IALA

IALA GUIDELINE

G1128

THE SPECIFICATION OF e-NAVIGATION TECHNICAL SERVICES

Edition 1.1

December 2018

10, rue des Gaudines - 78100 Saint Germain en Laye, France
Tél. +33 (0)1 34 51 70 01- Fax +33 (0)1 34 51 82 05 - contact@iala-aism.org
www.iala-aism.org

International Association of Marine Aids to Navigation and Lighthouse Authorities
Association Internationale de Signalisation Maritime

DOCUMENT HISTORY

Revisions to this IALA document are to be noted in the table prior to the issue of a revised document.

Date	Details	Approval
December 2017 Edition 1.0	1st issue	Council 65
December 2018 Edition 1.1	Update of minor issues, redrawing of figures: • All drawings are remade using Enterprice Architect • Figure 6: Added [0..1] to CoverageArea => geometryAsWKT o This corrects an error in the previous version • Figure 6: Changed URL to Endpoint o This change is in order to embrace services that are not web-services • Table 3: Changed Row "URL" to "Endpoint", slightly adjustment of the description text o Same as above • Annex C (ServiceInstanceSchema.xsd): Changed 2x URL to Endpoint, changed multiplicities of coversAreas, coversArea, unLoCode, geometryAsWKT o This reflects the above changes in the xsd • Introduction: A note on EU General Data Protection Regulation (GDPR)	Council 68

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 e-Navigation Technical Services. A Maritime Service (MS) can be implemented by one or more e-Navigation Technical Services [1].

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 digital service offered by an electronic device to another electronic device. As stated Operational Services are implemented by electronic devices that offer several Technical Services to use the Operational Service.

An e-Navigation Technical Service shall 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 designing 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 shall comply with this Guideline.

NOTE: When personal information (such as name, email address) about citizens in the EU are being stored, then the EU General Data Protection Regulation (GDPR) needs to be followed. More info at www.eugdpr.org.

1.2 Link to S-100 and Product Specifications

This Guideline is intended for, but not limited to, Technical Services based on S-100 Product Specifications. However, since e-Navigation is currently in a transitional phase, an S-100 Product Specification may not always exist for a particular application. In that case, the proposed specification or data model shall make use of the concepts of S-100, whenever possible.

Service descriptions following this Guideline shall provide references to the appropriate S-100 Product Specifications if available, or to the appropriate S-100 features and attributes in their data models.

Similar to the S100, the graphics and textual descriptions may use different data types than the XSD schemas. This is because the XSD technology does not provide these data types, but can be used synonymously.

Table 1 Type Definition

Type Name	Description
CharacterString (String)	A CharacterString is an arbitrary-length sequence of characters including accents and special characters from repertoire of one of the adopted character sets.
Real (float)	A signed real (floating point) number consisting of a mantissa and an exponent, the representation of a real is encapsulation and usage dependent.
Boolean	A value representing binary logic. The value can be either true or false.

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.

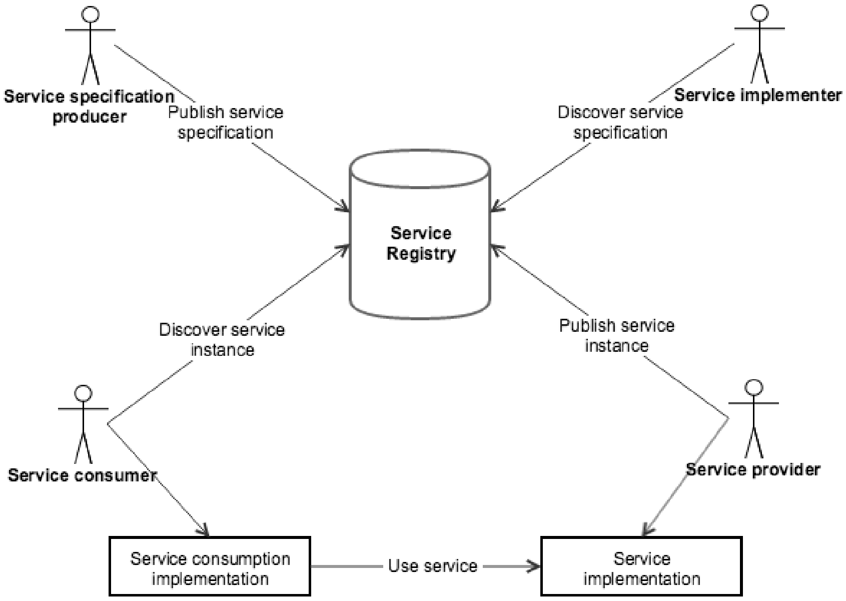
Service specification producer
Publish service specification
Service implementer
Discover service specification
Service Registry
Discover service instance
Publish service instance
Service consumer
Service provider
Service consumption implementation
Use service
Service implementation

Figure 1 Service Management Concept

Figure 2 illustrates 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 the 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.

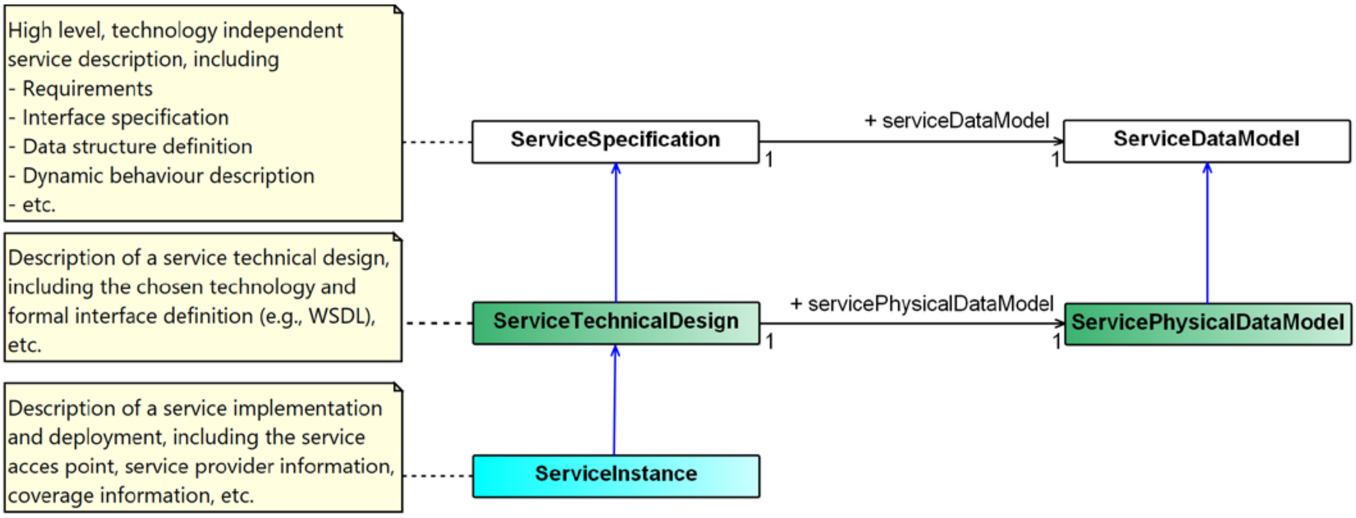
High level, technology independent
service description, including
- Requirements
- Interface specification
- Data structure definition
- Dynamic behaviour description
- etc.
Description of a service technical design, including the chosen technology and formal interface definition (e.g., WSDL), etc.
Description of a service implementation and deployment, including the service acces point, service provider information, coverage information, etc.
ServiceSpecification
+ serviceDataModel
1
1
ServiceDataModel
ServiceTechnicalDesign
+ servicePhysicalDataModel
1
1
ServicePhysicalDataModel
ServiceInstance

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

Figure 3 provides 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).

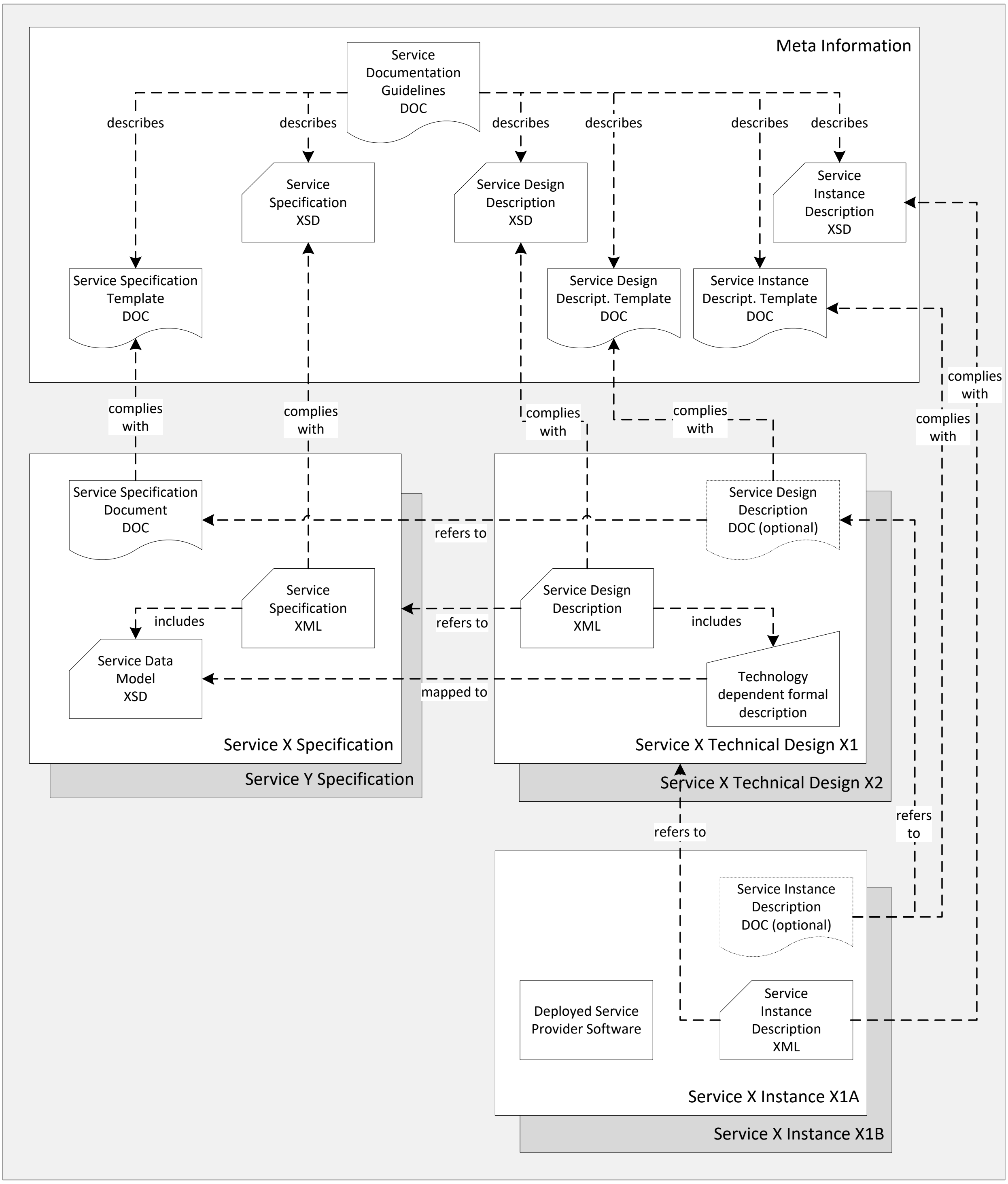
Meta Information
Service Documentation Guidelines DOC
describes
describes
describes
describes
describes
describes
Service Specification XSD
Service Design Description XSD
Service Instance Description XSD
Service Specification Template DOC
Service Design Descript. Template DOC
Service Instance Descript. Template DOC
complies with
complies with
complies with
complies with
complies with
complies with
Service Specification Document DOC
refers to
Service Design Description DOC (optional)
Service Specification XML
includes
Service Data Model XSD
refers to
Service Design Description XML
includes
Technology dependent formal description
mapped to
Service X Specification
Service Y Specification
Service X Technical Design X1
Service X Technical Design X2
refers to
refers to
Service Instance Description DOC (optional)
Deployed Service Provider Software
Service Instance Description XML
Service X Instance X1A
Service X Instance X1B

Figure 3 Service Documentation Overview

The centre part of Figure 3 contains 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 (endpoint) 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 (endpoint) 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) shall 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. Interested consumers may then look up services in the registry.

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 shall 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. 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 D 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.

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 [3] (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 shall 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 MS [1] they shall 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 shall 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 describes 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) shall be shown using diagram(s) and explanatory tables. A description shall be given to 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

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

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 shall 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 shall be declared as informative only – it is not an authoritative part of the service specification.

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 situations 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.

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 presented 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 describes 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)).

[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.

3.3 Service Specification XSD Structure

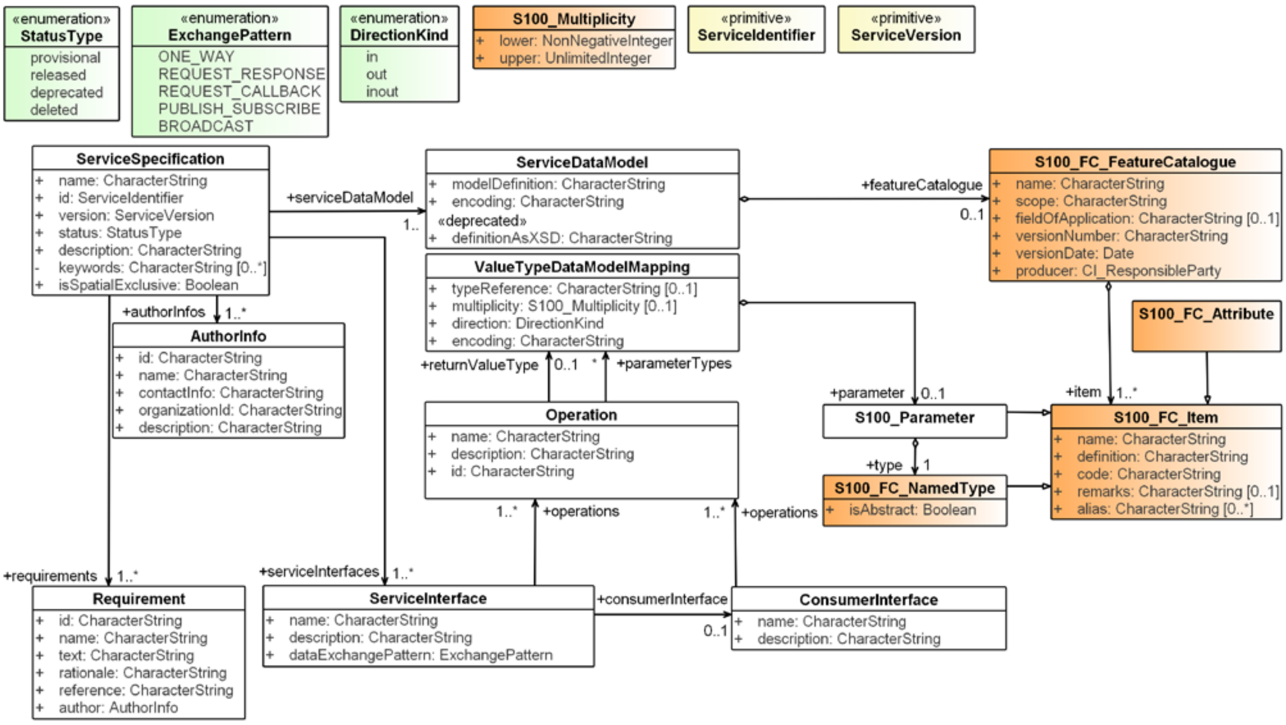

Figure 4 Structure of the Service Specification

Figure 4 gives an overview about the formal description of the service specification. The individual items are described in Table 2.

The service specification XSD file is presented in ANNEX A.

Table 2 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	CharacterString	The human readable service name. The service name shall be at maximum a one-line brief label for the service. Newer versions of the same service specification shall not change the name.
Id	ServiceIdentifier	Service identifier type to be used by service specifications, designs, instances. Currently, the identifier is defined as a string. Globally unique identification of the service. Newer versions of the same service specification shall not change the id.
version	ServiceVersion	Service version indicator type to be used by service specifications, designs, instances. Currently, the version indicator is defined as a 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	StatusType <<enumeration>>	Status of the service specification. The status field has one of the following values: • provisional; • released; • deprecated; • deleted.
description	CharacterString	A human readable short description of the service. The description shall contain an abstract of what a service implementing this specification would do.
keywords	CharacterString	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	Requirements	Refers to requirements specifications for the service. Business requirements, functional and non-functional requirements shall 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**	
Requirements		A requirement that the service shall fulfil.	
	Element Name	**Type**	**Description**
	id	CharacterString	Globally unique requirement identification
	name	CharacterString	Human readable requirement name/summary. Shall not be longer than one line.
	text	CharacterString	The human readable requirement text. Usually formulated in form of a 'shall'-statement.
	rationale	CharacterString	Rationale for this requirement. Textual explanation of why this requirement exists. Provides background information about the need of the service.
	reference	CharacterString	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	CharacterString	Unique identifier of the author.
	name	CharacterString	Human readable name of the author.
	description	CharacterString	Human readable description of the author.
	contactInfo	CharacterString	Human readable contact information of the author.
	organizationId	CharacterString	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	CharacterString	Human readable service interface name. The name shall be no longer than one line.
	description	CharacterString	Human readable description of the service interface.

	dataExchangePattern	ExchangePattern <<enumeration>>	Message exchange pattern for the entire interface. 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 shall 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	CharacterString	Human readable interface name. The name shall be no longer than one line.
	description	CharacterString	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**

<table>
<tr><td></td><td>name</td><td>CharacterString</td><td>Human readable operation name. The name shall be no longer than one line.</td></tr>
<tr><td></td><td>description</td><td>CharacterString</td><td>Human readable description of the operation.</td></tr>
<tr><td></td><td>returnValueType</td><td>ValueTypeData-ModelMapping</td><td>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.</td></tr>
<tr><td></td><td>parameterTypes</td><td>ValueTypeData-ModelMapping</td><td>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.</td></tr>
<tr><th colspan="2">Type Name</th><th colspan="2">Description</th></tr>
<tr><td colspan="2">ValueTypeDataModel-Mapping</td><td colspan="2">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.</td></tr>
<tr><td></td><th>Element Name</th><th>Type</th><th>Description</th></tr>
<tr><td></td><td>typeReference</td><td>CharacterString</td><td>Only used if not S-100 compliant.
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.</td></tr>
<tr><td></td><td>parameter</td><td>S100_Parameter</td><td>Reference to the S-100 attribute.</td></tr>
<tr><td></td><td>multiplicity</td><td>S100_Multiplicity</td><td>Minimum and maximum number of provided instances, where the maximum number may be infinitive. If no multiplicity is provided a multiplicity of 1 is assumed.</td></tr>
<tr><td></td><td>direction</td><td>DirectionKind
<<enumeration&gt;&gt;</td><td>Determining the direction. Can be one of:
• in
• out
• inout</td></tr>
<tr><td></td><td>encoding</td><td>CharacterString</td><td>Encoding of the parameter</td></tr>
<tr><th colspan="2">Type Name</th><th colspan="2">Description</th></tr>
<tr><td colspan="2">ServiceDataModel</td><td colspan="2">If the service is S-100 compliant it shall link to an existing Feature Catalogue in the featureCatalogue element. The other elements are then not used.
If the service is not S-100 compliant: The serviceDataModel 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.</td></tr>
<tr><td></td><th>Element Name</th><th>Type</th><th>Description</th></tr>
<tr><td></td><td>featureCatalogue</td><td>S100_FC_FeatureCatalogue</td><td>Reference to an S-100 compliant logical data model description. To be used to be S-100 compliant, otherwise the modelDefinition element must be used.</td></tr>
</table>

	modelDefinition	CharacterString	Only used if not S-100 compliant. The definition of the service data model described.
	encoding	CharacterString	Only used if not S-100 compliant. 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 E 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.

4.2.1 INTRODUCTION

The introduction section contains 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 shall 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 shall 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.

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

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 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). As a minimum the service specification document needs to be referenced.

4.3 Service Design Description XSD Structure

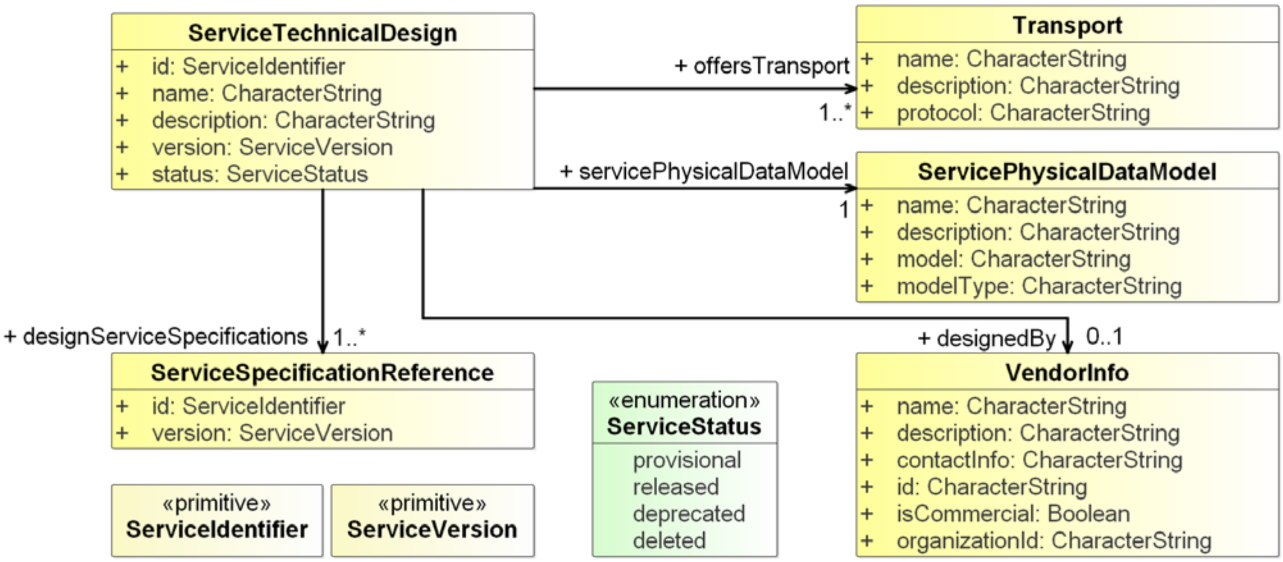

Figure 5 Structure of the Service Technical Design Description

Figure 5 gives an overview about the formal description of the service design description. A description of each item is given in the Table 3.

The service design description schema is presented in ANNEX B.

Table 3 Information Elements of the Service Design Description

Type Name		Description	
ServiceDesign		A service design description.	
	Element Name	Type	Description
	name	CharacterString	The human readable name of the service design. The service name shall be at maximum a one-line brief label. Newer versions of the same service design adopt the same name.
	id	ServiceIdentifier	Service identifier type to be used by service specifications, designs, instances. Currently, the identifier is defined as a string. Globally unique identification of the service design. Newer versions of the same service design shall adopt the same id.
	version	ServiceVersion	Service version indicator type to be used by service specifications, designs, instances. Currently, the version indicator is defined as a 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	ServiceStatus <<enumeration>>	Status of the service design. The status field has one of the following values: ● provisional; ● released; ● deprecated; ● deleted.
	description	CharacterString	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.

Type Name		Description	
	servicePhysicalDataModel	ServicePhysicalDataModel	Mandatory reference to the service physical data model description.

Type Name		Description	
ServiceSpecificationReference		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	ServiceIdentifier	Service identifier type to be used by service specifications, designs, instances. Currently, the identifier is defined as a string. Identification of the service specification realised by the service design.
	version	ServiceVersion	Service version indicator type to be used by service specifications, designs, instances. Currently, the version indicator is defined as a 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	CharacterString	Human readable name.
	description	CharacterString	Human readable description of the transport protocol used by the service design.
	protocol	CharacterString	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	CharacterString	Unique identification of the vendor.
	name	CharacterString	Human readable vendor name. The name shall be no longer than one line.
	description	CharacterString	Human readable description of the vendor.
	contactInfo	CharacterString	Human readable contact information of the vendor.
	organizationId	CharacterString	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	CharacterString	Human readable model name. The name shall be no longer than one line.
	description	CharacterString	Human readable description of the model.
	model	CharacterString	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	CharacterString	The modelType shall contain 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 ANNEX F.

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.

5.2.1 INTRODUCTION

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

5.2.2 SERVICE INSTANCE IDENTIFICATION

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

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.). As a minimum the service specification and service design documents need to be referenced.

5.3 Service Instance Description XSD Structure

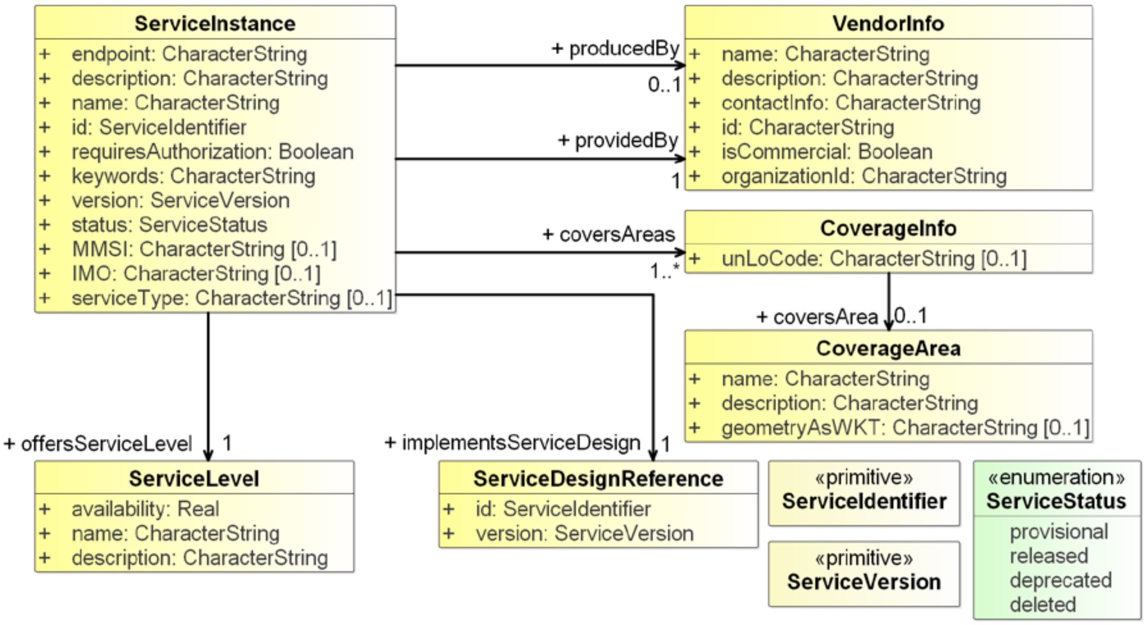
ServiceInstance
+ endpoint: CharacterString
+ description: CharacterString
+ name: CharacterString
+ id: ServiceIdentifier
+ requiresAuthorization: Boolean
+ keywords: CharacterString
+ version: ServiceVersion
+ status: ServiceStatus
+ MMSI: CharacterString [0..1]
+ IMO: CharacterString [0..1]
+ serviceType: CharacterString [0..1]
+ producedBy
0..1
+ providedBy
1
+ coversAreas
1..*
VendorInfo
+ name: CharacterString
+ description: CharacterString
+ contactInfo: CharacterString
+ id: CharacterString
+ isCommercial: Boolean
+ organizationId: CharacterString
CoverageInfo
+ unLoCode: CharacterString [0..1]
+ coversArea
0..1
CoverageArea
+ name: CharacterString
+ description: CharacterString
+ geometryAsWKT: CharacterString [0..1]
+ offersServiceLevel
1
+ implementsServiceDesign
1
ServiceLevel
+ availability: Real
+ name: CharacterString
+ description: CharacterString
ServiceDesignReference
+ id: ServiceIdentifier
+ version: ServiceVersion
«primitive»
ServiceIdentifier
«primitive»
ServiceVersion
«enumeration»
ServiceStatus
provisional
released
deprecated
deleted

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 4.

The service instance description schema is presented in ANNEX C.

Table 4 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	CharacterString	The human readable name of the service instance. The service name shall be at maximum a one-line brief label. Newer versions of the same service specification shall adopt the same name.
id	ServiceIdentifier	Service identifier type to be used by service specifications, designs, instances. Currently, the identifier is defined as a string. Globally unique identification of the service instance. Newer versions of the same service instance shall adopt the same id.
version	ServiceVersion	Service version indicator type to be used by service specifications, designs, instances. Currently, the version indicator is defined as a 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	ServiceStatus <<enumeration>>	Status of the service instance. The status field has one of the following values: • provisional; • released; • deprecated; • deleted.
description	CharacterString	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	CharacterString	A list of keywords associated to the service.
endpoint	CharacterString	Endpoint that describes where the service endpoint is located (e.g. URL)
MMSI	CharacterString	Optional Maritime Mobile Service Identity.
IMO	CharacterString	Optional International Maritime Organization (IMO) number
serviceType	CharacterString	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.

Type Name		Description	
	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	ServiceIdentifier	Service identifier type to be used by service specifications, designs, instances. Currently, the identifier is defined as a string. Service identifier type to be used by service specifications, designs, instances. Currently, the identifier is defined as a string. Identification of the service design implemented by the service instance.
	version	ServiceVersion	Service version indicator type to be used by service specifications, designs, instances. Currently, the version indicator is defined as a string. Service version indicator type to be used by service specifications, designs, instances. 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	CharacterString	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**

Type Name		Description	
	name	CharacterString	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	CharacterString	Human readable description of the coverage area.
	geometryAsWKT	CharacterString	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	CharacterString	Human readable service level name. The name shall be no longer than one line.
	description	CharacterString	Human readable description of the service level.
	availability	Real	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	CharacterString	Unique identification of the vendor.
	name	CharacterString	Human readable vendor name. The name shall be no longer than one line.
	description	CharacterString	Human readable description of the vendor.
	contactInfo	CharacterString	Human readable contact information of the vendor.
	organizationId	CharacterString	Unique identifier of the organization, the author belongs to.
	isCommercial	Boolean	Optional indication on the commercial status of the vendor.

6 GOVERNANCE

A governance model will be needed in the area of service management. This includes questions about the process to decide about:

- maturity of service specifications;
- the scope of service specifications;
- the evolution of service specifications;
- the life cycle of service specifications and service instances;
- 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 shall be considered as the authoritative source of definitions used in IALA documents.

7.1 Terminology

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, …

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 endpoints.

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.

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
MS	Maritime Service
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] Maritime Services (MS), NCSR 5 / WP.4 Draft Report of the Navigation Working Group

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

[3] Maritime Resource Name (mrnregistry.org)

SERVICE SPECIFICATION SCHEMA

ServiceBaseTypesSchema.xsd

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<schema xmlns="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: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"/>
  <!--
   At the time of writing, no official S100 schema was available online. Please use official schema and import
   like shown below.
   -->
  <!-- <import namespace="<S100 namespace>" schemaLocation="<S100.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.
    </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 shall be at maximum a one
          line brief label for the service. Newer versions of the same service
          specification shall 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 shall 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 for the entire interface.
        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.
        - parameter Reference to a S-100 parameter. Used if service is S-100 compliant.
        - multiplicity Minimum and maximum number of provided instances, where the maximum number may be
          infinitive.
          If no multiplicity is provided a multiplicity of 1 is assumed.
        - direction Determining the direction. Can be one of:
          in
          out
          inout
        - encoding Encoding of the parameter
      </documentation>
    </annotation>
    <all>
      <element name="typeReference" type="string" minOccurs="0" maxOccurs="1"/>
      <element name="parameter" type="S100_Parameter" minOccurs="0" maxOccurs="1"/>
      <element name="multiplicity" type="S100_Multiplicity" minOccurs="0" maxOccurs="unbounded"/>
      <element name="direction" minOccurs="0" maxOccurs="1">
        <simpleType>
          <restriction base="string">
            <enumeration value="IN "/>
            <enumeration value="OUT"/>
            <enumeration value="INOUT"/>
          </restriction>
        </simpleType>
      </element>
      <element name="encoding" type="string" minOccurs="0" maxOccurs="1"/>
    </all>
  </complexType>
  <complexType name="ServiceDataModel">
    <annotation>
      <documentation>
        If the service is S-100 compliant it shall link to an existing Feature Catalogue in the
        featureCatalogue element. The other elements are then not used.
        If the service is not S-100 compliant: The serviceDataModel 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:
        - featureCatalogue Reference to an S-100 compliant logical data model description. To be used to
          be S-100 compliant, otherwise the modelDefinition attribute has to be used
        - modelDefinition Only used if not S-100 compliant. The definition of the service data model
          described
        - encoding Only used if not S-100 compliant. The encoding of the attribute modelDefinition
      </documentation>
    </annotation>
    <all>
      <element name="featureCatalogue" type="S100_FC_FeatureCatalogue" minOccurs="0" maxOccurs="1">
      <element name="modelDefinition" type="string" minOccurs="0" maxOccurs="1">
      <element name="encoding" type="string" minOccurs="0" maxOccurs="1">
    </all>
  </complexType>
</schema>
```

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: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.
    </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 shall be at maximum a one-line brief
          label. Newer versions of the same service design shall 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 realized 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=" ServiceSpecificationSchema:ServiceIdentifier " minOccurs="1" maxOccurs="1"/>
      <element name="version" type=" ServiceSpecificationSchema:ServiceVersion" 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 shall contain 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>
```

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: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.
    </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 shall be
          at maximum a one line brief label. Newer versions of the same service
          specification shall 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.

        - endpoint Endpoint that describes where the service endpoint is located (e.g. URL)

        - 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="endpoint" 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="unbounded"
           type="ServiceInstanceSchema:CoverageInfo"/>
      <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=" ServiceSpecificationSchema:ServiceIdentifier " minOccurs="1" maxOccurs="1"/>
      <element name="version" type=" ServiceSpecificationSchema:ServiceVersion " 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="CoverageInfo">
    <choice>
      <element name="coversArea" type="ServiceInstanceSchema:CoverageArea" minOccurs="0"
           maxOccurs="1"/>
      <element name="unLoCode" type="string" minOccurs="0" maxOccurs="1"/>
    </choice>
  </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>
```

SERVICE SPECIFICATION TEMPLATE

Service Specification for the *xxx* Service

D 1 INTRODUCTION

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.*

This service specification document describes just one dedicated service in detail at logical level. In addition, there shall 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 shall be provided, where the realisation of the service with a dedicated technology shall be described.

This section shall 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 SOURCES

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 activities, 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 5 shall be completed.

Table 5 Service identification

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'.[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 shall 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 shall 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,

[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.

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 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 6 lists applicable existing requirements for the *XYZ* service.

Table 6 Requirements Tracing

Requirement Id	Requirement Name	Requirement Text	References

Table 7 defines additional requirements for the *XYZ* service.

Table 7 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 8 Operational Nodes providing the XYZ service

Operational Node	Remarks

Table 9 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 10 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.

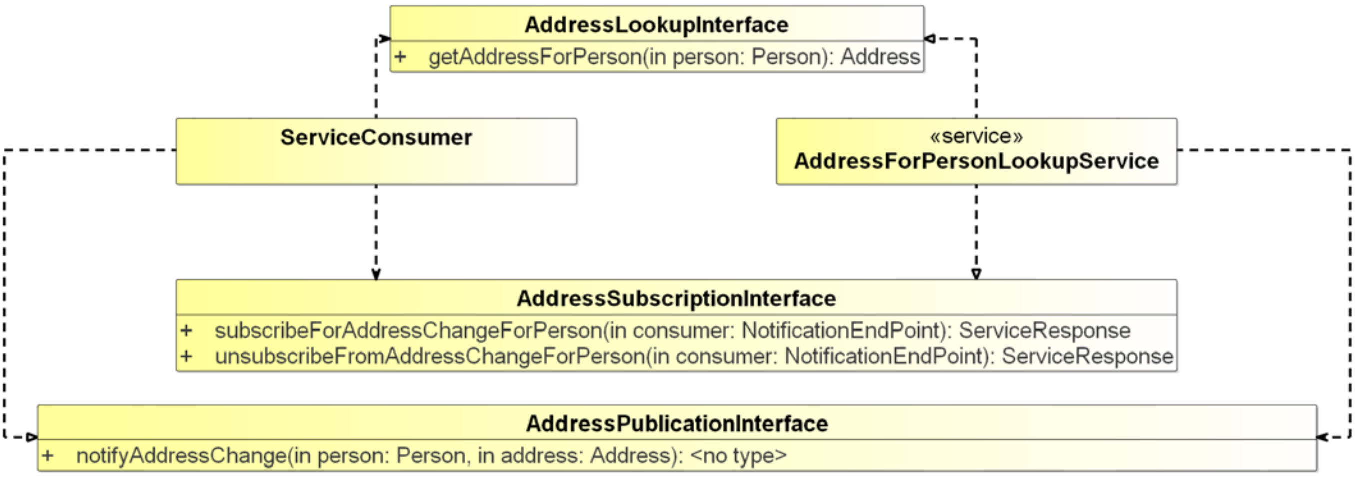
AddressLookupInterface
+ getAddressForPerson(in person: Person): Address
ServiceConsumer
«service»
AddressForPersonLookupService
AddressSubscriptionInterface
+ subscribeForAddressChangeForPerson(in consumer: NotificationEndPoint): ServiceResponse
+ unsubscribeFromAddressChangeForPerson(in consumer: NotificationEndPoint):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

Figure 7 <Service Name> Interface Definition diagram

Table 11 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) shall be shown using diagram(s) and explanatory tables (see below).

Example of an UML diagram:

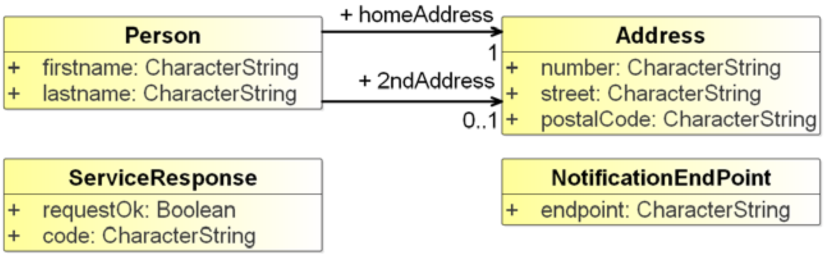
Person
+ firstname: CharacterString
+ lastname: CharacterString
+ homeAddress
1
+ 2ndAddress
0..1
Address
+ number: CharacterString
+ street: CharacterString
+ postalCode: CharacterString
ServiceResponse
+ requestOk: Boolean
+ code: CharacterString
NotificationEndPoint
+ endpoint: CharacterString

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 shall 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.

Table 12 Describing a service data model

<table>
<tr><th colspan="4">Element Name</th><th colspan="2">Description</th></tr>
<tr><td colspan="4">Person</td><td colspan="2">Describe here the ‘Person’ structure.</td></tr>
<tr><td></td><th colspan="2">Attribute Name</th><th colspan="2">Type</th><th>Description</th></tr>
<tr><td></td><td colspan="2">firstName</td><td colspan="2">String</td><td>Description of firstName goes here.</td></tr>
<tr><td colspan="2"></td><th colspan="2">Tracing Information</th><th colspan="2">Value</th></tr>
<tr><td colspan="2"></td><td colspan="2">External model trace</td><td colspan="2">Trace into the logical or physical model for firstName</td></tr>
<tr><td></td><th colspan="2">Attribute Name</th><th colspan="2">Type</th><th>Description</th></tr>
<tr><td></td><td colspan="2">lastName</td><td colspan="2">String</td><td>Description of lastName goes here.</td></tr>
<tr><td colspan="2"></td><th colspan="2">Tracing Information</th><th colspan="2">Value</th></tr>
<tr><td colspan="2"></td><td colspan="2">External model trace</td><td colspan="2">Trace into the logical or physical model for the lastName</td></tr>
<tr><td></td><th colspan="2">Attribute Name</th><th colspan="2">Type</th><th>Description</th></tr>
<tr><td></td><td colspan="2">homeAddress</td><td colspan="2">Address</td><td>The main home address of Person</td></tr>
<tr><td colspan="2"></td><th colspan="2">Tracing Information</th><th colspan="2">Value</th></tr>
<tr><td colspan="2"></td><td colspan="2">External model trace</td><td colspan="2">Trace into the logical or physical model for the homeAddress</td></tr>
<tr><td></td><th colspan="2">Attribute Name</th><th colspan="2">Type</th><th>Description</th></tr>
<tr><td></td><td colspan="2">2ndAddress</td><td colspan="2">Address</td><td>Any second address of Person (optional)</td></tr>
<tr><td colspan="2"></td><th colspan="2">Tracing Information</th><th colspan="2">Value</th></tr>
<tr><td colspan="2"></td><td colspan="2">External model trace</td><td colspan="2">Trace into the logical or physical model for the 2ndAddress</td></tr>
<tr><th colspan="4">Element Name</th><th colspan="2">Description</th></tr>
<tr><td colspan="4">Address</td><td colspan="2">Describe here the Address structure.</td></tr>
<tr><td></td><th colspan="2">Attribute Name</th><th colspan="2">Type</th><th>Description</th></tr>
<tr><td></td><td colspan="2">number</td><td colspan="2">String</td><td>Description of number goes here.</td></tr>
<tr><td colspan="2"></td><th colspan="2">Tracing Information</th><th colspan="2">Value</th></tr>
<tr><td colspan="2"></td><td colspan="2">External model trace</td><td colspan="2">Trace into the logical or physical model for the number attribute</td></tr>
</table>

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 [4] 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 should 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 shall 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.

D 6.1.1.1. Operation Functionality

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

D 6.1.1.2. 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).

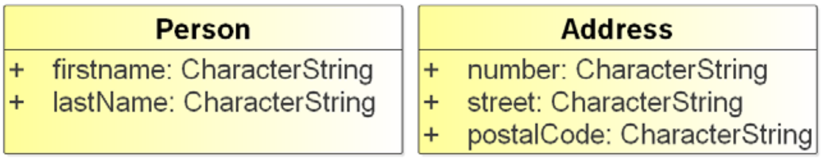
Person
+ firstname: CharacterString
+ lastName: CharacterString
Address
+ number: CharacterString
+ street: CharacterString
+ postalCode: CharacterString

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>

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

An example sequence diagram is shown in Figure 10.

[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.

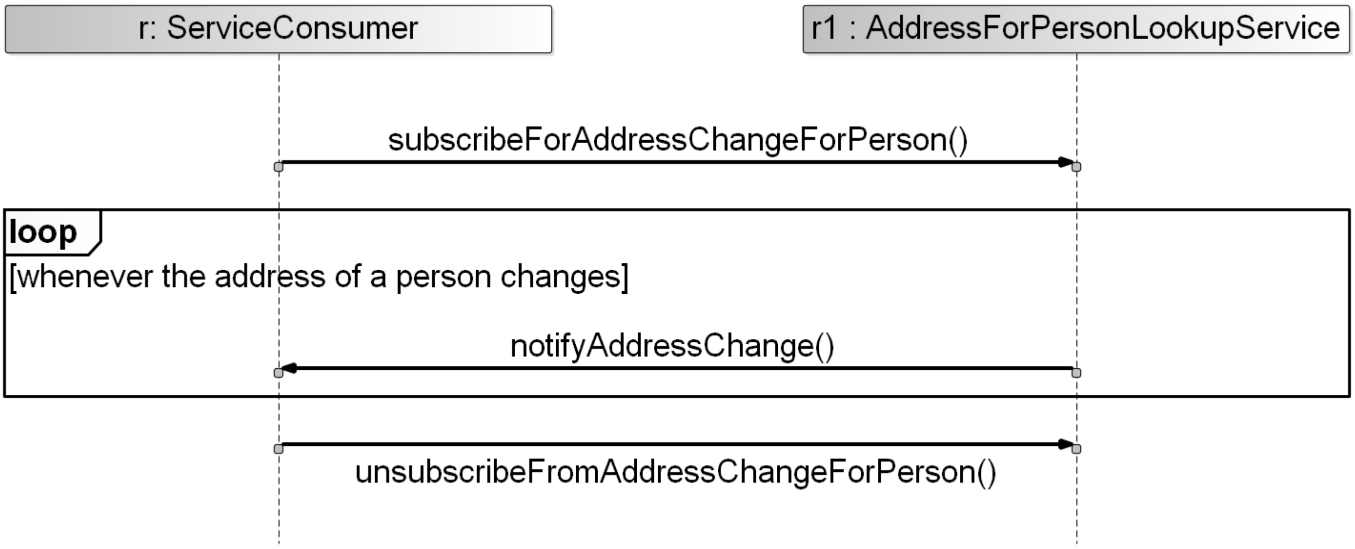
r: ServiceConsumer
r1 : AddressForPersonLookupService
subscribeForAddressChangeForPerson()
loop
[whenever the address of a person changes]
notifyAddressChange()
unsubscribeFromAddressChangeForPerson()

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 shall 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 shall be added here).

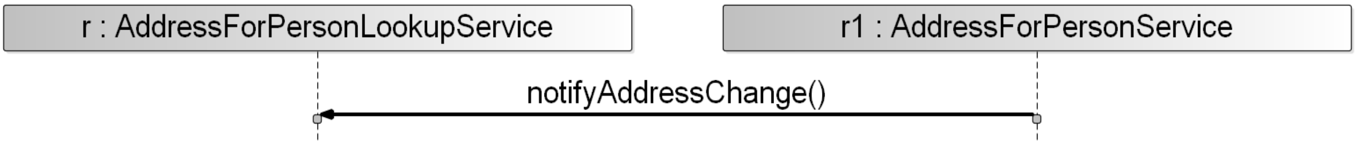
r : AddressForPersonLookupService
r1 : AddressForPersonService
notifyAddressChange()

Figure 11 <Service Name> Orchestration Sequence Diagram

D 8 SERVICE PROVISIONING (OPTIONAL)

This section shall 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 shall be considered as the authoritative source of definitions used in IALA documents.

D 9.1. TERMINOLOGY

Persons producing the Technical Service are invited to add definitions to the following list as appropriate.

Table 13 Definitions of terminology – Technical Service

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.

D 10 ACRONYMS

Persons producing the Technical Service are invited to provide a list of acronyms as appropriate.

ACRONYM	Meaning

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 1128 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.

SERVICE TECHNICAL DESIGN TEMPLATE

Service Design Description for the *xxx* Service

<xyz Technology>

E 1 INTRODUCTION

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 shall 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 Sources

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 activities, 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 14 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.*

[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.

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

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 15.

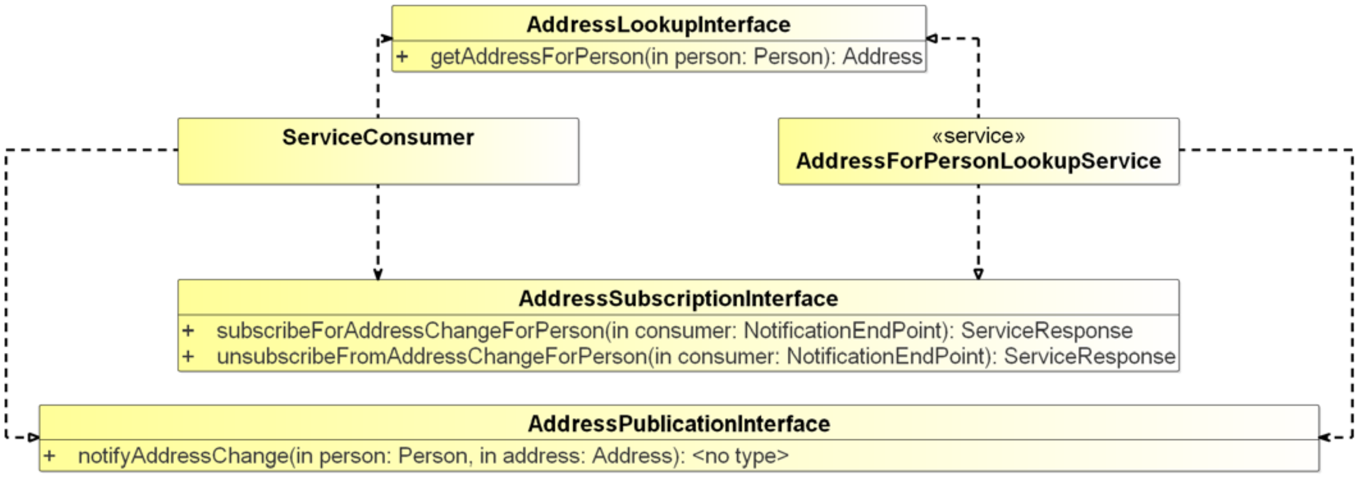
AddressLookupInterface
+ getAddressForPerson(in person: Person): Address
ServiceConsumer
«service»
AddressForPersonLookupService
AddressSubscriptionInterface
+ subscribeForAddressChangeForPerson(in consumer: NotificationEndPoint): ServiceResponse
+ unsubscribeFromAddressChangeForPerson(in consumer: NotificationEndPoint):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

Figure 12 <Service Name> Interface Definition diagram

Table 15 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.

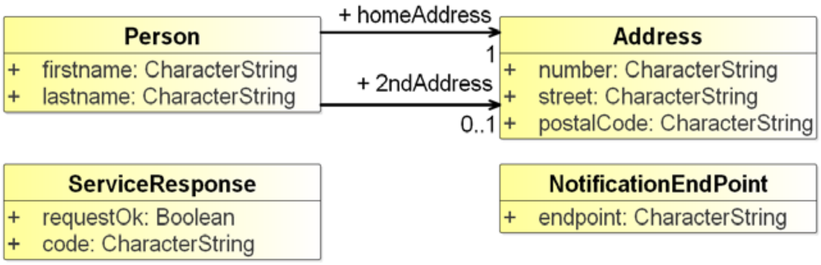
Person
+ firstname: CharacterString
+ lastname: CharacterString
+ homeAddress
1
+ 2ndAddress
0..1
Address
+ number: CharacterString
+ street: CharacterString
+ postalCode: CharacterString
ServiceResponse
+ requestOk: Boolean
+ code: CharacterString
NotificationEndPoint
+ endpoint: CharacterString

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 shall 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 16 is an example for describing a service data model including traces to an external model.

Table 16 Service Data Model Description

<table>
<tr><th colspan="4">Element Name</th><th colspan="2">Description</th></tr>
<tr><td colspan="4">Person</td><td colspan="2">Describe here the ‘Person’ structure.</td></tr>
<tr><td></td><th colspan="2">Attribute Name</th><th colspan="2">Type</th><th>Description</th></tr>
<tr><td></td><td colspan="2">firstName</td><td colspan="2">String</td><td>Description of firstName goes here.</td></tr>
<tr><td colspan="2"></td><th colspan="2">Tracing Information</th><th colspan="2">Value</th></tr>
<tr><td colspan="2"></td><td colspan="2">Spec.data model trace</td><td colspan="2">Trace into the service specification data model for firstName</td></tr>
<tr><td colspan="2"></td><td colspan="2">External model trace</td><td colspan="2">Trace into the external data model for firstName</td></tr>
<tr><td></td><th colspan="2">Attribute Name</th><th colspan="2">Type</th><th>Description</th></tr>
<tr><td></td><td colspan="2">lastName</td><td colspan="2">String</td><td>Description of lastName goes here.</td></tr>
<tr><td colspan="2"></td><th colspan="2">Tracing Information</th><th colspan="2">Value</th></tr>
<tr><td colspan="2"></td><td colspan="2">Spec.data model trace</td><td colspan="2">Trace into the service specification data model for lastName</td></tr>
<tr><td colspan="2"></td><td colspan="2">External model trace</td><td colspan="2">Trace into the logical or physical model for the lastName</td></tr>
<tr><td></td><th colspan="2">Attribute Name</th><th colspan="2">Type</th><th>Description</th></tr>
<tr><td></td><td colspan="2">homeAddress</td><td colspan="2">Address</td><td>The main home address of Person</td></tr>
<tr><td colspan="2"></td><th colspan="2">Tracing Information</th><th colspan="2">Value</th></tr>
<tr><td colspan="2"></td><td colspan="2">Spec. data model trace</td><td colspan="2">Trace into the service specification data model for homeAdd</td></tr>
<tr><td colspan="2"></td><td colspan="2">External model trace</td><td colspan="2">Trace into the logical or physical model for the homeAddress</td></tr>
<tr><td></td><th colspan="2">Attribute Name</th><th colspan="2">Type</th><th>Description</th></tr>
<tr><td></td><td colspan="2">2ndAddress</td><td colspan="2">Address</td><td>Any second address of Person (optional)</td></tr>
<tr><td colspan="2"></td><th colspan="2">Tracing Information</th><th colspan="2">Value</th></tr>
<tr><td colspan="2"></td><td colspan="2">External model trace</td><td colspan="2">Trace into the logical or physical model for the 2ndAddress</td></tr>
<tr><th colspan="4">Element Name</th><th colspan="2">Description</th></tr>
<tr><td colspan="4">Address</td><td colspan="2">Describe here the Address structure.</td></tr>
<tr><td></td><th colspan="2">Attribute Name</th><th colspan="2">Type</th><th>Description</th></tr>
<tr><td></td><td colspan="2">number</td><td colspan="2">String</td><td>Description of number goes here.</td></tr>
<tr><td colspan="2"></td><th colspan="2">Tracing Information</th><th colspan="2">Value</th></tr>
<tr><td colspan="2"></td><td colspan="2">Spec.data model trace</td><td colspan="2">Trace into the service specification data model for the number attribute</td></tr>
<tr><td colspan="2"></td><td colspan="2">External model trace</td><td colspan="2">Trace into the logical or physical model for the number attribute</td></tr>
</table>

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 should 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.

E 6.1.1.1 Operation Functionality

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

E 6.1.1.2 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).

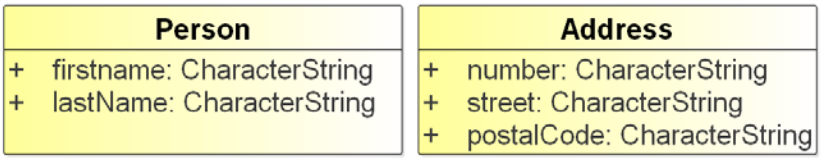
Person
+ firstname: CharacterString
+ lastName: CharacterString
Address
+ number: CharacterString
+ street: CharacterString
+ postalCode: CharacterString

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 17 shows an example operation parameter description table.

Table 17 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 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 shall be exposed on at least one diagram.

An example sequence diagram is given below.

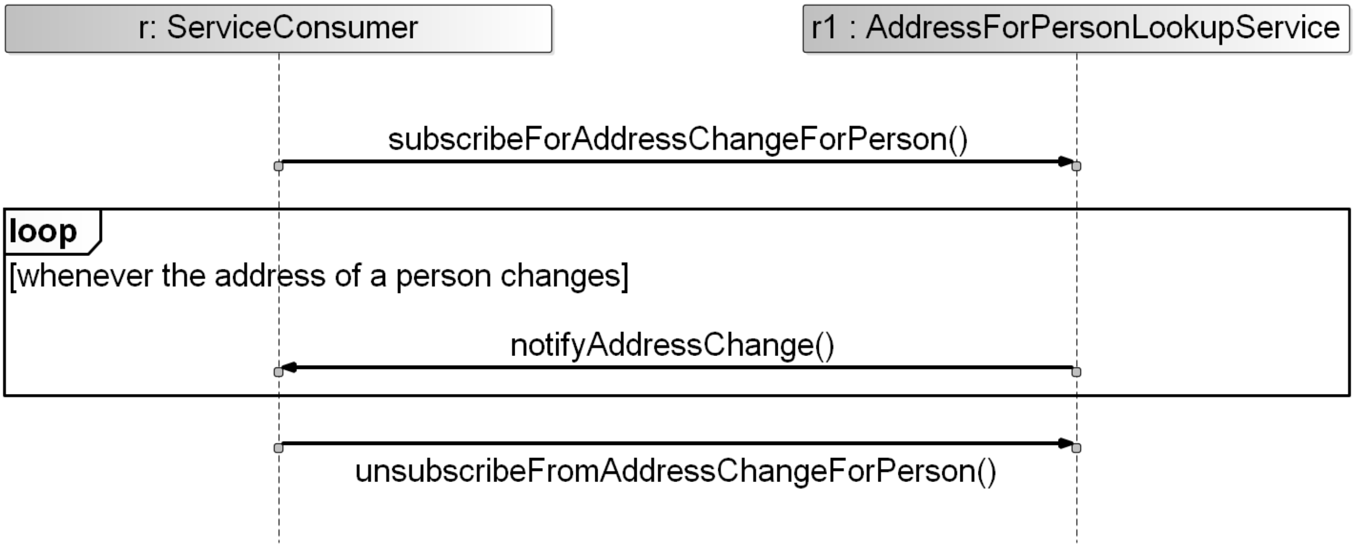
r: ServiceConsumer
r1 : AddressForPersonLookupService
subscribeForAddressChangeForPerson()
loop
[whenever the address of a person changes]
notifyAddressChange()
unsubscribeFromAddressChangeForPerson()

Figure 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 shall be considered as the authoritative source of definitions used in IALA documents.

E 8.1 TERMINOLOGY

[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.*

Persons producing the Technical Service are invited to add definitions to the following list as appropriate.

Table 18 Definition of terminology – Technical Service

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.

E 9 ACRONYMS

Persons producing the Technical Service are invited to provide a list of acronyms as appropriate.

ACRONYM Meaning

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 Guideline 1128 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 *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 shall 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 19 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 shall be considered as the authoritative source of definitions used in IALA documents.

F 5.1 TERMINOLOGY

Persons producing the Technical Service are invited to add definitions to the following list as appropriate.

Table 20 Definition of terminology – Technical Service

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

Persons producing the Technical Service are invited to provide a list of acronyms as appropriate.

ACRONYM	Meaning

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 1128 on Specification of e-Navigation Technical Services*

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

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