Document No: BS_WP3.1

Title: Functional requirements and specifications for STM-upgrade of navigational systems onboard ships

Date: 2019-10-30
DOCUMENT STATUS

Authors

<table>
<thead>
<tr>
<th>Name</th>
<th>Organisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Björn Andreasson</td>
<td>Swedish Maritime Authority</td>
</tr>
<tr>
<td>Håkan Heurlin</td>
<td>Swedish Maritime Authority</td>
</tr>
<tr>
<td>Anders Berg</td>
<td>Swedish Maritime Authority</td>
</tr>
<tr>
<td>Jouni Lindberg</td>
<td>Swedish Maritime Authority</td>
</tr>
<tr>
<td>Per Löfbom</td>
<td>Swedish Maritime Authority</td>
</tr>
<tr>
<td>Per Setterberg</td>
<td>Swedish Maritime Authority</td>
</tr>
<tr>
<td>Mikael Olofsson</td>
<td>Swedish Maritime Authority</td>
</tr>
<tr>
<td>Cajsa Jersler Fransson</td>
<td>Swedish Maritime Authority</td>
</tr>
</tbody>
</table>

Review

<table>
<thead>
<tr>
<th>Name</th>
<th>Organisation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Approval

<table>
<thead>
<tr>
<th>Name</th>
<th>Organisation</th>
<th>Signature</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Status</th>
<th>Initials</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>2019-03-06</td>
<td>HH</td>
<td></td>
<td>For procurement (EF)</td>
</tr>
<tr>
<td>BS</td>
<td>2019-05-21</td>
<td>CJF</td>
<td></td>
<td>For procurement STM BS</td>
</tr>
</tbody>
</table>

INTERREG PROJECT NO: R103

The sole responsibility of this publication lies with the author. The European Union is not responsible for any use that may be made of the information contained therein.
# Table of contents

1. Summary ................................................................................................................. 4
   1.1 General .................................................................................................................. 4
   1.2 Definitions ............................................................................................................ 4
      1.2.1 Terminology for denoting a requirement ....................................................... 4
      1.2.2 Select, turn-on, set etc. ................................................................................. 4
2. General requirements on the STM ship system ......................................................... 5
3. Functional/Technical Requirements .......................................................................... 6
   3.1 STM Ship system overall description ................................................................... 6
   3.2 Navigation system functionality .......................................................................... 7
   3.3 Communication between STM Module and navigation system ......................... 9
   3.4 STM Module ......................................................................................................... 9
   3.5 Communications between the STM Module and Online access point ............... 12
   3.6 Online access point (Service Instance) ............................................................... 12
4. STM Cyber security requirements .............................................................................. 13
   4.1 STM Cyber security requirements ....................................................................... 13
5. Non-Functional/Technical Requirements ................................................................ 14
   5.1 STM identified non-functional requirements ...................................................... 14
6. GDPR ......................................................................................................................... 15
   6.1 GDPR requirements ............................................................................................ 15
7. References .................................................................................................................. 15
8. Appendices ................................................................................................................. 16
1 Summary

1.1 General

This document contains the technical requirement for the procurement of STM ship system within the scope of the STM Validation Project. The document describes the scope of the required functional and technical characteristics of the systems. The technical specifications, APIs and xml-codes are anticipated to be accurate but some are yet to be prototyped. This means that minor updates might be necessary. Latest updates to the specifications will be presented at the latest before the negotiation phase of the tender is commenced in order to tenderers to take updates into consideration in their final bid.

1.2 Definitions

1.2.1 Terminology for denoting a requirement

The following requirement is valid throughout this document:

- A mandatory requirement is denoted with the word “shall” and must be fulfilled.
- A criteria is denoted with the word “should” and gives added value in the evaluation if fulfilled.
- NB. Other requirement specifications referring to this specification might stipulate that a should requirement is changed to a shall requirement.

All requirements are labeled with R-X.Y-Z, where X.Y is the current chapter number and Z is a consecutive numbering to separate each requirement within the chapter. There can be several “shall” or “should” in one numbered requirement, which means that all must be fulfilled in order for the requirement to be fulfilled.

Descriptive text occurs throughout the document but is not labeled.

1.2.2 Select, turn-on, set etc.

All functions that are said to be selectable, turned-on, set, etc. will implicitly also be said to be the reverse, i.e. de-selected, turned-off, de-set, etc. unless specified explicitly to something else.
2 General requirements on the STM ship system

In this chapter the requirements that cannot be categorized into one or more of the 5 different components of the STM ship system as depicted in Figure 1 below, is stated.

R-2.1.1. The Tenderer of STM ship system shall at milestone 2 prove that the STM ship system can exchange relevant payload formats defined in reference SeaSWIM specification.

R-2.1.2. The tenderer should be able to adopt to external standards and requirements i.e. RTZ that is being updated to S-421. See reference Standard forecast.

R-2.1.3. The tenderer shall assure that the delivered system adheres to all relevant requirements regarding normal functionality in all navigational equipment, The Supplier is fully responsible that all installations of software and/or hardware, fulfilling the tender specifications, does not interfere with existing navigational systems and that the System is compliant with existing rules and regulations including but not limited to SOLAS Ch V reg 17 on electromagnetic compatibility.

R-2.1.4. The Tenderer shall conduct appropriate training and provide documentation (manual) for onboard equipment users. The tenderer shall deliver all educational material to the ship in hard copy, CBT etc. The training material shall be held updated during the project period. Planned training shall be presented at SAT.

R-2.1.5. The Tenderer shall provide shipping companies included in the testbed sufficient information including a meeting presenting the STM-concept and demo the equipment updates introduced.
3 Functional/Technical Requirements

3.1 STM Ship system overall description

At least one navigation system workstation on the ship’s bridge shall be able to interact with the STM Module. The main functions are: to receive voyage plan (VP) for further processing until used for navigation, to send the VP used for navigation to the STM Module, to send ETA to the STM Module.

Communication between the navigation system and the STM Module will ensure that the maximum possible degree of consistency is upheld between information held in the navigation system and the STM Module. If information consistency decreases due to the status of communication link between the navigation system and the STM Module, the communication status shall be known by both ends.

The STM Module is a set of software functions on some hardware that presumably is located on the ship. The STM Module may be integrated in the navigation system, if certification allows, it may be integrated into an existing planning station or it may be running on a dedicated device which is supplied by the supplier. If information consistency can be assured, the STM module can be placed ashore.

Communication between the STM Module and Online access point\(^1\) will ensure that the maximum possible degree of consistency is upheld between information held in the STM Module and at the Online access point, depending on the online status of the ship. If information consistency decreases due to the status of communication links with the ship, the communication status and the age of information shall be known by both ends.

The online access point is constantly and stably connected to the internet and represents the ship towards other actors and services. The main functions in the online access point are the VIS, Port Information Service (optional) and the SeaSWIM Connector (SSC). The Online access point may be implemented onboard the ship or at another location. Each VIS is attached to a SSC and each ship has one (1) VIS instance.

---

\(^1\)
3.2 Navigation system functionality

Ship to Ship route Exchange

R-3.2.1. The STM ship system shall be able to send and receive route segment via AIS, ASM message (8), according to Appendix 6 - Route Message system requirements, F1.

R-3.2.2. The STM ship system shall be able to present route segment accordingly in navigation system.

R-3.2.3. STM ship systems shall be able to include legspeed in route message format.

R-3.2.4. Not applicable
(As the list of specifications are used in several projects, there might be specifications that does not apply to a project)

R-3.2.5. In the STM ship system it shall be possible to make a default choice that the route message (and/or RTZs) will be shared or not, with other STM-ships.

R-3.2.6. The STM ship system shall be able to present, via ASM message (8), according to appendix: Route message format (version1.0), info/flag on AIS targets that participates in STM test beds.

R-3.2.7. STM ship systems shall be able to calculate CPA and intersection points between own/other-ships route segments (including leg speed), even if the routes are not crossing each other. The operator should have the option to chose between legspeed or actual speed.

R-3.2.8. If a ship is deviating from its intended route (exceeding pre-set limits, geographically and in time, set by own ship), that shall be presented on own ships STM ship system.

R-3.2.9. It shall be possible to do "Route Trial Manoeuvre" including own and other ships routes. The operator should have the option to chose between legspeed or actual speed.

R-3.2.10. It shall be possible for the operator to activate other ships routes on a case by case basis. Eg. Acquire them manually in the navigation system. Manual activation of routes should be the default setting.

R-3.2.11. Alerts for low route CPA/TCPA should be easy to set and optional for the operator. If limits are exceeded other ships route might be triggered to show in the display.
STM Ship System functionality in Navigation System

R-3.2.12. The navigation system **shall** be able to upload, download and display VP and S-124 to and from the STM Module. Communication between the navigation system and the STM Module will ensure that the maximum possible degree of consistency is upheld between information held in the navigation system and the STM Module.

R-3.2.13. If information consistency decreases due to the status of communication link between the navigation system and the STM Module, the communication status shall be known by both ends. The communication status shall be displayed to the operator.

R-3.2.14. When difference between real time ETA and PTA (to a predefined waypoint) is more than a preset value the operator **shall** get a question if the operator wants to update the PTA. (New PTA must be easy to set.) If yes, schedule in RTZ shall automatically be updated and sent to relevant STM actors.

R-3.2.15. It **shall** be possible to define which waypoint that is used for ETA calculations by setting a PTA at that waypoint (ETA in manual schedule in RTZ) to mark it as “synchronization” WP.

R-3.2.16. When RTA is received from STM-actors, the PTA **shall** be possible to be updated, using the received RTA. The route shall not have to be safety checked due to only updated schedule. When new PTA is set, the VP shall be shared.

R-3.2.17. It **should** be clearly visible that only the schedule has been changed and which waypoint are affected by the change.

R-3.2.18. Area received in STM text message format **shall** be possible to display in navigation system.
3.3 Communication between STM Module and navigation system

R-3.5.1. The communication between the STM Module and navigation system shall have Information consistency (e.g. ship receives a VP in the STM module, accepts it, then the operator shall be able to choose it for monitoring in navigation system, it shall then be identical to the one in the STM module).

3.4 STM Module

R-3.4.1. STM module shall be able to send, receive and show text messages to/from other STM actor according to “STM text format”, in SeaSWIM specification, appendix 5 (e.g. Text body, Message subject, receiver id (actor id), sender id (actor id), reference to information object ID (e.g. UVID)).

R-3.4.2. The STM module shall include the functionality to communicate the following (but not limited to) to and from VIS/Online access point.

- Publish VP to VIS / Online access point.
- Retrieve VP proposals from VIS / Online access point.
- Send and receive VP, Area- and text messages to other STM services through VIS / Online access point.

R-3.4.3. The STM module shall include the functionality to show, create, edit and save VP. It shall be possible to graphically clearly distinguish received VP updates, e.g. route suggestions.

R-3.4.4. Not applicable

R-3.4.5. If an appropriate schedule is not attached to the monitored VP, operator shall get notified.

R-3.4.6. The STM module shall be able to set route status according to RTZ format in SeaSWIM specification.

R-3.4.7. The route status “inactive” shall only be shared with other STM actors when the voyage is completed or cancelled.

R-3.4.8. OOW shall be made aware of relevant events such as incoming messages (VP updates, text messages new RTA etc.) The event shall be detectable from ships conning position. Above events shall not interfere with navigational information.

R-3.4.9. The STM Module shall be able to upload/download VP and S-124 to/from the ships navigation system.
R-3.4.10. Operator **shall** be able to search for service instances in STM module according to SeaSWIM specification, based on attributes available in service registry.

R-3.4.11. Operator **shall** be able to search for services by means of Geometry (e.g. area search and/or Voyageplan search).

R-3.4.12. STM module **shall** be able to present the service descriptions included in service registration.

R-3.4.13. STM module **should** be able to present all relevant information available in instance descriptions as document (PDF, HTML).

R-3.4.14. STM module **shall** be able to store lists of services (for offline purposes).

R-3.4.15. The STM module **should** be able to support offline work when connectivity goes down, e.g. search for services and call them and when connectivity returns the “call” should be processed.

R-3.4.16. If stored service list is used, the list **shall** be updated minimum once per day and/or when service is called to prevent old entries to be used.

R-3.4.17. If search fails, the operator **shall** get an error message within 30 seconds.

R-3.4.18. Operator **should** be made aware if “messages” has been sent or not, due to connectivity or other issues.

R-3.4.19. Operator **shall** in the STM module be able to assign/remove access rights on information objects (VP) to services.

R-3.4.20. STM module **shall** be able to present for the operator which actors that have been given access rights to VP.

R-3.4.21. Operator **shall** in the STM module be able to share VP to selected services. It **shall** be possible to share a VP once or to set a service as a subscriber (i.e. automatically send future updates of VP).

R-3.4.22. Operator **should** in the STM module be able to subscribe on a service.

R-3.4.23. There **should** be a functionality in the STM Module to present relationship between different payload formats, e.g. a text message can be correlated to a VP.

R-3.4.24. The STM Module **shall** be able to show navigational warning according to S-124 (See SeaSWIM specification).
R-3.4.25. The STM Module **shall** be able to show area in STM text format.

R-3.4.26. The STM Module **should** handle real time calculation (including preplanned speeds) of ETA and or STG (Speed To Go to reach a WP at a given time) to one or more selected WP(s) (e.g. arrival traffic area or Pilot Boarding Position) along the route, not necessarily the last waypoint in the route.

R-3.4.27. The STM ship system **shall** automatically give the VP a Unique Voyage ID (UVID) using the maritime resource name (mrn) structure (see SeaSWIM specification).

R-3.4.28. When a STM actor has sent e.g. a VP to another STM actor (e.g Ship-Shore, Shore-Ship) the sender **shall** get information regarding that that message is “read” by the receiver. (Requires update of VIS technical design as an intermediary before the standardized service interface).
3.5 Communications between the STM Module and Online access point

R-3.5.1. Each ship shall have data link connectivity between the STM Module and the Online access point. The connectivity has the ability to be continuous with adequate capacity according the following: Communication between the STM Module and Online access point ensures that the maximum possible degree of consistency is upheld between information held in the STM Module and at the Online access point, depending on the online connectivity status of the ship. (Quantifying of above will be done in separate projects).

3.6 Online access point (Service Instance)

R-3.6.1. The online access point shall hold a service instance that is the internet connected representation of the ship, the service instance shall represent the ship towards other actors via a SSC (service instance shall be based on latest service design VIS).

R-3.6.2. The online access point should hold a service instance that is the internet connected representation of the ship, the service instance should represent the ship in receiving port call messages (PCMF) in port-call synchronization interactions towards other actors via a SSC.

R-3.6.3. Online access point shall be able to expose and consume (to other STM actors) RTZ-format, S124-format and text format using VIS (See SeaSWIM specification).

R-3.6.4. Online access point should be able to exchange PCM-format with other STM actors (See SeaSWIM specification).

R-3.6.5. Online access point shall be permanently and stably internet connected.

R-3.6.6. External events (according to service log specification) in and out to/from the system should be stored and logged for traceability. Events could include, but are not limited to: Exchange of VP and received S-124.

R-3.6.7. Service shall be registered in service registry according to guideline (See SeaSWIM specification).

R-3.6.8. All messages, sent and received, shall be validated against payload schema.
4 STM Cyber security requirements

4.1 STM Cyber security requirements

R-4.1.1. Service instance in online access point **shall** adhere to requirements stated in SeaSWIM Connector Specification and SeaSWIM Connector Design.

R-4.1.2. Each ship **shall** be individually identifiable for the receiver of information, hence a unique certificate for the ship **shall** be used as client certificate.

R-4.1.3. The information owner **shall** implement authorization mechanism for external actors, that ensures only authorized external actors have access to its information (See also requirement R-3.4.19).

R-4.1.4. Security events **should** be logged for traceability (time, denied access etc.) in Online access point.

R-4.1.5. The communication between STM Module and Online Access Point **shall** be secured to hinder unauthenticated requests and unauthorized access to information (e.g. VPN).
5 Non-Functional/Technical Requirements

5.1 STM identified non-functional requirements

R-5.1.1. Information **shall** not be lost due to restart or due to no connection to internet.

R-5.1.2. Information shared **shall** not exceed 400kb.

R-5.1.3. The online access point **shall** be available at least 99% of the time.

R-5.1.4. The status of service call **shall** be clearly indicated (e.g. by symbols for OK, Error) to the operator. Reason for the error **shall** be presented in text. See VIS Technical Design for details.

R-5.1.5. The response message from service call **shall** be possible for an operator to read. The response message may contain information regarding the service call even if call was technically a success, such as missing information in the voyage plan.

R-5.1.6. The Online access point VIS-based instance **shall** return HTTP response (code and message/information) according to VIS Technical Design.
6 GDPR

6.1 GDPR requirements

R-6.1.1. General Data Protection Regulation (GDPR) **shall** be applied.

R-6.1.2. Personal information **shall** be avoided in the system.

R-6.1.3. GDPR handling required in MCP **shall** be followed (See SeaSWIM specification).

7 References

All additional technical documentation can be found at: https://www.stmvalidation.eu/developers-forum/
8 Appendices

8.1 SeaSWIM Specification
Document No:
Title: SeaSWIM Specification (Testbed)
Date: 2019-03-19
## DOCUMENT STATUS

### Authors

<table>
<thead>
<tr>
<th>Name</th>
<th>Organisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anders Dalén</td>
<td>Viktoria Swedish ICT</td>
</tr>
<tr>
<td>Oliver Norkus</td>
<td>University of Oldenburg</td>
</tr>
<tr>
<td>Jens Kristian Jensen</td>
<td>Danish Maritime Authority</td>
</tr>
<tr>
<td>Christoph Rihacek</td>
<td>Frequentis</td>
</tr>
<tr>
<td>Anisa Rizvanollí</td>
<td>Fraunhofer - Center für Maritime Logistik und Dienstleistungen (CML)</td>
</tr>
<tr>
<td>Per Löfbom</td>
<td>Swedish Maritime Administration</td>
</tr>
<tr>
<td>Mikael Olofsson</td>
<td>SMA</td>
</tr>
<tr>
<td>Fabio Renda</td>
<td>CIMNE</td>
</tr>
</tbody>
</table>

### Review

<table>
<thead>
<tr>
<th>Name</th>
<th>Organisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Björn Andreasson</td>
<td>Swedish Maritime Administration</td>
</tr>
<tr>
<td>Håkan Heurlin</td>
<td>Swedish Maritime Administration</td>
</tr>
<tr>
<td>Per de Flon</td>
<td>Swedish Maritime Administration</td>
</tr>
<tr>
<td>Almir Zerem</td>
<td>Viktoria Swedish ICT</td>
</tr>
<tr>
<td>Cilli Sobiech</td>
<td>Viktoria Swedish ICT</td>
</tr>
<tr>
<td>Per Setterberg</td>
<td>Swedish Maritime Administration</td>
</tr>
</tbody>
</table>

### Approval

<table>
<thead>
<tr>
<th>Name</th>
<th>Organisation</th>
<th>Signature</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Status</th>
<th>Initials</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>160512</td>
<td>Final</td>
<td>AD</td>
<td>Spring 2016 Procurement</td>
</tr>
<tr>
<td>2.0</td>
<td>161118</td>
<td>Final</td>
<td>AD,MO</td>
<td>Fall 2016 Procurement</td>
</tr>
<tr>
<td>3.0</td>
<td>190318</td>
<td>Final</td>
<td>MO,</td>
<td>2019 Procurement</td>
</tr>
</tbody>
</table>

The sole responsibility of this publication lies with the author. The European Union is not responsible for any use that may be made of the information contained therein.
# STM

## Table of Contents

1. **Introduction** ................................................................................................................................. 4  
   1.1. Sea System Wide Information Management (SeaSWIM) ........................................... 4  
   1.2. SeaSWIM Principles and Architecture .............................................................................. 5  
   1.3. Interacting with SeaSWIM ........................................................................................................ 7  
      1.3.1. Setup and Discovery ....................................................................................................... 7  
      1.3.2. Interaction ....................................................................................................................... 9  

2. **Using SeaSWIM Connector (SSC)** ..................................................................................... 11  
   2.1. Introduction to SSC ............................................................................................................. 11  
   2.2. SSC as a Proxy ..................................................................................................................... 13  
   2.3. SSC as a Function Library .................................................................................................. 14  
   2.4. Identity Management, Authentication and Encryption ..................................................... 15  

3. **Building a SeaSWIM Connector** .......................................................................................... 16  
   3.1. Identity Registry Integration ............................................................................................. 16  
      3.1.1. Conceptual Description ............................................................................................... 16  
      3.1.2. Technical Description ................................................................................................ 17  
   3.2. Service Registry Integration ............................................................................................. 21  
      3.2.1. Conceptual Description ............................................................................................... 21  
      3.2.2. Technical Description ................................................................................................ 24  

4. **STM Registration Processes** ............................................................................................... 24  
   4.1. Registering Users in Identity Registry .............................................................................. 24  
   4.2. Registering Services with the Service Registry ............................................................... 25  

5. **Unique Identification** ............................................................................................................... 25  
   5.1. Unique Voyage_ID (UVID) ............................................................................................ 30  
   5.2. Unique Port Call ID (UPCID) ......................................................................................... 32  

6. **Reference List** ......................................................................................................................... 34
1. Introduction

This appendix document is intended to provide information on what is required to integrate with STM’s maritime digital infrastructure “Sea System Wide Information Management” (SeaSWIM). This document is solely focused on the STM testbed implementation of SeaSWIM, which represents a subset of services and functionality intended for the full STM concept scope.

The document provides two perspectives; one for interacting with SeaSWIM through a support interface, which hides the full complexity of the different support services (Section 2), and one for building and integrating SeaSWIM compliant services in house (Section 3). General prerequisites and standards that have to be adhered to is described in Section 4, Section 5 and the Data Models Reference List.

1.1. Sea System Wide Information Management (SeaSWIM)

Unifying the way maritime stakeholders communicate, enables a shared understanding and a new level of potential interaction and integration. The fundamental goal for SeaSWIM is to provide and maintain a harmonized way of communicating within the maritime industry. This implies that open and accessible standards are promoted.

The SeaSWIM environment enables maritime stakeholders to share certain data and information, in real time and without interpretation over organizational boundaries. In order to achieve this level of interoperability while including a wide range of stakeholders, the scope of what can be communicated is limited to what complies with the SeaSWIM concept. The SeaSWIM concept recognizes that new and customized exchanges of information between certain actors are necessary. However, customization and testing will happen outside of the stable SeaSWIM environment. The focus of the SeaSWIM environment is primarily to ensure semantic interoperability for the most critical and purposeful communication.

To achieve the inclusive scope of the SeaSWIM environment entry barriers to develop, produce and consume data and information in the environment should be as low as possible. The SeaSWIM Connector (SSC) is implemented for this purpose – to assist developers by instantiating generic functionality that is needed by most services in the SeaSWIM environment.

The use of the SSC is voluntary; compliance with the functionality that the SSC exemplifies is not. Any industry stakeholder is free to develop its own equivalent version as long as it follows the SeaSWIM technical specification it is based on. It
should be noted that while the SSC assists developers to access the SeaSWIM environment in a compliant way there are more requirements, beyond the SSC, to become compliant with the SeaSWIM environment (e.g. accepted standards, identities, interface specifications, etc.).

In the STM testbed the provided support services will be limited to the core functionality described in this document. For example, access management is recognized as a common need that should be defined with a SeaSWIM reference implementation. However, as the specific needs and potential implementation alternatives for this support service is currently unclear it will not be provided as part of the STM testbed. Instead, application service-providers are encouraged to evaluate their needs by testing alternative solutions.

1.2. SeaSWIM Principles and Architecture

SeaSWIM consists of specific support services that will ensure interoperability of the STM application services by facilitating data sharing in a common information environment and structure. The specification of SeaSWIM is developed to adhere to some important STM principles:

1. Only authenticated actors can provide and consume STM services, where authentication is the process of determining whether someone or something is, in fact, who or what it is declared to be.
2. The owner of data is the actor responsible for the original creation and provision. The owner has full control over the access management for this data.
3. STM strives after a service oriented and highly decentralized architecture.
4. Usage of open and widely accepted industry standards wherever these exist.

The various maritime actors extract specific data or information and translate it to the appropriate STM standard format (see Appendix: Reference List).

To simplify the run time connection with the SeaSWIM environment the SeaSWIM Connector (SSC) is provided. The SSC is developed as a reference and example that handles the interaction with the core SeaSWIM support services according to the SeaSWIM specification. There exists two potential ways to incorporate the SeaSWIM Connector functionality.

- The first option is to host the reference as is and use its interface to reach the SeaSWIM environment and its connected actors and services.
- The second option is to integrate the functions of the SeaSWIM connector service to replicate its functionality. All function calls to the core SeaSWIM support services will be accepted as long as they adhere to the same standard as the SeaSWIM reference connector. It should be noted that only very limited support can be given if a vendor chooses to build their own SeaSWIM
connector functionality. Both options will be described in Section 2 (Using SeaSWIM Connector) and Section 3 (Building a SeaSWIM Connector) respectively.

The SSC or its integrated equivalent is hosted both by the providing and consuming Application Service (cf. Figure 1). This way the SSC or its equivalent support communication according to the STM principles with minimal development and implementation efforts.

![Figure 1: The SeaSWIM Connector (SSC) or equivalent as a standardized interface between the application services, the interoperable information space and a shared security domain](image)

Behind the connector, SeaSWIM consists of two central components; the **Identity Registry** and the **Service Registry** (cf. Figure 2). The Identity Registry enables identity management and authentication mechanisms, while the Service Registry provides functionality to publish and find services, including their functionality and endpoints. Both services will be further detailed in sections 2 and 3.
Unique identifiers play a crucial role in SeaSWIM and are mandatory for avoiding conflicts and provide unique references.

1.3. Interacting with SeaSWIM

This section will give an overview of the interaction with the SeaSWIM Support Services; the SeaSWIM Connector, the Identity Registry and the Service Registry. To understand how the SeaSWIM registers will interact with the connector a generic scenario description is provided. More details and descriptions about the technical interfaces and standards will be provided in the subsequent chapters in this document.

1.3.1. Setup and Discovery

To provide a service the data or information provider register with the STM identity registry, which is shown as Step 1 (Figure 3) and explained more in depth in Section 4. The identity registry ensures unique identities that can be authenticated. Without a valid STM user identity, a user will not be able to provide or consume data within STM.
With a STM user identity the service provider can be authenticated as such and register a service in the service registry, shown in Step 2 (Figure 3). Registering a service entails publishing metadata of a service specification and then registering deployed instances of that service (Step 3 in Figure 3). This registration process can be done by separate parties and one service specification can have several deployed instances registered. The service specification and deployment description can subsequently be used by other STM users to interact with the service. It is important to note that published services can be simple data provision services, aggregated information services that consist of several sources of data or data consumer endpoints where data can be sent for processing.

Registering to become a STM member is also required to search and consume services, which is shown in Step 1 (Figure 4). With a STM identity the user can be authenticated by the service registry and search the published service specifications and/or deployed instances (Step 2 in Figure 4).
Developers find the appropriate functionality among service specifications through the initial search in Step 2. When a match has been found and an appropriate service has been built to interact with the specified service it is only relevant to search within its deployed instances. Step 3 (Figure 4) shows the interaction to search to list and choose between deployed service instances and how it is supported by the SSC. The service registry responds to the request with a list of deployed instances that matches the service deployment attributes (e.g. closest in proximity or current license).

1.3.2. Interaction

The user locates the service, its description and how to make use of it. For example, this could be a link to a graphical user interface that can be opened in a web browser or it could be a specification of the API that can be used to request a data point, machine to machine without user interaction. In any case, the consumer uses the metadata to establish a connection to the service producer directly.

The interaction requires that STM identities are used and are valid. By uploading the user credentials to the SeaSWIM connector, it provides support to ensure that tokens and certificates can be authenticated (Step 1 in Figure 5).
Based on the service description and the found deployed instance the service endpoint can then be called. The request is compiled with the user identity and credentials (supported by the SSC) and sends it to the receiving connector. The Provider SeaSWIM connector (on the left in Figure 5) authenticates the request and “works” on it (e.g. checks authorization, performs optimization) and responds as shown in Step 2 (Figure 5). Depending on the service and the completeness of the request this response could be the requested payload, a notification callback or an error message.
2. Using SeaSWIM Connector (SSC)

2.1. Introduction to SSC

The SeaSWIM Connector/SSC is a software component that enables the use of IR and SR, developed by STM/E2. To achieve the inclusive scope of the SeaSWIM environment, entry barriers to develop, produce and consume data and information in the environment should be as low as possible. The SeaSWIM Connector is implemented for this purpose – to assist developers by instantiating generic functionality that is needed by most services in the SeaSWIM/MCP environment.

To simplify the run time connection with the SeaSWIM environment the SSC is provided, which can be described as a standardized interface to the more generic MCP registers. The SSC is developed as a reference service that handles the interaction with the core SeaSWIM support services according to the SeaSWIM specification.

The SSC is described in further detail in the technical design/service specification document. This specification is intended to be read by architects, system engineers and developers in charge of designing and developing an instance of the connector. It is based on the following structure/content sketched in Figure 1.

![Figure 1: SeaSWIM Connector documentation and implementations](image)

The use of the SSC technical specification and core functionality is mandatory, which ensures all partners have a compliant way to communicate within the SeaSWIM environment. Any industry stakeholder is free to develop its own equivalent version/implementation as long as it follows the SeaSWIM Specification (Figure 1).

It should be noted that while the SSC assists developers to access the SeaSWIM environment in a compliant way, there are more requirements beyond the SSC, to
become compliant with the SeaSWIM environment (e.g. adherence to accepted standards, identities and interface specifications).

There are two potential ways to incorporate the SeaSWIM Connector functionality:

1. The first option is to host the example implementation as a proxy service and use its interface to reach the SeaSWIM environment and its connected actors and services.
2. The second option is to integrate the functions of the SeaSWIM Connector service to replicate its functionality. All function calls to the core SeaSWIM support services will be accepted as long as they adhere to the same standard as the SeaSWIM reference connector.

As pictured in Figure 2, the SSC is designed to facilitate the communication between services. The function will call a generic web service part of the STM infrastructure, checking the certificates authentications. The SSC will also facilitate the communication with the identity service in order to discover that the organization is part of the STM infrastructure. Finally, the SSC handles the communication with the Service Registry service in order to discover services of the STM infrastructure.

This means, that the SSC support service intercepts the incoming service request, handles authentication and, if source is authenticated, forwards the service call to the "master" service endpoint. Authentication and encryption (SSL/TLS) is added to all outgoing messages and checked on all incoming messages.

![General Architecture]

*Figure 2: SeaSWIM Connector architectural overview*
2.2. SSC as a Proxy

SSC example implementation is a software service component that works like a proxy web server, i.e. listens on a configurable port and intercepts the incoming calls. The SSC proxy is hosted both by the providing and consuming application service (Figure 3). This way the SSC or its equivalent support communication according to the STM principles with minimal development and implementation efforts.

The following functions are fulfilled as part of runtime SeaSWIM compliance (Figure 3):

- SSC provides functions and design constraints that applies to every service- or client interface, i.e. to comply with the SeaSWIM specifications.
- SSC is intended to hide the complexity of interacting and complying with the STM support services.
- SSC offers standardized means of defining communication end-points (API) and ensures encryption of all data transferred between the end-points.
- SSC is defined as an explicit service (stand-alone or integrated into a parental service).

Figure 3: SeaSWIM Connector functionalities

SSC is designed in order to support inbound and outbound communications. Inbound receive information from other services offering an out of the box, while outbound communications send information to core services (IR and SR) and application services.
2.3. SSC as a Function Library

The SeaSwim Connector can also be integrated as a function library that manage the functions implemented in the reference implementation as a proxy. In that way a parent service that works over the SSC such as a Voyage Information Service can be implemented in an integrated and simpler way.

The main responsibilities of the library are:
- support the application service to communicate with the Service Registry and Identity Registry
- support the communication between services part of the STM infrastructure
- decrease the integration barrier for an application service that want to join to STM infrastructure

SSC library supports two operational schemas:
- Inbound communications: Receive information from other services
- Outbound communications: Send information to core services and application services

Where the “App service” below is later referred to the “parent” of the SeaSWIM Connector.

The main operational activities are;
- Search for service instance(s)
- Consume service instance
- Search for (retrieve) global identities (mainly organizations)
- Authentication and identification of consumer and provider of service
2.4. Identity Management, Authentication and Encryption

Establishing connections with a SeaSWIM connector will be based on either digital certificates based on the global X.509 standard, or ‘single sign on’ to a web service based on the OpenID Connect standard, which is based on OAuth2.

Service providers or other actors that need the ability to be authenticated in the testbed of the STM project, will be registered in the Identity Registry. The actors participating in the testbed can contact the Swedish Maritime Administration, who will assist in completing the registration in the Portal for the Identity and Service registries, for the STM testbed. Please consult Section 4 of this document.

The SeaSWIM connector will assist its parent service or consumer application in establishing secure and authenticated connections, allowing only interactions with actors who are registered users or service providers with an identity that can be authenticated. The connector will only allow outbound connections to endpoints at services with a valid certificate. Inbound connections will be accepted from other SeaSWIM connectors, that also hold a valid certificate, or where logon can be authenticated based on a trusted identity provider.

If a service provider on land uses the SeaSWIM connector, a certificate for a service should be issued from the Identity Registry. For a SeaSWIM connector to represent a particular ship, that ship must have an individual certificate issued. Each entity that need to be authenticated (e.g. a service, a ship) must have its own certificate, multiple entities cannot be tied to the same certificate. The attributes of this certificate is further detailed here [http://developers.maritimecloud.net/identity/](http://developers.maritimecloud.net/identity/).

The SeaSWIM connector will be able to provide the parent application or service with the attributes associated with the actor involved in an external connection, based on the content of the certificate or the attributes stored for a user in the Identity Registry, such as name, organizational belonging, etc. For more detail, consult Section 3.1.2.

Once the ‘owner’ of the SeaSWIM connector has downloaded the digital certificate and the private key, they must be stored in a predefined place in the installation of the SeaSWIM connector, together with the root certificate(s) of the certificate provider(s) that the SeaSWIM connector should trust.

Once the certificate is securely stored the SeaSWIM connector will use it whenever it attempts to securely connect to remote hosts, or present it to any remote host trying to connect to the SeaSWIM connector. It could also be used for digitally signing documents and messages being transferred to other SeaSWIM compliant actors, using a SeaSWIM compliant connector.
3. Building a SeaSWIM Connector

The SeaSWIM Connector (SSC) presented in Section 2 is intended to hide the complexity of interacting and complying with the STM support services. This section, instead, provides the detail how the central registers and common support services function. The purpose is to provide transparency and to enable potential integration of SeaSWIM support services with the application services.

3.1. Identity Registry Integration

In SeaSWIM, the Resource Provider does not need to implement its own workflow to support identification of Clients. It can be assumed that Clients are registered via the Identity Registry, and that a PKI (Public Key Infrastructure) infrastructure supporting certificate validation or that the Authentication request can be facilitated by a trusted Identity Manager via the MCP Identity Registry.

Registration in the Identity Registry and issuing of certificates in the STM testbed is conducted through a workflow that does not involve the SeaSWIM connector.

When deciding to construct a SeaSWIM connector, functions that are relevant to the parent service or application must be considered. The technical details on implementing interactions between the SeaSWIM connector and another SeaSWIM connector or the central functions related to Identity management and authentication are described below.


3.1.1. Conceptual Description

Once the ‘owner’ of the SeaSWIM connector has downloaded the digital certificate and the private key they must be stored in a predestined place in the installation of the SeaSWIM connector. For added security the SeaSWIM connector should store them accordingly to the PKCS#8 standard or using a hardware security module (HSM).

The main interaction with the Identity Registry are related to validation of certificates and lookup of public keys of other actors based on the global X.509 standard or execution of an online authentication process based on OpenID Connect.
3.1.2. Technical Description

The authentication mechanism in the Maritime Connectivity Platform is based on open and proven standards, and their usages will be described below.

**X.509 Certificates - Machine to Machine**

For the machine-to-machine (M2M) communication primarily used in the STM setup the Maritime Connectivity Platform provides the ability to issue X.509 certificates for entities (users, vessels and devices). These certificates will enable entities to authenticate within the Public Key Infrastructure (PKI) provided by the Maritime Connectivity Platform. For services to be a part of this PKI they will need to be registered in the Maritime Connectivity Platform and obtain a certificate.

If an entity connects to a SeaSWIM service (via a SeaSWIM Connector) using a X.509 Certificate, the standard protocol for secure connection creation, automatically authenticates the entity, and therefore validates the information embedded in the certificate.

Besides authentication the X.509 certificates can be used for encrypting the communication between SeaSWIM Connectors, using standard SSL/TLS.

The X.509 certificate that is issued will contain basic information about the entity it is issued to. Each entity belongs to an Organization which is identified by an Organization Id. The Organization Id is also referred to as "shortname" of the organization, which consists of a minimum of 3 lowercase characters. Examples of an organization’s shortname could be "dma" (for the Danish Maritime Authority), "dmi" (for the Danish Meteorological Institute) or "maersk" (for Maersk Line).

The standard information that we currently envision to be included in the X.509 certificates for each entity category can be seen in the table below.

<table>
<thead>
<tr>
<th>Field</th>
<th>User</th>
<th>Vessel</th>
<th>Device</th>
<th>Service</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN (CommonName)</td>
<td>Full name</td>
<td>Vessel name</td>
<td>Device name</td>
<td>Service Domain Name</td>
<td>Organization Name</td>
</tr>
<tr>
<td>O (Organization)</td>
<td>Organization MRN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OU (Organizational Unit)</td>
<td>&quot;user&quot;</td>
<td>&quot;vessel&quot;</td>
<td>&quot;device&quot;</td>
<td>&quot;service&quot;</td>
<td>&quot;organization&quot;</td>
</tr>
<tr>
<td>E (Email)</td>
<td>User email</td>
<td></td>
<td></td>
<td></td>
<td>Organization email</td>
</tr>
<tr>
<td>C (Country)</td>
<td>Organization country code</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UID</td>
<td>Entity MRN</td>
<td></td>
<td></td>
<td></td>
<td>Organization MRN</td>
</tr>
</tbody>
</table>

An example of the fields for a vessel could look like this:

Besides the information stored in the standard X.509 attributes listed above, the X509v3 extension SubjectAlternativeName (SAN) extension is used to store extra information. There already exists some predefined fields for the SAN extension, but they do not match the need we have for maritime related fields. Therefore the “otherName” field is used, which allows for using an Object Identifier (OID) to define custom fields. The OIDs currently used are not registered at ITU, but is randomly generated using a tool provided by ITU (see [http://www.itu.int/en/ITU-T/asn1/Pages/UUID/uuids.aspx](http://www.itu.int/en/ITU-T/asn1/Pages/UUID/uuids.aspx)). See the table below for the fields currently defined, the OID of the fields and which kind of entity that uses the fields.

<table>
<thead>
<tr>
<th>Name</th>
<th>OID</th>
<th>Used by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flagstate</td>
<td>2.25.3231006332385601570573910217875371967771</td>
<td>Vessels</td>
</tr>
<tr>
<td>Callsign</td>
<td>2.25.208070283325144527098121348946727552</td>
<td>Vessels</td>
</tr>
<tr>
<td>IMO number</td>
<td>2.25.291283622413876360871493815653100799259</td>
<td>Vessels</td>
</tr>
<tr>
<td>MMSI number</td>
<td>2.25.328433707816814908768060331477217690907</td>
<td>Vessels</td>
</tr>
<tr>
<td>AIS shiptype</td>
<td>2.25.107857171638679641902842130101018412315</td>
<td>Vessels</td>
</tr>
<tr>
<td>Port of register</td>
<td>2.25.285632790821948647314354670918887798603</td>
<td>Vessels</td>
</tr>
<tr>
<td>MRN</td>
<td>2.25.27147759844977537367656021583931046283</td>
<td>Vessels, Users, Devices, Services</td>
</tr>
<tr>
<td>Name</td>
<td>OID</td>
<td>Used by</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Permissions</td>
<td>2.25.174437629172304915481663724171734402331</td>
<td>Vessels, Users, Devices, Services</td>
</tr>
</tbody>
</table>

The first 6 fields mentioned in the table are all vessel specific, and should be self explanatory in the maritime context, but the last 2 are specific to the Maritime Connectivity Platform domain:

- "permissions": A comma separated list of permissions or roles assigned to the entity by the organization it belongs to. Also see the section on Authorization below.
- "mrn": A comma separated lists of Maritime Resource Names of the entity, in prioritized order, the highest priority first.

The Maritime Resource Name (MRN) is a unique identifier for users, vessels, etc. in the maritime domain. In the Maritime Connectivity Platform during the test phase a MRN will look like below:

```
urn:mrn:stm:identity:<entity type>:<Organization Id>:<entity id>
```

Examples could be:

```
urn:mrn:stm:identity:user:DMA:tgc
urn:mrn:stm:identity:vessel:DMA:JENS_SOERENSEN
```

See Chapter 5 for more information regarding the Maritime Resource Name.

**OpenID Connect - browser based**

For browser based authentication the Maritime Connectivity Platform Identity Registry offers an OpenID Connect based solution with federated identity management. A central Maritime Connectivity Platform Identity Broker will point to organizations registered as Identity Providers in the Maritime Connectivity Platform. For the STM testbed a special STM Identity Provider will be set up to administrate STM testbed users.

When using OpenID Connect authentication the user is identified by a JWT ID token that contains various attributes that describes the users. The current attributes used in the Maritime Connectivity Platform are listed in the table below.
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>preferred_username</td>
<td>The username of the user in the parent organization.</td>
</tr>
<tr>
<td>email</td>
<td>The email of the user.</td>
</tr>
<tr>
<td>given_name</td>
<td>Firstname of the user.</td>
</tr>
<tr>
<td>family_name</td>
<td>Lastname of the user.</td>
</tr>
<tr>
<td>name</td>
<td>Full name of the user.</td>
</tr>
<tr>
<td>org</td>
<td>The full Maritime Resource Name of the organization the user is a member of.</td>
</tr>
<tr>
<td>permissions</td>
<td>List of permissions for this user assigned by the organization the user is a member of.</td>
</tr>
<tr>
<td>mrn</td>
<td>The Maritime Resource Name of the user.</td>
</tr>
</tbody>
</table>

See the section on Certificates above for a description for the “mrn”, “org” and “permissions” attributes.

**Interface for the Identity Registry**

Besides a browser based interface, where entities can be managed, the Maritime Connectivity Platform Identity Registry will also offer a web based API that will make it possible for the SeaSWIM Connector to:

- Create/update/delete entities.
- Issue X.509 certificates for entities.
- Revoke entity certificates.
- Check for revoked certificates.
3.2. Service Registry Integration

3.2.1. Conceptual Description

This chapter below describes a sample implementation of a service registry. It is a resource that aims to provide controlled access to data necessary for governance of SOA (service-oriented architecture) projects, like the Maritime Connectivity Platform development. In effect, it is a constantly evolving catalogue of information about the available services in a SOA implementation (for the moment only some basic means to search for services are included in this chapter; it is intended to complement this list considering the needs of the services implementers later on). A service registry allows businesses to efficiently discover and communicate with each other using certain services. The ultimate goal is to allow fast and reliable communication and interoperability among diverse applications with minimal human oversight.

Services themselves and the service-based economy are a central part of the Maritime Connectivity Platform and SeaSWIM. In the context of service-oriented architecture, a service usually refers to a set of related software functionalities that can be reused for different purposes, together with policies that governs and controls its usage.

The service registry contains service specifications according to a Service Specification Standard and provisioned service instances implemented according to these service specifications. The service registry improves the visibility and accessibility of available maritime information and services. This enables service providers, consumers, and regulatory authorities to share a common view on service standards and provisioned services. The service registry does not provide actual maritime information, but a specification of various services, the information they carry, and the technical means to obtain it. The service registry also provides the mechanisms to manage the lifecycle of service specifications and service instances.

As depicted below, the service registry enables the “provider” to “publish” information related to its service instances so that the “consumer” is able to “discover” them and obtain everything (e.g. interface information) required to ultimately use these services.
**Figure 4:** General architecture of the service registry

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Service Consumers</strong></td>
<td>Consumer uses service instances provided by service providers. All users within the maritime domain can be service consumers, e.g., ships and their crew, authorities, VTS stations, organizations (e.g., meteorological), commercial service providers, etc.</td>
</tr>
<tr>
<td><strong>Service Providers</strong></td>
<td>Provides instances of services according to a service specification, e.g. deploys the service to the Service Registry. All users within the maritime domain can be service providers, e.g., ships and their crew, authorities, VTS stations, organizations (e.g., meteorological), commercial service providers, etc.</td>
</tr>
<tr>
<td><strong>Service Specification Implementers</strong></td>
<td>Implementers of services from the service provider side and/or the service consumer side. Everybody can be a service implementer but mainly this will be commercial companies implementing solutions for shore and ship.</td>
</tr>
<tr>
<td><strong>Service Specification Producers</strong></td>
<td>Producers of service specifications in accordance with STM Governance procedures.</td>
</tr>
</tbody>
</table>

The service registry is intended to facilitate or implement the Maritime Service Portfolio (MSP) concept by providing a repository for the specification of operational...
and technical services and provisioned service instances. The service registry is intended to comprise all maritime services, not only digital services, thereby making it a single reference point for provisioning and discovery.

When documenting services it has to be distinguished between service specification, service technical design and service implementation (see figure below).

![Distinction between Service Specification, Service Technical Design and Service Instance](image)

**Figure 5:** Distinction between Service Specification, Service Technical Design and Service Instance

The Service Specification describes one dedicated service at logical level in a technology-agnostic manner. On the other hand, the Service Technical Design describes the details about the actual realisation of a specific service with a dedicated technology.

It is possible to provide different technical designs, all being compliant with the same service specification. It is also possible to provide one technical design that conforms to several versions of the same service specification, for example, to allow backward compatibility to older versions of a certain specification.

A service implementation (implemented according to a given technical design) may be deployed at different locations by different service providers.

For further details about the process to be followed for describing services in a standardized way, please refer to the STM-Service-Documentation-Guidelines and individual templates[^1].

[^1]: [http://stmvalidation.eu/service-registry/]
3.2.2. Technical Description

This section describes the operations for interacting with the runtime part of the Service Registry. The technical interface to the registry is based on the IETF RFC 5222 titled “LoST: A Location-to-Service Translation Protocol” ([https://tools.ietf.org/html/rfc5222](https://tools.ietf.org/html/rfc5222)), which describes a means to map the endpoint URI for a service (or services) within a given geographic region.

The standard above, that serves as baseline for the technical interface specification, has been developed for the public safety domain where a correlation of the availability of services to certain locations is essential and therefore this (temporary) mapping is reflected in the interface definition. Nevertheless, it is not mandatory to use this functionality and one (a service designer/implementer) can define services, which are not restricted to a certain region. From that perspective, the registry responds to queries with the required means to access one or more active services matching the query and operating in the specified region or at the specified position. This geographic mapping of services to locations is subject to timed expiration so as to reflect updates of service activity due to time of life cycle considerations.

A query may return one or more such mappings, if multiple services of the same type operate in overlapping regions. It is possible to define a default mapping which is returned in case specific service was found for a given point or area.

The queries are implemented as REST API calls via HTTP. The ability of a client to query the registry may be subject to authentication.

See Reference list for detailed documents describing the service interfaces to Service Registry.

4. STM Registration Processes

4.1. Registering Users in Identity Registry

In the STM testbed the registration of entities (users, vessels and devices) will be done using a browser based interface for Identity Registry. This interface is also called the “Maritime Connectivity Platform Portal”.

Organizations need to register in Identity Registry through the MCP Portal with selected checkbox STM.
Once an organization is created, the organization will be able to create and maintain entities and create the X.509 certificates needed for authenticating using the SeaSWIM Connector.

It is also provided as a REST based API that will make it possible to maintain entities and do bulk updates of users and vessels.

4.2. Registering Services with the Service Registry

In the STM testbed the registration of service instances in the Service Registry is done by each partner. Setting the status to “released” in Production is done when approved by STM.

5. Unique Identification

This section describes how unique identifiers are constructed for the STM validation project.

The syntax described in this section can be expected to be applied to identification of ships voyages (UVID – Unique Voyage ID) and Port Calls (UPCID – Unique Port Call ID), as well as Actor and Service instances in STM.

**Unique Voyage Identifier**

The MONALISA 2.0 project identified the unique voyage identifier (UVID) as necessary for an efficient and STM compliant information exchange among the different actors and the ship during a given voyage.
Unique Port Call Identifier

The concept of Port Collaborative Decision Making (PortCDM) has identified that the synchronization of a Port Call requires a unique identification of the a particular Port Call, i.e. the sequence of events that needs to be coordinated, when a particular Vessel arrive at the port, visit the port, and departs from the port.
The maritime actors involved in a ship voyage (some of them are shown in Figure 3) or a Port Call (some are shown in Figure 4) need to commit their changes to the voyage or Port Call and get informed about the changes made by others, based on the authorization they have.

The unique identifiers identifying a voyage – or a port call – during its lifetime should satisfy following requirements:

1. It should serve as a unique identifier for binding related information being exchanged in STM. Each information object exchanged in STM that related to a Voyage or Port Call should contain the relevant identifier as reference.
2. Actors or systems who/which create or hold a Voyage or Port Call information object, should be able to decentrally generate a unique ID without requiring online access to a central function.
3. Life cycle or time dependency of the id's uniqueness should be clear and unambiguous.

---

2 **Data model, data object or payload format**
4. The syntax of the ID should not reveal actor specific information if they don’t approve. The syntax may however provide reference to where actors can seek specific information about a Voyage or Port Call, if authorized to access the information.

The syntax of identifiers using ‘Maritime Resource Name’

The (draft) IALA guideline on ‘Unique Identifiers for Maritime Resources’ currently underway defines the syntax for Maritime Resource Names like this:

Uniform Resource Names (URNs) as defined by the IETF (Internet Engineering Task Force) are intended to serve as persistent, location-independent, resource identifiers and are designed to make it easy to map other namespaces (which share the properties of URNs) into URN-space. Therefore, the URN syntax provides a mean to encode character data in a form that can be sent in existing protocols (like ASCII), transcribed on most keyboards, etc.

The URN syntax provides a mechanism to ensure the uniqueness of the name of a resource, which is already widely used in different domains such as supply chain management, unique identification of books or laws. Furthermore, this structure (syntax) approach enables delegation of the authority and gives a very large contribution in decentralization as one of the main principles of STM and SeaSWIM.

The Syntax of a Maritime Resource Name (MRN) is based on RFC 2141 published by the Internet Engineering Task Force (IETF). The basic syntax of a MRN identifier is a string with a hierarchical structure as follows:

"urn:mrn:"<NSS>

The “urn” identifies this to be a special case of a Universal Resource Name (URN), while the “mrn” identify a unique namespace within the URN. Efforts to register the "urn:mrn:" namespace with the Internet Assigned Numbers Authority (IANA) are ongoing. <NSS> is the Namespace Specific String composed as follows:

<NSS>::=<governing-organization>"":"<type>"":"<type-specific-part>

Delegation of authority

The Maritime Resource Name is intended to be an extendable mechanism allowing delegation of authority to issue unique identifiers, yet making sure that these identifiers are unique right across the maritime domain, simply by adding a structured prefix to a unique identifier inside any system.

In other words – if two different ports use simple serial numbers to identify port calls in their own system, then putting a prefix that identifies the port in front of the serial
number when referencing a particular port call outside the context of the port’s own system, will ensure that the identifier of any port call becomes globally unique.

**Purpose and construction of identifiers**

The primary purpose of ‘identifiers’ is to (uniquely) identify something. Sometimes identifiers are constructed in a way, which embeds information directly into the identifier. For instance, a phone number, 72196000 may represents the main phone number of the Danish Maritime Authority, when used in a Danish context, but the number may be identical to the phone number of another organization or individual in another country. Internationally it is represented using an international dialing code in front of the national number (+45 72196000), to ensure universal uniqueness. This international dialing code constitutes an embedded piece of information, revealing the (national) belonging of the (national) identifier. Exceptions to the rule may apply, as for instance when buying a satellite phone: the ‘international dialing code’ will identify the satellite system, rather than a national telephone exchange. Similarly, it may be possible to derive certain information from an IP address or web address in the internet domain. Embedding information into the construction of a unique identifier, may serve two purposes:

1. It may achieve a simple way to **delegate authority** to issue identifiers that only need to be unique within your own domain (like national telephone numbers), providing a way to represent your identifier in a globally unique way
2. It may enable **transfer of information needed** to discover or route information to the ‘owner’ of the entity, which the identifier is intended to identify.

For option B to be useful, it requires the ability to decipher the information embedded in the identifier, which in turn requires a stable (standardized) definition of the identifier structure (a well defined way to decode the international dialing code). Otherwise it would be more safe, to embed the information to be transferred, in the **payloads** being exchanged between identities, rather than the identifier.

The STM validation project will apply a prefix structure for issuing identifiers, that separates responsibility of defining and utilizing identifiers in a project centric way (option A above), but also apply option B to embed information that promotes discoverability of the origin of an identifier, and thus validate the appropriateness of these approaches as part of the project. This structure is based on the resource name approach presented above.

**Use of 'Maritime Resource Name' in STM**

The STM validation project will apply the Maritime Resource Name methodology to validate the concepts of Unique Voyage ID and Unique Port Call ID in the maritime context, before bringing them to relevant standardization bodies. The project has requested IALA to be registered as `<governing-organization>` for namespace:

```
"urn:mrn:stm:"
```
The intention is to subdivide this namespace into four categories:

**ID’s related to PortCDM (STM Activity 1)**

urn:mrn:stm:portcdm:<someID_type_relativing_to_PortCDM>

**ID’s related to Voyage Management (STM Activity 2)**

urn:mrn:stm:voyage:<someID_type_relativing_to_Voyage_Management>

**ID’s related to Actor Identity Management and Service Management (STM Activity 4)** will use a similar syntax:

urn:mrn:stm:org:<someID_type_relativing_to_Actor_Identity_Management>

and

urn:mrn:stm:service:<someID_type_relativing_to_Service_Management>

(or a similar URN construction with a different prefix, noting that the actor and service identity concepts are being developed in collaboration with other projects, but ultimately based on the same URN based syntax.)

The division of the namespace urn:mrn:stm into these subspaces enables a higher degree of freedom in generating unique identifiers and supporting decentralization.

### 5.1. Unique Voyage_ID (UVID)

The update of IEC 61174 test standard for ECDIS in 2015, introduced a standardized data format for representation of a ship’s voyage plan (the RTZ format).

This format includes an identifier field, which can be used to uniquely identify an instance of a ship’s planned voyage, during the lifecycle of the voyage from strategic planning, through the dynamic updates underway, until completion. For unique identification of this instance of the voyage, when communicating updates between any group of stakeholders, a globally unique identifier is needed, and methods to manage the version history of changes applied.

The STM project will establish the concept of a ‘Voyage Information Service’ as the point of contact to enable authorized parties (authorized collaborators such as agents, pilots, ports, VTSs etc.) to interact electronically with information related to a ship’s voyage. The definition of the UVID is closely related to the definition of the Voyage Information Service by Activity 2 of the STM project, and thus Activity 2 ‘owns’ this definition.

It has been observed that centralized methods for issuing unique identifiers (such as Global Unique Flight Identifiers in the aviation industry) demand connectivity at the time of creation. This is seen as an undesirable requirement and possible point of failure. Instead a delegated approach is desired, where each registered provider of a Voyage Information Service is delegates the ability to issue their own identifiers is desired.
The following definition of the UVID has been proposed, and is provided for explanation of the syntax. The final definition of the UVID is part of the documentation of the Voyage Information Service:

“urn:mrn:stm:voyage:id:”<org>”:<localid>”

Where “urn:mrn:stm:voyage:id:” is the prefix, that identifies a UVID in the Voyage Management activity of the STM validation project. <org> denotes a Voyage Information Service Provider and/or owner of the voyage id, and it’s purpose is to help discover the service endpoint (the address of the technical interface) of the Voyage Information Service where information related to this UVID can be found, via the Service Registry. <localid> is a locally generated ID (syntax: a string of alphanumeric values (a-z, A-Z, 0-9), possibly characters ′-′ or ′_′, but no space or other special characters). It could be a serial number, UUID or something else. It is generated by the provider of the Voyage Information Service, and must be unique within the context of this particular instance of a Voyage Information Service.

**Example**

This is provided as an explanatory example, not stating additional requirements.

Shipping company A (operating Ship A) has internally established their own Voyage Information Service to manage its voyages, and it is registered in the Service Registry as Voyage Information Service “urn:mrn:stm:voyage:id:a:134”.

“urn:mrn:stm:voyage:id:a:134” denotes voyage number “134” held at Voyage Information Service Provider “a”.

The example above may also apply if shipping company A employs a third-party service to host its Voyage Information Service instance, since the naming of the unique identifier is rather a contractual matter between the parties than a consequence of the technical implementation.

Shipping Company B and C (operating Ships B and C) have both made a deal with Service Provider D to take care of representing their voyages electronically. Service Provider D operate the “urn:mrn:stm:voyage:id:stm-d” Voyage Information Service.

“urn:mrn:stm:voyage:id:stm-d:346” may denote the voyage number “346” of ship B held at Voyage Information Service Provider “stm-d”. Similarly “urn:mrn:stm:voyage:id:stm-d:134” may denote the latest version of voyage number “134” (could be ship B or C), but you don’t know the ship involved, unless you ask the
Voyage Information Service – and you must be nominated as a collaborator of this voyage – or request nomination - to retrieve any information about this voyage.

Holding the authority over the “urn:mrn:stm:voyage:id:stm-d” namespace allows the service provider, if so decided in the agreement with the client, to sub-divide the namespace into client-specific namespaces, which allows the clients to manage the issuing of UVIDs under the sub-namespace on their own. The examples above illustrate how the precise usage of the “urn:mrn:stm:voyage:id” namespace is flexible and allow for different technical and contractual arrangements between shipping companies and service providers.

5.2. Unique Port Call ID (UPCID)

Issuing unique identifiers to identify a Port Call is very similar to issuing identifiers for a Voyage. The Port Call ID is owned by Activity 1 of the STM Validation project and described in the documentation of the Port Information Service. The following is provided to explain the similar syntax:

“urn:mrn:stm:portcdm:port_call:”<portCode>”:<portspecificId>"

Thus, the software developed for the STM testbeds, should accept both the prefix “urn:x-mrn”,“urn:mrnx” and “urn:mrn” as equivalent prefixes of testbed data. The prefix of future production data is to be determined depending on standardization, however the syntax of the URN notation will apply.

<portCode> denotes a Port Information Service of a particular port, identified through the UNLOCODE of a particular port – or another unique identification of the provider of a Port CDM service (syntax: a string of alphanumeric values (a-z, A-Z, 0-9), possibly characters ‘-‘ or ‘_‘, but no space or other special characters).

<portspecificId> is a locally generated ID (syntax: a string of alphanumeric values (a-z, A-Z, 0-9), possibly characters ‘-‘ or ‘_‘, but no space or other special characters). It could be a serial number, UUID or something else. It is generated by the Port system and must be unique within the context of this particular instance of a Voyage Information Service.

Example

“urn:mrn:stm:portcdm:port_call:SEGOT:b44928d8-0e93-46be-baf9-b824e0fdebe90” will denote a port call in the Port of Gothenburg (UNLOCODE: SEGOT) with a UUDI as a local reference.
Post project considerations for utilization of the UVID and UPCID identifiers

Assuming that the STM validation project successfully validates this way of handling UVIDs, UPCIDs - and associated Actor and Service Provider IDs, the construction of these identifiers will be brought forward as a proposal for a standard to be published by some relevant standardization organization.

Regardless of which standardization organization is identified as the relevant host of such a standard, the intention is to replace the prefix of the identifier construction (“urn:mrn:stm:”) with another prefix, identifying the relevant host organization. Possible results could be:

- “urn:gs1:” if the STM project ends up proposing the STM services to become GS1 standards
- Either way, the technical implementations of STM Services in the testbeds may be prepared through configuration to accept the “urn:mrn:stm:” prefix for the testbeds, but another prefix (To Be Determined) should be anticipated as the ‘production’ prefix of STM services for STM related identifiers and type definitions in the future.
6. Reference List

All data model schemas currently supported in STM can be found at the STM Validation Developer's forum webpage [http://stmvalidation.eu/schemas/](http://stmvalidation.eu/schemas/). The current listed STM standard formats are:

<table>
<thead>
<tr>
<th>Format</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route Exchange Format (RTZ)</td>
<td>[<a href="http://stmvalidation.eu/schemas/">http://stmvalidation.eu/schemas/</a>] -RTZ Format</td>
</tr>
<tr>
<td>Text Message (TXT)</td>
<td>[<a href="http://stmvalidation.eu/schemas/">http://stmvalidation.eu/schemas/</a>] -Text Message Format</td>
</tr>
<tr>
<td>Area Exchange Format (S-124)</td>
<td>[<a href="http://stmvalidation.eu/schemas/">http://stmvalidation.eu/schemas/</a>] -Area Exchange Format</td>
</tr>
<tr>
<td>Port Call Message Format (PCMF)</td>
<td>[<a href="http://stmvalidation.eu/schemas/">http://stmvalidation.eu/schemas/</a>] -Port Call Message Format</td>
</tr>
</tbody>
</table>

The documentation of the SeaSWIM Connector can be found at STM Validation Developer's forum webpage.

<table>
<thead>
<tr>
<th>Document</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>SeaSWIM L1 Connector Specification</td>
<td>[<a href="http://stmvalidation.eu/service-catalogue/">http://stmvalidation.eu/service-catalogue/</a>] -SSC</td>
</tr>
<tr>
<td>SeaSWIM L2 Connector Technical Design</td>
<td>[<a href="http://stmvalidation.eu/service-catalogue/">http://stmvalidation.eu/service-catalogue/</a>] -SSC</td>
</tr>
<tr>
<td>SeaSWIM L3 Connector Technical Design (CIMNE)</td>
<td>[<a href="http://stmvalidation.eu/service-catalogue/">http://stmvalidation.eu/service-catalogue/</a>] -SSC</td>
</tr>
<tr>
<td>SeaSWIM L3 Connector Technical Design (SMA)</td>
<td>[<a href="http://stmvalidation.eu/service-catalogue/">http://stmvalidation.eu/service-catalogue/</a>] -SSC</td>
</tr>
</tbody>
</table>

The documentation of the Voyage Information Service can be found at STM Validation Developer's forum webpage.

<table>
<thead>
<tr>
<th>Document</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voyage Information Service (VIS) Specification</td>
<td>[<a href="https://www.stmvalidation.eu/developers-forum/vis/">https://www.stmvalidation.eu/developers-forum/vis/</a>]</td>
</tr>
<tr>
<td>Voyage Information Service (VIS) REST Technical Design</td>
<td>[<a href="https://www.stmvalidation.eu/developers-forum/vis/">https://www.stmvalidation.eu/developers-forum/vis/</a>]</td>
</tr>
<tr>
<td>VIS Log Description</td>
<td>[<a href="https://www.stmvalidation.eu/developers-forum/vis/">https://www.stmvalidation.eu/developers-forum/vis/</a>]</td>
</tr>
</tbody>
</table>

The documentation of the Maritime Connectivity Platform can be found at STM Validation Developer's forum webpage.

<table>
<thead>
<tr>
<th>Document</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCP Service Registry</td>
<td>[<a href="https://www.stmvalidation.eu/developers-forum/service-registry/">https://www.stmvalidation.eu/developers-forum/service-registry/</a>]</td>
</tr>
<tr>
<td>MCP Identity Registry</td>
<td><a href="https://www.stmvalidation.eu/developers-forum/identity-registry/">https://www.stmvalidation.eu/developers-forum/identity-registry/</a></td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------</td>
</tr>
</tbody>
</table>
39 partners from 13 countries
containerising maritime information

Demonstrating the function and business value of the Sea Traffic Management concept and its services.
Seaing is believing!

SAFETY - ENVIRONMENT - EFFICIENCY

Swedish Maritime Administration ◦ SSPA ◦ Viktoria Swedish ICT ◦ Transas ◦ Chalmers University of Technology ◦ The Swedish Meteorological and Hydrological Institute ◦ Danish Maritime Authority ◦ Navicon ◦ Novia University of Applied Sciences ◦ Fraunhofer ◦ Jeppesen ◦ Carnival Corp. ◦ Italian Ministry of Transport ◦ SASEMAR ◦ Valencia Port Authority ◦ Valencia Port Foundation ◦ CIMNE ◦ University of Catalonia ◦ Norwegian Coastal Administration ◦ GS1 ◦ Cyprys University of Technology ◦ Port of Barcelona ◦ Costa Crociere ◦ Svitzer ◦ OFFIS ◦ Finnish Transport Agency ◦ Southampton Solent University ◦ Frequentis ◦ SAM Electronics ◦ University of Flensburg ◦ Signalis ◦ Maritiem Instituut Willem Barentsz ◦ SAAB TransponderTech AB ◦ University of Oldenburg ◦ Magellan ◦ Furuno Finland ◦ Rörvik ◦ University of Southampton ◦ HiQ

www.stmvalidation.eu
8.2 Route message system requirements
Route Message system requirements, G1.docx
Date 2019-05-28
# DOCUMENT STATUS

## Authors

<table>
<thead>
<tr>
<th>Name</th>
<th>Organisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Johan Lindborg (JL)</td>
<td>Saab AB</td>
</tr>
<tr>
<td>Daniel Ferm (DF)</td>
<td>Saab AB</td>
</tr>
<tr>
<td>Mikhail Andrianov (MA)</td>
<td>Transas Marine Limited</td>
</tr>
</tbody>
</table>

## Review

<table>
<thead>
<tr>
<th>Name</th>
<th>Organisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Håkan Heurlin (HH)</td>
<td>SMA</td>
</tr>
</tbody>
</table>

## Approval

<table>
<thead>
<tr>
<th>Name</th>
<th>Organisation</th>
<th>Signature</th>
<th>Date</th>
</tr>
</thead>
</table>

## Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Status</th>
<th>Initials</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2016-03-03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>2016-03-23</td>
<td>HH</td>
<td></td>
<td>Second Revision</td>
</tr>
<tr>
<td>C</td>
<td>2016-11-03</td>
<td>JL</td>
<td></td>
<td>AIS DAC and FI defined</td>
</tr>
<tr>
<td>D</td>
<td>2017-04-26</td>
<td>JL/HH</td>
<td></td>
<td>VDES DAC and FI defined</td>
</tr>
<tr>
<td>E</td>
<td>2017-09-01</td>
<td>HH/DF/PT/PB/MA</td>
<td></td>
<td>Annex A and Annex B revised regarding passing of waypoint criteria</td>
</tr>
<tr>
<td>F</td>
<td>2017-10-13</td>
<td>HH/DF/PT/MA</td>
<td></td>
<td>Final revision</td>
</tr>
<tr>
<td>G1</td>
<td>2019-05-28</td>
<td>BA</td>
<td></td>
<td>To stop broadcast if speed &lt;1knot for more than 6 min added</td>
</tr>
</tbody>
</table>
Table of contents

1 Sharing and receiving Monitored Routes ................................................................. 4
2 System components ................................................................................................. 5
   2.1 HMI and operational equipment ....................................................................... 5
   2.2 Communication Device ..................................................................................... 5
   2.3 Interfaces ............................................................................................................. 5
       2.3.1 Data formats ................................................................................................. 5
3 Route Messages .......................................................................................................... 6
   3.1 Onboard message flow ....................................................................................... 6
   3.2 Message broadcast trigger events ....................................................................... 6
   3.3 Route segmentation for AIS broadcasts ............................................................ 6
   3.4 Route segmentation for VDES ASM broadcasts ............................................... 7
4 Route Message structure ........................................................................................... 8
   4.1 AIS Route Message ............................................................................................ 8
   4.2 VDES ASM Route Message .............................................................................. 9
   4.3 Interrogation ....................................................................................................... 10
5 ANNEX A – AIS Route Broadcast Logic ................................................................. 11
   5.1 Standard AIS Route Broadcasts ........................................................................ 11
   5.2 AIS Route Exceptions for Long Legs ................................................................. 13
       5.2.1 Length of the current route leg is more than delta LAT & LON limits. Distance to WP1 is more than delta LAT & LON limits. ......................................................... 13
       5.2.2 Length of the current route leg is more than delta LAT & LON limits. Distance to WP1 is less than delta LAN & LON limits. WP1-WP2 delta LAT & LON > ±209.7151’ limits. 14
       5.2.3 Length of the current route leg is more than delta LAT & LON limits. Distance to WP1 is less than delta LAT & LON limits. WP1-WP2 delta LAT & LON is less than ±209.7151’ limits. ................................................................. 15
6 ANNEX B – VDES ASM Route Broadcast Logic ..................................................... 17
1 Sharing and receiving Monitored Routes

This document describes a method for sharing route data between ships, with an STM compliant system using VDES or AIS equipment for communication. The Route Message Broadcast is used as a means to indicate intended navigation and route information to nearby ships, allowing ships to avoid ending up in a close quarter situation where the involved ships have to comply with the COLREG rules. The current and a fixed number of coming route legs of the monitored route is shared with other ships.

The use of this message requires that both the receiving and transmitting ship has the appropriate communication equipment as well as presentation and navigation systems.
2 System components

2.1 HMI and operational equipment
The Route Message sharing system presentation and is done through an ECDIS or similar system. The ECDIS is also responsible for assembly, disassembly, and interpretation of Route Message data.

   Note: ECDIS in all parts of this document may also be INS or other systems with the STM functional module integrated.

2.2 Communication Device
The ECDIS shall be connected to a VDES unit supporting both AIS and ASM-channel communication*. This communication device will function only as a modem in this system. The VDES unit shall employ a carrier sense technique to avoid VHF broadcast at the same time and channel as coastal VHF stations. The maximum length of one broadcast shall not exceed three AIS TDMA slots (<80 ms), to ensure compliance to AIS and VDES equipment standards.

   *Note: Standalone AIS may be used where VDES is not available. Loss of route message data fidelity will occur (fewer legs and no leg XTD values).

2.3 Interfaces
The ECDIS shall communicate with the Communication Device using IEC 61162-2 or IEC 61162-450 compliant interfaces.

2.3.1 Data formats
Incoming Data will be presented to the ECDIS using standard VDM messages as defined in above referenced interface standards.

Data sent from ECDIS to the communication device unit shall be using ABM and BBM messages, as defined in the same interface standards.
3 Route Messages

3.1 On-board message flow

The ECDIS initiates any Route Message broadcast to be transmitted by the Communication Device, as well as any Route Message interrogations. The ECDIS will compile the Route Message data payload, or interrogation message payload, as defined in section 4, and send a BBM or ABM message with the payload to the Communication Device.

The Communication Device will provide the ECDIS with VDM-message data as they are received on the VHF data links.

VDM message payloads may be:

- Remote AIS target dynamic, static and voyage data (AIS Msg 1,2,3,5,9,18,19,24A,24B)
- Remote ship route message data (AIS msg 8 or VDES ASM msgs)
- Route Message interrogation messages (AIS msg 6 or VDES ASM msgs)

3.2 Message broadcast trigger events

The message shall be broadcast so that ships in receiving range has accurate information at a reasonable delay, without creating unnecessary load on the VHF datalinks.

A new Route Message broadcast shall be initiated by ECDIS when any of the below events occurs:

- Six minutes have passed since last Route Message broadcast
- A Route Message interrogation was received and over one minute has passed since last Route Message broadcast on that channel
- Any of the data in the last Route Message broadcast has been changed
- When passing a waypoint (after completing the turn, see Annex A for definition)
- The Monitored Route has been deactivated
- A Monitored Route is activated

The Route Message broadcast shall be initiated by the ECDIS as soon as possible after any of the above broadcast triggers.

There shall be no autonomous Route Message broadcasts when there is no Monitored Route active in ECDIS, except for one additional Route Message transmission after stopping a Monitored Route. The system shall however respond to interrogations, with an empty route message. If the ship has had a speed below 1 (one) knot for more than 6 minutes no route message shall be broadcasted

3.3 Route segmentation for AIS broadcasts

The Route Message-payload is defined such that up to seven legs can be shared. The first leg shared during a turn shall be the leg leading up to the turn, otherwise it shall be the current leg. If any of the waypoints within the seven legs are over 200 NM (max ±209.7151') from the previous waypoint, this will be treated as the last WP in the currently shared monitored route segment, or a virtual FROM waypoint inserted (see 5.2 in Annex A for details).
3.4 Route segmentation for VDES ASM broadcasts

The Route Message-payload is defined such that the current leg as well as up to twelve additional legs can be shared, with Cross Track Distance (XTD) parameters for each leg.
4 Route Message structure

The Route Message structures are defined in the following tables. When no Monitored Route is active, the Route Message shall have an empty payload. Two Route Messages are defined, depending on the communication system used. VDES ASM channel communication allows for a higher bitrate, and more data within the three timeslot limit.

Leg parameters (planned speed, XTD, geometry) are for the leg ending at the waypoint following those parameters. Turn radius is for the next waypoint.

4.1 AIS Route Message

The first and last waypoints are fully qualified, while intermediate waypoints are defined as the difference from the previous waypoint in the route. This way of defining intermediate waypoints saves bits but limits the maximum length of those legs. In cases where representable length is insufficient, the total route segment reported in a message is cut short and the “intermediate” waypoint becomes the last waypoint in the message.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Bits</th>
<th>Description</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message ID</td>
<td>6</td>
<td>Identifier for Message 8; always 8</td>
<td></td>
</tr>
<tr>
<td>Repeat indicator</td>
<td>2</td>
<td>Used by the repeater to indicate how many times a message has been repeated.</td>
<td></td>
</tr>
<tr>
<td>Source ID</td>
<td>30</td>
<td>MMSI number of source station</td>
<td></td>
</tr>
<tr>
<td>Spare</td>
<td>2</td>
<td>Not used. Should be set to zero. Reserved for future use</td>
<td></td>
</tr>
<tr>
<td>DAC</td>
<td>10</td>
<td>$DAC = 265_{16} = 0100001001_{2}$</td>
<td></td>
</tr>
<tr>
<td>FI</td>
<td>6</td>
<td>Function identifier = $1_{16} = 000001_{2}$</td>
<td></td>
</tr>
<tr>
<td>First waypoint type</td>
<td>1</td>
<td>0 = Navigating away from first waypoint (on route)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Navigating towards first waypoint (start of route)</td>
<td></td>
</tr>
<tr>
<td>First waypoint longitude</td>
<td>28</td>
<td>Longitude in 1/10 000 min ($\pm 180^\circ$, East = positive (as per 2’s complement), West = negative (as per 2’s complement). $181^\circ = (6791AC0h) = not available = default$)</td>
<td>Standard AIS format</td>
</tr>
<tr>
<td>First waypoint latitude</td>
<td>27</td>
<td>Latitude in 1/10 000 min ($\pm 90^\circ$, North = positive (as per 2’s complement), South = negative (as per 2’s complement). $91^\circ = (3412140h) = not available = default$)</td>
<td>Standard AIS format</td>
</tr>
<tr>
<td>Intermediate legs</td>
<td>64</td>
<td>Repeats 0 … 6 times depending on total number of legs</td>
<td></td>
</tr>
<tr>
<td>Leg geometry</td>
<td>1</td>
<td>0 = Loxodrome (Rhumb line)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Orthodrome (Great circle)</td>
<td></td>
</tr>
<tr>
<td>Planned speed</td>
<td>10</td>
<td>Planned speed over ground in 1/10 knot steps (0-102.2 knots)</td>
<td>Standard AIS format</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$1023 = not available, 1022 = 102.2 knots or higher</td>
<td></td>
</tr>
<tr>
<td>Turn radius</td>
<td>9</td>
<td>Turn radius in 1/100 NM. 0 = not available</td>
<td>Max 5.11 NM</td>
</tr>
<tr>
<td>Longitude delta</td>
<td>22</td>
<td>Longitude difference from previous waypoint in 1/10 000 min. East = positive, West = negative (as per 2’s complement). Max $\pm209.7151^\circ$</td>
<td></td>
</tr>
<tr>
<td>Latitude delta</td>
<td>22</td>
<td>Latitude difference from previous waypoint in 1/10 000 min. East = positive, West = negative (as per 2’s complement). Max $\pm209.7151^\circ$</td>
<td></td>
</tr>
<tr>
<td>Final reported leg</td>
<td>66</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Leg geometry</td>
<td>1</td>
<td>0 = Loxodrome (Rhumb line)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Orthodrome (Great circle)</td>
<td></td>
</tr>
<tr>
<td>Planned speed</td>
<td>10</td>
<td>Planned speed over ground in 1/10 knot steps (0-102.2 knots)</td>
<td>Standard AIS format</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$1023 = not available, 1022 = 102.2 knots or higher</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Bits</td>
<td>Description</td>
<td>Comment</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>Last waypoint longitude</td>
<td>28</td>
<td>Longitude in 1/10 000 min (±180°, East = positive (as per 2’s complement), West = negative (as per 2’s complement). 181° = (6791AC0h) = not available = default)</td>
<td>Standard AIS format</td>
</tr>
<tr>
<td>Last waypoint latitude</td>
<td>27</td>
<td>Latitude in 1/10 000 min (±90°, North = positive (as per 2’s complement), South = negative (as per 2’s complement). 91° (3412140h) = not available = default)</td>
<td>Standard AIS format</td>
</tr>
<tr>
<td>Steering mode</td>
<td>2</td>
<td>0 = Manual (default)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Heading control</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = Track control</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = Reserved for future use</td>
<td></td>
</tr>
<tr>
<td>Spare</td>
<td>4</td>
<td>Padding to bring total message length to a byte boundary. Always 0</td>
<td>Required for AIS</td>
</tr>
</tbody>
</table>

### 4.2 VDES ASM Route Message

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Bits</th>
<th>Description</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message ID</td>
<td>6</td>
<td>Identifier for Message 8; always 8</td>
<td>To be defined by VDES standard</td>
</tr>
<tr>
<td>Repeat indicator</td>
<td>2</td>
<td>Used by the repeater to indicate how many times a message has been repeated.</td>
<td></td>
</tr>
<tr>
<td>Source ID</td>
<td>30</td>
<td>MMSI number of source station</td>
<td></td>
</tr>
<tr>
<td>Spare</td>
<td>2</td>
<td>Not used. Should be set to zero. Reserved for future use</td>
<td></td>
</tr>
<tr>
<td>DAC</td>
<td>10</td>
<td>DAC = 26510 = 01000010012</td>
<td></td>
</tr>
<tr>
<td>FI</td>
<td>6</td>
<td>Function identifier = 210 = 0000102</td>
<td></td>
</tr>
</tbody>
</table>

*If no monitored route is available the message ends here*

| First waypoint type             | 1    | 0 = Navigating away from first waypoint (on route)                           |                                              |
|                                 |      | 1 = Navigating towards first waypoint (start of route)                      |                                              |
| First waypoint longitude        | 28   | Longitude in 1/10 000 min (±180°, East = positive (as per 2’s complement), West = negative (as per 2’s complement). 181° = (6791AC0h) = not available = default) | Standard AIS format                           |
| First waypoint latitude         | 27   | Latitude in 1/10 000 min (±90°, North = positive (as per 2’s complement), South = negative (as per 2’s complement). 91° (3412140h) = not available = default) | Standard AIS format                           |

<table>
<thead>
<tr>
<th>Intermediate legs</th>
<th>(97)</th>
<th>Repeats 0 … 12 times depending on total number of legs</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>XTD port side</td>
<td>11</td>
<td>XTD port side in 1/1000 NM. 0 = not available</td>
<td>Max 2.047 NM</td>
</tr>
<tr>
<td>XTD starboard</td>
<td>11</td>
<td>XTD starboard in 1/1000 NM. 0 = not available</td>
<td>Max 2.047 NM</td>
</tr>
<tr>
<td>Leg geometry</td>
<td>1</td>
<td>0 = Loxodrome (Rhumb line)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Orthodrome (Great circle)</td>
<td></td>
</tr>
<tr>
<td>Planned speed</td>
<td>10</td>
<td>Planned speed over ground in 1/10 knot steps (0-102.2 knots)</td>
<td>Standard AIS format</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 023 = not available, 1 022 = 102.2 knots or higher</td>
<td></td>
</tr>
<tr>
<td>Turn radius</td>
<td>9</td>
<td>Turn radius in 1/100 NM. 0 = not available</td>
<td>Max 5.11 NM</td>
</tr>
<tr>
<td>Waypoint longitude</td>
<td>28</td>
<td>Longitude in 1/10 000 min (±180°, East = positive (as per 2’s complement), West = negative (as per 2’s complement). 181° = (6791AC0h) = not available = default)</td>
<td>Standard AIS format</td>
</tr>
<tr>
<td>Waypoint latitude</td>
<td>27</td>
<td>Latitude in 1/10 000 min (±90°, North = positive (as per 2’s complement), South = negative (as per 2’s complement). 91° (3412140h) = not available = default)</td>
<td>Standard AIS format</td>
</tr>
</tbody>
</table>

<p>| Final reported leg              | (88) |                                                                              |                                              |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Bits</th>
<th>Description</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>XTD port side</td>
<td>11</td>
<td>XTD port side in 1/1000 NM. 0 = not available</td>
<td>Max 2.047 NM</td>
</tr>
<tr>
<td>XTD starboard</td>
<td>11</td>
<td>XTD starboard in 1/1000 NM. 0 = not available</td>
<td>Max 2.047 NM</td>
</tr>
<tr>
<td>Leg geometry</td>
<td>1</td>
<td>0 = Loxodrome (Rhumb line)</td>
<td>Standard AIS format</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Orthodrome (Great circle)</td>
<td></td>
</tr>
<tr>
<td>Planned speed</td>
<td>10</td>
<td>Planned speed over ground in 1/10 knot steps (0-102.2 knots)</td>
<td>Standard AIS format</td>
</tr>
<tr>
<td>Last waypoint longitude</td>
<td>28</td>
<td>Longitude in 1/10 000 min (±180°, East = positive (as per 2’s complement),</td>
<td>Standard AIS format</td>
</tr>
<tr>
<td></td>
<td></td>
<td>West = negative (as per 2’s complement). 181° = (6791AC0h) = not available</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>= default)</td>
<td></td>
</tr>
<tr>
<td>Last waypoint latitude</td>
<td>27</td>
<td>Latitude in 1/10 000 min (±90°, North = positive (as per 2’s complement),</td>
<td>Standard AIS format</td>
</tr>
<tr>
<td></td>
<td></td>
<td>South = negative (as per 2’s complement). 91° (3412140h) = not available</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>= default)</td>
<td></td>
</tr>
<tr>
<td>Steering mode</td>
<td>2</td>
<td>0 = Manual (default)</td>
<td>For byte alignment</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = Heading control</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2 = Track control</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 = Reserved for future use</td>
<td></td>
</tr>
<tr>
<td>Spare</td>
<td>0..7</td>
<td>Padding to bring total message length to a byte boundary. Always 0</td>
<td></td>
</tr>
</tbody>
</table>

### 4.3 Interrogation

Standard AIS/VDES interrogation for a specific functional message shall be used to initiate a request of a Route Message from a remote ship. The reply shall always be as defined above (broadcast), to allow all ships within range to receive the message. The reply should be sent with empty content when there is no monitored route in use.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Bits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message ID</td>
<td>6</td>
<td>Identifier for Message 6; always 6</td>
</tr>
<tr>
<td>Repeat indicator</td>
<td>2</td>
<td>Used by the repeater to indicate how many times a message has been repeated.</td>
</tr>
<tr>
<td>Source ID</td>
<td>30</td>
<td>MMSI number of source station</td>
</tr>
<tr>
<td>Sequence number</td>
<td>2</td>
<td>0-3</td>
</tr>
<tr>
<td>Destination ID</td>
<td>30</td>
<td>MMSI number of destination station</td>
</tr>
<tr>
<td>Retransmit flag</td>
<td>1</td>
<td>Retransmit flag should be set upon retransmission:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = no retransmission = default; 1 = retransmitted</td>
</tr>
<tr>
<td>Spare</td>
<td>1</td>
<td>Not used. Should be zero</td>
</tr>
<tr>
<td>DAC</td>
<td>10</td>
<td>International DAC = 110 = 000000000102</td>
</tr>
<tr>
<td>FI</td>
<td>6</td>
<td>Function identifier = 210 = 00001000</td>
</tr>
<tr>
<td>Requested DAC code</td>
<td>10</td>
<td>DAC = 26510 = 010000100102</td>
</tr>
<tr>
<td>Requested FI code</td>
<td>6</td>
<td>Function identifier = 110 = 00000101 or Function identifier = 210 = 0000102</td>
</tr>
<tr>
<td>Total length</td>
<td>104</td>
<td>The resulting Message 6 occupies 1 slot.</td>
</tr>
</tbody>
</table>
5 ANNEX A – AIS Route Broadcast Logic

5.1 Standard AIS Route Broadcasts

Standard AIS route broadcast are used when all route segments are less than delta LAT & LON limits (max ±209.7151'), otherwise see 5.2 nedan.

Regular Conditions

Transmitter (ECDIS)

Step 1. Own ship position is on the straight part of leg. Transmitter (ECDIS) uses the standard broadcast trigger events:
- FIRST WP: “WP0”
- Following WPs: Max 7 Waypoints / 6 Route legs ahead, including turn radiuses (if available).

Step 2. Own ship passing ‘SOT’ (Start Of Turn line):
- No additional broadcast event.
**Step 3.** Own ship sailing on curve (i.e. on turn radius). No additional broadcast events, except the periodic broadcast interval (*every 6 minutes*) which may be reached since last Route Message broadcast, in that case (same as Step 1):
- FIRST WP: “WP0”
- Following WPs: Max 7 Waypoints / 6 Route legs ahead, including turn radiuses (if available).

**Receiver (ECDIS): Transmitting ship at step 3**

**Step 4.** Own ship passing ‘EOT’ (End Of Turn line). The ‘WP passing’ event is triggered, resulting in:
- FIRST WP: “WP1”
- Following WPs: Max 7 Waypoints / 6 Route legs ahead, including turn radiuses (if available).
- Previous “WP0” is not broadcast anymore and removed from the receiver ECDIS chart area including the WP1 radius.

**Receiver (ECDIS): Transmitting ship at step 5**

**Step 5.** Same as **Step 1** above.
5.2 AIS Route Exceptions for Long Legs

Due to the AIS Route message representation of intermediate waypoints in delta LAT & LON format very long legs (> ±209.7151') must be reported differently. Depending on the ship position either the broadcast AIS route contains fewer legs, or a virtual waypoint on the current leg is used to allow additional legs to be reported, in accordance the sections below. The virtual waypoint shall be inserted on the current leg so that future legs can be reported in ample time before reaching them, ideally as far away from the leg end waypoint as the delta LAT & LON format (≤ ±209.7151’) allows.

5.2.1 Length of the current route leg is more than delta LAT & LON limits.

**Own ship distance to WP1 is more than delta LAT & LON limits.**

```
WP0-WP1: Delta LAT & LON > ±209.7151'
```

**Transmitter (ECDIS)**

```
WP0-WP1: Delta LAT & LON > ±209.7151'
```

**Receiver (ECDIS): Transmitting ship at step 1**

```
WP0-WP1 delta LAT & LON > ±209.7151'
Distance To Waypoint 1 is more than delta LAT & LON limits
```

**Legend:**

- **WP0** Waypoint ID
- **WP1** Waypoint ID
- **SOT** ‘Start Of Turn’ point
- **EOT** ‘End Of Turn’ point
- **VWP** (Virtual Waypoint)
- **Curve (Turn Radius) segment**

**Step 1.** Own ship position is on the straight leg, WP0-WP1 delta LAT & LON > ±209.7151’. Ship’s distance to WP1 is also more than LAT & LON > ±209.7151’. Transmitter (ECDIS) uses the standard broadcast trigger events (every 6 minutes). AIS Route Message broadcast:
- FIRST WP: “WP0”
- LAST WP: “WP1”
5.2.2 Length of the current route leg is more than delta LAT & LON limits. Own ship distance to WP1 is less than delta LAN & LON limits. WP1-WP2 delta LAT & LON > ±209.7151’ limits.

\[ \text{WP0-WP1: Delta LAT & LON > ±209.7151’} \]
\[ \text{WP1-WP2: Delta LAT & LON > ±209.7151’} \]

**Transmitter (ECDIS)**

**Receiver (ECDIS): Transmitting ship at step 1**

**Step 1.** A regular broadcast event occurs and the own ship has passed the Virtual Waypoint (the ship’s distance to Waypoint 1 is less than LAT & LON ±209.7151’). There shall be no additional broadcast due to passing the Virtual Waypoint, but any broadcast after passing the waypoint shall use the Virtual Waypoint as the FIRST WP:
- FIRST WP: “VWP”
- SECOND WP: “WP1”, delta LAT & LON between VWP and WP1 is less than ±209.7151’
- LAST WP: “WP2”

**Step 2.** Own ship passing ‘SOT’ (Start Of Turn line):
- No additional broadcast event.

**Step 3.** Own ship sailing on curve (i.e. on turn radius). No additional broadcast events, except the periodic broadcast interval (every 6 minutes) which may be reached since last Route Message broadcast, in that case:
- FIRST WP: “VWP”
- SECOND WP: “WP1”, delta LAT & LON between VWP and WP1 is less than ±209.7151’
- LAST WP: “WP2”

**Step 4.** Own ship passing ‘EOT’ (End Of Turn line). The ‘WP passing’ event is triggered resulting in:
- FIRST WP: “WP1”
- LAST WP: "WP2", no more waypoints because delta LAT & LON between WP1 and WP2 is more than ±209.7151'
- Previous "VWP" is not broadcast anymore and removed from the receiver ECDIS chart area including the WP1 radius.

**Step 5.** Same as 5.2.1 ovam.

5.2.3 Length of the current route leg is more than delta LAT & LON limits. Own ship distance to WP1 is less than delta LAT & LON limits. WP1-WP2 delta LAT & LON is less than ±209.7151’ limits.

**WP0-WP1:** Delta LAT & LON > ±209.7151’

**Transmitter (ECDIS)**

**Receiver (ECDIS):** Transmitting ship at step 1

**Step 1.** A regular broadcast event occurs and the own ship has passed the Virtual Waypoint (the ship’s distance to Waypoint 1 is less than LAT & LON ±209.7151’). There shall be no additional broadcast due to passing the Virtual Waypoint, but any broadcast after passing the waypoint shall use the Virtual Waypoint as the FIRST WP:
- FIRST WP: “VWP”
- Following WPs: Max 7 Waypoints / 6 Route legs ahead, including turn radiuses (if available).

**Step 2.** Own ship passing ‘SOT’ (Start Of Turn line):
- No additional broadcast event.

**Step 3.** Own ship sailing on curve (i.e. on turn radius). No additional broadcast event, except the periodic broadcast interval (every 6 minutes) which may be reached since last Route Message broadcast, in that case:
- FIRST WP: “VWP”
- Following WPs: Max 7 Waypoints / 6 Route legs ahead, including turn radiuses (if available).
Step 4. Own ship passing ‘EOT’ (End Of Turn line). The ‘WP passing’ event is triggered, resulting in:

- FIRST WP: “WP1”
- Following WPs: Max 7 Waypoints / 6 Route legs ahead, including turn radiuses (if available).
- Previous “VWP” is not broadcast anymore and removed from the receiver ECDIS chart area including the WP1 radius.

Receiver (ECDIS): Transmitting ship at step 5

Step 5. Standard broadcast rules, see 5.1 ovan.
**6 ANNEX B – VDES ASM Route Broadcast Logic**

**Regular Conditions**

**Transmitter (ECDIS)**

Step 1. Own ship position is on the straight leg. Transmitter (ECDIS) uses the standard broadcast trigger events.
- FIRST WP: "WP0"
- Following WPs: Max 13 Waypoints / 12 Route legs ahead, including Port & Starboard cross track distances and turn radiuses (if available).

**Receiver (ECDIS): Transmitting ship at step 1**

Step 2. Own ship passing 'SOT' (Start Of Turn line):
- No additional broadcast event.

Step 3.Own ship sailing on curve (i.e. on turn radius). No additional broadcast event, except the periodic broadcast interval (*every 6 minutes*) which may be reached since last Route Message broadcast, in that case:
- FIRST WP: "WP0"
- Following WPs: Max 13 Waypoints / 12 Route legs ahead, including Port & Starboard cross track distances and turn radiuses (if available).
Step 4. Own ship passing ‘EOT’ (End Of Turn line). The ‘WP passing’ event is triggered, resulting in:
- FIRST WP: “WP1”
- Following WPs: Max 13 Waypoints / 12 Route legs ahead, including Port & Starboard cross track distances and turn radiuses (if available).
- Previous “WP0” is not broadcast anymore and removed from the receiver ECDIS chart area including the WP1 radius.

Step 5. Same as Step 1 above.
39 partners from 13 countries
containerising maritime information

Demonstrating the function and business value of the Sea Traffic Management concept and its services.
Seaiing is believing!

SAFETY - ENVIRONMENT - EFFICIENCY

Swedish Maritime Administration ◦ SSPA ◦ Viktoria Swedish ICT ◦ Transas ◦ Chalmers University of Technology ◦ The Swedish Meteorological and Hydrological Institute ◦ Danish Maritime Authority ◦ Navicon ◦ Novia University of Applied Sciences ◦ Fraunhofer ◦ Jeppesen ◦ Carnival Corp. ◦ Italian Ministry of Transport ◦ SASEMAR ◦ Valencia Port Authority ◦ Valencia Port Foundation ◦ CIMNE ◦ University of Catalonia ◦ Norwegian Coastal Administration ◦ GS1 ◦ Cyprys University of Technology ◦ Port of Barcelona ◦ Costa Crociere ◦ Svitzer ◦ OFFIS ◦ Finnish Transport Agency ◦ Southampton Solent University ◦ Frequentis ◦ SAM Electronics ◦ University of Flensburg ◦ Signalis ◦ Maritiem Instituut Willem Barentsz ◦ SAAB TransponderTech AB ◦ University of Oldenburg ◦ Magellan ◦ Furuno Finland ◦ Rörvik ◦ University of Southampton ◦ HiQ

www.stmvalidation.eu

Co-financed by the European Union
Connecting Europe Facility
8.3 Text message schema description
Document No:
Title: Text message schema description
Date: 2019-03-22
Authors

<table>
<thead>
<tr>
<th>Name</th>
<th>Organisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per Löfbom</td>
<td>SMA</td>
</tr>
<tr>
<td>Mikael Olofsson</td>
<td>SMA</td>
</tr>
<tr>
<td>Per de Flon</td>
<td>SMA</td>
</tr>
<tr>
<td>Mattias Johansson</td>
<td>SMA</td>
</tr>
</tbody>
</table>

Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Initials</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 1.1</td>
<td>20160914</td>
<td>MO, PL, PD</td>
<td>Created based on VIS Specification 2.0</td>
</tr>
<tr>
<td>Version 1.2</td>
<td>20170105</td>
<td>MO, PL, PD, MJ</td>
<td>Modified to align with VIS Specification 2.1</td>
</tr>
<tr>
<td>Version 1.3</td>
<td>20170518</td>
<td>MO, PL, PD, MJ</td>
<td>Added the following elements:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>informationObjectReferenceType</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>validityPeriodStart</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>validityPeriodStop</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Added textMessageId regex condition in MRN format</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Added surface element to cope with circle</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Minor patches to above changes.</td>
</tr>
</tbody>
</table>

Review

<table>
<thead>
<tr>
<th>Name</th>
<th>Organisation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table of contents

1 Introduction ....................................................................................................................................................4
  1.1 Release Notes ........................................................................................................................................ 4
  1.2 Intended use .......................................................................................................................................... 4
  1.3 Description ........................................................................................................................................... 4

2 Text Message Schema ......................................................................................................................................5

3 References .................................................................................................................................................. 28

4 Acronyms and Terminology .......................................................................................................................... 28

5 Appendix A textMessage - example .............................................................................................................. 29
1 Introduction

1.1 Release Notes

The text message schema is modified not to reference S100/ S124 schema for describing area element. This aims to reduce the complexity of implementing the text message. The naming, formats and constraints of included area elements are limited to only use point, polygon and circle element identical to what is specified in S100/ S124 schema.

1.2 Intended use

The text message is a lightweight message intended to be used in communication between Voyage Information Services implemented in STM. Normally a text message is submitted as a complement in sending voyage plans between different parties in STM.

1.3 Description

The contents of a text message is similar to that of a normal email where optionally position and/ or area information are added in a structured way.

The structure of the textMessageId follows MRN guideline.

\[ u:mrn:<governing organization>:txt:<own organization>:identity number \]

Example:

\[ u:mrn:stm:txt:sma:20170510104400-1 \]
2 Text Message Schema

schema location: http://s3-eu-west-1.amazonaws.com/stm-stmvalidation/uploads/20161118115716/textMessageSchema_1.3.xsd

attributeFormDefault: qualified
elementFormDefault: qualified
targetNamespace: http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd

Elements
- textMessage

Complex types
- GM_Point
- GM_Surface
- S100_CircleByCenterPointType

Simple types
- BearingType
- DateTimeUTC
- informationObjectTypeEnum
- LatitudeType
- LongitudeType
- positiveDouble
- textMessageURN

element textMessage
diagram
<table>
<thead>
<tr>
<th>property</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>namespace</td>
<td><a href="http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd">http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd</a></td>
</tr>
<tr>
<td>properties</td>
<td>complex</td>
</tr>
<tr>
<td>children</td>
<td></td>
</tr>
<tr>
<td>source</td>
<td></td>
</tr>
</tbody>
</table>

```xml
<xs:element name="textMessage">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="textMessageId" type="textMessageURN" minOccurs="1" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>Identifier of the text message, mandatory.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="informationObjectReferenceId" type="xs:string" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>A reference to an information object, optional.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="informationObjectReferenceType" type="informationObjectTypeEnum" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>STM payload format reference, optional.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="validityPeriodStart" type="DateTimeUTC" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>Start of validity period in ISO 8601 format, optional.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="validityPeriodStop" type="DateTimeUTC" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>Stop of validity period in ISO 8601 format, optional.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="author" type="xs:string" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>The message author, mandatory.</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="from" type="xs:string" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>Author of the message, mandatory.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```
<xs:element name="serviceType" type="xs:string" minOccurs="0" maxOccurs="1">
    <xs:documentation>The service type of the sender, optional.</xs:documentation>
</xs:element>
<xs:element name="createdAt" type="DateTimeUTC">
    <xs:documentation>The message creation date-time, mandatory.</xs:documentation>
</xs:element>
<xs:element name="subject" type="xs:string">
    <xs:documentation>The message subject, mandatory.</xs:documentation>
</xs:element>
<xs:element name="body" type="xs:string">
    <xs:documentation>The message body, mandatory.</xs:documentation>
</xs:element>
<xs:element name="position" type="GM_Point" minOccurs="0" maxOccurs="1">
    <xs:documentation>Geographic point, optional.</xs:documentation>
</xs:element>
<xs:element name="area" type="GM_Surface" minOccurs="0" maxOccurs="1">
    <xs:documentation>Geographic area, optional.</xs:documentation>
</xs:element>
</xs:sequence>
</xs:complexType>

element textMessage/textMessageId

<table>
<thead>
<tr>
<th>diagram</th>
<th>textMessageId</th>
</tr>
</thead>
<tbody>
<tr>
<td>namespace</td>
<td><a href="http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd">http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd</a></td>
</tr>
<tr>
<td>type</td>
<td>textMessageURN</td>
</tr>
<tr>
<td>properties</td>
<td>content simple</td>
</tr>
<tr>
<td>facets</td>
<td>Kind Value Annotation</td>
</tr>
</tbody>
</table>
element `textMessage/informationObjectReferenceId`

```xml
<xs:element name="textMessageId" type="textMessageURN" minOccurs="1" maxOccurs="1">
  <xs:annotation>
    <xs:documentation>Identifier of the text message, mandatory.</xs:documentation>
  </xs:annotation>
</xs:element>
```

**Source**

```xml
<xs:element name="textMessageId" type="textMessageURN" minOccurs="1" maxOccurs="1">
  <xs:annotation>
    <xs:documentation>Identifier of the text message, mandatory.</xs:documentation>
  </xs:annotation>
</xs:element>
```

---

**element `textMessage/informationObjectReferenceType`**

```xml
<xs:element name="informationObjectReferenceId" type="xs:string" minOccurs="0" maxOccurs="1">
  <xs:annotation>
    <xs:documentation>A reference to an information object, optional.</xs:documentation>
  </xs:annotation>
</xs:element>
```

**Source**

```xml
<xs:element name="informationObjectReferenceId" type="xs:string" minOccurs="0" maxOccurs="1">
  <xs:annotation>
    <xs:documentation>A reference to an information object, optional.</xs:documentation>
  </xs:annotation>
</xs:element>
```

---

**element `textMessage/informationObjectReferenceType`**

```xml
<xs:element name="informationObjectReferenceType" type="informationObjectTypeEnum" minOccurs="0" maxOccurs="1">
  <xs:annotation>
    <xs:documentation>STM payload format reference, optional.</xs:documentation>
  </xs:annotation>
</xs:element>
```

**Source**

```xml
<xs:element name="informationObjectReferenceType" type="informationObjectTypeEnum" minOccurs="0" maxOccurs="1">
  <xs:annotation>
    <xs:documentation>STM payload format reference, optional.</xs:documentation>
  </xs:annotation>
</xs:element>
```
element `textMessage/validityPeriodStart`

- **Diagram**: [Diagram](#)
- **Namespace**: `http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd`
- **Type**: `DateTimeUTC`
- **Properties**:
  - minOccurs: 0
  - maxOccurs: 1
  - content: simple
- **Facets**:
  - Kind: Value
  - Annotation: pattern: `.*Z`
- **Annotation**: `documentation` - Start of validity period in ISO 8601 format, optional.
- **Source**:
  ```xml
  <xs:element name="validityPeriodStart" type="DateTimeUTC" minOccurs="0" maxOccurs="1">
    <xs:annotation>
      <xs:documentation>
        Start of validity period in ISO 8601 format, optional.
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  ```

element `textMessage/validityPeriodStop`

- **Diagram**: [Diagram](#)
- **Namespace**: `http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd`
- **Type**: `DateTimeUTC`
- **Properties**:
  - minOccurs: 0
  - maxOccurs: 1
  - content: simple
- **Facets**:
  - Kind: Value
  - Annotation: pattern: `.*Z`
- **Annotation**: `documentation` - Stop of validity period in ISO 8601 format, optional.
- **Source**:
  ```xml
  <xs:element name="validityPeriodStop" type="DateTimeUTC" minOccurs="0" maxOccurs="1">
    <xs:annotation>
      <xs:documentation>
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  ```
**element textMessage/author**

- **diagram**
- **namespace** http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd
- **type** xs:string
- **properties** content simple
- **annotation documentation**
The message author, mandatory.

```xml
<xs:element name="author" type="xs:string" maxOccurs="1">
  <xs:annotation>
    <xs:documentation> The message author, mandatory. </xs:documentation>
  </xs:annotation>
</xs:element>
```

**element textMessage/from**

- **diagram**
- **namespace** http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd
- **type** xs:string
- **properties** content simple
- **annotation documentation**
The sending actor, mandatory.

```xml
<xs:element name="from" type="xs:string" maxOccurs="1">
  <xs:annotation>
    <xs:documentation> The sending actor, mandatory. </xs:documentation>
  </xs:annotation>
</xs:element>
```

**element textMessage/serviceType**

- **diagram**
- **namespace** http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd
- **type** xs:string
- **properties**
  - minOcc 0
  - maxOcc 1
  - content simple

```xml
<xs:element> Stop of validity period in ISO 8601 format, optional. </xs:element>
```
The service type of the sender, optional.

```xml
<xs:element name="serviceType" type="xs:string" minOccurs="0" maxOccurs="1">
  <xs:documentation>The service type of the sender, optional.</xs:documentation>
</xs:element>
```

element `textMessage/createdAt`

```xml
<xs:element name="createdAt" type="DateTimeUTC">
  <xs:annotation>
    <xs:documentation>The message creation dateTime, mandatory.</xs:documentation>
  </xs:annotation>
</xs:element>
```

element `textMessage/subject`

```xml
<xs:element name="subject" type="xs:string">
  <xs:annotation>
    <xs:documentation>The message subject, mandatory.</xs:documentation>
  </xs:annotation>
</xs:element>
```

element `textMessage/body`

```xml
```
namespace http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd
type xs:string
properties content simple
annotation documentation The message body, mandatory.
source
<xs:element name="body" type="xs:string">
  <xs:annotation>
    <xs:documentation>The message body, mandatory.</xs:documentation>
  </xs:annotation>
</xs:element>

element textMessage/position
diagram

namespace http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd
type GM_Point
properties minOcc 0 maxOcc 1 content complex
attributes Name Type Use Default Fixed Annotation
documentation
lat LatitudeType required
lon LongitudeType required
documentation Geographic point, optional.
source
<xs:element name="position" type="GM_Point" minOcc="0" maxOcc="1">
  <xs:annotation>
    <xs:documentation>Geographic point, optional.</xs:documentation>
  </xs:annotation>
</xs:element>
element textMessage/area
diagram

namespace http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd
type GM_Surface
properties minOcc 0 maxOcc 1 content complex
children  Polygon Circle

annotation documentation
Geographic area, optional.

source
<xs:element name="area" type="GM_Surface" minOccurs="0" maxOccurs="1">
  <xs:annotation>
    <xs:documentation>Geographic area, optional.</xs:documentation>
  </xs:annotation>
</xs:element>

complexType GM_Point
diagram

namespace http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd
used by elements textMessage/position S100_CircleByCenterPointType/position

attributes
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>lat</td>
<td>LatitudeType</td>
<td>required</td>
<td></td>
<td></td>
<td>Latitude in degrees.</td>
</tr>
<tr>
<td>lon</td>
<td>LongitudeType</td>
<td>required</td>
<td></td>
<td></td>
<td>Longitude in degrees.</td>
</tr>
</tbody>
</table>

source
<xs:complexType name="GM_Point">
  <xs:attribute name="lat" type="LatitudeType" use="required">
    <xs:annotation>
      <xs:documentation>Latitude in degrees.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="lon" type="LongitudeType" use="required">
    <xs:annotation>
      <xs:documentation>Longitude in degrees.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>

attribute GM_Point/@lat
type LatitudeType

properties use required

facets
<table>
<thead>
<tr>
<th>Kind</th>
<th>Value</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>minInclusive</td>
<td>-90.0</td>
<td></td>
</tr>
<tr>
<td>maxInclusive</td>
<td>90.0</td>
<td></td>
</tr>
</tbody>
</table>

annotation documentation Latitude in degrees.

source
<xs:attribute name="lat" type="LatitudeType" use="required">
  <xs:annotation>
    <xs:documentation>Latitude in degrees.</xs:documentation>
  </xs:annotation>
</xs:attribute>
attribute GM_Point/@lon

type LongitudeType

<table>
<thead>
<tr>
<th>properties</th>
<th>use</th>
<th>required</th>
</tr>
</thead>
</table>

facets

<table>
<thead>
<tr>
<th>Kind</th>
<th>Value</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>minInclusive</td>
<td>-180.0</td>
<td></td>
</tr>
<tr>
<td>maxInclusive</td>
<td>180.0</td>
<td></td>
</tr>
</tbody>
</table>

annotation

documentation Longitude in degrees.

source

```xml
<xs:attribute name="lon" type="LongitudeType" use="required">
  <xs:annotation>
    <xs:documentation>Longitude in degrees.</xs:documentation>
  </xs:annotation>
</xs:attribute>
```

complexType GM_Surface

```
<xs:complexType name="GM_Surface">
  <xs:sequence>
    <xs:element name="Polygon" minOccurs="0"> 
    </xs:complexType>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="posList" type="xs:string">
          <xs:annotation>
            <xs:documentation>
              posList instances (and other instances with the content model specified by DirectPositionListType) hold the coordinates for a sequence of direct positions within the same coordinate reference system (CRS). If no srsName attribute is given, the CRS shall be specified as part of the larger context this geometry element is part of, typically a geometric object like a point, curve, etc. The optional attribute count specifies the number of direct positions in the list. If the attribute count is present then the attribute srsDimension shall be present, too. The number of entries in the list is equal to the product
```
of the dimensionality of
the coordinate reference system (i.e. it is a derived value
of the coordinate
reference system definition) and the number of direct
positions.

The attribute gml:id supports provision of a handle for the
XML element
representing a GML Object. Its use is optional for all GML
objects.

The number of direct positions in the list

Positive integer

Any URI
element **GM_Surface/Polygon**

<table>
<thead>
<tr>
<th>Properties</th>
<th>minOcc</th>
<th>maxOcc</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>complex</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Children</th>
<th>posList</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>id</td>
<td>xs:string</td>
<td></td>
<td></td>
<td></td>
<td>The attribute gml:id supports provision of a handle for the XML element representing a GML Object. Its use is optional for all GML objects.</td>
</tr>
<tr>
<td></td>
<td>count</td>
<td>xs:unsignedInt</td>
<td></td>
<td></td>
<td></td>
<td>The number of direct positions in the list</td>
</tr>
</tbody>
</table>

namespace http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd
<table>
<thead>
<tr>
<th>srsDimension</th>
<th>xs:unsignedShort</th>
<th>2</th>
<th>documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>srsName</td>
<td>xs:string</td>
<td></td>
<td>documentation</td>
</tr>
<tr>
<td>source</td>
<td></td>
<td></td>
<td>Any URI</td>
</tr>
</tbody>
</table>

```
<xs:element name="Polygon" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="posList" type="xs:string">
        <xs:annotation>
          <xs:documentation>
            posList instances (and other instances with the content model specified by DirectPositionListType) hold the coordinates for a sequence of direct positions within the same coordinate reference system (CRS). If no srsName attribute is given, the CRS shall be specified as part of the larger context this geometry element is part of, typically a geometric object like a point, curve, etc. The optional attribute count specifies the number of direct positions in the list. If the attribute count is present then the attribute srsDimension shall be present, too. The number of entries in the list is equal to the product of the dimensionality of the coordinate reference system (i.e. it is a derived value of the coordinate reference system definition) and the number of direct positions.
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string">
      <xs:annotation>
        <xs:documentation>
          The attribute gml:id supports provision of a handle for the XML element representing a GML Object. Its use is optional for all GML objects.
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="count" type="xs:unsignedInt">
      <xs:annotation>

```
<xs:documentation>
The number of direct positions in the list
</xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute name="srsDimension" type="xs:unsignedShort" default="2">
  <xs:annotation>
    <xs:documentation>
      Positive integer
    </xs:documentation>
  </xs:annotation>
</xs:attribute>
<xs:attribute name="srsName" type="xs:string">
  <xs:annotation>
    <xs:documentation>
      Any URI
    </xs:documentation>
  </xs:annotation>
</xs:attribute>
</xs:complexType>

attribute GM_Surface/Polygon/@id

type | xs:string
---|---
annotation | documentation
The attribute gml:id supports provision of a handle for the XML element representing a GML Object. Its use is optional for all GML objects.
source | 
<xs:attribute name="id" type="xs:string">
  <xs:annotation>
    <xs:documentation>
      The attribute gml:id supports provision of a handle for the XML element representing a GML Object. Its use is optional for all GML objects.
    </xs:documentation>
  </xs:annotation>
</xs:attribute>

attribute GM_Surface/Polygon/@count

type | xs:unsignedInt
---|---
annotation | documentation
The number of direct positions in the list
source | 
<xs:attribute name="count" type="xs:unsignedInt">
  <xs:annotation>
  </xs:annotation>
</xs:attribute>
The number of direct positions in the list
</xs:documentation>
</xs:annotation>
</xs:attribute>

<table>
<thead>
<tr>
<th>attribute</th>
<th>GM_Surface/Polygon/@srsDimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>xs:unsignedShort</td>
</tr>
<tr>
<td>properties</td>
<td>default 2</td>
</tr>
<tr>
<td>annotation</td>
<td>documentation Positive integer</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:attribute name=&quot;srsDimension&quot; type=&quot;xs:unsignedShort&quot; default=&quot;2&quot;&gt; <a href="">xs:annotation</a> <a href="">xs:documentation</a> Positive integer &lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>attribute</th>
<th>GM_Surface/Polygon/@srsName</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>xs:string</td>
</tr>
<tr>
<td>annotation</td>
<td>documentation Any URI</td>
</tr>
<tr>
<td>source</td>
<td>&lt;xs:attribute name=&quot;srsName&quot; type=&quot;xs:string&quot;&gt; <a href="">xs:annotation</a> <a href="">xs:documentation</a> Any URI &lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</td>
</tr>
</tbody>
</table>
element GM_Surface/Polygon/posList

posList instances (and other instances with the content model specified by DirectPositionListType) hold the coordinates for a sequence of direct positions within the same coordinate reference system (CRS). If no srsName attribute is given, the CRS shall be specified as part of the larger context this geometry element is part of, typically a geometric object like a point, curve, etc. The optional attribute count specifies the number of direct positions in the list. If the attribute count is present then the attribute srsDimension shall be present, too. The number of entries in the list is equal to the product of the dimensionality of the coordinate reference system (i.e. it is a derived value of the coordinate reference system definition) and the number of direct positions.
element GM_Surface/Circle

namespace http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd
type S100_CircleByCenterPointType

properties
- minOcc: 0
- maxOcc: 1
- content: complex

children
- position
- radius

attributes
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>xs:string</td>
<td></td>
<td></td>
<td></td>
<td>The attribute gml:id supports provision of a handle for the XML element representing a GML Object. Its use is optional for all GML objects.</td>
</tr>
</tbody>
</table>

source
<xs:element name="Circle" type="S100_CircleByCenterPointType" minOccs="0"/>
complexType S100_CircleByCenterPointType

diagram

namespace http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd

children
  position
  radius

used by
  element GM_Surface/Circle

attributes
  Name | Type        | Use | Default | Fixed | Annotation documentation
  ---  | ----------- | --- | ------- | ----- | -----------------------------
  id   | xs:string  |     | Fixed   |       | The attribute gml:id supports provision of a handle for the XML element representing a GML Object. Its use is optional for all GML objects.

annotation
documentation
Type for S-100 arc by center point geometry using interpolation circularArcCenterPointWithRadius

source
<xs:complexType name="S100_CircleByCenterPointType">
  <xs:documentation>Type for S-100 arc by center point geometry using interpolation circularArcCenterPointWithRadius</xs:documentation>
  <xs:annotation>
    <xs:documentation>The radius is a double greater than zero and its unit is assumed to be nautical miles (nm). </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="position" type="GM_Point"/>
    <xs:element name="radius" type="positiveDouble" />
    <xs:annotation>
      <xs:documentation>The radius is a double greater than zero and its unit is assumed to be nautical miles (nm). </xs:documentation>
    </xs:annotation>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string">
    <xs:annotation>
      <xs:documentation>The attribute gml:id supports provision of a handle for the XML element representing a GML Object. Its use is optional for all GML objects. </xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
attribute `S100_CircleByCenterPointType/@id`

<table>
<thead>
<tr>
<th>Type</th>
<th>xs:string</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotation</td>
<td>documentation</td>
</tr>
<tr>
<td></td>
<td>The attribute gml:id supports provision of a handle for the XML element representing a GML Object. Its use is optional for all GML objects.</td>
</tr>
</tbody>
</table>

source

```
<xs:attribute name="id" type="xs:string">
  <xs:annotation>
    <xs:documentation>
      The attribute gml:id supports provision of a handle for the XML element representing a GML Object. Its use is optional for all GML objects.
    </xs:documentation>
  </xs:annotation>
</xs:attribute>
```

element `S100_CircleByCenterPointType/position`

<table>
<thead>
<tr>
<th>Diagram</th>
<th></th>
</tr>
</thead>
</table>

namespace http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd

type `GM_Point`

<table>
<thead>
<tr>
<th>Properties</th>
<th>content</th>
<th>complex</th>
</tr>
</thead>
</table>

attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>lat</td>
<td>LatitudeType</td>
<td>required</td>
<td></td>
<td></td>
<td>documentation</td>
</tr>
<tr>
<td>lon</td>
<td>LongitudeType</td>
<td>required</td>
<td></td>
<td></td>
<td>documentation</td>
</tr>
</tbody>
</table>

source

```
<xs:element name="position" type="GM_Point"/>
```

element `S100_CircleByCenterPointType/radius`

<table>
<thead>
<tr>
<th>Diagram</th>
<th></th>
</tr>
</thead>
</table>

namespace http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd

type `positiveDouble`

<table>
<thead>
<tr>
<th>Properties</th>
<th>content</th>
<th>simple</th>
</tr>
</thead>
</table>
The radius is a double greater than zero and its unit is assumed to be nautical miles (nm).

```
<xs:element name="radius" type="positiveDouble">
  <xs:annotation>
    <xs:documentation>The radius is a double greater than zero and its unit is assumed to be nautical miles (nm).</xs:documentation>
  </xs:annotation>
</xs:element>
```

**simpleType BearingType**

- **namespace**: http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd
- **type**: restriction of `xs:decimal`
- **properties**: base `xs:decimal`

```
<xs:simpleType name="BearingType">
  <xs:restriction base="xs:decimal">
    <xs:fractionDigits value="1"/>
    <xs:minInclusive value="0.0"/>
    <xs:maxInclusive value="360.0"/>
  </xs:restriction>
</xs:simpleType>
```

**simpleType DateTimeUTC**

- **namespace**: http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd
- **type**: restriction of `xs:dateTime`
- **properties**: base `xs:dateTime`
- **used by**: `textMessage/createdAt`, `textMessage/validityPeriodStart`, `textMessage/validityPeriodStop`

```
<xs:simpleType name="DateTimeUTC">
  <xs:annotation>
    <xs:documentation>UTC time.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:dateTime">
    <xs:pattern value=".*Z"/>
  </xs:restriction>
</xs:simpleType>
```
### simpleType `informationObjectTypeEnum`

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>namespace</td>
<td><a href="http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd">http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd</a></td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>restriction of xs:string</td>
<td></td>
</tr>
<tr>
<td>properties</td>
<td>base xs:string</td>
<td></td>
</tr>
<tr>
<td>used by</td>
<td>element textMessage/informationObjectReferenceType</td>
<td></td>
</tr>
<tr>
<td>facets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kind</td>
<td>Value</td>
<td>Annotation</td>
</tr>
<tr>
<td>enumeration</td>
<td>RTZ</td>
<td></td>
</tr>
<tr>
<td>enumeration</td>
<td>S124</td>
<td></td>
</tr>
<tr>
<td>enumeration</td>
<td>TXT</td>
<td></td>
</tr>
</tbody>
</table>

### simpleType `LatitudeType`

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>namespace</td>
<td><a href="http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd">http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd</a></td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>restriction of xs:decimal</td>
<td></td>
</tr>
<tr>
<td>properties</td>
<td>base xs:decimal</td>
<td></td>
</tr>
<tr>
<td>used by</td>
<td>attribute GM_Point/@lat</td>
<td></td>
</tr>
<tr>
<td>facets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kind</td>
<td>Value</td>
<td>Annotation</td>
</tr>
<tr>
<td>minInclusive</td>
<td>90.0</td>
<td></td>
</tr>
<tr>
<td>maxInclusive</td>
<td>-90.0</td>
<td></td>
</tr>
<tr>
<td>annotation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>documentation</td>
<td></td>
<td>The latitude of the point. Decimal degrees, WGS84 datum.</td>
</tr>
</tbody>
</table>

### simpleType `LongitudeType`

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>namespace</td>
<td><a href="http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd">http://stmvalidation.eu/schemas/textMessageSchema_1_3.xsd</a></td>
<td></td>
</tr>
<tr>
<td>type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>properties</td>
<td></td>
<td></td>
</tr>
<tr>
<td>used by</td>
<td></td>
<td></td>
</tr>
<tr>
<td>facets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kind</td>
<td>Value</td>
<td>Annotation</td>
</tr>
<tr>
<td>minInclusive</td>
<td>90.0</td>
<td></td>
</tr>
<tr>
<td>maxInclusive</td>
<td>-90.0</td>
<td></td>
</tr>
</tbody>
</table>
### simpleType longitudeType

<table>
<thead>
<tr>
<th>Kind</th>
<th>Value</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>minInclusive</td>
<td>-180.0</td>
<td></td>
</tr>
<tr>
<td>maxInclusive</td>
<td>180.0</td>
<td></td>
</tr>
</tbody>
</table>

The longitude of the point. Decimal degrees, WGS84 datum.

```
<x:simpleType name="LongitudeType">
  <xs:annotation>
    <xs:documentation>
      The longitude of the point. Decimal degrees, WGS84 datum.
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:decimal">
    <xs:minInclusive value="-180.0"/>
    <xs:maxInclusive value="180.0"/>
  </xs:restriction>
</xs:simpleType>
```

### simpleType positiveDouble

<table>
<thead>
<tr>
<th>Kind</th>
<th>Value</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>minExclusive</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

```
<x:simpleType name="positiveDouble">
  <xs:restriction base="xs:double">
    <xs:minExclusive value="0"/>
  </xs:restriction>
</xs:simpleType>
```

### simpleType textMessageURN

<table>
<thead>
<tr>
<th>Kind</th>
<th>Value</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxLength</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>pattern</td>
<td>urn:mrn:stm:txt:[0-9a-zA-Z:+_-]+</td>
<td></td>
</tr>
</tbody>
</table>

```
<x:simpleType name="textMessageURN">
  <xs:restriction base="xs:string">
    <xs:maxLength value="120"/>
    <xs:pattern value="urn:mrn:stm:txt:[0-9a-zA-Z:+_\-]+"/>
  </xs:restriction>
</xs:simpleType>
```
<table>
<thead>
<tr>
<th>annotation</th>
<th>documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Text message Id in STM URN format.</td>
</tr>
</tbody>
</table>

```xml
<xs:simpleType name="textMessageURN">
  <xs:annotation>
    <xs:documentation>
      Text message Id in STM URN format.
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:maxLength value="120"/>
    <xs:pattern value="urn:mrn:stm:txt:[0-9a-zA-Z:+_-]+"/>
  </xs:restriction>
</xs:simpleType>
```
### 3 References

<table>
<thead>
<tr>
<th>Nr</th>
<th>Version</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.2</td>
<td>VIS Service Specification Documentation</td>
</tr>
</tbody>
</table>

### 4 Acronyms and Terminology

<table>
<thead>
<tr>
<th>Type</th>
<th>Term</th>
<th>Definition/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acronym</td>
<td>SeaSWIM</td>
<td>Sea System Wide Information Management</td>
</tr>
<tr>
<td>Acronym</td>
<td>UVID</td>
<td>Unique Voyage Plan Identity</td>
</tr>
<tr>
<td>Maritime</td>
<td>VIS</td>
<td>Voyage Information Service</td>
</tr>
<tr>
<td>Maritime</td>
<td>VP</td>
<td>Voyage Plan</td>
</tr>
<tr>
<td>Acronym</td>
<td>XML</td>
<td>Extendible Mark-up Language</td>
</tr>
<tr>
<td>Acronym</td>
<td>XSD</td>
<td>XML Schema Definition</td>
</tr>
</tbody>
</table>
5 Appendix A textMessage - example

<?xml version="1.0" encoding="UTF-8"?>
  <textMessageId>urn:mrn:stm:txt:sma:20170510104400-1</textMessageId>
  <informationObjectReferenceId>urn:mrn:stm:voyage:id:test:100</informationObjectReferenceId>
  <informationObjectReferenceType>RTZ</informationObjectReferenceType>
  <validityPeriodStart>2017-05-01T01:00:00Z</validityPeriodStart>
  <validityPeriodStop>2017-06-10T01:00:00Z</validityPeriodStop>
  <author>urn:mrn:stm:user:sma:mikolo</author>
  <from>urn:mrn:stm:org:sma</from>
  <serviceType>SHIP-VIS</serviceType>
  <createdAt>2017-05-10T01:00:00Z</createdAt>
  <subject>Test message</subject>
  <body>Test message Hanöbukten</body>
  <position lat="55.50668" lon="14.29825"/>
  <area>
    <Polygon>
    </Polygon>
    <Circle>
      <position lat="55.50668" lon="14.29825"/>
      <radius>0</radius>
    </Circle>
  </area>
</textMessage>
Using STM to increase BALTic Sea SAFEty

Making the Baltic Sea even safer by improving the situational awareness on ships and shore, building tools that automate work and provide decision support to prevent risk situations and accidents.

Making STM happen!

SAFETY - ENVIRONMENT - EFFICIENCY

Swedish Maritime Administration ◦ VTS Finland ◦ Estonian Maritime Administration ◦ Norwegian Coastal Administration ◦ RISE Research Institutes of Sweden ◦ DNV GL

www.stmbaltsafe.eu
www.stmvalidation.eu/projects/stmbaltsafe