INTERNATIONAL MARITIME ORGANIZATION

INTERNATIONAL HYDROGRAPHIC ORGANIZATION

E

IMO/IHO HARMONIZATION GROUP ON DATA MODELLING
Agenda item 5

HGDM 1/5/2
15 September 2017
ENGLISH ONLY

DEVELOPMENT OF A DEFINITION FOR MSPS AND CONSIDERATION FOR THE HARMONIZATION OF THE FORMAT AND STRUCTURE OF MSPS

Proposed approach of the HGDM for development of MSPs Guidance

Submitted by Australia, Denmark, Republic of Korea

SUMMARY

Executive summary:	This document in conjunction with submissions from Norway and the Republic of Korea proposes a work program to the IMO/IHO HGDM to help develop guidance on the definition and harmonization of the format and structure of Maritime Service Portfolios (MSPs) in accordance with the instruction from MSC 98 (MSC 98/23, paragraph 11.37)
Action to be taken:	Paragraph 4
Related documents:	MSC 90/27, NCSR 4/27, MSC 98/20, MSC 98/23

Background

1 This document with its annexes is to be considered together with document HGDM 1/5/1 (Australia *et al.)*.

2 HGDM 1/5/1 proposes a generic template for MSP's and gives an example of an MSP (MSP5) with references to technical services.

3 This document contains two annexes that complements HGDM 1/5/1:

.1 Annex 1: Guideline for specification of e-navigation technical services
This guideline (in an earlier version originating from the European EfficienSea2 project) has been submitted to IALA (ENAV20), and IALA is in the process of making this an official IALA guideline.

This guideline explains how to define e-navigation technical services. This is done in three levels:

.1 Service specification (technology agnostic specification including reference to relevant S-100 data model);

.2 Service technical design (how to specify an implementation using a specific technology); and

.3 Service instance (a reference to a specific provider of a service).

.2 Annex 2: Example of an e-navigation technical service belonging to MSP5; Navigational Warnings (NW) and temporary / preliminary Notice to Mariners (NM T&P).

The service is described on the overall level (service specification) and with examples of associated technical design and service instance.

The services specification also makes reference to the current IHO draft S-124 product specification. At this stage neither the services specification nor the S-124 product specifications are finalised, and thus there is not a perfect match between the draft service specification and the draft S-124. Obviously, this must eventually be fixed.

The current draft S-124 product specification can be found here: http://www.iho.int/mtg_docs/com_wg/CPRNW/S100_NWG/2016/S-124NW-CG-01_2016-Draft_Product_Specification-03.12.2015.zip

It is important to stress that, although this annex may serve as input to a discussion of a specific MSP5 related technical service, the main purpose of including this document is to service as an example on how to use the guideline for specification on e-navigation technical services (annex 1).

Action requested of the HGDM

4 The HGDM is invited to note the information contained in this document.

***

ANNEX 1: GUIDELINE FOR THE SPECIFICATION OF e-NAVIGATION TECHNICAL SERVICES

CONTENTS

CONTENTS

List of Tables

CONTENTS

List of Figures

1 INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

This Guideline provides information on how to make specifications of technical e-Navigation services.

Taken from the concepts of service-oriented architectures, a technical service refers to a set of related software functionalities that can be reused for different purposes together with policies that govern and control its usage. A technical service is a service offered by an electronic device to another electronic device. Often operational services are implemented by electronic devices that offer several technical services to use the operational service.

A technical e-Navigation service should be formally specified and documented as described by this Guideline. This Guideline aims at improving the visibility and accessibility of available e-Navigation technical services and information provided by them. This enables service providers, consumers, and regulatory authorities to share a common understanding of a technical service and how to implement and use it.

This Guideline is intended for service architects, system engineers and developers in charge of designing and developing a technical service or design and developing a device to use it.

Furthermore, this Guideline is intended to be read by enterprise architects, service architects, information architects, system engineers and developers in pursuing architecting, design and development activities of other related services.

This document provides meta-information explaining how services shall be described and documented. The guidelines, help providing specifications, design documents and instance descriptions for any kind of e-Navigation technical service in a standardized way. Any technical service documentation in the context of e-Navigation should comply with this Guideline.

1.2 LINK TO S-100 AND PRODUCT SPECIFICATIONS

Service specification and design artefacts described in this document include the reference of data models. It must be mentioned that, when available, the appropriate Product Specification shall be used. In case there is no related Product specification available, whenever possible, make use of the concepts of S-100.

This means, service descriptions following this Guideline shall use an S-100 conformant data model.

2 OVERVIEW

2.1 SERVICE MANAGEMENT OVERVIEW

A service management concept can be visualised as shown in Figure 1. Both, service specifications as well as information about service instances can be published in a service registry. A service registry can be a collection of documents, or could be realised as a service itself that would have an Application Programming Interface (API) for automatic interfacing to the registry (lookup, updating, deleting etc.). This concept is implemented as the Maritime Service Registry within the Maritime Connectivity Platform (MCP, formerly called the Maritime Cloud), see http://www.maritimecloud.net.

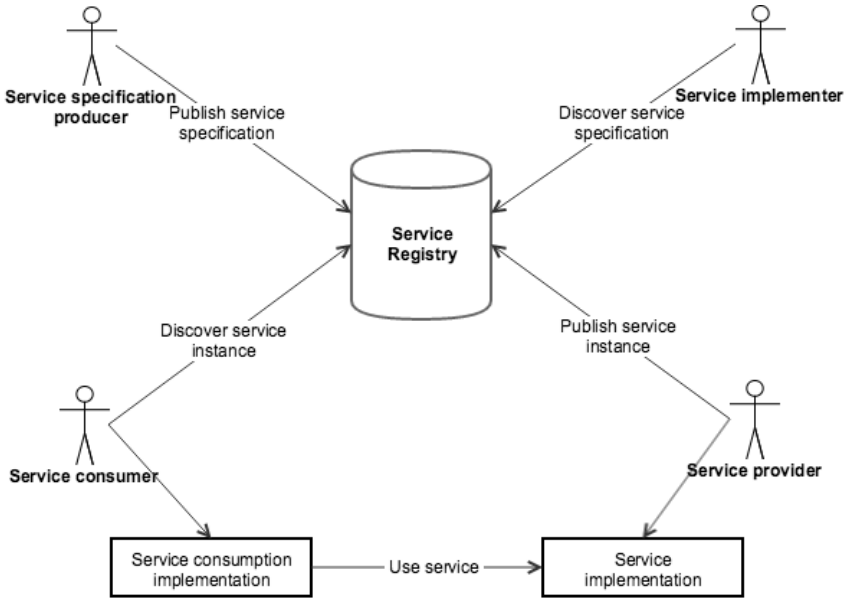


Figure 1 Service Management Concept

Figure 2 provides more insight in the distinction between service specification, service technical design and service implementation. The service specification describes one dedicated service at logical level in a technology-agnostic manner, by providing, for example:

- the operational context of the service (e.g. requirements, use cases);
- the service interface descriptions (operations, parameters);
- the data structures used by the service (the service data model);
- the dynamic behaviour of the service (sequence of operations);
- author of the service specification (organisation, contact person).

The service specification shall not describe the details of a specific service implementation. For that purpose, a service technical design description must be provided, where the actual realisation of the service with a dedicated technology shall be described.

It is possible to provide different technical designs (by using same or different technologies), all being compliant with the same service specification. It is also possible to provide one technical design that conforms to several service specifications, e.g. to allow backward compatibility to older versions of a certain specification.

Each service technical design shall be documented by providing, for example:

- reference to the service specification;
- description of the chosen technology
- detailed description of the used data structures (service physical data model);
- mapping of the used data structures to the service specification's service data model;
- author of the technical design (organisation, contact person).

A service instance (implemented according to a given technical design) may be deployed at different locations by different service providers. For each such service instance a service instance description shall be provided.

Each service instance shall be documented by providing, for example:

- reference to the service technical design (and thus, implicitly, to the service specification);

- information about service provider;
- coverage information.

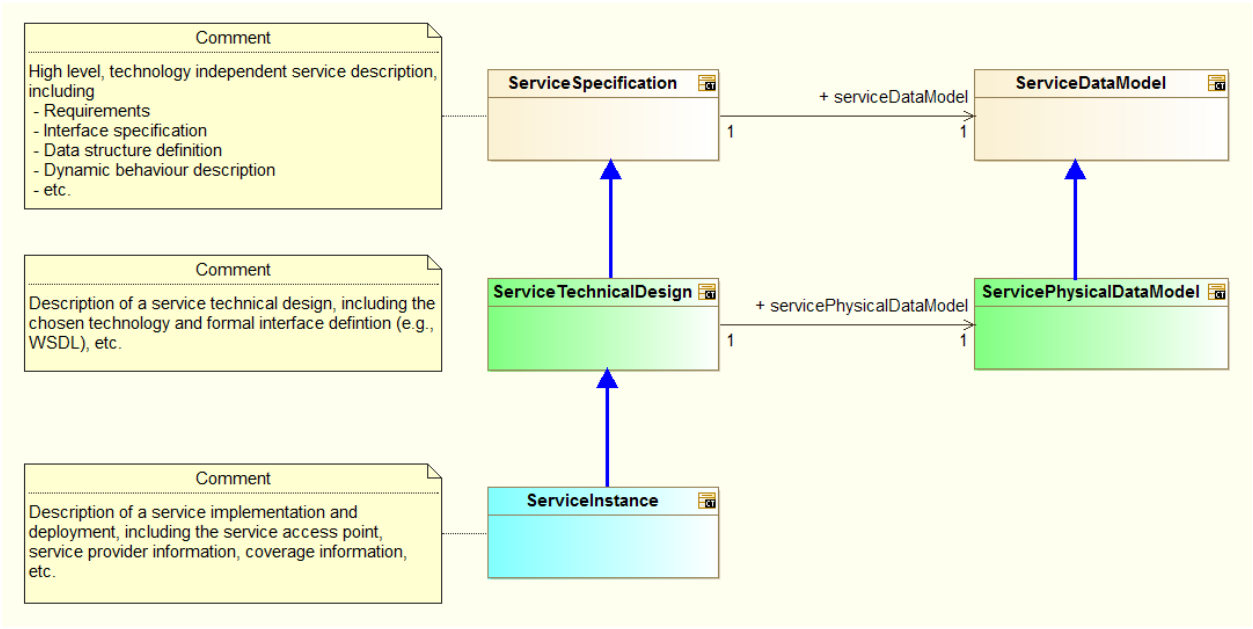


Figure 2 Distinction between Service Specification, Service Technical Design and Service Instance

Figure 3provides an overview about the service documentation artefacts. The upper rectangle contains meta-information explaining how to describe services. This meta-information consists of:

- service documentation guidelines - this document;
- service specification template - a word document providing the framework for a textual description of a service specification (ANNEX D);
- service specification XSD - an XML schema definition for the formal description of a service specification (ANNEX A);
- service design description XSD - an XML schema definition for the formal description of the service technical design (ANNEX B);
- service design description template - a word document providing the framework for a textual description of the service technical design (ANNEX E);
- service instance description XSD - an XML schema definition for the formal description of a service instance (ANNEX C);
- service instance description template - a word document providing the framework for a textual description of the service instance (ANNEX F).

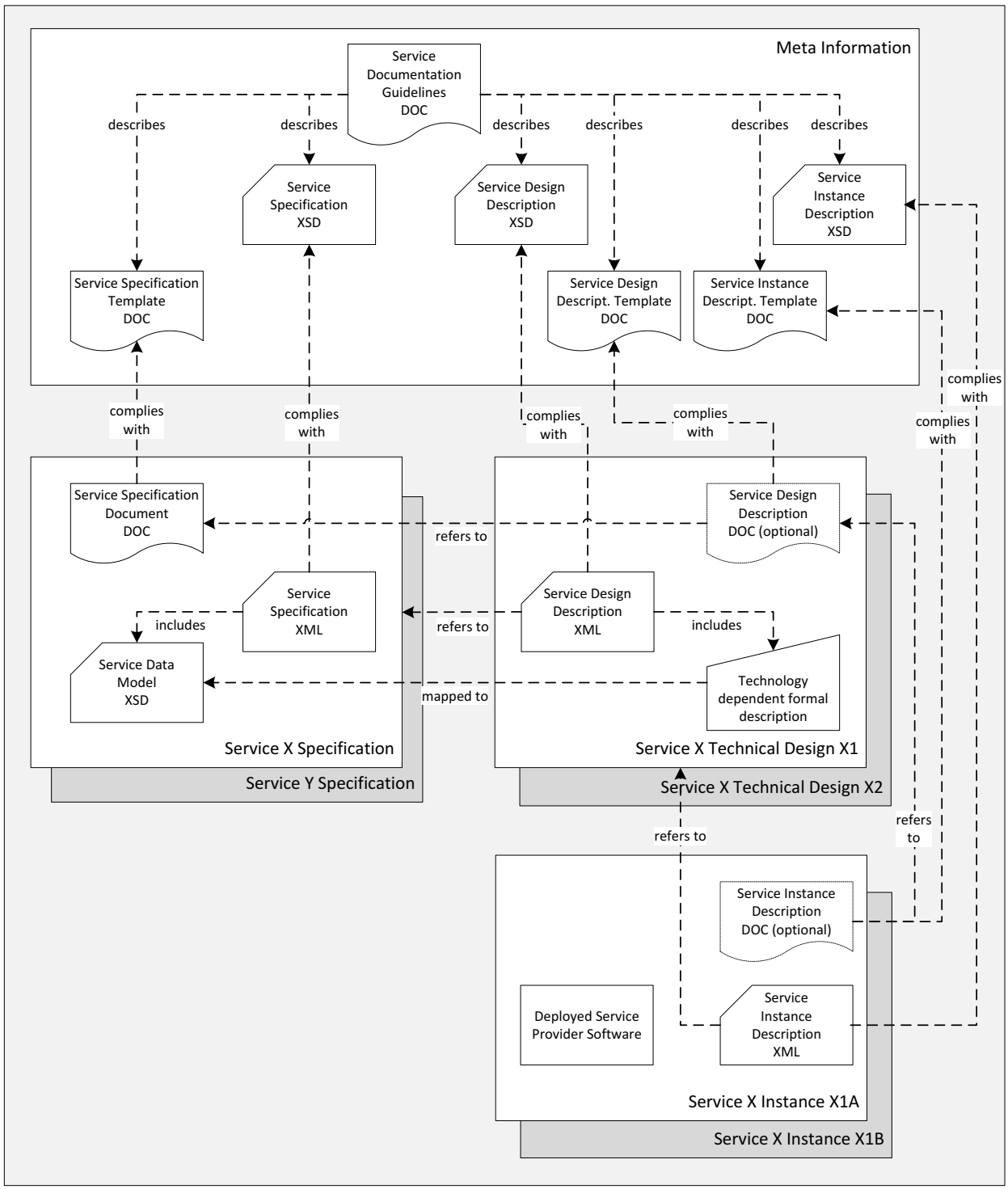


Figure 3 Service Documentation Overview

The centre part of Figure 3contains the artefacts for describing the specification and technical design of a dedicated service.

The service specification describes the 'What'-aspects of a service, e.g. what are the characteristics of a Weather Forecast Service. This service specification consists of:

- service specification document - a word document (complying with the service specification template ANNEX D) detailing the service specification in textual form supporting the readability for human beings;

- service specification XML - an XML file (following the service specification XSD schema), describing the service specification in a more formal manner;
- service data model XSD - an XML schema definition describing the data model used in the service; the service data model XSD is included in the service specification XML.

The service technical design comprises the 'How'-aspects of a service, e.g. how in detail is a Weather Forecast service instance implemented and accessible. Several different technical designs may exist at the same time for one service specification. A service technical design consists of:

- service design description - a document describing the technical service design.

  The provision of this document is optional and it is up to the implementer to choose a suitable format for this document, e.g. by using the service design description template at ANNEX F . If provided, the document shall refer to the service specification document;
- service design description XML - an XML file (following the service design description XSD), describing the service technical design in a formal manner, e.g. by providing information needed for the registration in the service registry;
- technology dependent formal description - additional service description files as appropriate for the chosen technology (e.g. WSDL, XSD, YAML, JSON, etc.), describing the details (syntax, protocol, etc.) of the exchanged data.

  The contents of this formal description must be mapped to the service data model. The means of how such mapping has to be performed is not prescribed, as they depend heavily on the chosen technology and service technical design. In some cases, the mapping is implicitly given (e.g. if the service technical design re-uses the service data model in a 1:1 manner). In other cases, a mapping table may be provided (e.g. as part of the service design description document), mapping each single data element of the service instance to a corresponding data element of the service data model.

The lower part of the figure contains the artefacts for a dedicated instance (implementation) of a service. The service instance comprises the 'Where'-aspects of a service, e.g. the actual access address (URL) of a weather forecast service and the geographical coverage of it. Several service instances may exist at the same time, all implementing the same service technical design. A service instance consists of:

- deployed service provider software implementation - the actual service implementation;

  This is not part of the description, but it is the 'subject' that shall be described. It consists of all the software and configuration artefacts needed for providing the service;
- service instance description: a document describing the actual service implementation and instantiation;

  The provision of this document is optional and it is up to the implementer to choose a suitable format for this document, e.g. by using the service instance description template. If provided, the document shall refer to the service technical design description document;
- service instance description XML: an XML file (following the service instance description XSD), describing the service instance in a formal manner, e.g. by providing information needed for the registration in the service registry.

Note: One service implementation (the same software) may be deployed several times at different access points. In this case, several service instance description XML files will need to be produced – one for each deployed instance.

2.2 PROCESS CONSIDERATIONS

This document describes how the service documentation shall look like in the context of e-Navigation. Intentionally however, this document does not prescribe any process to be followed when generating such

documentation. In particular, this document does not identify any governance rules for service technical design and implementation (see also section 6).

This means that sub-sections 2.2.1 and 2.2.2 shall be just seen as proposals, not meaning that these are the only valid approaches.

2.2.1 TOP-DOWN SERVICE DEVELOPMENT

In a top-down approach, the necessity of a new service and its basic outline would be first identified and described in an operational requirements document. This step is optional and out of the scope of this service documentation guidelines document.

Once the decision for building a service has been taken, a service architect (in the role of a service specification producer) creates the service specification by producing the service specification document and the service specification XML including the service data model. If an operational requirements document exists, the service specification refers to it; otherwise the requirements are documented in the service specification.

As soon as the service specification has reached sufficient maturity, it is published in the service registry.

An interested service provider or implementer (there could also be more than one) takes the service specification and elaborates a technical design for it. During this step, technology decisions are taken and documented in the service technical design. The service technical design may already be published in the service registry before the service is implemented and deployed. This is useful for developers of service consumer software, as they can already base their development on the service design description while the service provider software is still under development.

Having the service technical design in place, service implementers will develop the software required for service provision.

When the service software is sufficiently mature, the service provider deploys it and publishes the access information (URL) and coverage area in the service instance description in the service registry.

Interested service consumers can obtain service specifications and service technical design descriptions from the service registry and build the required client software.

Interested service consumers with existing clients look up the service registry for service instances (complying with their choice of technology) to get the access points for the provided services in their respective geographical area.

2.2.2 BOTTOM-UP SERVICE DOCUMENTATION

Existing services may be documented in a bottom-up approach. For the service to be published in the service registry, all of the following information shall be provided: a service instance description, a service design description and a service specification.

Assuming the service already exists, it should be easy to provide the service design description and service instance description in the structure/format described in this guidelines document. In this case, it is assumed that the technology specific data model already exists and can be directly taken as part of the service design description.

The service data model (part of the service specification) can be derived by abstracting the existing data model from technology-specific details.

The rest of the service specification (interface and operation descriptions as well as requirements) must be newly created.

Once the service instance description, the service design description and the service specification are sufficiently mature, they shall be published in the service registry and interested consumers may look them up.

3 SERVICE SPECIFICATION

The purpose of the service specification is to collect the results of service identification and service design activities. The aim is to document the key aspects of a dedicated service at the logical level:

- the operational and business context of the service:
  - requirements for the service (e.g. information exchange requirements);
  - involved nodes: which operational components provide/consume the service;
  - operational activities supported by the service;
  - relation of the service to other services.
- the service description:
  - service interfaces;
  - service interface operations;
  - service payload definition;
  - service dynamic behaviour description;
- service provision and validation aspects.

The purpose of the service specification is to provide a holistic overview of a service and its building blocks at the logical level. The service specification consists of:

- (mandatory) a service specification document - a human readable documentation of the service key aspects;
- (mandatory) a service specification XML - a formal description of the service specification in a more formal manner, including a formal definition of the service data model;
- (optional) a model based description - e.g. a UML model describing the service interfaces, operations and data structures.

  The service specification document as well as the service specification XML might re-use artefacts produced in the model based description.

Note: The service specification is intended to be technology-agnostic. The service specification shall not describe the details of a specific service implementation. For that purpose, a service design description must also be provided, where the actual realisation of the service with a dedicated technology shall be described.

3.1 SERVICE SPECIFICATION DOCUMENT

The purpose of the service specification document is to document in human readable manner all the information comprising a service specification. It should be noted that a service specification document describes one dedicated version of one dedicated service in detail at logical level.

The service specification document describes a well-defined baseline of the service and clearly identifies the service version. In this way, it supports the configuration management process.

The service specification document provides also the foundation material for the future standardisation process.

A template is available to assure a certain uniformity of service specification documents produced by different authors.

3.2 SERVICE SPECIFICATION TEMPLATE

The service specification template ANNEX E shall support the service architects in creating a document based description of the services at a high level of abstraction. The template prescribes a structure of sections (to be completed by the author of the service specification), and for each section descriptive instructions for the intended content.

The basic structure of the service specification template is replicated in the following subsections.

3.2.1 INTRODUCTION

The introduction section contains the usual basic information, such as purpose of the document, intended readership, etc.

3.2.2 SERVICE IDENTIFICATION

The service identification section provides a tabular overview of mainly administrative attributes needed for identification and lookup of the service. Examples: name, identifier, version, author, key words of the service specification.

The service identifier shall be in the form of an MRN [2] (Maritime Resource Name).

3.2.3 OPERATIONAL CONTEXT

The operational context section describes the context of the service from an operational perspective.

The operational context description should be based on the description of the operational model, consisting of a structure of operational nodes.

If the service is part of one or more MSPs [3] they should be referenced in this section.

Optionally, a simple high level use case, described in layman's terms, could be provided as an introduction to this section.

The operational context shall be a description of how the service supports interaction among operational nodes. This can be achieved in two different levels of granularity:

- A description of how the service supports the interaction between operational nodes;

  This basically consists of an overview about which operational nodes shall provide the service and which operational nodes will consume the service.

- A more detailed description stating what operational activities the service supports in a process model.

Moreover, the operational context shall describe any requirement the service shall fulfil or adhere to. This refers to functional as well as non-functional requirements at high level (business/regulatory requirements, system requirements, user requirements). Especially, information exchange requirements are of much interest since the major objective of services is to support interaction between operational nodes.

The source material for the operational context description should ideally be provided by operational users and is usually expressed in dedicated requirements documentation. Any applicable documents shall be mentioned in the References section. If no requirements documents are available, then the basic requirements for the service shall be defined in the service specification document in tabular form.

The service shall be linked to at least one requirement.

3.2.4 SERVICE OVERVIEW

This section aims at providing an overview of the main elements of the service. The elements in this view are all usually created by means of an UML modelling tool.

Architectural elements applicable for this description are:

- Service - the element representing the service, as a whole;
- Service Interfaces - the communication mechanisms of the service, i.e. interaction mechanisms between service provider and service consumer;

  Defined by allocating service operations to either the provider or the consumer of the service.

- Service Operations - describe the logical operations used to access the service;

- Service Operations Parameter Definitions - identify data structures being exchanged via Service Operations.

These elements may be depicted in one or several diagrams. Which and how many diagrams are needed depends on the chosen architecture description framework and complexity of the service.

The service overview may be described by using an UML diagram[1] that illustrates the service interfaces with their operations and their allocation to service provider and service consumer. This information should also be provided in tabular form.

It is also recommended to describe the considerations resulting in the selection of a certain Message Exchange Pattern (MEP) for the service interfaces.

A service interface supports one or several service operations. Depending on the Message Exchange Pattern, service operations are either to be implemented by the service provider (e.g. in a Request/Response MEP, query operations are provided by the service provider – the service consumer uses them in order to submit query requests to the service provider), or by the service consumer (e.g. in a Publish/Subscribe MEP, publication operations are provided by the service consumer – the service provider uses them to submit publications to the service consumer). This distinction shall be clearly visualised. For each service interface, it shall be stated whether it is provided or requested by the service. A service provides at least one service interface.

3.2.5 SERVICE DATA MODEL

This section shall describe the data structures to be exchanged between service providers and consumers. The data model shall provide enough information allowing to implement the service based on this information, but on the other hand it should describe the data structures sufficiently abstract; this means, it should be avoided to list all details or to define technology-specific data types. It is recommended to visualise the data structures by means of UML diagrams. The complete information model (in the form of logical data structures) should be shown using diagram(s) and explanatory tables. It is mandatory to give a description of each entity item (class), its attributes and the relations between entity items after each diagram that shows data items.

If the service data model is related to an external data model (e.g. being a subset of a standard data model, e.g. typically based on an S-100 specification), then the service data model shall refer to it: each data item of the service data model shall be mapped to a data item defined in the external data model. This mapping may be added in the same table that describes the data items and their attributes and associations. The idea is that, when reading the service specification (including the logical service data model), the reader must clearly understand the payload structures. If the service re-uses structures of an external data model, then these structures can be referred to rather than replicated in the service specification. The tabular presentation of the payload allows provision of references to an externally defined model.

A formal description of the service data model shall be attached to the service specification document in an annex. The formal description should be presented by means of an XML schema. XML schemas are proposed to provide some kind of formalism at the logical level, in order to enforce some uniformity across different service specifications and not primarily for the possibility of data validation.

Alternatively, a formal description may also be provided in another textual format, if XSD is considered as not appropriate.**Note**: The S-100 specification provides examples in its Appendix 4a-D and Appendix 9-B, about S-100 based data models formulated in XML schema format. In addition to the data model exchanged between service providers and consumers, this section may optionally also contain a description of the internal data model, as it seems appropriate to the service provider and/or the service consumer side. Such a description might be helpful for the understanding as it provides additional information of how the service might be built. However, this internal service data model must be declared as exemplary only – it is not an authoritative part of the service specification.

1 e.g. in NATO Architectural Framework (NAF), a NSOV-2 diagram could be used.

3.2.6 SERVICE INTERFACE SPECIFICATION

This section describes the details of each service interface. The static interface description provided in this section – together with the service data model provided in previous section – is one of the core parts of the service specification since it describes how the interfaces shall be constructed.

Architectural elements applicable for this description are:

- service interfaces;
- service operations - functions or procedures which enable programmatic communication with a Service via a Service interface;
- parameters - constants or variables passed into or out of a service interface as part of the execution of a service operation.

A service may have one or more service interfaces. Each of them shall be described in a separate sub-section. The sub-section title shall contain the service interface name.

For each service interface, the purpose, message exchange pattern and architecture of the Interface shall be described.

A service interface supports one or several service operations. Each of them shall be described in a separate sub-section. The sub-section title shall contain the name of the operation. Each service operation sub-section shall contain the following information:

- functionality - shall include a textual description of the operation functionality. In most instances this will be the same as the operation description taken from the UML modelling tool;
- parameters - shall describe the logical data structure of input and output parameters of the operation (payload) by using UML diagrams (which are usually sub-sets of the service data model described previously) and explanatory tables.

  It is mandatory to provide a table with a clear description of each service operation parameter and the information about which data types defined in the service data model are used by the service operation in its input and output parameters.

  Note: While the descriptions provided in the service data model shall explain the data types in a neutral format, the descriptions provided here shall explicitly explain the purpose of the parameters for the operation.

3.2.7 SERVICE DYNAMIC BEHAVIOUR

This section describes the interactive behaviour between service interfaces (interaction specification) and, if required, between different services (orchestration). Architectural elements applicable for this description are:

- service interaction specifications;
- service state machines;
- service orchestration.

Following types of views and UML diagrams can be used to describe the dynamic behaviour[2]:

- sequence diagrams;
- interaction diagrams;
- state machine diagrams.

2 e.g. in NATO Architectural Framework (NAF), state model and interaction specification (NAF3.1) or NSOV-5 Service constraints, state model could be used.

If the service consists of more than one interface, then the service dynamic behaviour description shall provide a dedicated sub-section per service interface. The minimum content of such sub-section is some information about the dynamic aspects of the service interface. Each operation shall be exposed on at least one diagram.

Additional sub-sections may be provided if there is the need to describe service composition or orchestration.

3.2.8 SERVICE PROVISIONING (OPTIONAL)

This section should describe the way services are planned to be provided and consumed. It is labelled optional since one of the key aspects of service-orientation is to increase flexibility of the overall system by separating the definition of services from their implementation. This means that a service can be provided in several different contexts that are not necessarily known at the time, when the service is designed.

3.2.9 REFERENCES

The References section contains a list of all documents referred to by the service specification (e.g. requirements documents (if any)).

3.3 SERVICE SPECIFICATION XSD STRUCTURE

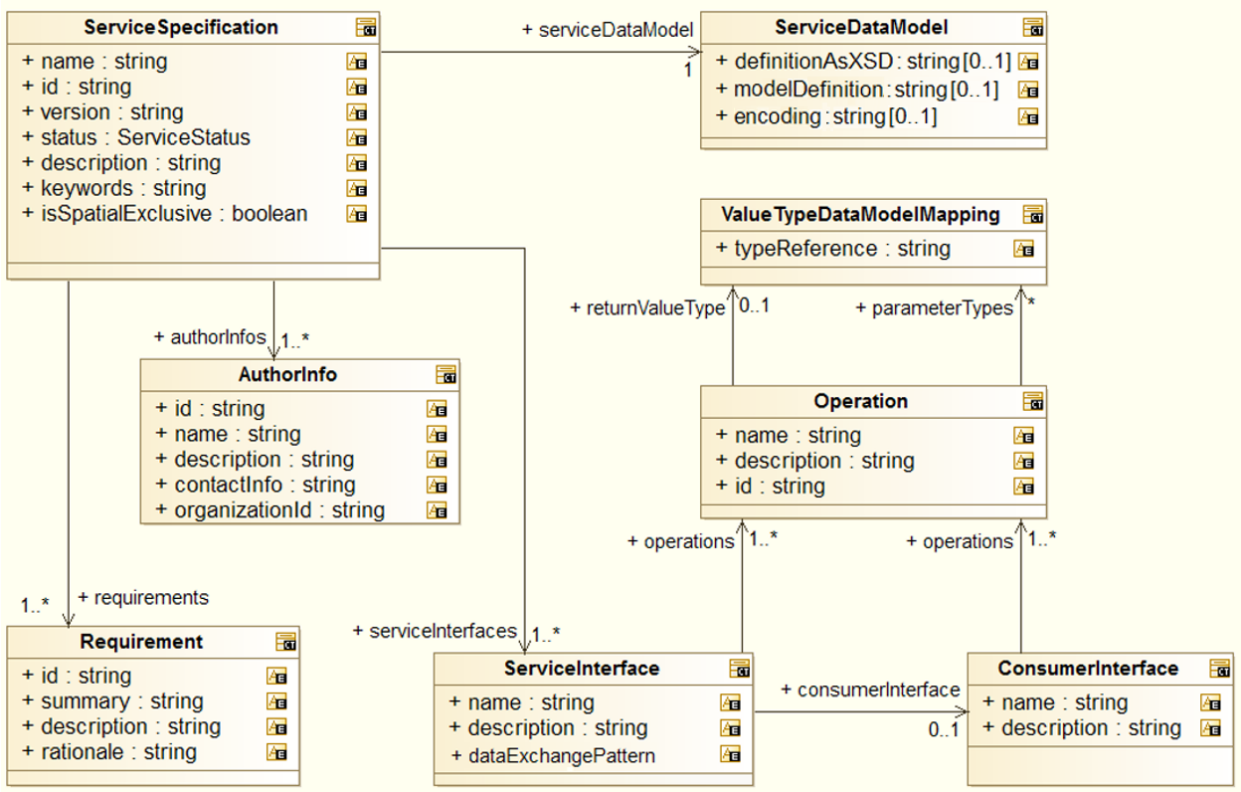


Figure 4 Structure of the Service Specification

Figure 4 gives an overview about the formal description of the service specification. The individual items are described in the Table 1.

Note that this formal description of a service specification is intentionally kept simple and plain. For most described objects, many more attributes could be added and standardized but, in order to get as much adoption as possible, the entry barrier should be low.

The service specification XSD file is presented in ANNEX A.

Table 1 Information elements of the Service Specification

Type Name		Description	
ServiceSpecification		A service specification describes one dedicated service at logical level in a technology-independent way. The service specification identifies a service by its id and version. The service specification refers to requirements for the service, defines a service data model at logical level, defines the service interface(s) and provides information about the author(s).	
	Element Name	**Type**	**Description**
	Name	string	The human readable service name. The service name should be at maximum a one line brief label for the service. Newer versions of the same service specification should not change the name.
	Id	string	Globally unique identification of the service. Newer versions of the same service specification shall not change the id.
	version	string	Version of the service specification. A service specification is uniquely identified by its id and version. Any change in the service data model or in the service interface definition requires a new version of the service specification
	status	enumeration	Status of the service specification. The status field has one of the following values: • provisional; • released; • deprecated; • deleted.
	description	string	A human readable short description of the service. The description shall contain an abstract of what a service implementing this specification would do.
	keywords	string	A list of keywords associated with the service.
	isSpatialExclusive	boolean	Flag to indicate whether the service shall be 'spatial exclusive'. 'Spatial exclusiveness' means that just one service instance of the same service specification providing the same technical design is allowed to be registered for a certain geographical area.
	requirements	Requirement	Refers to requirements specifications for the service. Business requirements, functional and non-functional requirements should be listed here. At least one requirement is mandatory.
	authorInfos	AuthorInfo	Refers to administrative information about the authors of the service. It is mandatory to provide at least one author information.
	serviceInterfaces	ServiceInterface	Refers to the definition of service interfaces. At least one service interface shall be defined.
	serviceDataModel	ServiceDataModel	Mandatory reference to the definition of the logical service data model.
Type Name		**Description**	

Requirement		A requirement that the service shall fulfil.	
	Element Name	**Type**	**Description**
	id	string	Globally unique requirement identification
	name	string	Human readable requirement name/summary. Shall not be longer than one line.
	text	string	The human readable requirement text. Usually formulated in form of a 'shall'-statement.
	rationale	string	Rationale for this requirement. Textual explanation of why this requirement exists. Provides background information about the need of the service.
	reference	string	Optional information about where the requirement was originally stated. If the requirement comes from external documents, this attribute shall refer to this source.
	author	AuthorInfo	Optional reference(s) to administrative information about the author(s) of the requirement.
Type Name		**Description**	
AuthorInfo		Describes an author of a service specification or requirement.	
	Element Name	**Type**	**Description**
	id	string	Unique identifier of the author.
	name	string	Human readable name of the author.
	description	string	Human readable description of the author.
	contactInfo	string	Human readable contact information of the author.
	organizationId	string	Unique identifier of the organization, the author belongs to.
Type Name		**Description**	
ServiceInterface		Specification of a service interface. One service can offer several interfaces, e.g. both a request/response interface and a publish/subscribe interface at the same time. Different interfaces will usually provide different service operations.	
	Element Name	**Type**	**Description**
	name	string	Human readable service interface name. The name shall be no longer than one line.
	description	string	Human readable description of the service interface.

dataExchangePattern	enumeration	Message exchange pattern can be one of: • ONE_WAY data are sent in one direction, from service consumer to service provider, without confirmation. • REQUEST_RESPONSE service consumer sends request to service provider and expects to receive a response from the service provider. • REQUEST_CALLBACK (asynchronous REQUEST_RESPONSE) service consumer sends a request to service provider; response is provided asynchronously in an independent call to the service. • PUBLISH_SUBSCRIBE service consumer subscribes at service provider for receiving publications sent out by the service provider. • BROADCAST service provider distributes information independently of any consumers.
operations	Operation	Refers to the specification of service operations supported by the service interface. At least one operation shall be defined.
consumerInterfaces	ConsumerInterface	Optional reference to an interface definition that must be provided by the service consumer to complement the service interface. Especially if a publish/subscribe service interface is designed, it is necessary to describe what the service expects to be available on the subscriber side.

Type Name	Description
ConsumerInterface	Interface specification that is expected to be provided by the service consumer. For example, if a publish/subscribe service interface is designed, it is necessary to describe what the service expects to be available on the subscriber side.

Element Name	Type	Description
name	string	Human readable interface name. The name shall be no longer than one line.
description	string	Human readable description of the interface.
operations	Operation	Refers to the specification of service operations supported by the consumer interface. At least one operation shall be defined.

Type Name	Description
Operation	Definition of a service operation. Operations allow a service consumer to interact with the service. An operation describes a dedicated function of the service or the consumer.

	Element Name	Type	Description
	name	string	Human readable operation name. The name shall be no longer than one line.
	description	string	Human readable description of the operation.
	returnValueType	ValueTypeData-ModelMapping	Optional definition of the return value for the operation. The return value could be a business object or a simple status code. The return value data type must be defined in the logical service data model.
	parameterTypes	ValueTypeData-ModelMapping	Definition of one or more parameters for the operation. This could be business objects or simple types. Parameters must be defined in the logical service data model.

Type Name	Description
ValueTypeDataModel-Mapping	Definition of a data type by providing a reference into the logical service data model. A value type data model mapping is used either in a service operation parameter or return value.

	Element Name	Type	Description
	typeReference	string	Reference to the logical service data model. It references a type (or element, though type is preferred) in the logical service model by the type's name attribute.

Type Name	Description
ServiceDataModel	The serviceDataModel is a logical model. It is formally described in the sub-element modelDefinition to achieve interoperability and decouple it from implementing physical data models described in e.g. SOAP or REST. Encodings are formally described using sub-element encoding. The model can either be described in-line, or existing schemata can be imported. One service specification has one logical service model. Sub-element modelDefinition has to be provided.

	Element Name	Type	Description
	definitionAsXSD (deprecated)	String	The definition of the service data model described in XSD. Deprecated – only for backwards compatibility reasons.
	modelDefinition	String	The definitionof the service data model described.
	encoding	String	The encoding of the attribute modelDefinition. To be compliant to S100, refer to the S100 Data formats as described in the S100 Specification Appendix 4aD.

4 SERVICE TECHNICAL DESIGN

4.1 SERVICE DESIGN DESCRIPTION DOCUMENT

The purpose of the service design description is to document in human readable manner all the information comprising the technical design of a service. This document shall provide a detailed description of how a service

shall be realized with a certain technology. For the technology-independent information this document shall refer to the service specification document, rather than replicating any information.

Note: In theory one service technical design may describe several different kinds of services. In this case, all service specifications shall be referenced in the service design description. On the other hand, it is obvious that one service specification may be referenced by several different service design descriptions. This is the case when a service shall be implemented/provided by using different technologies.

To assure a certain uniformity of service technical design description documents produced by different authors, the document shall be aligned with the service design description template.

4.2 SERVICE DESIGN DESCRIPTION TEMPLATE

The service design description template ANNEX F shall support the service architects/designers in creating a document based description of the service technical design. The template prescribes a structure of sections (to be completed by the author of the service technical design), and for each section descriptive instructions for the intended content.

The basic structure of the service design description template is replicated in the following subsections.

4.2.1 INTRODUCTION

The introduction section contains the usual basic information, such as purpose of the document, intended readership, etc.

4.2.2 SERVICE DESIGN IDENTIFICATION

The service design identification section provides a tabular overview of mainly administrative attributes needed for identification and lookup of the service design. Example content of this section: reference to service specification; name, identifier and version of the technical design; author (vendor information), key words, etc.

4.2.3 TECHNOLOGY INTRODUCTION

The technology introduction section contains a basic background about the chosen technology. In most cases this will be a short description of basic technology aspects accompanied with appropriate references to standard documents and best practice descriptions.

4.2.4 SERVICE DESIGN OVERVIEW

This section aims at providing an overview of the main elements of the service design and a mapping of the design elements to the service specification elements. The elements in this view are all usually created by means of an UML modelling tool.

Architectural elements applicable for this description are:

- service - the element representing the whole service;
- service interfaces - the communication mechanisms of the service, i.e. interaction mechanisms between service provider and service consumer;

  Defined by allocating service operations to either the provider or the consumer of the service.
- service operations - describe the operations used to access the service;
- service operations parameter definitions - identify data structures being exchanged via Service Operations.

The above elements may be depicted in one or several diagrams. Which and how many diagrams are needed, depends on the chosen architecture description framework, the chosen technology, and the complexity of the service.

If the structure of the service design largely follows the service specification then it is not necessary to replicate identical diagrams here in this section; in this case, this section shall contain references to the service specification document. However, it is assumed that in many cases, depending on the chosen technology, the

actual interface and/or operation names (and structuring) are not fully identical to the abstract definition given in the service specification.

The service design overview may be described by using an UML diagram[3] that illustrates the service interfaces with their operations and their allocation to service provider and service consumer. This information should also be provided in tabular form. Furthermore, it shall be described how the specified Message Exchange Patterns (MEP) are realised with the chosen technology.

4.2.5 PHYSICAL DATA MODEL

This section provides a detailed description of the data structures exchanged between service provider and service consumer. This description shall also include a mapping of the data structures to the service data model provided in the service specification.

The service design description template does not prescribe a detailed format for this section. Allowed presentations of the physical data model **include**

- UML diagrams representing the data structures including detailed physical data type descriptions at attribute level;
- a file describing the data structures (e.g. XML/XSD, JSON);
- tabular presentations.

Any mixture of the above formats is allowed. A S-100 compliant specification should refer to the Dataset Discovery Metadata in order to link to product specifications and S-100 compliant data formats.

If the physical service data model is related to an external data model (e.g. being a subset of a standard data model, i.e. based on an S-100 specification), then this section shall refer to it; each data item of the physical data model shall be mapped to a data item defined in the external data model. This mapping may be added in the same table that describes the data items and their attributes and associations.

4.2.6 SERVICE INTERFACE DESIGN

The structure of the service interface design section is identical to the structure of the service interface specifications section in the service specification document (see 3.2.6). This section may be limited to references to the service specification document, if all the following conditions are fulfilled:

- the service design reflects the service interfaces in a 1:1 manner;
- the service interfaces are sufficiently described in the service specification;
- the physical data model (section 4.2.5) contains an unambiguous mapping of all payload data items of the service specification to the detailed physical data items.

4.2.7 SERVICE DYNAMIC BEHAVIOUR

This section describes the interactive behaviour between service interfaces (interaction specification) and, if required, between different services (orchestration).

Following types of views and UML diagrams can be used to describe the dynamic behaviour:

- sequence diagrams;
- interaction diagrams;
- state machine diagrams;

This section is especially relevant, if the service design structure (see section 4.2.4) differs from the service structure introduced in the service specification. If designed service interfaces and operations are equivalent to

3 e.g. in NATO Architectural Framework (NAF), a NSOV-2 diagram could be used

those of the service specification, and if the dynamic behaviour is sufficiently described in the service specification, then this section may be limited to references to the service specification document.

4.2.8 REFERENCES

The References section contains a list of all documents referred to by the service design description (e.g. service specification or requirements documents). At least the service specification document needs to be referenced.

4.3 SERVICE DESIGN DESCRIPTION XSD STRUCTURE

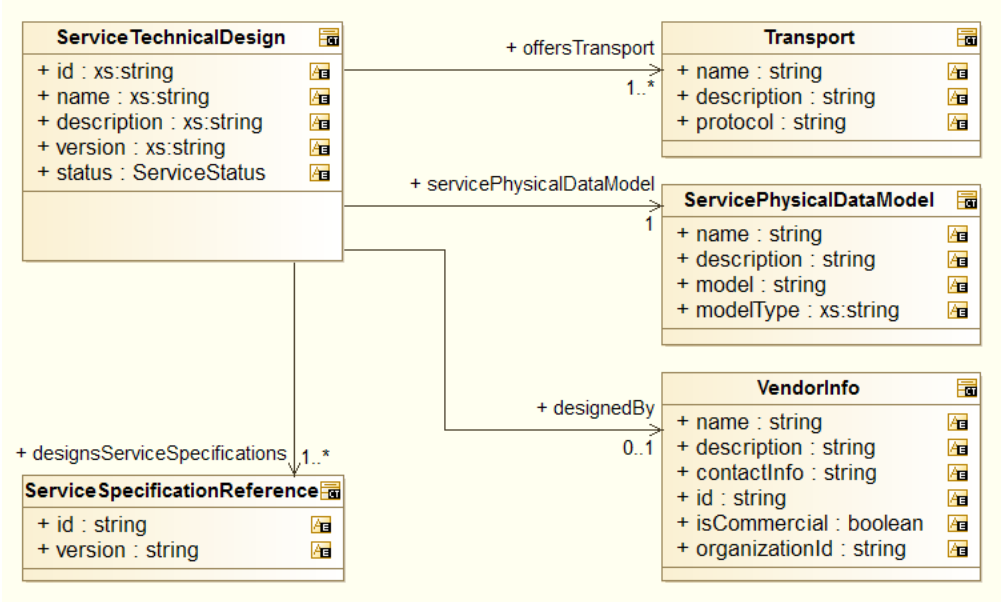


Figure 5 Structure of the Service Technical Design Description

Figure 5 gives an overview about the formal description of the service design description. A description of each item is given in the Table 2.

Note: This formal description of a service technical design is intentionally kept simple and plain. For all described objects, many more attributes and related objects could be added and standardized, but to get as much adoption as possible the entry barrier should be kept low, and therefore, intentionally, several aspects have been left out.

The service design description schema is presented in ANNEX B.

Table 2 Information Elements of the Service Design Description

Type Name		Description	
ServiceDesign		A service design description.	
	Element Name	**Type**	**Description**
	name	string	The human readable name of the service design. The service name should be at maximum a one line brief label. Newer versions of the same service design should adopt the same name.
	id	string	Globally unique identification of the service design. Newer versions of the same service design shall adopt the same id.
	version	string	Version of the service design. A service design is uniquely identified by its id and version. Any change in the service physical data model or in the service specification reference requires a new version of the service design.

status	enumeration	Status of the service design. The status field has one of the following values: • provisional; • released; • deprecated; • deleted.
description	string	A human readable short description of the service design. The description shall contain an abstract of what a service implementation does.
designsService-Specifications	ServiceSpecificationReference	Refers to service specification(s) that is/are realised by this service design. As a minimum, one service specification shall be referenced. One service design may realise several service specifications (either different versions of one specification, or even different specifications).
offersTransport	Transport	Refers to transport technologies offered by the service design. At least one reference shall be provided.
designedBy	VendorInfo	Mandatory reference to information about the author of the service design.
servicePhysicalData-Model	ServicePhysical-DataModel	Mandatory reference to the service physical data model description.

Type Name	Description
ServiceSpecification-Reference	A reference to the service specification that is realised by the service design. – It has the id and the version of the respective service specification.

Element Name	Type	Description
id	string	Identification of the service specification realised by the service design.
version	string	Version of the service specification realised by the service design.

Type Name	Description
Transport	Definition of the transport protocol used by the service design.

Element Name	Type	Description
name	string	Human readable name.
description	string	Human readable description of the transport protocol used by the service design.
protocol	string	A non-formal string representation of the transport (e.g. http/rest, http/soap, …) that provides enough information to a service consumer to be able to connect.

Type Name	Description
VendorInfo	Describes the vendor providing the service design.

Element Name	Type	Description
id	string	Unique identification of the vendor.
name	string	Human readable vendor name. The name shall be no longer than one line.

	description	string	Human readable description of the vendor.
	contactInfo	string	Human readable contact information of the vendor.
	organizationId	string	Unique identifier of the organization, the vendor belongs to.
	isCommercial	boolean	Optional indication on the commercial status of the vendor.

Type Name	Description
ServicePhysicalDataModel	The ServicePhysicalDataModel describes the data model for the service design. The ServicePhysicalDataModel describes in detail all the data structures being exchanged when service consumers interact with a service instance that implements this design.

	Element Name	Type	Description
	name	string	Human readable model name. The name shall be no longer than one line.
	description	string	Human readable description of the model.
	model	string	The model can e.g. be a WSDL file, a JSON API, or the like. It is recommended to wrap the model in a CDATA section, and provide enough information in the name and description to make clear how to deal with the content in model.
	modelType	string	The modelType should contain e.g. an abbreviation that indicates what technology is used to describe the mode, e.g. WSDL, JSON.

5 SERVICE INSTANCE

5.1 SERVICE INSTANCE DESCRIPTION DOCUMENT

The purpose of the service instance description is to document in human readable manner all the information specific to a certain implementation and instantiation of a service. This document shall provide a detailed description of how a service is realised. In most cases, this document will be rather short, since it is expected that the implementation follows the technical design and it is not supposed to replicate any information from the service design description document.

Note that one service implementation may be deployed several times at different access points (thus forming several different service instances). In this case, several service instance description XML files need to be produced – one for each deployed instance, whereas the service instance description document can be identical (if all instances behave equivalently).

To assure a certain uniformity of service instance description documents produced by different authors, the document shall be aligned with the service instance description template.

5.2 SERVICE INSTANCE DESCRIPTION TEMPLATE

The service instance description template (see ANNEX F) shall support the service developers in creating a document based description of the service implementation and instantiation. The template prescribes a structure of sections (to be completed by the service implementer) and for each section descriptive instructions for the intended content.

The basic structure of the service instance description template is replicated in the following subsections.

5.2.1 Introduction

The introduction section contains the usual basic information, such as purpose of the document or intended readership.

5.2.2 Service Instance Identification

The service identification section provides a tabular overview of mainly administrative attributes about the service instance. Example content of this section: reference to service technical design; name, identifier and version of the implementation and instance; author (vendor information), key words, etc.

5.2.3 Service Implementation and Instance Details

This section describes any information that appears useful for the understanding of the service implementation in general and of the actual service instance in particular. This may include internal design decisions, required configuration data, deployment pre-requisites, etc.

The service instance description template does not prescribe a detailed format for this section.

5.2.4 Release Notes

This section describes the release notes of the service instance. It shall contain at least the following set of information:

- release identification and date;
- feature list:
  - added features;
  - changed features;
  - removed features.
- bug list:
  - known open bugs;
  - resolved bugs.

The service instance description template does not prescribe a detailed format for this section.

5.2.5 References

The References section contains a list of all documents referred to by the service instance description (e.g. service specification, service design, requirements documents, etc.). At least service specification and service design documents need to be referenced.

5.3 SERVICE INSTANCE DESCRIPTION XSD STRUCTURE

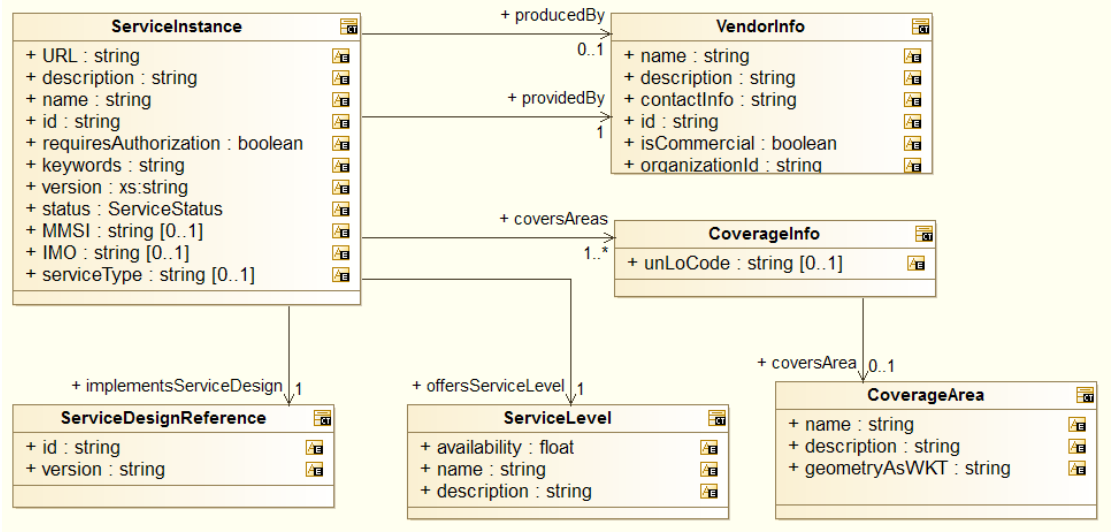


Figure 6 Structure of the Service Instance Description

Figure 6 gives an overview about the formal description of the service instance description. The individual items are described in the Table 3.

Note that this formal description of a service instance is intentionally kept simple and plain. For all described objects, many more attributes and related objects could be added and standardized, but to get as much adoption as possible the entry barrier should be low and therefore some aspects (e.g. when a service would be available) have been left out.

The service instance description schema id presented in ANNEX C.

Table 3 Information Elements of the Service Instance Description

Type Name			Description
ServiceInstance			A service instance description. One service implementation may be deployed at several places by same or different service providers; each such deployment represents a different service instance.
	Element Name	**Type**	**Description**
	name	string	The human readable name of the service instance. The service name should be at maximum a one line brief label. Newer versions of the same service specification should adopt the same name.
	id	string	Globally unique identification of the service instance. Newer versions of the same service instance shall adopt the same id.
	version	string	Version of the service instance. A service instance is uniquely identified by its id and version. Any change in the service design reference requires a new version of the service instance.

status	enumeration	Status of the service instance. The status field has one of the following values: • provisional; • released; • deprecated; • deleted.
description	string	A human readable short description of the service instance. The description shall contain an abstract of what a service implementation does and what the service consumer should know about how the service implementation works.
keywords	string	A list of keywords associated to the service.
URL	string	URL that describes where the service endpoint is located
MMSI	string	Optional Maritime Mobile Service Identity.
IMO	string	Optional International Maritime Organization (IMO) number
serviceType	string	Optional field to categorize the service type. Example: "VIS".
requiresAuthorization	boolean	Indicates whether authorization is required or not.
implementsService-Design	ServiceDesignReference	Refers to the service design that is implemented by this service instance. Exactly one service design shall be referenced.
coversAreas	CoverageInfo	Mandatory reference to the geographical area covered by the service instance.
offersServiceLevel	ServiceLevel	Refers to the definition of the service level fulfilled by the service instance. Exactly one service level definition shall be provided.
producedBy	VendorInfo	Optional reference to information about the producer of the service implementation.
providedBy	VendorInfo	Mandatory reference to information about the service provider of the service instance.

Type Name	Description
ServiceDesignReference	A reference to the service design that is implemented by the service instance. – It has the id and the version of the respective service design.

Element Name	Type	Description
id	string	Identification of the service design implemented by the service instance.
version	string	Version of the service design implemented by the service instance.

Type Name		Description	
CoverageInfo		Defines a geographical area from which the service instance is accessible. This is a choice between a geographical area defined by co-ordinates or the United Nations Code for Trade and Transport Locations (UN/LOCODE). One of the two options must be provided. Worldwide accessibility is indicated by a 'coversArea' element with a missing 'geometryAsWKT' element.	
	Element Name	**Type**	**Description**
	coversArea	CoverageArea	Identification of the service design implemented by the service instance.
	unLoCode	string	Version of the service design implemented by the service instance.
Type Name		**Description**	
CoverageArea		Defines a geographical area from which the service instance is accessible.	
	Element Name	**Type**	**Description**
	name	string	Human readable name of the coverage area, e.g. a well-known name like 'Bermuda Triangle'. The name shall be no longer than one line.
	description	string	Human readable description of the coverage area.
	geometryAsWKT	string	A polygon described in WKT (Well Known Text) with coordinates in co-ordinate reference system EPSG:4326, e.g. POLYGON(LON1 LAT1, LON2 LAT2, LON3, LAT3, LON1 LAT1). If the element is empty, the default is the whole world.
Type Name		**Description**	
ServiceLevel		Defines the service availability level.	
	Element Name	**Type**	**Description**
	name	string	Human readable service level name. The name shall be no longer than one line.
	description	string	Human readable description of the service level.
	availability	float	Indicates the guaranteed availability of the service in %, (e.g. 99.9).
Type Name		Description	
VendorInfo		Describes the vendor producing and/or providing the service instance.	
	Element Name	**Type**	**Description**
	id	string	Unique identification of the vendor.
	name	string	Human readable vendor name. The name shall be no longer than one line.
	description	string	Human readable description of the vendor.
	contactInfo	string	Human readable contact information of the vendor.
	organizationId	string	Unique identifier of the organization, the author belongs to.
	isCommercial	boolean	Optional indication on the commercial status of the vendor.

6 GOVERNANCE

It is anticipated that some kind of governance will be needed in the area of service management. This includes questions about the process to

- decide about maturity of service specifications;
- decide about the scope of service specifications;
- decide about the evolution of service specifications;
- decide about the life cycle of service specifications and service instances;
- decide about conformance of service instances to specifications.

The definition and description of governance structures and procedures are outside the scope of this document.

7 DEFINITIONS

The definitions of terms used in this IALA Guideline can be found in the International Dictionary of Marine Aids to Navigation (IALA Dictionary) at http://www.iala-aism.org/wiki/dictionary and were checked as correct at the time of going to print. Where conflict arises, the IALA Dictionary should be considered as the authoritative source of definitions used in IALA documents.

7.1 TERMINOLOGY

Term	Definition
External Data Model	Describes the semantics of the 'maritime world' (or a significant part thereof) by defining data structures and their relations. This could be at logical level (e.g. in UML) or at physical level (e.g. in XSD schema definitions), as for example standard data models, or S-100 based data produce specifications.
Message Exchange Pattern	Describes the principles two different parts of a message passing system (in our case: the service provider and the service consumer) interact and communicate with each other. Examples: In the Request/Response MEP, the service consumer sends a request to the service provider to obtain certain information; the service provider provides the requested information in a dedicated response. In the Publish/Subscribe MEP, the service consumer establishes a subscription with the service provider to obtain certain information; the service provider publishes information (either in regular intervals or upon change) to all subscribed service consumers.
Operational Activity	An activity performed by an operational node. Examples of operational activities in the maritime context are: Route Planning, Route Optimization, Logistics, Safety, Weather Forecast Provision, …
Operational Model	A structure of operational nodes and associated operational activities and their inter-relations in a process model.
Operational Node	A logical entity that performs activities. Note: nodes are specified independently of any physical realisation. Examples of operational nodes in the maritime context are: Maritime Control Centre, Maritime Authority, Ship, Port, Weather Information Provider, …

Term	Definition
Service	The provision of something (a non-physical object), by one, for the use of one or more others, regulated by formal definitions and mutual agreements. Services involve interactions between providers and consumers, which may be performed in a digital form (data exchanges) or through voice communication or written processes and procedures.
Service Consumer	A service consumer uses service instances provided by service providers. All users within the maritime domain can be service customers, e.g. ships and their crew, authorities, VTS stations, organizations (e.g. meteorological), commercial service providers, etc.
Service Data Model	Formal description of one dedicated service at logical level. The service data model is part of the service specification. Is typically defined in UML and/or XSD. If an external data model exists (e.g. a standard data model), then the service data model shall refer to it: each data item of the service data model shall be mapped to a data item defined in the external data model.
Service Design Description	Documents the details of a service technical design (most likely documented by the service implementer). The service design description includes (but is not limited to) a service physical data model and describes the used technology, transport mechanism, quality of service, etc.
Service Implementation	The provider side implementation of a dedicated service technical design (i.e. implementation of a dedicated service in a dedicated technology).
Service Implementer	Implementers of services from the service provider side and/or the service consumer side. Anybody can be a service implementer but mainly this will be commercial companies implementing solutions for shore and ship.
Service Instance	One service implementation may be deployed at several places by same or different service providers; each such deployment represents a different service instance, being accessible via different URLs.
Service Instance Description	Documents the details of a service implementation (most likely documented by the service implementer) and deployment (most likely documented by the service provider). The service instance description includes (but is not limited to) service technical design reference, service provider reference, service access information, service coverage information, etc.
Service Interface	The communication mechanism of the service, i.e. interaction mechanism between service provider and service consumer. A service interface is characterised by a message exchange pattern and consists of service operations that are either allocated to the provider or the consumer of the service.
Service Operation	Functions or procedure which enables programmatic communication with a service via a service interface.

Term	Definition
Service Physical Data Model	Describes the realisation of a dedicated service data model in a dedicated technology. This includes a detailed description of the data payload to be exchanged using the chosen technology. The actual format of the service physical data model depends on the chosen technology. Examples may be WSDL and XSD files (e.g. for SOAP services) or swagger (Open API) specifications (e.g. for REST services). If an external data model exists (e.g. a standard data model), then the service physical data model shall refer to it: each data item of the service physical data model shall be mapped to a data item defined in the external data model. To prove correct implementation of the service specification, there shall exist a mapping between the service physical data model and the service data model. This means, each data item used in the service physical data model shall be mapped to a corresponding data item of the service data model. (In case of existing mappings to a common external (standard) data model from both the service data model and the service physical data model, such a mapping is implicitly given.)
Service Provider	A service provider provides instances of services according to a service specification and service instance description. All users within the maritime domain can be service providers, e.g. authorities, VTS stations, organizations (e.g. meteorological), commercial service providers, etc.
Service Specification	Describes one dedicated service at logical level. The Service Specification is technology-agnostic. The Service Specification includes (but is not limited to) a description of the Service Interfaces and Service Operations with their data payload. The data payload description may be formally defined by a Service Data Model.
Service Specification Producer	Producers of service specifications in accordance with the service documentation guidelines.
Service Technical Design	The technical design of a dedicated service in a dedicated technology. One service specification may result in several technical service designs, realising the service with different or same technologies.
Service Technology Catalogue	List and specifications of allowed technologies for service implementations. Currently, SOAP and REST are envisaged to be allowed service technologies. The service technology catalogue shall describe in detail the allowed service profiles, e.g. by listing communication standards, security standards, stacks, bindings, etc.
Spatial Exclusiveness	A service specification is characterised as ‘spatially exclusive’, if in any geographical region just one service instance of that specification can be registered per technology. The decision, which service instance (out of several available spatially exclusive services) shall be registered for a certain geographical region, is a governance issue.

8 ACRONYMS

API	Application Programming Interface
EPSG	European Petroleum Survey Group (a spatial reference system)
IALA	International Association of Marine Aids to Navigation and lighthouse Authorities - AISM
IMO	International Maritime Organization
JSON	JavaScript Object Notation

LAT	Latitude
LON	Longitude
MCP	The Maritime Connectivity Platform (formerly the Maritime Cloud)
MEP	Message Exchange Pattern
MRN	Maritime Resource Name
MSP	Maritime Service Portfolio
NAF	NATO Architectural Framework
NATO	North Atlantic Treaty Organisation
NSOV	NATO Service-Oriented View
OGC	Open Geospatial Consortium
REST	Representational State Transfer
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SSD	Service Specification Document
S-100	Universal Hydrographic Data Model (IHO)
UML	Unified Modelling Language
UN/LOCODE	United Nations Code for Trade and Transport Locations
URL	Uniform Resource Locator
VTS	Vessel Traffic Services
WFS	Web Feature Service
WMS	Web Map Service
WSDL	Web Service Definition Language
XML	Extensible Mark-up Language
XSD	XML Schema Definition
YAML	YAML Ain't Markup Language

9 REFERENCES

[1] S-100 Universal Hydrographic Data Model Version 2.0.0

[2] Maritime Resource Name (mrnregistry.org)

[3] Maritime Service Portfolio (MSP) (IMO Strategic Implementation Plan on e-Navigation)

ANNEX A SERVICE SPECIFICATION SCHEMA

ServiceBaseTypesSchema.xsd

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ServiceSpecificationSchema="http://efficiensea2.org/maritime-cloud/service-
registry/v1/ServiceSpecificationSchema.xsd" targetNamespace="http://efficiensea2.org/maritime-cloud/service-
registry/v1/ServiceSpecificationSchema.xsd" elementFormDefault="qualified" version="1.0.2" xml:lang="EN">
  <annotation>
    <documentation>
                Authors:
                    EfficienSea2 WP 3 Partners
                         Thomas Lutz
                         Christoph Rihacek
                         Josef Jahn
                         Hubert Künig
    </documentation>
  </annotation>
  <annotation>
    <documentation>
                This file contains basic data type definitions for service specification, design and instance
                descriptions.
    </documentation>
  </annotation>

  <complexType name="AuthorInfo">
    <annotation>
      <documentation>
                Describes an author of a service specification or requirement.

                Elements of an authorInfo are:

                - id          Unique identifier of the author.

                - name        Human readable name of the author.

                - description Human readable description of the author.

                - contactInfo Human readable contact information of the author.

                - organizationId Unique identifier of the organization, the author belongs to.

      </documentation>
    </annotation>
    <sequence>
      <element name="id" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="name" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="description" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="contactInfo" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="organizationId" type="string" minOccurs="0" maxOccurs="1"/>
    </sequence>
  </complexType>

  <complexType name="VendorInfo">
    <annotation>
      <documentation>
                Describes the vendor producing and/or providing the service instance.
                Contains the same information as the AuthorInfo plus the isCommercial flag.

                - isCommercial  Optional indication on the commercial status of the vendor.
      </documentation>
    </annotation>
    <complexContent>
      <extension base="ServiceSpecificationSchema:AuthorInfo">
        <sequence>
          <element name="isCommercial" type="boolean" minOccurs="0" maxOccurs="1"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <simpleType name="ServiceIdentifier">
    <annotation>
      <documentation>
         Service identifier type to be used by service specifications, designs, instances.
         Currently, the identifier is defined as a string.
      </documentation>
    </annotation>
    <restriction base="string"/>
  </simpleType>

  <simpleType name="ServiceVersion">
    <annotation>
      <documentation>
        Service version indicator type to be used by service specifications, designs, instances.
```

```
        Currently, the version indicator is defined as a string.
      </documentation>
    </annotation>
    <restriction base="string"/>
  </simpleType>

  <simpleType name="ServiceStatus" final="restriction">
    <annotation>
      <documentation>
        Service status may be one of the values listed below. Service specifications,
        service designs and service instances each have their own status value.
            provisional     the service specification/design is not officially released, the service instance is
                            available, but not in official operation
            released        the service specification/design/instance is officially released / in operation
            deprecated      the service specification/design/instance is still available, but end of life is already
                            envisaged.
            deleted         the service specification/design/instance is not available any more.
      </documentation>
    </annotation>
    <restriction base="string">
      <enumeration value="provisional"/>
      <enumeration value="released"/>
      <enumeration value="deprecated"/>
      <enumeration value="deleted"/>
    </restriction>
  </simpleType>
</schema>
```

ServiceSpecificationSchema.xsd

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ServiceSpecificationSchema="http://efficiensea2.org/maritime-cloud/service-
registry/v1/ServiceSpecificationSchema.xsd" targetNamespace="http://efficiensea2.org/maritime-cloud/service-
registry/v1/ServiceSpecificationSchema.xsd" elementFormDefault="qualified" version="1.0.2" xml:lang="EN">
  <include schemaLocation="ServiceBaseTypesSchema.xsd"/>
  <annotation>
    <documentation>
                Authors:
                    EfficienSea2 WP 3 Partners
                        Thomas Lutz
                        Christoph Rihacek
                        Josef Jahn
                        Hubert Künig
    </documentation>
  </annotation>
  <annotation>
    <documentation>
                This formal description of a service specification is intentionally kept
                simple and plain.
                For all described objects lots of more attributes and related objects could
                be added and standardized, but in order to get as much adoption as possible
                the entry barrier should be low and therefore quite some aspects e.g. like
                when a service would be available have been left out.
                </documentation>
  </annotation>
  <element name="serviceSpecification" type="ServiceSpecificationSchema:ServiceSpecification">
    <annotation>
      <documentation>
                        The root element of a service specification.
                        Please refer to the type serviceSpecification for details.

      </documentation>
    </annotation>
  </element>
  <complexType name="ServiceSpecification">
    <annotation>
      <documentation>
        A service specification describes one dedicated service at logical level in a technology-independent
        way.
        The service specification identifies a service by its id and version. The service specification
        refers to requirements for the service, defines a service data model at logical level,
        defines the service interface(s) and provides information about the author(s).

        Elements of a service specification are:

        - name          The human readable service name. The service name should be at maximum a one
                        line brief label for the service. Newer versions of the same service
                        specification should not change the name.

        - id            Globally unique identification of the service. Newer versions of the same
                        service specification shall not change the id.

        - version       Version of the service specification. A service specification is uniquely
                        identified by its id and version. Any change in the service data model or in the
                        service interface definition requires a new version of the service specification.
```

```
        - status         Status of the service specification. One of the values 'provisional', 'released',
                         'deprecated', 'deleted'.

        - description    A human readable short description of the service. The description shall contain
                         an abstract of what a service implementing this specification would actually do.

        - keywords       A list of keywords associated to the service.

        - isSpatialExclusive Flag to indicate whether the service shall be "spatial exclusive". "Spatial
                         exclusiveness" means that at most one service instance of the same service
                         specification and providing the same technical specification is allowed to be
                         registered for any geographical area.

        - requirements   Refers to requirements specifications for the service. Business requirements,
                         functional and non-functional requirements should be listed here.
                         At least one requirement is mandatory.

        - authorInfos    Refers to administrative information about the authors of the service. At least
                         one author information is mandatory.

        - serviceInterfaces Refers to the definition of service interfaces. At least one service interface
                         shall be defined.

        - serviceDataModel  Mandatory reference to the definition of the logical service data model.
      </documentation>
    </annotation>
    <all>
      <element name="id" type="ServiceSpecificationSchema:ServiceIdentifier" minOccurs="1" maxOccurs="1"/>
      <element name="version" type="ServiceSpecificationSchema:ServiceVersion" minOccurs="1" maxOccurs="1"/>
      <element name="name" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="status" type="ServiceSpecificationSchema:ServiceStatus" minOccurs="1" maxOccurs="1"/>
      <element name="description" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="keywords" type="string" minOccurs="0" maxOccurs="1"/>
      <element name="isSpatialExclusive" type="boolean" default="false" minOccurs="0" maxOccurs="1"/>
      <element name="requirements" minOccurs="1" maxOccurs="1">
        <complexType>
          <sequence>
            <element name="requirement" type="ServiceSpecificationSchema:Requirement" minOccurs="1"
maxOccurs="unbounded"/>
          </sequence>
        </complexType>
      </element>
      <element name="authorInfos" minOccurs="1" maxOccurs="1">
        <complexType>
          <sequence>
            <element name="authorInfo" type="ServiceSpecificationSchema:AuthorInfo" minOccurs="1" maxOccurs="unbounded"/>
          </sequence>
        </complexType>
      </element>
      <element name="serviceDataModel" type="ServiceSpecificationSchema:ServiceDataModel" minOccurs="1" maxOccurs="1"/>
      <element name="serviceInterfaces" minOccurs="1" maxOccurs="1">
        <complexType>
          <sequence>
            <element name="serviceInterface" type="ServiceSpecificationSchema:ServiceInterface" minOccurs="1"
maxOccurs="unbounded"/>
          </sequence>
        </complexType>
      </element>
    </all>
  </complexType>
  <complexType name="Requirement">
    <annotation>
      <documentation>
        A requirement that the service specification fulfils.

        Elements of a requirement are:

        - id        Globally unique requirement identification

        - name      Human readable requirement name/summary. Shall not be longer than one line.

        - text      The human readable requirement text. Usually formulated in form of a "shall"-statement.

        - rationale Rationale for this requirement. Textual explanation of why this requirement exists.
                    Provides background information about the need of the service.

        - reference Optional information about where the requirement was originally stated. If the
                    requirement comes from external documents, this attribute shall refer to this source.

        - author    Optional reference(s) to administrative information about the author(s) of the requirement.

      </documentation>
    </annotation>
    <all>
      <element name="id" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="name" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="text" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="rationale" type="string" minOccurs="0" maxOccurs="1"/>
```

```
    <element name="reference" type="string" minOccurs="0" maxOccurs="1"/>
    <element name="authorInfos" minOccurs="0" maxOccurs="1">
      <complexType>
        <sequence>
          <element name="authorInfo" type="ServiceSpecificationSchema:AuthorInfo" minOccurs="1" maxOccurs="unbounded"/>
        </sequence>
      </complexType>
    </element>
  </all>
</complexType>
<complexType name="ServiceInterface">
  <annotation>
    <documentation>
      Specification of a service interface. One service can offer several interfaces,
      e.g. both a request/response interface and a publish/subscribe interface at the same time.
      Different interfaces will usually provide different service operations.

      Elements of a serviceInterface are:

      name                Human readable service interface name. The name shall be no longer than one line.

      description         Human readable description of the service interface

      dataExchangePattern Message exchange pattern can be one of
                            ONE_WAY,
                            REQUEST_RESPONSE,
                            REQUEST_CALLBACK,
                            PUBLISH_SUBSCRIBE,
                            BROADCAST.

      operations          Refers to the specification of service operations supported by the service
                            interface. At least one operation shall be defined.

      consumerInterfaces  Optional reference to an interface definition that has to be provided by the
                            service consumer in order to complement the service interface.
                            Especially if a publish/subscribe service interface is designed, it is
                            necessary to describe what the service expects to be available on the
                            subscriber side.

    </documentation>
  </annotation>
  <all>
    <element name="name" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="description" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="dataExchangePattern" minOccurs="1" maxOccurs="1">
      <simpleType>
        <restriction base="string">
          <enumeration value="ONE_WAY"/>
          <enumeration value="REQUEST_RESPONSE"/>
          <enumeration value="REQUEST_CALLBACK"/>
          <enumeration value="PUBLISH_SUBSCRIBE"/>
          <enumeration value="BROADCAST"/>
        </restriction>
      </simpleType>
    </element>
    <element name="operations" minOccurs="1" maxOccurs="1">
      <complexType>
        <sequence>
          <element name="operation" type="ServiceSpecificationSchema:Operation" minOccurs="1" maxOccurs="unbounded"/>
        </sequence>
      </complexType>
    </element>
    <element name="consumerInterface" type="ServiceSpecificationSchema:ConsumerInterface" minOccurs="0" maxOccurs="1"/>
  </all>
</complexType>
<complexType name="ConsumerInterface">
  <annotation>
    <documentation>
      Interface specification that is expected to be provided by the service consumer. For example,
      if a publish/subscribe service interface is designed, it is necessary to describe what the service
      expects to be available on the subscriber side.

      Elements of a consumerInterface are:

      - name        Human readable interface name. The name shall be no longer than one line.

      - description Human readable description of the interface.

      - operations  Refers to the specification of service operations supported by the consumer interface.
                      At least one operation shall be defined.
    </documentation>
  </annotation>
  <all>
    <element name="name" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="description" type="string" minOccurs="1" maxOccurs="1"/>
    <element name="operations" minOccurs="1" maxOccurs="1">
      <complexType>
        <sequence>
          <element name="operation" type="ServiceSpecificationSchema:Operation" minOccurs="1" maxOccurs="unbounded"/>
```

```
          </sequence>
        </complexType>
      </element>
    </all>
  </complexType>
  <complexType name="Operation">
    <annotation>
      <documentation>
        Definition of a service operation. Operations allow a service consumer to interact with
        the service. An operation describes a dedicated function of the service or the consumer.

        Elements of an operation are:

        - name            Human readable operation name. The name shall be no longer than one line.

        - description     Human readable description of the operation.

        - returnValueType Optional definition of the return value for the operation. The return value
                          could be a business object or a simple status code. The return value data type
                          has to be defined in the logical service data model.

        - parameterTypes  Definition of one or more parameters for the operation. This could be business
                          objects or simple types. Parameters have to be defined in the logical
                          service data model.
      </documentation>
    </annotation>
    <all>
      <element name="name" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="description" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="returnValueType" type="ServiceSpecificationSchema:ValueTypeDataModelMapping" minOccurs="0"
maxOccurs="1"/>
      <element name="parameterTypes" minOccurs="0" maxOccurs="1">
        <complexType>
          <sequence>
            <element name="parameterType" type="ServiceSpecificationSchema:ValueTypeDataModelMapping" minOccurs="1"
maxOccurs="unbounded"/>
          </sequence>
        </complexType>
      </element>
    </all>
  </complexType>
  <complexType name="ValueTypeDataModelMapping">
    <annotation>
      <documentation>
         Definition of a data type by providing a reference into the logical service data model.
         A value type data model mapping is used either in a service operation parameter or return value.

         Elements of a valueTypeDataModelMapping are:

         - typeReference  Reference to the logical service data model.
                          It references a type (or element, though type is preferred) in the logical
                          service model by the type's name attribute.
      </documentation>
    </annotation>
    <all>
      <element name="typeReference" type="string" minOccurs="1" maxOccurs="1"/>
    </all>
  </complexType>
  <complexType name="ServiceDataModel">
    <annotation>
      <documentation>
        The serviceDataModel is a logical model.  It is formally described in the sub-element modelDefinition to achieve
        interoperability and decouple it from implementing physical data models described in e.g. SOAP or REST. Encodings
        are formally described using sub-element encoding.
        The model can either be described in-line, or existing schemata can be imported.
        One service specification has one logical service model. Sub-element modelDefinition has to be provided.

        Elements of a serviceDataModel are:

        - definitionAsXSD  The definition of the service data model described in XSD. Deprecated - only for backwards
compatibility reasons
        - modelDefinition  The definition of the service data model described
        - encoding         The encoding of the attribute modelDefinition
      </documentation>
    </annotation>
    <all>
      <element name="definitionAsXSD" minOccurs="0" maxOccurs="1">
        <complexType>
          <sequence>
            <any namespace="http://www.w3.org/2001/XMLSchema" processContents="lax"/>
          </sequence>
        </complexType>
      </element>
     <element name="modelDefinition" minOccurs="0" maxOccurs="1">
     <element name="encoding" minOccurs="0" maxOccurs="1">
    </all>
  </complexType>
</schema>
```

ANNEX B SERVICE DESIGN DESCRIPTION SCHEMA

Note: For the ServiceBaseTypesSchema.xsd see ANNEX A.

ServiceDesignSchema.xsd

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ServiceDesignSchema="http://efficiensea2.org/maritime-cloud/service-registry/v1/ServiceDesignSchema.xsd"
xmlns:ServiceSpecificationSchema="http://efficiensea2.org/maritime-cloud/service-
registry/v1/ServiceSpecificationSchema.xsd" targetNamespace="http://efficiensea2.org/maritime-cloud/service-
registry/v1/ServiceDesignSchema.xsd" elementFormDefault="qualified" version="1.0.2" xml:lang="EN">
  <import namespace="http://efficiensea2.org/maritime-cloud/service-registry/v1/ServiceSpecificationSchema.xsd"
schemaLocation="ServiceBaseTypesSchema.xsd"/>
  <annotation>
    <documentation>
                Authors:
                    EfficienSea2 WP 3 Partners
                        Thomas Lutz
                        Christoph Rihacek
                        Josef Jahn
                        Hubert Künig
    </documentation>
  </annotation>
  <annotation>
    <documentation>
                This formal description of a service technical design is intentionally kept
                simple and plain.
                For all described objects lots of more attributes and related objects could
                be added and standardized, but in order to get as much adoption as possible,
                the entry barrier should be low, and therefore quite some aspects have been left out.
    </documentation>
  </annotation>
  <element name="serviceDesign" type="ServiceDesignSchema:ServiceDesign">
    <annotation>
      <documentation>
        The root element of a service technical design.
        Please refer to the type serviceDesign for details.
      </documentation>
    </annotation>
  </element>
  <complexType name="ServiceDesign">
    <annotation>
      <documentation>
          A service technical design description.

          Elements of a serviceDesign are:

           - name        The human readable name of the service design The name should be
                          at maximum a one line brief label. Newer versions of the same service
                         design should adopt the same name.

           - id           Globally unique identification of the service design Newer versions of
                          the same service design shall adopt the same id.

           - version     Version of the service design. A service design is uniquely identified
                           by its id and version. Any change in the service physical data model
                           or in the service specification reference requires a new version of the
                           service design.

          - status         Status of the service design. One of the values 'provisional', 'released',
                           'deprecated', 'deleted'.

           - description    A human readable short description of the service design. The
                           description shall contain an abstract of what a service implementation
                           actually does..

           - designsServiceSpecifications
                           Refers to service specification(s) that is/are realised
                           by this service design. As a minimum, one service specification shall
                           be referenced. One service design may realise several service
                           specification (either different versions of one specification, or even
                           different specifications).

           - offersTransport    Refers to transport technologies offered by the service design. At
                           least one reference shall be provided.

           - designedBy    Mandatory reference to information about the author of the service design.

           - servicePhysicalDataModel
```

```
                        Mandatory reference to the service physical data model description.
      </documentation>
    </annotation>
    <all>
      <element name="id" type="ServiceSpecificationSchema:ServiceIdentifier" minOccurs="1" maxOccurs="1"/>
      <element name="version" type="ServiceSpecificationSchema:ServiceVersion" minOccurs="1" maxOccurs="1"/>
      <element name="name" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="status" type="ServiceSpecificationSchema:ServiceStatus" minOccurs="1" maxOccurs="1"/>
      <element name="description" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="offersTransport" minOccurs="1" maxOccurs="1">
        <complexType>
          <sequence>
            <element name="offersTransport" type="ServiceDesignSchema:Transport" minOccurs="1" maxOccurs="unbounded"/>
          </sequence>
        </complexType>
      </element>
      <element name="designsServiceSpecifications" minOccurs="1" maxOccurs="1">
        <complexType>
          <sequence>
            <element name="designsServiceSpecifications" type="ServiceDesignSchema:ServiceSpecificationReference"
minOccurs="1" maxOccurs="unbounded"/>
          </sequence>
        </complexType>
      </element>
      <element name="designedBy" type="ServiceSpecificationSchema:VendorInfo" minOccurs="1" maxOccurs="1"/>
      <element name="servicePhysicalDataModel" type="ServiceDesignSchema:ServicePhysicalDataModel" minOccurs="1"
maxOccurs="1"/>
    </all>
  </complexType>
  <complexType name="ServiceSpecificationReference">
    <annotation>
      <documentation>
                A reference to the service specification that  is realised by  the service design
                Has the id and the version of the respective service specification.

                Elements of a ServiceSpecificationReference are:

                - id      Identification of the service specification realised by the service design.

                - version Version of the service specification realised by the service design.
      </documentation>
    </annotation>
    <all>
      <element name="id" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="version" type="string" minOccurs="1" maxOccurs="1"/>
    </all>
  </complexType>
  <complexType name="Transport">
    <annotation>
      <documentation>
                Definition of the transport protocol used by the service design.

                Elements of a transport are:

                - name        Human readable name.

                - description Human readable description of the transport protocol used by the service design.

                - protocol    A non-formal string representation of the transport (e.g. http/rest, http/soap,.. )
                              that provides enough information to a service consumer to be able to connect.
      </documentation>
    </annotation>
    <sequence>
      <element name="name" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="description" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="protocol" type="string" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
  <complexType name="ServicePhysicalDataModel">
    <annotation>
      <documentation>
                The ServicePhysicalDataModel describes the data model for the service design.
                The ServicePhysicalDataModel describes in detail all the data structures being
                actually exchanged when service consumers interact with a service instance that
                implements this design.

                - name        Human readable model name. The name shall be no longer than one line.

                - description Human readable description of the model.

                - model       The model can e.g. be a WSDL file, a JSON API, or the like. It is recommended to
                              wrap the model in a CDATA section, and provide enough information in the name and
                              description to make clear how to deal with the content in model.

                - modelType   The modelType should contain e.g. an abbreviation that indicates what technology is
                              used to describe the model. E.g. WSDL, JSON.
            </documentation>
    </annotation>
    <all>
```

```
      <element name="name" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="description" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="modelType" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="model" type="string" minOccurs="1" maxOccurs="1"/>
    </all>
  </complexType>
</schema>
```

ANNEX C SERVICE INSTANCE DESCRIPTION SCHEMA

Note: For the ServiceBaseTypesSchema.xsd see ANNEX A.

ServiceInstanceSchema.xsd

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ServiceInstanceSchema="http://efficiensea2.org/maritime-cloud/service-registry/v1/ServiceInstanceSchema.xsd"
xmlns:ServiceSpecificationSchema="http://efficiensea2.org/maritime-cloud/service-
registry/v1/ServiceSpecificationSchema.xsd" targetNamespace="http://efficiensea2.org/maritime-cloud/service-
registry/v1/ServiceInstanceSchema.xsd" elementFormDefault="qualified" version="1.0.2" xml:lang="EN">
  <import namespace="http://efficiensea2.org/maritime-cloud/service-registry/v1/ServiceSpecificationSchema.xsd"
schemaLocation="ServiceBaseTypesSchema.xsd"/>
  <annotation>
    <documentation>
                Authors:
                    EfficienSea2 WP 3 Partners
                        Thomas Lutz
                        Christoph Rihacek
                        Josef Jahn
                        Hubert Künig
    </documentation>
  </annotation>
  <annotation>
    <documentation>
        This formal description of a service instance is intentionally kept
        simple and plain.
        For all described objects lots of more attributes and related objects could
        be added and standardized, but in order to get as much adoption as possible
        the entry barrier should be low and therefore quite some aspects like
        e.g. when a service instance would be available have been left out.
      </documentation>
  </annotation>
  <element name="serviceInstance" type="ServiceInstanceSchema:ServiceInstance">
    <annotation>
      <documentation>
          The root element of a service instance.
          Please refer to the type serviceInstance for details.
        </documentation>
    </annotation>
  </element>
  <complexType name="ServiceInstance">
    <annotation>
      <documentation>
        A service instance description. One service implementation may be deployed
        at several places by same or different service providers; each such deployment
        represents a different service instance.

          Elements of a serviceInstance are:

                - name              The human readable name of the service instance. The service name should be
                                        at maximum a one line brief label. Newer versions of the same service
                                        specification should adopt the same name.

                - id                    Globally unique identification of the service instance. Newer versions of
                                        the same service instance shall adopt the same id.

                - version           Version of the service instance. A service specification is uniquely
                                        identified by its id and version. Any change in the service instance data model
                                        or in the service specification reference requires a new version of the
                                        service instance.

                - status      Status of the service instance. One of the values 'provisional', 'released',
                                       'deprecated', 'deleted'.

                - description     A human readable short description of the service instance. The description
                                        shall contain an abstract of what a service implementation actually does.

                - keywords       A list of keywords associated to the service.

                - URL               URL that describes where the service endpoint is located

                - MMSI              Optional Maritime Mobile Service Identity

                - IMO               Optional International Maritime Organization (IMO) number

                - serviceType       Optional field to categorize the service type. Example: "VIS"

                - requiresAuthorization
```

```
                                    Indicates whether authorization is required or not.

            - implementsServiceDesign
                                    Refers to the service design that is implemented by this service instance.
                                    Exactly one service design shall be referenced.

            - serviceLevel    Refers to the definition of the service level fulfilled by the service instance.
                                    Exactly one service level definition shall be provided.

            - coversAreas    Mandatory reference to the geographical area covered by the service instance.
                                     Defines a geographical area from which the service instance is accessible.
                                     This is a choice between a geographical area defined by coordinates or a
                                     United Nations location code (UN/LOCODE). One of the two options has to
                                     be provided. Worldwide accessibility is indicated by a "coversArea" element
                                     with a missing "geometryAsWKT" element.
            - producedBy      Optional reference to information about the producer of the service
                                                                                    implementation

            - providedBy      Mandatory reference to information about the service provider of the service
                              instance.
          </documentation>
    </annotation>
    <all>
      <element name="id" type="ServiceSpecificationSchema:ServiceIdentifier" minOccurs="1" maxOccurs="1"/>
      <element name="version" type="ServiceSpecificationSchema:ServiceVersion" minOccurs="1" maxOccurs="1"/>
      <element name="name" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="status" type="ServiceSpecificationSchema:ServiceStatus" minOccurs="1" maxOccurs="1"/>
      <element name="description" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="keywords" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="URL" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="MMSI" type="string" minOccurs="0" maxOccurs="1"/>
      <element name="IMO" type="string" minOccurs="0" maxOccurs="1"/>
      <element name="serviceType" type="string" minOccurs="0" maxOccurs="1"/>
      <element name="requiresAuthorization" type="boolean" minOccurs="1" maxOccurs="1"/>
      <element name="offersServiceLevel" type="ServiceInstanceSchema:ServiceLevel" minOccurs="1" maxOccurs="1"/>
      <element name="coversAreas" minOccurs="1" maxOccurs="1">
        <complexType>
          <choice>
            <element name="coversArea" type="ServiceInstanceSchema:CoverageArea" minOccurs="1" maxOccurs="unbounded"/>
            <element name="unLoCode" type="string" minOccurs="1" maxOccurs="1"/>
          </choice>
        </complexType>
      </element>
      <element name="implementsServiceDesign" type="ServiceInstanceSchema:ServiceDesignReference" minOccurs="1"
maxOccurs="1"/>
      <element name="producedBy" type="ServiceSpecificationSchema:VendorInfo" minOccurs="1" maxOccurs="1"/>
      <element name="providedBy" type="ServiceSpecificationSchema:VendorInfo" minOccurs="1" maxOccurs="1"/>
    </all>
  </complexType>
  <complexType name="ServiceDesignReference">
    <annotation>
      <documentation>
                A reference to the service design that is implemented by the service instance.
                Has the id and the version of the respective service design.

                Elements of a ServiceDesignReference are:

                - id      Identification of the service design implemented by the service instance.

                - version Version of the service design implemented by the service instance.
          </documentation>
    </annotation>
    <all>
      <element name="id" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="version" type="string" minOccurs="1" maxOccurs="1"/>
    </all>
  </complexType>
  <complexType name="ServiceLevel">
    <annotation>
      <documentation>
                Defines the service availability level.

                Elements of a serviceLevel are:

                - name         Human readable service level name. The name shall be no longer than one line.

                - description  Human readable description of the service level

                - availability Indicates the guaranteed availability of the service in %, (e.g. 99.9).
          </documentation>
    </annotation>
    <sequence>
      <element name="availability" type="float" minOccurs="1" maxOccurs="1"/>
      <element name="name" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="description" type="string" minOccurs="1" maxOccurs="1"/>
    </sequence>
  </complexType>
  <complexType name="CoverageArea">
    <annotation>
```

```
      <documentation>
                Defines a geographical area from which the service instance is accessible.

                Elements of a coverage area are:

                - name          Human readable name of the coverage area, e.g. a well-known name
                                like "Bermuda Triangle". The name shall be no longer than one line.

                - description   Human readable description of the coverage area.

                - geometryAsWKT A polygon described in WKT (Well Known Text) with coordinates in
                                coordinate reference system EPSG:4326,
                                e.g. POLYGON(LON1 LAT1, LON2 LAT2, LON3, LAT3, LON1 LAT1).
                                If the element is empty, the default is the whole world.
            </documentation>
    </annotation>
    <all>
      <element name="name" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="description" type="string" minOccurs="1" maxOccurs="1"/>
      <element name="geometryAsWKT" type="string" default="POLYGON(-180 -90, 180 -90, 180 90, -180 90, -180 -90"
minOccurs="0" maxOccurs="1"/>
    </all>
  </complexType>
</schema>
```

ANNEX D SERVICE SPECIFICATION TEMPLATE

Service Specification for the *xxx* Service

D 1 INTRODUCTION

The bulk of work on this document, has been made as a Deliverable for the EfficienSea2 project co-funded by the European Commission.

The *blue italic text* is meant to be replaced by those producing the technical service. The non-italic text is not necessarily meant to be replaced but maybe example text.

D 1.1. PURPOSE OF THE DOCUMENT

This template shall support the service architects in creating a description of the services (put down in writing) at a high level of abstraction, following the guidelines given in [1]. The template provides for each section descriptive instructions for the intended content. Formally, such instructions are written in blue italic font – they shall be deleted when writing the actual service specification document. In addition, some parts of this template provide suggested text fragments that may be directly re-used in the service specification document. Such proposed text fragments are given in black normal font.

The purpose of the service specification document is to write down the results of service identification and service design activities. The aim is to document the key aspects of a dedicated service at the logical level:

- *the operational and business context of the service;*
  - *requirements for the service (e.g. information exchange requirements);*
  - *involved nodes: which operational components provide/consume the service;*
  - *operational activities supported by the service;*
  - *relation of the service to other services:*
- *the service description;*
  - *service interface definitions;*
  - *service interface operations;*
  - *service payload definition;*
- *service provision and validation aspects.*

It should be noted that this service specification document describes just one dedicated service in detail at logical level. In addition, there should exist a service portfolio document, which presents all services of the maritime cloud that are available (or are planned to become available) at a higher level.

The purpose of this service specification document is to provide a holistic overview of one service and its building blocks at logical level. It may be complemented by a model based description (e.g. UML model describing the service interfaces, operations and data structures). The service specification document describes a well-defined baseline of the service and clearly identifies the service version. In this way, it supports the configuration management process.

The service specification document provides also the foundation material for the future standardisation process.

Note that the service specification is intended to be technology-agnostic. The service specification document shall not describe the details of a specific service implementation. For that purpose, a service instance description must be provided, where the realisation of the service with a dedicated technology shall be described.

This section should be replaced by a suitable description of the purpose. For instance:

The purpose of this service specification document is to provide a holistic overview of the XYZ service and its building blocks in a technology-independent way, according to the guidelines given in [1]. It describes a well-defined baseline of the service by clearly identifying the service version.

The aim is to document the key aspects of the *XYZ* service at the logical level:

- the operational and business context of the service;
  - requirements for the service (e.g. information exchange requirements);
  - involved nodes: which operational components provide/consume the service;
  - operational activities supported by the service;
  - relation of the service to other services;
- the service description;
  - service interface definitions;
  - service interface operations;
  - service payload definition;
  - service dynamic behaviour description.
- Service provision and validation aspects

D 1.2. INTENDED READERSHIP

This service specification template is intended to be read by service architects who shall produce service descriptions.

This section shall describe the intended readers. e.g.:

This service specification is intended to be read by service architects, system engineers and developers in charge of designing and developing an instance of the *XYZ* service.

Furthermore, this service specification is intended to be read by enterprise architects, service architects, information architects, system engineers and developers in pursuing architecting, design and development activities of other related services.

D 1.3. INPUTS FROM OTHER PROJECTS

This section lists previous work on the subject covered by this document.

Special emphasis shall be put on what has been reused from other (already finished) projects.

This section provides an overview of projects, which are dealing with similar topics and lists already finished ones that provided inputs to this activity.

D 2 SERVICE IDENTIFICATION

The purpose of this section is to provide a unique identification of the service and describe where the service is in terms of the engineering lifecycle.

Table 4 shall be completed.

Table 4 Service identification

Name	Service Name
ID	Unique identity, e.g. in form of an MRN (Maritime Resource Name, see [2]
Version	Version of the *XYZ* service specification
Description	Description of the *XYZ* service
Keywords	Keywords that can be used to find the service in the service catalogue and taxonomy
Architect(s)	Name of service architects and their organisation
Status	Status of the service in the engineering lifecycle – either 'Provisional', 'Released', 'Deprecated' or 'Deleted'.[4] 'Provisional' - the service necessity has been identified, and a short description is available, but the full-service specification is not yet ready. 'Released' - the full-service specification is ready. 'Deprecated' - service specification is announced to become invalid in the near future. 'Deleted' - service specification is not valid any more.

D 3 OPERATIONAL CONTEXT

This section describes the context of the service from an operational perspective.

The operational context description should be based on the description of the operational model, consisting of a structure of operational nodes and operational activities. If such an operational model exists, this section shall provide references to it. If no such operational model exists, then its main aspects shall be described in this section.

Optionally, a simple high level use case, described in layman's terms, could be provided as an introduction to this section.

The operational context shall be a description of how the service supports interaction among operational nodes. This can be achieved in two different levels of granularity:

1. *A description of how the service supports the interaction between operational nodes.*

   This basically consists of an overview about which operational nodes shall provide the service and which operational nodes will consume the service.

2. *A more detailed description that indicates what operational activities the service supports in a process model.*

Moreover, the operational context should describe any requirement the service will fulfil or adhere to. This refers to functional as well as non-functional requirements at high level (business/regulatory requirements, system requirements, user requirements). Especially, information exchange requirements are of much interest since the major objective of services is to support interaction between operational nodes.

4 If more elaborated governance rules for the service design process would become available, additional status values could be envisaged in the future: e.g. Validated, Verified.

The source material for the operational context description should ideally be provided by operational users and is normally expressed in dedicated requirements documentation. Ensure that the applicable documents are defined in the References section. If no requirements documents are available, then the basic requirements for the service shall be defined in the section D 3.1.

Architectural elements applicable for this description are:

- *Service;*
- *Nodes;*
- *Operational activities;*
- *Information exchange requirements.*

D 3.1. FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

This section lists all (functional and non-functional) requirements applicable to the service being described. A tabular list of requirements shall be added here. If external requirements documents are available, then the tables shall refer to these requirements, otherwise the requirements shall be documented here.

The service MUST be linked to at least one requirement. At least one of the following tables shall be presented in this section. The first table lists references to requirements available from external documents. Make sure you document the sources from where the requirements are coming from. The second table lists new requirements defined for the first time in this service specification document.

Table 5 lists applicable existing requirements for the *XYZ* service.

Table 5 Requirements Tracing

Requirement Id	Requirement Name	Requirement Text	References

Table 6 defines additional requirements for the *XYZ* service.

Table 6 Requirements Definition

Requirement Id	
Requirement Name	
Requirement Text	
Rationale	
Author	

Requirement Id	
Requirement Name	
Requirement Text	
Rationale	
Author	

D 3.2. OTHER CONSTRAINTS

D 3.2.1. RELEVANT INDUSTRIAL STANDARDS

List in this section the relevant industrial standards (if any) for the exchange of this type of data and or this type of service. These may include, for example, OGC, WFS, WMS, etc.

D 3.2.2. OPERATIONAL NODES

If an operational model exists in external documents, then this section just shows the Service to Nodes mapping by providing three tables, as described below.

If no external operational model exists, then the relevant operational nodes and their context shall be briefly described here before listing them in the tables of service providers and consumers.

Table 7 Operational Nodes providing the XYZ service

Operational Node	Remarks

Table 8 Operational Nodes consuming the XYZ service

Operational Node	Remarks

D 3.2.3. OPERATIONAL ACTIVITIES

Optional. If an operational model exists and provides sufficient details about operational activities, then this section shall include a mapping of the service to the relevant operational activities.

Table 9 Operational Activities supported by the XYZ service

Operational Node	Remarks

D 4 SERVICE OVERVIEW

This section aims at providing an overview of the main elements of the service. The elements in this view are all usually created by an UML modelling tool.

Architectural elements applicable for this description are:

- *Service - the element representing the service in its entirety;*
- *Service Interfaces - the mechanisms by which a service communicates. Defined by allocating service operations to either the provider or the consumer of the service;*
- *Service Operations - describe the logical operations used to access the service.*
- *Service Operations Parameter Definitions - identify data structures being exchanged via Service Operations.*

The above elements may be depicted in one or more diagrams. Which and how many diagrams are needed depends on the chosen architecture description framework and complexity of the service.

D 4.1. SERVICE INTERFACES

Describe the interfaces of the service including the selected Message Exchange Pattern (MEP) by using an UML diagram[5] that illustrates the service interfaces definitions and operations and in tabular form.

It is also recommended to describe the considerations resulting in the selection of a certain message exchange pattern.

A service interface supports one or several service operations. Depending on the message exchange pattern, service operations are either to be implemented by the service provider (e.g. in a Request/Response MEP, query operations are provided by the service provider – the service consumer uses them in order to submit query requests to the service provider), or by the service consumer (e.g. in a Publish/Subscribe MEP, publication operations are provided by the service consumer – the service provider uses them to submit publications to the service consumer). This distinction shall be clearly visualised in a service interface table (see example below): for each service interface, it shall be stated whether it is either provided or used by the Service. A service provides at least one service interface.

An example diagram and corresponding table is given below.

5 e.g. in NATO Architectural Framework (NAF), a NSOV-2 diagram could be used.

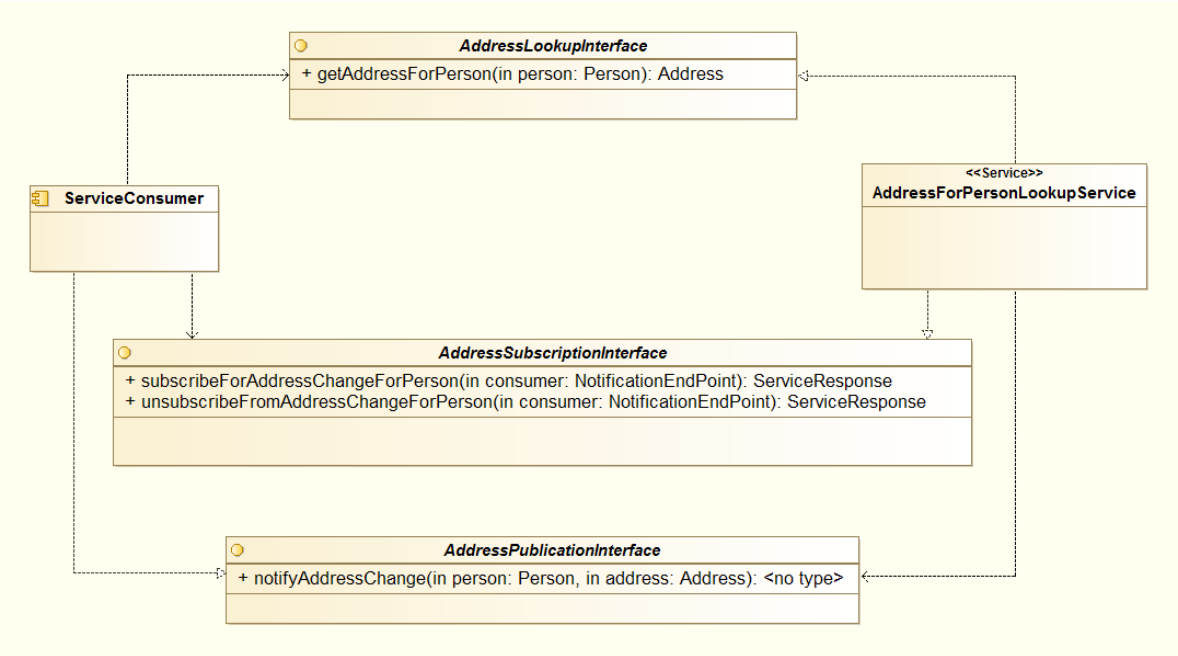


Figure 7 <Service Name> Interface Definition diagram

Table 10 Service Interfaces

ServiceInterface	Role (from service provider point of view)	ServiceOperation
AddressLookupInterface	Provided	getAddressForPerson
AddressSubscriptionInterface	Provided	subscribeForAddressChangeForPerson
		unsubscribeFromAddressChangeForPerson
AddressPublicationInterface	Required	notifyAddressChange

D 5 SERVICE DATA MODEL

This section describes the information model, i.e., the logical data structures to be exchanged between providers and consumers of the service.

It is recommended to visualise the data structures by using UML diagrams. The full information model (logical data structure) should be shown using diagram(s) and explanatory tables (see below).

Example of an UML diagram:

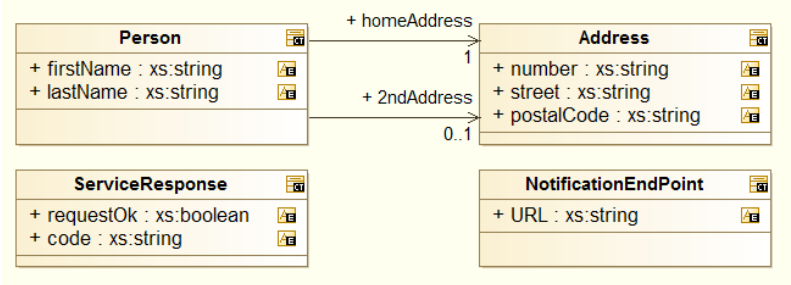


Figure 8 <Service Name> Service Data Model diagram

It is mandatory to give a description of each entity item (class), its attributes and the associations between entity items after each diagram showing data items.

If the service data model is related to an external data model (e.g. being a subset of a standard data model, e.g. based on an S-100 specification), then the service data model shall refer to it: each data item of the service data model shall be mapped to a data item defined in the external data model. This mapping may be added in the same table that describes the data items and their attributes and associations. The idea is: when reading the service specification (including the logical service data model), the payload structures must become clear to the reader. If the service re-uses structures of an external data model, then these structures can be referred to rather than replicated in the service specification. The tabular presentation of the payload allows for providing references to an externally defined model.

The table below is an example for describing a service data model including traces to an external model.

Element Name			**Description**
Person			Describe here the 'Person' structure.
	Attribute Name	**Type**	**Description**
	firstName	String	Description of firstName goes here.
	Tracing Information		**Value**
	External model trace		Trace into the logical or physical model for firstName
	Attribute Name	**Type**	**Description**
	lastName	String	Description of lastName goes here.
	Tracing Information		**Value**
	External model trace		Trace into the logical or physical model for the lastName
	Attribute Name	**Type**	**Description**
	homeAddress	Address	The main home address of Person
	Tracing Information		**Value**
	External model trace		Trace into the logical or physical model for the homeAddress
	Attribute Name	**Type**	**Description**
	2ndAddress	Address	Any second address of Person (optional)
	Tracing Information		**Value**
	External model trace		Trace into the logical or physical model for the 2ndAddress
Element Name			**Description**
Address			Describe here the Address structure.
	Attribute Name	**Type**	**Description**
	number	String	Description of number goes here.
	Tracing Information		**Value**
	External model trace		Trace into the logical or physical model for the number attribute

An XML schema for this data model is included in the formal service specification xml file attached in Appendix A.

Note: The S-100 specification [3] describes in its Appendix 9-B how S-100 based data models shall be formulated in XML schema format.

D 5.1. SERVICE INTERNAL DATA MODEL (OPTIONAL)

Optionally, this section may provide a description of the internal data model, as it seems appropriate to the service provider and/or the service consumer side. Such description might be helpful for the better understanding as it provides additional information about the building of the service. However, it must be considered just as an example – it is not an authoritative part of the service specification.

D 6 SERVICE INTERFACE SPECIFICATIONS

This section describes the details of each service interface. One sub-section is provided for each Service Interface.

The Service Interface specification covers only the static design description while the dynamic design (behaviour) is described in section D 5.

The static interface description is vital since it describes how the interfaces shall be constructed.

Architectural elements applicable for this description are:

- *Service Interfaces;*
- *Operations - function or procedures which enable programmatic communication with a Service via a Service interface;*
- *Parameters - constants or variables passed into or out of a Service interface as part of the execution of an Operation.*

A Service may have one or more Service Interfaces. Please describe each in separate sections below.

D 6.1. SERVICE INTERFACE <INTERFACE NAME>

Please explain the purpose, message exchange pattern and architecture of the Interface.

A Service Interface supports one or several service operations. Each operation in the service interface must be described in the following sections.

D 6.1.1. OPERATION <OPERATION NAME>

Give an overview of the operation: Include here a textual description of the operation functionality. In most situations this will be the same as the operation description taken from the UML modelling tool.

- OPERATION FUNCTIONALITY

Describe the functionality of the operation, i.e. how does it produce the output from the input payload.

- OPERATION PARAMETERS

Describe the logical data structure of input and output parameters of the operation (payload) by using an explanatory table (see below) and optionally UML diagrams (which are usually sub-sets of the service data model described in previous section above).

Figure 9 shows an example of a UML diagram (subset of the service data model, related to one operation).

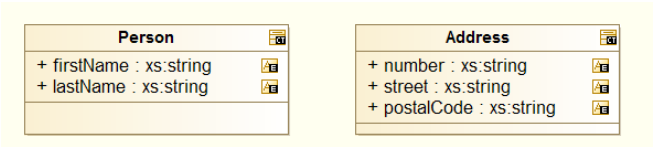


Figure 9 <Service Name> Interface Parameter Definition diagram for <operation name>

It is mandatory to provide a table with a clear description of each service operation parameter and the information about which data types defined in the service data mode are used by the service operation in its input and output parameters.

Note: While the descriptions provided in the service data model shall explain the data types in a neutral format, the descriptions provided here shall explicitly explain the purpose of the parameters for the operation.

D 6.1.2. OPERATION <OPERATION NAME>

Repeat previous section for every operation defined in the service interface definition operation.

D 6.2. SERVICE INTERFACE <INTERFACE NAME>

Repeat previous section for each interface.

D 7 SERVICE DYNAMIC BEHAVIOUR

This section describes the interactive behaviour between service interfaces (interaction specification) and, if required, between different services (orchestration). Architectural elements applicable for this description are:

- *Service Interaction Specifications;*
- *Service State machines;*
- *Service orchestration.*

Following types of views and UML diagrams can be used to describe the dynamic behaviour:[6]

- *Sequence diagrams;*
- *Interaction diagrams;*
- *State machine diagrams.*

D 7.1. SERVICE INTERFACE <INTERFACE NAME>

6 e.g. in NATO Architectural Framework (NAF), state model and interaction specification (NAF3.1) or NSOV-5 Service constraints, state model could be used.

Include some information about the dynamic aspects of the service interface; each operation should be exposed on at least one diagram.

An example sequence diagram is shown in Figure 10.

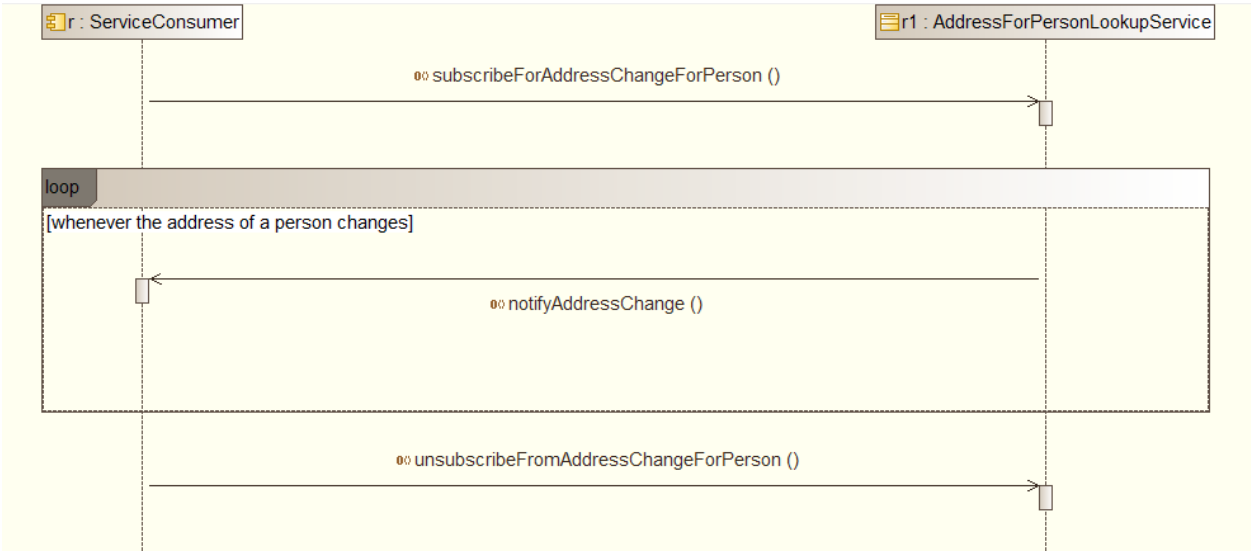


Figure 10 <Service Name> Operation Sequence Diagram

D 7.2. SERVICE INTERFACE <INTERFACE NAME>

Repeat previous section for each service interface.

D 7.3. SERVICE ORCHESTRATION (OPTIONAL)

This section should be provided, if the composition of the service and/or the relation to other services (e.g., which other services are used to provide this service; which other services are intended to use this service) is deemed relevant for the service specification.

An example sequence diagram is given below. This very simple example indicates that the AddressForPersionLookupService (i.e., the service that is being described in this Service Specification Document) acts as a consumer of a "notifyAddressChange" operation of another service, called "AddressForPersionService". Note that the other service needs to be described by its own Service Specification Document; a reference to that document should be added here).

Figure 11 <Service Name> Orchestration Sequence Diagram

D 8 SERVICE PROVISIONING (OPTIONAL)

This section should describe the way services are planned to be provided and consumed. It is labelled optional since one of the key aspects of service-orientation is to increase flexibility of the overall system by separating the

definition of services from their implementation. This means that a service can be provided in several different contexts that are not necessarily known at the time, when the service is designed.

D 9 DEFINITIONS

The definitions of terms used in this IALA Guideline can be found in the International Dictionary of Marine Aids to Navigation (IALA Dictionary) at http://www.iala-aism.org/wiki/dictionary and were checked as correct at the time of going to print. Where conflict arises, the IALA Dictionary should be considered as the authoritative source of definitions used in IALA documents.

D 9.1. TERMINOLOGY

Term	Definition
External Data Model	Describes the semantics of the 'maritime world' (or a significant part thereof) by defining data structures and their relations. This could be at logical level (e.g. in UML) or at physical level (e.g. in XSD schema definitions), as for example standard data models, or S-100 based data produce specifications.
Message Exchange Pattern	Describes the principles how two different parts of a message passing system (in our case: the service provider and the service consumer) interact and communicate with each other. Examples: In the Request/Response MEP, the service consumer sends a request to the service provider to obtain certain information; the service provider provides the requested information in a dedicated response. In the Publish/Subscribe MEP, the service consumer establishes a subscription with the service provider Service Specification XML to obtain certain information; the service provider publishes information (either in regular intervals or upon change) to all subscribed service consumers.
……	

D 10 ACRONYMS

…..

D 11 REFERENCES

This section shall include all references used when designing the service. Specifically, the applicable steering and requirements documents shall be listed.

[1] IALA Guideline 11?? on Specification of e-Navigation Technical Services

[2] http://mrnregistry.org/

[3] S-100 Universal Hydrographic Data Model, http://www.iho.int/iho_pubs/standard/S-100/S-100_Ed_2/S_100_V2.0.0_June-2015.pdf

APPENDIX 1 SERVICE SPECIFICATION XML

This Appendix contains the formal definition of the service specification.

It is up to the author whether the service specification xml file (which includes the XSD definition of the service data model) is presented in full text or just as an embedded file.

ANNEX E SERVICE TECHNICAL DESIGN TEMPLATE

Service Design Description for the *xxx* Service

<xyz Technology>

E 1 INTRODUCTION

The bulk of work on this document, has been made as a deliverable for the EfficienSea2 project co-funded by the European Commission.

The *blue italic text* is meant to be replaced by those producing the technical service. The non-italic text is not necessarily meant to be replaced but maybe example text.

E 1.1 PURPOSE OF THE DOCUMENT

This template shall support the service architects in creating a technical design description of the services (put down in writing), following the guidelines given in [1]. The template provides for each section descriptive instructions for the intended content. Formally, such instructions are written in blue italic font – they shall be deleted when writing the actual service design description document. In addition, some parts of this template provide suggested text fragments that may be directly re-used in the service design description document. Such proposed text fragments are given in black normal font.

The purpose of the service design description document is to write down the results of service technical design activity. The aim is to document how the service shall be realised by using a certain technology. The service design description document contains:

- *identification and summary of the service design;*
  - *reference to the service specification;*
  - *identification of the service design;*
- *identification and summary of chosen technology;*
- *detailed description of how to realize each service interface and service operation;*
  - *mapping of interfaced to the chosen technology;*
  - *mapping of operations to the chosen technology;*
  - *mapping of the message exchange patterns to the chosen technology;*
- *detailed description of the physical data model*
  - *mapping to the service data model of the service specification.*

***Note**: A service design description document usually describes the technical aspects of one dedicated service specification. In theory, however, it is possible to elaborate a service design that realises more than one service specification.*

The purpose of this service design description document is to provide a technology-specific description of how to realise a service specified by a service specification. The service design description document describes a well-defined baseline of the service design and clearly identifies the service design version. In this way, it supports the configuration management process.

Note that the service design description is intended to complement the technology-agnostic service specification. The purpose of the service design description document is to describe in detail the actual realisation of a service with a dedicated technology.

This section should be replaced by a suitable description of the purpose. For instance:

The purpose of this service design description document is to provide a detailed description of the *<XYZ>* service (see [2], realized by using the *<ABC>* technology, according to the guidelines given in [1]). It describes a well-defined baseline of the service design by clearly identifying the service design version.

The aim is to document the key aspects of the *<XYZ>* service technical design. This includes:

- identification and summary of the service design:
  - reference to the service specification;
  - identification of the service design;
- identification and summary of chosen technology.
- detailed description about the realization of each service interface and service operation:
  - mapping of interfaces to the chosen technology;
  - mapping of operations to the chosen technology;
  - mapping of the message exchange patterns to the chosen technology.
- detailed description of the physical data model:
  - mapping to the service data model of the service specification.

E 1.2 INTENDED READERSHIP

This service design description template is intended to be read by service architects and designers who shall produce service technical designs.

This section shall describe the intended readers of the service design description document. For instance:

This service design description document is intended to be read by service architects, designers, system engineers and developers in charge of designing and developing an instance of the *<XYZ>* service.

Furthermore, this service design description is intended to be read by service architects, information architects, system engineers and developers in pursuing architecting, design and development activities of other related services.

E 1.3 INPUTS FROM OTHER PROJECTS

This section lists previous work on the subject covered by this document.

Special emphasis shall be put on what has been reused from other (already finished) projects.

This section provides an overview of projects, which are dealing with similar topics and lists already finished ones that provided inputs to this activity.

E 2 SERVICE DESIGN IDENTIFICATION

The purpose of this section is to provide a unique identification of the service design and describe where the service is in terms of the engineering lifecycle.

The table below shall be completed.

Table 11 Service Design Identification

Name	Service Design Name
ID	Unique identity of service design
Version	Version of the XYZ service design
Technology	Indication of the technology for which this design is intended (e.g. REST or SOAP)
Service Specification ID	Reference to the service specification
Service Specification Version	Reference to the service specification

Description	Description of the XYZ service design
Keywords	Keywords that can be used to find the service design in the service registry
Architect(s)	Name of service architects/designers and their organisation
Status	Status of the service design in the engineering lifecycle – either 'Provisional', 'Released', 'Deprecated' or 'Deleted'.[7] 'Provisional': the service design is (partly) available, but not yet officially released. 'Released': the service design is ready to be used. 'Deprecated': service design is announced to become invalid in near future. 'Deleted': service design is not valid any more.

E 3 TECHNOLOGY INTRODUCTION

The technology introduction section contains a basic background about the chosen technology. In most cases this will be a short description of basic technology aspects accompanied with appropriate references to standard documents and best practice descriptions.

The template does not provide further details for the structure of this section. The actual structure is left to the author's choice.

E 4 SERVICE DESIGN OVERVIEW

This section provides an overview of the main elements of the service design and a mapping of the design elements to the service specification elements.

This section aims at providing an overview of the main elements of the service design and a mapping of the design elements to the service specification elements. The elements in this view are all usually created by a UML modelling tool.

Architectural elements applicable for this description are:

- *Service - the element representing the service in its entirety;*
- *Service Interfaces- the mechanisms by which a service communicates. Defined by allocating service operations to either the provider or the consumer of the service;*
- *Service Operations - describe the operations used to access the service;*
- *Service Operations Parameter Definitions - identify data structures being exchanged via Service Operations.*

Above elements may be depicted in one or many diagrams. Which and how many diagrams are needed depends on the chosen architecture description framework, the chosen technology, and the complexity of the service.

If the structure of the service design follows the service specification, then it is not necessary to repeat identical diagrams here in this section; in this case, this section shall contain references to the service specification document. However, it is assumed that in many cases, depending on the chosen technology, the actual interface and/or operation names (and structuring) are not 100% identical to the abstract definition given in the service specification.

E 4.1 SERVICE INTERFACES

7 If more elaborated governance rules for the service design process would become available, additional status values could be envisaged in the future: e.g. Validated, Verified.

Describe the interfaces of the service design and their mapping to the interfaces defined in the service specification. Furthermore, describe how the specified Message Exchange Patterns (MEP) are realised with the chosen technology.

An example diagram and corresponding table are shown at Figure 12 and Table 12.

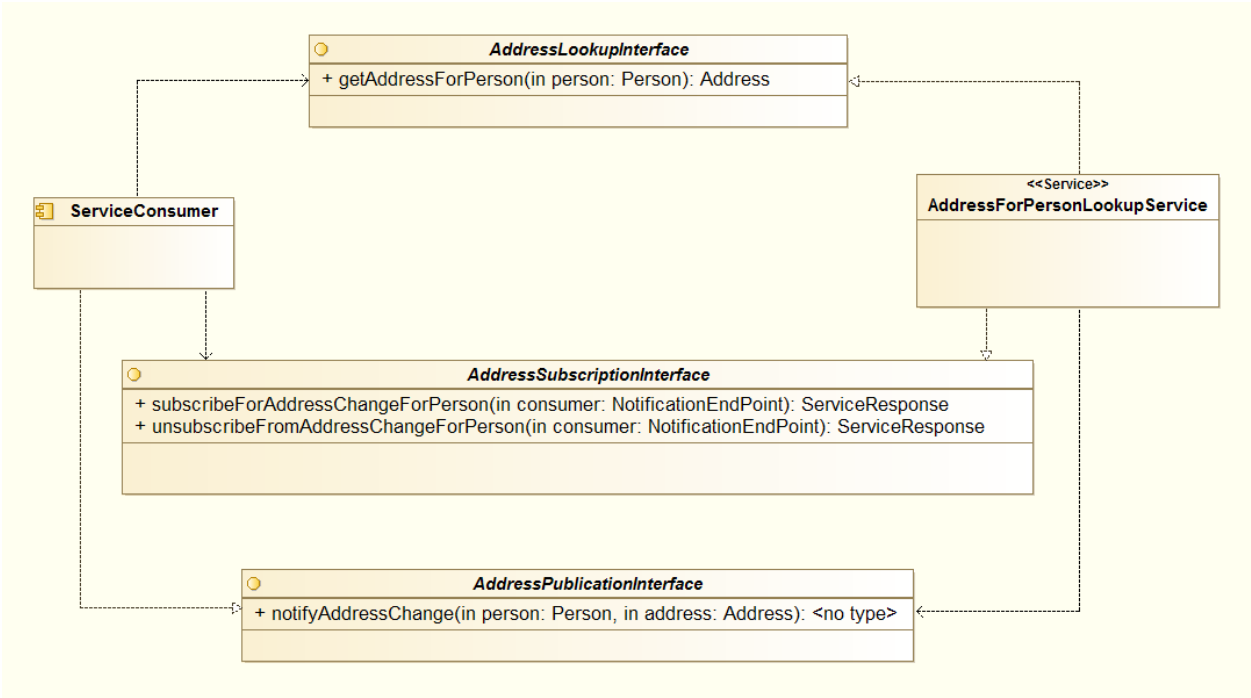


Figure 12 <Service Name> Interface Definition diagram

Table 12 Service Interface Mapping

Service Specification		Service Design	
ServiceInterface	**Service Operation**	**Service Interface**	**Service Operation**
AddressLookupInterface	getAddressForPerson	AddressLookup (see WSDL file [x])	findAddress()
subscribeForAddressChangeForPerson	subscribeForAddressChangeForPerson	WebService Notification interface specified by WSDL file [y]	Standard WS-N subscribe()
	unsubscribeFromAddressChangeForPerson		Standard WS-N unsubscribe()
AddressPublicationInterface	notifyAddressChange	WebService Notification interface specified by WSDL file [y]	Standard WS-N notify()

The table above (in this example for service design using SOAP) shall provide the mapping of service design to service specification, as well as references to the formal descriptions of the service interfaces and operations (these references are symbolised by [x], [y] in the table above). These may be references to external documents (e.g. standards) or to other sections in this document (e.g. to subsection of section 2).

E 5 PHYSICAL DATA MODEL

This section describes in detail the data structures to be exchanged between providers and consumers of the service.

This section provides a detailed description of the data structures exchanged between service provider and service consumer. This description shall also include a mapping of the data structures to the service data model provided in the service specification.

The service design description template does not prescribe a detailed format for this section. Allowed presentations of the physical data model include:

- *UML diagrams representing the data structures including detailed physical data type descriptions at attribute level;*
- *XML/XSD files describing the data structures;*
- *Tabular presentations.*

Any mixture of the above formats is allowed.

Figure 13shows an example of an UML diagram.

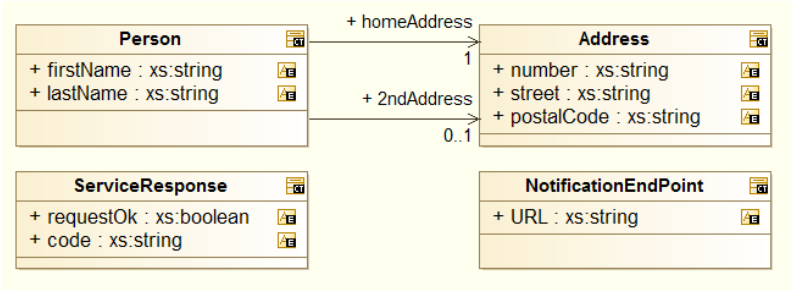


Figure 13 <Service Name> Service Data Model diagram

It is mandatory to give a description of each entity item (class), its attributes and the associations between entity items. The data type of each attribute must be provided, appropriate to the chosen technology.

If the physical service data model is related to an external data model (e.g. being a subset of a standard data model, e.g. based on an S-100 specification), then this section shall refer to it: each data item of the physical data model shall be mapped to a data item defined in the external data model. This mapping may be added in the same table that describes the data items and their attributes and associations.

Table 13 is an example for describing a service data model including traces to an external model.

Table 13 Service Data Model Description

Element Name			Description
Person			Describe here the 'Person' structure.
	Attribute Name	**Type**	**Description**
	firstName	String	Description of firstName goes here.
	Tracing Information		**Value**
	Spec.data model trace		Trace into the service specification data model for firstName
	External model trace		Trace into the external data model for firstName
	Attribute Name	**Type**	**Description**
	lastName	String	Description of lastName goes here.
	Tracing Information		**Value**
	Spec.data model trace		Trace into the service specification data model for lastName
	External model trace		Trace into the logical or physical model for the lastName
	Attribute Name	**Type**	**Description**
	homeAddress	Address	The main home address of Person
	Tracing Information		**Value**
	Spec. data model trace		Trace into the service specification data model for homeAdd
	External model trace		Trace into the logical or physical model for the homeAddress
	Attribute Name	**Type**	**Description**
	2ndAddress	Address	Any second address of Person (optional)
	Tracing Information		**Value**
	External model trace		Trace into the logical or physical model for the 2ndAddress
Element Name			**Description**
Address			Describe here the Address structure.
	Attribute Name	**Type**	**Description**
	number	String	Description of number goes here.
	Tracing Information		**Value**
	Spec.data model trace		Trace into the service specification data model for the number attribute
	External model trace		Trace into the logical or physical model for the number attribute

An XML schema for this data model is included in the formal service design xml file attached in APPENDIX 1.

E 5.1 SERVICE INTERNAL DATA MODEL (OPTIONAL)

Optionally, this section may provide a description of the internal data model, as it seems appropriate to the service provider and/or the service consumer side. Such description might be helpful for the understanding as it provides additional information of how the service might be built. However, it must be seen as exemplary only – it is not an authoritative part of the service design description.

E 6 SERVICE INTERFACE DESIGN

This section describes the details of each service interface. One sub-section is provided for each Service Interface.

The Service Interface design covers the static design description while the dynamic design (behaviour) is described in section E 7.

The static interface description is vital since it describes how the interfaces shall be constructed. The structure of this section is identical to the structure of the Service Interface Specifications section in the service specification document. This section may be limited to references to the service specification document, if all the following conditions are fulfilled:

- *the service design reflects the service interfaces in a 1:1 manner;*
- *the service interfaces are sufficiently described in the service specification;*
- *the physical data model (section E 5) contains an unambiguous mapping of all payload data items of the service specification to the detailed physical data items.*

Architectural elements applicable for this description are:

- *service Interfaces;*
- *operations - function or procedures which enable programmatic communication with a service via a service interface;*
- *parameters - constants or variables passed into or out of a service interface as part of the execution of an operation.*

A Service may have one or more service Interfaces. Please describe each in separate sections below.

E 6.1 SERVICE INTERFACE <INTERFACE NAME>

Please explain the purpose, messaging pattern and architecture of the Interface.

A Service Interface supports one or several service operations. Each operation in the service interface shall be described in the following sections.

E 6.1.1 OPERATION <OPERATION NAME>

Give an overview of the operation: Include here a textual description of the operation functionality. In most instances this will be the same as the operation description taken from the UML modelling tool.

- Operation Functionality

Describe here the functionality of the operation, i.e. how does it produce the output from the input payload.

- Operation Parameters

Describe the logical data structure of input and output parameters of the operation (payload) by using an explanatory table (see below) and optionally UML diagrams (which are usually sub-sets of the service data model described in previous section above).

Figure 14 shows an example of a UML diagram (subset of the service data model, related to one operation).

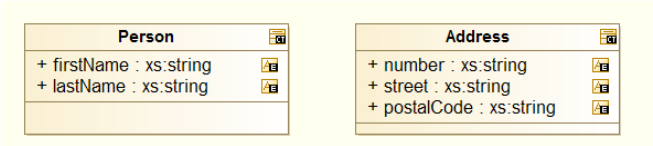


Figure 14 <Service name> Interface Parameter Definition diagram for <operation name>

It is mandatory to provide a table with a clear description of each service operation parameter and the information about which data types defined in the service data mode are used by the service operation in its input and output parameters.

***Note:** While the descriptions provided in the physical data model shall explain the data types in a neutral format, the descriptions provided here shall explicitly explain the purpose of the parameters for the operation.*

Table 14 shows an example operation parameter description table.

Table 14 Payload description of <operation name> operation

Parameter Name	Direction	Data Type	Description
person	Input	Person	The 'person' parameter specifies the person for which the address is being looked for.
<none>	Return	Address	The return value provides the address of the person.

E 6.1.2 OPERATION <OPERATION NAME>

Repeat previous section for every operation defined in the service interface definition operation.

E 6.2 SERVICE INTERFACE <INTERFACE NAME>

Repeat previous section for each interface.

E 7 SERVICE DYNAMIC BEHAVIOUR

This section describes the interactive behaviour between service interfaces (interaction specification) and, if required, between different services (orchestration). Architectural elements applicable for this description are:

- *Service Interaction Specifications;*
- *Service State machines;*
- *Service orchestration.*

Following types of views and UML diagrams can be used to describe the dynamic behaviour:[8]

- *Sequence diagrams;*
- *Interaction diagrams;*
- *State machine diagrams.*

This section is especially relevant, if the service design structure (see section E 4) differs from the service structure introduced in the service specification. If designed service interfaces and operations are equivalent to those of the

8 *e.g. in NATO Architectural Framework (NAF), state model and interaction specification (NAF3.1) or NSOV-5 Service constraints, state model could be used.*

service specification, and if the dynamic behaviour is sufficiently described in the service specification, then this section may be limited to references to the service specification document.

E 7.1 SERVICE INTERFACE <INTERFACE NAME>

Include some information about the dynamic aspects of the service interface; each operation should be exposed on at least one diagram.

An example sequence diagram is given below.

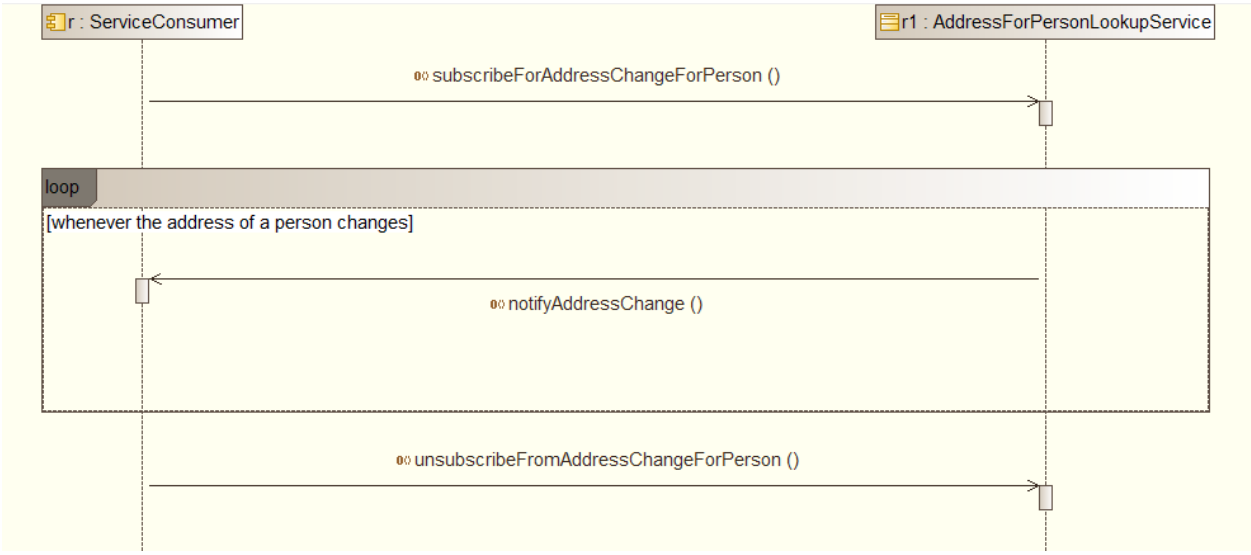


Figure 15 <Service Name> Operation Sequence Diagram

E 7.2 SERVICE INTERFACE <INTERFACE NAME>

Replicate previous section for each service interface.

E 8 DEFINITIONS

The definitions of terms used in this IALA Guideline can be found in the International Dictionary of Marine Aids to Navigation (IALA Dictionary) at http://www.iala-aism.org/wiki/dictionary and were checked as correct at the time of going to print. Where conflict arises, the IALA Dictionary should be considered as the authoritative source of definitions used in IALA documents.

E 8.1 TERMINOLOGY

Term	Definition
External Data Model	Describes the semantics of the 'maritime world' (or a significant part thereof) by defining data structures and their relations. This could be at logical level (e.g. in UML) or at physical level (e.g. in XSD schema definitions), as for example standard data models, or S-100 based data produce specifications.
Message Exchange Pattern	Describes the principles two different parts of a message passing system (in our case: the service provider and the service consumer) interact and communicate with each other. Examples: In the Request/Response MEP, the service consumer sends a request to the service provider to obtain certain information; the service provider provides the requested information in a dedicated response.

Term	Definition
	In the Publish/Subscribe MEP, the service consumer establishes a subscription with the service provider to obtain certain information; the service provider publishes information (either in regular intervals or upon change) to all subscribed service consumers.
……	

E 9 ACRONYMS

….

E 10 REFERENCES

This section shall include all references used when designing the service. Specifically, the service specification document as well as standard documents describing the chosen technology and documents describing any external data models (if applicable) shall be listed.

[1] IALA Guide on specification of e-navigation Technical Services

[2] *XYZ Service Specification for the XYZ service*

APPENDIX 1 SERVICE DESIGN DESCRIPTION XML

This appendix contains the formal definition of the service design description.

It is up to the author whether the service design description xml file (which includes the technology dependent definition of the physical data model) is presented in full text or just as an embedded file.

ANNEX F SERVICE INSTANCE DESCRIPTION

Service Instance Description for the *xxx* Service

<*Service Instance Name*>

F 1 INTRODUCTION

The bulk of work on this document, has been made as a deliverable for the EfficienSea2 project co-funded by the European Commission.

The *blue italic text* is meant to be replaced by those producing the technical service. The non-italic text is not necessarily meant to be replaced but maybe example text.

F 1.1 PURPOSE OF THE DOCUMENT

This template shall support the software architects and implementers in creating a description of the service implementation and instantiation (put down in writing), following the guidelines given in [1]. The template provides for each section descriptive instructions for the intended content. Formally, such instructions are written in blue italic font – they shall be deleted when writing the actual service instance description document. In addition, some parts of this template provide suggested text fragments that may be directly re-used in the service instance description document. Such proposed text fragments are given in black normal font.

The purpose of the service instance description document is to provide a detailed description of how a service is realised in software and hardware. In most cases, this document will be rather short, since it is expected that the implementation follows the technical design and it is not supposed to replicate any information from the service design description document. The service instance description document contains:

- *identification and summary of the service instance:*
  - *reference to the service design description;*
  - *reference to the service specification;*
  - *identification of the service instance.*
- *service implementation and instantiation details:*
  - *internal design decisions;*
  - *configuration data;*
  - *deployment information.*
- *release notes:*
  - *feature list;*
  - *bug list.*

This section should be replaced by a suitable description of the purpose. For example:

The purpose of this service instance description document is to provide a documentation of the implementation and instantiation of the *<XYZ>* service (see [2]), realized by using the *<ABC>* technology as described in [3], according to the guidelines given in [1]. It describes a well-defined baseline of the service implementation by clearly identifying the service implementation version.

The aim is to document the key aspects of the *<XYZ>* service instantiation. This includes:

- identification and summary of the service instance:
  - reference to the service design description;
  - reference to the service specification;
  - identification of the service instance.
- service implementation and instantiation details:
  - internal design decisions;

- configuration data;
- deployment information.

- release notes:
  - feature list:
  - bug list.

F 1.2 INTENDED READERSHIP

This service instance description template is intended to be read by software architects, designers and implementers who shall produce service implementation and instance description.

This section shall describe the intended readers of the service instance description document. For example:

This service instance description document is intended to be read by service providers, system engineers and developers in charge of deploying and operating an instance of the *<XYZ>* service

F 2 SERVICE INSTANCE IDENTIFICATION

Table 15 Service Instance Identification

Name	Service instance name.
ID	Unique identity of service instance.
Version	Version of the XYZ service instance.
Technology	Indication of the technology used and supported by this instance (e.g. REST or SOAP).
Service Specification ID	Reference to the service specification.
Service Specification Version	Reference to the service specification.
Service Design ID	Reference to the service design.
Service Design Version	Reference to the service design.
Description	Short description of the XYZ service instance. The description shall contain an abstract of what a service implementation does and what the service consumer should know about how the service implementation works in this instance.
Keywords	Keywords that can be used to find the service instance in the service registry.
Supplier	Identification of organisation supplying this service implementation/instance.

Status	Status of the service implementation/instance in the engineering lifecycle – either 'Provisional', 'Released', 'Deprecated' or 'Deleted'. 'Provisional': the service instance is (partly) available, but not yet officially released. 'Released': the full-service instance is ready. 'Deprecated': service instance is announced to become invalid in near future. 'Deleted': service instance is not valid any more.

F 3 SERVICE IMPLEMENTATION AND INSTANTIATION DETAILS

This section describes any information that appears useful for the understanding of the service implementation in general and of the actual service instance in particular. This may include internal design decisions, required configuration data, deployment pre-requisites, etc.

The template does not provide further details for the structure of this section. The actual structure is left to the author's choice.

F 4 RELEASE NOTES

This section describes the release notes of the service instance. It shall contain at least the following set of information:

- *release identification and date;*
- *feature list;*
  - *added features;*
  - *changed features;*
  - *removed features;*
- *bug list;*
  - *known open bugs;*
  - *resolved bugs.*

The template does not provide further details for the structure of this section. The actual structure is left to the author's choice.

F 5 DEFINITIONS

The definitions of terms used in this IALA Guideline can be found in the International Dictionary of Marine Aids to Navigation (IALA Dictionary) at http://www.iala-aism.org/wiki/dictionary and were checked as correct at the time of going to print. Where conflict arises, the IALA Dictionary should be considered as the authoritative source of definitions used in IALA documents.

F 5.1 TERMINOLOGY

Term	Definition
External Data Model	Describes the semantics of the 'maritime world' (or a significant part thereof) by defining data structures and their relations. This could be at logical level (e.g. in UML) or at physical level (e.g. in XSD schema definitions), as for example standard data models, or S-100 based data produce specifications.
Message Exchange Pattern	Describes the principles two different parts of a message passing system (in our case: the service provider and the service consumer) interact and communicate with each other. Examples: In the Request/Response MEP, the service consumer sends a request to the service provider to obtain certain information; the service provider provides the requested information in a dedicated response. In the Publish/Subscribe MEP, the service consumer establishes a subscription with the service provider to obtain certain information; the service provider publishes information (either in regular intervals or upon change) to all subscribed service consumers.
……	

F 6 ACRONYMS

….

F 7 REFERENCES

This section shall include all references used in the service instance description. Specifically, the service specification document as well as the applicable service design description shall be listed.

[1] IALA Guideline 11?? on Specification of e-Navigation Technical Services

[2] *ABC Service Specification for the ABC service*

[3] *XYZ Service Design Description for the XYZ service*

ANNEX 2: NW-NM SERVICE SPECIFICATION

NW-NM Service Specification v0.4

Contents

1 Introduction

1.1 NW-NM

Navigational Warnings (NW) and temporary / preliminary Notice to Mariners (NM T&P) have many similarities and few differences. They largely serve the same purpose, with the main differences being down to the current speed, quality assurance and methods of promulgation.

Navigational Warnings (NW) are part of the Maritime Safety Information (MSI) system. Currently, NW's are promulgated in text via SafetyNET, NAVTEX, and is in some countries accessible on the WWW or as voice broadcasts via coastal radio stations.

Notices to Mariners (NM) are promulgated weekly in order to keep nautical charts and publications, as far as possible, up to date. Temporary (T) and Preliminary (P) NMs advise mariners of important matters affecting navigational safety, including new hydrographic information (in advance of new editions or chart updates), changes to routing measures and aids to navigation, and other important categories of data. NM T&P's are today promulgated on paper weekly, fortnightly or monthly and are often accessible on the WWW in PDF format. Not all ENCs include T&P information currently.

As part of EfficienSea 2, a combined NW-NM model, and promulgation thereof, is being developed and tested. This involves the specification of an NW-NM service and implementation of an NW-NM service instance that can be integrated with the Maritime Cloud eco system.

The requirements for the NW-NM service is further detailed in the the project document "NW-NM service description and requirements gathering" [2].

1.2 Purpose of the document

The purpose of this service specification document is to provide a holistic overview of the MW-NM service and its building blocks in a technology-independent way, according to the guidelines given in the Service Description Guidelines [1].

1.3 Intended readership

This service specification is intended to be read by service architects, system engineers and developers in charge of designing and developing an instance of the NW-NM service.

Furthermore, this service specification is intended to be read by enterprise architects, service architects, information architects, system engineers and developers in pursuing architecting, design and development activities of other related services.

1.4 Inputs from other projects

An approach to NW-handling and promulgation via AIS was tested during the initial EfficienSea project. The EPD was used to test and evaluate portrayal of NW's on a ECDIS-like device.

This was further developed in the ACCSEAS project, which also developed a combined MSI-NM model and interchange format, see [4], plus an authoring system and promulgation via the Maritime Cloud Messaging Service (please refer to www.maritimecloud.net).

The MSI-NM interchange format devised in the ACCSEAS project was furthermore used as input for IHO, targeting he S-124 NW specification - see [5].

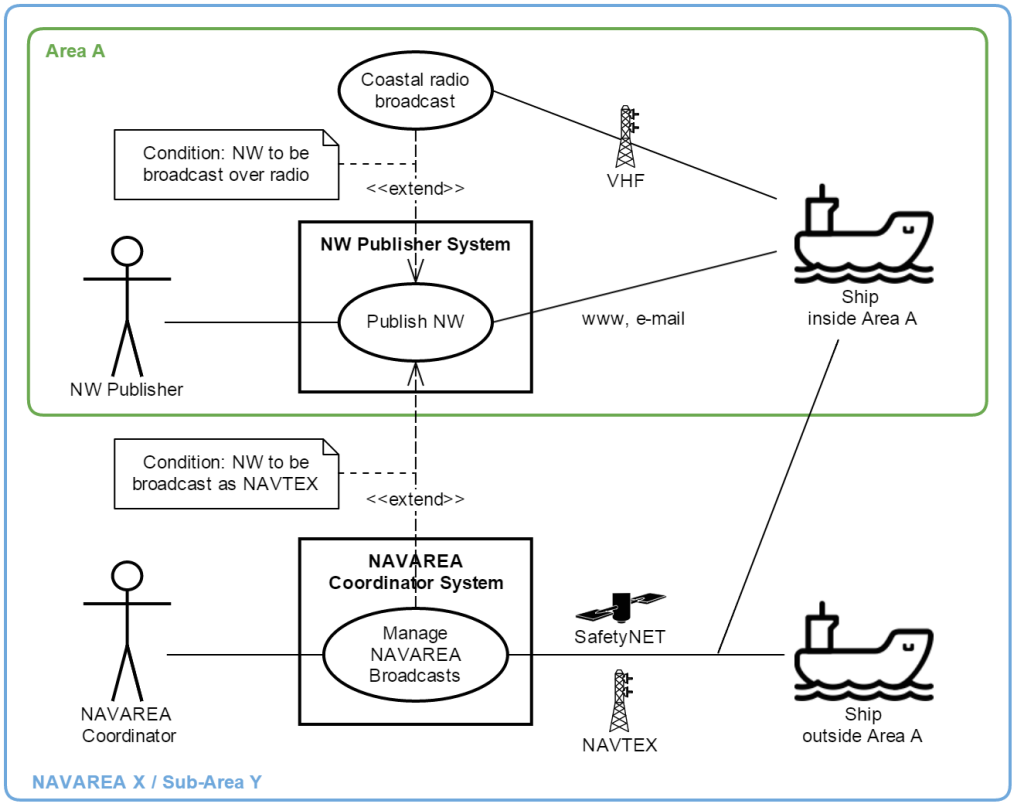
2 Service Identification

Name	NW-NM T&P Maritime Cloud Service
ID	urn:mrn:mcl:service:specification:dma:nw-nm
Version	0.4
Description	The NW-NM service specification defines a combined NW-NM T&P model along with the actual service API used for accessing NW-NM data, as registered in the Maritime Cloud service catalogue.
Keywords	NW, NM, Navigational Warnings, Notices to Mariners, MSI, Maritime Cloud Service.
Architect(s)	e-Navigation Team Danish Maritime Authority Carl Jacobsens Vej 31 DK-2500 København K Telephone: +45 40 72 61 08 Email: mfs@dma.dk
Status	Released.

3 Operational Context

3.1 Present Day Operational Context

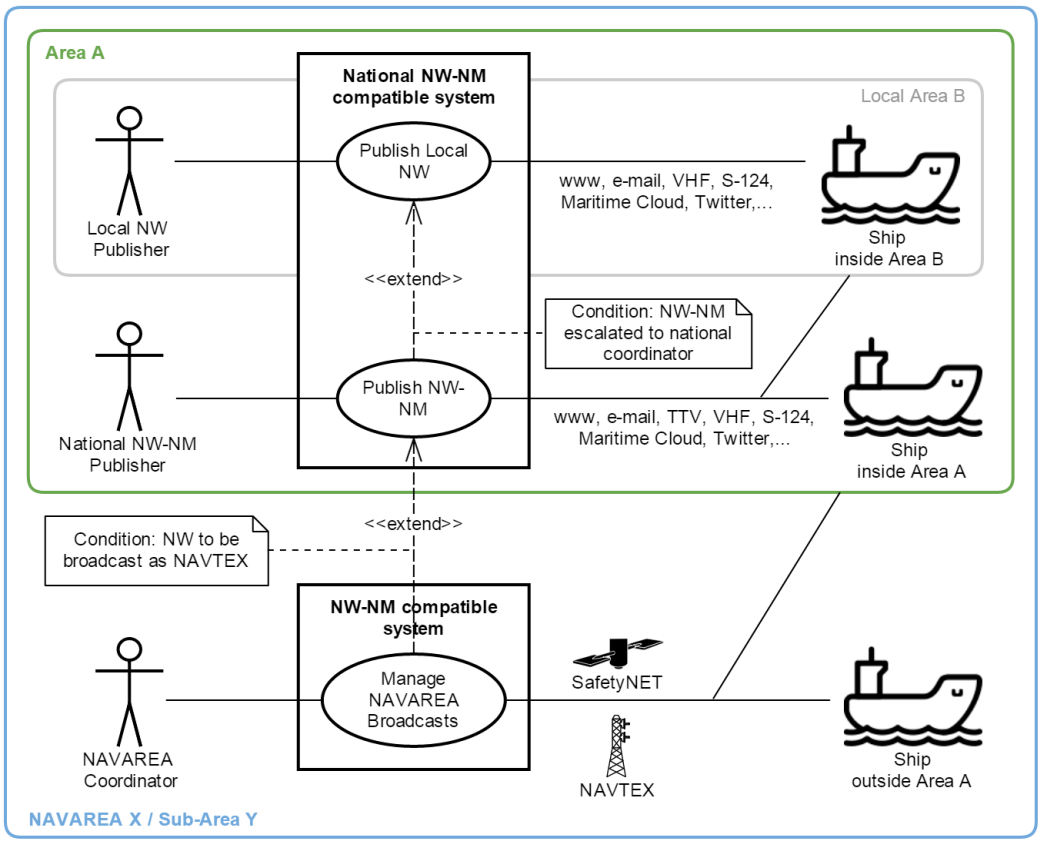
Todays NW broadcast regime, i.e. the operational context of NW promulgation at the component level, is depicted below:



Please note, NM T&P provision is currently via ordinary mail, or downloadable as PDF via a webpage of the NM provider.

System interfaces between NW publishers, NAVAREA (or Sub-Area) coordinator and broadcast service are not standardized, and may rely on manual processes involving e-mail, telephone, voice radio transmissions, fax, telex and manual re-entering of information from one system to another, or much more advanced solutions. Provision of NW or NM via web is not standardized. NAVTEX and SafetyNET cannot transport structured data formats for a joint NW-NM solution.

3.2 EfficienSea 2 Envisaged Operational Context



This Scenario depicts an envisaged future NW & NM T&P promulgation regime, as explored in EfficienSea 2.
Based on a standardized structured NW-NM format, compatible NW-NM systems will be able to exchange NW and NM T&P seamlessly. As depicted, each country may have a national NW-NM system, used by local authorities (e.g. harbor and port authorities) as well as the national authorities and agencies (e.g. national maritime safety agencies or hydrographic offices).
Local authorities will administer and publish local NW for their area of responsibility, whereas the national authorities will cater for NW and NM T&P on the national level. Local authorities should have the ability to escalate NW to the national coordinator.
The NW-NM received by ships will thus depend on the promulgation method of choice. If, say, a ship targets the website of a specific port authority; it may see the local NW published by this authority. If, however, the ship query for NW-NM via the Maritime Cloud, it will receive NW-NM from national and local authorities relevant to its current position and planned routes.
The NW-NM Service detailed in this specification only caters for a small part of this promulgation regime. It exposes a single service operation to fetch all currently published (*in force*) NW and NM message from the targeted authority. It may be used by any client, such as a ship, a website or an app.

3.3 Functional and Non-functional Requirements

The table below defines additional requirements for the NW-NM service.

Table 1: Requirements Definition

Requirement Id	urn:mrn:mcl:requirement:nw-nm:1

Requirement Name	Combined NW-NM model
Requirement Text	The data model should encapsulate a combined NW-NM model.
Rationale	Navigational Warnings (NW) and temporary / preliminary Notice to Mariners (NM T&P) have many similarities and few differences. They largely serve the same purpose, with the main differences being down to the current speed, quality assurance and methods of promulgation

Requirement Id	**urn:mrn:mcl:requirement:nw-nm:2**
Requirement Name	Return all published NW-NM messages.
Requirement Text	The NW-NM service should make it possible to retrieve all published NW-NM messages from the given service provider.

3.4 Other Constraints

3.4.1 Relevant Industrial Standards

The NW-NM model needs to cater for the IHO-IMO-WMO S-53 standard [S-53] on MSI (including NW) and the IHO S-4 standard [S-4] which covers NM T&P.

3.4.2 Operational Nodes

Table 2: Operational Nodes providing the NW-NM service

Operational Node	Remarks
National NW-NM Publisher	Typically, the national maritime safety agencies or hydrographic offices will produce the NW and NM messages and publish via various channels, including the NW-NM Service. Additionally, local authorities (e.g. harbor authorities) may publish local NW's via the national system.
NAVAREA (and Sub-AREA) publishers.	NAVAREA (or Sub-Area) coordinators will receive eligible NW messages from the national agencies and publish these in their area of responsibility.

Table 3: Operational Nodes consuming the NW-NM service

Operational Node	Remarks
Ships, websites and apps	All sorts of clients can be envisaged to consume the published NW-NM messages. Examples may be an ECDIS on a ship, or a safety related sailing App.

3.4.3 Operational Activities

Table 4: Operational Activities supported by the NW-NM service

Operational Activity	Remarks
Showing published NW-NM messages on an ECDIS.	An ECDIS may first look for NW-NM service instances for a relevant area in the Maritime Cloud Service Registry. If one or more NW-NM services instances have been resolved, it will call the operation to retrieve all published NW-NM messages, and display these on an ECDIS.

4 Service Overview

4.1 Service Interfaces

The NW-NM service consists of a single service, exposing a single operation to query the currently published NW-NM messages (Request/Reply Message Exchange Pattern).

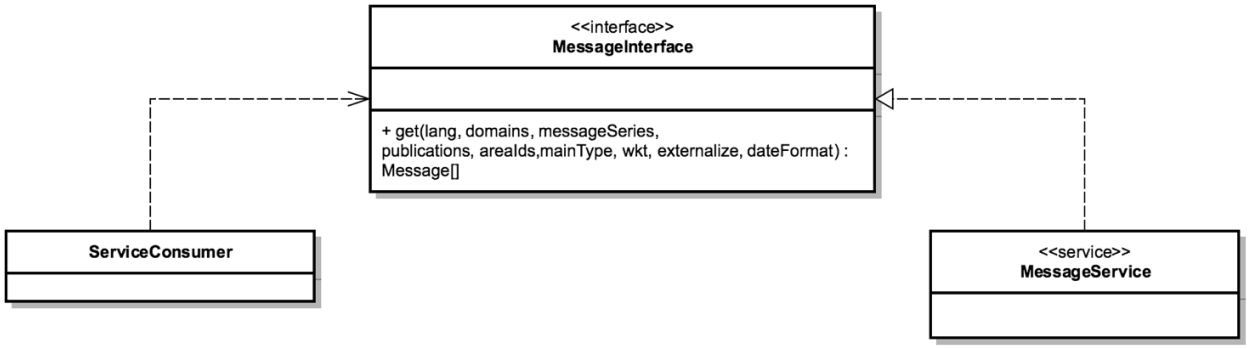


Figure 1: NW-NM Service Definition Diagram

ServiceInterface	Role (from service provider point of view)	ServiceOperation
MessageInterface	Provided	get(lang, domains, messageSeries, publications, areaIds, mainType, wkt, externalize, dateFormat)

The returned data model is detailed in chapter 5 and the service operation in chapter 6.

5 Service Data Model

This section describes the logical data structures of the NW-NM service. The combined NW-NM model needs to cater for the IHO-IMO-WMO S-53 standard on MSI (including NW) and the IHO S-4 standard which covers NM T&P.

The overarching idea has been to generalize the constituent parts and fields of NW and NM T&P messages, and make the format both backwards compatible and future-proof by e.g. adding support for:

- Multi-language support. All messages must be localizable to any number of languages, including the base data they reference (e.g. areas). The pattern adopted to support this, is to let all classes with localizable attributes (such as *Message*) have an associated list of description entities (*MessageDesc*) which contains a language code and the localizable fields. The description entities are yellow in the UML diagram below.
- Rich text support. NMs in particular, can contain a rich layout containing features such as tables, links, embedded pictograms, etc. By supporting HTML descriptions this can be accommodated.
- New identifier format. The S-4 and S-53 standards loosely specifies a numbering scheme for NWs and NMs. However, the numbering scheme does not guarantee uniqueness in a combined NW-NM model, let alone a system that may contain messages from multiple authorities. Thus, the NW-NM data model introduces *message series* and adds a unique MRN (maritime resource name) to each message.
- Base data. Part of a combined NW-NM model is to define a relationship between messages and base data such as charts, categories and areas. Previous proposals have opted for rigid solutions with a fixed number of area and category levels, and with enumerated category values.
- S-124 compatibility. The IHO S-124 specification for Navigational Warnings has not been released yet, but an aim of the NW-NM model has been to be relatively future-compatible with the S-124 data format, to make it easy to exchange data between the two formats.

The UML detailing the Message class, and its related classes, is given below:

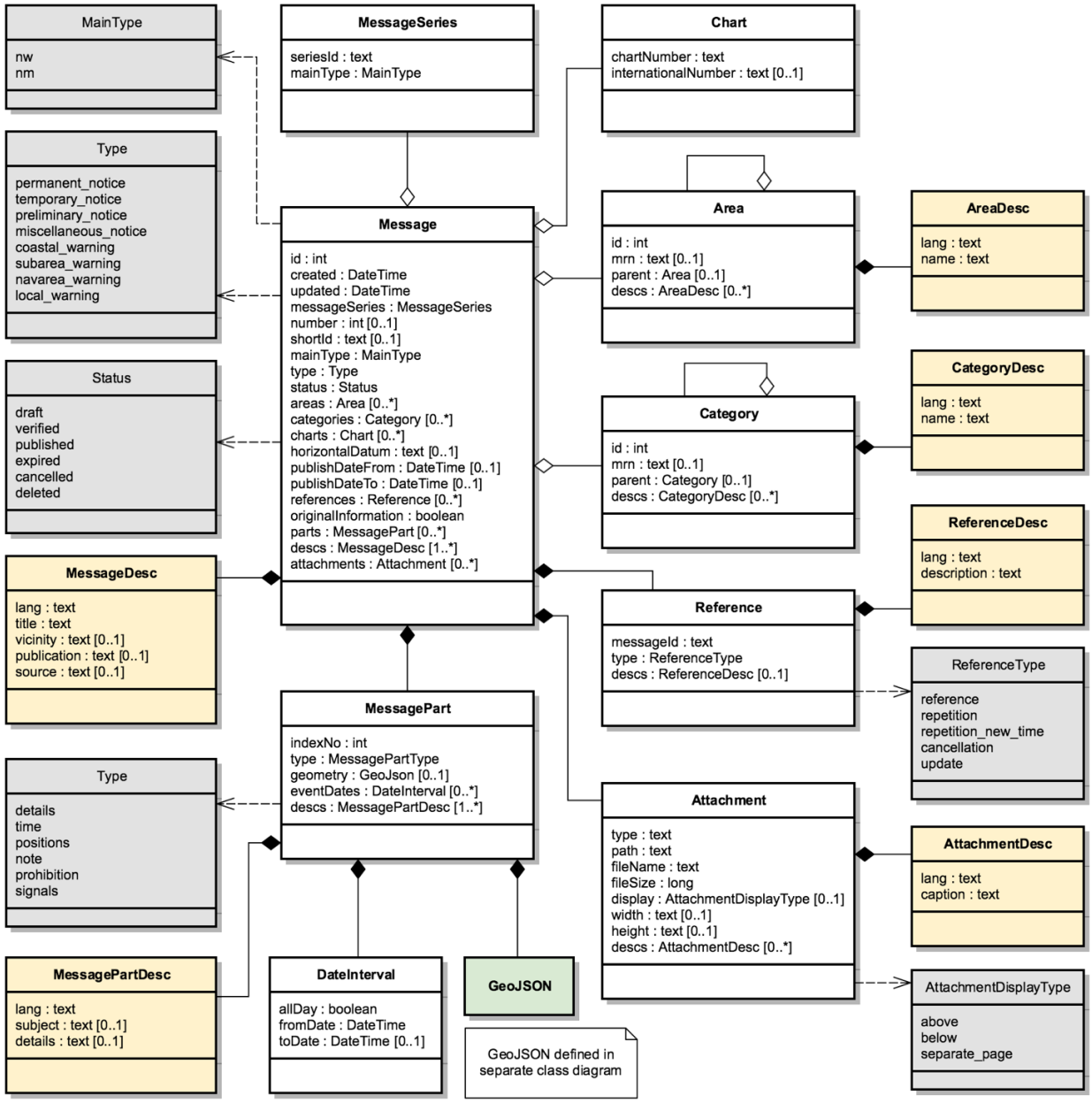


Figure 2: Message UML class diagram

The diagram uses the following colour codes:

- Light-gray background: Used for enumerations.
- Light-yellow background: Used for localized description entities - see *Design Pattern* section below.
- Light-green background: The *GeoJSON* model is expanded below:

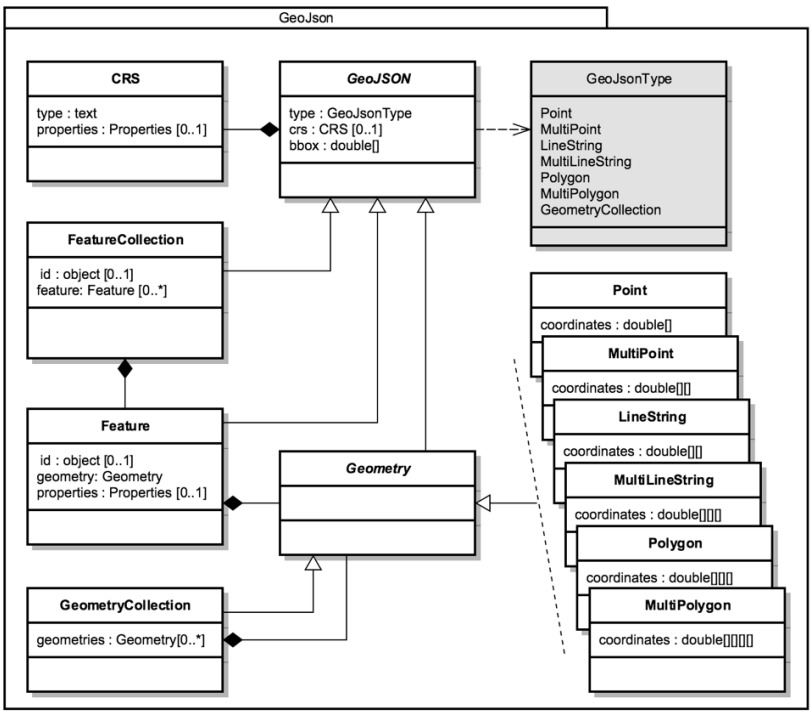


Figure 3: GeoJSON UML class diagram

The use of aggregation vs composition connectors above is mostly academic, since the UML is not a database model but merely an interchange format. However, the aggregation connector is used to signal that the associated entity represents base data in the producing system, and is not tied to the life cycle of the Message.
The remainder of the chapter will detail the individual classes.

5.1 MessageSeries

According to IHO, NW and NM messages must be numbered. For NW, it is e.g. mandated that:

> *Navigational warnings in each series should be consecutively numbered throughout the calendar year, commencing with 1/YY at 0000 UTC on 1 January.*

The numbering scheme does not guarantee uniqueness in a combined NW-NM model, let alone a system that may contain messages from multiple countries and authorities. Thus, *message series* have been introduced in the NW-NM data model to group messages as appropriate. A country may e.g. have separate message series for NW and NM. However, they may also introduce separate message series to allow, say, local harbour authorities to maintain their own message series for local NWs, or, as is the case with Canada, divide the country into five regions, each with their own message series.

Attribute Name	Type	Description
seriesId	String	The ID of the message series in the implementing system. Should be globally unique
mainType	MainType	Either NW or NM.

5.2 Chart

A message can be assigned a list of *charts*. The charts are maintained administratively as base data in the producing system.

Attribute Name	Type	Description
chartNumber	String	Mandatory regional chart number (and identifier).
internationalNumber	String	Optional international chart number.

5.3 Area

Existing IHO standards for NW and NM both provide support for specifying multiple area levels (*general area* and *locality* for NW; *general region, sub-region* and *specific location* for NMs). However, in the NW-NM system, this has been generalized, and areas are administratively maintained in a hierarchical area tree (with each area having a localized name) of arbitrary depth. A message can be assigned a list of these areas, and by implication, the parent areas of the selected area.
Additionally, a message can be assigned a localized textual *vicinity* description (part of the *MessageDesc* class – see 5.16), for detailed location information not defined in the area tree.

Attribute Name	Type	Description
id	Int	Internal system ID of the area
mrn	String	Optionally, an area may be assigned a globally unique MRN (maritime resource name). Adopting MRNs for areas would make interchange of message data between two NW-NM systems more robust.
parent	Area	Non-root areas will reference their parent areas. Example: Randers Havn -> Kattegat -> Danmark
descs	AreaDesc[]	The list of localizable attributes for an area. See 5.4.

5.4 AreaDesc

The AreaDesc class contains the list of localizable attributes for an area.

Attribute Name	Type	Description
lang	String	The ISO 639-1 language code.
name	String	The localized name of an area.

5.5 Category

Categories are administratively maintained in a hierarchical category tree (with each category having a localized name) of arbitrary depth. A message can be assigned a list of these categories, and by implication, the parent categories of a selected category.
At the top level, the categories will have entries such as *Aids to Navigation, Drifting Objects, Obstruction,* etc., which is the categorization used in the IHO standards. The sub-categories will represent the types of hazard relevant to the parent category. Examples of category lineages (top-down):

- Buoy -> Buoy Established
- Obstruction -> Wreck -> Marked Wreck

Attribute Name	Type	Description
id	Int	Internal system ID of the category
mrn	String	Optionally, a category may be assigned a globally unique MRN (maritime resource name). Adopting MRNs for categories would make interchange of message data between two NW-NM systems more robust.
parent	Category	Non-root category will reference their parent categories.

Attribute Name	Type	Description
descs	CategoryDesc[]	The list of localizable attributes for a category. See 5.6.

5.6 CateogryDesc

The CategoryDesc class contains the list of localizable attributes for a category.

Attribute Name	Type	Description
lang	String	The ISO 639-1 language code.
name	String	The localized name of a category.

5.7 Reference

The Reference class provides a typed, weak reference to a message.

Attribute Name	Type	Description
messageId	String	An identifier of the referenced message. If the messageId is recognized to be a *short-ID* of another message, it can e.g. be used to hyperlink to that message. However there are no requirements as to the format of the message ID.
type	ReferenceType	The type of the reference. One of the values “reference”, “repetition”, “repetition_new_time”, “cancellation” or “update”.
descs	ReferenceDesc[]	The list of localizable attributes for a Reference. See 5.8.

5.8 ReferenceDesc

The ReferenceDesc class contains the list of localizable attributes for a reference.

Attribute Name	Type	Description
lang	String	The ISO 639-1 language code.
description	String	The localized name of a reference.

5.9 Attachment

Messages can be associated with a list of attachments, such as images, PDF-files, etc. The physical attachment files will reside in a public repository on the producing system.

Attribute Name	Type	Description
type	String	The content type of the attachment file, such as "image/png".
path	String	The URL path to the attachment file in the producing system.
filename	String	The file name of the attachment.
fileSize	Long	The size of the attachment in bytes.
display	AttachmentDisplayType	If defined, this flag can be used to signal how the editor intended for the (image or video) attachment to be displayed when rendering the message for the end user. "above" and "below" signals that the attachment should be displayed above, respectively below, the message details. "separate_page" signals that the attachment

Attribute Name	Type	Description
		should be displayed on a separate page if rendered in paged media, such as a PDF file.
width	String	The width to use when displaying the (image or video) attachment. The width must include the type (i.e. *em, px, %, cm, mm, in, pt* or *pc*). If the *height* attribute is left unspecified, the attachment should be scaled proportionally.
height	String	The height to use when displaying the (image or video) attachment. The height must include the type (i.e. *em, px, %, cm, mm, in, pt* or *pc*). If the *width* attribute is left unspecified, the attachment should be scaled proportionally.
Descs	AttachmentDesc[]	The list of localizable attributes for an Attachment. See 5.10.

5.10 AttachmentDesc

The AttachmentDesc class contains the list of localizable attributes for an attachment.

Attribute Name	Type	Description
lang	String	The ISO 639-1 language code.
caption	String	A localized caption to display for the attachment.

5.11 DateInterval

A message will have an associated list of (possibly open-ended) date-time intervals, as defined by the DateInterval class. This defines the period of time for which the hazard described by the message applies.

Additionally, a message can be assigned a localized textual *time* description (part of the *MessageDesc* class – see 5.16).

Attribute Name	Type	Description
allDay	Boolean	If the *allDay* flag is set, the fromDate/toDate attributes should be treated as dates without a time-part by the producing system.
fromDate	DateTime	The start date-time of a date interval.
toDate	DateTime	An optional end date-time of a date interval.

5.12 MessagePart

A message defines an ordered list of *message parts*, which can be thought of as sub-stories. Conceptually, each message part defines the time, positions, key subject and description of the hazard or event that the story pertains to.

Attribute Name	Type	Description
indexNo	Int	Specified the ascending index of the message part within the message.
type	MessagePartType	May be used by the client to tag the message tag details with a type. Valid types are "details", "time", "positions", "note", "prohibition" and "signals".
geometry	GeoJSON	The positions of the message part. The GeoJSON

Attribute Name	Type	Description
		type is treated in more details in 5.14.. In practice, Niord will always return a *FeatureCollection* GeoJSON entity..
eventDates	DateTime[]	The list of event dates for which the message part hazard pertains. The list should not be rendered for the end user by the client, but may be used for computations.
descs	MessagePartDesc[]	The list of localizable attributes for a MessagePart. See 5.13.

5.13 MessagePartDesc

The MessagePartDesc class contains the list of localizable attributes for a message part.

Attribute Name	Type	Description
lang	String	The ISO 639-1 language code.
subject	String	The key subject of the hazard or event that the message part pertains to.
details	String	A detailed description of the hazard or event that the message part pertains to. The type of the *details* field is mandated to be HTML, and thus allows for fairly advanced layout and typography, and may contain elements such as tables, links, images, etc.

5.14 GeoJSON

The GeoJSON package is an implementation of an external data model, as defined at http://geojson.org/geojson-spec.html. The classes will not be detailed in this chapter.
GeoJSON was picked as the representation of a message geometry, because it is widely adopted by client libraries, and, unlike e.g. WKT, the GeoJSON Feature class has associated properties that can be used to store various information, that may be used in the portrayal of the messages.
The NW-NM system thus defines the following GeoJSON Feature properties, that a client may use for improved portrayal (but is not mandated to do so):

Property Name	Description
name:<<lang>>	Contains a language specific name that can be displayed for the geometry of the entire Feature. Example: name:en = Area of reduced depth.
name:<<x>>:<<lang>>	Contains a language specific name that can be displayed for the x'th coordinate of the geometry of the Feature. Example: name:12:en = yellow spar buoy with topmark.
parentFeatureId radius radiusType	These properties are mostly used by the producing system to let a geometry be defined from another geometry. As an example, an *affected area* may be defined as a buffered geometry with a radius of 200 meters around, say, the position of a wreck (this being the parent geometry). Clients may choose to adjust the portrayal of Features containing these properties, to signal that the feature represents an affected areas, and not the hazard itself.

5.15 Message

The Message class represents either an NW or an NM message.
It has been a deliberate, if slightly controversial, choice to let NWs and NMs share the same Message class, rather than having a separate sub-class for each type. The main rationale for this is that NWs and NMs T&P are expected to converge in the future, once promulgation is handled completely via electronic means.

Attribute Name	Type	Description
id	Int	Internal system ID of the message
created	DateTime	The timestamp the message was created in the system.
updated	DateTime	The timestamp the message was last updated in the system.
messageSeries	MessageSeries	The message series of the message. See 5.1.
number	Int	The sequence number of a published message. See 5.1.
shortId	String	The short-ID of a published message. See 5.1.
mainType	MainType	The main type of the message, either NW or NM. In reality, this attribute is redundant, since the main type is also defined by the associated message series, and may be implied by the message type.
type	Type	The sub-type of the message. One of: • permanent_notice • temporary_notice • preliminary_notice • miscellaneous_notice • coastal_warning • subarea_warning • navarea_warning • local_warning
status	Status	The status of the message. One of: • draft: • verified • published • expired • cancelled • deleted It is up to the producing system to manage the message life cycle, and enforce rules, such as: • Messages are created as drafts. • They can be assigned a secondary draft status, verified, to signal that they are ready for publication. • Only draft and verified messages can be deleted or published. • A published message can be cancelled manually or expired by the system if all associated date-intervals have passed.
areas	Area[]	A list of the areas of a message. See 5.3.
categories	Category[]	A list of the categories of a message. See 5.5.
charts	Chart[]	A list of the charts of a message. See 5.2.

Attribute Name	Type	Description
horizontalDatum	String	The horizontal datum for the message. If unspecified, assume WGS-84.
publishDateFrom	DateTime	The timestamp for when the message was published – or should be published.
publishDateTo	DateTime	The timestamp for when the message was cancelled – or should be expired.
references	Reference[]	A list of message references. See 5.7.
originalInformation	Boolean	If the message was based on original information or not.
parts	MessagePart[]	The list of message parts of the message. See 5.12.
descs	MessageDesc[]	The list of localizable attributes for a Message. See 5.16.
attachments	Attachment[]	The list of message attachments of the message. See 5.9.

5.16 MessageDesc

The MessageDesc class contains the list of localizable attributes for a message.

Attribute Name	Type	Description
lang	String	The ISO 639-1 language code.
title	String	A complete title line to show for a message. Typically composed by concatenating the area lineage, vicinity and subject of a message. Example: “Denmark. The North Sea. Hanstholm SW. AIS buoyage established.”
vicinity	String	May be used for localized arbitrary area information not defined in the area tree. See 5.3.
publication	String	A textual listing of all the publications relevant to the message. The format of the publication field is mandated to be HTML, as it may contain links to the actual publications.
source	String	The source of the message hazard information.

6 Service Interface Specifications

This chapter describes the details of the NW-NM service interface.

6.1 Service Interface MessageService

The NW-NM service is comprised of a single service (MessageService) and operation, which follows the Request/Reply Message Exchange Pattern. Sequence diagram:

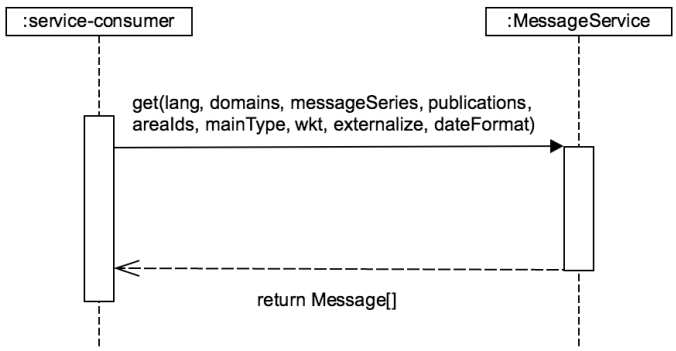


Figure 1: NW-NM Service Sequence Diagram

6.1.1 Operation get()

The get() operation returns the list of all published NW and NM messages. The returned result can be controlled using the following parameters:

Parameter	Type	Description
lang	String	An optional ISO 639-1 language code. If specified, only this language variant is returned for localized entities such as Message, Area and Category. However, if, say, “en” is requested and an entity only has a “da” language description entity, then this is returned instead. A client may want to flag this to the end user.
domains	String[]	List of domain ID’s. Each Niord system organizes messages into domains, e.g. for Navigational Warnings, Notices to Mariners, Firing Exercises, Almanacs, etc.
messageSeries	String[]	List of message series ID’s. Each message is associated with a message series, as defined by the producing Niord system. As an example, there may be separate messages series for local navigational warnings and other navigational warnings.
publications	String[]	Some messages may have been published in *publications*, such as a weekly NtM. The *publication* parameter can be used to filter by specific publications.
arealds	String[]	Messages may be associated with one or more area, as defined by the producing Niord System. The *areald* parameter can be used to filter messages by areas using either the internal ID or the MRN of the areas.
mainType	String	Used to filter messages by their main type, i.e. "NM" or

Parameter	Type	Description
		"NW".
wkt	String	Optional Well-Known Text representation of a geometry. If specified, only messages intersecting the geometry will be returned.
externalize	Boolean	Whether to rewrite all embedded links and paths to be absolute URL's pointing back to their repository URL in the producing system. Default value is *true*.
dateFormat	String	The date format to use for JSON date-time encoding. Either *UNIX_EPOCH* (default) or *ISO_8601*.
Return	**Type**	**Description**
	Message[]	The list of matching published NW and NM.

7 References

Nr.	Reference
[1] Service Description Guidelines	E2_Deliverable D3.4 – Service Specification Template, version 1.0
[2] NW-NM Service Description	E2 – 3.1.3 NW-NM service description and requirements gathering, version 0.1
[3] Maritime Cloud	Documented at www.maritimecloud.net
[4] MSI-NM in the ACCSEAS project	ACCSEAS MSI-NM Annex, version 1.0
[5] MSI-NM S-100 proposal	ACCSEAS MSI-NM S-100 Product Specification input paper, version 1.0.
[S-4]	Regulations of the IHO for International Charts and Chart Specifications of the IHO. Edition 4.3.0, August 2012, International Hydrographic Bureau, Monaco.
[S-53]	Manual on Maritime Safety Information (MSI). Special Publication No. 53, July 2009 Edition. International Hydrographic Bureau, Monaco.
[S-100]	Universal Hydrographic Data Model. IHO Special Publication No. S-100, Edition 1.0.0, January 2010. International Hydrographic Bureau, Monaco.

8 Acronyms and Terminology

8.1 Acronyms

Term	Definition
API	Application Programming Interface
DMA	Danish Maritime Authority
ECDIS	Electronic Chart Display Information Systems
ENC	Electronic Navigational Chart
EPD	e-Navigation Prototype Display
IHO	International Hydrographic Organisation
MC	Maritime Cloud
MEP	Message Exchange Pattern
MRN	Maritime Resource Name
MSI	Maritime Safety Information
NAF	NATO Architectural Framework
NM	Notices to Mariners
NW	Navigational Warning
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
SSD	Service Specification Document
UML	Unified Modelling Language
URL	Uniform Resource Locator
VTS	Vessel Traffic Service
WSDL	Web Service Definition Language
XML	Extendible Mark-up Language
XSD	XML Schema Definition

8.2 Terminology

Term	Definition
External Data Model	Describes the semantics of the “maritime world” (or a significant part thereof) by defining data structures and their relations. This could be at logical level (e.g., in UML) or at physical level (e.g., in XSD schema definitions), as for example standard data models, or S-100 based data produce specifications.
Navigational Warnings	Navigational Warnings (NW) are part of the Maritime Safety Information (MSI) system. Currently, NW’s are promulgated in text via SafetyNET, NAVTEX, and is in some countries accessible on the WWW or as voice broadcasts via coastal radio stations.
Niord	Anglified name of the Norse God Njord, associated with the sea and seafaring. Also the name of the EfficienSea 2 sub-project implementation of the NW-NM service and authoring system. See http://niord.org
Notices to Mariners	Notices to Mariners (NM) are promulgated weekly in order to keep nautical charts and publications, as far as possible, up to

	date. Temporary (T) and Preliminary (P) NMs advise mariners of important matters affecting navigational safety, including new hydrographic information (in advance of new editions or chart updates), changes to routing measures and aids to navigation, and other important categories of data. NM T&P's are today promulgated on paper weekly, fortnightly or monthly and are often accessible on the WWW in PDF format. Not all ENCs include T&P information currently.
Operational Activity	An activity performed by an operational node. Examples of operational activities in the maritime context are: Route Planning, Route Optimization, Logistics, Safety, Weather Forecast Provision, …
Operational Model	A structure of operational nodes and associated operational activities and their inter-relations in a process model.
Operational Node	A logical entity that performs activities. Note: nodes are specified independently of any physical realisation. Examples of operational nodes in the maritime context are: Maritime Control Center, Maritime Authority, Ship, Port, Weather Information Provider, …
Service	The contractual provision of something (a non-physical object), by one, for the use of one or more others. Services involve interactions between providers and consumers, which may be performed in a digital form (data exchanges) or through voice communication or written processes and procedures.
Service Consumer	A service consumer uses service instances provided by service providers. All users within the maritime domain can be service customers, e.g., ships and their crew, authorities, VTS stations, organizations (e.g., meteorological), commercial service providers, etc.
Service Data Model	Formal description of one dedicated service at logical level. The service data model is part of the service specification. Is typically defined in UML and/or XSD. If an external data model exists (e.g., a standard data model), then the service data model shall refer to it: each data item of the service data model shall be mapped to a data item defined in the external data model.
Service Implementer	Implementers of services from the service provider side and/or the service consumer side. Everybody can be a service implementer but mainly this will be commercial companies implementing solutions for shore and ship.
Service Instance	The implementation of a dedicated service in a dedicated technology. One service specification may result in several service instances, being implemented with different or same technologies.
Service Instance Description	Documents the details of a service instance (most likely documented by the service implementer). The service instance description includes (but is not limited to) a service instance model and describes the used technology, transport mechanism, quality of service, etc.
Service Instance	Describes the implementation of a dedicated service instance in

Model	a dedicated technology. This includes a detailed description of the data payload to be exchanged by this service instance. The actual format of the service instance model depends on the chosen technology. Examples may be WSDL and XSD files (e.g., for SOAP services) or swagger (Open API) specifications (e.g., for REST services). If an external data model exists (e.g., a standard data model), then the service instance model shall refer to it: each data item of the service instance model shall be mapped to a data item defined in the external data model. In order to prove correct implementation of the service specification, there shall exist a mapping between the service instance model and the service data model. This means, each data item used in the service instance model shall be mapped to a corresponding data item of the service data model. (In case of existing mappings to a common external (standard) data model from both the service data model and the service iInstance model, such a mapping is implicitly given.)
Service Interface	The mechanism by which a service communicates.
Service Provider	A service provider provides instances of services according to a service specification and service instance description. All users within the maritime domain can be service providers, e.g., authorities, VTS stations, organizations (e.g., meteorological), commercial service providers, etc.
Service Specification	Describes one dedicated service at logical level. The Service Specification is technology-agnostic. The Service Specification includes (but is not limited to) a description of the Service Interfaces and Service Operations with their data payload. The data payload description may be formally defined by a Service Data Model.
Service Specification Producer	Producers of service specifications in accordance with the service description guidelines.
Service Technology Catalogue	List and specifications of allowed technologies for service implementations. Currently, SOAP and REST are envisaged to be allowed service technologies. The service technology catalogue shall describe in detail the allowed service profiles, e.g., by listing communication standards, security standards, stacks, bindings, etc.

Appendix A Service Specification XML

```
<?xml version="1.0" encoding="UTF-8"?>
<serviceSpecification
      xmlns="http://efficiensea2.org/maritime-cloud/service-registry/v1/ServiceSpecificationSchema.xsd"
      xmlns:ServiceSpecificationSchema="http://efficiensea2.org/maritime-cloud/service-
registry/v1/ServiceSpecificationSchema.xsd"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://efficiensea2.org/maritime-cloud/service-
registry/v1/ServiceSpecificationSchema.xsd ServiceSpecificationSchema.xsd "
      xmlns:xs="http://www.w3.org/2001/XMLSchema" >

   <name>NW-NM TP Maritime Cloud Service</name>
   <status>released</status>
   <id>urn:mrn:mcl:service:specification:dma:nw-nm</id>
   <version>0.4</version>
   <description>The NW-NM service specification defines a combined NW-NM TP model along with the actual
service API used for accessing NW-NM data, as registered in the Maritime Cloud service
catalogue.</description>

   <keywords>NW, NM, Navigational Warnings, Notices to Mariners, MSI, Maritime Cloud Service</keywords>
   <isSpatialExclusive>false</isSpatialExclusive>

   <authorInfos>
      <authorInfo>
         <id>urn:mrn:mcl:user:dma:mfs</id>
         <name>Mads Friis Sørensen</name>
         <description>Responsible for the NW-NM service</description>
         <contactInfo>mfs@dma.dk</contactInfo>
      </authorInfo>
   </authorInfos>

   <requirements>
      <requirement>
         <id>urn:mrn:mcl:requirement:nw-nm:1</id>
         <name>Combined NW-NM model</name>
         <text>The data model should encapsulate a combined NW-NM model.</text>
      </requirement>
      <requirement>
         <id>urn:mrn:mcl:requirement:nw-nm:2</id>
         <name>Return all published NW-NM messages.</name>
         <text>The NW-NM service should make it possible to retrieve all published NW-NM messages from
the given service provider.</text>
      </requirement>
   </requirements>


   <serviceDataModel>
      <definitionAsXSD>

         <xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

            <xs:element name="area" type="areaVo"/>

            <xs:element name="attachment" type="attachmentVo"/>

            <xs:element name="category" type="categoryVo"/>

            <xs:element name="chart" type="chartVo"/>

            <xs:element name="feature" type="featureVo"/>

            <xs:element name="featureCollection" type="featureCollectionVo"/>

            <xs:element name="geometryCollection" type="geometryCollectionVo"/>

            <xs:element name="lineString" type="lineStringVo"/>
```

```
            <xs:element name="message" type="messageVo"/>

            <xs:element name="messageSeries" type="messageSeriesVo"/>

            <xs:element name="multiLineString" type="multiLineStringVo"/>

            <xs:element name="multiPoint" type="multiPointVo"/>

            <xs:element name="multiPolygon" type="multiPolygonVo"/>

            <xs:element name="point" type="pointVo"/>

            <xs:element name="polygon" type="polygonVo"/>

            <xs:complexType name="doubleArray" final="#all">
               <xs:sequence>
                  <xs:element name="item" type="xs:double" minOccurs="0" maxOccurs="unbounded"
nillable="true"/>
               </xs:sequence>
            </xs:complexType>

            <xs:complexType name="doubleArrayArray" final="#all">
               <xs:sequence>
                  <xs:element name="item" type="doubleArray" minOccurs="0" maxOccurs="unbounded"
nillable="true"/>
               </xs:sequence>
            </xs:complexType>

            <xs:complexType name="doubleArrayArrayArray" final="#all">
               <xs:sequence>
                  <xs:element name="item" type="doubleArrayArray" minOccurs="0" maxOccurs="unbounded"
nillable="true"/>
               </xs:sequence>
            </xs:complexType>

            <xs:complexType name="messageVo">
               <xs:sequence>
                  <xs:element ref="messageSeries" minOccurs="0"/>
                  <xs:element name="number" type="xs:int" minOccurs="0"/>
                  <xs:element name="shortId" type="xs:string" minOccurs="0"/>
                  <xs:element name="mainType" type="mainType" minOccurs="0"/>
                  <xs:element name="type" type="type" minOccurs="0"/>
                  <xs:element name="status" type="status" minOccurs="0"/>
                  <xs:element name="areas" type="areaVo" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
                  <xs:element name="categories" type="categoryVo" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
                  <xs:element name="charts" type="chartVo" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
                  <xs:element name="horizontalDatum" type="xs:string" minOccurs="0"/>
                  <xs:element name="publishDateFrom" type="xs:dateTime" minOccurs="0"/>
                  <xs:element name="publishDateTo" type="xs:dateTime" minOccurs="0"/>
                  <xs:element name="followUpDate" type="xs:dateTime" minOccurs="0"/>
                  <xs:element name="references" type="referenceVo" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
                  <xs:element name="originalInformation" type="xs:boolean" minOccurs="0"/>
                  <xs:element name="parts" type="messagePartVo" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
                  <xs:element name="descs" type="messageDescVo" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
                  <xs:element name="attachments" type="attachmentVo" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
               </xs:sequence>
               <xs:attribute name="created" type="xs:dateTime"/>
               <xs:attribute name="id" type="xs:string"/>
               <xs:attribute name="updated" type="xs:dateTime"/>
            </xs:complexType>

            <xs:complexType name="messageSeriesVo">
               <xs:sequence>
                  <xs:element name="seriesId" type="xs:string" minOccurs="0"/>
                  <xs:element name="mainType" type="mainType" minOccurs="0"/>
```

```
            </xs:sequence>
         </xs:complexType>

         <xs:complexType name="areaVo">
            <xs:sequence>
               <xs:element name="mrn" type="xs:string" minOccurs="0"/>
               <xs:element name="active" type="xs:boolean"/>
               <xs:element name="parent" type="areaVo" minOccurs="0"/>
               <xs:element name="descs" type="areaDescVo" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="id" type="xs:int"/>
         </xs:complexType>

         <xs:complexType name="areaDescVo">
            <xs:sequence>
               <xs:element name="name" type="xs:string" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="lang" type="xs:string"/>
         </xs:complexType>

         <xs:complexType name="categoryVo">
            <xs:sequence>
               <xs:element name="mrn" type="xs:string" minOccurs="0"/>
               <xs:element name="active" type="xs:boolean"/>
               <xs:element name="parent" type="categoryVo" minOccurs="0"/>
               <xs:element name="descs" type="categoryDescVo" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="id" type="xs:int"/>
         </xs:complexType>

         <xs:complexType name="categoryDescVo">
            <xs:sequence>
               <xs:element name="name" type="xs:string" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="lang" type="xs:string"/>
         </xs:complexType>

         <xs:complexType name="chartVo">
            <xs:sequence>
               <xs:element name="chartNumber" type="xs:string" minOccurs="0"/>
               <xs:element name="internationalNumber" type="xs:int" minOccurs="0"/>
               <xs:element name="active" type="xs:boolean"/>
               <xs:element name="scale" type="xs:int" minOccurs="0"/>
               <xs:element name="name" type="xs:string" minOccurs="0"/>
            </xs:sequence>
         </xs:complexType>

         <xs:complexType name="referenceVo">
            <xs:sequence>
               <xs:element name="messageId" type="xs:string" minOccurs="0"/>
               <xs:element name="type" type="referenceType" minOccurs="0"/>
               <xs:element name="descs" type="referenceDescVo" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
         </xs:complexType>

         <xs:complexType name="referenceDescVo">
            <xs:sequence>
               <xs:element name="description" type="xs:string" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="lang" type="xs:string"/>
         </xs:complexType>

         <xs:complexType name="messagePartVo">
            <xs:sequence>
               <xs:element name="indexNo" type="xs:int"/>
               <xs:element name="type" type="messagePartType" minOccurs="0"/>
               <xs:element name="eventDates" type="dateIntervalVo" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
               <xs:element name="geometry" type="featureCollectionVo" minOccurs="0"/>
```

```
                <xs:element name="descs" type="messagePartDescVo" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="hideSubject" type="xs:boolean" minOccurs="0"/>
             </xs:sequence>
          </xs:complexType>

          <xs:complexType name="dateIntervalVo">
             <xs:sequence>
                <xs:element name="allDay" type="xs:boolean" minOccurs="0"/>
                <xs:element name="fromDate" type="xs:dateTime" minOccurs="0"/>
                <xs:element name="toDate" type="xs:dateTime" minOccurs="0"/>
             </xs:sequence>
          </xs:complexType>

          <xs:complexType name="featureCollectionVo">
             <xs:sequence>
                <xs:element name="bbox" type="xs:double" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="crs" type="crs" minOccurs="0"/>
                <xs:element ref="feature" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="id" type="xs:anyType" minOccurs="0"/>
                <xs:element name="type" type="xs:string" minOccurs="0"/>
             </xs:sequence>
          </xs:complexType>

          <xs:complexType name="crs">
             <xs:sequence>
                <xs:element name="properties">
                   <xs:complexType>
                      <xs:sequence>
                         <xs:element name="entry" minOccurs="0" maxOccurs="unbounded">
                            <xs:complexType>
                               <xs:sequence>
                                  <xs:element name="key" minOccurs="0" type="xs:anyType"/>
                                  <xs:element name="value" minOccurs="0" type="xs:anyType"/>
                               </xs:sequence>
                            </xs:complexType>
                         </xs:element>
                      </xs:sequence>
                   </xs:complexType>
                </xs:element>
                <xs:element name="type" type="xs:string" minOccurs="0"/>
             </xs:sequence>
          </xs:complexType>

          <xs:complexType name="featureVo">
             <xs:sequence>
                <xs:element name="bbox" type="xs:double" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="crs" type="crs" minOccurs="0"/>
                <xs:choice>
                   <xs:element ref="point"/>
                   <xs:element ref="multiPoint"/>
                   <xs:element ref="lineString"/>
                   <xs:element ref="multiLineString"/>
                   <xs:element ref="polygon"/>
                   <xs:element ref="multiPolygon"/>
                   <xs:element ref="geometryCollection"/>
                </xs:choice>
                <xs:element name="id" type="xs:anyType" minOccurs="0"/>
                <xs:element name="properties">
                   <xs:complexType>
                      <xs:sequence>
                         <xs:element name="entry" minOccurs="0" maxOccurs="unbounded">
                            <xs:complexType>
                               <xs:sequence>
                                  <xs:element name="key" minOccurs="0" type="xs:string"/>
                                  <xs:element name="value" minOccurs="0" type="xs:anyType"/>
                               </xs:sequence>
                            </xs:complexType>
                         </xs:element>
                      </xs:sequence>
```

```
                    </xs:complexType>
                </xs:element>
                <xs:element name="type" type="xs:string" minOccurs="0"/>
             </xs:sequence>
          </xs:complexType>

          <xs:complexType name="geometry" abstract="true">
             <xs:sequence>
                <xs:element name="bbox" type="xs:double" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="crs" type="crs" minOccurs="0"/>
                <xs:element name="type" type="xs:string" minOccurs="0"/>
             </xs:sequence>
          </xs:complexType>

          <xs:complexType name="pointVo">
             <xs:complexContent>
                <xs:extension base="geometry">
                   <xs:sequence>
                      <xs:element name="coordinates" type="xs:double" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
                   </xs:sequence>
                </xs:extension>
             </xs:complexContent>
          </xs:complexType>

          <xs:complexType name="multiPointVo">
             <xs:complexContent>
                <xs:extension base="geometry">
                   <xs:sequence>
                      <xs:element name="coordinates" type="doubleArray" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
                   </xs:sequence>
                </xs:extension>
             </xs:complexContent>
          </xs:complexType>

          <xs:complexType name="lineStringVo">
             <xs:complexContent>
                <xs:extension base="geometry">
                   <xs:sequence>
                      <xs:element name="coordinates" type="doubleArray" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
                   </xs:sequence>
                </xs:extension>
             </xs:complexContent>
          </xs:complexType>

          <xs:complexType name="multiLineStringVo">
             <xs:complexContent>
                <xs:extension base="geometry">
                   <xs:sequence>
                      <xs:element name="coordinates" type="doubleArrayArray" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
                   </xs:sequence>
                </xs:extension>
             </xs:complexContent>
          </xs:complexType>

          <xs:complexType name="polygonVo">
             <xs:complexContent>
                <xs:extension base="geometry">
                   <xs:sequence>
                      <xs:element name="coordinates" type="doubleArrayArray" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
                   </xs:sequence>
                </xs:extension>
             </xs:complexContent>
          </xs:complexType>

          <xs:complexType name="multiPolygonVo">
             <xs:complexContent>
```

```
                  <xs:extension base="geometry">
                     <xs:sequence>
                        <xs:element name="coordinates" type="doubleArrayArrayArray" nillable="true"
minOccurs="0" maxOccurs="unbounded"/>
                     </xs:sequence>
                  </xs:extension>
               </xs:complexContent>
            </xs:complexType>

            <xs:complexType name="geometryCollectionVo">
               <xs:complexContent>
                  <xs:extension base="geometry">
                     <xs:sequence>
                        <xs:choice minOccurs="0" maxOccurs="unbounded">
                           <xs:element ref="point"/>
                           <xs:element ref="multiPoint"/>
                           <xs:element ref="lineString"/>
                           <xs:element ref="multiLineString"/>
                           <xs:element ref="polygon"/>
                           <xs:element ref="multiPolygon"/>
                           <xs:element ref="geometryCollection"/>
                        </xs:choice>
                     </xs:sequence>
                  </xs:extension>
               </xs:complexContent>
            </xs:complexType>

            <xs:complexType name="messagePartDescVo">
               <xs:sequence>
                  <xs:element name="subject" type="xs:string" minOccurs="0"/>
                  <xs:element name="details" type="xs:string" minOccurs="0"/>
               </xs:sequence>
               <xs:attribute name="lang" type="xs:string"/>
            </xs:complexType>

            <xs:complexType name="messageDescVo">
               <xs:sequence>
                  <xs:element name="title" type="xs:string" minOccurs="0"/>
                  <xs:element name="vicinity" type="xs:string" minOccurs="0"/>
                  <xs:element name="publication" type="xs:string" minOccurs="0"/>
                  <xs:element name="internalPublication" type="xs:string" minOccurs="0"/>
                  <xs:element name="source" type="xs:string" minOccurs="0"/>
               </xs:sequence>
               <xs:attribute name="lang" type="xs:string"/>
            </xs:complexType>

            <xs:complexType name="attachmentVo">
               <xs:sequence>
                  <xs:element name="type" type="xs:string" minOccurs="0"/>
                  <xs:element name="fileName" type="xs:string" minOccurs="0"/>
                  <xs:element name="path" type="xs:string" minOccurs="0"/>
                  <xs:element name="fileSize" type="xs:long" minOccurs="0"/>
                  <xs:element name="display" type="attachmentDisplayType" minOccurs="0"/>
                  <xs:element name="width" type="xs:string" minOccurs="0"/>
                  <xs:element name="height" type="xs:string" minOccurs="0"/>
                  <xs:element name="descs" type="attachmentDescVo" nillable="true" minOccurs="0"
maxOccurs="unbounded"/>
               </xs:sequence>
            </xs:complexType>

            <xs:complexType name="attachmentDescVo">
               <xs:sequence>
                  <xs:element name="caption" type="xs:string" minOccurs="0"/>
               </xs:sequence>
               <xs:attribute name="lang" type="xs:string"/>
            </xs:complexType>

            <xs:simpleType name="mainType">
               <xs:restriction base="xs:string">
                  <xs:enumeration value="NW"/>
                  <xs:enumeration value="NM"/>
               </xs:restriction>
```

```
            </xs:simpleType>

            <xs:simpleType name="type">
               <xs:restriction base="xs:string">
                  <xs:enumeration value="PERMANENT_NOTICE"/>
                  <xs:enumeration value="TEMPORARY_NOTICE"/>
                  <xs:enumeration value="PRELIMINARY_NOTICE"/>
                  <xs:enumeration value="MISCELLANEOUS_NOTICE"/>
                  <xs:enumeration value="COASTAL_WARNING"/>
                  <xs:enumeration value="SUBAREA_WARNING"/>
                  <xs:enumeration value="NAVAREA_WARNING"/>
                  <xs:enumeration value="LOCAL_WARNING"/>
               </xs:restriction>
            </xs:simpleType>

            <xs:simpleType name="status">
               <xs:restriction base="xs:string">
                  <xs:enumeration value="DRAFT"/>
                  <xs:enumeration value="VERIFIED"/>
                  <xs:enumeration value="PUBLISHED"/>
                  <xs:enumeration value="EXPIRED"/>
                  <xs:enumeration value="CANCELLED"/>
                  <xs:enumeration value="DELETED"/>
               </xs:restriction>
            </xs:simpleType>

            <xs:simpleType name="referenceType">
               <xs:restriction base="xs:string">
                  <xs:enumeration value="REFERENCE"/>
                  <xs:enumeration value="REPETITION"/>
                  <xs:enumeration value="REPETITION_NEW_TIME"/>
                  <xs:enumeration value="CANCELLATION"/>
                  <xs:enumeration value="UPDATE"/>
               </xs:restriction>
            </xs:simpleType>

            <xs:simpleType name="messagePartType">
               <xs:restriction base="xs:string">
                  <xs:enumeration value="DETAILS"/>
                  <xs:enumeration value="TIME"/>
                  <xs:enumeration value="POSITIONS"/>
                  <xs:enumeration value="NOTE"/>
                  <xs:enumeration value="PROHIBITION"/>
                  <xs:enumeration value="SIGNALS"/>
               </xs:restriction>
            </xs:simpleType>

            <xs:simpleType name="attachmentDisplayType">
               <xs:restriction base="xs:string">
                  <xs:enumeration value="ABOVE"/>
                  <xs:enumeration value="BELOW"/>
                  <xs:enumeration value="SEPARATE_PAGE"/>
               </xs:restriction>
            </xs:simpleType>
         </xs:schema>


      </definitionAsXSD>
   </serviceDataModel>

   <serviceInterfaces>
      <serviceInterface>
         <name>MessageService</name>
         <description>Works according to the request response pattern.</description>
         <dataExchangePattern>REQUEST_RESPONSE</dataExchangePattern>

         <operations>
            <operation>
               <name>get</name>
               <description>Retrieves published NW-NM messages.</description>
               <returnValueType>
                  <typeReference>Message[]</typeReference>
```

```
            </returnValueType>
            <parameterTypes>
               <parameterType>
                  <typeReference>lang</typeReference>
               </parameterType>
               <parameterType>
                  <typeReference>domains</typeReference>
               </parameterType>
               <parameterType>
                  <typeReference>messageSeries</typeReference>
               </parameterType>
               <parameterType>
                  <typeReference>publications</typeReference>
               </parameterType>
               <parameterType>
                  <typeReference>areaIds</typeReference>
               </parameterType>
               <parameterType>
                  <typeReference>mainType</typeReference>
               </parameterType>
               <parameterType>
                  <typeReference>wkt</typeReference>
               </parameterType>
               <parameterType>
                  <typeReference>externalize</typeReference>
               </parameterType>
               <parameterType>
                  <typeReference>dateFormat</typeReference>
               </parameterType>
            </parameterTypes>
         </operation>
      </operations>

   </serviceInterface>
</serviceInterfaces>

</serviceSpecification>
```

NW-NM REST Service Technical Design v0.4

Contents

1 Introduction

1.1 Purpose of the document

This document covers a REST-based technical design of the MW-NM service specification [1], according to the guidelines given in the Service Description Guidelines [2].

1.2 Intended readership

This service technical design is intended to be read by service architects, system engineers and developers in charge of designing and developing an instance of the REST-bases NW-NM service.

Furthermore, this service specification is intended to be read by enterprise architects, service architects, information architects, system engineers and developers in pursuing architecting, design and development activities of other related services.

2 Service Design Identification

Name	NW-NM T&P Maritime Cloud REST Service
ID	urn:mrnx:mcl:service:dma:nw-nm:rest
Version	0.4
Technology	REST
Service Specification ID	urn:mrnx:mcl:service:dma:nw-nm
Service Specification Version	0.4
Description	The NW-NM Maritime Cloud REST service specification defines a combined NW-NM T&P model in JSON, along with the actual REST service API used for accessing NW-NM data, as registered in the Maritime Cloud service catalogue.
Keywords	NW, NM, Navigational Warnings, Notices to Mariners, MSI, Maritime Cloud Service. REST.
Architect(s)	e-Navigation Team Danish Maritime Authority Carl Jacobsens Vej 31 DK-2500 København K Telephone: +45 40 72 61 08 Email: mfs@dma.dk
Status	Released.

3 Technology Introduction

3.1 REST

From the *Tide Level Information Technical Design*:

> *REST (REpresentational State Transfer) is one way of providing interoperability between system on the internet. It allows requesting systems to access and manipulate textual representations of web resources using a uniform and predefined set of stateless operations: more than efficiently WSDL and SOAP.*
>
> *In a web service which using REST, requests made to a resource's URI will elicit a response that maybe in XML, HTML, JSON or some other defined format. The response may confirm that some alteration has been made to the stored resource, and it may provide hypertext links to other related resources or collections of resources.*
>
> *Using HTTP, as is most common, the kind of operations available include those predefined by the HTTP verbs GET, POST, PUT, DELETE and so on. By making use of a stateless protocol and standard operations, REST aim for fast performance, reliability, and the ability to grow, by re-using components that can be managed and updated without affecting the system as a whole, even while it is running.*

For more details, please refer to https://en.wikipedia.org/wiki/Representational_state_transfer

4 Service Design Overview

This chapter will outline the REST implementation of the services described in the Service Specification [1].

4.1 Service Interfaces

The NW-NM REST service consists of a single public REST endpoint to query the currently published NW-NM messages (Request/Reply Message Exchange Pattern).

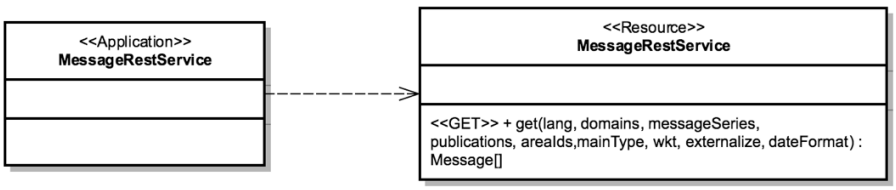


Figure 1: NW-NM Service Definition Diagram

ServiceInterface	Role (from service provider point of view)	ServiceOperation
MessageRestService	Provided	get(lang, domains, messageSeries, publications, areaIds, mainType, wkt, externalize, dateFormat)

The returned data model is detailed in chapter 5 and the service operation in chapter 6.

5 Physical Data Model

This chapter details the concrete JSON data model implementation of the data model described in the Service Specification [1].

5.1 Data Model

The JSON data model is detailed in Appendix B as a Swagger Specification (a.k.a. OpenAPI specification) [3].

There is a very direct 1:1 mapping between the UML/XML model detailed in the Service Specification and the JSON data model adopted in this technical design. In fact, both are generated from the same Java class library (niord-model).

Hence, this technical design does not provide an explicit mapping table between the Service Specification model and the Technical Design JSON model.

6 Service Interface Design

This chapter describes the details of the NW-NM REST service interface, which is the REST based implementation of the MessageService specified in the Service Specification [1].

6.1 Service Endpoint /public/v1/messages

The NW-NM service is comprised of a single REST endpoint, /public/v1/messages, which follows the Request/Reply Message Exchange Pattern. Sequence diagram:

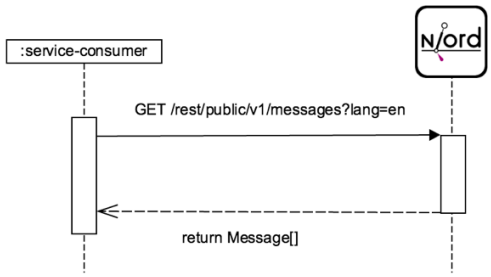


Figure 1: NW-NM Service Sequence Diagram

The formal service interface is detailed in Appendix B as a Swagger Specification (a.k.a. OpenAPI specification) [3].

6.1.1 Operation HTTP GET

The HTTP GET operation returns the list of all published NW and NM messages. The returned result can be controlled using the following parameters:

Parameter	Type	Description
lang	String	An optional ISO 639-1 language code. If specified, only this language variant is returned for localized entities such as Message, Area and Category. However, if, say, “en” is requested and an entity only has a “da” language description entity, then this is returned instead. A client may want to flag this to the end user.
domains	String[]	List of domain ID’s. Each Niord system organizes messages into domains, e.g. for Navigational Warnings, Notices to Mariners, Firing Exercises, Almanacs, etc.
messageSeries	String[]	List of message series ID’s. Each message is associated with a message series, as defined by the producing Niord system. As an example, there may be separate messages series for local navigational warnings and other navigational warnings.
publications	String[]	Some messages may have been published in *publications*, such as a weekly NtM.

Parameter	Type	Description
		The *publication* parameter can be used to filter by specific publications.
areaIds	String[]	Messages may be associated with one or more area, as defined by the producing Niord System. The *areaId* parameter can be used to filter messages by areas using either the internal ID or the MRN of the areas.
mainType	String	Used to filter messages by their main type, i.e. "NM" or "NW".
wkt	String	Optional Well-Known Text representation of a geometry. If specified, only messages intersecting the geometry will be returned.
externalize	Boolean	Whether to rewrite all embedded links and paths to be absolute URL's pointing back to their repository URL in the producing system. Default value is *true*.
dateFormat	String	The date format to use for JSON date-time encoding. Either *UNIX_EPOCH* (default) or *ISO_8601*.
Header	**values**	**Description**
Accept	application/json, application/xml	Request that the returned data is in the JSON or XML format. Only the JSON format is described in this document.
Return	**Type**	**Description**
	Message[]	The list of published NW and NM messages in the requested format.

7 References

Nr.	Reference
[1] NW-NM Service Specification	E2 - NW-NM Service Specification version 0.4
[2] Service Description Guidelines	E2_Deliverable D3.4 – Service Specification Template, version 1.0
[3] Swagger (OpenAPI) specification.	Available at https://github.com/OAI/OpenAPI-Specification

8 Acronyms and Terminology

8.1 Acronyms

Term	Definition
API	Application Programming Interface
DMA	Danish Maritime Authority
MC	Maritime Cloud
MEP	Message Exchange Pattern
MRN	Maritime Resource Name
NM	Notices to Mariners
NW	Navigational Warning
REST	Representational State Transfer
XML	Extendible Mark-up Language
XSD	XML Schema Definition

8.2 Terminology

Term	Definition
Navigational Warnings	Navigational Warnings (NW) are part of the Maritime Safety Information (MSI) system. Currently, NW's are promulgated in text via SafetyNET, NAVTEX, and is in some countries accessible on the WWW or as voice broadcasts via coastal radio stations.
Niord	Anglified name of the Norse God Njord, associated with the sea and seafaring. Also the name of the EfficienSea 2 sub-project implementation of the NW-NM service and authoring system. See http://niord.org
niord-model	A java implementation of the JSON/XML model used by the NW-NM REST service. Available at https://github.com/NiordOrg/niord/tree/master/niord-model
Notices to Mariners	Notices to Mariners (NM) are promulgated weekly in order to keep nautical charts and publications, as far as possible, up to date. Temporary (T) and Preliminary (P) NMs advise mariners of important matters affecting navigational safety, including new hydrographic information (in advance of new editions or chart updates), changes to routing measures and aids to navigation, and other important categories of data. NM T&P's are today promulgated on paper weekly, fortnightly or monthly and are often accessible on the WWW in PDF format. Not all ENCs include T&P information currently.
Swagger API	Swagger provides a framework (and utilities) for specifying a REST API.

Appendix A Service Design Description XML

This appendix contains the formal XML definition of the service Technical Design. The actual Swagger API has been included separately in Appendix B.

```
<?xml version="1.0" encoding="UTF-8"?>
<serviceDesign
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xsi:schemaLocation="http://efficiensea2.org/maritime-cloud/service-
registry/v1/ServiceDesignSchema.xsd ServiceDesignSchema.xsd "
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ServiceDesignSchema="http://efficiensea2.org/maritime-cloud/service-
registry/v1/ServiceDesignSchema.xsd"
   xmlns:ServiceSpecificationSchema="http://efficiensea2.org/maritime-cloud/service-
registry/v1/ServiceSpecificationSchema.xsd"
   xmlns="http://efficiensea2.org/maritime-cloud/service-registry/v1/ServiceDesignSchema.xsd">

  <id>urn:mrn:mcl:service:design:dma:nw-nm-rest</id>
  <version>0.4</version>
  <name>NW-NM TP Maritime Cloud REST Service</name>
  <status>released</status>
  <description>A REST-based implementation of the NW-NM Maritime Cloud service
specification.</description>
  <offersTransport>
    <offersTransport>
      <name>REST</name>
      <description>This service implementation is available as REST over HTTPS</description>
      <protocol>HTTPS</protocol>
    </offersTransport>
  </offersTransport>
  <designsServiceSpecifications>
    <designsServiceSpecifications>
      <id>urn:mrn:mcl:service:specification:dma:nw-nm</id>
      <version>0.4</version>
    </designsServiceSpecifications>
  </designsServiceSpecifications>
  <designedBy>
   <ServiceSpecificationSchema:id>urn:mrn:mcl:user:dma:mfs</ServiceSpecificationSchema:id>
   <ServiceSpecificationSchema:name>Mads Friis Sørensen</ServiceSpecificationSchema:name>
   <ServiceSpecificationSchema:description>Responsible for the NW-NM
service</ServiceSpecificationSchema:description>
   <ServiceSpecificationSchema:contactInfo>mfs@dma.dk</ServiceSpecificationSchema:contactInfo>
   <ServiceSpecificationSchema:isCommercial>false</ServiceSpecificationSchema:isCommercial>
  </designedBy>
  <servicePhysicalDataModel>
    <name>NW-NM REST Service Swagger API</name>
    <description>Swagger API of the NW-NM REST Service</description>
    <modelType>JSON</modelType>
    <model>
      http://niord.e-navigation.net/rest/swagger.json
    </model>
  </servicePhysicalDataModel>
</serviceDesign>
```

Appendix B Swagger API sans GeoJSON

This appendix contains the formal Swagger definition of the NW-NM REST service.

The complete Swagger definition can be found at http://niord.e-navigation.net/rest/swagger.json. The Swagger definition defines a few operations and classes not part of the NW-NM service specification. For clarity reasons, these have been removed from the Swagger file below:


```
{
  "swagger": "2.0",
  "info": {
    "version": "1.0.0"
  },
  "basePath": "/rest",
  "tags": [
    {
      "name": "messages"
    }
  ],
  "paths": {
    "/public/v1/messages": {
      "get": {
        "tags": [
          "messages"
        ],
        "summary": "Returns the published NW and NM messages",
        "description": "",
        "operationId": "searchMessages",
        "produces": [
          "application/json;charset=UTF-8"
        ],
        "parameters": [
          {
            "name": "lang",
            "in": "query",
            "description": "Two-letter ISO 639-1 language code",
            "required": false,
            "type": "string",
            "x-example": "en"
          },
          {
            "name": "domain",
            "in": "query",
            "description": "The IDs of the domains to select messages from",
            "required": false,
            "type": "array",
            "items": {
              "type": "string"
            },
            "collectionFormat": "multi",
            "x-example": "niord-client-nw"
          },
          {
            "name": "messageSeries",
            "in": "query",
            "description": "Specific message series to select messages from",
```

```
            "required": false,
            "type": "array",
            "items": {
              "type": "string"
            },
            "collectionFormat": "multi",
            "x-example": "dma-nw"
          },
          {
            "name": "publication",
            "in": "query",
            "description": "The IDs of the publications to select message from",
            "required": false,
            "type": "array",
            "items": {
              "type": "string"
            },
            "collectionFormat": "multi"
          },
          {
            "name": "areaId",
            "in": "query",
            "description": "The IDs of the areas to select messages from",
            "required": false,
            "type": "array",
            "items": {
              "type": "string"
            },
            "collectionFormat": "multi",
            "x-example": "urn:mrn:iho:country:dk"
          },
          {
            "name": "mainType",
            "in": "query",
            "description": "Either NW (navigational warnings) or NM (notices to mariners)",
            "required": false,
            "type": "array",
            "items": {
              "type": "string",
              "enum": [
                "NW",
                "NM"
              ]
            },
            "collectionFormat": "multi",
            "x-example": "NW"
          },
          {
            "name": "wkt",
            "in": "query",
            "description": "Well-Known Text for geographical extent",
            "required": false,
            "type": "string",
            "x-example": "POLYGON((7 54, 7 57, 13 56, 13 57, 7 54))"
          },
          {
            "name": "externalize",
            "in": "query",
            "description": "Whether to rewrite all embedded links and paths to be absolute
URL's",
            "required": false,
            "type": "boolean",
            "default": true,
            "x-example": true
```

```
          },
          {
            "name": "dateFormat",
            "in": "query",
            "description": "The date format to use for JSON date-time encoding. Either
'UNIX_EPOCH' or 'ISO_8601'",
            "required": false,
            "type": "string",
            "default": "UNIX_EPOCH",
            "x-example": "UNIX_EPOCH",
            "enum": [
              "UNIX_EPOCH",
              "ISO_8601"
            ]
          }
        ],
        "responses": {
          "200": {
            "description": "successful operation",
            "schema": {
              "type": "array",
              "items": {
                "$ref": "#/definitions/Message"
              }
            }
          }
        }
      }
    },
    "definitions": {
      "MultiPoint": {
        "allOf": [
          {
            "$ref": "#/definitions/GeoJson"
          },
          {
            "type": "object",
            "properties": {
              "coordinates": {
                "type": "array",
                "items": {
                  "type": "array",
                  "items": {
                    "type": "number",
                    "format": "double"
                  }
                }
              }
            },
            "description": "GeoJson MultiPoint type",
            "xml": {
              "name": "multiPoint"
            }
          }
        ]
      },
      "MultiLineString": {
        "allOf": [
          {
            "$ref": "#/definitions/GeoJson"
          },
          {
            "type": "object",
```

```
        "properties": {
          "coordinates": {
            "type": "array",
            "items": {
              "type": "array",
              "items": {
                "type": "array",
                "items": {
                  "type": "number",
                  "format": "double"
                }
              }
            }
          }
        },
        "description": "GeoJson MultiLineString type",
        "xml": {
          "name": "multiLineString"
        }
      }
    ]
  },
  "MessageSeries": {
    "type": "object",
    "properties": {
      "seriesId": {
        "type": "string"
      },
      "mainType": {
        "type": "string",
        "enum": [
          "NW",
          "NM"
        ]
      }
    },
    "description": "A message series",
    "xml": {
      "name": "messageSeries"
    }
  },
  "AttachmentDesc": {
    "type": "object",
    "properties": {
      "lang": {
        "type": "string",
        "xml": {
          "attribute": true
        }
      },
      "caption": {
        "type": "string"
      }
    },
    "description": "Translatable fields of the Attachment model"
  },
  "Category": {
    "type": "object",
    "properties": {
      "id": {
        "type": "integer",
        "format": "int32",
        "xml": {
          "attribute": true
```

```
        }
      },
      "mrn": {
        "type": "string"
      },
      "active": {
        "type": "boolean",
        "default": false
      },
      "parent": {
        "$ref": "#/definitions/Category"
      },
      "descs": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/CategoryDesc"
        }
      }
    },
    "description": "Hierarchical category model",
    "xml": {
      "name": "category"
    }
  },
  "Message": {
    "type": "object",
    "properties": {
      "id": {
        "type": "string",
        "xml": {
          "attribute": true
        }
      },
      "created": {
        "type": "string",
        "format": "date-time",
        "xml": {
          "attribute": true
        }
      },
      "updated": {
        "type": "string",
        "format": "date-time",
        "xml": {
          "attribute": true
        }
      },
      "messageSeries": {
        "$ref": "#/definitions/MessageSeries"
      },
      "number": {
        "type": "integer",
        "format": "int32"
      },
      "shortId": {
        "type": "string"
      },
      "mainType": {
        "type": "string",
        "enum": [
          "NW",
          "NM"
        ]
      },
```

```
        "type": {
          "type": "string",
          "enum": [
            "PERMANENT_NOTICE",
            "TEMPORARY_NOTICE",
            "PRELIMINARY_NOTICE",
            "MISCELLANEOUS_NOTICE",
            "COASTAL_WARNING",
            "SUBAREA_WARNING",
            "NAVAREA_WARNING",
            "LOCAL_WARNING"
          ]
        },
        "status": {
          "type": "string",
          "enum": [
            "DRAFT",
            "VERIFIED",
            "PUBLISHED",
            "EXPIRED",
            "CANCELLED",
            "DELETED"
          ]
        },
        "areas": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/Area"
          }
        },
        "categories": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/Category"
          }
        },
        "charts": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/Chart"
          }
        },
        "horizontalDatum": {
          "type": "string"
        },
        "publishDateFrom": {
          "type": "string",
          "format": "date-time"
        },
        "publishDateTo": {
          "type": "string",
          "format": "date-time"
        },
        "followUpDate": {
          "type": "string",
          "format": "date-time"
        },
        "references": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/Reference"
          }
        },
        "originalInformation": {
```

```
        "type": "boolean",
        "default": false
      },
      "parts": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/MessagePart"
        }
      },
      "descs": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/MessageDesc"
        }
      },
      "attachments": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/Attachment"
        }
      }
    },
    "description": "Main NW and NM message class",
    "xml": {
      "name": "message"
    }
  },
  "Crs": {
    "type": "object",
    "properties": {
      "type": {
        "type": "string"
      },
      "properties": {
        "type": "object",
        "additionalProperties": {
          "type": "string"
        }
      }
    },
    "description": "GeoJson Coordinate Reference System"
  },
  "Attachment": {
    "type": "object",
    "properties": {
      "type": {
        "type": "string"
      },
      "path": {
        "type": "string"
      },
      "fileName": {
        "type": "string"
      },
      "fileSize": {
        "type": "integer",
        "format": "int64"
      },
      "display": {
        "type": "string",
        "enum": [
          "ABOVE",
          "BELOW",
          "SEPARATE_PAGE"
```

```
        ]
      },
      "width": {
        "type": "string"
      },
      "height": {
        "type": "string"
      },
      "descs": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/AttachmentDesc"
        }
      }
    },
    "description": "Message attachment model",
    "xml": {
      "name": "attachment"
    }
  },
  "FeatureCollectionVo": {
    "allOf": [
      {
        "$ref": "#/definitions/FeatureCollection"
      },
      {
        "type": "object",
        "properties": {},
        "description": "GeoJson FeatureCollection type",
        "xml": {
          "name": "featureCollection"
        }
      }
    ]
  },
  "GeoJson": {
    "type": "object",
    "discriminator": "type",
    "properties": {
      "type": {
        "type": "string"
      },
      "crs": {
        "$ref": "#/definitions/Crs"
      },
      "bbox": {
        "type": "array",
        "items": {
          "type": "number",
          "format": "double"
        }
      }
    },
    "description": "Superclass for GeoJson types"
  },
  "DateInterval": {
    "type": "object",
    "properties": {
      "allDay": {
        "type": "boolean",
        "default": false
      },
      "fromDate": {
        "type": "string",
```

```
          "format": "date-time"
        },
        "toDate": {
          "type": "string",
          "format": "date-time"
        }
      },
      "description": "Date interval"
    },
    "Feature": {
      "type": "object",
      "discriminator": "type",
      "properties": {
        "type": {
          "type": "string"
        },
        "crs": {
          "$ref": "#/definitions/Crs"
        },
        "bbox": {
          "type": "array",
          "items": {
            "type": "number",
            "format": "double"
          }
        },
        "id": {
          "type": "object"
        },
        "geometry": {
          "$ref": "#/definitions/GeoJson"
        },
        "properties": {
          "type": "object",
          "additionalProperties": {
            "type": "object"
          }
        }
      },
      "description": "GeoJson Feature type",
      "xml": {
        "name": "feature"
      }
    },
    "FeatureVo": {
      "allOf": [
        {
          "$ref": "#/definitions/Feature"
        },
        {
          "type": "object",
          "properties": {},
          "description": "GeoJson Feature type",
          "xml": {
            "name": "feature"
          }
        }
      ]
    },
    "MessageDesc": {
      "type": "object",
      "properties": {
        "lang": {
          "type": "string",
```

```
        "xml": {
          "attribute": true
        }
      },
      "title": {
        "type": "string"
      },
      "vicinity": {
        "type": "string"
      },
      "publication": {
        "type": "string"
      },
      "internalPublication": {
        "type": "string"
      },
      "source": {
        "type": "string"
      }
    },
    "description": "Translatable fields of the Message model"
  },
  "GeometryCollection": {
    "allOf": [
      {
        "$ref": "#/definitions/GeoJson"
      },
      {
        "type": "object",
        "properties": {
          "geometries": {
            "type": "array",
            "items": {
              "$ref": "#/definitions/GeoJson"
            }
          }
        },
        "description": "GeoJson GeometryCollection type",
        "xml": {
          "name": "geometryCollection"
        }
      }
    ]
  },
  "ReferenceDesc": {
    "type": "object",
    "properties": {
      "lang": {
        "type": "string",
        "xml": {
          "attribute": true
        }
      },
      "description": {
        "type": "string"
      }
    },
    "description": "Translatable fields of the Reference model"
  },
  "Polygon": {
    "allOf": [
      {
        "$ref": "#/definitions/GeoJson"
      },
```

```
        {
          "type": "object",
          "properties": {
            "coordinates": {
              "type": "array",
              "items": {
                "type": "array",
                "items": {
                  "type": "array",
                  "items": {
                    "type": "number",
                    "format": "double"
                  }
                }
              }
            }
          },
          "description": "GeoJson Polygon type",
          "xml": {
            "name": "polygon"
          }
        }
      ]
    },
    "AreaDesc": {
      "type": "object",
      "properties": {
        "lang": {
          "type": "string",
          "xml": {
            "attribute": true
          }
        },
        "name": {
          "type": "string"
        }
      },
      "description": "Translatable fields of the Area model"
    },
    "Chart": {
      "type": "object",
      "properties": {
        "chartNumber": {
          "type": "string"
        },
        "internationalNumber": {
          "type": "integer",
          "format": "int32"
        },
        "active": {
          "type": "boolean",
          "default": false
        },
        "scale": {
          "type": "integer",
          "format": "int32"
        },
        "name": {
          "type": "string"
        }
      },
      "description": "Sea chart model",
      "xml": {
        "name": "chart"
```

```
    }
  },
  "CategoryDesc": {
    "type": "object",
    "properties": {
      "lang": {
        "type": "string",
        "xml": {
          "attribute": true
        }
      },
      "name": {
        "type": "string"
      }
    },
    "description": "Translatable fields of the Category model"
  },
  "Reference": {
    "type": "object",
    "properties": {
      "messageId": {
        "type": "string"
      },
      "type": {
        "type": "string",
        "enum": [
          "REFERENCE",
          "REPETITION",
          "REPETITION_NEW_TIME",
          "CANCELLATION",
          "UPDATE"
        ]
      },
      "descs": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/ReferenceDesc"
        }
      }
    },
    "description": "A message reference"
  },
  "MessagePart": {
    "type": "object",
    "properties": {
      "indexNo": {
        "type": "integer",
        "format": "int32"
      },
      "type": {
        "type": "string",
        "enum": [
          "DETAILS",
          "TIME",
          "POSITIONS",
          "NOTE",
          "PROHIBITION",
          "SIGNALS"
        ]
      },
      "eventDates": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/DateInterval"
```

```
        }
      },
      "geometry": {
        "$ref": "#/definitions/FeatureCollection"
      },
      "descs": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/MessagePartDesc"
        }
      },
      "hideSubject": {
        "type": "boolean",
        "default": false
      }
    },
    "description": "Main NW and NM message part class"
  },
  "MessagePartDesc": {
    "type": "object",
    "properties": {
      "lang": {
        "type": "string",
        "xml": {
          "attribute": true
        }
      },
      "subject": {
        "type": "string"
      },
      "details": {
        "type": "string"
      }
    },
    "description": "Translatable fields of the MessagePart model"
  },
  "Point": {
    "allOf": [
      {
        "$ref": "#/definitions/GeoJson"
      },
      {
        "type": "object",
        "properties": {
          "coordinates": {
            "type": "array",
            "items": {
              "type": "number",
              "format": "double"
            }
          }
        },
        "description": "GeoJson Point type",
        "xml": {
          "name": "point"
        }
      }
    ]
  },
  "Area": {
    "type": "object",
    "properties": {
      "id": {
        "type": "integer",
```

```
          "format": "int32",
          "xml": {
            "attribute": true
          }
        },
        "mrn": {
          "type": "string"
        },
        "active": {
          "type": "boolean",
          "default": false
        },
        "parent": {
          "$ref": "#/definitions/Area"
        },
        "descs": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/AreaDesc"
          }
        }
      },
      "description": "Hierarchical area model",
      "xml": {
        "name": "area"
      }
    },
    "FeatureCollection": {
      "type": "object",
      "discriminator": "type",
      "properties": {
        "type": {
          "type": "string"
        },
        "crs": {
          "$ref": "#/definitions/Crs"
        },
        "bbox": {
          "type": "array",
          "items": {
            "type": "number",
            "format": "double"
          }
        },
        "id": {
          "type": "object"
        },
        "features": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/Feature"
          }
        }
      },
      "description": "GeoJson FeatureCollection type",
      "xml": {
        "name": "featureCollection"
      }
    },
    "LineString": {
      "allOf": [
        {
          "$ref": "#/definitions/GeoJson"
        },
```

```
        {
          "type": "object",
          "properties": {
            "coordinates": {
              "type": "array",
              "items": {
                "type": "array",
                "items": {
                  "type": "number",
                  "format": "double"
                }
              }
            }
          },
          "description": "GeoJson LineString type",
          "xml": {
            "name": "lineString"
          }
        }
      ]
    },
    "MultiPolygon": {
      "allOf": [
        {
          "$ref": "#/definitions/GeoJson"
        },
        {
          "type": "object",
          "properties": {
            "coordinates": {
              "type": "array",
              "items": {
                "type": "array",
                "items": {
                  "type": "array",
                  "items": {
                    "type": "array",
                    "items": {
                      "type": "number",
                      "format": "double"
                    }
                  }
                }
              }
            }
          },
          "description": "GeoJson MultiPolygon type",
          "xml": {
            "name": "multiPolygon"
          }
        }
      ]
    }
  }
}
```

NW-NM DMA Service Instance v0.4

Contents

1 Introduction

1.1 Purpose of the document

This document covers a DMA instance of the REST-based technical design [3] of the MW-NM service specification [1], according to the guidelines given in the Service Description Guidelines [2].

1.2 Intended readership

This service instance document is intended to be read by service architects, system engineers and developers in charge of designing and developing client services consuming NW-NM messages from the Danish Maritime Waters.

2 Service Instance Identification

Name	DMA NW-NM T&P Maritime Cloud REST Service
ID	urn:mrn:mcl:service:instance:dma:nw-nm
Version	0.4
Technology	REST
Service Specification ID	urn:mrn:mcl:service:specification:dma:nw-nm
Service Specification Version	0.4
Service Design ID	urn:mrn:mcl:service:dma:nw-nm:rest
Service Design Version	0.4
Description	A DMA instance of the NW-NM REST Service.
Keywords	NW, NM, Navigational Warnings, Notices to Mariners, MSI, Maritime Cloud Service. REST. Danish Maritime Authority, DMA.
Supplier	e-Navigation Team Danish Maritime Authority Carl Jacobsens Vej 31 DK-2500 København K Telephone: +45 40 72 61 08 Email: mfs@dma.dk
Status	Released.

3 Service Implementation and Instantiation Details

This chapter will define the absolute URL of DMA's NW-NM REST service.

3.1 NW-NM REST Service URL

The DMA NW-NM REST service is located at:

- http://niord-dma.e-navigation.net/rest

3.2 Service Endpoint /public/v1/messages

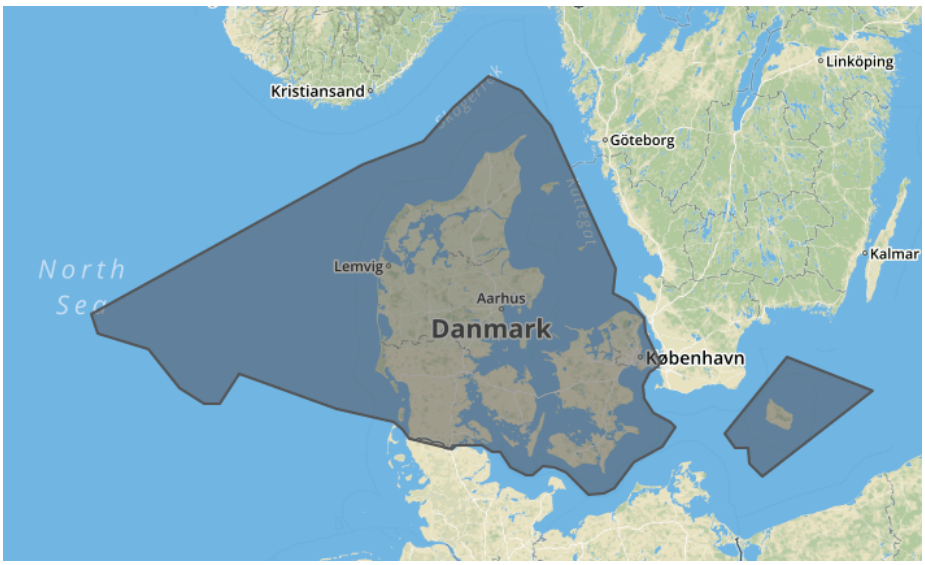
The NW-NM REST Service Specification only specifies a single service interface and operation (i.e. REST endpoint via HTTP GET): /public/v1/messages

The full URL of the service endpoint thus becomes:

- http://niord-dma.e-navigation.net/rest/public/v1/messages

4 Coverage Area

The coverage area of DMA's NW-NM service is the Danish Maritime Waters, which is approximately given by:



The actual WKT definition is part of the service instance XML in Appendix A.

5 Service Level

Availability is not guaranteed.

6 Commercial Information

DMA’s NW-NM REST service is publicly available and can be used for free.

7 References

Nr.	Reference
[1] NW-NM Service Specification	E2 - NW-NM Service Specification version 0.1
[2] Service Description Guidelines	E2_Deliverable D3.4 – Service Specification Template, version 1.0
[3] NW-NM REST Service Technical Design.	E2 - NW-NM REST Service Technical Design version 0.1

8 Acronyms and Terminology

8.1 Acronyms

Term	Definition
API	Application Programming Interface
DMA	Danish Maritime Authority
MC	Maritime Cloud
MEP	Message Exchange Pattern
MRN	Maritime Resource Name
NM	Notices to Mariners
NW	Navigational Warning
REST	Representational State Transfer
WKT	Well-Known Text
XML	Extendible Mark-up Language
XSD	XML Schema Definition

8.2 Terminology

Term	Definition
Docker	Docker containers wrap up a piece of software in a complete filesystem that contains everything it needs to run: code, runtime, system tools, system libraries.
Navigational Warnings	Navigational Warnings (NW) are part of the Maritime Safety Information (MSI) system. Currently, NW's are promulgated in text via SafetyNET, NAVTEX, and is in some countries accessible on the WWW or as voice broadcasts via coastal radio stations.
Keycloak	JBoss Keycloak is an integrated single-sign-on and identity management system for browser apps and RESTful web services.
Niord	Anglified name of the Norse God Njord, associated with the sea and seafaring. Also the name of the EfficienSea 2 sub-project implementation of the NW-NM service and authoring system. See http://niord.org
Notices to Mariners	Notices to Mariners (NM) are promulgated weekly in order to keep nautical charts and publications, as far as possible, up to date. Temporary (T) and Preliminary (P) NMs advise mariners of important matters affecting navigational safety, including new hydrographic information (in advance of new editions or chart updates), changes to routing measures and aids to navigation, and other important categories of data. NM T&P's are today promulgated on paper weekly, fortnightly or monthly and are often accessible on the WWW in PDF format. Not all ENCs include T&P information currently.

Appendix A Service Instance XML

This appendix contains the formal XML definition of the service instance.

```
<?xml version="1.0" encoding="UTF-8"?>
<serviceInstance
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xsi:schemaLocation="http://efficiensea2.org/maritime-cloud/service-
registry/v1/ServiceInstanceSchema.xsd ServiceInstanceSchema.xsd "
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:ServiceInstanceSchema="http://efficiensea2.org/maritime-cloud/service-
registry/v1/ServiceInstanceSchema.xsd"
   xmlns:ServiceSpecificationSchema="http://efficiensea2.org/maritime-cloud/service-
registry/v1/ServiceSpecificationSchema.xsd"
   xmlns="http://efficiensea2.org/maritime-cloud/service-registry/v1/ServiceInstanceSchema.xsd">

 <id>urn:mrn:mcl:service:instance:dma:nw-nm</id>
 <version>0.4</version>
 <name>Danish NW and NM T&amp;amp;P Service</name>
 <status>released</status>
 <description> A DMA instance of the NW-NM REST Service. </description>
 <keywords> NW, NM, Navigational Warnings, Notices to Mariners, MSI, Maritime Cloud Service. REST.
Danish Maritime Authority, DMA </keywords>
 <URL>https://niord-dma.e-navigation.net/rest</URL>
 <requiresAuthorization>false</requiresAuthorization>
 <implementsServiceDesign>
   <id>urn:mrn:mcl:service:design:dma:nw-nm-rest</id>
   <version>0.4</version>
 </implementsServiceDesign>
 <offersServiceLevel>
   <availability>0</availability>
   <name>DMA NW-NM Service Level</name>
   <description> No pay => no availability is guaranteed. </description>
 </offersServiceLevel>
 <coversAreas>
  <coversArea>
   <name>Danish Maritime Waters</name>
   <description> Loosely defined region in the western part of the North Atlantic Ocean. </description>
   <geometryAsWKT>MULTIPOLYGON (((9.624023437500002 54.838663612975125, 9.448242187500002
54.84498993218759,
       9.382324218750002 54.807017138462555, 9.206542968750002 54.832336301970344, 8.6572265625
54.90819859298938,
       8.536376953125 54.990221720048936, 8.382568359375002 55.065786886591724, 7.415771484375
55.19768334019969,
       5.778808593749998 55.528630522571916, 5.44921875 55.24781504467555, 5.185546875
55.24155203565252,
       4.757080078125 55.391592107033404, 4.229736328125 55.76421316483771, 3.3837890624999996
55.91227293006361,
       3.2739257812499996 56.09042714399155, 7.8662109375 57.48040333923342, 8.887939453125
57.692405535264584,
       9.404296875 57.99063188288076, 9.99755859375 58.269065573473284, 10.535888671875
58.14751859907358,
       11.041259765625002 57.83305491291088, 12.15087890625 56.5231395643722, 12.10693359375
56.29825315291387,
       12.384338378906248 56.20975914792473, 12.634277343749996 56.058235955596075, 12.664489746093746
56.015272531542365,
       12.656249999999998 55.91996893509676, 12.711181640624998 55.82134464477079, 12.892456054687496
55.64659898563684,
       12.878723144531248 55.60783270038269, 12.716674804687498 55.541064956111, 12.7056884765625
55.48819145580225,
       12.617797851562498 55.41654360858007, 12.6397705078125 55.285372382493534, 12.7935791015625
55.15376626853558,
       13.062744140624998 55.06893234377864, 13.1561279296875 55.01542594056298, 12.930908203124998
54.82917227452137,
       12.7276611328125 54.76267040025496, 12.453002929687498 54.680183097099984, 12.117919921875
54.41573362292809,
       11.942138671874996 54.36455818952146, 11.678466796874998 54.35815677227373, 11.321411132812498
54.56569261911193,
       11.118164062499996 54.62933821655574, 10.925903320312498 54.63569730606386, 10.739135742187498
```

```
54.54339315407256,
       10.623779296874998 54.54339315407256, 10.360107421874998 54.62933821655574, 10.184326171874998
54.77534585936445,
       10.057983398437496 54.77534585936445, 9.876708984374998 54.8386636129751, 9.624023437500002
54.838663612975125)),
       ((14.0020751953125 54.95869417101662, 15.0457763671875 55.6930679264579, 16.5069580078125
55.363502833950776,
       14.633789062500002 54.53383250794428, 14.414062499999998 54.65794628989232, 14.3975830078125
54.81334841741929,
       14.161376953124998 54.81334841741929, 14.0020751953125 54.95869417101662))) </geometryAsWKT>
  </coversArea>
 </coversAreas>
   <producedBy>
      <ServiceSpecificationSchema:id>urn:mrn:mcl:user:dma:mfs</ServiceSpecificationSchema:id>
      <ServiceSpecificationSchema:name>Mads Friis Sørensen</ServiceSpecificationSchema:name>
      <ServiceSpecificationSchema:description> Responsible for producing the DMA NW-NM REST service
</ServiceSpecificationSchema:description>
      <ServiceSpecificationSchema:contactInfo>mfs@dma.dk</ServiceSpecificationSchema:contactInfo>
      <ServiceSpecificationSchema:isCommercial>false</ServiceSpecificationSchema:isCommercial>
   </producedBy>
   <providedBy>
      <ServiceSpecificationSchema:id>urn:mrn:mcl:user:dma:mfs</ServiceSpecificationSchema:id>
      <ServiceSpecificationSchema:name>Mads Friis Sørensen</ServiceSpecificationSchema:name>
      <ServiceSpecificationSchema:description>Responsible for providing the DMA NW-NM REST
service</ServiceSpecificationSchema:description>
      <ServiceSpecificationSchema:contactInfo>mfs@dma.dk</ServiceSpecificationSchema:contactInfo>
      <ServiceSpecificationSchema:isCommercial>false</ServiceSpecificationSchema:isCommercial>
   </providedBy>
</serviceInstance>
```