Open
Geospatial
Consortium

TOWARDS A FEDERATED MARINE SDI: IHO AND OGC STANDARDS APPLIED TO MARINE PROTECTED AREA DATA ENGINEERING REPORT

——

ENGINEERING REPORT

PUBLISHED

CONTENTS

LIST OF TABLES

LIST OF FIGURES

I   EXECUTIVE SUMMARY

The Federated Marine Spatial Data Infrastructure (FMSDI) Pilot is an OGC Innovation Program initiative with the objective of enhancing Marine Spatial Data Infrastructures (MSDIs), to better understand MSDI maturity, and to demonstrate the power of FAIR (Findable, Accessible, Interoperable, Reusable) data in the context of the marine environment. It is organized in three phases. This Engineering Report is based on the work of the second phase.

Motivation and Objectives

One of the challenges of Marine Protected Area (MPA) data is to make it available for a wide variety of users, including those outside the MSDI domain, such as fishermen, resource extractors, utilities, tourists, or recreational boaters. These users, who do not have direct access to MPA databases to access the data they need to perform their activities, rely on smaller consumer-facing applications, which in turn rely on Application Programming Interfaces (APIs) to request and consume the data they work with.

The use of standards makes it easier for developers to build software applications. The more robust these standards are, the easier it is to build applications, and the more diverse the audiences that can utilize them in a variety of scenarios. Because of this, the demonstration of standards related to both MPA data and the APIs they are served through becomes of key importance.

Within this context, this pilot addressed the following research questions:

- What stages the data go through from MPA to S-122;

- What steps were taken in the server development to standardize the various data into an S-122 data set;

- What stages the data go through a fusion scenario, regarding format, metadata, etc;

- What steps were taken in the server development to synthesize the data and create digestible data for clients;

- Which OGC API standards were leveraged to perform transformations to this data;

- How the data were processed by the clients and what views were used; and

- What kind of modifications do the S-122 and OGC API standards need to better address the use of MPA data.

Technical Overview

The activities were divided into two concurrent stages or sections. The **first stage** focused on the demonstration of the transformation of MPA data into the S-122 standard and its achieved interoperability when being served through OGC APIs. The **second stage** went beyond marine protected areas and opened the examination to a broader set of data and standards. These stages saw the demonstration of seven components.

- **One Baltic/North Sea Server (D100)**: One processing server that ingested the data from various sources of the Baltic Sea / North Sea providers. These MPA data were brought into the server and transcribed into the International Hydrographic Organization (IHO) S-122 Marine Protected Area standard.

- **Two Baltic/North Sea Clients (D101 and D102)**: These client services demonstrated different viewpoints and methods for digesting the data from the server and standardized data.

- **Two Data Fusion Servers (D120 and D121)**: These servers ingested various data inputs including MPA data. D120 was implemented using the OGC API — Features standard, while D121 was implemented using the OGC API — Environmental Data Retrieval (EDR) standard.

- **Two Data Fusion Clients (D122 and D123)**: These clients ingested the outputs from the two servers and displayed the data to end users.

Lessons Learned and Recommendations

The development, testing, and demonstrations carried out throughout this Pilot provided lessons learned for all of the Pilot's participants. The following list summarizes the lessons learned and recommendations for IHO and OGC standards that resulted from the activities of this Pilot.

- <u>OGC Standards</u>

  - **Using OGC API — Features to Serve MPA Data**: The implementations of OGC API standards were found to be of great use because of their ability to retrieve data from the authoritative sources, the ease of automation by client services, the format-neutrality of such APIs, and the fine control over retrieval, which is not present in file based encodings. This ability allows for simple filters on data fields, and compound and spatial queries, as well as simpler queries against bounding boxes.

  - **Accessing MPA Data Through OGC API — Features**: Requesting all MPA features proved to be expensive from a computing perspective. Authentication and authorization was identified as not explored sufficiently.

  - **Spatial Filtering Using a Bounding Box Query**: The bbox spatial filter, which is specified in Part 1 of the OGC API — Features standard, returned unnecessary data back to clients, which can be a challenge for users operating in low connectivity environments where bandwidth is at a premium.

  - **Using Bounding Boxes to Represent Features in Denied, Degraded, Intermittent, or Limited Bandwidth (DDIL) environments**: Bounding boxes were created for some of the feature collections to reduce the complexity and size of the MPAs in order to make them suitable for DDIL environments.

  - **EDR API and Discrete Global Grid System (DGGS)**: While the EDR API has been shown in the pilot project to provide a naïve client with the tools to successfully "explore" DGGS data, any client that requests a location from a DGGS server must understand the DGGS geometry it is receiving.

- **Features API vs. EDR API**: While the Features API proved to do a well serving distinctly identifiable entities like ships, routes, lakes, and zones, among others, the EDR API provided a great way of accessing data that is not easily mapped to identifiable entities such as subsets of data about an area of interest.

- **Common Query Language (CQL) Support in Service Metadata**: The lack of information about CQL support in the API Service Metadata proved to be a downside.

- **Filtering Complex Features**: Searching Data from complex features was found to complicate the filtering processes for the clients.

- **Feature Styling in Features and EDR API**: The lack of styling support by both the Features and EDR API was not ideal.

- <u>IHO Standards</u>

  - **The S-100/S-122 Model from a Server Perspective**: The current S-122 model was found to be fairly basic in terms of its representation of MPA data. This encoding, whilst a good fit for maritime use cases, did not currently reflect the broader application of MPAs in different geospatial agencies and the richer attribution required for those uses.

    A number of proposed suggestions have been implemented in the model used for the server, specifically:

    - Added three new values to `categoryOfMarineProtectedArea`;

    - Added complex attribute to record other designations;

    - Added a dimensions complex attribute to record the calculated dimensions of the MPA;

    - Added enactment date (mandatory) and update date (optional) to all MPAs;

    - Added information representing Management Plans;

    - All Feature Types now have multiple identifiers;

    - `producerCode` was added as a simple attribute to Agency; and

    - Added regional to jurisdiction.

  - **Client Perspective of S-122**: The MPA Feature ID was not clearly 'universally persistent'. On the other hand, managing the MPA feature name as a complex type made it difficult to manage queries based on the well-known name of an MPA. Furthermore, the S-121 specification did not address the concept of marine protected area networks, preventing the modeling of the 'synergistic' properties of the MPA network and its application toward a common objective.

  - **Authority Names**: The authority is represented as a `featureName` in the S-100 model and is affected by the same naming convention issues identified in previous works. The

client application must have prior knowledge of the locale-specific authority name in order to provide this information as part of the query filter.

- **Use of Bounding Boxes by the S-122 Product Specification**: The S-122 product specification supports bounding boxes for individual features, however the part 10b Geography Markup Language (GML) encoding does not specify their use.

- GeoJSON

  - **Challenges With The GeoJSON Encoding**: The use of GeoJSON as an encoding is not part of S-100 itself. However, its ubiquity as a format for exchange of geospatial data raises the possibility of its use for modeling S-100 General Feature Model (GFM) data.

  - **Use of GeoJSON**: The GeoJSON format provides several advantages for interoperability because of wide adoption and support in mapping software. However, the return of non-standardized JSON objects made it difficult to know how to display the information to the user in a meaningful way.

  - **CovJSON vs. GeoJSON**: For an EDR service, Coverage JSON would represent the data better than GeoJSON.

Recommended Future Work

Future work should build upon the findings that emerged from the development and testing of these components, and answer questions that were left out of the scope. The following list summarizes the recommendations for future work that resulted from the activities of this Pilot.

- OGC Standards

  - **Disconnected, Degraded, Intermittent, Limited Bandwidth (DDIL) Environments**: Further investigation is required on how to optimize the retrieval and storage of MPA feature collections as a GeoPackage using a supported OGC file encoding.

  - **Further Enhancing MPA Filters**: There is a need to further develop the client application alongside a server implementing the OGC API — Features — Part 3:

Filtering candidate standard for additional filtering to reduce the amount of data requested from the server.

- **Using Vector Tiles**: Due to the many benefits that vector tiles offer, especially for users operating in a DDIL environment, future work should explore using vector tiles for MPA data.

- **Explore Potential Solutions to Challenges With DGGS**: While the D121 Fusion Server was implemented with a DGGS, the interface also shows promise for further development.

  - Beyond the core EDR query capabilities such as *query by position*, *radius*, *trajectory*, *bounding box*, etc., the DGGS service could provide a **refined query interface to allow the client to specify the tessellation level** for an area of interest.

  - Also of benefit would be **support for temporal extents** in which the client would provide both a spatial and temporal extent over which the DGGS service would provide aggregated datums.

- <u>IHO Standards</u>

  - **S-122 and ISO 19152**: More work is required to map examples and test the application of ISO 19152 for some providers.

  - **Establishing a Data Schema for DDIL Environments**: If using a DDIL twin for any data is to be considered going forward, then there needs to be some consideration for what the data schema would need to be.

- <u>GeoJSON</u>

  - **GeoJSON Elements that Remain to be Worked Out**: Information types, relationships, identifiers, other standardized metadata, and other geometries.

  - **Further Explorations on GeoJSON and S-100**: Most GeoJSON data tends to have the same attributes for each feature. So, whilst S-100 is conformant, it may pose challenges for some implementers and perhaps there are accommodations which can be made. Furthermore, the ability to seamlessly aggregate datasets together in GeoJSON would be a good step forward to working with S-100.

  - **GeoJSON and Moving Features**: For future work, application of the GeoJSON Moving Features specification would be well-positioned to represent vessel traffic.

I.A. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Table — Contacts

NAME	ORGANIZATION	ROLE
Sergio Taleisnik	Skymantics	Editor
Terry Idol, PhD.	Skymantics	Editor
Sara Saeedi, PhD.	OGC	Collaborator
Sina Taghavikish	OGC	Collaborator
Jason MacDonald	Compusult	Collaborator
Matthew Jestico	Helyx	Collaborator
Jonathan Pritchard	IIC Tech	Collaborator
Glenn Laughlin	Pelagis	Collaborator
Marta Padilla Ruiz	University of Calgary	Collaborator
Perry Peterson	University of Calgary	Collaborator

II   KEYWORDS

The following are keywords to be used by search engines and document catalogues.

Federated Marine, SDI, IHO, OGC, Spatial Data Infrastructure

III SECURITY CONSIDERATIONS

No security considerations have been made for this document.

IV SUBMITTING ORGANIZATIONS

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

- Skymantics, LLC

1

NORMATIVE REFERENCES

1 NORMATIVE REFERENCES

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IHO: IHO S-57, *IHO Transfer Standard for Digital Hydrographic Data*. International Hydrographic Organization, Monaco (2000–). https://iho.int/uploads/user/pubs/standards/s-57/31Main.pdf.

IHO: IHO S-100, *IHO Universal Hydrographic Data Model*. International Hydrographic Organization, Monaco (2018–). https://iho.int/uploads/user/pubs/standards/s-100/S-100 Ed%204.0.0 Clean 17122018.pdf.

IHO: IHO S-122, *Marine Protected Areas*. International Hydrographic Organization, Monaco (2019–). https://registry.iho.int/productspec/view.do?idx=73&product_ID=S-122&statusS=5&domainS=ALL&category=product_ID

OGC API — Processes, https://ogcapi.ogc.org/processes/

OGC API — Features, https://ogcapi.ogc.org/features/

OGC API — Records, https://ogcapi.ogc.org/records/

OGC API — Discrete Global Grid System, https://ogcapi.ogc.org/dggs/

OGC API — Environmental Data Retrieval, https://ogcapi.ogc.org/edr/

2

TERMS, DEFINITIONS AND ABBREVIATED TERMS

―――

2 TERMS, DEFINITIONS AND ABBREVIATED TERMS

This document uses the terms defined in OGC Policy Directive 49, which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word "shall" (not "must") is the verb form used to indicate a requirement to be strictly followed to conform to this document and OGC documents do not use the equivalent phrases in the ISO/IEC Directives, Part 2.

This document also uses terms defined in the OGC Standard for Modular specifications (OGC 08-131r3), also known as the 'ModSpec'. The definitions of terms such as standard, specification, requirement, and conformance test are provided in the ModSpec.

For the purposes of this document, the following additional terms and definitions apply.

2.1. DDIL

Denied, Disrupted, Intermittent, and Limited Bandwidth environments. Used to describe scenarios where the connectivity is not ideal and actions need to be taken to guarantee a normal or minimum operation of software applications.

2.2. Interoperability

Capability to communicate, execute programs, or transfer data among various functional units in a manner that allows the user to have little or no knowledge of the unique characteristics of those units [ISO 19119].

2.3. Marine Spatial Data Infrastructure (MSDI)

A specific type of Spatial Data Infrastructure (SDI) with a focus on the marine environment.

2.4. Abbreviated terms

API   Application Programming Interface

DDIL   Disconnected, Degraded, Intermittent, and Limited Bandwidth Environments

DGGS   Discrete Global Grid System

EDR   Environmental Data Retrieval

ER   Engineering Report

FAIR   Findable, Accessible, Interoperable, and Reusable

FMSDI   Federated Marine Spatial Data Infrastructure

IHO   International Hydrographic Organization

MPA   Marine Protected Area

MSDIWG   Marine Spatial Data Infrastructures Working Group

OGC   Open Geospatial Consortium

SDI   Spatial Data Infrastructure

TIE   Technology Integration Experiment

3

SUBJECT

———

3 SUBJECT

This Engineering Report (ER) summarizes the demonstrations, findings, and recommendations that emerged from the second phase of the OGC Federated Marine Spatial Data Infrastructure (FMSDI) Pilot. The goal of this initiative was to further advance the interoperability and usage of Marine Protected Area (MPA) data through the implementation of the IHO standard S-122 and several OGC API standards.

This ER describes a solution architecture consisting of a collection of interoperable components developed to demonstrate technologies that helped to achieve the objectives of this Pilot's phase. This document describes a server built to serve MPA data through an OGC API – Features endpoint and two servers that combined MPA data with additional datasets and served it through both an OGC API – Features and an OGC API — EDR endpoint. This document also describes the three clients built to consume under different scenarios the data offered by the aforementioned servers. Finally, this ER captures lessons learned and recommendations for IHO and OGC API standards, and recommendations for future work.

4

OVERVIEW

4 OVERVIEW

Due to the depth of the topic and the complexities of developing a Federated Marine Spatial Data Infrastructure, this initiative and therefore, the Engineering Report, will be ongoing, living documents beyond their initial publication. The new versions and additions will extend the development and structure of a Federated Marine Spatial Data Infrastructure in both technology and location of focus and testing.

Section 5: Towards an FMSDI (Initiative Overview)

This section explores the FMSDI Pilot, describes past initiatives, describes its problem statements and motivation, its phases, and the tasks within its phases.

Section 6: Background

This section describes the technologies and knowledge base that make up the baseline for this Pilot.

Section 7: Research Objectives and Technical Architecture

This chapter describes the motivations that guided this Pilot's work, the research objectives, and the component architecture that was demonstrated to address this Pilot's goals.

Section 8: Baltic Sea/North Sea Server

This section describes the Baltic Sea/North Sea Server. This component was designed to ingest the MPA data from various sources of the Baltic Sea / North Sea providers, transform the data to comply with the S-122 standard, and offer it through an API built using OGC API standards. This component was demonstrated by IIC Technologies.

Section 9: Baltic Sea/North Sea Client 1

This section describes the Baltic Sea/North Sea Client. This component was designed to ingest MPA data from the Baltic Sea/North Sea Server developed in this Pilot, consume it in S-122 format, and demonstrate different viewpoints and methods for digesting data from the server. This component was demonstrated by Pelagis.

Section 10: Baltic Sea/North Sea Client 2

This section describes the Baltic Sea/North Sea Client 2. This component was designed to ingest MPA data from the Baltic Sea/North Sea Server developed in this Pilot. It was designed to demonstrate an alternative viewpoint and method to the *Baltic Sea/North Sea Client 1* for digesting S-122 data. The viewpoint that this client utilized was that of a less connected service, such as that on an older vessel with limited technology or connectivity. This component was demonstrated by Helyx Secure Information Systems Ltd.

Section 11: Data Fusion Server 1

This section describes the Data Fusion Server #1. This component was designed to ingest MPA datasets as well as other datasets, combine them, and serve them through an API built using the OGC API standards. This component was demonstrated by IIC Technologies.

Section 12: Data Fusion Server 2

This section describes the Data Fusion Server #2. This component successfully published feature and coverage data sources processed into an Icosahedral Snyder Equal Area Aperture 3 Hexagonal (ISEA3H) DGGS for discovery as EDR Collections and EDR Parameter Queries. This component was demonstrated by the University of Calgary.

Section 13: Data Fused Client 1

This section describes the Data Fusion Client #1. This component was designed to ingest the various datasets (S-122 and others) served by the Data Fusion Servers developed throughout this Pilot. This component was demonstrated by Compusult.

Section 14: Data Fused Client 2

This section describes the Data Fusion Client #2. This component was designed to ingest various datasets served by the Data Fusion Servers federated with 3rd party open data services. This component was demonstrated by Pelagis.

Section 15: Technology Integration Experiments (TIEs)

The FMSDI Pilot Technology Integration Experiments (TIEs) focused on the exchange of MPA data through OGC APIs. Each TIE explored under different circumstances the potential of OGC API standards and IHO standards with the objective of relaying MPA data. This section describes each TIE and outlines its expected and actual results.

Section 16: Challenges and Recommendations for OGC Standards and IHO Standards

This section outlines a prescriptive list of challenges and lessons learned through the different stages of the initiative. This section also includes recommendations for the various standards utilized through the initiative.

Section 17: Recommendations for Future Works

This section outlines a descriptive list of various items that could be expanded upon in future initiatives or for the sponsors to utilize and build from.

5

TOWARDS AN FMSDI (INITIATIVE OVERVIEW)

5 TOWARDS AN FMSDI (INITIATIVE OVERVIEW)

The Federated Marine Spatial Data Infrastructure (FMSDI) Pilot is an OGC Innovation Program initiative with the objective to enhance Marine Spatial Data Infrastructures (MSDIs), to better understand MSDI maturity, and to demonstrate the power of FAIR (Findable, Accessible, Interoperable, and Reusable) data in the context the marine environment.

A Marine Spatial Data Infrastructure (MSDI) is a specific type of Spatial Data Infrastructure (SDI) with a focus on the marine environment. It is not only a collection of hydrographic products, but is also an infrastructure that promotes the interoperability of data at all levels (e.g., regional, national, and international). Like all SDIs, it tries to enhance the discoverability, accessibility, and interoperability of marine data. By doing so it supports a wider, non-traditional user-base of marine data far beyond what is typically used for navigation.

This initiative, which is conducted under the OGC Innovation Program, builds directly on what was accomplished earlier in the year through the Federated Marine Spatial Data Infrastructure Pilot 2021. The 2021 pilot again built on the works of prior initiatives, such as the Marine Spatial Data Infrastructure Concept Development Study, the Maritime Limits and Boundaries Pilot, and the Arctic Spatial Data Infrastructure Pilot.

5.1. Problem Statement and Motivation

Ocean and marine data are recognized as valuable resources that tend to have a high cost of acquisition. Large quantities of these data are collected and stored all over the world for a wide variety of purposes and by a variety of public and private entities. Due to its importance and value, these data should be well-managed and made as widely available to end users as possible for a variety of uses including planning, policy and decision making, marine management, scientific research, and economic activities.

The collection, protection and sharing of marine data provides enormous societal benefits. Data and information on the state and variability of the marine environment is crucial for understanding changes that may result from human activity, including the effects of human-induced climate change and ocean acidification.

Currently, Government Agencies, research institutions, and the private sector provide a considerable investment in marine monitoring and observation, data sharing and assembly, and downstream services. As a result, significant progress has been made to collect, aggregate, and make publicly available the data and information derived from monitoring and observing the Marine environment.

However, data-sharing initiatives still face common challenges in their efforts to unlock the full societal and economic potential of the wealth of marine data and observations at national, regional, or local levels. The ability to effectively share, use, and re-use geospatial information

and applications across and between governments and Non-Government Organizations (NGOs) is dependent upon having an effective SDI already in-place.

The motivations of the FMSDI Pilot include the following.

- **Demonstration** — A practical technology demonstration from global community experts showcasing federated Marine SDI for selected Land/Sea use cases. Possible examples include use cases for the Arctic, European Coastal Regions, and a southeast Asian region. The demonstration will show how using OGC, IHO, and other open standards enable the community with the ability to find, obtain, use, share, interoperate, and reuse data.

- **Impact on OGC Standards** — Lessons learned, gaps, and the need for changes to the OGC Standards Baseline will be summarized in an Engineering Report that will inform the OGC Standards Program.

- **Impact on IHO Standards** — Practical testing of relevant S-100 based IHO standards will accelerate the process for adoption and implementation of IHO standards. The resulting Engineering Report will help to inform the work of the IHO HSSC Working Group and will provide inputs to enhance the framework and its component standards.

- **Development of the Marine Spatial Data Infrastructure (MSDI) Maturity Model** — Provide a roadmap for MSDI development.

Several challenges facing Marine SDI can be identified:

- lack of an integrated policy and operational framework to facilitate rapid acceptance, qualification, ingest, and use of relevant geospatial information from a range of government, commercial providers, and citizens;

- the current focus on products supporting a single customer group;

- the inability with existing metadata approaches to quickly discover and understand which information sources are most useful in the context of a user's need;

- the inability to properly fuse and synthesize multiple data sources; and

- the need for a persistent platform to organize and manage marine information and the tools necessary for collaboration among organizations to fully utilize the variety of marine data.

5.2. Previous Marine SDI Initiatives

This initiative builds on what has been accomplished in previous initiatives: the Marine Spatial Data Infrastructure Concept Development Study; the Maritime Limits and Boundaries Pilot; and the Arctic Spatial Data Infrastructure Pilot. The Marine Spatial Data Infrastructure Concept Development Study summarized the efforts and information gathered from a Request for Information which focused on in-depth data requirements, architecture, and standards needs for a Marine Spatial Data Infrastructure. The Maritime Limits and Boundaries Pilot worked to build a

detailed implementation for testing S-121 Standard data. The Arctic Spatial Data Infrastructure Pilot aimed to utilize international standards to support a spatial data exchange focusing on the complex issues of Arctic marine space.

5.3. FMSDI Pilot Phases

The FMSDI pilot started in August 2021 and is currently planned to run until December 2022. It is organized into three phases: Phase I: Initial RFIs and Datasets Overview; Phase II: IHO and OGC standards applied to MPAs; and Phase III: Overview of the next phase.

5.3.1. Phase I: Initial RFIs and Datasets Overview

Ocean and marine data are recognized as valuable resources that tend to have a high cost of acquisition. Large quantities of these data are collected and stored all over the world for a wide variety of purposes and by a variety of public and private entities. Due to its importance and value, this data should be well-managed and made as widely available to end-users as possible for a variety of uses including planning, policy and decision making, marine management, scientific research, and economic activities.

The collection, protection, and sharing of marine data provide huge societal benefits. Data and information on the state and variability of the marine environment are crucial for understanding changes that may result from human activity, including the effects of human-induced climate change and ocean acidification.

The already completed first phase included the Marine Data Availability and Accessibility Study (MDAAS). MDAAS began with the release of a Request for Information (RFI) to help determine data availability and accessibility of Marine Protected Areas (MPA, IHO S-122) and other marine data in the North Sea and Baltic Sea. The MDAAS further helped assess interoperability, availability, and usability of data, geospatial Web services, and tools across different regions and uses of marine spatial data. MDAAS also provided identification of gaps and helped define reference use-cases and scenarios for use in future FMSDI Pilot activities.

Phase I brought together diverse stakeholders from the global marine community to assess the current state of Marine SDI. This RFI is used to gather knowledge from marine domain stakeholders and contributors. The summary of this RFI is available in Annex A and an overview of the dataset is listed in Table 1.

Table 1 — Phase I (RFI) Dataset Overview

ORGANIZATION	NOTES	LINK
HELCOM — Baltic Marine Environment Protection Commission	Reported tabular data is collected and made available via HELCOM MPA database	http://mpas.helcom.fi

ORGANIZATION	NOTES	LINK
	Spatial data on MPA areas is also available as a spatial dataset (shapefile). The spatial data can be accessed via web service	https://maps.helcom.fi/website/mapservice/?datasetID=d27df8c0-de86-4d13-a06d-35a8f50b16fa
	Metadata record for the above shapefile	http://metadata.helcom.fi/geonetwork/srv/eng/catalog.search#/metadata/d27df8c0-de86-4d13-a06d-35a8f50b16fa
	OGC Web Map Service (WMS)	https://maps.helcom.fi/arcgis/services/MADS/Biodiversity/MapServer/WMSServer?request=GetCapabilities&service=WMS
	ArcGIS REST	https://maps.helcom.fi/arcgis/rest/services/MADS/Biodiversity/MapServer/54
UK Hydrographic Office	UK Offshore Marine Protected Areas/JNCC Resource Hub	https://hub.jncc.gov.uk/assets/ade43f34-54d6-4084-b66a-64f0b4a5ef27
The Danish Agency for Culture and Palaces	Data can be downloaded through 2 locations	https://www.kulturarv.dk/ffreg/
	The data can also be accessed via web services	https://www.kulturarv.dk/ffpublic/wms/ows?service=wms&version=1.1.0&request=GetCapabilities
Danish Geodata Agency	Protected areas can be retrieved from the Danish Environmental Portal, which is a public partnership to make environmental data easily available	https://arealinformation.miljoeportal.dk/html5/index.html?viewer=distribution
	Geodata info is the national metadata portal for data discovery	www.geodata-info.dk
	Dataforsyningen — the Danish data and map supply — provides access to free public data	https://dataforsyningen.dk/
Finland Traficom	All Traficom data sets can be found from our geoportal	https://julkinen.traficom.fi/oskari/
	Calls to interfaces	https://www.traficom.fi/en/statistics-and-publications/spatial-dataset-material/calls-interfaces
	Additional resources	https://kartta.paikkatietoikkuna.fi/?lang=en

ORGANIZATION	NOTES	LINK
Lithuanian Transport Safety Administration	We use public data from national spatial data center	www.geoportal.lt
German Federal Maritime and Hydrographic Agency	The GeoSeaPortal is part of the integrated German and European MSDI network	https://www.geoseaportal.de/mapapps/?lang=en
Swedish Hydrographic Organization	Many GIS stakeholders rely on the national SDI for data discovery	www.geodata.se/geodataportalen
Flemish Hydrography	Data custodian for various relevant datasets as they are included on navigational charts. It concerns the 6 MPA's described in the Marine Spatial Plan.	https://www.geopunt.be/catalogus/webservicefolder/688b3a9c-025b-4872-b1c6-06126a821e25
Geoscience Australia	A whole-of-government data access and visualization application. Web Coverage Service (WCS) harvester compiles web services into a common framework	https://nationalmap.gov.au/
	Maritime boundaries thematic mapping applications. Internal curated datasets are made available with analysis tools	http://maps.ga.gov.au/interactive-maps/#/theme/amsis
	Seafloor thematic mapping application. Internal curated datasets are made available with analysis tools	https://portal.ga.gov.au/persona/marine
	Location Index (Loc-I) is a framework that provides a consistent way to seamlessly integrate data on people, business, and the environment. Open datasets are converted to linked data for research and development	http://www.locationindex.org/
AusSeabed	Seafloor topography GeoTIFF's available for download	https://portal.ga.gov.au/persona/marine
	AWS S3, eCat GeoNetwork	https://ecat.ga.gov.au/geonetwork/srv/eng/catalog.search#/home

5.3.2. Phase II: IHO and OGC standards applied to MPAs

The second phase, **which is what this Engineering Report is based on**, extended the MPA-focus of the first phase by digging into all the various data services and began building out an S-122 demonstration model, including the exploration of the S-100 data specifications and how other data (terrestrial, meteorological, Earth observation, etc.) can mingle to create a more holistic view of the region of focus. In addition, Phase II designed a Marine Spatial Data Infrastructure (MSDI) maturity model, to provide a roadmap for MSDI development. The maturity model is described in a separate Engineering Report.

This Pilot phase was broken into three segments of focus:

- Task 1: Developing a federation of S-122 Standard Marine Protected Area (MPA) data sets in the Baltic/North Sea area;

- Task 2: Exploring a data fidelity, mobility, and versatility of S-100 Product Specification as well as other marine standards and data; and

- Task 3: Designing a UNGGIM-IGIF (United Nations Global Geospatial Information Management-Integrated Geospatial Information Framework) derived Marine Spatial Data Infrastructure maturity model which provides a roadmap for MSDI development.

5.3.2.1. Task 1: BNS Overview (Scenarios and Architecture)

The Baltic/North Sea use case looked at utilizing numerous Marine Protected Area data and related data to identify Marine Protected Areas within the Baltic/North Sea. To accomplish this, a federation of MPA data was created from the various countries that have an interest in the Baltic/North Sea region.

5.3.2.2. Task 2: Fusion Overview (Scenarios and Architecture)

The second stage identified, examined, and expanded upon existing data sets to give them greater fidelity, mobility, and versatility. This went beyond marine protected areas and opened the examination to a broader set of data and standards. These included other data sets and standards that could be utilized to develop a firmer more holistic view of a region.

5.3.2.3. Task 3: IGIF-MSDI Maturity Roadmap Overview

UKHO has developed an IGIF-MSDI Maturity Framework with international contributions highlighting that an MSDI is a continual journey and not an end-state of technological sophistication. It asserts that nations are sovereign in what type of MSDI they genuinely require for their national needs, not by an externally imposed level of technological exploitation and concomitant expenditure (unless deliberately otherwise chosen). The foundation of this initiative is the UN IGIF Nine Pathways model that lays out the strategic vision and rationale for an All-Domain geospatial infrastructure (air, land, sea, and space), which equitably benefits all socioeconomic stakeholders and sectors of a nation.

The IGIF-MSDI Maturity Framework was developed in conjunction with the UN, IHO, OGC, and World Bank with representation from Denmark (DGA), Singapore (MPA), and the United States (NOAA). As a compact and accessible document its intent is to provide a more quantitative and prescriptive "Quick Start" or "Stepping Stone" for nations beginning their IGIF-aligned MSDI implementation plans. It seeks to supplement, not supplant, any existing resource within this area and is intended to be read alongside referenced publications from the UN, IHO, OGC and World Bank. One of the major outputs from its application is the quantitative baseline rating that a nation (or national agency) can use for self-improvement towards a defined future end-state (not for regional comparisons).

The World Bank, with its many decades of financial expertise in global development programs, is an indispensable partner for realizing this vision. The involvement of the World Bank was crucial

in providing answers to the Financing question (incl. costed business cases), alongside Why (UN), What (IHO), and How (OGC). The current World Bank 91-question SDI Diagnostic Toolkit with its Terrestrial Heritage is being augmented with IHO and OGC insights to maximize its benefits to the Marine community, whilst being aligned with the UN IGIF principles (and UN SDGs as a result). These insights from the Hydrographic and Standards communities are modular additions to a condensed subset of the current SDI Diagnostic Toolkit, which itself is an overlap of the World Bank's "Decision-Maker" and "End-User" question subsets. This ensures that the IGIF-MSDI Maturity Framework is fully interoperable with the World Bank's full IGIF implementation methodology, of which the SDI Diagnostic toolkit is only the first step.

5.3.3. Phase III: Overview of the next phase

Phase III, which will start with an additional Call for Participation in Summer 2022, will primarily extend the use cases developed in Phase II and add the Arctic region as a new location to the demonstration scenarios.
The Arctic Regional Marine Spatial Data Infrastructure Data Working Group (ARMSDIWG) of the International Hydrographic Organization (IHO) identifies and assesses the statuses of individual MSDI implementations and considers MSDI policies in related international projects and cooperates specifically with the spatial data infrastructure for the Arctic. Among other tasks, the working group analyzes how maritime authorities can contribute their spatial information and updates so information can easily be collated with other data to a current overall picture of the Arctic region. Through association with the OGC Marine DWG the ARMSDIWG monitors the development of relevant and applicable OGC standards and activities in the context of marine data services for the Arctic.
The Federated Marine Spatial Data Infrastructure 2022 Pilot will focus on several aspects that contribute to an overarching scenario that helps better understand both the challenges and potential opportunities for coastal communities, ecosystems, and economic activities in the Arctic region. Potential activities may include:

- demonstrating interoperability between land and marine data that is necessary to understand coastal erosion (e.g., ocean currents, geology, permafrost characteristics, etc.);

    - coastal erosion over time, which includes a temporal component (possible study area: Alaska, Canada, Greenland, and Iceland);

    - defining coastline (highest line) and coastal transition zone (intertidal zone);

- effects of climate change and a changing Arctic environment on wildlife migration corridors: land-sea ice-island (caribou) and sea (marine mammals);

- demonstrating the role of OGC standards to support the measurement of impacts of coastal erosion (e.g., infrastructure, food safety, traditional activities, wildlife migration, etc.) on coastal areas in the context of a changing Arctic;

- mapping of coastal sensitivity to climate change and the impacts on local communities;

- investigating the role of vector tiles and style sheets across the land-sea interface; and

- a sea-based health and safety scenario incorporating the land/sea interface in the Arctic. This scenario would demonstrate the technology and data used with OGC, IHO, and other

community standards in response to a grounding event and the evacuation of a cruise ship or research vessel in the Arctic.

6

BACKGROUND

6 BACKGROUND

This section describes the technologies and knowledge base that make up the baseline for this Pilot.

6.1. Marine Protected Areas (MPAs)

A Marine Protected Area (MPA) is a protected area whose boundaries include an area of the ocean. They include areas of the intertidal or sub-tidal terrain together with their overlying water and associated flora, fauna, historical, and cultural features which have been reserved by law or other effective means to protect part or all of the enclosed environment. For example, MPAs may be established to protect fish species, rare habitat areas, or entire ecosystems.

MPAs can range from simple declarations to protect a resource to areas that are extensively regulated. The degree to which environmental regulations affect shipping varies according to whether MPAs are located in territorial waters, exclusive economic zones, or high seas. These limits are regulated by the Law of the Sea. Most MPAs are located in the territorial waters of coastal states where enforcement can be ensured. MPAs can also be established in a state's exclusive economic zone or within international waters. [2]

6.2. S-122 Standard

The S-122 Product Specification is intended to encode Marine Protected Area (MPA) information for use in Electronic Chart Display and Information Systems (ECDIS) and other information systems. MPAs are protected areas of seas, oceans, estuaries, or large lakes. Marine Protected Area information may be considered supplementary additional information that compliments the S-101 ENC. [3]

The S-122 product is based on the S-100 General Feature Model (GFM) and is a feature-based vector product. Figure 1 shows how the S-122 application schema is realized from the S-100 GFM. [2]
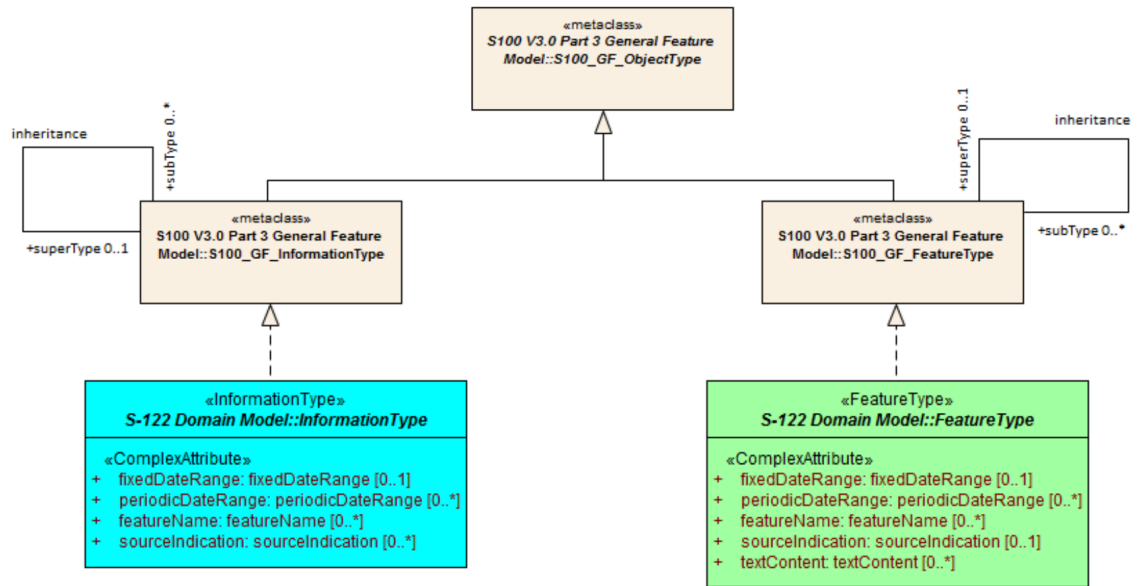
Figure 1 — Realizations from the S-100 General Feature Model

The S-100 Standard is a framework document that is intended for the development of digital products and services for hydrographic, maritime, and GIS communities. It comprises multiple parts that are based on the geospatial standards developed by the International Organization for Standardization, Technical Committee 211 (ISO/TC211). [4]

6.3. OGC APIs

For several years OGC members have worked on developing a family of Web API standards for the various geospatial resource types. These APIs are defined using the OpenAPI specification. As the OGC API Standards are developed, approved by the OGC, and then implemented by the community, the marine community can subsequently experiment and implement them as well.

The following OGC API Standards and Draft Specifications were used for the development of APIs during this Pilot.

OGC API – Features: a multi-part standard that defines the capability to create, modify, and query vector feature data on the Web and specifies requirements and recommendations for APIs that want to follow a standard way of accessing and sharing feature data. It currently consists of four parts.

- OGC API — Features — Part 1: Core. Approved September 2019, this standard defines discovery and query operations. [5]

- OGC API — Features — Part 2: Coordinate Reference Systems by Reference. Approved October 2020, extends the core capabilities specified in Part 1: Core with the ability to use coordinate reference system identifiers other than the defaults defined in the core. [6]

- Draft OGC API — Features — Part 3: Filtering. Part 3 specifies an extension to the OGC API — Features — Part 1: Core standard that defines the behavior of a server that supports enhanced filtering capabilities. [7]

- Draft OGC API — Features — Part 4: Create, Replace, Update and Delete. Part 4 specifies an extension that defines the behavior of a server that supports operations to add, replace, modify or delete individual resources from a collection. [8]

A specification for Version 2 of the Common Query Language (CQL2) is being developed together with Part 3 to standardize a language that is recommended for filter expressions. [9]

OGC API — Environmental Data Retrieval: The Environmental Data Retrieval (EDR) Application Programming Interface (API) provides a family of lightweight query interfaces to access spatiotemporal data resources by requesting data at a Position, within an Area, along a Trajectory or through a Corridor. A spatiotemporal data resource is a collection of spatiotemporal data that can be sampled using the EDR query pattern geometries. These patterns are detailed in the section describing the Core Requirements Class.

The goals of the EDR API are to make it easier to access a wide range of data through a uniform, well-defined simple Web interface and to achieve data reduction to only the data needed by the user or client while hiding much of the data storage complexity. A major use case for the EDR API is to retrieve small subsets from large collections of environmental data, such as weather forecasts, though many other types of data can be accessed. The important aspect is that the data can be unambiguously specified by spatiotemporal coordinates. [10]

Draft OGC API – Styles: This draft API specifies building blocks for OGC Web APIs that enable map servers and clients as well as visual style editors to manage and fetch styles. [1]

RESEARCH OBJECTIVES AND TECHNICAL ARCHITECTURE

7 RESEARCH OBJECTIVES AND TECHNICAL ARCHITECTURE

As seen in Clause 5 this Pilot was the second phase of a broader initiative. This chapter describes the motivations that guided this Pilot's work and the component architecture that was developed to address this Pilot's goals.

7.1. Problem Statement

One of the challenges of Marine Protected Area (MPA) data is to make it available for a wide variety of users, including those outside the MSDI domain, such as fishermen, resource extractors, emergency services, utilities, tourists, or recreational boaters. These users, who may not have direct access to MPA databases to access the data they need to perform their activities, rely on smaller consumer-facing applications, which in turn rely on APIs to request and consume the data they work with.

The use of standards makes it easier for developers to build software applications. The more robust these standards are, the easier it is to build applications and the more diverse the audiences that are able to utilize them in a variety of scenarios. Because of this, the demonstration of standards related to both MPA data and the APIs they are served through becomes of key importance.

Within this context, this pilot addressed the following research questions:

- What stages the data goes through from MPA to S-122;

- What steps were taken in the server development to standardize the various data into an S-122 data set;

- What stages the data goes through in a fusion scenario, regarding format, metadata, etc.;

- What steps were taken in the server development to synthesize the data and create digestible data for clients;

- Which OGC APIs were leveraged to perform transformations to this data;

- How the data were processed by the clients and what views were used; and

- What kind of modifications do the S-122 and OGC API standards need to better address the use of MPA data.

7.2. Technical Overview

The activities were divided into two concurrent stages or sections, each one with its own technical architecture.

7.2.1. First Stage

The first stage (Figure 2) focused on the demonstration of the transformation of MPA data into the S-122 standard, and its achieved interoperability when being served through OGC APIs. This stage demonstrated the access to MPA data through APIs built based on OGC API standards in order to test the ability of OGC API standards to build APIs that make MPA data more Findable, Accessible, Interoperable, and Reusable (FAIR) for a community greater than just the traditional domain experts. By demonstrating the use of the IHO S-122 MPA data standard, it also demonstrated its combined use with APIs based on OGC API standards.

This stage saw the demonstration of three components.

- **One Baltic/North Sea Server (D100)**: One processing server that ingested the data from various sources of the Baltic Sea / North Sea providers. These MPA data were brought into the server and transcribed into the IHO S-122 Marine Protected Area standard.

- **Two Baltic/North Sea Clients (D101 and D102)**: These client services demonstrated different viewpoints and methods for digesting the data from the server and standardized data, including:

  - A well-connected, online service that one may use to analyze the scenario from afar in an office or other remote location;

  - A less-connected (DDIL: Denied, Disrupted, Intermittent, and Low Bandwidth) service for use on the Baltic / North Sea on an older vessel that may have limited technology; and

  - An entity on a newer vessel with more recent technology that could be actively committing additional data to the input data.
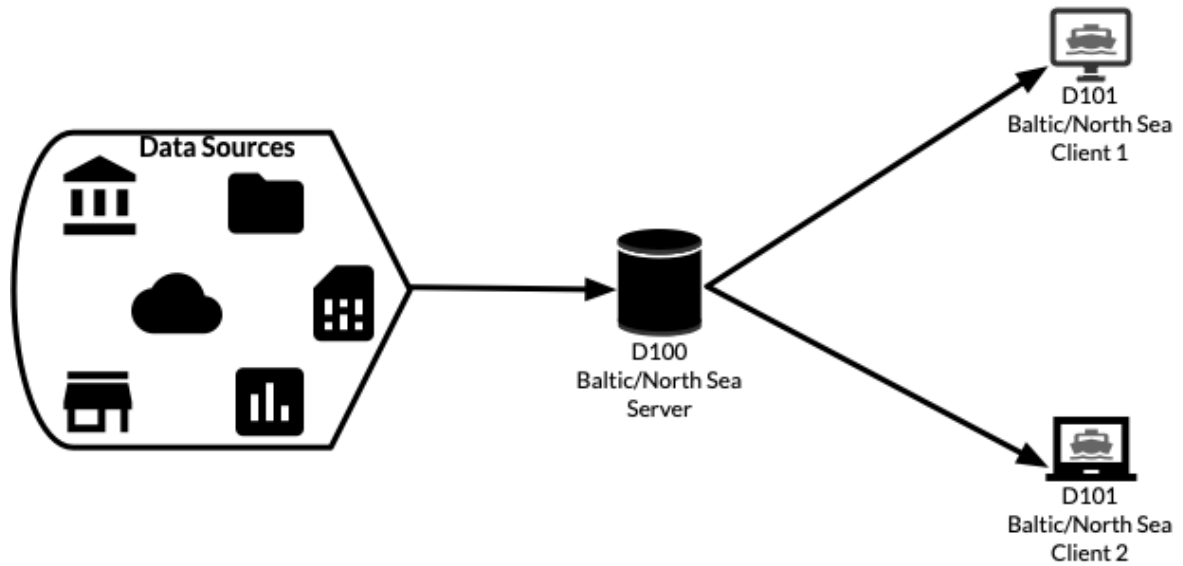
Figure 2 — First Stage Architecture

MPA datasets might also need to be combined with datasets from other domains for the purpose of providing a more comprehensive overview of a region. Scenarios that might have an impact on land and water would require the consumption of a combination of hydrographic, terrestrial, and meteorological data. Examples of these activities include construction, disaster response, and multimodal transportation. The second segment addressed this data combination challenge, demonstrating the use of OGC API standards and IHO standards in combination with other datasets and standards.

7.2.2. Second Stage

The second stage (Figure 3) identified, examined, and expanded upon existing data sets to give them greater fidelity, mobility, and versatility. This went beyond marine protected areas and opened the examination to a broader set of data and standards. These included other data sets and standards that could be utilized to develop a firmer, more holistic view of a region, such as terrestrial data, meteorological data, earth observation data, etc.
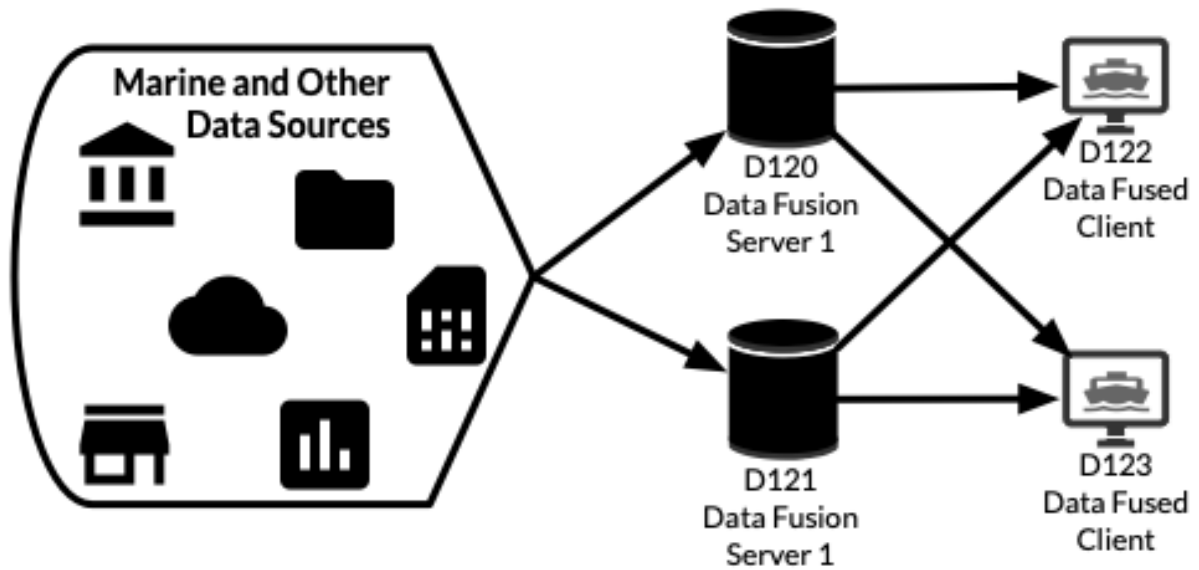
Figure 3 — Second Stage Architecture

This stage saw the demonstration of four components.

- **Two Data Fusion Servers (D120 and D121)**: These servers ingested various data inputs, including MPA data. D120 was implemented using the OGC API — Features standard, while D121 was implemented using the OGC API — EDR standard.

- **Two Data Fusion Clients (D122 and D123)**: These clients ingested the outputs from the two servers and displayed the data to end users.

The following seven chapters, Chapters 8 through 14, describe all these components individually followed by Chapter 15 which describes the interactions between them. Each component chapter includes a description of the baseline from which that component was demonstrated, the technical architecture of that component, and the challenges and lessons learned that emerged from the demonstration of that component.

The final two chapters, Chapters 16 and 17, present the main lessons learned and recommendations, and suggested future work, respectively.

8

BALTIC SEA/NORTH SEA SERVER

8 BALTIC SEA/NORTH SEA SERVER

The Baltic Sea/North Sea Server component was designed to ingest MPA data from various sources of the Baltic Sea / North Sea providers, transform the data to comply with the S-122 standard, and offer it through an API built using OGC API standards. This component was demonstrated by IIC Technologies.

8.1. Status Quo

IIC has previous experience designing and building proof of concept (POC) access points for data using open source tools and OGC Standards. Most platforms use combinations of open source geospatial databases (PostgreSQL/PostGIS) and middleware (most notably GeoServer) implementing existing OGC W*S standards. Although the concept of such access points is nothing new, there are a number of gaps in implementation and the project offered opportunities to advance the work further.

The IHO S-100 framework is approaching a level of maturity where it is ready for formalized adoption within the IMO SOLAS framework for marine charts. In addition, the potential of S-100, as an ISO implementation of model-driven marine geospatial data is enormous. Using the framework, a broad array of marine phenomena can be modeled and commissioned as standards conformant datasets with known structures and metadata. The maturity of S-100 has made it attractive to the MSDI community as a potential vehicle for broader marine geospatial data, hence the commissioning of the OGC FMSDI project to explore the subject in more detail. As well as exploring implementation/adaptation of the S-100 framework, the other area of research is the implementation of API based standards. There is an existing Part within S-100 dealing with data streaming but this is more of a conceptual framework around which to build specific web services than a specific methodology. In addition, the navigational community is in the process of development of an API-based structure for `eNavigation` on SOLAS vessels, under IEC63135, the SECOM standard for secure marine communications. This standard, developed in parallel with S-100, will offer a way of transferring file-based S-100 data to vessel navigation systems. There is currently no standardized way of transferring S-100 data via API to broader stakeholders, nor is there an understanding of the potential impact this would have on the traditional Hydrographic Office (HO) community.

In addition to these constraints S-100 currently has no formalized encodings specific to modern web services. Although a GML implementation is included in S-100 Part 10b, this is now somewhat dated (and has been updated for S-100 edition 5.0.0) and a draft GeoJSON encoding has been developed for use in other projects. This has proved useful for implementation of OGC API Features tools as most are designed for GeoJSON distribution natively. This in no way precludes distribution via GML (the current encoding supported by S-122) but is useful for GIS integration, web services, and API processing by bespoke scripts. The project provides a useful opportunity to refine the GeoJSON encoding, the eventual aim being to publish it as a GFM/OGC "bridging" specification between the two organizations. Many institutions currently hold marine data in a variety of database structures and formats. Interchange between agencies

(nationally, regionally, and internationally) is not regulated (in contrast to vessel navigation) and tends to be implemented according to national priorities. Therefore, datasets received as part of the project's Phase 1, a survey of S-122/MPA implementations, used a variety of formats and content models. It is hoped that the project can inform such infrastructure on the content, exchange, and transformation of such datasets to positively benefit stakeholders.

The main objectives explored in Phase 1 were to look in detail at the possibilities offered by S-100 within the domain of Marine Protected Areas (MSP). Currently, many agencies promulgate data on MPAs and aggregate other data sources into their own. This results in a great deal of duplication and a lack of "custodianship" across the region.

The main questions/issues explored by IIC were as follows.

- How can S-100, specifically S-122, help? By providing a common model across the domain of MPAs which can be implemented by all and used to create better interoperability across the region and across boundaries.

- S-122 creation from different agencies' data must be possible. So, all agencies' requirements must be accounted for (this implies some filtering of data which is being released for access).

- Interoperability between the different data sources must be achieved using a harmonized data model conforming to the S-100 modeling conventions and, preferably, within a revision of S-122 which is geared towards use by a broader class of agencies.

- What needs to be done to the S-122 model to try and make it work "better" for use cases other than navigation?

- What difference does access via API bring? This is really important for hydrographic offices who are used to distributing data via hard media. Most have "download" services but this means that data gets out of date and authenticity is lost.

- How does the use of OGC API Standards "help"? (Does it help?)

- Transboundary, Regional and multi-partner aspects are a high priority.

There is a need to reconcile data with existing limits and boundaries. This could be done with S-121 but boundaries can also be delimited by the S-122 data. It is also important to bring in WEND100 as a governance mechanism which ties distribution to EEZ limits. Furthermore, data integrity and traceability are always big issues for producers.

8.2. Technical Architecture

The server component converts data by mapping individual fields from the native format in which it was received to the fields defined by the S-122 feature catalogue. This process was achieved by mapping the fields individually and then constructing bespoke pipelines for the actual transformations.

The server uses an S-100 database with a simple schema implementing the framework. Features do not have their own tables; the database schema stores features and attributes together with the feature catalogue mediating attribute type, multiplicity, bindings, and relationships.

The S-122 form of the data is then stored in files which are served by the OGC API Features implementation built using pygeoapi, a framework-based tool not specific to any one dataset. Bespoke pipelines were constructed using IIC's API for the S-100 framework which simplifies the construction of features/attributes and sub-attributes and their storage in the database

This integration into the database also then enables manual GUI editing of the S-122 data via the IIC Feature Builder tool (Figure 4). Once the data is prepared, edited, and ready it can be exported into the correct format ready for deployment with pygeoapi.
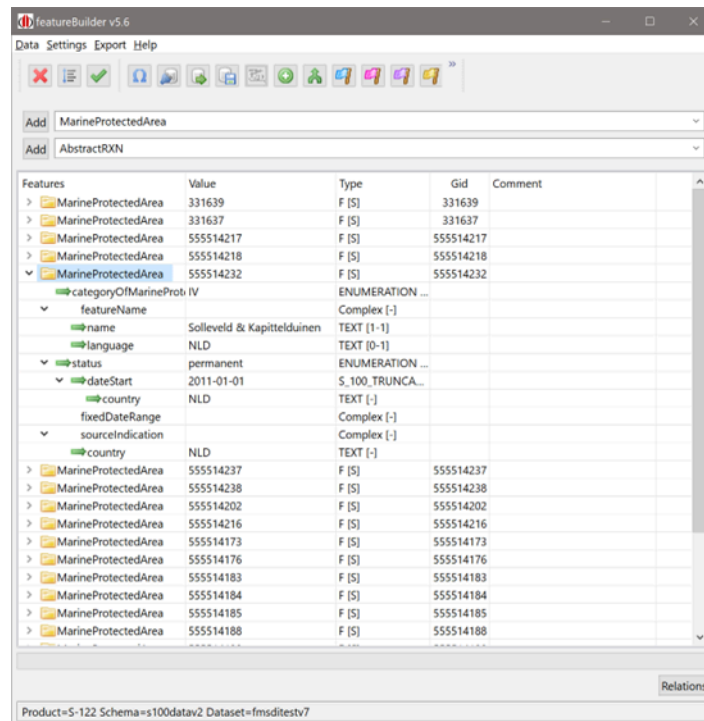
Figure 4 — IIC Feature Builder Tool

8.2.1. Data Transformation into S-122

Data was received in a variety of forms, mainly shapefile and OGC WFS. The format was first reduced to PostGIS database tables using OGR tools. A standardized interface to the database was then used to construct bespoke data processing pipelines to create the S-122 data. As IIC have an open database implementing S-100, this was a fairly rapid process and the GeoJSON outputs conforming to the S-122 feature catalogue were simple to produce. Pygeoapi was customized in a standard way to show bounding boxes, contact details, and other dataset metadata.

The transformation of data from the bespoke form to the S-122 standardized form was generally done using bespoke pipelines, as described. The server was a framework-level component requiring no customization specifically for S-122 data. The pipelines were run locally with data deployed to the S-122 server. The server had a customized HTML skin and was implemented

with both flat GeoJSON content as well as content using `Eleasticsearch` as a service provider – providing more comprehensive filtering and search capabilities.

8.2.2. Usage of OGC APIs

OGC APIs were not used to effect transformations to the data as the pipelines required bespoke processing for all the supplied data. Parts 1 and 3 of OGC API Features were used for distribution of the S-122 data only.

The servers implemented were API endpoints capable of distributing data in a variety of forms and with a selection of content drawn from both the existing IHO S-122 data model and an enhanced model drafted during the project. The Baltic Server collected endpoints using the basic model of S-122, as published by the IHO, and using data received at project outset collected from sponsors and questionnaire respondents by OGC.

The capabilities of the server extended to a full, conformant implementation of OGC API Features Part 1 and 3 together with some facilities for filtering and querying. Filtering was performed via individual simple attributes on the data and querying was bounding box only as per the current pygeoapi implementation.

Distribution of data was accomplished via an S-122 GeoJSON encoding developed prior to the project and enhanced during its execution. Although at an early stage, this encoding bridged many gaps between IHO S-100's default encodings (ISO8211, GML and HDF5) and those more commonly used in popular OGC implementations (OGC API commonly implements GeoJSON and HTML). An example of such an encoding is shown below in Clause 8.2.3.

8.2.3. Restricted Bandwidth Collections

One of the client implementations was to examine the possibility of data provision in areas with reduced bandwidth. Examination of the data has shown that the vast majority of the size of the datasets is for the representation of detailed coordinates since MPA data is often cut to coastlines, requiring large numbers of vertices to accurately represent the shoreline boundary. Conversely, the seaward boundaries are more normally delimited or tied to regulated positions and simple polygons only. Therefore, a separate API endpoint implemented a bounding box alternative to the full data. The code below shows one of these bounding box endpoints. The identifiers of the bounding box features are identical to those in the full data, allowing the client to establish potential intersections with areas of interest and then only request full data with authoritative positions once the initial list has been pre-filtered.

```json
{
    "type": "Feature",
    "id": "DEU:83:11837:27",
    "properties": {
        "foid": "DEU:83:11837:27",
        "featureName": {
            "name": "Niederschsisches Wattenmeer"
        },
        "categoryOfMarineProtectedArea": "IUCN Category II",
        "status": "permanent"
    },
    "geometry": {
```

```
        "type": "Polygon",
        "coordinates": [
            [ [ 6.580833209, 53.368864047 ],[6.580833209, 53.872514458],[8.
    558466819,53.872514458],[8.558466819, 53.368864047],[6.580833209,53.368864047
    ]]]
        }
    }
```

8.3. Challenges and Lessons Learned

Please also refer to the challenges and recommendations section of the chapter D120 Data Fusion Component.

8.3.1. API distribution of S-122 data

Many technical issues with the API distribution of S-122 data are related to the high vertex density of some of the polygons in the datasets. These are mainly in two areas.

1.   Polygons with a shoreline component, although which shoreline is rarely, if ever, documented. These introduce a large number of vertices into polygons which make downloading and viewing challenging from a performance perspective (Figure 5).

2.   Polygons with a large number of multiple components. These can also be challenging to download/retrieve because of their complex nature (Figure 6).
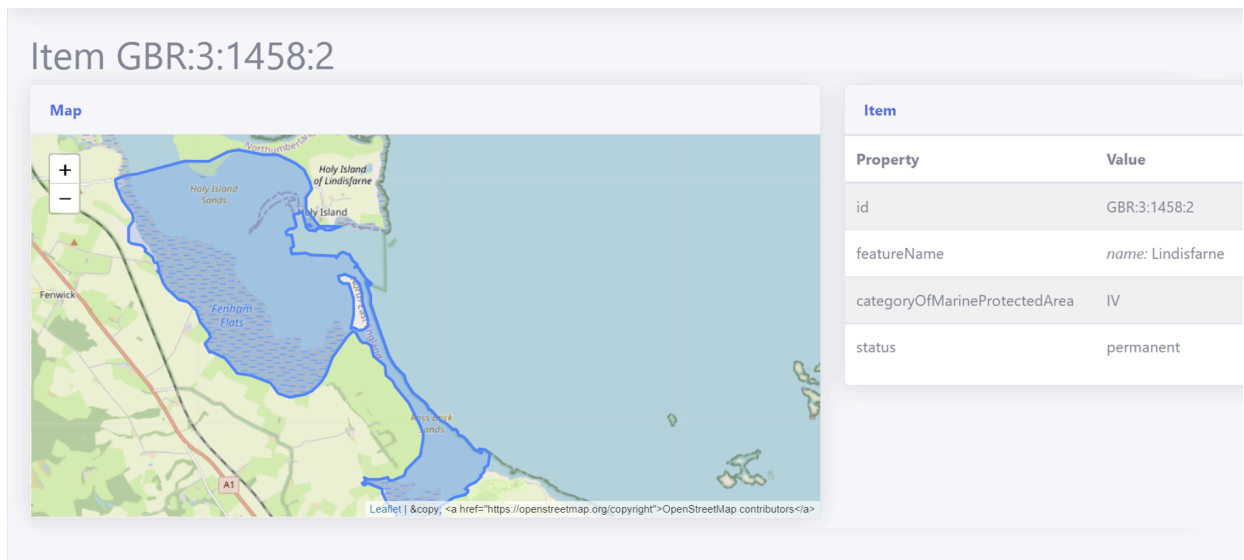
Figure 5 — Polygons With a Shoreline Component
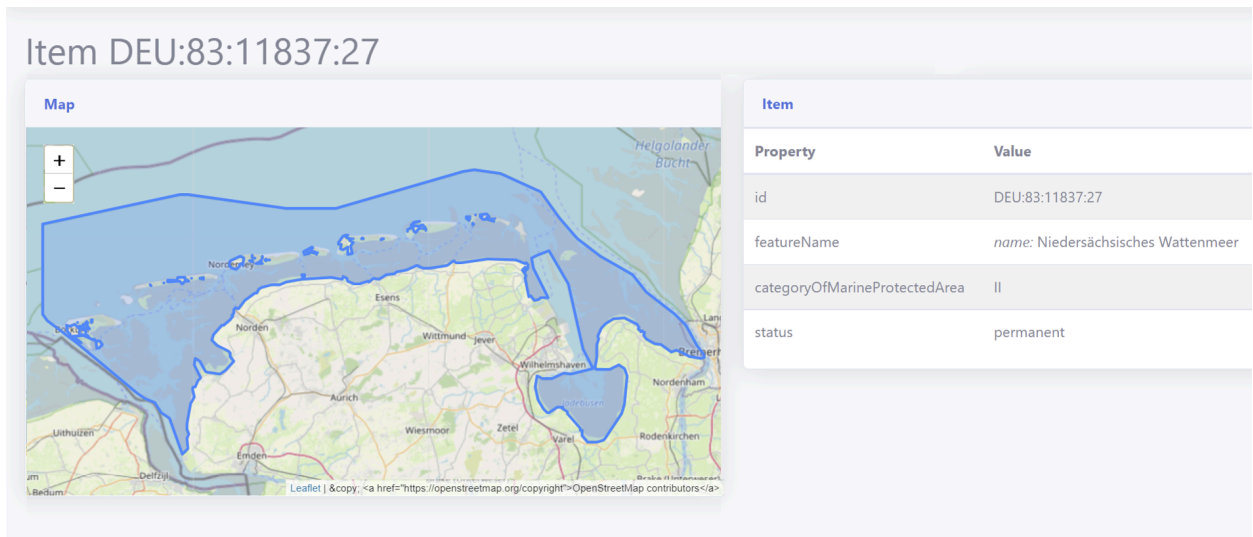
Figure 6 — Polygons With a Large Number of Multiple Components

A possible remedy for this situation would be to partition features into multiple sections based on a grid of variable resolution. One of the participants in the project used a DGGS representation, and partitioning the MPAs according to such grids alleviated the issue of having too many vertices. The client was then left with the task of piecing polygons together composed of multiple parts. This was a good solution for performance and one which is used in other contexts within the IHO. The following guidelines could be useful as a recommendation.

- Partitioning to a grid is fairly simple to do and can be done by the middleware components. It requires a suitable grid for the region in question and pre-processing of features to ensure good performance.

- It is likely some areas will be complete grid cells, i.e. all MPA, and others which are partial. So, a concept of "coverage" within a grid cell is important. Many IHO product specifications have a concept of "data coverage" which accounts for this in the feature model.

- A distinction should be made in the returned features of boundaries introduced by the grid partitioning and those which are part of the actual data. In the context of marine geo-regulation and MLB these could be "construction lines" and designated as such. There is no provision in the data model as yet for such things although product specifications like S-121 have already made such recommendations.

8.3.2. Use of GeoJSON

The implementation of the OGC API Features service in the project relied upon an encoding of data in GeoJSON, replacing the existing GML implementation which is currently part of S-100 edition 4.0.0. The GML implementation has been substantially revised for edition 5.0.0 of S-100 since previous implementations suffered from mismatches between GML and feature catalogue

structures. In S-100 the feature catalogue represents the single definition of the data structure of a product specification and binds entities drawn together from the IHO geospatial registry.

Product development is currently advanced for many product specifications and the IHOs Nautical Information Provision Working Group (NIPWG) oversees the creation of many of these. All product specifications, even gridded data, maintain a feature catalogue. Those with specific symbolization requirements also maintain an S-100 specific portrayal catalogue. Each product specification also contains a default encoding, normally drawn from the three included in S-100 Part 10 (a, b or c). These are:

1. Part 10a – ISO8211;

2. Part 10b – Geographic Markup Language (GML); and

3. Part 10c – HDF5.

The use of GeoJSON as an encoding is not currently part of S-100 itself. However, its ubiquity as a format for exchange of geospatial data raises the possibility of its use for modeling S-100 General Feature Model (GFM) data.

In order to progress this useful addition and to form a bridge between the OGC API family of standards with the S-100 GFM, a draft GeoJSON model has been developed and implemented during the project. This is based around the following principles.

1. Each S-100 GFM Feature is a named and identified GeoJSON Feature. The feature names used are those defined in the feature catalogue.

2. Similarly, all attribute names (simple or complex) are identical with those in the feature catalogue.

3. Both simple and complex attributes are GeoJSON properties.

4. Simple attributes are rendered as strings, although some simple types could be implemented as GeoJSON native types (String, Integer, Real, Boolean). String representation simplifies the initial encoding.

5. Where the feature catalogue multiplicities have cardinality > 1 and more than one is encoded, they are encoded as an array. Singular instances are encoded as a singular property.

6. Geometry is encoded without inline topology using Point, LineString, and Polygon primitives. Other GeoJSON geometry primitives can also be used ("Multi") if required. Most features do not use *Z coordinates* (depth) preferring to attribute depth but this could be implemented as a *z coordinate*.

The following elements of the GeoJSON encoding remain to be worked out. Some attempts at definition have been done but a consultation period with interested parties is probably required before a first draft of an encoding is published for testing.

- **Information Types** GeoJSON and its clients are often not good at dealing with features which have no geometry. This needs to be explored and resolved. Currently this is done

by optionally leaving out the information types or by including them inline with whatever features refer to them. This is wasteful on space, however, for the client. A separate download of JSON renderings of information types could also be accomplished if an inter-dataset referencing system can be accomplished.

- **Relationships** A systematic way of associating features together needs to be established in the encoding. This could use either standard JSON encodings or inline expansion of linked features. Relationships are absolutely key to S-100 GFM and an integral part of data. As with identifiers, each encoding in S-100 implements its own relationship methodology (ISO8211 uses LNAM and GML uses gml ids and XML references. So, GeoJSON can do the same. S-100 lacks (at a framework level) a way of relating features from different product specifications together. This could be achieved through Maritime Resource Name (MRN) identifiers though.

- **Identifiers** Each encoding implements its own identifiers. Also most product specifications will attribute identifiers, most likely MRNs. ISO8211 uses FOID while GML uses gml:id. A standardized identifier for features in the encodings needs to be defined, probably as simple as an "id" property or feature attribute – tbd. We have kept this broad as a string but formatting with the product specification may be better, e.g., "S-122:UKNCMPA020". The "links" fields have been used in the current implementation which seems to work, although links with the same key value need to be accounted for.

- **Other standardized metadata** The feature type should be included as a standard field in each feature. There may be others but testing will show those up. The Feature Collection could also contain some metadata as well. S-100 embeds more information in the dataset metadata so it would be good to be able to encode some of this somewhere in the feature collection. Although currently done by option, each feature should probably carry its product specification, as well, perhaps as a field in the id.

- **Geometry** It is possible, of course, to encode any valid GeoJSON geometry, but a stronger relationship to S-100 Part 7 is probably a good addition and would make it clearer. As topoJSON becomes more accepted, an extension of this encoding to include a formal topology would be extremely valuable. This should more well profile the topology structure encapsulated in the current ISO8211 Part 10a.

An example of a feature encoded in GeoJSON is shown in Figure 7. A sample source code is shown in the code below.

Item 1:UKNCMPA020

Map

Leaflet | © OpenStreetMap contributors

Item

Property	Value
id	1:UKNCMPA020
featureType	MarineProtectedArea
dateEnacted	23-07-2014
foid	GB:1:UKNCMPA020:2
featureName	*name:* Central Fladen
categoryOfMarineProtectedArea	Not Applicable
designation	*identifier:* UKNCMPA020 *designationValue:* NCMPA *jurisdiction:* National *designationValue:* 555560480 *jurisdiction:* International
status	permanent
dimensions	*valueOfDimension:* 92500.0 *categoryOfDimension:* Area *unitOfMeasure:* ha

Figure 7 — HTML Representation (pygeoapi) - GeoJSON Rendering on Right

```
{
    "type": "Feature",
    "id": "1:UKNCMPA020",
    "properties": {
        "featureType": "MarineProtectedArea",
        "dateEnacted": "23-07-2014",
        "foid": "GB:1:UKNCMPA020:2",
        "featureName": {
            "name": "Central Fladen"
        },
        "categoryOfMarineProtectedArea": "Not Applicable",
        "designation": [
            {
                "identifier": "UKNCMPA020",
                "designationValue": "NCMPA",
                "jurisdiction": "National"
            },
            {
                "designationValue": "555560480",
                "jurisdiction": "International"
            }
        ],
```

```
            "status": "permanent",
            "dimensions": {
                "valueOfDimension": "92500.0",
                "categoryOfDimension": "Area",
                "unitOfMeasure": "ha"
            }
        },
        "geometry": {
            "type": "Polygon",
            "coordinates": [
            ]
        }
    }
}
```

Feature Encoded in GeoJSON

This potentially extends to many product specifications. There are two aspects which need exploring in more detail.

- **The use of the IHO feature catalogue and an analogous structure for GeoJSON data** The JSON Schema is probably the way to validate data against a schema, and mapping from feature catalogue (FC) to JSON Schema for each product spec is probably achievable, but difficult. The key observation here is that each feature in S-100 normally is different than with the FC describing the range of possible attributes and values they can have. Most GeoJSON data tends to have the same attributes for each feature, essentially a tabular structure. So, whilst our encoding is conformant, it may pose challenges for some implementers and maybe there are accommodations which can be made. The API should probably return the FC / JSON Schema in its conformance classes.

- **Aggregation** S-100 aggregates features into datasets and datasets into exchange sets with appropriate placement of metadata. OGC API Features only have one level of aggregation, that of items into collections. A robust way of recursively aggregating would be a good step forward for OGC and provides an analogous structure for "exchange sets." The ability to seamlessly aggregate datasets together in GeoJSON would also be a good step forward. As noted earlier, if MRN is implemented and used to refer to features outside a feature aggregation, then inter-product relationships would be implemented.

9

BALTIC SEA/NORTH SEA CLIENT 1

9 BALTIC SEA/NORTH SEA CLIENT 1

The Baltic Sea/North Sea Client component was designed to ingest the MPA data from the Baltic Sea/North Sea Server developed in this Pilot, consume it in S-122 format, and demonstrate different viewpoints and methods for digesting the data from the server. This component was demonstrated by Pelagis.

9.1. Status Quo

Pelagis is an ocean-tech organization based in Nova Scotia, Canada. Their approach to address the needs of the marine environment, and the shared use of our ocean resources generally, is to make Marine Spatial Planning a core foundation on which to build out vertical applications. The platform architecture is based on a strongly typed federated information model represented as a unified social graph. This provides a decentralized approach towards designing various data pipelines each represented by their well-known and/or standardized model.

Pelagis' previous work has focused primarily around the coastal marine environment and in particular, has worked closely with various agencies and non-profits supporting the growth of sustainable ocean farming. This is the first experience for Pelagis as a member of OGC and its involvement with IHO standards.

Pelagis provides an opinionated framework that purposely abstracts a microservice architecture behind a strongly typed information graph model. Each service endpoint represents a feature domain encapsulating the business requirements and behavior specific to its domain. This separation of concerns provides the means to implement each service endpoint independently while hiding the complexity of the implementation from the client environment. In this way, client applications may be developed based on a unified information model hosted as an open, spatially-enabled application service (OpenSEA) while domain-specific endpoints are developed by separate teams. This federated graph model is represented as a centralized information schema accessible through a non-procedural query language representing 'what' information a client is requesting while delegating the 'how' to access the information to the platforms query optimizer.

9.2. Technical Architecture

The D100 Baltic/North Sea (BNS) service endpoint provides Marine Protected Area features and related data to identify Marine Protected Areas within the Baltic/North Sea. The service is designed to ingest various sources of MPA feature data provisioned through various agencies. The BNS service is implemented according to the OGC API — Features Standard and represents feature data compliant with the S-122 information model.

The D101 client issues queries against the BNS service endpoint conforming to OGC API — Features over http(s) to retrieve a feature collection of MPA features. The client environment demonstrates various viewpoints designed to stress the S-122 information model and the capabilities of the OGC Features API as a recommended API for managing the MPA feature domain.
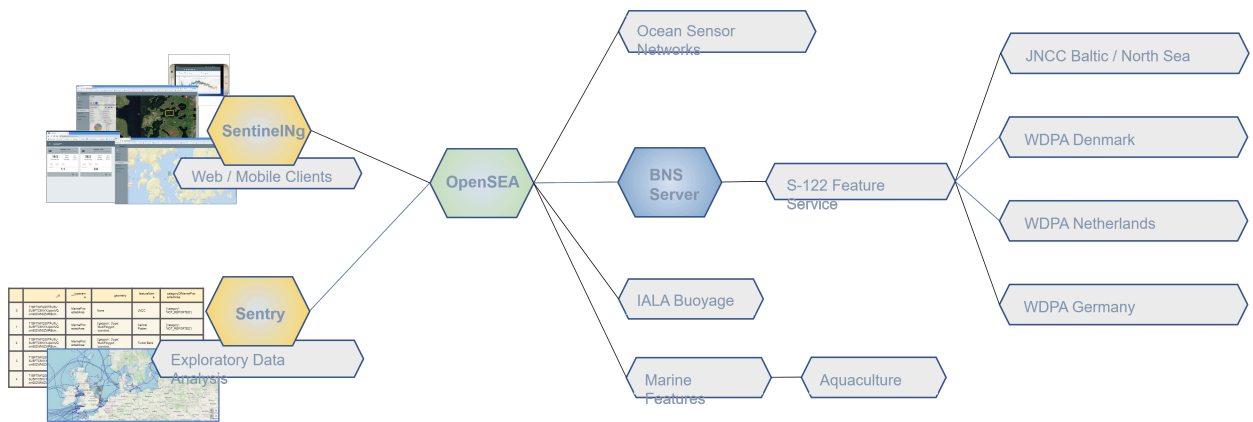
Figure 8 — D101 Client & Application Server Architecture

The framework seen in Figure 8 incorporates the requirements of the FMSDI project by extending the OpenSEA architecture to define the S-122 feature model for marine protected features. The core design of the OpenSEA application service aligns closely with our requirements to support a general marine spatial data infrastructure and in this case, directly supports our initiatives in providing a federated view of marine protected features.

The main components of the application architecture are the end-user client libraries, the OpenSEA application service, and each of the individual domain service endpoints.

Client Library

The client library is a declarative state management library responsible for interfacing between the end-user application and the OpenSEA service endpoint. The library is implementation-dependent based on the development framework of the application. The SentinelNg library is a Typescript library for web development providing a reactive-style design pattern. The Sentry client library is a Python library useful for analytics and exploratory data analysis at scale. Each of these libraries provides the core capabilities to query and modify application features through a local cache which supports both real-time and offline navigation of the domain feature model.

OpenSEA Application Service

The OpenSEA application service implements a unified graph model for each feature domain. A query issued to the service from the client library will validate against the feature domain schema and build the appropriate query plan to access each of the backend data services.

For the purpose of this project, the OpenSEA Application Service represents the client environment and is responsible for directly interfacing between the D100 Baltic/North Sea service, the D120 Data Fusion Server, and the D121 Data Fusion Server.

Data Service

The Baltic/North Sea (BNS) service publishes MPA features in accordance with the OGC API — Features Standard. This approach is based on an adapter design pattern in which each individual MPA feature source is published according to the S-122 information model. This provides a consistent interface to the MPA feature model independent of how the source system represents the MPA feature model.

9.2.1. Scenarios

Use Case: As a user, I want to see all marine protected areas

This scenario satisfies the basic requirement to query for all marine protected areas specific to a source authority and published through the D100 BNS feature service. The information model was consistent with the S-122 standard allowing the client applications to access individual MPA feature properties and geometry. The goal behind this scenario was to stress the core S-122 feature model as it aligned with the OGC set of standards.
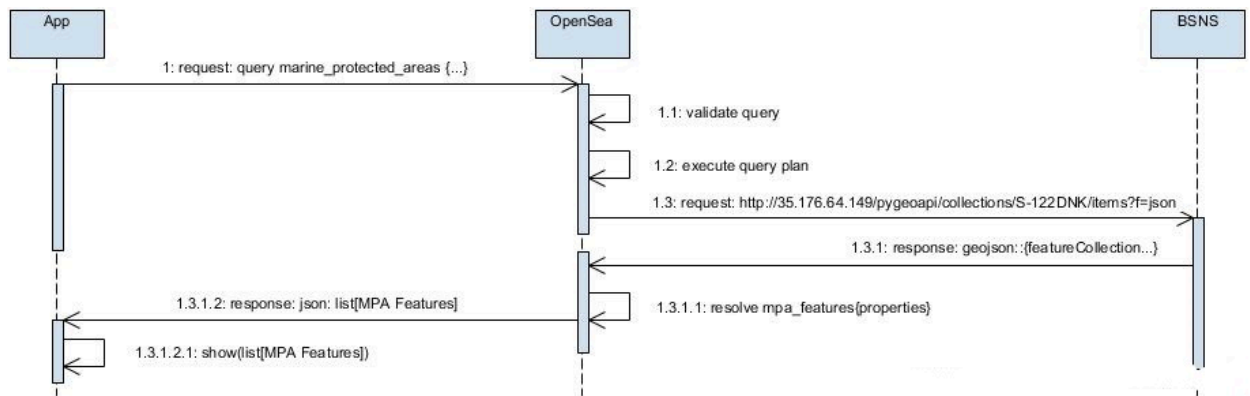
Figure 9 — D101 Client Workflow

Client query:

```
post( URI='https://…/ogcfmsdi/&#8221;, json='query all_MPAs ($authcode:
 AuthorityCode) { marine_protected_areas (authCode: $authcode) { _id
geometry{geojson} featureName categoryOfMarineProtectedArea { category } }}
```

Result:
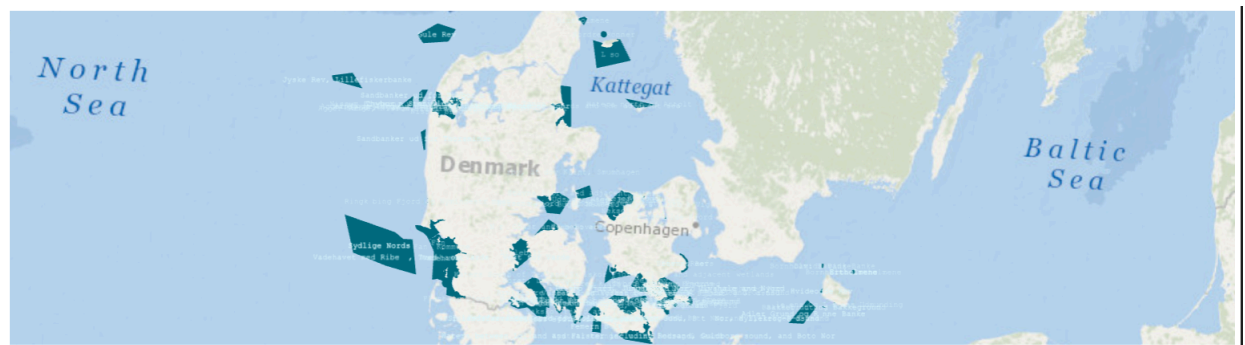
Figure 10 — D101 Client Query Result

9.3. Challenges and Lessons Learned

9.3.1. Related to OGC API — Features

- **FAIR**: The interface conforming to the OGC API – Features Standard provided the set of MPA features based on the OpenAPI specification. The key benefit of this approach is its adherence to the FAIR principles prescribed by OGC.

- **Collections & Source Authorities:** The BNS service endpoint exposes the MPA features through the /collections endpoint. Each /collection represents a set of features provided through a specific authority. This requires the client library to iterate over all collection endpoints to query for all MPA features. This could be an expensive operation requiring possible deduplication of MPA features and a complex security model to enforce against each authority. This implies the client has prior knowledge of how the BNS service endpoint manages its access plans to each authority.

  To partially address the previous issue, the client may filter the candidate selection of feature collections based on the spatial extent of each collection. Only collections with a spatial extent overlapping the area of interest would be used in the subsequent queries.

  Ideally, the BNS service would 'obfuscate' the agency-specific collections by providing one service endpoint that may filter sets of MPA features accordingly. This may have a side effect, however, in terms of the area specific metadata for each MPA feature such as the local coordinate system used for the MPA geometry.

- **Security:** The issue of security (authentication and authorization) has not been explored to its full extent. Although the client may authenticate to the BNS service to allow its usage, there does not appear to be a formal means to restrict the authorization of the client to use a particular service capability or collections endpoint. This latter case implies that all MPA features are accessible to the client regardless of the authentication and authorization requirements of the source authority. This may represent a serious issue as each source provider may have its own legal standards and compliance requirements.

- **Feature Cache:** The client application is designed to store the set of feature collections locally in a feature cache. This cache is used to look up features based the reuse of the query by the application. Given the disconnected session, additional queries relative to the feature collection may only be used if cached locally. No support for spatial query filters is provided when accessing the features in the local cache which may impact performance when viewing the features in a map component.

- **GeoPackage support:** Further investigation is required on how to optimize the retrieval and storage of MPA feature collections as a GeoPackage using a supported OGC file protocol.

9.3.2. Related to MPA Specifically

- **MPA Feature ID**: As per the S-122 specification, the feature identifier is a 'text' string composed of the feature subfields. It is assumed that this text string is stored as a UTF-8 character string. It is not clear whether this identifier is 'universally persistent', meaning that it will never change for the lifetime of the MPA feature which may, itself, be updated. For example, if the 'agency' property of the MPA feature is updated, would this affect the unique identifier of the feature which may be used as a reference from other features of interest. One consideration would be to assign a byte-encoded Globally Unique Identifier (GUID) for each MPA feature regardless of agency and subdivision to ensure that feature identifiers remain immutable. Alternatively, the IALA Maritime Resource Name (MRN) registry may also be a good candidate for assignment of unique resource names (urn) to S-100 marine features such as marine protected areas.

- **MPA Feature Name**: Managing the MPA feature name as a complex type makes it very difficult to manage queries based on the well-known name of a marine protected area. To search for a specific MPA feature by name, the client needs to manage the language code associated with the name and assume that the server will search for all MPA features based on indexing into each feature name. This may be an expensive operation if no specialized indices are configured against the set of MPA feature names.

  The OGC & W3C Spatial Data on the Web working group defines Location Information as a means of identifying the location of a thing. The location may be identified by a set of position coordinates or a well-known named place. As identified in the previous issue, the format of the MPA `featureName` will make it very difficult to comply with these W3C best practices to locate an MPA feature by its well-known name.

  For consideration, each MPA feature should have one and only one 'well known' feature name possibly modeled as an IALA MRN and managed as a UTF-8 character string. The system should then allow a client to request the MPA feature by this globally unique name in order to locate the feature.

  The manner in which feature names are managed appears heavily influenced toward the cartographic portrayal of the feature with accommodation for national languages. This influence should be managed outside the scope of the S-122 core feature model possibly in a separate 'Portrayal' package configured based on the ISO language codes. In this manner, an application may request the cartographic representation for a specific language by setting the application locale based on the ISO 639-3 language code and the server would then provide a MPA `displayName` translated to that locale for rendering purposes.

- **Authority Names**: The authority is represented as a `featureName` in the S-100 model and is affected by the same naming convention issues identified in previous works. The client application must have prior knowledge of the locale-specific authority name in order to provide this information as part of the query filter. This approach could benefit from the previous suggestion(s) to use the IALA MRN naming convention to identify each source authority of MPA features.

- **Marine protected area networks**: The IHO S-121 specification does not address the concept of marine protected area networks. Although the specification does loosely relate

MPA features based on the respective source authority, there is no capability to model the 'synergistic' properties of the MPA network and its application toward a common objective.

10

BALTIC SEA/NORTH SEA CLIENT 2

10 BALTIC SEA/NORTH SEA CLIENT 2

The Baltic Sea/North Sea Client 2 (D102) component was designed to ingest the MPA data from the Baltic Sea/North Sea Server (D100) developed in this Pilot. It was designed to demonstrate an alternative viewpoint and method to the D101 client for digesting the S-122 data. The viewpoint that D102 utilized in this pilot was that of a less connected service, such as that on an older vessel with limited technology or connectivity. This component was demonstrated by Helyx Secure Information Systems Ltd.

10.1. Status Quo

Low connected environments, often described as DDIL, are a concern for many communities operating across various sectors. The maritime domain certainly has a significant DDIL component, with vessels frequently operating in less-than-ideal network conditions. Vessels can have poor signal quality, limited bandwidth, and intermittent connectivity, all of which impact the ability to send and receive data. Whilst the technology continues to improve, there is a need to both understand and consider the range of technology currently equipped by the increasingly large user base of maritime data, and not to assume that all user needs are the same.

There are two overarching considerations when working in a less connected environment; the first concerns the various approaches that could be used to mitigate some or most of the effects in operating in a DDIL environment. For example, pre-loading or pre-caching the data, storing the data locally and updating via physical means, making the best use of networks, or using compression techniques to transmit the data. The second, and often overlooked, consideration concerns the data itself that is being transmitted to and from users operating in a less-connected environment. There remains a question of whether all users require the original data all the time, or if some derivative or simplified version would be sufficient for some users. This simplified version of the data would be much smaller in file size and could therefore be easier for vessels to receive whilst at sea. By having an alternative version of the data, it also allows the data to be potentially consumed and used by a wider user base that might not have been able to use the data in its original form. The simplified version of the data is envisaged to be a supplement to the original data rather than a replacement.

As this stage of the FMSDI pilot focuses on demonstrating improved access to MPA data for a broader variety of end users outside of the traditional MSDI domain, the D102 client primarily deals with this second consideration concerning the data. Previous work addresses approaches to compensate for DDIL environments, including networks, GeoPackages, and data compression techniques. Furthermore, as the MPA data access is widened to other maritime users, now is a good time to consider possible alternatives for how the data could be presented and delivered to an increasingly wide range of users that may not have optimal connectivity or onboard technology.

MPA features can vary in shape, extent, and complexity, with some features containing hundreds of vertices. These more complex features are inherently larger in file size than features with fewer vertices and will therefore take longer to transmit to a user operating in a DDIL

environment. MPA features located in the littoral or inland regions are especially complex in their shape as they follow the coastline or inland waterways.

10.2. Technical Architecture

10.2.1. Approach

In the pilot scenario, it was envisaged that a user aboard a vessel in a low connectivity environment would need to know the location of any MPAs upon a given route. For example, changing the planned route mid-journey in reaction to an emergency event or to respond to a natural disaster. This would require the user onboard the vessel to request data for a geographic area that was not necessarily pre-loaded onto the vessels system. It was assumed that having a small buffer on the vessels route would also be required so that MPAs within a known distance of the route would be returned from the server. For the D102 client an arbitrary buffer of 5 nautical miles was added to the vessels route.

The approach taken in this pilot was to **minimize the volume of data requested** and to **determine whether using an alternative to the original data** would be suitable and appropriate for the S-122 MPA standard. This was achieved in two ways.

- Utilizing bounding boxes of the MPA features, rather than the MPA features themselves, to reduce the size of the data requested from the D100 server.

- By using a spatial filter to only retrieve data from the D100 server that is located within a bounding box of the vessels route (buffered by 5 NM).

10.2.2. Architecture and Interactions
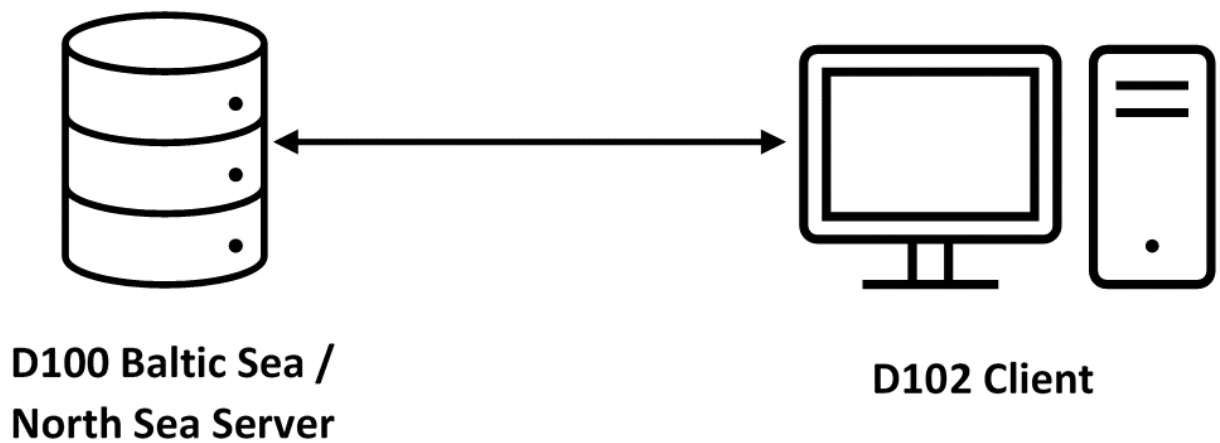
Figure 11 — BSNS Client #2 Architecture Overview

A lightweight client was built that connected to the D100 server (Figure 11). The D102 client was developed using Leaflet, a JavaScript library for interactive maps, and was chosen for its ease of use, the ability for it to be used on desktop and mobile devices, and its ability to be extended using plugins. Two Leaflet plugins were also used for the client.

- Leaflet Draw: This was employed to implement the draw feature in the client, enabling the user to draw their route.

- Turf.js: This enables geoprocessing to be done in the client, such as buffering features, performing intersections, etc.
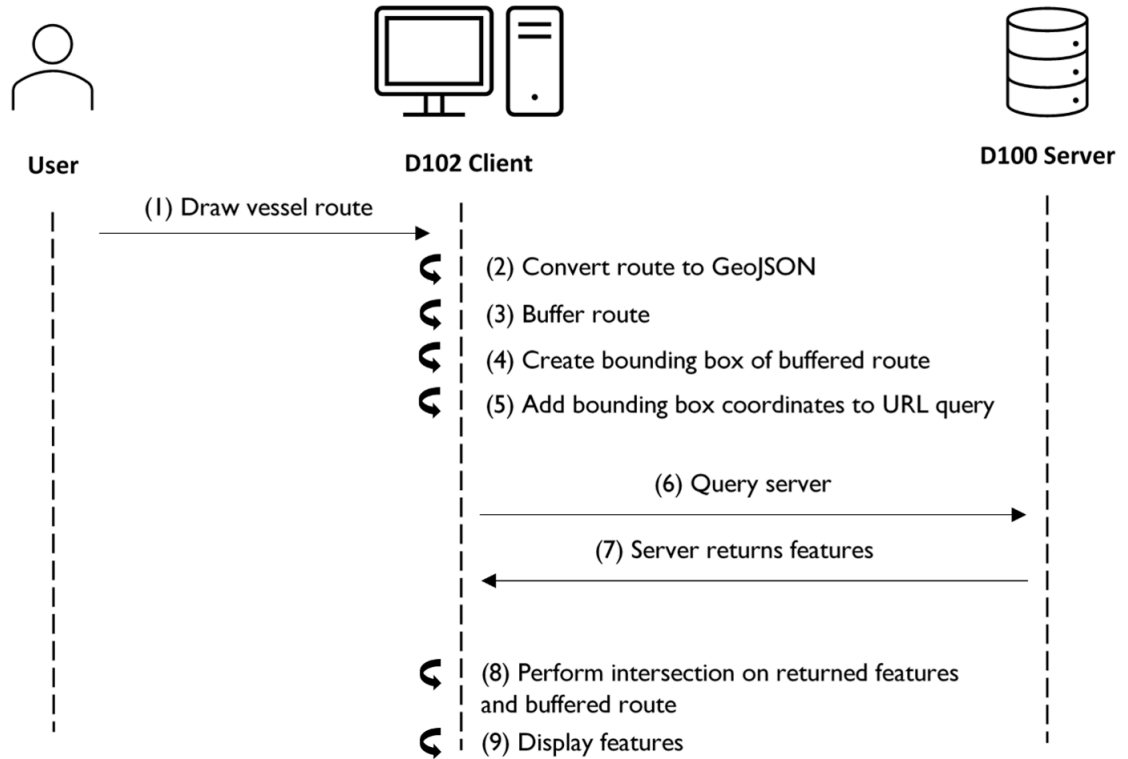
Figure 12 — Workflow for the BSNS Client

The sequence of interactions that take place is detailed below and shown in Figure 12.

1 - User creates a route of the vessel by drawing it on the map in the client. This uses the Leaflet Draw plugin functionality.
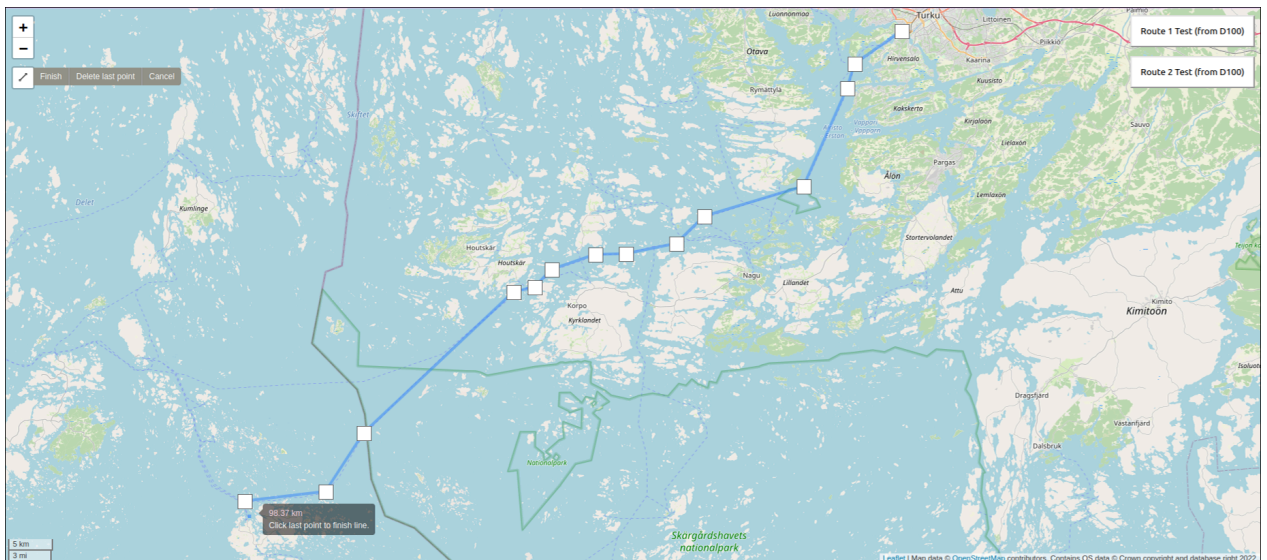
Vessel Route Drawn in the Client Application

2 - The route drawn by the user is then converted in the client. First, the route is converted into a GeoJSON object and logged in the console. Then the GeoJSON object is converted into a Turf `Linestring` object using the `turf.lineString` method from the *Turf.js* plugin, as seen on the code below. This conversion into a Turf `Linestring` is required to perform the buffer in Step 3.

```
The drawn route: {"type":"Feature","properties":{},"geometry":{"type":
"LineString","coordinates":[[22.212982,60.43622], [22.118225,60.40368],[22.
103119,60.379254],[22.015228,60.281366],[21.814728,60.251397],[21.758423,60.
224129],[21.656799,60.213898], [21.595001,60.213216],[21.507111,60.198203],[21.
472778,60.180453],[21.430206,60.175672],[21.128082,60.033988],[21.051178,59.
974944], [20.887756,59.965322],[20.89325,59.950885]]}}
```

3 - The Turf `Linestring` object created in Step 2 is then buffered in the client using the `turf.buffer` method from the *Turf.js* plugin. This is currently set at an arbitrary distance of 9.26km (5NM) for demonstration purposes. The buffer is not displayed in the client, but the coordinates of the buffer are shown in the console log.

```
The route buffer coordinates are logged below as an array-->
▼ Array [ (89) […] ]
   ▼ 0: Array(89) [ (2) […], (2) […], (2) […], … ]
      ▶ 0: Array [ 21.001566660340927, 59.88819868820486 ]
      ▶ 1: Array [ 21.070370573024128, 59.89222423788814 ]
      ▶ 2: Array [ 21.10583479989879, 59.89629761170469 ]
      ▶ 3: Array [ 21.138692659176336, 59.90413940764619 ]
      ▶ 4: Array [ 21.167374363731863, 59.91537555233287 ]
```

Route Buffer Coordinates as Shown in the Console Log

4 - The geographical extent of the buffered route is calculated in the client using Leaflet's `.getBounds()` and `.toBboxString()` methods. This returns the bottom-left and top-right coordinates of the bounding box that encloses the buffered route and outputs them as a string X1, Y1, X2, Y2, as seen on the code below. This is used as the bbox spatial filter when querying the D100 server and is added to the console log.

```
Bounding Box of the Buffered Line: 20.72139348936155,59.86761112867299,22.
381746235013203,60.519483021624964
```

5 - The coordinates of the buffered route bounding box are added to the URL query as a bbox spatial filter, as seen on the code below. The URL query is added to the console log as a hyperlink.

```
The URL query of the feature collection is: http://35.176.64.149/pygeoapi/
coliections/S-122WOPA_ES_BB/it_61112867299,22.381746235013203,60.
519483021624964&limit=2000
```

6 - D102 client then requests the MPA features from the D100 server using the query created in Step 5. Note that the map extent of the feature collection on the D100 server is confined to the bounding box created in Step 4.

MPA Features Retrieved From the D100 Server

7 - D100 server then returns the MPA features back to the D102 client that are located within the bbox spatial filter. The number of features returned is added to the console log.

FeatureCollection Object Returned Back to the Client

8 - The MPA features returned in Step 7 are then processed in the client using the Turf.js plugin. The MPA features are converted into Turf multi-polygons using the `turf.multipolygon` method and the buffered route is converted into a turf polygon using the `turf.polygon` method. An intersection between these two is then performed in the client using the `turf.intersect` method.
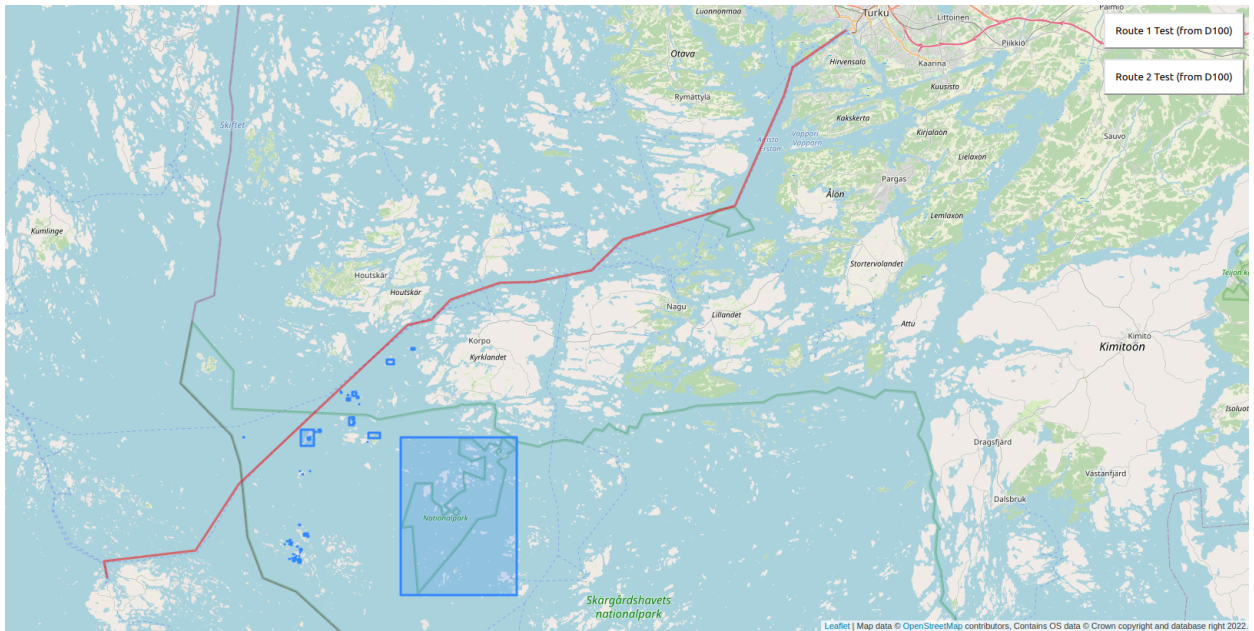
MPA Features Displayed in the Client

9 — The D102 client then only displays the MPA features that intersect the route buffer. The number of MPA features that are displayed in the client is added to the console log. Note in this example there were 544 MPA features returned from the D100 server and only the 65 features that intersected the buffered route were displayed in the client.

```
There are 65 bounding boxes of Marine Protected Areas within 5NM of the
vessels route
```

Two of the feature collections (WDPA source and JNCC source) on the D100 server were provided in two different formats; one had the original MPA features, and the other had bounding boxes that represented the original MPA features. Of the two collections, the WDPA contained 2000 features (Figure 13) whilst the JNCC collection only contained 25 features. Figure 14 and Figure 15 contain an overview of the differences between querying the original data and the bounding boxes when using the two test routes hosted on the D100 server. **It was evident that using a simplified version of the MPA data can significantly reduce the size of the returned query.**
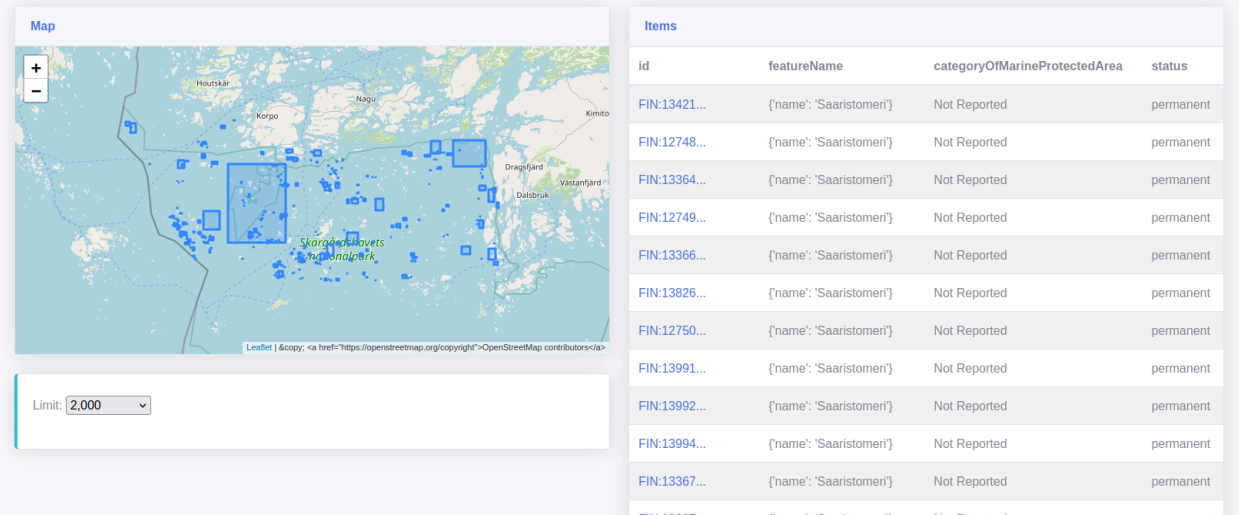
Figure 13 — WDPA Collection

Test Route #	Route Length (km)	Area of bounding box query (km²)	WDPA Source: Original MPA Data			WDPA Source: Bounding Box MPA Data		
			Features returned from bounding box query	Size of returned query	Features within route buffer	Features returned from bounding box query	Size of returned query	Features within route buffer
1	351	63 777	43	31.43 MB	5	43	52.43 kB	5
2	1 950	1 092 970	294	108.50 MB	0	294	353.49 kB	2

Figure 14 — Overview of using WDPA data on the two test routes

Test Route #	Route Length (km)	Area of bounding box query (km²)	JNCC Source: Original MPA Data			JNCC Source: Bounding Box MPA Data		
			Features returned from bounding box query	Size of returned query	Features within route buffer	Features returned from bounding box query	Size of returned query	Features within route buffer
1	351	63 777	8	70.93 MB	4	8	17.21 kB	6
2	1 950	1 092 970	15	96.99 MB	1	15	31.07 kB	1

Figure 15 — Overview of using JNCC data on the two test routes

Network emulation was implemented to simulate a network that had limited bandwidth, packet loss, and an increased delay. Even with low bandwidth the D102 client was able to successfully query the D100 server and display the bounding box MPA features that intersected the vessels buffered route. Conversely, using network emulation and then executing a query using the original MPA features would take significantly longer and often fail to load on the client.

10.3. Challenges and Lessons Learned

10.3.1. Spatial Filtering Using a Bounding Box Query

The bbox spatial filter, which is specified in Part 1 of the OGC API — Features Standard, is a useful method to query a server and reduce the volume of data returned to the client. However, using this method often covers a significantly larger geographic area than what is required (Figure 3). Additionally, maritime routes that have significant displacement in both the X and Y axes will have a large bounding box enclosing the route, whereas a route that has minimal displacement in either the X or Y axis will have a much smaller bounding box. Using the bounding box spatial filter can therefore mean that valuable network traffic in low-connected environments can be consumed by returning features that are irrelevant to the user's query. An alternative approach using Part 1 of the OGC API — Features Standard could be to use a series of bbox spatial filters for each segment of a route, either using a set specified distance or based on a change in bearing. Whilst not the most efficient technique, if only OGC API — Features — Part 1 is implemented by a server it would still reduce the overall area of the query and lower the number of features returned when compared to using a single bbox spatial query.

A more efficient method of performing a spatial filter would be to use the functionality of CQL2 in creating the spatial filter, as described in Part 3 of the OGC API — Features Standard. Using this additional functionality enables spatial filtering to be confined to other shape types and is not limited to a simple bounding box shape. In this scenario the buffered route of the vessel, or the route itself with some additional conditions specified in the spatial filter (i.e. distance from the route), could be used as the spatial filter (Figure 16). **By using a server that implements OGC API — Features — Part 3 this would significantly reduce the area of the query sent to the server and subsequently reduce the number of MPA features returned to the client.**
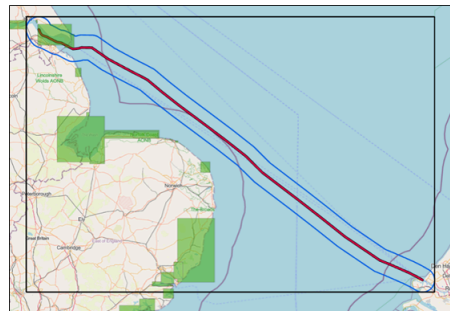
Figure 16 — Example of how using a bbox spatial filter (black line) covers a much larger area compared to a polygon shape of the buffered route (blue line).

During the pilot the D100 server used pygeoapi v0.12 to publish MPA vector data through an OGC API Features endpoint. However, the bbox spatial filter only works on certain data providers; GeoJSON and CSV providers cannot be filtered using a bbox query when using pygeoapi v0.12. If the bbox spatial filter approach is to be used, then the data would need to be in an appropriate format (i.e., provider) or hosted on a server with suitable spatial filtering capability. Additionally, CQL filtering on pygeoapi 0.12 only works with Elasticsearch providers;

this functionality was not enabled on the D100 server and therefore could not be tested during the pilot.

10.3.2. Using Bounding Boxes to Represent Features

It was clear from very early stages in the pilot that MPA features can be complex in their shape, especially in the littoral regions. This impacts the number of vertices that a particular feature has, and significantly increases the file size of the feature. When dealing with a small number of features this is not necessarily an insurmountable problem. However, when dealing with a collection that has hundreds or thousands of complex features it can severely impact the end users experience and is unsuitable for low connectivity environments. Whilst this affects the S-122 standard for MPAs it is not unique to S-122. There are a number of other datasets that potentially suffer from the same problem, i.e., anything with very complex shapes.

Therefore, creating a DDIL twin (a similar concept to a digital twin, whereby a simple and lightweight version of the data coexists to the original data) was suggested when the MPA shape needed to be simplified. Bounding boxes were created for some of the feature collections to test whether this concept would work and what issues would arise.

During the pilot there were several instances of MPA feature collections that had been converted to bounding boxes before the final two collections (WDPA, JNCC). There were some issues identified, especially with these earlier instances of bounding box data collections. For example:

- the features that were converted into bounding boxes used single polygons for all features, which was an issue for multi-part features; and

- the bounding boxes were an envelope-type minimum-bounding geometry of the feature.

MPAs that are multipart features, i.e., features that have the same attributes but consist of numerous individual features, were represented by a single bounding box (Figure 17). This would cause the vessel route to return a bounding box which may contain many individual MPA features.
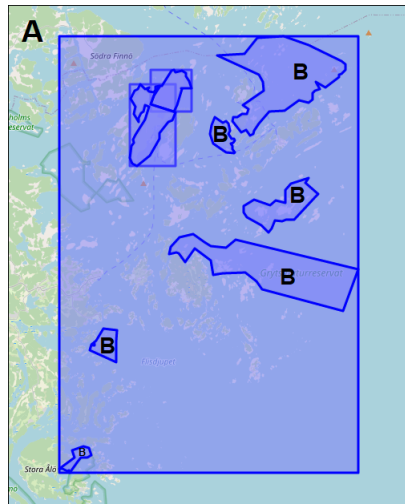
Figure 17 — Example of a single bounding box (A)
that represents numerous multi-part features (B)

Additionally, depending on the shape of the MPA feature, using an envelope type bounding box would sometimes provide large areas that did not contain MPA features (Figure 18). An alternative option is to use a convex-hull type bounding box which would not only provide a better representation of the feature, but also reduce the amount of empty space that does not actually contain MPA features (Figure 19). This would reduce the occasions of 'false positives' whereby a bounding box feature is returned to the client as it is within 5NM of the vessels route, however the actual MPA feature the bounding box represents is not close to the vessel's route. If the bounding box approach is going to be taken forward, these issues will need to be considered in future implementations.
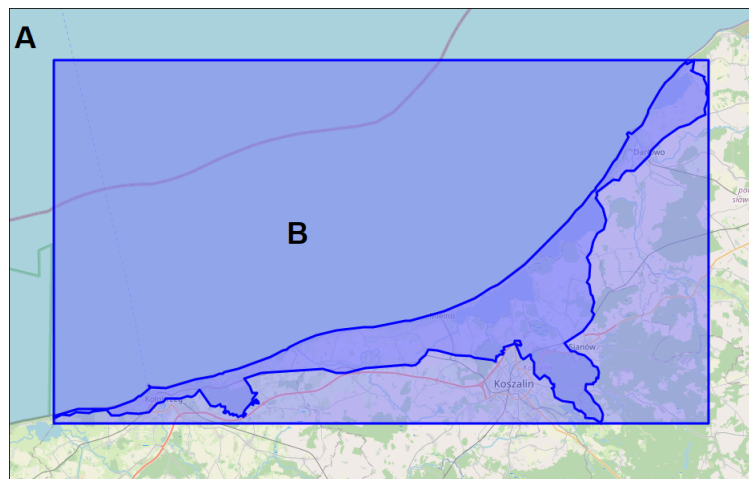
Figure 18 — Example of how an envelope type bounding box (A)
can lead to a large geographic area (B) that contains no MPA feature
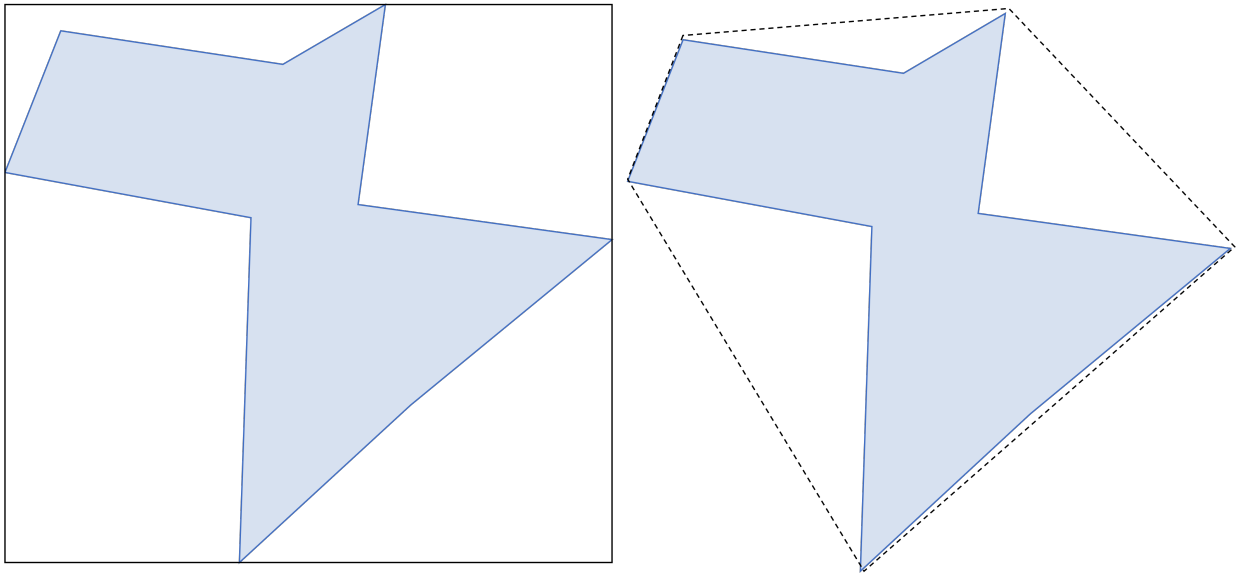
Figure 19 — Example of an envelope bounding box (left) and convex
hull (right) minimum bounding geometry of the same feature

The S-122 standard doesn't appear to readily use bounding boxes, only that of the overall dataset extent, however in the GML data format documentation there are several sections where `gml:boundedBy` attributes could be utilized to store the coordinates of a bounding box of the features and include the information in the metadata. Alternatively, the creation of bounding boxes for individual features can easily be achieved using S-122 MPA datasets in several GIS software applications (ArcGIS, QGIS, etc.) before the bounding box features are hosted onto a server.

10.4. Recommendations for Future Work

10.4.1. Further Enhancing MPA Filters

There is an opportunity for further development of the client alongside a server implementing OGC API — Features — Part 3 to use additional filtering to reduce the amount of data requested from the server. For example, giving the user the option to filter depending on what is important for their needs, which may change over time depending on their location or mission.

This could be accomplished by allowing the user to query using the attributes in the S-122 standard, such as the data provider, country, MPA status, category, vessel restrictions, etc. There is also the potential to increase the functionality of the client by allowing the user to request specific original MPA feature(s) once the bounding boxes have been returned.

Finally, the functionality on the server implementation could be expanded to clip the MPA features that intersect the query from the client and only return these segments of intersecting MPA features, further reducing the amount of data returned to the client.

10.4.2. Establishing a Data Schema for DDIL Environments

If using a DDIL twin for any data is to be considered going forward, then there needs to be some consideration for what the data schema would need to be. For the S-122 MPA data this would be the simplified bounding box versions of the original data. As the simplified version of the data is envisaged to be a supplement to the original data rather than a replacement, it would need to share common attributes with the original data and have a clear link back to the original features.

Whilst having additional metadata would slightly increase the file size, this is not the main contributing factor as the features remain a simple shape with few vertices. Some important parts of the DDIL schema are listed below. This is not an exhaustive list and consultation with relevant stakeholders in the community and beyond should be undertaken to fully understand what is important.

- Feature ID

- Feature Name

- Category

- Status

- Country

- Any restrictions in the area

- A clear notation that the feature is a DDIL representation and not the original feature

There are many other considerations that could be leveraged if the objective is to reduce either the size of the response payload or the complexity of the MPA feature boundary. The former can be addressed, as an example, through compression to (possibly) *geoParquet*. The latter point is a discussion that should be oriented towards the accuracy the client is willing to give up to approximate the MPA boundary based on BBOX.

10.4.3. Using Vector Tiles

Tiled feature data, colloquially referred to as Vector Tiles, enables the delivery of vector feature data in pre-defined tiles which enables small amounts of data to be delivered to the client and has been previously proven to work in DDIL environments. Some of the potential benefits could be:

- efficiently storing, delivering, and visualizing vector data from large datasets (such as S-122);

- varying levels of data 'resolution' allows low resolution over a large geographical area, or high resolution over a much smaller area;

- enables efficient caching of data;

- can provide clients with a hierarchy of available data, while awaiting requests for higher resolution tiles; and

- uses established techniques and APIs (OGC API – Tiles, OGC WFS 3.0, OGC WMTS).

This could be an alternative approach for using S-122 in low connectivity environments by hosting and serving the S-122 data as vector tiles. It is recommended that this is explored in the next phase of the FMSDI pilot.

11

DATA FUSION SERVER 1

11 DATA FUSION SERVER 1

The Data Fusion Server component was designed to ingest MPA datasets as well as other datasets, combine them, and serve them through an API built using the OGC API Standards. This component was demonstrated by IIC Technologies.

11.1. Status Quo

The fusion server was a development of some of the facilities on the Baltic Server designed to enhance the data content available in respect of Marine Protected Areas, to enhance the S-122 model following feedback from stakeholders, and to produce data fusing the old/new aspects of the model. An additional aim was to enable more sophisticated search, filter, and retrieval technologies to facilitate clients with richer functionality. These facilities have traditionally been non-integrated, using middleware such as GeoServer implementing older OGC Web Service standards.

The process of model enhancements and data examples conforming to S-100, which take advantage of them, is generally a process which takes many months of evolution with stakeholders. The IHO-established process for such development is model-driven but tends to be a predominantly manual one. The FMSDI project used a rapid application development methodology to speed up and prototype through demonstration the possibilities of enhanced data models using S-100 as a framework.

Data distribution by the stakeholders is generally currently accomplished by a "portal" methodology where specific websites offer data for browsing and download in a number of formats, both proprietary and open standards based. Although some have used previous OGC API Standards to stream data, many still use file download as a primary mechanism. Additionally, a number of regional "aggregators" exist where individual contributions are joined together under a single data model, representative of the regional aggregators identify and specific to them. Therefore, although OGC Standards have made inroads into the SDI development it is by no means pre-eminent and the presence of aggregators adds to the complexity. The fusion server concept contributes to this picture by providing specific vector feature API endpoints with a high degree of interoperability, data conformance to S-100, and adaptability using structured query techniques.

11.2. Technical Architecture

The fusion server was implemented with a set of extra API Endpoints co-located with the Baltic Server implementation. The technical services re-use the components from D100, namely PostGIS, pygeoapi, GeoJSON and bespoke software pipelines to create data conforming to a number of model enhancements proposed following stakeholder consultations. Unified

Modeling Language (UML) diagrams expressed using Enterprise Architect were transformed via IIC's embedded IHO Feature Catalogue extensions and then transferred into an S-100 conformant FC for implementation in the enhanced pipelines creating MPA data.

At this stage in the project it became apparent that the implementation of OGC API Features — Part 3 CQL querying was fairly basic in the pygeoapi components. Therefore, a previous version was sought, installed, and configured with a selection of datasets to try a more complete implementation of CQL. This seems to have been more successful and has been enough to try querying out via API programming, although, of course, GIS currently do not provide query interfaces conforming to the CQL standard.

It should also be recalled that many of the fusion aspects of the data endpoints can be realized by using multiple feature types within individual data types. Unlike most data frameworks, S-100 is inherently multi-feature so a "collection" in the IHO context (normally associated with a "dataset" is an aggregation of many different kinds of feature and relationships between them) so "fusion" in an S-100 ecosystem is often accomplished at the content level, not at the web services level.

The first element of the Fusion Server that was focussed on was the inclusion of data from broader authorities and its content.

The Danish habitat data, described in the section on the S-122 model, is an example of this mixture of individual data from different sources and the S-122 data model.

The integration of the habitat (WFS source) data with the S-122 model was also performed using the IIC Feature Builder Tool (Figure 20).
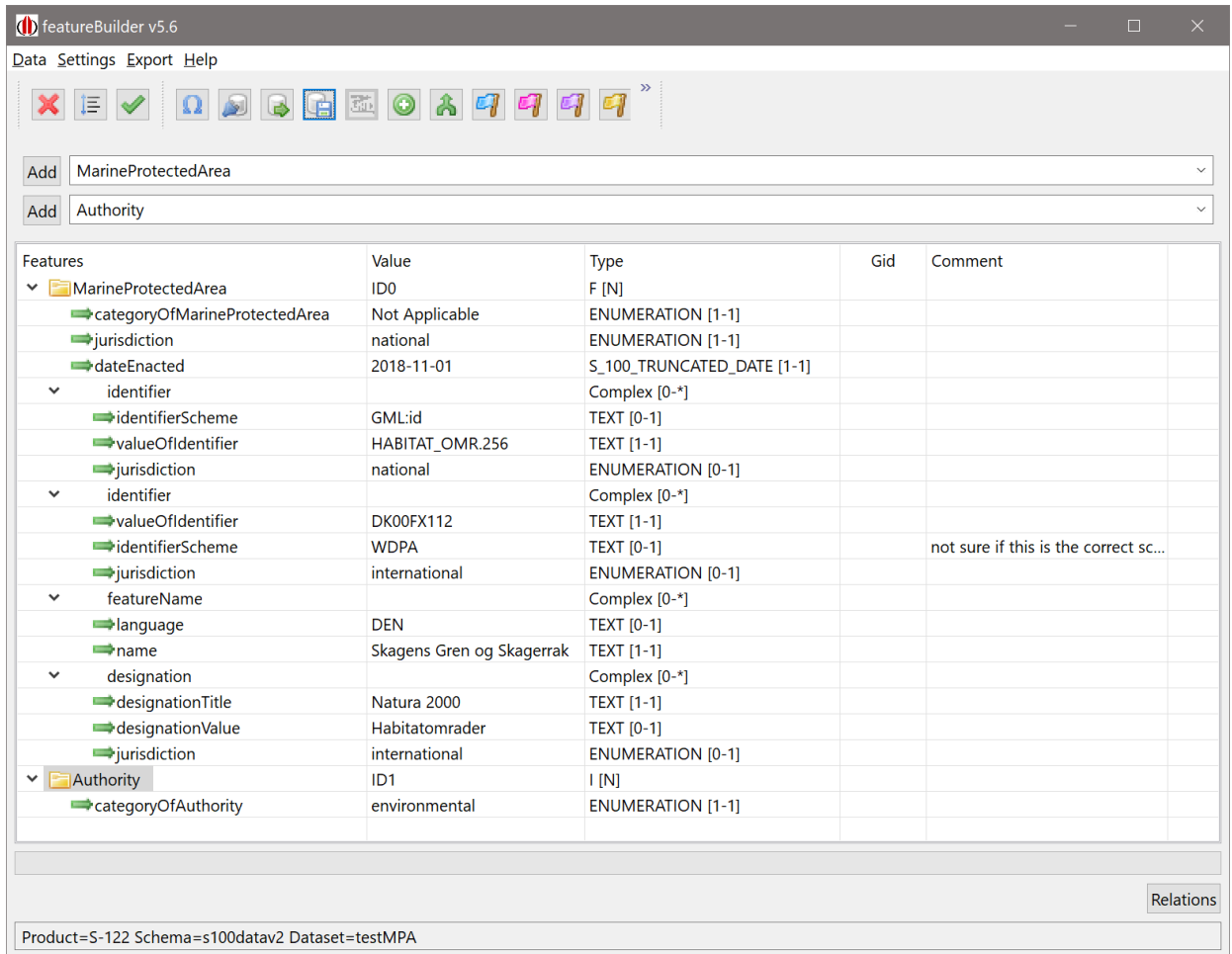
Figure 20 — Integrating Data With the IIC Feature Builder Tool

This data modeling and mapping exercise provided many suggestions for the data model enhancements which were then back-propagated into the original data sources to enhance the S-122 representation of the data.

The fusion server data was hosted in parallel with the Baltic Server and was accessed through a different port (5000). This was an implementation configured during the course of the project using an older version of the pygeoapi implementation which supports fuller capabilities for intelligent querying, specifically the CQL interfaces from OGC API Features — Part 3.

Data was ingested into PostGIS databases and then custom automated processes created S-100 GFM features and information types using an S-100 GFM database schema. Once populated, the data was then used to extract S-100 GFM GeoJSON which was then deployed under pygeoapi.

The current version of pygeoapi under development (13 at the time of writing) is fairly limited in its CQL support, but an earlier version implemented more complete support (v0.9.0). So, in order to provide facilities for broader scenarios the earlier version was implemented in parallel. This allows full search capabilities in addition to bounding box queries.

In order to demonstrate the processing potential of the API model a python *jupyter* notebook was written which explored various aspects of the server operation in annotated code. The URL below shows an example of a URL which makes accessible a simple "WITHIN" query.

```
http://35.176.64.149:5000/collections/S122DNK/items?f=html&filter-lang=cql-
text&filter=WITHIN%28geometry%2CPOLYGON%28%286.0+57.0%2C9.0+57.0%2C9.0+54.
0%2C6.0+54.0%2C6.0+57.0%29%29%29+AND+%28id+%3D+%22H18%22+OR+id+%3D+%22H19%22%29
```

Compound queries were also possible, such as queries selecting features from within a polygon and then filtering on the id:

Figure 21 — Testing CQL Queries on the D120 Fusion Server

There were many difficulties in implementing such queries reliably with *pygeoapi* as the CQL implementation is at an early stage and such complex spatial queries require full implementation in *pygeoapi* and more robust error detection and messaging. The initial results are encouraging, though feedback to *pygeoapi* should be structured and passed on as part of the project outputs.

In order to support more detailed scenarios of establishing which MPAs fit along individual routes, a pair of routes were digitized from existing maritime routing data from within the public domain through the IMO's recommended routes publication. These were made available on the query server for clients to ingest, buffer, and create proximity queries for, as seen in Figure 22.
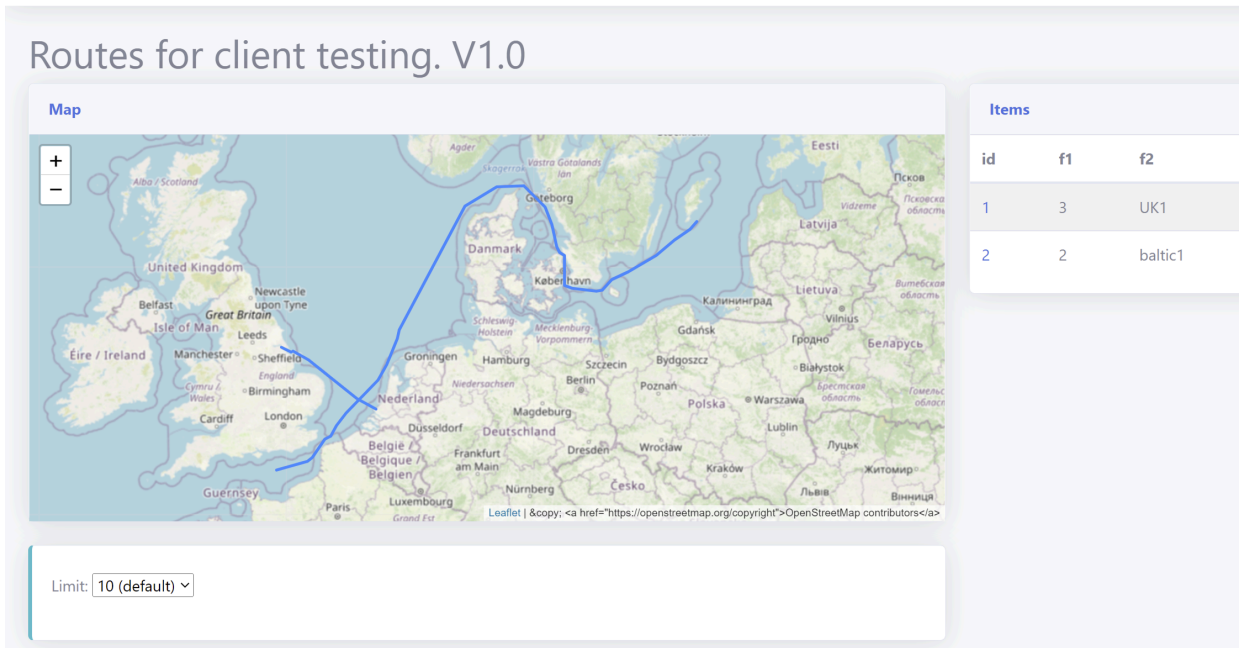
Figure 22 — *pygeoapi* Displaying Routes

The routes created themselves to represent a fusion of datasets, as they are created from maritime route data and digitized from Automatic Identification System (AIS) heat maps obtained from datasets in the public domain. There is no automated means to create such routes since the constraints on them in terms of maritime traffic flow are substantial. In order to support such scenarios, these routes were created manually – there is an interesting extension to this work in trying to come up with such routing data automatically from point cloud AIS data and navigational data constraints – this is beyond the scope of the existing project but much of the source data undoubtedly exists. Figure 23 shows Denmark MPA data, IMO routing features and AIS heatmap tracks.
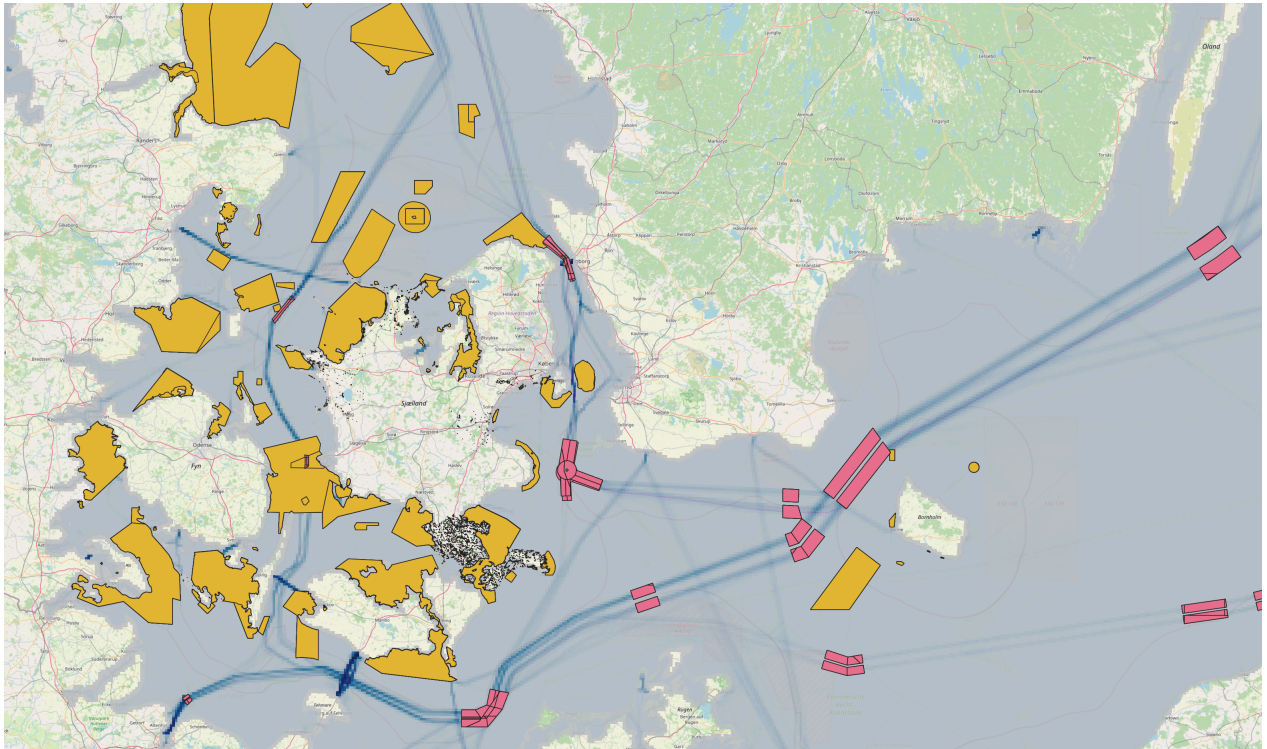
Figure 23 — Denmark MPA Data, IMO Routing Features and AIS Heatmap Tracks

11.3. Challenges and Lessons Learned

11.3.1. Using OGC API — Features to Serve MPA Data

Throughout the Pilot, many conversations were related to the use of API as opposed to more traditional methods of data distribution (i.e., flat files) and this could use some systematic description and application in our case. Much of this is generic, i.e., it applies to all API migration, not just those in an MSDI context, but MSDI stakeholders have a few unique characteristics and it is worth trying to describe such constructs for the marine geospatial data stakeholders.

API extracts data from the source

The most attractive element of the API model for data producers, particularly of MPA data, is the retrieval from the authoritative source of the data. The API model offers data producers a greater assurance that those wishing to access their data will do so through an authoritative route, rather than using aggregators who copy and store copies of data on intermediary websites.

This desire for users to have access to up-to-date, authoritative data is a long-standing requirement of data producers and the availability of open standards specifically related to APIs

favors this distribution route. This poses questions in the navigational context, obviously, most notably:

- when data producers distribute data both via API and via intermediaries for navigational use, how can they be assured that data is being used in the correct context; and

- should attempts be made to standardize data import to the bridge in such API formats as well. Some inroads have been made to this by existing standards bodies, most notably the Secure Communications (SECOM) standard of the International Electrotechnical Commission (IEC).

The question of intermediaries is also of note when data relating to real-time observations is transferred to vessels but these questions are being addressed elsewhere. An extremely important issue in the MSDI ecosystem is that of "provenance", also known as "authoritative" or, in the UN-GGIM framework "custodianship", the ability of the user to ascertain from whom the data has come.

The existence and ease of implementation of OGC API standards for S-100 GFM data is likely to provoke such debates and undoubtedly offers data producers the opportunity to migrate certain, if not all, services to a more dynamic, API model of distribution. When such migration is executed, metadata and attribution will be adapted to account for API distribution. For instance, data can be issued with shorter "expiry dates" and per-feature digital signatures to enable tighter control/influence over its onward use.

Filter/query retrieval

The other major changes with API distribution of data is the ease of automation by client services, the format-neutrality of such APIs, and the fine control over retrieval, which is not present in any of the native S-100 encodings in Part 10. APIs allow for a rich set of server-side filters to be implemented. The Baltic server implemented used a combination of the latest version of open source tools, together with an earlier version equipped with a full CQL-enabled OGC API — Features implementation. This allows for simple filters on data fields, compound queries (using "AND" and "OR" conjunctions) and spatial queries (e.g., "WITHIN", "INTERSECTS") as well as simpler queries against bounding boxes. Although specification of such queries is fairly mature, the implementations are still at a somewhat early stage. The pilot was able to demonstrate retrieval via API and filtering server-side using test datasets and the potential is obviously enormous.

This is an area in need of further development as it is a major advantage for implementers of automated services. The CQL framework (supported by OGC API – Features — Part 3) sets out ways of organizing such querying capabilities and a more in-depth investigation of its capabilities, against the data structures implemented in this project should be done. This would show any limitations of using S-100 GFM data as the object of the queries, e.g. querying against complex attribution and querying against linked/relationships to information types.   ==== The S-100/S-122 Model

During the course of the data exploration from Phase 2 a number of potential enhancements to the S-122 data model were observed. These stem from:

- the fairly narrow focus of Marine Protected Areas within S-122 itself; and

- dialogues with stakeholders in the region concerning how MPA data is managed and maintained and its content.

Throughout the Pilot, a number of consultations were held with stakeholder communities, not just hydrographic offices. This focused on whether definitions, content, and management information was sufficient and complete in the current S-122 standard and what, if anything, could be done to enhance it. The consultations reinforced the enormous breadth of agencies involved in the creation of MPA data across the region. Many countries have several agencies responsible for maintaining databases of MPA data, against legislation, policies, or conventions. What became clear during the project was that MPA data is a very broad category of identified marine spaces. Each state has an individual approach to its management and any enhancements to S-122 should try to account for those if the aim is to enable its implementation across such agencies.
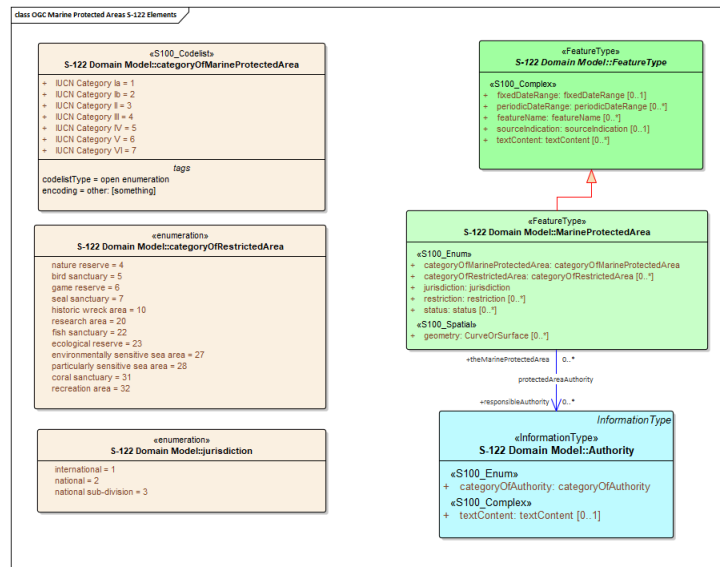
Figure 24 — S-122 Elements

The current S-122 model is fairly basic in terms of its representation of MPA data. It allows for a single feature (Marine Protected Area), shown in the standalone elements diagram in Figure 24. The model was supplied by IHO for use within the project. In order to focus on MPAs, only the directly relevant elements were included in the initial feature catalogue and data instances.

The model clearly shows how MPAs, implemented with curve or surface geometry, have a single mandatory attribute representing the IUCN category, a simple enumeration for jurisdiction, status, and restrictions which may be in place for the MPA. Restrictions, as seen in Figure 25, in this case are purely maritime in nature and profiled from the IHO registry code list.

Figure 25 — S-122 Class MPAOtherEnumsAndCodelists

This encoding, whilst a good fit for maritime use cases, does not currently reflect the broader application of MPAs in different geospatial agencies and the richer attribution required for those uses. The proposed amendments to the data model are initial proposals which need consideration by an applicable IHO working group, although there is a broader question in terms of how IHO itself should approach the design/modeling of such product specifications – should MSDIWG manage its own domain or should it extend those features currently within the registry and maritime domain?

A number of proposed suggestions have been implemented in the model used for the server.

- Add three new values to `categoryOfMarineProtectedArea`. This is the primary designation — the IUCN category for any MPA. This allows for Not-Includes, Not specified, Unknown – it was noted that many MPAs in the source datasets simply do not have IUCN categories associated with them.

- Add complex attributes to record other designations, e.g., WDPA, HELCOM, JNCC, and also include a jurisdiction for recording at what level the designation exists. Designation records the scheme and the categorization within the scheme.

- Add a dimensions complex attribute to record the calculated dimensions of the MPA. These are often used for reporting. Include unit of measure, which needs harmonization with registry.

- Add mandatory enactment date and optional update date to all MPAs. This could equally be termed `validFrom` date but enactment seems to be used more frequently. There is a distinct difference between enactment dates and start/end date.

- New information added representing Management Plans. This information has a `planStatus` attribute (currently only draft but planStatus needs more values). A new association was added between MPA and management plan. Many MPAs have management plans are associated via URL.

- All Feature Types now have multiple identifiers. These can be from different schemes creating a complex attribute with a scheme designation, a value, and a jurisdiction as many

identifiers are set at different levels. Identifiers also have a `textContent` attribute to hold arbitrary textual data about the identifier.

- `producerCode` added as a simple attribute to Agency. This is the code used to identify them — it could be part of dataset metadata but needs to be at a feature level as multiple agencies could exist within a single dataset.

- Added regional to jurisdiction. There is a potential to add local as well.

These proposals will form the core of a more formal proposal to the relevant IHO subgroup(s) (MSDIWG and NIPWG) on how the S-122 model can be enhanced to better represent the needs of broader stakeholders. The exact form of the enhancements may require adaptation; this is also independent of any adaptation of the IHO working group structures in this area. The core question is whether S-122 should be adapted, or a new product specification, e.g. for marine geo-regulation, is convened within the IHO for such data.

S-100's ability to implement information types provides a useful way of representing the management agencies responsible for individual MPAs – in the S-122 model these are "Authorities", responsible for administration and management of the MPA. In reality this relationship, nationally, is a complex one and there is an open question of whether those richer relationships should be modeled within S-122. From a maritime perspective they are probably not useful for end users (e.g. a vessel is unconcerned with whether a national environment agency or federal authority are responsible for the management and maintenance of an MPA) but interoperability between non maritime actors could reasonably be used as a justification for a richer "Authority" structure.

ISO 19152 potentially provides such a way of representing both Authority structures and their rights, responsibilities and restrictions. When S-122 was originally drafted no ISO 19152 entities existed in the IHOs geospatial registry. The implementation of ISO 19152 within S-121 entered features into the registry transcribed from the ISO standard. Therefore, inclusion in S-122 would be less onerous.

More work is required to map examples and test the application of ISO 19152 for some providers. In addition, the question of what to do with the existing structures in S-122 and the numerous other NIPWG/IHO product specifications using Authority, Restriction and who implements Responsibilities as types of restrictions, etc. This potentially supports the idea of S-122 or a future MPA product spec being under the domain of an MSDI body, rather than attempting to co-exist wholly with maritime or navigational products. However, these debates are outside the scope of the OGC project. The project asserts that ISO 19152 provides a richer structure than that currently implemented and one which may have application for administrations with particular mandates for using such standards. Using ISO 19152 would help with more complex restrictions other than those covered in the current enumerations.

Most of these enhancements are still focused on data exchange. There were numerous other identifiers in the Baltic data which were raised under discussion which related to how the agency used/managed the data but which were visible to end users, including:

- responsible users (this could be done with contact details);

- change codes (summarizing the content of changes) – an enhancement to the status enumeration could deal with these; and

- last updated dates – these weren't added to the model as the start/end date could be used to represent them. This is one to explore, though.

11.3.1.1. Revised MPA Model

A revised version of the proposed (MPA only) model is shown in Figure 26 and Figure 27.
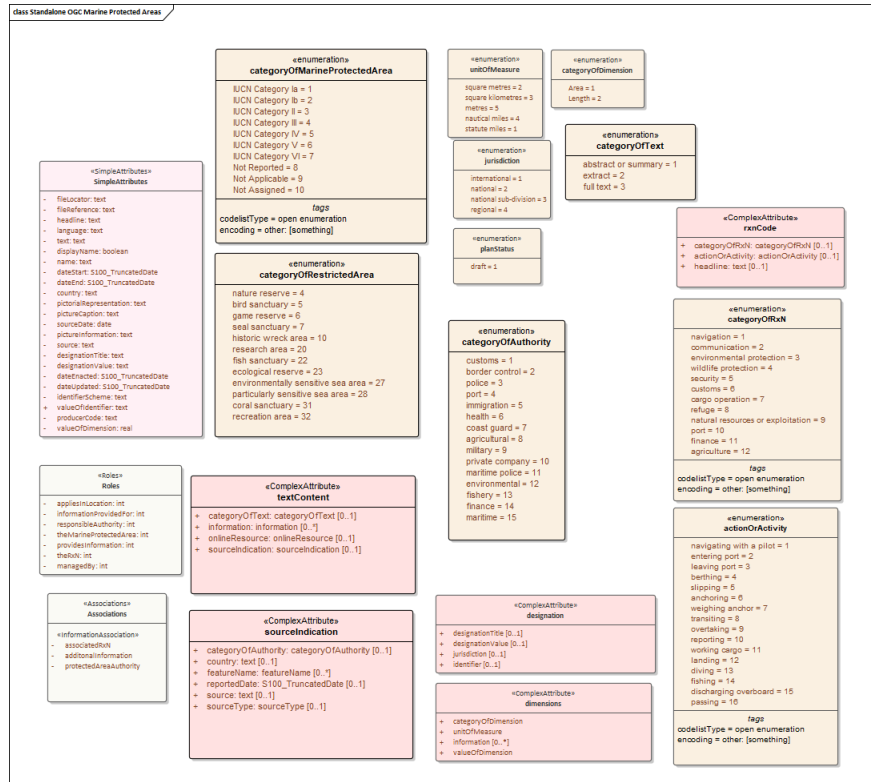
Figure 26 — Revised MPA Model
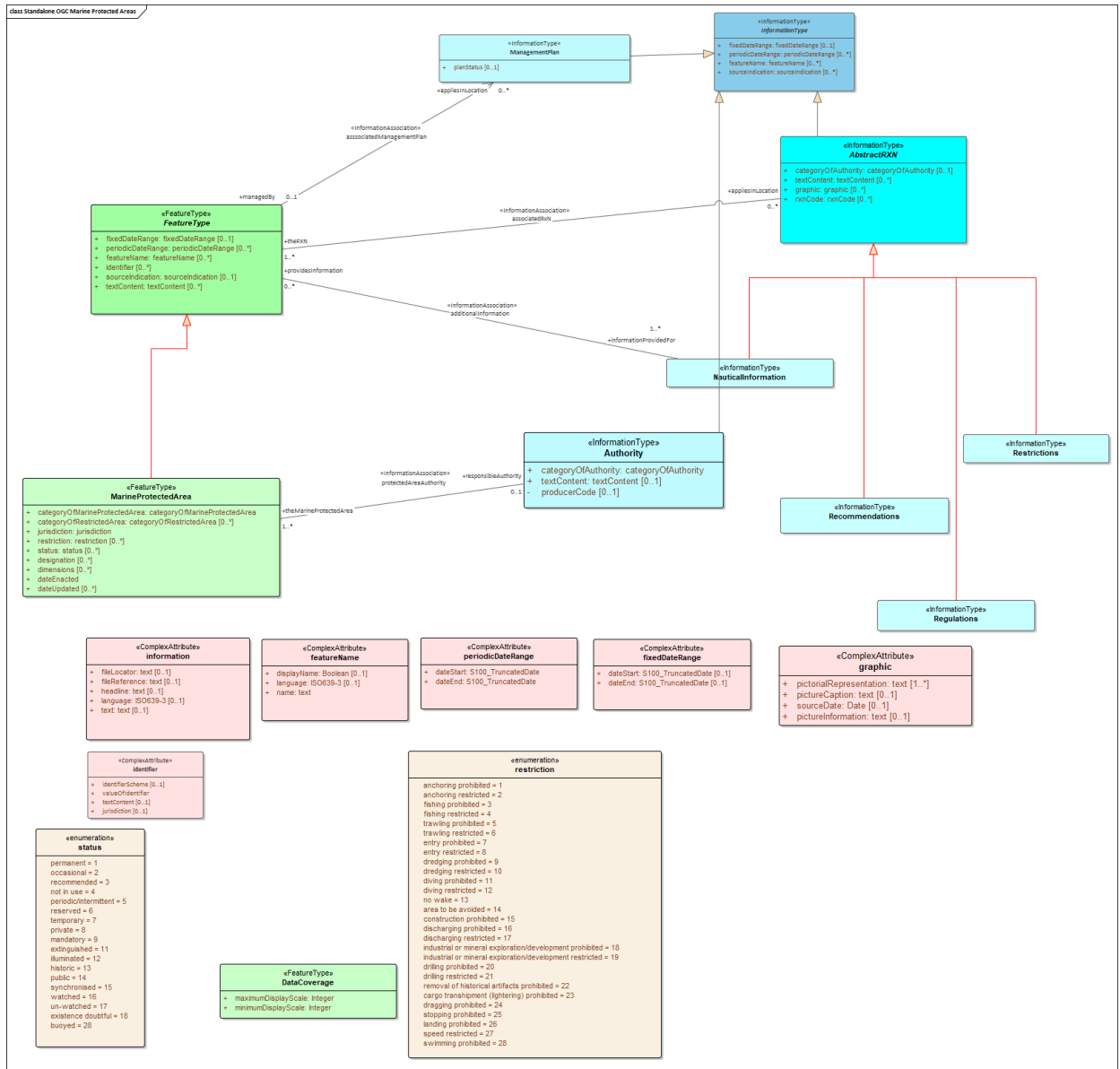
Figure 27 — Revised MPA Model

A number of proposals to the S-122 model were made and then implemented as enhancements to the S-122 feature catalog. An automated process converts the UML diagrams into IHO feature catalog components, including abstract and sub/super type relationships. IIC's own Feature Designer application resolves these abstract features into complete, realized feature catalogs suitable for data processing, data creation and distribution via OGC web services.

Data from sponsors, as well as supplementary data received during the course of the project, were transformed into the revised data model and added to feature collections. The UK Joint Nature Conservation Committee (JNCC) dataset was a primary example of the enhanced data model containing alternative designations, dimensions, and enactment dates, as seen in Figure 28.
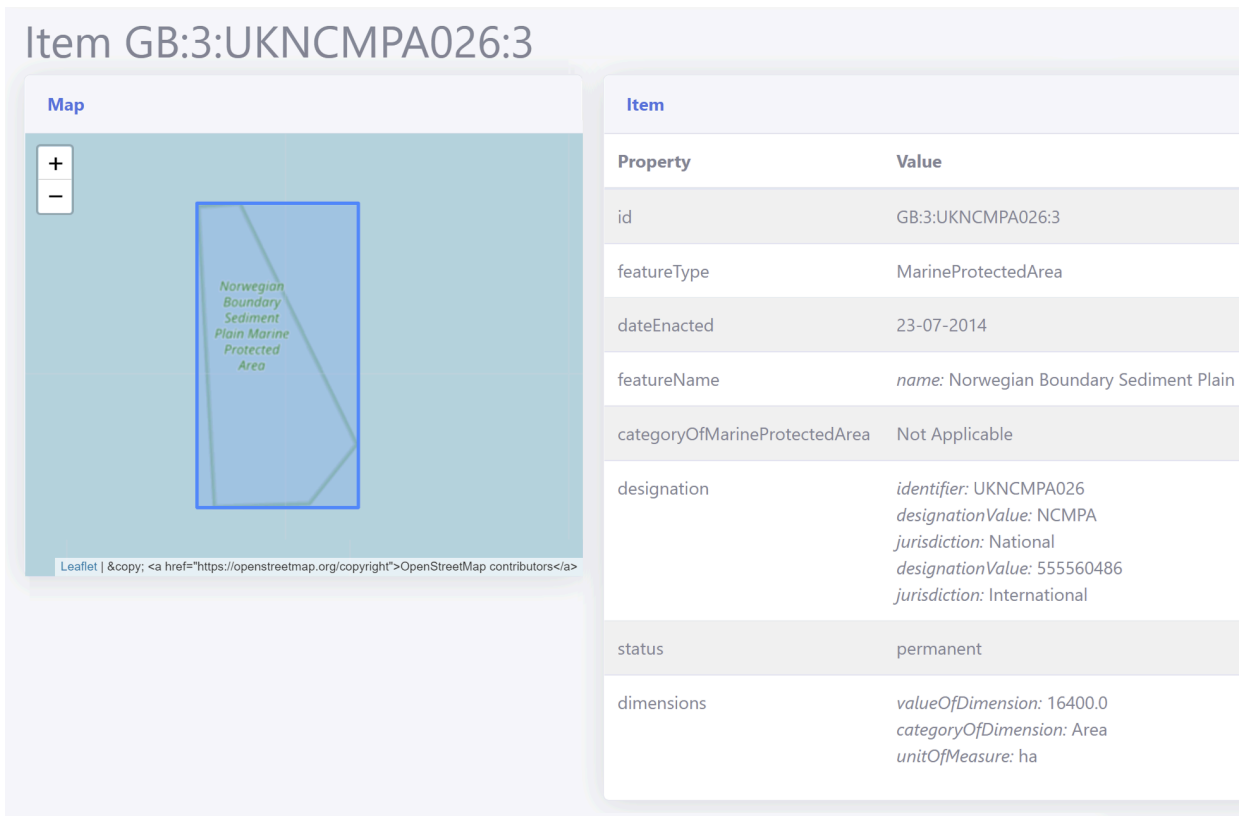
Figure 28 — JNCC Dataset Mapped With the Revised MPA Model

The Danish Habitat dataset is an example of data which was received through the sponsors during the project (in dialogue with the Danish authorities) and its values mapped to those in the enhanced S-122 model. This data, not traditionally used for maritime MPAs, fits the description of a protected area and is an example of how data from broader sources can be brought into the S-122 MPA domain, as seen in Figure 29.

http://35.176.64.149/pygeoapi/collections/S-122JNCC_ES_BB/items/GB:3:
UKNCMPA026:3
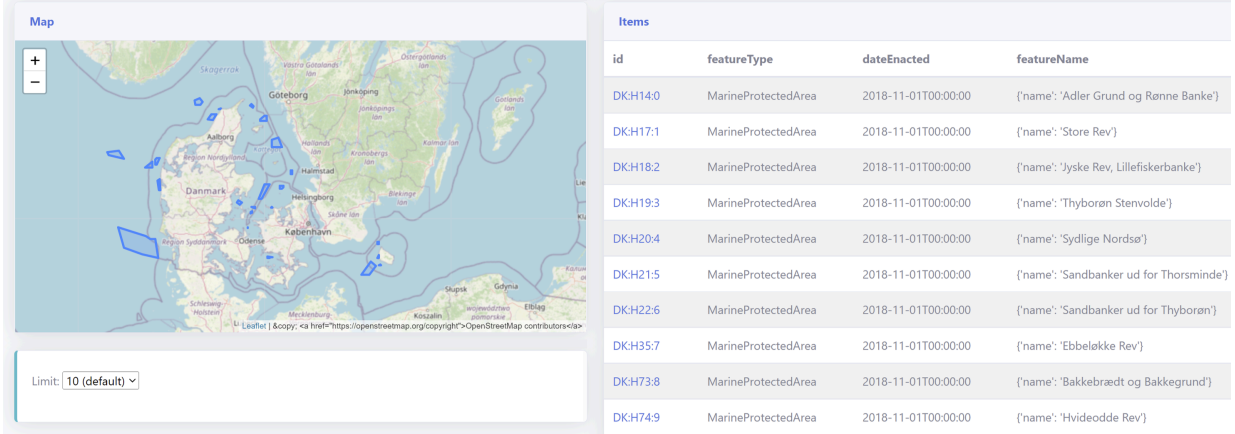
Figure 29 — Danish Habitat Dataset Mapped With the Revised MPA Model

12

DATA FUSION SERVER 2

12 DATA FUSION SERVER 2

The D-122 Fusion Server successfully published feature and coverage data sources processed into an Icosahedral Snyder Equal Area Aperture 3 Hexagonal (ISEA3H) DGGS for discovery as EDR Collections and EDR Parameter Queries. The D-122 Fusion Server component has been demonstrated by the University of Calgary and has successfully received and responded to EDR query requests from the Fusion Clients D-122 and D-123.

12.1. Status Quo

The University of Calgary Geomatics Department includes a leading group of DGGS researchers who have published widely about DGGS since 2009 and are active on the OGC DGGS Standard Working Group and Domain Working Group, several previous OGC Testbeds and Pilot Projects, and through the Standards Council of Canada – ISO. The team includes members that have developed the PYXIS ISEA3H DGGS that was used in this pilot project and their work is disclosed in several DGGS related patents.

12.2. Technical Architecture

An implementation of the OGC API — EDR Standard could serve as an interface to access a DGGS-based fusion server. This implementation should be done in combination with the OGC API — Features endpoint, providing an interface to features that could be used to query the resources, narrow the search, and drill up/down through the data from each resource.

A DGGS fusion server ingests data from different sources (features and coverages) into a common spatial framework, as seen on Figure 30.
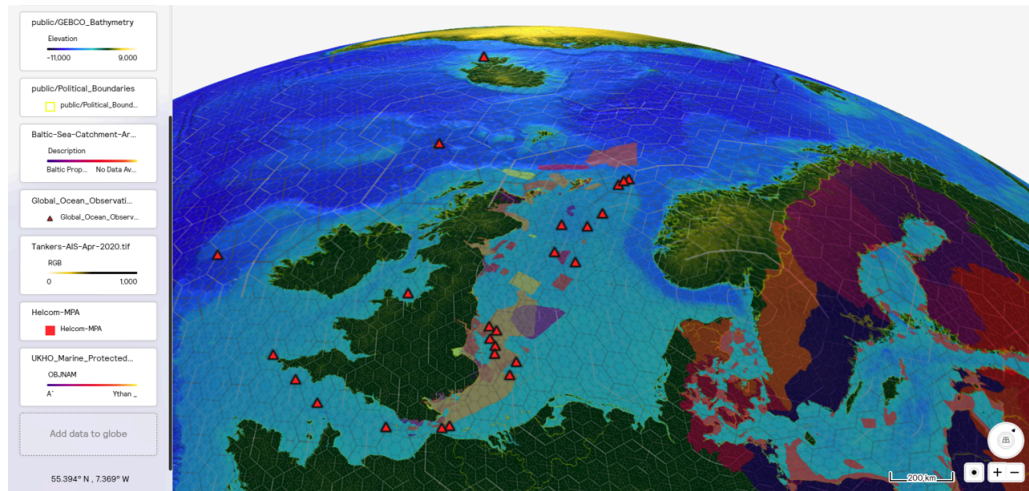
Figure 30 — MPAs From Diverse Sources

Figure 31 presents a schema of the fusion server and the request/response to the API endpoint {root}/collections, with the optional spatial filter by bounding box. The returning JSON contains a collections JSON containing all the collection items matching the query.
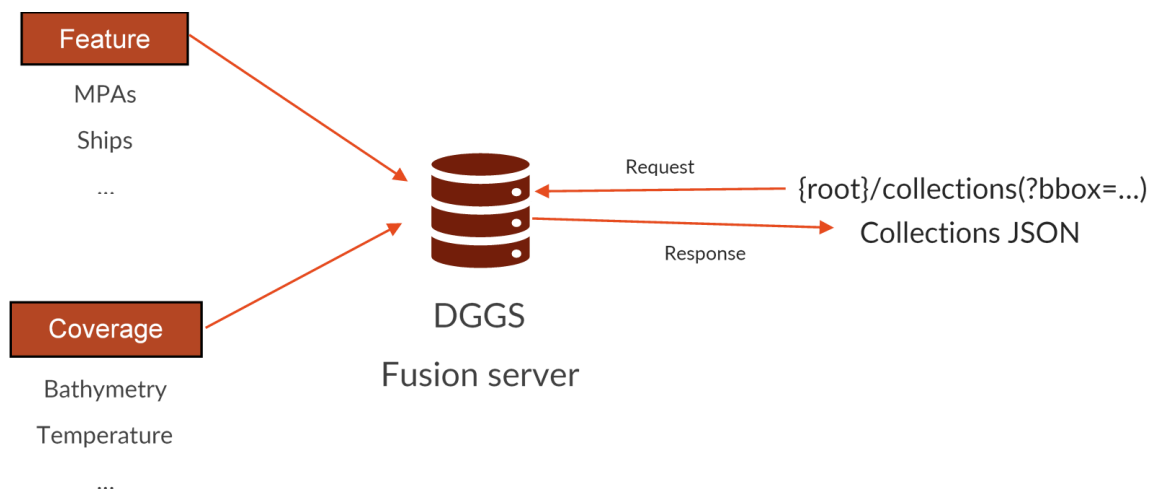
Figure 31 — DGSS Fusion Server Workflow

Even though the JSON schema is the same, the collection items returned will have different properties and can be used differently depending whether they come from a feature datasource or from a coverage datasource. However, in terms of data structure, this distinction is no longer necessary since the data is fused into a common framework, in this case a hexagonal DGGS. Both coverages and vectors get mapped into a hierarchy of hexagonal cells, which spatially aligns the data. The data values delivered through the EDR API have been calculated using this data structure, and it implies a change of mindset in terms of thinking of it as a traditional coverage. The EDR standard specifies, _"A typical EDR data resource is a multidimensional dataset that could be accessed via an implementation of the Web Coverage Service (WCS) standard". There is still a gap on DGGS research in knowing what kind of data structure or standard could be used to deliver its data. An extension of WCS could be an option. The objective was to explore the option of delivering calculated values using EDR API, allowing a

traditional web mapping client to access this data, and enabling as many DGGS capabilities or advantages as possible.

The code below shows a collection item coming from a coverage datasource, in this case, GEBCO elevation. This collection item maps to the EDR API collection schema, containing id, extents, links and parameter names. The parameter names specify all the parameters offered by the collection and their types. A query to the collection can retrieve one or any number of those parameters. For example, a collection coming from a NetCDF coverage datasource might contain temperature, precipitation, and wind speed as parameter names. An area query on this collection could ask for only temperature, temperature and wind, or all the parameter names.

```
{
"id": "gebco_2014_1d-nc",
"title": "Bathymetry",
    "description": "Dataset that contains height data from land and sea from
the GEBCO dataset of 2014",
    "links": [
        {
            "href": "https:digitalearthsolutions/api/v1/edr/collections",
            "rel": "self",
            "type": "application/json"
        }
    ],
    "extent": {
        "spatial": {
            "bbox": [
                -180.0,
                -90.0,
                180.0,
                90.0
            ],
            "crs": "4326"
        }
    },
    "parameter_names": {
        "Elevation": {
            "id": "Elevation",
            "observedProperty": {
                "label": "Elevation"
            },
            "type": "Parameter",
            "description": "",
            "data_type": "knInt16",
            "unit": {
                "label": "meters",
                "symbol": {
                    "value": "m"
                }
            }
        }
    },
    "data_queries": {
        "area": {
            "link": {
                "href": "http:34.95.36.182/api/v1/edr/collections/gebco_2014_
1d-nc/area",
                "rel": "data",
                "hreflang": "en",
                "variables": {
```

```
            "title": "Area query",
            "query_type": "area",
            "output_formats": [
                "GeoJSON"
            ],
            "default_output_format": "GeoJSON",
            "crs_details": {
                "crs": "4326",
                "wkt": "GEOGCS[\"WGS 84\",DATUM[\"WGS_
1984\",SPHEROID[\"WGS 84\",6378137,298.257223563,AUTHORITY[\"EPSG
\",\"7030\"]],AUTHORITY[\"EPSG\",\"6326\"]],PRIMEM[\"Greenwich
\",0,AUTHORITY[\"EPSG\",\"8901\"]],UNIT[\"degree\",0.
01745329251994328,AUTHORITY[\"EPSG\",\"9122\"]],AUTHORITY[\"EPSG\",\"4326\"]]"
            }
        }
    }
  }
}
```

Collection Item Coming From a Coverage Datasource.

The code below shows a collection item coming from a feature datasource, in this case, MPAs. This item conceptually represents a feature collection. However, as explained before, there is no distinction between features and coverages, and "features" is treated as an EDR datasource, and it can be delivered using the same EDR API collection schema. The parameters-name in this case is an object containing the attribute schema of the feature collection item and the type of each attribute. This would allow querying by certain attributes of the feature, the same way a collection coming from a coverage datasource can be queried by their coverage fields.

```
{
    "id": "helcom-mpas",
"title": "localhost.637800519905322715.json",
"description": "c:\\GGS\\gallery\\.pyx\\localhost\\rr7NBqI6zOKHlir1avJY7Js4M5Jr
IDGncqgyn0QocQ\\localhost.637800519905322715.json",

    "links": [
        …
    ],
    "extent": {
        …
        }
    },
    "parameter_names": {
        "MPA_ID": {
          "id": "MPA_ID",
           "observedProperty": {
                "label": "MPA_ID"
          },
    "type": "Parameter",
        "description": "",
        "data_type": "knInt32",
        "unit": {
            "label": "",
            "symbol": {
                "value": ""
        }
     }
     },
```

```
                    "Country": {
                        "id": "Country",
                        "observedProperty": {
                            "label": "Country"
                        },
                        "type": "Parameter",
                    "description": "",
                    "data_type": "knInt32",
                    "unit": {
                        "label": "",
                        "symbol": {
                            "value": ""
                        }
                    }
                    },

                    "MPA_status": {
                        "id": "MPA_status",
                        "observedProperty": {
                            "label": "MPA_status"
                        },
                        "type": "Parameter",
                    "description": "",
                    "data_type": "knInt32",
                    "unit": {
                        "label": "",
                        "symbol": {
                            "value": ""
                        }
                    }
                    }
                    …
                },
                "data_queries": {
                    …
                }
            }
```

Collection Item Coming From a Feature Datasource.

12.2.1. Collections

Data offered through an implementation of OGC API — Environmental Data Retrieval is organized into one or more collections of data. The 'Collections' resource ({root}/collections) provides access to the list of collections. The code below presents an example of a returning JSON from the collections ({root}/collections) resource. It contains an array of links and an array of collection items.

```
{
    "links": [
        {
            "href": "http:host/api/v1/edr/collections",
            "rel": "self",
            "type": "application/json"
        },
        ...
    ],
    "collections": [
```

```
              ...
       ]
}
```

Each collection item is an object that can contain the following information.

- Information about the collection: title, description, id, CRS, and extents.

- Parameter names: provides information about the data parameters supported by the collection. As a set of key-value pairs where the key is the name of the parameter and the value is a Parameter object.

- A set of links:

  - Detailed description of the collection, represented by the path `{root}/collections/{collectionId}` and link relation self;

  - Links to retrieve data according to supported query patterns, represented by the path `{root}/collections/{collectionId}/{queryType}` and link relation data; and

  - Link to the metadata about items available in the collection (OGC Features API), represented by the path `{root}/collections/{collectionId}/items` and link relation items.

The collection query endpoint (`{root}/collections`) also allows specification of a bounding box parameter (`?bbox=`) to narrow down the collection search and select an overall area of interest (Figure 32). The returning JSON contains a list of collection items spatially filtered by a BBOX.

{root}/collections?bbox=-170, -80,170,80

Figure 32 — First selection of an area of interest represented by the bounding box parameter of the collection endpoint

12.2.2. Queries

The query operation supports access to the EDR resource and drill-down through the data values with more specific spatiotemporal queries. The query endpoint (`{root}/collections/{collectionId}/{queryType}`) supports spatiotemporal queries based on a sampling geometry; position, area, location, corridor, or trajectory.
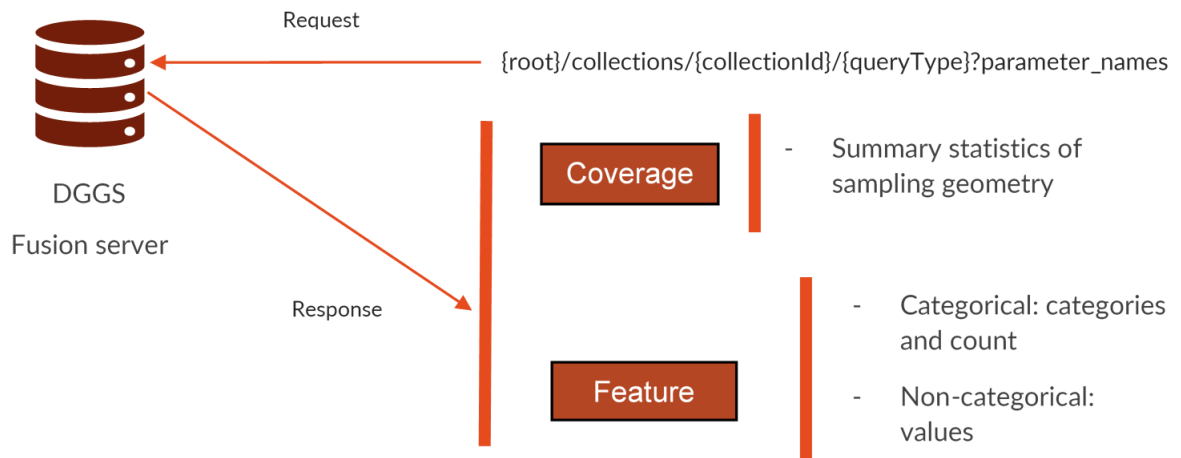
Figure 33 — DGSS Fusion Server response to the question "what data values are here?"

The fusion server response will consist of features containing the sampling geometry and a set of properties. This set of properties is again different if the datasource is coming from a feature or a coverage (Figure 33).

- Coverage: statistical summaries (min, max, average) of the data values contained within the sampling geometry. The code below shows an example of a statistical summaries return.

```
{
    "id": "1-000400002030050",
    "properties": {
        "gebco_2014_1d-nc/Elevation/Mean": "123",
        "gebco_2014_1d-nc/Elevation/Max": "143",
        "gebco_2014_1d-nc/Elevation/Min": "59"
    },
    "geometry": { ... },
    "type": "Feature"
}
```

Server JSON Response of Data Query Coming From a Coverage Datasource

- Feature: In this case, it depends on the type of parameter name that was queried. If the parameter is categorical, the response will be a list of categories and the count of features per category. If the parameter is not categorical, it will merely return the values contained in the sampling geometry. The following two codes show an example of a categorical parameter return, and a non-categorical parameter return.

```
{
    "id": "1-000400002030050",
    "properties": {
        "helcom-maps/Country/Denmark": 2,
        "helcom-maps/Country/Finland": 1
    },
    "geometry": { … },
    "type": "Feature"
```

```
        }
```

**Server JSON Response of Data Query Coming From
a Feature Datasource and a Categorical Parameter**

```
{
    "id": "1-000400002030050",
    "properties": {
        "helcom-maps/ID": [
            123,
            12,
            1
        ]
    },
    "geometry": { … },
    "type": "Feature"
}
```

**Server JSON Response of Data Query Coming From a
Feature Datasource and a Non-Categorical Parameter**

12.2.3. Endpoints Examples

Get all collections available on the server

Collections query on server.

https://digitalearthsolutions.com/api/v1/edr/collections

Get all collections available on the server by bbox

Collections query on server. Bbox parameter is a comma separated list of an area of interest.

https://digitalearthsolutions.com/api/v1/edr/collections?bbox=1.3959503173828125,43.26695632044282,1.402130126953125,43.269956206904

Get elevation summaries within an area defined by coordinates

Area query pattern on GEBCO collection. Params is the coordinates of the area in WKT. Parameter names specify what values to retrieve, in this case RGB.

https://digitalearthsolutions.com/api/v1/edr/collections/gebco_2014_1d-nc/area?coords=POLYGON-107.196 51.411,-99.460 55.983,-88.206 49.395,-95.943 42.310,-103.328 49.395,-107.196 51.411&parameter_name=RGB

Get temperature summaries within an area defined by coordinates

Area query pattern on temperature collection. Params is the coordinates of the area in WKT. Parameter names specify what values to retrieve, in this case RGB.

https://digitalearthsolutions.com/api/v1/edr/collections/temperature/area?
coords=POLYGON-107.196 51.411,-99.460 55.983,-88.206 49.395,-95.943 42.310,-
103.328 49.395,-107.196 51.411&parameter_name=RGB

This query also allows multi polygon features in WKT.

https://digitalearthsolutions.com/api/v1/edr/collections/temperature/area?
coords=MULTIPOLYGON(-107.196 51.411,-99.460 55.983,-88.206 49.395,-95.943 42.
310,-103.328 49.395,-107.196 51.411, -107.196 51.411,-99.460 55.983,-88.206 49.
395,-95.943 42.310,-103.328 49.395,-107.196 51.411)

Endpoints – future implementation

The following are endpoints envisioned for future implementations.

- Get elevation summaries within a feature coming from feature server.

  Feature API items request to access the particular feature by id (in this case 1).

  https://digitalearthsolutions.com/api/v1/edr/collections/helcom-mpas/
  items/1`

  Returns: location id to use for subsequent queries.

- Location query pattern using the location id retrieved by the feature server query. Parameter names specify what values to retrieve, in this case elevation.

  https://digitalearthsolutions.com/api/v1/edr/collections/gebco_2014_1d-nc/
  locations/helcom-maps-1&parameter_name=Elevation

- What MPAs are within an area defined by coordinates.

  Area query pattern on MPAs collection. Params is the coordinates of the area in WKT. Parameter names specify the parameter name id, to be able to obtain the IDs of the features to query by.

  https://digitalearthsolutions.com/api/v1/edr/collections/helcom-mpas/area?
  coords=POLYGON-107.196 51.411,-99.460 55.983,-88.206 49.395,-95.943 42.
  310,-103.328 49.395,-107.196 51.411&parameter_name=ID

  Return: list of MPAs Ids (1,2,3)

- Feature API items request to access the particular feature by id (in this case 1).

  https://digitalearthsolutions.com/api/v1/edr/collections/helcom-mpas/
  items/1

  Returns: full feature

- How many MPAs per country within an area defined by coordinates.

  Area query pattern on MPAs collection. Params is the coordinates of the area in WKT. Parameter names specify the parameter name country, to be able to group by country.

  `https://digitalearthsolutions.com/api/v1/edr/collections/helcom-mpas/area?coords=POLYGON-107.196` `51.411,-99.460 55.983,-88.206 49.395,-95.943 42.310,-103.328 49.395,-107.196 51.411&parameter_name=Country`

  Return: list of countries and its count

  `https://digitalearthsolutions.com/api/v1/edr/collections/helcom-mpas/area?coords=POLYGON-107.196` `51.411,-99.460 55.983,-88.206 49.395,-95.943 42.310,-103.328 49.395,-107.196 51.411&parameter_name=Country/Denmark`

  Return: list of ids of features with country Denmark

12.3. Challenges and Lessons Learned

12.3.1. Using DGGS

In summary, a DGGS is designed to fuse the data values – attributes, parameters — of two or more data sources into the cells of the DGGS. There they are indexed and can be aggregated up the hierarchical data structure of the DGGS as statistical summaries and lists for efficient processing, storage, transmission, visualization, and analysis. DGGS serve two distinct end purposes:

- To Explore: *what is here?* — a response to a query requesting a summary of values and or lists of qualities describing a given geographic area of interest; and

- To Search: *where is it?* — a response to a query requesting geographic areas that correspond to a filter on the available data values.

The need to efficiently search and explore multiple sources of heterogenous data is the reason DGGSs were designed as a fixed earth equal-area, hierarchical, global coverage. While there are other less efficient ways to search and explore heterogenous geographic data, Pilot work has proven that there are no reasonable replacements for DGGS geometric structure using either vector features or raster coverage geometries. If there were, DGGS unique geometric representation of the globe would not have been a necessary standard.

This means that while the EDR API has been shown in the pilot project to provide a naïve client with the tools to successfully "explore" DGGS data by requesting statistical summaries of aggregated fused data values for a particular area of interest – polygon, bounding box, etc. – it would not give the client the capability to "search".

The Common Query Language would be a convenient method of searching by filtering data values in a DGGS, such as a range query of Bathymetric Elevations, but the resulting geometric representation calculated by the DGGS server as a set of DGGS IDs and the efficiency of DGGS progressive transmission could not be utilized by the naïve client.

In other words, any client that requests a location from a DGGS server must understand the DGGS geometry it is receiving, just as it would understand traditional coordinate reference systems to place a polygon on a map or transform a coverage onto a globe. Until clients have DGGS knowledge – vis-à-vis a library that translates DGGS Zone IDs to coordinates – then the suite of OGC API Standards will have limited ability to support access to the full power offered by a DGGS fusion server.

13

DATA FUSED CLIENT 1

13 DATA FUSED CLIENT 1

The Data Fusion Client component was designed to ingest the various datasets (S-122 and others) served by the Data Fusion Servers developed throughout this Pilot. This component was demonstrated by Compusult.

13.1. Status Quo

Compusult has been continuously active in defining and validating new OGC standards and participating in many OGC Web Services testbeds and other projects since 1999.

Additionally, Compusult's Standards-based Commercial-Off-The-Shelf (SCOTS) Web Enterprise Suite (WES) software, GO Mobile app, and WES SensorHub Internet of Things (IoT) solution all incorporate and exploit many OGC open standards for extensive interoperability with other systems. WES is a suite of applications, tools, and middleware that have been successfully used to implement numerous enterprise geospatial data management projects.

During this pilot Compusult focused on the new API standards and how they can enhance the user experience when searching for geospatial data. This pilot allowed Compusult to look at new technologies to allow client-side rendering of feature data instead of having to go back to the server to retrieve images.

13.2. Technical Architecture

Compusult leveraged open-source software for this project to build a map-based web client that could communicate with OGC API Features and EDR.

Compusult developed a custom React Leaflet map-based web client for this pilot to communicate with OGC API Features and EDR. As seen on Figure 34, the client focuses on having the browser process the raw data from the OGC API implementations and having no knowledge of the service or the data it represented. By treating all services as equals, regardless of OGC API implementation, allowed the client to dynamically ingest and combine data from a dynamic list of services.

The individual services could be manually added to the client or selected from a catalog of known OGC API implementations, Web Map Servers, Web Map Tile Services, JSON documents, or user imports to create layers of related content. The client contacts the services directly, requesting only a subset of the data, which allows the client to interact with large data sets without creating a full copy of the data. This process produces responsive queries while removing concerns of storing large datasets as well as inconsistencies that could accrue between copied data and source data.
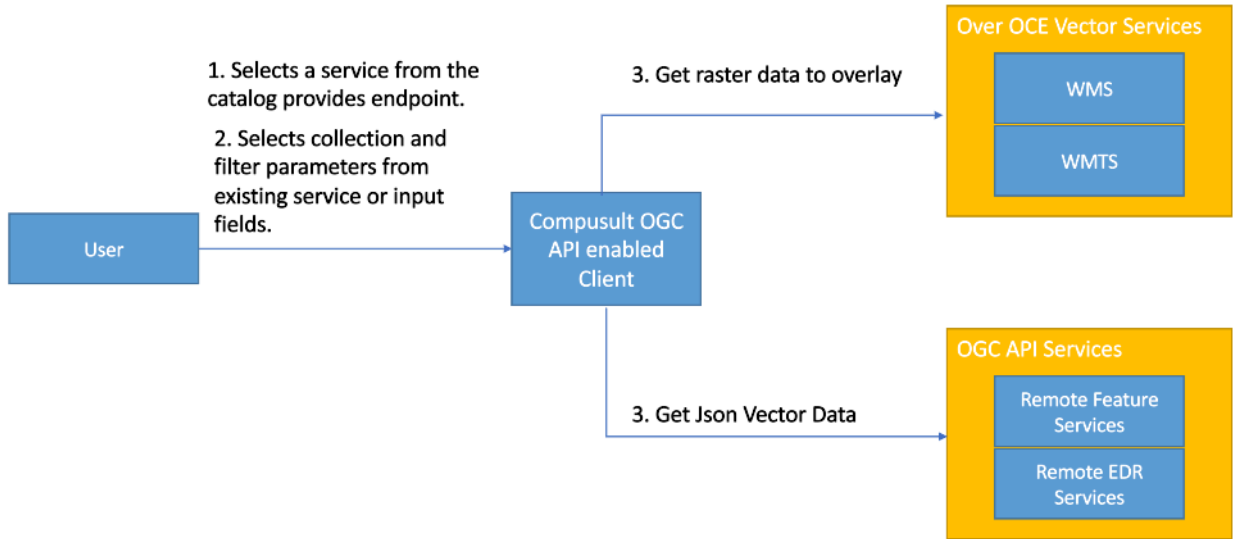
Figure 34 — Data Fused Client

Knowledge on how to build the web form comes from the service. The client makes a call to the service to determine what form values it provides. Once information is collected about how to execute the search, a request is built to perform a web request against the Feature server or Coverage Server. The data is returned in GeoJSON format containing the geometry and metadata for each feature which can then be used in a new search. This flow is shown in detail on Figure 35.

Data Fusion Server Work Flow
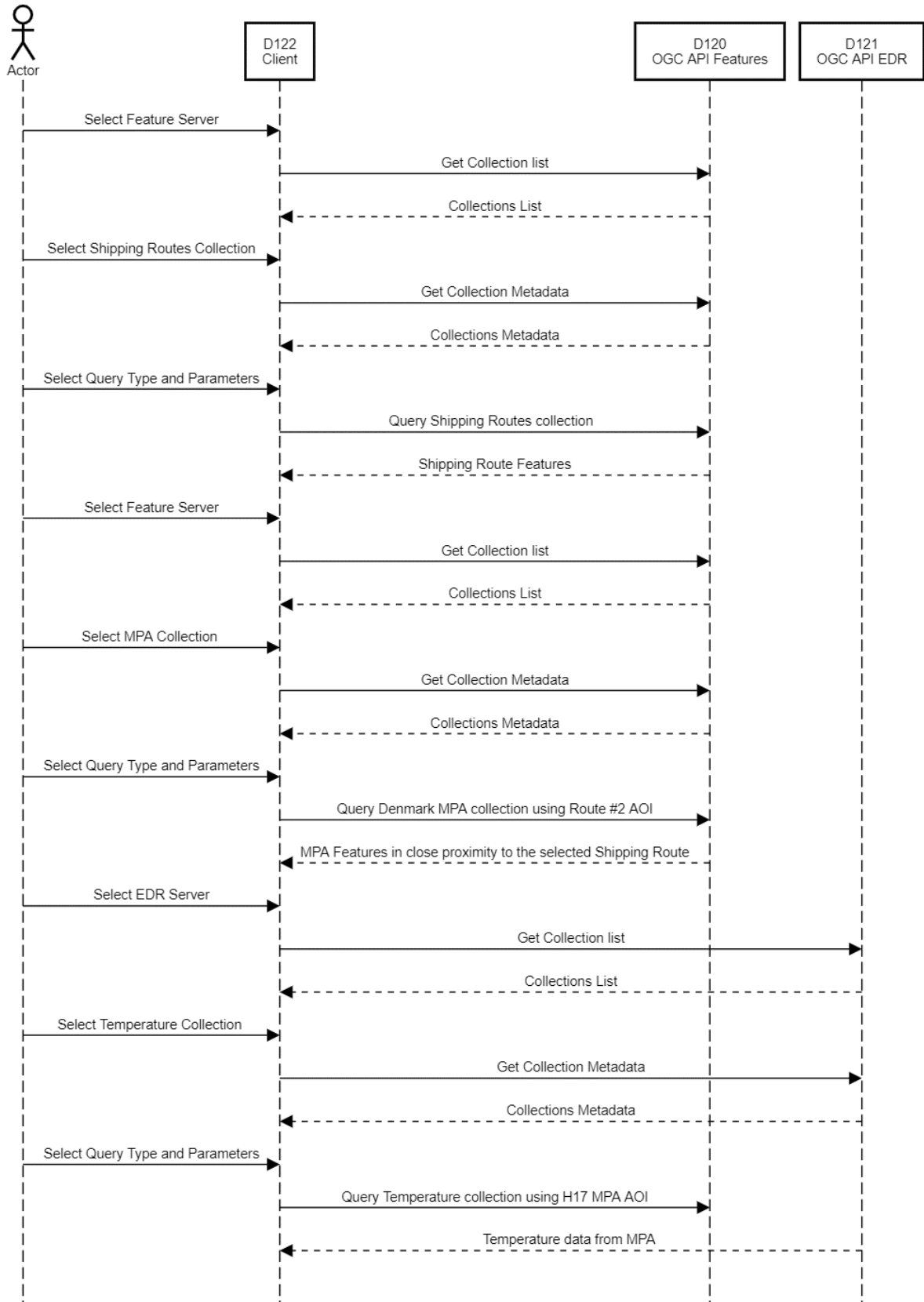
Figure 35 — Data Fused Client Workflow

The client combines both service types to not only allow the user to obtain the raw information but also to create aggregate data. For example, a user could search a feature service for an MPA and then use the resulting feature to query the coverage service for information such as water temperature, water quality, or counts from different endangered species.

13.3. Challenges and Lessons Learned

13.3.1. Use of GeoJSON

GeoJSON format provides many advantages with interoperability because of wide adoption and support in mapping software like ArcGIS, Leaflet, Cesium, Mapbox, and Google Maps. This allows us to natively render the resulting GeoJSON using Leaflet in our client with the combination of a selectable style provided by our client. The metadata provided in the GeoJSON is presented as a key value pair allowing for easy display of simple metadata. These simple values can be easily rendered in a table or used in a new search. The type and unit can be obtained from the service but the return of non standardized JSON objects such as `{name:"Joe Brown", locale:"en", organization:"foo"}` makes it difficult to know how to display the information to the user in a meaningful way such to allow searching those attributes.

CovJSON vs. GeoJSON: For an EDR API, Coverage JSON would represent the data better than GeoJSON. This would allow for data values to be more precise rather than a min, max and average value representing the whole MPA. CovJSON would provide the ability to efficiently produce heat maps of data using values such as temperature and elevation.

13.3.2. Features API vs. EDR API

Features and EDR APIs are utilized depending on the data backing the API. When you have distinctly identifiable entities such as ships, routes, lakes, zones, and MDR with attributes such as size, temperature, and owner, then OGC API — Features does a great job of providing a searchable list of entities. But Features APIs do not work well if the data cannot easily be mapped to identifiable entities. An example of data not working for a feature service is a GRIB file containing environmental data. On the other hand, a Feature service would be better for providing weather data for cities because the same data can be grouped into entities. Our client uses the Features API to derive data about entities such as MPAs, Ships, and Shipping Routes. The EDR API does not require the construct of grouping data into discoverable entities but instead provides an excellent way of accessing subsets of data about an area of interest.

13.3.3. Min, Max and Average data format

It is not possible to differentiate whether or not a Features API supports CQL queries from the metadata of the service/collection. Clients need to be able to know that a service supports CQL or any of the other parts of the OCG Features API from the metadata alone.

13.3.4. Complex Features

Searching Data from complex features complicates the filtering processes for the clients. A complex field may return JSON syntax such as '{name:"Joe Brown", locale:"en", organization:"foo"}'. This would require the client application to have knowledge of the response in order to build a specific request. Such syntax would make it difficult to search for features where Name = "Joe Brown" for example. Our recommendation would be to provide the queryable fields using a path structure such as user/name, user/locale and user/organization instead of a JSON object.

13.3.5. Multi-Geometry features and URL Length limitations in EDR API

The specific EDR API implementation that has been deployed for this pilot does not support searching by multi-geometries. However, the OGC API EDR Standard does provide examples of queries filtered using MULTIPOINT and MULTIPOLYGON geometries. This results in the client having to use a bounding box or polygon that fully encapsulates the multi-polygon features and thus some of the results that are returned are not pertinent to the Marine Protected Area as they are outside the multi-polygon coverage area.

Furthermore, `getItems` requests on OGC API Features implementations and `getArea` requests on EDR APIs have a limitation on URL length preventing building requests with multi-geometries as the URL length would exceed the maximum.

13.3.6. Feature Styling in Features and EDR API

OGC API Features and EDR implementations do not contain any style information. This might not be as simple as having the source service provide the style because different countries and organizations could have different standards or priorities when visualizing the data. Some possible solutions are listed below.

- Standardize the data fields and then indicate the type of data accosted with that standard on the collection level. Then the Client can use predefined styling rules based on the provided data type.

- Provide styling rules with the data for the client to use.

14

DATA FUSED CLIENT 2

14 DATA FUSED CLIENT 2

The Data Fusion Client component was designed to ingest the various datasets (S-122 and others) served by the Data Fusion Servers developed throughout this Pilot. This component was demonstrated by Pelagis.

14.1. Technical Architecture

The solution architecture is designed as an extension to the D101 client, also demonstrated by Pelagis, architecture by defining new service endpoints for the D120 and D121 Fusion Servers. These service endpoints represent the extended domain for marine protected features and marine environmental features, respectively.

The information graph model represented through the OpenSEA application service is extended to represent these domains as new feature services. The graph model resolves references to MPA features through the D120 Fusion Server using the OGC Features API. References to marine environmental features are resolved to D121 Fusion Server through the OGC API EDR implementation.
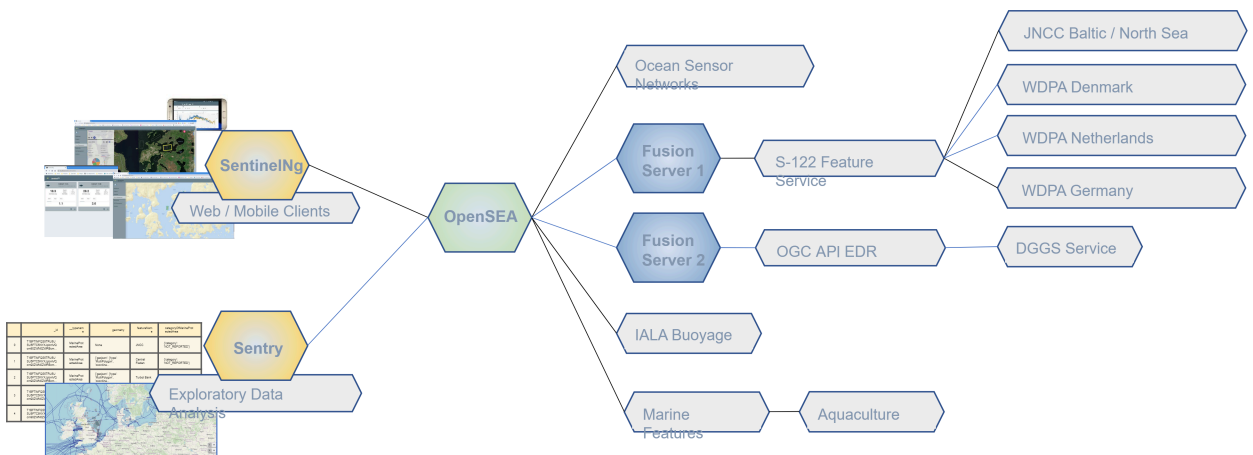
Figure 36 — D123 Client & Application Server Architecture

The focus of this stage was the integration of the D120 and D121 services as part of the extended federated information model for marine features. The scope of work addressed higher-value user scenarios across these respective domains and were designed to highlight the benefits of a federated marine spatial data infrastructure.

14.1.1. Scenarios

Use Case: As a user, I want to see the environmental information for each marine protected area within my authority

This scenario provides a view of the surface temperature information for each marine protected area within a specific source authority. The goal of this scenario is to demonstrate the capability of a federated marine SDI in support of changing environmental concerns.

The client issues the query to the OpenSEA application service. The query is resolved into an access plan for the D120 service to identify each MPA feature belonging to the source authority. For each of these MPA features a query is issued to the D121 service to retrieve the sea surface temperatures for each MPA coverage area.

Client query:

```
post( URI='https://…/ogcfmsdi/”, json='query = "query all_MPAs
($authcode: AuthorityCode) { marine_protected_areas (authCode: $authcode) { _
id featureName geometry{geojson} observations(phenomenon: $phenomenon) { …
on Measurement { value uom } } 'variables': {"authcode": 'DNK', "phenomenon":
'SST', "bbox": None}
```

Workflow: This scenario leverages the OGC Observations & Measurements (O&M) standard by modeling the relationship between the MPA features and recorded sea surface temperatures.
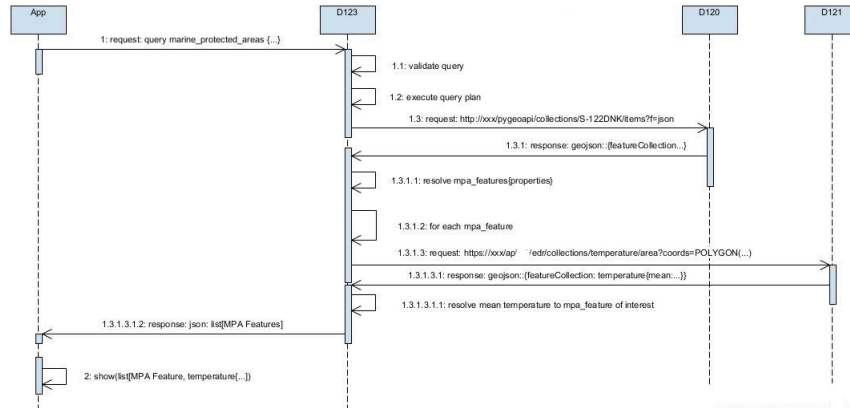
Figure 37 — D123 Sequence Diagram

The response to the client application allows the user to further analyze changing sea surface temperatures across the marine protected areas.

Figure 38 — D123 Marine Protected Area Environmental Information

Use Case: As a user, I want to see the vessel traffic for an agency relative to the network of marine protected areas

This scenario visualizes the vessel routes for an area of interest as they relate to a set of marine protected areas. This scenario extends the FMSDI architecture with a new service endpoint representing the Denmark Open Data portal for vessel traffic. Vessel routes for marine traffic relative to the MPA features of interest are combined with the D122 service endpoint to represent a federated query graph of MPA features and human activity for the Danish authority.

Client query:

```
post( URI='https://…/ogcfmsdi/&#8221;, json='query = "query vessel_routes
($authcode: AuthorityCode) { marine_protected_areas (authCode: $authcode) { _
id featureName iucn_category geometry{geojson} } vessel_routes ( authCode:
 $authcode, period: $period ) { _id MMSI geometry{geojson} } 'variables':
 {"authcode": 'DNK', "period": 'May 14, 2022', "bbox": None}
```

Workflow: This scenario queries against the D120 Fusion Server to retrieve the set of MPA features for Denmark. It then queries against the Denmark Open Data Portal to retrieve the set of ship plans for the date period of May 14, 2022. The ship plans are converted from the native.csv file of waypoints to a feature set of vessel routes, one route per MMSI, and provided in response to the client request as a set of features. As this is an expensive operation, the set of vessel routes is cached on the application server (D123) for subsequent requests.
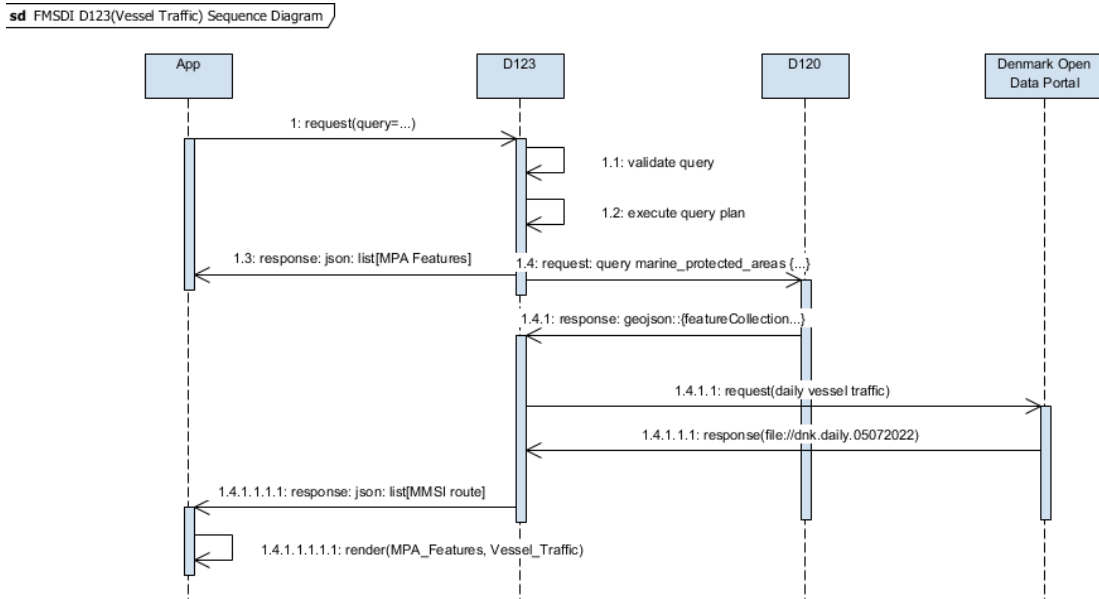
Figure 39 — D123 Sequence Diagram Vessel Traffic for Denmark
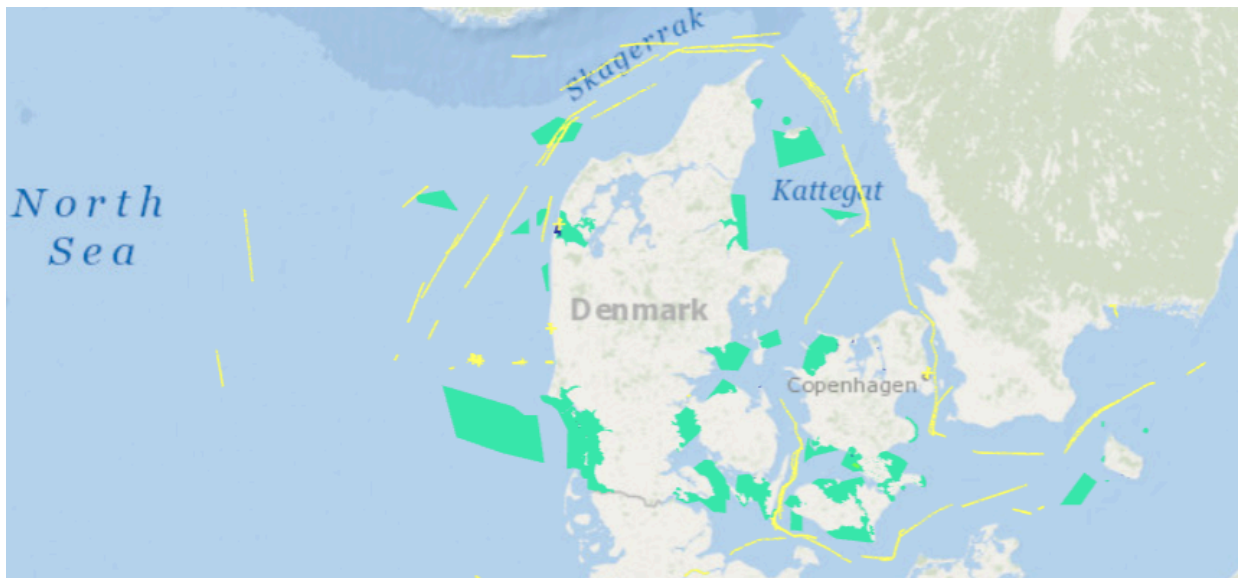
Figure 40 — D123 Vessel Traffic

The system identifies the MPA features managed by the source authority of the region and for each MPA coverage area, determines the set of ship plans whose routes were within the jurisdiction of the area.

Further analysis capabilities are available to determine the locality of vessel traffic to one or more MPA features based on proximity of the vessel trajectory or the likelihood of an unauthorized vessel traveling through an MPA feature.

14.2. Challenges and Lessons Learned

- **Tessellation level**: The D121 Fusion Server is backed by a Discrete Global Grid System (DGGS) and publishes environmental information through an interface conforming to the OGC API — EDR Standard. Although the implementation of the EDR specification was limited to *query by area* filters, the interface shows promise for further development. Beyond the core EDR query capabilities such as *query by trajectory*, the DGGS service could provide a refined query interface to allow the client to specify the tessellation level for an area of interest. The expected result would be a feature collection of gridded features tessellated to the requested level over the coverage area.

- **Temporal support**: Also of benefit would be support for temporal extents in which the client would provide both a spatial and temporal extent over which the DGGS service would provide aggregated datums.

- **MPA coverage area**: An MPA coverage area may be quite large. The nature of the D121 Fusion Server is to provide a single gridded feature representing the environmental information related to the area of interest. For large coverage areas, a single datum of temperature information (max, mean, min) is not very useful. It would be ideal if the DGGS capabilities were exposed through the EDR API to allow a client request to specify the degree of tessellation across the coverage area. This would provide a higher degree of resolution of the aggregated information for the area of interest.

15

TECHNOLOGY INTEGRATION EXPERIMENTS (TIES)

15 TECHNOLOGY INTEGRATION EXPERIMENTS (TIES)

The FMSDI Pilot Integration Experiments (TIEs) focused on the exchange of MPA data through OGC APIs. Each TIE explored under different circumstances the potential of OGC API and IHO standards with the objective of relaying MPA data.

15.1. TIE Summary Table

The following tables summarize the TIEs performed during the FMSDI Pilot during phases one and two, respectively.

Table 2 — MPA-Specific Scenarios

SERVER / CLIENT	D101 BALTIC/NORTH SEA CLIENT 1	D102 BALTIC/NORTH SEA CLIENT 2
D100 Baltic/North Sea Server	Clause 15.2	Clause 15.3

Table 3 — Fusion Scenarios

SERVER / CLIENT	D122 DATA FUSED CLIENT 1	D123 DATA FUSED CLIENT 2
D120 Data Fusion Server 1	Clause 15.4	Clause 15.6
D121 Data Fusion Server 2	Clause 15.5	

15.2. D101 Baltic/North Sea Client 1 for D100 Baltic/North Sea Server

This set of TIE tests focused on the query capabilities of the D100 Baltic/North Sea service endpoint. The set of integration tests ranged from simple queries against a set of MPA features through to complex requests involving MPA features and other related entities described by the S-100/S-122 standard.

Table 4 — D101 TIEs with D100

CAPABILITY	D100	D101	ROLE
Query	✓	✓	Verify the BNS service can provide the set of MPA features without constraint.
Query by unique identifier	✓	✓	Verify a specific MPA feature may be fetched by primary key
Query by name	✓	✓	Verify a specific MPA feature may be fetched by its unique name
Query by area of interest	✓	✓	Verify a set of MPA features may be fetched where each MPA feature is wholly or partially contained within an area of interest
Query by authority	✓	✓	Verify a set of MPA features may be fetched where each MPA feature is managed by a specific authority provided as a constraint
Query by property	✓	✓	Verify a set of MPA features may be fetched where each MPA feature satisfies a property constraint
Query by complex filter	X		Verify a set of MPA features may be fetched where each MPA feature satisfies the constraint of a complex filter
Query by temporal range	X		Verify the retrieval of MPA features based on their temporal properties.
Query by marine protected network	X		Verify the retrieval of MPA features based on a common 'network'

✓indicates the TIE test scenario was successfully completed + X indicates the feature was either limited or not available and as a result, not testable.

15.2.1. Query

This set of integration test cases validated the capability of the BNS service to provide a full set of MPA features with no constraints or filters applied.

Test Scenario: The BNS service was configured to provide a collection of MPA features. No filters (queryables) were provided by the client to constrain the set of MPA features expected in the response

Expected Result(s): The full set of MPA features for the Baltic/North Sea.

15.2.2. Query by unique identifier (UID)

This set of integration test cases validated the capability of the BNS service to provide a specific MPA feature by its unique identifier.

As per the S-122 specification:
IHO feature records must provide a unique world-wide identifier for each feature. The feature identifier is composed of the combination of the subfields 'agency', 'featureObjectIdentifier',

and 'featureIdentificationSubdivision' elements of the feature record. Features, information types, collection objects, meta features, and geometries (inline or external) are all required by the schema to have a gml:id attribute with a value that is unique within the dataset. The gml:id values must be used as the reference for the object from another object in the same dataset or another dataset.

Test Scenario: The S-122 information model defined a unique identifier as a combination of the subfields agency, featureObjectIdentifier, and featureIdentificationSubdivision. The query was configured to use the 'id' property of the MPA feature to fetch a specific MPA feature from the service endpoint.

Expected Result(s): At most one MPA feature that matched the identifier provided in the query.

15.2.3. Query by name

This set of integration test cases validated the capability of the BNS service to provide a specific MPA feature by 'name'.

As per the S-122 specification:
MPA features inherit the `featureName` property from the S-122 abstract type `FeatureType`. To accommodate multiple languages, the `featureName` is a complex type containing a list of named values. Each named value has the properties 'displayName'::`Boolean`, 'language'::`ISO-639-3`, and 'name'::`text`.

Test Scenario: The query was configured to use the 'name' property of the MPA feature to fetch a specific MPA feature from the service endpoint.

It was assumed that the MPA feature name was unique across the set of MPA features sourced from a common authority.

Expected Result(s): At most one MPA feature that matched the MPA feature name provided in the query.

15.2.4. Query by area of interest

This set of integration tests validated the capability of the server to provide a set of MPA features overlapping an area of interest. Most MPAs were located in the territorial waters of coastal states, where enforcement can be ensured. MPAs were also established in a state's exclusive economic zone and even within international waters. For example in 1999, Italy, France, and Monaco jointly established a cetacean sanctuary in the Ligurian Sea named the Pelagos Sanctuary for Mediterranean Marine Mammals. This sanctuary includes both national and international waters.

Test Scenario: A generic area of interest based on a polygon was provided as a filter to the BNS service request. Additional testing was used to verify an area of interest based on a complex polygon, a multi-polygon, and a bounding box representation.

Expected Result(s): The set of MPA features that overlap the area of interest.

15.2.5. Query by authority

This set of integration tests validated the capability of the server to provide a set of MPA features specific to an agency having jurisdiction over the marine protected area. S-122 products were based on data sources released by an appropriate MPA defining authority.

Test Scenario: The name of the authority was provided as a query filter to the BNS service.

Expected Result(s): All MPA features managed by the authority were provided in the query response.

15.2.6. Query by property

This set of test cases further refined the query filters based on simple properties of the MPA feature model such as the feature status.

Test Scenario: MPA feature properties were provided in the query filter.

Expected Result(s): MPA features would be retrieved from the BNS service based on the query property.

15.2.7. Query by complex filter

Verify that a set of MPA features may be fetched where each MPA feature satisfies the constraint of a complex filter.

Test Scenario: This scenario could not be completed as complex filters were not available for this set of TIE tests.

Expected Result(s): MPA features would be retrieved from the BNS service based on satisfying the temporal requirements of the query.

15.2.8. Query by temporal range

This test provisions MPA features based on the temporal properties of the feature. The S-122 specification models the active period of an MPA feature with a fixed date range (if permanent) or a periodic date range (if seasonal). In either case, it is important to verify that an MPA feature may be retrieved based on its active period to ensure the most recent version of the feature is retrieved if active or not retrieved in the event the feature is no longer active.

Background: S-122 datasets can only contain replacements, deletions, and additions of whole feature instances or information instances. This means that when a feature or information instance is updated, the new version must contain all the attributes of the old instance, including any inline spatial attributes (i.e., inline geometry), except those attributes that are being removed. Feature and information type instances are deleted without replacement by setting the

fixedDateRange.dateEnd attribute of the instance to the date of deletion, which will usually be the issue date of the update.

Test Scenario: This scenario could not be completed as temporal filters were not available for this set of TIE tests.

Expected Result(s): The set of MPA features satisfying the temporal range filter would be retrieved from the service endpoint. Of note is that the query provided the set of MPA features that were considered 'active' for the specified date range.

15.2.9. Query by marine protected network

Identify marine protected areas by their inclusion in a designated marine protected area network.

Background:
According to the World Wildlife Fund, a Marine Protected Area Network can be defined as "a collection of individual MPAs or reserves operating cooperatively and synergistically, at various spatial scales, and with a range of protection levels that are designed to meet objectives that a single reserve cannot achieve". Such a network can include several MPAs of different sizes, located in critical habitats, containing components of a particular habitat type or portions of different kinds of important habitats, and interconnected by the movement of animals and plant propagules.

Test Scenario: This scenario could not be completed as the concept of marine protected networks is not modeled by the S-122 specification.

Expected Result(s): A set of MPA features that belong to an identified marine protected network.

15.2.10. Connectivity

This set of TIE tests focused on issues that may arise from a sometimes-connected client environment.

Table 5 — Connectivity TIEs

ID	CAPABILITY	D100	D101	ROLE
	Well connected client	✓	✓	Identify issues that may arise in a client environment remotely connected to the D100 BNS Server
	Disconnected client	✓	✓	Identify issues that may arise in a disconnected web client environment

✓ *indicates the TIE test scenario was successfully completed.*
X indicates the feature was either limited or not available and as a result, not testable.

15.2.10.1. Web Client

This test case focused on the interaction between a well-connected web client and the D100 BNS Server. The design of the test case leveraged the OGC Features API to issue requests to the BNS server over http(s). Responses to the client would provide a feature collection for MPA features together with the requisite meta data as prescribed by the OpenAPI specification.

Test Scenario: The client issued a basic request to the D100 BNS Server to retrieve all MPA features within an area of interest. The server responded with the feature collection.

Expected Result(s): The full set of MPA features for the Baltic/North Sea.

15.2.10.2. Disconnected Client

This test case focused on the ability of the client application to persist MPA features locally when disconnected from the BNS Service.

Test Scenario: The client issued a basic request to the D100 BNS Server to retrieve MPA features within an area of interest. The server responded with the feature collection. The client then disconnected from the BNS service while maintaining its ability to view the feature collection locally.

Expected Result(s): The client could view the MPA feature collection locally with no affect to the behavior of the application environment.

15.3. D102 Baltic/North Sea Client 2 for D100 Baltic/North Sea Server

The D102 client was able to connect and interact with the various collections on the D100 BSNS server. As the D102 client represented the viewpoint of a vessel in a less-connected environment, the focus was on ingesting and querying one of the two feature collections that used bounding boxes to represent the original features. Bounding boxes were used as they were a simpler and lighter representation of the original features, but could still be useful to some end-users in less connected environments. The following table summarizes the tests performed between both clients as well as the expected and actual outcomes of them.

Table 6 — TIEs between D102 and D100

#	TEST	EXPECTED OUTCOME	ACTUAL OUTCOME
1	Connect to one of the feature collections on the D100 server and load in the data to the client	Connection is successful and all the data within the	Data is displayed in the client

#	TEST	EXPECTED OUTCOME	ACTUAL OUTCOME
		collection is displayed in the client	
2	The query from the client to the D100 server is confined to the bounding box of the buffered route.	Only features within the bounding box are returned to the client.	Does not work on the feature collections that use GeoJSON as a provider. Only works on Elasticsearch providers
3	The query from the client to the D100 server is limited to the buffered route polygon.	Only features within the buffered route polygon are returned to the client.	Ability not enabled on the server. Only bbox spatial queries are permitted.
4	The client only displays the MPA features that intersect with the buffered route	Only features that intersect the buffered route are displayed in the client	MPA bounding boxes that intersect the buffered route are displayed.
5	Document the time taken to run query of a dataset hosted in two different formats: a) the original features b) bounding boxes of the features	Bounding boxes of the dataset are returned much quicker than the original features	Bounding box collection returns features significantly quicker than the original data.
6	Using a simulated throttled network connection, document the time taken to run query of a dataset hosted in two different formats: a) the original features b) bounding boxes of the features	The overall time taken to display features will increase for both formats.	Bounding boxes of the dataset are returned much quicker than the original features. Queries take longer using a throttled network. Bounding box collection returns features significantly quicker than the original data.

The bounding box collections and their original equivalents were as follows.

- JNCC Bounding Boxes: http://35.176.64.149/pygeoapi/collections/S-122JNCC_ES_BB

- JNCC Original Data: http://35.176.64.149/pygeoapi/collections/S-122JNCC_ES

- WDPA Bounding Boxes: http://35.176.64.149/pygeoapi/collections/S-122WDPA_ES_BB

- WDPA Original Data: http://35.176.64.149/pygeoapi/collections/S-122WDPA_ES

At this stage, the D102 client is only connected to, and able to query features from, the S-122 WDPA bounding box collection as this contains a significantly larger number of MPA features (2000) compared to the JNCC bounding box collection (25).

The user can query these MPA features that are hosted on the D100 server by drawing a route in the D102 client. This route is then buffered by an arbitrary 5NM on the client and a bounding box of the buffered route is generated. This bounding box of the buffered route is then used as a bbox spatial query to request data from the WDPA bounding box collection on the D100 server, thus minimizing the amount of data being requested.

When the request comes back to the client it performs an intersection and only displays the bounding box features that intersect with the buffered route that the user created previously.

The user can click on the returned features and view the attributes of the data, such as the feature name, ID, category, and MPA status (Figure 41).
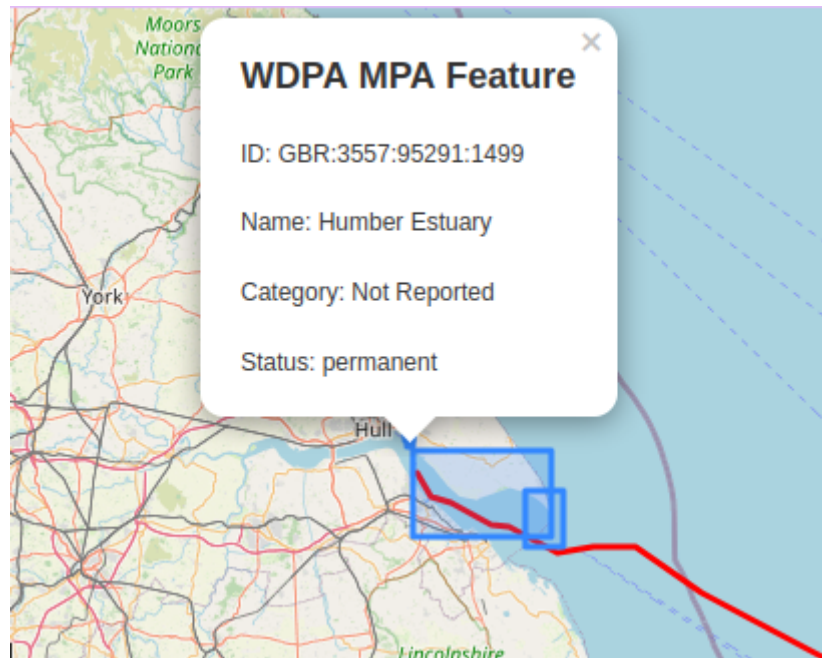
Figure 41 — Example of viewing feature attributes by clicking on them in the client.

To simulate a less connected environment some network emulation was undertaken by limiting the bandwidth, incorporating a delay, and simulating packet loss. The D102 client was still able to query the D100 server in these conditions, with the WDPA bounding box collection having a significant improvement in performance over the original WDPA features. This is almost certainly due to the small file size of the bounding box features, which are significantly smaller compared to the original MPA features. Using bounding boxes therefore offers good performance over low bandwidth networks.

Issues

- The bbox spatial query method only worked on the Elasticsearch providers on the D100 server. It appears from the documentation that v12 of pygeoapi doesn't support bbox queries for GeoJSON providers (https://docs.pygeoapi.io/en/0.12.0/data-publishing/ogcapi-features.html).

- Querying the original features even over a good network connection was rather slow. This is almost certainly due to the complexity of the MPA features that are situated in the littoral region as they follow the coastline and inland waterways. This means that there are potentially hundreds of vertices for each feature. Further enhancements to the client could include the following.

  - Further refining the query by selecting which data provider or agency (e.g., WDPA, JNCC, etc.) or MPA attribute (status, category, etc.) the user is interested in. This would further reduce the volume of data that is requested by the client from the server.

  - Linking the client to another bounding box feature collection on the D100 server.

- Providing the option for the user to select an MPA bounding box of interest and then retrieve the original data.

Further enhancements to the server could include the following.

- Implementing OGC API — Features — Part 3 to support CQL filtering so that the user can query using a polygon shape instead of a simple bbox, i.e., the buffered vessel route drawn by the user instead of using the bounding box of the buffered route.

- If using GeoJSON providers, ensure the server allows simple bbox spatial filters.

- Merging the different data providers into one collection.

- Enable some basic geoprocessing to clip the MPA features that fall within a given route and return the clipped MPA features to the client, minimizing the volume of data even further.

15.4. D122 Data Fusion Client 1 for D120 Data Fusion Server 1

The Data Fused Client 1 successfully retrieved MPA data from the Data Fused Client 1 (D120) which runs an API based on OGC API — Features.

The client first made a request to get the list of Collections available from the service.

Web Call: http://35.176.64.149:5000/collections

The JSON returned from this call, as seen on the code below allowed the client to parse and present a list of collections for the user to choose from. The links section provided the links to obtain the items and queryables for the collection.

```
{
    "collections": [
        {
            "links": [
                {
                    "type": "application/json",
                    "rel": "queryables",
                    "title": "Queryables for this collection as JSON",
                    "href": "http://35.176.64.149:5000/collections/S122DNK/quer
yables?f=json"
                },
                {
                    "type": "application/geo+json",
                    "rel": "items",
                    "title": "items as GeoJSON",
                    "href": "http://35.176.64.149:5000/collections/S122DNK/item
s?f=json"
                },
```

```
            ],
            "id": "S122DNK",
            "title": "Denmark MPA data (v2 model), Habitats",
            "description": "Denmark MPA data (v2 model), Habitats. Revised
data model",
            "keywords": [
                "Navigation",
                "Oceans",
                "MSDI",
                "Marine Protected Areas"
            ]
        }
    ]
}
```

Figure 42 shows a page generated in the client to show the list of available collections with the collection name, description, item type, and keywords being displayed to the user to allow them to decide.
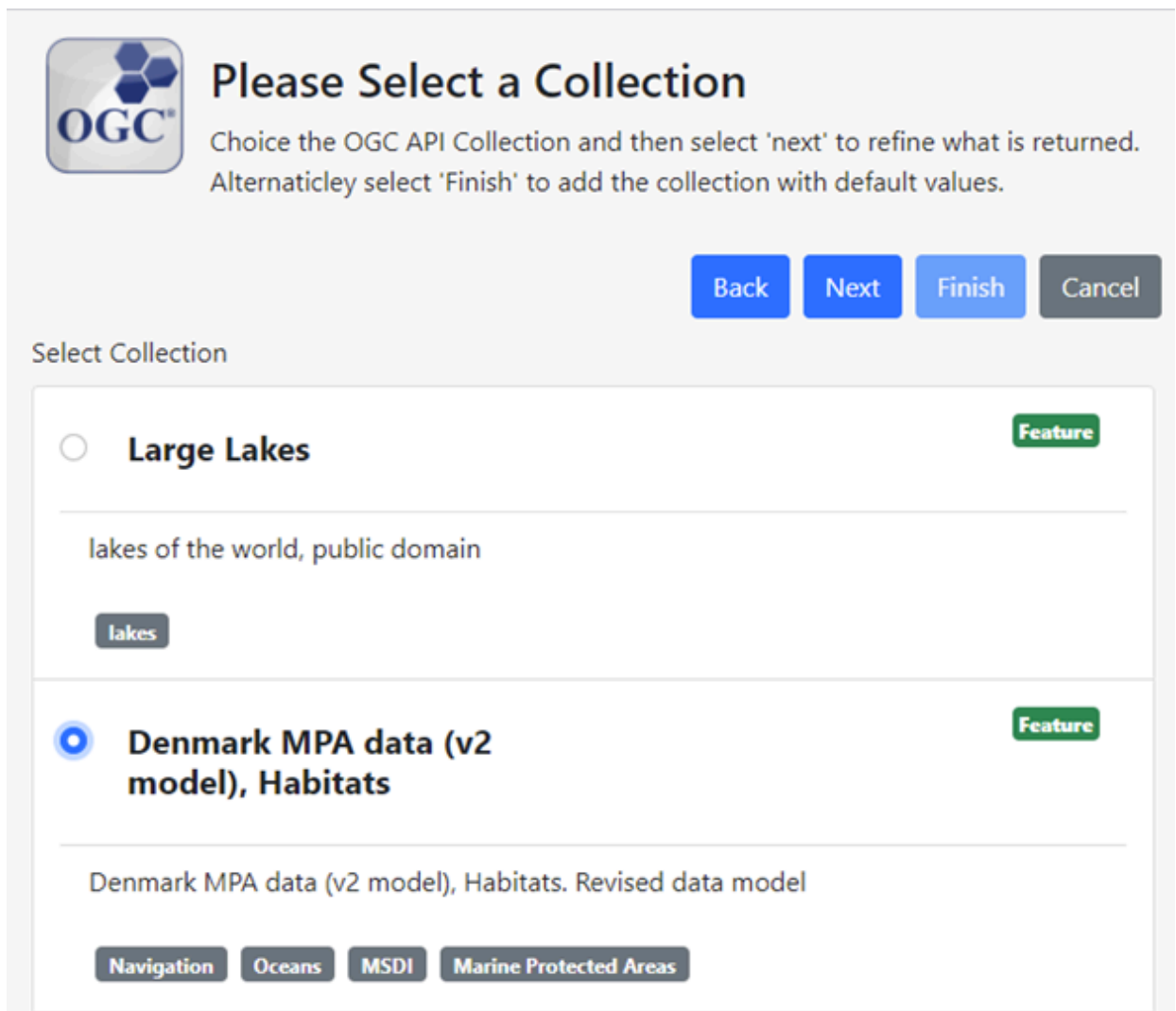
Figure 42 — Data Fused Client Displaying Available Collections

The next call being made from the client is to obtain a list of queryables in order to build the search criteria page. To do this the client uses the queryables link as obtained from the collections response.

Web Call: http://35.176.64.149:5000/collections/S122DNK/queryables

The JSON returned from this call, seen on the code below allows the client to parse and present a search criteria page to the user.

```
{
    "queryables": [
        {
            "queryable": "featureType",
            "type": "string"
        },
        {
            "queryable": "dateEnacted",
            "type": "string"
        },
        {
            "queryable": "foid",
            "type": "string"
        },
        {
            "queryable": "featureName",
            "type": "string"
        },
        {
            "queryable": "categoryOfMarineProtectedArea",
            "type": "string"
        },
        {
            "queryable": "status",
            "type": "string"
        }
    ]
}
```

Figure 43 shows the queryable attributes from the selected collection. This page is dynamically generated in the client by parsing the JSON response to the queryables call.

Figure 43 — Data Fused Client Displaying Search Parameters

An items query is then generated using the values input by the user to obtain the features from the server. In the sample web call below a user is searching using a bounding box for a limit of 10 items with the requested format of JSON.

Web call: http://35.176.64.149:5000/collections/S-122WDPA_DNK/items?bbox=4.658203,52.975642,18.017578,59.413082&f=json&limit=10

The response seen below is a list of features and associated geometries as well as the number of features matched and returned.

```
{
    "type": "FeatureCollection",
    "features": [
        {
            "type": "Feature",
            "id": "555639817",
            "properties": {
```

```
            "sourceIndication": {
                "country": "DNK"
            },
            "fixedDateRange": {
                "dateStart": "1984-01-01"
            },
            "featureName": {
                "name": "\u00d8le\u00e5ens Og L\u00e6s\u00e5ens Udmunding"
            },
            "jurisdiction": "national",
            "categoryOfMarineProtectedArea": "IUCN Category IV",
            "status": "permanent"
        },
        "geometry": {
            "type": "MultiPolygon",
            "coordinates": [
                [
                    [
                        [
                            15.003568956,
                            54.9980614040001
                        ],
                        [
                            15.002605616,
                            54.995384598
                        ],
                        ….
                    ],
                    [
                        [
                            [
                                14.9181869540001,
                                55.0184807810001
                            ],
                            [
                                14.9150687860001,
                                55.016360766
                            ],
                        ]
                    ]
                ]
            ]
        }
    }
],
"numberMatched": 309,
"numberReturned": 10,
"links": [
]
}
```

The available features, shown in the response above, are then presented to the user as shown in Figure 44.

555639817

- **Type:** Feature
- **Geometry Type:** MultiPolygon
- **sourceIndication:** {"country":"DNK"}
- **fixedDateRange:** {"dateStart":"1984-01-01"}
- **featureName:** {"name":"Øleåens Og Læsåens Udmunding"}
- **jurisdiction:** "national"
- **categoryOfMarineProtectedArea:** "IUCN Category IV"
- **status:** "permanent"

64596

- **Type:** Feature
- **Geometry Type:** MultiPolygon
- **sourceIndication:** {"country":"DNK"}
- **fixedDateRange:** {"dateStart":"1926-01-01"}
- **featureName:** {"name":"Nysø, Bøndernes Egehoved"}
- **jurisdiction:** "national"
- **categoryOfMarineProtectedArea:** "IUCN Category IV"
- **status:** "permanent"

67895

- **Type:** Feature
- **Geometry Type:** MultiPolygon
- **sourceIndication:** {"country":"DNK"}
- **fixedDateRange:** {"dateStart":"1977-01-01"}
- **featureName:** {"name":"Præsto Fjord, Jungshoved Nor, Ulvshale and Nyord"}
- **jurisdiction:** "national"
- **categoryOfMarineProtectedArea:** "Not Reported"
- **status:** "permanent"

Figure 44 — Data Fused Client Listing Features

The client then allows the users to select one of more of these features to show the geographic location on the map as seen on Figure 45. Through the use of the spatial area of the selected MPA, as shown in the image below, users can do a deep dive search using another service to find related information.
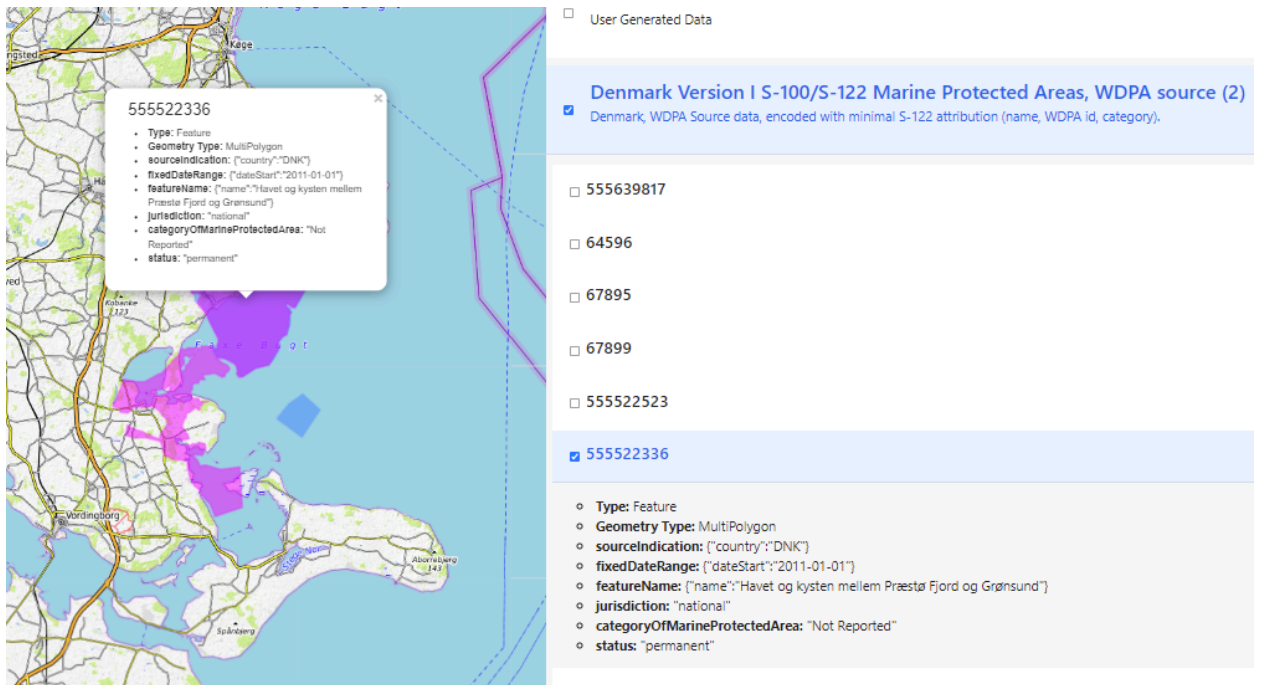
Figure 45 — Data Fused Client Displaying Features on a Map

15.5. D122 Data Fusion Client 1 for D121 Data Fusion Server 2

The Data Fused Client 1 successfully retrieved MPA data from the Data Fused Client 1 (D121) which runs an API based on OGC API — EDR.

The client made a request to get the list of Collections available from the OGC API service.

Web Call: https://digitalearthsolutions.com/api/v1/edr/collections

The response seen below is a list of available collections, available search parameters and the query links. Below is a section of the JSON response showing the temperature collection.

```
{
    "id": "temperature",
    "title": "Temperature",
    "description": "Dataset that contains the surface temperature of the entire
world",
    "links": [
      {
        "href": "https:digitalearthsolutions/api/v1/edr/collections",
        "rel": "self",
        "type": "application/json"
      }
    ],
    "extent": {
      "spatial": {
```

```
          "bbox": [
            -180.0,
            -90.0,
            180.0,
            90.0
          ],
          "crs": "4326"
        }
      },
      "parameter_names": {
        "Temperature": {
          "id": "Temperature",
          "observedProperty": {
            "label": "Temperature"
          },
          "type": "Parameter",
          "description": "This is the surface temperature",
          "data_type": "knFloat",
          "unit": {
            "label": "Kelvin",
            "symbol": {
              "value": "K"
            }
          }
        }
      },
      "data_queries": {
        "area": {
          "link": {
            "href": "http:34.95.36.182/api/v1/edr/collections/temperature/area",
            "rel": "data",
            "hreflang": "en",
            "variables": {
              "title": "Area query",
              "query_type": "area",
              "output_formats": [
                "GeoJSON"
              ],
              "default_output_format": "GeoJSON",
              "crs_details": {
                "crs": "4326",
                "wkt": "GEOGCS[\"WGS 84\",DATUM[\"WGS_1984\",SPHEROID[\"WGS
84\",6378137,298.257223563,AUTHORITY[\"EPSG\",\"7030\"]],AUTHORITY[\"EPSG
\",\"6326\"]],PRIMEM[\"Greenwich\",0,AUTHORITY[\"EPSG
\",\"8901\"]],UNIT[\"degree\",0.01745329251994328,AUTHORITY[\"EPSG
\",\"9122\"]],AUTHORITY[\"EPSG\",\"4326\"]]"
              }
            }
          }
        }
      }
    }
  }
}
```

This JSON response was then used by the client to generate the collection choices and subsequent search criteria page.

Below is an example of a multi-polygon search against the EDR service that resulted in error due to the length of the URL being too long with all the points included.

Web Request: https://digitalearthsolutions.com/api/v1/edr/collections/
temperature/area?coords=MULTIPOLYGON

Error Response:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN""http://www.w3.org/TR/html4/
strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Request URL Too Long</TITLE>
    <META HTTP-EQUIV="Content-Type" Content="text/html; charset=us-ascii"></
HEAD>
    <BODY><h2>Request URL Too Long</h2>
         <hr><p>HTTP Error 414. The request URL is too long.</p>
    </BODY>
</HTML>
```

Figure 46 shows the selection of the multi-polygon that caused the error shown above.
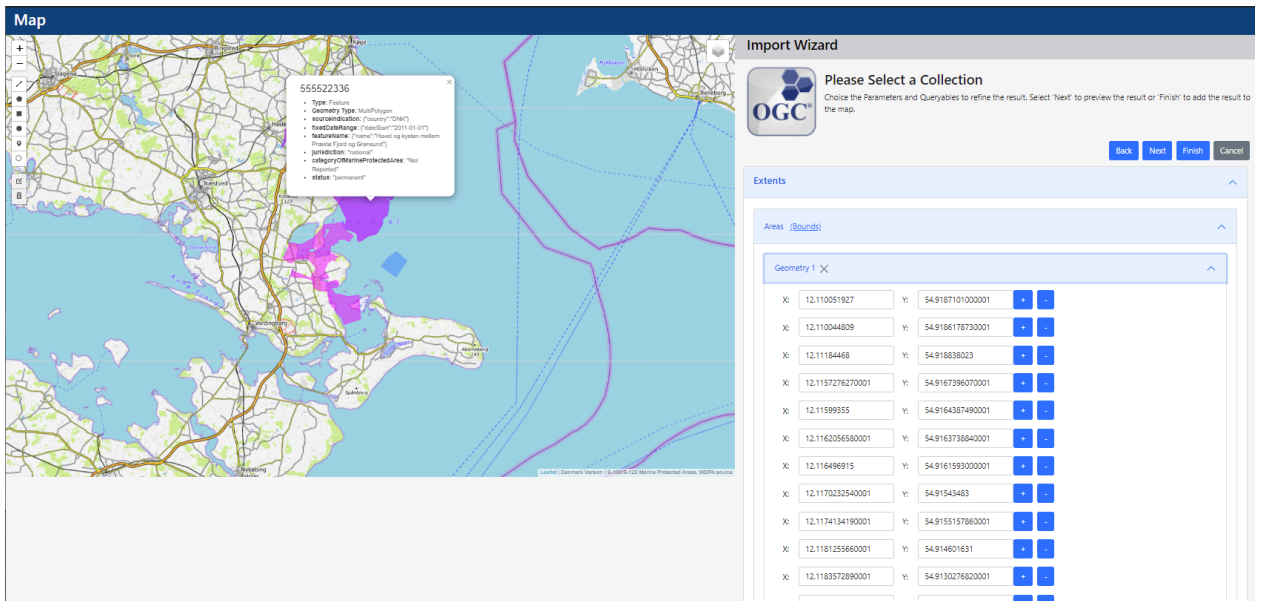
Figure 46 — Data Fused Client Displaying a Multi-Polygon

When the spatial area selected contains fewer points the results are shown to the user to allow data analysis on the area as shown in the image below. Figure 47 shows the minimum, maximum and average temperature data for the MPA titled *555522523*.
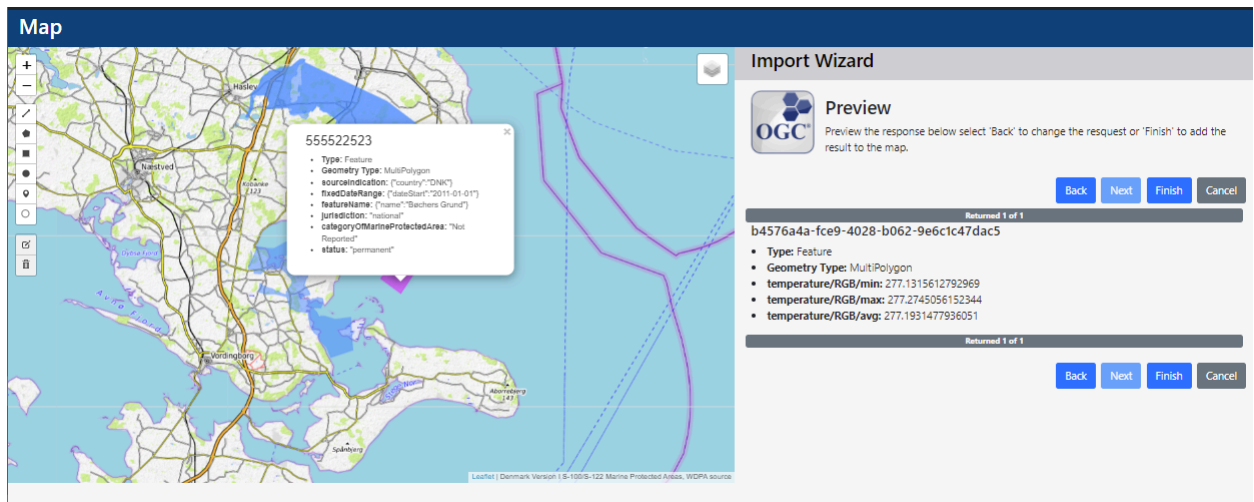
Figure 47 — Data Fused Client Displaying Feature Data

Using the flow described above users are able to request a subset of data from one fusion server collection and then take the results of that search to search another fusion server collection to provide related data. Repeating this process of using the data from the previous search to perform a new search allows for correlating data that is not directly related.

This workflow allows a user to exercise the scenario as outlined in the pilot project. Users can search the OGC API Features server for shipping routes. Using the spatial area representing the shipping routes users are then able to find intersecting MPAs. Users can select the MPA to use as the spatial area in a query against another service to find details regarding the MPA.

15.6. D123 Fusion Client for D120 Data Fusion Server 1 and D121 Data Fusion Server 2

This section represents a summary of technical integration points between the D123 Data Fused Client and the D120 Data Fusion Server 1 and D121 Data Fusion Server 2. Each set of tables identifies the role for each TIE test, whether the capability was provided by the either of the D12x Data Fusion Servers, and the status of each client implementation against the expected results.

These TIEs are designed to support various stakeholder scenarios based on the features made available through the D120 Data Fusion Server and the D121 Data Fusion Server. The D120 Fusion Server provides extended capabilities of the D100 Baltic/North Sea Server consistent with the S-122 standard. The D121 Fusion Server is designed to publish environmental information through an interface conforming to the OGC API for – EDR Standard for an area of interest consistent with the Baltic Sea and North Sea coverage area.

15.6.1. Scope of Work

The scope of this exercise focuses primarily on the integration requirements associated with the D120 and D121 Fusion Servers. The D120 Data Fusion Server provides the core capabilities prescribed by the D100 Baltic/North Sea Server as well as extended capabilities based on additional data providers and features. The D121 Data Fusion Server provides environmental data appropriate for the coverage areas of the Baltic Sea and North Sea.

The TIE tables for the D101 client are applied to the D120 Fusion Server to regress the expected capabilities of the S-121 Feature service. Additional TIE tests are provided to stress these additional capabilities of the D120 Fusion Server. The set of tests are dependent on the data made available and, where applicable, are used as input to the set of TIE tests provided to stress the capabilities of the D121 Fusion Server.

For the purpose of this work item, the D120 Fusion Server is considered the master repository of IHO and MPA features. The D121 Fusion Server represents environmental features related to the IHO and MPA features provided by the D120 Fusion Server. In this way, the general approach was to query for a set of IHO and MPA features from the D120 Fusion Server and for each feature (or set of features), query for the associated environmental features from the D121 Fusion Server. This approach supports the development of the extended stakeholder scenarios identified earlier.

15.6.2. Summary TIE table

This set of TIE tests focuses on the query capabilities across the D120 and D121 Fusion Servers. The set of integration tests range from simple queries against a set of MPA features through to complex requests requiring the use of a federated model for MPA features and environmental features.

Table 7 — D123 Queries

ID	CAPABILITY	D123	D120	D121	ROLE
123.2.1	Query (D120)	✓	✓		Verify the core query capabilities for D120 Data Fusion Server as per the TIE tables for D100 and D101.
123.2.2	Query by area: polygon	✓		✓	Verify the core query capabilities for the D121 Data Fusion Server based on MPA feature geometry
123.2.3	Query by area: bounding box			X	Verify the environmental aggregate information for an area of interest determined by a bounding box
123.2.4	Query by area: collection of polygons			X	Verify that a discrete set of features are returned where each feature represents the environmental aggregate information for one specific MPA in a collection of polygons.
123.2.5	Query by temporal range			X	Verify the aggregate environmental information is determined over a temporal period for a specific area of interest

15.6.2.1. Query

This test suite is based on the test scenarios identified in the TIE tables for the D100 BNS service and the D101 client. The D120 Data Fusion service provides extended capabilities to the D100 BNS service and, as such, is regressed in terms of features and functionality.

Test Scenario Application of the D100 client test scenarios to the D120 Fusion Server 1 service endpoint.

Expected Result(s) As per D101 Client TIE tables.

15.6.2.2. Query by MPA Feature Geometry

This test suite is used to verify the capability of the D121 service endpoint to query for environmental features based on the coverage area for an MPA feature. The feature's geometry, which may be represented as either a Polygon or Multi-polygon, is passed to the query filter of the EDR request to retrieve the aggregated environmental information for the area of interest.

Test Scenario: The D121 service endpoint is configured to provide features based on the OGC — EDR standard and supports the filter 'area' and a value of type Polygon or MultiPolygon.

Expected Result(s): Environmental features providing aggregate temperature information for the area of interest is returned. The features provide the max, min, and average temperature values for the coverage area identified by the Polygon.

15.6.2.3. Query by bounding box

This test suite is designed to highlight the features of a Discrete Global Grid System backed through the use of the OGC EDR API.

Test Scenario: The D121 service endpoint is configured to provide features based on the OGC EDR API and supports the 'bbox' filter.

Expected Result(s): Environmental features providing aggregate temperature information tessellated over the area of interest. The features provide the max, min, and average temperature values for a discrete grid over the coverage area of the bounding box.

CHALLENGES, LESSONS LEARNED AND RECOMMENDATIONS

16 CHALLENGES, LESSONS LEARNED AND RECOMMENDATIONS

The development, testing, and demonstrations carried out throughout this Pilot provided lessons learned for all of the Pilot's participants. While many chapters include lessons learned for its corresponding component, this chapter summarizes and groups them for easier reference.

16.1. RFI Lessons Learned

The need for international collaboration in the FMSDI is prominent. Driven by global expectations or guidance, a regional approach for the FMSDI may be best; based on regional partners operating under a common goal, federating data & services from their individual and national organizations, and utilizing the same open standards for releasing their similar themes of data. As these regional MSDIs become established, they can coordinate with neighboring regions to ensure interoperability and share best practices for developing the MSDI.

Another observation was the need for interoperability between IHO and OGC standards. By implementing interoperability between OGC and IHO, it is possible to make more IHO S-100 datasets available to the public.

Use-cases for interoperability between OGC and IHO S-100 compliant products, datasets, land/sea interfaces, and marine protected areas from neighboring countries might be some of the topics of interest.

16.2. OGC Standards

16.2.1. Using OGC API — Features to Serve MPA Data

API extracts data from the source: The most attractive element of the API model for data producers, particularly of MPA data, is the retrieval from the authoritative source of the data. This poses questions in the navigational context, most notably: How can they be assured that data is being used in the correct context; and should any attempts be made to standardize data import to the bridge in such API formats?

Filter/query retrieval: The other major changes that come with API distribution of data is the ease of automation by client services, the format-neutrality of such APIs, and the fine control over retrieval which is not present in file-based encodings, such as the native S-100 encodings in Part 10. This allows for simple filters on data fields, compound queries (using AND and OR

conjunctions), and spatial queries (e.g., `WITHIN`, `INTERSECTS`) as well as simpler queries against bounding boxes.

16.2.2. Accessing MPA Data Through OGC API — Features

- **Collections & Source Authorities:** The BNS service endpoint exposed the MPA features through the /collections endpoint. Each /collection represented a set of features provided through a specific authority. This required the client library to iterate over all /collection endpoints to query for all MPA features. This proved to be an expensive operation from a computing perspective.

- **Security:** The issue of security (authentication and authorization) has not been explored to its full extent. Although the client may authenticate to the BNS service to allow its usage, there does not appear to be a formal means to restrict the authorization of the client to use a particular service capability or collections endpoint.

16.2.3. Spatial Filtering Using a Bounding Box Query

The `bbox` spatial filter, which is specified in the OGC API — Features — Part 1 standard, is a useful method to query a server and reduce the volume of data returned to the client. However, using this method often covers a significantly larger geographic area than what is required, especially if the user is only interested in data that intersects a line, in this case a route. The `bbox` spatial filter can therefore return unnecessary data back to the client which can be a challenge for users operating in low-connectivity environments where bandwidth is at a premium. A more efficient method of performing a spatial filter would be to use the functionality of CQL2 in creating the spatial filter, as described in OGC API — Features — Part 3. This enhanced functionality is recommended for future server implementations of the OGC API.

16.2.4. Using Bounding Boxes to Represent Features in DDIL

MPA features can be complex in their shape, especially in the littoral regions. This impacts the number of vertices that a particular feature has, and significantly increases the file size of the feature, making them unsuitable for DDIL environments. Therefore, creating a DDIL twin was proposed where the MPA shape needed some simplification. Bounding boxes were therefore created for some of the feature collections to reduce the complexity and size of the MPAs.

There were some issues identified, especially with these earlier instances of bounding box data collections. One issue was that some of the MPAs were multipart features, i.e., features that have the same attributes but consist of numerous individual features but were represented by a single bounding box. It needs to be ensured that when creating the bounding boxes, each individual feature has its own bounding box. Another issue was the bounding box itself, which frequently had large areas of empty space due to the shape of the MPA feature it represented. It is recommended that an alternative type of minimum bounding geometry, such as a convex hull, could be used to minimize these areas of empty space.

16.2.5. EDR API and DGGS

While the EDR API has been shown in the pilot project to support a naïve client with the tools to successfully "explore" DGGS data by requesting statistical summaries of aggregated fused data values for a particular area of interest – polygon, bounding box etc. – it would not give the client the capability to "search" for a location that matched a criterion.

The Common Query Language would be a convenient method of searching by filtering data values in a DGGS – such as a range query of Bathymetric Elevations — but the resulting geometric representation calculated by the DGGS server as a set of DGGS IDs and the efficiency of DGGS progressive transmission could not be utilized by the naïve client.

In other words, any client that requests a location from a DGGS server must understand the DGGS geometry it is receiving. Until clients have DGGS knowledge – vis-à-vis a library that translates DGGS Zone IDs to coordinates – then the suite of OGC API Standards will have limited ability to access the full power offered by a DGGS fusion server.

16.2.6. Features API vs. EDR API

Features and EDR services are utilized depending on the data backing the service. When there are distinctly identifiable entities like ships, routes, lakes, zones, and MDR with attributes such as size, temperature, and owner, then the Features API does a great job of providing a searchable list of entities. But the Features API does not work well if the data can't easily be mapped to identifiable entities. An example of data not working for a feature service is a *GRIB file* containing environmental data. On the other hand, a Features API would be better providing weather data for cities because the same data can be grouped into entities. The D122 client used the Features API to derive data about entities such as MPAs, Ships, and Shipping routes. The EDR API did not require the construct of grouping data into discoverable entities but instead provided a great way of accessing subsets of data about an area of interest.

16.2.7. CQL Support in Service Metadata

It is not possible to differentiate whether or not a Features API supports CQL queries from the metadata of the service/collection. Clients need to be able to know that a service supports CQL or any of the other parts of the OGC API — Features standard from the metadata alone.

16.2.8. Filtering Complex Features

Searching Data from complex features complicates the filtering processes for the clients. A complex field may return JSON syntax such as {`name:"Joe Brown"`, `locale:"en"`, `organization:"foo"`}. This requires the client application to have knowledge of the response in order to build a specific request. Such syntax would make it difficult to search for features where Name = "Joe Brown" for example. The recommendation is to provide the queryable fields

using a path structure such as user/name, user/locale, and user/ organization instead of a JSON object.

16.2.9. Multi-Geometry features and URL Length limitations in EDR API

The specific EDR API implementation that has been deployed for this pilot doesn't support searching by multi-geometries. However, the OGC API EDR Standard does provide examples of queries filtered using MULTIPOINT and MULTIPOLYGON geometries. This results in the client having to use a bounding box or polygon that fully encapsulates the multi-polygon features and thus some of the results that are returned are not pertinent to the Marine Protected Area as they are outside the multi-polygon coverage area.

Furthermore, `getItems` requests on Feature APIs and `getArea` requests on EDR APIs have a limitation on URL length preventing building requests with multigeometries as the URL length would exceed the maximum.

16.2.10. Feature Styling in Features and EDR API

OGC API Features and EDR Services do not contain any style information. This might not be as simple as having the source service provide the style because different countries and organizations could have different standards or priorities when visualizing the data. Some possible solutions are listed below.

- Standardize the data fields and then indicate the type of data accosted with that standard on the collection level. Then the Client can use predefined styling rules based on the provided data type.

- Provide styling rules with the data for the client to use.

16.3. IHO Standards

16.3.1. The S-100/S-122 Model

The current S-122 model **is fairly basic** in terms of its representation of MPA data. It allows for a single feature (Marine Protected Area). The model clearly shows how MPAs (implemented with curve or surface geometry) have a single mandatory attribute representing the IUCN category, a simple enumeration for jurisdiction, status, and restrictions which may be in place for the MPA. Restrictions, in this case, are purely maritime in nature and profiled from the IHO registry `codelist`.

This encoding, while a good fit for maritime use cases, **does not currently reflect the broader application of MPAs in different geospatial agencies and the richer attribution required for those uses**. The proposed amendments to the data model are an initial proposal which need

consideration by an applicable IHO working group, although, there is a broader question to answer in terms of how IHO itself should approach the design/modeling of such product specifications – should MSDIWG manage its own domain or should it extend those features currently within the registry and maritime domain?

A number of proposed suggestions have been implemented in the model used for the server.

1. Added three new values to `categoryOfMarineProtectedArea` (this is the primary designation — the IUCN category for any MPA). This allows for Not Includes, Not specified, and Unknown – it was noted that many MPAs in the source datasets simply do not have IUCN categories associated with them.

2. Added complex attribute to record other designations (like WDPA, HELCOM, and JNCC) also has a jurisdiction for recording at what level the designation exists. Designation records the scheme and the categorization within the scheme.

3. Added a dimensions complex attribute to record the calculated dimensions of the MPA. These are often used for reporting. Includes unit of measure (needs harmonization with registry)

4. Added enactment date (mandatory) and update date (optional) to all MPAs. This could equally be termed `validFrom` date but enactment seems to be used more frequently. There is a distinct difference between enactment dates and start/end date.

5. New information representing Management Plans. This has a `planStatus` attribute (currently only draft but `planStatus` needs more values added). New association added between MPA and management plan. Many MPAs have management plans associated via URL.

6. All Feature Types now have multiple identifiers. These can be from different schemes so they are a complex attribute with a scheme designation, a value, and a jurisdiction as many identifiers are set at different levels. Identifiers also have a `textContent` attribute to hold arbitrary textual data about the identifier.

7. `producerCode` added as a simple attribute to Agency. This is the code used to identify them — it could be part of dataset metadata but it needs to be at a feature level as multiple agencies could exist within a single dataset.

8. Added regional to jurisdiction. There is a potential to add local as well.

16.3.2. Client Perspective of S-122

- **MPA Feature ID**: As per the S-122 specification, the feature identifier is a 'text' string composed of the feature subfields. It is assumed that this text string is stored as a UTF-8 character string. It is not clear whether this identifier is 'universally persistent', meaning that it will never change for the lifetime of the MPA feature which may, itself, be updated.

- **MPA Feature Name**: Managing the MPA feature name as a complex type makes it very difficult to manage queries based on the well-known name of a marine protected area.

To search for a specific MPA feature by name, the client needs to manage the language code associated with the name and assume that the server will search for all MPA features based on indexing into each feature name. This may be an expensive operation if no specialized indices are configured against the set of MPA feature names.

- **Authority Names**: The authority is represented as a `featureName` in the S-100 model and is affected by the same naming convention issues identified in previous works. The client application must have prior knowledge of the locale-specific authority name in order to provide this information as part of the query filter. This approach could benefit from the previous suggestion of using the IALA MRN naming convention to identify each source authority of an MPA feature.

- **Marine protected area networks**: The IHO S-121 specification does not address the concept of marine protected area networks. Although the specification does loosely relate MPA features based on the respective source authority, there is no capability to model the 'synergistic' properties of the MPA network and its application toward a common objective.

16.3.3. Use of Bounding Boxes by the S-122 Product Specification

The S-122 product specification does support bounding boxes for individual features, however the part 10b GML encoding does not currently specify their use. At this time the workaround is to create them using GIS software applications (ArcGIS, QGIS, etc.) using some basic geoprocessing tools. While achievable for this pilot, this is not a long-term solution and presents a challenge if bounding boxes were to be used more widely. Surveying the wider community is suggested to determine whether having bounding boxes for individual features would be useful, and if so, to amend the GML encoding to specify their use.

16.4. GeoJSON

16.4.1. Challenges With The GeoJSON Encoding

During the Pilot, being able to encode S-122 data in GeoJSON coherently and consistently from the S-122 data was a major need. The implementation of the OGC API Features service in the project relied upon an encoding of data in GeoJSON, replacing the exiting GML implementation which is currently part of S-100 edition 4.0.0. The GML implementation has been substantially revised for edition 5.0.0 of S-100, previous implementations having suffered from mismatches between GML and feature catalogue structures. In S-100 the feature catalogue represents the single definition of the data structure of a product specification and binds entities drawn together from the IHO geospatial registry.

Product development is advanced for many product specifications now and the IHO's NIPWG group oversees the creation of many of these. All product specifications (even gridded data) maintain a feature catalogue. Those with specific symbolization requirements also maintain an

S-100 specific portrayal catalogue. Each product specification also contains a default encoding, normally drawn from the three included in S-100 Part 10 (a, b, or c). These are:

1. Part 10a – ISO8211;

2. Part 10b – Geographic Markup Language (GML); and

3. Part 10c – HDF5.

The use of GeoJSON as an encoding is not currently part of S-100 itself. However, its ubiquity as a format for exchange of geospatial data raises the possibility of its use for modeling S-100 General Feature Model (GFM) data.

16.4.2. Use of GeoJSON

GeoJSON format provides many advantages with interoperability because of wide adoption and support in mapping software like ArcGIS, Leaflet, Cesium, Mapbox, and Google Maps. This allows us to natively render the resulting GeoJSON using Leaflet in our client with combination of a selectable style provided by our client. The metadata provided in the GeoJSON is presented as a key value pair allowing for easy display of simple metadata. These simple values can be easily rendered in a table or used in a new search. The type and unit can be obtained from the service but the return of non-standardized JSON objects such as {name:"Joe Brown", locale:"en", organization:"foo"} makes it difficult to know how to display the information to the user in a meaningful way such as to allow searching those attributes.

CovJSON vs. GeoJSON: For an EDR API, Coverage JSON would represent the data better than GeoJSON. This would allow for data values to be more precise rather than a min, max, and average value representing the whole MPA. CovJSON would provide the ability to efficiently produce heat maps of data using values such as temperature and elevation.

17

FUTURE WORKS

17 FUTURE WORKS

Future work should build upon the findings that emerged from the development and testing of these components and answer questions that were left out of its scope. While many chapters include future work recommendations, this chapter summarizes and groups them for easier reference.

17.1. RFI Future Work

In the context of future work, the links to portals and datasets provided through the RFI will be used in future FMSDI phases. Also, due to the responses received in the RFI, a somewhat more refined questionnaire could be devised for future RFIs. Another future work may be building a stakeholder community of data users, producers, and enablers to expand this effort and collaborate on the finding of the RFI.

17.2. OGC API Standards

17.2.1. Disconnected, degraded, intermittent, limited (DDIL) Environments

The client application was designed to store the set of feature collections locally in a feature cache. This cache was used to look up features based on the reuse of the query by the application. Given a disconnected session, additional queries relative to the feature collection may only be used if cached locally.

Further investigation is required on how to optimize the retrieval and storage of MPA feature collections as a GeoPackage using a supported OGC file encoding.

17.2.2. Further Enhancing MPA Filters

There is a need to further develop the client alongside a server implementing OGC API Features — Part — 3 for additional filtering to reduce the amount of data requested from the server. An example is giving the user the option to filter the data using the MPA metadata. This would allow the user to filter the data based on what is important for their needs, which may change over time depending on the geographic location, type of mission, or other requirement. Use of this enhanced filtering will allow the user to only obtain pertinent data and help to further reduce the amount of data returned to the client

Another example would be to use a more advanced spatial filter due to the limitations with using bounding boxes as spatial filters observed during this Pilot. Future work should test accessing

MPA data through APIs implementing advanced filtering capabilities such as the ones provided by OGC API — Features — Part 3. This would further explore whether the approach of using bounding boxes as a DDIL twin to represent the MPA features would be a viable option. Some questions to consider if using OGC API — Features — Part 3 are listed below.

- Would the DDIL twin approach be needed if the data is filtered to a smaller geographical area?

- Would this approach be suitable for the users of S-122 data if a server only implements Part 1 of the OGC API Features?

17.2.3. Using Vector Tiles

Vector tiles enable the delivery of data in pre-defined tiles which enables small amounts of data to be delivered to the client and has been previously proven to work in DDIL environments. Some of the potential benefits of vector tiles are:

- Efficiently storing, delivering and visualizing vector data from large datasets (such as S-122);

- Varying levels of data 'resolution';

- Efficient caching of data;

- Providing clients with a hierarchy of available data, while awaiting requests for higher resolution tiles; and

- Using established techniques and APIs (OGC API – Tiles, OGC API — Features, OGC WMTS)

Due to the many benefits that vector tiles offer, especially for users operating in a DDIL environment, future work should explore using vector tiles for MPA data.

17.2.4. Explore Potential Solutions to Challenges With DGGS

Sufficient capabilities exist within the new and emerging suite of OGC API Standards to take advantage of a DGGS to fuse heterogenous data values and aggregate the cells to present as statistical summaries characterizing any area of interest defined by a EDR Query Type. However, the inverse problem (i.e., a `Select Where` Query such as CQL Filter Query, a Range Query, a Boolean Operation, or other criteria where the result is a geographic location) requires an efficient method of encoding and decoding DGGS cell location by the client application where the general case is millions of DGGS cells as pixels of information tiled in an hierarchy. Because of this, future work should address the challenge of representing the unique DGGS geometric representation sufficient for a naïve client application, perhaps through a software library or service, and/or as a product of the DGGS API, or other innovation.

While the D121 Fusion Server was implemented with a DGGS, publishing environmental information through the OGC API — EDR standard was limited to query by area filters. The interface also shows promise for further development.

- **Tessellation level**: Beyond the core EDR query capabilities such as *query by position*, *radius*, *trajectory*, *bounding box*, etc., **the DGGS service could provide a refined query interface to allow the client to specify the tessellation level for an area of interest**. The expected result would be a feature collection of gridded features tessellated to the requested level over the coverage area which might not exceed the *"pixelled"* coverage of a typical DGGS but provide useful coarse level data the client could utilize. This would likely require the EDR specification and other such standards to include the concept of a spatial resolution whenever constructing a spatial query.

- **Temporal support**: Also of benefit would be **support for temporal extents** in which the client would provide both a spatial and temporal extent over which the DGGS service would provide aggregated datums.

17.3. IHO Standards

17.3.1. S-122 and ISO 19152

More work is required to map examples and test the application of ISO 19152 for some providers. There is also the question of what to do with the existing structures in S-122 and the numerous other NIPWG/IHO product specifications using Authority, Restrictions, who should implement these Responsibilities as types of restrictions, among others. This potentially supports the idea of S-122 (or a future MPA product spec) being under the domain of an MSDI body, rather than attempting to co-exist wholly with maritime or navigational products. However, this debate is outside the scope of the current OGC project. The project raises the basic assertion that ISO 19152 provides a richer structure than the one currently implemented and may have application for administrations with particular mandates to use such standards. Using ISO 19152 would help with more complex restrictions other than those covered in the current enumerations.

17.3.2. Establishing a Data Schema for DDIL Environments

If using a DDIL twin for any data is to be considered going forward, then there needs to be some consideration for what the data schema would need to be. For the S-122 MPA data this would be the simplified bounding box versions of the original data. As the simplified version of the data is envisaged to be a supplement to the original data rather than a replacement it would need to share common attributes with the original data and have a clear link back to the original features.

17.4. GeoJSON

17.4.1. Work on GeoJSON

The following elements of the GeoJSON encoding remain to be worked out. Some attempts at definition have been done but a consultation period with interested parties is probably required before a first draft of an encoding is published for testing.

1. **Information Types**: GeoJSON and its clients are often not good at dealing with features which have no geometry.

2. **Relationships**: A systematic way of associating features and `InformationTypes` together needs to be established in the encoding.

3. **Identifiers**: Each encoding implements its own identifier scheme and the GeoJSON encoding should implement one.

4. **Other standardized metadata**: The feature type should be included as a standard field in each feature.

5. **Geometry**: As topoJSON becomes more accepted, an extension of this encoding to include a formal topology would be valuable.

The purpose of any future work should be to make a much closer mapping between the S-100 General Feature Model and GeoJSON. This would make it suitable for use as a generic encoding for other S-100 product specifications, forming a stronger link between the OGC and IHO family of standards. The current implementation encodes around 80% of the existing data but the remaining 20% requires clarification and further specification as described.

17.4.2. GeoJSON and S-100

There are two aspects related to GeoJSON and S-100 which might need exploring in more detail.

- **The use of the IHO feature catalogue and an analogous structure for GeoJSON data:** Each feature in S-100 is, normally, different with the FC describing the range of possible attributes and values they can have. Most GeoJSON data tends to have the same attributes for each feature (essentially a tabular structure). So, while our encoding is conformant, it may pose challenges for some implementers and perhaps there are accommodations which can be made (the API should probably return the FC / JSON Schema in its conformance classes).

- **Aggregation:** S-100 aggregates features into data sets and data sets into exchange sets with appropriate placement of metadata. OGC API — Features only has one level of aggregation, that of items into collections. A robust way of recursively aggregating would be a good step forward for OGC and provide an analogous structure for "exchange sets".

The ability to seamlessly aggregate datasets together in GeoJSON would be a good step forward.

17.4.3. GeoJSON and Moving Features

The extended use case for querying vessel traffic as it relates to marine protected areas loses important information specific to the vessel route. Vessel traffic is represented as a series of waypoints, each waypoint providing information for the vessel's heading, current draught, rate of turn, etc. Conversion of a vessel route into a GeoJSON compliant feature requires the waypoints to be represented as a `LineString` of (latitude,longitude) points. GeoJSON natively cannot maintain the relationship between the `LineString` points and the respective vessel's waypoint information.

For future work, application of the OGC Moving Features JSON standard would be well-positioned to represent vessel traffic. The Moving Features JSON standard leverages GeoJSON for moving feature data (OGC 19-045r3).

A

ANNEX A (INFORMATIVE) FMSDI PHASE I (RFI) SUMMARY AND RESULTS

ANNEX A (INFORMATIVE)
FMSDI PHASE I (RFI) SUMMARY AND RESULTS

This Request for Information (RFI) was issued in September 2021 to help determine data availability and accessibility of IHO S-122 data in Marine Protected Areas (MPA) and other marine data in the North Sea and Baltic Sea. The RFI was part of the Marine Data Availability and Accessibility Study (MDAAS) to help further assess interoperability, availability, and usability of data, geospatial Web services, and tools across different regions and uses of marine spatial data. MDAAS was the first phase of the OGC Federated Marine Spatial Data Infrastructure Pilot (FMSDI). It identified gaps and helped define reference use-cases and scenarios for future FMSDI Pilot activities.

The responses to the RFI were requested by October 1, 2021, and the results were finalized by December 2021. As a result of this phase, a diverse group of stakeholders from the global marine community has been brought together to assess the current state of Marine SDI. The RFI is used to gather knowledge from marine domain stakeholders and contributors.

A.1. Respondents

In total, 17 organizations have responded to this RFI. Table A.1 shows the organization names and contact information.

Table A.1 — RFI Respondents

ORGANIZATION	CONTACT
Danish Geodata Agency	Sophie Hohwü-Christensen, Senior Consultant
Finland Traficom	Juha Tiihonen, Chief specialist
Flemish Hydrography	Alexander Cattrysse, Contract Manager
Dutch Marine Information and Data Centre	Ellen Vos

ORGANIZATION	CONTACT
Lithuanian Transport Safety Administration	Mindaugas Zakarauskas
German Federal Maritime and Hydrographic Agency	Jens Schröder-Fuřstenberg
Danish Maritime Authority	Trine Skovgaard Kirkfeldt, GIS specialist and planner
The Danish Environmental Portal	Dorthe Holme, Senior Consultant
Swedish Hydrographic Organization	Benjamin Hell, S-100 Expert
HELCOM — Baltic Marine Environment Protection Commission	Joni Kaitaranta, Data Coordinator
UK Hydrographic Office	International Bodies and Technical Engagement
IIC Technologies UK	Jonathan Pritchard, Senior Technical Manager
The Agency for Culture and Palaces	Christian Hofma, System Owner
Geoscience Australia	Jonah Sullivan, Geospatial Advisor
AusSeabed	Kim Picard, AusSeabed Team Lead
NGA (National Geospatial-Intelligence Agenc)	Matt Boeding, Sebastian Carisio, Caitlin Johnson, All Lead Technical Cartographic Analyst
ESRI	Rafael Ponce, Maritime Practice Lead

Although this RFI was meant for Baltic Sea and North Sea stakeholders, due to a small number of respondents other similar organizations around the world were reached.

A.2. Questions & Response Summaries

A total of 29 questions were asked in eight categories. The summary of this RFI is provided in the following sections.

A.2.1. Stakeholders

Question: *What organization are you affiliated with, and what is your role in the marine domain? (e.g. transportation, marine biology, oceanography) Are you a data provider/owner (e.g. data, tools, applications, services)? Are you primarily a marine data user (e.g., science, research)? Are you a data enabler (e.g., help provide access to the data, software company, data standards organization, app developer)?*

There were somewhat varied responses among the 14 respondents. Although 29% of respondents assumed two roles, the majority (43%) were hydrography (Figure A.1). Figure A.2 shows that 86% of the respondents were data producer/owner, while 42% identified themselves as data broker/enabler, and 25% as data users.
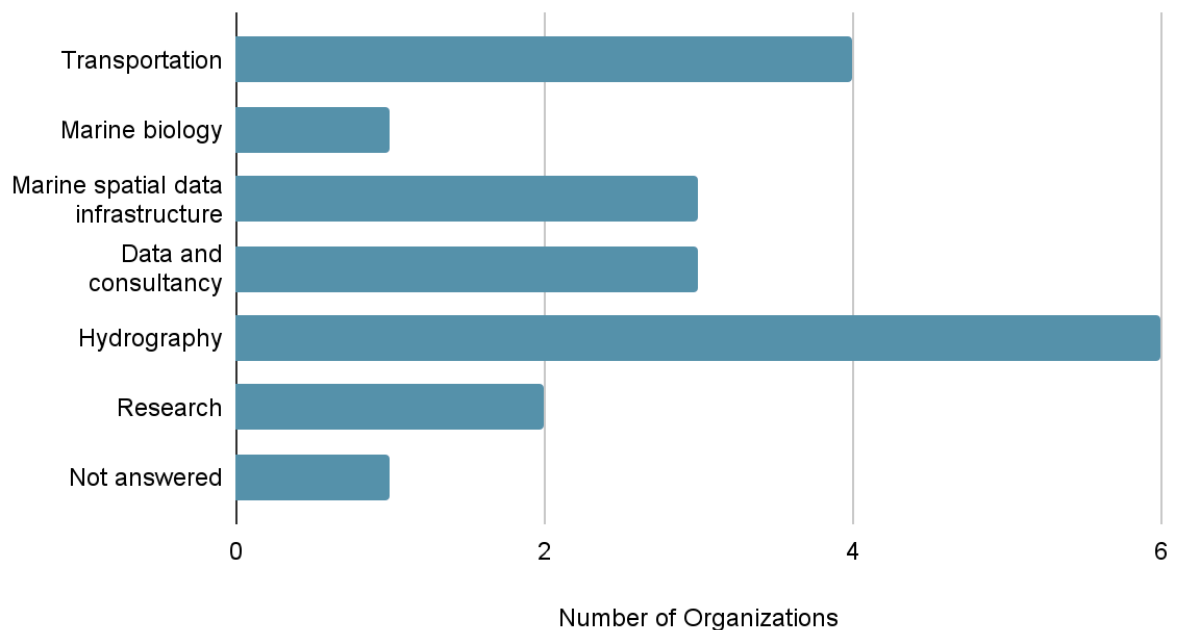

Figure A.1 — Summary of the answers for the role of the 14 respondents in the marine domain
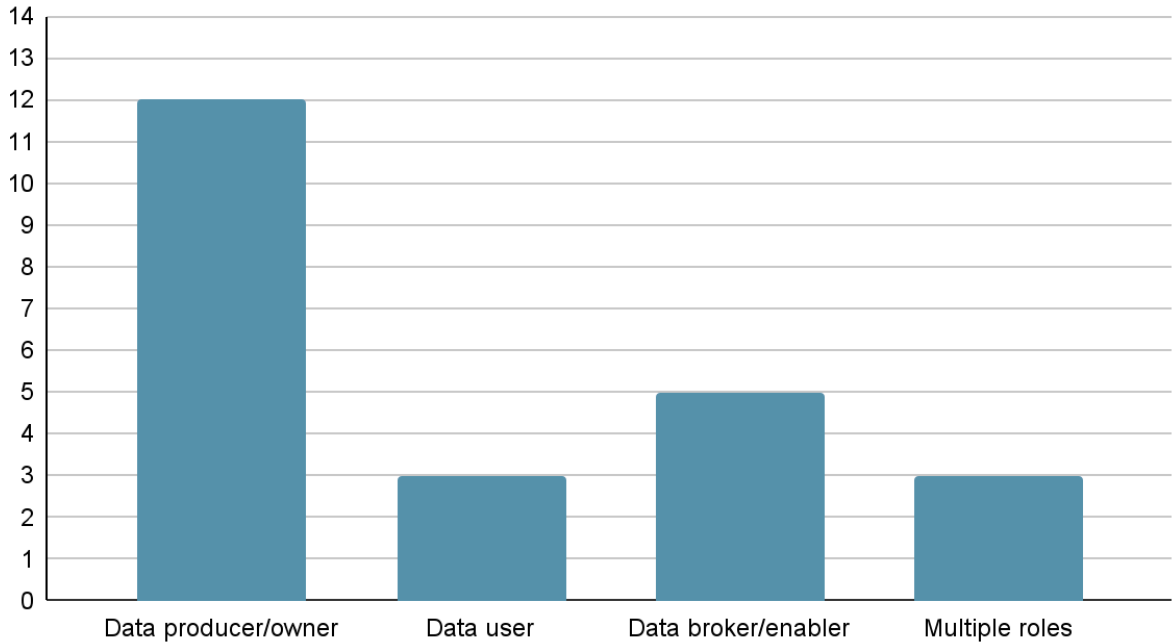
Figure A.2 — Summary of the 14 responses received to the question regarding their role as a data "producer/owner", or "user" or "broker/enabler" or have a "multiple roles"

A.2.2. Marine SDIs and Data Architectures

Question 1: *How significantly do you/your organization rely on MSDIs for data dissemination and/or data access?*

Figure A.3 shows that out of 12 responses received for this question, only 8% use MSDI extensively, 25% use it partially, 8% use it minimally, 33% are MSDI providers but do not use it internally, 17% feel the need for it but do not use it or have access to it, and 8% do not use it at all. The general observation is that an MSDI is currently relied upon at varying degrees and is expected to become increasingly important.
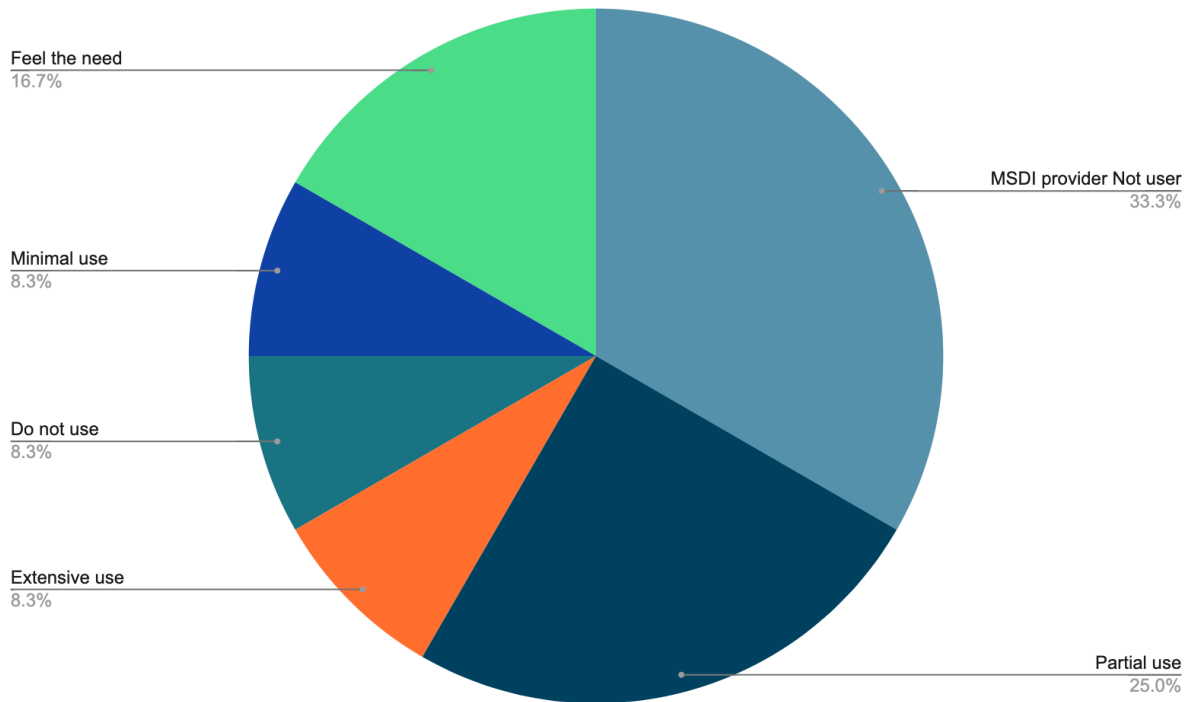
Figure A.3 — Summary of the 12 responses to the question regarding
the participants' reliance on MSDIs for data dissemination and/or access

Question 2: *Does your organization currently contribute data and/or services to a Federal/National spatial data infrastructure? If so, please provide a brief description of how this is accomplished, and the scope of data provided.*

Out of 12 received responses, 92% of the organizations contribute to a national SDI, and 8% do not contribute to any SDI.

Question 3: *What do you think should be the key technology components (e.g., standards, networks, clients, web services, data storage) of an MSDI?*

Out of 12 received responses, 17% of respondents did not specify a priority (e.g. standards, etc.). A summary of the received replies is available in Figure A.4. Much of what was touched on focused on well-defined metadata and data storage, possibly leveraging cloud technology for storage and computation. The respondents who mentioned standards in their replies were focused on utilizing IHO standardized formats and OGC web services.
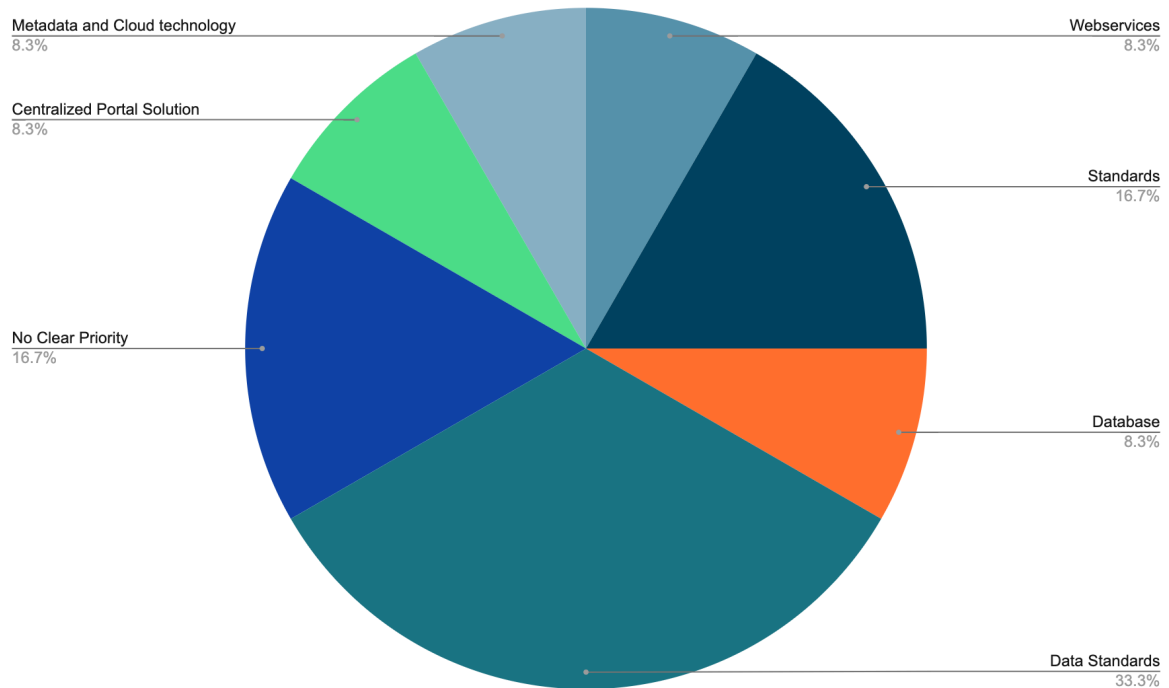
Figure A.4 — Summary of the 12 responses to the question regarding the key technology components of an MSDI.

Question 4: *Do spatial data infrastructure currently support your need to make available, or access, data related to the Marine domain?*

Out of the 13 responses, 39% of the organizations' needs are met regarding the accessibility of data in the marine domain. The needs of 31% are partially accommodated, 23% did not know, and only 8% are unsatisfied with the current data infrastructure.

31% of the organizations did touch on a lack of interoperability between SDIs and a lack of clarity behind metadata standards.

Question 5: *What do you think is the best way to support an international/regional/national MSDI?*

In this case, the responses were very diverse, with the main themes being:

- Clear, concise, short, and simple to understand standards;

- Clear international avenues for discussion amongst contributors;

- Data integration between clouds;

- Close cooperation, trust, and FAIR principles; and

- It is easy to make data, but it is difficult to work together(data and governance).

Question 6: *Does your organization have a marine data management system? If so, please briefly describe the system's capability.*

Figure A.5 shows the wide variety of responses that have been received from 13 respondents. While only 16% currently do not have a system, 33% have in-house solutions, and 17% did not specify. Others (33%) have a multi-database system with data compliant with standards such as S-57, CARIS, FME, and S-100.
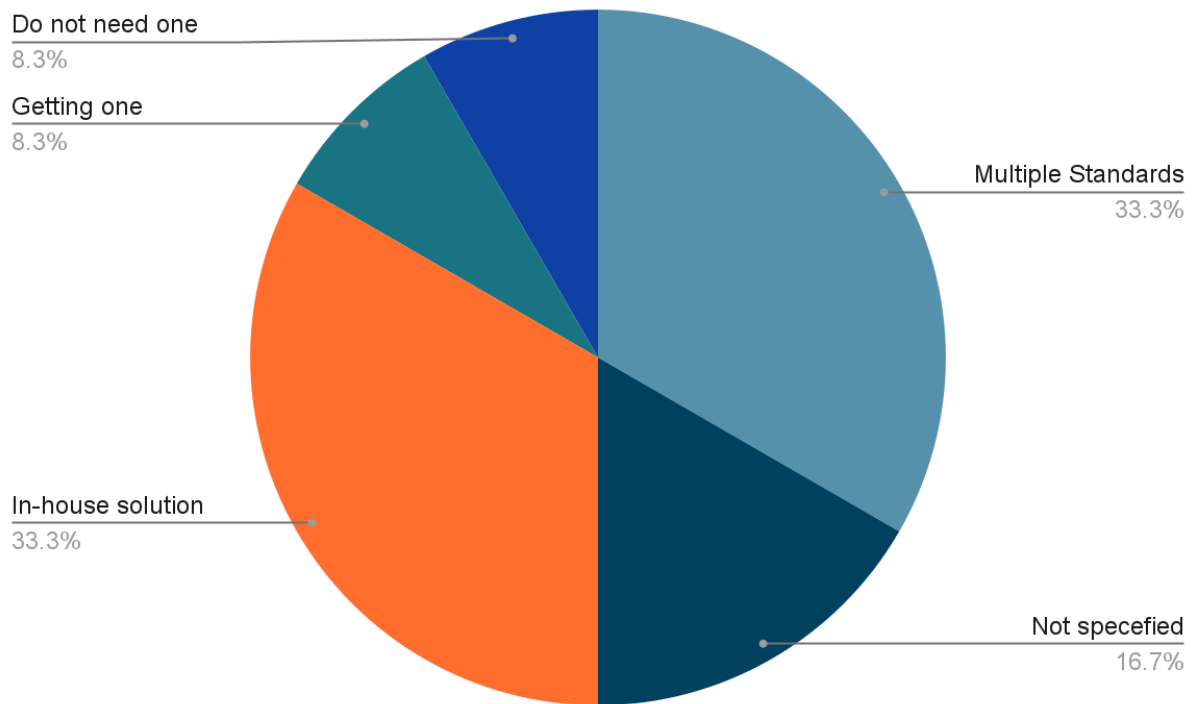

Figure A.5 — Summary of the 13 responses received to the question regarding the participants' ownership of a marine data management system

Question 7: *Do you currently use geospatial standards to access data and services? If so, what are the key geospatial standards you use?*

Figure A.6 shows the quite varied answers received for this question: OGC W*S, IHO S-57, IHO S-100, GML, RDF, GDAL, and GIS standards
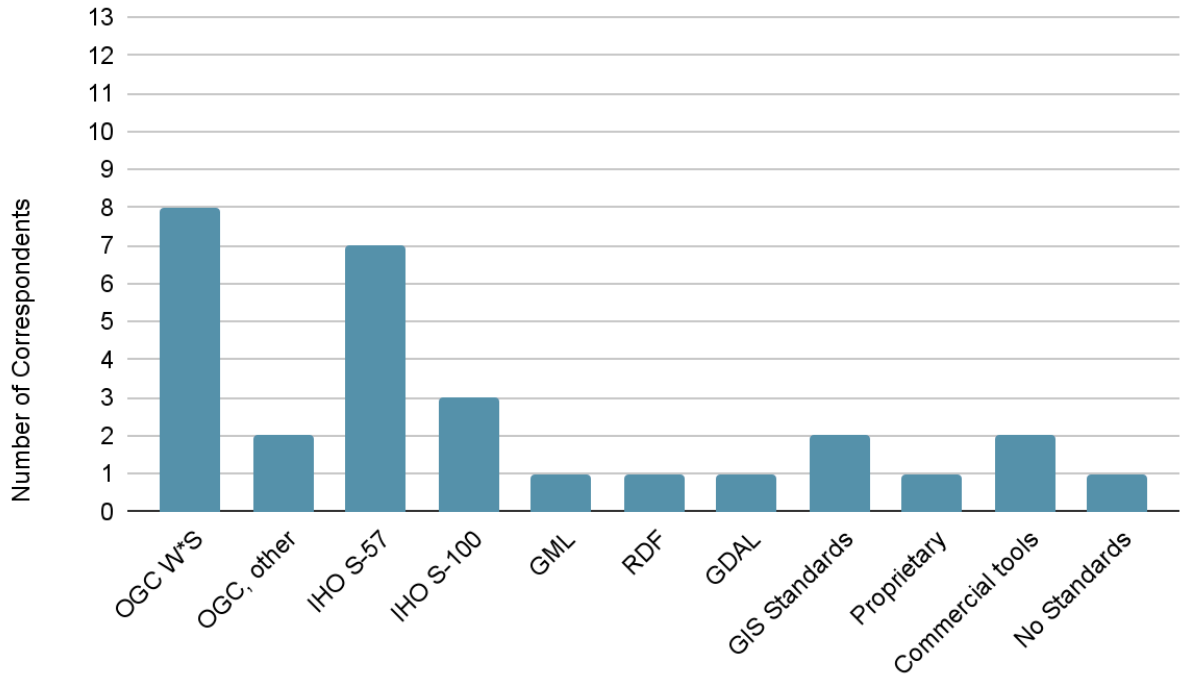
Figure A.6 — Summary of the answers for the key geospatial standards used by the 13 respondents

A.2.3. Data for Marine Protected Areas and Other Marine Data

Question 1: *What data do you/your organization provide that could be included within a Federated MSDI architecture to support Marine Protected Areas or other marine uses? (Please provide detailed accessibility information)*

Out of the 17 responses (Figure A.7), 35% provided links to access some data when applicable; the rest of the organizations were unclear whether their data was publicly available. Much of the data are depths, bathymetry, and seafloor topography.

Only 57% of the correspondence mentioned Marine Protected Area data. These observations are based on the limited information provided by the correspondents.
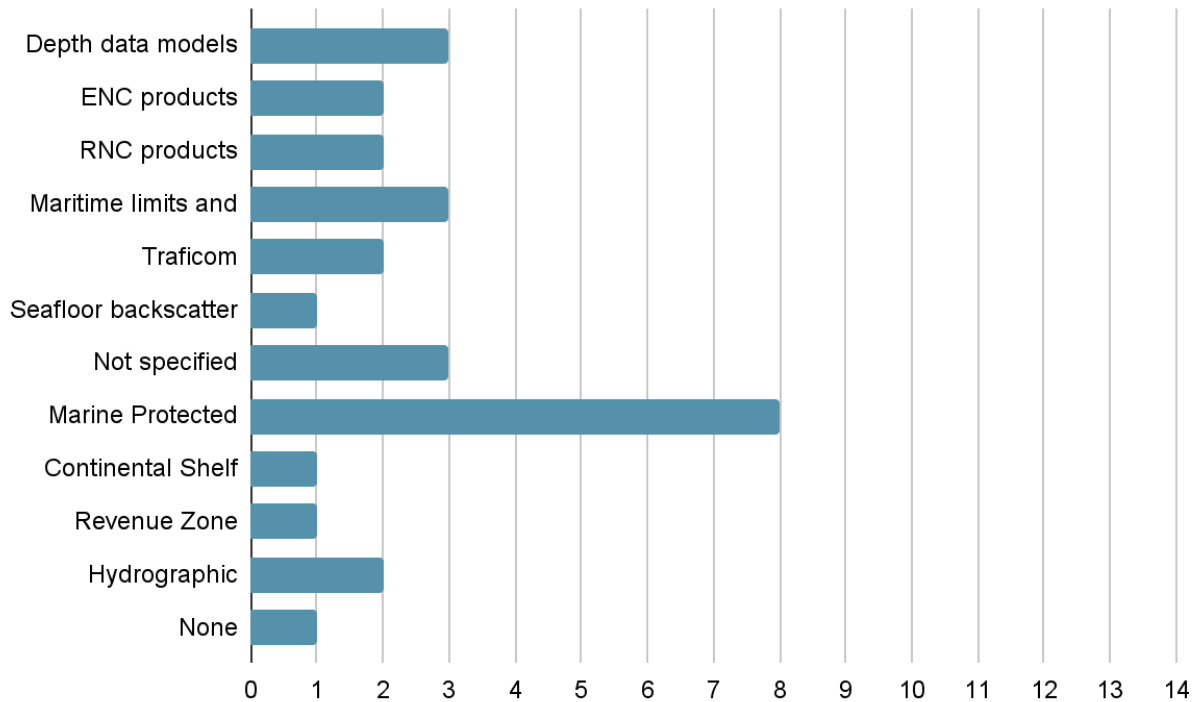
Figure A.7 — Summary of the answers for what data the 17 respondents
can provide that could be included within a Federated MSDI
architecture to support Marine Protected Areas or other marine uses

Question 2: *In what formats or by what means do you/your organization share, or would be most able to share, this data?*

OGC WMS/WFS, WCS, S-57, S-100, S-121, S-122, GeoTIFF, GeoPDF, ShapeFile, ESRI geodatabase, GeoPackages, GML, KML.

Question 3: *Within the context of an MSDI, what current and/or emerging open international standards does you or your organization currently employ?*

Out of 14 responses, 86% support OGC Web Services and 36% can support OGC APIs. 36% support IHO standards and 14% have self-describing data. The following graph, Figure A.8, provides more information.
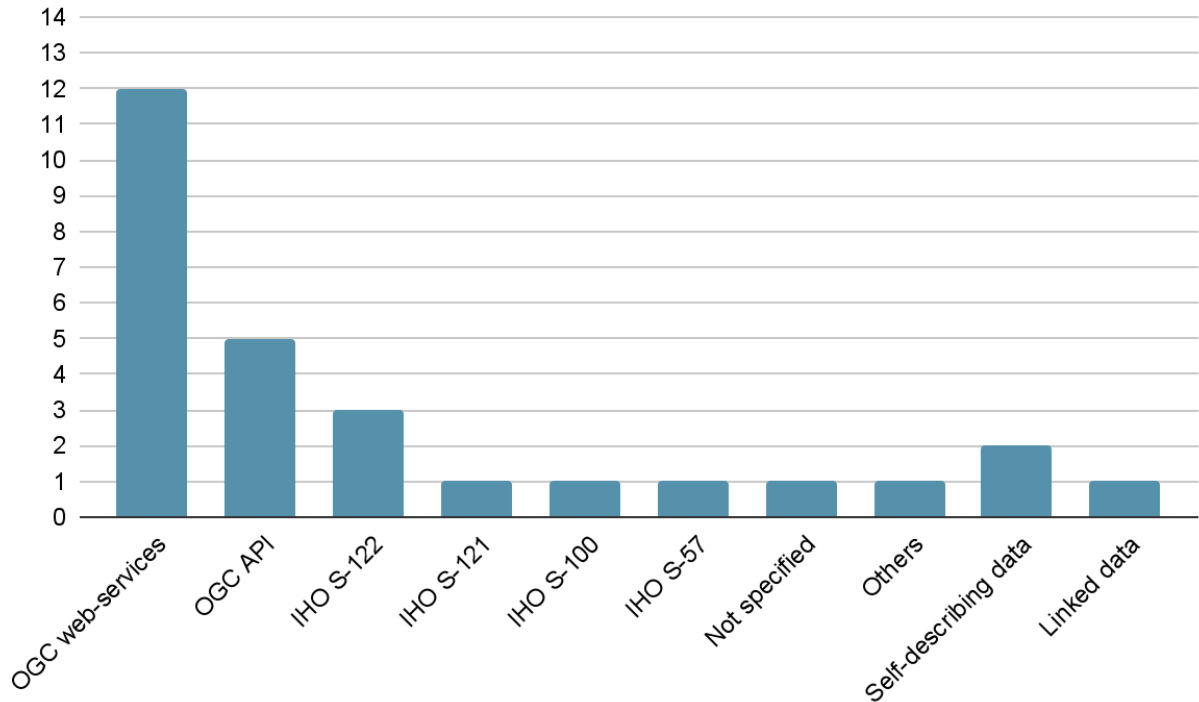
Figure A.8 — Summary of the answers from 14 respondents regarding what current and/or emerging open international standards they employ within the context of an MSDI.

Question 4: *Is the data you provide "analysis ready" or "fit for use"?*

Out of the 14 respondents, 57% had difficulty answering this question, as 21% of the organizations point out that the definition of "analysis-ready" is an ill-defined term that varies by user, and they were unable to answer this question. This feedback speaks volumes about how we should approach defining our data usage and standards usage and take deep consideration of the end-user when scoping. 43% of the organizations believe their data is "analysis-ready" but defined their data as based upon the user, database views, standardized for intended use, or requiring minimal user processing.

Question 5: *Is the data you provide free/open source, require payment for access, or does the data require some other criteria to access?*

Out of 17 respondents, 76% of the organizations have open/free data available to varying degrees. The 18% that do not currently have open/free data require payment, and only one respondent (6% of the statistical population) is exploring avenues to commercialize the data.

Question 6: *Do you provide tools for analysis of your data?*

Overall (14 replies), 57% provide no tools to analyze the data, 29% depend on web tools and provide avenues to those tools or have a minimal suite for reporting (14%).

Question 7: *Are the tools or data you provide only accessible to limited, experienced people or general populations?*

Out of 13 replies, only 8% provide data only to limited users. Some 46% have fully accessible data for the general population, and the remaining 46% have some partially restricted, partially open data. Those with open data have the data open to the general population with an expectation that the population is familiar with the standards utilized to compose the data.

Question 7: *Do you use models, and if so, how?*

Out of 11 received replies, 56% use models, although only half of these respondents described how they use the models.

A.2.4. Technologies and Applications

Question 1: *Are there other national, regional, or topical portals that can be used to support the marine domain that are currently available and serve your needs? How might they be improved?*

Out of 12 responses, 67% supplied links or names when applicable. Two European countries are on EMODnet. Suggestions for improvement are as follows.

- Keeping up with changes between different responsible parties can be challenging, and interdepartmental communications should be improved.

- Not every portal has a clear catalogue or registry of services which is important for ease of connecting.

- The accuracy of the provided data should be improved.

- Ensure that "authoritative data" is easy to find and use.

Question 2: *What other types of applications, tools, and services do you believe should be developed or integrated as part of an international Federated MSDI?*

Only seven respondents provided suggestions. Those that answered focused on an international standard set of data, akin to an overview from all contributing nations. Going further, there were nods to having shared compute environments, data streaming capabilities, and semantic translation tools to move data from one standard to another (e.g., INSPIRE Natura2000 to S-122), up-to-date MSDIs', and easy access to overview data..

A.2.5. Requirements

Question 1: *What requirements, (including constraints) do you experience that should be considered for future design and development of an international marine spatial data infrastructure architecture?*

- A common data frame utilizing international standards with a coordinated approach to acquisition and storage of data/analytics.

- A public and private gating system that is data-centric.

- MSDI datasets and services should be interoperable with other related SDI.

- A common data frame, interoperable standards, and cross-border harmonization.

- Acquisition and storage costs are prohibitive.

- Web service-enabled information.

- Privacy requirements for commercial activities.

Question 2: *Are there sufficient tools available to help you meet your requirements? Please describe any performance issues you may experience? If so, what are the issues?*

Ample free tools, web tools, and in-house tools exist. The issue lies in working in low-bandwidth (DDIL) areas and overall network performance — simultaneous data-heavy requests, an influx of downloads, scalability concerns, and costs associated with maintenance and support.

Question 3: *What privacy and/or confidentiality requirements or concerns are associated with the datasets you provide?*

- Out of 12 responses, 50% had no privacy and/or confidentiality requirements, 17% were copy-write protected, and the rest of the answers were unclear.

Question 4: *Are there any data licensing/rights requirements associated with the datasets you provide?*

As depicted in Figure A.9, out of 14 respondents, 21% made the data available under Creative Commons, 21% had some other licensing, and 35% had no licensing.
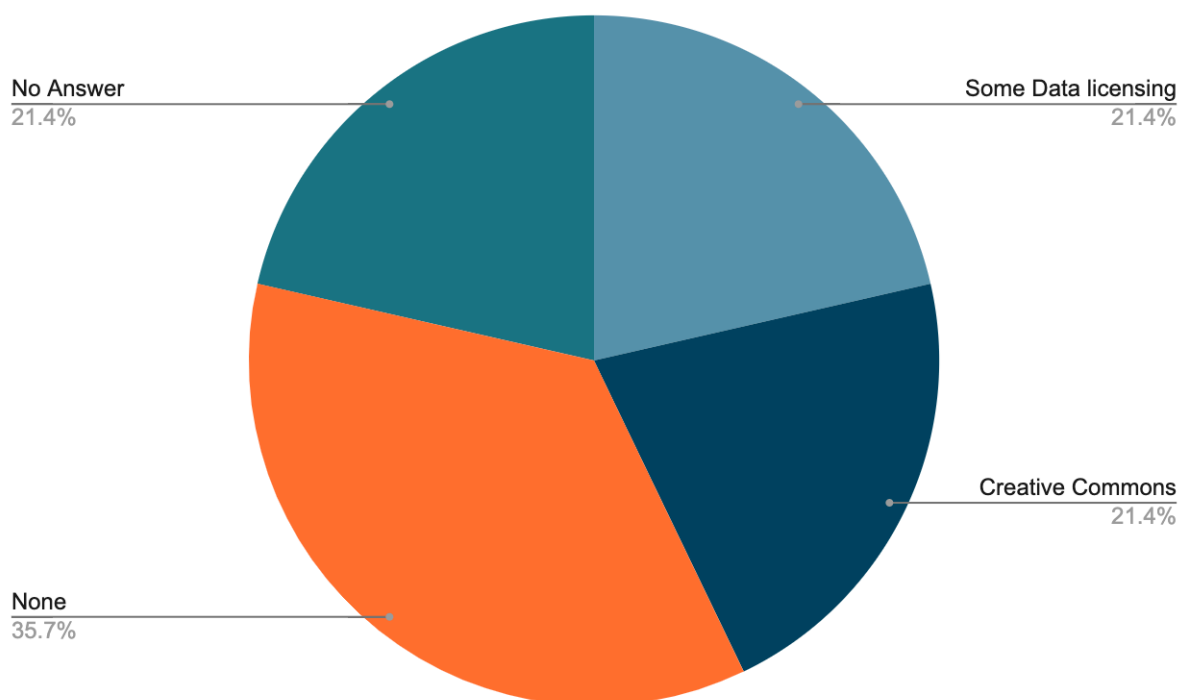
Figure A.9 — Summary of the 14 responses to the question regarding the participants' data licensing/rights requirements associated with their provided datasets

A.2.6. Scenarios and Use Cases

Question 1: *What scenarios and use cases would you like to recommend as part of Pilot Activities?*

- Global models for land/sea interface technology.

- Interoperability between IHO and OGC standards, with the implementation of OGC API standards for data access. Meeting UN-GGIM requirements and the fundamental data themes with content-neutral encodings of marine data.

- Modern techniques for data conversion technology between different model representations, real-time/near-real-time automatic identification system usage.

- Use of iso-latitude DGGS for representation of vector and coverage data in polar areas, arctic regional voyage planning, and maritime search and rescue

Question 2: *Do you have any information on the benefits or successes (e.g. societal or economic benefits) of establishing a MSDI?*

- Good governance practice

- International collaboration

- Enhancing focus and study for environmental protections

- Disaster and disease response expanding through and beyond international boundaries

- Improvement of the safety of navigation

- Analytical tools and web services freely available to the community for anyone doing work in the region

- Open data hosting for governments that want to contribute their data to the community at no cost

- Interoperable platform that works with a variety of locally run software programs

- Access to leading-edge solutions, innovations, and services that improve the livelihood of people throughout the region

A.2.7. Operation and Organization

Question 1: *What policy, organizational, and administrative challenges do you have that must be addressed to improve a MSDI architecture internationally?*

Financial structuring to encourage data owners to share their data, how the data itself is structured — does an organization define the data well, provide ample enough metadata, and have an interface to bring together multiple servers intuitively?

Question 2: *Are there unique needs that need to be considered at various levels of marine operations (local, state, regional, national, international levels), and by various players (government, commercial, NGO, academia/research)?*

Yes, each operating level will value the data differently, and those needs must be considered. Additionally, data may need to be considered as politically charged and treated within specific constraints.

A.2.8. Other Factors and Items

Question 1: *What other success factors or considerations do you see as needed for a successful international MSDI?*

Somewhat varied responses, ranging from success being strictly quantified through continued funding or additional financial incentives to widely recognized standards use and the creation of an interoperable system meeting the majority of user needs.

Question 2: *Are there any other data terrestrial, meteorological, etc. that you encounter or use?*

Many different data are used: transportation data, terrestrial layers, meteorological data, digital elevation models, 3D Seismic data, satellite observations, and applications of machine learning on various data.

Question 3: *What additional data would be of interest to combine and mingle for analysis?*

Another highly varied response: marine traffic density, sea chart information, terrestrial/land-based data, voyage planning data, water currents, wave height, tidal data, wind strength, sentinel 2/satellite data, and Spatio-temporal datasets for tracking vessel movements

A.3. Conclusion and Future Work

The need for international collaboration in the FMSDI is clear. Driven by global expectations or guidance, a regional approach for the FMSDI may be best: based on regional partners operating under a common goal, federating data & services from their individual and national organizations, and utilizing the same open standards for releasing their similar themes of data. As these regional MSDIs become established, they can coordinate with neighboring regions to ensure interoperability and share best practices for developing the MSDI.

Another observation was the need for interoperability between IHO and OGC standards. By implementing interoperability between OGC and IHO, it is possible to make more IHO S-100 datasets available to the public.

Use-cases for interoperability between OGC and IHO S-100 compliant products, and datasets, land/sea interface, and marine protected areas from neighboring countries might be some of the topics of interest.

In the context of future work, the links to portals and datasets provided through this RFI will be used in future FMSDI phases. Also, due to the responses received in this RFI, a somewhat more refined questionnaire could be devised for future RFIs. Another future work may be building a stakeholder community of data users, producers, and enablers to expand this effort and collaborate on the finding of this RFI.

BIBLIOGRAPHY

BIBLIOGRAPHY

1.   Portele, C.: OGC API — Styles. Open Geospatial Consortium, http://docs.opengeospatial.org/DRAFTS/20-009.html .

2.   International Hydrographic Organization: Marine Protected Area Product Specification. IHO Publication S-122. Edition 1.0.0 (2019).

3.   International Hydrographic Organization: S-122 Marine Protected Areas (MPAs), http://s100.iho.int/product%20specification/division-search/s-122-marine-protected-areas-mpas.

4.   International Hydrographic Organization: S-100 Introduction, http://s100.iho.int/home/s100-introduction.

5.   Portele, C., Vretanos, P.A., Heazel, C.: OGC API — Features — Part 1: Core. Open Geospatial Consortium, https://docs.opengeospatial.org/is/17-069r3/17-069r3.html (2019).

6.   Portele, C., Vretanos, P.A.: OGC API — Features — Part 2: Coordinate Reference Systems by Reference. Open Geospatial Consortium, https://docs.ogc.org/is/18-058/18-058.html (2020).

7.   Vretanos, P.A., Portele, C.: OGC API — Features — Part 3: Filtering. Open Geospatial Consortium, http://docs.opengeospatial.org/DRAFTS/19-079.html .

8.   Panagiotis (Peter) A. Vretanos, Clemens Portele: OGC API — Features — Part 4: Create, Replace, Update and Delete. Open Geospatial Consortium, http://docs.ogc.org/DRAFTS/20-002.html

9.   Vretanos, P.A., Portele, C.: Common Query Language (CQL2).

10.   Mark Burgoyne, Chris Little, Chuck Heazel, Dave Blodgett: OGC API — Environmental Data Retrieval Standard. Open Geospatial Consortium, https://docs.ogc.org/is/19-086r4/19-086r4.html