**ialA** **Technical service**

SD0002

Service Design for VTS – Vessel Route Exchange Service using SECOM



Edition 1.1

December 2025

**urn.mrn.iala:techsvc:sd0002:ed1.1**

|  |  |  |
| --- | --- | --- |
| **Date** | **Details** | **Approval** |
| Feb 2025 | Edition 1.0.0 First version of Service Specification Annex to accompany Guideline G1128 | VTS |
| April 2025 | Edition 1.0.1 Editorial changes applied to align the content with the official document format. | Secretariat |
| December 2025 | Changed to reflect changes in service specification ed1.1. |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

[1 introduction 5](#_Toc196558343)

[1.1 Purpose 5](#_Toc196558344)

[1.2 Intended readership 5](#_Toc196558345)

[1.3 Input from other sources 5](#_Toc196558346)

[2 Service design template Identification 6](#_Toc196558347)

[3 TECHNOLOGY INTRODUCTION 7](#_Toc196558348)

[3.1 General 7](#_Toc196558349)

[3.2 Service Technology and Service Transportation protocol 7](#_Toc196558350)

[3.3 Security 7](#_Toc196558351)

[3.3.1 Communication Channel Security 7](#_Toc196558352)

[3.3.2 Data Protection 7](#_Toc196558353)

[3.3.3 Data Signature 7](#_Toc196558354)

[3.3.4 Data Encryption 8](#_Toc196558355)

[4 Service OVERVIEW 9](#_Toc196558356)

[4.1 General 9](#_Toc196558357)

[4.2 Service interfaces 9](#_Toc196558358)

[4.3 Service Discovery 12](#_Toc196558359)

[5 PHYSICAL DATA MODEL 13](#_Toc196558360)

[5.1 GML:ID of the root element 13](#_Toc196558361)

[6 Service Interface Behaviour 14](#_Toc196558362)

[6.1 Common Information 14](#_Toc196558363)

[6.1.1 Common parameter values 14](#_Toc196558364)

[6.1.2 Error codes 14](#_Toc196558365)

[6.2 Applicable SECOM interfaces 14](#_Toc196558366)

[6.2.1 Upload 15](#_Toc196558367)

[6.2.2 Upload link 15](#_Toc196558368)

[6.2.3 Acknowledgement 15](#_Toc196558369)

[6.2.4 Get 16](#_Toc196558370)

[6.2.5 Get summary 16](#_Toc196558371)

[6.2.6 Get by link 16](#_Toc196558372)

[6.2.7 Access 16](#_Toc196558373)

[6.2.8 Access notification 17](#_Toc196558374)

[6.2.9 Subscription 17](#_Toc196558375)

[6.2.10 Remove subscription 18](#_Toc196558376)

[6.2.11 Subscription notification 18](#_Toc196558377)

[6.2.12 Capability, Ping, EncryptionKey and PublicKey 18](#_Toc196558378)

[7 Service Dynamic Behaviour 19](#_Toc196558379)

[7.1 Generic sequence for signatures 19](#_Toc196558380)

[7.2 Dynamic behaviour of the service interface 19](#_Toc196558381)

[7.2.1 Upload, subscribe and acknowledgement interfaces 20](#_Toc196558382)

[7.2.2 UploadLink and GetByLink interfaces 21](#_Toc196558383)

[7.2.3 Get and RequestAaccess interface from shore to vessel 21](#_Toc196558384)

[7.2.4 Get from vessel to shore 22](#_Toc196558385)

[7.2.5 Remove Subscription interface 23](#_Toc196558386)

[8 Definitions 24](#_Toc196558387)

[9 Abbreviations 25](#_Toc196558388)

[10 References 26](#_Toc196558389)

[11 Appendix 1 – Service Design Template OPENAPI (SWAGGER) in JSON 27](#_Toc196558390)

[12 Appendix 2 – Guidance on Implementation 28](#_Toc196558391)

**List of Tables**

[Table 1 Service Design Identification 5](#_Toc196553406)

[Table 2 Service interfaces 9](#_Toc196553407)

**List of Figures**

[Figure 1 GML:ID with UUID 13](#_Toc196553416)

[Figure 2 Dynamic behaviour of Upload, Acknowledgement and Subscription interfaces 20](#_Toc196553417)

[Figure 3 Dynamic behaviour of GetByLink and UploadLink interfaces 21](#_Toc196553418)

[Figure 4 Dynamic behaviour of Get and Request Access interfaces 21](#_Toc196553419)

[Figure 5 Dynamic behaviour of Get interface from vessel to shore 22](#_Toc196553420)

[Figure 6 Dynamic behaviuor of route cancellation / termination and removal of subscriptions 23](#_Toc196553421)

# introduction

This document specifies the design of the route exchange service according to the technical service specification [2] using SECOM as the transport mechanism for route exchange. This design defines the implementation of the route exchange service that fulfils the requirements defined by IMO [3].

This document was produced as part of the work of IALA VTS committee task group on development of technical service specifications for VTS. The document is structured according to the IALA Guideline G1128 The Specification of e-Navigation Technical Services [1] and is based on the SECOM template found therein.

## Purpose

The purpose of this document is to define a service design using SECOM for the route exchange service defined in the technical service specification. This is done by using SECOM and the S-421 XML representation as defined in the specification. It defines the actual way the interactions and logical interfaces defined in the specification are to be implemented. It also describes how the requirements for authentication etc are to be implemented.

It describes the actual requirements for both ship and shore side systems in order to support route exchange as defined in the specification.

*Note that this design will undergo changes between versions 1.0 and 2.0 as the SECOM and S-421 standard should be updated during that transition period and the design will be changed to reflect the latest changes. Some of the anticipated changes to SECOM and S-421 have already been included in this design and as such this design may not be fully compatible with SECOM 1.0 and S-421 1.0.*

## Intended readership

This Service Design is intended to be read by service architects, system engineers and developers in charge of designing and developing an instance of the route exchange service.

## Input from other sources

This service design follows the requirements, use cases and dynamic behaviour outlined in [2]. The interfaces, parameters and logic of the API is compliant with IEC 63173-2 SECOM [4] and borrows much of its content from there.

“The author thanks the International Electrotechnical Commission (IEC) for permission to reproduce Information from its International Standards. All such extracts are copyright of IEC, Geneva, Switzerland. All rights reserved. Further information on the IEC is available from www.iec.ch. IEC has no responsibility for the placement and context in which the extracts and contents are reproduced by the author, nor is IEC in any way responsible for the other content or accuracy therein.”[[1]](#footnote-2).

# Service design template Identification

This section provides a unique identification of the service and describes where the service is in the engineering lifecycle.

Table Service Design Identification

|  |  |
| --- | --- |
| **Name** | VTS – Vessel Route Exchange using SECOM Technical Service Design |
| **Implements** | VTS – Vessel Route Exchange Technical Service Specification 1.1  urn:mrn:iala:techsvc:ss:vts-res:1.1 |
| **ID[[2]](#footnote-3)** | urn:mrn:iala:techsvc:design:vts-res-secom:1.1 |
| **Version** | 1.1 |
| **Description** | The VTS – Vessel Route Exchange Service using SECOM design describes how to implement a Route Exchange Service using SECOM as the transport mechanism between ship and VTS in the exchange of route plans. |
| **Keywords** | REST, SECOM, VTS, MS1, Route Exchange, Ship Traffic Management, S-421 |
| **Architect(s)** | Ramin Miraftabi |
| **Status** | Provisional |

# TECHNOLOGY INTRODUCTION

## General

This design template realizes the service specification [2] using SECOM as defined in IEC 63173-2.

The services conforming to this design must be implemented with REST APIs using HTTPS with TLS protection to encrypt all communication in transit.

## Service Technology and Service Transportation protocol

*Reference: IEC 63173-2 SECOM v1.0.0 Clause 5.3 Service Technology*

The technology (architectural style) chosen is REST (REpresentational State Transfer) upon HTTP/1.1 (RFC 7231).

## Security

### Communication Channel Security

*Reference: IEC 63173-2 SECOM v1.0.0 Clause 6 SECOM communication channel security*

The channel security between the SECOM REST service and a consumer are

* HTTP/1.1 according to RFC-7231
* HTTPS over TLS according to RFC-2818

Valid versions of TLS for this version of service design template are

* TLS version 1.2 and 1.3 (RFC-8446)

X.509 Certificates are used in the TLS according to RFC 5280 and RFC 2459.

Certificates shall be verified with OCSP and/or CRL methods.

### Data Protection

*Reference: IEC 63173-2 SECOM v1.0.0 Clause 4.3.4 SECOM data protection*

*Reference: IHO S-100 ed5.2.0 Part 15 Data Protection Scheme*

The data is mandatory to be signed by the sender to enable data authentication and integrity check by the receiver.

The data can optionally be encrypted by the sender, and the sender is responsible for exchanging the encryption key with the receiver.

While sending standalone S-421 documents without the additional metadata of datasets or exchange sets is preferred, if datasets or exchange sets are used the data (one or more data files) can optionally be packaged and compressed before being signed and sent.

### Data Signature

*Reference: IEC 63173-2 SECOM v1.0.0 Clause 7.3 Data authentication and signing*

*Reference: IHO S-100 ed5.2.0 Part 15-8 Data Authentication*

*Reference: NIST Digital Signature Standard (DSS–FIPS Publication 186)*

The algorithm for signing data must be ECDSA-384-SHA2.

The signature is transported in HEX.

### Data Encryption

*Reference: IEC 63173-2 SECOM v1.0.0 Clause 7.4 Data encryption*

*Reference: IHO S-100 ed5.2.0 Part 15-6 Data Encryption*

The encryption algorithm for encryption is AES and CBC mode.

The symmetric encryption key can be exchanged by different means, including using the SECOM REST API to exchange the encryption key.

# Service OVERVIEW

## General

The design uses SECOM defined APIs. As such, it is important to understand that both the route exchange service and its consumers must be able to function as client and server as understood in the traditional HTTP world. As such from here on when the term service is used, it applies to the route exchange service and the consumer is the ship system that is communicating VTS.

This design does not concern itself with how the service will communicate with the VTS system. The instances of this design may be developed as components directly integrated with the VTS system or as independent microservices that communicate with the VTS system via different integration mechanisms e.g., APIs or by emitting and consuming events.

## Service interfaces

*Reference: IEC 63173-2 SECOM v1.0.0 Clause 5.7 SECOM service interface definitions*

SECOM does not require that all the interfaces defined in the standard must be implemented. Thus, for the purposes of this service and its consumers, only the following interfaces are required:

Table Service interfaces

| **Interface** | **SECOM Reference** | **Comment** |
| --- | --- | --- |
| Upload | IEC 63173-2 SECOM v1.0.0 Clause 5.7.2 service interface – Upload | This interface is called when the client uploads (pushes) a route to the server. The sender decides the protection of the route. This interface is provided by both the consumer and service. |
| Upload Link | IEC 63173-2 SECOM v1.0.0 Clause 5.7.3 service interface – Upload Link | This interface is called when the client uploads (pushes) a reference pointer to a route that is larger than what the server allows via HTTP POST. The data is downloaded using interface Get By Link. This interface is provided by both the consumer and service. |
| Acknowledgement | IEC 63173-2 SECOM v1.0.0 Clause 5.7.4 service interface – Acknowledgement | This interface is called as response to Acknowledgement request in the previous request. This interface is provided by both the consumer and service. |
| Get | IEC 63173-2 SECOM v1.0.0 Clause 5.7.5 service interface – Get | This interface is called when the client gets (pulls) data from the service. This interface must be provided by the consumer, but supporting get-requests from the consumer is optional for the service. |
| Get Summary | IEC 63173-2 SECOM v1.0.0 Clause 5.7.6 service interface – Get Summary | This interface is called when the client gets a summary of available data from the service. The data is retrieved (pulled) using the interface Get. |
| Get By Link | IEC 63173-2 SECOM v1.0.0 Clause 5.7.7 service interface – Get By Link | This interface is called when the client downloads (pulls) a large route by reference given from interface Upload Link. This interface is provided by both the consumer and service. |
| Access | IEC 63173-2 SECOM v1.0.0 Clause 5.7.8 service interface – Access | This interface is called when the client asks for access to data from the service. Response is given by callback to Access Notification. This interface is provided by the consumer. Support for access is optional for the consumer. |
| Access Notification | IEC 63173-2 SECOM v1.0.0 Clause 5.7.9 service interface – Access Notification | This interface is called as response to interface Access. This interface is provided by the service. |

|  |  |  |
| --- | --- | --- |
| **Interface** | **SECOM Reference** | **Comment** |
| Subscription | IEC 63173-2 SECOM v1.0.0 Clause 5.7.10 service interface – Subscription | This interface is called when the client or server initiates subscription on data from the service. Response is given with interface Upload and Subscription Notification. This interface is provided by both the consumer and service. |
| Remove Subscription | IEC 63173-2 SECOM v1.0.0 Clause 5.7.11 service interface – Remove Subscription | This interface is called when the client or server removes subscription. Response is given with interface Subscription Notification. This interface is provided by both the consumer and service. |
| Subscription Notification | IEC 63173-2 SECOM v1.0.0 Clause 5.7.12 service interface – Subscription Notification | This interface is called as response from Subscription or Remove Subscription. This interface is provided by both the consumer and service. |
| Capability | IEC 63173-2 SECOM v1.0.0 Clause 5.7.13 service interface – Capability | This interface is called when the client asks for the service capabilities. This interface is provided by both the consumer and service. |
| Ping | IEC 63173-2 SECOM v1.0.0 Clause 5.7.14 service interface – Ping | This interface is called when the client checks the availability of the service. This interface is provided by both the consumer and service. |
| EncryptionKey | IEC 63173-2 SECOM v1.0.0 Clause 5.7.15 service interface – EncryptionKey | This interface is called when sending (pushing) encryption key to a receiver. This interface is provided the service and is optional for the consumer. |
| PublicKey | IEC 63173-2 SECOM v1.0.0 Clause 5.7.16 service interface – PublicKey | This interface is called when the client gets (pulls) the public certificate(s) from the service. This interface is provided by the service and is optional for the consumer. |

All unused interfaces of SECOM should be implemented to return HTTP 501 as specified in [4].

There are three components that are of interest from the perspective of the service design:

* The service has a SECOM-component which supports the SECOM REST APIs defined in the table above. All other components of the service are left to the decisions of the implementing party.
* The vessel has a SECOM-component which will accepts the incoming connections from the service and store all messages until delivered to the vessel. This interface is typically on a shoreside server as it must be always available and at a static address.
* The vessel has an implementation of a SECOM client which allows it to make direct SECOM calls to the service without having to proxy all calls via the SECOM-component on shore.

In this service design we will not define the communication between the service and VTS system or between the vessel and the vessel’s shoreside SECOM interface. These are specific for each implementation and depend on the VTS system and vessel’s system.

## Service Discovery

Services implemented according to this design must submit their instance description to a valid service registry that follows the Maritime Service Registry definition [11].

An XML template for the instance description is provided as an annex to this design ANNEX B.

# PHYSICAL DATA MODEL

The data model of the service is a combination of JSON (SECOM calls) and XML (the S-421 payload). The SECOM JSON is defined in [4] section 5.

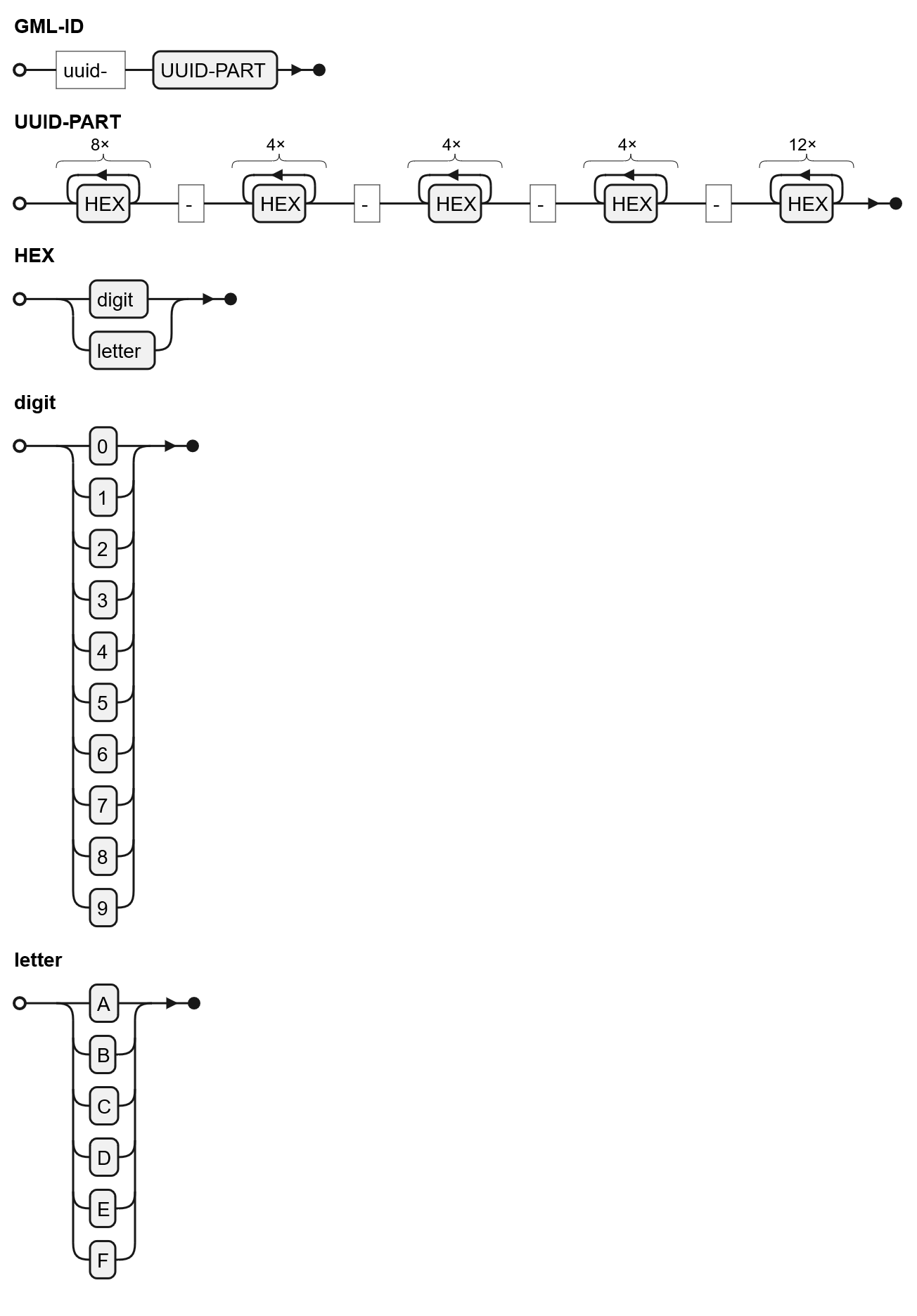
The S-421 data that is used by the service must be a valid XML document according to the S-421 schema[9] and follow the requirements defined in the service specification [2]. The service specification has extended requirements on what constitutes a valid route in the context of this service and the different use cases it is designed to support.

While the S-421 schema supports S-100 datasets and exchange sets, the more minimal version of the document without dataset information is preferred.

Implementing services should send the route without dataset or exchange set information.

Services must support sending datasets or exchange sets if requested by client.

## GML:ID of the root element

The root element of the XML document is required to have a gml:id attribute. While the gml:id is not required to be globally unique in S-100 or GML for the purpose of route exchange every effort should be made to ensure that the gml:id is globally unique.

If the system generating or storing the route does not have a method of defining a gml:id which can reasonably be expected to be globally unique using a UUID as a part of the gml:id is recommended.

Because the gml:id is an xml:id which must begin with a letter and a UUID may begin with a letter or number, the following requirement is placed: The gml:id of the dataset element of the route must follow the format: uuid-[uuid] where uuid is the UUID also used in the transactionIdentifier of the SECOM calls.

A gml:id should not contain a colon character ( : ), thus the urn:uuid:[uuid] format which would conform with having URNs in the gml:id cannot be used. The requirement to avoid a colon applies also to other formats of the gml:id.

Figure GML:ID with UUID

# Service Interface Behaviour

As defined by SECOM, all communication between the components of the service that are in the scope of this design document is done via REST calls.

Additional information on the interfaces and parameters is available in the attached OpenAPI template of service in ANNEX A.

## Common Information

For all parameters that are not set they should be omitted or can passed with a null value.

### Common parameter values

For all interfaces, the following parameter values must be used when applicable:

* containerType – should always be “NONE” (i.e. 2)
* dataProductType – must always be “S-421”
* exchangeMetadata
* protectionScheme – must always be “SECOM”
* callbackEndpoint – The endpoint of the consumer’s SECOM service that accepts the interface calls. The endpoint must be defined in a way that appending the SECOM standard REST endpoints (e.g. /v2/subscription) produces a valid URL. If the callbackEndpoint is not a part of the subscription, the service must search for the vessel’s SECOM service from a service registry.
* transactionIdentifier (transactionId) must be a UUID. It must not be connected to the UUID of the route in the gml:id. It must be unique for each transaction.

### Error codes

If incoming route does not pass schema validation, the SECOM\_ResponseCode must be 3.

If incoming route does not pass business rule validation, the SECOM\_ResponseCode must be 0.

If certificate provided is invalid, the SECOM\_ResponseCode must be 2.

If the signature verification fails, the SECOM\_ResponseCode must be 1.

In all of these cases, the response returned to the client must have the HTTP status 400.

## Applicable SECOM interfaces

The Capability and Ping interfaces are not discussed in this design as they follow the requirements defined in SECOM [4] and need no further elaboration here.

All interface descriptions below only have the information necessary for the business logic of this service. All definitions and descriptions that are directly derived from the SECOM standard are left to be read from the SECOM standard and the API documentation template in ANNEX A.

All SECOM communication is done via REST APIs with JSON as the data format for the SECOM data and XML for the S-421 data.

For all operations only those parameters that have predefined values or their usage needs to be explained are described below. Otherwise refer to SECOM [4].

### Upload

The Upload interface must be available both on the service as well as the SECOM interface of the vessel.

For all parameters, follow the requirements specified in the general section and:

* fromSubscription –
* must always be true when sent from service to vessel as service never uploads without subscription;
* must always be false when route is initially sent from ship system to shore;
* must always be true when ship is sending updates to route following the subscription for updates from shore.
* ackRequest – By default, set to 3 (both delivered and opened) unless:
* ship system has not requested acknowledgements of route opening when sending updates as part of subscription from shore. It is set to 1 (delivered) in this case.
* Route is moved to monitoring. Set to 1 in this case.
* data – S-421 route that follows the requirements specified in [9] with the error codes specified above.

The return value of the upload interface must follow the SECOM standard. The message is not expected and should only be returned in the case of an error to provide additional information.

If the uploaded route is too large the returned HTTP status code must be 413. The route must then be shared using the upload link interface.

The vessel may return HTTP status 403 to any route it receives that it has not requested or has not previously shared with shore.

Shore systems should not return HTTP status 403 (Not authorized) to vessels.

### Upload link

The Upload link interface must be available both on the service as well as the SECOM interface of the vessel.

This is used when the route shared is too large to share via the upload interface and a HTTP status 413 has been received. If the sender knows the route will be too large due to previous interactions, then it may be used directly.

It must not be used as the first method of interaction between consumer and service and any upload link that is received that is not a part of an established route exchange initiated by at least one call to the upload interface must be rejected and HTTP status code 403 must be returned.

For all parameters, follow the requirements specified in the general section and what is specified for upload.

See the Get by link interface definition below for how the retrieval of the data is handled.

### Acknowledgement

There are no specific business rules that must be defined here for the acknowledgment interface that are not covered by the SECOM API specification document.

The requested level of acknowledgment is specified in other parts of this document.

### Get

Support for the Get interface is optional for the consumer and may be available in the service if the reference routes use case has been implemented.

The consumer uses the get interface to retrieve reference routes from a route library.

The service uses the get interface to retrieve the current route from the ship once the route has been initially shared by the ship (use case 15).

When used by the consumer use the parameters specified in the general section and:

* dataReference – used when the id of the reference route is known, and the latest version of the route is being fetched. Note, when used this must be the UUID part of the gml:id.
* geometry – used when searching for reference routes that intersect a geometry
* unlocode – used to search for reference routes that intersect with the UN/LOCODE, generally either starting or ending at location. The UN/LOCODE search can also be done with just the first 2 characters of the UN/LOCODE to search for all UN/LOCODEs in that country.
* validFrom and validTo – used to specify the time range in which the reference route must be valid. The times must include a time zone and should be in UTC.

Return values must follow what is specified in SECOM.

When used by the service use the parameters specified in the general section and:

* dataReference – must be the UUID part of the gml:id of the route..

When the service is retrieving an update to the route, no other parameters are allowed and if passed the consumer should respond with HTTP status code 400.

If the route is requested by service without it being previously shared (use case 10 in specification) then the consumer must respond with HTTP status code 403.

### Get summary

The Get summary interface may be available on the service if the reference route library functionality has been implemented.

The interface is used instead of Get to get a list of available routes if the consumer so wishes.

All parameters and their usage should follow what was previously described for the get interface.

### Get by link

The Get by link interface must be available both on the service as well as the SECOM interface of the vessel.

This is used when the route shared is too large to share via the upload interface and the route has been shared via the upload link interface.

### Access

The Access interface can be supported by the consumer.

The interface is used when VTS is requesting the route from the vessel without the route being shared previously according to use case 10 in the specification. Calling this interface requires that the vessel is discoverable in a maritime service registry.

The vessel is not required to share its route and may respond with HTTP status code 403 if not allowed.

For all parameters, follow the requirements specified in the general section and:

* dataReference must not be set
* reason – must be provided and contain rationale for the request
* reasonEnum – must be set to 0 (required by authority)

If the vessel’s SECOM interface is able to determine automatically that access is not granted (e.g. based on the identity of the requester) then HTTP status code 403 may be returned and the mariner aboard the vessel need not be notified of the request.

### Access notification

The Access interface must be supported by the service.

The interface is used when the mariner onboard the vessel has been notified of the request by VTS to receive the route.

The following use of the parameters is required:

* transactionIdentifier – if access is granted follows the rules described in the general section, otherwise is a unique UUID.
* decision and decisionReason – according to SECOM specification.

### Subscription

The subscription interface must be implemented in both the service and consumer.

The following parameters of the SubscriptionRequestObject are expected to be passed as a part of the request:

* dataReference –the gml:id dataset element in the route that was shared.
* productVersion – must be left unset. The service specification defines the allowed version of S-421 that the service and consumer must support.
* geometry – must not be provided by the consumer, may be provided by the service. Consumer may ignore the geometry in a subscription request.
* unlocode – must not be provided
* subscriptionPeriodStart – may be provided, if not provided or in the past, the recipient will default to time of reception of call to interface
* subscriptionPeriodEnd – must not be provided. Both the consumer and the service must end subscriptions.

The returned SubscriptionResponseObject contains the following:

* message – optional. Consumers may or may not support this.
* subscriptionIdentifier – must have an UUID that can be used to remove the subscription once communication is no longer necessary.

The service must use the MRN of the vessel retrieved from the certificate submitted in the SECOM request headers to identify the vessel for which the subscription is created. The vessel must not receive routes intended for other vessels because of this subscription.

The vessel must also ensure that it does not send route updates to those shore systems that have not subscribed to updates or send route updates intended for another shore system to the incorrect shore system.

If the vessel creates an open subscription (e.g. regular transit along a route) then the dataReference parameter must be vessel MRN encoded as a UUIDv5 with the MRN as the name and URL as the namespace.

### Remove subscription

The remove subscription interface must be implemented in both the service and consumer.

The removeSubscriptionObject must have the subscriptionIdentifier returned when the subscription was created.

The consumer should call removeSubscription when

1. The route changes and no longer passes through a VTS area and the subscription for that VTS area can be removed.
2. The route has been completed and no further sharing of the route is necessary.
3. The route is cancelled.

The above causes assume that the ship system creates a new subscription for each route that is shared. If the ship systems create an open-ended subscription for route sharing (e.g. vessel is on a regular route and subscriptions are created for each VTS area) then the subscription can be kept open as long as necessary.

The service must call removeSubscription when the consumer ends its subscription as the sharing of the route is no longer needed and no more updates to the route from the consumer are to be expected.

### Subscription notification

The subscription interface must be implemented in both the service and consumer.

The service is not required to call the subscriptionNotification of the consumer unless the creation or removal of the subscription requires asynchronous action or is the result of a maintenance operation.

The consumer should call the subscriptionNotification of the service in most cases as the recipient of the request from the service is typically not the ship system that is required to know about the subscription for the sending of route updates. If the subscription and removeSubscription are synchronous processes when called from VTS to ship system, subscriptionNotification is not required.

Use of the interface is according to the SECOM specification.

### Capability, Ping, EncryptionKey and PublicKey

The interfaces Capability, Ping, EncryptionKey and PublicKey are not discussed here as they follow the requirements defined in IEC 63173-2 SECOM and need no further elaboration here.

# Service Dynamic Behaviour

In the following diagrams and descriptions, the communication between the vessel’s onboard systems and its SECOM service is out of scope for the purposes of this design.

Similarly, the communication and interfaces between the service and the VTS system is out of scope of this design.

## Generic sequence for signatures

This sequence assumes the following actors:

Data Provider → Data Sender → Data Receiver → Data Consumer

The sequence describes the data from its source through transport using SECOM to the end-user.

Provider preparation:

* The data provider must have a valid maritime identity according to G1183 [10]
* The data provider creates a private-public key pair
* The data provider sends the public key to the MIR to get it signed and connected to a specific identity/identifier in the MIR (Certificate Signing Request)
* The data sender must have a valid maritime identity according to G1183
* The data sender creates a private-public key pair
* The data sender sends the public key to the MIR to get it signed and connected to a specific identity/identifier in the MIR (Certificate Signing Request)

Data Provider and Data Sender:

* The data provider selects identifier and certificate to use for the data signing
* The data provider creates a signature for the data
* The data provider transfers the data with its signature to the data sender.
* The data sender protects the data transferring by the selected communication mechanism, e.g. SECOM. SECOM Upload wraps the data, the signature and metadata into an envelope. The envelope is signed by the data sender.
* The data sends contacts the data receiver and transfers the data

Data Receiver and Data Consumer:

* The data receiver receives the transferred object with envelope. The envelope signature is verified.
* The data receiver forwards the data and data signature to the data consumer
* The data consumer verifies the data signature

## Dynamic behaviour of the service interface

The following sections contain sequence diagrams for the service interfaces. This section expands upon the dynamic behavior described in the service specification.

In the following sequence diagrams the “ship SECOM interface” may be onboard, onshore or a mixture thereof. This design does not place constraints on the implementation of the ship systems from a deployment perspective.

The dynamic behavior of ping, capability, encryptionKey and publicKey are not described here to avoid unnecessary duplication.

### Upload, subscribe and acknowledgement interfaces

We will begin by looking at the sequence in the service specification describing use case 1 and 2 in more detail by using SECOM with the subscription required by use case 3 present in the diagram as well.

The “routeId” in the following diagram is the UUID part of the gml:id described in 5.1.

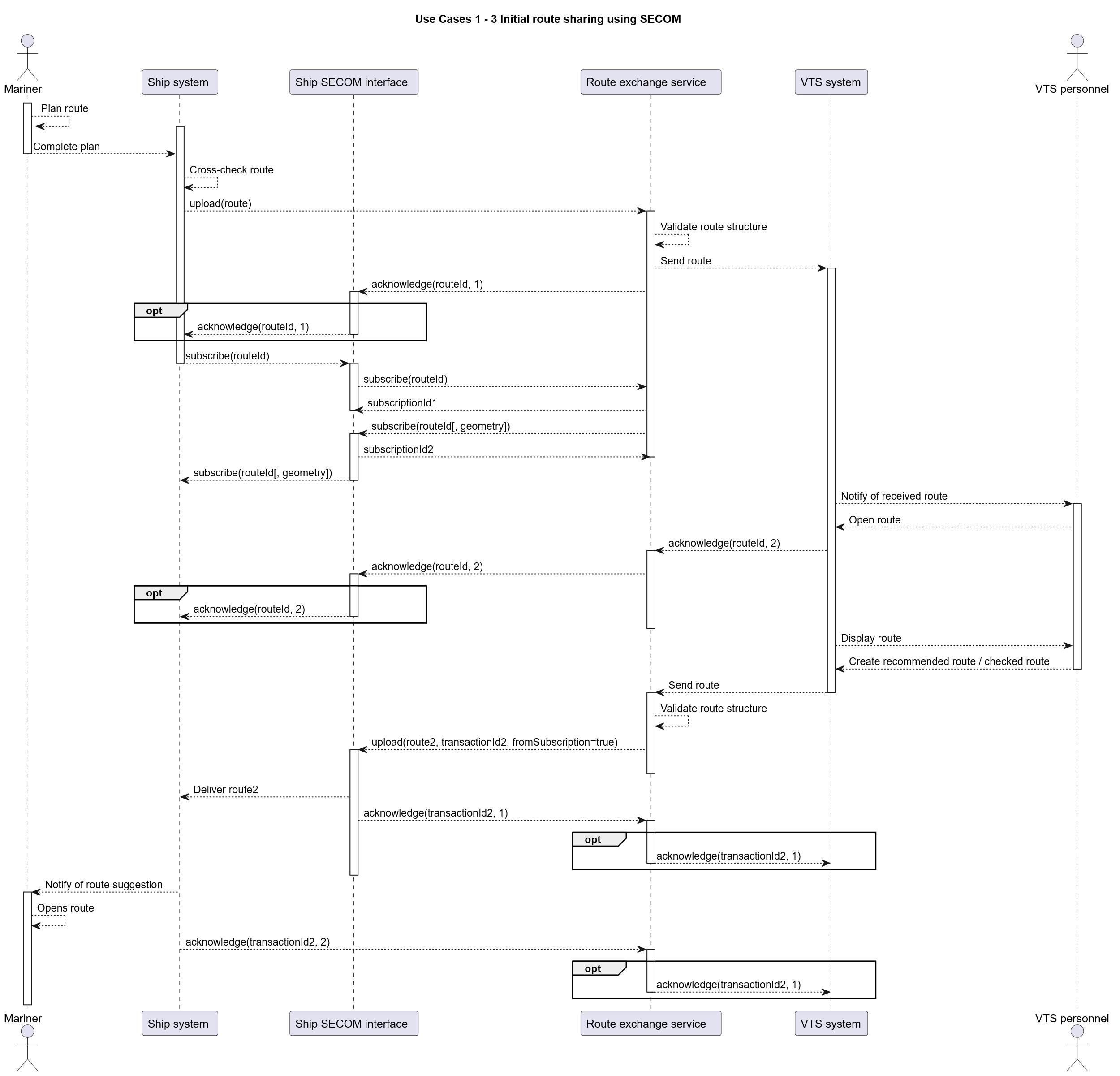


Figure Dynamic behaviour of Upload, Acknowledgement and Subscription interfaces

As previously defined, the transactionIdentifier of the initial sharing of the route is the same as the UUID part of the gml:id of the route.

The diagram above describes how the upload, acknowledge and subscription interfaces behave. Any subsequent uploads from either the vessel or service behaves in a similar way.

### UploadLink and GetByLink interfaces

If the content of the upload is too large to be processes and HTTP status 413 is returned, the route must be transferred by using a combination of the Upload link and Get by link interfaces. The diagram below describes this with a client and server where both the vessel SECOM interface and the service may act in either role.

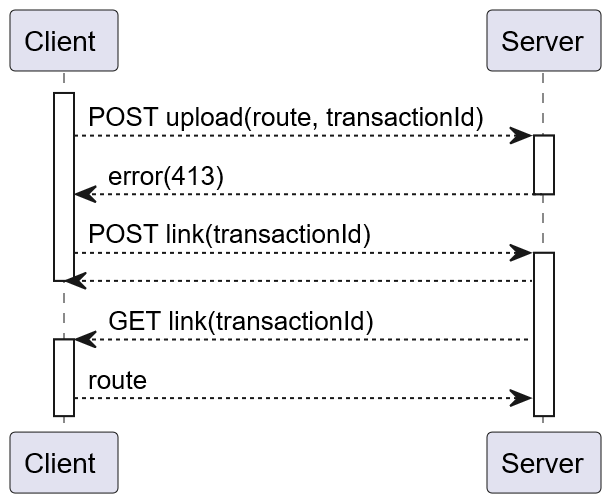


Figure Dynamic behaviour of GetByLink and UploadLink interfaces

Here the POST from the client tells the server to use the transactionId that was passed to retrieve the route. Then the server uses the GET request with the transactionId provided previously to retrieve the route.

The transactionId throughout should be the same. It must be the same in the calls to the link interfaces from both client and server.

### Get and RequestAaccess interface from shore to vessel

Support for both interfaces is optional for the consumer.

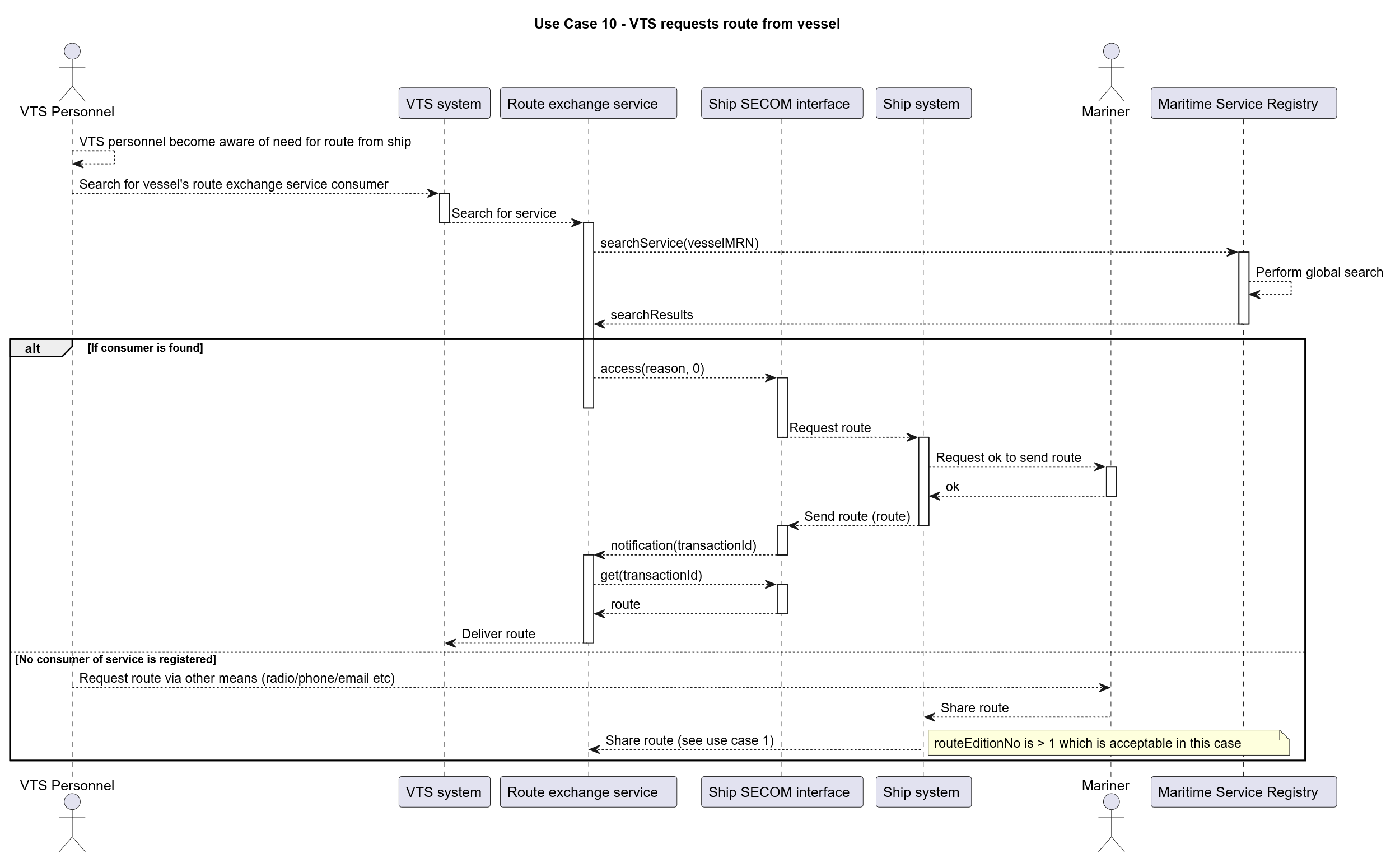


Figure Dynamic behaviour of Get and Request Access interfaces

Here we see the process of how the service can be used to find and request access to the route from the vessel if no route has previously been shared as described in use case 10 in the service specification.

Ship systems can have the necessary functionality for the use of access and notification for future use even if they automatically currently deny all requests. This diagram does not describe the scenario where either the mariner or the ship system automatically denies the request.

In this case, the transactionId of the get interface call by the service must be equal to the transactionId of the notification request.

If VTS is requesting a route update following a previously shared route then follow the data requirements listed in 6.2.4.

### Get from vessel to shore

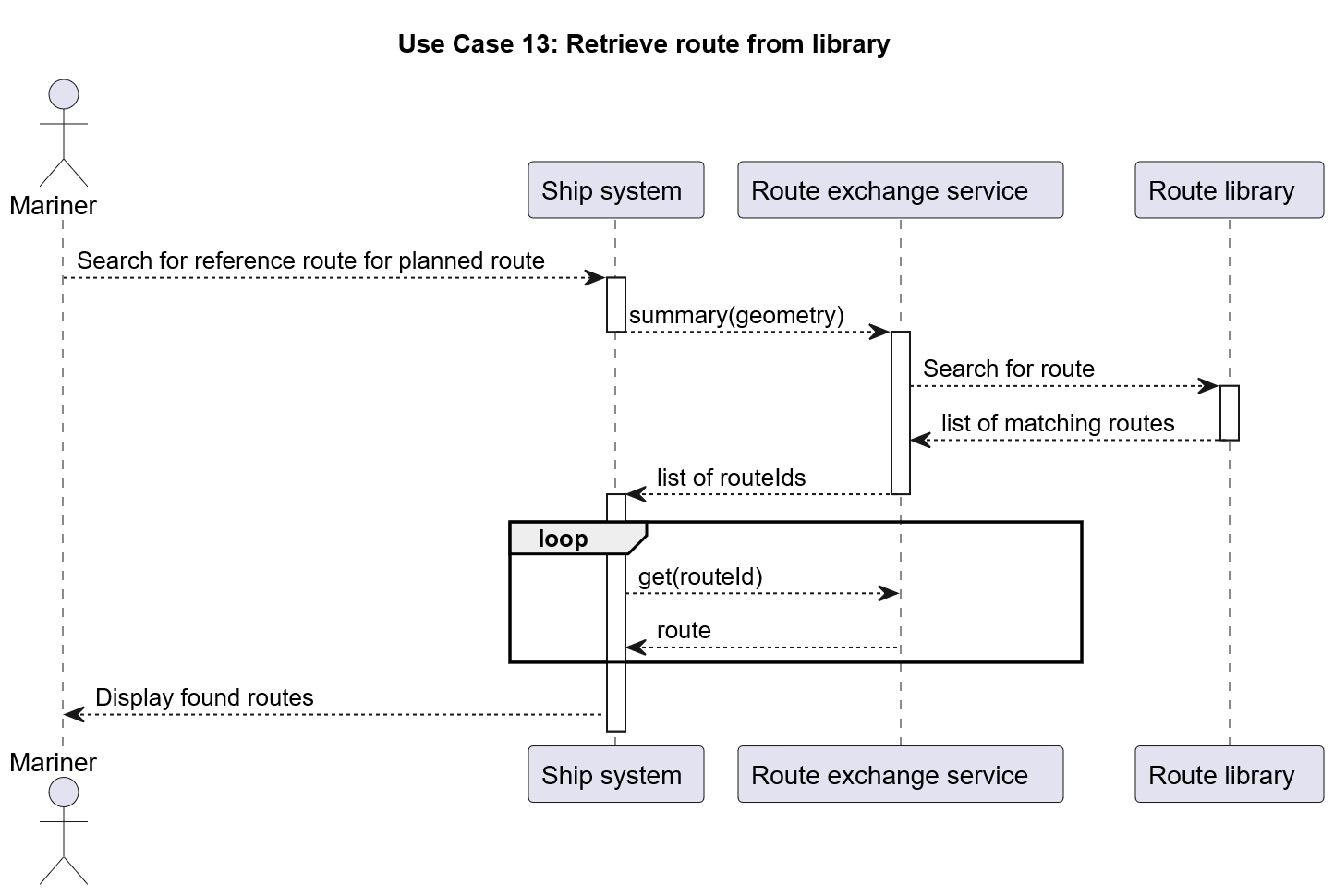


Figure Dynamic behaviour of Get interface from vessel to shore

Here we describe the use of getSummary to search for reference routes in an area to illustrate the connection between getSummary that returns a list of routeIds and the looping over them to retrieve the actual routes using the get interface.

The initial summary request may be replaced with a request to the get interface where the result will directly be the full list of routes without the need to loop through the get requests for each route.

How this is used is up to the ship system but both approaches must be supported.

### Remove Subscription interface

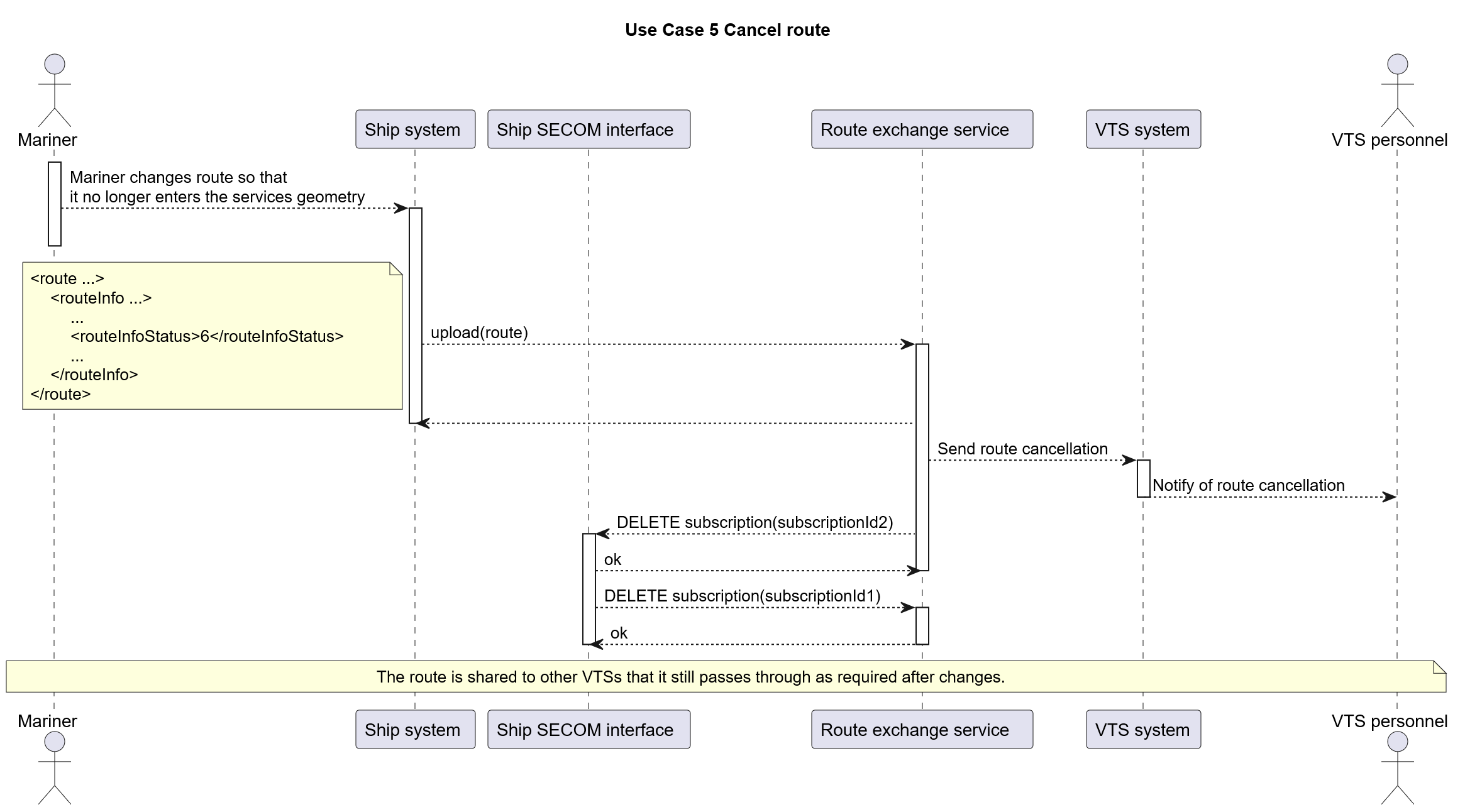


Figure Dynamic behaviuor of route cancellation / termination and removal of subscriptions

Here we see the actual sequence of how the route cancellation from the vessel causes the removal of the subscriptions.

The service must remove the subscription it has when the route is cancelled if the subscription was initiated with a dataReference that is not based on the vessel MRN.

When the Ship SECOM interface receives the subscription removal from the service it must also remove its subscription.

The cancellation of the route also applies to when the route has been completed. The ship system must in that case send information to VTS indicating that the route is completed as the same status indicates that the route is completed or cancelled.

# Definitions

|  |  |
| --- | --- |
| 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. |

# Abbreviations

API Application Programming Interface

MIR Maritime Identity Registry

MRN Maritime Resource Name

MSR Maritime Service Registry

REST REpresentational State Transfer

SECOM Secure Communication

UUID Universally Unique IDentifier

# References

1. IALA. 2022. Guideline 1128 on Specification of e-Navigation Technical Services v1.6
2. Technical Service Specification for VTS – Vessel Route Exchange v1.1
3. IMO 2024. MSC.1/CIRC.1610/REV.1 DESCRIPTIONS OF MARITIME SERVICES IN THE CONTEXT OF E-NAVIGATION https://wwwcdn.imo.org/localresources/en/OurWork/Safety/Documents/enavigation/MSC.1-CIRC.1610%20- %20Initial%20Descriptions%20Of%20Maritime%20ServicesIn%20The%20Context%20Of%20E-Navigation%20(Secretariat)%20(1).pdf
4. IEC 63173-2 SECOM Committee Draft 80/2532/PCC
5. SECOM Service Interface OpenAPI
6. IHO. 2024. Universal Hydrographic Data Model Standard S-100 5.2.0
7. NIST Digital Signature Standard (DSS–FIPS Publication 186)
8. RFC 9110 HTTP Semantics
9. IEC 63173-1 S-421 Route Plan based on S-100 Committee Draft 80/2542/PCC
10. IALA. 2022 Guideline G1183 Provision of Maritime Identities v1.1
11. IALA Guideline GXXXX Maritime Service Registry Technical Specification v1.0

# Appendix 1 – Service Design Template OPENAPI (SWAGGER) in JSON

The OpenAPI JSON file can be found in ref [5].

# Appendix 2 – Guidance on Implementation

More information can be found in ref [4]

1. EC 63173-2 ed. 1.0 “Copyright © 2022 IEC Geneva, Switzerland. www.iec.ch” [↑](#footnote-ref-2)
2. Author to insert actual service name and version number [↑](#footnote-ref-3)