EUROPEAN COMMISSION - **HORIZON EUROPE PROGRAMME**

CLUSTER CLIMATE, ENERGY AND MOBILITY
TOPIC HORIZON-CL5-2022-D5-01-01

Co-funded by
the European Union

NEMOSHIP

NEW **MO**DULAR ELECTRICAL ARCHITECTURE
AND DIGITAL PLATFORM
TO OPTIMISE LARGE BATTERY SYSTEMS ON **SHIP**S

GRANT AGREEMENT No. 101096324

D2.1: Report on specifications and infrastructure definition for the digital tool platform

December 2023

Report details

Deliverable No.	2.1
Deliverable Title	Report on specifications and infrastructure definition for the digital tool platform
Deliverable Date	31.12.2023
Dissemination Level	PU (Public report)
Author	Dan PESCARU, Daniela SASU, Călin HUSAR (Siemens)
Contributors	Dai-Duong TRAN (VUB), Lars Petter LARSEN (Corvus)
WP Leader	Călin HUSAR (Siemens)
Reviewers	Dai-Duong TRAN (VUB), Erdeniz EROL, Yunus Emre BAYGELDI, Nazlı Nur ÇETIN (Elkon)
Coordinator	Guénaël LE SOLLIEC, Solène GOY (CEA)

Document History

Version	Date	Editors	Description
1.0	01-11-2023	Siemens	Initial template
1.1	20-11-2023	Siemens	Structure updated
2.0	30-11-2023	VUB, Corvus	Partner contribution
3.0	11-12-2023	Siemens	Aligned contributions - Complete version ready for internal review
3.1	20-12-2023	VUB, Elkon	Internal review
3.2	31-12-2023	CEA	Final review

Disclaimer

Co-funded by the European Union

This project has received funding from the European Union's Horizon Europe Research and Innovation program under grant agreement No 101096324. The information and views set out in this publication does not necessarily reflect the official opinion of the European Commission. Neither the European Union institutions and bodies nor any person acting on their behalf, may be held responsible for the use which may be made of the information contained therein.

Project Summary

The ambition of the NEMOSHIP project is to develop, test and demonstrate new innovative technologies, methodologies, and guidelines to better optimize large electric battery power technology within hybrid and fully electrically powered ships. The project will act as a key enabler of the new co-programmed European Partnership Zero Emission Waterborne Transport (ZEWT) roadmap to better reach International Maritime Organization (IMO) objectives regarding the reduction of Greenhouse Gas (GHG) emissions from waterborne transport by 2030 and 2050.

To help achieve this ambition, NEMOSHIP will develop a modular and standardized battery energy storage solution that is able to exploit heterogeneous storage units and a cloud-based digital platform to enable data-driven, optimized, and safe exploitation. The project will demonstrate the maturity of these innovations at Technology Readiness Level (TRL) 7 for hybrid ships as well as their adaptability towards fully electric ships thanks to the observations collected from a retrofitted hybrid Offshore Service Vessel (OSV) (diesel/electric propulsion), a newly designed hybrid cruise vessel (LNG/electric propulsion) and a semi-virtual demonstration of two additional fully electric vessels employed in tasks such as ferrying and short-sea shipping.

The NEMOSHIP consortium estimates that these innovations will contribute to the electrification of about 7% of the European fleet by 2030 and the reduction by 30% of EU maritime GHG emissions compared to a business-as-usual scenario.

The NEMOSHIP consortium is composed of 11 partners (3 RTO, 1 SME, 7 large companies) from six European countries and covers the whole value chain, from research-oriented partners to software developers, energy system designers, integration partners, naval architects, and end-users.

Report summary

The ambition of the NEMOSHIP project is to develop, test and demonstrate new innovative technologies, methodologies and guidelines to accelerate large BESS deployment and optimal exploitation toward 2030 on both hybrid and full-electric arrangement. To help achieve this ambition, NEMOSHIP will develop a modular and standardised battery energy storage solution that is able to exploit heterogeneous storage units and a cloud-based digital platform to enable data-driven, optimised, and safe exploitation. The project will demonstrate the maturity of these innovations at TRL 7 for hybrid ships as well as their adaptability towards fully electric ships. This report presents the proposed digital platform's design and operational framework which will be developed and implemented as part of the NEMOSHIP project.

A fundamental aspect of this architecture is related in providing a comprehensive view of a ship's energy systems, empowering operators with precise data and predictive analytics for informed decision-making. The proposed platform architecture supports multiple functionalities, from monitoring battery health to managing energy distribution, ensuring optimal system performance and durability, thereby reducing environmental impact.

With robust support for monitoring, managing energy distribution, and consumption patterns, it ensures optimal system performance, fostering operational efficiency and environmental responsibility.

Advanced security protocols and encryption techniques safeguard sensitive maritime data, aligning with industry cybersecurity standards. The platform's interoperability with existing maritime infrastructure facilitates a smooth transition to data-driven operational models, setting a new industry benchmark for digital solutions.

This report delves deeper into the platform's architectural components, functional capabilities, and strategic implementations, showcasing its essential role in steering the maritime industry towards a sustainable, efficient future.

Table of contents

List of acronyms

Abbreviation	
AFE	Active Front End
AI	Artificial Intelligence
API	Application Programming Interface
AWS	Amazon Web Services
Azure AD B2C	Azure active directory business to customer identity service
BESS	Battery Energy Storage Systems
CLI	Command Line Interface
DT	Digital Twin
EPS	Electrical Propulsion System
GBM	Gradient Boosting Machines
GHG	Greenhouse Gas
gRPC	Google's Remote Procedure Call Framework
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secured
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
IMO	International Maritime Organization
KNN	K-Nearest Neighbors
LNG	Liquefied natural gas
MDTA	Maritime Digital Twin Architecture
.NET	Microsoft .Net Framework
OData	Open Data Protocol
OSV	Offshore Service Vessel
REST	Representational state transfer
RTO	Research and Technology Organization
SME	Small and Medium-sized Enterprises
SQL	Structured Query Language
SVM	Support Vector Machines
TLS/SSL	Transport Layer Security/Secure Sockets Layer
TRL	Technology Readiness Level
UI	User Interface
WPF	Microsoft's Windows Presentation Foundation Framework
ZEWT	Zero Emission Waterborne Transport

1 Introduction

The maritime industry stands at the cusp of a transformative era, driven by the imperatives of sustainability and technological advancement. The ambition of the NEMOSHIP project is to develop, test and demonstrate new innovative technologies, methodologies and guidelines to accelerate large BESS deployment and optimal exploitation toward 2030 on both hybrid and full-electric arrangement. To help achieve this ambition, the NEMOSHIP project will develop a modular and standardised battery energy storage solution that is able to exploit heterogeneous storage units and a cloud-based digital platform to enable data-driven, optimised, and safe exploitation.

The NEMOSHIP Digital Cloud Platform, is an innovative initiative that redefines the landscape of maritime transport through cutting-edge digital technologies. This architecture report delves into the comprehensive design and operational framework of the platform, underscoring its pivotal role in revolutionizing the management of large battery systems on ships.

The NEMOSHIP Digital Cloud Platform offers a modular and scalable architecture that seamlessly integrates with diverse maritime operational environments. This platform is engineered to enhance the efficiency and reliability of energy storage and utilization, pivotal in steering the shipping industry towards a sustainable future. The intricate design of the platform focuses on maximizing the potential of big data analytics, machine learning algorithms, and real-time data processing to optimize performance and energy consumption.

Central to the platform's architecture is its ability to provide a holistic view of a ship's energy systems, enabling operators to make informed decisions based on precise data and predictive analytics. The platform's robust architecture supports a wide array of functionalities, from monitoring battery health to managing energy distribution and consumption patterns. This ensures optimal performance and longevity of the battery systems, a critical aspect in maintaining operational efficiency and reducing environmental impact.

Furthermore, the NEMOSHIP Digital Cloud Platform is designed with a strong emphasis on security and data integrity. Recognizing the sensitivity of maritime operational data, the platform incorporates advanced security protocols and encryption techniques to safeguard against cyber threats. This security framework is not only a testament to the platform's reliability but also aligns with the industry's growing focus on cybersecurity in the digital age.

In addition, the platform's architecture is crafted to facilitate seamless integration with existing maritime infrastructure and systems. This interoperability is vital for ensuring a smooth transition to more advanced, data-driven operational models within the shipping industry. By enabling easy integration, the platform paves the way for widespread adoption and sets a new standard in maritime digital solutions.

As this report unfolds, it will provide an in-depth analysis of the architectural components, functional capabilities, and strategic implementations of the NEMOSHIP Digital Cloud Platform that will be developed and implemented as part of the collaborative project. We will explore how each element of the platform's architecture contributes to its overall efficacy and how it stands as a cornerstone in the journey towards a more sustainable and efficient maritime future.

2 Architecture Overview

For the NEMOSHIP project, as shown in Table 1, we selected and prioritized a set of important architecture features that will act as a guide for the architecture decisions ahead. An overview of the architecture is provided in Figure 1.

The Architecture feature priorities are different for the prototype version of the software (part of the NEMOSHIP project) compared to the production version (post NEMOSHIP project). Table 1 lists the prototype priorities while Table 2 lists the production priorities.

Table 1 - NEMOSHIP Digital Cloud Platform prototype architecture features

Feature Name	Description	Priority
Modularity	The division of a software system into smaller, independent components that encapsulate specific functionality and can be developed, modified, and replaced independently.	1
Security	The practice of protecting the software from unauthorized access, data breaches, and other malicious activities that could compromise its functionality and data integrity.	2
Complexity	A characteristic that quantifies the level of intricacy in a software system, in terms of its components, relationships, or the algorithms it employs.	3
Fault Tolerance	The capability of a software system to continue operating correctly even when one or more of its components fail.	4
Interoperability	The ability of a software system to function cohesively with other systems, either within the same organization or externally, without special effort for integration.	5
Cost-Efficiency	The optimization of resource utilization—such as time, money, and manpower—in a software system to deliver maximum value with minimum investment.	6
Testability	The ease with which a software system can be tested to validate its functionality, performance, and reliability, usually facilitated by built-in testing frameworks or architectures.	7
Performance	The measure of a software system's efficiency in terms of speed, responsiveness, and resource utilization under specific conditions.	8
Usability	The extent to which a software system can be used effectively, efficiently, and satisfactorily by its intended users.	9
Reliability	The measure of a software system's ability to perform its required functions accurately and consistently over a specified period of time.	10
Scalability	The ability of a software system to gracefully handle increased loads by dynamically adding resources or optimizing existing infrastructure.	11

Table 2 - NEMOSHIP Digital Cloud Platform production architecture features

Feature Name	Description	Priority
Modularity	The division of a software system into smaller, independent components that encapsulate specific functionality and can be developed, modified, and replaced independently.	1
Testability	The ease with which a software system can be tested to validate its functionality, performance, and reliability, usually facilitated by built-in testing frameworks or architectures.	2
Security	The practice of protecting the software from unauthorized access, data breaches, and other malicious activities that could compromise its functionality and data integrity.	3
Fault Tolerance	The capability of a software system to continue operating correctly even when one or more of its components fail.	4
Interoperability	The ability of a software system to function cohesively with other systems, either within the same organization or externally, without special effort for integration.	5
Cost-Efficiency	The optimization of resource utilization—such as time, money, and manpower—in a software system to deliver maximum value with minimum investment.	6
Complexity	A characteristic that quantifies the level of intricacy in a software system, in terms of its components, relationships, or the algorithms it employs.	7
Performance	The measure of a software system's efficiency in terms of speed, responsiveness, and resource utilization under specific conditions.	8
Usability	The extent to which a software system can be used effectively, efficiently, and satisfactorily by its intended users.	9
Reliability	The measure of a software system's ability to perform its required functions accurately and consistently over a specified period of time.	10
Scalability	The ability of a software system to gracefully handle increased loads by dynamically adding resources or optimizing existing infrastructure.	11

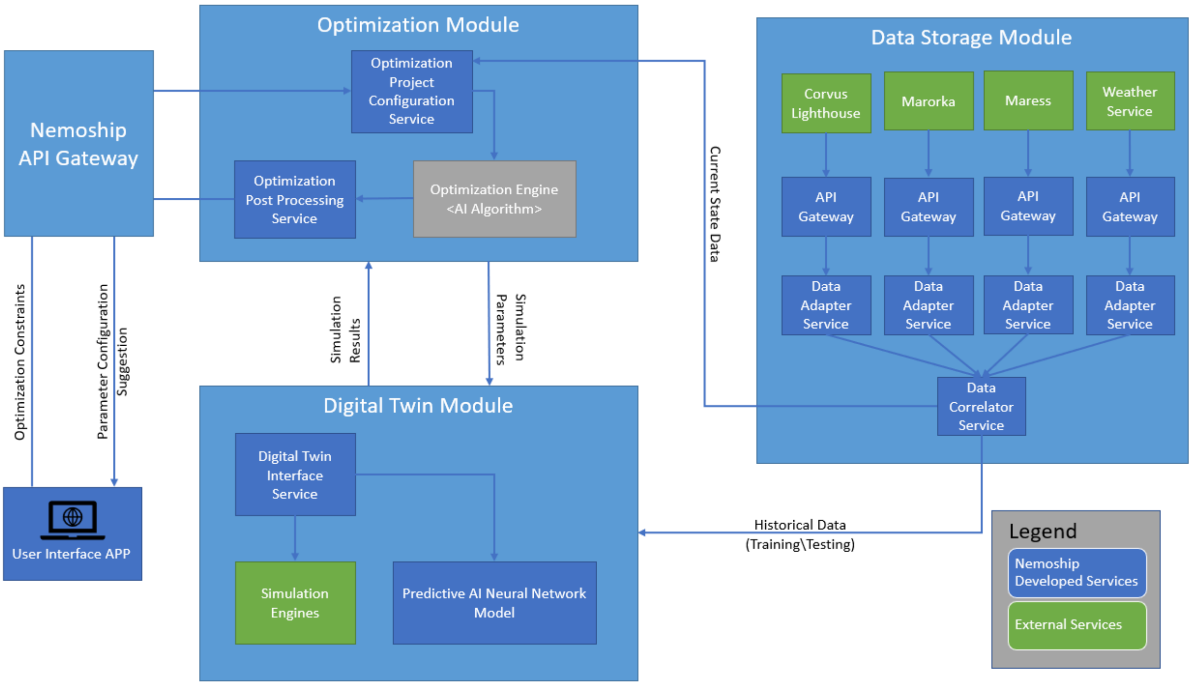

Figure 1 - Architecture overview

In the architectural design of this project, the prioritization of features is crucial for achieving the desired outcomes and ensuring the system's robustness and efficiency. The most critical feature, modularity, forms the cornerstone of our architecture, promoting flexibility and ease of maintenance by encapsulating functionalities into discrete, interchangeable components [1] [2]. This design choice significantly enhances testability, allowing for isolated testing of individual modules, thus improving the overall quality and robustness of the system. Security, a paramount concern, is integrated into each module to safeguard data and operations against unauthorized access and threats, while fault tolerance ensures that the system remains operational even in the event of component failures, thereby preserving service continuity. Interoperability is emphasized to ensure seamless communication and data exchange with external systems and technologies, expanding the system's utility and integration capabilities.

Cost-efficiency is a key consideration, guiding decisions to optimize resource allocation and reduce expenses without compromising on quality. This leads us to complexity management, where the architecture is designed to be as simple as necessary but as complex as required, balancing sophistication with maintainability. Performance is another vital feature, ensuring that the system operates swiftly and efficiently, meeting the demands of users without excessive resource consumption. Usability is prioritized to guarantee that the system is intuitive and user-friendly, enhancing user satisfaction and productivity. Reliability is crucial, as it ensures consistent and correct functioning over time, fostering user trust. Finally, scalability is important to accommodate growing user numbers and data volumes, ensuring the system can expand and adapt to changing requirements without significant redesign or downtime. This prioritization reflects a balanced approach, aiming to create a robust and efficient platform.

2.1 Data Handling Overview

In the NEMOSHIP architecture, data sources play a critical role in enriching the system's capabilities and functionalities. External data providers such as Corvus Lighthouse, Marorka [3], Maress [4], and the Weather Service are integrated to offer specialized data streams that can be leveraged for various analytical and operational purposes. It is important to note that this list of data providers is not exhaustive; the architecture is designed with modularity in mind, allowing for the seamless incorporation of additional data sources during the implementation phase. This modular approach not only enhances the system's adaptability but also ensures that it can evolve to meet changing requirements or integrate emerging technologies and data providers.

When we analyse data, we have the following criteria in mind:

- Granularity: The level of granularity of data can be:
    - Low – Data points are 1 hour or more apart
    - High – Data points are less than 1 hour apart up to multiple data points per second
- Accuracy: The level of accuracy represents how close the data values are to the real values
- Dependency: Describes how correlated the values within a data point are. For example, if a data point contains fuel consumption and carbon emissions but we notice that there is a clear linear dependency between the two (ex: CI=Fuel*X) this means that there is a high degree of dependency which makes part of the data redundant.
- Completeness: Defines how much of the vessel’s operation and state the data describes. A high degree of completeness means that a lot of data is available for a specific moment in time about a high number of the ship’s parameters and state.

For the purposes of this document, we categorize data into two primary types to facilitate better understanding and management within the architectural framework.

The first type is "Historical Data" (Figure 2) which refers to data that is available online and is stored for extended durations, typically ranging from one to two years or even longer. This type of data is invaluable for conducting long-term analyses, trend identifications, and predictive modelling.

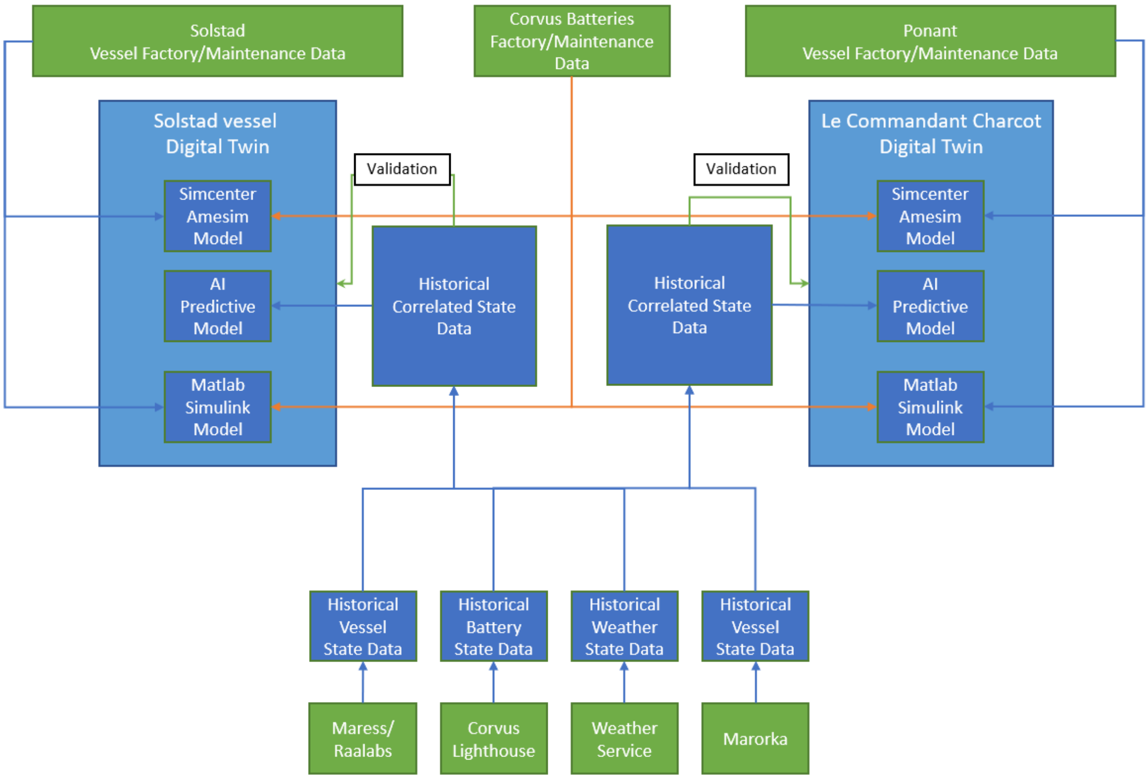

Figure 2 - Historical data usage

In the architecture of the current software project, historical data is obtained from multiple external sources, including but not limited to Corvus Lighthouse, Maress, and Marorka. Each of these providers delivers the data in a unique format and at varying levels of granularity, posing a challenge for consistent data integration. To render this disparate data usable for our specific applications, it must undergo a process of correlation and transformation so as to standardize it into a universally acceptable format across all data providers. Once harmonized, this refined historical data serves dual purposes. Firstly, it is instrumental in validating digital twin simulations constructed using software platforms such as Simcenter Amesim or Matlab Simulink. Secondly, it is employed to train Artificial Intelligence-based predictive models, which are integral components of these digital twins. In addition to the aforementioned data sources, there exists a separate category of historical data, specifically originating from factory and maintenance records. This type of data is generally procured from the original builders of the vessels and their subsequent maintenance logs. The primary utility of this category of data lies in its ability to both establish and continuously improve digital twin models. By doing so, the software aims to achieve high-fidelity simulations that mirror the physical characteristics of the ships with utmost accuracy.

The second category of data, termed "Real-time Data" (Figure 3), is distinguished by its prompt or near-prompt accessibility via online platforms, exhibiting a maximal latency threshold of no more than 15 minutes relative to the present moment. This form of data is pivotal for decision-making processes where time is a critical factor, as it enables the optimization module to make informed decisions based on the most current information available. Furthermore, real-time data plays an indispensable role in the continuous monitoring of ongoing activities or the status of various systems. Its prompt availability ensures that any significant changes or anomalies are quickly detected and addressed, thereby facilitating proactive management and response strategies. This immediacy in data availability is essential in dynamic environments where delayed information could lead to suboptimal guidance or missed opportunities.

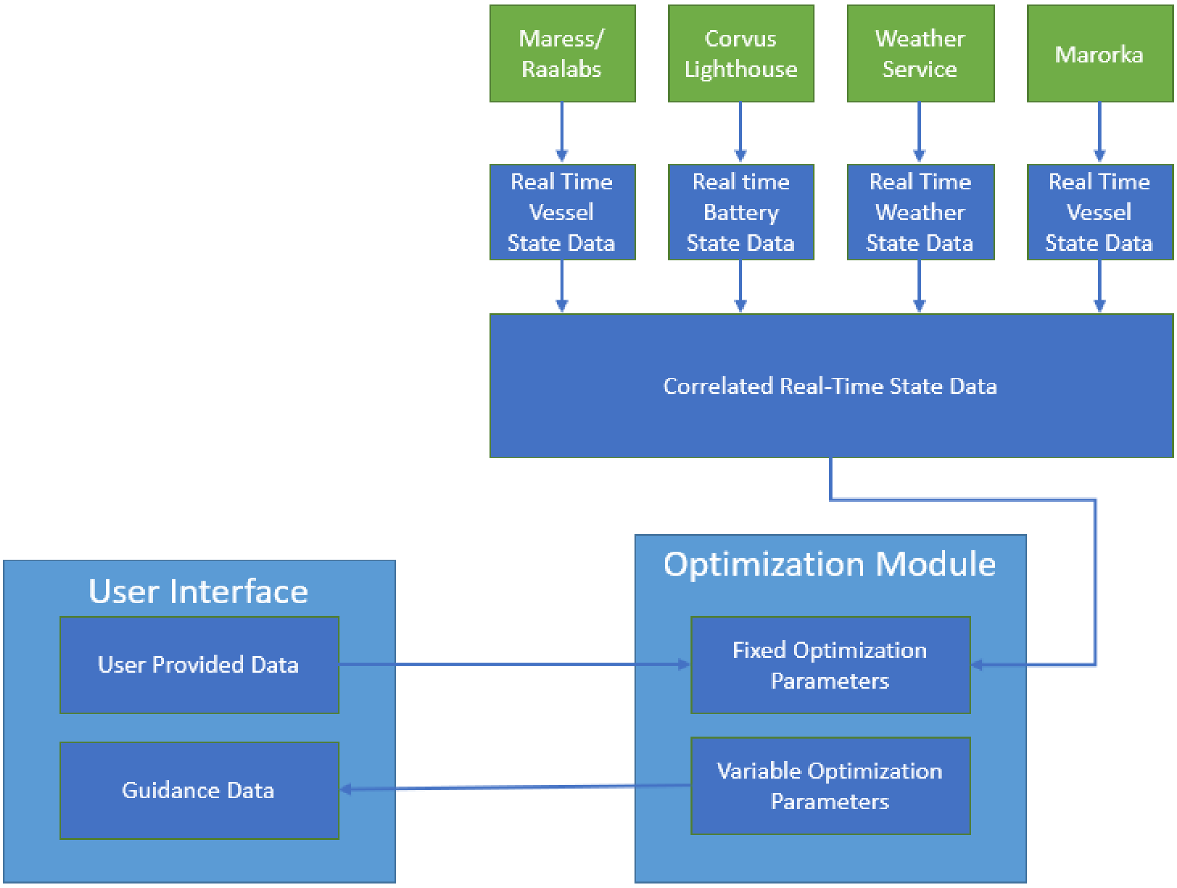

Figure 3 - Real-time data usage

Real-time data is gathered from the external data providers and from the user interface. The data is then correlated so that it can be used by the optimization module. The optimization module uses the real time data to update its fixed parameter values and run the digital twin simulations with them.

By classifying data into these two types, we aim to provide a structured approach to data management and utilization within the system.

Within the framework of the NEMOSHIP system, there are several key data producers that contribute distinct types of data, serving specialized functions:

- The "Optimization Module" within the system architecture holds the critical responsibility of computing and producing a refined set of parameters, each intricately associated with distinct temporal values. This module's functionality is central to the enhancement of operational guidance provided to the crew. By meticulously analyzing and optimizing these parameters, the module significantly contributes to elevating the precision and effectiveness of the instructions and strategies relayed to the crew. This optimization is not merely about fine-tuning individual elements; it involves a comprehensive assessment of the interplay between different variables and their impact over time. The resultant data, characterized by its optimized nature, is instrumental in driving efficiency across various procedural facets. It enables the identification of the most effective approaches and techniques, thereby streamlining vessel operations and fostering an environment of more efficient battery usage. This module, therefore, serves as a pivotal tool in the quest for optimal battery performance, ensuring that every action taken is backed by data-driven insights and is aligned with the overarching objectives of efficiency and quality enhancement.

- The "Digital Twin Module" is a central component of the system architecture, specifically engineered to generate sophisticated simulation data that accurately models the behavior and characteristics of a physical asset, with a primary focus on optimizing battery usage. This module's capabilities are critical in creating a detailed and dynamic virtual representation of battery systems, which mirrors their real-world operation in an intricate manner. The paramount importance of this module lies in its application towards optimizing battery usage. By employing the virtual model, it allows for an in-depth analysis of battery performance under various conditions, facilitating the identification of the most efficient usage patterns and charging cycles.
  This advanced modeling is invaluable for predictive maintenance specifically tailored to battery systems, enabling early detection of potential issues that could affect remaining useful battery life and performance. In the context of scenario planning, the Digital Twin Module offers a robust platform for simulating different operational strategies and their impact on battery longevity and efficiency. This enables the crew to make data-driven decisions on battery management, ultimately leading to enhanced battery lifecycle and reduced operational costs.
  Moreover, the module's contribution to analytical endeavors in battery usage optimization is profound. It provides comprehensive insights into battery performance metrics, degradation patterns, and lifecycle management. This empowers decision-makers with actionable intelligence to implement strategies that maximize battery efficiency and sustainability. In summary, the Digital Twin Module is not just a replication tool but a strategic asset in battery usage optimization, offering a blend of predictive maintenance, efficient operation planning, and enhanced analytical capabilities.
- The "Data Storage Module" in this system’s architecture serves a unique and vital role, distinct from other modules in that it does not generate new data. Instead, its primary function is to aggregate data by correlating, extracting, and organizing existing datasets sourced from a variety of providers. This module acts as a sophisticated nexus for data integration, employing advanced algorithms and data processing techniques to synthesize and harmonize disparate data sources into a cohesive, unified whole. The process involves not just the mere collection of data, but a deeper level of analysis and organization, which includes identifying patterns, relationships, and consistencies across different datasets.
  The capability to correlate and aggregate data from multiple sources is invaluable in providing a comprehensive and holistic view of the system's overall performance and status. It enables the extraction of meaningful insights from a sea of information, transforming raw data into a structured, easily accessible format that is conducive to analysis and decision-making. The Data Storage Module, therefore, plays a crucial role in enhancing the system's analytical capabilities, supporting a wide range of strategic endeavors from trend analysis to predictive modeling.
  Furthermore, the effective organization of data by this module ensures that it is not just stored, but is made readily available for various applications and stakeholders in a format that is most useful to them. This streamlined access to aggregated data enhances the efficiency of the entire system, allowing for quicker retrieval and utilization of information, which is essential in fast-paced decision-making environments. In summary, the Data Storage Module is a cornerstone of the platform's data architecture, pivotal in transforming scattered datasets into a coherent, valuable resource that underpins the platform's decision-making and analytical processes.

2.2 Digital Cloud Platform Testing

In this chapter we address a comprehensive strategy for testing various modules, ensuring that each fulfils its intended functionality and integrates seamlessly into the overall system (Table 3). The User Interface (UI) module, being the primary point of interaction for users, will undergo extensive usability and functional testing to ensure an intuitive and error-free experience. Tests will include user guidance simulations, interface responsiveness, and compatibility across different devices. For the API Gateway, a critical component for managing and routing requests, we will conduct thorough API testing, including testing for request/response handling, security (authentication and authorization), and rate limiting. This will involve both unit testing for individual endpoints and integration testing to ensure seamless interaction with other system components.

The Optimization Module, responsible for processing and generating optimized solutions, will be subjected to a series of performance and functional tests. These tests will validate the module's efficiency in processing inputs and its ability to generate reliable and optimal outputs under various scenarios. The Digital Twin Module, which handles the virtual representations of the physical vessels, will be tested for accuracy and synchronization with real-world data. This will include validation of the data modelling processes, real-time data handling capabilities, and the integrity of the simulation outcomes.

Lastly, the Data Adapter Module, which plays a pivotal role in interfacing with different data sources and formats, will undergo extensive data compatibility and transformation tests. This will ensure that it accurately interprets and converts data into a usable format for other system components. Each module will also be part of an end-to-end system testing regimen to validate their interoperability and collective performance in a simulated production environment. This comprehensive testing approach is designed to guarantee that each module not only functions as intended in isolation but also contributes effectively to the system's overall goals and performance.

Table 3 - Digital platform test coverage by module

Module	Unit Tests	Integration Tests	System Tests	Manual Tests
User Interface	No	No	Yes	Yes
API Gateway	Yes	Yes	Yes	No
Optimization Module	Yes	Yes	Yes	No
Digital Twin Module	Yes	Yes	Yes	Yes (where applicable)
Data Adapter Module	Yes	Yes	Yes	No

2.3 Supported Use-Cases

For all supported use-cases the first step that the users will have to go through is authentication. For this a login page is available like the following mock-up (Figure 4).

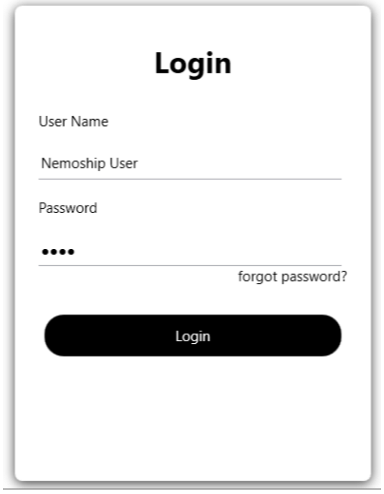

Figure 4 - Login page

Once the user is authenticated, they can access the list of available tools offered by the platform as well as other platform specific configurations. Once the user selects one of the available tools the user interface switches to the respective screens in the use-case that corresponds to the tool (Figure 5).

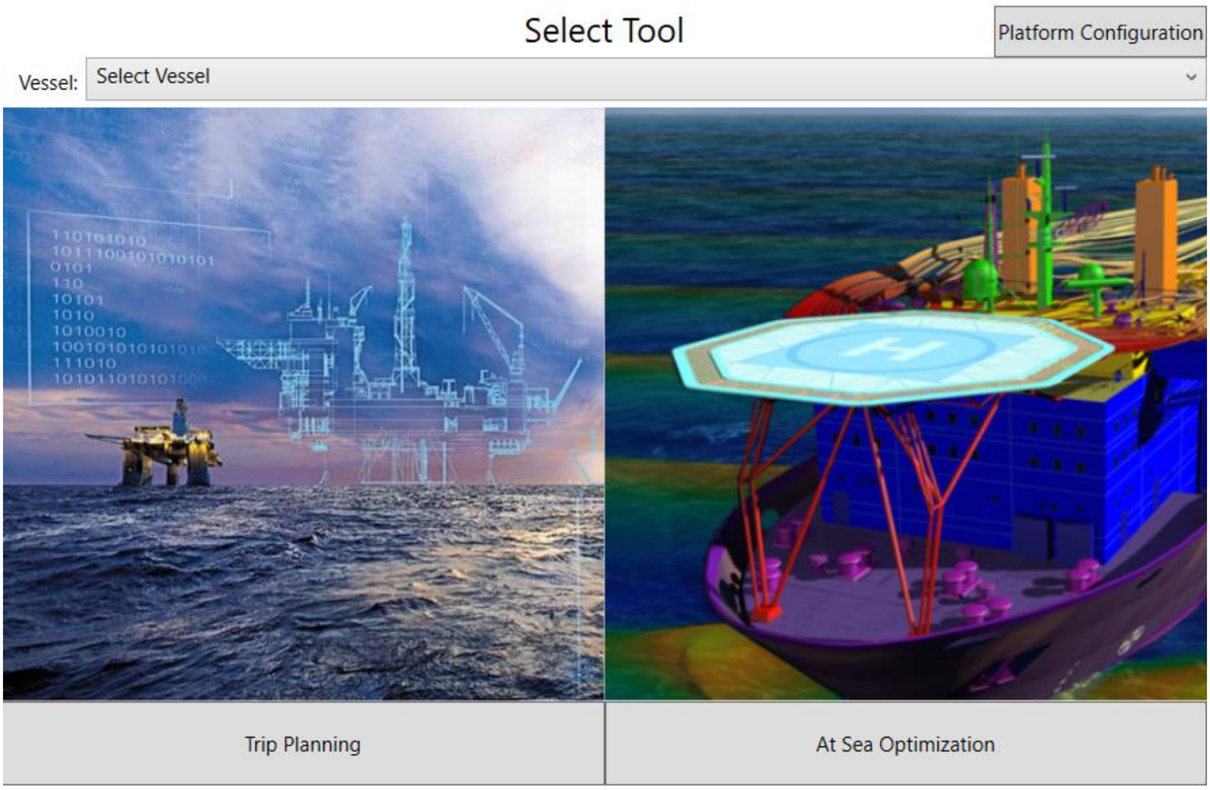

Figure 5 - Tool selection page (design is an example)

2.3.1 Battery Usage Optimization During Trip Planning

This scenario allows the user to import or define all the parameters of a trip in the NEMOSHIP Digital Platform interface and see the recommended guidance values for all the vessel optimization parameters available in the digital twin.

To do this, the user is presented with a trip planning screen as shown in Figure 6.

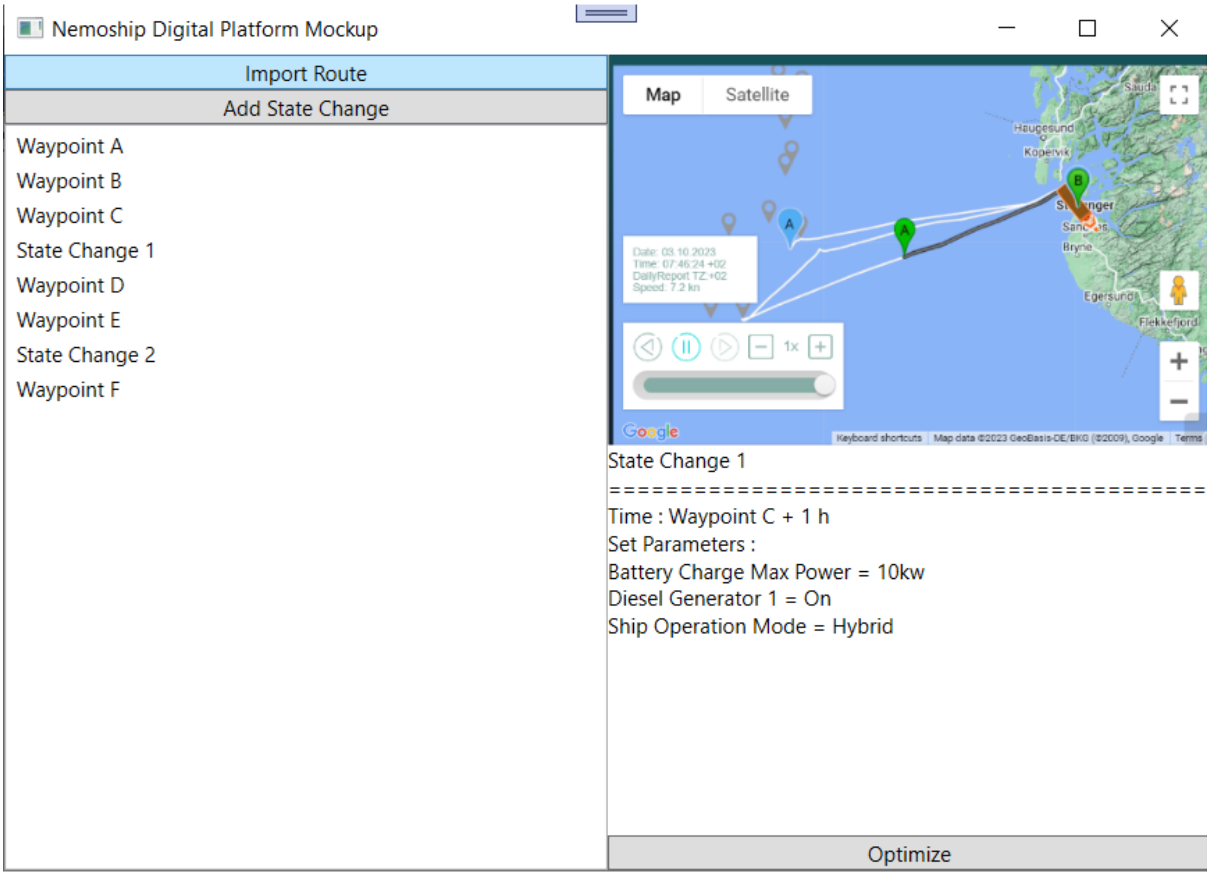

Figure 6 - Trip planning page (example)

The user can import a route, add state changes by specifying a waypoint and optionally a time difference. The purpose of the digital platform is not to replace existing and established maritime trip planning software, instead it will be a complementary addition of features that are needed to optimize battery usage. Therefore, the platform will support importing waypoints and other data from the existing maritime platforms and services.

When the user then selects a state change and hits "Optimize" button and the following page is displayed (Figure 7).

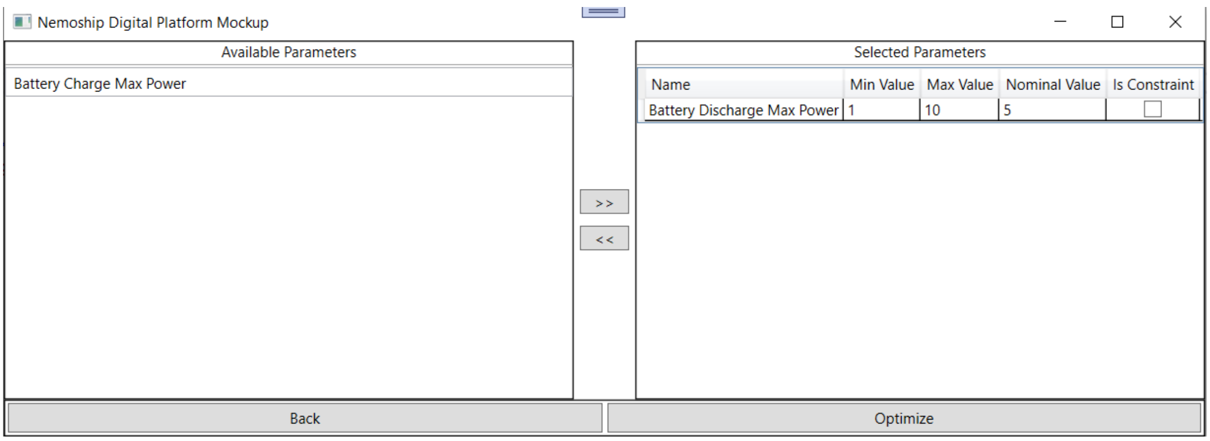

Figure 7 - Parameter selection page

Here the user can select which parameters they are interested in optimizing by moving them from the "Available Parameters" tab to the "Selected Parameters" tab. The user can also specify the parameters that need to be fixed to specific values by checking the “Is Constraint” checkbox in the “Selected Parameters” tab and specifying the desired fixed values.

Once the user is content with the parameter selection, they can start the optimization process by clicking the “Optimize” button. Once the optimization is completed the user will be taken back to the Trip Planning page where the guidance will now be displayed for the respective State Change items.

2.3.2 At Sea Battery Usage Optimization

In this use-case, the system's adaptability and dynamic functionality are exemplified through its application in both trip planning and at-sea operations. Initially, during the trip planning phase, users interact with a set of pages designed to assist in mapping out the voyage. These pages provide vital information, including proposed waypoints, anticipated state changes, and other relevant navigational data.

However, once in this scenario the vessel is at sea, the same pages transform to reflect real-time conditions and requirements. This dynamic shift is critical for effective battery efficiency management. The vessel's position is updated continuously, ensuring accurate and current location data is always available. In this mode, the system smartly filters and displays only the relevant waypoints and state changes pertinent to the current leg of the journey. This selective visualization ensures that the crew is not overwhelmed with unnecessary information, enabling them to focus on immediate priorities.

Behind the scenes, a more complex and adaptive optimization process takes place. Unlike the pre-voyage planning phase, where the optimization is based on forecasts and static data, the at-sea optimization harnesses current data from an array of providers such as Marorka, Maress, and the Weather Service. This real-time data is fed into the digital twin module, which plays a crucial role in this phase. The digital twin, with its capability to simulate the vessel's behaviour and interactions with various environmental factors, utilizes the incoming data to generate more accurate and timely predictions. These predictions are essential for adjusting the vessel parameters in response to changing conditions at sea, such as weather changes, sea currents, or unforeseen operational constraints.

The integration of current data from diverse providers into the digital twin ensures that the optimization is based on the most up-to-date and comprehensive information available. This not only enhances the accuracy of the predictions but also allows for more effective decision-making. Crew members can leverage these insights to adjust the mode of operation, speed, and other operational parameters in real-time, optimizing the vessel's performance and efficiency while ensuring safety and adherence to the planned schedule.

In conclusion, these use-cases demonstrate the platform's versatility and advanced capabilities in adapting to different phases of a voyage. From detailed pre-voyage planning to real-time adjustments at sea, the system provides an integrated, data-driven solution that enhances battery efficiency, safety, and decision-making accuracy.

2.3.3 At Sea Scenario Prediction

In the realm of maritime operations, the implementation of digital twin technology marks a significant advancement, particularly in the context of this use case. The digital twin, a virtual replica of the vessel, becomes an invaluable tool during the vessel's deployment at sea, enabling a multitude of simulated scenarios that offer deep insights into vessel behavior under varying conditions. This technology not only simulates real-time operations but also extends to hypothesize numerous 'what-if' situations, thereby providing a robust platform for strategic decision-making and operational optimization.

At the core of this application is the ability to simulate the vessel's behavior using a range of input parameters and scenarios. This simulation is not a mere theoretical exercise; it is a sophisticated analysis that incorporates varying environmental conditions, operational modes, and functional parameters. By manipulating these variables within the digital twin framework, the system can forecast the vessel's performance, considering factors such as fuel efficiency, speed, maneuverability, and safety.

One of the most notable features of this technology is its comparative analysis capability. The system can run parallel simulations, each varying in certain parameters, and then juxtapose these scenarios against each other. This comparison is crucial for understanding how specific changes in operational modes or functional parameters can significantly impact key performance indicators (KPIs). For instance, by adjusting variables like engine speed, navigation route, or weather conditions, the system can demonstrate the resultant effects on fuel consumption and carbon emissions. Such analysis is invaluable for enhancing operational efficiency and environmental sustainability.

Furthermore, the technology's retrospective analysis capability offers another layer of functionality. By inputting actual historical data into the digital twin, the system can retrospectively simulate different scenarios that the vessel might have encountered. This retrospective simulation allows for a comprehensive analysis of past decisions. It answers critical questions like, 'What would have been the impact on fuel consumption if a different route was taken?' or 'How would a change in speed have affected carbon emissions during a particular leg of the journey?' This hindsight perspective is crucial for learning from past voyages, informing future operational strategies, and continually refining the vessel's performance.

In conclusion, the application of digital twin technology in this use-case transcends conventional simulation. It offers a multi-faceted platform that not only predicts and optimizes vessel behavior in real-time but also provides a retrospective analysis to learn from past voyages. This dual capability of forward-looking simulation and backward-looking analysis positions the digital twin as a critical tool in modern maritime operations, driving efficiency, sustainability, and strategic decision-making in the face of ever-changing sea conditions and operational demands.

3 Main Modules and Communication

The NEMOSHIP Digital Cloud platform, designed as modular platform, embodies the principles of distributed computing and microservices architecture. Each module within this platform is designed to fulfil a distinct and specialized role, contributing a unique piece to the collective functionality of the system (Figure 8). This modular design not only enhances the system's flexibility and scalability but also facilitates maintenance and upgrades by allowing individual modules to be modified or replaced independently.

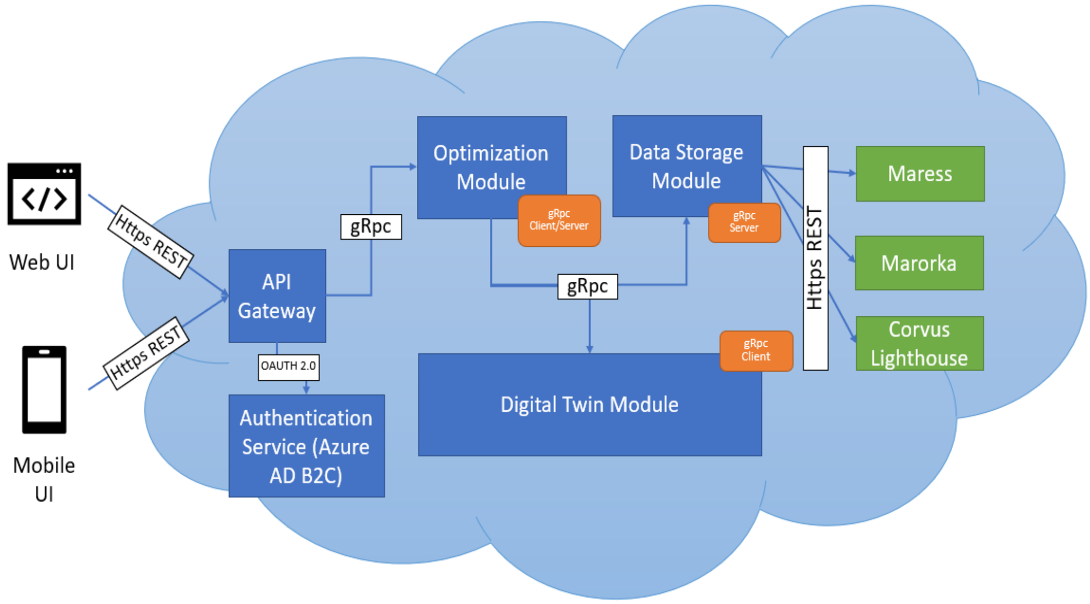

Figure 8 - Module communication

In such a modular environment, effective communication is paramount, both between the modules themselves and with external entities like users and data providers. Since the entire platform is cloud-deployed, network-based communication protocols are employed to facilitate this interaction. The selection of these protocols is critical to the efficiency, security and reliability of the system's communication framework.

The platform predominantly utilizes two main communication protocols: REST (Representational State Transfer) and gRPC [5] (gRPC Remote Procedure Calls). REST, known for its simplicity and widespread adoption, is primarily used for communication with external services. This includes interactions with the User Interfaces and various external data providers. The choice of REST for these external communications is due to its stateless nature and its ability to seamlessly integrate with the web, where many of these external services operate. REST's use of standard HTTP methods (GET, POST, PUT, DELETE) makes it intuitively understandable and easy to implement, ensuring broad compatibility with a wide range of services (as illustrated in Figure 8 of the architecture document).

On the other hand, gRPC is chosen for internal communications within the NEMOSHIP Digital Cloud platform, particularly between its various modules. Developed by Google, gRPC is a high-performance, open-source framework that supports bidirectional streaming and multiplexing over a single connection. It is particularly suited for scenarios where the handling of potentially large amounts of data is expected, a common occurrence within the platform's internal operations. gRPC's use of Protocol Buffers as its interface description language results in smaller payloads and faster serialization and deserialization, offering a significant performance advantage over traditional RESTful services. The server application used to host the modules will be based on the Microsoft Kestrel server and thus it will utilize TLS/SSL prototcols for secure transfert through gRPC. This makes gRPC an ideal choice for the platform's internal module-to-module communications, where efficiency and speed are critical.

In conclusion, the NEMOSHIP Digital Cloud platform's communication strategy is thoughtfully architected to leverage the strengths of both REST and gRPC protocols. REST's simplicity and compatibility are harnessed for external communications, while gRPC's performance efficiency is utilized for the more data-intensive and high-frequency internal communications. This dual-protocol approach ensures that the platform maintains robust, efficient, and scalable communication capabilities, essential for its overall functionality and effectiveness.

3.1 Digital Twin Module

The Digital twin module's role is to simulate to a high degree of accuracy of the real-world ships it is designed to digitally replicate. To do this, it needs to rely on state-of-the-art simulation software like Simcenter Amesim. The Digital Twin module (Figure 9) has the components described in the following subsections.

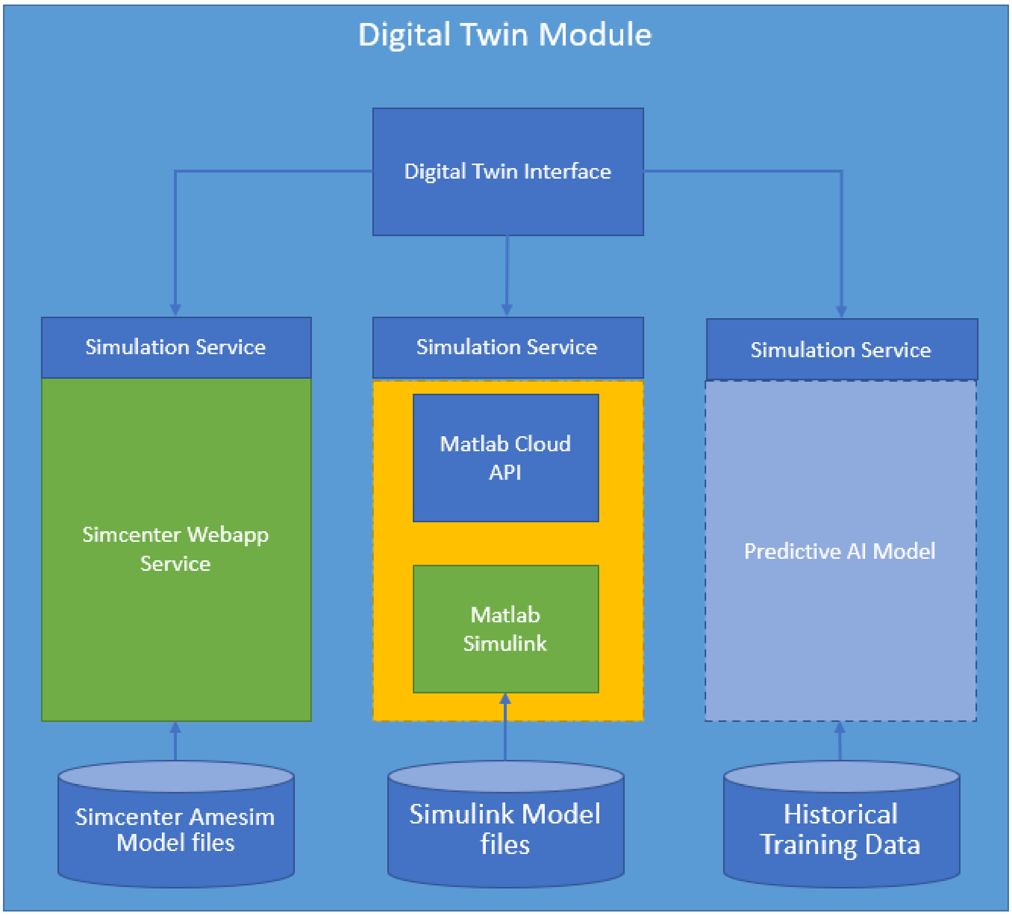

Figure 9 - Digital twin module

3.1.1 Digital Twin Interface

The digital twin interface is a webAPI [6] cloud-based service that provides a point of entry to the other module's components. It acts as a gatekeeper coordinating access and sending back responses in a centralized common format to its clients. It also handles any synchronization needed between the simulation services as well as deciding which simulation service to choose based on a particular parameter input.

3.1.2 Simulation Services

Each Simulation Service is a deployment of a simulation platform like Simcenter Amesim [7] or Matlab Simulink in the cloud with an exposed API that can be accessed by the Digital Twin Interface. Through this API the digital twin interface needs to be able to change simulation parameters, start/stop simulations and extract simulation results when the simulation has finished.

The idea of a Maritime Digital Twin Architecture (MDTA) is offered to organize practical Digital Twin (DT) features in order to increase the general knowledge of DTs and their application. A typical DT is a replica of a physical system that is connected to and receives updates from its real-world counterpart via a bi-directional data link. This is in line with the definition given in a broader sense: DT is sometimes referred to as a virtual representation of a physical system and its surroundings that is updated via information transfer between the two instances.

In this section, MDTA performed by Siemens Amesim and MATLAB/Simulink is described in general. DT Amesim prototype model is composed of two main simulation platform, Simcenter Amesim, and MATLAB Simulink in which firstly a comprehensive model for hybrid electrical propulsion system (EPS) is deployed in Simcenter Amesim, and secondly an optimization process is provided in MATLAB/Simulink to manage required cost functions. The general schematic of EPS model is shown in Figure 10.

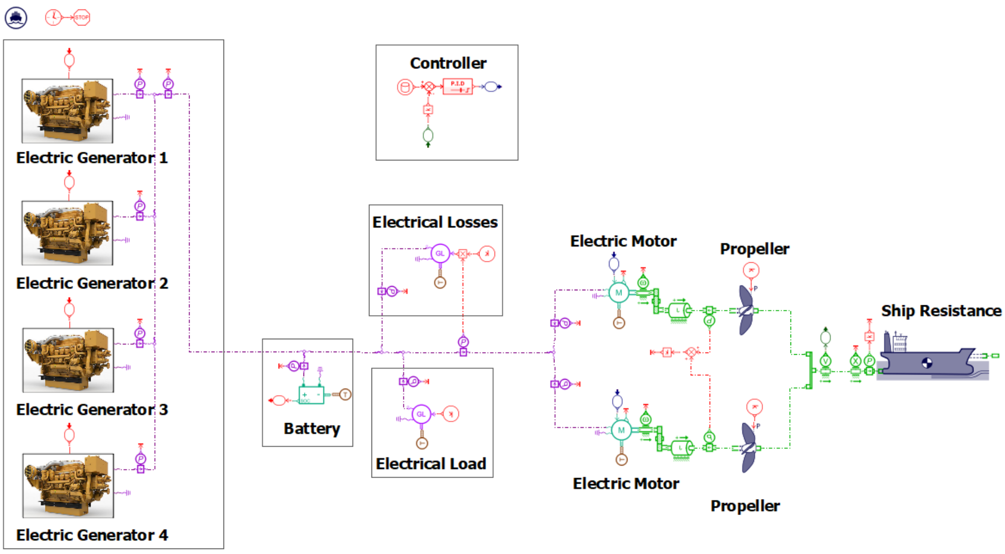

Figure 10 - Schematic of the electric propulsion system (EPS) for a vessel

An interface is also required in order to perform co-design and co-simulation, in which optimization in MATLAB/Simulink will be possible on the model that is created in Simcenter Amesim. Few possible interface models are available for this digital twin co-design. Figure 11 illustrates different interfaces provided by Simcenter Amesim.

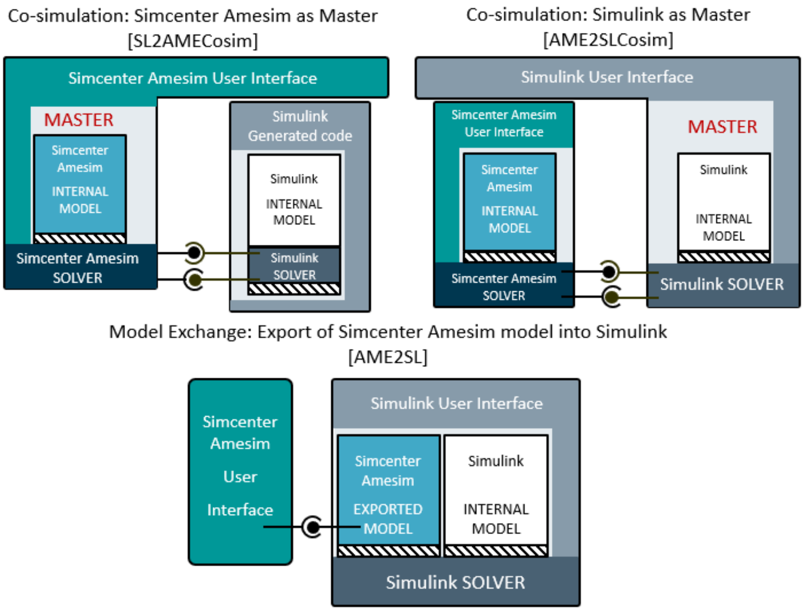

Figure 11 - Interface strategies provided by Siemens Amesim

Several strategies could be used for the optimization process. As an example, the Sl2Amecosim interface can be chosen, in which the model created in Amesim is designed as the master and the optimizer model in MATLAB works as slave (Figure 12).

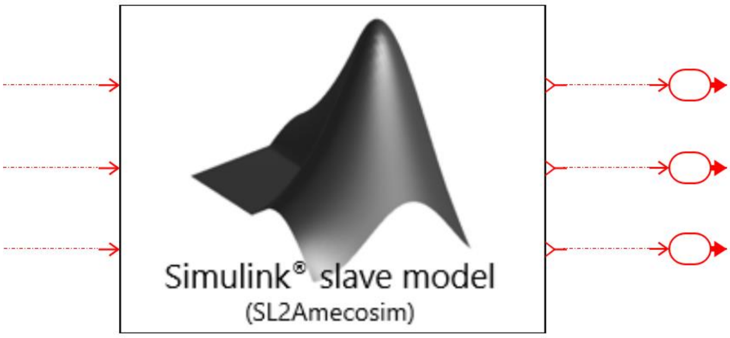

Figure 12 - Amesim interface for co-simulation

Finally, the EPS model is upgraded to an optimized model using the real-world data, and the optimization algorithm performed by MATLAB/Simulink in a co-simulation process. An overview of the simulation performed in Simcenter Amesim is showed in Figure 13.

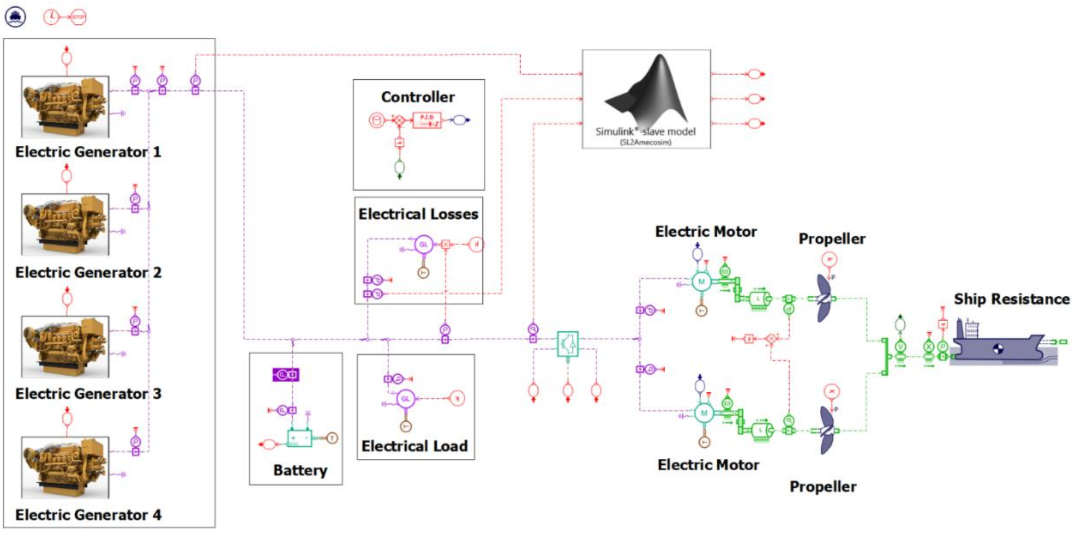

Figure 13 - Siemens Amesim model for vessel EPS in co-simulation with MATLAB

Next steps will be to upgrade model and update the realistic parameters of the simulation model. For example, the real microgrid of the hybrid vessel is described as follows. Vessel microgrid consists of two identical groups, where each group includes 11kV AC main bus 2 batteries, 3 generators and one motor. Two energy sources, i.e., batteries and generators, are connected to the main 11kV AC bus. Generators could be directly connected to the AC bus. The batteries, on the other hand, are connected through power transformers and power converters. Batteries are grouped by two, where each battery has its own DC/AC power converter. The power transformer has two windings on the side of batteries and one at the side of the main AC bus. The vessel motors are connected through power transformer, Active Front End (AFE), DC bus and DC/AC power converter. Each motor is connected to both AC buses, and each connection includes a power transformer, two AFE, and a DC bus, where two DC/AC converters connect to control the motor.

3.2 Optimization Module

Figure 14 shows an overview of the optimization module.

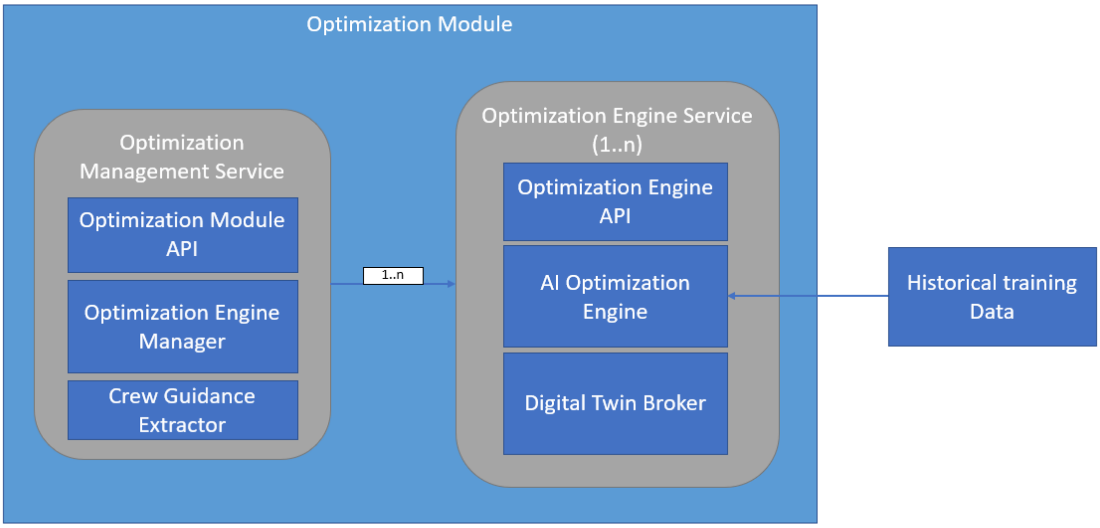

Figure 14 - Optimization module

The Optimization Module is composed of two main services:

- The Optimization Management Service:

Handles all the tasks necessary to get the crew guidance data out of the user input data using the Optimization Engine Service. It has 3 main components to help with this: The Optimization Module API handles receiving all the inputs from the user interface and transfers it to the Optimization Engine Manager. The Optimization Engine Manager decides if it needs to start up a new instance of an Optimization Engine Service or reuse an existing one and process and pass the correct parameters to it. It also handles receiving the output from the Optimization Engine services and passing it to the Crew Guidance Extractor. The Crew Guidance Extractor interprets the results from the optimization Engine Service and converts it to a format that can be used by the User Interface to display the guidance to the user.

- The Optimization Engine Service:

It is responsible for receiving the optimization parameters and finding optimized values for the variable parameters while keeping the fixed parameters constant. It does this with the help of its 3 components: The Optimization Engine API is a sub-component that can communicate with the Optimization Engine Manager and receive the parameters for the optimization run. The AI Optimization Engine is the sub-component that is trained based on historical data to be able to find a set of optimal values based on the fixed parameter values and digital twin simulation run results. It will do this in multiple tries until it reaches a satisfactory result. The Digital Twin Broker is responsible for handling a digital twin run by sending the simulation run parameters that the AI Optimization Engine has found and retrieving the results from the digital twin simulation passing them back to the AI Optimization Engine.

3.3 Data Storage Module

Figure 15 shows the data storage module.

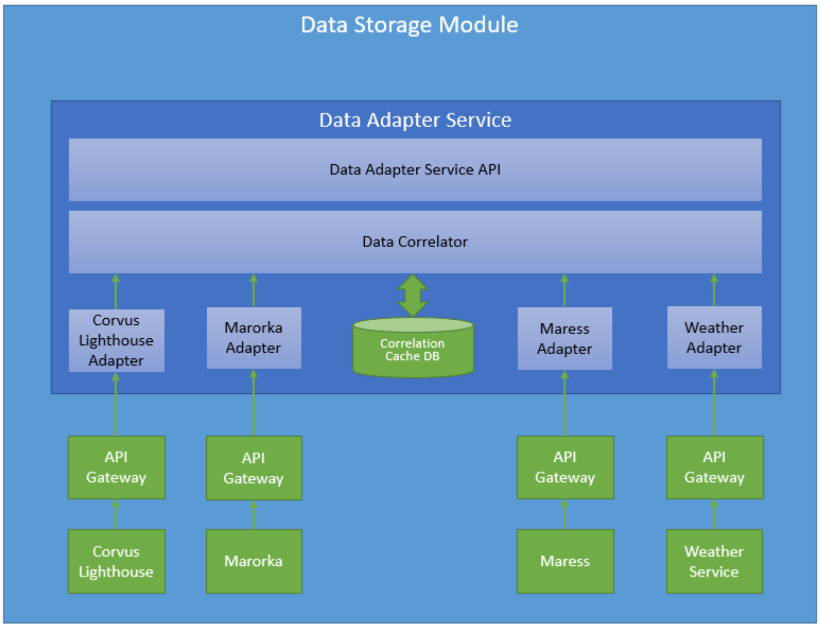

Figure 15 - Data storage module

The Data Storage Module has one main component: The Data Adapter Service. This service exposes a gRPC interface (Data Adapter Service API) that allows it to communicate to the other modules. It contains subcomponent adapters for each data provider in the project with the possibility to always add more. The Data Adapter subcomponents are responsible for all the communication, retrieval and formatting of the data for their respective provider. The Data is then passed to the Data Correlator in a common format which will normalize the data from a duration perspective and correlate the data from all the providers. Once correlated the data is passed back to the interface API which sends it back to the client.

3.4 External Components

External components are the pieces of the NEMOSHIP digital platform that are already existing either as part of the partners' portfolio or software that is being widely used by the maritime industry. We will only go into detail on a few of them that are representative for their category.

3.4.1 Marorka

The Ascenz Marorka Web API is a sophisticated interface designed to facilitate access to data stored within Ascenz Marorka's extensive data center. This API is a critical component for external systems and software tools that require interaction with Ascenz Marorka's data repositories. It provides a standardized, well-documented set of methods, allowing for seamless and secure data retrieval and manipulation.

A key feature of the Ascenz Marorka Web API is its utilization of the OData (Open Data Protocol) for data access. OData is an ISO/IEC approved, OASIS standard that defines a set of best practices for building and consuming RESTful APIs. It is specifically designed to facilitate data sharing across disparate systems and platforms, making it an ideal choice for the Ascenz Marorka Web API.

One of the primary strengths of using OData in the Ascenz Marorka Web API is its ability to perform tasks through queries that are constructed in a manner akin to SQL (Structured Query Language), the standard language used for managing and manipulating data in relational databases. This similarity to SQL provides a familiar framework for developers and data analysts, allowing for the crafting of rich and complex queries with ease. OData's query conventions enable clients to tailor their data requests, specifying exactly what data should be returned, how it should be filtered, and the order in which it should be presented. This capability significantly reduces the amount of data transmitted over the network, as only the specifically requested data is retrieved and sent.

Furthermore, OData supports a range of functionalities such as filtering, sorting, pagination, and aggregations, which are essential for efficient data handling and analysis. It also provides metadata that describes the data model, offering insights into the structure and relationships within the data, thus enhancing the usability of the API.

In addition to these features, the Ascenz Marorka Web API's adoption of OData ensures compatibility with a wide range of client-side data access frameworks and libraries, simplifying integration and development efforts. This makes the Ascenz Marorka Web API a powerful tool for developers and analysts who require direct, flexible, and efficient access to the data stored in Ascenz Marorka's data center, supporting a broad spectrum of data-driven applications and services.

3.4.2 Corvus Lighthouse

The Lighthouse is a data logger that collects data from the Corvus Energy BESS interpack communications network, aggregates it in an efficient format for transfer and then sends the data up to the cloud-based storage for storage and later processing. The main interfaces to the Lighthouse are:

- Ethernet connection via RJ45 port – Inter-pack communications network
- Ethernet connection via RJ45 port – Internet connection (onboard the vessel)
- DC power connection
- Micro SD card slot for additional storage

The Lighthouse plays a non-critical role in the operation of the Orca ESS and the ESS can operate without a Lighthouse connected to the system because it only passively monitors ethernet data on the interpack network. Lighthouse is delivered with all deliveries of Corvus Energy BESS systems.

The Orca BESS packs form an internal inter-pack communications network that uses multi-cast messages to send data to all other packs. The pack has been network storm tested and shown to be resistant. This network is isolated from all other networks and its members are all to be Corvus equipment. The Lighthouse is one member and can listen to the multicast endpoints to record data from all the packs and store it locally. When enough data is collected, it is uploaded to the cloud service and stored.

The Internet port should be connected to the Internet behind a customer firewall. All traffic is initiated from the Lighthouse side and it uses a TLS connection to cloud services to encrypt data transmitted over the connection. The internal firewall in the Lighthouse is configured to ignore incoming connections through the Internet port and the customer is expected to only allow outgoing connections (i.e. connections initiated by the Lighthouse) via a selected set of ports.

Note that although the Lighthouse hardware supports Wi-Fi and Bluetooth protocols, neither is enabled by the software and the antenna are not installed (a cap is placed over the connector).

The Lighthouse collects data each second and there is approx. 90 data points collected (e.g. cell state of charge, voltage, temperature, pack voltage, current, etc.). The data is further processed by Corvus Energy and used in internal data dashboards for analysis, troubleshooting and data monitoring. Through a monitoring agreement with customer Corvus Energy will monitor the data and provide proactive approach, expert based annual report on usage with guidance on how to operate within recommended limits in order to achieve design lifetime. Further on the customer will get access to a customer portal named VIP, Vessel Information Portal, where selected key data is displayed in easily understandable graphs for battery cooling fan inlet temperature, module temperature, state of charge, state of health and full-cycle equivalents. The portal further gives the end user the possibility to easily choose which time view of historical data to display (e.g. last month, year or selected timeframe from date to date) as well as choosing arrays and packs. Technical documentation, user manuals and service reports for the system are available in the document center in the portal. Features such as fleet view and battery pack heat maps are recently added features valuable for the end user.

The data collected from the Lighthouse can also be made available as an API as described in this document on how it is being used.

3.4.3 Webapp Cloud Platform (powered by Simcenter Amesim)

The Simcenter Webapp Server is a versatile and advanced tool designed to democratize the access to system simulation models across an organization.

- Ease of Use and Accessibility: The server is highly user-friendly and accessible, making system simulation available to various users within an organization, including project engineers and technical salespeople. This broad accessibility allows for the provision of predictive information to customers, even by those who do not have desktop access to Simcenter Amesim.
- Customization and Enhanced User Experience: The 2019.2 update of Simcenter Webapp Server focused on increasing customization capabilities. Model owners can now customize the results view for model consumers, streamlining the process of result interpretation. This customization includes the support for standard Simcenter Amesim variables and post-processing variables, allowing for a more tailored results layout.
- API Availability for Integration: The server's application programming interface (API) is available for reuse in third-party applications, offering a great deal of flexibility. Users can benefit from Simcenter Webapp Server's services, such as centralized model storage and access right management, to build specific front ends that are customized for their end-users.
- Workflow Simplification and User Library Support: The 2019.1 update introduced workflow simplification, where model authors can embed pictures of the model directly within Simcenter Amesim, simplifying the upload process. Additionally, user library support was added, allowing for the reuse of Simcenter Amesim user libraries created with Submodel Editor, facilitating the incorporation of both standard and non-standard libraries into models.
- Post-processing and Export Capabilities: Post-processing variables can be reused, enabling model authors to define meaningful variables for model consumers. Moreover, model consumers can export the results as an Excel file for further detailed post-processing, enhancing data analysis and reporting capabilities.
- Security and Intellectual Property Protection: The server operates in a secure, password-protected environment, enabling model authors to control access using parameters and variables. This functionality ensures that intellectual property is safeguarded while still being integrated into a company's infrastructure. Additionally, it allows the visualization of simulations on any device, providing a personalized display for each end-user.
- Sharing and Collaboration: Users can upload and share system simulation models within the server, eliminating the need to send models via email. Shared models are visible in a user's workspace, where new analyses can be started. Each analysis's parameterization and results are stored on the server for later reuse, facilitating collaborative efforts and knowledge sharing.

Overall, Simcenter Webapp Server is an innovative solution that offers simplicity, cost-effectiveness, and a range of features that enhance the accessibility, customization, and security of system simulations in a web-based environment. These capabilities make it a valuable tool for organizations looking to leverage system simulation models across different departments and user types.

4 Prototypes

We want to make sure the proposed architecture is feasible and to reduce the risks associated with unforeseen circumstances during the implementation phase. For this we have chosen a set of prototypes that target the core aspects of the NEMOSHIP Digital platform software components and modules. Through the implementation of these prototypes, we can see various aspects and issues that we will need to deal with during implementation and we plan to address them in a timely manner. These prototypes can also help us validate our architectural choices or point out potential conflicts or deficiencies in the plan.

4.1 Data Adapter

The purpose of the prototype is to define a data adapter module that can access and export data from external Lighthouse and Marorka platforms (Figure 16 and Figure 17). This will show the capabilities of the technologies used as well as the needed dependencies. It will also potentially show the issues that can come into play when developing such a module.

Both the Lighthouse prototype and the Marorka prototype were written in .Net C# WPF framework and show a desktop user interface. The user interface is able to interact with the Lighthouse and Marorka data services and allows the user to go through the specific processes to select the data for export. These processes are specific to each data provider service, so they need to be implemented in each data adapter module.

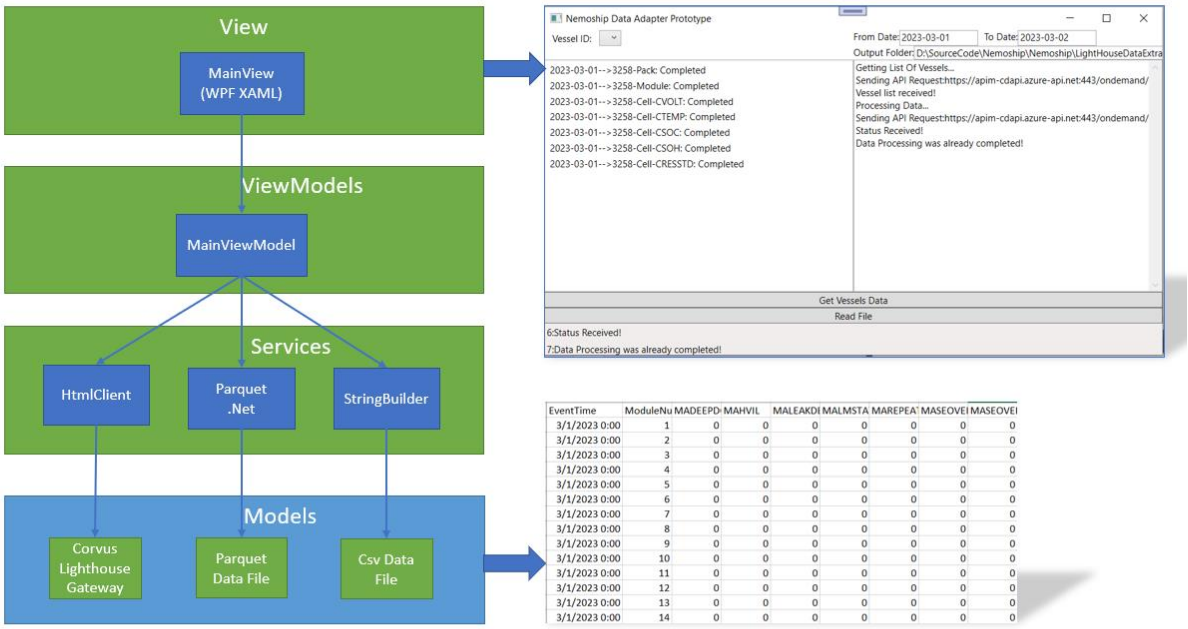

Figure 16 - Lighthouse data adapter

For the Lighthouse service the process is as follows:

1. A request is made to the service to retrieve the accessible vessels. This is done by calling the GetSubscriptionInfo api.
2. The user selects a vessel to access its data.
3. A request is then sent to the service to process a chunk of data that is max 65 days in length. This is done by calling the StartProcessing api with the corresponding parameters.
4. A query is sent to the service to check the processing status periodically until the response returned is Completed or Failed. This is done using the GetProcessingStatus api.

5. Once the processing is completed successfully, a selection is needed to download data for the battery pack or for a specific cell. The GetData api is used for this and provides the data for a single day in the form of a parquet file.

Since the service limits the amount of data that the user can download in a single step, the prototype needs to execute some steps multiple times to download the entire range of data selected by the user.

The UI of the prototype allows the user to perform the following steps:

1. Select a vessel from the list of vessels available on the platform.
2. Select a date range for which the data will be downloaded and exported.
3. Select a csv file name where the data will be exported.
4. Start the export and wait for the completed message to access the file.

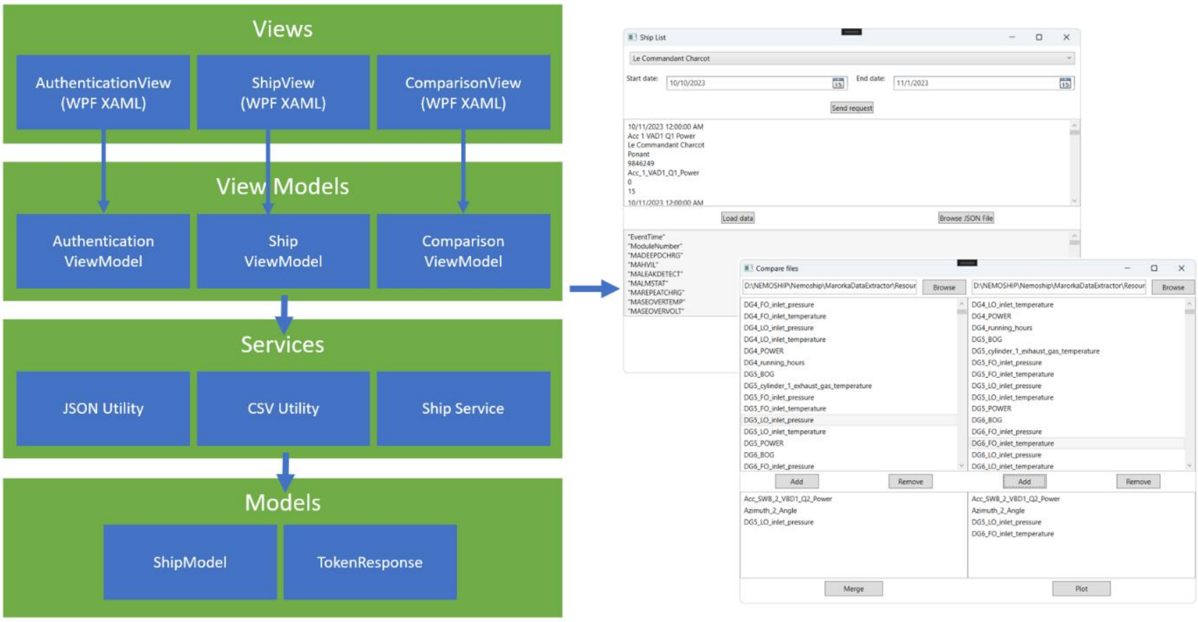

Figure 17 – Marorka Data Adapter

For the Marorka service the process is as follows:

1. A request is made to the service to retrieve the access token.
2. The user selects a vessel and a time interval to access its data.
3. A request is then sent to the service with the parameters set by the user and together with the access token.
4. After processing is successfully completed, a selection is required to load the stored data into a JSON file.
5. The user can choose any number of columns from any two files.
6. The selected columns will be merged into a new file.

The UI of the prototype needs to allow the user to perform the following steps:

1. Select a vessel from the list of vessels available on the platform.
2. Choose a date range for downloading and exporting data.
3. Upload a file (from either Marorka or LightHouse) for visualization.

4. Compare two files.
5. Combine two files by selecting the desired columns.

A Python script was developed to merge information from the Lighthouse service with data sourced from the Marorka service. This script executes a data integration operation, bringing together data from distinct origins: a CSV file (derived from Corvus Lighthouse) and a JSON file (obtained from Marorka). The procedure includes transforming the JSON data into CSV format, manipulating date-time values, and merging the datasets using a shared timestamp (Figure 18).

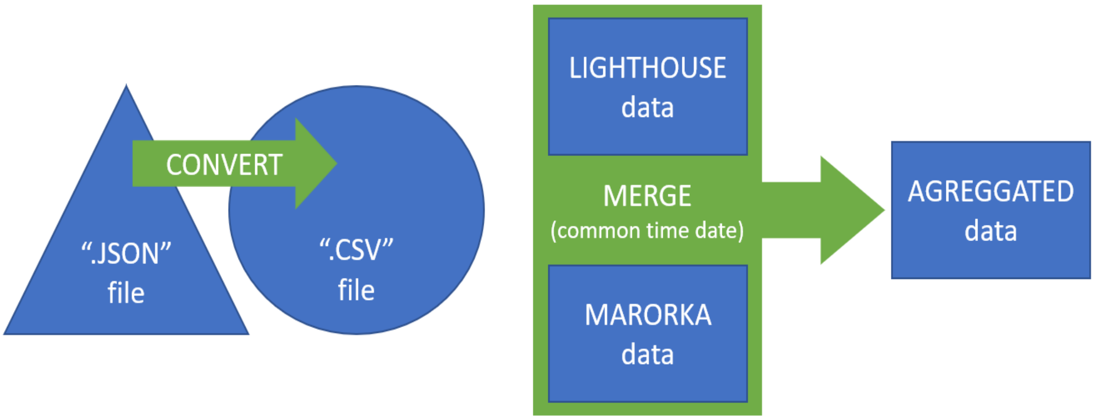

Figure 18 - Combining Lighthouse and Marorka data

Conclusions:

The .Net technology stack proved simple and sufficient to handle the communication protocols required by the Corvus Lighthouse and Marorka services.

The amount of data that we need to extract for historical reference takes considerable time to process.

We need to limit the amount of data we request from the external services/platforms for the real-time data access scenario in order to get acceptable processing performance.

We need to consider the authentication token and the validity period of the token.

4.2 User Interface Prototype

The purpose of this prototype is to identify the best technology stack to build a graphics-rich data visualization user interface similar to other marine platforms like Marorka or Maress. To achieve this a set of minimal visualization elements was decided:

- Gauge / Speedometer indicators (Figure 19)
- Time series visualization (line graphs)
- Map trajectory
- Histogram for distributions
- Scatter plots for value correlations
- Bar chart / pie chart for grouping (by usage/consumer, etc.)

- Spider (radar) diagrams for multi KPI comparison (yearly avg vs current, etc.)

Because of the high amount of specialized UI controls needed and to the high availability of free open-source controls compared to the desktop-based frameworks, we opted for a web-based approach. The chosen technologies were React typescript & electron. Using react and electron together allows to have a single codebase that can be deployed both as a web and a desktop application.

We identified and tried-out a set of open-source control libraries:

- d3 / Docs: https://d3js.org/
- Chart.js / Docs: https://www.chartjs.org/
- Plotly JavaScript / Docs: https://plotly.com/javascript/
- OpenLayers / Docs: https://openlayers.org/
- ECharts (used by Maress) / Docs: https://echarts.apache.org/en/index.html

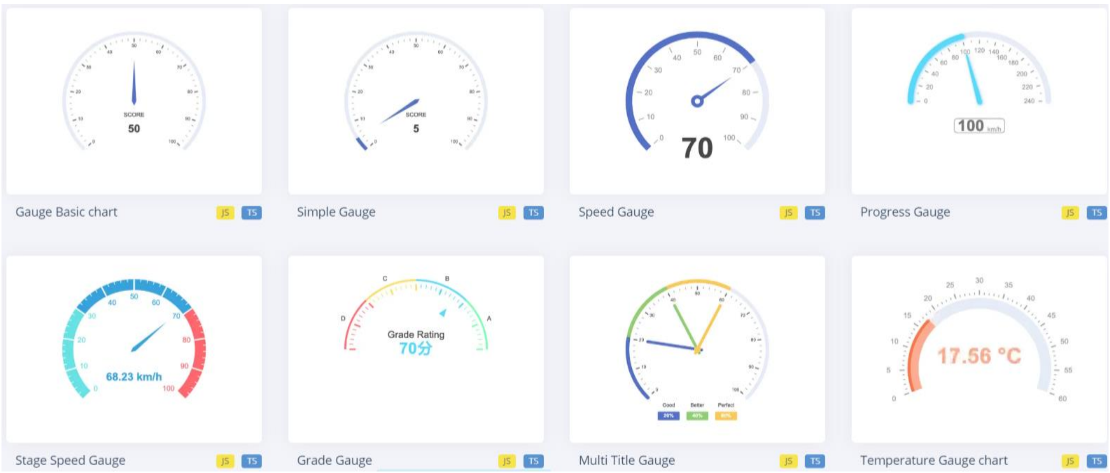

Figure 19 - Gauge examples

After experimenting with the above-mentioned frameworks, the most promising option when it comes to the variety of available visualizations and the ease of use is Echarts. Unfortunately map visualization will need to be rendered using another library since ECharts integrate Baidu Maps, a commercial web map service. Apart from Baidu Maps, ECharts can use GeoJSON maps, but this is not useful for our use case (rendering ship position during a voyage) since GeoJSON are useful for defining areas on a map.

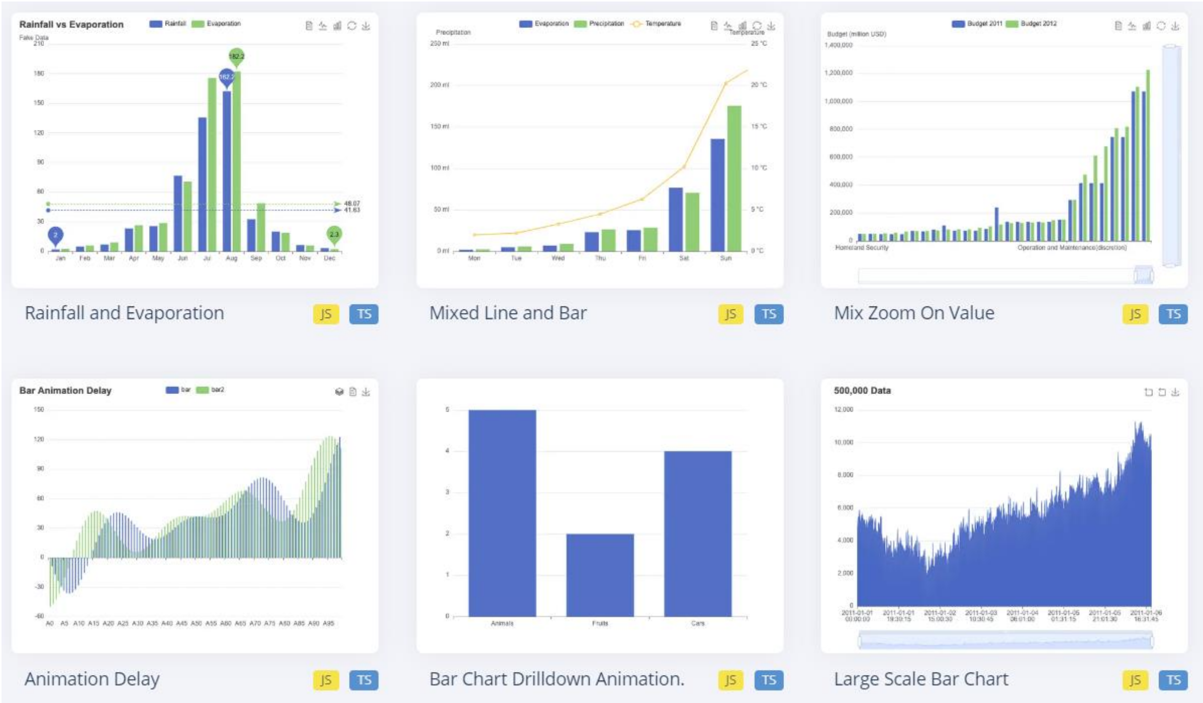

Figure 20 - Bar chart examples

Azure Maps is a collection of geospatial services and SDKs provided by Microsoft, designed to add spatial analytics, location intelligence, and mapping capabilities to applications. It offers a wide range of features including interactive maps, search, routing, traffic, and spatial operations. The Azure Maps Web SDK allows for the customization of interactive maps with custom content and imagery, suitable for integration into web or mobile applications. This platform supports various data visualization options, real-time location tracking, and geospatial analytics, making it a versatile choice for applications requiring advanced mapping functionalities. Azure Maps (Figure 21) is known for its seamless integration with other Azure services, providing a robust, secure, and scalable environment for developing location-aware applications.

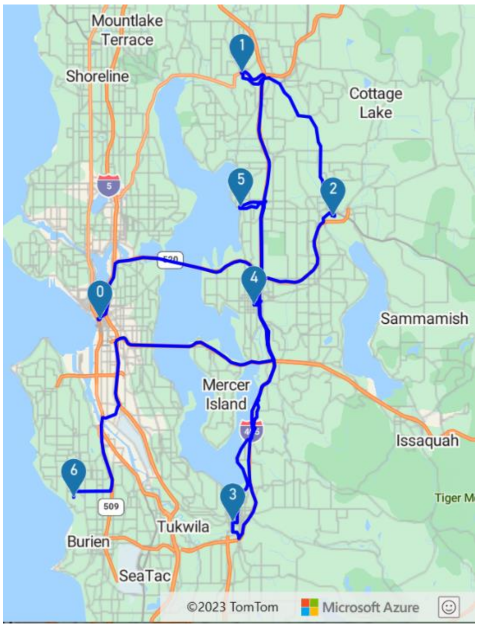

Figure 21 - Azure Maps example

Conclusions:

- We can use Echarts and Azure Maps components for most of our UI needs
- React Typescript and Electron technologies offer good options for both web-based and desktop variants.

4.3 AI Optimization Module

The purpose of the prototype is to demonstrate the possibility to optimize specific digital twin parameters based on an objective function. The focus was placed on the software aspect and not on the engineering and accuracy of the simulation or relevance of parameters for optimization.

To achieve the goals of the prototype we needed to verify two software and technology communication aspects:

1. The communication with a custom optimization algorithm (genetic algorithm)

For this prototype, a web application was created using C# and Blazor. The application communicates with a Flask server utilizing SSL/TLS encryption and enabled authentication and authorization, which hosts a genetic algorithm implemented in Python.

Genetic algorithm (Figure 22) is an optimization algorithm inspired by natural selection. It is used to solve complex optimization problems by evolving a population of potential solutions through selection, crossover, and mutation. This process is repeated over several generations until a satisfactory solution is found.

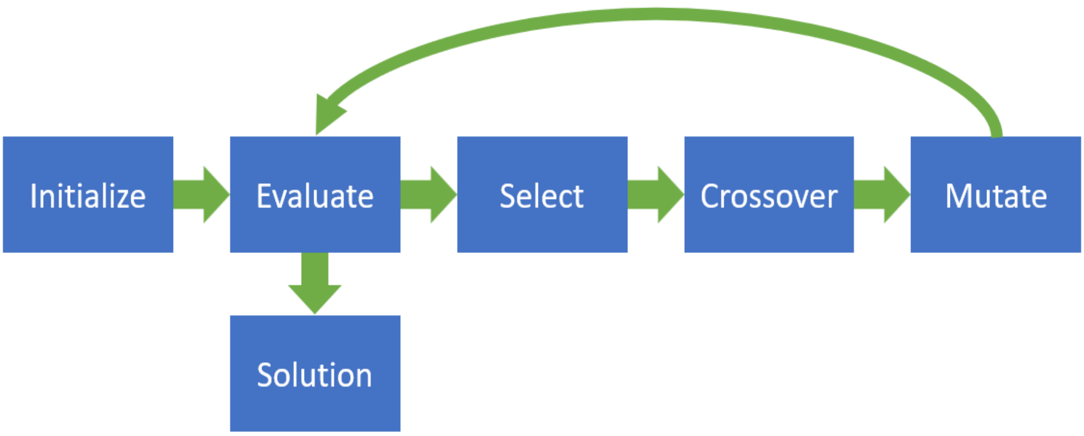

Figure 22 - Genetic algorithm approach

For this approach, we have the following steps:

- **Representation of individuals**

In our genetic algorithm, an individual represents a configuration of ship parameters that are randomly selected based on historical data, which can lead to invalid configurations. To overcome this problem, we can analyse the relationships between parameters or integrate an Amesim model to validate and suggest valid configurations in real time.

- **Evaluation function**

In our case, the evaluation function, also known as the objective function, is given by the minimum fuel consumption. The individuals with the lowest consumption will be selected for the next generation.

- **Selection**

In the selection phase of our genetic algorithm, we use a roulette method. This method takes as input the current population and its fitness values. Using a probability-based approach, it selects individuals for the next generation, favoring those with lower fuel consumption. This is consistent with our goal of minimizing fuel consumption over the evolution of potential solutions.

- **Crossover method**

At the crossing stage, our genetic algorithm uses a single-point crossing method. The crossover function mixes pairs of parents in the selected population at a specified rate. The single-point crossing method combines genetic material at a random point, producing two descendants. This process increases genetic diversity, which helps to explore potential solutions to optimize fuel consumption but can also generate invalid configurations.

- **Mutation**

In the mutation phase, our genetic algorithm introduces variability using the mutation function. This function, applied to each gene in a chromosome, randomly mutates genes based on a specified mutation rate.

Determining an optimal genetic algorithm for this specific problem is a complex task. It is evident that this approach may not represent the most straightforward or optimal solution, particularly when addressing the difficulties of the optimization process and handling potential invalid configurations. Recognizing these challenges underscores the necessity of considering alternative approaches.

2. The communication with an existing optimization software (Heeds MDO)

Heeds is a design space optimization tool that uses state of the art parameter optimization algorithms to find optimal solutions. By choosing to use Heeds we avoid having to implement manage and maintain specific AI optimization algorithms. It also ensures that as time passes the parameter optimization algorithms stay up to date with the current technology.

For our prototype we use Heeds Desktop version, and we develop a CLI application that can perform the communication needed between Heeds and a sample Simcenter Amesim model used to simulate a digital twin (Figure 23).

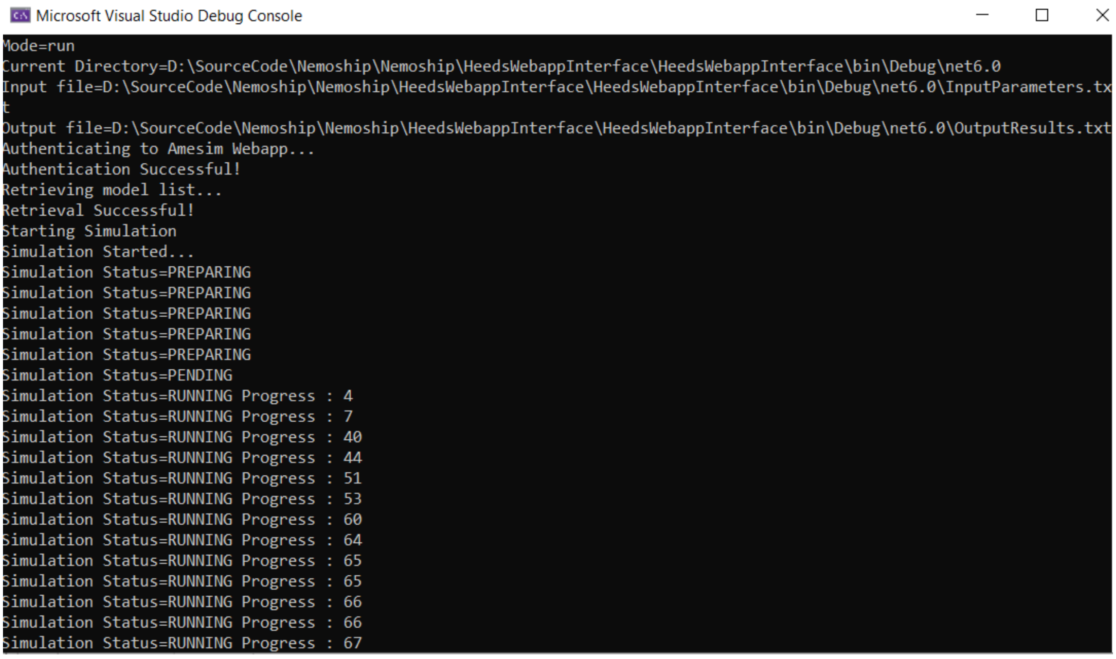

Figure 23 - CLI Application Heeds-Amesim

This CLI application is the prototype for the digital twin interface component in the NEMOSHIP cloud platform (Figure 24). The Simcenter Amesim model will be hosted and managed by the Webapp cloud service which provides a set of REST APIs that can be used to change model parameters, perform simulations, and extract the results.

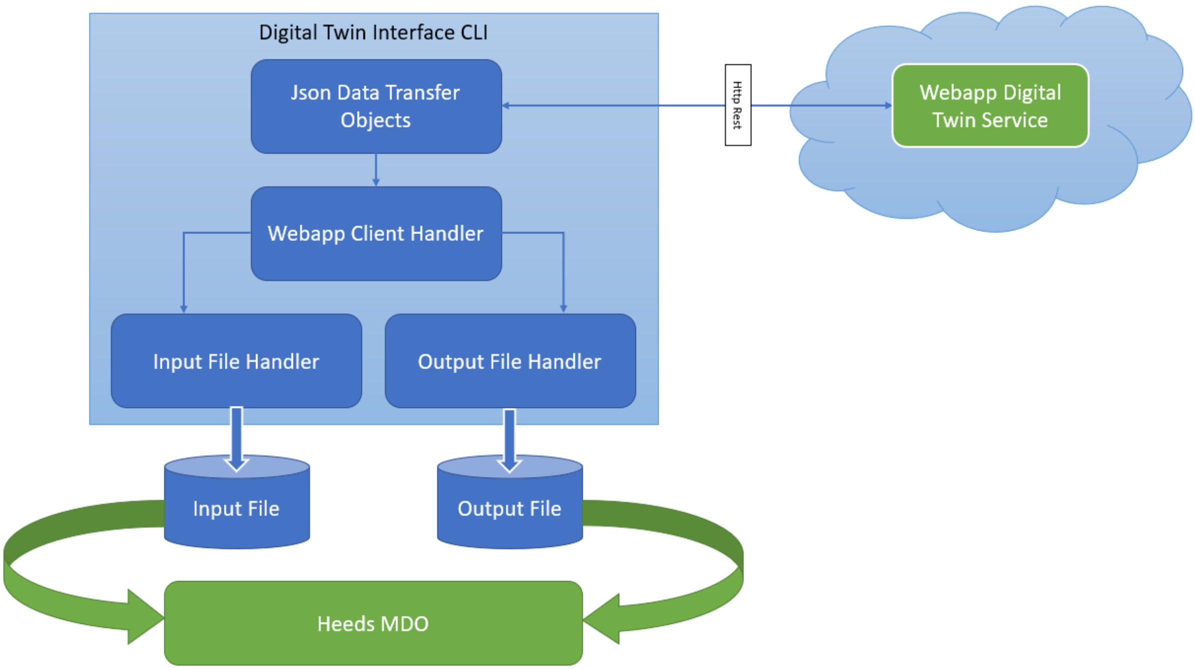

Figure 24 - CLI Application prototype for the Digital Twin

Our prototype digital twin interface is responsible for reading a parameter input file provided by Heeds, connecting to the Webapp digital twin model, changing the corresponding parameters in the digital twin model, launching a simulation and finally extracting the results in a format that can then be sent back to Heeds through an output file. To achieve this, we used the .Net Framework to create the CLI application and handle the authentication required for the webapp service.

Conclusions:

- Using Heeds MDO simplifies the optimization process
- Using .Net Framework is sufficient to achieve a full communication loop between Heeds and Webapp service that is required for the parameter optimization process.

4.4 External Platform Plugin

We wanted to investigate the possibility to create NEMOSHIP plugin modules that can be integrated in existing established maritime platforms. To this end this prototype should create a plugin that can be integrated into an existing maritime platform. The advantage for NEMOSHIP would be that the users would already be trained to use the platform and we could take advantage of the existing functionality of the platform while bringing in the additional functionality NEMOSHIP provides. As a concept and from the point of view of the NEMOSHIP Digital Platform this is technically feasible. However, the existing platforms that we contacted do not have built-in support for plugins and thus it would require extra work on their side to make it possible.

Conclusion: This prototype allowed us to see the issues we could encounter while developing a plugin module.

4.5 Authentication and Authorization Prototype

The purpose of the Authentication and Authorization Prototype (Figure 25) is to showcase a secure login process. The application aims to provide a seamless experience for users attempting to log in and differentiate between administrators and regular users.

The Authentication and Authorization Prototype is a Windows Presentation Foundation (WPF) application designed to demonstrate authentication and authorization using the Azure Active Directory B2C service.

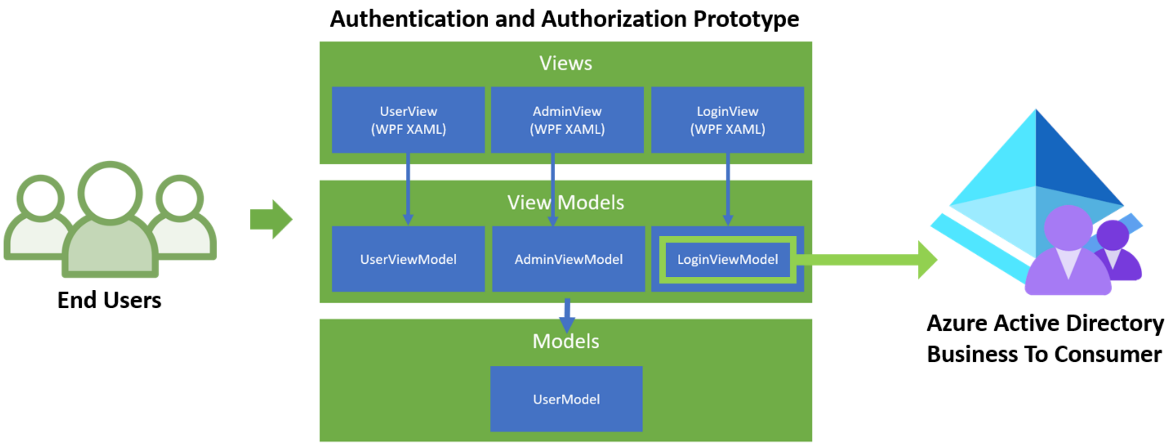

Figure 25 - Authentication and Authorization Prototype architecture

Azure Active Directory Business To Consumer (Azure AD B2C) is a comprehensive identity and access management service that stands out for its robust cybersecurity features. Designed to empower businesses with a secure and seamless customer authentication experience, Azure AD B2C employs advanced security protocols to safeguard user identities and sensitive information. One of its key strengths lies in its ability to support multi-factor authentication, adding an extra layer of security by requiring users to verify their identity through multiple means. Additionally, Azure AD B2C offers risk-based conditional access policies, allowing organizations to dynamically adjust access controls based on the perceived risk level of a user or a specific transaction. This adaptability enhances the service's capacity to thwart potential cyber threats, ensuring that only authorized users gain access to critical resources. Furthermore, Azure AD B2C adheres to industry-leading security standards and compliance certifications, underscoring its commitment to maintaining the highest cybersecurity standards.

The prototype has the following functionalities (Figure 26):

- **User Authentication:** The prototype allows users to log in securely using Azure AD B2C.
- **Role-Based Access:** The prototype demonstrates role-based access control by checking if the logged-in user has an "Administrator" role. Depending on the role, the application navigates to either the administrator page or the user page.

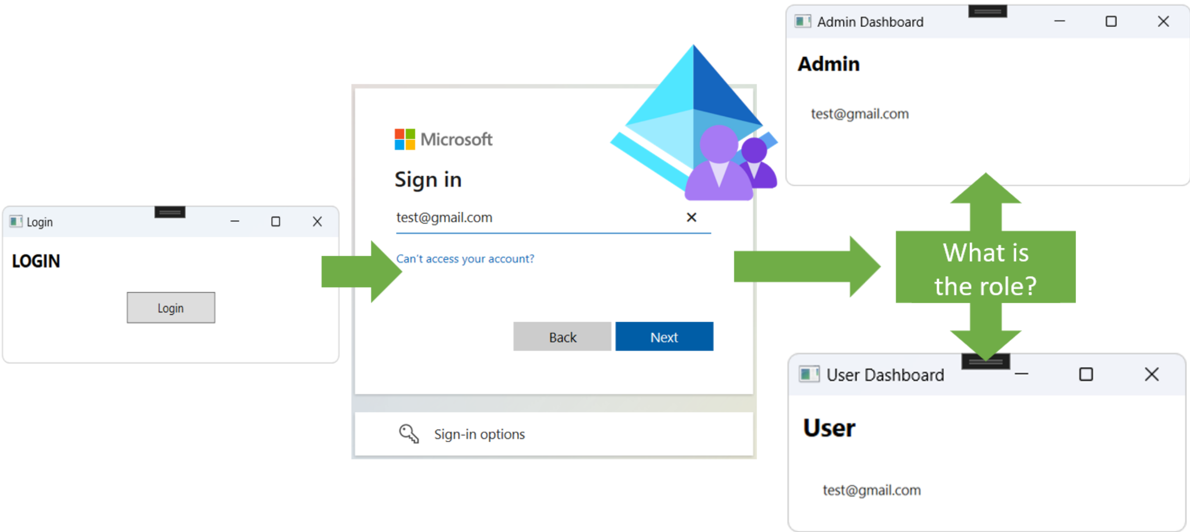

Figure 26 - Authentication and Authorization Prototype workflow

The Authentication and Authorization Prototype successfully achieves its goals by implementing a secure and intuitive authentication process using Azure AD B2C. The use of role-based access control enhances the application's versatility, allowing for different views and functionalities based on the user's role.

The WPF application, coupled with Azure AD B2C services, offers a robust foundation for building scalable and secure identity management solutions.

5 Architecture Decisions

To help us make the best choices when it comes to the architecture decisions for the NEMOSHIP digital cloud platform we rely on the features described in the Architecture Overview section. These features represent the most important considerations that we need to keep in mind when we decide on various technical and technology related choices. The features are ranked in order of priority for the NEMOSHIP project, and the priority corresponds to the goals and requirements of the project.

To be able to use the list we need to rank each possible choice accordingly by giving a score for each architectural features that are impacted by it. The score will be represented by either a + or - value, indicating how the choice will be impacting the specific feature. Some decisions do not directly impact any of the architecture features in which case we will argument the decisions based on what they do impact.

This process will guide us to making informed decisions based on the priorities that are relevant to the architecture of the NEMOSHIP digital platform.

5.1 Predictive AI Module vs scaling Webapp deployment

The need for a Predictive AI model comes from a performance requirement: the guidance should be given at a max 15 min intervals. This means that it should take no longer than 15 min to find the optimal values for the parameters selected for optimization which means that all the simulations that the Optimization module needs to run must complete in this time as well. From initial testing we have seen a single simulation taking up to 1 min. This limits the total number of tries that the Optimization module can run significantly. The solutions are (Table 4):

1. Implement and train a predictive AI module that would be capable to produce a limited result set comparable to the digital twin's Simcenter Amesim model runs with an acceptable error margin. The AI module will complete this run in a fraction of the time and therefore help decrease the number of actual simulations of the Simcenter Amesim models. The downside is that this will increase development costs.
2. Webapp allows for good scalability which means we could run multiple simulations in parallel which would help increase the number of total simulations that the Optimization module can run. The downside is that this will increase running costs.

Table 4 - Predictive AI Module vs scaling Webapp deployment

Criteria	Custom Predictive AI	Webapp
Cost-Efficiency	⊖ **Effort will be needed to train the model and gather and prepare the training data.**	⊕ **While the webapp license does represent a running cost it is much lower than the effort it would take to replace it.**
Scalability	⊕ **Improves vertical scaling**	⊕ **Improves horizontal scaling**

Decision: Use both approaches, the Webapp service for horizontal scaling and a predictive AI module for vertical scaling if needed.

5.2 Digital twin modeling Tools

The Digital Twin module relies on state-of-the-art simulation software to be able to correctly predict the real-world vessel state. Choosing which simulation software is used is therefore highly important from multiple points of view (Table 5). The choices are:

1. Simcenter Amesim-Standalone
2. Simcenter Amesim-Webapp

3. Matlab Simulink-Standalone
4. Simcenter Amesim and Matlab Simulink (co-simulation)

Optimization Module Tools

For the core engine of the optimization module, we have a couple of choices:

1. Heeds MDO
2. Custom AI Optimization Component
3. Optuna framework

Heeds is an established optimization platform that can be used for parameter and design optimization. It offers both local and cloud deployments and can be accessed through an API.

We could implement our own custom AI Optimization Component. This would allow us to customize and train the AI algorithm using the available historical data for the two physical vessels. The downside is that this will increase the complexity of the overall implementation and deployment since we would also need to replicate part of what is already implemented in Heeds.

We conducted preliminary experimentation with the Optuna framework for the AI algorithm. Optuna is an open-source tool designed for optimizing hyperparameters in machine learning models. Challenges emerged when defining a precise objective function to optimize fuel consumption while using this framework. Establishing this function is crucial for enabling Optuna to effectively navigate the parameter space.

Table 5 – Digital twin modelling tools

Criteria	Simcenter Amesim Webapp	Matlab Simulink
Complexity	⊕ Reduced complexity due to existing cloud API	⊖ Increased complexity when working in cloud and interfacing with other services
Scalability	⊕ Supports horizontal and vertical scaling	⊕ Supports horizontal and vertical scaling but without an API to control it.

Decision: Since Simcenter Amesim Webapp is capable of using existing Matlab Simulink models we can eliminate Matlab Simulink as a dependency with the mention that should there be any scenario in which it is not feasible to use only Simcenter Amesim Webapp, the architecture is designed in such a way as to allow the seamless integration of Matlab Simulink.

5.3 Cloud Platform

For cloud platforms we have reduced the number of choices to the following:

1. Microsoft Azure:
   - Integration with .Net Technologies: Azure, being a Microsoft product, naturally offers extensive support for .Net based technologies. This is a significant advantage for organizations heavily invested in .Net, as it allows for seamless integration and efficient deployment of existing .Net applications.

- Enterprise-Friendly Environment: Azure is known for its strong emphasis on enterprise needs. It offers a range of services tailored for large-scale business applications, including advanced security features, compliance offerings, and a robust set of tools for analytics, machine learning, and Internet of Things (IoT) applications.
- Hybrid Cloud Capabilities: Azure provides robust solutions for hybrid cloud environments, which is beneficial for businesses that want to maintain some of their workloads on-premises (local) while leveraging the cloud for others.
- Microsoft Ecosystem Compatibility: Organizations already using a range of Microsoft products (like Office 365, Outlook, etc.) may find Azure more appealing due to its integrated nature within the Microsoft ecosystem.

2. Amazon Web Services (AWS):
    - Market Leadership and Innovation: AWS is a market leader in the cloud space and is known for its rapid pace of innovation. It frequently introduces new services and updates to its existing offerings.
    - Wide Range of Services: AWS offers an extensive array of services covering various aspects such as computing power, storage options, and advanced data analytics, which can cater to a wide range of application requirements.
    - Community and Ecosystem: AWS has a vast developer community and a broad ecosystem of partners, which can be beneficial for support, resources, and third-party tools.
3. Comparative Analysis:
    - Feature Set and Performance: Both platforms offer a comprehensive set of features that are quite similar in capabilities. Performance metrics such as uptime, compute power, and data management are comparable, with slight variations depending on specific use cases.
    - Pricing Models: The pricing structures of both AWS and Azure are similar, offering pay-as-you-go models. However, the actual cost can vary based on the specific services used, the scale of deployment, and any long-term commitments.
    - Support and Documentation: Both platforms provide extensive documentation and support. Azure's advantage in .Net support is a key differentiator for businesses primarily using .Net for their software development.

In conclusion, while both Microsoft Azure and Amazon Web Services are competent and broadly equivalent in terms of general features and pricing, the decision for NEMOSHIP is guided by the specific needs, especially the existing technology stack, scalability requirements, and specific cloud service needs.

Decision: Microsoft Azure

5.4 AI Algorithms

The AI algorithms used by the digital platform are split in two main categories: Predictive Algorithms and Optimization Algorithms.

5.4.1 Predictive Algorithms

Predictive AI algorithms are a cornerstone in the field of machine learning and data analysis. Below is a detailed list of some of the most prominent algorithms, each with its own unique characteristics and applications:

1. Linear Regression:
    - Description: A statistical method used to model the relationship between a dependent variable and one or more independent variables. The algorithm assumes a linear relationship between the variables.
    - Applications (examples): Used in economics, real estate, business, and other fields for trend forecasting.
    - Strengths: Simplicity and interpretability.
    - Limitations: Assumes a linear relationship, which is not always the case in real-world data.
2. Logistic Regression:
    - Description: Similar to linear regression but used for binary classification problems. It models the probability of a certain class or event.
    - Applications (examples): Medical fields for disease diagnosis, banking for credit scoring.
    - Strengths: Provides probabilities for outcomes; good for classification.
    - Limitations: Assumes linearity between the dependent and independent variables.
3. Decision Trees:
    - Description: A tree-like model of decisions and their possible consequences. It splits the data into subsets based on the value of input features.
    - Applications (examples): Used in finance for risk assessment, healthcare for diagnosing.
    - Strengths: Easy to understand and interpret.
    - Limitations: Prone to overfitting; can be unstable.
4. Random Forest:
    - Description: An ensemble learning method that operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes of individual trees.
    - Applications (examples): Broad use in classification and regression tasks, like stock price prediction.
    - Strengths: Reduces overfitting in decision trees; highly accurate.
    - Limitations: More complex and computationally intensive.
5. Support Vector Machines (SVM):
    - Description: A supervised learning model used for classification and regression analysis. It finds the hyperplane that best divides a dataset into classes.
    - Applications (examples): Face detection, text and hypertext categorization.
    - Strengths: Effective in high dimensional spaces.
    - Limitations: Not suitable for larger datasets; less effective on noisier datasets with overlapping classes.
6. Neural Networks:
    - Description: Inspired by the structure of the human brain, these algorithms are designed to recognize patterns using a large number of processing nodes.
    - Applications (examples): Widely used in image and speech recognition, gaming, language processing.
    - Strengths: Can model complex, non-linear relationships.

- Limitations: Requires a lot of data; can be a "black box" in terms of interpretability.

7. K-Nearest Neighbors (KNN):
    - Description: A simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions).
    - Applications (examples): Recommendation systems, classification in finance and healthcare.
    - Strengths: Simple and effective.
    - Limitations: Slows significantly as the size of the data in use grows.
8. Naive Bayes:
    - Description: A set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of independence between every pair of features.
    - Applications (examples): Spam filtering, text classification, sentiment analysis.
    - Strengths: Works well with high dimensions; fast.
    - Limitations: Based on the assumption that features are independent, which is rarely the case.
9. Gradient Boosting Machines (GBM):
    - Description: An ensemble technique that builds the model in a stage-wise fashion; it generalizes by allowing optimization of an arbitrary differentiable loss function.
    - Applications (examples): Used in web search ranking and ecology.
    - Strengths: Highly effective and flexible.
    - Limitations: Prone to overfitting; computationally intensive.
10. Deep Learning:
    - Description: A subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled.
    - Applications (examples): Autonomous vehicles, AI in healthcare, natural language processing.
    - Strengths: Excellent performance on a range of complex tasks.
    - Limitations: Requires large amounts of data and computational power; can be difficult to interpret.

Decision: Considering our needs, available data and limitations we concluded that Neural Networks AI algorithms will fit our needs the best since we expect our data size to be large and non-linear but not large enough to train a deep learning algorithm.

5.4.2 Optimization Algorithm

AI optimization algorithms are a cornerstone of modern computational techniques, designed to find optimal or near-optimal solutions to complex problems where traditional methods fall short. These algorithms range from those inspired by natural processes and biological evolution to sophisticated statistical models. They excel in handling large-scale, multi-dimensional, and non-linear problems, making them indispensable in fields like machine learning, logistics, finance, and engineering. While powerful, they often require careful tuning and understanding of the underlying problem to be effective.

1. SHERPA
    - Description: SHERPA is a Python library for hyperparameter optimization of machine learning models. It uses a combination of algorithms, including Bayesian optimization, grid search, and random search, to efficiently find the best hyperparameters.

- Applications (examples): Primarily used for hyperparameter tuning in machine learning models, including but not limited to deep learning networks, ensemble methods, and support vector machines.
- Strengths:
  - Offers a variety of algorithms, making it versatile for different optimization tasks.
  - Integrates easily with popular machine learning frameworks.
  - Efficient in finding optimal parameters compared to manual tuning.
- Limitations:
  - Requires a good understanding of the underlying model to set appropriate parameter ranges.
  - The efficiency of optimization can vary depending on the complexity of the model and the initial parameter space.
  - Might be overkill for simple models or when computational resources are limited.

2. Genetic Algorithms
   - Description: Genetic Algorithms are inspired by natural selection, using mechanisms such as mutation, crossover, and selection to evolve solutions to problems.
   - Applications (examples): Used in machine learning, data mining, optimization of complex systems, and scheduling problems.
   - Strengths:
     - Good at exploring a large and complex search space.
     - Robust and can handle noisy data.
   - Limitations:
     - Can be slow and computationally expensive.
     - May converge to suboptimal solutions.
3. Simulated Annealing
   - Description: This technique mimics the annealing process in metallurgy, where controlled cooling is used to reduce defects in materials. It is used to find the approximate global optimum of a function.
   - Applications (examples): Schedule optimization, routing problems, and energy minimization in physical systems.
   - Strengths:
     - Good at avoiding being trapped in local optima.
     - Simple and easy to implement.
   - Limitations:
     - Performance depends heavily on the cooling schedule.
     - Can be slow for complex problems.
4. Particle Swarm Optimization
   - Description: PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality.
   - Applications (examples): Neural network training, function optimization, and control problems.
   - Strengths:
     - Simple to implement and requires few parameters.
     - Effective in high-dimensional spaces.

- Limitations:
    - Can be prone to getting stuck in local optima.
    - May have slow convergence rates for certain problems.

5. Gradient Descent
    - Description: A first-order iterative optimization algorithm to find the local minimum of a differentiable function.
    - Applications (examples): Fundamental in training machine learning models, especially in deep learning.
    - Strengths:
        - Widely used and well-understood.
        - Efficient for large datasets and high-dimensional spaces.
    - Limitations:
        - Can get stuck in local minima.
        - Sensitive to the choice of learning rate.
6. Evolutionary Strategies
    - Description: Similar to genetic algorithms but focuses more on mutation. Effective for continuous parameter optimization.
    - Applications (examples): Robotics optimization, real-time strategy game AI, and complex function optimization.
    - Strengths:
        - Does not require gradient information.
        - Effective in high-dimensional and complex search spaces.
    - Limitations:
        - Computationally intensive.
        - May have slower convergence compared to other methods.
7. Ant Colony Optimization
    - Description: Inspired by ant foraging behavior, it is used to find optimal paths in graphs.
    - Applications (examples): Routing, network topology optimization, and scheduling problems.
    - Strengths:
        - Good at solving discrete and combinatorial optimization problems.
        - Robust and adaptive.
    - Limitations:
        - Not suitable for continuous optimization problems.
        - Can be computationally expensive.
8. Bayesian Optimization
    - Description: Used for optimizing expensive-to-evaluate objective functions, particularly useful in hyperparameter tuning.
    - Applications (examples): Hyperparameter optimization in machine learning and tuning of complex systems.
    - Strengths:
        - Efficient for problems with expensive evaluations.
        - Balances exploration and exploitation effectively.
    - Limitations:
        - Requires a good probabilistic model.

- Not as effective for high-dimensional problems.

Decision: Based on our prototyping both SHERPA and Genetic Algorithms have yielded interesting results, we therefore defer the choice between the two to the implementation phase when more concrete decision points will be available.

5.5 UI Technologies

For the user interface the choice of technology comes between .Net WPF which is a desktop-based framework and Typescript, Electron and React which are web based.

WPF, a mature framework designed for developing rich desktop applications on the .NET platform, offers a robust environment for UI development with extensive support for graphics, animation, and media. However, the ecosystem for open-source graph and chart controls in WPF, while existent, may not be as expansive or rapidly evolving as in the web domain.

In contrast, web-based technologies like React, TypeScript, and Electron present a dynamic and continuously growing landscape for UI development. These technologies, particularly in conjunction with modern JavaScript libraries, offer a wealth of high-quality, open-source options for graph and chart controls. The web development community, known for its collaborative and innovative nature, contributes to a vast repository of controls that are regularly updated, optimized, and extended with new features. This results in a rich selection of graph and chart controls that are not only visually appealing but also highly customizable and responsive, catering to the diverse and evolving needs of web applications.

Electron, which bridges the gap between desktop and web applications, leverages the strengths of web technologies by allowing developers to create desktop applications with web technologies. This integration means that developers can utilize the vast array of graph and chart controls available in the web ecosystem within a desktop application context, offering a blend of both worlds.

Therefore, while WPF stands strong in its domain with its own set of advantages for desktop application development, the web-based technologies, especially when considering the aspect of open-source graph and chart controls, offer a more varied and rapidly advancing landscape. This reflects a broader trend in software development, where open-source web technologies are increasingly setting the pace for innovation and community-driven development.

Decision: The chosen technologies need to be web-based so that we can utilize the existing open-source resources.

5.6 Backend Technologies

When evaluating backend technology choices, particularly .NET versus other popular options like Node.js, Python frameworks, and Ruby on Rails, a distinct advantage of .NET emerges in several key areas. .NET, a framework developed by Microsoft, has established itself as a robust, versatile, and efficient platform for building a wide range of applications, from web to mobile to desktop.

One of the primary strengths of .NET lies in its performance. With the advent of .NET Core and its evolution into .NET 5 and beyond, the framework has significantly improved in terms of speed and efficiency. This is particularly noticeable in high-load scenarios, where .NET demonstrates superior performance compared to Node.js and dynamic languages like Python and Ruby. The optimized runtime and extensive libraries enable developers to build highly efficient and scalable applications.

Another compelling aspect of .NET is its language versatility. Primarily using C#, a language known for its powerful features and strong typing, .NET provides a robust environment for backend development. C#'s syntax is expressive yet concise, and the language continually evolves, incorporating modern programming features. This contrasts with languages like JavaScript in Node.js, which, despite their flexibility, can sometimes lead to less maintainable code due to dynamic typing.

Furthermore, .NET excels in enterprise-level application development. It offers a comprehensive ecosystem that includes ASP.NET for web applications, Entity Framework for data access, and a wide array of APIs for integrating with various services. This cohesive ecosystem simplifies the development process, especially for complex enterprise applications, which might be more challenging to achieve with Node.js or Python's Django and Flask, where third-party libraries are often needed to provide equivalent functionality.

Security is another crucial aspect where .NET shines. Microsoft has invested heavily in making .NET a secure platform. Regular updates, comprehensive documentation, and a strong emphasis on security best practices contribute to making .NET a reliable choice for applications where security is paramount.

Lastly, .NET's support for cross-platform development is a significant advantage. With .NET Core and later versions, developers can build applications that run seamlessly on Windows, Linux, and macOS. This cross-platform capability provides a level of flexibility and scalability that is highly valued in today's diverse computing environments.

In summary, while other backend technologies like Node.js and Python offer their own set of advantages, .NET stands out due to its performance, language robustness, comprehensive ecosystem, security focus, and cross-platform support. These aspects make it an excellent choice for the NEMOSHIP Cloud Platform.

Decision: .Net based backend technology stack

5.7 Cyber Security

In the NEMOSHIP Digital Cloud Platform project, a critical aspect of our architectural strategy revolves around ensuring robust cybersecurity. In this context, our decision to integrate Azure Active Directory B2C (Azure AD B2C) [8] emerges as a pivotal move. This selection is not just a mere addition to our technology stack but a strategic alignment with our overarching goal of providing a secure, scalable, and seamless digital experience.

Azure AD B2C stands out as a highly capable cloud-based identity management service, designed to facilitate applications accessed by a varied user base. Its core strength lies in its robust security and compliance features. These include advanced multi-factor authentication, which adds an additional layer of security beyond traditional password-based access; real-time threat detection, which proactively identifies and mitigates potential security breaches; and compliance with numerous data protection regulations, ensuring that the platform adheres to the latest standards in data security and privacy.

The scalability of Azure AD B2C is another key factor in its selection. In the dynamic digital landscape, where user volumes and demands can fluctuate dramatically, Azure AD B2C's cloud-based architecture allows it to dynamically scale. This scalability is crucial for maintaining a consistent and high-performing user experience, particularly during periods of high demand or sudden spikes in user activity.

Customization capabilities of Azure AD B2C also play a significant role. The service allows for extensive customization of user journeys, including sign-up, sign-in, and profile management processes. This flexibility is vital for creating a user experience that is not only functional but also resonates with the NEMOSHIP goals. The ability to integrate custom user interfaces and align them with our maritime domain enhances the overall user interaction with the platform, making it more engaging and intuitive.

Integration with other Azure services and various external applications is another compelling reason for choosing Azure AD B2C. This interoperability is pivotal in creating an interconnected cloud ecosystem, which is essential for the NEMOSHIP platform's efficient operations and streamlined data flows. The seamless integration ensures that various components of our cloud infrastructure can communicate and function harmoniously, enhancing the overall efficiency and effectiveness of the platform.

From a development perspective, Azure AD B2C presents a familiar and developer-friendly environment, especially for teams with substantial experience in .Net C#. It supports a range of programming languages and frameworks, thus reducing the learning curve and simplifying the development process. This compatibility is crucial for ensuring that our team can quickly adapt to and efficiently work with Azure AD B2C, minimizing development time and accelerating the deployment of new features.

The cost-effectiveness of Azure AD B2C cannot be overlooked. Its pay-as-you-go pricing model is particularly suited for a growing platform like NEMOSHIP. This model allows for predictable budgeting and reduces upfront costs, a crucial factor for any digital platform that needs to manage its resources effectively while scaling up.

Implementing Azure AD B2C in the NEMOSHIP platform involves a comprehensive strategy. It begins with detailed integration planning in collaboration with the development team. This stage includes outlining the integration process, configuring user flow, and identifying customization requirements. Following this, the focus shifts to setting up and rigorously testing security policies, encompassing multi-factor authentication, conditional access, and threat detection mechanisms.

The customization and branding phase involves designing user interfaces for the authentication processes in alignment with the NEMOSHIP brand identity. This step is crucial for ensuring that the platform's visual and interactive elements resonate with our users and provide a cohesive and branded experience.

Testing and validation are integral to the implementation process. This phase involves conducting thorough testing to ensure that the integration meets our stringent standards for security, performance, and user experience. It is during this phase that any potential issues are identified and rectified, ensuring the reliability and robustness of the platform.

Finally, continuous monitoring and maintenance form the backbone of our long-term strategy with Azure AD B2C. This ongoing process involves regularly updating security policies, refining user experience features, and ensuring that the platform remains aligned with the latest cybersecurity trends and challenges.

Azure Active Directory B2C (Azure AD B2C), in its role as a part of the NEMOSHIP Digital Cloud Platform's cybersecurity framework, not only meets general cybersecurity requirements but also aligns with specific international standards such as ISO 27001 and IEC 62443. The integration of Azure AD B2C supports compliance with these standards in several ways:

1. ISO 27001 Compliance: ISO 27001 is a widely recognized standard for information security management systems (ISMS). It provides a framework for establishing, implementing, operating, monitoring, maintaining, and improving an ISMS.
    - Risk Management: Azure AD B2C includes robust risk management features that align with ISO 27001's emphasis on identifying and managing information security risks. This includes regular security assessments and proactive threat detection.
    - Data Protection and Privacy: The service's compliance with data protection laws and regulations supports ISO 27001's requirements for data security and privacy.
    - Access Control: Azure AD B2C's strong access control mechanisms, including multi-factor authentication and conditional access policies, align with ISO 27001's control objectives related to access management.
    - Continuous Improvement: Azure AD B2C's ongoing updates and improvements support the ISO 27001 principle of continuous improvement in security management.
2. IEC 62443 Compliance: IEC 62443 is a series of standards focused on the security of Industrial Automation and Control Systems (IACS). While primarily targeted at industrial environments, its principles are broadly applicable to any information system, especially in terms of network and system security.
    - System Segregation: Azure AD B2C facilitates secure segregation of systems, an important aspect of IEC 62443, ensuring that user access is appropriately managed and controlled.
    - Security Patch Management: Regular updates and security patches in Azure AD B2C align with IEC 62443's requirements for maintaining the security of software and systems.
    - Incident Management: Azure AD B2C's incident management capabilities, including its ability to detect and respond to security incidents, align with IEC 62443's focus on incident preparedness and response.

By incorporating Azure AD B2C into the NEMOSHIP Digital Cloud Platform, the platform can leverage these features to support compliance with both ISO 27001 and IEC 62443.

In conclusion, adopting Azure AD B2C for the NEMOSHIP Digital Cloud Platform is a strategic decision. It addresses not only our immediate cybersecurity concerns but also provides a platform that is flexible, developer-friendly, and capable of evolving with the changing digital landscape. This integration ensures that the NEMOSHIP platform remains at the forefront of cybersecurity, offering a secure, scalable, and user-friendly environment for our user base.

6 Conclusion

Pioneering a New Era in Maritime Technology: The NEMOSHIP Digital Cloud Platform

The NEMOSHIP project represents a groundbreaking endeavor in the maritime industry. NEMOSHIP will develop a modular and standardised battery energy storage solution that is able to exploit heterogeneous storage units and a cloud-based digital platform to enable data-driven, optimized, and safe exploitation. The project will demonstrate the maturity of these innovations at TRL 7 for hybrid ships as well as their adaptability towards fully electric ships. This report presented the proposed platform's design and operational framework which will be developed and implemented during the project.

Technological Integration and Innovation

At the heart of NEMOSHIP lies a commitment to technological excellence, exemplified by the utilization of the .Net technology stack. This choice is a cornerstone in the project's architecture, offering a robust and versatile framework capable of handling complex communication protocols with unprecedented efficiency. The project's journey in the realm of user interface development has been marked by an astute adoption of web-based technologies, including React, TypeScript, and Electron. This strategic decision leveraged the dynamic capabilities of modern web interfaces, resulting in a user experience that is both intuitive and functionally rich, setting a new benchmark in maritime software applications.

The AI optimization module stands as a testament to the project's innovative spirit. The ingenious integration of a custom optimization algorithm, developed using a synergy of C# and Python, showcases proficiency in bridging different technological paradigms for a common goal.

Strategic Cloud and Security Solutions

Selecting Microsoft Azure as the cloud platform was a strategic move that allows enhancing the project's scalability and security. This choice ensures that the project remains at the forefront of technological advancements, ready to incorporate future innovations in cloud computing.

The project's approach to security, particularly through the implementation of Azure Active Directory B2C for authentication and authorization, underscores a deep commitment to protecting sensitive maritime data. This robust security framework has set a new standard in the industry, ensuring that access to critical systems is both secure and seamlessly controlled.

AI Algorithms: The Future of Maritime Optimization

The decision to employ neural networks and genetic algorithms for predictive modeling and optimization marks a significant leap forward in maritime technology. These AI algorithms will not only enhance the project's capability to make data-driven decisions but have also opens new avenues for exploration in predictive maintenance, route optimization, and fuel efficiency. The potential of these algorithms to revolutionize maritime operations cannot be overstated, as they provide a glimpse into a future where AI-driven insights lead to safer, more efficient, and environmentally friendly maritime travel.

A Vision for Sustainable and Efficient Maritime Operations

As NEMOSHIP moves forward, it embodies the promise of a new era in maritime technology characterized by sustainability, efficiency, and technological prowess. The project is laying a solid foundation for the future with the integration of advanced digital solutions in maritime contexts.

7 References

[1] Microsoft, "Microservice architecture style," Microsoft, [Online]. Available: https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/microservices.

[2] Microsoft, "N-Tier architecture style," Microsoft, [Online]. Available: https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/n-tier.

[3] Marorka, "Marorka," [Online]. Available: https://ascenzmarorka.com/.

[4] Maress, "Maress," [Online]. Available: https://www.maress.no/login.

[5] Microsoft, "Overview of gRpc," Microsoft, [Online]. Available: https://learn.microsoft.com/en-us/aspnet/core/grpc/?view=aspnetcore-8.0.

[6] Microsoft, "Web API .Net," [Online]. Available: https://learn.microsoft.com/en-us/aspnet/core/web-api/?view=aspnetcore-8.0&WT.mc_id=dotnet-35129-website.

[7] Siemens, "Simcenter Amesim Documentation," Siemens, [Online]. Available: https://support.industrysoftware.automation.siemens.com/html/documentation.shtml.

[8] Microsoft, "Azure AD B2C Documentation," [Online]. Available: https://learn.microsoft.com/en-us/azure/active-directory-b2c/.