP, this product will send the required commands to WiseCORP in order to properly answer WG Cockpit needs.

In an implicit DR situation, WiseCOOP studies and processes the demand and production forecast of WiseHOME, WiseCORP, WG StaaS/VPP and WiseEVP (in a day ahead time frame) in order to send them post-day ahead price signals that will help to balance the retailer's portfolio.

Moreover, WiseCOOP will provide to WiseHOME all the required data in order to allow domestic users to watch their energy behaviour.

5.2 WISECORP

WiseCOOP is the only tool which WiseCORP is linked to. As explained in this section here before, in an implicit DR context, WiseCORP sends to WiseCOOP the expected information of energy demand and production in a day ahead basis. Then, WiseCORP receives from WiseCOOP price signals that will modulate WiseCORP assets

consumption.

Apart from that, when the DSO triggers an explicit DR campaign, WiseCORP will send to WiseCOOP the data of the available flexibility in the building. In case that WiseCOOP finally needs WiseCORP's flexibility, the WiseCORP tool will send the required commands to the corresponding assets (directly or through an EMS).

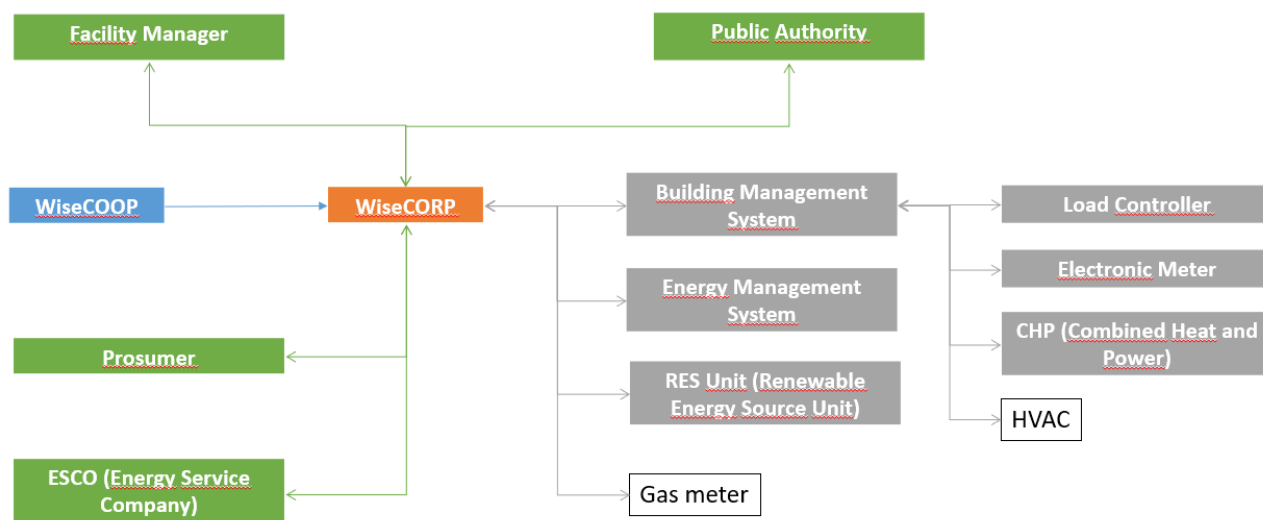


Figure 62 – WiseCORP interactions

WiseCORP's interactions with WiseCOOP and the EMS (Energy Management System) and the other assets, are of course performed through WG IOP.

6 DPIA CONSIDERATIONS

WiseCOOP and WiseCORP are specific tools within the Project and therefore they become 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 neither WiseCOOP nor WiseCORP tools 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 WiseCOOP and WiseCORP are in the Table 19 and Table 20.

Table 19 – Threat and feared events identification for WiseCOOP

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
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.

Feared events	Threat ID	Threat name	Brief explanation why relevant
Diverting of personal data to other users: they are	DoS	Denial of service	Denial of service will lead to unavailability of computing system
Unwanted change in personal data: they are altered or changed	LQD	Lack of quality of data for the purpose of use	If data is used for certain processes it should be adequate.

There is no critical risk level but, due to some risks level at the border between Limited and Significant in case ED, proposed control is:

- Reducing the vulnerabilities of computer communications networks

The aim of this step is to obtain a list of planned and implemented controls for mitigating the identified risks and a new risk map with location of residual risks.

The results of the Risk treatment (Risk modification) based on applied control for WiseCOOP, have brought the risk level within the "Limited" level or below.

Table 20 – Threat and feared events identification for WiseCORP

Feared events	Threat ID	Threat name	Brief explanation why relevant
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.
Breach of use of personal data distributed to people	LT	Lack of transparency	Data processing is not made transparent, or information is not provided in a timely manner...
Illegitimate access to personal data: they are known by unauthorized persons	IACP	Insufficient access control procedures	Access rights are not revoked when they are no longer necessary.

There are no controls necessary so far for WiseCORP as risk level is limited (at low value) or below.

However, due to complete integration of WiseCOOP and WiseCORP within WiseGRID, the general controls should be applied for safer approach:

- 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).
- Data will be anonymized as soon as affordable within the process, with no influence in the functionality.

7 CONCLUSIONS

In this deliverable, the work performed in the framework of task T7.1 has been presented.

Task 7.1 *WiseCOOP and WiseCORP Apps Design* focuses in the analysis of the different requirements and use cases defined by the WP2, the identification of the necessary modules to fulfil those and the specification of the functions to be implemented within each one of the 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 module. This will allow the parallel development of the different modules that are responsibility of different teams within the consortium, as well as 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 T7.2 includes the development of the modules described in this deliverable. These modules will be developed independently and will implement the documented APIs contained in this report. Task T7.2 will finalise with the integration of the different modules that compose the WiseCOOP and WiseCORP applications.

Finally, under task T7.3 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 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 D7.2 *WiseCOOP and WiseCORP Apps implementation and lab-testing*.

8 REFERENCES AND ACRONYMS

8.1 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] [Online]. Available: <http://nobelgrid.eu/>.
- [5] “D4.2 WiseGRID interoperable Integrated Process (IOP)”.
- [6] “<https://spark.apache.org/docs/latest/ml-guide.html>,” [Online].

8.2 ACRONYMS

Table 21 – Acronyms list

Acronyms List	
AC	Alternative Current
AMI	Advanced Metering Infrastructure
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
BACnet	Building Automation and Control Networks

CHP	Combined Heat Power
CIM	Common Information Model
CNEA	Cascaded Neuro-Evolutionary Algorithm
COSEM	COmpanion Specification for Energy Metering
DB	Data Base
DC	Direct Current
DER	Distributed Energy Resource
DLMS	Device Language Message Specification
DPIA	Data Protection Impact Assessment
DR	Demand Response
DSO	Distribution System Operator
EMS	Energy Management System
ESB	Enterprise Service Bus
ESCO	Energy Service Company
HL-UC	High Level Use Case
HTTP	Hypertext Transfer Protocol
HVAC	Heating, Ventilation and Air Conditioning
IoT	Internet of Things
KPI	Key Performance Indicator
MQTT	Message Queue Telemetry Transport
RES	Renewable Energy Source
RPC	Remote Procedure Call
RT	Real-Time
SVM	Support Vector Machine
TCP/IP	Transmission Control Protocol/Internet Protocol
TSO	Transmission System Operator
USEF	Universal Smart Energy Framework
USM	Unbundled Smart Meter

9 ANNEXES

9.1 MESSAGE SPECIFICATIONS FOR BUILDING ASSETS MONITORING AND CONTROL

As different types of events will be handled from the assets, those need to be organized. More specifically, the following event types are considered:

- **Indoor occupancy & environmental** conditions: namely luminance, temperature, humidity, indoor air quality and occupancy data. The structure of the message exchanged is defined below;
- **External environmental conditions**: namely temperature and humidity as retrieved from available weather services;
- **DER operational state** for WiseGRID controllable devices: namely {status, dimming level} for lighting devices, {status, mode, set-point, etc.} for heating/cooling devices, {status} for dual state devices;
- **Energy Consumption** data: setting the baseline (real time) consumption status of the device.

It is clear that raw and processed information is required towards the extraction of accurate Context-Aware Flexibility Profiles. The definition of the aforementioned data types is in line with the overall objective of the profiling engine, which analyses the demand flexibility as a function of multiple parameters, such as time, device operational characteristics, environmental context/ conditions, occupant comfort preferences and health/ hygienic constraints...

9.1.1 Sensor and Devices Monitoring and Control Interfaces specification

A list of topics has been defined towards subscribing to the (near) real time information (Raw events) from building contextual environment. The indicative list of services and the messages exchanged are described below. The following messages are exchanged through the internal ESB on the respective MQTT Topic.

Monitoring of Sensors and Devices

“temperature sensor no.1 associated to SHIC01”

=> TOPIC: ASSET01/SHIC01/0-1-96-9-0-1

Document:

```
{
  "_id" : "0-1-96-9-0-1",
  "assetID" : "assetID",
  "value" : "24.3",
  "unit" : "grdC",
  "status" : "1",
  "captureTime" : ISODate("2016-04-15T10:00:00Z"),
  "description" : "Ambient temperature"
}
```

“luminance sensor no.1 associated to LUX01”

=> TOPIC: ASSET01/LUX01/0-1-151-7-0-1

Document:

```
{
  "_id" : "0-1-151-7-0-1",
  "assetID" : "asset01",
  "value" : <latest reading>,
  "unit" : "lux",
  "status" : "1",
  "captureTime" : ISODate("2016-04-15T10:00:00Z"),
  "description" : "Ambient light"
}
```

"luminance sensor no.1 that PROVIDES TEMPERATURE READINGS (combo)

=> TOPIC: ASSET01/LUX01/0-1-96-9-0-1

Document:

```
{
  "_id" : "0-1-96-9-0-1",
  "assetID" : "asset01",
  "value" : "24.3",
  "value" : "grdC",
  "status" : "1",
  "captureTime" : ISODate("2016-04-15T10:00:00Z"),
  "description" : "Ambient temperature"
}
```

Energy consumption of associated HVAC (SMART-PLUG)

=> TOPIC: ASSET01/SPLUG01/0-1-165-7-0-1

Document:

```
{
  "_id" : "0-1-165-7-0-1",
  "assetID" : "asset01",
  "value" : [cumulative_active_power, cumulative_reactive_power], → as floats

  "unit" : "W",
  "status" : "1",
  "captureTime" : ISODate("2017-03-01T00:00:00Z"),
}
```

```
"description" : "smartplug"
}
```

Energy consumption of associated HVAC (SMART-METER)

=> **TOPIC: ASSET01/SLAM01/0-1-165-7-0-1**

Document:

```
{
  "_id" : "0-1-165-7-0-1",
  "assetID" : "asset01",
  "value" : [cumulative_active_power, cumulative_active_power], → as floats
  "unit" : "W",
  "status" : "1",
  "captureTime" : ISODate("2017-03-01T00:00:00Z"),
  "description" : "smartmeter"
}
```

Control of IR A/C and Lighting devices

TOPIC: ASSET01/SHIC01/0-1-160-7-0-1

Document:

```
{
  "_id" : "0-1-160-7-0-1",
  "assetID" : "asset01",
  "value" : "23",
  "unit" : "grdC",
  "status" : "1",
  "captureTime" : ISODate("2017-03-01T00:00:00Z"),
  "description" : "serial / modbus / IR",
  "mode" : "<heating|cooling>",
  "command" : "auto"
  "state" : "manual"
}
```

"command" : "auto" //indicates the request of an **application** to new status; for example, "auto" if

it is send by auto DR application, "manual" if requested by the user

"state" : "manual" // this variable describes the current state (whose application sent the *running* command)

If an off-value is sent ("status": "0"), the setpoint is ignored.

Return value from internal ESB

Content of MQTT message:

```
{
  "_id" : "0-1-160-7-0-1",
  "assetID" : "asset01",
  "value" : "23",
  "unit" : "grdC",
  "status" : "1",
  "captureTime" : ISODate("2017-03-01T00:00:00Z"),
  "description" : "IR",
  "mode": "cooling",
  "command" : "" // blank field indicates that the abovementioned message has been executed
  "state" : "auto" // This is the current status of the device (after control command requested)
}
```

Commands of type set value (e.g. for controlling a LED lamp)

TOPIC: ASSET01/LED0X/0-1-163-7-0-1

Document:

```
{
  "_id" : "0-1-163-7-0-1",
  "assetID" : "asset01",
  "value" : "20",
  "unit" : "%",
  "status" : "1",
  "captureTime" : ISODate("2017-03-01T00:00:00Z"),
  "description" : "led lamp",
  "command" : "auto"
  "state" : "manual"
}
```

EXAMPLE request to internal ESB

Document:

```
{
  "_id" : "0-1-163-7-0-1",
  "assetID" : "asset01",
  "value" : "20",
  "unit" : "%",
  "status" : "1",
  "captureTime" : ISODate("2017-03-01T00:00:00Z"),
  "description" : "led lamp",
  "command" : "auto"
  "state" : "manual"
}
```

EXAMPLE response from internal ESB.

Document:

```
{
  "_id" : "0-1-163-7-0-1",
  "assetID" : "asset01",
  "value" : "20",
  "unit" : "%",
  "status" : "1",
  "captureTime" : ISODate("2017-03-01T00:00:00Z"),
  "description" : "led lamp",
  "command" : ""
  "state" : "auto"
}
```

After presenting the messages exchanged between the internal ESB implementation and the Demand Flexibility Engine, we proceed with the presentation of the interface between the Demand Flexibility Engine and the Asset Dispatcher.

9.1.2 Interface with Asset Dispatcher

The Asset Dispatcher is responsible for dispatching signals to the available devices in the premises. In the following table, we present an example for interfacing with **Asset Dispatcher** on a publish/subscribe manner via the internal ESB implementation. This is required in case of explicit DR as the Demand Flexibility Engine requires the setpoints that would otherwise run in order to define baseline consumption:

TOPIC : AssetDispatcher

List of object with day-ahead timestamp on a per 15-minute interval for HVAC device:

```
[{
  "_id" : "0-1-160-7-0-1",
  "assetID" : "asset01",
  "value" : "23",
  "unit" : "grdC",
  "status" : "1",
  "captureTime" : ISODate("2017-03-01T00:00:00Z"),
  "description" : "serial / modbus / IR",
  "mode" : "<heating|cooling>",
  "command" : "assetDispatcher",
  "state" : "manual"
},
{
  "_id" : "0-1-160-7-0-1",
  "assetID" : "asset01",
  "value" : "24",
  "unit" : "grdC",
  "status" : "1",
  "captureTime" : ISODate("2017-03-01T00:15:00Z"),
  "description" : "serial / modbus / IR",
  "mode" : "<heating|cooling>",
  "command" : "assetDispatcher",
  "state" : "manual"
}, ...,
{
  "_id" : "0-1-160-7-0-1",
  "assetID" : "asset01",
  "value" : "23",
  "unit" : "grdC",
  "status" : "1",
```

```

"captureTime" : ISODate("2017-03-01T23:45:00Z"),
"description" : "serial / modbus / IR",
"mode": "<heating|cooling>",
"command" : "assetDispatcher",
"state" : "manual"
}}

"command" : "auto" //indicates the request of an application to new status; for example, "auto" if
it is send by the auto DR application, "manual" if requested by the user,
).

"state" : "manual" // this variable describes the current state (whose application sent the running
command)

List of object with day-ahead timestamp on a per 15-minute interval for led lamp:
[
{
  "_id" : "0-1-163-7-0-1",
  "assetID" : "asset01",
  "value" : "20",
  "unit" : "%",
  "status" : "1",
  "captureTime" : ISODate("2017-03-01T00:00:00Z"),
  "description" : "led lamp",
  "command" : "assetDispatcher",
  "state" : "manual"
},
{
  "_id" : "0-1-163-7-0-1",
  "assetID" : "asset01",
  "value" : "20",
  "unit" : "%",
  "status" : "1",
  "captureTime" : ISODate("2017-03-01T00:15:00Z"),
  "description" : "led lamp",
  "command" : "assetDispatcher",
  "state" : "manual"
}, ...,
{

```

```

    "_id" : "0-1-163-7-0-1",
    "assetID" : "asset01",
    "value" : "20",
    "unit" : "%",
    "status" : "1",
    "captureTime" : ISODate("2017-03-01T23:45:00Z"),
    "description" : "led lamp",
    "command" : "assetDispatcher"
    "state" : "manual"
  }}

```

9.1.3 Configuration Parameters (Commissioning time)

The following is a general configuration file that includes information on all devices installed in premises (someone has to fill that).

After the following unified JSON body (in which some parts are abstractly defined), we specifically define the template (configuration JSON body) for each device and **combo** sensors (luminance + temperature).

```

{
  "WiseCORPConfig": {
    "Controllers": {
      "SHIC": [{
        "asset id": "String",
        "shic id": "String",
        "shic obis code": "String",
        "metadata": { "type": "\"String\"" },
        "control_type": "SG-ready / serial / modbus / IR",
        "submeter": {
          "present": "true/false",
          "Splug or SLAM obis code": "String",
          "settings": {
            "LP_reporting_interval": "integer",
            "LP_resolution": "integer"
          },
        },
        "temp_sensor": {
          "temp obis code": "string",
          "offset": "float"
        }
      ]
    }
  }
}

```

```

    }
  }
},...],
"Smartplug": [{
  "asset id": "String",
  "splug id": "String",
  "splug obis code": "String",
  "nominal_power": "watts, integer",
  "submeter": {
    "settings": {
      "LP_reporting_interval": "integer",
      "LP_resolution": "integer"
    }
  }
},...],
"LED_lamp": [{
  "asset id": "String",
  "led id": "String",
  "led obis code": "String",
  "nominal_power": "'watts, integer'"
}
},...],
"Sensors": {
  "lux_sensor": [{
    "asset id": "String",
    "lux id": "string",
    "lux obis code": "String",
    "lux_reporting_interval": "integer",
    "lux_threshold": "'lux, float'",
    "temp_sensor": {
      "temp obis code": "string",
      "offset": "float"
    }
  },...]
}
}
}

```

}

EACH TEMPLATE SEPARATELY

SHIC template

Database: SMX

Collection: SHIC

Document: SHIC_ID?

```
{
  "smx id": SMX01,
  "shic id": SHIC01,
  "shic obis code": "0-1-160-7-0-1"
  "type": "SHIC",
  "appliance metadata":
  {
    "type": <HVAC / AC>
    "category": <inverter/non inverter, ID>
    "nominal power": <watts, integer>
    "cooling capacity": <watts, integer>
    "heating capacity": <watts, integer>
    "cooling efficiency": <percentage, float>
    "heating efficiency": <percentage, float>
    "min setpoint": <temperature value, float>
    "max setpoint": <temperature value, float>
  },
  "control type": <SG-ready / serial / modbus / IR>,
  "submeter": <true/false>,
  "splug obis code": "0-1-165-7-0-1",
  "submeter settings": {
    "LP reporting interval": 60 sec,
    "LP resolution": 60 sec,
  },
  "temp obis code": "0-1-96-9-0-1" or "null"
  "temp sensor offset": <float>
}
```

#Comments: The load threshold can be set to get a specific notification on specific load changes.

Lux sensor template

```
{
  "asset id": "ASSET01",
  "lux id": "LUX01",
  "lux obis code": "0-1-151-7-0-1",
  "type": "lux sensor",
  "lux reporting interval": <60 / 120 / xyz sec>,
  "lux threshold":<lux, float>
  "temp obis id": "0-1-96-9-0-1" or "null",
  "temp sensor offset": <float>
}
```

#Comments: The lux threshold can be set to get a specific notification on when the light changes.

Smartplug template

```
{
  "asset id": "ASSET01",
  "splug id": "SPLUG01",
  "splug obis code": "0-1-165-7-0-1",
  "type": "smartplug",
  "nominal power":<watts, integer>
  "submeter settings": {
    "LP reporting interval": 60 sec,
    "LP resolution": 60 sec,
  }
}
```

#Comments: The load threshold can be set to get a specific notification on specific load changes.

LED lamp template

```
{
  "asset id": "ASSET01",
  "led id": "LED01"
  "led obis code": "0-1-163-7-0-1"
  "type": "LED_lamp",
  "nominal power": <watts, integer>
}
```

SLAM template

```
{
  "asset id": "ASSET01",
  "slam id": "SLAM01"
  "slam obis code": "0-1-165-7-0-1"
  "type": "smartmeter",
  "nominal power": <watts, integer>
  "submeter settings": {
    "LP reporting interval": 60 sec,
    "LP resolution": 60 sec,
  }
}
```

9.2 ENERGY USAGE OPTIMIZER EQUATIONS

9.2.1 General Energy Storage

9.2.1.1 Optimization variables

The optimization variables for the time step i are: ed_i , ea_i , ee_i , eb_i and cv_i .

The vector expression is:

$$\begin{pmatrix} ed_1 \\ ed_n \\ ea_1 \\ ea_n \\ ee_1 \\ ee_n \\ eb_1 \\ eb_n \\ cv_1 \\ cv_n \end{pmatrix}$$

9.2.1.2 Optimization function

Optimization function corresponds to the cost of the used energy that is the sum of the stored energy ea and the energy supplied directly ed .

$$f = \sum c_i \cdot (ea_i + ed_i)$$

The vector expression is:

$$f = (cpr_1 \quad cpr_n \quad cpr_1 \quad cpr_n \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0) \cdot \begin{pmatrix} ed_1 \\ ed_n \\ ea_1 \\ ea_n \\ ee_1 \\ ee_n \\ eb_1 \\ eb_n \\ cv_1 \\ cv_n \end{pmatrix}$$

9.2.1.3 Lower limits

Energy must be greater or equal to 0.

$$ed_i \geq 0$$

$$ea_i \geq 0$$

$$ee_i \geq 0$$

$$eb_i \geq 0$$

$$cv_i \geq 0$$

The vector expression is:

$$\begin{pmatrix} ed_1 \\ ed_n \\ ea_1 \\ ea_n \\ ee_1 \\ ee_n \\ eb_1 \\ eb_n \\ cv_1 \\ cv_n \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

9.2.1.4 Upper limits

Supplied energy directly cannot be greater than the produced energy.

$$ed_i \leq pr_i$$

The stored energy cannot be greater than the produced energy.

$$ea_i \leq pr_i$$

The stored energy cannot be greater than de input energy limited by the input power.

$$ema_i = pma \cdot dt_i$$

The input energy is limited by the produced energy and the input power.

$$emma_i = \min(pr_i, ema_i)$$

The extracted energy is limited by the maximum output power.

$$eme_i = pme \cdot dt_i$$

So:

$$ee_i \leq eme_i$$

The stored energy is limited by the storage capacity.

$$eb_i \leq eamax$$

The variable consumption in each time step is limited by sum of the produced energy, the fixed consumption and the maximum extracted energy.

$$cv_i \leq pr_i + cf_i + eme_i$$

The matrix expression is:

$$\begin{pmatrix} ed_1 \\ ed_n \\ ea_1 \\ ea_n \\ ee_1 \\ ee_n \\ eb_1 \\ eb_n \\ cv_1 \\ cv_n \end{pmatrix} \leq \begin{pmatrix} pr_1 \\ pr_n \\ emma_1 \\ emma_n \\ eme_1 \\ eme_n \\ eamax \\ eamax \\ pr_1 - cf_1 + eme_1 \\ pr_n - cf_n + eme_n \end{pmatrix}$$

9.2.1.5 Equalities

The total variable consumption is equal to the total consumption required.

$$\sum cv_i = ereq$$

In each time step the fixed consumption plus the variable consumption is equal the energy supplied directly plus the extracted energy.

$$ed_i + ee_i = cf_i + cv_i$$

In each time step the variation in the stored energy is equal the input energy (considering the performance) minus the extracted energy.

$$eb_i - eb_{i-1} = rnd \cdot ea_i - ee_i$$

The matrix expression is:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & -rnd & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -rnd & 0 & 1 & -1 & 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} ed_1 \\ ed_n \\ ea_1 \\ ea_n \\ ee_1 \\ ee_n \\ eb_1 \\ eb_n \\ cv_1 \\ cv_n \end{pmatrix} = \begin{pmatrix} ereq \\ cf_1 \\ cf_n \\ 0 \\ 0 \end{pmatrix}$$

9.2.1.6 Inequalities

The directly supplied energy cannot be greater than the produced energy.

$$ea_i + ed_i \leq pr_i$$

In each time step the extracted energy cannot be greater than the stored energy.

$$ee_i \leq eb_i$$

In each time step the fixed consumption plus the variable consumption cannot be greater than the directly supplied energy plus the extracted energy.

$$ed_i + ee_i \geq cf_i + cv_i$$

The matrix expression is:

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} ed_1 \\ ed_n \\ ea_1 \\ ea_n \\ ee_1 \\ ee_n \\ eb_1 \\ eb_n \\ cv_1 \\ cv_n \end{pmatrix} \leq \begin{pmatrix} ed_1 \\ ed_n \\ 0 \\ 0 \\ -cf_1 \\ -cf_n \end{pmatrix}$$

9.2.2 HVAC

9.2.2.1 Equalities

The Stored energy variation is equal the Input energy (considering performance) minus the Supplied energy.

$$eb_{i,j} = eb_{i,j-1} + r_i \cdot ea_{i,j} - es_{i,j}$$

For the first time step the constraint has the form:

$$eb_{i,1} = eb_{i,0} + r_i \cdot ea_{i,1} - es_{i,1}$$

In time step j and in device i , the Supplied energy is equal the fixed consumption plus the variable consumption.

The fixed consumption in device i is:

$$cf_i = pf_i \cdot dt$$

The total consumption for device i in time step j is:

$$ct_{i,j} = cv_{i,j} + cf_i$$

So:

$$es_{i,j} = ct_{i,j}$$

The matrix expression is:

$$\begin{pmatrix} -r & 0 & 1 & 0 & 1 & 0 & -r & 0 & 1 & 0 & 1 & 0 \\ 0 & -r & -1 & 1 & 0 & 1 & 0 & -r & -1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} ea_{1,1} \\ ea_{1,nj} \\ eb_{1,1} \\ eb_{1,nj} \\ es_{1,1} \\ es_{1,nj} \\ ea_{ni,1} \\ ea_{ni,nj} \\ eb_{ni,1} \\ eb_{ni,nj} \\ es_{ni,1} \\ es_{ni,nj} \end{pmatrix} = \begin{pmatrix} eb_{0,1} \\ 0 \\ ct_{1,1} \\ ct_{ni,nj} \end{pmatrix}$$

9.2.3 Global Optimization

9.2.3.1 Optimization Variables

Optimization is defined by the following variables: ee_{ij} , $111si_j$, es_{ij} , eu_{ij} y ep_{ij} y $111si_j$.

These variables are expressed in vector form as:

$$\begin{pmatrix} ee_{1,1} \\ ee_{1,nj} \\ eb_{1,1} \\ eb_{1,nj} \\ es_{1,1} \\ es_{1,nj} \\ eu_{1,1} \\ eu_{1,nj} \\ ep_{1,1} \\ ep_{1,nj} \\ ec_{1,1} \\ ec_{1,nj} \\ ee_{ni,1} \\ ee_{ni,nj} \\ eb_{ni,1} \\ eb_{ni,nj} \\ es_{ni,1} \\ es_{ni,nj} \\ eu_{ni,1} \\ eu_{ni,nj} \\ ep_{ni,1} \\ ep_{ni,nj} \\ ec_{ni,1} \\ ec_{ni,nj} \end{pmatrix}$$

Note: to simplify vector and matrix the expression of the variables, only the first and last device are shown as well as the first and last interval.

These variables can be expressed in a general form:

$$e_{i,j} \equiv x_l$$

In vector expression:

$$\begin{pmatrix} x_1 \\ x_{nl} \end{pmatrix}$$

Where l is:

$$l = (i - 1) \cdot nk \cdot nj + (k - 1) \cdot nj + j$$

And $nk=6$ with $nl = ni \cdot nj \cdot nk$.

9.2.3.2 Optimization Function

The optimization function represents a cost of the input energy. This cost can be expressed in monetary units or in other types of cost like CO₂ emissions, pollutant emissions, etc.

The optimization function is defined as:

$$f = \sum_{i=1}^{ni} \sum_{j=1}^{nj} c_j \cdot ee_{i,j}$$

The vector expression is:

$$\begin{pmatrix} c_1 & c_{nj} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_1 & c_{nj} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} ee_{1,1} \\ ee_{1,nj} \\ eb_{1,1} \\ eb_{1,nj} \\ es_{1,1} \\ es_{1,nj} \\ eu_{1,1} \\ eu_{1,nj} \\ ep_{1,1} \\ ep_{1,nj} \\ ec_{1,1} \\ ec_{1,nj} \\ ee_{ni,1} \\ ee_{ni,nj} \\ eb_{ni,1} \\ eb_{ni,nj} \\ es_{ni,1} \\ es_{ni,nj} \\ eu_{ni,1} \\ eu_{ni,nj} \\ ep_{ni,1} \\ ep_{ni,nj} \\ ec_{ni,1} \\ ec_{ni,nj} \end{pmatrix}$$

It is possible to extend the model by costs that affect other variables such as the cost due to energy storage.

9.2.3.3 Lower limits

Energy must be positive:

$$ee_{i,j} \geq 0$$

$$eb_{i,j} \geq 0$$

$$es_{i,j} \geq 0$$

$$eu_{i,j} \geq 0$$

$$ep_{i,j} \geq 0$$

$$ec_{i,j} \geq 0$$

The vector expression is:

$$- \begin{pmatrix} ee_{1,1} \\ ee_{1,nj} \\ eb_{1,1} \\ eb_{1,nj} \\ es_{1,1} \\ es_{1,nj} \\ eu_{1,1} \\ eu_{1,nj} \\ ep_{1,1} \\ ep_{1,nj} \\ ec_{1,1} \\ ec_{1,nj} \\ ee_{ni,1} \\ ee_{ni,nj} \\ eb_{ni,1} \\ eb_{ni,nj} \\ es_{ni,1} \\ es_{ni,nj} \\ eu_{ni,1} \\ eu_{ni,nj} \\ ep_{ni,1} \\ ep_{ni,nj} \\ ec_{ni,1} \\ ec_{ni,nj} \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

9.2.3.4 Upper Limits

9.2.3.4.1 Input energy

Input energy is limited by the input power pe_i .

$$emp_i = pe_i \cdot dt$$

And:

$$ee_{i,j} \leq emp_i$$

9.2.3.4.2 Stored Energy

The energy stored in the accumulator is limited by storage capacity.

$$eb_{i,j} \leq ebmax_i$$

9.2.3.4.3 Supplied energy

Supplied energy is limited by the output power ps_i .

$$ems_i = ps_i \cdot dt$$

And:

$$es_{i,j} \leq ems_i$$

9.2.3.4.4 Used energy

Input energy is limited by the input power pu_i

$$emu_i = pu_i \cdot dt$$

And:

$$eu_{i,j} \leq emu_i$$

9.2.3.4.5 Self-stored energy

The energy stored in the accumulator for self-usage is limited by storage capacity.

$$ep_{i,j} \leq epmax_i$$

9.2.3.4.6 Consumed energy

Consumed energy is limited by the output power pc_i .

$$emc_i = pc_i \cdot dt$$

And:

$$ec_{i,j} \leq emc_i$$

9.2.3.4.7 Vector expression

The vector expression is:

$$\begin{pmatrix} ee_{1,1} \\ ee_{1,nj} \\ eb_{1,1} \\ eb_{1,nj} \\ es_{1,1} \\ es_{1,nj} \\ eu_{1,1} \\ eu_{1,nj} \\ ep_{1,1} \\ ep_{1,nj} \\ ec_{1,1} \\ ec_{1,nj} \\ ee_{ni,1} \\ ee_{ni,nj} \\ eb_{ni,1} \\ eb_{ni,nj} \\ es_{ni,1} \\ es_{ni,nj} \\ eu_{ni,1} \\ eu_{ni,nj} \\ ep_{ni,1} \\ ep_{ni,nj} \\ ec_{ni,1} \\ ec_{ni,nj} \end{pmatrix} \leq \begin{pmatrix} emp_1 \\ emp_1 \\ ebmax_1 \\ ebmax_1 \\ ems_1 \\ ems_1 \\ emu_1 \\ emu_1 \\ epmax_1 \\ epmax_1 \\ emc_1 \\ emc_1 \\ emp_{ni} \\ emp_{ni} \\ ebmax_{ni} \\ ebmax_{ni} \\ ems_{ni} \\ ems_{ni} \\ emu_{ni} \\ emu_{ni} \\ epmax_{ni} \\ epmax_{ni} \\ emc_{ni} \\ emc_{ni} \end{pmatrix}$$

9.2.3.5 Equalities Constraint

9.2.3.5.1 Equality 1

Variation of the stored energy is equal to the input energy (considering performance) minus the Energy supplied.

$$eb_{i,j} - eb_{i,j-1} = r_i \cdot ee_{i,j} - es_{i,j}$$

So:

$$-r_i \cdot ee_{i,j} - eb_{i,j-1} + eb_{i,j} + es_{i,j} = 0$$

9.2.3.5.2 Equality 2

Variation of the self-stored energy is equal to the used energy (considering performance) minus the consumed energy.

$$ep_{i,j} - ep_{i,j-1} = r_i \cdot eu_{i,j} - ec_{i,j}$$

So:

$$-r_i \cdot eu_{i,j} - ep_{i,j-1} + ep_{i,j} + ec_{i,j} = 0$$

9.2.3.5.3 Equality 3

The used energy is equal the supplied energy in each time step.

$$\sum_{i=1}^{ni} es_{i,j} - \sum_{i=1}^{ni} eu_{i,j} = 0$$

9.2.3.5.4 Equality 4

The total variable consumption plus the total fixed consumption is equal to the total consumed energy.

$$\sum_{i=1}^{ni} \sum_{j=1}^{nj} ec_{i,j} = \sum_{i=1}^{ni} \sum_{j=1}^{nj} cv_{i,j} + \sum_{i=1}^{ni} \sum_{j=1}^{nj} cf_i = ctt$$

9.2.3.5.5 Equality 5

The total consumption energy by device is equal the total fixed consumption by device plus the total variable consumption by device (Total required consumption by device).

$$\sum_{j=1}^{nj} ec_{i,j} = \sum_{j=1}^{nj} cf_{i,j} + \sum_{j=1}^{nj} cv_{i,j} = cft_i + crt_i$$

9.2.3.5.6 Matrix Expression

The matrix expression has the form:

$$Aeq \cdot x = beq$$

Given the complexity of the matrix Aeq instead of being expressed by components it is expressed by submatrices:

$$\begin{pmatrix}
 -r_1 I & S & I & C & C & C & C & C & C & C & C & C \\
 C & C & C & C & C & C & -r_{ni} I & S & I & C & C & C \\
 C & C & C & -r_1 I & S & I & C & C & C & C & C & C \\
 C & C & C & C & C & C & -r_{ni} I & S & I & C & C & C \\
 C & C & I & -I & C & C & C & C & I & -I & C & C \\
 c & c & c & c & c & u & c & c & c & c & c & u \\
 c & c & c & c & c & u & c & c & c & c & c & c \\
 c & c & c & c & c & c & c & c & c & c & c & u
 \end{pmatrix} \cdot \begin{pmatrix}
 ee_{1,1} \\
 ee_{1,nj} \\
 eb_{1,1} \\
 eb_{1,nj} \\
 es_{1,1} \\
 es_{1,nj} \\
 eu_{1,1} \\
 eu_{1,nj} \\
 ep_{1,1} \\
 ep_{1,nj} \\
 ec_{1,1} \\
 ec_{1,nj} \\
 ee_{ni,1} \\
 ee_{ni,nj} \\
 eb_{ni,1} \\
 eb_{ni,nj} \\
 es_{ni,1} \\
 es_{ni,nj} \\
 eu_{ni,1} \\
 eu_{ni,nj} \\
 ep_{ni,1} \\
 ep_{ni,nj} \\
 ec_{ni,1} \\
 ec_{ni,nj}
 \end{pmatrix} = \begin{pmatrix}
 cc \\
 cc \\
 cc \\
 cc \\
 cc \\
 ctt \\
 ctd_1 \\
 ctd_{ni}
 \end{pmatrix}$$

Where:

Table 22 – Global Optimization Model, equalities matrix expression

Submatrix / Vector	Description
C	Null matrix of dimension $nj \cdot nj$.
C	Null vector of dimension $1 \cdot nj$.
Cc	Null vector of dimension $nj \cdot 1$.
I	Identity matrix of dimension $nj \cdot nj$.
S	Special matrix of dimension $nj \cdot nj$ and form: <div style="text-align: center;"> $\begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$ </div>
u	Ones vector of dimension $1 \cdot nj$.

9.2.3.6 Inequalities Constraint

9.2.3.6.1 Inequality 1

Input energy for all devices in the time step j cannot be greater than the energy available in the interval.

$$\sum_i^{ni} ee_{i,j} \leq ed_j$$

9.2.3.6.2 Inequality 2

Input energy (considering performance) in all the devices and time steps must be greater or equal the total consumption.

$$\sum_i^{ni} \sum_j^{nj} r_i \cdot ee_{i,j} \geq \sum_{i=1}^{ni} \sum_{j=1}^{nj} ec_{i,j} = \sum_{i=1}^{ni} \sum_{j=1}^{nj} cv_{i,j} + \sum_{i=1}^{ni} \sum_{j=1}^{nj} cf_i = ctt$$

So:

$$-\sum_i^{ni} \sum_j^{nj} r_i \cdot ee_{i,j} \leq -ctt$$

9.2.3.6.3 Inequality 3

Total input energy (considering performance) must be greater or equal the total supplied energy.

$$\sum_{i=1}^{ni} \sum_{j=1}^{nj} r_i \cdot ee_{i,j} \geq \sum_{i=1}^{ni} \sum_{j=1}^{nj} es_{i,j}$$

So:

$$-\sum_{i=1}^{ni} \sum_{j=1}^{nj} r_i \cdot ee_{i,j} + \sum_{i=1}^{ni} \sum_{j=1}^{nj} es_{i,j} \leq 0$$

9.2.3.6.4 Inequality 4

Total used energy (considering performance) must be greater or equal the total consumed energy.

$$\sum_{i=1}^{ni} \sum_{j=1}^{nj} r_i \cdot eu_{i,j} \geq \sum_{i=1}^{ni} \sum_{j=1}^{nj} ec_{i,j}$$

So:

$$-\sum_{i=1}^{ni} \sum_{j=1}^{nj} r_i \cdot eu_{i,j} + \sum_{i=1}^{ni} \sum_{j=1}^{nj} ec_{i,j} \leq 0$$

9.2.3.6.5 Inequality 5

The consumed energy by device and time step must be greater or equal the fixed consumption.

$$ec_{i,j} \geq cf_{i,j}$$

So:

$$-ec_{i,j} \leq -cf_{i,j}$$

9.2.3.6.6 Matrix Expression

The Matrix expression, following the nomenclature of submatrices takes the form:

$$\begin{pmatrix}
 I & C & C & C & C & C & I & C & C & C & C & C \\
 -r_1 u & c & c & c & c & c & -r_{ni} u & c & c & c & c & c \\
 -r_1 u & c & u & c & c & c & -r_{ni} u & c & u & c & c & c \\
 c & c & c & -r_1 u & c & u & c & c & c & -r_{ni} u & c & c \\
 C & C & C & C & C & -I & C & C & C & C & C & C \\
 C & C & C & C & C & C & C & C & C & C & C & -I
 \end{pmatrix} \cdot \begin{pmatrix}
 ee_{1,1} \\
 ee_{1,nj} \\
 eb_{1,1} \\
 eb_{1,nj} \\
 es_{1,1} \\
 es_{1,nj} \\
 eu_{1,1} \\
 eu_{1,nj} \\
 ep_{1,1} \\
 ep_{1,nj} \\
 ec_{1,1} \\
 ec_{1,nj} \\
 ee_{ni,1} \\
 ee_{ni,nj} \\
 eb_{ni,1} \\
 eb_{ni,nj} \\
 es_{ni,1} \\
 es_{ni,nj} \\
 eu_{ni,1} \\
 eu_{ni,nj} \\
 ep_{ni,1} \\
 ep_{ni,nj} \\
 ec_{ni,1} \\
 ec_{ni,nj}
 \end{pmatrix} \leq \begin{pmatrix}
 ed_1 \\
 ed_{nj} \\
 -ctt \\
 0 \\
 0 \\
 -cf_{1,1} \\
 -cf_{1,nj} \\
 -cf_{ni,1} \\
 -cf_{ni,nj}
 \end{pmatrix}$$