

PROGRAMME: H2020-LC-SC3-2020-EC-ES-SCC

START OF PROJECT: 01.10.2020

DURATION: 60 MONTHS

DELIVERABLE 6.1:

INTEGRATION TEST PLAN

Authors: Denys Díaz, Alejandro Chito (AYE), Alejandro Trujillo (IDE), Carlos Barco (ZIG)

Due date of deliverable: 31.12.2022

Actual submission date: 05.05.2023

Deliverable Name	Integration test plan
Deliverable Number	D 6.1
Work Package	WP 6
Associated Task	T 6.1
Covered Period	M21-M29
Due Date	31.12.2022
Completion Date	05.05.2023
Submission Date	05.05.2023
Deliverable Lead Partner	AYE
Deliverable Authors	AYE, IDE, ZIG
Version	3.0

DISSEMINATION LEVEL

PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Change Control

DOCUMENT HISTORY

Version	Date	Change History	Author(s)	Organisation
1.0	03.02.2023	Table of content drafted	Denys Díaz	AYE
2.0	07.02.2023	Document drafted	Denys Díaz, Alejandro Chito (AYE), Alejandro Trujillo (IDE), Carlos Barco (ZIG)	AYE, IDE, ZIG
3.0	05.05.2023	Last version	Denys Díaz	AYE

DISTRIBUTION LIST

Date	Issue	Group
07.02.2023	Revision table of content	Task leader and Task 6.1 partners
25.04.2023	Revision document	IDE
05.05.2023	Submission and distribution to partners	PO + All partners

TABLE OF CONTENT

Deliverable 6.1:	1
Integration test plan	1
Table of Content.....	3
Abbreviations.....	5
Executive summary.....	6
1 Introduction	7
Scope and deliverable structure.....	7
2 Integration Infrastructure	7
2.1 Architecture.....	8
2.2 Use cases.....	9
2.2.1 Use Case 1: RESS – Household.....	9
2.2.2 Use Case 2: RESS – Building.....	10
2.2.3 Use Case 3: Seawater district heating.....	10
2.2.4 Use Case 4: EV charging station.....	11
2.2.5 Use Case 5: HESS Large scale. UCAP+Li-ion storage.....	11
2.2.6 Use Case 6: Hydrogen-based storage.....	12
2.2.7 Use Case 7: Street lighting system.....	13
2.3 Device interconnection.....	14
2.4 Integration Requirements.....	17
Network Requirements.....	17
3 Integration Environment	18
3.1 Smart IT platform environment.....	18
3.2 Technical Specifications.....	19
3.2.1 Version control.....	19
3.2.2 Continuous integration.....	19
3.2.3 Monitoring.....	20
3.2.4 Data analytics.....	20
3.2.5 Security.....	20
3.2.6 Flexibility.....	20
3.2.7 Testing.....	21
3.3 Gridpilot interface.....	21
4 Implementation and integration plan	22
4.1 Overview of the integration plan.....	22
4.2 Integration phases.....	23
4.2.1 Phase 1. Pre-installation test.....	23
4.2.2 Phase 2. Installed component test.....	26
4.2.3 Phase 3. Full integration.....	29
5 System and integration testing	29
5.1 Integration testing.....	30
5.1.1 Test execution 1: RESS – Household.....	30

5.1.2	Test execution 2: RESS – Building	32
5.1.3	Test execution 3: Seawater district heating	34
5.1.4	Test execution 4: EV charging station	34
5.1.5	Test execution 5: HESS Large scale. UCAP+Li-ion storage	38
5.1.6	Test execution 6: Hydrogen-based storage.....	41
5.1.7	Test execution 7: Street lighting system.....	42
6	Main Conclusions.....	44
	Deviations.....	45
7	References.....	45

ABBREVIATIONS

List of abbreviations

AYE	AYESA
CEG	CEGASA
DER	Distributed Energy Resources
EV	Electrical Vehicle
HES	Hybrid Energy Storage Systems
HTTP	Hypertext Transfer Protocol
IDE	IDENER
IT	Information Technology
IoT	Internet of Things
KPI	Key Performance Indicator
MQTT	Message Queuing Telemetry Transport
NBG	Nordseeheilbad Borkum GmbH
OCPP	Open Charge Point Protocol
POC	Proof of Concept
RESS	Renewable Energy Service System
SSH	Secure Shell
UCAP	Ultracapacitor
UI	User Interface
VPN	Virtual Private Network
ZIG	Zigor

EXECUTIVE SUMMARY

The present deliverable presents the integration test plan to be performed on the island of Borkum (Germany) as a part of the European project ISLANDER. This project aims to increase the decarbonisation of the island by using smart grid solutions combining renewable energy production with storage technologies in real-life settings, monitored by a smart IT platform.

Before installing the equipment on the island, it is recommended to ensure that all devices can be integrated and appropriately interconnected. To complete the integration and monitoring of all energy assets, it is necessary to check if the interconnection of all of them with the smart IT platform can be achieved. Since the goal of this platform is to monitor, manage and coordinate each device considering the status of the power grid.

This deliverable is included in task T6.1 of the ISLANDER project. As specified in the Grant agreement of this project (Project ISLANDER, 2020): *“The motivation of this task is to ensure that all equipment and systems involved are integrated optimally and reliably in the smart IT platform. To do so, several items will be considered: Elements to be integrated; Integration strategy; and Sequence of feature/function and hardware/software integration (including software integration sequence, hardware integration sequence, full hardware/software integration sequence and subsystem integration sequence).”*

Several chapters have been included in this document to complete the goal mentioned. The main idea is to give a broad perspective of the problem and provide a short outline of the plan for the implementation and integration of all components that must be installed on Borkum. For that reason, this document includes an explanation of each of the points that make up the complete integration plan with all the energy assets explained by each of the partners in charge of carrying out the installation and integration of the equipment.

1 INTRODUCTION

Scope and deliverable structure

The ISLANDER project aims to analyse the performance of various energy systems in order to determine the necessary infrastructure to help decarbonise the island of Borkum. The size of each of the systems to be installed in this project, as well as their intended use and analysis, means that they can be referred to as pilot systems. One of the main points to be achieved is to coordinate the use of all the energy systems through a computer platform in such a way that all the devices work together as if they were a single system. The main objective of this document is to define the integration guide for all the devices that will be installed on the island, and which will be connected to the smart IT platform developed by AYESA and called GridPilot. The idea is to have an overview of the architecture and the integration tests to be carried out in order to establish the steps to be followed and check the correct integration of the systems. To this end, a description of the integration process of each of the components will be included, and if necessary, their dependence on the other devices within the same pilot. The following is an overview of the content of this document to give the reader a better understanding.

This deliverable is structured in several chapters which offer an explanation of each of the steps to be taken to complete the integration. The first chapter called Introduction tries to give an overview of the subject matter of the document. This is followed by Chapter 2, Integration Infrastructure, which includes an overview of the integration architecture consisting of the pilots and the IT platform. The next section, Integration Environment, details the environment that makes up the IT platform, including technical specifications, interface, and user roles. The fourth chapter, Implementation and integration plan, shows the tasks to be developed in each phase of the integration plan as well as the communication flows that exist between the devices. This point is followed by chapter 5, System and integration testing, which gives an explanation of the tests carried out in each of the pilots. Finally, and to complete this document, the final conclusions obtained in this part of the project are given.

2 INTEGRATION INFRASTRUCTURE

This chapter presents the interaction of each energy asset installed in Borkum by various project partners (CEG, IDE, NBG, ZIG and AYE) with the IT platform. The design of the infrastructure depends on the components of each pilot connected to the platform. For that

reason, the approach taken in this section provides a perspective that helps to have a better understanding of the integration and testing of each component, its relationship with other devices and the information flow with the platform.

2.1 Architecture

This section depicts an overview representation of the architecture, defining how each of the energy systems is integrated into the computing platform developed by AYESA. Such a representation of the architecture is crucial to understanding the information flow between the devices and detecting potential errors more quickly.

It is important to highlight that not all energy assets in this project will be directly connected to the same cloud. In fact, as can be seen below in Figure 1, two virtual clouds have been developed to manage the information provided by the different systems and allow data storage.

The first of these clouds is developed by one of the project partners, IDENER. This cloud will manage the automatic connection and bi-directional data flow between the systems and the AYESA platform. Specifically, the energy systems connected to this cloud are the following:

- Hydrogen-based storage system.
- 30 RESS household solutions.

Given that this partner is responsible for installing and commissioning these systems, developing an intermediary platform will ease the communication management and allow it to perform the tests set out throughout this deliverable.

The second of the virtual clouds is developed by AYESA. This cloud encompasses the connections of the components of the rest of the pilots installed on the island:

- Ultracaps and Li-ion batteries.
- 5 EV charging stations.
- Seawater district heating network + heat storage.
- Street lighting network.
- 3 RESS building solutions.

Figure 1 depicts pilot interactions, information flow, and the components connected to the platform that are located within these two clouds. Some of the pilots contains more than one software and device that interact with each other inside the same platform.

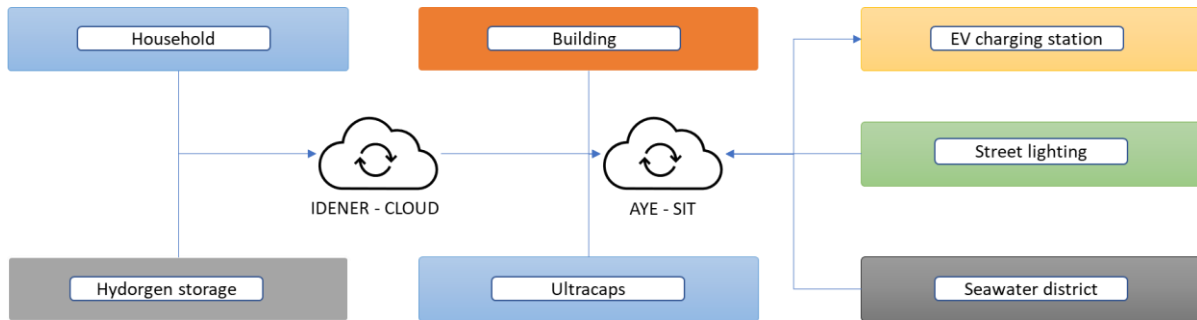


Figure 1 – Overview of integration architecture

2.2 Use cases

This section aims to present the interactions of the energy assets required to support each use case on the ISLANDER project and define the test cases specifically designed to validate that the different energy systems are correctly integrating with the smart IT platform.

2.2.1 Use Case 1: RESS – Household

The RESS Household solution encompasses a total of 30 residential households. Each of these systems is connected to IDENER's cloud, which, in turn, is connected to AYESA's SIT platform. The purpose of IDENER's cloud is to act as an intermediary between the SIT and the systems, managing the communication channels and the messages. All relevant telemetries will be systematically retrieved by IDENER's cloud and appropriately sent to the SIT platform. Additionally, whenever there is a setpoint given by the SIT platform for the battery, i.e., an order of charge or discharge, IDENER's cloud will process such an order, sending it in the correct format to the battery inverter.

The control device selected to perform the aforementioned tasks is the Raspberry Pi 4B, which is physically connected to the battery inverter and the user's router. The connectivity and information characteristics of the systems are as follows:

- Communication between the Raspberry Pi 4B and the battery inverter will be done via Ethernet cable using the Modbus protocol
- Communication between the Raspberry Pi 4B and IDENER's cloud will be done via the house router, using the MQTT protocol for telemetry sending and setpoint receiving.
- Communication between IDENER's cloud and AYESA's IT platform will be done using messaging queue system (RabbitMQ), also using the MQTT protocol.

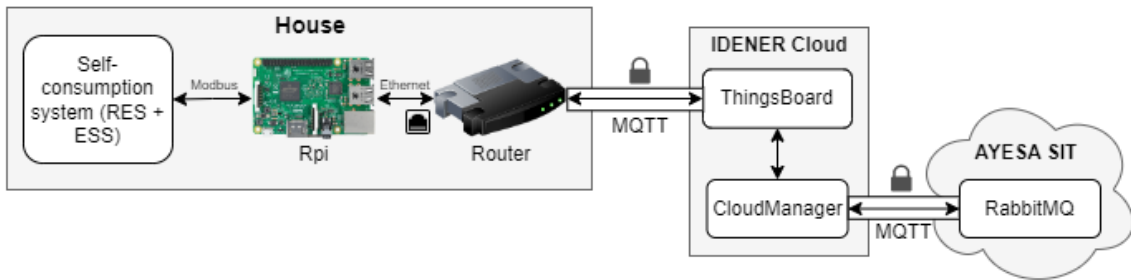


Figure 2 - Overview of RESS Household solution interconnection to the IT platform

2.2.2 Use Case 2: RESS – Building

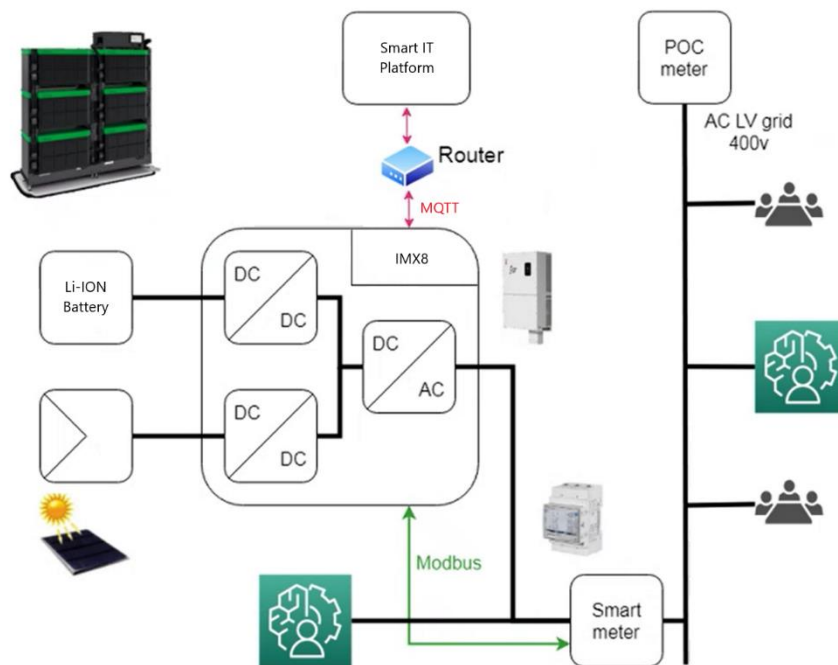


Figure 3 - Overview of Storage Management Systems interconnection to the IT platform

Communication between Ayesa’s control system (Smart IT Platform) and the converter for RESS solution is done through MQTT protocol. For this purpose, the communications board of the system has a SOM that is, specifically, an IMX8 processor.

For measuring the conditions of the grid, there is a specific device represented as Smart meter that provides information to the converter through Modbus TCP protocol.

This way, all the instructions of operation for the converter are directly provided from the Smart IT Platform.

2.2.3 Use Case 3: Seawater district heating

The seawater district heating system has suffered delays in its development due to supply

problems and price increases in the materials needed for its construction. In other words, many of its components are at the beginning of their development and, as such, this use case and the related tests cannot be analysed at the time of writing this deliverable. However, this information can be included in later deliverables that depend on the results of these tests so that the reader can analyse in depth all the development carried out.

2.2.4 Use Case 4: EV charging station

This use case includes a total of five electric vehicle chargers. The aim is to connect them to the IT platform and coordinate their operation, sending the respective orders for charging and discharging to the users' electric vehicles.

The model selected for installation in Borkum is called Quasar 2 and belongs to the Spanish company Wallbox (Wallbox, 2022). Following the datasheet information provided by the supplier, the connectivity and identification characteristics are as follows:

- Connectivity: this model of charger allows connection to the platform through diverse ways: Wi-Fi / Ethernet / Bluetooth / 3G/4G.
- Communication protocol: currently, the communication protocol used by this type of device is OCPP.

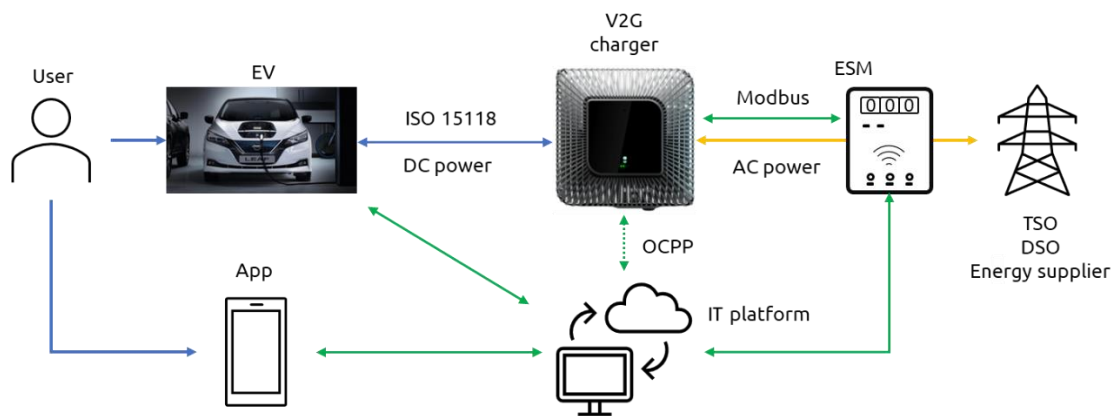


Figure 4 – Overview of EV charger interconnection to the IT platform

2.2.5 Use Case 5: HESS Large scale. UCAP+Li-ion storage

This use case includes two different storage management systems. A fast, responsive ultracaps storage management system and a Li-ion battery storage management system.

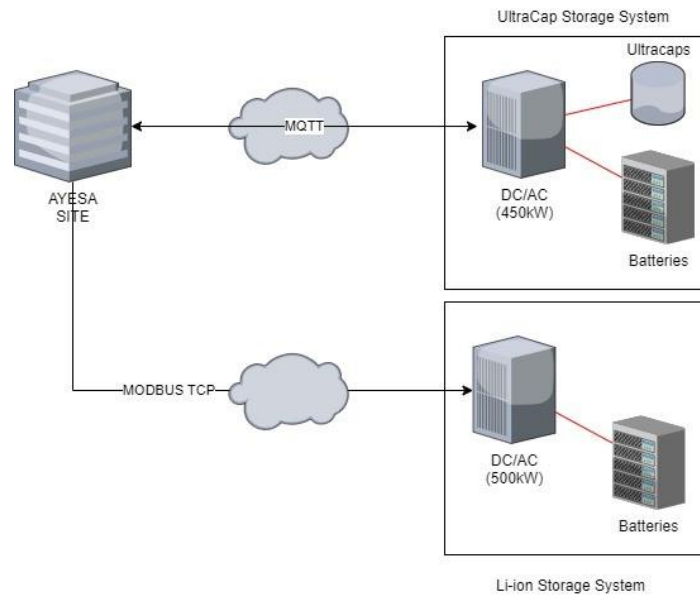


Figure 5 – Overview of Storage Management Systems interconnection to the IT platform

Communication between Ayesa’s control system and the ultracaps storage management system is done through the MQTT protocol, whilst communication with the Li-ion storage management system is done through the Modbus TCP protocol.

2.2.6 Use Case 6: Hydrogen-based storage

This use case is based on a hydrogen-based storage system for seasonal energy storage purposes. The connection of the system to AYESA’s IT platform is analogous to the RESS Household solution, i.e., the hydrogen system is connected to the IDENER’s cloud, which is then connected to the aforesaid SIT platform. The objective of this layout is the same as it was for the RESS Household solution in terms of telemetries and setpoints.

The control device selected for this system is also the Raspberry Pi 4B, which is connected to an external and an internal router. The former acts as a gateway to IDENER’s cloud, and the latter (included in the containerised solution) is responsible for communicating with the different components depending on the circumstances. It is also worth mentioning that the system is also connected to SOLENCO’s cloud to safeguard the correct operation of the system and offer an additional layer of security.

The connectivity and information characteristics of the systems are as follows:

- Communication between the Raspberry Pi 4B and the internal router will be done via Ethernet cable using the Modbus protocol.
- Communication between the Raspberry Pi 4B and IDENER’s cloud via external router, using MQTT protocol.
- Communication between IDENER’s cloud and AYESA’s IT platform using a messaging

queue.

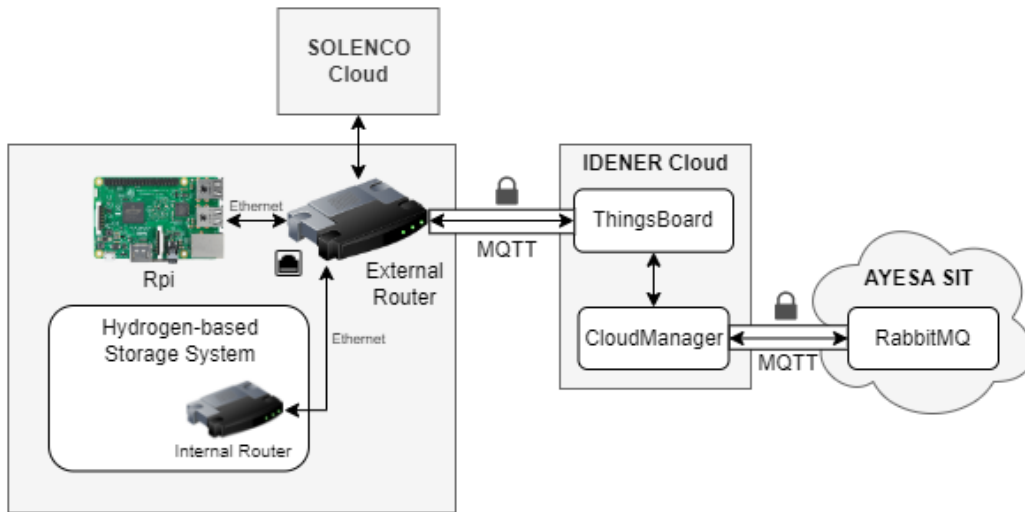


Figure 6 – Overview of the hydrogen-based storage system interconnection to the IT platform

2.2.7 Use Case 7: Street lighting system

Of all the energy systems to be installed on the island, the street lighting system has the most straightforward interconnection with the platform. The reason for this is that this pilot is connected through a smaller number of components. As can be seen in Figure 7, the IT platform acts in an analogous way to the previous pilots. It first establishes a connection to a device with a remote signal (a router or a Raspberry) via a VPN. After making the connection, the Raspberry or router enables communication between the platform and the second element called HomeLink. This device is the link between the platform and the lighting system, sending commands directly to the lighting system and ensuring that the on and off orders are carried out.

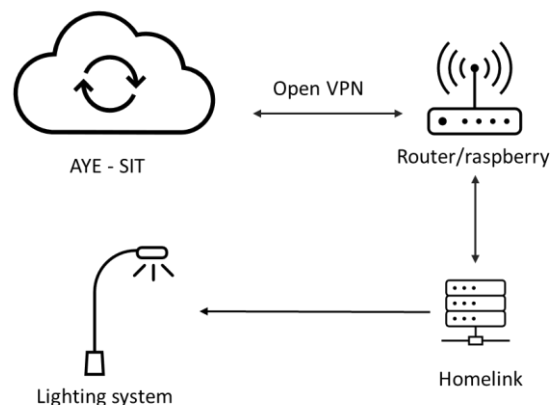


Figure 7 – Overview of street lighting system interconnection to the IT platform

2.3 Device interconnection

With all the above information, it is possible to have an overview of all interactions within the integration infrastructure. The level of difficulty depends on the degree of dependency of the devices, the installed software, the communication protocols, and the requirements or intended use.

For the sake of completeness, Table 1 encompasses all interactions between all single components explained in the previous systems. Additionally, the information flow is shown, indicating its direction and whether it is unidirectional or bidirectional.

2.4 Integration Requirements

The integration requirements of the ISLANDER project encompass all the actions necessary to achieve the interconnection of multiple distributed energy resources (DER) and hybrid energy storage systems (HES) with the smart computing platform. In this aspect, the partners included in this document (CEG, ZIG, NBG, IDE and AYE) have worked jointly and in a coordinated manner to establish the technology required to enable the seamless bi-directional flow of information.

In this context, the integration plan of the ISLANDER project has different aspects that need to be fulfilled. The first of these is to ensure the proper functioning of the IT platform and the different components of each energy system, including the DER and HES systems, by correctly designing them to be interoperable with each other. This implies the usage of standard communication protocols and interfaces to enable seamless data exchange and control. The second relates to the collection, processing, and management of large volumes of data from energy systems to be done by the platform. This requires advanced tools for data management and analysis to make real-time decisions considering the state of the equipment and the power grid. The third aspect encompasses the control and automation of the different components of each energy system, ensuring their optimal operation using intelligent algorithms. The last of these aspects relates to privacy and the handling of sensitive data. The integration plan set out in the present deliverable ensures full compliance with relevant regulations.

Ultimately, the requirements of the ISLANDER project integration plan entail the integration of multiple components of various energy systems into a single intelligent IT platform in a cohesive manner to manage energy supply and demand on the island in an efficient and sustainable way.

Network Requirements

Coordinating several pilots with different objectives and functionalities can be quite challenging when developing a global solution for all of them. In today's market, there are several options to overcome this issue and allow remote connection and management through intelligent platforms. One of them can be remote access protocols such as the one provided by Microsoft (RDP) or the Secure Shell (SSH) protocol. Other systems allow components to be remotely and centrally managed and monitored by management software such as System Center Configuration Manager or JamfPro. IoT (Internet of Things) platforms are the third group of systems that enable the connection and management of devices

through data collection and analysis, process automation and decision-making. Examples of these platforms are Amazon's AWS IoT and Microsoft's Azure IoT.

As the ISLANDER project has progressed, the different systems on the market have been assessed by trial-and-error methods. One of the criteria is to use the same technology for as many elements as possible. In this way, the aim is to ensure a homogeneous and global system and methodology for sending information. After analysing various connection options available on the market, it was decided to opt for the solution described below.

VPN (Virtual Private Network) is a software or set of applications that allows to establish a secure and proven connection between remote devices and an intelligent platform. This technology is used to create a private network over the internet through an encrypted communication tunnel between the devices and the platform, guaranteeing the privacy and security of the data transmitted. To access the network, private access credentials are required. This makes it a secure, reliable, and accessible system that guarantees the privacy and security of the data transmitted. Another important feature is that it allows the privacy of the users to be protected by hiding their IP address and geographical location, which increases the anonymity of the participants in the project.

3 INTEGRATION ENVIRONMENT

This chapter includes a detailed description of the smart IT platform environment, which will serve as the central location for accessing software components and their functionalities. It also covers the development details of the integration environment, as well as user management and technical specifications, with an architectural view of the smart IT platform environment.

The choice of the best integration environment for an IT platform will depend on the specific needs of the project and the technological infrastructure available.

3.1 Smart IT platform environment

Gridpilot is an IT platform that emulates a desktop environment within a single browser window. Its main advantage is that it allows the integration of several pilots in one place, allowing users to access developed applications from any computer easily. In addition, this approach eliminates the need for other software installations. As all application-related information is loaded and displayed in the user's browser, this allows for centrally performed updates.

Integrating all ISLANDER pilots into Gridpilot enables features such as version control, continuous integration, monitoring and data analysis. The following table shows each of the tools used by the platform, the features, and a brief description of their purpose.

Table 2 – Smart IT platform environments

Tool	Propose	Description
GitLab	Version control	Version control of source code and platform versions
Jenkins, JUnit, Selenium	Continuous integration	Perform integration testing on a continuous basis and detect potential errors as code changes occur
MQTT, API-REST, Java, Python	Monitoring	Monitor the performance and availability of energy assets
Python, CBC	Data analytics	Process and analyse large amounts of data
HTTPS, VPN, LDAP	Security	Establish security measures to protect data and platform infrastructure
Python, Java	Flexibility	Allow integration of different technologies, programming languages and tools
JUnit, Selenium	Testing	Implement testing to ensure software quality and interoperability between the different components of the platform

3.2 Technical Specifications

The technology of Gridpilot consists of the following core technologies.

3.2.1 Version control

The version control technology used in the Gridpilot platform is GitLab (Gitlab, 2023). This software allows the Gridpilot developers to maintain code control, working together in the same code repository and releasing versions in an agile way.

3.2.2 Continuous integration

The continuous integration tool used in Gridpilot is Jenkins. This tool allows the Gridpilot developer team to deploy the code created to the several functionalities the Islander project need in different environments isolated among them. Currently, Gridpilot uses two environments: Preproduction and Production. Before the code is deployed in Production

mode, it must be tested in Preproduction mode first to ensure its correct behaviour. In order to ensure this, the Gridpilot development team support the JUNIT (JUnit5, 2023) and Selenium (Selenium, 2023) tools to guarantee the quality of the code deployed.

3.2.3 Monitoring

Every use case defined in the ISLANDER project (Household, Building, EV, District Heating, Hydrogen and Street lighting) utilises different types of electrical devices (although some of them are the same). Accordingly, Gridpilot will appropriately address the required access methods or formats for making requests and sending orders for each of them.

At the time of writing this document, it was decided to use the MQTT protocol (MQTT, 2023) and invocation API-REST (Wikipedia, 2023). These will be the languages in which Gridpilot will establish continuous communication with the numerous electrical devices from each energy system. However, Gridpilot also uses other languages to process and manage the data obtained. Mainly, these languages are JAVA (Oracle, 2023) and Python (Python, 2023).

3.2.4 Data analytics

All energy assets provide a large amount of data that is going to be stored by the smart IT platform. To check each device's availability and proper operation, a technology that allows the data analytics gathered must be integrated into the system. For the ISLANDER project, two technologies have been chosen: Python for data management and Echart (ECharts, 2023) for data visualization.

3.2.5 Security

The security between the electrical devices from the different use cases and the Gridpilot platform will be entrusted to HTTPS protocol and VPN connections. The connections will be constantly secured by a private and public certificate, ensuring the proper and secure connection with the devices. In parallel, Gridpilot will offer a VPN connection to be used by the ISLANDER use cases, providing specific credentials to each use case.

Furthermore, from a sign-in point of view, every user will have private credentials to connect with Gridpilot's UI, which will be updated and set by the own user. The private user data will be stored in an LDAP database.

3.2.6 Flexibility

The data provided by the use cases will be stored in a non-relational database with a common data model regardless of the source or origin of such data. In other words, the data will be requested and provided only with this format, being the common point between the different functionalities of the platform.

Additionally, thanks to the distributed nature of the Gridpilot platform, different code programming languages may be working among them simultaneously. Every functionality could be implemented in a specific language without needing to know the language from other functionalities by leveraging the aforementioned common data model.

Mainly, the code programming languages are JAVA and Python.

3.2.7 Testing

To ensure the software quality and interoperability, the Gridpilot development team supports the JUNIT and Selenium tools to guarantee the quality of the code deployed, establishing a minimum standard to release a version.

3.3 Gridpilot interface

In order to be able to preview the final result, the most relevant information of the ISLANDER IT platform interface will be shown here. The Gridpilot interface is an essential tool for users to interact with the platform and manage energy resources efficiently. The design of the platform will be shown in more detail in other deliverables and will be improved as the project develops and meets the needs. The most important parts are:

1. Home screen of the platform. Through this first screen, users will be able to access Gridpilot using the credentials provided by AYESA. In addition, the ISLANDER integrated solution provides users with access to specific tools and applications based on well-defined roles. This is accomplished through a role-based security model that assigns one or more roles to each ISLANDER stakeholder, dictating their level of access, modification, and execution privileges.



Figure 8 – Credentials page of Gridpilot

2. Dashboard. The energy monitor allows users to view the island in real time. In addition, they can also access detailed information on the energy production of each distributed energy resource connected to the platform.



Figure 9 – Dashboard example

4 IMPLEMENTATION AND INTEGRATION PLAN

4.1 Overview of the integration plan

One of the main issues addressed in this document is ensuring that communication between the devices installed on the island and the IT platform works properly. To this end, this chapter includes the ISLANDER implementation and integration plan. Such plans are based on strategies to successfully deploy and integrate smart computing platforms, such as Gridpilot, within the project's technological infrastructure.

In order to carry out the implementation and integration of devices with computing platforms, the following milestones need to be completed.

1. The first is to define the technology to be incorporated, indicating the problems that the platform is intended to address, as well as the benefits it can provide.
2. Once step 1 has been completed, the next step is to analyse the hardware, software, and network for compatibility between them and the platform.
3. After this analysis, a smaller-scale proof of concept (POC) of the systems to be implemented should be conducted to test their functionality and to identify potential connection or information delivery problems that may occur.

4. After completing the above steps, the implementation plan can be executed. This includes the integration and configuration of the platform and the different components so that they can be tested.
5. Finally, it is necessary to carry out a continuous evaluation of the functioning of the platform and adjust, if required.

4.2 Integration phases

After explaining a general case, this section will describe the integration plan developed in ISLANDER for all the pilots that are going to be installed on the island of Borkum. This integration plan is organised in various phases to ensure that each device can be connected to the platform, avoiding as many problems as possible. For this, the three phases of the integration to be completed are: pre-installation, post-installation, and global integration.

4.2.1 Phase 1. Pre-installation test

The first phase explains how the devices can be integrated into the platform, i.e., the procedure for testing the connection prior to the installation.

4.2.1.1 RESS – household

In order to validate the RESS Household solution, a test environment has been set up at IDENER's facilities. Such an environment includes all parts of the system interconnected as they would be in Borkum households.

As discussed in section 2.1, the RESS Household solutions are connected to the Smart IT Platform through the IDENER's cloud, which acts as an intermediary that collects and processes the data from all the households. Therefore, the integration tests require the following conditions to be met:

3. Successful connection from the houses to IDENER's cloud
4. Successful connection from IDENER's cloud to AYESA's IT platform

The former will be done by connecting the control device to the inverter via an Ethernet cable and using the Modbus protocol. Such a control device will also use the user's router as a gateway to connect to IDENER's cloud via MQTT protocol. The latter will be performed via RabbitMQ and MQTT protocol as well. Two communication channels will be opened, one to send the telemetries and one to receive the setpoints. Furthermore, every connection will be made under a VPN, either IDENER's or AYESA's, to increase the level of security.

Once both connections are established, it is essential to conduct extensive tests to verify that the information being exchanged in both directions is in accordance with the proposal.

Different telemetries and setpoints will be sent and received, respectively, using the previous scheme to verify the successful connection between all the parts. These tests include measuring the accuracy and completeness of the data transmitted, as well as the speed and efficiency of the overall system. By conducting these tests, we can ensure that the RESS Household solution is not only functional but also reliable and efficient.

4.2.1.2 RESS – Building

Communication with Ayesa's control system uses TCP-based communications (MQTT protocol). Therefore, pre-installation tests shall consist of verifying the connectivity between Ayesa's control system and inverters. It is not necessary to test the protocol integration at this stage, just the connectivity (i.e., ping method) between the HMI and the inverter. This simple test is sufficient to verify the correctness of the communication link.

The steps to be followed are:

1. To create a VPN link between the HMI and Ayesa's control system
2. Pinging from HMI to Ayesa's control system
3. Pinging from Ayesa's control system to HMI.

4.2.1.3 Seawater district heating

As mentioned before, this energy asset has suffered delays in its development. For this reason, the integration phases and the implementation plan cannot be applied to this energy asset. However, this information will be included in future deliverables.

4.2.1.4 EV charging station

The EV charging stations are composed of one or several charging units where the EVs are connected. Therefore, Gridpilot will open a connection with each of these charging units utilising the OCCP protocol. Gridpilot will also account for an adapter in charge of maintaining this connection and understanding the data provided by the charging units.

However, an secure internet connection must be set in place beforehand in order to guarantee the connection between Gridpilot and the charging units. Additionally, the charging units should be added to Gridpilot's inventory, providing it with physical and electrical features typical of these charging units.

In this case, OCCP protocol establishes that when a charging unit is connected to a controller (Gridpilot adapter), this will be through a BootNotification message sent by the charging unit. This message will be received by Gridpilot's adapter and, depending on the settings for every case, accepted or rejected. Regardless of the connection outcome, get a BootNotification message, ensure the communication has been established and it is ready to be used basing in OCCP messages protocol between Gridpilot and charging units, so the pre-installation test has

been successful.

4.2.1.5 HESS large scale. UCAP+Li-ion batteries

The communication between the HESS large-scale systems and Ayesa's control system is performed via TCP-based communications (MQTT or Modbus TCP).

Therefore, pre-installation tests shall consist of verifying the connectivity between Ayesa's control system and inverters. It is not necessary to test the protocol integration at this stage, just the connectivity (i.e., ping method) between the HMI and the HESS. This simple test is sufficient to verify the correctness of the communication link.

The steps to be followed are:

1. To create a VPN link between the HMI of the HESS and Ayesa's control system.
2. Pinging from HMI to Ayesa's control system.
3. Pinging from Ayesa's control system to HMI.

4.2.1.6 Hydrogen-based storage

The pre-installation steps for Hydrogen-based storage are similar to those for the RESS Household solution, as both utilize the same technology stack for data retrieval, processing, and connection with the SIT Platform. Therefore, the same conditions need to be met (successful connection to IDENER's cloud and AYESA's IT platform) and similar tests will be performed, i.e., successful retrieval of relevant telemetries and capability to send setpoints to the hydrogen system.

Similarly to the RESS Household solution, a test environment that will be employed to ensure the proper operation of the entire system. As it is stated in the contract with SOLENCO, several Factory Acceptance Tests will be performed after the assembly of the hydrogen system solution at the provider's facilities. As the system will be completely assembled and functional, IDENER will leverage this environment to perform the aforementioned tests. SOLENCO has already agreed to this, and they will also support IDENER during these tests.

4.2.1.7 Street lighting system

The connection between Gridpilot and the Street lighting system will be performed between the adapter on the Gridpilot side and a router on NBG side. A secure internet connection via a VPN must be set in place before establishing this connection.

Once the connection has been established, Gridpilot will pull data a determined period to monitor the Street lighting system status in real time. If the request receives a response, the pre-installation test is successful.

4.2.2 Phase 2. Installed component test

This phase aims to explain the integration plan between each energy asset and the smart IT platform once all components are installed in Borkum. This means that all pilots are working well individually.

4.2.2.1 RESS – household

Once the system has been deployed in each of the households, two types of tests need to be performed: individual and cluster tests. The former is relatively straightforward as these are the same as the ones considered in the first phase. Performing these tests again will allow verifying the correct operation of the systems now in the real environment.

Regarding the cluster tests, the objective is the monitoring of all the devices registered on the platform. This is essential to keep track of the health of the devices and detect any issues or anomalies that may arise. To this end, Thingsboard and its user interface will be leveraged, as it provides an easy-to-use dashboard that displays all the information required in one place. Therefore, checking the status of each device and ensuring that everything is working as intended will be straightforward.

In addition to monitoring the devices, Thingsboard will also be exploited in terms of its plotting capabilities to visualise all the telemetries being transmitted by each device. This will allow identifying any trends and patterns in the data, which can lead to optimising the system's performance.

Another benefit of using Thingsboard is the possibility of creating alarms based on these telemetries. This entails that if any value falls outside its expected range, the system will send an alert immediately. This last feature is crucial to ensure that every issue is addressed promptly, reducing the likelihood of downtime or system failure.

Accordingly to the described characteristics of Thingsboard, different tests will be performed evaluating each of these features. Ultimately, the pursued objective is to verify the proper operation of these systems individually and their performance and health as a whole.

4.2.2.2 RESS – Building (ZIG)

- Channels Creation:
 - Creation of the **general_info** MQTT channel between the inverter and the control system
 - Creation of the **telemetries** MQTT channel between the inverter and the control system
 - Creation of the **setpoints** MQTT channel between the inverter and the control system

- Publication of the equipment information through the general_info channel
- Publication of the power variables through the telemetries channel
- Configuration and Commanding of the equipment through the setpoints channel

The verification of these tests consists of double-checking that all parameters and variables defined in the equipment interface have been published. Moreover, the format used by the publisher must coincide with the one expected by the subscriber.

4.2.2.3 Seawater district heating (NBG)

The lack of information for this system has been explained in section 2.2.3.

4.2.2.4 EV charging station

Upon establishing a successful connection between the Gridpilot platform and the charging units that constitute a charging station, there are two aspects to verify: accurate data reception and appropriate submission of charging profiles.

These connection tests will be performed based on the UI provided by the Gridpilot platform:

Data reception

Once the connection has been received, according to settings previously established (as demand response or energy services actions), the charging unit should start charging or discharging the EV battery or remain in stop mode. Whatever the outcome, the UI provided by Gridpilot should present the session created at the moment an EV is connected to the charging unit and show its connection status.

Charging profiles

A charging profile is a set of consignments generated in Gridpilot, produced by a demand response or energy service action. This charging profile is created as the result of executing an optimization algorithm, whose purpose is to command the necessary orders to accomplish the action to the charging unit.

To validate the tracking orders sent, it will be necessary to use the UI and the monitoring tools to compare the consignment received with the charging unit status.

Additionally, Gridpilot UI would provide a tool to send individual consignments to the charging units.

4.2.2.5 HESS large scale. UCAP+Li-ion batteries

Once the HESS are commissioned in their respective locations, the tests to be performed will consist of verifying the correct communication between the HESS and Ayesa's control system.

The set of tests for this phase 2 is listed below.

HESS Ultracaps

- Channels Creation:
 - Creation of the **general_info** MQTT channel between the HESS and the control system
 - Creation of the **telemetries** MQTT channel between the HESS and the control system
 - Creation of the **setpoints** MQTT channel between the HESS and the control system
- Publication of the equipment information through the **general_info** channel
- Publication of the power variables through the **telemetries** channel
- Configuration and Commanding of the equipment through the **setpoints** channel

The verification of these tests consists of double-checking that all parameters and variables defined in the equipment interface have been published. Moreover, the format used by the publisher must coincide with the one expected by the subscriber.

HESS Li-ion

In this use case, the control system performs the master role. Integration shall be verified by checking the reading of all the input registers supplied by the HESS, as well as checking the writing of all the HESS holding registers.

All writing and reading commands shall be single registers.

Once verified that single register operations are correct, including the data format, it shall be verified the multiple writing and reading registers.

For this device, testing and verification of the communications will be analog. There will be sent instructions of operations for acting for and against the grid, by feeding and consuming energy from it.

4.2.2.6 Hydrogen-based storage

Similarly to the previous phase, the hydrogen-based storage will also undergo analogous tests to the RESS Household solution during this phase. After all, data collection, processing, redirection, inventory management, and connection with the Smart IT Platform are the same.

In summary, Thingsboard's user interface and graphing capabilities, along with its alarm system, will be used to monitor the health of the system and the correctness of the telemetries obtained, deducing patterns and trends to act appropriately.

A key difference compared to the RESS Household is that the hydrogen system (which for IDENER essentially acts as a black box) will also be monitored by SOLENCO, which will take internal telemetry of the system at all times to check its proper operation. This added layer of

monitoring by the provider provides an extra level of assurance and helps to guarantee that the system operates efficiently and effectively.

4.2.2.7 Street lighting system

Gridpilot UI will provide KPIs to monitor the street lighting status and functionalities to control the system (on-off). The integration test will must monitor the current state of the street lighting system and the Gridpilot. Additionally, it will also check if the order has been interpreted by the street lighting system at the destination. For example, if an OFF consignment has been sent, the street lighting system should read it and turn the lights off. In contrast, if an ON consignment has been sent, the street lighting system should read it and turn the lights on.

4.2.3 Phase 3. Full integration

The aim of this phase is to explain the full integration plan between all energy assets and the smart IT platform once all components are installed in Borkum. The idea is to demonstrate that everything works well together and that all devices are well integrated.

5 SYSTEM AND INTEGRATION TESTING

In this chapter, we are going to show the integration tests of the platform in several areas of the ISLANDER project. In terms of the project, there are several functionalities that must be tested for the platform to be appropriately used by the users. The first test is verifying the data flow to ensure that information is sent and received seamlessly between the devices and the platform. The next test is the verification of the functioning of the optimisation algorithms and the functionalities associated with the Demand Response for managing the energy associated with each pilot. The result of this test is the optimisation of production and energy consumption in an effective way.

Another essential point is the DER and HES systems. The testing of this group of devices must ensure that they integrate properly with the platform and that the information they provide is presented correctly within the user interface. The last test is the integration of the electric lighting system. This test must ensure not only the integration but also that there are no failures that could disable this system, given the importance of this system for the island.

Only the tests relating to the energy pilots and the public lighting system will be presented in this document. The reason for this is that the remaining tests (i.e., those related to the algorithms and the functionalities of the Demand Response) will be addressed in other

deliverables of the ISLANDER project.

5.1 Integration testing

This section is intended to explain the performance test for each of the pilots. Therefore, it is important to include a test for each of the connections to be performed. The number of tests must coincide with the number of arrows included in Table 1, i.e., the number of communication tests between devices and the platform.

5.1.1 Test execution 1: RESS – Household

Test Case ID ¹	IDE-HH-001	Short Name	Device registration
Description ²	Automatic registration of Household RPis with the platform		
Module ³	HouseManager	Req ID ⁴	
Prepared By	IDENER	Tested By	IDENER
Pre-condition/s ⁵	RPis connected to the router with internet connection		
Test Steps ⁶			
1	Flash SD with HouseManager-OS		
2	Boot up RPi with the previous SD		
3	Wait until the device appears in Thingsboard UI		
4	Check that device attributes sent are as expected		
Input Data			
Expected result	When booted up with the HouseManagerOS, the RPis should automatically configure and register themselves to IDENER's platform (Thingsboard). Also, the RPis should send some general information about themselves, like serial number, model, IP and connected devices.		

Test Case ID ¹	IDE-HH-002	Short Name	Channel creation
Description ²	Creation of communication channel using AYESA REST-API		
Module ³	PlugnPlay-Controller	Req ID ⁴	
Prepared By	IDENER	Tested By	IDENER
Pre-condition/s ⁵	RPI just registered in Thingsboard		
Test Steps ⁶			
1	Wait for the PlugnPlay-Controller to detect a newly registered device		
2	Check PlugnPlay-Controller logs (optional)		
3	Check if a new communication channel was successfully created		
Input Data	Device serial number (SNXXXXXX)		

Expected result	When completed, a new channel called islander/houses/SNXXXX should be created in AYESA's RabbitMQ server
-----------------	--

Test Case ID ¹	IDE-HH-003	Short Name	Telemetry sending
Description ²	Send telemetries from devices to AYE RabbitMQ		
Module ³	HouseManager, Communication-Controller	Req ID ⁴	
Prepared By	IDENER	Tested By	IDENER, AYESA
Pre-condition/s ⁵	Device registered in Thingsboard, and communication channel created		
Test Steps ⁶			
1	Validate that telemetries are being successfully retrieved in the RPis by checking HouseManager logs		
2	Validate that telemetries are successfully arriving at Thingsboard by checking its UI		
3	Validate that telemetries are successfully sent to RabbitMQ by checking RabbitMQ UI or Communication-Controller logs		
Input Data			
Expected result	Telemetries should be sent to the appropriate channel (islander/houses/SNXXXX/telemetries), with the proper format, every minute.		

Test Case ID ¹	IDE-HH-004	Short Name	Setpoint receiving
Description ²	Receive setpoints and apply them appropriately		
Module ³	HouseManager, Command-Controller	Req ID ⁴	
Prepared By	IDENER	Tested By	IDENER
Pre-condition/s ⁵	Device registered in Thingsboard and healthy		
Test Steps ⁶			
1	Publish some setpoint to RabbitMQ islander/houses/SNXXXX/setpoints channel		
2	Check that Command-Controller successfully parses the setpoint messages,		

	schedules them appropriately and redirects them to the devices.
3	Check that HouseManager receives the setpoint and applies it
Input Data	
Expected result	A setpoint message published in the setpoints channel should be correctly interpreted, meaning that setpoints should be successfully applied to the devices and scheduled accordingly

5.1.2 Test execution 2: RESS – Building

Test Case ID ¹	ZGR-BUILD-001	Short Name	Power Generation
Description ²	Active Power Generation		
Module ³	Inverter 30kW	Req ID ⁴	
Prepared By		Tested By	
Pre-condition/s ⁵	<p>Inverter powered on.</p> <p>Inverter is configured in Standalone mode (Mode=1)</p> <p>Battery is charged</p>		
Test Steps ⁶			
1	Configure the equipment to standalone mode (Mode=1)		
2	Set PREF to a value greater than 0		
3			
Input Data			
Expected result	Inverter supplies the active power set in the PREF setpoint		

Test Case ID ¹	ZGR-BUILD-002	Short Name	Reactive Power Generation
Description ²	Reactive Power Generation		
Module ³	Inverter 30kW	Req ID ⁴	

Prepared By		Tested By	
Pre-condition/s ⁵	Inverter powered on. Inverter is configured in PPOC mode (Mode=2) Battery is charged		
Test Steps ⁶			
1	Configure the equipment to PPOC mode (Mode=2)		
2	Set QREF to a value different to 0		
3			
Input Data			
Expected result	Inverter supplies the reactive power set in the QREF setpoint		

Test Case ID ¹	ZGR-BUILD-003	Short Name	SOC-Mode
Description ²	SOC Mode		
Module ³	Inverter 30kW	Req ID ⁴	
Prepared By		Tested By	
Pre-condition/s ⁵	Inverter powered on. Inverter is configured in PPOC mode (Mode=3) Battery is partially charged		
Test Steps ⁶			
1	Configure the equipment to SOC mode (Mode=3)		
2	Modify the setpoint REFSOC below the current value of SOC		
3	Modify the setpoint REFSOC greater than the current value of SOC		
Input Data			

Expected result	Inverter gives active power to the grid when SOC>REFSOC, and takes power from the grid when SOC<REFSOC.
-----------------	---

Test Case ID ¹	ZGR-BUILD-004	Short Name	Charge Mode
Description ²	Inverter in Charge mode		
Module ³	Inverter 30kW	Req ID ⁴	
Prepared By		Tested By	
Pre-condition/s ⁵	Inverter powered on. Inverter is configured in SOC mode (Mode=4) Battery is partially charged		
Test Steps ⁶			
1	Configure the equipment to CHARGE mode (Mode=4)		
2	Configure V battery max parameter greater than current battery voltage		
3	Configure V battery min parameter below current battery voltage		
4	Check behaviour (1)		
5	Configure V battery min parameter greater than current battery voltage		
6	Check behaviour (2)		
Input Data			
Expected result	(1) Inverter charges the battery (2) Inverter stops charging the battery		

5.1.3 Test execution 3: Seawater district heating

The lack of information in this point has been explained in section 2.2.3.

5.1.4 Test execution 4: EV charging station

Test Case ID ¹	AYE-EV-001	Short Name	Accepted charging unit connection
Description ²	Accept the connection from a charging unit.		
Module ³	Gridpilot-EV	Req ID ⁴	
Prepared By	AYE	Tested By	
Pre-condition/s ⁵	VPN stablished		
Test Steps ⁶			
1	BootNotification message received		
2	SCSMS valid the message		
3	Charging unit identification in Gridpilot inventory has been set to be accepted		
4	SCSMS accept the connection		
Input Data	<pre>"BootNotification",{"chargePointVendor":"AVT-Company", "chargePointModel":"AVT-Express", "chargePointSerialNumber":"avt.001.13.1", "chargeBoxSerialNumber":"avt.001.13.1.01", "firmwareVersion":"0.9.87", "iccid":""," "imsi":""," "meterType": "AVT NQC-ACDC", "meterSerialNumber":"avt.001.13.1.01" }</pre>		
Expected results	{"currentTime":1649837683817,"interval":300,"status":"Accepted"}		

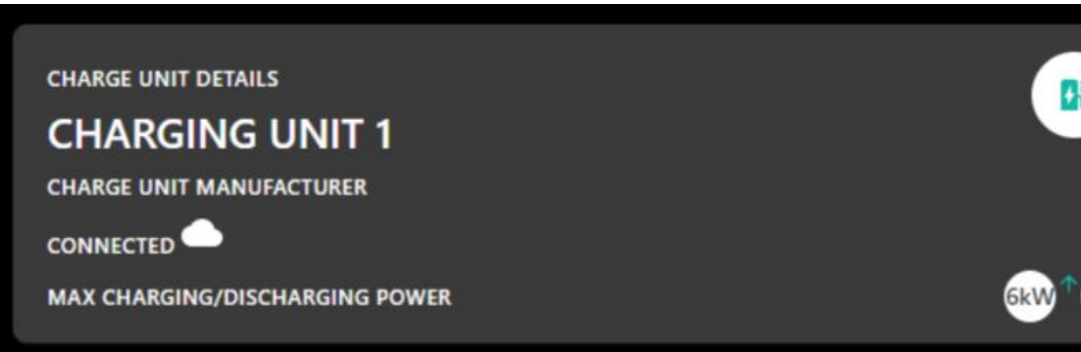
Test Case ID ¹	AYE-EV-002	Short Name	Rejected charging unit connection
Description ²	Rejected the connection from a charging unit.		
Module ³	Gridpilot-EV	Req ID ⁴	
Prepared By	AYE	Tested By	
Pre-condition/s ⁵	VPN stablished		
Test Steps ⁶			
1	BootNotification message received		
2	SCSMS valid the message		
3	Charging unit identification in Gridpilot inventory has been set to be rejected		

4	SCSMS reject the connection
Input Data	<pre>"BootNotification",{"chargePointVendor":"AVT-Company", "chargePointModel":"AVT-Express", "chargePointSerialNumber":"avt.001.13.1", "chargeBoxSerialNumber":"avt.001.13.1.01", "firmwareVersion":"0.9.87", "iccid":""," "imsi":""," "meterType": "AVT NQC-ACDC", "meterSerialNumber":"avt.001.13.1.01" }</pre>
Expected results	{"currentTime":1649837683817,"interval":300,"status":"Rejected"}

Test Case ID ¹	AYE-EV-003	Short Name	Start charging an EV
Description ²	Once a connection has been established, the SCSMS sends an order to start charging the EV		
Module ³	Gridpilot-EV	Req ID ⁴	
Prepared By	AYE	Tested By	
Pre-condition/s ⁵	VPN established and charging unit previously validated and accepted by SCSMS		
Test Steps ⁶			
1	SCSMS sends startTransaction received		
2	Gridpilot UI shows the transaction has been started		
3			
Input Data	[{"connectorId":1,"idTag":"123","meterStart":11,"timestamp":"2023-03-06T08:10:06Z"}]		
Expected result	[3,"67e28d0a-cec3-44cc-bf13-0263d13d477f",{"idTagInfo":{"expiryDate":"2023-03-07T08:10:06Z"},"status":"Accepted"},"transactionId":103}]		

Test Case ID ¹	AYE-EV-004	Short Name	Stop charging an EV
Description ²	Once a connection has been established, the SCSMS sends an order to stop charging the EV		
Module ³	Gridpilot-EV	Req ID ⁴	

Prepared By	AYE	Tested By	
Pre-condition/s ⁵	VPN established and charging unit previously validated by SCSMS		
Test Steps ⁶			
1	SCSMS sends stopTransaction received		
2	Gridpilot UI shows the transaction has been stopped		
3			
Input Data	[{"connectorId":1,"idTag":"123","meterStart":11,"timestamp":"2023-03-06T08:10:06Z"}]		
Expected result	"RemoteStopTransaction", {"transactionId":10342 }		

Test Case ID ¹	AYE-EV-005	Short Name	Current charging unit status
Description ²	Once a connection has been established, monitor their values		
Module ³	Gridpilot-EV	Req ID ⁴	
Prepared By	AYE	Tested By	
Pre-condition/s ⁵	VPN established, charging unit previously validated and accepted, and a start transaction consignment sent by SCSMS		
Test Steps ⁶			
1	Gridpilot UI shows the session created and the current status		
2			
3			
Input Data			
Expected result	Gridpilot UI shows the connected charging unit status in real time (The picture below may change when the UI has been defined and implemented):		
			

Test Case ID ¹	AYE-EV-006	Short Name	Send charging profiles
Description ²	Once a connection has been established, Gridpilot sends a charging profile generated by an energy service or demand response action.		
Module ³	Gridpilot-EV	Req ID ⁴	
Prepared By	AYE	Tested By	
Pre-condition/s ⁵	VPN established and the charging unit validated by SCSMS. Previously, an energy services or a demand response event has been set to add the charging unit as a part of its process		
Test Steps ⁶			

1	Gridpilot sends the charging profile.
2	Check if charging unit follows the charging profile sent
3	
Input Data	
Expected result	<pre>{ "connectorId":1, "chargingStationId":"C5031WHT48GJ", "csChargingProfiles": { "chargingProfileId": 1, "transactionId": 103, "stackLevel": 99, "chargingProfilePurpose": "TxProfile", "chargingProfileKind": "Absolute", "chargingSchedule": { "duration": 3000, "startSchedule": "2023-03-08T09:25:00Z", "chargingRateUnit": "W", "chargingSchedulePeriod": [{ "startPeriod": 0, "limit": 400, "numberPhases": 1 }] } } }</pre>

5.1.5 Test execution 5: HESS Large scale. UCAP+Li-ion storage

Test Case ID ¹	ZGR-HESS-001	Short Name	Power Generation
Description ²	Active Power Generation		
Module ³	HESS Ultracaps	Req ID ⁴	
	HESS Li-ion		
Prepared By		Tested By	
Pre-condition/s ⁵	HESS powered on. HESS is configured in Standalone mode (Mode=1) Battery is charged		
Test Steps ⁶			
1	Configure the equipment to standalone mode (Mode=1)		
2	Set PREF to a value greater than 0		
3			

Input Data	
Expected result	HESS supplies the active power set in the PREF setpoint. Depending on the energy requirements, power is provided directly from the UCAPS system or both with UCAPS and battery systems.

Test Case ID ¹	ZGR-HESS-002	Short Name	Reactive Power Generation
Description ²	Reactive Power Generation		
Module ³	HESS Ultracaps	Req ID ⁴	
	HESS Li-ion		
Prepared By		Tested By	
Pre-condition/s ⁵	HESS powered on. HESS is configured in PPOC mode (Mode=2) Battery is charged		
Test Steps ⁶			
1	Configure the equipment to PPOC mode (Mode=2)		
2	Set QREF to a value different to 0		
3			

Input Data	
Expected result	HESS supplies the reactive power set in the QREF setpoint

Test Case ID ¹	ZGR-HESS-003	Short Name	SOC-Mode
Description ²	SOC Mode		
Module ³	HESS Ultracaps	Req ID ⁴	

	HESS Li-ion		
Prepared By		Tested By	
Pre-condition/s ⁵	HESS powered on. HESS is configured in PPOC mode (Mode=3) Battery is partially charged		
Test Steps ⁶			
1	Configure the equipment to SOC mode (Mode=3)		
2	Modify the setpoint REFSOC below the current value of SOC		
3	Modify the setpoint REFSOC greater than the current value of SOC		
Input Data			
Expected result	HESS gives active power to the grid when SOC>REFSOC, and takes power from the grid when SOC<REFSOC.		

Test Case ID ¹	ZGR-HESS-004	Short Name	Charge Mode
Description ²	HESS in Charge mode		
Module ³	HESS Ultracaps	Req ID ⁴	
	HESS Li-ion		
Prepared By		Tested By	
Pre-condition/s ⁵	HESS powered on. HESS is configured in SOC mode (Mode=4) Battery is partially charged		
Test Steps ⁶			
1	Configure the equipment to CHARGE mode (Mode=4)		
2	Configure V battery max parameter greater than current battery voltage		
3	Configure V battery min parameter below current battery voltage		
4	Check behaviour (1)		
5	Configure V battery min parameter greater than current battery voltage		
6	Check behaviour (2)		

Input Data	
Expected result	<ul style="list-style-type: none"> (1) HESS charges the battery (2) HESS stops charging the battery

5.1.6 Test execution 6: Hydrogen-based storage

Test Case ID ¹	IDE-HS-001	Short Name	Device registration
Description ²	Automatic registration of the RPi controlling the Hydrogen storage with the platform		
Module ³	HydrogenManager	Req ID ⁴	
Prepared By	IDENER	Tested By	IDENER
Pre-condition/s ⁵	Rpi connected to the router with an internet connection		
Test Steps ⁶			
1	Flash SD with HydrogenManager-OS		
2	Boot up RPi with the previous SD		
3	Wait until the device appears in Thingsboard UI		
4	Check that device attributes sent are as expected		
Input Data			
Expected result	When booted up with the HydrogenManager-OS, the RPi should automatically configure and register itself to IDENER's platform (Thingsboard). Also, the RPi should send some general information about themselves, like serial number, model and IP.		

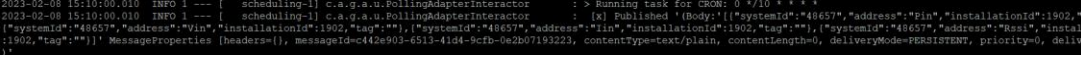

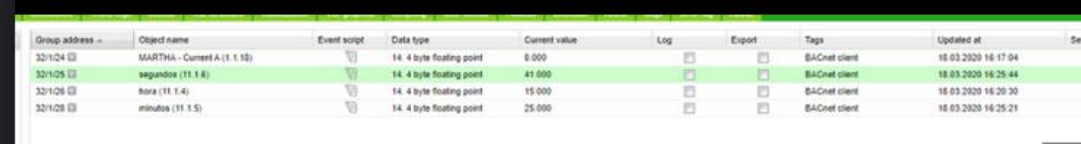
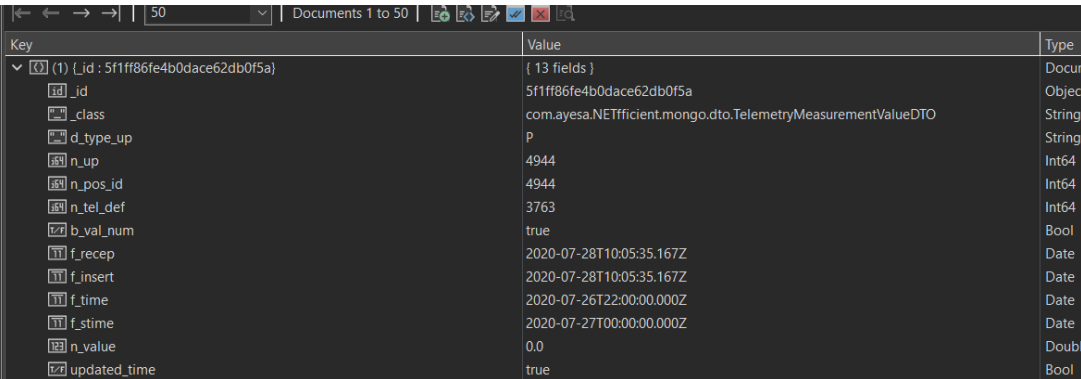
Test Case ID ¹	IDE-HS-002	Short Name	Telemetry sending
Description ²	Send telemetries from the Hydrogen Storage system to AYE RabbitMQ		
Module ³	HydrogenManager, Communication-Controller	Req ID ⁴	
Prepared By	IDENER	Tested By	IDENER, AYESA
Pre-condition/s ⁵	Device registered in Thingsboard and communication channel created		
Test Steps ⁶			

1	Validate that telemetries are being successfully retrieved in the RPi (meaning a successful connection to the system) by checking HydrogenManager logs
2	Validate that telemetries are successfully arriving at Thingsboard by checking its UI
3	Validate that telemetries are successfully sent to RabbitMQ by checking RabbitMQ UI or Communication-Controller logs
Input Data	
Expected result	Telemetries should be sent to the appropriate channel with the proper format, every minute.

Test Case ID¹	IDE-HS-003	Short Name	Setpoint receiving
Description²	Receive setpoints and apply them appropriately		
Module³	HydrogenManager, Command-Controller	Req ID⁴	
Prepared By	IDENER	Tested By	IDENER
Pre-condition/s⁵	Device registered in Thingsboard and healthy		
Test Steps⁶			
1	Publish some setpoint to the RabbitMQ channel in charge of commands and setpoints		
2	Check that Command-Controller successfully parses the setpoint messages, schedules them appropriately and redirects them to the devices.		
3	Check that HydrogenManager receives the setpoint and applies it		
Input Data			
Expected result	A setpoint message published in the setpoints channel should be correctly interpreted, meaning that setpoints should be successfully applied to the device and scheduled accordingly		

5.1.7 Test execution 7: Street lighting system

Test Case ID	AYE-SL-001	Short Name	Get data from HomeLynk
---------------------	------------	-------------------	------------------------

Description	The SIT will request data from Street Lightning HomeLynk.		
Module		Req ID	
Prepared By	AYESA	Tested By	AYESA
Pre-condition/s	VPN established		
Test Steps			
1	Generate a pulse to get the data.		
2	The query has been requested rightly according to HL endpoints and protocols		
3	Check manually in HL the value in the current minute		
4	The data has been obtained and stored in historical DB with the right timing and value.		
Input Data			
Expected result	<p>1. Generate a pulse to get the data: Polling adapter µService will request every minute to HL</p>  <p>2. The query has been requested rightly according to HL endpoints and protocols:</p>  <p>3. Check manually in HL the value in the current minute</p>  <p>4. The data has been obtained and stored in historical DB with the right timing and value</p> 		

Test Case ID	AYE-SL-002	Short Name	Send OFF order
---------------------	------------	-------------------	----------------



Description	Gridpilot will order to turn the lights off.		
Module		Req ID	
Prepared By	AYESA	Tested By	AYESA
Pre-condition/s	VPN established		
Test Steps			
1	A turn-the-lights-off order has been created by Gridpilot UI		
2	Check the data has been acknowledged by the Street Lightning device		
3	Check Gridpilot UI shows the lights have been turned off		
Input Data			
Expected result	The HL shows it has received and applied the consignment for shutting down the lights		

Test Case ID	AYE-SL-003	Short Name	Send ON order
Description	Gridpilot will order to turn the lights on.		
Module		Req ID	
Prepared By	AYESA	Tested By	AYESA
Pre-condition/s	VPN established		
Test Steps			
1	A turn-the-lights-on order has been created by Gridpilot UI		
2	Check the data has been acknowledged by the Street Lightning device		
3	Check Gridpilot UI shows the lights have been turned on		
Input Data			
Expected result	The HL shows it has received and applied the consignment for turning the lights on		

6 MAIN CONCLUSIONS

This document aims to explain the integration plan of all the energy assets that are going to be installed in Borkum into the ISLANDER project.

Considering the information provided in the previous point, it can be concluded that each device from every energy asset can be connected and integrated with the rest of the components. Furthermore, the test steps performed showed the expected results for each partner. With these outcomes, the integration test plan explained in this deliverable can be deemed completed and is ready to be included in the next step of the infrastructure integration, deliverable *D6.2. Report on the integration tests outcomes*.

DEVIATIONS

Delivery of the content was delayed by 5 months to increase the quality of the outcome. No deviations are expected for subsequent actions.

7 REFERENCES

- [1] ECharts, A. (2023). *Apache Echarts*. Von <https://echarts.apache.org/en/index.html> abgerufen
- [2] Gitlab. (2023). *Gitlab*. Von <https://about.gitlab.com/> abgerufen
- [3] JUnit5. (2023). *JUnit5*. Von <https://junit.org/junit5/> abgerufen
- [4] MQTT. (2023). *MQTT*. Von <https://mqtt.org/> abgerufen
- [5] Oracle. (2023). Von Java: <https://www.java.com/es/> abgerufen
- [6] Selenium. (2023). *Selenium*. Von <https://www.selenium.dev/> abgerufen
- [7] Wallbox. (February 2022). Von <https://wallbox.com/> abgerufen
- [8] Wikipedia. (2023). *Representational state transfer*. Von https://en.wikipedia.org/wiki/Representational_state_transfer abgerufen