INTERNATIONAL MARITIME ORGANIZATION

INTERNATIONAL HYDROGRAPHIC ORGANIZATION

E

IMO/IHO HARMONIZATION GROUP ON DATA MODELLING
Agenda item 5

HGDM 2/5
26 September 2018
ENGLISH ONLY

CONSIDERATION OF DESCRIPTIONS OF MARITIME SERVICES FROM DOMAIN COORDINATING BODIES

Best practice for writing guidelines for a maritime service – experiences from the STM project

Submitted by Sweden

SUMMARY	
Executive summary:	This document provides information regarding the work carried out in the Sea Traffic Management (STM) project, which can act as a best practice for writing guidelines for Maritime Services. STM connects and updates the maritime world in real time, creating an efficient information exchange among parties involved, such as ships, port actors, service providers and shipping companies. The different levels of service descriptions for a Maritime Service that were developed within STM has proved to create vendor independent interoperable information exchange and could act as a best practice for the further development of maritime services within e-navigation
Action to be taken:	Paragraph 17
Related documents:	MSC 99/22, MSC 98/23, MSC 94/21; MSC.1/Circ.1595; NCSR 5/8, NCSR 5/8/3, NCSR 5/23 and HGDM 1/5

Background

1 MSC 98 agreed to activate the IMO/IHO HGDM to work on the output "Develop guidance on definition and harmonization of the format and structure of MSPs". NCSR 5 supported the work done by HGDM 1 and MSC 99 approved a second meeting of HGDM (HGDM 2) to finalize the guidelines.

2 Maritime transport is a global business that, to a great extent, relies on an international framework of regulations and technical standards. It's only through global cooperation, commitment, standards and joint coordinated action that global shipping can be taken from a random kind of activity into a more organized form of integrated transport system with the help of digitalization and information sharing. In that process, we can make maritime transport more energy efficient, cost effective and, at the same time, avoid ship collisions and groundings, including the negative impact on the marine environment, and save lives. This is the objective of Sea Traffic Management (STM).

3 STM is in line with the vision and strategic directions of IMO and the E-navigation Strategy Implementation Plan (SIP) - Update 1 (MSC.1/Circ.1595). STM addresses certain aspects of four of the five prioritized e-navigation solutions, namely, S2 (means for standardized and automated reporting), S3 (Improved reliability, resilience and integrity of bridge equipment and navigation information), S4 (integration and presentation of available information in graphical displays received via communication equipment) and S5 (improved communication of VTS Service Portfolio (not limited to VTS stations)).

4 STM-services allow personnel on-board and on shore to make decisions based on real-time information and intentions. Example of services that are being tested and that are using the STM maritime digital service infrastructure are:

- Route optimization services;
- Ship to ship route exchange;
- Enhanced monitoring;
- Port call synchronization;
- Winter navigation;
- Baltic navigational warning service;
- Pilot route service; and
- Search and rescue.

5 The STM Validation Project is a major e-navigation project co-founded by the European Union and coordinated by the Swedish Maritime Administration. STM has more than 50 partners and is currently finishing its validation phase with 300 ships, 13 ports and 5 shore centers (Vessel Traffic Center, Joint Rescue Coordination Center and Fleet Operating Centers) equipped with STM functionality exchanging information.

6 Some of the leading manufacturers of systems (Transas, SAAB, Wärtsilä, Airbus, Kongsberg, Furuno and others) have, within the STM Validation Project, updated their systems to allow interoperability in the information exchange ship-to-ship and ship-to-shore.

Introduction

7 STM has created a vendor independent maritime digital service infrastructure, enabling seamless interoperability between different systems (ECDIS, VTS and other) and has served as a testbed for the evolving S124 standard encompassing Navigational Warnings. A testbed report of the initial findings of STM will be submitted to NCSR 6.

8 The STM project has produced the Route Plan Exchange Format (RTZ), which is included in IEC standard 61174 ed. 4. STM members are also participating in IEC WG 17 who transforms RTZ to an S-100 compatible standard: the "S-421 route plan based on S-100". STM is also working with a port call format together with IALA for an S-211 standard.

9 STM has been working with IALA to write a Maritime Service description based on the guideline produced under HGDM 1 and later noted by NCSR 5. STM also wrote, based on the IALA Guideline on Specification of e-navigation technical services, an implementation of a Maritime Service with help of the draft guideline to validate the ongoing work for guidance in describing implementation in detail.

Conclusions

10 Producing the above mentioned guidelines is not easy. It requires skill in software architecture and may be done in different ways. Therefore, Sweden suggests to use the results of the work done within the STM Validation project as an example on how descriptions can be expressed for a particular Maritime Service under the guidelines set up by IMO.

11 The three attached annexes provide a description of how to implement a voyage information service for sending and receiving route information. This can, for example, be a weather service that the onboard system uses to do weather routing or it could be used for exchanging the route information with other ships in close navigational situations.

12 The STM concept and ship-to-ship and ship-to-shore information exchange, based on the STM digital maritime service infrastructure currently being tested in the STM Validation Project with positive results. A number of operational services (in test version and in production) have been prepared within the STM and been made available to ships with STM capability.

Experiences from STM to be used as best practices

13 Annex 1 provides a description for a Voyage Information Service specification in accordance with the draft IALA Guideline G1128 (*The Specification of e-Navigation Technical Services*) and presented at HGDM 1 (HGDM 1/5/2) and NCSR 5 (NCSR 5/8/3). The first version of this guideline was approved by IALA Council 67 and G1128 is a new draft based on that guideline.

14 Annex 2 contains a description for a Voyage Information Service technical design specification in accordance with IALA Guideline G1128.

15 Annex 3 presents a description for a Voyage Information Service technical instance description in accordance with IALA Guideline G1128.

16 Annexes 1, 2 and 3 could act as a best practice when writing service specifications for a Maritime Service description.

Action requested of the HGDM

17 The HGDM is invited to:

.1 note the information given in this document, as well as the guidance in annexes 1, 2 and 3;

.2 consider annexes 1, 2 and 3 as a best practice of a Maritime Service description and its different levels; and

.3 take any other actions if considered appropriate.

***

ANNEX 1

Voyage Information Service Specification

Contents

Table of figures

List of tables

1 Introduction

1.1 Purpose of the Document

The purpose of this service specification document is to provide a holistic overview of the Voyage Information service and its building blocks in a technology-independent way, according to the guidelines given in [1]. It describes a well-defined baseline of the service by clearly identifying the service version.

The aim is to document the key aspects of the Voyage 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

1.2 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 Voyage 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 specification of the service in focus.

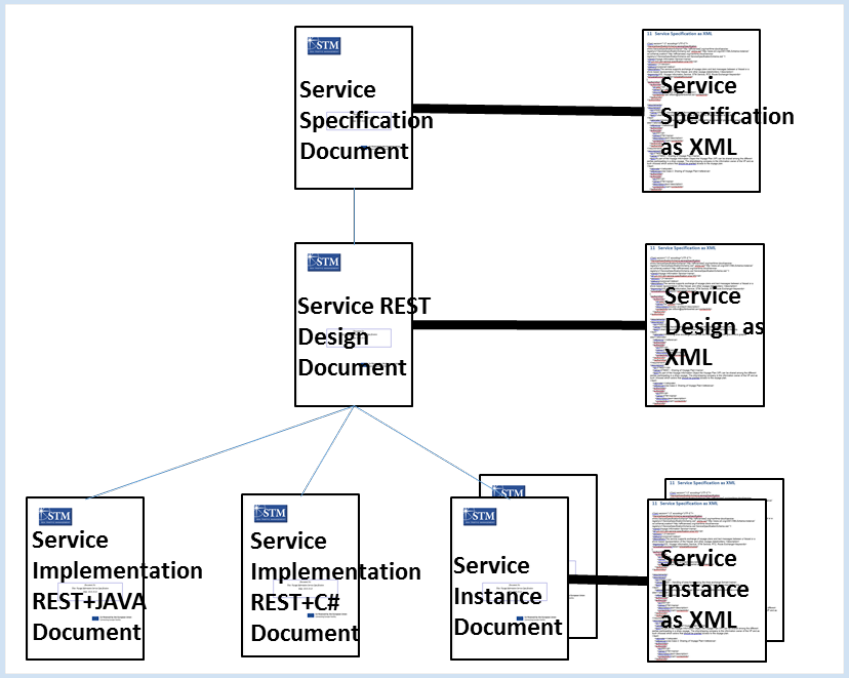


Figure 1 Overview of service descriptions

1.3 Inputs from Other Projects

No Information.

2 Service Identification

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

Name	*Voyage Information Service*
ID	*urn:mrn:stm:service:specification:sma:vis*
Version	*2.2*
Description	*The service supports exchange of voyage plans, text messages and area messages.*
Keywords	*VIS, Voyage Information Service, STM Service, RTZ, Route Exchange, Area, S-124, TXT, Text Message*
Architect(s)	Per Löfbom, SMA *Mikael Olofsson, SMA (Combitech)*
Status	*released*

3 Operational Context

The main purpose with Voyage Information Service is to support sharing of voyage plans to authorized actors. Sharing of voyage plan is primarily initiated by the ship by authorizing the voyage plan to concerned actors and by direct accessing e.g. route optimization or route check, but sharing can also be on request by other service providers such as enhanced monitoring.

The Voyage Information Service can be used both to support exchange of voyage plans from ship as well as other service providers and consumers such as shore centers and route optimization providers.

The Voyage Information Service is specified in such way that by using VIS on all consumers and providers that intend to share/exchange voyage plans, interoperability can be reached. That enables new services to arise in Service Registry based on VIS Design for voyage plan exchange to be used without new implementation on consumer side.

Each Voyage Plan shall refer to a UVID (Unique Voyage Identity) generated by the service provider and contain status on the voyage/route.

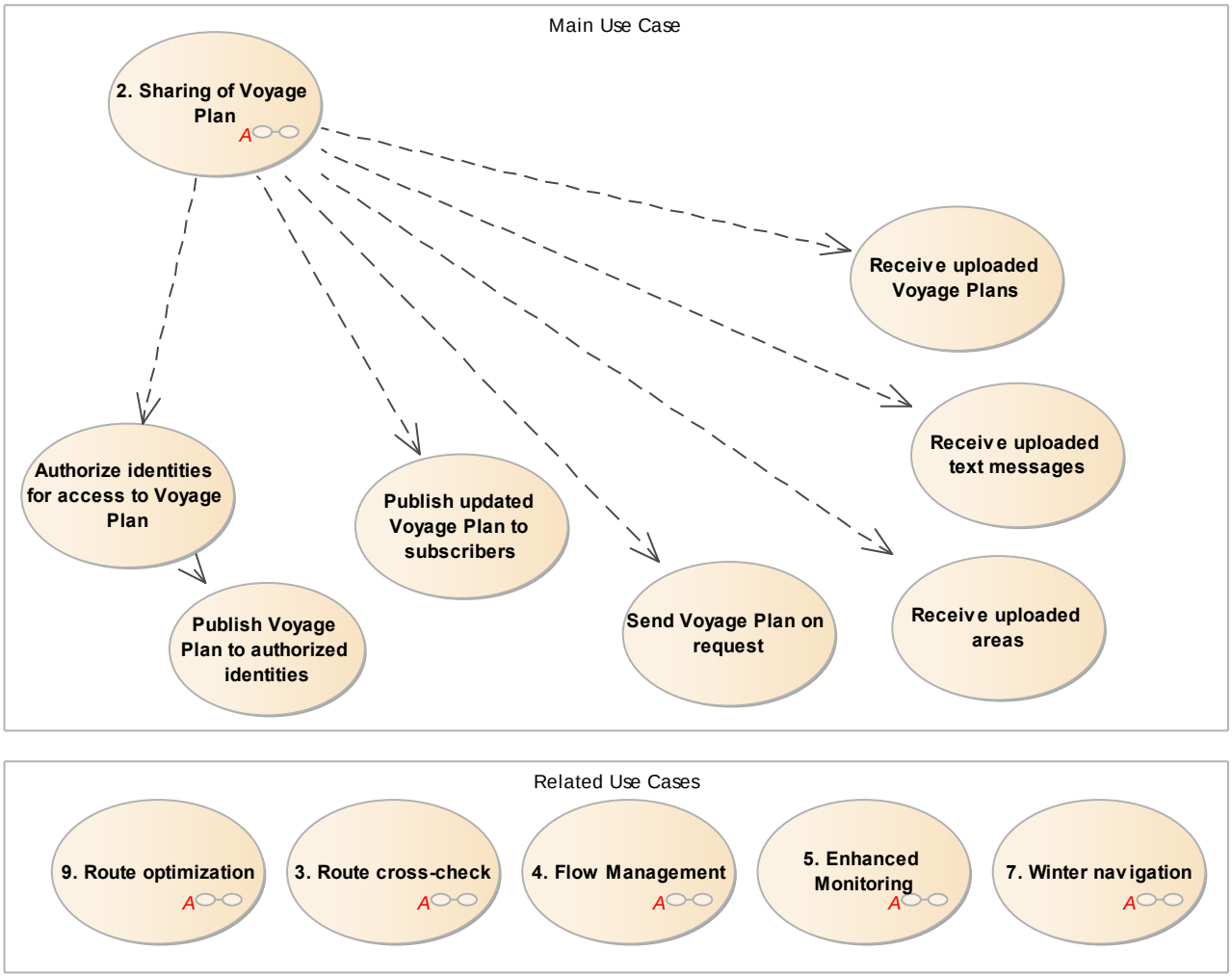


Figure 2 Use Cases

3.1 Functional and Non-functional Requirements

Requirements briefly:

- VIS has a **storage** (for storing sent and received messages, XML schemas, logs)
- VIS is an **information service registered** in SeaSWIM central Service Registry
- VIS has service **endpoints** for exposing **methods**
- VIS has a **function to validate** message payload according to the following predefined schemas (rtz, text, area)
- **All communication** between VIS and SeaSWIM Central services or and other information services is achieved using SeaSWIM connector.

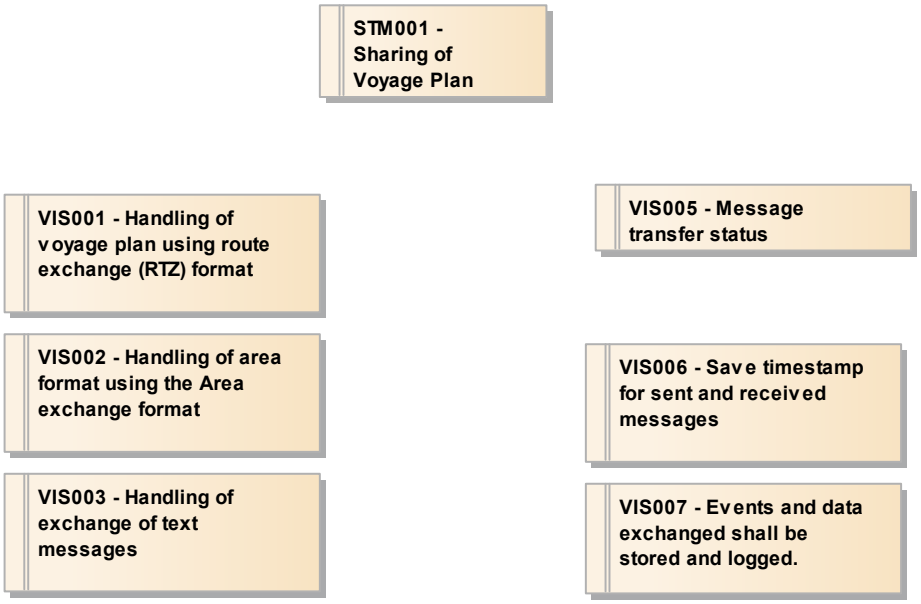


Figure 3 Requirements

Requirement Id	VIS002
Requirement Name	VIS002 - Handling of area format using the Area exchange format
Requirement Text	It shall be possible to upload an Area (S-124) to VIS.
Rationale	Introducing area exchange format into the maritime domain will give a more graphic view
Author	STM
Reference	

Requirement Id	STM001
Requirement Name	STM001 - Sharing of Voyage Plan
Requirement Text	As part of the Voyage Information Object the Voyage Plan (VP) can be shared among the different parties participating in a ships voyage. The ship/shipping company is the information owner of the VP and as such chooses which actors that should be granted access to the voyage plan.
Rationale	
Author	STM
Reference	Use-Case 2: Sharing of Voyage Plan

Requirement Id	VIS003
Requirement Name	VIS003 - Handling of exchange of text messages
Requirement Text	It shall be possible to upload a STM Text Message to VIS.
Rationale	
Author	STM
Reference	

Requirement Id	VIS005
Requirement Name	VIS005 - Message transfer status
Requirement Text	Handling of message statuses sent to be able to support messages transferred ok
Rationale	
Author	
Reference	

Requirement Id	VIS006
Requirement Name	VIS006 - Save timestamp for sent and received messages
Requirement Text	The age of information shall be known by VIS
Rationale	
Author	
Reference	

Requirement Id	VIS001
Requirement Name	VIS001 - Handling of voyage plan using route exchange (RTZ) format
Requirement Text	Handle voyage plans in RTZ format identified by UVID for sending on request, publish to subscribers and forward incoming from external parties.
Rationale	
Author	STM
Reference	

Requirement Id	VIS007
Requirement Name	VIS007 - Events and data exchanged shall be stored and logged.
Requirement Text	All data to and from VIS shall be stored and logged with metadata to support the validation of STM.
Rationale	
Author	
Reference	

3.2 Other Constraints

3.2.1 Relevant Industrial Standards

IEC 61174:2015

3.3 Operational Nodes

The section describes the context surrounding the information service.

The Voyage Information Service, VIS, is a generic service for exchange of voyage information, primarily in RTZ format. Thus the VIS can be used both representing a ship, but also representing a Shore Centre or a Service Provider of e.g. Route Optimization.

In the picture below, the SSC (SeaSWIM Connector) is shown to represent the functionality required by the SeaSWIM infrastructure for mainly security reasons. The SSC can be either a separate service or component, or built-in functionality in the Voyage Information Service. For further details related to SeaSWIM and SeaSWIM Connector, please see http://stmvalidation.eu/seaswim-overview

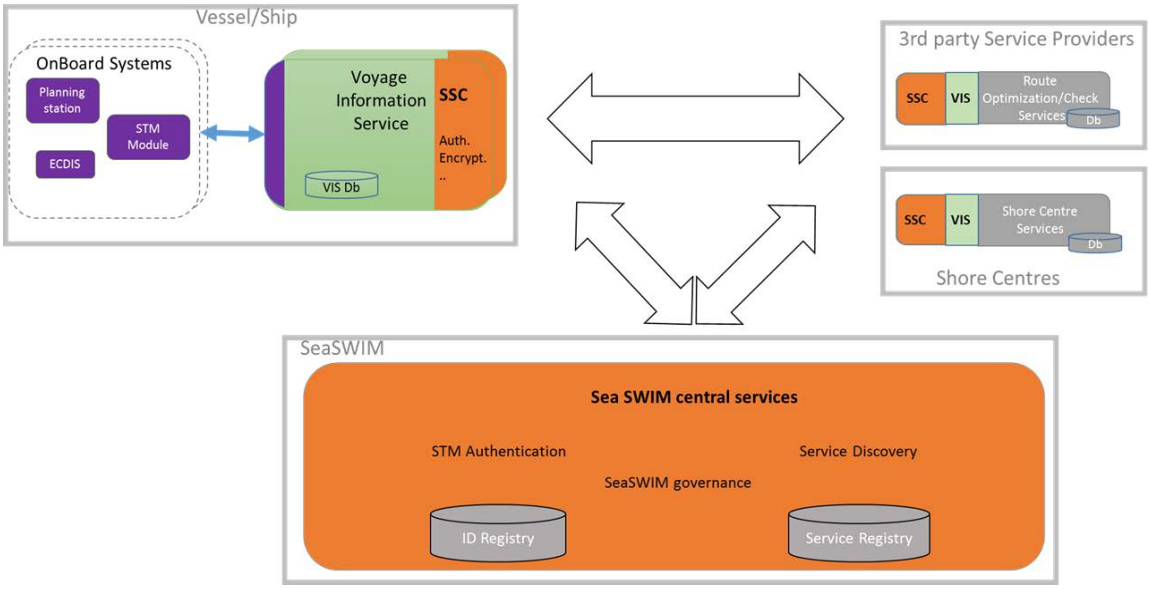


Figure 4 Overview of VIS

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.
Service provider	Organizations/ authorities offering e.g. route optimization services possible consumers of voyage plans provided by a vessel or a representation thereof. SMHI (Swedish Metrological & Hydrological Institute) is one example.
Shore Center	Collection of services, activities and procedures of Shore Center Refers to entities offering services such as route check and/ or enhanced monitoring.
SeaSWIM	SeaSWIM enables information security and service lookup in a structured and governed manner.

Table 1 Operational Nodes

3.3.1 Operational Activities

Operational Activities (processes) has not been more elaborated than on Use Case level.

4 Service Overview

The main purpose with VIS is to handle the communication around voyage information and the main artefact Voyage Plan (VP) in RTZ format. VIS implements methods for exposing new and updated VP's and to consume external VP's. VIS also supports subscription of voyage plans.

In addition to voyage plans (RTZ), VIS also supports exchange of STM Text Message and area message (S-124).

VIS is also consuming the same Upload and Acknowledgement interface specified in VIS, hence VIS assumes VIS or equal on subscriber consumer.

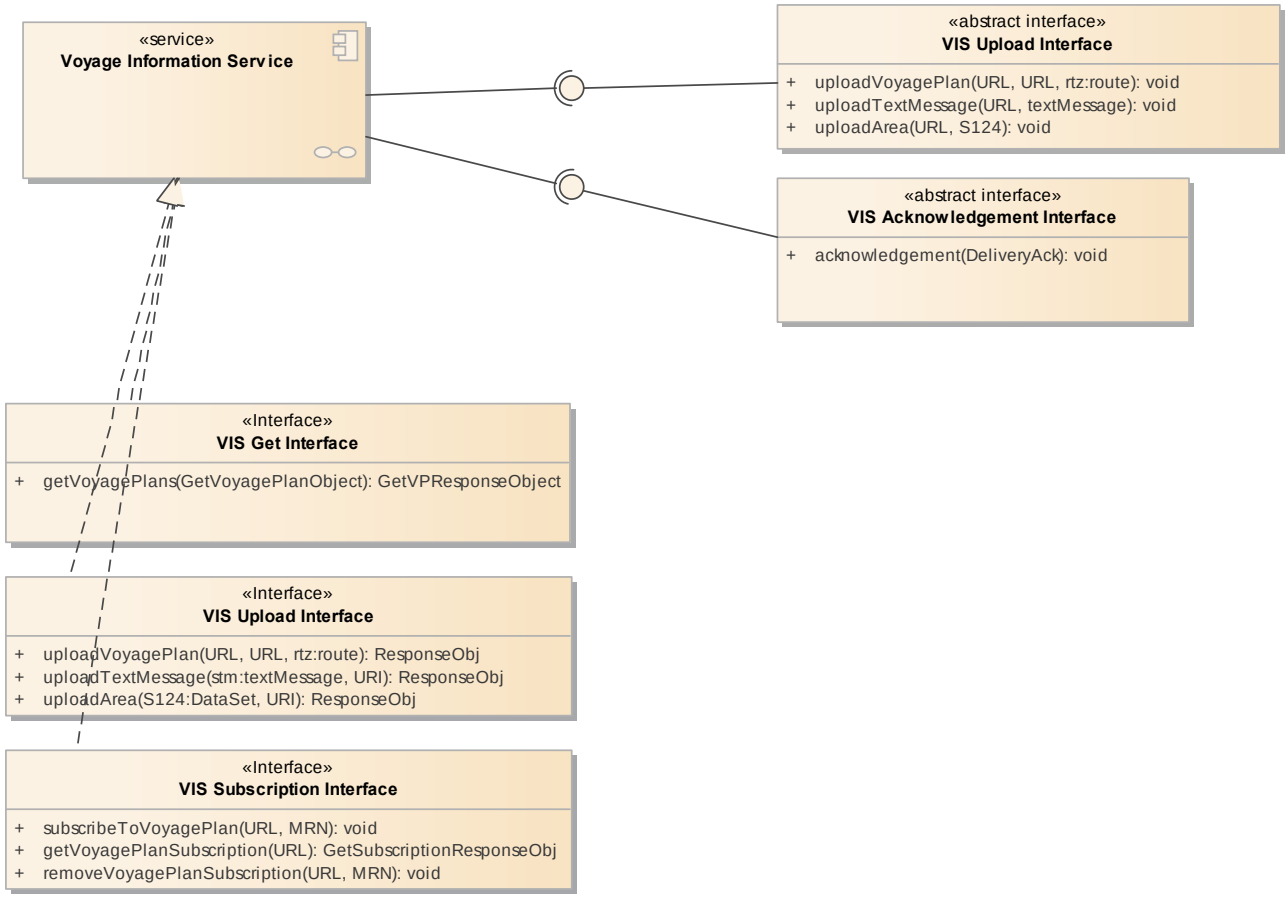


Figure 5 Service overview

4.1 Service Interfaces

The following set of interfaces and operations are provided by VIS.

ServiceInterface	Role (from service provider point of view)	ServiceOperation
VIS Get Interface	Provided	getVoyagePlans
VIS Subscription Interface	Provided	subscribeToVoyagePlan getVoyagePlanSubscription removeVoyagePlanSubscription
VIS Upload Interface	Provided	uploadVoyagePlan uploadTextMessage uploadArea
VIS Acknowledgement Interface	Provided	acknowledgement

4.2 Consumer Interfaces

The following set of interfaces and operations are consumed by VIS.

Service Interface	Role	Service Operation
VIS Upload Interface	Consumed	uploadVoyagePlan
VIS Acknowledgement Interface	Consumed	acknowledgement

5 Service Data Model

5.1 Service Data Exchange Model SeaSWIM interface

The Voyage Information Service exchange three main artefacts; RTZ (route), TXT (textMessage) and S124 (area message).

The RTZ (route) is further described in reference Route Exchange format (IEC 61174 App S).

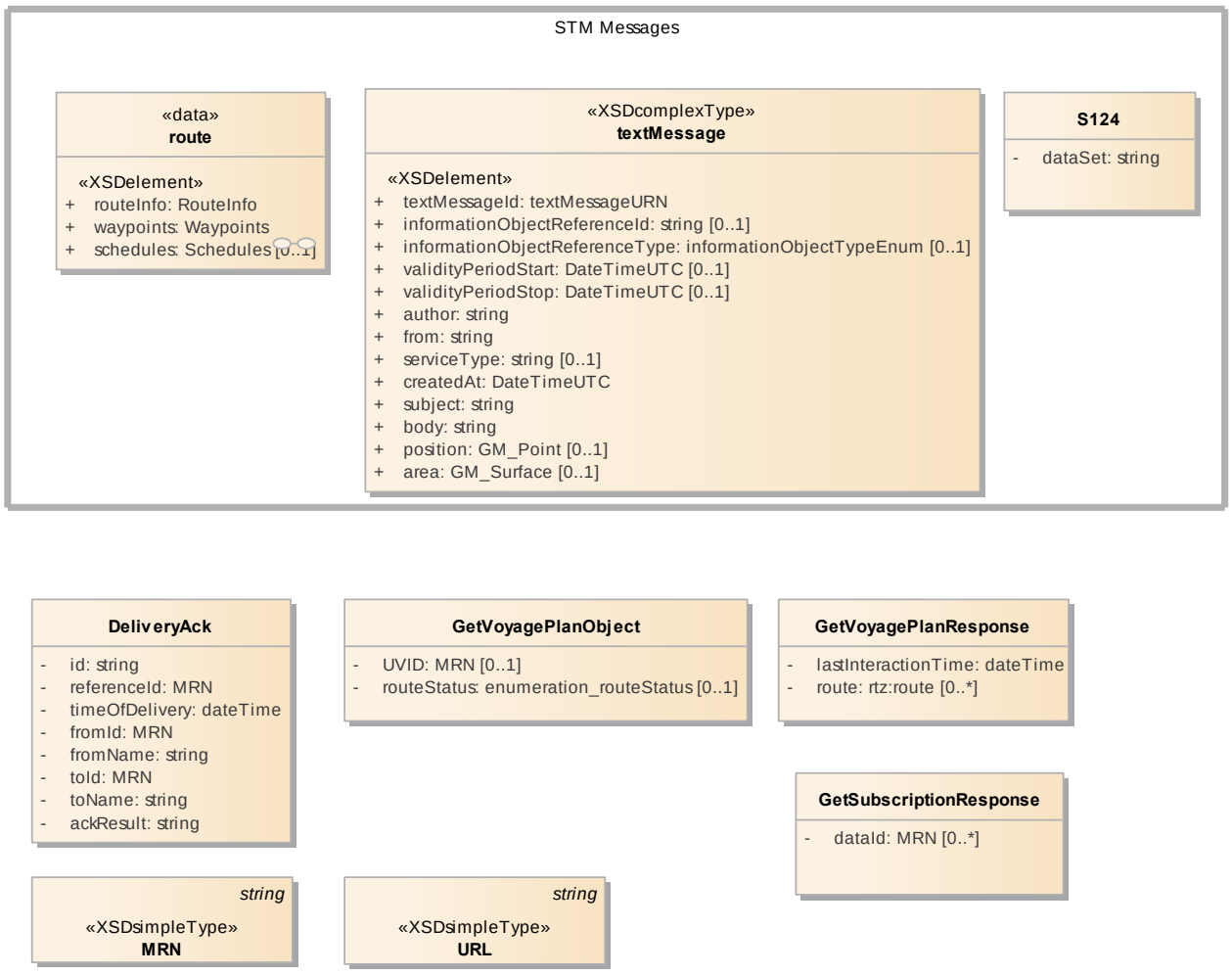


Figure 6 Service Data Model

5.1.1 route

RTZ files contain a number of waypoints, followed with auxiliary schedules.

For detailed information, see http://stmvalidation.eu/schemas/

Element Name	Attributes		
route	**Name**	**Type**	**Description**
	routeInfo	RouteInfo	Generic route information.
	waypoints	Waypoints	A list of waypoints.
	schedules	Schedules	Optional list of schedules.
	extensions	Extensions	You can add extend RTZ by adding your own elements from another schema here.
	version	NonEmpty String	Format version

5.1.2 textMessage

Text message defined in STM project.

For detailed information, see http://stmvalidation.eu/schemas/

Element Name	Attributes		
textMessage	**Name**	**Type**	**Description**
	textMessageId	textMessageURN	Identifier of the text message, mandatory.
	informationObjectReferenceId	string	A reference to an information object, optional.
	informationObjectReferenceType	informationObjectTypeEnum	STM payload format reference, optional.
	validityPeriodStart	DateTimeUTC	Start of validity period in ISO 8601 format, optional.
	validityPeriodStop	DateTimeUTC	Stop of validity period in ISO 8601 format, optional.
	author	string	The message author, mandatory.
	from	string	The sending actor, mandatory.
	serviceType	string	The service type of the sender, optional.
	createdAt	DateTimeUTC	The message creation dateTime, mandatory.
	subject	string	The message subject, mandatory.
	body	string	The message body, mandatory.
	position	GM_Point	Geographic point, optional.
	area	GM_Surface	Geographic area, optional.

5.1.3 S124

S124 area message

For detailed information, see http://stmvalidation.eu/schemas/

Element Name	Attributes		
S124	**Name**	**Type**	**Description**
	dataSet	string	S124 area message as defined at STM Developer Forum site http://stmvalidation.eu.

5.1.4 DeliveryAck

Object for message ACK

Element Name	Attributes		
DeliveryAck	**Name**	**Type**	**Description**
	id	string	Id for the ACK
	referenceId	MRN	Reference to delivered message according to the STM MRN identifier. For example an unique voyage identifier: urn:mrn:stm:voymgt:uvid:<organizationId>:<local voyagenumber>
	timeOfDelivery	dateTime	Time of delivery
	fromId	MRN	Identity of source (sender) of message that have been delivered according to the STM MRN identifier. Example: urn:mrn:stm:org:<organizationId>
	fromName	string	Friendly name of sender
	toId	MRN	Identity of target (receipient) of message delivery according to the STM MRN identifier. Example: urn:mrn:stm:org:<organizationId>
	toName	string	Friendly name of recipient
	ackResult	string	

5.1.5 GetVPResponseObject

Response object from request for voyage plan

Element Name	Attributes		
GetVPResponseObject	**Name**	**Type**	**Description**
	lastInteractionTime	dateTime	Last interaction time with private application
	route	rtz:route	Sequence of 0 or more route messages (RTZ) in XML format

5.1.6 GetVoyagePlanObject

Element Name	Attributes		
GetVoyagePlanObject	**Name**	**Type**	**Description**
	UVID	MRN	UVID in MRN format, optional routeStatus as string "1" to "8", optional
	routeStatus	enumeration_routeStatus	

5.1.7 MRN

Marine Resource Name identifier, based on URN.

Element Name	Attributes		
MRN	**Name**	**Type**	**Description**

5.1.8 URL

Uniform Resource Identifier

Element Name	Attributes		
URL	**Name**	**Type**	**Description**

6 Service Interface Specifications

The main purpose with VIS is to handle the communication around voyage information and the main artefact Voyage Plan (VP) in RTZ format. VIS implements methods for exposing new and updated VP's and to consume external VP's. VIS also supports subscription of voyage plans.

In addition to voyage plans (RTZ), VIS also supports exchange of STM Text Message and area message (S-124).

VIS is also consuming the same Upload and Acknowledgement interface specified in VIS, hence VIS assumes VIS or equal on subscriber consumer.

6.1 Voyage Information Service

The Voyage Information Service provides interfaces for requesting voyage plan (Get), requesting subscription of voyage plans (Subscription) and to upload voyage plan, text message and areas (Upload).

6.1.1 VIS Get Interface

Facilitates operations for requesting a Voyage Plan.

getVoyagePlans()

Returns active voyage plans according to parameters that the requester is authorized to. The response can contain 0 or more (0..*) voyage plans, but only 0 or one (0..1) voyage plan with same UVID.

An active voyage is a voyage plan with routeStatus not equal 8.

Will return the latest published voyage plan with routeStatus not equal 8,for each UVID (vesselVoyage). If two or more voyage plans have routeStatus=7, only the latest published voyage plan for each ship shall be returned.

If not authorized, a message will be returned and a message will be sent on private side as a authorization request.

Also returns "Last known time" with interaction on private side. The intention is to give a consumer a possibility to assess validity of given voyage plan(s) if the e.g. ship is offline.

Operation functionality

Retrieve requester identity

Retrieve messages from cache/repository that apply to get criterias (parameters)

Check authorization to data against ACL

If authorized, return VPs

Operation Parameters

Parameter Name	Direction	Data Type	Description
getVoyagePlanObj	Input	GetVoyagePlanObject	Request can be of a specific UVID and/or a specific route status

Return	Direction	Data Type	Description
	Return	GetVPResponseObject	

6.1.2 VIS Subscription Interface

Facilitates operations for subscribing and unsubscribing to a Voyage Plan.

subscribeToVoyagePlan()

Facilitates request for subscription of voyage plans to the given callbackEndpoint.

If UVID is empty, VIS will search and add a subscription of all active voyageplans (UVIDs) the requester has access to.

If the requester doesn't have access to any UVIDs, a error message will be returned and private application will be notified.

If the requester have access to some of the active UVIDs, a subscription will be added to the authorized UVID(s). The private application will be notified regarding the non-authorized UVID(s).

If the requester have access to all active UVIDs, a subscription for each authorized UVID will be created.

An active voyage is a voyage plan with routeStatus not equal 8.

Operation functionality

The received request is stored in VIS database.

The authorization to data is checked and if authorized an OK is given and the callbackEndpoint from calling party is stored in a subscription table in VIS database together with subscription parameters.

The latest active Voyage Plan is sent to callbackEndpoint

Operation Parameters

Parameter Name	Direction	Data Type	Description

callbackEndpoint	Input	URL	
dataId	Input	MRN	

Return	Direction	Data Type	Description
	Return	void	

getVoyagePlanSubscription()

Get information on subscribed voyage plans.

Operation Parameters

Parameter Name	Direction	Data Type	Description
callbackEndpoint	Input	URL	

Return	Direction	Data Type	Description
	Return	GetSubscriptionResponseObj	

removeVoyagePlanSubscription()

Remove subscription of Voyage Plans

Operation functionality

The subscription attached to the subscription parameters is removed

Operation Parameters

Parameter Name	Direction	Data Type	Description
callbackEndpoint	Input	URL	
dataId	Input	MRN	

Return	Direction	Data Type	Description
	Return	void	

6.1.3 VIS Upload Interface

Facilitates operations for uploading a Voyage Plan, Text Message and Area.

uploadVoyagePlan()

Facilitates sending (uploading) a voyage plan to VIS. The route shall be uncompressed (RTZ).

If endpoint/URL provided for deliveryACK, an ACK will be sent when message has been forwarded to private application.

Operation functionality

The uploaded message is stored in cache

The voyage plan is checked against the RTZ schema

If correct, a notification is sent to STM Module

If delivery ACK is requested, the flag is set in cache and when the STM Module calls getMessage, VIS sends the message to STM Module and a delivery ACK to the requested endpoint.

Operation Parameters

Parameter Name	Direction	Data Type	Description
deliveryAckEndpoint	Input	URL	Name of ACK callback endpoint if ACK is requested, otherwise empty
callbackEndpoint	Input	URL	
voyagePlan	Input	rtz:route	The route in RTZ format

Return	Direction	Data Type	Description
	Return	ResponseObj	

uploadTextMessage()

Facilitates sending (uploading) a text message to VIS to be forwarded to the ship (STM Module).

If endpoint/URL provided for deliveryACK, an ACK will be sent when message has been forwarded to private application.

Operation functionality

The uploaded message is stored in cache

The textMessage is checked against the textMessage schema

If correct, a notification is sent to STM Module

If delivery ACK is requested, the flag is set in cache and when the STM Module calls getMessage, VIS sends the message to STM Module and a delivery ACK to the requested endpoint.

Operation Parameters

Parameter Name	Direction	Data Type	Description
textMessage	Input	stm:textMessage	The text message in STM Text Message format
deliveryAckEndpoint	Input	URI	Name of ACK callback endpoint if ACK is requested, otherwise empty

Return	Direction	Data Type	Description
	Return	ResponseObj	

uploadArea()

Facilitates sending (uploading) an area to VIS to be forwarded to the ship (STM Module).

If endpoint/URL provided for deliveryACK, an ACK will be sent when message has been forwarded to private application.

Operation functionality

The uploaded message is stored in cache

The message is checked against the area schema

If correct, a notification is sent to STM Module

If delivery ACK is requested, the flag is set in cache and when the STM Module calls getMessage, VIS sends the message to STM Module and a delivery ACK to the requested endpoint.

Operation Parameters

Parameter Name	Direction	Data Type	Description
area	Input	S124:DataSet	The area in S-124 format

deliveryAckEndpoint	Input	URI	Name of ACK callback endpoint if ACK is requested, otherwise empty

Return	Direction	Data Type	Description
	Return	ResponseObj	

6.1.4 VIS Acknowledgement Interface

acknowledgement()

During upload of message, an acknowledgement can be requested by giving a callback endpoint to a Acknowledge interface. If such an acknowledgement has been requested, the service will send an Acknowledgement message when the uploaded message has been retrieved/sent to private application.

Operation functionality

AckDelivery endpoint stored from upload interface

When uploaded message to VIS is sent to STM Module, VIS sends an ACK to uploaded Ack endpoint.

Operation Parameters

Parameter Name	Direction	Data Type	Description
DeliveryAck	Input	DeliveryAck	

Return	Direction	Data Type	Description
	Return	ResponseObj	

7 Service Dynamic Behaviour

7.1 VIS SeaSWIM Interface

This section contains sequence diagrams related to VIS SeaSWIM interface.

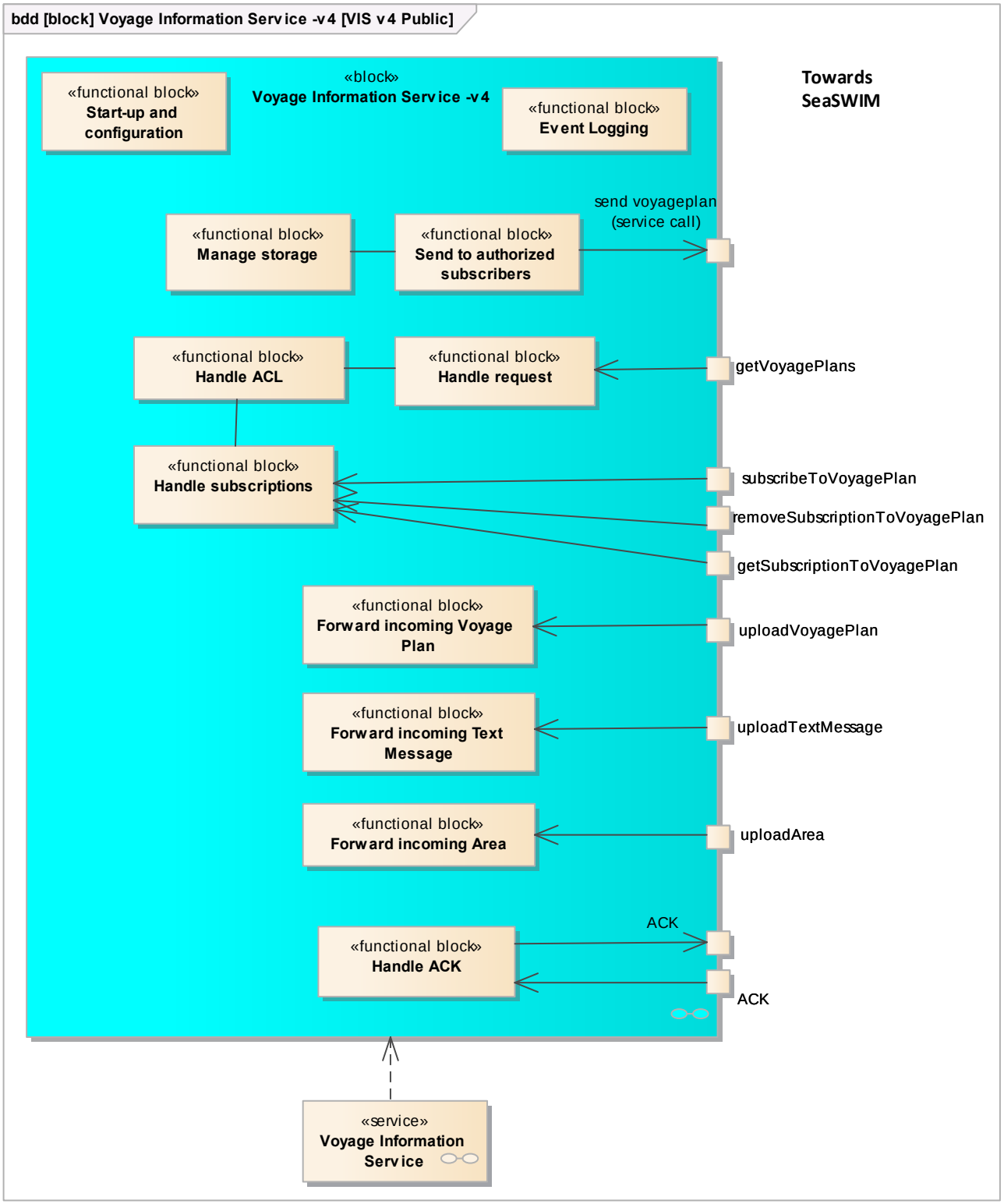


Figure 7Service details

7.1.1 VIS Get Interface

This section contains sequence diagrams related to VIS SeaSWIM Get interface.

A service consumer may request a voyage plan any time, either asking for a known UVID or just ask for any voyage plan published in VIS instance.

A service consumer can ask for voyage plans in a certain status, according to routeStatus enumeration, or ask for any voyage plan.

If the service consumer is not authorized by the "owner" of the VIS instance, a notification is forwarded to the "owner" and the service consumer don't get any voyage plans back until "owner" has authorized the service consumer.

If several unique voyage plans have been published in the VIS instance, all will be returned in the request. This enables the VIS to be deployed as a catalogue of voyage plans and routes.

However be aware that only zero or one (0..1) voyage plans in routeStatus=7 (Used for monitoring) can be returned from a VIS.

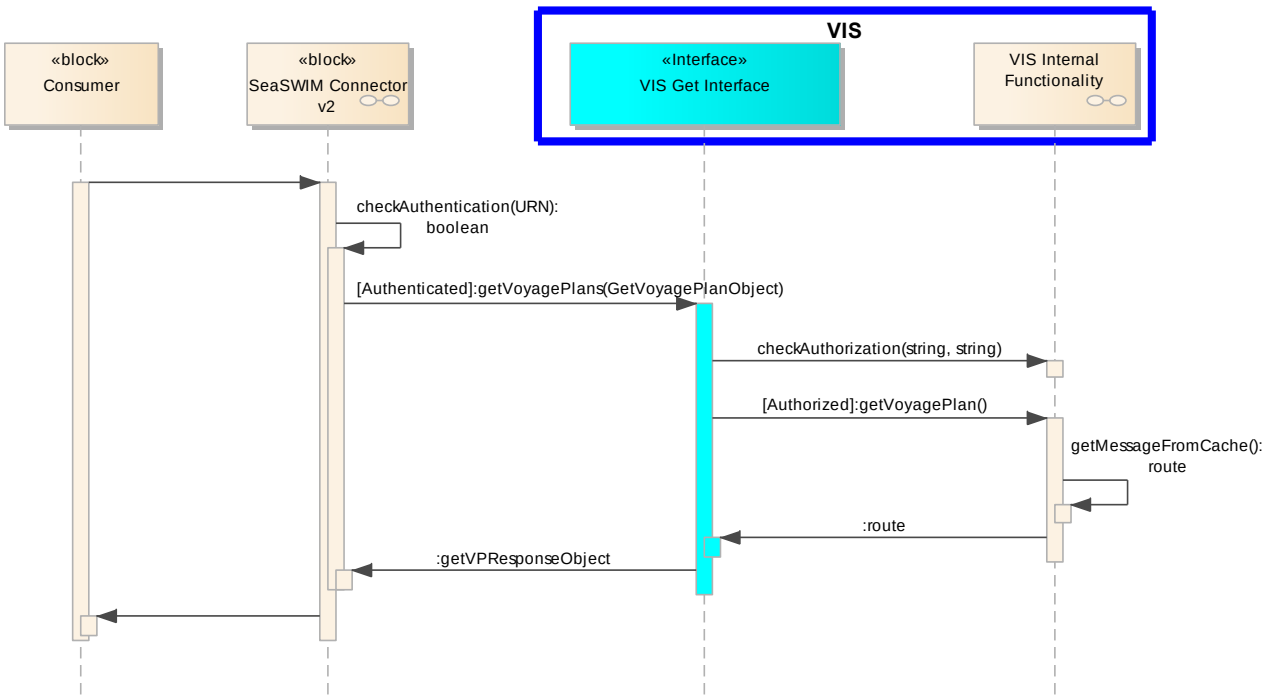
The service consumer must always check the routeStatus and act according the purpose by the service consumer. If the service consumer only wants "Used for monitoring", the request should be for routeStatus="7".

VIS will only handle requests from service consumer that are authenticated in STM.

Interaction getVoyagePlan

Message exchange pattern: REQUEST_RESPONSE

At receipt of request for a voyage plan in VIS, the user authorization is checked using an Access Control List (ACL). In case of successful authorization, the requested voyage plan(s) are fetched and returned to the calling service. If unsuccessful authorization, a non-authorized error response is sent. See further diagram Not authorized.


VIS
«block»
Consumer
«block»
SeaSWIM Connector
v2
«Interface»
VIS Get Interface
VIS Internal
Functionality
checkAuthentication(URN):
boolean
[Authenticated]:getVoyagePlans(GetVoyagePlanObject)
checkAuthorization(string, string)
[Authorized]:getVoyagePlan()
getMessageFromCache():
route
:route
:getVPResponseObject

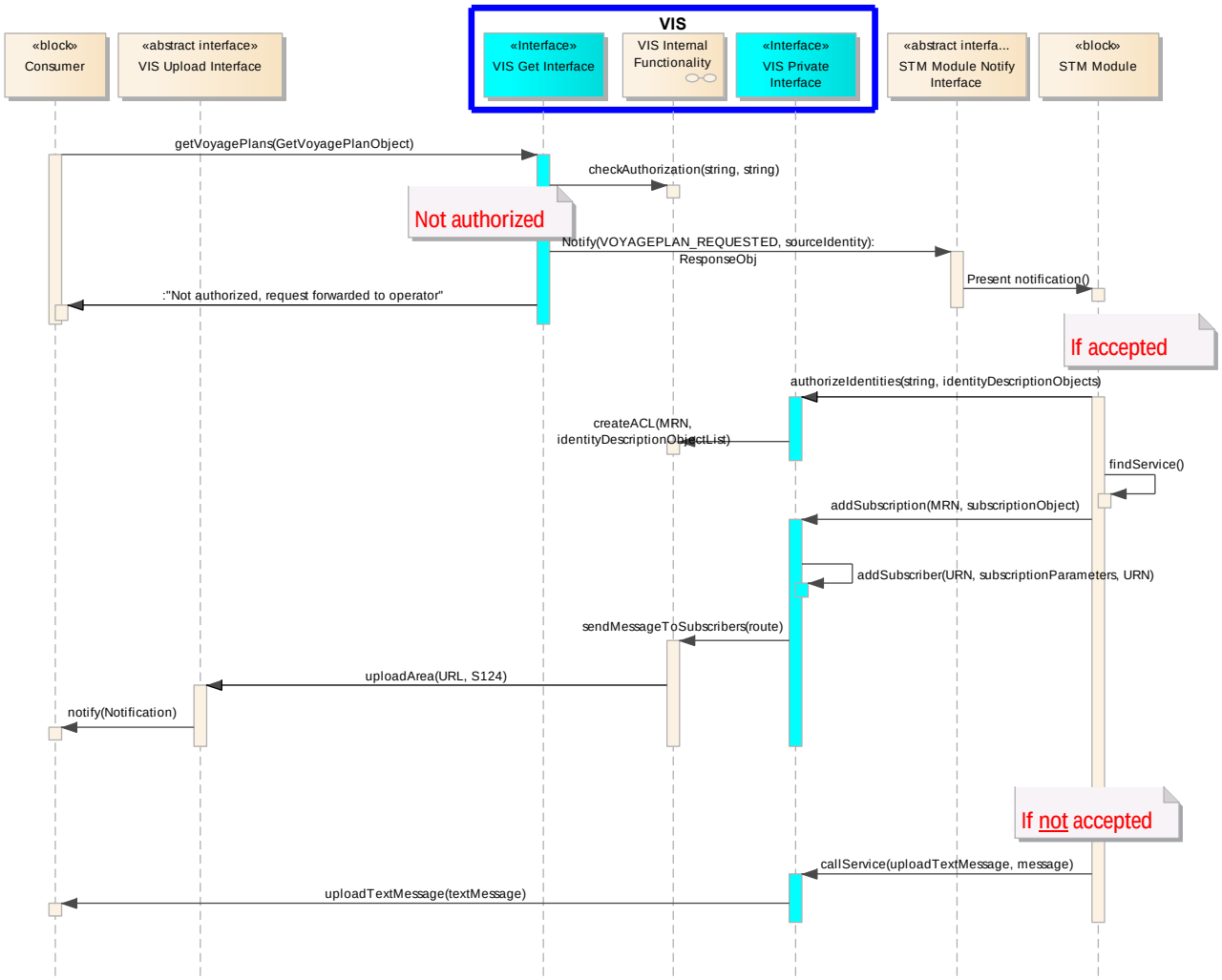
Service orchestration - Not authorized

In case the consumer is not authorized to requested data, the private application is notified hereof. The service consumer receives a message “Not authorized, request forwarded to operator”.

If no UVID is provided as parameter, a notification is sent to the private application only if the requester is not authorized to the voyage plan Used for monitoring (latest published voyage plan with routeStatus="7" for one ship).

It is then up to the user operating the private application to authorize the consumer to the requested voyage plan. Hereby creating a record in VIS ACL for the consumer identity.

In the case the operator chooses not to authorize the consumer, a textMessage can be sent to the consumer with notification of an unsuccessful authorization.



7.1.2 VIS Upload Interface

This section contains sequence diagrams related to VIS SeaSWIM Upload interface.

Asynchronous Acknowledgement can be requested.

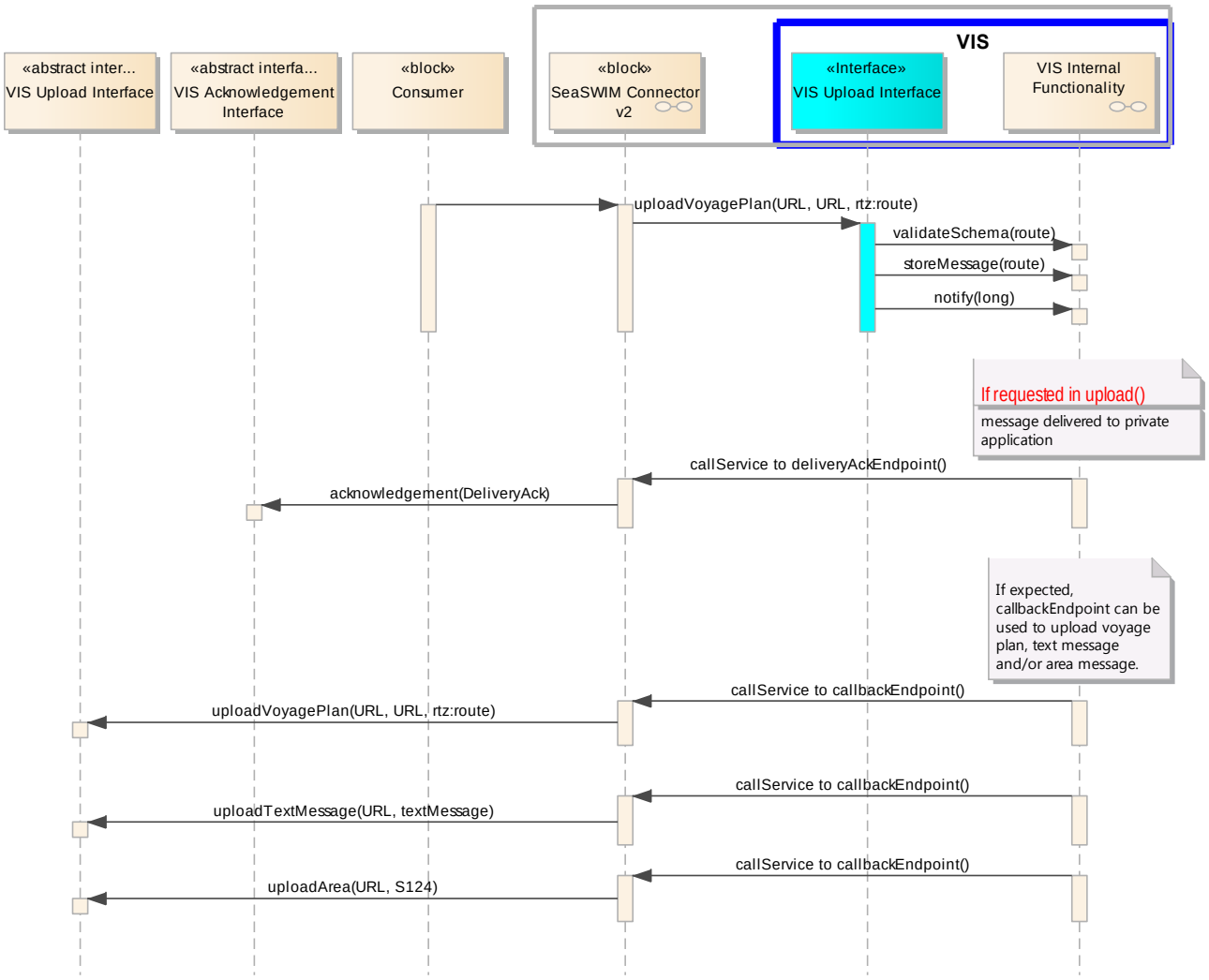
A service provider can always upload a voyageplan, text message or area message to VIS.

The service provider can always request an acknowledge message by providing an acknowledgement endpoint in the upload service request. When the message has been delivered to VIS private side, an acknowledgement is sent to the service provider. This acknowledgement however does not ensure that the message have reached the end user. This depends on the deployment on the private side where the STM Module may be an application on shore side and proprietary system to the end user, such as a ship.

VIS will only handle uploaded messages from service providers that are authenticated in STM.

Interaction uploadVoyagePlan

Message exchange pattern: REQUEST_CALLBACK

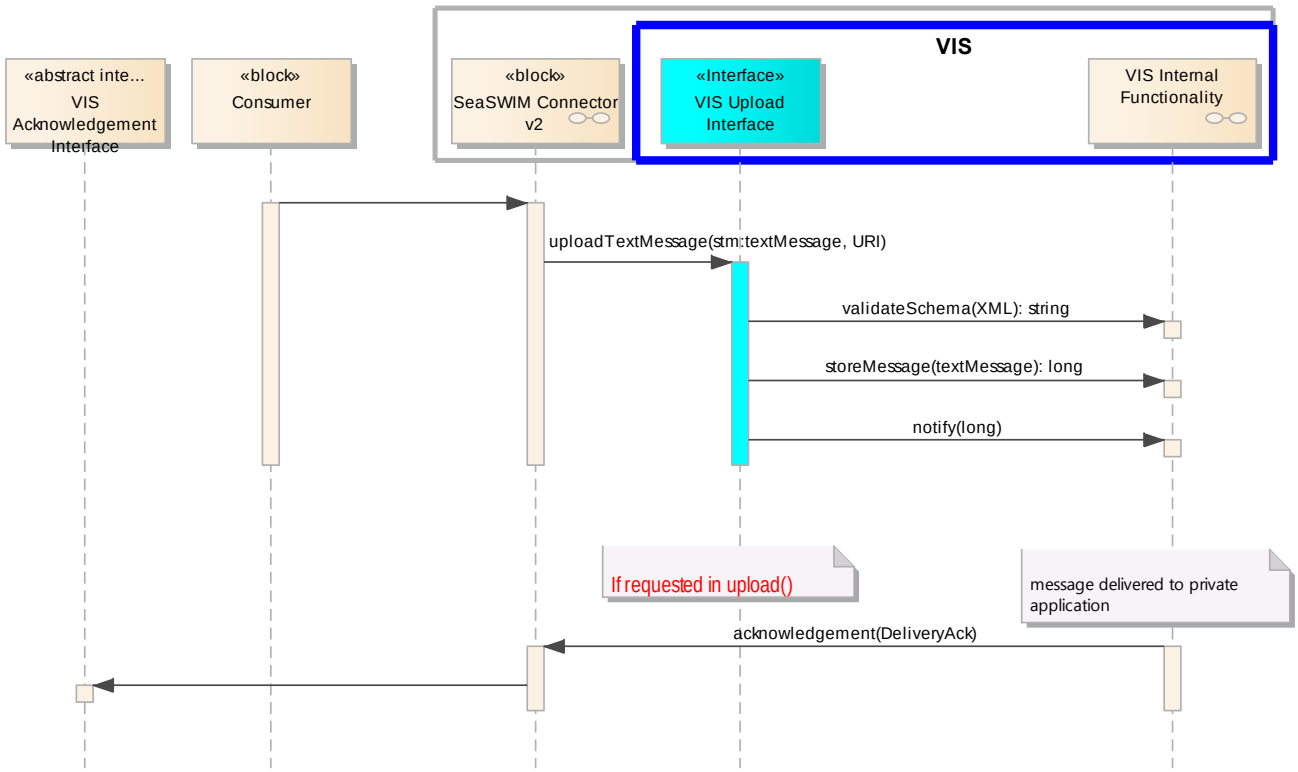

VIS
«abstract inter...
VIS Upload Interface
«abstract interfa...
VIS Acknowledgement Interface
«block»
Consumer
«block»
SeaSWIM Connector v2
«Interface»
VIS Upload Interface
VIS Internal Functionality
uploadVoyagePlan(URL, URL, rtz:route)
validateSchema(route)
storeMessage(route)
notify(long)
If requested in upload()
message delivered to private application
callService to deliveryAckEndpoint()
acknowledgement(DeliveryAck)
If expected, callbackEndpoint can be used to upload voyage plan, text message and/or area message.
callService to callbackEndpoint()
uploadVoyagePlan(URL, URL, rtz:route)
callService to callbackEndpoint()
uploadTextMessage(URL, textMessage)
callService to callbackEndpoint()
uploadArea(URL, S124)

Interaction uploadTextMessage

Message exchange pattern: REQUEST_CALLBACK

After receipt of a text message, the originating user organization is authenticated. Following a successful authentication the payload of the received message is validated against the schema.

In the case a deliveryAckEnpoint is supplied as parameter, an acknowledgement message is returned to the consumer after the delivery to the private application.

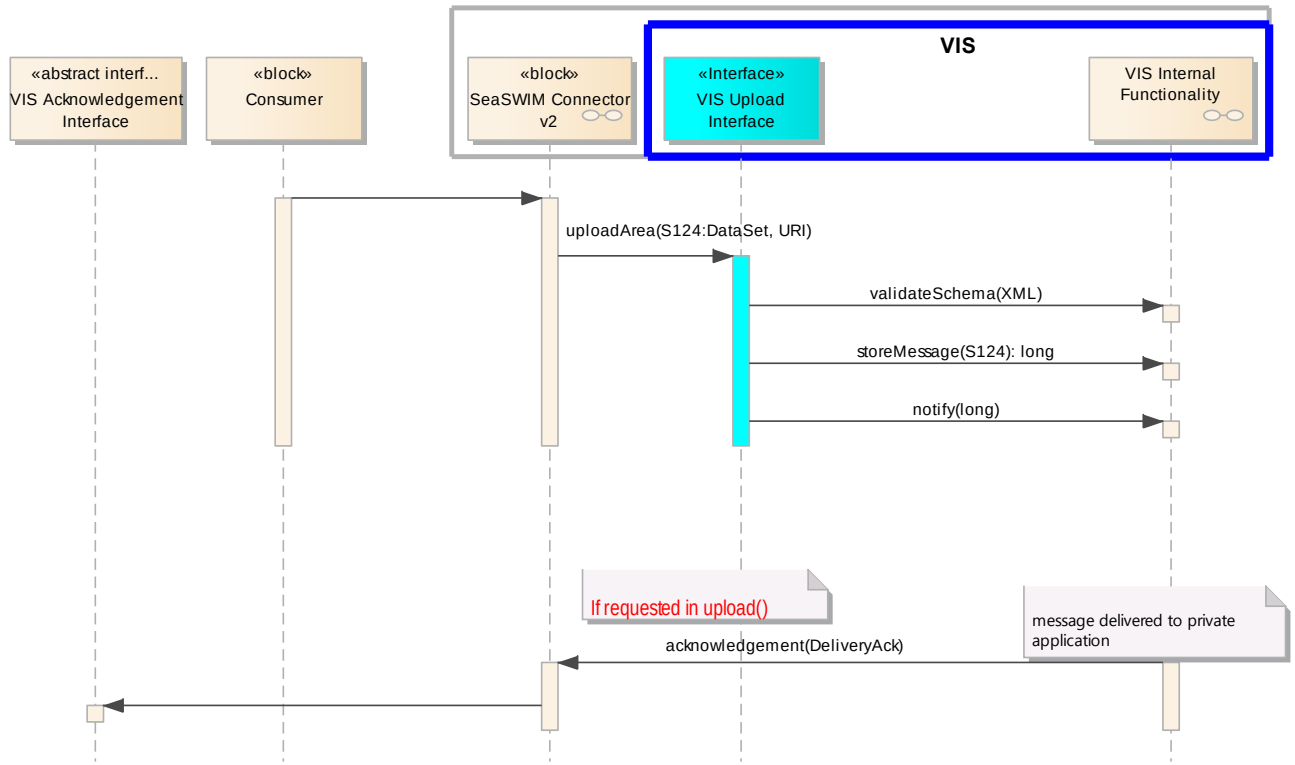


Interaction uploadArea

Message exchange pattern: REQUEST_CALLBACK

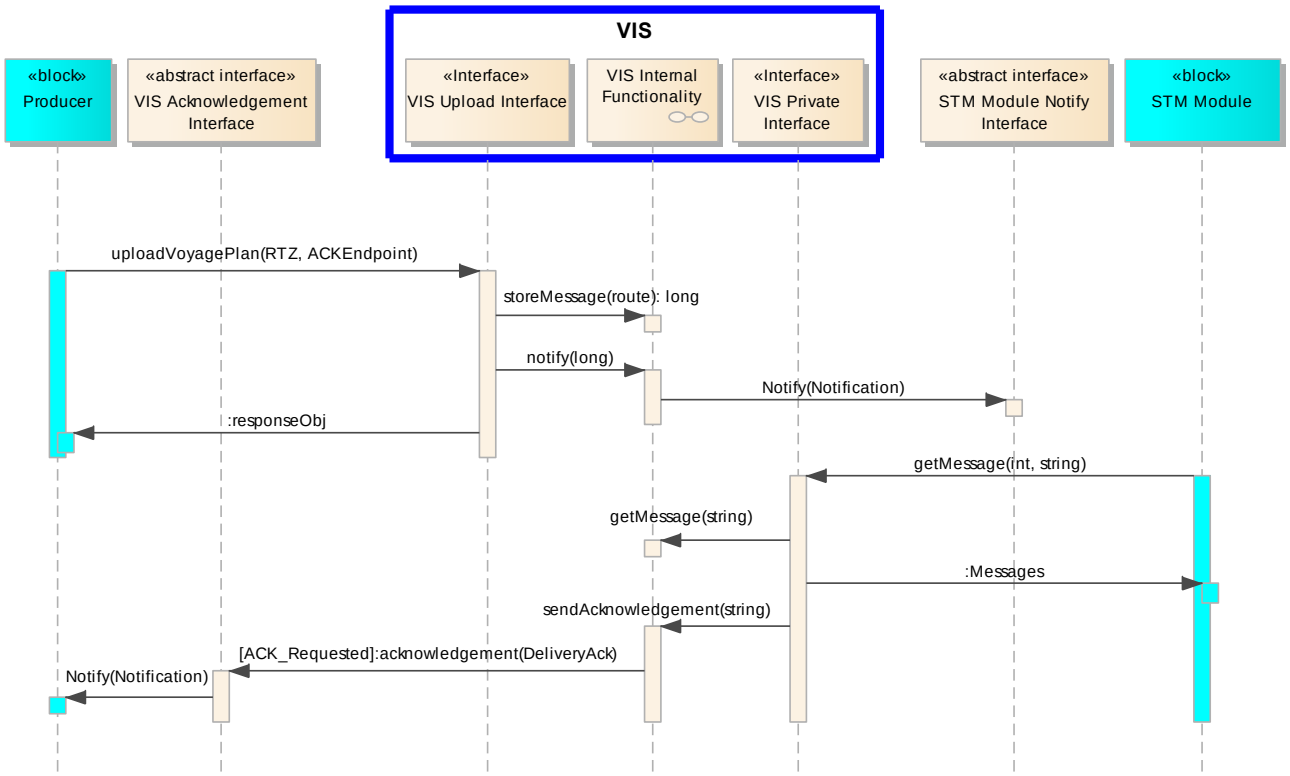
After receipt of a area message (S-124), the originating user organization is authenticated. Following a successful authentication the payload of the received message is validated against the schema.

In the case a deliveryAckEnpoint is supplied as parameter, an acknowledgement message is returned to the consumer after the delivery to the private application.



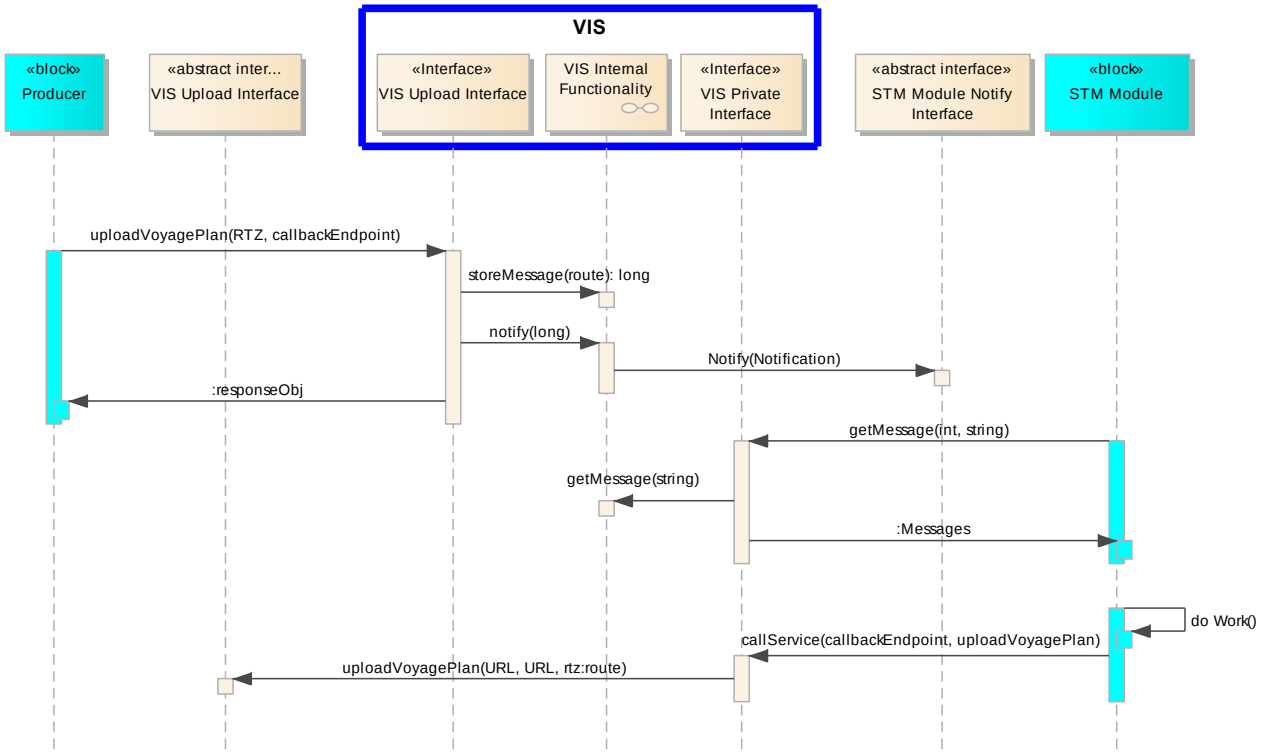
Service orchestration - Upload with ACK

The acknowledgement interface VIS exposes, is the endpoint for acknowledgement messages optionally requested by use of parameter deliveryAckEndpoint at upload of messages to VIS. The acknowledgement message is created for a specific message when it is successfully retrieved by the STM Module using VIS private interface getMessage, i.e. forwarded to the vessel. When the ACK is received, a notification is sent to the STM Module. The STM Module is responsible for checking and acting if ACK is not received.



Service orchestration - Upload with Callback

The callbackEndpoint can be provided for two purposes; inform that callback with information is expected, such as an optimized route; and inform to which endpoint the data is expected to be delivered to. The service responding on the provided callbackEndpoint shall still be an authenticated service.



7.1.3 VIS Subscription Interface

This section contains sequence diagrams related to VIS SeaSWIM Subscription interface.

A service consumer can always ask to subscribe to voyageplans. Either a known specific UVID or all voyageplans published in the VIS instance.

If the service consumer is not authorized by the "owner" of the VIS instance, a notification is forwarded to the "owner" and the service consumer don't get eny voyage plans back until "owner" has authorized the service consumer.

VIS will only handle requests from service consumer that are authenticated in STM.

Interaction subscribeToVoyagePlan

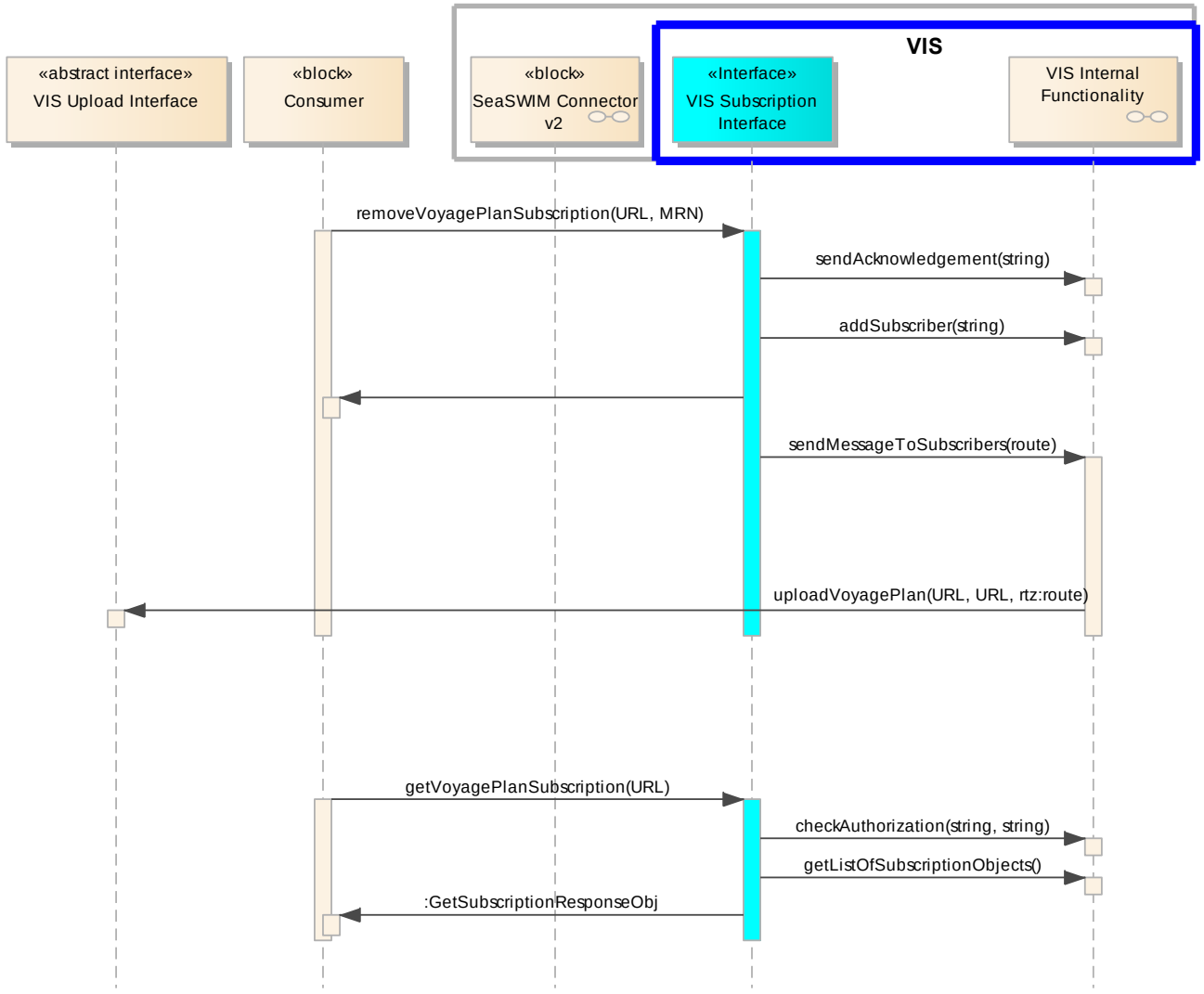
Message exchange pattern: REQUEST_CALLBACK

Consumer requests subscription by invoking interface subscribeToVoyagePlan providing the URI (address to consuming service uploadVoyagePlan interface - callbackEndpoint), optionally an uvid parameter can be passed for subscription on a specific voyagePlan. Following a successful authorization the subscriber identity and corresponding callbackEndpoint is stored in VIS dB subscription table and a voyagePlan is sent to the added subscriber. Every time a voyagePlan is published in VIS, the voyagePlan is forwarded to all selected subscribers. In response to the subscription request a responseObj is returned with statusCode=200, successful.

If UVID is not provided (is blank), VIS will try to set up a subscription to all "active" UVID with route with routeStatus 1-7 the requester has access to.

If there are 2 ore more voyage plans with routeStatus="7" for one ship, only the latest published of them will generate a subscription.

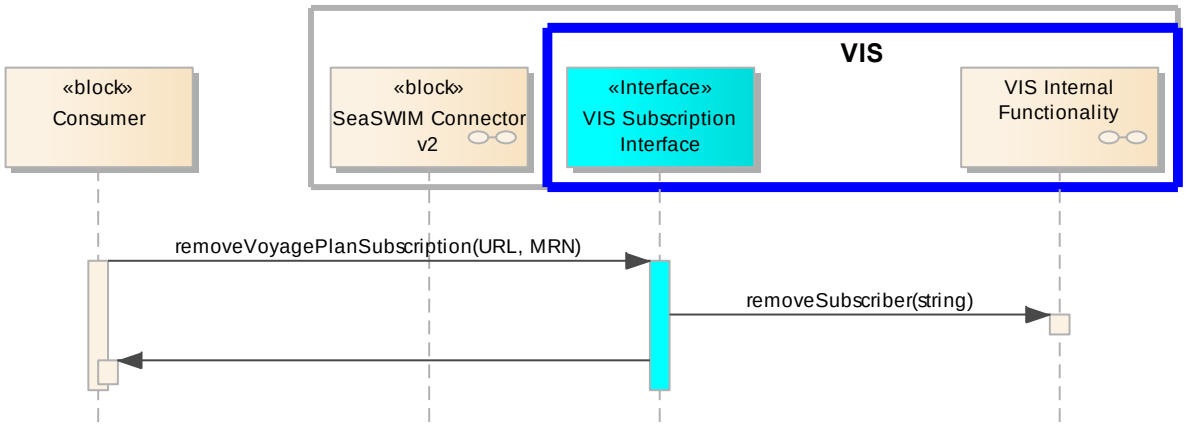
I.e. if there are one VP with routeStatus=7 and one in routeStatus=3, subscription will be enabled for both UVIDs.


VIS
«abstract interface»
VIS Upload Interface
«block»
Consumer
«block»
SeaSWIM Connector v2
«Interface»
VIS Subscription Interface
VIS Internal Functionality
removeVoyagePlanSubscription(URL, MRN)
sendAcknowledgement(string)
addSubscriber(string)
sendMessageToSubscribers(route)
uploadVoyagePlan(URL, URL, rtz:route)
getVoyagePlanSubscription(URL)
checkAuthorization(string, string)
getListOfSubscriptionObjects()
:GetSubscriptionResponseObj

Interaction removeVoyagePlanSubscription

Message exchange pattern: ONE_WAY

At removal of a subscription the removeVoyagePlanSubscription is invoked by the consumer. Parameters are the consumer callBackendpoint (mandatory) and optionally a specific uvid. At receipt of the subscription removal request VIS deletes all subscriptions for the callBackendpoint or a specific subscription for an uvid. In response to the subscription removal request a responseObj is returned with statusCode=200, successful.

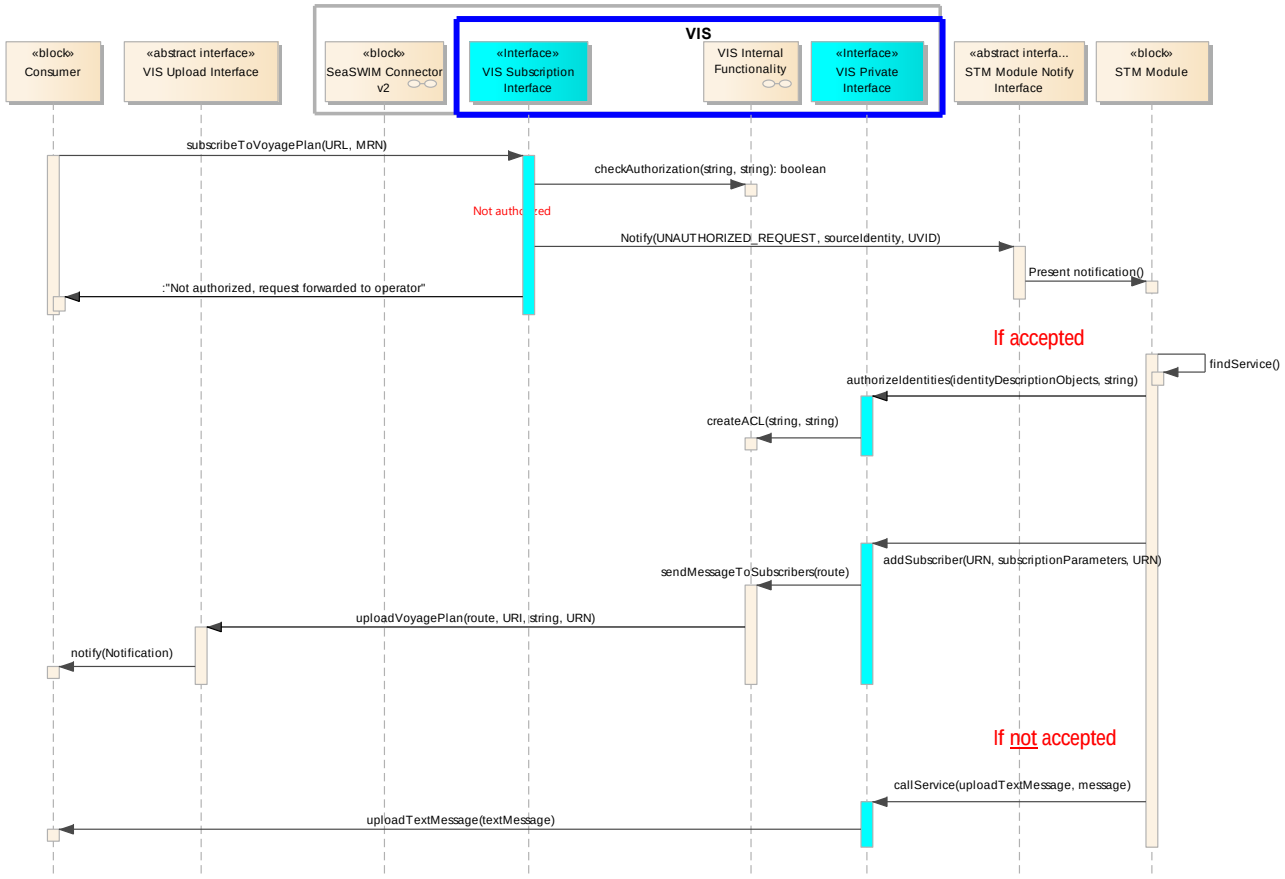


Service orchestration - Not authorized

In case the consumer is not authorized the STM Module operator onboard the vessel is notified hereof (message includes the consumer STM identity). The consumer receives a message “Not authorized request forwarded to operator”.

If authorized it is up to the user operating the STM Module to authorize the consumer to the requested voyage plan using VIS private interface authorizeIdentities. Hereby creating a record in VIS ACL for the consumer identity. VIS then searches SeaSWIM service registry in order to find the consuming service endpoint for receiving voyagePlans (findServices) and sends the requested voyagePlan to the consumer.

In the case the operator chooses not to authorize the consumer a textMessage is sent to the consumer with notification of an unsuccessful authorization.



7.2 Logging

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

7.2.1 VIS Event Log

Message exchange pattern:

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

The following events are proposed to be logged:

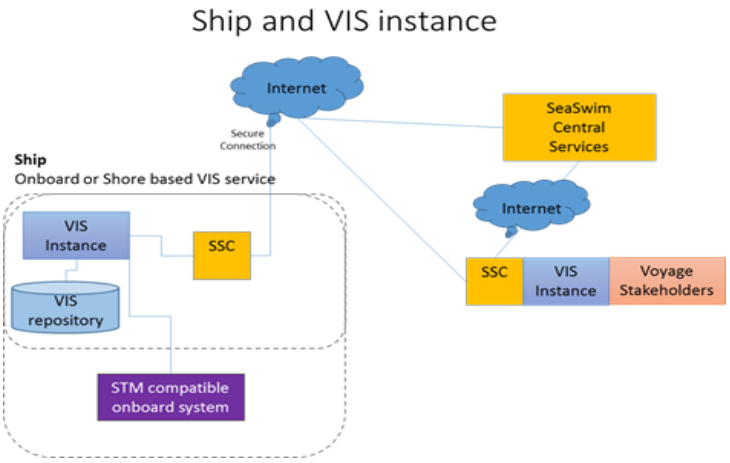
- Messages in and out of the service
- Failure events (Schema validation failure, Service operation failure)

Incoming service calls on SeaSWIM side	
Event	**Log description**
getVoyagePlans	Log event for incoming request Log event with returned data
subscribeToVoyagePlan	Log event for incoming request Log event with returned data
uploadVoyagePlan	Log event with incoming data
uploadTextMessage	Log event with incoming data
uploadArea	Log event with incoming data
acknowledgement	Log event with incoming data

Outgoing service calls on SeaSWIM side	
Event	**Log description**
<callService>	Log event with outgoing data

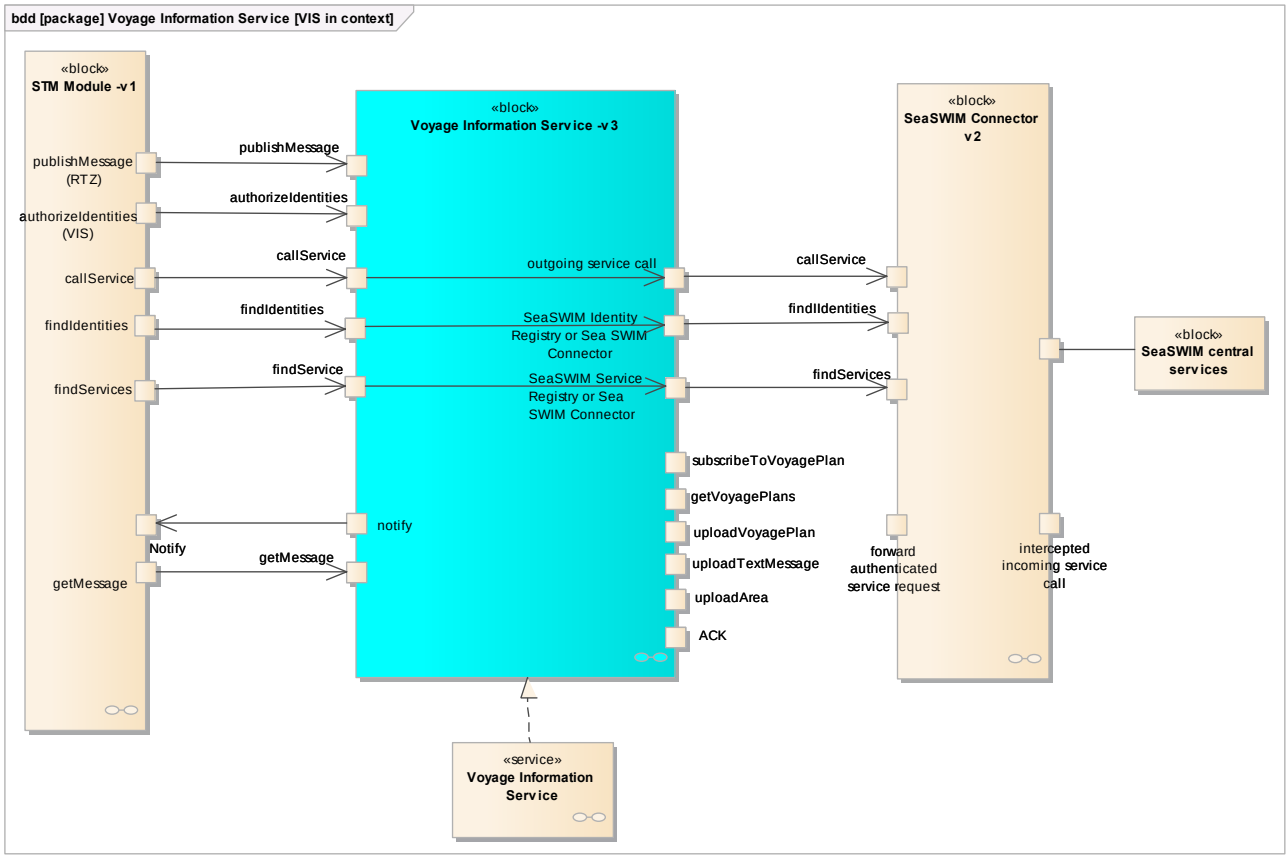
8 Service Provisioning (optional)

The VIS service is intended to be provided by either shore side server or onboard server depending on available connectivity and architecture.



The goal when specifying the Voyage Information Service was to support exchange of voyage information both on a ship, from ship operator planning centre but also a shore centre such as VTS with Enhanced Monitoring and 3rd party service providers for e.g. Route Optimization.

Using the same uploadVoyagePlan interface at all parties enables interoperability and scalability. When new services arises that are built around sharing of voyage information, they will be found in the service registry and can be used without new software.


bdd [package] Voyage Information Service [VIS in context]
«block»
STM Module -v1
publishMessage
(RTZ)
authorizeIdentities
(VIS)
callService
findIdentities
findServices
Notify
getMessage
«block»
Voyage Information Service -v3
publishMessage
authorizeIdentities
callService
findIdentities
findService
notify
getMessage
outgoing service call
SeaSWIM Identity
Registry or Sea SWIM
Connector
SeaSWIM Service
Registry or Sea
SWIM Connector
subscribeToVoyagePlan
getVoyagePlans
uploadVoyagePlan
uploadTextMessage
uploadArea
ACK
«block»
SeaSWIM Connector
v2
callService
findIdentities
findServices
forward
authenticated
service request
intercepted
incoming service
call
«block»
SeaSWIM central
services
«service»
Voyage Information
Service

9 References

Nr.	Version	Reference
[1] Service Documentation Guidelines	01.00	SG_Annex_A_Service_Documentation_Guidelines
[2] Route Exchange format (IEC 61174 App S)		http://stmvalidation.eu/schemas/
[3] STM Governance Handbook		-
[4] STM Text Message		http://stmvalidation.eu/schemas/
[5] Area Exchange Format.pdf		http://stmvalidation.eu/schemas/
[6] SeaSWIM Connector Specification		http://stmvalidation.eu/seaswim-overview/

10 Acronyms and Terminology

10.1 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

10.2 Terminology

Term	Definition
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
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

Appendix A Service Specification XML

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

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

<ServiceSpecificationSchema:id>urn:mrn:stm:service:specification:sma:vis</ServiceSpecificationSchema:id>

<ServiceSpecificationSchema:version>2.2</ServiceSpecificationSchema:version>

<ServiceSpecificationSchema:name>Voyage Information Service</ServiceSpecificationSchema:name>

<ServiceSpecificationSchema:status>released</ServiceSpecificationSchema:status>

<ServiceSpecificationSchema:description>The service supports exchange of voyage plans, text messages and area messages.</ServiceSpecificationSchema:description>

<ServiceSpecificationSchema:keywords>VIS, Voyage Information Service, STM Service, RTZ, Route Exchange, Area, S-124, TXT, Text Message</ServiceSpecificationSchema:keywords>

<ServiceSpecificationSchema:isSpatialExclusive>false</ServiceSpecificationSchema:isSpatialExclusive>

<ServiceSpecificationSchema:requirements>

<ServiceSpecificationSchema:requirement>

<ServiceSpecificationSchema:id>STM001</ServiceSpecificationSchema:id>

<ServiceSpecificationSchema:name>STM001 - Sharing of Voyage Plan</ServiceSpecificationSchema:name>

<ServiceSpecificationSchema:text>As part of the Voyage Information Object the Voyage Plan (VP) can be shared among the different parties participating in a ships voyage. The ship/shipping company is the information owner of the VP and as such chooses which actors that should be granted access to the voyage plan.</ServiceSpecificationSchema:text>

<ServiceSpecificationSchema:rationale></ServiceSpecificationSchema:rationale>

<ServiceSpecificationSchema:reference>Use-Case 2: Sharing of Voyage Plan</ServiceSpecificationSchema:reference>

<ServiceSpecificationSchema:authorInfos>

<ServiceSpecificationSchema:authorInfo>

<ServiceSpecificationSchema:id>urn:mrn:stm:user:sma:pelo</ServiceSpecificationSchema:id>

<ServiceSpecificationSchema:name>Per Löfbom</ServiceSpecificationSchema:name>

<ServiceSpecificationSchema:description>Solution architect</ServiceSpecificationSchema:description>

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

<ServiceSpecificationSchema:organizationId>urn:mrn:stm:org:sma</ServiceSpecificationSchema:organizationId>

</ServiceSpecificationSchema:authorInfo>

</ServiceSpecificationSchema:authorInfos>

</ServiceSpecificationSchema:requirement>

</ServiceSpecificationSchema:requirements>

<ServiceSpecificationSchema:authorInfos>

<ServiceSpecificationSchema:authorInfo>

<ServiceSpecificationSchema:id>urn:mrn:stm:user:sma:pelo</ServiceSpecificationSchema:id>

<ServiceSpecificationSchema:name>Per Löfbom</ServiceSpecificationSchema:name>

<ServiceSpecificationSchema:description>Solution architect</ServiceSpecificationSchema:description>

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

<ServiceSpecificationSchema:organizationId>urn:mrn:stm:org:sma</ServiceSpecificationSchema:organizationId>

</ServiceSpecificationSchema:authorInfo>
</ServiceSpecificationSchema:authorInfos>
<ServiceSpecificationSchema:serviceDataModel>
<ServiceSpecificationSchema:definitionAsXSD>

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="messageType">
    <xs:annotation>
      <xs:documentation>Type of messages (including version)</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="RTZ">
        <xs:annotation>
          <xs:documentation>Route Exchange Format in XML</xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="TXT">
        <xs:annotation>
          <xs:documentation>STM Textmessage in XML</xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="S124">
        <xs:annotation>
          <xs:documentation>S124 in XML</xs:documentation>
        </xs:annotation>
      </xs:enumeration>
      <xs:enumeration value="PCM">
        <xs:annotation>
          <xs:documentation>Port Call Message in XML</xs:documentation>
        </xs:annotation>
      </xs:enumeration>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="DeliveryAck" type="DeliveryAck"/>
  <xs:complexType name="DeliveryAck">
    <xs:annotation>
      <xs:documentation>Object for message ACK</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="id" type="xs:string" minOccurs="1" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>Id for the ACK</xs:documentation>
        </xs:annotation>
      </xs:element>
```

<xs:element name="referenceId" type="URN" minOccurs="1" maxOccurs="1">
<xs:annotation>
<xs:documentation>Reference to delivered message according to the STM MRN identifier. For example an unique voyage identifier: urn:mrn:stm:voymgt:uvid:<organizationId>:<local voyagenumber></xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="timeOfDelivery" type="xs:dateTime" minOccurs="1" maxOccurs="1">
<xs:annotation>
<xs:documentation>Time of delivery</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="fromId" type="URN" minOccurs="1" maxOccurs="1">
<xs:annotation>
<xs:documentation>Identity of source (sender) of message that have been delivered according to the STM MRN identifier. Example: urn:mrn:stm:org:<organizationId></xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="fromName" type="xs:string" minOccurs="1" maxOccurs="1">
<xs:annotation>
<xs:documentation>Friendly name of sender</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="toId" type="URN" minOccurs="1" maxOccurs="1">
<xs:annotation>
<xs:documentation>Identity of target (receipient) of message delivery according to the STM MRN identifier. Example: urn:mrn:stm:org:<organizationId></xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="toName" type="xs:string" minOccurs="1" maxOccurs="1">
<xs:annotation>
<xs:documentation>Friendly name of recipient</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="ackResult" type="xs:string" minOccurs="1" maxOccurs="1"/>
</xs:sequence>
</xs:complexType>
<xs:element name="GetVPResponseObject" type="GetVPResponseObject"/>
<xs:complexType name="GetVPResponseObject">
<xs:sequence>
<xs:element name="code" type="xs:int" minOccurs="1" maxOccurs="1">
<xs:annotation>
<xs:documentation>Status code (20x, 40x)</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="subject" type="xs:string" minOccurs="1" maxOccurs="1">
<xs:annotation>

```
            <xs:documentation>Message e.g. error text</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="lastInteractionTime" type="xs:dateTime" minOccurs="1" maxOccurs="1">
          <xs:annotation>
            <xs:documentation>Last interaction time with ship or ship representative.</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="route" type="rtz:route" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="route" type="route" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="routeStatus">
      <xs:restriction base="xs:int">
        <xs:enumeration value="Original">
          <xs:annotation>
            <xs:documentation>1</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="Planned_for_voyage">
          <xs:annotation>
            <xs:documentation>2</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="Optimized">
          <xs:annotation>
            <xs:documentation>3</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="Cross_Checked">
          <xs:annotation>
            <xs:documentation>4</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="Safety_Checked">
          <xs:annotation>
            <xs:documentation>5</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="Approved">
          <xs:annotation>
            <xs:documentation>6</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
```

```
        <xs:enumeration value="Used_for_monitoring">
          <xs:annotation>
            <xs:documentation>7</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
        <xs:enumeration value="Inactive">
          <xs:annotation>
            <xs:documentation>8</xs:documentation>
          </xs:annotation>
        </xs:enumeration>
      </xs:restriction>
    </xs:simpleType>
    <xs:element name="GetVoyagePlanObject" type="GetVoyagePlanObject"/>
    <xs:complexType name="GetVoyagePlanObject">
      <xs:sequence>
        <xs:element name="UVID" type="URN" minOccurs="0" maxOccurs="1"/>
        <xs:element name="routeStatus" type="routeStatus" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
    <xs:element name="GetSubscriptionResponseObj" type="GetSubscriptionResponseObj"/>
    <xs:complexType name="GetSubscriptionResponseObj">
      <xs:annotation>
        <xs:documentation>Object with array of dataId, in MRN format, such as a list of UVIDs.</xs:documentation>
      </xs:annotation>
      <xs:sequence>
        <xs:element name="dataId" type="MRN" minOccurs="0" maxOccurs="unbounded">
          <xs:annotation>
            <xs:documentation>Data id in MRN format, such as UVID.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
</xs:schema>
                    </ServiceSpecificationSchema:definitionAsXSD>
          </ServiceSpecificationSchema:serviceDataModel>
          <ServiceSpecificationSchema:serviceInterfaces>
                    <ServiceSpecificationSchema:serviceInterface>
                              <ServiceSpecificationSchema:name>String</ServiceSpecificationSchema:name>
                              <ServiceSpecificationSchema:description>String</ServiceSpecificationSchema:description>

          <ServiceSpecificationSchema:dataExchangePattern>PUBLISH_SUBSCRIBE</ServiceSpecificationSchema:dataExchangePatter
n>
                              <ServiceSpecificationSchema:operations>
                                        <ServiceSpecificationSchema:operation>
                                                  <ServiceSpecificationSchema:name>String</ServiceSpecificationSchema:name>
```

```
<ServiceSpecificationSchema:description>String</ServiceSpecificationSchema:description>
                                        <ServiceSpecificationSchema:returnValueType>

<ServiceSpecificationSchema:typeReference>String</ServiceSpecificationSchema:typeReference>
                                        </ServiceSpecificationSchema:returnValueType>
                                        <ServiceSpecificationSchema:parameterTypes>
                                                <ServiceSpecificationSchema:parameterType>

<ServiceSpecificationSchema:typeReference>String</ServiceSpecificationSchema:typeReference>
                                                </ServiceSpecificationSchema:parameterType>
                                        </ServiceSpecificationSchema:parameterTypes>
                                </ServiceSpecificationSchema:operation>
                        </ServiceSpecificationSchema:operations>
                        <ServiceSpecificationSchema:consumerInterface>
                                <ServiceSpecificationSchema:name>String</ServiceSpecificationSchema:name>
                                <ServiceSpecificationSchema:description>String</ServiceSpecificationSchema:description>
                                <ServiceSpecificationSchema:operations>
                                        <ServiceSpecificationSchema:operation>

<ServiceSpecificationSchema:name>String</ServiceSpecificationSchema:name>

<ServiceSpecificationSchema:description>String</ServiceSpecificationSchema:description>
                                                <ServiceSpecificationSchema:returnValueType>

<ServiceSpecificationSchema:typeReference>String</ServiceSpecificationSchema:typeReference>
                                                </ServiceSpecificationSchema:returnValueType>
                                                <ServiceSpecificationSchema:parameterTypes>
                                                        <ServiceSpecificationSchema:parameterType>

<ServiceSpecificationSchema:typeReference>String</ServiceSpecificationSchema:typeReference>
                                                        </ServiceSpecificationSchema:parameterType>
                                                </ServiceSpecificationSchema:parameterTypes>
                                        </ServiceSpecificationSchema:operation>
                                </ServiceSpecificationSchema:operations>
                        </ServiceSpecificationSchema:consumerInterface>
                </ServiceSpecificationSchema:serviceInterface>
        </ServiceSpecificationSchema:serviceInterfaces>
</ServiceSpecificationSchema:serviceSpecification>
```


***

ANNEX 2

Service Design Description for the Voyage Information Service

REST

Contents

Table of figures

List of tables

Hittar inga figurförteckningsposter.

1 Introduction

1.1 Purpose of the Document

The purpose of this service design description document is to provide a detailed description of the Voyage Information service (see ***Fel! Hittar inte referenskälla.***), realized by using the REST technology, according to the guidelines given in [1]. It describes a well-defined baseline of the service design by clearly identifying the service design version.

The aim is to document the key aspects of the Voyage Information REST service technical design. This includes:

- identification and summary of the service design
    - reference to the service specification
    - identification of the service design
- identification and summary of chosen technology
- detailed description about the realization of each service interface and service operation
    - mapping of interfaces to the chosen technology
    - mapping of operations to the chosen technology
    - mapping of the message exchange patterns to the chosen technology
- detailed description of the physical data model
    - mapping to the service data model of the service specification.

1.2 Intended Readership

This service design description document is intended to be read by service architects, designers, system engineers and developers in charge of designing and developing an instance of the Voyage Information REST service.

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

1.3 Inputs from Other Projects

No information.

2 Service Design Identification

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

Name	*Voyage Information Service REST Design, SMA*
ID	*urn:mrn:stm:service:design:sma:vis-rest-2.2*
Version	*2.2*
Technology	*REST*
Service Specification ID	*urn:mrn:stm:service:specification:sma:vis*
Service Specification Version	*2.2*
Description	*Exchange Voyage information constituted of voyage plans (RTZ), text message (STM Text Message) and areas (S-124)*
Keywords	*Voyage Information Service, VIS, REST, RTZ, TXT, S-124,ROS,RCS,EMS,Route Exchange*
Architect(s)	*Per Löfbom, SMA* *Per de Flon, SMA* *Mikael Olofsson, SMA (Combitech)*
Status	*Released*

3 Technology Introduction

This service design is realized using RESTful API's described in JSON using the Swagger interface.

3.1 REST

REST (REpresentational State Transfer) is an architectural style, and an approach to communications that is often used in the development of Web services. The use of REST in VIS is preferred over the more heavyweight SOAP (Simple Object Access Protocol) style because REST does not leverage as much bandwidth, which makes it a better fit for use in communication between vessels and shore based representation of the same.

REST, which typically runs over HTTP (Hypertext Transfer Protocol), has several architectural constraints:

- *Decoupling* – Decouples consumers from producers which suits SeaSWIM decentralized architecture well.
- *Stateless existence* – Also a good prerequisite for a decentralized architecture design.
- *Able to leverage a cache* – Probably less important in SeaSWIM since most of the interaction is between machines, although for services with man-machine interfaces this is of importance.
- *Leverages a layered system* – SeaSWIM is dependant on good scaling capabilities which has REST support.
- *Leverages a uniform interface* – Again since SeaSWIM defines the available services centrally in a Service registry this constraint supports implementations being decoupled from the services they provide.

3.2 Swagger

Swagger is a simple yet powerful representation of RESTful API. With the largest ecosystem of API tooling on the planet, thousands of developers are supporting Swagger in almost every modern programming language and deployment environment. With a Swagger-enabled API, you get interactive documentation, client and server SDK generation together with discoverability.

A reference to provided Swagger JSON file is included in the Service Design XML description.

References:

- *Fielding, Roy Thomas (2000). "Chapter 5: Representational State Transfer (REST)". Architectural Styles and the Design of Network-based Software Architectures (Ph.D.). University of California, Irvine.*
- *Richardson, Leonard; Ruby, Sam (2007), RESTful Web service, O'Reilly Media, ISBN978-0-596-52926-0, retrieved 18 January 2011.*

- *Richardson, Leonard; Amundsen, Mike (2013), RESTful Web APIs, O'Reilly Media, ISBN978-1-449-35806-8, retrieved 15 September 2015*
- Swagger Open API specification - http://swagger.io/specification/

4 Service Design Overview

4.1 Service Interface Design

The main purpose with VIS is to handle the communication around voyage information and the main artefact Voyage Plan (VP) in RTZ format. VIS implements methods for exposing new and updated VP's and to consume external VP's. VIS also supports subscription of voyage plans.

In addition to voyage plans (RTZ), VIS also supports exchange of STM Text Message and area message (S-124).

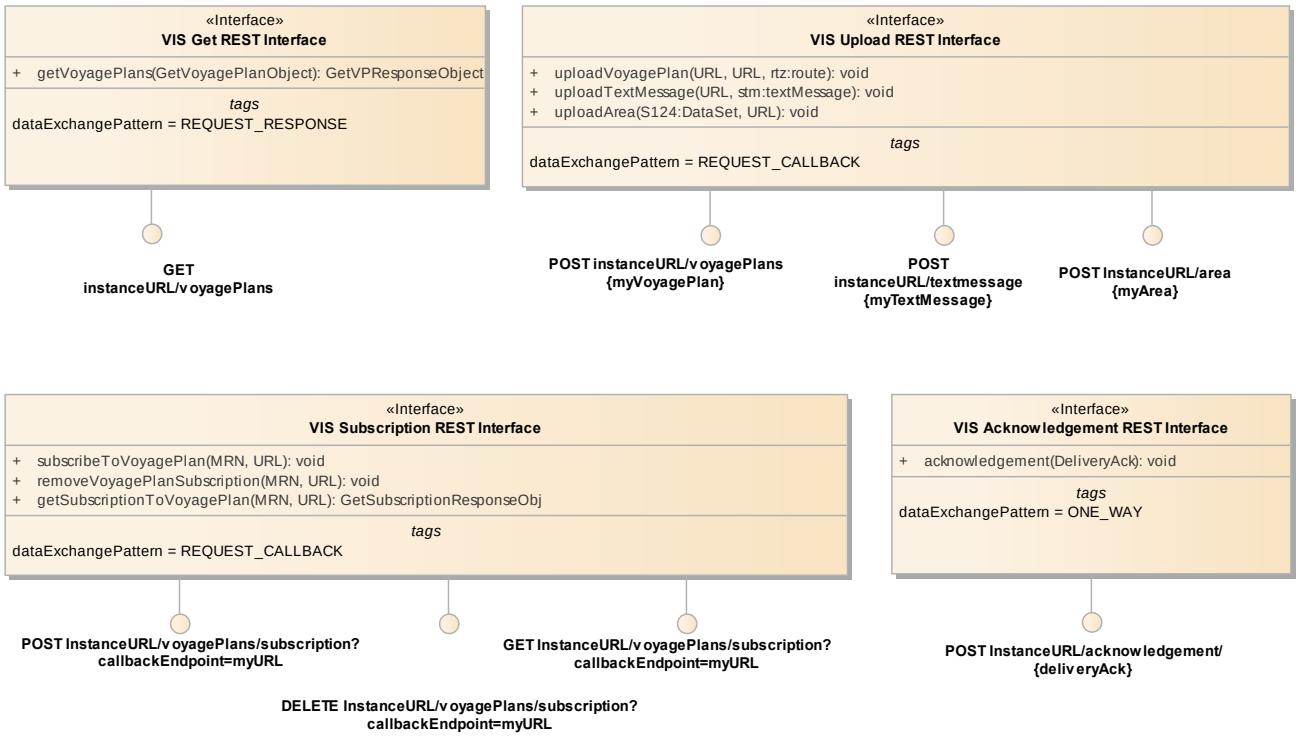


Figure 1 Service Design Overview

4.1.1 Service Interfaces

The table below shows the REST interface designed for the corresponding operation in the Service Specification.

In the table, only the mandatory parameters are shown. For detailed description of each operation including optional parameters, see chapter 6.

<table>
<tr><td>Service Specification</td><td colspan="2">Service Design</td></tr>
<tr><td>Service Interface</td><td colspan="2">Service REST Operation</td></tr>
<tr><td rowspan="2">VIS Get REST Interface</td><td>REST</td><td>Operation-id</td></tr>
<tr><td>GET instanceURL/voyagePlans</td><td>getVoyagePlans</td></tr>
<tr><td>Service Interface</td><td colspan="2">Service REST Operation</td></tr>
<tr><td rowspan="4">VIS Upload REST Interface</td><td>REST</td><td>Operation-id</td></tr>
<tr><td>POST instanceURL/voyagePlans{myVoyagePlan}</td><td>uploadVoyagePlan</td></tr>
<tr><td>POST instanceURL/textmessage{myTextMessage}</td><td>uploadTextMessage</td></tr>
<tr><td>POST InstanceURL/area{myArea}</td><td>uploadArea</td></tr>
<tr><td>Service Interface</td><td colspan="2">Service REST Operation</td></tr>
</table>

<table>
<tr><th>Service Specification</th><th colspan="2">Service Design</th></tr>
<tr><th>Service Interface</th><th colspan="2">Service REST Operation</th></tr>
<tr><td rowspan="4">VIS Subscription REST Interface</td><td>REST</td><td>Operation-id</td></tr>
<tr><td>POST InstanceURL/voyagePlans/subscription?callbackEndpoint=myURL</td><td>subscribeToVoyagePlan</td></tr>
<tr><td>GET InstanceURL/voyagePlans/subscription?callbackEndpoint=myURL</td><td>getSubscriptionToVoyagePlan</td></tr>
<tr><td>DELETE InstanceURL/voyagePlans/subscription?callbackEndpoint=myURL</td><td>removeVoyagePlanSubscription</td></tr>
<tr><th>Service Interface</th><th colspan="2">Service REST Operation</th></tr>
<tr><td rowspan="2">VIS Acknowledgement REST Interface</td><td>REST</td><td>Operation-id</td></tr>
<tr><td>POST InstanceURL/acknowledgement/{deliveryAck}</td><td>acknowledgement</td></tr>
</table>

5 Physical Data Model

The following version of payload formats are valid in this version of VIS Design;

- RTZ v1.1 with STM Extension v1.0.0
  http://stmvalidation.eu/schemas/
- S124 v0.0.7
  http://stmvalidation.eu/schemas/
- TXT v1.3
  http://stmvalidation.eu/schemas/

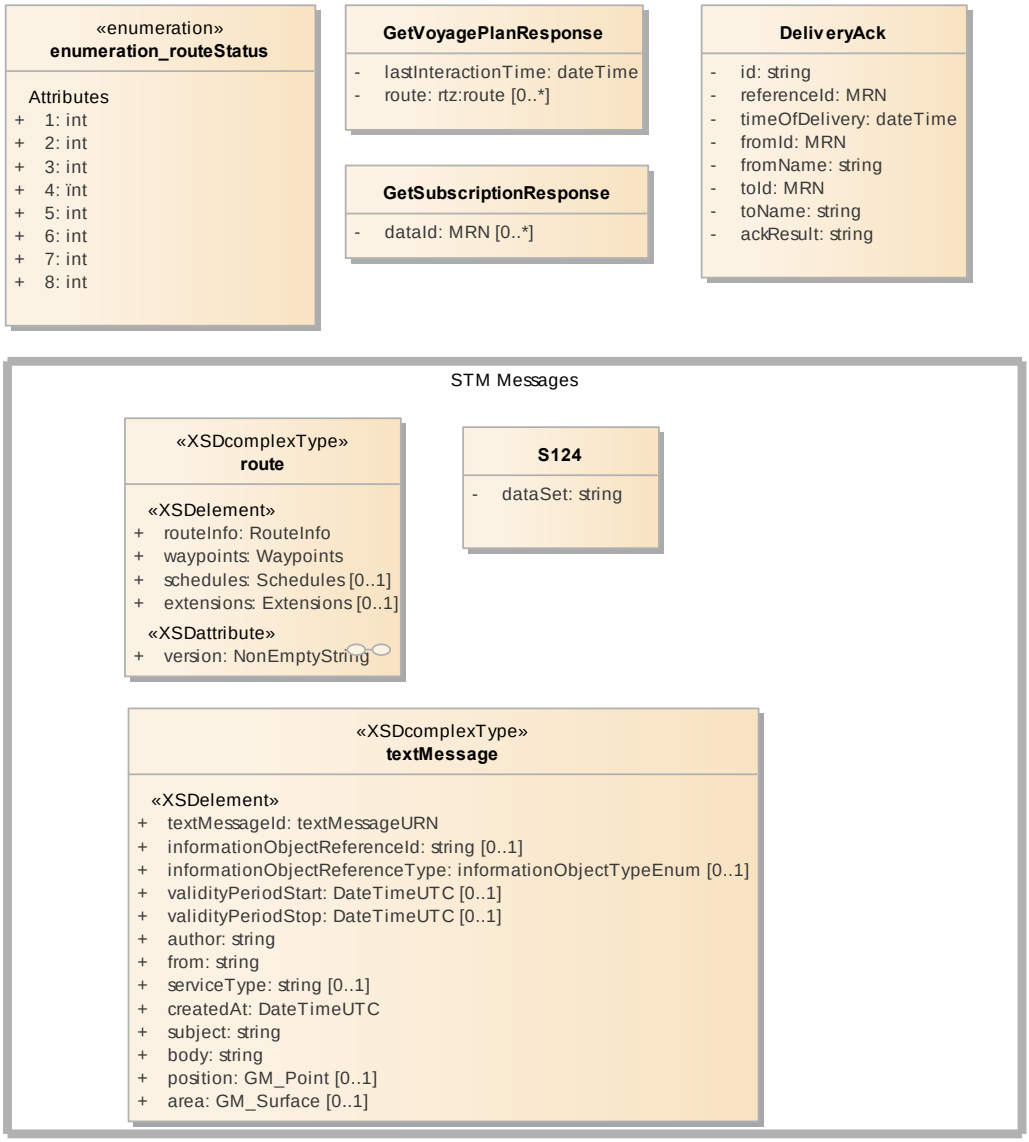


Figure 2 Physical Data Model

5.1.1 route

RTZ files contain a number of waypoints, followed with auxiliary schedules.

For detailed information, see http://stmvalidation.eu/schemas/

Element Name	Attributes		
route	**Name**	**Type**	**Description**
	routeInfo	RouteInfo	Generic route information.
	waypoints	Waypoints	A list of waypoints.
	schedules	Schedules	Optional list of schedules.
	extensions	Extensions	You can add extend RTZ by adding your own elements from another schema here.
	version	NonEmpty String	Format version

5.1.2 enumeration_routeStatus

Enumeration as string "1" to "8"

Element Name	Enumeration		
enumeration_routeStatus	**Name**	**Type**	**Description**
	1	int	Original
	2	int	Planned_for_voyage
	3	int	Optimized
	4	ïnt	Cross_Checked
	5	int	Safety_Checked
	6	int	Approved
	7	int	Used_for_monitoring
	8	int	Inactive

5.1.3 textMessage

Text message defined in STM project.

For detailed information, see http://stmvalidation.eu/schemas/

Element Name	Attributes		
textMessage	**Name**	**Type**	**Description**
	textMessageId	textMessageURN	Identifier of the text message, mandatory.
	informationObjectReferenceId	string	A reference to an information object, optional.
	informationObjectReferenceType	informationObjectTypeEnum	STM payload format reference, optional.
	validityPeriodStart	DateTimeUTC	Start of validity period in ISO 8601 format, optional.
	validityPeriodStop	DateTimeUTC	Stop of validity period in ISO 8601 format, optional.
	author	string	The message author, mandatory.
	from	string	The sending actor, mandatory.
	serviceType	string	The service type of the sender, optional.
	createdAt	DateTimeUTC	The message creation dateTime, mandatory.
	subject	string	The message subject, mandatory.
	body	string	The message body, mandatory.
	position	GM_Point	Geographic point, optional.
	area	GM_Surface	Geographic area, optional.

5.1.4 S124

S124 area message

For detailed information, see http://stmvalidation.eu/schemas/

Element Name	Attributes		
S124	**Name**	**Type**	**Description**
	dataSet	string	S124 area message as defined at STM Developer Forum site http://stmvalidation.eu.

5.1.5 DeliveryAck

Object for message ACK

Element Name	Attributes		
DeliveryAck	**Name**	**Type**	**Description**
	id	string	Id for the ACK
	referenceId	MRN	Reference to delivered message according to the STM MRN identifier. For example an unique voyage identifier: urn:mrn:stm:voymgt:uvid:<organizationId>:<local voyagenumber>
	timeOfDelivery	dateTime	Time of delivery in UTC (e.g. 2017-03-29T11:33:00Z)
	fromId	MRN	Identity of source (sender) of message that have been delivered according to the STM MRN identifier. Example: urn:mrn:stm:org:<organizationId>
	fromName	string	Friendly name of sender
	toId	MRN	Identity of target (receipient) of message delivery according to the STM MRN identifier. Example: urn:mrn:stm:org:<organizationId>
	toName	string	Friendly name of recipient
	ackResult	string	

5.1.6 GetVoyagePlanResponse

Response object from request for voyage plan

Element Name	Attributes		
GetVoyagePlanResponse	**Name**	**Type**	**Description**
	lastInteractionTime	dateTime	Last interaction time with private application in UTC (e.g. 2017-03-29T11:33:00Z)
	route	rtz:route	Sequence of 0 or more route messages (RTZ) in XML format

5.1.7 GetSubscriptionResponse

Object with array of dataId, in MRN format, such as a list of UVIDs.

Element Name	Attributes		
GetSubscriptionResponse	**Name**	**Type**	**Description**
	dataId	MRN	Array of data id in MRN format, such as UVID.

6 Service Interface Design

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

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

6.1 Voyage Information Service REST

The Voyage Information Service provides interfaces for requesting voyage plan (Get), requesting subscription of voyage plans (Subscription) and to upload voyage plan, text message and areas (Upload).

6.1.1 VIS Get REST Interface

Message exchange pattern: REQUEST_RESPONSE

Facilitates operations for requesting a Voyage Plan.

GET /voyagePlans

Operation id from specification: getVoyagePlans()

Operation for requesting Voyage Plans.

Request type GET

Endpoint path: /voyagePlans

In Parameters

uvid is optional, e.g. urn:mrn:stm:voyage:id:sma:voyage-001

routeStatus is optional, e.g. 7

In Body

none

Return

http code

If http code 200

GetVoyagePlanResponse in JSON

lastInteractionTime containing last known interaction with private application

sequence of RTZ (0..*) in XML format (text/xml)

If http code 40x

Optional message as string

Returns the following HTTP response codes and messages:

200=Successful

400=Bad Request

401=Unauthorized

403=Forbidden

403=Not Found

500=Internal Server Error

Operation functionality

Depending on the provided parameters, the following will be returned:

GET /voyagePlans

No parameters given;

Return the latest published voyage plan with routeStatus "not inactivated" (routeStatus != "8") for all UVIDs the requester have access to.

If two or more voyage plans have routeStatus "Used for monitoring" (routeStatus=="7") for one ship, then only the latest published of them shall be returned.

GET /voyagePlans?UVID=myUVID

Return the latest published message with requested UVID if the requester have access.

GET /voyagePlans?routeStatus=myRouteStatus

Return the latest published message with requested routeStatus the requester have access to.

If two or more voyage plans have routeStatus "Used for monitoring" (routeStatus=="7") for one ship, then only the latest published of them shall be returned.

GET /voyagePlans?UVID=myUVID&routeStatus=myRouteStatus

Return the latest published message with requested UVID and routeStatus the requester have access.

6.1.2 VIS Upload REST Interface

Message exchange pattern: REQUEST_RESPONSE

Facilitates operations for uploading a Voyage Plan, Text Message or Polygon/Area.

POST /voyagePlans{myVoyagePlan}

Operation id from specification: uploadVoyagePlan()

Facilitates sending (uploading) a voyage plan to VIS to be forwarded to private application.

If endpoint provided for deliveryACK, an ACK will be sent when message has been delivered to private application.

If endpoint provided for callback, a result is expected to be uploaded to callback endpoint. E.g. when ship requesting route optimization, the ship may provide the ships endpoint to inform the route optimization provider that the optimized route is expected on this endpoint upload operation(s), voyage plan, text message and/or area message.

Request type POST

Endpoint path: /voyagePlans

In Parameters

deliveryAckEndPoint is optional, e.g. "https://stm.eu/vis/imo1234567"

callbackEndpoint is optional, e.g. "https://stm.eu/vis/imo1234567"

In Body

voyageplan (RTZ) in XML format (text/xml) is mandatory

Return

http code

optional information such as id on uploaded message

Returns the following HTTP response codes and messages:

200=Successful

400=Bad Request

401=Unauthorized (the user cannot be authenticated in Identity Registry)

403=Forbidden

500=Internal Server Error

Operation functionality

The voyage plan is checked against the RTZ schema and internal rules

- In addition to the RTZ schema the following attributes is mandatory; vesselVoyage and routeStatus

If delivery ACK is requested, VIS sends a delivery ACK to the requested endpoint when VIS has delivered the uploaded message to private application.

POST /textmessage{myTextMessage}

Operation id from specification: uploadTextMessage()

Facilitates sending (uploading) a text message to VIS to be forwarded to private application

If endpoint provided for deliveryACK, an ACK will be sent when message has been delivered to private application.

Request type POST

Endpoint path: /textmessage

In Parameters

deliveryAckEndPoint is optional, e.g. "https://stm.eu/vis/imo1234567"

In Body

text message in STM Text Message in XML format (text/xml) is mandatory

Return

http code

optional information such as id on uploaded message

Returns the following HTTP response codes and messages:

200=Successful

400=Bad Request

401=Unauthorized (the user cannot be authenticated in Identity Registry)

403=Forbidden

500=Internal Server Error

Operation functionality

The textMessage is checked against the textMessagew schema

If delivery ACK is requested, VIS sends a delivery ACK to the requested endpoint when VIS has delivered the uploaded message to private application.

POST /area{myArea}

Operation id from specification: uploadArea()

Facilitates sending (uploading) a polygon (area) to VIS to be forwarded to private application

If endpoint provided for deliveryACK, an ACK will be sent when message has been delivered to private application.

Request type POST

Endpoint path: /area

In Parameters

deliveryAckEndPoint is optional, e.g. "https://stm.eu/vis/imo1234567"

In Body

Area in S-124 (dataset) in XML format (text/xml) is mandatory

Return

http code

optional information such as id on uploaded message

Returns the following HTTP response codes and messages:

200=Successful

400=Bad Request

401=Unauthorized (the user cannot be authenticated in Identity Registry)

403=Forbidden

500=Internal Server Error

Operation functionality

The message is checked against the area schema

If delivery ACK is requested, VIS sends a delivery ACK to the requested endpoint when VIS has delivered the uploaded message to private application.

6.1.3 VIS Subscription REST Interface

Message exchange pattern: REQUEST_CALLBACK

Facilitates operations for subscribing and unsubscribing to a Voyage Plan.

POST /voyagePlans/subscription?callbackEndpoint=

Operation id from specification: subscribeToVoyagePlan()

Operation for subscription of voyage plans. The operation will store the incoming callbackEndpoint and upload voyage plans to this interface whenever they are changed. The operation expects that the callbackEndpoint adhere to VIS uploadVoyagePlan interface (POST /voyagePlans). The subscription remains active until removed either by private application or by requester.

If UVID is not provided (is blank), VIS will try to set up a subscription to all "active" UVID with route with routeStatus 1-7 the requester has access to.

If there are 2 or more voyage plans with routeStatus="7" for one ship, only the latest published of them will generate a subscription.

I.e. if there are one VP with routeStatus=7 and one in routeStatus=3, subscription will be enabled for both UVIDs.

Request type POST

Endpoint path: /voyagePlans/subscription

In Parameters

callbackEndpoint is mandatory, e.g. "https://stm.eu/vis/imo1234567"

uvid is optional, e.g. "urn:mrn:stm:voyage:id:sma:voyage-001"

In Body

none

Return

http code

optional information such as id

Returns the following HTTP response codes and messages:

200=Successful

400=Bad Request

401=Unauthorized (the user cannot be authenticated in Identity Registry)

403=Forbidden (the user is not authorized to requested voyagePlan)

404=Not Found (the requested Voyage Plan is not found)

500=Internal Server Error

Operation functionality

Handle the subscription request according to authorization list

Send back the latest voyage plan if authorized

Send published voyage plans according to subscription until subscription is removed

DELETE /voyagePlans/subscription?callbackEndpoint=

Operation id from specification: removeVoyagePlanSubscription()

Remove subscription from the ship for my identity/callbackEndpoint.

Request type DELETE

Endpoint path: /voyagePlans/subscription

In Parameters

callbackEndpoint is mandatory

uvid is optional

In Body

none

Return:

http code

optional information such as id

Returns the following HTTP response codes and messages:

200=Successful

400=Bad Request

401=Unauthorized (the user cannot be authenticated in Identity Registry)

403=Forbidden (the user is not authorized to requested voyagePlan)

404=Not Found (the requested Voyage Plan is not found)

500=Internal Server Error

Operation functionality

The subscription attached to the callbackEndpoint is removed

GET /voyagePlans/subscription?callbackEndpoint=

Operation id from specification: getSubscriptionToVoyagePlan()

Get information on subscribed voyage plans.

Request type GET

Endpoint path: /voyagePlans/subscription

In Parameters

callbackEndpoint is mandatory

In Body

none

Return:

http code

If HTTP Code 200

GetSubscriptionResponse with array of dataId that the requester subscribes to

Returns the following HTTP response codes and messages:

200=Successful

400=Bad Request

401=Unauthorized (the user cannot be authenticated in Identity Registry)

403=Forbidden (the user is not authorized to requested voyagePlan)

404=Not Found (the requested Voyage Plan is not found)

500=Internal Server Error

Operation functionality

Return list of data identities related to the given callbackEndpoint and the requesters organisation identity

6.1.4 VIS Acknowledgement REST Interface

Message exchange pattern: ONE_WAY

POST /acknowledgement{deliveryAck}

Operation id from specification: acknowledgement()

Facilitates acknowledgement of e.g. uploaded message.

Request type POST

Endpoint path: /acknowledgement

In Parameters

none

In Body

DeliveryAck in JSON

Return

http code

optional information such as id

Returns the following HTTP response codes and messages:

200=Successful

400=Bad Request

401=Unauthorized (the user cannot be authenticated in Identity Registry)

403=Forbidden

500=Internal Server Error

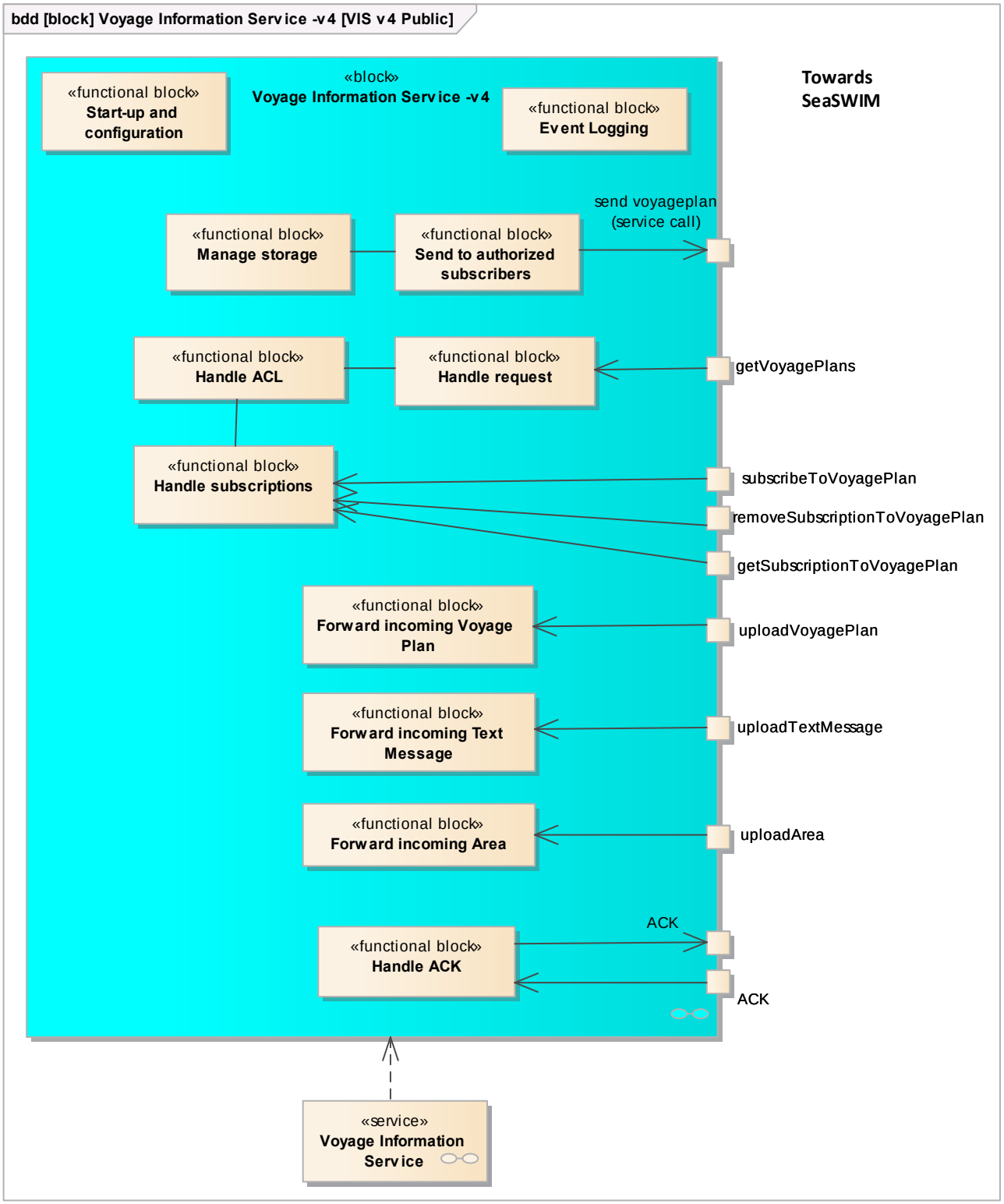
Operation functionality

Check and forward incoming acknowledgement to private application

7 Service Dynamic Behaviour

7.1 VIS SeaSWIM Interface

This section contains sequence diagrams related to VIS SeaSWIM interface.



7.1.1 VIS Get Interface

This section contains sequence diagrams related to VIS SeaSWIM Get interface.

A service consumer may request a voyage plan any time, either asking for a known UVID or just ask for any voyage plan published in VIS instance.

A service consumer can ask for voyage plans in a certain status, according to routeStatus enumeration, or ask for any voyage plan.

If the service consumer is not authorized by the "owner" of the VIS instance, a notification is forwarded to the "owner" and the service consumer don't get any voyage plans back until "owner" has authorized the service consumer.

If several unique voyage plans have been published in the VIS instance, all will be returned in the request. This enables the VIS to be deployed as a catalogue of voyage plans and routes.

However be aware that only zero or one (0..1) voyage plans in routeStatus=7 (Used for monitoring) can be returned from a VIS.

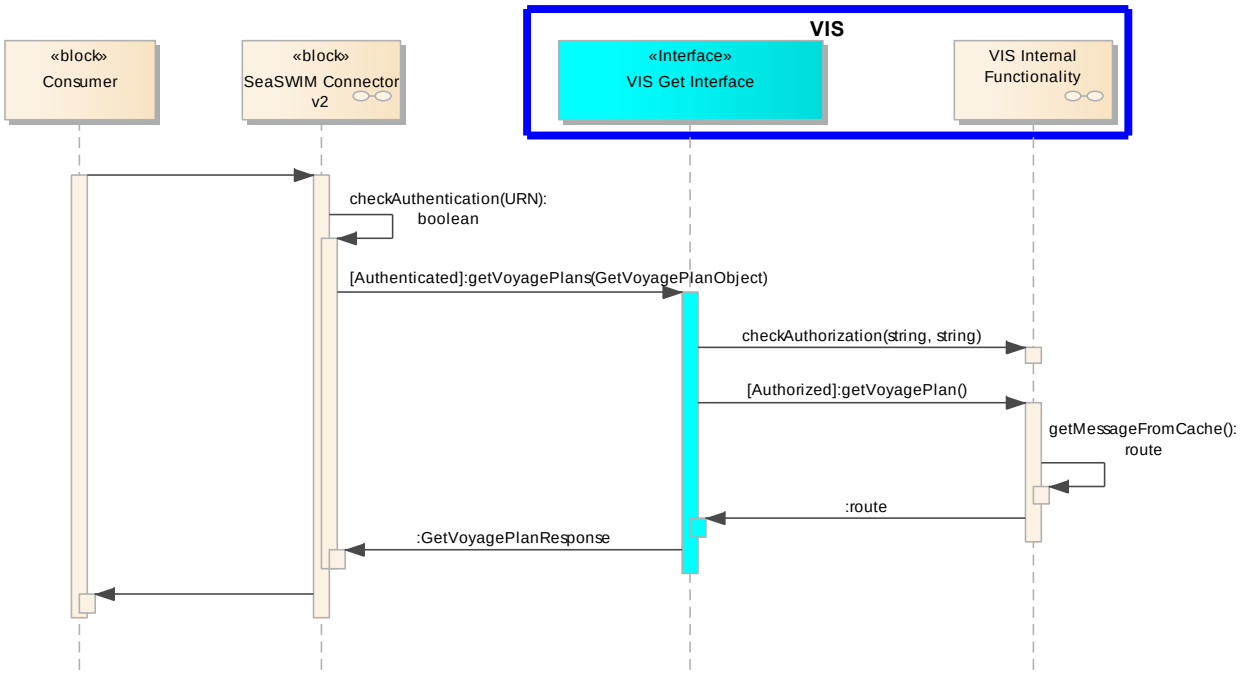
The service consumer must always check the routeStatus and act according the purpose by the service consumer. If the service consumer only wants "Used for monitoring", the request should be for routeStatus="7".

VIS will only handle requests from service consumer that are authenticated in STM.

Interaction getVoyagePlan

Message exchange pattern: REQUEST_RESPONSE

At receipt of request for a voyage plan in VIS, the user authorization is checked using an Access Control List (ACL). In case of successful authorization, the requested voyage plan(s) are fetched and returned to the calling service. If unsuccessful authorization, a non-authorized error response is sent. See further diagram Not authorized.


VIS
«block»
Consumer
«block»
SeaSWIM Connector
v2
«Interface»
VIS Get Interface
VIS Internal
Functionality
checkAuthentication(URN):
boolean
[Authenticated]:getVoyagePlans(GetVoyagePlanObject)
checkAuthorization(string, string)
[Authorized]:getVoyagePlan()
getMessageFromCache():
route
:route
:GetVoyagePlanResponse

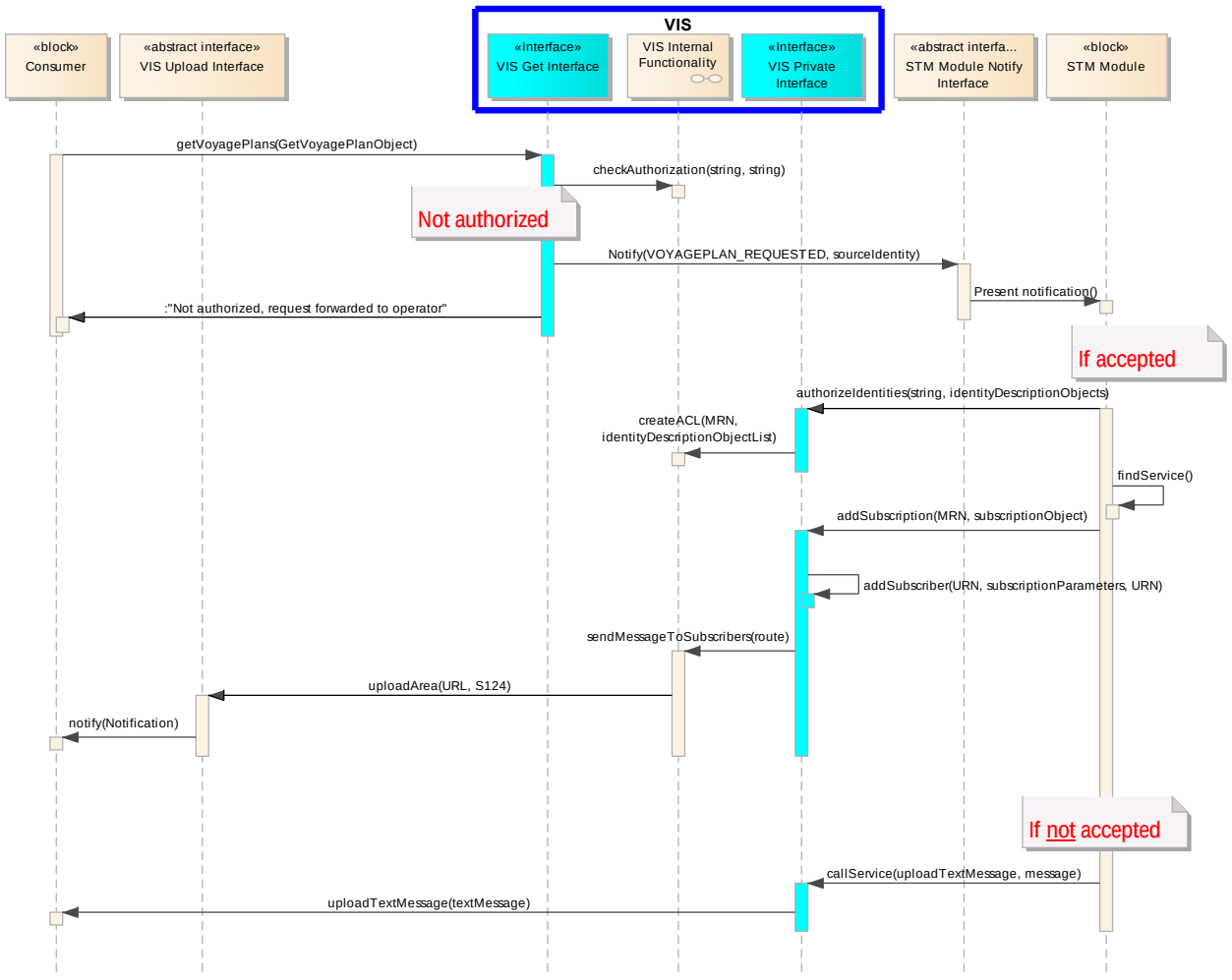
Service orchestration - Not authorized

In case the consumer is not authorized to requested data, the private application is notified hereof. The service consumer receives a message “Not authorized, request forwarded to operator”.

If no UVID is provided as parameter, a notification is sent to the private application only if the requester is not authorized to the voyage plan Used for monitoring (latest published voyage plan with routeStatus="7" for one ship).

It is then up to the user operating the private application to authorize the consumer to the requested voyage plan. Hereby creating a record in VIS ACL for the consumer identity.

In the case the operator chooses not to authorize the consumer, a textMessage can be sent to the consumer with notification of an unsuccessful authorization.



7.1.2 VIS Upload Interface

This section contains sequence diagrams related to VIS SeaSWIM Upload interface.

Asynchronous Acknowledgement can be requested.

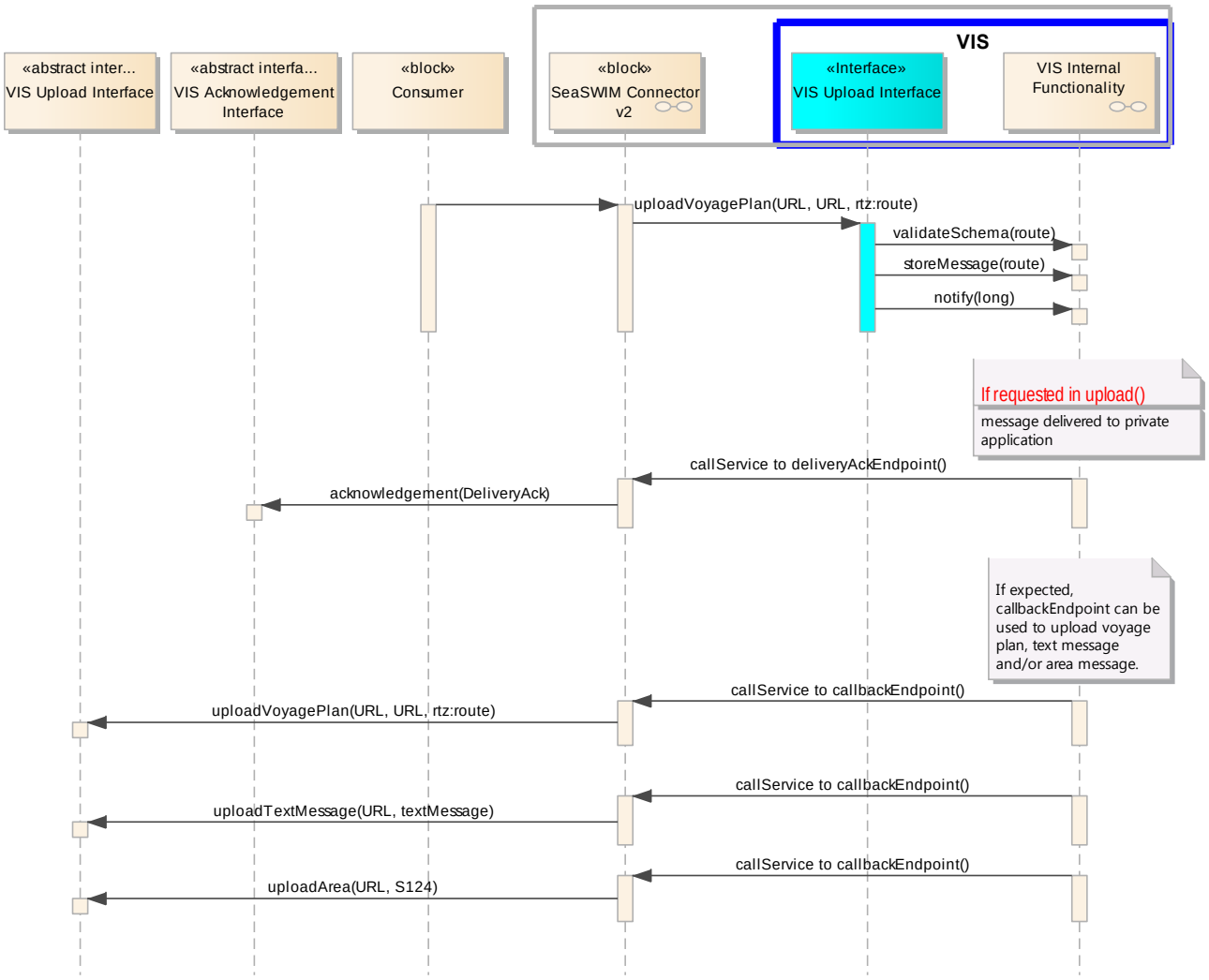
A service provider can always upload a voyageplan, text message or area message to VIS.

The service provider can always request an acknowledge message by providing an acknowledgement endpoint in the upload service request. When the message has been delivered to VIS private side, an acknowledgement is sent to the service provider. This acknowledgement however does not ensure that the message have reached the end user. This depends on the deployment on the private side where the STM Module may be an application on shore side and proprietary system to the end user, such as a ship.

VIS will only handle uploaded messages from service providers that are authenticated in STM.

Interaction uploadVoyagePlan

Message exchange pattern: REQUEST_CALLBACK

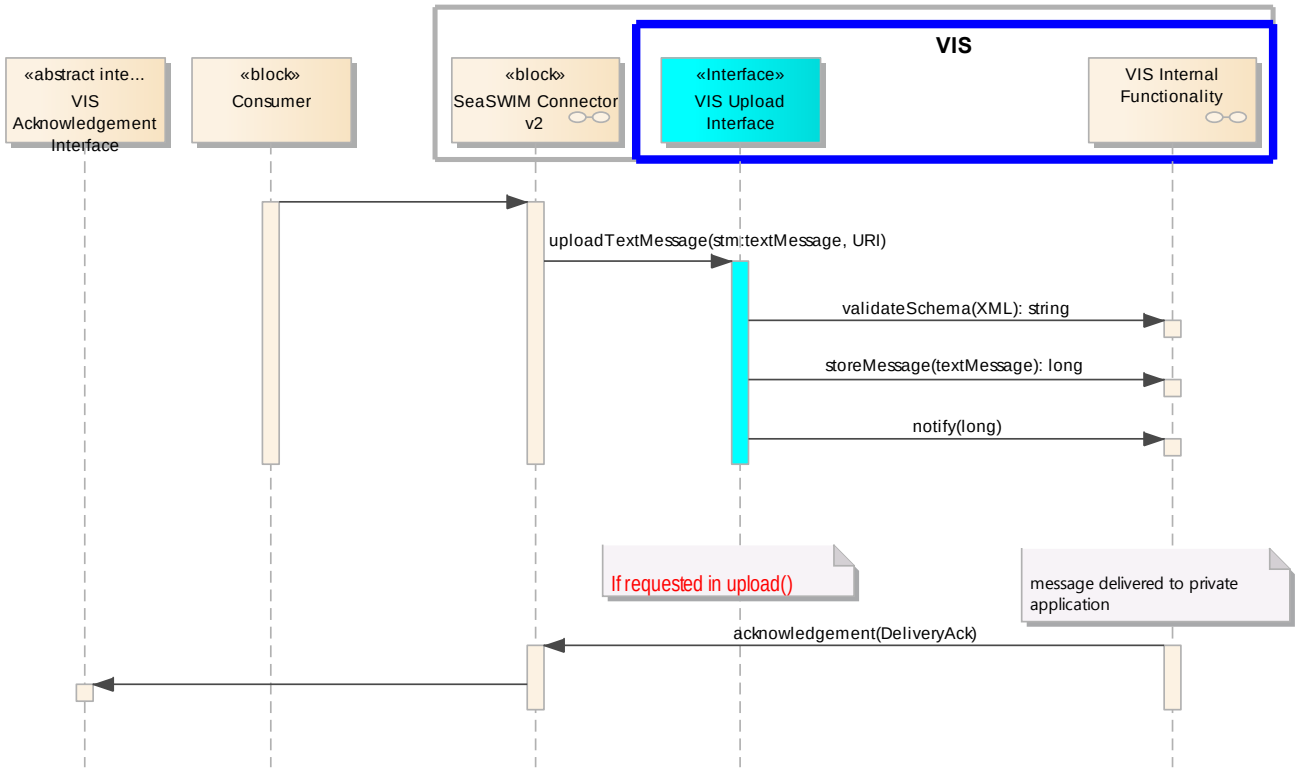

VIS
«abstract inter...
VIS Upload Interface
«abstract interfa...
VIS Acknowledgement Interface
«block»
Consumer
«block»
SeaSWIM Connector v2
«Interface»
VIS Upload Interface
VIS Internal Functionality
uploadVoyagePlan(URL, URL, rtz:route)
validateSchema(route)
storeMessage(route)
notify(long)
If requested in upload()
message delivered to private application
callService to deliveryAckEndpoint()
acknowledgement(DeliveryAck)
If expected, callbackEndpoint can be used to upload voyage plan, text message and/or area message.
callService to callbackEndpoint()
uploadVoyagePlan(URL, URL, rtz:route)
callService to callbackEndpoint()
uploadTextMessage(URL, textMessage)
callService to callbackEndpoint()
uploadArea(URL, S124)

Interaction uploadTextMessage

Message exchange pattern: REQUEST_CALLBACK

After receipt of a text message, the originating user organization is authenticated. Following a successful authentication the payload of the received message is validated against the schema.

In the case a deliveryAckEnpoint is supplied as parameter, an acknowledgement message is returned to the consumer after the delivery to the private application.

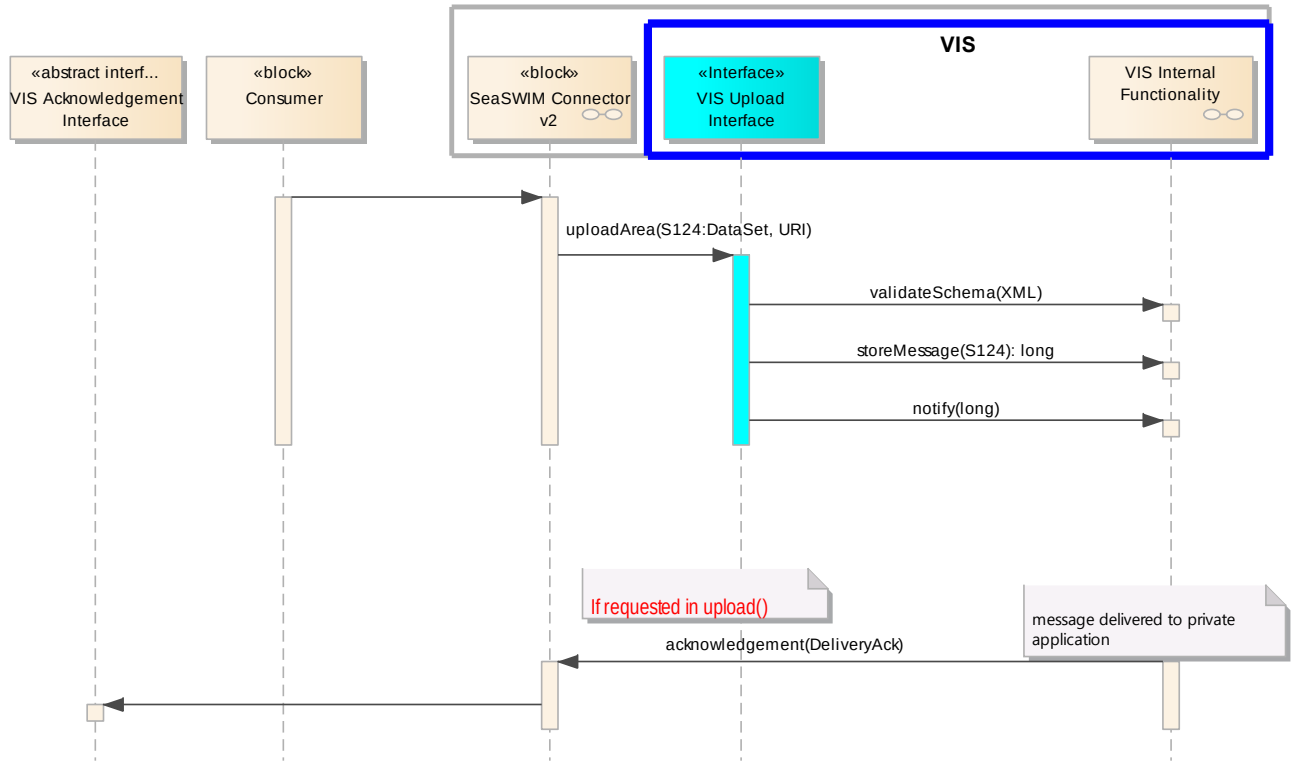


Interaction uploadArea

Message exchange pattern: REQUEST_CALLBACK

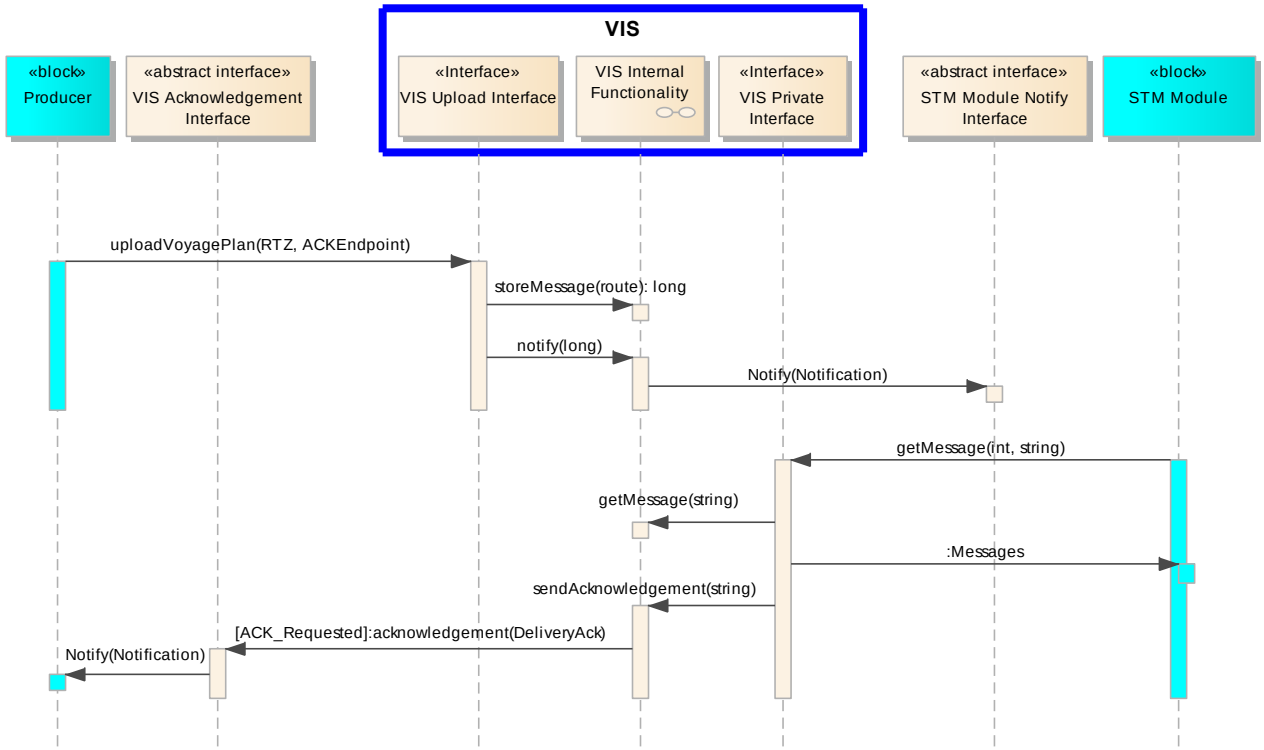
After receipt of a area message (S-124), the originating user organization is authenticated. Following a successful authentication the payload of the received message is validated against the schema.

In the case a deliveryAckEnpoint is supplied as parameter, an acknowledgement message is returned to the consumer after the delivery to the private application.



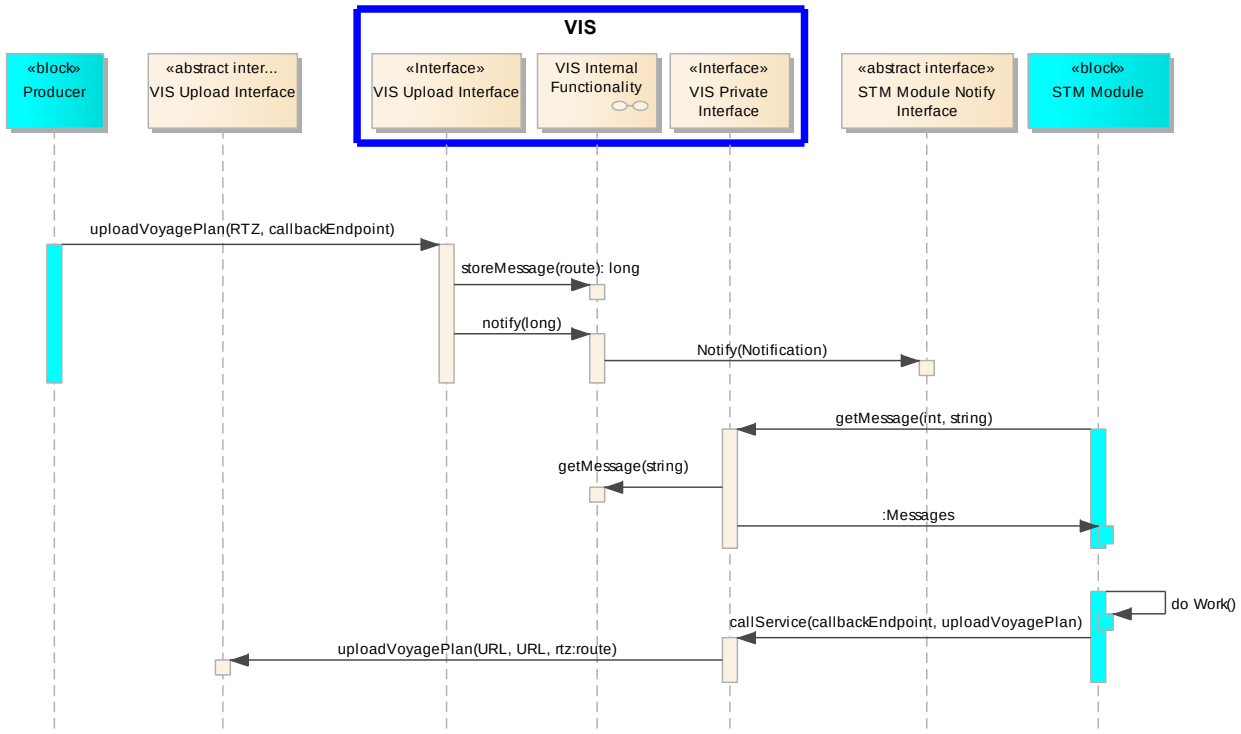
Service orchestration - Upload with ACK

The acknowledgement interface VIS exposes, is the endpoint for acknowledgement messages optionally requested by use of parameter deliveryAckEndpoint at upload of messages to VIS. The acknowledgement message is created for a specific message when it is successfully retrieved by the STM Module using VIS private interface getMessage, i.e. forwarded to the vessel. When the ACK is received, a notification is sent to the STM Module. The STM Module is responsible for checking and acting if ACK is not received.



Service orchestration - Upload with Callback

The callbackEndpoint can be provided for two purposes; inform that callback with information is expected, such as an optimized route; and inform to which endpoint the data is expected to be delivered to. The service responding on the provided callbackEndpoint shall still be an authenticated service.



7.1.3 VIS Subscription Interface

This section contains sequence diagrams related to VIS SeaSWIM Subscription interface.

A service consumer can always ask to subscribe to voyageplans. Either a known specific UVID or all voyageplans published in the VIS instance.

If the service consumer is not authorized by the "owner" of the VIS instance, a notification is forwarded to the "owner" and the service consumer don't get eny voyage plans back until "owner" has authorized the service consumer.

VIS will only handle requests from service consumer that are authenticated in STM.

Interaction subscribeToVoyagePlan

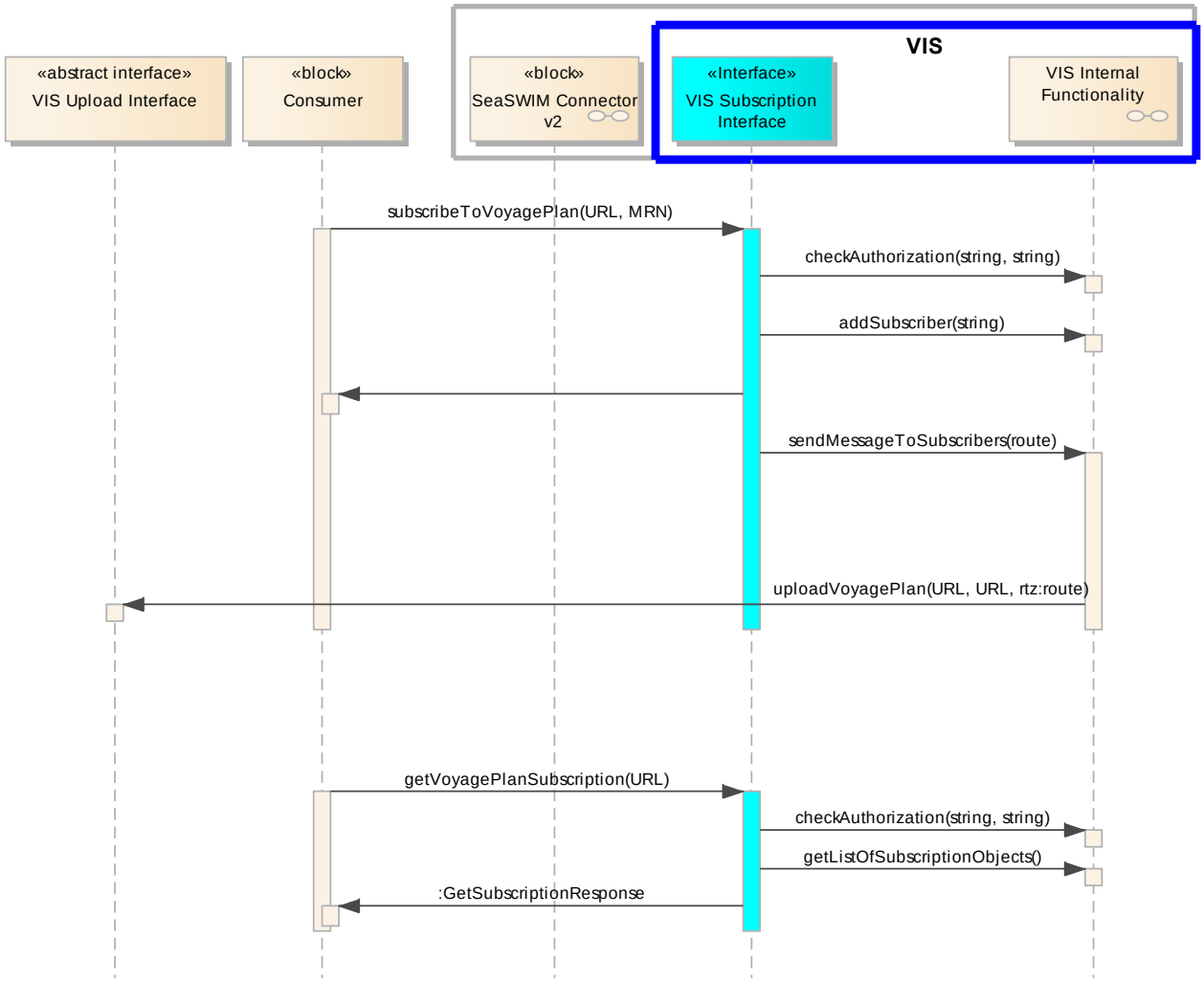
Message exchange pattern: REQUEST_CALLBACK

Consumer requests subscription by invoking interface subscribeToVoyagePlan providing the URI (address to consuming service uploadVoyagePlan interface - callbackEndpoint), optionally an uvid parameter can be passed for subscription on a specific voyagePlan. Following a successful authorization the subscriber identity and corresponding callbackEndpoint is stored in VIS dB subscription table and a voyagePlan is sent to the added subscriber. Every time a voyagePlan is published in VIS, the voyagePlan is forwarded to all selected subscribers.

If UVID is not provided (is blank), VIS will try to set up a subscription to all "active" UVID with route with routeStatus 1-7 the requester has access to.

If there are 2 ore more voyage plans with routeStatus="7" for one ship, only the latest published of them will generate a subscription.

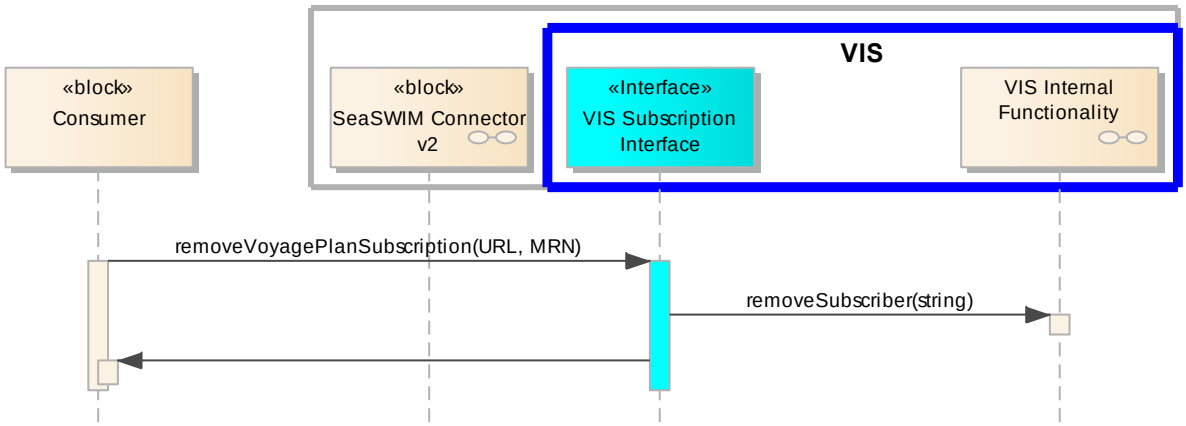
I.e. if there are one VP with routeStatus=7 and one in routeStatus=3, subscription will be enabled for both UVIDs.


VIS
«abstract interface»
VIS Upload Interface
«block»
Consumer
«block»
SeaSWIM Connector
v2
«Interface»
VIS Subscription
Interface
VIS Internal
Functionality
subscribeToVoyagePlan(URL, MRN)
checkAuthorization(string, string)
addSubscriber(string)
sendMessageToSubscribers(route)
uploadVoyagePlan(URL, URL, rtz:route)
getVoyagePlanSubscription(URL)
checkAuthorization(string, string)
getListOfSubscriptionObjects()
:GetSubscriptionResponse

Interaction removeVoyagePlanSubscription

Message exchange pattern: ONE_WAY

At removal of a subscription the removeVoyagePlanSubscription is invoked by the consumer. Parameters are the consumer callBackendpoint (mandatory) and optionally a specific uvid. At receipt of the subscription removal request VIS deletes all subscriptions for the callBackendpoint or a specific subscription for an uvid. In response to the subscription removal request a responseObj is returned with statusCode=200, successful.

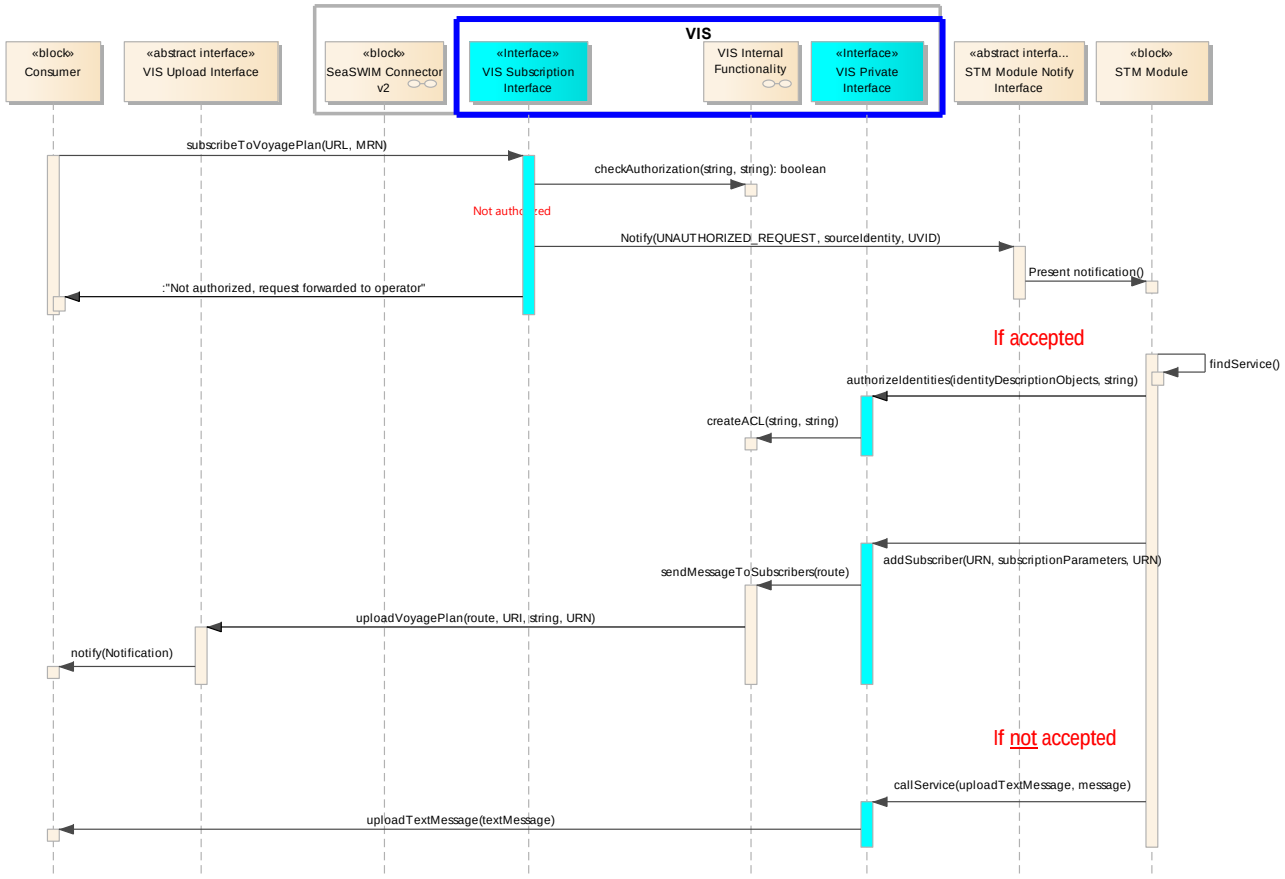


Service orchestration - Not authorized

In case the consumer is not authorized the STM Module operator onboard the vessel is notified hereof (message includes the consumer STM identity). The consumer receives a message “Not authorized request forwarded to operator”.

If authorized it is up to the user operating the STM Module to authorize the consumer to the requested voyage plan using VIS private interface authorizeIdentities. Hereby creating a record in VIS ACL for the consumer identity. VIS then searches SeaSWIM service registry in order to find the consuming service endpoint for receiving voyagePlans (findServices) and sends the requested voyagePlan to the consumer.

In the case the operator chooses not to authorize the consumer a textMessage is sent to the consumer with notification of an unsuccessful authorization.



7.2 Logging

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

7.2.1 VIS Event Log

Message exchange pattern:

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

The following events are proposed to be logged:

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

Incoming service calls on SeaSWIM side	
Event	**Log description**
getVoyagePlans	Log event for incoming request Log event with returned data
subscribeToVoyagePlan	Log event for incoming request Log event with returned data
uploadVoyagePlan	Log event with incoming data
uploadTextMessage	Log event with incoming data
uploadArea	Log event with incoming data
acknowledgement	Log event with incoming data

Outgoing service calls on SeaSWIM side	
Event	**Log description**
<callService>	Log event with outgoing data

8 References

Nr.	Version	Reference
[1] Service Documentation Guidelines	01.00	SG_Annex_A_Service_Documentation_Guidelines
[2] Route Exchange format (IEC 61174 App S)		http://stmvalidation.eu/schemas/
[3] VIS Specification Documentation	2.2	http://stmvalidation.eu/service-catalogue/

9 Acronyms and Terminology

9.1 Acronyms

Term	Definition
API	Application Programming Interface
MC	Maritime Cloud
MEP	Message Exchange Pattern
NAF	NATO Architectural Framework
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
SSD	Service Specification Document
UML	Unified Modelling Language
URL	Uniform Resource Locator
VTS	Vessel Traffic Service
WSDL	Web Service Definition Language
XML	Extendible Mark-up Language
XSD	XML Schema Definition
SSC	SeaSWIM Connector
URN	Uniform Resource Locator
UVID	Unique Voyage Identity
VIS	Voyage Information Service
VP	Voyage Plan

9.2 Terminology

Term	Definition
External Data Model	Describes the semantics of the “maritime world” (or a significant part thereof) by defining data structures and their relations. This could be at logical level (e.g., in UML) or at physical level (e.g., in XSD schema definitions), as for example standard data models, or S-100 based data produce specifications.
Message Exchange Pattern	Describes the principles two different parts of a message passing system (in our case: the service provider and the service consumer) interact and communicate with each other. Examples: In the Request/Response MEP, the service consumer sends a request to the service provider in order to obtain certain information; the service provider provides the requested information in a dedicated response. In the Publish/Subscribe MEP, the service consumer establishes a subscription with the service provider in order to obtain certain information; the service provider publishes information (either in regular intervals or upon change) to all subscribed service consumers.
Operational Activity	An activity performed by an operational node. Examples of operational activities in the maritime context are: Route Planning, Route Optimization, Logistics, Safety, Weather Forecast Provision, …

Operational Model	A structure of operational nodes and associated operational activities and their inter-relations in a process model.
Operational Node	A logical entity that performs activities. Note: nodes are specified independently of any physical realisation. Examples of operational nodes in the maritime context are: Maritime Control Center, Maritime Authority, Ship, Port, Weather Information Provider, …
Service	The provision of something (a non-physical object), by one, for the use of one or more others, regulated by formal definitions and mutual agreements. Services involve interactions between providers and consumers, which may be performed in a digital form (data exchanges) or through voice communication or written processes and procedures.
Service Consumer	A service consumer uses service instances provided by service providers. All users within the maritime domain can be service customers, e.g., ships and their crew, authorities, VTS stations, organizations (e.g., meteorological), commercial service providers, etc.
Service Data Model	Formal description of one dedicated service at logical level. The service data model is part of the service specification. Is typically defined in UML and/or XSD. If an external data model exists (e.g., a standard data model), then the service data model shall refer to it: each data item of the service data model shall be mapped to a data item defined in the external data model.
Service Design Description	Documents the details of a service technical design (most likely documented by the service implementer). The service design description includes (but is not limited to) a service physical data model and describes the used technology, transport mechanism, quality of service, etc.
Service Implementation	The provider side implementation of a dedicated service technical design (i.e., implementation of a dedicated service in a dedicated technology).
Service Implementer	Implementers of services from the service provider side and/or the service consumer side. Anybody can be a service implementer but mainly this will be commercial companies implementing solutions for shore and ship.
Service Instance	One service implementation may be deployed at several places by same or different service providers; each such deployment represents a different service instance, being accessible via different URLs.
Service Instance Description	Documents the details of a service implementation (most likely documented by the service implementer) and deployment (most likely documented by the service provider). The service instance description includes (but is not limited to) service technical design reference, service provider reference, service access information, service coverage information, etc.
Service Interface	The communication mechanism of the service, i.e., interaction mechanism between service provider and service consumer. A service interface is characterised by a message exchange pattern and consists of service operations that are either allocated to the provider or the consumer of the service.

Service Operation	Functions or procedure which enables programmatic communication with a service via a service interface.
Service Physical Data Model	Describes the realisation of a dedicated service data model in a dedicated technology. This includes a detailed description of the data payload to be exchanged using the chosen technology. The actual format of the service physical data model depends on the chosen technology. Examples may be WSDL and XSD files (e.g., for SOAP services) or swagger (Open API) specifications (e.g., for REST services). If an external data model exists (e.g., a standard data model), then the service physical data model shall refer to it: each data item of the service physical data model shall be mapped to a data item defined in the external data model. In order to prove correct implementation of the service specification, there shall exist a mapping between the service physical data model and the service data model. This means, each data item used in the service physical data model shall be mapped to a corresponding data item of the service data model. (In case of existing mappings to a common external (standard) data model from both the service data model and the service physical data model, such a mapping is implicitly given.)
Service Provider	A service provider provides instances of services according to a service specification and service instance description. All users within the maritime domain can be service providers, e.g., authorities, VTS stations, organizations (e.g., meteorological), commercial service providers, etc.
Service Specification	Describes one dedicated service at logical level. The Service Specification is technology-agnostic. The Service Specification includes (but is not limited to) a description of the Service Interfaces and Service Operations with their data payload. The data payload description may be formally defined by a Service Data Model.
Service Specification Producer	Producers of service specifications in accordance with the service documentation guidelines.
Service Technical Design	The technical design of a dedicated service in a dedicated technology. One service specification may result in several technical service designs, realising the service with different or same technologies.
Service Technology Catalogue	List and specifications of allowed technologies for service implementations. Currently, SOAP and REST are envisaged to be allowed service technologies. The service technology catalogue shall describe in detail the allowed service profiles, e.g., by listing communication standards, security standards, stacks, bindings, etc.
Spatial Exclusiveness	A service specification is characterised as “spatially exclusive”, if in any geographical region just one service instance of that specification is allowed to be registered per technology. The decision, which service instance (out of a number of available spatially exclusive services) shall be registered for a certain geographical region, is a governance issue.

Appendix A Service Design Description XML

This appendix contains the formal definition of the service design description.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceDesignSchema:serviceDesign xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ServiceDesignSchema="http://efficiensea2.org/maritime-cloud/service-registry/v1/ServiceDesignSchema.xsd"
xmlns:ServiceSpecificationSchema="http://efficiensea2.org/maritime-cloud/service-registry/v1/ServiceSpecificationSchema.xsd"
xsi:schemaLocation="http://efficiensea2.org/maritime-cloud/service-registry/v1/ServiceDesignSchema.xsd ServiceDesignSchema.xml">

<ServiceDesignSchema:id>urn:mrn:stm:service:design:sma:vis-rest-2.2</ServiceDesignSchema:id>
<ServiceDesignSchema:version>2.2</ServiceDesignSchema:version>
<ServiceDesignSchema:name>Voyage Information Service Design</ServiceDesignSchema:name>
<ServiceDesignSchema:status>released</ServiceDesignSchema:status>
<ServiceDesignSchema:description>Exchange Voyage information constituted of voyage plans (RTZv1.1STM), text message (STM Text
Message v1.3) and areas (S-124 v0.0.7)</ServiceDesignSchema:description>

<ServiceDesignSchema:offersTransport>
<ServiceDesignSchema:offersTransport>
        <ServiceDesignSchema:name>REST</ServiceDesignSchema:name>
        <ServiceDesignSchema:description>This service is designed as REST over HTTPS</ServiceDesignSchema:description>
        <ServiceDesignSchema:protocol>HTTPS</ServiceDesignSchema:protocol>
</ServiceDesignSchema:offersTransport>
</ServiceDesignSchema:offersTransport>
<ServiceDesignSchema:designsServiceSpecifications>
        <ServiceDesignSchema:designsServiceSpecifications>
                <ServiceDesignSchema:id>urn:mrn:stm:service:specification:sma:vis</ServiceDesignSchema:id>
                <ServiceDesignSchema:version>2.2</ServiceDesignSchema:version>
        </ServiceDesignSchema:designsServiceSpecifications>
</ServiceDesignSchema:designsServiceSpecifications>
<ServiceDesignSchema:designedBy>
        <ServiceSpecificationSchema:id>urn:mrn:stm:org:sma:pelo</ServiceSpecificationSchema:id>
        <ServiceSpecificationSchema:name>Per Löfbom</ServiceSpecificationSchema:name>
        <ServiceSpecificationSchema:description></ServiceSpecificationSchema:description>
        <ServiceSpecificationSchema:contactInfo>per.lofbom@sjofartsverket.se</ServiceSpecificationSchema:contactInfo>
        <ServiceSpecificationSchema:organizationId>urn:mrn:stm:org:sma</ServiceSpecificationSchema:organizationId>
        <ServiceSpecificationSchema:isCommercial>false</ServiceSpecificationSchema:isCommercial>
</ServiceDesignSchema:designedBy>
<ServiceDesignSchema:servicePhysicalDataModel>
        <ServiceDesignSchema:name>Voyage Information Service SMA Swagger JSON API</ServiceDesignSchema:name>
        <ServiceDesignSchema:description>API of VIS in JSON format</ServiceDesignSchema:description>
        <ServiceDesignSchema:modelType>JSON</ServiceDesignSchema:modelType>
        <ServiceDesignSchema:model>
        {
        "swagger": "2.0",
        "info": {
                "version": "2.2.0",
                "title": "STM Voyage Information Service SeaSWIM API",
                "description": "2.2.0  ed2  Updated description of payload version valid for RTZ, S124 and TXT"
        },
        "host": "localhost",
        "schemes": ["http",
        "https"],
        "paths": {
                "/acknowledgement": {
                        "post": {
                                "tags": ["Acknowledgement"],
                                "summary": "",
                                "description": "Endpoint for receipt of acknowledgement of uploaded message",
                                "operationId": "Acknowledgement",
                                "consumes": ["application/json"],
                                "produces": ["application/json"],
                                "parameters": [{
                                        "name": "deliveryAck",
```

```
                                        "in": "body",
                                        "description": "Acknowledgement",
                                        "required": true,
                                        "schema": {
                                                "$ref": "#/definitions/DeliveryAck"
                                        }
                                }],
                                "responses": {
                                        "200": {
                                                "description": "OK"
                                        },
                                        "400": {
                                                "description": "Bad Request"
                                        },
                                        "401": {
                                                "description": "Unauthorized (the user cannot be authenticated in the Identity
Registry)"
                                        },
                                        "403": {
                                                "description": "Forbidden"
                                        },
                                        "405": {
                                                "description": "Method not allowed"
                                        },
                                        "500": {
                                                "description": "Internal Server Error"
                                        },
                                        "default": {
                                                "description": "unexpected error"
                                        }
                                }
                        }
                },
                "/area": {
                        "post": {
                                "tags": ["Area"],
                                "summary": "",
                                "description": "Upload area message to VIS from other services i.e. Route Check service as an
informational message",
                                "operationId": "UploadArea",
                                "consumes": ["text/xml"],
                                "produces": ["application/json"],
                                "parameters": [{
                                        "name": "area",
                                        "in": "body",
                                        "description": "Area message in S124 v0.0.7",
                                        "required": true,
                                        "schema": {
                                                "type": "string"
                                        }
                                },
                                {
                                        "name": "deliveryAckEndPoint",
                                        "in": "query",
                                        "description": "Acknowledgement expected. Base URL for VIS as in Service Registry. An
ACK is expected to this URL when the receiving private application retrieve the message",
                                        "required": false,
                                        "type": "string"
                                }],
                                "responses": {
                                        "200": {
                                                "description": "OK"
                                        },
                                        "400": {
```

```
                                        "description": "Bad Request"
                                },
                                "401": {
                                        "description": "Unauthorized (the user cannot be authenticated in the Identity
Registry)"
                                },
                                "403": {
                                        "description": "Forbidden"
                                },
                                "405": {
                                        "description": "Method not allowed"
                                },
                                "500": {
                                        "description": "Internal Server Error"
                                },
                                "default": {
                                        "description": "unexpected error"
                                }
                        }
                }
        },
        "/textMessage": {
                "post": {
                        "tags": ["TextMessage"],
                        "summary": "",
                        "description": "Upload text message to VIS from other services i.e. Route Optimization service.",
                        "operationId": "UploadTextMessage",
                        "consumes": ["text/xml"],
                        "produces": ["application/json"],
                        "parameters": [{
                                "name": "textMessageObject",
                                "in": "body",
                                "description": "STM Text message v1.3",
                                "required": true,
                                "schema": {
                                        "type": "string"
                                }
                        },
                        {
                                "name": "deliveryAckEndPoint",
                                "in": "query",
                                "description": "Acknowledgement expected. Base URL for VIS as in Service Registry. An
ACK is expected to this URL when the receiving private application retrieve the message",
                                "required": false,
                                "type": "string"
                        }],
                        "responses": {
                                "200": {
                                        "description": "OK"
                                },
                                "400": {
                                        "description": "Bad Request"
                                },
                                "401": {
                                        "description": "Unauthorized (the user cannot be auhtenticated in the Identity
Registry)"
                                },
                                "403": {
                                        "description": "Forbidden"
                                },
                                "405": {
                                        "description": "Method not allowed"
                                },
                                "500": {
```

```
                                        "description": "Internal Server Error"
                                },
                                "default": {
                                        "description": "unexpected error"
                                }
                        }
                }
        },
        "/voyagePlans": {
                "get": {
                        "tags": ["VoyagePlan"],
                        "summary": "",
                        "description": "Returns active VoyagePlans",
                        "operationId": "GetVoyagePlans",
                        "consumes": [],
                        "produces": ["application/json"],
                        "parameters": [{
                                "name": "uvid",
                                "in": "query",
                                "description": "Unique identity (UVID) of a voyage plan",
                                "required": false,
                                "type": "string"
                        },
                        {
                                "name": "routeStatus",
                                "in": "query",
                                "description": "Status of a route for a voyageplan: 1-Original  2-Planned_for_voyage  3-
Optimized 4-Cross_Checked 5-Safety_Checked  6-Approved  7-Used_for_monitoring  8-Inactive",
                                "required": false,
                                "type": "string"
                        }],
                        "responses": {
                                "200": {
                                        "description": "OK",
                                        "schema": {
                                                "$ref": "#/definitions/GetVoyagePlanResponse"
                                        }
                                },
                                "400": {
                                        "description": "Bad Request"
                                },
                                "401": {
                                        "description": "Unauthorized (the user cannot be authenticated in the Identity
Registry)"
                                },
                                "403": {
                                        "description": "Forbidden (Not authorized request forwarded to operator)"
                                },
                                "404": {
                                        "description": "Not Found (the requested voyagePlan is not found)"
                                },
                                "405": {
                                        "description": "Method not allowed"
                                },
                                "500": {
                                        "description": "Internal Server Error"
                                },
                                "default": {
                                        "description": "unexpected error"
                                }
                        }
                },
                "post": {
                        "tags": ["VoyagePlan"],
```

```
                                        "summary": "",
                                        "description": "Upload VoyagePlan to VIS from other services i.e. Route Optimization service.",
                                        "operationId": "UploadVoyagePlan",
                                        "consumes": ["text/xml"],
                                        "produces": ["application/json"],
                                        "parameters": [{
                                                "name": "voyagePlan",
                                                "in": "body",
                                                "description": "Voyage Plan in RTZ v1.1STM. vesselVoyage and routeStatusEnum is
required",
                                                "required": true,
                                                "schema": {
                                                        "type": "string"
                                                }
                                        },
                                        {
                                                "name": "deliveryAckEndPoint",
                                                "in": "query",
                                                "description": "Acknowledgement expected. Base URL for VIS as in Service Registry. An
ACK is expected to this URL when the receiving private application retrieve the message",
                                                "required": false,
                                                "type": "string"
                                        },
                                        {
                                                "name": "callbackEndpoint",
                                                "in": "query",
                                                "description": "Callback expected. Base URL of the VIS instance as in the Service
Registry. The callback response will be sent to the voyagePlans endPoint of the instance",
                                                "required": false,
                                                "type": "string"
                                        }],
                                        "responses": {
                                                "200": {
                                                        "description": "OK"
                                                },
                                                "400": {
                                                        "description": "Bad Request"
                                                },
                                                "401": {
                                                        "description": "Unauthorized (the user cannot be auhtenticated in the Identity
Registry)"
                                                },
                                                "403": {
                                                        "description": "Forbidden"
                                                },
                                                "405": {
                                                        "description": "Method not allowed"
                                                },
                                                "500": {
                                                        "description": "Internal Server Error"
                                                },
                                                "default": {
                                                        "description": "unexpected error"
                                                }
                                        }
                                }
                        },
                        "/voyagePlans/subscription": {
                                "post": {
                                        "tags": ["VoyagePlan"],
                                        "summary": "",
                                        "description": "Request subscription for active Voyage Plan from other services i.e. Enhanced
Monitoring",
                                        "operationId": "SubscribeToVoyagePlan",
```

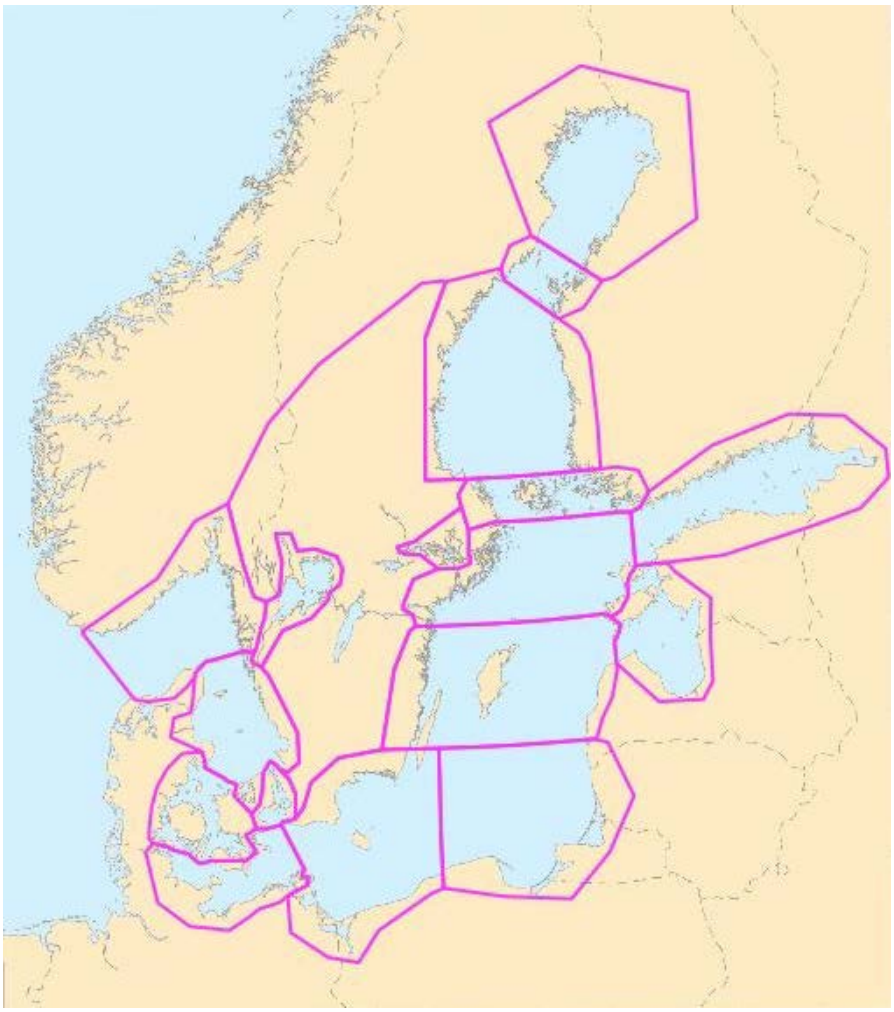
```
                "consumes": ["application/json"],
                "produces": ["application/json"],
                "parameters": [{
                        "name": "callbackEndpoint",
                        "in": "query",
                        "description": "Callback expected. Base URL of the vis instance as in the Service Registry. The callback response will be sent to the voyagePlans endPoint of the instance",
                        "required": true,
                        "type": "string"
                },
                {
                        "name": "uvid",
                        "in": "query",
                        "description": "Unique identity (UVID) of a voyageplan. If no uvid is provided, the subscription is to all the active uvid that your organization has access to",
                        "required": false,
                        "type": "string"
                }],
                "responses": {
                        "200": {
                                "description": "OK"
                        },
                        "400": {
                                "description": "Bad Request"
                        },
                        "401": {
                                "description": "Unauthorized (the user cannot be auhtenticated in the Identity Registry)"
                        },
                        "403": {
                                "description": "Forbidden (Not authorized request forwarded to operator)"
                        },
                        "404": {
                                "description": "Not Found (the requested Voyage Plan is not found)"
                        },
                        "405": {
                                "description": "Method not allowed"
                        },
                        "500": {
                                "description": "Internal Server Error"
                        },
                        "default": {
                                "description": "unexpected error"
                        }
                }
        },
        "get": {
                 "tags": ["VoyagePlan"],
                "summary": "",
                "description": "Retrieve a list of subcribed UVID for the callBackEndPoint and Organization",
                "operationId": "GetSubscriptionToVoyagePlans",
                "consumes": [],
                "produces": ["application/json"],
                "parameters": [{
                        "name": "callbackEndpoint",
                        "in": "query",
                        "description": "Callback expected. Base URL of the vis instance as in the Service Registry. The callback response will be sent to the voyagePlans endPoint of the instance",
                        "required": true,
                        "type": "string"
                }],
                "responses": {
                        "200": {
                                "description": "OK",
```

```
                                        "schema": {
                                                "type": "array",
                                                "items": {
                                                        "$ref": "#/definitions/GetSubscriptionResponse"
                                                }
                                        }
                                },
                                "400": {
                                        "description": "Bad Request"
                                },
                                "401": {
                                        "description": "Unauthorized (the user cannot be authenticated in the Identity
Registry)"
                                },
                                "403": {
                                        "description": "Forbidden (Not authorized request forwarded to operator)"
                                },
                                "404": {
                                        "description": "Not Found (the requested Voyage Plan is not found)"
                                },
                                "405": {
                                        "description": "Method not allowed"
                                },
                                "500": {
                                        "description": "Internal Server Error"
                                },
                                "default": {
                                        "description": "unexpected error"
                                }
                        }
                },
                "delete": {
                        "tags": ["VoyagePlan"],
                        "summary": "",
                        "description": "Remove subscription for active Voyage Plan from other services i.e. Enhanced
Monitoring",
                        "operationId": "RemoveVoyagePlanSubscription",
                        "consumes": [],
                        "produces": ["application/json"],
                        "parameters": [{
                                "name": "callbackEndpoint",
                                "in": "query",
                                "description": "Callback expected. Base url of the vis instance as in the Service Registry.
The callback response will be sent to the voyagePlans endPoint of the instance",
                                "required": true,
                                "type": "string"
                        },
                        {
                                "name": "uvid",
                                "in": "query",
                                "description": "Unique identity (UVID) of a voyage plan",
                                "required": false,
                                "type": "string"
                        }],
                        "responses": {
                                "200": {
                                        "description": "OK"
                                },
                                "400": {
                                        "description": "Bad Request"
                                },
                                "401": {
                                        "description": "Unauthorized (the user cannot be authenticated in the Identity
Registry)"
```

```
                                        },
                                        "403": {
                                                "description": "Forbidden"
                                        },
                                        "404": {
                                                "description": "Not Found (the requested Voyage Plan is not found)"
                                        },
                                        "405": {
                                                "description": "Method not allowed"
                                        },
                                        "500": {
                                                "description": "Internal Server Error"
                                        },
                                        "default": {
                                                "description": "unexpected error"
                                        }
                                }
                        }
                }
        },
        "definitions": {
                "DeliveryAck": {
                        "description": "Acknowledgement message that incoming (uploaded) message has been delivered to
consumer",
                        "type": "object",
                        "properties": {
                                "id": {
                                        "description": "Acknowledgement ID",
                                        "type": "string"
                                },
                                "referenceId": {
                                        "description": "Reference ID such as a UVID, TXT id or area message id",
                                        "type": "string"
                                },
                                "timeOfDelivery": {
                                        "format": "date-time",
                                        "description": "Time of Delivery of message to consumer",
                                        "type": "string"
                                },
                                "fromId": {
                                        "description": "Identity O (organisation) of the message sender in MRN format",
                                        "type": "string"
                                },
                                "fromName": {
                                        "description": "\"Identity O (organisation) of the message sender in full name",
                                        "type": "string"
                                },
                                "toId": {
                                        "description": "Identity O (organisation) of the message receiver in MRN format",
                                        "type": "string"
                                },
                                "toName": {
                                        "description": "IIdentity O (organisation) of the message receiver in full name",
                                        "type": "string"
                                },
                                "ackResult": {
                                        "description": "Descriptive acknowledgement message",
                                        "type": "string"
                                }
                        }
                },
                "GetVoyagePlanResponse": {
                        "description": "Response object from GET voyagePlans. Can contain 0 or several (0..*) voyage plans",
                        "type": "object",
```

```
                        "properties": {
                                "lastInteractionTime": {
                                        "format": "date-time",
                                        "description": "Last interaction time with private application. Can indicate the current connectivity on private side of VIS",
                                        "type": "string"
                                },
                                "voyagePlans": {
                                        "description": "Array of voyage plans in RTZ XML format",
                                        "type": "array",
                                        "items": {
                                                "$ref": "#/definitions/VoyagePlan"
                                        }
                                }
                        }
                },
                "VoyagePlan": {
                        "description": "A voyage plan in RTZ XML format",
                        "type": "object",
                        "properties": {
                                "route": {
                                        "description": "A voyage plan in RTZ v1.1STM. vesselVoyage and routeStatusEnum is required",
                                        "type": "string"
                                }
                        }
                },
                "GetSubscriptionResponse": {
                        "description": "DataId object containing the UVID in URN format",
                        "type": "object",
                        "properties": {
                                "DataId": {
                                        "description": "Unique identity (UVID) of a voyageplan",
                                        "type": "string"
                                }
                        }
                }
        }
}
</ServiceDesignSchema:model>
</ServiceDesignSchema:servicePhysicalDataModel>
</ServiceDesignSchema:serviceDesign>
```

Service Instance Description for the Baltic Navigational Warning Service



1 INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

The purpose of this service instance description document is to provide an operational description of the specific service instance.

The aim is to document the key aspects of the service instance. This includes:

- identification and summary of the service instance
  - reference to the design description
  - identification of the service instance
- service instance details
  - operational details
  - specific interaction pattern
- release notes
  - feature list
  - bug list.

1.2 INTENDED READERSHIP

This service instance description document is intended to be read by service consumers in charge of selecting the service instance to consume.

F 1

F 2 SERVICE INSTANCE IDENTIFICATION

Table 1 *Service Instance Identification*

Name	Baltic Navigational Warning Service
ID	urn:mrn:stm:service:instance:sma:bnw
Version	0.1
Technology	REST
Service Specification ID	urn:mrn:stm:service:specification:sma:vis
Service Specification Version	2.2
Service Design ID	urn:mrn:smt:service:design:sma:vis-rest-2.2
Service Design Version	2.2
Description	The service provides Navigational Warnings in the Baltic region and Swedish T&P notices
Keywords	NW, Navigational Warning, Baltic, S-124, T&P
Supplier	Swedish Maritime Administration, SMA urn:mrn:stm:org:sma
Status	Released.

F 3 SERVICE IMPLEMENTATION AND INSTANTIATION DETAILS

1.3 OVERALL DESCRIPTION

DISCLAIMER:
The Baltic Navigational Warning service is not intended to relieve the service users from ordinary receipt of Maritime Safety Information (MSI) as part of the Global Maritime Distress and Safety System (GMDSS), which every ship, while at sea, has to comply to. Since the service is intended to be used for test and validation purposes during the STM Validation Project the Swedish Maritime Administration, as service provider, cannot guarantee any service level or take any responsibility that all relevant warnings and information are provided by the service.

The purpose with the Baltic Navigational Warning service is to provide the service consumer, i.e. ship, with only those warnings that are relevant for that specific route that they intend to sail/are currently at and at the time specified in the route schedule. Moreover, the warnings will be displayed directly in ECDIS and automatically deleted when they are expired and no longer valid.

The benefits are:

- Reduced workload – No need to manually plot positions/areas received by NAVTEX/voice communication at ENC/paper chart. This allows the navigator to concentrate on safely navigating the ship
- Increased safety of navigation – According to London P&I Club Insurance inspections regularly find deficiencies in managing navigation warnings and notices to mariners as officers fail to implement navigational safety notices. By providing the notices directly to ships ECDIS manual work and risk of missing important information is reduced and T&P notices can be received digitally already before sent out as ENC updates. In addition all Temporary and Provisional (T&P) Notice to Mariners are not sent out today which means that full ECDIS ships, sailing paperless, do not get all notices.
- Reduced human errors – As warnings are provided digitally and seamlessly shown directly on ECDIS possible human errors possible errors in misunderstandings and manual plotting can be avoided.
- Increased Navigational Warning focus - Since only notices relevant for the planned and/ or actual route will be sent to the ECDIS, the Officer On Watch can concentrate on these and need not bother with warnings issued outside the adjacent areas.

The service provides safety notices to ships in S-124 format. The S-124, navigational warnings, product specification is being developed by an IHO Correspondence Group with the purpose to submit it for endorsement. Before being mature for endorsement the STM Validation Project will serve as one of the testbeds to validate a draft version of the specification.
The service is initiated when a ship shares its Voyage Plan (VP) with the Baltic Navigational Warning service. In response, the Baltic Navigational Warning service initially provides the ship with all related safety notices in the concerned area(s), and then continuously all updates in the concerned area(s). Notices that are within the sub-areas that the route crosses, see figure 1 in paragraph 3.2 for sub-area division, are deemed as relevant and returned to the ship. Notices in other sub-areas will not be returned. When ship has left the service coverage area, the Baltic Navigational Warning service stops sending updates to the ship. More operational details are to be found in paragraph 3.5, functional description.

The Baltic Navigational Warning service provides the following navigational safety notices:

- **Coastal warnings** - Navigational warnings that apply to open waters are classified as coastal. The same information that today is transmitted on NAVTEX.
- **Local warnings** for Swedish waters - Warnings that apply only to waters inside the belt of the skerries are regarded as local. Today transmitted only on VHF.
- **Temporary and Provisional notices** for Swedish waters

NOTE: weather/ice information is not provided by the service. T&P notices is not included in the first release.

1.4 SERVICE COVERAGE

The service covers the following area:

- The Baltic sea area

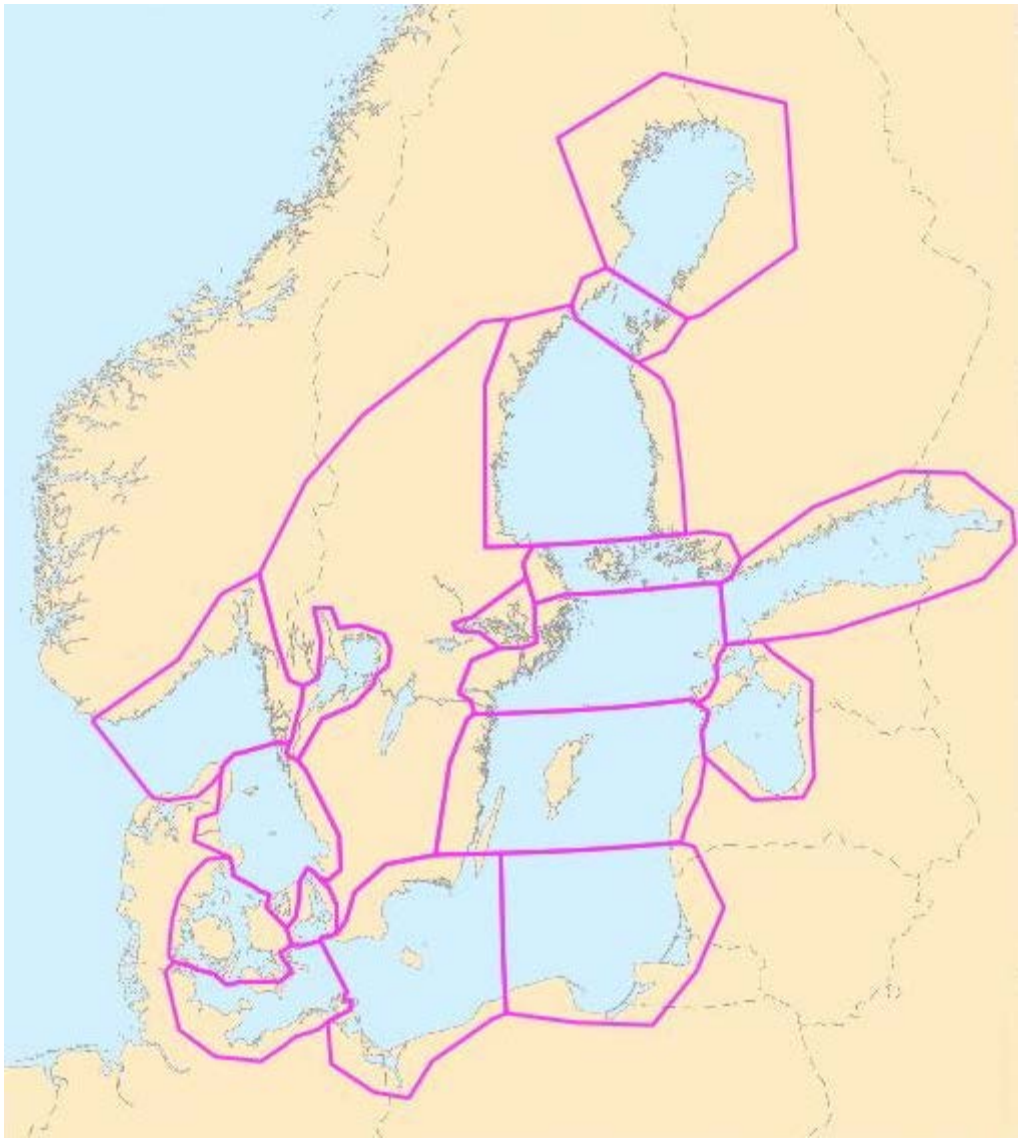


Figure 1 Service coverage area

1.5 REQUIRED INPUT

The incoming Voyage Plan must be an RTZ version 1.1 with STM Extension according to the VIS Design 2.2.
The Voyage Plan must include a schedule, which can be of type manual or calculated, with a date and time within the validity period of the issued Navigational warning.

1.6 OUTPUT FROM THE SERVICE

The output from the service is Navigational Warnings in S124 v0.0.7 format according to the VIS Design 2.2.

1.7 FUNCTIONALITY DESCRIPTION

When voyage plan is received by the service, the consumer/ship is added in a subscription list for Navigational Warnings in the Baltic Sea area. The ship will initially receive all active warnings concerning the sub-areas that the route crosses/enters into, see Figure 2, and then continuously receive updates, new and cancelled messages until the route leaves the area and subscription is removed by the service.

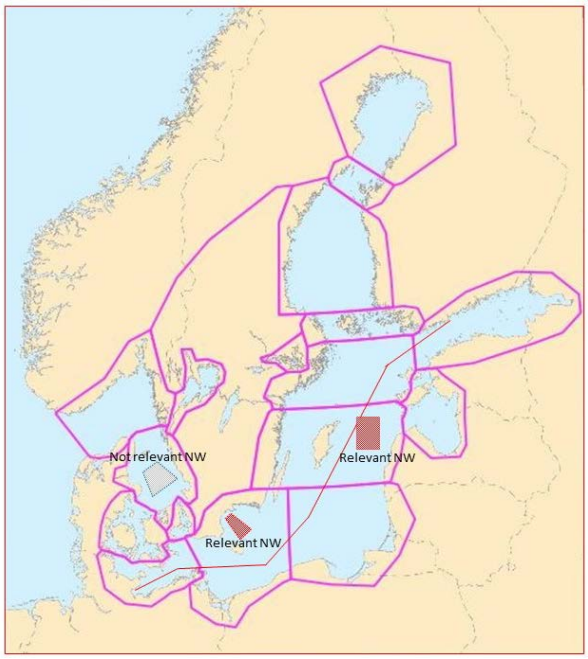


Figure 2 Example of relevant notices based on ships Voyage Plan and sub-area division

The notices are not limited to specific transmission times but are sent as soon as warnings are registered in Sweden Traffic.
If a Navigational Warning belongs to several areas or if it not assigned to a specific (in Sweden traffic the Swedish management systems for Navigational Warnings) area it will be sent out as if it concerns all sub-areas i.e. if a route passes any sub-area will receive the warning.
The service will remove the ship from subscription list when the ship has:

- Left the service coverage area, according to the Voyage Plan schedule and waypoint locations
- The Voyage Plan is inactivated onboard the ship
- The Voyage Plan is finalised (the ship has arrived to a destination/port in the service coverage area) according to arrival time at last waypoint
- If a new Voyage Plan is shared with the service without prior plan being inactivated the service will use the last plan received and filter so that no duplicate messages are sent to the ship

As an example a ship that departs from the Mediterranean Sea with a Voyage Plans that goes into the Baltic will receive all active warnings when they call the service and updates throughout the voyage. A ship with a voyage in the opposite direction will only receive updates until they leave, based on waypoint geography and arrival time, the service coverage area.

How received notices are handled in each STM compatible ship system are described in respective user manual but the common requirements are that the ECDIS/bridge system should be capable of:

- Display received areas
- Handle updated notice area
- Delete notices when expired/obsolete

1.8 DYNAMIC DESCRIPTION

Interaction with the service is initiated from the ship by sending a Voyage Plan. The service then responds by sending back relevant Navigational Warnings.

- The service is searchable through the Service Registry
- Ship will request safety notices by uploading (sending) their voyage plan to the Baltic Navigational Warning service
- The Baltic Navigational Warning service will return a list of safety notices by uploading area messages (S-124) to the ship

Interacting with the Baltic Navigational Warning service is done through the VIS public interface UploadVoyageplan. The warning messages are returned to the ships VIS public interface uploadArea. For further details about the interface, see the VIS documentation

1.9 ALLOWED OPERATIONS

The Baltic Navigational Warning Service is based on the Voyage Information Service design version 2.2, but only handle a set of the methods.

The service handles interaction on the following methods:

Operation	Method	Allowance/handling	Comment
Receive voyage plan in RTZ	uploadVoyagePlans	**Yes**	RTZ v1.1STM
Receive STM text messages	uploadTextMessage	No	
Receive area (S124) messages	uploadArea	No	
Return list of voyage plans on request	getVoyagePlans	No	
Accept subscription request	subscribeToVoyagePlans	No	

The Baltic Navigational Warning Service does not nominate actors internally, but will respond with new or updated navigational warnings when receiving uploaded voyage plan to the service.

F 4 RELEASE NOTES

The service is released in its first version.

F 5 DEFINITIONS

F 5.1 TERMINOLOGY

Table 2 Definition of terminology

Term	Definition
Service	The provision of something (a non-physical object), by one, for the use of one or more others, regulated by formal definitions and mutual agreements. Services involve interactions between providers and consumers, which may be performed in a digital form (data exchanges) or through voice communication or written processes and procedures.
Service Consumer	A service consumer uses service instances provided by service providers. All users within the maritime domain can be service customers, e.g., ships and their crew, authorities, VTS stations, organizations (e.g., meteorological), commercial service providers, etc.
Service Instance	One service implementation may be deployed at several places by same or different service providers; each such deployment represents a different service instance, being accessible via different URLs.
Service Instance Description	Documents the details of a service implementation (most likely documented by the service implementer) and deployment (most likely documented by the service provider). The service instance description includes (but is not limited to) service technical design reference, service provider reference, service access information, service coverage information, etc.
Service Interface	The communication mechanism of the service, i.e., interaction mechanism between service provider and service consumer. A service interface is characterised by a message exchange pattern and consists of service operations that are either allocated to the provider or the consumer of the service.
Service Operation	Functions or procedure which enables programmatic communication with a service via a service interface.
Service 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.

F 6 ACRONYMS

F 7 REFERENCES

[1] IALA Guideline 1128 on Specification of e-Navigation Technical Services

[2] VIS REST Design - for SeaSWIM ver 2.2.2, http://stmvalidation.eu/developers-forum/vis/