ENAV20-9.26

Service Specification for the xxx Service

Contents

Table of figures

List of tables

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.

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 dynamic behaviour description*
- *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 particular 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 has to be provided, where the actual 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

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.

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.

2 Service Identification

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

The tables below shall be completed.

Name	*Service Name*
ID	*Unique identity, e.g., in form of an MRN (Maritime Resource Name, see [3])*
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”.*[1] *“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.*

[1] 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.

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.

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 dedicated sub-section below.

Architectural elements applicable for this description are:

- *Service*
- *Nodes*
- *Operational Activities*
- *Information Exchange Requirements*

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.

The table below lists applicable existing requirements for the *XYZ* service.

Table 1: Requirements Tracing

Requirement Id	Requirement Name	Requirement Text	References

The table below defines additional requirements for the *XYZ* service.

Table 2: Requirements Definition

Requirement Id	
Requirement Name	
Requirement Text	
Rationale	
Author	

Requirement Id	
Requirement Name	
Requirement Text	
Rationale	

Author	

3.2 Other Constraints

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.

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 two 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 3: Operational Nodes providing the ***XYZ*** service

Operational Node	Remarks

Table 4: Operational Nodes consuming the ***XYZ*** service

Operational Node	Remarks

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 5: Operational Activities supported by the ***XYZ*** service

Operational Activity	Remarks

4 Service Overview

This chapter 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.

4.1 Service Interfaces

Describe the interfaces of the service including the selected Message Exchange Pattern (MEP) by using an UML diagram[2] *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.

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

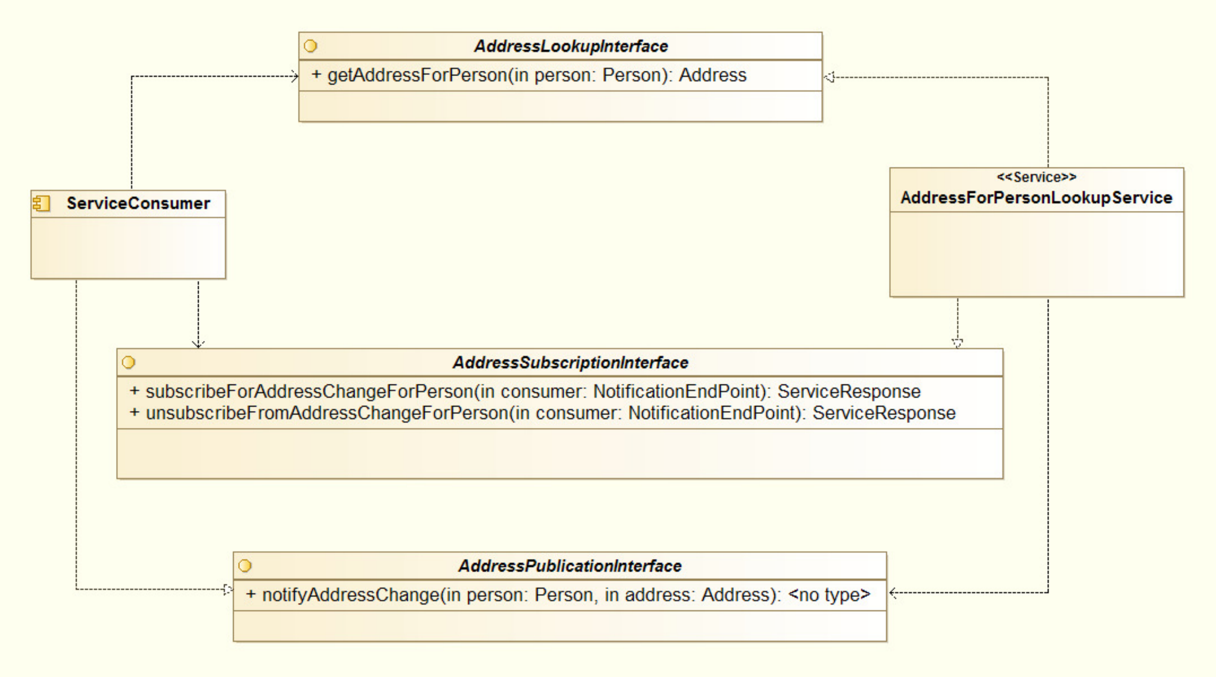

AddressLookupInterface
+ getAddressForPerson(in person: Person): Address
ServiceConsumer
<<Service>>
AddressForPersonLookupService
AddressSubscriptionInterface
+ subscribeForAddressChangeForPerson(in consumer: NotificationEndPoint): ServiceResponse
+ unsubscribeFromAddressChangeForPerson(in consumer: NotificationEndPoint): ServiceResponse
AddressPublicationInterface
+ notifyAddressChange(in person: Person, in address: Address): <no type>

Figure 1: <Service Name> Interface Definition diagram

Table 6: Service Interfaces

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

5 Service Data Model

This section describes 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 **logical** data structure should be shown using diagram(s) and explanatory tables (see below).*

Example of an UML diagram:

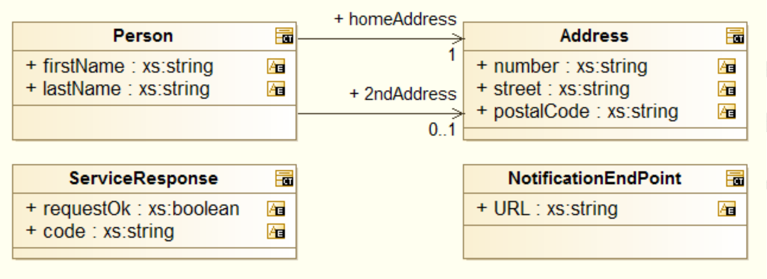


Figure 2: <Service Name> Service Data Model diagram

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*
	Attribute Name	**Type**	**Description**
	…		

An XML schema for this data model is included in the formal service specification xml file attached in Appendix A. Note that the S-100 specification [4] describes in its Appendix 9-B how S-100 based data models shall be formulated in XML schema format.

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 has to be considered just as an example – it is not an authoritative part of the service specification.

6 Service Interface Specifications

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

The Service Interface specification covers only the static design description while the dynamic design (behaviour) is described in chapter 7.

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.

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.

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).

Below is an example of a UML diagram (subset of the service data model, related to one operation):

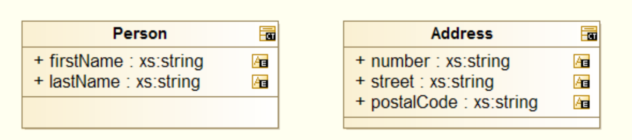


Figure 3: <Service Name> Interface Parameter Definition diagram for <operation name>

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.

Below is an example operation parameter description table.

Table 7: 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.*

6.1.2 Operation <Operation Name>

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

6.2 Service Interface <Interface Name>

Repeat previous section for each interface

7 Service Dynamic Behaviour

This chapter 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*

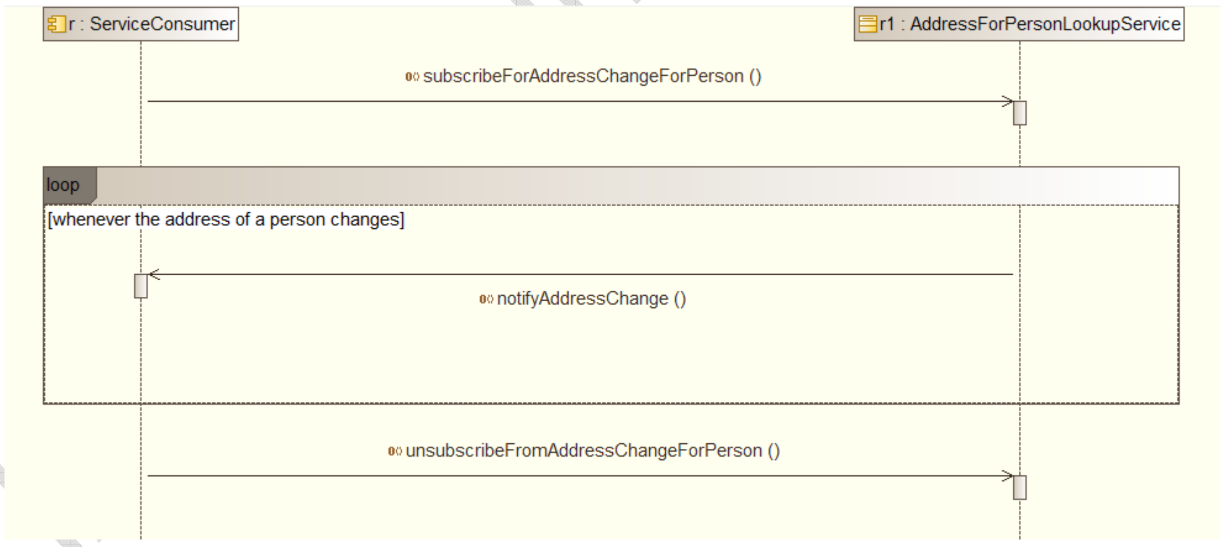
Following types of views and UML diagrams can be used to describe the dynamic behaviour[3]:

- *Sequence diagrams*
- *Interaction diagrams*
- *State machine diagrams*

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.



r : ServiceConsumer
r1 : AddressForPersonLookupService
subscribeForAddressChangeForPerson ()
loop
[whenever the address of a person changes]
notifyAddressChange ()
unsubscribeFromAddressChangeForPerson ()

Figure 4: <Service Name> Operation Sequence Diagram

7.2 Service Interface <Interface Name>

Repeat previous section for each service interface

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

8 Service Provisioning (optional)

This chapter 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.

9 References

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

Nr.	Version	Reference
[1] Service Documentation Guidelines	01.00	SG_Annex_A_Service_Documentation_Guidelines
[3] Maritime Resource Name		Maritime Resource Name, ENAV17-n.n.n
[4] S-100 Universal Hydrographic Data Model	2.0.0	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

10 Acronyms and Terminology

10.1 Acronyms

Term	Definition
API	Application Programming Interface
MC	Maritime Cloud
MEP	Message Exchange Pattern
MRN	Maritime Resource Name
NAF	NATO Architectural Framework
REST	Representational State Transfer
SOA	Service Oriented Architecture
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

10.2 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.
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 in order 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 in order 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 Center, Maritime Authority, Ship, Port, Weather Information

	Provider, …
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.
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. In order 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 is allowed to be registered per technology. The decision, which service instance (out of a number of available spatially exclusive services) shall be registered for a certain geographical region, is a governance issue.

Appendix A 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.