CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

MARITIME NAVIGATION AND RADIOCOMMUNICATION EQUIPMENT AND SYSTEMS – DATA INTERFACE –

Part 2: Secure exchange and communication of S-100 based products (SECOM)

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 63173-2 has been prepared by IEC technical committee 80: Maritime navigation and radiocommunication equipment and systems.

The text of this standard is based on the following documents:

FDIS	Report on voting
80/XX/FDIS	80/XX/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

The National Committees are requested to note that for this publication the stability date is 2018

THIS TEXT IS INCLUDED FOR THE INFORMATION OF THE NATIONAL COMMITTEES AND WILL BE DELETED AT THE PUBLICATION STAGE.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

E-navigation with specified maritime services have been defined by the International Maritime Organization (IMO) as the means of providing electronic information in a harmonized way. IMO´s e-navigation Strategy Implementation Plan (SIP) expects that IEC will implement the details as outlined in the SIP.

The maritime services are operational services for actors both ashore and onboard, which in turn rely on technical services that provides digitalized information. The data formats are typically standardized in Data Model Specifications and many organisations are working on establishing new standards e.g. within the International Hydrographic Organization (IHO) S-100 framework based on the Universal Hydrographic Data Model.

A technical service specification template is included in the International Association of Marine Aids to Navigation and Lighthouse Authorities (IALA) Service Documentation Guidelines (IALA G1128) and is also a part of IHO S-100 ed.4 part 14. This IEC document specifies a service interface which aims to utilize standardized data formats and to implement IALA's and IHO's templates and guidelines for information services. The standardisation of a common service interface for exchange of S-100 based products will enable wider technical interoperability where the same service interface is used for exchanging information regardless of operational use.

Accordingly, the purpose of the standard is to:

- enable standardized service exchange of information part of Maritime Services e.g. Route Plans, Chart updates and Navigational Warnings;
- facilitate interoperability between maritime systems;
- reduce the need to support many different (proprietary) service designs;
- utilize the benefits of Service Oriented Architecture, for example to enable ship systems to interact with port systems on the first call to that specific port.

This document specifies a set of service interfaces, i.e. APIs, to be used when exchanging information based on the IHO S-100 Universal Hydrographic Data Model. It also contains protection scheme for the information exchange in line with IHO S-100 and service discoverability.

SECOM is applicable for IP based session-less interactive web services for information exchange. Other possible means of exchange, for example general distribution of files is not included. The standard is applicable for both public (governmental) and private services. SECOM is not intended to define by which physical layer the transport is performed as long as IP communication can be supported e.g. satellite, VDES, 3-4-5G or other means of communication.

The scope of SECOM data protection is between end users. The scope of the SECOM service interface is the public side exposed on the internet and how the service is discoverable by the consumer. The "last mile" links between SECOM service instance and the end-user application is described in informative annex only. This allows different solutions between a ship's onboard system/application and the service.

MARITIME NAVIGATION AND RADIOCOMMUNICATION EQUIPMENT AND SYSTEMS – DATA INTERFACE –

Part 2: Secure exchange and communication of S-100 based products (SECOM)

1 Scope

The scope of SECOM data protection is between end users. The scope of the SECOM service interface is the public side exposed on the internet and how the service is discoverable by the consumer. The “last mile” links between SECOM service instance and the end-user application is described in informative annex only. This allows different solutions between a ship’s onboard system/application and the service.

This document specifies a set of service interfaces, i.e. APIs, to be used when exchanging information based on the IHO S-100 Universal Hydrographic Data Model. It also contains protection scheme for the information exchange in line with IHO S-100 and service discoverability.

SECOM is applicable for IP based session-less interactive web services for information exchange. Other possible means of exchange, for example general distribution of files is not included. The standard is applicable for both public (governmental) and private services. SECOM is not intended to define by which physical layer the transport is performed as long as IP communication can be supported e.g. satellite, VDES, 3-4-5G or other means of communication.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

S-100,	*IHO Universal Hydrographic Data Model*
RFC-2246,	*TLS version 1.0 (1999)*
RFC-4346,	*TLS version 1.1 (2006)*
RFC-5246,	*TLS version 1.2 (2008)*
RFC-8446,	*TLS version 1.3 (2018)*
RFC-2818,	*HTTP Over TLS (2000)*
RFC-5280,	*Profile of X.509*
RFC-2459,	*Internet X.509 Public-key infrastructure and attribute certificate frameworks*

RFC-7231, *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*

3 Terms, definitions and abbreviated terms

For the purposes of this document the following terms, definitions and abbreviated terms apply.

ISO and IEC maintain termonilogical databases for use in standardization at the following addresses:

- IEC Electopedia: available at http://www.electropedia.org
- ISO Online browsing platform: available at http://www.iso.org/obp

3.1 Terms and definitions

3.1.1
Actor
For the purpose of this standard, actor is defined as “specifies a role played by a user or any other system that interacts with the subject.”.

[SOURCE: Unified Modelling Language (UML)]

3.1.2
Feature
Abstraction of real world phenomena. A feature may occur as a type or an instance. A feature type or a feature instance should be used when only one is meant.

[SOURCE: IHO S-100]

3.1.3
Feature catalogue
A catalogue containing definitions and descriptions of the feature type, feature attributes, and feature associations occurring in one or more sets of geographic data

[SOURCE: IHO S-100]

3.1.4
Service
OASIS defines a service as "a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description".

[SOURCE: OASIS Reference Model for Service Oriented Architecture 1.0]

3.1.5
Public Key Infrastructure
A public key infrastructure (PKI) is a set of roles, policies, hardware, software and procedures needed to create, manage, distribute, use, store and revoke digital certificates and manage public-key encryption.

[SOURCE: Wikipedia]

3.1.6
REpresentational State Transfer
Representational state transfer (REST) is a software architectural style that defines a set of constraints to be used for creating Web services. Web services that conform to the REST architectural style, called RESTful Web services (RWS), provide interoperability between computer systems on the Internet.

[SOURCE: Wikipedia]

3.2 Abbreviated terms

API	Application Program Interface
ENC	Electronic Navigational Chart
IALA	International Association of Marine Aids to Navigation and Lighthouse Authorities
IHO	International Hydrographic Organization
IMO	International Maritime Organization
ISO	International Organization for Standardization
REST	REpresentational State Transfer
PKI	Public Key Infrastructure

4 General Description

The maritime services are operational services for actors both ashore and onboard, which in turn rely on technical services that provides digitalized information, see Figure 2 derived from IALA ENAV23-12.1.4 eNav-Description. The data formats are typically standardized in Data Model Specifications and many organisations are working on establishing new standards e.g. within IHO’s S-100 framework based on the Universal Hydrographic Data Model.

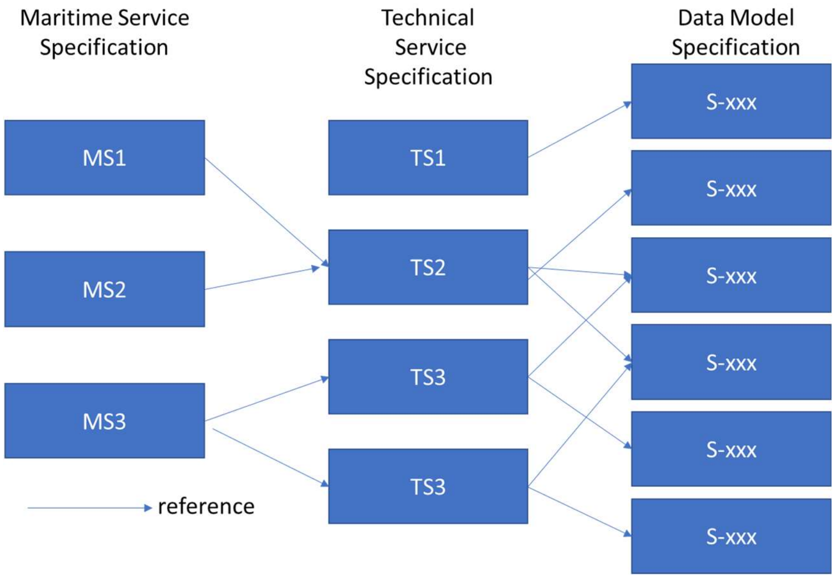

Figure 2 - The relationship between specifications of Maritime Services, Technical Services and data models in e-Navigation

This document specifies a set of service interfaces, i.e. APIs, to be used when exchanging information based on the IHO S-100 Universal Hydrographic Data Model. It also contains protection scheme of the communication channel, authentication procedures for web services and service discoverability. Data protection (encryption) refers to IHO data protection Scheme (IHO S-100 ed.4 part 15). gives the scope and relation to existing IHO standards. The communication is bi-directional which means that both the ship/shipping company as well as shore authorities and private service providers can initiate communication and act as information provider.

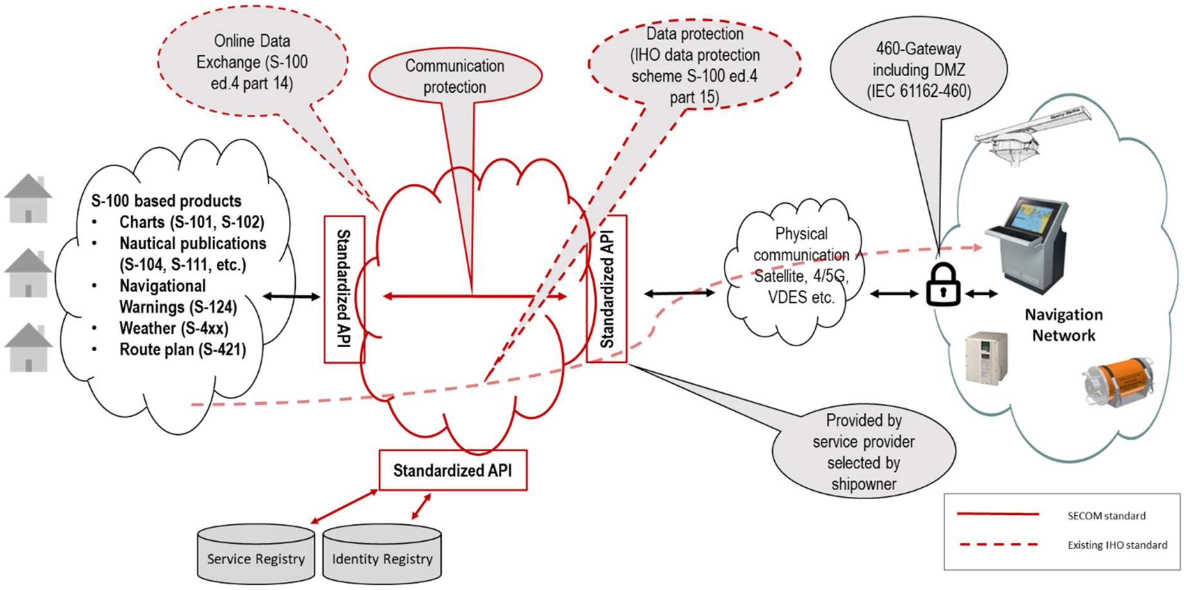
Online Data Exchange (S-100 ed.4 part 14)
Communication protection
Data protection (IHO data protection scheme S-100 ed.4 part 15)
460-Gateway including DMZ (IEC 61162-460)
S-100 based products
• Charts (S-101, S-102)
• Nautical publications (S-104, S-111, etc.)
• Navigational Warnings (S-124)
• Weather (S-4xx)
• Route plan (S-421)
Standardized API
Standardized API
Physical communication Satellite, 4/5G, VDES etc.
Navigation Network
Provided by service provider selected by shipowner
Standardized API
Service Registry
Identity Registry
SECOM standard
Existing IHO standard

Figure 3 - Scope of this document.

SECOM is applicable for IP based session-less interactive web services for information exchange. Other possible means of exchange, for example general distribution of files is not included. The standard is applicable for both public (governmental) and private services. SECOM is not intended to define by which physical layer the transport is performed as long as IP communication can be supported e.g. satellite, VDES, 3-4-5G or other means of communication.

The scope of SECOM data protection is between end users. The scope of the SECOM service interface is the public side exposed on the internet and how the service is discoverable by the consumer. The “last mile” links between SECOM service instance and the end-user application is described in informative annex only. This allows different solutions between a ship’s onboard system/application and the service.

The standardisation of service interface for exchange of S-100 based information enables wider technical interoperability where the same service interface is used for exchanging the information regardless of operational use. This enables a user that implemented the standard service interface for e.g. exchange of S-421 Route Plan to consume a set of operational services with different purpose, but all based on the exchange (push) of S-421 Route Plan. The standard service interface also contains interfaces for subscription, access request, pulling and dynamic request for the instance capability, status and description.

Implementing a standardized interface for a certain product, creates significant advantages for a ship as opposed to being forced to implement a vast number of different interfaces in order to adhere to proprietary interfaces from various service providers and consumers. The focus on service interface (information service) for the exchange of information products will facilitate ships and ECDIS suppliers having equipment approved for exchange of certain information products. However the usage of these services from an operational perspective are dynamic and up to the user and will most likely generate less requirements for changes onboard when new services are to be introduced.

Service providers will have similar benefits in producing services based on exchange of standard information products, hereby gaining technical interoperability with all ships having implemented the standard service interface.

The standard of communication protection scheme enables technical interoperability for secure online transfer, and service discoverability enables search for the service to consume.

This standard complements S-100 to gain technical interoperability up to the level of exchanging information service-to-service securely online.

The following clauses are independent on each other and describe the modules:

Clause 5	Service Interface
Clause 6	Communication Protection
Clause 7	Data Protection
Clause 8	Service Discoverability
Clause 9	Key management interface on PKI System

5 Service Interface

5.1 General

This chapter describes the standard service interface for exchanging S-products, such as S-421.

The core service interface is the Upload service interface. This service interface is used by producer of information to push S-product (e.g. S-421) to a consumer. This will cover the primary need for exchange of information. Additional service interfaces are specified to support and complement the Upload service interface, such as subscriptions, notifications, acknowledgements etc..

5.2 Service Definition Model

The service interface is described according to S-100 v4 chapter 14 “Online Data Exchange”, complemented with guidelines from IALA Service Documentation Guidelines.

5.3 Service Technology

The technology chosen is REST (REpresentational State Transfer) upon HTTP/1.1 (RFC-7231). The definition of each operation contain the error codes that the operation specifically shall respond with as complement to the HTTP response codes defined by HTTP/1.1 (RFC-7231).

5.4 Service Interface Defintion

This chapter describe standard service interface. The interface described here is not for any specific S-product. The placeholder <s-nnn> is used instead.

See Annex Service REST Technical Design for traceability to S100 v4service definition model.

Table 1 gives an overview of the standardised interfaces

Table 1 - Service interface overview

	Service Interface	Exchange Pattern	Definition
PUSH	Upload	REQUEST_CALLBACK	Interface for uploading (pushing) information to consumer ConsumerInterface: Acknowledgement
	Acknowledgement	ONE_WAY	Interface for acknowledgement on uploaded information.
PULL	Get	REQUEST_RESPONSE	Interface to ask for (pulling) information from producer
	Get List	REQUEST_RESPONSE	Interface to ask for (pulling) a list of information from producer
ACCESS	Access	REQUEST_CALLBACK	Interface to ask for access to information ConsumerInterface: AccessNotification
	Access Notification	ONE_WAY	Interface for notification from access request
SUBS	Subscription	PUBLISH_SUBSCRIBE	Interface to create subscription of information

			ConsumerInterface: SubscriptionNotification Upload
	Subscription Notification	ONE_WAY	Interface for notification from subscription events
	Remove Subscription	ONE_WAY	Interface to remove subscription
	Get Subscription List	REQUEST_RESPONSE	Interface to retrieve a list of active subscriptions
HELP	Capability	REQUEST_RESPONSE	Interface to ask for the interface capabilities
	Description	REQUEST_RESPONSE	Interface to ask for a short description of the service

5.4.1 Service interface – Upload

5.4.1.1 Specification

The purpose with this interface is to upload (push) one information object to a consumer. Hence, a consumer of the message format need to implement this interface in order to receive an information object.

This operation is used both in single uploads and uploads during subscription. The parameter <FromSubscription> indicates true/false whether within or outside any subscription by the consumer.

When uploading the message, an acknowledgement can be requested which is expected to be received when the uploaded message has been delivered to end system (technical acknowledgement), and, if supported, an acknowledgement when the message has been opened (read) by the end user (operational acknowledgement).

The <purpose> parameter is used to indicate a specific purpose with the upload of information.

Table 2 – Parameter binding

Information (in)	Mult.	Definition
Information	1	The payload based on S-Product Specification
Metadata		
FromSubscription	0..1	Flag to indicate whether the payload has been uploaded within an active subscription or not.
AckRequest	0..1	Flag to indicate that acknowledgement is expected when delivered, and an acknowledgement when message has been opened (read) by end user.
Purpose	0..1	Purpose with the upload. The expected action by the consumer.
Information (out)	**Mult.**	
result from operation	1	Reference for acknowledgement

5.4.1.2 REST Design

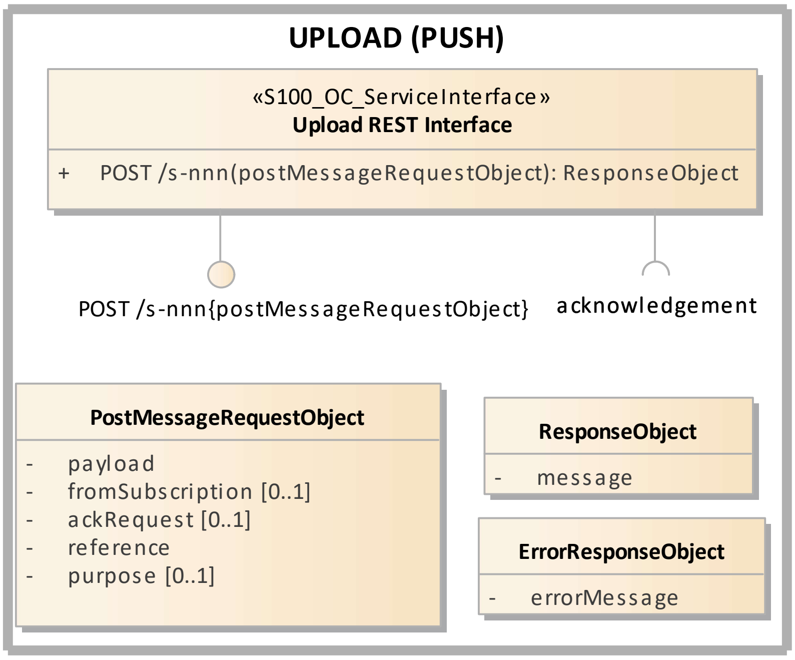

Figure 4 - Upload interface

ExchangePattern

- ONE_WAY if no ACK requested
- REQUEST_CALLBACK when ACK is requested

ConsumerInterface

- Consumes Acknowledgement service interface, if acknowledgement is requested.

5.4.1.3 Operation - POST <s-nnn>

The operation shall be used for uploading (push) information object to a consumer. The operation expect one single information object in specified format as payload and its metadata.

Table 3 - Parameters in REST operation Upload

REST Parameter (in)	Type	Encoding	Mult.	Definition
-				
REST Body (in)	**Type**	**Encoding**	**Mult.**	**Definition**
PostMessageRequestObject	String	application/json	1	The data (payload) package with its metadata
Return (out)	**Type**	**Encoding**	**Mult.**	**Definition**
ResponseObject ErrorResponseObject	String	application/json	1	Confirmation of upload or error message, including description of incorrect or missing values in payload for the specific service.

5.4.1.4 Data Exchange Model

Table 4 - Data exchange model for the REST operation

PostMessageRequestObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
payload	String	<information product specification>	1	The data according to Product Specification embedded in JSON. The data may be in XML or encrypted binary string.
reference	String	UUID	1	Reference identifier to be used in acknowledgement
fromSubscription	boolean		0..1	Flag to indicate whether the payload has been uploaded within an active subscription or not
ackRequest	boolean		0..1	Flag to indicate that acknowledgement is expected.
purpose	String	enumPurpose	0..1	Purpose with the upload. The expected action by the consumer.
enumPurpose				
Enum	**Type**	**Encoding**		**Definition**
<purpose>	string		1	The list if valid purposes is dependent on the type of payload exchanged, if not a generic list of purposes can be created
ResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
message	string		1	Success response message
ErrorResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
errorMessage	string		1	Error response message

5.4.1.5 Service response

The service shall respond with HTTP codes and message according to Table 5.

Table 5 – HTTP Response codes

HTTP Code	Message
200	Successfully received <reference>
403	Not authorized to upload
500	Attribute <x> is missing in <reference>
500	Attribute <x> is incorrect in <reference>

5.4.1.6 Dynamic behavior

<add sequence diagram for the service interface>

5.4.1.7 Test methods and expected result

Test of the standard

– Procedures used to test the applicability, consistence, completeness and correctness of the standard

Test work item against standard

- Procedure to test X against the standard, hence “Does my work item follow the standard?”
- Test Data

5.4.2 Service interface - Acknowledgement

5.4.2.1 Specification

During upload of information, an acknowledgement can be requested when the information has been delivered to end system by the consumer service. The acknowledgement contains reference to information product delivered.

Table 6 – Parameter binding

Information (in)	Mult.	Definition
Reference	1	Reference to acknowedged object
Time	1	Time when delivered or opened
AcknowledgementType	1	Type of acknowledgement (technical delivery ACK, or operational "message opened" ACK)
Information (out)	**Mult.**	**Definition**
result from operation	1	

5.4.2.2 REST Design

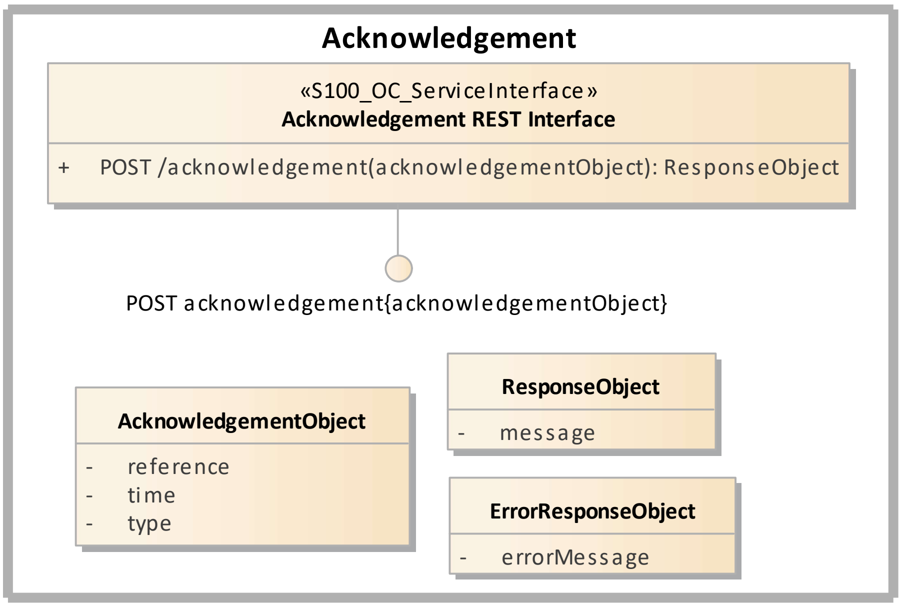
Acknowledgement
«S100_OC_ServiceInterface»
Acknowledgement REST Interface
+ POST /acknowledgement(acknowledgementObject): ResponseObject
POST acknowledgement{acknowledgementObject}
AcknowledgementObject
- reference
- time
- type
ResponseObject
- message
ErrorResponseObject
- errorMessage

Figure 5 - Acknowledgement REST interface

ExchangePattern

- ONE_WAY

ConsumerInterface

- No interface consumed.

5.4.2.3 Operation - POST acknowledgement

Table 7 - Parameters for the operation

REST Parameter (in)	Type	Encoding	Mult.	Definition
-				

REST Body (in)	Type	Encoding	Mult.	Definition
AcknowledgementObject	String	application/json	1	Object with reference to information and time when delivered
Return (out)	**Type**	**Encoding**	**Mult.**	**Definition**
ResponseObject ErrorResponseObject	String	application/json	1	Confirmation or error message

5.4.2.4 Data Exchange Model

Table 8 - Data exchange model for the REST operation

AcknowledgementObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
reference	String	UUID	1	Reference identifier given in upload
time	String	dateTime	1	Creation time for the acknowledgement
type	string	enumType	1	Type of acknownledgement
ResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
message	string		1	Success response message
ErrorResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
errorMessage	string		1	Error response message
enumType				
Enum	**Type**	**Encoding**		**Definition**
technicalACK	string			Technical acknowledgement such as delivered to end system.
operationalACK	string			Operational acknowledgement such as when opened/read by end user.

5.4.2.5 Service response

Table 9 – HTTP Response codes

HTTP Code	Message
200	Successfully received ACK for <reference>
403	Not authorized to upload ACK
404	Message <reference> not found

5.4.2.6 Dynamic behaviour

5.4.2.7 Test methods and expected result

5.4.3 Service interface - Get

5.4.3.1 Specification

The Get interface is used for pulling information from a service producer. The owner of the information (producer) is responsible for authorization procedure before returning information. The consumer can ask for information by its reference, identifier, status, geometry and time.

<Describe paging>
<Describe what to return if no parameters is provided, unfiltered, hence all authorized information is returned>

Table 10 – Parameter binding

Information (in)	Mult.	Definition
Reference	0..1	Reference to information object, e.g. from Get List
InformationIdentifier	0..1	Identifier on information object, e.g. from Get List
Status	0..1	Status in information object, e.g. from Get List
Geometry	0..1	Geometry condition for geolocated information objects
TimePeriod	0..1	Time related to validity of information objects
Information (out)	**Mult.**	**Definition**
List of Information Objects	0..*	Information object(s) in return

5.4.3.2 REST Design

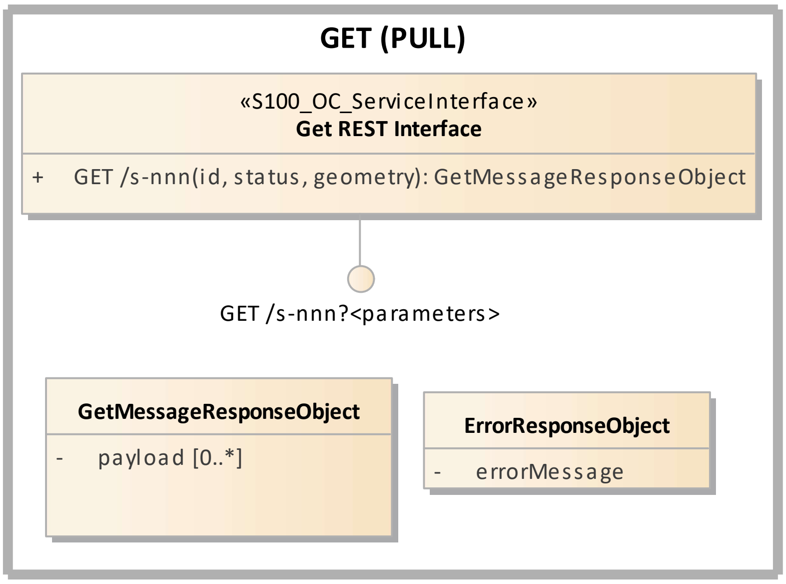

Figure 6 - Get REST interface

ExchangePattern

- REQUEST_RESPONSE

ConsumerInterface

- No interface consumed.

5.4.3.3 Operation - GET <s-nnn>

This operation receives a get request for information. If authorized, the data is sent back in the response. It's up to the service to apply relevant authorization procedure and access control to information.

Table 11 - Parameter binding for the operation

REST Parameter (in)	Type	Encoding	Mult.	Definition
informationIdentifier	String	depends on Information Product Specification	0..1	Identifier provided as search criteria. A list of identifiers can be retrieved via Get List interface.
reference	String	UUID	0..1	Information retrieved by using reference given in Get List
statusEnum	String	depends on Information Product Specification	0..1	Status on information as search criteria.
geometry	String	WKT	0..1	A geometry in WKT can be provided as search criteria.
Name of area				
unlocode				
fromTime	String	dateTime	0..1	Time related to validity period start for information object
toTime	String	dateTime		Time related to validity period end for information object
REST Body (in)	**Type**	**Encoding**	**Mult.**	**Definition**
-				
Return (out)	**Type**	**Encoding**	**Mult.**	**Definition**
GetMessageResponseObject ErrorResponseObject	Character String	application/json	1	Set of messages package with metadata or an error message

5.4.3.4 Data Exchange Model

Table 12 - Data exchange model for the REST operation

GetMessageResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
payload	string	According to S-Product Specification	0..*	The data according to Product Specification embedded in JSON. The data may be encrypted and signed.
ErrorResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
errorMessage	string	string	1	Error response message

5.4.3.5 Service response

Table 13 – HTTP Response codes

HTTP Code	Message
200	GetMessageResponseObject
403	Not authorized to requested information

404	Information with <informationIdentifier> not found
404	Information with <reference> not found
500	Parameter status=<status> not correct
500	Parameter id=<id> incorrect format

5.4.3.6 Dynamic behaviour

5.4.3.7 Test methods and expected result

5.4.4 Service interface - Get List

5.4.4.1 Specification

A list of information is returned from this interface. The list contains identity, status and short description of each information object. The actual information object can be retrieved using the Get interface.

Information (in)	Mult.	Definition
Geometry	0..1	Geometry related to information objects
TimePeriod	0..1	Time related to validity of information objects
Information (out)	**Mult.**	**Definition**
For each object: Reference Information Identifier Status Description of information object	0..*	

5.4.4.2 REST Design

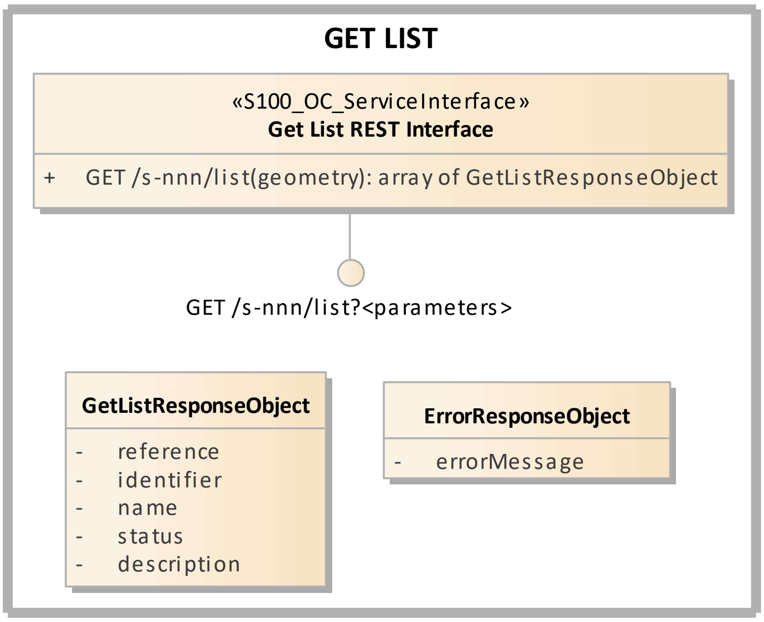

Figure 7 - Get List REST interface

ExchangePattern

- REQUEST_RESPONSE

ConsumerInterface

- No dependency.

5.4.4.3 Operation - GET <s-nnn>/list

This operation receives a get list request. The response contains a list of information objects that the requester is authorized to. The actual information objects can be retrieved through subscription request or get request.

Table 14 - Parameter binding

REST Parameter (in)	Type	Encoding	Mult.	Description
geometry	String	WKT	0..1	Geometry as search parameter in WKT format
fromTime	String	dateTime	0..1	Time related to validity period start for information object
toTime	String	dateTime		Time related to validity period end for information object
REST Body (in)	**Type**	**Encoding**	**Mult.**	**Description**
-				
Return (out)	**Type**	**Encoding**	**Mult.**	**Description**
array of GetListResponseObject ErrorResponseObject	Character String	application/json	1	List of information objects available (with access), identified by identity, status and short description or an error message

5.4.4.4 Data Exchange Model

Table 15 - Data exchange model for the REST operation

GetListResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
reference	string	UUID	1	Reference to object
identifier	string	depends on Information Product Specification	1	Identifier of the information object
name	string		1	Name of the information object
statusEnum	string	depends on Information Product Specification	0..1	Status of the information object
description	string		1	Description of the information object
ErrorResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
errorMessage	string	string	1	Error response message

5.4.4.5 Service response

Table 16 – HTTP Response codes

HTTP Code	Message
200	array of GetListResponseObjects
403	Not authorized to requested information
404	Information not found

5.4.4.6 Dynamic behaviour

5.4.4.7 Test methods and expected result

5.4.5 Service interface - Subscription

5.4.5.1 Specification

The purpose of the interface is to request subscription on information, either specific information according to parameters, or the information accessible upon decision by information provider.

Table 17 – Parameter binding

Information (in)	Mult.	Definition
Information Identifier	0..1	Identifier on information object
Status	0..1	Status on information object
Geometry	0..1	Geometry condition for geolocated information objects
Information (out)	**Mult.**	**Definition**
Subscription Identifier	0..1	Identifier for the created subscription, if authorized. To be used in remove subscription. If the subscription request corresponds to more than one information object, all information objects will be part of one subscription.

5.4.5.2 REST Design

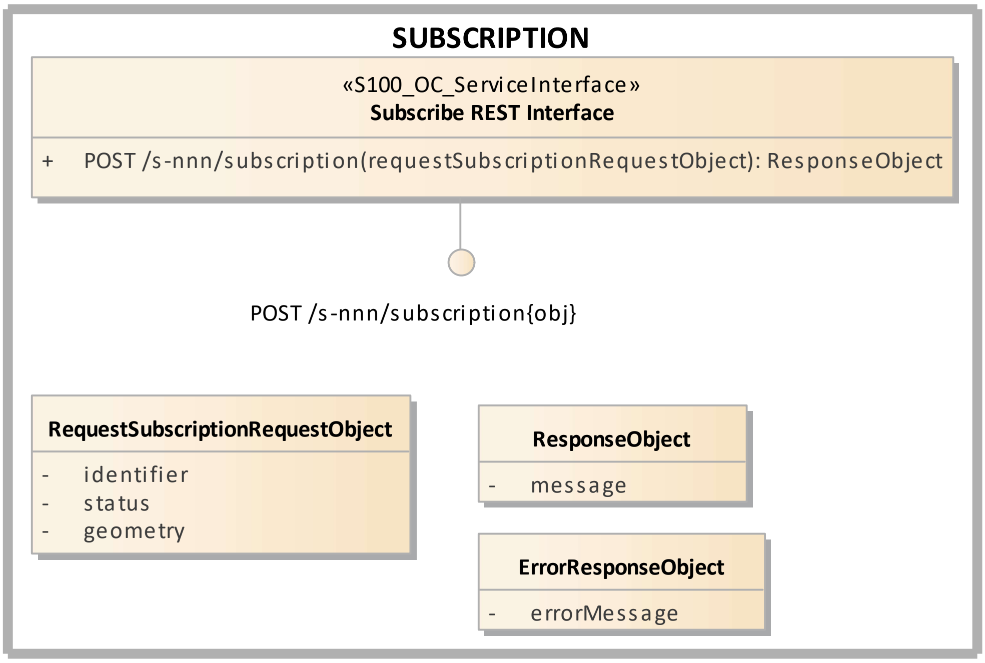
SUBSCRIPTION
«S100_OC_ServiceInterface»
Subscribe REST Interface
+ POST /s-nnn/subscription(requestSubscriptionRequestObject): ResponseObject
POST /s-nnn/subscription{obj}
RequestSubscriptionRequestObject
- identifier
- status
- geometry
ResponseObject
- message
ErrorResponseObject
- errorMessage

Figure 8 - Subscribe REST interface

ExchangePattern

- PUBLISH_SUBSCRIBE

ConsumerInterface

- Upload service interface
- Subscription Notification service interface

5.4.5.3 Operation - POST <s-nnn>/subscription

This operation receives a request for subscription. If the requester is authorized, a subscription is created and the latest message is sent.

Table 18 - Parameter binding for the operation

REST Parameter (in)	Type	Encoding	Mult.	Definition
-				
REST Body (in)	**Type**	**Encoding**	**Mult.**	**Definition**
RequestSubscriptionRequestObject	String	application/json	0..1	Specific id on the information object subscription is requested for Specific status on the information Specific area of interest in WKT format
Return (out)	**Type**	**Encoding**	**Mult.**	**Definition**
ResponseObject ErrorResponseObject	String	application/json	1	Confirmation or error message Subscription identifier in return, if ok.

5.4.5.4 Data Exchange Model

Table 19 - Data exchange model for the REST operation

RequestSubscriptionRequestObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
identifier	string	depends on Information Product Specification	0..1	Identifier of the information object
statusEnum	string	depends on Product Specification	0..1	Status in the information object
geometry	string	WKT	0..1	Geometry for geolocated request for subscription,
ResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
message	string			Success response message
ErrorResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
errorMessage	string	string	1	Error response message

5.4.5.5 Service response

Table 20 – HTTP Response codes

HTTP Code	Message
200	Successfully received <reference>
403	Not authorized to requested information
404	Information not found

5.4.5.6 Dynamic behaviour

5.4.5.7 Test methods and expected result

5.4.6 Service interface - Remove Subscription

5.4.6.1 Specification

Subscription(s) can be removed either internally by information owner, or externally by the consumer. This interface is used by the consumer to request removal of subscription.

Table 21 – Parameter binding

Information (in)	Mult.	Definition
Subscription Identifier	0..1	Identifier of the subscription. Retrieved when subscription was created
Information (out)	**Mult.**	**Definition**
result from operation	1	

5.4.6.2 REST Design

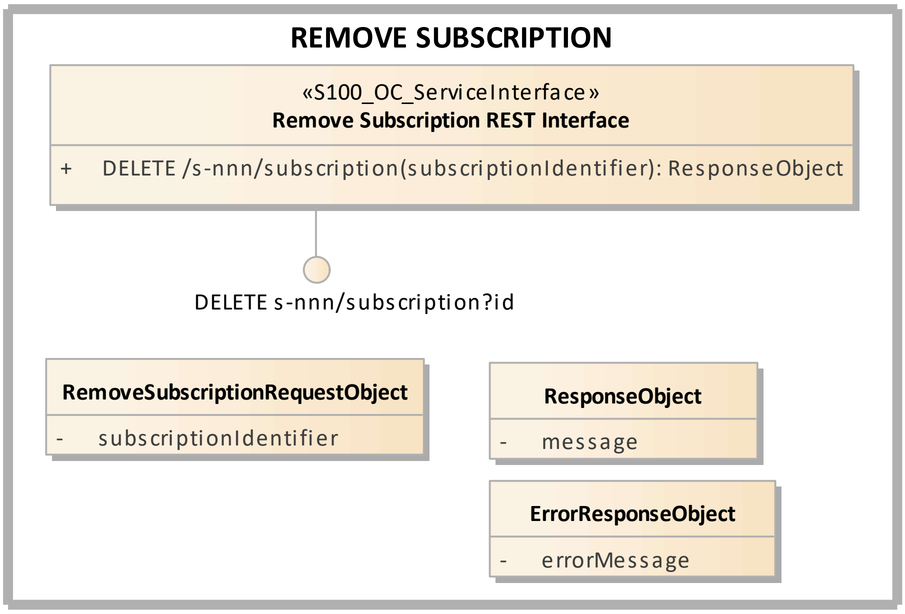
REMOVE SUBSCRIPTION
«S100_OC_ServiceInterface»
Remove Subscription REST Interface
+ DELETE /s-nnn/subscription(subscriptionIdentifier): ResponseObject
DELETE s-nnn/subscription?id
RemoveSubscriptionRequestObject
- subscriptionIdentifier
ResponseObject
- message
ErrorResponseObject
- errorMessage

Figure 9 - Remove Subscription REST interface

ExchangePattern

- ONE_WAY

ConsumerInterface

- Subscription Notification service interface

5.4.6.3 Operation - DELETE <s-nnn>/subscription

This operation receives a request to remove subscription.

Table 22 - Parameter binding for the operation

REST Parameter (in)	Type	Encoding	Mult.	Description
-				
REST Body (in)	**Type**	**Encoding**	**Mult.**	**Description**
RemoveSubscriptionRequestObject	Character String	application/json	0..1	Specific identity of the information object to remove subscription for. If no id entity provided, all subscriptions for the caller is removed
Return (out)	**Type**	**Encoding**	**Mult.**	**Description**
ResponseObject ErrorResponseObject	Character String	application/json	1	Confirmation or error message

5.4.6.4 Data Exchange Model

Table 23 - Data exchange model for the REST operation

RemoveSubscriptionRequestObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
subscriptionIdentifier	string	UUID	1	Subscription identifer to the subscripotion to be removed. Received in subscription request or subscription notification.
ResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
message	string		1	Success response message
ErrorResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
errorMessage	string	string	1	Error response message

5.4.6.5 Service response

Table 24 – HTTP Response codes

HTTP Code	Message
200	Subscription <identifier> removed
403	Not authorized to remove subscription
404	Subscriber identifier not found

5.4.6.6 Dynamic behaviour

5.4.6.7 Test methods and expected result

5.4.7 Service Interface - Get Subscription List

5.4.7.1 Specification

A list of active subscriptions is retrieved through this interface.

Table 25 – Parameter binding

Parameters (in)	Mult.	Definition
-		
ReturnType (out)	**Mult.**	**Definition**
Subscription Identifier	0..*	Identifier of the subscription.
Subscription Parameters	0..*	Parameters used when creating subscription.

5.4.7.2 REST Design

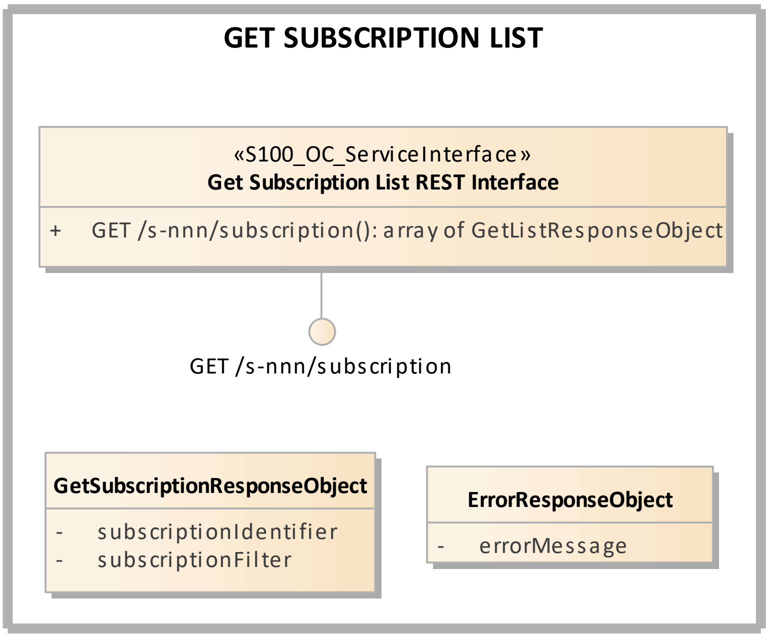
GET SUBSCRIPTION LIST
«S100_OC_ServiceInterface»
Get Subscription List REST Interface
+ GET /s-nnn/subscription(): array of GetListResponseObject
GET /s-nnn/subscription
GetSubscriptionResponseObject
- subscriptionIdentifier
- subscriptionFilter
ErrorResponseObject
- errorMessage

Figure 10 - Get Subscription List REST interface

ExchangePattern

- REQUEST_RESPONSE

ConsumerInterface

- No dependency.

5.4.7.3 Operation - GET <s-nnn>/subscription

This operation receives a request for a list of active subscription(s) for the requester. The response contains all active subscriptions for the requester.

Table 26 - Parameter binding for the operation

REST Parameter (in)	Type	Encoding	Mult.	Definition
-				
REST Body (in)	**Type**	**Encoding**	**Mult.**	**Definition**
-				
Return (out)	**Type**	**Encoding**	**Mult.**	**Definition**
array of GetListResponseObject ErrorResponseObject	Character String	application/json	1	List of identities or error message

5.4.7.4 Data Exchange Model

Table 27 - Data exchange model for the REST operation

GetListResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
subscriptionIdentifier	string	UUID	1	Subscription identifier. Can be used when removing the subscription.
subscriptionFilter	string	SubscriptionFilter	0..*	Description of the subscription filter parameters
SubscriptionFilter				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
key	string		1	Filter parameter key
value	string		1	Filter parameter value
ErrorResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
errorMessage	string	string	1	Error response message

5.4.7.5 Service response

Table 28 – HTTP Response codes

HTTP Code	Message
200	array of GetListResponseObjects
403	Not authorized to requested information

5.4.7.6 Dynamic behaviour

5.4.7.7 Test methods and expected result

5.4.8 Service interface - Subscription Notification

5.4.8.1 Specification

The interface receives notifications when subscription is created or removed by information producer.

Table 29 – Parameter binding

Parameters (in)	Mult.	Definition
Subscription Identifier	1	Identifier of the subscription.
Event	1	Type of event; Create, Delete
ReturnType (out)	**Mult.**	**Definition**
result from operation	1	

5.4.8.2 REST Design

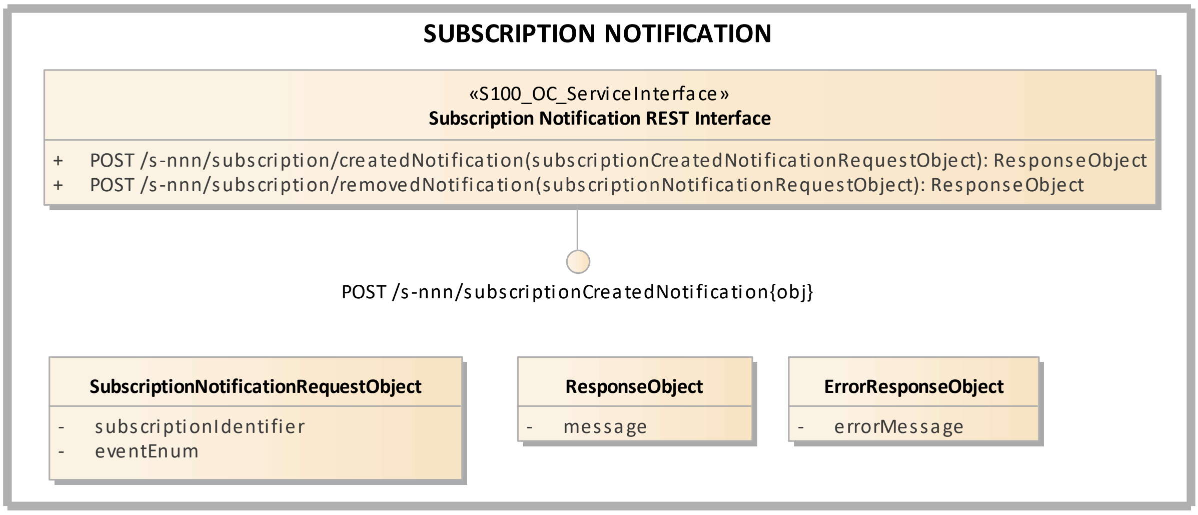
SUBSCRIPTION NOTIFICATION
«S100_OC_ServiceInterface»
Subscription Notification REST Interface
+ POST /s-nnn/subscription/createdNotification(subscriptionCreatedNotificationRequestObject): ResponseObject
+ POST /s-nnn/subscription/removedNotification(subscriptionNotificationRequestObject): ResponseObject
POST /s-nnn/subscriptionCreatedNotification{obj}
SubscriptionNotificationRequestObject
- subscriptionIdentifier
- eventEnum
ResponseObject
- message
ErrorResponseObject
- errorMessage

Figure 11 - Subscription Notification REST interface

ExchangePattern

- ONE_WAY

ConsumerInterface

- No dependency.

5.4.8.3 Operation - POST <s-nnn>/subscription/notification

This operation receives notification when subscription is created, either internally by information producer, or externally on request by consumer.

Table 30 - Parameter binding for the operation

REST Parameter (in)	Type	Encoding	Mult.	Description
-				
REST Body (in)	**Type**	**Encoding**	**Mult.**	**Description**
SubscriptionNotificationRequestObject	String	application/json		Contains the identity of the information object in focus and type of event; Create or Delete.
Return (out)	**Type**	**Encoding**	**Mult.**	**Description**
ResponseObject ErrorResponseObject	String	application/json		Confirmation or error message

5.4.8.4 Data Exchange Model

Table 31 - Data exchange model for the REST operation

SubscriptionNotificationRequestObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
subscriptionIdentifier	string		1	
eventEnum	string	EnumEvent	1	
ResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
message	string		1	Success response message
ErrorResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
errorMessage	string		1	Error response message
EnumEvent				
Enum	**Type**	**Encoding**		**Definition**
Create	string			Create event
Delete	string			Delete event

5.4.8.5 Service response

Table 32 – HTTP Response codes

HTTP Code	Message
200	tbd
403	Not authorized to requested information

5.4.8.6 Dynamic behaviour

5.4.8.7 Test methods and expected result

5.4.9 Service interface - Request Access

5.4.9.1 Specification

Access to information can be requested through the Access interface. The result is sent asynchronous through the Access Notification interface.

Table 33 – Parameter binding

Parameters (in)	**Mult.**	**Definition**
Reason	1	Human readable reason why information is requested
ReasonEnum	1	Machine readable
Status		e.g. routeStatus Monitored
ReturnType (out)	**Mult.**	**Definition**
result from operation	1	

5.4.9.2 REST Design

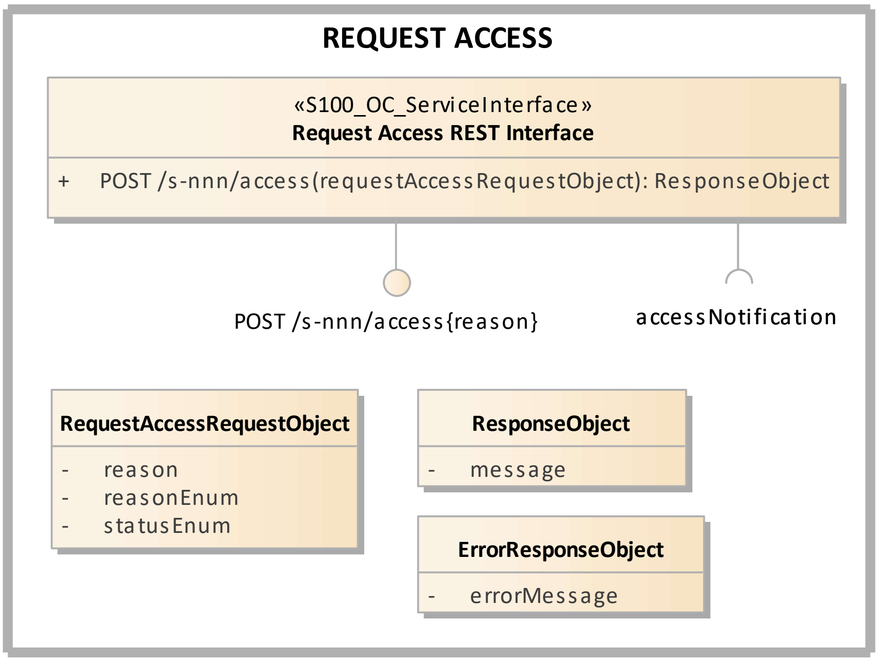
REQUEST ACCESS
«S100_OC_ServiceInterface»
Request Access REST Interface
+ POST /s-nnn/access(requestAccessRequestObject): ResponseObject
POST /s-nnn/access{reason}
accessNotification
RequestAccessRequestObject
- reason
- reasonEnum
- statusEnum
ResponseObject
- message
ErrorResponseObject
- errorMessage

Figure 12 - Request Access REST interface

ExchangePattern

- REQUEST_CALLBACK

ConsumerInterface

- Access Notification service interface

5.4.9.3 Operation - POST <s-nnn>/access/request

This operation receives a requerst for access by a consumer. The result is sent asynchronously.

Table 34 - Parameter binding for the operation

REST Parameter (in)	Type	Encoding	Mult.	Definition
-				
REST Body (in)	**Type**	**Encoding**	**Mult.**	**Definition**
RequestAccessRequestObject	Character String	application/json	1	Description of reason for requesting access to information
Return (out)	**Type**	**Encoding**	**Mult.**	**Definition**
ResponseObject ErrorResponseObject	Character String	application/json	1	Confirmation or error message. Result from request is sent asynchronous through Access Notification interface.

5.4.9.4 Data Exchange Model

Table 35 - Data exchange model for the REST operation

RequestAccessRequestObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
reason	string		1	Human readable reason for requesting access
reasonEnum	string	enumReason	1	Machine readable reason for requesting access
statusEnum	string	EnumStatus	0..1	Status on information object requesting access to
ResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
message	string		1	Success response message
ErrorResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
errorMessage	string		1	Error response message
EnumReason				
Enum	**Type**	**Encoding**	**Mult.**	**Definition**
TBD				
EnumStatus				
Enum	**Type**	**Encoding**	**Mult.**	**Definition**
TBD				<Depends on Information Product Specification>

5.4.9.5 Service response

Table 36 – HTTP Response codes

HTTP Code	Message
200	tbd
403	Not authorized to requested information

5.4.9.6 Dynamic behaviour

5.4.9.7 Test methods and expected result

5.4.10 Service interface - Access Notification

5.4.10.1 Specification

Result from Access Request is sent asynchronous through this interface.

Table 37 – Parameter binding

Parameters (in)	Mult.	Definition
Decision	1	Decision
Reason	1	Human readable reason for decision
ReasonEnum	1	Machine readable reason for decision

ReturnType (out)	Mult.	Definition
result from operation	1	

5.4.10.2 REST Design

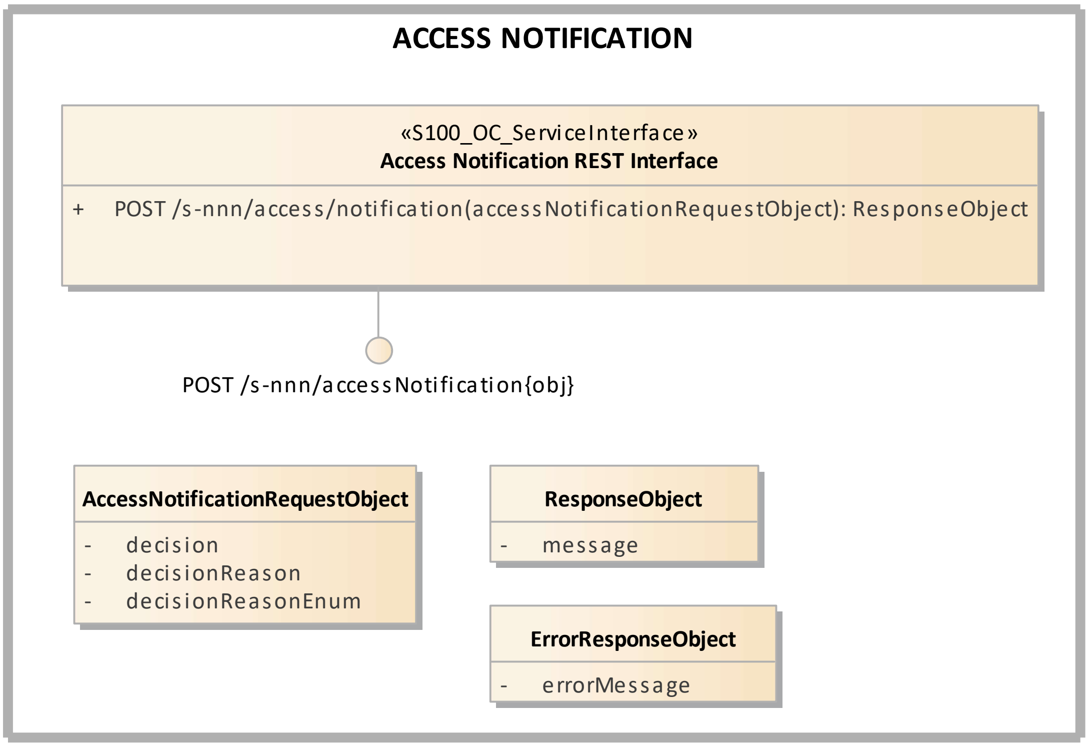
ACCESS NOTIFICATION
«S100_OC_ServiceInterface»
Access Notification REST Interface
+ POST /s-nnn/access/notification(accessNotificationRequestObject): ResponseObject
POST /s-nnn/accessNotification{obj}
AccessNotificationRequestObject
- decision
- decisionReason
- decisionReasonEnum
ResponseObject
- message
ErrorResponseObject
- errorMessage

Figure 13 - Access Notification REST interface

ExchangePattern

- ONE_WAY

ConsumerInterface

- No dependency.

5.4.10.3 Operation - POST <s-nnn>/access/notification

This operation receives result from access request.

Table 38 - Parameter binding for the operation

REST Parameter (in)	Type	Encoding	Mult.	Definition
-				
REST Body (in)	**Type**	**Encoding**	**Mult.**	**Definition**
AccessNotificationRequestObject	Character String	application/json	1	Result from the request for access; True or False
Return (out)	**Type**	**Encoding**	**Mult.**	**Definition**
ResponseObject ErrorResponseObject	Character String	application/json	1	Confirmation or error message

5.4.10.4 Data Exchange Model

Table 39 - Data exchange model for the REST operation

AccessNotificationRequestObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
decision	boolean		1	Access request decision, yes or no
decisonReason	string		1	Human readable reason for decision
decisionReasonEnum	string	DecisionReasonEnum	1	Machine readable for decision
ResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
message	string		1	Success response message
ErrorResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
errorMessage	string		1	Error response message
DecisionReasonEnum				
Enum	**Type**	**Encoding**		**Definition**
TBD				

5.4.10.5 Service response

Table 40 – HTTP Response codes

HTTP Code	Message
200	tbd
403	Not authorized to requested information

5.4.10.6 Dynamic behaviour

5.4.10.7 Test methods and expected result

5.4.11 Service interface - Capability

5.4.11.1 Specification

The purpose of the interface is to provide a dynamic method to ask a service instance at runtime what interfaces that are accessible, and what payload formats and version that are valid.

Table 41 – Parameter binding

Parameters (in)	Mult.	Definition
-		
ReturnType (out)	**Mult.**	**Definition**
Capability	1	Description of the service interface capability. The service interface capability includes flag that indicate which interfaces the service are capable to handle, which information products the service are capable to handle

5.4.11.2 REST Design

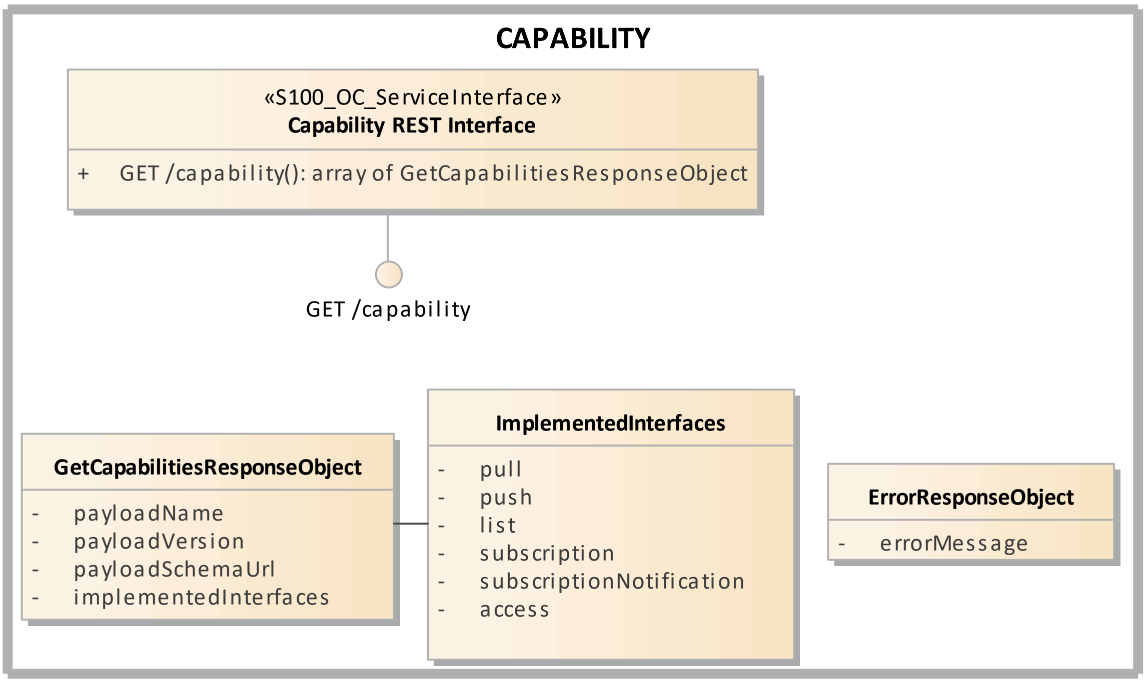

Figure 14 - Capability REST interface

ExchangePattern

- REQUEST_RESPONSE

ConsumerInterface

- No dependency.

5.4.11.3 Operation - GET capability

This operation receives request for service interface capabilities. The result contains the interfaces provided by the service instance, and the information product type handled.

Table 42 - Parameter binding for the operation

REST Parameter (in)	Type	Encoding	Mult.	Definition
-				
REST Body (in)	**Type**	**Encoding**	**Mult.**	**Definition**
-				
Return (out)	**Type**	**Encoding**	**Mult.**	**Definition**
array of GetCapabilitiesResponseObject ErrorResponseObject	Character String	application/json	1	Description of service capabilities; Which interfaces that are valid for the specific service instance, the accepted payload format and version, and specific requirements in payload etc.

5.4.11.4 Data Exchange Model

Table 43 - Data exchange model for the REST operation

GetCapabilitiesResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
payloadName	string		1	
payloadVersion	string		1	
payloadSchemaUrl	string	URL	1	
implementedInterfaces	ImplementedInterfaces		1	
ImplementedInterfaces				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
push	boolean		1	
pull	boolean		1	
list	boolean		1	
subscription	boolean		1	
access	boolean		1	
description	boolean		1	
ErrorResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
errorMessage	string		1	Error response message

5.4.11.5 Service response

Table 44 – HTTP Response codes

HTTP Code	Message
200	GetCapabilityResponseObject
403	Not authorized to requested information

5.4.11.6 Dynamic behaviour

5.4.11.7 Test methods and expected result

5.4.12 Service interface - Description

5.4.12.1 Specification

The purpose of the interface is to provide a dynamic method to ask for operational/user description of the specific service instance.

The response from the service contains how to use/consume the service instance and expected input and outcome of the service instance.

Table 45 – Parameter binding

Parameters (in)	Mult.	Definition
-		
ReturnType (out)	**Mult.**	**Definition**
Description	1	Description of the service reported at runtime by the service instance

5.4.12.2 REST Design

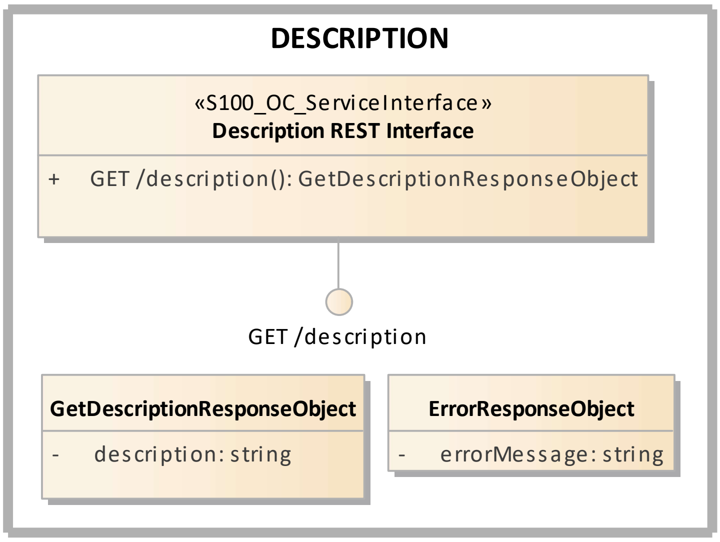
DESCRIPTION
«S100_OC_ServiceInterface»
Description REST Interface
+ GET /description(): GetDescriptionResponseObject
GET /description
GetDescriptionResponseObject
- description: string
ErrorResponseObject
- errorMessage: string

Figure 15 - Description REST interface

ExchangePattern

- REQUEST_RESPONSE
- ConsumerInterface
  No dependency.

5.4.12.3 Operation - GET description

This operation receives a request for description on the service instance.

Table 46 - Parameter binding for the operation

REST Parameter (in)	Type	Encoding	Mult.	Description
-				
REST Body (in)	**Type**	**Encoding**	**Mult.**	**Description**
-				
Return (out)	**Type**	**Encoding**	**Mult.**	**Description**
GetDescriptionResponseObject ErrorResponseObject	Character String	application/json	1	Description of the service instance.

5.4.12.4 Data Exchange Model

Table 47 - Data exchange model for the REST operation

GetDescriptionResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
description	string	TBD?	1	Short operational description of the service instance
ErrorResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
errorMessage	string		1	Error response message

5.4.12.5 Service response

Table 48 – HTTP Response codes

HTTP Code	Message
200	GetDescriptionResponseObject
403	Not authorized to requested information

5.4.12.6 Dynamic behaviour

5.4.12.7 Test methods and expected result

5.4.13 Service interface - EncryptionKey

5.5 Interface for other S-products

The REST service interface above is described for a generaic S-product with the placeholder <s-nnn>. This chapter describes how to apply the service interface for a specific S-product, such as S-421.

The following steps describe how to tailor the standard REST service interface for a specific information product, such as S-421.

1) Replace <s-nnn> with s-421
2) Define encoding and enumeration for Status
3) Define encoding for Identity
4) Define encoding and enumeration for Purpose
5) tbc

See ANNEX A for a swagger file (interface description) for S-421 REST service interface.

5.6 Service composition

To be elaborated in work group.

Example/description where a service can be composed to handle several information products.

E.g. A service (such as a ship voyage information service) may be capable of receiving both S-421 and S-124 information products. Either two services are created, or one composed service.

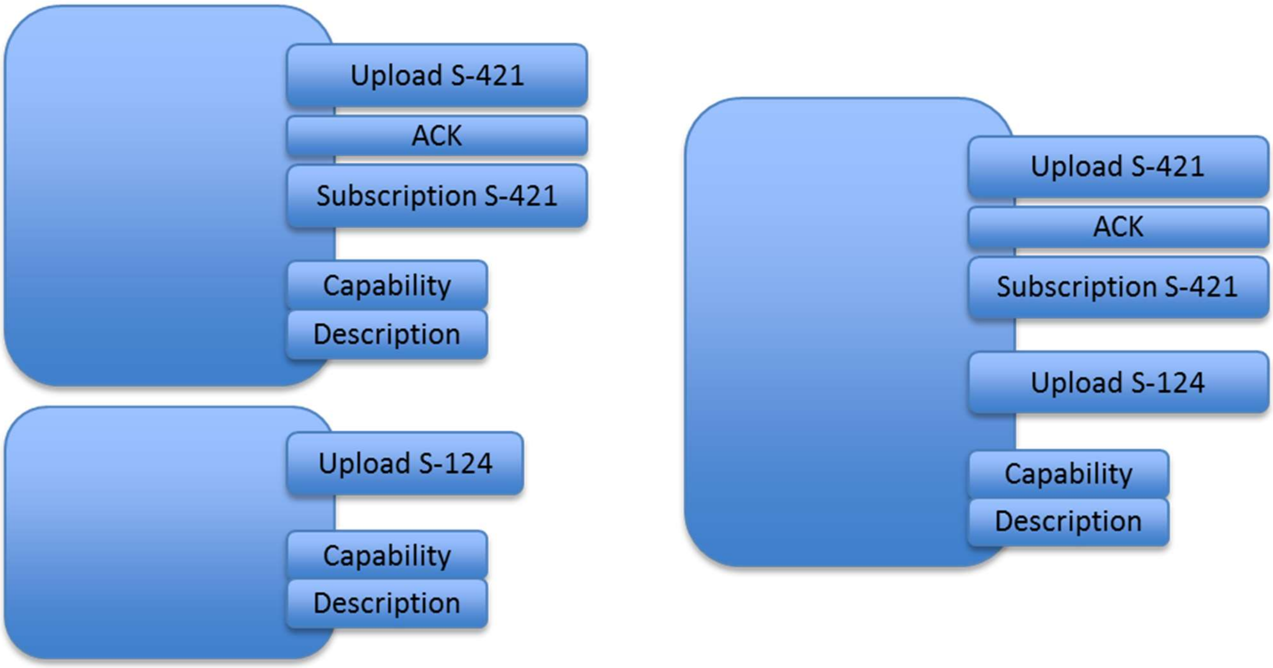

Figure 16 - Service composition example

6 Communication Protection

6.1 General

The rationale to standardise the communication protection scheme is to gain interoperability. Interoperability can only be gained if all participants use the same communication protection scheme, weather its open or secured.

This chapter describes technology and procedure for the protection of IP and HTTP based communication channel for web services.

6.2 Secure transfer

6.2.1 Secure communication channel

Encryption of channel by the use of Transport Layer Security, TLS according to valid version.

Valid version at the time of writing is TLS version 1.0 (RFC-2246), TLS version 1.1 (RFC-4346), TLS version 1.2 (RFC-5246) and TLS version 1.3 (RFC-8446).

TLS over HTTP shall be used according to RFC-2818.

HTTP/1.1 shall be used according to RFC-7231.

This standard describe business-to-business communication with mutual authentication, where both client and server authenticate each other using certificate-based TLS mutual authentication (TLS client-side X.509 authentication).

Service authentication shall be based on X.509 Certificates (RFC-5280 and RFC-2459).

Service certificates shall be possible to revoke, hence the authentication procedure shall contain check against Certificate Revocation List, CRL.

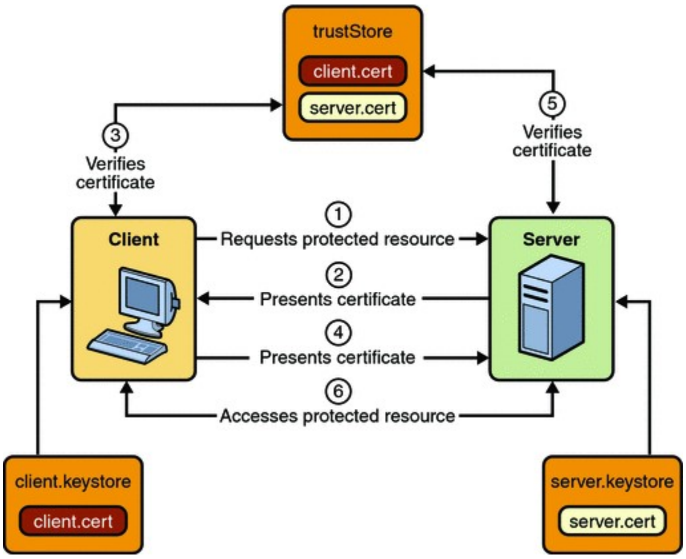

Figure 17 - Principle for mutual authentication

6.3 Authentication procedure

The procedure to verify the server certificate shall be:

1. The client shall verify that the server’s certificate is valid.
2. The client shall verify that the server’s certificate is issued by trusted CA
3. The client shall verify that the server’s domain name corresponds to the domain name in the certificate

The procedure to verify the client certificate shall be:

1. The server shall verify that the client’s certificate is valid
2. The server shall verify that the client is issued by trusted CA

6.4 Scheme Administrator Processes

To be elaborated in work group.

Scheme administrator processes includes procedure to issue Certificate for client, for server and revoke certificate, Vetting procedure and ceritificate distribution procedure.

There are several possibilities for administrator processes, either IHO similar to the Data Protection Scheme described in S-100 v4 or commercial alternatives such as Maritime Connectivity Platform (MCP).

7 Data protection

7.1 General

7.2 Data authentication and signing

Tbd

7.3 Data encryption

8 Service Discoverability

8.1 General

The rationale to standardise the service discoverability is to gain interoperability for searching for the service to consume.

This chapter describes interfaces and procedures to enable service discoverability. The interface has been tested in real use between 2017 and 2018.

First section describe how to discover (find) a service to consume. The scope for this chapter is how to to discover service instance to consume. This section contains a service interface to search for service instance.

The second section describe how to make the service discoverable for others to consume.

8.2 Discover Service

8.2.1 Search Parameters

To be elaborated

Based on IALA Service Model and attributes used to describe the service instance.

The S-100 v4 proposed Service Model does not have the service instance in scope.

8.2.2 Content in result

To be elaborated

The content in the result shall be enough data to consume the service.

- Service Metadata for the instance to consume
- Reference to Interface Description, WSDL for SOAP, SWAGGER for REST

8.2.3 Service interface – Search Service

8.2.3.1 Purpose

The purpose with this interface is to search for service instance to consume.

8.2.3.2 Operation – Search Service

The operation shall be used for searching for service instance(s) to consume based on provided parameters.

Table 49 - Parameter binding for operation Search Service

Parameter (in)	Encoding	Multiplicity	Definition
Query		1	
Geometry	WKT	1	
Return (out)	**Encoding**	**Multiplicity**	
<attributes based on Service Data Model>	JSON	1	

ConsumerInterface

- No dependencies.

ExchangePattern

- REQUEST_RESPONSE

8.2.3.3 REST service interface – Search/Find Service

ServiceTechnology

- REST

8.2.3.4 Operation - GET /service-instance/search

This operation searches for service. If no parameters given, a complete list of services is given in result.

8.2.3.4.1 ParameterBinding

Table 50 - Parameter binding for REST operation Get Service

REST Parameter (in)	Type	Encoding	Mult.	Definition
query	String	[+-]key:value AND OR ()	0..1	Search parameters
geometry	String	WKT	0..1	WKT
page	Integer		0..1	
pageSize	Integer		0..1	
REST Body (in)	**Type**	**Encoding**	**Mult.**	**Definition**
-				
Return (out)	**Type**	**Encoding**	**Mult.**	**Definition**
FindServiceResponseObject ErrorResponseObject	String	application/json	0..*	list of services

8.2.3.4.2 Data Exchange Model

Table 51 - Data exchange model for the REST operation

FindServiceResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
instanceId	string	MRN	1	Service instance identity
version	string		1	Version of the service instance
name	string		1	Name on the service instance
status	string	S100_OC_StatusType	1	Status on the service
comment/description	string		1	Description of the service instance

organizationId	string		1	Organization identity of the registrator
endpointUri	string	URI	1	Endpoint to the service instance
endpointType/technology	string	S100_OC_ServiceTechnology	1	Service technology used for the service instance
keywords	string		0..1	Searchable keywords for discoverability
unlocode	string		0..1	Geolocation by UnLoCode
instanceAsXml	string	XML	0..1	Original XML based on service registry used.
ErrorResponseObject				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**
errorMessage	string		1	Error response message

Enumeration for filter parameters (keys) are, but not limited to:

Table 52 - Search parameters

enumKey				
Enum	**Type**	**Encoding**		**Definition**
name	string			From S-100 Service Model
status	string			From S-100 Service Model
version	string			From S-100 Service Model
keywords	string			From S-100 Service Model
technology	string			From S-100 Service Model
description	string			From S-100 Service Model
specificationId	string	MRN		From IALA Service Model
designId	string	MRN		From IALA Service Model
instanceId	string	MRN		From IALA Service Model
comment				From IALA Service Model
mmsi				From IALA Service Model
imo				From IALA Service Model
serviceType				From IALA Service Model
unlocode				From IALA Service Model
endpointUri		URI		From IALA Service Model

8.2.3.4.3 Swervice response

Table 53 – HTTP Response codes

HTTP Code	Message
200	Array of FindServiceResponseObject
403	Not authorized

8.3 Make service discoverable

To be elaborated in work group.

Service is expected to be registered in a well-known and available service registry in order to be discoverable.

- Selection of service registry where service will be discoverable/visible
- Service Definition Model
  The descriptive model for service is the base for the data into service registry, and data that are used to make the service discoverable.
- Complemented with service instance metadata, such as the URL

9 Key management interface on PKI System

9.1 General

<Description of interface for public and private key generation, upload and download>

<Description of interface for revoking keys>

<Description of check for revoked key>

Annex A
(normative)

REST Service Swagger

A.1 Purpose

This annex contains an realization of the REST service interface for S-421 based on the proposed standard REST service interface.

A-2 Service REST Technical Design S-421

http://stmstandardvis.azurewebsites.net/swagger/index.html

Annex B
(informative)

Use Cases and profiles

B.1 Purpose

This annex contains a set of use cases and profiles that describe the usage of service interface in operational context.

Use cases are operational scenarios where the service usage can be derived.

To be elaborated in work group.

TASK-20 Add use case for Service Discovery

TASK-XX Add different acknowledgements in the use cases

The service interface described in this document is focused on the exchange of information products and not designed towards a specific operational service. Different operational services will require different exchange patterns and also different sets of active interfaces such as subscription. An enhanced monitoring service for example will most likely not allow ships to subscribe to route plans from them, but the enhanced monitoring service will want to subscribe to route plans from the ship.

This chapter describe use cases or profiles that describe usage of the service interface in different operational contexts.

The use cases from the S-421 have been used as basis.

List of use case

- UC-1 Ship shares route plan with service providing enhanced monitoring
- Enhanced monitoring service requests route plan from/for ship for monitoring
- Service consumer request capability description of a service interface
- Service consumer requests status from service
- Service consumer presents a list of services to consume
- Ship requests pilot routes
- Ship requests ice routes
- Under keel clearance
- tbc

Based on a set of defined use cases for a certain actor, the profile for that actor can be defined. Hence, a ship service interface profile is the sum of all service interfaces used in the use cases for the ship.

B.2 Use Cases and Service Interface Profiles

B.2.1 UC-1 Ship shares route plan with service providing enhanced monitoring

Pre-requisites

Service provider of enhanced monitoring exposes discoverable service interface to receive S-421 Route Plan, receive acknowledgement and receive subscription notifications.

Ship exposes discoverable service interface to receive S-421 Route Plan proposal.

Description

1) The ship finds enhanced monitoring service, nominates and initiates enhanced monitoring by sending (uploading, pushing) the S-421 Route Plan to. The ship creates an internal subscription for the enhanced monitoring service and continue sending updated S-421 Route Plans.
2) The enhanced monitoring service makes a proposal and sends (uploads) the proposed S-421 Route Plan to the ship and requests acknowledgement when delivered to the ship.
3) When voyage is finalised, the subscription is removed by the ship and a notification is sent to the enhanced monitoring service.

Service Interfaces required

Table B.1 describe used interfaces in this use case.

Table B.1 – Required service interfaces in UC-1

Service interface	Ship	Enhanced Monitoring
Upload	YES (step 2)	YES (step 1)
Acknowledgement		YES (step 2)
Get		
Get List		
Access		
Access Notification		
Subscription		
Subscription Notification		YES (step 3)
Remove Subscription		
Get Subscription List		
Capability		
Status		
Description		

B.2.2 UC-2 Enhanced monitoring service requests route plan from/for ship for monitoring

Pre-requisites

Ship has not shared S-421 Route Plan with enhanced monitoring service.

Ship exposes discoverable service interface to receive S-421 Route Plan proposal.

Service provider of enhanced monitoring exposes discoverable service interface to receive S-421 Route Plan, receive acknowledgement and receive subscription notifications.

Description

Enhanced monitorng operator identitfies ship through AIS and wants the ships route plan. Operator search for service using MMSI as search parameter.

1. Enhanced monitoring requests subscription on S-421 Route Plan from the ship.
2. The ship check access rights to its route plan
    a. If access; The ship sends (uploads) the S-421 Route Plan
    b. If subscription is created, the ship notifies subscription created.
    c. If no access; The ship respond with 403 NOT AUTHORIZED
    See further Use case for no access.
3. When voyage is finalised, the subscription is removed by the ship and a notification is sent to the enhanced monitoring service.

Service Interfaces required

Table B.2 describe used interfaces in this use case.

Table B.2 – Required service interfaces in UC-2

Service interface	Ship	Enhanced Monitoring
Upload		YES (step 2a)
Acknowledgement		
Get		
Get List		
Access		
Access Notification		
Subscription	YES (step 1)	
Subscription Notification		YES (step 2b, 3)
Remove Subscription		
Get Subscription List		
Capability		
Status		
Description		

B.2.3 UC-3 Service unauthorized to data

Pre-requisites

Ship has not shared S-421 Route Plan with enhanced monitoring service.

Ship exposes discoverable service interface to request access to its S-421 Route Plan.

Service provider of enhanced monitoring exposes discoverable service interface to receive Access Notification as a result from access request.

Description

Enhanced monitoring get “not authorized” response from ship when requesting subscription or pulling information from ship.

1. Enhanced monitoring sends Access request with the reason “Enhanced monitoring service for area X”.

2. Ship receives and displays access request from enhanced monitoring service

    a. Ship accepts the request

    b. Ship rejects the request

Service Interfaces required

Table B.3 describe used interfaces in the use case.

Table B.3 – Required service interfaces for UC-3

Service interface	Ship	Enhanced Monitoring
Upload		
Acknowledgement		
Get		
Get List		
Access	YES (step 1)	
Access Notification		YES (step 2)
Subscription		
Subscription Notification		
Remove Subscription		
Get Subscription List		
Capability		
Status		
Description		

B.2.4 UC-4 Service check capability of consumer interface

Pre-requisites

Service producer exposes discoverable service interface to request dynamic service interface capability.

Description

Service consumer wants to interact with service producer.

1. Service consumer asks for the producers service interface capability to find different possibilities

The service consumer now knows which information products, including version, the service producer is capable to handle, and which service interfaces the service producer is capable to handle.

Based on capability description, the enhanced monitoring selects the adeqauate service interface to consume.

Service Interfaces required

Table B.4 describe used interfaces in the use case.

Table B.4 - Required service interfaces for UC-4

Service interface	Service producer	Service consumer
Upload		
Acknowledgement		
Get		
Get List		
Access		
Access Notification		
Subscription		
Subscription Notification		
Remove Subscription		
Get Subscription List		
Capability	YES (step 1)	
Status		
Description		

Annex C
(informative)

<Annex removed (requirements)>

Annex D
(informative)

S100 Service Model traceability

D.1 Purpose

This annex contains traceability between S100 v4 service definition model and the description of service in this standard.

D.2 Service Interface Specification

The following sections in this chapter contains a specification of the standard service according to S100 v4 service model.

Each section below describes one service interface as follows;

Service interface - <name of the service interface>

<description of the service interface>

ConsumerInterface
<S100_OC_ConsumerInterface.name>

ExchangePattern
<S100_OC_ExchangePattern.enumeration value>

Operation - <S100_OC_Operation.name>

<S100_OC_Operation.description>

Parameters (in) **S100_OC_Operation role “parameters”**	**Encoding**	**Mult.**	**Definition**
S100_OC_Parameter.name	S100_OC_ParameterBinding.encoding	S100_OC_ParameterBinding.multiplicity	S100_OC_Parameter.definition
ReturnType (out) **S100_OC_Operation role “returnType”**	**Encoding**	**Mult.**	
S100_OC_Parameter.name	S100_OC_ParameterBinding.encoding	S100_OC_ParameterBinding.multiplicity	S100_OC_Parameter.definition

D.3 Service REST Technical Design

The following sections in this chapter contains a REST Technical Design of the standard service according to S100 v4 service model.

Each section below describes one service interface as follows;

REST Service interface - <name of the service interface>

<description of the service interface>

ServiceTechnology
<S100_OC_ServiceInterface.technology.enumeration value>

ExchangePattern
<S100_OC_ExchangePattern.enumeration value>

ConsumerInterface
<S100_OC_ConsumerInterface.name>

Operation - <S100_OC_Operation.name>

<S100_OC_Operation.description>

ParameterBinding

REST Parameter (in) **S100_OC_Operation role “parameters”**	**Type**	**Encoding**	**Mult.**	**Definition**
S100_OC_Parameter.name		S100_OC_ParameterBinding.encoding	S100_OC_ParameterBinding.multiplicity	S100_OC_Parameter.definition
REST Body (in)	**Type**	**Encoding**	**Mult.**	**Definition**
Return (out) **S100_OC_Operation role “returnType”**	**Type**	**Encoding**	**Mult.**	**Definition**
				S100_OC_Parameter.definition

Data Exchange Model

<object>				
Attribute	**Type**	**Encoding**	**Mult.**	**Definition**

Annex E
(informative)

Message Exchange Patterns

E.1 Purpose

This annex contains the used message exchange patterns.

E.2 Message Exchange Pattern

This chapter contains dynamic description of service interface.

The first section describes generic message exchange patterns applicable in the service interface.

The second section describes service orchestration related to operational procedures.

E.2.1.1 Message Exchange Patterns

HTTP-based communication has been used when describing the squences, hence the default return (if no other information) is the HTTP Code from the HTTP protocol. The sequences also do not describe the authentication and authorization procedures.

E.2.1.1.1 ONE_WAY

Fire-and-forget, sends data to consumer without expecting something back, other than technical response.

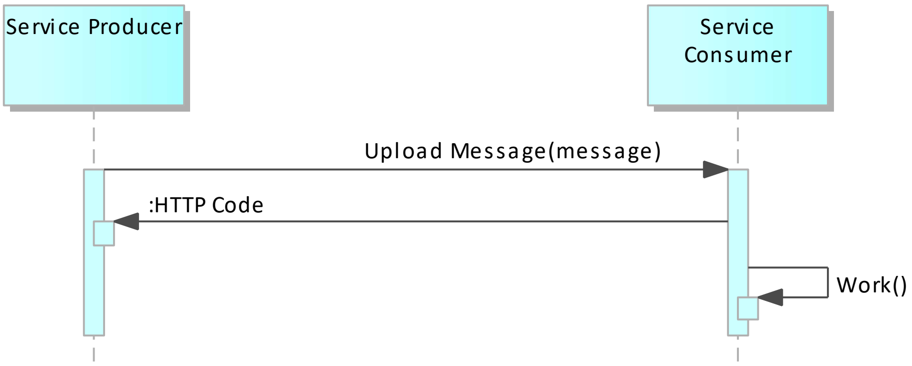

Figure E.1 - Message Exchange Pattern - ONE_WAY

E.2.1.1.2 REQUEST_CALLBACK

Sent message are expected to result in message back, such as optimized route.

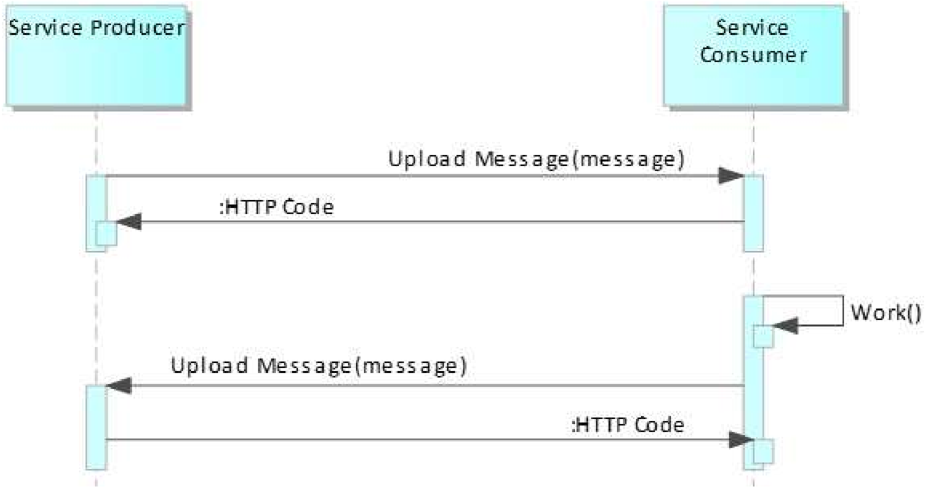

Figure E.2 - Message Exchange Pattern - REQUEST_CALLBACK

E.2.1.1.3 REQUEST_RESPONSE

Consumer requests information from producer, and if authorized, information is sent back synchronously in the response.

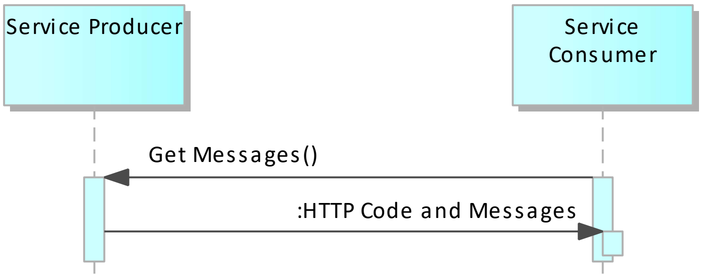

Figure E.3 - Message Exchange Pattern - REQUEST_RESPONSE

E.2.1.1.4 PUBLISH_SUBSCRIBE (Producer nominates)

The parent on producer side nominates a consumer and sends messages until either producer or consumer ends the subscription.

All uploaded message can be combined with acknowledgement when sent message has been forwarded to parent.

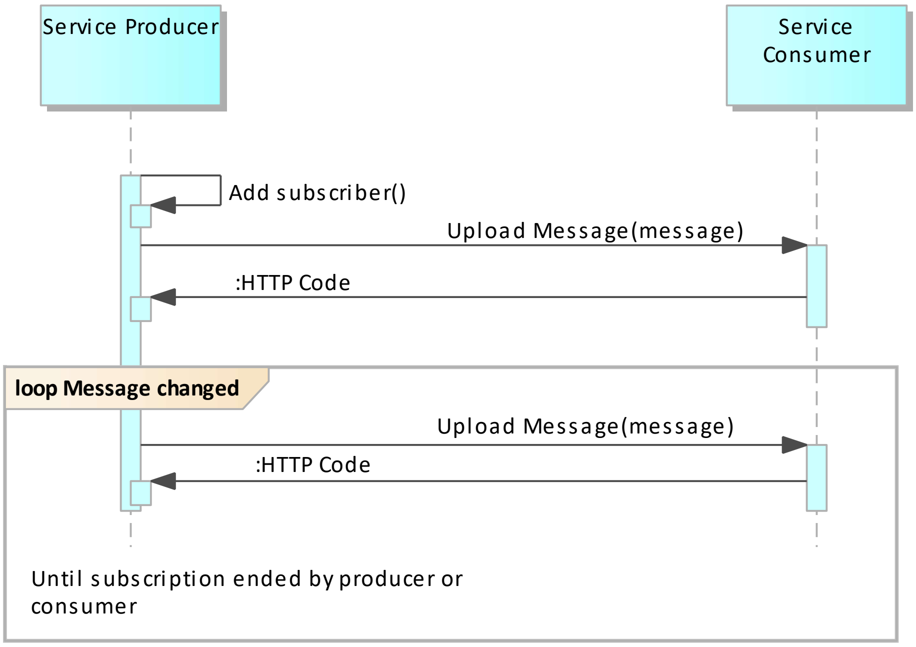

Figure E.4 - Message Exchange Pattern - PUBLISH_SUBSCRIBE (Producer nominates)

E.2.1.1.5 PUBLISH_SUBSCRIBE (Consumer requests)

Consumer request subscription on producer. The producer either rejects or accepts the subscription request. If accepted, message(s) in given subscription are sent, and then all updated messages are sent to consumer until either producer or consumer ends the subscription.

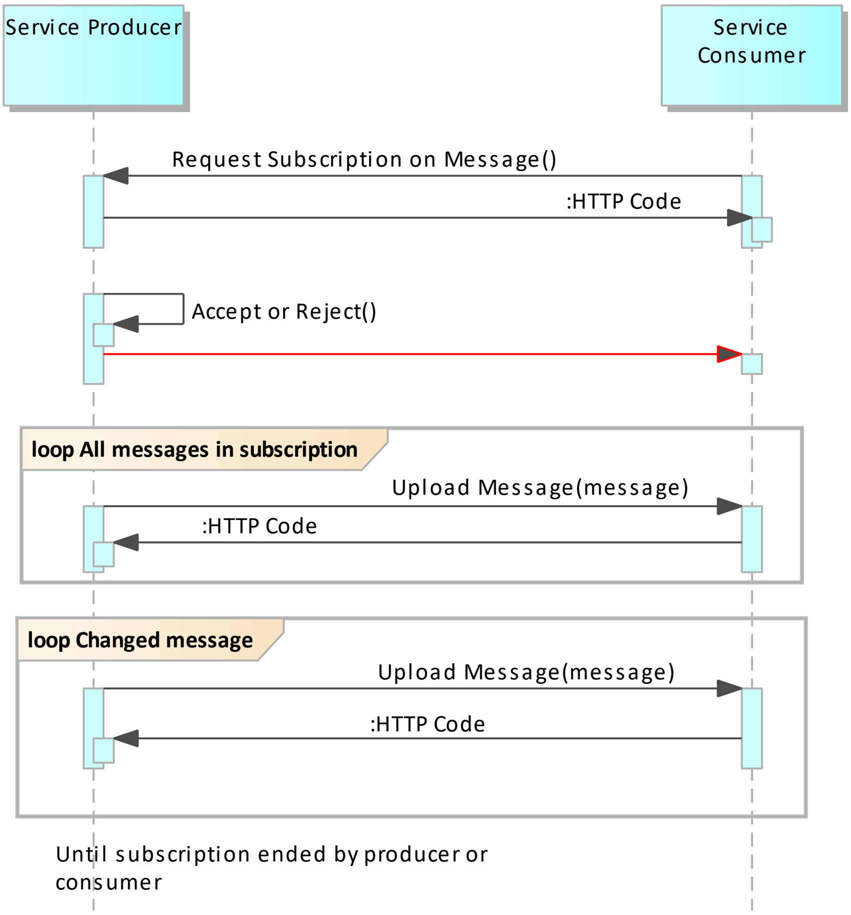

Figure E.5 - Message Exchange Pattern - PUBLISH_SUBSCRIBE (Consumer request)

E.2.1.2 Service Orchestration

To be elaborated in work group.

This clause describe variants of exchange patterns based on operational usage. For further information regarding use cases and profiles, see Annex B.

E.2.1.2.1 PUBLISH_SUBSCRIBE using information as request

Same procedure as REQUEST_CALLBACK wilt multiple callback, but the pattern also generates an automatic subscription on producer and all changes in messages are distributed to consumer until either producer or consumer ends the subscription.

All uploaded message can be combined with acknowledgement when sent message has been forwarded to parent.

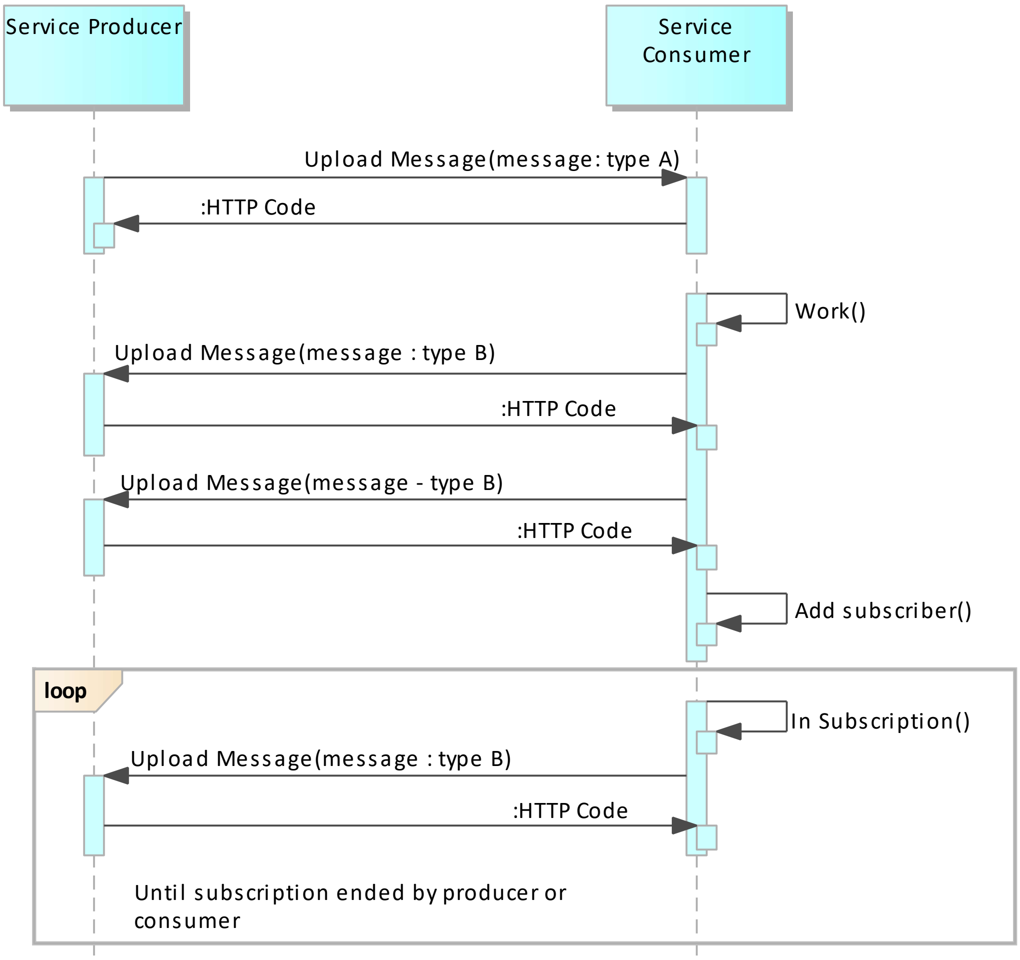
Service Producer
Service Consumer
Upload Message(message: type A)
:HTTP Code
Work()
Upload Message(message : type B)
:HTTP Code
Upload Message(message - type B)
:HTTP Code
Add subscriber()
loop
In Subscription()
Upload Message(message : type B)
:HTTP Code
Until subscription ended by producer or consumer

Figure E.6 - PUBLISH_SUBSCRIBE using information as request

Annex F
(informative)

Deployment examples

F.1 Purpose

This annex contains the examples of SECOM deployment.

F.2 On ship

F.3 On shore

Annex G
(informative)

Use of white list

G.1 Purpose

This annex contains the tbd.

G.2 White list