Document No:

Title: SeaSWIM Connector Technical Design

Date: 2019-01-14 v1.0

Co-financed by the European Union
Connecting Europe Facility

Authors

Name	Organisation
Fabio Renda	CIMNE
Mikael Olofsson	SMA

Document History

Version	Date	Initials	Description
Version 0.1	2016-11-06	FR,MO	(aka 1.0) Created and sent on internal review
Version 0.2	2016-11-16	FR,MO	Updated after internal review • WSDL file in APPENDIX is updated • JSON file in APPENDIX • Sequence diagrams are updated
Version 0.3	2016-11-16	MO	Minor corrections
Version 1.0	2019-01-14	MO	Moved SOAP to its own Technical Design Minor editorial corrections and alignment to delivered SeaSWIM Connector Proxy Service

Review

Name	Organisation
Mikael Olofsson	SMA

Co-financed by the European Union
Connecting Europe Facility

Table of contents

Co-financed by the European Union
Connecting Europe Facility

1 Introduction

1.1 Purpose of the document

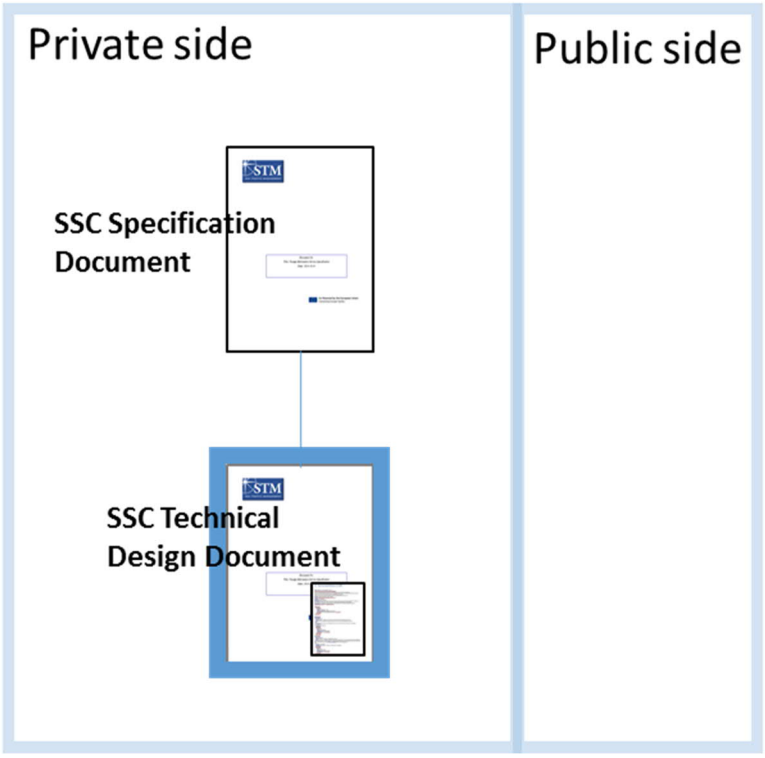
The purpose of this technical design description document is to provide a detailed description of the SeaSWIM Connector, realized by using a specific technology. It describes a well-defined baseline of the SeaSWIM Connector technical design by clearly identifying the design version.

The aim is to document the key aspects of the technical design. This includes:

- identification and summary of the design
  - reference to the specification
  - identification of the design
- identification and summary of chosen technology

1.2 Intended readership

This technical design description document is intended to be read by architects, designers, system engineers and developers in charge of designing and developing an instance of the SeaSWIM Connector.



1.3 Inputs from other projects

Designed based on interfaces and behaviour of identity and service registry in Maritime Connectivity Platform (MCP) from EfficienSea2 project.

Co-financed by the European Union
Connecting Europe Facility

2 Design Identification

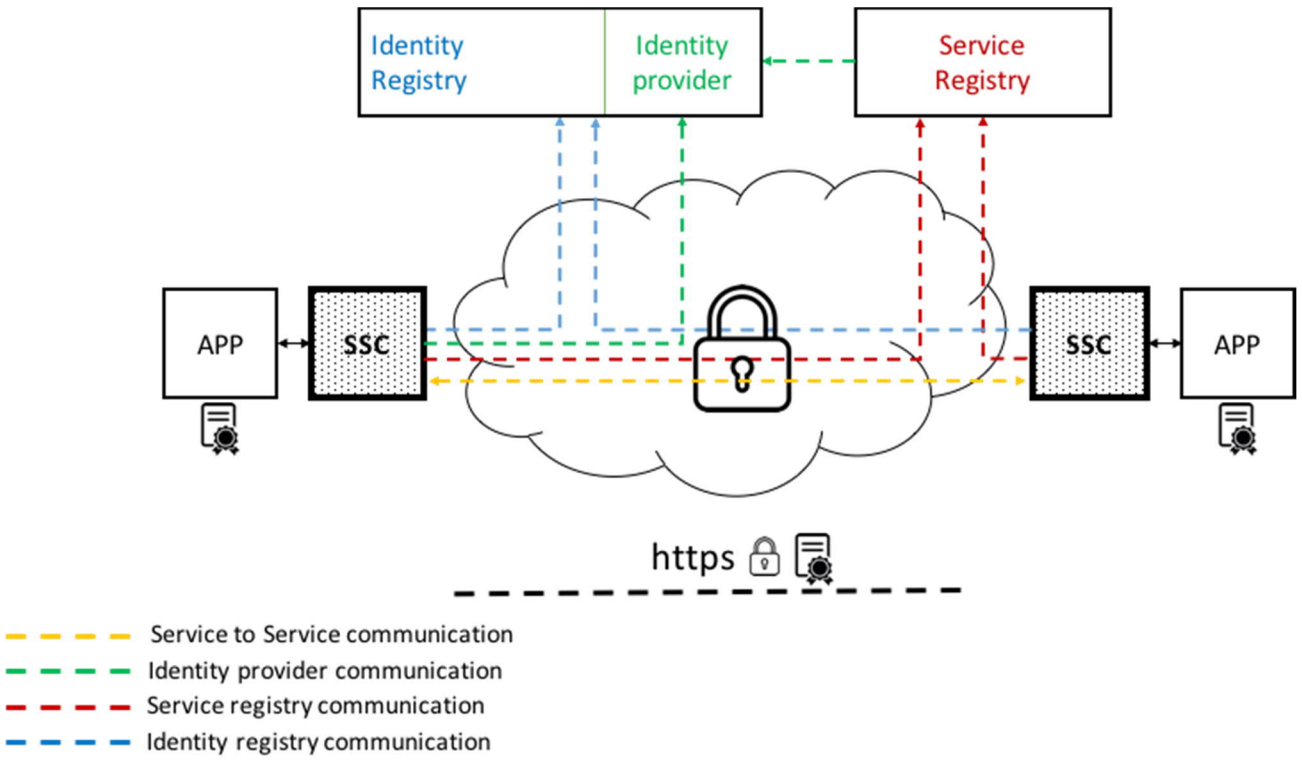
Name	SeaSWIM Connector Technical Design, CIMNE
ID	urn:mrn:stm:technical:design:cimne:ssc
Version	1.0
Specification	SeaSWIM Connector Specification v1.0
Description	Describe technical design for connection to SeaSWIM
Keywords	SSC, SeaSWIM Connector
Architect(s)	Fabio Renda
Status	released

Co-financed by the European Union
Connecting Europe Facility

3 Technology Introduction

3.1 General architecture overview

The role of the functionalities in SeaSWIM Connector (SSC) in the overall architecture shall be the place where the operation of safe/secure communication and interoperability shall be deployed.



SeaSWIM Connector shall take in account the communication flow between the core services of the SeaSWIM (identity registry and identity provider, service registry) and the rest of application services deployed in the infrastructure.

SeaSWIM Connector functionalities shall support the use of secure communication channels and authentication standards implementing the general requirements of the STM infrastructure.

Co-financed by the European Union
Connecting Europe Facility

3.2 Communication standards

3.2.1 HTTP over TLS

HTTPS provides three security guarantees:

- Server authentication allows users to have some confidence that they are talking to the true application server. Without this guarantee, there can be no guarantee of confidentiality or integrity.
- Data confidentiality means that eavesdroppers cannot understand the content of the communications between the user's browser and the web server, because the data is encrypted.
- Data integrity means that a network attacker cannot damage or alter the content of the communications between the user's browser and the web server, because they are validated with a cryptographic message authentication code.

References:

- RFC-2246, TLS version 1.0 (1999), https://www.ietf.org/rfc/rfc2246.txt
- RFC-4346, TLS version 1.1 (2006), https://www.ietf.org/rfc/rfc4346.txt
- RFC-5246, TLS version 1.2 (2008), https://www.ietf.org/rfc/rfc5246.txt
- RFC 8446, TLS version 1.3 (2018), https://tools.ietf.org/html/rfc8446
- RFC 2818, HTTP Over TLS (2000), https://tools.ietf.org/html/rfc2818
- *"HTTP Over TLS". The Internet Engineering Task Force. RetrievedFebruary 27,2015.* - https://tools.ietf.org/html/rfc2818
- *"Use of client certificates"* - https://tools.ietf.org/html/rfc5246#section-7.4.6
- *"SSL/TLS Strong Encryption: FAQ".apache.org.* - http://httpd.apache.org/docs/2.0/mod/mod_ssl.html

Co-financed by the European Union
Connecting Europe Facility

3.3 Authentication standards

3.3.1 X.509 certificates

Industrial standards certificates delivered by the identity registry and used in the STM infrastructure to assure the authenticity of the information and to create secure communication channels.

References:

- RFC 5280 profile of X.509 (https://tools.ietf.org/html/rfc5280)
- RFC 4630 (https://tools.ietf.org/html/rfc4630)

- *"Use of client certificates"* - https://tools.ietf.org/html/rfc5246#section-7.4.6

Co-financed by the European Union
Connecting Europe Facility

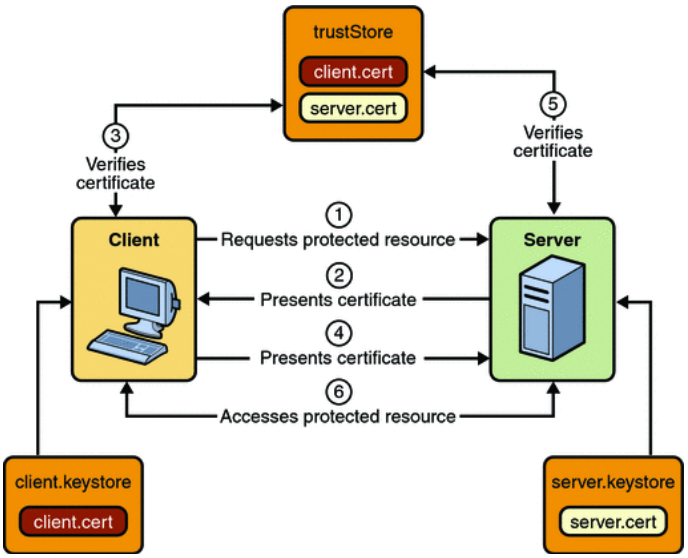
3.3.2 Authentication procedure

3.3.3 Certificate based mutual authentication M2M

With mutual authentication, the server and the client authenticate one another. This assure that the machines in the STM infrastructure have the right to send messages across the platform.

When using certificate-based mutual authentication, the following actions occur:

- A client requests access to a protected resource.
- The web server presents its certificate to the client.
- The client verifies the server's certificate.
- If successful, the client sends its certificate to the server.
- The server verifies the client's credentials.
- If successful, the server grants access to the protected resource requested by the client.



References:

- *"HTTP Over TLS". The Internet Engineering Task Force. RetrievedFebruary 27,2015.* - https://tools.ietf.org/html/draft-ietf-httpauth-mutual-10

3.3.3.1 Verify the server's certificate

The procedure to verify the server certificate are:

1. The client shall verify that the server's certificate is valid and not revoked.
2. The client shall verify that the server's certificate is trusted.
3. The client shall verify that the server's Common Name (CN) correspond to the domain in the URL
4. The client shall verify that the server's Subject Alternative Name (SAN) correspond to the domain in the URL

3.3.3.2 Verify the client's certificate

The procedure to verify the client certificate are:

1. The server shall verify that the client's certificate is valid and not revoked
2. The server shall verify that the client is trusted by searching for the common Root Certificate in the trust chain in the client certificate

3.3.4 Certificate Revocation

The certificates shall be constantly validated against the Certificate Revocation List (CRL) using the link in the certificate.

Co-financed by the European Union
Connecting Europe Facility

3.4 Web Service Technologies

3.4.1 REST

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

Representational State Transfer was first described in Architectural Styles and the Design of Network-based Software Architectures, chapter 5 and has gained popularity due to its simplicity.
https://web.archive.org/web/20060717120412/http://rest.blueoxen.net/cgi-bin/wiki.pl?ShortSummaryOfRest

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

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

3.4.1.1 Swagger

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

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

References:

- *Fielding, Roy Thomas (2000). "Chapter 5: Representational State Transfer (REST)". Architectural Styles and the Design of Network-based Software Architectures (Ph.D.). University of California, Irvine.*
- *Richardson, Leonard; Ruby, Sam (2007), RESTful Web service, O'Reilly Media, ISBN978-0-596-52926-0, retrieved 18 January 2011.*
- *Richardson, Leonard; Amundsen, Mike (2013), RESTful Web APIs, O'Reilly Media, ISBN978-1-449-35806-8, retrieved 15 September 2015*
- Swagger Open API specification - http://swagger.io/specification/

3.4.2 SOAP

XML-based protocol for web services.

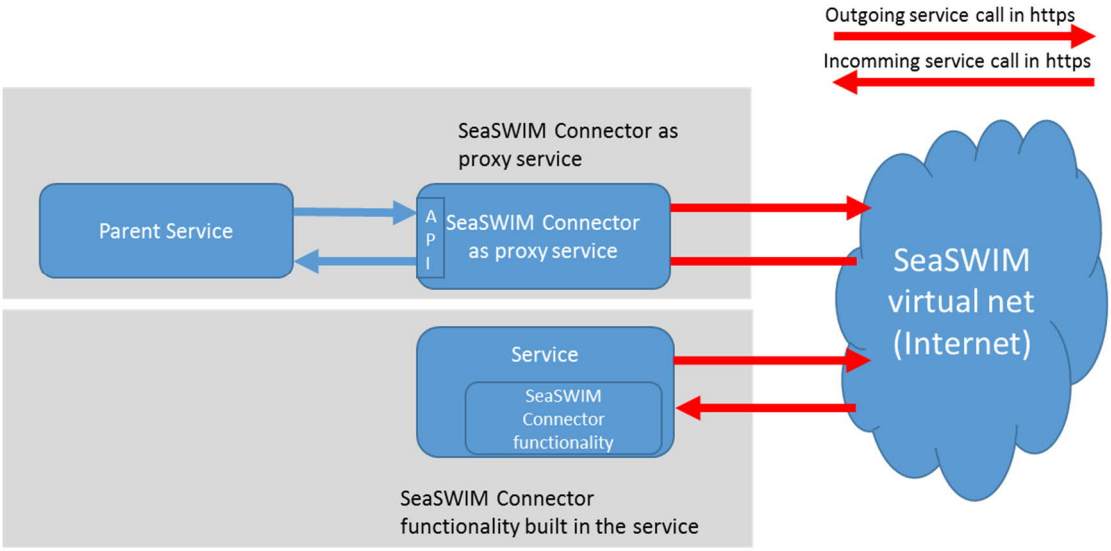
http://www.w3.org/TR/soap12-part1/

Co-financed by the European Union
Connecting Europe Facility

4 Design Overview

A SeaSWIM Connector represent the principals and procedures to enable Your service compliant with SeaSWIM. Every service in SeaSWIM is expected to follow the rules and guidelines described here.

The SeaSWIM Connector can be designed as a proxy service in front of the parent service, or built into the parent service itself.



4.1 Design principals

The following design principals shall be followed:

- HTTP over TLS
- X.509 Certificates for authentication
  - Client Certificates shall contain common root certificate (MCP Root Certificate) in trustchain, hence issued from common source (MCP)

4.2 Service Interface Technical Design

For the technical design of the private service interface towards its parent, see separate documents for REST and SOAP technology.

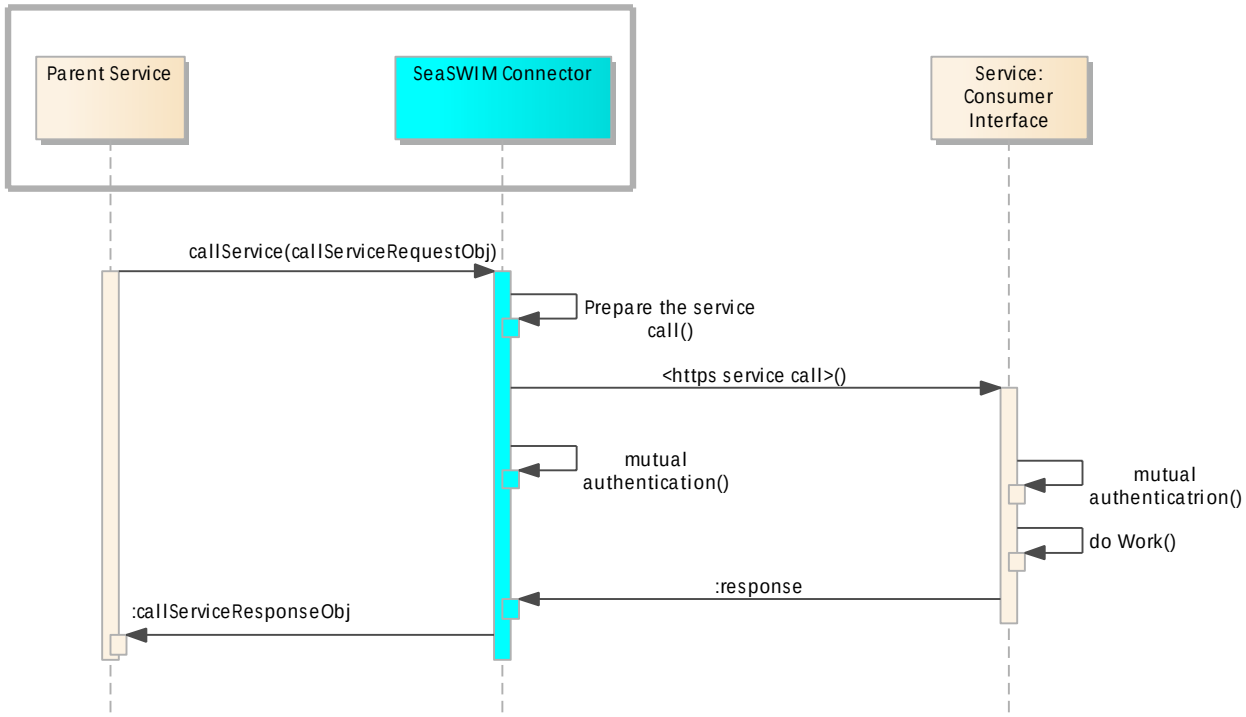
Co-financed by the European Union
Connecting Europe Facility

5 Dynamic Behaviour

This chapter describes the dynamic behaviour of SeaSWIM Connector in context of the parent service/application and an opposing service consumer/producer. The SeaSWIM does not have to be a separate proxy service, but the dynamic behaviour shall be the same from the perspective of the opposing service.

5.1 Interaction – <outgoing calls>

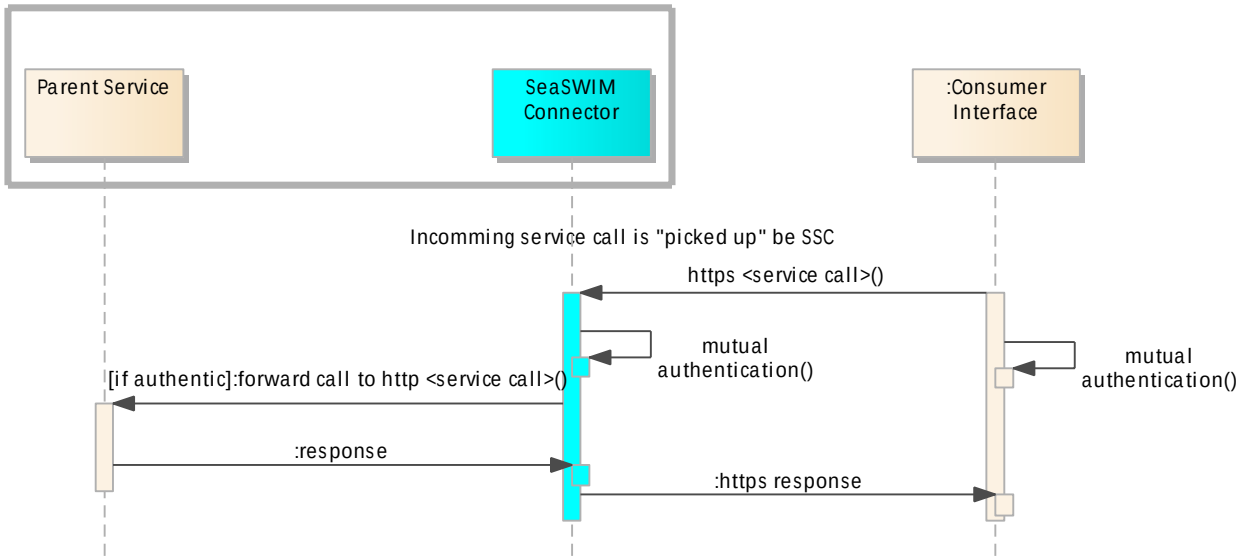
SeaSWIM Connector shall support mutual authentication and HTTPS on outgoing service calls. The operation shall be generic and transparent and the parent is responsible for providing the correct endpoint, request type, header/parameters and body according to the consumed service description.



Co-financed by the European Union
Connecting Europe Facility

5.2 Interaction - <incomming call>

SeaSWIM Connector shall intercept incoming service request and perform mutual authentication according to procedure in chapter 0 and, if source is authenticated, forward the service call to the "parent" service or application.

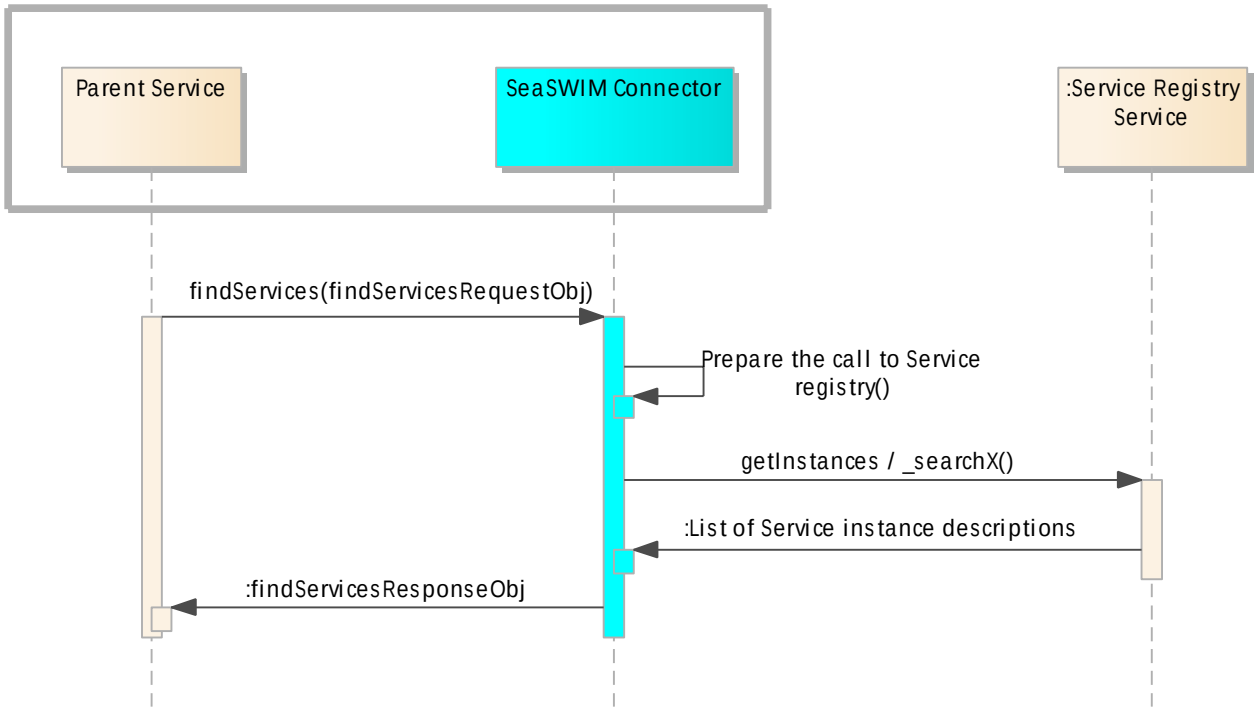


Co-financed by the European Union
Connecting Europe Facility

5.3 Interaction – find Service

SeaSWIM Connector shall support search in service registry for service instances to consume that matches the provided search parameters.

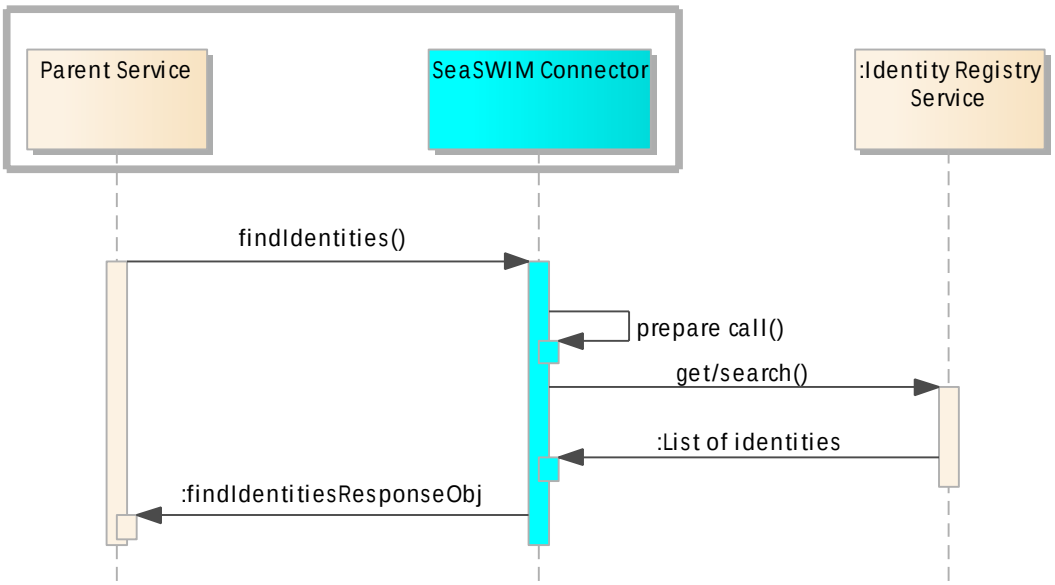
Search shall be possible on, but not limited to, service keywords, geolocation, service status, service type, service instance identity and service design identity.



Co-financed by the European Union
Connecting Europe Facility

5.4 Interaction – find Identity

SeaSWIM Connector shall support search in identity registry.



Co-financed by the European Union
Connecting Europe Facility

6 SeaSWIM Connector Provision

SeaSWIM Connector can be further designed as a proxy service with different technical interfaces or as software library API to be included in parent service or application.

Co-financed by the European Union
Connecting Europe Facility

7 References

Reference name	Comment	Link
SeaSWIM Connector Documentation		http://stmvalidation.eu/developers-forum/ssc
HTTP Over TLS	*The Internet Engineering Task Force. Retrieved February 27, 2015.*	https://tools.ietf.org/html/rfc2818
SSL/TLS Strong Encryption: FAQ	*apache.org.*	http://httpd.apache.org/docs/2.0/mod/mod_ssl.html
Use of client certificates		https://tools.ietf.org/html/rfc5246#section-7.4.6
X.509	Certificates	https://tools.ietf.org/html/rfc4630
HTTP Over TLS - Mutual Authentication	*The Internet Engineering Task Force. Retrieved February 27, 2015.*	https://tools.ietf.org/html/draft-ietf-httpauth-mutual-10

Co-financed by the European Union
Connecting Europe Facility

8 Acronyms and Terminology

8.1 Acronyms

Term	Definition
API	Application Programming Interface
MEP	Message Exchange Pattern
REST	Representational State Transfer
SSC	SeaSWIM Connector
URN	Uniform Resource Locator
XML	Extendible Mark-up Language
XSD	XML Schema Definition

8.2 Terminology

Term	Acronym	Definition
External Data Model		Describes the semantics of the "maritime world" (or a significant part thereof) by defining data structures and their relations. This could be at logical level (e.g., in UML) or at physical level (e.g., in XSD schema definitions), as for example standard data models, or S-100 based data produce specifications. *Source* *E2 D3.4 Service Documentation Guidelines v01.01*
Service		The provision of something (a non-physical object), by one, for the use of one or more others, regulated by formal definitions and mutual agreements. Services involve interactions between providers and consumers, which may be performed in a digital form (data exchanges) or through voice communication or written processes and procedures. *Source* *E2 D3.4 Service Documentation Guidelines v01.01*
Service Data Model		Formal description of one dedicated service at logical level. The service data model is part of the service specification. Is typically defined in UML and/or XSD. If an external data model exists (e.g., a standard data model), then the service data model shall refer to it: each data item of the service data model shall be mapped to a data item defined in the external data model. *Source* *E2 D3.4 Service Documentation Guidelines v01.01*
Service Technical Design		The technical design of a dedicated service in a dedicated technology. One service specification may result in several technical service designs, realising the service with different or same technologies. *Source* *E2 D3.4 Service Documentation Guidelines v01.01*

Co-financed by the European Union
Connecting Europe Facility

Service Endpoint		A Service Endpoint is the URL where your service can be accessed by a client application. The same web service can have multiple endpoints, for example in order to make it available using different protocols. *Source* *http://stackoverflow.com/questions/9807382/what-is-a-web-service-endpoint*
Service Operation		Functions or procedure which enables programmatic communication with a service via a service interface. *Source* *E2 D3.4 Service Documentation Guidelines v01.01*
Service Parameters		Service Parameters are input to a Service Operation and can be described formally in a data exchange model as e.g. XML Schemas. *Source* *MO*
Service Response		Service Response are output from a Service Operation and can be described formally in a data exchange model as e.g. XML Schemas. *Source* *MO*
Service Consumer		A service consumer uses service instances provided by service providers. All users within the maritime domain can be service customers, e.g., ships and their crew, authorities, VTS stations, organizations (e.g., meteorological), commercial service providers, etc. *Source* *E2 D3.4 Service Documentation Guidelines v01.01*
Service Provider		A service provider provides instances of services according to a service specification and service instance description. All users within the maritime domain can be service providers, e.g., authorities, VTS stations, organizations (e.g., meteorological), commercial service providers, etc. *Source* *E2 D3.4 Service Documentation Guidelines v01.01*
Service Request		*Source*

Co-financed by the European Union
Connecting Europe Facility