|  |
| --- |
| IALA Guideline |



G1128

The Specification of e-Navigation Technical Services

Edition 1.5

Xxx 2023

urn:mrn:iala:pub:g1128:ed1.5

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

| Date | Details | Approval |
| --- | --- | --- |
| December 2017 | First edition | 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   * + This corrects an error in the previous version   Figure 6: Changed URL to Endpoint   * + 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   * + Same as above   Annex C (ServiceInstanceSchema.xsd): Changed 2x URL to Endpoint, changed multiplicities of coversAreas, coversArea, unLoCode, geometryAsWKT   * + This reflects the above changes in the xsd   Introduction: A note on EU General Data Protection Regulation (GDPR). | Council 68 |
| June 2021 | Edition 1.2 Editorial corrections |  |
| December 2021 | Update of minor issues;  • Section 2: Added a description of the different stakeholders in Figure 1, moved explicit mentions of the MCP to informative references.  • Section 3: Introduced MRNs for Service Identifiers and Organization IDs, fixed a non-existent reference to S100\_Parameter by referencing S100\_FC\_NamedType  • Section 4: Introduced MRNs for Service Identifiers and Organization IDs  • Section 5: Extended the list of possible contents for the “Service Implementation and Instance Details” to composition of services and internal service structure, introduced MRNs for Service Identifiers and Organization IDs, added guidance for using keywords, defined the format of IMO and MMSI numbers in the instance description, changed the serviceType attribute to an open enumeration (including Maritime Services) and changed the multiplicity to 1..\*, revision of incorrect definitions.  • XSD-Schemas: Fixed the reference to non-existent element “S100\_Parameter”, correction of multiplicity of the “coversAreas” in the service instance schema, changed the namespace of the xsd-Files to http://ialaaism.org/g1128/v1.3/ServiceSpecificationSchema.xsd, correction of wrong reference to S100\_FC\_FeatureCatalogue | Council 74 |
| July 2022 | Edition 1.4 Editiorial corrections |  |
| Xxx 2023 | Restructuring of what information is actually contained in which specification level.   * Removed interfaces from service specification as interfaces cannot be technology agnostic |  |

1. INTRODUCTION 7

1.1. Purpose of the Document 7

1.2. Link to S-100 and Product Specifications 7

2. OVERVIEW 8

2.1. Service Management Overview 8

2.2. Process Considerations 13

2.2.1. Top-Down Service Development 13

2.2.2. Bottom-Up Service Documentation 13

3. SERVICE SPECIFICATION 14

3.1. Service Specification Document 15

3.2. Service Specification Template 15

3.2.1. Introduction 15

3.2.2. Service Identification 15

3.2.3. Operational Context 15

3.2.4. Service Overview 16

3.2.5. Service Data Model 16

3.2.6. Service Interface Specification 17

3.2.7. Service Dynamic Behaviour 18

3.2.8. Service Provisioning (optional) 18

3.2.9. References 18

3.3. Service Specification XSD Structure 19

4. SERVICE TECHNICAL DESIGN 24

4.1. Service Design Description Document 24

4.2. Service Design Description Template 24

4.2.1. Introduction 24

4.2.2. Service Design Identification 24

4.2.3. Technology Introduction 25

4.2.4. Service Design Overview 25

4.2.5. Physical Data Model 25

4.2.6. Service Interface Design 26

4.2.7. Service Dynamic Behaviour 26

4.2.8. References 26

4.3. Service Design Description XSD Structure 27

5. SERVICE INSTANCE 30

5.1. Service Instance Description Document 30

5.2. Service Instance Description Template 30

5.2.1. Introduction 30

5.2.2. Service Instance Identification 30

5.2.3. Service Implementation and Instance Details 30

5.2.4. Release Notes 30

5.2.5. References 31

5.3. Service Instance Description XSD Structure 32

6. GOVERNANCE 36

7. DEFINITIONS 36

8. Abbreviations 39

9. REFERENCES 39

ANNEX A SERVICE SPECIFICATION SCHEMA 40

ANNEX B SERVICE DESIGN DESCRIPTION SCHEMA 50

ANNEX C SERVICE INSTANCE DESCRIPTION SCHEMA 53

ANNEX D SERVICE SPECIFICATION TEMPLATE 57

APPENDIX 1 SERVICE SPECIFICATION XML 71

ANNEX E SERVICE TECHNICAL DESIGN TEMPLATE 72

APPENDIX 1 SERVICE DESIGN DESCRIPTION XML 83

ANNEX F SERVICE INSTANCE DESCRIPTION 84

List of Tables

Table 1 Type Definition 8

Table 2 21

Table 3 Information Elements of the Service Instance Description 24

Table 4 Service identification 36

Table 5 Requirements Tracing 38

Table 6 Requirements Definition 38

Table 7 Operational Nodes providing the XYZ service 39

Table 8 Operational Nodes consuming the XYZ service 39

Table 9 Operational Activities supported by the XYZ service 39

Table 10 Service Interfaces 41

Table 11 Describing a service data model 42

Table 12 Definitions of terminology – Technical Service 46

Table 13 Service Design Identification 50

Table 14 Service Interface Mapping 52

Table 15 Service Data Model Description 54

Table 16 Payload description of <operation name> operation 56

Table 17 Definition of terminology – Technical Service 58

Table 18 Service Instance Identification 62

Table 19 Definition of terminology – Technical Service 65

List of Figures

Figure 1 Service Management Concept 9

Figure 2 Distinction between Service Specification, Service Technical Design and Service Instance 10

Figure 3 Service Documentation Overview 11

Figure 4 Structure of the Service Instance Description 23

Figure 5 <Service Name> Interface Definition diagram 41

Figure 6 <Service Name> Service Data Model diagram 41

Figure 7 <Service Name> Interface Parameter Definition diagram for <operation name> 43

Figure 8 <Service Name> Operation Sequence Diagram 45

Figure 9 <Service Name> Orchestration Sequence Diagram 45

Figure 10 <Service Name> Interface Definition diagram 52

Figure 11 <Service Name> Service Data Model diagram 53

Figure 12 **<Service name>** Interface Parameter Definition diagram for <operation name> 56

Figure 13 <Service Name> Operation Sequence Diagram 57

# INTRODUCTION

## 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.

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 guidance included helps provide 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.

## Link to S-100 and Product Specifications

This Guideline is intended for but is 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.

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. |

# OVERVIEW

## Service Management Overview

A service management concept can be visualized, 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 realized as a service itself that would have an Application Programming Interface (API) for automatic interfacing to the registry (lookup, updating, deleting etc.).

In Figure 1, four different roles are envisioned in managing the contents of a service registry following this guideline.

* Service specification producers might represent organizations governing a certain type of service type like Navigational Warning Service, Route Optimization Service, Search and Rescue Service etc.
* Service implementers typically develop services in accordance with published specifications resulting in service technical designs. i.e. realizing a service specification by use of a certain technology like REST, SOAP, FTP etc.
* Service providers publish implemented technical services following a service technical design. Service providers are responsible for publishing service instances ready for consumption in a service registry. Examples of service providers are Shipping companies, VTS, National Maritime Administrations etc.
* Service consumers discover service instances for consumption. In this context, the service consumer considered is could be an actual person or an application like an ECDIS software or perhaps a VTS software. In discovering service instances, a service consumer might need to make use of all three service description artefacts (specification, design and instance) in order to determine the full specification of the published service instance.



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 a logical level in a technology-agnostic manner by providing, for example:

* The operational context of the service (e.g., requirements, use cases)
* The logical operations and the parameters required with respect to the logical data model (e.g. if a get-type query is an operation that is expected, what kinds of query parameters will it allow and what is expected as a return value)The logical data model used by the service (preferably defining the subset of S-100 series data that used by the business requirements of the service)
* The dynamic behaviour of the service in the context of the use cases (sequence of logical operations)
* Author of the service specification (organization, contact person)

The service specification shall not describe the details of specific service implementation. For that purpose, a service technical design description must be provided, where the actual realization of the service with a dedicated technology shall be described. Earlier versions of this specification included the definition of interfaces and parameters in the service specification. This has now been removed as interfaces and parameters differ between technologies. For this reason the dynamic behavior of the service must also be specified in the service specification at the level of logical operations without tying the implementation to a given technology or architecture.

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
* The service interface descriptions (operations, parameters)
  + When using a standardized communication pattern (e.g. SECOM) the interface description may gloss over some of the operations and parameters that are defined in the standard and focus on those interfaces and parameters that have actual business case value for the use cases the service is designed to solve.
  + It is important to document how the actual interfaces correspond to the logical operations described in the specification.
* 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 (organization, 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

**MUST be redrawn!**

* Coverage information



1. Distinction between Service Specification, Service Technical Design and Service Instance

Figure 3 provides an overview of 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 design description template - a word document providing the framework for a textual description of the service technical design (ANNEX E); Service design templates based on a specific technology may also be made available to ease the writing and development of designs.
* service instance description XSD - an XML schema definition for the formal description of a service instance (ANNEX C); and
* service instance description template - a word document providing the framework for a textual description of the service instance (ANNEX F).



**MUST be redrawn!**

1. 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 a:

* service specification document - a word document (complying with the service specification template ANNEX D) detailing the service specification and focusing on the definition of the use cases, logical operations and logical data model of the service. The service specification must provide sufficient detail in the use cases and logical data model that the technical designs are only an additional layer of specification without any duplication and focus on detailing how to implement the use cases and logical data model in an actual implementation using a chosen technological approach.

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;

* 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 and the interfaces supplied or required.

The contents of this formal description must be mapped to the service data model and the logical operations defined in the specification. 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; and

* a template of the instance specification XML with all of the relevant interface specification and data models defined in a formal way to reduce duplication of work from the parties implementing an instance of the service

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; and

* 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. This should be an easy edit of the template provided by the service design.

*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.

## 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.

### 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 including the logical operations and 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 an appropriate registry, most likely the envisioned IALA Maritime Resource Registry (MRR) with the unique MRN obtained from the approving body of the specification

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 appropriate 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 appropriate 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 areas.

### 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 specifications (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 applicable service registries. Interested consumers may then look up services in the registries.

# 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:
* Logical operations of the service
* Service payload definition at a logical level
* Service dynamic behaviour description at the level of actors and use case derived operations without technical details that tie the implementation to a certain technical design
* 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;
* (recommended) a model based description - e.g., a UML model describing the logical operations and data structures.

The service specification document may and should 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 realization of the service with a dedicated technology shall be described. This is the reason why the service specification must not include interface defnitions or a formal definition of the service data model as they are dependent on the logical data model as well as the implementation technology.

## Service Specification Description 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 also provides the foundation material for the future standardization process.

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

## Service Specification Description 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.

### Introduction

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

### Service Identification

The service identification section provides a tabular overview of mainly administrative attributes needed for identification and lookup of the service. Examples include 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).

### 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 Maritime Services, 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). In particular, information exchange requirements are of great 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.

### 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 a UML modelling tool.

Architectural elements applicable for this description are:

* Service - the element representing the service, as a whole
* Service Operations - describe the logical operations used to access the service. Actual technical designs may rename some of the operations or ignore some of the operations if the operation is triggered in a different way e.g. API-based designs require more operations than an event-based design.
* Service Operations Parameter Definitions - identify logical data structures being exchanged via Service Operations. These parameters must not be an exhaustive list but describe the parameters that are required for the business logic to work.

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 a UML diagram[[1]](#footnote-2) that illustrates the logical operations and their interaction between the service provider and consumer and other necessary parties. This information shall also be provided in text to ensure accessibility and further understandability. The text may be in tabular form or freeform text.

It is also recommended to describe the considerations resulting in the selection of a logical Message Exchange Pattern (MEP) for the logical operations. However, it is important to note that the logical operations do not need to map one-to-one to actual interfaces in the design.

### Service Logical 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 avoid listing all details or to defining technology-specific data types. It is recommended to visualize the data structures by means of suitable diagrams. The complete information model (in the form of logical data structures) shall be shown using diagram(s) and explanatory text e.g. 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 logical data model is related to an external data model (e.g., being a subset of a standard data model, e.g., typically based on an S-100 specification), then the service data model shall refer to it. Each data item of the service data model shall be mapped to a data item defined in the external data model. This mapping may be added in the same table that describes the data items and their attributes and associations. The idea is that, when reading the service specification (including the logical service data model), the reader must clearly understand the payload structures. If the service re-uses structures of an external data model, then these structures can be referred to rather than replicated in the service specification. The tabular presentation of the payload allows the provision of references to an externally defined model.

However, while references to external data models are allowed and encouraged, all of the fields and enumerations that are used should be described in relation to the business logic of the service and make implicit how an implementing service should use different fields and enumeration values.

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 on 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.

### Service Dynamic Behaviour

This section describes the interactive behaviour between logical service components including users, consumers and providers, and other services. The actual interfaces, technical state machines and technical orchestration is not of specific interest at this level. The focus must be to document the logical sequences beteen actors, human or software.

The following types of views and diagrams can be used to describe the dynamic behaviour[[2]](#footnote-3):

* Activity diagrams
* Sequence diagrams
* Other interaction diagrams

The dynamic service behaviour description should be at the level of user stories and should cover a single user story per sub-section. Related user-stories should be grouped close to each other.

This section Should also include the definition of the logical operations and their expected parameters and types of return data in relation to the logical data model to better formalize the required behavior of the service while staying technology agnostic.

### References

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

# SERVICE TECHNICAL DESIGN

## 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, including the communication pattern selected. 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.

## 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.

### Introduction

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

### 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), keywords, etc.

### Technology Introduction

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

### Service Design Overview

This section aims at providing an overview of the main elements of the service design and mapping the design elements to the service specification elements. The overview should include necessary diagrams and accompaniying explanatory text.

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 a suitable diagram[[3]](#footnote-4) 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 Message Exchange Patterns (MEP) are realized with the chosen technology.

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 visualized. 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.

### 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*:

* Diagrams that represent the data structures, including detailed physical data type descriptions at attribute level.
* A file describing the data structures (e.g., XML/XSD, JSON). This is required when data structures are not covered by existing standards (e.g. S-100 series, SECOM, MMS);
* Tabular presentations if not covered by existing standards and their documentation.

Any mixture of the above formats is allowed. An 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.

### Service Interface Design

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 suitable diagrams or API documentation.
* Parameters – shall describe the unambiguous data structure of input and output parameters of the operation (payload) by using suitable diagrams or references to existing standards and explanatory tables as required while avoiding duplication of documentation that already exists in other documents.

\*\*\*TODO\*\*\* add specific that design may have operations and parameters that are not in specification as long as all operations and parameters that are required in specification are covered.

It is mandatory to provide 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. If such a documentation exists in the referenced standards a reference to it must be provided, but duplication of effort is not required.

It is suggested that the service interface design also provides a sample interface documentation of the service in e.g OpenAPI format [5]. And example definition of the service instance XML (see 5.3) should also be provided to avoid duplication of work by implementing parties.

### Service Dynamic Behaviour

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

The 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 to extend the service design structure (see section 4.2.4) from a logical description of interactions to a structure where actual operations and interfaces are referenced. This will also allow better understanding of the interaction between different services.

### 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.



# SERVICE INSTANCE

## 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 realized. 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.

## 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.

### Introduction

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

### 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) and key words.

### 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, composition of services, internal service structure etc.

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

### Release Notes

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

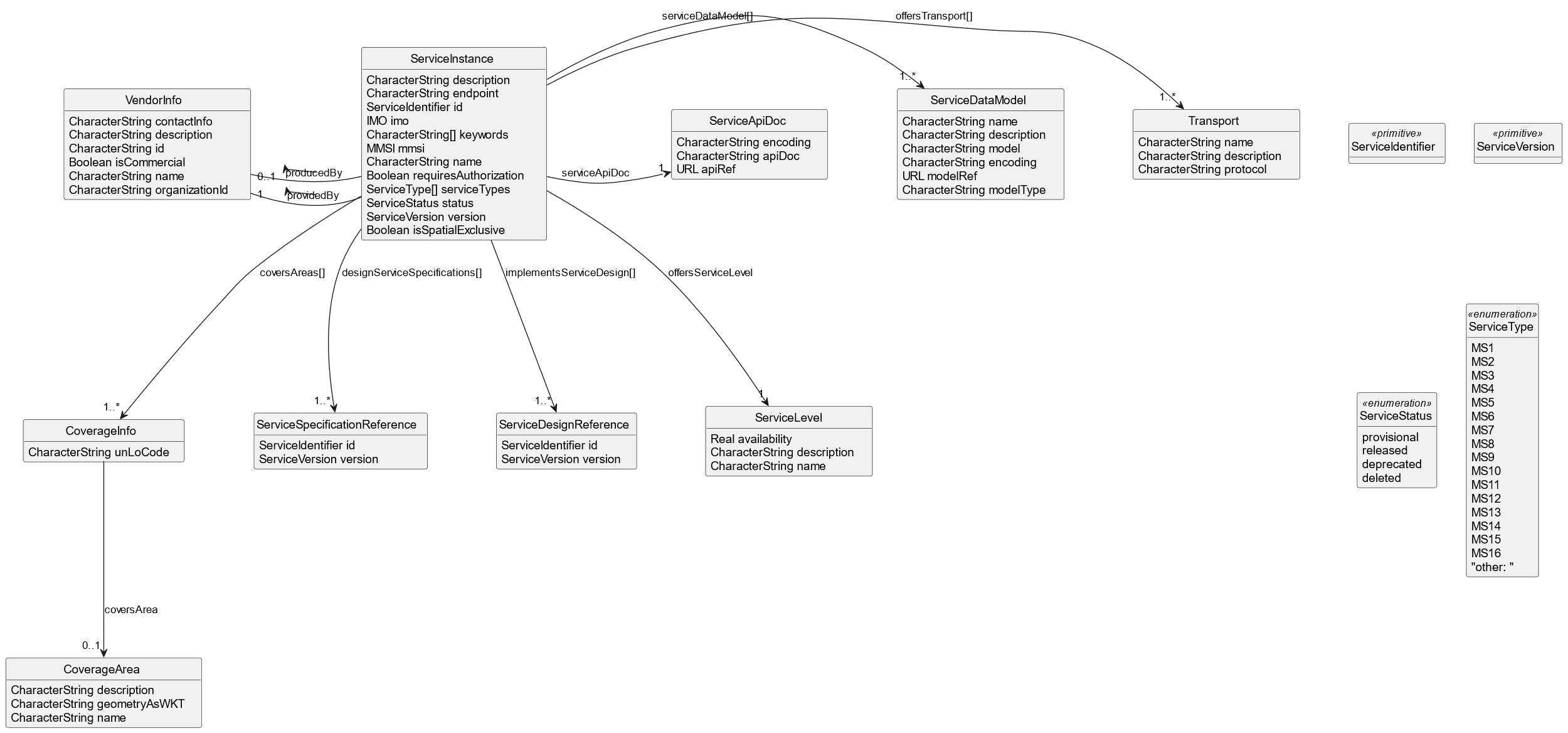
* release identification and date;
* feature list, including:
* added features;
* changed features; and
* removed features.
* bug list including:
* known open bugs; and
* resolved bugs.

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

### 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.

## Service Instance Description XSD Structure



1. Structure of the Service Instance Description

Figure 4 gives an overview about the formal description of the service instance description. The individual items are described in the Table 3.

The service instance description schema is presented in ANNEX C.

1. 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.  Globally unique identification of the service instance. Newer versions of the same service instance shall adopt the same id.  The identifier should conform to a Maritime Resource Name (MRN) identity. |
| 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.  Keywords shall contain searchable words characterizing the service provided.  The keyword list is recommended to reflect type of service but can also contain dataformats, datamodels, datatypes etc. possible to exchange by the service instance. |
| endpoint | CharacterString | Endpoint that describes where the service endpoint is located (e.g., URL) |

|  |  |  |  |
| --- | --- | --- | --- |
|  | MMSI | CharacterString | Optional Maritime Mobile Service Identity.  MMSI should be registered with 9 digits or left empty (blank). |
| IMO | CharacterString | Optional International Maritime Organization (IMO) number.  IMO should be registered with 7 digits or left empty (blank). |
| serviceTypes | ServiceType  <<enumeration>> | Optional field to categorize a list of servicetypes.  The service type shall reflect the associated operational service type provided according to defined types in paragraph Service Type. If type not yet defined, a proposed type will be reviewed and added when accepted, i.e., an open enumeration.  The following values are currently accepted in accordance with IMO Maritime Service portfolio:   * MS1 : MS 1 – VTS Information service (INS) * MS2 : MS 2 – VTS Navigational assistance service (NAS) * MS3 : MS 3 – Traffic organization service (TOS) * MS4 : MS 4 – Port support service (PSS) * MS5 : MS 5 – Maritime safety information (MSI) service * MS6 : MS 6 – Pilotage service * MS7 : MS 7 – Tug service * MS8 : MS 8 – Vessel shore reporting * MS9 : MS 9 – Telemedical assistance service (TMAS) * MS10 : MS 10 – Maritime assistance service (MAS) * MS11 : MS 11 – Nautical chart service * MS12 : MS 12 – Nautical publications service * MS13 : MS 13 – Ice navigation service * MS14 : MS 14 – Meteorological information service * MS 15 : MS 15 – Real-time hydrographic and environmental information services * MS16 : MS 16 – Search and rescue (SAR) service * Other: [something] |
| requiresAuthorization | Boolean | Indicates whether authorization is required or not. |
| designServiceSpecifications | ServiceSpecificationReference | Refers to the service specification that the implemented service design is an actualization of. Must also include the version number of the service specification. May refer to multiple different service specifications and/or multiple versions of the same service specification. |
| implementsService-Designs | ServiceDesignReference | Refers to the service design including version that is implemented by this service instance. A service may reference multiple service designs if it implements them. It may also refer to multiple service design versions if it supports them. |
| coversAreas | CoverageInfo | Mandatory reference to a list of geographical areas 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. |
|  | serviceApiDoc | ServiceApiDoc | The API documentation of the service. This should be in OpenAPI or other widely used and standardized format. The elements of this are:   * Encoding The encoding of the API documentation * apiDoc The actual API documentation if embedded (this is not preferred) * apiRef The reference to the location of the API documentation (this is preferred)   The API doc of the implemented designs should only be modified to exclude any unsupported unrequired operations and to combine the API docs of multiple designs if the service supports them. |
|  | offersTransport | Transport | 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. |
|  | serviceDataModel | ServiceDataModel | The ServicePhysicalDataModel describes the data model for the service design. The ServiceDataModel describes in detail all the data structures being actually exchanged when service consumers interact with a service instance that implements this design.  It is important to note that for findability having the supported data models listed in this element has value. In most cases, the actual data model will be documented in detail as a part of the API documentations.   * 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. Alternatively, the model may refer to an externally available file that defines the model URL with the following attributes. This prevents the need for duplication of effort or embedding possibly changing model documentation in this metadata file. * modelRef The URL of an external file that defines the data model used. Alternative to embedding the data model in the model attribute. * modelType The modelType shall contain an abbreviation that indicates what technology is used to describe the model. E.g. WSDL, JSON. * Encoding The encoding of the model. |
|  | isSpatialExclusive | Boolean | 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. |
| Type Name | | Description | |
| ServiceDesignReference / ServiceSpecificationReference | | A reference to the service specification/design that is implemented by the service instance. – It has the id and the version of the respective service specification/design. | |
|  | Element Name | Type | Description |
| id | ServiceIdentifier | Service identifier type to be used by service specifications, designs, instances.  Identification of the service design implemented by the service instance.  The identifier should conform to a Maritime Resource Name (MRN) identity. |
| 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 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 | Defines a geographical area from which the service instance is accessible. |
| unLoCode | CharacterString | The "United Nations Code for Trade and Transport Locations" is commonly more known as "UN/LOCODE".  The unLoCode should be registered with 5 characters, no space, capital letters or left empty (blank). |
| Type Name | | Description | |
| CoverageArea | | Defines a geographical area from which the service instance is accessible. | |
|  | Element Name | Type | 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.  The identifier should conform to a Maritime Resource Name (MRN) identity. |
| isCommercial | Boolean | Optional indication on the commercial status of the vendor. |

# 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 publication of service specifications;
* the life cycle of service specifications and service instances; and
* conformance of service instances to specifications.

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

# 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.

In addition, for this document, the following terms are relevant:

|  |  |
| --- | --- |
| *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, etc. |
| *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 realization.  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 characterized 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 realization 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 focuses on describing the use cases of the service and high level interactions between the service and other actors (human or services, application, etc). The service specification also includes the description of the logical data model of the service with a reference to any or all external data models that are used. Ideally the logical data model is a subset of an existing 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 characterized 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. |

# Abbreviations

API Application Programming Interface

EPSG European Petroleum Survey Group (a spatial reference system)

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 Organization

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

# 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)
4. This concept is implemented as the Maritime Service Registry within the Maritime Connectivity Platform (MCP, formerly called the Maritime Cloud), see <http://www.maritimeconnectivity.net>.
5. OpenAPI Initiative <https://www.openapis.org/>
6. SERVICE INSTANCE DESCRIPTION SCHEMA

**Note:** For readability and reuse, the XSD file is published at: XXX . The contents listed here are for reference only.

TODO actual XSDs once completed

ServiceInstanceSchema.xsd

1. SERVICE SPECIFICATION TEMPLATE

TODO this requires updating once the content changes defined above have been locked down.

**Service Specification for the *xxx* Service**

* 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.

* + 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 standardization 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 realization 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
  + 1. Intended Readership

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

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

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

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

* + 1. 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.

* 1. 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.

1. 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 organization |
| **Status** | Status of the service in the engineering lifecycle – either ‘Provisional’, ‘Released’, ‘Deprecated’ or ‘Deleted’.[[4]](#footnote-5)  ‘Provisional’ - the service necessity has been identified, and a short description is available, but the full-service specification is not yet ready.  ‘Released’ - the full-service specification is ready.  ‘Deprecated’ - service specification is announced to become invalid in the near future.  ‘Deleted’ - service specification is not valid any more. |

* 1. 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:

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 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, 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.
  + 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.

1. Requirements Tracing

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement Id** | Requirement Name | Requirement Text | References |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

1. Requirements Definition

|  |  |
| --- | --- |
| **Requirement Id** |  |
| Requirement Name |  |
| Requirement Text |  |
| Rationale |  |
| Author |  |

|  |  |
| --- | --- |
| **Requirement Id** |  |
| Requirement Name |  |
| Requirement Text |  |
| Rationale |  |
| Author |  |

* + 1. Other constraints
       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.

* + - 1. 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.

1. Operational Nodes providing the XYZ service

|  |  |
| --- | --- |
| **Operational Node** | Remarks |
|  |  |

1. Operational Nodes consuming the XYZ service

|  |  |
| --- | --- |
| **Operational Node** | Remarks |
|  |  |

* + - 1. 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.

1. Operational Activities supported by the XYZ service

|  |  |
| --- | --- |
| **Operational Node** | Remarks |
|  |  |

* 1. 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.

* + 1. Service Interfaces

Describe the interfaces of the service including the selected Message Exchange Pattern (MEP) by using an UML diagram[[5]](#footnote-6) 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 visualized 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.



1. <Service Name> Interface Definition diagram
2. Service Interfaces

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

* 1. 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 visualize 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:



1. <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.

1. Describing a service data model

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

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

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

* + 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.

* 1. 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.

* + 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.

* + - 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.

* + - * 1. Operation Functionality

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

* + - * 1. 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).



1. <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.

* + - 1. OPERATION <OPERATION NAME>

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

* + 1. Service Interface <Interface Name>

Repeat previous section for each interface.

* 1. 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]](#footnote-7)

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



1. <Service Name> Operation Sequence Diagram
   * 1. Service Interface <Interface Name>

Repeat previous section for each service interface.

* + 1. 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).



1. <Service Name> Orchestration Sequence Diagram
   1. 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.

* 1. 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.

* + 1. Terminology

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

1. 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. |

* 1. Abbreviations

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

|  |  |
| --- | --- |
| **ACRONYM** | Meaning |
|  |  |
|  |  |
|  |  |

* 1. 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>
4. 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.

1. SERVICE TECHNICAL DESIGN TEMPLATE

TODO this requires updating

Service Design Description for the *xxx* Service

<*xyz Technology*>

* 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.

* + 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 realized 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 realizes more than one service specification.

The purpose of this service design description document is to provide a technology-specific description of how to realize 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 realization 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.
  + 1. 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.

* + 1. 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.

* 1. 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.

1. 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 organization |
| Status | Status of the service design in the engineering lifecycle – either ‘Provisional’, ‘Released’, ‘Deprecated’ or ‘Deleted’.*[[7]](#footnote-8)*  ‘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. |

* 1. 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.

* 1. SERVICE DESIGN OVERVIEW

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

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

Architectural elements applicable for this description are:

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

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

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

* + 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 realized with the chosen technology.

An example diagram and corresponding table are shown at Figure 12 and Table 15.



1. <Service Name> Interface Definition diagram
2. 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 symbolized 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).

* 1. 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 a UML diagram.



1. <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.

1. Service Data Model Description

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

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

* + 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.

* 1. 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.

* + 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.

* + - 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.

* + - * 1. Operation Functionality

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

* + - * 1. 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).



1. <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.

1. 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. |

* + - 1. Operation <Operation Name>

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

* + 1. Service Interface <Interface Name>

*Repeat previous section for each interface.*

* 1. 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]](#footnote-9)

* 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.

* + 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.



1. <Service Name> Operation Sequence Diagram
   * 1. Service Interface <Interface Name>

Replicate previous section for each service interface.

* 1. 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.

* + 1. Terminology

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

1. 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. |
|  |  |
|  |  |
|  |  |

* 1. Abbreviations

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

|  |  |
| --- | --- |
| **ACRONYM** | Meaning |
|  |  |
|  |  |
|  |  |

* 1. 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 Guidelineline 1128 11?? on Specification of e-Navigation Technical Services
2. *XYZ Service Specification* *xx.yy*  *for the XYZ service*
3. 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.

1. SERVICE INSTANCE DESCRIPTION

Service Instance Description for the *xxx* Service

<Service Instance Name>

* 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.

* + 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 realized 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.
  + 1. 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

* 1. SERVICE INSTANCE IDENTIFICATION

1. 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 organization 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. |

* 1. 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.

* 1. 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.

* 1. 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.

* + 1. Terminology

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

1. 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. |
|  |  |
|  |  |
|  |  |

* 1. Abbreviations

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

|  |  |
| --- | --- |
| **Abbreviation** | Meaning |
|  |  |
|  |  |
|  |  |

* 1. 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 xx.yy for the ABC service*
3. *XYZ Service Design* *xx.yy*  *Description for the XYZ service*

1. e.g., in NATO Architectural Framework (NAF), a NSOV-2 diagram could be used. [↑](#footnote-ref-2)
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. [↑](#footnote-ref-3)
3. e.g. in NATO Architectural Framework (NAF), a NSOV-2 diagram could be used [↑](#footnote-ref-4)
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. [↑](#footnote-ref-5)
5. e.g. in NATO Architectural Framework (NAF), a NSOV-2 diagram could be used. [↑](#footnote-ref-6)
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. [↑](#footnote-ref-7)
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. [↑](#footnote-ref-8)
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.* [↑](#footnote-ref-9)