

*Document No:*

*Title: Ship-Port Information Service Specification*

*Date: 2016-11-16*



Authors

|  |  |
| --- | --- |
| **Name** | **Organisation** |
| Per Löfbom | SMA |
| Mikael Olofsson | SMA |

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Initials** | **Description** |
| Version 1.0 | 2016-05-16 | MO, PL, PD | Used in tendering documentation |
| Version 1.1 | 2016-10-25 | MO | + Updated service interface  + Changed name to Ship-Port Information Service (SPIS)  + Added possibility to push Port Call Message (PCM) to SPIS  + Added handling of Access Control List |
| Version 1.2 | 2016-11-01 | MO,PL | Update after internal review |
| Version 1.3 | 2016-11-14 | MO | Update after review   * Private interface Service Support interface findOrganisations changed name to findIdentities in harmony with SSC Input and output harmonised with SSC |
| Version 2.0 | 2016-11-16 | MO | Accepted by PMT |

Review

|  |  |
| --- | --- |
| **Name** | **Organisation** |
| Håkan Heurlin | SMA |
| Per Löfbom | SMA |
| Mikael Lind | Viktoria |
| Niklas Mellegård | Viktoria |
| Almir Zerem | Viktoria |
| Sandra Haraldson | Viktoria |
| Oliver Norkus | University of Oldenburg |
| PMT 2016 Nov 11 |  |

**Table of contents**

1 Introduction 5

1.1 Purpose of the document 5

1.2 Intended readership 5

1.3 Inputs from other projects 5

2 Service identification 6

3 Operational Context 7

3.1 Functional and non-functional Requirements 8

3.2 Other Constraints 9

3.2.1 Relevant Industrial Standards 9

3.3 Operational Nodes 10

3.4 Operational Activities 12

4 Service Overview 14

4.1 Service Interfaces 14

5 Service Data Model 17

5.1 Service Data Exchange Model 17

5.2 Service Data Exchange Model Service Support 17

5.2.1 callServiceRequestObject 19

5.2.2 callServiceResponseObj 19

5.2.3 requestType 19

5.2.4 findIdentitiesResponseObj 19

5.2.5 findServicesRequestObj 20

5.2.6 findServicesResponseObj 20

5.2.7 portCallMessage 21

5.2.8 Notification 21

5.2.9 responseObj 22

5.2.10 Message 22

5.2.11 MessageEnvelope 22

5.2.12 messageType 23

5.2.13 enumNotificationType 23

5.2.14 identityDescriptionObjects 23

5.2.15 responseIdObj 24

6 Service Interface Specification 25

6.1 Ship-Port Information Service 25

6.1.1 SPIS Public SeaSWIMInterface 25

6.1.2 SPIS Private Interface 25

6.1.3 SPIS Private ACL Interface 26

6.1.4 SPIS Private Service Support Interface 27

7 Service Dynamic Behaviour 29

7.1 Service State Model 29

7.1.1 SPIS Statemachine 30

7.2 Service Sequence diagrams 30

7.2.1 Interaction Private authorizeIdentities 30

7.2.2 Interaction Private publishMessage 31

7.2.3 Interaction Private Notify 31

7.2.4 Interaction Private getMessage 32

7.2.5 Interaction uploadPCM 32

7.2.6 Service Orchestration - <start-up> 33

7.2.7 Service orchestration - Poll PortCDM queue (Get RTA ) 33

7.2.8 Service Orchestration - Ship Port Synchronisation 34

7.3 Logging 35

7.3.1 Event Log 35

8 Service Provisioning 37

9 References 37

10 Acronyms and Terminology 38

11 Service Specification as XML 39

12 Document lifecycle 41

12.1 Maturity 41

12.2 Forecast 41

# Introduction

## Purpose of the document

The purpose of this service specification document is to provide a holistic overview of the Voyage Informationservice and its building blocks in a technology-independent way, according to the guidelines. 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 Information 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

## Intended readership

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

This document contains definition of the SeaSWIM interfaces but also information on the private interface.



## Inputs from other projects

No information.

# Service identification

|  |  |
| --- | --- |
| **Name** | Ship-Port Information Service |
| **ID** | urn:mrn:stm:service:specification:sma:spis |
| **Version** | 2.0 |
| **Description** | The service supports communication between ship and Port CDM. |
| **Keywords** | SPIS, Ship Port Information Service, STM Service, PCM,PCMF, Port Call Message |
| **Status** | provisional |

# Operational Context

The main purpose with Ship-Port Information Service is to support ship system with communication with PortCDM services in port. The service may be integrated in the onboard system or at shore, depending on the situation.

The main functionality of the service is to send updates on times at locations (e.g. PTA/TTA, ETA, ATA) and receive recommendations. The purpose is to support a collaborative decision making process with port.

The diagram below show the primary use case for the STM Validation project voyage management validation.



## Functional and non-functional Requirements

The following requirements are captured from the STM Validation project voyage management validation use cases. See references.



|  |  |
| --- | --- |
| **Requirement Id** | SPIS003 |
| Requirement Name | SPIS003 Send Port Call Message to PortCDM |
| Requirement Text | The ship shall send updated states to Port, such as Planned Time of Arrival (PTA) and Estimated Time of Arrival (ETA). |
| Rationale |  |
| Author | STM |
| Reference | Use Case and Requirements |

|  |  |
| --- | --- |
| **Requirement Id** | SPIS002 |
| Requirement Name | SPIS002 Receive Port Call Messages from PortCDM |
| Requirement Text | The ship shall receive recommendations from Port, such as Recommended Time of Arrival (RTA) |
| Rationale |  |
| Author | STM |
| Reference | Use Case and Requirements |

## Other Constraints

No information.

### Relevant Industrial Standards

No information.

## Operational Nodes

Operational nodes involved in the operational context around the service.



| **Operational Node/Activity** | **Remarks** |
| --- | --- |
| Ship | A floating structure designed for the transport of cargo and/or passengers.  The operational node represents a collection of services, activities and procedures of Ship. |
| Onboard system | The operational node represents a collection of services, activities and procedures of ship onboard system. |
| Port | A port is a location on a coast or shore containing one or more harbours where ships can dock and transfer people or cargo to or from land. Port locations are selected to optimize access to land and navigable water, for commercial demand, and for shelter from wind and waves.  The operational node represents a collection of services, activities and procedures of Port. |
| PortCDM | The operational node represents a collection of services, activities and procedures of PortCDM. |
| SeaSWIM | Maritime digital infrastructure to enable secure and interoperable exchange of information.  The operational node represents a collection of services, activities and procedures of SeaSWIM. |
| Ship Operator | A vessel operator is responsible for managing the vessels performances and works in close collaboration with the master onboard the vessel and with the charterer and ship-owner. A vessel operator plans several voyages ahead.  The operational node represents a collection of services, activities and procedures of Vessel Operator. |

## Operational Activities

Operational Activities supported by the service.



| **Operational Node/Activity** | **Remarks** |
| --- | --- |
| Port Call Synchronization | Ship and port synchronize their efforts in enabling just-in-time arrivals. From the ship’s point of view this means a possibility to save fuel by green steaming, in order to arrive just-in-time and enable just-in-time operations. |

# Service Overview

The Ship-Port Information Service has one private service side towards the STM Module, and one exposed service side on SeaSWIM.

STM Module in this context is a set of functionality that enables ship/ship operator to interact according to STM principals and technologies. The STM Module can be located onboard or onshore.

The Ship-Port Information Service is dependent on a SeaSWIM Connector that support with SeaSWIM specific functionality for authentication, and also supports with service and identity lookup in Service and Identity Registry.

The Ship-Port Information Service consumes PortCDM services for exchanging port call messages in Port Call Message format.

The interaction pattern (similar to the Voyage Information Service) is as follows:

* STM Module finds PortCDM identity for the arrival port related to the voyage
* STM Module defines or identifies an identity of the Port Call
* STM Module calls “authorizeIdentities” to establish a link to PortCDM
* STM Module then generates and publishes port call messages at certain intervals to SPIS which forwards them to PortCDM
* SPIS receives Port Call Messages and forwards them to STM Module, by

- SPIS checks queues in PortCDM at configurable intervals and forward messages to STM Module (pull)

- SPIS waits for uploaded port call message and forward message to STM Module (push)



## Service Public Interfaces (towards SeaSWIM)

| **Service Interface** | **Role** | **Service Operation** |
| --- | --- | --- |
| SPIS Public SeaSWIMInterface | Provided | uploadPCM |

## Service Private Interfaces (towards STM Module)

| **Service Interface** | **Role** | **Service Operation** |
| --- | --- | --- |
| SPIS Private Interface | Provided | publishMessage  getMessage |
| SPIS Private ACL Interface | Provided | authorizeIdentities  removeAuthorizedIdentitites  findAuthorizedIdentitites |
| SPIS Private Service Support Interface | Provided | findIdentities  findServices  callService |

## Consumed Service Interfaces

| **Service Interface** | **Role** | **Service Operation** |
| --- | --- | --- |
| SeaSWIM Connector Service Private Interface | Consumed | callService  findIdentities  findServices  identityServiceRequest  registryServiceRequest |
| PortCDM SeaSWIM Interface (port\_call\_finder) | Consumed | port\_call\_finder |
| STM Module Notify Interface | Consumed | Notify |
| PortCDM SeaSWIM Interface (state\_update) | Consumed | state\_update  state\_update\_queues  state\_update\_queues |

# Service Data Model

## Service Data Exchange Model

The data exchange model describes the data exchanged in the different service operations. The primary payload in this service is the Port Call Message described further in the referenced Port Call Message Format.



### portCallMessage

Port Call Message Format (PCMF). See references for more detailed information.

For details, see <http://stmvalidation.eu/schemas/>

### MessageEnvelope

Container for messages to STM Module

|  |  |
| --- | --- |
| **Element Name** | **Attributes** |
| **MessageEnvelope** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Type** | | **Description** | | numberOfMessages | int | | Total number of messages in this container | | | remainingNumberOfMessages | int | | Total number of remaining messages waiting to be fetched | | | message | Message | | Uploaded message(s) | | |

### <enumeration> messageType

Type of messages

|  |  |
| --- | --- |
| **Element Name** | **Enumeration Values** |
| **messageType** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Type** | | **Description** | | RTZ | string | | Route Exchange Format in XML | | | TXTMSG | string | | STM Textmessage in XML | | | S124 | string | | S124 in XML | | | PCM | string | | Port Call Message in XML | | |

### Message

Message to the STM Module

|  |  |
| --- | --- |
| **Element Name** | **Attributes** |
| **Message** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Type** | | **Description** | | id | URN | | Identity of the message | | | receivedAt | dateTime | | Date and time of reception | | | from | URN | | Identity of the message source | | | messageType | messageType | | Type of STM message | | | message | stmMessage | | The message of any STM format | | |

### Notification

To inside/private application, such as STM Module

|  |  |
| --- | --- |
| **Element Name** | **Attributes** |
| **Notification** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Type** | | **Description** | | dataId | URN | | Identity of the notification and the stored message which can be retrieved with "getMessage(dataId)", mandatory | | | notificationType | enumNotificationType | | Type of notification by enumeration | | | notificationCreatedAt | dateTime | | Notification created at date and time, mandatory | | | fromId | URN | | Source of notification and source for retrieving the complete message, mandatory, according to the STM MRN identifier. Example: urn:mrn:stm:org:<organization> | | | fromName | string | | Friendly name of sender for presentation | | | receivedAt | dateTime | | Date and time for the reception of the message. | | | messageWaiting | int | | >0 if a message is waiting in server, otherwise 0, mandatory | | | subject | string | | Notification subject, mandatory | | | body | string | | Notification body, optional | | |

### <enumeration> enumNotificationType

Types of notifications

|  |  |
| --- | --- |
| **Element Name** | **Enumeration Values** |
| **enumNotificationType** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Type** | | **Description** | | MESSAGE\_WAITING | int | | 1 | | | VOYAGEPLAN\_REQUESTED | int | | 2 | | | AUTHORIZATION\_REQUESTED | int | | 3 | | | ERROR\_MESSAGE | int | | 4 | | |

### responseObj

Generic response object

|  |  |
| --- | --- |
| **Element Name** | **Attributes** |
| **responseObj** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Type** | | **Description** | | statusCode | string | | 20x if OK  40x if NOK | | | body | string | | Error text if code is 40x | | |

### responseIdObj

Generic response object with id in return

|  |  |
| --- | --- |
| **Element Name** | **Attributes** |
| **responseIdObj** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Type** | | **Description** | | statusCode | string | | 20x if OK  40x if NOK | | | body | string | | Error text if code is 40x | | | dataId | URN | | Reference to data id in URN/MRN format, prefixed "urn:mrn" | | |

## Service Data Exchange Model Service Support

For full detailed description, please see SeaSWIM Connector Service Specification.



### callServiceRequestObject

Contain the data needed by the callService function to execute the request. The parameters contained in the callServiceRequestObj are the general parameters contained in a standard HTTP request.

|  |  |
| --- | --- |
| **Element Name** | **Attributes** |
| **callServiceRequestObject** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Type** | | **Description** | | requestType | requestType | | Type of request according to enumeration | | | endpoint\_method | URI | | URL contains <ipaddress>:<port>/<request> | | | body | string | | Request body needed by the requested service | | | headers | string | | Request headers needed by the requested service | | |

### callServiceResponseObj

This object is a response container with response from called service

|  |  |
| --- | --- |
| **Element Name** | **Attributes** |
| **callServiceResponseObj** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Type** | | **Description** | | statusCode | int | | HTTP Status codes | | | body | string | | Request Body | | |

### <enumeration> requestType

Type of request to be used in callService

|  |  |
| --- | --- |
| **Element Name** | **Enumeration Values** |
| **requestType** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Type** | | **Description** | | GET | int | |  | | | POST | int | |  | | | DELETE | int | |  | | | PATCH | int | |  | | |

### findIdentitiesResponseObj

Contains the result from search in Identity Registry.

**TBD**

|  |  |
| --- | --- |
| **Element Name** | **Attributes** |
| **findIdentitiesResponseObj** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Type** | | **Description** | | statusCode | int | | HTTP Status codes | | | body | string | | Response from Identity Registry | | |

### findServicesRequestObj

Contains parameters for search in Service Registry

|  |  |
| --- | --- |
| **Element Name** | **Attributes** |
| **findServicesRequestObj** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Type** | | **Description** | | keyword | string | | Example: **#VIS**# | | | organisationId | URN | | Service provider id in MRN format | | | coverageArea | string | | UnLoCode,wkt,Location,geoJson | | | serviceDesignId | URN | | Service Design Id in MRN format | | | serviceId | URN | | Service id in MRN format | | |

### findServicesResponseObj

Contains the result from search in Identity Registry.

**TBD**

|  |  |
| --- | --- |
| **Element Name** | **Attributes** |
| **findServicesResponseObj** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Type** | | **Description** | | statusCode | int | | HTTP Status codes | | | body | string | | Following the Service Registry Specification | | |

### identityDescriptionObjects

|  |  |
| --- | --- |
| **Element Name** | **Attributes** |
| **identityDescriptionObjects** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Name** | | **Type** | | **Description** | | identityID | URN | | Identity in MRN format that related to an identity in Identity Registry.  The identity must also be possible to link to services of specific kinds, such as PortCDM Port synchronisation service in order to find endpoint to send message to. | | | identityName | string | | Human readable name of the identity | | |

# Service Interface Specification

## Ship-Port Information Service

The Ship-Port Information Service provides a private interface where informing the port regarding voyage plan schedule (PTA, ETA etc.) and receiving recommendations from port (RTA etc.).

### SPIS Public SeaSWIMInterface

The service provides an interface on SeaSWIM for future use to receive pushed state updates such as RTA (Recommended Time of Arrival).

##### uploadPCM()

Receives state updates in PCM (Port Call Message) format, such as RTA (Recommended Time of Arrival).

**Operation functionality**

Received PCM message is validated and stored in cache

STM Module is notified and then retrieves the message with getMessage

**Operation Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter Name** | **Direction** | **Data Type** | **Description** |
| portCallMessage | Input | portCallMessage |  |
| deliveryAckEndpoint | Input | DeliveryAck |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Direction** | **Data Type** | **Description** |
|  | Return | responseObj |  |

### SPIS Private Interface

Facilitates private operations towards ship and ship operator.



##### publishMessage()

Enables publishing Port Call message in PCM format to SPIS, which will store and forward it to subscribers set by authorizeIdentitites.

When a publishMessage request is received, the message payload is validated and subsequently stored.

Endpoints for subscribers are read from the subscription table and the SSC exposed interface for consuming a service, callService is invoked. Hereby passing the relevant payload received in the STMMessage together with endpoint (URI) for the called service found in the subscription table.

In case the schema validation fails a message informing the caller of an invalid format is returned.

**Operation functionality**

Validates message towards schema

Forwards the message to subscriber(s) according to subscription parameters and authorization (Access Control List)

**Operation Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter Name** | **Direction** | **Data Type** | **Description** |
| messageId | Input | URN |  |
| messageType | Input | messageType |  |
| message | Input | stmMessage |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Direction** | **Data Type** | **Description** |
|  | Return | responseObj |  |

##### getMessage()

Enables the STM Module to retrieve messages waiting in SPIS. The messages in SPIS is in PCM format. A limit can be given which allows the STM Module to set maximum number of messages in the answer.

**Operation functionality**

All messages that has not been fetched is sent as response limited by the parameter.

All messages sent is flagged as fetched with timestamp

The response contains information of remaining number of messages that are waiting.

**Operation Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter Name** | **Direction** | **Data Type** | **Description** |
| limitQuery | Input | int | max number of wanted messages in response |
| dataId | Input | string | dataId given in Notification message |

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Direction** | **Data Type** | **Description** |
|  | Return | MessageEnvelope |  |

### SPIS Private ACL Interface

The private interface in SPIS is not mandatory and shall be treated as information only. The private interface regards only this project SPIS.

Facilitates private operations towards ship onboard system.

##### authorizeIdentities()

Give access (authorization) for a specific information object to a set of identitites.

The information object is identified by dataID, where dataId can be a UVID, UPCID or local ID.

The dataId must match the dataId used in publishMessage.

Authorized identities is added to subscription list (send list).

The function should be called in collaboration with publishMessage.

**Operation functionality**

Create a POSIX Access Control List of the given identities and data object id

**Operation Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter Name** | **Direction** | **Data Type** | **Description** |
| dataId | Input | string | Identity to the information object that the authorization shall apply to. |
| identityDescriptionObjects | Input | identityDescriptionObjects | A complete set of identityDescriptionObjects that each describe the identity that shall have access to the information object. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Direction** | **Data Type** | **Description** |
|  | Return | responseObj |  |

##### removeAuthorizedIdentitites()

Removes a set of identitites from Access Control List for dataId.

**Operation functionality**

Remove identities from ACL for a given data object id

**Operation Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter Name** | **Direction** | **Data Type** | **Description** |
| dataId | Input | string |  |
| identities | Input | identityDescriptionObjects |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Direction** | **Data Type** | **Description** |
|  | Return | responseObj |  |

##### findAuthorizedIdentitites()

Returns the set of identities that are authorized to dataId.

**Operation functionality**

Return list of identities for a given data object id

**Operation Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter Name** | **Direction** | **Data Type** | **Description** |
| dataId | Input | string |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Direction** | **Data Type** | **Description** |
|  | Return | identityDescriptionObjects |  |

### SPIS Private Service Support Interface

The private interface regards only this STM Validation project SPIS.

##### findIdentities()

Propagates to SeaSWIM Connector Service operation findOrganizations.

Find identities in Identity Registry to be used in e.g. for authorization.

Returns a list of all identities discoverable in Identity Registry.

**Operation functionality**

Call SeaSWIM Connector and return the response

**Operation Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter Name** | **Direction** | **Data Type** | **Description** |
| findIdentitiesRequestObj | Input | findIdentitiesRequestObj |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Direction** | **Data Type** | **Description** |
|  | Return | findIdentitiesResponseObj |  |

##### findServices()

Propagates to SeaSWIM Connector service operation findService.

Facilitates service discovery in the central SeaSWIM service registry using query service parameters like service type, service category, location, service id etc.

The response is a list of endpoints and corresponding service descriptions.

**Operation functionality**

Call SeaSWIM Connector and return the response

**Operation Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter Name** | **Direction** | **Data Type** | **Description** |
| findServiceParameters | Input | findServicesRequestObj |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Direction** | **Data Type** | **Description** |
|  | Return | findServicesResponseObj |  |

##### callService()

Propagates to SeaSWIM Connector service operation callService.

Supports consumption of other information services assisting with authentication and secure transfer.

Invokes a service instance and returns the result. SPIS is transparent and does not check input or return.

**Operation functionality**

Call SeaSWIM Connector and return the response

**Operation Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter Name** | **Direction** | **Data Type** | **Description** |
| callServiceObj | Input | callServiceRequestObject |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Return** | **Direction** | **Data Type** | **Description** |
|  | Return | callServiceResponseObj |  |

# Service Dynamic Behaviour

The diagram describes the expected internal functionality of the Ship-Port Information Service based on present decisions.



## Service Sequence diagrams

This section contains interaction diagrams (sequence diagrams) to show the interactions to SPIS interface.



### Interaction Private authorizeIdentities

The STM Module (private consumer) authorizes a dataId (created by STM Module or port) to a set of identities that shall have access to the published information.

SPIS creates an Access Control List (ACL) and searches for PortCDM service endpoints based on authorized identities. Each PortCDM endpoint is stored in a subscription list that is used when publishing PCM messages.

SPIS also creates queues in PortCDM (if required) and initiates poll of data with configurable interval.



### Interaction Private publishMessage

The STM Module (private consumer) publishes messages in PCM format (Port Call Message) with same dataId as was used in authorizedIdentities. The PCM messages is forwarded by SPIS to the correct PortCDM service and returns the response.



### Interaction Private Notify

When a message is received in SPIS, either retrieved from polling a queue or pushed to SPIS, the message generates a notification which is sent to the STM Module.

If there are messages waiting in SPIS, the STM Module calls getMessage to retrieve waiting messages.



### Interaction Private getMessage

The private interface getMessage can be called any time by the STM Module. Normally it's related to a received Notification but can also be called at start-up or re-connection to purge messages from SPIS.

The call to getMessage can contain a upper boundary to limit the number of messages returned, and it can contain a specific identity given by a notification message to retrieve exactly one message.



### Interaction uploadPCM

Whenever a new e.g. RTA is created in PortCDM based on activities in the port community the exposed SeaSWIM interface, uploadPCM, is invoked. A PCM message is sent to SPIS stored in cache and forwarded to the ship. The sending of a PCM message to the ship is initiated by a notification sent to the ship followed by subsequent invocation of SPIS private interface getMessage, should the onboard operator running the STM Module choose to do so.

The producer may request an acknowledgement when message has been retrieved by the STM Module.



### Service Orchestration - <start-up>

Shows the service interactions at start-up



### Service orchestration - Poll PortCDM queue (Get RTA )

Shows the interactions when a timer event is fired and the PortCDM queues is polled. Messages in PortCDM queue will be forwarded to the ship.



### Service Orchestration - Ship Port Synchronisation

The diagram outlines the overall sequence of initiation of a port call from ship and following distribution of Port Call Messages. Incoming messages is shown both from poll of queue and pushed incoming messages on upload interface.



## Logging

Logging in the service is required for validation purposes to enable analysis of data in order to assess the STM Concept.

### Event Log

The following events are proposed to generate a log:

* Messages in and out of the service
* Failure events (Schema validation failure, Service operation failure)
* Authorization events

|  |  |
| --- | --- |
| **Incoming service call on SeaSWIM side** | |
| **Event** | **Log description** |
| uploadPCM | Log event for incoming data  Log event with notification to STM Module |
| **Incoming service call on private side** | |
| **Event** | **Log description** |
| authorizeIdentities | Log event for incoming request  Log event for added row in ACL  Log event for added actor in subscription list |
| removeAuthorizedIdentitites | Log event for incoming request  Log event for removed row in ACL  Log event for removed actor in subscription list |
| findAuthorizedIdentitties | Log event for incoming request |
| publishMessage | Log event for incoming data  Log event for outgoing data |
| getMessage | Log event for incoming request  Log event with returned data |
| callService | Log event for incoming request  Log event with returned data |
| findOrganizations | Log event for incoming request  Log event with returned data |
| findServices | Log event for incoming request  Log event with returned data |
| **Internal events** | |
| **Event** | **Log description** |
| Start-up | Log event start-up |
| Close-down | Log event close-down |

# Service Provisioning

The diagram describes the relationships between the STM Module, the Ship-Port Information Service, the corresponding SeaSWIM Connector and PortCDM.

The STM Module may be located at another site than the Ship-Port Information Service.

The SeaSWIM Connector should always be co-located with the Ship-Port Information Service.

PortCDM services are assumed to be registered in Service Registry and correspond to SeaSWIM Connector.



# References

|  |  |  |
| --- | --- | --- |
| **Reference name** | **Comment** | **Link** |
| SeaSWIM Technical Documentation |  |  |

# Acronyms and Terminology

## Acronyms

|  |  |
| --- | --- |
| **Term** | **Definition** |
| SSC | SeaSWIM Connector |
| URN | Uniform Resource Locator |
| UVID | Unique Voyage Identity |
| VIS | Voyage Information Service |
| VP | Voyage Plan |
| XML | Extendible Mark-up Language |
| XSD | XML Schema Definition |

## Terminology

|  |  |  |
| --- | --- | --- |
| **Term** | **Acronym** | **Definition** |
| 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, …  *Source*  *E2 D3.4 Service Documentation Guidelines*  *v01.01* |
| Operational Model |  | A structure of operational nodes and associated operational activities and their inter-relations in a process model.  *Source*  *E2 D3.4 Service Documentation Guidelines*  *v01.01* |
| Operational Node |  | A logical entity that performs activities. Note: nodes are specified independently of any physical realisation.  Examples of operational nodes in the maritime context are: Maritime Control Center, Maritime Authority, Ship, Port, Weather Information Provider, …  *Source*  *E2 D3.4 Service Documentation Guidelines*  *v01.01* |
| 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.  *Source*  *E2 D3.4 Service Documentation Guidelines*  *v01.01* |
| 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.  *Source*  *E2 D3.4 Service Documentation Guidelines*  *v01.01* |
| 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.  *Source*  *E2 D3.4 Service Documentation Guidelines*  *v01.01* |
| 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.  *Source*  *E2 D3.4 Service Documentation Guidelines*  *v01.01* |
| Service Instance Model |  | Describes the implementation of a dedicated service instance in a dedicated technology. This includes a detailed description of the data payload to be exchanged by this service instance. The actual format of the service instance model depends on the chosen technology. Examples may be WSDL and XSD files (e.g., for SOAP services) or swagger (Open API) specifications (e.g., for REST services). If an external data model exists (e.g., a standard data model), then the service instance model shall refer to it: each data item of the service instance model shall be mapped to a data item defined in the external data model.  In order to prove correct implementation of the service specification, there shall exist a mapping between the service instance model and the service data model. This means, each data item used in the service instance model shall be mapped to a corresponding data item of the service data model. (In case of existing mappings to a common external (standard) data model from both the service data model and the service instance model, such a mapping is implicitly given.)  *Source* |
| 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.  *Source*  *E2 D3.4 Service Documentation Guidelines*  *v01.01* |
| 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.  *Source*  *E2 D3.4 Service Documentation Guidelines*  *v01.01* |
| Service Physical Data Model |  | Describes the realisation of a dedicated service data model in a dedicated technology. This includes a detailed description of the data payload to be exchanged using the chosen technology. The actual format of the service physical data model depends on the chosen technology. Examples may be WSDL and XSD files (e.g., for SOAP services) or swagger (Open API) specifications (e.g., for REST services). If an external data model exists (e.g., a standard data model), then the service physical data model shall refer to it: each data item of the service physical data model shall be mapped to a data item defined in the external data model.  In order to prove correct implementation of the service specification, there shall exist a mapping between the service physical data model and the service data model. This means, each data item used in the service physical data model shall be mapped to a corresponding data item of the service data model. (In case of existing mappings to a common external (standard) data model from both the service data model and the service physical data model, such a mapping is implicitly given.)  *Source*  *E2 D3.4 Service Documentation Guidelines*  *v01.01* |
| Service Specification Producer |  | Producers of service specifications in accordance with the service documentation guidelines.  *Source*  *E2 D3.4 Service Documentation Guidelines*  *v01.01* |
| Authentication |  | The process of verifying the identity claimed by an entity based on its credentials.  *Source*  *developers.maritimecloud.net*  *2016-11-11* |
| Service Specification |  | Describes one dedicated service at logical level. The Service Specification is technology-agnostic. The Service Specification includes (but is not limited to) a description of the Service Interfaces and Service Operations with their data payload. The data payload description may be formally defined by a Service Data Model.  *Source*  *E2 D3.4 Service Documentation Guidelines*  *v01.01* |
| 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.  *Source*  *E2 D3.4 Service Documentation Guidelines*  *v01.01* |
| Service Implementation |  | The provider side implementation of a dedicated service technical design (i.e., implementation of a dedicated service in a dedicated technology).  *Source*  *E2 D3.4 Service Documentation Guidelines*  *v01.01* |
| Service Instance |  | One service implementation may be deployed at several places by same or different service providers; each such deployment represents a different service instance, being accessible via different URLs.  *Source*  *E2 D3.4 Service Documentation Guidelines*  *v01.01* |
| Service Endpoint |  | A Service Endpoint is the URL where your service can be accessed by a client application. The same web service can have multiple endpoints, for example in order to make it available using different protocols.  *Source*  *http://stackoverflow.com/questions/9807382/what-is-a-web-service-endpoint* |
| Service Interface |  | The communication mechanism of the service, i.e., interaction mechanism between service provider and service consumer. A service interface is characterised by a message exchange pattern and consists of service operations that are either allocated to the provider or the consumer of the service.  *Source*  *E2 D3.4 Service Documenation Guidelines*  *v01.01* |
| Service Operation |  | Functions or procedure which enables programmatic communication with a service via a service interface.  *Source*  *E2 D3.4 Service Documentation Guidelines*  *v01.01* |
| Service Parameters |  | Service Parameters are input to a Service Operation and can be described formally in a data exchange model as e.g. XML Schemas.  *Source*  *MO* |
| Service Response |  | Service Response are output from a Service Operation and can be described formally in a data exchange model as e.g. XML Schemas.  *Source*  *MO* |
| Authentication |  | Authentication is the process of determining whether someone or something is, in fact, who or what it is declared to be.  *Source*  *http://searchsecurity.techtarget.com/definition/authentication* |
| Authorization |  | Authorization is the process of giving someone permission to do or have something.  *Source*  *http://searchsoftwarequality.techtarget.com/definition/authorization* |
| 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.  *Source*  *E2 D3.4 Service Documentation Guidelines*  *v01.01* |
| 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.  *Source*  *E2 D3.4 Service Documentation Guidelines*  *v01.01* |
| Proxy Service |  | A proxy service is an intermediary role played by software or a dedicated computer system between an endpoint device and a client which is requesting the service. The proxy service may exist on the same machine or on a separate server. The proxy service enables the client to connect to a different server and provides easy access to services like Web pages, connections or files.  *Source*  *https://www.techopedia.com/definition/31705/proxy-service* |
| Service Request |  | *Source* |





# APPENDIX Service Specification as XML

<?xml version="1.0" encoding="UTF-8"?>

<ServiceSpecificationSchema:serviceSpecification xmlns:ServiceSpecificationSchema="http://efficiensea2.org/maritime-cloud/service-registry/v1/ServiceSpecificationSchema.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://efficiensea2.org/maritime-cloud/service-registry/v1/ServiceSpecificationSchema.xsd ServiceSpecificationSchema.xsd ">

<name>Ship-Port Information Service</name>

<id>urn:mrn:stm:service:specification:sma:spis</id>

<version>2.0</version>

<status>provisional</status>

<description>The service supports communication between ship and Port CDM.</description>

<keywords>SPIS, Ship-Port Information Service, STM Service, PCM,PCMF, Port Call Message</keywords>

<isSpatialExclusive>false</isSpatialExclusive>

<authorInfos>

<authorInfo>

<id>urn:mrn:stm:org:sma:pelo</id>

<name>Per Löfbom</name>

<description>Solution architect</description>

<contactInfo>per.lofbom@sjofartsverket.se</contactInfo>

</authorInfo>

</authorInfos>

<requirements>

<requirement>

<id>SPIS003</id>

<name>SPIS003 Send Port Call Message to PortCDM</name>

<text>The ship shall send updated states to Port, such as Planned Time of Arrival (PTA) and Estimated Time of Arrival (ETA).

</text>

<rationale></rationale>

<reference>Use Case and Requirements</reference>

<authorInfos>

<authorInfo>

<id>000</id>

<name>STM</name>

<description>text</description>

<contactInfo>mail</contactInfo>

</authorInfo>

</authorInfos>

</requirement>

<requirement>

<id>SPIS002</id>

<name>SPIS002 Receive Port Call Messages from PortCDM</name>

<text>The ship shall receive recommendations from Port, such as Recommended Time of Arrival (RTA)

</text>

<rationale></rationale>

<reference>Use Case and Requirements</reference>

<authorInfos>

<authorInfo>

<id>000</id>

<name>STM</name>

<description>text</description>

<contactInfo>mail</contactInfo>

</authorInfo>

</authorInfos>

</requirement>

</requirements>

<serviceDataModel>

<definitionAsXSD>

</definitionAsXSD>

</serviceDataModel>

<serviceInterfaces>

<serviceInterface>

<name>SPIS Public SeaSWIMInterface</name>

<description>The service provides an interface on SeaSWIM for future use to receive pushed state updates such as RTA (Recommended Time of Arrival).

</description>

<dataExchangePattern></dataExchangePattern>

<operations>

<operation>

<name>uploadPCM</name>

<description>Receives state updates in PCM (Port Call Message) format, such as RTA (Recommended Time of Arrival).

</description>

<returnValueType>

<typeReference>responseObj</typeReference>

</returnValueType>

<parameterTypes>

<parameterType>

<typeReference>portCallMessage</typeReference>

</parameterType>

<parameterType>

<typeReference>DeliveryAck</typeReference>

</parameterType>

</parameterTypes>

</operation>

</operations>

</serviceInterface>

</serviceInterfaces>

</ServiceSpecificationSchema:serviceSpecification>

# Document lifecycle

## Maturity

The document describes a feature for future use to enable push from a producer of port call messages to a consumer. The interface may need to be revised when such service producers have been designed.

## Forecast

The foreseen updates on the document is the following:

|  |  |  |
| --- | --- | --- |
| **Chapter** | **Rationale** | **Time and version** |
|  |  |  |
|  | The service consumes PortCDM services and may need to be revised if and when PortCDM changes.  The state\_update and port\_call\_finder interfaces will be changed. |  |
|  |  |  |