HYPOBATT

Hyper powered vessel battery charging system

Document Title	**Report on Operational Optimization**
Document type and number	**Deliverable D2.3**
Primary Author(s)	Endika Bilbao Muruaga | IKERLAN
Document Version | Status	2.0 | Final
Distribution level	PU – PUBLIC
Project Acronym	**HYPOBATT**
Project Title	Hyper Powered vessels battery charging system
Project website	WWW.HYPOBATT.EU
Project Coordinator	Endika Bilbao | IKERLAN | ebilbao@ikerlan.es
Grant agreement number	101056853

Date of deliverable: [30.11.2023]

Date of submission: [04.12.2023]

©HYPOBATT has received funding from the European Horizon Europe Framework Programme (HORIZON) under grant agreement No. 101056853. This reflects only the author's views. The Agency is not liable for any use that may be made of the information contained therein

CONTRIBUTIOR AND FORMAL REVIEWRS

	Name | Organisation	Date
Document Manager	Endika Bilbao Muruaga (IKERLAN)	21/07/2023
Contributor 1	Álvaro Reina Illanes (BRING)	03/11/2023
Contributor 2	Eneko Vaquero (IKERLAN)	06/11/2023
Contributor 3	Gorka Elezgarai (IKERLAN)	06/11/2023
Contributor 4	Mikel Lopez (IKERLAN)	06/11/2023
Contributor 5	Victor Collazos (FV)	07/11/2023
Contributor 6	Paulo Cardoso (BRING)	07/11/2023
Contributor 7	George Kostalas (RHOE)	07/11/2023
Contributor 8	Stefanos Sekis (RHOE)	07/11/2023
Contributor 9	Stelios Gkogkas (RHOE)	10/11/2023
Internal Reviewer 1	George Kostalas (RHOE)	20/11/2023
Internal Reviewer 2	Paulo Cardoso (BRING)	24/11/2023
External Reviewer 1	Pietro Grippi (STT)	29/11/2023
External Reviewer 2	Sara Moqaddamerad (USTRATH)	30/11/2023
External Reviewer 3	Deniz Suzen (IMECAR)	30/11/2023
External Reviewer 4	Alessandro Maccari (RINA)	29/11/2023

DOCUMENT HISTORY

Document Version	Date	Author | Organization	Description
0.1	21/07/2023	Endika Bilbao Muruaga (IKERLAN)	Initial version
1.0	12/11/2023	Endika Bilbao Muruaga (IKERLAN)	First internal released version
1.1	24/11/2023	George Kostalas (RHOE)	Reviewed version
1.2	24/11/2023	Paulo Cardoso (BRING)	Reviewed version
1.3	24/11/2023	Endika Bilbao (IKERLAN)	Edited version
1.4	04/12/2023	Endika Bilbao (IKERLAN)	Review – Quality assurance
1.5	04/12/2023	Endika Bilbao (IKERLAN)	Edited final version
2.0	04/12/2023	Endika Bilbao (IKERLAN)	Final version – Submitted to EC

Approval Status			
WP Leader Approval	Organisation Name	Checked By	Approved
	RHOE	George Kostalas	Date: 04.12.2023 Signature:

TABLE OF CONTENTS

LIST OF ABBREVIATIONS, ACRONYMS AND DEFINITIONS

Abbreviation	Word
ACLs	Access Control Lists
AIS	Automatic Identification System
AWS	Amazon Web Services
BESS	Battery Energy Storage System
CC	Constant Current
CV	Constant Voltage
DC	Direct Current
DT	Digital-Twin
EC2	Elastic Compute Cloud
EMS	Energy Management Strategy
ESS	Energy Storage System
EVESSEL	Electric Vessel
HTTP	Hypertext Transfer Protocol
IAM	Identity and Access Management
ID	Identification
IoT	Internet of Things
LoRaWAN	Long Range Wide Area Network
MFA	Multi-Factor Authentication
MQTT	Message Queuing Telemetry Transport
RBAC	Role-Based Access Control
S3	Simple Storage Service
SoC	State of Charge
SoH	State of Health
TCP	Transmission Control Protocol
TLS/SSL	Transport Layer Security/ Secure Sockets Layer
WSL	Windows Subsystem for Linux
API	Application Programming Interface

LIST OF TABLES

LIST OF FIGURES

1. EXECUTIVE SUMMARY

In this deliverable summary, the successful implementation of Digital-Twin (DT) and Energy Management Strategies within a cloud platform, utilizing a DataLake for storage, is outlined. The primary objective of this initiative is to seamlessly integrate these models and algorithms into the daily operational framework of the designated application, i.e. fast charging of battery systems and operational optimization of electric vessels (EVESSEL). A comprehensive framework has been designed and established, encompassing both front-end and back-end components, ensuring a user-friendly experience and efficient data processing. Rigorous measures have been taken to guarantee robust communication and cybersecurity aspects to safeguard data integrity and confidentiality.

The incorporation of a DataLake within this cloud-based environment provides distinct advantages for data management. It allows for the centralized storage of vast volumes of structured and unstructured data, facilitating efficient data access, retrieval, and analysis. This DataLake enhances the system's capacity to handle diverse data types generated by the Digital-Twin and Energy Management Strategies.

Leveraging a cloud-based environment with a DataLake, offers several benefits, including scalability, real-time data access, and enhanced collaboration possibilities. It also facilitates the swift deployment of updates and optimizations, ensuring the system remains technologically current and adaptable.

Extensive testing has been conducted to assess the performance and efficiency of this cloud-based system with DataLake storage, comparing it to standalone solutions of the Digital-Twin and Energy Management Strategy. This analysis reveals the benefits of cloud-based implementation with DataLake storage, such as improved resource utilization, reduced response times, and the potential for cost savings. These findings underscore the potential for streamlined and data-driven operational enhancements across various industrial applications, emphasizing the prospects for sustainability and efficiency improvements.

This deliverable represents a significant milestone in the deployment and practical application of Digital-Twin and Energy Management Strategies within a cloud environment enhanced by DataLake storage, setting the stage for more efficient and data-driven operational improvements.

Keywords: Cloud Platform, IoT, Digital-Twin, Energy Management Strategy, Control layer, Optimization, Front-End, Back-End, Real-Time Control

2. OBJECTIVES

Main Objective: Operational optimization of electric vessels is a complex project that revolves around seamless integration of various components to achieve its goals. Central to this effort is the implementation of control strategies and a Digital-Twin within a cloud platform, connecting the digital and physical realms for real-time monitoring and enhanced performance.

It is recalled that the overarching objective is to develop a digital twin of the ship side (battery model) and the shore side (hyper power charger) including control, power, and energy management to optimize the overall performance and operation of the whole integrated system.
Based upon the activities in T2.1 (digital twin of whole system including sub models of ship and shore systems) and T2.2 (control and energy management with smart fast charging algorithm), this D2.3 focus on the aspects of operational optimization of daily operation of charger, from booking to sailing of the vessels, integrated in a cloud platform.

To achieve this objective, the architectural design is pivotal, with distinct components fulfilling specific roles. The user-centric front-end facilitates interaction and offers insights for recharging planning, optimizing vessel operations. On the back end, the control system manages the recharging process, orchestrating charging infrastructure and ensuring efficient power transfer, forming the system's backbone.

Crucial to the project is the establishment of secure communication channels among the electric vessel, charging infrastructure, and cloud platform. This involves defining communication protocols and addressing cybersecurity concerns to safeguard sensitive data and maintain system integrity.

Data management is another significant challenge, necessitating an intermediate database for data storage and sharing within the cloud platform. Additionally, a DataLake will be implemented to store vast data volumes, enabling in-depth analysis, machine learning, and training of control strategies. This repository supports data-driven decision-making, enhancing system efficiency and performance.

In summary, optimizing electric vessels is a complex project driven by the synergy of control strategies, Digital-Twin technology, and a robust cloud platform. It relies on secure communication, reliable recharging, and efficient data management, aiming to advance the electric vessel industry towards greater sustainability and performance.

Additional Objectives:

- Digital-Twin transformation and implementation from MATLAB-SIMULINK
- Database and DataLake definition and implementation
- Full-Solution Validation

3. INTRODUCTION

Electric vessels for maritime transport have a significant challenge ahead, the management of battery charging and the availability of chargers in the different ports. For this type of application, Digital-Twins are becoming more and more common, providing a reliable model of the real system, integrating measurements made with real-world sensors and allowing the hardware to run digitally in parallel to the real system. In addition, cloud-based systems are becoming more common, providing access from any location and enabling decentralized management.

To provide a solution to the electric vessels management, the Digital-Twin composed by the battery, charger and grid models is integrated into the cloud, enabling to simulate the different ports charging systems. The model results are then integrated in a DataLake that stores the real system measurements as well as the Digital-Twin outputs, facilitating the real time data access for multiple issues.

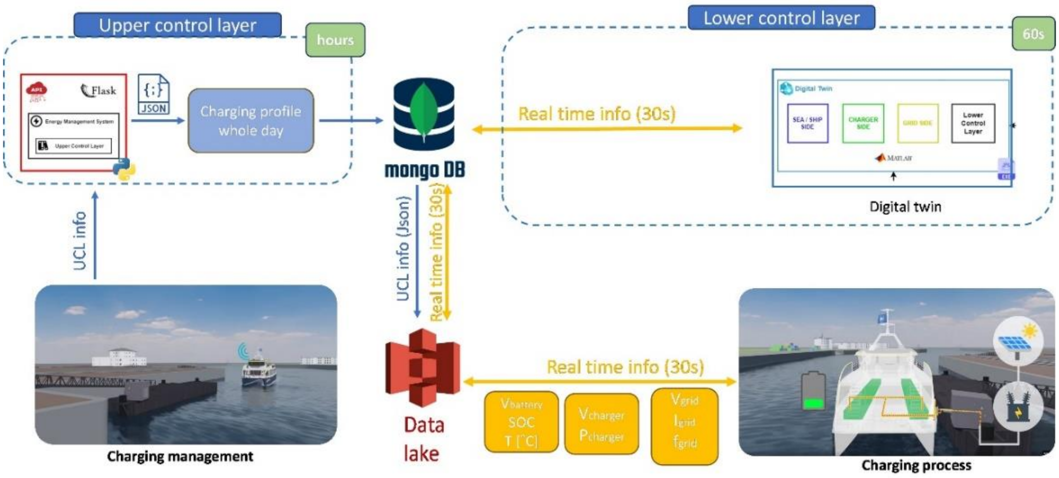

Figure 1 Cloud-based architecture example

Directing into this project, the cloud-based architecture is composed by two main parts, the upper control layer and the DataLake, and the lower control layer. An example of the cloud architecture is shown in Figure 1.

The first is a predictive model to estimate the charging time and the necessary conditions for charging, such as the power required. With this, the port manager is able to decide what services to provide to the ship in question before the ship is loaded, offering a forecasting of the charging process and storing all this data into a DataLake.

The second one is the Digital-Twin model that runs in parallel to the real charging process of the vessel. This model has the real system inputs such as temperature or grid voltages and is able to execute during the charging process in order to emulate the physical system in the cloud.

The whole cloud implementation provides a solution for optimizing vessel charging in the maritime industry, estimating the needed resources for a charging process and allowing to have the real time model available with no need of being in the specific port.

4. DESCRIPTION OF WORK

In this chapter all the elements of the cloud platform as well as the communications are addressed.

4.1 Overall Framework Description

Structure of the operation is as follows, shown in Figure 2: the electric vessel arrives in port and emits a signal for charging need. This signal is received by the port authority, which also receives the data to start the simulation by means of the edge devices implemented at the charger and the vessel. These will transmit data through JSON files towards the DataLake, whose description is given in a chapter below.

With the data, both upper and lower layers can be initialized. The upper will grab the data from the DataLake to run the scheme that provides a daily schedule of charging, with the optimized amount of energy at each charging event, with a final step of uploading the end results into the database. With this information, the digital twin will use both the obtained results of the upper control layer and the physical initial conditions by gathering the upper layer inputs from the database and the initial physical inputs from the DataLake. This will allow the lower control layer to run an optimal charging cycle for both the digital twin and the physical twin, by providing the profile to the computer in the port. After the vessel is charged, the data of the physical system with the charging cycle provided are stored in the DataLake and can be compared with the data obtained from the virtual twin to assess the operation of the virtualized system and proceed with any optimization to further align operations with the physical system.

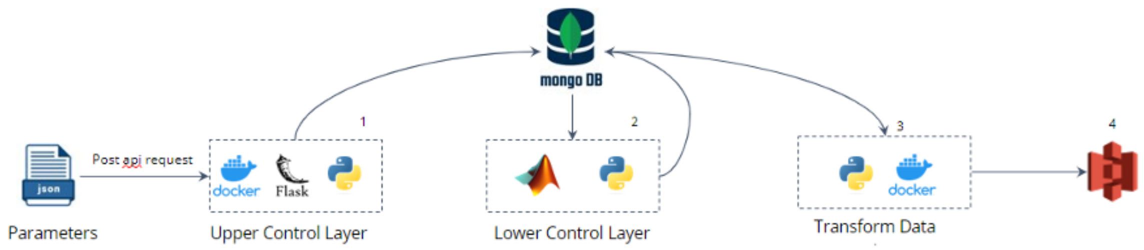

Figure 2 Pipeline for the operation of the cloud solution, considering only the data feeding from the virtual side.

To account for the daily operation effects on the battery degradation, the digital twin will gather the capacity degradation data from the previous simulation.

4.1.1 Upper Control Layer Description

4.1.1.1 Initialization

The upper control layer is initialized with a POST request that carries the simulation parameters in JSON format. To facilitate this, a dedicated API application has been developed using the Flask framework in Python. The upper control layer design allows for flexibility in its parameters, providing options to focus optimization on either cost or reduction of the number of charges.

This flexibility extends to the choice of optimization algorithms, with both genetic and linear programming methods available.

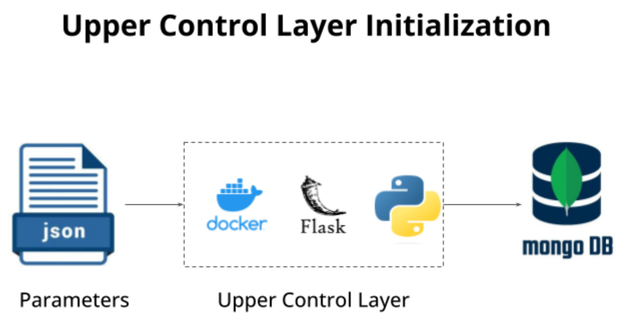

Figure 3 Upper Control Layer Initialization procedure.

Following initialization Figure 3,the upper control layer determines the optimal charging schedules, based on the selected optimization strategy. The results are formatted as JSON files and stored in MongoDB

4.1.1.2 Connection with lower control layer

For the integration between the Upper control layer and the lower control layer, two Python scripts have been developed. The first script is responsible for extracting data from the 'Lower Control Exports' collection within MongoDB. This script transforms the data into a format compatible with MATLAB arrays, preparing it for the Lower Control Layer environment. Once the model completes its execution, the second Python script is triggered to transfer the output back to the 'Simulation Results' collection (Figure 4).

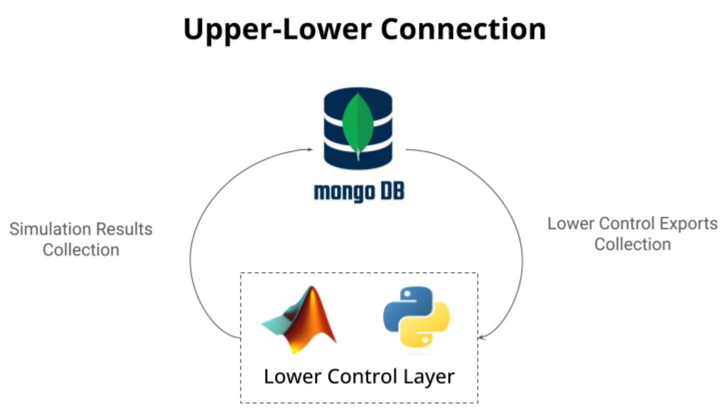

Figure 4 Upper-Lower Control Connection framework.

4.1.1.3 Data transformation and DataLake upload

The terminal phase of the pipeline involves the conversion and upload of the simulation results from both layers to the HYPOBATT DataLake. The automation of this process is achieved through the execution of a Python script, which handles the necessary data transformations and subsequent uploading to the project's S3 (Simple Storage Service - object storage service of AWS) bucket. Initially the script converts the data from JSON files into pandas data frames. Then these data frames are converted to Parquet format, which is recognized for its efficiency in both storage and retrieval operations, particularly in cloud environments. The data, now in Parquet format, is systematically uploaded to the designated folders within the S3 bucket, ensuring organized storage and easy accessibility (Figure 5).

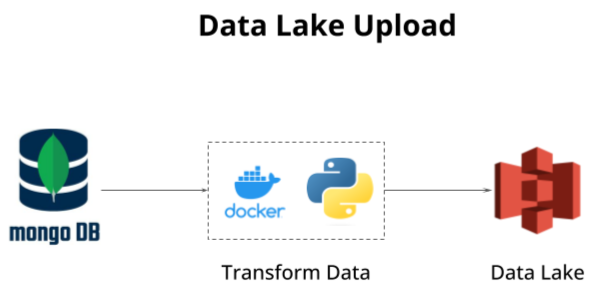

Figure 5 DataLake Upload procedure.

4.2 Cloud Back-End Solution

This section of the document outlines the development of a cloud-based architecture that integrates upper control layer, Digital-Twin lower control layer and front-end using various resources from Amazon Web Services (AWS). Initially, a study of the different platforms and cloud service providers such as Azure, AWS, Google Cloud and OVH was carried out. As already specified, AWS was chosen because it provides services such as IoT (Internet of Things) Core required for data exchange and S3 as storage to define the DataLake.

The IoT Core service will be used to exchange the data captured by the sensors of vessels, chargers, and power grid. This service enables support for multiple protocols such as MQTT (Message Queuing Telemetry Transport) and HTTP (Hypertext Transfer Protocol) allowing devices to interoperate with the platform. It also offers high scalability and security by providing multiple layers of security to protect data. But one of the great advantages of this service is that rules and customized actions can be set, allowing to the possibility to indicate where this data will be stored. In this case the captured data will be stored in a DataLake using the AWS S3 service for its definition.

The core of this solution is based on two Elastic Compute Cloud (EC2) instances which provide scalable computing capacity on demand, as it is shown in Figure 6. The Digital-Twin which already includes the lower control layer, the upper control layer and the front-end have been deployed on these instances. One of them includes the Digital-Twin and the lower control layer developed with MATLAB-SIMULINK while the other one includes the upper control layer and the front-end with back-end built with Python and JavaScript.

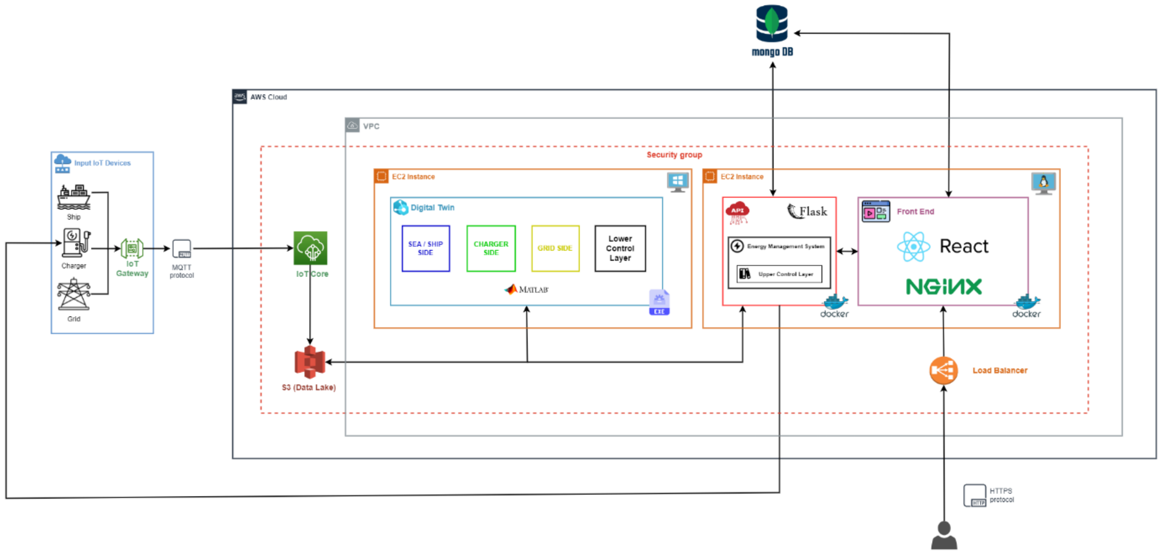

Figure 6 Cloud Back-End EMS Implementation.

4.2.1 EMS Implementation

The EMS is composed by the upper control layer and the lower control layer but as the first one is developed with MATLAB-SIMULINK and the other one with Python it is deployed separately. The EC2 instance containing the lower control layer, shown in Figure 7, is a T2.Large Windows based instance. This system has been used to be able to install the MATLAB Runtime that allows to launch the executable which contains the lower control layer previously created.

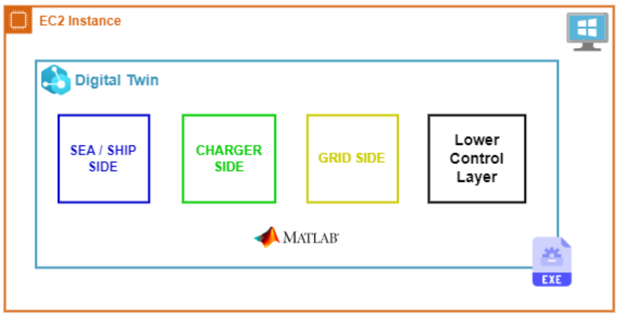

Figure 7 Digital Twin scheme.

The other EC2 is an Ubuntu based T2.Micro instance of Figure 8, where both the upper control layer and the front-end with back-end have been deployed. Both services are dockerized, which greatly facilitates their implementation. It has been necessary to create this instance to avoid an incompatibility created while trying to install the Windows Subsystem for Linux (WSL) in a Windows virtual instance, which is a technology needed for running the docker software.

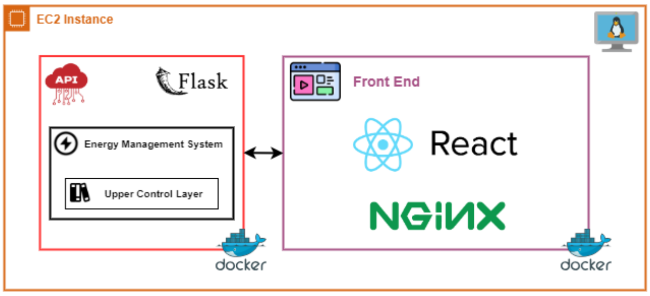

Figure 8 Upper control layer and front-end with back-end scheme.

4.2.2 Digital-Twin Implementation

In order to run the Digital-Twin on the Windows-based EC2 instance, an executable, i.e. an *.exe file, was initially generated. Subsequently, as mentioned above, MATLAB Runtime has been installed to launch this executable. But in addition, to enable controlled execution of the executable, an API has been developed and deployed using the Flask framework on this instance. This API features two endpoints, enabling the initiation of a loop executing the executable and facilitating on-demand termination.

This execution loop leverages inputs stored in MongoDB by the upper control layer, combined with the results from the preceding execution of the Digital-Twin. The outcomes generated by the Digital-Twin are stored in a dedicated DataLake and MongoDB.

4.2.3 DataLake Implementation

The DataLake for HYPOBATT project has been implemented within AWS, utilizing Amazon S3 (Figure 9). Currently, there is one demo bucket in use named 'HYPOBATT-test-bucket'. This bucket is hosted on the AWS Europe (Frankfurt) server (eu-central-1 region). It contains two subfolders: 'upper control layer' and 'lower control layer'. Presently only data from upper and lower control layers are uploaded each time a simulation is conducted. The data are managed in Parquet format, known for storage efficiency and query processing performance.

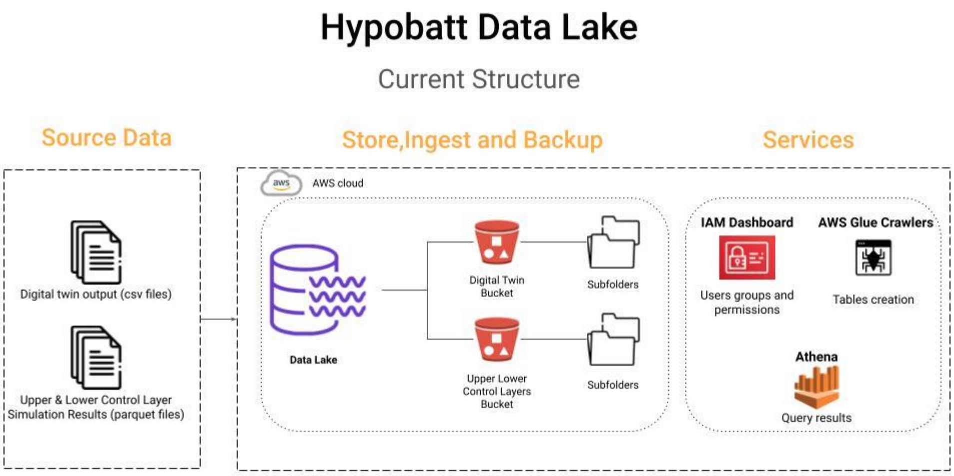

Figure 9 DataLake structure.

User access is defined across three levels: viewers, who have permissions for only viewing data; programmers, who can view, retrieve and upload data without the ability to modify or delete; and admins, who have full DataLake access. Users are also grouped per company/organization inside identity and access management (IAM).

For data handling, six Athena tables have been configured to facilitate efficient analysis. These tables are part of a demo database named 'HYPOBATT-test-db'. These tables are essential for managing metadata effectively. For data integration automation six AWS Glue crawlers are associated with the subfolders of the 'upper-control-layer' and 'lower-control-layer'. These crawlers are designed to automatically synchronize the Athena tables with any newly added data, ensuring data retrieval optimization and reducing the necessity for manual intervention.

4.3 Cloud Front-End Solution

The Cloud Front-End Solution for HYPOBATT begins with a welcoming and secure login page, serving as the gateway to the hyper-powered vessel battery charging system. This login page Figure 10 is the first touchpoint for Admins, Editors, and Viewers, each greeted by the project's mission statement and a clear, concise login form.

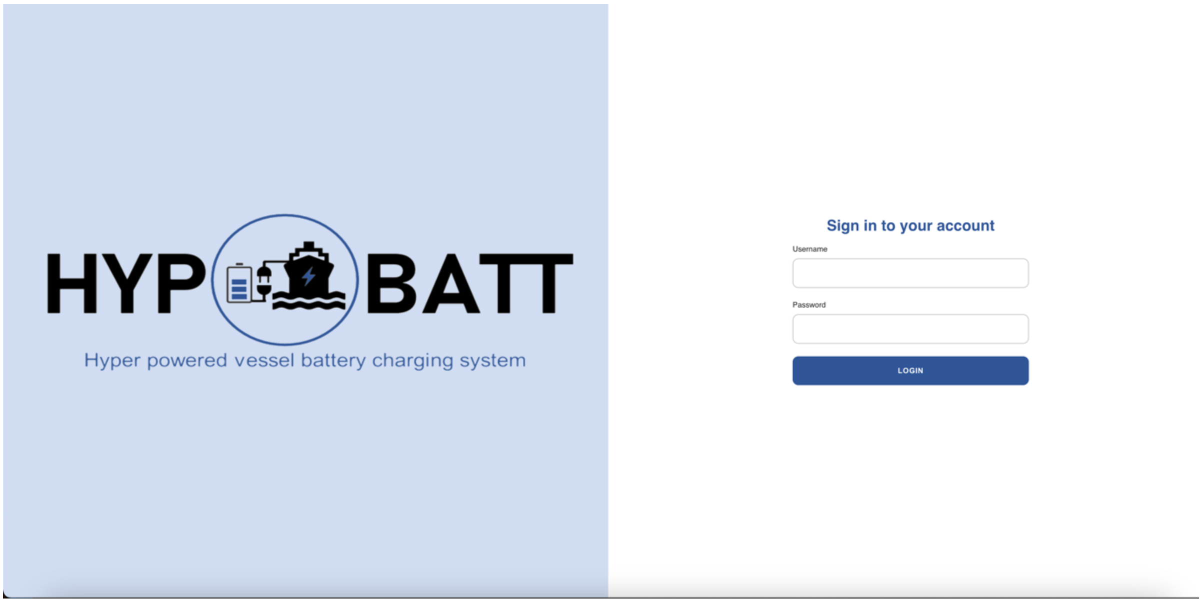

Figure 10 Login page

Admin Interface (*Figure* 11): The Admin interface is the most comprehensive, providing complete control and access to the system's backend functionalities. Admins can manage vessel information, oversee port operations, schedule charging times, and run simulations. They have the ability to add or remove vessels, edit charging parameters, and adjust pricing. The interface is designed to handle complex tasks such as user management, where admins can assign roles, set permissions, and control access to different parts of the system. Admins can also generate reports and export data, facilitating the deeper analysis required for strategic decision-making.

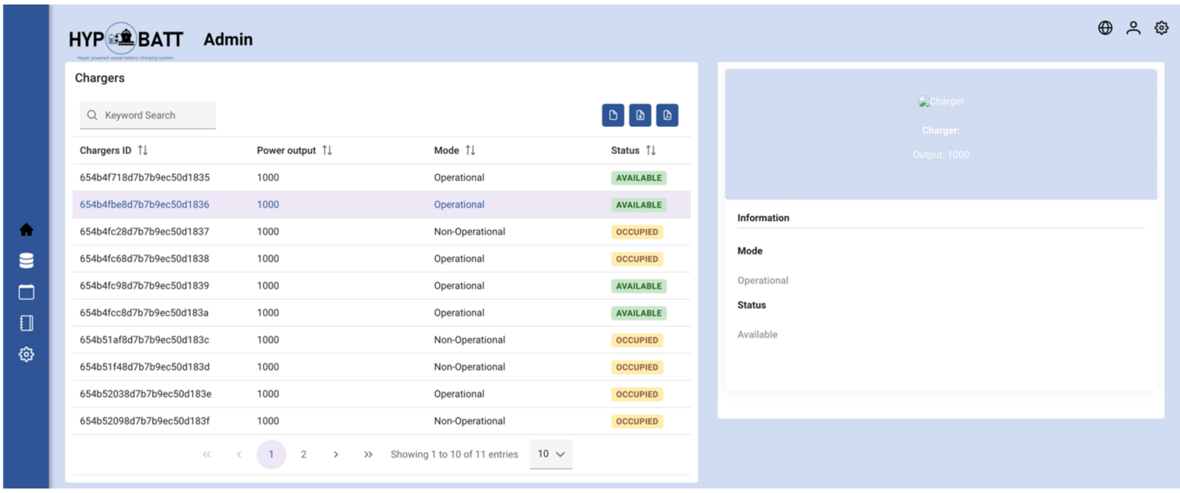

Figure 11 Admin interface screen.

Editor Interface (*Figure* 12): Editors have a more focused interface that allows them to manage day-to-day operational tasks without the broader system controls available to admins. This includes updating vessel details, managing port statuses, and inputting charging and pricing parameters. Editors can view schedules and are often responsible for ensuring that the data on the platform is current and accurate. Their interface is designed for ease of use, with intuitive forms and clear navigation that streamline the editing process. Editors can also access simulations and planning tools to assist with operational efficiency.

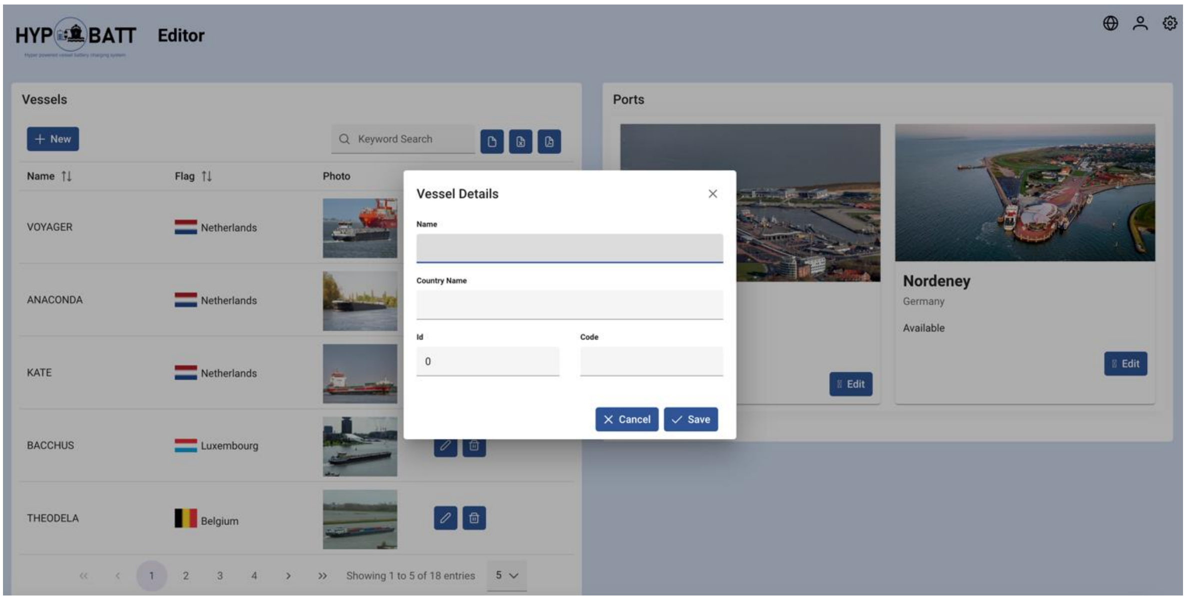

Figure 12 Editor interface screen.

Viewer Interface (*Figure* 13): Viewers have a streamlined interface designed for monitoring and overview. They can view vessel and port information but cannot alter them. The interface provides real-time updates on vessel locations, port availability, charger status, and energy consumption. Viewers can access scheduling calendars and pricing information, which are presented in easy-to-understand formats such as graphs and tables. The Viewer interface is optimized for clarity and simplicity, ensuring that stakeholders can quickly understand the operational state of the system without getting bogged down by the complexity of data entry or system management tasks.

Across all three interfaces, the design is consistent in terms of branding and layout, maintaining a unified look and feel that reinforces the HYPOBATT identity. Each interface layer is built with the same underlying structure, ensuring stability and performance, while the frontend displays are customized to meet the interaction needs of each user role. This role-based access control is a critical aspect of the system's security framework, ensuring that users only have access to the functions necessary for their role. Each interface is purpose-built to cater to the varying levels of interaction and control required by these roles, ensuring efficiency, security, and a seamless user experience.

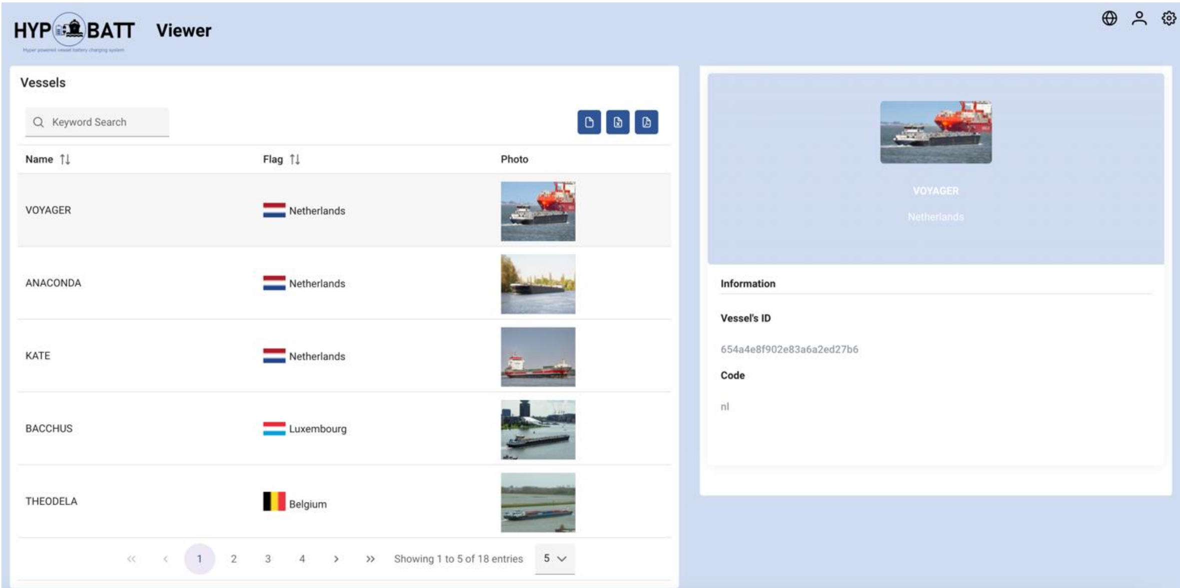

Figure 13 Viewer interface screen.

4.4 Safe Communications

To achieve the objective set at the beginning of this D2.3, i.e. properly demonstrate the whole WP2 digital solution for future Digital Port ecosystem, it was necessary to have the correct identification and communication of the data coming from the forecasted IoT edge-devices, coming from the real world to ensure the back-and-forth connectivity and data-feeding needs. The procedure followed to ensure this has been **Data Itemization**.

This work has framed all the needed data items that will need to be provided to the architected software solution to boost and ensure all its features and beneficial capabilities. Furthermore, the intended **Communication Protocol** addressed by both offshore and onshore OEMs have been preliminary treated with the respective OEMs.

This section consists of three subsections: In the first one, all the concept used in the data itemization are structured and explained. Subsections 2 and 3 follow by addressing the advancements achieved so far in ensuring both, feeding-data availability and its intrinsic communication channel safety, for the HYPOBATT WP2 port's solution.

4.4.1 Definitions for the WP2 Cloud data itemization

Data is described as follows:

Column A – Data item ID

The wide content of this column will be defined for both offshore and onshore data in the following subsections respectively.

Column B – Data Item Purpose (needed for)

- **DT operation**: This means that the data item is needed for generating the DT outcomes, that will generate the Port's EMS charging solution. The software will generate a daily solution as a baseline, and as many solutions as charging events will be requested, to ensure that an accurate solution, at the specific timings for each charging action is provided to the PC before the charging event.
- **Upper layer operation**: Data required for generating optimal solution for berthing times and vessel operativity.
- **DT optimization**: This data is needed for keep updating in-time the DT, meanwhile the charging process is taking place.
- **Overall Cloud operation**: This data is to be provided to the cloud operation for that specific charging event and vessel use-case.
- **Charging actions**: Data which needs to be provided to the Power Cabinets to facilitate and enhance the charging event.
- **Vessel captain plans and operations**: Data which needs to be provided to the E-Vessel to enhance the vessels plans.
- **Safety**: Implicit purpose to keep the system operation in a safety manner.

Column C - Event triggers

- **Triggered at "Charging request"**: This data exchange will be activated according to the request of any charging action.
- **Triggered at "EMS optimal solution found"**: These data items will be communicated once the optimal solution is found within the cloud for the specific charging event.
- **Triggered at "charging initiation"**: This data exchange will be emplaced when it is sent the starting signal for Charging phase (phase 4, see HYPOBATT Definitive Operational Procedure).
- **Triggered at "Connection confirmed" & Disconnection confirmed"**: These data items will be exchanged when it is sent the connected & disconnected signal for the end of Connection and Disconnection phase (phase 3 and 5, see HYPOBATT Definitive Operational Procedure).
- **Triggered at "Emergency Button Activated":** This data exchange will run when the forecasted "Emergency Push Button" (EPB, coded) will be activated.
- **Periodically**: The data exchange will be automatically executed *(see latency on next column D)*.

- **Port Static (manual)**: imputable Grid's fixed parameter.
- **Vessel Static (manual)**: imputable Vessel's fixed parameter.

Column D - Data exchange frequency

- **Single event**: explicitly defined.
- **Specific time values**: indicated in second (s).

Column E – Data packaging nature

The nature of the data packaging is addressed within this column. Further information will be enlarged for each data communication channel side (offshore/onshore).

Colum F – Assurance status as for the HYPOBATT demonstration

This column states the assurance status of data as for the final demonstration of HYPOBATT System at FRISIA. This critical matter has also been duly worked out in this preliminary stage and it is available upon EU representative's request.

4.4.2 Offshore Data Communication

Based on the baselines provided in the previous subsection, Table 1 shows the corresponding offshore data to be exchange between the solution presented and the Vessel receiver device.

As for the HYPOBATT use-case, to preliminary ensure the communication channel between the Electric Vessel and the Cloud architecture, a preliminary agreement (not binding at this stage by any means) has set this communication process as Modbus TCP (Transmission Control Protocol) / Ethernet / LoRaWAN (Long Range Wide Area Network)

WP2 involved partners will work in partnership with Vessel Procurement team, and FRISIA team, to bring this WP to a conclusion, framed within the scope of WP5.

Table 1 Data itemization for the high-level communications between the Electric Vessel and Port Authority Office

HYPOBATT Hyper powered vessel battery charging system		**TOPIC: *High-Level communication between Electric Vessel and Port Authority Office sub-systems***		
Edge Vessel device (EV) to Cloud-Based Architecture (PA)		***Comms nature:***	Modbus TCP | Ethernet | LoRaWAN	
Data Item	**Needed for**	**Event trigger**	**Frequency (s)**	**Data packaging**
Vessel ID	Overall Cloud operation	Vessel Static (Manual): Vessel's fixed parameter	Single event	Under discussions
Module/system SoC	DT operation & DT in-time Optimization	Triggered at "Charging request" & at "Charging initiation"	Single event & 60s [1min]	Under discussions
Module/System SoH	DT operation & DT in-time Optimization	Triggered at "Charging request" & at "Charging initiation"	Single event & 3600s [1hour]	Under discussions
Pack Voltage	DT operation & DT in-time Optimization	Triggered at "Charging request" & at "Charging initiation"	Single event & 60s [1min]	Under discussions
Cells/Modules Temperature	DT operation & DT in-time Optimization	Triggered at "Charging request" & at "Charging initiation"	Single event & 30s	Under discussions
Cells/Modules Enclosure Temperature	DT operation & DT in-time Optimization	Triggered at "Charging request" & at "Charging initiation"	Single event & 30s	Under discussions
BESS Overvoltage parameters	DT operation	Triggered at "Charging request" & at "Charging initiation"	Single event	Under discussions

Edge Vessel device (EV) to Cloud-Based Architecture (PA)		***Comms nature:***	ModBus TCP | Ethernet | LoRaWAN	
Data Item	**Needed for**	**Event trigger**	**Frequency (s)**	**Data packaging**
BESS Overcurrent parameters	DT operation	Triggered at "Charging request" & at "Charging initiation"	Single event	Under discussions
BESS Undervoltage parameters	DT operation	Triggered at "Charging request" & at "Charging initiation"	Single event	Under discussions
BESS (warning status (string level))	DT operation	Triggered at "Charging request" & at "Charging initiation"	Single event	Under discussions
Vessel Timetable (routes from port-to-port, pier to pier)	Upper layer operation	Vessel Static (Manual) [fixed parameter]	86400s [24h]	Under discussions
Weather margin	Upper layer operation	Vessel Static (Manual) [fixed parameter]	86400s [24h]	Under discussions
Average Speed Input per waterway/Route (shallow/deep part/arrival/departure)	Upper layer operation	Vessel Static (Manual) [fixed parameter]	Single event	Under discussions
Move to charger time	Upper layer operation	Vessel Static (Manual) [fixed parameter]	Single event	Under discussions
SEA Current	Upper layer operation	Vessel Static (Manual) [fixed parameter]	Single event	Under discussions
Vessel hotel load	Upper layer operation	Vessel Static (Manual) [fixed parameter]	Single event	Under discussions
Manoeuvring power	Upper layer operation	Vessel Static (Manual) [fixed parameter]	Single event	Under discussions
Manoeuvring time	Upper layer operation	Vessel Static (Manual) [fixed parameter]	Single event	Under discussions

Edge Vessel device (EV) to Cloud-Based Architecture (PA)		***Comms nature:***	ModBus TCP | Ethernet | LoRaWAN	
Data Item	**Needed for**	**Event trigger**	**Frequency (s)**	**Data packaging**
System efficiency	Upper layer operation	Vessel Static (Manual) [fixed parameter]	Single event	Under discussions
Safety margin	Upper layer operation	Vessel Static (Manual) [fixed parameter]	Single event	Under discussions
Charging target SoC	Upper layer operation	Vessel Static (Manual) [fixed parameter]	Single event	Under discussions
Cloud-Based Architecture (PA) to Edge Vessel device (EV)		***Comms nature:***	ModBus TCP | Ethernet | LoRaWAN	
Data Item	**Needed for**	**Event trigger**	**Frequency (s)**	**Data packaging**
Forecasted optimal charging profile and timing (including targeted SoC)	Vessel captain plans and operations	Triggered at "EMS optimal solution found"	Single event	raw data [hex] / decoded data [txt] / structured data [csv]
Forecasted optimal charging power (modules used)	Vessel captain plans and operations	Triggered at "EMS optimal solution found"	Single event	structured data [csv]
Technical data set of the charging planned	Vessel captain plans and operations	Triggered at "EMS optimal solution found"	Single event	Graphical data
Optimal Price of recharge	Vessel captain plans and operations	Triggered at "EMS optimal solution found"	Single event	raw data [hex] / decoded data [txt] / structured data [csv]
Pier ID where to moor	Vessel captain plans and operations	Triggered at "EMS optimal solution found"	Single event	decoded data [txt]

4.4.3 Onshore Data Communication

Based on the baselines provided in the previous subsection, Table 2 shows the corresponding offshore data to be exchange between the solution presented and the Power Cabinet edge-device.

As for the HYPOBATT use-case, to preliminary ensure the communication channel between the Electric Vessel and the Cloud architecture, a preliminary agreement (not binding at this stage by any means) has selected ethernet for communication, underpinned by the OCPP 1.6J protocol.

WP2 involved partners will work in partnership with the Power Cabinet manufacturer and HELIOX team to bring this WP to a conclusion, framed within the scope of WP5.

Table 2 Data itemization for the high-level communications between the Power Cabinet and Port Authority Office.

HYPOBATT Hyper powered vessel battery charging system		**TOPIC: *High-Level communication between Power Cabinet and Port Authority Office sub-systems***		
Edge Power Cabinet device (PC) to Cloud-Based Architecture (PA)		***Comms nature:***	Protocol OCPP 1.6J | Ethernet	
Data Item	**Needed for**	**Event trigger**	**Frequency (s)**	**Data packaging**
Voltage and Current references for CC/CV	DT operation	Triggered at "Charging request"	Single event	JSON (OCPP 1.6j)
Grid Voltage	DT operation	Periodically	30s	JSON (OCPP 1.6j)
Grid Frequency	DT operation	Periodically	30s	JSON (OCPP 1.6j)
Grid Current	DT operation	Periodically	30s	JSON (OCPP 1.6j)
Grid Power Factor	DT operation	Periodically	30s	JSON (OCPP 1.6j)
Ambient Temperature	DT operation	Periodically	30s	JSON (OCPP 1.6j)
Bus Voltage	DT operation	Periodically	30s	JSON (OCPP 1.6j)
Bus Current	DT operation	Periodically	30s	JSON (OCPP 1.6j)
DC-Link Capacitor	DT operation	Port Static (manually) [Grid fixed parameter]	Single event	JSON (OCPP 1.6j)
Grid L Filter Inductance	DT operation	Port Static (manually) [Grid fixed parameter]	Single event	JSON (OCPP 1.6j)
Grid L Filter Resistance	DT operation	Port Static (manually) [Grid fixed parameter]	Single event	JSON (OCPP 1.6j)

Edge Power Cabinet device (PC) to Cloud-Based Architecture (PA)		***Comms nature:***	Protocol OCPP 1.6J | Ethernet	
Data Item	**Needed for**	**Event trigger**	**Frequency (s)**	**Data packaging**
Pre-Charge Parameters	DT operation	Triggered at "Charging request"	Single event	JSON (OCPP 1.6j)
Grid Connection times	DT operation	Triggered at "Charging request"	Single event	JSON (OCPP 1.6j)
Number of transformers/converters to be used	DT operation	Triggered at "Charging request"	Single event	JSON (OCPP 1.6j)
Power Losses	DT in-time optimization	Periodically	60s [1 min]	JSON (OCPP 1.6j)
Energy index	DT in-time optimization	Periodically	10s	JSON (OCPP 1.6j)
Charging starting moment & detention moment	DT optimization	Triggered at “Connection confirmed” & "Disconnection confirmed”	Single event	JSON (OCPP 1.6j)
Cloud-Based Architecture (PA) to Edge Power Cabinet device (PC)		***Comms nature:***	Protocol OCPP 1.6J | Ethernet	
Data Item	**Needed for**	**Event trigger**	**Frequency (s)**	**Data packaging**
Charging profile	Charging actions	Triggered at “EMS optimal solution found”	Single event	JSON (OCPP 1.6j)
Charging duration	Charging actions	Triggered at “EMS optimal solution found”	Single event	JSON (OCPP 1.6j)
Forecasted Nº of power modules (1,25MW each) needed for the charging action	Inputs coming from the Vessel to the PC (slave to the Vessel)	Triggered at “EMS optimal solution found”	Single event	JSON (OCPP 1.6j)
Signal for Emergency mode coming from the Port Operator	Safety	Triggered at "Emergency Button Activated"	Single event	Under discussions

4.4.4 Cybersecurity Aspects

Maritime ports serve a critical function in facilitating domestic and international supply-chain activities by connecting sea and inland transport services, playing a strategic role in supporting international trade and economic growth.

Port operations are increasingly dependent on digitization, integration and automation systems in order to improve the efficiency of the current processes, which makes the ports more exposed to risks related to cyber-attacks. Therefore, cybersecurity is of paramount importance and the protection against cyber-attacks has become a top priority for the ports, se Figure 14.

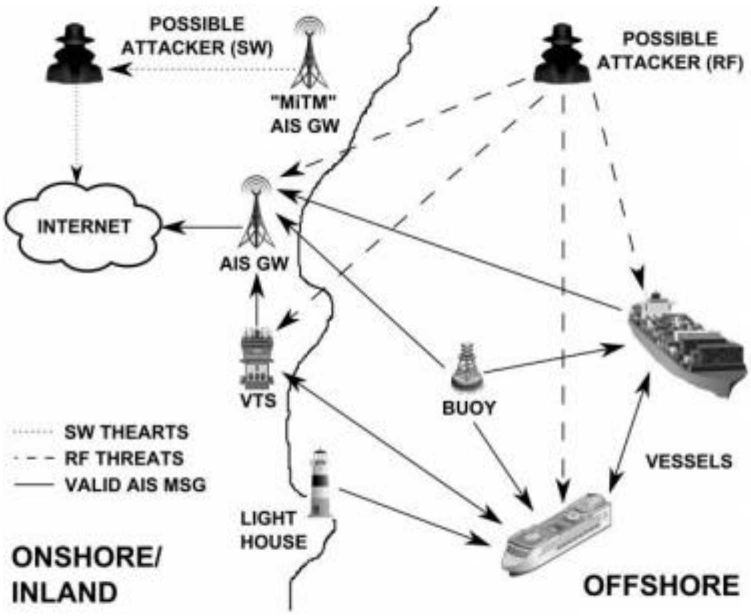

Figure 14 Cybersecurity diagram

Successful breaches in cyber security can cause a significant disruption of services and even introduce safety risks. This is the reason why the development of all the HYPOBATT Digital-Twin components must follow cyber-security best practices, from the collection of data in the on-shore and off-shore elements up to the storage and visualization in the cloud platform.

Securing IoT devices is crucial to protect against potential cybersecurity threats and vulnerabilities. Here are some best practices for enhancing the cybersecurity of IoT devices:

- **Change default credentials**, many attacks target devices with default usernames and passwords.
- **Strong authentication and access to the devices**, implement strong authentication methods, such as Multi-Factor Authentication (MFA), when possible and ensure that only authorized users can access the device.
- **Firmware and software updates**, regularly update firmware and software on IoT devices to patch known vulnerabilities.

- **Network segmentation and security**, isolate IoT devices on separate network segments to limit the potential impact of a compromised device on the rest of the network and employ network security practices.

The communication between the IoT devices collecting the data and the cloud platform is another important cybersecurity aspect that should be addressed. The MQTT protocol has been the one selected for the data transmission and this communication and to ensure the security of MQTT-based systems, several security requirements and best practices should be followed. Here are some of the key MQTT protocol security requirements:

- **Authentication:** MQTT brokers should require clients to authenticate themselves before allowing them to publish or subscribe to topics. Authentication mechanisms can include username/password, client certificates, or API keys.
- **Access Control:** MQTT brokers should implement access control mechanisms to restrict which clients can publish or subscribe to specific topics. Access Control Lists (ACLs) or Role-Based Access Control (RBAC) can be used to define and enforce access policies.
- **Encryption:** Data exchanged between MQTT clients and brokers should be encrypted to protect it from eavesdropping. The use of TLS/SSL (Transport Layer Security/ Secure Sockets Layer) for secure communication is essential to ensure data privacy and integrity.
- **Secure Transport:** Ensure that MQTT brokers and clients use secure transport mechanisms like TLS/SSL to encrypt data in transit. This prevents attackers from intercepting or tampering with messages**.**

Cloud platform security is a shared responsibility between the cloud provider and the customer. The provider is responsible for the security of the cloud infrastructure, while the customer is responsible for the security of their applications, data, and configurations. The following best practices will increase the security of the application deployed in the cloud:

- **Identity and Access Management (IAM):** Implement the principle of least privilege (only grant permissions necessary for each user or role). This approach should be followed both for access to the application, providing just the necessary right for the users according to the actions that should perform, and for the access of the virtual private servers.
- **Network Security**: implement security groups and network ACLs to control inbound and outbound traffic to and from the cloud resources.
- **Patch and regularly** update the cloud resources, including server, databases, and other services.
- Perform regular **data backups** to ensure business continuity in case of disaster.

Finally, it is highly recommended to conduct code security audits to identify vulnerabilities in the software.

4.5 Cloud and Safe Communications Integration

The primary aim of this section is to establish a clear understanding of the connectivity between the cloud platform and the physical application. Currently, the intricacies of communication are being meticulously delineated by key project stakeholders, including the charger manufacturer (HELIOX – WP3), the automatic connection device (STT – WP4), and the port and vessel owner responsible for system validation (FRISIA – WP5), with valuable support from DAMEN, BRING, and IKERLAN.

It is important to note that this aspect of the project remains an ongoing effort, and some crucial details are yet to be fully defined. These details encompass elements such as variables mapping and the communication protocol, which play pivotal roles in ensuring seamless integration. As the project continues to unfold, the collaborative work on this aspect will progress concurrently, aligning closely with the developments within WP3, where it forms an integral part of the broader framework. This coordinated effort reflects the commitment of the project team to deliver a robust and effective cloud-based solution that seamlessly interfaces with the physical application, while staying adaptable and responsive to evolving requirements.

4.6 Full-Solution Validation

For full solution validation of the system, data from the use case of FRISIA is used. The relevant data is provided in deliverable 1.2 and summarized in Table 3. Through the human interface, this data is entered into the system. After this, the back-end upper control layer can be used to create charging profiles for day planning. To be able to fully test the back-end lower control layer, additional test components, virtual chargers and vessels replacing the physical charger and vessel, are required to simulate the interaction visualized in *Figure* 1. With this test data and procedure, the correct operation of the software system can be verified.

<table>
<tr><td>Ports</td><td>Norddeich
Charger power availability varies with P_{max} = 2.5MW (800kW power limitation of the grid)
Electricity price varies throughout the day.
Norderney
Charger with P_{max} = 5MW
Fixed electricity price of 0.07 euro/kWh</td></tr>
<tr><td>Routes & Schedule</td><td>Norddeich -> Norderney
Departure scheduled at 6h00, 8h00, 10h00, 13h00, 15h00,17h00,19h00
Route stages:
1min of maneuvering Norddeich: requires 210kW and 4kWh
27min of sailing requires 1000kW and 350kWh
1min of maneuvering at Norderney: requires 210kW and 4kWh
Berth at Norderney for 31min requires 22kW and 12kWh
Norderney -> Norddeich
Departure scheduled at 6h50, 8h50, 10h50, 13h50,15h50,17h50, 19h50
Route stages:
1min of maneuvering Norderney: requires 210kW and 4kWh
27min of sailing requires 1000kW and 350kWh
1min of maneuvering at Norddeich: requires 210kW and 4kWh
Berth at Norddeich for 31min requires 22kW and 12kWh
At least two vessels are required for these routes and schedule.</td></tr>
<tr><td>Vessels</td><td>Damen Fast Ferry 3209
Battery capacity 1.8MWh</td></tr>
</table>

Table 3 Input data for full-solution validation from D1.2.

However, at the time the report, not all connections between the different back-end and front-end components are present yet. Additionally, the virtual chargers and vessel components, replacing the physical components, are still to be added. Therefore, more actions are currently taken to provide these and to execute the aforementioned test. Results of these will be reported in future updates to this report.

However, partial tests have been conducted for the back-end upper and lower control layers. These tests and their results are discussed in the next sections.

4.6.1 Back-End Upper Control Layer Tests

This test validates the general optimization of the electric vessel, defined in the EMS and trying to reduce the total cost.

4.6.1.1 Input

The vessel input parameters for testing are:

- **Vessel Average Speed**
    - Departure: Random values between 9 and 11 knots
    - Shallow part: Random values between 16 and 18 knots
    - Deep part: Random values between 10 and 12 knots
    - Arrival: Random values between 8 and 12 knots
- **Current Towards Sea**
    - An array with alternating values of -2 and 2 (e.g., -2,2, -2,2, etc.)
- **Time to Connect to Charger**
    - Set to 30 seconds.
- **Weather Margin**
    - Set to 0

4.6.1.2 Results

Optimization Method

The linear approach was selected as the optimization method for calculating the optimal charging strategy.

Energy Consumption Estimation

Correlating with the vessel's average speed per waterway the algorithm calculates the energy consumption for the planned route. This calculation accounts for varying speeds of the waterway (departure, shallow part, deep part, arrival) and external factors such as current towards sea and weather margin (Figure 15).

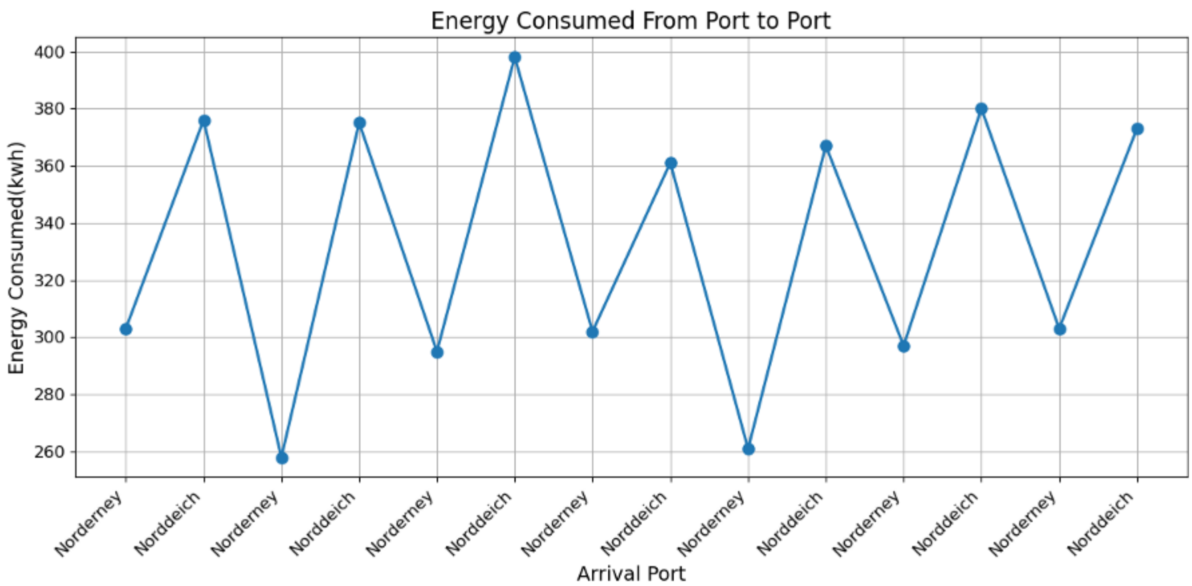

Figure 15 Energy Consumed from Port-to-Port diagram.

Strategy Identification

Utilizing the input hourly electricity pricing data, and the available charging power in each port, the algorithm successfully determines an optimal charging schedule. This strategy focuses on maximizing energy intake during periods of lower electricity costs (Figure 16, Figure 17, Figure 18).

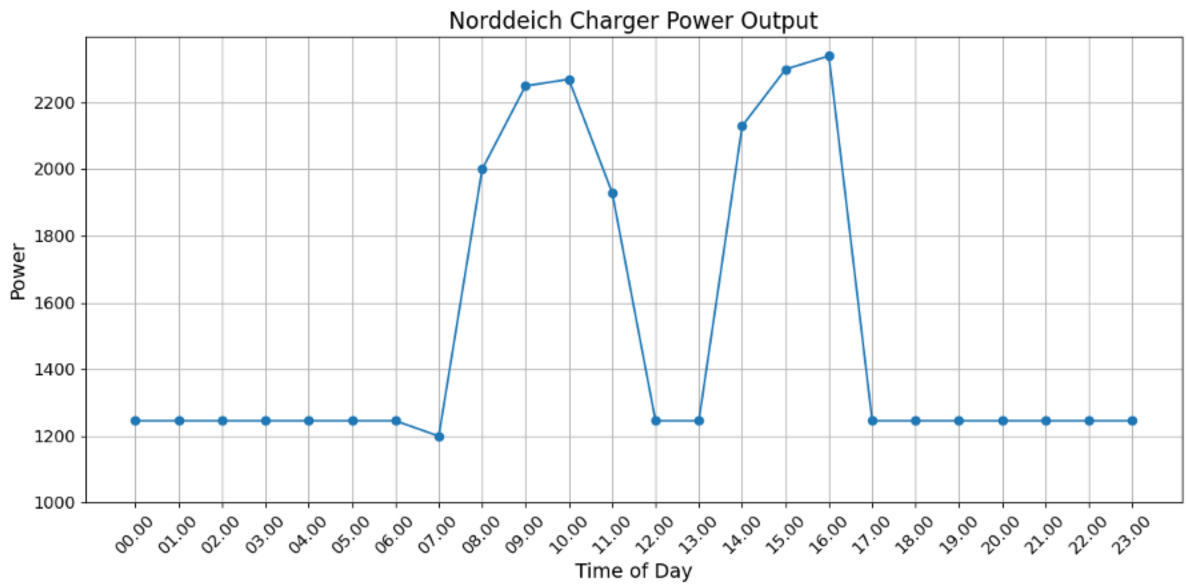

Figure 16 Norddeich Charger Power Output diagram.

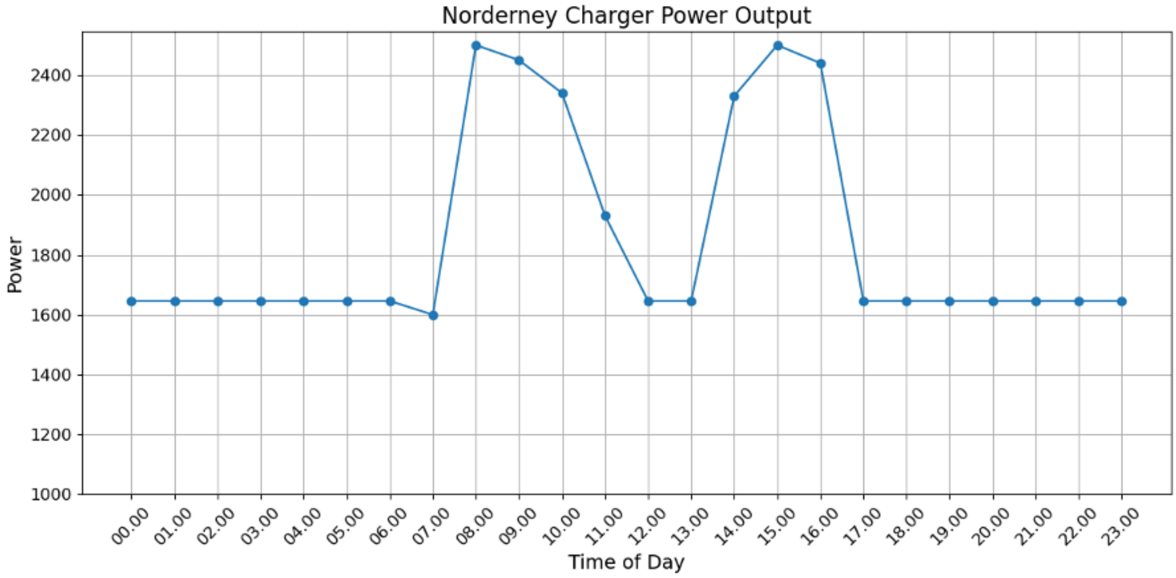

Figure 17 Norderney Charger Power Output diagram.

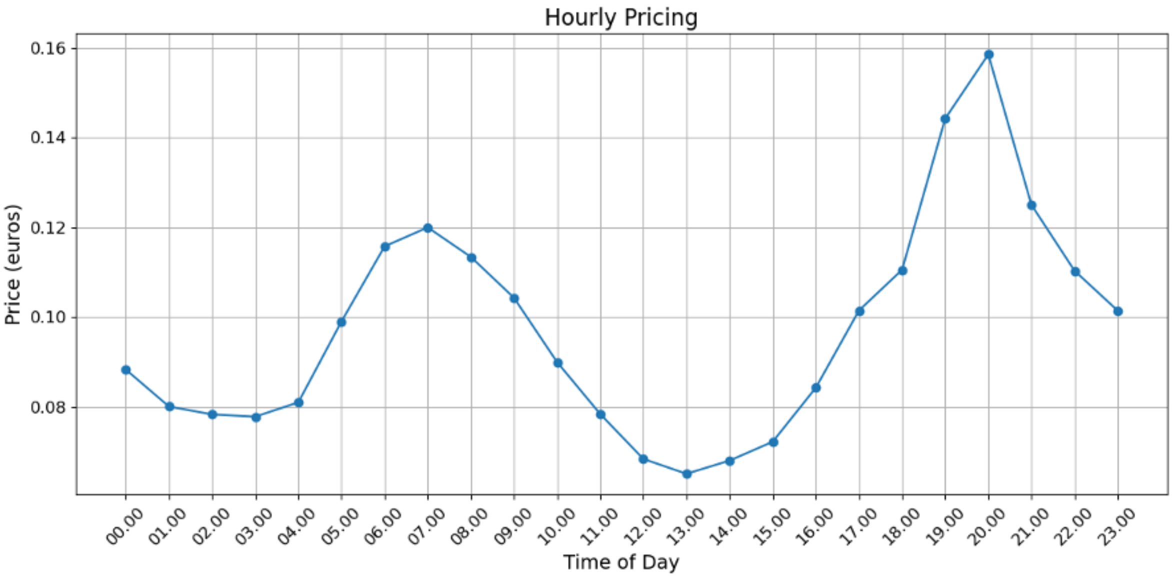

Figure 18 Hourly Electricity price diagram.

As shown at the Figure 19 below, the algorithm identifies a distinct charging pattern, where the majority of energy is estimated to be charged during the hours when the electricity prices are at their lowest, highlighting the cost efficiency objective.

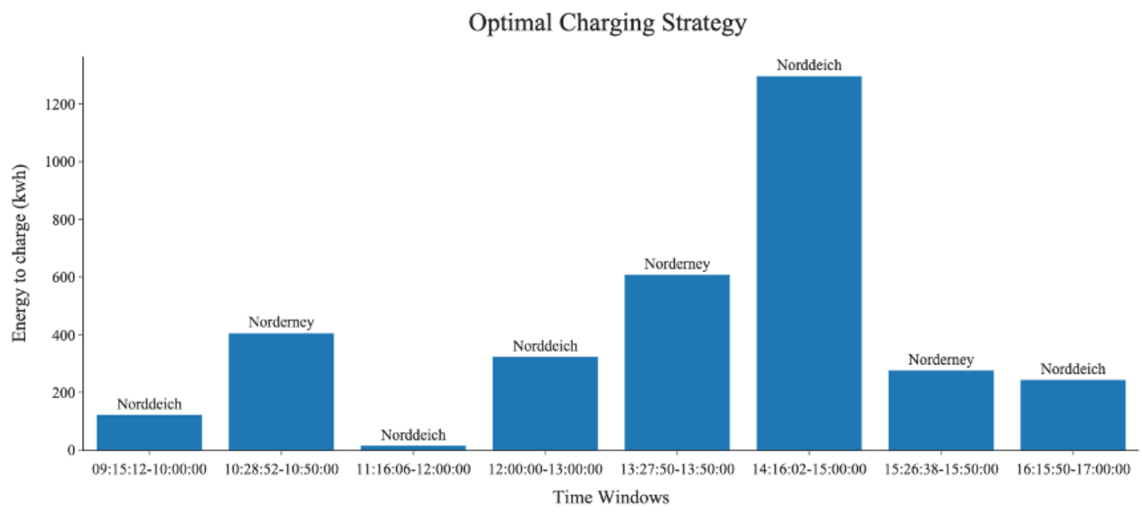

Figure 19 Optimal Charging Strategy diagram.

The provided table (Table 4) offers a comprehensive overview of the charging events and associated costs at Norddeich and Norderney ports considering both chargers available within specified time windows. The data indicates that a total of 3295 kWh is estimated to be charged over the course of the day, distributed across time slots to leverage the fluctuating electricity rates. The corresponding total cost for these events amounted to 241.09 euros.

Port	Time window	Charge events (khw)	Cost (euros)
Norddeich	09:15:12-10:00:00	123.00	12.83
Norderney	10:28:52-10:50:00	406.00	36.53
Norddeich	11:16:06-12:00:00	16.00	1.25
Norddeich	12:00:00-13:00:00	324.00	22.16
Norderney	13:27:50-13:50:00	608.11	39.56
Norddeich	14:16:02-15:00:00	1296.89	88.20
Norderney	15:26:38-15:50:00	277.00	20.00
Norddeich	16:15:50-17:00:00	244.00	20.56
	Total	***3295.0***	***241.09***

Table 4 Overview of charging events and associated costs at Norddeich, Norderney.

4.6.2 Back-End Lower Control Layer Tests

After getting the data through the upper layer scripts, the lower layer and the digital twin were run. To understand how the script was operating, the same vessel initial conditions were provided to the lower that were provided to the upper, emulating the physical signal obtained from the physical system. With the generated outputs from the upper control layer, one can run distinct instances of the DT and the lower control layer to provide a different charging scenario for each charging event drafted by the upper control layer, with already defined set points and initial conditions. Using the information provided by Figure 19, 8distinct charging events were simulated, with 8 different charging strategies taking into account the current SOC status at each moment, provided by the upper layer algorithm, considering the initial temperature before charging constant.

The algorithm was also tested to its capabilities of successfully uploading the daily results to the database, and then the DataLake, which was verified. The next steps are ensuring a constant flow of information from the DataLake to the digital twin to provide needed data from previous days, like end-of-day capacity, to follow the degradation path of the battery pack. With a flowing algorithm structure, the validation of the connection between the back end and front end can be pursued and the physical connection can be created through dummy data.

5. DISSEMINATION, EXPLOITATION AND STANDARDISATION

In terms of dissemination, the focus is on sharing the development and results of this operational optimization initiative with a wider audience. A key aspect of this dissemination strategy involves creating a public deliverable that will be submitted to the European Commission. This document will serve as a comprehensive overview of the project's objectives, methodologies, and, most importantly, the outcomes achieved. Simultaneously, the project website will be a dynamic platform where regular updates and insights into the ongoing progress will be made available to a broader audience.

Moreover, the project team plans to leverage the scientific community's knowledge and expertise in the field of digitalization and optimization for recharging structures in marine ports. This will be achieved by preparing a scholarly publication that delves deeper into the technical aspects of the project and its implications. The intention is to present this publication at a relevant industry conference, where experts, stakeholders, and peers can engage in discussions and share valuable insights.

These dissemination activities are centrally coordinated within Work Package 8 (WP8), and further dissemination plans and actions are outlined in the same work package. The goal is to ensure that the project's findings and advancements are not only shared with relevant authorities but also contribute to the broader scientific discourse in the maritime industry.

The work focusing on the exploitation and standardisation of recharging infrastructure promises to be a pivotal step towards future installations. This initiative showcases how recharging processes can be fine-tuned, effectively minimizing time requirements while simultaneously ensuring the longevity of Energy Storage Systems (ESS). What's remarkable is that these advancements are achieved through a cloud-based platform, eliminating the need for additional server installations. Notably, this solution is set to undergo experimental validation in the port of Norddeich by FRISIA, a practical demonstration of its real-world applicability. Such a trial run underscores the system's robustness and practicality.

Furthermore, the beauty of this innovation lies in its scalability; it's designed to be readily expanded and implemented in other ports where electric vessels are in widespread use. As electric marine transportation gains momentum, this standardized, cloud-based recharging infrastructure could play a significant role in supporting the growing fleet of electric vessels, making it a noteworthy advancement in sustainable maritime transportation.

6. RESULTS AND DISCUSSION

In this deliverable, we have presented the implementation of HYPOBATT cloud platform, which marks a significant milestone in our project. At the core of our development are ports equipped with similar implementations for vessel management. What sets this project apart is the integration of an advanced Energy Management Strategy (EMS) and a Digital-Twin that can operate in real-time on a single, unified platform. This platform offers the flexibility to accommodate diverse source codes, allowing them to run seamlessly alongside each other.

One of the standout features of our project is the continuous synchronization between the real application and the Digital-Twin. This synchronization enables us to maintain an extremely precise virtual model of the real-world application, which in turn is leveraged by our control strategies. This real-time feedback loop provides invaluable insights for optimization, making our system highly adaptable and efficient.

Furthermore, all the data generated within our platform is meticulously stored in a DataLake. This rich repository of data serves as a valuable resource for ongoing optimization processes, ensuring that we can fine-tune our strategies and improve overall system performance.

This achievement was made possible through the collaborative efforts of our task partners and their valuable contributions. RHOE played a pivotal role by developing the front-end interface and implementing the DataLake, making data access and management more user-friendly. BRING focused on defining safe communication procedures and addressing critical cybersecurity aspects, with valuable support from FV. The full-system validation was expertly conducted by FM, with crucial assistance from CEA, which was also responsible for the development of novel optimization algorithms. Finally, IKERLAN took on the critical task of developing the backend, bringing together all the components mentioned above and integrating the outcomes of tasks T2.1 (Digital-Twin) and T2.2 (Energy Management Strategy).

This collaborative effort highlights the synergistic nature of our project, where each partner's expertise and contributions have converged to create a robust and innovative platform for vessel management and energy optimization. We look forward to further refining and expanding our platform to benefit the maritime industry and beyond.

7. CRITICAL RISKS

This section provides a summary of critical risks, derived from the earlier WP2 tasks (T2.1 and T2.2) and new ones from the current task. Task T2.3 is consolidating these findings, making it easier for stakeholders to understand and address the key risks impacting the project. Note that, these risks are also summarized in the monthly meeting technical report, see Table 5.

ID	WP/Task	Risk Item	Type	Probability	Impact	Severity	Counter Measures	Risk Owner
3	WP1 WP2 WP3 WP5 WP6	Battery lifetime degradation in demonstration not in accordance with estimated and required to achieve the calculated TCO	Technical	Moderate	Major	Medium-High	Improved charging profiles and thermal preconditioning of battery pack controlled by the charger to minimize effect of charging on degradation	BRING OTASKIES
16	WP2	Availability of Ageing data for the used cells (NMC) if the battery provider do not want to disclose the data	Technical	Likely	Major	Medium-high	Exploring in Houses data (IKERLAN, BRING, FM, AB Members)	BRING WP1 WP2
27	WP2	Heliox and Frisia cannot provide data for create accurate models & Inputs from T3.1 and 3.2 to 2.1 are needed but tasks did not start yet	Technical	Likely	Major	Medium-High	Meetings to find solutions with partners from each one of the three sides (onboard batteries, offboard chargers and grid side)	BRING

44	WP2 WP3 WP4 WP5 WP9	Communications layout between the charger/cloud/acd´s point of view do not match	Technical	Moderate	Extreme	High	Weekly meeting addressing all integration challenges in advances are taking place under scrum methodology to boost a clear picture before the designs go freeze	STT HELIOX DAMEN FRISIA FV IKERLAN BRING
50	WP2 WP5	Data itemization to feed the cloud solution is not properly available	Technical	Likely	Major	Medium-High	This potential problem which will jeopardize the WP2 demonstration in FRISIA has been taken into consideration by BRING, who is ensuring that off/onshore edge-devices owners	WP2 WP5
51	WP2 WP5	Validation of WP2 cloud software is impossible without getting involved the owner of the cloud architectural solution into T5.4	Technical	Likely	Extreme	High	IKERLAN will explore the allocation of PMs into the WP5 T5.5, to fine-tune the WP2 solution for FRISIA in-site, besides BRING, when it will be the time to do so	IKERLAN BRING FRISIA
52	WP2 WP5	Too long of a time to provide a charging profile to the physical system (computational cost)	Technical	Moderate	Major	Medium-High	Model simplification and/or physical data gathering with more time in advance	BRING IKERLAN RHOE

53	WP2 WP5	Digital-Twin does not accurately follow the physical behaviour of the system	Technical	Likely	Moderate	Moderate	Algorithms will be developed with focus on adaptability of the DT to new physical responses	BRING IKERLAN
54	WP2	No capability to trigger emergency procedures embedded in the cloud solution	Technical	Likely	Trivial	Low-Medium	The needed emergency concerns will be evaluated and added into the already done work	IKERLAN RHOE

Table 5 Critical risks summary (T2.1, T2.2 and T2.3).

8. CONCLUSIONS

The comparison between different cloud-based architecture resources has been carried out as an initial analysis, prior to the implementation of the chosen model. Finally, the AWS solution has been chosen for its capabilities and its globalised use in the cloud model.

Initially, the idea was to create a docker image of each block, i.e. of the front-end and EMS in the upper control layer and of the Digital Twin in the lower control layer, both using the Linux operating system. Finally, it has been decided to replace the dockerized image of the digital twin on Linux with an executable that is launched via Windows within AWS.

Some difficulties have been encountered when integrating the DT model in the lower control layer, developed in MATLAB/SIMULINK, with the front-end with back-end in the upper control layer, developed in Python and JavaScript. Finally, it has been solved by creating an executable of the DT model that allows it to be executed on demand iteration loop, thus integrating it into the energy manager system environment as an API. This API enables to iterate into the DT and connect it with the MongoDB database that interconnects the external input data and the DT model outputs.

The integration of both lower and upper control layer into the cloud architecture, offers a single platform that has the flexibility to integrate the different source codes for run them smoothly.

This project is ongoing, and crucial details, such as variable mapping and communication protocols, are still being defined. These aspects are pivotal for seamless integration and are progressing collaboratively with rest of partners involve in the system development. The coordinated effort reflects the team's commitment to delivering a robust, cloud-based solution that interfaces seamlessly with the physical application, remaining adaptable to evolving requirements. This task summarized the work with the information accessible, and cloud platform will be improved in the upcoming steps once the rests of the elements of the systems are completed and final information available.