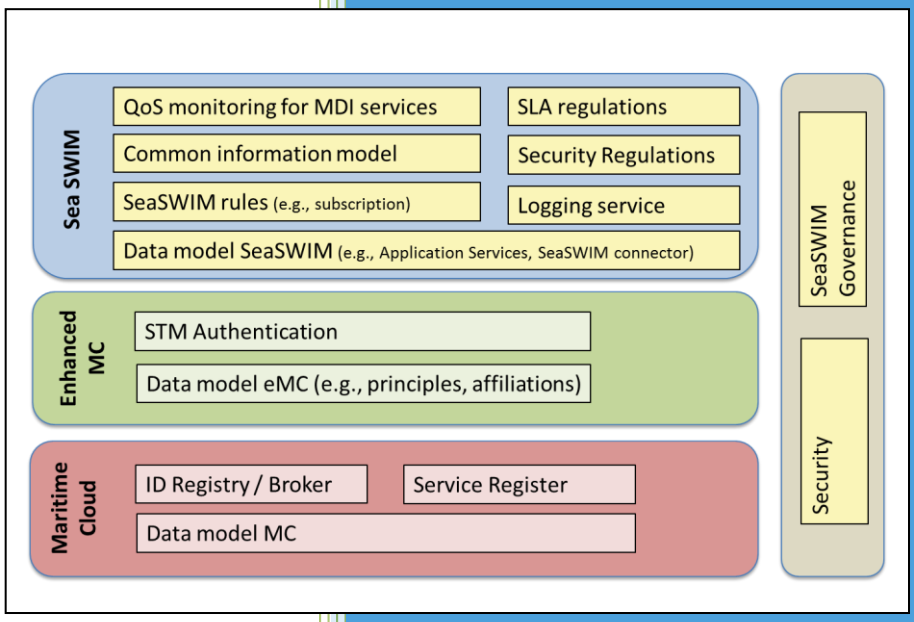
Document No: STM_Validation_D4.18
Title: Specification of MC (former: MCeNAV) enhancements to support SeaSWIM
Date: 2016-07-08



Co-financed by the Connecting Europe Facility of the European Union

Co-financed by the Connecting Europe Facility of the European Union

DOCUMENT STATUS

Authors

Name	Organisation
Oliver Norkus	University of Oldenburg
Cilli Sobiech	Viktoria Swedish ICT
Marius Brinkmann	University of Oldenburg
Sören Schweigert	OFFIS

Review

Name	Organisation
Mikael Olofsson	Combitech AB
Tomas G. Christensen	DMA
Anders Dalén	Viktoria Swedish ICT
Christoph Rihacek	Frequentis
Thomas Lutz	Frequentis

Approval

Name	Organisation	Signature	Date

Document History

Version	Date	Status	Initials	Description
V1	2016-03-24	Draft	MB	Draft version uploaded distributed for comments
V2	2016-06-27	Draft	ON, CS	Updated Draft version
V3	2016-07-01	Draft	ON	Draft with derived requirements distributed
V4	2016-07-04	Draft	CS	Draft with some resolved comments
V5	2016-07-07	Draft	CS, ON	Restructured report

V6	2016-07-08	Final	CS, ON	Final report with resolved comments created

TEN-T PROJECT NO: 2014-EU-TM-0206-S

Table of contents

1 Introduction

Two of the largest projects focusing on developing and utilizing; Sea Traffic Management Validation (STM: 2015-2018) and EfficienSea 2 (E2: 2015-2018) aim at utilization of a common maritime digital infrastructure. As the two projects emphasize different aspects of potential improvements to the maritime industry, their underlying digital infrastructure needs inevitably to complement one another in their technical implementation. The two projects are cooperating closely and continuously to ensure that this leads to complementary synergies and avoids redundant work in the development.

For the beginning, we shortly explain the relation/mapping between the necessary functional components to implement and validate STM and the two concepts; Sea System Wide Information Management (SeaSWIM) and Maritime Cloud (MC). The complementing needs of the projects will be handled on an, individual functional component basis, which means that the two projects will align and diverge to best suit the services they are supporting. The aim is not to produce competing infrastructures, on the contrary, the intention is to exchange best practices between the concepts, to harmonize standards and avoid unnecessary duplicating efforts. This mapping document is therefore substantial to facilitate the connection between the projects.

This report is the result of the work of Sub-Activity 4.2 “Apply and adapt the MC for SeaSWIM STM”. Its objective is to enhance, deploy and maintain MC as SeaSWIM communication infrastructure. In this document we derive requirements that enable SeaSWIM applications and services to support all aspects of the SeaSWIM concept.

The relation between the Core Maritime Cloud and the SeaSWIM layer is called the Enhanced MC. The identification and analysis of components and tasks of the Enhanced MC are the subject of this report. Therefore, first, the requirements and use cases are identified and then, second, they are checked against the provided MC functions. As third part, a gap analysis shows which requirements are fulfilled by standard MC functions and which requirements needs further observance.

Further information and background were provided in internal reports for the project.

2 Layering and Architecture Overview

The general layers of SeaSWIM, Enhanced MC, Core MC, pictured in Figure 1, should not be confused with the technical implementation of a Maritime Digital Infrastructure. Contrary to the shown monolithic structure, the technical implementation intends to use a service-oriented approach where individual modules are specified to make up the support infrastructure.

The division in Figure 1 is driven by organizational aspects. The overview shows two project specific areas: SeaSWIM (STM) and Core MC (E2). The connection between the two projects is called Enhanced MC and means the development effort needed in the STM project to adapt the Core MC solutions for the Maritime Digital Infrastructure instance required for the STM Validation.

These layers are shortly introduced in the following sub-chapters.

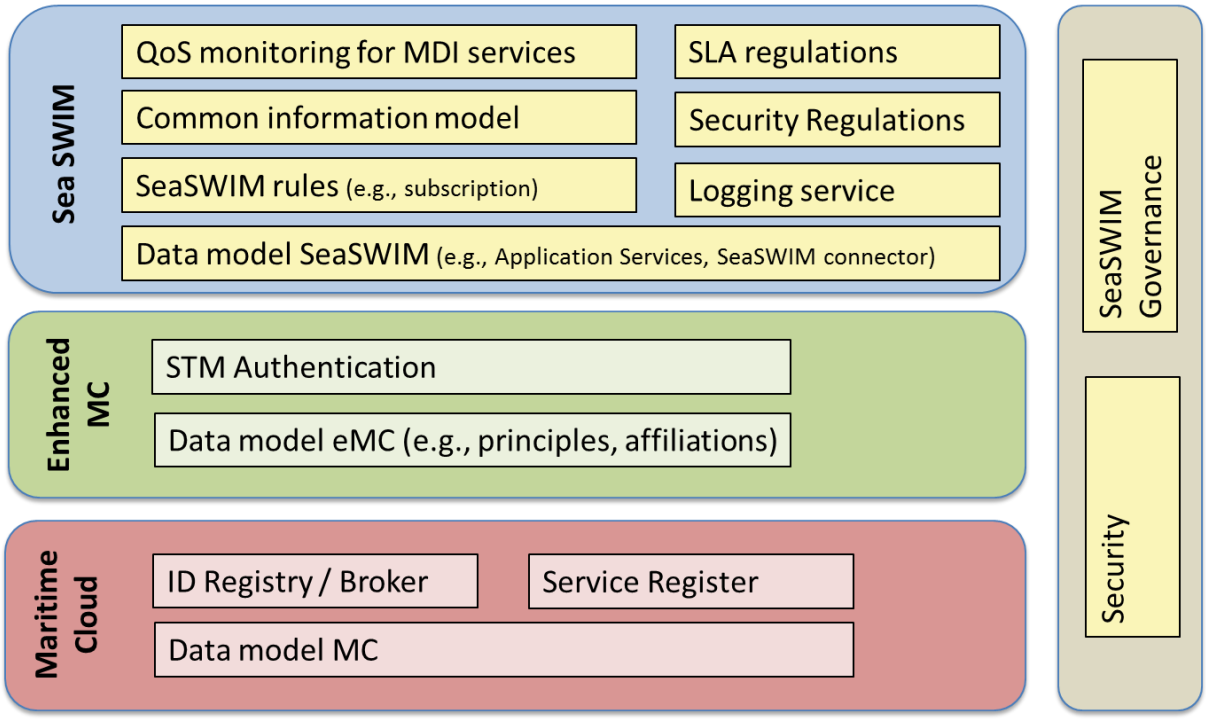


Figure 1: Layering of the Service Landscape

2.1 Maritime Cloud

The Maritime Cloud is a key part of the EfficienSea2 project. E2 plans to lay the foundation for a widely used framework facilitating interoperable solutions. In a first stage, it is a framework providing standardized protocol and functional support for identity and role management, authentication, encryption, and service discovery. This enables easy development of innovative solutions for maritime end users in a context of global interoperability. The Maritime Cloud shall be regarded much like the Internet as the enabler of interoperable systems for email, VoIP, webpages, blogs, social networks, or online shopping sites.

Services themselves and the service-based economy are a central part of the vision of the Maritime Cloud and explicitly include services that do not solely rely on machine to

machine communication such as services delivered over telephone calls (voice or fax), email, websites, Navtex and other “primitive” solutions.

The **Service Registry** is a central component of the MC and allows to:

- Store service specifications according to an envisioned Service Specification Standard and provisioned service instances implemented according to these service specifications
- Improve the visibility and accessibility of available maritime information and services
- Share a common view on service standards and provisioned services between service providers, consumers, and regulatory authorities
- Have a single reference point for provisioning and discovery by comprising all maritime services, not only digital services, the information they carry, and the technical means to obtain it
- Manage the life cycle of service specifications and service instances
- Implement the Maritime Service Portfolio (MSP) concept by providing a repository for the specification of operational and technical services and provisioned service instances.

Together with the concept of Identity Management, E2 envisions to enable service providers to deliver their services to customers with increased security and productivity while decreasing the cost and effort. Identity Management deals with identifying individuals such as users, devices and computer systems, and controlling their access to resources within some kind of organizational context, e.g. private company, country or a whole industry. To overcome the lack of a global digital identity for service users, vessels and systems and to require confidentiality and integrity the **Maritime Identity Registry** concept has been proposed that allows to:

- Manage the identities of all organizations, users, devices, and systems that need to communicate with each other in some way, not necessarily via the MC services
- Integrate with any form of external information carrier using standard security protocols
- Manage human users with a single identity (password/certificate, later also fingerprints, mobiles, etc.) for authentication that can be used across all maritime systems
- Digitally sign messages and documents by human users
- Provide digital certificates and support authentication for user and systems in the MC to allow machine to machine communication
- Integrate with companies’ existing identity management systems in order to “reuse” the identities already set up internally in a company (federated identity management)
- Further provide (strict) guidelines how service providers should handle information about client´s shared information (e.g. a vessel's position)

2.2 Enhanced MC

The relation between the Core Maritime Cloud and the SeaSWIM layer is called the Enhanced MC. The identification and analysis of components and tasks of the

Enhanced MC are the subject of this report. Therefore, first, the requirements and use cases are identified and then, second, they are checked against the provided MC functions. As third part, a gap analysis shows which requirements are fulfilled by standard MC functions and which requirements needs further observance.

2.3 Sea System Wide Information Management (SeaSWIM)

SeaSWIM (System-wide Information Management) consists of various components that will ensure interoperability of the STM services by facilitating data sharing in a common information environment and structure. Hereby, STM aims at overcoming many of the challenges of communication and information sharing between stakeholders in the maritime transport industry. Activity 4 of the STM Validation Project aims at providing a Maritime Service Infrastructure based on the SeaSWIM principles, defined in MONALISA 2.0 project, and will provide the basis for the testbeds being realized in Activity 1 and Activity 2 of the STM Validation Project. The Maritime Cloud is used as the infrastructure for information exchange/sharing in the STM testbeds, i.e. for the validation of SeaSWIM.

A user of STM can either provide or use application services (also called parental services; e.g. enhanced monitoring, route optimization service) that rely on support services (e.g. a logging service). Yet, more than just the support services, the SeaSWIM infrastructure also consists of SeaSWIM rules, regulations, management services and governance structures.

In Sub-Activity 4.1 of STM Act 4, these rules, regulations, management services and governance structures will be defined, this involves:

- **Security regulations** - dealing with encryption of data, protection from unauthorized access in communications, etc.
- **Logging service** - for monitoring and management of service usage (patterns)
- **Quality-of-Service (QoS) monitoring** for MarSI services - dealing with consumption and quality of the services (quantitative/qualitative)
- **SeaSWIM Governance structure** - for capturing ownership, economic engagement, pricing model, risk profile of a service, etc.
- **Service level agreement (SLA) regulations** - e.g. formalised engagement and quality evaluation models
- **SeaSWIM rules** - e.g. interaction patterns, common information model

Furthermore, to enable safe and reliable communication and to ensure the control of data access, the **SeaSWIM connector (SSC)** has been introduced. The SeaSWIM connector is a piece of software hosted at each client's domain offering convenient and STM standardized procedures and templates for identity and access management, including authentication and authorization. Each communication unit is to be equipped with an SSC to ensure that when two different communication units interact, the SSCs speak on a technical level. Therefore, every client and/or every resource provider implicitly contains a SeaSWIM connector. And every SeaSWIM connector of a e.g. resource provider or client will need to implement a harmonized interface supporting establishing a session with another SeaSWIM connector, based on HTTPS.

The SSC thus acts as an interface to come in contact with other components, also the application services. The SSC is so to say “the collection of functions and design constraints that applies to every service- or client interface, to comply with the SeaSWIM specifications”. Therefore, the SeaSWIM connector is also developed to adhere to some important STM principles:

a. Only authenticated actors can provide and use STM services

b. Data creators are owners and have full control over the authorization process

c. STM strives after a service oriented and highly decentralized architecture

d. Usage of widely accepted industry standards wherever these exist

Furthermore, the SeaSWIM connector offers standardized means of defining communication end-points (API) and ensures encryption of all data transferred between the end-points.

2.4 Relevant technical decisions

For the identification of the enhancement of the MC a set of decisions are relevant to take into account. These are:

- No central or standardized authorization functions and no central or standardized usage of ACL (Access Control List). Implementation of authorization logic is left to each application/parent service to resolve. Obviously, authentication and access control is still supported by functionality such as id registry (with certificates) and service registry. Even though no generalized Access Management is implemented in the Act2 testbeds, it is an important feature for the STM project which has to be considered further on in the project.
- Encryption and decryption is by SSL, which gives a sufficient level of security for the testbeds, no other data encryption is needed at this stage.
- All service interactions run over SSC, no other “way in or out” to parent services or applications by being STM compliant
- If subscription is needed, the parental service will have to add this functionality as needed. The specific requirements and potential implementation alternatives for this support services are currently unclear.
- SSCs in the test-bed will not support subscription features. If necessary, this will need to be managed by each parent service.
- Schema validation is done by the parent service, not in the SSCs.

3 STM Use Cases and Requirements

The following list of use-cases and requirements have been taken from the document (1).

Thereby we considered only those functional use cases and requirements which affect the surroundings of ID and service registry. Requirements relating to the security, governance and quality of service, are left out deliberately. Details and a complete list of use cases and requirements is in document (1).

3.1 Use Cases

UC_ID	Name	Description
STM_UC#1	Log in	User authenticated; Session opened; Session ID assigned
STM_UC#1.1	Log out	Log out, close session
STM_UC#2	Search a service	Based on the parameters of a service, the service can be searched; the user matching to the search is displayed
STM_UC#3	Get all services provided by a Service Provider	The services offered by a specific provider be determined and displayed
STM_UC#4	Ask for being authorized	Authenticated user wishes to use a service. He has already found this service (e.g., via UC#2 or UC#3) is currently not allowed to read the data objects due to a lack of appropriate permissions.
STM_UC#5	Subscribe a service	User is a registered subscriber of this service; this service sends (pushes) new information automatically to the user
STM_UC#6	Use a service	see further details in STMVal_D.17 Service Demand STM Services
STM_UC#7	Register a new service	The information (the new service) should be available and discoverable after uploading
STM_UC#8	Manage visibility in the Service Registry	The discoverability of a (new) service shall be managed and configured
STM_UC#9	Update a service	Change meta information of a service
STM_UC#10	Authorize collaborator	Give access for a user to read data objects provided by the service
STM_UC#11	De-authorize collaborator	The user who requested the read access can now read this information (use the service)

3.2 Requirements

The requirements are separated regarding their concerning of ID and Service Registry. See further in STMVal_D4.17 Service Demand STM Services.

3.2.1 Identity management and role-based access control

R_ID	Requirement
R_ID_1	All types and attributes of identities must be identifiable.
R_ID_2	A Universal Identifier for all identities is needed. The UID concept must be flexible, decentralized, and forward compatible.
R_ID_3	An ID registry is needed, which can uniquely identify all identity entities and facilitate lookup of secondary identifying attributes.
R_ID_4	Identities shall be discoverable based on different criteria.
R_ID_5	A user-role concept is needed. A user can hold multiple roles.
R_ID_6	Standardized functions for authentication of identities and validation of authenticity is needed.
R_ID_7	Standardized functions for authorization is needed.
R_ID_8	Ownership of information elements and authorization to pass it on, must be managed

3.2.2 Service definition, discoverability and interactions

R_ID	Requirement
R_SR_1	The system shall provide a Service Registry and a look up function.
R_SR_2	Services should be grouped together (Cluster). For instance, clusters can be platform specific, add-on services or a geographical subset.
R_SR_3	Different types of service interactions shall be implemented.
R_SR_4	Services shall be discoverable based on different criteria.
R_SR_5	The usage of services shall be restrictable (e.g. restricted access from mobile actors requiring global location services)
R_SR_6	Extensibility: also third-party developers should be able to provide their services as a part of service portfolio management.

4 Provided MC Functions

4.1 Functions in service registry API

The following functions are suggested for the Service Registry (SR) API alpha version to be delivered in October 2016 (the ones with high priority). On the left side is the method with parameters and on the right side the event based on the method call:

MC Funct. ID	Area	Prio.	Function
MC_SR_1	Service Specification	high	*createServiceSpecification(serviceSpecification) → Status*
MC_SR_2	Service Specification	high	*getServiceSpecifications(organisationId) → list of service specifications including some metadata, e.g., lastupdated, lastupdatedby, etc.*
MC_SR_3	Service Specification	high	*getServiceSpecifications(keywords) → list of service specifications*
MC_SR_4	Service Specification	high	*readServiceSpecification(serviceSpecificationId, versionID) → ServiceSpecification (including human readable documents)*
MC_SR_5	Service Specification	high	*updateServiceSpecification(serviceSpecification) → Updated ServiceSpecification*
MC_SR_6	Service Specification	high	*deleteServiceSpecification(serviceSpecificationId, versionID) → Deleted ServiceSpecification (logical deletion)*
MC_SR_7	Service Specification	high	*deprecateServiceSpecification(serviceSpecificationId , versionID) → Deprecated ServiceSpecification*
MC_SR_8	Service Specification	low	*endorseServiceSpecification(serviceSpecificationId, endorsingOrganizationId) → status*
MC_SR_9	Service Specification	low	*revokeEndorsementServiceSpecification(serviceSpecificationId, endorsingOrganizationId) → status*
MC_SR_10	Service Specification	low	*getServiceSpecificationsEndorsedBy(organisationId) → list of service specifications*

MC_SR_11	Service Technical Design	high	*createServiceTechnical(serviceTechnicalDesign) → Status*
MC_SR_12	Service Technical Design	high	*getServiceTechnicals(organisationId | serviceSpecificationId) → list of service*
MC_SR_13	Service Technical Design	high	*Technicals Designs, including some metadata, e.g., lastupdated, lastupdatedby, etc.*
MC_SR_14	Service Technical Design	high	*readServiceTechnical(serviceTechnicalDesignId, versionID) → ServiceTechnicalDesign (including human readable documents)*
MC_SR_15	Service Technical Design	high	*updateServiceTechnical(serviceTechnicalDesign) → Updated ServiceTechnicalDesign*
MC_SR_16	Service Technical Design	high	*deleteServiceTechnical(serviceTechnicalDesignId) → Deleted ServiceTechnicalDesign*
MC_SR_17	Service Technical Design	high	*deprecateServiceTechnical(serviceTechnicalDesignId, versionID) → Deprecated ServiceTechnicalDesign*
MC_SR_18	Service Technical Design	low	*endorseServiceTechnical(serviceTechnicalDesignId, endorsingOrganizationId) → status*
MC_SR_19	Service Technical Design	low	*revokeEndorsementServiceTechnical(serviceTechnicalDesignId, endorsingOrganizationId) → status*
MC_SR_20	Service Technical Design	low	*getServiceTechnicalsEndorsedBy(organisationId, [serviceSpecificationId]) → list of service Technical Designs*
MC_SR_21	Service Instance	high	*getServiceInstances(<[serviceSpecificationId]|[technicalDesignId]>) → list of service Instances*
MC_SR_22	Service Instance	high	*getServiceInstances(serviceSpecificationId, technicalDesignId, simpleOGCgeometrytypes) → list of service Instances*
MC_SR_23	Service Instance	high	*createServiceInstance(serviceInstance) → Status*
MC_SR_24	Service Instance	high	*readServiceInstance(serviceInstanceId, versionID) → ServiceInstance*

MC_SR_25	Service Instance	high	*updateServiceInstance(serviceInstance) → Updated ServiceInstance*
MC_SR_26	Service Instance	high	*deleteServiceInstance(serviceInstanceId, versionID) → Deleted ServiceInstance*
MC_SR_27	Service Instance	low	*endorseServiceInstance(serviceInstanceId, endorsingOrganizationId) → status*
MC_SR_28	Service Instance	low	*revokeEndorsementServiceInstance(serviceInstanceId, endorsingOrganizationId) → status*
MC_SR_29	Service Instance	low	*getServiceInstancesEndorsedBy(organisationId, [serviceSpecificationId]) → list of service Instances*

4.2 Functions in identity registry API

The following functions are suggested for the ID Registry API alpha version to be delivered in October 2016.

There are also functions that are provided by the ID brokers. These are out-of-the-box ID functions that are provided with Open ID or Keycloak and affecting Log In and Log Out, and the checking of identity (for details see: http://developers.maritimecloud.net/identity/index.html#openid-connect-authentication-flow)

Furthermore, there is in addition to the identities of the ID registry also the concept of Maritime Resource Names (MRN), so that all entities can be clearly identified.

So, here only the ID Registry functions, going beyond the Open Id and Keycloak functions are mentioned. On the left side is the method with parameters and on the right side the event based on the method call:

MC Funct. ID	**Area**	**Prio.**	**Function**
MC_ID_1	Organizations	high	*apply(organizationDetails) → status*
MC_ID_2	Organizations	high	*readOrganization(organizationId) → organizationDetails*
MC_ID_3	Organizations	high	*updateOrganization(organizationDetails) → status*
MC_ID_4	Organizations	high	*getOrganizations() → list of all organizations*
MC_ID_5	Services	high	*getServices(organizationId) → list of service for organization*

MC_ID_6	Services	high	*createService(serviceDetails) → status*
MC_ID_7	Services	high	*readService(serviceId) → serviceDetails*
MC_ID_8	Services	high	*updateService(serviceDetails) → status*
MC_ID_9	Services	high	*deleteService(serviceId) → status*
MC_ID_10	Services	high	*createServiceCertificate(serviceId) → certificate and public and private keys*
MC_ID_11	Services	high	*revokeServiceCertificate(serviceId, CertificateId, revokeReason) → status*
MC_ID_12	Vessels	high	*getVessels(organizationId) → list of vessel for organization*
MC_ID_13	Vessels	high	*createVessel(vesselDetails) → status*
MC_ID_14	Vessels	high	*readVessel(vesselId) → vesselDetails*
MC_ID_15	Vessels	high	*updateVessel(vesselDetails) → status*
MC_ID_16	Vessels	high	*deleteVessel(vesselId) → status*
MC_ID_17	Vessels	high	*createVesselCertificate(vesselId) → certificate and public and private keys*
MC_ID_18	Vessels	high	*revokeVesselCertificate(vesselId, CertificateId, revokeReason) → status*
MC_ID_19	Users	high	*getUsers(organizationId) → list of user for organization*
MC_ID_20	Users	high	*createUser(userDetails) → status*
MC_ID_21	Users	high	*readUser(userId) → userDetails*
MC_ID_22	Users	high	*updateUser(userDetails) → status*
MC_ID_23	Users	high	*deleteUser(userId) → status*
MC_ID_24	Users	high	*createUserCertificate(userId) → certificate and public and private keys*
MC_ID_25	Users	high	*revokeUserCertificate(userId, CertificateId, revokeReason) → status*
MC_ID_26	Devices	high	*getDevices(organizationId) → list of device for organization*
MC_ID_27	Devices	high	*createDevice(deviceDetails) → status*

MC_ID_28	Devices	high	*readDevice(deviceId) → deviceDetails*
MC_ID_29	Devices	high	*updateDevice(deviceDetails) → status*
MC_ID_30	Devices	high	*deleteDevice(deviceId) → status*
MC_ID_31	Devices	high	*createDeviceCertificate(deviceId) → certificate and public and private keys*
MC_ID_32	Devices	high	*revokeDeviceCertificate(deviceId, CertificateId, revokeReason) → status*
MC_ID_33	Certificates	high	*getRevokedCertificates() → list of revoked certificates*
MC_ID_34	Certificates	high	*verifyCertificate(certificateId) → certificate status*

5 Enhancements of the MC

The relation between the two projects is called Enhanced MC [Figure 1] and means the development effort needed in the STM project to adapt the MC solutions for the Maritime Digital Infrastructure instance needed for the STM Validation.

For example, to prevent that each resource provider in SeaSWIM needs to implement support for a workflow for validated identification of clients or users accessing their resource, the STM project will provide an **Identity Registry** in collaboration with the EfficienSea 2 project, based on the 'Maritime Cloud' concept. This way, resource providers can instead focus on managing access control and the nomination of access to already validated user identities. By **Authentication** a system verifies the identity of a user (human or machine) who wishes to access it. The Identity Registry will provide facilities for registration of users via a trusted Identity Manager, and provide a Public Key Infrastructure (PKI), capable of issuing digital certificates for the STM validation project, where necessary. Authentication is part of the Core MC ID Registry.

However, there is a need to adapt the functionality of the authentication to comply with the services being built in STM. For example, STM requires more specific and detailed categories of users. Currently, the implementation effort in the 'Maritime Cloud' that deliver an Identity Registry for the STM project are concentrating on passwords for human users and certificates for machine users. In the future, more advanced methods could be supported by Identity Providers associated with the Identity Registry broker in the Maritime Cloud or potentially other multi factor security methods.

The purpose of the **Authorization** mechanism in STM is to provide a way for data providers to secure which specific user identities can read their data. This is particularly important as data will be delivered to other service providers that have subscribers who are not direct partners or lack the contractual arrangements with the data provider.

But authorization in the STM is not part of the Enhanced MC. Authorization as well as subscription is part of the parental / application service. Thus, a service provider is willing to have an authorization mechanism (e.g., via an access control list) he has to design and implement it by its own.

Based on the suggested SR API functions listed above and the current functionalities of the Identity Registry provided by E2, the following requirements were derived from a SeaSWIM/STM perspective. Architects and developers of Act1, Act2 and Act4 of the STM team participated and cooperated with E2 developers in order to derive the most prioritized requirements for the testbed implementation. This means that during the STM Activity 2 testbeds the provided SeaSWIM services will be limited to the core functionality.

5.1 Discussion of enhancement in the area of Service Registry

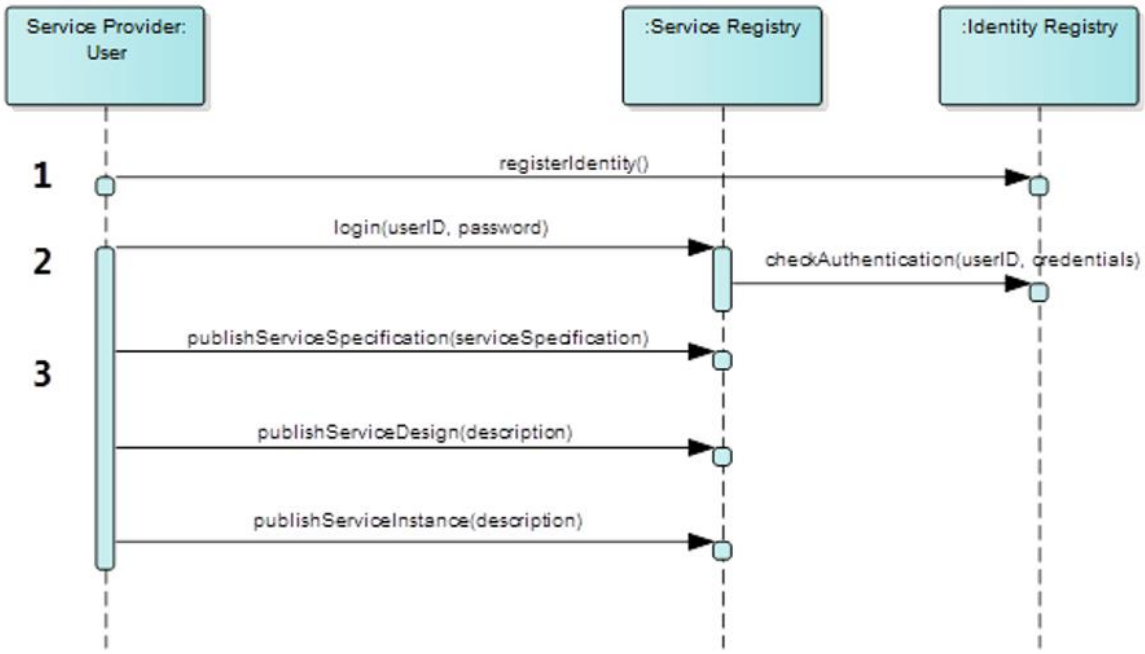
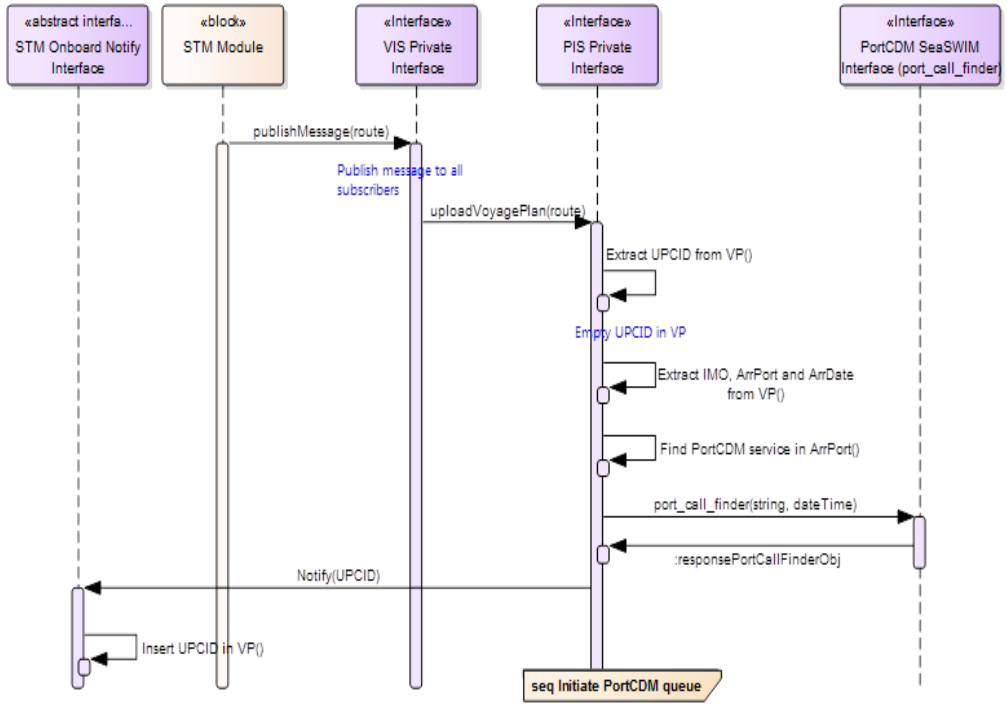


Figure 2: Service Publication – Publish a new Service Specification and Instance

ID	EH_MC_SR_1	
Name	Find service instance	
Origin	Act 1 (PortCDM) and Act2 (VoyMan)	
Scope	The enhanced SR should be able to search and filter the MC – SR for multiple attributes.	
Objectives	ID	Description
	EH_MC_SR_1_O1	Port from Voyage Plan
	EH_MC_SR_1_O2	Service instance

Narrative	
	Figure 3: Service Orchestration – Get Port Call ID (SeaSWIM Connector not shown in this as it is implicitly built into each information service)
	
	Here, find Service Instance is in focus. But there is also the method “find Service Specification”, which is not in focus here.
	
	The following procedure is foreseen:
	• Port in UN/LOCODE format is extracted from the Voyage Plan
	• Find the service instance for the PortCDM service “port_call_finder” and “state_update” in Port=UN/LOCODE.
	  ○ The SeaSWIM Connector “findServices” can be used
	  ○ One service instance is found through Service Registry/SeaSWIM Connector
	
	In this case the organisation could be UN/LOCODE=“SEGOT” which holds a “PortCDM” service with “port_call_finder” and “state_update” endpoints.
	
	➔ findService(organisation=“SEGOT”, serviceType=“PortCDM”) : list of service instances (description of service instances and endpoints)

ID	EH_MC_SR_2	
Name	Service call	
Origin	Discussion among architects and developers on the basis of current architectural sketches and requirements	
Scope	As an **example** for “find service instance” (see above): serviceType = PortCDM Main issues: 1) Possibility to use keywords & search services by keywords 2) Search services by organisation	
Objectives	ID	Description
	EH_MC_SR_2_O1	Transformation result of UN/LOCODE
	EH_MC_SR_2_O2	Result of service call
	EH_MC_SR_2_O3	
Narrative	There needs to be the possibility to find the serviceType=PortCDM and the endpoints to the “port_call_finder” and the “state-Update” in a specific port identified in UN/LOCODE format. Based on information in the Voyage Plan (RTZ format) the Arrival Port is extracted and forms the basis for a service call to the PortCDM. The Arrival Port in RTZ is in UN/LOCODE which need to be transformed into something that is possible to search for a service instance. Just as you can call a service, it is also possible to use and receive a service specification. but here the focus is on the call of a Service Instance Another scenario should be the use of an IMO number to search for service instances by meta data. FREQ also suggested the service coverage element (e.g. modelled port).	

ID	EH_MC_SR_3	
Name	Usage of keywords and categories	
Origin	Discussion among architects and developers on the basis of current architectural sketches and requirements	
Scope	STM should have a set of service types (categories) defined that each service is mapped to	
Objectives	ID	Description
	EH_MC_SR_4_O1	Subset of keywords depending on the level
	EH_MC_SR_4_O2	
	EH_MC_SR_4_O3	

Narrative	Keywords could be unstructured free text words, or it can be a defined enumeration of keywords. In any case it is unlikely that keywords in a free string will satisfy search for a specific type of service. STM should have a set of service types (categories) defined that each service is mapped to, and keywords are complementary search items. A keywords should be a simple string in the context of STM. The procedure is as follows: Look for a specification → Look for instances that apply to the spec → Query object. The result is a subset of keywords depending on the level. There needs to be an attribute that defines geographically mutually exclusive services to harmonise use cases (i.e. machine-to-machine communication). For the service search not all three levels are needed. For the service registration all three levels are needed. There will be a list where actors, i. e. a captain, can choose from. For this, there needs to be a pre-configuration to choose a list entry. STM will have two technical designs: SOAP based/REST based. Two instances will be running in the same area. Part of SSC or client configuration? i.e. has the SSC a search function e.g. are there services in STM that you choose from a list? - The search function is implemented in the SSC, but the filtering parameters are not clear at the moment in the Service Registry - ONE SSC can serve only ONE service.

ID	EH_MC_SR_5	
Name	Documentation of Service Registry	
Origin	Discussion among architects and developers on the basis of current architectural sketches and requirements	
Scope	There need to be a description of the Service Spec, Service Design and Service Instance description for the (STM) Service Registry in the testbed. The format of this description has to be aligned with the Service Documentation Guideline from ES 2 project.	
Objectives	ID	Description
	EH_MC_SR_5_O1	Documentation for Service Registry
	EH_MC_SR_5_O2	
	EH_MC_SR_5_O3	
Narrative	The Service Registry needs to be described as an own structure by templates. One instance document for the service registry will be existent. For this, the Service Spec, Technical Design and Service Instance documents/XSD are used to describe the Service Registry: https://service.projectplace.com/pp/pp.cgi/0/1253114819 This documentation should include information on how to use it correctly.	

ID	EH_MC_SR_6	
Name	Various interfaces	
Origin	Discussion among architects and developers on the basis of current architectural sketches and requirements	
Scope	A service should be able to define more than one interface / endpoint.	
Objectives	ID	Description
	EH_MC_SR_6_O1	Various interface for one service
	EH_MC_SR_6_O2	
	EH_MC_SR_6_O3	
Narrative	One service can have various interfaces. For this, the dataExchangePattern is defined on an interface level in the service specification.	

ID	EH_MC_SR_7	
Name	Access Management	
Origin	Discussion among architects and developers on the basis of current architectural sketches and requirements	
Scope	It shall be possible to find identities to authorise data. I. e. a captain will find organisations and grant access to his data (a voyage plan).	
Objectives	ID	Description
	EH_MC_SR_6_O1	Find identities
	EH_MC_SR_6_O2	Grant access to data
	EH_MC_SR_6_O3	
Narrative		

ID	EH_MC_SR_8	
Name	Specification Access	
Origin	Discussion among architects and developers on the basis of current architectural sketches and requirements	
Scope	The service specification and technical design shall be accessiable from an service instance	
Objectives	ID	Description
	EH_MC_SR_6_O1	Find identities
	EH_MC_SR_6_O2	Grant access to data

	EH_MC_SR_6_O3	
Narrative	To get all relevant information about a service instance, the service specification as well as the technical design shall be accessiable for each service instance.	

5.2 Discussion of enhancement in the area of Identity Registry

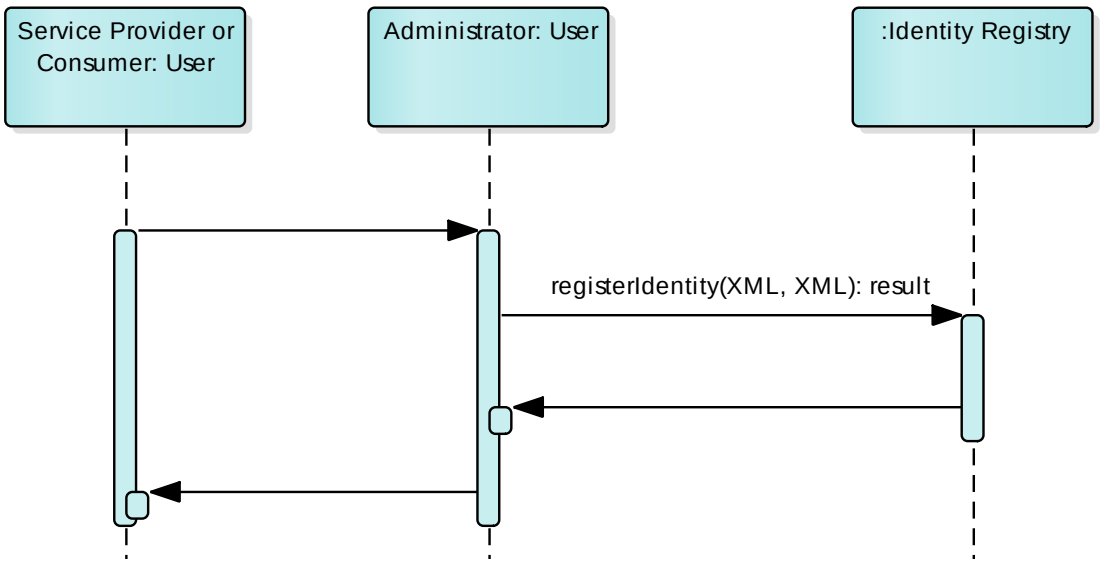
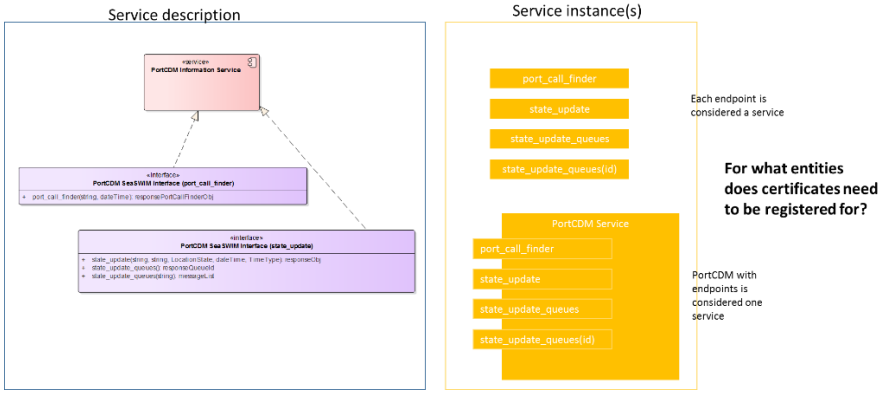


Figure 4: Basic interaction to register an identity

ID	EH_MC_IR_1	
Name	Creation of entities	
Origin	Discussion among architects and developers on the basis of current architectural sketches and requirements	
Scope	There need to be the functionality to get a certificate in return that represents the Identity registered.	
Objectives	ID	Description
	EH_MC_IR_1_O1	Create Organisation
	EH_MC_IR_1_O2	Create Entity
	EH_MC_IR_1_O3	
Narrative	The procedure to create a new Identity should be as follows: • Create Organisation (or select an existing organisation) • Create type of entity; vessel, device, human, service The creation of types of entities (vessel, device, human, service) presupposes that the item owning organisation is already registered – e.g., by creating a vessel, the organisation it belongs to has to be selected. The ID registry manager should manage the possible actors/IDs of the use case.	

ID	EH_MC_IR_2	
Name	Organisation certification	
Origin	Discussion among architects and developers on the basis of current architectural sketches and requirements	
Scope		
Objectives	ID	Description
	EH_MC_IR_2_O1	Link service registry and ID registry
	EH_MC_IR_2_O2	Link entities to organization
	EH_MC_IR_2_O3	
Narrative	An organisation certificate will authorise entities linked to this certificate. The ID & Service Registry need to be combined. STM will only use the service-entity in the testbed, so the vessel-entity will not be used (but the Identity Registry will still support it). To avoid any other entity type being used, it was suggested to hide the other types in the management portal. But since PortCDM will use browser-based interfaces to some extent, they will most probably use the "user" entity type.	

ID	EH_MC_IR_3	
Name	Check Certificate	
Origin	Discussion among architects and developers on the basis of current architectural sketches and requirements	
Scope		
Objectives	ID	Description
	EH_MC_IR_3_O1	Authentication of identities
	EH_MC_IR_3_O2	Authorisation of identities
	EH_MC_IR_3_O3	
Narrative	The SeaSWIM Connector (SSC) shall check authentication of incoming calls and only forward info from authenticated sources. A certificate for authorisation is needed. The authorisation of service entities will be validated. All entities in STM are services (vessel, device, etc.). A certificate to lookup a service and find an identification should be required. The resulting list should be received from service registry or own organization. The link between service registry and ID registry is strictly required. SSC will authenticate certificate parameter. In the case of a valid certificate, the business logic could be used. The service certificate contains organizational information.	

ID	EH_MC_IR_4		
Name	Check service certificate		
Origin	Discussion among architects and developers on the basis of current architectural sketches and requirements		
Scope			
Objectives	ID	Description	
	EH_MC_IR_3_O1		
	EH_MC_IR_3_O2		
	EH_MC_IR_3_O3		
Narrative	 *Figure 5: Service instance description* **Question**: Do we need a specific certificate for each service instance? **Discussion**: Is it possible to share certificate among several services provided by the same organization? What is a service instance? Does the instance correspond to each individual endpoint or all endpoints provided by a service? **Answer**: One agreement between DMA/Frequentis was that the URL supplied for the service instance should be thought of as a "base" URL. For example for the Identity Registry, this means that the base URL is "https://api.maritimecloud.net/". The technical description of the service type (perhaps in the form of a swagger-file) can then describe the relative paths to the endpoints of the services, such as "/oidc/api/orgs". Combining the two paths you then get the endpoint of that call: "https://api.maritimecloud.net/oidc/api/org/". Not final - Should be further discussed with Frequentis/DMA.		

ID	EH_MC_IR_5	
Name	Interaction	
Origin	Discussion among architects and developers on the basis of current architectural sketches and requirements	
Scope		
Objectives	ID	Description
	EH_MC_IR_3_O1	Machine-to-Machine interaction
	EH_MC_IR_3_O2	Human-to-Machine interaction
	EH_MC_IR_3_O3	
Narrative	There are two ways to interact: • Human interaction to a service with graphical interface • Machine interaction with service If a human wants to use a GUI, i.e. in a browser, there has to be OpenID connection support as method for credential authentication. In this case, SSC will be bypassed but SSC functionality will be used (for example by a library). Strict government is needed anway. For humans it is possible to login with credentials from the ID registry or for machines by certificate. Humans will be able to login to ID registry UI and manage credentials/create services/vessels/etc. For human users a certificate is not needed. Not final - Should be further discussed with Frequentis/DMA.	

ID	EH_MC_IR_6	
Name	Find organisations	
Origin	Discussion among architects and developers on the basis of current architectural sketches and requirements	
Scope	There will not be a vessel ID for testbed but a service ID that has a link to Identity Registry. SSC authenticates the service certificate. Everyone should be able to find all organisations.	
Objectives	ID	Description
	EH_MC_IR_3_O1	
	EH_MC_IR_3_O2	
	EH_MC_IR_3_O3	
Narrative		

ID	EH_MC_IR_7	
Name	Coupling of IR and SR	
Origin	Discussion among architects and developers on the basis of current architectural sketches and requirements	
Scope	The identities of the identity and service registry need to be linked to each other	
Objectives	ID	Description
	EH_MC_IR_3_O1	
	EH_MC_IR_3_O2	
	EH_MC_IR_3_O3	
Narrative	We have to ensure, that the identity assigned to a service (specification, technical design and instance) is the same as the identity in the identity registry. This can either be done manually (for the testbeds) or automatically.	

ID	EH_MC_IR_8	
Name	Creation of organizations	
Origin	Discussion among architects and developers on the basis of current architectural sketches and requirements	
Scope	There needs to be a governance procedure that describes who is able to create an organization within the identity registry	
Objectives	ID	Description
	EH_MC_IR_3_O1	
	EH_MC_IR_3_O2	
	EH_MC_IR_3_O3	
Narrative		

ID	EH_MC_IR_9	
Name	Certificate Attributes	
Origin	Discussion among architects and developers on the basis of current architectural sketches and requirements	
Scope	The attributes, saved within the certificate has to be defined	
Objectives	ID	Description
	EH_MC_IR_3_O1	
	EH_MC_IR_3_O2	
	EH_MC_IR_3_O3	
Narrative	We do need a list of attributes that will be saved for the different types of entities, within the generated certificates. Thereby the most important attribute for the testbed is the service id, for a service entity (see also requirement EH_MC_IR_7).	

6 Gap Analysis

Here it is pointed out, which requirement and which use cases are met by which MC function or at which point extensions and additions to the MC are necessary.

UC_ID	MC_Func. ID	Enhancement ID
STM_UC#1	in the ID Broker (OpenID, Keycloak instance), separated from the ID registry	
STM_UC#2	in the ID Broker (OpenID, Keycloak instance), separated from the ID registry	
STM_UC#3	MC_SR_3, MC_SR_4, MC_SR_12, MC_SR_13, MC_SR_14, MC_SR_20, MC_SR_22, MC_SR_24, MC_SR_27, MC_SR_28	Refinement: EH_MC_SR_1
STM_UC#4	MC_SR_3, MC_SR_4	
STM_UC#5	depends on the parental service	
STM_UC#6	depends on the parental service	
STM_UC#7	depends on the parental service	Refinement: EH_MC_SR_2
STM_UC#8	MC_SR_1, MC_SR_7, MC_SR_8, MC_SR_9, MC_SR_11, MC_SR_18, MC_SR_19, MC_SR_23	
STM_UC#9	MC_SR_1, MC_SR_7, MC_SR_9, MC_SR_11, MC_SR_16, MC_SR_17, MC_SR_21, MC_SR_27, MC_SR_28, MC_SR_29	
STM_UC#10	MC_SR_5, MC_SR_15, MC_SR_25, MC_SR_26	
STM_UC#11	depends on the parental service	
R_ID_1	ensured by the Maritime Resource Name (MRN) Concept	
R_ID_2	ensured by the Maritime Resource Name (MRN) Concept	
R_ID_3	ensured by the Maritime Resource Name (MRN) Concept	
R_ID_4	ensured by the Maritime Resource Name (MRN) Concept	
R_ID_5	ensured by the Maritime Resource Name (MRN) Concept	
R_ID_6	ensured by the Maritime Resource Name (MRN) Concept	
R_ID_7	in the ID Broker (OpenID, Keycloak instance), separated from the ID registry	
R_ID_8	Regarding authorization: depends on the parental service	
R_SR_1	MC_SR_1, MC_SR_2, MC_SR_3, MC_SR_4, MC_SR_5, MC_SR_11, MC_SR_12, MC_SR_13, MC_SR_14, MC_SR_15, MC_SR_16, MC_SR_21, MC_SR_22, MC_SR_23, MC_SR_24, MC_SR_25, MC_SR_26	

R_SR_2		EH_MC_SR_3
R_SR_3	--- depends on the parental service	
R_SR_4	MC_SR_3, MC_SR_4, MC_SR_12, MC_SR_13, MC_SR_14, MC_SR_20, MC_SR_22, MC_SR_24, MC_SR_27, MC_SR_28	
R_SR_5	--- depends on the parental service	
R_SR_6	--- no relation to the ID and / or SR	

7 Conclusion

This report is intended to show the gaps between the STM requirements regarding the service and identity registry and the functions delivered by the maritime cloud / E2. This report identified these gaps, the subject is here not to solve these gaps. Further discussions are needed considering these gaps, timeline and prioritization.

Currently, the following enhancements and refinements are identified:

Enhancement 1)

Services should be grouped together (Cluster). For instance, clusters can be platform specific, add-on services or a geographical subset.	R_SR_2

This issue is not considered by MC E2. For this requirement the solutions discussion has been started in the enhancement no. EH_MC_SR_3. But this issue has also to be discussed further with Act 1 and Act 2 of the STM Validation project.

Refinement 1)

Search a service	STM_UC#3

Basically, the requirement is met. However, some aspects discussed could lead to a refinement. The discussion will be continued.

Refinement 2)

All types and attributes of identities must be identifiable.	R_ID_1

The ID Reg is limited to ‘active’ entities (Service, Device, User, Organizations). But basically any maritime object (containers, voyages etc.) can get a MRN. There is a need to ensure that – in this context – not active entities (Voyage, Port etc.).

In this report, the STM requirements were referred specifically to the ID and Service Registry. The proposed functions of the MC were presented. These aspects have been merged into a gap analysis and the gaps were identified and considered.

38 partners from 13 countries -
Creating a safer more efficient and environmentally friendly maritime sector

Demonstrating the function and business value of the Sea Traffic Management concept and its services.

SAFETY - ENVIRONMENT - EFFICIENCY

Swedish Maritime Administration ◦ SSPA ◦ RISE Viktoria ◦ Transas/ Wärtsilä Voyage ◦ Chalmers University of Technology ◦ The Swedish Meteorological and Hydrological Institute ◦ Danish Maritime Authority ◦ Navicon ◦ Novia University of Applied Sciences ◦ Fraunhofer ◦ Carnival Corp. ◦ Italian Ministry of Transport ◦ SASEMAR ◦ Valencia Port Authority ◦ Valencia Port Foundation ◦ CIMNE ◦ University of Catalonia ◦ Norwegian Coastal Administration ◦ GS1 ◦ Cyprus University of Technology ◦ Port of Barcelona ◦ Costa Crociere ◦ Svitzer ◦ OFFIS ◦ Finnish Transport Agency ◦ Southampton Solent University ◦ Frequentis ◦ Wärtsilä SAM Electronics ◦ University of Flensburg ◦ Airbus ◦ 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 Connecting Europe Facility of the European Union