



Title:	Document Version:
D7.2 WiseCOOP and WiseCORP Apps implementation and lab-testing	0.5

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*:	
M21 (July 20 <mark>18</mark>)	M21 (July 2018)	RD-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), Pascual Mullor (ITE), Antonis Papanikolaou (HYP), Benjamin Kraft (VS), Leandro Lombardo (ENG), Alexandre Lapedra (BYES), Vincent Dierickx (ECO-EID), Ine Swennen (ECO-EID), Foivos Palaiogiannis (ICCS)

Abstract:

This document reports the work performed within Task 7.2 "WiseCOOP and WiseCORP Apps Implementation" and Task 7.3 "WiseCOOP and WiseCORP Apps lab-testing and refinement", following the specifications and architecture designed in Task 7.1 and reported in D7.1.

Keywords:

Energy efficiency, Building, Cooperative, Facility manager, ESCO, Retailer, DR, Flexibility; Aggregator, Implementation, Integration, Lab-Testing

ETRA I+D INVESTIGACIÓN Y DESARROLLO, S.A.

Avda. Tres Forques, 147 46014 Valencia (España) Tel. +34 96 313 40 82 Fax +34 96 350 32 34 http://www.wisegrid.eu amarias.etraid@grupoetra.com





Revision History

Revision	Date	Description	Author (Organisation)
0.1	22.06.2018	ТоС	Alberto Zambrano (ETRA)
0.2	09.07.2018	Sections 3.2 and 4.2. Integration of test templates	Alberto Zambrano (ETRA), Antonis Pa- panikolaou (HYP), Pascual Mullor (ITE)
0.3	18.07.2018	Sections 3 and 4	Alberto Zambrano (ETRA)
0.4	27.07.2018	Last Contributions and ready for peer review	Alberto Zambrano
0.5	02.08.2018	Deliverable ready for Submission	Alberto Zambrano (ETRA), Álvaro Nofuentes (ETRA)







INDEX

LIS	t of Tables	7
EXI	ECUTIVE SUMMARY	8
1	INTRODUCTION	1
1.1	Purpose of the document	
1.2	Scope of the document	
1.3	Structure of the document	
2	LAB-TESTING APPROACH	2
	Definition of terms	
2.2	Test plan	
	Features to be tested	
	Test cases specification	
2.5	Lab-testing platform details	
2		
~ 4	WISECOOP APPLICATION	1
3.1	Implementation	1
3.1	Implementation 21 3.1.1 Architecture overview 21	1
3.1	Implementation213.1.1Architecture overview213.1.2Back-office modules22	1
	Implementation213.1.1Architecture overview213.1.2Back-office modules223.1.3User interface35	1
	Implementation213.1.1Architecture overview213.1.2Back-office modules223.1.3User interface35Lab-testing results46	1
	Implementation 21 3.1.1 Architecture overview 21 3.1.2 Back-office modules 22 3.1.3 User interface 35 Lab-testing results 46 3.2.1 RT monitor tests 46	1
	Implementation213.1.1Architecture overview213.1.2Back-office modules223.1.3User interface35Lab-testing results46	1
	Implementation 21 3.1.1 Architecture overview 21 3.1.2 Back-office modules 22 3.1.3 User interface 35 Lab-testing results 46 3.2.1 RT monitor tests 46	1
	Implementation 21 3.1.1 Architecture overview 21 3.1.2 Back-office modules 22 3.1.3 User interface 35 Lab-testing results 46 3.2.1 RT monitor tests 46 3.2.2 KPI engine tests 48	1
	Implementation. 21 3.1.1 Architecture overview. 21 3.1.2 Back-office modules. 22 3.1.3 User interface. 35 Lab-testing results 46 3.2.1 RT monitor tests. 46 3.2.2 KPI engine tests. 48 3.2.3 Forecast modules tests 55	1
	Implementation. 21 3.1.1 Architecture overview. 21 3.1.2 Back-office modules. 22 3.1.3 User interface. 35 Lab-testing results 46 3.2.1 RT monitor tests. 46 3.2.2 KPI engine tests. 48 3.2.3 Forecast modules tests 55 3.2.4 Tariff designer and comparer tests. 60	1

แรดอาป



4.1	Imple	mentation67
	4.1.1	Architecture overview67
	4.1.2	Back-office modules
	4.1.3	User interface75
4.2	Lab-te	esting results
		RT monitor tests
	4.2.2	KPI engine tests
	4.2.3	Forecast modules tests
	4.2.4	Tariff comparer tests
	4.2.5	Energy usage optimizer
		DR framework tests
	4.2.7	Asset dispatcher
5	CONCI	USIONS AND NEXT STEPS112
		ENCES AND ACRONYMS
6.1	Refere	ences
6.2	Acron	yms 113





List of figures

Figure 1 – Screenshot of wisegridpre.lab.id server on vSphere platform	18
Figure 2 – Screenshot of wisegridpre.lab.id server on vSphere platform	18
Figure 3 – Lab-testing platform core modules overview	20
Figure 4 – Dashboard of wisegridpre.lab.id, all modules installed as Docker containers	20
Figure 5 – Overview of interactions among the modules of the WiseCOOP application	21
Figure 6 – Queues created in the internal ESB for exchange of information among modules o	
Figure 7 – List of actives connections (32) to the internal ESB of WiseCOOP	24
Figure 8 – Screenshot of the Real-time monitor UI, showing energy readings schema	25
Figure 9 – Screenshot of Spark server with executing WiseCOOP jobs	27
Figure 10 – WiseCOOP long-term database screenshot	33
Figure 11 – Screenshot of forecast response message	33
Figure 12 – WiseCOOP Login	36
Figure 13 – WiseCOOP UI – Dashboard	37
Figure 14 – WiseCOOP UI – Prosumer detail	37
Figure 15 – WiseCOOP UI – Geographical demand heat-map	38
Figure 16 – WiseCOOP UI – Tariff period definition	39
Figure 17 – WiseCOOP UI – Tariff definition	39
Fi <mark>g</mark> ure 18 – WiseCOOP UI – Portfolio profiling, identified groups	40
Figure 19 – WiseCOOP UI – Portfolio profiling, member comparison with average profile	41
Figure 20 – WiseCOOP UI – Tariff comparer, selection of criteria	41
Figure 21 – WiseCOOP UI – Tariff comparer, results	42
Figure 22 – WiseCOOP UI - Tariff comparer, simulated bills	42
Figure 23 – WiseCOOP UI – Energy trade assistant	4 <mark>3</mark>
Figure 24 – WiseCOOP UI – Implicit demand response	44
Figure 25 – WiseCOOP – List of active and finished explicit demand response campaigns	45
Figure 26 – WiseCOOP – Details of an explicit demand response campaigns	46
Figure 27 – Overview of interactions among the modules of the WiseCORP application	67
Figure 28 – Screenshot of the Real-time monitor UI, showing asset status schema	70
Figure 29 – Screenshot of Spark server with executed WiseCORP jobs	71
Figure 30 – WiseCORP long-term database screenshot	72
Figure 31 – Screenshot of forecast response message.	72
Figure 32 – WiseCORP Login	75
Figure 33 – WiseCORP UI – Dashboard	
Figure 34 – WiseCORP UI – Building energy usage details (i)	77





Figure 35 – WiseCORP UI – Building energy usage details (ii)	77
Figure 36 – WiseCORP UI – Building energy usage details (iii)	78
Figure 37 – WiseCORP UI – Building automation details	78
Figure 38 – WiseCORP UI – Representation of the buildings in a map	79
Figure 39 – WiseCORP UI – Monitoring	80
Figure 40 – WiseCORP UI – Tariff optimization, selection of criteria	80
Figure 41 – WiseCORP UI – Tariff optimization, results	81
Figure 42 – WiseCORP UI - Tariff optimization, simulated bills	81
Figure 43 – WiseCORP – Implicit Demand Response	82
Figure 44 – WiseCORP – Explicit Demand Response	83





List of Tables

Table 1 – Test case specification template8
Table 2 – Definition of terms12
Table 3 – Test groups13
Table 4 – Test case specification template17
Table 5 – Characteristics of the lab-testing platform servers
Table 6 – Server directions
Table 7 – Data items tracked by Real-time monitor in WiseCOOP application
Table 8 – WiseCOOP – Spark jobs of the KPI engine
Table 9 – Summary of demand-based cluster centres on lab-testing data
Table 10 – Summary of production-based cluster centres on lab-testing data
Table 11 – Summary of CO ₂ -based cluster centres on lab-testing data
Table 1 <mark>2</mark> – Summary of economic cost-based cluster centres on lab-testing data
Table 13 – Capture of a conversation between WiseCORP and WiseHOME
Table 14 - Elicitation of demand flexibility potential per building
Table 15 – Request from WiseCOOP for specific asset demand modulation as a result of a DR event34
Table 16 – WiseCOOP UI – List of possible status for explicit demand response campaigns
T <mark>ab</mark> le 17 – Log of command dispatched by Asset dis <mark>patch</mark> er via MQTT
T <mark>ab</mark> le 18 – Data items tracked by Real-time monitor in WiseCORP application
Table 19 – WiseCORP – Spark jobs of the KPI engine
Table 20 – Sample response from WiseCORP to the demand flexibility request from WiseCOOP (as shown in Table 14)
Table 21 – WiseCORP – Data inputs of the Energy Usage optimizer
Table <mark>2</mark> 2 – WiseCORP – Data outputs of the Energy <mark>Usag</mark> e optimizer
Table 23 – Acronyms list





EXECUTIVE SUMMARY

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 Smart Grid actors interfacing consumers and prosumers (particularly focused on domestic and small businesses), supporting them in their roles of energy retailers, third-party aggregators, local communities and cooperatives. The main goal of the solution is help-ing 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.

Taking into consideration the aim of these tools, the respective architecture and modules were designed also having in mind the design of the other WiseGRID tools. Furthermore, having in mind the process of developing software, this document explains the implementation and lab-testing activities performed for assuring the quality of the tools previously designed.

For that purpose, it was first needed to align the terminology to be used during the lab-testing phase in order to assure that all the partners involved in this stage, work in the same direction and there are no misunderstandings. Then, the test plan was established which basically consists in the following 5 steps:

- 1) Review the project requirements and use cases
- 2) Define the features to be tested from those and classify them into test groups
- 3) Detail test cases for validation of named features
- 4) Execute the test cases
- 5) Document the test protocols

Moreover, for assuring the coherence and the easy understanding of each test, the following template was created, which summarizes the most important information to be shown and the features to be tested.

Table 1 – Test case specification template				
Name	The test case code and name which is unique to the project.			
Module under test	The devices or systems	under test	Resp.	Main partne <mark>r</mark> responsible for the test
Module	The requirement, use c	ase, or certificat	ion rule whicl	h is validated by the
requirement	test case			·
Test environment	List of elements needed	d for the test exe	cution	
Features to be tested	List of features to be te	ested		
Features not to be tested	Optional			
Preparation	Short list of steps needed for preparing the test environment for test execution			

Table 1 – Test case specification template





Dependencies (Optional) List of test case codes defining test cases which need to passed before the test case at hand can be started	
Steps Testing procedures	
Pass criteria	Expected (measurable) results, allowing to unambiguously judge if the test is passed or not passed (i.e. the product requirement was validated or not validated)
-	(Optional) Conditions under which continuation of the test is considered pointless because testing results would be invalid
Results	(Optional) Short list of results

Anyway, before starting the lab-testing phase, the implementation activities were performed in order to correctly integrate all the modules of the tools.

The following tables below list all the test cases and test results covered in the document at hand:

est case code	Table 2 – WiseCOOP tests results Test case	
RTM001	Read smart meter data from IOP	\checkmark
RTM002	Store smart meter data to long-term DB	✓
KPI001	Individual energy delta calculation	\checkmark
KPI002	Portfolio data aggregation	✓
KPI003	Portfolio profiling, demand-related behaviour	\checkmark
KPI004	Portfolio profiling, production-related behaviour	✓
KPI005	Portfolio profiling, monthly CO ₂ emissions	\checkmark
KPI006	Portfolio profiling, monthly cost	~
KPI007	Calculation of average profiles per clusterization and cluster	
FOR001	Demand/production forecasting training	
FOR002	Demand/Production forecasting	
FOR003	Request message parsing test of WiseCOOP forecast module	\checkmark
FOR004	Forecast response message generation test of WiseCOOP forecast module	\checkmark
FOR005	Forecast is periodically triggered	\checkmark
FOR006	Forecast results are saved to operational DB	\checkmark
TDC001	Create a tariff	\checkmark
TDC002	Edit existing tariff	\checkmark
DRF001	Obtain portfolio demand forecast	\checkmark
DRF002	Obtain generation forecast	
DRF003	Price signal calculation	
DRF004	Dispatch price signal	
DRF005	Elicitation of demand flexibility per building	
DRF006	Estimate & dispatch device commands	





Table 3 – WiseCORP tests results

Test case code	Test case	
RTM001	Read smart meter data from IOP	✓
RTM002	Read sensor data from IOP	
RTM003	Read battery data from IOP	\checkmark
RTM004	Read HVAC data from IOP	\checkmark
RTM005	Store smart meter data to Long-term DB	\checkmark
RTM006	Store sensor data to Long-term DB	\checkmark
RTM007	Store battery data to long-term DB	\checkmark
RTM008	Store HVAC data to Long-term DB	\checkmark
KPI001	Associated CO ₂ emissions	\checkmark
КРІОО2	Associated economic costs	\checkmark
KPI003	Distribution of demand per tariff period	\checkmark
КРІ <mark>00</mark> 4	Calculation of indicators per building. Monthly economic costs	✓
KPI005	Calculation of indicators per building. CO ₂ emissions	\checkmark
<mark>КР</mark> 1006	Calculation of indicators per building. Total demand	✓
KPI007	Calculation of indicators per building. Total production	\checkmark
KPI008	Calculation of indicators per building. Self-consumption ratio	~
FOR001	Demand/production forecasting training	\checkmark
FOR002	Demand/Production forecasting	
FOR003	Request message parsing test of WiseCORP forecast module	
FOR004	Forecast response message generation test of WiseCORP forecast module	
FOR005	Forecast is periodically triggered	
FOR006	Forecast results are saved to operational DB	
TC001	Create simulated bill for building	\checkmark
EUO001	Unit testing	✓ ✓
EUO002	Produce day-ahead optimum schedule for assets	
DRF001	Estimate occupant thermal/visual comfort profile	
DRF002	Calculate human-centric demand flexibility of building	
DRF003	Receive request to activate demand flexibility	
DRF004	Asset schedule modification	
AD001	Load schedule from operational database	
AD002	Read current asset status from operational database	
AD003	Detect deviation from schedule	
AD004	Trigger asset setpoint	

All these activities have been supported by the setup of a virtual environment which replicates to the extent possible the conditions that will be found in the deployment of the applications in the different pilot sites. The lab-testing platform consists of a couple of virtual machines running in the VMWare vSphere infrastructure of ETRA I+D.





1 INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

The purpose of this document is to summarise the results from Tasks 7.2 "WiseCOOP and WiseCORP implementation" and 7.3 "WiseCOOP and WiseCORP lab-testing and refinement". In these tasks, the WiseCOOP and WiseCORP designs and developments within Task 7.1 "WiseCOOP and WiseCORP Apps design" is verified in a controlled environment before deployment at the pilot sites.

1.2 SCOPE OF THE DOCUMENT

This deliverable covers the development of the WiseCOOP and WiseCORP tools during their implementation and lab-testing phases, including an overview of the designed architectures in order to make the reader aware about the previous work performed. In this way, this document describes the test cases that were performed to validate the WiseCOOP and WiseCORP frameworks before deploying them at the pilot sites.

1.3 STRUCTURE OF THE DOCUMENT

The document starts with the establishment of the lab-testing basis that will be used for the evaluation of the test cases. Then, the document continues with the explanation of the implementation of the different WiseCOOP modules. After this implementation starts the lab-testing phase of WiseCOOP, which describes the different tests done to evaluate the performance of the tool and their results. After these sections there are similar ones for WiseCORP. Finally, a section for extracting the conclusions of these tasks closes the documents and settles the next steps to be followed.





2 LAB-TESTING APPROACH

The lab-testing approach that has been followed for these tools is the same one that was followed in the NOBEL GRID project. This methodology has demonstrated to be successful for this kind of projects so it has been properly studied and used taking into account the particularities of the WiseGRID project.

2.1 DEFINITION OF TERMS

In order to provide a common methodology for testing the WiseCOOP and WiseCORP tools, a common definition of terms was used. The following definitions were developed considering the state of the art in software, smart grid and system integration testing, especially with respect to the IEEE 829 Standard for software test documentation [1] [2] [3].

Town	Table 4 – Definition of terms Definition		
Term Device under test	a product or software which is verified by a certain test case. It is part of the test		
Expected results	environment a description of the status of the test environment after a test case was carried out and pass criteria have been met		
Features (not) to be tested	a list of product requirements or specifications which are (not) covered by a certain test case		
Pass/fail criteria	a definition of how to judge or measure if a product under test conforms to specifications and requirements that shall be validated by a certain test case		
Retesting	re-execution of a test case that previously returned a "fail" result, to evaluate the effectiveness of intervening correction actions		
Subsystem acceptance criteria	conditions to be fulfilled by a subsystem for including it into the system integration test. Conditions should include the availability of testing protocols for standalone subsystem tests. Also, subsystems should have similar level of maturity		
Suspension criteria	a description of conditions which indicate that the test was carried out incorrectly or that any situation was produced which renders the testing results unusable, making test continuation pointless and requiring the test to be halted and restarted		
System integration test	a test designed to verify that a system made up of two or more interacting products (subsystems) conforms to system-wide specifications and requirements. The device under test is the system itself. It is specially designed for finding inconsistencies which emerge only through the subsystem interaction. The system integration test plan may define partial system integration tests which allow for adding subsystems subsequently		
(System integration test) Level	The number of system layers which are included in a system integration test case minus one		
System layer	a group of one or more subsystems which is defined prior to the system integration test. According to group definition for a given system should be used for all system integration test cases		
Testing	set of activities conducted to facilitate discovery, validation and/or evaluation of properties of one or more test WiseGRID components		
Test analysis	elaboration about why a test result emerged. It may also include a conclusion about what the test result implies for the future work		
Test case	a collection of features (not) to be tested, testing procedures, and pass/fail criteria used for testing a system or device under test. Test cases may refer to specific types of product requirements, e.g. the function, reliability, stability, safety, or vulnerability. Test cases may be applied to different test environments, e.g. the same test case may be applied to different pilot sites		
Test case code	an identifier for a test case which is unique throughout the project, e.g. KPI001		





Test case	documentation of one or more test cases	
specification		
Test coverage	a list of product requirements or specifications which are verified by a test plan	
Test data	data created or selected which is needed for executing one or more test cases. It may	
	be defined in the test case specification	
Test environment	a list of all elements (software, hardware, information, external conditions) needed to	
	carry out a test case, including the device or system under test and all elements needed	
	to judge the test outcome	
Test environment	a list of actions needed for establishing and maintaining a required test environment	
set-up process		
Test execution	the actions needed to carry out the testing procedures for a given test case	
Test group	a collection of test cases which share at least one defined criterion. E.g. all test cases	
	which relate to cybersecurity testing might be defined to make up a test group	
Test method	a general definition of testing procedures and test environment for a test plan	
Test plan	a strategy or list of tasks used to verify that a product conforms to design specifications	
	and product requirements	
Test preparation	a definition of steps which are needed to prepare a test environment for test execution	
Testing procedures	a specific list of steps which are needed to carry out a test case	
Test protocol	a summary of the test results of all test cases defined in a test plan. It may also contain	
	the test analysis for said test cases	
Test requirements	a definition stating the status of the test environment which is needed for carrying out	
	a specific test case or a test group. Ideally, it is also stated how it can be checked if the	
	test environment is ready for test execution	
T <mark>es</mark> t responsibilities	a definition stating which persons or organizations are needed for the test. It may also	
	include an assignment of tasks to those people or organizations	
Test result	an indication of whether a specific test case has passed or failed. May also include any	
	data that has been obtained through execution of the test case	

2.2 TEST PLAN

The test plan used for testing both applications consists of the following steps:

- 1) Review the project requirements and use cases
- 2) Define the features to be tested from those and classify them into test groups
- 3) Detail test cases for validation of named features
- 4) Execute the test cases
- 5) **Document the test protocols**

2.3 FEATURES TO BE TESTED

The partners have defined a number of features to be tested. Those features are based on the project requirements defined by the consortium and use cases for the WiseGRID project.

The features to be tested were classified into different test groups which are defined in Table 5. The table also defines which criteria are shared by the test cases within the test groups.

Test group	Common Criteria		
Visualisation and analysis	The feature under test provides visualization and/or analysis of the data collected		
Control	The feature under test provides control of assets.		





Compliance	The feature under test relates to compliance of the tool with the USEF standard or other standards.	
Functionality	The feature under test is a complex function provided by a combination of software and communication between multiple WiseGRID subsystems.	
Communication	The feature under test is basic data transmission between two communication endpoints, one being the tool.	
Robustness and stability	The feature under test is related to fault tolerant and stability.	
Cyber Security	The feature under test mitigates vulnerabilities of the software or malicious attacks aimed at it.	

Table 6 and Table 7 show the features tested as defined for testing the WiseCOOP and WiseCORP. Each feature shall define one test case.

Table 6 – WiseCOOP's features tested

Test	Feature to be tested	Test group
case		
code		
RTM001	Data from SMX is properly collected in the operational database of WiseCOOP	Communication
RTM002	Data from SMX is properly collected in the long-term database of WiseCOOP (big data)	Communication
KP1001	 Smart meters provide information of the total accumulated energy demand/production. The system therefore needs to calculate the energy deltas across consecutive readings in order to properly monitor the energy demand/production profiles. Three different aggregations of the deltas are considered: quarterly, hourly and daily. 	Visualisation and analysis
KP1002	An overview of the demand and production metrics for the whole portfolio is required	Visualisation and analysis
KP1003	All portfolio members get classified based on the distribution of the energy demand in the following time slots: - Working days, 08h-18h - Working days, 18h-08h - Non-working days, 08h-18h Non-working days, 18h-08h	Visualisation and analysis
KP1004	All portfolio members get classified based on their daily production	Visualisation and analysis
KPI005	All portfolio members get classified based on the monthly equivalent CO2 emissions	Visualisation and analysis
KP1006	All portfolio members get classified based on the economic cost associated to their demand	Visualisation and analysis
KP1007	For each group of each clusterization, the <i>average member</i> is computed, in order to enable comparison of individual behaviour with the group of similar individuals	Visualisation and analysis
FOR001	WiseCOOP forecast module is trained	Functionality
FOR002	WiseCOOP forecast module performs demand/production forecasting training	Functionality
FOR003	Performance of WiseCOOP forecast module, at parsing forecast queries.	Functionality
FOR004	Performance of WiseCOOP forecast module, at generating and submitting	Functionality





	the demand forecast response.	
FOR005	WiseCOOP periodically posts a demand and a production forecast request per bus to the corresponding queue of the internal ESB	Functionality
FOR006	WiseCOOP receives the results of the forecast module, formats them following the same format used to store real-time data, and stores the in the operational database	Functionality
TDC001	WiseCOOP facilitates to retailers the ability to define tariffs from the UI	Functionality
TDC002	WiseCOOP facilitates to retailers the ability to define tariffs from the UI	Functionality
DRF001	Retrieval of the aggregated day-ahead demand of the retailer portfolio from the long-term DB of the WiseCOOP tool	Communication
DRF002	Retrieval of the day-ahead generation forecast of the retailer portfolio from the long-term DB of the WiseCOOP tool	Communication
DRF003	Estimation of the day-ahead 24-hour retail electricity price forecast based on the availability of renewable generation	Visualisation and analysis
DRF004	Dispatching of the price signal (encoded in the format specified in D10.2) to a queue in the IOP so that other WiseGRID products can receive it.	Communication
DRF005	Elicitation and collection of the demand flexibility potential from the all the buildings that belong to the portfolio of the WiseCOOP operator	Visualisation and analysis
DRF006	Breakdown of demand flexibility requested from the DSO into the optimal flexibility per building device, based on the information retrieved about the demand flexibility potential.	Control
Test case code	Feature to be tested	Test group
case code	Feature to be tested Data from SMX is properly collected in the operational database of WiseCORP	Test group Communication
case code RTM001	Data from SMX is properly collected in the operational database of	r
case code RTM001 RTM002	Data from SMX is properly collected in the operational database of WiseCORP Data from sensor Wrapper is properly collected in the operational database	Communication
case	Data from SMX is properly collected in the operational database of WiseCORP Data from sensor Wrapper is properly collected in the operational database of WiseCORP Data from battery is properly collected in the operational database of	Communication
case code RTM001 RTM002 RTM003	Data from SMX is properly collected in the operational database of WiseCORP Data from sensor Wrapper is properly collected in the operational database of WiseCORP Data from battery is properly collected in the operational database of WiseCORP Data from HVAC is properly collected in the operational database of WiseCORP Data from SMX is properly collected in the long-term database of WiseCORP	Communication Communication Communication
case code RTM001 RTM002 RTM003 RTM004 RTM005	Data from SMX is properly collected in the operational database of WiseCORP Data from sensor Wrapper is properly collected in the operational database of WiseCORP Data from battery is properly collected in the operational database of WiseCORP Data from HVAC is properly collected in the operational database of WiseCORP Data from SMX is properly collected in the long-term database of WiseCORP (big data) Data from sensor is properly collected in the long-term database of WiseCORP (big data)	Communication Communication Communication Communication Communication Communication
case code RTM001 RTM002 RTM003 RTM005 RTM006	Data from SMX is properly collected in the operational database of WiseCORP Data from sensor Wrapper is properly collected in the operational database of WiseCORP Data from battery is properly collected in the operational database of WiseCORP Data from HVAC is properly collected in the operational database of WiseCORP Data from SMX is properly collected in the long-term database of WiseCORP (big data) Data from sensor is properly collected in the long-term database of WiseCORP (big data) Data from battery is properly collected in the long-term database of WiseCORP (big data)	Communication Communication Communication Communication Communication
case code RTM001 RTM002 RTM003 RTM004	Data from SMX is properly collected in the operational database of WiseCORP Data from sensor Wrapper is properly collected in the operational database of WiseCORP Data from battery is properly collected in the operational database of WiseCORP Data from HVAC is properly collected in the operational database of WiseCORP Data from SMX is properly collected in the long-term database of WiseCORP (big data) Data from sensor is properly collected in the long-term database of WiseCORP (big data) Data from battery is properly collected in the long-term database of WiseCORP (big data)	Communication Communication Communication Communication Communication Communication
case code RTM001 RTM002 RTM003 RTM004 RTM005 RTM006	Data from SMX is properly collected in the operational database of WiseCORP Data from sensor Wrapper is properly collected in the operational database of WiseCORP Data from battery is properly collected in the operational database of WiseCORP Data from HVAC is properly collected in the operational database of WiseCORP Data from SMX is properly collected in the long-term database of WiseCORP (big data) Data from sensor is properly collected in the long-term database of WiseCORP (big data) Data from battery is properly collected in the long-term database of WiseCORP (big data) Data from battery is properly collected in the long-term database of WiseCORP (big data)	Communication Communication Communication Communication Communication Communication

 associated economic costs
 KPI003

 This module must crosscheck the energy demand of the building with the periods of the contracted tariff, giving an overview of the distribution of
 Visualisation and analysis





	the demand over a period of time (e.g. monthly)	
KP1004	This module analyses historic information available in the long-term database to provide insights on the economic costs faced by the measured	Visualisation and analysis
KPI005	facilities This module analyses historic information available in the long-term	Visualisation and analysis
KF1005	database to provide insights on the equivalent CO_2 emissions produced by the measured facilities.	
KP1006	This module analyses historic information available in the long-term database to provide insights on the total energy demand of the measured facilities.	Visualisation and analysis
KP1007	This module analyses historic information available in the long-term database to provide insights on the total energy production of the measured facilities.	Visualisation and analysis
KP1008	This module analyses historic information available in the long-term database to provide insights on the total self-consumption and the energy production surplus of the measured facilities	Visualisation and analysis
FOR001	WiseCORP forecast module is trained	Functionality
FOR002	WiseCORP forecast module performs demand/production forecasting training	Functionality
FOR003	Performance of WiseCORP forecast module, at parsing forecast queries.	Functionality
FOR004	Performance of WiseCORP forecast module, at generating and submitting the demand forecast response.	Functionality
FOR005	WiseCORP periodically posts a demand and a production forecast request per bus to the corresponding queue of the internal ESB	Functionality
FOR006	WiseCORP receives the results of the forecast module, formats them following the same format used to store real-time data, and stores the in the operational database	Functionality
TC001	WiseCORP facilitates to ESCOs the ability to simulate bills according to historical demand data and tariff definitions	Visualisation and analysis
EUO001	The energy usage optimizer must calculate the optimum 24-hours long schedule for the given assets, considering usage calendar and energy price.	Functionality
EUO002	Upon completion of the execution of the energy usage optimizer module, results are stored in the operational database of WiseCORP	Functionality
DRF001	Generation of the comfort profile for the individual occupants regarding thermal and visual comfort	Visualisation and analysis
DRF002	Estimation of demand flexibility time series	Functionality
DRF003	Reception of the appropriate message from WiseCOOP specifying the detailed break-down of demand flexibility per device to be activated	Communication
DRF004	Estimation of optimal setpoint per device and dispatch to the "asset dispatcher" component that sends the setpoints to the device wrappers.	Visualisation and analysis
AD001	The asset dispatcher module can read from operational database all necessary information about the schedule of the controllable assets	Communication
AD002	The asset dispatcher module can read from operational database all necessary information about the current setpoint executed by the controllable assets	Communication
AD003	Given a point in time when current setpoint and scheduled setpoint differs for a controllable asset, the asset dispatcher must be able to detect the incoherence	Control
AD004	Given a point in time when current setpoint and scheduled setpoint differs for a controllable asset, the detection of this incoherence must result in the publication of a command to the controllable asset to set the appropriate setpoint	Control





2.4 TEST CASES SPECIFICATION

Name	The test case code and name which is unique to the project.		
Module under test	The devices or systems under test Resp. Main partner responsible for the test		responsible for
Module requirement	The requirement, use case, or certification rule which is validated by the test case		
Test environment	List of elements needed for the test exe	cution	
Features to be tested	List of features to be te <mark>ste</mark> d		
Features not to be tested	Optional		
Preparation	Short list of steps needed for preparin execution	g the test er	nvironment for test
Dependencies	(Optional) List of test <mark>case c</mark> odes defining test cases which need to be passed before the test case at hand can be started		
Steps	Testing procedures		
Pass criteria	Expected (measurable) results, allowing to unambiguously judge if the test is passed or not passed (i.e. the product requirement was validated or not validated)		
Suspension criteria	(Optional) Conditions under which continuation of the test is considered pointless because testing results would be invalid		
Results	(Optional) Short list of <mark>resu</mark> lts		

Table 8 – Test case specification template

2.5 LAB-TESTING PLATFORM DETAILS

The implementation and lab-testing phases of the development of WiseCOOP and WiseCORP applications has been supported by the setup of a virtual environment which replicates up to the extent possible the conditions that will be found in the deployment of the applications in the different pilot sites.

The lab-testing platform consists of a couple of virtual machines running in the VMWare vSphere infrastructure of ETRA I+D.

Table 9 – Characteristics of the lab-testing platform servers				
Characteristics	wisegridpre.lab.id	wintest.lab.id		
OS	Ubuntu Server 16.04	Microsoft Windows Server 2012		
CPU	2 CPU	1 CPU		
Memory	8GB	5GB		
Hard disk	50GB	35GB		





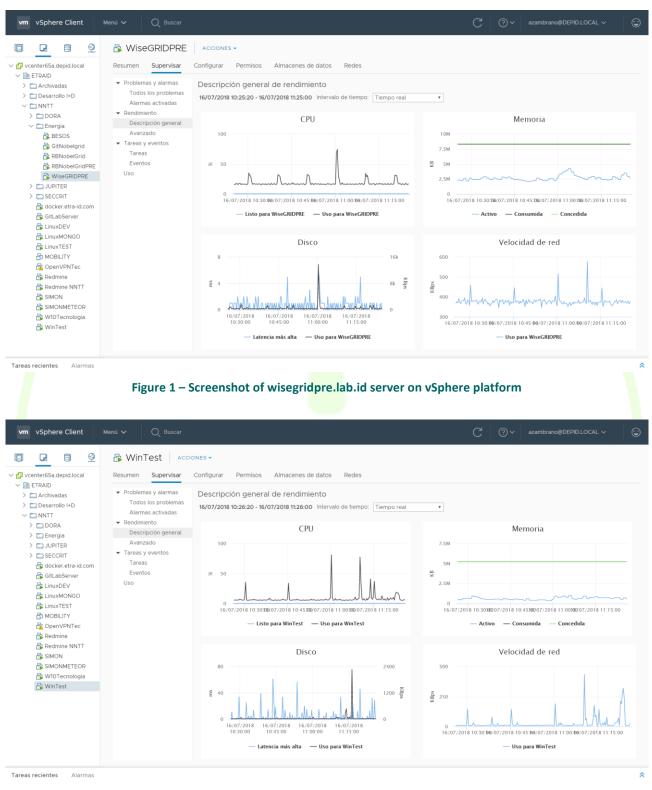


Figure 2 – Screenshot of wisegridpre.lab.id server on vSphere platform

Following the architecture of the tools, a number of common services have been installed on those servers and made accessible to the rest of the partners taking part in the project via public URL (protected with the





corresponding access credentials). Data protection principles have been also considered in line with deliverable D3.2.

	Table 10 – Server directions		
Server	Module	URL	
wise-	Internal ESB (RabbitMQ).	AMQP: amqp://etra-id.com	
gridpre.lab.id	Two virtual hosts:	MQTT: tcp://etra-id.com:1883	
	- /wisecoop		
	- /wisecorp		
	Database server (MongoDB)	mongodb://etra-id.com	
	Big data processing (Spark Server – 1 mast 1 server)	er + [internal access only]	
	WiseCOOP User Interface	https://wisecoop.etra-id.com	
	WiseCORP User Interface	https://wisecorp.etra-id.com	
winte <mark>st</mark> .lab.id	Maintenance management module	http://windeptec.etra-id.com/GiManWise- GRIDWSRest	

The described configuration, together with the lab-testing instance of the WiseGRID IOP, allowed the different partners, in charge of the development and testing the different specific modules within each one of the applications, to connect each parts to the others and perform the necessary integration tests to make sure that all modules work together as expected. Particularly, the following points have been covered:

- Connection of real and simulated assets to the WiseGRID IOP, in order to test the data ingestion of the applications;
- Intercommunication among the different modules, by connecting them all via Internet to the internal ESB of each application;
- Hosting most of the modules composing the applications;
- Testing of the KPI engine implementation on a local instance of Spark Server (in parallel to the development of the Big Data Platform);
- Testing technologies that will facilitate deployment of the modules in the pilot sites (Docker and Docker-compose);
- Access to preliminary versions of the User Interfaces of the applications.





ETD						
	RA I+D					
wisegrid	dpre.lab.id					
Spark	mongoDB.					
Radoitivic				$ \longrightarrow $		Engineering Ingegneria Informatica
			()		
		•		,) .		
wintes	st.lab.id		~	\mathcal{V}		WiseGRID IOP
			$\overline{\langle}$			
u ás	eorid					
h						
vm ware [.]						
			REAL & S	IMULATED ASSETS		
					•	
		Figure 3 – I	Lab-testing p	latform core mod	ules overview	
t [#] ☆ Default ~ STAC			Lab-testing p	latform core mod	ules overview	و
	cks√ catalog√ infras			latform core mod	ules overview	Ω ≣Active AddContainer II :
st: wisegridpre.lab.id	cks√ catalog√ infras			atform core mod	ules overview	
st: wisegridpre.lab.id : !22.168.168.59	cks√ catalog√ infras	CPU 200% 120% 120%	NU 19 ADIU	O System O User	Memory 778 GB 6.22 GB 4.67 GB	Active Add Container
st: wisegridpre.lab.id : :2.168.168.59 : U: tal: 2x2.1 GHz () L emory:	IMFRAS	CPU 200% 120% 30% 120% 40%		O System O User	Memory 7.78 GB 6.22 GB 4.67 GB 3.11 GB 1.56 GB	Active Add Container
wisegridpre.lab.id : 2.168.168.59 PU: tal: 2x2.1 GHz ① emory: tal: 7.78 GiB	oksv catalogv infras	CPU 200% 120% 80%	NU 19 ADIU	O System O User	Memory 7.78GB 6.22GB 4.67GB 3.11GB	Active Add Container
• •	IMFRAS	CPU 200% 120% 30% 120% 40%	NU 19 ADIU	O System O User	Memory 7.78 GB 6.22 GB 4.67 GB 3.11 GB 1.56 GB	Active Add Container
st: wisegridpre.lab.id ::	INFRAS	TRUCTURE ADMIN	NU 19 ADIU	O System O User	Memory 7.78 GB 6.22 GB 4.67 GB 3.11 GB 1.56 GB 0.8 Storage 173 Mips	Active Add Container II :
• • • •	INFRAS	CPU 200% 120% 120% 120% 40% 40% 40% 40% 40% 40% 40% 4	NVI ARIV	O System O User	Memory 7.78 GB 6.22 GB 4.67 GB 3.11 GB 1.56 GB 08 Storage 123 Maps 130 Maps 130 Maps	Active Add Container II :
st: wisegridpre.lab.id 2.168.168.59 20: tal: 2x.2.1 GHz () L emory: tal: 7.78 GiB L tal: 7.78 GiB L tal: 2.78 GiB L t	INFRAS	CPU 2005 1605 1205 405 405 405 405 405 405 405 405 405 4	NVI ARIV	O System O User	Memory 7.78 CBB 4.67 CBB 4.67 CBB 3.11 CBB 1.56 CBB 0.8 Storage 173 Mbps 130 Mb	Active Add Container II :
t: wisegridpre.lab.id 2.168.168.59 U: tal: 2x.2.1 GHz 0 L amory: tal: 7.78 GiB L brage: 1.1 GiB 0 L sovider: stom rnel: 1.0 ko	INFRAS	CPU 200% 160% 40% 120% 40% 90% 40% 40% 40% 54.8 Mbps 43.9 Mbps 32.9 Mbps 21.9 Mbps 13 Mbps 13 Mbps 13 Mbps 13 Mbps		O System O User	Memory 7/8 GB 6.22 GB 4.67 GB 3.15 GB 1.56 GB 08 Storage 133 Maps 134 Maps 934 Maps	CUsed
st: wisegridpre.lab.id 2.168.168.59 20: tal: 2x2.1 GHz ① L emory: tal: 7.78 GiB L orage: 11 GIB ① L ovider: stom rmel: to0: cker: t03.1-ce	INFRAS	CPU 200% 160% 40% 120% 40% 90% 40% 40% 40% 54.8 Mbps 43.9 Mbps 32.9 Mbps 21.9 Mbps 13 Mbps 13 Mbps 13 Mbps 13 Mbps	NVI ARIV	O System O User	Memory 7/8 GB 6.22 GB 4.67 GB 3.15 GB 1.56 GB 08 Storage 133 Maps 134 Maps 934 Maps	CUsed
t: wisegridpre.lab.id 2.168.168.59 U: tal: 2x.2.1 GHz ① L amory: tal: 7.78 GiB L brage: 1:1 GiB ① L stom rnet: t0 cker: 0.03.1-ce 8:	INFRAS	CPU 200% 160% 40% 120% 40% 90% 40% 40% 40% 54.8 Mbps 43.9 Mbps 32.9 Mbps 21.9 Mbps 13 Mbps 13 Mbps 13 Mbps 13 Mbps		O System O User	Memory 7/8 GB 6.22 GB 4.67 GB 3.15 GB 1.56 GB 08 Storage 133 Maps 134 Maps 934 Maps	CUsed
st: wisegridpre.lab.id 2.168.168.59 201 tal: 24.21 GHz () L emory: tal: 7.78 GiB L tal: 7.78 GiB L tal: 60 () L tal: 7.78 GiB L tal:	INFRAS	TRUCTURE > ADM!	NV B APIV	O System O User	Memory 7/8 GB 6.22 GB 4.67 GB 3.15 GB 1.56 GB 08 Storage 133 Maps 134 Maps 934 Maps	CUsed
st: wisegridpre.lab.id 2.168.168.59 201 tal: 24.21 GHz () L emory: tal: 7.78 GiB L tal: 7.78 GiB L tal: 60 () L tal: 7.78 GiB L tal:	INFRAS	TRUCTURE ADMI CPU 2005 12096 12096 12096 0296 0296 0296 0296 0296 0296 0296	NV 1 APIV	O System O User	Memory 7/8 GB 6.22 GB 4.67 GB 3.15 GB 1.56 GB 08 Storage 133 Maps 134 Maps 934 Maps	Active Add Container Used OUsed OWrite Read
st: wisegridpre.lab.id : :2.168.168.59 DU: tal: 2x2.1 GHz ① L emory: tal: 7.78 GIB L orage: t1 GIB ① L ovider: ustom strel: 4.0 oxcher: b.03.1-ce s:	INFRAS	TRUCTURE ADMI	NV 1 APIV	O System O User O Transmit O Receive O Transmit O Receive O Transmit O Receive O Command) C Stats docker.settra-id.co. n/0 rancher/setver (/u n/0	Memory 7/8 GB 6.22 GB 4.67 GB 3.15 GB 1.56 GB 08 Storage 133 Maps 134 Maps 934 Maps	Active Add Container Used OUsed OWrite Read
st: wisegridpre.lab.id s2.168.168.59 PU: tda: 2x2.1 GHz () L emory: tda: 7.78 GIB L orage: r1GIB () L ovider: ustom arnel: 4.0 scker: 303.1-ce S:	INFRAS	CPU 200% 1200% 120% 1200% 120% 1200% 120% 1200% 120% 1200% 1100 <td>NUT APIU Labels Storage Name Q IPAddres common_obliS2cL, None common_rancher, None healthcheck-heal, 10.42234 jpsec-cni-driver-1 None</td> <td>CSystem O User O User O Transmit O Receive O Transmit</td> <td>Memory 7/8 GB 6.22 GB 4.67 GB 3.15 GB 1.56 GB 08 Storage 133 Maps 134 Maps 934 Maps</td> <td>Active Add Container Used Used OUted OWrite Read OWrite O I I I I I I I I I I I I I I I I I I</td>	NUT APIU Labels Storage Name Q IPAddres common_obliS2cL, None common_rancher, None healthcheck-heal, 10.42234 jpsec-cni-driver-1 None	CSystem O User O User O Transmit O Receive O Transmit	Memory 7/8 GB 6.22 GB 4.67 GB 3.15 GB 1.56 GB 08 Storage 133 Maps 134 Maps 934 Maps	Active Add Container Used Used OUted OWrite Read OWrite O I I I I I I I I I I I I I I I I I I
st: wisegridpre.lab.id 22.168.168.59 PU: ptal: 2x2.1 GHz ① L lemory: stal: 7.78 GiB L torage: 7.1 GiB ① L trovider: ustom ernel: 4.0 ocker: 8.03.1-ce S:	INFRAS	CPU 200% 120% 120% 120% 120% 120% 120% 120	NUT APIU Labels Storage Name Q IPAddres common_oblis2cL None common_rancher, None healthcheck-heal 10.42234 [psec-ipsec-1 10.42128	CSystem Ouser Constantial Constantian Con	Memory 7/8 GB 6.22 GB 4.67 GB 3.15 GB 1.56 GB 08 Storage 133 Maps 134 Maps 934 Maps	Active Add Container OUsed OWrite Read OWrite Read O C C C C C C C C C C C C
st: wisegridpre.lab.id 22.168.168.59 PU: ptal: 2x2.1 GHz ① L lemory: stal: 7.78 GiB L torage: 7.1 GiB ① L trovider: ustom ernel: 4.0 ocker: 8.03.1-ce S:	INFRAS	CPU 200% 120% 200% 120% 200% 120% 20% 40% 20% 40% 20% 00% 20% 21.9 Mbps 21.9 Mbps 018ps 21.9	NUT APIU Labels Storage Name Q IP Addres common_obiS2cL. None common_obiS2cL. None common_rancher., None healthcheck-heal. 10.42.234 ipsec-ipsec-1 0.42.128 ipsec-ipsec-con None	CSystem O User OSystem O User OTransmit O Receive OTransmit O Receive odder.etra-id.co n/0 rancher/nerve(n n/0 rancher/nerve(n n/0 rancher/nerve(n	Memory 7/8 GB 6.22 GB 4.67 GB 3.15 GB 1.56 GB 08 Storage 133 Maps 134 Maps 934 Maps	Active Add Container O Used OWrite Read OWrite C C C C C C C C C C C C C
wisegridpre.lab.id >: 92.168.168.59 PU: otal: 2x2.1 GHz () L temory: otal: 7.78 GiB L torage:	INFRAS	CPU 200% 120% 120% 120% 120% 120% 120% 120	NUT APIU Labels Storage Name Q IPAddres common_oblis2cL None common_rancher, None healthcheck-heal 10.42234 [psec-ipsec-1 10.42128	CSystem Ouser Constantial Constantian Con	Memory 7/8 GB 6.22 GB 4.67 GB 3.15 GB 1.56 GB 08 Storage 133 Maps 134 Maps 934 Maps	Active Add Container OUsed OWrite Read OWrite Read O C C C C C C C C C C C C
st: wisegridpre.lab.id 22.168.168.59 PU: ptal: 2x2.1 GHz ① L lemory: stal: 7.78 GiB L torage: 7.1 GiB ① L trovider: ustom ernel: 4.0 ocker: 8.03.1-ce S:	INFRAS	CPU 200% 120% 120% 120% 120% 120% 120% 120	NUT APIU Labels Storage Name Q IPAddres common_oblis2cL None common_rancher, None healthcheck-heal 10.42234 [psec-ipsec-1 10.42128	CSystem O User OSystem O User OTransmit O Receive dodersetra-doco. n/n rancher/netvol.13. n/n rancher/netvol.13. n/n	Memory 7/8 GB 6.22 GB 4.67 GB 3.15 GB 1.56 GB 08 Storage 133 Maps 134 Maps 934 Maps	Active Add Container OUsed OWrite Read OWrite Read O C C C C C C C C C C C C

Figure 4 – Dashboard of wisegridpre.lab.id, all modules installed as Docker containers





3 WISECOOP APPLICATION

3.1 IMPLEMENTATION

3.1.1 Architecture overview

The architecture of the WiseCOOP application finally implemented does not differ significantly from the architecture presented in the previous deliverable D7.1 WiseCOOP and WiseCORP Apps Design [4]. The architecture is summarized in this section for completeness of this document.

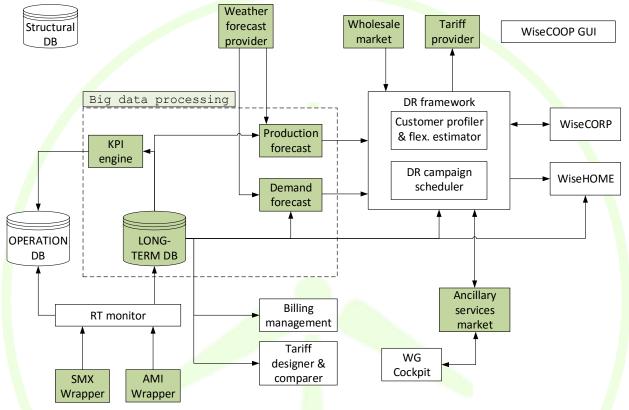


Figure 5 – Overview of interactions among the modules of the WiseCOOP application

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. <u>Publication of data from Wrappers to the WiseGRID IOP Message Broker</u>. 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;
- Subscription to data flows of interest. In the case of the WiseCOOP application, 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. <u>Store data for further analysis</u>. 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 include 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 -, allowing 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 are 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*, *Wholesale Market* and *Tariff Provider* modules, as well as the *Big Data platform* that will support the long-term storage and analysis. In addition, the *WiseCOOP User Interface* is included in this category, providing web-based access to the information and functionalities provided by the other modules.

3.1.2 Back-office modules

3.1.2.1 Internal Enterprise Service Bus

As depicted in the architectural overview, the application is actually composed of several modules with welldefined functionalities, which collaborate with each other in order to enable the high-level functionalities of the application. In order to facilitate communication among the modules, it was decided during the design





phase to incorporate an internal Enterprise Service Bus to the application. The selected technology for deploying this communication bus is RabbitMQ, since it covers most of the requirements settled during the design phase and exposed in D7.1.

RabbitMQ has been configured with the following main characteristics:

- Credential-based access control: one credential has been given to each partner requiring access.
- Protocols enabled: AMQP, MQTT and HTTP.
- Virtual hosts: a specific virtual host (/wisecoop) has been configured to partition the communication flows of the modules of this application.

D-	hhit 10										Refreshed 2018-07-16 17:03:08 Refresh every 5 seconds
ыка	bbitMQ 3.7.4 Erlang 20	0.2.4									Virtual host wisecoop
											Cluster rabbit@rabbitm
Overview	Connections Channels E	xchange	s	Queu	es Adr	nin					User etraid Log ou
agination											
Page 1 🔹	of 1 - Filter:	Rege	x ?								Displaying 19 items , page size up to: 100
Overview					Messages			Message r	ates		+/-
/irtual host	Name	Feature	5 5	State	Ready	Unacked	Total		deliver / get	ack	
visecoop	amq.gen-5seMonLK801FoCisntlQ	AD E	ccl	idle	0	0	C				
wisecoop	amq.gen-nwjpCtEju7X1xlXv3ww1BQ	AD E	ccl	idle	0	0	C				
visecoop	amq.gen-p4tL9tYTI7btj9FA7UZvJw	AD E	ccl	idle	0	0	C				
wisecoop	energymix	D	1	idle	0	0	C	0.00/s	0.00/s	0.00/s	
visecoop	energymixforecast	D		idle	0	0	C	0.00/s	0.00/s	0.00/s	
visecoop	energymixprovider			idle	0	0	C	0.00/s	0.00/s	0.00/s	
visecoop	esiosprovider			idle	0	0	C	0.00/s	0.00/s	0.00/s	
visecoop	forecasting_demand	D	1	idle	0	0	C	0.00/s	0.00/s	0.00/s	
wisecoop	forecasting_production	D		idle	0	0	0	0.00/s	0.00/s	0.00/s	
wisecoop	spotprices	D		idle	0	0	0	0.00/s	0.00/s	0.00/s	
visecoop	weather	D		idle	0	0	0	0.00/s	0.00/s	0.00/s	
visecoop	weatherforecast	D	1	idle	0	0	0	0.00/s	0.00/s	0.00/s	
wisecoop	weatherforecastprovider			idle	0	0	C	0.00/s	0.00/s	0.00/s	
wisecoop	wisecoopTest1	D	1	idle	0	0	(
visecoop	wisehome	D		idle	0	0	C	0.00/s	0.00/s	0.00/s	
visecoop	wisehomeNotifications	D	1	idle	0	0	C				
wisecoop	wisehomeTest	D		idle	0	0	C				
wisecoop	wisehome_test_ETRA	D	1	idle	0	0	0				
	wisehome test ETRA resp	D		idle	0	0	0				

▼ Add a new queue

Figure 6 – Queues created in the internal ESB for exchange of information among modules of WiseCOOP





LLD-	hhit MO								Refreshed 2018-07-17 15:40:05 Refresh every 5 seconds
Ra	ibbitMQ.	3.7.4 Er	lang 20.2.4						Virtual host wisecoop
									Cluster rabbit@rabb
Overview	Connections	Channels	Excha	nges	Queues	Admin			User etraid Log
Connec	ctions								
- All conne	ections (32)								
agination									
Page 1 🔹	of 1 - Filter:		R	egex ?					Displaying 32 items , page size up to:
Overview				Details			Network		+/-
Virtual host		User name			S Protocol		From client 0B/s		1
wisecoop wisecoop	172.17.0.1:42072 172.17.0.1:50362	etra etra	running	0	AMQP 0-9-1 AMQP 0-9-1		0B/s	0B/s 0B/s	
wisecoop	172.17.0.1:50362	etra	running	0	AMOP 0-9-1		0B/s	0B/s	
wisecoop	172.17.0.1:50390	etra	running	0	AMQP 0-9-1		0B/s	0B/s	
wisecoop	172.17.0.1:53854 undefined	etra	running	0	AMQP 0-9-1		0B/s	0B/s	
wisecoop	172.17.0.1:54450	etra	running	0	AMQP 0-9-1	1	0B/s	0B/s	
wisecoop	172.17.0.1:54456	etra	running	0	AMQP 0-9-1	1	0B/s	0B/s	
wisecoop	172.17.0.1:54504	etra	running	0	AMQP 0-9-1	1	0B/s	0B/s	
wisecoop	172.17.0.1:54506	etra	running	0	AMQP 0-9-1	1	0B/s	0B/s	
wisecoop	172.17.0.1:54510	etra	running	0	AMQP 0-9-1	1	0B/s	0B/s	
wisecoop	172.17.0.1:54516	etra	running	0	AMQP 0-9-1		0B/s	0B/s	
wisecoop	172.17.0.1:54520	etra	running	c	AMQP 0-9-1		0B/s	0B/s	
wisecoop	172.17.0.1:54578	etra	running	0	AMQP 0-9-1		0B/s	0B/s	
wisecoop	172.17.0.1:56812	etra	running	0	AMQP 0-9-1		0B/s	0B/s	
wisecoop	172.17.0.1:56814	etra	running	0	AMQP 0-9-1		0B/s	0B/s	
wisecoop	172.17.0.1:56820	etra	running	0	AMQP 0-9-1		0B/s	0B/s	
wisecoop	172.25.1.95:38034	etra	running	0	AMQP 0-9-1		0B/s	0B/s	
wisecoop	172.25.1.95:38036 172.25.1.95:38038	etra etra	running	0	AMQP 0-9-1 AMQP 0-9-1		0B/s 0B/s	0B/s 0B/s	
wisecoop wisecoop	172.25.1.95:38038	etra	running	0	AMQP 0-9-1 AMQP 0-9-1		0B/s	0B/s	
wisecoop	undefined	edd	= running	0	Wildb 0-3-1	1	00/5	00/5	
wisecoop	172.25.1.95:44548	etra	running	0	AMQP 0-9-1	1	0B/s	0B/s	
wisecoop	172.25.1.95:50494	etra	running	0.	AMQP 0-9-1	1	0B/s	0B/s	
wisecoop	172.25.1.95:50496	etra	running	0	AMQP 0-9-1	1	0B/s	0B/s	
wisecoop	172.25.1.95:50498	etra	running	0	AMQP 0-9-1	1	0B/s	0B/s	
	172.25.1.95:59168	etra	running	0	AMOP 0-9-1	1	0B/s	0B/s	
wisecoop			- ranning						
wisecoop wisecoop	172.25.1.95:59170 172.25.1.95:59174	etra etra	running	0	AMQP 0-9-1 AMQP 0-9-1	1	0B/s 0B/s	0B/s 0B/s	

Figure 7 – List of actives connections (32) to the internal ESB of WiseCOOP

3.1.2.2 Real-time monitor

The *Real Time monitor* is the horizontal module that will handle the data ingestion for most of the applications of the project. It has been designed in order to fulfil the requirements for data ingestion accordingly to the requirements and the architecture of communications proposed for the applications.

Particularly for WiseCOOP, this module is in charge of tracking and storing in the databases of WiseCOOP the data items shown in the following table.

Data item	Source	Operational DB	Long-term DB
Energy mix	ENTSOE energy mix provider	x	Х
Energ <mark>y</mark> mix forecast	ENTSOE energy mix provider	x	x
Energy readings	Field assets (SMX, AMI systems)	x	х
Weather	Weather forecast provider	r X	х
Weather forecast	Weather forecast provider	r X	Х
Wholesale market prices	Wholesale market	X	

Table 11 – Data items tracked by Real-time monitor in WiseCOOP application





$\mathbf{\widehat{O}}$	Layers	Schema wisecoop_values	^
RB	▼ wisecoop	NO DESCRIPTION	L
^	wisecoop_clustering	Measurements (object) No description	
Home	wisecoop_contracts	Active_Energy_+_Total [number] No description	
=	wisecoop_usagePointLocations	Active_EnergyTotal [number] No description	
Layers	wisecoop_things	Reactive_Energy_QITotal [number] No description	
cayore	wisecoop_stats	Reactive_Energy_QILTotal [number] No description	
	wisecoop_energymix	Reactive_Energy_QIIITotal [number] No description	
Security	wisecoop_energymixforecast	Reactive_Energy_QIVTotal [number] No description	
æ	wisecoop_values	Active_Energy_+Incr [number] No description	
Metrics	wisecoop_tariffPlans	Active_EnergyIncr [number] No description	
	wisecoop_tariffRules	Reactive_Energy_QLIncr [number] No description	
	wisecoop_weather	Reactive_Energy_QIIIncr [number] No description	
Ecosystem	wisecoop_weatherforecast	Reactive_Energy_QIIIIncr (number) No description	
Q	wisecoop_rules	Reactive_Energy_QIVIncr [number] No description	
Map	wisecoop_tariffs	Meter [object] No description	
	wisecoop_flexRequests	mRID [string] No description	
-	wisecoop_flexOffers	reason [string] No description	
Catalog	wisecoop_flexOrders	remark (string) No description	
•	wisecoop_spotprices	value [string] No description	
		Readings [array] No description	L
		ReadingType [string] No description	
		timeStamp [string] No description	
		value [number] No description	
		ReadingQualities [object] No description	а.
		comment [string] No description	
		id [string]id de la colección	
		 timestamp [string] No description 	¥
	Figure 8 – Screens	shot of the Real-time monitor UI, showing energy readings schema	

3.1.2.3 KPI Engine

The KPI engine of WiseCOOP application has been implemented as a set of Spark jobs that are periodically triggered on the long-term database to perform the necessary calculations and push the results back to different collections of the database.

Spark job	Module	Description	Result KPIs
WiseCOOP energyDeltaCalc 15m	energydeltacalculator-assembly- 1.3.jar	Calculates aggregated registers for every meter and every 15 minutes	Energy demand Energy production Equivalent CO ₂ emissions
WiseCOOP energyDeltaCalc 1h	energydeltacalculator-assembly- 1.3.jar	Calculates aggregated registers for every meter and every hour	Energy demand Energy production Equivalent CO ₂ emissions
WiseCOOP energyDeltaCalc 1d	energydeltacalculator-assembly- 1.3.jar	Calculates aggregated registers for every meter and every day	Energy demand Energy production Equivalent CO ₂ emissions
WiseCOOP energyDeltaCalc portfolio 15m	energydeltacalculator_portfolio- assembly-0.3.jar	Calculates aggregated registers for the whole portfolio of prosumers every 15 minutes	Energy demand Energy production Equivalent CO ₂ emissions

Table 12 – WiseCOOP – Spark jobs of the KPI engine





WiseCOOP energyDeltaCalc portfolio 1h	energydeltacalculator_portfolio- assembly-0.3.jar	Calculates aggregated registers for the whole portfolio of prosumers every hour	Energy demand Energy production Equivalent CO ₂ emissions
WiseCOOP energyDeltaCalc portfolio 1d	energydeltacalculator_portfolio- assembly-0.3.jar	Calculates aggregated registers for the whole portfolio of prosumers every day	Energy demand Energy production Equivalent CO ₂ emissions
WiseCOOP clustering demand 2p	clustering-assembly-0.1.jar	Identifies similar behaviours attending to the demand metrics of the prosumers of the portfolio, classifying those together with other members with similar behaviours	Prosumer profiling (assignment to a group with other members with similar behaviour)
WiseCOOP clustering production	clustering-assembly-0.1.jar	Identifies similar behaviours attending to the production metrics of the prosumers of the portfolio, classifying those together with other members with similar behaviours	Prosumer profiling (assignment to a group with other members with similar behaviour)
WiseCOOP cluster demand v1 1h	energydeltacalculator_cluster- assembly-0.3.jar	Calculates hourly average behaviour of the members of each demand-based cluster identified	Energy demand Energy production Equivalent CO ₂ emissions
WiseCOOP cluster production v1 1h	energydeltacalculator_cluster- assembly-0.3.jar	Calculates hourly average behaviour of the members of each production-based cluster identified	Energy demand Energy production Equivalent CO ₂ emissions





Soork Spark Master at spark://wisegridpre.lab.id:7077

URL: spark://wisegridpre.lab.id:70 REST URL: spark://wisegridpre.lal Alive Workers: 1 Cores in use: 1 Total. 0 Used Wemory in use: 1024.0 MB Total. Applications: 0 Running, 8 Comp Drivers: 0 Running, 0 Completed Status: ALIVE	b.ld:6066 (cluster mod	io)										
Vorkers (1) Worker Id				Address		State	Cores		Memory			
worker-20180718095710-172.17.	0.5-8881			172.17.0.5:8881		ALIVE	1 (0 Use	d)		3 (0.0 B Used	Ì	
Completed Applications (8)											
									Time	User	State	-
Application ID	Name				Cores	Memory per l	Executor	Submitted	THING			Duration
		nergydeltacalculator	_cluster_production_	v1_ <mark>1</mark> h	Cores	Memory per B 1024.0 MB	Executor	Submitted 2018/07/18		etraid	FINISHED	1.7 min
app-20180718102417-0007	WiseCOOP_en		_cluster_production_ _cluster_demand_v1	-			Executor		10:24:17			
app-20180718102417-0007 app-20180718102228-0006	WiseCOOP_en WiseCOOP_en		_cluster_demand_v1	-	1	1024.0 MB	Executor	2018/07/18	10:24:17	etraid	FINISHED	1.7 min
app-20180718102417-0007 app-20180718102228-0006 app-20180718102128-0005	WiseCOOP_en WiseCOOP_en WiseCOOP_en	nergydeltacalculator	demand_v1 tfolio_1d	-	1	1024.0 MB 1024.0 MB	Executor	2018/07/18	10:24:17 10:22:28 10:21:28	etraid etraid	FINISHED FINISHED	1.7 min 1.6 min
app-20180718102417-0007 app-20180718102228-0006 app-20180718102128-0005 app-20180718102024-0004	WiseCOOP_en WiseCOOP_en WiseCOOP_en WiseCOOP_en	nergydeltacalculator	demand_v1 tfollo_1d tfollo_1h	-	1 1 1	1024.0 MB 1024.0 MB 1024.0 MB	Executor	2018/07/18 2018/07/18 2018/07/18	10:24:17 10:22:28 10:21:28 10:20:24	etraid etraid etraid	FINISHED FINISHED FINISHED	1.7 min 1.6 min 49 s
app-20180718102417-0007 app-20180718102228-0006 app-20180718102128-0005 app-20180718102024-0004 app-20180718101923-0003	WiseCOOP_en WiseCOOP_en WiseCOOP_en WiseCOOP_en WiseCOOP_en	nergydeltacalculator nergyDeltaCalc_port nergyDeltaCalc_port	demand_v1 tfollo_1d tfollo_1h	-	1 1 1 1	1024.0 MB 1024.0 MB 1024.0 MB 1024.0 MB	Executor	2018/07/18 2018/07/18 2018/07/18 2018/07/18	10:24:17 10:22:28 10:21:28 10:20:24 10:19:23	etraid etraid etraid etraid	FINISHED FINISHED FINISHED FINISHED	1.7 min 1.6 min 49 s 60 s
Application ID app-20180718102417-0007 app-20180718102228-0005 app-20180718102128-0005 app-20180718102024-0004 app-20180718101223-0003 app-20180718101235-0002 app-2018071810233-0001	WiseCOOP_en WiseCOOP_en WiseCOOP_en WiseCOOP_en WiseCOOP_en	nergydeltacalculator nergyDeltaCalc_port nergyDeltaCalc_port nergyDeltaCalc_port	demand_v1 tfollo_1d tfollo_1h	-	1 1 1 1 1 1	1024.0 MB 1024.0 MB 1024.0 MB 1024.0 MB 1024.0 MB	Executor	2018/07/18 2018/07/18 2018/07/18 2018/07/18 2018/07/18	10:24:17 10:22:28 10:21:28 10:20:24 10:19:23 10:19:23	etraid etraid etraid etraid etraid	FINISHED FINISHED FINISHED FINISHED FINISHED	1.7 min 1.6 min 49 s 60 s 55 s

Figure 9 – Screenshot of Spark server with executing WiseCOOP jobs

3.1.2.3.1 Prosumer profiling (clustering)

The KPI engine of WiseCOOP includes modules with the purpose of analysing the behaviour of the prosumers of the portfolio according to different metrics, and grouping them under groups of members with similar behaviours. This is useful to the user of the WiseCOOP tool, since it allows to identify common behaviour partners that can be used as baseline profiles for different tasks, such as identifying the better tariff for a group of prosumers, identify anomalous behaviours (i.e. prosumers whose behaviour deviates significantly from the baseline) or provide key figures to promote and incentive green behaviour on the prosumers (e.g. motivate individuals to achieve lower CO₂ emissions than other similar prosumers).

3.1.2.3.1.1 Demand profiling

The profiling of the prosumers based on the demand analyses the following 4 features using the k-means clustering algorithm:

- Average daily demand for working days, from 08:00 to 18:00
- Average daily demand for working days, from 18:00 to 08:00
- Average daily demand for non-working days, from 08:00 to 18:00
- Average daily demand for non-working days, from 18:00 to 08:00

The optimum number of clusters is automatically determined by following the Elbow method [5].

Table 13 – Summary of demand-based cluster centres on lab-testing data





```
"wd p2" : 1591.2962962963,
            "nwd p1" : 780.925925925926,
            "nwd_p2" : 1591.18518518518,
            "members" : [
                ...
            ]
        },
        {
            "wd p1" : 3988.38095238095,
            "wd_p2" : 4167.52380952381,
            "nwd p1" : 2111.52380952381,
            "nwd p2" : 5734.23809523809,
            "members" : [
                ...
            ]
        },
        {
            "wd p1" : 4260.75,
            "wd p2" : 4030.25,
            "nwd_p1" : 1989.5,
            "nwd p2" : 22515.5,
            "members" : [
                ...
            ]
        }
   ],
   "description" : "2-period (08:00 - 18:00) distribution of demand
(working/non-working days)"
```

3.1.2.3.1.2 Production profiling

The profiling of the prosumers based on their production just considers one feature: the average daily production. Clustering is performed using the k-means clustering algorithm.

The optimum number of clusters is also automatically determined by following the Elbow method [5].

Table 14 – Summary of production-based cluster centres on lab-testing data

```
{
    "_id" : "5b1bbab48d6d6c32ac7ed83f",
    "type" : "demand v1",
```





```
"timestamp" : ISODate("2018-06-09T11:32:04.211Z"),
    "clusterCenters" : [
        {
             "wd p1" : 1693.62962962963,
            "wd p2" : 1591.2962962963,
            "nwd p1" : 780.925925925926,
            "nwd p2" : 1591.18518518518,
             "members" : [
                 ...
             "wd_p1" : 3988.38095238095,
            "wd p2" : 4167.5238095<mark>238</mark>1,
            "nwd p1" : 2111.523809<mark>523</mark>81,
            "nwd p2" : 5734.23809523809,
            "members" : [
                 ....
            1
        },
        {
            "wd p1" : 4260.75,
            "wd_p2" : 4030.25,
             "nwd p1" : 1989.5,
             "nwd p2" : 22515.5,
             "members" : [
                 ...
             ]
        }
    ],
    "description" : "2-period (08:00 - 18:00) distribution of demand
(working/non-working days)"
```

3.1.2.3.1.3 CO₂ equivalent emissions profiling

The profiling of the prosumers based on the equivalent CO₂ emissions of their demand just considers one feature: the daily equivalent CO₂ emissions. Clustering is performed using the k-means clustering algorithm.

The optimum number of clusters is also automatically determined by following the Elbow method [5].





```
Table 15 – Summary of CO<sub>2</sub>-based cluster centres on lab-testing data
```



3.1.2.3.1.4 Economic cost profiling

The profiling of the prosumers based on the cost associated to their demand just considers one feature: the daily economiccost. Clustering is performed using the k-means clustering algorithm.

The optimum number of clusters is also automatically determined by following the Elbow method [5].

```
Table 16 – Summary of economic cost-based cluster centres on lab-testing data
```

```
{
    "_id" : "cost_v1",
    "type" : "cost_v1",
    "timestamp" : ISODate("2018-07-25T14:06:06.086Z"),
    "clusterCenters" : [
```







3.1.2.3.2 Indicators required by WiseHOME

Additionally, a module has been developed with the objective of calculating and providing the required information for the WiseHOME application. This module uses pre-computed indicators – as exposed above – but also computes additional indicators on the fly as required, by posting aggregation queries to the long-term MongoDB database.

Table 17 – Capture of a conversation between WiseCORP and WiseHOME

>>
{"header":{"messageType":"REQUEST","conversationID":"ECPDK1004001","rec
ipient":"ECOPOWER","sender":"SampleSender"},"body":[{"reference":"USER"
,"assetKey":"ZIV0034949309","dataType":"TIMESERIES","metricType":"ENERG
Y_CONSUMPTION","startTime":"2018-07-17T10:18:21.939Z","endTime":"201807-18T10:18:21.939Z","sampleTime":"15
MIN","value":null,"metricTimeseries":null},{"reference":"USER","assetKe





```
y":"ZIV0034949309","dataType":"TIMESERIES","metricType":"ENERGY CONSUMP
TION", "startTime": "2017-07-17T10:18:21.939Z", "endTime": "2017-07-
18T10:18:21.939Z", "sampleTime": "15
MIN", "value":null, "metricTimeseries":null}]}
<<
{"header": {"conversationID": "ECPDK1004001", "messageType": "RESPONSE" }, "b
ody":[{"reference":"USER","assetKey":"ZIV0034949309","dataType":"TIMESE
RIES", "metricType": "ENERGY CONSUMPTION", "startTime": "2018-07-
17T10:18:21.939Z", "endTime": "2018-07-18T10:18:21.939Z", "sampleTime": "15
MIN", "value":null, "metricTimeseries": [{"timestamp": "2018-07-
17T11:15:00.000Z", "value":30}, {"timestamp": "2018-07-
17T12:15:00.000Z", "value":30}, {"timestamp":"2018-07-
17T13:15:00.000Z", "value":29}, {"timestamp":"2018-07-
17T14:15:00.000Z", "value":30}, {"timestamp":"2018-07-
17T15:15:00.000Z", "value":29}, {"timestamp": "2018-07-
17T16:15:00.000Z", "value":29}, {"timestamp": "2018-07-
17T17:15:00.000Z", "value":28}, {"timestamp":"2018-07-
17T18:15:00.000Z", "value":28}, {"timestamp": "2018-07-
17T19:15:00.000Z", "value":294}, {"timestamp": "2018-07-
17T20:15:00.000Z", "value":801}, {"timestamp":"2018-07-
17<mark>T</mark>21:15:00.000Z","value":207},{"timestamp":"2018-07-
17T22:15:00.000Z", "value":29}, {"timestamp": "2018-07-
17T23:15:00.000Z", "value":30}, {"timestamp":"2018-07-
18T00:15:00.000Z", "value":29}, { "timestamp": "2018-07-
18T01:15:00.000Z","value":30},{"timestamp":"2018-07-
18T02:15:00.000Z","value":29},{"timestamp":"2018-07-
18T04:15:00.000Z", "value":75}, {"timestamp": "2018-07-
18T05:15:00.000Z", "value":77}, {"timestamp": "2018-07-
18T06:15:00.000Z","value":332},{"timestamp":"2018-07-
18T07:15:00.000Z","value":308},{"timestamp":"2018-07-
18T08:15:00.000Z", "value": 327}, {"timestamp": "2018-07-
18T09:15:00.000Z", "value":95}]}, {"reference": "USER", "assetKey": "ZIV0034
949309", "dataType": "TIMESERIES", "metricType": "ENERGY CONSUMPTION", "star
tTime":"2017-07-17T10:18:21.939Z","endTime":"2017-07-
18T10:18:21.939Z", "sampleTime":"15
MIN", "value":null, "metricTimeseries": [] }] }
```

3.1.2.4 Demand and production forecast service

This module of the WiseCOOP has been implemented over a RPC server which makes use of the ESB. In addition, this module makes use of the long-term database of the WiseCOOP, which is implemented over a MongoDB database. The RPC servers (demand and generation forecast) of this module are permanently running to manage the received queries through the RabbitMQ queues enabled to make use of the demand and production forecast.

The ID of the supply point and the period and the horizon of the desired forecast are specified within the message queries. In the case of production forecast, in addition to the defined fields, the type of generation technology is specified.

Once the query is de-serialized and parsed, the forecast module retrieves from the long-term database the necessary information to perform the forecast. To perform the forecast it retrieves information related to the consumed/produced energy, working calendar and weather information related to the queried installation. This information is available in the long-term database, as it is possible to appreciate in the next picture.





by wiscoop by wiscoop by data_wiscoop (1) by data_wiscoop by data_wiscoop	X	- 0				Robo 3T - 1.2
b discorp (1)						View Options Window Help
* b.jbjdt.,wiscop () b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * clendar * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * users () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * wiscop.genergymic () * d. 0.04% ct. * b.jbjdt.,wiscop () * d. 2.000042435 > Ding () * wiscop,wisc., 1/s * d. 2.000042435 > Ding () > Die () </th <th></th> <th></th> <th></th> <th></th> <th></th> <th>! • 📄 🔒 🕨 🔳 🖏 👘</th>						! • 📄 🔒 🕨 🔳 🖏 👘
* b.jbjdt.,wiscop () b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * clendar * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * users () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * b.jbjdt.,wiscop () * wiscop.genergymic () * d. 0.04% ct. * b.jbjdt.,wiscop () * d. 2.000042435 > Ding () * wiscop,wisc., 1/s * d. 2.000042435 > Ding () > Die () </th <th></th> <th></th> <th></th> <th></th> <th>Welcome × th getCollection(wisecoon value ×</th> <th>Ib_wisecoop (1)</th>					Welcome × th getCollection(wisecoon value ×	Ib_wisecoop (1)
• Displate viewcop • Displate viewcop • Displate viewcop • Oiget • Calcebraic (14) • Oiget • Oiget • Oiget • Undersop_views_15m • 0.494 sec. • 0 90 • 0 90 • 0 90 • Undersop_views_15m • 0.494 sec. • 0 90 • 0						b 📴 rb_wisecoop
Contentions (1) Contentions (1) Image: accounts loginServic. Key Value Type > unders (1) Objectid("Sace4a582:0bb000073da:0") (8 India) Objectid("Sace4a582:0bb00073da:0") Objectid("Sace4a5838:0bb00073da:0") Objectid("Sace				_bigdata_wisecoop	📑 rb_bigdata_wisecoop 📃 etra-id.com:27017 📄 rb	rb_bigdata_wisecoop (1)
> celendar <					db.getCollection('wisecoop_values_15m	rb_bigdata_wisecoop
> Meter Value Type > Underse Count_loginServic Value Type > Underse Colored Colored Colored > Wisecop_energymine J Massummatic (12 fields) Colored > Wisecop_values J Massummatic (12 fields) Colored > Wisecop_values J Partine 20004003483 String > Wisecop_values J Colored String > Wisecop_values String Colored			-			Collections (14)
i update v iii (1) 0jc;tdl("sacc4as380:0b00073dac0") (Firlds) 0jc;tdl i users j Messurements [12 felds] 00jc;tdl i wisccop_energymin/recest j Messurements [12 felds] Array i wisccop_values_1 iiii Readings [12 dements] Array i wisccop_values_1 iiii Readings [12 dements] Array i wisccop_values_1/h iiii genation Anage Sting i wisccop_values_1/h iiii genation Sting Sting i wisccop_waterMericreat iiii genation Sting Sting Sting i wisccop_waterMericreat iiii iii iiiii iiiiiiiiiiiiiiiiiiiiii						> 🧰 calendar
<pre>views of wisecop_energymid views of wisecop_energymid views of wisecop_energymid views of wisecop_energymid views of wisecop_values if wisecop_values i</pre>	^		Туре	Value	Key	> 🥅 meteor_accounts_loginServic
wisecop_energymid wisecop_energymid wisecop_unles; wisecop_values;			Object	{ 8 fields }		
visccop_net_grindroct visc visc cop_value_1 in visc cop_value_2			1		-	
• wisccop, value: iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii			Object	{ 12 fields }		
i wiscoop_valuer_15 iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii			· · · · · · · · · · · · · · · · · · ·		2	
 wisccop_values_1h wisccop_values_1h prerTimetamp 2018-05-7100000002 Date 2018-05-7100000002 Date Discop_values_1h Qi Objectid/Tasecka5880bb00073dac?) Sindal Qi Objectid/Tasecka58			2	-		
visiccop_values_ht_tt visiccop_values_cluters_ht visiccop_values_th_tt Date visiccop_values_cluters_ht visiccop_values_th visiccop_values_th Diget1 visiccop_values_th visiccop_values_th Visiccop_values_th Diget1 visiccop_values_th Visiccop_values_th Diget1 Diget1 visicop_values_th <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>						
> > > 0			Date		👼 updatedTime	
> >						
> iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii						
> Punctions > See (5) Objectid("Saec4as58:0bb000673dac4") (8 fields) Object > > G6 (0 b)jectid("Saec4as58:0bb000673dac5") (8 fields) Object > G6 (0 b)jectid("Saec4as58:0bb000673dac7") (8 fields) Object > G6 (0 b)jectid("Saec4as58:0bb000673dac7") (8 fields) Object > G6 (0 b)jectid("Saec4as58:0bb00073dac7") (8 fields) Object > G6 (0 b)jectid("Saec4as58:0bb00073dac7") (8 fields) Object > G6 (1) Objectid("Saec4as58:0bb00073dac7") (8 fields) Object > G6 (1) Objectid("Saec4as58:0bb00073dac7") (8 fields) Object > G6 (1) Objectid("Saec4as58:0bb00073dac7") (8 fields) Object > G6 (1) Objectid("Saec4as58:0bb00073dac7") (8 fields) Object > G6 (1) Objectid("Saec4as58:0bb00073dac7") (8 fields) Object > G6 (1) Objectid("Saec4as78:0bb00073dac7") (8 fields) Object > G6 (1) Objectid("Saec4as78:0bb00073dac7") (8 fields) Object			Object	{ 8 fields }	> (3) ObjectId("5aec4ea588c0bb000673dac2")	> wisecoop_weather
Users > @ (6) Objectld("Saec4as58c0bb000673dac5") (8 fields) Object > @ (7) Objectld("Saec4as58c0bb000673dac5") (8 fields) Object > @ (8) Objectl("Saec4as58c0bb000673dac5") (8 fields) Object > @ (9) Objectl("Saec4as58c0bb000673dac5") (8 fields) Object > @ (10) Objectl("Saec4as58c0bb000673dac5") (8 fields) Object > @ (10) Objectl("Saec4as58c0bb000673dac5") (8 fields) Object > @ (12) Objectl("Saec4as58c0bb000673dac5") (8 fields) Object > @ (12) Objectl("Saec4as58c0bb000673dac5") (8 fields) Object > @ (13) Objectl("Saec4as58c0bb000673dac5") (8 fields) Object > @ (15) Objectl("Saec4as58c0bb000673dac5") (8 fields) Object > @ (15) Objectl("Saec4as58c0bb000673dac7") (8 fields) Object > @ (15) Objectl("Saec4as58c0b000673dac7") (8 fields) Object > @ (15) Objectl("Saec4as78c0b000673dac7") (8 fields) Object > @ (15) Objectl("Saec4as78c0b000673dac7") (8 fields) Object > @ (16) Objectl("Saec4as78c0b000673dac7") (8 fields) Object > @ (16) Ob			Object	{ 8 fields }	> 4 ObjectId("5aec4ea588c0bb000673dac3")	
> 600 > 600 (7) Objectld("Saec4ea588:Obb000673dac5") (8) fields) Object > 600 (9) Objectld("Saec4ea588:Obb000673dac5") (8) fields) Object > 600 (9) Objectld("Saec4ea588:Obb000673dac5") (8) fields) Object > 600 (10) Objectld("Saec4ea588:Obb000673dac5") (8) fields) Object > 600 (10) Objectld("Saec4ea588:Obb000673dac5") (8) fields) Object > 600 (13) Objectld("Saec4ea588:Obb000673dac5") (8) fields) Object > 600 (15) Objectld("Saec4ea588:Obb000673dac5") (8) fields) Object > 600 (15) Objectld("Saec4ea588:Obb000673dac5") (8) fields) Object > 600 (15) Objectld("Saec4ea788:Obb000673dac5") (8) fields) Object > 600 (16) Objectld("Saec4ea788:Obb000673dac5") (8) fiel			Object	{ 8 fields }	>	
> 63 (8) Objectld("Saec4ea588:Obb000673da:7") (8) Fields) > 63 (9) Objectld("Saec4ea588:Obb000673da:7") (8) Fields) Object > 63 (10) Objectld("Saec4ea588:Obb000673da:7") (8) Fields) Object > 63 (11) Objectld("Saec4ea588:Obb000673da:7") (8) Fields) Object > 63 (12) Objectld("Saec4ea588:Obb000673da:7") (8) Fields) Object > 63 (13) Objectld("Saec4ea588:Obb000673da:7") (8) Fields) Object > 64 (13) Objectld("Saec4ea588:Obb000673da:7") (8) Fields) Object > 64 (15) Objectld("Saec4ea788:Obb000673da:7") (8) Fields) Object > 64 (15) Objectld("Saec4ea788:Obb000673da:7") (8) Fields) Object > 64 (16) Objectld("Saec4ea788:Obb000673da:7") (8) Fields) Object > 64 (20) Objectld("Saec4ea788:Obb000673da:1") (8) Fie			Object	{ 8 fields }	> (6) ObjectId("5aec4ea588c0bb000673dac5")	> 📙 Users
> 63 (9) Objectid("Sacc4as588:Obb000673dac3") (8 fields) Object > 63 (10) Objectid("Sacc4as588:Obb000673dac3") (8 fields) Object > 63 (11) Objectid("Sacc4as588:Obb000673dac3") (8 fields) Object > 63 (12) Objectid("Sacc4as588:Obb000673dac5") (8 fields) Object > 63 (13) Objectid("Sacc4as588:Obb000673dac5") (8 fields) Object > 63 (13) Objectid("Sacc4as588:Obb000673dac5") (8 fields) Object > 64 (15) Objectid("Sacc4as588:Obb000673dac5") (8 fields) Object > 64 (15) Objectid("Sacc4as588:Obb000673dac5") (8 fields) Object > 64 (15) Objectid("Sacc4as788:Obb000673dc12") (8 fields) Object > 64 (15) Objectid("Sacc4as788:Obb000673dc12") (8 fields) Object > 64 (15) Objectid("Sacc4as788:Obb000673dc12") (8 fields) Object > 64 (20) Objectid("Sacc4as788:Obb000673dc13") (8 fields) Object > 64 (21) Objectid("Sacc4as788:Obb000673dc13") (8 fields) Object > 64 (22) Objectid("Sacc4as788:Obb000673dc13") (8 fields) Object <			Object	{ 8 fields }	>	
> 60 (10) Objectld("Saec4as388:Obb000673dac3") (8 fields) Object > 60 (11) Objectld("Saec4as388:Obb000673dac3") (8 fields) Object > 60 (12) Objectld("Saec4as388:Obb000673dac3") (8 fields) Object > 60 (12) Objectld("Saec4as388:Obb000673dac4") (8 fields) Object > 60 (13) Objectld("Saec4as388:Obb000673dac4") (8 fields) Object > 60 (15) Objectld("Saec4as388:Obb000673dac4") (8 fields) Object > 60 (15) Objectld("Saec4as388:Obb000673dac4") (8 fields) Object > 60 (15) Objectld("Saec4as388:Obb000673dac4") (8 fields) Object > 60 (16) Objectld("Saec4as388:Obb000673dac4") (8 fields) Object > 60 (16) Objectld("Saec4as388:Obb000673dac4") (8 fields) Object > 60 (19) Objectld("Saec4as388:Obb000673dac4") (8 fields) Object > 60 (19) Objectld("Saec4as38:Obb000673dac4") (8 fields) Object > 60 (20) Objectld("Saec4as788:Obb000673dc13") (8 fields) Object > 60 (20) Objectld("Saec4as788:Obb000673dc13") (8 fields) Object <			Object	{ 8 fields }	> (8) ObjectId("5aec4ea588c0bb000673dac7")	
> 60 (1) Objectld("Saec4a5880:0b000673daca") (8 fields) Object > 60 (12) Objectld("Saec4a5880:0b000673daca") (8 fields) Object > 60 (13) Objectld("Saec4a5880:0b000673daca") (8 fields) Object > 60 (13) Objectld("Saec4a5880:0b000673daca") (8 fields) Object > 60 (14) Objectld("Saec4a5880:0b000673daca") (8 fields) Object > 60 (15) Objectld("Saec4a7880:0b000673dac1") (8 fields) Object > 60 (15) Objectld("Saec4a7880:0b000673dac1") (8 fields) Object > 60 (16) Objectld("Saec4a7880:0b000673dac1") (8 fields) Object > 60 (19) Objectld("Saec4a788:0b000673dc1") (8 fields) Object > 60 (20) Objectld("Saec4a788:0b000673dc1") (8 fields) Object > 60 (21) Objectld("Saec4a788:0b000673dc1") (8 fields) Object > 60 (21) Objectld("Saec4a788:0b000673dc15") (8 fields) Object > 61 (21) Objectld("Saec4a788:0b000673dc15") (8 fields) Object			Object	{ 8 fields }	> (9) ObjectId("5aec4ea588c0bb000673dac8")	
> Will (12) Objectld("Saec-4as388:Obb000673dacb") (8 fields) Object > Will (14) Objectld("Saec-4as388:Obb000673dacb") (8 fields) Object > Will (14) Objectld("Saec-4as388:Obb000673dacb") (8 fields) Object > Will (15) Objectld("Saec-4as788:Obb000673dc12") (8 fields) Object > Will (15) Objectld("Saec-4as788:Obb000673dc12") (8 fields) Object > Will (15) Objectld("Saec-4as788:Obb000673dc12") (8 fields) Object > Will (15) Objectld("Saec-4as788:Obb000673dc13") (8 fields) Object > Will (15) Objectld("Saec-4as788:Obb000673dc13") (8 fields) Object > Will (23) Objectld("Saec-4as788:Obb000673dc13") (8 fields) Object > Will (23) Objectld("Saec-4as788:Obb000673dc13") (8 fields) Object > Will (23) Objectld("Saec-4as788:Obb000673dc13") (8 fields)			Object	{ 8 fields }	>	
> 600 (13) Objectld("Saec-4as388:Obb000673dacc") (8 fields) Object > 600 (14) Objectld("Saec-4as388:Obb000673dacc") (8 fields) Object > 600 (15) Objectld("Saec-4as388:Obb000673dacc") (8 fields) Object > 600 (15) Objectld("Saec-4as388:Obb000673dact") (8 fields) Object > 600 (15) Objectld("Saec-4as388:Obb000673dact") (8 fields) Object > 600 (17) Objectld("Saec-4as388:Obb000673dact") (8 fields) Object > 600 (19) Objectld("Saec-4as388:Obb000673dact") (8 fields) Object > 600 (19) Objectld("Saec-4as388:Obb000673dact") (8 fields) Object > 600 (19) Objectld("Saec-4as388:Obb000673dact") (8 fields) Object > 600 (20) Objectld("Saec-4as788:Obb000673dct") (8 fields) Object > 600 (20) Objectld("Saec-4as788:Obb000673dct") (8 fields) Object > 601 (20) Objectld("Saec-4as788:Obb000673dct") (8 fields) Object > 601 (20) Objectld("Saec-4as788:Obb000673dct") (8 fields) Object > 601 (20) Objectld("Saec-4as788:Obb000673dct") (8 fields) Obj			Object	{ 8 fields }	> (11) ObjectId("5aec4ea588c0bb000673daca")	
> 400 140 Objectdl("Saec-4ea388:Obb000673dacd") (8 fields) Objectd > 600 (15) Objectdl("Saec-4ea388:Obb000673dacd") (8 fields) Objectd > 600 (15) Objectdl("Saec-4ea388:Obb000673dacd") (8 fields) Objectd > 600 (15) Objectdl("Saec-4ea788:Obb000673dacd") (8 fields) Objectd > 600 (15) Objectdl("Saec-4ea788:Obb000673dacd") (8 fields) Objectd > 600 (19) Objectdl("Saec-4ea788:Obb000673dacd") (8 fields) Objectd > 600 (20) Objectdl("Sae			Object	{ 8 fields }	> (12) ObjectId("5aec4ea588c0bb000673dacb")	
> 600 (15) Objectd("5aec4ea788:0bb000673dace") (8 fields) Object > 600 (16) Objectd("5aec4ea788:0bb000673dac1") (8 fields) Object > 600 (17) Objectd("5aec4ea788:0bb000673dac1") (8 fields) Object > 600 (17) Objectd("5aec4ea788:0bb000673dc12") (8 fields) Object > 600 (18) Objectd("5aec4ea788:0bb000673dc12") (8 fields) Object > 600 (19) Objectd("5aec4ea788:0bb000673dc12") (8 fields) Object > 600 (20) Objectd("5aec4ea788:0bb000673dc13") (8 fields) Object > 600 (20) Objectd("5aec4ea788:0bb000673dc15") (8 fields) Object > 600 (20) Objectd("5aec4ea788:0bb000673dc15") (8 fields) Object > 600 (20) Objectd("5aec4ea788:0bb000673dc15") (8 fields) Object			Object	{ 8 fields }	> (13) ObjectId("5aec4ea588c0bb000673dacc")	
> 600 (15) Objectd('5aec4ar38300b000673dc1') (8 fields) Object > 600 (17) Objectd('5aec4ar38300b000673dc1') (8 fields) Object > 600 (18) Objectd('5aec4ar38300b000673dc1') (8 fields) Object > 600 (18) Objectd('5aec4ar38300b000673dc1') (8 fields) Object > 600 (19) Objectd('5aec4ar3800bb000673dc1') (8 fields) Object > 600 (20) Objectd('5aec4ar3800bb000673dc1') (8 fields) Object > 600 (21) Objectd('5aec4ar3800bb000673dc1') (8 fields) Object > 600 (22) Objectd('5aec4ar3800bb000673dc1') (8 fields) Object > 600 (22) Objectd('5aec4ar3800bb000673dc1') (8 fields) Object > 600 (23) Objectd('5aec4ar3800bb000673dc1') (8 fields) Object > 600 (23) Objectd('5aec4ar3800bb000673dc1') (8 fields) Object > 600 (23) Objectd('5aec4ar3800bb000673dc1') (8 fields) Object > 600 (25) Objectd('5aec4ar3800bb000673dc1') (8 fields) Object > 600 (26) Objectd('5aec4ar3800bb000673dc1') (8 fields) Object > 600			Object	{ 8 fields }	> (14) ObjectId("5aec4ea588c0bb000673dacd")	
> 630 (17) Objectld("Saec4as288:Obb000673dacf") (8 fields) Object > 630 (18) Objectld("Saec4as288:Obb000673dc12") (8 fields) Object > 630 (19) Objectld("Saec4as288:Obb000673dc12") (8 fields) Object > 630 (19) Objectld("Saec4as288:Obb000673dc12") (8 fields) Object > 630 (20) Objectld("Saec4as288:Obb000673dc13") (8 fields) Object > 630 (22) Objectld("Saec4as288:Obb000673dc13") (8 fields) Object > 630 (22) Objectld("Saec4as288:Obb000673dc13") (8 fields) Object > 630 (22) Objectld("Saec4as288:Obb000673dc13") (8 fields) Object > 630 (23) Objectld("Saec4as288:Obb000673dc15") (8 fields) Object > 630 (23) Objectld("Saec4as288:Obb000673dc15") (8 fields) Object > 630 (23) Objectld("Saec4as288:Obb000673dc15") (8 fields) Object > 630 (25) Objectld("Saec4as088:Obb000673dc15") (8 fields) Object > 630 (25) Objectld("Saec4as088:Obb00073dc15") (8 fields) Object > 630 (25) Objectld("Saec4ab88:Obb00073dc15") (8 fields) Object			Object	{ 8 fields }	> (15) ObjectId("5aec4ea588c0bb000673dace")	
> 60 (18) Objectd("5aec4ea788.0bb000673dc12") (8 fields) Object > 60 (19) Objectd("5aec4ea788.0bb000673dc12") (8 fields) Object > 60 (19) Objectd("5aec4ea788.0bb000673dc13") (8 fields) Object > 60 (20) Objectd("5aec4ea788.0bb000673dc13") (8 fields) Object > 60 (21) Objectd("5aec4ea788.0bb000673dc13") (8 fields) Object > 60 (22) Objectd("5aec4ea788.0bb000673dc13") (8 fields) Object > 60 (22) Objectd("5aec4ea788.0bb000673dc15") (8 fields) Object > 60 (23) Objectd("5aec4ea788.0bb000673dc15") (8 fields) Object > 60 (23) Objectd("5aec4ea788.0bb000673dc15") (8 fields) Object > 60 (23) Objectd("5aec4ea788.0bb000673dc15") (8 fields) Object > 60 (25) Objectd("5aec4ea788.0bb000673dc15") (8 fields) Object > 60 (26) Objectd("5aec4ea788.0bb000673dc15") (8 fields) Object > 60 (26) Objectd("5aec4ea788.0bb000673dc15") (8			Object	{ 8 fields }	> (16) ObjectId("5aec4ea788c0bb000673dc11")	
> 63 (19) Objectld("Saec4eb088:Obb000673e0e9") (8 fields) Object > 63 (20) Objectld("Saec4eb088:Obb000673e0en") (8 fields) Object > 63 (21) Objectld("Saec4eb088:Obb000673dc13") (8 fields) Object > 63 (21) Objectld("Saec4ea788:Obb000673dc13") (8 fields) Object > 63 (22) Objectld("Saec4ea788:Obb000673dc14") (8 fields) Object > 63 (23) Objectld("Saec4ea788:Obb000673dc15") (8 fields) Object > 63 (23) Objectld("Saec4ea788:Obb000673dc15") (8 fields) Object > 63 (25) Objectld("Saec4eb088:Obb000673dc15") (8 fields) Object > 63 (27) Objectld("Saec4eb088:Obb000673dc15") (8 fields) Object > 63 (27) Objectld("Saec4eb088:Obb000673dc15") (8 fields) Object </td <td></td> <td></td> <td>Object</td> <td>{ 8 fields }</td> <td>></td> <td></td>			Object	{ 8 fields }	>	
> 630 (20) Objecttl("Saec-4eb088.0bb000673e0ea") { 8 fields } Objectt > 630 (21) Objecttl("Saec-4e3788.0bb000673dc13") (8 fields } Objectt > 630 (22) Objecttl("Saec-4e3788.0bb000673dc14") (8 fields } Objectt > 630 (23) Objecttl("Saec-4e3788.0bb000673dc14") (8 fields } Objectt > 630 (23) Objecttl("Saec-4e3788.0bb000673dc15") (8 fields } Objectt > 630 (24) Objecttl("Saec-4e3788.0bb000673dc15") (8 fields } Objectt > 630 (25) Objecttl("Saec-4e3788.0bb000673dc15") (8 fields } Objectt > 630 (27) Objecttl("Saec-4e3788.0bb000673dc15") (8 fields }			Object	{ 8 fields }	> (18) ObjectId("5aec4ea788c0bb000673dc12")	
> (a) (2) Objecttl("Saec4ea788c0bb000673dc13") (8 fields) Object > (a) (2) Objecttl("Saec4ea788c0bb000673dc14") (8 fields) Objectt > (a) (2) Objecttl("Saec4ea788c0bb000673dc15") (8 fields) Objectt > (a) (2) (b)jecttl("Saec4ea788c0bb000673dc15") (8 fields) Objectt			Object	{ 8 fields }	>	
> 63 (22) Objectld("Saec4an788.Obb000673dc14") { 8 fields } Objectl > 63 (23) Objectld("Saec4an788.Obb000673e0eb") { 8 fields } Objectl > 63 (24) Objectld("Saec4an788.Obb000673e0eb") { 8 fields } Objectl > 63 (25) Objectld("Saec4an788.Obb000673e0eb") { 8 fields } Objectl > 63 (25) Objectld("Saec4an788.Obb000673e0eb") { 8 fields } Objectl > 63 (25) Objectld("Saec4an788.Obb000673dc15") { 8 fields } Objectl > 63 (25) Objectld("Saec4an788.Obb000673dc15") { 8 fields } Objectl > 63 (27) Objectld("Saec4an788.Obb000673dc15") { 8 fields } Objectl > 63 (27) Objectld("Saec4an788.Obb000673dc15") { 8 fields } Objectl > 63 (27) Objectld("Saec4an788.Obb000673dc15") { 8 fields } Objectl			Object	{ 8 fields }	> (20) ObjectId("5aec4eb088c0bb000673e0ea")	
> 633 (23) Objecttl("5aec4eb088C0bb000673e0eb") (8 fields) Objectt > 633 (24) Objecttl("5aec4er378e0bb000673e0eb") (8 fields) Objectt > 633 (25) Objecttl("5aec4er378e0bb000673e0eb") (8 fields) Objectt > 633 (25) Objecttl("5aec4er378e0bb000673e0eb") (8 fields) Objectt > 633 (26) Objecttl("5aec4er378e0bb000673e0eb") (8 fields) Objectt > 633 (27) Objecttl("5aec4er378e0bb000673e0eb") (8 fields) Objectt			Object	{ 8 fields }	> (21) ObjectId("5aec4ea788c0bb000673dc13")	
> 63 (24) Objectld("5aec4ea788c0bb000673dc15") (8 fields) Object > 63 (25) Objectld("5aec4ea788c0bb000673dc15") (8 fields) Object > (26) Objectld("5aec4e388c0bb000673dc15") (8 fields) Object > (27) Objectld("5aec4e388c0bb000673dc15") (8 fields) Object > (27) Objectld("5aec4e388c0bb000673dc15") (8 fields) Object			Object	{ 8 fields }	> 🖾 (22) Objectld("5aec4ea788c0bb000673dc14")	
> (25) Objecttd("5aec4eb088c0bb000673e0ec") { 8 fields } Object > (20) Objecttd("5aec4ea788c0bb000673dc16") { 8 fields } Object > (20) Objecttd("5aec4eb088c0bb000673e0ed") { 8 fields } Object			Object	{ 8 fields }	> (23) ObjectId("5aec4eb088c0bb000673e0eb")	
> 600 (26) Objecttld("5aec4ea788c0bb000673dc16") { 8 fields } Object > 600 (27) Objecttld("5aec4eb088c0bb000673e0ed") { 8 fields } Object			Object	{ 8 fields }	> 🖾 (24) Objectld("5aec4ea788c0bb000673dc15")	
620 (27) Objecttld("Saec4eb088c0bb000673e0ed") (8 fields) Object Objec			Object	{ 8 fields }	> (25) ObjectId("5aec4eb088c0bb000673e0ec")	
			Object	{ 8 fields }	> (26) ObjectId("5aec4ea788c0bb000673dc16")	
> 63 (28) Objectld("5aec4ea788c0bb000673dc17") { 8 fields } Object			Object	{ 8 fields }	> 🖾 (27) ObjectId("5aec4eb088c0bb000673e0ed")	
			Object	{ 8 fields }	> (28) ObjectId("5aec4ea788c0bb000673dc17")	
20 (29) Objectld("5aec4eb088c0bb000673e0ee") { 8 fields } Object	~		Object	{ 8 fields }	> (29) ObjectId("5aec4eb088c0bb000673e0ee")	

Figure 10 – WiseCOOP long-term database screenshot

Once the algorithm is ran, the response provided by this is serialized and sent back through the corresponding RabbitMQ queue, providing the queried information. The next message is an example of the response received by the WiseCOOP, which is printed in the graphic of the forecast view.

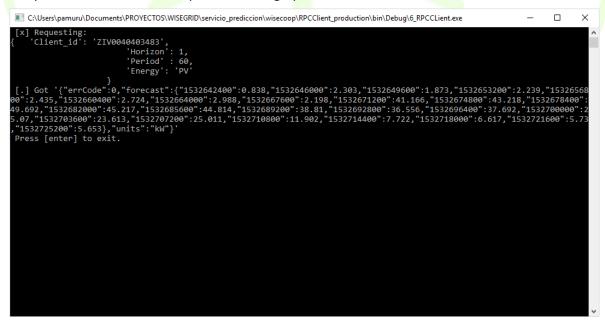


Figure 11 – Screenshot of forecast response message.





3.1.2.5 Demand response framework

The Demand Response framework is an end-to-end functionality that spans both the WiseCOOP and the WiseCORP tools. Within WiseCOOP, lie two main components of this functionality that exchange information with WiseCORP.

The first is responsible for collecting the demand flexibility potential of the various buildings that belong to the portfolio of the WiseCOOP operator. For this purpose, WiseCOOP issues commands – such as the following – in order to elicit the flexibility per building. This command is encapsulated in a message that specifies the target buildings (via their SMX ID), the time-frame of interest for the Demand Response action, as well as – optionally – a comfort level indication that can be used as a threshold to avoid the disruption of internal operations in the building. The information specified in the message usually results from the corresponding DSO request for demand response. So, the buildings are selected based on eligibility criteria, such as location of connection point, and the time-frame corresponds to the time of the expected demand shifting/shedding.

Table 18 – Elicitation of demand flexibility potential per building

```
"shoIds": [
   "SMX1",
   "SMX2"
],
"comfortLevel": 0.6,
"timeperiod": 60
```

Once the buildings – actually the WiseCORP instance that is responsible for each of the target buildings – respond to the WiseCOOP request, WiseCOOP will calculate the combination of assets that can best fulfil the request of the DSO. Then the tool issues messages, like the following, to inform assets about the action they must undertake in order to comply with the DSO request. This message specifies exactly which devices (venID) should modify their energy consumption by how much (aggregatedPnode) and at what specific time (startTime/activePeriod).

Table 19 – Request from WiseCOOP for specific asset demand modulation as a result of a DR event Send DR Request

```
Γ
 {
    "eiEventDescriptor": {
      "eventID": "1",
      "createdDateTime": "2018-07-13T12:12:12"
   },
    "eiEventSignals": {
      "eiEventSignal": [
        {
          "signalID": "17",
          "startTime": "2018-07-13T12:00:00",
          "activePeriod": "PT15M",
          "eiTarget": {
            "venID": "SMX2",
            "aggregatedPnode": "10"
          }
        },
        {
          "signalID": "97",
```





```
"startTime": "2018-07-13T12:15:00",
        "activePeriod": "PT15M",
        "eiTarget": {
          "venID": "SMX2",
          "aggregatedPnode": "8"
        }
      },
      {
        "signalID": "107",
        "startTime": "2018-07-13T12:30:00",
        "activePeriod": "PT15M",
        "eiTarget": {
          "venID": "SMX2",
          "aggregatedPnode": "1.5"
        }
      }
    ],
    "numDataSources": "3"
  },
  "activePeriod": "PT45M",
  "eiTarget": {
    "venID": "SMX2",
    "aggregatedPnode": "19.5"
  }
}
```

3.1.3 User interface

In this part of the document, the main sections and functionalities of the WiseCOOP GUI are described, including some screenshots of the actual interfaces.

For the implementation of the WiseCOOP unified GUI, the **MeteorJS** web framework has been used. MeteorJS (or simply 'Meteor'), is "A free and open-source JavaScript web framework written using Node.js. Meteor allows for rapid prototyping and produces cross-platform (Android, iOS, Web) code. It integrates with MongoDB and uses the Distributed Data Protocol and a publish–subscribe pattern to automatically propagate data changes to clients without requiring the developer to write any synchronization code. On the client, Meteor depends on jQuery and can be used with any JavaScript UI widget library".

On the client part of a Meteor application, a number of plugins and technologies can be used to provide the user a better experience. The main plugins we have used for the client side are:

- SemanticUI as CSS framework. CSS frameworks are pre-prepared software frameworks that are
 meant to allow for easier, more standards-compliant web design using the Cascading Style Sheets
 language. They are mostly design oriented and unobtrusive. This differentiates these from functional
 and full JavaScript frameworks. By using this CSS framework we achieve easily a modern and coherent style across the whole user interface. The selection of SemanticUI over other CSS framework is
 mainly based on our expertise and the fact that this one has been designed to be easily understandable and usable. Other framework tends to become quite hard to use as interfaces becomes bigger.
- LeafletJS as the mapping solution for the web client. This JavaScript-based framework provides a wide range of mapping providers to use and offers a big set of plugins to personalize the user interaction with the map and the display of the information. And everything is open source and free.
- **HighchartsJS** is a charting JavaScript framework that helps displaying data in the form of charts for web environments.
- **BlazeJS** for the user interface lay out. It is a powerful library for creating user interfaces by writing reactive HTML templates.





For the server part, **MongoDB** has been used as the database for storing real-time data. This is a no-SQL database that stores unstructured information. It is tightly coupled with Meteor. The reactive nature of the data changes in MongoDB database is at the core of the web application.

The web application is protected with a user/password credential system to avoid non-authorized personnel to access sensible information. These credentials are requested before accessing the rest of the application.

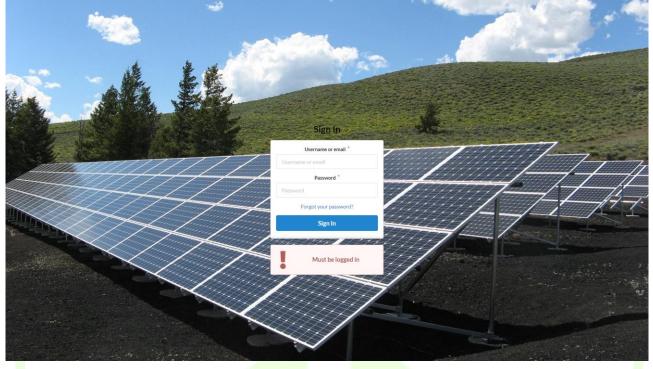


Figure 12 – WiseCOOP Login

This credential system also permits the definition of different user profiles to grant or deny access to each section of the application independently. This functionality provides an additional level of privacy, as well as flexibility for the system administrator and the operators that use the application.

Once the user has been granted access to the application, diverse functionalities will be available as described in the sections below.

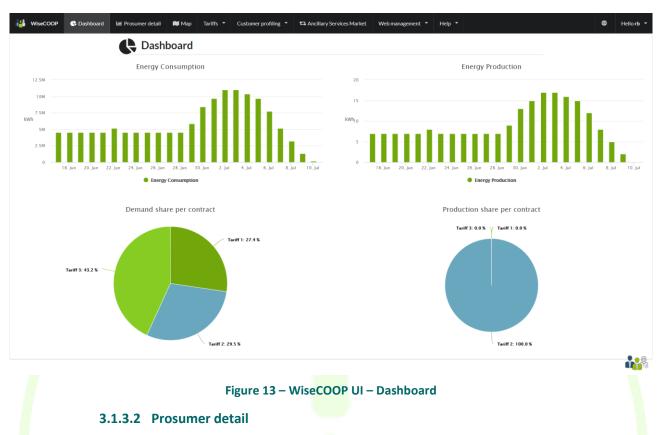
3.1.3.1 Dashboard

The dashboard presents aggregated indicators for the overall portfolio and the last 30 days, namely:

- Daily demand
- Daily production
- Demand share per contract type
- Production share per contract type







Under this section, access to raw energy readings is provided. Upon selection of the data to be visualized (asset, date range, metric and integration period), the data is displayed in a chart.

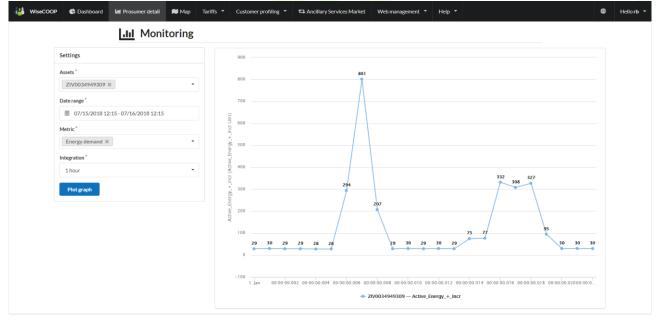


Figure 14 – WiseCOOP UI – Prosumer detail

3.1.3.3 Geographical demand heat-map





Under this section, a visualization of the demand of the last 30 days is geographically displayed in a map. The demand measured at each geo-referenced asset is aggregated and provided a relative colour (red represents the greatest demand, green the lowest ones).

This visualization can be of use to retailers and aggregators to understand where the members of the portfolio are located, where most of the energy is demanded, identify areas with expansion capabilities, etc.

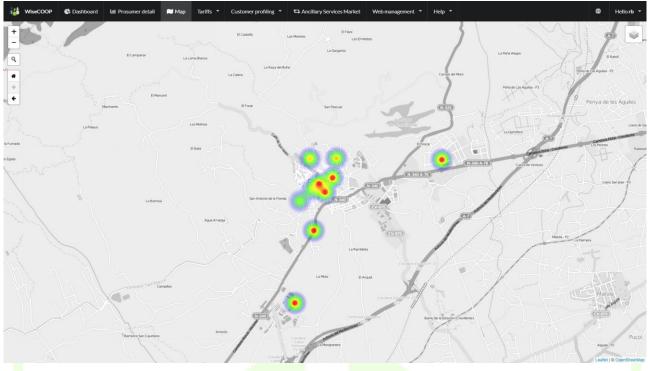


Figure 15 – WiseCOOP UI – Geographical demand heat-map

3.1.3.4 Tariff designer

The tariff designer section comprises two different sites: tariff period definition and tariff definition.

3.1.3.4.1 Tariff period definition

This section allows the user of WiseCOOP to introduce in the database new definitions of tariff periods that may be reused among different tariffs. In order to define the periods, the following information is required:

- Name
- Site: pilot site where the tariff period applies (for internal project purposes)
- Country: country the tariff period applies
- Set of period definition rules:
 - Price type: name of the period to which the rule applies (e.g. p1). A single period can be referenced in more than one rule (for instance, several rules are required in order to define different hourly periods for winter and summer time).
 - Application period: specifies whether the rule is defined for a set of hours or a set of days
 - In case the rule applies to a range of hours, the start and end hour (in local time) must be provided.
 - In case the rule applies to a range of days, the start and end date must be provided
 - Special dates: for each type of special day, it defines whether the rule is applied (*true* selected), is not applied (*false* selected) or if the type of day doesn't need to be taken in





consideration (*undefined* selected). The types of special days defined are:

- Saturdays
- Sundays
- Daylight Saving Time (summer time)
- National holidays
- Special holidays (e.g. 6th January in Spain)

WiseCOOP Cashboard	🕍 Prosumer detail 📦 Map 🛛 Tariffs	Customer profiling * t3 Ancillary Services Market Web management * Help *	🕀 Hellorb 🔻
	Tariff periods		
	Rules	C Periods	
	Name 3.0A	Price type	
	3.0A	Name p1	
	3.1A	Application period	
	Country	From To O 0x00 7:59	
	+ 🖻 🗙	Special dates	
		Price type	
		Name p2	
		Application period	
		From To 8:00 17:59	
		Special dates	
		Daylight saving time	₽ ₽₽

Figure 16 – WiseCOOP UI – Tariff period definition

3.1.3.4.2 Tariff definition

This section allows the user of WiseCOOP to create the definition of the tariff. A tariff is composed of the following elements:

- Name
- Tariff period definition that is applicable
- For each tariff period
 - O Energy term price (€/kWh)
 - Power term price (annual €/kW in contracted capacity)

🕌 WiseCOOP	🚱 Dashboard	<section-header> Prosumer detail</section-header>	🛤 Map	Tariffs 🔻	Customer profilin	 Ancillary Services Market 	Web management 🔻	Help 🔻		•	Hellorb 🔻
		€ Tariff	s								
		Tariffs	;		€ Peri	ods					
		Tariff Name Tariff 1		-		Înergy	Power				
		Rule 3.0A			p1	0.161604 €/kWh	35.649473 €/kW				
		+ 2 3	ĸ		p2	0.083916 €/kWh	35.649473 €/kW				
			-		р3	0.056259 €/kWh	35.649473€/kW				

Figure 17 – WiseCOOP UI – Tariff definition





3.1.3.5 Portfolio profiling

This section displays the results of the different modules composing the portfolio profiling section of the KPI engine.

The first site displays, for a selected profiling criteria (demand or production profile), the identified groups, their average values for each one of the considered features (cluster centres), and the total share of the portfolio members that take part of that group.

🕌 WiseCOOP	🚯 Dashboard	<section-header> Prosumer detail</section-header>	🛤 Map	Tariffs 🔻	Customer profiling *	🔁 Ancillary Services Mark	t Web management 🔻	Help 🔻		۲	Hello rb 🔻
	Clusters										
Select type of cluste Demand	Demand •										
				/orking day			Non-working day				o share
	Average	demand (08:00-18:00)		Avera	ge demand (18:00-08:00)	Averag	e demand (08:00-18:00)		Average demand (18:00-08:00)	Portione	o share
Cluster 1		1693	3.63 kWh		159	91.30 kWh	780.93	kWh	1591.19 kWh		27 (51.92%)
Cluster 2		3988	3.38 kWh		416	7.52 kWh	2111.5	kWh	5734.24 kWh		21 (40.38%)
Cluster 3		4260	0.75 kWh		403	80.25 kWh	1989.50	kWh	22515.50 kWh		4 (7.69%)

Figure 18 – WiseCOOP UI – Portfolio profiling, identified groups

A second site gives further insight in the analysis of the profiling results, in the form of time charts of demand and production. In this section, the user of WiseCOOP needs to select the criteria of the visualization:

- Profiling criteria (demand or production)
- Customer
- Date range

For the selected criteria, two charts are displayed:

- Energy demand chart, comparing the actual demand of the selected customer with the average of the group the customer has been associated to (i.e. the average behaviour of all members with similar demand patterns)
- Energy production chart, comparing the actual production of the selected customer with the average of the group the customer has been associated to (i.e. the average production of all members with similar production patterns)





WiseCOOP C Dashboard Prosumer detail	Map Tariffs * Customer profiling *	t3 Ancillary Services Market Web management * Help *	٠	Hellorb 🔻
C Time	comparison			
Select cluster "				
production_v1 ·				
Select customer *				
ZIV0040403524 •				
Date range *				
6/12/2018 00:00 - 06/13/2018 00:00				
Plot				
Energy demand	1	Energy production		
1000	- Customer Portfolio		 ◆ Customer ◆ Portfolio 	
750		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 		
	B + B + B + B + B + B + B + B + B + B +	6 ⁴⁹ 1 ⁴⁹		

Figure 19 – WiseCOOP UI – Portfolio profiling, member comparison with average profile

3.1.3.6 Tariff comparer

The tariff optimization section allows the aggregators to use the historical energy demand and production data retrieved from the portfolio of prosumers in energy cost simulations, thus providing valuable input to the tariff selection decision support. Users of WiseCOOP can trigger a new simulation by selecting the prosumer whose data they want to use in the simulation, and a complete month.

WiseCOOP @ Dishboard Lill Protume	ir detail 💷 Map Tariffs 🕋 Customer profiling 🕋 🕮 Ens	argytrade Demand response campaigns 🗢 Web management 👻 Help 🔹	O Holozb
	E T New tariff optimization		
	Select entities	Select month	ottimization .
	▲ 🗐 🖗 Assets	Month [*] June 2018 •	
	Octimitzan	Settings	
	ZIV0040403498	Tariff Green Fixed prices Discount	
	ZIV00404	Priority Consumption cost Global cost	
	Z100040403318	Optimization label	2
	Image: Simulation 21/00404 Image: Simulation 21/00404 Image: Simulation 21/00404 Image: Simulation 21/00404	Label* Label mample.	
	Auro 2016 ZIV00405320		17.16

Figure 20 – WiseCOOP UI – Tariff comparer, selection of criteria

Once the simulation is triggered, WiseCOOP automatically computes the applicable costs that would result for each one of the tariffs defined within the database. When the simulation finishes, a comprehensive comparison of the results is provided to the aggregator, making it very easy to identify which tariffs would benefit or would increase the costs associated to the energy demand.





WiseCOOP	🕍 Prosumer detail	I Map Tariffs - Cust € Tariff optimiza	omer profiling *	🕅 Energy trade	Demand response campaigns	 Web managemen 	t - He	lp -			⊕ ⊦	н
		Filter per label	9						New optimization			
		Optimization settings	Measured valu	~	Recommended energy p	lans	Cost (€)	Sa	vings (€)			
		Optimization settings	Measured valu		Energy plan name	Company	Global	Energy En	ergy			
		Simulation for client ZIV0040403321 - July 2018 July 2018 1 entities involved 0 Request date: 27/07/2018 11:20 Status: OnHold	Peak active por (kW) Consumption (I EnergyCost (€)	- kWh) -	0 · 0 · 0 ·	•	-	-	• • • • • •			
		Simulation for client ZIV0040403320 - June 2018		52.54	1 Precio Fijo 6.1	Nexus Energía	567.63		• -1.93			
		1 entities involved • × 13/06/2018 22:34 × Status: Completed ×	Consumption (EnergyCost (€)		 Tarifa 3.0 Ahorro +15 3.0A 	Alcanzia Energía Gesternova Energía	676.61					
		Show 2 rows per page						Page 1	of 8 🕑			

Figure 21 – WiseCOOP UI – Tariff comparer, results

In addition, the simulation results include the detail of how the energy bill would look like with each one of the tariffs used in the simulation. These can be used to check the details during the comparison of different tariffs, as relevant information during negotiation of better tariffs with the retailer, or as an assessment of the correctness of the energy bills delivered by the retailer.

all 🛍 Map Tariffs 👻 Customer profilir	g 💌 🕅 Energy trade	Demand response campaign	is 👻 Web management 👻	Help *	
€ Ta € Simulation fo	or client ZIV004	4540020 - Jur	ne 2018		
Site	Sort solutions by	Solution			
Barcelona *	EnergyCost *	250.12 € > Tarifa 3.0 (Alcanzia Energía)	•	
Optimizati				()	
Month Days Entities	Peak active power (kW)	At	Total consumption (kWh)	Total cost (€)	
Simulation 30 1	52.54	30/06/2018 19:15	2796.5	232.96	
July 2018					
1 entities in Plan name Comp Request da		iff type Green		Discount	
Status: On Tarifa 3.0 🗅 Alcan	tla Energía 3.0	A No	Yes	No	
Simulation ZIV00445 Concept	Price	Value		Total	
June 2018	0.111585986 €/kW/day	47 kW		157.34€	
1 entities Ir 13/06/201 ContractedPower (P1)	0.066951589 €/kW/day	52 kW		104.44 €	
ContractedPower (P3)	0.044634397 €/kW/day	15 kW		20.09€	
Show 2	CONTROL OF CRIMINARY	13	TotalContractedPowe		of 8 🕥
ConsumedEnergy (P1)	0.1045 €/kWh	733.2 kWh		76.62€	
ConsumedEnergy (P2)	0.0869 €/kWh	1833.3 kWh		159.31€	
ConsumedEnergy (P3)	0.06168 €/kWh	230 kWh		14.19€	
			TotalConsumedEnerg	y 250.12€	
EnergyTaxes	5.11269632%	531.99€		27.2€	
Vat	21%	559.18€		117.43€	

Figure 22 – WiseCOOP UI - Tariff comparer, simulated bills

3.1.3.7 Energy trade assistant

The energy trade assistant section has the objective of providing to the retailer valuable information about all the information retrieved by WiseCOOP that may assist them in the decision on how to approach the wholesale market. Within this context, the following information is provided for the selected time range:

- Wholesale market prices
- Energy mix of the energy available from wholesale market (RES / non-RES)
- Visualization of the demand forecast (as provided by WiseCOOP) against the actual demand measured by smart meters and AMI systems integrated with the tool. Indicators on deviations among the two are also provided, with the objective of evaluating how the forecasting module can help the





retailer to minimize the deviation between the energy bought from the wholesale market and the actual demand of the portfolio of prosumers

- Visualization of the production forecast (as provided by WiseCOOP) against the actual production measured by smart meters and AMI systems integrated with the tool. Indicators on deviations among the two are also provided, as an indication of the forecasting reliability
- Total economic cost associated to the demand covered by the wholesale market
- Equivalent CO2 emissions of the demand covered by the wholesale market

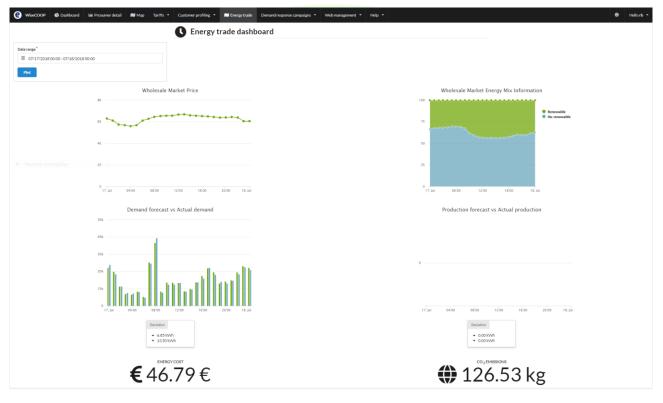


Figure 23 – WiseCOOP UI – Energy trade assistant

3.1.3.8 Demand response campaigns

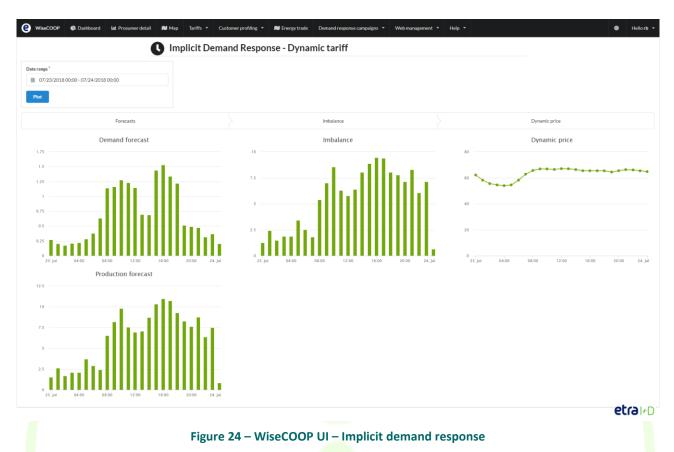
3.1.3.8.1 Implicit demand response – dynamic tariff

WiseCOOP includes all features needed to enable aggregators and retailers to implement implicit demandresponse strategies by offering to their customers a dynamic tariff that can be modulated accordingly to the interests of the WiseCOOP user. The design of these modules and ideas behind them are described in detail in D10.2. This section of the UI presents the main outcomes of those modules for a selected time range, namely:

- Demand and production forecast, which are the main input for the calculation of the dynamic tariff;
- Imbalance among forecasted demand and production;
- Dynamic price generated with the objective of minimizing the imbalance.







3.1.3.8.2 Explicit demand response

With regards to the participation of the aggregator in the explicit demand response campaigns, this section displays information on the active and finished campaigns initiated by the DSO, providing the following details:

- Congestion point where the demand response campaign is triggered;
- Starting timestamp;
- End timestamp;
- Required flexibility: requested power reduction (or increase) in the congestion area for the given period of time;
- Status: current status of the demand-response campaign (*Request posted*, *Offer sent*, *Order poster*, *DR campaign started*, *DR campaign finished*).





wiseCOOP & Dashboard		omer profiling * the Ancillary Services Market	Webmanagement * Help *		Hello
• Demand respons	e campaigns	Active campaigns			
Congestion point	Start	End	Flex. req.	Status	
		Finished campaigns			
Congestion point	Start	End	Flex. req.	Status	
CMTREN	09/06/2018 17:45	09/06/2018 19:45	50 kW	DR campaign finished	>
CMTREN	09/06/2018 17:45	09/06/2018 19:45	50 kW	DR campaign finished	>
CMDOMADOR	11/06/2018 00:00	12/06/2018 00:00	500 KW	DR campaign finished	>
CMTREN	09/06/2018 14:45	09/06/2018 16:45	50 kW	DR campaign finished	>
CMTREN	09/06/2018 14:45	09/06/2018 16:45	50 kW	DR campaign finished	>
CMTREN	09/06/2018 14:45	09/06/2018 16:45	50 kW	DR campaign finished	>
CMPARDO	09/06/2018 15:00	09/06/2018 17:00	50 kW	DR campaign finished	>
CMDOMADOR	09/06/2018 15:00	09/06/2018 17:00	50 KW	DR campaign finished	>
CMTREN	09/06/2018 15:15	09/06/2018 17:15	50 kW	DR campaign finished	>
CMTREN	09/06/2018 17:30	09/06/2018 19:30	50 kW	DR campaign finished	Ň

Figure 25 – WiseCOOP – List of active and finished explicit demand response campaigns

By selecting one of the campaigns, the corresponding details are displayed, including:

- Starting timestamp;
- Duration of the campaign;
- Required flexibility;
- History of events for this campaign: shows all events that happened in the flexibility market related to this campaign.

Table 20 – WiseCOOP UI – List of possible status for explicit demand response campaigns

	Timestamp	Status	Sender	Recipient	Flexibility	Price
Flex. request	Indicates when the request was posted by DSO	Request posted	DSO Operator	Aggregators subscribed to the USEF communications	Indicates the amount of flexibility requested	
Offer sent	Indicates when the offer was sent to the DSO	Offer sent	Name of the aggregator	DSO Operator	Indicates the amount of flexibility offered	Indicates the price requested for the offered flexibility
Order posted	Indicates when the order was posted by DSO	Order posted	DSO Operator	Name of the selected aggregator	Indicates the amount of flexibility ordered (the amount offered by the aggregator)	Indicates the price agreed with the aggregator (the price requested by the aggregator)





🕌 WiseCOOP	G Dashboard	🔟 Prosumer detail	🛤 Map	Tariffs 🔻	Customer profiling *	t Ancillary Services Market	Web managen	nent 🔻 Help 🔻		۲	Hello rb 🔻
℃ Dema	and respo	nse campaig	gn deta	ils							
		Date			R	equired flexibility			Duration		
	09/06	/2018 17:45				50 kW			02h 00min		
Histor	У										
1	limestamp		Stat	tus		Sender		Recipient	Flexibility	Price	
09/0	06/2018 13:47		Request	posted		DSO Operator					
09/0	06/2018 13:47		Offer	sent	Р	rosumer aggregator		DSO Operator	50 KW	150€	
09/0	06/2018 13:48		Order p	posted		DSO Operator		Fleet Manager	50 KW	150€	

Figure 26 – WiseCOOP – Details of an explicit demand response campaigns

3.2 LAB-TESTING RESULTS

This section contains a set of templates with the definition, objectives, steps and results of all tests executed during this period on the different modules of the tool.

3.2.1 RT	monitor tests		
Name	RTM001. Read smart meter data from IOP		
Module under test	RT Monitor	Resp.	ETRA
Module requirement	HL-UC 7_PUC_2_Dynamic aggregation operation operation operation into energy market	f distributed ener	gy assets and active
Test environment	SMX running and sending information to IOF IOP platform up and running		
Features to be tested	Data from SMX is properly collect <mark>ed i</mark> n the o	perational database	of WiseCOOP
Features not to be tested			
Preparation	Configure one SMX to send data to the lab-to	esting IOP environm	ent
Dependencies			
Steps	The execution of this test must h <mark>appen</mark> auto	matically upon publi	cation of data in the IOP
Pass criteria	Data from the SMX is correctly updated in the keeps a register of the last value <mark>s sent</mark> by the		se. Operational database
Suspension criteria			
Results	Test successful. The following screenshot shows how the ope testing environment SMX	erational DB is popul	ated with data from lab-





💊 Robo 3T - 1.2			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
File View Options Window He	lp			
📓 - 🍃 🖬 🕨 🗮 🗱				
▷	db.getCollection('wisec ×			
▷ □ _events	wisegridpre wisegridpre.lab.id:27017			
Iayers	db.getCollection('wisecoop valu			
_messages		les).lind({}).sort({timesta		
▷profiles	wisecoop_values () 0.049 sec.		4 0	50 🕨 🛅 🗐 🗎 🖞
▷ 🔤 _schema	Key	Value	Туре	
▷ 🔤 cronHistory	▲ 🚇 (1) ZIV0040403472	{ 5 fields }	Object	
meteor_accounts_loginSe	id	ZIV0040403472	String	
users	timestamp	2019-01-19 02:00:00.000Z	Date	
wisecoop_clustering	→ Meter	{ 4 fields }	Object	
wisecoop_contracts	Measurements	{ 12 fields }	Object	
wisecoop_demandForeca	Active_Energy_+_Total	7170379	Int32	
wisecoop_energymix	Active_EnergyTotal	0	Int32	
wisecoop_energymixfore.	Reactive_Energy_QITotal	3538771	Int32	
wisecoop_flexOffers	Reactive_Energy_QILTotal	0	Int32	
wisecoop_flexOrders	Reactive_Energy_QIIITotal	0	Int32	
wisecoop_flexRequests	Reactive_Energy_QIVTotal	16082	Int32	
wisecoop_productionFor	 Active_Energy_+_Incr 	123	Int32	
wisecoop_rules	Active_EnergyIncr	0	Int32	
wisecoop_spotprices	Reactive_Energy_QI - Incr	171	Int32	
wisecoop_tariffs	Reactive_Energy_QIIIncr	0	Int32	
wisecoop_usagePointLoc.	Reactive_Energy_QIIIncr Reactive_Energy_QIIIIncr	0	Int32	
wisecoop_values	Reactive_Energy_QIVIncr	1	Int32	
wisecoop_weather	Readings	[12 elements]	Array	
wisecoop_weatherforecas	t (2) ZIV0040409996	{ 5 fields }	Object	
Functions	▷ ○ (2) 21000404035550 ▷ ○ (3) 2100040403498	{ 5 fields }	Object	
🛛 🖉 📙 Users	→ (3) 21/00/04/03/98	{ 5 fields }	Object	
b B rb_wisecorp	▼ ○ ○ (4) ∑IV0044340057 ▼ ○ ○ (5) ZIV0040403522	{ 5 fields }	Object	
N 🖂 ek utissaun	· · · · · · · · · · · · · · · · · · ·	(Sincida)		
Logs				

Name	RTM002. Store smart meter data to long-term DB				
Module under test	RT Monitor Resp. ETRA				
	HL-UC 7_PUC_2_Dynamic aggregation of distributed energy assets and active participation into energy market				
Test environment	SMX running and sending information to IOP IOP platform up and running				
Features to be tested	Data from SMX is properly collect <mark>ed i</mark> n the long-term database of WiseCOOP (big data)				
Features not to be tested					
Preparation					
Dependencies	RTM001. Read smart meter data from IOP				
Steps	The execution of this test must h <mark>appe</mark> n automatically upon publication of data in the IOP				
Pass criteria	Data from the SMX is correctly appended to the historic registry held in the long-term database				
Suspension criteria					
Results	Test successful. The following screenshot shows how the long-term DB is populated with the history data from a lab-testing environment SMX				





\land Robo 3T - 1.2	CONTRACTOR OF THE OWNER	40	
File View Options Window Help			
🖩 🗝 🔛 🕨 🖿 🖏			
🔺 🗏 wisegridpre (13)	db.getCollection('wisec ×		
V L System	wisegridpre wisegridpre.lab.id:27017	indata winomon	
WiseCORP_EMS	tCollection('wisecoop values').fi		timostamp: -1))
WiseCORP_Server		Ma({1d: 2100044540052.}).SOLC({	
Image: Second	wisecoop_values 🕔 0.097 sec.		4 0 50 🕨 🛅 🕅 🗎
b log rb_bigdata_wgcockpit	Key	Value	Туре
Image: A state of the state	(1) ObjectId("5aec51ba88c0bb0006759b9	{ 7 fields }	Object
 Collections (14) 	id	ObjectId("5aec51ba88c0bb0006759b98")	ObjectId
calendar	Readings	[12 elements]	Array
meteor_accounts_loginSe	Measurements	{ 12 fields }	Object
updates	timestamp	2018-07-19 12:00:00.000Z	Date
▷ 🛄 users	id id	ZIV0044540032	String
wisecoop_energymix	operation	change	String
wisecoop_energymixfore	■ updatedTime	2018-05-04 12:27:38.901Z	Date
wisecoop_values	 (2) ObjectId("5aec51ba88c0bb0006759b9 	{ 7 fields }	Object
wisecoop_values_15m	id	ObjectId("5aec51ba88c0bb0006759b97")	ObjectId
wisecoop_values_1d	Readings	[12 elements]	Array
wisecoop_values_1h	Measurements	{ 12 fields }	Object
wisecoop_values_1h_tst	timestamp	2018-07-19 11:00:00.000Z	Date
wisecoop_values_clusters	··· id	ZIV0044540032	String
wisecoop_weather	· operation	change	String
wisecoop_weatherforecast	C updatedTime	2018-05-04 12:27:38.892Z	Date
Functions	(3) ObjectId("5aec51ba88c0bb0006759b9	{ 7 fields }	Object
Users	(4) ObjectId("5aec51ba88c0bb0006759b9	{ 7 fields }	Object
b logdata_wisecorp	(5) ObjectId("5aec51ba88c0bb0006759b9		Object
b log rb_bigdata_wiseevp	6) ObjectId("5aec51ba88c0bb0006759b9		Object
Image: Book in the second s	(7) ObjectId("5aec51ba88c0bb0006759b9		Object
b I rb_wisecoop	(8) ObjectId("5aec51ba88c0bb0006759b9)		Object
Logs			

3.2.2 KPI engine tests

Name	KPI001. Individual energy delta calculation					
Module under test	KPI engine Resp. ETRA					
Module requirement	HL-UC 7_PUC_2_Dynamic aggregation of distributed energy assets and active participation into energy market					
	SMX running and sending information to IOP IOP platform up and running RT monitor storing smart meter readings in long-term database Spark server up and running					
	Smart meters provide information of the total accumulated energy demand/production. The system therefore needs to calculate the energy deltas across consecutive readings in order to properly monitor the energy demand/production profiles. Three different aggregations of the deltas are considered: quarterly, hourly and daily.					
Features not to be tested						
Preparation						
Dependencies						
Steps	 Execute Spark job for quarterly delta calculation Execute Spark job for hourly delta calculation Execute Spark job for daily delta calculation Manually inspect long-term database collections with processed data 					





Pass criteria	aggregatio	ns		e quarterly, hourly and daily e energy deltas for the given
Suspension criteria				
Results	Test successful. The following (wisecoop_values_ deltas calculated. Robo 31 - 1.2 File View Options Window Help WiseCORP_Server System WiseCORP_Server Collections (14) Collections (14) Collections (14) Collections (14) Collections (14) Collections (14) Wisecoop_values_1 Wisecoop_	screenshot sho 15m 1h 1d) as well a 15m 1h 1d as well as	is an example of the	e active and reactive energy

Name	KPI002. Portfolio data aggregation	n			
Module under test	KPI engine		Resp.	ETRA	
Module requirement	HL-UC 7_PUC_2_Dynamic aggregation of distributed energy assets and active participation into energy market				
Test environment	SMX running and sending information to IOP IOP platform up and running RT monitor storing smart meter readings in long-term database Spark server up and running				
Features to be tested	An overview of the demand and production metrics for the whole portfolio is required				
Features not to be tested					
Preparation					





Dependencies	KPI001. Individual e	energy delta calculation	<u>ו</u>			
Steps	 Execute Spark job for quarterly delta calculation (portfolio) Execute Spark job for hourly delta calculation (portfolio) Execute Spark job for daily delta calculation (portfolio) Manually inspect long-term database collections with processed data 					
Pass criteria	 The long-term database contains 3 collections with processed data The long-term database contains 3 collections with the quarterly, hourly and daily aggregations Each collection contains documents for a virtual smart meter named "PORTFOLIO" that represent the energy deltas for the given period of the overall portfolio 					
Suspension criteria						
Results	-	<pre>enshot shows an examp lues of the whole portfo * db.getCollection(Wisec X wisegridpre = wisegridpre.lab.id:27017 * db_b dc .getCollection(*viseconp_values wiseccop_values_1h @ 2.17 sec. Key > @ (1) PORTFOLIO2018-07-19T120000000+ = (2) PORTFOLIO2018-07-19T120000000+ = (3) PORTFOLIO2018-07-19T100000000+ = (4) PORTFOLIO2018-07-19T100000000+ = (5) PORTFOLIO2018-07-19T100000000+ > @ (5) PORTFOLIO2018-07-19T100000000+ > @ (6) PORTFOLIO2018-07-19T100000000+ > @ (6) PORTFOLIO2018-07-19T00000000+ > @ (6) PORTFOLIO2018-07-19T00000000+ > @ (6) PORTFOLIO2018-07-19T050000000+ > @ (9) PORTFOLIO2018-07-19T050000000+ > @ (10) PORTFOLIO2018-07-19T050000000+ > @ (2) PORTFOLIO2018-07-19T0300000000 > @ (12) PORTFOLIO2018-07-19T0000000000 > @ (12) PORTFOLIO2018-07-19T000000000 > @ (12) PORTFOLIO2018-07-19T0000000000 > @ (12) PORTFOLIO2018-07-19T0000000000 > @ (12) PORTFOLIO2018-07-19T0000000000 > @ (12) PORTFOLIO2018-07-19T0000000000 > @ (13) PORTFOLIO2018-07-19T0000000000 > @ (14) PORTFOLIO2018-07-19T0000000000 > @ (15) PORTFOLIO2018-07-19T0000000000 > @ (15) PORTFOLIO2018-07-19T000000000000000000000000000000000000</pre>	Diio.			

Name	KPI003. Portfolio profiling, demand-related behaviour				
Module under test	KPI engine Resp. ETRA				
	HL-UC 7_PUC_2_Dynamic aggregation of distributed energy assets and active participation into energy market				
	Historical readouts for portfolio members stored in the long-term database Spark server up and running				
	All portfolio members get classified based on the distribution of the energy demand in the following time slots:				





·					
	- Working days, 08h-18h				
	- Working days, 18h-08h				
	-	ng days, 08h-18h			
	- Non-worki	ng days, 18h-08h			
Features not to					
be tested					
Preparation					
Dependencies	KPI001. Individual (energy delta calculati	on		
Steps		ark job for clusters canspect operational date		ection for results	
Pass criteria		ional database conta ional database conta			
Suspension criteria					
	of the demand pro	db.getCollection(Wisecoo x wisegridpre Wisecool x wisegridpre Wisegridpre.lab.id:27017	rb_wisecoop		
	_messages	db.getCollection('wisecoop_clu	stering').lind({})	4 0	50 🕨 🛐 📄 🗖
	▷profiles ▷schema	wisecoop_clustering () 0.007 sec.	Value	∢ o Type	
	▷	 key (1) 5b0ed71c0ee51a4bd09989db 	{ 5 fields }	Object	
	meteor_accounts_loginSe	▲ 😡 (2) 5b1bbab48d6d6c32ac7ed83f	{ 5 fields }	Object	
	users wisecoop_clustering	id _id	5b1bbab48d6d6c32ac7ed83f	String	
Results	wisecoop_contracts	😁 type 👼 timestamp	demand_v1 2018-06-09 11:32:04.211Z	String Date	=
	wisecoop_demandForeca	 ClusterCenters 	[3 elements]	Array	=
	 wisecoop_energymix wisecoop_energymixfore 	⊿ © [0]	{ 5 fields } 1693.62962962963	Object Double	
	wisecoop_flexOffers	-	1000/02002002000		
		■ wd_p2	1591.2962962963	Double	
	wisecoop_flexOrders	m nwd_p1	780.925925925926	Double	
		em nwd_p1 em nwd_p2	780.925925925926 1591.18518518518	Double Double	
	wisecoop_flexOrders wisecoop_flexRequests wisecoop_productionFor wisecoop_rules	m nwd_p1	780.925925925926 1591.18518518518 [27 elements] { 5 fields }	Double	
	 wisecoop_flexOrders wisecoop_flexRequests wisecoop_productionFor wisecoop_rules wisecoop_spotprices 	w nwd_p1 w nwd_p2 ▷ w members w wd_p1 w wd_p1	780.925925925925 1591.18518518518 [27 elements] (5 fields) 3988.38095238095	Double Double Array Object Double	
	 wisecoop_flexOrders wisecoop_flexRequests wisecoop_rolectionForm wisecoop_routes wisecoop_spotprices wisecoop_spotprices wisecoop_usagePointLoc 	∞ nwd_p1 ∞ nwd_p2 ▷ © members ⊿ © [1]	780.925925925926 1591.18518518518 [27 elements] { 5 fields }	Double Double Array Object	
	wisecoop_flexOrders wisecoop_flexRequests wisecoop_roductionFor wisecoop_spotprices wisecoop_spotprices wisecoop_rariffs wisecoop_rariffs wisecoop_values		780.925925925926 1591.18518518518 [27 elements] (5 fields) 3988.38095238095 4167.52380952381	Double Double Array Object Double Double	
	 wisecoop_flexOrders wisecoop_flexRequests wisecoop_productionFor wisecoop_spotprices wisecoop_spotprices wisecoop_tariffs wisecoop_tariffs wisecoop_vagePointLoc wisecoop_wather 	wwd_p1 wmwd_p2 wmwd_p2 wwd_p2 wwd_p1 wwd_p1 wwd_p1 wwd_p1 wwd_p1 wmd_p2 wmmemers	780.925925925926 1591.18518518518 [27 elements] (5 fields) 3988.38095238095 4167.52380952381 2111.52380952381 5734.23809523809 [21 elements]	Double Double Array Object Double Double Double Array	
	wisecoop_flexOrders wisecoop_flexRequests wisecoop_roductionFor wisecoop_spotprices wisecoop_spotprices wisecoop_rariffs wisecoop_rariffs wisecoop_values	wwd_p1 mwd_p2 w members wd_p2 w members wd_p1 w wd_p2 w mwd_p1 w mwd_p1 w mwd_p2 w members w members w mod	780.925925925926 1591.18518518518 [27 elements] (5 fields) 3968.38095238095 4167.52380952381 2111.52380952381 5734.2380952380	Double Double Array Object Double Double Double Double	
	wisecoop_flexOrders wisecoop_flexRequests wisecoop_productionFor wisecoop_spotprices wisecoop_spotprices wisecoop_spotprices wisecoop_spotprices wisecoop_spotprices wisecoop_wagePointLoc wisecoop_weather wisecoop_weather wisecoop_weather wisecoop_weatherforecast Lording Users	wwd_p1 wmwd_p2 wmwd_p2 wwd_p2 wwd_p1 wwd_p1 wwd_p1 wwd_p1 wwd_p1 wmd_p2 wmmemers	780.925925925926 1591.18518518518 [27 elements] (5 fields) 3988.38095238095 416.752380952381 2111.52380952381 5734.2380952381 5734.2380952380 [21 elements] ZIV004403478	Double Array Object Double Double Double Array String	
	 wisecoop_flexOrders wisecoop_flexRequests wisecoop_roductionFor wisecoop_patprices wisecoop_sapePointLoc wisecoop_values wisecoop_weather wisecoop_weatherforecast Functions 	wid_p1 mwd_p2 wid_p2 wid_p1 wid_p2 wid_p2 wid_p2 wid_p2 mwd_p1 wid_p2 mwd_p2 im mwd_p2 iim mwd_p3 iim mwd_p4 iim mwd_p4	780.925925925926 1591.18518518518 [27 elements] (5 fields) 3988.38095238095 4167.52380952381 2111.52380952381 5734.2380952380 [21 elements] ZTV0040403478 ZTV0040403483	Double Double Array Object Double Double Double Array String String	

Name	KPI004. Portfolio profiling, production-related	behaviour	
Module under test	KPI engine	Resp.	ETRA
	HL-UC 7_PUC_2_Dynamic aggregation of participation into energy market	distributed ener	rgy assets and active
Test	Historical readouts for portfolio members stor	red in the long-terr	n database





environment	Spark server up and running				
Features to be tested	All portfolio members get classified based on their daily production				
Features not to be tested					
Preparation					
Dependencies	KPI001. Individual energy delta calculation				
Steps	 Execute Spark job for clusters calculation Manually inspect operational database clusters collection for results 				
Pass criteria	 The operational database contains the cluster centres for each identified group (average daily production) The operational database contains the list of members for each identified group 				
Suspension criteria					
	The following screenshot of the operational database shows the results of the execution of the demand profiling module.				
Results	key Value Type cronHistory imetor_accounts_loginSe Object users j.d Sboed71c0ee51a4bd09989db (5 fields) Object wisecoop_clustering poduction_v1 String Object wisecoop_clustering clusterCenters [2 elements] Array wisecoop_energymix OI [2 fields] Object wisecoop_flex/Orders [69 elements] Array wisecoop_flex/Orders [69 elements] Array wisecoop_flex/Orders [1] [2 fields] Object wisecoop_flex/Orders [69 elements] Array wisecoop_flex/Orders [69 elements] Array wisecoop_flex/Orders [2 element] Array wisecoop_flex/Orders [2 element] Array wisecoop_flex/Orders [2 element] Array wisecoop_values [2 element]				

Name	KPI005. Portfolio profiling, monthly CO ₂ emissions				
Module under test	KPI engine	Resp.	ETRA		
	HL-UC 7_PUC_2_Dynamic aggregation of participation into energy market	distributed ener	rgy assets and active		
Test	Historical readouts for portfolio members stor	red in the long-terr	n database		





environment	Spark server up and running				
Features to be tested	All portfolio members get classified based on the monthly equivalent CO_2 emissions				
Features not to be tested					
Preparation					
Dependencies	KPI001. Individual energy delta o	alculation			
Steps	 Execute Spark job for clu Manually inspect operat 		ters collection f	or results	
Pass criteria	 The operational database (average monthly equivational database) The operational database 	alent CO ₂ emissions)		
Suspension criteria					
Results	The following screenshot of the of the CO ₂ -based profiling modu	Image: Collection(wisecoo×) Image: Wisegridgre_lab.id:22017 db.getCollection(*Wisecoop) Image: Wisegridgre_lab.id:22017 Image: Wisegridgre_lab.id:22017	<pre>tb_wisecoop clustering').find(()) Value (5 fields) (5 fields)</pre>	Vpet Object	
nesuits	j.profiles j	 (3) co2_y1 (j) co2_y1 (j) type timestamp (c) co2Entissions (c) members (l) (c) members (l) (c) co2Entissions (c) members 	(5 fields) co2_v1 co2_v1 2018-07-25 13:05:52:369Z [3 elements] (2 fields) 706:427166739829 [34 elements] (2 fields) 5097:34058807193 [3 elements] (2 fields) 2128:00813436411 [17 elements] Average daily CO2 emissions	Object String Date Array Object Double Array Object Double Array Object Double Array String	

Name	KPI006. Portfolio profiling, monthly cost		
Module under test	KPI engine Re	tesp. ETRA	
	HL-UC 7_PUC_2_Dynamic aggregation of di participation into energy market	distributed energy assets and ac	tive
	Historical readouts for portfolio members stored Spark server up and running	d in the long-term database	





_					
Features to be	All portfolio members get classified based on the economic cost associated to their				
tested	demand				
Features not to be tested					
Preparation					
Dependencies	KPI001. Individual energy delta d	calculation			
Steps	 Execute Spark job for clu Manually inspect operat 			for results	
Pass criteria	- The operational databas (average monthly costs) - The operational databas				
Suspension criteria					
	The following screenshot of the of the cost-based profiling modu Rebo 37-12 File View Options Window Help 		db.getCollection('wisec × 27017 ⊜ rb_wiseccop	esults of the execution	
	cronHistory incidents items	wisecoop_clustering () 0.003 sec.	—	 0 50 E E E 	
Results	meteor_accounts_loginServiceConfiguration users wgcockpit_config wgcockpit_data wgcockpit_demandResponse wgcockpit_flexOffers wgcockpit_gradData wgcockpit_meters wgcockpit_neters wgcockpit_setematics wgcockpit_setematics wgcockpit_simulations wgcockpit_substations wgcockpit_substations	Key ▷ □□ (3) co2_v1 □ □ (4) cost_v1 □ □ (1) □ □ □ (1) □ □ □ (1) □ □ □ (1) □ □ □ (2) □ □ □ cost □ □ members □ □ (2) □ □ □ cost □ □ members □ □ (3) □ □ □ cost □ □ members □ □ cost □ □ cost □ □ cost □ □ cost □ □ cost	Value (5 fields) (5 fields) (cost_v1 cost_v1 cost_v1 cost_v1 cost_v1 (2 fields) (2 field	Type Object Object String String Date Array Object Double Array String T 	

Name	KPI007. Calculation of average profiles per clusterization and cluster			
Module under test	KPI engine ETRA			
	HL-UC 7_PUC_2_Dynamic aggregation of distributed energy assets and active participation into energy market			
	Historical readouts for portfolio members stored in the long-term database Spark server up and running			
	For each group of each clusterization, the <i>average member</i> is computed, in order to enable comparison of individual behaviour with the group of similar individuals			





be tested							
Preparation							
	KPI001. Individual	energy delta calculatio	n				
	KPI003. Portfolio p	rofiling, demand-relat	ed behaviour				
Dependencies	KPI004. Portfolio p	rofiling, production-re	lated behaviour				
	KPI005. Portfolio p	rofiling, monthly CO ₂ e	emissions				
		rofiling, monthly cost					
Steps		nspect long-term data	• •				
	- The long-t	erm database contai	ns registries for or	ne virtu <mark>al s</mark> mart meter per			
Pass criteria	clusterizati	on and group					
rass citteria	 Those sma 	rt meter contai <mark>n t</mark> he a	veraged values for o	energy demand, production,			
	cost and er	nissions of all m <mark>em</mark> be	rs of the group				
Suspension							
criteria							
	Test successful.						
	The following scree	enshot of the l <mark>ong-t</mark> err	n database shows th	a collection with the results			
		U		ie conection with the results,			
	as well as an exam			ind-based clusterization.			
	as well as an exam						
	Robo 3T - 1.2 File View Options Window Help			nd-based clusterization.			
	▲ Robo 3T - 1.2 File View Options Window Help ↓ ↓ ↓ ↓ ↓ ↓ ↓	ple registry fo <mark>r the f</mark> irs		nd-based clusterization.			
	Robo 3T - 1.2 File View Options Window Help Image: Comparison of the state	ple registry for the firs	t group of the dema	nd-based clusterization.			
	Robo 3T - 1.2 File View Options Window Help Image: Complex of the system Image: System Image: WiseCORP_EMS	ple registry fo <mark>r the f</mark> irs	t group of the dema	ind-based clusterization.			
	Robo 3T - 1.2 File View Options Window Help	ple registry for the firs	t group of the dema	ind-based clusterization.			
	Robo 3T - 1.2 File View Options Window Help Wisegridpre (13) System Wisegridpre (13) Wisegridpre (13) WiseCORP_Server WiseCORP_Server WisecoRP_Server Style WisecoRP_Server Style rb_bigdata_wgcockpit	<pre>ple registry for the firs * db.getCollection('wisec x * wisegridpre = wisegridpre.lab.di:27017 @ rb db.getCollection('wiseccop_value wiseccop_values_dusters_1h @ 2.29 sec. Key</pre>	t group of the dema	<pre>ind-based clusterization. ind_v1_0').sort((timestamp:-1))</pre>			
	Robo 3T - 1.2 File View Options Window Help	ple registry for the firs	Lbigdata_wiseccoop rs_clusters_lh').find((id:'dema Value (6 fields)	<pre>ind-based clusterization. ind_v1_0')).sort((timestamp:-1))</pre>			
	Robo 3T - 1.2 File View Options Window Help WiseCorp WiseCORP_EMS WiseCORP_Server WiseCORP_Server Collections (14) Collections (14) Collections (14)	<pre>ple registry for the firs * db.getCollection('wisec x * wisegridpre = wisegridpre.lab.di:27017 @ rb db.getCollection('wiseccop_value wiseccop_values_dusters_1h @ 2.29 sec. Key</pre>	t group of the dema	<pre>ind-based clusterization. ind_v1_0').sort((timestamp:-1))</pre>			
Results	Robo 3T - 1.2 File View Options Window Help View Options View Options View Options	<pre>ple registry for the firs * db.getCollection('wisec x * wisegrdpre * wisegrdpre.lab.id:27017 * ft db.getCollection('wiseccop_value wiseccop_value wiseccop_value dusters_1h * 2.29 sec. Key * * (1) demand_v1_0_2018-07-19 140000 * * (2) demand_v1_0_2018-07-19 130000</pre>	<pre>bigdota_wiseccop bg_clusters_lh').find((id:'dema Value (6 fields) (6 fields)</pre>	<pre>ind-based clusterization. ind_v1_0').sort((timestamp:-1)) ind_v1_0') sort((timestamp:-1)) ind_v1_0') ind_v1_0' ind_v1_0') ind_v1_0' ind_v1_0') ind_v1_0' ind_v1_0') ind_v1_0' ind_v1_0') ind_v1_0' ind_v1_0' ind_v1_0') ind_v1_0' ind_v</pre>			
Results	Robo 3T - 1.2 File View Options Window Help WiseCORP_IS WiseCORP_Server Collections (14) Collections (<pre>ple registry for the firs ple registry for the firs * db.getCollection('wisec x * wisegrdpre * wisegrdpre.lab.ld:27017 * ft db.getCollection('wiseccop_value wiseccop_value dusters_1h * 2.29 sec. Key * @ (1) demand v1_0_2018-07-19 14:00:00 * @ (2) demand_v1_0_2018-07-19 13:00:00 * @ (3) demand_v1_0_00 * @ (3) demand_v1_00 * @ (3) demand_v1_00</pre>	t group of the dema bigdsta_wiseccop rs_clusters_th').find((id:'dema Value (6 fields) (6 fields) demand_y1_0_2018-07-19 13:00:00 demand_y1_0 2018-07-19 11:00:00.000Z	<pre>ind-based clusterization. ind_v1_0').sort((timestamp:-1))</pre>			
Results		<pre>ple registry for the firs ple registry for the firs * db.getCollection('wisec x * wisegridpre = wisegridpre.lab.id:27017 = ft db.getCollection ('wiseccop_value wiseccop_values_dusters_1h @ 2.29 sec. Key * @ (1) demand_v1_0_2018-07-19 14:00:00 * @ (2) demand_v1_0_2018-07-19 13:00:00 * @ (2) demand_v1_0_2018-07-19 13:00:00 * @ (1) demand_v1_0_2018-07-19 13:00:00 * @ (1) demand_v1_0_2018-07-19 13:00:00 * @ (2) demand_v1_0_2018-07-19 13:00:00 * @ (2) demand_v1_0_2018-07-19 13:00:00 * @ (2) demand_v1_0_2018-07-19 13:00:00 * @ (2) demand_v1_0_2018-07-19 13:00:00 * @ (2) demand_v1_0_2018-07-19 13:00:00 * @ (2) demand_v1_0_2018-07-19 13:00:00 * @ (2) demand_v1_0_2018-07-19 13:00:00 * @ (2) demand_v1_0_2018-07-19 13:00:00 * @ (2) demand_v1_0_2018-07-19 13:00:00 * @ (2) demand_v1_0_2018-07-19 13:00:00 * @ (2) demand_v1_0_2018-07-19 10-100 * @ (2) demand_v1_0_2018-07-100 * @ (2) d</pre>	bigdata_wiseccop bigdata_wiseccop big_clusters_1h').find ({id:'demain Value (6 fields) (6 fields) (6 fields) demand_v1_0_2018-07-19 13:00:00 demand_v1_0 2018-07-19 13:00:002 (3 fields)	Ind-based clusterization.			
Results	Robo 3T - 1.2 File View Options Window Help WiseCORP_IS WiseCORP_Server Collections (14) Collections (<pre>ple registry for the firs * db.getCollection(wisec x * wisegridpre * wisegridpre.lab.id:27017 * rb db.getCollection('wiseccop_value wiseccop_values_dusters_1h * 2.29 sec. Key * * (1) demand v1 0_2018-07-19 14:00:00 * * (2) demand v1 0_2018-07-19 13:00:00 * id timestamp * * Measurements * Active_Energy_+_Incr</pre>	t group of the dema bigdata_wiseccop s_clusters_th') .find((id:'dema Value (6 fields) demand_v1_0_2018-07-19 13:00:00 demand_v1_0 2018-07-19 13:00:00 2018-07-19 13:00:00 2018-07-19 13:00:00 2018-07-19 13:00:00 2018-07-19 13:00:00 2018-07-19 13:00:00 2018-07-19 13:00:00 2018-07-19 13:00:00 2018-07-10 2018-07-19 13:00 2018-07-10 2	Ind-based clusterization.			
Results	Robo 3T - 1.2 File View Options Window Help View Options View Op	<pre>ple registry for the firs ple registry for the firs * db.getCollection('wisec x * visegridpre visegridpre visegridpre.lab.id:27017 to db.getCollection ('wiseccoop_value viseccop_value viseccop_values_dusters_1h 2.29 sec. Key * (2) demand_v1_0_2018-07-19 14:00:00 * (2) demand_v1_0_2018-07-19 13:00:00 * (2) demand_v1_0_2018-07-19 13:00:00 * (3) d * (4) demand_v1_0_2018-07-19 13:00:00 * (4) demand_v1_0_2018-07-19 13:00:00 * (5) d * (5) demand_v1_0_2018-07-19 13:00:00 * (5) d * (5) demand_v1_0_2018-07-19 13:00:00 * (5) d * (5) d</pre>	t group of the dema bigdsta_wiseccop rs_clusters_th') find ((id: 'dema (6 fields) (6 fields) demand_y1_0_2018-07-19 13:00:00 demand_y1_0 2018-07-19 11:00:00.000Z (3 fields) 284.962962962963 0.037037037037037037	Ind-based clusterization.			
Results	Robo 3T - 1.2 File View Options Window Help View Options View Options Window Help View Options View Options View Options View Options View Options View Options View Options View Options View Options View Options View Options View Options View	<pre>ple registry for the firs * db.getCollection(wisec x * wisegridpre * wisegridpre.lab.id:27017 * rb db.getCollection('wiseccop_value wiseccop_values_dusters_1h * 2.29 sec. Key * * (1) demand v1 0_2018-07-19 14:00:00 * * (2) demand v1 0_2018-07-19 13:00:00 * id timestamp * * Measurements * Active_Energy_+_Incr</pre>	t group of the dema bigdata_wiseccop s_clusters_th') .find((id:'dema Value (6 fields) demand_v1_0_2018-07-19 13:00:00 demand_v1_0 2018-07-19 13:00:00 2018-07-19 13:00:00 2018-07-19 13:00:00 2018-07-19 13:00:00 2018-07-19 13:00:00 2018-07-19 13:00:00 2018-07-19 13:00:00 2018-07-19 13:00:00 2018-07-10 2018-07-19 13:00 2018-07-10 2	<pre>ind-based clusterization. ind_v1_0').sort((timestamp:-1))</pre>			
Results	Robo 3T - 1.2 File View Options Window Help Wisegridpre (13) System WiseCORP_ENS WiseCORP_Server Collections (14) Collections (14) Collections (14) updates updates wisecoop_energymix wisecoop_energymixfore wisecoop_energymixfore wisecoop_aulees wisecoop_values_15m wisecoop_values_11d	<pre>ple registry for the firs ple registry for the firs</pre>	t group of the dema bigdata_wisecoop ts_clusters_lh').find({id:'dema value (6 fields) demand_v1_0_2018-07-19 13:00:00 demand_v1_0_2018-07-19 13:00:00 demand_v1_0_2018-07-19 13:00:00 (3 fields) 2018-07-19 11:00:00.000Z (3 fields) 2018-07-19 11:00:00.000Z (3 fields) 284:0520520520530 0.037037037037037 88.6764206570707	<pre>ind-based clusterization. ind_v1_0').sort((timestamp:-1))</pre>			
Results	Robo 3T - 1.2 File View Options Window Help WiseCORP_IS WiseCORP_EMS WiseCORP_Server Gonfig tr_bigdata_wgcockpit tr_bigdata_wgcockpit Collections (14) Collections (14) Collections (14) Gendar wisecoop_energymix wisecoop_energymix wisecoop_energymix wisecoop_energymix wisecoop_values_15m wisecoop_values_11d wisecoop_values_1h	<pre>ple registry for the firs ple registry for the firs</pre>	t group of the dema bigdata_wiseccop clusters_lh') find ({id:'dema value (6 fields) (6 fields) demand_v1_0_2018-07-19 13:00:00 demand_v1_0_2018-07-19 13:00:00 demand_v1_0_2018-07-19 13:00:00 (3 fields) 2018-07-19 11:00:00.0002 (3 fields) 2018-07-19 11:00:00.0002 (3 fields) 284:062962962963 0.037037037037037 88:6764206570707 (4 fields) [2 elements] (6 fields)	Ind-based clusterization.			
Results	Robo 3T - 1.2 File View Options Window Help Wisegridpre (13) System WiseCORP_ENS WiseCORP_Server Collections (14) Collections (14) Collections (14) updates updates wisecoop_energymix wisecoop_energymixfore wisecoop_energymixfore wisecoop_aulees wisecoop_values_15m wisecoop_values_11d	<pre>ple registry for the firs ple registry for the firs</pre>	t group of the dema bigdata_wisecoop ts_clusters_th').find((id:'dema Value (6 fields) (6 fields) (6 fields) 284.962962962963 0037037037037037037 88.676420557007 (4 fields) [2 fields] [2 fields] [Ind-based clusterization.			
Results	Robo 3T - 1.2 File View Options Window Help View Options View Options Window Help View Options V	<pre>ple registry for the firs ple registry for the firs</pre>	t group of the dema bigdst_wieccop bg_clusters_lh') find((id:'dema value (6 fields) (6 fields) demand_v1_0_2018-07-19 13:00:00 demand_v1_0 2018-07-19 11:00:00.000Z (3 fields) 204.962:082:092:093 0.037037037037037 88.6764206570707 (4 fields) [2 elements] (6 fields) (6 fields)	Ind-based clusterization. Ind_v1_0').sort((timestamp:-1)) 0 50 Type Object Object Double Double Double Object Array Object			
Results	Robo 3T - 1.2 File View Options Window Help View Options Wiew Op	<pre>ple registry for the firs ple registry for the firs</pre>	t group of the dema bigdata_wisecoop ts_clusters_th').find((id:'dema Value (6 fields) (6 fields) (6 fields) 284.962962962963 0037037037037037037 88.676420557007 (4 fields) [2 fields] [2 fields] [Ind-based clusterization.			
Results	Robo 3T - 1.2 File View Options Window Help Wisegridpre (13) System WiseCORP_ENS WiseCORP_Server Gonfig Grong Collections (14) Collections (14) Collections (14) Godecong_energymix wisecoop_energymixfore wisecoop_aules_15n wisecoop_values_1nt wisecoop_values_1nt wisecoop_values_1nt wisecoop_watherforecast wisecoop_watherforecast Functions	<pre>ple registry for the firs ple registry for the firs</pre>	t group of the dema bigdata_wiseccop clusters_lh') find (fid: 'dema value (6 fields) (6 fields) demand_v1_0_2018-07-19 13:00:00 demand_v1_0 2018-07-19 11:00:00.000Z (3 fields) 2018-07-19 11:00:00.000Z (3 fields) 2018-07-19 11:00:00.000Z (3 fields) 2018-07-19 11:00:00.000Z (3 fields) 24:4629629629623 0.0370370370370370 88:6764206570707 (4 fields) (6 fields)	and-based clusterization.			
Results	Robo 3T - 1.2 File View Options Window Help Wisegridpre (13) System WiseCORP_ENS WiseCORP_ENS WiseCORP_Server Collections (14) Collections (14) Collections (14) updates wisecoop_energymix wisecoop_energymix wisecoop_values_1 wisecoop_values_15m wisecoop_values_11h wisecoop_values_11h wisecoop_values_11h wisecoop_values_11h wisecoop_values_11h wisecoop_values_11h wisecoop_values_11h wisecoop_walter wisecoop_values_1h wisecoop_values_1h wisecoop_values_1h wisecoop_values_1h wisecoop_values_1h wisecoop_values_1h wisecoop_walter wisecoop_walter	<pre>ple registry for the firs ple registry for the firs</pre>	t group of the dema bigdata_wiseccop clusters_lh') find (fid: 'dema value (6 fields) (6 fields) demand_v1_0_2018-07-19 13:00:00 demand_v1_0_2018-07-19 13:00:00 demand_v1_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_	and-based clusterization.			
Results	Robo 3T - 1.2 File View Options Window Help View Options View Option View Options View Option View Options View Option View Options View Options	<pre>ple registry for the firs ple registry for the firs * db.getCollection('wisec x * visegridpre = visegridpre.lab.id:27017 = rh db.getCollection('wisecoop_value visecoop_values_dusters_lh @ 2.29 sec. Key * @ (1) demand_v1_0_2018-07-19 13:00:00 # @ (2) demand_v1_0_2018-07-19 13:00:00 # @ (2) demand_v1_0_2018-07-19 12:00:00 * @ Meter * @ Readings_Doll8-07-19 12:00:00 * @ (4) demand_v1_0_2018-07-19 11:00:00 * @ (5) demand_v1_0_2018-07-19 11:00:00 * @ (6) demand_v1_0_2018-07-19 10:00:00 * @ (7) demand_v1_0_2018-07-19 08:00:00 * @ (9) demand_v1_0_2018-07-19 08:00:00 * @ (1) demand_v1_0_000 * @ (1) demand_v1_0_0000 * @ (1) demand_v1_00000 * @ (1) demand_v1_000000 * @ (1) demand_v1_000000 * @ (1) demand_v1_000000 * @</pre>	t group of the dema bigdata_wiseccop big_clusters_1h').find ((id:'dema value (6 fields) (6 fields) (6 fields) (6 fields) 2018-07-19 11:00:00.000Z (3 fields) 284.962962962963 0.037037037037037 88.6764206570707 (4 fields) (2 fields) (6 fields) (7 fields)	Ind-based clusterization.			
Results	Robo 3T - 1.2 File View Options Window Help Wisegridpre (13) System WiseCORP_ENS WiseCORP_ENS WiseCORP_Server Collections (14) Collections (14) Collections (14) updates wisecoop_energymix wisecoop_energymix wisecoop_values_1 wisecoop_values_15m wisecoop_values_11h wisecoop_values_11h wisecoop_values_11h wisecoop_values_11h wisecoop_values_11h wisecoop_values_11h wisecoop_values_11h wisecoop_walter wisecoop_values_1h wisecoop_values_1h wisecoop_values_1h wisecoop_values_1h wisecoop_values_1h wisecoop_values_1h wisecoop_walter wisecoop_walter	<pre>ple registry for the firs ple registry for the firs</pre>	t group of the dema bigdst_wisecop bg_clusters_th') find ({id: 'dema 'sg_clusters_th') find ({id: 'sg_clusters_th'}) find ({id: '	Ind-based clusterization. Ind_v1_0').sort((timestamp:-1)) 0 50 Type Object Object Object Date Object Double Double Object			
Results	Robo 3T - 1.2 File View Options Window Help Wisegridpre (13) WiseCORP_ENS WiseCORP_ENS WiseCORP_Server Collections (14) calendar meteor_accounts_loginSe updates wisecoop_energymixfore wisecoop_values_ln wisecoop_values_lnt wisecoop_values_lnt wisecoop_values_lnt wisecoop_values_lnt wisecoop_values_lnt wisecoop_watherforecast	ple registry for the firs ple registry for the firs * db.getCollection(wisec × wisegridpre	t group of the dema bigdst_wisecoop 	and-based clusterization.			
Results	Robo 3T - 1.2 File View Options Window Help Wisegridpre (13) System WiseCORP_ENS WiseCORP_Server Grifig Grib_tigdata_wisecop Collections (14) Collections (14) Collections (14) Gridpata_wisecop_energymix Wisecoop_energymix Wisecoop_energymix Wisecoop_values_1 Wisecoop_values_15m Wisecoop_values_1h Wisecoop_values_1h Wisecoop_values_1h Wisecoop_walues_1h Wisecoop_walues_1h Wisecoop_walues_1h Wisecoop_walues_1h Wisecoop_values_1h Wisecoop_walues_1h Wisecoop_walues_lh Wisecob_walues_lh Wisecoop_walues_lh Wisecoop_walues_lh Wis	<pre>ple registry for the firs ple registry for the firs</pre>	t group of the dema bigdst_wisecop bg_clusters_th') find ({id: 'dema 'sg_clusters_th') find ({id: 'sg_clusters_th'}) find ({id: '	Ind-based clusterization. Ind_v1_0').sort((timestamp:-1)) 0 50 Type Object Object Object Date Object Double Double Object			

3.2.3 Forecast modules tests

Name	FOR001. Demand/production forecasting training			
Module under test	WiseCOOP forecast module	Resp.	ITE	





	HL-UC 7_PUC_2_Dynamic aggregation of distributed energy assets and active participation into energy market			
Test environment	WiseCOOP forecast module up and running Historical data available in long-term DB			
Features to be tested	WiseCOOP forecast module is trained			
Features not to be tested				
Preparation				
Dependencies	RTM002			
Steps	Perform WiseCOOP forecast training			
Pass criteria	Training MAPE below pre-defined threshold			
Suspension criteria				
Results	Test successful. WiseCOOP demand forecast model trained			

Name	FOR002. Demand/Production forecasting				
Module under test	WiseCOOP for	ecast	: module	Resp.	ITE
Module requirement	HL-UC 7_PUC_2_Dynamic aggregation of distributed energy assets and active participation into energy market				
llest environment	WiseCOOP forecast module up and running Historical data available in long-term DB				
Features to be tested	WiseCOOP forecast module performs demand/production forecasting training				
Features not to be tested					
Preparation	Train WiseCOOP demand/production forecast module				
Dependencies	FOR001 RTM002				
Steps	WiseCOOP for <mark>ecast</mark> module				
Pass criteria	Prediction MAPE below pre-defined threshold				
Suspension criteria					
Results	Test successfu 24 hours hourl		gregated load and p	roduction pr	ediction

Name FOR003. Request message parsing test of WiseCOOP forecast mo	dule
---	------





Module under test	WiseCOOP forecast module	Resp.	ITE		
Module requirement	HL-UC 7_PUC_2_Dynamic aggregation active participation into energy market		d energy assets and		
Test environment	Development RabbitMQ environment WiseCOOP forecast module up and running Historical data available in long-term DB				
Features to be tested	Performance of WiseCOOP forecast module, at parsing forecast queries.				
Features not to be tested					
Preparation	Enable RabbitMQ queues, and run Wis	eCOOP forec	ast module		
Dependencies					
Steps	 Receipt of request Request parsing DB request according to the requested data Treatment of the retrieved data 				
Pass criteria	 The forecast module is able to decode the queries properly The forecast module is able to retrieve information from the long-term DB with the parsed information 				
Suspension criteria					
Results	Test successful. The module is able to parse the request messages and process it to retrieve information from the long-term DB.				

Name	OR004. Forecast response	message generation	test of WiseCOOP	
Module under test	WiseCOOP forecast module	Resp.	ITE	
Module requirement	HL-UC 7_PUC_2_Dynamic aggregation of distributed energy assets and active participation into energy market			
Test environment	RabbitMQ environment WiseCOOP forecast module up and running Historical data available in long-term DB			
Features to be tested	Performance of WiseCOOP forecast module, at generating and submitting the demand forecast response.			
Features not to be tested				
Preparation	Enable RabbitMQ queues Run the demand forecast mo	odule		





Dependencies			
Steps	 Parsing of the forecasting algorithm output Generating forecast response message 		
Pass criteria	 The forecast module is able to analyse properly the outport provided by the forecasting algorithm The forecast module is able to generate properly the forecast response message 		
Suspension criteria			
	Test successful. The module is able to analyse the information provided by the forecast algorithm, and generates the response.		

Name	FOR005. Foreca <mark>st is</mark> periodically triggered			
Module under test	Forecast orchest	rator	Resp.	ETRA
Module requirement		_Dynamic aggregation ion into energy market		d energy assets and
Test environment	Internal ESB up a Hist <mark>orical d</mark> ata a	and running vailable in long-term D	В	
Features to be tested	WiseCOOP periodically posts a demand and a production forecast request per bus to the corresponding queue of the internal ESB			
Features not to be tested				
Preparation	Open RabbitMQ monitor, monitor demand and production forecast queues			
Dependencies				
Steps	1. Execute	forecast orchestrator r	module	
Pass criteria	 Periodically, every hour, one request per smart meter appears in the demand and production forecast queues Requests claim next 24 hours hourly prediction 			
Suspension criteria				
Results	Test successful The following extract of logs of the Docker container wisecoop_forecastbridge_demand_1 shows that one forecast query for each asset is being posted every hour. etraid@wisegridpre:~\$ docker logs -ttail 100 wisecoop_forecastbridge_demand_1 grep querying grep ZIV0036406533			





2018-07-18T13:37:55.132492293Z	[ZIV0036406533]
querying	
2018-07-18T14:37:55.420413088Z	[ZIV0036406533]
querying 2018-07-18T15:37:56.411212047Z	[77370026406522]
querying	[2100030400333]
2018-07-18T16:37:54.949073578Z	[ZTV0036406533]
querying	[1110000100000]
2018-07-18T17:37:54.528547460Z	[ZIV0036406533]
querying	
2018-07-18T18:37:55.309216318Z	[ZIV0036406533]
querying	
2018-07-18T19:37:54.674470058Z	[ZIV0036406533]
querying 2018-07-18T20:37:54.560616957Z	
querying	[2100030400333]
2018-07-18 <mark>T2</mark> 1:37:54.313838874Z	[ZIV0036406533]
querying	
2018-07-18 T22 :37:54.998547669Z	[ZIV0036406 <mark>53</mark> 3]
querying	
2018-07-1 <mark>8T23</mark> :37:55.138371328Z	[ZIV003640653 <mark>3</mark>]
querying	

Name	FOR006. Forecast results are saved to operational DB			
Module under test	Forecast orchestrator Resp. ETRA			
Module requirement	HL-UC 7_PUC_2_Dynamic aggregation of distributed energy assets and active participation into energy market			
Test environment	Internal ESB up and running Historical data available in long-term DB Demand and production forecast modules up and running			
Features to be tested	WiseCOOP receives the results of the forecast module, formats them following the same format used to store real-time data, and stores the in the operational database			
Features not to be tested				
Preparation	Open oper <mark>ationa</mark> l database, query next 24 hours of demand/prod <mark>uction</mark> forecasts			
Dependencies				
Steps	1. Execute forecast orchestrator module			
Pass criteria	Periodically, every hour, next 24 hours forecast metrics get updated in the operational database			
Suspension criteria				





	Test successful
	The following extract of logs of the Docker container
	wisecoop_forecastbridge_demand_1 shows that the forecast module
	answers to the posted requests.
	etraid@wisegridpre:~\$ docker logs -ttail
	1000 wisecoop_forecastbridge_demand_1 grep
	"forecast received" grep ZIV0036406533
	2018-07-18T16:40:36.196281873Z [ZIV0036406533]
	forecast received
	2018-07-18T17:40:38.580263496Z [ZIV0036406533]
Results	forecast received
Nesults	2018-07-18T18:40:37.195192766Z [ZIV0036406533]
	forecast received 2018-07-18T19:40:39.462108051Z [ZIV0036406533]
	forecast received
	2018-07-18T20:40:41.464303878Z [ZIV0036406533]
	forecast received
	2018-07-18T21:40:37.059925621Z [ZIV0036406533]
	forecast received
	2018-07-18T22:40:22.290691821Z [ZIV0036406533]
	forecast received
	2018-07-18T23:40:10.960955447Z [ZIV0036406533]
	forecast received

3.2.4 Tariff designer and comparer tests

Name	TDC001. Create a tariff
Module under test	Tariff designer and comparer Resp. ETRA
requiremen	HL-UC 7_PUC_2_Dynamic aggregation of distributed energy assets and active participation into energy market
Test environmen t	WiseCOOP UI up and running
Features to be tested	WiseCOOP facilitates to retailers the ability to define tariffs from the UI
Features not to be tested	
Prenaration	Open WiseCOOP UI Open WiseCOOP operational database
Dependenci	





es			
Steps	1. Define a new rule with the WiseCOOP UI		
	2. Define a new tariff associated to that rule with the WiseCOOP UI		
Pass criteria	The database contains registries with the definition of both items		
Suspension criteria			
	Test successful New rules and tariffs can be successfully created from the UI		
	$\leftarrow \rightarrow \mathbb{C} \ \widehat{\Omega}$ Not secure linuxtest.lab.id:4013/tariffperiods $\frown \bigstar$		
	🕌 WiseCOOP 🚯 Dashboard 🔛 Prosumer detail 🕅 Map 🛛 Tariffs 🔹 Customer profiling 🔹 17: Ancillary Services		
	Tariff periods × New tariff		
Results	Name Test		
	Site		
	Test		
	Spain		
	Cancel Insert		





WiseCOOP ×	
← → C û ③ Not secure linuxtest.lab.id:4013/tariffperiods	o- Q 🏠 🗏
👬 WiseCOOP 🚯 Dashboard 🖾 Prosumer detail 🕮 Map 🛛 Tariffs 🔻 Customer profiling 😤 😂 Ancillary Services Market. 🤅 Web management	* 🗙 Help. *
Tariff period New period	
Price type p1 Application Period Hours © Days Test Range of hours Site Test Prom O Country Sicin Special Dates Saturday © undefined © true © false Daylight Saving Time © undefined © true © false National Holidays © undefined © true © false	
Cancel	

Name	TDC002. Edit ex	istir	ng tariff		
Module under test	Tariff designer a	and	comparer	Resp.	ETRA
Module requirement	HL-UC 7_PUC_2_Dynamic aggregation of distributed energy assets and active participation into energy market				
Test environment	WiseCOOP UI up and running				
Features to be tested	WiseCOOP faci <mark>litate</mark> s to retailers the ability to defin <mark>e ta</mark> riffs from the UI				
Features not to be tested					
Preparation	Open WiseCOOP UI Open WiseCOOP operational database				
Dependencies	TDC001. Create a tariff				
Steps	 Edit a rule with the WiseCOOP UI Edit a tariff associated to that rule with the WiseCOOP UI 				
Pass criteria	The database registries are modified accordingly				
Suspension criteria					





Roculto	Test successful Editing of tariffs is reflected in the database
---------	--

3.2.5 DR framework tests

Name	DRF001. Obtain portfolio demand forecast		
Module under test	Elasticity estimation Resp. HYP		
Module requirement	HL-UC 7_PUC_2_Dynamic aggregation of distributed energy assets and active participation into energy market		
Test environment	WiseCOOP up and running, including DB and forecast orchestrator		
Features to be tested	Retrieval of the aggregated day-ahead demand of the retailer portfolio from the long-term DB of the WiseCOOP tool		
Features not to be tested			
Preparation	Launch WiseCOOP, including its internal ESB, long-term DB and Forecast Orchestrator		
Dependencies	FOR005		
Steps	 Request demand forecast from long-term DB via internal ESB Listen to queue for response message 		
Pass criteria	Reception of portfolio demand forecast		
Suspension criteria			
Results	Test successful		

Name	DRF002. Obtain	generation forecast		
Module under test	Elasticity estima	tion	Resp.	нүр
Module	HL-UC 7_PUC_2	_Dynamic aggregation	of di <mark>stri</mark> buted	d energy assets and
requirement	active participat	<mark>ion i</mark> nto energy market		
Test environment	WiseCOOP up and running, including DB and forecast orchestrator			
leatures to be tested	Retrieval of the day-ahead generation forecast of the retailer portfolio from the long-term DB of the WiseCOOP tool			
Features not to be tested				
Preparation	Launch WiseCOO Orchestrator	OP, including its interna	IESB, long-te	rm DB and Forecast





Dependencies	FOR005
Steps	 Request generation forecast from long-term DB via internal ESB Listen to queue for response message
Pass criteria	Reception of generation forecast message
Suspension criteria	
Results	Test successful

Name	DRF003. Price signal calculation			
Module under test	Price calculator Resp. HYP			
Module requirement	HL-UC 7_PUC_2 <mark>_Dy</mark> namic aggregation of distributed energy assets and active participa <mark>tion</mark> into energy market			
Test environment	WiseCOOP & I <mark>OP up</mark> and running			
Features to be tested	Estimation of the day-ahead 24-hour retail electricity price forecast based on the availability of renewable generation			
Features not to be tested				
Preparation	Launch WiseCOOP & IOP			
Dependencies	DRF001, DRF002			
Steps	 Estimate the imbalance time-series at the level of the retailer portfolio Fix discrete price levels Estimate retail price per hour based on the elasticity of demand and the projected portfolio imbalance at the specific time interval 			
Pass criteria	Generation of retail price time signal with 24 entries. Low prices should correspond to time intervals with high renewable generation and vice versa.			
Suspension criteria				
Results	Test successful			

Name	DRF004. Dispatch price signal			
Module under test	Price calculator	Resp.	НҮР	
Module	HL-UC 7_PUC_2_Dynamic aggregation of distributed energy assets and			





requirement	active participation into energy market			
Test environment	WiseCOOP & IOP up and running			
Features to be tested	Dispatching of the price signal (encoded in the format specified in D10.2) to a queue in the IOP so that other WiseGRID products can receive it.			
Features not to be tested				
Preparation	Launch WiseCOOP & IOP			
Dependencies	DRF003			
Steps	Push the price signal message to the specified queue of the IOP MQTT broker			
Pass criteria	The message containing the price signal is available in the MQTT broker queue and any WiseGRID product listening to the specific queue can receive it.			
Suspension criteria				
Results	Test successful			

Name	DRF005. Elicita <mark>tion o</mark> f demand flexibility per building				
Module under test	Aggregated flex estimation Resp. HYP				
Module requirement	HL-UC 7_PUC_2_Dynamic aggregation of distributed energy assets and active participation into energy market				
Test environment	WiseCOOP, WiseCORP & IOP up and running				
Features to be tested	Elicitation and collection of the demand flexibility potential from the all the buildings that belong to the portfolio of the WiseCOOP operator				
Features not to be tested					
Preparation					
Dependencies					
Steps	 Send message to the target buildings to elicit their demand flexibility potential Listen for the response messages per WiseCORP instance Message decoding and collection of flexibility time-series per building Calculation of aggregated portfolio demand flexibility 				
Pass criteria	Generation of an aggregated demand flexibility potential profile that is in-line with the demand flexibility that can be provided per building				
Suspension criteria					
Results	Test successful				





Name	DRF006. Estimate & dispatch device commands			
Module under test	Optimization & dispatcher Resp. HYP			
Module requirement	HL-UC 7_PUC_2_Dynamic aggregation of distributed energy assets and active participation into energy market			
Test environment	WiseCOOP, WiseCORP & IOP up and running			
Features to be tested	Breakdown of demand flexibility requested from the DSO into the optimal flexibility per building device, based on the information retrieved about the demand flexibility potential.			
Features not to be tested				
Preparation	WiseCOOP up & <mark>run</mark> ning			
Dependencies	DRF005			
Steps	 Collect demand flexibility per building Analyse building-level demand flexibility based on eligibility criteria specified in the flexRequest Calculate which building devices can deliver the target flexibility in the most optimal manner Dispatch commands with demand modification per device 			
Pass criteria	Dispatch of request for demand modification that are according to the flex potential that has been specified by WiseCORP for each asset and which optimises a global objective function.			
Suspension criteria				
Results	Test successful			





4 WISECORP APPLICATION

4.1 IMPLEMENTATION

4.1.1 Architecture overview

The WiseCORP architecture of the application finally implemented does not differ significantly from the architecture presented in the previous deliverable D7.1 WiseCOOP and WiseCORP Apps Design [4]. The architecture is summarized in this section for completeness of this document.

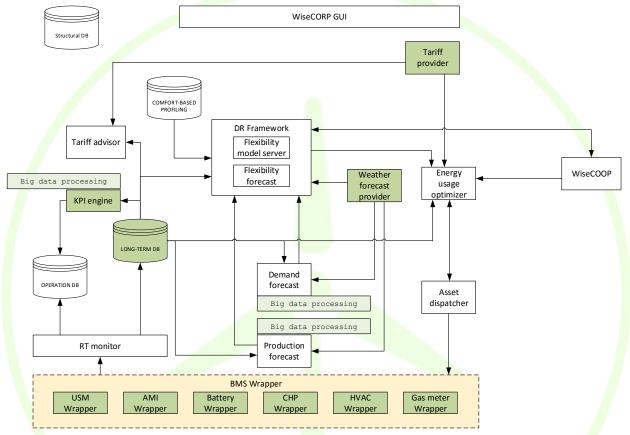


Figure 27 – Overview of interactions among the modules of the WiseCORP application

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. <u>Publication of data from Wrappers to the WiseGRID IOP Message Broker</u>. 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
- 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 submetering), and status from sensors and controllable assets within the building. This subscription is performed by the *RT monitor* module
- 3. <u>Store data for further analysis</u>. 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

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, 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 to 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 significant events occurring in the system (e.g. triggering of explicit demand response campaigns).

4.1.2 Back-office modules

4.1.2.1 Internal Enterprise Service Bus

The description of this module it is also explained in Section 3.1.2.1.

4.1.2.2 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 module has been developed using the .NET Framework 4.5, and gets triggered every minute in order to check that the current set points of the assets (stored in the operational database by the *Real-time monitor*) correspond to the scheduled one (also stored in the operational database by the *Energy usage optimizer module* and the demand response framework). Commands are dispatched using the MQTT-based protocol defined in the D7.1 [4].

Table 21 – Log of command dispatched by Asset dispatcher via MQTT

```
>> [HVAC001/SHIC01/0-1-160-7-0-1] { "_id" : "0-1-160-7-0-1", "assetID" :
"HVAC001", "value" : "26", "unit" : "grdC", "status" : 1, "captureTime"
: ISODate("2018-07-18T14:35:16.379Z"), "description" : "modbus", "mode"
: "cooling", "command" : "auto", "state" : "manual" }
```

4.1.2.3 Real-time monitor

As in WiseCOOP, the *Real Time monitor* is the horizontal module that will handle the data ingestion for most of the applications of the project. It has been designed in order to fulfil the requirements for data ingestion accordingly to the requirements and the architecture of communications proposed for the applications.

Particularly for WiseCORP, this module is in charge of tracking and storing in the databases of WiseCORP the data items shown in the following table.

Data item	Source	Operational DB	Long-term DB
Energy mix	ENTSOE energy mix provider	х	х
Energy mix forecast	ENTSOE energy mix provider	x	×
Energy readings	Field assets (SMX, AMI systems)	х	x
Weather	Weather forecast provider	Х	х
Weather forecast	Weather forecast provider	Х	Х
Asset status (HVAC, batteries, sensors, CHP)	Field assets	x	×

Table 22 – Data items tracked by Real-time monitor in WiseCORP application





\bigcirc	Layers	Schema wisecorp_assetsstatus	
RB	▼ wisecorp	NO DESCRIPTION	
^	wisecorp_things	id [string]_id de la colección	
Home	wisecorp_groupsOfThings	assetID [string] No description	
_	wisecorp_measurements	command [string] No description	
Layers	wisecorp_analysis	description [string] No description	
Layers	wisecorp_tariffRequests	message [string] No description	
<u>-</u>	wisecorp_electricVehicleStatus	mode [string] No description	
ecurity	wisecorp_alerts	state [string] No description	
•	wisecorp_chats	status [number] No description	
Aetrics	wisecorp_weather	topic [string] No description	
Metrics	wisecorp_weatherforecast	unit [string] No description	
	wisecorp_schedule	value [number] No description	
osystem	wisecorp_energyCost		
•	wisecorp_energymix		
S	wisecorp_energymixforecast		
Мар	wisecorp_assets		
-	wisecorp_assetsstatus		
	wisecorp_batteries		
~			
€			
	Help		
		Contact The interoperability platform	
	Development Release notes	About powered by etra	

Figure 28 – Screenshot of the Real-time monitor UI, showing asset status schema

4.1.2.4 KPI engine

The KPI engine of WiseCORP application has been implemented as a set of Spark jobs that are periodically triggered on the long-term database to perform the necessary calculations and push the results back to different collections of the database.

Spark job	Module	Description	Result KPIs				
WiseCORP summaryCalculator 15m	summarycalculator- assembly-0.1.jar	Calculates aggregated registers for every meter and every 15 minutes	Energy demand Energy production Equivalent CO ₂ emissions Associated cost				
WiseCORP summaryCalculator 1h	summarycalculator- assembly-0.1.jar	Calculates aggregated registers for every meter and every hour	Energy demand Energy production Equivalent CO ₂ emissions Associated cost				
WiseCORP summaryCalculator 1d	summarycalculator- assembly-0.1.jar	Calculates aggregated registers for every meter and every day	Energy demand Energy production Equivalent CO ₂ emissions Associated cost				

Table 23 – WiseCORP – Spark jobs of the KPI engine





Spork 2.3.0 Spark Master at spark://wisegridpre.lab.id:7077 URL: spark://wisegridpre.lab.id:7077 REST URL: spark://wisegridpre.lab.id:6066 (cluster mode) Alive Workers: 1 Cores in use: 1 Total, 0 Used Memory in use: 1024.0 MB Total, 0.0 B Used Applicati ons: 0 Runn ng 3 C rivers: 0 Running, 0 Completed Status: ALIVE Workers (1) Worker Id Address State Cores Memory worker-20180718094118-172.17.0.5-8881 172.17.0.5:8881 ALIVE 1 (0 Used) 1024.0 MB (0.0 B Used) Running Applications (0) nory per Executor Application ID Name Cores Submitted Time User State Duration **Completed Applications (3)** Application ID Core Memory per Executor Submitted Time User State Durati app-20180718094617-0002 WiseCORP_summaryCalculation_1d 1024.0 MB 2018/07/18 09:46:17 etraid FINISHED 1.7 min app-20180718094421-0001 1024.0 MB 2018/07/18 09:44:21 FINISHED 1.8 min WiseCORP_summaryCalculation_1h etraid app-20180718094130-0000 WiseCORP summaryCalculation 15m 1024.0 MB 2018/07/18 09:41:30 etraid FINISHED 2.6 min

Figure 29 – Screenshot of Spark server with executed WiseCORP jobs

4.1.2.5 Demand and production forecast service

This module of WiseCORP has been implemented over a RPC server which makes use of the ESB. In addition, this module makes use of the long-term database of WiseCORP, which is implemented over a MongoDB database. The RPC servers (demand and generation forecast) of this module are permanently running to manage the received queries through the RabbitMQ queues enabled to make use of the demand and production forecast.

Within the message queries are specified the ID of the supply point, and the period and the horizon of the desired forecast. In the case of production forecast, in addition of the defined fields, it is specified the type of generation technology.

Once the query is de-serialized and parsed, the forecast module retrieves from the long-term database the necessary information to perform the forecast. To perform the forecast it is retrieved information related to the consumed/produced energy, working calendar and weather information related to the queried installation, being this information available in the long-term database, as it is possible to appreciate in the next picture.





· 📄 🔒 🕨 🔳 🖏				
rb_wisecorp (1) rb_wisecorp	Welcome × I db.getCollection('wisecorp_valu ×			
rb_wisecorp rb_bigdata_wisecorp (1)	📑 rb_bigdata_wisecorp 🗏 etra-id.com:27017 📄 rb_bigd	ata_wisecorp		
rb_bigdata_wisecorp	db.getCollection('wisecorp values 15m').	find({})		
Collections (9)				
> calendar	wisecorp_values_15m 🕔 0.101 sec.			4 0 50 🕨 📴 📝
> updates	Key	Value	Туре	
> wisecorp_energymix	✓	{ 5 fields }	Object	
> wisecorp_energymixforecast	id _id	etra-id.com/WiseCORP/VIcBuildings_2018-06-26 12:30:00	String	
> iiii wisecorp_values_15m	"" id	etra-id.com/WiseCORP/VIcBuildings	String	
> wisecorp_values_1d	👼 timestamp	2018-06-26 10:30:00.000Z	Date	
> wisecorp_values_1h	> 🚥 Measurements	{ 2 fields }	Object	
> wisecorp_weather	👼 prevTimestamp	2018-06-26 10:15:00.000Z	Date	
> wisecorp_weatherforecast	> 🖾 (2) etra-id.com/WiseCORP/VIcBuildings/0001/C5_20	{ 5 fields }	Object	
	> 🖾 (3) etra-id.com/WiseCORP/VIcBuildings/0001/C1_20	{ 5 fields }	Object	
Users	> (4) etra-id.com/WiseCORP/VIcBuildings/0001/C5_20	{ 5 fields }	Object	
	> (5) etra-id.com/WiseCORP/VIcBuildings/0001/C4_20	{ 5 fields }	Object	
	> (6) etra-id.com/WiseCORP/VIcBuildings/0001/C3_20	{ 5 fields }	Object	
	> (7) etra-id.com/WiseCORP/VIcBuildings/0001/C2_20	{ 5 fields }	Object	
	> (8) etra-id.com/WiseCORP/VIcBuildings/0001/C4_20	{ 5 fields }	Object	
	> (9) etra-id.com/WiseCORP/VIcBuildings/0001/C5_20		Object	
	III (10) etra-id.com/WiseCORP/VIcBuildings/0001/PV1		Object	
	III) etra-id.com/WiseCORP/VIcBuildings/0001/C1_2		Object	
	> (12) etra-id.com/WiseCORP/VIcBuildings/0001/PV1		Object	
	III (13) etra-id.com/WiseCORP/VIcBuildings/0001/C2_2		Object	
	> (14) etra-id.com/WiseCORP/VIcBuildings/0001/C1_2		Object	
	> (15) etra-id.com/WiseCORP/VIcBuildings/0001/C2_2		Object	
	> (16) etra-id.com/WiseCORP/VIcBuildings/0001/PV1		Object	
	>		Object	
	>		Object	
	> (19) etra-id.com/WiseCORP/VIcBuildings/0001/C5_2		Object	
	> (20) etra-id.com/WiseCORP/VIcBuildings/0001/C3_2		Object	
	> (21) etra-id.com/WiseCORP/VIcBuildings/0001/C4_2		Object	
	See (22) etra-id.com/WiseCORP/VIcBuildings/0001_2018		Object	
	> (23) etra-id.com/WiseCORP/VIcBuildings/0001_2018		Object	
	See (24) etra-id.com/WiseCORP/VIcBuildings/0001/C4_2		Object	
	20 (25) etra-id.com/WiseCORP/VIcBuildings/0001/C2_2		Object	
	3 (26) etra-id.com/WiseCORP/VIcBuildings/0001/C1_2		Object	
	SOM (27) etra-id.com/WiseCORP/VIcBuildings/0001/C3_2		Object	
	SQ (28) etra-id.com/WiseCORP/VIcBuildings/0001/C5_2		Object	
	 (29) etra-id.com/WiseCORP/VIcBuildings/0001_2018 (30) etra-id.com/WiseCORP/VIcBuildings/0001_2018 		Object	
			Object	
	 Image: Second State Sta		Object Object	

Figure 30 – WiseCORP long-term database screenshot

Once the algorithm is ran, the response is serialized and sent back through the corresponding RabbitMQ queue, providing the queried information. The next message is an example of the response received by the WiseCORP, which is printed in the graphic of the forecast view.

C:\Users\pamuru\Documents\PROYECTOS\WISEGRID\servicio_prediccion\wisecoop\RPCClient_production\bin\Debug\6_RPCCLient.exe	_		\times	
<pre>[x] Requesting: { 'Client_id': 'ZIV0040403483',</pre>			^	
<pre>[.] Got '{"errCode":0, "forecast":{"1532642400":0.838,"1532646000":2.303,"1532649600":1.873,"1532653200 00":2.435,"1532666400":2.724,"1532664000":2.988,"1532667600":2.198,"1532671200":41.166,"1532674800":43. 49.692,"1532682000":45.217,"1532685600":44.814,"1532689200":38.81,"1532692800":36.556,"153269400":37.6 5.07,"1532703600":23.613,"1532707200":25.011,"1532710800":11.902,"1532714400":7.722,"1532718000":6.617, ,"1532725200":5.653,"units":"\W"}</pre>	218,"15 92,"153	3267840 2700000	90": 9":2	
Press [enter] to exit.				

Figure 31 – Screenshot of forecast response message.





4.1.2.6 Comfort-based demand flexibility forecast

The comfort-based demand flexibility forecast component is responsible for estimating the potential demand flexibility of the building, while keeping the building occupants under comfortable indoor conditions. It monitors occupant preferences as well as indoor conditions and continuously estimates the available demand flexibility that could be offered to an aggregator – if asked. The message below is a sample response from WiseCORP to such a request of an aggregator – the WiseCOOP tool in particular. WiseCORP specifies per building (shoID) and per time interval (interval, in multitudes of 15 minutes), the demand flexibility potential. The reference time is the time of reception of the demand by WiseCOOP.

Table 24 – Sample response from WiseCORP to the demand flexibility request from WiseCOOP (as shown in Table

```
18)
RESPONSE
{
    "sholdList": [
        {
             "shoId": "SMX1",
             "flexPotentialList": [
                 {
                     "flexibility": 7.047032,
                     "interval": 1
                 },
                 {
                     "flexibility": 6.575823,
                     "interval": 2
                 },
                 {
                     "flexibility": 14.912035,
                     "interval": 3
                 },
                 {
                     "flexibility": 15.432471,
                     "interval": 4
                 }
             1
        },
        {
             "shold": "SMX2",
             "flexPotentialList": [
                 {
                     "flexibility": 20.054508,
                     "interval": 1
                 },
                 {
                     "flexibility": 16.763426,
                     "interval": 2
                 },
                 {
                     "flexibility": 3.182806,
                     "interval": 3
```





```
{
    "flexibility": 0.9159729,
    "interval": 4
    }
    ]
    }
}
```

4.1.2.7 Energy usage optimizer

The energy usage optimizer is a module developed in Matlab, wrapped in a .NET Framework 4.5 application for facilitating the deployment. This module is triggered once at the end of the day (23:30) in order to calculate the optimum schedule of the controllable assets for the next day. The next 24 hours are divided in equally long slots (24 slots of 1 hour by default), and results are provided as the optimum setpoints for each controllable asset in each slot accordingly to the modelled constraints.

Current model has been developed considering generic characteristics of controllable assets, with the objective of being able to apply it to the majority of cases that will be found in the project. Nevertheless, specific customizations of this module for considering pilot-site related constraints and characteristics is still a possibility likely to be taken in the roadmap of the project.

Input	Rationale
Demand forecast	Forecasted demand for the next day is basic data for performing the evaluation of the optimum schedule of the controllable assets. This data accounts for the demand of the non-controllable assets of the building
Production forecast	Forecasted production for the next day is basic data for performing the evaluation of the optimum schedule of the controllable assets. This data accounts for the production of the non-controllable energy sources of the building
Battery characteristics and status	In those case where the building accounts with batteries in the list of controllable assets, its characteristics (capacity, max. charge power, max. discharge power, state of charge by the end of the day) need to be taken into account
Controllable demand assets characteristics	 Under this term, information about the controllable assets that impose demand to the building is considered (e.g. industrial machines). For each asset, the following information is considered: Power of the asset when operating Time interval when it can operate (e.g. during non-working hours) Minimum and maximum period of time the asset must operate the following day (e.g. between 2 and 3 hours)
Controllable production assets characteristics	 Under this term, information about the controllable production assets is considered (e.g. CHP). For each asset, the following information is considered: Max. power of the asset Associated costs (economic and environmental) bound to the operation of the asset Time interval when it can operate (e.g. during non-working hours)
Energy price	If the module is configured to perform economic optimization, the objective function is defined as the total costs of the energy demanded from the grid, whose calculation is based in the energy price
Energy mix	If the module is configured to perform environmental impact optimization, the objective

Table 25 – WiseCORP – Data inputs of the Energy Usage optimizer





	function is defined as the total equivalent CO_2 emissions of the energy demanded from the grid and self-produced, whose calculation is based in the energy mix of the grid
Contracted capacity	Contracted active power capacity of the building that must not be surpassed by the resulting schedule

Table 26 – WiseCORP – Data outputs of the Energy Usage optimizer

Input	Rationale
Value of objective function	Total foreseen economic cost or CO ₂ emissions of the optimized schedule
Battery schedule	Optimum set of charge/discharge setpoints to be commanded to the battery the following day
Controllable demand schedule	Optimum schedule for the controllable demand assets for the next 24 hours (ON/OFF periods)
Controllable production schedule	Optimum schedule for the controllable production assets (setpoints) for the next 24 hours

4.1.3 User interface

In this part of the document, the main sections and functionalities of the WiseCORP GUI are described, including some screenshots of the actual interfaces.

The implementation of WiseCORP was dperformed in the same way than WiseCOOP (Section 3.1.3)

The web application is protected with a user/password credential system to avoid non-authorized personnel to access sensible information. These credentials are requested before accessing the rest of the application.

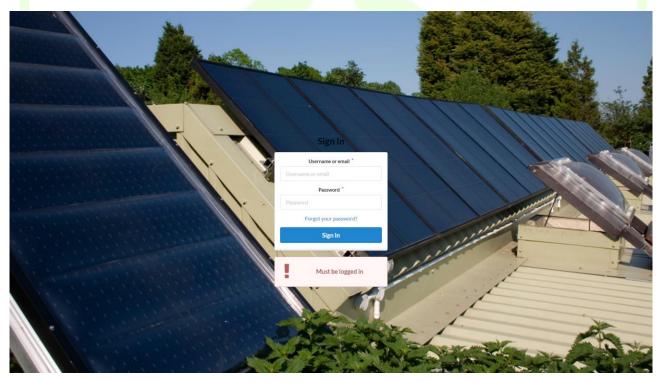


Figure 32 – WiseCORP Login





This credential system also permits the definition of different user profiles to grant or deny access to each section of the application independently. This functionality provides an additional level of privacy, as well as flexibility for the system administrator and the operators that make use of the application.

Once the user has been granted access to the application, diverse functionalities will be available as described in the sections below.

4.1.3.1 Dashboard

The dashboard presents data and indicators per building calculated over the las 30 days, namely:

- Name of the building
- Address
- Self-consumption ratio
- CO₂ emissions
- Economic cost of the energy
- Energy demand
- Energy production

E q	WiseCORP	Buildings 🔻	🙆 Мар	Monitoring	Optimization 🝷	Settings 🔻		۰	Hello rb 👻
			6	Dashboar	ď				
			ETF	RA (ETRA bu	ilding)				
			c/T	res Forques	, 147. 46014	València	(València)		
				Mar Par	Values recor	ded over the last	30 days		
					Self-cor 5.397 %	sumption	CO₂ Emissions Energy Cost 3.2 t 1.588.17 €		
			<		Energy 13.24 MW	C <mark>onsumptio</mark> r ^{/h}	Energy Production 714.66 kWh		
						igure 3	3 – WiseCORP UI – Dashboard		
			1.2	a	line dete				

4.1.3.2 Building details

This section provides an insight and detailed indicators for each one of the buildings managed by WiseCORP. Particularly, when a building is selected, for each month the following information is provided:

- Self-consumption ratio
- CO₂ emissions
- Economic cost of the energy
- Energy demand
- Energy production
- Chart with daily demand
- Chart with daily production
- Energy consumption share per tariff period





🛐 WiseCORP Buildings - 🙆 Map 🔛	d Monitoring Optimization * Settings *			<table-cell> Hellorb 👻</table-cell>
	Buildings details			
└ ₽ València Buildings └ 및 ETRA	ETRA (ETRA building)			
	Overview Demand distribution			
	June * 2018	Refresh		
	Self-consumption	CO ₂ Emissions	Energy Cost	
	4,644 %	5.57 t	2,362€	
	Energy Consumption 23.98 MWh	Energy Production		
	Energy Consumption			
	1,500k		La se	
	1,000k	110.		
	§ 750k			
	500k		-	
	250k			
	Ok 1. jun 3. jun 5. jun 7. jun 5). Jun 11. Jun 13. Jun 15. Jun 17. Jun 19. Jun	21. Jun 23. Jun 25. Jun 27. Jun 29. Jun	
	5. 24 147 6			
	Figure 34 – WiseC	ORP UI – Building ener	rgy usage details (i)	
	23.98 MWh	111 MWh		
	Energy Consumption			
	1,500k		28/5/2018 1,289,970 kWh	
	1,250k		<u> </u>	
	1,000k			
	500k			
	250k —			
	Ok 1. jun 3. jun 5. jun 7. jun 5	, jun 11. jun 13. jun 15. jun 17. jun 19. jun	21. jun 23. jun 25. jun 27. jun 29. jun	
	Energy consumption by tariff period			
	p3: 2	5.79 % p1: 2	27.33 %	
		p2: 46.88 %		

Figure 35 - WiseCORP UI - Building energy usage details (ii)

A second tab of the same section displays a calendar view of the hourly demand and production of the building over the selected week. A colour gradient scale is used to make it possible to easily identify the periods where demand or production is concentrated.





	🗒 Bu	ildings detail	s						
ia Buildings RA	ETRA (ETRA buildin	g)							
	Overview Demand distri	oution							
	Filter								
	2018 -	26 (25/6/2018 - 1/7/	2018l • Refresh						
	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Scale (kWh)	
	00:00 - 00:59							104	
	01:00 - 01:59							98	
	02:00 - 02:59							94	
	03:00 - 03:59							90	
	04:00 - 04:59 05:00 - 05:59							85	
	06:00 - 06:59							77	
	07:00 - 07:59							73	
	08:00 - 08:59							68	
	09:00 - 09:59							64	
	10:00 - 10:59							60	
	11:00 - 11:59							56	
	12:00 - 12:59							51	
	13:00 - 13:59							47	
	14:00 - 14:59							43	
	15:00 - 15:59							38	
	16:00 - 16:59							34	
	17:00 - 17:59							30	
	18:00 - 18:59							26	
	19:00 - 19:59							21	
	20:00 - 20:59							17	
	21:00 - 21:59							13	
	22:00 - 22:59							9	
	23:00 - 23:59							4	

Figure 36 – WiseCORP UI – Building energy usage details (iii)

Additionally, a section about automation is also included in the User Interface. This section provides an overview of the schedule planned and executed for each one of the controllable assets of the building (HVAC, batteries, controllable loads, CHP, etc.). The information displayed is actually the output of the Energy Usage Optimizer – the module that calculates the optimum schedule day-ahead – and the demand-response framework – which sets the necessary changes to the original schedule in order to accomplish the requirements of participation in demand-response campaigns.

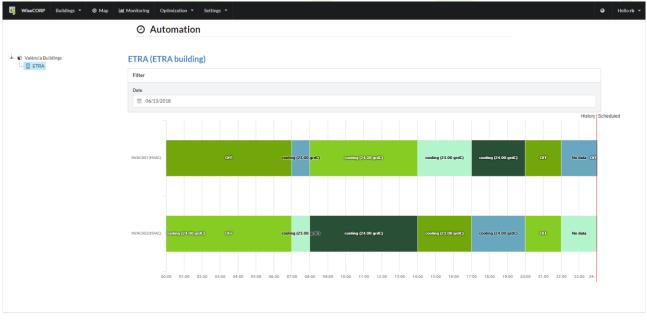


Figure 37 – WiseCORP UI – Building automation details





4.1.3.3 Map

Under this section, a geographical representation of the managed buildings is presented. The map displays the location of the different buildings. By clicking in each one of them, the different sensors measuring production and demand in the building are presented, together with a short summary of the last set of values produced by those.

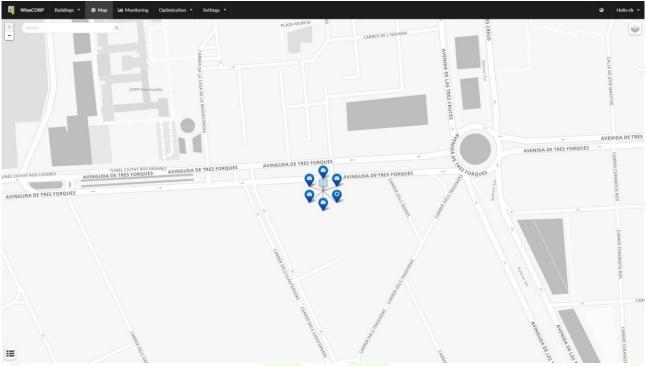


Figure 38 – WiseCORP UI – Representation of the buildings in a map

By clicking in each one of the sensors, the site gives access to the representation of all data for the last 24 hours and 30 days.

4.1.3.4 Monitoring

The monitoring section allows the Facility Manager to analyse in detail the different metrics retrieved from the building. Upon selection of a particular asset, data range, set of metrics (active power, energy demand, voltage...) and integration period (15 minutes, 1 hour, 1 day), a chart is displayed showing the required data.

The selection form is versatile enough to allow the Facility Manager to print simultaneously and compare data from different metrics or sensors.





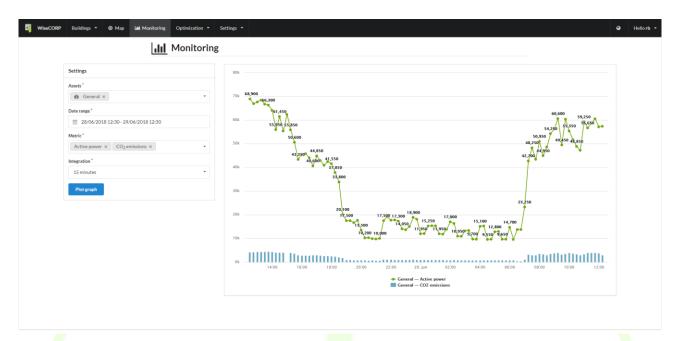


Figure 39 – WiseCORP UI – Monitoring

4.1.3.5 Tariff optimization

The tariff optimization section allows the Facility Manager to use the historical energy demand and production data retrieved from the sensors in the different building in energy cost simulations. Facility Manager can trigger a new simulation by selecting the building or set of sensors whose data wants to analyse, and a complete month.

🔄 WiseCORP Buildings 👻 @ Map			
Tarif Filter per label	f optimization	New optimi	
Optimization se	New tariff optimization	vings (€)	
	Select entities	Select month ergy	
test May 2018 6 entities involve	 ✓ ♥ València Buildings ✓ ♥ □ ETRA 	Month* June 2018 -	
Request date: 20	₩ 46 CP1 ₩ 46 FM1	Settings	
Status: Complete Green Fixed pri	😴 🍪 General 😴 🍪 HVAC	Tariff Green Fixed prices Discount	
May 2018	👻 🍪 LIGHTING	Priority Consumption cost Global cost	
May 2018 6 entities involve		Optimization label -6119,	
Request date: 0 Status: Completi Green Ried pr		Label * June	
Show 5			
		Send optimization request	



Once the simulation is triggered, WiseCORP automatically computes the applicable costs that would result for each one of the tariffs defined within the WiseCOOP application (whose definition is shared in the IOP via





the tariff provider module). When the simulation is finishes, a comprehensive comparison of the results is provided to the Facility Manager, making it very easy to identify which tariffs would benefit or would increase the costs associated to the energy demand.

Filter per label								New optin	nization
Optimization settings	Measured value	Recommended energy plans		C	ost (€)		Savings (€)	
Optimization settings	Measured value	5	Energy plan name	Company	G	lobal E	Energy	Energy	
test Z May 2018	Peak active	315.75	1 Plan Batería de condensado	res Negocios	Iberdrola	15045.7	5 10823	3.4 🕥	-6071.74
6 entities involved Request date: 20/06/2018 17:13	Consumption (kWh)	76699	2 Ahorro Pro Luz 3.0		Aldro Energía	11846.3	8 10870	0.7 🛇	6119.03
Status: Completed Green Fixed prices Discount	Energy cost (€)	4751.66	3 Descuento fijo >15kW 3.1		FactorEnergía	11933.0	4 11116	5.63 📀	6364.97
May 2018 2 May 2018 2	Peak active	315.75	Descuento fijo >15kW 3.0	FactorEnerg	gía 1	1515	11847.16	-709	5.50
6 entities involved Request date: 08/06/2018 12:21	Commention	76699	2 Ahorro Pro Luz 3.0	Aldro Energ	ía 1	1846.38	10870.7	o -611	9.03
Status: Completed Green Fixed prices Discount	Energy cost (€)	4751.66	3 Descuento fijo >15kW 3.1	FactorEnerg	țía 1	1933.04	11116.63	o -636	4.97
Show 5 rows per page	•						Page	1	of 1

Figure 41 – WiseCORP UI – Tariff optimization, results

In addition, the simulation results include the detail of how the energy bill would look like with each one of the tariffs used in the simulation. These can be used to check the details during the comparison of different tariffs, as relevant information during negotiation of better tariffs with the retailer, or as an assessment of the correctness of the energy bills delivered by the retailer.

 Tari 	 May 20 	10						
Filler yes line	Site		Sort solutions by	Solution				ew optimization
	Valencia	*	Global cost •	11515.00 € > D	escuento fijo	>15kW 3.0 (FactorEn	ergía) 🔹	
Optimization s								iavings (C)
	Month Days	Entities	Peak active power (kW)	at	Total co	onsumption (kWh)	Total cost (€)	inergy
test May 2018	May 2018 31	5	315.75	23/05/2018 11:45	76699		4751.66	○ -6071.74
6 entities invol-	Plan name		Company	Tariff type	Green	Fixed prices	Discount	Q −6119.03
Status: Comple Grine Fields	Descuento fijo >15kW	000		100				S (0) 6364.97
	Descuento njo >15kvv	3.0 🖻	FactorEnergía	3.0A	Yes	Yes	Yes	
May 2018 May 2018	Concept		Price	Value			Total	•7095.50
6 entities invol- Request date 1	Contracted power (P1)		0.11498097 €/kW/day	315 kW			1122.79€	0 6112.03
Status Comple Greet Rolds	Contracted power (P2)		0.069551537 €/kW/day	286 kW			616.64€	 (6364.97
	Contracted power (P3)		0.049072373 €/kW/day	185 kW			281.43€	
Show 5					То	tal contracted power	2020.86€	1 of 1
	Consumed energy (P1)		0.146595 €/kWh	20717 kWh			3037.01€	
	Consumed energy (P2)		0.124274 €/kWh	39179 kWh			4868.93€	
	Consumed energy (P3)		0.091295 €/kWh	16803 kWh			1534.03€	
					To	tal consumed energy	9439.97€	
	Energy consumption di	scount	25.5 %	9439.97€			-2407.19€	
						Total discounts	-2407.19€	

Figure 42 – WiseCORP UI - Tariff optimization, simulated bills





4.1.3.6 Demand response

The Demand response section of the UI provides details on the participation of the facilities in the different demand-response modes enabled by the WiseGRID project. Therefore, upon selection of one building information is presented in two separated tabs.

The first tab is named "Implicit -Dynamic price", and gives details of the participation of the facility assets in implicit demand-response campaigns, which in the framework of the project have been implemented by means of dynamic tariffs. As explained in detail in D10.2 [6], retailers/aggregators can produce dynamic tariffs for their portfolio of users by taking advantage of the Implicit DR module of the WiseCOOP application. These dynamic tariffs get afterwards communicated to other applications of the WiseGRID ecosystem, enabling those to use the energy price in their specific optimization logic. In the context of WiseCORP, the Energy Usage Optimizer module is in fact capable of using the prices defined in this dynamic tariff to calculate day-ahead the optimum schedule for the different controllable assets, which will minimize the overall energy costs for the facility manager. In consequence, this section of the UI shows the relevant information around this use case: the current day energy price and the optimum schedule for the assets.

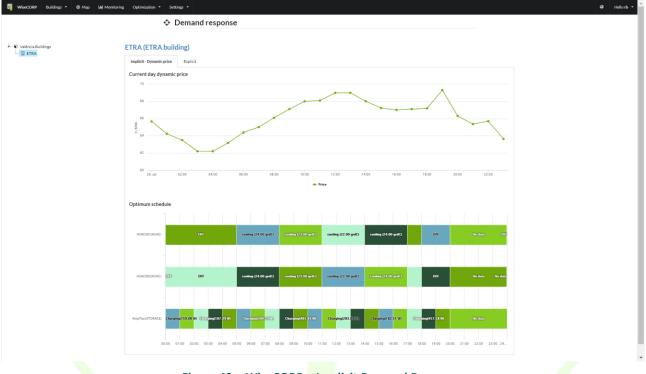


Figure 43 – WiseCORP – Implicit Demand Response

The second tab focuses on the second type of demand-response mechanism enabled within the WiseGRID project, explicit demand response. In this context, the aggregator, using WiseCOOP, is authorized to explicitly request a certain amount of flexibility from the controllable assets of the facility. The Demand-Response framework is in position of calculating how the schedule of the assets shall be changed in order to meet with the new constraints imposed by the participation in the explicit demand-response campaign. This section of the UI therefore focuses on displaying the relevant changes produced in the schedule of the assets as a result of the execution of those campaigns.





	Demand resp	onse										
València Buildings	ETRA (ETRA buildin	ig)										
EIKA	Implicit - Dynamic price	Explicit										
	Reschedules due to expli	icit DR										
	HVAC001(HVAC)		OFF		cooling (2	3.00 grdC)		cooling (24.0)0 grdC)	OFF	No data	OFF
	HVAC001(HVAC) – Reschedule	-										
	HVAC002(HVAC)	OFF	OFF		cooling (2	3.00 grdC)		cooling (24.0	00 grdC)	OFF	No data	No data
	HVAC002(HVAC) - Reschedule										cooling (22.	.00 grdC)
	AmpTest(STORAGE)	Charging(1	59.08 W)	Charging(219.51 W)		Cha	rging(283,84 W)	Charg	t <mark>ing(170</mark> .92 W)		No data	
	AmpTest(STORAGE) - Reschedule					D	ischarging(200.0	ow)				
		0.00	02:00	04:00 06:00	08:00				5.00 18			24.00

Figure 44 – WiseCORP – Explicit Demand Response

4.2 LAB-TESTING RESULTS

This section contains a set of templates with the definition, objectives, steps and results of all tests executed during this period on the different modules of the tool.

4.2.1 KI	monitor tests			
Name	RTM001. Read smart meter data f	rom IOP		
Module under test	RT Monitor		Resp.	ETRA
Module requirement	HL-UC 7_PUC_1_Dynamic manage	ement of dem	nand side assets in	tertiary sector
Test environment	SMX running and sending informa IOP platform up and running	tion to IOP		
Features to be tested	Data from SMX is properly collect	ed in the ope	rational database	of WiseCORP
Features not to be tested				
Preparation	Configure one SMX to send data <mark>t</mark>	<mark>o the</mark> lab-test	ting IOP environme	ent
Dependencies				
Steps	The execution of this test must ha	ppen automa	atically upon publi	cation of data in the IOP
Pass criteria	Data from the SMX is correctly upc keeps a register of the last values			e. Operational database
Suspension criteria				

4.2.1 RT monitor tests





	testing environm	reenshot shows how the operational Di nent SMX	B is populated with da	
	Robo 3T - 1.2 File View Options Window H P	eip	instead Operational Scholaus (#	
	■ wisegridpre (13) ▷ System ▷ WiseCORP_EMS ■ WiseCORP_Server ■ WiseCORP_Server	* db.getCollection("Mess X wisegndpre wisegn	-	p:-1}).limit(100 50 🕨 📴 🖻 🖵
Results	 Daylight KpiCalculations KpiCalculations Gasurements Optimizations TariffData Functions Functions Config th_bigdata_wisecop th_wisecop th_wisecop th_wisecop th_wisevp 	Key Image: Strate in the image	<pre>(9 fields) etra-id.com/WiseCORP-VIcBuildings-0001-C1_Acc etra-id.com/WiseCORP-VIcBuildings/0001/C1 ActivePower 2018-07-19 14:30:00.000Z 900 Minute15 0.0 null true (9 fields) </pre>	Type Object Object Object String String Date Int32 String Double Null Boolean Object Object

Name	RTM002. Read sensor data from IOP			
Module under test	RT Monitor		Resp.	ETRA
Module requirement	HL-UC 7_PUC_1_Dynamic managemen	t of dem	nand side assets in	tertiary sector
Test environment	Sensor wrapper simulator IOP platform up and running			
Features to be tested	Data from sensor Wrapper is prop <mark>erly</mark> o	collected	d in the operationa	l database of WiseCORP
Features not to be tested				
Preparation	Configure one sensor wrapper simulate	or to ser	nd data to the lab-t	esting IOP environment
Dependencies				
Steps	The execution of this test must happer	automa	atically upon public	cation of data in the IOP
Pass criteria	Data from the sensor is correctly up database keeps a register of the last va		•	database. Operational
Suspension criteria				
Results	Test successful Sensor data (temperature, lighting and	l smart	olug) sent by the s	imulators is successfully





File View Options Window Help • D. bjdjdata wysockpit • D. bjdjdata wisecorp • D. wisecorp.assetstatus • O.049 sec • Value • Type • D. wisecorp.assetstatus • O.049 sec • Value • Type • O.050 • O.0001 (8 freids) Object UUX001 (8 freids) Object UUX001 Sting • Contentions • Contentions • Contentions • D. mesoages • Status 1 • Discounts JoginServiceConfigur. • Status 1 • Discounts JoginServiceConfigur. • Wisecorp.assetstatus 1 • Status	Robo 3T - 1.2	nn Conversion To	ale		
Image: bigdata_wisecop Image: bigdata_wisecop Image: bigdata_wise	File View Options Window Help				
> in b_bigdata_wisecorp > in b_bigdata_wisecorp > in b_bigdata_wisecorp > in b_bigdata_wisecorp > in b_wisecorp. > in b_wisecorp.asets > wisecorp.asets > wisecorp.asetstatus </th <th>🛃 🗕 🎬 🔚 U 🕨 📑 🔅 ings 🔹 📀 Map</th> <th></th> <th></th> <th></th> <th></th>	🛃 🗕 🎬 🔚 U 🕨 📑 🔅 ings 🔹 📀 Map				
> the_bigdata_wisecoop > the_bigdata_wisecoop > the_bigdata_wisecoop > the_bigdata_wisecoop > the_bigdata_wisecoop > the_wisecoop > the t	Image: bigdata_wgcockpit	A db aetCollection//wisecor _X			
> B. douglata, wisecorp > douglata, wisecorp > Collections (28) > douglata, wisecorp > goother > douglata, wisecorp > douglata, wisecorp > douglata, wisecorp > douglata, wisecorp > douglata, wisecorp > goother > dougl	b 📴 rb_bigdata_wisecoop	Deve eve el ve eve eve	-		
> ■ in building, Wiseopp ● in building ●	Image: Book of the second s				
> in dwiscoop_hitings Key (CTRAbuilding) Value Type > in dwiscoop_hitings jd UX001 (a fields) Object >	b B rb_bigdata_wiseevp	db.getCollection('wisecorp_as	ssetsstatus').find({})		
Image: Description of the public of the p	b B rb_wgcockpit	wisecorp_assetsstatus 🕓 0.004 sec.		۹ 0	50 🕨
Image: Provide constraints Image: Provide constraints <t< td=""><td>→ B rb_wisecoop</td><td>Key (ETPA building)</td><td>Value</td><td></td><td></td></t<>	→ B rb_wisecoop	Key (ETPA building)	Value		
↓ Collections (28) assett/or is astat/or is assett/or is assett/or is assett/or is ass	▲				
▶ _activity Image: Section of the sectin of the sectin of the sectin of the section of the sec		id id			
>		Implicit - Dynamic price Explicit			
▶connections ↓connections ↓connections ↓connections ↓connections ↓connections connections ↓source ↓source ↓source ↓connections ↓source ↓source ↓source ↓source ↓source ↓layers ↓source ↓uit ↓source ↓uit ↓source ↓ooder ↓					
▶ _cronHistory ■ status 1 Int32 ▶ _events UXX001/SHIC01/0-1-151-7-0-1 String String ▶ _intexes 1 Int32 ▶ _messages Interversion 1 Int32 ▶ _messages Interversion 1 Int32 ▶ _messages Int32 Int32 Int32 > _messages Int32 Int32 Int32 > _messages _messages Object String Int32 > _messages _messages _messages Object String Int32 > _messages _messages _messages _messages Int32 Int32 Int32 > _messages _messages _messages _messages Int32 Int32 Int32 Int32 Int32 Int32 I			Ambient light		
	cronHistory				
Indexes Indexes Indexes Indexes Indexes Indexes Index Ind		= topic	LUX001/SHIC01/0-1-151-7-0-1	String	
> messages > 000 intl2 > messages intl2 intl2 > schema > 000 intl2 > schema > 000 000 > meteor_accounts_loginServiceConfigur SetUlo01 String > meteor_accounts_loginServiceConfigur Capture Time 2018-07-26 134254327Z Date > wisecorp_analysis is status 1 Intl2 > wisecorp_assets is topic SPLUG001/SHIC01/0-1-165-7-0-1 String > wisecorp_assets intl2 Wisecorp_assets VCO SE unit Terreleview > wisecorp_assets VCO SE unit Terreleview V String > wisecorp_assets int12 Array Int12		unit/AC001(HVAC)	OFlux	ooling (23.0String)	
>		· value	500	Int32	
b cronHistory String b meteor_accounts_loginServiceConfigur sasetID SPLUGO01 String b users capture Time 2018-07-26 13:42:54.927Z Date b wisecorp_alerts capture Time 2018-07-26 13:42:54.927Z Date b wisecorp_alerts capture Time 2018-07-26 13:42:54.927Z Date b wisecorp_analysis capture Time 2018-07-26 13:42:54.927Z Date b wisecorp_assets 1 Int32 b wisecorp_assets SPLUGO01/SHIC01/0-1-165-70-01 String b wisecorp_assets W String b wisecorp_batteries V String		4 😡 (3) SPLUG001	{ 8 fields }	Object	
> meteor_accounts_loginServiceConfigur > ServiceConfigur > 2018-07-26 1342:54.9272 Date > users 0 0************************************		HVAC00m (idC) - Reschedule	SPLUG001	String	
▷ users □ description VCI OF O		assetID	SPLUG001	String	
b wisecorp_alerts a astropy athus b wisecorp_analysis iiii int32 b wisecorp_analysis iiiii int32 b wisecorp_assets wisecorp_assets wisecorp_assets b wisecorp_assets wisecorp_assets W b wisecorp_assets value [2 elements] c (4) TEMPO01 (8 fields) Object		🗊 captureTime	2018-07-26 13:42:54.927Z	Date	
b wisecorp_analysis imits/secorp_assets structure		description (HVAC) OFF	OFsmartplug	cooling (23. GrandC)	
b wisecorp_assets VACV @ unit Bit Schedule W String b wisecorp_assetsstatus b walue [2 elements] Array b wisecorp_batteries 4 @ (4) TEMP001 [8 fields] Object			1	Int32	
b wisecorp_assetsstatus b walue [2 elements] Array b wisecorp_batteries 4 (a) (EMP001 (8 fields) Object			SPLUG001/SHIC01/0-1-165-7-0-1	String	
Wisecorp_batteries Autory Autor		HVAC00 Init - Reschedule	W	String	
a contraction (4) TEMPOUL (8 Helds)		value	[2 elements]	Array	
Wisecorp_chats Wisecor					
Logs		Charging(15	9.08 TEMPONI Charging(219.51 W)	String Charging(283:84	

Name	RTM003. Read battery data from I	ОР		
Module under test	RT Monitor		Resp.	ETRA
Module requirement	HL-UC 7_PUC_1_Dynamic manage	ment of dem	and side assets in	tertiary sector
	Battery running <mark>and send</mark> ing infor IOP platform up and running	mation to IOF		
Features to be tested	Data from battery is properly colle	cted in the o	perational databas	se of WiseCORP
Features not to be tested				
Preparation	Configure one battery to send dat	<mark>a to</mark> the lab-t	esting IOP environ	ment
Dependencies				
Steps	The execution of this test must ha	<mark>ppe</mark> n automa	atically upon public	cation of data in the IOP
Pass criteria	Data from the battery is correct database keeps a register of the la			database. Operational
Suspension criteria				
Results	Test successful. The following screenshot shows he testing environment battery from	•	itional DB is popula	ated with data from lab-





F	ile View Options Window Help			
	₹ - 📄 🔒 🕨 💷 沈			
	A L Collections (27)	db.getCollection('wisecor ×		
	activity			
	▷cluster	wisegridpre 🗏 wisegridpre.lab.id:27017		
	connections	db.getCollection('wisecorp_	<pre>batteries').find({})</pre>	
	▷cronHistory	wisecorp_batteries 🕔 0.002 sec.		4 0 50 🕨 🔞 📰
	events	Key	Value	Type
	Iayers	▲ (1) AMPTest	{ 22 fields }	Object
	messages	id	AMPTest	String
	▷ 🛄 _schema	Alarms	0	Int32
	cronHistory	BatterySOC	7.8	Double
	meteor_accounts_loginSe	BatterySOH	93.2	Double
	users	BatteryVoltage	48.5	Double
	wisecorp_alerts	ChargeAvailable	2425	Int32
	wisecorp_analysis	Demand	116.2	Double
	wisecorp_assets	DischargeAvailable	0	Int32
	wisecorp_assetsstatus	ExternalPVPower	Ő	Int32
	wisecorp_batteries	InverterActivePower	Ő	Int32
	wisecorp_chats	InverterBatteryPower	ů	Int32
	wisecorp_energyCost	InverterPVPower	ů 0	Int32
	wisecorp_energymix	InverterPVVoltage	312	Int32
	wisecorp_energymixforec	InverterReactivePower	0	Int32
	wisecorp_groupsOfThings	MeterActivePower	116.2	Double
	wisecorp_measurements	MeterGridFrequency	50	Int32
	wisecorp_schedule	MeterGridVoltage	227	Int32
	wisecorp_tariffRequests	MeterGravortage MeterReactivePower	-1611.1	Double
	wisecorp_things	Status	4	Int32
	wisecorp_weather	 Image: Status Image: S	3 elements 1	Array
	wisecorp_weatherforecast	WorkingMode	[5 elements]	Int32

Name	RTM004. Read HVAC data from IOP			
Module under test	RT Monitor		Resp.	ETRA
Module requirement	HL-UC 7_PUC_1_Dynamic manager	nent of dem	hand side assets in	tertiary sector
Test environment	HVAC Wrapp <mark>er s</mark> imulator IOP platform up and running			
Features to be tested	Data from HVAC is properly collecte	ed in the op	erational database	e of WiseCORP
Features not to be tested				
Preparation	Configure one HVAC simulator to se	end data to	the lab-testing IOF	environ <mark>men</mark> t
Dependencies				
Steps	The execution of this test must hap	<mark>pen</mark> automa	atically upon public	cation of data in the IOP
Pass criteria	Data from the HVAC is correctly database keeps a register of the las			database. Operational
Suspension criteria				
Results	Test successful. The following screenshot shows ho testing environment wrapper simul		ational DB is popula	ated with data from lab-





 Robo 3T - 1.2						
View Options Window Help						
- 📄 🖬 🕨 🔳 🗱						
A L Collections (27)	db.getCollection('wisecor ×					
▷						
V mCruster						
connections						
cronHistory	wisecorp_assetsstatus 🕔 0.002 sec.			4 0 50 🕨 🛅 🖬 🖻 🗖		
_events	Кеу	Value	Туре			
Iayers	4 💷 (1) HVAC001	{ 11 fields }	Object			
 _messages _schema 	_id	HVAC001	String			
scnema	📟 assetID	HVAC001	String			
ronHistory meteor_accounts_loginSe	🚥 command	auto	String			
users	description	modbus	String			
wisecorp_alerts	🚥 mode	cooling	String			
wisecorp_analysis	😁 state	manual	String			
wisecorp_assets	status	1	Int32			
wisecorp_assetsstatus	== topic	HVAC001/SHIC01/0-1-160-7-0-1	String			
wisecorp_batteries	📼 unit	grdC	String			
▷ wisecorp_chats	a value	23	Int32			
wisecorp_energyCost	··· message	{ "_id" : "0-1-160-7-0-1", "assetID" : "HVAC				
wisecorp_energymix	▷ 🖼 (2) HVAC002	{ 11 fields }	Object			
wisecorp_energymixforec						
wisecorp_groupsOfThings						
wisecorp_measurements						
wisecorp_schedule						
wisecorp_tariffRequests						
wisecorp_things	1					
wisecorp_weather						
wisecorp_weatherforecast						
N III Functions						
ogs						

Name	RTM005. Store smart meter data to Long-term DB
Module under test	RT Monitor ETRA
Module requirement	HL-UC 7_PUC_1_Dynamic management of demand side assets in tertiary sector
Test environment	SMX running and sending information to IOP IOP platform up and running
Features to be tested	Data from SMX is properly collected in the long-term database of WiseCORP (big data)
Features not to be tested	
Preparation	
Dependencies	RTM001. Read smart meter data from IOP
Steps	The execution of this test must h <mark>appe</mark> n automatically upon publication of data in the IOP
Pass criteria	Data from the SMX is correctly appended to the historic registry held in the long-term database
Suspension criteria	
Results	Test successful. The following screenshot shows how the operational DB is populated with data from lab- testing environment SMX





wisegridpre (13) System	• * db.getCollection('Meas ×		
▷ System	🚽 wisegridpre 🗏 wisegridpre.lab.id:27017 🗦 WiseCORP_Server		
▲	<pre>wasurements').find({thingId:'etra-id.com/WiseCORP/VlcButerents')</pre>	<pre>uildings/0001/C1'}).sort({timestamp</pre>	:-1}).limit
 Collections (5) 	Measurements 🕔 6.49 sec.	4 0 5	0 🕨 🖪 📝
Daylight	Key	Value	Туре
KpiCalculations	(1) etra-id-com-WiseCORP-VIcBuildings-0001-C1_AcFrequency_20180719	(9 fields)	Object
Measurements	▲ ⁽²⁾ (2) etra-id-com-WiseCORP-VIcBuildings-0001-C1_ActivePower_20180719		Object
Description	id jd	etra-id-com-WiseCORP-VIcBuildings-0001-C1_Ac	
TariffData	· thingId	etra-id.com/WiseCORP/VIcBuildings/0001/C1	String
Functions	· metricType	ActivePower	String
Disers	👼 timestamp	2018-07-19 14:30:00.000Z	Date
▷	interval	900	Int32
B rb_bigdata_wgcockpit	intervalString	Minute15	String
▷	an value	0.0	Double
B rb_bigdata_wisecorp	error error	null	Null
▷	isValid	true	Boolean
▷	4 (3) etra-id-com-WiseCORP-VIcBuildings-0001-C1_Co2Emissions_2018071	{ 9 fields }	Object
▷ 🗟 rb_wisecoop	4) 4) etra-id-com-WiseCORP-VIcBuildings-0001-C1_Current_2018071914300.	. { 9 fields }	Object
▷	(5) etra-id-com-WiseCORP-VIcBuildings-0001-C1_EnergyConsumption_20		Object
▷	6) etra-id-com-WiseCORP-VIcBuildings-0001-C1_EnergyCost_201807191		Object
	Image: Provide the second s		Object
	(8) etra-id-com-WiseCORP-VIcBuildings-0001-C1_PhaseActivePowerB_20		Object
	(9) etra-id-com-WiseCORP-VIcBuildings-0001-C1_PhaseActivePowerC_20		Object
	(10) etra-id-com-WiseCORP-VIcBuildings-0001-C1_PhaseCurrentA_201807		Object
	III) etra-id-com-WiseCORP-VIcBuildings-0001-C1_PhaseCurrentB_201807		Object
	Image:		Object
	Image: Provide the second s	{ 9 fields }	Object

Name	RTM006. Store sensor data to Lo <mark>ng-ter</mark> m DB
Module under test	RT Monitor Resp. ETRA
Module requirement	HL-UC 7_PUC_1_Dynamic management of demand side assets in tertiary sector
Test environment	SMX running and sending information to IOP IOP platform up and running
Features to be tested	Data from sensor is properly colle <mark>cted</mark> in the long-term database of WiseCORP (big data)
Features not to be tested	
Preparation	
Dependencies	RTM002. Read sensor data from IOP
Steps	The execution of this test must happen automatically upon publication of data in the IOP
Pass criteria	Data from the sensor is correctly appended to the historic registry held in the long-term database
Suspension criteria	
Results	Test successful History of changes of sensor data gets successfully stored to the long-term DB





▲	# db.getCollection('wisec)	c		
Collections (28)	🚽 wisegridpre 🗏 wisegridp	re.lab.id:27017 📄 rb_bigdata_wisecorp		
activity cluster	db.getCollection('	wisecorp_assetsstatus').find({id:'Lt	JX001'}).sor	t({updatedTime:
	wisecorp assetsstatus	0.000	√ 0	50 🕨 🛅 📝 🗎
			•	50 🕨 🛅 🕎 📃
events	Кеу	Value	Туре	
≥	4 💷 (1) ObjectId("5b59ec		Object	
iayers		ObjectId("5b59ed85ba72460006717b3	. ObjectId	
>schema	觉 captureTime	2018-07-26 15:49:25.655Z	Date	
CronHistory	status	1	Int32	
meteor_accounts_loginServiceConfigur =	= id	LUX001	String	
Intereor_accounts_roginserviceconingur Image of the second	operation	change	String	
wisecorp alerts	🗊 updatedTime	2018-07-26 15:49:25.722Z	Date	
wisecorp_analysis	4 💷 (2) ObjectId("5b59ec		Object	
wisecorp_analysis	id	ObjectId("5b59ed49ba72460006717b3	. ObjectId	
wisecorp_assets	🗊 captureTime	2018-07-26 15:48:25.655Z	Date	
wisecorp_assetsstatus		1	Int32	
Wisecorp_batteries wisecorp chats	== id	LUX001	String	
	operation	change	String	
wisecorp_energyCost	ö updatedTime	2018-07-26 15:48:25.703Z	Date	
wisecorp_energymix	4 💷 (3) ObjectId("5b59ec	l0dba7246 { 6 fields }	Object	
wisecorp_energymixforecast wisecorp_groupsOfThings	id	ObjectId("5b59ed0dba72460006717b3	. ObjectId	
	👼 captureTime	2018-07-26 15:47:25.655Z	Date	
wisecorp_measurements	status	1	Int32	

Name	RTM007. Store battery data to lo	ng-term DB		
Module under test	RT Monitor		Resp.	ETRA
Module requirement	HL-UC 7_PUC_1_Dynamic manag	ement of dem	nand side assets in	tertiary sector
Test environment	Battery running and sending infor IOP platform up and running	rmation to IOF	þ	
Features to be tested	Data from battery is properly coll	ected in the lo	ong-term database	e of WiseCORP (big data)
Features not to be tested				
Preparation				
Dependencies	RTM004. Read battery data from	IOP		
Steps	The execution of this test must ha	appe <mark>n</mark> automa	atically upon public	cation of data in the IOP
Pass criteria	Data from the battery is correctly database	appended to	the historic regist	ry held in the long-term
Suspension criteria				
Results	Test successful The long-term database succes monitored by WiseCORP	ssfully stores	all information	sent by the batteries





i - 📄 🖬 🕨 🖩 🔅	8	
 ▷ I WiseCORP_Server ▷ I config ▷ I config ▷ I ch_bigdata_wgcockpit 	Welcome × * db.getCollection('wisec x wisegridpre wisegridpre bb.bid:27017 rb_bigdata_wisecorp	
▷ B rb_bigdata_wgcockpit	db.getCollection('wisecorp_batteries').find({id:'A	<pre>MPTest'}).sort({updatedTime:-1})</pre>
▲ 🗟 rb_bigdata_wisecorp	wisecorp_batteries () 0.169 sec.	4 0 50
4 📙 Collections (11)	Key Value	Туре
calendar	(1) ObjectId("5b5ab71265edd90006cf { 24 fields }	Object
updates	(2) ObjectId("5b5ab70d65edd90006cf { 24 fields }	Object
wisecorp_assetssta	Operation (3) Objectio("505ab70865edd90006ct { 24 fields }	Object
wisecorp_batterie	A model of the second secon	Object
wisecorp_energyn	V V (5) Objectid(5b5ab6le65edd90006cf { 24 fields }	Object
wisecorp_energyn	V V (6) Objectio(SpSab6196Sedd90006ct { 24 fields }	Object
wisecorp_values_1	V See (7) UDIECTID(5058001465e0090006ct { 24 TIELDS }	Object
b wisecorp_values_1	V V (8) Objectio(SpSaboetoSedd9000oct { 24 fields }	Object
wisecorp_values_1	V U (9) Objectio(SpSabbeabSedd90006ct { 24 fields }	Object
wisecorp_weather	Image: Section (10) Objection (20) Spoableoobood 90006c { 24 fields }	Object
wisecorp_weather	to b (11) ObjectId("5b5ab6e065edd90006c { 24 fields }	Object
Functions	(12) ObjectId("5b5ab6db65edd90006 { 24 fields }	Object
▷ 📜 Users	I3) ObjectId("5b5ab6d665edd90006 { 24 fields }	Object
▷ 🗐 rb_bigdata_wiseevp	III (14) ObjectId("5b5ab6d165edd90006 { 24 fields }	Object
▷	(15) ObjectId("5b5ab6cc65edd90006c { 24 fields }	Object
▷ 🗟 rb_wisecoop	Image: Wight the state of th	Object
▷ 🗟 rb_wisecorp	III (17) ObjectId("5b5ab6c265edd90006c { 24 fields }	Object
▷ 🗐 rb_wiseevp	Image: Second	Obiect
Logs		

Name	RTM008. Store HVAC data to Long-	term DB		
Module under test	RT Monitor		Resp.	ETRA
Module requirement	HL-UC 7_PUC_1_Dynamic manager	nent of dem	nand side assets in	tertiary sector
	HVAC Wrapper simulator IOP platform up and running			
Features to be tested	Data from HVAC is properly collecte	ed in the lor	ng-term database o	of WiseCORP (big data)
Features not to be tested				
Preparation				
Dependencies	RTM005. Read HVAC data from IOP			
Steps	The execution of this test must h <mark>ap</mark>	<mark>pe</mark> n automa	atically upon public	cation of data in the IOP
Pass criteria	Data from the HVAC is correctly an database	pended to	the historic regist	r <mark>y he</mark> ld in the long-term
Suspension criteria				
Results	Test successful The long-term database successfull by WiseCORP	y stores all	information sent b	by the HVACs monitored





\land Robo 3	and the second	Andrew Statement of A			_ 🗆 🗙
File View	w Options Window Help	0			
	ੇ 🖬 🕨 🔳 🍪 🦳				
	iseCORP_Server	• * db.getCollection('wisec ×			
⊳ 🗎 co	-	🚽 wisegridpre 🗏 wisegridpre.lab.id:27017 📄 r	b bigdata wisecorp		
	_bigdata_wgcockpit	db.getCollection('wisecorp asse).sort({updatedTime:-1	+)
	_bigdata_wisecoop	wisecorp assetsstatus (1) 0.017 sec.		-	
	C-II(11)				[] [
	calendar	Key	Value	Туре	
	updates	(1) ObjectId("5b5a362f65edd90006ce	{ 7 fields }	Object	
	wisecorp_assetsstatus	🔲 _id	ObjectId("5b5a362f65edd90006cedec4")	ObjectId	-
		aptureTime	2018-07-26 20:59:27.622Z	Date	
	wisecorp_batteries	status	1	Int32	
	wisecorp_energymix	📼 value	24	String	
	wisecorp_energymix	id id	HVAC001	String	
	wisecorp_values_15m	📼 operation	change	String	
	wisecorp_values_1d	👼 updatedTime	2018-07-26 20:59:27.652Z	Date	
	wisecorp_values_1h	▲ (2) ObjectId("5b5a35f365edd90006ce	{ 7 fields }	Object	
	wisecorp_weather		ObjectId("5b5a35f365edd90006cedeb3")	ObjectId	
	wisecorp_weatherfo	👼 captureTime	2018-07-26 20:58:27.622Z	Date	
	Functions	iii status	1	Int32	
	Users	i value	24	String	
	_bigdata_wiseevp	id id	HVAC001	String	
	_wgcockpit	operation	change	String	
	_wisecoop	🗟 updatedTime	2018-07-26 20:58:27.653Z	Date	
	_wisecorp	Image: ObjectId("5b5a35b765edd90006ce	{ 7 fields }	Object	
⊳ 🐸 rb	_wiseevp	4) ObjectId("5b5a357b65edd90006ce	{ 7 fields }	Obiect	
Logs					

4.2.2 KPI engine tests

Name	KPI001. Associated CO ₂ emissions					
Module under test	KPI engine Resp. ETRA					
Module requirement	HL-UC 7_PUC_1_Dynamic management of demand side assets in tertiary sector					
Test environment	SMX running and sending information to IOP IOP platform up and running RT monitor storing smart meter readings in long-term database Spark server up and running Energy mix provider module up and running (ENTSOE provider)					
	This module must crosscheck the energy mix information with the individual energy demand readouts in order to compute the equivalent CO ₂ emissions					
Features not to be tested						
Preparation						
Dependencies	RTM001. Read smart meter data from IOP RTM007. Store smart meter data to Long-term DB					
Steps	1. Execute Spark job for CO ₂ emissions calculation					
Pass criteria	 The long-term database contains 3 collections with the quarterly, hourly and daily aggregations Each collection contains documents that contain the equivalent CO₂ emissions of 					





	the demand they represent							
Suspension criteria								
Results	V ENTSO-E) in orde	the eque the eque db.getCollection('Meas × wisegridpre wisegridpre.lab.id:27017 wisegridpre.lab.id:27017 Measurements 0.795 sec. Key	/Cl1', metricType:/CO2Emissions', Value (9 fields) etra-id.com/WiseCORP-VIcBuildings-0001-C etra-id.com/WiseCORP/VIcBuildings/0001/Cl Co2Emissions 2018-07-18 13:00:00.000Z 3600 Hour1 3964.88537294156 null true (9 fields) etra-id.com/WiseCORP-VIcBuildings-0001-C etra-id.com/WiseCORP-VIcBuildings/0001/Cl Co2Emissions 2018-07-18 12:00:00.000Z 3600 Hour1 16931.7532556851 null true (9 fields)	of the interval √ ype Object String String Date Int32 String Double Null Boolean Object String String String String String Double Int32 String String String Double Int32 String Date Int32 String Double String Date Int32 String Double String String String String String Double String Double String Double String Double String Double String Double String Double String Double String Double String Double String Double String Double String Double String Double String Double String Double String Double String String String String String String String String String String String String String String String Double String String String String Double String String Double String String String String Double String	dem	nand	ae: {\$g	

Name	KPI002. Associated economic cost	s		
Module under test	KPI engine		Resp.	ETRA
Module requirement	HL-UC 7_PUC_1_Dynamic manage	ment of der	mand side assets ir	etertiary sector
Test environment	SMX running and sending informa IOP platform up and running RT monitor storing smart meter re Spark server up and running Tariff provider module up and run	adings in lor	ng-term database	
Features to be tested	This module must cross-check the energy demand readouts in order			
Features not to be tested				
Preparation				





Dependencies		art meter data from IO	F			
	TMUU7. Store sm	art meter data to Long	-term DB			
Steps	1. Execute Spark job for economic costs calculation					
Pass criteria	 The long-term database contains 3 collections with the quarterly, hourly and daily aggregations Each collection contains documents that contain the associated economic costs of the demand they represent 					
Suspension criteria						
V Ci		associated to the dem	and WiseCORP_Server nterval: 3600, value: (\$gt:0))) Value (9 fields) etra-id.com.WiseCORP-VIcBuildings-0001-C etra-id.com.WiseCORP.VIcBuildings/0001/C1 EnergyCost 2018-07-18 13:00:00:000Z 3600 Hourl 21:546 null true (9 fields) etra-id.com.WiseCORP-VIcBuildings-0001-C etra-id.com.WiseCORP-VIcBuildings/0001/C1 EnergyCost 2018-07-18 12:00:00:000Z 3600 Hourl 9:20732400000083 null true	licable tariff in order to		

Name	KPI003. Distribution of demand pe	er tariff perio	od		
Module under test	KPI engine		Resp.	ETRA	
Module requirement	HL-UC 7_PUC_1_Dynamic management of demand side assets in tertiary sector				
Test environment	5MX running and sending information to IOP OP platform up and running RT monitor storing smart meter readings in long-term database Spark server up and running Fariff provider module up and running				





Features to be tested	This module must crosscheck the energy demand of the building with the periods of the contracted tariff, giving an overview of the distribution of the demand over a period of time (e.g. monthly)
Features not to be tested	
Preparation	
Dependencies	RTM001. Read smart meter data from IOP RTM007. Store smart meter data to Long-term DB
Steps	 Open the WiseCORP UI Access the building indicators site
Pass criteria	The distribution of demand per tariff period for the selected month is presented
Suspension criteria	
Results	Test successful The UI displays the required information, calculated over the data of a complete month and a building Energy consumption by tariff period p3: 25.04 % p1: 31.94 %

Name	KPI004. Calculat	ion of indicators per bu	ilding. Mont	hly economic costs		
Module under test	KPI engine		Resp.	ETRA		
	HL-UC 7_PUC_1_Dynamic management of demand side assets in tertiary sector					
	SMX running and sending information to IOP IOP platform up and running RT monitor storing smart meter readings in long-term database					
Features to be tested	This module an	alyses historic informa	This module analyses historic information available in the long-term			





	database to provide insights on the economic costs faced by the measured facilities				
Features not to be tested					
Preparation					
Dependencies	KPI002. Associated economic costs				
Steps	 Open the WiseCORP UI Access the building indicators site 				
Pass criteria	The UI represents the total monthly economic cost per building				
Suspension criteria					
Results	Test successful The UI displays the required information, calculated over the data of a complete month and a building Energy Cost 359.99 €				

Name	KPI005. Calculation of indicators per building. CO ₂ emissions				
Module under test	KPI engine	Resp.	ETRA		
Module requirement	HL-UC 7_PUC_1_Dynamic manageme tertiary sector	ent of dema	and side assets in		
Test environment	SMX running and sending information to IOP IOP platform up and running RT monitor storing smart meter readings in long-term database				
Features to be tested	This module analyses historic information available in the long-term database to provide insights on the equivalent CO ₂ emissions produced by the measured facilities				
Features not to be tested					
Preparation					
Dependencies	KPI001. Associated CO₂ emissions				
Steps	 Open the WiseCORP UI Access the building indicators site 				
Pass criteria	The UI represents the total monthly equivalent CO ₂ emissions per building				
Suspension criteria					
Results	Test successful The UI displays the required information, calculated over the data of a				





complete month and a building	
CO ₂ Emissions 777.7 kg	

Name	KPI006. Calculation of indicators per building. Total demand				
Module under test	KPI engine		Resp.	ETRA	
Module requirement	HL-UC 7_PUC_2 tertiary sector	HL-UC 7_PUC_1_Dynamic management of demand side assets in tertiary sector			
Test environment	IOP platform up	SMX running and sending information to IOP IOP platform up and running RT monitor storing smart meter readings in long-term database			
Features to be tested	This module analyses historic information available in the long-term database to provide insights on the total energy demand of the measured facilities				
Features not to be tested					
Preparation					
Dependencies	RTM001. Read smart meter data from IOP RTM007. Store smart meter data to Long-term DB				
Steps	 Open the WiseCORP UI Access the building indicators site 				
Pass criteria	The UI represents the total monthly demand per building				
Suspension criteria					
Results	Test successful The UI displays complete mont	the required information h and a building Energy Consumption 3 MWh		d over the data of a	

Name	KPI007. Calculation of indicators per building. Total production			
Module under test	KPI engine Resp. ETRA			
	HL-UC 7_PUC_1_Dynamic management of demand side assets in tertiary sector			





Features to be tested database to provide insights on the total energy production of measured facilities Features not to be tested Preparation Dependencies RTM001. Read smart meter data from IOP RTM007. Store smart meter data to Long-term DB Steps 1. Open the WiseCORP UI 2. Access the building indicators site Pass criteria The UI represents the total monthly production per building Suspension criteria Test successful The UI displays the required information, calculated over the data complete month and a building	Test environment	SMX running and sending information to IOP IOP platform up and running RT monitor storing smart meter readings in long-term database		
Preparation Dependencies RTM001. Read smart meter data from IOP RTM007. Store smart meter data to Long-term DB Steps 1. Open the WiseCORP UI 2. Access the building indicators site Pass criteria The UI represents the total monthly production per building Suspension Test successful The UI displays the required information, calculated over the data or complete month and a building	Features to be tested	This module analyses historic information available in the long-term database to provide insights on the total energy production of the measured facilities		
Dependencies RTM001. Read smart meter data from IOP RTM007. Store smart meter data to Long-term DB Steps 1. Open the WiseCORP UI 2. Access the building indicators site Pass criteria The UI represents the total monthly production per building Suspension Test successful The UI displays the required information, calculated over the data complete month and a building	Features not to be tested			
Dependencies RTM007. Store smart meter data to Long-term DB Steps 1. Open the WiseCORP UI 2. Access the building indicators site Pass criteria The UI represents the total monthly production per building Suspension criteria Test successful The UI displays the required information, calculated over the data complete month and a building	Preparation			
Steps 2. Access the building indicators site Pass criteria The UI represents the total monthly production per building Suspension criteria Test successful The UI displays the required information, calculated over the data complete month and a building	Dependencies			
Suspension criteria Test successful The UI displays the required information, calculated over the data complete month and a building	Steps			
criteria Test successful The UI displays the required information, calculated over the data complete month and a building	Pass criteria	The UI represen <mark>ts</mark> the total monthly production per building		
The UI displays the required information, calculated over the data	-			
Results Energy Production 172.6 kWh	Results	The UI displays the required information, calculated over the data of a complete month and a building Energy Production		

Name	KPI008. Calculation of indicators per building. Self-consumption ratio			
Module under test	KPI engine		Resp.	ETRA
Module requirement	HL-UC 7_PUC_1_Dynamic management of demand side assets in tertiary sector			
Test environment	SMX running and sending information to IOP IOP platform up and running RT monitor storing smart meter readings in long-term database			
Features to be tested	This module analyses historic information available in the long-term database to provide insights on the total self-consumption and the energy produ <mark>ction</mark> surplus of the measured facilities			
Features not to be tested				
Preparation				
Dependencies	RTM001. Read smart meter data from IOP RTM007. Store smart meter data to Long-term DB			
Steps	 Open the WiseCORP UI Access the building indicators site 			
Pass criteria	The UI represents the self-consumption ration for the selected month			





	per building The UI represents the total production surplus for the selected month per building
Suspension	
criteria	
Results	Test successful The UI displays the required information, calculated over the data of a complete month and a building
	Self-consumption 5.753 %

4.2.3 Forecast modules tests

Name	FOR001. Demand/production forecasting training				
Module under test	WiseCORP for	ecas	t module	Resp.	ITE
Module requirement	HL-UC 7_PUC_1_Dynamic management of demand side assets in tertiary sector				
Test environment	WiseCORP forecast module up and running Historical data available in long-term DB				
Features to be tested	WiseCORP forecast module is trained				
Features not to be tested					
Preparation					
Dependencies	RTM002. Read sensor data from IOP				
Steps	Perform WiseCORP forecast training				
Pass criteria	Training MAP <mark>E bel</mark> ow pre-defined threshold				
Suspension criteria					
Results	Test successfu WiseCORP for		t model trained		

Name	FOR002. Demand/Production forecasting		
Module under test	WiseCORP forecast module	Resp.	ITE





Module requirement	HL-UC 7_PUC_1_Dynamic management of demand side assets in tertiary sector			
Test environment	WiseCORP forecast module up and running Historical data available in long-term DB			
Features to be tested	WiseCORP forecast module performs demand/production forecasting training			
Features not to be tested				
Preparation	Train WiseCORP demand/production forecast module			
Dependencies	FOR001. Demand/production forecasting training RTM002. Read sensor data from IOP			
Steps	WiseCORP forecast module			
Pass criteria	Prediction MAPE below pre-defined threshold			
Suspension criteria				
Results	Test successful 24 hours hourly load and production prediction			

Name	FOR003. Request message parsing test of WiseCORP forecast module		
Module under test	WiseCORP forecast module Resp. ITE		
	HL-UC 7_PUC_1_Dynamic management of demand side assets in tertiary sector		
Test environment	Development RabbitMQ environment WiseCORP forecast module up and running Historical data available in long-term DB		
Features to be tested	Performance of <mark>Wis</mark> eCORP forecast module, at parsing foreca <mark>st</mark> queries.		
Features not to be tested			
Preparation	Enable RabbitMQ queues, and run WiseCORP forecast module		
Dependencies			
Steps	 Receipt of request Request parsing DB request according to the requested data Treatment of the retrieved data 		
Pass criteria	 The forecast module is able to decode the queries properly The forecast module is able to retrieve information from the long-term DB with the parsed information 		
Suspension criteria			





Results The module is able to parse the request messages of retrieve information from the long-term DB.	and process it to
---	-------------------

Name	FOR004. Forecast response message generation test of WiseCORP forecast module
Module under test	WiseCORP forecast module Resp. ITE
Module requirement	HL-UC 7_PUC_1_Dynamic management of demand side assets in tertiary sector
Test environment	RabbitMQ environment WiseCORP forecast module up and running Historical data available in long-term DB
Features to be tested	Performance of WiseCORP forecast module, at generating and submitting the forecast response.
Features not to be tested	
Preparation	Enable RabbitM <mark>Q qu</mark> eues Run the forecast module
Dependencies	
Steps	 Parsing of the forecasting algorithm output Generating forecast response message
Pass criteria	 The forecast module is able to analyse properly the output provided by the forecasting algorithm The forecast module is able to generate properly the forecast response message
Suspension criteria	
Results	The module is able to analyse the information provided by the forecast algorithm, and generates the response.

Name	FOR005. Forecast is periodically triggered			
Module under test	Forecast orchestrator Resp. ETRA			
	HL-UC 7_PUC_1_Dynamic management of demand side assets in tertiary sector			
Test environment	Internal ESB up and running			





	Historical data available in long-term DB		
Features to be tested	WiseCORP periodically posts a demand and a production forecast		
	request per bus to the corresponding queue of the internal ESB		
Features not to be tested			
Preparation	Open RabbitMQ monitor, monitor demand and production forecast		
	queues		
Dependencies			
Steps	2. Execute forecast orchestrator module		
	 Periodically, every hour, one request per smart meter appears 		
Pass criteria	in the demand and production forecast queues		
	Requests claim next 24 hours hourly prediction		
Suspension			
criteria			
	Test successful		
	The following <mark>ex</mark> tract of logs of the Docker co <mark>nt</mark> ainer		
	wisecoop_forec <mark>astbr</mark> idge_demand_1 shows that one forecast que <mark>ry</mark> for		
	each asset is bei <mark>ng p</mark> osted every hour.		
	etraid@wis <mark>egri</mark> dpre:~\$ docker logs -ttai <mark>l</mark>		
	1000 wise <mark>corp</mark> _forecastbridge_demand_1 gre <mark>p</mark>		
	querying grep etra-id-com-WiseCORP-		
Results	VlcBuildings-0001-C1		
	2018-07-19T11:15:06.654098746Z [etra-id-com-		
	WiseCORP-VlcBuildings-0001-C1] querying		
	2018-07-19T12:15:09.975139716Z [etra-id-com-		
	WiseCORP-VlcBuildings-0001-C1] querying		
	2018-07-19T13:15:04.758048669Z [etra-id-com-		
	WiseCORP-VlcBuildings-0001-C1] querying		
	2018-07-19T14:15:04.668450527Z [etra-id-com-		
	WiseCORP-VlcBuildings-0001-C1] querying		

Name	FOR006. Forecast results are saved	d to operation	al DB	
Module under test	Forecast orchestrator		Resp.	ETRA
Module requirement	HL-UC 7_PUC_1_Dynamic management of demand side assets in tertiary sector			
l est environment	nternal ESB up and running Historical data available in long-term DB Demand and production forecast modules up and running			
	WiseCORP receives the results of the forecast module, formats them following the same format used to store real-time data, and stores the in the operational database			
Features not to be tested				





Preparation	Open operational databas	se, query next 24 hour	s of demand/producti	on forecasts
Dependencies				
Steps	1. Execute forecast	orchestrator module		
Pass criteria	Periodically, every hour, next 24 hours forecast metrics get updated in the operational database			
Suspension criteria				
Results	Test successful Results of the forecast mod Robo 3T - 12 File View Options Window Help Solution Stream	 ♦ * db.getCollection('wisec × ♦ * db.get wisegridpre wisegridpre.lab.id:270) 	etCollection('wisec × 	

4.2.4 Tariff comparer tests

Name	TC001. Create simulated bill for b	uilding		
Module under test	Tariff comparer		Resp.	ETRA
Module requirement	HL-UC 7_PUC_1_Dynamic management of demand side assets in tertiary sector			
Test environment	WiseCORP UI up and running Existing predefined set of tariffs			
	WiseCORP facilitates to ESCOs the ability to simulate bills according to historical demand data and tariff definitions			
Features not to be tested				
Preparation	Open WiseCORP UI			





Dependencies							
Steps	 Go to th Select a Trigger b 	buildin	5				
Pass criteria	The user interfa tariffs The user interfac				•	on of results of th lected criteria	ne simulated
Suspension criteria							
	Test successful The tariff optimi for any of the ta May 20: Site	riffs cre 18				cessfully create a s	simulated bill
	Valencia	•	Global cost		E > Ahorr	o Pro Luz 3.0 (Aldro Energía)	•
	Month Days May 2018 31	Entities	Peak active power (kW)	at 23/05/2018 11	:45	Total consumption (kWh)	Total cost (€) 4751.66
	Plan name Ahorro Pro Luz 3.0 🗹		Company	Tariff type	Green	•	Discount
Results	Concept		Aldro Energía Price	Value	Yes	Yes	Yes
	Contracted power (P1)		0.116746 €/kW/day	315 kW			1140.02€
	Contracted power (P2)		0.072233 €/kW/day	286 kW			640.42€
	Contracted power (P3)		0.049977 €/kW/day	185 kW			286.62€
						Total contracted power	2067.06€
	Consumed energy (P1)		0.140545 €/kWh	20717 kWh			2911.67€
	Consumed energy (P2)		0.119795 €/kWh	39179 kWh			4693.45€
	Consumed energy (P3)		0.08652€/kWh	16803 kWh			1453.8€
						Total consumed energy	9058.91€
	Energy consumption dis	count	20 %	9058.91€			-1811.78€
						Total discounts	-1811.78€
	Energy taxes		5.11269632%	9314.19€			476.21€
	VAT		21%	9790.4€			2055.98€
						Global total	11846.38€

4.2.5 Energy usage optimizer





Name	EUO001. Unit testing				
Module under test	Energy Usage Optimizer		Resp.	ETRA	
	HL-UC 7_PUC_2_Dynamic aggregatio into energy market	n of distrik	outed energy assets	and active participation	
Test environment	MATLAB Predefined set of input curves (prices Model of 2 HVACs and 1 battery	, energy d	emand)		
	The energy usage optimizer must ca given assets, considering usage calen		•	s long schedule for the	
Features not to be tested					
Preparation	Prepare input files for algorithm				
Dependencies					
Steps	1. Run algorithm on MATLAB with prepared input files				
Pass criteria	The optimum energy usage distribution for the simulated 24 hours is produced				
Suspension criteria					
Results	Test successful As an example, the results of the opti 2 HVACs, PV production and subje economic cost minimization). 3000 2000 1000 0 1 2 3 4 5 6 7 8 -1000 0 1 2 3 4 5 6 7 8 -PRODUCTION (Wh) DEMAND (Wh) HVA SUPPLY DEMAND (W	9 10 11 9 10 11 C002 -	e-period tariff are	presented (subject to 18 19 20 21 22 23 C001	







Name	EUO002. Produce day-ahead optimum schedule for assets				
Module under test	Energy Usage Optimizer Resp. ETRA				
Module requirement	HL-UC 7_PUC_2_Dynamic agg <mark>regati</mark> on of distributed energy assets and active participation into energy market				
Test environment	WiseCORP lab-testing environment Operational database contains necessary inputs, including the list of assets, usage calendar and dynamic prices from the DR framework module				
Features to be tested	Upon completion of the execution of the energy usage optimizer module, results are stored in the operational database of WiseCORP				
Features not to be tested					
Preparation	Prepare lab-testing platform to m <mark>imic</mark> the context of unit testing				
Dependencies	EUO002. Read dynamic price				
Steps	1. Executed energy usage optimizer module				
Pass criteria	After algorithm execution comple <mark>tes,</mark> the operational database of WiseCORP contains the 24-hour schedule for the configu <mark>red a</mark> ssets.				
Suspension criteria					
Results	Test successful The results of the Energy Usage Optimizer are successfully transferred to the operational database, therefore available to other modules of the tool				





🙁 Robo 3T - 1.2			
File View Options Window He	lp		
🛃 – 📄 🖬 🕨 🔳 🗱			
connections cronHistory events	● * db.getCollection('wisec × wisegridpre ■ wisegridpre.lab.id:27017	rb_wisecorp	
▷ layers	db.getCollection('wisecorp_sche	edule').find({assetID:'STOR001	'}).sort({captureTime:-1})
▷ □_messages	wisecorp_schedule 🕔 0.007 sec.		 0 50 50
▶schema	Key	Value	Туре
▷ _ cronHistory			
▷ meteor_accounts_Io	▲		Object
▷ users	id id	STOR001_2018-06-13T22:00:00.0000000	String
wisecorp_alerts	assetID	0-1-166-7-0-1 STOR001	String
▷ wisecorp_analysis	w value	915.84	String Double
▷ ■ wisecorp_assets	i unit	915.84 W	String
wisecorp_assetsstatus		1	Int32
wisecorp_batteries	a captureTime	2018-06-13 20:00:00.000Z	Date
wisecorp_chats	e description	storage	String
wisecorp_energyCost	= command	assetDispatcher	String
wisecorp_energymix	= state	manual	String
wisecorp_energymix	▲ (2) STOR001_2018-06-13T21:00:00.000		Object
wisecorp_groupsOfT	id	STOR001 2018-06-13T21:00:00.0000000	String
▷ 🔤 wisecorp_measurem 🗄	id _id	0-1-166-7-0-1	String
wisecorp_schedule	assetID	STOR001	String
wisecorp_spotprices	· value	356.96	Double
wisecorp_tariffRequ	= unit	W	String
wisecorp_things	··· status	1	Int32
b isecorp_weather	aptureTime	2018-06-13 19:00:00.000Z	Date
wisecorp_weatherfo	== description	storage	String
Functions	= command	assetDispatcher	String
> 📙 Users	= state	manual	String
▷ 🗟 rb_wiseevp	Glipping State (3) STOR001_2018-06-13T20:00:00.000		Object
Logs			*

4.2.6 DR framework tests

Name	DRF001. Estima	te occupant thermal/vis	ual comfort	profile
Module under test	Profiling		Resp.	НҮР
Module requirement	HL-UC 7_PUC_ tertiary sector	1_Dynamic manageme	nt of dema	ind side asse <mark>t</mark> s in
requirement				
Test environment	WiseCORP up and running, temperature/ humidity/ luminance sensors reporting ambient conditions, HVAC operational setpoint monitoring capabilities			
Features to be tested	Generation of t <mark>he co</mark> mfort profile for the individual occupants regarding thermal and visual comfort			
Features not to be tested	Optional			
Preparation	WiseCORP up a	<mark>nd ru</mark> nning, sensor/ actu	uator networ	k working properly
Dependencies				
Steps	about a 2. Data cle 3. Launch	o internal WiseCORP I mbient conditions, HVA eaning, filtering and noru of machine learning algo onal to data validity/val	C operationa malization prithm to upo	al setpoint





Vace critoria	Availability of comfort profile per building thermal zone or occupant (depending on feasibility/granularity of control)
Suspension criteria	Non-availability of data regarding ambient conditions or HVAC setpoint
Results	Test successful

Name	DRF002. Calculate human-centric demand flexibility of building			
Module under test	Flexibility Estimation Resp. HYP			
Module requirement	HL-UC 7_PUC_1_Dynamic management of demand side assets in tertiary sector			
Test environment	WiseCORP up and running, temperature/ humidity/ luminance sensors reporting ambient conditions, energy (sub)metering data, HVAC operational setpoint monitoring capabilities			
Features to be tested	Estimation of demand flexibility time series			
Features not to be tested				
Preparation	 Association of (sub)metering information with HVAC/lighting system setpoints to calculate the impact of devices on energy consumption Set up stream of data from sensors, energy meters and device setpoints by capturing the messages on the WiseCOOP internal ESB 			
Dependencies	EUO003. Produce day-ahead optimum schedule for assets DRF008. Estimate occupant thermal/visual comfort profile			
Steps	 Create energy models of HVAC system and lighting system Obtain day-ahead building asset operational setpoint schedule Estimate available flexibility based on comfort profiles and asset operational schedule Generate demand flexibility forecast 			
Pass criteria	Continuous generation and dispatch of demand flexibility forecasts per time interval			
Suspension criteria				
Results	Test successful			

Name	DRF003. Receive request to activate demand flexibility
------	--

E

7





Module under test	Control Optimization	Resp.	НҮР		
Module requirement	HL-UC 7_PUC_1_Dynamic management of demand side assets in tertiary sector				
Test environment	WiseCORP up and running, WiseCOOP up and running, IOP up and running				
Features to be tested	Reception of the appropriate message from WiseCOOP specifying the detailed break-down of demand flexibility per device to be activated				
Features not to be tested					
Preparation	Listen to the appropriate queue of the IOP for messages from WiseCOOP				
Dependencies					
Steps	 Instantiate listener to wait for n the IOP Upon reception of message, va 	-			
Pass criteria	The message – an example of which can found in Table 19 – has been correctly received and validated.				
Suspension criteria					
Results	Test successful				

Name	DRF004. Asset schedule modification		
Module under test	DR framework Resp. HYP		
Module requirement	HL-UC 7_PUC_1_Dynamic management of demand side assets in tertiary sector		
Test environment	WiseCORP up and running		
Features to be tested	Estimation of optimal setpoint per device and dispatch to the "asset dispatcher" component that sends the setpoints to the device wrappers.		
Features not to be tested			
Preparation	WiseCOOP has sent a request for modification of the demand of specific assets.		
Dependencies			
Steps	 Decode the incoming message from WiseCOOP Identify the target demand modification per asset Translate demand modification to target setpoint Send target setpoint per device to the WiseCORP asset 		





4.2.7 Asset dispatcher	
Results	Test successful
Suspension criteria	
Pass criteria	The setpoints that will induce the required demand modifications per device are communicated to the asset dispatcher component.
	dispatched component by placing them in the appropriate queue of the WiseCORP-internal ESB.

4.2.7 Asset dispatcher

Name	AD001. Load schedule from operational database				
Module under test	Asset dispatcher			Resp.	ETRA
Module requirement	HL-UC 7_PUC_2_Dynamic aggregation of distributed energy assets and active participation into energy market				
Test environment	WiseCORP lab- <mark>testin</mark> g environment Operational database contains the schedule for each controllable asset configured in the building				
Features to be tested	The asset dispatcher module can read from operational database all necessary information about the schedule of the controllable assets				
Features not to be tested					
Preparation	Prepare lab-test	ing	platform to mimic t	<mark>he</mark> context o	f unit testing
Dependencies	EUO003. Produ <mark>ce d</mark> ay-ahead optimum schedule for assets				
Steps	1. Executed asset dispatcher				
Pass criteria	Asset dispatche <mark>r ha</mark> s access to the operational database and is able to retrieve the cur <mark>rent</mark> schedules for each controllable asset				
Suspension criteria					
Results			ner accesses the of the operational d		contained in the cessfully

Name	AD002. Read current asset status from operational database		
Module under test	Asset dispatcher	Resp.	ETRA
Module	HL-UC 7_PUC_2_Dynamic aggregation of	of distributed	d energy assets and





requirement	active participation into energy market		
Test environment	WiseCORP lab-testing environment Operational database contains the current status (setpoint) of each one of the controllable assets		
Features to be tested	The asset dispatcher module can read from operational database all necessary information about the current setpoint executed by the controllable assets		
Features not to be tested			
Preparation	Prepare lab-testing platform to mimic the context of unit testing		
Dependencies	RTM003. Read CHP data from IOP RTM004. Read battery data from IOP RTM005. Read HVAC data from IOP		
Steps	1. Executed asset dispatcher		
Pass criteria	Asset dispatcher has access to the operational database and is able to retrieve the current status for each controllable asset		
Suspension criteria			
Results	Test successful The asset dispatcher accesses the information contained in the current status collection of the operational database successfully		

Name	AD003. Detect deviation from schedule		
Module under test	Asset dispatcher	Resp. ETRA	
Module requirement	HL-UC 7_PUC_2 <mark>_Dy</mark> namic aggregation of distributed energy assets and active participation into energy market		
Test environment	WiseCORP lab-t <mark>esti</mark> ng environment		
Features to be tested	Given a point in time when current setpoint and scheduled setpoint differs for a controllable asset, the asset dispatcher must be able to detect the incoherence		
Features not to be tested			
Preparation	Prepare lab-testing platform to mimic the context of unit testing Use HVAC simulator to set current setpoint to value that differs from the scheduled one		
Dependencies	AD001. Load schedule from operational database AD002. Read current asset status from operational database		
Steps	1. Executed asset dispatcher		
Pass criteria	Asset dispatcher detects the deviation in the schedule and logs this detection to the log file		





Suspension criteria	
Results	Test successful The asset dispatcher successfully compares the necessary information to decide whether a new setpoint must be dispatched

Name	AD004. Trigger asset setpoint		
Module under test	Asset dispatcher Resp. ETRA		
Module requirement	HL-UC 7_PUC_2_Dynamic aggregation of distributed energy assets and active participation into energy market		
Test environment	WiseCORP lab-testing environment		
Features to be tested	Given a point in time when current setpoint and scheduled setpoint differs for a controllable asset, the detection of this incoherence must result in the publication of a command to the controllable asset to set the appropriate setpoint		
Features not to be tested			
Preparation	Prepare lab-testing platform to mimic the context of unit testing Use HVAC simulator to set current setpoint to value that differs from the sch <mark>eduled</mark> one		
Dependencies	AD003. Detect deviation from schedule		
Steps	 Executed asset dispatcher Connect to IOP and monitor commands to controllable assets (MQTT) 		
Pass criteria	Asset dispatche <mark>r pu</mark> blishes a command targeting the controlled asset in which the devia <mark>tion</mark> was detected		
Suspension criteria			
Results	Test successful The following extract of the logs of the assets dispatcher shows the command dispatched via MQTT upon detection of a deviation from the current status of an HVAC assets and its schedule etraid@wisegridpre:~\$ docker logstail 0 -f wisecorp_asset_dispatcher_1 >> [HVAC001/SHIC01/0-1-160-7-0-1] { "_id" : "0- 1-160-7-0-1", "assetID" : "HVAC001", "value" : "25", "unit" : "grdC", "status" : 1, "captureTime" : ISODate("2018-07- 19T15:16:03.864Z"), "description" : "modbus", "mode" : "cooling", "command" : "auto", "state" : "manual" }		



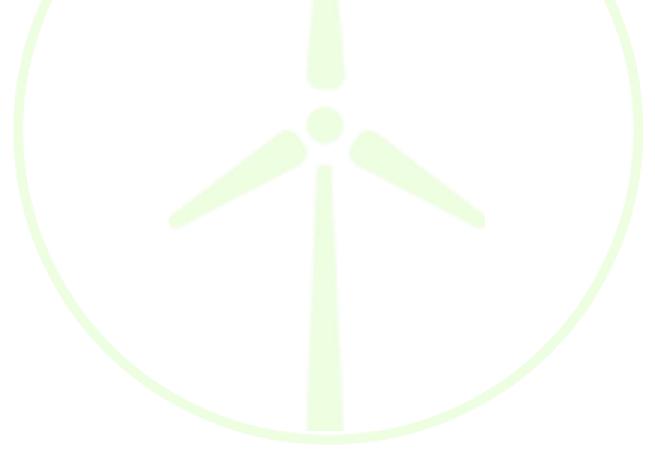


5 CONCLUSIONS AND NEXT STEPS

The main conclusion of the work presented in this deliverable is that the methodology followed during the implementation and lab-testing phase was optimal for both tools. The standardization of a process and its explanation to the partners involved in this phase, allowed to avoid any misunderstanding and to follow the same steps so the final result is a coherent and homogeneous work.

All the tests and activities performed within this deliverable have been successful even if, in some cases, it has been necessary to refine the implementation of the modules and repeat the tests, which allowed the involved partners to better understand the singularities of each module.

Although it has not been possible to make all the test that the partners would like to perform for this tool, the main ones and some complementary ones have been done. For Task 14.2 "WiseGRID integrated ecosystem Lab-Testing" more tests will be performed in order to prove the integration of the different tools together. During the deployment and demonstration phases, as all the tools will be integrated in real-life conditions and the consortium will have better knowledge of the particularities of each Pilot Site, the partners will be able to collect some feedback and continuing refining the tools and perform some more tests in order to develop the tools and optimally adapt them for the different Pilot Sites.







6 REFERENCES AND ACRONYMS

6.1 REFERENCES

- [1] Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Software_test_documentation. [Accessed 14 July 2016].
- [2] The International Software Testing Standard, "ISO/IEC/IEEE 29119 Software Testing," 10 September 2014. [Online]. Available: http://www.softwaretestingstandard.org/. [Accessed 14 July 2016].
- [3] G. Arnold, D. Craciun, W. Heckmann and N. Schäfer , "Utility-scale PV installations and their challenges in grid-code compliance testing," *storage & grids: Technical Briefing*, pp. 74-78, February 2015.
- [4] "D7.1 WiseCOOP and WiseCORP Apps Design".
- [5] [Online]. Available: https://bl.ocks.org/rpgove/0060ff3b656618e9136b.
- [6] "D10.2 WiseGRID Flexibility-based DR Optimization Framework Specifications".

Acronyms List		
AMI	Advanced Metering Infrastructure	
AMQP	Advanced Message Queuing Protocol	
СНР	Combined Heat Power	
DB	Data Base	
DR	Demand Response	
DSO	Distribution System Operator	
ENTSOE	European Network of Transmission System Operators for Electricity	
ESB	Enterprise Service Bus	
ESCO	Energy Service COmpany	
GUI	Graphical User Interface	
HL-UC	High Level Use Case	
НТТР	Hypertext Transfer Protocol	
HVAC	Heating, Ventilation and Air Conditioning	
IOP	InterOperable Platform (one of the WiseGRID products)	
КРІ	Key Performance Indicator	
	•	

Table 27 – Acronyms list

6.2 ACRONYMS





MQTT	Message Queue Telemetry Transport
RES	Renewable Energy Source
RPC	Remote Procedure Call
RT	Real-Time
SMX	Smart Meter eXtension
UI	User Interface
USEF	Universal Smart Energy Framework

