

EUROPEAN COMMISSION

EUROPEAN MARITIME SAFETY AGENCY

Cais Do Sodré 1249-206 Lisbon, Portugal

SafeSeaNet System Design Document

SSN EIS

Document version: 1.78

Document release date: March 2019

Application version: 4.2

XMLRG version: 4.0

Document Approval

	NAME	DATE	SIGNATURE
Prepared by:	I. Ioannou, Y. Tassopoulos,	14.01.2019	
Checked by:	A. Argyropoulos	18.01.2019	
Quality control by:	N. Karioti	18.01.2019	
Approved by:			

Distribution List

COMPANY	NAME	FUNCTION	FOR INFO / APPROVAL
EMSA			FA
Member States			FI
SSN central system contractor			FI

Change control History

VERSION	DATE	AUTHOR	DESCRIPTION
0.90	25.04.2014	Intrasoft International	Submitted to EMSA for review.
0.95	23.05.2014	Intrasoft International	Updated to integrate SSN specific requirements based on SC#09 under FC 11/EMSA/OP/08/2011. Submitted to EMSA for review.
0.96	18.06.2014	Intrasoft International	Working version (processing EMSA comments).
1.00	01.07.2014	Intrasoft International	Incorporated EMSA comments.
1.10	18.07.2014	Intrasoft International	Incorporated EMSA comments. Submitted for acceptance.
1.20	06.08.2014	Intrasoft International	Corrected text according to the acceptance review by EMSA.
1.30	01.09.2014	Intrasoft International	Updated to incorporate the new version of the MRS notification in the context of SC#10.
1.35	17.09.2014	Intrasoft International	Incorporated EMSA comments in the context of SC#10. Submitted for acceptance.
1.40	22.10.2014	Intrasoft International	Incorporated final set of EMSA acceptance review comments.
1.45	14.11.2014	Intrasoft International	Incorporated further comments from EMSA.

1.46	28.01.2015	Intrasoft International	Updated sections 3.1.2, 4.4 and 5 to define the ssn-xmlprotocol-app for XMLRG v2 messages support. New section 4.5 for the Central Database services.
1.50	31.07.2015	Intrasoft International	Updated in the context of SC#13 and SC#14.
1.51	23.10.2015	Intrasoft International	Updated to include the UC for the COD, CLD and CSD. Incorporated EMSA review comments.
1.60	07.06.2016	Intrasoft International	Updated in the context of SC#1.
1.62	22.07.2016	Intrasoft International	Updated with SEG Enrichment in the context of SC#1.
1.63	30.11.2016	Intrasoft International	Updated with CSD changes in the context of SC#1.
1.64	20.01.2017	Intrasoft International	EMSA comments incorporated (including COD/CLD changes in the context of SC#1).
1.65	24.02.2017	Intrasoft International	EMSA changes and comments in the previous version analysed and incorporated.
1.67	13.03.2017	Intrasoft International	Open issues – negotiation & resolution.
1.68	10.08.2017	Intrasoft International	Updated in view of EMSA/NEG/16/2017. Chapters 4.9 and 4.11.
1.69	04.09.2017	Intrasoft International	EMSA comments incorporated.
1.70	22.12.2017	Intrasoft International	Updated sections 3 and 4, SSN v4 release. Annex A, Principles for sharing of ship particulars"" is removed as deprecated. Relevant Use Cases need revision.
1.71	26.01.2018	Intrasoft International	EMSA comments incorporated.
1.72	26.04.2018	Intrasoft International	ES namespace reverted to eu.emsa.ssn.voyage – REST enrichment messages reverted to SSN2SEG
1.73	05.06.2018	Intrasoft International	Updated section 3.2.8 to incorporate the revised CSD use cases. Added annexes B and C
1.74	14.08.2018	Intrasoft International	Updated steps of section 3.2.8 CSD Management
1.75	23.01.2019	Intrasoft International	Updated in the context of SC#10 under FWC 15/EMSA/OP/18/2015: EIS Integration with IdM V2
1.76	08.02.2019	Intrasoft International	Updated in view of review meeting based on EMSA comments.
1.77	15.02.2019	Intrasoft International	Updated sections 3.2.2.6, 3.2.2.7, 3.2.2.8, 3.2.2.12, added section Annex D: Tasks following EIS Design Review meeting.

1.78	20.03.2019	Intrasoft International	<p>Updated sections:</p> <ul style="list-style-type: none">• 3.2.4.1, 3.2.4.2, 3.2.4.3: Alternative Use Cases, Output(s);• 3.2.3.4, 3.2.3.14, 3.2.7.1: References to "SSN participant" regional agreement removed (Please refer to email: "Possible change in EIS design for 4.2" regarding "SSN participant" countries);• 4.3.2.2.1, 4.3.2.2.2, 4.3.2.3.1, 4.3.2.3.3: Minor changes to UML class diagrams;• 4.12, 4.13, 5.2.7: SSN GI and respective STIRES/ACCIIS Annex D: Tasks – DEPRECATED.
------	------------	----------------------------	---

Contents

SafeSeaNet System Design Document.....	1
Contents.....	5
1 Introduction	9
1.1 Purpose.....	9
1.2 Scope	9
1.3 Reference documents.....	9
1.4 Abbreviations and acronyms	10
2 Architectural Goals and Constraints	13
2.1 Service-Oriented Architecture (SOA).....	13
2.2 Java EE Technologies	14
2.3 Industry Standards.....	14
2.4 Business Processes	15
2.5 Open Source Frameworks.....	15
2.5.1 Spring framework	15
2.5.2 Hibernate framework	17
2.5.3 ESAPI (The OWASP Enterprise Security API)	17
2.6 RESTful services using Jersey	17
2.7 XML Web Services (JAX-WS).....	18
2.8 Web features	18
2.8.1 HTML5	18
2.9 Non-functional requirements.....	18
2.9.1 JavaServer Faces.....	19
2.10 Application Server	19
2.11 Database.....	20
3 Overall System Architecture	21
3.1 Definitions.....	21
3.2 Functional Architecture.....	21
3.2.1 Handle Incoming Message	24
3.2.2 Data Provider.....	30
3.2.3 Data Request	57
3.2.4 Data Receive.....	97
3.2.5 Monitoring IncidentReport	107
3.2.6 MRS Management.....	109
3.2.7 System Interface Management.....	112
3.2.8 CSD Management.....	113
3.2.9 Voyage Calculation Process.....	142

4	Design of System Components	147
4.1	SSN-EIS	147
4.2	SSN Core Application - ssn-core-app.....	149
4.2.1	SSN Core Main Components.....	149
4.2.2	UML Class and Sequence Diagrams.....	156
4.2.3	Security on the ssn-core-app	209
4.3	SSN Resources Core Application - ssn-resources-core-app.....	210
4.3.1	SSN Resources Core Main Components	210
4.3.2	UML Class and Sequence Diagrams.....	212
4.4	SSN Console Application - ssn-console-app.....	230
4.5	SSN XML Protocol Application - ssn-xmlprotocol-app.....	231
4.5.1	SSN XML Protocol Main Components	232
4.5.2	UML Class and Sequence Diagrams.....	233
4.5.3	Interfaces between the ssn-xmlprotocol-app and the ssn-core-app	242
4.5.4	Security on the ssn-xmlprotocol-app	242
4.6	SSN-Central Database (CD)	242
4.6.1	Ship Particulars Service	245
4.6.2	IdM User Service	248
4.6.3	ssn-resources-core-app / enterprise archive	250
4.7	SSN-IMDaTe	250
4.7.1	Domain module/package	253
4.8	STIRES Core	260
4.8.1	Domain module/package	263
4.9	SSN-VMS	268
4.9.1	Domain module/package	270
4.9.2	VMS-Proxy.....	272
4.10	SSN-SI	274
4.11	SSN SI Central Management Console	277
4.12	SSN GI - DEPRECATED.....	278
4.12.1	Spatial Information Visualisation Scheme	278
4.12.2	Structure.....	279
4.12.3	Behaviour	281
4.13	SSN-GI / EIS notification details request protocol mechanism upgrade - DEPRECATED.....	282
4.13.1	Current mechanism	282
4.13.2	Proposed improvement.....	283
4.14	Message Queues	283
4.15	Vessel Management – Upload Single Hull Tankers	285

4.16	Vessel Management – OSD Synchronization.....	286
4.17	Vessel Management – CSD Synchronization.....	287
4.18	EIS & STIRES interoperability – Get Enrichment data.....	288
5	Deployment view	290
5.1	Design Decisions	290
5.2	SSN EIS	290
5.2.1	EIS Console Server	292
5.2.2	EIS Core Server	292
5.2.3	EIS Resources Console Server.....	293
5.2.4	EIS Resources Core Server	293
5.2.5	SI Central Console Server.....	294
5.2.6	SI Central Core Server	294
5.2.7	SSN GI - DEPRECATED	294
5.3	SSN-IMDaTe.....	294
5.4	SSN-VMS	295
Annex A:	Business Rules.....	298
	Voyage Status Indicators.....	298
	Voyage retrieval specific rules.....	299
	Voyage correlation further rules	302
	STIRES/STAR notification consolidation - Business Processes and voyageID assignment rules for "detected" voyages.....	305
Annex B:	CSD Specific Business Rules.....	307
Annex C:	CSD Management process diagrams.....	309
	UC-CSD-SPN-14 Process Ship Particulars Notifications	309
	UC-CSD-SPN-15 Process Ship Particulars notifications in EIS	309
	UC-18 CSD synchronisation with OSD	309
	UC-19 OSD synchronisation with CSD	310
Annex D:	Tasks	311

Table of Figures

FIGURE 2-1: EXAMPLE OF EIS SCA PROTOTYPE BASED ON EXISTING SPRING APPLICATION	14
FIGURE 3-1 SSN SPECIFIC USE CASE DIAGRAM 1	23
FIGURE 3-2 SSN SPECIFIC USE CASE DIAGRAM 2	24
FIGURE 4-1 SSN EIS COMPONENTS CONNECTIONS DIAGRAM.....	147
FIGURE 4-2 SSN SERVICES PROVIDED OPERATIONS.....	148
FIGURE 4-3 SSN SERVICES PROVIDED OPERATIONS.....	148
FIGURE 4-4 SSN CORE APPLICATION - SSN-CORE-APP.....	149
FIGURE 4-5 SSN RESOURCES CORE APPLICATION - SSN-RESOURCES-CORE-APP.....	210
FIGURE 4-6 SSN XML PROTOCOL APPLICATION - SSN-XMLPROTOCOL-APP	231
FIGURE 4-7 NOTIFICATION / REQUEST / RESPONSE MESSAGES	241
FIGURE 4-8 SSN CD SERVICES PROVIDED OPERATIONS.....	243
FIGURE 4-9 SSN CD COMPONENT DIAGRAM	243
FIGURE 4-10 CONNECTION DIAGRAM OF SSN IMDATE COMPONENTS EXCHANGE AIS MESSAGES.	251
FIGURE 4-11 CONNECTION DIAGRAM OF SSN IMDATE COMPONENTS EXCHANGE VOYAGE MESSAGES.....	252
FIGURE 4-12 SSN SERVICE PROVIDED OPERATIONS.	253
FIGURE 3—13 ARCHITECTURAL LAYER DEPENDENCIES (LOGICAL VIEW) OF STIRES-CORE	260
FIGURE 3—14 TRANSFORMATION ADAPTERS FOR EXCHANGED AIS MESSAGES	261
FIGURE 3-15 CONNECTION DIAGRAM OF VMS COMPONENTS.	268
FIGURE 3-16 SSN-SI VMS SERVICE PROVIDED OPERATIONS.	270
FIGURE 3-17 ARCHITECTURAL LAYER DEPENDENCIES (LOGICAL VIEW) OF VMS-PROXY	273
FIGURE 3-18 CONNECTION DIAGRAM OF SSN-SI COMPONENTS.	274
FIGURE 4-19 COMPONENT DIAGRAM PROVIDING A HIGH-LEVEL VIEW OF BOTH THE EXISTING AND THE NEW SYSTEM'S ARCHITECTURE	278
FIGURE 4-20 INTERACTION PATHS BETWEEN THE VARIOUS ELEMENTS OF THE NEW SYSTEM'S SPATIAL INFORMATION VISUALISATION SCHEME	280
FIGURE 4-21 COMMUNICATION DIAGRAM OF A TYPICAL MAP RENDERING CYCLE	281
FIGURE 4-22 CURRENT EIS NOTIFICATION REQUEST PROTOCOL MECHANISM (NUMBERS IN PARENTHESES ARE INDICATIVE OF THE SEQUENCE OF INFORMATION DELIVERY – OPERATIONS ARE OTHERWISE ASYNCHRONOUS)	282
FIGURE 4-23 PROPOSED EIS NOTIFICATION REQUEST SYNCHRONOUS PROTOCOL MECHANISM.....	283
FIGURE 5-1 SSN-EIS DEPLOYMENT MODEL.....	291
FIGURE 5-2 IMDATE DEPLOYMENT VIEW	295
FIGURE 5-3 VMS DEPLOYMENT VIEW	296

Table of Tables

TABLE 1-1: REFERENCE DOCUMENTS	10
TABLE 1-2: ABBREVIATIONS AND ACRONYMS	12
TABLE 3-1: USE CASES NOTATION	22
TABLE 4-1 SSN-SUPPORT.....	150
TABLE 4-2 SSN-DAO	150
TABLE 4-3 SSN-CORE.....	155
TABLE 4-4 SSN-CORE-EJB.....	156
TABLE 4-5 FUNCTION DESCRIPTION	195
TABLE 4-6 SSN RESOURCES AND VESSEL DAO	211
TABLE 4-7 SSN-RESOURCES-CORE	212
TABLE 4-8 SSN-XMLPROTOCOL-MAPPING	232
TABLE 4-9 SSN-XMLPROTOCOL-CORE.....	232
TABLE 4-10 SSN-XMLPROTOCOL-WEB	233
TABLE 4-11 SSN-XMLPROTOCOL-EJB.....	233
TABLE 4-12 TYPES OF MESSAGES PER QUEUE	284
TABLE 4-13 PROCESS MANAGER – ROUTING TABLE	285
TABLE 5-1 VMS-PROXY MINIMUM HARDWARE REQUIREMENTS.	296

1 Introduction

1.1 Purpose

This document defines the SSN system design document. Additional documents that complete the SSN system design specification are:

- GIDD-TI: Graphical Design Document – Textual Interface
- GIDD-GI: Graphical Design Document – Graphical Interface
- SDDB: System Database Design document

1.2 Scope

This document is the *System Design Document* (hereinafter the SDD) for SSN system. It presents a number of different architectural views to depict different aspects of the system. The purpose of this document is to present the technical details of the system components and more specifically:

- The definition of domain entities and the methods that implement the requested functionality.
- The creation of UML Class Diagrams and UML Sequence Diagrams, which associate classes and depict the overall flow of control within the system components respectively.

The primary intended audience of this document are system designers and system builders. The document intends to provide the members of the SSN project a unified view of the technical details of the system design to be followed during the development of the respective application. The document may need to be updated later to incorporate possible changes during development.

1.3 Reference documents

Id	Reference	Title	Version
R1	RUP Formal Resources	Rational Unified Process Formal Resources based on RUP Version: 2003.06.13	1.2
R2	N/A	Business Process Modelling Notation (BPMN)	1.2
R3	SSN-EIS-SRS	SSN EIS System Requirements Specifications	1.00
R4	SSN-XMLMessagingRefGuide	SSN XML Reference Guide	4.02
R5	Answer to RFS-EMSA-OP08-2015-SC#10-v1.00	System Requirements - SC#10 under FWC2015/EMSA/OP/18/2015	1.00
R6	SSN-SIG-PartB_ShipParticularsExchange	SSN SIG Part B. Ship particulars exchange	1.32
R7	SSN-SIG CLD	SSN SIG Location Data exchange	1.08
R8	SSN-SIG COD	SSN SIG Organisation Data exchange	1.33
R9	SSN SIGSTAR MRS	SSN SIGPart I – STAR MRS Notifications interface guide	0.91
R10	Answer to EMSA-NEG-16-2017	SSN SI and STIRES adaptations to process AIS data backlogs	-
R11	SSN-SIG-PartG-SEGVoyageExchange	SEG Voyage ExchangePart G – SEG Voyage Exchange	1.07

Id	Reference	Title	Version
R12	SSN-ICM	SafeSeaNet Installation and Configuration Manual	1.48

Table 1-1: Reference Documents

1.4 Abbreviations and acronyms

A list of the principal abbreviations and acronyms used in the document is provided here for a better understanding of this document.

Abbreviation	Definition
AIS	Automatic Identification Systems
CLD	Central Locations Database
COD	Central Organizations Database
CSD	Central Ship Database
EIS	European Index Server
EMSA	European Maritime Safety Agency
ER	Entity Relationship
ERD	Entity Relationship Diagram
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol over SSL
ID	Identification number
IdM V2	Identity and Access Management, version 2
IMO	International Maritime Organisation
ISO	International Organization for Standardization
LCA	LOCAL Competent Authority
MMSI	Maritime Mobile Service Identity
MRS	Mandatory Ship Reporting System
IMDatE	Integrated Maritime Data Environment
IMO	International Maritime Organisation
ISO	International Organization for Standardization
IoC	Inversion of Control
IE	Internet Explorer
II	INTRASOFT International

ITU	International Telecommunications Commission
ITU_1	AIS Message Type 1 AIS Vessel position report using SOTDMA (Self-Organizing Time Division Multiple Access). This is the most common AIS message type.
ITU_3	AIS Message Type 3 AIS Vessel position report using ITDMA (Incremental Time Division Multiple Access).
ITU_5	AIS Message Type 5 Ship static and voyage related data. This is the third-most common AIS message type. Due to its length it is generally a 2-part message.
JMS	Java Message Service
JNDI	Java Naming and Directory Interface; used for reference to Java EE components defined in WebLogic server.
JSON	JavaScript Object Notation is a lightweight data-interchange format
JSP	Java Server Pages
JSTL	JavaServer Pages Standard Tag Library
MMSI	Maritime Mobile Service Identity
MRS	Mandatory Ship Reporting System
MS	Member States.
MSS	Maritime Support Services
MSS Tool	Messages Availability MSS tool
MVC	Model-view-controller
N/A	Not Applicable or Not Available
NCA	National Competent Authority
OSD	Operational Ship Database
Req	Request
ROT	Rate of turn
RUP	Rational Unified Process
SAT	Ship Activity Tracking
SEG	SafeSeaNet Ecosystem GUI
SOG	Speed over ground
SPM	Ship Position Message
SQL	Structured Query Language
SSL	Secure Socket Layer
SSN	SafeSeaNet

TR	Table Reference
TRD	Table Reference Diagram
UML	Unified Modelling Language
URL	Unified Resource Locator
UTC	Coordinated Universal Time
UVI	Unique Vessel Identifier; the ship particulars used as UVI are: IMO, IR and EMSA-R; used for identification purposes in the CSD.
V&V	Vessel Verification and Validation
XML	eXtensible Markup Language

Table 1-2: Abbreviations and Acronyms

2 Architectural Goals and Constraints

This section describes the software requirements and objectives that have some significant impact on the architecture.

The SSN architecture is based on the following architectural principles and technologies:

2.1 Service-Oriented Architecture (SOA)

A SOA is an architecture principle that is based on the key concept of services. A service, in its simplest form, consists of an interface and an implementation. SOA defines software applications in terms of discrete services, which are implemented using service components that can be used to perform business activities for a given business process.

Service Component Architecture (SCA) is a set of specifications which describe a model for building applications and systems using a Service Oriented Architecture. SCA models solutions as sets of service components offering services and making references to services supplied by others, which are combined together by composites which wire references to services and which declaratively apply bindings for communication methods and also apply policies for aspects such as security and transactions. SCA extends and complements prior approaches to implementing services, and SCA builds on open standards such as Web services.

The Service Component Architecture (SCA) assembly model abstracts the implementation and allows assembly of components, with little implementation details. SCA facilitates the business logic representation as reusable service components that can be easily integrated into any SCA-compliant application. The resulting application is known as a SOA composite application.

It is possible to use an existing Spring application context as a component implementation in SCA. An SCA runtime (Weblogic SCA container) that supports Spring integration can use an application context as-is in an SCA assembly. For such a component it is possible to wire Spring services and references without the need to introduce SCA metadata into the Spring configuration. The Spring context needs to know very little about the SCA environment. Two points where the SCA metadata interacts with the Spring context are services and references. Any policy enforcement such as the provision of Security features is done by the SCA runtime on calls into the Spring application context before the final message is delivered to the target Spring bean. On outbound calls from the application context, references supplied by the SCA can provide policy enforcement.

It is also possible to specify SCA-related metadata as beans inside a Spring configuration. The Spring Component Implementation Specification makes it possible to specify:

- Spring beans that are made available to SCA as component services
- Spring beans that represent SCA properties
- Spring beans that represent SCA references

Three elements: `sca:service`, `sca:reference` and `sca:property`, can be used in a Spring application context configuration to identify a SCA service, a SCA reference or a SCA property, respectively.

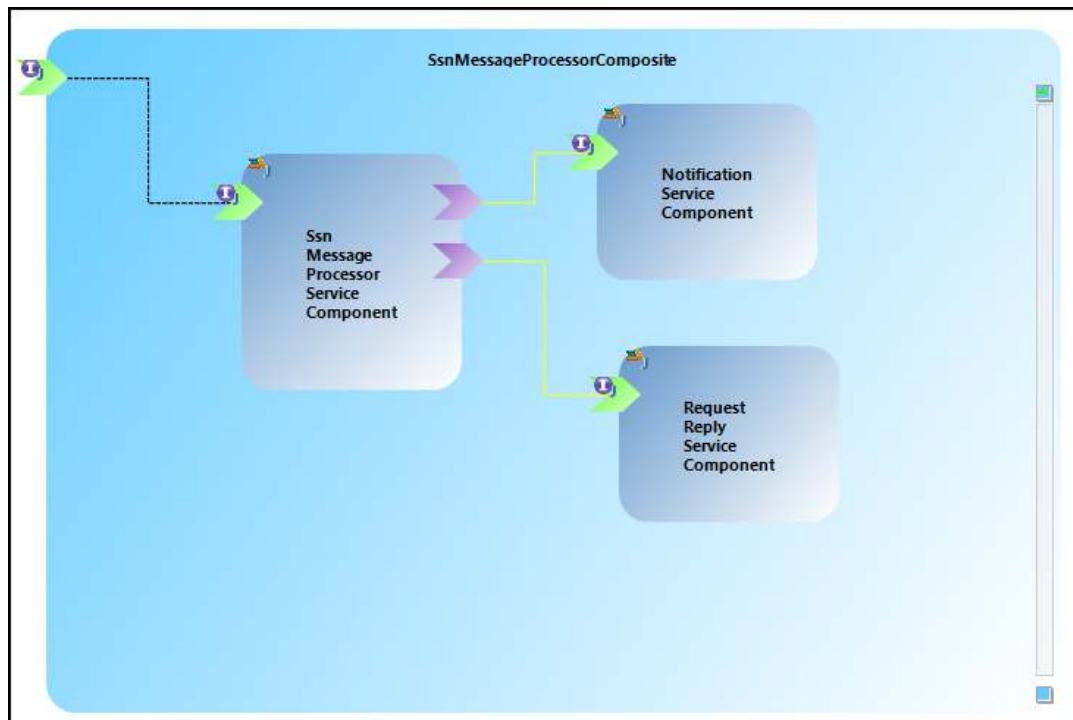


Figure 2-1: Example of EIS SCA prototype based on existing Spring application

2.2 Java EE Technologies

Java Platform, Enterprise Edition 6 (Java EE 6) is used for SSN system implementation.

Key technologies in Java EE 6 include the following:

Presentationlayer

- Java Servlet
- JavaServer Faces
- Web application internationalization and localization

Enterprise JavaBeans (EJB) 3.2.

- Session beans
- Message-driven beans
- Timer Services

Platform services

- Transactions
- Resource connections
- Security
- Java Message Service (JMS)

2.3 Industry Standards

Web Services including SOAP and XML.

Apache [Maven](#) v3.6.0 tool is used for projects' build.

2.4 Business Processes

The methodology used for gathering all business processes identified on the —Business Process Modelling Notation, also called BPMN (reference [R2]).

The primary goal of the BPMN effort is to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes.

Thus, BPMN creates a standardized bridge for the gap between the business process design and process implementation.

BPMN defines a Business Process Diagram (BPD), which is based on a flowcharting technique tailored for creating graphical models of business process operations.

A Business Process Model, then, is a network of graphical objects, which are activities (i.e., work) and the flow controls that define their order of performance.

2.5 Open Source Frameworks

2.5.1 Spring framework

Spring Framework is a Java platform that provides comprehensive infrastructure support for developing Java applications. Spring facilitates the applications building from “plain old Java objects” (POJOs) and to apply enterprise services non-invasively to POJOs. This capability applies to the Java SE programming model and to full and partial Java EE.

Some of the Spring platform advantages are:

- Make a Java method execute in a transaction without having to deal with transaction APIs.
- Make a local Java method a remote procedure without having to deal with remote APIs.
- Make a local Java method a message handler without having to deal with JMS APIs.

Spring Framework version 4.3 shall be used; Java 8 fully supported.

2.5.1.1 Spring Web MVC framework

The Spring web MVC framework provides model-view-controller architecture and ready components that can be used to develop flexible and loosely coupled web applications. The MVC pattern results in separating the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements.

- The Model encapsulates the application data and in general they will consist of POJO.
- The View is responsible for rendering the model data and in general it generates HTML output that the client's browser can interpret.
- The Controller is responsible for processing user requests and building appropriate model and passes it to the view for rendering.

2.5.1.2 Spring Web Flow

Spring Web Flow is a Spring MVC extension that allows implementing the “flows” of a web application. A flow encapsulates a sequence of steps that guide a user through the execution of some business task. It spans multiple HTTP requests, has state, deals with transactional data, is reusable, and may be dynamic and long-running in nature.

The sweet spot for Spring Web Flow are stateful web applications with controlled navigation such as checking in for a flight, applying for a loan, shopping cart checkout, or even adding a confirmation step to a form. What these scenarios have in common is one or more of the following traits:

- There is a clear start and an end point.

- The user must go through a set of screens in a specific order.
- The changes are not finalized until the last step.
- Once complete it shouldn't be possible to repeat a transaction accidentally.

Spring Web Flow provides a declarative flow definition language for authoring flows on a higher level of abstraction. It allows it to be integrated into a wide range of applications without any changes (to the flow programming model) including Spring MVC, JSF, and even Portlet web applications.

2.5.1.3 Spring Security

Spring Security is a powerful and highly customizable authentication and access-control framework. It is the standard for securing Spring-based applications. It provides a comprehensive security solution for Java EE-based enterprise software applications.

2.5.1.4 Service layer: Spring Web Services

Spring Web Services is a product of the Spring framework focused on creating document-driven Web services. Spring Web Services aims to facilitate contract-first SOAP service development, allowing for the creation of flexible web services using one of the many ways to manipulate XML payloads.

Spring-WS supports multiple transport protocols. The most common is the HTTP transport, for which a custom servlet is supplied, but it is also possible to send messages over JMS, and even email.

The Spring-WS also supports XML API and XML marshalling and un-marshalling.

The advantages of using Spring-WS are:

- Spring based.
- Pluggability.
- Focus on SOAP.
- Use available implementations.
- Sensible defaults.
- Fully message based.

Spring-WS are being deployed as simple Web Modules. The Web Module defines a Dispatcher Servlet which is an alternative to the standard Spring-MVC Dispatcher Servlet with separate Adapters for the messages and the wsdl definitions. The Servlet detects automatically any wsdl definition defined in its application context. The wsdl is exposed under its bean name. The servlet also detects Endpoint Adapters which are interfaces implemented for each endpoint type in order to handle separate SOAP requests.

The Server side of Spring-WS is designed around a central class that dispatches incoming XML messages to endpoints. The Spring-WS's MessageDispatcher is extremely flexible, allowing the use of any sort of class to an endpoint, as long as it can be configured in the spring IoC container.

EndpointAdapter provides the interface that must be implemented for each endpoint type to handle a message request. In Spring-WS, Endpoints are handling incoming XML messages. Message endpoints give access to the entire XML message including SOAP headers. The payload is simply the contents of the SOAP body. How incoming messages are routed to these endpoints is the responsibility of and EndpointMapping. The routing can be done based on different criteria. Spring-WS offers many out of the box implementations not only for the mapping criteria but also for the endpoint implementations.

Spring Web Services supports server-side JMS handling through the JMS functionality provided in the Spring framework. Spring Web Services provides the WebServiceMessageListener to plug in to a MessageContainer. This message listener requires a WebServiceMessageFactory to and MessageDispatcher to operate.

The Spring-WS includes also JMS transports. Besides the standard JMS configuration(connection factory and destination name to listen to), we only have to define a WebServiceMessageListener and give it a reference to the message factory we are using and the message dispatcher.

2.5.1.5 Platform services: Spring JMS

The JMS API exposes two types of send methods, one that takes delivery mode, priority, and time-to-live as Quality of Service (QOS) parameters and one that takes no QOS parameters which uses default values. Since there are many send methods in `JmsTemplate`, the setting of the QOS parameters has been exposed as bean properties to avoid duplication in the number of send methods. Similarly, the timeout value for synchronous receive calls is set using the property `setReceiveTimeout`. Some JMS providers allow the setting of default QOS values administratively through the configuration of the `ConnectionFactory`. This has the effect that a call to `MessageProducer`'s send method *send(Destination destination, Message message)* will use different QOS default values than those specified in the JMS specification. To provide consistent management of QOS values, the `JmsTemplate` must therefore be specifically enabled to use its own QOS values by setting the boolean property `isExplicitQosEnabled` to true.

2.5.2 Hibernate framework

Hibernate is an object-relational mapping (ORM) library for the Java language, providing a framework for mapping an object-oriented domain model to a traditional relational database. Hibernate solves object-relational impedance mismatch problems by replacing direct persistence-related database accesses with high-level object handling functions.

Hibernate's primary feature is mapping from Java classes to database tables (and from Java data types to SQL data types). Hibernate provides the development of persistent classes following common Java idiom - including association, inheritance, polymorphism, composition and the Java collections framework. Hibernate also provides data query and retrieval facilities. Hibernate generates the SQL calls and relieves the developer from manual result set handling and object conversion, keeping the application portable to all supported SQL databases.

Hibernate provides a variety of the characteristics that SDO provides. For example, Hibernate provides the convenient static data APIs, the optimistic concurrency, disconnected model. Hibernate framework could be extended to become SDO-capable data access services, which allows this framework to work within the SDO solution.

2.5.3 ESAPI (The OWASP Enterprise Security API)

ESAPI (The OWASP Enterprise Security API) is a free, open source, web application security control library that makes it easier for programmers to write lower-risk applications. The ESAPI libraries are designed to make it easier for programmers to retrofit security into existing applications. The ESAPI libraries also serve as a solid foundation for new development.

Allowing for language-specific differences, all OWASP ESAPI versions have the same basic design:

- There is a set of security control interfaces. They define for example types of parameters that are passed to types of security controls.
- There is a reference implementation for each security control. The logic is not organisation-specific, and the logic is not application-specific. An example: string-based input validation.
- There are optionally your own implementations for each security control. There may be application logic contained in these classes which may be developed by or for your organisation. An example: enterprise authentication.

2.6 RESTful services using Jersey

Jersey is open source, production quality, framework for developing RESTful Web Services in Java that provides support for JAX-RS APIs and serves as a JAX-RS (JSR 311 & JSR 339) Reference Implementation. Jersey extension that supports Spring DI shall be also used to enable Jersey to use Spring beans as JAX-RS components (e.g. resources and providers) and also allow Spring to inject into Jersey managed components.

Jersey framework version 5.2.1 shall be used that is provided by JEE application server (Oracle WebLogic Server 12c – 12.1.3).

2.7 XML Web Services (JAX-WS)

The Java API for XML Web Services (JAX-WS) is a Java programming language API for creating web services, particularly SOAP services. JAX-WS is one of the Java XML programming APIs. It is part of the Java EE platform.

The JAX-WS 2.2 specification JSR 224 defines a standard Java- to-WSDL mapping which determines how WSDL operations are bound to Java methods when a SOAP message invokes a WSDL operation.

JAX-WS uses annotations, introduced in Java SE 5, to simplify the development and deployment of web service clients and endpoints. JAX-WS 2.0 replaced the JAX-RPC API in Java Platform, Enterprise Edition 5 which leans more towards document style Web Services.

Standards Supported are:

- JAX-WS 2.0/2.1/2.2 (JSR 224)
- WS-I Basic Profile 1.2 and 2.0
- WS-I Attachments Profile 1.0
- WS-I Simple SOAP Binding Profile 1.0
- WS-Addressing 1.0 - Core, SOAP Binding, WSDL Binding.

2.8 Web features

2.8.1 HTML5

[HTML5](#) is a markup language for structuring and presenting content for the World Wide Web and a core technology of the Internet. Its core aims have been to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices (web browsers, parsers, etc.). HTML5 is intended to subsume not only HTML 4, but also XHTML 1 and DOM Level 2 HTML.

Following its immediate predecessors HTML 4.01 and XHTML 1.1, HTML5 is a response to the observation that the HTML and XHTML in common use on the World Wide Web are a mixture of features introduced by various specifications, along with those introduced by software products such as web browsers, those established by common practice, and the many syntax errors in existing web documents. It is also an attempt to define a single markup language that can be written in either HTML or XHTML syntax. It includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalises the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications. For the same reasons, HTML5 is also a potential candidate for cross-platform mobile applications. Many features of HTML5 have been built with the consideration of being able to run on low-powered devices such as smartphones and tablets.

In particular, HTML5 adds many new syntactic features as the new <video>, <audio> and <canvas> elements, as well as the integration of scalable vector graphics (SVG) content (that replaces the uses of generic <object> tags) and MathML for mathematical formulas. These features are designed to make it easy to include and handle multimedia and graphical content on the web without having to resort to proprietary plugins and APIs. Other new elements, such as <section>, <article>, <header> and <nav>, are designed to enrich the semantic content of documents. New attributes have been introduced for the same purpose, while some elements and attributes have been removed. Some elements, such as <a>, <cite> and <menu> have been changed, redefined or standardized. The APIs and Document Object Model (DOM) are no longer afterthoughts but are fundamental parts of the HTML5 specification. HTML5 also defines in some detail the required processing for invalid documents so that syntax errors will be treated uniformly by all conforming browsers and other user agents.

2.9 Non-functional requirements

The non-functional requirements are addressed separately in terms of the architectural solution offered:

- Portability: The technical architecture solution is based on standard Java EE technologies. Any application server specific extensions/frameworks should be avoided where possible. Non-portable parts of the SSN System, if applicable, should be isolated and documented.
- Scalability: The design of the system and selection of technologies support a scalable solution.

In particular, some SSN services components may be deployed onto a separate host machine to address this requirement. However, considering the WebLogic Server Clustering feature, a homogenous SSN system deployment will provide increased performance due to the lack of remote calls between the SSN components.

Additionally, the SSN System includes software frameworks/products that are known to be scalable.
- Reusability: A design based on components lends support for reusability. In addition, exposing these components as web services should further support this requirement.
- Modularity: The SSN System is delivered as software components.
- Maintainability: The components are loosely coupled therefore they should require less maintenance overhead.
- Availability: The technical solution allows for the SSN System to be deployed onto more than one application server to meet availability demands.
- Performance: The design of the SSN System and selection of software products/frameworks has been undertaken with performance in mind. As the SSN System may be deployed into different environments, one possible solution to addressing this requirement may be offered through the scalability capabilities of the application as it allows the workload to be distributed to more than one SSN application and therefore increasing available resources allocated to service user requests.
- Security: The infrastructure (router, reverse proxy, application server) hosts the SSN components shall provide security techniques for secure communication using TLS protocol 1.0 or later.

[OWASP Enterprise Security API \(ESAPI\) Project](#) shall be used on the implementation of NSW web applications. The targeted level for the NSW is OWASP Level 2.

2.9.1 JavaServer Faces

JavaServer Faces (JSF) is a user interface (UI) framework for Java web applications. It is designed to significantly ease the burden of writing and maintaining applications that run on a Java application server and render their UIs back to a target client. JSF provides ease-of-use in the following ways:

- Makes it easy to construct a UI from a set of reusable UI components
- Simplifies migration of application data to and from the UI
- Helps manage UI state across server requests
- Provides a simple model for wiring client-generated events to server-side application code
- Allows custom UI components to be easily built and re-used

Most importantly, JSF establishes standards which are designed to be leveraged by tools to provide a developer experience which is accessible to a wide variety of developer types, ranging from corporate developers to systems programmers. A "corporate developer" is characterized as an individual who is proficient in writing procedural code and business logic but is not necessarily skilled in object-oriented programming. A "systems programmer" understands object-oriented fundamentals, including abstraction and designing for re-use. A corporate developer typically relies on tools for development, while a system programmer may define his or her tool as a text editor for writing code.

Therefore, JSF is designed to be toolled, but also exposes the framework and programming model as APIs so that it can be used outside of tools, as is sometimes required by systems programmers.

2.10 Application Server

The WebLogic Server 12c is used to host SSN system, which provides the following functionality

- clustering feature provides increased scalability, availability and reliability
- WebLogic Server is compliant with Java Secure Socket Extension (JSSE). JSSE is a set of packages that support and implement the SSL and TLS v1 protocol.

WebLogic Server provides Secure Sockets Layer (SSL) support for encrypting data transmitted across WebLogic Server clients, as well as other servers.

WebLogic Server supports the RSA cipher suites listed in COTS related documentation.

The current implementation of SSN uses Java platform.

Components used:

JDK 1.8

2.11 Database

Oracle RDBMS Server 12.1.0.2 shall be used for SSN data storage; Oracle Exadatatechnology provides fault tolerance, security, load balancing, and scalability.

3 Overall System Architecture

3.1 Definitions

Concept	Definition
UVI (Unique Vessel Identifier)	Unique Vessel Identifier. This is the way of referring to vessel attributes that are used for identification purposes in the CSD. The vessel attributes used as UVI are: IMO, IR and EMSA-R.
Vessel attributes	Are either vessel identifiers or vessel particulars.
Vessel identifiers	IMO Number, MMSI Number, CallSign, ShipName, IR Number, EMSA-R Number, Flag Registry, XR Number.
Vessel particulars	All attributes available for a vessel other than the identifiers.
Vessel version	<p>Refers to old and the latest versions of a vessel.</p> <p>A new version is created when one of the UVI, MMSI, CallSign and/or ShipName is updated.</p> <p>A version of a vessel holds the value of each attribute at a given timestamp. Vessel versions can have status:</p> <ul style="list-style-type: none">– Temporary: a vessel that is created temporarily until it undergoes manual intervention to be verified and validated.– Validated: a vessel that has been verified and is considered to be valid.– Invalid: a vessel that has been verified and is considered to be invalid. <p>The statuses Temporary and Invalid are only used in the OSD.</p>
Consolidated vessel version	The latest valid vessel version.
Active/Inactive	Defines whether the vessel is sea worthy (Active) or not (Inactive).

3.2 Functional Architecture

For the purposes of the use case definition the following actors and external systems are identified:

- Data provider (human via web or system via XML/SOAP) represents the external systems (National SSN or SSN Ecosystem application) that submit messages to SSN Central system.
- The national administrator (human) is in charge of the management of vessels, and configuration of the SSN Central application.
- SSN Central (system) will provide the revamped PortPlus message exchange services and ShipCall requests with National SSN or SSN Ecosystem application.

For identifying the functionality, the typical UML use cases notation is used:




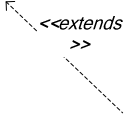

Symbol	Description
	Use Case <ul style="list-style-type: none"> Represents a discrete unit of interaction between a user (human or machine) and the system. Each Use Case has a description which describes the functionality that will be built in the proposed system. A Use Case may 'include' another Use Case's functionality or extend another Use Case with its own behaviour. Use Cases are typically related to 'actors'.
	Actor <ul style="list-style-type: none"> Human or machine entity that interacts with the system to perform meaningful work.
	Association <ul style="list-style-type: none"> A relationship between two or more entities. Implies a connection of some type, for example one entity uses the services of another or one entity is connected to another over a network link.
	Extends Relationship <ul style="list-style-type: none"> A relationship between two use cases in which one use case extends the behaviour of another.
	Includes Relationship <ul style="list-style-type: none"> A relationship between two use cases in which one use case includes the behaviour.

Table 3-1: Use Cases Notation

The following diagram provides an overview of the Use Cases.

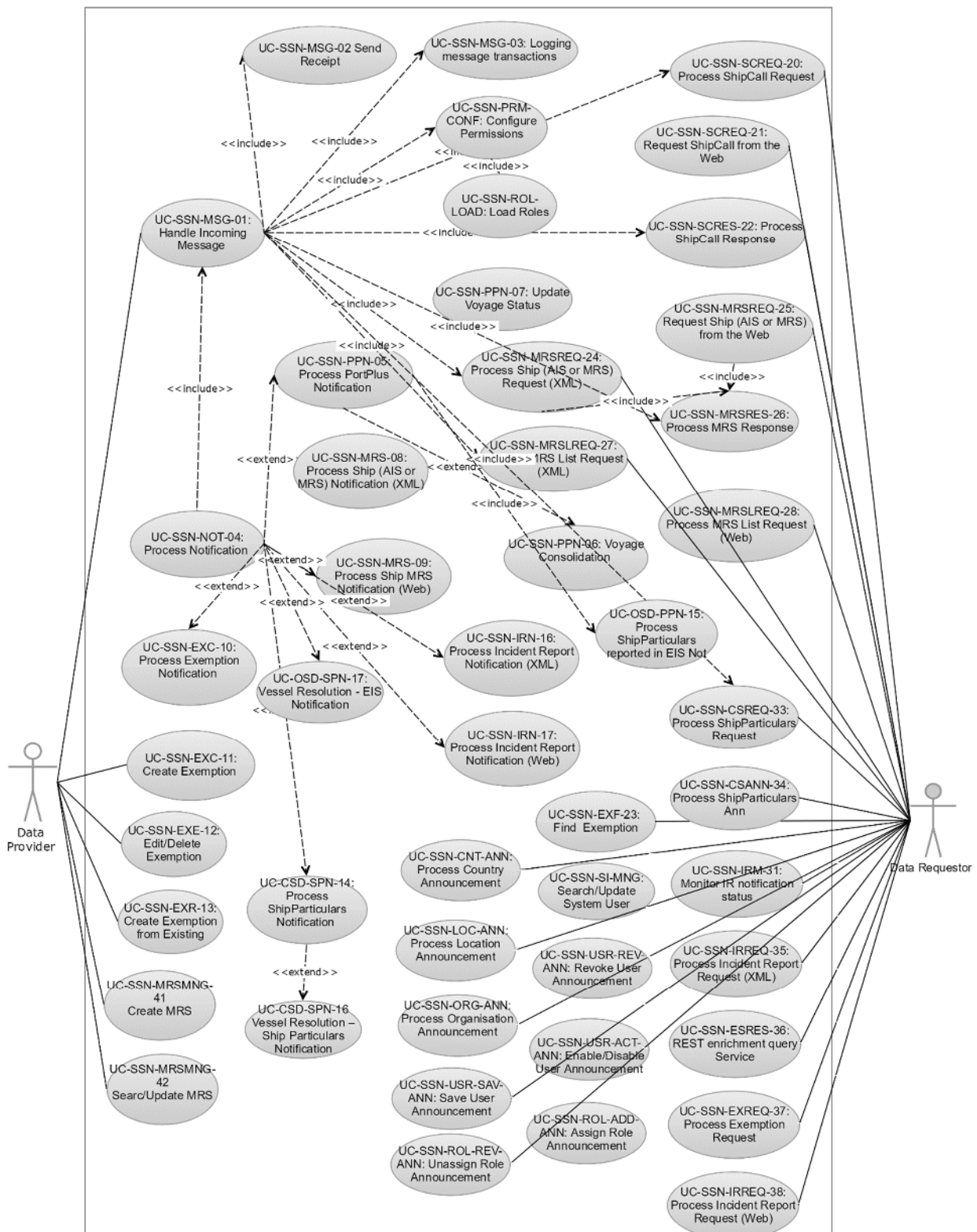


Figure 3-1 SSN specific Use Case diagram 1

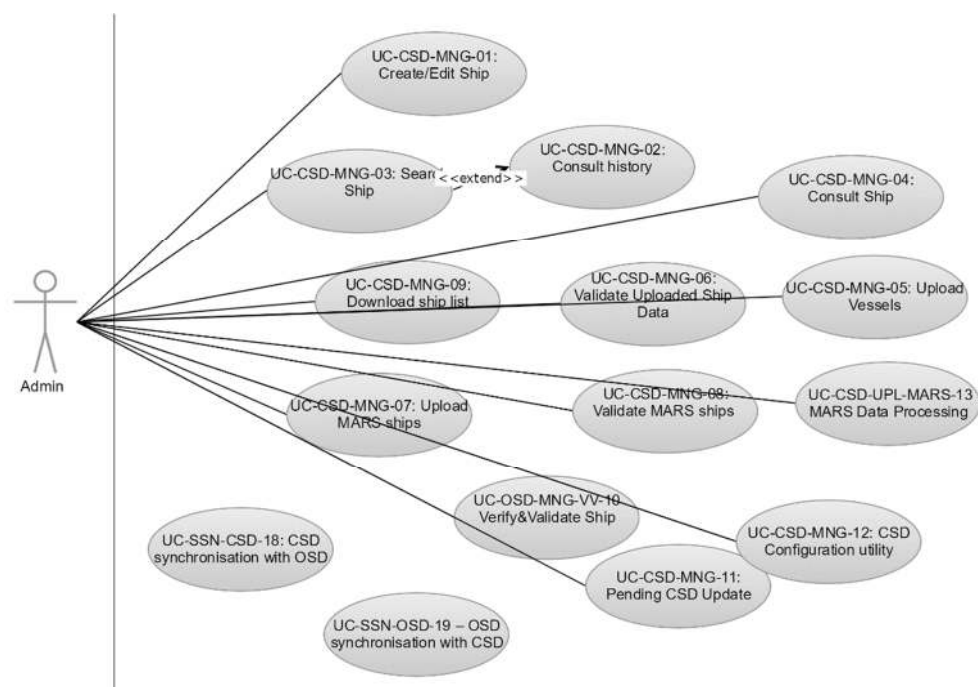


Figure 3-2 SSN specific Use Case diagram 2

3.2.1 Handle Incoming Message

This system package includes the services required in order for the SSN Central to process messages via the XML/SOAP. This system package consists of the following use-cases:

3.2.1.1 UC-SSN-MSG-01: Handle Incoming Message

Use Case Req ID	UC-SSN-MSG-01	
Use Case Name	Handle Incoming Message	
Purpose	Covers the functionality related to the system's actions upon receiving a message either from Web or National Application or SSN Ecosystem application.	
Subsystem	SSN Core	
Primary Actor(s)	SSN Human User / SSN System User	
Precondition(s)	A message either from a Human User via the web interface or from a System User via the XML interface has been received by the system.	
Postcondition(s)	System has identified the message scope.	
Trigger(s)	Message Reception from Actor.	
Use Case Description	Primary Scenario	
Step 1.	The system upon reception of the message checks its integrity and validates the message according to the applicable Reference Guide. The message header attribute Version has value '3' or '4'.	

Step 2.	After successful integrity check, the system identifies the scope and type of the received message (Notification/Information Request/Response, Announcement).
Step 3.	<p>After the type of message is identified, the system determines its actual originator.</p> <p>In case of xml messages submitted by SSN Ecosystem Application to EIS ("ES2SSN-SSN2ES" messages via HTTP/XML or HTTP/SOAP), the actual originator is defined by a new "RequestOriginator" or "NotificationOriginator" element. It is used to identify the user ID of the end-user of the application of the SSN Ecosystem who is the actual originator of the message.</p> <p>In any other case, the actual originator is the message "Sender".</p>
Step 4.	<p>Originator is authorised for accessing to the system;</p> <p>More precisely</p> <ul style="list-style-type: none"> • The originator is known to SSN application. • The originator is active. • • The sender is authorized to send messages from interface (e.g. XML/SOAP). The message "Sender" is used for interface validation. • The client certificate name of the message "Sender", in case it is defined in SSN application for the validated interface, is specified in the "CN" parameter of the incoming message header*. • The originator is granted the required permissions for sending such message types. <p>* - The message is only rejected in case that the CN parameter is defined in SSN application for the message sender and does not match the CN parameter coming in the message header. In any other case the validation of the CN will be successful.</p>
Step 5.	SSN time stamps (in UTC) the incoming message with the time of receipt.
Step 6.	Sends a receipt to the SSN User that sent the message.
Alternative Use Case Description	Invalid Message Structure
Step 1.1	The system identifies that the message is not correctly structured.
Step 1.2	Logs the invalid message and sends a receipt to the SSN User synchronously.
Alternative Use Case Description	Unidentified message type
Step 2.1	The system cannot identify the received message as one of the predefined message types.
Step 2.2	Go to 1.2
Alternative Use Case Description	Not Authorised User

Step 3.1	The system identifies that the originator is not authorised to send such message types.
Step 3.2	Go to 1.2
Alternative Use Case Description	Not Unique MsRefId
Step 4.1	The system identifies that Message Reference Id of an XML message received from an NCA Application is not unique.
Step 4.2	Go to 1.2
Alternative Use Case Description	Tracking Service Request
Step 5.1	If there is a tracking service request associated to the specific message type, the system shall send the requested email to the subscribers.
Input(s)	XML Message / Information submitted from Web.
Output(s)	Identified scope of Message; the message content is logged (EIS_ADMIN.LOG_XML_CONTENTS); the processing status is logged (EIS_ADMIN.SSN_LOGS).
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	Included Case: UC-SSN-MSG-02 Send Receipt Included Case: UC-SSN-MSG-03: Logging message transactions Included Case: UC-SSN-PRM-CONF: Configure Permissions
Special Requirements	.

3.2.1.2 UC-SSN-MSG-02: Send Receipt

Use Case Req ID	UC-SSN-MSG-02	
Use Case Name	Send Receipt	
Purpose	The current use case describes the system's functionality related to the receipt sending for a received message.	
Subsystem	SSN Core	
Primary Actor(s)	SSN Human User/SSN System User	
Precondition(s)	A new message has been received from the system either through the SSN Web or XML Interface.	
Postcondition(s)	A receipt message is sent to the sender of a message.	
Trigger(s)	Validation results of a received message.	
Use Case Description	Primary Workflow	
Step 1	The system generates a receipt which acknowledges that the XML message is received and assigns the StatusCode="OK" if the message was valid or StatusCode="InvalidFormat" if the message was invalid.	

Step 2	The system sends the receipt to the System User's Application.
Input(s)	Validation Results.
Output(s)	Receipt message.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	

3.2.1.3 UC-SSN-MSG-03: Logging message transactions

Use Case Req ID	UC-SSN-MSG-03	
Use Case Name	Logging message transactions	
Purpose	The current use case describes the system functionality related to the logging of message transactions of SafeSeaNet events.	
Subsystem	Log Mechanism	
Primary Actor(s)	SSN Administrator	
Precondition(s)	System is operational and may send and accept messages.	
Postcondition(s)	Information related to the exchanged message is inserted in the system's log file.	
Trigger(s)	Any information exchange performed through the SafeSeaNet (Message Exchange, Receipts sent and received, Acknowledgments, Storage in database).	
Use Case Description	Primary Workflow	
Step 1	<p>The system updates the log database with the exchanged data after each message related transaction of the system with the SSN Human Users/SSN System Users. Data stored include:</p> <ul style="list-style-type: none"> • Message type (if XML compliant message or valid REST method) • Sender (if XML compliant message or valid REST method) • Timestamp • MsRefId or ESRefId of the message (if XML compliant message) • XML content (if any) 	
Step 2	The SSN Admin is provided with access to the systems' logged information through the EIS database.	
Input(s)	Information related to message exchange.	
Output(s)	Updated log information.	
Timer(s)	-	
Business Process(es) Reference		
Associated Use Case(s)	-	

Special Requirements	The system keeps the log information in the EIS database.
----------------------	---

3.2.1.4 UC-SSN-PRM-CONF: Configure Permissions

Use Case Req ID	UC-SSN-PRM-CONF	
Use Case Name	Configure Permissions	
Purpose	Covers the functionality related to configuring actor's permissions and their respective restrictions, required for further authorization.	
Subsystem	SSN Core	
Primary Actor(s)	SSN Human User / SSN System User	
Precondition(s)	<ul style="list-style-type: none"> Actor is known to SSN application and its IdM roles have been retrieved from SSN database. Definition of IdM roles mapped to a set of tasks and respective restrictions is loaded in "memory". 	
Postcondition(s)	System has resolved the actor's permissions and restrictions.	
Trigger(s)	A message from a SSN System User application has been received by the system or an authenticated SSN Human User has accessed SSN Console application.	
Use Case Description	Primary Scenario	
Step 1.	For each IdM role assigned to the actor, the system retrieves from memory the list of permissions together with their source and location restriction level information.	
Step 2.	For each permission that is included in several IdM roles assigned to the Actor, the system applies the following rules to determine source and location restrictions: <ul style="list-style-type: none"> If the restriction levels are not equal, the system will consider the one with the lowest (less narrow/limited) level of geographical restriction (None < Countries list < User's country < Specific locations). If the restriction levels are equal, the system will apply the union of each restriction (i.e. sum of countries or sum of duty codes). 	
Step 3.	The system, upon determining source and location restrictions, assigns the retrieved permissions to the actor.	
Input(s)	XML Message / SSN Console application startup.	
Output(s)	Permissions and their restrictions are resolved and assigned to actor.	
Timer(s)	-	
Business Process(es) Reference	-	
Associated Use Case(s)	UC-SSN-ROL-LOAD: Load Roles	
Special Requirements		

3.2.1.5 UC-SSN-ROL-LOAD: Load Roles

Use Case Req ID	UC-SSN-ROL-LOAD	
Use Case Name	Load Roles	
Purpose	Covers the functionality related to loading IdM roles together with their corresponding set of tasks and restrictions in SSN application "memory".	
Subsystem	SSN Core	
Primary Actor(s)	SSN Core	
Precondition(s)	N/A	
Postcondition(s)	System has stored/cached in "memory" the definition of IdM roles mapped to a set of tasks and respective restrictions.	
Trigger(s)	SSN Core application startup.	
Use Case Description	Primary Scenario	
Step 1.	The system opens an input stream for reading the configuration file resource.	
Step 2.	For every line read, the system parses its contents to extract the following information: <ul style="list-style-type: none"> • Role; • Task; • Source restriction level (country of the provider of the data): "None" (default value), "Countries list" or "User's country"; • Source regional agreement code, if the Source restriction level is "Countries list"; • Location restriction level (location to which the data concerns): "None" (default value), "Countries list", "User's country" or "Specific locations" • Regional agreement code if the Location restriction level is "Countries list" or Duty code if the location restriction level is "Specific locations". 	
Step 3.	The system constructs an in "memory" map data structure that accepts each IdM role as key and a collection of tasks together with their respective restrictions as mapped value.	
Alternative Use Case Description	I/O error occurs while reading configuration file resource.	
Step 1.1	A runtime exception with the specified cause and a detail message of cause will be logged at the ERROR level and thrown on SSN Core application startup (stopping the startup process).	
Input(s)	Configuration file providing the definition of IdM roles mapping to tasks and restrictions.	
Output(s)	IdM roles and their corresponding set of tasks and restrictions stored/cached in "memory".	
Timer(s)	-	
Business Process(es) Reference	-	

Associated Use Case(s)	-
Special Requirements	The configuration file that allows mapping of IdM roles to tasks and restrictions is included in SSN application's classpath.

3.2.2 Data Provider

This system package includes the services required in order for the SSN Central to process notification messages via the XML/SOAP and Web interfaces. More specifically, this system package consists of the following use-cases:

3.2.2.1 UC-SSN-NOT-04: Process Notification

Use Case Req ID	UC-SSN-NOT-04	
Use Case Name	Process Notification	
Purpose	Covers the functionality related to notification processing.	
Subsystem	SSN Core	
Primary Actor(s)	SSN Human User / SSN System User	
Precondition(s)	Handle Incoming Message UC has identified the incoming message as a valid Generic Notification.	
Postcondition(s)	System sends a receipt to the Data Provider about the acceptance of the notification and the transaction is logged.	
Trigger(s)	Notification Received	
Use Case Description	Primary Scenario	
Step 1	<p>The system identifies that the following information are contained in the notification:</p> <ul style="list-style-type: none"> • Data Provider; • Vessel Identification (IMO, IRNumber for Incident Report Notifications, MMSI, Call Sign, Ship Name). IMO and/or MMSI should be present (except for Incident Report Notifications). <p>The system also identifies that the received notification has one of the following types:</p> <ul style="list-style-type: none"> • PortPlus notification; • Ship notification (AIS or MRS); • Incident Report notification; • Exemption notification. 	
Step 2	System executes the processes foreseen in UC-OSD-PPN-15 (Process Ship Particulars reported in EIS notification) – refer to primary scenario	
Step 3	The system validates the notification applying the business rules for the specific Notification type (as defined in [R4]) and checks whether the Actor is Activated.	

Step 4	The system persists the extracted information in the database and sends the Acknowledgement receipt to the data provider to indicate the successful reception and processing of the message details. Note that if during the process in step 2 is detected that ship is SHT or banned a warning will be included in the SSN_Receipt provided back to the notification provider.
Alternative Use Case Description	Unidentified Vessel
Step 1.1	In case of a notification of type Ship, PortPlus, IR for identified vessel or Exemption the vessel information is missing both IMO or IRNumber (only for Incident Report Notifications), and MMSI numbers.
Step 1.2	The system sends an error receipt to the data provider.
Alternative Use Case Description	Validation Failure
Step 3.1	If the system fails to validate the received notification against the business rules for the identified notification type.
Step 3.2	Notification is rejected (The system sends an error receipt to the data provider)
Alternative Use Case Description	No Vessel in EISDB
Step 4.1	As step 1 in primary scenario
Step 4.2	System executes the processes foreseen in UC-OSD-PPN-15 (Process Ship Particulars reported in EIS notification) – refer to the alternative use case description (steps 2.1, 2.2, 2.3)
Step 4.3	As Step 3 in primary scenario
Step 4.4	As Step 4 in primary scenario
Alternative Use Case Description	The Vessel in SSN DB is updated (Ship reported UVI is listed in SSN OSD, but the ship identity is updated (her MMSI changed))
Step 5.1	As step 1 in primary scenario
Step 5.2	System executes the processes foreseen in UC-OSD-PPN-15 (Process Ship Particulars reported in EIS notification) – refer to the alternative use case description (steps 3.1, 3.2, 3.3)
Step 5.3	As Step 3 in primary scenario
Step 5.4	As Step 4 in primary scenario
Alternative Use Case Description	Ship reported UVI is listed in SSN OSD - The incoming ship identifiers match those of a temporary vessel in the OSD.
Step 6.1	As step 1 in primary scenario
Step 6.2	System executes the processes foreseen in UC-OSD-PPN-15 (Process Ship Particulars reported in EIS notification) – refer to the alternative use case description (steps 4.1, 4.2, 4.3)

Step 6.3	As Step 3 in primary scenario
Step 6.4	As Step 4 in primary scenario
Alternative Use Case Description	Unidentified Port
Step 7.1	<p>In case of Ship, PortPlus and Exemption notification if the system fails to identify the Ports ("Next Port of Call", "Port of Call", "Last Port", ContactSource.Locode) in the SSN database the system shall check whether the LOCODE is correctly structured.</p> <p>In case the LOCODE is valid the system shall store it in the database as temporary and accept the notification.</p> <p>In case the LOCODE is invalid the system shall reject the notification.</p>
Step 7.2	Go to Step 4 of the primary workflow.
Input(s)	New Notification
Output(s)	Notification passed to appropriate UC (PortPlus, Ship, Incident Report, Exemption)
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	<p>Included Case: UC-SSN-MSG-01: Handle Incoming Message, UC-CSD-SPN-14: Process Ship Particulars reported in EIS notification</p> <p>Extended by: UC-SSN-PPN-05: Process PortPlus Notification, UC-SSN-MRS-08: Process Ship AIS orMRS Notification (XML), UC-SSN-MRS-09:Process Ship MRS Notification (Web), UC-SSN-IRN-16: ProcessIncident Report Notification (XML), UC-SSN-IRN-17: Process Incident Report Notification (Web), UC-SSN-EXC-10: Process Exemption Notification, UC-SSN-EXC-11: Create Exemption, UC-SSN-EXC-12: Edit/Delete Exemption, UC-SSN-EXR-13:Create Exemption from Existing</p>
Special Requirements	Submission of notification can be achieved either through XML or Web interface. AIS and PortPlus Notifications are only submitted through the XML interface.

3.2.2.2 UC-SSN-PPN-05: Process PortPlus Notification

Use Case Req ID	UC-SSN-PPN-05
Use Case Name	Process PortPlus Notification
Purpose	Covers the functionality related toPortPlus Notification processing.
Subsystem	SSN Core
Primary Actor(s)	SSN System User
Precondition(s)	Process Notification UC has identified the incoming message as a valid PortPlus Notification.
Postcondition(s)	System persists Data in the SSN Database
Trigger(s)	Notification Received

Use Case Description	Valid New PortPlus Notification
Step 1	<p>The system identifies access rights restrictions on PortPlus notifications, based on the following:</p> <ul style="list-style-type: none"> On the PORTPLUS_NOTIFIER task and geographical location restrictions based on the PortOfCall. <ul style="list-style-type: none"> In case the PORTPLUS_NOTIFIER permission has a "Specific locations" location restriction, PortOfCall shall be within the locodes assigned to the applicable duty/ies for the actor's organisation. In case the PORTPLUS_NOTIFIER permission has a "User's country" location restriction, PortOfCall shall belong to actor's country; In case the PORTPLUS_NOTIFIER permission has a "Countries list" location restriction, PortOfCall shall belong to the countries assigned to the applicable regional agreement(s). On the HAZMAT_NOTIFIER task if HazmanNotificationInfoNonEUDepartures or HazmanNotificationInfoEUDepartures information is provided. On the SECURITY_NOTIFIER task if Security Notification information is provided. On the WASTE_NOTIFIER task if WasteNotification information is provided. On the CREWPAX_NOTIFIER task if CrewAndPaxNotificationOnArrival or CrewAndPaxNotificationOnDeparture information is provided. On the BUNKERS_NOTIFIER task if BunkersNotificationTowardsPortOfCall or BunkersNotificationTowardsNextPort information is provided.
Step 2	The PortPlus UpdateStatus = N.
Step 3	The system checks that the ShipCallId is unique for the Country that the SSN Authority providing the notification belongs to.
Step 4	The system validates the notification contents against the XMLRG business rules.
Step 5	The system sets the status of the PortPlus validation to Valid.
Step 6	The system persists the notification data in the DB as a new record. The notification data are stored as reported and then they are sent to the voyage consolidation process. The business rules defined in Annex A: Business Rules are applicable.
Step 7	The system Logs the message and processing details.
Alternative Use Case Description	Valid Update PortPlus Notification
Step 2.1	The PortPlus UpdateStatus = U.

Step 2.2	The system checks if the ShipCallId exists in the database and if yes, it is unique for the MS of the userId reporting the notification. The voyage is identified, and the status is not Canceled
Step 2.3	The system validates the notification contents against the XMLRG business rules. The system checks if the ArrivalNotificationDetails and DepartureNotificationDetails are recorded in the corresponding voyage and if yes, they are reported in the Update PortPlus Notification.
Step 2.4	The system sets the status of the PortPlus validation to Valid.
Step 2.5	The system persists the notification data in the DB as a new record and sends to the voyage consolidation process. The business rules defined in Annex A: Business Rules are applicable.
Step 2.6	The system Logs the message and processing details.
Alternative Use Case Description	Not Unique New PortPlus Notification
Step 3.1	The system identifies that for UpdateStatus = N the ShipCallId is not unique in the DB. This is done by querying the PORTPLUS_NOTIFICATIONS table by the reported ShipCallId and UpdateStatus='N'. If found, then the ShipCallId is not unique for the MS of the userId reporting the notification.
Step 3.2	The system sets the status of the PortPlus validation to Invalid.
Alternative Use Case Description	ShipCallId not in DB for Update Notification
Step 2.2.1	The system identifies that for UpdateStatus = U the ShipCallId does not exist in the DataBase.
Step 2.2.2	The system checks that the ShipCallId does not exist in the database for the MS of the userId reporting the notification.
Step 2.2.3	The system validates the notification contents against the XMLRG business rules.
Step 2.2.4	The system sets the status of the PortPlus validation to Valid.
Step 2.2.5	The system produces a warning to be appended in the SSN_Receipt message that the corresponding PortPlus with UpdateStatus=N shall be sent. The system produced an e-mail warning to the 24/7 NCA to request the NCA to send the original message as soon as possible.
Step 2.2.6	The system persists the notification data in the DB and sends to the voyage consolidation process. The business rules defined in Annex A: Business Rules are applicable.
Alternative Use Case Description	Cancel a PortPlus notification
Step 2.3.1	The system identifies that for UpdateStatus = U the ShipCallId exists in the DataBase. PortOfCall = ZZCAN.

Step 2.3.2	The system checks that the ShipCallId exists in the database.
	The system validates the notification contents against the XMLRG business rules. No ATA to PortOfcall is reported for the voyage.
Step 2.3.3	The system sets the status of the PortPlus validation to Valid.
Step 2.3.4	The system persists the notification data in the DB and sends to the voyage consolidation process. The business rules defined in Annex A: Business Rules are applicable.
Alternative Use Case Description	New PortPlus Notification - Invalid
Step 5.1	Validation of the submitted data fails against the XMLRG business rules.
Step 5.2	The system sets the status of the PortPlus validation to Invalid. The error message will be communicated in the SSN_Receipt. Go to Step 5.6.
Step 5.3	Not executed.
Input(s)	New PortPlus Notification
Output(s)	Notification Information stored in the SSN database.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	Extended UC: UC-SSN-NOT-04 - Process Notification, UC-SSN-PPN-06: Voyage Consolidation
Special Requirements	The message header attribute Version value '3' will be supported until all NCA applications are upgraded to support the XMLRG version '4'.

3.2.2.3 UC-SSN-PPN-06: Voyage Consolidation

Use Case Req ID	UC-SSN-PPN-06	
Use Case Name	Voyage Consolidation	
Purpose	<p>Covers the functionality related to the voyage consolidation processing. The primary scope of the voyage calculation is to identify all the notifications transmitted to SSN-EIS that refer to the same ship call (defined in SafeSeaNet as voyage). PortPlus notifications sent for the same ShipCallID are parts of the same voyage. However, a voyage may include notifications sent by more than one data providers and therefore different ShipCallID. A voyage of the same vessel may be reported by the departure port (as NextPort) and by the arrival port (as PortOfCall). These two cases which an associated to the same voyage shall be correlated based on the vessel, port and ETA/ATA.</p> <p>The VoyageId is defined as [ShipCallId '#' Country alpha-2 ISO code].</p>	

	The PortPlus notification may be reported by a NCA application or by STIRES-STAR. In the later case the PortPlus notification reports calculated data, and the voyage to be created is identified as "detected". The process will consolidate PortPlus notification reported from NCA applications only with "non-detected" voyage, while the PortPlus notifications from STIRES-STAR will be consolidated only with "detected" voyages. A detected voyage may be co-related with a non-detected voyage if reported for the same ship and port of call and according to the rules defined in Annex A , sections: Voyage retrieval specific rules, Voyage Status Indicators, Voyage correlation further rules
Subsystem	SSN Core
Primary Actor(s)	NCA Application/STIRES-STAR
Precondition(s)	Process PortPlus Notification UC has identified the incoming message as a valid PortPlus Notification.
Postcondition(s)	System persists Data in the SSN Database
Trigger(s)	PortPlus Notification Received
Use Case Description	Valid PortPlus Notification –reported from NCA Application
Step 1	The process checks in the data store if exists a voyage with VoyageId equal to the VoyageId of the notification.
Step 2a	<p>A voyage exists with the same VoyageId. Check if the new PortPlus notification reports HazmatNonEUDeparture or BunkersNotificationTowardsPortOfCall or CrewandPaxNotificationOnArrival:</p> <ul style="list-style-type: none"> • If yes, then update the voyage with the data from the new PortPlus. • If no, then update the voyage with the data from the new PortPlus. Then check for active Hazmat of the same ship, reported by a different data provider, with ETA in the future compared to the ETA reported in the notification, with ATA and ATD null. If found, then assign the Hazmat to the new voyage created and delete Hazmat from the previous voyage. <p>If the notification reports Security and/or Waste then if the existing voyage already has these data update with the corresponding new data; if the voyage has no such data, then insert the new data.</p> <p>If the new PortPlus reports NextPort go to step 3 else go to step 5.</p>
Step 2b	<p>No voyage exists with the same VoyageId. Check for existing voyage that match the ship and PortOfCall and has noHazmat or CrewandPax or Bunkers.</p> <ul style="list-style-type: none"> • If no existing voyage found and the new PortPlus has Hazmat or CrewandPax or Bunkers then create a new voyage for the PortOfCall with the data reported in the new PortPlus notification. Insert also the Security and/or Waste and/or HazmatNonEUDeparture and/or BunkersNotificationTowardsPortOfCall and/or CrewAndPassengersOnArrival reported in the new notification. If no existing voyage found and the new PortPlus has no Hazmat or CrewandPax or Bunkers then create a new voyage for the PortOfCall

	<p>with the data reported in the new PortPlus notification. Insert also the Security and/or Waste and/or HazmatNonEUDepartureand/or BunkersNotificationTowardsPortOfCall and/or CrewAndPassengersOnArrival reported in the new notification. Then check for active Hazmat the same ship, reported by a different data provider, with ETA in the future compared to the ETA reported in the notification, with ATA and ATD null. If found, then assign the Hazmat to the new voyage created and delete Hazmat from the previous voyage.</p> <ul style="list-style-type: none"> If an existing voyage is found, then check the data in the PortPlus with the data in the existing voyage based on the business rules defined in Annex A: Business Rules. If a match is found, then update the existing voyage found with the data from the new PortPlus. If no match then create a new voyage for the PortOfCall with the data from the new PortPlus. Insert also the Security and/or Waste and/or HazmatNonEUDeparture and/or BunkersNotificationTowardsPortOfCall and/or CrewAndPassengersOnArrivall reported in the new notification. <p>If the new PortPlus has UpdateStatus = "U", then set the status of the new voyage to "On-hold".</p> <p>If the new PortPlus reports NextPort go to step 3 else go to step 5.</p>
Step 3	The valid PortPlus notification reports NextPort and ETA to NextPort.
Step 4a	<p>The notification reports HazmatEUDeparture or BunkersNotificationTowardsNextPort or CrewAndPassengersOnDeparture. Check for existing voyage that match the ship and NextPort in the notification with the PortOfCall in the voyage, was reported by a different data provider and has no Hazmat or CrewandPax or Bunkers:</p> <ul style="list-style-type: none"> If no match, then create a new voyage for theNextPort with the data from the new PortPlus including the HazmatEUDepartureand/orBunkersNotificationTowardsNextPortand/or CrewAndPassengersOnDeparturein the new notification. If an existing voyage is found, then check the data in the PortPlus with the data in the existing voyage based on the business rules defined in Annex A: Business Rules. If a match is found, then update the existing voyage found with the data from the new PortPlus for the NextPort. Update alsothe HazmatEUDeparture and/orBunkersNotificationTowardsNextPortand/or CrewAndPassengersOnDeparture reported in the notificationOr delete the relevant data if not reported in the PortPlus notification. If no match then create a new voyage with the data from the new PortPlus for the NextPort including the the HazmatEUDepartureand/orBunkersNotificationTowardsNextPortand/or CrewAndPassengersOnDeparturereported in the new notification.
Step 4b	<p>The notification reports noHazmatEUDeparture or BunkersNotificationTowardsNextPort or CrewAndPassengersOnDeparture. Check for existing voyage that match the ship and NextPort in the notification with the PortOfCall in the voyage, was reported by a different data provider.</p>

	<ul style="list-style-type: none"> If no match then perform no action. No need to create a voyage for NextPort when no Hazmat or CrewandPax or Bunkers are reported. If a voyage is found, then update the existing voyage with data from the new PortPlus for the NextPort. If the voyage found has Hazmat or Crewand Pax or Bunkersthendelete the relevantdata. Do not update LastPort, ETDFromLastPort, ETAToPortOfCall and ETDFromPortOfCall.
Step 5	The system persists the notification data in the VOYAGES specific tables that store the consolidated ship call information reported in the related notifications. The voyage is identified as Non-Detected (reported from an NCA application).
Alternative Use Case Description	Valid detected PortPlus Notification –reported from STIRES-STAR
Step 1.1	<p>The process checks in the data store if exists a voyage based on the ShipCallId. The voyage must be a detected one meaning it has been reported from STIRES-STAR.</p> <p>Condition: The detected voyage VoyageId value equals the Database_Id of the matching voyage reported from an NCA application.</p>
Step 1.2a	<p>A detected voyage exists with the same ShipCallId. Check if the new PortPlus notification matches a voyage reported by an NCA application based on the ship and PortOfCall.</p> <ul style="list-style-type: none"> If found then check the data in the new PortPlus with the data in the NCA application voyage based on the business rules defined in Annex A: Business Rules for calculated ATA, ATD and ETD. If found set the existing detected voyage VoyageId value to the Database_Id of the matching voyage reported from an NCA application. If not match is found set the detected voyage VoyageId = Null. <p>Update the existing detected voyage with the data from the new PortPlus notification.</p>
Step 1.2b	<p>No detected voyage exists with the same ShipCallId. Check if the new PortPlus notification matches a voyage reported by an NCA application based on the ship and PortOfCall.</p> <ul style="list-style-type: none"> If found then check the data in the new PortPlus with the data in the NCA application voyage based on the business rules defined in Annex A: Business Rules for calculated ATA, ATD and ETD. If found set the new detected voyage VoyageId value to the Database_Id of the matching voyage reported from an NCA application. If not found set the new detected voyage VoyageId = Null. <p>Insert a new detected voyage with the data from the new PortPlus notification.</p>
Step 1.3	The system persists the notification data in the VOYAGES specific tables that store the consolidated ship call information reported in the related notifications. The voyage is identified as Detected (reported from STIRES-STAR).

Alternative Use Case Description	Delete HazmatNonEuPort / HazmatEUPort / Security / Waste / BunkerTowardsPortOfCall/ BunkerTowardsNextPort / CrewandPassengerOnArrival / CrewandPassengerOnDeparture
Step 2.1	<p>The valid PortPlus notification has at least one of the following parameters set to Y</p> <ul style="list-style-type: none"> – DeleteHazmatNotificationInfoNonEUDepartures – DeleteHazmatNotificationInfoEUDepartures – DeleteWasteNotification – DeleteSecurityNotification – DeleteBunkersNotificationTowardsPortOfCall – DeleteBunkersNotificationTowardsNextPort – DeleteCrewandPassngerNotificationc – DeleteCrewandPassengerNotificationOnDeparture
Step 2.2a	<p>A voyage exists with the same VoyageId.</p> <p>Update the existing voyage with the data from the new PortPlus notification.</p> <p>Delete the data related to the HazmatNonEuPort / HazmatEUPort / Security / Waste/ BunkerTowardsPortOfCall / BunkerTowardsNextPort / CrewandPassengerOnArrival/ CrewandPassengerOnDeparture if the corresponding parameter is Y.</p> <p>If the HazmatOnBoardYorN is N any previously reported data are ignored by setting the parameter in the database to the value N.</p> <p>If the BunkersReportedYorNis N any previously reported data are ignored by setting the parameter in the database to the value N.</p> <p>Go to step 2.3</p>
Step 2.2b	<p>No voyage exists with the same VoyageId.</p> <p>Do nothing. Cannot delete data from a non-existing voyage.</p> <p>Use case ends.</p>
Step 2.3	The system persists the notification data in the VOYAGES specific tables that store the consolidated ship call information reported in the related notifications.
Alternative Use Case Description	Cancel a PortPlus notification
Step 3.1	The valid PortPlus notification has UpdateStatus = "U" and PortOfCall = "ZZCAN". The process checks in the data store if exists a voyage based on the VoyageId.
Step 3.2a	<p>A voyage exists with the same VoyageId.</p> <p>Update the existing voyage with the data from the new PortPlus notification. Set the voyage status to Cancelled. The voyage will not longer be considered for a ShipCall request or consolidation of a new notification.</p> <p>Go to step 3.3.</p>
Step 3.2b	<p>No voyage exists with the same VoyageId.</p> <p>Do nothing. Cannot cancel a non-existing voyage.</p> <p>Use case ends.</p>

Step 3.3	The system persists the notification data in the VOYAGES specific tables that store the consolidated ship call information reported in the related notifications.
Input(s)	Valid PortPlus Notification
Output(s)	Notification Information are consolidated in the database with previously sent PortPlus notification data related with the same ship voyage. The consolidated ship call data are stored in the VOYAGES tables. Details specific to Hazmat, Security, Waste, Bunkers, CrewandPassengers are stored in the VOYAGES child tables specific to the data.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	Extended UC: UC-SSN-PPN-05: Process PortPlus Notification
Special Requirements	<ul style="list-style-type: none"> The business rules defined in Annex A: Business Rules are applicable. Voyages with status "Dummy" or "Closed" are not considered in the voyage consolidation.

3.2.2.4 UC-SSN-PPN-07: Update Voyage Status

Use Case Req ID	UC-SSN-PPN-07	
Use Case Name	Update Voyage Status	
Purpose	<p>The current use case describes the system functionality related to the update of the voyage status based on the reported data.</p> <p>The update voyage status will run at predefined intervals to:</p> <ul style="list-style-type: none"> set the voyage Status = "Closed" based on the reported ATD and ETA. set the voyage Status = "Dummy" based on the reported ETA and ATA to a PortOfCall. 	
Subsystem	SSN database	
Primary Actor(s)	Oracle Job	
Precondition(s)	System is operational	
Postcondition(s)	The status of voyages that satisfy the conditions defined in the Purpose are updated.	
Trigger(s)	Oracle job scheduled to run at predefined intervals.	
Use Case Description	Primary Workflow	
Step 1	<p>The system updates the voyages with</p> <ul style="list-style-type: none"> ETAToPortOfCall in the past (compared with the application parameter VOY_ETA_COND) and with no ATAPortOfCall (compared with the application parameter VOY_ATA_COND) AND If there is at least one voyage to another port with a later ATAPortOfCall compared to the voyage's ETAToPortOfCall (reported by a MS or detected in STIRES-STAR). 	

	By setting the Status = "Dummy".
Step 2	The system updates the voyages with: <ul style="list-style-type: none"> - ATDPortOfCall before the configurable archival period or - ETAToPortOfCall before the configurable archival period, if no ATAPortOfCall nor ATDPortOfCall By setting the Status = "Closed".
Input(s)	Current timestamp
Output(s)	Voyage status update.
Timer(s)	Hourly.
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	The systems consider two distinct configurable archival periods: one applied to ATDPortOfCall, and one applied to ETAToPortOfCall

3.2.2.5 UC-SSN-MRS-08: Process Ship (AIS or MRS) Notification (XML)

Use Case Req ID	UC-SSN-MRS-08
Use Case Name	Process Ship AIS or MRS Notification (XML)
Purpose	Covers the functionality related to ShipNotification processing via the system to system XML interface.
Subsystem	SSN Core
Primary Actor(s)	SSN System User
Precondition(s)	Process Notification UC has identified the incoming message as a valid Ship Notification.
Postcondition(s)	System persists Data in the SSN Database
Trigger(s)	Notification Received
Use Case Description	Valid New Ship Notification
Step 1	The system identifies access rights restrictions on ship notifications, based on the following: <ul style="list-style-type: none"> • On the SHIP_AIS_NOTIFIER task, in case of Ship notification of type AIS;On the SHIP_MRS_NOTIFIER task, in case of Ship notification of type MRS.
Step 2	The system checks that the MSRefId is unique.
Step 3	The system validates the notification contents against XMLRG business rules. Furthermore, in case of MRS, the system will verify that the actor is submitting the Notification for a MRS that is managed by its country (e.g. Poland can submit GDANREP notifications but not WETREP notifications). Reference information regarding member states entitled

	to provide MRS information will be maintained by central SSN authority. [SSNV3_2].
Step4	in case of MRS, the system checks the contents of the identification of the Coastal Station (attribute "CSTIdentification"). If the content does not match any authority responsible for a MRS in the EMSA Central Organisation Database (COD), a warning message will be returned in the StatusMessage in the SSN_Receipt.xml (no rejection). The message will state the following: "WARNING: theCoastal Station (attribute CSTIdentification) is unknown to the system". [SSNV3_4].
Step 5	The system persists the notification data in the DB as a new record.
Step 6	The system logs the message and processing details.
Step 7	The system "pushes" the notification content to STAR
Alternative Use case Description	Duplicate message; the actor has already sent a message with the same MSRefId
Ste 2.1	The system returns an SSN Receipt with StatusCode="InvalidFormat"
Alternative Use Case Description	Authorization fails
Step 3.1	The system returns an SSN_Receipt with StatusCode="AccessDenied"
Alternative Use Case Description	Validation against XMLRG business rules fails
Step 4.1	The system returns an SSN_Receipt with StatusCode="InvalidFormat"
Input(s)	New AIS or MRS Notification
Output(s)	Notification Information stored in the SSN database.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	Extended UC: UC-SSN-NOT-04 - Process Notification
Special Requirements	

3.2.2.6 UC-SSN-MRS-09: Process Ship MRS Notification (Web)

Use Case Req ID	UC-SSN-MRS-09
Use Case Name	Process Ship MRS Notification (Web)
Purpose	<p>Covers the functionality related to MRS Notification processing via the Web interface.</p> <p>This communication channel is an alternative to the XML based interface with central SSN application.</p> <p>In that sense, the information they should provide from the alternative browser-based web channel should conceptually include the union of information defined in ship notification (MS2SSN_Ship_Not) message and ship response (SSN2MS_Ship_Res) message. Hereafter, SSN central application will be able to respond to ship notification requests</p>

	without further consultation of the data provider.
Subsystem	SSN Core, SSN Send Notifications console
Primary Actor(s)	SSN Human User
Precondition(s)	The Actor has been authenticated in the system and is authorised to create MRS Notifications. The actor is granted the permission SHIP_MRS_NOTIFIER.
Postcondition(s)	System persists Data in the SSN Database
Trigger(s)	The Actor has selected to provide MRS Notification for a vessel.
Use Case Description	Create a new, valid MRS Notification
Step 1	The Actor navigates to MRS notification screen (Generic Notifications >MRS Notification) and searches for the vessel to send the MRS Notification for.
Step 2	The system returns a list of vessels corresponding to the search criteria specified by the actor. The Actor selects the desired vessel and the system redirects to the screen where it prompts for MRS Notification details.
Step 3	<p>The Actor fills in the information required to form a full MRS Notification and submits it. As explained above, this information merely includes non-duplicated information contained in both MS2SSN_Ship_Not and SSN2MS_Ship_Res messages.</p> <p>To populate contact information (LastName, FirstName, Phone, Fax, Email, Locode) for the new MRS notification, the contact details of the IdM Human User's Organisation will be used .</p> <p>Since a Notification for an MRS can be submitted only from authorized users, the relevant list of MRS displayed in the input forms contains only the authorized MRS [SSNV3_2].</p>
Step 4	The system successfully validates the notification contents against XMLRG business rules.
Step 5	The system checks the contents of the identification of the Coastal Station (attribute "CSTIdentification"). If the content does not much any authority responsible for a MRS in the EMSA Central Organisation Database (COD), a warning message will be displayed in the success page returned to the actor. The message will state the following: "WARNING: The Coastal Station (attribute CSTIdentification) is unknown to the system". [SSNV3_4].
Step 6	The system persists the notification data in the DB as a new record. The system will reuse the existing MrsNotification entity along with a new, optional, dependant entity MrsNotificationDetails for that purpose.
Step 7	The system logs the message and processing details.
Step 8	The system "pushes" the notification content to STAR
Alternative Use Case Description	The information provided for MRS Notification is not valid.

Step 2.1	The system returns to the input screen where all rule violations are presented and prompts the actor to correct them and re-submit the MrsNotification information.
Step 1.2	The flow continues from Step2.
Input(s)	New MRS Notification
Output(s)	Notification Information stored in the SSN database.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	Extended UC: UC-SSN-NOT-04 - Process Notification
Special Requirements	

3.2.2.7 UC-SSN-EXC-10: Process Exemption Notification

Use Case Req ID	UC-SSN-EXC-10	
Use Case Name	Process Exemption Notification	
Purpose	Covers the functionality related to Exemption Notification processing.	
Subsystem	SSN Core	
Primary Actor(s)	SSN System User	
Precondition(s)	Process Notification UChas identified the incoming message as a valid Exemption Notification.	
Postcondition(s)	System persists Data in the SSN Database	
Trigger(s)	A notification XML message is received.	
Use Case Description	Valid New Exemption Notification	
Step 1	SSN is invoked to receive the incoming exemption notification or from an external application. UpdateStatus = N.	
Step 2	The system identifies access rights restrictions on Exemption notifications, based on the EXEMPTIONS_NOTIFIER task.	
Step 3	<p>The system validates the reported data based on the XMLRG business rules.</p> <p>The system applies the location restrictions on the reported granting country (i.e. Exemption's authority's country):</p> <ul style="list-style-type: none"> • In case the EXEMPTIONS_NOTIFIER permission has a "Specific locations" or "User's country" location restriction, granting country shall be the actor's country. • In case the EXEMPTIONS_NOTIFIER permission has a "Countries list" location restriction, granting country shall be within the countries assigned to the applicable regional agreement(s). 	

	<p>The system applies the location restrictions on the reported exempted ports:</p> <ul style="list-style-type: none"> • In case the EXEMPTIONS_NOTIFIER permission has "Specific locations" location restriction, exempted ports shall be within the locodes assigned to the applicable duty/ies for the actor's organisation. • . • In case the EXEMPTIONS_NOTIFIER permission has "User's country" location restriction, exempted ports shall belong to actor's country; • In case the EXEMPTIONS_NOTIFIER permission has "Countries list" location restriction, exempted ports shall belong to the countries assigned to the applicable regional agreement(s).
Step 4	The system persists the notification data in the DB as a new record. The notification data are stored as reported.
Step 5	The system Logs the message and processing details.
Alternative Use Case Description	Invalid New Exemption Notification
Step 1.6	Validation of the submitted data fails against the XMLRG business rules. The error message will be communicated in the SSN_Receipt.
Alternative Use Case Description	Update Exemption Notification
Step 2.1	SSN is invoked to receive the incoming exemption notification or from an external application. UpdateStatus = U
Step 2.2	<p>The system identifies the access rights restrictions on Exemption notifications, based on the EXEMPTIONS_NOTIFIER task and checks if actor is authorized to update an existing exemption based on the applicable location restrictions. More specifically:</p> <ul style="list-style-type: none"> • In case of "User's country" location restriction, the system checks if the actor's country is equal to the existing exemption's granting country. • In case of "Countries list" location restriction, the system checks if the existing exemption's granting country is within the countries assigned to the applicable regional agreement(s). • In case of "Specific locations" location restriction, the system checks if the existing exemption's exempted ports are within the locodes assigned to the applicable duty/ies for the actor's organisation.
Step 2.3	<p>The system validates the reported data based on the XMLRG business rules.</p> <p>The system controls the reported granting country:</p> <ul style="list-style-type: none"> • The granting country shall have the same value as the granting country defined in the existing exemption identified by ExemptionID. <p>The system applies the location restrictions on the reported exempted ports:</p>

	<ul style="list-style-type: none"> In case the permission has "Specific locations" location restriction, the exempted ports shall be within the locodes assigned to the applicable duty/ies for the actor's organisation. In any other case, the exempted ports shall belong to the granting country.
Step 2.4	The system identifies the exemption in the DB by ExemptionID.
Step 2.5	The system persists the notification data in the DB by updating the existing exemption with the updated data: all data of the existing exemption is replaced with the data from the notification (new data from the notification is added, existing data is replaced with the data from the notification, existing data which is not provided in the notification is deleted).
Step 2.6	The system Logs the message and processing details.
Alternative Use Case Description	Delete Exemption Notification
Step 3.1	SSN is invoked to receive the incoming exemption notification or from an external application. UpdateStatus = D
Step 3.2	<p>The system identifies access rights restrictions on exemption notifications, based on the EXEMPTIONS_NOTIFIER task and checks if actor is authorized to delete an existing exemption based on the applicable location restrictions. More specifically:</p> <ul style="list-style-type: none"> In case of "Specific locations" or "User's country" location restriction, the system checks if actor's country is equal to the existing exemption's granting country. In case of "Countries list" location restriction, the system checks if the existing exemption's granting country is within the countries assigned to the applicable regional agreement(s). In case of "Specific locations" location restriction, the system checks if the existing exemption's exempted ports are within the locodes assigned to the applicable duty/ies for the actor's organisation.
Step 3.3	The system identifies the exemption in the DB by ExemptionID.
Step 3.4	The system deletes the existing exemption identified by ExemptionID.
Step 3.5	The system Logs the message and processing details.
Input(s)	Exemption information received as XML message for a vessel.
Output(s)	New exemption is submitted to the system and associated to a vessel.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	Extended UC: UC-SSN-NOT-04 - Process Notification
Special Requirements	-

3.2.2.8 UC-SSN-EXC-11: Create Exemption

Use Case Req ID	UC-SSN-EXC-11	
Use Case Name	Create Exemption	
Purpose	The current Use Case describes the SSN Users' interaction with the system upon creating a new exemption related to ship through the web.	
Subsystem	SSN Web Application	
Primary Actor(s)	SSN Human User	
Precondition(s)	The Actor has been authenticated in the system and is authorised to create exemptions. The actor is granted the permission EXEMPTIONS_NOTIFIER.	
Postcondition(s)	A new exemption is submitted in the system.	
Trigger(s)	The Actor has selected to create an exemption for a vessel.	
Use Case Description	Create an Exemption – Valid	
Step 1	The Actor searches for a valid vessel to which the new exemption shall be associated. Search by IMO, MMSI, Call Sign and/or Ship Name is allowed.	
Step 2	The system displays a list with the vessels that match the given criteria.	
Step 3	The Actor selects a vessel from the list.	
Step 4	<p>The system displays a form with fields for the new exemption.</p> <p>The provided fields are:</p> <p>Vessel Identification</p> <ul style="list-style-type: none"> IMO Number MMSI Number Call Sign Ship Name Flag <p>The fields that constitute the vessel information are filled with the data of the vessel that has been selected. Values can be updated.</p> <p>Exemption Details</p> <p>Exemption Type (Provided types are: "Pre-arrival", "Hazmat", "Waste", "Exemption for waste notification", "Exemption for waste delivery", "Exemption for waste fees" and "Security")</p> <p>Company Name</p> <p>Exemption Validity (From, To)</p> <p>Scheduled/Route, with list of:</p> <ul style="list-style-type: none"> - Port <p>Exemption applies to, with list of:</p> <ul style="list-style-type: none"> - Port and - List of Port Facilities 	

	<p>Exempted waste types, with list of Waste Code and Description (include option to "All waste types")</p> <p>Exemption Granted by - Issuing Country (based on the geographical restrictions applied to the user permission)</p> <ul style="list-style-type: none"> - Authority Type - Authority Name; prefilled with user's organisation's name. <p>Contact 24/7</p> <ul style="list-style-type: none"> - First Name - Last Name - Location code - Email - Phone - Fax <p>To populate contact 24/7 (Last Name, First Name, Phone, Fax, Email, Location Code) for new the exemption notification, the contact details of the IdM Human User's Organisation will be used.</p> <p>In addition, the system generates a new Exemption ID (not visible on the form).</p>
Step 5	The Actor keys-in exemption information to the provided fields and submits.
Step 6	<p>The system applies the location restrictions on the reported granting country:</p> <ul style="list-style-type: none"> • In case the EXEMPTIONS_NOTIFIER permission has a "Specific locations" or "User's country" location restriction, the granting country shall be the actor's country. • In case the EXEMPTIONS_NOTIFIER permission has a "Countries list" location restriction, the granting country shall be within the countries assigned to the applicable regional agreement(s). <p>The system applies the location restrictions on the reported exempted ports:</p> <ul style="list-style-type: none"> • In case the EXEMPTIONS_NOTIFIER permission has "Specific locations" location restriction, the exempted ports shall be within the locodes assigned to the applicable duty/ies for the actor/s organisation. In case the EXEMPTIONS_NOTIFIER permission has "User's country" location restriction, exempted ports shall belong to actor's country; • .In case the EXEMPTIONS_NOTIFIER permission has "Countries list" location restriction, exempted ports shall belong to the countries assigned to the applicable regional agreement(s).
Step 7	The system validates the submitted information.
Step 8	The system persists the notification data in the DB. The notification data are stored as reported.

Step 9	The system Logs the message and processing details and informs the user about the result.
Alternative Use Case Description	Create an Exemption – Invalid
Step 8.1	Validation of the submitted information fails. The system displays the failure reason(s) to the Actor.
Step 8.2	Return to Primary Workflow – Step 2.
Alternative Use Case Description	Create an Exemption for a non-existing vessel
Step 2.1	No existing vessel matches the criteria.
Step 2.2	The Actor selects to create a new vessel (with status temporary)
Step 2.3	The system displays a form with fields for the new exemption (Vessel Identification and Exemption Details)
Step 2.4	The Actor keys-in exemption information to the provided fields and submits.
Step 2.5	The system checks if the data provided for the vessel and as long as they are technically correct, a new temporary vessel will be created and associated with the new exemption.
Alternative Use Case Description	Create an Exemption with invalid vessel input
Step 1.1	The Actor specifies invalid vessel.
Step 1.2	The system checks if the data provided for the vessel and prompts the user to enter valid vessel data.
Input(s)	Exemption information for a vessel.
Output(s)	New exemption is submitted to the system and associated to a vessel.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	Validation rules are based on the Exemption Notification definition in [R4] A new Exemption ID is generated based on a dedicated database sequence that generates unique numbers. The Exemption_ID is the concatenation of the NSW Country ISO alpha-2 code and the sequence number.

3.2.2.9 UC-SSN-EXE-12: Edit/Delete Exemption

Use Case Req ID	UC-SSN-EXE-12	
Use Case Name	Edit/Delete Exemption	

Purpose	The current Use Case describes the SSN Users' interaction with the system upon editing-deleting an exemption related to ship through the web.
Subsystem	SSN Web Application
Primary Actor(s)	SSN Human User
Precondition(s)	The Actor has been authenticated in the system and is authorised to modify exemptions. The actor is granted the permission EXEMPTIONS_NOTIFIER. The exemption is defined for a given vessel.
Postcondition(s)	An exemption is deleted, or its data are modified.
Trigger(s)	The Actor has selected to modify an exemption related to a vessel.
Use Case Description	Primary Workflow
Step 1	The Actor searches for an exception by entering Vessel and type of Exemption as criteria.
Step 2	The system displays a list with exemptions that match the criteria and the user is allowed to edit.
Step 3	The Actor selects an exemption from the list.
Step 4	The system displays a form with fields filled in with the exemption data.
Step 5	The user modifies the exemption data (listed in UC-SSN-EXC-11 Step 4) and submits.
Step 6	The system applies the location restrictions on the modified/reported exemption data: <ul style="list-style-type: none"> The modified/reported granting country shall have the same value as the granting country defined in the selected exemption. In case the permission has a "Specific locations" location restriction, the modified/reported exempted ports shall be within the locodes assigned to the applicable duty/ies for the actor/s organisation. In any other case, modified/reported exempted ports shall belong to the granting country.
Step 7	The system validates the submitted data.
Step 8	The system updates the notification data in the DB and displays the pertinent message to the user.
Alternative Use Case Description	Edit an Exemption - Invalid
Step 8.1	Validation of the submitted information fails. The system displays the failure reason(s) to the Actor.
Step 8.2	Return to Primary Workflow – Step 2.
Alternative Use Case Description	Delete an Exemption
Step 5.1	The actor selects to delete the selected exemption.
Step 5.2	The system deletes the exemption data.

Input(s)	Exemption information for a vessel.
Output(s)	An exemption's data are modified or deleted.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	<p>The actor is authorized to edit/delete an existing exemption based on the location restrictions applied to EXEMPTIONS_NOTIFIER task. More specifically:</p> <ul style="list-style-type: none"> In case of "User's country" location restriction, the system checks if the actor's country is equal to the existing exemption's granting country. In case of "Countries list" location restriction, the system checks if the existing exemption's granting country is within the countries assigned to the applicable regional agreement(s). In case of "Specific locations" location restriction, the system checks if the existing exemption's exempted ports are within the locodes assigned to the applicable duty/ies for the actor's organisation.

3.2.2.10 UC-SSN-EXR-13: Create Exemption from Existing

Use Case Req ID	UC-SSN-EXR-13	
Use Case Name	Create Exemption from Existing	
Purpose	The current Use Case describes the SSN Users' interaction with the system upon creating a new exemption related to ship through the web by using the data of an exemption that already exists.	
Subsystem	SSN Web Application	
Primary Actor(s)	SSN User	
Precondition(s)	The Actor has been authenticated in the system and is authorised to create exemptions. The actor is granted the permission EXEMPTIONS_NOTIFIER. Some exemptions already exist in the system.	
Postcondition(s)	A new exemption is submitted in the system.	
Trigger(s)	The Actor has selected to create an exemption for a vessel using the data of an existing exemption.	
Use Case Description	Primary Workflow	
Step 1	The Actor navigates to the page where he can create a new exemption from an existing one.	
Step 2	The Actor searches for an exemption by entering Vessel and type of Exemption as criteria.	
Step 3	The system displays a form with fields for the new exemption populated with the values of the exemption that has been selected from the Actor.	

Step 4	The Actor modifies the details listed in UC-SSN-EXC-11 Step 4.
Step 5	<p>The system applies the location restrictions on the modified/reported exemption data:</p> <ul style="list-style-type: none"> The modified/reported granting country shall have the same value as the granting country defined in the selected exemption. In case the permission has a "Specific locations" location restriction, the modified/reported exempted ports shall be within the locodes assigned to the applicable duty/ies for the actor/s organisation. In any other case, modified/reported exempted ports shall belong to the granting country.
Step 6	The system validates the submitted information.
Step 7	The system persists the notification data in the DB. The notification data are stored as reported.
Step 8	The system Logs the message and processing details and informs the user about the result.
Alternative Use Case Description	Create a new Exemption from an existing one - Invalid
Step 7.1	Validation of the submitted information fails. The system displays the failure reason(s) to the Actor.
Output(s)	New exemption is submitted to the system and associated to a vessel.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	<p>The actor is authorized to reuse an existing exemption based on the geographical restrictions applied to EXEMPTIONS_NOTIFIER task. More specifically:</p> <ul style="list-style-type: none"> In case of "User's country" location restriction, the system checks if actor's country is equal to the existing exemption's granting country. In case of "Countries list" location restriction, the system checks if the existing exemption's granting country is within the countries assigned to the applicable regional agreement(s). In case of "Specific locations" location restriction, the system checks if the existing exemption's exempted ports are within the locodes assigned to the applicable duty/ies for the actor's organisation .

3.2.2.11 UC-SSN-IRN-16: Process Incident Report Notification (XML)

Use Case Req ID	UC-SSN-IRN-16
Use Case Name	Process Incident Report Notification (XML)
Purpose	Covers the functionality related to Incident Report Notification processing via the system to system XML interface.
Subsystem	SSN Core / SSN XML Protocol / SSN SEG WS

Primary Actor(s)	SSN System User
Precondition(s)	Process Notification UChas identified the incoming message as a valid Incident Report Notification.
Postcondition(s)	System persists Data in the SSN Database
Trigger(s)	Notification Received
Use Case Description	Valid New Incident Report Notification
Step 1	<p>The system identifies the access rights on Incident Report notifications, based on the following:</p> <ul style="list-style-type: none"> • On the ALERT_SITREP_NOTIFIER task if incident type="SITREP"; • On the ALERT_POLREP_NOTIFIER task if incident type="POLREP"; • On the ALERT_WASTE_NOTIFIER task if incident type="Waste"; • On the ALERT_LFC_NOTIFIER task if incident type="LostFoundContainers"; • On the ALERT_OTHERS_NOTIFIER task if incident type="Others"; • On the ALERT_FAILED_NOTIFIER task if incident type="FailedNotification"; • On the ALERT_BANNED_NOTIFIER task if incident type="BannedShip"; • On the ALERT_VTS_NOTIFIER task if incident type="VTSRulesInfringement"; • On the ALERT_PILOT_NOTIFIER task if incident type="PilotOrPortReport"; • On the ALERT_INSURANCE_NOTIFIER task if incident type="InsuranceFailure";
Step 2	The system checks that the MSRefId is unique.
Step 3	The system validates the notification contents against XMLRG business rules.
Step4	<p>The SSN-Core application receives the Incident Report message and performs the following tasks, depending on the type of message (UpdateStatusReason=N(New)/U(Update)/D>Delete) or a Feedback message):</p> <ul style="list-style-type: none"> – In case of a new Incident Report message (UpdateStatusReason=N): <ul style="list-style-type: none"> ○ A new incident report notification is created in the database. – In case of an Incident Report Update message (UpdateStatusReason=U) <ul style="list-style-type: none"> ○ The notification update data are stored in the database. – In case of an Incident Report Delete message (UpdateStatusReason=D)

	<ul style="list-style-type: none"> ○ The data related to the notification reported as deleted are maintained in the database, although the notification is marked as deleted. – When an Incident Report Delete is sent, no further updates can be sent for that incident and it's incident Id cannot be reused. In case of an Incident Report Feedback message <ul style="list-style-type: none"> ○ The notification feedback data are stored in the database.
Step 5	The system persists the notification data in the DB as a new record.
Step 6	The system logs the message and processing details.
Step 7	The Incident Report message (New/Update/Delete/Feedback) is distributed to the list of recipients via HTTP (SSN2MS_IncidentDetail_Tx) or Email.
Step 8	<p>If the SSN_Receipt message is not received, or its status is not OK, a warning email is sent to that recipient (system user's email address). The generic email template is the following:</p> <p><i>Please take a note that an incident report was provided (with Incident ID: \${incidentId}) identifying [\${Incident recipient country}] as incident report recipient.</i></p> <p><i>There was a failure in the delivery of this report to your national system. The report is stored in SSN central system and could be consulted e.g. using the SSN central's system textual interface.</i></p>
Step 9	The acknowledgments received by each of the recipients are consolidated and sent as a new message (SSN2MS_IncidentDetail_Tx_Ack) back to the initial IR Provider.
Alternative Use case Description	Duplicate message; the actor has already sent a message with the same MSRefId
Step 2.1	The system returns an SSN Receipt with StatusCode="InvalidFormat"
Alternative Use Case Description	Validation against XMLRG business rules fails
Step 4.1	The system returns an SSN_Receipt with StatusCode="InvalidFormat"
Input(s)	New Incident Report Notification
Output(s)	<p>Notification Information stored in the SSN database.</p> <p>The Incident Report message (New/Update/Delete) is distributed to the list of recipients via HTTP (SSN2MS_IncidentDetail_Tx) or Email.</p> <p>The acknowledgments received by each of the recipients are consolidated and sent as a new message (SSN2MS_IncidentDetail_Tx_Ack) back to the initial IR Provider.</p>
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	Extended UC: UC-SSN-NOT-04 - Process Notification
Special Requirements	

3.2.2.12 UC-SSN-IRN-17: Process Incident Report Notification (Web)

Use Case Req ID	UC-SSN-IRN-17
Use Case Name	Process Incident Report Notification (Web)
Purpose	<p>Covers the functionality related to Incident Report Notification processing via the Web interface.</p> <p>This communication channel is an alternative to the XML based interface with central SSN application.</p> <p>In that sense, the information they should provide from the alternative browser-based web channel should conceptually include the union of information defined in Incident Report notification (MS2SSN_IncidentDetail_Not) message and Incident Report response (SSN2MS_IncidentReport_Res) message.</p>
Subsystem	SSN Core, SSN Send Notifications console
Primary Actor(s)	SSN Human User
Precondition(s)	<p>The Actor has been authenticated in the system and is authorised to create Incident Report Notifications for a specific Incident Type.</p> <p>The system identifies the access rights on Incident Report notifications, based on the following:</p> <ul style="list-style-type: none"> • On the ALERT_SITREP_NOTIFIER task if incident type="SITREP"; • On the ALERT_POLREP_NOTIFIER task if incident type="POLREP"; • On the ALERT_WASTE_NOTIFIER task if incident type="Waste"; • On the ALERT_LFC_NOTIFIER task if incident type="LostFoundContainers"; • On the ALERT_OTHERS_NOTIFIER task if incident type="Others"; • On the ALERT_FAILED_NOTIFIER task if incident type="FailedNotification"; • On the ALERT_BANNED_NOTIFIER task if incident type="BannedShip"; • On the ALERT_VTS_NOTIFIER task if incident type="VTSRulesInfringement"; • On the ALERT_PILOT_NOTIFIER task if incident type="PilotOrPortReport"; • On the ALERT_INSURANCE_NOTIFIER task if incident type="InsuranceFailure".
Postcondition(s)	System persists Data in the SSN Database
Trigger(s)	The Actor has selected to provide Incident Report Notification.
Use Case Description	Create a new, valid Incident Report Notification
Step 1	The Actor navigates to Incident Report notification screen (Incident Report>Create).
Step 2	The system generates an Incident Id and fills the Authority Reporting Incident in the Incident Report notification page where it prompts for Notification details.

	To populate "Authority Reporting Incident" information for new Incident Report notification, the contact details (Authority Name, Locode, Phone, Fax, Email) of the IdM Human User's Organisation will be used.
Step 3	The Actor selects one of the available/permitted Incident Type; the screen presents the corresponding Incident Details form.
Alternative Use Case Description	Update/Delete/Feedback
Step 1.2	The Actor navigates to Incident Report Search screen (Incident Report>Update/Delete/Feedback) and provides the search criteria; either Vessel Identification, and/or Incident Id and/or Incident Type.
Step 1.3	The system returns a list of Incident Reports corresponding to the search criteria specified by the actor. The Actor selects the desired action (Edit/Feedback) and the system redirects to the screen where it prompts for Incident Report Notification details.
Step 4	The Actor fills in the information required to form a full Incident Report Notification <ul style="list-style-type: none"> ➤ Incident Report Vessel List ➤ Associated Incidents ➤ Incident Details ➤ Distribution (Recipients) and submits it.
Step 5	The system successfully validates the notification contents against XMLRG business rules.
Step 6	The system persists the notification data in the DB as a new record. The system distributes the Incident Report message to the list of recipients via HTTP (SSN2MS_IncidentDetail_Tx) or Email. On Notification processing result page, the system gives the option to the Actor to Check Acknowledgement Status; the acknowledgments received by each of the recipients.
Step 7	The system logs the message and processing details.
Input(s)	New Incident Report Notification
Output(s)	Notification Information stored in the SSN database. The Incident Report message (New/Update/Delete/Feedback) is distributed to the list of recipients via HTTP (SSN2MS_IncidentDetail_Tx) or Email. The acknowledgments received by each of the recipients are consolidated and prompted to the actor.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	Extended UC: UC-SSN-NOT-04 - Process Notification
Special Requirements	

3.2.3 Data Request

This system package includes the services required in order for the SSN Central to process requests for details and response via the XML/SOAP and Web interfaces. More specifically, this system package consists of the following use-cases:

3.2.3.1 UC-SSN-SCREQ-20: Process ShipCall Request

Use Case Req ID	UC-SSN-SCREQ-20	
Use Case Name	Process ShipCall Request	
Purpose	Covers the functionality related to the systems' actions upon receiving a request for information from either the SSN GI Interface or a National Application.	
Subsystem	SSN Core / SSN XML Protocol / SSN SEG WS	
Primary Actor(s)	SSN System User	
Precondition(s)	A v3 or v4 request for information has been received by the system.	
Postcondition(s)	The system has processed the request.	
Trigger(s)	Request message received	
Use Case Description	Primary Workflow	
Step 1	A System User's application has sent a valid request for information to the System.	
Step 2	<p>The system identifies the access rights on ShipCall requests, based on the following:</p> <ul style="list-style-type: none"> On the SHIPCALL_REQUESTOR task, location restrictions on the PortOfCall and source restrictions on the data provider. On the HAZMAT_REQUESTOR task if HazmatDetails are requested. On the SECURITY_REQUESTOR task if SecurityDetails are requested. On the WASTE_REQUESTOR task if WasteDetails are requested. On the BUNKERS_REQUESTOR task if BunkersDetails are requested. On the CREWPAX_REQUESTOR task if CrewAndPaxDetails are requested. On the EXEMPTIONS_REQUESTOR task if Exemption information exists. 	
Step 3	<p>The System identifies one of the following requests (queries):</p> <p><i>ExpectedCallOfSelectedShip:</i></p> <p>Mandatory parameters: IMONumber or MMSINumber; Optional parameters: StartDateTime, GetHazmat, GetWaste, GetSecurity, GetBunkers.</p> <p>Business rules applicable [R4] Table 4– Description of queries supported by the ShipCall_Req message.</p> <p>Result: Details of a Voyage</p>	

- Without ATAPortOfCall, and
- With ETAToPortOfCall after and closest to StartDateTime.
- Hazmat/Security/Waste/Bunkers information in results is as reported before arrival to the port.

MostRecentArrivalOfSelectedShip:

Mandatory parameters: IMONumber or MMSINumber;

Optional parameters: StartDateTime, GetHazmat, GetWaste, GetSecurity, GetBunkers.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCall_Req message.

Result: A Voyage

- With ATAPortOfCall before and closest to StartDateTime, and
- With ATDPortOfCall, if available, after StartDateTime.
- Hazmat/Security/Waste/Bunkers information in results is as reported before arrival to the port.

MostRecentDepartureOfSelectedShip:

Mandatory parameters: IMONumber or MMSINumber;

Optional parameters: StartDateTime, GetHazmat, GetWaste, GetSecurity, GetBunkers.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: a Voyage

- Central SSN will provide the ship call with ATDPortOfCall before and closest to StartDateTime.
- Hazmat/Security/Waste/Bunkers information in results is as reported before departure from the port.

RecentAndCurrentShipCallsOfSelectedShip:

Mandatory parameters: IMONumber or MMSINumber;

Optional parameters: StartDateTime, EndDateTime, NumberOfCalls.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: List of Voyages

- With ATAPortOfCall within the time period defined by StartDateTime and EndDateTime.
- If no time period is defined, Central SSN will provide the list of latest [NumberOfCalls] consolidated PortPlus messages with ATAPortOfCall before SentAt.
- Only summary data are provided. No request for additional details from the data provider.

ExpectedShipCallsAtEUPort:

Mandatory parameters: PortOfCall;

Optional parameters: StartDateTime, EndDateTime, NumberOfCalls.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: List of Voyages

- With ETAToPortOfCall within the time period defined by StartDateTime and EndDateTime, and
- Without ATAPortOfCall.
- If no time period is defined, Central SSN will provide the list of [NumberOfCalls] correlated voyages:
- With ETAToPortOfCall after SentAt, and
- Without ATAPortOfCall.
- Only summary data are provided. No request for additional details from the data provider.

CurrentShipCallsAtEUPort: Mandatory parameters: PortOfCall;

Optional parameters: StartDateTime.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: List of Voyages

- With PortOfCall = defined PortOfCall, and
- With ATAPortOfCall after StartDateTime, and
- Without ATDPortOfCall.
- Only summary data are provided. No request for additional details from the data provider.

CompletedShipCallsAtEUPort:

Mandatory parameters: PortOfCall;

Optional parameters: StartDateTime.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: List of Voyages

- With PortOfCall = defined PortOfCall, and
- With ATDPortOfCall after StartDateTime.
- Only summary data are provided. No request for additional details from the data provider.

LatestCallUpdates:

Mandatory parameters: StartDateTime, EndDateTime.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: List of Voyages which were registered or updated within the specified time period.

- Only summary data are provided. No request for additional details from the data provider.

ListExpectedCallsOfSelectedShip:

Mandatory parameters: IMONumber or MMSINumber;

Optional parameters: StartDateTime.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: List of Voyages

- Without ArrivalDetails element and
- With ETAToPortOfCall after StartDateTime, and
- Without ATAPortOfCall.
- Only summary data are provided. No request for additional details from the data provider.

SelectedShipCall:

Mandatory parameters: ShipCallID;

Optional parameters: GetHazmat, GetWaste, GetSecurity, GetBunkers.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: a Voyage

- With the specified ShipCallID.
- If the voyage does not have an ATD, the system will provide the Hazmat/Bunkers information in results as reported before arrival to the port.
- If the voyage has an ATD, the system will provide the Hazmat/Bunkers information of the voyage towards next port.
- Security/Waste information in results is as reported before arrival to the port.

GetActiveHazmatForSelectedShip:

Mandatory parameters: IMONumber or MMSINumber, GetHazmat;

Optional parameters: StartDateTime.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: the most relevant Voyage and associated hazmat details which are active at StartDateTime.

- Information may come from different PortPlus messages (different values of ShipCallID).
- Hazmat information in results. Details in case requested and are provided by the data provider. Summary in case requested or details are not provided by the data provider.

GetActiveSecurityForSelectedShip:

Mandatory parameters: IMONumber or MMSINumber, GetSecurity;

Optional parameters: StartDateTime.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: the most relevant Voyage and associated security details.

- Without ATDPortOfCall and
- With closest ETAToPortOfCall or ATAPortOfCall to StartDateTime
- Security information in results. Details in case requested and are provided by the data provider. Summary in case requested or details are not provided by the data provider.

GetActiveWasteForSelectedShip:

Mandatory parameters: IMONumber or MMSINumber, GetWaste;

Optional parameters: StartDateTime.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: the most relevant Voyage and associated waste details.

- Without ATDPortOfCall and
- With closest ETAToPortOfCall or ATAPortOfCall to StartDateTime
- Waste information in results. Details in case requested and are provided by the data provider. Summary in case requested or details are not provided by the data provider.

GetActiveBunkersForSelectedShip:

Mandatory parameters: IMONumber or MMSINumber, GetBunkers;

Optional parameters: StartDateTime.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCall Req message.

Result: the most relevant Voyage and associated bunkers details which are active at StartDateTime.

- Information may come from different PortPlus messages (different values of ShipCallID).
- Bunkers information in results. Details in case requested and are provided by the data provider.

GetActiveCrewAndPaxForSelectedShip:

Mandatory parameters: IMONumber or MMSINumber, GetCrewAndPax;

Optional parameters: StartDateTime.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: the most relevant Voyage and associated crew and passengers details which are active at StartDateTime.

- Information may come from different PortPlus messages (different values of ShipCallID).

	<p>Crew and passengers information in results. Details in case requested and are provided by the data provider. Summary in case requested or details are not provided by the data provider.</p> <p>For additional requests (queries) initiated only by SEG and not NCAs, to SafeSeaNet EIS application, please refer to [R11].</p>
Step 4	<p>The system verifies that the search criteria are correct based on the XMLRG business rules for ship call request.</p> <p>The system searches the SSN database for a relevant voyage(s) based on the provided request criteria. The business rules for voyage retrieval defined in Annex A: Business Rules are also applicable.</p> <p>In case of a request for Hazmat/Waste/Security/Bunkers summary/details go to step 5a.</p> <p>Otherwise go to step 5b.</p>
Step 5a	<p>The system has identified a voyage which complies with the search criteria for Hazmat/Waste/Security/Bunkers summary/details and the requestor's access rights and location restrictions (applied to the "PortOfCall" of the voyage) and source restrictions (applied to the provider of the voyage data). If voyage is within the location and source restrictions, then the voyage is provided.</p> <p>Location restrictions are applied to the PortOfCall of the voyage as follows:</p> <ul style="list-style-type: none"> ○ In case the SHIPCALL_REQUESTOR permission has a "Specific locations" location restriction, the PortOfCall shall be within the locodes assigned to the applicable duty/ies of the requestor's organisation. ○ In case the SHIPCALL_REQUESTOR permission has a "User's country" location restriction, the PortOfCall shall belong to the requestor's country; ○ In case the SHIPCALL_REQUESTOR permission has a "Countries list" location restriction, the PortOfCall shall belong to the countries assigned to the applicable regional agreement(s). <p>Source restrictions are applied to the provider of the voyage data as follows:</p> <ul style="list-style-type: none"> ○ In case the SHIPCALL_REQUESTOR permission has a "Countries list" source restriction, the notifier's country (country of the provider of the voyage data) shall be within the countries assigned to the applicable regional agreement(s). ○ In case the SHIPCALL_REQUESTOR permission has a "User's country" source restriction, the notifier's country (country of the provider of the voyage data) shall be equal to the requestor's country. <p>The system fetches the voyage data from the data store. Go to step 6.</p>
Step 5b	<p>The system has identified a list of voyage(s) which complies with the search criteria and the requestor's access rights, location restrictions (applied to the "PortOfCall" of each voyage) and source restrictions (applied to the provider of each voyage). Location and source restrictions are applied to each voyage as in Step 5a. Only the voyages which comply with the location and source restrictions are provided. The system fetches the voyage data from the data store. Go to step 9.</p>

Step 6	<p>In case of a request for Hazmat/Waste/Security/Bunkers details, the system identifies that the requested detailed information in XML must be acquired from an NCA Application (data provider). The system issues a ShipCall Request to the NCA application for the details.</p> <p>In case the details are waste, and waste details had been provided already in a PortPlus notification, EIS will provide the information stored at central level without sending a request to the NCA application (data provider) for the details.</p> <p>Go to step 7.</p>
Step 7	<p>The system receives within the timeout interval the response from the NCA application. It parses the QueryResults part of the response message.</p> <p>In case of a v2 MS2SSN_ShipCall_Res, the system generates a receipt indicating the error and StatusCode="InvalidFormat" and a StatusMessage indicating that version value "2.0" is incorrect.</p> <p>Go to step 8.</p>
Step 8	<p>If the requestor has the access rights to requesting exemption data, the system searches for exemptions in the SSN database which are relevant. However, no geographical restrictions will be applied on exemption results included in a ship call response.</p> <p>An exemption is considered as relevant for a ship call if:</p> <ul style="list-style-type: none"> • It applies to the same ship, and • Its list of exempted ports includes the Port Of Call, and • If the details are regarding Non-EU-Departure hazmat, or waste, or security or Bunkers towards Port of Call, its validity period covers the ATAPortOfCall, or the ETAToPortOfCall (if there is no ATAPortOfCall), and • If the details are regarding EU-Departure Hazmat or Bunkers towards Next Port, its validity period covers the ATDPortOfCall, or the ETDFromPortOfCall (if there is no ATDPortOfCall). <p>If one or more exemptions are found, the details are fetched to be included in the response message.</p>
Step 9	<p>The QueryResults content are validated (against xsd patterns to ensure that valid XML content is to be generated).</p> <p>In case of invalid content, this specific record is excluded from the QueryResults and</p> <p>an error message is sent via email to the requestor with the information "SafeSeaNet ShipCall Request / Response. Invalid PortPlus notification content".</p>
Step 10	The system constructs a response message.
Step 11	The system sends the response to the data requestor.
Alternative Use Case Description	Invalid Request
Step 1.1	The ShipCall request criteria are invalid

Step 1.2	The system generates a receipt indicating the error and StatusCode = "InvalidFormat" and a StatusMessage indicating that the search criteria are incorrect.
Alternative Use Case Description	No relevant data found.
Step 2.1	The system identifies that no voyage satisfies the request criteria. The system generates and sends a negative response to the data requestor.
Step 2.2	The system generates a ShipCallRes with StatusCode = "NotFound".
Alternative Use Case Description	Exceeded Response Time/ Negative receipt
Step 3.1	If the response time (as defined in the request – see Special requirements) has been exceeded, or if a negative receipt has been received from the data provider's NCA Application, the system sends a Response message to the data requester with the summary that it holds in the database (notification details).
Alternative Use Case Description	Access Control Check fails
Step 4.1	If the Access Control fails to apply the restrictions of Step 2 and 3, The system identifies that the user is not granted the permission to submit the request (is not granted the relative TASK; see step 3) or the geographical access rights restrictions on ShipCall requests does not permit submitting a request for the PortOfCall indicated in the request criteria.
Step 4.2	The system generates a receipt indicating the error and StatusCode = "AccessDenied".
Input(s)	Request specific data.
Output(s)	Notification Details are sent to the Data Requester (SSN User or NCA Application). The system shall respond to a requesting NCA Application in XML format. For Cargo Manifest the response message defines the URL where the document can be downloaded from or the contact details of the authority/person to acquire the details from. [Optional - in case of invalid ShipCall content] email to the requestor with the information that the "following <i>ShipCall Ids</i> could not be processed".
Timer(s)	-
Business Process(es) Reference	Response Timer
Associated Use Case(s)	Included UC: UC-SSN-MSG-01: Handle Incoming Message
Special Requirements	<ol style="list-style-type: none"> 1. Once a request has been received, the system initiates a response timer. If the time interval to be applied is not defined in the request message, the system sets the timer with a predefined value. 2. Voyages with status "On-hold" are not provided in the results

3.2.3.2 UC-SSN-SCREQ-21: Request ShipCall from the Web

Use Case Req ID	UC-SSN-SCREQ-21	
Use Case Name	Request ShipCall from the Web	
Purpose	The current Use Case describes the functionality related to the search for Voyages from the Web interface.	
Subsystem	SSN Web application	
Primary Actor(s)	SSN Human User	
Precondition(s)	<p>The Actor has been authenticated in the system. At least one vessel with voyage(s) exist in the system. The actor is granted the permission for ShipCall requests, based on the following:</p> <p>On the SHIPCALL_REQUESTOR task, location restrictions on the PortOfCall and source restrictions on the data provider.</p> <p>On the HAZMAT_REQUESTOR task if HazmatDetails are requested.</p> <p>On the SECURITY_REQUESTOR task if SecurityDetails are requested.</p> <p>On the WASTE_REQUESTOR task if WasteDetails are requested</p> <p>On the BUNKERS_REQUESTOR task if BunkersDetails are requested</p> <p>On the EXEMPTIONS_REQUESTOR task if Exemption information exists</p>	
Postcondition(s)	Voyage(s).	
Trigger(s)	The Actor wants to find information for a voyage(s) of a vessel.	
Use Case Description	Primary Workflow	
Step 1	The user navigates to the web page for searching voyages.	
Step 2	<p>The user selects a predefined menu item that corresponds to a ship call request of the types indicated hereunder. The user shall enter the criteria applicable per case:</p> <p><i>Relevant Voyages:</i></p> <p>Mandatory parameters: IMONumber or MMSINumber.</p> <p>Result:</p> <ul style="list-style-type: none"> • Latest Incident Report notification for the specified vessel, considering that Actor is assigned the ALERT_REQUESTOR permission that corresponds to the incident type of the retrieved notification. • List of up to 13 most relevant Voyages in relation to the query Timestamp. <ul style="list-style-type: none"> – With ETA/ATAPortOfCall after and closest to the query timestamp or – ATDPortOfCall and ATAPortOfCall before and closest to the query timestamp or – ETDLastPort before and closest to the query timestamp, ATAPortOfCall not null and ATDPortOfCall after and closest to the query timestamp or – ETDLastPort before and closest to the query timestamp and PortOfCall unknown. 	

- Hazmat/Waste/Security/Bunkers summary data are provided.
- Request for Hazmat/Waste/Security/Bunkers details is included for each relevant voyage given that actor is granted the respective permission.
- If actor has the access rights to requesting exemption data, the system searches for exemptions in the SSN database which are applicable to the ship. However, no geographical restrictions will be applied on exemption results included in a ship call response.

ExpectedCallOfSelectedShip:

Mandatory parameters: IMONumber or MMSINumber;

Optional parameters: StartDateTime.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: Details of a Voyage

- Without ATAPortOfCall, and
- With ETAToPortOfCall after and closest to StartDateTime.
- Hazmat/Security/Waste/Bunkerssummary dataare provided.
- Request for Hazmat/Waste/Security/Bunkers details is included for the voyage given that actor is granted the respective permission.

MostRecentArrivalOfSelectedShip:

Mandatory parameters: IMONumber or MMSINumber;

Optional parameters: StartDateTime.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: Details of a Voyage

- With ATAPortOfCall before and closest to StartDateTime, and
- With ATDPortOfCall, if available, after StartDateTime.
- Hazmat/Security/Waste/Bunkerssummary dataare provided.
- Request for Hazmat/Waste/Security/Bunkers details is included for the voyage given that actor is granted the respective permission.

MostRecentDepartureOfSelectedShip:

Mandatory parameters: IMONumber or MMSINumber;

Optional parameters: StartDateTime.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: Details of a Voyage

- Central SSN will provide the ship call with ATDPortOfCall before and closest to StartDateTime.

- Hazmat/Security/Waste/Bunkers summary data are provided.
- Request for Hazmat/Waste/Security/Bunkers details is included for the voyage given that actor is granted the respective permission.

RecentAndCurrentShipCallsOfSelectedShip:

Mandatory parameters: IMONumber or MMSINumber;

Optional parameters: StartDateTime, EndDateTime, NumberOfCalls.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: List of Voyages

- With ATAPortOfCall within the time period defined by StartDateTime and EndDateTime.
- If no time period is defined, Central SSN will provide the list of latest [NumberOfCalls] consolidated PortPlus messages with ATAPortOfCall before SentAt.
- Hazmat/Security/Waste/Bunkers summary data are provided.
- Request for Hazmat/Waste/Security/Bunkers details is included for each voyage in the list given that actor is granted the respective permission.

ExpectedShipCallsAtEUPort:

Mandatory parameters: PortOfCall (At least one of: Location Code, Port Facility Code, Country shall be defined);

Optional parameters: StartDateTime, EndDateTime, Ship Flag, NumberOfCalls. Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: List of Voyages

- With ETAToPortOfCall within the time period defined by StartDateTime and EndDateTime, and
- Without ATAPortOfCall.
- If no time period is defined, Central SSN will provide the list of [NumberOfCalls] correlated voyages:
- With ETAToPortOfCall after SentAt, and
- Without ATAPortOfCall.
- Hazmat/Security/Waste/Bunkers summary data are provided.
- Request for Hazmat/Waste/Security/Bunkers details is included for each voyage in the list given that actor is granted the respective permission.

CurrentShipCallsAtEUPort:

Mandatory parameters: PortOfCall (At least one of Location Code, Port Facility Code, Country shall be defined);

Optional parameters: StartDateTime, Ship Flag.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: List of Voyages

- With PortOfCall = defined PortOfCall, and
- With ATAPortOfCall after StartDateTime, and
- Without ATDPortOfCall.
- Hazmat/Security/Waste/Bunkers summary data are provided.
- Request for Hazmat/Waste/Security/Bunkers details is included for each voyage in the list given that actor is granted the respective permission.

CompletedShipCallsAtEUPort:

Mandatory parameters: PortOfCall (At least one of Location Code, Port Facility Code, Country shall be defined);

Optional parameters: StartDateTime, Ship Flag.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: List of Voyages

- With PortOfCall = defined PortOfCall, and
- With ATDPortOfCall after StartDateTime.
- Hazmat/Security/Waste/Bunkers summary data are provided.
- Request for Hazmat/Waste/Security/Bunkers details is included for each voyage in the list given that actor is granted the respective permission.

LatestCallUpdates:

Mandatory parameters: StartDateTime, EndDateTime. Business rules applicable [R4]Table 4– Description of queries supported by the ShipCallReq message.

Result: List ofVoyages which were registered or updated within the specified time period.

- Hazmat/Security/Waste/Bunkers summary data are provided.
- Request for Hazmat/Waste/Security/Bunkers details is included for each voyage in the list given that actor is granted the respective permission.

ListExpectedCallsOfSelectedShip:

Mandatory parameters: IMONumber or MMSINumber;

Optional parameters: StartDateTime.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: List of Voyages

- Without ArrivalDetails element, and
- With ETAToPortOfCall after StartDateTime, and

- Without ATAPortOfCall.
- Hazmat/Security/Waste/Bunkerssummary data are provided.
- Request for Hazmat/Waste/Security/Bunkers details is included for each voyage in the list given that actor is granted the respective permission.

SelectedShipCall:

Mandatory parameters: ShipCallID;

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Results:

- a) a Voyage
 - With the specified ShipCallID;
 - Registered vessel IMO Number, MMSI Number, CallSign, ShipName, Flag;
 - Hazmat/Security/Waste/Bunkerssummary data are provided.
 - Request for Hazmat/Waste/Security/Bunkers details is included for the voyage given that actor is granted the respective permission.
- b) list of notifications transmitted to SSN-EIS that refer to the same ship call.

GetActiveHazmatForSelectedShip:

Mandatory parameters: IMONumber or MMSINumber;

Optional parameters: StartDateTime.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: the Voyage that is the most relevant and associated hazmat details which are active at StartDateTime.

- Information may come from different PortPlus messages (different values of ShipCallID).
- Hazmat summary data are provided.
- Request for Hazmat details is included for the voyage given that actor is granted HAZMAT_REQUESTOR permission.
- Security/Waste/Bunkerssummary data are provided, if respective information has also been reported for the voyage.
- Request for Security/Waste/Bunkers details is included given that respective information has also been reported for the voyage and actor is granted the respective permission.
- If actor has the access rights to requesting exemption data, the system searches for exemptions of type "Hazmat" in the SSN database, which are applicable to the ship.

GetActiveSecurityForSelectedShip:

Mandatory parameters: IMONumber or MMSINumber;

Optional parameters: StartDateTime.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: the most relevant Voyageand associated security details.

- Without ATDPortOfCall and with closest ETAToPortOfCall or ATAPortOfCall to StartDateTime
- Security summary data are provided.
- Request for Security details is included for thevoyage given that actor is granted SECURITY_REQUESTOR permission.
- Hazmat/Waste/Bunkerssummary data are provided, if respective information has also been reported for the voyage.
- Request for Hazmat/Waste/Bunkers detailsis included given that respective information has also also been reported for the voyage and actor is granted the respective permission.
- If actor has the access rights to requesting exemption data, the system searches for exemptions of type "Security" in the SSN database, which are applicable to the ship.

GetActiveWasteForSelectedShip:

Mandatory parameters: IMONumber or MMSINumber;

Optional parameters: StartDateTime.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCallReq message.

Result: the most relevant Voyageand associated waste details.

- Without ATDPortOfCall and
- With closest ETAToPortOfCall or ATAPortOfCall to StartDateTimeWaste summary data are provided.
- Request for Waste details is included for thevoyage given that actor is granted WASTE_REQUESTOR permission.
- Hazmat/Security/Bunkerssummary data are provided, if respective information has also been reported for the voyage.
- Request for Hazmat/Security/Bunkers detailsis included given that respective information has also been reported for the voyage and actor is granted the respective permission.

If actor has the access rights to requesting exemption data, the system searches for exemptions of type "Waste", "WasteNotification", "WasteDelivery" and "WasteFees" in the SSN database, which are applicable to the ship.

GetActiveBunkersForSelectedShip:

Mandatory parameters: IMONumber or MMSINumber;

Optional parameters: StartDateTime.

Business rules applicable [R4] Table 4– Description of queries supported by the ShipCall Req message.

	<p>Result: the most relevant Voyage and associated bunkers details which are active at StartDateTime.</p> <ul style="list-style-type: none"> – Bunkers confirmation data are provided. – Request for Bunkers details is included for the voyage given that actor is granted BUNKERS_REQUESTOR permission. – Hazmat/Waste/Security summary data are provided, if respective information has also been reported for the voyage. – Request for Hazmat/Waste/Security details is included given that respective information has also been reported for the voyage and actor is granted the respective permission.
Step 3	<p>The system searches the SSN database for a relevant voyage(s) based on the provided request criteria. The business rules for voyage retrieval defined in Annex A: Business Rules are also applicable.</p>
Step 4a	<p>In case of a request for a single voyage, the system displays a voyage which complies with the search criteria and the requestor's access rights, location restrictions (applied to the "PortOfCall" of the voyage) and source restrictions (applied to the provider of voyage data). If the voyage complies with the location and source restrictions, then the voyage is provided.</p> <p>Location restrictions are applied to the PortOfCall of the voyage as follows:</p> <ul style="list-style-type: none"> ○ In case the SHIPCALL_REQUESTOR permission has a "Specific locations" location restriction, the PortOfCall shall be within the locodes assigned to the applicable duty/ies for the requestor's organisation. ○ In case the SHIPCALL_REQUESTOR permission has a "User's country" location restriction, the PortOfCall shall belong to the requestor's country; ○ In case the SHIPCALL_REQUESTOR permission has a "Countries list" location restriction, the PortOfCall shall belong to the countries assigned to the applicable regional agreement(s). <p>Source restrictions are applied to the provider of voyage data as follows:</p> <ul style="list-style-type: none"> ○ In case the SHIPCALL_REQUESTOR permission has a "Countries list" source restriction, the notifier's country (country of the provider of the voyage data) shall be within the countries assigned to the applicable regional agreement(s). ○ In case the SHIPCALL_REQUESTOR permission has a "User's country" source restriction, the notifier's country (country of the provider of the voyage data) shall be equal to the requestor's country. <p>A magnifier glass – per case - indicates if the voyage has Hazmat and/or Security and/or Waste and/or Bunkers.</p> <p>The user clicks on a magnifier glass for the details.</p>
Step 4b	<p>In case of a request for a list of voyages the system displays the voyages that match the criteria and the requestor's access rights, location restrictions (applied to "PortOfCall" of each voyage) and source restrictions (applied to the provider of each voyage). Location and source restrictions are applied as in Step 4a. Only the voyages which comply with the location and source restrictions are provided.</p>

	Per record and per case a magnifier glass indicates if the voyage has Hazmat and/or Security and/or Waste and/or Bunkers. The user clicks on a magnifier glass for the details.
Step 5	The system identifies the relative information being Hazmat and/or Security and/or Waste and/or Bunkers. The system identifies that the requested detailed information in XML must be acquired from an NCA Application (data provider). The system issues a ShipCall Request to the NCA application for the details. In case the details are waste and waste details had been provided already in a PortPlus notification, EIS will provide the information stored at central level without sending a request to the NCA application (data provider) for the details.
Step 6	The system receives within the timeout interval the response from the NCA application. It parses the QueryResults part of the response message.
Step 7	If the requestor has the access rights to requesting exemption data, the system searches for exemptions in the SSN database which are relevant based on the same rule as introduced in UC-SSN-SCREQ-20. If one or more exemption is found the details are fetched to be included in the details of the response to the user.
Step 8	The system constructs the response to the user.
Step 9	The system displays the response.
Alternative Use Case Description	No relevant data found.
Step 1.1	The system identifies that no voyage satisfies the request criteria. The system prompts the user that no voyages were found.
Alternative Use Case Description	Exceeded Response Time/ Negative receipt
Step 2.1	If the response time (as defined in the request – see Special requirements) has been exceeded, or if a negative receipt has been received from the data provider's NCA Application, the system displays to the data requester with the summary that it holds in the database (notification details).
Input(s)	Request specific data.
Output(s)	Notification Details are displayed to the user.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	Voyages with status "On-hold" are not provided in the results

3.2.3.3 UC-SSN-SCRES-22: Process ShipCall Response

Use Case Req ID	UC-SSN-SCRES-22
-----------------	------------------------

Use Case Name	Process ShipCall Response	
Purpose	Covers the functionality related to the systems' actions upon receiving a response for details from a Member State.	
Subsystem	SSN Core	
Primary Actor(s)	NCA Application	
Precondition(s)	The NCA Application (data provider) has received a request from SSN for providing detailed information which is declared available through a previously sent notification.	
Postcondition(s)	The system sends the response message with the requested information to the NCA (data requester) that initiated the request.	
Trigger(s)	Response message received from the NCA Application (data provider).	
Use Case Description	Primary Workflow	
Step 1	The system upon reception of a response message checks the associated response timer and the associated request message.	
Step 2	If the response period has not been exceeded, the system generates and sends the response to the Member State that initiated the request through XML message.	
Alternative Use Case Description	Response Time Exceeded	
Step 1.1	If the response time has been exceeded, the system shall only log the received response.	
Alternative Use Case Description	No matching Vessel Data	
Step 2.1	If the Vessels criteria defined in the response message do not match the vessel definition in the SSN data store then the system shall return a status code OK to the NCA application along with a warning that indicates the vessel particulars as defined in the SSN data store.	
Input(s)	Response message.	
Output(s)	Requested information.	
Timer(s)	Response Timer	
Business Process(es) Reference	-	
Associated Use Case(s)	Included UC: UC-SSN-MSG-01: Handle Incoming Message	
Special Requirements	-	

3.2.3.4 UC-SSN-EXF-23: Find Exemption (Web)

Use Case Req ID	UC-SSN-EXF-23	
Use Case Name	Find Exemption (Web)	
Purpose	The current Use Case describes the functionality related to the search for an exemption associated to a vessel in the Web interface.	

Subsystem	SSN Web Application
Primary Actor(s)	SSN Human User
Precondition(s)	The Actor has been authenticated in the system. At least one vessel with associated exemption has been inserted in the system. The actor is granted the permission for exemption requests, based on the following: On the EXEMPTIONS_REQUESTOR task, location restrictions on granting country and exempted ports, as well as source restrictions on the data provider.
Postcondition(s)	A list of exemptions.
Trigger(s)	The Actor wants to find information for an exemption associated to a vessel.
Use Case Description	Primary Workflow
Step 1	The user navigates to the web page for searching exemptions.
Step 2	<p>The Actor searches for an exemption by entering Vessel, Issuing Country, type of Exemption and Validity Period as criteria.</p> <p>In case the EXEMPTIONS_REQUESTOR permission has any location restriction the system adds criterion on granting country being SSN participant (a specific type, i.e. CountryTypeId: 10, is created in CCD for SSN participant countries; SSN-EIS application saves this information as a Boolean indicator in COUNTRIES.IS_SSN column).</p> <p>In case of "Specific locations" location restriction, the system adds a criterion on the exempted ports being contained in the locodes assigned to the applicable duty/ies for the requestor's organisation ;</p> <p>In case of "User's country" location restriction, the system adds a criterion on the granting country being equal to the requestor's country.</p> <p>In case of "Countries list" location restriction, the system adds a criterion on the granting country being contained in the countries assigned to the applicable regional agreement(s).</p>
Step 3	<p>The system displays the list of exemptions that match the criteria as well the requestor's access rights, location restrictions (applied to granting country and exempted ports, as described in Step 2) and source restrictions (applied to the provider of each exemption). Only the exemptions which comply with the location and source restrictions are provided.</p> <p>Source restrictions are applied to the provider of the exemption information as follows:</p> <ul style="list-style-type: none"> ○ In case the EXEMPTIONS_REQUESTOR permission has "Countries list" source restriction, the notifier's country (country of the provider of exemption information) shall be within the countries assigned to the applicable regional agreement(s). ○ In case the EXEMPTIONS_REQUESTOR permission has "User's country" source restriction, the notifier's country (country of the provider of exemption information) shall be equal to the requestor's country.

	Exemption details include the information listed in UC-SSN-EXC-11 (Step 4).
Alternative Use Case Description	Search an Exemption – No Results
Step 3.1	No Exemption matches the search criteria. The system returns a related message and the flow continues from step 2 of the Primary Workflow.
Input(s)	Search Criteria related to the vessel and exemption.
Output(s)	Exemptions that satisfy the criteria are displayed.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	-

3.2.3.5 UC-SSN-MRSREQ-24: Process Ship (AIS or MRS) Request (XML)

Use Case Req ID	UC-SSN-MRSREQ-24
Use Case Name	Process Ship AIS or MRS Request (XML)
Purpose	Covers the functionality related to the system's actions upon receiving a request for Ship notification information through the System Interface
Subsystem	SSN Core, SSN XML Protocol, SSN SEG WS
Primary Actor(s)	SSN System User
Precondition(s)	A request for information has been received by the system.
Postcondition(s)	The system has processed the request and the response message is provided to the System User's application.
Trigger(s)	A request for information has been sent to SSN in XML format
Use Case Description	Primary Workflow
Step 1	<p>The system performs the authorization checks defined in UC-SSN-MSG-01 for the Actor submitting the request.</p> <p>The system identifies the access rights restrictions on ship requests, based on the following:</p> <p>On the SHIP_AIS_REQUESTOR/SHIP_MRS_REQUESTOR task, in case of Ship AIS/MRS respectively, location restrictions on the PortOfCall and source restrictions on the data provider.</p>
Step 2	<p>Depending on the Ship Notification Type (ShipNotType field) the system performs as follows:</p> <p>When ShipNotType=AIS,</p> <p>the system retrieves the latest AIS information available in SSN (on the basis of "Timestamp") for the specified vessel (identified by the "IMONumber", or the "MMSINumber" if "IMONumber" is not provided).</p>

If the attribute "SenderCountryId" (optional) is also provided, it will be used - in addition to previous criteria - to retrieve AIS information originating from that country; that is the sender's (domain attribute AisNotification->message->sender) country matching the given one.

The system retrieves the latest AIS information that matches the criteria as well as requestor's access rights, location restrictions (applied to the "PortOfCall" of the notification) and source restrictions (applied to the provider of AIS information). AIS information is provided only if it complies with the location and source restrictions.

Location restrictions are applied to the PortOfCall of the notification as follows:

- In case the SHIP_AIS_REQUESTOR permission has "Specific locations" location restriction, the PortOfCall shall be within the locodes assigned to the applicable duty/ies for the requestor's organisation.
- In case the SHIP_AIS_REQUESTOR permission has "User's country" location restriction, the PortOfCall shall belong to requestor's country;
- In case the SHIP_AIS_REQUESTOR permission has "Countries list" location restriction, the PortOfCall shall belong to the countries assigned to the applicable regional agreement(s).

Source restrictions are applied to the provider of AIS information as follows:

- In case the SHIP_AIS_REQUESTOR permission has "Countries list" source restriction, the notifier's country (country of the provider of AIS information) shall be within the countries assigned to the applicable regional agreement(s).

In case the SHIP_AIS_REQUESTOR permission has "User's country" source restriction, the notifier's country (country of the provider of AIS information) shall be equal to the requestor's country.

For the AIS Notification selected, the system will form a request (SSN2MS_Ship_Req) of type AIS, asking for additional information from the data provider (sender of AisNotification).

When **ShipNotType=MRS**,

the system retrieves the latest MRS information available in SSN (on the basis of "ReportingDateTime") for the specified vessel (identified by the "IMONumber", or the "MMSINumber" if "IMONumber" is not provided).

If the attribute "MRSIdentification" (optional) is provided, it will be used to track down the latest MRS Notification corresponding to this identifier only.

If the attribute "SenderCountryId" (optional) is also provided, it will be used - in addition to previous criteria - to retrieve MRS Information originating from that country; that is the sender's (domain attribute MrsNotification->message->sender) country matching the given one.

The system retrieves the latest MRS information that matches the criteria as well as requestor's access rights, location restrictions (applied to the "PortOfCall" of the notification) and source restrictions (applied to the provider of MRS information). MRS information is provided only if it complies with the location and source restrictions.

Location restrictions are applied to the PortOfCall of the notification as follows:

- In case the SHIP_MRS_REQUESTOR permission has "Specific locations" location restriction, the PortOfCall shall be within the locodes assigned to the applicable duty/ies for the requestor 's organisation.
- In case the SHIP_MRS_REQUESTOR permission has "User's country" location restriction, the PortOfCall shall belong to requestor's country;
- In case the SHIP_MRS_REQUESTOR permission has "Countries list" location restriction, the PortOfCall shall belong to the countries assigned to the applicable regional agreement(s).

Source restrictions are applied to the provider of MRS information as follows:

- In case the SHIP_MRS_REQUESTOR permission has "Countries list" source restriction, the notifier's country (country of the provider of MRS information) shall be within the countries assigned to the applicable regional agreement(s).
- In case the SHIP_MRS_REQUESTOR permission has "User's country" source restriction, the notifier's country (country of the provider of MRS information) shall be equal to the requestor's country.

The matching MRS Notification can be of three kinds:

- **MRS Notification received from Web interface.** Again, no detailed information needs to be obtained from the data provider; all information including MRS Notification Details (MrsNotificationDetails entity) is available in SSN, hence the system forms a full reply to data requestor.
- **MRS Notification received from XML interface.** The system needs to retrieve MRS Notification details from the data provider; thus, the system:
 - Creates an Additional Information Request (SSN2MS_Ship_Req) message and dispatches it to the corresponding Member State.
 - Persists the request to an intermediate waiting queue. The messages will remain in the queue for a configurable time period or until a corresponding response is received from the data provider.

The system will process the response returned from the Member State as described in "UC-SSN-MRSRES-26 – Process MRS Response".

Step 3	Given that the system has processed the response returned from data provider, it has already created a reply (consolidated information including the initial request, the relevant notification and the detailed information from the data provider) stored in the waiting queue. It should be noted that SSN is not examining the response content. Even a negative response will be forwarded to data requestor.
Step 4	The system logs and sends the reply to the data requestor.
Alternative Use Case Description	No matching Notifications found for the given request criteria
Step 2.1	The system creates a reply which indicates that no notification was found.
Step 2.2	The system logs the response message and sends it to the data requestor.
Alternative Use Case Description	No response has been received from the data provider (MS2SSN_Ship_Res) in the configured time period.
Step 3.1	The system retrieves from the intermediate waiting queue the request awaiting response from the data provider and forms a reply with the data available in the SSN central system; that is the MRS Notification data alone, no additional information from the data provider.
Step 3.2	The system logs and sends the reply to the data requestor.
Alternative Use Case Description	The Actor sending the request does not comply to authorization requirements
Step 1.1	The system creates a reply which indicates that the Actor is not authorized to send the request.
Step 1.2	The system logs the response message and sends it to the data requestor.
Input(s)	Ship Request
Output(s)	Ship Response
Timer(s)	Response Timer
Business Process(es) Reference	-
Associated Use Case(s)	Included UC: UC-SSN-MSG-01: Handle Incoming Message Included UC: UC-SSN-MRSRES-26: Process MRS response
Special Requirements	

3.2.3.6 UC-SSN-MRSREQ-25: Request Ship (AIS or MRS) from the Web

Use Case Req ID	UC-SSN-MRSREQ-25
Use Case Name	Request Ship (AIS or MRS) from the Web
Purpose	The current Use Case describes the functionality related to the search for Ship notifications from the Web interface.
Subsystem	SSN Web application

Primary Actor(s)	SSN Human User
Precondition(s)	The Actor has been authenticated in the system. The actor is granted the pertinent permission, based on the following: On the SHIP_AIS_REQUESTOR/SHIP_MRS_REQUESTOR task, in case of Ship AIS/MRS respectively, location restrictions on the PortOfCall and source restrictions on the data provider.
Postcondition(s)	The request has been treated and the answer provided to the SSN user
Trigger(s)	-
Use Case Description	The Actor wants Ship Notification details.
Step 1	The Actor navigates to the web page for requesting Ship Notification details (AIS/MRS Information -> Latest AIS/MRS for selected ship).
Step 2	The system prompts the Actor to fill in the criteria to be used for requesting Ship Notification details (ShipNotificationDetailsRequest page).
Step 3	The system validates criteria according to the XML RG.
Step 4	<p>The system displays a list of Notifications matching the criteria.</p> <p>Depending on the Ship Notification Type (ShipNotType field) the results are collected as follows:</p> <p>When ShipNotType=AIS,</p> <p>the system retrieves the latest AIS information available in SSN (on the basis of "Timestamp") for the specified vessel (identified by the "IMONumber", or the "MMSINumber" if "IMONumber" is not provided).</p> <p>If the attribute "SenderCountryId" (optional) is also provided, it will be used - in addition to previous criteria - to retrieve AIS information originating from that country; that is the sender's (domain attribute AisNotification->message->sender) country matching the given one.</p> <p>When ShipNotType=MRS,</p> <p>the system retrieves the latest MRS information available in SSN (on the basis of "ReportingDateTime") for the specified vessel (identified by the "IMONumber", or the "MMSINumber" if "IMONumber" is not provided).</p> <p>If the attribute "MRSIdentification" (optional) is provided, it will be used to track down the latest MRS Notification corresponding to this identifier only.</p> <p>If the attribute "SenderCountryId" (optional) is also provided, it will be used - in addition to previous criteria - to retrieve MRS Information originating from that country; that is the sender's (domain attribute MrsNotification->message->sender) country matching the given one.</p>

	<p>The system retrieves the latest AIS/MRS information that matches the criteria as well as requestor's access rights, location restrictions (applied to the "PortOfCall" of the notification) and source restrictions (applied to the provider of AIS/MRS information). AIS/MRS information is provided only if it complies with the location and source restrictions.</p> <p>Location restrictions are applied to the PortOfCall of the notification as follows:</p> <ul style="list-style-type: none"> ○ In case the SHIP_AIS_REQUESTOR/SHIP_MRS_REQUESTOR permission has "Specific locations" location restriction, the PortOfCall shall be within the locodes assigned to the applicable duty/ies for the requestor's organisation. ○ In case the SHIP_AIS_REQUESTOR/SHIP_MRS_REQUESTOR permission has "User's country" location restriction, the PortOfCall shall belong to requestor's country; ○ In case the SHIP_AIS_REQUESTOR/SHIP_MRS_REQUESTOR permission has "Countries list" location restriction, the PortOfCall shall belong to the countries assigned to the applicable regional agreement(s). <p>Source restrictions are applied to the provider of AIS/MRS information as follows:</p> <ul style="list-style-type: none"> ○ In case the SHIP_AIS_REQUESTOR/SHIP_MRS_REQUESTOR permission has "Countries list" source restriction, the notifier's country (country of the provider of AIS/MRS information) shall be within the countries assigned to the applicable regional agreement(s). ○ In case the SHIP_AIS_REQUESTOR/SHIP_MRS_REQUESTOR permission has "User's country" source restriction, the notifier's country (country of the provider of AIS/MRS information) shall be equal to the requestor's country.
Step 5	The Actor selects a Notification from the list.
Step 6	The system displays Notification information; includes only data maintained in SSN local store.
Step 7	The Actor requests details for the currently displayed Notification by clicking on the magnifier button under details.
Step 8	<p>Depending on the Ship Notification Type (ShipNotType field) the system performs as follows:</p> <p>When ShipNotType=AIS, the system:</p> <ul style="list-style-type: none"> • forms a request (SSN2MS_Ship_Req) of type AIS, asking for additional information from the data provider (sender of AisNotification). • Persists the request to an intermediate waiting queue. The messages will remain in the queue for a configurable time period or until a corresponding response is received from the data provider <p>The system will process the response returned from the Member State as described in "UC-SSN-MRSRES-26 – Process MRS Response". All data, including Notification details returned from data provider are temporarily stored in an intermediate storage (waiting queue).</p>

	<p>When ShipNotType=MRS, the currently displayed MRS Notification can be of three kinds:</p> <ul style="list-style-type: none"> • MRS Notification received from Web interface. No detailed information needs to be obtained from the data provider; all information including MRS Notification Details (MrsNotificationDetails entity) is available in SSN, hence the system stores the information (Notification details) temporarily stored in an intermediate storage (waiting queue). • MRS Notification received from XML interface. The system needs to retrieve MRS Notification details from the data provider; thus, the system: <ul style="list-style-type: none"> ○ Creates an Additional Information Request (SSN2MS_Ship_Req) message and dispatches it to the corresponding Member State. ○ Persists the request to an intermediate waiting queue. The messages will remain in the queue for a configurable time period or until a corresponding response is received from the data provider. <p>The system will process the response returned from the Member State as described in "UC-SSN-MRSRES-26 – Process MRS Response".</p> <p>In every case, the requested data are placed in an intermediate queue, awaiting the web application to get them (poll for data).</p> <p>It should be noted that SSN is not examining the response content. A negative response is a perfectly valid response and will be returned to the Actor's screen.</p>
Step 9	The system redirects to a Ship Notification Details Response page, where it constantly (with a configurable interval) polls for the corresponding response.
Step 10	The system displays Notification details.
Alternative Use Case Description	Validation failed
Step 2.9	The system redirects to the Ship Notification Details Request page, where it clearly indicates the errors preventing the successful submission of the request.
Step.2.10	The flow continues from Step 2 of the main flow.
Alternative Use Case Description	Time-out occurred while waiting for response.
Step 3.9	The system informs the Actor that the system wasn't able to collect the requested information in a timely fashion.
Input(s)	Request specific data.
Output(s)	Ship Notification Details are displayed to the Actor.
Timer(s)	-

Business Process(es) Reference	-
Associated Use Case(s)	Included UC-SSN-MRSRES-26: Process MRS response
Special Requirements	

3.2.3.7 UC-SSN-MRSRES-26: Process MRS Response

Use Case Req ID	UC-SSN-MRSRES-26	
Use Case Name	Process MRS Response	
Purpose	Describes the system actions performed upon reception of a MRS response. The message is sent from the data provider as a response to previously sent request, for detailed MRS information, from SSN central application.	
Subsystem	SSN Core	
Primary Actor(s)	NCA Application	
Precondition(s)	A request for Ship Notification details has been accepted and SSN has sent a request for detailed MRS information to the data provider NCA Application (SSN2MS_Ship_Req)	
Postcondition(s)	The system has consolidated all the information for the MRS in an intermediate waiting queue ready to be dispatched to data requestor. This information includes, the request sent from requestor NCA application, the relevant notification in the SSN central application and the response returned from data provider's NCA application.	
Trigger(s)	The NCA Application has sent the response message (MS2SSN_Ship_Res)	
Use Case Description		
Step 1	The system validates the response against the XMLRG business rules.	
Step 2	The system associates the response with the corresponding request awaiting reply.	
Step 3	The system verifies that the response is compliant to the corresponding request (verifies the search criteria/vessel identification).	
Step 5	<p>The system complements the corresponding MRS information stored in the waiting queue, with the detailed information provided in the response.</p> <p>In fact, the information dispatched to data requestor is the one returned from the data provider; locally stored information is not taken into account.</p>	
Alternative Use Case Description	No request found for the received response. This case may arise when a delayed response from the data provider is arrived, but the corresponding request has been removed due to timeout.	

Step 2.1	The delayed response is silently discarded. No further Ship Response is sent to the initial; data requestor.
Alternative Use Case Description	Response is not compliant to corresponding request
Step 3.1	The response is silently discarded. No further Ship Response is sent to the initial; data requestor.
Input(s)	Response message.
Output(s)	Requested information.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	Included UC: UC-SSN-MSG-01: Handle Incoming Message
Special Requirements	-

3.2.3.8 UC-SSN-MRSLREQ-27: Process MRS List Request (XML)

Use Case Req ID	UC-SSN-MRSLREQ-27	
Use Case Name	Process Ship List Request (XML)	
Purpose	Covers the functionality related to the system's actions upon receiving a request for a list of MRS notification information for a specified geographical area in given time period through the system interface.	
Subsystem	SSN Core	
Primary Actor(s)	SSN System User	
Precondition(s)	A request for a list of information has been received by the system.	
Postcondition(s)	The system has processed the request and returned a response to data requestor.	
Trigger(s)	A request for a list of MRS notifications has been sent to SSN in XML format.	
Use Case Description	Primary Workflow	
Step 1	<p>The system performs the authorization checks defined in UC-SSN-MSG-01 for the Actor submitting the request.</p> <p>The system identifies the access rights on ship list requests, based on the SHIP_MRS_REQUESTOR task, location restrictions on the PortOfCall and source restrictions on the data provider;</p>	
Step 2	The system validates the request against the XMLRG business rules.	
Step 3	<p>The system retrieves all ShipNotifications corresponding to the criteria defined in the request;</p> <p>all MRS notifications where ReportingDateTime is between "StartDateTime" and "EndDateTime"</p>	

	<p>If attribute "SenderCountryId" (optional) is used, the results are narrowed to notifications sent by a specific MS.</p> <p>The system retrieves the MRS information that matches the criteria as well as requestor's access rights, location restrictions (applied to the "PortOfCall" of the notification) and source restrictions (applied to the provider of MRS information). MRS information is provided only if it complies with the location and source restrictions.</p> <p>Location restrictions are applied to the PortOfCall of the notification as follows:</p> <ul style="list-style-type: none"> ○ In case the SHIP_MRS_REQUESTOR permission has "Specific locations" location restriction, the PortOfCall shall be within the locodes assigned to the applicable duty/ies for the requestor 's organisation. ○ In case the SHIP_MRS_REQUESTOR permission has "User's country" location restriction, the PortOfCall shall belong to requestor's country; ○ In case the SHIP_MRS_REQUESTOR permission has "Countries list" location restriction, the PortOfCall shall belong to the countries assigned to the applicable regional agreement(s). <p>Source restrictions are applied to the provider of MRS information as follows:</p> <ul style="list-style-type: none"> ○ In case the SHIP_MRS_REQUESTOR permission has "Countries list" source restriction, the notifier's country (country of the provider of MRS information) shall be within the countries assigned to the applicable regional agreement(s). ○ In case the SHIP_MRS_REQUESTOR permission has "User's country" source restriction, the notifier's country (country of the provider of MRS information) shall be equal to the requestor's country.
Step 4	The system forms a reply (SSN2MS_Ship_List_Res) for the given request. No request to the data providers is required. The system logs and sends the reply to the data requestor.
Alternative Use Case Description	No matching Notifications found for the given request criteria
Step 1.1	The system creates a reply which indicates that no MRS notification were found.
Step 1.2	The system logs the response message and sends it to the data requestor.
Alternative Use Case Description	The Actor sending the request does not comply to authorization requirements
Step 2.1	The system creates a reply which indicates that the Party sending the request is not authorized to do so.
Step 2.2	The system logs the response message and sends it to the data requestor.
Input(s)	Ship List Request (MS2SSN_Ship_List_Req)
Output(s)	Ship List Response (MS2SSN_Ship_List_Res)

Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	Included UC: UC-SSN-MSG-01: Handle Incoming Message
Special Requirements	

3.2.3.9 UC-SSN-MRSLREQ-28: Process MRS List Request (Web)

Use Case Req ID	UC-SSN-MRSLREQ-28	
Use Case Name	Process Ship List Request submitted from the Web	
Purpose	Covers the functionality related to the system's actions upon receiving a request for a list of MRS notification information for a specified geographical area in given time period submitted from the Web interface.	
Subsystem	SSN Core / SSN Web Application	
Primary Actor(s)	SSN Human User	
Precondition(s)	The Actor has been authenticated in the system. The actor is granted the pertinent permission, based on the following: On the SHIP_MRS_REQUESTOR task, location restrictions on the PortOfCall and source restrictions on the data provider.	
Postcondition(s)	The system has processed the request and returned a response to the Actor.	
Trigger(s)	A request for a list of MRS notifications has been sent to SSN through information submitted from the SSN Web Application.	
Use Case Description	Primary Workflow	
Step 1	The user navigates to the web page for requesting a list of Ship Notifications (AIS/MRS Information > List of MRS notifications for a selected system).	
Step 2	The system prompts the Actor to fill in the criteria to be used for requesting the List of Ship Notifications.	
Step 3	The system validates the ship list request criteria against the XMLRG business rules.	
Step 4	<p>The system retrieves all ShipNotifications corresponding to the criteria defined in the request and renders the results in the same page, just below the given criteria.</p> <p>All MRS notifications where ReportingDateTime is between "StartDateTime" and "EndDateTime"</p> <p>If attribute "SenderCountryId" (optional) is used, the results are narrowed to notifications sent by a specific MS.</p> <p>The system retrieves the MRS information that matches the criteria as well as requestor's access rights, location restrictions (applied to the "PortOfCall" of the notification) and source restrictions (applied to the</p>	

	<p>provider of MRS information). MRS information is provided only if it complies with the location and source restrictions.</p> <p>Location restrictions are applied to the PortOfCall of the notification as follows:</p> <ul style="list-style-type: none"> ○ In case the SHIP_MRS_REQUESTOR permission has "Specific locations" location restriction, the PortOfCall shall be within the locodes assigned to the applicable duty/ies for the requestor 's organisation. ○ In case the SHIP_MRS_REQUESTOR permission has "User's country" location restriction, th PortOfCall shall belong to requestor's country; ○ In case the SHIP_MRS_REQUESTOR permission has "Countries list" location restriction, the PortOfCall shall belong to the countries assigned to the applicable regional agreement(s). <p>Source restrictions are applied to the provider of MRS information as follows:</p> <ul style="list-style-type: none"> ○ In case the SHIP_MRS_REQUESTOR permission has "Countries list" source restriction, the notifier's country (country of the provider of MRS information) shall be within the countries assigned to the applicable regional agreement(s). ○ In case the SHIP_MRS_REQUESTOR permission has "User's country" source restriction, the notifier's country (country of the provider of MRS information) shall be equal to the requestor's country.
Step 5	<p>The Actor selects any of the returned results and navigate to a page where Ship Notification information is displayed. It should be noted that only the notification information stored in the SSN central system is displayed; notification details originating from the data provider are not available.</p> <p>The Actor may move back to results and select another Ship Notification to view available data.</p>
Alternative Use Case Description	Validation fails for the criteria given by the user
Step 1.3	The Actor remains in the search criteria page, where the system clearly indicates errors preventing successful submission of the request. The Actor is prompted to correct them and re-submit the request.
Alternative Use Case Description	No matching Notifications found for the given request criteria.
Step 2.4	A corresponding message is displayed in Actor's screen and is prompted to attempt another search.
Alternative Use Case Description	Cancel Ship List Request.
Step 3.5	<p>Cancellation in any of the screens of the web flow, will clear search results and will let the Actor start the search over over.</p> <p>The flow continues from step 2.</p>
Input(s)	Ship List Request
Output(s)	Ship List Response

Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	
Special Requirements	

3.2.3.10 UC-SSN-CSREQ-33: Process ShipParticulars Request

Use Case Req ID	UC-SSN-CSREQ-33	
Use Case Name	Process ShipParticulars Request	
Purpose	Covers the functionality related to the system's actions upon receiving a request for ShipParticulars information through the system interface.	
Subsystem	SSN Core	
Primary Actor(s)	External Application	
Precondition(s)	A request for ShipParticulars information has been sent to SSN in XML format.	
Postcondition(s)	The system has processed the request and returned a response to data requestor.	
Trigger(s)	A request of type: <ul style="list-style-type: none"> MS2SSN_ShipParticulars_Req MS2SSN_ShipParticulars_Sub has been received by the system.	
Use Case Description	Primary Workflow (MS2SSN_ShipParticulars_Req)	
Step 1	The system performs the authorization checks defined in UC-SSN-MSG-01 for the Actor submitting the request. The system identifies access rights restrictions on ship requests, based on the SHIPPARTICULARS_REQUESTOR task;	
Step 2	The system validates the request against the XMLRG business rules.	
Step 3	The system retrieves the details of a specific Ship or a list of Ship that correspond to the search criteria defined in the request. More specifically: If ShipIdentificationCriteria: a specific Ship that satisfies the search criteria. The criteria updated to also include the CSDID option. If FlagCriteria: a list of Ship that satisfies the search criteria. If CSDrecordUpdateTimeCriteria: a list of Ship that satisfies the search criteria. In addition, the CSDRequestTypeCriteria defines the type of the request [R6].	
Step 4	The system forms a reply (SSN2MS_ShipParticulars_Res) for the given request. The system logs and sends the reply to the data requestor.	
Alternative Use Case Description	Alternative Scenario 1: (MS2SSN_ShipParticulars_Sub)	

Step 2.1	The system validates the request against the XMLRG business rules.
Step 2.2	The system records the request for announcement by the originator for specific flag, date/time criteria. In the case of a cancelation, then the system identifies if an announcement request is already recorded (identified by the MSRefId) and if yes records the cancelation request. If the announcement is not identified then go to step 2.1.
Alternative Use Case Description	No matching Ship or Subscription found for the given request criteria.
Step 3.1	The system creates a reply which indicates that no Ship or Subscription were found.
Step 3.2	The system logs the response message and sends it to the data requestor.
Alternative Use Case Description	The Actor sending the request does not comply to authorization requirements
Step 4.1	The system creates a reply which indicates that the Actor sending the request is not authorized to do so.
Step 4.2	The system logs the response message and sends it to the data requestor.
Input(s)	<ul style="list-style-type: none"> MS2SSN_ShipParticulars_Req MS2SSN_ShipParticulars_Sub
Output(s)	<ul style="list-style-type: none"> SSN2MS_ShipParticulars_Res Subscription recorded in the database.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	Included UC: UC-SSN-MSG-01: Handle Incoming Message
Special Requirements	-

3.2.3.11 UC-SSN-CSANN-34: Process ShipParticulars Announcement

Use Case Req ID	UC-SSN-CSANN-34	
Use Case Name	Process ShipParticulars Announcement	
Purpose	Covers the functionality related to the system's actions when processing subscriptions for Ship Particulars information through the system interface.	
Subsystem	SSN Core	
Primary Actor(s)	External Application	
Precondition(s)	A request for announcement of Ship Particulars information has been recorded in SSN.	
Postcondition(s)	The system sends the announcement response to data requestor.	

Trigger(s)	SHIP_PARTICULARS_SUB scanner
Use Case Description	Primary Workflow
Step 1	The system scans the SHIP_PARTICULARS_SUB subscription tables for the information of Ship Particulars subscriptions. At least one subscription is found.
Step 2	For each subscription, the system retrieves from the data store, the Ship Particulars that are associated with the flag and time criteria of the subscription.
Step 3	The system forms a reply (SSN2ES_ShipParticulars_Ann) for the given request. The system logs and sends the reply to the data requestor.
Alternative Use Case Description	No subscriptions were found
Step 4.1	The system scans the SHIP_PARTICULARS_SUB subscription tables for the information of Ship Particulars subscriptions. No subscriptions found, or no subscriptions include the current timestamp in the subscription period.
Step 4.2	The processing stops.
Input(s)	-
Output(s)	SSN2ES_ShipParticulars_Ann
Timer(s)	Every 60 seconds
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	-

3.2.3.12 UC-SSN-IRREQ-35: Process Incident Report Request (XML)

Use Case Req ID	UC-SSN-IRREQ-35	
Use Case Name	Process Incident Report Request (XML)	
Purpose	Covers the functionality related to the system's actions upon receiving a request for Incident Report notification information through the System Interface	
Subsystem	SSN Core, SSN XML Protocol, SSN SEG WS	
Primary Actor(s)	NCA/SEG Application	
Precondition(s)	A request for information has been received by the system.	
Postcondition(s)	The system has processed the request and the response message is provided to the NCA/SEG Application.	
Trigger(s)	A request for information has been sent to SSN in XML format	
Use Case Description	Primary Workflow	

Step 1	<p>The system performs the authorization checks defined in UC-SSN-MSG-01 for the Actor submitting the request.</p> <p>The system identifies access rights restrictions on Incident Report requests, based on the following:</p> <p>On the ALERT_[type]_REQUESTOR task (corresponding to the requested Incident Report Types), or any ALERT_[type]_REQUESTOR task (if no specific Incident Report Type is requested);</p> <p>And source restrictions on the data provider.</p>
Step 2	<p>The system identifies one of the following requests (TypeOfQuery):</p> <ul style="list-style-type: none"> • All IR of Selected Ship <ul style="list-style-type: none"> ○ Required: ShipIdentificationCriteria (IMO, MMSI or IRNumber_FishingVessel) ○ Optional: ShipIdentificationCriteria (CallSign, ShipName and Flag) - TimePeriodCriteria (StartDateTime, EndDateTime). If StartDateTime and EndDateTime are not quoted the query provides data related to the last 6 months. • Specific TypeIncident Report of selected ship <ul style="list-style-type: none"> ○ Required: ShipIdentificationCriteria (IMO, MMSI or IRNumber_FishingVessel) – IncidentSelectionCriteria (IncidentSelectionType). ○ Optional: ShipIdentificationCriteria (CallSign, ShipName and Flag) – TimePeriodCriteria (StartDateTime, EndDateTime). If StartDateTime and EndDateTime are not quoted the query provides data related to the last 6 months. • IRs for specific port <ul style="list-style-type: none"> ○ Required: GeographicCriteria (PortOfDepartureQuotedInIR or "PortOfDestinationQuotedInIR) ○ Optional: TimePeriodCriteria (StartDateTime, EndDateTime). If StartDateTime and EndDateTime are not quoted the query provides data related to the last 6 months. • Get specific IR <ul style="list-style-type: none"> ○ Required: IncidentSelectionCriteria (IncidentID)
Step 3	<p>The system verifies that the search criteria are correct based on the XMLRG business rules for incident report request.</p> <p>The system searches the SSN database for Incident Reportnotifications based on the provided request criteria.</p>
Step 4	<p>The system has identified a list of Incident Report notifications which complies with the search criteria, the requestor's access rights and the source restrictions (applied to the provider of each Incident Report notification). Only the Incident Report notifications which comply with the source restrictions are provided.</p> <p>The system identifies access rights restrictions on Incident Report notifications, based on the following:</p>

	<ul style="list-style-type: none"> On the ALERT_SITREP_REQUESTOR task if incident type="SITREP"; On the ALERT_POLREP_REQUESTOR task if incident type="POLREP"; On the ALERT_WASTE_REQUESTOR task if incident type="Waste"; On the ALERT_LFC_REQUESTOR task if incident type="LostFoundContainers"; On the ALERT_OTHERS_REQUESTOR task if incident type="Others"; On the ALERT_FAILED_REQUESTOR task if incident type="FailedNotification"; On the ALERT_BANNED_REQUESTOR task if incident type="BannedShip"; On the ALERT_VTS_REQUESTOR task if incident type="VTSRulesInfringement"; On the ALERT_PILOT_REQUESTOR task if incident type="PilotOrPortReport"; On the ALERT_INSURANCE_REQUESTOR task if incident type="InsuranceFailure"; <p>Source restrictions are applied to the provider of each Incident Report notification as follows:</p> <ul style="list-style-type: none"> In case the pertinent ALERT_[Type]_REQUESTOR permission has "Countries list" source restriction, the notifier's country (country of the provider of Incident Report information) shall be within the countries assigned to the applicable regional agreement(s). In case the pertinent ALERT_[Type]_REQUESTOR permission has "User's country" source restriction, the notifier's country (country of the provider of Incident Report information) shall be equal to the requestor's country. <p>The system fetches the Incident Report notifications from the data store.</p>
Step 5	The system constructs a response message.
Step 6	The system logs and sends the reply to the data requestor.
Alternative Use Case Description	No matching Notifications found for the given request criteria
Step 1.1	The system creates a reply which indicates that no notifications were found.
Step 1.2	The system logs the response message and sends it to the data requestor.
Alternative Use Case Description	The Actor sending the request does not comply to authorization requirements
Step 2.1	The system creates a reply which indicates that the Actor is not authorized to send the request.
Step 2.2	The system logs the response message and sends it to the data requestor.
Input(s)	Incident Report Request
Output(s)	Incident Report Response

Timer(s)	Response Timer
Business Process(es) Reference	-
Associated Use Case(s)	Included UC: UC-SSN-MSG-01: Handle Incoming Message
Special Requirements	

3.2.3.13 UC-SSN-ESRES-36: REST enrichment query Service

Use Case Req ID	UC-SSN-ESRES-36
Use Case Name	REST enrichment query Service
Purpose	Covers the functionality related to the provision of the enrichment ("voyages" - "incidents" - "MRS" - "exemptions") information.
Subsystem	SSN Core, SSN SEG WS
Primary Actor(s)	SSN System User
Precondition(s)	Actor is authorised for accessing to the system. "HTTP Basic Authentication" is used for Actor authorisation.
Postcondition(s)	SSN System provides the requested information.
Trigger(s)	Actor requests the enrichment information.
Use Case Description	Primary Scenario
Step 1	The Actor submits an HTTP Get on REST Enrichment query Service on /ssn-seg-ws/enrichment/ for paths "voyages" - "incidents" - "MRS" - "exemptions".
Step 2	The system identifies access rights restrictions <ul style="list-style-type: none"> for paths "voyages" - "incidents" - "MRS" based on ENRICHMENT_REQUESTOR task; for path "exemptions" based on EXEMPTIONS_REQUESTOR task.
Step 3	The system returns the information in XML format (according to seg_4_0.xsd) exists in EIS Database (VESSELS_ENRICHMENT and EXEMPTIONS table). More specifically: <ul style="list-style-type: none"> VesselIdentification information is returned for any selected path; VoyageInformation data is returned for path "voyages"; IncidentIdentification information is returned for path "incidents"; MRSInformation and MRSVoyageInformation data is returned for path "MRS"; ExemptionDetails information (as listed in UC-SSN-EXC-11-Step 4) is returned for path "exemptions".
Alternative Use Case Description	No data found
Step 3.2	The HTTP Header Status 204 No Content is used to inform the Actor for no information exists for the selected case (path; i.e. "voyages" - "incidents" - "MRS" - "exemptions").

Input(s)	Search criteria.
Output(s)	The enrichment information in XML format (according to seg_4_0.xsd).
Timer(s)	<ul style="list-style-type: none"> - An Oracle job exists named UPDATE_VESSEL_ENRICHMENT_JOB (executed i.e. every 2 minutes) that <ul style="list-style-type: none"> - scans the EIS notifications and updates the EIS Database (VESSELS_ENRICHMENT table) with the most recent information for voyages, incidents and MRS - cleanup the EIS Database (VESSELS_ENRICHMENT table) from old/outdated voyages, incidents and MRS
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	-

3.2.3.14 UC-SSN-EXREQ-37: Process Exemption Request

Use Case Req ID	UC-SSN-EXREQ-37	
Use Case Name	Process Exemption Request	
Purpose	Covers the functionality related to the systems' actions upon receiving a request for exemption information from a National Application.	
Subsystem	SSN Core / SSN XML Protocol	
Primary Actor(s)	SSN System User	
Precondition(s)	A request for exemption information has been received by the system.	
Postcondition(s)	The system has processed the request and returned a response to data requestor.	
Trigger(s)	An exemption request message has been sent to SSN in XML format.	
Use Case Description	Primary Workflow	
Step 1	A System user' applilication has sent a valid request for exemption information to the System.	
Step 2	<p>The system identifies the access rights restrictions on exemption requests, based on the following:</p> <p>On the EXEMPTIONS_REQUESTOR task, location restrictions on granting country and exempted ports, as well as source restrictions on the data provider..</p>	
Step 3	<p>The System identifies one of the following requests (queries):</p> <ul style="list-style-type: none"> • All active exemptions for a ship; • All active exemptions issued by a country; • All active exemptions applied in a port; • All active exemptions of a specific type; 	

	Or a combination of the requests (i.e. country, type of exemption, etc.) or none (all active exemptions).
Step 4	<p>The system searches the SSN database for relevant exemptions based on the combination of the request criteria.</p> <ul style="list-style-type: none"> In case the EXEMPTIONS_REQUESTOR permission has any location restriction, the system searches for exemptions with granting country being SSN participant (a specific type, i.e. CountryTypeId: 10, is created in CCD for SSN participant countries; SSN-EIS application saves this information as a Boolean indicator in COUNTRIES.IS_SSN column). In case the EXEMPTIONS_REQUESTOR permission has "Specific locations" location restriction, the system searches for exemptions having at least one exempted port contained in the locodes assigned to the applicable duty/ies for the requestor's organisation; In case of "User's country" location restriction, the system searches for exemptions with granting country being equal to the requestor's country. In case of "Countries list" location restriction, the system searches for exemptions with granting country being contained in the countries assigned to the applicable regional agreement(s).
Step 5	<p>The system has identified a list of exemptions which complies with the search criteria as well as requestor's access rights, location restrictions (as described in Step 4) and source restrictions (applied to the provider of each exemption). Only the exemptions which comply with the location and source restrictions are provided.</p> <p>Source restrictions are applied to the provider of exemption information as follows:</p> <ul style="list-style-type: none"> In case the EXEMPTIONS_REQUESTOR permission has "Countries list" source restriction, the notifier's country (country of the provider of exemption information) shall be within the countries assigned to the applicable regional agreement(s). In case the EXEMPTIONS_REQUESTOR permission has "User's country" source restriction, the notifier's country (country of the provider of exemption information) shall be equal to the requestor's country. <p>The system fetches the exemption data from the data store.</p>
Step 6	The system constructs a response message.
Step 7	The system logs and sends the response to the data requestor. Exemption details in response include the information listed in UC-SSN-EXC-11 (Step 4).
Alternative Use Case Description	No relevant data found.
Step 5.1.1	The system identifies that no exemption satisfies the request criteria. The system generates and sends a negative response to the data requestor.
Step 5.1.2	The system generates anExemptionRes with StatusCode="NotFound".
Alternative Use Case	Access Control Check fails

Description	
Step 5.2.1	If the Access Control fails to apply the restrictions of Step 2 and 3, The system identifies that the user is not granted the permission to submit the request (is not granted the relative TASK; see step 2) or the geographical access rights restrictions on exemptionresults does not permit submitting a request for the specified request criteria.
Step 5.2.2	The system generates a receipt indicating the error and StatusCode="AccessDenied".
Input(s)	Request specific exemption data.
Output(s)	Exemption Details are sent to the Data Requester (NCA Application). The system shall respond to a requesting NCA Application in XML format.
Timer(s)	
Business Process(es) Reference	
Associated Use Case(s)	Included UC: UC-SSN-MSG-01: Handle Incoming Message
Special Requirements	

3.2.3.15 UC-SSN-IRREQ-38: Process Incident Report Request (Web)

Use Case Req ID	UC-SSN-IRREQ-38
Use Case Name	Process Incident Report Request (Web)
Purpose	The current Use Case describes the functionality related to the search for Incident Information from the Web interface.
Subsystem	SSN Web application
Primary Actor(s)	SSN Human User
Precondition(s)	The Actor has been authenticated in the system. The Actor is granted any of the ALERT_[type]_REQUESTOR permissions any ALERT_[type]_REQUESTOR task.
Postcondition(s)	The request has been treated and the answer provided to the SSN user
Trigger(s)	-
Use Case Description	The Actor wants Incident Report Notification details.
Step 1	<p>The Actor navigates to the web page for requesting Incident Report Notification details (Incidents Information) for the next options</p> <ul style="list-style-type: none"> • All IR of Selected Ship <ul style="list-style-type: none"> ○ Required: Vessel IdentificationCriteria (IMO, MMSI, Callsign, Name, IRNumber or Flag) ○ Optional: TimePeriodCriteria (StartDateTime, EndDateTime). • Specific Type of IR for selected ship. Values are limited based on the ALERT_[type]_REQUESTOR tasks assigned to the actor.

	<ul style="list-style-type: none"> ○ Required: Vessel IdentificationCriteria (IMO, MMSI, Callsign, Name, IRNumber or Flag). ○ Required: Incident Type. ○ Optional: TimePeriodCriteria (StartDateTime, EndDateTime). • IRs for specific port <ul style="list-style-type: none"> ○ Required: GeographicCriteria (PortOfDepartureQuotedInIR or PortOfDestinationQuotedInIR) ○ Optional: TimePeriodCriteria (StartDateTime, EndDateTime). • Get specific IR <ul style="list-style-type: none"> ○ Required: IncidentId • Distributed Incidents <ul style="list-style-type: none"> ○ Optional: IncidentId ○ Optional: Status (new/old/both) ○ Optional: Incident Type.
Step 2	The system prompts the Actor to fill in the criteria to be used for requesting Incident Report Notification details.
Step 4	<p>The system displays a list of Notifications matching the criteria, the requestor's access rights and the source restrictions (applied to the provider of each Incident Report notification). Only the Incident Report notifications which comply with the source restrictions are provided.</p> <p>The system identifies the access rights on Incident Report notifications, based on the following:</p> <ul style="list-style-type: none"> • On the ALERT_SITREP_REQUESTOR task if incident type="SITREP"; • On the ALERT_POLREP_REQUESTOR task if incident type="POLREP"; • On the ALERT_WASTE_REQUESTOR task if incident type="Waste"; • On the ALERT_LFC_REQUESTOR task if incident type="LostFoundContainers"; • On the ALERT_OTHERS_REQUESTOR task if incident type="Others"; • On the ALERT_FAILED_REQUESTOR task if incident type="FailedNotification"; • On the ALERT_BANNED_REQUESTOR task if incident type="BannedShip"; • On the ALERT_VTS_REQUESTOR task if incident type="VTSRulesInfringement"; • On the ALERT_PILOT_REQUESTOR task if incident type="PilotOrPortReport"; <p>On the ALERT_INSURANCE_REQUESTOR task if incident type="InsuranceFailure";</p> <p>Source restrictions are applied to the provider of each Incident Report notification as follows:</p>

	<ul style="list-style-type: none"> ○ In case the relevant ALERT_[type]_REQUESTOR permission has "Countries list" source restriction, the notifier's country (country of the provider of Incident Report information) shall be within the countries assigned to the applicable regional agreement(s). ○ In case the relevant ALERT_[type]_REQUESTOR permission has "User's country" source restriction, the notifier's country (country of the provider of Incident Report information) shall be equal to the requestor's country.
Step 5	The Actor selects a Notification from the list.
Step 6	The system logs the request for the selected Incident Report Notification details and displays Notification information.
Input(s)	Request specific data.
Output(s)	The system logs the request for the specific Incident Report Notification and Notification Details are displayed to the Actor.
Timer(s)	
Business Process(es) Reference	
Associated Use Case(s)	
Special Requirements	

3.2.4 Data Receive

This system package includes the services required for the SSN Resources to process announcements of updates regarding resources (countries, locations, organisations, users) provided by CCD, CLD, COD and IdM via the XML/SOAP interface. More specifically, this system package consists of the following use-cases:

3.2.4.1 UC-SSN-CNT-ANN: Process Country Announcement

Use Case Req ID	UC-SSN-CNT-ANN	
Use Case Name	Process Country Announcement	
Purpose	Covers the functionality related to the system's actions when notified of the creation, update, deletion of a country from CCD Announcement Web Service.	
Subsystem	Subscriber Web Service (ssn-subscriber-ws)	
Primary Actor(s)	External system (CCD Announcement Web Service)	
Precondition(s)	SSN application has signed for the CCD subscription service to be informed about modifications in the country base registry data.	
Postcondition(s)	System returns a SOAP response informing about the acceptance of the change notification. System persists "updated Country information" in SSN database.	
Trigger(s)	A notification of the creation, update, deletion of a country from CCD is received.	

	Note: the country announcement message does not contain the country information that is to be persisted in SSN database, rather than a notification that a change has occurred for the specified country in CCD.
Use Case Description	Primary Workflow
Step 1	The system determines whether the announcement message consists of a notification of creation, update or deletion of a country.
Step 2	The system requests the actual country information, in case of a notification of creation or update, for the country specified by the alpha2Code defined in the country announcement message by calling getCountry method of Country Information Web Service.
Step 3	The system requests the associated MIDs for the country specified by the alpha2Code defined in the country announcement message by calling getCountryMIDList method of Country Information Web Service.
Step 4	The system resolves retrieved country information in SSN database.
Step 4. a	No country exists in SSN database, with the same alpha2Code as the one defined in the "notify" message. Country information and associated MIDs retrieved from CCD will get: <ul style="list-style-type: none"> persisted;
Step 4. b	Country exists in SSN database with the same alpha2Code as the one defined in the "notify" message, resolved country will get: <ul style="list-style-type: none"> updated with the country information and associated MIDs retrieved from CCD, if announcement status is "U" or "C" (Announcement of the ES about a creation/update in a subscribed Country); deleted, if announcement status is "D" (Announcement of the ES about a deletion of a subscribed Country);
Step 5	The system persists the modified country data in SSN database and returns a SOAP response with payload a "notifyResponse" with ResponseType=0 ("OK") informing that the announcement was successfully processed.
Alternative Use Case Description	Announcement of the ES about a country deletion - No country exists in SSN database, with the same alpha2Code as the one defined in the "notify" message.
Step 4. a.1	Country information is ignored. A detail message of cause is logged at the INFO level, i.e. "Unable to delete country. Country does not exist: alpha2Code".
Step 4. a.2	The system returns a SOAP response with payload a "notifyResponse" with ResponseType=70 ("NOK_UNABLE_TO_UPDATE_CODE") and ResponseMsg="Unable to update."
Input(s)	"Notify" WSDL message.
Output(s)	"Updated Country information" persisted in SSN database: <ul style="list-style-type: none"> Country two-letter code, Country name, Country category,

	<ul style="list-style-type: none"> • EU member (a specific type, i.e. CCD TYPE_ID: 0, is created in CCD for EU Member countries; SSN-EIS application saves this information as a Boolean indicator in COUNTRIES.IS_EUMEMBER column), • SSN participant (a specific type, i.e. CountryTypeId: 10, is created in CCD for SSN participant countries; SSN-EIS application saves this information as a Boolean indicator in COUNTRIES.IS_SSN column), • List of associated country codes (case where the country's category is Regional Agreement), • List of MIDs.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	
Special Requirements	-

3.2.4.2 UC-SSN-LOC-ANN: Process Location Announcement

Use Case Req ID	UC-SSN-LOC-ANN	
Use Case Name	Process Location Announcement	
Purpose	Covers the functionality related to the system's actions when notified of changes to locations from CLD Location Web Service.	
Subsystem	Subscriber Web Service (ssn-subscriber-ws)	
Primary Actor(s)	External system (CLD Location Web Service)	
Precondition(s)	SSN application is registered in CLD database as a subscriber for location updates.	
Postcondition(s)	System persists "updated Locations information" in SSN database.	
Trigger(s)	A location announcement message from CLD, containing created or updated location entries associated with the specific (SSN application) subscription options, is received.	
Use Case Description	Primary Workflow	
Step 1	The system resolves the country information in SSN database for each of the created or updated CLD location entries.	
Step 2	The system resolves each of the created or updated CLD location entries in SSN database.	
Step 2. a	No location exists in SSN database with the same Digit5LoCode as the one defined in the CLD location entry. The latest location details provided by CLD will get: <ul style="list-style-type: none"> • ignored, if the location entry is not active for SSN application; • persisted, if the location entry is active for SSN application; 	

Step 2. b	<p>A location exists in SSN database with the same Digit5LoCode as the one defined in the CLD location entry. The resolved SSN location will:</p> <ul style="list-style-type: none"> • remain unchanged, in case of outdated Location information from CLD (i.e. the timestamp when the resolved SSN location was last updated is after the "LastUpdatedOn" timestamp of the CLD location entry); • get updated with the latest details provided by CLD, if the CLD location entry is active for the SSN application; • get deactivated, if the CLD location entry is not active for the SSN application;
Step 3	The system persists the "updated Locations information" in SSN database.
Alternative Use Case Description	Unresolved country
Step 1.1	No country information is found in SSN database for the CLD location entry.
Step 1.2	The location details provided by CLD for the location entry specifying the unresolved country are not persisted (ignored). An UnknownCountryException with the specified cause and a detail message of cause is logged at the ERROR level.
Input(s)	"AnnouncementLocation" WSDL message.
Output(s)	<p>"Updated Locations information" persisted in SSN database:</p> <ul style="list-style-type: none"> • Location code, • Location name with diacritics, • Location name without diacritics, • Country code, • Comments, • Type (UNECE or SSN specific), • Position (Longitude/Latitude, • Status (Active/Inactive), • List of alternative Location names
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	
Special Requirements	-

3.2.4.3 UC-SSN-ORG-ANN: Process Organisation Announcement

Use Case Req ID	UC-SSN-ORG-ANN	
Use Case Name	Process Organisation Announcement	
Purpose	Covers the functionality related to the system's actions when notified of changes to organisations from COD Organisation Web Service.	
Subsystem	Subscriber Web Service (ssn-subscriber-ws)	

Primary Actor(s)	External system (COD Organisation Web Service)
Precondition(s)	SSN application is registered in COD database as a subscriber for organisation updates.
Postcondition(s)	System persists the "updated Organisations information" in SSN database.
Trigger(s)	An organisation announcement message from COD, containing created or updated organisation entries associated with the specific (SSN application) subscription options, is received.
Use Case Description	Primary Workflow
Step 1	The system resolves the country information in SSN database for each of the created or updated COD organisation entries.
Step 2	The system resolves each of the created or updated organisation entries in SSN database.
Step 2. a	No organisation exists in SSN database with the same OrganisationID as the one defined in the COD organisation entry. The latest organisation details provided by COD will get: <ul style="list-style-type: none"> • ignored, if the COD organisation entry is not active; • persisted, if the COD organisation entry is active;
Step 2. b	An organisation exists in SSN database with the same OrganisationID as the one defined in the COD organisation entry. The resolved SSN organisation will: <ul style="list-style-type: none"> • remain unchanged, in case of outdated Organisation information from COD (i.e. the timestamp the resolved SSN organisation when it was last updated is after the "LastUpdatedOn" timestamp of the COD organisation entry); • get updated with the latest details provided by COD, if the COD organisation entry is active; • get deactivated, if the COD organisation entry is not active;
Step 3	The system persists the "updated Organisations information" in SSN database.
Alternative Use Case Description	Unresolved country
Step 1.1	No country information is found in SSN database for the COD organisation entry.
Step 1.2	The organisation details provided by COD for the organisation entry specifying the unresolved country are not persisted (ignored). An UnknownCountryException with the specified cause and a detail message of cause is logged at the ERROR level.
Input(s)	"AnnouncementOrganisation" WSDL message.
Output(s)	"Updated Organisations information" persisted in SSN database: <ul style="list-style-type: none"> • Organisation ID, • Organisation name, • Business phone (from COD element Contact.ContactNumbers) • Fax (from COD element Contact.ContactNumbers)

	<ul style="list-style-type: none"> • E-mail (from COD element Contact.ContactNumbers) • First name (from COD element Contact.Person.GivenName), • Last name (from COD element Contact.Person.FamilyName), • Location code • Country, • Status (Active/Inactive), • List of duty codes, • For each duty code, the list of areas codes (LOCODES).
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	-

3.2.4.4 UC-SSN-USR-SAV-ANN: Save User Announcement

Use Case Req ID	UC-USR-SAV-ANN	
Use Case Name	Save User Announcement	
Purpose	Covers the functionality related to the system's actions when notified of the creation or update of a user from IdM Provisioning Service.	
Subsystem	IdM Web Service (ssn-idm-ws)	
Primary Actor(s)	External system (OIM/IdM Provisioning Service)	
Precondition(s)	"SafeSeaNet" application must be associated in OIM to the user being provisioned.	
Postcondition(s)	<p>The system returns a SOAP response informing about the result of the "Save User" process.</p> <p>The system persists the "updated User information" in SSN database.</p>	
Trigger(s)	A new user is created, or changes are made to an existing user in OIM. OIM has effectively sent information on the new user (or the changes made to an existing user) by invoking the "createUser" or "updateUser" WSDL operation of the IdM Web Service.	
Use Case Description	Primary Workflow	
Step 1	<p>The system performs validation checks on the user account details specified in "CreateUserMessage"/"UpdateUserMessage" received. Validation errors refer to:</p> <ul style="list-style-type: none"> • Invalid COD organisationId (the organisation does not exist); • Invalid CCD country (the country does not exist); • Invalid email (wrongly formatted mail address). <p>The user account details pass the validation checks.</p>	
Step 2	The system resolves the reported user in SSN database.	
Step 2. a	No user exists in SSN database with the same "userId" as the one defined in the "CreateUserMessage"/"UpdateUserMessage".	

	The system persists the user account details provided by IdM and assigns WEB interface to the new user, only if user is "Human (userType parameter coming from IdM)".
Step 2. b	The user exists in SSN database with the same "userId" as the one defined in the "CreateUserMessage"/"UpdateUserMessage". The system updates the resolved SSN user with the user account details provided by IdM.
Step 3	The system returns a SOAP response with payload an "IDMResponse" with StatusCode=" USER_SAVE_SUCCESSFUL".
Alternative Use Case Description	Unresolved COD organisation
Step 1.1.1	No COD organisation is found in SSN database for the specified "organisationId".
Step 1.1.2	The system returns a SOAP response with payload an "IDMResponse" with StatusCode="ORG_DOESNOT_EXIST".
Alternative Use Case Description	Unresolved CCD country
Step 1.2.1	No CCD country is found in SSN database for the specified "userCountry".
Step 1.2.2	The system returns a SOAP response with payload an "IDMResponse" with StatusCode = " COUNTRY_DOESNOT_EXIST".
Input(s)	"CreateUserMessage"/" UpdateUserMessage" WSDL message.
Output(s)	"Updated User information" persisted in SSN database: <ul style="list-style-type: none"> • User ID, • Country code, • Organisation ID, • Type: Human (the user account relates to a physical person) or System (the user account relates to a system), • E-mail, • Roles identified by their codes (a user may have none, one or several Roles).
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	-

3.2.4.5 UC-SSN-USR-REV-ANN: Revoke User Announcement

Use Case Req ID	UC-USR-REV-ANN	
Use Case Name	Revoke User Announcement	
Purpose	Covers the functionality related to the system's actions when notified of the revocation of a user from IdM Provisioning Service.	

Subsystem	IdM Web Service (ssn-idm-ws)
Primary Actor(s)	External system (OIM/IdM Provisioning Service)
Precondition(s)	"SafeSeaNet" application must be associated in OIM to the user being provisioned.
Postcondition(s)	The system returns a SOAP response informing about the result of the "Revoke User" process. System deletes the revoked user from SSN database.
Trigger(s)	User is deleted or loses access to "SafeSeaNet" application using OIM. OIM has effectively sent information on the revoked user by invoking "revokeUser" WSDL operation of IdM Web Service.
Use Case Description	Primary Workflow
Step 1	The system resolves the revoked user in SSN database based on the reported "userId".
Step 2	The system deletes the resolved user from SSN database and returns a SOAP response with payload an "IDMResponse" with StatusCode="USER_REVOKE_SUCCESSFUL".
Alternative Use Case Description	Unresolved revoked user
Step 1.1	No IdM user is found in SSN database for the specified "userId".
Step 1.2	The system returns a SOAP response with payload an "IDMResponse" with StatusCode="USER_DOESNOT_EXIST".
Input(s)	"RevokeUserMessage" WSDL message.
Output(s)	Revoked user deleted from SSN database.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	-

3.2.4.6 UC-SSN-USR-ACT-ANN: Enable/Disable User Announcement

Use Case Req ID	UC-USR-ACT-ANN	
Use Case Name	Enable/Disable User Announcement	
Purpose	Covers the functionality related to the system's actions when notified of the user's status (enabled/disabled) from IdM Provisioning Service.	
Subsystem	IdM Web Service (ssn-idm-ws)	
Primary Actor(s)	External system (OIM/IdM Provisioning Service)	
Precondition(s)	"SafeSeaNet" application must be associated in OIM to the user being provisioned.	

Postcondition(s)	The system returns a SOAP response informing about the result of the "Enable/Disable User" process. The system updates user's status in SSN database.
Trigger(s)	A Disable/Enable command has been issued to disable user access to "SafeSeaNet" application in OIM. OIM has effectively sent information on the user status by invoking "enableDisableUser" WSDL operation of IdM Web Service.
Use Case Description	Primary Workflow
Step 1	The system resolves the reported user in SSN database.
Step 2	The system updates the resolved user's status in SSN database, based on the reported "enableUser" attribute, and returns a SOAP response with payload an "IDMResponse" with StatusCode="USER_ENABLE_DISABLE_SUCCESSFUL".
Alternative Use Case Description	Unresolved reported user
Step 1.1	No IdM user is found in SSN database for the "userId" specified in the "EnableDisableUserMessage".
Step 1.2	The system returns a SOAP response with payload an "IDMResponse" with StatusCode="USER_DOESNOT_EXIST".
Input(s)	"EnableDisableUserMessage" WSDL message.
Output(s)	Reported user's status updated in SSN database.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	-

3.2.4.7 UC-SSN-ROL-ADD-ANN: Assign Role Announcement

Use Case Req ID	UC-ROL-ADD-ANN	
Use Case Name	Assign Role Announcement	
Purpose	Covers the functionality related to the system's actions when notified of a user's role assignment from the IdM Provisioning Service.	
Subsystem	IdM Web Service (ssn-idm-ws)	
Primary Actor(s)	External system (OIM/IdM Provisioning Service)	
Precondition(s)	"SafeSeaNet" application must be associated in OIM to the user being provisioned.	
Postcondition(s)	The system returns a SOAP response informing about the result of the "Assign Role" process. The system adds the user's role in SSN database.	

Trigger(s)	A "SafeSeaNet" application role has been assigned to user using OIM. OIM has effectively sent information on user's role assignment by invoking "assignRole" WSDL operation of IdM Web Service.
Use Case Description	Primary Workflow
Step 1	The system resolves the reported user in SSN database.
Step 2	The system checks whether the reported role is already assigned to the resolved user.
Step 3	The system updates resolved the user's roles in SSN database by adding the reported role and returns a SOAP response with payload an "IDMResponse" with StatusCode=" USER_ADD_ROLE_SUCCESSFUL".
Alternative Use Case Description	Unresolved reported user
Step 1.1	No IdM user is found in SSN database for the "userId" specified in the "AssignRoleMessage".
Step 1.2	The system returns a SOAP response with payload an "IDMResponse" with StatusCode=" USER_DOESNOT_EXIST".
Alternative Use Case Description	Role assigned
Step 2.1	Reported role is already assigned to the resolved user.
Step 2.2	The system returns a SOAP response with payload an "IDMResponse" with StatusCode=" DUPLICATE_ROLE".
Input(s)	"AssignRoleMessage" WSDL message.
Output(s)	The reported user's role added in SSN database.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	-

3.2.4.8 UC-SSN-ROL-REV-ANN: Unassign Role Announcement

Use Case Req ID	UC-ROL-REV-ANN	
Use Case Name	Unassign Role Announcement	
Purpose	Covers the functionality related to the system's actions when notified of a user's role revocation from the IdM Provisioning Service.	
Subsystem	IdM Web Service (ssn-idm-ws)	
Primary Actor(s)	External system (OIM/IdM Provisioning Service)	
Precondition(s)	"SafeSeaNet" application must be associated in OIM to the user being provisioned.	
Postcondition(s)	The system returns a SOAP response informing about the result of the "Unassign Role" process.	

	The system deletes user's role from SSN database.
Trigger(s)	A "SafeSeaNet" application role has been revoked from user using OIM. OIM has effectively sent information on user's role revocation by invoking "unassignRole" WSDL operation of IdM Web Service.
Use Case Description	Primary Workflow
Step 1	The system resolves the reported user in SSN database.
Step 2	The system checks whether the revoked role is not assigned to the resolved user.
Step 3	The system updates the resolved user's roles in SSN database by deleting the reported role and returns a SOAP response with payload an "IDMResponse" with StatusCode="USER_REMOVE_ROLE_SUCCESSFUL".
Alternative Use Case Description	Unresolved reported user
Step 1.1	No IdM user is found in SSN database for the "userId" specified in the "UnassignRoleMessage".
Step 1.2	The system returns a SOAP response with payload an "IDMResponse" with StatusCode="USER_DOESNOT_EXIST".
Alternative Use Case Description	Role not assigned
Step 2.1	Revoked role is not assigned to the resolved user.
Step 2.2	The system returns a SOAP response with payload an "IDMResponse" with StatusCode="USER_DOESNOT_HAVE_ROLE".
Input(s)	"UnassignRoleMessage" WSDL message.
Output(s)	Reported user's role deleted from SSN database.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	-

3.2.5 Monitoring IncidentReport

This system package includes the services required to provide the end users with a textual interface to consult the IR notifications distribution status. This system package consists of the following use-case:

3.2.5.1 UC-SSN-IRM-31: Monitor IR notification status

Use Case Req ID	UC-SSN-IRM-31	
Use Case Name	Monitor IR notification status	

Purpose	Covers the functionality related to the consultation of the distribution status of Incident Report notifications and Incident Report feedback notifications (MS2SSN_IncidentDetail_Tx) in the Web interface.
Subsystem	SSN Core, SSN Send Notifications console
Primary Actor(s)	SSN Human User
Precondition(s)	Actor is authorised for accessing to the system. The Actor is granted any of the ALERT_[type]_REQUESTOR permissions
Postcondition(s)	System provides the requested information.
Trigger(s)	Actor requests the IR notification status.
Use Case Description	Primary Scenario
Step 1	The Actor navigates to SSN TI > Send notification > Incident Reports > Check distribution web console.
Step 2	<p>The Actor enters the search criteria:</p> <ul style="list-style-type: none"> – Incident ID (text search). – Incident type (single choice, default value: any). It applies to IR notifications only. If an incident type is selected, then only IR notifications are provided. Values are limited based on the ALERT_[type]_REQUESTOR tasks assigned to the actor. – Recipient country (single choice, default value: any). – Date/Time From & To of SentAt (default values: from [now-3 months] to [now]).
Step 3	<p>The system returns a list of IR notifications distributed by the user's country, matching the criteria, the requestor's access rights and the source restrictions (applied to the provider of each IR notification). Only the IR notifications which comply to the source restrictions are provided.</p> <p>The system identifies the Actor's access rights on IR notifications based on the following:</p> <ul style="list-style-type: none"> • On the ALERT_SITREP_REQUESTOR task if incident type="SITREP"; • On the ALERT_POLREP_NOTIFIER task if incident type="POLREP"; • On the ALERT_WASTE_NOTIFIER task if incident type="Waste"; • On the ALERT_LFC_NOTIFIER task if incident type="LostFoundContainers"; • On the ALERT_OTHERS_NOTIFIER task if incident type="Others"; • On the ALERT_FAILED_NOTIFIER task if incident type="FailedNotification"; • On the ALERT_BANNED_NOTIFIER task if incident type="BannedShip"; • On the ALERT_VTS_NOTIFIER task if incident type="VTSRulesInfringement"; • On the ALERT_PILOT_NOTIFIER task if incident type="PilotOrPortReport";

	<ul style="list-style-type: none"> On the ALERT_INSURANCE_NOTIFIER task if incident type="InsuranceFailure"; <p>Source restrictions are applied to the provider of each IR notification as follows:</p> <ul style="list-style-type: none"> In case the pertinent ALERT_[Type]_REQUESTOR permission has a "Countries list" source restriction, the notifier's country (country of the provider of IR information) shall be within the countries assigned to the applicable regional agreement(s). In case the pertinent ALERT_[Type]_REQUESTOR permission has a "User's country" source restriction, the notifier's country (country of the provider of IR information) shall be equal to the requestor's country. <p>Information that will be displayed:</p> <ul style="list-style-type: none"> Type of message (IR notification / IR Feedback) Incident Id Incident Type Sent At (default sort criterion, most recent on top) Recipient countries (2-letter codes separated by commas) Link to "Distribution status"
Step 4	The Actor click on the "Distribution status" link and the application displays the "Distribution Acknowledgment status of Incident Report" and in addition general information regarding the Incident report (Incident ID, Incident Type, Sent At).
Step 4	The Actor select to export the results in a predefined file format: XLS, XML, CSV.
Alternative Use Case Description	No data found
Step 2.2	No IR notifications satisfy the serahc criteria.
Step 2.3	The system prompts the Actor with a message that no data were found.
Input(s)	Search criteria.
Output(s)	List of IR notification with distribution status.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	-

3.2.6 MRS Management

This system package includes the services required to provide the end users with a textual interface to create, search and update MRS. This system package consists of the following use-cases:

3.2.6.1 UC-SSN-MRSMNG-41: Create MRS

Use Case Req ID	UC-SSN-MRSMNG-41	
Use Case Name	Create MRS	
Purpose	Covers the functionality related to the create an MRS in the Web interface.	
Subsystem	SSN Application Management Console	
Primary Actor(s)	SSN Human User	
Precondition(s)	Actor is authorised for accessing to the system and for managing MRS (granted MRS_MANAGER permission).	
Postcondition(s)	System provides the requested information.	
Trigger(s)	Actor creates a new MRS.	
Use Case Description	Primary Scenario	
Step 1	The Actor navigates to SSN Application Management > MRS Management > Create MRS console.	
Step 2	The Actor enters the MRS identification and selects the countries assigned to the MRS. Once selected, the countries are listed below the MRS Identification field.	
Step 3	The user clicks on the "Next" button.	
Step 4	If there is a validation error, the user is prompted with the validation errors on the top of the page; he/she can alter the values and click again on "Next". If no errors the system displays the confirmation page.	
Step 4	The Actor clicks on the "Submit" button to save the new MRS.	
Input(s)	MRS identification and countries.	
Output(s)	MRS defined in the database.	
Timer(s)	-	
Business Process(es) Reference	-	
Associated Use Case(s)	-	
Special Requirements	-	

3.2.6.2 UC-SSN-MRSMNG-42: Search/Update MRS

Use Case Req ID	UC-SSN-MRSMNG-42	
Use Case Name	Search/Update MRS	
Purpose	Covers the functionality related to the search/update an MRS in the Web interface.	
Subsystem	SSN Application Management Console	
Primary Actor(s)	SSN Human User	

Precondition(s)	Actor is authorised for accessing to the system and for managing MRS (granted MRS_MANAGER permission).
Postcondition(s)	System provides the requested information.
Trigger(s)	Actor searches/updates an MRS.
Use Case Description	Primary Scenario
Step 1	The Actor navigates to SSN Application Management > MRS Management > Search/Update MRS console.
Step 2	The Actor enters the search criteria: MRS Identification; Countries that assigned to the MRS. Click on "Search" button.
Step 3	From the list of MRS that satisfy the criteria, click on the MRS Identification to edit.
Step 4	The user may modify the MRS Identification (e.g. correct typos) or modify the set of countries tha may provide MRS information for the selected MRS Identification.
Step 5	The user clicks on the "Next" button.
Step 6	If there is a validation error, the user is prompted with the validation errors on the top of the page; he/she can alter the values and click again on "Next". If no errors the system displays the confirmation page.
Step 7	The Actor clicks on the "Submit" button to save the new MRS.
Alternative Use Case Description	Delete MRS
Step 2.5	The user clicks on the "Delete" button.
Step 2.6a	If the MRS is referenced in an MRS_NOTIFICATION, the system prompts the user that the MRS cannot be deleted.
Step 2.6b	If the MRS is not referenced an MRS_NOTIFICATION, the system prompts the user to confirm the deletion.
Input(s)	Search criteria; MRS Identification.
Output(s)	MRS definition is updated or deleted from the database.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	-

3.2.7 System Interface Management

This system package includes the services required to provide the end users with a textual interface to search for system users created by IdM V2 and configure/define the system interface(s) for system users. This system package consists of the following use-cases:

3.2.7.1 UC-SSN-SI-MNG: Search/Update System User

Use Case Req ID	UC-SSN-SI-MNG	
Use Case Name	Search/Update System User	
Purpose	Covers the functionality related to searching a system user and updating its system interface(s) in the Web interface.	
Subsystem	SSN Application Management Console	
Primary Actor(s)	SSN Human User	
Precondition(s)	<p>The actor is authorised for accessing the system.</p> <p>The actor is granted the permission for managing system users, based on the following:</p> <p>On the USER_MANAGER task and location restrictions on the country of the system user and location of system user's organisation.</p>	
Postcondition(s)	The system updates the system user's interface(s).	
Trigger(s)	The actor searches for a system user.	
Use Case Description	Primary Scenario	
Step 1	The Actor navigates to SSN Application Management > System Interface Management > Search System User.	
Step 2	<p>The Actor enters the search criteria:</p> <ul style="list-style-type: none"> • System User Id; • Country (options are populated based on location restrictions as described in Step 3); • Location Code (options are populated based on location restrictions as described in Step 3); • Status (Active/Inactive) <p>The actor launches the search.</p>	
Step 3	<p>In case the USER_MANAGER permission has no location restriction, the system adds a criterion on the system user's country being SSN participant (a specific type, i.e. CountryTypeId: 10, is created in CCD for SSN participant countries; SSN-EIS application saves this information as a Boolean indicator in COUNTRIES.IS_SSN column).</p> <p>In case of a "Specific locations" location restriction, the system adds a criterion on the location code of system user's organisation being contained in the locodes assigned to the applicable duty/ies of the actor's organisation;</p> <p>In case of a "User's country" location restriction, the system adds a criterion on the system user's country being equal to the actor's country.</p>	

	In case of a "Countries list" location restriction, the system adds a criterion on the system user's country being contained in the countries assigned to the applicable regional agreement(s).
Step 4	From the list of system users that satisfy the criteria, the Actor selects a System User to edit it.
Step 5	The actor may modify the "Interface Information" section (e.g. add/edit/delete system interfaces, define Provider Url, Requestor Url, Certificate Name information). Actor is not allowed to add web interface to a system user. The rest of the system user information (i.e. System User Details, Contact Information, Roles) are managed by IdM and therefore rendered as read-only.
Step 6	The actor submits the changes.
Step 7	If there is a validation error, the actor is prompted with the validation errors; he/she can alter the values and submit the changes again. Validation errors refer to: <ul style="list-style-type: none"> • Missing/Invalid provider url • Missing/Invalid requestor url If there is no error, the system displays the confirmation page.
Step 8	The actor confirms the changes. The system updates the system user's interfaces.
Input(s)	Search criteria;
Output(s)	The system user's interface is updated in SSN database.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	-

3.2.8 CSD Management

This system package includes the services required for the CSD management. This system package consists of the following use-cases:

1. UC-CSD-MNG-01: Create/Edit Ship
2. UC-CSD-MNG-03: Search Ship
3. UC-CSD-MNG-04: Consult Ship
4. UC-CSD-MNG-02 (Updated): Consult history of changes of Ship data
5. UC-CSD-MNG-05: Upload Vessels (EMSA may consider revision of this functionality)
6. UC-CSD-MNG-06: Validate Uploaded Ship Data (EMSA may consider revision of this functionality)
7. UC-CSD-MNG-07 (Updated): Upload MARS ships
8. UC-CSD-MNG-08 (Updated): Validate uploaded MARS ships
9. UC-CSD-MNG-09: Download ship list
10. UC-OSD-MNG-VV-10 (Updated): Verify and validate ship
11. UC-CSD-MNG-11 (Updated): Pending CSD Update
12. UC-CSD-MNG-12 (Updated): CSD Configuration utility
13. UC-CSD-UPL-MARS-13: MARS Data Processing (EMSA may consider revision of this functionality)

14. UC-CSD-SPN-14: Process ShipParticulars Notification (Revised)
15. UC-OSD-PPN-15 (Revised): Process Vessel Attributes reported in EIS notification
16. UC-CSD-SPN-16 (Revised): Vessel Resolution – Ship Particulars Notification
17. UC-OSD-SPN-17 (New): Vessel Resolution - EIS Notification
18. UC-SSN-CSD-18 (Revised): CSD synchronisation with OSD
19. UC-SSN-OSD-19 (Revised) – OSD synchronisation with CSD

3.2.8.1 UC-CSD-MNG-01: Create/Edit Ship

Use Case Req ID	UC-CSD-MNG-01
Use Case Name	Create/Edit Ship
Purpose	Covers the functionality related to the creation/edit of a ship via the web interface.
Subsystem	SSN CSD web console
Primary Actor(s)	Manager User
Precondition(s)	Actor is authorised for accessing to the system and for creating and editing vessel.
Postcondition(s)	The vessel is defined/updated in the data store.
Trigger(s)	The actor selects to create/update a vessel via the webinterface.
Use Case Description	Primary Scenario: [Create new vessel]
Step 1.1	The Actor navigates to Central Ship Database (CSD) > Create Vessel
Step 1.2	The actor defines the vessel attributes: <ul style="list-style-type: none"> - Ship Name - EMSA-R Number - IMO Number - MMSI Number - Inmarsat Call Number - Call Sign - Flag Registry - IR Number (if - Ship Type (LRIT) - Ship Type (PSC) - Ship Type (IHS) - Ship Type (AIS) - Ship Type(Lloyds') - Ship Type (UN) - Keel Laying Date - Contract Date - Service Indicator - Status (from PSC) - Gross Tonnage - Length (Overall) - Max Number of Passengers - Reduced Gross Tonnage - Length between Perpendiculars - Number of crew - Net Tonnage - Keel To Mast Height - Number of RORO Compartments - Breadth (Moulded)

	<ul style="list-style-type: none"> - DeadWeight - TEU - Beam - Draught - Number of Cargo Tanks - Number of Holds - ISM Company IMO Number - Registered Owner – The Owner Organisation ID; Note: The Country and Address information is fetched and displayed here in case that the Owner Organisation ID exists on COD. - Registered Owner IMO Number - Registered Manager (ISM company) – The Manager Organisation ID; Note: The Country and Address information is fetched and displayed here in case that the Manager Organisation ID exists on COD. - Hull Shape - Power kW (MCR) - Fishing Gear - Hull Type - Main Engine RPM - Number of all engines - Service Speed (max speed at 75% of the MCR) - Fuel Type - Number of main engines - Max Manoeuvre Speed - Ice Class - Main engine designer - Max Speed - Number of Bow Thrusters - Engine manufacturer - Type of propulsion system - Number of Stern Thrusters - Engine model - Construction material - Year of Construction - Shaft Generator - Recognised Organization identifying if the organisation is: <ul style="list-style-type: none"> o Class Society o PMoU Recognised o EU Recognised - Banned - Detained - Vessel Image Upload <ul style="list-style-type: none"> o Add an image o Url Address
Step 1.3	The actor clicks on the “Next” button.
Step 1.4	<p>If there is a validation error, the user is prompted with the validation errors on the top of the page; he/she can alter the values and click again on “Next”.</p> <p>If no errors the system displays the confirmation page.</p>
Step 1.5	The confirmation page is displayed. User clicks the “Create” button to save the new vessel.
AlternativeUse Case Description	Edit existing vessel

Step 2.1	The Actor navigates to Central Ship Database> Search/Update Vessel.
Step 2.2	Execute UC-CSD-MNG-03 (Search vessel) From the search results select the desired vessel by clicking on its row.
Step 2.3	The available versions of the selected vessel are displayed. The highlighted first row represents the vessel's latest version. The user may select a vessel version to edit its details by clicking the corresponding row.
Step 2.4	On the bottom of the page press "Edit vessel details" button in order for the following fields to be editable. Update any of the fields listed in Step 1.2 above.
Step 2.5	The actor clicks on the "Next" button.
Step 2.6	If there is a validation error, the user is prompted with the validation errors on the top of the page; he/she can alter the values and click again on "Next". If no errors the system displays the confirmation page.
Step 2.7	The confirmation page is displayed. User clicks the "Update" button to save the updated vessel.
Input(s)	Vessel details mentioned in step 1.2
Output(s)	The vessel definition is saved (primary scenario) or updated (alternative scenario) in the database.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	Included use case UC-CSD-MNG-03 (Search vessel)
Special Requirements	<ol style="list-style-type: none"> 1. Upon creation the user should always include the MMSI of the vessel plus her UVI. Otherwise a validation error is to be produced to warn the user <ul style="list-style-type: none"> • Depending on the access rights of the logged administrator performing the action the system should assign the relevant reason for update to the vessel particulars created/ updated <ul style="list-style-type: none"> ○ UP_MS_VERIFIED - in case of users with CSD_MANAGER permission ○ UP_MSSO for users with CSD_ADMIN_EMSA permission.

3.2.8.2 UC-CSD-MNG-02: Consult history of changes of Vessel data

Use Case Req ID	UC-CSD-MNG-02
User Case Name	Consult history of changes of Vessel data (in CSD console)
Purpose	Covers the functionality related to the consultation of vessel history of updates in the web interface.
Subsystem	SSN CSD web console
Primary Actor(s)	Administrator
Precondition(s)	Actor is authorised for accessing to the system and for consulting vessel history.
Postcondition(s)	System provides the requested information.

Trigger(s)	The actor clicks the "Versions History" row (every row is treated as a hyperlink).
Use Case Description	Primary Scenario [Search Vessel History]
Step 1	The Actor navigates to Central Vessel Database> Search Vessel History.
Step 2	Execute UC-CSD-MNG-03 (Search vessel) From the search results select the desired vessel by clicking on its row.
Step 3	Step 2.3 of UC-CSD-MNG-01 is executed
Step 4	The Vessel Detail Versions With History Items page is displayed. The available versions of the selected vessel are displayed. The highlighted first row represents the vessel's latest version.. Per record the the fields displayed are: <ul style="list-style-type: none"> - IMO Number - MMSI Number - Call Sign - Ship Name - IR - EMSA-R - Flag registry - Date of Update - Service Indicator - Status (Active or not) -
Input(s)	Vessel search criteria.
Output(s)	The vessel detail versions with history items.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	Included use case: UC-CSD-MNG-03 (Search Vessel)
Special Requirements	N/A

3.2.8.3 UC-CSD-MNG-03: Search Vessel

Use Case Req ID	UC-CSD-MNG-03
Use Case Name	Search Vessel
Purpose	Covers the functionality related to the search of a vessel in the web interface.
Subsystem	SSN CSD web console
Primary Actor(s)	SSN User
Precondition(s)	<ul style="list-style-type: none"> - Actor is authorised for accessing to the system and for searching a vessel. - The vessel has been defined.
Postcondition(s)	System provides the requested information.
Trigger(s)	The actor selects to search a vessel via the the web interface.
Use Case Description	Primary Scenario [Search vessel]

Step 1	The Actor navigates to Central Ship Database> Search/Update Vessel.
Step 2	<p>The Actor enters the search criteria.</p> <p>At least one of the following criteria should be entered:</p> <ul style="list-style-type: none"> - IMO Number - MMSI Number - Call Sign - Ship Name <p>Actor can also enter as criteria:</p> <ul style="list-style-type: none"> - IR number - Flag registry - From-To date range: applies to the date that the vessel was created or last updated in the data store
Step 3	The actor clicks the "Search" button to query the database and fetch the vessel(s) that satisfy the criterion
Step 4	<p>The list of results is displayed.</p> <p>The search results include one entry per vessel. The results respect the source visualisation property (refer to UC-CSD-MNG-12: CSD Configuration utility). The results are exportable in XLS, CSV and XML format.</p> <p>To select a vessel to consult or edit click on its row.</p>
Input(s)	Vessel search criteria.
Output(s)	The list of vessel(s) that satisfy the criteria.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	N/A

3.2.8.4 UC-CSD-MNG-04: Consult Vessel Data

Use Case Req ID	UC-CSD-MNG-04
Use Case Name	Consult Vessel Data
Purpose	Covers the functionality related to the consultation of vessel data via the web interface.
Subsystem	CSD web console
Primary Actor(s)	SSN User
Precondition(s)	Actor is authorised for accessing the system and for consulting a vessel.
Postcondition(s)	System provides the requested information.
Trigger(s)	The actor selects to search a vessel via the the web interface.
Use Case Description	Primary Scenario: [Consult Vessel Data]
Step 1	The Actor navigates to CSD web console> Search/Update Vessel.

Step 2	Execute UC-CSD-MNG-03 (Search vessel)
Step 3	Step 2.3 of UC-CSD-MNG-01 is executed
Step 4	The details page opens and the details listed in Step 1.1 of UC-CSD-MNG-01 are displayed as described in Step 4 of UC-CSD-MNG-02 . In case of CSD_READER, the data respects the source visualisation and "data export" properties (refer to UC-CSD-MNG-12: CSD Configuration utility).
Input(s)	Vessel search criteria.
Output(s)	The vessel details that satisfy the criteria.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	Included use case: UC-CSD-MNG-03 (Search Vessel)
Special Requirements	

3.2.8.5 UC-CSD-MNG-05: Upload Vessels

Use Case Req ID	UC-CSD-MNG-05
User Case Name	Upload Vessels
Purpose	Covers the functionality related to the upload of vessels file via the web interface.
Subsystem	CSD web console
Primary Actor(s)	Administrator
Precondition(s)	Actor is authorised for accessing to the system.
Postcondition(s)	System provides the requested information.
Trigger(s)	The actor selects to upload vessel data via the the web interface.
Use Case Description	Primary Scenario [Upload Vessels]
Step 1	The Actor navigates to Central Ship Database> Upload Vessels.
Step 2	Press the "Add" button to select a .csv file to upload. Once selected, press upload to upload the file. If you want to cancel the selection, press the Cancel link. Once uploaded, the file is displayed in the <i>Uploaded file(s)</i> section. Press the "Delete File" link next to the file to delete the file.
Step 3	Press "Submit" button to finalize the upload of the file. A message that the file has been successfully uploaded is displayed.
Input(s)	A CSV formatted file.
Output(s)	A message that the data have been uploaded successfully.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-

Special Requirements	-
----------------------	---

3.2.8.6 UC-CSD-MNG-06: Validate Uploaded Vessels

Use Case Req ID	UC-CSD-MNG-06
User Case Name	Validate uploaded vessels
Purpose	Covers the functionality related to the validation of the uploaded vessels via the web interface.
Subsystem	CSD web console
Primary Actor(s)	Administrator (CSD_ADMIN EMSA)
Precondition(s)	Actor is authorised for accessing to the system. Vessels file has been uploaded.
Postcondition(s)	-
Trigger(s)	The actor selects to validate the uploaded vessel data via the web interface.
Use Case Description	Primary Scenario [Validate Uploaded Vessel Data]
Step 1	The Actor navigates to Central Ship Database> Validate Uploaded Vessels.
Step 2	<p>A list of the vessels that have been uploaded is displayed. For every vessel the following elements are displayed:</p> <ul style="list-style-type: none"> - IMO Number - MMSI Number - Ship Name - Call Sign - Duplicate Reason - User Id - Uploaded At - Existing <ul style="list-style-type: none"> • Select a vessel to validate by checking the checkbox at the end of the corresponding record. • Press the "Update" button to declare that the selected vessel records are going to be updated or registered if they are new. • Press "Finalize" button in order for the selected vessel records to be updated or registered in the database. <p>.The vessels are updated/saved in the database.</p>
Input(s)	-
Output(s)	Records are updated or registered in the database
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	UC-CSD-MNG-05 (Upload Vessel Data)

Special Requirements	
----------------------	--

3.2.8.7 UC-CSD-MNG-07 : Upload MARS Vessels

Use Case Req ID	UC-CSD-MNG-07
User Case Name	Upload MARS Vessels
Purpose	Covers the functionality related to the upload of MARS vessels file via the web interface to a stage area.
Subsystem	CSD web Console
Primary Actor(s)	Administrator (CSD_ADMIN EMSA)
Precondition(s)	Actor is authorised for accessing to the system.
Postcondition(s)	-
Trigger(s)	The actor selects to upload vessel data via the web interface.
Use Case Description	Primary Scenario [Upload MARS Vessels]
Step 1	The Actor navigates to Central Ship Database> Upload MARS vessels.
Step 2	Press the "Purge MARS stage area" to "clean" stage area.
Step 3	<p>Press the "Add" button to select a .csv file to upload. Once selected, press upload to upload the file.</p> <p>If you want to cancel the selection, press the Cancel link.</p> <p>Once uploaded, the file is displayed in the <i>Uploaded file(s)</i> section.</p> <p>Press the "Delete File" link next to the file to delete the file.</p>
Step 4	<p>Press "Submit" button to finalize the upload of the file.</p> <p>A message that the file has been successfully uploaded is displayed.</p>
Step 5	<p>An Oracle job is scheduled to handle the uploaded file;</p> <p>For MARS case, the uploaded file is loaded on the stage area. The stage area is actually tables in the database schema of the CSD that store temporarily MARS vessel records before they are integrated in the CSD database tables. Such tables are the MARS_VESSEL_EXT and MARS_VESSEL_STG.</p> <p>The uploaded information is cross-checked against the CSD/OSD. The flag Existing is set to true if at least one vessel exists on CSD or OSD with the same MMSI and Callsign.</p> <p>Given that the combination of MMSI + Callsign is double-sourced for the Vessels Registry:</p> <ul style="list-style-type: none"> For CSD, the vessel with identical pair of MMSI + Callsign if more than 1 vessels are identified consider the latest one); For OSD, the vessel with identical pair of MMSI + Callsign with the latest "nvl". The "nvl" is the date/time when the record was updated on, or if there was no update the latest date/time the record was created on. <p>The Date provided on the file is also stored on the stage area (MARS_VESSEL_STG.MARS_UPDATED_ON); this value is used for MARS version createdOn and updatedOn and Date of Effect during vessel consolidation.</p>
Input(s)	A CSV formatted file contains the vessel attributes in the following order:

	<ol style="list-style-type: none"> 1. MMSI 2. Shipname (truncated after the 48th character) 3. Callsign 4. Country (3 chars represents NATO alpha-3 codes) 5. Country (3 chars represents NATO alpha-3 codes) – it seems to be the same information twice. 6. Ships Class (2 chars General Classification)Date (10 characters - format DD/MM/YYYY) on which ship station was Added/Modified in the List of Ship Stations 7. Action A = Added to the List / M = Modified
Output(s)	A message that the data have been uploaded successfully.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	

3.2.8.8 UC-CSD-MNG-08 : Validate uploaded MARS vessels

Use Case Req ID	UC-CSD-MNG-08
User Case Name	Validate Uploaded MARS vessels
Purpose	Covers the functionality related to the upload of MARS vessels file via the web interface.
Subsystem	CSD web Console
Primary Actor(s)	Administrator
Precondition(s)	Actor is authorised for accessing to the system. Mars vessels file has been uploaded.
Postcondition(s)	System provides the requested information.
Trigger(s)	The actor selects to validate uploaded MARS Vessels.
Use Case Description	Primary Scenario [Validate Uploaded MARS Vessels]
Step 1	The Actor navigates to Central Ship Database> Validate Uploaded MARS Vessels.
Step 2	<p>A list of the vessels that have been uploaded is displayed.</p> <p>For every vessel the following elements are displayed:</p> <ul style="list-style-type: none"> - IMO_Number – nullable for all cases. - MMSI Number - Ship Name up to 48 characters – truncated to 35 characters on consolidation. - Call Sign - Country Code (3char) – ignored since it represents NATO alpha-3 codes, while SSN EIS stores ISO 3166-1 alpha-3 ones. - Geographical Area – ignored. - Ship Type – ignored. - Last action - ignored - User Id – The user performed the upload or the latest validation

	<ul style="list-style-type: none"> - Uploaded At –when the upload action performed. - Existing; true means that at least one vessel exists on CSD or OSD with the same MMSI and Callsign. <ul style="list-style-type: none"> • Select a vessel to validate by checking the checkbox at the end of the corresponding record. <p>Press the “Update” button to declare that the selected vessel records are going to be updated.</p> <ul style="list-style-type: none"> • Press “Proceed with upload” button in order for the selected vessel records to be updated in the MARS stage area. “Proceed with upload” will schedule immediately a database procedure to run on the background. All the vessels that have previously been selected with Existing flag true will be uploaded onto the database. The procedure allows to validate and finalise/upload 999 records at one go. The message “More Rows Exist”, displayed at the bottom of the page, indicates that more records are uploaded on the staging area. • Press “Proceed with bulk upload” button in order to update all the records in the MARS stage area. Proceed with bulk upload will schedule immediately a database procedure to run on the background. All the vessels with Existing flag true on the staging area will be uploaded onto the database. <p>The vessels are updated/saved in the MARS stage area. Only the records with “Existing” flag true shall be processed, the records with “Existing” flag false are ignored.</p>
Step 3	On “Finalize” and “Finalize All” actions, a database procedure is executed that creates chunks of 1000 records from uploaded-finalized MARS stage data.
Input(s)	-
Output(s)	Records are updated/registered in the stage area for MARS
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	
Special Requirements	

3.2.8.9 UC-CSD-MNG-09: Download ship list

Use Case Req ID	UC-CSD-MNG-09
User Case Name	Download Vessel Lists
Purpose	Covers the functionality related to the download of vessel lists via the web interface.
Subsystem	CSD web console

Primary Actor(s)	SSN user with CSD rights
Precondition(s)	Actor is authorised for accessing to the system.
Postcondition(s)	System provides the requested information.
Trigger(s)	The actor selects to download the list of vessels via the the web interface.
Use Case Description	Primary Scenario [Download lists]
Step 1	The Actor navigates to Central Ship Database> Download Lists > Vessels.
Step 2	<p>A list of vessels is displayed. For every vessel the following details are displayed:</p> <ul style="list-style-type: none"> - IMO Number - MMSI Number - Ship Name - Call Sign <p>At the bottom of the list there are three buttons representing the three options mentioned below.</p> <ul style="list-style-type: none"> - Export result as Excel file Press that button to download the list in an excel file - Export results as XML file Press that button to download the list in a XML file. - Export results as CSV file Press that button to download the list of vessels in a CSV file <p>Hoovering over the button an explanation message is displayed clarifying what kind of list will be downloaded if the specific button is pressed.</p>
Input(s)	-
Output(s)	A file with a list of vessels.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	N/A

3.2.8.10 UC-OSD-MNG-VV-10 : Verify and validate vessel

Use Case Req ID	UC-OSD-MNG-VV-10
Use Case Name	Verify and validate vessel
Purpose	Covers the functionality related to the manual verification and validation of vessel data via the web interface.
Subsystem	Operational Ship Database (OSD) console
Primary Actor(s)	Manager User (CSD_Admin_EMSEA)
Precondition(s)	Actor is authorised for accessing to the system and for verifying and

	<p>validating vessel data.</p> <p>A store procedure should have been executed in order to compare the new vessel records created in EIS OVR during the last day against the vessel records stored at the MarInfo DB and store the differences in a temporary storage area.</p>
Postcondition(s)	The vessel data are updated in the data store.
Trigger(s)	The actor selects to verify and validate the vessel data via the web interface.
Use Case Description	Primary Scenario: [Verify and validate vessel data]
Step 1	The Actor navigates to Management Console >Operational Ship Database (OSD) console>" Vessel Verify and Validate"
Step 2	<p>The actor defines the search criteria. The following criteria should be defined:</p> <ul style="list-style-type: none"> - From date - To date <p>Actor may also select:</p> <ul style="list-style-type: none"> - Whether to show only vessels that contain IMO or not <p>The actor clicks on the "Search" button.</p>
Step 3	<p>If there is a validation error, the user is prompted with the validation errors on the top of the page; he/she can alter the criteria values and click again on "Search".</p> <p>If no errors the system displays the results that satisfy the search criteria.</p>
Step 4	<p>The search results are displayed. For each of the following vessel attributes, their reported values in EIS_OVR data store along with their values coming from different sources (e.g. MARS, IHS, THETIS etc.) are displayed:</p> <ul style="list-style-type: none"> - IMO - MMSI - Call Sign - Ship Name <p>User may define the actual value for each of the above vessel attributes by:</p> <ul style="list-style-type: none"> - Selecting a value corresponding to the reported one or a value coming from an external source, by checking the appropriate check box at the right of each value record. - Filling manually the input field labelled 'Manual' with the desirable attribute value. <p>User should select one or more vessels to validate, by checking the checkbox at the right of each record with label "Select for Validation".</p>
Step 5	<p>User clicks on "Validate" on the bottom of the page. If he/she forgets to select at least one vessel for validation, a message is displayed on the top of the page, prompting him/her accordingly.</p> <p>If he/she selects one or more vessels, the selected vessels are displayed in the confirmation page.</p>
Step 6	The confirmation page is displayed. User may change the actual attribute values.

	He/she may select "Accept Validation" to accept a vessel as valid or leave it as invalid.
Step 7	<p>User clicks on "Accept" to accept the changes and update the data store.</p> <p>The following database tables will be updated in case user clicks on this button:</p> <ul style="list-style-type: none"> i) The vessel record in VESSELS table that corresponds to the registered vessel identified by the IMO number is updated with the reported vessel attributes. ii) A new vessel version record is inserted in VESSEL_DETAIL_VERSIONS table having as values the reported vessel attributes. iii) A new record is inserted in VESSEL_DETAILS_HISTORY table for every vessel attribute (imo, mmsi, callSign, shipName) for which a reason for update has been defined by the user, iv) If a record for the given IMO number exists in the CSD_VESSELS table, the reported vessel details will be updated in CSD_VESSELS. Otherwise, a new vessel record will be created in CSD_VESSELS. v) If at least one of the shipName, mmsi or callSign has changed, a new vessel version record is inserted in CSD_VESSEL_DETAIL_VERSIONS table. vi) For each not-null vessel identifier attribute value (e.g. imo, mmsi, callSign, flag, name, irNumber), a new record is inserted in CSD_VESSEL_DETAIL_HISTORY.
Input(s)	Search criteria mentioned in step 2 and vessel attributes mentioned in step 4.
Output(s)	The vessel data are updated in the data store.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	N/A

3.2.8.11 UC-CSD-MNG-11 : Pending CSD Update

Use Case Req ID	UC-CSD-MNG-11
Use Case Name	Pending CSD Update
Purpose	Covers the functionality related to the manual acceptance of the values of the vessel identifiers, being in "pending V&V" status, via the web interface.
Subsystem	CSD web console
Primary Actor(s)	Manager User (CSD_Admin_EMSEA)
Precondition(s)	<p>Actor is authorised for accessing to the system and for Pending CSD Update.</p> <p>Vessels exist with vessel identifiers' in "pending V&V" status.</p>

Postcondition(s)	The vessel attributes are updated in the data store.
Trigger(s)	The actor selects to update the vessel attributes via the web interface.
Use Case Description	Primary Scenario: [Update pending vessel attributes]
Step 1	The Actor navigates to Management Console > CSD web console > Pending CSD Update.
Step 2	<p>The actor defines the search criteria. The following criteria should be defined:</p> <ul style="list-style-type: none"> - From date - To date <p>The actor clicks on the "Search" button.</p>
Step 3	<p>If there is a validation error, the user is prompted with the validation errors on the top of the page; he/she can alter the criteria values and click again on "Search".</p> <p>If no errors the system displays the results that satisfy the search criteria.</p>
Step 4	<p>The search results (for every identifier its value from the current active version and its value from versions corresponding to PENDING_V_V entries in CSD_VESSEL_DETAIL_HISTORY respecting the aforementioned criteria) are displayed. For each of the following vessel identifiers, information on the source quoting the change with its date of effect along with their values registered in CSD with their corresponding date of effect are displayed:</p> <ul style="list-style-type: none"> - Flag registry - MMSI - Call Sign - Ship Name <p>The default values of vessel identifiers are those of the latest version (the PENDING versions do not update the latest one).</p> <p>User may update the value in the CSD of each of the above vessel attributes by:</p> <ul style="list-style-type: none"> - Selecting a value proposed by an external source, by checking the appropriate check box at the left side of each value record - Filling manually the input field labelled 'Manual' with the desirable attribute value. <p>User should select to accept the update of the values, by checking the "Accept Update" checkbox at the bottom right of each vessel record.</p>
Step 5	<p>User clicks on "Update" on the bottom of the page. If there is a validation error, the user is prompted with the validation errors on the top of the page; he/she can change the vessel attribute values and click again on "Update".</p> <p>If there are no validation errors, the vessel attributes are updated in the data store. Specifically, the following updates take place:</p> <ul style="list-style-type: none"> - The new values of the mentioned vessel attributes are updated in the corresponding vessel record in the CSD_VESSELS table. - The new values of the mentioned vessel attributes are updated in the latest vessel version record in the CSD_VESSEL_DETAIL_VERSIONS table.

	<p>The relevant log entry of the records in CSD_VESSEL_DETAIL_HISTORY being in "PENDING V&V" status is changed to null.</p> <p>In case that the default values (described in the previous step) are used, no more update is executed on CSD.</p> <p>In case that the user changes any of the default values (either selecting a value from the list or manually entry) a new version is created with the user changes. The reason for update is UP_MSSO, the source is SSN and the date of effect is the execution time. These changes are updated in the CSD_VESSEL_DETAIL_HISTORY records that are linked to the latest vessel version record.</p>
Input(s)	Vessel versions corresponding to PENDING_V_V entries in CSD_VESSEL_DETAIL_HISTORY
Output(s)	The vessel attributes are updated in CSD.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	N/A
Special Requirements	-

3.2.8.12 UC-CSD-MNG-12 : CSD Amend Configuration utility

Use Case Req ID	UC-CSD-MNG-12
Use Case Name	CSD Amend Configuration Utility
Purpose	Covers the functionality related to the configuration of rules per external source, such as the level of confidence, the UVI, the provided vessel attributes, etc, via the web interface
Subsystem	CSD Console
Primary Actor(s)	Manager User (CSD_Admin_EMSEA)
Precondition(s)	Actor is authorised for accessing to the system and for accessing the configuration utility.
Postconditions(s)	<p>The configuration is saved/updated in the database.</p> <p>The rules defined per source are applied to the user interface and system interface of CSD.</p>
Trigger(s)	The actor selects to create/update the configuration of a source via the web interface.
Use Case Description	Primary Scenario: [Create a new source]
Step 1	The Actor navigates to Management Console >Central Ship Database (CSD) >Configuration Utility.
Step 2	User selects to create a new source
Step 3	<p>User defines the following attributes for the new source:</p> <ul style="list-style-type: none"> - Source Code (maximum 10 characters) and Source Name

	<p>(maximum 10 characters). The values must coincide.</p> <ul style="list-style-type: none"> - if the source is private or public - Unique Vessel Identifier (UVI); IMO_Number is selected by default. - Reason for update (Primary). Prefilled by "UP_" followed by Source Name - Level of confidence (1,2,3,...) - Method for the injection of the data (Applications): <ol style="list-style-type: none"> 1) Ship particular notification 2) EIS notification 3) Other (Upload via database stored procedure). - The Secondary "ReasonForUpdate" and the Level of Confidence for the Source (optional, as there is no effect on the configuration) - The visualization restrictions (in web consoles and exporting tools) and data exchange restrictions (via the system interface on a single vessel basis or in bulk) per vessel attributes. That is the user should be able to define, per vessel attributes: <ol style="list-style-type: none"> 1. The visualisation possibility (Yes/No) (via web interface for CSD_READERS), 2. The data export possibility (Yes/No) (via web interface for CSD_READERS), 3. The possibility to exchange the particular via the system interface on a single vessel basis (Yes/No), and 4. The possibility to exchange the particular via the system interface in bulk (Yes/No).
Step 4	User saves the configuration
Use Case Description	Alternative Scenario: [Update an existing source]
Step 1	The Actor navigates to Central Ship Database> Configuration Utility.
Step 2	User selects to update an existing source. The configuration details of the selected source are displayed
Step 3	User may update the following configuration attributes. The changes will take affect from that time onwards. User may change the rest of the configuration attributes mentioned in step 3 of the primary scenario.
Step 4	User saves the configuration
Input(s)	The configuration details defined in step 3
Output(s)	The configuration for the source is saved in the CSD
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	-

Special Requirements	
----------------------	--

3.2.8.13 UC-CSD-UPL-MARS-13: MARS Data Processing

Use Case Req ID	UC-CSD-UPL-MARS-13
Use Case Name	Data from MARS processing
Purpose	Covers the functionality of processing the vessel attributes reported from MARS
Subsystem	SSN Resources Core
Primary Actor(s)	EIS Resources system
Precondition(s)	MARS data from external files should have been loaded to the MARS staging area.
Postcondition(s)	The vessel attributes are updated considering the data from MARS.
Trigger(s)	
Use Case Description	Primary Scenario: [Update vessel attributes in MARS dataset]
Step 1	SSN Resources Core loads MARS data from the stage area. It processes each record separately.
Step 2	The latest version of the existing vessel is loaded (CSD/OSD). The MARS version has <ul style="list-style-type: none"> ReasonForUpdate the configured primary reason of update of Source with code "MARS" and DateEffect the datetime of the data load from the MARS stage area.
Step 3	The two versions mentioned in Step 2 are consolidated and the Vessel Registry of the existing vessel is updated.
Input(s)	MARS vessel data loaded in stage area
Output(s)	The vessel data are updated in the CSD and OSD.
Timer(s)	The vessels synchronisation procedure triggered according to EIS application parameterSSN_VESSELS_UPD_PERIOD.
Business Process(es) Reference	-
Associated Use Case(s)	-
Special Requirements	

3.2.8.14 UC-CSD-SPN-14: Process ShipParticulars Notification

Use Case Req ID	UC-CSD-SPN-14 (Revised)
Use Case Name	Process Ship Particulars Notification
Purpose	Covers the functionality related to Ship Particulars Notification processing via the system to system SOAP interface (MS2SSN_ShipParticulars_Not message as SOAP payload) or from another source stored in the database i.e. MARINFO).

	Inactive vessels are excluded.
Subsystems	SSN Ship Particulars Web Service, SSN Resources Core.
Primary Actor(s)	External reference source provides vessel attributes.
Precondition(s)	Actor is authorised for accessing to the system with task <i>SHIPPARTICULARS_NOTIFIER</i> .
Postcondition(s)	The vessel is inserted / updated in the CSD/OSD. In addition, in case of MS2SSN_ShipParticulars_Not the system sends a receipt to the Data Provider to indicate the successful reception and processing of the message details.
Trigger(s)	In case of MS2SSN_ShipParticulars_Not Received; the actor sends an SOAP message with payload of type MS2SSN_ShipParticulars_Not to SSN Ship Particulars Exchange Web Service Endpoint. In case of another source stored in the database, a timer triggers the process at "SSN_VESSELS_UPD_PERIOD" interval defined as an application parameter..
Use Case Description	Primary Scenario – Valid Incoming notification
Step 1	<p>The system identifies that the following information are provided:</p> <p>Vessel Attributes:</p> <p>Vessel Identifiers:</p> <ul style="list-style-type: none"> - IMO Number - MMSI Number - Ship Name - CallSign - IR Number - EMSA-R Number - Flag Registry - XR Number <p>Vessel Particulars:</p> <ul style="list-style-type: none"> - Inmarsat Call Number - Ship Type (LRIT) - Ship Type (PSC) - Ship Type (IHS) - Ship Type (AIS) - Ship Type(Lloyds') - Ship Type (UN) - Keel Laying Date - Contract Date - Gross Tonnage - Length (Overall) - Max Number of Passengers - Reduced Gross Tonnage - Length between Perpendiculars - Number of crew - Net Tonnage - Keel To Mast Height - Number of RORO Compartments - Breadth (Moulded) - DeadWeight - TEU

	<ul style="list-style-type: none"> - Beam - Draught - Number of Cargo Tanks - Number of Holds - ISM Company IMO Number - Registered OwnerId - Registered Owner IMO Number - Hull Shape - Power kW (MCR) - Fishing Gear - Hull Type - Main Engine RPM - Number of all engines - Service Speed (max speed at 75% of the MCR) - Fuel Type - Number of main engines - Max Manoeuvre Speed - Ice Class - Main engine designer - Max Speed - Number of Bow Thrusters - Engine manufacturer - Type of propulsion system - Number of Stern Thrusters - Engine model - Construction material - Recognised Organization <ul style="list-style-type: none"> o Class Society o PMoU Recognised o EU Recognised <p>Current Vessel Status</p> <ul style="list-style-type: none"> - Service Indicator - Banned - Detained - Status (from PSC)
Step 2	The system validates the contents against the SSN-SIG-PartB_ShipParticularsExchange [R6] business rules.
Step 3	<p>The Vessel Resolution (UC-CSD-SPN-16) is executed and returns:</p> <ul style="list-style-type: none"> • Vessel is not resolved. The system will then consider the level of confidence of the source: <ul style="list-style-type: none"> o Level is the highest (see Complementary Scenario 4.1) o Level is not the highest (see Complementary Scenario 4.2) • Vessel is resolved in CSD. The system will then consider the vessel in the CSD: <ul style="list-style-type: none"> o Vessel is not broken-up (see Complementary Scenario 4.3) o Vessel is broken-up (see Complementary Scenario 4.4) • Vessel is resolved in OSD (see Complementary Scenario 4.5)
Step 4	Proceed with the complementary scenarios 4.x.
Step 5	To check if the MMSI reported for the vessel is found to be listed in the active identity of another vessel "B" in the CSD, search for vessels with the reported MMSI in the CSD.

Step 6	If more than one matching vessels are found then update the vessels found in CSD by setting their status = "Pending V&V". The vessel detail version found will also be flagged as "Pending V&V". If no matching vessels are found do nothing.
Use Case Description	Complementary Scenario 4.1: Vessel is not resolved and the level of confidence of the reporting source is the highest ("1").
Step 4.1.1	A new vessel is created in the CSD. The ReasonForUpdate and Date of Effect to be registered for each of the vessel attributes are the attached to the vessel attributes reported.
Use Case Description	Alternative Complementary Scenario 4.2: Vessel is not resolved and the level of confidence of the reporting source is not the highest ("2" and above).
Step 4.2.1	A new vessel is created in the OSD. The version created has status = "Temporary". The ReasonForUpdate and Date of Effect to be registered for each of the vessel attributes are the attached to the vessel attributes reported.
Use Case Description	Complementary Scenario 4.3: Vessel is resolved in CSD and is not Broken-up (dead) vessel.
Step 4.3.1	A new version of the vessel is created. The ReasonForUpdate and Date of Effect to be registered for each of the vessel attributes are the attached to the vessel attributes reported.
Step 4.3.2	If the reported vessel is from a "private" source while the vessel resolved is from a "public" source and the "public" and "private" source have the same level of confidence then do not update the consolidated vessel version. Else go to the step 4.3.3.
Step 4.3.3	Update the the consolidated vessel version based on BR 10 [Annex B: CSD Specific Business Rules]. Specifically, for the vessel name update based on BR 7 [Annex B: CSD Specific Business Rules] (the indication if the name differs is already provided by the Vessel Resolve).
Use Case Description	Complementary Scenario 4.4: Vessel is resolved in CSD and is Broken-up (dead) vessel.
Step 4.4.1	The vessel Service Indicator and Status indicate the vessel is Broken-up (dead) vessel. The system responds with an SSN_Receipt with StatusCode=OK. The Ship Notification is ignored.
Use Case Description	Complementary Scenario 4.5: Vessel is resolved in OSD
Step 4.5.1	A new version for the vessel is created in the OSD.
Step 4.5.2	If the reported vessel is from a "private" source while the vessel resolved is from a "public" source and the "public" and "private" source have the same level of confidence then do not update the consolidated vessel version. Else go to the step 4.5.3.
Step 4.5.3	Update the consolidated vessel version based on BR 10 [Annex B: CSD Specific Business Rules]. Specifically, for the vessel name update based on BR 7 [Annex B: CSD Specific Business Rules] (the indication if the name differs is already provided by the Vessel Resolve).
Alternative Use Case	Alternative Scenario 2: Invalid notification according to SIG rules.

Description	
Step 2.1	The Ship Particular Notification is rejected.
Input(s)	Ship Notification details
Output(s)	The vessel definition is saved or updated in the CSD/OSD. In addition, in case of MS2SSN_ShipParticulars_Not the receipt (the StatusMessage included the CSDID in case of StatusCode=OK) is sent to the data provider.
Timer(s)	In case of another source stored in the database, the trigger fires at intervals defined by the application parameter 'SSN_VESSELS_UPD_PERIOD'.
Business Rules Reference	- BR 10 [Annex B: CSD Specific Business Rules]. - BR 7 [Annex B: CSD Specific Business Rules]
Associated Use Case(s)	Included Case: UC-SSN-MSG-01: Handle Incoming Message, , UC-CSD-SPN-16 Extended UC: UC-SSN-NOT-04 - Process Notification
Special Requirements	N/A
Reference Diagram	UC-CSD-SPN-14 Process Ship Particulars Notifications

3.2.8.15 UC-OSD-PPN-15 : Process Vessel Attributes reported in EIS notification

Use Case Req ID	UC-OSD-PPN-15 (Revised)
Use Case Name	Process Vessel Attributes reported in EIS notification
Purpose	Updates the SSN OSD with the vessel attributes included in the EIS notification messages of type PortPlus, Ship (MRS) and IncidentDetails and Exemption received by EIS.
Subsystem	SSN Core
Primary Actor(s)	SSN User / NCA Application
Precondition(s)	The Incoming Message is valid and includes vessel data.
Postcondition(s)	SSN OSD updated.
Trigger(s)	EIS Notification Received
Use Case Description	Primary Scenario – Valid Reporting Vessel
Step 1	<p>The system identifies that the vessel identification particulars are contained in the message body:</p> <p>Vessel attributes:</p> <p>Vessel Identifiers:</p> <ul style="list-style-type: none"> - IMO Number - IR Number - MMSI Number - Call Sign - Ship Name - Flag <p>Vessel Particulars may be contained in the message body:</p> <ul style="list-style-type: none"> - Ship Type - Gross Tonage

	<ul style="list-style-type: none"> - Certificate of Registry date - Certificate of Registry number - Certificate of Registry port - Certificate of Registry PMane - IMO Company number - Company name
Step 2	<p>The Vessel Resolution (UC-OSD-SPN-17) is executed and returns:</p> <ol style="list-style-type: none"> 1. Vessel is not resolved in OSD (see Complementary Scenario 3.1) 2. Vessel is resolved in OSD (see Complementary Scenario 3.2)
Step 3	Proceed with the complementary scenarios 3.x.
Step 4	To check if for the same vessel with status = "Temporary" exist 2 vessel versions from different sources, search the OSD for vessels with status = "Temporary" that have at least 2 vessel versions from different sources.
Step 5	<p>If more than one vessel versions are found then go to step 4.3.</p> <ul style="list-style-type: none"> - Update the vessels with status = "Temporary" to "Validated". - Update the latest consolidated vessel version status to "Validated". <p>If not then do nothing.</p>
Use Case Description	Complementary Scenario 3.1: Vessel is not resolved in OSD
Step 3.1.1	<p>A new vessel and a new vessel version is created in the OSD with the vessel identifiers.</p> <ul style="list-style-type: none"> - The status created has status = "Temporary". - The created date is the SYSDATE. - The ReasonForUpdate for each vessel identifier is "UP_MS_VERIFIED". - The Source for each vessel identifier is "MS". - The Date of Effect for each vessel identifier is the SYSDATE.
Step 3.1.3	<p>The vessel particulars listed hereunder are contained in the EIS notification:</p> <ul style="list-style-type: none"> - Gross Tonnage - Certificate of Registry date - Certificate of Registry number - Certificate of Registry port - Certificate of Registry PMane - IMO Company number - Company name <p>Store the values in the newly created vessel version.</p> <ul style="list-style-type: none"> - The ReasonForUpdate for each vessel particular is "UP_MS_VERIFIED". - The Source for each vessel particular is "MS". - The Date of Effect for each vessel particular is the SYSDATE.
Use Case Description	Complementary Scenario 3.2: Vessel is resolved in OSD
Step 3.2.1	<p>Check if the existing vessel is Validated and Active.</p> <p>If yes, then do nothing.</p> <p>Else continue with the next step 3.2.2.</p>

Step 3.2.2	<p>Check if a vessel version exists matching the vessel identifiers that are reported i.e. MMSI Number and/or CallSign and/or ShipName.</p> <p>If no matching version is found then a new vessel version is created in the OSD with the vessel identifiers.</p> <ul style="list-style-type: none"> – The version created has status = "Temporary". – The ReasonForUpdate for each vessel identifier is "UP_MS_VERIFIED". – The Source for each vessel identifier is "MS". – The Date of Effect for each vessel identifier is the SYSDATE. – The created date is the SYSDATE. <p>If a matching version is found then go to step 3.2.3.</p>
Step 3.2.3	<p>A vessel version exists matching the vessel identifiers that are reported i.e. MMSI Number, CallSign, ShipName.</p> <p>Check if the matching version's source is the same (e.g. "MS").</p> <p>If no, then create a new vessel version in the OSD with the reported vessel identifiers:</p> <ul style="list-style-type: none"> – The version created has status = "Temporary". – The ReasonForUpdate for each vessel identification is "UP_MS_VERIFIED". – The Source for each vessel identification is "MS". – The Date of Effect for each vessel identification is the SYSDATE. – The created date is the SYSDATE. <p>If yes, then do not create a new vessel version.</p>
Step 3.2.4	<p>The vessel particulars listed here are contained in the EIS notification:</p> <ul style="list-style-type: none"> – Gross Tonnage – Certificate of Registry date – Certificate of Registry number – Certificate of Registry port – Certificate of Registry PMane – IMO Company number – Company name – Update the consolidated vessel version with the vessel particulars listed above. The ReasonForUpdate for each vessel particular is "UP_MS_VERIFIED". – The Source for each vessel particular is "MS". – The Date of Effect for each vessel particular is the SYSDATE.
Input(s)	EIS Notification
Output(s)	OSD updated with reported vessel attributes.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	Extended Case: UC-SSN-PPN-05: Process PortPlus Notification, UC-SSN-MRS-08: Process MRS Notification.
Special Requirements	
Reference Diagram	UC-CSD-SPN-15 Process Ship Particulars notifications in EIS

3.2.8.16 UC-CSD-SPN-16 : Vessel Resolution – Ship Particulars Notification

Use Case Req ID	UC-CSD-SPN-16 (Revised)
Use Case Name	Vessel Resolution – Ship Particulars Notification
Purpose	The objective of the vessel resolution is to identify if the vessel attributes reported are already registered in ship database (CSD or OSD). Inactive vessels are excluded.
Subsystem	SSN Resources Core
Primary Actor(s)	NCA Application/STIRES-STAR provides EIS notifications; External source provides vessel attributes.
Precondition(s)	Process ShipParticulars Notification
Postcondition(s)	-
Trigger(s)	Notification Received
Use Case Description	Vessel UVI is reported
Step 1	Search the CSD based on the vessel identifiers that comprise the UVI (IMO Number or if IMO Number is not reported the IR Number or if IMO Number and IR Numbers are not reported the EMSA_R Number). Reported MMSI, Ship name and Call Sign ignored; not included in the query where clause. Note: IMO number prevails as the primary identifier when both IR and IMO numbers are provided for the same vessel record.
Step 2	If a matching vessel is found, then return the latest valid version of the vessel including: <ul style="list-style-type: none"> An indication if the vessel name (when reported) differs to the vessel name of the identified vessel based on BR 7 [Annex B: CSD Specific Business Rules]. The vessel Service indicator and Status (this is an indicator if the vessel is inactive (dead) or not). If no matching vessel is found and the reporting source level of confidence is other than the highest then go to step 3.
Step 3	If no matching vessel is found in Step 2. Search the OSD based on the vessel identifiers that comprise the UVI (IMO Number or IR Number) or MMSI Number if reported. Note: IMO number shall prevail as the primary identifier when both IR and IMO numbers are provided for the same vessel record.
Step 4	If a matching vessel is found then return the latest valid version of the vessel including the vessel Service indicator and Status (this is an indicator if the vessel is inactive (dead) or not). Else if no matching vessel is found then return null.
Input(s)	The reported vessel attributes.

Output(s)	If the Vessel is not resolved neither in CSD nor in OSD: Null. Else if the Vessel is resolved in CSD: the resolved CSD Vessel Record. Else if Vessel is resolved in OSD: the resolved OSD Vessel Record.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	Extended UC: UC-SSN-NOT-04: Process Notification
Special Requirements	
Reference Diagram	UC-CSD-SPN-14 Process Ship Particulars Notifications

3.2.8.17 UC-OSD-SPN-17 : Vessel Resolution - EIS Notification

Use Case Req ID	UC-OSD-SPN-17 (New)
Use Case Name	Vessel Resolution – EIS Notification
Purpose	Covers the functionality related to the vessel resolution reported in an EIS notification. The primary scope of the vessel resolution is to identify if the vessel attributes reported are registered in operational ship database (OSD). Inactive ships are excluded.
Subsystem	SSN Resources Core
Primary Actor(s)	NCA Application / STIRES-STAR
Precondition(s)	Process EIS Notification reporting the vessel IMO Number, IR Number or MMSI Number.
Postcondition(s)	-
Trigger(s)	Notification Received
Use Case Description	Vessel IMO Number or IR Number are reported
Step 1	Search the OSD based on the vessel identifiers that comprise the UVI (IMO Number or IR Number). Note: IMO number shall prevail as the primary identifier when both IR and IMO numbers are provided for the same vessel record.
Step 2	If a matching vessel is found then return the latest valid version of the vessel. Else if no matching vessel is found then return null.
Step 3	If no matching vessel is found in Step 2. Search the OSD based on the MMSI Number plus ShipName and-or CallSign.
Step 4	If a matching vessel is found then return the latest valid version of the vessel. Else if no matching vessel is found then return null.

Alternative Use Case Description	
	Vessel IMO Number and IR Number are not reported (Vessel MMSI is reported).
Step 2.1	Search the OSD based on the MMSI Number for vessels with status valid.
Step 2.2	If a matching vessel is found then return the latest valid version of the vessel. Else if no matching vessel is found and the Ship Name and/or Call Sign are reported then go to step 2.3.
Step 2.3	Search the OSD based on the Ship Name and/or Call Sign.
Step 2.4	If a matching vessel is found then return the latest valid version of the vessel. Else if no matching vessel is found then return null.
Input(s)	The reported vessel attributes.
Output(s)	If the Vessel is not resolved in OSD: Null. Else if Vessel is resolved in OSD: the resolved OSD Vessel Record.
Timer(s)	-
Business Process(es) Reference	-
Associated Use Case(s)	Extended UC: UC-SSN-NOT-04: Process Notification
Special Requirements	
Reference Diagram	UC-CSD-SPN-15 Process Ship Particulars notifications in EIS

3.2.8.18 UC-SSN-CSD-18 : CSD synchronisation with OSD

Use Case Req ID	UC-SSN-CSD-18 (Revised)
Use Case Name	CSD synchronisation with OSD
Purpose	Covers the functionality related to the CSD synchronisation with the OSD. The scope is to update the vessels in the CSD based on valid vessels that are not broken-up in the OSD.
Subsystem	SSN Resources Core
Primary Actor(s)	OSD
Precondition(s)	Vessel is among those that are candidate for the CSD conditional update job OSD vessels with status = "Validated" and not broken-up exist that were updated in the past "SSN_VESSELS_SYNC_INTERVAL".
Postcondition(s)	The vessel is insterted/updated in the CSD
Trigger(s)	A timer triggers the process at "SSN_VESSELS_SYNC_INTERVAL" interval defined as an application parameter.

Use Case Description	Primary Scenario – Valid OSD vessel resolved in CSD
Step 1	Select the vessels from the OSD with status = "Validated" and are not broken-up that were updated in the past "SSN_VESSELS_SYNC_INTERVAL". Per vessel select all the attributes in the OSD.
Step 2	Search the CSD based on the vessel identifiers that comprise the UVI (IMO Number or IR Number). MMSI, Ship name and Call Sign ignored; not included in the query where clause. Note: IMO number prevails as the primary identifier when both IR and IMO numbers are provided for the same vessel record.
Step 3	If a matching vessel is found the vessel is resolved. <ul style="list-style-type: none"> • If the Vessel is not broken-up (see Complementary Scenario 4.2) • If the Vessel is broken-up (see Complementary Scenario 4.3) If no matching vessel is found the vessel is not resolved (see Complementary Scenario 4.1).
Step 4	Proceed with the complementary scenarios 4.x.
Step 5	To check if the MMSI reported for the vessel is found to be listed in the active identity of another vessel "B" in the CSD, search for vessels with the reported MMSI in the CSD.
Step 6	If more than one matching vessels are found then update the vessels found in CSD by setting their status = "Pending V&V". are found then update the vessels found in CSD by setting their status = "Pending V&V". The vessel detail version found will also be flagged as "Pending V&V". If no matching vessels are found there is no further action.
Use Case Description	Complementary Scenario 4.1: Vessel is not resolved. The level of confidence of the reporting source is considered highest ("1").
Step 4.1.1	A new vessel is created in the CSD. The ReasonForUpdate and Date of Effect to be registered for each of the vessel attributes are the attached to the vessel attributes reported.
Use Case Description	Complementary Scenario 4.2: Vessel is resolved in CSD and is not Broken-up (dead) vessel.
Step 4.2.1	The ReasonForUpdate and Date of Effect to be registered for each of the vessel attributes are the attached to the vessel attributes reported.
Step 4.2.2	If the reported vessel is from a "private" source while the vessel resolved is from a "public" source and the "public" and "private" source have the same level of confidence then do not update the consolidated vessel version. Else go to the step 4.2.3.
Step 4.2.3	Update the the consolidated vessel version based on BR 10 [Annex B: CSD Specific Business Rules]. Specifically, for the vessel name update based on BR 7 [Annex B: CSD Specific Business Rules] (the indication if the name differs is already provided by the Vessel Resolve).
Use Case Description	Complementary Scenario 4.3: Vessel is resolved in CSD and is Broken-up (dead) vessel.

Step 4.3.1	The vessel Service Indicator and Status indicate the vessel is Broken-up (dead) vessel. The vessel from OSD is ignored.
Input(s)	Vessels with UVI+MMSI in validated status that are not broken-up in the OSD and were updated in the past "SSN_VESSELS_SYNC_INTERVAL" (defined as application parameter).
Output(s)	The vessel definition is saved or updated in the CSD.
Timer(s)	The trigger fires at intervals defined by the application parameter "SSN_VESSELS_SYNC_INTERVAL".-
Business Process(es) Reference	- BR 10 [Annex B: CSD Specific Business Rules]. - BR 7 [Annex B: CSD Specific Business Rules]
Associated Use Case(s)	N/A
Special Requirements	N/A
Reference Diagram	UC-18 CSD synchronisation with OSD

3.2.8.19 UC-SSN-OSD-19 : OSD synchronisation with CSD

Use Case Req ID	UC-SSN-OSD-19 (Revised)
Use Case Name	OSD synchronisation with CSD
Purpose	Covers the functionality related to the OSD synchronisation with the CSD. The scope is to update the vessels in the OSD based on vessels in the CSD.
Subsystem	SSN Resources Core
Primary Actor(s)	CSD
Precondition(s)	CSD vessels exist that were updated in the past "SSN_VESSELS_SYNC_INTERVAL".
Postcondition(s)	The vessel is insterted/updated in the OSD
Trigger(s)	A timer triggers the process at "SSN_VESSELS_SYNC_INTERVAL" interval defined as an application parameter.
Use Case Description	Primary Scenario – CSD vessel resolved in OSD
Step 1	Select the vessels from the CSD that are updated in the past "SSN_VESSELS_SYNC_INTERVAL". Per vessel select all the attributes in the CSD.
Step 2	Search the OSD based on the vessel identifiers that comprise the UVI (IMO Number or IR Number), if the UVI is not found in OSD search by MMSI Number if reported. Note: IMO number shall prevail as the primary identifier when both IR and IMO numbers are provided for the same vessel record.
Step 3	If a matching vessel is found the vessel is resolved (see Complementary Scenario 4.2) If no matching vessel is found the vessel is not resolved (see Complementary Scenario 4.1). Proceed with the complementary scenarios hereunder.

Use Case Description	Complementary Scenario 4.1: Vessel is not resolved. The level of confidence of the reporting source is considered highest ("1").
Step 4.1.1	A new vessel is created in the OSD. The ReasonForUpdate and Date of Effect to be registered for each of the vessel attributes are the attached to the vessel attributes reported.
Use Case Description	Complementary Scenario 4.2: Vessel is resolved in OSD
Step 4.2.1	If the reported vessel is from a "private" source while the vessel resolved is from a "public" source and the "public" and "private" source have the same level of confidence then do not update the consolidated vessel version. Else go to the step 4.2.2.
Step 4.2.2	Update the consolidated vessel version based on BR 10 [Annex B: CSD Specific Business Rules]. Specifically, for the vessel name update based on BR 7 [Annex B: CSD Specific Business Rules] (the indication if the name differs is already provided by the Vessel Resolve).
Input(s)	Vessels with UVI+MMSI in the CSD that were updated in the past "SSN_VESSELS_SYNC_INTERVAL" (defined as application parameter).
Output(s)	The vessel definition is inserted or updated in the OSD.
Timer(s)	The trigger fires at intervals defined by the application parameter "SSN_VESSELS_SYNC_INTERVAL".-
Business Process(es) Reference	- BR 10 [Annex B: CSD Specific Business Rules]. - BR 7 [Annex B: CSD Specific Business Rules]
Associated Use Case(s)	N/A
Special Requirements	N/A
Reference Diagram	UC-19 OSD synchronisation with CSD

3.2.9 Voyage Calculation Process

Business process specifics	
Responsibilities	<p>Define the vessel's voyage by correlating Port, PortPlus, Hazmat and Ship notification data regarding the vessel direction and expected/actual times of departure/arrival to and from Ports.</p> <p>The primary scope of the voyage calculation is to identify all the notifications transmitted to SSN-EIS that refer to the same ship voyage. PortPlus notifications send for the same ship call are parts of the same ship voyage. However, a ship voyage may include notifications send by more than one data providers. This acceptance is based on the fact that data providers associate the PortPlus notifications for the same ship call identified by the same <i>ShipCallId</i>. A voyage may contain also notifications of type Port, Hazmat and Ship.</p>
User role(s)	SSN Core
Action History	<p>Every Port, PortPlus, Hazmat and Ship notification will be processed to calculate the voyage of the resolved vessel.</p> <p>Any errors are logged into the database log tables (TLOG).</p>

Business process specifics

Processing logic

Assumption 1: all PortPlus notifications send for the same ship call are parts of the same ship voyage.

Assumption 2: a ship voyage may include notifications send by more than one data providers.

Assumption 3: it is expected that for the same ShipCall - to a given PortOfCall - either a Port notification (with NextPortOfCall at ETA) or a PortPlus notification (with PortOfCall at ETAToPortOfCall) will be send but not both.

A new voyage may begin:

- from the *LastPort* (LP1) at *ETDFromLastPort*, provided only in a PortPlus notification at NVL(ATAPortOfCall, ETAToPortOfCall)

and end

- to the *PortOfCall* (PC1) at NVL(ATAPortOfCall, ETAToPortOfCall) provided as *PortOfCall* in the PortPlus notification

OR anew voyage may begin:

- from the *PortOfCall* (LP2) at *ATDPortOfCall* provided in the PortPlus notification

OR

- from the *NextPortOfCall* (LP2) at *ETD* provided in the Port notification

and end either

- to the *NextPort* (PC2) at *ETAToNextPort* provided in the PortPlus notification

OR

- to the *NextPortOfCall* (PC2) at *ETA* provided in the Port notification, Hazmat notification and/or Ship notification.

Any notifications send during the voyage of the vessel from LP1 to PC1 will be assigned the same *VoyageId*.

The rules for assigning a new *VoyageId* or assigning a new notification to an existing voyage:

1. Receipt of a PortPlus notification with a new ShipCallId1.

If no Voyage exists with equal ShipCallId1 PortOfCall and NVL(ATAPortOfCall, ETAToPortOfCall) at the approximation of 1 hour then

- a new Voyage1 is defined;
Start: LastPort at ETDFromLastPort
End: PortOfCall and NVL(ATAPortOfCall, ETAToPortOfCall)
- a new Voyage2 is defined;
Start: PortOfCall at ATDPortOfCall

Business process specifics

End: NextPort at ETAToNextPort

Receipt of a PortPlus notification with a new ShipCallId2

If Voyage2 exists with EndPort = ShipCallId2 PortOfCall and NVL (ATAPortOfCall, ETAToPortOfCall) ≈ Voyage2 ETAToNextPort with an approximation of 1 hour then ShipCall2 is assigned to Voyage2.

If Voyage2 exists and EndPort is not defined but NVL (ATAPortOfCall, ETAToPortOfCall) > Voyage2 ATDPortOfCall with an approximation of 1 hour then ShipCall2 is assigned to Voyage2.

Receipt of a PortPlus notification with a new ShipCallId3

In the exceptional case that Voyage1 exists and StartPort = ShipCallId3 PortOfCall and NVL (ATAPortOfCall, ETAToPortOfCall) ≈ Voyage1 ETDFromLastPort with an approximation of 1 hour then ShipCall3 is assigned to Voyage1.

Receipt of a PortPlus notification with a new ShipCallId4

If Voyage4 exists (see Port notification below) and ShipCallId4 LastPort = NextPortOfCall and ETDFromLastPort ≈ Voyage4ETA with an approximation of 1 hour, then ShipCall4 is assigned to Voyage4.

If Voyage4 exists (see Port notification below) and ShipCallId4 NextPort = NextPortOfCall and ETAToNextPort ≈ Voyage4ETA with an approximation of 1 hour, then ShipCall4 is assigned to Voyage4.

If defined the *LastPort* will be the starting point. The *LastPort* is optional and as such it could be unknown if not defined. If not defined the voyage is incomplete.

The *PortofCall* attribute is mandatory and will define the ending point of 1 voyage and the starting point of another voyage if *ATDPortOfCall* is given. In case of cancellation the *PortOfCall* is cancelled hence the voyage becomes incomplete.

If defined the *NextPort* will be the ending point. The *NextPort* is optional and as such it could be unknown if not defined. If not defined the voyage is incomplete.

2. Receipt of Port notification.

If no Voyage exists with equal NextPortOfCall and ETA at the approximation of 1 hour, then a new Voyage4 is defined.

If a Voyage2 exist and EndPort = NextPortOfCall and ETA ≈ Voyage2 ETAToNextPort with an approximation of 1 hour, then the Port notification is assigned to Voyage2.

Business process specifics

If a Voyage2 exist and EndPort is not defined but ETA > Voyage2 ATDPortOfCall then the Port notification is assigned to Voyage2.

In the exceptional case that Voyage1 exists and StartPort = NextPortOfCall and ETA ≈ Voyage1 ETDFromLastPort with an approximation of 1 hour then the Port notification is assigned to Voyage1.

In case of cancellation the *NextPortOfCall*, *ETA*, *ETD* are cancelled hence the voyage becomes incomplete.

3. Hazmat Notification. Depending on the *NextPortOfCall* and the ETA (with an approximation of 1 hour) the notification will be assigned a voyage created for a Port or PortPlus notification send before. If no such case, then a new voyage will be created and assigned to the Hazmat notification.

In case of cancellation of the Port or PortPlus of the voyage the Hazmat will remain as part of the voyage. The Hazmat could/could not be updated by the data provider. In the 1st case the system will maintain the previous Hazmat in the voyage and the next Port or PortPlus notification will give the next Port of Call after the cancellation. In the 2nd case – if the data provider sends a Hazmat update – the new Hazmat will be assigned to the voyage. This way the Hazmat remains known and correlated with the ship's voyage.

In case of *NextPortOfCall* = "ZZUKN" no voyage is assigned.

4. Ship Notification. Depending on the *NextPortOfCall* and the ETA (with an approximation of 1 hour) the notification will be assigned a voyage created for a Port or PortPlus notification send before. If no such case, then a new voyage will be created and assigned to the Ship notification.

In case of Ship (AIS) notification with a technically incorrect *NextPortOfCall* will not be considered in the voyage calculation.

In case of cancellation of the Port or PortPlus of the voyage the Ship notification will remain as part of the voyage.

In case of *NextPortOfCall* = "ZZUKN" no voyage is assigned.

Based on the time of departure from the *LastPort* and the time of arrival to the *PortOfCall* a voyage can be characterized as:

- Previous Voyage: the voyage is completed, and the vessel has departed from the *PortOfCall*.
Criteria: current timestamp is greater than ATDFromPortOfCall.
- Current Voyage: The vessel is in between the *LastPort* and the *PortOfCall*.
Criteria: current timestamp in between ETDFromLastPort and ETAToPortOfCall / ETA.
- Future Voyage(s): the vessel has not departed from the *PortOfCall* yet.

Business process specifics	
	<p>Criteria: current timestamp is less than NVL (ATAPortOfCall, ETAToPortOfCall) / ETA.</p> <p>It is expected that the <i>LastPort</i> and the expected/actual time of departure is not notified to EIS on time; in this case a provisional indication of the <i>LastPort</i> and departure time could be given in the form of a PortPlus Notification. PortPlus notification data that originate from STIRES can be distinguished based on the data provider (being STIRES) and as such must be considered as provisional only.</p> <p>It should be noted that provisional ShipCall information shall not be provided to the request for details by a data requestor.</p>
Validation Rules	<ul style="list-style-type: none"> • Vessels must be identified by IMO Number and/or MMSI Number. • The LASTPORT and PORTOFCALL must define a technically correct LOCATION registered in SSN-EIS. • LASTPORT must be different from PORTOFCALL. • NVL (ATAPortOfCall, ETAToPortOfCall) must be less than ETDFromPortOfCal. • NVL (ATAPortOfCall, ETAToPortOfCall) must be greater than ETDFromLastPort. • ATAPortOfCall must be less than ATDFromPortOfCall. • ETAToNextPort must be greater than NVL (ATDPortOfCall, ETDFromPortOfCall).
Database Transactions	<ul style="list-style-type: none"> • Vessel voyage details are persisted on the VOYAGES table.

4 Design of System Components

4.1 SSN-EIS

This section provides an overview of the SSN-EIS System.

A sovereign element in the design of the European Index Server (hereinafter EIS), as depicted in the logical view diagram presented in Figure 4-1, is the division of system in three distinguishable applications:

1. **ssn-core-app** (ref: section 4.2)
2. **ssn-console-app** (ref: section 4.4)
3. **ssn-xmlprotocol-app** (ref: section 4.5)

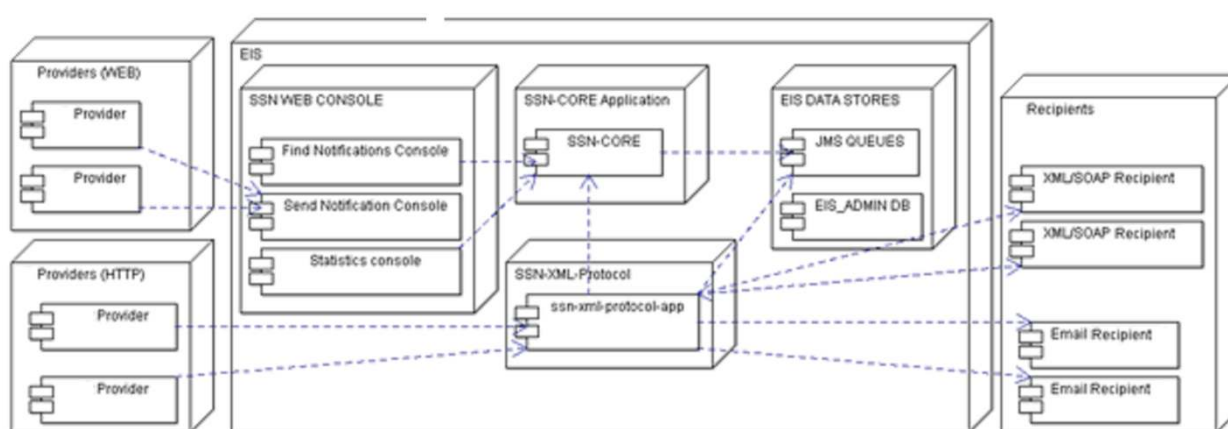


Figure 4-1 SSN EIS components connections diagram

The first application materialises the main functionality of EIS, which is the management of Messages, Vessels in a way that is independent from the channel of communication through which messages are exchanged (HTTP/Html via the web console applications) or the natural representation of these messages (e.g. XML/SOAP via the SSN web XML application and web services respectively).

The second application provides the graphical user interface (GUI) and handling user-to-business requests. It is the presentation layer providing the functionality of the EIS system.

The third application, in its essence, is acting as a protocol adapter, providing the functionality of the main application, over the communication protocol specified in the XML Reference Guide; the supported protocols are

- HTTP/XML; served by ssn-xmlprotocol-web application and
- HTTP/SOAP; served by ssn-xmlprotocol-ws module exposes the web services.

The decomposition of EIS in three applications allows for:

- The disengagement of the business logic of EIS from the protocol for which it is offered. This disengagement allows also for the independent implementation of the business logic of EIS.
- The development of the communication protocol (e.g. changes in the XML Schema that specify the structure of messages) has local repercussions in the corresponding application and not in the entire system (EIS).

The SSN system is identified by the following primary entities:

- "Message" as the primary artifact owned by the "Message Management" Business System.

- "Vessel" as the primary artifact owned by the "Vessel Management" Business System.

Thus, SSN system provides a service that enables the access to, and update of, these entities as shown in the Figure 4-2.

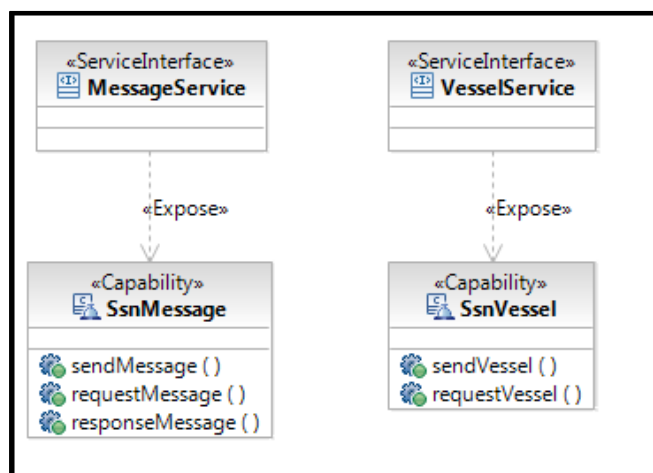


Figure 4-2 SSN Services provided operations

In addition, the IR service is implemented that extends the existing MessageService as shown in Figure 4-7.

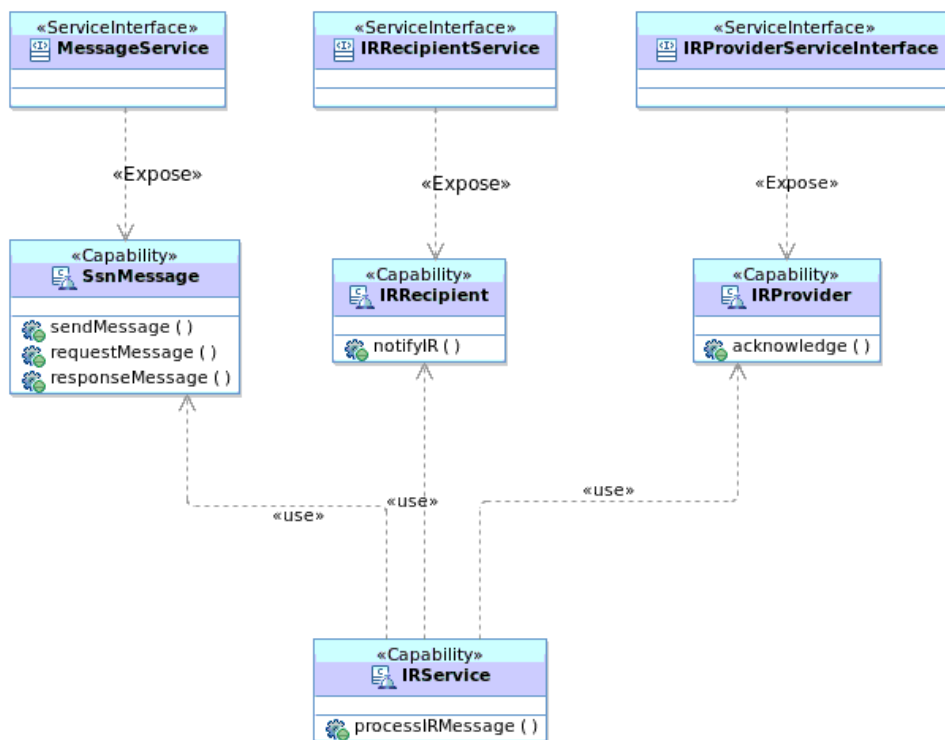


Figure 4-3 SSN Services provided operations.

Further to the core functionality of the system, independent business processes that carry out specific tasks are also deployed.

4.2 SSN Core Application - ssn-core-app

The diagram below depicts communication of SSN EIS application with database and jms resources.

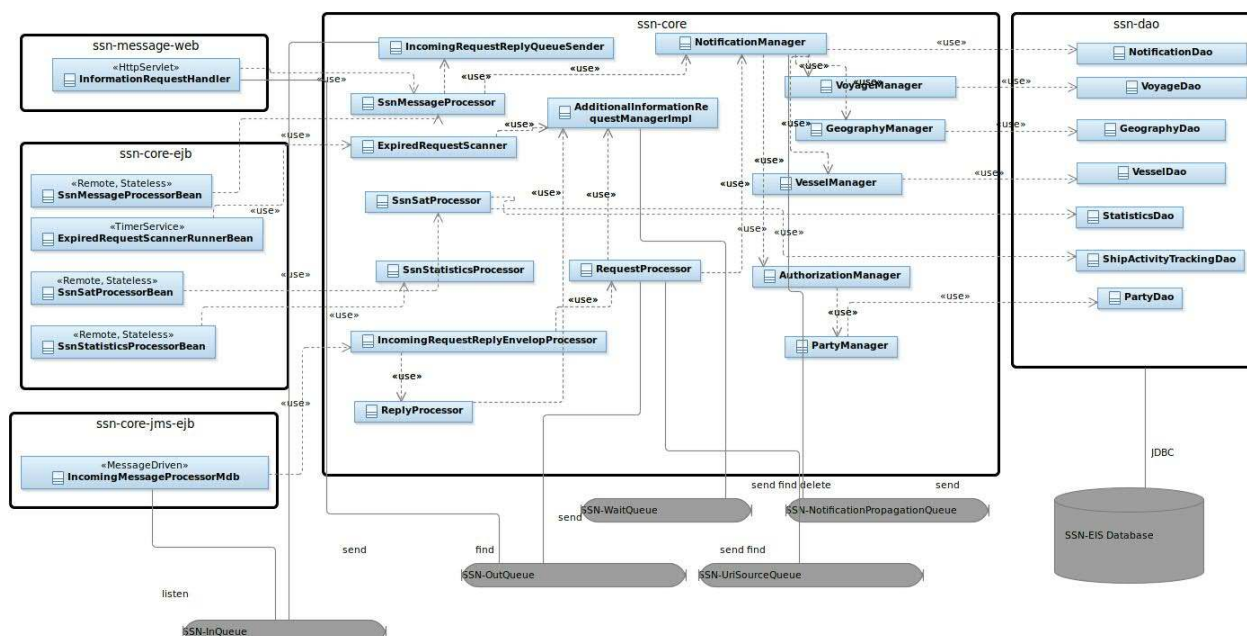


Figure 4-4 SSN Core Application - ssn-core-app

The application is constituted from the modules:

1. **ssn-domain**: SSN Business Domain Objects Modules. It includes the SSN Domain objects designed using Plain Java Classes and Interfaces. The SSN Domain objects encapsulate the state and behaviour of business entities. Examples of business entities in SSN application are Vessels, Locations, Organisations, Notifications, etc. This module is used by all the other modules of SSN-EIS.
2. **ssn-support**: This module contains classes that provide support for e-mail creation, http sending, logging, management of SSN JMS queues and validation of SSN objects. This module is also used by all the other modules of SSN-EIS.
3. **ssn-dao**: This module implements the EIS data access. It *decouples application code from data access code*.
4. **ssn-core**: It is the heart of application. This module implements the SSN functionality.
5. **ssn-core-ejb**: It is a bundle of lightweight EJB's that lend Remote Access semantics to the main services of ssn-core, as well as asynchronous queue listeners (Message Driven Beans) and time services.
6. **ssn-message-web**: It implements a REST-based, HTTP interface to the application domain of EIS, by processing SSN message information requests and returning JSON formatted responses.

4.2.1 SSN Core Main Components

4.2.1.1 SSN Domain module – ssn-domain

This module contains the domain model; the data object. This module is used by all the other modules of SafeSeaNet.

4.2.1.2 SSN Support module - ssn-support

This module contains classes that are used by all the other modules of SafeSeaNet. It provides support for e-mail creation, http sending, logging, management of SSN JMS queues and validation of SSN objects. These classes are used by all the other modules of SSN-EIS

The basic components are listed in Table 4-1.

	Component	Description
1.	AbstractEnvelopProcessorMdb	An abstract processor implements the JMS Message Listener interface (onMessage() method).It provides the asynchronous processing of SSN Queues'messages.
2.	EmailManager	It provides the SSN Email functionality.
3.	HttpUtilityApacheImpl	It provides the SSN Http functionality; it is used to send and receive XML messages via HTTP to/from data providers/requestors.
4.	WebServiceClient	It provides the SSN "web service client" functionality; it is used to send SOAP messages via HTTP to data requestors.
5.	SsnPropertyPlaceholderConfigurer	It loads the application parameters (stored on EIS database) specified in the bean's definition.
6.	DelegatingValidator	An abstract implementation of the validator interfaces that resolves the SSN validator according to the object to be validated.

Table 4-1 ssn-support

4.2.1.3 SSN Data Services module - ssn-dao

The basic components are listed in Table 4-3.

	Component	Description
1.	NotificationDao	Database data access object for the Notifications.
2.	AlertDistributionDao	Database data access object for the Incident Report Notifications.
3.	VoyageDao	Database data access object for the Voyages.
4.	ExemptionDao	A data access object for managing the database operations on Exemptions Information.
5.	PartyDao	Database data access object for the resolution of SSN users/authorities included in the messages.
6.	VesselDao	Database data access object for the resolution of Vessels included in the messages.
7.	GeographyDao	Database data access object for the resolution of Location codes included in the messages.
8.	StatisticsDao	A data access object for executing the statistical queries on EIS database.

Table 4-2 ssn-dao

4.2.1.4 SSN Core module - ssn-core

The basic components are listed in Table 4-3.

	Component	Description
1.	SsnMessageProcessor	<p>It is a Facade for the synchronous management of incoming messages (see also Sequence Diagram: Process Message-Notification & Request/Reply in section 4.2.2.3.4).</p> <p>The implementation simply delegates the actual message processing to one of the following processors:</p> <ul style="list-style-type: none"> - NotificationManager for the synchronous processing of notifications - IncomingRequestReplyQueueSender for the processing of request/ reply messages. <p>The SSN Message Processor synchronously replies with a ProcessResult on each incoming message processed.</p>
2.	ProcessResult	<p>It models the result of the aforementioned SSN processing – the SSN_Receipt message.</p>
3.	NotificationManager	<p>It provides the synchronous management of Notifications received as well as the possibility to retrieve Notification that satisfies the criteria of an Information Request.</p> <p>The following rules apply to the way SafeSeaNet handle the Notifications received:</p> <ul style="list-style-type: none"> - Notifications received will be recorded if technically correct. A technically correct notification must be a valid XML message, is compliant to the corresponding XSD, the reported vessel, in case a vessel is reported, and satisfies the existing rules for the validation of the vessel IMO number, MMSI number, Ship Name and Call Sign and finally the reported location code is technically correct meaning that it consists of a 2 letter country code (according to ISO 3166) followed by a 3 character code element for the location name that will normally comprise three letters or the numerals 2-9. - The vessel data are used as input to the vessel repository. Inserts and updates of the vessel data in the vessel repository must adhere to the existing rules for managing the reference repositories and on the XML reference guide. - The location codes reported in the Notifications received can be used as input to the locations repository. A new location that is technically correct and has not previously been defined will be inserted in the locations repository and will be labelled. The system considers the technically correct locations reported in the Notifications received when performing an Area search or when filtering locations by Country in the Web application. - All the Notifications with technically correct vessel data that are recorded in the system can be queried by the SSN users when requesting for Notification details.

	Component	Description
		<p>It processes the incoming notifications using the following logic (see also Sequence Diagram: Notification Processing - Valid Notification in section 4.2.2.3.5):</p> <ul style="list-style-type: none"> - The provided authorization manager is used to check the notification sender and his/her permissions. - The provided notification validator is used to validate the incoming notification. - The provided vessel manager is used to resolve the notification reported vessel data against to the vessel registry. - The provided geography manager is used to update the location registry with the technically correct locations reported in the Notifications received. - In case of PortPlus notification the provided notification dao used to update the notification registry if the notification refers to the same Ship Call (identified by a unique ShipCallId) (see also Sequence Diagram: Notification Processing - PortPlus Notification in section 4.2.2.3.5). - In case of Port, PortPlus, Ship and Hazmat notification the provided notification dao used to process the voyage calculation (see also Sequence Diagram: Notification Processing - Voyage Calculation in section 4.2.2.3.5). - The provided notification dao used to update the notification registry. - The provided alert distribution manager is used to manage the recipients - The provided notification propagation sender is used to notify the SSN users according to <ul style="list-style-type: none"> o ship activity tracking services; o alert functionality upon receipt of an xml Incident Report notification. <p>It actually sends a message to the "notification propagation" JMS queue using the SsnMessageQueueSender interface.</p>
4.	IncomingRequestReplyQueue Sender	<p>It provides the management of the incoming xml Request or Additional Information Reply.</p> <p>Actually, it sends the incoming xml request/reply messages to the Incoming Queue via/using the SSN message Queue Sender.</p>
5.	SsnMessageQueueSender	An interface responsible to provide access to the JMS queues.
6.	IncomingRequestReplyEnvelopeProcessor	It provides the asynchronous processing of incoming xml requests/replies. It delegates the request/reply message from the Incoming Queue to RequestProcessor/ ReplyProcessor accordingly.

	Component	Description
7.	RequestProcessor	<p>It provides the management of incoming Information Request. It processes the incoming request using the following logic (see also Sequence Diagram: Process Request Reply from EIS in section):</p> <ul style="list-style-type: none"> - The provided authorization manager is used to check the request sender and his/her permissions. - The provided notification manager is used to find whether there is a notification that satisfies the criterion of the given request. <ul style="list-style-type: none"> o If there is not such notification, an information not found reply is created and sent to the requestor via the provided outgoingQueueSender. o If there is a notification, the corresponding reply builder is being asked to decide whether the SSN can form a reply on its own or whether SSN has to send a request for additional information to the member state that had sent the notification o If the SSN can form on its own the reply, then the reply builder is called to perform this task, and the reply is sent to the requestor via the outgoingQueueSender. <p>Otherwise, a relevant AdditionalInformationRequest is created and sent to the data provider via the outgoingQueueSender and to the additionalInformationRequestManager.</p> <p>The following rules apply to the way SSN must handle the Requests for Notification details received with regards to the vessel criteria:</p> <ul style="list-style-type: none"> - Requests for notification details with search criteria the vessel IMO and or MMSI numbers; if IMO and MMSI are provided the search is performed based on both the criteria for a matching pair: IMO and MMSI should match. The system will execute the query in the vessels' registry. - The Exemptions Information is taken into account – via the provided exemption manager- in case of Requests for Hazmat notification details.
8.	outgoingQueueSender	It sends a message to the outgoing JMS queue using the SsnMessageQueueSender interface.
9.	AdditionalInformationRequest Manager	An interface responsible to provide access to the wait JMS.
10.	ReplyProcessor	<p>It provides the management of incoming replies from the data providers.</p> <p>It processes the incoming reply using the following logic (see also Sequence Diagram: Process Message- Reply in section4.2.2.3.4):</p>

	Component	Description
		<ul style="list-style-type: none"> - The additionalInformationRequestManager is used to find the waiting request made by SSN. - The appropriate reply builder – according to the initial data request – is used to create the reply. - The reply is sent to the requestor via the outgoingQueueSender - Finally, the additionalInformationRequest Manager is used to remove the processed the waiting request (additionalInformationRequest) from the wait queue.
11.	VesselManager	<p>It provides the management of vessels (ships) and the logic of updating the vessel details based on the data from the Vessel and Message Notifications.</p> <p>The “resolve processing” of the incoming notifications’ reported vessel data is the following (see also Sequence Diagram: Update Vessel Registry Using Reported vessel (IMO and MMSI Number, CallSign, Ship name and Flag in the case of PortPlus messages) in section 4.2.2.3.5):</p> <ol style="list-style-type: none"> a. for vessel that can be resolved – based on the rules defined for the vessel identification/validation procedure (refer also to CS-0202 Vessel V&V) – a reference (Foreign Key) will be made to the “Valid” or “Invalid” vessel; b. for a vessel that is not resolved– based on the rules defined for the vessel identification/validation procedure – a new “Temporary” vessel will be created and a reference (Foreign Key) will be made to that vessel. <p>The “Search Vessel” functionality, provided by the web console as the initial step of the “Send Notifications” interactive process, proposes ONLY the resolved – these are the vessels classified as “Valid”. If the user wishes to send a notification for a new vessel, not registered in the EIS OSD yet, the user is able to enter the Vessel Identification attributes (IMO and MMSI plus CallSign, ShipName and Flag in the case of PortPlus messages). On notification submit, the aforementioned “resolve processing” is executed to create a new Temporary vessel record.</p> <p>Similarly, the user is able – using the management console - to create a new vessel and /or update the vessel attributes (including the MMSI, CallSign, ShipName and Flag) identified by the IMONumber for a particular vessel version.</p>
12.	ExemptionManager	It provides the management of the Exemptions Information.
13.	ExpiredRequestScanner	<p>It offers the possibility of discovery and management of expired InformationRequest</p> <p>The “scan processing” is the following (see also Sequence Diagram: Process Expired Request from EIS in section0):</p> <ul style="list-style-type: none"> - The additionalInformationRequestManager is used to search the wait queue for expired messages; the time

	Component	Description
		<p>living in the queue of an expired message is greater than its timeout value.</p> <ul style="list-style-type: none"> - The expiredRequestProcessor is used to process the list of expired messages.
14.	ExpiredRequestProcessor	<p>Processes an expired request that SSN has sent to a data provider.</p> <p>It processes the expired messages using the following logic:</p> <ul style="list-style-type: none"> - An ExpiredRequestReply message is created to inform the requestor for the TimeOut occurred on the request processing. <p>The reply is sent to the requestor via the outgoingQueueSender</p> <ul style="list-style-type: none"> - Finally, the additionalInformationRequest Manager is used to remove the waiting request from the wait queue.
15.	AuthorizationManager	<p>It is a Facade for the authorisation of incoming xml messages. It uses the party manager</p>
16.	PartyManager	<p>It provides the management of system users' interfaces, and the user authorization/access rights in terms of permissions, location and source restrictions.</p>
17.	GeographyManager	<p>It provides the the logic of updating the location code repository with the location codes included in the Location and Message Notifications as well as location(s) and country/ies retrieval operations.</p>
18.	AlertDistributionManager	<p>It provides the management of Distributed Incident Report Notifications' recipients.</p>
19.	StatisticsManager	<p>It provides the management of EIS reporting.</p>
20.	SsnShipParticularsProcessor	<p>It scans the SHIP_PARTICULARS_SUB subscription tables for the information of ship particulars subscriptions.</p>
21.	SsnUserProcessor	<p>This class provides the system user functionality (Business Logic) to application management console; it delegates the entity processing to PartyManagerImpl class for CRUD user interface management.</p>
22.	SsnLocationProcessor	<p>This class provides location and country retrieval functionality to SSN EIS web consoles; it delegates the processing to GeographyManagerImpl class.</p>

Table 4-3 ssn-core

4.2.1.5 SSN Core EJB module - ssn-core-ejb

The basic components are listed in Table 4-4.

	Component	Description
1.	SsnMessageProcessorEJB	Stateless Session Bean acting as proxy for the SsnMessageProcessor from ssn-core.
2.	SsnVesselProcessorEJB	Stateless Session Bean acting as proxy for the SsnVesselProcessor from ssn-core.
3.	SsnUserProcessorEJB	Stateless Session Bean acting as proxy for the SsnUserProcessor from ssn-core.
4.	SsnLocationProcessorEJB	Stateless Session Bean acting as proxy for the SsnLocationProcessor from ssn-core.
5.	SsnStatisticsEJB	Stateless Session Bean acting as proxy for the StatisticsManager from ssn-core.
6.	ExpiredRequestScannerRunnerBean	Time programmed Stateless Session Bean that periodically calls (on ejbTimeout method) the ExpiredRequestScanner in ssn-core. The interval duration of the timer is a system parameter; the SSN_INTERVAL_DURATIONApplication parameter defined in EIS database.
7.	IncomingMessageProcessorMDB	Message Driven Bean that calls the IncomingRequestReplyEnvelopProcessor in ssn-core in correspondence to the asynchronous reception of message from the Incoming Queue.

Table 4-4 ssn-core-ejb

4.2.2 UML Class and Sequence Diagrams

This section covers the architectural significant elements of the design model. It presents the definition of the most significant classes that will implement the requested functionality, organised into packages.

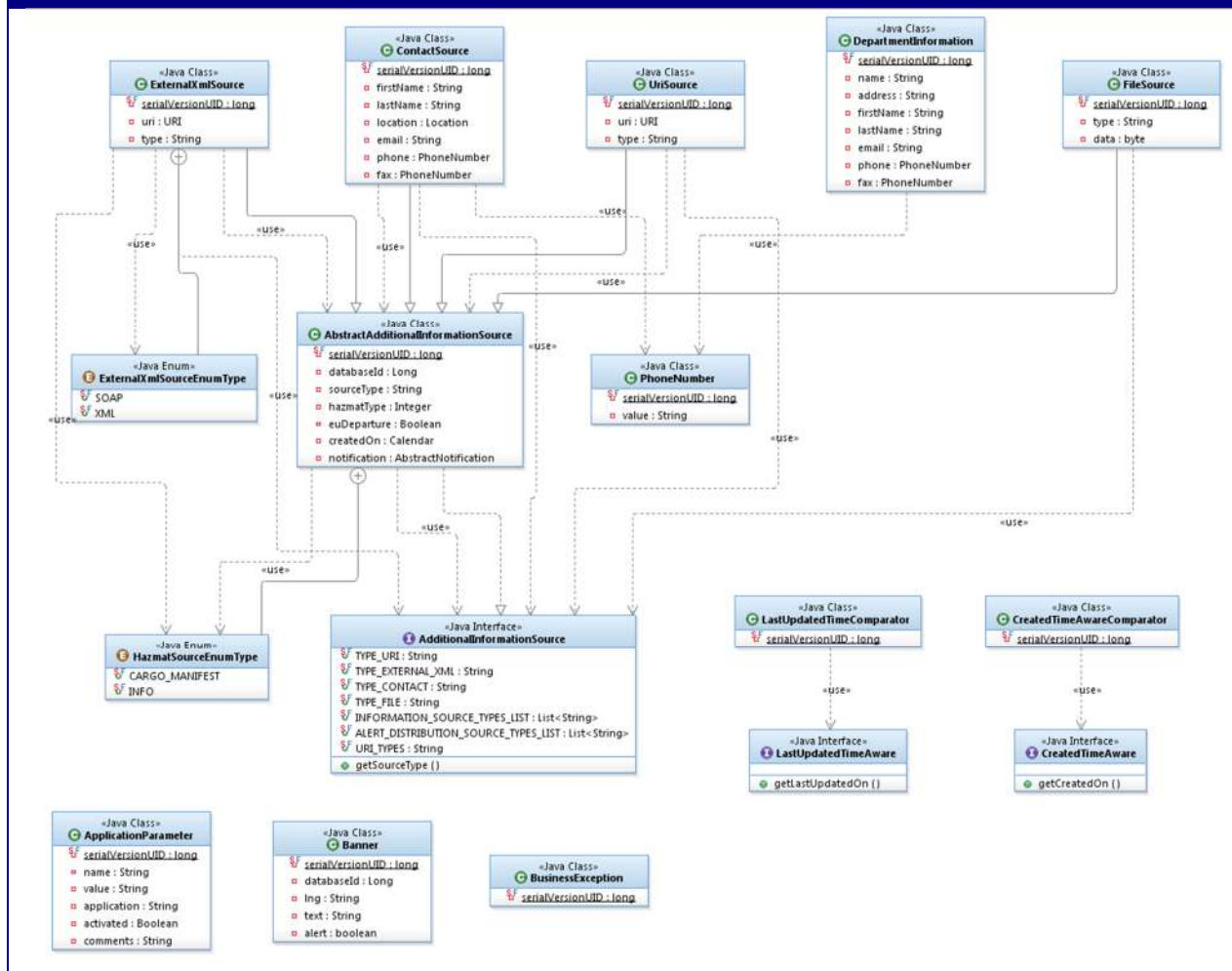
The UML Sequence Diagrams also presented in this section, associate classes and depict the overall flow of control within the system components.

The classes are organised in packages according to the functionality they provide. A package is a general-purpose model element that organizes model elements into groups. Each package contains a set of classes and interfaces, representing what will become components in the implementation.

4.2.2.1 Module: ssn-domain

4.2.2.1.1 Package: common

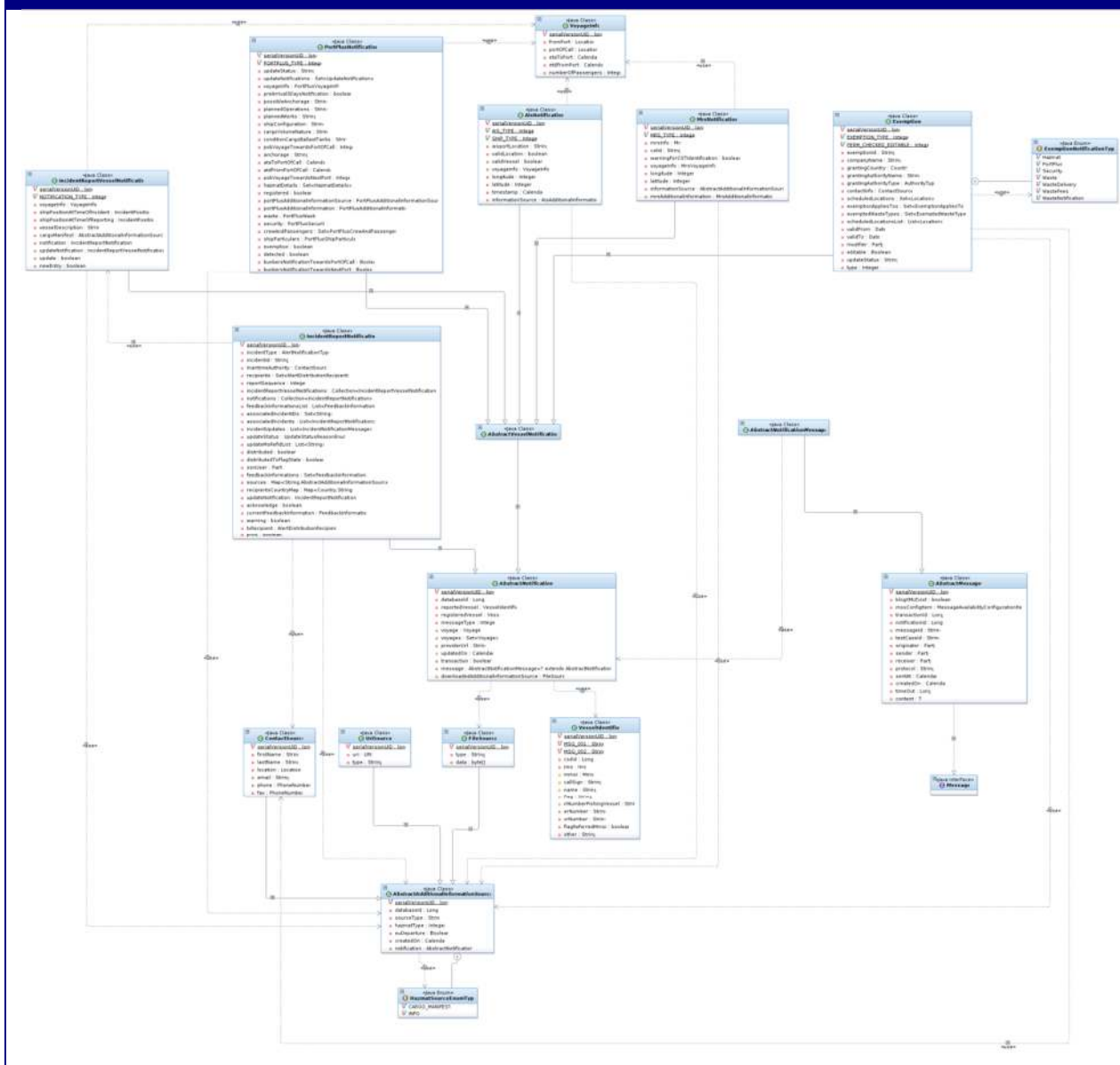
Class Diagram: common



Class	ApplicationParameter Represents an application parameter such as the Environment (Training, Production, etc).
Class	Banner Represents the Ticket Banner text displayed on the Web console header.
Class	BusinessException The super class of all SSN business exceptions.
Class	AbstractAdditionalInformationSource Abstract implementation of the AdditionalInformationSource interface.
Class	ContactSource Represents a contact person's contact details (phone, fax etc).

Class Diagram: common	
Class	PhoneNumber This class represents a phone or fax number.
Class	UriAdditionalInformation Represents the encoded document holding the notification details.
Class	UriSource Represents the URL of a Web site where additional details can be accessed and the type of document the details can be downloaded in.
Class	FileSource Represents Additional information source file.
Class	ExternalXmlSource Represents Additional information that the data provided makes available upon request.
Class	DepartmentInformation This class represents the information about the department of an organisation.
Interface	AdditionalInformation Defines an interface that has to be implemented by class that will process the contact details or the encoded document.
Interface	AdditionalInformationSource Defines an interface that has to be implemented by class that will decide the way details can be requested.
Interface	CreatedTimeAware An interface that should be implemented by persistent domain objects that want to know the time they have been created in the database.
Interface	LastUpdatedTimeAware An interface that should be implemented by persistent domain objects that want to know the last time they have been updated in the database.

Class Diagram: message & notification

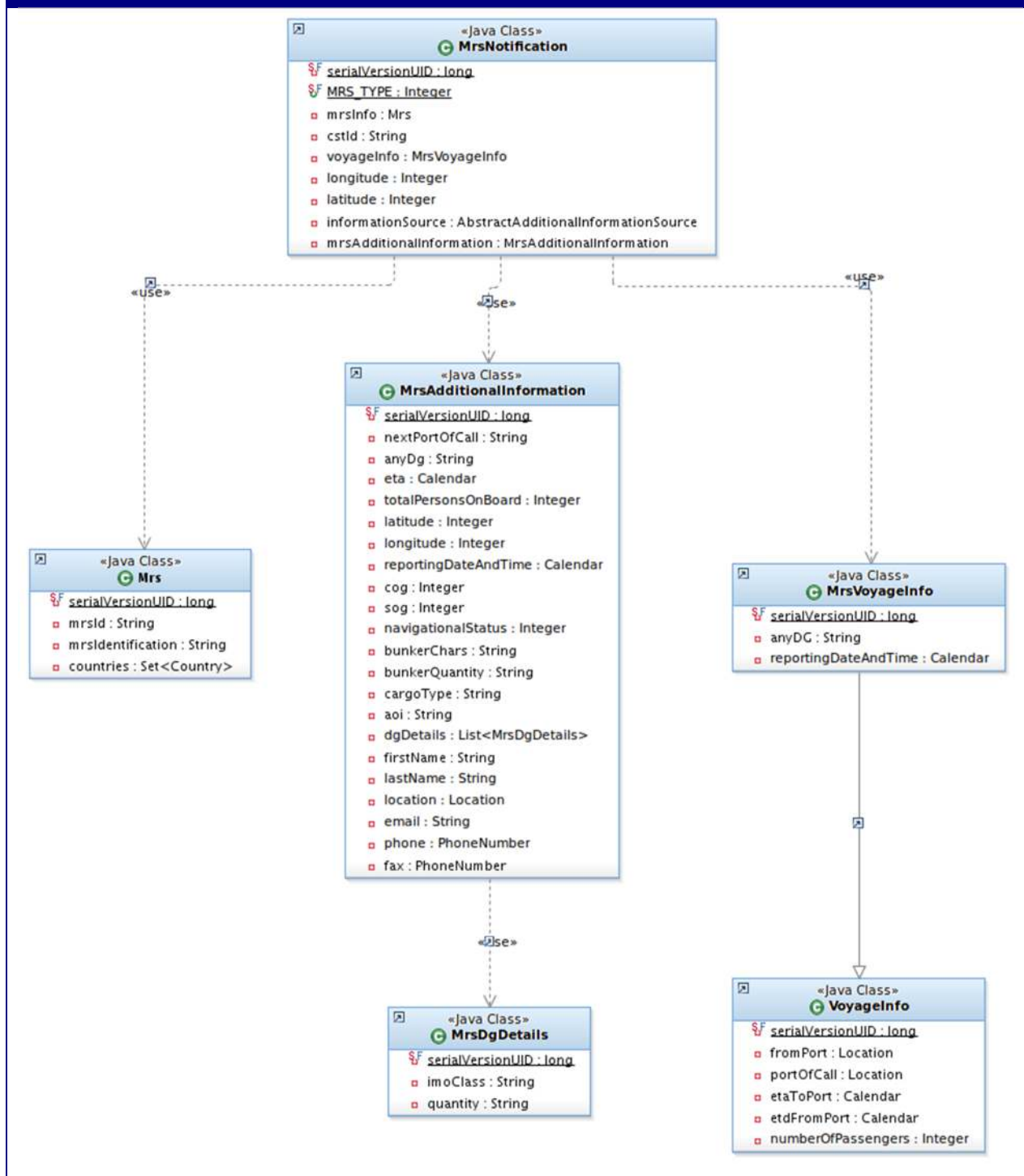


Interface	<p><u>Message</u></p> <p>This interface represents all the messages that Safe Sea Net exchanges between the member states. It provides the methods/getters of common message attributes such as the message reference identifier, the sender, the time it was sent. The template parameter <T> represents the message content, such as notification, request, response.</p>
Class	<p><u>AbstractMessage</u></p> <p>An abstract implementation of the message interface</p>
Class	<p><u>AbstractNotification</u></p> <p>The notification information sent to SafeSeaNet by a Member State specifies the reported vessel, the registered (resolved on OSD) vessel.</p>

Class Diagram: message & notification

Class	<p><u>AbstractNotificationMessage</u></p> <p>This class extends the aforementioned AbstractMessage class to provide the methods/getters of common message attributes such as the reported and registered vessel. The template parameter <T> represents the notification information.</p>
Class	<p>VesselIdentifier</p> <p>This class represents the ship particulars reported in a notification. A provider can send a notification identifying a vessel either using an IMO number and/or a MMSI number. Additional attributes include: ShipName, CallSign and the vessel Flag.</p>
Class	<p>FishingVesselIdentifier</p> <p>This class extends the aforementioned VesselIdentifier class to represent the fishing vessel. Additional attributes include: IRNumber.</p>
Class	<p>VoyageInfo</p> <p>It holds the voyage information of a specific vessel such as the last port, the port of call, the number of passengers on-board.</p>
Class	<p>AisNotification</p> <p>AisNotification is a type of the ShipNotification.</p> <p>The notification is sent by a Member State to SafeSeaNet in order to notify SafeSeaNet about a vessel's voyage and cargo information. The ship notification in this case was originally captured via an AIS signal.</p>

Class Diagram: Mrs notification



Class

MrsNotification

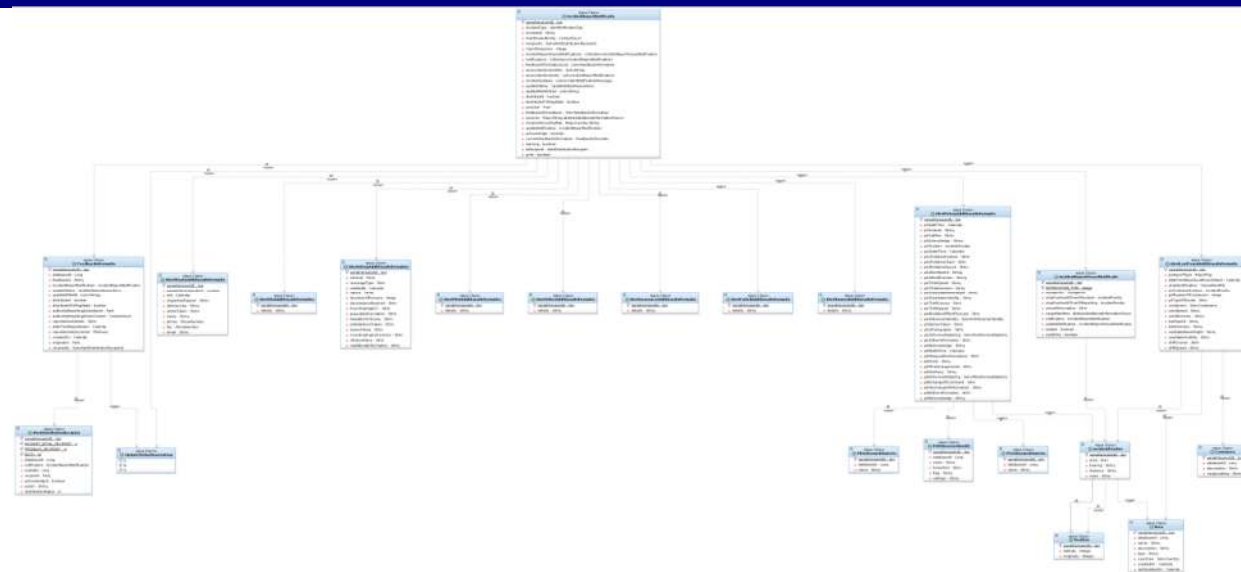
MrsNotification is a type of the ShipNotification.

The notification is sent by a Member State to SafeSeaNet in order to notify SafeSeaNet about a vessel's voyage and cargo information. The ship notification in this case was originally captured via an MRS signal.

Class Diagram: Mrs notification

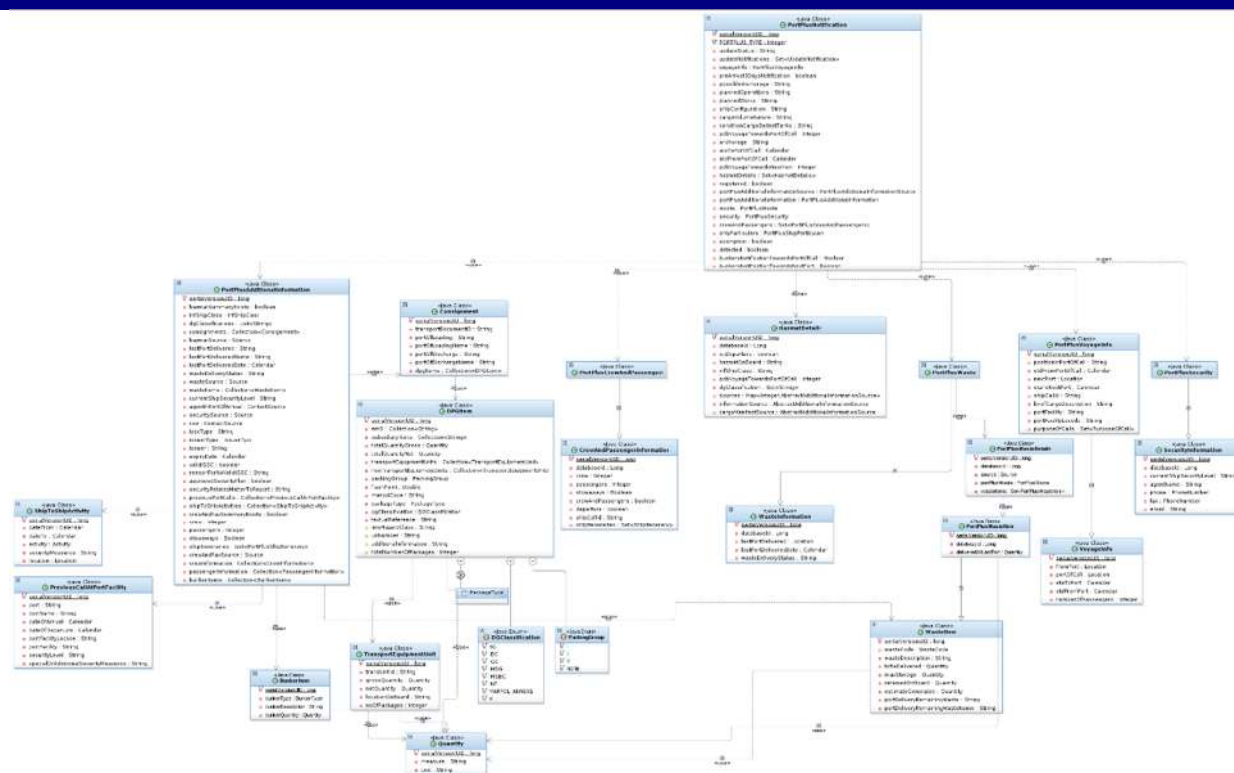
Class	MrsAdditionalInformation The additional information of a ship (MRS) notification. This additional information is submitted by a Member State to SafeSeaNet as a reply to a previous corresponding request from the SafeSeaNet.
--------------	--

Class Diagram: Incident Report notification



Class	<u>IncidentNotificationMessage</u> This class extends the aforementioned AbstractNotificationMessage generic to represent the Incident Report notification message.
Class	IncidentReportNotification It represents the information of Incident Report notification such as the incident ID, the report sequence, the status, the incident details document – if any. Additional attributes include: the collection of the associated incident reports, the collection of the feedback information.
Class	IncidentReportVesselNotification It extends the aforementioned AbstractNotification class to represent the information of new Incident Report notification per associated vessel such as the vessel voyage information, the vessel positions at time of incident and reporting, the vessel cargo manifest.
Class	IncidentPosition It holds the position information of a specific vessel such as the position coordinates, the area, the bearing distance.
Class	FeedbackInformation It holds the feedback information of new Incident such as the authority reporting this action, the action details, the list of recipients.

Class Diagram: PortPlus notification



Class

PortPlusNotification

The notification is sent by a MemberState to SafeSeaNet in order to notify SafeSeaNet in case of

- 72h pre-arrival Notification
- 24h pre-arrival Notification
- Arrival Notification
- Departure Notification
- Hazmat information; a set of HazmatDetails class that actually can include two items, one for HazmatEUDeparture and a second for NonEUDeparture.
- Crew and Passengers information.
- Security information.
- Waste information; it is upgraded, the waste details added.
- Flags for existence of Bunkers for "port of call" and "next port".

The very 1st PortPlus notification received by SSN-EIS for a new ship call will create a new instance of a PortPlus notification in the SSN-EIS database. Every other PortPlus notification for the same ship call will not create a new instance of the PortPlus notification. Instead it will update the previously defined instance. This way the system consolidates automatically at notification process all the PortPlus notification data that refer to the same ship call. As such the system can directly service any request for details concerning a given ship call.

The PortPlus notification update process depends on the *NotificationStatus* element *UpdateStatus* attribute. Permitted values are:

Class Diagram: PortPlus notification

- *UpdateStatus="N"*
- *UpdateStatus="U"*

The initial PortPlus notification for one ship call is expected with *UpdateStatus="N"*.

Every consecutive PortPlus notification for the same ship call shall have *UpdateStatus="U"*.

UpdateStatus rules include:

1. A ship call is uniquely identified by the *ShipCallId* attribute value. The *ShipCallId* is assigned by the data provider and in SSN-EIS must be unique per ship call per data provider. Shall the SSN-EIS system receive a PortPlus from the same data provider with an existing *ShipCallId* but for another vessel (identified by another *IMONumber* or another *MMSINumber* and *IMONumber* NULL) the message will be rejected with *StatusCode="InvalidFormat"* and *StatusMessage="A ShipCallId has already been sent from user_id_xyz"*.
2. Shall the SSN-EIS system receive 2 or more PortPlus for the same ship call with *UpdateStatus="N"* then the 1st PortPlus will be accepted while all the others will be rejected.
3. Shall SSN-EIS receive a PortPlus with *UpdateStatus="N"* or *UpdateStatus="U"* with *PreArrival3Days* and/or *PreArrivalNotification24Hours* and/or *Arrival* and/or *Departure* notification details the message is valid and the update rules specified hereunder will be considered.
4. Shall the SSN-EIS system receive a PortPlus with *UpdateStatus="U"* then it should update the values of attributes – defined previously – by the values in the new PortPlus.

To cancel a value previously sent the data provider can send a PortPlus with the attribute value being NULL (e.g. *PossibleAnchorage=""*). In this case the previous value of *PossibleAnchorage* will be deleted in the database.
5. Shall the SSN-EIS system receive a PortPlus with *UpdateStatus="U"* for a new ship call (prior to the PortPlus with *UpdateStatus="N"*) then the system must process the PortPlus. If at a later stage the system receives a PortPlus for the same ship call with *UpdateStatus="N"* the values of common attributes with previous PortPlus for the same ship will be those of the previous PortPlus and not of the latest one with *UpdateStatus="N"*.

The system, upon receipt of the PortPlus with *UpdateStatus="U"* prior to the PortPlus with *UpdateStatus="N"* shall alert the data provider – instead of using an email - by the *SSN_Receipt* with *StatusCode="OK"* and *StatusMessage="The original PortPlus notification must be send"*.

Class Diagram: PortPlus notification

Class	<p>CrewAndPassengerInformation</p> <p>Attributes described bellow:</p> <p>departure: Flag indicating whether the person on board information is for an arrival or for a departure notification.</p> <p>passenger: Total number of passengers</p> <p>crew: Total number of crew</p> <p>stowaways: Flag indicating whether the ship call reported any stowaways.</p>
Class	<p>HazmatDetails</p> <p>This class contains information that indicates whether the ship carries hazardous material. It consists of</p> <ul style="list-style-type: none"> – euDeparture flag; true for HazmatEUDeparture and NonEUDeparture otherwise; – HazmatSummary provides INFShipClass and the list of DGClassification; – HazmatDetails provides the Source and CargoInformation consists of a list of Consignments.
Class	<p>Consignment</p> <p>This class represents the Consignment information that includes</p> <ul style="list-style-type: none"> – TransportDocumentID; – Port ofLoading; – Port ofDischarge – The list of DPGItem.
Class	<p>DPGItem</p> <p>This class represents the DPGItem information that provides</p> <ul style="list-style-type: none"> – DGClassification; – TextualReference; – IMO HazardClass; – UN Number; – PackageTypethat holds EDIFACT codes (7065); – FlashPoint; – MarpolCode; – TotalNumber ofPackages; – AdditionalInformation; – The list of EmSNumbers; – The list of SubsidiaryRisks; – TotalQuantityGross and TotalQuantity Net; – The lists of TransportEquipmentUnit and NonTransportEquipmentUnit

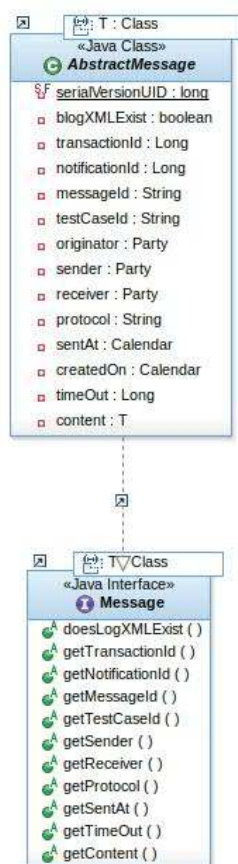
Class Diagram: PortPlus notification

Class	TransportEquipmentUnit This class includes information for <ul style="list-style-type: none"> – transUnitId – locationOnBoard – noOfPackages – gross and net Quantities.
Class	Security Security represents the generic CertificateType information provided by the NSW. Attributes described bellow: issuingAgency: Name of agency which issued the International Ship Security Certificate (ISSC) currentShipSecurityLevel: Ship's current security level according to the ISPS standard cso: The company security officer shipToShipActivities: The list of ship to ship activities previousCallAtPortFacility: The list of previous calls.
Class	ShipToShipActivity This data type contains a list with the description of recent ship-to-ship activities and any security measures applied during these activities. The description shall contain the time and date when activity started and ended, the position and/or location (at least one), what activity was performed and if any special security measures were taken. Attributes: from/to/activity/additionalSecurityMeasures/location/ reportedLocationName/reportedPosition
Class	PreviousCallAtPortFacility This data type contains a list with the <ul style="list-style-type: none"> – Port name – Date of Arrival and Departure – Security Level and Special orAdditionalSecurityMeasures – The port facility.

Class Diagram: PortPlus notification

Class	<p>Waste</p> <p>This class contains information that shall be sent to a port in conjunction with an arrival.</p> <p>Attributes described below.</p> <p>lastPortDelivered: Last port where ship-generated waste was delivered. Name of the port where the generated waste of the ship was discharged.</p> <p>lastPortdeliveredDate: Last date when ship-generated waste was delivered Date indicating when the last disposal has been conducted.</p> <p>wasteDeliveryStatus: If ship delivers all, some or none of its waste in the port it reports to.</p>
Class	<p>PortPlusWasteItem</p> <p>This class extends the Waste class to provide the "Delivered at last port" quantity.</p>
Class	<p>PortPlusWasteDetails</p> <p>This class provides a set of PortPlusWasteItem for a specific source on PortPlus notification.</p>
Class	<p>WasteType</p> <p>Type of waste. The code shall be the one defined in Annex B of [R4]. In addition, the proper shipping name is required for codes 504 (cargo residues) and all types of waste in category 2 (NLS).</p> <p>Otherwise, the text description of waste is optional.</p> <p>Attributes: Code/Description</p>
Class	<p>BunkerItem</p> <p>The class represents one BunkerItem per Bunker Type on board; the Bunker information submitted by the ShipCall providers consists of a 'BunkerItems' collection. The BunkerItem includes the type, the description and the quantity information.</p>

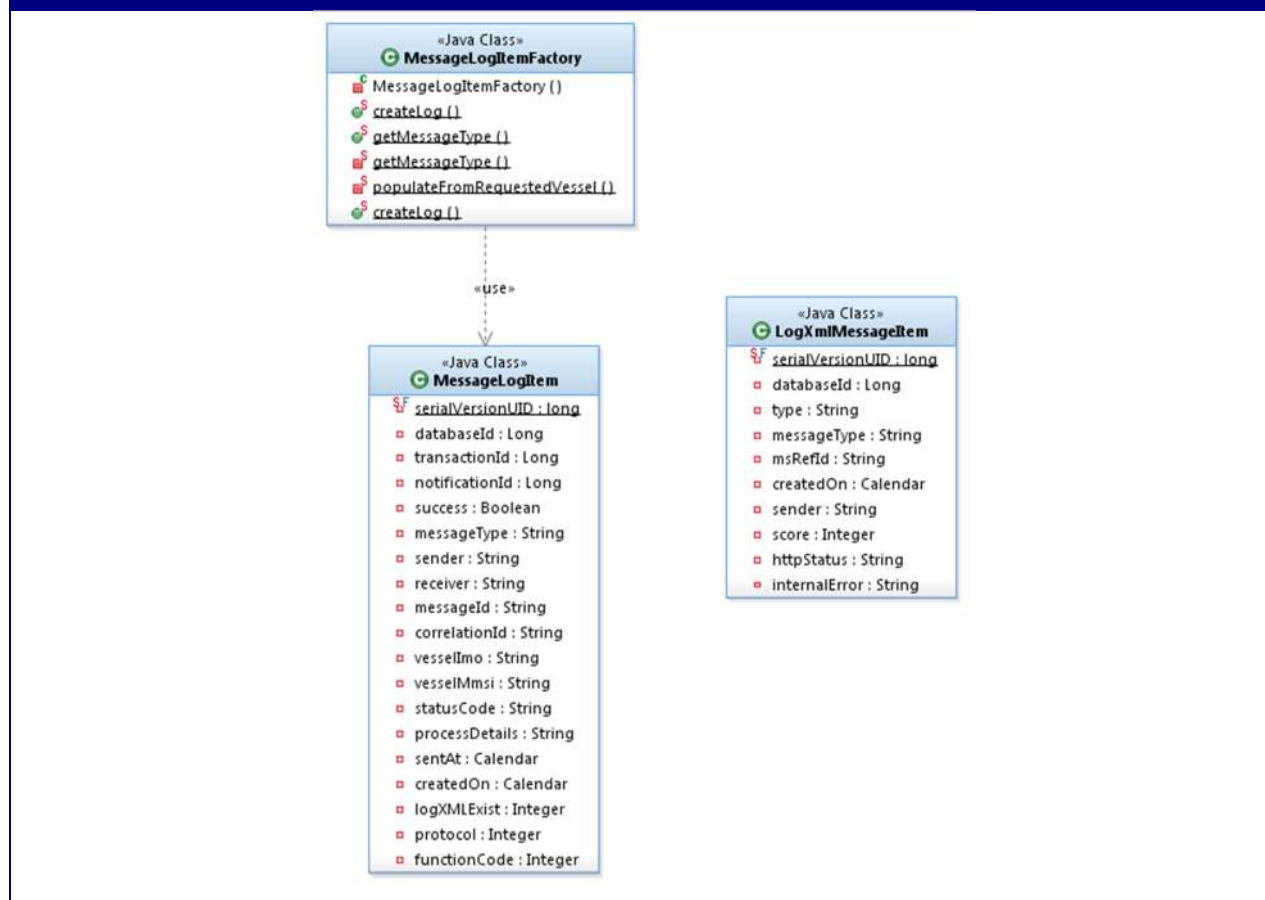
Class Diagram: message



Class	<u>AbstractMessage</u> An abstract implementation of the message interface
Interface	<u>Message</u> Provides the methods/getters of message attributes.

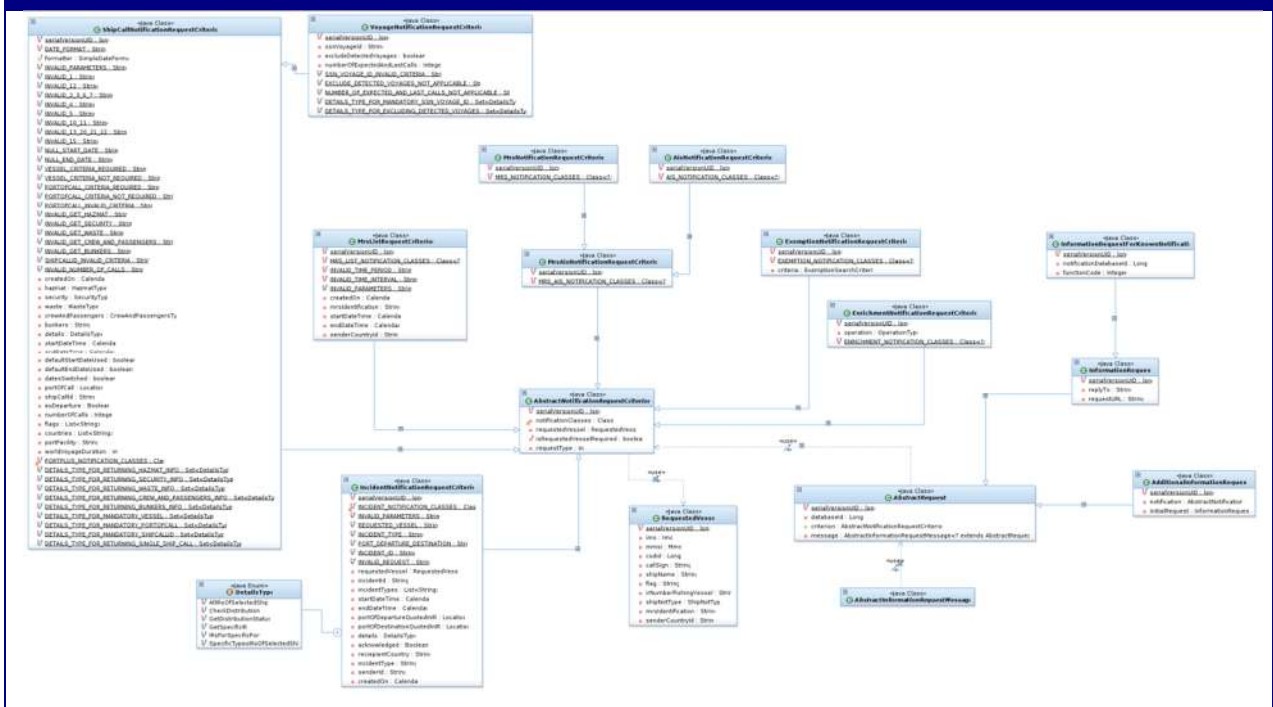
4.2.2.1.3 Package: message-log

Class Diagram: message-log



Class	<u>MessageLogItem</u> This class represents the entries logged on EIS database for the incoming messages (notification, request, reply) and the related process results. The attribute named <code>transactionId</code> identifies the Request/Response monitoring.
Class	<u>LogXmlMessageItem</u> This class represents the entries logged on EIS database for the incoming xml messages (MS notifications, requests, replies) and the related SSN xml responses (SSN replies/receipts).
Class	<u>MessageLogItemFactory</u> An abstract factory class used to create the message log items.

Class Diagram: request

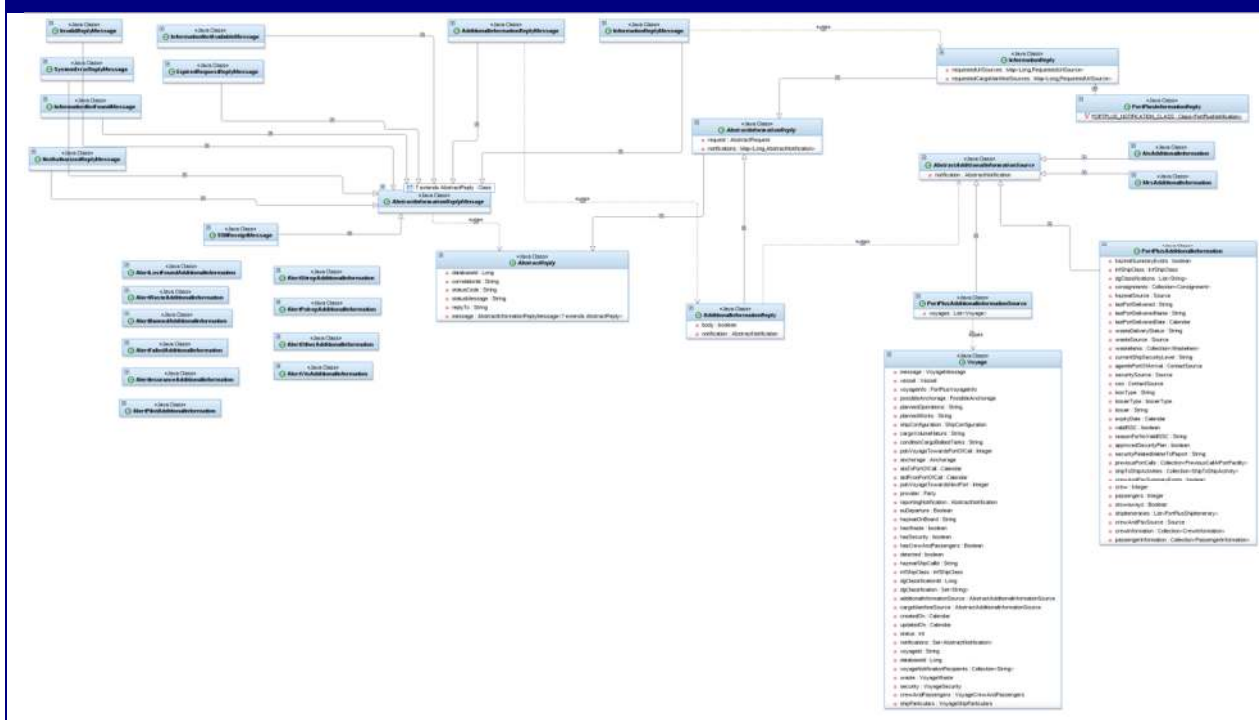


Class	AbstractInformationRequestMessage This abstract template extends the aforementioned AbstractMessage to represent the request sent to SafeSeaNet by a MemberState and vice-versa, and it specifies the message header, such as, the sender, the time it was sent.
Class	AbstractRequest This abstract class defines the content of the aforementioned template; it holds the request criterion.
Class	InformationRequest It implements the aforementioned abstract class (AbstractRequest) for the request sent to SafeSeaNet by a MemberState.
Class	InformationRequestForKnownNotification Extends InformationRequest for a request submitted by the ssn-find-notification-console where the notification is already resolved.
Class	AdditionalInformationRequest It implements the aforementioned abstract class (AbstractRequest) for the request sent by SafeSeaNet to the data provider member state. The request for additional information details is related to a notification sent previously to SafeSeaNet by the data provider that is expected to service the request with a response message.
Class	AbstractNotificationRequestCriterion This abstract class includes the request criteria for identifying the vessel.

Class Diagram: request	
Class	RequestedVessel This class represents the vessel for which an Information Request is made. A requestor can send a request identifying a vessel either using the CSDID or IMO number, MMSI number, call sign, ship name, ship flag, etc
Class	MrsAisNotificationRequestCriterion This class extends <i>AbstractNotificationRequestCriterion</i> to implement request criterion for MRS or AIS notifications.
Class	AisNotificationRequestCriterion This class extends <i>MrsAisNotificationRequestCriterion</i> to implement request criterion for AIS notifications.
Class	MrsNotificationRequestCriterion This class extends <i>MrsAisNotificationRequestCriterion</i> to implement request criterion for MRS notifications.
Class	MrsListRequestCriterion This class extends <i>AbstractNotificationRequestCriterion</i> to implement list request criterion for MRS notifications.
Class	IncidentNotificationRequestCriterion This class extends <i>AbstractNotificationRequestCriterion</i> to implement request criterion for Incident notifications; it includes additional criteria for Incident notifications attributes, such as, incident id, port of destination and departure, etc.
Class	ShipCallNotificationRequestCriterion This class extends <i>AbstractNotificationRequestCriterion</i> to implement request criterion for PortPlus notifications and correlated Voyage information; it includes additional criteria for voyages' attributes, such as, shipcall id, port of call, hazmat/security/waste/bunker types, etc.
Class	VoyageNotificationRequestCriterion This class extends <i>ShipCallNotificationRequestCriterion</i> to implement request criterion for PortPlus notifications and correlated Voyage information provided to SEG/ES system; it includes additional criteria, such as, interface id, originator id, as well as the new criteria of SSN Voyage Id and last port.
Class	EnrichmentNotificationRequestCriterion This class extends <i>AbstractNotificationRequestCriterion</i> to implement request criterion for enrichment information (active voyages, incidents and MRS notifications).
Class	ExemptionNotificationRequestCriterion This class extends <i>AbstractNotificationRequestCriterion</i> to implement request criterion for exemption notifications.

4.2.2.1.5 Package: reply

Class Diagram: reply



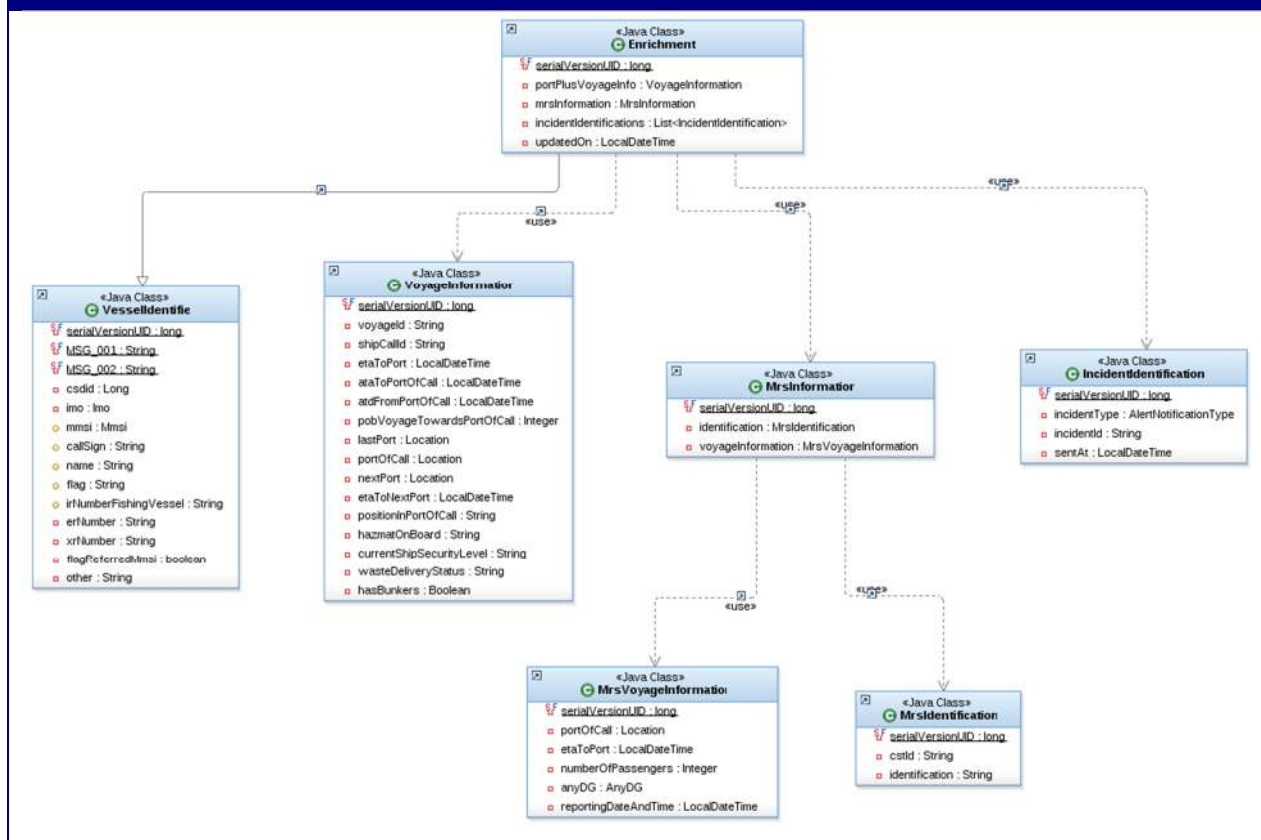
Class	AbstractInformationReplyMessage This abstract template extends the aforementioned AbstractMessage to represent the reply sent from SafeSeaNet to a MemberState and vice-versa, and it specifies the message header, such as, the sender, the time it was sent.
Class	AbstractReply This abstract class defines the content of the aforementioned template; it holds the header attributes of the reply message, such as, the correlation id, the reply status and the reply URL.
Class	AbstractInformationReply This class is the common parent of all classes that implement discrete reply messages; it includes the request message and the list of notifications include the reply information.
Class	InformationReply Represents a reply sent by SSN to MS; it consists of the information included in the notifications stored in SSN database and the additional information (described below) received from data provider.
Class	AdditionalInformationReply Represents a reply from a data provider to SSN; the following classes extends this one to represent the information of each notification type; for presentation purposes the associations between the alert specific information are eliminated.
Class	AisAdditionalInformation The additional information of a ship (AIS) notification.

Class Diagram: reply	
Classes	<p>Incident specific information per incident type</p> <p>AlertSitrepAdditionalInformation</p> <p>AlertPolrepAdditionalInformation</p> <p>AlertLostFoundAdditionalInformation</p> <p>AlertWasteAdditionalInformation</p> <p>AlertOtherAdditionalInformation</p> <p>AlertBannedAdditionalInformation</p> <p>AlertFailedAdditionalInformation</p> <p>AlertInsuranceAdditionalInformation</p> <p>AlertPilotAdditionalInformation</p> <p>AlertVtsAdditionalInformation</p>
Class	<p>MrsAdditionalInformation</p> <p>The additional information of a ship (MRS) notification.</p>
Class	<p>PortPlusAdditionalInformation</p> <p>The additional information of a PortPlus notification sent by the data provider; it includes</p> <ul style="list-style-type: none"> – The voyage information; – The crew and passengers information; – The Hazmat information; – The Security information; – The Waste information; – (upgraded to also provide) the Bunkers information.
Class	<p>PortPlusAdditionalInformationSource</p> <p>The voyage information retrieved from SSN database and included in replies of ShipCall requests.</p>
Class	<p>SSNReceiptMessage</p> <p>This information is submitted by a MemberState to SafeSeaNet as a reply to a previous corresponding request from the SafeSeaNet.</p>
Exceptional Replies	
Class	<p>NotAuthorisedReplyMessage</p> <p>Represents response messages send by SSN to the MS in reply to a request by a non-authorised party.</p>
Class	<p>ExpiredRequestReplyMessage</p> <p>Represents response messages send by SSN to the MS in reply to a timed out request.</p>
Class	<p>InformationNotFoundReplyMessage</p> <p>Represents response messages send by SSN to the MS in reply to a request for which no data was found.</p>

Class Diagram: reply	
Class	InformationNotAvailableReplyMessage Represents response messages send by SSN to the MS in reply to a request for which no reply was received from the data provider.
Class	SystemErrorReplyMessage Represents response messages send by SSN to the MS when a system error occurred during the request processing.

4.2.2.1.6 Package: enrichment

Class Diagram: enrichment



Class	Enrichment Enrichment class extends VesselIdentifier class (4.3.2.1.1) and provides information for active voyages, incidents and MRS notifications of valid vessels.
Class	VoyageInformation It provides information for active voyage such as VoyageID, port of call, the last port, the next port, the estimated time when the vessel will arrive to the port, the actual time of arrival at port of call, the actual time of departure from port of call, the number of passengers on-board the vessel, the actual position in port of call, a flag indicates whether the ship carries hazardous material, the current ship security level and waste delivery status, flag for Bunkers' existence.
Class	MrsInformation It provides information for MRS notification; MrsIdentification and MrsVoyageInformation described below.
Class	MrsIdentification It provides the Coast and MRS identification.
Class	MrsVoyageInformation It extends Position class (4.3.2.1.2) and provides information for MRS notification such as the portof call, the number of passengers on-board the vessel, the estimated time when the vessel will arrive to the port, the reporting date and AnyDG.

Class Diagram: enrichment

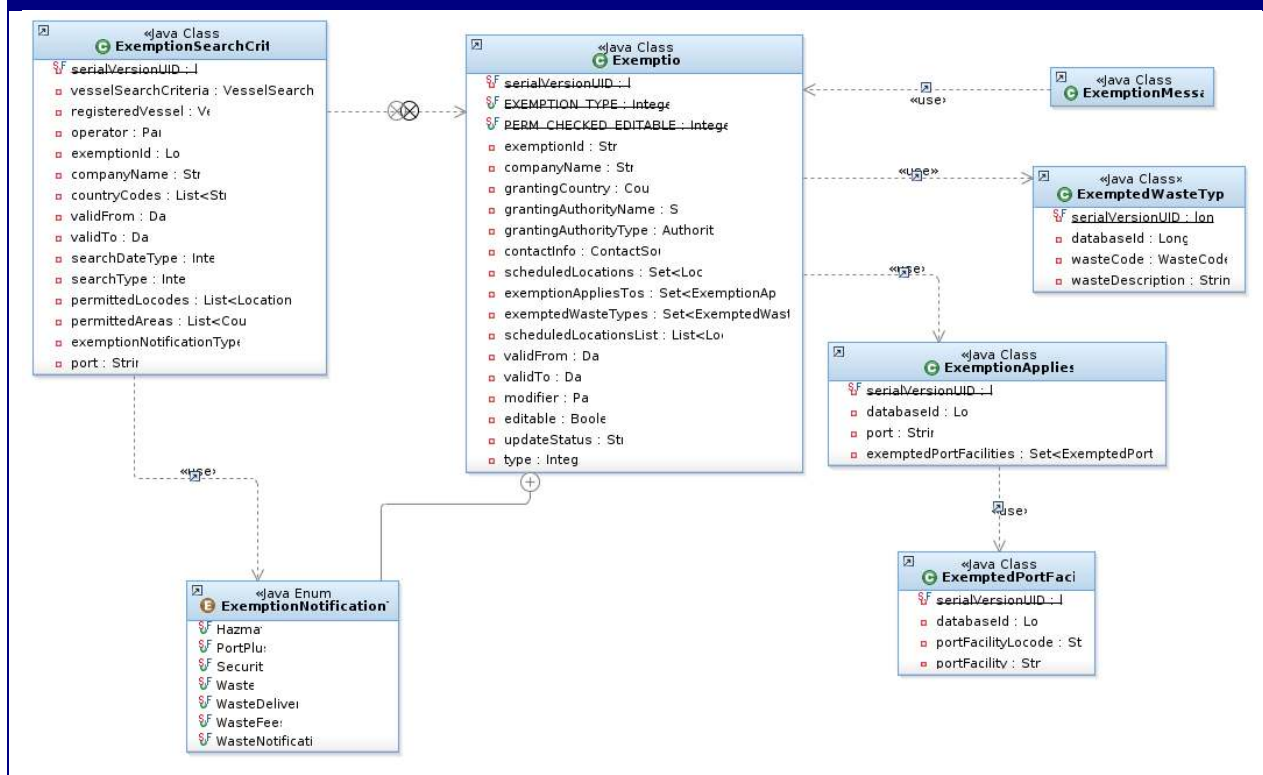
Class

IncidentIdentification

It provides the Incident identification information such as the Incident Type and Id.

4.2.2.1.7 Package: exemption

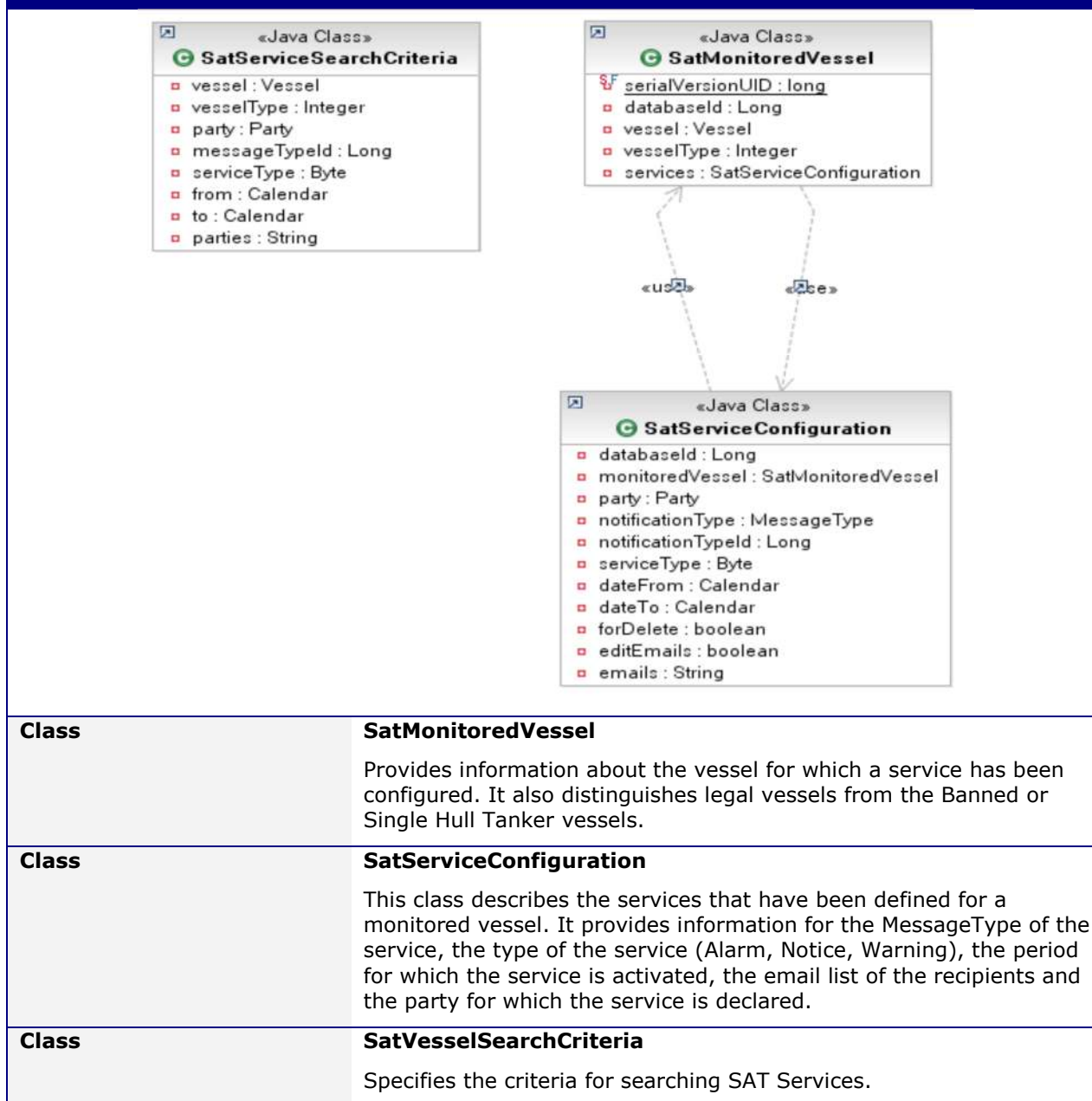
Class Diagram : exemption



Class Diagram : exemption	
Class	Exemption Describes information about exemptions defined per vessels. The exemption information includes: <ul style="list-style-type: none"> – A Unique identifier preceded by the MS 2-letter country code; – The type of the exemption, i.e. Pre-arrival, hazmat, waste, security; – The exemption validity period; – The company name; – The granting country; – The granting authority type, i.e. National Competent Authority, Port Authority, Other; – The contact info; – The scheduled/route locations; – The set of ports and port facilities where this exemption applies to; – The set of the waste types where this exemption applies to.
Class	ExemptionSearchCriteria Specifies criteria for searching exemptions.
Class	ExemptionMessage The implementation of the exemption message

4.2.2.1.8 Package: sat (Ship Activity Tracking)

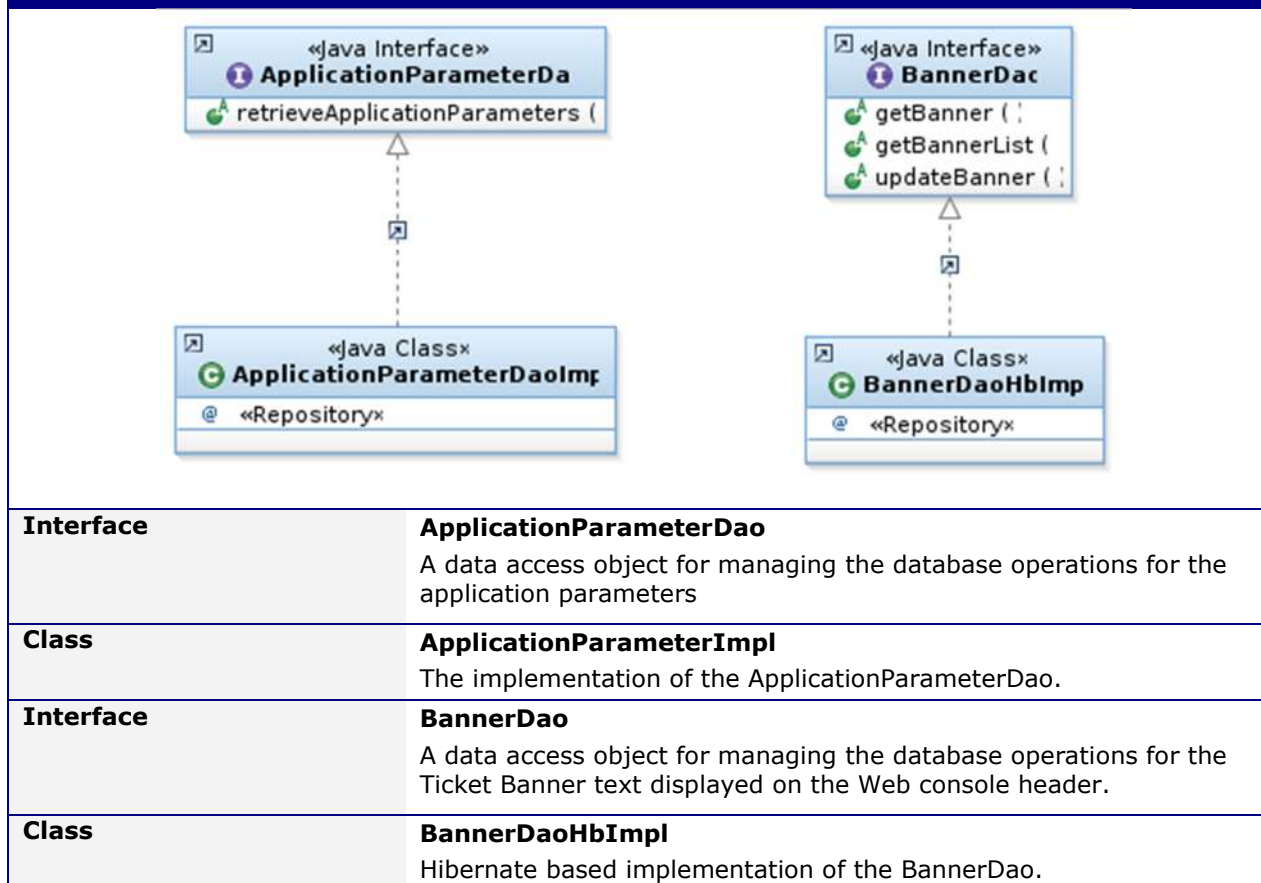
Class Diagram: sat



4.2.2.2 Module ssn-dao

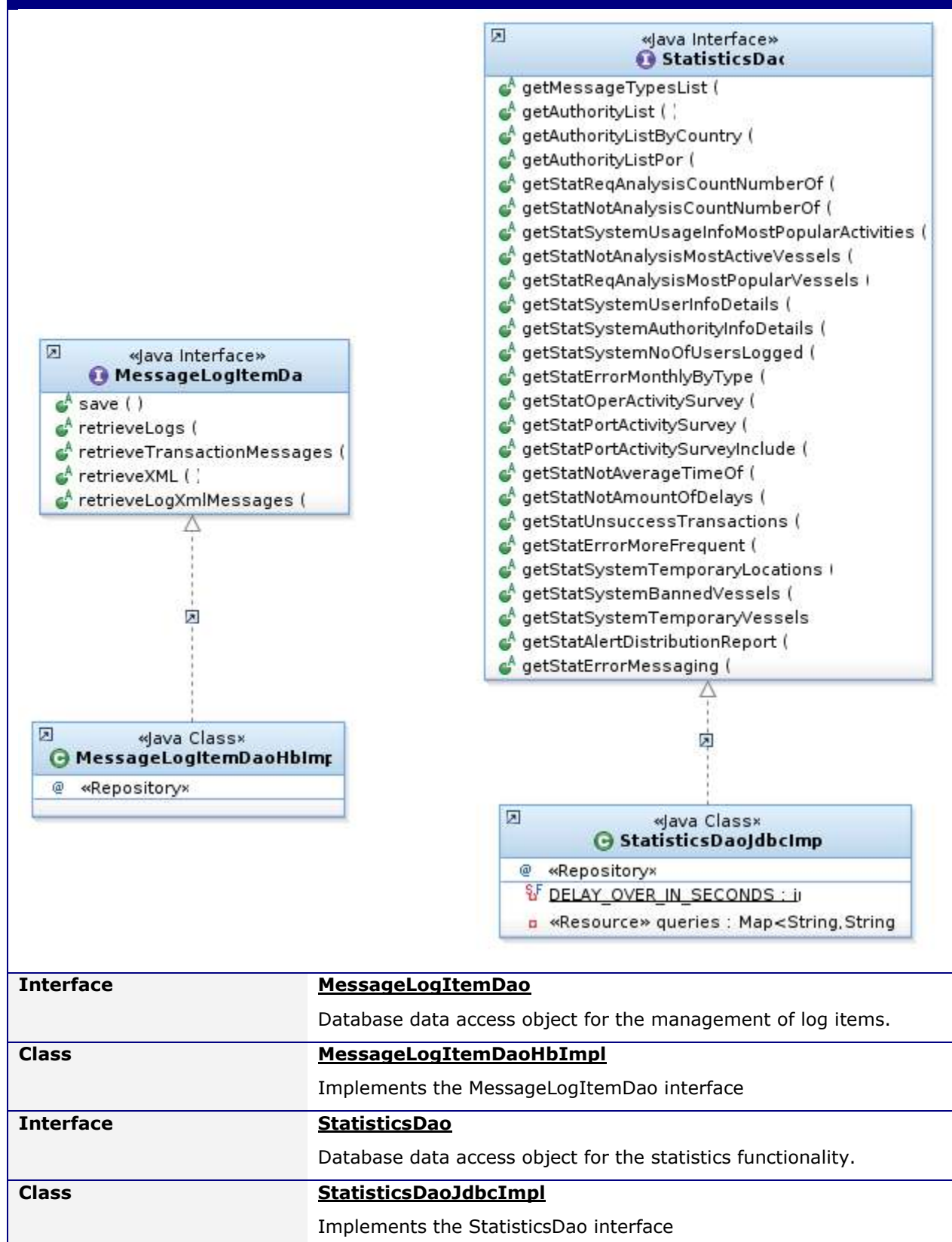
4.2.2.2.1 Package: common

Class Diagram : common-dao



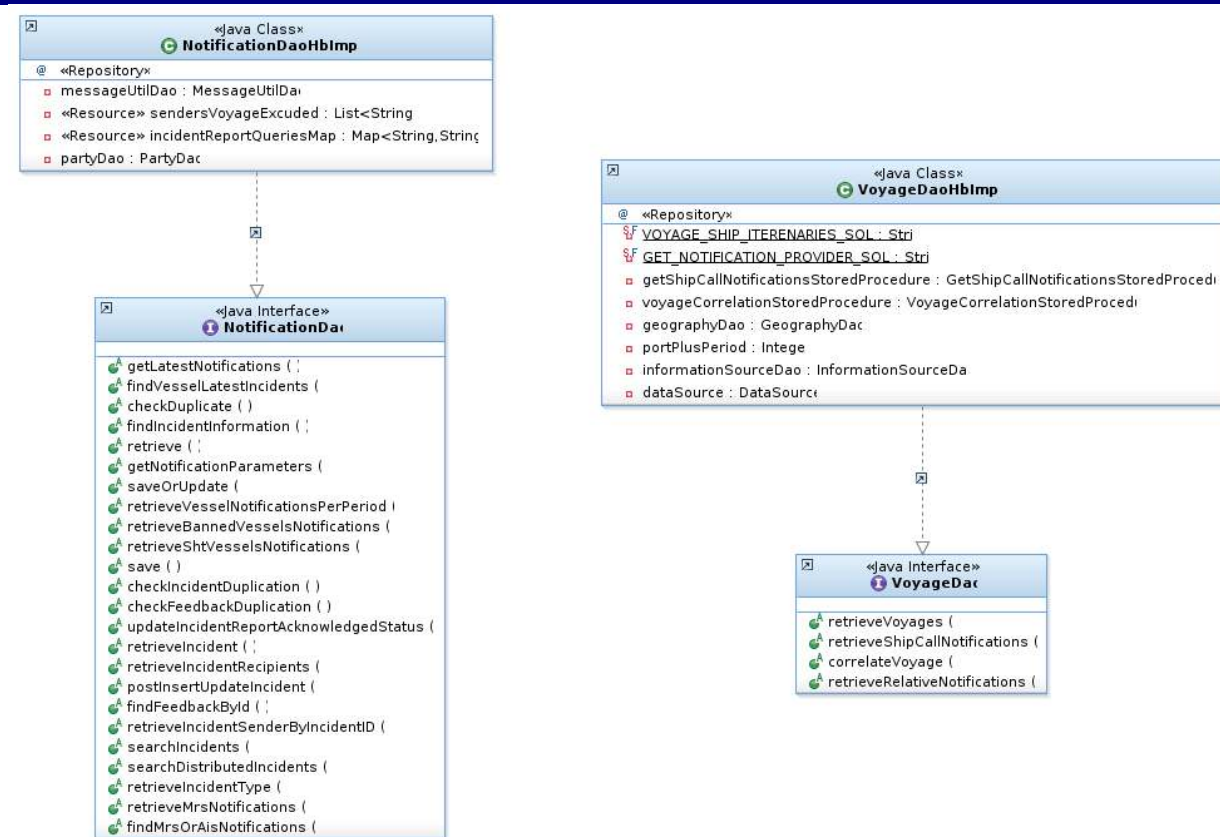
4.2.2.2.2 Package: message-log-dao

Class Diagram: message-log-dao



4.2.2.2.3 Package: notification-dao

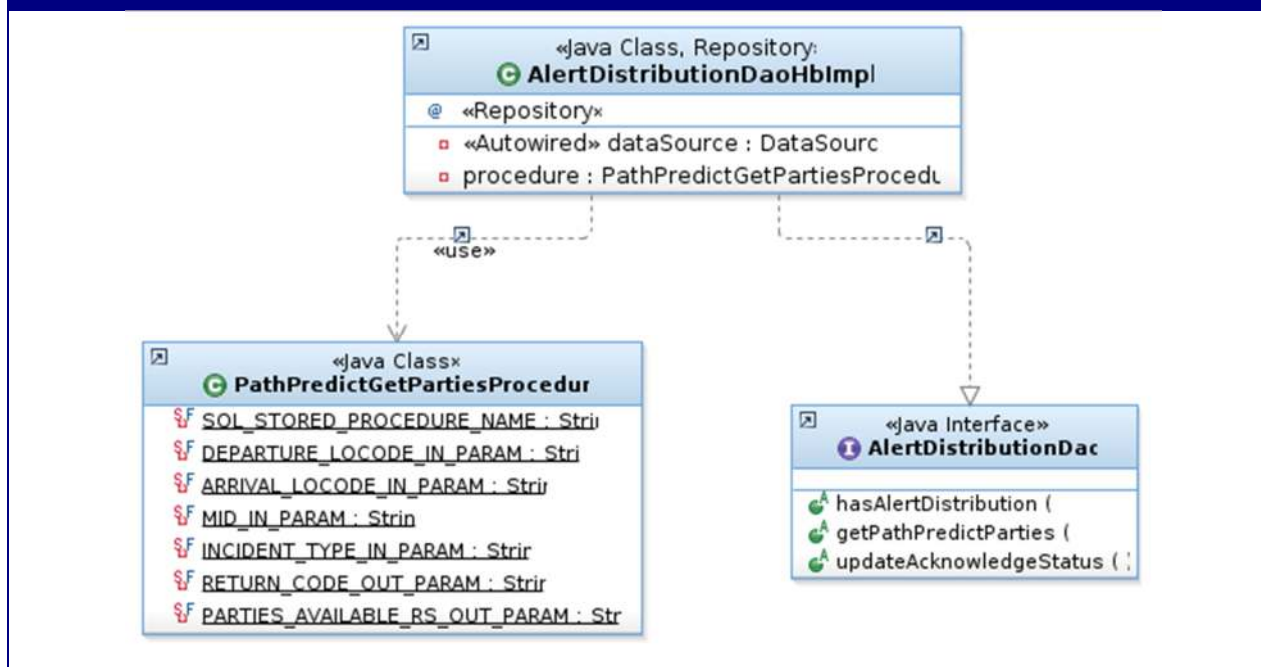
Class Diagram: notification-dao



Interface	NotificationDao A data access object for managing the database operations for notifications.
Class	NotificationHbImpl Hibernate based implementation of the NotificationDao.
Interface	VoyageDao A data access object for managing the database operations for voyages.
Class	VoyageHbImpl Hibernate based implementation of the VoyageDao.

4.2.2.2.4 Package: alert-distribution-dao

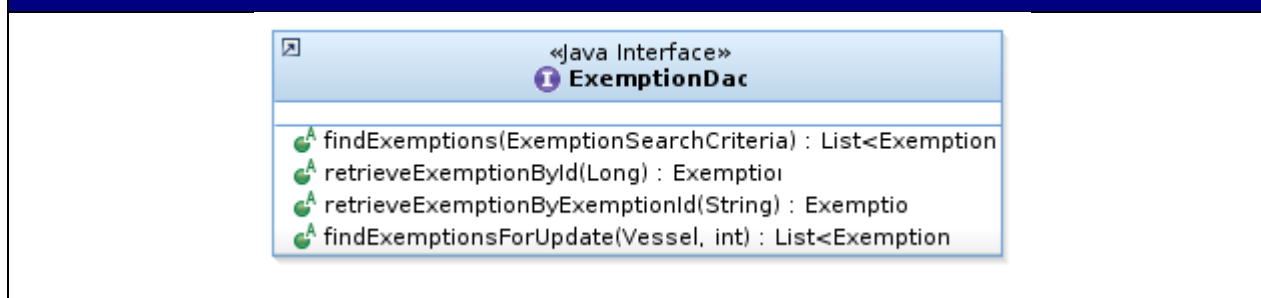
Class Diagram: Alert Distribution Dao package



Interface	AlertDistributionDao Database data access object for managing the database operations for IR notifications' recipients.
Class	AlertDistributionDaoHbImpl Hibernate implementation of AlertDistributionDao.
Class	PathPredictGetPartiesProcedure A class used to execute the RDBMS stored procedure that implements the 'Voyage Plan Prediction'.

4.2.2.2.5 Package: exemption-dao

Class Diagram : exemption-dao



Interface	ExemptionDao Interface describing the data access object of Exemptions.
------------------	---

4.2.2.2.6 Package: sat-dao

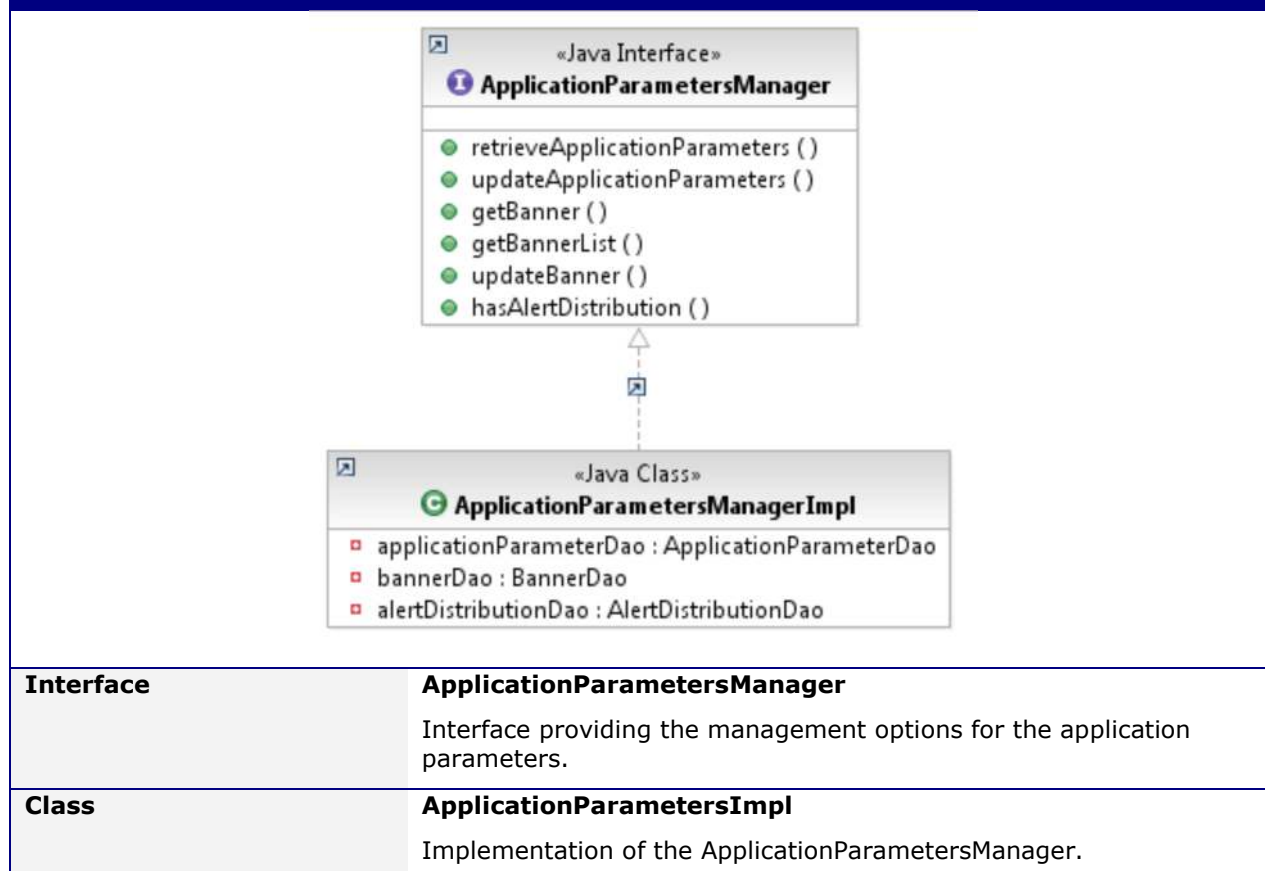
Class Diagram : sat-dao



4.2.2.3 Module: ssn-core

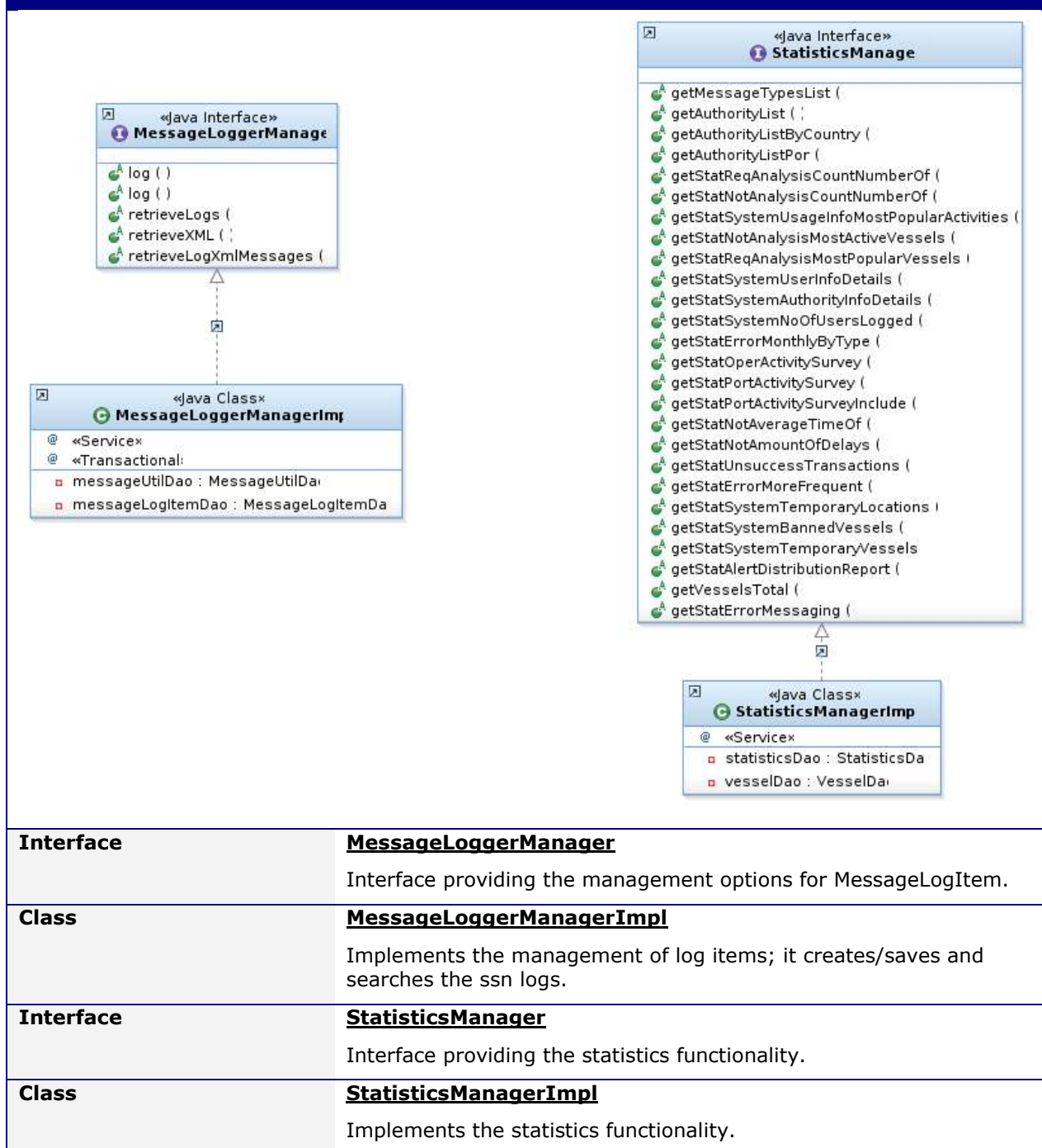
4.2.2.3.1 Package: common-manager

Class Diagram: common-manager



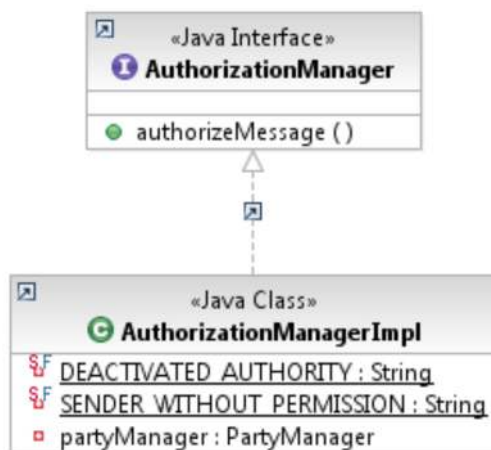
4.2.2.3.2 Package: message-log-manager

Class Diagram : message-log-manager



4.2.2.3.3 Package: message-manager

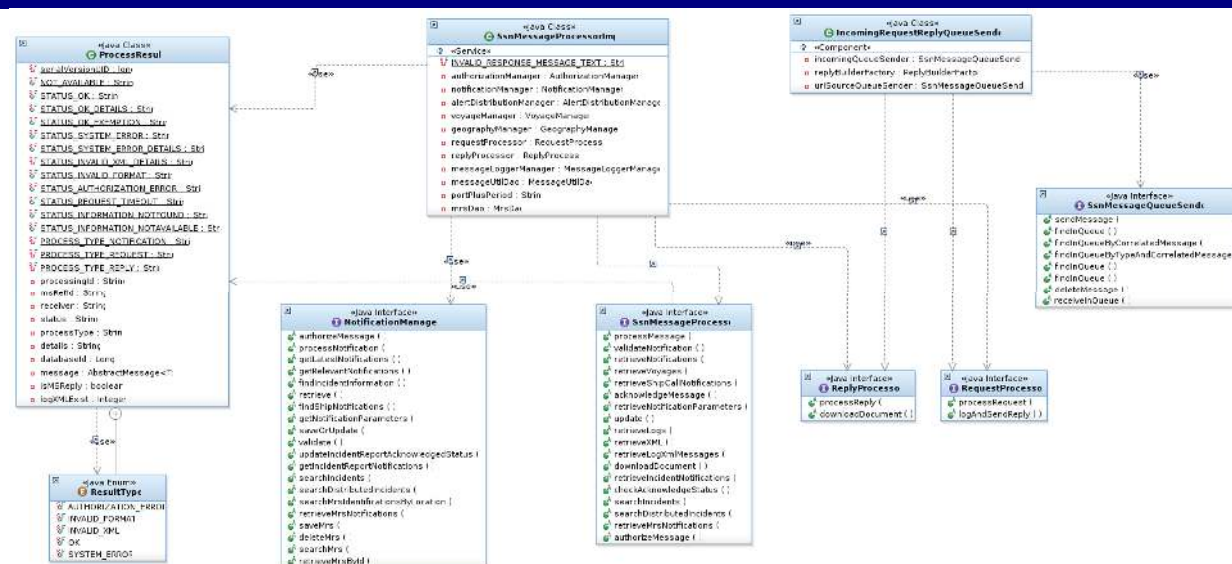
Class Diagram: message-manager



Interface	<u>AuthorizationManager</u> An interface provides the authorization of the incoming xml messages.
Class	<u>AuthorizationManagerImpl</u> Implements the authorization of the incoming xml messages via/using the Party manager.

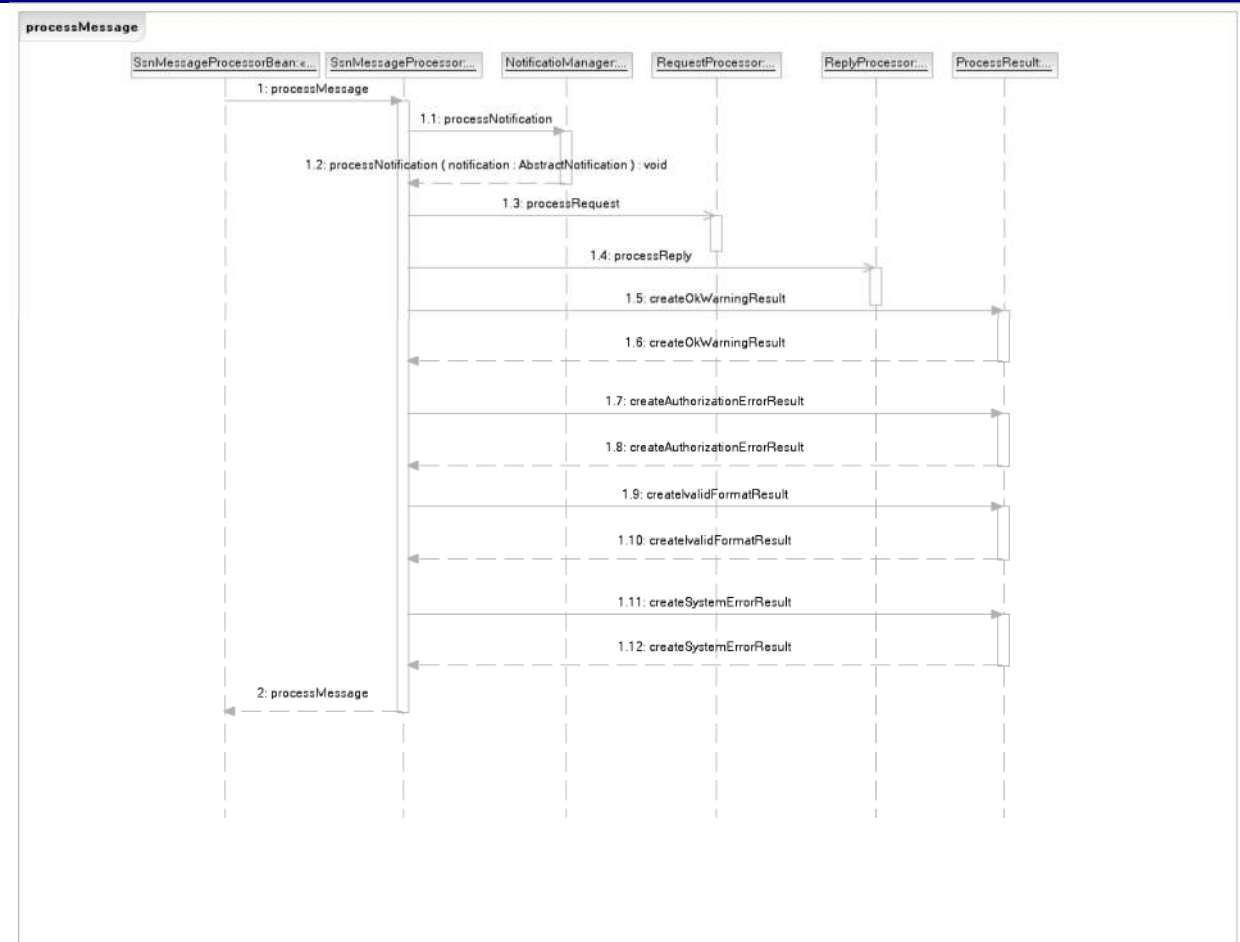
4.2.2.3.4 Package: message-processor

Class Diagram: message-processor



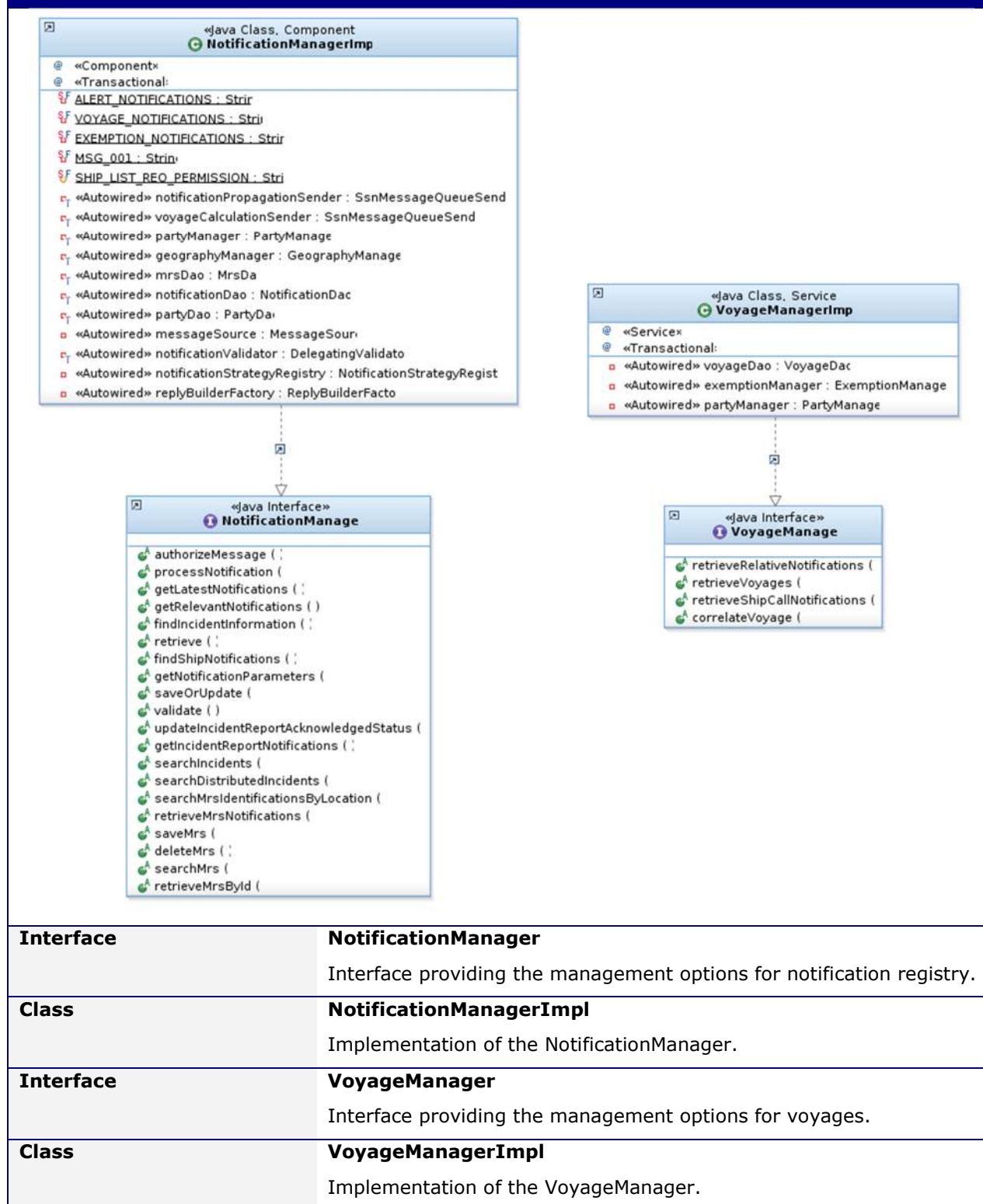
Sequence Diagram: Process Message- Notification & Request/Reply

Class Diagram: message-processor

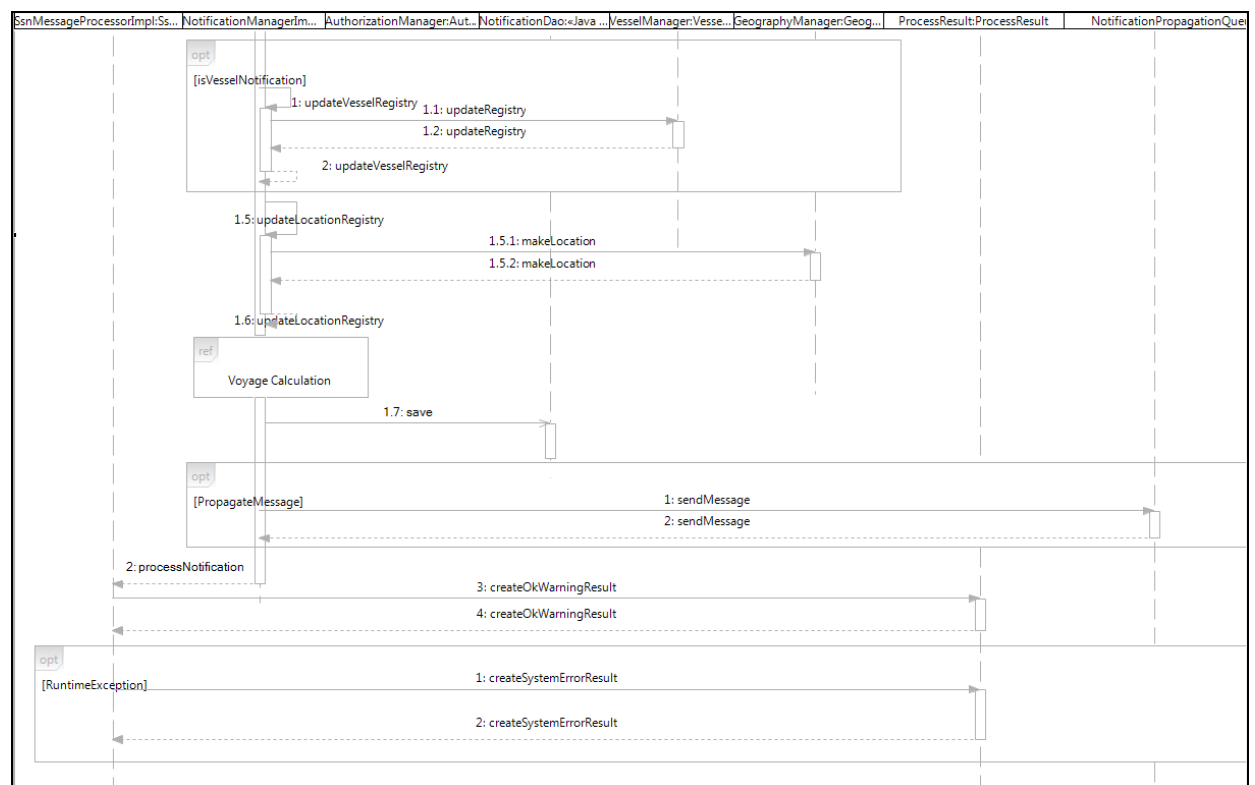
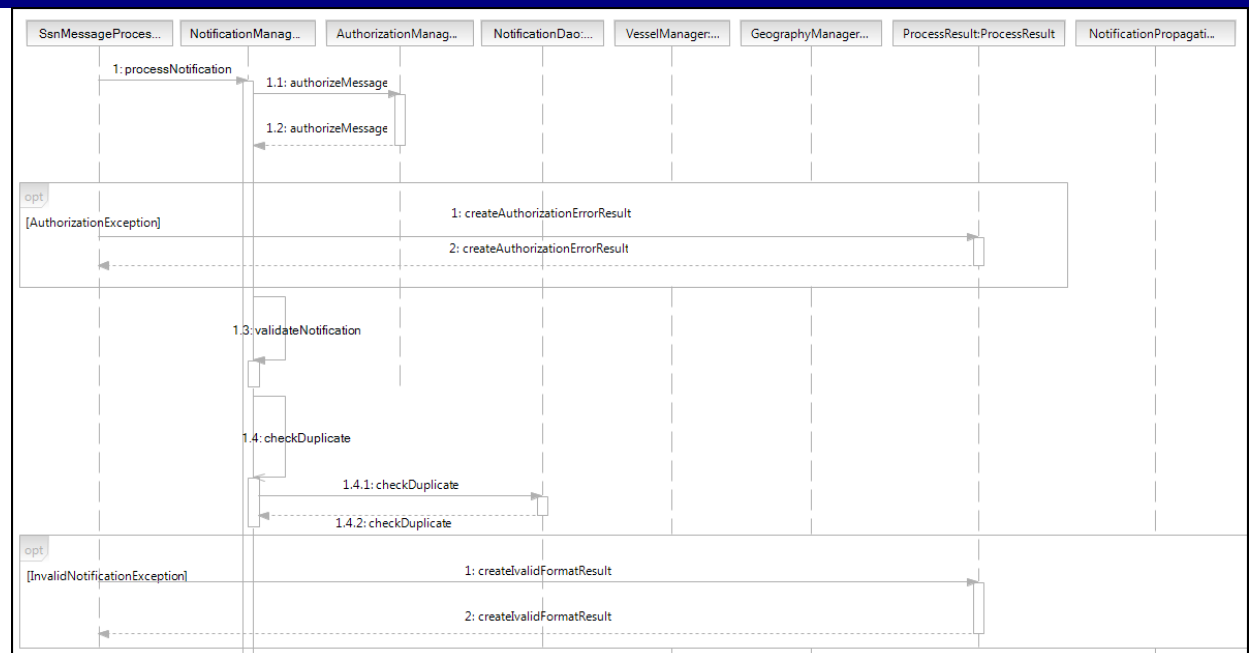


4.2.2.3.5 Package: notification-manager

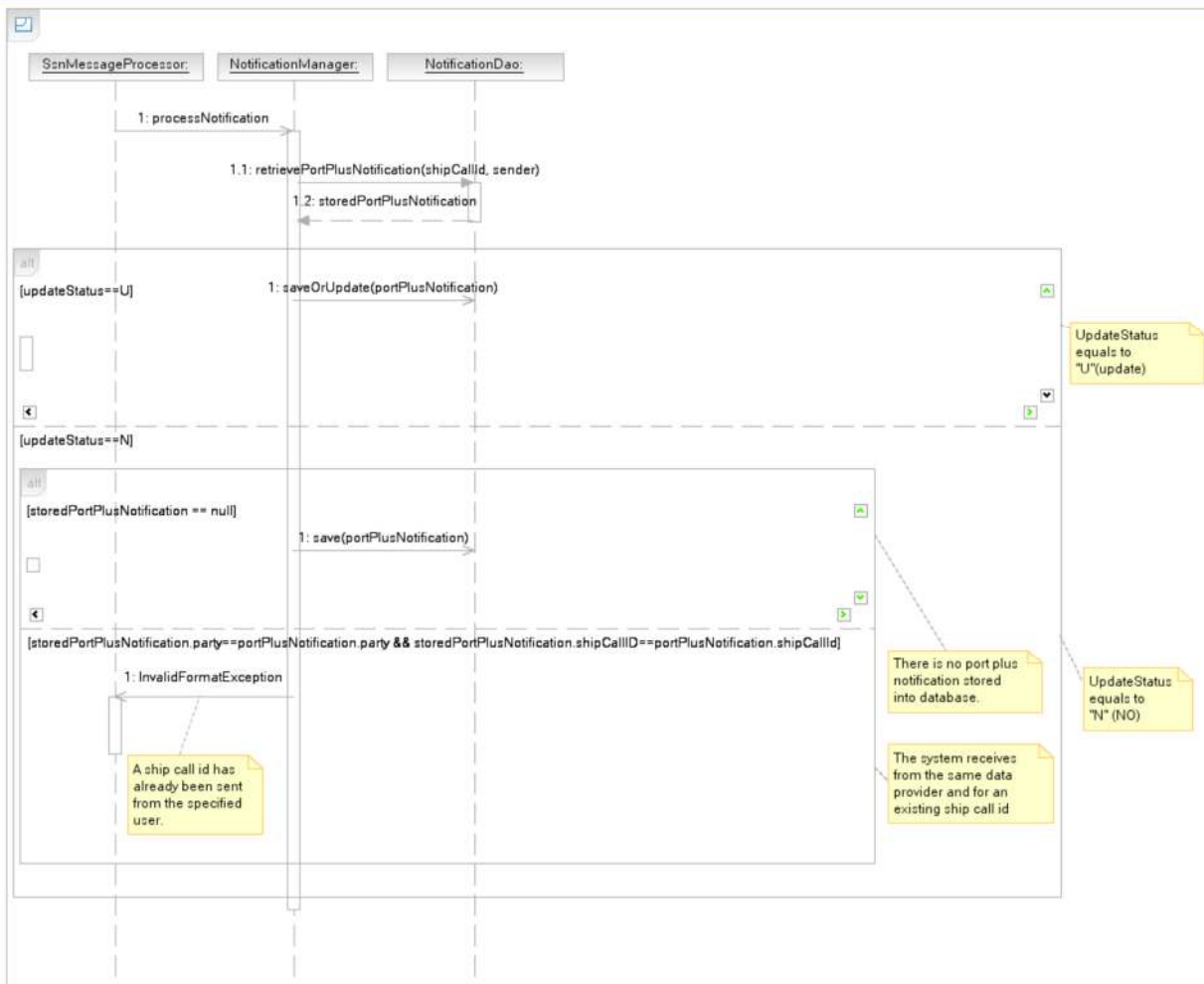
Class Diagram: notification-manager



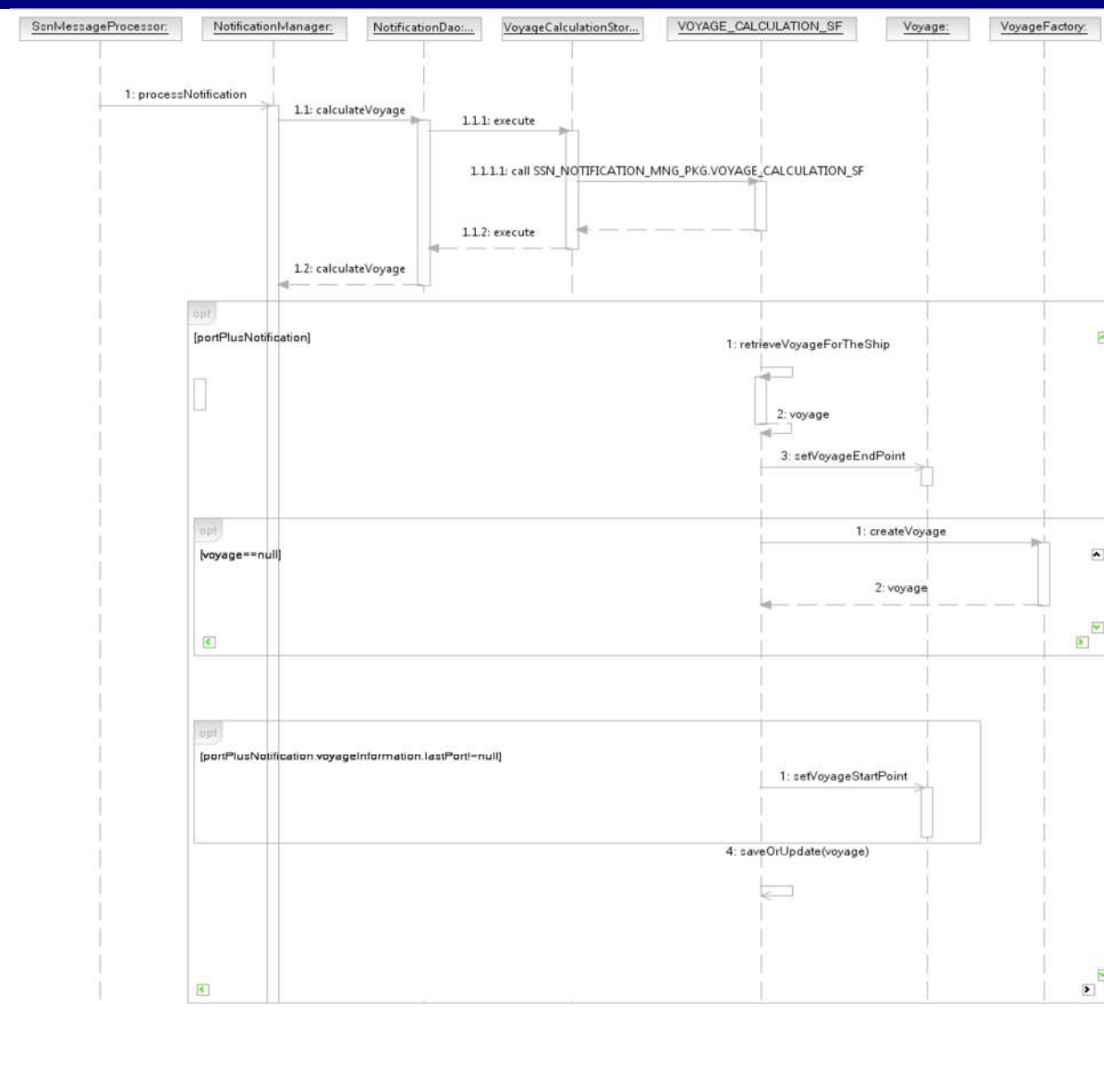
Sequence Diagram : Notification Processing - Valid Notification

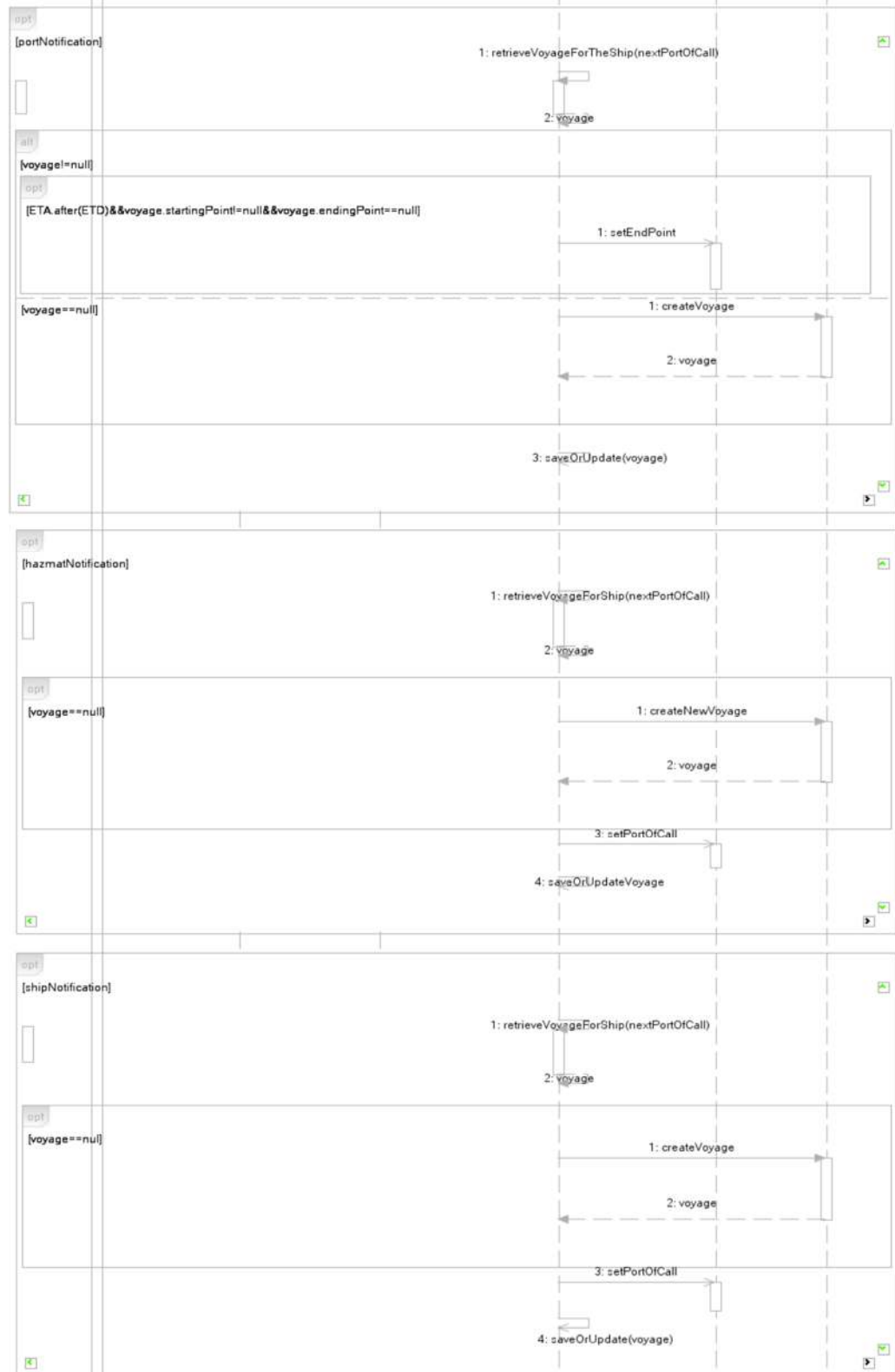


Sequence Diagram : Notification Processing – PortPlus Notification

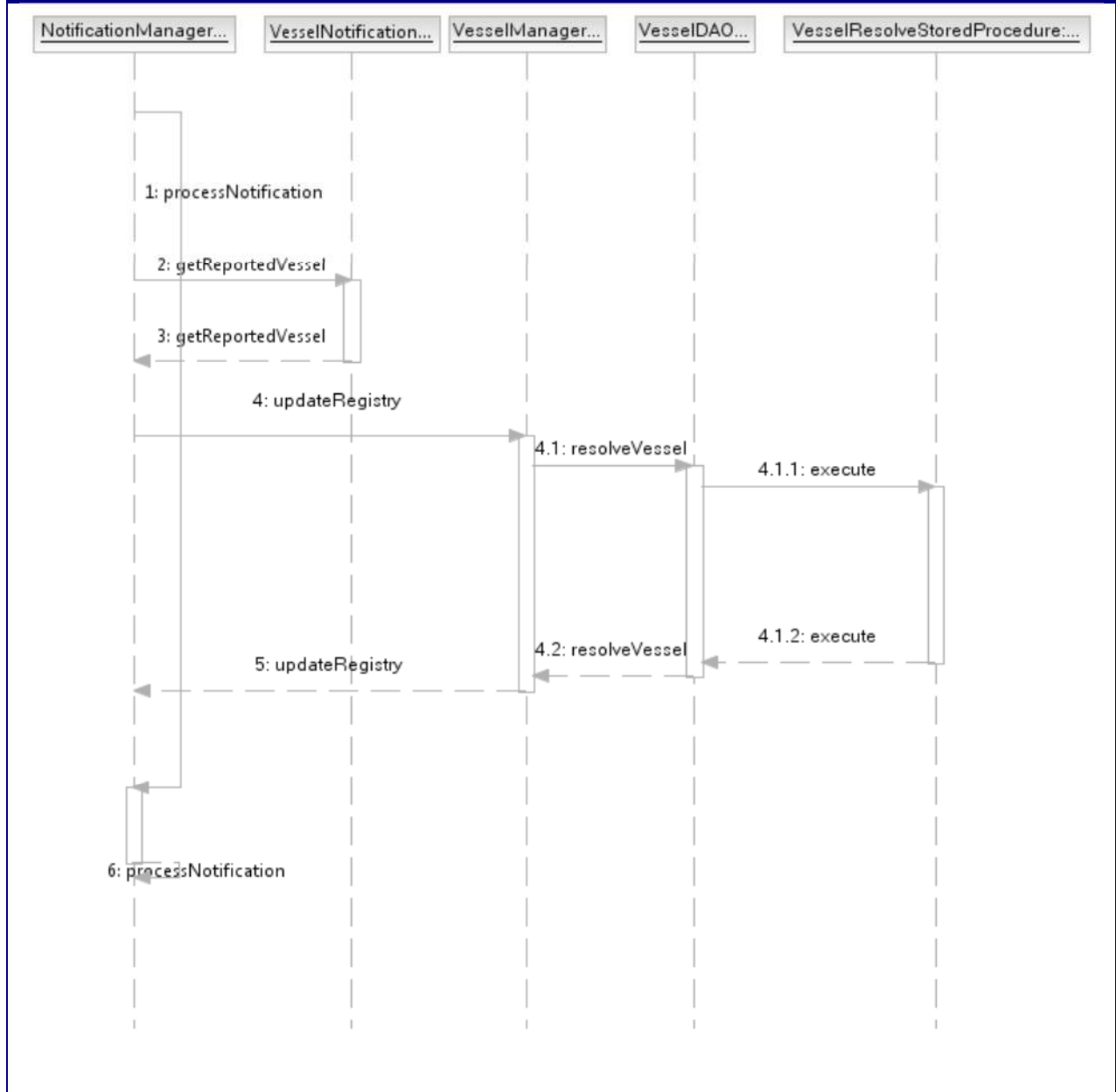


Sequence Diagram : Notification Processing – Voyage Calculation

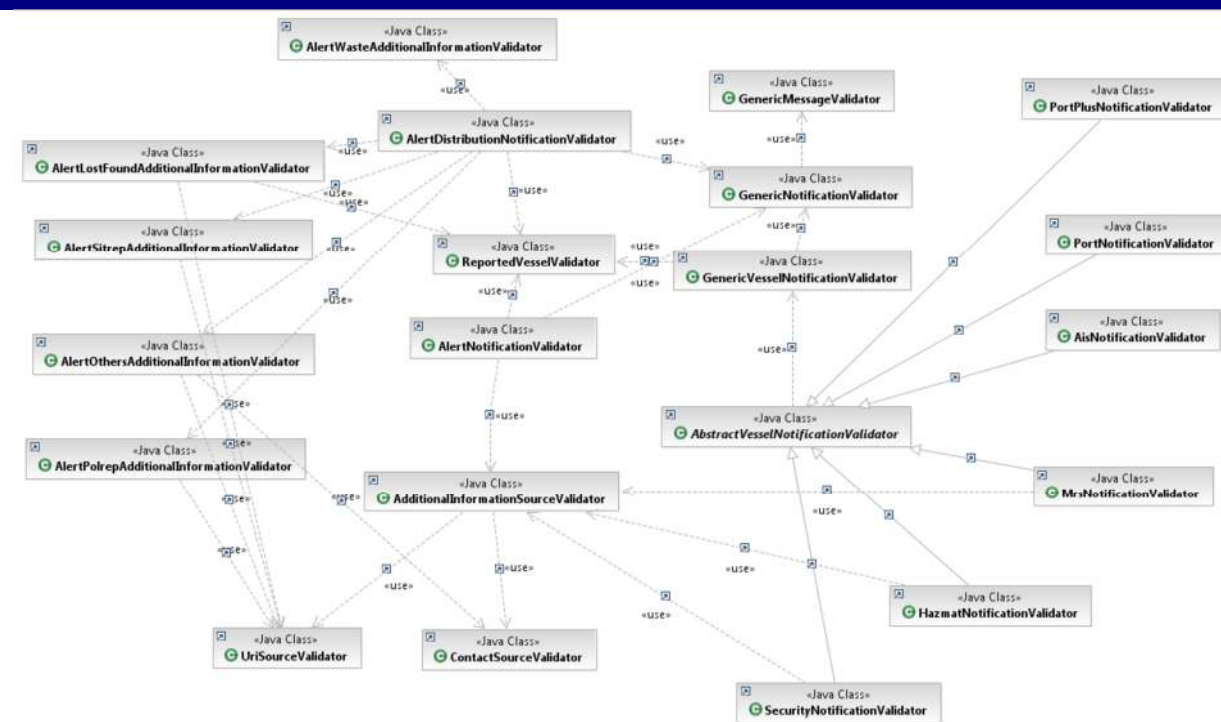




Sequence Diagram: Update Vessel Registry Using Reported vessel (IMO and MMSI Number, CallSign, Ship name)



Class Diagram: notification-validation



Package notification.validation

Class GenericMessageValidator

Validates the notification's header – MsRefId, sender, receiver, test case id.

Class GenericNotificationValidator

Validates the generic notification attributes.

Class GenericVesselNotificationValidator

Invokes the reported vessel validator.

Class ReportedVesselValidator

Validates the IMO number, MMSI number, the Call Sign, the Ship Name and Flag of the reported vessel. It actually delegates the validation of the vessel particulars to the corresponding vessel validators.

Class AbstractVesselNotificationValidator

This class is the common parent of all classes that implement discrete notification validators.

Class AisNotificationValidator

The business rules for the AIS (Ship) notification.

Class IncidentReportNotificationValidator

The business rules for the IncidentReportnotification for identified and non-identified vessels and the additional information source details. In case of notification for identified vessel, the ReportedVesselValidator is invoked.

Class Diagram: notification-validation	
Class	MrsNotificationValidator The business rules for the MRS (Ship) notification and the additional information source details
Class	PortPlusNotificationValidator The business rules for the PortPlus notification.
Package	common.validation
Class	AdditionalInformationSourceValidator The business rules for the notification details source (Contact source, URI).
Class	ContactSourceValidator Validates the contact's details such as location, name, email, phone/fax numbers.
Class	UriSourceValidator Validates the URI details

4.2.2.3.6 Voyage Plan Prediction

The process described in the following sequence diagram concerns the prediction of the voyage Plan based on the departing and arriving location code given by the Web user. As specified in the voyage plan prediction algorithm, later in this subsection, three alternative cases exist: cases 1, 2 and 3. Each of the alternative cases is presented in a separate sequence diagram.

A PL/SQL stored procedure implements this algorithm for performance purposes – it avoids JDBC database roundtrips. As an input the package takes the Starting and Ending points (location codes) that the user specifies in the Incident ReportsDistribution Web application. As an output the package provides the list of proposed recipients based on the location codes to be produced by the algorithm and the NCA authorities that reside on a location code selected.

The basic description is presented in the Table 4-5. The prediction of the plan is based primarily on the definition of the Rout-Points. The convention used for defining a Rout-Point is **R (i, j, k, l)**, where:

- **R**⇒ Stand for Rout-Point
- **i**⇒ Is the Level of the branch. Values: 1 is the basic branch; 2 is the first branch level out of branch 1; level 3 is a branch to branch 2 and so on.
- **j**⇒ Is the branch number. For level 1, the branch number “b” is omitted since only one branch is considered at this level.
- **k**⇒ Is the Rout-Point number for that specific branch.
- **l**⇒ Is an activity parameter. Values: 1 = active (send alert notification); 0 = inactive (do not send alert notification details).

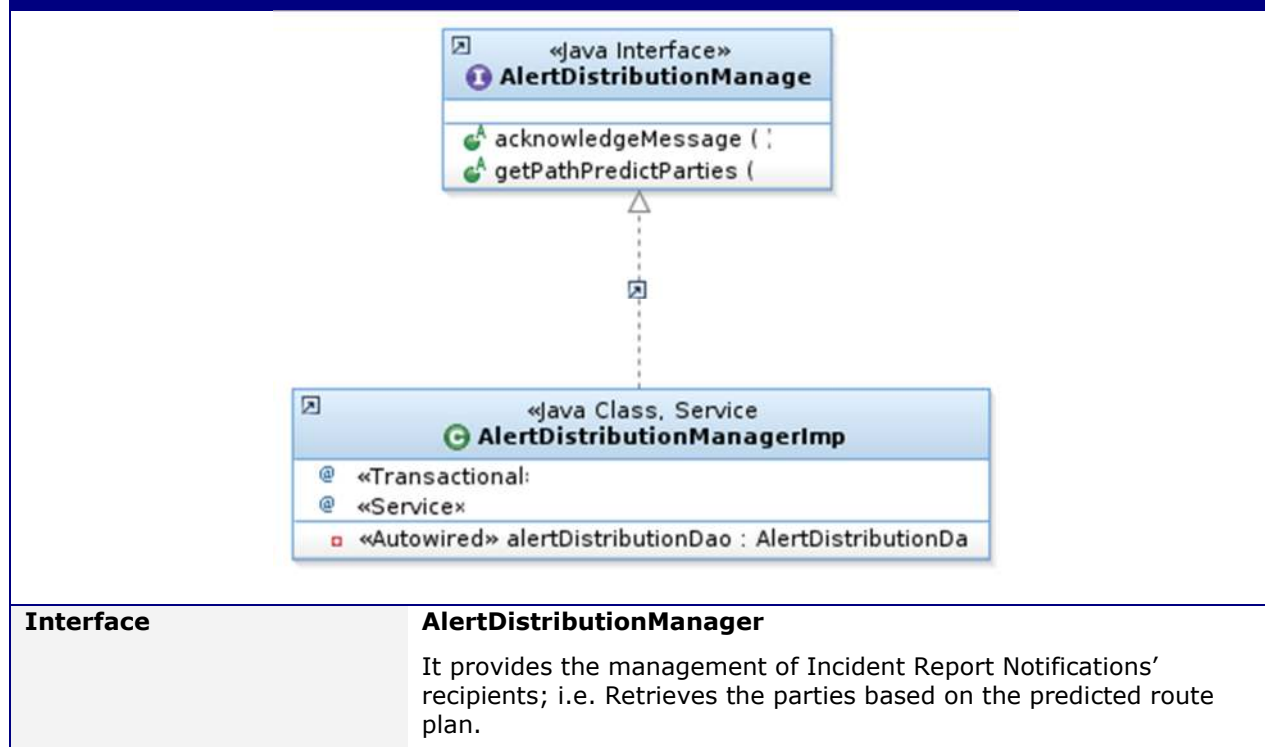
Functional description				
Input →				VoyagePlan (Departing Set; Arriving Set)
1.1				R (i ₁ ; j ₁ ; k ₁ ; l ₁) = Departing set R (i ₂ ; j ₂ ; k ₂ ; l ₂) = Arriving set

	2.1			If $i_1 = i_2$ and $j_1 = j_2$ and $k_1 = k_2$ then Case 1
	2.2			# The departing and arriving is in the same branch and set
		3.1		Case 1
		3.2		Message: The departure and the arriving are in the same area. No Alert distribution is needed
	2.3			If $i_1 = i_2$ and $j_1 = j_2$ and $k_1 \neq k_2$ then Case 2
	2.4			# The departing and arriving is in the same branch
		3.3		Case 2
		3.4		$n = 0$
		3.5		$a = 0$
		3.6		$m_1 = k_2 - k_1$
		3.7		$m_2 = m_1 /m_1$
		3.8		$n_1 = 0 * m_2$ (the increment)
		3.9		# Create the Array $A_1(i_1+n; j_1+n; k_1+ n_1; l_1)$
		3.10		$A_a(i_1+n; j_1+n; k_1+ n_1; l_1) = R(i_1+n; j_1+n; k_1+ n_1; l_1)$
		3.11		$n_1 = n_1 + m_2$
		3.12		$a = a + 1$
		3.13		If $ n_1 > m_1 $ then exit else go to 3.10
			4.1	Create a Union of the sets in A_a with single occurrence of countries, and the NCA's shall be identified
		3.14		Exit Case 2
	2.5			# The departing port and the arriving port are in different branches, level 1 or 2
	2.6			If $i_1 = i_2 = 2$ and $j_1 \neq j_2$ then Case 3
		3.8		Case 3
		3.9		# For simplified reasons we calculate this occurrence as above. Each branch is connected together, so we can use the method above; just anticipate that we have 2 or three separate branches. From start Set to Set before intersection ($k=1$), Then find intersection with the main branch, Set $R(1.1.k_1.l)$. Then find the arriving Set in branch 1, Set $R(1.1.k_2.l)$. For this case use method under 2.4 for each part. The total Array will be: $A_T = A_1 \cap A_2$. Recognise that an Array consists of one or more Sets, $A = R(i_1; j_1; k_1; l_1) \cap R(i_2; j_2; k_2; l_2) \cap \dots \cap R(i_n; j_n; k_n; l_n)$. If we have the arriving port in another level two we will then have 3 branches, then we will get the following result: $A_T = A_1 \cap A_2 \cap A_3$. If we cal a Set R, then the Array will be: $A = R_1 \cap R_2 \cap \dots \cap R_n$

Table 4-5 Function Description

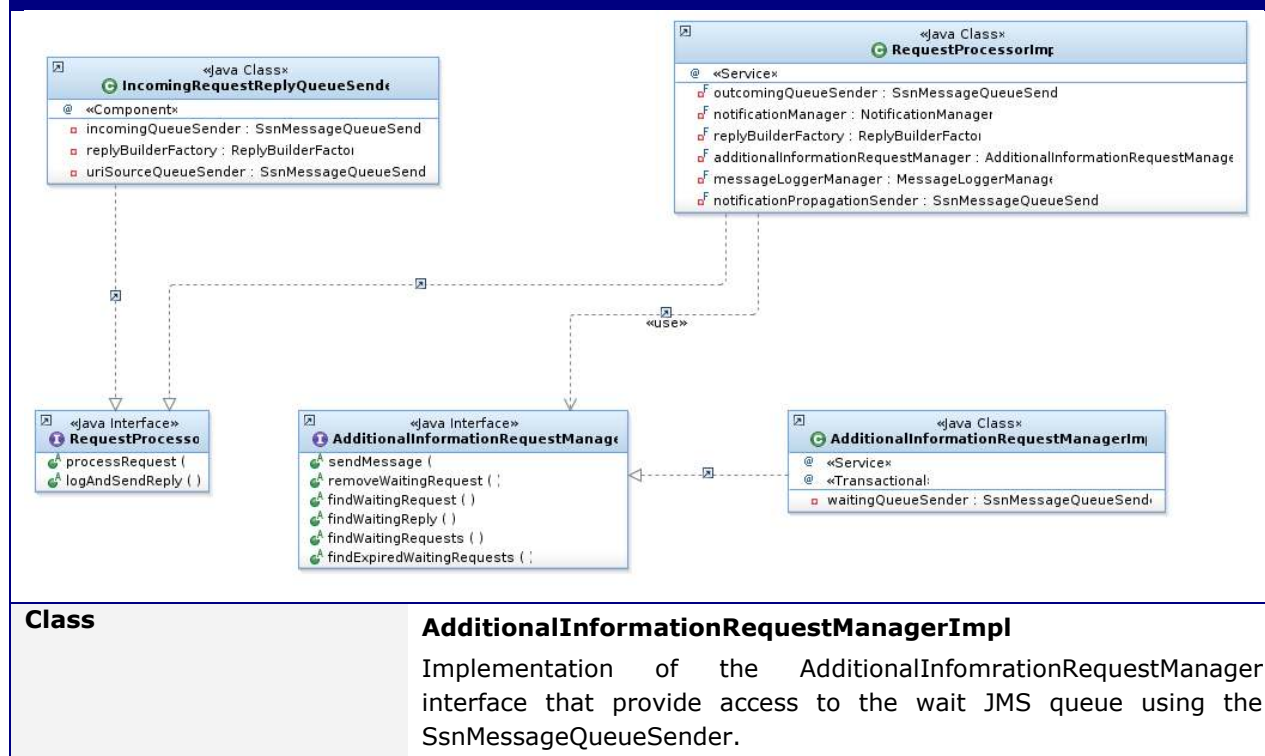
4.2.2.3.7 Package: alert-distribution-manager

Class Diagram: Alert Distribution manager package



4.2.2.3.8 Package: request-processor

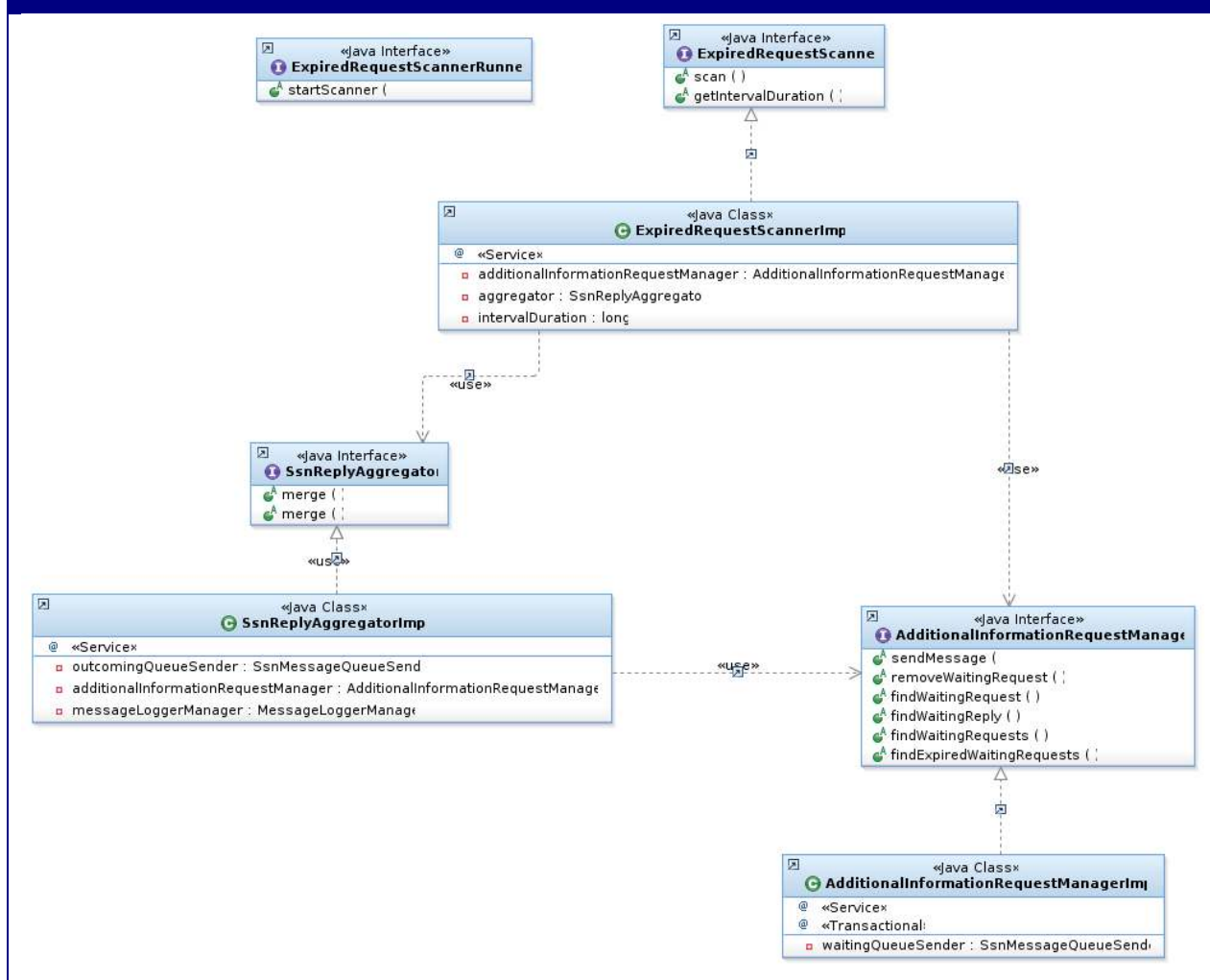
Class Diagram: request-processor



Class Diagram: request-processor	
Class	RequestProcessorImpl Implementation of the RequestProcessor interface that process an information request.
Interface	RequestProcessor An interface implemented by classes that are capable of processing an information request. There are two implementations of the interface: IncomingRequestReplyQueueSender: which is used to send a request to the incoming queue. RequestProcessorImpl: which is used to actually process a request from the incoming queue.
Class	AdditionalInformationRequestManager Implementation of the AdditionalInformationRequestProcessor interface that process an information request.
Interface	IncomingRequestReplyQueueSenderProcessor Implementation of the RequestProcessor interface that process an information request.

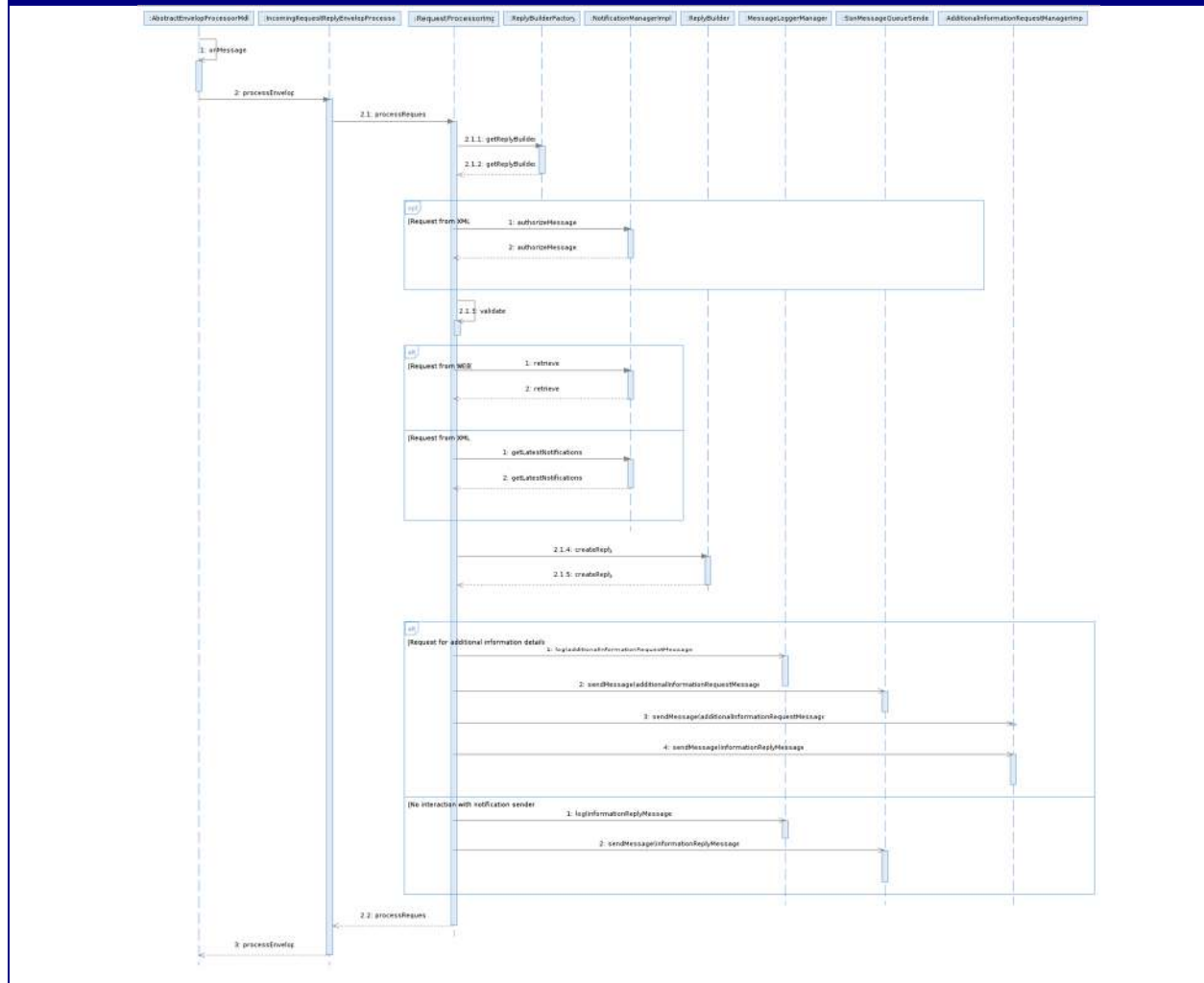
4.2.2.3.9 Package: expired-request-processor

Class Diagram: expired-request-processor



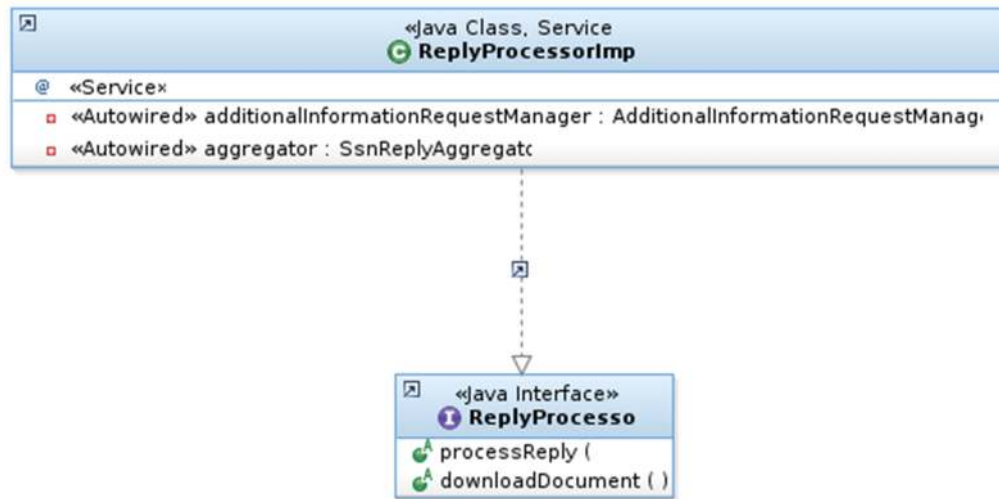
Class	AdditionalInformationRequestManagerImpl Implementation of the AdditionalInfomrationRequestManager interface that provide access to the wait JMS queue using the SsnMessageQueueSender.
Class	SsnReplyAggregatorImpl Implementation of the SsnReplyAggregator interface that process an expired request due to time out.
Class	ExpiredRequestScannerImpl Implementation of the ExpiredRequestScanner interface that scans the wait queue for expired requests.
Interface	ExpiredRequestScannerRunner An interface implemented by classes to trigger the scanning of wait queue for expired requests.

Sequence Diagram: Process Request Reply from EIS



4.2.2.3.10 Package: reply-processor

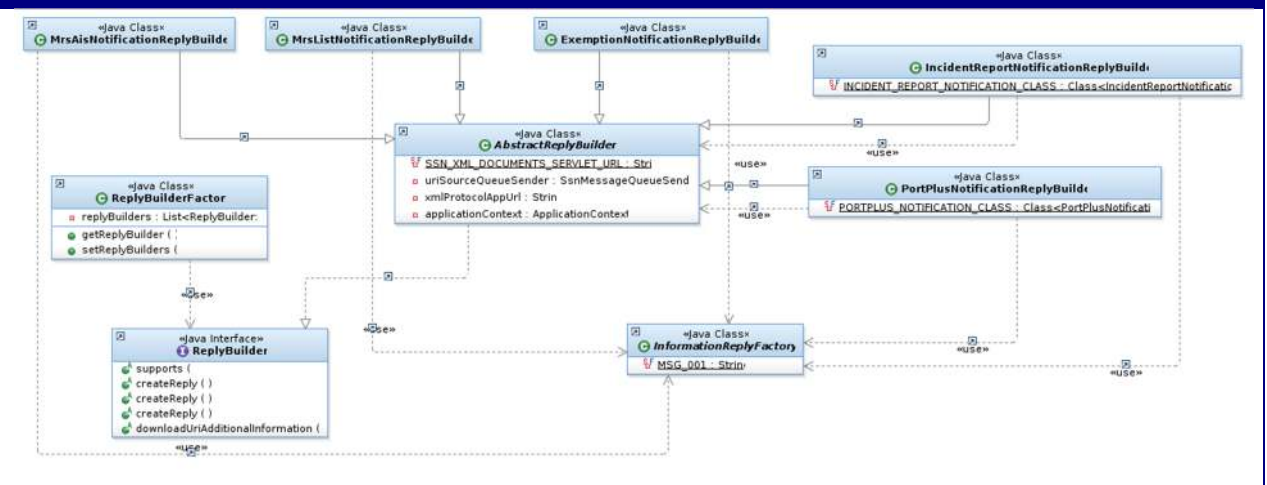
Class Diagram: reply-processor



Class	ReplyProcessorImpl Implementation of the ReplyProcessor interface that process the reply from the data provider to the requester for additional details based on a notification previously sent.
Interface	ReplyProcessor An interface implemented by classes that are capable of processing an information reply received from the data provider. The reply corresponds to a request (waiting in the queue) for additional details received from the data requester. Equivalent to RequestProcessor, there are two implementations of the interface: <ol style="list-style-type: none"> 1. IncomingRequestReplyQueueSender: which is used to send a request to the incoming queue. 2. ReplyProcessorImpl: which is used to actually process a reply from the incoming queue.

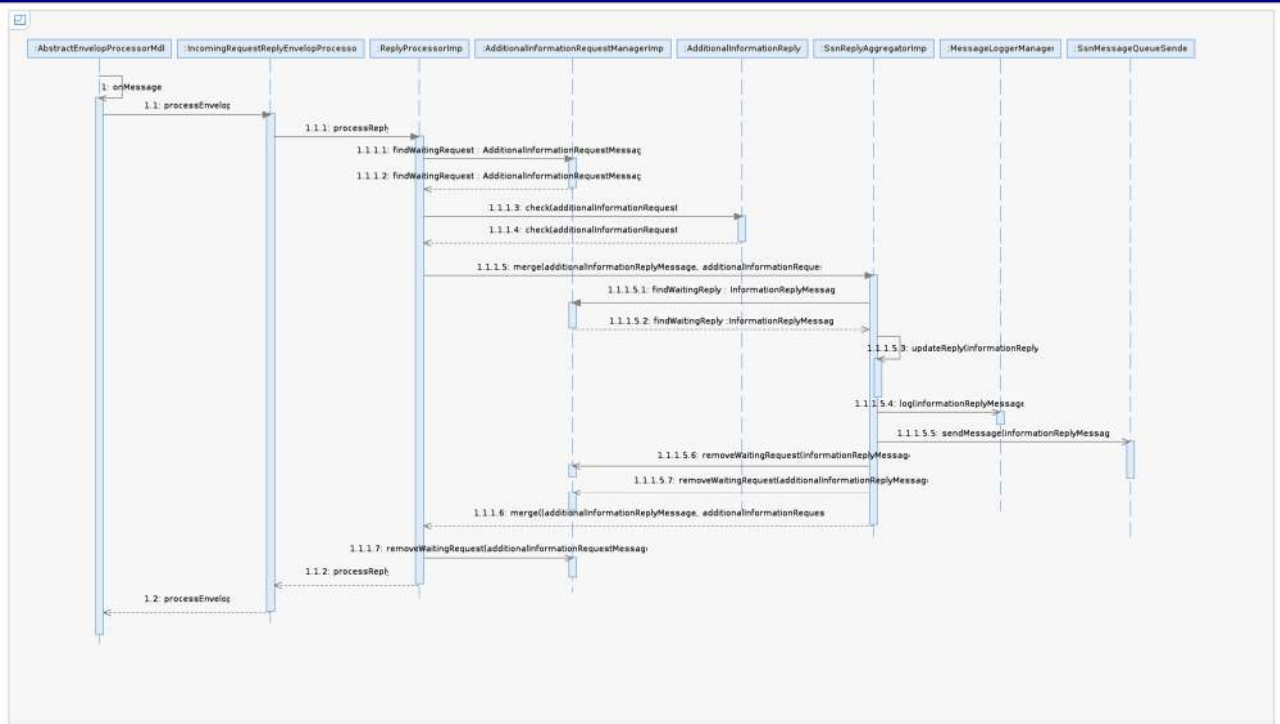
4.2.2.3.11 Package: reply-builder

Class Diagram: reply-builder

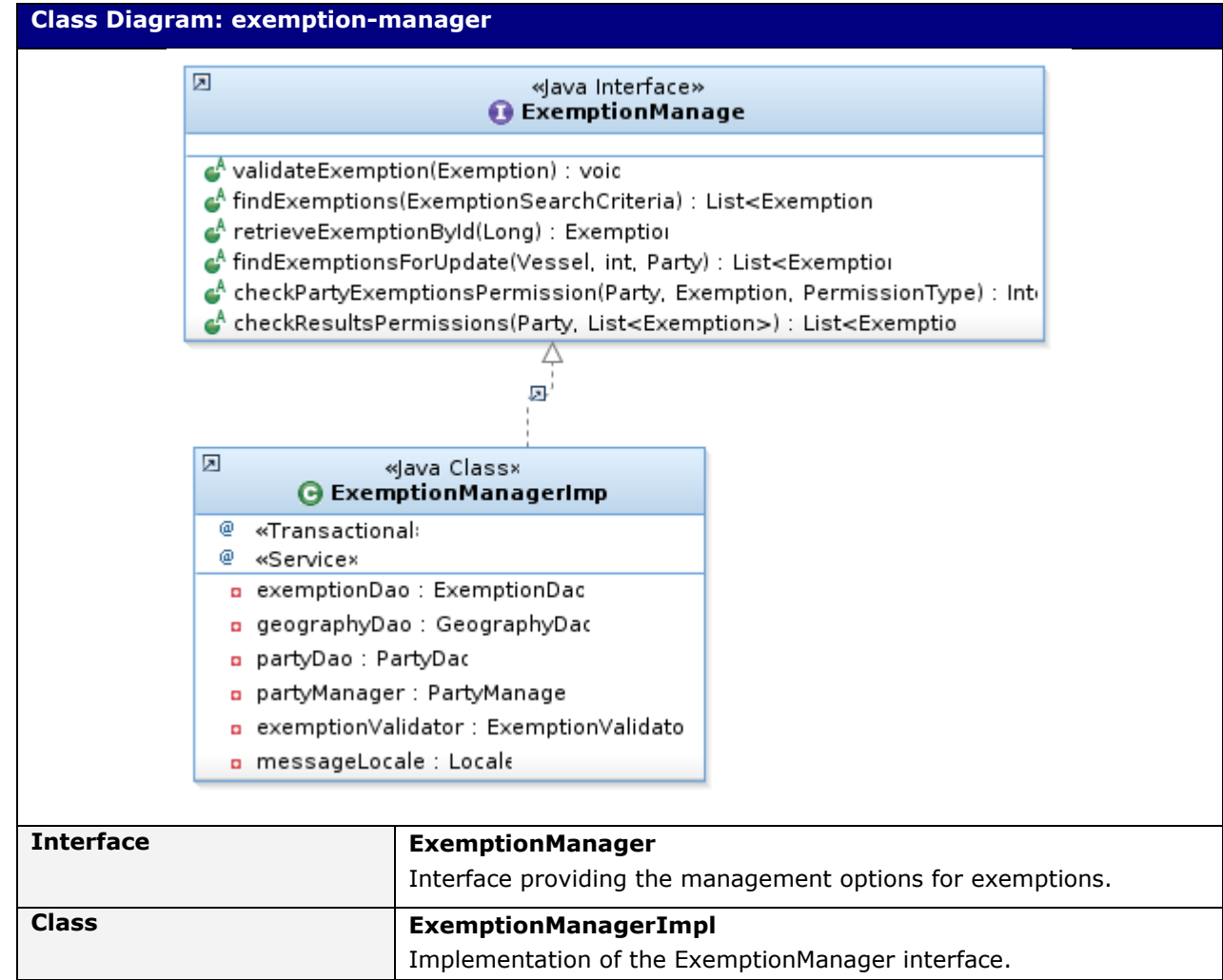


Class	AbstractReplyBuilder An abstract implementation of the ReplyBuilder interface that implements the "createReply" method.
Class	ReplyBuilderFactory A factory class used to create the ReplyBuilder.
Class	InformationReplyFactory An abstract factory class used to create the reply upon a request; i.e. a normal and/or an exception (e.g. InformationNotFoundReply) reply.
Interface	ReplyBuilder Indicates whether the current builder can handle the given request. There are two types of replies: the one build based on the reply sent by the data provider and the one build by SSN based on its own resources.
Class	IncidentReportNotificationReplyBuilder An implementation of the ReplyBuilder interface for the Incident Reportnotification requests. The reply is build based on the response received by the data provider.
Class	MrsAisNotificationReplyBuilder An implementation of the ReplyBuilder interface for the Ship (MRS/AIS) notification requests. The reply is build based on the response received by the data provider.
Class	MrsListNotificationReplyBuilder An implementation of the ReplyBuilder interface for the Ship List (MRS) notification requests.
Class	PortPlusNotificationReplyBuilder An implementation of the ReplyBuilder interface for the PortPlus notification requests. The reply is build based on the response received by the data provider.
Class	ExemptionNotificationReplyBuilder An implementation of the ReplyBuilder interface for the Exemption notification requests.

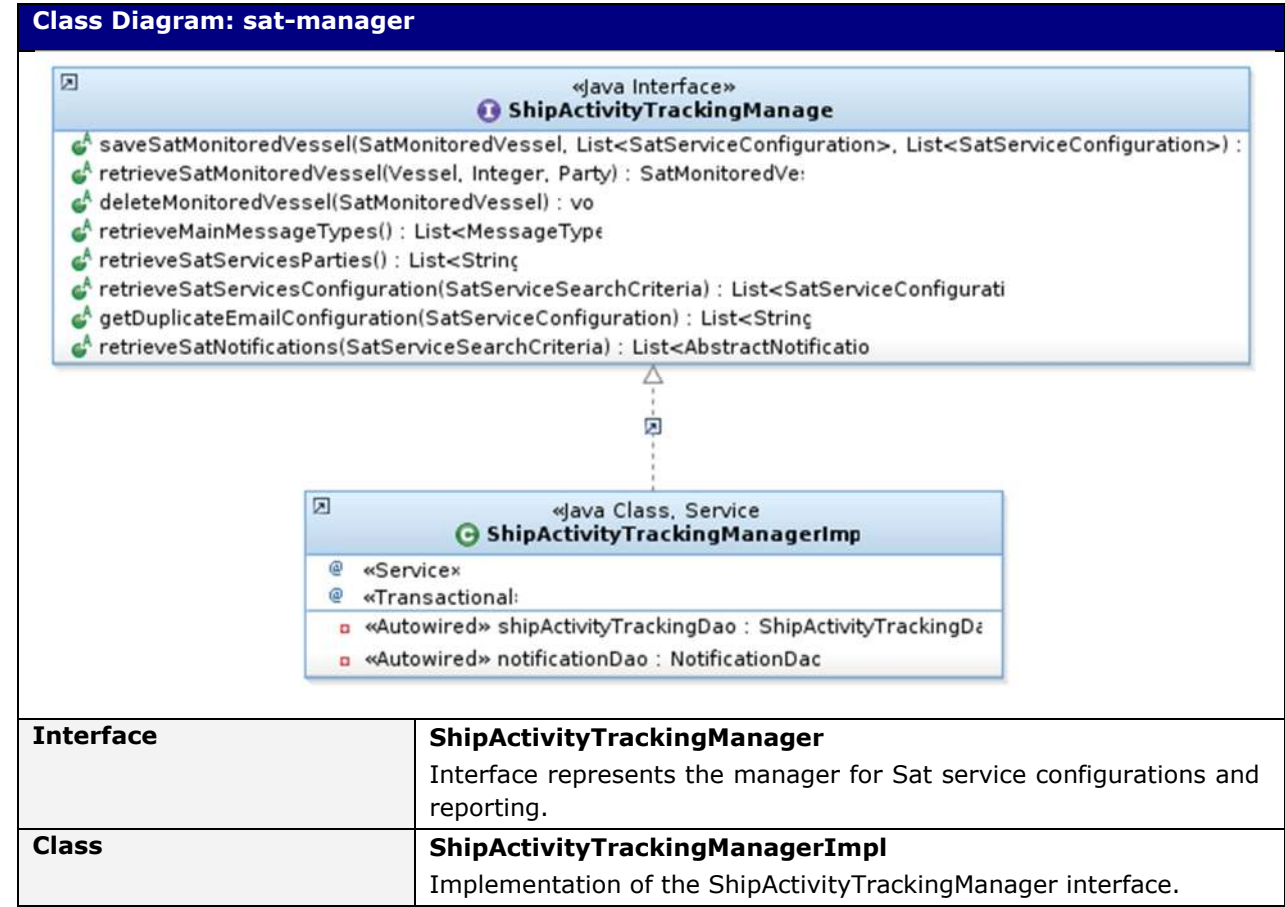
Sequence Diagram: Reply Processor



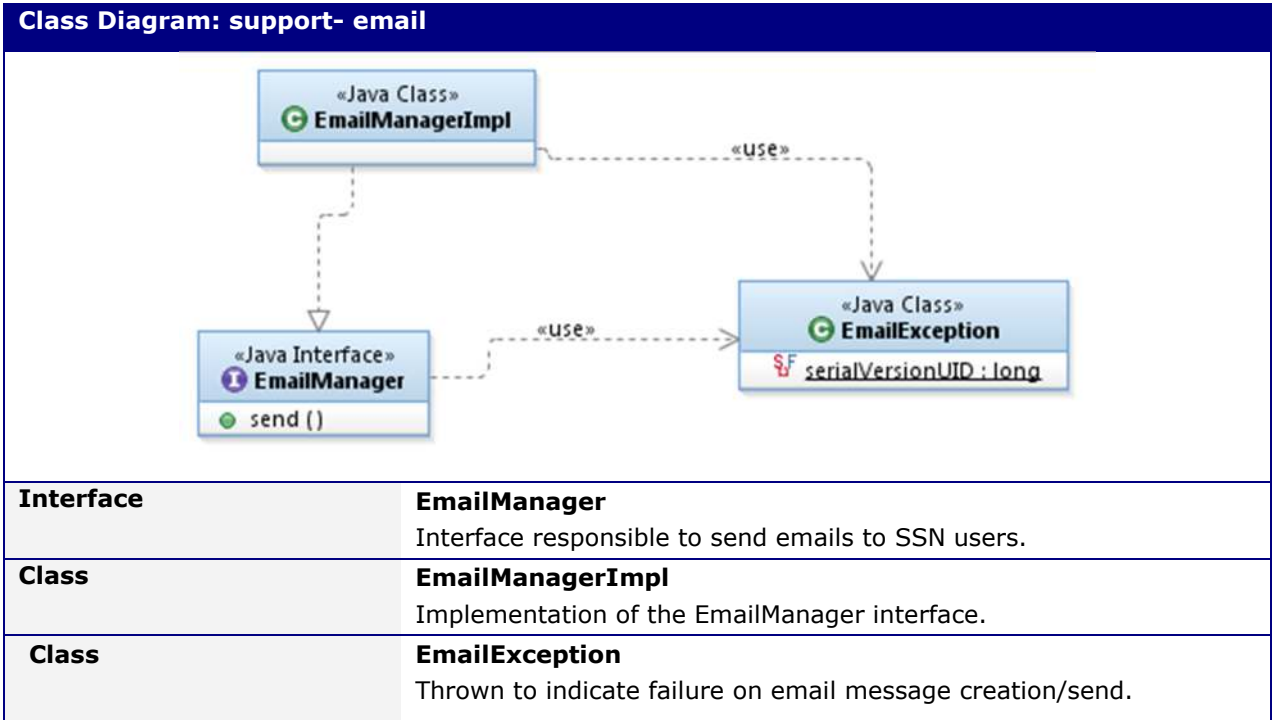
4.2.2.3.12 Package: exemption-manager

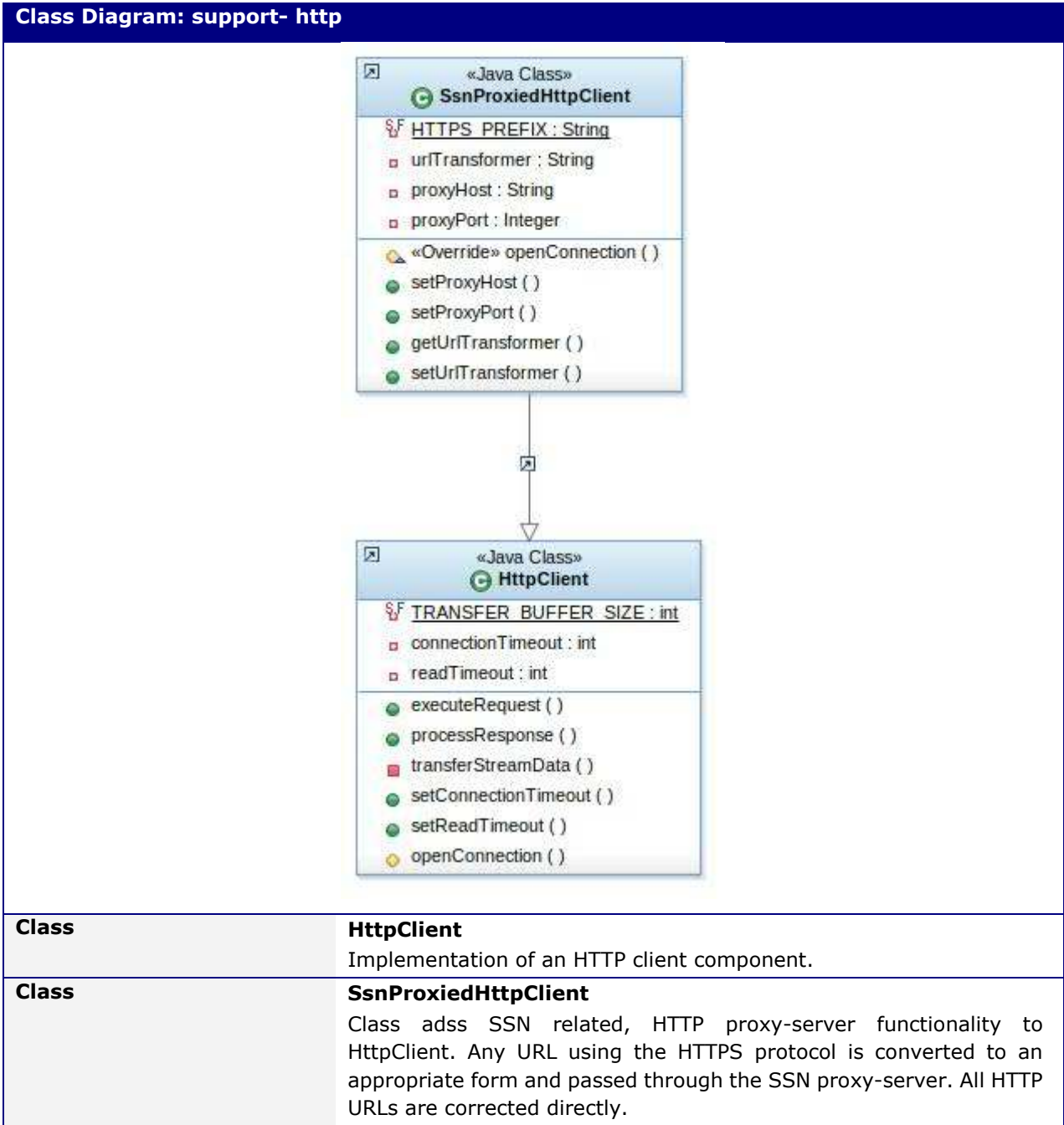


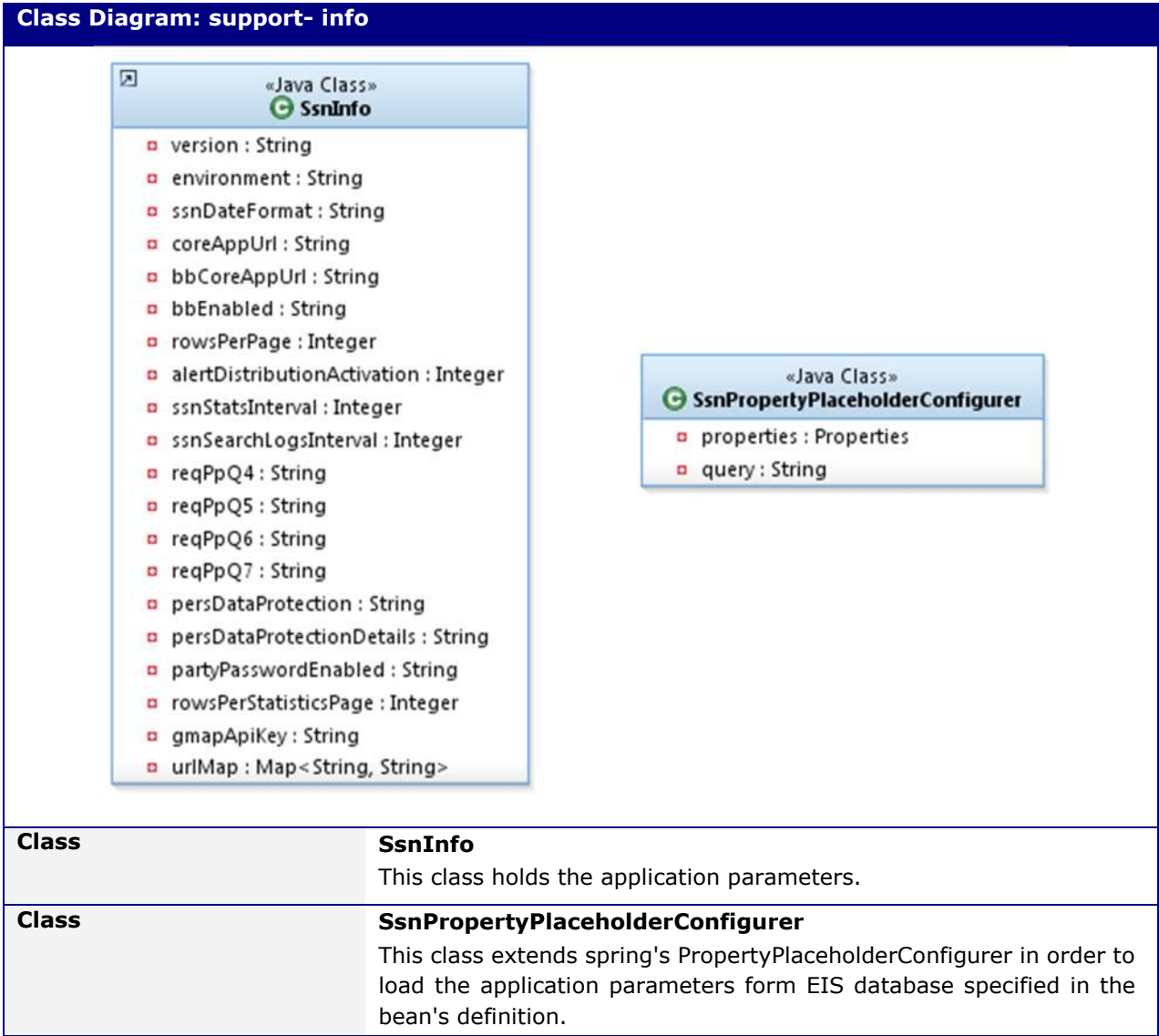
4.2.2.3.13 Package: sat-manager



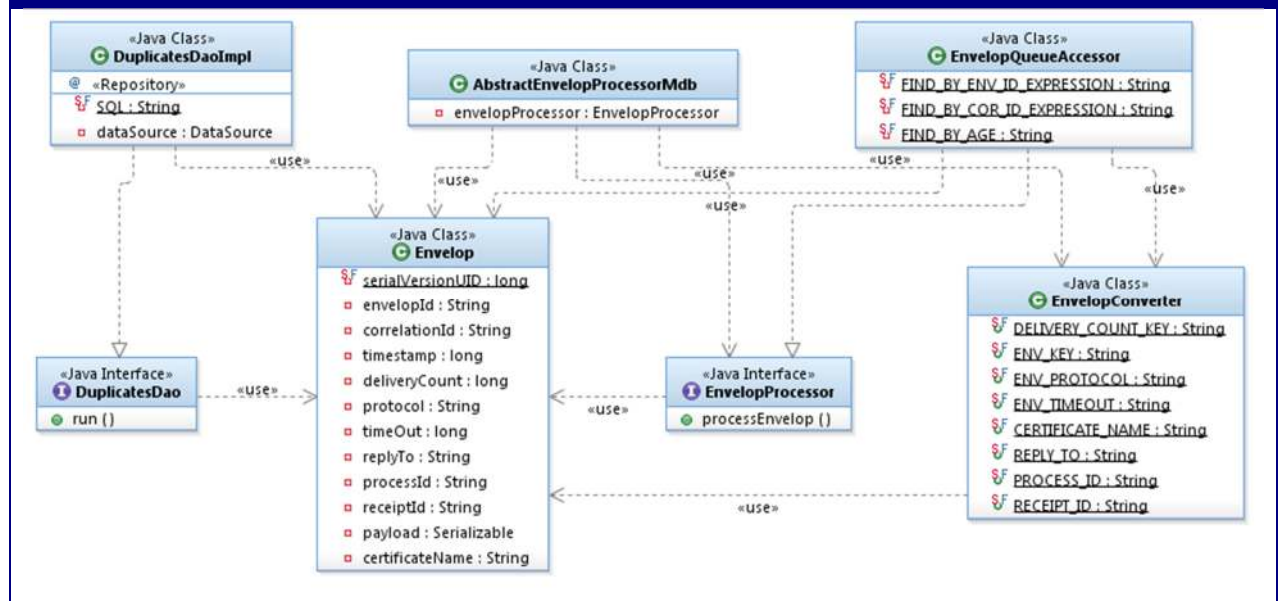
4.2.2.4 Module: ssn-support





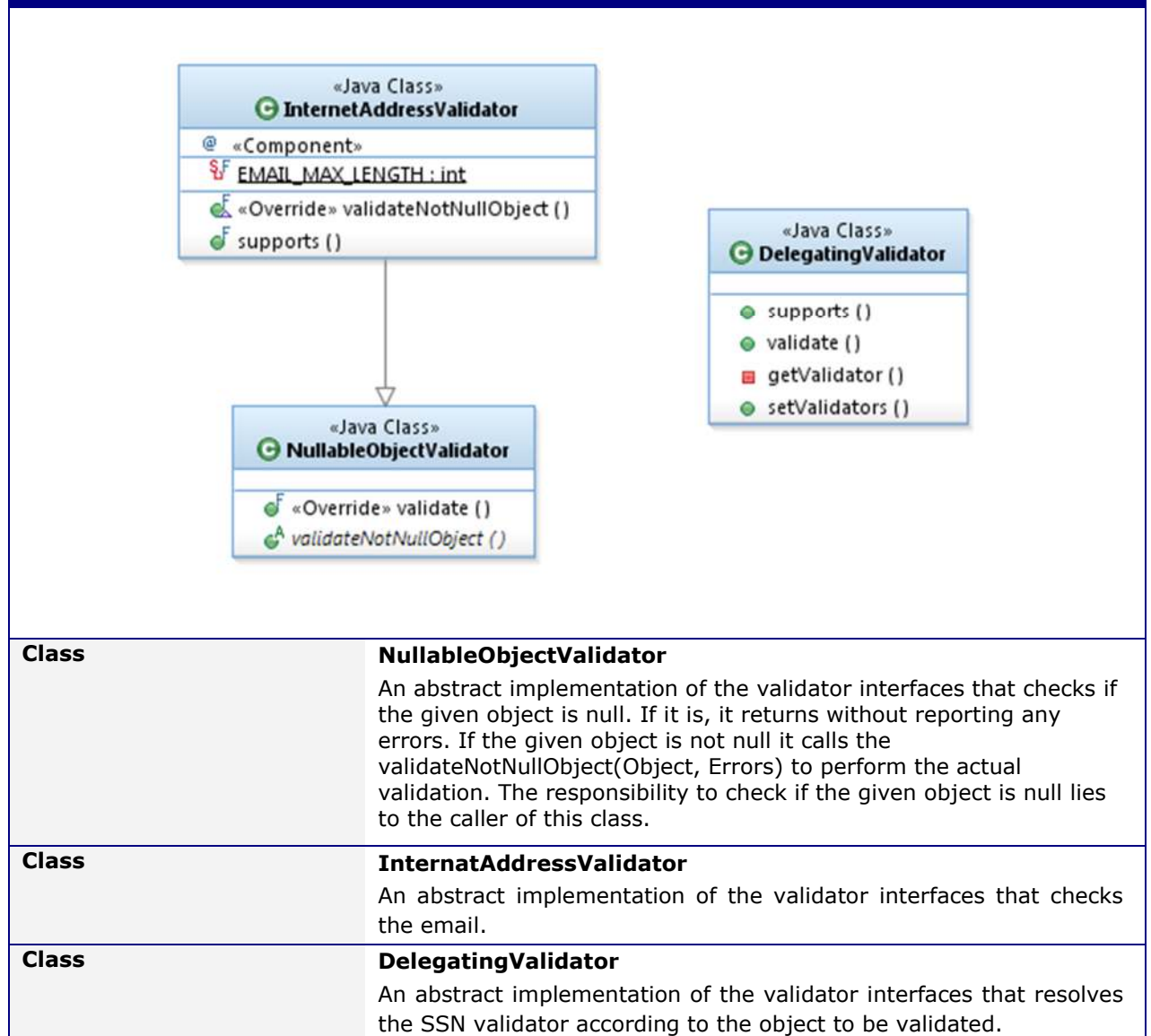


Class Diagram: support-jms



Class	Envelop This class it is used to model the information carried in a JMS message.
Class	EnvelopConverter This class implements the message converter; it creates an envelop from a JMS message and vice versa.
Class	EnvelopProcessorRetryCountAdvice This class implements the interceptor handles the failures of JMS message processing. In case of IOException on message processing of the SSN outgoing JMS queue items, the message remains on the queue to be reprocessed – the “retry count” application parameter is taken also into account.
Class	AbstractEnvelopProcessorMdb An abstract processor implements the JMS Message Listener interface (onMessage() method).
Interface	EnvelopProcessor An interface defines the JMS queues’ message processing.
Class	EnvelopQueueAccessor Implements the EnvelopProcessor interface; it is used by SsnMessageQueueSenderImpl.

Class Diagram: support- validation



4.2.3 Security on the ssn-core-app

The ssn-core-app application security relies on:

- The system level security provided by the runtime environment (SSL and Client Certificate).
- The application level security offered by ssn-core-app.

More specifically on the application level security, each message carries the credentials of its sender. So ssn-core-app uses only this form of credentials to authenticate a message.

The **ssn-core-app** application uses the sender of the message and the type of the message to perform authorization decisions.

The authorization methods are implemented by the **AuthorizationManagerImpl** class – located under the package **ssn.message.manager**.

Limitations: **ssn-core-app** must be compliant with the current version of the SSN protocol for exchanging XML messages. So, **ssn-core-app**:

- Must use a rather weak form of credentials (only username) in order to authenticate messages

Cannot leverage advanced non-repudiation techniques such as XML Signature¹ and XML Encryption².

4.3 SSN Resources Core Application - ssn-resources-core-app

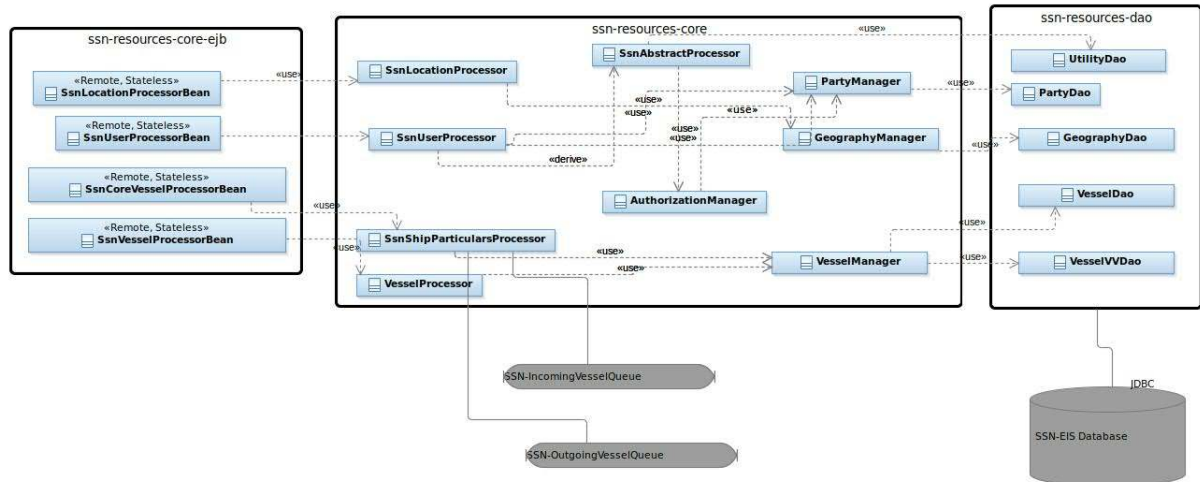


Figure 4-5 SSN Resources Core Application - ssn-resources-core-app

The application is constituted from the modules:

1. **ssn-resources-dao:** This module implements the resources data access. It *decouples application code from data access code*.
2. **ssn-resources-core:** It is the heart of application. This module implements the SSN resources functionality.
3. **ssn-resources-core-ejb:** It is a bundle of lightweight EJB's that lend Remote Access semantics to the organisation and location services of ssn-resources-core.

4.3.1 SSN Resources Core Main Components

4.3.1.1 SSN Resources Data Services module

The basic components are listed in Table 4-6

	Component	Description
1.	PartyDao	Database data access object for the SSN users.
2.	VesselDao	Database data access object for the Vessels.
3.	VesselVVDao	Database data access object for the Vessels Verify and Validate codes.
4.	GeographyDao	Database data access object for the Location codes.

¹XML Signatures provide integrity, message authentication, and/or signer authentication services for data of any type, whether located within the XML that includes the signature or elsewhere. (ref: W3C Recommendation 12 February 2002).

²Requirements on the encryption syntax, data model, format, cryptographic processing, and external requirements and coordination.(ref: W3C Note 04 March 2002)

	Component	Description
5.	UtilityDao	A data access object for executing the logging.

Table 4-6 ssn resources and vessel dao

4.3.1.2 SSN Resources Core module

The basic components are listed in Table 4-7

	Component	Description
1.	SsnVesselProcessor	<p>It is a Facade for the synchronous management of incoming messages.</p> <p>The implementation simply delegates the actual message processing to the VesselManager for the processing of vessel notifications and request messages.</p> <p>The SSN Vessel Processor synchronously replies with a ProcessResult on each incoming message processed.</p>
2.	SsnUserProcessor	<p>It is a Facade for the synchronous management of incoming user and organisation messages.</p> <p>The implementation simply delegates the actual message processing to the PartyManager for the processing of IdM user notifications and organisation COD announcement messages.</p>
3.	SsnLocationProcessor	<p>It is a Facade for the synchronous management of incoming location messages.</p> <p>The implementation simply delegates the actual message processing to the GeographyManager for the processing of operational temporarylocation notifications, and central location and country announcement messages.</p>
4.	VesselManager	<p>It provides the management of vessels (ships) and the logic of updating the vessel details based on the data from the Vessel and Message Notifications.</p> <p>The "resolve processing" of the incoming notifications' reported vessel data is the following (see also Sequence Diagram: Update Vessel Registry Using Reported vessel (IMO and MMSI Number, CallSign, Ship name and Flag in the case of PortPlus messages) in section 4.2.2.3.5):</p> <ul style="list-style-type: none"> c. for vessel that can be resolved – based on the rules defined for the vessel identification/validation procedure (refer also to CS-0202 Vessel V&V) – a reference (Foreign Key) will be made to the "Valid" or "Invalid" vessel; d. for a vessel that is not resolved– based on the rules defined for the vessel identification/validation procedure – a new "Temporary" vessel will be created and a reference (Foreign Key) will be made to that vessel. <p>The "Search Vessel" functionality, provided by the web console as the initial step of the "Send Notifications" interactive process, proposes ONLY the resolved – these are the vessels classified as</p>

	Component	Description
		<p>"Valid". If the user wishes to send a notification for a new vessel, not registered in the EIS OSD yet, the user is able to enter the Vessel Identification attributes (IMO and MMSI plus CallSign, ShipName and Flag in the case of PortPlus messages). On notification submit, the aforementioned "resolve processing" is executed to create a new Temporary vessel record.</p> <p>Similarly, the user is able – using the management console - to create a new vessel and /or update the vessel attributes (including the MMSI, CallSign, ShipName and Flag) identified by the IMONumber for a particular vessel version.</p>
5.	AuthorizationManager	It is a Facade for the authorisation of incoming xml messages. It uses the party manager
6.	PartyManager	It provides the IdM user CRUD management, COD organisation management, , as well as the user authorization.
7.	GeographyManager	It provides the management of operational temporary locations as well as the CRUD management of central locations and countries.

Table 4-7 ssn-resources-core

4.3.2 UML Class and Sequence Diagrams

This section covers the architectural significant elements of the design model. It presents the definition of the most significant classes that will implement the requested functionality, organised into packages.

The UML Sequence Diagrams also presented in this section, associate classes and depict the overall flow of control within the system components.

The classes are organised in packages according to the functionality they provide. A package is a general-purpose model element that organizes model elements into groups. Each package contains a set of classes and interfaces, representing what will become components in the implementation.

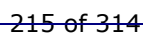
4.3.2.1 Module: ssn-resources-domain

This module/package includes the definition of SSN Resources entities (users, locations/countries, vessel).

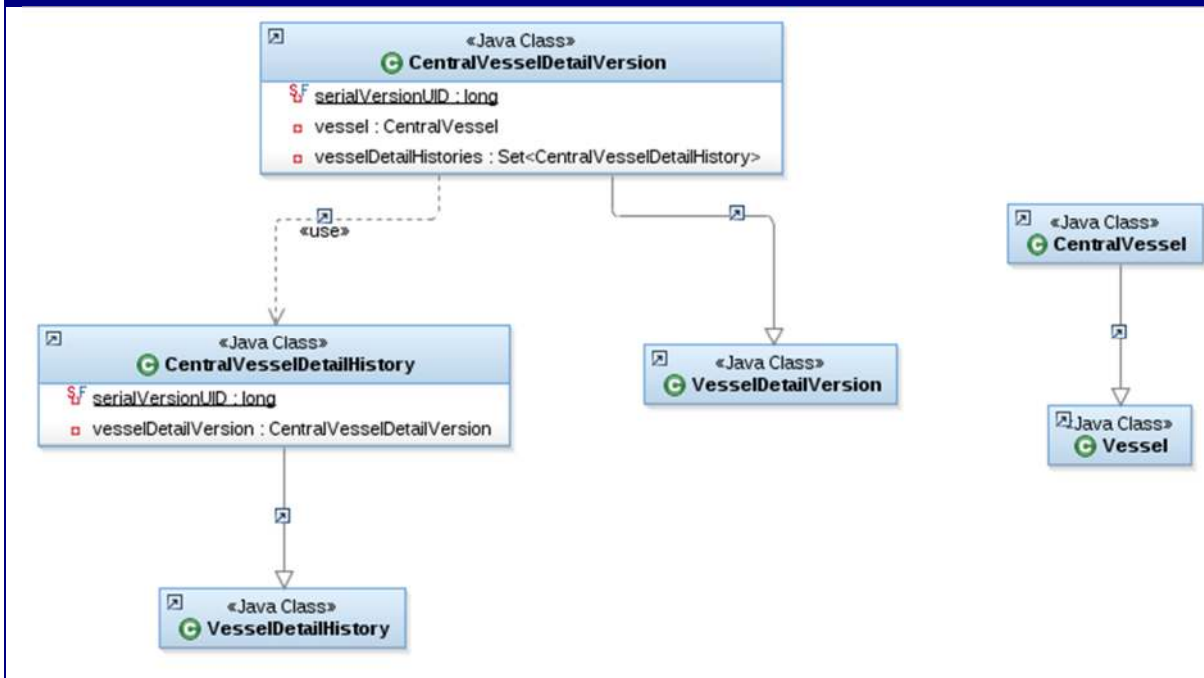
The aforementioned OOD, OLD, OSD and CSD data exchanges are associated to the entities described in this section. So, they shall be updated accordingly to incorporate the new requirements.

4.3.2.1.1 Package: vessel

Class Diagram: vessel



Class Diagram: vessel



Class	Imo This class represents an IMO number. EU Member states are expected to send notifications to SSN, using either the IMO or the MMSI in order to identify the vessel
Class	Mmsi This class represents the MMSI number of a vessel. The MMSI number is a secondary vessel identifier.
Class	MaritimeIdentificationDigit This class represents a maritime id of a vessel signified by the three first digits of the MMSI number.
Class	VesselIdentifier This class represents the vessel identified by the IMO number or CSDID. It also holds the ship particulars MMSI number, call sign, ship name, IR number, ship flag.
Class	Vessel This class represents the vessel stored in OSD and identified by the internal system id. It holds the vessel details described below, the set of images, the set of versions and the set of ship particulars history. It also includes the processing flag.
Class	CentralVessel This class extends the aforementioned one to represent the vessel stored on CSD.

<p>Class</p>	<p>VesselDetail</p> <p>This class extends the vessel identification to holds the vessel details such as</p> <ul style="list-style-type: none"> • the activated indicator showing whether the vessel is sea wordy, • the status vessel validation check (valid, non-valid, temporary), • the list of inmarsats, • the flags banned, detained, eligible for banning and for an expanded inspection, class society, eu recognised, Paris MoU recognised, • the type of ship, • the year of the shipconstruction, • the gross and net tonnage, • the length, • the beam, • the depth of a loaded vessel in the water, • the dead weight, • the types AIS / PSC / LMIU / UN / LRIT / Hull, • the main material that the hull is made of, • the classification of vessels according to IHS Fairplay, • the date when the contract to build the vessel is signed, • the number of bow thrusters, • the number of stern thrusters, • the number of cargo tanks of the ship, • the number of RORO compartments • the ICE class, • the inspection priority, • the risk profile, • the keel laying date, • the Status (from PSC): Active/Dead, • the max speed, • the max service speed, • the max maneuver speed, • the max number passengers, • maximum output a Power kW can produce over a year, • maximum Revolutions per minute of the main engine of the vessel, • the type of fuel that the main engine of the vessel uses, • the moulded breadth and depth, • the tool with which aquatic resources are captured by fishing vessels.
---------------------	--

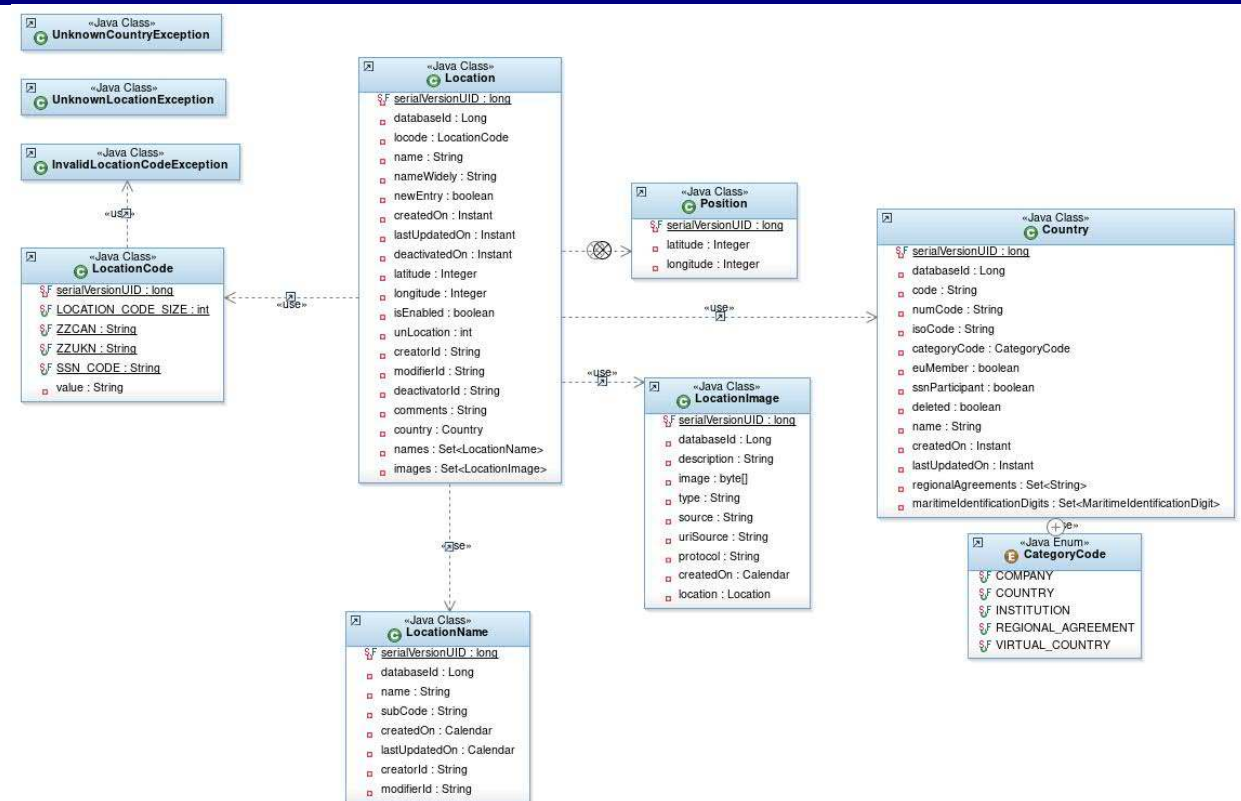
Class Diagram: vessel	
Class	VesselDetailVersion This class represents the versions of the vessel definitions in vessels history stored on OSD.
Class	CentralVesselDetailVersion This class extends the aforementioned one to represent the vessel version stored on CSD.
Class	VesselDetailHistory This class represents the update details of vessel particulars such as the reason and timestamp the updates became effective, as well as, the source and the relevant log entry stored on OSD.
Class	CentralVesselDetailHistory This class extends the aforementioned one to represent the vessel version history stored on CSD.
Class	VesselVVDetail This class represents the vessel entity stored in OSD in verify and validate process.
Class	VVInfo This class represents the vessel particulars per source, e.g. LMIU, MARS, SIRENAC in verify and validate process.
Class	VVDuplicateInfo This class represents the duplicate vessels for the provided IMO, MMSI, Call Sign and ship name in verify and validate process.
Class	VesselSearchCriteria This class represents the vessel identifiers used as search criteria: CSDID, IMO number, MMSI number, CallSign and ship Name patterns, the ship flag, the IR / ER number, the vessel type.
Class	VesselVVDetail This class represents vessel details used for the vessels validation and verification procedure provided by the management console
Class	VesselImage This class represents vessel image stores in OSD.
Class	UploadShtVessels A class represents the uploaded single hull tanker used by the upload "Single Hull Tankers" procedure in OSD.
Class	UploadMarsVessels A class represents the uploaded MARS vessels in both OSD and CSD.
Class	UploadMSVessels A class represents the uploaded MS vessels in both OSD and CSD.

Class Diagram: vessel

Class	CannotUpdateTestVesselException This exception is thrown by processNotification method in order to indicate that an undefined version of the test vessel is included in a notification message.
Class	CannotUpdateBannedVesselException This exception is thrown by processNotification method in order to indicate that a new version of a banned vessel is included in a notification message.
Class	InvalidVesselException This exception is thrown in order to indicate that a specific reported vessel is invalid.
Class	UnregisteredVesselException This exception is thrown in order to indicate that a specific vessel cannot be found in the OSD.

4.3.2.1.2 Package: geography

Class Diagram: geography



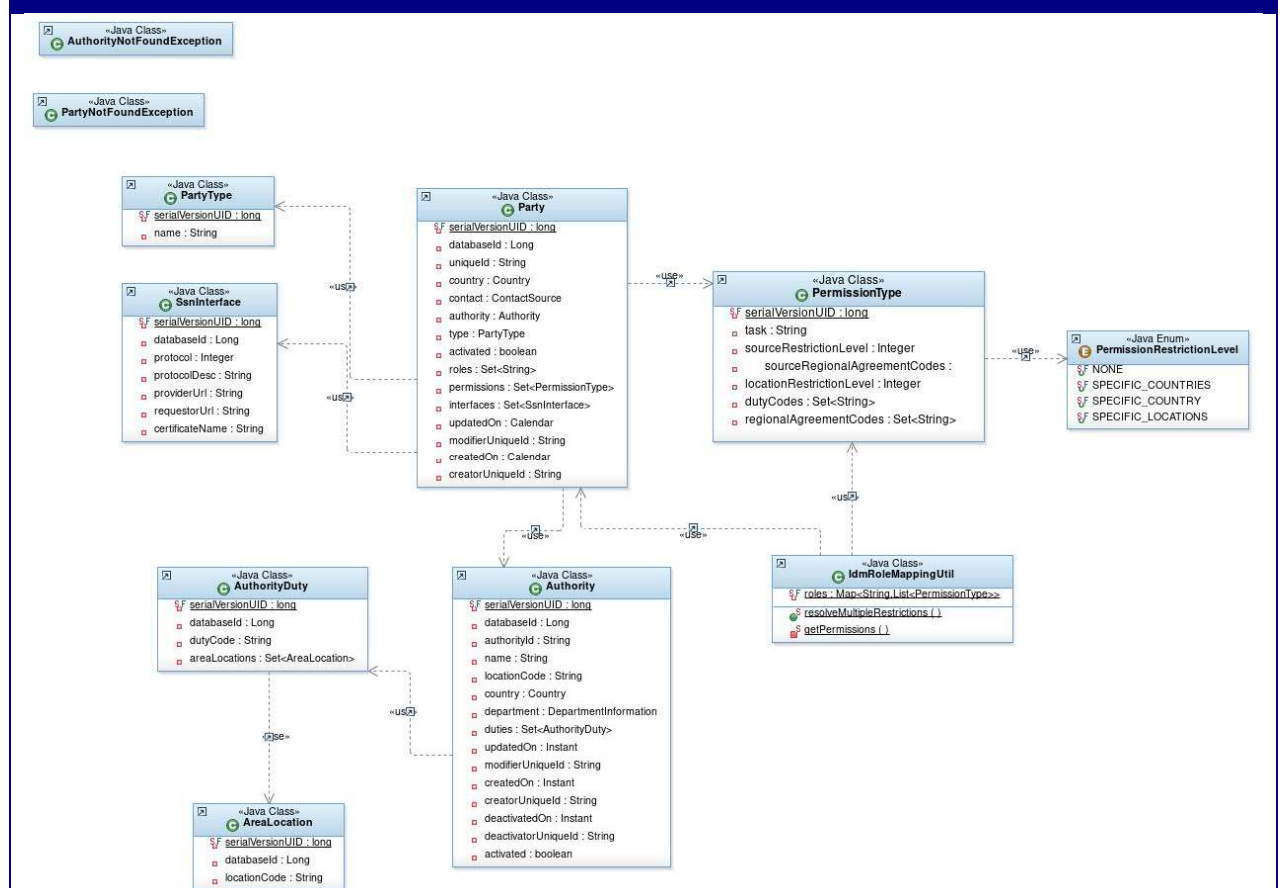
Class	CategoryCode It represents the category associated to a country (i.e. Country, Virtual Country, Regional Agreement, Institution, Company).
Class	Country Representation of the countries (EU and NON-EU) defined in OLR.
Class	Location Representation of the location codes defined in OLR. Types of location codes are: SSN Specific, UN Specific and Temporary (marked for validation).
Class	LocationName It represents the names associated to a location and stored on OLR.
Class	LocationImage It represents the images associated to a location and stored on OLR.
Class	LocationCode Representation of the actual LOCODE for the location codes in SafeSeaNet. LOCODEs may originate from UN or be SSN specific (e.g. for way points).
Class	Position Represents the position (longitude, latitude).
Class	UnknownLocationException Is thrown if the specified Location is unknown.

Class Diagram: geography

Class	InvalidLocationCodeException Is thrown if the specified LocationCode is not valid.
Class	UnknownCountryException Is thrown if the specified Country is unknown.

4.3.2.1.3 Package: organisation

Class Diagram: organisation

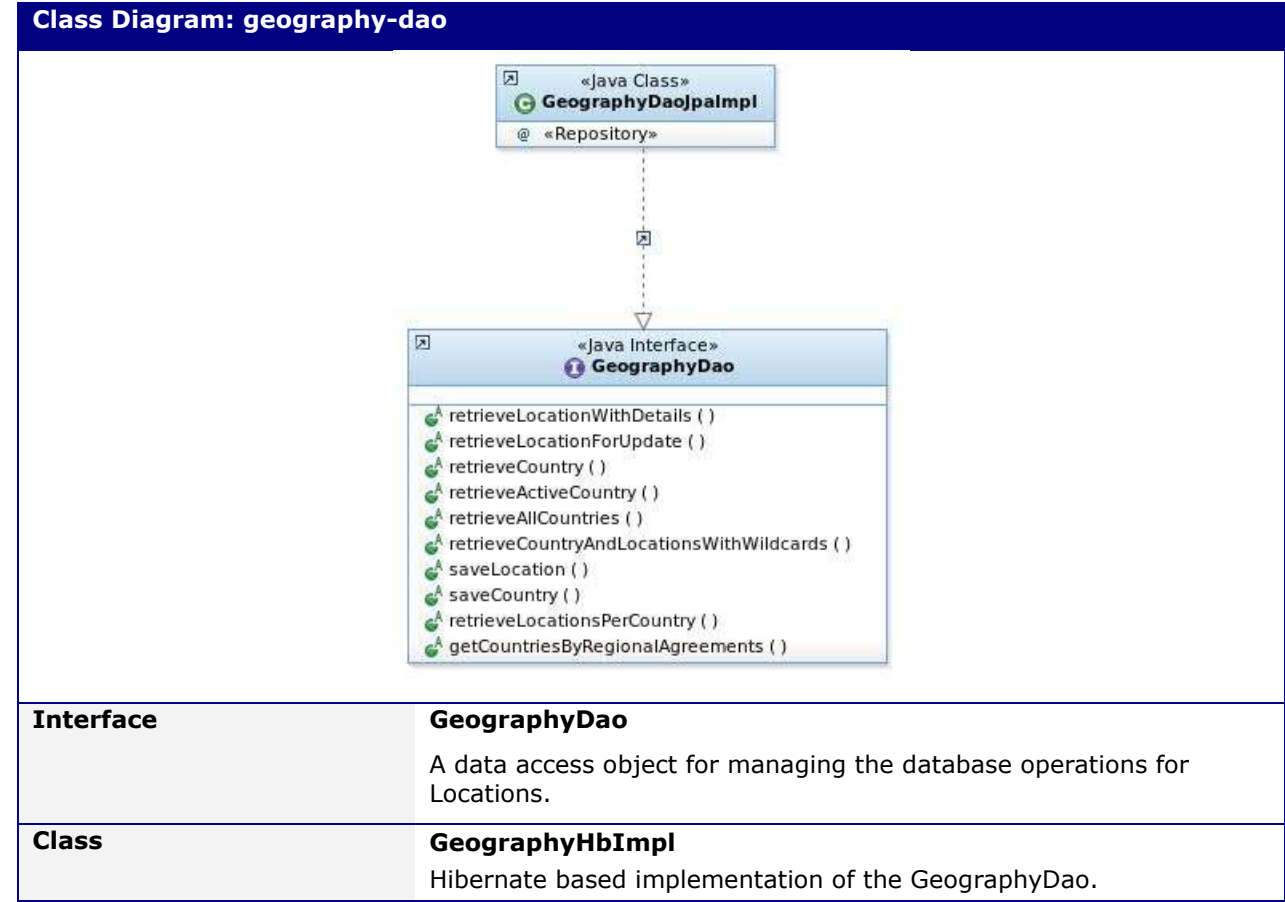


Class	AreaLocation <u>Representation of the area codes (LOCODES) associated to an AuthorityDuty.</u>
Class	Authority <u>Representation of the COD organisations in the SSN eco system stored on OOD. An authority can be assigned to a party.</u>
Class	AuthorityDuty Representation of the duties assigned to an authority.
Class	AuthorityNotFoundException Is thrown if the specified Authority is unknown.

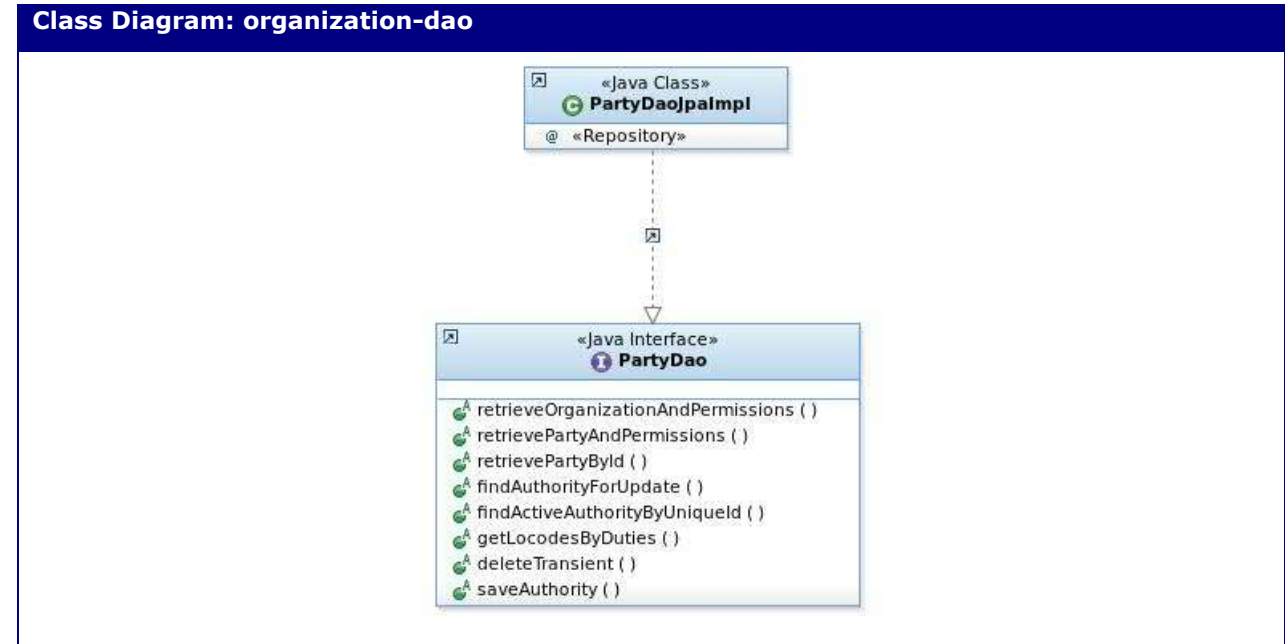
Class Diagram: organisation	
Class	IdmRoleMappingUtil Class is responsible for loading of IdM roles together with their corresponding set of tasks and restrictions from configuration file resource as well as for resolving source and restriction level information in case the same task is included in several IdM roles. The system constructs on application startup an in "memory" map data structure that accepts each IdM role as key and a collection of tasks together with their respective restrictions as mapped value. From that point onward, for each IdM role assigned to a user, the system retrieves from memory the list of mapped tasks together with their source and location restriction level information.
Class	Party Representation of the Parties (Users and Organisations) in the SSN eco system stored on OOD.
Class	PartyNotFoundException Is thrown if the specified Authority is unknown.
Class	PartyType Representation of the Party types: Persons ("Human" users) and Organisations ("System" users)
Class	PermissionType Representation of the Permission types (SendPortNotifications etc) could be assigned to a party
Class	PermissionRestrictionLevel Represents geographical restriction level values associated to either source restriction (country of the provider of the data), or location restriction (location to which the data concerns). Possible values are: <ul style="list-style-type: none"> • Specific locations • User's country • Countries list • None
Class	SsnInterface Representation of the interfaces associated to a Party, needed for its communication with SSN.

4.3.2.2 Module: ssn-resources-dao

4.3.2.2.1 Package: geography-dao



4.3.2.2.2 Package: organization-dao



Class Diagram: organization-dao

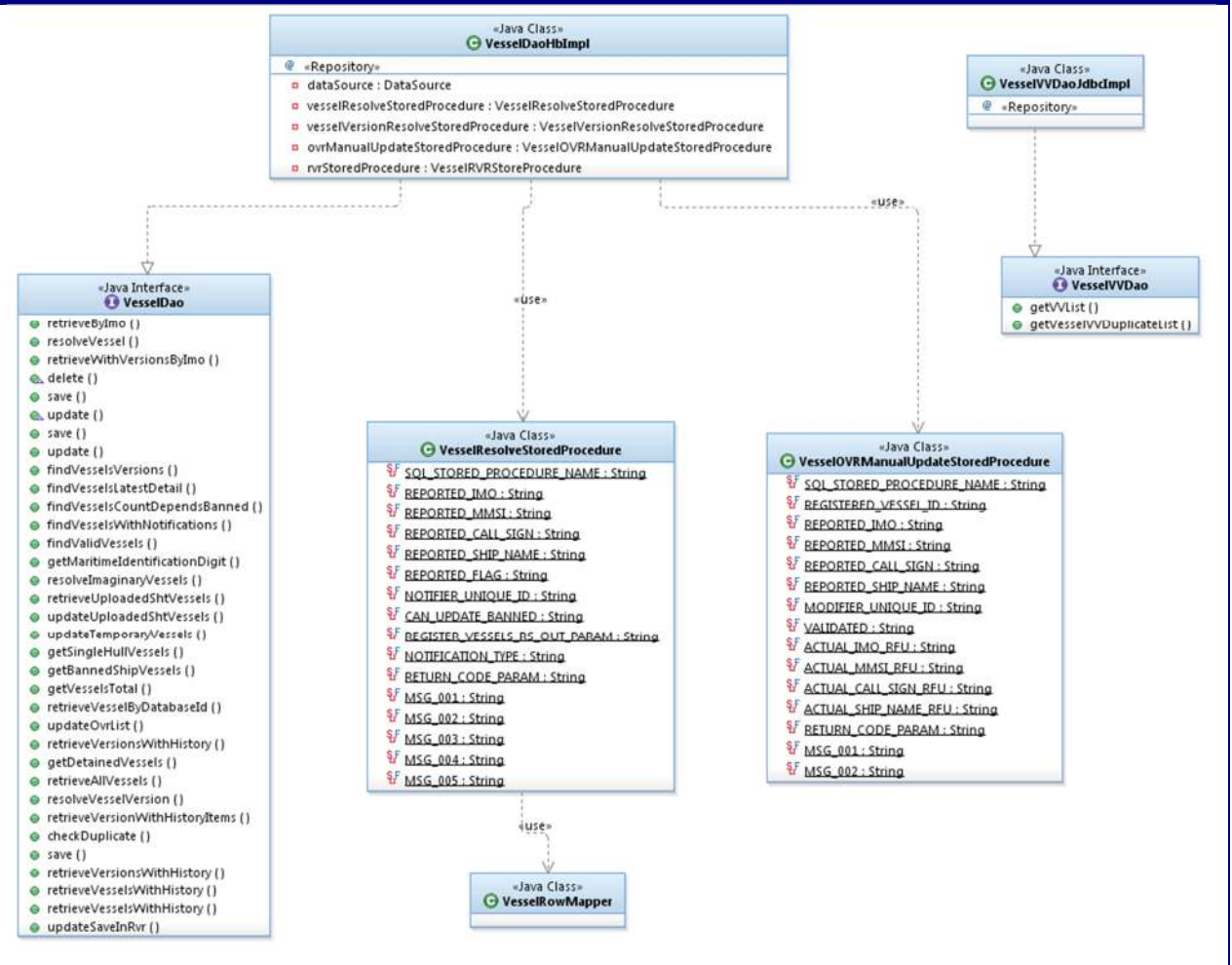
Interface

PartyDao

A data access object for managing the database operations for parties and their permissions.

4.3.2.2.3 Package: vessel-dao

Class Diagram: vessel-dao



Interface

VesselDao

A data access object for managing the database operations for vessels

Class

VesselDaoHbImpl

Hibernate based implementation of the VesselDao.

Class

VesselResolveStoredProcedure

A class used to execute the RDBMS stored procedure that implements 'the vessel's resolve on notification arrival'

Class

VesselOVRManualUpdateStoredProcedure

A class used to execute the RDBMS stored procedure that implements the 'OSD manual update' that is part of the vessels validation and verification procedure.

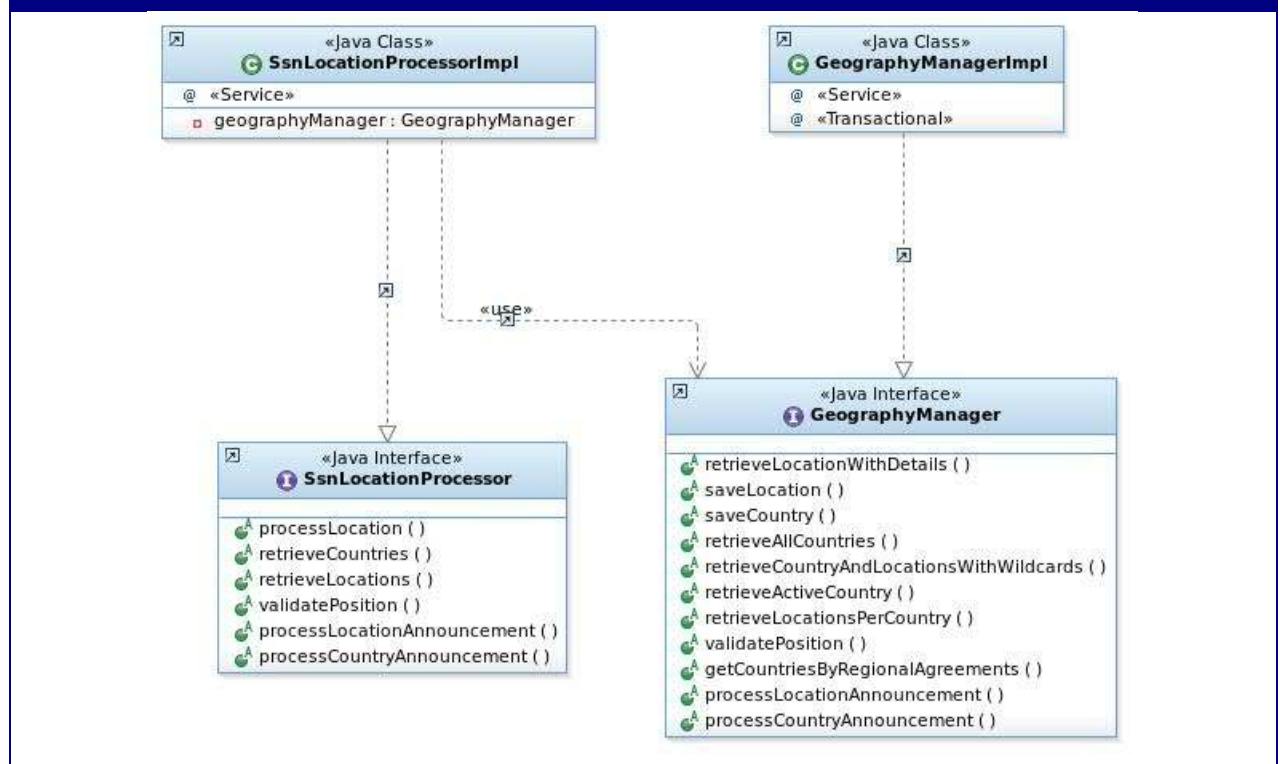
Class Diagram: vessel-dao

Interface	VesselIVVDao A data access object for managing the database operations for vessels validation and verification procedure.
Class	VesselIVVDaoJdbcImpl JDBC based implementation of the VesselIVVDao.

4.3.2.3 Module: ssn-resources-core

4.3.2.3.1 Package: geography-manager

Class Diagram: geography-manager



Interface	SsnLocationProcessor It is a Facade for the processing of incoming location notification and announcement messages.
Class	SsnLocationProcessorImpl Implementation of the SsnLocationProcessor. It uses the geography manager. The implementation simply delegates the actual message processing to the GeographyManager for the processing of operational temporary location notifications, and central location and country announcement messages.
Class	GeographyManagerImpl Default implementation of the LOCODE management processor. It provides the management of operational temporary locations as well as the CRUD management of central locations and countries received by CLD and CCD respectively.

Class Diagram: geography-manager

Interface

GeographyManager

Defines the processing logic of the LOCODEs, countries.

4.3.2.3.2 Package: geography-validation

Class Diagram: geography-validation



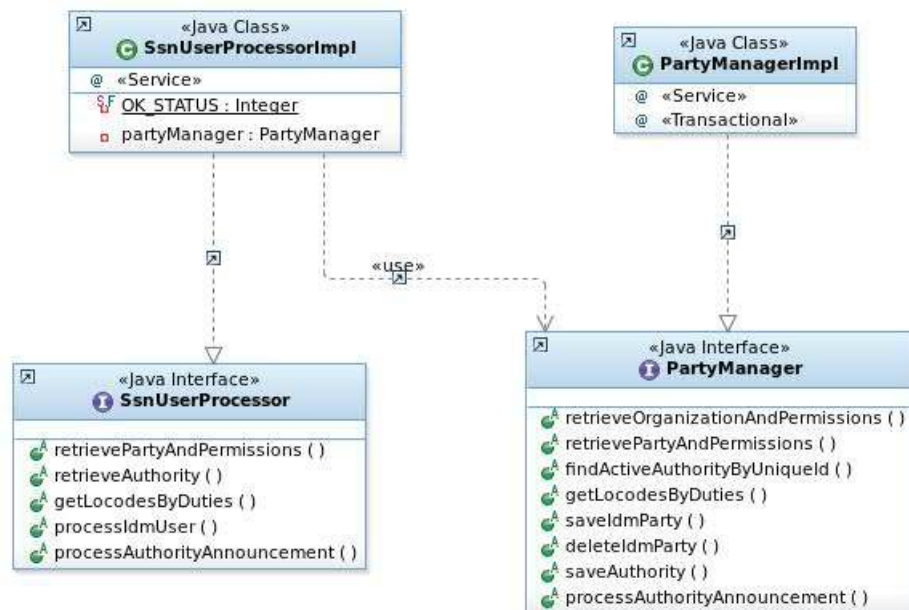
Class

PositionValidator

The position business rules (applicable only to operational temporary locations).

4.3.2.3.3 Package: organisation-manager

Class Diagram: organisation-manager



Interface

SsnUserProcessor

It is a Facade for the processing of incoming xml messages.

Class

SsnUserProcessorImpl

Implementation of the SsnUserProcessor. It uses the party manager.

Interface

PartyManager

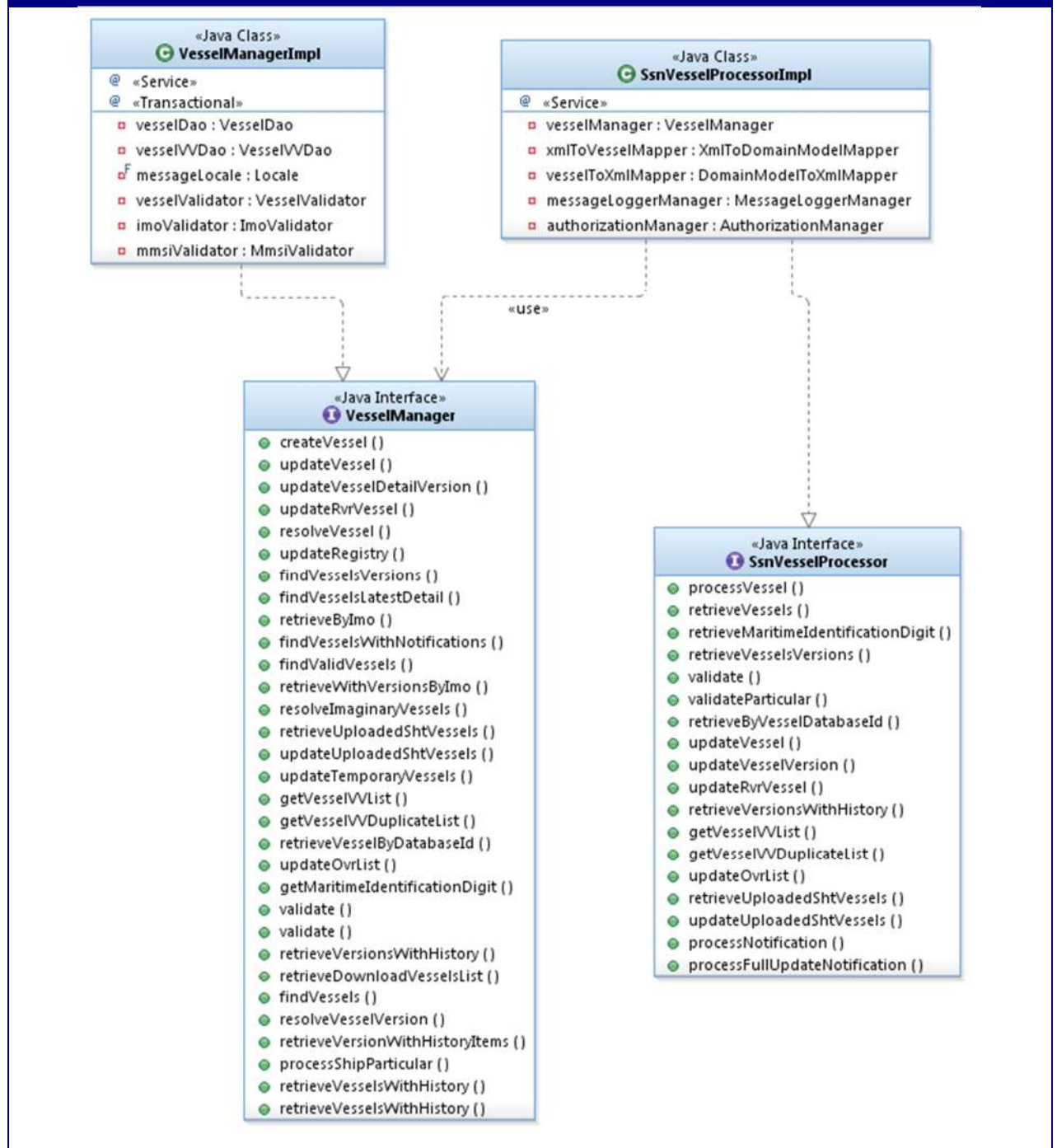
Defines the processing logic of the parties (SSN user management).

Class Diagram: organisation-manager

Class	PartyManagerImpl Default implementation of the parties' management processor.
Class	PartyNotFoundException Exception is thrown when a requested party is not defined in the database.

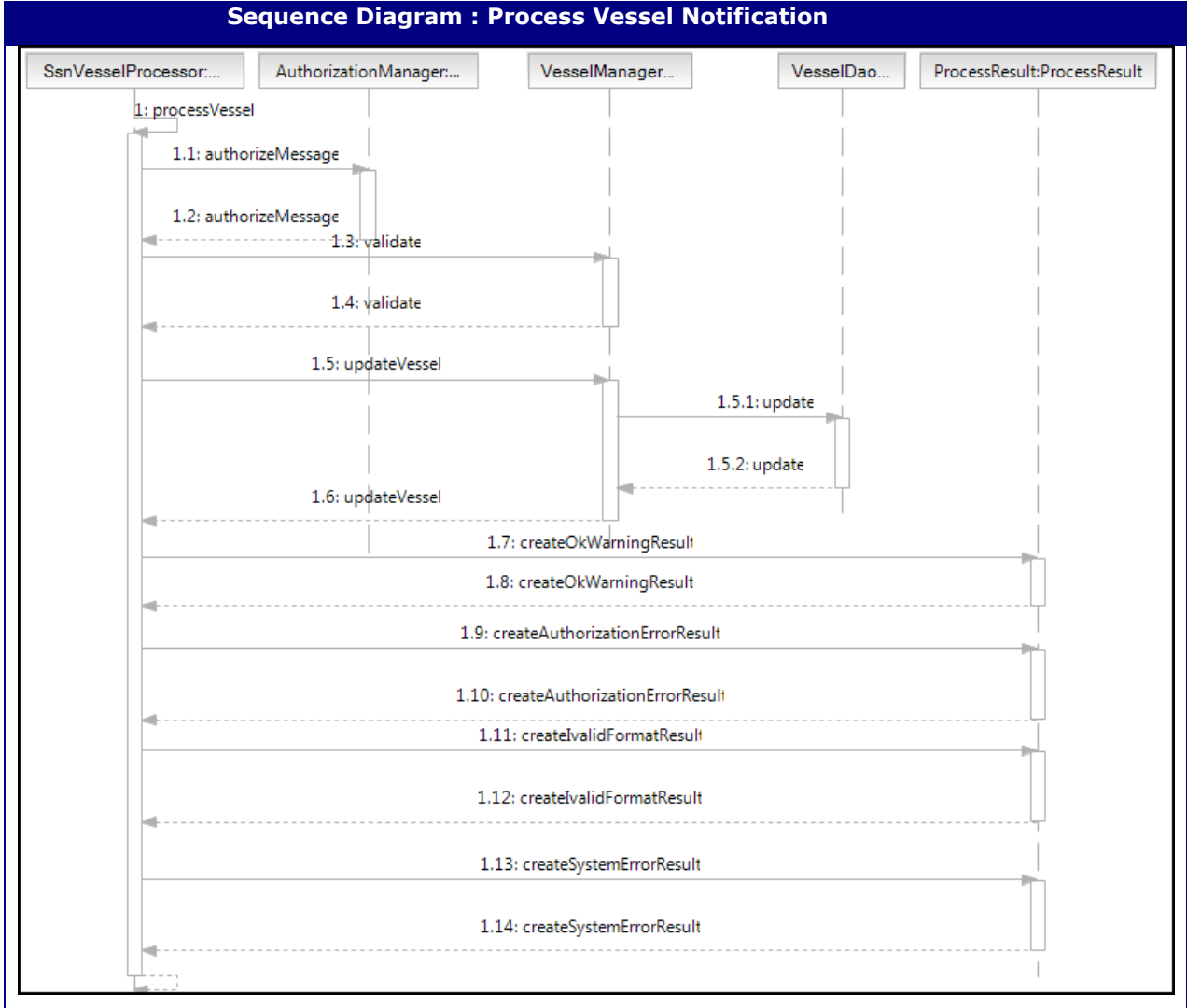
4.3.2.3.4 Package: vessel-manager

Class Diagram: vessel-manager



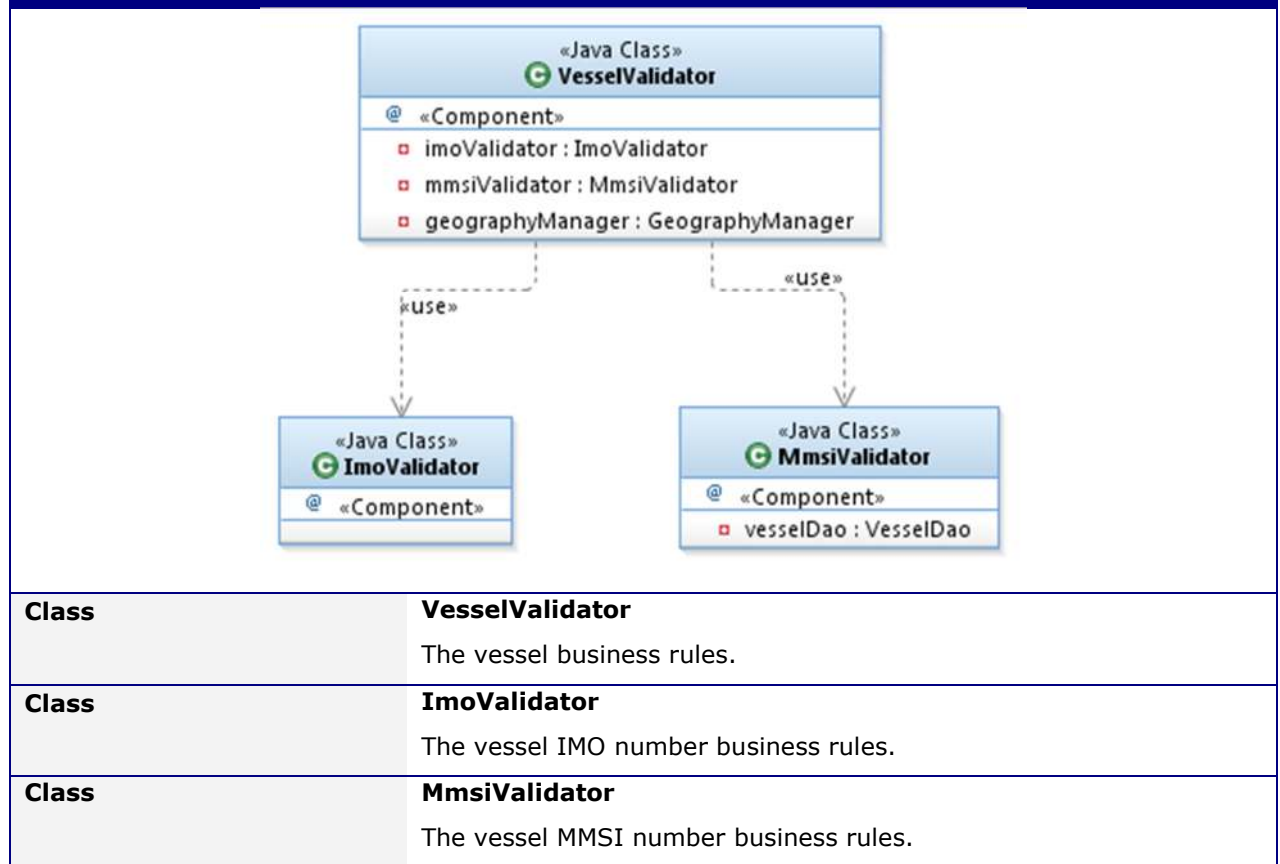
Interface	SsnVesselProcessor It is a Facade for the processing of incoming xml messages. It uses the vessel manager.
Class	SsnVesselProcessorImpl Implementation of the SsnVesselProcessor.
Interface	VesselManager Interface providing the management options for vessel registry.

Class Diagram: vessel-manager	
Class	VesselManagerImpl
	Implementation of the VesselManager.



4.3.2.3.5 Package: vessel-validation

Class Diagram: vessel-validation



4.4 SSN Console Application - ssn-console-app

The application is constituted from the following web applications:

1. .
1. **ssn-web-common** component is a library that provides
 - the spring framework functionality as spring web MVC, spring web flow and spring security. Regarding spring security authorization,
 - a custom SSN authentication processing filter intercepts any request and attempts to perform authentication, if no authentication object is placed in the spring security context, by retrieving the SM_USER HTTP header token from the request. Based on the token value it then loads the IdM user with its IdM roles from SSN EIS DB, creates the spring-security user with the the authorities granted (IdM roles) and stores user information which is encapsulated into the Authentication object. Further authorization will be handled by the application solely considering the granted authorities (IdM roles);
 - the JSF framework functionality;
 - the [RichFaces](#) JSF implementation;
 - ESAPI functionality.

It is used by all the following UI components;

2. **send-notification-console**: This web application is used for sending Notifications.
3. **find-notification-console**: This web application is used for sending Requests for details.

4. **reports-statistics-console:** This web application is used for EIS reports generation.
5. **application-management-console:** It is a Web application, which will be used by the EIS users for the management of EIS assets.

The web applications do not contain distinguishable components and is simply a UI. It is simply using the functionality provided by the ssn-core-app.

4.5 SSN XML Protocol Application - ssn-xmlprotocol-app

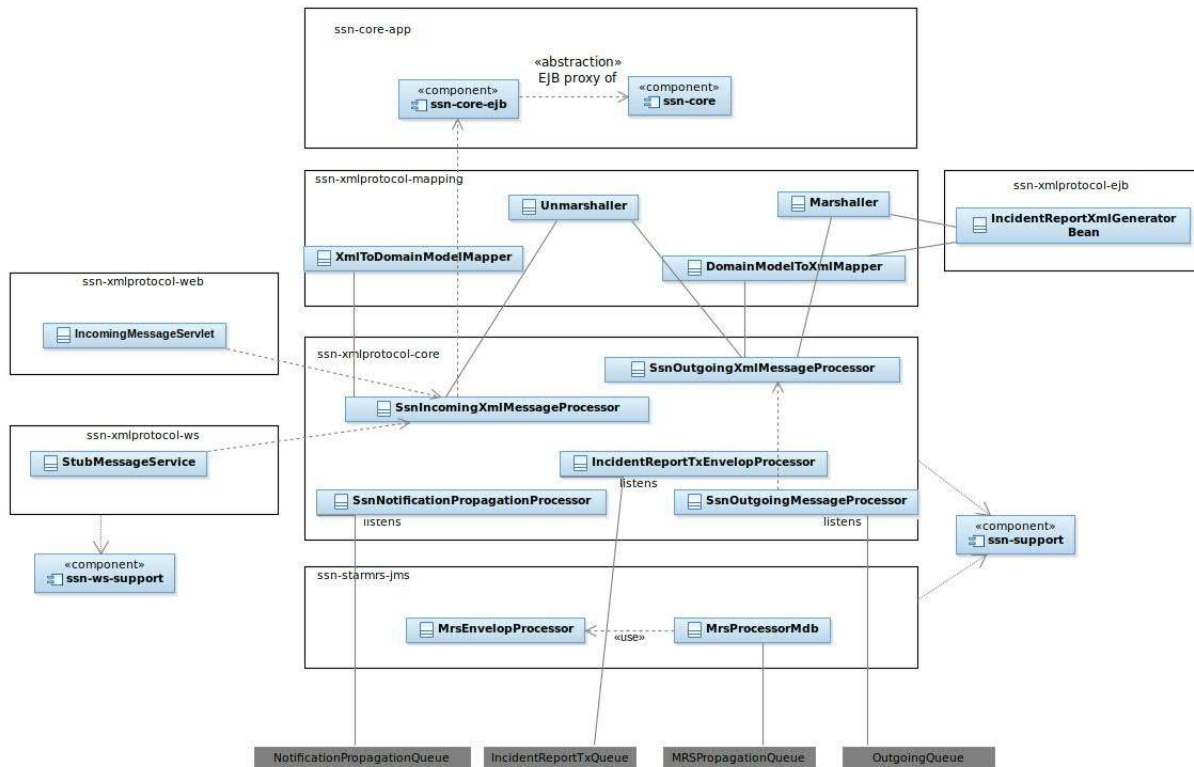


Figure 4-6 SSN XML Protocol Application - ssn-xmlprotocol-app

The followings modules constitute the application:

1. **ssn-xmlprotocol-mapping:** It provides the functionality of representing the XML messages in Java Objects - Data Transfer Objects (DTO's) - and the mapping of these objects to domain model - Domain Objects (DO's)- of ssn-core and vice versa.
 DTO's are also an important part of the design in an SOA environment where the Domain object model structurally is not compatible with the messages that are received and sent from a business service. The messages are typically defined and maintained in as XML Schema Definition documents (XSD's) and it's a common practice to write (or code generate) DTO objects from the XSD's and use them for data (message) transfer purposes between domain and SOA service layers.
2. **ssn-xmlprotocol-core:** It provides the management of the incoming and outgoing Xml messages independent from the channel of communication. It uses the ssn-xmlprotocol-mapping for the mapping of the XML messages to domain model object and it calls directly ssn-core-app in order to process the messages. It receives - asynchronously -
 - a. outgoing messages from ssn-core-app and dispatches them via HTTP to the recipients (data Requestors),
 - b. propagation messages from ssn-core-app and dispatches them via the mail session (SMTP) to the recipients.

3. **ssn-xmlprotocol-web**: It receives incoming XML messages from the HTTP communication channel and delivers them to ssn-xmlprotocol-core.
4. **ssn-xmlprotocol-ws**: It receives incoming SOAP messages from the HTTP communication channel and delivers them to ssn-xmlprotocol-core.
5. **ssn-xmlprotocol-ejb**: It provides reporting functionality for ssn-find-notification-console IR messages to XML format.
6. **ssn-starmrs-jms**: A module responsible for mapping mrs notification objects to xml for STAR MRS.

4.5.1 SSN XML Protocol Main Components

4.5.1.1 SSN XML Protocol Mapping Module - ssn-xmlprotocol-mapping

The basic components are listed in Table 4-8.

	Component	Description
1	Marshaller	It provides the possibility of representing a Java object in XML.
2	Unmarshaller	It provides the possibility of representing XML in a Java object.
3	DomainModelToXmlMapper	It provides the correspondence of a domain model object in ssn-core to a Java object representing XML.
4	XmlToDomainModelMapper	It provides the correspondence of a Java object that represents XML in the domain model of ssn-core.

Table 4-8 ssn-xmlprotocol-mapping

4.5.1.2 SSN XML Protocol Core module: ssn-xmlprotocol-core

The basic components are listed in Table 4-9.

	Component	Description
1	SsnIncomingXmlMessageProcessor	It provides the synchronous management of incoming XML messages - independent from the channel of communication they were received.
2	SsnOutgoingXmlMessageProcessor	Transforms outgoing messages in XML independent from the channel of communication in which they will be delivered.
3	SsnNotificationPropagationProcessor	Transforms notification messages to e-mail messages to be propagated via SMTP email server.

Table 4-9 ssn-xmlprotocol-core

4.5.1.3 SSN XML Protocol Web module - ssn-xmlprotocol-web

The basic component is listed in Table 4-10.

	Component	Description
1	IncomingMessageServlet	It receives incoming XML messages from the HTTP communication channel and delivers them to ssn-xmlprotocol-core.

Table 4-10 ssn-xmlprotocol-web

4.5.1.4 SSN XML Protocol EJB module - ssn-xmlprotocol-ejb

The basic components are listed in Table 4-11.

	Component	Description
1	IncidentReportXmlGeneratorBean	– Session Bean that is responsible for transforming Incident Report messages to XML-string used for IR reporting.

Table 4-11 ssn-xmlprotocol-ejb

4.5.1.5 SSN XML Protocol WebServices module - ssn-xmlprotocol-ws

The ssn-xmlprotocol-ws module will be included into the existing ssn-xmlprotocol-app application because there are common parts like ssn-xmlprotocol-core or ssn-xmlprotocol-mapping. Furthermore, the ssn-xmlprotocol-ws will be implemented using Spring-WS which is integrated with Spring in accordance to the existing application modules.

The SOAP message is dispatched to an Endpoint implementation through mappings (EndpointMappings) that bound incoming SOAP messages to custom Endpoints (depending on the SOAP message itself). After retrieving the Endpoint, Spring-WS is searching for Adapters (EndpointAdapter) supporting the acquired endpoint. EndpointAdapter defines the way a SOAP message is converted in order to be handled by the endpoint.

The endpoint uses the SsnIncomingXmlMessageProcessor for the management of the incoming messages. The ssn-xmlprotocol-mapping is used in order to map JAXB elements to SSN Domain objects.

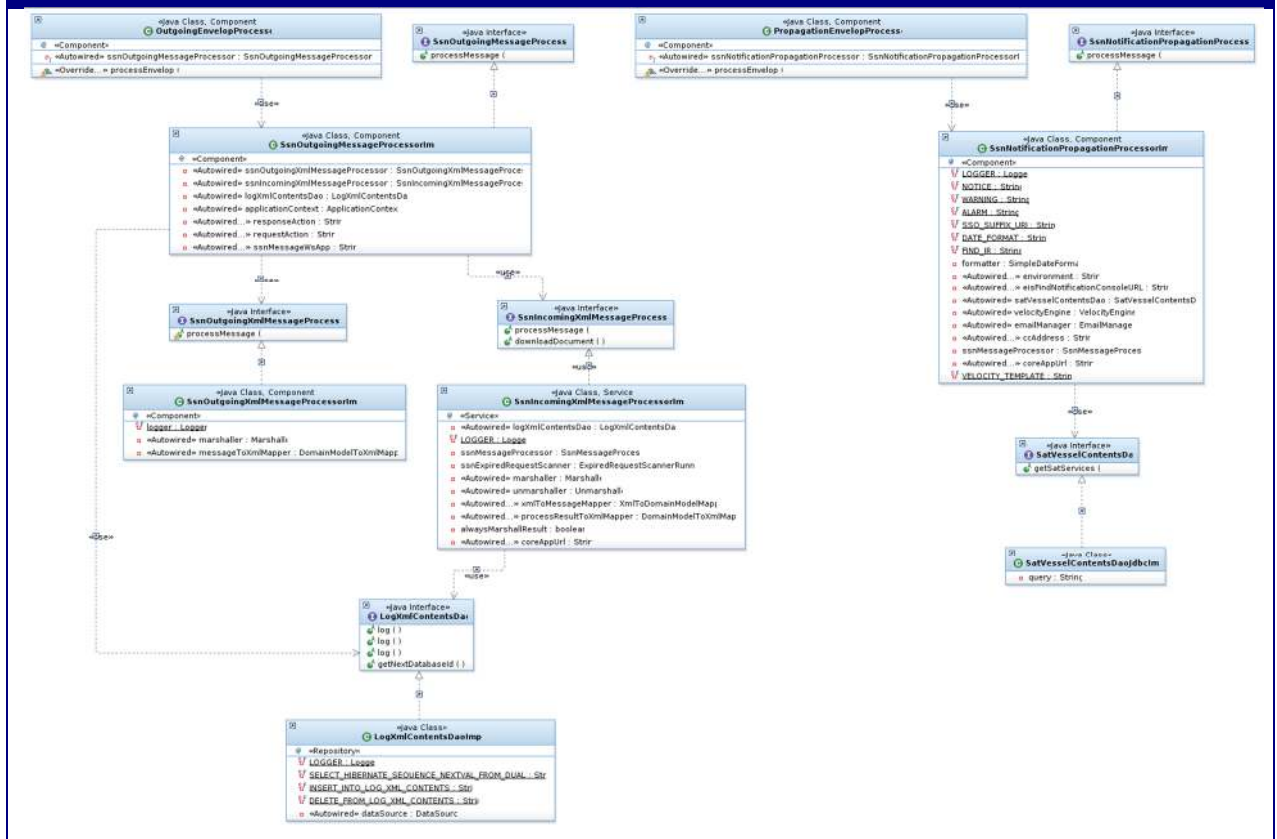
4.5.2 UML Class and Sequence Diagrams

This section covers the architectural significant elements of the design model. It presents the definition of the most significant classes that will implement the requested functionality, organised into packages.

The UML Sequence Diagrams also presented in this section, associate classes and depict the overall flow of control within the system components.

The classes are organised in packages according to the functionality they provide. A package is a general-purpose model element that organizes model elements into groups. Each package contains a set of classes and interfaces, representing what will become components in the implementation.

Class Diagram: xmlprotocol core

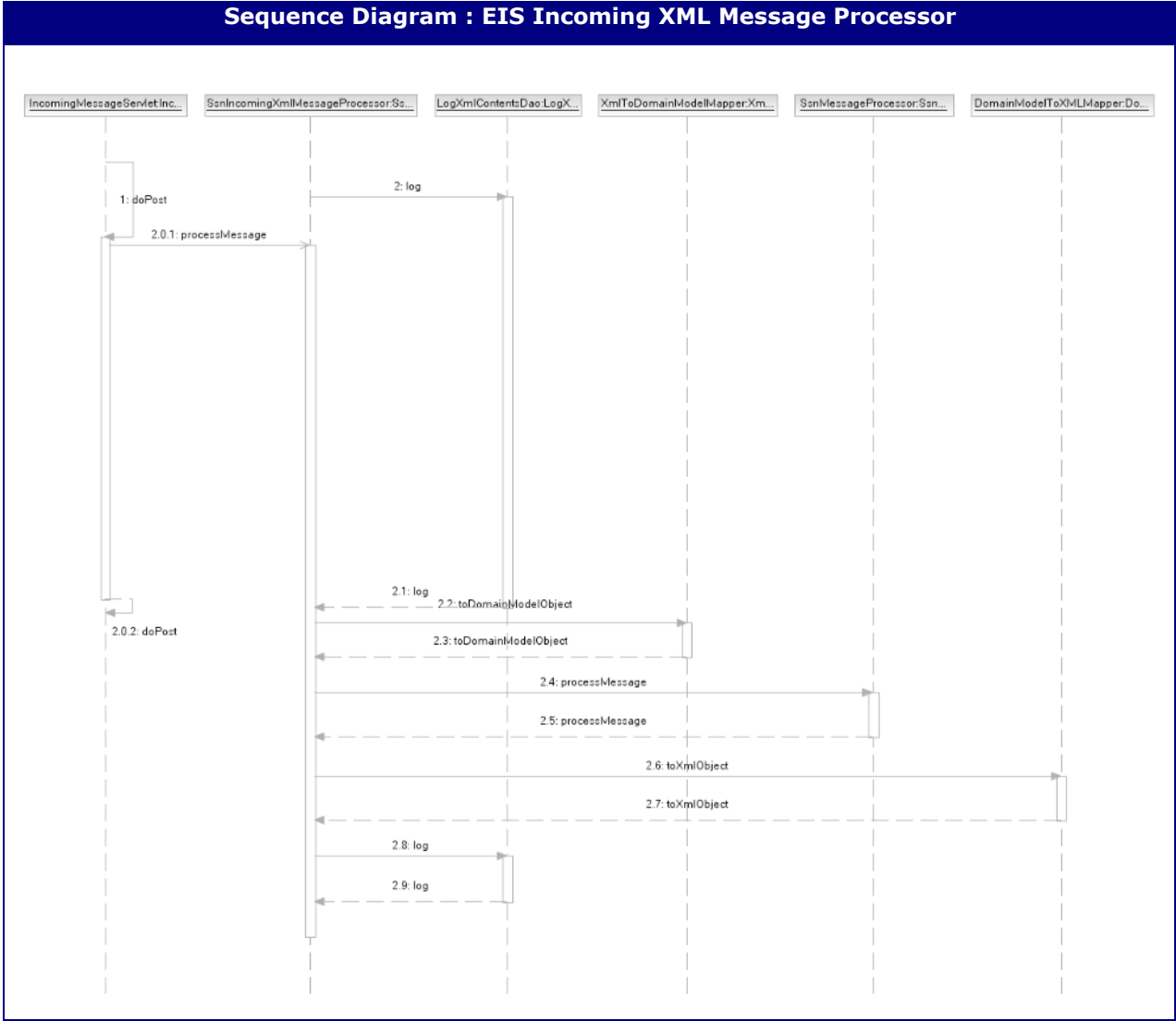


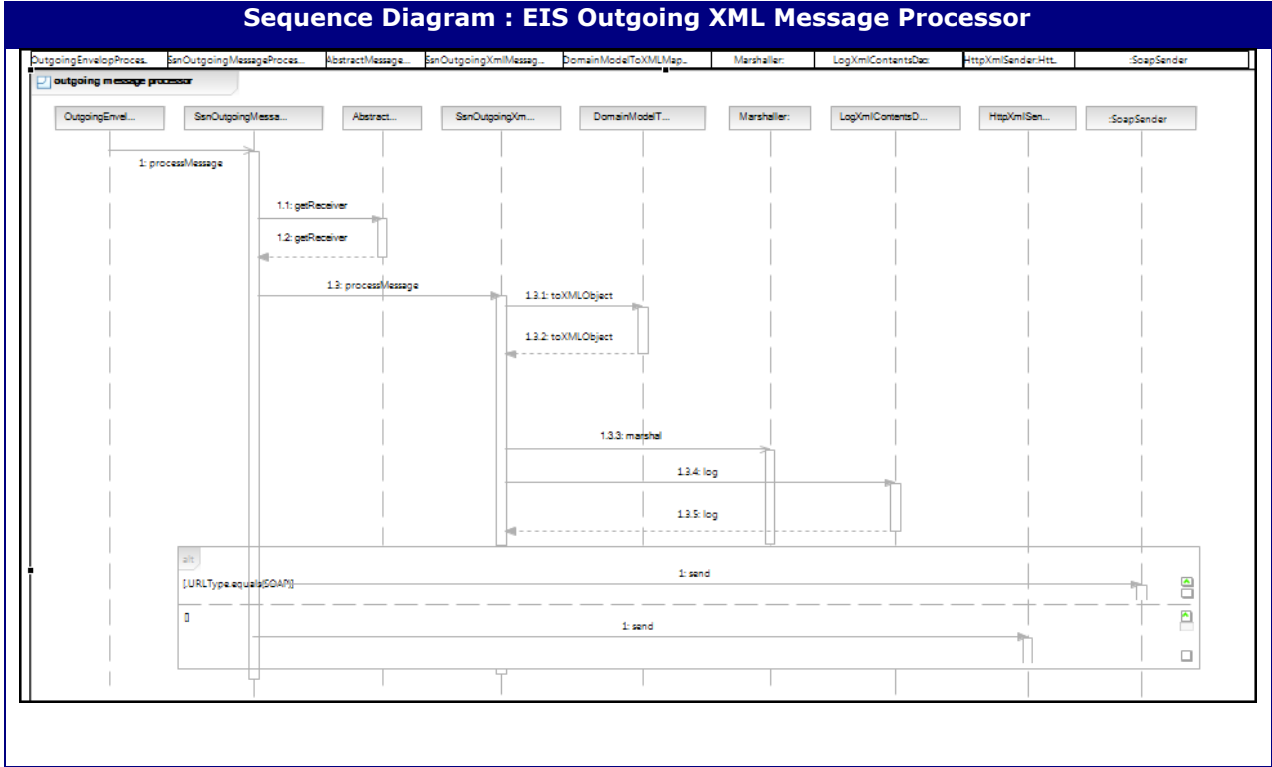
Class	OutgoingEnvelopProcessor It implements the EnvelopProcessor to process the messages located at the SSN outgoing queue. It uses the SsnOutgoingMessageProcessor for the message processing.
Interface	SsnOutgoingMessageProcessor Defines the processing of messages located at the SSN outgoing queue.
Class	SsnOutgoingMessageProcessorImpl Default implementation of the SsnOutgoingMessageProcessor interface. It uses the SsnOutgoingXmlMessageProcessor to transform the outgoing message to xml. Then, it sends the xml message using the HttpXmlSender / SoapSender according to the data requestor type. It uses the LogXmlContentsDao to log the xml message to EIS database.
Interface	SsnOutgoingXmlMessageProcessor Defines an interface that transforms the outgoing message to xml.
Class	SsnOutgoingXmlMessageProcessorImpl Default implementation of the SsnOutgoingXmlMessageProcessor interface.
Interface	SsnIncomingXmlMessageProcessor Defines the processing of incoming xml messages.

Class Diagram: xmlprotocol core

Class	<p>SsnIncomingXmlMessageProcessorImpl</p> <p>Default implementation of the SsnIncomingXmlMessage Processor interface. Handles SSN Receipt as a synchronous reply sent by the recipient of the outgoing message and delegates it to the core.</p> <p>This implementation follows the next processing procedure (see also Sequence Diagram: EIS Incoming XML Message Processor):</p> <ol style="list-style-type: none"> 1. Unmarshalls the incoming XML stream into a java object that represents this XML. 2. The provided message mapper is being used to map the previously generated java object into a domain model message. 3. The provided message processor is being called to process the domain model message and returns a process result. 4. The provided process result mapper maps the process result into a java object that represents XML. 5. Finally, the provided marshaller marshalls the previously generated object to the XML output stream. <p>It uses the LogXmlContentsDao to log the incoming xml message and the process result in xml format (SSN_Receipt) to EIS database.</p>
Interface	<p>LogXmlContentsDao</p> <p>Interface providing methods for logging the xml messages on EIS database</p>
Class	<p>LogXmlContentsDaoImpl</p> <p>JDBC based implementation of the LogXmlContentsDao.</p>
Class	<p>PropagationEnvelopProcessor</p> <p>It implements the EnvelopProcessor to process the messages located at the SSN notification propagation queue. It uses the SsnNotificationPropagationProcessor for the message processing.</p>
Interface	<p>SsnNotificationPropagationProcessor</p> <p>Defines the processing of messages located at the SSN notification propagation queue.</p>
Class	<p>SsnNotificationPropagationProcessorImpl</p> <p>Default implementation of the SsnNotificationPropagationProcessor interface.</p> <p>It creates an email according to the message</p> <ul style="list-style-type: none"> - Vessel notification with sat service: it uses the SatVesselContentsDao to retrieve the sat services in order to create the corresponding emails - - Incident Report notification: it creates an email to alert the notification tx recipients; it creates an email to non-acknowledged recipients. <p>It sends the email message using the EmailManager.</p>
Interface	<p>SatVesselContentsDao</p> <p>A dao that retrieves the sat services.</p>

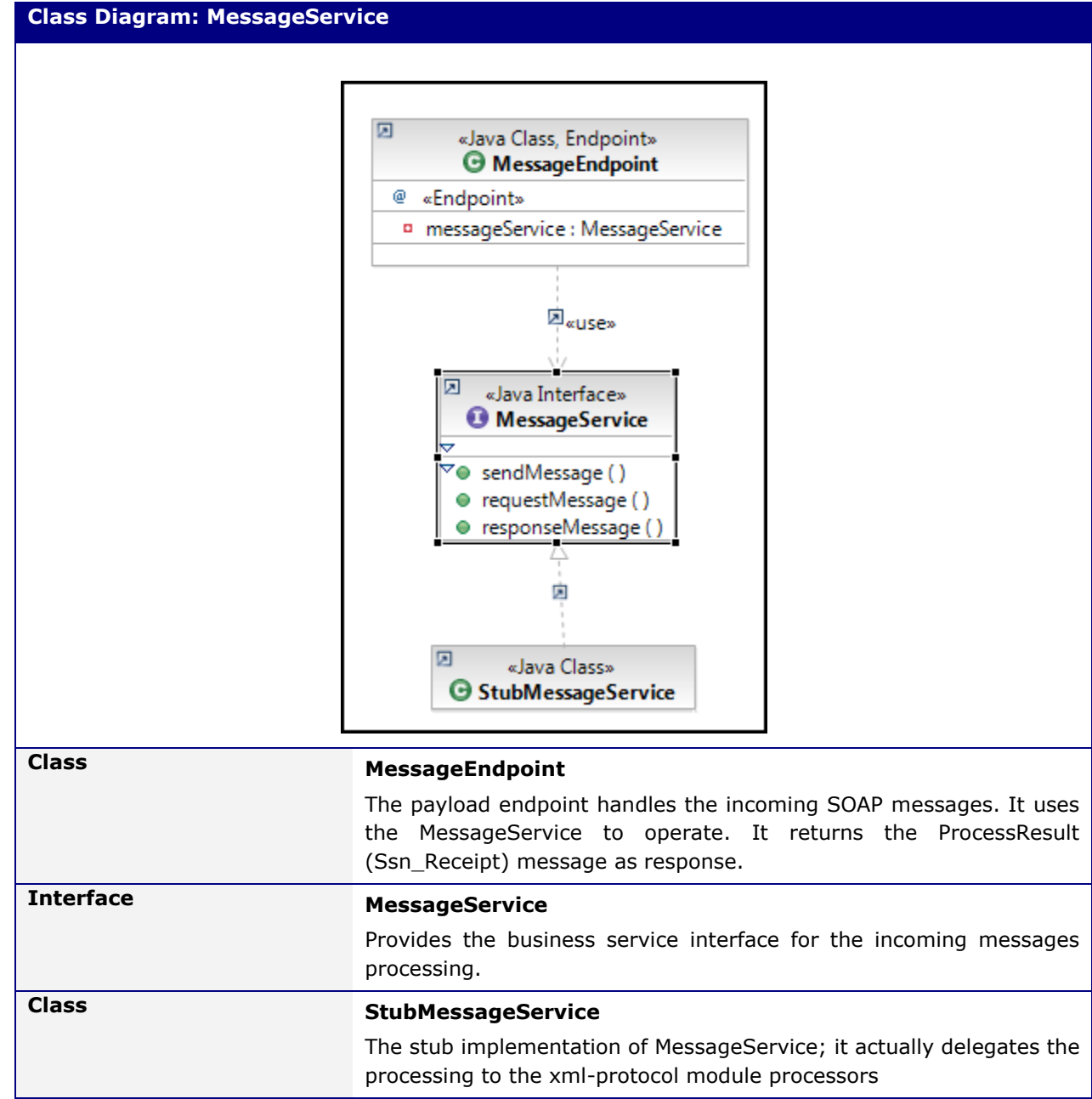
Class Diagram: xmlprotocol core	
Class	SatVesselContentsDaoJdbcImpl
	JDBC based implementation of the SatVesselsContentsDao interface.



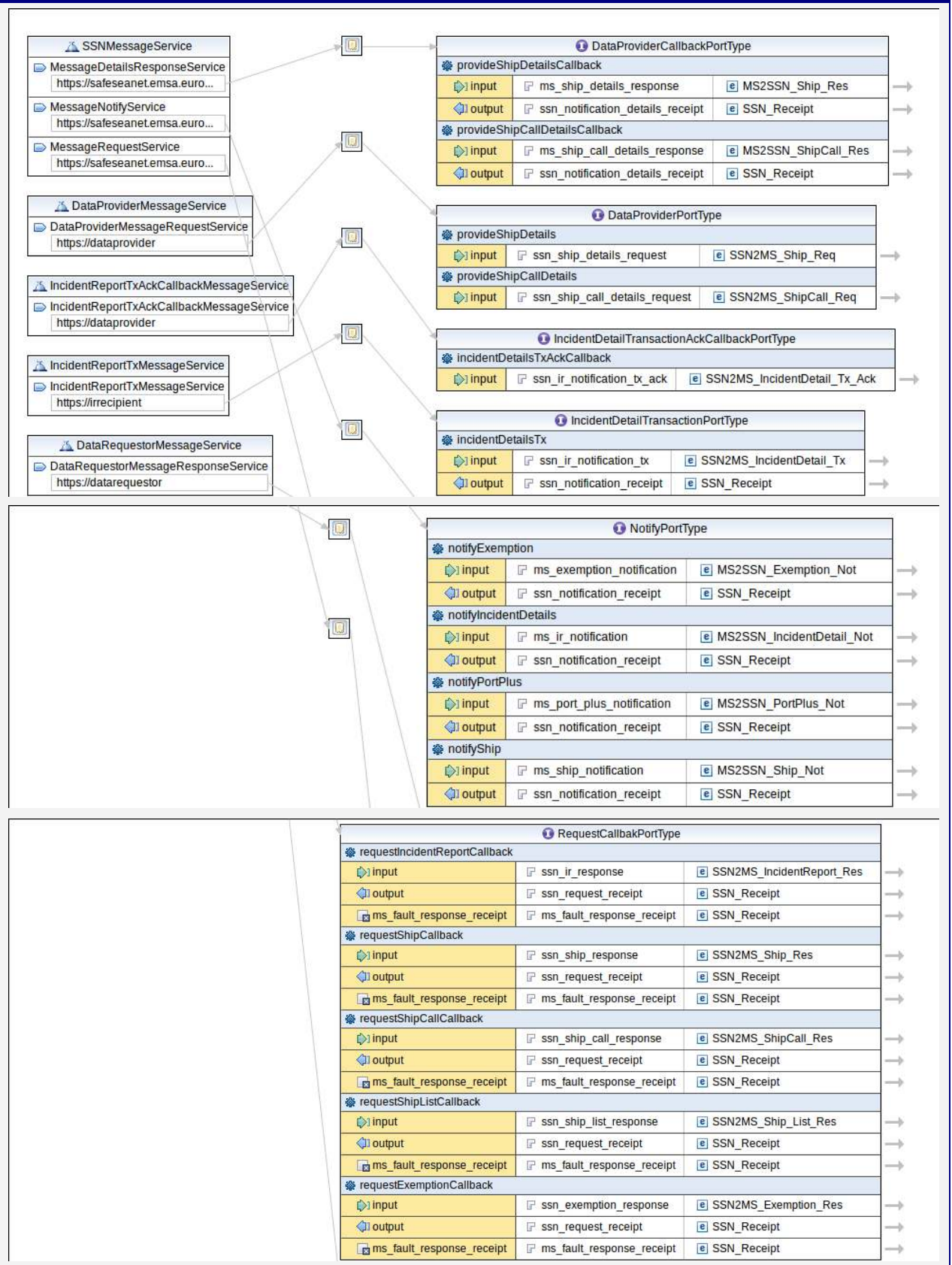


4.5.2.2 Package xmlprotocol-ws

4.5.2.2.1 WebService: MessageService



Service Contract: message.wSDL



Service Contract: message.wsdl			
WSDL	Message.wsdl Example of service contract WSDL file; it is based on ssn.xsd schema (data contract).		
Data contract	ssn.xsd		

Notification Messages

The Figure 4-7 shows the messages (SOAP over HTTP) exchanges between the Data Requestors and EIS Service Provider (Web Services) related to the Message Domain as well as the relationship of these messages with the domain classes. More specifically,

- The **MessageNotification (MS2SSN_<SSN_type>_Not)** message is sent by a Member State or system linked to SafeSeaNet (acting as IR Data Provider) in order to **notify** SafeSeaNet that a Member State owns some kind of information about an incident. Message Information is represented by the classes consist the notification package of the domain model.

The IR Data Provider should provide the **acknowledgeCallback** operation where SSN shall send the **ssn_ir_notification_tx_ack (SSN2MS_IncidentDetail_Tx_Ack)** asynchronously; this message is generated by SSN system and delivered by SsnOutgoingMessageProcessor of the xmlprotocol-core module acting as webservice client.

The role of the IR Recipient is also related with the new IR notifications;

the IR Recipient should provide the **notifyIR** operation where SSN shall send the **ssn_ir_notification_tx (SSN2MS_IncidentDetail_Tx)** asynchronously; this message is generated by SSN system and delivered by SsnOutgoingMessageProcessor of the xmlprotocol-core module acting as webservice client. This operation requires a synchronous response **ms_ir_notification_receipt (SSN_Receipt)**.

- The operations request/response (messages of type **MS2SSN_<SSN_type>_Req, MS2SSN_<SSN_type>_Res**) are not affected.

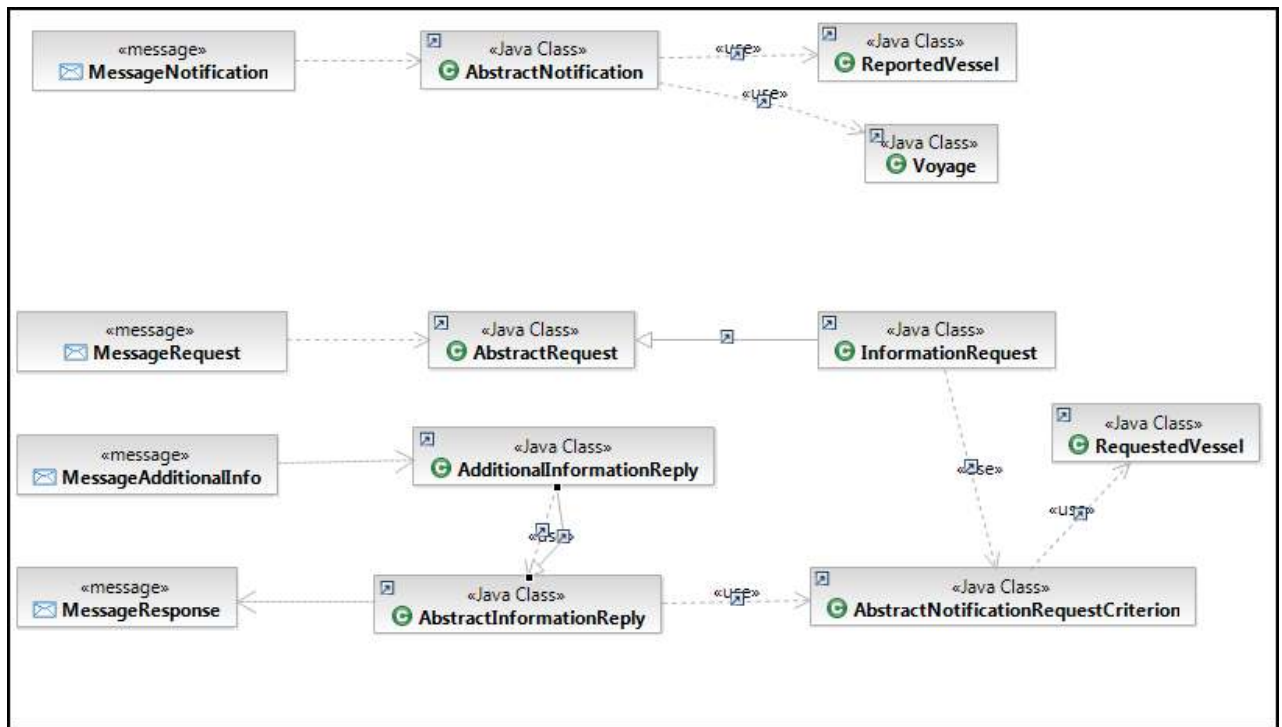
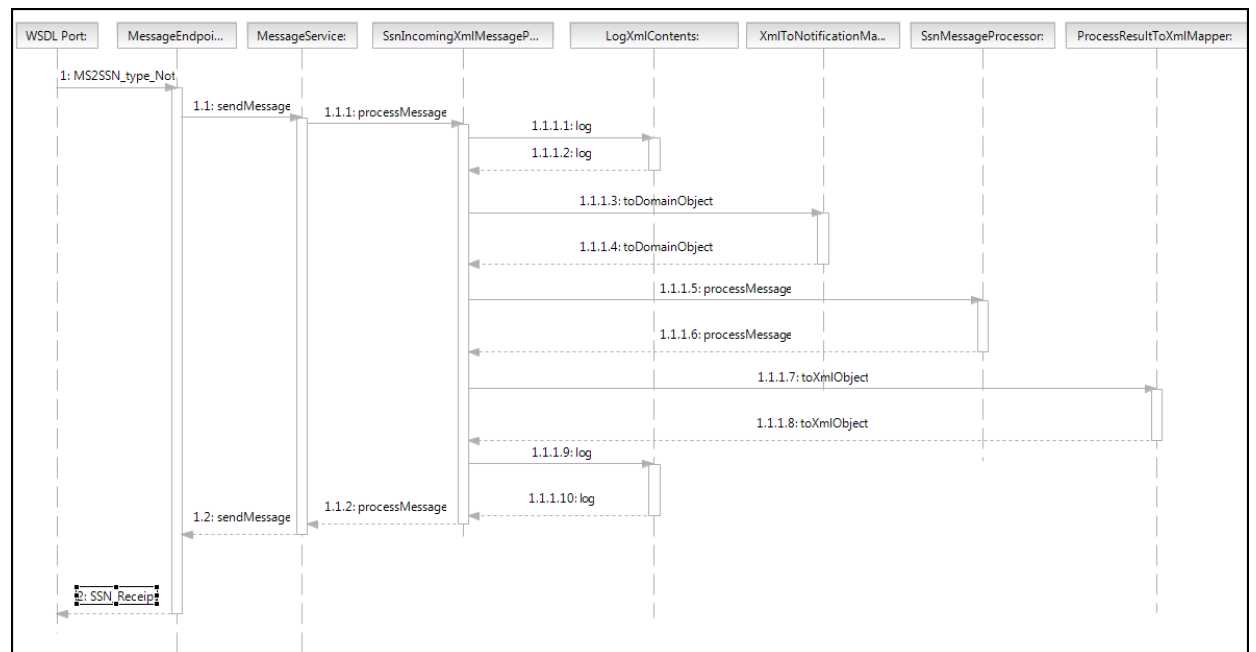


Figure 4-7 Notification / Request / Response Messages

Sequence Diagram : EIS Incoming XML Message Processor



4.5.3 Interfaces between the *ssn-xmlprotocol-app* and the *ssn-core-app*

In general, the ***ssn-core-app*** application, accepts calls for processing messages - both synchronously and asynchronously - via the remote stateless session EJB ***SsnMessageProcessorBean***, which is located at package ***ssn.message.processor.ejb*** and actually is a proxy of the class ***SsnMessageProcessorImpl*** - located under the package ***ssn.message.processor***.

The ***SsnMessageProcessorImpl*** implements a single method:

processMessage(Message message) : ProcessResult

Based on the type of the message the ***SsnMessageProcessorBean*** bean:

- Either calls synchronously the ***NotificationManager*** in case the message is a notification (MS2SSN_<SSN_type>_Not)
- Or sends the message to the ***ssn-core-app***'s incoming messages queue in order the message to be processed asynchronously, if the message is one of the following:
 - InformationRequest (MS2SSN_xxx_Req),
 - AdditionalInformationReply (MS2SSN_xxx_Res)

The ***ssn-xmlprotocol-app*** uses the ***SsnMessageProcessor*** EJB to route incoming messages to the ***ssn-core-app*** application.

The ***ssn-core-app*** applications sends asynchronous messages to its clients (*ssn-xml-protocol-app*, and management console in our case) using its outgoing queue. Each outgoing message has an attribute (attribute protocol of class *Message*), which denotes the target client application.

The ***ssn-xmlprotocol-app*** has a permanent listener on the outgoing queue - implemented as the message driven bean ***OutgoingMessageProcessorMdb***, located under package ***ssn.xmlprotocol.message.processor.ejb***. The listener filters the outgoing queue and consumes only messages that target the ***ssn-xmlprotocol-app***.

The types of the messages that are accepted are:

- AdditionalInformationRequest (SSN2MS_<type>_Req), or
- InformationReply (SSN2MS_<type>_Res).

Each message is converted into xml/soap and posted to the appropriate MS via http protocol.

It should be noted that the system tries to re-deliver a message in case that the http protocol (http client post method) fails - return code >= 400. The number of retries is defined by the SSN_MAX_RETRY_COUNT application parameter.

4.5.4 Security on the *ssn-xmlprotocol-app*

The *ssn-xmlprotocol-app* application does not have its own security layer. It relies on:

- The system level security provided by the runtime environment (SSL and Client Certificate)
- The application level security offered by *ssn-core-app*.

4.6 SSN-Central Database (CD)

This section provides an overview of the required changes on SSN-EIS System for the management of the SSN entities on the Users, Authorities, Countries, Locations, Ship and Central Ship Database and the corresponding consolidation mechanisms.

The SSN system is identified by the following entities owned by the "SSN Resources" Business System

- Users; they are classified to organisations ("System" users) and persons ("Human" users)
- Authorities; COD organisations
- Locations
- Countries
- Vessels

Thus, SSN system provides a number of services that enable the management (*CRUD*) of these entities as shown in Figure 4-8.

An important upgrade of the aforementioned entities is the identification of the registry where the instance of Vessel entity is / to be stored. The type of *registry* attribute is an enumeration with values **Operational**, **Central**.

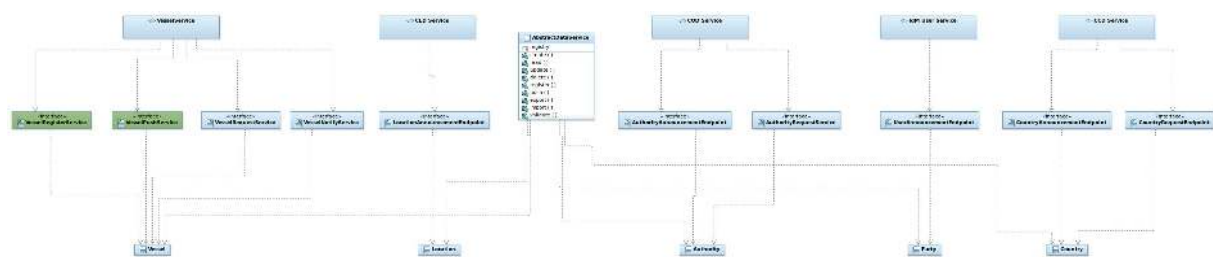


Figure 4-8 SSN CD Services provided operations.

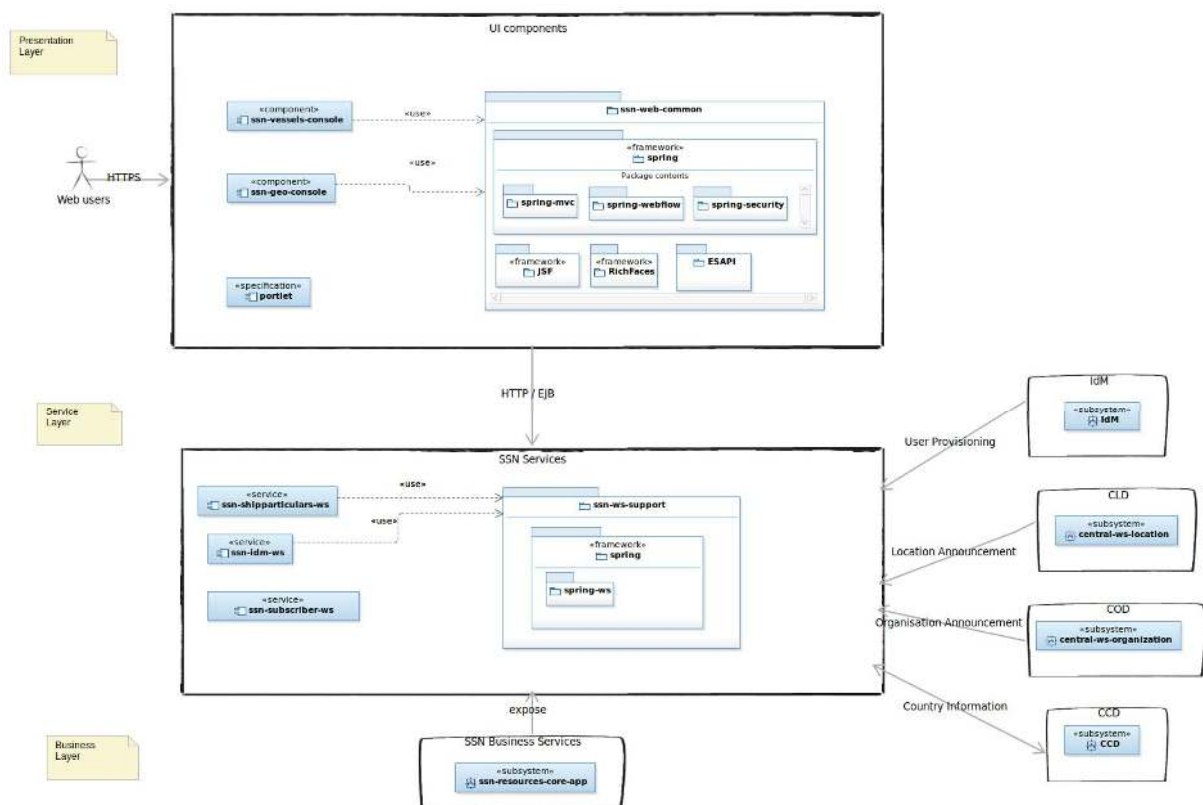


Figure 4-9 SSN CD component diagram

SSN Resources system consists of the following subsystems as depicted in Figure 4-9:

1. **EIS Resources management consoles UI** components provide web interface for the management of the vessel (Operational and Central) and temporary location (Operational) entities (Presentation Layer). More specifically,
 - a. **ssn-web-common** component is a library that provides
 - the spring framework functionality as spring web MVC, spring web flow and spring security; Regarding spring security authorization,
 - a custom SSN authentication processing filter intercepts any request and attempts to perform authentication, if no authentication object is placed in the spring security context, by retrieving the SM_USER HTTP header token from the request. Based on the token value it then loads the IdM user with its IdM roles from SSN EIS DB, creates the spring-security user with the the authorities granted (IdM roles) and stores user information which is encapsulated into the Authentication object. Further authorization will be handled by the application solely considering the granted authorities (IdM roles);
 - the JSF framework functionality;
 - the [RichFaces](#) JSF implementation;
 - ESAPI functionality.

It is used by all the following UI components;

- b. **ssn-vessels-console UI** component that provides web interface for Vessel management. The packaging of this component exposes two deployment artifacts,
 - one for the management of the Operational Ship Database; it is actually the isolation of the Vessel management provided by the current version of ssn-admin-console UI component (backward compatibility), and
 - a second for the management of the Central Ship Database (CSD);
- c. **ssn-geo-console UI** component that provides web interface for temporary Operational Locations management. The packaging of this component exposes one deployment artifact,
 - for the management of the Operation Registry;
 -

The authentication of the aforementioned UI components shall be done via the SSN SSO (IdM).

The communication mechanism between the UI components and the SSN EIS Business services shall not be changed; it shall be based on EJB.

2. **SSN EIS Resources Services** subsystem (Service Layer) exposes the EIS Business functionality as
 - a. Web Services (SOAP over HTTP) and
 - b. EJB 3.0 (stateless session beans).

More specifically,

- a. **ssn-ws-support** component is a library that provides the spring-ws functionality and it is used by ssn-ship particulars-ws and ssn-idm-ws components expose their functionality as Web Services;
- b. **ssn-ship particulars-ws** exposes the EIS Vessel functionality. The Web Services communication is based on the EIS Vessel Services (vessel.service.wsdl) and the SSN Ship Particulars XML schema (SSN_Ship_Particulars_Exchange.xsd).
- c. **ssn-idm-ws** exposes the EIS Users functionality in respect to receiving announcements of updates in IdM. The Web Services communication is based on the IdM User Service (idm.wsdl) and its respective XML schema (idm.xsd).

- d. **ssn-subscriber-ws** exposes the EIS functionality in respect to receiving announcements of updates in CCD, CLD and COD. The Web Services communication is based on the Country (countryBaseRegistryAnnouncementService.wsdl, countryBaseRegistryInformationService.wsdl), Location (locationService.wsdl) and Authority (authorityService.wsdl) Services and their respective XML schemata (CBR_SubscriberAnnouncement_1.0.xsd, CBR_Receipt_1.0.xsd, countryBaseRegistryInformationService.xsd, SSN_Location_Exchange.xsd, SSN_Organisation_Exchange.xsd), as defined in CD system. Component uses JAX-WS technology for building the endpoints that communicate with CD system using SOAP to receive announcement messages.
3. **ssn-resources-core-app** subsystem implements the SSN EIS Resources Business functionality (Business Layer). This subsystem exposes the aforementioned **SSN EIS Resources Services**. The existing version shall be upgraded to provide the required CD functionality.

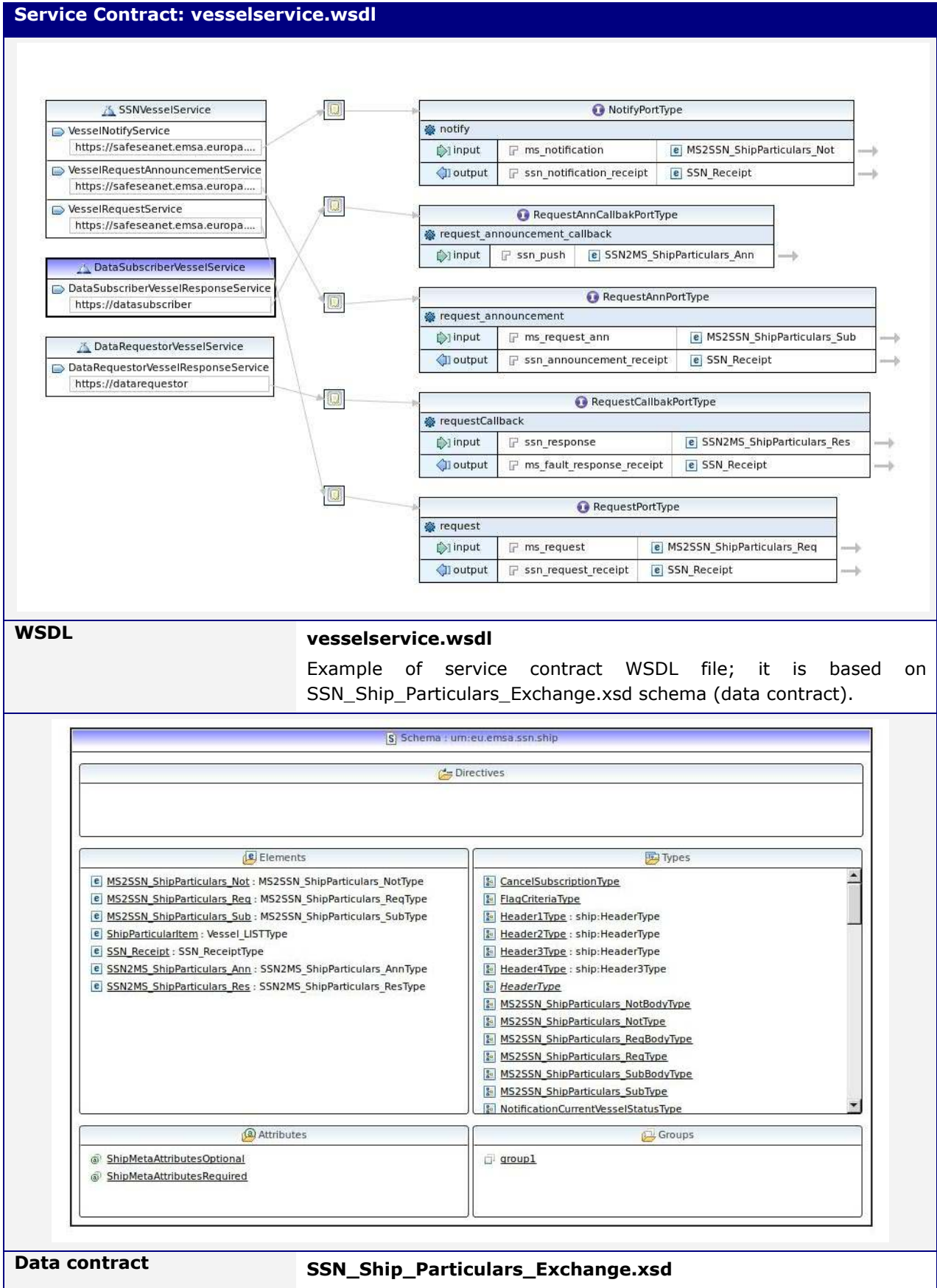
Data Services; ssn-resources-dao module, included in the ssn-resources-core-app subsystem, implements the EIS Resources Data Services; it shall be upgraded to take into account the aforementioned *registry* attribute of the SSN EIS entities so to update accordingly the corresponding registry (Operational / Central). The search criteria classes shall be also updated with *registry* attribute for the information retrieval from the corresponding registry.
4. **IdM** subsystem represents the EMSA Identity Management (IdM) system provides the user provisioning service; SSN SSO functionality. This node is out of the scope of the document.

The SSN Central Database provides an electronic interface for EIS entities management by Member States and other EMSA systems (External Systems). This section describes this communication provided by the SSN CD Web Services.

4.6.1 Ship Particulars Service

The SSN Ship Particulars Web Service is upgraded to

- "Push" CSD updates to MS subscribers;
- Provide details on the history of changes in the records of the CSD upon request.



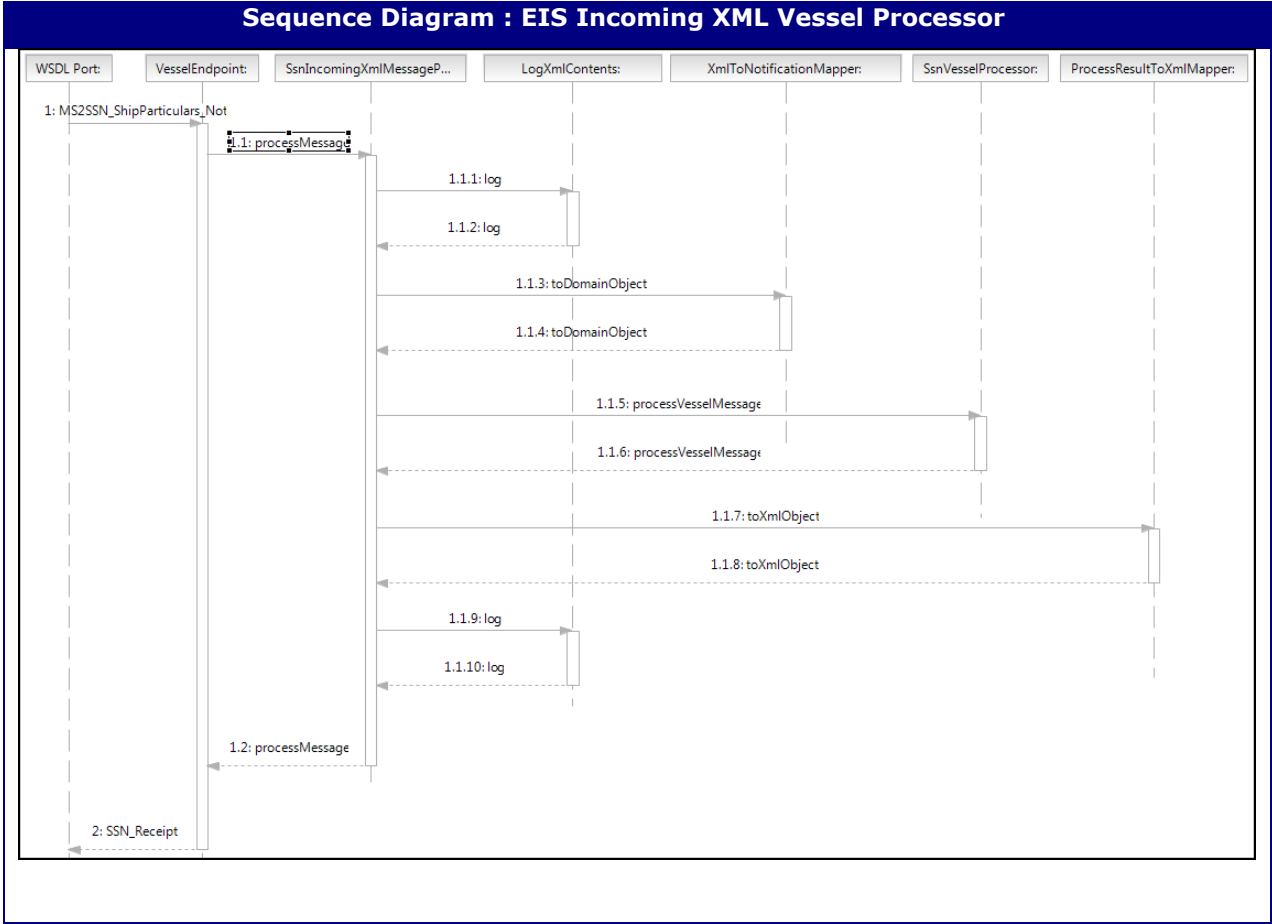
Ship Particulars Messages

The following messages (SOAP over HTTP) implements the aforementioned new functionality. More specifically,

- The **RequestAnnouncement (MS2SSN_ShipParticulars_Sub)** message is sent by a Member State system or any other system linked to SSN (data **subscriber**) in order to **subscribe/unsubscribe (ms_request_ann) for the ship particulars announcement**. Such service is implemented by exchanging different messages between the data **subscriber** and the SafeSeaNet system. The data **subscriber** should provide the **request_ann_callback** operation where SSN shall send the **AnnouncementShipParticulars (SSN2MS_ShipParticulars_Ann)** message asynchronously. The **AnnouncementShipParticulars** message is generated by SSN system and delivered to data subscribers; SafeSeaNet system acts as webservice client on this operation (**request_shipparticulars_callback**).
- The **AnnouncementShipParticulars (SSN2MS_ShipParticulars_Ann)** message is sent by the SafeSeaNet system to the Member State system or any other system linked to SSN (data **subscriber**) in order to **deliver (ssn_push) the ship particulars information to the registered data subscribers**. This service (**DataSubscriberVesselService**) is implemented by the data **subscriber**.

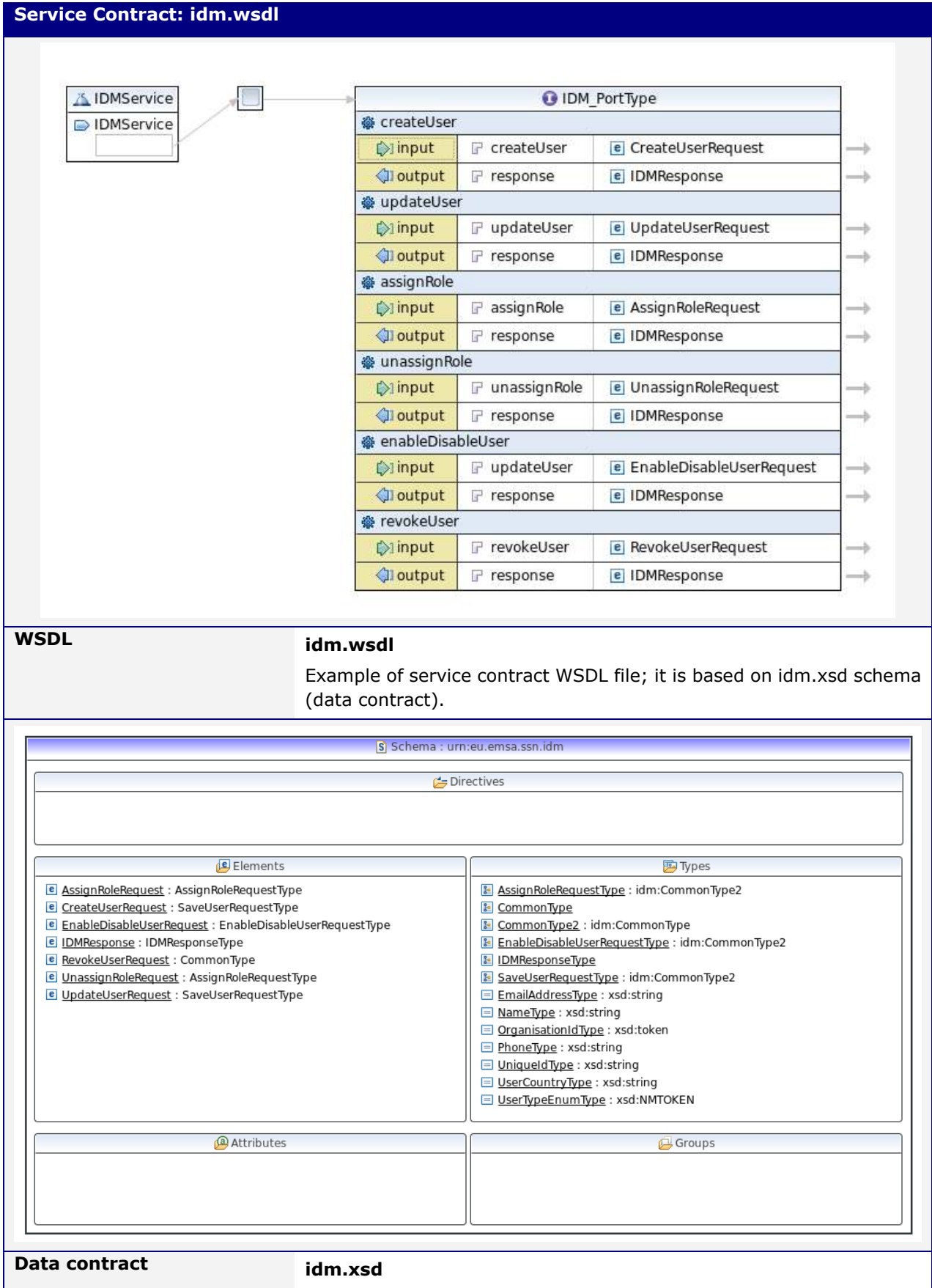
The following messages (SOAP over HTTP) implements the existing functionality;

- The **ShipParticularsNotification** message is sent by a Member State or any other system linked to SSN in order to **notify** SafeSeaNet new insertions or updates in the ship registry maintained by the data provider. It actually includes the vessel information to be recorded on the EIS database.
- The **ShipParticularsRequest** message is sent by a Member State system or any other system linked to SSN (data requester) in order to **request** the ship particulars of a specific ship registered into the EIS database. It actually consists of the vessel search criteria. The data requester should provide the **requestCallback** operation where SSN shall send the **ssn_response (ShipParticularsResponse)** asynchronously.
- The **ShipParticularsResponse** message is the response sent by SafeSeaNet to a Member State system or EMSA system (e.g. THETIS) any other future system to be connected to SSN requesting ship particulars (**ShipParticularsRequest**). It consists of a list of vessels satisfies the applied search criteria submitted by the corresponding request.



4.6.2 IdM User Service

The IdM User Web Service defines the IdM users' data exchange.



WSDL

idm.wSDL

Example of service contract WSDL file; it is based on idm.xsd schema (data contract).

Schema : urn:eu.emsa.ssn.idm

Directives

Elements

AssignRoleRequest : AssignRoleRequestType

CreateUserRequest : SaveUserRequestType

EnableDisableUserRequest : EnableDisableUserRequestType

IDMResponse : IDMResponseType

RevokeUserRequest : CommonType

UnassignRoleRequest : AssignRoleRequestType

UpdateUserRequest : SaveUserRequestType

Types

AssignRoleRequestType : idm:CommonType2

CommonType

CommonType2 : idm:CommonType

EnableDisableUserRequestType : idm:CommonType2

IDMResponseType

SaveUserRequestType : idm:CommonType2

EmailAddressType : xsd:string

NameType : xsd:string

OrganisationIdType : xsd:token

PhoneType : xsd:string

UniqueIdType : xsd:string

UserCountryType : xsd:string

UserTypeEnumType : xsd:NMTOKEN

Attributes

Groups

Data contract

idm.xsd

IdM Messages

The following messages (SOAP over HTTP) implement the IdM users' data exchange functionality. More specifically,

- The **CreateUserRequest** message is sent by OIM/IdM Provisioning Service to SSN to notify SafeSeaNet of the creation of a user. It includes the newly created user information to be recorded on the EIS database.
- The **UpdateUserRequest** message is sent by OIM/IdM Provisioning Service to SSN to notify SafeSeaNet of the update of a user. It includes the updated user information to be recorded on the EIS database.
- The **RevokeUserRequest** message is sent by OIM/IdM Provisioning Service to SSN to notify SafeSeaNet of the revocation of a user. It includes the "userId" information required to resolve and delete user from the EIS database.
- The **EnableDisableUserRequest** message is sent by OIM/IdM Provisioning Service to SSN to notify SafeSeaNet of the the user's status (enabled/disabled). It includes the "userId" along with the "enableUser" (boolean) status information required to update resolved user's status in the EIS database.
- The **AssignRoleUserRequest** message is sent by OIM/IdM Provisioning Service to SSN to notify SafeSeaNet of a user's role assignment. It includes the "userId" along with the new role information required to add user's role in the EIS database.
- The **UnassignRoleUserRequest** message is sent by OIM/IdM Provisioning Service to SSN to notify SafeSeaNet of a user's role revocation. It includes the "userId" along with the revoked role information required to delete user's role from the EIS database.
-

4.6.3 ssn-resources-core-app / enterprise archive

This module extracts the Central Resources Core functionality in EJB.

4.7 SSN-IMDaTe

This section provides an overview of the SSN-IMDaTe integration diagram. It describes

- the SSN-SI system; to be upgraded to support S-AIS messages;
- the SSN system; to be upgraded to exchange information in CDF format with
- the IMDaTe system.

- b. Servicesfor data transfer to JMS server
- 8. SSN JMS Server provides
 - a. JMS queuing services.
- 9. **IMDatE**provides
 - a. Servicesfor data transfer from/to JMS server;
- 10. LRIT and VMS Data provider systems (out of the scope of this document).

The first application (ssn-si-core-app) materialises the main functionality of SSN-SI at national proxy level, which is the management of AIS position reports (in VDM format) (plus connection configuration settings, etc).

The second application (stires-core-app) materialises the main functionality of SSN STIRES, which is the management of AIS messages (plus connection configuration/authentication settings, monitoring, etc).

Both applications manage the information of AIS messages in a way that is independent from the channel of communication through which messages are exchanged.

The third application (ssn-core) materialises the main functionality of SSN EIS, which is (for the context of current document) the management of Ship Call Request / Response messages.

The forth application (Proxy Server) provides transformation services for position messages from raw to SPCDF format so to exchange the position information with SSN in canonical format.

The fifth application (JMS Server) provides JMS queuing services.

The RESTful Services (HTTPS) as exchange endpoints are recommended due to the generic/independent protocol they use; furthermore, the payloads of RESTful services shall be stored on JMS Queues for further processing by IMDatE. OSB supports RESTful Services.

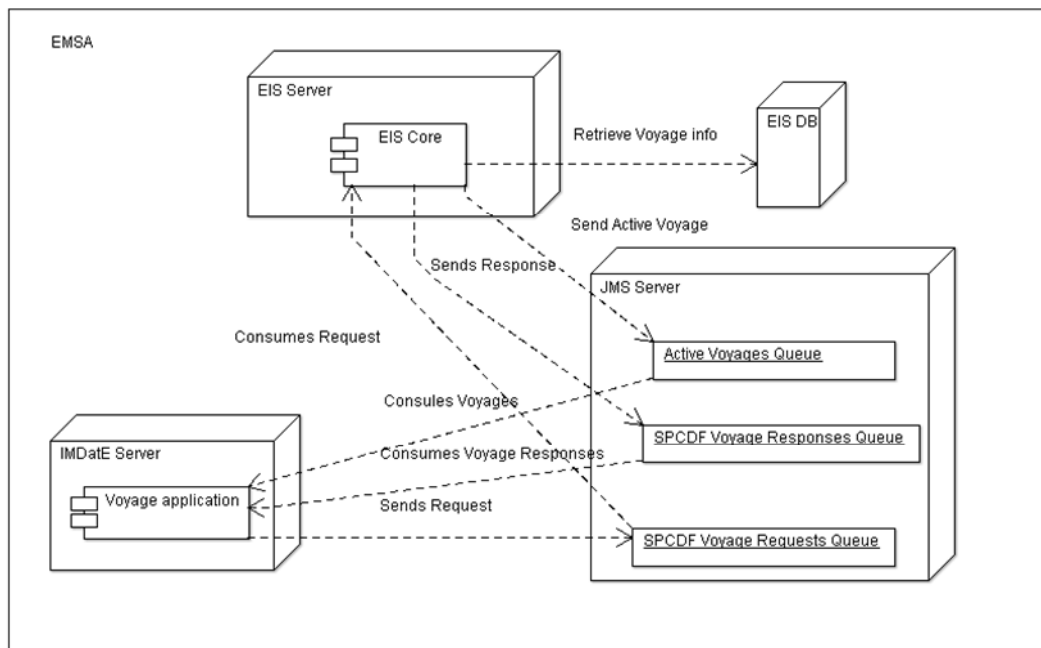


Figure 4-11Connection diagram of SSN IMDatE components exchange Voyage messages.

Figure 4-11 depicts the following subsystems exchange Voyage information:

- 1. **SSN EIS** provides:
 - a. JMS servicesfor data transfer from/to the JMS Server;

- b. Transformation services (marshal/unmarshal) of exchanged information;
 - c. PL/SQL stored procedures for the SSN Database interaction (voyages retrieval and calculation).
2. SSN JMS Server provides
 - a. JMS queuing services.
3. **IMDatE** provides
 - a. Services for data transfer from/to JMS Server.

The SSN-IMDatE system integration requires the following information (entities) to be owned by the "SSN" Business System

- AIS Data/Position Report;
- Vessel Identification and
- Enrichment;
- Voyage Information
- Connection information/configuration (Data Providers/Subscribers).

Thus, SSN system provides services that enable the access to, and update of, these entities as shown in Figure 4-12.

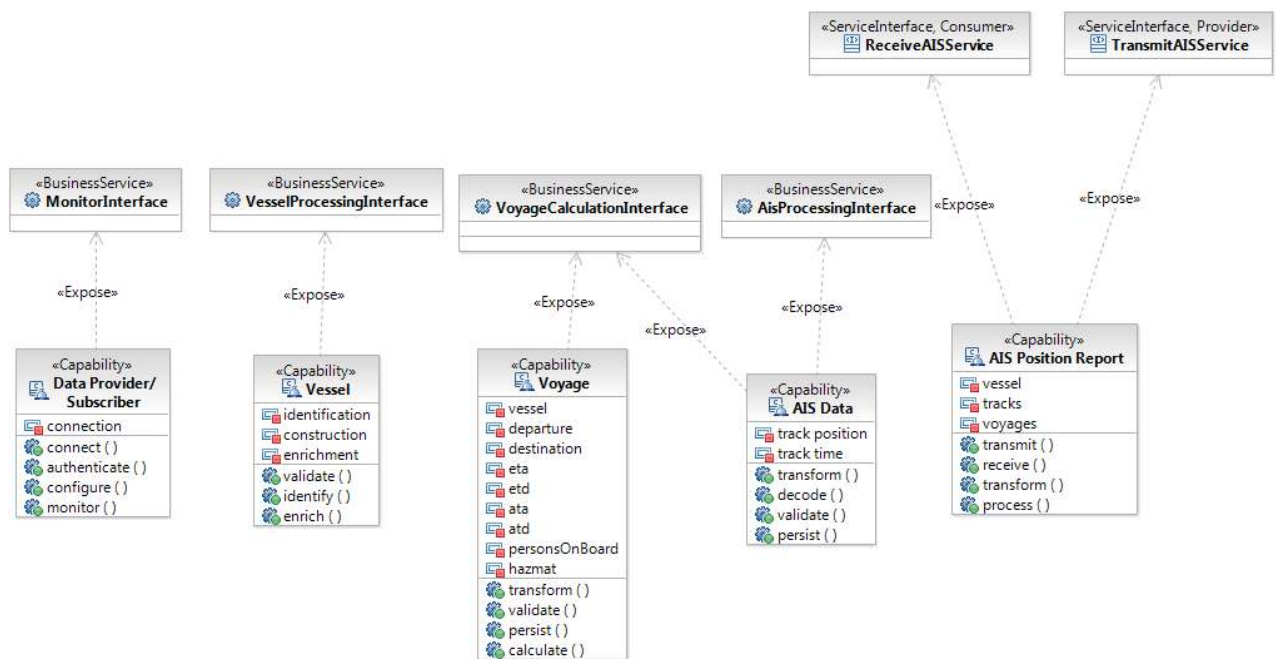


Figure 4-12 SSN Service provided operations.

4.7.1 Domain module/package

This module/package includes the definition of AIS entities (position reports, vessel, proxy, etc) and it is used by all the AIS sub-systems (sources).

4.7.1.1 UML Class Diagrams

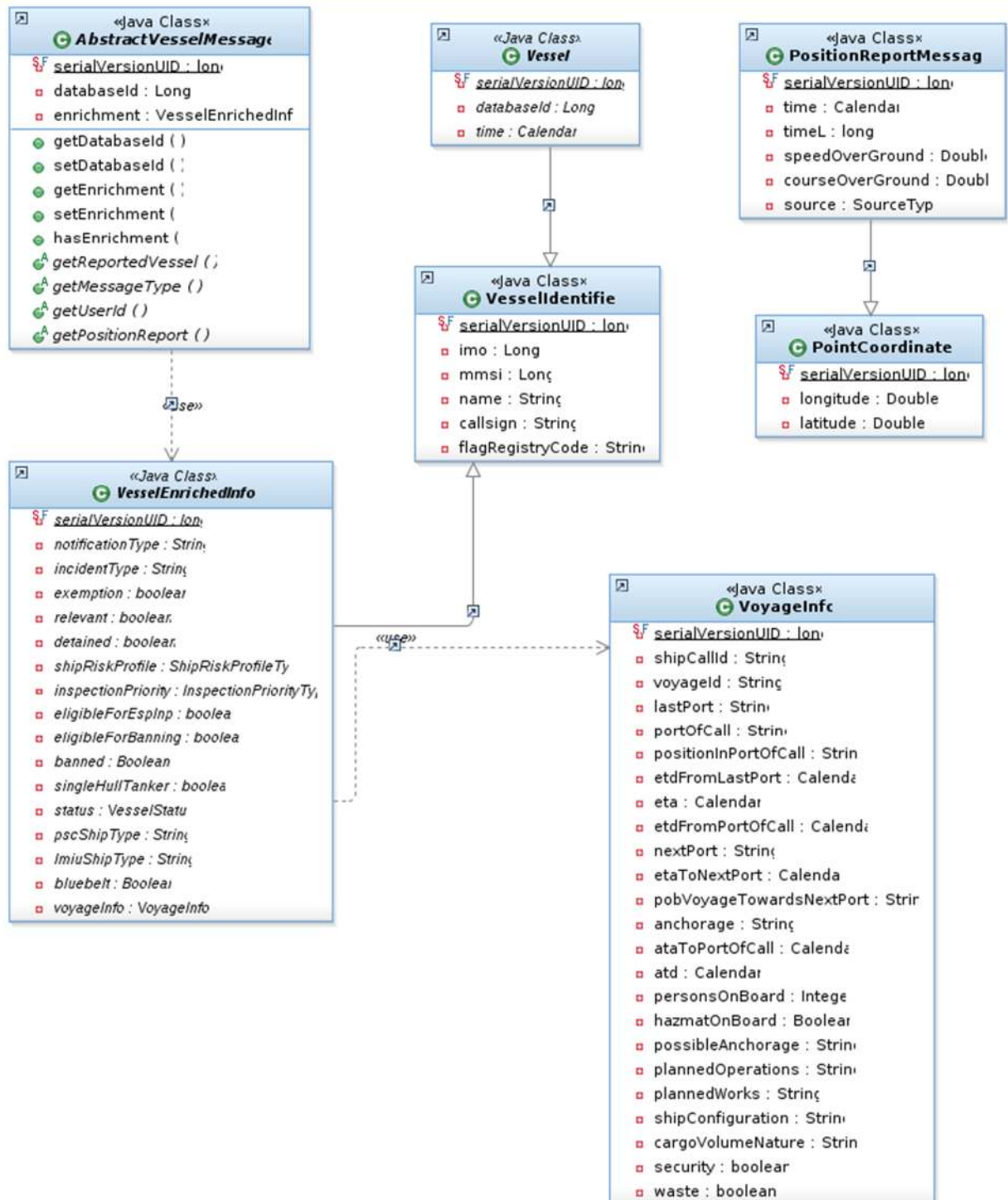
This section covers the architectural significant elements of the design model. It presents the definition of the most significant classes that will implement the requested functionality, organised into packages.

The classes are organised in packages according to the functionality they provide. A package is a general-purpose model element that organizes model elements into groups. Each package contains a set of classes and interfaces, representing what will become components in the implementation.

4.7.1.2 Module: ssn-spm-domain

4.7.1.2.1 Package: ssn.spm.domain

Class Diagram: Common position message



Class

VesselIdentifier

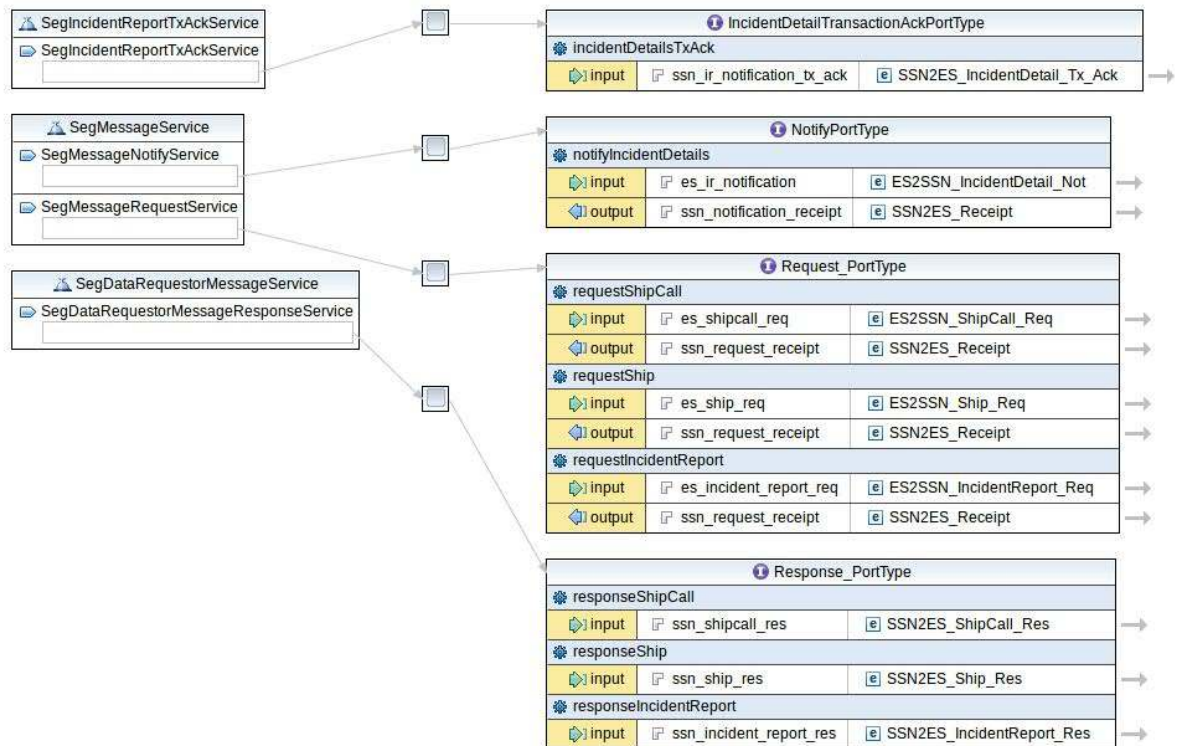
This class represents the vessel identification details such as the IMO,

Class Diagram: Common position message	
	MMSI, Callsign, ship name and the flag registry code.
Class	<p><u>AbstractVesselMessage</u></p> <p>The abstract message related to a SSN particular vessel. It includes the following attributes:</p> <ul style="list-style-type: none"> ➤ enrichment information and abstract <i>getters</i> for; ➤ reported vessel: vessel's identification ➤ userId: the message originator; ➤ message type; ➤ position report
Class	<p><u>VesselEnrichedInfo</u></p> <p>This class extends VesselIdentification class to include the SSN enrichment properties such as banned, relevant, single hull tanker, detained and the voyage information provided by the class below.</p>
Class	<p><u>VoyageInfo</u></p> <p>This class encapsulates the properties of a vessel voyage (enrichment); it includes departure and destination information, persons on board, hazmat on board, etc. The class content shall be upgraded to also provide the waste, security and exemption information.</p>
Class	<p><u>PointCoordinates</u></p> <p>This class encapsulates the properties of a point such as latitude, longitude, altitude.</p>
Class	<p><u>PositionReportMessage</u></p> <p>This class extends PointCoordinates class to encapsulate the various dynamic properties of Position report messages at a given point in time and space; it includes SOG, COG, ROT, data source (S-AIS, T-AIS).</p>
Class	<p><u>Vessel</u></p> <p>This class extends VesselIdentification class to represent a vessel that exists on STIRES_SCHEMA database.</p>

4.7.1.3 SEG Service

The SEG Web Service (seg_4_0.xsd) defines the data exchanged between EIS and SEG.

Service Contract : segservice.wSDL



WSDL

segservice.wSDL

Example of service contract WSDL file; it is based on seg_4_0.xsd schema (data contract) – it uses ssn_4_0.xsd.

Schema : urn:eu.emsa.ssn.voyage

Directives

- ssn_4_0.xsd {urn:eu.emsa.ssn}

Elements	Types
ES2SSN_IncidentDetail_Not : ES2SSN_IncidentDetail_NotType	ActiveExemptionsType
ES2SSN_IncidentReport_Req : ES2SSN_IncidentReport_ReqType	ActiveIncidentsType
ES2SSN_Ship_Req : ES2SSN_Ship_ReqType	ActiveMRSType
ES2SSN_ShipCall_Req : ES2SSN_ShipCall_ReqType	ActiveVoyagesType
SSN2ES_IncidentDetail_Tx_Ack : SSN2ES_IncidentDetail_Tx_AckType	AdditionalVoyageCriteriaType : ssn:ShipCallIdentificationCriteriaType
SSN2ES_IncidentReport_Res : SSN2ES_IncidentReport_ResType	CommonHeader2Type : voyage:CommonHeaderType
SSN2ES_Receipt : SSN2ES_ReceiptType	CommonHeader3Type : voyage:CommonHeaderType
SSN2ES_Ship_Res : SSN2ES_Ship_ResType	CommonHeaderType : ssn:HeaderType
SSN2ES_ShipCall_Res : SSN2ES_ShipCall_ResType	EnrichmentResultsType
SSN2SEG_Enrichment_Res : SSN2SEG_Enrichment_ResType	ES2SSN_IncidentDetail_NotBodyType
SSN2SEG_Exemption_Res : SSN2SEG_Exemption_ResType	ES2SSN_IncidentDetail_NotType
	ES2SSN_IncidentReport_ReqBodyType
	ES2SSN_IncidentReport_ReqType
	ES2SSN_RequiredResponseCriteriaType

Attributes	Groups
	EnrichmentResultsGroup
	SEGPortPlusNotificationType

Data contract **seg_4_0.xsd**

SEG Messages

The following messages (SOAP over HTTP) implements the SEG data exchange. More specifically,

- The **ES2SSN_ShipCall_Req** message is sent by SEG to EIS (data requestor) over HTTP in order to **request (es_shipcall_req)** the operational information (ship specific/ ship call specific/ port of call specific) and it always return SSN2ES_Receipt as response synchronously – output: ssn_request_receipt SSN_Receipt.

It should be noted that: This operation is implemented according to the WS-Addressing Request-Response Message Exchange Pattern (MEP). So, the mandatory “reply endpoint” – “ReplyTo” – property is used by SSN EIS for the asynchronous response to the data requestor.

The “reply endpoint” property actually provides the operation **responseShipCall** described below and implemented by SegDataRequestorMessageService (SEG system).

Additionally, the mandatory “message id” – “MessageID” – property should have the same value with the value of ESRefId attribute of the XML Header element.

- The **SSN2ES_ShipCall_Res** message is sent by the EIS system to the SEG system over HTTP as asynchronous response (**ssn_shipcall_res**) including the voyage information related to the criteria set of the aforementioned **ES2SSN_ShipCall_Req** message. This operation is the callback interface of the request (**es_shipcall_req**) operation. It is implemented according to the WS-Addressing Request-Response MEP. The mandatory “relationship” – “RelatesTo” – property includes the value of correlation Id (ESRefId) of the SSN2ES_ShipCall_Res.

- The **ES2SSN_Ship_Req** message is sent by SEG to EIS (data requestor) over HTTP in order to **request (es_ship_req)** the MRS operational information (ship specific/ MRS identification specific/ country specific) and it always return SSN2ES_Receipt as response synchronously – output: ssn_request_receipt SSN_Receipt.

It should be noted that: This operation is implemented according to the WS-Addressing Request-Response Message Exchange Pattern (MEP). So, the mandatory “reply endpoint” – “ReplyTo” – property is used by SSN EIS for the asynchronous response to the data requestor.

The “reply endpoint” property actually provides the operation **responseShip** described below and implemented by SegDataRequestorMessageService (SEG system).

Additionally, the mandatory “message id” – “MessageID” – property should have the same value with the value of ESRefId attribute of the XML Header element.

- The **SSN2ES_Ship_Res** message is sent by the EIS system to the SEG system over HTTP as asynchronous response (**ssn_ship_res**) including the MRS information related to the criteria set of the aforementioned **ES2SSN_Ship_Req** message. This operation is the callback interface of the request (**es_ship_req**) operation. It is implemented according to the WS-Addressing Request-Response MEP. The mandatory “relationship” – “RelatesTo” – property includes the value of correlation Id (ESRefId) of the SSN2ES_Ship_Res.

- The **ES2SSN_IncidentReport_Req** message is sent by SEG to EIS (data requestor) over HTTP in order to **request (es_incident_report_req)** the operational information (ship specific/ incident type specific/ geographic criteria) and it always return SSN2ES_Receipt as response synchronously – output: ssn_request_receipt SSN_Receipt.

It should be noted that: This operation is implemented according to the WS-Addressing Request-Response Message Exchange Pattern (MEP). So, the mandatory “reply endpoint” – “ReplyTo” – property is used by SSN EIS for the asynchronous response to the data requestor.

The “reply endpoint” property actually provides the operation **responseIncidentReport** described below and implemented by SegDataRequestorMessageService (SEG system).

Additionally, the mandatory “message id” – “MessageID” – property should have the same value with the value of ESRefId attribute of the XML Header element.

- The **SSN2ES_IncidentReport_Res** message is sent by the EIS system to the SEG system over HTTP as asynchronous response (**ssn_incident_report_res**) including the Incident Report information related to the criteria set of the aforementioned **ES2SSN_IncidentReport_Req** message. This operation is the callback interface of the request (**es_incident_report_req**) operation. It is implemented according to the WS-Addressing Request-Response MEP. The mandatory "relationship" – "RelatesTo" – property includes the value of correlation Id (ESRefId) of the SSN2ES_IncidentReport_Res.
- The **ES2SSN_IncidentDetail_Not** message is sent by SEG to EIS (data provider) over HTTP in order to **notify (es_ir_notification)** the Incident Details information and it always return SSN2ES_Receipt as response synchronously – output: ssn_notification_receipt SSN_Receipt.

It should be noted that SSN EIS uses the SOAP interface URL of the notification's provider (PARTIES_INTERFACES.PROVIDER_URL of the sender) to submit the asynchronous Txacknowledgment.

The "Tx acknowledgment endpoint" property actually provides the operation **incidentDetailsTxAck** described below and implemented by SegIncidentReportTxAckService (SEG system).

- The **SSN2ES_IncidentDetails_Tx_Ack** message is sent by the EIS system to the SEG system over HTTP as asynchronous acknowledgment (**ssn_ir_notification_tx_ack**) including the transmission status of the aforementioned **ES2SSN_IncidentDetails_Not** message. This operation is the callback interface of the **notify (es_ir_notification)** operation.

4.7.1.4 Enrichment Service

The SSN Enrichment Web Service defines the data exchanged between EIS and EMSA Systems (SEG).

Service Contract : enrichment.wadl

```

version="1.0" encoding="UTF-8"
application xmlns="http://wadl.dev.java.net/2009/02", xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance", xml...
  grammars
    include href="/seg_4_0.xsd"
  resources base="http://localhost/ssn-seg-ws/enrichment"
    doc xml:lang="en", title="SEG WS Enrichment/"
    resource path="/voyages"
      doc xml:lang="en", title="Active Voyages"
      method name="GET", id="requestVoyages"
        response status="200"
          representation mediaType="application/xml", element="seg:SSN2SEG_Enrichment_Res"
    resource path="/incidents"
      doc xml:lang="en", title="Active Incidents"
      method name="GET", id="requestincidents"
        response status="200"
          representation mediaType="application/xml", element="seg:SSN2SEG_Enrichment_Res"
    resource path="/MRS"
      doc xml:lang="en", title="Active MRS"
      method name="GET", id="requestMRS"
        response status="200"
          representation mediaType="application/xml", element="seg:SSN2SEG_Enrichment_Res"
    resource path="/exemptions"
      doc xml:lang="en", title="Active Exemptions"
      method name="GET", id="requestExemptions"
        request
          param name="CSDID", type="xsd:string", style="query", required="false"
        response status="200"
          representation mediaType="application/xml", element="seg:SSN2SEG_Exemption_Res"

```

WADL

enrichment.wadl

Service Contract : enrichment.wadl

Example of service contract WADL file; it is based on seg_4_0.xsd schema (data contract) – it uses ssn_4_0.xsd.

Enrichment Message

The **SSN2SEG_Enrichment_Res** message provides the Enrichment data. More specifically,

The **SSN2SEG_Enrichment_Res** message is the synchronous response in XML format sent by SSN on SEG HTTP request (GET method) providing vessel enrichment data related to either the voyages, incidents or MRS notifications. The request element is a choice of:

- /voyages (returns all voyages currently active; request the active voyage information for all the ships with a CSDID.)
- /incidents (returns all incidents currently active; request enrichment information for the active incidents at the time of the request.)
- /MRS (returns all MRS notifications currently active; request enrichment information for the latest received MRS report for all the vessels listed in the CSD)

4.8 STIRES Core

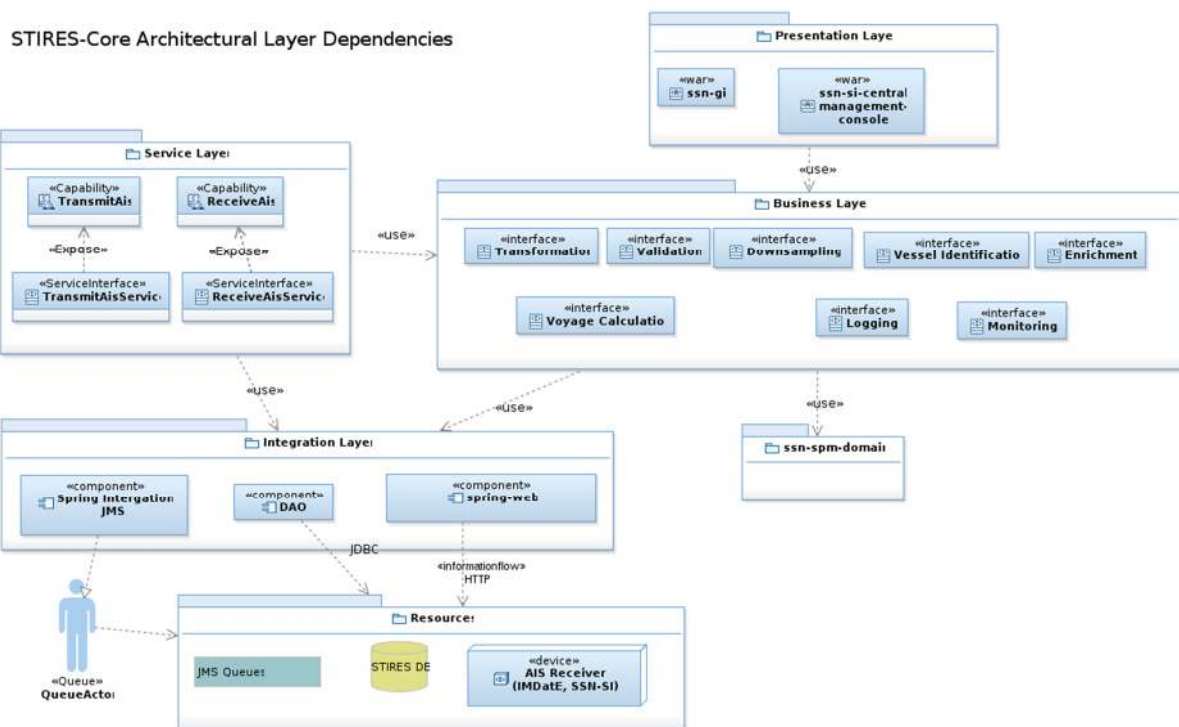


Figure 3–13 Architectural Layer dependencies (Logical view) of STIRES-Core

The SSN VDM core application exchanges the AIS messages with

- SSN-SI applications over HTTPS (RESTful Services) with JSON payload,
- JMS server over t3 with XML payload

The **incoming AIS messages** (from SSN-SI proxies and IMDatE stored to **Incoming AIS Queue**) are processed by a pool of “consumers” (AIS Message Processor/Data Processing Service). Using the Spring Integration module; each AIS message are transformed (to POJO), validated, identified,

enriched and stored to STIRES DB by the corresponding channels. The application performs separate processing (and queuing) for T-AIS and S-AIS.

It shall be noted that the list of the JMS queues is included in the readme.txt of the SSN SI release.

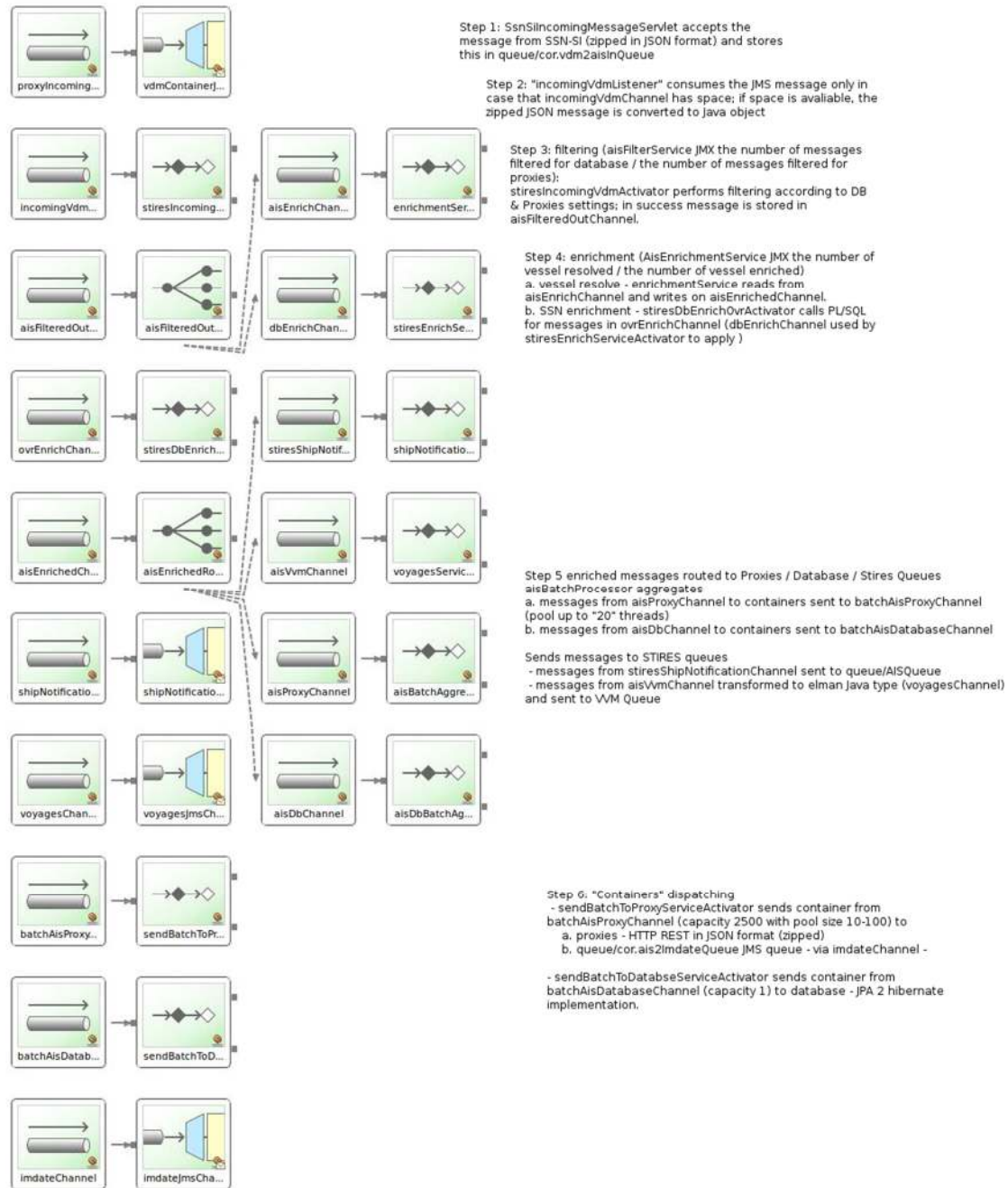


Figure 3—14 Transformation adapters for exchanged AIS messages

Figure 3—14 shows the usage of Spring Integration transformation adapters so that pre-processing functionality is independent of the format of incoming messages (JSON and XML). Corresponding transformation adapters are used for the enriched outgoing messages according to their destination (JSON for SSN-SI, XML for IMDate).

The STIRES-Core performs pre-processing using the information stored on the SSN DB and creates AIS messages for the enabled Proxies (and IMDatE that treated as a Proxy) that includes

- Position Reports;
- Enriched Vessel information;
- Voyage information.

The **outgoing AIS messages** (stored to **Outgoing AIS Queue**) are processed by a pool of "consumers". Using the Spring Integration module; each AIS message shall be packed (JSON/XML message) and sent to corresponding Proxies via HTTPS and to the JMS server via t3 when available (the standard redelivery JMS mechanism shall be configured to retry submit the AIS messages for temporarily unavailable proxies).

Monitoring process

The SSN VDM core performs collects and persists on a regular basis the monitoring information exchanged between the SSN-SIs.

The existing "monitoring channel" is used to "deliver" SSN-SI downsampling information to the corresponding SSN-SI so to be applied (downsampling per source, message type, etc).

1. Data Services

The DAO module provides CRUD operations to Business Layer; it uses JDBC for the connectivity with STIRES DB.

2. Business Layer

This layer provides the Data processing such as the message transformation, validation, identification, enrichment.

Business Support module/package

This module/package provides/implements the common business processing as

- Validation
- Down-sampling;
- Monitoring.

Validation

VDM application is upgraded to "resolve" the proxy on message arrival based on the proxy name provided in the corresponding HTTP header. In case of unknown/ disabled proxy the message shall be rejected.

Message filtering is performed by:

- data-age filters: filters messages older than X seconds.
When empty/null or zero no data-age filtering is performed.
For incoming messages, a default data age (3600 seconds) is applied for filtering in all sites (STIRES database, SSNSI providers and subscribers. In addition, for VDM proxies the data-age filter is defined and stored in the already existing PROXIES_SOURCES.PREF_DATA_AGE, which is defined through SSN SI central console "Update Proxy" page, "Preferences" tab and it applies to the AIS data that is distributed by SSN for a specific source (T-AIS or S-AIS).
- A selective down-sampling can be performed on the AIS messages (recorded in the STIRES database or sent to proxies/SSN SIs). In case the downsampling value is zero then no downsampling is performed. For example, suppose that an AIS message with ITU number 5 is received from the AIS target 247234222 at time T0 and that this message is sent to the VDM server. Later, at time T1, a new message with ITU number 5 is received from the same AIS target 247234222. The timestamp (T0) of the latest AIS messages of the same type for the same vessel (MMSI) is kept on a structure (Map<String, ExpiryItem<T>>) in memory with key `FIND_BY_MMSI0_PROXYID1_SOURCE2_MESSAGE_TYPE3_EXPRESSION` =

"{0}#{1}#{2}#{3}". Suppose that the parameter `downsampling_interval_itu_5` is set to 6(minutes).

The possible situations are:

$ABS^4(T1 - T0) \geq \text{downsampling_interval_itu_5}$ then the message will be sent.

$ABS(T1 - T0) < \text{downsampling_interval_itu_5}$ then the message will be considered eligible for discarding.

An application parameter is defined for S-AIS messages that stores the down-sampling values for each ITU message which should be stored in the DB (STIRES_SCHEMA.TSYS_SYSTEM_PARAMETERS). This parameter is applied on DB downsampling filter.

- A new table is created in the database (STIRES Oracle and SSNSI Derby) named `SSNSI_Cache` to store the "Write-through" cache data; its structure consists of two columns, `key/time` where `key` holds the data according to the expression `{0}#{1}#{2}#{3}` (`MMSI0_PROXYID1_SOURCE2_MESSAGE3`) and `time` holds the message contact date. For performance reasons the structure will be kept in memory and will be refreshed whenever needed.
- A new filter parameter is added on proxies preferences configuration so to provide the possibility of sending out data of either (both options should be possible):
 - specific combinations of originator codes or
 - AIS dynamic/position messages inside a geographical polygon shaped area;in addition to data of ships belonging to a certain ship flag (regardless of the other restrictions).
- enrichment filter; the SSN VDM core applies the configured Enrichment filter attribute to the outgoing message transmitted to the corresponding proxies.

NOTE: We shall point out that the new algorithm allows N number of messages to be stored with the same or close to the same *contactDt*. More than one newly received AIS messages of the same type for the same vessel (MMSI) may satisfy the criterion: the timespan between the newly received message and last received message is negative. The *contactDt* of the newly received messages may be equal to an existing message or very close to the *contactDt* of an existing message. Hence, the distribution of messages in time does not respect the downsampling value (R10).

3. Service Layer

This layer exposes the STIRES-Core functionality as RESTful Services.

Receive AIS Report capability; it accepts the AIS Report messages over HTTPS. The AIS Report messages include the Position Reports received from Data Providers (SSN-SI and JMS server).

Transmit AIS Report capability; it sends the Enriched AIS messages to Data Subscribers (SSN-SI and JMS server).

4.8.1 Domain module/package

This module/package includes the definition of AIS entities (position reports, vessel, proxy, etc) and it is used by all the AIS sub-systems (stires-vdm-core, ssn-si, etc).

4.8.1.1 UML Class Diagrams

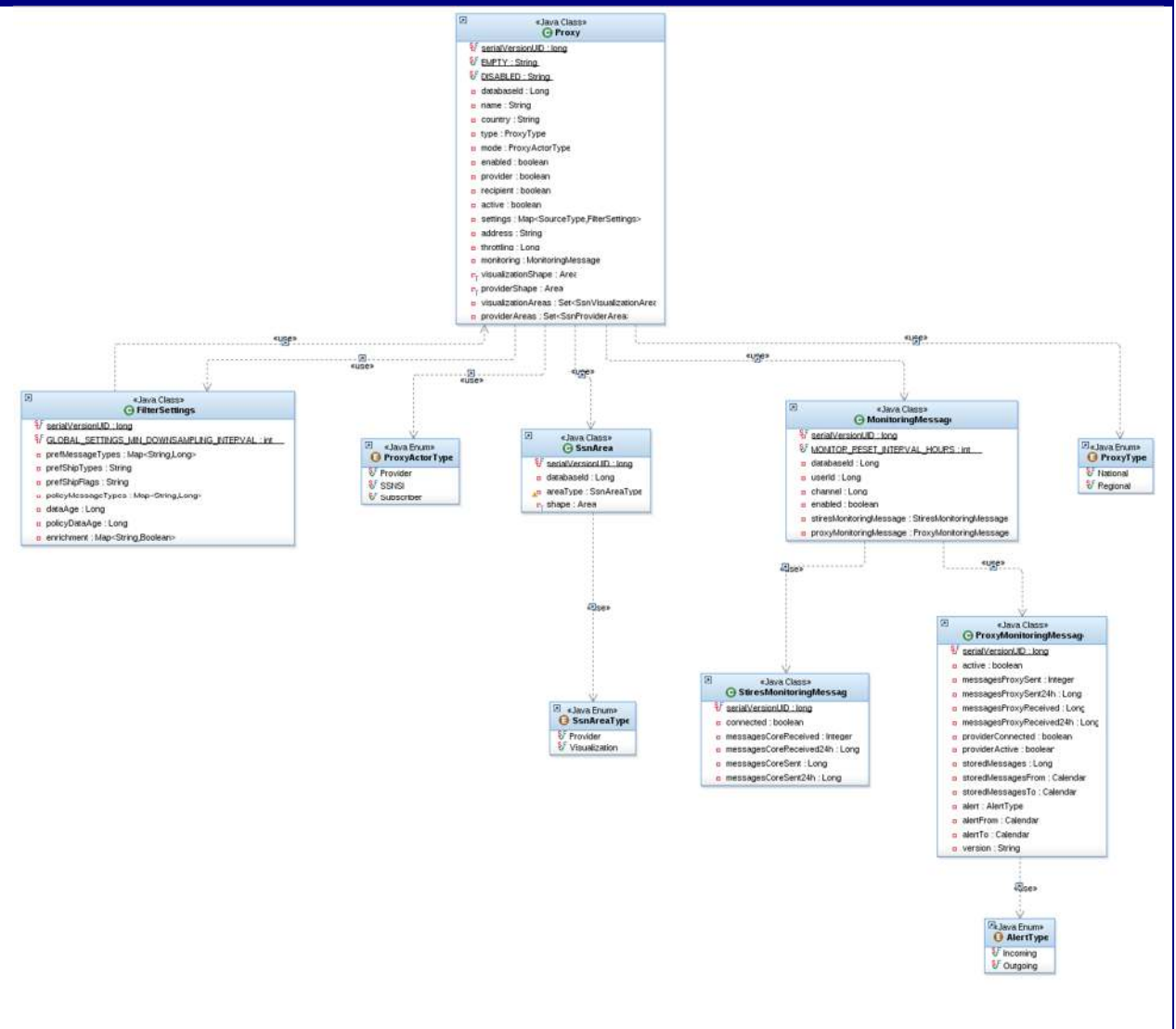
⁴the absolute value

Class Diagram : AIS message	
	<ul style="list-style-type: none"> ➤ The information included in the comment block; ➤ The time the message arrived at EMSA; ➤ The originator that provided this message; ➤ The recipients either defined in comment block or "decided" by the stires-vdm-core processing.
Class	<p><u>ReportedAisVessel</u></p> <p>This class extends the common VesselIdentifier class to represent the additional vessel attributes included in VDM sentences such as</p> <ul style="list-style-type: none"> ➤ Ship type. ➤ Ship Dimension: length, width. ➤ <u>ATON (AidToNavigationReport) information: type, name, status, flag</u> ➤ <u>Off position flag</u>
Class	<p><u>AisPositionReport</u></p> <p>This class encapsulates the various dynamic properties of AIS objects at a given point in time and space; it includes SOG, COG, ROT, longitude, latitude, true heading, data source.</p>
Class	<p><u>CommentBlock</u></p> <p>This class encapsulates the information supporting the Comment Block (CB) extension as defined in the IEC 62320-1 standard such as</p> <ul style="list-style-type: none"> ➤ Timestamp (c:), ➤ destination (d:) - identification of intended listener device or process for the attached sentence, ➤ source (s:)- identification of the talker device or process for the attached sentence, ➤ information (i:) described by the next class
Class	<p><u>CbInformation</u></p> <p>This class holds the data provided by the XML tags contained in the information comment block; it includes</p> <ul style="list-style-type: none"> ➤ identification of the system generating the information - tag <S>; ➤ information on the data quality - tag <Q>; ➤ information on the data originator - tag <O>; ➤ information on the data usage policy - tag <U>; ➤ information on a restricted set of recipients- tag <R>; ➤ information on the ship main identifiers - tag <I>.
Class	<p><u>SCbInfoCb</u></p> <p>This class extends the aforementioned CbInformation Class to provide satellite related information; it includes</p> <ul style="list-style-type: none"> ➤ the Satellite information - tag <T>; ➤ the position computation method that represents the method used to validate a position, by comparison or correlation with another observation (e.g. * EO) - tag <M>; ➤ the position validation method characterizes the accuracy/reliability of a position computed with a prediction algorithm or Doppler shift measurements - tag <N>.
Class	<u>SCbSatInfo</u>

Class Diagram : AIS message

	<p>This class provides S-AIS specific comment block tag <T>; it includes</p> <ul style="list-style-type: none"> ➤ The ground station acquisition timestamp; ➤ The data center ingestion timestamp; ➤ The data center delivery timestamp; ➤ The satellite id; ➤ The ground station id; ➤ frequency shift of arrival with respect to the centre of the AIS channel, in mHz; ➤ precise time of arrival within the detection second given after c:, in μs (optional)
Class	<p><u>AisMessageEntity</u></p> <p>This class extends the aforementioned AisMessage</p> <p>Class to provide information required by VDM core processing; it includes</p> <ul style="list-style-type: none"> ➤ the proxy (SSN-SI) and the originator from which the data was received; ➤ the message destinations (DB included with id -1); ➤ message visualization mask; it consists of the provider areas of the VDM message originator/proxy; ➤ Boolean attribute flags indicate the message's processing and routing, i.e. whether the message is or not to be: <ul style="list-style-type: none"> ○ saved on Db ○ <u>dispatched to proxies (SSN-SI)</u> ○ <u>sent to the component for voyage calculation</u>

Class Diagram : AIS connection & monitoring



Class	<p><u>Proxy</u></p> <p>This class represents the SSN-SI proxies that can be a Data Provider or a Subscriber or an instance of SSN-SI system. It provides information for</p> <ul style="list-style-type: none"> ➤ the mode acting – SSNSI, Provider or Subscriber; ➤ the name (originator) ➤ the address ➤ the maximum throughput - throttling (in msg/second) ➤ the proxy settings per source type ➤ the monitoring information.
Class	<p><u>FilterSettings</u></p> <p>This class represents the proxy settings for a specific source type. It includes:</p> <ul style="list-style-type: none"> ➤ the list of AIS messages type and their downsampled rate that proxy wants to receive ➤ the list of AIS ship types and their downsampled rate that

Class Diagram : AIS connection & monitoring

	<p>proxy wants to receive</p> <ul style="list-style-type: none"> ➤ data age for messages received ➤ the list of AIS messages type and their downsampled rate that proxy wants to provide ➤ data age for messages provided ➤ a boolean flag indicating that receiving data shall be only from neighbouring countries and ships under this proxy's flag ➤ a string holding the list of sforementioned lidt of countries in CSV. ➤ the flags for comment block information – including enrichment.
Class	<p>MonitoringMessage</p> <p>This class represents the traffic information between a proxy and SSN as well as probable alert occurrences.</p>

4.9 SSN-VMS

This section provides an overview of the SSN VMS System. SSN VMS System shall be a new part of current STIRES System; i.e. new modules shall be created to implement the required VMS messages. The SSN VMS System shall use the current SSN GIS Services to provide the functionality concerned with the visualization requirements.

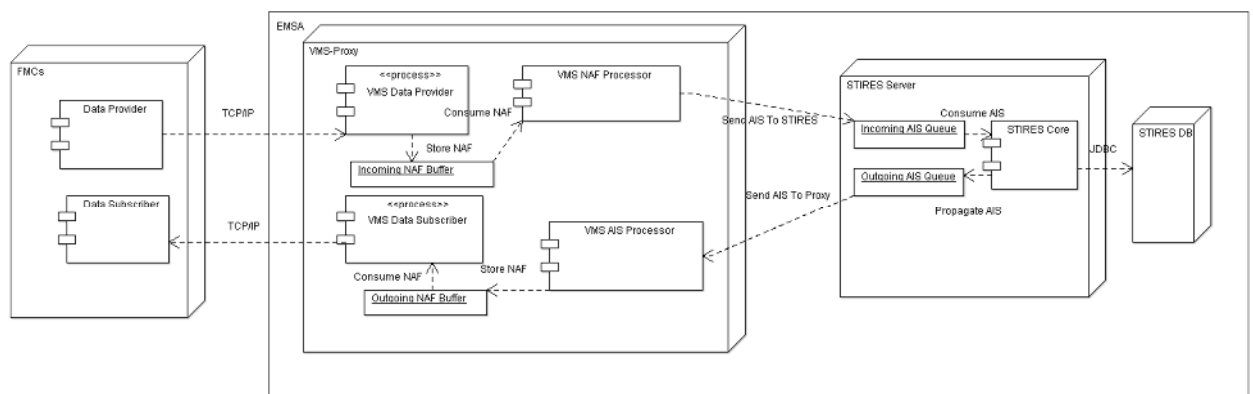


Figure 3-15 Connection diagram of VMS components.

The VMS System as depicted in the diagram presented in Figure 3-15 consists of the following subsystems:

1. VMS Proxy (*ref:* section) has three interfaces that provide:
 - a. "real time" data transfer from the data provider (FMCs);
 - b. "real time" data transfer to the data subscribers (FMCs);
 - c. data exchange in canonical format (RESTful services) with VDM Core.
2. **STIRES Core** (*ref:* section) provides:
 - a. RESTful service for data transfer from/to the VMS-Proxy;
 - b. PL/SQL stored procedures for the STIRES Database interaction;

3. **Web Application** includes the following subsystem:

Cartographic Viewer Services; the **SSN GI** shall be also **upgraded** to provide the visualization of VMS functionality.

4. **Database**

The first application (vms-proxy-core-app) materialises the main functionality of VMS at EMSA proxy level, which is the management of VMS position reports (in NAF format) in a way that is independent from the channel of communication through which messages are exchanged.

The second application (stires-core-app) materialises the main functionality of STIRES, which is the management of AIS messages (plus connection configuration/authentication settings, monitoring, etc) in a way that is independent from the channel of communication through which messages are exchanged.

The forth application (SSN GI) provides the graphical user interface (GUI) and handling user-to-business requests. It is the presentation layer providing the functionality of the VMS system.

The decomposition of VMS in three applications allows for:

- The disengagement of the business logic of VMS from the protocol for which it is offered. This disengagement allows also for the independent implementation of the business logic of VMS.
- The development of the communication protocol has local repercussions in the corresponding application and not in the entire system (VMS).

The VMS system is identified by the following entities owned by the "VMS Management" Business System

- AIS Data/Position Report
- Ship Particulars/Identification (including/plus IR)
- Proxy Identification
- FMC Identification (Provider & Subscribers)
- "use" the ssn-spm-domain entities Vessel, Location, Proxy
- VMS AIS Report Information that includes the above entities; i.e. it encapsulates the ships AIS position reports and the connections (FMCs Provider & Subscribers, Proxy, VDM Core) information

Thus, SSN VMS system provides a web service that enables the access to, and update of, these entities as shown in Figure 3-16.

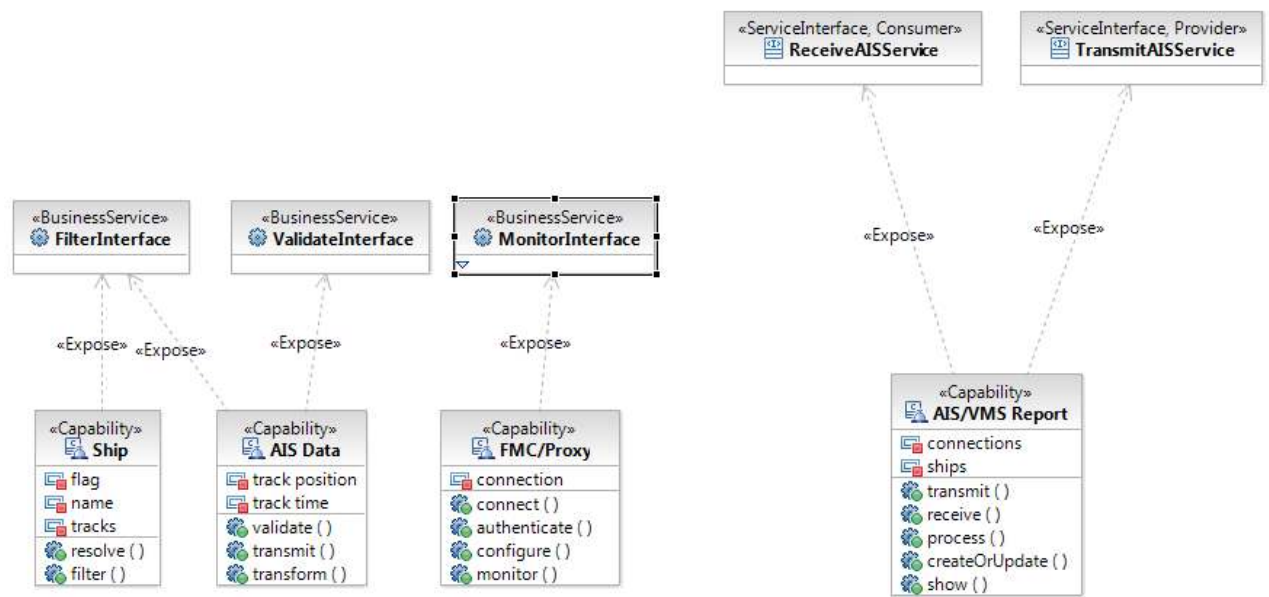


Figure 3-16 SSN-SI VMS Service provided operations.

4.9.1 Domain module/package

This module/package includes the definition of VMS entities (position reports, vessel, proxy, etc) and it is used by all the VMS sub-systems.

4.9.1.1 UML Class Diagrams

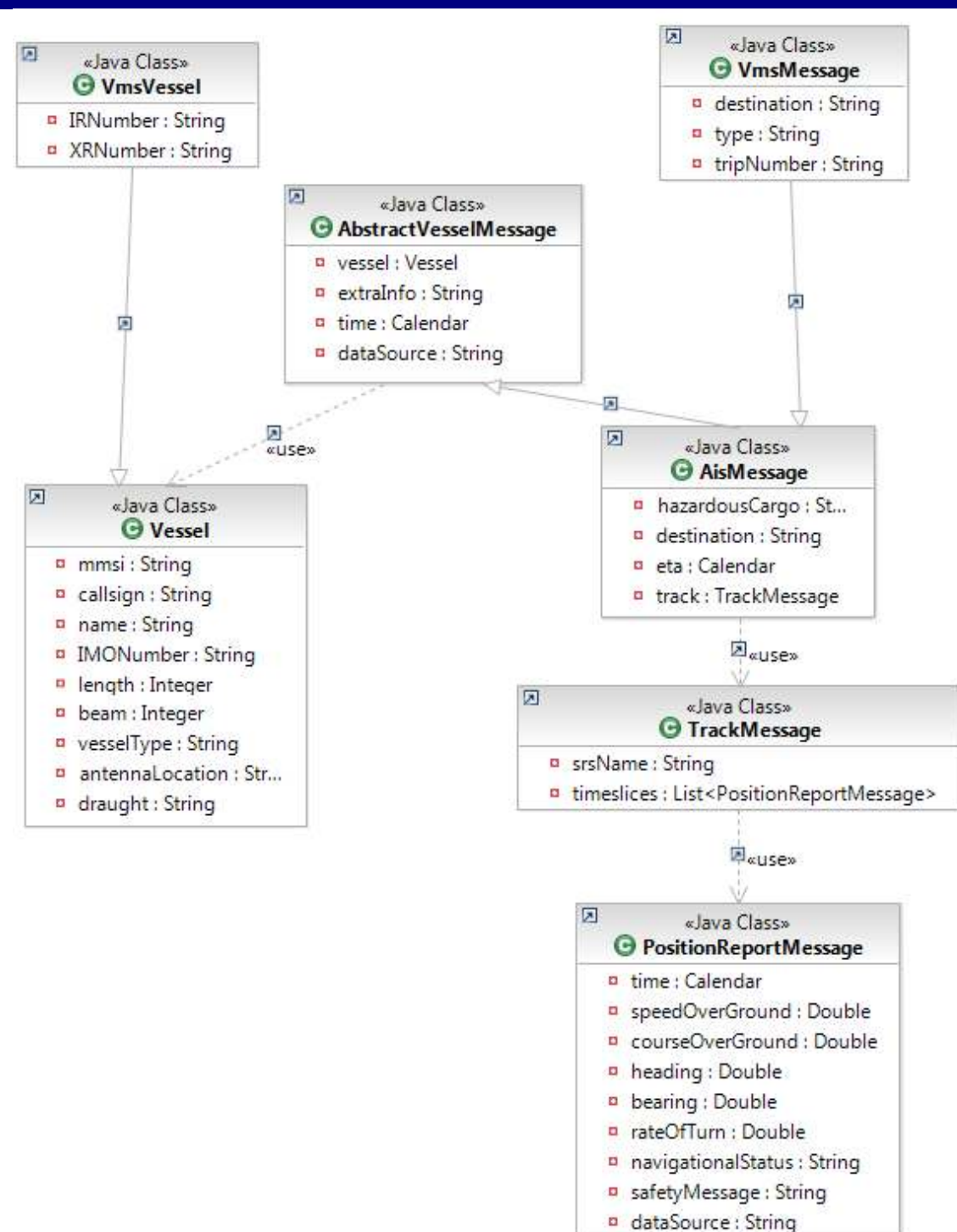
This section covers the architectural significant elements of the design model. It presents the definition of the most significant classes that will implement the requested functionality, organised into packages.

The classes are organised in packages according to the functionality they provide. A package is a general-purpose model element that organizes model elements into groups. Each package contains a set of classes and interfaces, representing what will become components in the implementation.

4.9.1.2 Module: ssn-spm-domain VMS specific

4.9.1.2.1 Package: ssn.spm.domain.vms

Class Diagram : vms vessel & message



Class	<u>Vessel</u> This class represents the vessel details.
--------------	---

Class Diagram : vms vessel & message	
Class	<p><u>VmsVessel</u></p> <p>This class extends the aforementioned Vessel class to represent the additional VMS attributes such as</p> <ul style="list-style-type: none"> ➤ Internal reference No: Vessel registration detail. Unique vessel number as flag state Alpha-3 ISO country code followed by number. ➤ External registration No: Vessel registration detail; the side number of the vessel.
Class	<p><u>AbstractVesselMessage</u></p> <p>The abstract message related to a particular vessel. It includes the following attributes:</p> <ul style="list-style-type: none"> ➤ vessel: ship information ➤ time: the message time stamp (in UTC timeframe) ➤ dataSource: Organizational source of data for the message and may represent a data service provider, a data management system, an AIS transmission system, etc.
Class	<p><u>AisMessage</u></p> <p>This class extends the aforementioned AbstractVesselMessage class to represent the additional AIS message attributes such as</p> <ul style="list-style-type: none"> ➤ eta: the expected time of arrival (in UTC timeframe) ➤ track: the instance of TrackMessage class (refer below).
Class	<p><u>TrackMessage</u></p> <p>This class is a sequence of specialized timeslices (PositionReportMessage) that indicate the dynamic status of the AIS message.</p>
Class	<p><u>PositionReportMessage</u></p> <p>This class encapsulates the various dynamic properties of AIS objects at a given point in time and space; it includes SOG, COG, ROT, longitude, latitude, true heading, data source.</p>
Class	<p><u>VmsMessage</u></p> <p>This class extends the aforementioned AisMessage class to represent the additional VMS message attributes such as:</p> <ul style="list-style-type: none"> ➤ destination: the address of the party receiving the message; ➤ type: the VMS message type, e.g. 'POS'; ➤ tripNumber: Fishing trip serial number in current year.

4.9.2 VMS-Proxy

VMS-Core Architectural Layer Dependencies

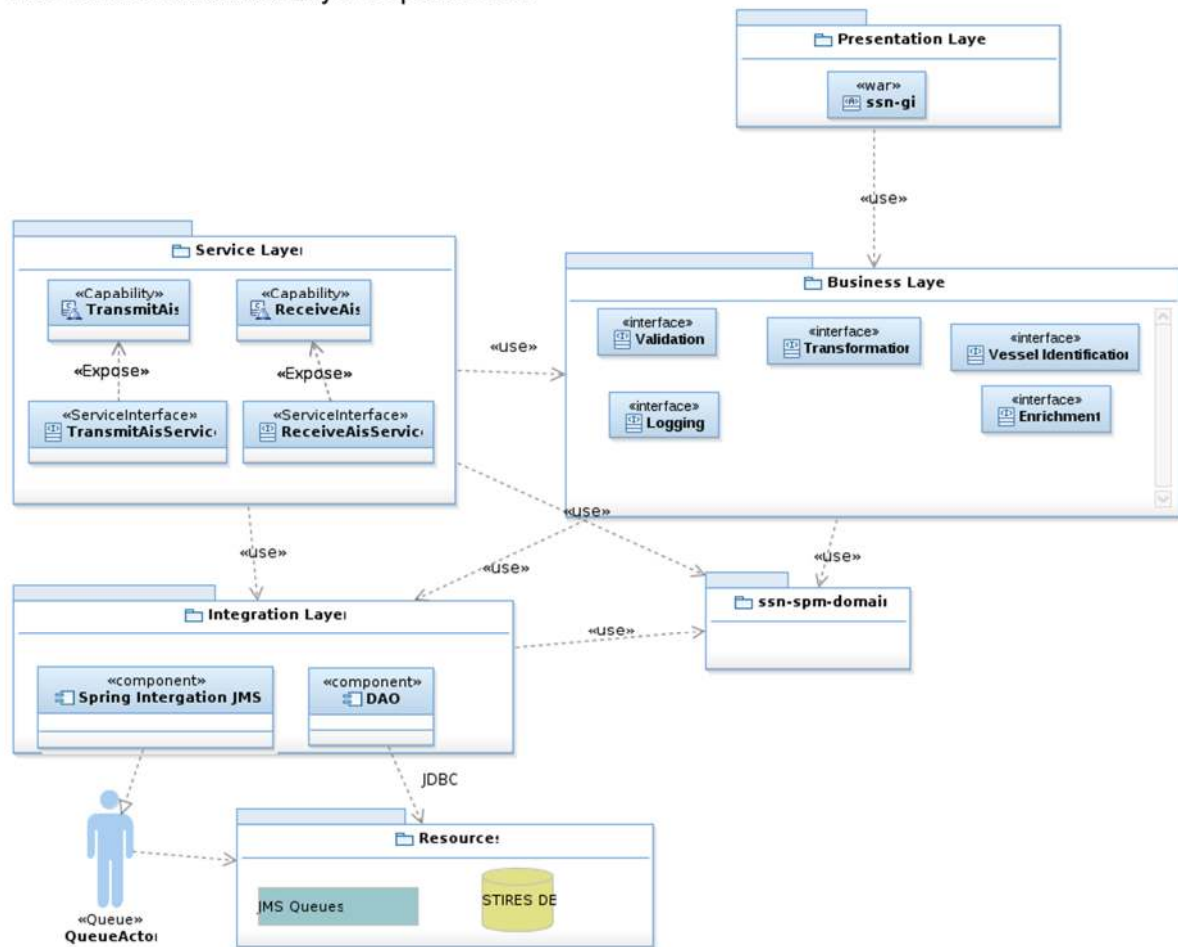


Figure 3-17 Architectural Layer dependencies (Logical view) of VMS-Proxy

The VMS-Proxy shall be implementing using Java and shall behave as **Light Service Bus** (providing features of Validation, Transformation, Queuing and Transportation using Spring Integration module).

4. Integration Layer

Thread management

- The **incoming NAF messages** (from Data Provider stored to **Incoming NAF Buffer**) shall be processed by a pool of "consumers" (VMS-Proxy NAF Processor/Data Processing Service). Using the Spring Integration module; each NAF message shall be validated and transformed to AIS message (POJO) by the corresponding channels. Then, the AIS messages shall be packed (XML message) and sent to STIRES system via HTTPS.
- The **incoming AIS messages** (AIS messages from STIRES Server via HTTPS) shall be processed by a pool of "consumers" (VMS AIS Processor). Using the Spring Integration module; each AIS message shall be transformed (common transformer bean) to NAF message by the corresponding channels. Finally, the NAF messages shall be stored to the **Outgoing NAF Buffer**. A consumer shall dispatch the NAF messages to the FMCs application (Data Subscriber) - according to vessel flag and the coastal fishing zones determined by the STIRES Core - via HTTP(S) connection when the connection to FMC Server could be established.

5. Business Layer

This layer provides the Data processing such as the message validation, transformation.

6. Service Layer

This layer exposes the VMS-Proxy functionality as RESTful Services.

Transmit AIS Report capability; it sends the AIS Report messages to STIRES Core System – XML messages over HTTPS.

The AIS Report messages include the Position Reports received from Data Providers.

Receive AIS Report capability; it accepts the AIS Report messages from STIRES Core System – XML messages over HTTPS.

The AIS Report messages include the Position Reports that STIRES received from sources different than FMCs.

4.10 SSN-SI

This section provides an overview of the SSN SI System.

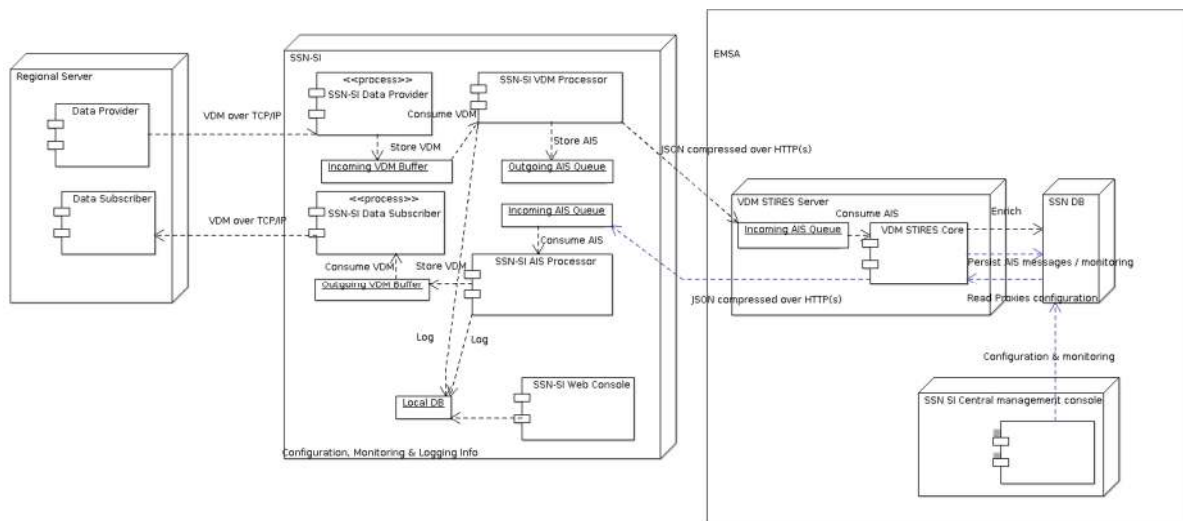


Figure 3-18 Connection diagram of SSN-SI components.

The upgraded SSN-SI System, as depicted in the diagrams presented in Figure 3-18, has four interfaces that provide:

- "real time" data transfer from the data provider (Regional Server);
- "real time" data transfer to the data subscribers (Regional Server);
- data decoding / encoding
- data exchange in JSON format (RESTful services) with VDM STIRES Core.
- storage/retrieval of configuration and monitoring data (e.g. distinct stream monitoring, filters, authentication, displaying new availability indicators, potential addition of configuration parameters, etc).

This application (ssn-si) materialises the main functionality of SSN-SI at national proxy level, which is the management of AIS position reports (in VDM format) (plus connection configuration settings, etc) in a way that is independent from the channel of communication through which messages are exchanged. The SSN SI application performs separate processing (and queuing) for T-AIS and S-AIS messages.

1. Business Layer

This layer provides the Data processing such as the message transformation, validation, identification, enrichment.

It provides/implements the common business processing as

- Decoding/Encoding of VDM sentences;
- Down-sampling;
- Throttling;
- Data aggregation;
- Monitoring.

Decoding of VDM sentences is performed by ssn-ais module – part of ssn-si application - the incoming VDM sentences are validated, aggregated (in case the message consists of multiple sentences), decoded in POJO and finally is mapped to AIS domain to be submitted to SSN STIRES; the decoder is upgraded to also decode the AIS Message 27 and the VSI sentences. The originator code (in case of provider connections in server mode) is assigned to the data if not transmitted by the data provider system in the comment block part of VDM sentence.

Encoding of AIS messages to be routed to the Subscribers is also performed by ssn-ais module; actually, the enrichment information is encoded in the Comment Block format and the VSI part (if it exists) is skipped – the probable VSI information of the initial VDM sentence is provided in the Comment Block part- parameter-code 'i'; the VDM sentence is kept in AIS message structure in raw format and it is added at the end of the aforementioned Comment Block part to compose the final VDM sentence.

Message filtering is performed by:

- data-age filter: a default age ($X = 3600$ seconds) is applied for filtering in all sites (STIRES database, SSNSI providers and subscribers;
- SSN-SI is able to downsample the AIS data per vessel (MMSI) and AIS message type (1 to 27) as presented in SSN VDM Core corresponding section. The configuration shall be defined for specific provider/subscriber connections (originator codes/recipient codes defined in the comment block) or source type (T-AIS or S-AIS).
- enrichment filter; the SSN VDM core applies the configured Enrichment filter attribute to the outgoing message transmitted to the corresponding proxies.

Throttling

A new parameter is defined to configure the maximum throughput (in msg/second) for the messages forwarded by the SSN SI to VDM (SSN); its default value is 100 msgs/sec.

When maximum throughput value is reached, the “extra” messages are stored on a JMS queue (PROVIDER_INCOMING or PROVIDER_SAIS_INCOMINGqueue). The processing of these messages stored in the JMS queue is done later, when the measured throughput is again below the limit.

Note: in case that no live data are received, no throttling is applied on messages stored on JMS queue.

Queueing and redelivery policy (ActiveMQ configuration)

In case of transmission failure from SSN SI to SSN ecosystem the messages are stored in JMS PROVIDER_INCOMING or PROVIDER_SAIS_INCOMING queue (depending on the source type). These messages will be re-attempted to be delivered according to following a redelivery policy:

- initialRedeliveryDelay=60 seconds, meaning that the first attempt to redeliver is 1 minute after the message is placed in the queue

- backOffMultiplier=2, meaning that the 2nd attempt to redeliver is 2 minutes after the first redelivery (failure) - i.e. after 3 min of the message was placed in the queue -, the 3rd time attempt after 4 min (7 min of the message persistence), etc.
- maximumRedeliveryDelay=600 seconds, meaning that the time between attempts is set at maximum to 10 minutes, despite the value calculated by the abovementioned backOffMultiplier parameter
- SSN SI will continue to transmit until SSN_SI_AMQ_TTL or SSN_SI_SAIS_AMQ_TTL (i.e. time-to-live defined in seconds) is reached, depending on the source type.
- Preservation of the order of messages: FIFO

The messages coming from SSN (VDM application) are firstly stored in STIRES_INCOMING queue for performance reasons; i.e. asynchronous consumption of SSN (VDM) messages. The redelivery policy is the same described above; however, the consumption of a JMS message is rolled back only in exceptional cases (SSN SI system failure).

Data Aggregation

For performance reason, the decoded (in POJO) and filtered (downsampling) messages are aggregated according to its originator. The structure holds the aggregated data per originator and source type (T-AIS, S-AIS) is transformed to JSON, zipped and submitted to SSN.

2. Integration Layer

TCP/IP connection

The Socket component is responsible to establish, manage and monitor the connection between the SSN-SI system and the regional servers (providers/subscribers). It implements a "keepalive" mechanism. It also provides the filtering of IP addresses if this feature is enabled for regional servers (Providers and Subscribers).

It establishes TCP/IP (raw Socket) connection with the Provider application to receive (Data Provider) and send (Data Subscribers) VDM sentences.

Two Buffers are used to store/buffer the incoming and the outgoing VDM sentences.

3. Service Layer

This layer exposes the SSN SI functionality as RESTful Services.

Transmit AIS Report capability; it sends the AIS Report messages to SSN VDM Core System. The AIS Report messages include the Position Reports received from Data Providers.

Receive AIS Report capability; it accepts the AIS Report messages from SSN VDM Core System. The AIS Report messages include the Position Reports with the enrichment information.

RESTful Services are also used for the communication of the SSN SI core application with the SSN SI console application (described below) to provide the configuration and monitoring capabilities.

4. Presentation Layer

This layer provides the configuration and monitoring functionality of SSN SI in HTML interface. The SSN SI console application

Monitoring capability; it provides the visual monitoring of each individual connection:

- SSN Ecosystem – distinguishing the incoming and outgoing streams;
- Each provider connection (client and/or server) with possibility to expand per originator code (connection of server mode);
- Each subscriber connection (client and/or server) with possibility to expand per originator code and recipient code (server);
- Message timeliness (timestamp of the position received vs current system timestamp).

- Alert logs;
- Availability metrics of AIS data and SSN SI machine/component.

Configuration capability; it provides:

- The settings definition for the providers' and subscribers' downsampling and filtering;
- Threshold definition;

4.11 SSN SI Central Management Console

This section provides an overview of the SSN SI Central Management Console. It is the presentation layer provides the management and monitoring of VDM providers and subscribers. It actually segregates the management of proxies (and corresponding authorities) from the SSN user/authority management a.k.a. Operational Organisation Database (OOD). The new console is accessible via the existing SSN Liferay portlet (emsa-portal-ssn-portlet); a new regular role named SSNSIC is created – and the associated permission VIEW_SSNSIC – that enables to access the new aforementioned console. This console provides:

- the definition and the configuration of proxies/SSN SIs; the definition of originator (and recipient) codes shall be also covered; A new filter parameter shall be added to provide the possibility of sending out data of either (both options should be possible):
 - specific combinations of originator codes or
 - AIS dynamic/position messages inside a geographical polygon shaped areain addition to data of ships belonging to a certain ship flag (regardless of the other restrictions);
- the definition of the thresholds for incoming or outgoing data;
- the definition and the configuration of the comment block part of the VDM sentence; i.e. filter attributes enable or disable the distribution of comment block information such as:
 - ship identifiers (tag I)
 - enrichment information (tags E, P and L)
 - Quality and S-AIS specific information (tags Q, T, N, M)
 - usage policy (tag U) – in this case it shall also be possible to assign a certain value
- monitoring at proxy and originator level; filtering indication shall be included;
- service availability monitoring with daily, monthly and yearly metrics recorded and exposed by each SSN SI and also recorded by SSN VDM core;
- "bulk" management of Parties Interface via a new menu option called e.g. "External interfaces";
- monitoring of Parties Interface; i.e. status: Enabled / Disabled.

Two EIS tasks shall be able to access the console functionality;

- SSN SI manager (PROXIES_MANAGER already exists) has full access, i.e. can change all configuration settings and see monitoring information.
- SSN SI operator (a new task named PROXIES_OPERATOR) – has limited access, i.e., read-only access to configuration settings and see monitoring information.

New EIS tasks shall be created to give access to "External interfaces" functionality of this console:

- SSN_LRIT_MANAGER and SSN_LRIT_OPERATOR (read-only) for the configuration settings and monitoring information of LRIT interface.
- SSN_IMDATE_MANAGER and SSN_IMDATE_OPERATOR (read-only) for the configuration settings and monitoring information of IMDATE interface.

4.12 SSN GI - DEPRECATED

The main aim of this intervention is to implement a new visualisation mechanism that will allow for a more efficient management and graphical rendering of spatial information and, in particular, of the large amount of vessel position data that are constantly being entered into the system. The goal of the new design is not to dismiss or provide replacements for all aspects of the existing system's architecture. Rather, it intends to improve on the current architecture by adopting previous design decisions and elements that were based on sound software development principles and technologies, and modify only those aspects of the system that it is required in order to meet the new specifications. As such, viewing the new system from the perspective of its conceptual architecture, as displayed in Figure 4-19, the new design is consistent with that of the existing system. Deviations to the original concept become apparent at a lower level and are limited to the visualisation scheme utilised for rendering spatial information, while other design aspects relating to the business domain and data access layers remain mostly the same.

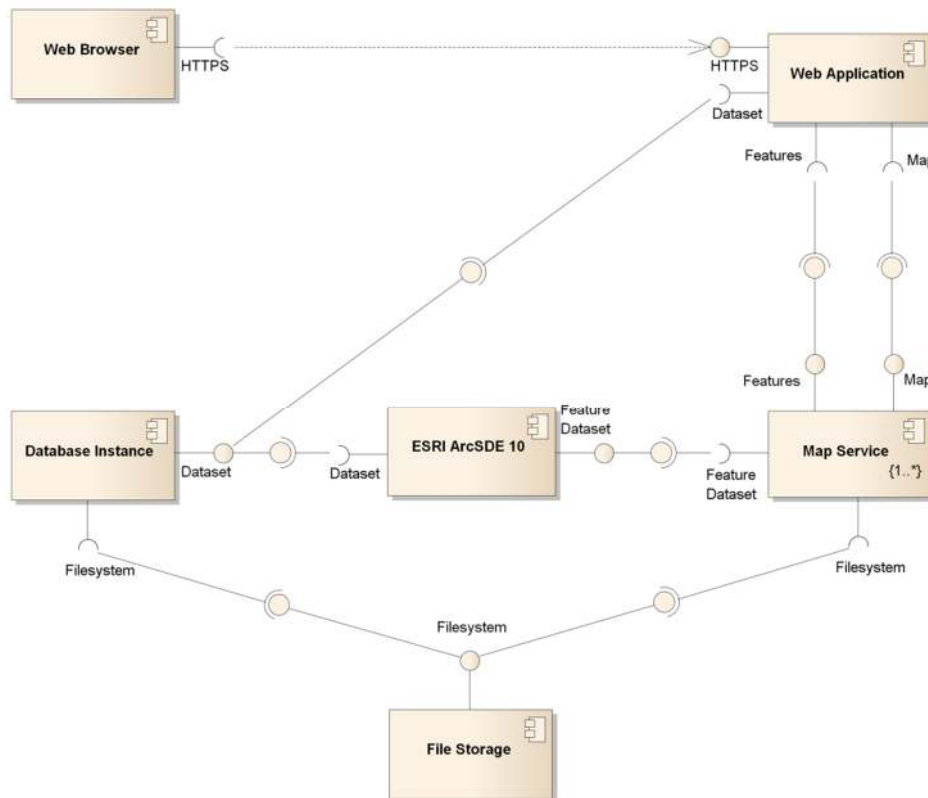


Figure 4-19 Component diagram providing a high-level view of both the existing and the new system's architecture

The following section presents the design of the new visualisation scheme for spatial information, which includes the implementation of a client-based, graphic rendering engine, as part of a new, redesigned user interface, the consolidation and reorganisation of the existing map services and layers deployed via ArcGIS Server and the remodelling of the related parts of the existing system's database.

4.12.1 Spatial Information Visualisation Scheme

As previously mentioned, the new system intends to replace the existing system's mechanism for visualising spatial information through the introduction of an improved graphic rendering engine. The need for a new rendering engine stems from the fact that, in its current form, the application depends on ArcGIS server's own drawing engine for rendering both static (e.g. world map, maritime

information, ports etc.) and dynamic (e.g. vessel tracks, routes etc.) geographic information. Even though this technique works sufficiently well for displaying mostly static information with low user interactivity requirements, when it comes to displaying dynamic information via a feature-rich GUI it becomes largely inefficient. This is due to the fact that, since the visual information is prepared and rendered on the server side, the client (web-browser) is unaware of the kind of information that is being displayed. Consequently, even for the simplest of user actions like hovering over a vessel track, the browser resorts to having to perform multiple queries to the server in order to discover what information is contained on the application's viewport.

In order to enable the application to provide users with a more feature-rich GUI with a high rate of interactivity and low latency between user actions, it is inevitable that a lot of the information rendering functionality is transferred from the server to the web-browser. In this way, the latter becomes more aware of the information it is required to display and can minimise the amount of round-trips to the application server as a result of the queries that need to be performed, improving the general performance of the application by orders of magnitude. Also, since the required geometrical and textual attributes of the vessel tracks will be transferring to the web-browser for rendering, many of the ArcGIS server mapping services/layers that are currently used for distinguishing between the various types of data sources and vessel types and for indicating enrichment attributes and labelling options can be discarded as they will no longer be necessary. As a result, the remaining services will be reorganised so that they provide much more efficient access to the database, while the latter's schema will be remodelled and become normalised, so as to avoid having to perform multiple database queries in order to retrieve information from multiple data sources, as in the existing system. The proposed modifications will also aid in increasing application performance by lessening the burden of the application, GIS and database servers.

4.12.2 Structure

In the new design, as shown in Figure 4-20, even though the map-viewer control is still directly responsible for requesting and displaying background, base-map images from the respective web services, requesting image overlays that contain vessel and port icons are no longer part of its duties. This responsibility now lies with the new, JavaScript-based rendering engine, whose major task is to perform the appropriate queries requesting spatial feature data instead of tiled image overlays. This data is consequently processed, and the appropriate vessel and port icons are created which are then inserted in the graphics layer of the map-viewer control as interactive graphical objects.

Displaying graphical and textual decorations over vessel and port icons is also part of the functionality of the client rendering engine, a fact that is reflected by the significantly smaller number of layers that are required in the new visualisation scheme. Another important aspect of the new scheme is the fact that, concerning the vessel data sources, i.e. AIS, LRIT and VMS, there is no longer a distinction between them at the map service level and also, partly, at the database level. In the new system, the vessel data source becomes just another attribute of all vessel tracks which are maintained in a collective fashion. In comparison with the current system, the proposed consolidation of map services and database tables will result in a threefold decrease of the total number of vessel related web and database queries that are required to fully render a map instance at runtime.

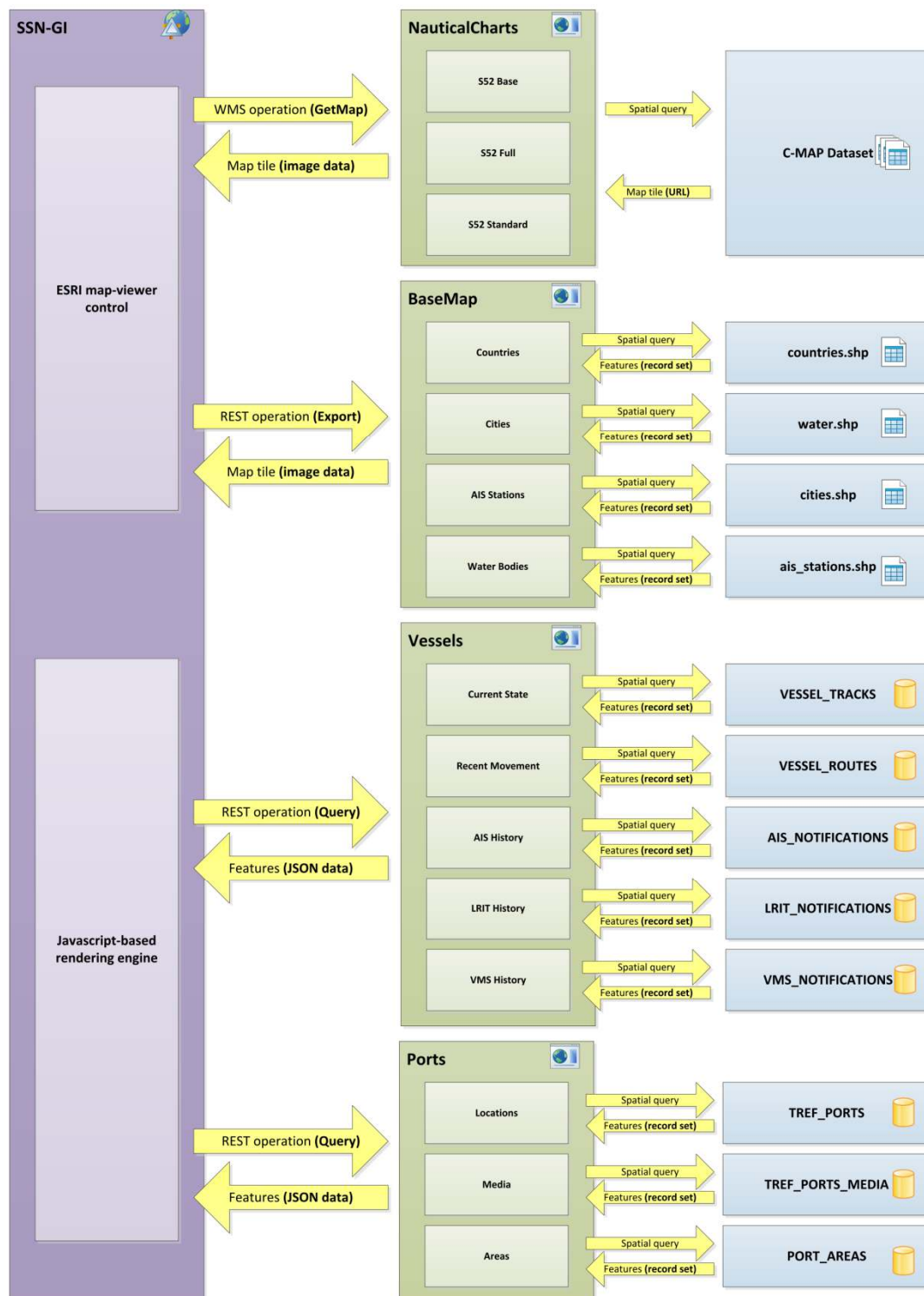


Figure 4-20 Interaction paths between the various elements of the new system's spatial information visualisation scheme

4.12.3 Behaviour

The graphical rendering of spatial information in the context of the new visualisation scheme is split into two distinct phases:

- rendering geographical features as static information, placed in the background
- rendering vessel features, port features and other graphical shapes as interactive objects, in the foreground

As shown in Figure 4-21, both of these processes execute asynchronously, triggered by user actions such as panning or zooming that modify the map's viewable area, i.e. the map's extent.

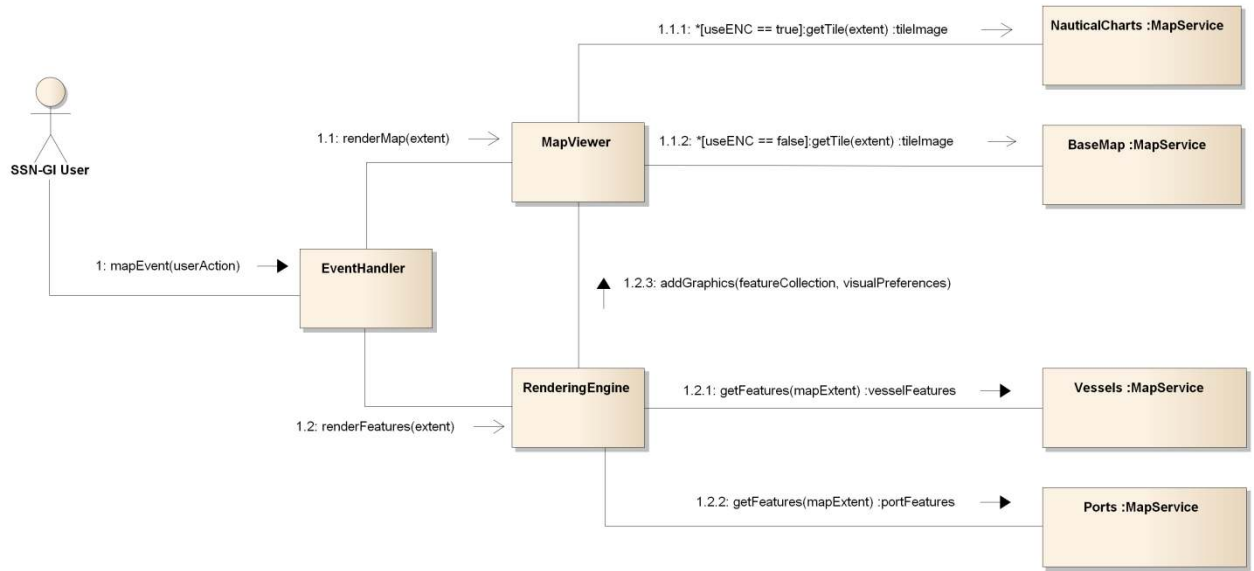


Figure 4-21 Communication diagram of a typical map rendering cycle

For the first phase, the same mechanism that is currently in use in the existing system is also employed in the new one, i.e. user actions changing the map's extent like panning and zooming, become delegated to the map-viewer web control, which in turn depends on appropriate HTTP queries to web services for retrieving tile images. Since the information to be displayed during this phase is static by nature, map rendering performance is generally dependent on I/O bandwidth since most or, in some cases, all images returned to the map-viewer control by the web services are retrieved from spatially aware file caches containing pre-rendered tiles. There are various factors affecting the population scheme of these caches, i.e. whether the entire dataset is pre-rendered in advance or whether it is generated on demand, however, since they do not have direct impact on the application's design, these factors will all be considered during the project's implementation phase and an appropriate configuration will be put in place.

For the second phase, the custom designed rendering engine performs spatial queries to the web mapping services that, instead of returning rendered image overlays for the map, they rather return feature objects serialised in JSON format. These objects undergo further processing in order to be converted into graphical objects, suitable for being overlaid on the map and exhibiting interactive functional attributes such as context menus, custom cursors, pop-up windows etc. Since the generated graphical objects are inserted in the DOM tree of the client browser, in order not to hinder the performance of the latter, a special clustering algorithm will be employed by the rendering engine for cases where the number of objects exceeds a pre-specified limit. This algorithm will be applied specifically on vessel track icons, which in particular zoom levels and/or port areas can amount to a range of many thousands. In those cases, neighbouring icons are clustered into single, larger graphical objects that have the form of rectangles, labelled using the total number of clustered tracks.

The mechanism for rendering other user drawn graphical shapes such as points, lines, polygons and text, as well as newly introduced object types such as placemark icons, remains the same as in the existing application. Those objects stored to and retrieved from the database directly in the form of JSON formatted strings, without relying on map services.

4.13 SSN-GI / EIS notification details request protocol mechanism upgrade - DEPRECATED

This section describes the mechanism of communication between the SSN-GI and EIS applications and, in particular, the request/response protocol of EIS notification details exchange. The current situation is presented, identifying the various shortcomings of the present scheme and the proposed improved solution in order to overcome them.

4.13.1 Current mechanism

In its current form, the mechanism by which the SSN-GI application requests notification details from EIS is realised through the asynchronous exchange of XML protocol request/response messages (Figure 4-22). A notification details request is initiated by the SSN-GI front-end (web browser) as a parameterised HTTP GET request. Once received by the SSN-GI back-end, the request parameters become marshalled into an XML request message that is in turn sent as payload of an appropriate HTTP POST request to the EIS back-end for further processing.

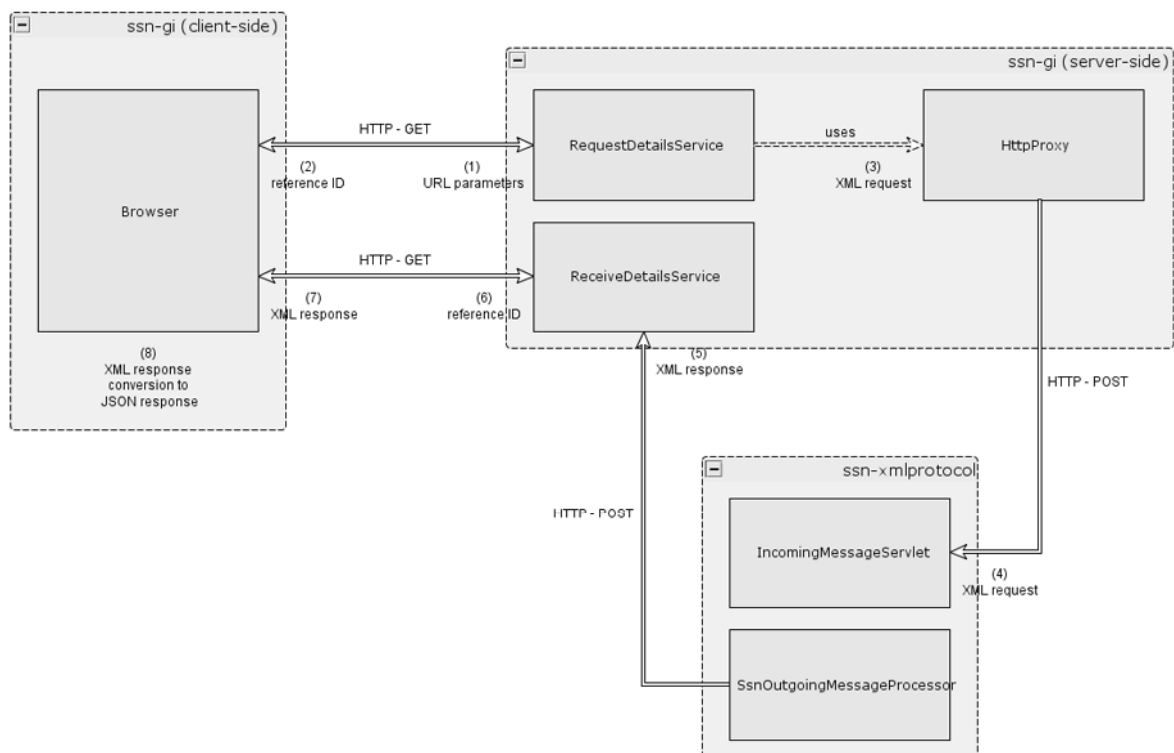


Figure 4-22Current EIS notification request protocol mechanism (numbers in parentheses are indicative of the sequence of information delivery – operations are otherwise asynchronous)

After the message has been processed in EIS, a new HTTP POST request is initiated towards the SSN-GI back-end, whose payload consists of the notification details XML response message. Because

of the asynchronous nature of this mechanism, during the time that a request is being processed by EIS, the browser periodically polls the SSN-GI server using a reference ID number that was assigned to the original request, until the appropriate response message from EIS becomes available, in which case it is passed as the response to the browser. There it becomes transformed into a JSON response object to be used for populating the appropriate GUI elements.

Even though this mechanism is computationally less expensive than the respective mechanism of previous versions of WEB-GI v2.1, which also involved additional steps in message transport through JMS queues, it is still not the most efficient way to transfer notification details information between GI and EIS. The participation of the *ssn-xmlprotocol* module in the current mechanism introduces many unnecessary conversion and mapping operations between URL parameters, application domain objects, protocol objects (JAXB), XML content and JSON. Also, the asynchronous processing implemented by the *ssn-xmlprotocol* enforces the use of polling from the client-side that, under certain operational conditions, can potentially hinder application performance.

4.13.2 Proposed improvement

In the solution proposed herein, it is suggested that the *ssn-xmlprotocol* module becomes excluded from the new mechanism and the whole process becomes replaced by a synchronous HTTP transaction between the SSN-GI front-end and EIS itself, with the SSN-GI server assuming the role of an intermediary component between the two (Figure 4-23). In the new scheme, the HTTP GET notification details request originating from the browser is sent as-is to EIS by the SSN-GI back-end, which acts as a simple, protocol-agnostic proxy. This solution removes SSN-GI back-end dependencies to JAXB generated objects, used for marshalling requests into XML messages and also introduces the *ssn-message-web* EIS module. It acts as an HTTP “bridge” between various other system components and the EIS application domain, providing information in browser-friendly JSON format. It shall be noted that this mechanism can be utilized for the interface between SSN-GI and EIS to exchange all types of SSN request & response messages.

The proposal ensures that the 2 systems (SSN-GI and SSN-EIS) can be deployed on different servers.

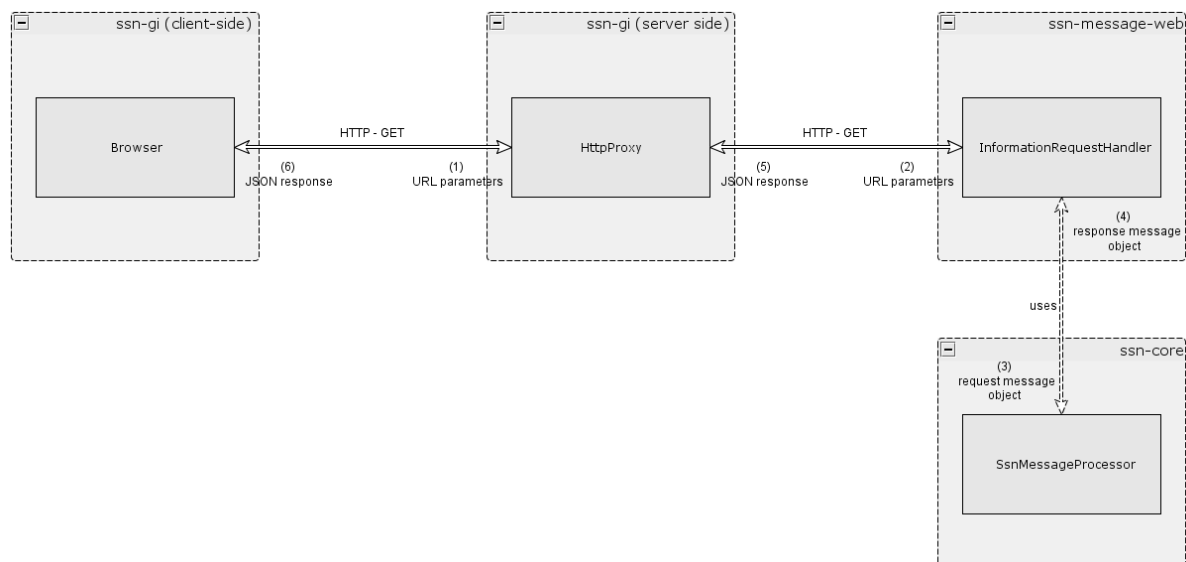


Figure 4-23 Proposed EIS notification request synchronous protocol mechanism

4.14 Message Queues

The *ssn-core-app* application uses four queues as listed in Table 4-12.

Each queue is capable of accepting multiple types of messages:

Queue	Accepted Message Types
Ingoing Queue	<ul style="list-style-type: none"> • InformationRequest (MS2SSN_<SSN_type>_Req) • AdditionalInformationReply (MS2SSN_<SSN_type>_Res)
Outgoing Queue	<ul style="list-style-type: none"> • AdditionalInformationRequest (SSN2MS_<SSN_type>_Req) • InformationReply (SSN2MS_<SSN_type>_Res)
Notification Propagation Queue	<ul style="list-style-type: none"> • Notifications <ul style="list-style-type: none"> ○ Vessel notifications with sat service ○ Incident Reportnotification ○ ShipCall Request / Response. Invalid PortPlus notification content notification
WaitForReply	<ul style="list-style-type: none"> • AdditionalInformationRequests

Table 4-12 Types of messages per Queue

The queues namely incoming, outgoing and notification propagation messages queue – are used to communicate in an asynchronous manner with its clients (in this case the ssn-xmlprotocol-app). These queues offer, basically, point-to-point⁵ communication.

Additionally, the ssn-core-app application uses a forth queue – named WaitForReply queue – in order to keep a copy of the **AdditionalInformationRequests** that performs to a data provider. Periodically (refer also to 4.2.1.5), ssn-core-app polls this queue in order to discover AdditionalInformationRequest messages for which the responsible data provider MS has not reply within a time interval. This time interval is the timeout value of the corresponding request message. It should be noted that this timeout value is also included in the generated request message for additional information.

Furthermore, ssn-core-app when processing a data provider reply (AdditionalInformationReply), uses the WaitForReply queue to find the initial data request – stored within a AdditionalInformationRequest – in order to produce the appropriate InformationReply message.

In other words the wait for reply queue is mainly a temporary storage. The major advantage of adopting a queue as a temporary storage as opposed to the use of the RDBMS is that the application server effectively serializes (using java serialization) the messages and there is no need to develop and manage a separate a persistence layer.

End points:

- The ssn-core-app application maintains a permanent listener that consumes messages from the incoming message queue and following the “[process manager](#)” design pattern. The listener is implemented as the *Message Driven Bean* **IncomingMessageProcessorMdb** located under the package **ssn.message.processor.ejb**, which is a proxy of the actual process manager **IncomingRequestReplyEnvelopProcessor** located at the package **ssn.message.processor**. It should be noted that in case of any exception during the processing of the message, this message is discarded.

⁵ A Point-to-Point Channel ensures that only one receiver consumes any given message.

- This process manager uses the following routing table:

Message Type	Processor
InformationRequest (MS2SSN_<SSN_type>_Req)	ssn.request.processor.RequestProcessorImpl
AdditionalInformationReply (MS2SSN_<SSN_type>_Res)	ssn.reply.processor.ReplyProcessorImpl

Table 4-13 Process manager – routing table

4.15 Vessel Management – Upload Single Hull Tankers

Business process specifics	
Responsibilities	<ul style="list-style-type: none"> • Load onto the database a list of Single Hull Tanker. • The list of codes is provided in a predefined comma separated (CSV) file format.
User role(s)	The EIS system administrator performs the task upon request.
Action History	<ul style="list-style-type: none"> • The action will be logged into the database log tables (TLOG). • A log file will be created listing the processing of each record.
Processing logic	<ul style="list-style-type: none"> • An Oracle database External table exists that maps the definition of the CSV file. External table makes use of the SQL Loader functionality. • The uploaded CSV file is stored as blob on SSN database. The "upload" action triggers the procedure being described. • The uploaded file is extracted as ASCII file on an Oracle directory and the external table is altered so to access the data of the extracted file. • The data of the external table are copied to the stage area. On copy, a flag is initiated that indicates when a vessel already exists on the registry. The EIS system administrator validates the data on stage area and classify the vessels to be stored on Vessels registry or to be ignored. • After the EIS system administrator validation, the data from the stage area are merged to the operational tables. New vessels identified by IMO numbers that do not exist on the Vessels registry, are inserted; one record is created on the Vessels table including the IMO number and a second one on the Vessel_Detail_Versions table including the MMSI number, the call sign and the ship name. <p>Existing Vessels are updated: New versions of vessel details identified by non-existing MMSI numbers, are inserted; otherwise the call sign and the ship name are updated.</p>
Validation Rules	<ul style="list-style-type: none"> • Vessels must be identified by IMO number. • All the new Vessels will be activated and classified as valid upon administrator validation.

Business process specifics	
	<ul style="list-style-type: none"> All the Vessels – new and updated - will be classified as single hull tankers upon administrator validation.
Database Transactions	<ul style="list-style-type: none"> Create an Oracle directory that points to an existing path in the hosting server. The Oracle database owner must have full access to that path. Create an External table to map the definition of the CSV file. Create a stage area to allow the EIS system administrator identifies the Vessels of the CVS file to be stored on Vessels registry. Merge data in the operation tables that store the definition of Vessels.

4.16 Vessel Management – OSD Synchronization

Business process specifics	
Responsibilities	<ol style="list-style-type: none"> Update the OSD (Operational Ship Database) based on the newly updated CSD (Central Ship Database) data. OSD semi-automatic update. Automatically compare the new vessel records created in EIS OSD during the last day against the vessel records stored at the MarInfo database. Differences will be stored in a stage area, an operator via the Management Console will verify and accept differences.
User role(s)	<p>The programs are scheduled for execution on a daily basis.</p> <p>Update OSD is executed 1st while the semi-automatic OSD update follows.</p>
Action History	The action will be logged into the database log tables (EIS_ADMIN.TLOG).
Processing logic	<ol style="list-style-type: none"> OSD update: The procedure will consider only the updated records following the last EIS OSD update. Select all IMO+MMSI pairs from the CSD. Compare each IMO+MMSI pair with the vessel records listed in EIS OSD. <ul style="list-style-type: none"> If the OSD vessel Temporary record has null IMO and is resolved based on the IMO+MMSI pair from CSD. Then if the IMO+MMSI pair is already defined as a vessel in EIS OSD then the notification reference will be updated to point to the IMO+MMSI pair vessel and the Temporary record will be deleted. Alternatively, if the OSD vessel record is NOT resolved based on the IMO+MMSI pair from CSD it will remain as "Temporary" for the semi-automatic procedure. <ul style="list-style-type: none"> The IMO+MMSI pairs from CSD will be compared with the vessels in EIS OSD with IMO NOT NULL to identify any new vessels or update any existing valid vessel details. Newly identified vessels (that is new active IMO+MMSI pairs) will be inserted in the EIS OSD. A check with the EIS OSD vessels will ensure that a pair is new. The OSD vessel flag IS_VALIDATED will be updated/defined based on the IMO+MMSI pair status in CSD.

Business process specifics	
	<p>If the status is 0 for NonActive the OSD vessel will be flagged as Temporary.</p> <p>If the status is 1 for Active the OSD vessel will be flagged as Valid.</p> <p>If the status is 2 for Active IMO NonActive MMSI/pair the OSD vessel will be flagged as Temporary.</p> <p>The particulars of a vessel that already exists are updated based on the details of the pair from CSD; to maintain the history of changes the previous values will be kept and a new record with the new particulars is created. The comparison is based on the IMO+MMSI pair while the particulars to update include the CallSign, ShipName and vessel Flag.</p> <p>2. OSD semi-automatic update: The validation procedure automatically compares the new vessel records created in EIS OSD during the last day against the vessel records stored at the Central Ship database.</p> <p>The comparison of the EIS OSD vessel records and the MarInfo DB are based on the IMONumber if the IMONumber exists in the SSN DB. In case a record cannot be identified distinctively (e.g. compared based on MMSI when more than 1 potential entries exist from one MarInfo source), the latest entry is selected; it will be up to the user performing the manual intervention to decide if the vessel will remain unresolved or the status will change to InValid or accepted as Valid.</p> <p>Based on the comparison results the EIS OSD vessel records are updated.</p>
Validation Rules	<ul style="list-style-type: none"> The validation rules for vessel identification are applicable in [Error! Reference source not found.].
Database Transactions	<ul style="list-style-type: none"> Both programs read data from the CSD. The OSD update modifies the values of the resolved vessel directly in the EIS tables that hold the definition and particulars of vessels. The semi-automatic update stored the divergent values in a table specifically defined for holding the vessel divergent particular's values.

4.17 Vessel Management – CSD Synchronization

Business process specifics	
Responsibilities	<ol style="list-style-type: none"> Update the CSD (Central Ship Database) based on the newly updated OSD (that includes , MarInfo data. OSD semi-automatic update. Automatically compare the new vessel records created in EIS OSD during the last day against the vessel records stored at the MarInfo database. Differences will be stored in a stage area, an operator via the Management Console will verify and accept differences.

Business process specifics	
	Sources SSN, MARINFO, MARS, LRITDB, THETIS, SSN_MS, MS
User role(s)	The programs are scheduled for execution on a daily basis. Update CSD is executed prior to the execution of the OSD semi-automatic update and OSD update.
Action History	The action will be logged into the database log tables (REF_ADMIN.TLOG).
Processing logic	Update of the ship records - including the key particulars like IMO, MMAI number, Ship Name and Call Sign - stored in the CSD database by comparing the ship records based on the active IMO+MMSI pairs identified in the various sources. Sources include: Sources SSN OSD, MARINFO (LLI, IHS), MARS.
Validation Rules	The IMO+MMSI pairs must be found in at least 2 of the sources.
Database Transactions	<ul style="list-style-type: none"> Read data from the OSD and CSD. The CSD update modifies the values of the ship record directly in the CSD tables that hold the definition and particulars of vessels.

4.18 EIS & STIRES interoperability – Get Enrichment data

Business process specifics	
Responsibilities	Provide STIRES with Notification and vessel specific data upon request.
User role(s)	STIRES requests EIS for the enrichment data for a given vessel.
Action History	The enrichment per vessel will be done every 2 hours and prior to the sending of the Ship (AIS) notification to EIS Any errors are logged into the database log tables (TLOG).
Processing logic	<p>STIRES requests EIS for the enrichment data by IMO Number and/or MMSI Number. At least one of the 2 must be specified and must be technically correct.</p> <p>EIS will use the IMO and/or MMSI as input variables. EIS will resolve the vessel based on the EIS OSD and will provide in response:</p> <ol style="list-style-type: none"> Vessel particulars: IMO, MMSI, CallSign, Ship Name, Vessel Status + Vessel indicators: Banned and/or SHT. The vessel Flag will also be added. This should be synchronised with the upgrade of the STIRES design. The enrichment procedure will select per requested vessel: The ExpectedCallOfSelectedShip ShipCall with ETAToPortOfCall closer to the request timestamp with the indication of Waste and Security. The MostRecentArrivalOfSelectedShip ShipCall ATAPortOfCall closer to the request timestamp with the indication of Waste and Security. The latest Incident Reportnotification received by EIS.

Business process specifics	
Validation Rules	<ul style="list-style-type: none">• Vessels must be identified by IMO Number and/or MMSI Number.• If the vessel is identified in EIS OSD then the vessel particulars and any notification that exists for which STIRES have access right to are provided in return.• If the vessel is not identified in EIS OSD then a NULL record is provided in return.
Database Transactions	<ul style="list-style-type: none">• A vessel is resolved against the EIS OSD.• Vessel particulars and relative relevant notifications are selected from the EIS notification specific tables.

5 Deployment view

5.1 Design Decisions

Since central SSN applications should support Ship MRS notification message formats for a transition period, this section describes certain design decisions to accommodate this need.

1. SSN-EIS applications (ssn-xmlprotocol-app, ssn-core-app, ssn-console).
2. Deployment diagrams below have been updated accordingly to depict changes introduced due to new application.

5.2 SSN EIS

The SSN EIS deployment view is shown in Figure 5-1

It should be noted that

- SSN EIS topology is not changed. Two JEE application server are used for the deployment of the SSN artifacts; the first for the HTML interface (web consoles) and the second the XML (SOAP) and EJB interfaces.
- IdM system that provides the SSN SSO authentication services is out the scope of this document.
-
- EMSA Portals are out the scope of this document; the portals server included in the deployment view shows the communication path with EIS Business Services.

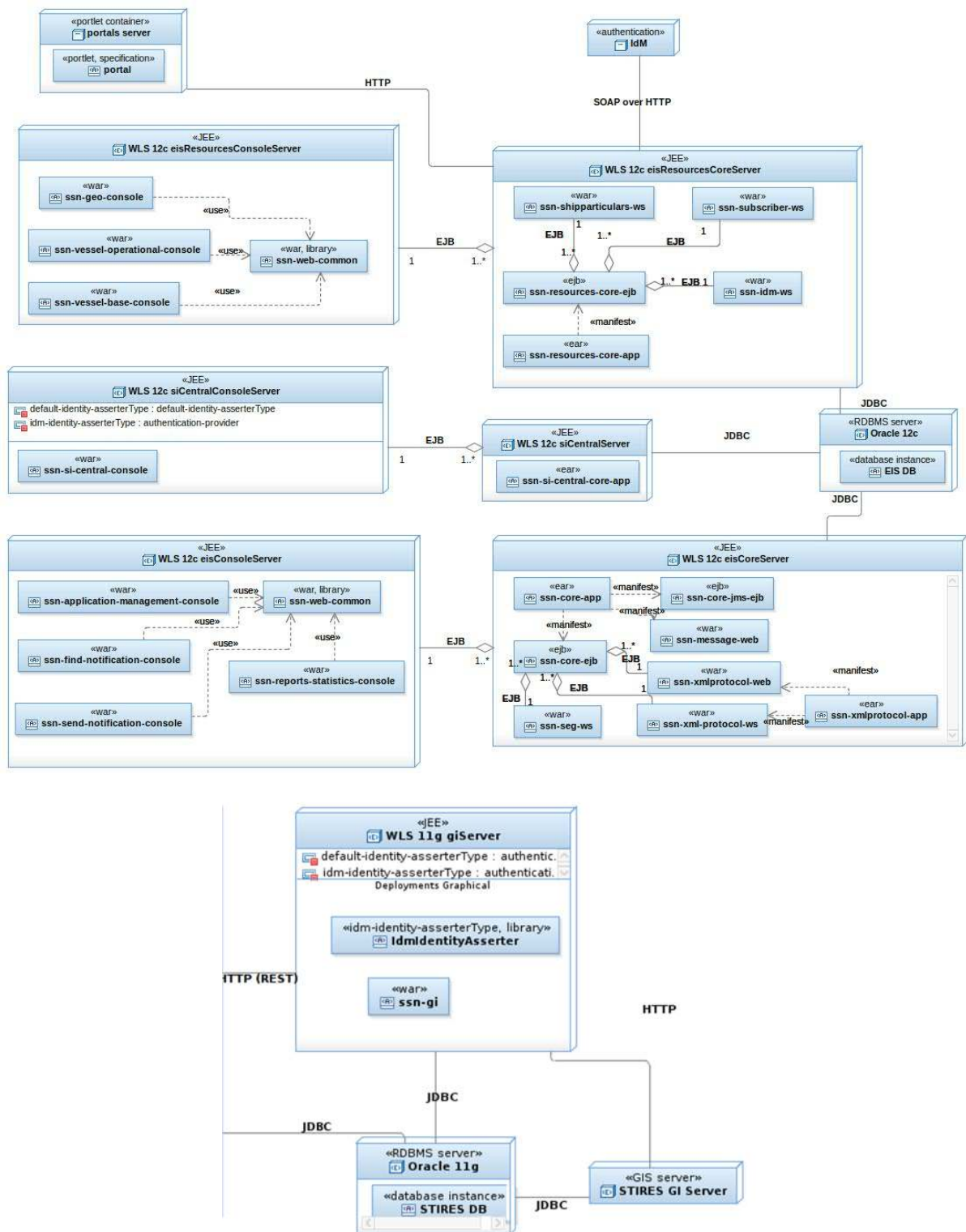


Figure 5-1 SSN-EIS Deployment model

5.2.1 EIS Console Server

The application server hosts the EIS consoles artifacts shall be a host running Red Hat Enterprise Linux 5.3 or later.

Technical Platform: EIS console artifacts shall be deployed onto a JEE application server (Oracle WebLogic Server 12c). The clustering feature (active/active) may be enabled.

Deployment artifacts that compose the SSN EIS web applications at runtime:

- `ssn-web-common`: war provided as library and used by all the following web console applications;
- `ssn-application-management-console`: war deployed to provide SSN EIS management console (SSN MC);
- `ssn-send-notification-console`: war deployed to provide send notifications functionality (SSN TI);
- `ssn-find-notification-console`: war deployed to provide find notifications functionality (SSN TI);
- `ssn-reports-statistics-console`: war deployed to provide SSN EIS reporting and statistics functionality.

Java Version: JDK 1.8 (1.8.0_92) required.

The Deployment war artifacts (`ssn-application-management-console`, `ssn-send-notification-console`, `ssn-find-notification-console` and `ssn-reports-statistics-console`) will communicate with

EIS Core server via EJB

5.2.2 EIS Core Server

The application server hosts EIS core artifacts shall be a host running Red Hat Enterprise Linux 5.3 or later.

Technical Platform: Authority artifacts shall be deployed onto a JEE application server (Oracle WebLogic Server 12c). The clustering feature (active/active) may be enabled.

Transaction: The SSN system is transactional, leveraging the technical platform capabilities.

Persistence: Data persistence will be addressed using the Oracle RDBMS (version 12c) relational database that stores all data related to SSN.

Deployment artifacts that compose the SSN EIS exposed business services at runtime:

- `ssn-core-app`: ear bundles EIS core modules; it consists of
 - `ssn-core`: jar used by all the following modules; it actually implements the EIS Business Services;
 - `ssn-core-ejb`: jar deployed to expose EIS Business Services to web consoles and xmlprotocol applications as EJB (Stateless Session Beans);
 - `ssn-core-jms-ejb`: jar deployed to handle the JMS messages (Message-Driven Beans);
 - `ssn-message-web`: war deployed to expose EIS Message Services as RESTful – XML over HTTP.
- `ssn-xmlprotocol-app`: ear bundles EIS XML protocol applications for SSN schema; it consists of
 - `ssn-xmlprotocol-web`: war exposes the service for SSN EIS XML messages via HTTP(S);
 - `ssn-xmlprotocol-ws`: war exposes the service for SSN EIS SOAP messages via HTTP(S).
- `ssn-seg-ws`: war provides
 - SEG query Service; SOAP messages via HTTP(S);
 - Enrichment query Service; REST Services with XML messages payload via HTTP(S)

Member states applications should be configured to send Ship Notifications via XML to different URLs, depending on the version of protocol they implement. SSN-EIS will accept post requests for

- a. messages at <SSN_HOST>/ssn-xmlprotocol-web /ssn.do

Java Version: JDK 1.8 (1.8.92) required.

The Deployment artifact ssn-core.war will communicate with
EIS DB via JDBC

5.2.3 EIS Resources Console Server

The application server hosts EIS Resources consoles artifacts shall be a host running Red Hat Enterprise Linux 5.3 or later.

Technical Platform: EIS Resources console artifacts shall be deployed onto a JEE application server (Oracle WebLogic Server 12c). The clustering feature (active/active) may be enabled.

Deployment artifacts that compose the web applications at runtime:

- ssn-web-common: war provided as library and used by all the following web console applications;
- ssn-vessel-operational-console: war deployed to provide OSD functionality;
- ssn-vessel-base-console: war deployed to provide CSD functionality;
- ssn-geo-console: war deployed to provide temporary Locations management for Operational registry;

Java Version: JDK 1.8 (1.8.0_92) required.

The Deployment war artifacts (ssn-vessel-operational-console, ssn-vessel-base-console, and ssn-geo-console) will communicate with

EIS Resources Core server via EJB

5.2.4 EIS Resources Core Server

The application server hosts EIS Resources core artifacts shall be a host running Red Hat Enterprise Linux 5.3 or later.

Technical Platform: Authority artifacts shall be deployed onto a JEE application server (Oracle WebLogic Server 12c). The clustering feature (active/active) may be enabled.

Transaction: The SSN system is transactional, leveraging the technical platform capabilities.

Persistence: Data persistence will be addressed using the Oracle RDBMS (version 12c) relational database that stores all data related to SSN.

Deployment artifacts that exposed business services at runtime:

- ssn-resources-core-app: ear deployed to exposes the EIS Business Services as EJB services;
- ssn-resources-core-ejb: jar deployed to expose EIS User, Location, Vessel Business Services to web consoles, ssn-idm-ws and and ssn-shipparticulars-ws as EJB (Stateless Session Beans);
- ssn-shipparticulars-ws: war deployed to expose EIS Vessels Business Services as Web Service – SOAP over HTTP;
- ssn-idm-ws: war deployed to expose the IDM Web Services as Web Service – SOAP over HTTP.
- ssn-subscriber-ws: war deployed to expose the endpoints for announcements of updates in CCD, CLD, COD.

Java Version: JDK 1.8 (1.8.0_92) required.

The Deployment artifact ssn-core.war will communicate with

- EIS DB via JDBC;
- IdM system via SOAP over HTTP for user management.

5.2.5 SI Central Console Server

The application server hosts SI Central console's artifacts is a host running Red Hat Enterprise Linux 5.3 or later.

Technical Platform: SI Central console artifacts are deployed onto a JEE application server (Oracle WebLogic Server 12c).

Deployment artifacts that compose the web applications at runtime:

- `idm-asserter`: jar used for the Identity Assertion Provider Configuration;
- `ssn-si-central-console`: war deployed to provide SSNSI Proxies (Originators and Recipients) management functionality.

Java Version: JDK 1.8 (1.8.0_92) required.

The Deployment war artifact (`ssn-si-central-console`) will communicate with SI Central Core server via EJB.

5.2.6 SI Central Core Server

The application server hosts SI Central core artifact is a host running Red Hat Enterprise Linux 5.3 or later.

Technical Platform: SI Central core artifact is deployed onto a JEE application server (Oracle WebLogic Server 12c).

Transaction: The SSN system is transactional, leveraging the technical platform capabilities.

Persistence: Data persistence will be addressed using the Oracle RDBMS (version 12c) relational database that stores all data related to SSN.

Deployment artifacts that exposed business services at runtime:

- `ssn-si-central-core-app`: ear deployed to exposes the SI Central Business Services as EJB services.

5.2.7 SSN GI - DEPRECATED

The application server hosts `ssn-gi` artifact shall be a host running Red Hat Enterprise Linux 5.3 or later.

Technical Platform: SSN GI artifact shall be deployed onto a JEE application server (Oracle WebLogic Server 12c). The clustering feature (active/active) may be enabled.

Deployment artifacts that compose the SSN EIS web applications at runtime:

- `idm-asserter`: jar used for the Identity Assertion Provider Configuration;
- `ssn-gi`: war deployed to provide GI functionality.

Java Version: JDK 1.7 (1.7.55) required.

The Deployment war artifact (`ssn-gi`) will communicate with

- EIS Core server via HTTP (RESTfull services).
- STIRES Database Instance via JDBC.

5.3 SSN-IMDaTe

The SSN IMDaTe deployment view is presented in Figure 5-2.

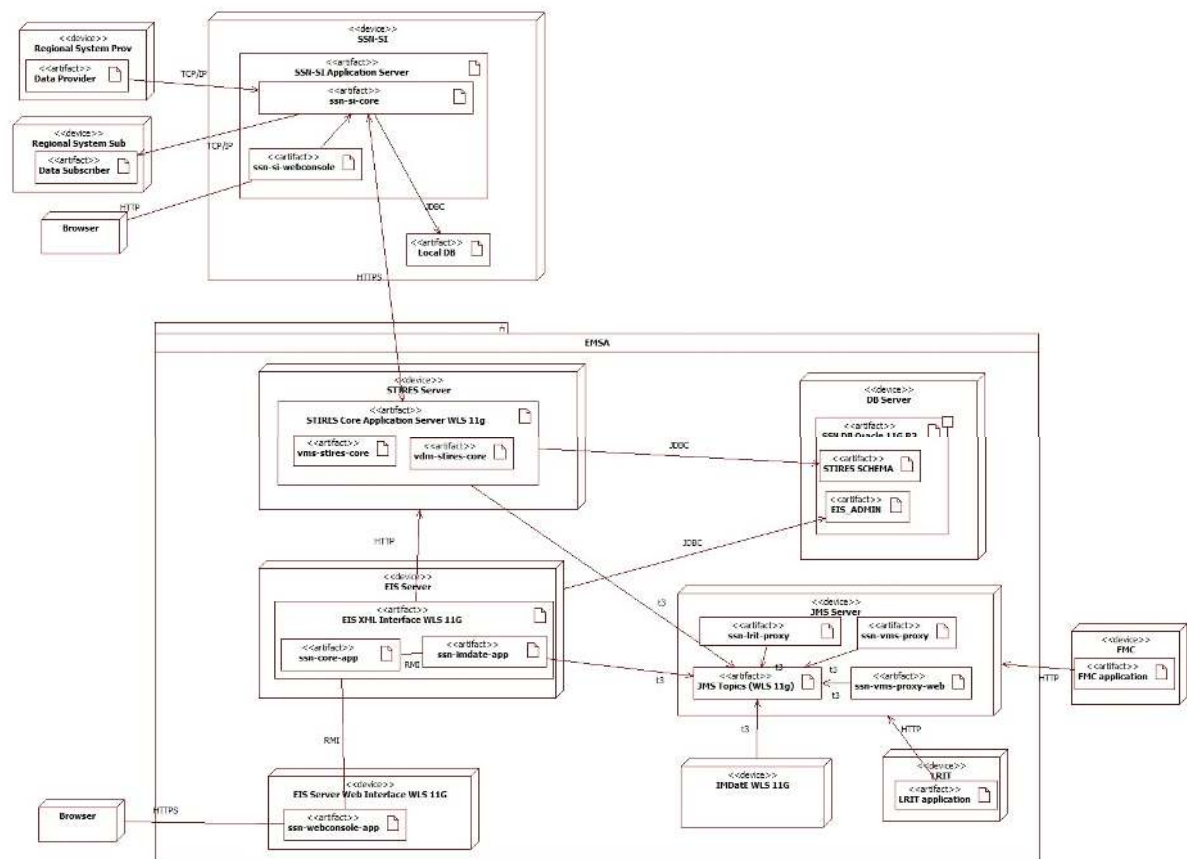


Figure 5-2 IMDatE Deployment view

SSN IMDatE is a distributed system and different components will be installed in different computer nodes. The SSN IMDatE topology comprises of the following machines:

- The Application servers are Redhat Linux machines running Enterprise Edition v. 5.0 operating system. The "Oracle WebLogic application server" is installed on these machines. The EIS, STIRES and IMDatE Proxies applications are deployed on this server.
- The Database server is a Redhat Linux machine running Enterprise Edition v. 5 operating system where the ORACLE RDBMS version 11g R2 is installed and it provides the data storage. The ORACLE RDBMS stores the SSN Database.

5.4 SSN-VMS

The VMS deployment view is presented in Figure 5-2.

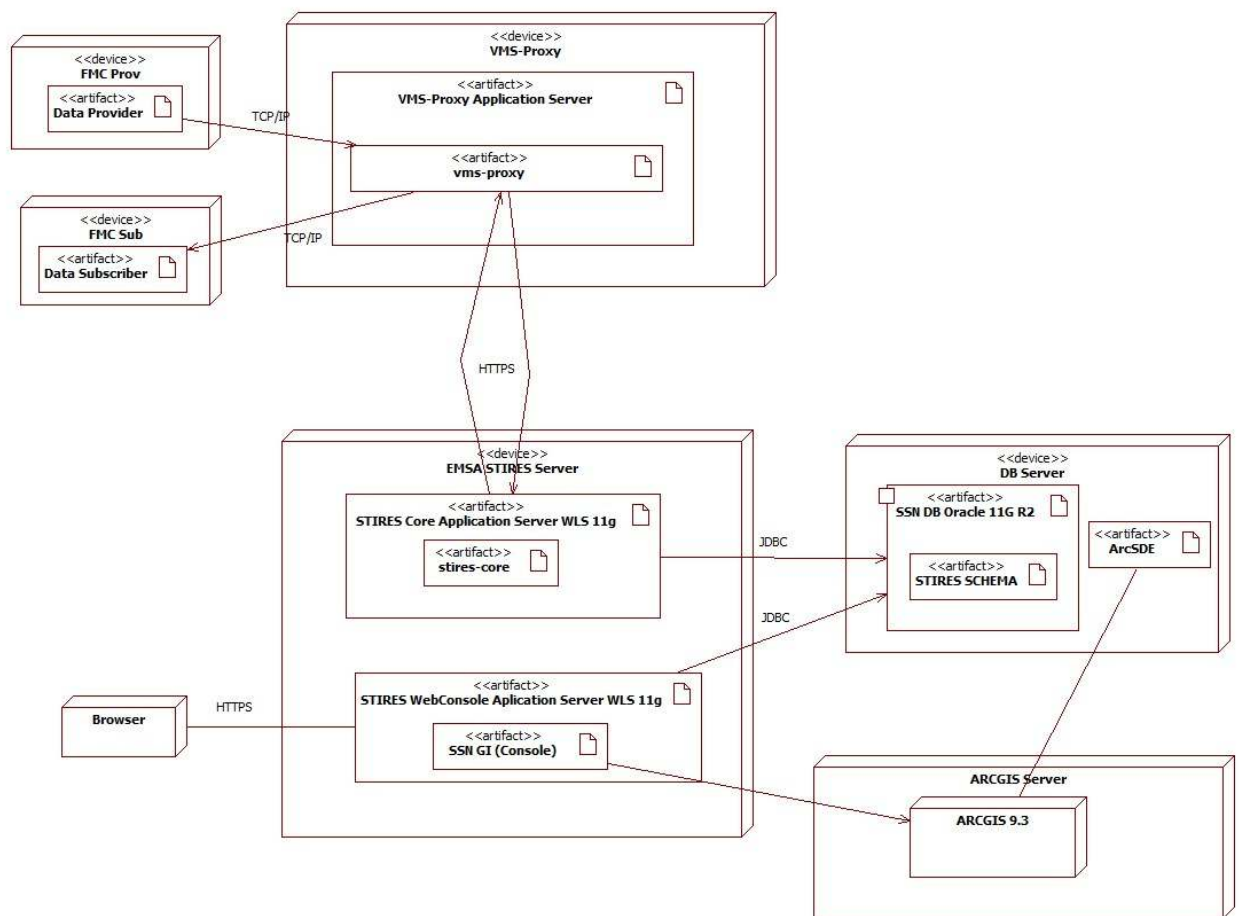


Figure 5-3 VMS Deployment view

The VMS system artifacts shall be deployed onto two servers (devices) at EMSA:

- VMS-Proxy to communicate with FMCs applications;
- STIRES.

The VMS-Proxy is available for the following Operating Systems:

- Microsoft Windows XP, Server 2003.
- Red Hat Enterprise Linux 5, Centos Linux 5, Ubuntu Linux (8.04 & 9.04).

The minimum Hardware requirements are listed in the following table:

Item	Description
Processor	2.66 GHz Dual core
RAM memory	1 GB
Storage	80 GB
Network card	Gigabit network adapter
Other	Video Card, monitor, keyboard and mouse, USB ports

Table 5-1 VMS-Proxy minimum Hardware requirements.

Technical Platform: VMS-Proxyartifact shall be deployed onto a J2EE application server (Apache Tomcat 7.0.x – 7.0.16). The clustering feature (active/active) may be enabled.

Transaction: The VMS-Proxy system is transactional, leveraging the technical platform capabilities.

Deployment artifact that implements the VMS-Proxy application at runtime: vms-proxy-app.ear.

Java Version: JDK 1.6 (1.6.26) required.

The Deployment artifact vms-proxy-app.ear will communicate

- with FMCs (Data Provider and Subscribers) via HTTP(S).
- with STIRES via HTTPS.

Annex A: Business Rules

Voyage Status Indicators

The following table defines the rules based on which a voyage is perceived to be at a given status at a given point in time (e.g. of the notification processing).

Status indicator	Rules
Voyage on going	<ul style="list-style-type: none"> A. The ATDlastPort (derived from ATD portofCallPreviousVoyage) or (in case of ATDlastPort absence) the calcATDlastPort or (in case of absence of ATDlastPort or calcATDlastPort) the ETD last port is in the "past" with respect to the query timestamp in UTC, and B. the [ETAPortofCall + [a configurable parameter, e.g. 2 hours]] is in the future with respect to query timestamp in UTC, and C. there is no ATA known (or in case of ATA absence) no calcATA known for the voyage.
Vessel at port	<ul style="list-style-type: none"> A. The ATDlastPort or (in case of ATDlastPort absence) the calcATDlastPort or (in case of absence of ATDlastPort or calcATDlastPort) the ETD last port) is in the "past" with respect to the query timestamp in UTC, and B. The ATA or (in case of absence of ATA) the calcATA is available for the voyage, and C. There is no ATD or calcATD known for the voyage.
Voyage closed	<ul style="list-style-type: none"> A. The ATDlastPort or (in case of ATDlastPort absence) the calcATDlastPort or (in case of absence of ATDlastPort or calcATDlastPort) the ETD last port) is in the "past" with respect to the query timestamp in UTC. and B. The ATD or (in case of absence of ATD) the calcATD is available for the voyage. <p>The voyage is assigned the status "Closed" which indicates that the voyage is completed.</p> <p>Closed voyages are provided in requests but are not considered in voyage consolidation.</p>
Future (known) voyage	Any voyage that ETDlastPort declared in the notification and set in a future time with respect to the timestamp of the query and with an ETAPortOfCall also in the future. The ETDlastPort should not be dummy (that is it must be within the limit constraint by the WVD).
Voyage cancelled	Receipt of ZZCAN for the port call reported either via PortPlus or via Port notification.
Dummy voyage	<p>Dummy voyages report an ETA to a given Port in the past; no ATA has been reported for the given Port while a later ATA is reported by a MS or detected by STIRES to another Port.</p> <p>Dummy voyages are also considered in voyage consolidation.</p>

Voyage retrieval specific rules

The following table defines the rules of voyage retrieval that are used during the "matching" process. They are used to decide if there is a voyage in the database whose data could be correlated with the data reported in the incoming notification.

Data Provider	Notification includes	Voyage status	Which voyage has to be retrieved from the EIS database for the reported vessel (if exists) and action to be taken in case the Notification does not report the ShipCallId.
MS	ATD	Voyage closed	<p>Fetch the most recently "closed" status voyage from the database, if exists, and initiate the voyage matching process.</p> <p>If no match is to be found fetch the voyage with status "at port", if exists and try to resolve the notification with it. If no match found fetch the most recently created voyage" with status unknown and try to match. If not, a match found create a new voyage in the database with the data in the notification.</p>
STIRES-STAR	calcATD	Voyage closed	<p>Fetch the most recently "closed" (ATD is reported) status voyage for the ship from the database, if exists, and check if the port of call info match with the port of call reported by STIRES-STAR. If yes register in the STIRES voyage linked with the incoming STIRES-STAR notification the voyage ID of the voyage fetched from the database (should this VoyageID is not already registered in the STIRES voyage record or should in the STIRES voyage record was registered a different voyageID).</p> <p>If a match is not found, fetch from the database the voyage of the ship with status "at port" (ATA reported, no ATD reported) if exists and repeat the matching process. If a match is found register in the STIRES voyage linked with the incoming STIRES notification the voyage ID of the voyage fetched from the database (should this VoyageID is not already registered in the STIRES voyage record or should in the STIRES voyage record was registered a different voyageID).</p> <p>If no match is found the VoyageID already registered in the STIRES voyage shall remain un-altered). The value of the VoyageID could be NULL.</p>
STIRES-STAR	calcETDLastPort	Voyage on going	<p>Fetch the voyage with status "on going" from the database, if exists. If the LastPort data match with those in the STIRES-STAR notification (based on the rules for matching last port) assign to the STIRES voyage that is related with the notification received the VoyageID of the voyage fetched from the database. If a match is not found the VoyageID in the STIRES voyage that is related to the STIRES notification shall remain empty.</p>

Data Provider	Notification includes	Voyage status	Which voyage has to be retrieved from the EIS database for the reported vessel (if exists) and action to be taken in case the Notification does not report the ShipCallId.
MS	ATA	Vessel at port	First retrieve the voyage with status "at port" in the database for the vessel, if exists. If no match is found fetch the voyage with status "on-going" and start the matching process. If no match is found fetch the most recent voyage with status unknown for the ship and start the matching process. If no match is found create a new voyage based on the data in the notification.
STIRES-STAR	calcATA	Vessel at port	<p>First retrieve the voyage with status "on going" in the database for the vessel, if exists.</p> <p>If a match is found (PortofCall in the voyage fetched from the database equals with the calculated port of call from STIRES-STAR register in the STIRES voyage linked with the incoming STIRES notification the voyage ID of the voyage fetched from the database (should this VoyageID is not already registered in the STIRES voyage record or should in the STIRES voyage record was registered a different voyageID).</p> <p>If a match is not found fetch the voyage with status "at port" (ATA reported, no ATD reported) and start the matching process). If match is found update the voyage ID in the STIRES voyage. If no match is found fetch the most recent voyage with status unknown for the ship, if exists, and start the matching process (based on LastPort and ETDLastPort??). If match is found update the voyageID in the STIRES voyage record. If no match found the voyage ID in the STIRES record shall remain un-altered.</p>
MS	ETA Notification reports an on-going voyage (according to the on-going voyage definition)		First retrieve the voyage with status "on-going" in the database for the vessel, if exists. If no match is found fetch the voyage with status "at port", if exists, and start the matching process. If no match is found fetch the most recent voyage with status unknown for the ship and start the matching process. If no match is found create a new voyage.
MS	ETA Notification reports a future voyage (according to the future or plan voyage definition)		Check if there exists a future voyage in the database where the ETAPortOfCall is closest in the future with respect to the ETDLastPort reported in the notification and start the matching process. If no match found create a voyage.

Data Provider	Notification includes	Voyage status	Which voyage has to be retrieved from the EIS database for the reported vessel (if exists) and action to be taken in case the Notification does not report the ShipCallId.
MS	ETA Notification reports an unknown voyage with ETA in the future		Fetch the voyage with status "on-going" from the database, if exists, and try to resolve. If no match found fetch If no match found fetch the voyage meeting the following conditions: <ul style="list-style-type: none"> a. with ETAPortOf Call "in the past" in relation to the query timestamp) for up to a configurable period (period ETAPortOfCall [(configurable, e.g. 2hours), b. (If there no voyage fulfilling the condition (a) above, the voyage with the "closest" ETAPortOfCall _in the future with respect to query timestamp. If such a voyage exists, try to merge the notification to it. If a match is not found create a new voyage in the database.
MS	Dummy voyage with ETAToPortOfCall in the past		Retrieve the most recently created voyage with dummy status from the same data provider, if exists, and set it to dummy status.
MS	ZZCAN reported with a Portplus notification	Voyage cancelled	Change of destination. Fetch the voyage with the same ShipcallID and cancel it. <u>Be aware that the Hazmat linked to the voyage is not cancelled and should be linked with another voyage in the database.</u> That is why, following the cancellation of the ship call the system should check again the current status of voyage as reported by the notification that initially included the Hazmat information (you can retrieve this notification from the database using the Hazmat ShipcallID or the Hazmat data provider stored in the cancelled voyage record). Re-initiate the process of voyage retrieval. Based on the status of the voyage as reported in the notification and based on the processes highlighted above for retrieving a voyage from the database identify the voyage where the Hazmat info has to be linked.

Data Provider	Notification includes	Voyage status	Which voyage has to be retrieved from the EIS database for the reported vessel (if exists) and action to be taken in case the Notification does not report the ShipCallId.
MS	An update of a PortPlus message reporting Hazmat EU departure where the data provider changed the HazmatYesNo attribute value from "Yes" to No (by doing this essentially the user "cancels" the previously sent Hazmat notification		<p>Retrieve the voyage with the Hazmat ShipcallID pointing to the notification that included the changed HazmatYesNo attribute change and update the voyage. In case the voyage record does not include a ShipCallID (that is it was created by the next portdata and it is not yet confirmed by the port of arrival as yet, "cancel" it completely.</p> <p>Actually here the MS ask us to ignore a Hazmat EU notification previously sent with a PortPlus message – Should for the voyage next Port has been already received a Portplus from the Port of arrival the voyage cannot be cancelled.</p> <p>This is only way for a MS to report that to cancel a Hazmat attached to a Portplus message reporting EU departure.</p>

Voyage correlation further rules

Determine the conditions of correlating voyage information from different data providers	
Description:	<p>SSN will only attempt to correlate ship call information from different providers into a single voyage only in the case that at the moment of registration of a new notification in the system there exist another notification for the same ship providing an ETA in the future. Correlation cannot be considered definitive if done. SSN shall attempt to correlate notifications whenever a new Port Plus notification will arrive given that new or updated operational information provided in the new notification might change the conditions on which the correlation is based.</p> <p>The correlation process is successful (that is ship call information provided by different providers is merged to a single voyage) in the case the SSN central application can safely determine that:</p> <ul style="list-style-type: none"> in the notifications provided by different providers the information related to departure and arrival ports locations match to each other, or at least in the notifications provided by different providers the location of the port of departure as declared by the port of destination is in the same country with the departure location declared in the port of departure notification. <p>In case the notifications provided by different providers the information related to departure ports locations match to each other but the PortOfCall do not match then the vessel has most probably changed destination. In such case the 2 notifications are not correlated, instead distinct voyages are created.</p> <p>Notes:</p>

	<ul style="list-style-type: none"> a) A calculated Voyage – reported by a PortPlus from STIRES – can indicate the correct PortOfCall; this is an indication of a change of destination. b) For consistency the other voyage (prior to the destination change) will appear as unknown in a ship call request.
EU departure - determines when the Hazmat is active	
Description:	<p>The active Hazmat rules are considered to determine if the Hazmat relates with the voyage; in other terms is “active” during that voyage. These rules are applicable in case of a request for active Hazmat as well as during the processing of a new PortPlus notification to identify if a Hazmat are active during the voyage defined by the PortPlus notification.</p> <p>No calculated values (ATA/ATD) are to be considered.</p> <p>Active Hazmat (EU departure): A Hazmat EU departure is considered “active” from the ATD provided by the departing port:</p> <ul style="list-style-type: none"> A. Until an ATA Port of Call notification will be received for the ship “in the future” with respect to the ATD (ETD) from the port of departure, or B. Until a new Hazmat declaration for the ship will become active, or C. Until the period is elapsed: <ul style="list-style-type: none"> I. Until the period [ATD (ETD) from departure port+EVD] is elapsed if vessel is heading towards a European destination II. Until the period [ATD (ETD) from departure port+WVD] is elapsed if vessel is heading towards a non-EU port or unknown destination.
Non-EU departure - determines when the Hazmat is active	
Description:	<p>Active Hazmat (Non-EU Departure): A Hazmat Non-EU departure is considered “active” for a period:</p> <ul style="list-style-type: none"> A. From ETD port of departure (if available) until ATA (ETA) port of Call, or B. From its registration (defined by the SentAt) to the system and until the ATA (or in case of non-availability of ATA, the ETA Port of Call). Conditions are: <ul style="list-style-type: none"> I. [ATA (ETA) port of call] - SentAt timestamp] <= WVD (proposed 30 days) II. In case this condition is not met the notification is active for a maximum period defined by [ATA (ETA) port of call] – WVD (planned 30 days)] C. Ship is not departed, or Hazmat is not registered. The later although self-evident is defined for completeness.
EU departure - determines which voyage to associate a Hazmat declaration with	
Description:	<ul style="list-style-type: none"> A. <i>Unknown NextPortOfCall:</i> <ul style="list-style-type: none"> I. If notifications sent by the departure port concerning EU departures quote an unknown destination and the Hazmat declaration associated to the notification becomes active, SSN shall link the Hazmat declaration associated to the notification to the ship voyage with the closest ETA in the future. II. If for a ship exists, the following co-exist in the system: <ul style="list-style-type: none"> a. An “active” Hazmat EU departure declaration with ZZUKN quoted as destination b. An active Hazmat Non-EU departure declaration provided by the port of Call with no information on the last port (no ETD from departure port) <p>SSN will consider the “unknown” destination of the ship, declared in the EU departure notification actually as a non-EU port (provided that no further ship calls are found in the system with an ETA in the future quoting as last port</p>

	<p>the sender of the above notification.). In such a case the system will consider that the date/ time the Hazmat EU declaration stops to be active will be the date data that the Hazmat Non-EU declaration will become active (SentAt of the Hazmat Non-EU Departure).</p> <p>B. <i>NextPortOfCall is known and reported:</i></p> <p>If notifications sent by the departure port concerning EU departures quote as prospective port of call a known destination, and</p> <p>the Hazmat declaration associated to the notification becomes active,</p> <p>SSN shall link the Hazmat declaration to the ship voyage with the closest ETA in the future (with respect to the ATD, or in case of ATD absence, the ETD from port of call). For the correlation will be considered the notification from arrival port which either:</p> <ol style="list-style-type: none"> Identifies as last port, the port that provided the Hazmat EU departure notification or, Does not include any information on last port. <p>If the port of call (in the ship call notification with nearest ETA in the future) differs from the one identified within the notification provided by the departure port, preference is given to the port of call information as defined in the notification sent by the Port of Call. That is, in such a case the next port of arrival information provided by the port of departure is ignored.</p>
EU departure - adjust Hazmat active period	
Description:	<p>EU departure - adjust Hazmat active period</p> <p>For a given ship the following exist in the system:</p> <ol style="list-style-type: none"> A Hazmat EU departure destination towards non-EU country ; A Hazmat Non-EU departure with last port = non-EU country. <p>More specifically:</p> <p>A. If for a ship the following exist in the system:</p> <ol style="list-style-type: none"> A Hazmat EU departure destination towards no- EU country A Hazmat Non-EU departure with last port = non-EU country <p>and their "active" period is "overlapping" the end-active date for Hazmat EU departure declaration and start-active date for Hazmat non-EU declaration the following adjustment will be made:</p> <ol style="list-style-type: none"> If the Hazmat EU departure notification provides a "not dummy" ETA to destination port and the Hazmat non-EU departure notification provides a "not dummy" ETD from the Non-EU port, the ETA to destination port is ignored and system will consider: $[EndActiveDateTime \text{ for Hazmat EU departure}] = [ETD \text{ from Non-EU port declared in Hazmat Non-EU departure notification}] = [StartActiveDateTime \text{ for Hazmat Non-EU departure notification}]$ If the Hazmat EU departure notification provides a "not dummy" ETA to destination port and there is no ETD from the Non-EU port declared in the Hazmat non EU departure notification the system will consider: $[EndActiveDateTime \text{ for Hazmat EU departure}] = [ETA \text{ to destination declared in Hazmat EU notification}] = [StartActiveDateTime \text{ for Hazmat Non-EU departure notification}]$

	<p>c. If both estimated times are missing or are considered dummies the system will consider:</p> <p><i>[EndActiveDateTime for Hazmat EU departure] = [SentAt of Hazmat non-EU notification] = [StartActiveDateTime for Hazmat Non-EU departure notification]</i></p> <p>B. If for a ship exists, the following co-exist in the system:</p> <p>I. A Hazmat EU departure destination towards non-EU country where the ETA to destination (ETA1) is provided and is not dummy</p> <p>II. A Hazmat Non-EU departure with last port = non-EU country where the ETD from departure port (ETD1) is provide and it is not dummy</p> <p>III. There is a logical relationship between ETA1 and ETD1 (ETA1<ETD1)</p> <p>Then the active period for the Hazmat EU departure notification and Hazmat non-EU departure notification will be set as follows.</p> <p>a. Active period Hazmat EU departure notification :</p> <p><i>From ATD (or in case of ATA absence the ETD) from port of departure to ETA1;</i></p> <p>b. Active period Hazmat non EU departure notification :</p> <p><i>From ETD1 to ATA (in case of absence ETA+2hours) to destination.</i></p>
--	---

STIRES/STAR notification consolidation - Business Processes and voyageID assignment rules for "detected" voyages

The SSNVoyageID shall be used to link events detected by STIRES/STAR (departure from a port and arrival to another port) to ShipCalls reported by the MS. The objective of linking STIRES detected events to shipcalls reported by the MS is twofold:

- A. Detect a change of destination not reported via an MS
- B. Assign the events reported by STIRES/ STAR to the "right" shipcall (this would be the case that SSN receives a new notification with closer ETA in the future with respect to the most recent departure of the vessel from a port.

The following rules apply:

1. There is one2one relationship between a shipcallID and a VoyageID **for the voyages that are reported via the notifications the MS send to SSN.**
2. Within the SSN EIS database the "voyages" that have as origin the STIRES / STAR application are maintained in a distinct manner (this is possible because such voyages abide to the SSN XML protocol). Data recorded in the STIRES/ STAR-generated voyages shall not be included in the EIS textual interface and/ or included as response to MS user requests via the XML/ SOAP interface.
3. In order to link events detected by STIRES/ STAR ~~by STIRES~~ to calls reported by the MS, the voyageID related to the reported (by a MS) a ShipCall X will be inserted as voyage IDs in the voyage(s) that have origin the STIRES/ STAR application and are linked to the ShipCall X. (the way STIRES-generated voyages are linked to ShipCalls is in line with the processes identified in this Appendix)
4. At a given moment and taking into account the eventuality of a ShipCall not being reported by an MS, two or more "Detected" voyages might be linked to the same MS-reported ShipCall (thus exactly the same VoyageID will be assigned to such voyages). By linking more than one detected voyage to the next "known" (from an MS notification) destination, the system can correctly track "deviations" from the planned route.
5. The assignment or change of voyageID for STIRES/ STAR -generated voyages is done upon receipt of a new Portplus notification provided by STIRES/ STAR. In case of change of a

voyageID assigned to a STIRES-generated voyage the procedure, as per rule (6) below should be followed.

6. The receipt of notifications from MS may alter the sequence of voyages/ ShipCalls within the EIS database. That is why, there is always a need to check if the STIRES/ STAR voyages previously linked to a ShipCall e.g. (B) have to be linked to a ShipCall e.g. (A) should the voyage corresponding to Call (A) will be completed before the start of the voyage corresponding to Call (B). To ensure proper linking of STIRES events to shipcalls reported by the MS the following process should be applied:

Let's assume that in a STIRES/ STAR -reported voyage the VoyageID was a **NOT** null value "A1" and – due to detection of a change in the sequence of the calls reported by the MS, is to change and become "B1". In this case there should be a process that will retrieve all STIRES/ STAR-generated voyages previously assigned the Voyageid "A1" and update it to "B1".

Annex B: CSD Specific Business Rules

Business Rule 7

The ship name is to be updated in the CSD when the following conditions have been met:

- The characters included in the names (i.e. those existing in the database and those proposed by the reference source) differ by more than a configurable parameter (e.g. 20%), OR;
- There is a difference in the last three or the first three characters of the name. This check should ignore the inclusion of characters such as: " ", " " (whitespace), - (dash) _ (underscore), (point) within the three being compared.

Business Rule 10

The table below defines the rules for updating vessel attributes for a vessel listed in the CSD. The checks are based on the "date of effect" of the proposed update, plus the "relative" ranking of the confidence level assigned to the "reason to update".

Considering the higher value of confidence being "1" and that the following numbers (2, 3, etc.) have a lower level of confidence and that

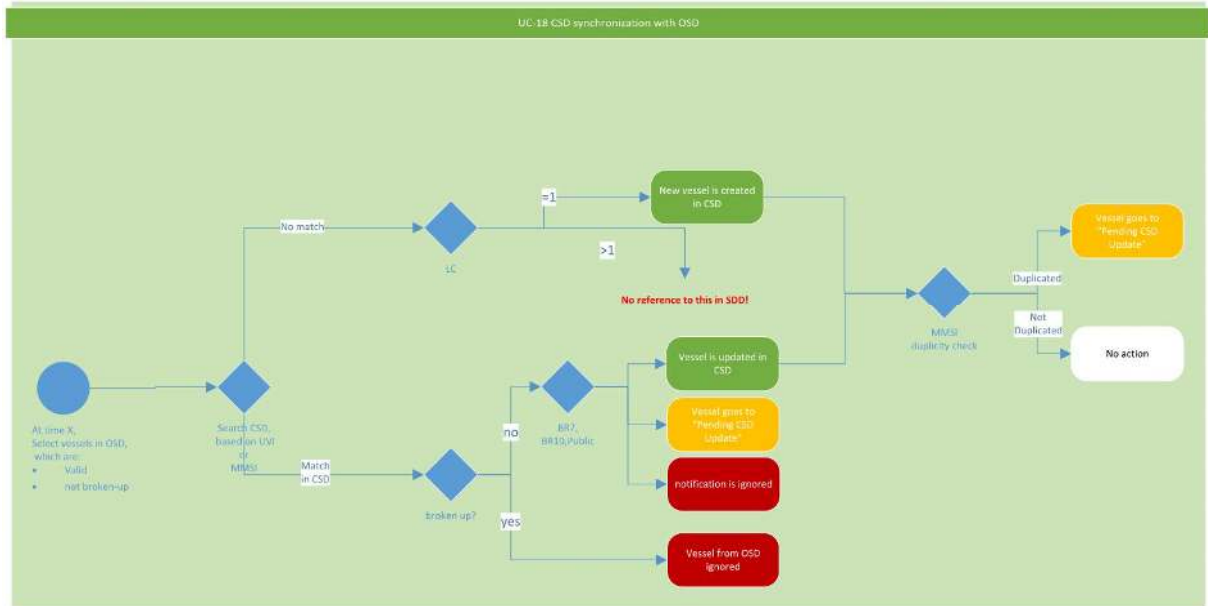
- "N" the value allocated to the level of confidence for a specific vessel attribute stored in the CSD and
- "X" the value of the "level of confidence" of the incoming source,

Then the following apply:

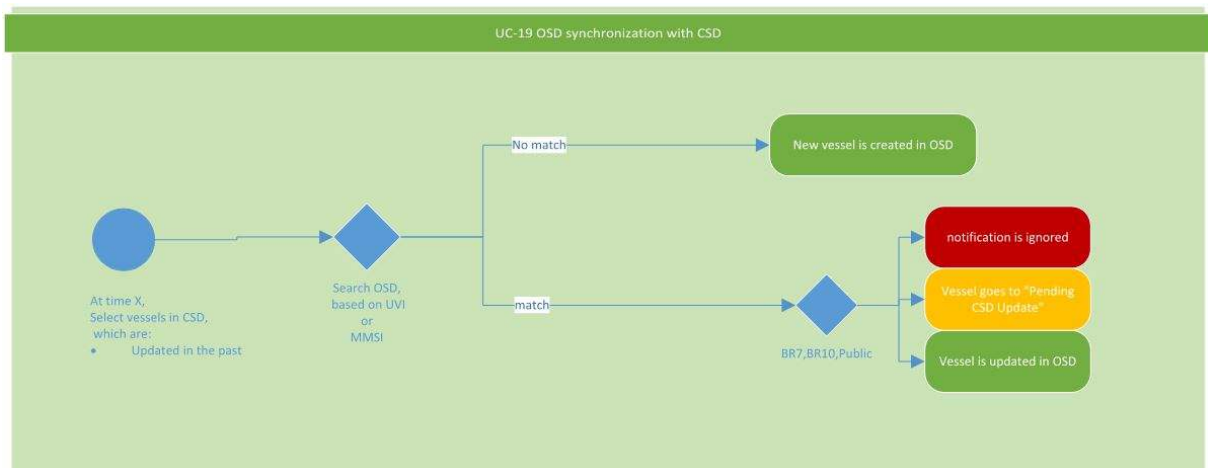
- If [**N>X**] then the relative level of confidence assigned to the incoming update is **HP**;
- If [**N<X**] then the relative level of confidence assigned to the incoming update is **LP**;
- If [**N=X**] then the relative level of confidence assigned to the incoming update is **EQ**.

Relative priority of the update	Rule	Relevant CSD log entries
HP	a. IF [SYSDATETIME > (Date of Effect of the proposed Update) > (Date Effect of Ship particular in the CSD)] THEN UPDATE the vessel attribute in the CSD	UPDATE
	b. IF [SYSDATETIME > (Date Effect of Vessel attribute in the CSD) > (Date Effect of proposed Update)] AND [(SYSDATETIME - Date Effect of proposed Update) <= (A configurable parameter, e.g. 30 days)] THEN initiate "PENDING V&V" processes	PENDING V&V
	c. IF [(SYSDATETIME >= Date Effect of Vessel attribute in the CSD) > Date Effect of proposed Update)] AND [(SYSDATETIME - DateEffectOfproposedUpdate) > (A configurable parameter, e.g. 30 days)] THEN IGNORE	IGNORE

Relative priority of the update	Rule	Relevant CSD log entries
EQ	a. IF [SYSDATETIME> = (Date Effect of proposed Update)> (Date Effect of Vessel attribute in the CSD)] THEN UPDATE the vessel attribute in the CSD	UPDATE
	b. IF [SYSDATETIME >= (Date Effect of Vessel attribute in the CSD) > = (Date Effect of proposed Update)] THEN IGNORE	IGNORE
LP	c. IF [SYSDATETIME >= (Date Effect of proposed Update)> (Date Effect of Vessel attribute in the CSD)] THEN initiate "PENDING V&V" processes	PENDING V&V
	d. IF SYSDATETIME>= (Date Effect of Vessel attribute in the CSD) > (Date Effect of proposed Update)] THEN IGNORE	IGNORE



UC-19 OSD synchronisation with CSD



Annex D: Tasks

The following table defines the list of tasks defined per function provided by SSN EIS application, with the indication for each whether a location and/or source restriction can be applied.

Task	Location Restriction	Source Restriction
SSN-EIS		
PORTPLUS_NOTIFIER	Yes	-
HAZMAT_NOTIFIER	-	-
WASTE_NOTIFIER	-	-
SECURITY_NOTIFIER	-	-
BUNKERS_NOTIFIER	-	-
CREWPAX_NOTIFIER	-	-
SHIP_MRS_NOTIFIER	-	-
SHIP_AIS_NOTIFIER	-	-
EXEMPTIONS_NOTIFIER	Yes	-
ALERT_SITREP_NOTIFIER	-	-
ALERT_POLREP_NOTIFIER	-	-
ALERT_WASTE_NOTIFIER	-	-
ALERT_LFC_NOTIFIER	-	-
ALERT_OTHERS_NOTIFIER	-	-
ALERT_BANNED_NOTIFIER	-	-
ALERT_FAILED_NOTIFIER	-	-
ALERT_INSURANCE_NOTIFIER	-	-
ALERT_PILOT_NOTIFIER	-	-
ALERT_VTS_NOTIFIER	-	-
SHIPPARTICULAR_NOTIFIER	-	-
SHIPCALL_REQUESTOR	Yes	Yes
HAZMAT_REQUESTOR	-	-
WASTE_REQUESTOR	-	-
SECURITY_REQUESTOR	-	-
BUNKERS_REQUESTOR	-	-
CREWPAX_REQUESTOR	-	-
SHIP_MRS_REQUESTOR	Yes	Yes
SHIP_AIS_REQUESTOR	Yes	Yes
EXEMPTIONS_REQUESTOR	Yes	Yes
ALERT_SITREP_REQUESTOR	-	Yes
ALERT_POLREP_REQUESTOR	-	Yes

Task	Location Restriction	Source Restriction
ALERT_WASTE_REQUESTOR	-	Yes
ALERT_LFC_REQUESTOR	-	Yes
ALERT_OTHERS_REQUESTOR	-	Yes
ALERT_BANNED_REQUESTOR	-	Yes
ALERT_FAILED_REQUESTOR	-	Yes
ALERT_INSURANCE_REQUESTOR	-	Yes
ALERT_PILOT_REQUESTOR	-	Yes
ALERT_VTS_REQUESTOR	-	Yes
ALERT_DISTR_ADMIN_REQUESTOR	-	-
SHIPPARTICULARS_REQUESTOR	-	-
SHIPPARTICULARS_SUBSCRIBER	-	-
SAT_REQUESTOR	-	-
ENRICHMENT_REQUESTOR	-	-
INCIDENT_REP_RECIPIENT	-	-
INCIDENT_REP_RECIPIENT_EMAIL	-	-
MRS_MANAGER	-	-
USER_MANAGER	Yes	-
APPLICATION_PAR_MANAGER	-	-
NOTIFICATION_PAR_MANAGER	-	-
BANNER_MANAGER	-	-
EIS_MONITOR_JMSQUEUE	-	-
EIS_MONITOR_USERACTIVITY	-	-
LOGS_SEARCH	-	-
FREE_TEXT_SEARCH	-	-
STATS_REQUESTOR	-	-
STIRES_PORTAL	-	-
STIRES_REPORTS_REPORTS	-	-
STIRES_REPORTS_STATISTICS	-	-
LOCATION_MANAGER	Yes	-
LOCATION_DOWNLOAD	-	-
VESSEL_MANAGER	-	-
BANNED_VESSEL_MANAGER	-	-
BANNED_VESSEL_DOWNLOAD	-	-
SHT_VESSEL_DOWNLOAD	-	-
DETAINED_VESSEL_DOWNLOAD	-	-
CSD_MANAGER	-	-

Task	Location Restriction	Source Restriction
CSD_READER	-	-
CSD_ADMIN_EMSA	-	-
SSN-SI		
PROXIES_MANAGER	-	-
AIS_PROVIDER	-	-
AIS_RECIPIENT	-	-
SSN-VMS		
VOYAGE_NOT_RECIPIENT	-	-
VMS_AIS_RECEIVER	-	-
VMS_NOTIFIER	-	-
SSN-GI (DEPRECATED)		
STIRES_VIEWER_GIS	-	-
STIRES_VIEWER_GIS_ACCESS_7D_WINDOW_DB	-	-
STIRES_VIEWER_GIS_ACCESS_COMMON_BOOKMARKS	-	-
STIRES_VIEWER_GIS_ACCESS_COMMON_FILTERS	-	-
STIRES_VIEWER_GIS_ACCESS_COMMON_WATCHDOG	-	-
STIRES_VIEWER_GIS_ACCESS_PLAYBACK	-	-
STIRES_VIEWER_GIS_CREATE_COMMON_BOOKMARKS	-	-
STIRES_VIEWER_GIS_CREATE_COMMON_FILTER	-	-
STIRES_VIEWER_GIS_CREATE_COMMON_WATCHDOG	-	-
STIRES_VIEWER_GIS_CREATE_COMMON_SHAPE	-	-
STIRES_VIEWER_GIS_CREATE_PROXIES_FILTERS	-	-
STIRES_VIEWER_GIS_CREATE_STATISTICS_OVERLAYS	-	-
STIRES_VIEWER_GIS_VIEW_GEOPICTURE	-	-
STIRES_VIEWER_GIS_VISUALISE_AIS_DATA	-	-
STIRES_VIEWER_GIS_ALLOW_DYNAMIC_VIEW	-	-
STIRES_VIEWER_GIS_VISUALISE_LRIT_DATA	-	-
STIRES_VIEWER_GIS_VISUALISE_VMS_DATA	-	-
STIRES_VIEWER_GIS_ACCESS_STATISTICS_OVERLAYS	-	-
STIRES_VIEWER_GIS_VISUALISE_SAT_AIS_DATA	-	-
STIRES_VIEWER_GIS_ACCESS_COMMON_SHAPE	-	-
STIRES_ADMIN_REFDATA_TREFPORTS_AREA	-	-
STIRES_REPORTS_STATISTICS_CROSS_LMIU_TYPE	-	-
STIRES_REPORTS_STATISTICS_CROSS_PSC_TYPE	-	-
STIRES_ACCIIS_PERMISSION_INPUT_TOOL_LOGIN	-	-
STIRES_ACCIIS_PERMISSION_GI_LOGIN	-	-

Task	Location Restriction	Source Restriction
ACCIIS_PERMISSION_VISUALIZE_SHIP_BUILD_YEAR	-	-
ACCIIS_PERMISSION_VISUALIZE_SHIP_CLASS_SOCIETY	-	-
ACCIIS_PERMISSION_VISUALIZE_SHIP_FLAG	-	-
ACCIIS_PERMISSION_VISUALIZE_SHIP_MANAGER	-	-
ACCIIS_PERMISSION_VISUALIZE_SHIP_OWNER	-	-
ACCIIS_PERMISSION_VISUALIZE_SHIP_TONNAGE	-	-
ACCIIS_PERMISSION_VISUALIZE_SHIP_TYPE_LMIU	-	-
ACCIIS_PERMISSION_VISUALIZE_SHIP_TYPE_PSC	-	-