Innovation Action

inteGRIDy

integrated Smart GRID Cross-Functional Solutions for Optimized Synergetic Energy Distribution, Utilization & Storage Technologies

H2020 Grant Agreement Number: 731268

WP5 – Deployment of inteGRIDy Framework at Pilot Site Areas

D5.3 – Report on inteGRIDy Framework deployment at Pilot Sites

Document Info	
Contractual Delivery Date:	31/01/2020
Actual Delivery Date:	31/01/2020
Responsible Beneficiary:	ATOS
Contributing Beneficiaries:	AIGUASOL, ATOS, CERTH, ENG, M7, POLIMI, SIEMENS, SIVEVO, TEES, TREK, UCY, UNE, VPS.
Dissemination Level:	Public
Version:	1.0
Type:	Report

This project has received funding from the European Union's H2020 research and innovation programme under the grant agreement No **731268**. This report reflects only the author's view and the Commission is not responsible for any use that may be made of the information it contains.

Document Information

Document ID:	**D5.3 Report on inteGRIDy Framework deployment at Pilot Sites**
Version Date:	31/01/2020
Total Number of Pages:	117
Abstract:	The aim of this deliverable is to evaluate the final deployment of each pilot site according to their specific implementation of the inteGRIDy framework, how well the tools integrate (or can be integrated) with each other, and how those integration capabilities were tested to ensure proper deployment.
Keywords:	Deployment, tools integration, integration assessment, integration tests analysis.

Authors

Full Name	Beneficiary / Organisation	Role
Abdon Vivas	ATOS	Overall Editor
Davide Falabretti	POLIMI	Contributor
Jim Fawcett	IWC	Contributor
Massimo Cresta	ASM	Contributor
Alberto Pérez	AIGUASOL	Contributor
Angelina Katsifaraki	TREK	Contributor
Venizelos Efthymiou	UCY	Contributor
Carlos Raposo	ENOVA	Contributor
Dimitris Trigkas	CERTH	Contributor
Otilia Bularca	SIVECO	Contributor

Reviewers

Full Name	Beneficiary / Organisation	Date
Paschalis Gkaidatzis	CERTH	27/01/2020
Lorenzo Corghi	UNE	28/01/2020
Javier Valiño	ATOS	31/01/2020
Athanasios Tryferidis	CERTH	31/01/2020

Version history

Version	Date	Comments
0.1	14/10/2019	ToC
0.2	21/12/2019	Included integration assessments
0.3	22/01/2020	Included integration tests analysis
0.4	29/01/2020	1st review modifications
0.5	29/01/2020	2nd review modifications
1.0	29/01/2020	Final version to be released to the EC

Executive Summary

inteGRIDy framework has been extensively defined in D1.5/D1.6 (inteGRIDy Architecture and Functional/Technical Specifications) [IND15, IND16], where the project architecture, layers and recommendations in the form of functional and non-functional requirements for tools were defined. Using this as baseline, WP4 (Distribution Grid Optimization Framework) conveyed the development of each particular tool, so, in theory, all inteGRIDy tools must have been adapted, at developing time, to those requirements which will ensure the integration.

This is the point where WP5 (Framework Integration and Pilot Site Deployment) takes over. The deployment integration planning is defined in D5.2 (Pilot Areas Deployment Planning) [IND52], whose main objective is to ascertain that both small-scale and large-scale pilot sites accomplish their objectives in terms of the inteGRIDy Reference Architecture.

This report uses all aforementioned documents as reference, and also builds upon the results of D5.1 (inteGRIDy Integrated Prototype) [IND51], including an integration assessment to measure the compliance of each pilot site with the guidelines described in D5.1 regarding the integration at data level (data model and data format) and communication level (interfaces with external systems). The methodology used for such assessment involved requesting an integration survey to be filled out by each pilot site with questions regarding the interfaces of their corresponding tools. Afterwards, the results were analysed in order to assign scores depending on an integration criterion that is also exposed in this document and is related to the inteGRIDy integrated demonstrator (D5.5) [IND55].

Furthermore, this document also gathers the testing reports provided by each pilot site to control that every tool operated as expected in terms of integration and analyses them in order to describe the integration testing approach they followed and give recommendations towards the validation tests that are to come within the next phase of the project.

Table of Contents

Table of Figures

Table of Tables

List of Acronyms and Abbreviations

Term	Description
AC	Alternating Current
ADR	Automated Demand Response
API	Application Programming Interface
BESS	Basic Operation of a Battery Energy Storage System
CIM	Common Information Model
CSV	Comma-Separated Values
DB	Database
DC	Direct Current
DR	Demand Response
EMS	Energy Management System
FTP	File Transfer Protocol
GOFLEX	Generalized Operational FLEXibility for Integrating Renewables in the Distribution Grid
IEC	International Electrotechnical Commission
IoW	Isle of Wight
JPA	Java Persistence API
JSON	JavaScript Object Notation
M2M	Machine-to-Machine
MPC	Model Predictive Control
MQTT	Message Queuing Telemetry Transport
MV	Medium Voltage
OPC	Open Platform Communications
PDF	Portable Document Format
PV	Photovoltaic
REST	Representational State Transfer
RKW	Reference Knowledge Warehouse
SAREF	Smart Appliances REFerence
TCP	Transmission Control Protocol
VEN	Virtual End Node
VM	Virtual Machine
VTN	Virtual Top Node
WS	Web Service
XLS	eXceL Spreadsheet
XML	Extensible Markup Language

1.Introduction

1.1 Scope and objectives of the deliverable

During WP1, the inteGRIDy Reference Architecture was defined in D1.5 [IND15] and D1.6 [IND16] and can be seen in Figure 1.

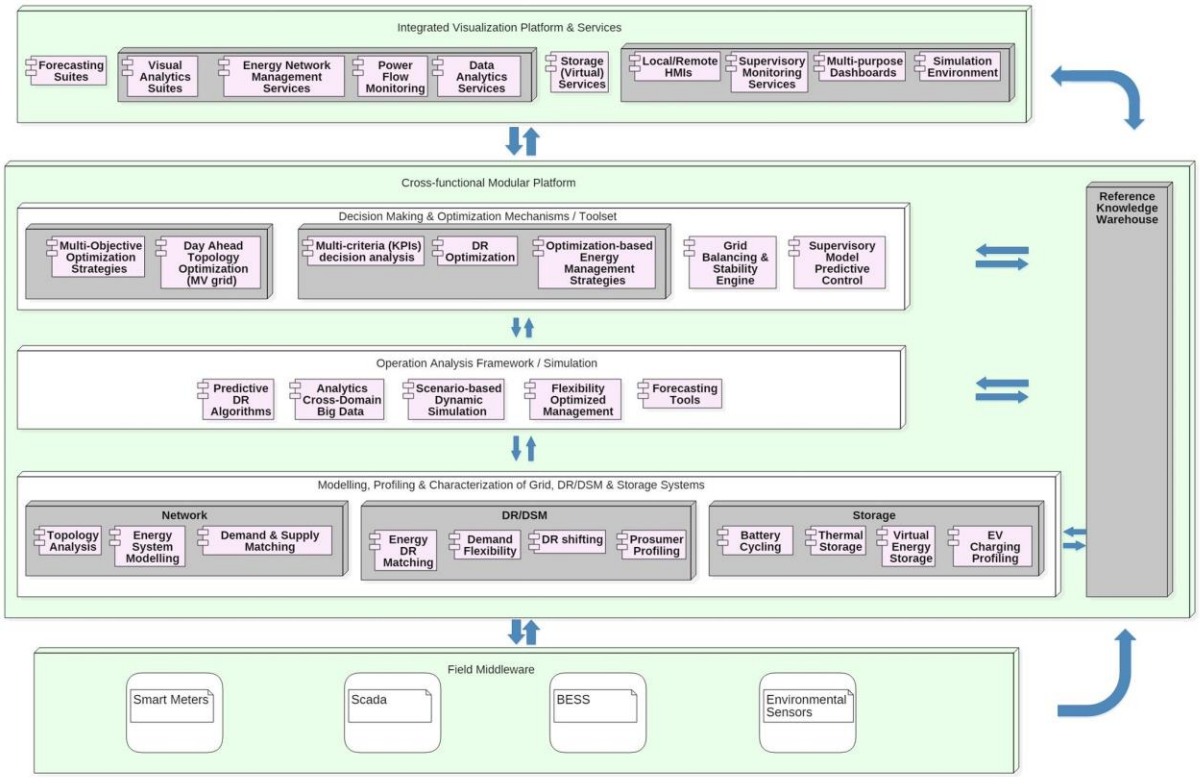


Figure 1. Logical View of inteGRIDy Reference Architecture [IND15]

The reference architecture developed through this WP ensures:

- Each tool is, at least, allocated in one of the architectural layers
- For each layer where the tool is placed, requirements with respect to the way it should interact with other modules inside or outside the layer are imposed
- Each tool is, at least, allocated in one use case
- For each use case where the tool is placed, requirements in terms of the needed functionality and behaviour are included
- Each tool is, at least, allocated in one pilot site
- For each pilot site, concrete Hardware and interconnection details are provided to make sure it can interact with the rest of tools and devices connected

During WP4, all inteGRIDy tools were developed based on this architecture and requirements, but the integration was a theoretical concept that did not become tangible until WP5, in which the pilots were deployed. Based on the deployment plan drafted in D5.2 [IND52] and the guidelines for real integration also described in D5.1 [IND51], this report presents the assessment on the way tools are integrated in the framework.

Consequently, the tools underwent an integration process that was essential to prove the success of the objectives defined for each pilot site in D1.5 and D1.6. Therefore, the objective of this deliverable is to report how well these tools adapted to the original architecture by collecting information about their interfaces with external systems, such as hardware, third-

party software or other inteGRIDy tools and evaluating in the most objective way possible their integration capabilities. Moreover, in order to illustrate the effort put from pilot sites to integrate these tools, they were asked to supply test integration reports that were analysed and used as a baseline in order to give recommendations for the validation tests that are to come in WP8.

1.2 Structure of the deliverable

This deliverable is structured in the following manner:

- Section 1 serves as an introductory point in which the relationship of this deliverable and the rest of the project is brought out,
- Section 2 describes the methodology used for assessing the integration capabilities of each tool and then that assessment is performed by pilot sites,
- Section 3 makes an analysis of the test reports given per each pilot site, making suitable recommendations towards validation tests, and
- Section 4 encompasses the results acquired from sections 2 and 3 and draws conclusions.

1.3 Relation to Other Tasks and Deliverables

The inputs received to produce this deliverable are T1.5, in which the architecture, use cases and functional requirements of each pilot site are defined; T5.1, in which the guidelines for increasing integration capabilities of the tools are outlined; and T5.2, which details the deployment planning in pilot site areas.

This report is also tightly linked to other deliverables in WP5 which are coinciding in delivery time, such as D5.4 (Simulation Environment Prototype) [IND54] and D5.5 (inteGRIDy Integrated Prototype, final version) [IND55]. In both cases, the alignment is achieved through close collaboration and the delivery of the joint demonstrator of both simulation (D5.4) and operational (D5.5) tools.

2. Tools Integration assessment

In order to measure the integration capability, a survey was sent to every pilot site to characterise both the input and the output data of each tool (Table 1 and Table 2).

After receiving the answers, an assessment was performed following the criteria described in Table 3 at data level, communication level and deployment level; giving higher ranks to those tools facilitating integration. The data level and communication level integration assessments were driven by factual data and by the guidelines described in D5.1 [IND51], whereas the deployment level integration assessment was done in an empirical way, giving the highest ranks to those tools that followed the recommendations given in D5.1 about dockerization/containerization, and then higher ranks depending on the difficulty of integrating the corresponding tool in the inteGRIDy integrated demonstrator, described in detail in D5.5 [IND55].

Table 1. Input data integration survey.

Tool name	Input data				
	Type	Data model	Format	From another tool?	Integration API used
Name of the tool	What is the concrete input the tool receives?	What is the data model used as input for the tool?	What is the format in which the input is expected?	Is this information gathered from other inteGRIDy tool? Which one?	How is another tool or system interfacing with the tool to provide its input?
[Example]	[Consumption data, forecast, etc.]	[SAREF, SmartM2M, solution-specific data model]	[JSON, XML, CSV, XLS, Oracle DB]	[Yes/No]. [Name of the other tool]	[WS, Messaging protocols, FTP, Direct connection to database]

Table 2. Output data integration survey.

Tool name	Output data				
	Type	Data model	Format	To another tool?	Integration API used
Name of the tool	What is the concrete output the tool produces?	What is the data model used as output of the tool?	What is the format in which the output is given?	Is this information sent to other inteGRIDy tool? Which one?	How is the tool interfacing with another tool or system provide its output?

[Example]	[Consumption data, forecast, etc.]	[SAREF, SmartM2M, solution-specific data model]	[JSON, XML, CSV, XLS, Oracle DB]	[Yes/No]. [Name of the other tool]	[WS, Messaging protocols, FTP, Direct connection to database]

Table 3. Integration assessment criteria.

	Data level		Communication level	Deployment level
Integration Rank	**Data format**	**Data model**	**Integration API used**	**Deployment scheme**
3	The data format is one of the recommended approaches (JSON, XML or CSV) or a similar well-known data formats depending on the context	The data model used is standard or well-known	The integration API is a recommended approach (Web Services or messaging protocols) or a standard depending on the context	The tool is dockerized
2	The data format is different from the recommended, but still sufficiently known and its output does not connect to any tool from another pilot site	The data model used is customized, but the input or output tools have connectors to communicate between them or do not need connectors	The tool is directly connected to a DB	The tool is deployed in a VM and/or can easily be integrated in the inteGRIDy integrated demonstrator
1	The data format is different from the recommended, but still sufficiently known and its output does not connect to any tool, included the same pilot site	The data model used is customized and it cannot be mapped to match another tool's data model	A different approach is used (File sharing, manual integration)	The tool can be integrated in the inteGRIDy integrated demonstrator, but the process is not straight-forward

0	The data format is not well-known or customized	There is no data model and the data are shared in an unstructured way	The tool can't be integrated	The tool is deployed locally and cannot be integrated in a common environment, nor an API gateway

In the following, a dedicated sub-section is presented per pilot site, as this is the way inteGRIDy proves the **real** integration of tools. It is important to note that the overall integration ability of inteGRIDy tools is proven through the demonstrator on Simulation Environment (D5.4) [IND54] and the joint demonstrator for operational tools (D5.5).

2.1 Isle of Wight

Following are the survey responses for the Isle of Wight pilot site (Table 4 and Table 5). This pilot consists of 4 tools that make sense to integrate—two of which are also shared with other inteGRIDy pilots: The Supervisory Model Predictive Control for Energy Systems and the Integrated Decision & Support Supervisory System, from Xanthi.

Most of the tools have input data types somewhat generic, which gives them a modular perspective that combined with the right integration API approach makes it easy to integrate in other environments, such as the inteGRIDy integrated demonstrator. This aspect is reflected in Table 6, where the majority of the tools have a high deployment level integration score despite not being containerized. It is recommended to consider dockerization and using standard or well-known data models for future upgrades on the pilot site, potentially beyond inteGRIDy lifetime.

Table 4. IoW pilot site integration survey result (input data).

Tool	Input data				
	Type	**Data Model**	**Format**	**From another tool?**	**Integration API Used**
Energy Portfolio for Advanced Building Management System Control (SIEMENS)	- kWh - Time - Price	Custom object-oriented data model	CSV	No	RESTful API
Heat pump remote control (M7)	Control signals	Custom data model	XML	No	Azure Platform
Supervisory Model Predictive Control for Energy Systems MPC (CERTH)	Historical/ predicted weather and load data	Custom data model	- JSON - CSV	Integrated Decision & Support Supervisory System EMS	MQTT

Integrated Decision & Support Supervisory System EMS (CERTH)	- Historical and online data from devices and meters - Historical and predicted weather and load data	Custom data model	- JSON - CSV	Supervisory Model Predictive Control for Energy Systems MPC	MQTT

Table 5. IoW pilot site integration survey result (input data).

Tool	Output data				
	Type	Data Model	Format	To another tool?	Integration API Used
Energy Portfolio for Advanced Building Management System Control (SIEMENS)	- Revenue and cost reduction strategies for GHG - kWh	Custom object-oriented data model	CSV	No	RESTful API
Heat pump remote control (M7)	In house data collection	Custom data model	XML	No	Web Services
Supervisory Model Predictive Control for Energy Systems MPC (CERTH)	Suggested control/ control data (current) for the operation of the energy exchange converters	Custom data model	- JSON - CSV	Integrated Decision & Support Supervisory System EMS	MQTT
Integrated Decision & Support Supervisory System EMS (CERTH)	Suggested coefficients for MPC objective function	Custom data model	- JSON - CSV	Supervisory Model Predictive Control for Energy Systems MPC	MQTT

Table 6. IoW pilot site integration assessment.

	Data level		Communication level	Deployment level
Tool	**Data format**	**Data model**	**Integration API used**	**Deployment scheme**
Energy Portfolio for Advanced Building Management System Control (SIEMENS)	3.0	2.0	3.0	2.0
Heat pump remote control (M7)	3.0	2.0	3.0	2.0
Supervisory Model Predictive Control for Energy Systems MPC (CERTH)	3.0	2.0	3.0	2.0
Integrated Decision & Support Supervisory System EMS (CERTH)	3.0	2.0	3.0	2.0
Pilot score	**3.0**	**2.0**	**3.0**	**2.0**

2.2 Terni

The results for Terni pilot site integration survey for those tools that make sense to integrate are shown in Table 7 and Table 8. By looking at the output and input tools, one can clearly see the pipeline starting with the Flexibility Optimized Management tool, going through the Open-ADR based DR communication Manager, and ending with the Multi-carrier hub Optimisation engine, which takes other additional inputs to complete its task. This intra-pilot interconnection mixed with the full-dockerization of the three tools and the use of well-known data model in the industry (OpenADR profiles) is the reason why these tools have such high integration assessment ranks, as shown in Table 9.

The only suggestion that can be made, if any, is to use a different integration API for the input data that does not involve direct readings to a database to increase the reusability of the tools by lowering the necessary adaptations to be made in case of implementation in different scenarios.

Table 7. Terni pilot site integration survey result (input data).

Tool	**Input data**				
	Type	**Data Model**	**Format**	**From another tool?**	**Integration API Used**
Flexibility Optimized Management tool – EMS (ENG)	-Forecasting data (generation and consumption forecast, flexibility assessment) -Optimised data -User choice	Custom data model	Direct reading from DB	No	JPA libraries to access data of the DB

Multi-carrier hub Optimisation engine (ENG)	- Demand Response signals - Power data (monitoring, historical and costs) - Forecasting data (generation and consumption forecast, flexibility assessment)	OpenADR profiles	- XML for the OpenADR input - N/A for data read directly from the DB	Open-ADR based DR communication Manager	- OpenADR wrapper exposing REST services - JPA libraries to access data of the DB
Open-ADR based DR communication Manager (ENG)	Demand Response signals	OpenADR profiles	XML	Flexibility Optimized Management tool – EMS	Interfaces exposed by the VTN and VEN for the OpenADR communication exchange

Table 8. Terni pilot site integration survey result (output data).

Tool	Output data				
	Type	**Data Model**	**Format**	**To another tool?**	**Integration API Used**
Flexibility Optimized Management tool – EMS (ENG)	Demand Response signals	OpenADR profiles	XML	Open-ADR based DR communication Manager	OpenADR wrapper exposing REST services
Multi-carrier hub Optimisation engine (ENG)	Optimised power data; economic data about DSO rewards	Custom data model	Text file	No	JPA libraries to access data of the DB

Open-ADR based DR communication Manager (ENG)	Demand Response signals	OpenADR profiles	XML	Multi-carrier hub Optimisation engine	Interfaces exposed by the VTN and VEN for the OpenADR communication exchange

Table 9. Terni pilot site integration assessment.

	Data level		Communication level	Deployment level
Tool	**Data format**	**Data model**	**Integration API used**	**Deployment scheme**
Flexibility Optimized Management tool – EMS (ENG)	2.0	2.5	2.5	3.0
Multi-carrier hub Optimisation engine (ENG)	2.5	2.5	2.5	3.0
Open-ADR based DR communication Manager (ENG)	3.0	3.0	3.0	3.0
Pilot score	**2.5**	**2.7**	**2.7**	**3.0**

2.3 San Severino

Table 10 and Table 11 show the integration survey results from San Severino pilot site. A characteristic feature of this pilot site is the looping behaviour between its tools, complementing each other. From the assessment (Table 12), one can see that the integration scores are sufficiently high, not completely exposing the data through an API in the MV Distribution Networks Management Tool due to security purposes (the tool is designed to operate inside the DSO control centre) and the amount of interest not compensating that account. That aside, a suggestion that can be made is to use standard or well-known data models.

Table 10. San Severino pilot site integration survey result (input data).

Tool	Input data				
	Type	**Data Model**	**Format**	**From another tool?**	**Integration API Used**
Zhero Technology (UNE)	- Setpoint P-Q (active/reactive power) and frequency - Economic evaluation of the services provision (useful for ancillary services)	Custom data model	JSON	MV Distribution Networks Management Tool	RESTful API (Through software component)

MV Distribution Networks Management Tool (POLIMI)	- Distribution network model and data - BESS data - Residential metering data - Weather data	Custom data model	- Direct access to Oracle DB - JSON	Zhero Technology	- Direct access to Oracle DB - RESTful API (Through software component)

Table 11. San Severino pilot site integration survey result (output data).

Tool	Output data				
	Type	**Data Model**	**Format**	**To another tool?**	**Integration API Used**
Zhero Technology (UNE)	- Public net Frequency Measure - Public net Active Power Measure - PV production - User Power Measure - Public net Reactive Power Measure - User Reactive Power Measure calculated - Battery State of charge - Battery Bus Voltage - Public net Voltage Measure	Custom Data Model	JSON	MV Distribution Networks Management Tool	RESTful API (Through software component)
MV Distribution Networks Management Tool (POLIMI)	- Optimal grid topology - BESS power set point - Economic evaluation of the services provision (useful for ancillary services)	Custom Data Model	- Direct access to Oracle DB - JSON	Zhero Technology	- Direct access to Oracle DB - RESTful API (Through software component)

Table 12. San Severino pilot site integration assessment.

	Data level		Communication level	Deployment level
Tool	**Data format**	**Data model**	**Integration API used**	**Deployment scheme**
Zhero Technology (UNE)	3.0	2.0	3.0	2.0
MV Distribution Networks Management Tool (POLIMI)	2.5	2.0	2.5	1.5
Pilot score	**2.8**	**2.0**	**2.8**	**1.8**

2.4 Barcelona

Barcelona pilot site uses four tools, including a tool which is also present in Thessaloniki site and one fully dockerized tool: The Intelligent Building Control & Flexibility Prediction-Forecasting, and the Swimming Pool Model tool, respectively. The integration survey results are shown in Table 13 and Table 14.

By simply looking at the data flow resulting from the interconnection between tools in this pilot site, one cannot argue the intrinsic integration capability that these tools have. Therefore, the high ranks obtained (Table 15) are mostly thanks to the use of standard and well-known data models in combination with integration APIs that facilitate integration. A recommendation to this pilot site can be to extend even more the practice of dockerization and the use of standard data models.

It is also important to note that this particular pilot is currently waiting for an amendment to approve the leadership and venue changes, so the full integration is still on hold, and the particularities as described in these tables might vary with the final implementation.

Table 13. Barcelona pilot site integration survey result (input data).

Tool	**Input data**				
	Type	**Data Model**	**Format**	**From another tool?**	**Integration API Used**
Intelligent Building Control & Flexibility Prediction-Forecasting (CERTH)	- Forecasting - Weather - User energy consumption measurements - User energy consumption profile - Energy prices - User Actions-feedback	Custom data model	JSON	- NEMO tool - Swimming Pool Model	REST API

NEMO tool (TEES)	- OpenADR signals	- OpenADR profiles - IEC CIM	XML JSON	- Intelligent Building Control & Flexibility Prediction-Forecasting - Distributed Energy Management System	- REST API - File-based messages - OpenADR
Distributed Energy Management System (SIEMENS)	- DR Events information - Smart meter - Consumption data - Assets status	- OpenADR profiles - IEC CIM	- XML - JSON - Manually entry (user interface) - CSV - IEC 104	- NEMO tool - Swimming Pool Model	- Web service - JMS - File-based messages - OpenADR - IEC 104
Swimming Pool Model (AIGUASOL)	- Forecasting - User energy consumption profile - User Actions-feedback	Custom data model	JSON	Intelligent Building Control & Flexibility Prediction-Forecasting	REST API

Table 14. Barcelona pilot site integration survey result (output data).

Tool	Output data				
	Type	**Data Model**	**Format**	**To another tool?**	**Integration API Used**
Intelligent Building Control & Flexibility Prediction-Forecasting (CERTH)	- BESS dis-/charge schedules - DR schedules - DR point system	Custom data model	JSON	- NEMO tool - Swimming Pool Model	REST API
NEMO tool (TEES)	OpenADR signals	- OpenADR profiles - IEC CIM	XML JSON	- Intelligent Building Control & Flexibility Prediction-Forecasting - Distributed Energy Management System	- REST API - File-based messages - OpenADR

Distributed Energy Management System (SIEMENS)	- OpenADR signals - Assets set points	- OpenADR profiles - IEC CIM	- XML - JSON - CSV - IEC 104	NEMO tool	- Web service - JMS - File-based messages - OpenADR - IEC 104
Swimming Pool Model (AIGUASOL)	- BESS dis-/charge schedules - DR schedules - DR point system	Custom data model	- JOSN - XML	- Intelligent Building Control & Flexibility Prediction-Forecasting - Distributed Energy Management System	- REST API - File-based messages

Table 15. Barcelona pilot site integration assessment.

	Data level		Communication level	Deployment level
Tool	**Data format**	**Data model**	**Integration API used**	**Deployment scheme**
Intelligent Building Control & Flexibility Prediction-Forecasting (CERTH)	3.0	2.0	3.0	2.0
NEMO tool (TEES)	3.0	3.0	3.0	2.0
Distributed Energy Management System (SIEMENS)	3.0	3.0	3.0	2.0
Swimming Pool Model (AIGUASOL)	3.0	2.0	3.0	3.0
Pilot score	**3.0**	**2.5**	**3.0**	**2.3**

2.5 St. Jean

By looking at the results of the integration survey for this pilot site (Table 16 and Table 17), one can see that the integration capability of these tools is good in general. They both have well-defined inputs and outputs with REST web services as integration API at both boundaries of the data flow. This makes it an easy task to accommodate them in the inteGRIDy integrated demonstrator despite the lack of dockerization.

Reflected in the integration assessment (Table 18) are the recommendations that the project gives to this pilot site: to extend even further the use of standard and well-known data models, and to dockerize the tools.

Table 16. St. Jean pilot site integration survey result (input data).

Tool	Input data				
	Type	Data Model	Format	From another tool?	Integration API Used
Demand Side Energy Profiling (TREK)	- User data (unique id, profiles, type of user: prosumer or consumer) - Device data (unique id) - Energy demand - Date intervals	Custom data model	JSON	No	REST Web services
Visualization Analytics Engine (TREK)	- Metrics (Cost, environmental, energy consumption) - User data (unique id) - Device data (unique id) - Date intervals - DR requests	Custom data model	- JSON - XML	No	REST Web services

Table 17. St. Jean pilot site integration survey result (output data).

Tool	Output data				
	Type	Data Model	Format	To another tool?	Integration API Used
Demand Side Energy Profiling (TREK)	- Profile probabilities - Demand flexibility potential - VES potential	Custom data model	JSON	No	REST Web services

Visualization Analytics Engine (TREK)	- Historical data analytics - Asset clusters - Optimised DR strategies / What-if scenarios - DR report	- Custom data model - OpenADR profiles	- JSON - XML	No	REST Web services

Table 18. St. Jean pilot site integration assessment.

	Data level		Communication level	Deployment level
Tool	**Data format**	**Data model**	**Integration API used**	**Deployment scheme**
Demand Side Energy Profiling (TREK)	3.0	2.0	3.0	2.0
Visualization Analytics Engine (TREK)	3.0	2.5	3.0	2.0
Pilot score	**3.0**	**2.3**	**3.0**	**2.0**

2.6 Nicosia

The results for the integration survey for Nicosia pilot site can be seen in Table 19 and Table 20. The simple input and output types combined with the use of recommended data formats and a well-known integration API for the specific context of this pilot site increase the integration capabilities of the tools.

On the other hand, the low ranks obtained in the integration assessment shown in Table 21 are justified by the use of a proprietary platform developed within the context of another European project (GOFLEX). Therefore, the recommendations to this pilot site are to use standard data models and considering dockerization.

Table 19. Nicosia pilot site integration survey result (input data).

Tool	**Input data**				
	Type	**Data Model**	**Format**	**From another tool?**	**Integration API Used**
Demand Response Tool (UCY)	Energy consumption	Custom data model	- CSV - XML	No	MODBUS TCP
Monitoring and Supervision of microgrid Tool (UCY)	Power measurements	Custom data model	- CSV - XML	No	MODBUS TCP

Table 20. Nicosia pilot site integration survey result (output data).

Tool	Output data				
	Type	Data Model	Format	To another tool?	Integration API Used
Demand Response Tool (UCY)	- Binary signals - DR Reports	Custom data model	- CSV - XML	No	MODBUS TCP
Monitoring and Supervision of microgrid Tool (UCY)	- Energy management reports	Custom data model	- CSV - XML - PDF - Pictures	No	NA

Table 21. Nicosia pilot site integration assessment.

	Data level		Communication level	Deployment level
Tool	Data format	Data model	Integration API used	Deployment scheme
Demand Response Tool (UCY)	3.0	2.0	3.0	0.0
Monitoring and Supervision of microgrid Tool (UCY)	3.0	2.0	3.0	2.0
Pilot score	**3.0**	**2.0**	**3.0**	**1.0**

2.7 Lisbon

This pilot site is composed by just one tool with simple inputs and outputs that uses the recommended data formats and has an appropriate integration API to facilitate integration, as shown in the survey results in Table 22 and Table 23.

The high ranks observed in the integration assessment shown in Table 24, apart from the obvious, are due to the possibility of dockerization in some parts of the tool, which we suggest executing. Other suggestion would be to use standard data models.

Table 22. Lisbon pilot site integration survey result (input data).

Tool	Input data				
	Type	Data Model	Format	From another tool?	Integration API Used
Kisense Energy Management System (VPS)	- Weather data and forecast - Energy prices - Consumption data - Generation data	Custom Data Model	- JSON - CSV - XLS	No	- RESTful API - File importer

Table 23. Lisbon pilot site integration survey result (output data).

Tool	Output data				
	Type	Data Model	Format	To another tool?	Integration API Used
Kisense Energy Management System (VPS)	- Raw and aggregated consumption data - Generation and consumption forecasts - Load profiles	Custom Data Model	- JSON - CSV - XLS	No	- RESTful API - File importer

Table 24. Lisbon pilot site integration assessment.

	Data level		Communication level	Deployment level
Tool	Data format	Data model	Integration API used	Deployment scheme
Kisense Energy Management System (VPS)	3.0	2.0	3.0	2.5
Pilot score	**3.0**	**2.0**	**3.0**	**2.5**

2.8 Xanthi

Table 25 and Table 26 show the integration survey results for this pilot site and in Table 27 one can see the integration assessment. The high ranks obtained are not surprising considering that all the tools are reused in other pilots and, therefore, proven their ease of integration in different contexts somehow. Consequently, a proposition to this pilot site could be to maximise even further those capabilities by extending the use of well-known data models and applying dockerization.

Table 25. Xanthi pilot site integration survey result (input data).

Tool	Input data				
	Type	Data Model	Format	From another tool?	Integration API Used
Integrated Decision Support & Supervisory EMS (CERTH)	- Historical and online data from devices and meters - Historical and predicted weather and load data	Custom data model	- JSON - CSV	- Plant / Process / System Data Exchange Tool - Supervisory Model Predictive Control for Energy Systems	MQTT

Supervisory Model Predictive Control for Energy Systems (CERTH)	Historical/ predicted weather and load data	Custom data model	- JSON - CSV	Integrated Decision Support & Supervisory EMS	MQTT
Plant/Process/System Data Exchange Tool (CERTH)	Data from devices and meters. - Voltage - Current - Power - Pressure - Temperature (Production and consumption from PVs, Wind Gens, Batteries and battery cells, loads, DC and AC bus, Grid DC/DC converters, data from Hydrogen infrastructure, FC, Electrolyser, Converters Voltage, Current, Power, H2)	OPC	OPC	No	OPC Data Access

Table 26. Xanthi pilot site integration survey result (output data).

Tool	Output data				
	Type	Data Model	Format	To another tool?	Integration API Used
Integrated Decision Support & Supervisory EMS (CERTH)	Suggested coefficients for MPC objective function	Custom data model	- JSON - CSV	Supervisory Model Predictive Control for Energy Systems	MQTT

Supervisory Model Predictive Control for Energy Systems (CERTH)	Suggested control/ control data (current) for the operation of the energy exchange converters	Custom data model	- JSON - CSV	Integrated Decision Support & Supervisory EMS	MQTT
Plant/Process/System Data Exchange Tool (CERTH)	Data from devices and meters. - Voltage - Current - Power - Pressure - Temperature (Production and consumption from PVs, Wind Gens, Batteries and battery cells, loads, DC and AC bus, Grid DC/DC converters, data from Hydrogen infrastructure, FC, Electrolyser, Converters Voltage, Current, Power, H2)	OPC	JSON	Integrated Decision Support & Supervisory EMS	MQTT

Table 27. Xanthi pilot site integration assessment.

	Data level		Communication level	Deployment level
Tool	**Data format**	**Data model**	**Integration API used**	**Deployment scheme**
Integrated Decision Support & Supervisory EMS (CERTH)	3.0	2.0	3.0	2.0
Supervisory Model Predictive Control for Energy Systems (CERTH)	3.0	2.0	3.0	2.0
Plant/Process/System Data Exchange Tool (CERTH)	3.0	3.0	3.0	2.0
Pilot score	**3.0**	**2.3**	**3.0**	**2.0**

2.9 Ploiesti

Ploiesti pilot site consists of a single tool that provides it with the highest average ranks in the integration assessment (Table 30). One can see the reason for this by looking at the survey results shown in Table 28 and Table 29. The Energy Integrated Information System has clearly defined inputs and outputs and uses the recommended approach for data formats and integration APIs. Furthermore, the tool is fully dockerized, which facilitates even more its integration capabilities. The only recommendation that can be done to this pilot site is to use standard data models.

Table 28. Ploiesti pilot site integration survey result (input data).

Tool	Input data				
	Type	Data Model	Format	From another tool?	Integration API Used
Energy Integrated Information System (SIVECO)	Energy consumption user data	Custom data model	- XML - JSON	No	- MQTTO - REST Services

Table 29. Ploiesti pilot site integration survey result (output data).

Tool	Output data				
	Type	Data Model	Format	To another tool?	Integration API Used
Energy Integrated Information System (SIVECO)	Proposed optimized consumption, forecasting and simulation results	Custom data model	- XML - JSON	No	REST WS

Table 30. Ploiesti pilot site integration assessment.

	Data level		Communication level	Deployment level
Tool	Data format	Data model	Integration API used	Deployment scheme
Energy Integrated Information System (SIVECO)	3.0	2.0	3.0	3.0
Pilot score	**3.0**	**2.0**	**3.0**	**3.0**

2.10 Thessaloniki

The distinctive tool setup for this pilot site makes it interesting to analyse from an integration perspective. From the integration survey results shown in Table 31 and Table 32, one can notice that the data flow involves three tools also used in Xanthi pilot site: Integrated Decision Support & Supervisory, Supervisory Model Predictive Control for Energy Systems and Plant/Process/System Data Exchange Tool. Besides, there is a tool developed for this pilot site that is again used in Barcelona pilot (Intelligent Building Control & Flexibility Prediction-Forecasting). This cross-pilot integration proves somehow the integration capability of these tools. However, it is recommended to use a different Integration API such as RESTful web services for the tools that use direct connection with the database, or a containerized deployment scheme in all the tools to minimise integration adaptations when reusing the tools.

Table 31. Thessaloniki pilot site integration survey result (input data).

Tool	Input data				
	Type	Data Model	Format	From another tool?	Integration API Used
Visual Analytics Tool for Flexibility Analysis, Aggregation and Forecasting (CERTH)	- Energy consumption forecasting - Weather - User energy consumption measurements - BESS dis-/charge schedules - DR schedules - DR point system - Energy prices	Custom data model	JSON	Intelligent Building Control & Flexibility Prediction-Forecasting	Database API (direct access to RKW)
Intelligent Building Control & Flexibility Prediction-Forecasting (CERTH)	- Energy consumption forecasting - Weather - User energy consumption measurements - User energy consumption profile - Energy prices - User Actions-feedback	Custom data model	JSON	- Visual Analytics Tool for Flexibility Analysis, Aggregation and Forecasting - Building Occupancy & Energy Consumption Simulation Tool	- Weather API (Web Service) - Energy price API (Web Service) - Database API (direct access to RKW)

Integrated Decision Support & Supervisory EMS (CERTH)	- Historical and online data from devices and meters - Historical and predicted weather and load data	Custom data model	- JSON - CSV	- Plant/Process/System Data Exchange Tool - Supervisory Model Predictive Control for Energy Systems	MQTT
Building Occupancy & Energy Consumption Simulation Tool (CERTH)	- Energy consumption forecasting - Weather - User energy consumption measurements - Energy prices - User Actions-feedback	Custom data model	JSON	- Visual Analytics Tool for Flexibility Analysis, Aggregation and Forecasting - Intelligent Building Control & Flexibility Prediction-Forecasting	Database API
Supervisory Model Predictive Control for Energy Systems (CERTH)	Historical/ predicted weather and load data	Custom data model	- JSON - CSV	Integrated Decision Support & Supervisory EMS	MQTT

Plant/Process/System Data Exchange Tool (CERTH)	Data from devices and meters. - Voltage - Current - Power - Pressure - Temperature (Production and consumption from PVs, Wind Gens, Batteries and battery cells, loads, DC and AC bus, Grid DC/DC converters, data from Hydrogen infrastructure, FC, Electrolyser, Converters Voltage, Current, Power, H2)	OPC	OPC	No	OPC DA
Facility Management Web-based Interface (CERTH)	- Energy consumption forecasting - Weather - User energy consumption measurements - BESS dis-/charge schedules - DR schedules - Energy prices	Custom data model	JSON	Intelligent Building Control & Flexibility Prediction-Forecasting	Database API (direct access to RKW)

Table 32. Thessaloniki pilot site integration survey result (output data).

Tool	Output data				
	Type	Data Model	Format	To another tool?	Integration API Used
Visual Analytics Tool for Flexibility Analysis, Aggregation and Forecasting (CERTH)	User Action-feedback	Custom data model	JSON	Intelligent Building Control & Flexibility Prediction-Forecasting	Database API

Intelligent Building Control & Flexibility Prediction-Forecasting (CERTH)	- BESS dis-/charge schedules - DR schedules - DR point system	Custom data model	JSON	- Visual Analytics Tool for Flexibility Analysis, Aggregation and Forecasting - Building Occupancy & Energy Consumption Simulation Tool	Database API
Integrated Decision Support & Supervisory EMS (CERTH)	Suggested coefficients for MPC objective function	Custom data model	- JSON - CSV	Supervisory Model Predictive Control for Energy Systems	MQTT
Building Occupancy & Energy Consumption Simulation Tool (CERTH)	User energy consumption profile	Custom data model	JSON	- Visual Analytics Tool for Flexibility Analysis, Aggregation and Forecasting - Intelligent Building Control & Flexibility Prediction-Forecasting	Database API
Supervisory Model Predictive Control for Energy Systems (CERTH)	Suggested control/ control data (current) for the operation of the energy exchange converters	Custom data model	- JSON - CSV	Integrated Decision Support & Supervisory EMS	MQTT

Plant/Process/System Data Exchange Tool (CERTH)	Data from devices and meters. - Voltage - Current - Power - Pressure - Temperature (Production and consumption from PVs, Wind Gens, Batteries and battery cells, loads, DC and AC bus, Grid DC/DC converters, data from Hydrogen infrastructure, FC, Electrolyser, Converters Voltage, Current, Power, H2)	OPC	JSON	Integrated Decision Support & Supervisory EMS	MQTT
Facility Management Web-based Interface (CERTH)	User Action-feedback	Custom data model	JSON	Intelligent Building Control & Flexibility Prediction-Forecasting	Database API

Table 33. Thessaloniki pilot site integration assessment.

	Data level		Communication level	Deployment level
Tool	**Data format**	**Data model**	**Integration API used**	**Deployment scheme**
Visual Analytics Tool for Flexibility Analysis, Aggregation and Forecasting (CERTH)	3.0	2.0	2.0	1.0
Intelligent Building Control & Flexibility Prediction-Forecasting (CERTH)	3.0	2.0	2.0	1.0
Integrated Decision & Support Supervisory System EMS (CERTH)	3.0	2.0	3.0	2.0
Building Occupancy & Energy Consumption Simulation Tool (CERTH)	3.0	2.0	2.0	1.0
Supervisory Model Predictive Control for Energy Systems (CERTH)	3.0	2.0	3.0	2.0

Plant/Process/System Data Exchange Tool (CERTH)	3.0	3.0	3.0	2.0
Facility Management Web-based Interface (CERTH)	3.0	2.0	2.0	2.0
Pilot score	**3.0**	**2.1**	**2.4**	**1.6**

3. Integration tests analysis

Integration tests can be done at the three levels: data level, communication level and deployment level. At data level, one can ensure that the data model is maintained and that there are no missing fields or validate that the format of the output is still correct and readable; at communication level, one can guarantee the interconnection between two components by verifying that they exchange data successfully; and at deployment level one can execute the tool in different environments (e.g., a different version of the operating system) and validate that the tool is still working as intended.

Considering that, and in order to ensure the quality of the integration capabilities that inteGRIDy tools have, each pilot site was asked to conduct tests and provide test reports as a prove, showing that effort was made to detect and correct defects in a timely manner.

After requesting tests reports, it is possible to observe that even though every pilot used the proposed test case template from D1.5 [IND15] (shown in Table 34 for convenience) or similar, each pilot used a different methodology for the integration tests. Therefore, the purpose of this section is to analyse the reported integration tests and, in light of that, evaluate and assess the proof of integration provided. It is also important to note that this report contains the integration tests conducted. That is, those tests aimed at validating that tools are effectively connecting both with the devices/apparatuses on field and other tools/modules they should interact with. Therefore, this report concentrates on the real instantiation of tools and inteGRIDy framework at pilot level. Again, the full integration and interoperability proof, in general terms, not pilot by pilot, is made in D5.5 [IND55]. In addition, further validation tests will be also made in the context of WP8. Those tests will be oriented to validate the behaviour of tools and their adequacy to the requirements and objectives for performance set in the DoA and as detailed in WP1.

This report documents two different approaches: implicit integration testing and explicit integration testing. The former refers to testing a tool being executed in a controlled environment and connected to the corresponding external systems (e.g., physical equipment, a database or another tool), and validating that the output is correct; and the latter refers to explicitly testing the integration by including the verification of the well-functioning of the tool within the environment and the interconnection with external systems separately.

Table 34. Test case design Template

TC		
UC		
FRs		
Precondition		
Test environment (optional)		
TC Step (actions)	**Obtained result**	**Verdict**
1		
2.		
3.		

Please note that in order not to populate this document with tests reports from all pilot sites, these were included in the Annex I. Integration tests

3.1 Isle of Wight

As different partners participated in the development of the tools from this pilot site, the two types of test approaches can be seen in the integration test reports. On the one hand, an explicit testing was conducted for the Energy Portfolio for Advanced Building Management System Control, and on the other hand, implicit testing was conducted for the Heat pump remote control tool. No test reports are needed for the Integrated test environment tool since it is a simulation tool and it does not actually interact with external systems (it runs in standalone mode) and for the Supervisory Model Predictive Control for Energy Systems and Integrated Decision & Support Supervisory System because these tools can run independently as used also in Xanthi pilot site (their integration tests analysis can be found in Section 3.8).

A recommendation that could be done to this pilot site towards the validation tests is to create more test cases and try to increase the test coverage of the tools. Another recommendation is to describe more precisely what is being tested, avoiding generalizations. This will be achieved at WP8 level as appropriate KPIs have been selected and allocated to each tool, use case and requirements.

3.2 Terni

For all the tools in this pilot site explicit integration testing was performed. Every interaction with external systems was validated. Moreover, one can see that tests were clear and concise, making references to inteGRIDy use cases and functional requirements described in D1.5.

Few recommendations can be given to this pilot site regarding validation tests: Only trying to maximise the test coverage by increasing even further the conditions under which each tool is being tested. Again, the allocation of specific KPIs as described in D8.1 [IND81] will ease this task.

3.3 San Severino

As in Terni pilot site, the tests reports show evidence of explicit testing with well-defined test cases and concise test descriptions, also referencing inteGRIDy uses cases and functional requirements.

A suggestion for the validation tests could be to increase the number of test cases to improve test coverage. As in previous cases, the KPI allocation of WP8 will facilitate the validation.

3.4 Barcelona

Despite the variety of partners that work on this pilot site tools, all of them performed explicit integration testing, referring to inteGRIDy use cases in the tools proper of this pilot site. Moreover, tests for integration with the Intelligent Building Control & Flexibility Prediction-Forecasting tool also used in Thessaloniki pilot were executed explicitly in the rest of the tools.

Looking forward to validation test reports to be done in WP8, it is recommended to make the descriptions of individual actions and results more precise and structured such as the ones seen in the Swimming Pool Model tool. The elicitation of KPIs in D8.1 is in line with this recommendation.

3.5 St. Jean

The test reports in this pilot site expose explicit integration testing, with well-defined test descriptions and highlighted relationships between the test cases. A positive aspect of the reports is that it is structured in such a way that same tests done with different scenarios are grouped together, facilitating their evaluation.

A recommendation for the validation tests, if any, is to increase the test coverage, using the KPI environment which is set by WP8.

3.6 Nicosia

For Nicosia pilot site, the test reports show implicit integration testing. However, the tests defined are concise and make the corresponding references to inteGRIDy use cases and functional requirements.

A recommendation to this pilot site towards the validation tests is to conduct more detailed tests and go a step down in abstraction in order to increase the robustness of the tests and the assurance in what they are testing. Also, to increase as much as possible the number of tests cases using the KPIs defined in D8.1.

3.7 Lisbon

Similar to Nicosia pilot site, Lisbon's test reports show implicit integration testing, with concise and clear test descriptions, related with their respective references to inteGRIDy use cases and functional requirements.

Two recommendations for validation testing could be to both increase the granularity of tests and define as much test cases as possible. This task can be performed using the WP8 KPIs as baseline.

3.8 Xanthi

For all the tools in Xanthi pilot site explicit integration tests were conducted. These integration tests were very concise, well defined and clear, and made corresponding references to inteGRIDy use cases and functional requirements.

The only recommendations that can be made towards validation tests is to use binary assertions and remember to maximise the test coverage using the KPI framework set in WP8.

3.9 Ploiesti

The test reports provided for this pilot site show explicit integration testing, with very well defined and exhaustive test cases that refer to inteGRIDy functional requirements. Furthermore, the quality of the tests is boosted by using Selenium, a software testing framework.

Therefore, very little can be suggested to this pilot site, but to keep their approach during the validation tests and to remember the importance of maximising test coverage using WP8 KPIs.

3.10 Thessaloniki

For this pilot site, one can see in the test reports explicit integration testing. The test cases were well defined and concise and made references to inteGRIDy use cases and functional requirements.

Recommendations towards validation tests can be to describe the test environment for each test case instead of using conditionals in the description of the result and using binary assertions. Moreover, it is never too much to remind the importance of maximising test coverage making use of the KPI allocation per pilot as done in D8.1.

4. Conclusions

From an integration point of view, the quantified analysis shown in this deliverable acknowledges that all the pilot sites have good capabilities in general, following most of the good practices and approaches proposed in the inteGRIDy integrated prototype (D5.1) [IND51] and, by extension, to the requirements for the inteGRIDy framework and tool development described in D1.5/D1.6 [IND15, IND16] and WP4 reports [IND42][IND43][IND44].

Figure 2 shows a comparison of the integration aspects quantified per pilot. At a first glance, it is easy to see that except for one pilot site (which has a justified reason), the combination of tools in all the pilots are above 1.5, which can be considered a good starting point towards integration. Indeed, this may be the reason why it is possible to consider an integrated demonstrator, such as the one described in D5.5 [IND55]. Moreover, one can also see in this comparison that the two aspects in which the tools better followed the recommendations are data formats and Integration APIs, pursued by data models and deployment schemes. In consequence, the most recommended approach to increase the average integration capabilities is to apply containerization to as many tools as possible, followed by the use of standard data models.

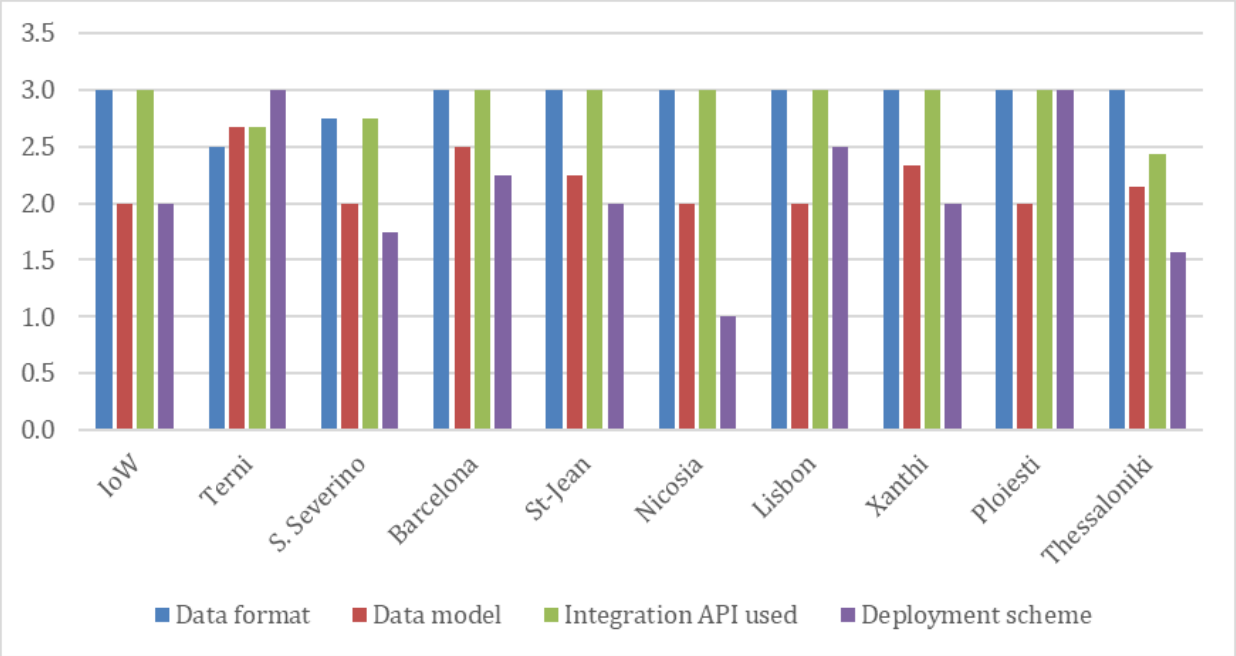


Figure 2. Integration assessment comparison.

It is important to reiterate in this conclusions section that the aim of this report is to assess the **real** integration of the inteGRIDy framework as instantiated in the project pilot sites. This is due to the reason that only by really deploying tools gathering real data and interacting with one another, full integration can be documented.

Nevertheless, being the scope of inteGRIDy developing a framework of interoperable tools, D5.5, as described through this report, will also consider the full interoperability of operational tools, making sure that they can all exchange information at data level and, therefore, interoperate in a single environment. Similarly, D5.4 [IND54] will also prove the interoperability at simulation level for those tools in inteGRIDy.

Regarding the integration tests, after studying the reports, the general recommendation for the testing methodology to be addressed in the validation tests in WP8 is to try to make more atomic tests. In other words, tests that assert the proper operation of a single feature or action instead of grouping together several aspects of the tool in a single test.

It is true that if a system composed of two elements connected one after another is fed with an input and, after analysing and validating the output we find out that everything is right, then that means that the interconnection between the two elements is also right. However, in order to give the validation tests more clarity and robustness, it is proposed instead to perform unitary tests to the first element, then to the interconnection between the elements and finally to the second element. This way it is also possible to test edge cases and validate that even under extreme conditions (but likely to happen) the system will still work as intended. The more tests are performed, the more likely is to successfully integrate a tool in a different environment.

Finally, as also stressed through the text, this report covers the integration tests (proof of real interaction between tools, modules, data sources and devices) performed by pilot sites, while the validation tests (assurance of accuracy of tools as per the requirements set) will be performed under the umbrella of WP8.

5.References

[IND15] inteGRIDy project D1.5 “inteGRIDy Architecture and Functional/Technical Specifications” December 2017. http://www.integridy.eu/sites/default/files/integridy/public/content-files/deliverables/inteGRIDy_D1.5_Architecture_Functional_Technical_Specifications_v1.0.pdf

[IND16] inteGRIDy project D1.6 “inteGRIDy Architecture and Functional/Technical Specifications (Updated)” December 2018. http://www.integridy.eu/sites/default/files/integridy/public/content-files/deliverables/inteGRIDy_D1.6_Architecture_Specifications_v1.0.pdf

[IND42] inteGRIDy project D4.2 “inteGRIDy Modelling Mechanisms” March 2019.

[IND43] inteGRIDy project D4.3 “inteGRIDy Operation Analysis Framework” March 2019

[IND44] inteGRIDy project D4.4 “inteGRIDy Decision Making & Optimization Mechanisms” March 2019

[IND51] inteGRIDy project D5.1 “inteGRIDy Integrated Prototype” March 2019.

[IND52] inteGRIDy project D5.2 “Pilot Areas Deployment Planning” October 2019.

[IND54] inteGRIDy project D5.4 “Simulation Environment Prototype” January 2020.

[IND55] inteGRIDy project D5.5 “inteGRIDy Integrated Prototype, final version” January 2020.

[IND81] inteGRIDy project D8.1 “inteGRIDy Pilot Evaluation Methodology and Framework” December 2019.

Annex I. Integration tests

IoW

Test Case	[001]	[End to control Population of data model, UI Integration and Overall Control of Assets and Data Retrieval]
Related Use Cases	VPP for Advanced Building Management System Control	
Functional Requirements	Where an asset is configured to be used as part of a VPP there is a requirement that measurement data is used instead of metering data. Measurement data will be provided on an ongoing basis via a connection between Siemens Solution and a number of Programmable Logic Controllers (PLC) installed onsite. Communication with the PLC will be performed through the use of IEC 60870-5-104 protocol. This protocol is also used by the VPP functionality to send set-points against which individual assets are expected to follow. Siemens Solution will not attempt to estimate any missing metering or measurement data; it will instead highlight its presence through a report that can be used operational to, where possible, investigate and rectify the reasons behind this.	
Precondition	Siemens Solution implements a specific data model that is needed in order to enable both DR and VPP functionality. This data model will be used to represent the core physical entities on campus, such as meters, DR/VPP assets, RTU's and buildings.	
Test environment (optional)	Offline mode	
Step (actions)	**Obtained result**	**Verdict**
1) Population of data model	Whilst the separate elements within the data model can be created by hand this will be a time-consuming process. As such it is proposed that the data model is populated as completely as possible through the use of an interface called FlexSync.	Passed
2) UI Integration	The data model will be implemented so that all related entities are mapped together appropriately and where available names and identifiers, such as asset numbers, will be included so that users of the Siemens Solution UI can easily correlate information across other platforms.	Passed
3) Control of assets and data retrieval	It enables entities such as premises and meters to be created within Siemens Solution with a minimum amount of intervention and will allow for the automated creation of relationships between entities, such as premises and meters. It does this through the use of structured XML that complies with a specific format specified by EnergyIP. The XML data will be received through an HTTPS Web-service.	Passed

Test Case	[Test ID]	[Test Description]
Related Use Cases	M7 system Monitoring - Prototype for MK2 system	
Functional Requirements	Remote data collection of • Hot store temperature • Cold store temperature • Electricity used by heat pump • Heat delivered by heat pump to thermal store • Heat delivered to each property (in a mini district setting). • Back up heater condition in each property.	
Precondition	Installation of new monitoring system in 6 properties	
Test environment (optional)	Prototype system installed in one property and integrate with 6 property analysis	
Step (actions)	**Obtained result**	**Verdict**
1	Build new control box	Passed
2.	Collate data from 6 properties and perfect data collection process	Passed
3.	Monitor outputs	Passed

Terni

Term	Description
MUCHO	Multi Carrier Hub Optimisation Engine
FOMT	Flexibility Optimised Management Tool
OADR	Open-ADR based DR communication Manager

Test Case	1	Creation of the power profile at the MV connection point	
Related Use Cases	ASM_UCA1: Creation of the power profile at the MV connection point		
Functional Requirements	The file with the power profile is created and stored. The file is sent to the Multi Carrier hub Optimisation Engine. The following functional requirements are therefore tested: ASM_ MUCHO_FR1: The tool shall provide the forecasted data and flexibility capability of the microgrid ASM_ MUCHO_FR2: The tool shall compute the forecasted power profile of the microgrid at the MV connection point. ASM_ FOMT_FR1: The tool shall show the forecasted power profile of the microgrid at the MV connection point and the flexibility limits of the microgrid. ASM_ FOMT_FR2: the tool shall allow to select the KPI related to the desired grid service inside the flexibility range and provide the economic value related to the service. ASM_ FOMT_FR3: The tool shall create the file with the power profile and store it		
Precondition	DSO fully operational in the ASM MV distribution network The metering infrastructure (Power Quality Analysers - Wally) has been set up in order to collect near real time data for each energy devices in the microgrid		
Test environment (optional)			
Step (actions)	**Obtained result**		**Verdict**
1.	MUCHO computes the forecasted power profile of the microgrid at the MV connection point		Passed
2	MUCHO stores the forecasted power profile into the DB		Passed
3	MUCHO computes the flexibility assessment and store it into the DB		Passed
4	MUCHO store the flexibility into the DB		Passed
5	FOMT retrieves the forecasted power profile from the DB and show it to the end user (DSO)		Passed

6	FOMT retrieves the flexibility assessment from the DB and show it to the end user (DSO)	Passed
7	DSO can select the KPI or draws the curve by itself	Passed
8	FOMT creates the power request	Passed
9	FOMT stores the power request into the DB	Passed

Test Case	2	The DSO request is sent to the optimisation platform
Related Use Cases	ASM_UCA1: Creation of the power profile at the MV connection point	
Functional Requirements	The file with the power profile is sent to the Multi Carrier hub Optimisation Engine. The following functional requirements are therefore tested: ASM_ FOMT_FR4: The tool shall send the requested service profile (power profile enriched with the price information) to the optimisation platform ASM_OADR_FR1: The tool shall receive the requested service profile (power profile enriched with the price information) to the optimisation platform	
Precondition	The file with the power profile is created and stored.	
Test environment (optional)		
Step (actions)	**Obtained result**	**Verdict**
1	FOMT initialises the VTN	Passed
2	OADR manages the creation of an openADR Event	Passed
3	OADR starts listen for Reports	Passed

Test Case	3	Optimisation of microgrid power profiles
Related Use Cases	ASM_UCABC1: Day-ahead optimisation of the micro-grid power profiles	
Functional Requirements	The optimisation process is performed and one of the proposed optimised solution is selected by the end user (the microgrid energy manager). The following FRs are therefore tested: ASM_ MUCHO_FR3: The tool shall start from the current status of the microgrid, evaluates the flexibility that could be provided by each energy units of the microgrid ASM_ MUCHO_FR4: The tool shall show the optimised micro grid power profiles ASM_ MUCHO_FR5: The tool shall compute the optimised power profiles ASM_ MUCHO_FR6: The tool shall compute the forecasted data, flexibility capability of the microgrid and retrieve data about energy prices ASM_ MUCHO_FR7: The tool shall compute the flexibility capability of the microgrid, the related incentives and the related power constraints ASM_ MUCHO_FR8: The tool shall retrieve data about energy prices from a dynamic energy prices service (external service) ASM_ MUCHO_FR9: The tool shall retrieve historical data from the Microgrid Energy Unit Monitoring Tool (External Tool) ASM_ MUCHO_FR10: The tool shall compute the forecasted power profile for the addressed energy units ASM_ MUCHO_FR11: The tool shall request the current status of the microgrid from Microgrid Energy Unit Monitoring Tool (External Tool) ASM_ MUCHO_FR12: The tool shall compute the flexibility that can be provided by the addressed energy units of the microgrid ASM_ MUCHO_FR14: The tool shall save the optimised solution to be applied within the microgrid in the data storage and send it to the DSO ASM_ MUCHO_FR15: The tool shall compute the optimised power flow data (power flow at the MV connection point of the optimised solution - set points as output of the optimisation process)	
Precondition	The file with the requested power profile is available.	
Test environment (optional)		

Step (actions)	**Obtained result**	**Verdict**
1.	MUCHO retrieves from the DB data needed for performing the optimisation process. These data are: -Demand Response signals (file with the requested power profiles) -Power data (monitoring, historical and costs) -Forecasting data (generation and consumption forecast, flexibility assessment)	Passed
2	MUCHO runs the optimisation algorithm and the optimised solutions are provided	Passed
3	MUCHO shows the optimises solution to the end user (the microgrid energy manager)	Passed
4	The microgrid energy manager selects one of the optimised solutions	Passed
5	The selected optimised solution (optimised data) is stored in the DB	Passed

Test Case	4	Sending the optimised solution to the DSO	
Related Use Cases	ASM_UCABC1: Day-ahead optimisation of the micro-grid power profiles		
Functional Requirements	The optimised solution is sent to the DSO. The following FRs are therefore tested: ASM_ MUCHO_FR13: The tool shall perform the optimisation process and send the optimised power profiles ASM_OADR_FR2: The tool shall receive the optimised solution (optimised micro grid power profiles)		
Precondition	The optimisation process is performed and the optimised solution is stored in the DB		
Test environment (optional)			
Step (actions)	**Obtained result**		**Verdict**
1.	MUCHO initialises the VTN		Passed
2	OADR manages the creation of an openADR Report		Passed
3	OADR sends the optimised data encapsulated in the Report		Passed

Test Case	5	Report of the optimised micro-grid energy behaviour
Related Use Cases	ASM_UCA2: Report of the optimised micro-grid energy behaviour	
Functional Requirements	The energy behaviour of the microgrid is available and it is showed to the end user (DSO) The following functional requirements are therefore tested: ASM_ FOMT_FR5: The tool shall show the power flow at the MV connection point of the optimised solution (set points as output of the optimisation process) ASM_ FOMT_FR6: The tool shall show the requested power profile connection point (DSO request) ASM_ FOMT_FR7: The tool shall show the effective measure power flow at the MV connection point	
Precondition	DSO fully operational in the ASM MV distribution network The metering infrastructure (Power Quality Analysers - Wally) has been set up in order to collect near real time data for each energy devices in the microgrid Set points as output of the optimisation process are available The file with the requested power profile is available	
Test environment (optional)		
Step (actions)	**Obtained result**	**Verdict**
1.	FOMT shows to the end user (DSO) the following information: - DSO power profile request - the power flow at the MV connection point of the optimised solution - show the effective measure power flow at the MV connection point	Passed
		Passed

Test Case	6	Access to optimised plan for execution
Related Use Cases	ASM_UCABC2: Day-ahead optimisation plan execution	
Functional Requirements	The optimised data, consisting in the optimised energy configuration of the microgrid, are retrieved and accessed for being executed The following functional requirements are therefore tested:	

	ASM_ MUCHO_FR14: The tool shall save the optimised solution to be applied within the microgrid in the data storage and send it to the DSO ASM_ MUCHO_FR16: The tool shall permit to download the optimised solution calculated by the optimisation tool for the microgrid power profiles	
Precondition	DSO fully operational in the ASM MV distribution network All the processes regarding the optimisation have been successfully completed and the results are available.	
Test environment (optional)		
Step (actions)	**Obtained result**	**Verdict**
1.	MUCHO saves a .txt file with the complete timeseries of optimised date	Passed
2	The Microgrid Energy Manager downloads the .txt file and accesses it	Passed

San Severino

Test Case	T_UCA1	Demand Response based on energy price signals	
Related Use Cases	ASS_ UC01		
Functional Requirements	ASS_ MV-DMNT_FR1, ASS_ MV-DMNT_FR2, ASS_ MV-DMNT_FR3, ASS_ZH_FR1, ASS_ZH_FR2, ASS_ZH_FR3		
Precondition	Zhero technology BESS upgraded with smart meter functionalities and communication services		
Test environment (optional)	POLIMI Lab		
Step (actions)	**Obtained result**		**Verdict**
1. emulation of Italian ancillary services market	realistic prices simulated		Passed
2. relevant data retrieval	Public net Frequency Measure, Public net Active Power Measure, PV production, User Power Measure, Public net Reactive Power Measure, User Reactive Power Measure calculated, Battery State of charge, Battery Bus Voltage, Public net Voltage Measure retrieved through Zhero technology RESTful API and stored in the MV-DMNT Oracle Database		Passed

3. Energy Aggregation and set point sending to the Zhero BESS technology	BESS power set-point sent to the Zhero technology	Passed
4. Inform user about energy behaviour	Relevant data are shown to the user through specific GUI	Passed

Test Case	T_UCB1	Real-time estimation of the network status	
Related Use Cases	ASS_ UC02		
Functional Requirements	ASS_ MV-DMNT_FR4, ASS_ MV-DMNT_FR5, ASS_ MV-DMNT_FR6		
Precondition	DSO acquires grid information (voltage levels, currents, etc.) thanks to a dedicated monitoring infrastructure on the MV grid (linked with the SCADA)		
Test environment (optional)	POLIMI Lab, ASSEM control centre		
Step (actions)	**Obtained result**		**Verdict**
1. SCADA data collection	- Functionality to collect data in place - Communication channel between SCADA/DMS and inteGRIDy workstation available		Passed (functionality and communication channel available and operating in standalone mode. To be tested in the real condition)
2. MV grid "State Estimation problem" solving	Functionality to solve MV grid		Passed
3. Storage of the output from the resolution of the "State Estimation problem"	Relevant data stored in the MV-DMNT Oracle Database		Passed

Test Case	T_UCB2	Forecast of loads consumption and generators' injection	
Related Use Cases	ASS_ UC02		
Functional Requirements	ASS_ MV-DMNT_FR7, ASS_ MV-DMNT_FR8, ASS_ MV-DMNT_FR9, ASS_ MV-DMNT_FR10		
Precondition	Weather forecast service available		
Test environment (optional)	POLIMI Lab		
Step (actions)	**Obtained result**		**Verdict**
1. Weather forecast data acquisition	Weather forecast data acquired from the external service		Passed
2. Historical Data acquisition	Capacity of the MV-DMNT to retrieve historical data already stored in the MV-DMNT Oracle Database Oracle Database		Passed
3. Estimate forecast power profile for loads and generators	Forecasting of power profile for loads and generators		Passed
3. Storage of estimated forecast power profile	Estimated forecast power profile stored in the MV-DMNT Oracle Database		Passed

Test Case	T_UCB3	MV distribution grid optimization	
Related Use Cases	ASS_ UC03		
Functional Requirements	ASS_ MV-DMNT_FR11, ASS_ MV-DMNT_FR12, ASS_ MV-DMNT_FR13		
Precondition	Grid behaviour modelled through "Grid State Estimation" procedure		
Test environment (optional)	POLIMI Lab, ASSEM control centre		
Step (actions)	**Obtained result**		**Verdict**
1. Optimal grid topology identification	Optimal grid topology identified		Passed
2. Storage of optimal grid topology identified	Optimal grid stored in the MV-DMNT Oracle Database		Passed

3. Notification of the optimal grid topology configuration to the DSO	MV-DMNT is able to notify the optimal grid topology to the DSO	Passed (functionality available and operating in standalone mode. To be tested in the real condition)

Test Case	T_UCC1	Energy Storage for DSO oriented Ancillary Services Provision
Related Use Cases	ASS_UC03	
Functional Requirements	ASS_ MV-DMNT_FR14, ASS_ MV-DMNT_FR15, ASS_ MV-DMNT_FR16, ASS_ZH_FR4	
Precondition	Both measures on the distribution grid and in the users' point of common coupling available	
Test environment (optional)	POLIMI Lab	
Step (actions)	Obtained result	Verdict
1. Check if there is a violation in the grid	MV-DMNT is able to detect violation in the grid	Passed
2. Elaboration of power set point	MV-DMNT elaborates power set point for Zhero technology BESS	Passed
3. Sending of request for Zhero technology BESSs regulation	MV-DMNT is able to send regulation to the BESSs by using Zhero technology RESTful API	Passed
4. Zhero technology perform frequency and voltage regulation	Zhero technology set the set-point as received by the MV-DMNT through the technology RESTful API	Passed

Test Case	T_UCC2	Energy Storage for behind the meter services
Related Use Cases	ASS_UC03	
Functional Requirements	ASS_ZH_FR5, ASS_ZH_FR6, ASS_ZH_FR7	

Precondition	Zhero technology BESS available	
Test environment (optional)	POLIMI Lab	
Step (actions)	**Obtained result**	**Verdict**
1. Zhero technology monitor the energy flows	Zhero Technology is able to monitor the energy flows and to show through the GUI the useful information to improve the user awareness	Passed
2.Zhero technology drive the battery charge/discharge in autonomous way or on the base of the optimum power set points coming from the MV-DMNT	Zhero Technology is able to drive battery charge/discharge also on the base of the feedback coming from the MV-DMNT	Passed

Barcelona

Test Case	1	Swimming pool initial conditions and Optimized set points	
Related Use Cases	BCN_UCA1 (Demand Response) Optimization of swimming pool control		
Functional Requirements	CERTH's Intelligent Building Control & Flexibility Prediction-Forecasting tool asks for swimming pool initial conditions from NEMO. NEMO asks and receives the swimming pool initial conditions from the SCADA system via MODBUS protocol and send it to CERTH's Intelligent Building Control & Flexibility Prediction-Forecasting tool. CERTH's Intelligent Building Control & Flexibility Prediction-Forecasting tool uses the information for optimal configuration of swimming pool and sends the optimized setpoints to NEMO, which then forwards it to the SCADA system.		
Precondition	SCADA MODBUS needs to be in place and configured.		
Test environment (optional)	Serial communication via RS232 to send and retrieve information between two computers.		
Step (actions)	**Obtained result**		**Verdict**
1	Opening the serial port communication via RS232		Passed
2.	Setting serial port properties		Passed
3.	Reading and writing data		Passed

Test Case	2	Swimming pool initial conditions	
Related Use Cases	BCN_UCA2 (Demand Response) Usage of swimming pool as thermal storage		
Functional Requirements	CERTH's Intelligent Building Control & Flexibility Prediction-Forecasting tool asks for swimming pool initial conditions from NEMO. NEMO asks and receives the swimming pool initial conditions from the SCADA system via Modbus protocol and send it to CERTH's Intelligent Building Control & Flexibility Prediction-Forecasting tool.		
Precondition	SCADA MODBUS needs to be in place and configured.		
Test environment (optional)	Serial communication via RS232 to send and retrieve information between two computers.		
Step (actions)	**Obtained result**		**Verdict**
1	Opening the serial port communication via RS232.		Passed
2.	Setting serial port properties.		Passed
3.	Reading and writing data.		Passed

Test Case	1	Smart meters consumption data collection. This data is then processed and stored for access by downstream consumers such as the utility, billing systems, or other data transfer services.
Related Use Cases	BCN_UCB2_Smartening the distribution grid: service to the grid	
Functional Requirements	Meter read of service points are compared against the baseline calculated for the service point during demand response events to measure the demand response event performance and settlement.	
Precondition	The universal AMI adapter (UAA) installed and correctly configured in DEMS.	
Test environment (optional)	None	
Step (actions)	**Obtained result**	**Verdict**
1	Preparation of xml files with the smart meter reads example in the appropriate format for the UUA interface.	Passed
2.	Data collection and files processing in DEMS	Passed
3.	Verification of data availability on the platform via the user interface	Passed

Test Case	2	OpenADR integration test
Related Use Cases	BCN_UCB2_Smartening the distribution grid: service to the grid	
Functional Requirements	The OpenADR test integration will support the control between various devices in the Sport Centre and DEMS. We will use the OpenADR communication protocol to send information and signals to cause electrical power-using devices to be turned off or to reduce its consumption during periods of high demand or according to the rules of the Market Program in which they are enrolled.	
Precondition	There must be one Virtual End Nodes (VEN) that support Pull or Push mode of OpenADR profile 2.0b and obey the relevant OpenADR profile 2.0b VEN conformance rules. Since the VEN development (part of the NEMO tool) is not yet installed and operational, tests have been performed using an open source VEN simulator developed by the Electric Power Research Institute (EPRI) VENs are treated as a special type of gateway device (DR Gateway) and are assigned a unique gateway ID and an SSL fingerprint. Thus, to register a VEN in DEMS, it must be registered as a DR Gateway. An account must be created before enabling the OpenADR Adapter.	

	A premise must exist in the system, indicating the location of the gateway. OpenADR adapter correctly installed and configured within the platform.	
Test environment (optional)	None	
Step (actions)	**Obtained result**	**Verdict**
1	Validate OpenADR Adapter Installation in DEMS	Passed
2.	DEMS integration with OpenADR Virtual End Node. This step consists of the data configuration required to enable an OpenADR Virtual End Node (VEN) to register itself with the OpenADR VTN in DEMS application, register the reports it can send, request, and respond to DEMS notification or load-control events, and stream telemetry data.	Passed
3.	Monitor the VEN communication status in DEMS.	Passed
4.	Request an Event for the VEN. Schedule an event from DEMS to the VEN and check that the event information is received satisfactorily in the VEN.	Passed

Test Case	[Test 01]	[Load configuration data]	
Related Use Cases	BCN_UCA1, BCN_UCA2&BCN_UCB2		
Functional Requirements	TRNSYS		
Precondition	None		
Test environment (optional)			
Step (actions)	**Obtained result**		**Verdict**
1. Load parameters	Parameters required, such as timestep duration and URL addresses for communication were loaded from file		Passed

2. Load assets properties	JSON with properties of the controllable assets was parsed	Passed
3. Load occupancy patterns	JSON with occupancy data per hour and weekday from the sports centre was parsed	Passed

Test Case	[Test 02]	[Retrieve required input data]	
Related Use Cases	BCN_UCA1		
Functional Requirements	BCN_IBC&FPF_FR1		
Precondition	None		
Test environment (optional)			
Step (actions)	**Obtained result**		**Verdict**
1. Communication with the Repository to get weather forecasting data	Weather forecasting data for the next 24-hours were retrieved and loaded, per each timestep		Passed after corrections
2. Communication with Esios service to retrieve dynamic electricity pricing	Dynamic electricity pricing data were retrieved in JSON format for the requested time period		Passed after corrections
3. Prepare dynamic electricity pricing data	Retrieved values were matched to each timestep (15-min interval)		Passed after corrections
4. Prepare occupancy data	Estimated occupancy value matched to each timestep (15-min interval)		Passed after corrections

Test Case	[Test 03]	[Calculate new setpoints]
Related Use Cases	BCN_UCA1	
Functional Requirements	BCN_IBC&FPF_FR1, BCN_IBC&FPF_FR2, BCN_IBC&FPF_FR3	
Precondition	Input data have been retrieved	
Test environment (optional)		

Step (actions)	**Obtained result**	**Verdict**
1. Data check	All needed data have been passed successfully to the optimisation layer (DSS engine)	Passed
2. Set value of each setpoint per controllable asset for each 15-min interval (96 intervals)	Setpoint values of controllable assets have been revised	Passed after corrections

Test Case	[Test 01]	Swimming pool model validation	
Related Use Cases	BCN_UCA1, BCN_UCA2&BCN_UCB2		
Functional Requirements	Swimming pool model		
Precondition	None		
Test environment (optional)			
Step (actions)	**Obtained result**		**Verdict**
1. Load parameters	Parameters required recovered from site visits and equipment catalogue data		Passed
2. Load monitoring data	Recovery of monitored data of electrical and thermal consumption of the swimming pool HVAC equipment and on-site meteorological data on 10 minutes timestep for 12 months.		Passed
3. Model calibration	Iterative simulation of the model to calibrate system operational parameters. Electrical consumption error of 1 %, thermal consumption error 8 % (for the monitored period).		Passed

Test Case	[Test 02]	Web API swimming pool application	
Related Use Cases	BCN_UCA1, BCN_UCA2&BCN_UCB2		
Functional Requirements	Swimming pool model		
Precondition	None		
Test environment (optional)	Docker container		
Step (actions)	**Obtained result**		**Verdict**
1. Create basic requests	Postman Collection of the different API requests.		Passed

2. Run requests	Standard requests for every functionality of the Web API application: login, Forecast and Debug mode.	Passed
3. Results review	Validation of the results obtained, including those generated by the TRNSYS calculation engine.	Passed

	[Test 03]	Stress test of the Web API swimming pool application	
Related Use Cases	BCN_UCA1, BCN_UCA2&BCN_UCB2		
Functional Requirements	Swimming pool model		
Precondition	None		
Test environment (optional)	Docker container		
Step (actions)	**Obtained result**		**Verdict**
1. Set data sampling	Establish representative data sampling to be used for stress test.		Passed
2. Collection run	Inside a docker container, we ran the API and we launch a set of 10000 calls to the forecast functionality. No problems were detected, no residual files were found, and a very stable behaviour were found during all along the 17 hours test. The mean time for simulations was 6 seconds.		Passed

Test Case	**TC_DSEP_01**	Evaluation of the ability of the of the DSEP tool to generate prosumer profiles based on collected data
Related Test Cases	N/A	
Functional Requirements	INN_DSEP_FR1, INN_DSEP_FR4	
Precondition	Internet connection availability. Historical data availability.	
Test environment (optional)	-	
Test scenario 1: Thermal Comfort Profile		
Step (actions)	**Obtained result**	**Verdict**
1. Log in the database	-	Passed
2. Check availability of environmental data (indoor temperature, humidity), occupancy status, HVAC operation status/mode/registered control actions	-	Passed
3. Extract data for a pre-defined period under the conditions that the space has been occupied and the occupants have interacted with the HVAC system	Extraction of data with duration of 2 weeks	Passed
4. Generate the thermal comfort profile for the selected data set	Thermal comfort profile through the generation of a pair of comfort/discomfort probability functions	Passed
5. Evaluate the generated profile against the temperature and the registered user control actions in the room during the training period	A good agreement between the derived thermal comfort profile with the registered control actions to the HVAC has been observed. The requested temperature in the room has been maintained within the defined thermal comfort boundaries.	Passed
Test scenario 2: Visual Comfort Profile		
Step (actions)	**Obtained result**	**Verdict**
1. Log in the database	-	Passed

2. Check availability of environmental data (indoor luminance), occupancy status, lighting system dimming level/registered control actions	-	Passed
3. Extract data for a pre-defined period under the conditions that the space has been occupied and the occupants have interacted with the lighting system	Extraction of data with duration of 1 week	Passed
4. Generate the visual comfort profile for the selected data set	Visual comfort profile through the generation of a pair of comfort/discomfort probability functions	Passed
5. Evaluate the generated profile against the measured luminance in the room and the registered user control actions	A good agreement between the derived visual comfort profile with the registered control actions to adjust the dim level of the lighting system has been observed. The requested luminance in the room has been maintained above the defined luminance boundaries.	Passed
Test scenario 3: Domestic Hot Water (DHW) Demand Profile		
Step (actions)	**Obtained result**	**Verdict**
1. Log in the database	-	Passed
2. Check data availability of occupancy status, water heater operational status/registered control actions	-	Passed
3. Extract data for a pre-defined period under the conditions that the space has been occupied and the occupants have interacted with the DHW system	Extraction of data with duration of 2 weeks	Passed
4. Generate the DHW demand profile for the selected data set	DHW demand profile	Passed
5. Evaluate the generated profile against the actual	The generated demand profile exhibits a good agreement against the actual user actions - verifying the assumption of user patterns	Passed

registered control actions		

Test Case	**TC_DSEP_02**	Evaluation of the ability of the of the DSEP tool to periodically update the generated prosumer profiles
Related Test Cases	TC_DSEP_01	
Functional Requirements	INN_DSEP_FR2, INN_DSEP_FR3, INN_DSEP_FR6	
Precondition	Internet connection availability. Generated prosumer profiles. Historical data availability.	
Test environment (optional)	-	
Test scenario 1: Thermal Comfort Profile		
Step (actions)	**Obtained result**	**Verdict**
1. Log in the database	-	Passed
2. Check availability of environmental data (indoor temperature, humidity), occupancy status, HVAC operation status/mode/registered control actions for a period different than the initial training period	-	Passed
3. Enable the DSEP tool to update the thermal comfort profile on new events	Extraction of data for the updating period. Defined updating period of 1 week	Passed
4. Generate the updated thermal comfort profile for the selected period	Updated thermal comfort profile	Passed
5. Evaluate the updated profile against the initial profile and the temperature & registered user control actions in the room during the update period	-	The thermal comfort profile has been successfully updated in line with the registered user actions during the defined period.

Test scenario 2: Visual Comfort Profile		
Step (actions)	**Obtained result**	**Verdict**
1. Log in the database	-	Passed
2. Check availability of environmental data (indoor luminance), occupancy status, lighting system dimming level/registered control actions	-	Passed
3. Enable the DSEP tool to update the visual comfort profile on new events	Extraction of data for the updating period. Defined updating period of 1 week	Passed
4. Generate the updated visual comfort profile for the selected period	Updated visual comfort profile	Passed
5. Evaluate the updated profile against the initial profile and the luminance & registered user control actions in the room during the update period	-	The visual comfort profile has been successfully updated in agreement with the registered user actions during the defined period.

Test scenario 3: Domestic Hot Water (DHW) Demand Profile		
Step (actions)	**Obtained result**	**Verdict**
1. Log in the database	-	Passed
2. Check data availability of occupancy status, water heater operational status/registered control actions	-	Passed
3. Enable the DSEP tool to update the DHW demand profile on new events	Extraction of data for the updating period. Defined updating period of 1 week	Passed
4. Generate the updated DHW demand profile for the selected period	Updated DHW demand profile	Passed

5. Evaluate the updated profile against the initial profile and the registered user control actions during the update period	-	The DHW demand profile has been successfully updated in line with the registered user actions during the defined period.

Test Case	**TC_DSEP_03**	Evaluation of the ability of the DSEP tool to calculate human-centric demand flexibility
Related Test Cases	TC_DSEP_01	
Functional Requirements	INN_DSEP_FR2, INN_DSEP_FR5, INN_DSEP_FR9, INN_DSEP_FR14	
Precondition	Internet connection availability. Generated prosumer profiles. Historical data availability.	
Test environment (optional)	-	

Test scenario 1: Lighting system flexibility

Step (actions)	**Obtained result**	**Verdict**
1. Log in the database	-	Passed
2. Define the time of requested demand flexibility and a predefined test group of prosumers	-	Passed
3.Check availability of environmental data (indoor luminance), occupancy status, lighting system dimming level/registered control actions for the defined time	-	Passed
4. Extract the updated visual comfort profile for the selected period	Visual comfort profiles assumed correct based on the results of TC_DSEP_01	Passed
5. Extract demand flexibility for the selected period in line with the current indoor luminance levels and the visual comfort boundaries as determined by the visual comfort profile	Demand flexibility related to the lighting system	Passed

6. Evaluate the demand flexibility for the selected period based on the current luminance value, visual comfort boundaries and lighting system dim level	• Defined demand flexibility in line with the current dim level • Luminance level occurring after the proposed demand flexibility adjustment - over the lower boundary as defined from the user's visual comfort profile.	The demand flexibility of the lighting system complies with the comfort restrictions of the users; therefore, the human-centric requirement is fulfilled
Test scenario 2: HVAC flexibility		
Step (actions)	**Obtained result**	**Verdict**
1. Log in the database	-	Passed
2. Define the time of requested demand flexibility and a predefined test group of prosumers	-	Passed
3. Check availability of environmental data (indoor temperature, humidity), occupancy status, HVAC operation status/mode/registered control actions for a period different than the initial period	-	Passed
4. Extract the updated thermal comfort profile for the selected period	Thermal comfort profiles assumed correct based on the results of TC_DSEP_01	Passed
5. Extract demand flexibility for the selected period based on the current indoor temperature, thermal comfort boundaries and HVAC operational status	Demand flexibility related to the HVAC system	Passed
6. Evaluate the human-centric demand flexibility by inserting it in the building thermal model and evaluate whether the indoor temperature remains within the thermal comfort boundaries during the defined period	• Defined demand flexibility in line with the HVAC operation status • Temperature evolution within the room for the defined demand flexibility and durations remains within the thermal comfort boundaries as defined from the user's thermal comfort profile.	The demand flexibility of the HVAC system complies with the comfort restrictions of the users; therefore, the human-centric requirement is fulfilled

Test scenario 3: Virtual Energy Storage potential		
Step (actions)	**Obtained result**	**Verdict**
1. Log in the database	-	Passed
2. Define the time of requested virtual energy storage potential and a predefined test group of prosumers	-	Passed
3. Check data availability of occupancy status, water heater operational status, room temperature/HVAC operational status	-	Passed
4. Extract the updated DHW demand profile and thermal comfort profile for the selected period	DHW demand profile assumed correct based on the results of TC_DSEP_01	Passed
5. Extract virtual energy storage potential in total (for DHW and building envelop)	VES potential	Passed
5. Evaluate virtual energy storage potential in total (for DHW and building envelop)	• Defined demand flexibility of DHW in line with the DHW demand profile • Temperature evolution within the room for the defined demand flexibility and durations remains within the thermal comfort boundaries as defined from the user's thermal comfort profile.	The extracted VES potential complies with the thermal comfort profile of the user and with the DHW demand profile

Test Case	**TC_DSEP_04**	Evaluation of the ability of the DSEP tool to periodically update human-centric demand flexibility
Related Test Cases	TC_DSEP_2, TC_DSEP_3	
Functional Requirements	INN_DSEP_FR6, INN_DSEP_FR7, INN_DSEP_FR8	
Precondition	Internet connection availability. Generated demand flexibility profiles. Historical data availability.	
Test environment (optional)	-	

Test scenario 1: Lighting system flexibility		
Step (actions)	**Obtained result**	**Verdict**
1. Log in the database	-	Passed
2. Select a period of model training and requested demand flexibility for a specific interval of time and a predefined test group of prosumers	-	Passed
3.Check availability of environmental data (indoor luminance), occupancy status, lighting system dimming level/registered control actions for the selected period	-	Passed
4. Extract the updated visual comfort profile for the selected period	Updated visual comfort profile assumed correct based on the results of TC_DSEP_02	Passed
5. Extract the demand flexibility for the selected time interval	Updated demand flexibility related to electric lights	Passed
6. Assess whether the extracted human-centric demand flexibility has been updated according to the updated visual comfort profile and the contextual information at the time of the request	• As in TC_DSEP_03 the defined demand flexibility was in line with the updated visual comfort profile • The extracted demand flexibility was updated in relation to the demand flexibility extracted from TC_DSEP_03 and was in line with the contextual information from the prosumer premises	It was concluded that the DSEP tool successfully updated the human-centric demand flexibility of the lighting system periodically.
Test scenario 2: HVAC flexibility		
Step (actions)	**Obtained result**	**Verdict**
1. Log in the database	-	Passed
2. Select a period of model training and requested demand flexibility for a specific interval of time and a predefined test group of prosumers	-	Passed

3. Check availability of environmental data (indoor temperature, humidity), occupancy status, HVAC operation status/mode/registered control actions for the selected period	-	Passed
4. Extract the updated thermal comfort profile for the selected period	Updated thermal comfort profile assumed correct based on the results of TC_DSEP_02	Passed
5. Extract the demand flexibility for the selected time interval	Updated demand flexibility of HVAC system	Passed
6. Assess whether the extracted human-centric demand flexibility has been updated according to the updated thermal comfort profile and given the contextual information at the time of the request	• As in TC_DSEP_03 the defined demand flexibility was in line with the updated thermal comfort profile • The extracted demand flexibility was updated in relation to the demand flexibility extracted from TC_DSEP_03 and in line with the contextual information from the prosumer premises	It was concluded that the DSEP tool successfully updated the human-centric demand flexibility of the HVAC system periodically.
Test scenario 3: Virtual Energy Storage flexibility		
Step (actions)	**Obtained result**	**Verdict**
1. Log in the database	-	Passed
2. Select a period of model training and a specific interval of time and a predefined test group of prosumers to evaluate VES potential	-	Passed
3. Check data availability of occupancy status, water heater operational status, room temperature/HVAC operational status for the selected period	-	Passed
4. Extract the updated DHW demand profile and thermal comfort profile for the selected period	Updated thermal comfort profile assumed correct based on the results of TC_DSEP_02 Updated DHW demand profile assumed correct based on the results of TC_DSEP_02	Passed

5. Extract virtual energy storage potential in total (for DHW and building envelop)	Updated VES potential	Passed
6. Assess whether the extracted human centric VES potential has been updated according to the updated thermal comfort profile and given the contextual information at the time of the request	• As in TC_DSEP_03 the defined VES potential referring to the building envelop was in line with the updated thermal comfort profile • The extracted VES potential referring to DHW was updated according to the updated DHW demand profile of the prosumers	It was concluded that the DSEP tool successfully updated the VES potential periodically

Test Case	**TC_DSEP_05**	Evaluation of the DSEP tool's capability of handling exceptions referring to lack of data or lack of communication.
Related Test Cases	N/A	
Functional Requirements	INN_DSEP_FR15	
Precondition	N/A	
Test environment (optional)	-	
Step (actions)	**Obtained result**	**Verdict**
1. Log in the database	-	Passed
2. Select a time period with missing data	2 hours	Passed
3. Run the fill-up algorithm for the selected period	Note: The missing information refers to indoor temperature and illuminance	Passed
4. Select and plot the same period to evaluate the results of the fill-up algorithm	The algorithm has successfully filled out the missing values.	The DSEP deals satisfactorily with the missing values

Test Case	**TC_VAE_01**	Evaluation of the VAE tool's capability to correctly evaluate the available DR potential and to define optimized strategies based on the energy demand flexibility profiles extracted through the DSEP tool
Related Test Cases	TC_DSEP_02	
Functional Requirements	INN_DSEP_FR10, INN_DSEP_FR12, INN_VAE_FR1, INN_VAE_FR2	

Precondition	Availability of updated demand flexibility profiles from the DSEP tool. Availability of DR reports from previously implemented campaigns. Availability of electricity tariff information from the DSO. Definition of synthetic DR scenarios.	
Test environment (optional)	-	
Step (actions)	Obtained result	Verdict
1. Log in the VAE interface	-	Passed
2. Navigate to the what-if analysis screen	-	Passed
3. Define the required flexibility and DR duration according to the synthetic scenarios.	For testing purposes, the required flexibility was set to different values between 10-20kWh and the DR duration to 10 minutes with a limited asset availability.	Passed
4. Evaluate the extracted DR strategies against the synthetic DR scenarios	Optimised strategies have been offered for each required flexibility defined.	Each time, the DR strategies extracted from the VAE interface agree with the anticipated results of the independent asset analysis based on their demand flexibility. However, the effect of rating parameters of the optimisation algorithm, such as user participation, will be tested during the pilot testing period.

Test Case	**TC_VAE_02**	Evaluation of the capability of the dispatch DR functionality of the VAE interface to send correct and timely requests and confirmation that for each request, a DR report is received back.
Related Test Cases	N/A	
Functional Requirements	INN_DSEP_FR11, INN_DSEP_FR13 (It should be noted that the DR dispatch functionality is implemented in the VAE tool and not in the DSEP tool, therefore the corresponding functional requirements should be corrected.)	
Precondition	Internet connection availability. Establishment of communication with the field middleware. Smart actuators installed at the prosumer premises.	
Test environment (optional)	-	

Step (actions)	**Obtained result**	**Verdict**
1. Log in to VAE	-	Passed
2. Navigate to the What-if analysis screen	-	Passed
3. Define the required flexibility and DR duration	For testing purposes, the required flexibility was set to 1kWh and the DR duration to 10 minutes.	Passed
4. Select the predefined asset to dispatch DR and implement the control action	A single asset has been selected for the despatch DR functionality testing	Passed
5. Evaluate the resulting action at the test asset premises	The registered actions agree with the dispatched DR signals. The signals where implemented in almost real time.	Passed
6. Evaluate the report received after the DR request	The received DR reports following the DR request appropriately represent the DR strategy.	Passed

Test Case	**TC_VAE_03**	Evaluation of the VAE visualisations as to their content and their ability to dynamically update the presented information to the user defined parameters and filters.
Related Test Cases	N/A	
Functional Requirements	INN_VAE_FR3, INN_VAE_FR4, INN_VAE_FR5, INN_VAE_FR6, INN_VAE_FR7, INN_VAE_FR13, INN_VAE_FR14, INN_VAE_FR15	
Precondition	Availability of historical data	
Test environment (optional)	-	
Step (actions)	**Obtained result**	**Verdict**
1. Log in to VAE	-	Passed
2. Define a time period	-	Passed
3. Select an asset on the map to visualise its performance information	-	Passed
4 Evaluate the visualisation of this asset	The visualisations were correct as to the data representation of the selected asset and applied filters.	Passed

5. Select each available KPI to navigate to the corresponding time series visualisations	-	Passed
6. Evaluate the visualisations of each KPI	The VAE interface interactively navigated the user to the selected KPIs (energy consumption at peak hours, CO2 emissions, energy costs). The offered visualisations were in line with the underlying data for the selected asset and defined filters.	Passed
7. Repeat the process for all different filters offered	The testing has been repeated for a number of different assets, every time changing the time periods and available visualisation, such as KPIs, outlier detection, timeseries. Minor issues have been identified and corrected in order to improve the visualisation aesthetics and intuitiveness	Passed

Test Case	**TC_VAE_04**	Evaluation of the effectiveness with which the VAE back-end component manages and processes the incoming data streams.
Related Test Cases	N/A	
Functional Requirements	INN_VAE_FR8, INN_VAE_FR9, INN_VAE_FR10	
Precondition	Internet connection availability. Establishment of communication with the field middleware. Uninterrupted raw data flow from the prosumer premises.	
Test environment (optional)	-	
Step (actions)	**Obtained result**	**Verdict**
1. Log in the VAE	-	Passed
2. Navigate to the performance analysis for selected assets with historical data availability	-	Passed
3. Evaluate the quality of historical data visualisation and the speed of retrieving information	The response time of the interface in retrieving historical data is considered acceptable. Depending on the required period, small delays of a few seconds in retrieving historical data might be observed.	Passed

Test Case	**TC_VAE_05**	Evaluation of the interfaces defined for data exchange with other tools
Related Test Cases	N/A	
Functional Requirements	INN_VAE_FR12, INN_VAE_FR16	
Precondition	N/A	
Test environment (optional)	-	
Step (actions)	**Obtained result**	**Verdict**
1. Interface definition	Interfaces for data exchange	Passed
2. Control that the input and output are in line with the common information model	As	Passed

INN_DSEP_FR1	The Demand Side Energy Profiling tool shall subscribe to retrieve all the information useful for fine-tune prosumers Profiling Models
INN_DSEP_FR2	The Demand Side Energy Profiling tool shall retrieve periodically (event-based approach) information about environmental and operational conditions
INN_DSEP_FR3	The Demand Side Energy Profiling tool shall periodically update prosumers Profiling Models
INN_DSEP_FR4	The Demand Side Energy Profiling tool shall retrieve upon request the information required for fine-tuning profiling model parameters
INN_DSEP_FR5	The Demand Side Energy Profiling tool shall subscribe to retrieve all the information useful for fine-tuning demand flexibility Profiling Models and calculate demand flexibility in (near) real time
INN_DSEP_FR6	The Demand Side Energy Profiling tool shall retrieve periodically (event-based approach) information about environmental conditions, device operational status and energy consumption
INN_DSEP_FR7	The Demand Side Energy Profiling tool shall periodically update the demand flexibility Profiling Models
INN_DSEP_FR8	The Demand Side Energy Profiling tool shall periodically update (short term) the demand flexibility potential of the different controllable DERs
INN_DSEP_FR9	The Demand Side Energy Profiling tool shall retrieve upon request the information required for fine-tuning of demand flexibility profiling parameters
INN_DSEP_FR10	The Demand Side Energy Profiling tool shall evaluate the potential of triggering a DR strategy
INN_DSEP_FR11	The Demand Side Energy Profiling tool shall trigger the DR strategy
INN_DSEP_FR12	The Demand Side Energy Profiling tool shall select the assets (and the associated smart devices) to participate in an auto DR campaign
INN_DSEP_FR13	The Demand Side Energy Profiling tool shall manage Acknowledge Messages from Field Middleware for auto DR command activation
INN_DSEP_FR14	The Demand Side Energy Profiling tool shall acquire raw and processed data (time series: energy consumption, demand flexibility & contextual conditions)

INN_DSEP_FR15	The Demand Side Energy Profiling tool shall manage exceptions in case of non-availability of data or lack of communication
INN_VAE_FR1	Simulation analysis to evaluate the potential of the portfolio to participate in DR campaigns. The results from the simulation process should further enable the implementation of automated DR strategies at the FR pilot site
INN_VAE_FR2	Actual Flexibility Calculation & Remuneration for DR participation
INN_VAE_FR3	Gets insights about portfolio performance. The end user must be able to get an overview of the portfolio performance
INN_VAE_FR4	Presentation of KPI results to the end users. The Analytics tool should provide analysis and visualization of KPI trends
INN_VAE_FR5	The end user must be able to select from a list of KPIs and set the temporal, spatial, operational etc. values in order to retrieve dynamic reports about clusters of prosumers
INN_VAE_FR6	The tool must be interactive - i.e. offer the option to drill-down to individual (building) details, focusing on the performance of each customer of the portfolio
INN_VAE_FR7	Users should be able to select from a list of KPIs to compare prosumers performance over a selected time period. The KPIs selected will set the parameters for customized analytics
INN_VAE_FR8	The Analytics tool must allow working with different levels of system detail (data hierarchy and aggregation)
INN_VAE_FR9	The Analytics tool must allow working with (near) real time & historical data sets
INN_VAE_FR10	For analytics over historical data, a database with different data types (energy, flexibility etc.…) should be managed by the tool
INN_VAE_FR11	Users must be able to evaluate the impact of different DSM strategies through interaction with the visualization component
INN_VAE_FR12	In order to retrieve data related to DSM strategies implementation (level of fulfilment at DSM strategy, prosumers compensation for participating in DSM programs), required for simulation process, interfaces with other software module deployed in FR pilot sites should be defined
INN_VAE_FR13	The Analytics tool must support portfolio segmentation & clustering over KPI values from different domains (Energy, Flexibility etc...)
INN_VAE_FR14	The Analytics tool should support metric/KPIs classification, classifying data into predefined groups based on their features
INN_VAE_FR15	The Analytics tool should support outlier's detection for metrics/KPIs, based on thresholds provided by the end users of the system
INN_VAE_FR16	In order to retrieve data required for analytics process, interfaces with other software module deployed in FR pilot sites should be defined

Nicosia

Test Case	T_UCA1	Monitoring of the energy production and consumption of the dispersed prosumers with RES and storage (DR tool)	
Related Use Cases	UC06		
Functional Requirements	UCY_DR_FR1, UCY_DR_FR2, UCY_DR_FR3, UCY_DR_FR4, UCY_DR_FR5, UCY_DR_FR6, UCY_DR_FR7, UCY_DR_FR8, UCY_DR_FR9		
Precondition	Setup of HI-Q Universe		
Test environment (optional)	Manual		
No.	Step (actions)	Obtained result	Verdict
1	Login into HI-Q Universe	Presentation page displayed with predefined dashboards.	Passed
2	Tools: Basic Information	User information	Passed
3	Tools: Instance manager	HEMS Interactive list and aggregate results including KPIs	Passed
4	Tools: Instance manager: HEMS_Number: Settings	Device details, Energy price, Time plots range	Passed
5	Tools: Instance manager: HEMS_Number: Operation	FOA Operation, Grid power data series, Producer power data series, Consumer power data series, Battery power data series, KPIs	Passed
6	Tools: Instance manager: HEMS_Number: Alarms	Alarms and Events page	Passed
7	Tools: Instance manager: HEMS_Number: My things	Available flexibility data, Trading service type	Passed
8	Tools: reports	Alarm notification page	Passed
9	Tools: Data API account manager	Data API account manager page	Passed

Test Case	T_UCB1	Centrally monitoring the energy production and consumption of the university micro-grid
Related Use Cases	UC06	
Functional Requirements	NIC_MSM_FR1, NIC_MSM_FR2, NIC_MSM_FR3, NIC_MSM_FR4, NIC_MSM_FR5, NIC_MSM_FR6, NIC_MSM_FR7, NIC_MSM_FR8	

Precondition		Installation and configuration of inEIS	
Test environment (optional)		Manual	
No.	Step (actions)	**Obtained result**	**Verdict**
1	Login into inEIS	Presentation page displayed with predefined dashboards.	Passed
2	Control Panel: Map	Microgrid Map with the position and instantaneous readings of smart meters	Passed
3	Control Panel: Energy Flow	Power distribution, generation and consumption flow	Passed
4	Control Panel: Electrical Meters	Display of metered values, generation and consumption	Passed
5	Control Panel: FEB01	Air temperature control (Normal/Automatic/Save/Boost mode) of Faculty of Economics and Business Building No. 1	Passed
6	Control Panel: FEB02	Air temperature control (Normal/Automatic/Save/Boost mode) of Faculty of Economics and Business Building No. 2	Passed
7	Control Panel: CTF02	Air temperature control (Normal/Automatic/Save/Boost mode) of Common Teaching Facilities	Passed
8	Control Panel: LRC	Air temperature control (Normal/Automatic/Save/Boost mode) of Library Building	Passed
9	Monitoring: Measured Custom	Load Curves: Print, XML Export	Passed
10	Monitoring: Measured Hierarchy	Production Curves: Print, XML Export	Passed
11	Monitoring: Time Comparison	Compare measurements across different time spans	Passed
12	Alarms: Current alarms	Current alarm list, importance, status	Passed
13	Alarms: History	Alarm history	Passed
14	Alarms: Tag Status	System status	Passed
15	Consumption: Consumption overview	Consumption overview	Passed
16	Consumption: Consumption compare	Consumption in period	Passed
17	Consumption: Consumption history	Consumption history	Passed
18	Consumption: Spectral Analysis	Spectral analysis	Passed
19	Settings: Events and Alarms	Events will trigger alarms and other actions	Passed

20	Settings: Users	Manage access and contact info	Passed
21	Settings: User groups	User Groups are used for permissions and alarming	Passed
22	Settings: Charts	Charts will display under Monitoring section	Passed
23	Settings: Milestones	Milestones will display on all timeseries charts	Passed
24	Settings: Schedules	Schedule List - Scheduler Parameters	Passed
25	Info	System information and manual	Passed

Test Case		T_UCB2	Manual control of the energy production and consumption of the university micro-grid (platform installed at the university)
Related Use Cases		UC06	
Functional Requirements		NIC_MSM_FR1, NIC_MSM_FR2, NIC_MSM_FR3, NIC_MSM_FR4, NIC_MSM_FR5, NIC_MSM_FR6, NIC_MSM_FR7, NIC_MSM_FR8	
Precondition		Local control through proprietary software, Siemens BEMS, Honeywell BEMS, ENC Boiler System,	
Test environment (optional)		Manual	
Step (actions)		**Obtained result**	**Verdict**
1	Login Page	Presentation page displayed with predefined dashboards.	Passed
2	Overview page	Overview page where a system overview is presented	Passed
3	Operation pages	Automatic / Manual operation of systems	Passed
4	Reporting pages	Alarm/Event pages, Reporting, Data export	Passed

Lisbon

Test Case	T_UCA1	Retrieve the energy usage profile of the pilot building	
Related Use Cases	UC01		
Functional Requirements	LIS_EMS_FR1, LIS_EMS_FR2, LIS_EMS_FR3, LIS_EMS_FR4		
Precondition	Kisense EMS upgrade		
Test environment (optional)	Production platform		
Step (actions)	**Obtained result**		**Verdict**
1. Login into Kisense	Presentation page displayed with predefined dashboards.		Passed
2. Select the 'Profile' option on the 'Analysis' module	Profile page of the Analysis module displayed with input fields for selecting the circuit, time interval, and type of profile.		Passed
3. Select a circuit from the 'Consumption' tree node and a time interval	Measured consumption profile for the selected circuit (incomer or partial) and time interval displayed.		Passed
4. Download excel file	A excel file with the calculate profile was downloaded.		Passed

Test Case	T_UCA2	Suggest energy usage profile for the pilot building	
Related Use Cases	UC01		
Functional Requirements	LIS_EMS_FR5, LIS_EMS_FR6, LIS_EMS_FR7, LIS_EMS_FR8		
Precondition	Kisense EMS upgrade		
Test environment (optional)	Production platform		
Step (actions)	**Obtained result**		**Verdict**
1. Login into Kisense	Presentation page displayed with predefined dashboards.		Passed
2. Select the 'Profile' option on the 'Analysis' module	Profile page of the Analysis module displayed with input fields for selecting the circuit, time interval, and type of profile.		Passed
3. Select a circuit from the 'Optimizer' tree node and a time interval	Optimized consumption profile for the selected circuit is displayed on and time interval selected.		Passed
4. Download excel file	A excel file with the optimized profile was downloaded.		Passed

Test Case	T_UCA3	Implement the optimal energy usage profile	
Related Use Cases	UC01		
Functional Requirements	LIS_EMS_FR9, LIS_EMS_FR10, LIS_EMS_FR11, LIS_EMS_FR12, LIS_EMS_FR13		
Precondition	Kisense EMS upgrade		
Test environment (optional)	Production platform		
Step (actions)	**Obtained result**		**Verdict**
1. Login into Kisense	Presentation page displayed with predefined dashboards.		Passed
2. Select the 'Profile' option on the 'Analysis' module	Profile page of the Analysis module displayed with input field for selecting the circuit, time interval, and type of profile.		Passed
3. Select a circuit from the 'Optimizer' tree node and a time interval	Optimized consumption profile for the selected circuit is displayed on and time interval selected.		Passed
4. Select 'Apply'	Scheduled load actuations displayed on a pop-up window and sent to control module		failed, no flexible loads available

Test Case	T_UCA4	Retrieve the PV plant production profile	
Related Use Cases	UC01		
Functional Requirements	LIS_EMS_FR14, LIS_EMS_FR15, LIS_EMS_FR16, LIS_EMS_FR17		
Precondition	Kisense EMS upgrade and PV plant installation		
Test environment (optional)	Production platform		
Step (actions)	**Obtained result**		**Verdict**
1. Login into Kisense	Presentation page displayed with predefined dashboards.		Passed
2. Select the 'Basic' option on the 'Explorer' module	Basic page of the Explorer module displayed with input field for selecting the time interval.		Passed
3. Select the 'Generation' circuit on the tree and a time interval	PV production data for the selected circuit and time interval was displayed.		Passed, with simulated production data

Test Case	T_UCA5	Retrieve the PV plant production forecast	
Related Use Cases	UC01		
Functional Requirements	LIS_EMS_FR18, LIS_EMS_FR19		
Precondition	Kisense EMS upgrade and PV plant installation		
Test environment (optional)	Production platform		
Step (actions)	Obtained result		Verdict
1. Login into Kisense	Presentation page displayed with predefined dashboards.		Passed
2. Select the 'Basic' option on the 'Explorer' module	Basic page of the Explorer module displayed with input field for selecting the time interval.		Passed
3. Select the 'Generation' circuit on the tree and a time interval	PV production and forecast data for the selected circuit and time interval was displayed. Forecast data is only displayed for dates in the future.		Passed
4. Select the option "Show all forecast".	PV forecast for the selected circuit are displayed on a distinct data series, so it can be compared with the PV production.		Passed

Test Case	T_UCC1	Retrieve the charging status of the ice banks of the pilot building	
Related Use Cases	UC02		
Functional Requirements	LIS_EMS_FR20, LIS_EMS_FR21, LIS_EMS_FR22, LIS_EMS_FR23		
Precondition	Kisense EMS upgrade and integration with BMS (SCADA)		
Test environment (optional)	Production platform		
Step (actions)	**Obtained result**		**Verdict**
1. Login into Kisense	Presentation page displayed with predefined dashboards.		Passed
2. Select the 'Basic' option on the 'Explorer' module	Basic page of the Explorer module displayed with input field for selecting the circuit, time interval, and variables.		Passed
3. Select a circuit (Ice Bank) from the 'HVAC Cooling' tree node, the 'Water level'	Measured charging status data for the selected ice bank and time interval displayed.		Passed

variable, and a time interval		

Test Case	T_UCC2	Suggest energy usage for the chillers	
Related Use Cases	UC02		
Functional Requirements	LIS_EMS_FR28, LIS_EMS_FR29, LIS_EMS_FR30, LIS_EMS_FR31		
Precondition	Kisense EMS upgrade and integration with BMS (SCADA)		
Test environment (optional)	Production platform		
Step (actions)	**Obtained result**		**Verdict**
1. Login into Kisense	Presentation page displayed with predefined dashboards.		Passed
2. Select the 'Basic' option on the 'Explorer' module	Basic page of the Explorer module displayed with input field for selecting the circuit, time interval, and variables.		Passed
3. Select a circuit (Ice Bank) from the 'Optimizer' tree node and a time interval	Optimized charging profile for the selected ice bank is displayed on and time interval selected.		failed, no charging data available

Test Case	T_UCD1	Retrieve EV charging profile	
Related Use Cases	UC03		
Functional Requirements	LIS_EMS_FR32, LIS_EMS_FR33, LIS_EMS_FR34, LIS_EMS_FR35		
Precondition	Kisense EMS upgrade and integration with EV charging platforms		
Test environment (optional)	Production platform		
Step (actions)	**Obtained result**		**Verdict**
1. Login into Kisense	Presentation page displayed with predefined dashboards.		Passed
2. Select the 'Profile' option on the 'Analysis' module	Profile page of the Analysis module displayed with input field for selecting the circuit, time interval, and type of profile.		Passed

3. Select a circuit of the 'Consumption\EV Chargers' tree node and a time interval	Measured charging profile for the selected charging point and time interval displayed.	Passed

Test Case	T_UCD2	Manage EV charging	
Related Use Cases	UC03		
Functional Requirements	LIS_EMS_FR36, LIS_EMS_FR37, LIS_EMS_FR38		
Precondition	Kisense EMS upgrade and integration with EV charging platforms		
Test environment (optional)	Production platform		
Step (actions)	**Obtained result**		**Verdict**
1. Login into Kisense	Presentation page displayed with predefined dashboards.		Passed
2. Select the 'Profile' option on the 'Analysis' module	Profile page of the Analysis module displayed with input fields for selecting the circuit, time interval, and type of profile.		Passed
3. Select a circuit from the 'Optimizer\EV Chargers' tree node and a time interval	Optimized charging profile for the selected circuit is displayed on and time interval selected.		Passed, with settings restrictions
4. Download excel file	A excel file with the optimized profile was downloaded.		Passed

Test Case	[Test ID]	testing of Plant/Process/System Data Exchange tool (DET) performance and functionalities	
Related Use Cases	SUN_UC01, SUN_UC03		
Functional Requirements	SUN_PPSDET_FR1, SUN_PPSDET_FR2, SUN_PPSDET_FR3, SUN_PPSDET_FR5, SUN_PPSDET_FR5		
Precondition	The OPC server and the MQTT server are online and running		
Test environment (optional)			
Step (actions)	**Obtained result**		**Verdict**
1. Communication with OPC server	Search of network OPC servers, selection of an OPC and connection made. Desired test data exchanged.		Passed
2. Communication with MQTT server	Connection to the desired MQTT server made. Desired test data exchanged.		Passed
3. Retrieve namespace of OPC server	Problems occurred when tried to retrieve namespace with large amount of tags on a remote PC. Problem solved. Namespace retrieved successfully.		Passed
4. Categorize variables in groups. Initiate and terminate transmission of grouped variables	process occurs with no bugs		Passed
5. processing	process occurs with no bugs		Passed

Test Case	[Test ID]	testing of Integrated Decision Support & Supervisory EMS (IDS&SS) performance and functionalities
Related Use Cases	SUN_UC01	
Functional Requirements	SUN_ID&SSS_FR1, SUN_ID&SSS_FR2, SUN_ID&SSS_FR3, SUN_ID&SSS_FR4, SUN_ID&SSS_FR5, SUN_ID&SSS_FR6, SUN_ID&SSS_FR7, SUN_ID&SSS_FR8, SUN_ID&SSS_FR9, SUN_ID&SSS_FR10, SUN_ID&SSS_FR11, SUN_ID&SSS_FR12, SUN_ID&SSS_FR13, SUN_ID&SSS_FR14, SUN_ID&SSS_FR19, SUN_ID&SSS_FR20, SUN_ID&SSS_FR21, SUN_ID&SSS_FR22, SUN_ID&SSS_FR23, SUN_ID&SSS_FR24, SUN_ID&SSS_FR28, SUN_ID&SSS_FR29, SUN_ID&SSS_FR30	
Precondition	the MQTT server is online and running	
Test environment (optional)		

Step (actions)	**Obtained result**	**Verdict**
1. Connect to MQTT server	connection made. Test data exchanged.	Passed
2. Retrieve online data	Desired test data retrieved.	Passed
3. Visualize online data	Desired test data visualized in LabVIEW monitoring interface	Passed
4. input weather and load profiles	Input method needs adjustments	needs adjustments
5. Invoke microgrid models for day ahead profiling	Given input data the process occurs with no problems	Passed
6. Visualize day ahead profiling	Visualization of processed data occurs well, selection of final visualized data needs adjustments	needs adjustments
7. Data communication with the visualization module	Communication with test data occurred well. Needs adjustments for final data format	needs adjustments
8.Optimization processing	process occurs with no bugs	Passed

Test Case	[Test ID]	testing Supervisory Model Predictive Control for Energy Systems (MPC) performance and functionalities
Related Use Cases	SUN_UC01, SUN_UC02, SUN_UC03	
Functional Requirements	SUN_SMPCFES_FR1 , SUN_SMPCFES_FR2, SUN_SMPCFES_FR3, SUN_SMPCFES_FR4, SUN_SMPCFES_FR6, SUN_SMPCFES_FR7, SUN_SMPCFES_FR8, SUN_SMPCFES_FR9, SUN_SMPCFES_FR10, SUN_SMPCFES_FR11, SUN_SMPCFES_FR12, SUN_SMPCFES_FR13, SUN_SMPCFES_FR14, SUN_SMPCFES_FR15	
Precondition		
Test environment (optional)		
Step (actions)	**Obtained result**	**Verdict**
1. input weather and load profiles	Input method needs adjustments	needs adjustments
2. Model predictive control algorithm processing	Given input data the process occurs with no problems	Passed
3. Retrieve control actions profile for	Control actions profile is obtained with no problems	Passed

energy exchange		
4. Communication with IDS&SS tool Visualization module)	Communication with test data occurred well. Needs adjustments for final data format	needs adjustments

Ploiesti

Functional testing

TC	1.1	**Visualising relevant data about consumption – Accessing the inteGRIDy Visualization Platform & Services (IVP) layer**
UC	A.1 Display relevant data regarding own consumption	
FRs	PLO_EIIS_FR10, PLO_EIIS_FR10.1, PLO_EIIS_FR10.2, PLO_EIIS_FR11, PLO_EIIS_FR7	
Precondition(s)	- The software platform is operational. The specific mechanisms of authentication are configured in the system. - The interoperability link (interconnection) between the software platform and the smart metering infrastructure is operational.	
Test environment (optional)		

TC Step (actions)	**Obtained result**	**Verdict**
1. The user can login in the system with the given credentials and role.	Authentication process is successfully completed.	Passed
2. The user can change the password and the user profile data.	The required updates are successfully completed.	Passed
3. The user (Consumer / individual person) can login in the system and require data about the own consumption.	The system is capable to return data about the own consumption to the user.	Passed
4. The user (DSO) can login in the system and request information about consumption for one or more Consumers, for a given period.	The system successfully returns the specific information to the user.	Passed
5. The user (DSO) can login in the system and request the prediction of consumption for a period of time for one or more Consumers.	The system successfully returns the specific data about the requested prediction of consumption.	Passed

TC	1.2	**Visualising relevant reports and history – Accessing the inteGRIDy Visualization Platform & Services (IVP) layer**
UC	A.1 Display relevant data regarding own consumption	
FRs	PLO_EIIS_FR24, PLO_EIIS_FR25	
Precondition	The software platform is operational.	

	The specific mechanisms of reporting and statistics are configured in the system.
Test environment (optional)	

TC Step (actions)	Obtained result	Verdict
1. The system should provide various types of reports through the reporting component implemented at the logical level.	The system successfully issues the required reports.	Passed
2. The system should provide system usage reports.	The system successfully issues the system usage reports.	Passed

TC	1.3	inteGRIDy Integration & interconnection Plan and Report - Field Middleware
UC	A.1 Display relevant data regarding own consumption	
FRs	PLO_EIIS_FR7.4	
Precondition	The interoperability link between the software system and the database comprising historical data is operational.	
Test environment		

TC Step (actions)	Obtained result	Verdict
1. The system should provide historical data coming from the Field Middleware.	The system is able to request and retrieve historical data issued from the Field Middleware.	Passed

TC	1.4	Accessing relevant data concerning the consumption profile – Data handling within RKW Reference Knowledge Warehouse layer
UC	A.1 Display relevant data regarding own consumption	
FRs	PLO_EIIS_FR7.1, PLO_EIIS_FR7.2, PLO_EIIS_FR7.3	
Precondition	The interoperability link between the software system and the Reference Knowledge Warehouse is operational.	
Test environment		

TC Step (actions)	Obtained result	Verdict
1. The system should be able to upload data in	The system successfully uploads data in the Reference Knowledge Warehouse.	Passed

Reference Knowledge Warehouse.		
2. The system should be able to request and retrieve the consumption profile from the Reference Knowledge Warehouse.	The system is able to request and retrieve the consumption profile from the Reference Knowledge Warehouse.	Passed
3. The system should be able to store historical data in a local database.	The system successfully stores historical data in a local database.	Passed

TC	**2.1**	**Connection with the smart metering infrastructure (smart meters) - inteGRIDy Integration & interconnection Plan and Report**
UC	A.2 Data registration regarding the consumption	
FRs	PLO_EIIS_FR1, PLO_EIIS_FR2, PLO_EIIS_FR3, PLO_EIIS_FR4, PLO_EIIS_FR5, PLO_EIIS_FR8	
Precondition	- Smart meters are physically connected in the smart metering infrastructure and they are operational. - The interoperability link (interconnection) between the software platform and the smart metering infrastructure is operational.	
Test environment		

TC Step (actions)	**Obtained result**	**Verdict**
1. Smart meters should be able to send metering data to the software platform / system.	Smart meters (belonging to smart metering infrastructure) are able to send metering data to the software platform / system.	Passed
2. The system should be able to receive / collect data from the smart metering infrastructure in given time intervals of 15 minutes.	The system successfully receives / collects data from the smart metering infrastructure in given time intervals of 15 minutes.	Passed
3. The system should be capable to store data received from the smart metering infrastructure.	The system successfully stores data received from the smart metering infrastructure.	Passed
4. The system should be capable of automatically reading data from the smart metering infrastructure.	The system is able to automatically read data from the smart metering infrastructure.	Passed
5. The interoperability connection (interconnection) between	The dedicated API (Application Programming Interface) which realizes the interconnection	Passed

the software platform / system and the smart metering infrastructure should be realized by a dedicated communication API (Application Programming Interface).	between the software platform and smart metering infrastructure is fully operational.	
6. The system should be able to register data about consumption over a given period of time for all Consumers.	The system successfully registers data about consumption over a given period of time for all Consumers,	Passed

TC	**2.2**	**System administration - inteGRIDy Security Access Control Framework**	
UC	A.2 Data registration regarding the consumption		
FRs	PLO_EIIS_FR9		
Precondition	The software platform is operational. The administration component is configured in the system.		
Test environment			
TC Step (actions)		**Obtained result**	**Verdict**
1. The system should have an administration console that would manage the roles and privileges (access rights) for all users.		The system has an administration console which allows the management of roles and privileges (access rights) for all users.	Passed
2. The administration console should be accessed only by users with admin privileges (admin user rights) configured in the system.		The users having the admin role (admin privileges) are the only ones to access the administration console.	Passed

TC	**2.3**	**Displaying the collected / registered data - Accessing the inteGRIDy Visualization Platform & Services (IVP) layer**	
UC	A.2 Data registration regarding the consumption		
FRs	PLO_EIIS_FR9.1		
Precondition	The software platform is operational and has successfully stored historical data on consumption for the given time interval.		
Test environment			
TC Step (actions)		**Obtained result**	**Verdict**

1. The system should be able to display data as Report or Chart formats.	The system successfully displays data as Report or Chart.	Passed
		Passed

TC	**2.4**	**Management of data collected / registered – Accessing the inteGRIDy Reference Knowledge Warehouse (RKW) layer**
UC	A.2 Data registration regarding the consumption	
FRs	PLO_EIIS_FR8.1	
Precondition	The software platform is operational. OLAP (Online Analytical Processing) data structures (in RKW) are defined and able to receive data.	
Test environment		

TC Step (actions)	**Obtained result**	**Verdict**
1. The system should be able to feed an OLAP type structure (data warehouse).	The system is able to feed an OLAP type structure (in RKW).	Passed
		Passed

TC	**3.1**	**Accessing relevant data concerning the consumption – Data handling within inteGRIDy Decision Making & Optimization Mechanisms layer**
UC	A3. Identify peak times	
FRs	PLO_EIIS_FR12.1, PLO_EIIS_FR12.2, PLO_EIIS_FR12.5	
Precondition	The software platform is operational and has successfully stored historical data on consumption for the given time interval.	
Test environment		

TC Step (actions)	**Obtained result**	**Verdict**
1. The system should be able to read historical data for a period of time.	The system is able to read historical data for a period of time.	Passed
2. The system should be able to compute peaks times for a given period of time.	The system is able to compute peaks times for a given period of time.	Passed
3. The system should be able to access historical data from the local database.	The system is able to access historical data from the local database.	Passed

TC	3.2	Visualising relevant data about peak times of consumption for a certain period – Accessing the inteGRIDy Visualization Platform & Services (IVP) layer
UC	A3. Identify peak times	
FRs	PLO_EIIS_FR12, PLO_EIIS_FR12.3, PLO_EIIS_FR12.4	
Precondition	The software platform is operational and has successfully stored historical data on consumption for the given time interval.	
Test environment		

TC Step (actions)	Obtained result	Verdict
1. The user (DSO) can login in the system and request information about peak times of consumption for a given time interval.	The system successfully returns the specific data about the requested peak times of consumption for the given time interval.	Passed
2. The system should be able to display peaks times for a given period of time in reports and charts.	The system successfully displays peaks times for the given period of time in Reports or Charts.	Passed
3. The user (DSO) can login in the system and request data about the optimization of the grid.	The system successfully returns the specific data concerning the optimization of the grid.	Passed

TC	4.1	Modelling the grid behaviour – Accessing the inteGRIDy Modelling Mechanisms layer
UC	A4. Optimize power distribution	
FRs	PLO_EIIS_FR14.1	
Precondition	The software platform is operational and has successfully stored historical data on consumption for the given time interval.	
Test environment		

TC Step (actions)	Obtained result	Verdict
1. The system should allow to model (implement modelling mechanisms for) the grid behaviour.	The system is able to model the grid behaviour.	Passed

TC	4.2	Managing the operations – Accessing the inteGRIDy Operation Analysis Framework layer
UC	A4. Optimize power distribution	
FRs	PLO_EIIS_FR14.3	
Precondition	The software platform is operational and has successfully stored historical data on consumption for the given time interval.	
Test environment		
TC Step (actions)	**Obtained result**	**Verdict**
1. The system should be able to analyse the operations in a period of time and propose optimal setup parameters.	The system is able to analyse the operations for the particular period of time and successfully returns / proposes the optimal setup parameters.	Passed

TC	4.3	Optimizing the power distribution – Accessing the inteGRIDy Decision Making & Optimization Mechanisms
UC	A4. Optimize power distribution	
FRs	PLO_EIIS_FR14; PLO_EIIS_FR14.2; PLO_EIIS_FR14.4; PLO_EIIS_FR14.5	
Precondition	The software platform is operational and has successfully stored historical data on consumption for the given time interval. The Reference Knowledge Warehouse is operational.	
Test environment		
TC Step (actions)	**Obtained result**	**Verdict**
1. The system should propose an optimized consumption pattern for the power grid (related to the Pilot infrastructure).	The system proposes an optimized consumption pattern for the power grid (related to the Pilot infrastructure).	Passed
2. The system should be able to create / compute the optimum grid structure based on the historical data analysed.	The system is able to create / compute the optimum grid structure based on the historical data analysed.	Passed
3. The system should be able to read data from the Reference Knowledge Warehouse.	The system is able to read data from the Reference Knowledge Warehouse.	Passed
4. The system should be able to provide inputs for decision making process.	The system is able to provide inputs for decision making process.	Passed

TC	5.1	Optimizing the consumer bill – Accessing the inteGRIDy Decision Making & Optimization Mechanisms
UC	A5. Optimize consumer bill	
FRs	PLO_EIIS_FR16.3; PLO_EIIS_FR16.4; PLO_EIIS_FR16.5; PLO_EIIS_FR16; PLO_EIIS_FR17	
Precondition	The software platform is operational and has successfully stored historical data on consumption and prices for the given time interval. Data about prices are available.	
Test environment		

TC Step (actions)	Obtained result	Verdict
1. The system should be able to read and store information about (energy) prices.	The system is able to read and store information about (energy) prices.	Passed
2. The system should provide the implementation of price models where the price optimization is an objective.	The system successfully implements price models pursuing the price optimization.	Passed
3. The system should be able to apply smart algorithms to get the optimal consumption in terms of economic savings.	The system successfully uses smart algorithms to get the optimal consumption such as to obtain costs reduction.	Passed
4. The system should propose to consumer an optimized consumption pattern that could enable the consumer to reduce the energy bill.	The system successfully proposes to consumer an optimized consumption pattern enabling the consumer to reduce the energy bill.	Passed
5. The system should propose an optimized consumption pattern using different rates (CPP, TOU, PTR, CTP).	The system successfully proposes an optimized consumption pattern using different rates (CPP, TOU. PTR, CTP).	Passed

TC	5.2	Visualising relevant data about the energy bill based on optimized consumption patterns – Accessing the inteGRIDy Visualization Platform & Services (IVP) layer
UC	A5. Optimize consumer bill	

FRs	PLO_EIIS_FR16.1; PLO_EIIS_FR16.2; PLO_EIIS_FR17.1	
Precondition	The software platform is operational and has successfully stored historical data on consumption and prices for the given time interval.	
Test environment		
TC Step (actions)	**Obtained result**	**Verdict**
1. The system should provide to the consumer (user) a specific UI (User Interface) where it would be possible to ask for the Consumer bill optimization.	The system provides a specific UI where the consumer (user) can ask for Consumer bill optimization.	Passed
2. The user (Consumer) can login in the system and request the bill optimization.	The system successfully returns the specific data concerning the optimization of the bill.	Passed
3. The system should be able to display the proposed consumption patterns for the consumer in reports and charts.	The system successfully displays the proposed consumption patterns for the consumer in Reports and Charts.	Passed

TC	**6.1**	**Accessing external data – inteGRIDy Integration & interconnection Plan and Report**
UC	A6. Elaborate prognosis	
FRs	PLO_EIIS_FR19.1	
Precondition	The software platform is operational and has successfully stored historical data on consumption. External data (provided by external systems) are available.	
Test environment		
TC Step (actions)	**Obtained result**	**Verdict**
1. The system should be able to read external data (provided by external systems, like weather forecasts).	The system is able to read external data (like weather forecasts).	Passed

TC	**6.2**	**Modelling the grid behaviour using external factors – Accessing the inteGRIDy Modelling Mechanisms layer**
UC	A6. Elaborate prognosis	
FRs	PLO_EIIS_FR19.2	

Precondition	The software platform is operational and has successfully stored historical data on consumption. External data (provided by external systems) are available.
Test environment	

TC Step (actions)	Obtained result	Verdict
1. The system should allow to model (implement modelling mechanisms for) the grid behaviour using also external factors like weather forecasts.	The system allows to model the grid behaviour using also external factors like weather forecasts.	Passed

TC	6.3	Managing the operations (simulations) – Accessing the inteGRIDy Operation Analysis Framework layer
UC	A6. Elaborate prognosis	
FRs	PLO_EIIS_FR27	
Precondition	The software platform is operational.	
Test environment		

TC Step (actions)	Obtained result	Verdict
1. The system should be capable of running simulations within a simulation component at the logical level.	The system successfully implements and runs simulations within the simulation component at the logical level.	Passed

TC	6.4	Implementing prognosis – Accessing the inteGRIDy Decision Making & Optimization Mechanisms
UC	A6. Elaborate prognosis	
FRs	PLO_EIIS_FR18; PLO_EIIS_FR19.3	
Precondition	The software platform is operational and has successfully stored historical data on consumption.	
Test environment		

TC Step (actions)	Obtained result	Verdict
1. The system should be able to define a prognosis of power consumption for a given (future) period.	The system is able to define a prognosis of power consumption for a given period (in the future).	Passed
2. The system should be able to elaborate prognosis based on	The system successfully elaborates prognosis based on historical data stored in the local	Passed

historical data from the local database and the Reference Knowledge Warehouse.	database and the Reference Knowledge Warehouse.	

TC	**6.5**	**Visualising relevant data about the prognosis – Accessing the inteGRIDy Visualization Platform & Services (IVP) layer**
UC	A6. Elaborate prognosis	
FRs	PLO_EIIS_FR19; PLO_EIIS_FR27.1	
Precondition	The software platform is operational and has successfully stored historical data on consumption.	
Test environment		
TC Step (actions)	**Obtained result**	**Verdict**
1. The user (DSO) can login in the system and request a prognosis for a given (future) consumption period.	The system successfully returns the prognosis for the required future consumption period.	Passed
2. The system should be able to display the prognosis in reports and charts.	The system successfully displays the prognosis results in Reports and Charts.	Passed

TC	**7.1**	**Automatic control based on DR / DSM algorithms – Accessing the inteGRIDy Decision Making & Optimization Mechanisms**
UC	A7. Automated controlling	
FRs	PLO_EIIS_FR34.1; PLO_EIIS_FR34.2; PLO_EIIS_FR34.3	
Precondition	The software platform is operational. Devices belonging to Field Middleware (smart metering infrastructure) are able to send data in real time and are able to run commands.	
Test environment		
TC Step (actions)	**Obtained result**	**Verdict**
1. The system should be able to read data in real time.	The system is able to read data in real time.	Passed
2. The system should be able to send commands.	The system is able to send commands.	Passed
3. The system should be able to receive feedbacks.	The system is able to receive feedbacks.	Passed

TC	7.2	Visualising relevant data concerning DR (automatic control and notifications) – Accessing the inteGRIDy Visualization Platform & Services (IVP) layer
UC	A7. Automated controlling	
FRs	PLO_EIIS_FR20; PLO_EIIS_FR34	
Precondition	The software platform is operational and has successfully stored historical data on consumption.	
Test environment		

TC Step (actions)	Obtained result	Verdict
1. The user (DSO) may have access to the automatic control of power demand based on smart algorithms incorporated in the software platform considering a given (future) time interval.	The user (DSO) has access to the features of automatic control of power demand based on smart algorithms incorporated in the software platform considering a given (future) time interval.	Passed
2. The user (Consumer) can login in the system and receive notifications about optimized consumption patterns based on smart DR algorithms.	The authorized user receives / visualises notifications about optimized consumption patterns resulting from the application of DR algorithms.	Passed

TC	8.1	Generating alerts and notifications – Accessing the inteGRIDy Decision Making & Optimization Mechanisms
UC	A8. Alerts	
FRs	PLO_EIIS_FR21.4; PLO_EIIS_FR21; PLO_EIIS_FR22	
Precondition	The software platform is operational.	
Test environment		

TC Step (actions)	Obtained result	Verdict
1. The system should have a rule engine which is able to create notifications or alerts based on data received.	The system has a rule engine which is able to create notifications or alerts based on data received.	Passed
2. The system should be capable of issuing various types of standard alerts within an alerts	The system successfully implements and issues various types of standard alerts within the alerts component at the logical level.	Passed

component at the logical level.		
3. The system should be able of issuing alerts on peak consumptions based on definable thresholds for a given reference period.	The system is able to issue alerts on peak consumptions based on definable thresholds for a given reference period.	Passed

TC	**8.2**	**Visualising alerts – Accessing the inteGRIDy Visualization Platform & Services (IVP) layer**
UC	A8. Alerts	
FRs	PLO_EIIS_FR21.1; PLO_EIIS_FR21.2; PLO_EIIS_FR21.3	
Precondition	The software platform is operational.	
Test environment		
TC Step (actions)	**Obtained result**	**Verdict**
1. The system should have a specific UI (User Interface) where alerts are immediately displayed, based on their severity and type.	The system provides a specific UI where the alerts are immediately displayed, based on their severity and type.	Passed
2. The user should be able to dismiss the alerts.	The user is able to dismiss the alerts.	Passed
3. The user should be able to define the alerts (conditions, time intervals, severity, messages).	The user is able to define the alerts (conditions, time intervals, severity, messages).	Passed

TC	**9.1**	**Visualising information about the results of optimizations – Accessing the inteGRIDy Visualization Platform & Services (IVP) layer**
UC	A9. Get information about the results of optimizations	
FRs	PLO_EIIS_FR26; PLO_EIIS_FR26.1; PLO_EIIS_FR26.2; PLO_EIIS_FR26.3	
Precondition	The software platform is operational. The software platform has successfully stored historical data on consumption for the given time interval.	
Test environment		

TC Step (actions)	Obtained result	Verdict
1. The user (DSO) can login in the system and request reports on consumption and peak load decrease based on the implementation of smart DR algorithms.	The authorized user (DSO) visualises reports on consumption and peak load decrease resulting from the application of DR algorithms.	Passed
2. The user (Consumer) can login in the system and request information resulted from the comparison about two periods of time referring to peak intervals.	The authorized user (Consumer) receives / visualises information resulted from the comparison about two periods of time referring to peak intervals.	Passed
3. The user (Consumer) can login in the system and request information resulted from the comparison about two periods of time referring to prices.	The authorized user (Consumer) receives / visualises information resulted from the comparison about two periods of time referring to prices.	Passed
4. The system should be able to display comparisons of data for two different periods of time in reports and charts.	The system successfully displays comparisons of data for two different periods of time in Reports and Charts.	Passed

TC	**10.1**	**Implementing Monte Carlo simulations – Accessing the inteGRIDy Modelling Mechanisms layer**
UC	A10. Run what-if scenarios	
FRs	PLO_EIIS_FR29.2	
Precondition	The software platform is operational. Historical data are available. The Reference Knowledge Warehouse is available.	
Test environment		

TC Step (actions)	Obtained result	Verdict
1. The system should be able to record the results of Monte Carlo simulations.	The system is able to record the results of Monte Carlo simulations.	Passed

TC	10.2	Managing the operations (what-if scenarios, Monte Carlo simulations) – Accessing the inteGRIDy Operation Analysis Framework layer
UC	A10. Run what-if scenarios	
FRs	PLO_EIIS_FR28; PLO_EIIS_FR29.1	
Precondition	The software platform is operational. Historical data are available. The Reference Knowledge Warehouse is available.	
Test environment		

TC Step (actions)	Obtained result	Verdict
1. The system should be capable of running what-if scenarios within a simulation component at the logical level.	The system successfully runs what-if scenarios within the simulation component at the logical level.	Passed
2. The system should be able to run Monte Carlo Simulations by using historical data and data from the Reference Knowledge Warehouse.	The system successfully runs Monte Carlo simulations based on historical data and data from the Reference Knowledge Warehouse.	Passed

TC	10.3	Visualising the results of what-if scenarios – Accessing the inteGRIDy Visualization Platform & Services (IVP) layer
UC	A10. Run what-if scenarios	
FRs	PLO_EIIS_FR29; PLO_EIIS_FR29.3	
Precondition	The software platform is operational. Historical data are available. The Reference Knowledge Warehouse is available.	
Test environment		

TC Step (actions)	Obtained result	Verdict
1. The user (DSO) can login in the system and request what-if scenarios on the usage of the Grid in order to identify the optimum setup based on DR algorithms given a reference period.	The authorized user (DSO) access what-if scenarios on the usage of the Grid in order to identify the optimum setup based on DR algorithms given a reference period.	Passed
2. The system should be able to display / present	The system is able to display / present comparisons between Monte Carlo simulations.	Passed

comparisons between Monte Carlo simulations.		

TC	10.4	Management of simulation results – Accessing the inteGRIDy Reference Knowledge Warehouse (RKW) layer
UC	A10. Run what-if scenarios	
FRs	PLO_EIIS_FR29.4; PLO_EIIS_FR29.5	
Precondition	The software platform is operational. Historical data are available. The Reference Knowledge Warehouse is available.	
Test environment		
TC Step (actions)	**Obtained result**	**Verdict**
1. The system should be able to upload the simulation results in the Reference Knowledge Warehouse.	The system is able to upload the simulation results in the Reference Knowledge Warehouse.	Passed
2. The system should be able to retrieve simulations from the Reference Knowledge Warehouse.	The system is able to retrieve simulations from the Reference Knowledge Warehouse.	Passed

TC	11.1	Management of prices and costs – Accessing the inteGRIDy Modelling Mechanisms layer
UC	A11. View prices and costs	
FRs	PLO_EIIS_FR30	
Precondition	The software platform is operational.	
Test environment		
TC Step (actions)	**Obtained result**	**Verdict**
1. The system should be able to manage data on pricing and pricing levels.	The system successfully manages data on pricing and pricing levels.	Passed

TC	11.2	Visualising information about pricing – Accessing the inteGRIDy Visualization Platform & Services (IVP) layer
UC	A11. View prices and costs	
FRs	PLO_EIIS_FR32; PLO_EIIS_FR31; PLO_EIIS_FR31.1; PLO_EIIS_FR31.2; PLO_EIIS_FR31.3	

Precondition	The software platform is operational.
Test environment	

TC Step (actions)	Obtained result	Verdict
1. The user (Consumer) can login in the system and request optimization of consumption patterns based on pricing schemes given a reference period.	The authorized user (DSO) successfully access / receives information about optimization of consumption patterns based on pricing schemes given a reference period.	Passed
2. The user (Consumer) can login in the system and request pricing schemes given a reference period.	The authorized user (Consumer) successfully access in the system the pricing schemes given a reference period.	Passed
3. The system should be able to display the prices schemes for different periods of time.	The system is able to display the prices schemes for different periods of time.	Passed
4. The system should be able to compare prices for two different periods of time.	The system is able to compare prices for two different periods of time.	Passed
5. The system should be able to display the influence of price differences in the final bill.	The system successfully displays the influence of price differences in the final bill.	Passed

TC	**12.1**	**Managing the operations (Monte Carlo simulations using historical data and data from RKW, based on prices) – Accessing the inteGRIDy Operation Analysis Framework layer**
UC	A12. Run what-if scenarios based on prices	
FRs	PLO_EIIS_FR33.1	
Precondition	The software platform is operational. Historical data are available. The Reference Knowledge Warehouse is available.	
Test environment		

TC Step (actions)	Obtained result	Verdict
1. The system should be able to run Monte Carlo simulations by using historical data and data from the Reference	The system successfully runs Monte Carlo simulations by using historical data and data from the Reference Knowledge Warehouse, based on prices.	Passed

Knowledge Warehouse, based on prices.		

TC	12.2	Management of simulation results – Accessing the inteGRIDy Reference Knowledge Warehouse (RKW) layer
UC	A12. Run what-if scenarios based on prices	
FRs	PLO_EIIS_FR33.2; PLO_EIIS_FR33.4; PLO_EIIS_FR33.5	
Precondition	The software platform is operational. Historical data are available. The Reference Knowledge Warehouse is available.	
Test environment		
TC Step (actions)	**Obtained result**	**Verdict**
1. The system should be able to record the results of Monte Carlo simulations based on prices.	The system is able to record the results of Monte Carlo simulations based on prices.	Passed
2. The system should be able to upload the results of simulation based on prices in the Reference Knowledge Warehouse.	The system is able to upload the results of simulation based on prices in the Reference Knowledge Warehouse.	Passed
3. The system should be able to retrieve the simulations based on prices from the Reference Knowledge Warehouse.	The system is able to retrieve the simulations based on prices from the Reference Knowledge Warehouse.	Passed

TC	12.3	Visualising the results of what-if scenarios – Accessing the inteGRIDy Visualization Platform & Services (IVP) layer
UC	A12. Run what-if scenarios based on prices	
FRs	PLO_EIIS_FR33; PLO_EIIS_FR33.3	
Precondition	The software platform is operational. Historical data are available. The Reference Knowledge Warehouse is available.	
Test environment		
TC Step (actions)	**Obtained result**	**Verdict**
1. The user (Consumer) can login in the system and run what-if scenarios	The authorized user (Consumer) successfully runs what-if scenarios for various price schemes, given a reference period.	Passed

for various price schemes, given a reference period.		
2. The system should be able to display / present comparisons, based on prices, between Monte Carlo simulations.	The system is able to display / present comparisons, based on prices, between Monte Carlo simulations.	Passed

TC	**13.1**	**Visualising alerts and notifications – Accessing the inteGRIDy Visualization Platform & Services (IVP) layer**
UC	A13. Receive alerts on peak time	
FRs	PLO_EIIS_FR23; PLO_EIIS_FR23.1; PLO_EIIS_FR23.2	
Precondition	The software platform is operational.	
Test environment		
TC Step (actions)	**Obtained result**	**Verdict**
1. The user (Consumer) can login in the system and receive alerts and notifications through an alert component at the logical level of the system, given a reference period.	The authorized user (Consumer) successfully receives alerts and notifications through an alert component at the logical level of the system, given a reference period.	Passed
2. The user should be able to dismiss the alerts.	The user is able to dismiss the alerts.	Passed
3. The user should be able to define the alerts (conditions, time intervals, severity, messages).	The user is able to define the alerts (conditions, time intervals, severity, messages).	Passed

TC	**14.1**	**Proposing optimized consumption patterns – Accessing the inteGRIDy Decision Making & Optimization Mechanisms**
UC	A14. Receive advice for optimum consumptions	
FRs	PLO_EIIS_FR13; PLO_EIIS_FR13.3; PLO_EIIS_FR13.4; PLO_EIIS_FR13.5	
Precondition	The software platform is operational and has successfully stored historical data on consumption and prices for the given time interval.	
Test environment		

TC Step (actions)	Obtained result	Verdict
1. The system should be able to propose an optimized consumption pattern for one or more consumers.	The system successfully proposes optimized consumption patterns for one or more consumers.	
2. The system should be able to apply smart algorithms to get the optimal consumption in terms of economic savings.	The system successfully uses smart algorithms to get the optimal consumption such as to obtain economic savings (costs reduction).	Passed
3. The system should be able to propose an optimized consumption pattern that enables the user to reduce the energy bill.	The system successfully proposes an optimized consumption pattern that enables the user to reduce the energy bill.	Passed
4. The system should propose an optimized consumption pattern using different rates (CPP, TOU, PTR, CTP).	The system successfully proposes an optimized consumption pattern using different rates (CPP, TOU. PTR, CTP).	Passed

TC	**14.2**	**Visualising information about the bill optimization – Accessing the inteGRIDy Visualization Platform & Services (IVP) layer**
UC	A14. Receive advice for optimum consumptions	
FRs	PLO_EIIS_FR13.1; PLO_EIIS_FR13.2; PLO_EIIS_FR13.6	
Precondition	The software platform is operational and has successfully stored historical data on consumption and prices for the given time interval.	
Test environment		

TC Step (actions)	Obtained result	Verdict
1. The system should have a specific UI (User Interface) where the Consumer can ask for bill optimization.	The system provides a specific UI where the Consumer can ask for bill optimization.	Passed
2. The user (Consumer) can login in the system and request for the optimization of the bill.	The authorized user (Consumer) successfully access in the system the information needed for the optimization of the bill.	Passed

3. The system should be able to display the proposed consumption patterns in reports and charts.	The system successfully displays the proposed consumption patterns in Reports and Charts.	Passed

Functional testing results

TC	Verdict
Test case 1.1	Passed
Test case 1.2	Passed
Test case 1.3	Passed
Test case 1.4	Passed
Test case 2.1	Passed
Test case 2.2	Passed
Test case 2.3	Passed
Test case 2.4	Passed
Test case 3.1	Passed
Test case 3.2	Passed
Test case 4.1	Passed
Test case 4.2	Passed
Test case 4.3	Passed
Test case 5.1	Passed
Test case 5.2	Passed
Test case 6.1	Passed
Test case 6.2	Passed
Test case 6.3	Passed
Test case 6.4	Passed
Test case 6.5	Passed
Test case 7.1	Passed
Test case 7.2	Passed
test case 8.1	Passed
Test case 8.2	Passed
Test case 9.1	Passed
Test case 10.1	Passed
Test case 10.2	Passed
Test case 10.3	Passed
Test case 10.4	Passed

Test case 11.1	Passed
Test case 11.2	Passed
Test case 12.1	Passed
Test case 12.2	Passed
Test case 12.3	Passed
Test case 13.1	Passed
Test case 14.1	Passed
Test case 14.2	Passed

Selenium test

Multiple tests where done using Selenium:

1. ConsumerDataTest,
2. ConsumersInvoiceTest,
3. ConsumptionTest,
4. ForecastTest,
5. InvoiceOptimizationPeakTest,
6. InvoiceOptimizationTest,
7. OptimizationTest,
8. RecommendationTest,
9. WhatIfConsumerTest,
10. WhatIfScenarioTest
11. LoginControllerTest.

To run those tests we needed a configuration class, so we created TesteHelper.java class

```
import ...

/**
 * Created by alingr on 11/14/2019.
 */
public class TestHelper {
    public final static String  appURl = "http://localhost:8080/faces/login.xhtml";
    public final static String  user = "admin1";
    public final static String  password = "1234";
    public final static Integer  waitTime = 30;


    public static boolean clickOnWithRetry(By by, WebDriver driver, int retries) {
        boolean result = false;
        int attempts = 0;
        while(attempts < retries) {
            try {
                driver.findElement(by).click();
                result = true;
                break;
            } catch(StaleElementReferenceException | ElementClickInterceptedException e) {
                try {
                    Thread.sleep( millis: 50);
                } catch (InterruptedException e1) {
                    e1.printStackTrace();
                }
            }
            attempts++;
        }
        return result;
    }

}
```

It contains the details used to run the tests such as user, password and also a method used to simulate a user click, if the test worked the method will change parameter “result” value into true and will increase the number of “attempts” parameter, this means the click worked and the result will be displayed.

Every test has 3 main steps. BeforeAll, Test, AfterAll.

- **BeforeAll** consists of set-up method
- **Test** represents the get up method
- **AfterAll** represents the clean method

```
/**
 * Created by alinpr on 11/13/2019.
 */
class OptimizationTest {

    private static ChromeDriver driver;
    private static WebDriverWait wait;


    @BeforeAll
    static void setUp() {
        WebDriverManager.chromedriver().version("79.0.3945.117").setup();
        driver = new ChromeDriver();
        wait = new WebDriverWait(driver, timeOutInSeconds: 10);
        driver.get(TestHelper.appURL);
        driver.manage().window().maximize();
        driver.findElement(By.id("username")).sendKeys(TestHelper.user);
        driver.findElement(By.id("password")).sendKeys(TestHelper.password);
        TestHelper.clickOnWithRetry(By.id("loginBtn"),driver,TestHelper.waitTime);
    }

    @Test
    void getItems() {
        TestHelper.clickOnWithRetry(By.xpath("//span[.='Optimization']"),driver, retries: 10);
        TestHelper.clickOnWithRetry(By.id("integridy-navForm:calendarStartDate_input"),driver, retries: 10);
        while (!driver.findElement(By.className("ui-datepicker-year")).getText().equals("2018")) {
            TestHelper.clickOnWithRetry(By.xpath("//span[.='Previous']"),driver, retries: 10);
        }

        while (!(driver.findElement(By.className("ui-datepicker-month")).getText().equals("July"))) {
            TestHelper.clickOnWithRetry(By.xpath("//span[.='Previous']"),driver, retries: 10);
        }
        TestHelper.clickOnWithRetry(By.linkText("20"),driver, retries: 10);
        wait.until(ExpectedConditions.elementToBeClickable(By.id("integridy-navForm:applyFiltersBtn")));
        TestHelper.clickOnWithRetry(By.id("integridy-navForm:applyFiltersBtn"),driver, retries: 10);
        wait.until(ExpectedConditions.elementToBeClickable(By.id("canvas")));
        assertNotEquals(driver.findElements(By.id("canvas")).size(), actual: 0);
    }

    @AfterAll
    static void clean() { driver.quit(); }

}
```

BeforeAll step is the same to all tests. This step is a configuration method. This step consists of the configurations necessary to open the browser (minimized), connect to the web page (address URL) and log on with the specified user credentials.

```
@BeforeAll
static void setUp() {
    WebDriverManager.chromedriver().version("79.0.3945.117").setup();
    driver = new ChromeDriver();
    wait = new WebDriverWait(driver, timeOutInSeconds: 10);
    driver.get(TestHelper.appURL);
    driver.manage().window().maximize();
    driver.findElement(By.id("username")).sendKeys(TestHelper.user);
    driver.findElement(By.id("password")).sendKeys(TestHelper.password);
    TestHelper.clickOnWithRetry(By.id("loginBtn"),driver,TestHelper.waitTime);
```

AfterAll step is also the same to all tests. This script finishes the test by exiting the browser.

```
@AfterAll
static void clean() { driver.quit(); }
```

Test step. It contains the main test that is done over a section.

```
@Test
void login() {
    driver.get(TestHelper.appURL);
    driver.manage().window().maximize();
    driver.findElement(By.id("username")).sendKeys(TestHelper.user);
    driver.findElement(By.id("password")).sendKeys(TestHelper.password);
    TestHelper.clickOnWithRetry(By.id("loginBtn"),driver,TestHelper.waitTime);
    wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//span[.='admin1']")));
    assertNotEquals(driver.findElements(By.xpath("//span[.='admin1']")).size(), actual: 0);
}
```

<ConsumptionTest.java>

```
@Test
void getItems() {
    TestHelper.clickOnWithRetry(By.xpath("//span[.='Consumption']"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.id("integridy-navForm:calendarStartDate_input"),driver, retries: 10);
    while (!driver.findElement(By.className("ui-datepicker-year")).getText().equals("2018")) {
        TestHelper.clickOnWithRetry(By.xpath("//span[.='Previous']"),driver, retries: 10);
    }

    while (!(driver.findElement(By.className("ui-datepicker-month")).getText().equals("July"))) {
        TestHelper.clickOnWithRetry(By.xpath("//span[.='Previous']"),driver, retries: 10);
    }
    TestHelper.clickOnWithRetry(By.linkText("20"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.id("integridy-navForm:applyFiltersBtn"),driver, retries: 10);
    wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//tbody[@id='hConsumption:datalistCumulDsoH_data']/tr[1]/td/span")));
    assertNotEquals(driver.findElements(By.xpath("//tbody[@id='hConsumption:datalistCumulDsoH_data']/tr[1]/td/span")).size(), actual: 0);
    assertNotEquals(driver.findElements(By.xpath("//tbody[@id='hConsumption:datalistCumulDsoH_data']/tr[10]/td/span")).size(), actual: 0);
}
```

Goes on “Consumption” tab from the menu. Clicking the Previous button until the compiler reaches year 2018 than clicking again Previous button until it reaches July after that it selects the 20th day and clicks “Apply changes” and it waits until the data received is bigger than 0, if not it retries 10 times. After the getItems method is finished clean method is run.

<ConsumerDataTest.java>

```
@Test
void getItems() {
    TestHelper.clickOnWithRetry(By.xpath("//span[.='Consumers data']"),driver, retries: 10);
    wait.until(ExpectedConditions.elementToBeClickable(By.className("ui-selectonemenu-trigger")));
    TestHelper.clickOnWithRetry(By.className("ui-selectonemenu-trigger"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.cssSelector("li[data-label='consumer b 1']"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.id("integridy-navForm:calendarStartDate_input"),driver, retries: 10);
    while (!driver.findElement(By.className("ui-datepicker-year")).getText().equals("2018")) {
        TestHelper.clickOnWithRetry(By.xpath("//span[.='Previous']"),driver, retries: 10);
    }

    while (!(driver.findElement(By.className("ui-datepicker-month")).getText().equals("July"))) {
        TestHelper.clickOnWithRetry(By.xpath("//span[.='Previous']"),driver, retries: 10);
    }
    TestHelper.clickOnWithRetry(By.linkText("20"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.id("integridy-navForm:applyFiltersBtn"),driver, retries: 10);
    wait.until(ExpectedConditions.elementToBeClickable(By.id("ConsumptionRecordListForm:currentTotalChart")));
    assertNotEquals(driver.findElements(By.id("ConsumptionRecordListForm:currentTotalChart")).size(), actual: 0);
}
```

Navigates to “Consumer data” tab from the menu, then the script is selecting “consumer b1” from the dropdown menu. After selecting the consumer, it selects the date like in the last test (20 July 2018) then it applies the changes and it waits until the data received is bigger than 0, if not, it retries 10 times. After getItems method is finished, clean method is called.

<OptimizationTest.java>

```
@Test
void getItems() {
    TestHelper.clickOnWithRetry(By.xpath("//span[.='Optimization']"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.id("integridy-navForm:calendarStartDate_input"),driver, retries: 10);
    while (!driver.findElement(By.className("ui-datepicker-year")).getText().equals("2018")) {
        TestHelper.clickOnWithRetry(By.xpath("//span[.='Previous']"),driver, retries: 10);
    }

    while (!(driver.findElement(By.className("ui-datepicker-month")).getText().equals("July"))) {
        TestHelper.clickOnWithRetry(By.xpath("//span[.='Previous']"),driver, retries: 10);
    }
    TestHelper.clickOnWithRetry(By.linkText("20"),driver, retries: 10);
    wait.until(ExpectedConditions.elementToBeClickable(By.id("integridy-navForm:applyFiltersBtn")));
    TestHelper.clickOnWithRetry(By.id("integridy-navForm:applyFiltersBtn"),driver, retries: 10);
    wait.until(ExpectedConditions.elementToBeClickable(By.id("canvas")));
    assertNotEquals(driver.findElements(By.id("canvas")).size(), actual: 0);
}
```

Navigates to “Optimization” tab from the menu. Selects the date 20 July 2018. Then waits for the canvas diagram to be displayed. After getItems method finishes, clean method is called.

<InvoiceOptimizationPeakTest.java>

```
@Test
void getItems() {
    TestHelper.clickOnWithRetry(By.xpath("//span[.='Invoice optimization peak']"),driver, retries: 10);
    wait.until(ExpectedConditions.elementToBeClickable(By.className("ui-selectonemenu-trigger")));
    TestHelper.clickOnWithRetry(By.className("ui-selectonemenu-trigger"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.cssSelector("li[data-label='consumer b 1']"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.id("integridy-navForm:calendarStartDate_input"),driver, retries: 10);
    while (!driver.findElement(By.className("ui-datepicker-year")).getText().equals("2018")) {
        TestHelper.clickOnWithRetry(By.xpath("//span[.='Previous']"),driver, retries: 10);
    }

    while (!(driver.findElement(By.className("ui-datepicker-month")).getText().equals("July"))) {
        TestHelper.clickOnWithRetry(By.xpath("//span[.='Previous']"),driver, retries: 10);
    }
    TestHelper.clickOnWithRetry(By.linkText("20"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.id("integridy-navForm:applyFiltersBtn"),driver, retries: 10);

    wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//*[@id='userData:consumersList']/div[3]")));
    TestHelper.clickOnWithRetry(By.xpath("//*[@id='userData:consumersList']/div[3]"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.cssSelector("li[data-label='consumer b 1']"),driver, retries: 10);

    TestHelper.clickOnWithRetry(By.xpath("//*[@id='userData:detalistPrices_data']/tr[2]/td[1]/div/div[2]/span"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.id("userData:computeCosts"),driver, retries: 10);

    wait.until(ExpectedConditions.elementToBeClickable(By.className("jqplot-title")));
    assertNotEquals(driver.findElements(By.className("jqplot-title")).size(), actual: 0);
}
```

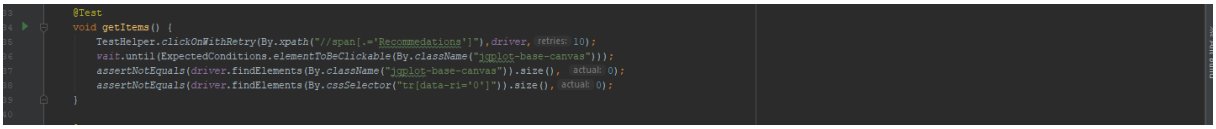
Navigates to “Invoice optimization peak” tab from the menu. Selects “consumer b1” from the consumers dropdown. Selects the date 20 July 2018 and applies changes. Selects the second checkbox from the list of 3 registrations and pushes the “Compute Costs” button. After that it expects the title of the chart to appear, if not it retries 10 times. After getItems method finishes, clean method is called.

<InvoiceOptimizationTest.java>

```
@Test
void getItems() {
    TestHelper.clickOnWithRetry(By.xpath("//span[.='Invoice optimization']"),driver, retries: 10);
    wait.until(ExpectedConditions.elementToBeClickable(By.className("ui-selectonemenu-trigger")));
    TestHelper.clickOnWithRetry(By.className("ui-selectonemenu-trigger"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.cssSelector("li[data-label='consumer b 1']"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.id("integridy-navForm:calendarStartDate_input"),driver, retries: 10);
    while (!driver.findElement(By.className("ui-datepicker-year")).getText().equals("2018")) {
        TestHelper.clickOnWithRetry(By.xpath("//span[.='Previous']"),driver, retries: 10);
    }

    while (!(driver.findElement(By.className("ui-datepicker-month")).getText().equals("July"))) {
        TestHelper.clickOnWithRetry(By.xpath("//span[.='Previous']"),driver, retries: 10);
    }
    TestHelper.clickOnWithRetry(By.linkText("20"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.id("integridy-navForm:applyFiltersBtn"),driver, retries: 10);

    wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//*[@id='userData:consumersList']/div[3]")));
    TestHelper.clickOnWithRetry(By.xpath("//*[@id='userData:consumersList']/div[3]"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.cssSelector("li[data-label='consumer b 1']"),driver, retries: 10);

    TestHelper.clickOnWithRetry(By.xpath("//*[@id='userData:datalistPrices_data']/tr[2]/td[1]/div/div[2]/span"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.id("userData:computeCosts"),driver, retries: 10);

    wait.until(ExpectedConditions.elementToBeClickable(By.id("ConsumptionRecordListForm:dataInvoiceDay")));
    assertNotEquals(driver.findElements(By.id("ConsumptionRecordListForm:dataInvoiceDay")).size(), actual: 0);
}
```

Navigates to “Invoice optimization” tab from the menu. Selects “consumer b1” from the dropdown list. Selects the date 20 July 2018 and applies changes. Selects second checkbox from the price table and then pushes “Compute costs” button. At the end it waits for the data to be displayed, if not it retries 10 times. After getItems method finishes, clean method is called.

<RecommendationTest.java>



```
@Test
void getItems() {
    TestHelper.clickOnWithRetry(By.xpath("//span[.='Recommedations']"),driver, retries: 10);
    wait.until(ExpectedConditions.elementToBeClickable(By.className("jqplot-base-canvas")));
    assertNotEquals(driver.findElements(By.className("jqplot-base-canvas")).size(), actual: 0);
    assertNotEquals(driver.findElements(By.cssSelector("tr[data-ri='0']")).size(), actual: 0);
}
```

Navigates to “Recommendations” tab from the menu. It waits for the tables to be displayed. After getItems method finishes, clean method is called.

<ConsumersInvoiceTest.java>

```
@Test
void getItems() {
    TestHelper.clickOnWithRetry(By.xpath("//span[.='Consumers invoice']"),driver, retries: 10);
    wait.until(ExpectedConditions.elementToBeClickable(By.className("ui-selectonemenu-trigger")));
    TestHelper.clickOnWithRetry(By.className("ui-selectonemenu-trigger"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.cssSelector("li[data-label='consumer b 1']"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.id("integridy-navForm:calendarStartDate_input"),driver, retries: 10);
    while (!driver.findElement(By.className("ui-datepicker-year")).getText().equals("2018")) {
        TestHelper.clickOnWithRetry(By.xpath("//span[.='Previous']"),driver, retries: 10);
    }

    while (!(driver.findElement(By.className("ui-datepicker-month")).getText().equals("July"))) {
        TestHelper.clickOnWithRetry(By.xpath("//span[.='Previous']"),driver, retries: 10);
    }
    TestHelper.clickOnWithRetry(By.linkText("20"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.id("integridy-navForm:applyFiltersBtn"),driver, retries: 10);

    wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//*[@id='userData:consumersList']/div[3]")));
    TestHelper.clickOnWithRetry(By.xpath("//*[@id='userData:consumersList']/div[3]"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.cssSelector("li[data-label='consumer b 1']"),driver, retries: 10);

    TestHelper.clickOnWithRetry(By.xpath("//*[@id='userData:datalistPrices_data']/tr[2]/td[1]/div/div[2]/span"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.id("userData:computeCosts"),driver, retries: 10);

    wait.until(ExpectedConditions.elementToBeClickable(By.id("ConsumptionRecordListForm:currentTotalCostDay")));
    assertNotEquals(driver.findElements(By.id("ConsumptionRecordListForm:currentTotalCostDay")).size(), actual: 0);
}
```

Navigates to “Consumer invoice” tab from the menu. The test sets the consumer to “consumer b 1 “from the dropdown menu. Sets the date to 20 July 2018. After setting the date it sets the second registration from the price table and pushes the “consumer cost” button, then it waits for the data to be shown. After getItems method finishes, clean method is called.

<ForecastTest.java>

```
@Test
void getItems() {
    TestHelper.clickOnWithRetry(By.xpath("//span[.='Forecast DSO']"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.id("integridy-navForm:calendarStartDate_input"),driver, retries: 10);
    while (!driver.findElement(By.className("ui-datepicker-year")).getText().equals("2018")) {
        TestHelper.clickOnWithRetry(By.xpath("//span[.='Previous']"),driver, retries: 10);
    }



    while (!(driver.findElement(By.className("ui-datepicker-month")).getText().equals("July"))) {
        TestHelper.clickOnWithRetry(By.xpath("//span[.='Previous']"),driver, retries: 10);
    }
    TestHelper.clickOnWithRetry(By.linkText("20"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.id("integridy-navForm:applyFiltersBtn"),driver, retries: 10);
    wait.until(ExpectedConditions.elementToBeClickable(By.id("canvas")));
    assertNotEquals(driver.findElements(By.id("canvas")).size(), actual: 0);

}
```

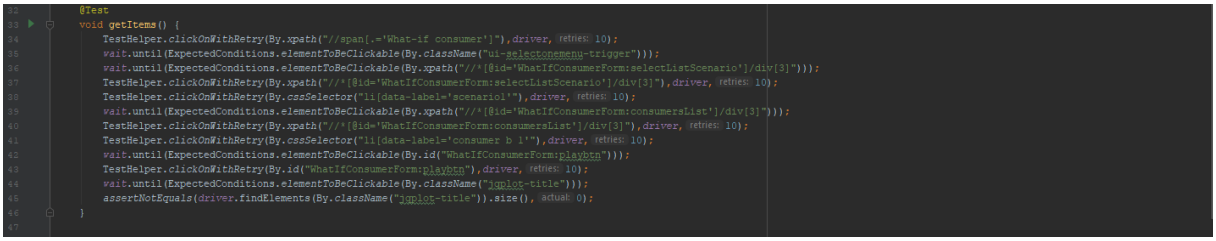
Navigates to “Forecast DSO” tab from the menu. Sets the date to 20 July 2018 and waits for the canvas to display. After getItems method finishes, clean method is called.

<WhatIfScenarioTest.java>

```
@Test
void getItems() {
    TestHelper.clickOnWithRetry(By.xpath("//span[.='What-if DSO']"),driver, retries: 10);
    wait.until(ExpectedConditions.elementToBeClickable(By.className("ui-selectonemenu-trigger")));
    TestHelper.clickOnWithRetry(By.className("ui-selectonemenu-trigger"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.cssSelector("li[data-label='scenario1']"),driver, retries: 10);
    wait.until(ExpectedConditions.elementToBeClickable(By.id("WhatIfForm:playbtn")));
    TestHelper.clickOnWithRetry(By.id("WhatIfForm:playbtn"),driver, retries: 10);
    wait.until(ExpectedConditions.elementToBeClickable(By.cssSelector("div[data-widget='widget_WhatIfForm_chartPanel']")));
    assertNotEquals(driver.findElements(By.cssSelector("div[data-widget='widget_WhatIfForm_chartPanel']")).size(), actual: 0);
}
```

Navigates to “What if DSO” after that the script sets the scenario to “scenario1” and pushes “play scenario” button. The script waits for the response to have the size bigger than 0, if not it retries 10 times. After getItems method finishes, clean method is called.

<WhatIfConsumerTest.java>



```
@Test
void getItems() {
    TestHelper.clickOnWithRetry(By.xpath("//span[.='What-if consumer']"),driver, retries: 10);
    wait.until(ExpectedConditions.elementToBeClickable(By.className("ui-selectonemenu-trigger")));
    wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//*[@id='WhatIfConsumerForm:selectListScenario']/div[3]")));
    TestHelper.clickOnWithRetry(By.xpath("//*[@id='WhatIfConsumerForm:selectListScenario']/div[3]"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.cssSelector("li[data-label='scenario1']"),driver, retries: 10);
    wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//*[@id='WhatIfConsumerForm:consumersList']/div[3]")));
    TestHelper.clickOnWithRetry(By.xpath("//*[@id='WhatIfConsumerForm:consumersList']/div[3]"),driver, retries: 10);
    TestHelper.clickOnWithRetry(By.cssSelector("li[data-label='consumer b 1']"),driver, retries: 10);
    wait.until(ExpectedConditions.elementToBeClickable(By.id("WhatIfConsumerForm:playbtn")));
    TestHelper.clickOnWithRetry(By.id("WhatIfConsumerForm:playbtn"),driver, retries: 10);
    wait.until(ExpectedConditions.elementToBeClickable(By.className("jqplot-title")));
    assertNotEquals(driver.findElements(By.className("jqplot-title")).size(), actual: 0);
}
```

Navigates to “What-if consumer”, after that the script sets the scenario to “scenario1” and the consumer to “consumer b 1” and pushes “play scenario” button. The script is expecting the response to have size bigger than 0, if not it retries 10 times. After getItems method finishes, clean method is called.

Selenium test results

All tests performed had positive results, running without any error during the process.

Thessaloniki

Test Case	[Test 01]	testing of Integrated Decision Support Supervisory System EMS tool overall performance and functionalities	
Related Use Cases	UC_02 (from D1.3)		
Functional Requirements	TH_ID&SSS_FR1, TH_ID&SSS_FR2, TH_ID&SSS_FR3, TH_ID&SSS_FR4		
Precondition	none		
Test environment (optional)			
Step (actions)	**Obtained result**		**Verdict**
1. Communication with SMPC tool	connection made. Desired data exchanged. Validated data consistency and based on the range set.		Passed after corrections
2. processing	Some data were out of expected bounds therefore a pre-processing of received data was implemented, to avoid having faulty data. Some bugs emerged when feeding with forced faulty data, which were further fixed during the debugging process.		Passed after corrections
3. Communication with DET	connection made. BESS schedule sent via MQTT Broker		Passed after corrections

Test Case	[Test 02]	testing of Plant/Process/System Data Exchange tool (DET) overall performance and functionalities	
Related Use Cases	UC_02 (from D1.3)		
Functional Requirements	TH_PPSDET_FR1, TH_PPSDET_FR2, TH_PPSDET_FR3, TH_PPSDET_FR4		
Precondition	none		
Test environment (optional)			
Step (actions)	**Obtained result**		**Verdict**
1. Communication with ID&SSS tool	connection made. Desired data exchanged through MQTT Broker Validated data consistency and based on the range set.		Passed after corrections
2. processing	Some data were out of expected bounds therefore a pre-processing of received data was implemented, to avoid having faulty data. Some bugs emerged when feeding with forced faulty data, which were further fixed during the debugging process.		Passed after corrections

3. Communication with BESS Inverters	connection made. Data Retrieved. Control Signals Sent. Validated data consistency and based on the range set.	Passed after corrections

Test Case	[Test 03]	testing of Supervisory Model Predictive Control for Energy Systems (MPC) overall performance and functionalities	
Related Use Cases	UC_02 (from D1.3)		
Functional Requirements	TH_SMPCFES_FR1, TH_SMPCFES_FR2		
Precondition	none		
Test environment (optional)			
Step (actions)	**Obtained result**		**Verdict**
1. Communication with ID&SSS tool	connection made. Desired data exchanged.		Passed after corrections
2. processing	Some data were out of expected bounds therefore a pre-processing of received data was implemented, to avoid having faulty data. Some bugs emerged when feeding with forced faulty data, which were further fixed during the debugging process.		Passed after corrections
2. Communication back to ID&SSS tool	connection made. Desired data exchanged.		Passed after corrections

Test Case	[Test 04]	testing of Visual Analytics Tool for Flexibility Analysis, Aggregation and Forecasting overall performance and functionalities	
Related Use Cases	UC_01, UC_02, UC_03 (from D1.3)		
Functional Requirements	TH_VATFFAFF_FR1, TH_VATFFAFF_FR2, TH_VATFFAFF_FR3, TH_VATFFAFF_FR4, TH_VATFFAFF_FR5, TH_VATFFAFF_FR6, TH_VATFFAFF_FR7, TH_VATFFAFF_FR8, TH_VATFFAFF_FR9		
Precondition	none		
Test environment (optional)			
Step (actions)	**Obtained result**		**Verdict**

1. Retrieve historical data from Reference Knowledge Warehouse (RKW)	If internet access is active, desired data retrieved. Validated data consistency and based on the range set.	Passed after corrections
2. Process data	Some data were out of expected bounds therefore a pre-processing of received data was implemented, to avoid having faulty data. Some bugs emerged when feeding with forced faulty data, which were further fixed during the debugging process.	Passed after corrections
3. Visualize data	Problems detected when displaying in another monitor due to fixed ratio monitor display were resolved. Some graphs were not shown correctly when fed with too many data, solved by setting new rules in post-processing of data to be displayed. A slight delay was noted on the refresh rate, when updating new values coming from an occupancy change. By adopting a shorter refresh rate and using cashing mechanisms the latency was reduced and was considered within acceptable boundaries.	Passed after corrections
4. Expect input from end-user	Testing iteratively to log in with wrong credentials. Failure to do so. Thus, Authentication/Authorization procedures operate successfully. No other issues so far.	Passed
5. Store input from end-user to RKW	If internet access is active, desired data retrieved and stored. Validated that data were successfully stored through several iterations and with different test data sent.	Passed after corrections
6. Send input from end-user to relative tools	If internet access is active, desired data retrieved and sent. Validated communication by iterative testing with input sent.	Passed after corrections

Test Case	[Test 05]	testing of Intelligent Building Control & Flexibility Prediction Forecasting tool overall performance and functionalities
Related Use Cases	UC_01, UC_02, UC_03 (from D1.3)	
Functional Requirements	TH_IBC&FPF_FR1, TH_IBC&FPF_FR2, TH_IBC&FPF_FR3, TH_IBC&FPF_FR4, TH_IBC&FPF_FR5, TH_IBC&FPF_FR6, TH_IBC&FPF_FR7, TH_IBC&FPF_FR8, TH_IBC&FPF_FR9, TH_IBC&FPF_FR10	
Precondition	none	

Test environment (optional)		
Step (actions)	**Obtained result**	**Verdict**
1. Retrieve historical data from Reference Knowledge Warehouse (RKW)	If internet access is active, desired data retrieved. Validated data consistency and based on the range set.	Passed after corrections
2. Retrieve data from a weather API	If internet access is active, desired data retrieved. Validated data consistency and based on the range set.	Passed after corrections
3. Retrieve data from energy price API	If internet access is active, desired data retrieved. Validated data consistency and based on the range set.	Passed after corrections
4. Retrieve real-time measurements from field devices (BESS - energy meters)	If internet access is active, desired data retrieved. Validated data consistency and based on the range set.	Passed after corrections
5. process data for forecasting	Some data were out of expected bounds therefore a pre-processing of received data was implemented, to avoid having faulty data. Some bugs emerged when feeding with forced faulty data, which were further fixed during the debugging process.	Passed after corrections
6. process data for optimal BESS Schedule	Some data were out of expected bounds therefore a pre-processing of received data was implemented, to avoid having faulty data. Some bugs emerged when feeding with forced faulty data, which were further fixed during the debugging process.	Passed after corrections
7. process data for optimal DR schedule	Some data were out of expected bounds therefore a pre-processing of received data was implemented, to avoid having faulty data. Some bugs emerged when feeding with forced faulty data, which were further fixed during the debugging process.	Passed after corrections
8. connect with RKW to store optimal results	If internet access is active, desired data are stored. Validated that data were successfully stored through several iterations and with different test data sent.	Passed after corrections
9. connect with BESS via other tools to monitor its operation	If internet access is active, desired data retrieved. Validated data consistency and based on the range set.	Passed after corrections

10. connect with BESS via other tools to send setpoints	If internet access is active, desired data are sent. Validated data consistency and based on the range set.	Passed after corrections

Test Case	[Test 06]	testing of Building Occupancy & Energy Consumption Simulation Tool overall performance and functionalities	
Related Use Cases	UC_01, UC_02, UC_03 (from D1.3)		
Functional Requirements	TH_BO&ECST_FR1, TH_BO&ECST_FR2, TH_BO&ECST_FR3		
Precondition	none		
Test environment (optional)			
Step (actions)	**Obtained result**		**Verdict**
1. Retrieve historical data from Reference Knowledge Warehouse (RKW)	If internet access is active, desired data retrieved. Validated data consistency and based on the range set.		Passed after corrections
2. process data and conduct simulations	Some data were out of expected bounds therefore a pre-processing of received data was implemented, to avoid having faulty data. Some bugs emerged when feeding with forced faulty data, which were further fixed during the debugging process.		Passed after corrections
3. connect with RKW to store simulation results	If internet access is active, desired data are stored. Validated that data were successfully stored through several iterations and with different test data sent.		Passed after corrections

Test Case	[Test 07]	testing of Facility Management Web-based Interface overall performance and functionalities	
Related Use Cases	UC_03 (from D1.3)		
Functional Requirements	TH_FMWBI_FR_1, TH_FMWBI_FR_2, TH_FMWBI_FR_3, TH_FMWBI_FR_4new		
Precondition	none		
Test environment (optional)			
Step (actions)	**Obtained result**		**Verdict**

1. Retrieve historical data from Reference Knowledge Warehouse (RKW)	If internet access is active, desired data retrieved. Validated data consistency and based on the range set.	Passed after corrections
2. process data	Some data were out of expected bounds therefore a pre-processing of received data was implemented, to avoid having faulty data. Some bugs emerged when feeding with forced faulty data, which were further fixed during the debugging process.	Passed after corrections
3. Visualize data	Problems detected when displaying in another monitor due to fixed ratio monitor display were resolved. Some graphs were not shown correctly when fed with too many data, solved by setting new rules in post-processing of data to be displayed. A slight delay was noted on the refresh rate, when updating new values coming from an occupancy change. By adopting a shorter refresh rate and using cashing mechanisms the latency was reduced and was considered within acceptable boundaries.	Passed after corrections
4. expect input front end-user	Testing iteratively to log in with wrong credentials. Failure to do so. Thus, Authentication/Authorization procedures operate successfully. No other issues so far.	Passed
5. Store input from end-user to RKW	If internet access is active, desired data retrieved and stored. Validated that data were successfully stored through several iterations and with different test data sent.	Passed after corrections
6. Send input from end-user to relative tools	If internet access is active, desired data retrieved and sent. Validated communication by iterative testing with input sent.	Passed after corrections
7. Send input to field devices	If internet access is active, desired data retrieved and sent. Validated communication by iterative testing with input sent.	Passed after corrections

http://www.integridy.eu