Open-Source Web Service for Fast and Reliable Handling of Weather Data for Use in Maritime Simulations

Jon S. Dæhlen, SINTEF Ocean AS, Trondheim/Norway, jon.daehlen@sintef.no
Håvard Heierli-Nesse, SINTEF Ocean AS, Trondheim/Norway, havard.nesse@sintef.no
Torstein A. Bø, SINTEF Ocean AS, Trondheim/Norway, torstein.bo@sintef.no

Abstract

Efficient ship design requires understanding the impact of operational and environmental conditions on the final ship. Discrete-event simulation and quasi-static ship models using MetOcean data can help, but storing the large amounts of weather data required can be impractical, and accessing data directly through open services can be slow and unreliable. Furthermore, metrological organizations updating or deleting data can compromise the reproducibility of simulation results. To address these issues, we propose a web service that caches MetOcean data between the ship simulator and the met-rological organizations services, cloud-based file storage and a PostGIS database used as storage layer. This ensures that data is not requested more than once for a given temporal and geographical coordinate, and remains available as long as needed. The performance of the new service is bench-marked against downloading data and accessing it directly using OpenDAP.

1. Introduction

Meeting IMOs goal of 40% GHG reduction in the maritime sector by 2030 and 50% by 2050, *IMO (2018)*, a variety of new fuels, ship designs and energy-saving devices and strategies has been conceived in a frenzy of opportunistic innovation. To reach the IPCC goal of limiting global warming to 1.5°C, the worlds GHG emissions need to be net zero within the decade, *IPCC (2022).* Although retrofitting existing ships with low-, or zero-carbon fuels are possible, fuels such as hydrogen and ammonia possesses explosive, toxic and space-demanding properties implying requirements to the ship design and construction to meet regulations and become practical. New fuels also being more expensive for each unit of energy delivered to the propulsor, underlines the importance of energy efficiency, which may be achieved by tailoring the ship design to conditions and operational requirements it will face if built. New builds that are conceived as zero-emission ships from the ground up gives the flexibility to accommodate these factors, if the designer can reveal the true impact of a given design choice on the economic and environmental performance.

In practice, this impact is hard to understand and quantise, much due to the sheer complexity of a ship system, not to mention when put into a logistic transportation system with factors such as commercial requirements, fuel availability and weather conditions. Traditionally, this is handled by using an existing ship design with similar requirements as a starting point for a design spiral. With the usual pressure on the naval architect in terms of an approaching deadline, it may be hard or almost impossible to do bold explorations of the design space and being sure that a design change has no negative effect on some other part of the ship or in the logistic system.

The rapid increase of available computational power at the naval architect’s desktop during the past decades has revolutionized our ability to evaluate the performance of a given ship design. Empirical, frequency domain and time domain analysis methods are blended into tools that can evaluate the ship design increasingly more holistic. Examples of recent significant developments are:

- Ship design suites such as NAPA, ShipX and WAMIT, each incorporating several tools for ship design and analysis, often focusing on empirical and frequency-domain analysis in addition to graphical tools for altering the hull.
- Computational Fluid Dynamics software such as Star-CCM+ and OpenFOAM. Being general-purpose tools, these are still much used in the maritime domain for calm-water predictions and to a certain degree wave interactions.

- The Horizon 2020 EU project HOLISHIP applied the CAESES optimization platform to automate the exploration of the design space of parametric ship models, *Papanikolaou et al. (2022).*
- Use of co-simulation, *Skjong et al. (2018)*, through Open Simulation Platform, *Smogeli et al. (2020)*, to mix-and-match simulation models of sub-systems of a ship, such as hull, propulsion, power plant, cranes, etc., into a holistic representation of the ship concept to simulate certain manoeuvres and wave responses.
- Use of quasi-static (frequency-domain) modelling of the ship to allow for evaluation of the performance of the ship concept in a logistic transportation system, *Dæhlen et al. (2021), Fathi et al. (2013).*

General for these methods, and for ship design, is the need for knowledge of the sea-, and weather conditions that the ship will operate in. The main dimensions, hull shape and stability of the ship greatly impacts the ships performance, both in terms of energy usage and safety for crew and cargo. The naval architect often has guidelines for typical weather conditions given a geographical area of operation, used as input in the design process. However, the wave period and direction relative to the ships heading (again affected by the ship speed), determines the ship response to a given wave height. As the ship is (usually) a highly non-stationary object, determining the conditions and duration that the ship will encounter during its lifetime is an exhausting effort using traditional methods.

Retrieving representative sea-, and weather (MetOcean) data for prolonged periods of operation is possible using historical observed data, fitted onto forecast models for geospatial continuity (hindcast data). This is published openly from several major metrological services, including European Centre for Medium-Range Weather Forecasts (ECMWF) and The Norwegian Meteorological Institute (met.no). Copernicus Marine Service and SeaDataNet are examples of central services for providing access to the produced data from metrological services. Throughout this paper, MetOcean data will be used as a general term for waves, wind, and sea current data, hindcast or unprocessed, with geospatial and temporal dimensions.

The MetOcean data is usually stored as three-dimensional regularly gridded data, latitude, longitude and time, using either the GRIB, GRIB2, HDF5, or its derivate, NetCDF, file format, *Rew et al. (2006)*. In general terms the two former formats are more primitive in terms of user friendliness, however the latter tend to be more space demanding. The industry-standard software THREDDS, *Caron (2005)* is used to access and retrieve the data, *Magariño et al. (2014), Nativi et al. (2010).* A common practice is to use a sub-setter to select a reduced geographical, temporal and a selection of variables to download from the potentially enormous file on the server. This has the advantage of full control of the downloaded data, and the user has fast and reliable access to the file stored locally on the user's personal hard drive. However, several drawbacks are apparent when applied to simulation of ships on long, potentially inter-continental routes with a simulation horizon of several years:

- Depending on the resolution of the data set, the downloaded file may become impractically large (>10GB).
- Due to the nature of sub-setting, a rectangular grid must be selected for the sub-set. Seeking to cover a route going from south-west to north-east, the rectangle area is maximized, and large majority of data points will never be used. E.g. to cover an area between the Gulf of Mexico and the English Channel for one year (as described in the second use-case in Ch.4), the data set dimension is 1202 (longitude) x 525 (latitude) x 241 (time), resulting in a total of 152 083 050 data points, while the simulation only used 3057 of the points.
- It is time-consuming to figure out the geospatial and temporal limitations of the sub-set.
- In a commercial setting, but also for research, it should always be possible to reproduce a simulation at any point in time.
- The metrological services have the option of moving, altering or deleting data at any time. This conflicts with the above bullet point.
- There is no standard naming convention for the variables in the data files, neither are

projections sufficiently standardized to be automated.

- When simulating active routing decisions, *Dæhlen et al. (2022)*, the user does not know at forehand exactly where the ship will sail. This requires in the best case a conservative choice for the sub-set, resulting in a larger data set.

Some of the above points can be improved using the OpenDAP framework, allowing accessing and reading large data files directly on the server containing the full data set. For the user, the process of accessing files through OpenDAP resembles accessing files stored locally. When the user, or a software using the data files requests a data point, it will be routed to the remote file location.

MetOcean data is available through OpenDAP from several metrological services, including Copernicus Marine and SeaDataNet. The GYMIR simulator has been using OpenDAP successfully, however it puts great demand on the server hosting the full data file, as each simulated data point will eventually generate a new web request to the service. As the number of sampling points are on the scale of thousands for moderate cases, a significant increase in the overall simulation time has been observed, large variations depending on the traffic on the service, and when introducing parallel computing, users have been banned from the service due to several pending requests from the same IP address.

Lastly, visualisation of how the weather evolves and impacts the simulation is not strictly necessary from a naval architect's point of view but can be a great feature to understand the simulation results as well as build confidence among customers. Seemingly the easiest and most common way of doing this is by using a map client such as Leaflet and use plug-ins to read pre-processed "tiles" from a remote service to visualise pressure fields, wave heights, wind strength and direction, etc. from a WMS service. This is efficient, as the client only downloads pre-processed image files and puts on top of a geographical map. Although most meteorological services offer such tiles produced by the GEOServer software, the colour themes vary, and few offers vectors showing wind and sea current speed and direction. This is inconvenient as simulations using different weather sources will have different "look and feel" and visualizing what is required will not always be possible.

Significant research effort is aimed against storing MetOcean sensor data, *Petersen et al. (2014), Smart et al. (2017)*, and navigating, processing and aggregating large weather data sets, *Fathi et al. (2022), Ricky and Rahim (2021).* However, software tailored for handling MetOcean data for use in maritime simulations, a "Metocean-Cache", seems to be needed. It should standardize the variable naming convention for easy mix-and-match of various data sources. Further it should be able to run on a cloud-based service for easy scaling of computational power as well as storage space to ensure that data can be kept indefinitely.

This paper will cover the ongoing effort to implement this service, "MetoceanCache", and describe how it relates to the GYMIR-simulator which is the first use-case.

2. MetoceanCache's place in a simulation framework

The first application of the MetoceanCache service will be the GYMIR discrete-event simulation framework for quasi-static ship simulation models. It is intended to simulate ships serving in a maritime transportation system, along realistic routes across several years of operation to uncover a ship concepts true performance as part of a larger system. This makes MetOcean data highly valuable, to emulate how the sea and weather alters along large geospatial and temporal dimensions. Also, the quasi-static nature of the simulation models adapts well to the statistical formulation of the variables in the MetOcean data, as significant wave height and average wind speeds. Gusts and rarely occurring wave conditions are not considered, as transients are left to the high-fidelity time-domain simulations.

The GYMIR simulator is built around a discrete event scheduling engine, handling various models in the simulation as agents.

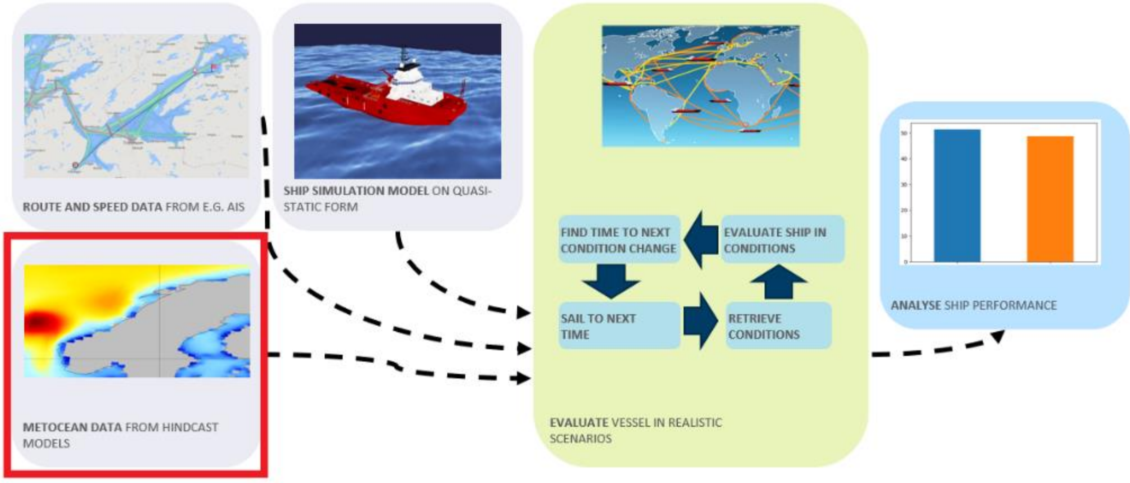

Fig.1: MetoceanCache's place in the GYMIR simulation framework, *Dæhlen et al. (2021*)

The basic idea behind the framework is that each simulation model, being an agent (weather, logistic system and ship) reports the time of its next significant change. This may be new weather data available along the temporal axis, ship heading changes or ship arrives at port. The discrete event scheduler selects among the reported instants the closest into the future and evaluates all agents at this instant. This minimizes computational effort but ensures that all agents consider significant changes.

The client-side of the MetoceanCache, marked in red in figure 1, becomes such an agent, and must be able to provide the instant when new data is available, in addition to the MetOcean data itself. The latter must be retrieved from the server-side at every sampling point, which is on the scale of several thousand for each simulation.

3. Software architecture and technology choices

Several considerations are made when drawing the architecture of the MetoceanCache, and flexibility is key to allow for exchanging the technology choice or implementation of separate components at later stages. The ability to deploy the application as a web-service in a cloud service requires a REST-controller and an API definition, and obviously strategies for data storage must be selected. Further, the solution must allow for accessing new weather sources at later stages. Figure 2 summarizes the architecture in terms of technology decisions.

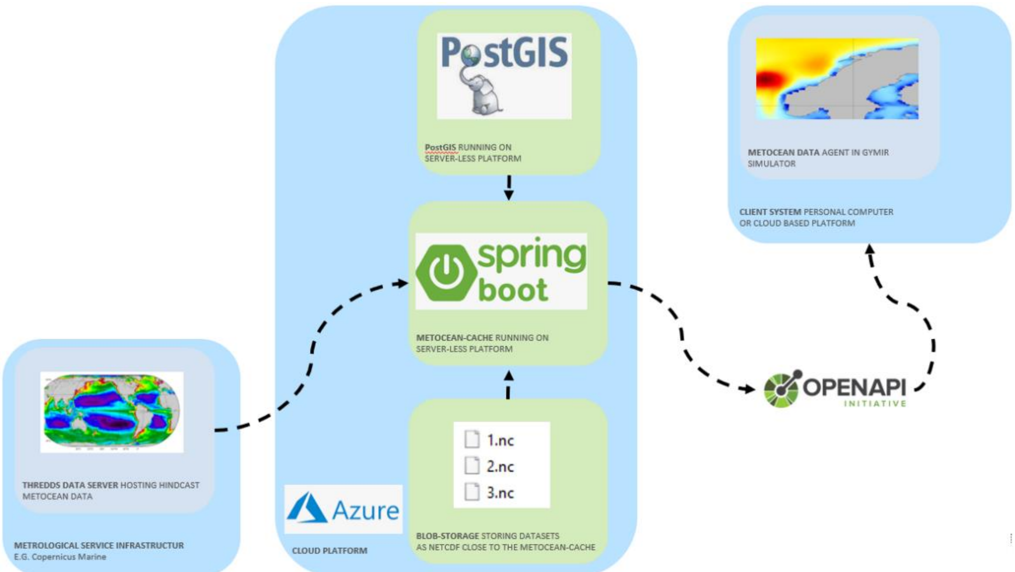

Fig.2: MetoceanCache technology architecture. Arrows shows data flow direction

3.1. REST-controller

Although python and javascript-based frameworks such as Django, Flask and Express have gained significant foothold the recent years and are well-established, the REST-controller for Metocean-Cache is built upon the Spring Boot framework. It has been around for a long time, and Java as a programming language is by many considered easier to maintain for long periods of time. Also, most simulation platform developed by SINTEF Ocean, including GYMIR and OSP is based on Java, making direct interfacing easier if that should be necessary for performance.

3.2. Storage layer

Two storage solution, each with different advantages, were chosen to work in parallel to cover different use cases:

Blob storage, acting as a network drive with fast access for the Metocean-Cache REST-controller:

- Blob storage is cheaper than PostgreSQL storage. The largest cloud suppliers charge 5-10x more per GB for data stored in PostgreSQL than data blob storage.
- Independent of weather data providers. All data is stored in the file storage, hence no external dependencies. The open access OpenDAP api's has a tendency of being slow at peak hours. This is avoided since all data is prefetched.

PostGIS database, storing individual data points as they are requested by the user(s):

- Cached data can be retrieved quickly. Retrieving populated data can be done within few milliseconds (ms), while responses from the file storages takes 50-1000 ms. This is useful for simulations reruns or optimization tasks using similar simulations multiple times.
- Large flexibility of data variables. When using a file store, all variables must be selected when the data is retrieved and stored. With a PostGIS cache the variables can be selected in runtime.

3.3. API and client access

In its basic form, Spring Boot provides an API for RESTful access. However, it requires either an extensive manual job of documentation to enable other developers to make client code to access the API, or one must simply dive into the MetoceanCache-code base. OpenAPI is a framework to specify the API in Yaml format, then providing tools to generate the client-side code in most common languages. Using the Maven build system, the Yaml file can be updated automatically from the REST-controller code, thus making it easy to create and update the client-side in case of API updates.

4. Experimental results

The MetoceanCache, available as open-source at “https://github.com/sintef-ocean/MetOcean-cache”, is installed on an Azure App Service, a platform-as-a-service for Spring Boot applications, using Premium v2 P1V2 for performance benchmarking.

The benchmark simulations are done using a simulation horizon of 1 year for the GYMIR simulator running on a personal computer, thus a network latency of approximately 30-40 ms between simulator and the MetoceanCache must be anticipated and is thus reflected in the benchmark figures. Both time spent for retrieving weather data, as well as time spent on ship simulation calculation (including weather retrieval), was measured to compare performance on using different strategies for retrieving MetOcean data. It should be noted that the traffic load on the servers accessed through OpenDAP (met.no and Copernicus Marine service) is unpredictable and has a significant variation, and this uncertainty should be kept in mind when comparing the results.

4.1. Use-case: Using blob-storage for simulation of small geographical extent

A simulation of a south-north bound route of the Trondheimsfjord is implemented to represent a typical short-sea simulation case with limited geographical extent. A high-resolution dataset for the Norwegian coastal areas from met.no (MyWaveWam800) is stored on Azure File Storage Container for the MetoceanCache to test the blob storage functionality. The dataset covers 5 years of data with a geographic and temporal resolution of 800 meters and 1 hour respectively, giving a total size of 480 GB. Using the data set, the simulation produces 12853 data points independent of the weather source.

Storing a full data set may be attractive if the user has a great deal of simulation cases in a geographic area of limited extent. The storage cost is approximately € 100 per year. It could be noted that storing similar data from Copernicus with 1/12-degree resolution of the entire world would require approximately 1TB of storage per year of data (costing €200 per year).

For weather sources using the blob storage, the average response time between the MetoceanCache service and the blob storage is 51 ms, when a file has recently been used by the server. For cases where a file is not accessed recently, the same figure is 742 ms. This indicates a cache mechanism in the file system between the MetoceanCache web application and the blob storage, however the precise workings of this is at the time of writing unknown to the authors.

Table I: Benchmarking simulations using alternative access methods to met.no MyWaveWam800 MetOcean dataset. All figures in seconds.

	Computational time (s)	Time spent retrieving weather data (s)	Average time / retrieval (ms)
Dataset stored on local hard drive	5.1	0.4	~ 0
Dataset accessed on met.no Thredds server through OpenDAP	1385.6	1364.1	106
Dataset access through MetoceanCache using Blob storage	1226.2	1211.0	94

Not surprisingly, using files stored locally on the user's computer hard drive is extremely fast compared to retrieving data from a web service. However, as the OpenDAP service of met.no has become slow in the recent months due to high traffic, a larger advantage of using MetoceanCache with blob storage was anticipated. Investigations has pointed in the direction of performance issues between the Azure File Storage Container and the MetoceanCache web service platform (which physically is a network connection on a server centre), however more investigations are needed. The File Storage Container is intended as a price-efficient way of storing large quantities of data, probably with a performance trade off. Looking into other storage options may be a path forward.

4.2. Use-case: Using PostGIS-database for simulation of large geographical extent

As a contrast, the transatlantic route from the Gulf of Mexico to the English Channel, typical for a Medium-Range Tanker, presented in previous work is used to test the MetoceanCache using PostGIS as storage layer. The downloaded data set, reduced to cover the area of interest, requires 11 GB of storage space using maximum temporal and geospatial resolution, 0.085 degrees and 3 hours respectively, storing wave height, wave direction and wave peak period variables. Although possible to store on a local harddrive, its troublesome to download as care must be taken to ensure the whole route is covered, and due to limitations in file size in the sub-setter, the data set must be split into several files along the temporal axis.

Table II: Benchmarking simulations using alternative access methods to Copernicus Marine Global Oceans Waves Analysis and Forecast MetOcean data set. All figures in seconds.

	Computational time (s)	Time spent retrieving weather data (s)	Average time / retrieval (ms)
Dataset stored on local hard drive	4.0	1.4	~ 0
Dataset accessed on met.no Thredds server through OpenDAP	702.3	697.5	228
Dataset access through MetoceanCache using PostGIS database (1st run)	3627.9	3624.9	1186
Dataset access through MetoceanCache using PostGIS database (2nd run)	128.4	125.7	41

Due to lower resolution on the data set along all axes, as well as significantly more distance between the way points of the route, the number of sampling points are reduced to 3057. This is reflected in the computational time, however due to the data downloaded to the local hard drive is compressed, the time spent retrieving data is greater.

Comparing the first and second run using the PostGIS database on the MetoceanCache, the cache-functionality works as intended as a significant performance increase is observed. However, as the cache defaults to OpenDAP in the first run when data is not yet cached, its surprising that the retrieval time is a factor of five greater than using OpenDAP directly. If this is due to slow insertion into the database, inefficient retrieval or simply large traffic on the Copernicus Marine service should be investigated. The performance during the second run is on the other hand surprisingly satisfactory as it outperforms the blob-storage in the first use case (although the data sets are not the same).

Although further investigations into the poor performance of the PostGIS storage during the first uncached run, as well as the blob storage, other requirements to the MetoceanCache such as permanent storage and access to reproduce simulations, as well as convenient access to data are addressed. Several ways forward to improve performance can be mentioned:

- Evaluating various options for improved indexing of the PostGIS database, such as Time-scaleDB.
- Evaluate other ways to store the MetOcean data in a PostGIS database, e.g., store small NetCDF or Gribfiles instead of the values themselves.
- Apply other types of databases, such as MongoDB.
- Investigate alternative cloud storage options for the blob-storage.
- Investigate alternatives to RESTful API, such as gRPC.

To further improve data retrieval efficiency, simulations can estimate the ships geospatial and temporal trajectory to retrieve data points for a longer period in a single request to minimize network latency. For simulations with uncertain routes and speeds due to varying weather data (like GYMIR), an added margin along the trajectory in all dimensions can ensure complete coverage. PostGIS database is well-suited for such requests and tailoring towards MetoceanCache's intended domain is likely to give the greatest performance benefits.

5. Conclusion

A software for retrieving, storing, and accessing MetOcean data for use in maritime simulations requires key requirements. The paper presented requirements and developed a novel open-source implementation on a cloud platform. The software was applied for providing MetOcean data to a maritime simulation platform running two use-cases for performance benchmarking.

The software proved to solve many practical shortcomings of current solutions used for MetOcean data in simulations, making it more convenient and user-friendly to retrieve such data. However, potential for performance increase was identified, and further work will investigate various options in this matter.

Acknowledgements

This study has been financially supported by and is a part of the dissemination activities for:

- Norwegian Research Council project 237917: SFI Smart Maritime
- Norwegian Research Council project 332298: EcoRouter
- EU Horizon 2020 grant number 815012: AUTOSHIP

The study has been conducted using MetOcean data from:

- E.U. Copernicus Marine Service Information (https://doi.org/10.48670/moi-00017)
- The Norwegian Metrological Institute (https://thredds.met.no/thredds/fou-hi/mywavewam800.html)

References

CARON, J. (2005), *The Thredds Data Server and NetCDF Common Data Model*, AGU Fall Meeting Abstracts, IN33B-1184

DÆHLEN, J.S.; SANDVIK, E.; JØRGENSEN, U. (2022), *Model Predictive Approach for Realistic Operational Decisions in Sea Passage Simulation*, COMPIT Conf., Pontignano, pp.201–211, http://data.hiper-conf.info/compit2022_pontignano.pdf

DÆHLEN, J.S.; SANDVIK, E.; RIALLAND, A.I.; LAGEMANN, B. (2021), *A Method for Evaluating Ship Concepts in Realistic Operational Scenarios using Agent-based Discrete-Event Simulation*, COMPIT Conf., Mülheim, pp.141–150, http://data.hiper-conf.info/compit2021_muelheim.pdf

FATHI, D.E.; GRIMSTAD, A.; JOHNSEN, T.A.; NOWAK, M.P.; STÀLHANE, M. (2013), *Integrated decision support approach for ship design*, IEEE OCEANS, Bergen, pp.1–8

FATHI, M.; HAGHI KASHANI, M.; JAMEII, S. M.; MAHDIPOUR, E. (2022), *Big data analytics in weather forecasting: A systematic review*, Archives of Comp. Methods in Eng. 29(2), pp.1247–1275

IMO (2018), *Strategy on Reduction of GHG Emissions from Ships*, Resolution MEPC.304(72), Int. Mar. Org., London

IPCC (2022), *Summary for Policymakers*, in Pörtner et al. (Eds.), Climate Change 2022: Impacts, Adaptation, and Vulnerability, Contribution of Working Group II to the 6th Assessment Report of the Intergovernmental Panel on Climate Change, Cambridge University Press

MAGARIÑO, M.E.; COFIÑO, A.; BEDIA, J.; VEGA, M.; FERNÁNDEZ, J.; MANZANAS, R.; GUTIÉRREZ, J.M. (2014), *The ECOMS user data gateway: Homogeneous seasonal-to-decadal forecast data access for end-users*, EGU General Assembly Conference Abstracts, 14992

NATIVI, S.; MAZZETTI, P.; SANTORO, M.; BOLDRINI, E.; MANZELLA, G.; SCHAAP, D. (2010), *CDI/THREDDS interoperability in the SeaDataNet framework*, Advances in Geosciences 28, pp.17–27

PAPANIKOLAOU, A.; HARRIES, S.; HOOIJMANS, P.; MARZI, J.; LE NÉNA, R.; TORBEN, S.; YRJÄNÄINEN, A.; BODEN, B. (2022), *A holistic approach to ship design: Tools and applications*, J. Ship Research 66(01), pp.25–53

PETERSEN, L.; GOMEZ, R.; HELZEL, T.; THOMAS, N. (2014), *New Data Management System for Coastal Radar WERA to Support Decision Making*, 11th Int. Conf. Hydroscience & Engineering (ICHE), Hamburg, pp.921–926

REW, R.; HARTNETT, E.; CARON, J. (2006), *NetCDF-4: Software implementing an enhanced data model for the geosciences*, 22nd Int. Conf. Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology, 6

RICKY, S.K.; RAHIM, L.A. (2021), *Metocean Prediction using Hadoop, Spark & R*, Int. Conf. Computer & Information Sciences (ICCOINS), Kuching, pp.312–315

SKJONG, S.; RINDARØY, M.; KYLLINGSTAD, L.T.; ÆSØY, V.; PEDERSEN, E. (2018), *Virtual prototyping of maritime systems and operations: Applications of distributed co-simulations*, J. Marine Science and Technology 23(4), pp.835–853

SMART, S.D.; QUINTINO, T.; RAOULT, B. (2017), *A scalable object store for meteorological and climate data*, Platform for Advanced Scientific Computing Conf., pp.1–8

SMOGELI, Ø.R.; LUDVIGSEN, K.B.; JAMT, L.; VIK, B.; NORDAHL, H.; KYLLINGSTAD, L.T.; YUM, K.K.; ZHANG, H. (2020), *Open Simulation Platform–An Open-Source Project for Maritime System Co-Simulation*, COMPIT Conf., Pontignano