Document Revisions (Title style)

International Association of Marine Aids to Navigation and Lighthouse Authorities

***AISM***Association Internationale de Signalisation Maritime ***IALA***

10, rue des Gaudines

78100 Saint Germain en Laye, France

Telephone: +33 1 34 51 70 01 Fax: +33 1 34 51 82 05

e-mail: [contact@iala-aism.org](mailto:iala-aism@wanadoo.fr) Internet: [www.iala-aism.org](http://www.iala-aism.org)

**IALA Recommendation**

**[ENAV-###]**

**On the**

**Generic Service Engineering Model [for e-Navigation Shore Infrastructure]**

**[Working towards] Edition 1**

**[2015]**

**Initial Edition**

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

|  |  |  |
| --- | --- | --- |
| **Date** | **Page / Section Revised** | **Requirement for Revision** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

IALA Recommendation on Generic Service Engineering Model

(Recommendation ENAV-[####])

THE COUNCIL:

**RECALLING** the function of IALA with respect to Safety of Navigation, the efficiency of maritime transport and the protection of the environment;

**RECOGNISING** that Text ;

**RECOGNISING ALSO** that Text ;

**RECOGNISING FURTHER** that Text :

**NOTING the** Text ;

**NOTING ALSO** that Text ;

**NOTING FURTHER** that Text ;

**CONSIDERING that** Text

**ADOPTS** in the annex of this recommendation; and,

**RECOMMENDS** that National Members and other appropriate Authorities providing marine aids to navigation services:

1 - Text

Table of Contents

[1.1 Heading 2 5](#_Toc395600713)

[1.1.1 Heading 3 5](#_Toc395600714)

[1.1.2 Heading 3 5](#_Toc395600715)

[2.1 Heading 2 again 5](#_Toc395600716)

[2.1.1 Heading 3 again 5](#_Toc395600717)

Index of Tables

[Table 1 Title required (Title goes above the table) 4](#_Toc216497068)

Index of Figures

[Figure 1 Title required (Title goes below the figure) 4](#_Toc216497075)

Annex

**Main Part of the Generic Service Engineering Model**

# Introduction

The main building block of the Common Shore-based System Architecture (CSSA) [0.2] is the so-called technical service. This Recommendation provides a common generic engineering model template for the technical services of the CSSA.

This common generic engineering model template views the technical service from a variety of angles of relevant perspectives, which complement each other. To present this bounty of material in a reader-friendly, orderly and consistent manner, this Recommendation consists of this main part and a variety of Appendices. This main part introduces the different angles of perspectives and thereby the different Appendices and ties them together.

This engineering model, as composed by the different relevant perspectives complementing each other, is meant to be *the* template for technical services, i.e. all technical services described *within* the CSSA should adhere to *this* engineering model.

As a well developed example for the application of this generic engineering model to a specific technology, the AIS Service description has been developed in parallel (refer to IALA Recommendation A-124, Ed. 2). The AIS Service provides an example of a technical service with a Physical Layer and also exhibits a very large technology specific complexity. Hence, this generic engineering model was validated by using a “difficult” application.

For other technologies which exhibit a lesser degree of complexity in functional and architectural terms, this generic engineering model may be tailored to need while maintaining its core concepts and tenets.

**Note: For the purpose of this generic engineering service model, the term “technical service” is abbreviated with “service”. Once a specific service is addressed, it is capitalized, e.g. in “Basic Service” or “AIS Service”. Also, when considering the “service under consideration”, this is abbreviated to “own service” (as opposed to the “requesting service” or the “requested service”).**

**Please note that it is advisable to read this main part at least once before diverging to any Appendix to gain an overview. Also, it is advisable to re-visit this main part several times after consulting Appendices.**

# Context of the document

Figure 1 (overleaf) provides the reader with the relative place of the present document within the IALA e-Navigation documentation framework: This document builds upon the documents “above it” in Figure 1, in particular upon

* the general e-Navigation architecture, in particular its shore aspects, as explained in IALA Recommendation e-Nav 101 [0.1]
* the system layout definition of the common shore-based e-Navigation system architecture, as defined in IALA Recommendation e-Nav 201 [0.2].

This document informs the documents “below it” in Figure 1, in particular all of the technology-specific descriptions of the individual shore-based technical e-Navigation services at the bottom of Figure 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **e-Nav Recommendations on generalities** | | | | |
| **Table of Contents for “e-Navigation compliant” IALA Documentation** | | | | |
| **Shore-based e-Navigation system terminology (= glossary)** | | | | |
| **e-Nav Recommendations on**  **e-Navigation conceptional aspects and user requirements viewed from ashore** | | | | |
| **The e-Navigation Architecture – the Shore-Based Perspective** | | | | |
| **e-Navigation system requirement analysis and e-Navigation applications descriptions:**  arranged in fields of e-Navigation applications as defined by OPS WG | | | | |
| **e-Nav Recommendations on  technical generics of the common shore-based e-Navigation system architecture** | | | | |
| **common shore-based e-Navigation system architecture** | | | | |
| **technical e-Navigation services (generic engineering model)** (= this Recommendation) | | | | |
| **The Universal Maritime Data Model  and the data model of the common shore-based e-Navigation system architecture** | | | | |
| **Interface considerations with shore-based systems and between shore-based systems** | | | | |
| **Infrastructure considerations** | | | | |
| **Life-Cycle-Management Considerations: Technical Operation Personnel + Tasks/ Technical Development Personnel + Tasks** | | | | |
| **Quality Management and Certification Considerations** | | | | |
| **e-Nav Recommendations on shore-based e-Navigation applications mapping to e-Navigation services involved - examples** | | | | |
| **Shore-based e-Navigation applications involving “visual” Aids-to-Navigation** | | | | |
| **Shore-based e-Navigation applications involving VTS** | | | | |
| … | | | | |
| **e-Nav Recommendations on technology-specific e-Navigation services – examples** | | | | |
| **AIS Service** | **Radar Service** | **“Visual” AtoN Service** | **User Interaction Service** | … other technical e-Navigation services … |

Figure 1: [Future] IALA e-Navigation documentation layout

# 

# OVERVIEW ON THE RECOMMENDATION STRUCTURE

For several practical reasons this Annex to this Recommendation was sub-divided into a main part (i.e. this part) and a number of topical Appendices. The overall list of Appendices is given as follows.

The following list is the list of Appendices of the present Recommendation on the generic service engineering model template.

Since the present Recommendation is meant to be a template to inform IALA Recommendations on individual, i.e. technology-specific services (such as AIS Service or Radar Service), the order of appearance and the numbering scheme was *developed to be used in an identical fashion throughout the IALA Recommendations on individual, i.e. technology-specific services*. This would allow for *ease of reference* to this Recommendation while providing some *flexibility* regarding in particular technology-specific aspects related to installation and life-cycle management: Additional appendices on e.g. technology-specific runtime configurations may be added as Appendices 15ff. *Hence, designers of technology-specific service documentations are strongly encouraged to adhere to this setup of Appendices.*

**Main Part = this document**

**Appendix 0 – References, Glossary of Terms, Abbreviations**

This Appendix 0 contains all references, terms and abbreviations from throughout this Recommendation.

**Deliverables of a service to the Common Shore-based System (CSS):**

**Appendix 1 – Basic Service Categories definitions and Minimum Set of Required Basic Services**

**Appendix 2 – Data objects of a service and their properties (data model)**

**Appendices related to the layout and setup of a service:**

**Appendix 3 – Distribution model**

**Appendix 4 – Interaction and data storage model**

**Appendix 5 – Interfacing model**

**Appendix 6 – Internal Time Latency model**

**Appendix 7 – Internal Reliability model**

**Appendix 8 – Test model**

**Appendices related to functional components of a service**

**Appendix 9 – generic functional description of the Logical Shore Station**

**Appendix 10.1 – generic functional description of the Physical Layer as a whole**

**Appendix 10.2 – generic functional description of the Physical Shore Station**

**Appendix 10.3 – generic functional description of the Physical Link Terminal Equipment**

**Appendix 10.4 – generic functional description of the Physical Link Couplers**

**Appendix 11 – generic functional description of the Service Management**

**Appendices related to installation and life-cycle management issues**

**Appendix 12 – Site co-location issues and on-site infrastructure considerations**

**Appendix 13 – Recommendation regarding efficient operation and maintenance of a service**

**Appendix 14 – Runtime configuration management of a service**

The meaning of those Appendices and their respective content specification will be given in due course in the remainder of this main part.

# Overview on shore-based technical services

The layout of the CSSA [0.2] reflects the *user-requirement driven top-down approach of IMO’s e-Navigation concept.* The functionality and data delivery required from the technical Common Shore-based System (CSS) by the various users are fulfilled by arranging the necessary functionality and data storage in a so-called *Service-Oriented Architecture (SOA)*.

In a SOA a “service” is the main building block and its composition is developed starting from the service-delivery to the requesting entity, which may be the (human) user or another technical service or system. So, the *very name of a “service” reflects the user-requirement driven approach stipulated by IMO*.

IALA members are tasked with the deployment and continuous operation of the user-required functionality and data delivery of their shore-based system. When deployment and continuous operation is considered, the user requirements are not the only aspects to consider, however. Cost efficiency of deployment as well as minimization of expenditure over the life-cycle of the system need to be taken into account, for instance. These aspects are summarized by the term “Life Cycle Management”, and they have a large impact on the engineering design of the services besides the user requirements.

Analyses have shown, that there are also certain structural and functional commonalities between different services, which must be exploited taking into account the above cost efficiency requirements. These commonalities have lead to the development of the present generic engineering model for the services.

The present generic engineering model was developed based on in-depth understanding of a variety of different technologies. The shore-based AIS Service was one very important technology in this regard (and it still serves as application example of this generic service model to a complex technology). Now that the generic engineering model exists, it may be applied to any other technology contained in the CSSA. Hence, it serves as a generic engineering model template.

The benefits of that approach are:

* A pool of well defined generic solutions exists, and these solutions may easily be applied to different technologies;
* Thereby this generic engineering model template also serves as a “checklist” for the international community when addressing the deployment of any new technology for the shore side of the e-Navigation concept;
* A common understanding between engineers for the services is achieved internationally; this is not only one of the objectives of IMO’s e-Navigation concept but also serves as a common ground between IALA National members and industry;
* This generic engineering model template provides a powerful tool for management decisions within IALA national members, i.e. Aids-to-Navigation authorities.
* This generic engineering model template promotes the concept of modularity and will allow mixing and matching of best in class components.

# Fundamental service design considerations

The service is embedded in its CSS, as indicated above. Therefore, there are features of the service under consideration, i.e. the “own service”, which are of interest to the system at large, in particular to any of the so-called “requesting services”, and which should be therefore known or be “visible” to the system at large. This requirement applies to the discrete subsets of the overall functionality of own service and to the data elements exchanged between own service and the system. Since these elements are visible to the system at large, they will be described first (in this chapter). Conversely, there are features of own service which are of interest within the boundaries of own service and for the technical personnel operating, maintaining and further developing own service. These features, which are the majority of the generic service model, should be “encapsulated” within own service in order not to unnecessarily burden the system at large. These features will be discussed in this chapter afterwards. The boundary of own service thus demarcates the difference between “external”, i.e. visible to system at large and “internal”, i.e. encapsulated within own service. This terminology will be used throughout this document.

## The notion of the Basic Services (BS)

Definition: A *Basic Service (BS)* is a discrete subset of the overall functionality of a of a service. It represents the functional delivery of a defined set of data objects or a defined set of functionality performed by that service. Basic implies that it is one consistent fundamental functionality of own service. The notion of the Basic Services is essentially driven by a requirement-fulfilment (i.e. client-server) relationship from the shore’s point of view, *eventually* driven by the shore-based user’s point of view.

Every service of the CSS comprises a minimum set of Basic Services. A Basic Service is always represented by a name and a mnemonic meaningful in the context of that service.

### The mitigation of complexity by the definition of External Basic Services

Any service is part of the CSS. Hence it would be of primary interest to any other service, what capabilities the specific service under consideration contributes to the CSS. These capabilities can be defined as *External Basic Services (E-BS)* of own service. They are called “external” because data and/or the effects of some functionality cross the boundary of own service .

External Basic Services are all *justified by at least one desired interaction between own service and some entity outside the boundary of own service.* These entities are other services within the CSS which request data and/or functionality, and are hence called *“requesting service”*. In the case of the User Interaction Service and the Gateway Service, both of which interface the CSS to shore-based users (humans; external systems), these users could also be that entity requesting External Basic Services.

All External Basic Services of own service taken together provide the summary capability statement of that service from the perspective of the CSS or – in the case of the User Interaction Service and the Gateway Service – by the shore-based user. They thus contribute to the reduction of complexity by encapsulation (compare Figure 2).

**CSS**

**Point of view**

**from ashore**

**Encapsulation of complexity   
for shore-based e-Nav applications**



**Data  
Collection and Data Transfer Service**

**Physical Link**

**Appli-cation**

**Appli-cation**

**Appli-cation**



**External Basic Services of the service as its functional interface to the CSS**



**Figure 2: External Basic Services of a Data Collection and Transfer Service (by example) as comprehensive capability statement of that service**

For every service a list of its External Basic Services is given.

For each of the External Basic Services one functional description needs to be developed. The full description of the External Basic Service consists of

* its name,
* a mnemonic,
* its External Basic Service Category indication (see below),
* the data objects involved
* a standardized table describing the functions of the interacting components of own service,
* a figure, which maps it to the appropriate components of own service, and
* additional functional detail description as appropriate.

The encoding format of the data objects will be detailed in the appropriate Basic Service definition.[[1]](#footnote-1)

External Basic Services may request the support of other services of the CSS, i.e. rendering them *“requested services”* and turn own service into a “requesting service”.

For a description of the minimum set of External Basic Services refer to **Appendix 1**.

### Basic Service Categories

The *definition of BS Categories* is useful for the following reasons:

* BS Categories *reflect the different net data flow mechanism* within own:
* There are different data paths and data processing requirements for receiving and transmitting within own service with associated consequences for design and operation.
* Although apparently the same data object being treated in different BS Categories, the *meaning* *of that data object depends on the net data flow direction, i. e. its context*. That means, that although the data object looks the same and there is a similarity in appearance and therefore even identity of description, the data objects are different due to their role in the data flow from an ontological point of view.
* To illustrate this, an example from the AIS Service is given: Data received from AtoN AIS stations by ATON\_DAT (received) means something different than transmitting AtoN messages via the AIS Service by ATON\_DAT (transmit), although the data object description looks identical. In the first case there is a physically existing AtoN AIS Station involved, while in the second case, there is a synthetic or even virtual AtoN involved.
* BS Categories *best reflect the data orientation paradigm (as opposed to the technological orientation)*, which is a fundamental feature of the e-Navigation concept.
* BS Categories *reduce the complexity of own service* by *employing functional similarities* to the largest extent possible.
* BS Categories, once defined, *provide a fixed reference point throughout the description of own service* for statements like: “For BS Category x the following applies: ...”.
* BS Categories *increase description efficiency of the functionality of own service substantially*, i.e. even *allow for a minimum description*.
* BS Categories *make the description of own service robust to amendments* of the precise data object content.
* BS Categories *simplify the maintenance of the documentation of own service at large* because additional data contents can be added easily to the appropriate BS Category without the need to change much of the otherwise BS category-related description: The amendment to the BS Category description will have an immediate effect on all individual BS of the same BS Category.

### Internal Basic Services and their categories

There are also Basic Services, which are not justified or requested by an external “requesting service” of the CSS or – in the case of the User Interaction Service or the Gateway Service – by a shore-based external user. These Basic Services are called *Internal Basic Services (I-BS)*: They are supportive in nature, but are essentially required. Examples of Internal Basic Services are recording of service data (pre-requisite to any replay), service initialization, service termination and component management of own service. Internal Basic Services may not be directly visible outside own service on its Physical Link (in the case of a service with a Physical Layer); Internal Basic Services are not directly visible forthe CSS, although their effect may be.

Frequently the more sophisticated functionality related to the specific technology of own service often is an Internal Basic Service: Internal Basic Services often interact with the Technical Operation Personnelresponsible for own service, while this functionality is not requested by the CSS, i.e. while there is no “requesting service”.

The Internal Basic Services are also categorized into Internal Basic Services Categories in a similar fashion to the External Basic Services.

An Internal Basic Service may request the support of other services of the CSS, i.e. rendering them *“requested services”* and turn own service into a “requesting service”.

For a description of the Internal Basic Services which every service needs to have refer to the description of the minimum set of Basic Services in **Appendix 1**.

### Example for a Data Collection and Data Transfer Service

Figure 3 provides an example overview on the External Basic Services of a Data Collection and Data Transfer Service, using the above external Basic Service Categories.

**Figure 3: Abstract example of External Basic Services for a Data Collection and Data Transfer Service**



Note: The dotted lines indicate inheritance relationship.

Figure 4 (overleaf) shows an overview on the Internal Basic Services of a Data Collection and Data Transfer Service as an example in overview.

**Figure 4: Overview on the Internal Basic Services of a Data Collection and Data Transfer Service**



Note: The dotted lines indicate inheritance relationship.

## The Service Data Model – or: data a service delivers and how it is being modelled

### Introduction

The above Basic Services define the functionality provided by own service to the CSS in an abstract manner. They are operations performed on certain data objects, or: they are *f(x)*, to express it in mathematical terms. The data objects treated by the Basic Services are the operands, i.e. the *‘x’*, of these functions. To provide a complete picture of what own service delivers to the CSS, it is necessary to consider those data objects and their structure and mutual relationship. Eventually, it is the data, which is only relevant for other services.

### Usage of the Common Maritime Data Structure

--- Introduce Common Maritime Data Structure

The *Service* *Data Model* provides the well-structured and abstract (functional) description of the data objects, that own service exchanges with requesting services of the CSS, [*and* of the data objects own service uses for internal purposes]. Thus the Service Data Model is an application of the CMDS: In the CMDS all data objects are expressively described in one place together, with data attributes.

Also, the Service Data Model of own service shows the contribution of its different Basic Services to the overall knowledge the CSS. Hence, the Service Data Model shows the contributions of own service to the CMDS.

The Service Data Model of own service provides the designer of requesting services with precise information on what data own service may provide to them, i.e. what the requesting service is going to receive, and on what data own service may expect, i.e. what the requesting service transmits to own service, *without the need to know the technological and/or algorithm details of own service*. Hence the Service Data Model helps to encapsulate the “science” of own service. This reduces complexity for CSS as a whole.

Although, the Service Data Model of any service does ***not*** contain encoding details, encoding is required on application level (i.e. regarding the Basic Services). The Service Data Model supports a variety of general purpose encoding techniques on application level, however. Hence, the Service Data Model provides flexibility in selecting an encoding technique as deemed appropriate. A portfolio of appropriate and compatible encoding techniques to be used with data objects of own service is given in its Interfacing Model (refer to **Appendix 5**).

## The parts of a service and their external interactions in general

**Figure 5** (overleaf) presents the parts of a service and their external interactions.

The *material parts* of a service are called *components of the service*, which are further sub-divided into so-called *service components* and so-called *service-owned infrastructure (components)*.

The service components may be functional software modules or technology-specific hardware components (such as technology-specific base stations, antennas and the like). The service components carry the functionality and/or the data delivery to create the Basic Services, while the service-owned infrastructure components (such as computers and Local Area Networks) are involved in that process but only as a tool and should remain “invisible” in the process. This distinction of service components and their respective functional description from the service-owned infrastructure and their respective requirement statement allows to decouple the more steadily changing or even lasting definition of the actually desired functionality from the rapid technology change at the service-owned infrastructure (compare for example the short-lived computer and IT network generations). Thus this distinction effectively and efficiently supports life-cycle management.[[2]](#footnote-2)

The service components are described in this IALA Recommendation as far as generic features are concerned. [The service-owned components will be described in some other IALA Recommendation as well as the on-site infrastructure (compare Figure 1).] The technology-specific functionalities of the service components are described in the technology-specific description of the individual services, which are informed by this IALA Recommendation. For example, this means that the AIS technology specifics of AIS are described in the IALA Recommendation on the AIS Service as informed by this recommendation.

There are immaterial parts of own service as well, such as various immaterial “models” mentioned in **Figure 5**. Their purpose is mainly to show the interactions between the material components from different angles of perspective. The description of the immaterial parts of own service forms the largest part of this Recommendation. Refer to a dedicated chapter below for a more detailed description.

Finally, there are interactions between both material and immaterial parts of own service with external entities (association indicated by lines in **Figure 5**). “External” means that those entities lie outside the boundary of own service. They may be either entities external to own service or may even be external to the CSS. These interactions are taken into account in the appropriate description of the part.

Note, that the possible interactions with external entities depend on the relative position of a service within the CSS, as indicated by the footnotes in **Figure 5**. From this also the distinction between services with and without Physical Layer is derived, which will be used throughout this Recommendation. Compare IALA Recommendation on the CSSA system layout for a list of the different services in this regard.



**Figure 5: Parts of own service and their interactions with external entities**

## The influence of geographical topology on own service

It is necessary to consider the *influence of geographical topology on the design of own service*. **Figure 6** provides a generic geographic distribution setup of a CSS.

**Figure 6: Generic geographic distribution setup of a CSS and the place of different services**



Despite the differences in individual geography and topology there are *functional similarities* *regarding the distribution of components of the different services of the CSSA*, and also *certain criteria* to select a specific *Service Distribution Configuration*. The Service Distribution Model of own service describes some standard distribution configurations which would apply to most administrations, and also provides technology-specific rationales and criteria.

The Service Distribution Model is essentially functional and generic. It needs to be applied (“tailored”) to the actual geography and topology by every administration. To that end guidance is provided how to arrive at a proper individual distribution model taking into account the following aspects:

* coverage requirements and coverage planning
* coastal topology (concave, convex, line, one spot, ...)
* required functionality setup of own service, i.e. required Basic Services
* distribution of receive / transmit functionalities (if appropriate)
* maintenance and life-cycle considerations (rationalization)
* data transfer network considerations
* infrastructure requirements

In most cases, due to coverage area requirements, own service spans a larger part of an administration’s coast or even the whole of its coast(s). The actual topology of the coastlines may vary from country to country. The actual number of sites needed depends on a variety of aspects. In some cases, own service may shrink to an “one spot service”.

The full development of the different generic geographic distribution setups are contained in the **Appendix 3** (Distribution model).

# The layered structure of a service - Structure Model

The **Structure Model** describes the layered structure of own service and thereby defines in a general way its specific functionality.

## Services with and without Physical Layer

As indicated above, there are services with Physical Layer and without. Hence, the Structure Model will be different in this regard, while there is a lot of commonality regarding the other aspects of the Structure Model.

### Services with Physical Layer

Each and every service with a Physical Layer consists of three main functional layers as follows (refer to **Figure 7**):

* Logical Layer
* Physical Layer: comprises all Remote Shore Stations (RSS) of own service
* Service Management Layer

The *Logical Layer* is the layer which contains *all* functionality of own service, which treats system-requested net data (as opposed to e.g. management and/or maintenance data) *and* which is not tied to a specific location at the waterway (as opposed to the Physical Layer). Hence, by definition, the Logical Layer resides at a Node site (compare Distribution model). The Logical Layer comprises the functional component Logical Shore Station (LSS) and other supportive service-owned components.

The Logical Shore Station (LSS) is the functional component of own service, where the actual net data treatment is done. Hence, by definition, the LSS resides at a Node site (compare Distribution model). In most cases it is a software module residing in a computer (= service-owned infrastructure component).

The *Physical Layer* directly interacts with its Physical Link. The Physical Link could be a radio link or a visual link or an acoustic link, depending on the actual technology of own service. The Physical Link as such does not belong to own service, although own service may influence it by its usage and/or management. Also, the Physical Link is outside the CSS in most cases.

For the description of Remote Shore Stations (RSS), the Physical Layer can be further subdivided into three layers as follows:

* *Physical Shore Station (PSS) Layer*
* *Physical Link Terminal Equipment (PLTE) Layer*
* *Physical Link Couplers (PLC) Layer*

For reasons of *space/direction diversity in coverage of a RSS*, the lowest two layers are often sub-divided into *Sectors*, while their stacked or hierarchical relationship remains. A Sector comprises always Physical Link Terminal Equipment and Physical Link Couplers *together* which are dedicated to a certain space direction regarding coverage. Conceptually wise, omnidirectional coverage ranges can be construed as a One-Sector RSS. Well known examples for the above concept of sectorization are e.g. radio services using directional antennas or fixed visual aids with sector lights.

The Physical Layer of own service may serve *different Physical Links or different physical link technologies simultaneously*, which would lead to another subdivision of the two lowest Layers of the Physical Layer *together*: Physical Link Terminal Equipment X together with Physical Link Couplers X would together and in conjunction serve the Physical Link X. Similarly, the Physical Link Terminal Equipment Y together with the Physical Link Couplers Y would together and in conjunction serve Physical Link Y. A well known example for a service with a Physical Layer serving different Physical Links concurrently is Floating Visual Aids Service, and buoys as the RSSs of that service often operate on various Physical Links simultaneously, namely on the visual link (colour and lights), the acoustic link (fog horn), the radar link (radar reflector and RACON), and on the AIS VHF Data Link (AIS AtoN station).

All different Physical Links served and/or all sectors of a RSS are *co-ordinated by the PSS alone and concurrently, and all data delivered to/from all different Physical Links served and/or to/from all sectors are processed by the PSS concurrently.* Hence, the PSS Layer is not subject to the sectorization, but it needs to take into account different Physical Links, if present. Also, in order to avoid ambiguity and discrepancies, the Service Management does interact directly with the PSS alone, and not with any component within the Physical Link Terminal Equipment and the Physical Link Couplers and within any sector.

The *Service Management Layer* is the layer which contains all functionality of own service to configure it, both from the requesting service of the CSS and from the Technical Operation Personnel, and manage its components at run-time. The Service Management Layer is represented at the Nodes and at the Remote Sites. Service components within the Service Management Layer are a *Service Management Master*, which contains all centralized management functionality and data storage and which resides at a Node, and *Service Management Agents* at a Node (*SM-Node*) and at a *Remote Shore Station* (*SM-RSS*).

Each layer comprises:

* *service component*, which provides the required functionality in terms of service-specific data processing;
* *supporting components and resources*, which are exclusively used by own service, such as computers and local networking devices, i.e. the so called service-owned infrastructure;
* (remote) *Human Machine Interfaces (HMI)* to allow for access to Technical Operation Personnel.

Each layer is supported by on-site infrastructure, such as energy supply, which may be shared with other services co-located on the same site.

Figure 7: Layered structure of own service with Physical Layer – Structure model

Technical Operation Personnel (master control)

**HMI**

**Logical Layer -**

**Logical Shore Station (LSS)**



**HMI**

**To/from CSS**

**Net Data**

**Status**

**Service  
Management Layer**

**HMI**

**Physical Shore Station (PSS)  
Layer**

**HMI**

**Physical Link Terminal Equipment (PLTE) Layer**

**HMI**

**Physical Link Couplers (PLC)  
Layer**

**Physical Link   
(= medium)**







**Technical Operation Personnel**(there are tasks for each layer in principle)

**Configuration**

**Physical  
Layer /  
Remote**

**Shore**

**Station**

\* Note: This symbolic representation does not imply any inference as to the amount of required personnel for the technical operation of the service. This symbolic representation aims at indicating that human interaction with the largely automated service is required as the last resort - and in some cases possibly on a regular basis - on all layers of the service.

**Boundary of CSS**

## Services without Physical Layer

Each and every service without a Physical Layer consists of two main functional layers as follows (refer to **Figure 8**):

* Logical Layer
* Service Management Layer

--- introduce description (only in terms of differences to above)

--- introduce simplified drawing without Physical Layer: **Figure 8 --- Placeholder ---**

# Resulting service setup and its component’s tasks

## Resulting service setup

So far, the most fundamental aspects for own service have been introduced such as its deliverables to the CSS, as well as its functional parts, the influence of geographical distribution, and its layered structure. These elements have been introduced largely independent of each other. This chapter aims at tying these elements together and provides some synthesis in general.

**Figure 9** (overleaf) shows a typical setup of a service with Physical Layer (for services without Physical Layer, simply drop the statements related to the Physical Layer and replace its role by the contribution of all of the *requested services* of the CSS). It is one possible translation of the more abstract layered structure depiction of own service in Figure 7 (see above) into one example setup using specific physical components on specific functional sites.

**Figure 9** shows a so-called One-Node Service Configuration setup.[[3]](#footnote-3) **Figure 9** shows the functional components, the functional site designations, where these components reside[[4]](#footnote-4), and also gives some indication regarding the interfacing of own service to its context within the CSS.[[5]](#footnote-5) **Figure 9** shows some important features of own service as follows:

* The *setup of the Remote Shore Station (RSS)*, i.e. the components internal to it, will be described in **Appendix 10.1** in more detail. Regarding **Figure 9** it is sufficient to note, that there is Physical Link Terminal Equipment (symbolized by antenna on roof top of RSS), that allows own service to access its Physical Link(s). The depiction of the RSS as a house also gives an indication of the relevance of on-site infrastructure consideration.[[6]](#footnote-6)
* *A Node site with the Logical Shore Stations* is shown, and how they are connected to the components of the other services of the CSS at the same Node site. As an (typical) example, particularly the User Interaction Service of the CSS is shown. The “other services” may be any other service of the CSS.
* The *components of the Service Management (SM)* that reside at the Node site are shown. (It should be noted, that components of the SM also reside at the Remote Shore Station; see **Appendices 10.1** and **11**).
* Also the *support for Technical Operation Personnel by Human Machine Interface(s) (HMI)* provided by own service is shown.
* Note the *concept of the functional sites* in combination with the *premises concept*.
* All components at the premise N where the Node site, the Technical Operation Personnel site and the User site reside in the example are connected with *LAN functional links* / LAN technology components. *WAN feeder links* will be used, when it comes to connecting the components of own service at the Node site with its Remote Shore Stations on the Remote sites.

*Cautionary note:* **Figure 9** should *not* be construed as normative or the only recommended way to set up own service. **Figure 9** *does* claim however to be correct and consistent with the normative statements in this Recommendation in regard to what it shows.

*Cautionary note:* While **Figure 9** describes one example of a workable service setup, it should be noted, that the available features and the available options in setting up own service can only precisely be captured by the more abstract descriptions provided in this Recommendation.

**Figure 9: Example of a service setup (One-Node Service Distribution Configuration within CSS context)**



## Resulting tasks of the functional components of own service

### Tasks of the functional components in the context of their interaction

The functional components can now (more) precisely be defined by *their individual set of tasks in the context of their interaction* with each other and with entities external to own service. The **Figure 10** (overleaf) shows the different tasks of the different functional components of own service in context.

Please note

* the different *task orientation* of functional components, i.e.
* the orientation of the Logical Layer towards the requesting / requested services of the CSS and their requirements;
* the orientation of the Physical Layer towards the Physical Link(s) served/employed by own service and its specific physical requirements;
* the need to have own-service-wide data storage which allows own-service-wide visibility and access of relevant data objects;
* the configuration and control capabilities of the Service Management (SM).
* that the *instances* are now implied, namely
* that the Physical Layer consists of *all* Remote Shore Stations (RSS) of the individual service under consideration as deployed by an individual administration;
* that the Logical Layer consists of *all* Logical Shore Stations (LSS) set up as individual software processes of the individual service under consideration as deployed by an individual administration;
* that there are necessary functionalities, such as own-service-wide data storage, which are ***in***visible to the net data flow between the Physical Link(s) and the requesting services, and vice versa. These functionalities are supported by infrastructure components owned by own service (so called service-owned infrastructure), namely by distributed data bases, local area networking, replicator devices, etc.

To specify the precise *features and* *attributes* of those functional components, such as data processing capabilities, interaction capabilities, interfacing requirements, timing constraints, and reliability figures, the appropriate service models need to be considered as introduced above.

These models have discussed own service on *class level*.[[7]](#footnote-7) In order to set up an individual service by an individual administration, it is necessary to eventually arrive at *specific* components which materialize at *specific* sites, i.e. at the *instance level*. This is where the Distribution model fits in.

All these different aspects need to be merged within the description of the functional components. This will be done in the appropriate Appendices, describing the particular functional component. In the following sections, a short introduction is provided.

**Figure 10: Overview over tasks of the functional components of own service**



This **Figure 10** will be used and explained in several Appendices which elaborate certain aspects of this figure.

### Mapping of service capabilities to component functionalities and component requirements

To finally arrive at the required functionalities and at the requirements for the components *specifically*, it is necessary to map the Basic Services (both external and internal) to the relevant individual components internal to own service. In accordance with the layered structure of own service the three main components to be considered for the mapping are Logical Shore Station (LSS), Physical Shore Station (PSS), and Service Management (SM).

In the case of services without Physical Layer there is no PSS. In the case of services with Physical Layer the Physical Link Terminal Equipment, the Physical Link Couplers (PLC), and also the Physical Link as such are omitted from the consideration *at this point* since they are transparent in regard to the net data flow. They will re-appear when considering secondary issues of own service, such as component management proper or management of the Physical Link.

As an example for the mapping in the case of a service with a Physical Layer, the **Figure 11** shows the *mapping* of an External Basic Service (Category E-1) *to the component functionalities* of the above components. As an example for the mapping in the case of a service without a Physical Layer, the **Figure 12** is given.

**Figure 11: Sample mapping of External BS (Cat. E-1) to component functionality in the case of service *with* at Physical Layer**



**Figure 12: Sample mapping of External BS (Cat. E-1) to component functionality in the case of service *without* at Physical Layer**



The required component functionalities of own service are collected in **Appendices 9, 10 and 11** of this document which deal with the functional descriptions of the respective components.

### Type approval considerations for PSS, LSS, and SM software modules

IMO, within its e-Navigation strategy [---Reference---], has formulated certain requirements regarding quality level of services offered by a competent shore authority:

--- Provide examples/references

Ultimately these requirements can only be achieved and validated by passing the procedure of type approval for the functional components involved. This section elaborates on the type approval aspect using the above introduction of the functionality descriptions for the PSS, the LSS, and the SM.

Regarding type approval the following notions should be taken into account:

* Type approval for software modules should *only apply to the service components* of the PSS, LSS, and SM software modules. Ideally these specific functionalities would be *plug-in software modules*. Type approval would *not* address or include service-owned infrastructure components, such as computers, LAN components etc.
* The description of the service components of PSS, LSS, and SM in **Appendices 9, 10**, and **11** would be *precise in terms of functionality* while ***not*** *being prescriptive regarding implementation* method by a manufacturer. Thus, the functional components PSS, LSS, and SM would be modelled as “black boxes”, rendering open interface definitions.
* The following aspects and the associated degree of impact should guide the selection of specific functionality of these functional components subject to type approval:

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Degree of impact** | |
| **HIGHER** | **LOWER** |
| Local impact of Physical Link(s) related run-time configurations (1) | **PSS** | **SM** |
| Coastal-wide impact of Physical Link(s) related run-time configuration (1) | **SM** (in particular SM-Master functionality) | **PSS** |
| Quality of data delivery to/from CSS | **LSS + PSS** | **SM** |

(1) for services with Physical Layer only, and only when Physical Layer has capability to influence Physical Link(s) as such.

# Interaction between services using the Structure Model

--- introduce in an overview manner (with Figures), how the Structure Model helps in understanding the interaction of different services of the CSS and how it shows, what layers are involved in what interaction and (eventually) what interfaces are needed to that end => Pointer to Interfacing Model…

# Overview description of the parts of the generic service engineering model

The previous chapters provided a solid introduction and overview. The individual parts of the generic service model need to be introduced in more detail to arrive a that precision level necessary for specific use by IALA, IALA membership and the supply industry at large for planning, procurement, deployment and operation of any specific service. There is an Appendix dedicated to one of the following parts each (compare List of Appendices). The following introduction also serves as an overview content definition for the Appendices.

The generic service engineering model is described by the following parts, which complement each other:

## Deliverables of a service to the CSS

**Basic Services Categories and Minimum Set of Basic Services**

Introduces the normative definitions of the categories of both External and Internal Basic Service Categories and introduces the Minimum Set of Basic Services, i.e. a comprehensive normative definition of Basic Services generally required for each service of the CSS. Refer to **Appendix 1** for further discussion.

**Data objects of own service and their properties (data model):**

Introduces abstract descriptions of the data provided by own service at its Basic Services level to the CSS, i.e. defines data objects and their properties of own service. Refer to **Appendix 2** for further discussion.

## Parts of the generic service model related to the layout and setup of own service

**Distribution model**

The Distribution model deals with issues arising from geographical distributed service layouts, including generic aspects of coverage planning as one major factor if appropriate. The Distribution model is given in **Appendix 3**.

**Interaction and Data Storage model**

The *Interaction model* introduces the dynamics of the interaction of the components within own service (data flow dynamics, such as push or pull concepts) and deals with associated data storage issues on a generic level.

The *Data Storage Model* deals with storage entities of own service. The need for data storage is due to the geographical distribution of own service components. The Data Storage Model deals with aspects like visibility of data objects, age of data objects, and rule base for data processing in own service, and consequences of geographically distributed data storage, such as the need for replication.

The Interaction Model and Data Storage models together are given in **Appendix 4**.

**Interfacing model**

The Interfacing model of own service describes the interfaces of own service to other services of the CSS *and* the interfaces between the components of own service, i.e. within own service. The following aspects are taken into account:

1. the application encoding methods; a portfolio of encoding techniques is recommended to suit the data objects of the Data Model;
2. the protocol stacks recommended for data transfer internal and external of own service.

All application encoding methods and protocol stacks recommended are in keeping with the open system philosophy of the CSSA. The Interfacing Model is given in **Appendix 5**

**Internal Time Latency model**

The Internal Time Latency Model of own service describes the *time latency limits* regarding the Basic Services (by external and internal BS Category). In particular the *resulting permissible latency* of data delivery of own service operating *within* the CSS is discussed. This is relevant, for example, when correlating time-critical data from different Data Collection and Data Transfer Services within the Value Added Data Processing Services of the CSS. From the timing requirements of the Basic Services, concrete timing requirements for components of own service and their mutual interconnection are derived. The Internal Time Latency Model is given in **Appendix 6.**

**Internal Reliability model**

The Internal Reliability Model of own service *analyses the internal reliability aspects* of own service and describes how to *measure internal reliability* during run-time. The Internal Reliability Model is based on the Structure Model, the Distribution Model and the Interaction Model. Details are given in **Appendix 7**.

**Test model**

The Test Model for own service:

* describes what are good test *strategies* for own service as a whole for Technical Operation Personnel, and
* provides guidance how to ascertain the proper function of own service after cold or warm start.

The Test Model does *not* generally deal with testing of individual components. Details are given in **Appendix 8.**

## Parts of the generic service model related to components of a service

**Generic functional description of the Logical Shore Station**

The *generic* functional description of the LSS describes the functionality needed on the Logical Layer to provide the External and Internal Basic Services, employing the concept of the Basic Services Category. In particular, this part contains the contribution of the LSS to the Minimum Set of Basic Services. For the description of an individual service, the *technology-specific* functionalities of the LSS of this individual service is contained in this part. See **Appendix 9**.

**Generic functional description of the Physical Layer as a whole – the Remote Shore Station**

The *generic* functional description of the Physical Layer *as a whole* describes how the different Sub-Layers of the Physical Layer interact to achieve the contribution of the Physical Layer to the External and Internal Basic Services. Also the additional components of own service and their interactions at the Remote site are introduced, thus rendering a comprehensive description of the *Remote Shore Station* of own service at a Remote site consideration. See **Appendix 10.1.**

**Generic functional description of the Physical Shore Station**

The Physical AIS Shore Station (PSS) performs the following main tasks:

* Pre-processing of data in one or both directions (receive and/or transmit), depending on configuration
* Control of the Physical Link Terminal Equipment Layer of the RSS depending on the configuration pre-set of the Service Management; this includes also control of the Sectors, if defined.

The PSS essentially is a software process running as application software on a computer as a physical entity or as an integral firmware of a dedicated unit, such as a smart switch.

Details are described in **Appendix 10.2.**

**Generic functional description of the Physical Link Terminal Equipment**

The Physical Link Terminal Equipment comprises all devices and/or functionality needed to – regarding “receiving“ - transform a signal received from the Physical Link Couplers into a data object as appropriate for the specific technology of own service. The data object is forwarded to the Physical Shore Station (PSS).

Regarding “transmission”, a data object received from the Physical Shore Station is transformed into an appropriate signal which is forwarded to the Physical Link Couplers for eventual transmission on the Physical Link. Examples of PLTE are radio transceiver devices in the case of radio based services or light generating/receiving devices in the case of visual services. Details are described in **Appendix 10.3.**

**Generic functional description of the Physical Link Couplers**

Regarding “transmission”, Physical Link Couplers are all devices and/or functionality needed to effectively couple a signal generated by the Physical Link Terminal Equipment into the appropriate Physical Link(s) and/or – regarding “receiving” – to extract signal(s) present in the Physical Link(s) and forward it/them to the Physical Link Terminal Equipment. Examples of Physical Link Couplers are radio antennas for radio wave signals or optical lenses for light signals. Details are described in **Appendix 10.4.**

**Generic functional description of the Service Management**

The Service Management of a competent authority performs the following tasks:

* It acts as a management entity for the whole of own service,
* it manages the Internal Basic Services, and
* it is the last resort regarding faulty behaviour.

Specifically, the SM:

* invokes, initialises, configures and terminates all the instances of LSS and PSS at run-time;
* determines the functional connections between PSS and their associated LSS for them to use during run-time;
* determines the functional connections between the LSS and the requesting services’ logical interface associated with them, i.e. this top level acts as a "switch-board" for the data exchange relationships between the different processes;
* provides the Human Machine Interface for the Technical Operation Personnel to monitor the current status of own service and potentially of the appropriate Physical Link(s) and configure them accordingly.

For most Service Distribution Configurations the SM consists of several distributed agents and of one centralized master functionality, all of which are functional sub-components of the SM and are generally software processes. There are the following functional sub-components:

* SM-Agent at the Remote Site, which represents the ASM at the Remote Shore Station of own service,
* SM-Agent at the Node site, representing the SM to the LSS,
* SM-Master (regularly only one SM master per service).

Details are described in **Appendix 11.**

## Parts of the generic service model related to installation and life-cycle management issues

**On-site infrastructure considerations and site co-location issues**

These two aspects are related:

* All the components of own service, i.e. the service components proper and the service-owned infrastructure components, have *requirements regarding the on-site infrastructure*, such as housing, protection against a variety of external influences, energy supply, source for a time base, and so forth. These requirements are discussed in a generic fashion in this part.
* Also, at one Remote Site of the CSS usually *several different services* of the CSS are present. For example, there are Remote Sites typically populated by a variety of radio services, such as AIS, VHF voice communication, and GMDSS DSC Services. Sharing on-site infrastructure may be a economical necessity in most cases. This results in site co-location issues, which are addressed in a generic fashion in this part.

Refer to **Appendix 12**.

**Recommendation regarding efficient operation and maintenance of a service**

In this part all generic aspects are discussed regarding the efficient operation by Technical Operation Personnel as well as the efficient maintenance and further development by Technical Maintenance/Development Personnel. By this part, the concept of own service is tied into the well established conceptual framework of technical operation and life-cycle management. Means to maintain and improve efficiency of service delivery over the full life-cycle are described here generically. Refer to **Appendix 13**.

**Runtime configuration management of a service**

This part addresses generic runtime configuration of the service by the Technical Operation Personnel. In this Recommendation the conceptual framework is provided, which is used by the individual descriptions of the technology-specific individual services. Refer to **Appendix 14**.

1. For further details in particular on the mechanisms refer to **Appendix 5**. [↑](#footnote-ref-1)
2. Note: “Service components” and “components of a service” is ***not*** the same, since the latter comprises the former plus the service-owned infrastructure components. [↑](#footnote-ref-2)
3. Refer to Distribution model in **Appendix 3** for further details [↑](#footnote-ref-3)
4. Refer again to the Distribution model, **Appendix 3**, for further explanation regarding the sites. [↑](#footnote-ref-4)
5. For further detail refer to the Interfacing Model, **Appendix 5**. [↑](#footnote-ref-5)
6. For details regarding the setup and installation of an RSS also in infrastructure terms refer to the Installation guidelines for RSS, **Appendix 12**. [↑](#footnote-ref-6)
7. The term “class” is used as defined by the object-oriented paradigm. [↑](#footnote-ref-7)