EUROCONTROL Specification for Airspace Management (ASM) Support System Requirements supporting the ASM processes at local and FAB level
Part II – ASM Systems Interface Requirements

Edition: 1.1
Edition date: 21/03/2024
Reference nr: EUROCONTROL-SPEC-179
ISBN: 978-2-87497-105-1

SUPPORTING EUROPEAN AVIATION

EUROCONTROL Specification for Airspace Management (ASM) Support System Requirements supporting the ASM processes at local and FAB level - Part II - ASM Systems Interface Requirements

DOCUMENT IDENTIFIER : EUROCONTROL-SPEC-179

Edition Number	**:**	**1.1**
Edition Date	**:**	**21/03/2024**
Status	**:**	**Released issue**
Intended for	**:**	**General Public**
Category	**:**	**EUROCONTROL Specification**

DOCUMENT CHARACTERISTICS

TITLE		
EUROCONTROL Specification for Airspace Management (ASM) Support System Requirements supporting the ASM processes at local and FAB level Part II – ASM Systems Interface Requirements		
	Publications Reference:	SPEC-179
	ISBN Number:	978-2-87497-105-1
Document Identifier	**Edition Number:**	1.1
EUROCONTROL-SPEC-179	**Edition Date:**	21/03/2024

Abstract

This Part II document covers system interface requirements for ASM Support System supporting the ASM processes at local and FAB level. It defines the service definition of the ASM service.

Conformance with these requirements, despite the differences in the systems detailed specifications, will ensure interoperability and will facilitate the interface between corresponding stakeholder ASM systems and External Users.

Keywords

ASM/ATFCM/ATC	Process	System support	Automation
Requirements	Interface	Service	Information Definition
Messages			

Contact Person(s)	Email
Pavlin BELICHOVSKY Sasho NESHEVSKI	standardisation@eurocontrol.int

STATUS, AUDIENCE AND ACCESSIBILITY					
Status		**Intended for**		**Accessible via**	
Working Draft	☐	General Public	☒	Intranet	☒
Draft	☐	EUROCONTROL	☐	Extranet	☐
Proposed Issue	☐	Restricted	☐	Internet (www.eurocontrol.int)	☐
Released Issue	☒				

DOCUMENT APPROVAL

AUTHORITY	NAME AND SIGNATURE	DATE
Director General	001 Raúl MEDINA CABALLERO	21 MAR 2024

DOCUMENT CHANGE RECORD

The following table records the complete history of the successive editions of the present document.

EDITION NUMBER	EDITION DATE	REASON FOR CHANGE	PAGES AFFECTED
1.0	13/01/2020	Update following formal stakeholder consultation	All
1.0.1	11/07/2022	Updated draft following internal review	All
1.0.2	16/01/2023	Updated draft following internal review	All
1.1	21/03/2024	Updated draft for formal stakeholder consultation Updates primarily to support implementation of CP1 Regulation requirements.	
		- Modification of title: “ASM to ASM” changed to “ASM”	Cover page
		- review of the background	2
		- review of purpose of the document	2, 3
		- review of the scope of the document	3
		- review of the applicability	3-5
		- review of maintenance of the specification	6
		- review of target audience of the specification	6
		- updated definitions: added Booking, removed cross border airspace, interoperability, procedures, system, modified external user	10-13
		- updated list of reference material in section 1.10	14, 15
		- updated document structure	15
		- review and update of section 2, including service abstract, service identification, operational information exchange requirements, and use of the service	16-21
		- review and update of service interfaces overview,	22
		- review and update of subscription management interface - new provisions on message timeout, heartbeat mechanism, heartbeat technical messages and subscription technical messages; added a new interface function: “Listing existing Subscriptions provides access to existing Subscriptions in the ASM Support System for the External User” and updated requirement ASM-INTF-SUBS-020.	22-27
		- new requirement: ASM-INTF-SUBS-115	28
		- new requirements: ASM-INTF-SUBS-163, 165, 167, 168, 215, 217, 270, 280 and 290	29-31
		- modified requirement ASM-INTF-PUB-010, 020	33

		- new requirement ASM-INTF-PUB-025	33
		- updated requirement ASM-INTF-LTPL-010	37
		- updated requirement ASM-INTF-LTPL-090, 100 and 110	38
		- new requirement ASM-INTF-LTPL-120	38
		- updated requirement ASM-INTF-LAS-010, 030, 060, 070	40, 41
		- new requirement ASM-INTF-LAS-055, 085	41
		- updated requirement ASM-INTF-ARES-010, 130, 190, 240,	45-48
		- updated requirement ASM-INTF-CON-010, 030	50
		- Updated requirement ASM-INTF-MIS-010, 100, 110,	52, 53
		- new requirement ASM-INTF-MIS-130	53
		- updated requirement ASM-INTF-NEG-010, 060	55-56
		- new requirement ASM-INYF-NEG-085	56
		- updated Airspace Status interface role	57
		- updated Figure 32: Airspace status interface overview	58
		- new use cases added	58-60
		- updated Figure 33: Airspace status interface information exchange flow	59
		- updated Figure 34: Use of Airspace Status AMQP interface	60
		- updated requirement ASM-INTF-STAT-010	61
		- new requirements ASM-INTF-STAT-055, 060, 070, 080, 090, 100	61, 62
		- updated requirements ASM-INTF-TIP-010, 020, 040	63, 64
		- updated requirements ASM-INTF-SDR-020, 030	65
		- update information definition scope	66
		- updated abstract messages	66
		- added ActivationReceivedRequest message	66
		- updated ActivationNotification message	66
		- added HearbeatTechnicalMessage	72
		- added SubscriptionTechnicalMessafge	75
		- updated ActivityData type	76
		- updated BookingStatus type	85
		- updated FeatureType type	87
		- updated AirspaceName Filter	89
		- updated AirspaceTypeFilter	89
		- updated Service quality – Non-functional requirements	94-95
		- updated airspace status behaviour	98
		- updated mission behaviour	98
		- updated Annex A: Reference to PCP Regulation replaced by reference to CP1 Regulation	100
		- updated Annex B	102-116
		- updated Annex C:	118-120

		- to take account of new edition of EUROCONTROL-SPEC-168 - new text added on conformity with EUROCONTROL-SPEC-170 - updated Annex D – new requirements added: ASM-INTF-SUBS-163, ASM-INTF-SUBS-165, ASM-INTF-SUBS-167, ASM-INTF-SUBS-215, ASM-INTF-SUBS-217, ASM-INTF-SUBS-270, ASM-INTF-SUBS-280, ASM-INTF-SUBS-290, ASM-INTF-PUB-025, ASM-INTF-STAT-060, ASM-INTF-STAT-070, ASM-INTF-STAT-080, ASM-INTF-STAT-090, ASM-INTF-STAT-100	122-127

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Page left blank intentionally

EXECUTIVE SUMMARY

To date, the document which provides technical requirements for an Airspace Management (ASM) Support System at European level is the EUROCONTROL Specification for the application of Flexible Use of Airspace (FUA Specification). The technical requirements in the Specification are focusing on States' compliance with the requirements stemming from the Commission Regulation (EC) No 2150/2005 (FUA Regulation), namely Art.3 (b), Art.5.3, Art.6.1, Art.6.2, and Art.6.3. Additional system requirements are identified as Recommendations to support and automate the FUA processes.

The technical requirements listed in the FUA Specification remain at high level and do not cater for harmonisation of the system functionalities required by different States. Moreover, some of the functionalities required by the Member States to ensure compliance with the provisions of the FUA Regulation related to ASM are not addressed.

The objective of this document is to provide a set of commonly agreed ASM Support System requirements. Conformance with these requirements, despite any possible differences in detailed specifications of the systems, will ensure harmonisation of the application of the systems, their interoperability and will facilitate the development of a standard system-to-system interfaces.

Part I covers the baseline system requirements for ASM Support System supporting the ASM processes at local and FAB level.

Part II of the Specification - this document - covers system interface requirements for ASM Support System supporting the ASM processes at local and FAB level.

The technical interface requirements are derived from the operational needs and requirements as specified by Annex 11 of ERNIP Part 3 - ASM Handbook with regard to local ASM Support System interoperability requirements.

This document facilitates compliance with the FUA Regulation [2] and other relevant SES Regulations.

1. Introduction

1.1 Background

This document is the EUROCONTROL[1] Specification for Airspace Management (ASM) Support System Requirements supporting the ASM processes at local and FAB level - Part II: ASM Systems Interface requirements. It has been developed in collaboration with stakeholders from civil and military air navigation service providers and airspace users.

EUROCONTROL Specifications have voluntary status and are developed to support Members States and stakeholders.

This Specification does not mandate the implementation of ASM interfaces but specifies the requirements that need to be met when implementing such facilities. If interfaces are implemented as the result of regulatory provisions, or based on bilateral agreements between States/AMCs, then the requirements outlined as mandatory (requirements using “shall”) in this Specification for those interfaces have to be considered mandatory for implementation. This is required in order to meet the purpose of the interfaces and to ensure interoperability between local ASM Support systems.

To date, the document which provides technical requirements for an Airspace Management (ASM) Support System at European level is the EUROCONTROL Specification for the application of Flexible Use of Airspace (FUA Specification, EUROCONTROL-SPEC-0112). The technical requirements in the Specification are focusing on States’ compliance with the requirements that are stemming from the Commission Regulation (EC) No 2150/2005 of 23 December 2005 laying down common rules for the flexible use of airspace, namely Art.3 (b), Art.5.3, Art.6.1, Art.6.2, and Art.6.3.

EUROCONTROL Specification for Airspace Management (ASM) Support System Requirements supporting the ASM processes at local and FAB levels (Part II ASM Systems Interface Requirements) – this document - defines how to exchange the optimised set of ASM data in a standardised manner (at local and FAB levels). It describes the ASM service and the consolidated technical interface requirements. This includes the development of SWIM-conformant ASM Support System interfaces, i.e. ASM service will be developed in line with the three EUROCONTROL SWIM Specifications (see Ref. [10], [11], and [12]).

1.2 Purpose of the document

The purpose of this document is to provide a set of commonly-agreed ASM Support System performance requirements applicable to ASM Support Systems to standardise the necessary interfaces for systems supporting/offering airspace management functionalities in the context of, and in support of, the implementation of the Advanced Flexible Use of Airspace Concept (AFUA Concept).

Conformance with these requirements, despite any possible differences in the detailed specifications of the systems, will ensure harmonisation of the application of the systems, their interoperability and will enable the exchange of information between ASM Support Systems via standard interface. This Specification complements EUROCONTROL–SPEC-0112 with regard to ASM Support System requirements.

---

[1] EUROPEAN ORGANISATION FOR THE SAFETY OF AIR NAVIGATION

The Specification also supports Stakeholders in achieving compliance with regulatory requirements set out in the Common Project One (CP1) Regulation AF#3 and AF#5.

1.3 Scope of the document

The ASM Specification – Part II - covers the technical requirements for a standardised interface between national ASM Support Systems, as well as between ASM Support Systems and External Users, involved in airspace management processes. The requirements are consistent with the high-level system requirements set out in the CP1 Regulation.

The technical ASM Support System interface requirements at local/FAB level are derived from the operational requirements as specified by Annex 11 of ERNIP Part 3 - ASM Handbook. The scope of the Specification is as follows:

- Definition of the ASM Service, including:
    - Information definition
    - Error handling
    - Service quality – non-functional considerations
    - Technical Infrastructure requirements – e.g. transmission protocols

This Specification does not address interface requirements between NM, AIS providers, and AOs/CFSPs systems.

With specific regards to the interface requirements with NM, the NM B2B Airspace Structure and Airspace Availability Services are already specified in the document NM NOP/B2B Reference Manuals – Airspace Services, Document Reference B2B/Airspace, latest release, and are outside of the scope of this Specification.

The Specification does not cover hardware requirements and system specifications. It does not aim to determine networks, conceptual and/or physical links.

1.4 Applicability

The CP1 Regulation [3] requires the deployment of the “Flexible Airspace Management and Free Route Airspace” functionality. It sets out that:

“Airspace management and advanced flexible use of airspace contributes to achieving the ‘Fully dynamic and optimised airspace’ EOC. Increased performance of ATM requires that changes in airspace status are constantly shared with all concerned ATM actors, in particular the Network Manager, the ANSPs and the airspace users (flight operations centre/wing operations centre (‘FOC/WOC’)). Airspace management (‘ASM’) and advanced flexible use of airspace (‘A-FUA’) aim at providing the most efficient airspace organisation and management in response to airspace user’s needs. ASM with A-FUA provides a solution for dynamically managing airspace users’ demands in various operating environments.”

The CP1 Regulation [3] also establishes the high-level system requirements, of which the following ones are particularly relevant to the development of this ASM Specification:

- In AF 3: Flexible Airspace Management and Free Route Airspace, point 3.1.1 ATM sub-functionality: Airspace management and advanced flexible use of airspace:

“System Requirements

(a) *The ASM support systems must support the fixed and conditional route networks, FRA and flexible sector configurations and must be able to respond to changing demands for airspace.*

(b) *The ASM system must support cross-border activities resulting in shared use of volume of airspace regardless of national boundaries.*

…

(g) *The ASM, ATFCM and ATC systems must be interoperable allowing the provision of air navigation services based on a common understanding of the airspace and traffic environment.*

…

(m) *Data exchange between stakeholders mandated to deploy the flexible airspace management and FRA set out in AF3 must be implemented using SWIM services as set out in AF5, where SWIM is available. The concerned systems must be able to provide or use SWIM services. Existing data exchange technology may be used until SWIM is available.*

…

- In AF 5: System Wide Information Management, point 5.1.2. ATM sub-functionality on SWIM yellow profile technical infrastructure and specifications

…

“SWIM yellow profile technical infrastructure must be used for ATM data exchange for all other ATM functionalities.”

…

- In AF 5: System Wide Information Management, point 5.1.3. ATM sub-functionality on Aeronautical information exchange

“Operational stakeholders must implement the following services that support the exchange of the aeronautical information using the SWIM yellow profile technical infrastructure as described in the deployment programme:

(a) *notification of the activation of an airspace reservation/restriction (‘ARES’);*

(b) *notification of the de-activation of an ARES;*

(c) *pre-notification of the activation of an ARES;*

(d) *notification of the release of an ARES;*

(e) *aeronautical information feature on request; filtering possible by feature type, name and an advanced filter with spatial, temporal and logical operators;*

(f) *query ARES information;*

(g) digital aerodrome charts;

(h) ASM level 1;

(i) airspace use plans (AUP, UUP) – ASM level 2 and 3;

(j) digital NOTAM.

System requirements

- *a) All the services listed in point 5.1.3 must comply with applicable SWIM specifications*

This specification can also be adopted outside of the specific CP1 context.

This specification is expected to be applied by civil and military ATS service providers when implementing ASM Support Systems.

This specification does not apply to Aircraft Operators who are required to use the EUROCONTROL NM Services.

The military operational stakeholders shall deploy those ATM functionalities only to the extent necessary to comply with the fourth and fifth subparagraphs of point 3.2 of Annex VIII to Regulation (EU) 2018/1139 of the European Parliament and of the Council (EU2021/116 art.3). [14]

Point 3.2 Annex VIII of 2018/1139]

….

“The EATMN, its systems and their constituents shall support the progressive implementation of civil/military coordination, to the extent necessary for effective airspace and air traffic flow management, and the safe and efficient use of airspace by all users, through the application of the concept of the flexible use of airspace.

To achieve those objectives, the EATMN, its systems and their constituents shall support the timely sharing of correct and consistent information covering all phases of flight, between civil and military parties, without prejudice to security or defence policy interests, including requirements on confidentiality.”

1.5 Conventions

The following conventions are used in this EUROCONTROL Specification:

a. “**Shall**” – indicates a statement of specification, the compliance with which is mandatory to achieve the implementation of this EUROCONTROL Specification.

b. “**Should**” – indicates a recommendation or best practice, which may or may not be satisfied by all systems claiming conformity to this EUROCONTROL Specification.

c. “**May**” – indicates an optional element.

Keywords are highlighted in the requirement text using **bold** as shown above.

Numbers within square brackets are used to identify reference documents listed in section 1.10 e.g. [1] identifies the first reference documents of section 1.10.

Every requirement in this EUROCONTROL Specification is followed by a structured identifier, which can be used to uniquely reference the requirement/recommendation from associated documents and traceability tools. Such identifiers have the form:

ASM-[yy]-[Fn]-[nnn],

where:

ASM	stands for ASM Support System requirement;
[yy]:	Is a sequence of 2 to 4 characters to identify the environment to which the requirement is referring to; *Note: In this Specifications this sequence of characters will be* ***INTF***
[Fn]:	Is a sequence of 2 to 4 characters to identify the functionality of ASM Support System to which the requirement applies (e.g. ARES – airspace reservation);
[nnn]:	Is a numeric identifier, for a sequence of requirements with the same **[Fn]** identifier[2].

1.6 Maintenance of the Specification

This EUROCONTROL Specification has been developed under the EUROCONTROL Standards development process and is maintained by EUROCONTROL in line with the EUROCONTROL Standards Development Procedures. This process is summarised in Annex E.

1.7 Target Audience

The target audience for the specification includes, but is not limited to:

- operational stakeholders, civil and military, implementing services supporting the exchange of information in the context of Airspace Management. This audience includes:
  - technical experts designing and implementing ASM support systems and services; and
  - operational experts using ASM support systems and services to fulfil operational needs.
- oversight authorities.

1.8 Abbreviations and Acronyms

Abbreviation	Term
A/B	Air Base
ACC	Area Control Centre

---

[2] Requirement numbers are initially allocated incrementally in tens. This aids the subsequent management of this specification allowing new requirements to be inserted between existing requirements whilst maintaining a logical number sequence.

AIP	Aeronautical Information Publication
ADEP	Aerodrome of Departure
ADES	Aerodrome of Destination
AF	ATM Functionality (in the PCP regulation)
AFUA	Advanced Flexible Use of Airspace
AIRM	ATM Information Reference Model
AIXM	Aeronautical Information Exchange Model
AMQP	Advanced Message Queuing Protocol
AO	Aircraft Operator
ARES	Airspace Reservation/Restriction
ASM	Airspace Management
ATC	Air Traffic Control
ATFCM	Air Traffic Flow and Capacity Management
ATM	Air Traffic Management
ATS	Air Traffic Services
AUP/UUP	Airspace Use Plan / Updated Airspace Use Plan
B2B	Business-to-Business
CBA	Cross Border Area
CBO	Cross Border Operations
CDM	Collaborative Decision Making
CDR	Conditional Route
CR	Change Request
eAMI	Electronic ASM Information message
EC	European Commission
ERAF	EUROCONTROL Advisory Framework
EU	European Union
FAB	Functional Airspace Block

FBZ	Flight Plan Buffer Zone
ENV	Environment
FDPS	Flight Data Processing System
FF	Fire and Forget
FL	Flight Level
FMP	Flow Management Position
FUA	Flexible Use of Airspace
GML	Geographical Markup Language
HTTP	Hypertext Transport Protocol
ICAO	International Civil Aviation Organisation
ID	Identifier
IP	Internet Protocol
MTBF	Mean Time Between Failures
MTTR	Mean Time To Recovery
NOP	Network Operations Plan
NM	Network Manager
NOTAM	Notice to Airmen
PCP	Pilot Common Project (EC Regulation)
POC	Point of Contact
ERNIP	European Route Network Improvement Plan
SES	Single European Sky
SOAP	Simple Object Access Protocol
sR/R	Synchronous Request Reply
SWIM	System Wide Information Management
TI	Technical Infrastructure
TRA	Temporary Reserved Area
TSA	Temporary Segregated Area

UUID	Universally Unique Identifier
UOM	Unit of Measurement
WS	Web Service
XML	Extensible Markup Language
WSDL	Web Services Description Language

1.9 Definitions

Term	Definition	Source
airspace management	A planning function with the primary objective of maximising the utilisation of available airspace by dynamic time-sharing and, at times, the segregation of airspace among various categories of airspace users on the basis of short-term needs.	European Route Network Improvement Plan – Part 3 – ASM Handbook
action	A specific permission to interact with a booking via the interfaces offered by this service. Actions may change with time and the state of the booking. Different actions may be performed via the same service operation in which case the modifiable data fields may be restricted by the currently available actions.	-
activity data	Represents additional system specific data associated with an airspace structure.	-
airspace reservation	A defined volume of airspace temporarily reserved for exclusive or specific use by categories of users.	European Route Network Improvement Plan – Part 3 – ASM Handbook

Term	Definition	Source
airspace structure	Specific portions of airspace designed to accommodate the safe operation of aircraft.	European Route Network Improvement Plan – Part 3 – ASM Handbook
ASM actors	Human or system that participate in the ASM process	EUROCONTROL Specification for Airspace Management (ASM) System Requirements supporting the ASM processes at local and FAB level Part I - Baseline Requirements
asynchronous	An interaction is said to be asynchronous when the associated messages are chronologically and procedurally decoupled. For example, in a request-response interaction, the client agent can process the response at some indeterminate point in the future when its existence is discovered. Mechanisms to do this include polling, notification by receipt of another message, etc.	W3C Web Services Glossary [13]
Booking	A defined grouping of one or more airspace reservations (ARES) across different airspaces on which a single activity will take place.	-

Term	Definition	Source
conflict	An overlap, both spatial and temporal, between any of the airspace structures in one airspace reservation (ARES) with any of the airspace structures of another ARES.	EUROCONTROL Specification for Airspace Management (ASM) System Requirements supporting the ASM processes at local and FAB level Part I - Baseline Requirements
client	An application making use of any given service.	-
consumer	A Consumer is the application that receives the messages from the message queue.	-
cross border area (CBA)	An airspace restriction or reservation established over international borders for specific operational requirements. This may take the form of a Temporary Segregated Area or Temporary Reserved Area.	European Route Network Improvement Plan – Part 3 – ASM Handbook
external user	A local/FAB external systems that is a client of the ASM service. For the purposes of this specification, an External User includes other ASM Support Systems and ATC systems.	-
flexible use of airspace (FUA)	Concept is based on the fundamental principle that airspace should not be designated as either pure civil or military airspace, but rather be considered as one continuum in which all user requirements have to be accommodated to the extent possible.	European Route Network Improvement Plan – Part 3 – ASM Handbook

Term	Definition	Source
free route operations airspace (FRA)	A specified airspace within which users may freely plan a route between a defined entry point and a defined exit point, with the possibility to route via intermediate (published or unpublished) way points, without reference to the ATS route network, subject to airspace availability. Within this airspace, flights remain subject to air traffic control.	European Route Network Improvement Plan – Part 3 – ASM Handbook European Route Network Improvement Plan – Part 1 – Technical Specifications for Airspace Design
FUA restriction	The restriction introduced in the CACD database in order to manage the acceptance of FPLs through the related restricted/reserved area. With the activation of the FUA restriction, all the FPL passing through the related restricted/reserved area will be rejected, unless related to any inclusions and exclusions defined in the restriction. The activation of the FUA restriction will be triggered by the allocation of the associated reserved/restricted area through AUP/UUP.	European Route Network Improvement Plan – Part 3 – ASM Handbook
Interface binding	Specification of the protocol and data format to be used in transmitting messages defined by the associated interface.	W3C Web Services Glossary [13]

Term	Definition	Source
message exchange pattern (MEP)	A Message Exchange Pattern (MEP) is a template, devoid of application semantics, that describes a generic pattern for the exchange of messages between agents. It describes the relationships (e.g., temporal, causal, sequential, etc.) of multiple messages exchanged in conformance with the pattern, as well as the normal and abnormal termination of any message exchange conforming to the pattern.	W3C Web Services Glossary [13]
producer	A Producer is the application that is sending the messages to the message queue.	-
proposal	Update to a subset of the reservation data, namely FLs, times, or area. Proposals could be accepted or rejected.	-
publisher	A Publisher is the application that is sending the messages to the message queue.	-
subscriber	A Subscriber is the application that receives the messages from the message queue.	-
simple object access protocol (SOAP)	The formal set of conventions governing the format and processing rules of a SOAP message. These conventions include the interactions among SOAP nodes generating and accepting SOAP messages for the purpose of exchanging information along a SOAP message path.	W3C Web Services Glossary [13]

Term	Definition	Source
synchronous	An interaction is said to be synchronous when the participating agents must be available to receive and process the associated messages from the time the interaction is initiated until all messages are actually received or some failure condition is determined. The exact meaning of "available to receive the message" depends on the characteristics of the participating agents (including the transfer protocol it uses); it may, but does not necessarily, imply tight time synchronization, blocking a thread, etc.	W3C Web Services Glossary [13]
topic or **subscription topic**	A subject to which clients can subscribe to receive related notifications.	-

1.10 Reference material

[1] Regulation (EC) No 552/2004 of the European Parliament and of the Council of 10 March 2004 on the interoperability of the European Air Traffic Management network (the interoperability Regulation), as amended by Regulation (EC) No 1070/2009 of the European Parliament and of the Council of 21 October 2009 amending Regulations (EC) No 549/2004, (EC) No 550/2004, (EC) No 551/2004 and (EC) No 552/2004 in order to improve the performance and sustainability of the European aviation system

[2] Commission Regulation (EC) No 2150/2005 of 23 December 2005 laying down common rules for the flexible use of the airspace

[3] Commission Implementing Regulation (EU) No 2021/116 of 1 February 2021 on the establishment of the Common Project One supporting the implementation of the European Air Traffic Management Master Plan provided for in Regulation (EC) No 550/2004 of the European Parliament and of the Council, amending Commission Implementing Regulation (EU) No 409/2013 and repealing Commission Implementing Regulation (EU) No 716/2014 (The CP1 Regulation).

[4] Commission Regulation (EC) No 482/2008 of 30 May 2008, establishing a software safety assurance system to be implemented by air navigation service providers and amending Annex II to Regulation (EC) No 2096/2005

[5] Commission Implementing Regulation (EU) 2019/123 of 24 January 2019 laying down

detailed rules for the implementation of air traffic management (ATM) network functions and repealing Commission Regulation (EU) No 677/2011

[6] EUROCONTROL Specification for application of the Flexible Use of Airspace (FUA), EUROCONTROL-SPEC-0112, Edition 1.1, dated 10.01.2009

[7] European Route Network Improvement Plan, Part 1, European Airspace Design Methodology Guidelines - General Principles and Technical Specifications for Airspace Design, Latest Edition , https://www.eurocontrol.int/publication/european-route-network-improvement-plan-ernip-part-1

[8] European Route Network Improvement Plan - Part 3, Airspace Management Handbook, Procedures for Airspace Management, Latest Edition, https://www.eurocontrol.int/publication/european-route-network-improvement-plan-ernip-part-3

[9] NMOC Flight Planning Requirements - Guidelines, Latest Edition, https://www.eurocontrol.int/publication/nm-flight-planning-requirements-guidelines

[10] EUROCONTROL Specification for SWIM Service Description, EUROCONTROL-SPEC-168, Edition 2.0, 15 March 2022

[11] EUROCONTROL Specification for SWIM Information Definition, EUROCONTROL-SPEC-169, Edition 1.0, 01 December 2017

[12] EUROCONTROL Specification for SWIM Technical Infrastructure (TI) Yellow Profile, EUROCONTROL-SPEC-170, Edition 1.1, 05 July 2020

[13] World Wide Web Consortium (W3C) Web Services Glossary (2004), http://www.w3.org/TR/ws-gloss/

[14] Regulation (EU) 2018/1139 of the European Parliament and of the Council of 4 July 2018 on common rules in the field of civil aviation and establishing a European Union Aviation Safety Agency, and amending Regulations (EC) No 2111/2005, (EC) No 1008/2008, (EU) No 996/2010, (EU) No 376/2014 and Directives 2014/30/EU and 2014/53/EU of the European Parliament and of the Council, and repealing Regulations (EC) No 552/2004 and (EC) No 216/2008 of the European Parliament and of the Council and Council Regulation (EEC) No 3922/91

1.11 Document structure

- Section 1 describes the context, the purpose and scope of the document. It also describes the structure of the document and the applicable maintenance process.
- Section 2 contains the ASM Service definition and requirements.

  The requirements in this section have been defined taking a service approach; an ASM service definition is provided preceding the set of requirements. Information definition requirements are also included.

Annex A provides traceability to regulatory requirements.

Annex B provides traceability to operational requirements from Annex 11 / ASM Handbook Ed. [ ], relevant to this Specification.

Annex C provides a conformity matrix with SWIM requirements.

Annex D provides a conformity checklist with this Specification.

Annex E details the Specification update procedures.

Annex F contains the report on the semantic correspondence of the Information Definition found in this document with the semantics of the ATM Information Reference Model (AIRM).

Annex G specifies the Web Services Description Language (WSDL).

2. ASM service

2.1 Overview

2.1.1 Service Abstract

The ASM service enables the complete set of ASM functions necessary for the interactions between ASM Support System and External Users, when communicating relevant ASM process information at local/FAB level.

2.1.2 Service Identification

Service Name	ASM
Service Version	2.0

Table 1: ASM Service Identification

2.1.3 Service category

Category	**value**
information exchange area	Aeronautical information exchange
availability status	N/A (this is a service definition)
business activity	airspace management
intended service provider	ASM Support System
intended service consumer	External ASM User
geographical extent	N/A (this is a service definition)

Table 2: ASM Service Category

2.2 Operational Information Exchange Requirements

ASM Support Systems facilitate the exchange of pre-tactical and tactical ASM data between ASM Support Systems and External Users.

This information exchange supports the optimisation of ASM operations from the beginning of the airspace planning cycle.

Furthermore, a national ASM Support System will allow the creation of bookings in airspace structures managed by another national ASM Support System, according to the procedures defined in ERNIP part 3 ASM Handbook and bilateral agreements between States.

The traceability of the requirements specified in this specification document to information exchange requirements, defined in Annex 11 of the ASM Handbook, can be found in Annex B of this document. These information exchange requirements have been used in the identification for the service interfaces defined in this specification document.

2.3 Use of the service

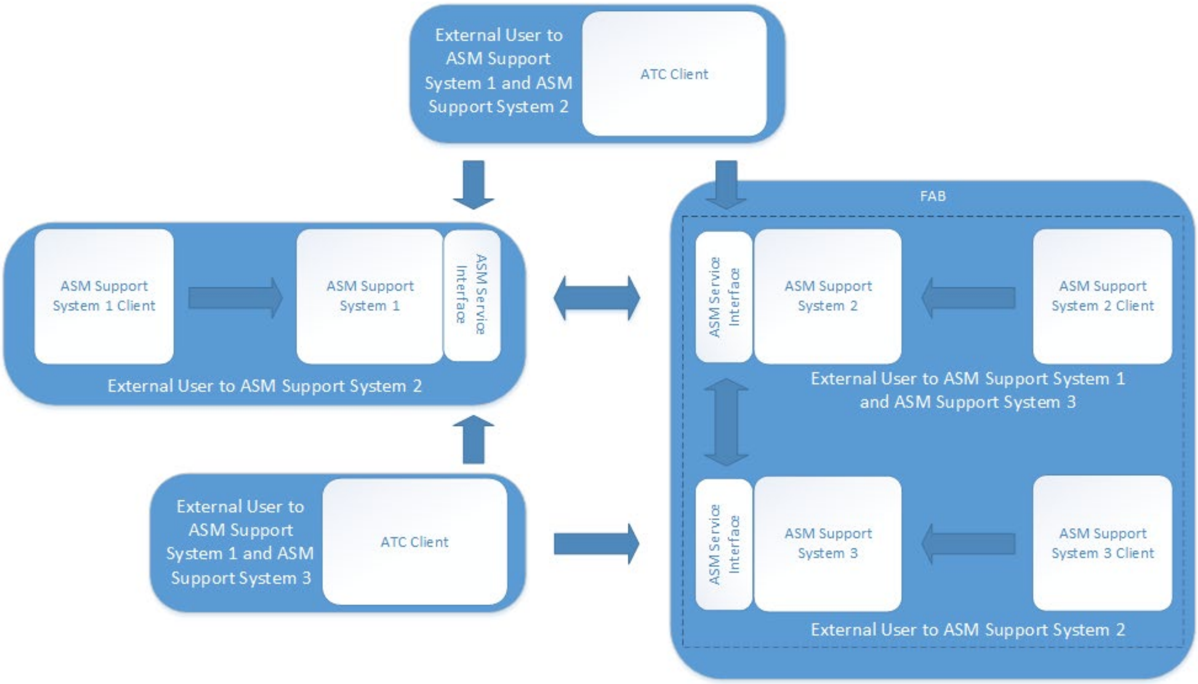

External User to ASM Support System 1 and ASM Support System 2
ATC Client
ASM Support System 1 Client
ASM Support System 1
ASM Service Interface
External User to ASM Support System 2
External User to ASM Support System 1 and ASM Support System 3
ATC Client
FAB
ASM Service Interface
ASM Support System 2
ASM Support System 2 Client
External User to ASM Support System 1 and ASM Support System 3
ASM Service Interface
ASM Support System 3
ASM Support System 3 Client
External User to ASM Support System 2

Figure 1: Use of the ASM Service

2.3.1 Overview

The ASM Service is designed to support data exchange between ASM Support Systems and External Users both within and across FAB boundaries.

Such an exchange is shown on Figure 1 above between ASM Support System 1 and ASM Support System 2 and between ASM Support System 2 and ASM Support System 3. The ASM support systems both connect to each other acting as both a client and a server to the other, whereas the ASM clients only connect directly to their own ASM support system. Data flow between the client and server is bidirectional.

On Figure 1 above, the arrows do not represent data flow. They indicate the direction of a request being made using the ASM service.

The ATC clients shown in the diagram connect to the relevant ASM support system that is responsible for managing the data the ATC client wishes to subscribe for. Provision of data from the ASM Support System to the ATC client is unidirectional, in that the ATC client does not provide data to the ASM support system. However, the ATC client notifies of the receipt of activation data to which it has subscribed.

2.3.2 ASM Support System - External User

In this configuration the ASM Support Systems can each act as a client to each other, though the relationship does not need to be bidirectional. This allows each ASM Support System privilege to distribute general information from the other system to its users. It also allows each ASM Support System to request and modify bookings in the other system and engage in coordination with the other system directly. It is the responsibility of the system providing the services to restrict the data shared with all or each client as appropriate. Equally, it is the responsibility of the client system to ensure booking, creation and editing is granted for the appropriate individual users.

In the case when the External User is an ATC system, the ASM system acts as the provider while the ATC system acts as the consumer. However, the consumer can notify that it has received activation data.

2.3.3 Obtaining up to date data

The interfaces defined within this document are supported by both a Synchronous Request/Reply mechanism and a Publish/Subscribe Push mechanism.

2.3.3.1 Publish/Subscribe Push

The Publish/Subscribe Push mechanism allows a client to subscribe to a topic and receive messages published on that topic. The Publish/Subscribe mechanism is supported by two procedures:

- subscription management: the client creates a new subscription and, in general, a corresponding message queue is allocated to collect messages related to the requested subscription topic. (see SubscriptionManagement interface)
- message consumption: the client consumes the messages from the message queue for the subscription topic. (see Publication interface)

The Publish/Subscribe mechanism is implemented using AMQP to support coordination between ASM Support Systems and External User to exchange the near real-time updates of their ASM data.

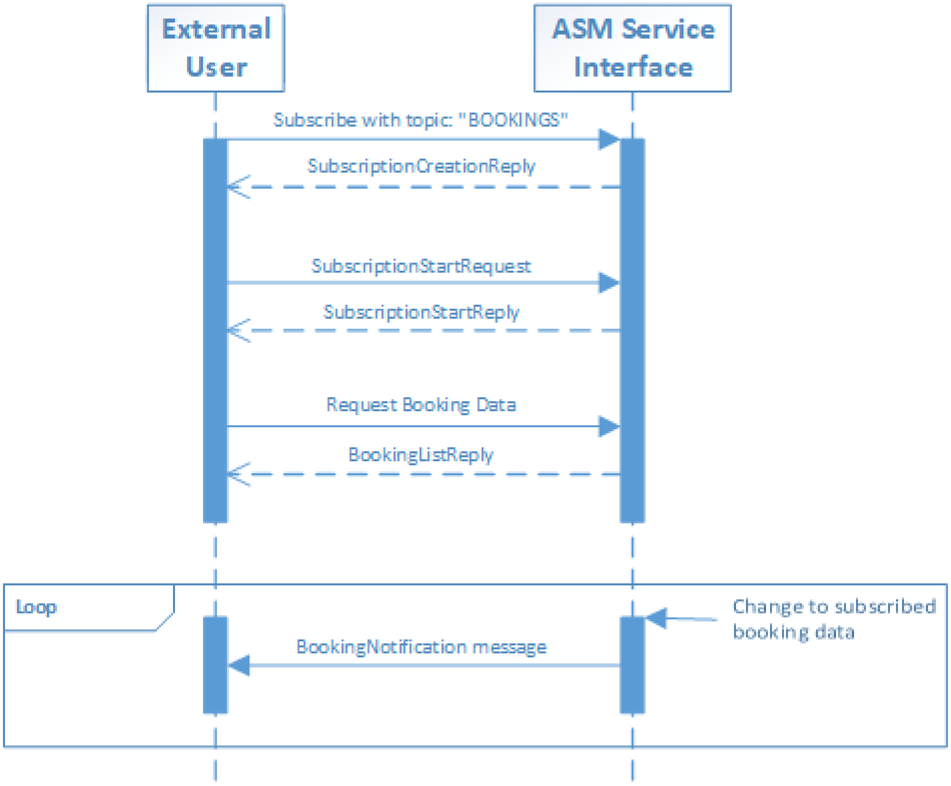


Figure 2: Publish/subscribe mechanism

2.3.3.2 Request/Reply

The Synchronous Request/Reply mechanism allows a client to request and to receive ASM data. It is used to retrieve the current ASM data of the other system either on the initial connection between ASM Support Systems and External Users or to re-establish synchronisation potentially after one has experienced downtime or due to network failures.

This Request/Reply mechanism may also be implemented by clients that are interested in a small subset of the ASM service where the data is subject to change infrequently or when using latest data is not crucial.

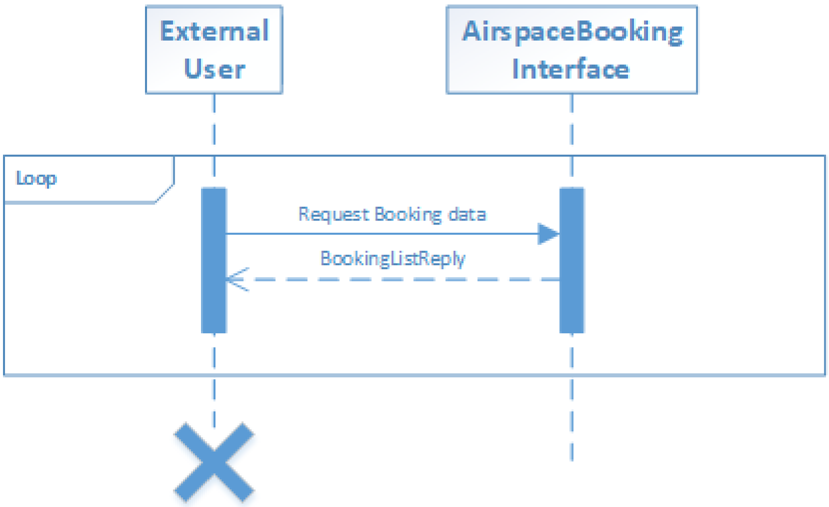


Figure 3: Synchronous request-reply mechanism

As two ASM Support Systems may not be fully synchronised all of the time, Request/Reply mechanism is implemented to re-establish synchronisation by exchanging the latest snapshot of the available ASM data. The systems are synchronised again and they can continue to be updated by using the Publish/Subscribe mechanism.

2.3.3.3 Filtering

To avoid unnecessary processing of irrelevant messages on the client side some message filtering functionalities have to be implemented by both Request/Reply and Publish/Subscribe mechanisms through a combination of the following filter criteria for each ASM Service Interface:

- Activity,
- Airspace,
- Airspace Name,
- Airspace Feature Type,
- Change Period,
- Geometry,

- o Interested Interval,
- o Mission.

When using the Publish/Subscribe mechanism, the External User has to set the required message filtering criteria when creating the subscription so that the messages for each type of subscriptions can be filtered before they are published to the message queues allocated to that External User.

The Request/Reply mechanism requires the External Users to set the required message filtering criteria when generating the request.

To allow ASM data requests to be aligned with subscriptions a single abstract Filter definition is available for use by both mechanisms, in that the SubscriptionCreationRequest message is an extension to the basic FilteredRequest message as shown in the diagram below. Consequently, the same Filters can be used when requesting ASM data and subscribing for ASM data to ensure the same data is retrieved via both mechanisms.

Specific filter implementations that have been deemed necessary for the services to be usable are defined in this document. The available filters may be extended beyond those specified in any implementation of this service. This allows for custom filtering to be defined by a provider of this service while still conforming to the interfaces defined in this document. It has to be noted that only the filters defined in the document are to be relied upon in order to create an implementation independent client.

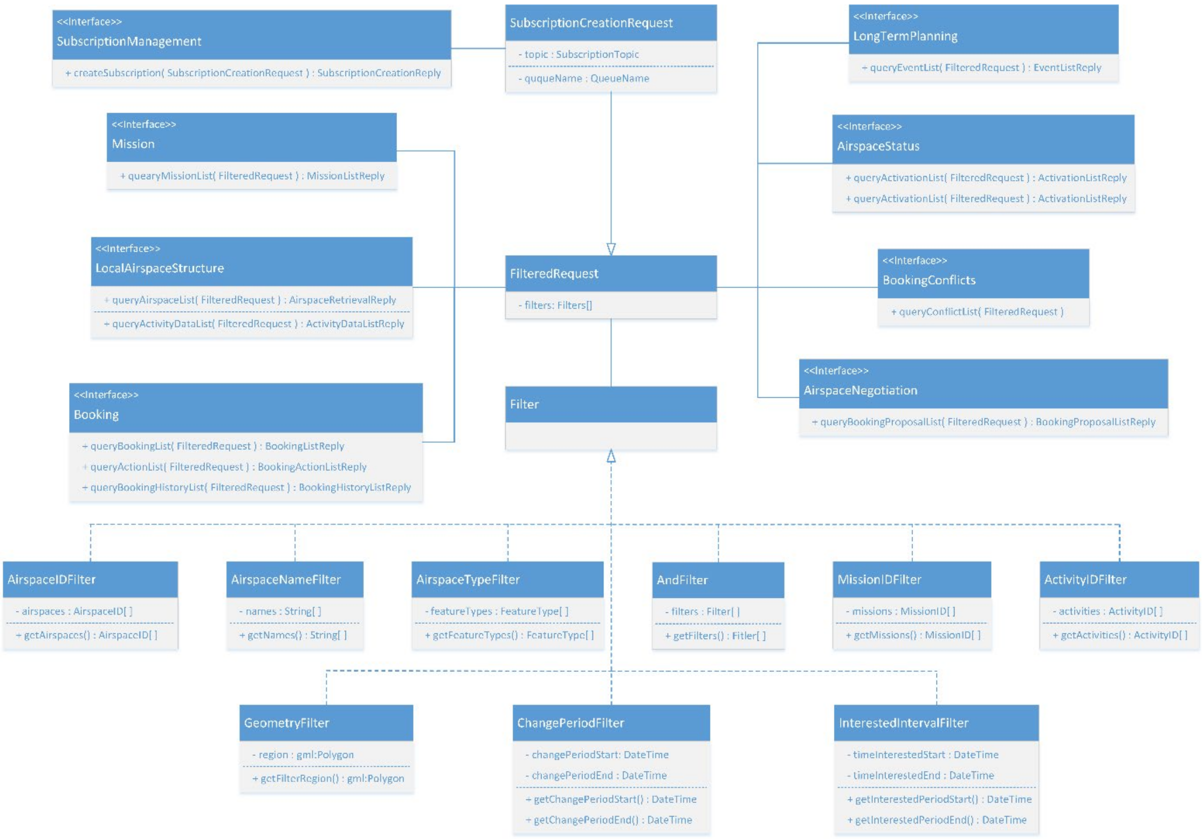

<<Interface>>
SubscriptionManagement
+ createSubscription(SubscriptionCreationRequest) : SubscriptionCreationReply
SubscriptionCreationRequest
- topic : SubscriptionTopic
- ququeName : QueueName
<<Interface>>
LongTermPlanning
+ queryEventList(FilteredRequest) : EventListReply
<<Interface>>
Mission
+ quearyMissionList(FilteredRequest) : MissionListReply
<<Interface>>
AirspaceStatus
+ queryActivationList(FilteredRequest) : ActivationListReply
+ queryActivationList(FilteredRequest) : ActivationListReply
<<Interface>>
LocalAirspaceStructure
+ queryAirspaceList(FilteredRequest) : AirspaceRetrievalReply
+ queryActivityDataList(FilteredRequest) : ActivityDataListReply
FilteredRequest
- filters: Filters[]
<<Interface>>
BookingConflicts
+ queryConflictList(FilteredRequest)
<<Interface>>
Booking
+ queryBookingList(FilteredRequest) : BookingListReply
+ queryActionList(FilteredRequest) : BookingActionListReply
+ queryBookingHistoryList(FilteredRequest) : BookingHistoryListReply
Filter
<<Interface>>
AirspaceNegotiation
+ queryBookingProposalList(FilteredRequest) : BookingProposalListReply
AirspaceIDFilter
- airspaces : AirspaceID[ ]
+ getAirspaces() : AirspaceID[ ]
AirspaceNameFilter
- names : String[ ]
+ getNames() : String[ ]
AirspaceTypeFilter
- featureTypes : FeatureType[ ]
+ getFeatureTypes() : FeatureType[ ]
AndFilter
- filters : Filter[ ]
+ getFilters() : Fitler[ ]
MissionIDFilter
- missions : MissionID[ ]
+ getMissions() : MissionID[ ]
ActivityIDFilter
- activities : ActivityID[ ]
+ getActivities() : ActivityID[ ]
GeometryFilter
- region : gml:Polygon
+ getFilterRegion() : gml:Polygon
ChangePeriodFilter
- changePeriodStart: DateTime
- changePeriodEnd : DateTime
+ getChangePeriodStart() : DateTime
+ getChangePeriodEnd() : DateTime
InterestedIntervalFilter
- timeInterestedStart : DateTime
- timeInterestedEnd : DateTime
+ getInterestedPeriodStart() : DateTime
+ getInterestedPeriodEnd() : DateTime

Figure 4: Filtering mechanism

As shown in the diagram above, the FilteredRequest is used throughout the service interfaces and is extended to form the SubscriptionCreationRequest. The FilteredRequest defines a series of Filters which will be applied using the logical OR operator. Filters may be combined with a logical "AND" using the "AndFilter".

Later in this document, each service interface defines in detail the Filters it accepts.

ASM-INTF-FIL-010: All ASM service interfaces specified in this document **shall** implement a filtering functionality both for Synchronous Request/Reply and Publish/Subscribe Push application message exchange patterns.

2.4 Service Interfaces

2.4.1 Service Interfaces Overview

The following diagram shows the high-level interfaces breakdown of the ASM service.

Note: None of the interfaces of the ASM Service is mandatory, however, if any of them is to be deployed, the requirements specified in this Specification shall be complied with in full.

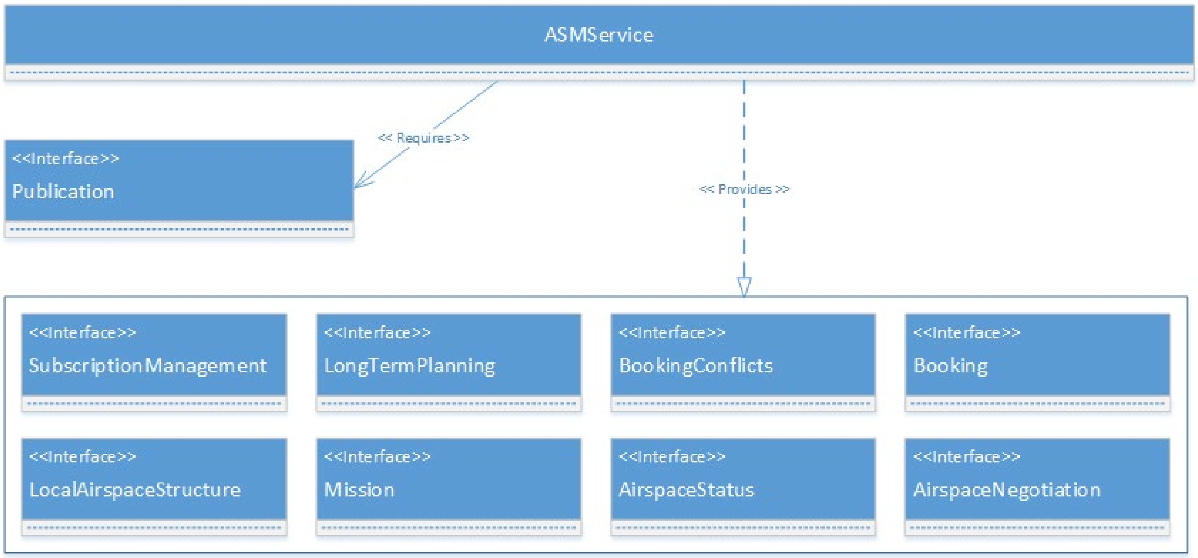


Figure 5: Service interfaces overview

2.4.2 Subscription Management Interface

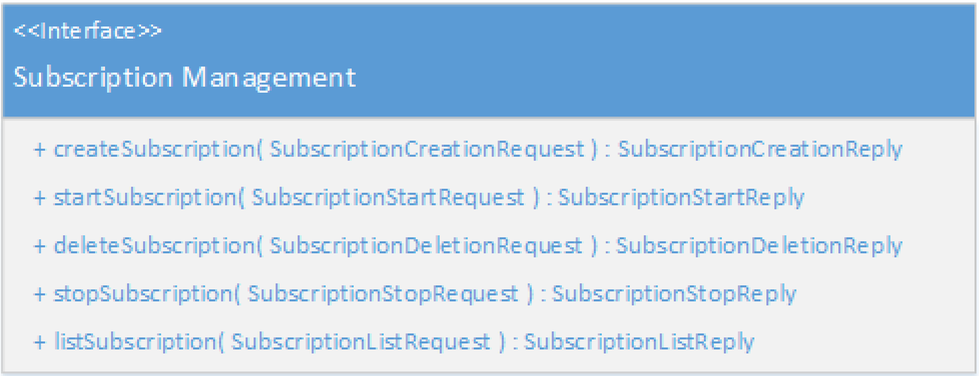

<<Interface>>
Subscription Management
+ createSubscription(SubscriptionCreationRequest) : SubscriptionCreationReply
+ startSubscription(SubscriptionStartRequest) : SubscriptionStartReply
+ deleteSubscription(SubscriptionDeletionRequest) : SubscriptionDeletionReply
+ stopSubscription(SubscriptionStopRequest) : SubscriptionStopReply
+ listSubscription(SubscriptionListRequest) : SubscriptionListReply

Figure 6: Subscription management interface operations

2.4.2.1 Interface role

The Subscription Management interface allows the management of subscriptions to the other services by External Users. External Users are able to receive the information of their interest generated by the ASM Support Systems and provided via the interfaces. Each interface applies filtering options in order to ensure data provision aligned with the requirements of the External User.

The interface allows for the creation and deletion of subscriptions to consume data from all other ASM service interfaces. A client may create a subscription to a single topic with each request but only one subscription per topic may be created at a time. In response the client will receive the name of an AMQP message queue. The name can then be supplied when creating subscriptions for additional topics to use the same queue. The client can consume the messages from the queue with the given name.

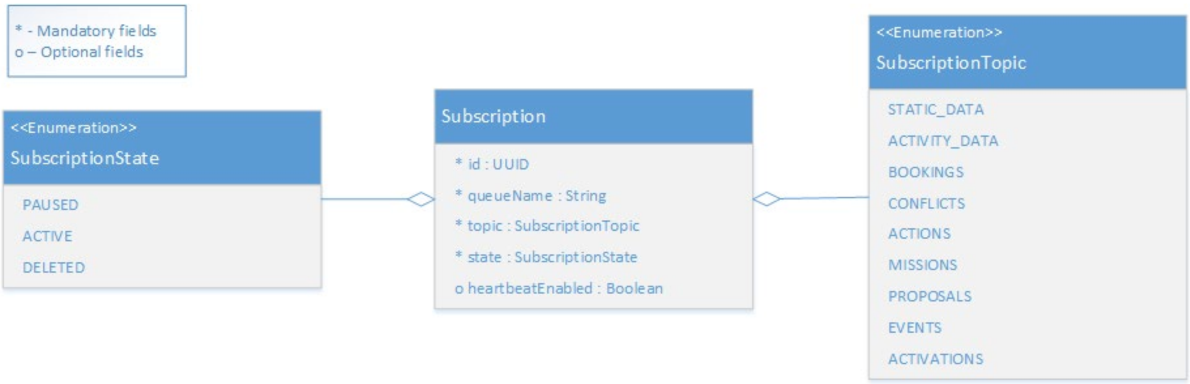


Figure 7: Subscription management interface overview

The elements in the data model are described in full detail in section 2.7.4.

2.4.2.2 Information Exchange Flow

The following diagram presents the information exchange flow for the Subscription Management interface.

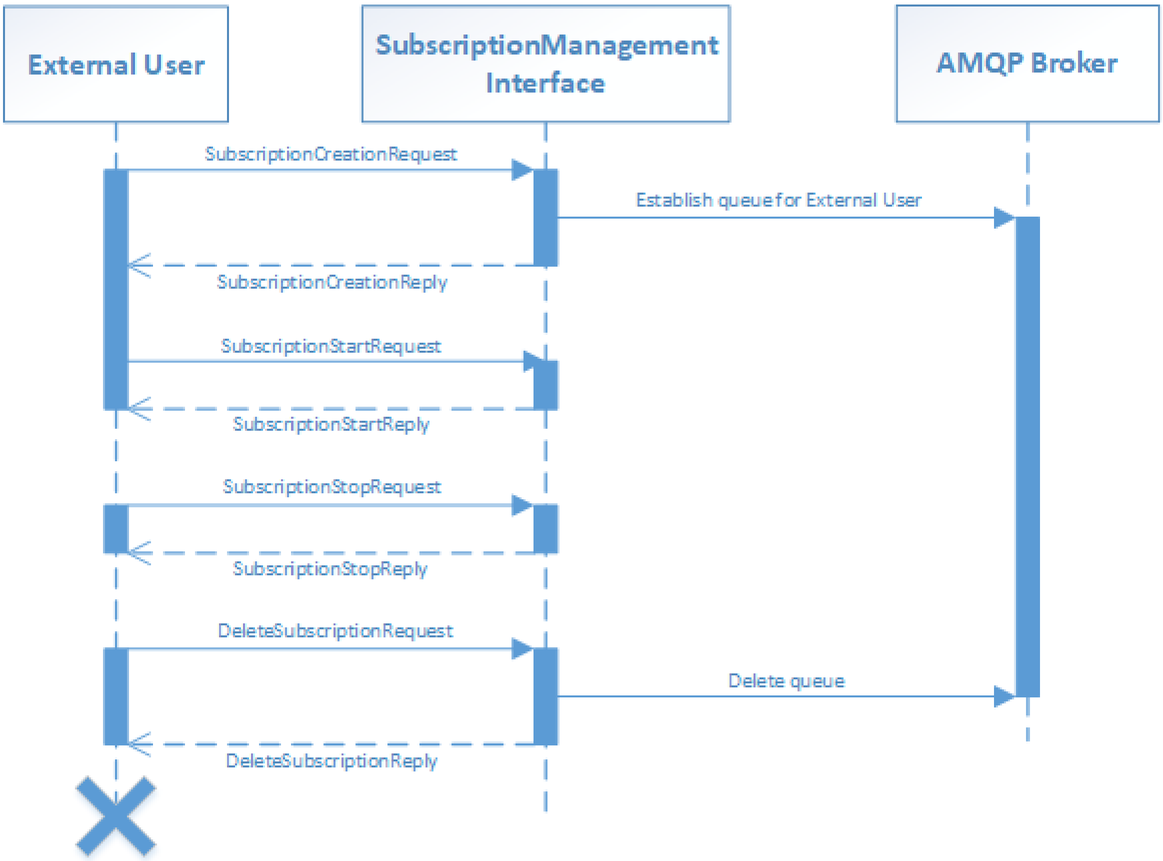


Figure 8: Subscription management interface information exchange flow

The diagram shows an External User establishing and starting a subscription, in practice this is likely to be many requests to establish and start subscriptions for each topic of interest. While the External User is subscribed, any changes in the subscribed data are notified via the AMQP queue identified in the Subscription returned as part of the subscription creation reply messages. The External User can then stop the subscription which will stop any more data being placed on the queue. Finally, the subscription can be deleted, leading to the message queue being deleted.

2.4.2.3 Message Timeouts

AMQP messages have a Time-To-Live (TTL) attribute defining the time in the future that the message will expire. This time is to be defined by the service provider and may differ between subscription topics. The TTL should be a relatively short period to limit the demand on the AMQP message broker.

If the message expires due to the TTL being reached, then the associated subscription will be paused. In this situation the client will be notified of the change of state via a Subscription Technical Message described in section 2.4.2.6.

2.4.2.4 Heartbeat Mechanism

A heartbeat mechanism is implemented as part of the subscription interface, to allow the consumer to detect failures in the service.

Depending on how often the subscribed data is changing there might be a long time between the notification of data on the queue. So that the consumer application can be aware that the connection is still alive, special technical messages are published onto the subscription queue. If a message is not received on the queue at the expected interval, then the consumer application can presume there has been some fault and act accordingly.

The special messages are called HeartbeatTechnicalMessages and are described below.

2.4.2.5 Heartbeat Technical Messages

Once a subscription is in the state ACTIVE then the client will start to receive messages related to the subscription topic along with HeartbeatTechnicalMessages.

HeartbeatTechnicalMessages are sent because the client application may experience long periods of time where there is no change resulting in the need to send a message on the queue to which the client is subscribed and therefore may question whether the subscription is still active and if the server is still functioning correctly. To overcome this issue the server sends a HeartbeatTechnicalMessage at a regular interval such that the client always receives a message within a given time period, defined by the server as the maximum-message-interval. If no message for the subscribed topic or HeartbeatTechnicalMessages are received within the maximum-message-interval, then the client application assumes that there is a fault with the service.

The HeartbeatTechnicalMessage contains a DateTime to indicate when the message was generated by the server, and a separate DateTime indicating when the server expects to send the next HeartbeatTechnicalMessage. If no subscription message is sent before then, then a HeartbeatTechnicalMessage will be sent again at that time. If a subscription message is sent to the queue before the next HeartbeatTechnicalMessage, then the next HeartbeatTechnicalMessage is postponed until maximum-message-interval time has elapsed without a message being sent.

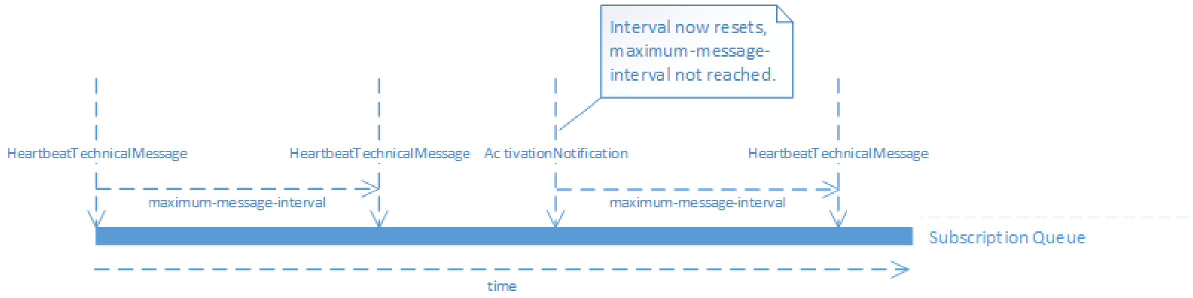

Interval now resets, maximum-message-interval not reached.
HeartbeatTechnicalMessage
HeartbeatTechnicalMessage
ActivationNotification
HeartbeatTechnicalMessage
maximum-message-interval
maximum-message-interval
Subscription Queue
time

Figure 9 Heartbeat Messages

The diagram shows that a HearbeatTechnicalMessage is received on the queue every time that maximum-message-interval has passed without a message being received.

HeartbeatTechnicalMessages can be stopped while the queue is in the PAUSED state by calling stopSubscription with heartbeatEnabled=false in the SubscriptionStopRequest. HeartbeatTechnicalMessages are always sent on ACTIVE Subscription queues.

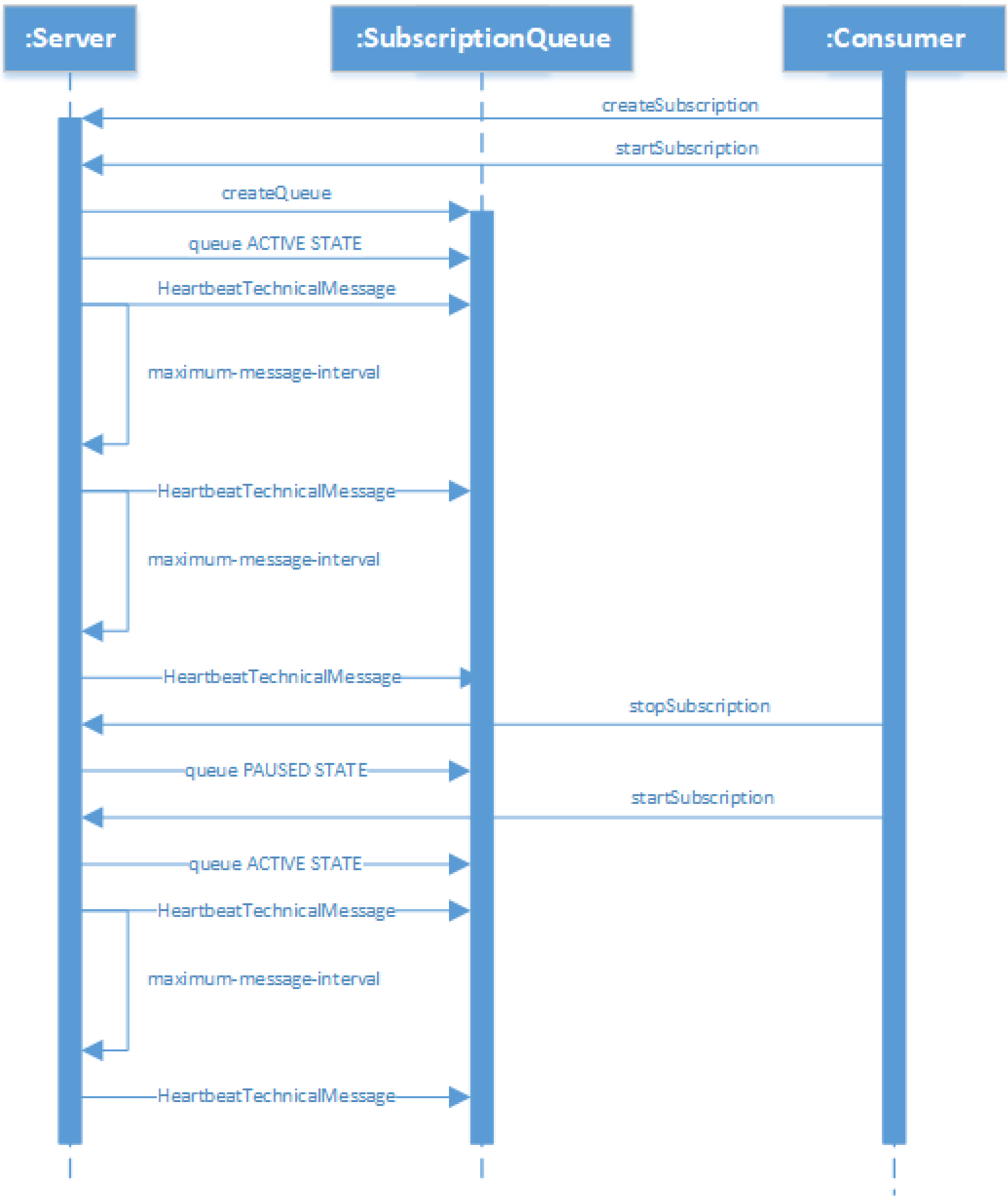


Figure 10 Pausing Heartbeats when queue state is PAUSED

2.4.2.6 Subscription Technical Messages

When a subscription is paused it is still valid but it is inactive i.e. the queue does not receive messages as the data changes. A subscription can be paused by the consumer or by the server.

If the subscription is paused by the server, then a SubscriptionTechnicalMessage is published to the subscription queue. The client can consume this message in order to be notified that the subscription has been paused. The SubscriptionTechnicalMessage contains the new state of the subscription, a dateTime to timestamp when the state of the subscription changed, and a reason to describe why the server paused the subscription.

2.4.2.7 Interface Functions

The Service interface performs the following functions:

- Creating a Subscription introduces a new Subscription into the ASM Support System.
- Starting a Subscription starts an existing Subscription in the ASM Support System.
- Stopping a Subscription stops an existing Subscription in the ASM Support System.
- Deleting a Subscription allows for deletion of an existing Subscription.
- Listing existing Subscriptions provides access to existing Subscriptions in the ASM Support System for the External User.

ASM-INTF-SUBS-010: ASM Service **shall** be supported by the Subscription Management interface to manage the subscriptions.

ASM-INTF-SUBS-020: The Subscription Management interface **shall** support the following operations:

- createSubscription,
- startSubscription,
- stopSubscription,
- deleteSubscription
- listSubscriptions

2.4.2.8 Service Operations and Associated Messages

2.4.2.8.1 ***createSubscription***

This operation is intended to create a subscription in the ASM Support System in response to a request from an External User. As a result, a subscription is either created in the local ASM Support System and the External User is notified, or a subscription is not created and an appropriate error message is transmitted to the External User.

It is down to the implementation as to whether or not subscriptions need to be remade at any other subsequent connection.

ASM-INTF-SUBS-030: The createSubscription operation **shall** receive and process the SubscriptionCreationRequest message from an External User.

ASM-INTF-SUBS-040: If no queue name is provided in the SubscriptionCreationRequest message the service **shall** assign a queue name to the External User subscription.

ASM-INTF-SUBS-050: If an active queue name is provided in the SubscriptionCreationRequest message the service **shall** reuse this queue name for the External User subscription.

***Note**: The queue name allows for connection to an Advanced Message Queuing Protocol (AMQP) message queue.*

ASM-INTF-SUBS-070: The service **shall** apply the filters defined in the SubscriptionCreationRequest to all data notified via the AMQP message queue.

ASM-INTF-SUBS-080: The createSubscription operation **shall** validate the SubscriptionCreationRequest message against the following criteria which must be met:

- All mandatory data for SubscriptionCreationRequest message are provided
- The requestor does not have an existing PAUSED or ACTIVE subscription for the same topic.
- If a queue name is set in the SubscriptionCreationRequest message it must match an existing queue name belonging to the requestor.
- The filters provided in the SubscriptionCreationRequest message must be applicable to the topic being subscribed for. Acceptable filters are defined by the query operation in this document that maps to the subscription topic.

ASM-INTF-SUBS-090: If the subscription request is valid, the createSubscription operation **shall** transmit the details of the newly created subscription, including the name of the AMQP queue to be used, in the SubscriptionCreationReply message to the requesting External User.

ASM-INTF-SUBS-100: The subscription **shall** always be created in the ‘PAUSED’ state.

ASM-INTF-SUBS-110: If the request or the resulting subscription is not valid, the createSubscription operation **shall** transmit an appropriate error in the SubscriptionCreationReply message to the requesting External User.

ASM-INTF-SUBS-115: The service provider **should** define a TTL (Time-To-Live) for messages of each subscription topic.

***Note**: The definition of these messages can be found in section 2.7.3* Interface Messages.

*2.4.2.8.2 **startSubscription***

This operation is intended to start a subscription process in the ASM Support System in response to a request from an External User. As a result, a subscription process is either started in the local ASM Support System and the External User starts receiving subscribed data, or a subscription is not started, and an appropriate error message is transmitted to the External User. Once the subscription has started HeartbeatTechnicalMessages may be sent by the ASM Support System.

ASM-INTF-SUBS-120: The startSubscription operation **shall** receive and process the SubscriptionCreationRequest message from an External User.

ASM-INTF-SUBS-130: The startSubscription operation **shall** validate the SubscriptionStartRequest message against the following criteria which must be met:

- All mandatory data for SubscriptionStartRequest message are provided
- The identified subscription to be started must belong to the requestor.
- The identified subscription to be started must be PAUSED.

ASM-INTF-SUBS-140: If the request is valid, the startSubscription operation **shall** return the SubscriptionStartReply message to the requesting External User.

ASM-INTF-SUBS-150: The SubscriptionStartReply message **shall** contain the details of the subscription, including the name of the queue to be used and the state which will be 'ACTIVE'.

ASM-INTF-SUBS-160: If the request is not valid, the startSubscription operation **shall** transmit an appropriate error in the SubscriptionStartReply message to the requesting External User.

ASM-INTF-SUBS-163: HeartbeatTechnicalMessages **should** be published at least every *maximum-message-interval* once the subscription is in state 'ACTIVE'.

ASM-INTF-SUBS-165: HeartbeatTechnicalMessages **should** be postponed if a subscription message is sent to the queue before the *maximum-message-interval* expires.

ASM-INTF-SUBS-167: The service provider **should** define a *maximum-message-interval* defining the interval between HeartbeatTechnicalMessages published to the queue.

ASM-INTF-SUBS-168: The service provider **shall** pause a subscription and publish a SubscriptionTechnicalMessage if any message published to the queue expires.

Note: The definition of these messages can be found in section 2.7.3 Interface Messages*.*

2.4.2.8.3 ***stopSubscription***

This operation is intended to stop a subscription process in the ASM Support System in response to a request from an External User. As a result, a subscription process is either stopped in the local ASM Support System and the External User stops receiving subscribed data, or a subscription is not stopped and an appropriate error message is transmitted to the External User. Once the subscription has stopped HeartbeatTechnicalMessages can be stopped by the ASM Support System.

ASM-INTF-SUBS-170: The stopSubscription operation **shall** receive and process the SubscriptionStopRequest message from an External User.

ASM-INTF-SUBS-180: The stopSubscription operation **shall** validate the SubscriptionStopRequest message against the following criteria which must be met:

- All mandatory data for SubscriptionStopRequest message are provided
- The identified subscription to be stopped must belong to the requestor.
- The identified subscription to be stopped must be ACTIVE.

ASM-INTF-SUBS-190: If the request is valid, the stopSubscription operation **shall** return the SubscriptionStopReply message to the requesting External User.

ASM-INTF-SUBS-200: The SubscriptionStopReply message **shall** contain the details of the subscription, including the name of the queue to be used and the state which will be 'PAUSED'.

ASM-INTF-SUBS-210: If the request is not valid, the stopSubscription operation **shall** transmit an appropriate error in the SubscriptionStopReply message to the requesting External User.

ASM-INTF-SUBS-215: The stopSubscription operation **may** optionally stop the HeartbeatTechnicalMessages while the subscription is in the state PAUSED by adding the attribute heartbeatEnabled with the value false to the StopSubscriptionRequest.

ASM-INTF-SUBS-217: The server **should** publish a SubscriptionTechnicalMessage to the subscription queue when the server itself has PAUSED the subscription, independent of the external user requesting to PAUSE the subscription via the stopSubscription operation.

Note: The definition of these messages can be found in section 2.7.3 Interface Messages*.*

2.4.2.8.4 ***deleteSubscription***

This operation is intended to delete a subscription in the ASM Support System in response to a request from an External User. As a result, a subscription is either deleted in the local ASM Support System and the External User is notified, or a subscription is not deleted and an appropriate error message is transmitted to the External User. If a subscription is deleted and no other subscriptions reference its message queue then the queue will be deleted.

ASM-INTF-SUBS-220: The deleteSubscription operation **shall** receive and process the SubscriptionDeletionRequest message from an External User.

ASM-INTF-SUBS-230: The deleteSubscription operation **shall** validate the SubscriptionDeletionRequest message against the following criteria which must be met:

- All mandatory data for SubscriptionDeletionRequest message are provided
- The identified subscription to be deleted must belong to the requestor.

ASM-INTF-SUBS-240: If the request is valid, the deleteSubscription operation **shall** return confirmation that the subscription has been deleted in the SubscriptionDeletionReply message to the requesting External User.

ASM-INTF-SUBS-250: If the request is valid and the subscription's queue is no longer in use by any other subscriptions, the queue **shall** be deleted.

ASM-INTF-SUBS-260: If the request is not valid, the deleteSubscription operation **shall** transmit an appropriate error in the SubscriptionDeletionReply message to the requesting External User.

2.4.2.8.5 ***listSubscription***

This operation is intended to allow the External User to list the subscriptions currently established for that specific External User.

This method returns all the subscriptions owned by the calling External User.

ASM-INTF-SUBS-270: The listSubscriptions operation **shall** receive and process the SubscriptionListRequestSubscriptionDeletionRequest message from an External User.

ASM-INTF-SUBS-280: The listSubscriptions operation **shall** validate the SubscriptionListRequest message to ensure that any subscriptionStates listed in the request are valid.

ASM-INTF-SUBS-290: If the request is valid, the listSubscriptions operation **shall** return a SubscriptionListReply containing:

- a list of subscriptions owned by the calling External User that match one of the subscription states listed in the SubscriptionListRequest,

- or all subscriptions owned by the calling External User if no subscription states were provided in the SubscriptionListRequest.

Note: The definition of these messages can be found in section 2.7.3 Interface Messages*.*

2.4.3 Publication Interface

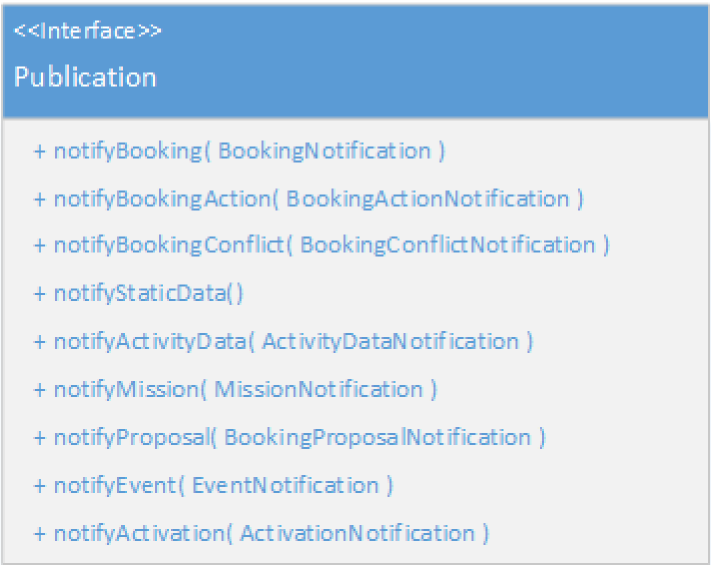

<<Interface>>
Publication
+ notifyBooking(BookingNotification)
+ notifyBookingAction(BookingActionNotification)
+ notifyBookingConflict(BookingConflictNotification)
+ notifyStaticData()
+ notifyActivityData(ActivityDataNotification)
+ notifyMission(MissionNotification)
+ notifyProposal(BookingProposalNotification)
+ notifyEvent(EventNotification)
+ notifyActivation(ActivationNotification)

Figure 11: Publication interface operations

2.4.3.1 Interface role

This section describes the expected behaviour in relation to publishing data to subscribed consumers.

The Publication interface enables the management of updates based on subscriptions by External Users. External Users are able to receive notifications of updates of the information of their interest generated by the ASM Support Systems. Each interface applies filtering options in order to ensure data provision aligned with the requirements of the External User.

This interface publishes the data that would otherwise be directly requested from other interfaces. As such the data coming via this interface is contained within the notification messages defined in section *2.7.3* Interface Messages and containing data items described by each of the other interfaces. As a result, this interface does not define its own data model.

2.4.3.2 Information Exchange Flow

The following diagram presents the information exchange flow for the Publication interface. The diagram shows a change occurring within the ASM Support System, the changed data being passed to the Publication interface which applies the filters related to each subscription and the relevant data being published to subscribed consumers.

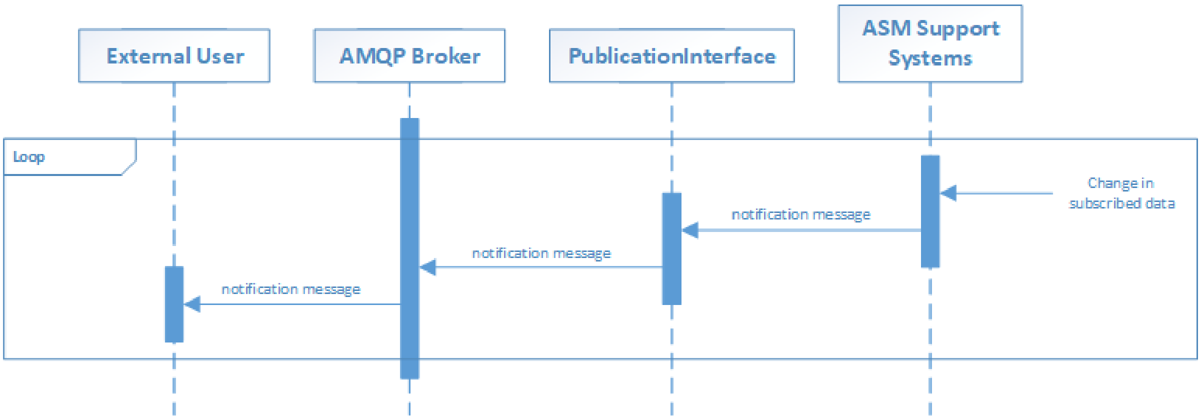


Figure 12: Publication interface information exchange flow

2.4.3.3 Interface Functions

The interface performs the following function:

- Notifying a change in the data according to the subscription topics that have been subscribed to.

ASM-INTF-PUB-010: ASM Service **shall** be supported by the Publication interface to manage the updates.

ASM-INTF-PUB-020: The Publication interface **shall** support the following operations:

- notifyStaticData
- notifyActivityData
- notifyBooking
- notifyBookingAction
- notifyBookingConflict
- notifyActivation

ASM-INTF-PUB-025: The Publication interface **should** support the following operations:

- notifyMission
- notifyProposal
- notifyEvent

ASM-INTF-PUB-030: Any data item considered for publication **shall** be published if:

- the data item was of interest to the subscriber before the change according to the filter defined at the time of subscription, or
- the data item is of interest to the subscriber post the change according to the filter defined at the time of subscription.

ASM-INTF-PUB-040: All notifications **shall** contain the full definition of the changed data item.

2.4.3.4 Service Operations and Associated Messages

*2.4.3.4.1 **notifyStaticData***

This operation is intended to notify an External User of a change in the definition of static data.

ASM-INTF-PUB-050: Any change to the definition of static data within the ASM Support System **shall** be considered for publication via an AirspaceNotification message.

*2.4.3.4.2 **notifyActivityData***

This operation is intended to notify an External User of a change in the definition of activity data.

ASM-INTF-PUB-060: Any change to the definition of activity data within the ASM Support System **shall** be considered for publication via an ActivityDataNotification message.

*2.4.3.4.3 **notifyBooking***

This operation is intended to notify an External User of a change in the definition of an existing booking.

ASM-INTF-PUB-070: Any change to the definition of a booking within the ASM Support System **shall** be considered for publication via a BookingNotification message.

*2.4.3.4.4 **notifyBookingAction***

This operation is intended to notify an External User of an action that has been performed to an existing booking.

ASM-INTF-PUB-080: Any change to the definition of action data within the ASM Support System **shall** be considered for publication via a BookingActionNotification message.

*2.4.3.4.5 **notifyBookingConflict***

This operation is intended to notify an External User of conflict data related to an existing booking.

ASM-INTF-PUB-090: Any change to the definition of booking conflict data within the ASM Support System **shall** be considered for publication via a BookingConflictNotification message.

*2.4.3.4.6 **notifyMission***

This operation is intended to notify an External User of a change in the definition of mission data.

ASM-INTF-PUB-100: Any change to the definition of mission data within the ASM Support System **shall** be considered for publication via a MissionNotification message.

*2.4.3.4.7 **notifyProposal***

This operation is intended to notify an External User of a change in the definition of proposal data.

ASM-INTF-PUB-110: Any change to the definition of proposal data within the ASM Support System **shall** be considered for publication via a BookingProposalNotification message.

2.4.3.4.8 ***notifyEvent***

This operation is intended to notify an External User of a change in the definition of event data.

ASM-INTF-PUB-120: Any change to the definition of event data within the ASM Support System **shall** be considered for publication via an EventNotification message.

2.4.3.4.9 ***notifyActivation***

This operation is intended to notify an External User of a change in the definition of activation data.

ASM-INTF-PUB-130: Any change to the definition of activation data within the ASM Support System **shall** be considered for publication via an ActivationNotification message.

Note: The definition of these messages can be found in section 2.7.3 Interface Messages*.*

2.4.4 Long Term Planning Interface

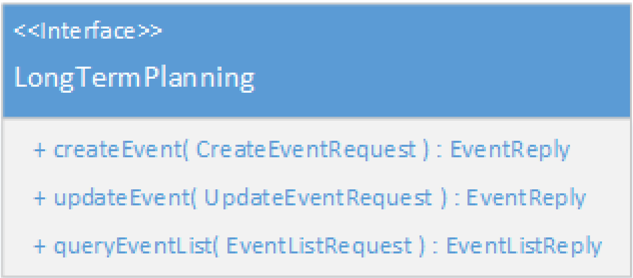

<<Interface>>
LongTermPlanning
+ createEvent(CreateEventRequest) : EventReply
+ updateEvent(UpdateEventRequest) : EventReply
+ queryEventList(EventListRequest) : EventListReply

Figure 13: Long term planning interface operations

2.4.4.1 Interface role

The Long Term Planning interface allows for the creation, modification and retrieval of long term planning data held within the ASM Support System. It is not foreseen that one ASM Support System may create long term planning in another ASM Support System and so the creation and update mechanisms may be limited to External Users acting as a user of the ASM Support System under the External ASM User - ASM Support System use of the service case (see 2.3.2).

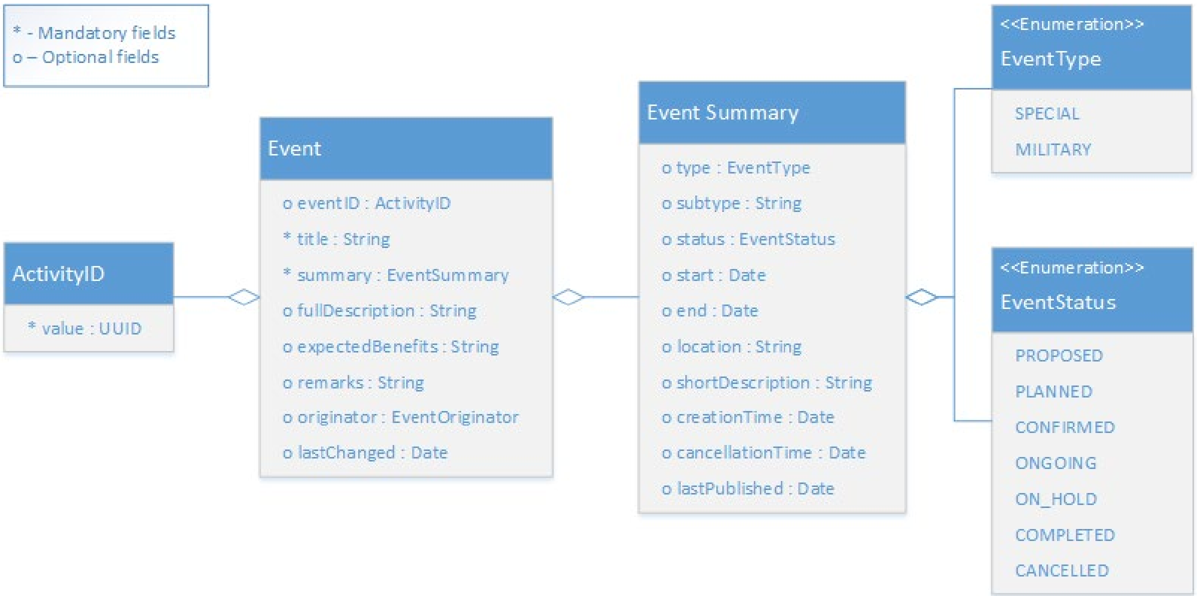


Figure 14: Long Term Planning interface overview

The elements in the data model are described in full detail in section *2.7.3* Interface Messages.

2.4.4.2 Information Exchange Flow

The following diagram presents the information exchange flow for the Long Term Planning interface.

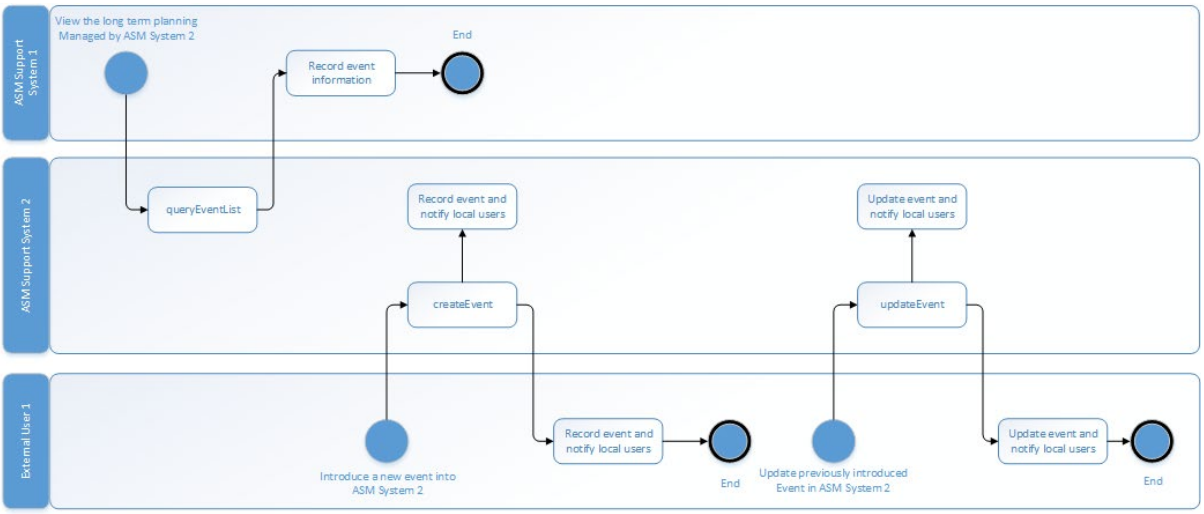

ASM Support System 1
View the long term planning Managed by ASM System 2
Record event information
End
ASM Support System 2
queryEventList
Record event and notify local users
createEvent
Update event and notify local users
updateEvent
External User 1
Introduce a new event into ASM System 2
Record event and notify local users
End
Update previously introduced Event in ASM System 2
Update event and notify local users
End

Figure 15: Long Term Planning interface exchange flow

2.4.4.3 Interface Functions

The interface performs the following functions:

- Creating an Event - introduces a new Event into the ASM Support System
- Updating an Event - updates an existing Event in the ASM Support System
- Requesting a List of Events - allows access to the Event information from within the ASM Support System

ASM-INTF-LTPL-010: ASM Service **should** be supported by the Long Term Planning interface to manage the events.

ASM-INTF-LTPL-020: The Long Term Planning interface **shall** support the following operations:

- createEvent,
- updateEvent,
- queryEventList

2.4.4.4 Service Operations and Associated Messages

*2.4.4.4.1 **createEvent***

This operation is intended to introduce a new Event into the ASM Support System.

ASM-INTF-LTPL-030: The createEvent operation **shall** receive and process the EventCreationRequest message from an External User.

ASM-INTF-LTPL-040: If the event request is valid, the createEvent operation **shall** transmit the newly created event in the EventReply message to the requesting External Users.

ASM**-INTF-LTPL-050:** If for any reason the request or the resulting event is not valid, the createEvent operation **shall** transmit an appropriate error in the EventReply message to the requesting External Users.

2.4.4.4.2 ***updateEvent***

This operation is intended to update an existing Event in the ASM Support System.

ASM-INTF-LTPL-060: The updateEvent operation **shall** receive and process the EventUpdateRequest message from an External User.

ASM-INTF-LTPL-070: If the event update is valid, the updateEvent operation **shall** transmit the updated event in the EventReply message to the requesting External User.

ASM-INTF-LTPL-080: If for any reason the request or the resulting updated event is not valid, the updateEvent operation **shall** transmit an appropriate error in the EventReply message to the requesting External User.

2.4.4.4.3 ***queryEventList***

This operation is intended to allow access to the Event information from within the ASM Support System.

ASM-INTF-LTPL-090: The queryEventList operation **shall** receive and process the FilteredRequest message from an External User.

ASM-INTF-LTPL-100: If the request is valid, the queryEventList operation **shall** transmit the matching list of events in the EventListReply message to the requesting External User.

ASM-INTF-LTPL-110: If for any reason the request is not valid, the queryEventList operation **shall** transmit an appropriate error in the EventListReply message to the requesting External Users.

ASM-INTF-LTPL-120: The queryEventList operation **shall** accept any combination of the following filters in the FilteredRequest message:

- ActivityIDFilter
- ChangePeriodFilter
- InterestedIntervalFilter
- AndFilter

Note: The definition of these messages can be found in section 2.7.3 Interface Messages*.*

2.4.5 Local Airspace Structure Interface

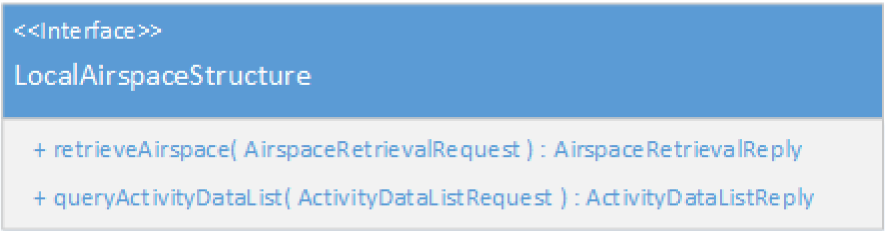

<<Interface>>
LocalAirspaceStructure
+ retrieveAirspace(AirspaceRetrievalRequest) : AirspaceRetrievalReply
+ queryActivityDataList(ActivityDataListRequest) : ActivityDataListReply

Figure 16: Local airspace structure interface operations

2.4.5.1 Interface role

The Local Airspace Structure interface allows for retrieval of airspace structures and activity data from within the ASM Support System by External Users.

Local ASM Support Systems maintain a local database that contains national airspace structures. The interface allows External Users to obtain the definitions of the airspace structures, provided by the providing system in accordance with the provider requirements, as well as activity data, in order to follow their status and the reservations associated to them.

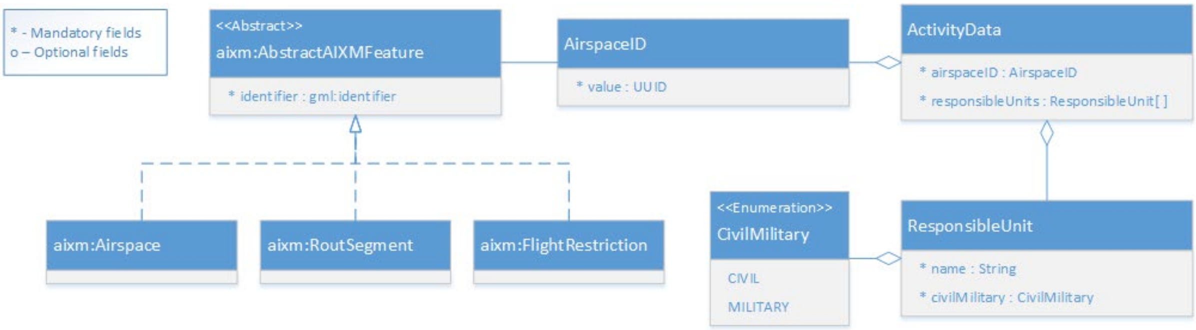


Figure 17: Local Airspace Structure interface overview

The elements in the data model are described in full detail in section 2.7.3 Interface Messages.

2.4.5.2 Information Exchange Flow

The following diagram presents the information exchange flow for the Local Airspace Structure interface.

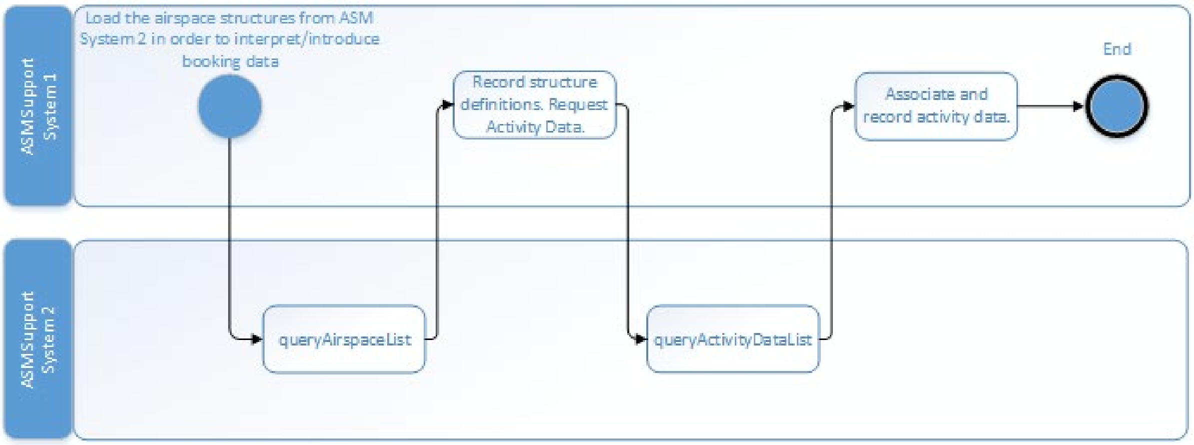

ASM Support System 1
ASM Support System 2
Load the airspace structures from ASM System 2 in order to interpret/introduce booking data
Record structure definitions. Request Activity Data.
Associate and record activity data.
End
queryAirspaceList
queryActivityDataList

Figure 18: Local Airspace Structure interface information exchange flow

2.4.5.3 Interface Functions

The interface performs the following functions:

- Requesting Static Data - allows access to static data from within the ASM Support System
- Requesting Activity Data – allows access to activity data from within the ASM Support System

ASM-INTF-LAS-010: ASM Service **shall** be supported by the Local Airspace Structure interface to manage the access to local airspace static and activity data.

ASM-INTF-LAS-020: The LocalAirspaceStructure interface **shall** support the following operations:

- retrieveAirspace
- queryActivityDataList

2.4.5.4 Service Operations and Associated Messages

*2.4.5.4.1 **queryAirspace***

This operation is intended to manage the access to static data from the ASM Support System.

ASM-INTF-LAS-030: The queryAirspaceList operation **shall** receive and process the FilteredRequest message from an External User.

ASM-INTF-LAS-040: If the request is valid, the queryAirspaceList operation **shall** transmit the matching static data in the AirspaceListReply message to the requesting External User.

ASM-INTF-LAS-050: If for any reason the request is not valid, the queryAirspaceList operation **shall** transmit an appropriate error in the AirspaceListReply message to the requesting External User.

ASM-INTF-LAS-055: The queryAirspaceList operation **shall** accept any combination of the following filters in the FilteredRequest message:

- AirspaceIDFilter
- AirspaceNameFilter
- AirspaceTypeFilter
- GeometryFilter
- AndFilter
- ChangePeriodFilter
- InterestedIntervalFilter

*2.4.5.4.2 **queryActivityDataList***

This operation is intended to manage the access to activity data from the ASM Support System.

ASM-INTF-LAS-060: The queryActivityDataList operation **shall** receive and process the FilteredRequest message from an External User.

ASM-INTF-LAS-070: If the request is valid, the queryActivityDataList operation **shall** transmit the matching activity data list in the ActivityDataListReply message to the requesting External User.

ASM-INTF-LAS-080: If for any reason the request is not valid, the queryActivityDataList operation **shall** transmit an appropriate error in the ActivityDataListReply message to the requesting External User.

ASM-INTF-LAS-085: The queryActivityDataList operation **shall** accept any combination of the following filters in the FilteredRequest message:

- AirspaceIDFilter

Note: The definition of these messages can be found in section 2.7.3 Interface Messages*.*

2.4.6 Booking Interface

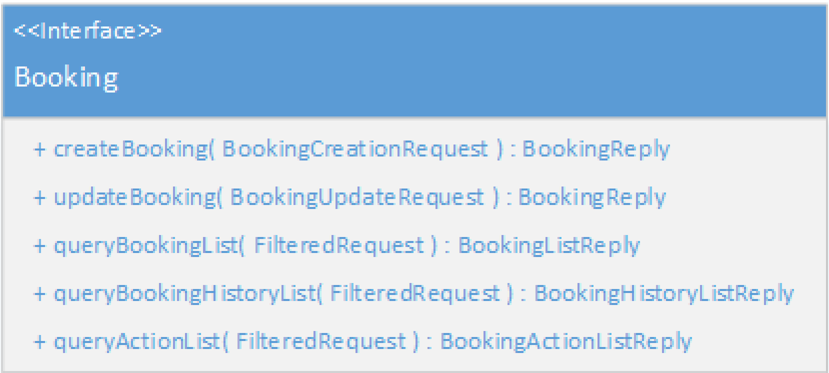

<<Interface>>
Booking
+ createBooking(BookingCreationRequest) : BookingReply
+ updateBooking(BookingUpdateRequest) : BookingReply
+ queryBookingList(FilteredRequest) : BookingListReply
+ queryBookingHistoryList(FilteredRequest) : BookingHistoryListReply
+ queryActionList(FilteredRequest) : BookingActionListReply

Figure 19: Booking interface operations

2.4.6.1 Interface role

Booking is an interface that enables the creation, modification and retrieval of bookings, including details of conflicts between different bookings within the ASM Support System by External Users. The interface allows for continuous updates in real time of Booking information among authorised ASM Support Systems/External Users in order to enhance cross border coordination based on the most recent information.

The interface also enables booking of airspace structures across national borders; airspace users are able to book foreign airspace using their local ASM Support System. The process of a foreign booking follows the ASM process defined in dedicated LoAs between States.

The creation and modification of bookings is supported through the provision of ‘Actions’ by the interface. The Actions describe the allowed modifications that a user may take on a specific booking. Actions are subject to change based on the state of a booking and the current time.

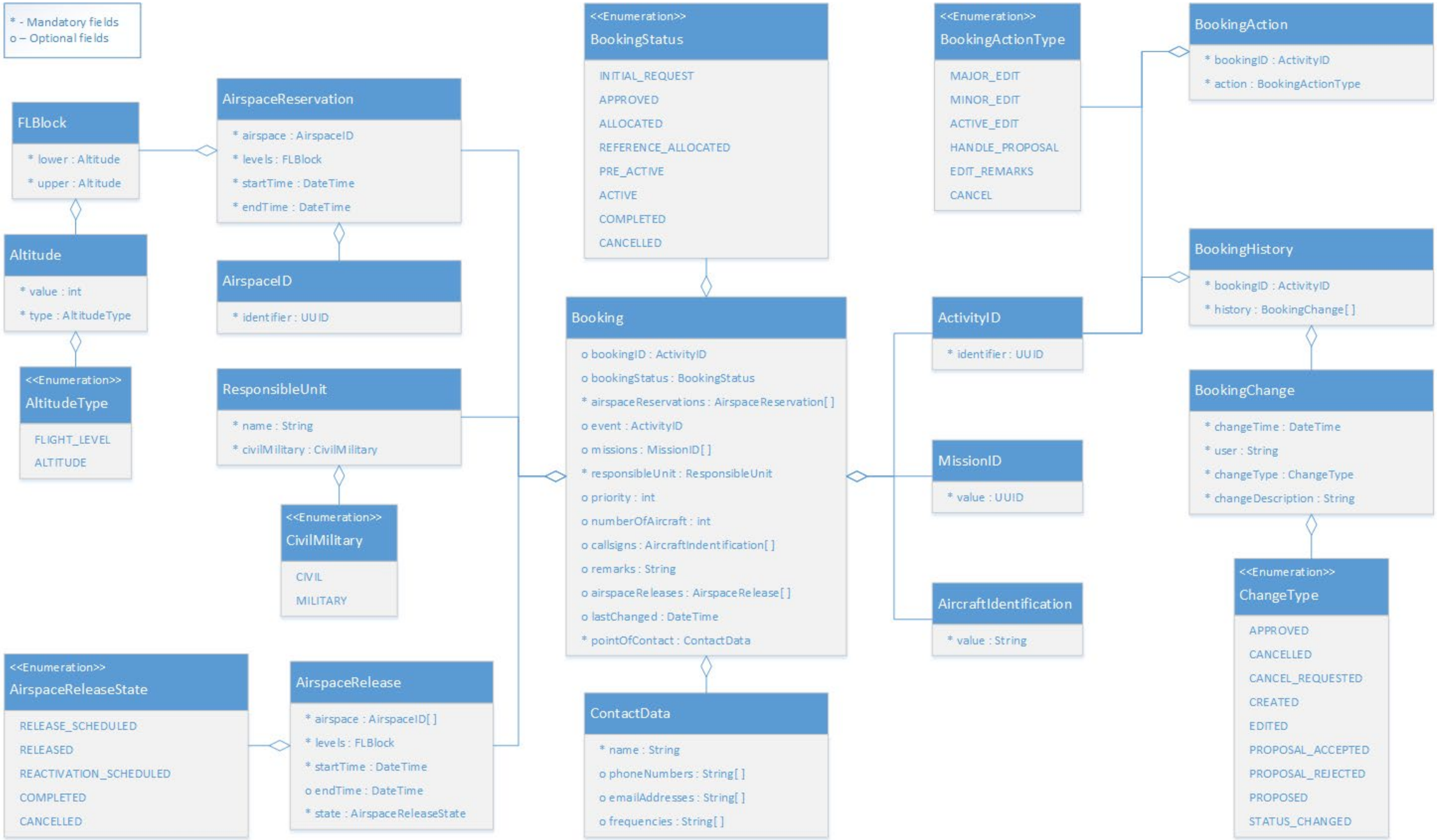


Figure 20: Booking interface overview

The elements in the data model are described in full detail in section 2.7.3 Interface Messages.

2.4.6.2 Information Exchange Flow

The following diagram presents the information exchange flow for the Booking interface.

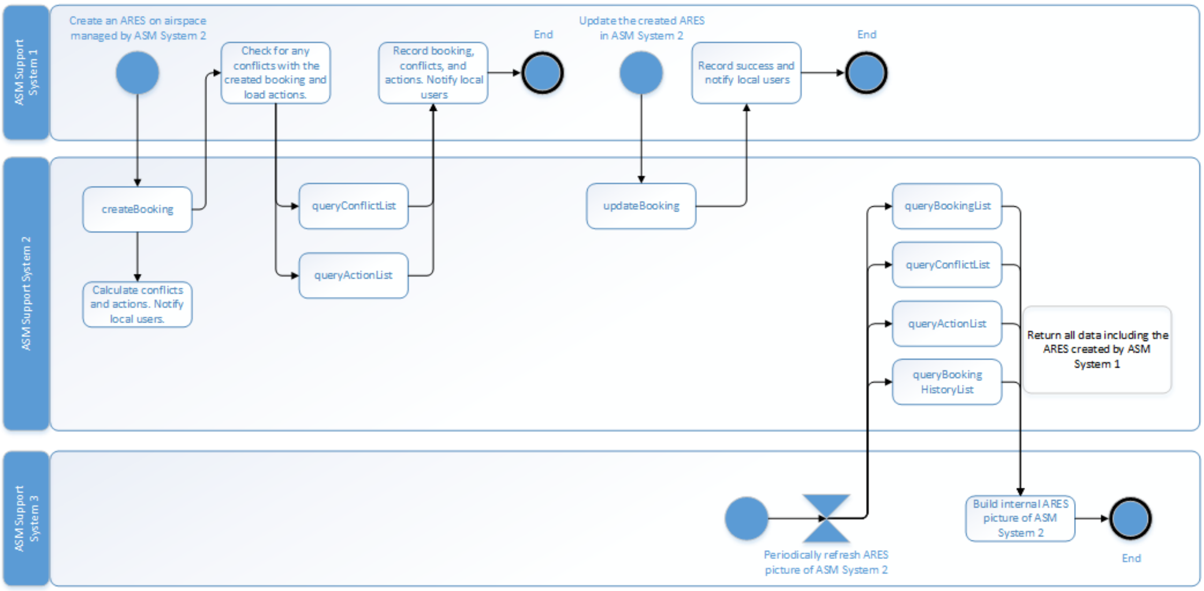

ASM Support System 1
Create an ARES on airspace managed by ASM System 2
Check for any conflicts with the created booking and load actions.
Record booking, conflicts, and actions. Notify local users
End
Update the created ARES in ASM System 2
Record success and notify local users
End
ASM Support System 2
createBooking
Calculate conflicts and actions. Notify local users.
queryConflictList
queryActionList
updateBooking
queryBookingList
queryConflictList
queryActionList
queryBooking HistoryList
Return all data including the ARES created by ASM System 1
ASM Support System 3
Periodically refresh ARES picture of ASM System 2
Build internal ARES picture of ASM System 2
End

Figure 21: Booking interface information exchange flow

The diagram shows three ASM Support Systems with ASM Support System 2 offering a Booking interface. ASM Support System 1 creates a booking on airspace managed by ASM Support System 2 and then looks for any conflicts and any actions that ASM Support System 2 has calculated for it. At a later time ASM Support System 1 uses the actions that it previously retrieved to update its booking in ASM Support System 2.

ASM Support System 3 is making use of the request/reply interfaces to periodically synchronise with the data in ASM Support System 2. Requesting all bookings, actions and booking history.

2.4.6.3 Interface Functions

The interface performs the following functions:

- Creating a Booking introduces a new Booking into the ASM Support System to be approved to be incorporated into the plan.
- Updating a Booking updates an existing Booking in the ASM Support System potentially as a result of a CDM process or simply to update the plan to reflect changes to planned activities.
- Requesting Booking List allows access to the Booking information from within the ASM Support System to allow for CDM processes.
- Requesting Actions allows access to the Actions that can performed by the actor. Identifying where they may contribute to the CDM process within the ASM Support System and which Airspace Reservation they can update to better inform the plan.
- Requesting History allows access to the history of all actions that have been performed on a Booking.

ASM-INTF-ARES-010: ASM Service **shall** be supported by the Booking interface to manage the reservations.

ASM-INTF-ARES-020: The Booking interface **shall** implement synchronous Request-Reply application message exchange pattern.

ASM-INTF-ARES-030: The Booking interface **shall** support the following operations:

- createBooking,
- updateBooking,
- queryBookingList
- queryActionList
- queryBookingHistoryList

2.4.6.4 Service Operations and Associated Messages

2.4.6.4.1 ***createBooking***

This operation is intended to introduce a new booking in the ASM Support System in response to a request from an External User. As a result, the new booking is either created in the local ASM Support System and the External User is notified, or the booking is not created and an appropriate error message is transmitted to the External User.

ASM-INTF-ARES-040: The createBooking operation **shall** receive and process the BookingCreationRequest message from an External User.

ASM-INTF-ARES-050: The createBooking operation **shall** validate the BookingCreationRequest message against the following criteria which must be met:

- All mandatory data for the BookingCreationRequest message are provided
- The booking ID must be null/not set
- The start date and time of all AirspaceReservations must be in the future
- The end date and time of an AirspaceReservation must be after its start date and time
- The External User must have permission to book the airspace structures as defined by the ActivityData.
- The entire booked period must be within the life time of all booked airspace structures
- The booked flight levels must be within or equal to the flight level bounds of the airspace structures
- The booked lower flight level of an airspace structure must be below the booked upper flight level of the airspace structure
- The selected responsible unit must be common to all booked airspace structures as defined by the ActivityData.

ASM-INTF-ARES-060: If the booking is valid, the createBooking operation **shall** transmit the newly created booking in the BookingReply message to the requesting External User.

ASM-INTF-ARES-070: If the request or the resulting booking is not valid, the createBooking operation **shall** transmit an appropriate error in the BookingReply message to the requesting External User.

Note: The definition of these messages can be found in section 2.7.3 Interface Messages.

2.4.6.4.2 *updateBooking*

This operation is intended to introduce updates to an existing booking in the ASM Support System in response to a request from an External User. As a result, the booking is either updated in the local ASM Support System and the External User is notified, or the booking is not updated and an appropriate error message is transmitted to the External User.

ASM-INTF-ARES-080: The updateBooking operation **shall** receive and process the BookingUpdateRequest message from an External User.

ASM-INTF-ARES-090: The updateBooking operation **shall** validate the BookingUpdateRequest message against the following criteria which must be met:

- All mandatory data for the BookingUpdateRequest message are provided
- The booking ID must be set
- The lastChangeTime in the booking must match that held by the service
- The start date and time of all AirspaceReservations must be in the future; if the booking is active the start dates and times must match the original start dates and times and the end time should be in the future
- The end date and time of an AirspaceReservation must be after its start date and time
- The External User must have permission to book the airspace structures as defined by the ActivityData
- The entire booked period must be within the life time of all booked airspace structures
- The booked flight levels must be within or equal to the flight level bounds of the airspace structures
- The booked lower flight level of an airspace structure must be below the booked upper flight level of the airspace structure
- The selected responsible unit must be common to all booked airspace structures as defined by the ActivityData

ASM-INTF-ARES-100: If the booking update is valid, the updateBooking operation **shall** transmit the updated booking in the BookingReply message to the requesting External User.

ASM-INTF-ARES-110: If the request or the resulting updated booking is not valid, the updateBooking operation **shall** transmit an appropriate error in the BookingReply message to the requesting External User.

ASM-INTF-ARES-120: An updateBookingRequest **should** be rejected by the service if the user performing the update does not have the appropriate actions matching their update.

Note: The definition of these messages can be found in section 2.7.3 Interface Messages*.*

2.4.6.4.3 *queryBookingList*

This operation is intended to introduce a query for the list of bookings in the ASM Support System in response to a request from an External User. As a result, the list of bookings is either transmitted to the External User or not, and an appropriate error message is transmitted to the External User.

ASM-INTF-ARES-130: The queryBookingList operation **shall** receive and process the FilteredRequest message from an External User.

ASM-INTF-ARES-140: The queryBookingList operation **shall** validate the FilteredRequest message against the following criteria which must be met:

- All mandatory data for FilteredRequest message are provided

ASM-INTF-ARES-150: If the request is valid, the queryBookingList operation **shall** transmit the filtered list of bookings in the BookingListReply message to the requesting External User.

ASM-INTF-ARES-160: If the request is not valid, the queryBookingList operation **shall** transmit an appropriate error in the BookingListReply message to the requesting External User.

ASM-INTF-ARES-170: The queryBookingList operation **shall** accept any combination of the following filters in the FilteredRequest message:

- ActivityIDFilter
- ChangePeriodFilter
- InterestedIntervalFilter
- AndFilter

ASM-INTF-ARES-180: The queryBookingList operation **should** accept any combination of the following filters in the FilteredRequest message:

- AirspaceIDFilter
- GeometryFilter
- MissionIDFilter

Note: The definition of these messages can be found in section 2.7.3 Interface Messages.

*2.4.6.4.4 **queryActionList***

This operation is intended to introduce a query for the list of actions in the ASM Support System in response to a request from an External User. As a result, the list of actions is either transmitted to the External User or not, and an appropriate error message is transmitted to the External User.

ASM-INTF-ARES-190: The queryActionList operation **shall** receive and process the FilteredRequest message from an External User.

ASM-INTF-ARES-200: The queryActionList operation **shall** validate the FilteredRequest message against the following criteria which must be met:

- All mandatory data for the FilteredRequest message are provided

ASM-INTF-ARES-210: If the request is valid, the queryActionList operation **shall** transmit the list of actions in the BookingActionListReply message to the requesting External User.

ASM-INTF-ARES-220: If the request is not valid, the queryActionList operation **shall** transmit an appropriate error in the BookingActionListReply message to the requesting External User.

ASM-INTF-ARES-230: The queryActionList operation **shall** accept the following filter in the FilteredRequest:

- ActivityIDFilter

Note: The definition of these messages can be found in section 2.7.3 Interface Messages*.*

2.4.6.4.5 ***queryBookingHistoryList***

This operation is intended to introduce a query for the list of history of actions performed on a booking in the ASM Support System in response to a request from an External User. As a result, the list of history of actions performed on a booking is either transmitted to the External User or not, and an appropriate error message is transmitted to the External User.

ASM-INTF-ARES-240: The queryBookingHistoryList operation **shall** receive and process the FilteredRequest message from an External User.

ASM-INTF-ARES-250: The queryBookingHistoryList operation **shall** validate the FilteredRequest message against the following criteria which must be met:

- All mandatory data for the FilteredRequest message are provided

ASM-INTF-ARES-260: If the request is valid, the queryBookingHistoryList operation **shall** transmit the complete list of history of actions performed on a booking in the BookingHistoryListReply message to the requesting External User.

ASM-INTF-ARES-270: If the request is not valid, the queryBookingHistoryList operation **shall** transmit an appropriate error in the BookingHistoryListReply message to the requesting External User.

ASM-INTF-ARES-280: The queryBookingHistoryList operation **shall** accept the following filter in the FilteredRequest message:

- ActivityIDFilter

Note: The definition of these messages can be found in section 2.7.3 Interface Messages*.*

2.4.7 Booking Conflicts Interface

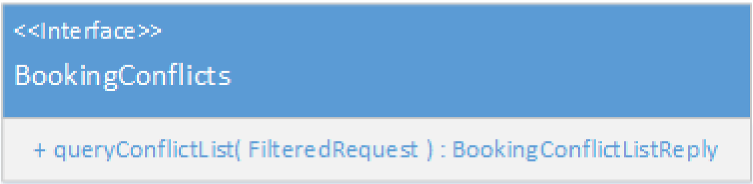

<<Interface>>
BookingConflicts
+ queryConflictList(FilteredRequest) : BookingConflictListReply

Figure 22: Booking conflicts interface operations

2.4.7.1 Interface role

Requesting Conflict List allows access to the Booking conflict information from within the ASM Support System to enhance CDM processes.

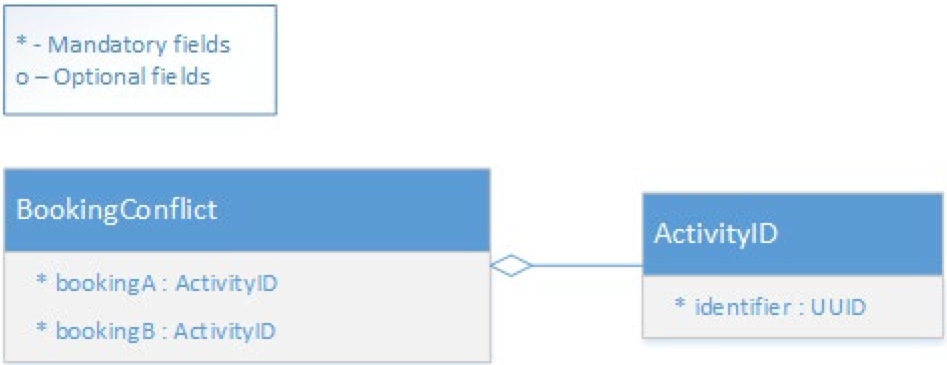


Figure 23: Booking conflicts interface overview

The elements in the data model are described in full detail in section 2.7.3 Interface Messages.

2.4.7.2 Information Exchange Flow

The following diagram presents the information exchange flow for the Local Airspace Structure interface.

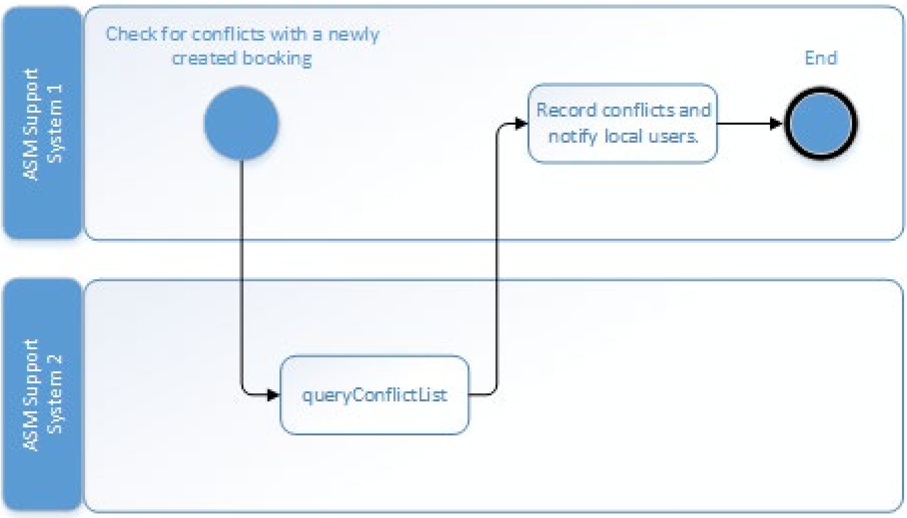

ASM Support System 1
Check for conflicts with a newly created booking
Record conflicts and notify local users.
End
ASM Support System 2
queryConflictList

Figure 24: Booking conflicts interface information exchange flow

The diagram above shows ASM Support System 1, having recently created a booking in ASM Support System 2, query for any conflict information that has been generated for the booking. Once any conflicts have been calculated they can be presented to the users of ASM Support System 1 who can decide whether or not they will modify their booking for it to be approved by ASM Support System 1.

2.4.7.3 Interface Functions

The interface performs the following functions:

- Requesting booking conflict data – allows access to booking conflict data from within the ASM Support System

ASM-INTF-CON-010: ASM Service **should** be supported by the Booking Conflicts interface to manage booking conflicts.

ASM-INTF-CON-020: The Booking Conflicts interface **shall** support the following operation:

- queryConflictList

2.4.7.4 Service Operations and Associated Messages

2.4.7.4.1 ***queryConflictList***

This operation is intended to introduce a query for the list of conflicts between bookings in the ASM Support System in response to a request from an External User. As a result, the list of conflicts between bookings is either transmitted to the External User or not, and an appropriate error message is transmitted to the External User.

ASM-INTF-CON-030: The queryConflictList operation **shall** receive and process the FilteredRequest message from an External User.

ASM-INTF-CON-040: The queryConflictList operation **shall** validate the FilteredRequest message against the following criteria which must be met:

- All mandatory data for the FilteredRequest message are provided

ASM-INTF-CON-050: If the request is valid, the queryConflictList operation **shall** transmit the list of conflicts in the BookingConflictListReply message to the requesting External User.

ASM-INTF-CON-060: If the request is not valid, the queryConflictList operation **shall** transmit an appropriate error in the BookingConflictListReply message to the requesting External User.

ASM-INTF-CON-070: The queryConflictList operation **shall** accept any combination of the following filters in the FilteredRequest:

- ActivityIDFilter
- AndFilter

Note: The definition of these messages can be found in section 2.7.3 Interface Messages.

2.4.8 Mission Interface

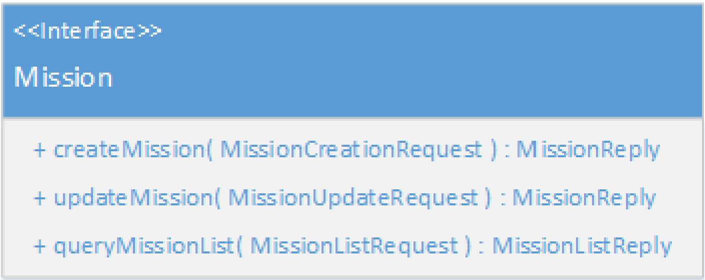

<<Interface>>
Mission
+ createMission(MissionCreationRequest) : MissionReply
+ updateMission(MissionUpdateRequest) : MissionReply
+ queryMissionList(MissionListRequest) : MissionListReply

Figure 25: Mission interface operations

2.4.8.1 Interface role

The Mission interface allows for the creation, modification and retrieval of missions held within the ASM Support System by External Users. This interface allows exchange of mission information between host and foreign ASM Support Systems.

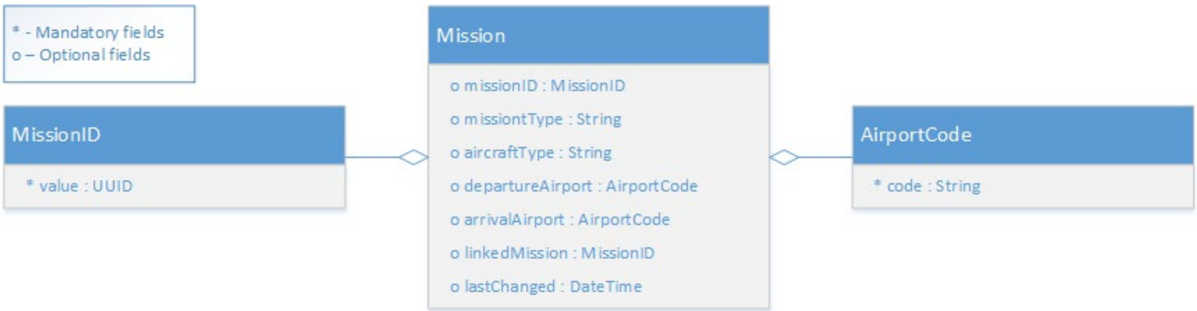


Figure 26: Mission interface overview

The elements in the data model are described in full detail in section [2.7.3](#) Interface Messages.

2.4.8.2 Information Exchange Flow

The following diagram presents the information exchange flow for the Mission service.

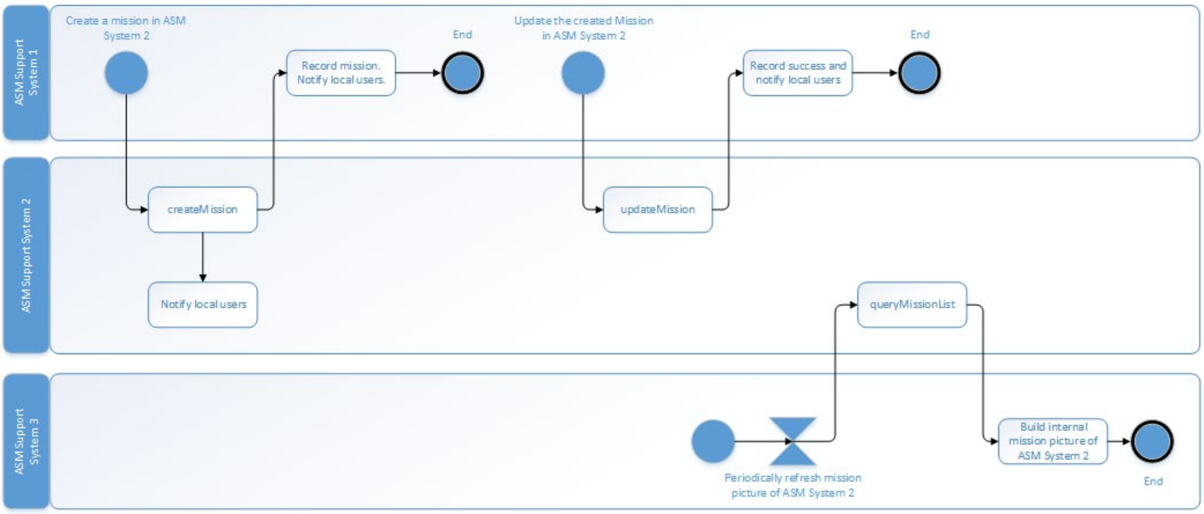

ASM Support System 1
ASM Support System 2
ASM Support System 3
Create a mission in ASM System 2
Record mission. Notify local users.
End
Update the created Mission in ASM System 2
Record success and notify local users
End
createMission
Notify local users
updateMission
queryMissionList
Periodically refresh mission picture of ASM System 2
Build internal mission picture of ASM System 2
End

Figure 27: Mission interface information exchange flow

ASM Support System 1 is shown as having two clients, ASM Service Client (i.e. External User) and ASM Support System 2. External User creates a mission within the ASM Support System 1 via the createMission operation. The mission is returned to the External User which presents the mission to its users. At a later time, External User makes a change to its previously created mission via the updateMission operation. The update succeeds and again is presented to its users. At the same time ASM Support System 2 is polling ASM Support System 1 for its mission information via the queryMissionList operation which it presents to its users.

2.4.8.3 Interface Functions

The interface performs the following functions:

- Creating a Mission - introduces a new Mission into the ASM Support System to be approved to be incorporated into the plan.
- Updating a Mission - updates an existing Mission in the ASM Support System potentially as a result of a CDM process or simply to update the plan to reflect real world changes to planned activities.
- Requesting Missions - allows access to the Missions information from within the ASM Support System to allow for CDM processes.

ASM-INTF-MIS-010: ASM Service **should** be supported by the Mission interface to manage the exchange of mission information.

ASM-INTF-MIS-020: The Mission interface **shall** support the following operation:

- createMission,
- updateMission,
- queryMissionList

2.4.8.4 Service Operations and Associated Messages

*2.4.8.4.1 **createMission***

ASM-INTF-MIS-030: The createMission operation **shall** receive and process the MissionCreationRequest message from an External User.

ASM-INTF-MIS-040: If the mission is valid, the createMission operation **shall** transmit the newly created mission in the MissionReply message to the requesting External Users.

ASM-INTF-MIS-050: If for any reason the request or the resulting mission is not valid, the createMission operation **shall** transmit an appropriate error in the MissionReply message to the requesting External Users.

Note: The definition of these can be found in section 2.7.3 Interface Messages*.*

*2.4.8.4.2 **updateMission***

ASM-INTF-MIS-060: The updateMission operation **shall** receive and process the MissionUpdateRequest message from an External User.

ASM-INTF-MIS-070: If the mission update is valid, the updateMission operation **shall** transmit the updated mission in the MissionReply message to the requesting External Users.

ASM-INTF-MIS-080: If for any reason the request or the resulting updated mission is not valid, the updateMission operation **shall** transmit an appropriate error in the MissionReply message to the requesting External Users.

ASM-INTF-MIS-090: A MissionUpdateRequest **should** be rejected by the service if the user performing the update does not have the appropriate actions matching their update.

Note: The definition of these can be found in section 2.7.3 Interface Messages*.*

*2.4.8.4.3 **queryMissionList***

ASM-INTF-MIS-100: The queryMissionList operation **shall** receive and process the FilteredRequest message from an External User.

ASM-INTF-MIS-110: If the request is valid, the queryMissionList operation **shall** transmit the list of missions in the MissionListReply message to the requesting External Users.

ASM-INTF-MIS-120: If for any reason the request is not valid, the queryMissionList operation **shall** transmit an appropriate error in the MissionListReply message to the requesting External Users.

ASM-INTF-MIS-130: The queryMissionList operation **shall** accept any combination of the following filters in the FilteredRequest message:

- MissionIDFilter
- ActivityIDFilter
- ChangePeriodFilter
- InterestedIntervalFilter
- AndFilter

Note: The definition of these messages can be found in section 2.7.3 Interface Messages*.*

2.4.9 Airspace Negotiation Interface

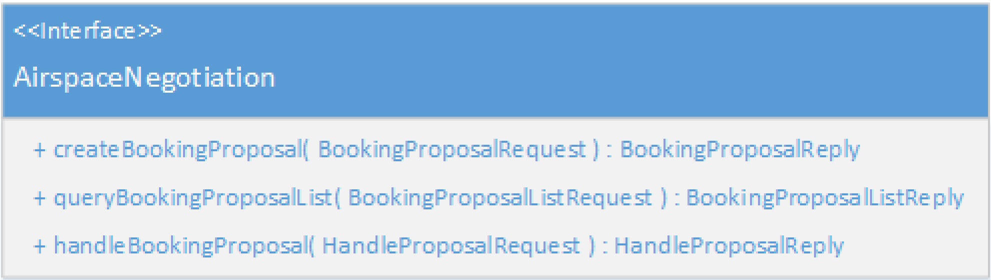

<<Interface>>
AirspaceNegotiation
+ createBookingProposal(BookingProposalRequest) : BookingProposalReply
+ queryBookingProposalList(BookingProposalListRequest) : BookingProposalListReply
+ handleBookingProposal(HandleProposalRequest) : HandleProposalReply

Figure 28: Airspace negotiation interface operations

2.4.9.1 Interface role

The Airspace Negotiation interface enables the retrieval and handling of proposals held within the ASM Support System by External Users.

A proposal refers to a specific reservation and effectively re-defines a subset of the reservation data. The Negotiation interface enables a proposal to be accepted or rejected as per the actions accessible through the service.

The Negotiation interface allows users managing the airspace to make proposals to booking requests introduced by foreign airspace requesters. The proposals could address the FLs, times, or even different area. These proposals could be accepted or rejected.

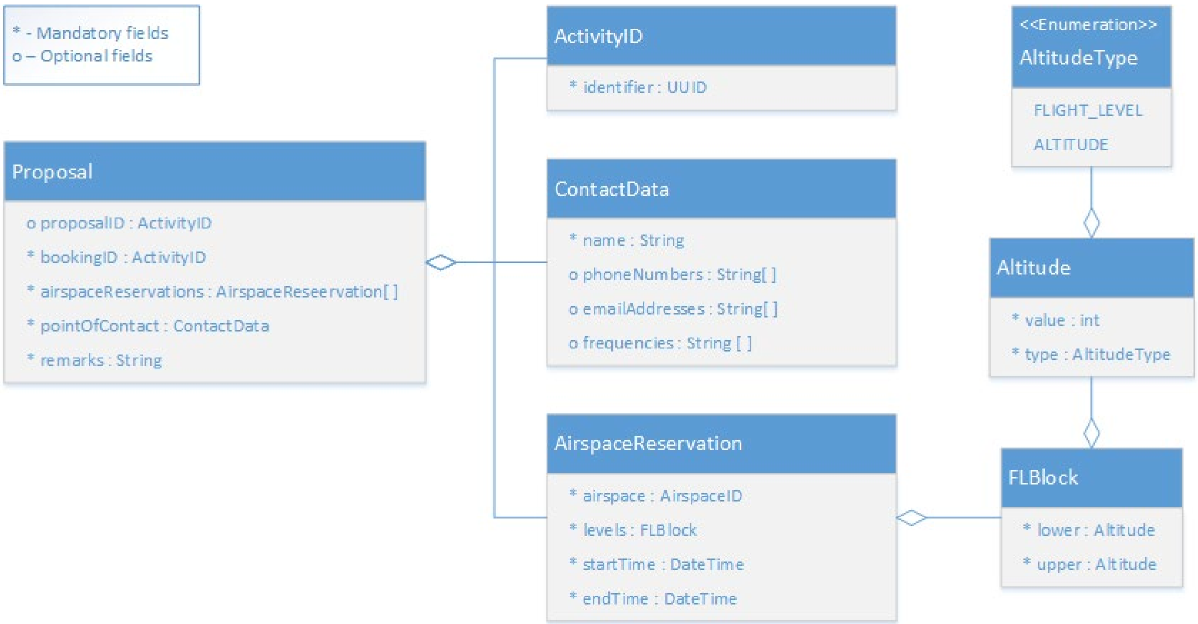


Figure 29: Airspace negotiation interface overview

The elements in the data model are described in full detail in section 2.7.3 Interface Messages.

2.4.9.2 Information Exchange Flow

The following diagram presents the information exchange flow for the Airspace Negotiation interface.

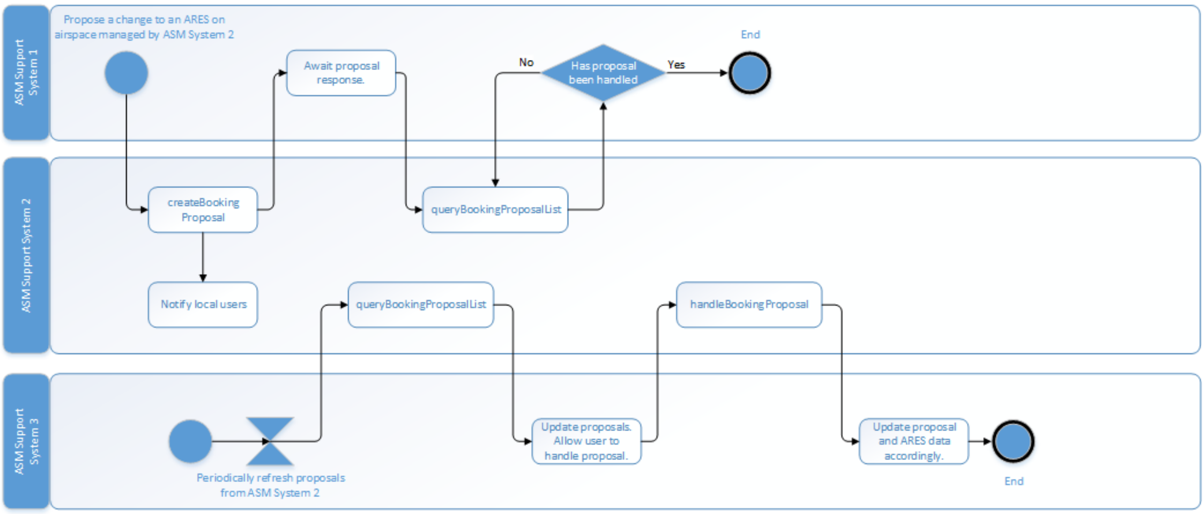

ASM Support System 1
Propose a change to an ARES on airspace managed by ASM System 2
Await proposal response.
No
Has proposal been handled
Yes
End
ASM Support System 2
createBooking Proposal
queryBookingProposalList
Notify local users
queryBookingProposalList
handleBookingProposal
ASM Support System 3
Periodically refresh proposals from ASM System 2
Update proposals. Allow user to handle proposal.
Update proposal and ARES data accordingly.
End

Figure 30: Airspace negotiation interface information exchange flow

The diagram shows the ASM Support System with two External Users. External User 1 identified an ARES that it has permission to propose a change on. It proposes a change through the createBookingProposal operation. The ASM Support System processes this request and notifies its local users as normal. External User 1 then polls the queryBookingProposalList proposal with filters for its proposal to determine whether the proposal has been acted on. At the same time External User 2 is also polling the service and picks up the new proposal which it then presents to the end operator. The end operator decides whether the proposal is to be accepted or rejected and their response is sent to the ASM Support System through the handleBookingProposal operation. External User 2 then updates its stored ARES data with the result of the proposal. External User 1 sees that the proposal no longer exists in the service and so has been either accepted or rejected.

2.4.9.3 Interface Functions

The interface performs the following functions:

- Creating a Proposal - introduces a new Proposal into the ASM Support System.
- Requesting Proposal List - allows access to the Proposal information from within the ASM Support System to allow for CDM processes.
- Handling Proposals – accepts or rejects the Proposal.

ASM-INTF-NEG-010: ASM Service **should** be supported by the Airspace Negotiation interface to manage the reservations.

ASM-INTF-NEG-020: The AirspaceNegotiation interface **shall** support the following operations:

- createBookingProposal,
- queryBookingProposalList
- handleBookingProposal

2.4.9.4 Service Operations and Associated Messages

*2.4.9.4.1 **createBookingProposal***

ASM-INTF-NEG-030: The createBookingProposal operation **shall** receive and process the BookingProposalRequest message from an External User.

ASM-INTF-NEG-040: If the proposal is valid, the createBookingProposal operation **shall** transmit the newly created proposal in the BookingProposalReply message to the requesting External Users.

ASM-INTF-NEG-050: If for any reason the request or the resulting proposal is not valid, the createBookingPrposal operation **shall** transmit an appropriate error in the BookingProposalReply message to the requesting External Users.

*2.4.9.4.2 **queryBookingProposalList***

ASM-INTF-NEG-060: The queryBookingProposalList operation **shall** receive and process the FilteredRequest message from an External User.

ASM-INTF-NEG-070: If the request is valid, the queryBookingProposalList operation **shall** transmit the list of proposals in the BookingProposalListReply message to the requesting External Users.

ASM-INTF-NEG-080: If for any reason the request is not valid, the queryBookingProposalList operation **shall** transmit an appropriate error in the BookingProposalListReply message to the requesting External Users.

ASM-INTF-NEG-085: The queryBookingProposalList operation **shall** accept any combination of the following filters in the FilteredRequest message:

- ActivityIDFilter
- ChangePeriodFilter
- InterestedIntervalFilter
- AndFilter

*2.4.9.4.3 **handleBookingProposal***

ASM-INTF-NEG-090: The handleBookingProposal operation **shall** receive and process the HandleBookingProposalRequest message from an External User.

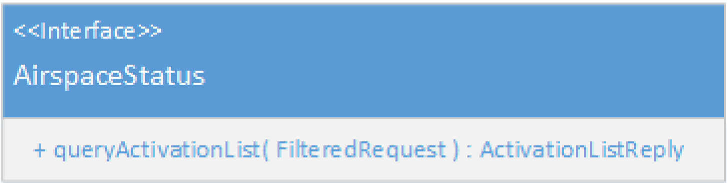
ASM-INTF-NEG-100: If the request is valid, the handleBookingProposal operation **shall** perform the requested action.

ASM-INTF-NEG-110: If for any reason the request is not valid, the updateBooking operation **shall** transmit an appropriate error in the HandleBookingProposalReply message to the requesting External User.

ASM-INTF-NEG-120: A HandleBookingProposalRequest **should** be rejected by the service if the user performing the update does not have the appropriate actions matching their update.

Note: The definition of these messages can be found in section 2.7.3 Interface Messages.

2.4.10 Airspace Status Interface



<<Interface>>
AirspaceStatus

+ queryActivationList(FilteredRequest) : ActivationListReply

Figure 31: Airspace status interface operations

2.4.10.1 Interface role

The Airspace Status interface enables External Users (service consumers) to retrieve activation data on the status of airspace in the ASM system in real time.

An activation refers to the specific state of an airspace structure at the current time and until a specified end time. The end time of an activation is subject to change, it may be shortened or extended.

This interface allows consumers to see in real time the status of the airspace managed by the ASM Support System. The activation data provided contains the information known by the ASM system including the start time of the current activation state and the expected end time. Within the ASM system, the actual end time may be changed during the tactical phase. The changes to the current activation state are communicated immediately. The current activation state can be supplemented with the planned airspace usage information available from the Booking Interface in section 2.4.6, thus allowing visibility of the latest planned upcoming activations and deactivations within the ASM system.

This interface is treated as safety critical when used via the publish-subscribe mechanism detailed in the Publication Interface section 2.4.3. Under such use it requires that the External User acknowledges receipt and understanding of the activation data allowing the ASM system to detect in cases that a change in status is not received and processed by the External User. In contrast to the other interfaces defined in this service, subscribing to activation data will cause the current activation state to be automatically fully published to the subscriber. This ensures the initial state is always transmitted to External Users subscribing to activation data and also ensures that the initial state is acknowledged.

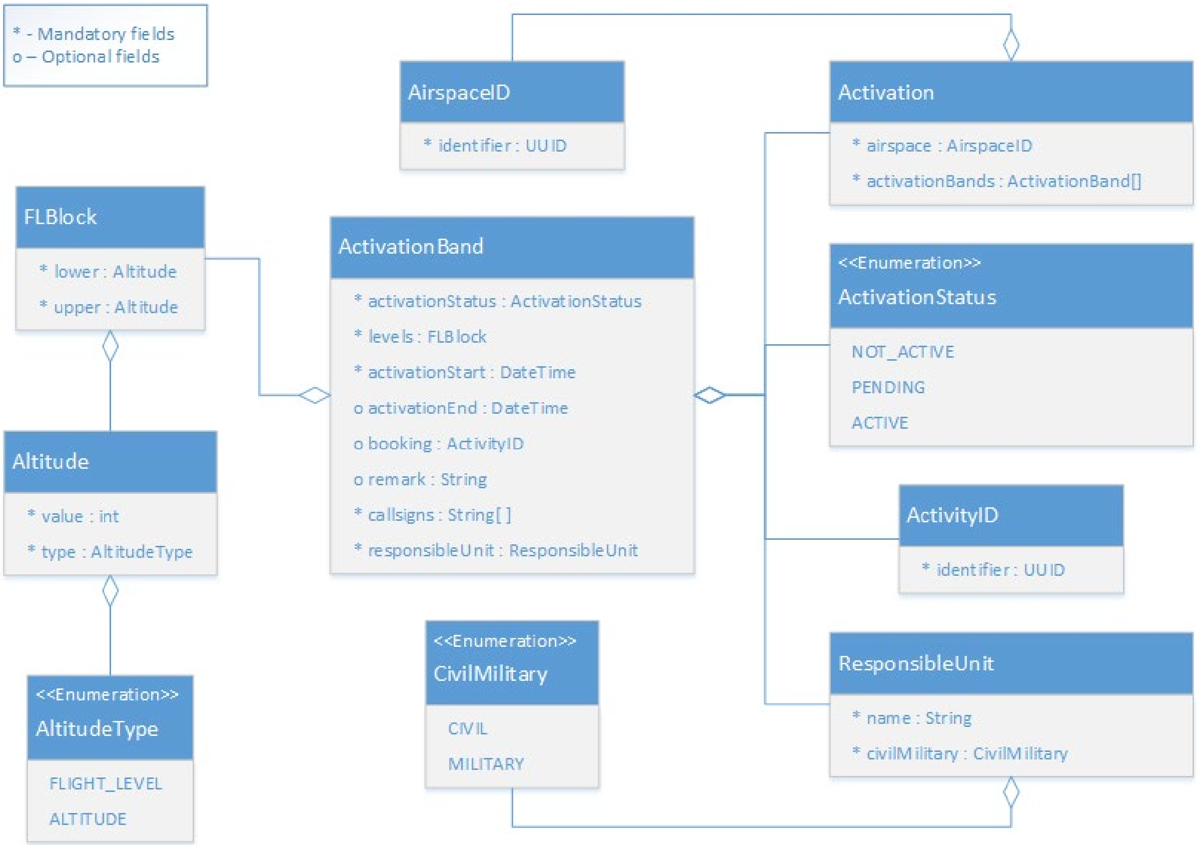


Figure 32: Airspace status interface overview

The elements in the data model are described in full detail in section 2.7.3 Interface Messages.

2.4.10.2 Information Exchange Flow

2.4.10.3 Basic Use Case

The External User (service consumer) requests the status of the airspace structures managed by the ASM Support System via the Synchronous Request/Reply mechanism by calling *queryActivationList*. The ASM System sends the status (active, inactive, released) of all airspace structures to the External user.

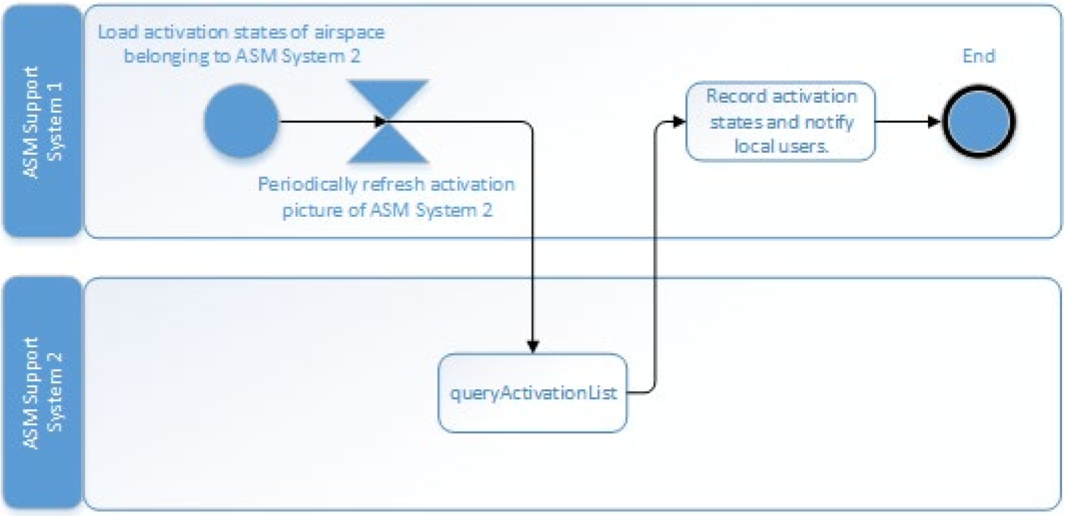

ASM Support System 1
Load activation states of airspace belonging to ASM System 2
Periodically refresh activation picture of ASM System 2
Record activation states and notify local users.
End
ASM Support System 2
queryActivationList

Figure 33: Airspace status interface information exchange flow

2.4.10.4 An ATC system is a Service Consumer of an ASM Support System

Due to the safety critical nature of the data exchange in this use case, it is required that the data be exchanged using the publish subscribe mechanism. This mechanism will supplement the `ActivationNotification` messages with `HeartbeatTechnicalMessage` messages to prove to the External User (service consumer) that the connection remains active (see 2.4.3 Publication Interface).

Upon initial connection, the consumer subscribes to the ASM Support System to the `ACTIVATIONS` topic via the Publish/Subscribe Push mechanism.

On receiving a new request for the `ACTIVATIONS` topic, the server sends the current status of all airspace meeting the filter conditions to the queue assigned to that topic.

All messages are acknowledged by the ATC system. In case a message is not acknowledged the ASM Support system detects that external consumers may not be aware of the current activation status.

The acknowledgement that the `ActivationNotification` was received is performed by having the consumer push an `ActivationReceivedRequest` to the queue referenced in the `ActivationNotification` message. The UUID used in the `ActivationReceivedRequest` matches the UUID in the `ActivationNotification` Message.

A filtering mechanism can be applied as specified in section 2.3.2 thus ensuring efficiency by filtering out irrelevant data. For example, the service consumer may wish to receive activations for airspace structure only in a particular geographical region, or only above a certain altitude.

Local Airspace Structure interface can be applied as specified in section 2.4.5 thus ensuring consistency of reference static data, for example locally created in ASM system airspace structure not available on the ATC side.

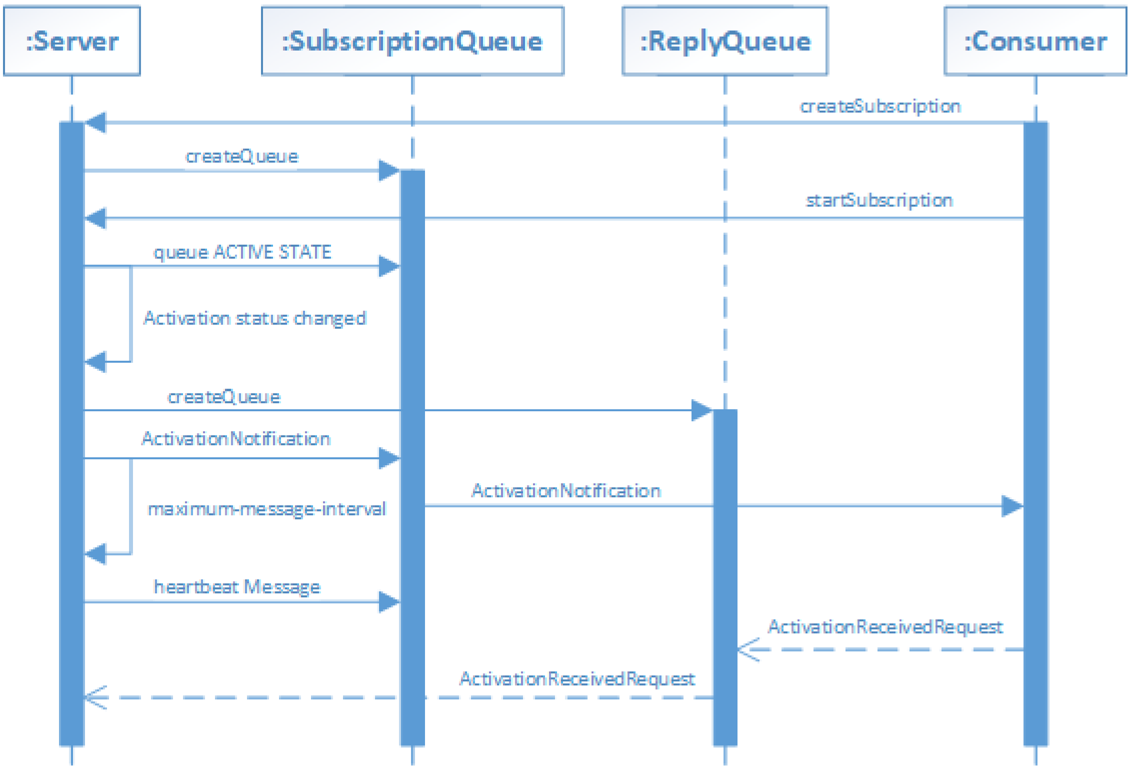


Figure 34 Use of Airspace Status AMQP interface

The above diagram shows the sequence of calls for use of AMQP service to receive airspace status information in safety related environments.

The External User (service consumer) application requests that the ACTIVATIONS subscription be created and started, the server consequently creates a subscription queue in the ACTIVE state.

Whenever an airspace changes activation state the server pushes an `ActivationNotification` to the subscription queue. This `ActivationNotification` contains an `acknowledgeQueueName` field. The consumer is then required to push an `ActivationReceivedRequest` message containing the UUID of the `ActivationNotification` onto the AMQP queue with the corresponding name, in order to acknowledge that the consumer application has received and understood valid `ActivationNotification` message. The information definition describes what is required for messages to be valid. It is the responsibility of the consumer to determine that a message is valid.

It is necessary to acknowledge all `ActivationNotification` messages sent to the AMQP queue.

It is not necessary to acknowledge any `HeartbeatTechnicalMessage` messages.

If an `ActivationReceivedRequest` is not received within a pre-defined Activation-acknowledgement-time then the server application alerts the appropriate ASM users of a failure with a safety-related activation client.

2.4.10.5 Interface Functions

The interface performs the following functions:

- Requesting activations - Allows access to the activation data held within the ASM

Support System.

ASM-INTF-STAT-010: ASM Service **should** be supported by the Airspace Status interface to manage the retrieval of activation data held within the ASM Support System by external users.

ASM-INTF-STAT-020: The AirspaceStatus interface **shall** support the following operations:

- queryActivationList

2.4.10.6 Service Operations and Associated Messages

2.4.10.6.1 ***queryActivationList***

ASM-INTF-STAT-030: The queryActivationList operation **shall** receive and process the FilteredRequest message from an External User

ASM-INTF-STAT-040: If the activation list request is valid, the queryActivationList operation **shall** transmit the matching activation data in the ActivationListReply message to the requesting External User

ASM-INTF-STAT-050: If for any reason the request is not valid, the queryActivationList operation **shall** transmit an appropriate error in the ActivationListReply message to the requesting External User

ASM-INTF-STAT-055: The queryActivationList operation **shall** accept any combination of the following filters in the FilteredRequest message:

- ActivityIDFilter
- AirspaceIDFilter
- ChangePeriodFilter
- InterestedIntervalFilter
- AndFilter

Note: The definition of these messages can be found in section 2.7.3 Interface Messages*.*

ASM-INTF-STAT-060: The External User **shall** only make use of data returned in an ActivationListReply for non-safety critical information purposes.

ASM-INTF-STAT-070: The External User **shall** make use of the AMQP service for safety-relevant purposes.

ASM-INTF-STAT-080: The External User **should** acknowledge receipt of any valid ActivationNotification published onto an AMQP queue, by publishing an AcknowledgementReceiptRequest message onto the AMQP queue referenced in the ActivationNotification message.

ASM-INTF-STAT-090: The ASM Support System **shall** select an appropriate value for *Activation-acknowledgement-time* and ensure that this value is appropriately documented for use by service consumers.

ASM-INTF-STAT-100: If the ASM Support System does not receive an ActivationNotification within the time defined by *Activation-acknowledgement-time* then a warning **shall** be shown to applicable users in the ASM support system.

2.5 Technologies

2.5.1 Interface Bindings

The requirements in this section are intended to enable technical interoperability by specifying the technical interfaces and capabilities that are necessary to enable a reliable, secure and efficient exchange of information. These requirements are based on the SWIM TI Yellow Profile Specification [12].

ASM-INTF-TIP-010: The ASM services **shall** be implemented using either Synchronous Request/Reply or Publish/Subscribe Push message exchange patterns.

ASM-INTF-TIP-020: The ASM service that implement the Synchronous Request/Reply message exchange pattern **shall** provide an interface that is conformant to the WS (Web Services) SOAP (Simple Object Access Protocol) binding as defined in the SWIM TI Yellow Profile Specification [12].

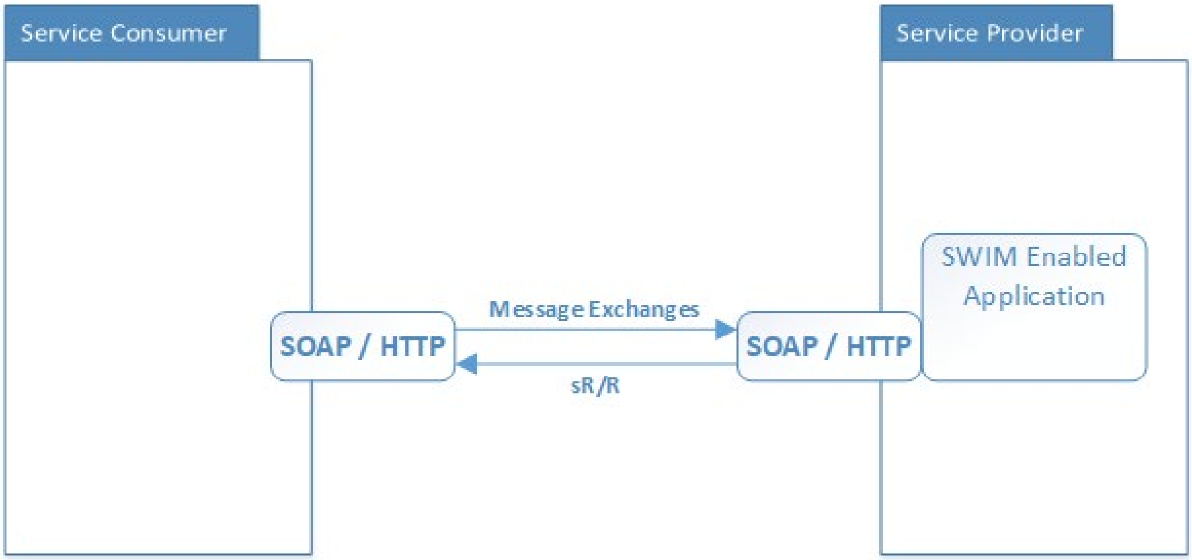


Figure 35: ASM Interface High Level Dynamic Behaviour for Synchronous R/R

ASM-INTF-TIP-030: Interfaces that implement a Publish/Subscribe Push message exchange pattern **shall** provide:

- A subscription interface that is conformant to the WS (Web Services) SOAP (Simple Object Access Protocol) binding as defined in the SWIM TI Yellow Profile Specification [12].
- A publication interface AMQP as defined in the SWIM TI Yellow Profile Specification [12].

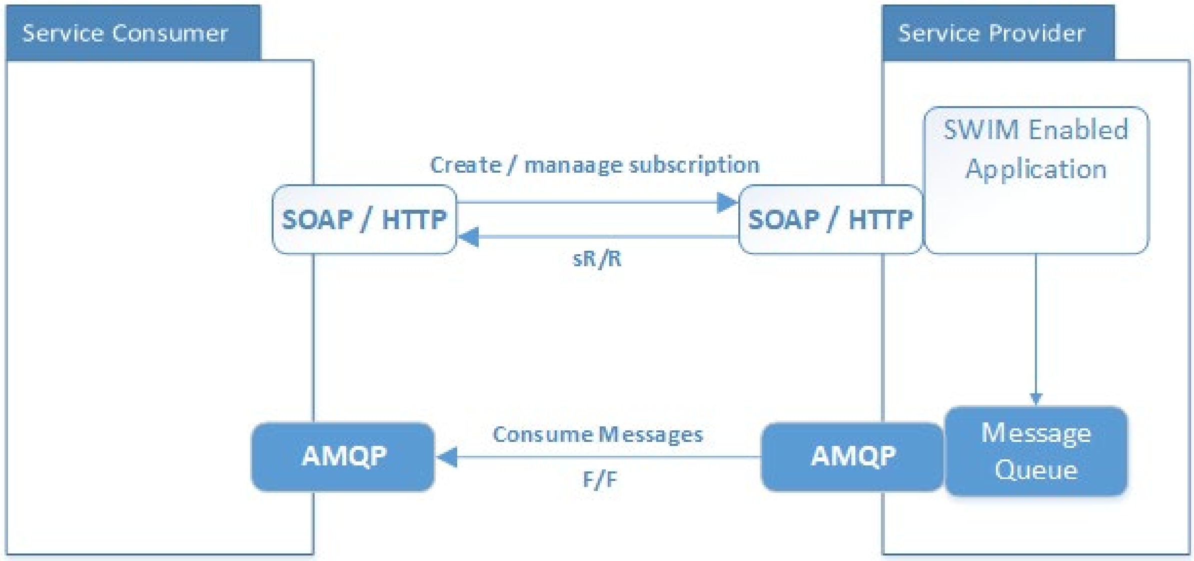


Figure 36: ASM Interface High Level Dynamic Behaviour for Publish-Subscribe Push

ASM-INTF-TIP-040: The ASM service **shall** implement the network interface bindings IPv4 Unicast and/or the IPv6 Unicast as defined in the SWIM TI Yellow Profile Specification [12].

2.5.2 Standard services definition formalism/language

The Web Service Description Language (WSDL) is provided in Annex G. The WSDL describes the structure of the messages and data types though without some of the explanatory text included in section 2.7 Information Definition.

In terms of guaranteeing interoperability, if two ASM systems are implementing from the same WSDL this will ensure that they are compatible and will significantly limit the scope for differences in implementation.

2.6 Service description requirements

ASM-INTF-SDR-010: The service providers **shall** provide a service description in accordance with the SWIM Service Description Specification [10].

ASM-INTF-SDR-020: During the implementation of the ASM service, details **shall** be provided in accordance with the following requirements:

- SWIM-SERV-150 - Service access and use conditions: “legal constraint, service policies, and service consumption constraints.”
- SWIM-SERV-160 – Security constraints
- SWIM-SERV-170 – Additional technical information for the service consumer.
- SWIM-SERV-180 - Quality of service: “performance, reliability, and security.”
- SWIM-SERV-190 – Source of information:

ASM-INTF-SDR-030: The Service provider **shall** include in the service description information in accordance with the following requirements applicable to products:

- SWIM-SERV-060 - Service provider;
- SWIM-SERV-070 – Provider point of contact;
- SWIM-SERV-080 – Support availability;
- SWIM-SERV-090 – Geographical extent of information;
- SWIM-SERV-110 – Service lifecycle information;
- SWIM-SERV-120 - Service standard reference;
- SWIM-SERV-200 - Service validation information;
- SWIM-SERV-340 - Examples of code.

2.7 Information Definition

2.7.1 Scope

Information definition in support of the ASM Service. This includes full definition of the data models for each service interface along with the messages and filters used to interact with the service.

The content of this section is fully aligned with the baseline requirements in Part I of this document.

2.7.2 Abstract Messages

2.7.2.1 <<abstract>> Reply

Abstract ancestor of all replies that will be returned by the service.

No XML reply is sent if the request is such that the system returned an HTTP error instead.

Attributes:

a. `ReplyStatus` status (Mandatory)
Defines the success of the request.

b. `String[]` errors (Optional)
Describes any specific error conditions that have been encounter while processing a request. Shall only be set if the status is not ‘OK’.

2.7.3 Interface Messages

2.7.3.1 ActivationReceivedRequest

Message pushed onto the queue referenced in the ActivationNotification used by the consumer to acknowledge receipt of the ActivationNotification.

Attributes:

a. `UUID` uuid (Mandatory)
ID of the activationNotification that you are acknowledging.

2.7.3.2 ActivationListReply

Reply returned from the queryActivationList operation.

Inherits from Reply.

Attributes:

a. `Activation[]` activations (Mandatory)
The requested activations.

2.7.3.3 ActivationNotification

Message notified to an Activation subscription as the result of a change in activation state.

Attributes:

a. `UUID` uuid (Mandatory)
   The ID of this activation, to be used to acknowledge receipt.
b. `Activation[]` activations (Mandatory)
   The changed activations.
c. `String` acknowledgeQueueName (Mandatory)
   Name of the queue used to acknowledge receipt of this activation notification by pushing an ActivationReceivedRequest.

2.7.3.4 ActivityDataListReply

Reply returned from the queryActivityDataList operation.

Inherits from Reply

Attributes:

a. `ActivityData[]` activityData (Mandatory)
   The activity data fulfilling the constraints of the FilteredRequest. The array can be empty.

2.7.3.5 ActivityDataNotification

Message notified to an activity data subscription when activity data changes.

Attributes:

a. `ActivityData[]` activityData (Mandatory)
   The changed activity data. The array can be empty.

2.7.3.6 AirspaceNotification

Message notified to a Static Data subscription as the result of a change to an airspace feature.

Attributes:

a. `AIXMBasicMessage` features (Mandatory)
   The returned AIXM features.

2.7.3.7 AirspaceListReply

Reply returned from the queryAirspaceList operation.

Inherits from Reply

Attributes:

a. `AIXMBasicMessage` features (Mandatory)
   The returned AIXM features.

2.7.3.8 BookingCreationRequest

Request to validate a new Booking and, on success, to create it.

Attributes:

a. `Booking` booking (Mandatory)
   The booking to be validated and created by the service.

b. `String` remark (Optional)
A remark for the booking.

2.7.3.9 BookingUpdateRequest

Request to validate an update to an existing Booking and, on success, to update it.

Updating can also be used to Cancel a booking by updating the bookingStatus attribute to CANCELLED providing the user performing the update has the appropriate action.

The Booking.lastChange serves as the concurrency control mechanism, in order to perform the update to the booking the supplied Booking.lastChange must match the Booking.lastChange held by the service.

Attributes:

a. `Booking` booking (Mandatory)
The booking to be validated and created by the service.

b. `String` remark (Optional)
A remark for the booking update.

2.7.3.10 BookingReply

Reply returned in response to a BookingCreationRequest or a BookingUpdateRequest

Inherits from Reply

Attributes:

a. `Booking` booking (Optional)
The created booking if the request was deemed valid.

2.7.3.11 BookingListReply

Reply returned from the queryBookingList operation.

Inherits from Reply.

Attributes:

a. `Booking[]` bookings (Mandatory)
The Bookings that fulfil the constraints of the `FilteredRequest`. The array can be empty.

2.7.3.12 BookingNotification

Message notified to a Booking subscription as the result of a change to a booking or the creation of a new booking.

Attributes:

a. `Booking` booking (Mandatory)
The changed Booking.

2.7.3.13 BookingConflictListReply

Reply returned from the queryConflictList operation.

Inherits from Reply.

Attributes:

a. `BookingConflict[]` conflicts (Mandatory)
The conflicts involving the Bookings identified in the request. The array may be empty.

2.7.3.14 BookingConflictNotification

Message notified to a Conflict subscription as the result of a change to one or more conflicts resulting from a change to a booking.

Attributes:

a. `BookingConflict[]` newConflicts (Mandatory)
Any new conflicts.

b. `BookingConflict[]` resolvedConflicts (Mandatory)
Any resolved/deleted conflicts.

2.7.3.15 BookingActionListReply

Reply returned from the queryActionList operation.

Inherits from Reply.

Attributes:

a. `BookingAction[]` actions (Mandatory)
The retrieved actions. The array can be empty.

2.7.3.16 BookingActionNotification

Message notified to an Action subscription as the result of a change to one or more actions resulting from a change to a booking.

Attributes:

a. `BookingAction[]` allowedActions (Mandatory)
All newly available actions.

b. `BookingAction[]` disallowedActions (Mandatory)
All pre-existing actions that are no longer allowed.

2.7.3.17 BookingHistoryListReply

Reply returned from the queryBookingHistoryList operation.

Inherits from Reply.

Attributes:

a. `BookingHistory[]` history (Mandatory)

The history for the requested bookings.

2.7.3.18 BookingProposalRequest

Request to validate a new Proposal and, on success, to create it.

Can be performed at any time.

Attributes:

a. `Proposal` proposal (Mandatory)
   The proposal to be validated and created by the service.

2.7.3.19 BookingProposalReply

Reply returned in response to ProposalRequest

Inherits from Reply

Attributes:

a. `Proposal` proposal (Optional)
   The created proposal if the request was deemed valid.

2.7.3.20 BookingProposalListReply

Reply returned from the queryBookingProposalList operation.

Inherits from Reply.

Attributes:

a. `Proposal []` proposals (Mandatory)
   The retrieved Proposals. The array can be empty.

2.7.3.21 BookingProposalNotification

Message notified to a Proposal subscription as the result of a proposal being created or handled (accepted/rejected).

Attributes:

a. `Proposal` proposal (Mandatory)
   The new or handled proposal.
b. `boolean` handled (Mandatory)
   True if the proposal has been handled and the associated booking updated. False if this is a new proposal.

2.7.3.22 EventCreationRequest

Request to validate a new Event and, on success, to create it.

This service may be constrained in terms of timing/process.

Attributes:

a. `Event` event (Mandatory)
   The event to be validated and created by the service.

2.7.3.23 EventUpdateRequest

Request to validate an update to an existing Event and, on success, to update it.

This service may be constrained in terms of timing/process.

Updating can also be used to Cancel an event by updating the eventStatus attribute to CANCELLED providing the user performing the update has the appropriate privilege.

The Event.lastChange serves as the concurrency control mechanism, in order to perform the update to the booking the supplied Event.lastChange must match the Event.lastChange held by the service.

Attributes:

a. `Event` event (Mandatory)
   The event to be validated and created by the service.

2.7.3.24 EventReply

Reply returned in response to an EventCreationRequest or an EventUpdateRequest.

Inherits from Reply

Attributes:

a. `Event` event (Optional)
   The created or updated event if the request was deemed valid.

2.7.3.25 EventListReply

Reply returned from the queryEventList operation.

Inherits from Reply.

Attributes:

a. `Event[]` events (Mandatory)
   The Events that fulfil the constraints of the FilteredRequest. The array can be empty.

2.7.3.26 EventNotification

Message notified to an Event subscription as the result of an event being created or modified.

Attributes:

a. `Event` event (Mandatory)
   The created or updated event.

2.7.3.27 FilteredRequest

Request to retrieve filtered data definitions from a service.

Constraints will be defined by the operation the request is targeted to, in particular the Filters that are supported for an operation.

Attributes:

a. `Filter[]` filters (Optional)

Defines the filters to be applied. Filters are applied with a logical OR. If no filters are applied all available data shall be returned.

2.7.3.28 HandleBookingProposalRequest

Request to handle a proposal, either accepting or rejecting it.

This service is constrained in terms of timing/process.

Attributes:

a. `Proposal` proposalID (Mandatory)
The ID of the proposal being ‘handled’

b. `ProposalActionType` acceptReject (Mandatory)
How the user wants to ‘handle’ the proposal.

2.7.3.29 HandleBookingProposalReply

Reply returned in response to HandleProposalReply.

Inherits from Reply.

Attributes:

a. `boolean` success (Mandatory)
Whether the action was successful.

2.7.3.30 HeartbeatTechnicalMessage

Message sent at periodic intervals on subscription queue used to indicate to the client that the server is operational and still serving the subscription.

Attributes:

a. `UUID` subscriptionUUID (Mandatory)
Unique Identifier.
b. `SubscriptionTopic` subscriptionTopic (Optional)
The topic of this subscription.
c. `SubscriptionState` currentSubscriptionState (Mandatory)
The current state of the subscription that this heartbeat is about.
d. `dateTime` timeSent (Mandatory)
The time on the server that this heartbeat message was sent.
e. `dateTime` timeToNextPulse (Mandatory)
The time when the next HeartbeatTechnicalMessage will be sent for this subscription.

2.7.3.31 MissionCreationRequest

Request to validate a new Mission and, on success, to create it.

Attributes:

a. `Mission` mission (Mandatory)
The mission to be validated and created by the service.

2.7.3.32 MissionUpdateRequest

Request to validate an update to an existing Mission and, on success, to update it.

The Mission.lastChange serves as the concurrency control mechanism, in order to perform the update to the mission the supplied Mission.lastChange must match the Mission.lastChange held by the service.

Attributes:

a. `Mission` mission (Mandatory)
The mission to be validated and created by the service.

2.7.3.33 MissionReply

Reply returned in response to a MissionCreationRequest or a MissionUpdateRequest.

Inherits from Reply.

Attributes:

a. `Mission` mission (Optional)
The created mission if the request was deemed valid.

2.7.3.34 MissionListReply

Reply returned from the queryMissionList operation.

Inherits from Reply.

Attributes:

a. `Mission[]` missions (Mandatory)
The Missions that fulfil the constraints of the MissionListRequest. The array can be empty.

2.7.3.35 MissionNotification

Message notified to a Mission subscription as the result of a change to a mission or the creation of a new mission.

Attributes:

a. `Mission` mission (Mandatory)
The changed missions.

2.7.3.36 SubscriptionCreationRequest

Request to create a subscription to a specific topic.

Inherits from FilteredRequest.

Attributes:

a. `SubscriptionTopic` topic (Mandatory)
The topic of the subscription.

b. `String` queueName (Optional)
If set, the queue named in this attribute will be reused for this subscription. Errors will be returned if the named queue does not exist.

2.7.3.37 SubscriptionCreationReply

Reply returned in response to SubscriptionCreationRequest.

Inherits from Reply.

Attributes:

a. `Subscription` subscription (Optional)
The definition of the newly created subscription. May be null if the original request was not valid.

2.7.3.38 SubscriptionListRequest

List all subscriptions owned by the calling External User

Attributes:

a. `SubscriptionTopic[]` states (Optional)
Return subscriptions that match one of the supplied states.

By default, returns all subscriptions owned by the calling external user, if the array is not present or empty.

2.7.3.39 SubscriptionListReply

Reply returned in response to SubscriptionListRequest.

Inherits from Reply.

Attributes:

a. `Subscription[]` subscriptions (Mandatory)
The subscriptions that fulfil the constraints of the SubscriptionListRequest. The array can be empty.

2.7.3.40 SubscriptionStartRequest

Request to start notifications for a subscription.

Attributes:

a. `UUID` uuid (Mandatory)
The UUID of the subscription to start.

2.7.3.41 SubscriptionStartReply

Reply returned in response to SubscriptionStartRequest.

Inherits from Reply.

Attributes:

a. `Subscription` subscription (Optional)
The updated subscription. May be null if the original request was not valid.

2.7.3.42 SubscriptionStopRequest

Request to stop notifications for a subscription.

Attributes:

a. `Subscription` uuid (Mandatory)
The UUID of the subscription to stop.

b. `boolean` heartbeatEnabled (Mandatory)
Set to true to disable the heartbeat mechanism while the subscription is in the state PAUSED.

2.7.3.43 SubscriptionStopReply

Reply returned in response to SubscriptionStopRequest

Inherits from Reply.

Attributes:

a. `Subscription` subscription (Optional)
The updated subscription. May be null if the original request was not valid.

2.7.3.44 SubscriptionDeletionRequest

Request to delete a subscription to a specific topic.

Attributes:

a. `UUID` uuid (Mandatory)
The UUID of the subscription to delete.

2.7.3.45 SubscriptionDeletionReply

Reply returned in response to SubscriptionDeletionRequest.

Inherits from Reply.

Attributes:

a. `boolean` success (Mandatory)
Whether the deletion was successful.

2.7.3.46 SubscriptionTechnicalMessage

Technical message added to the queue by the server to indicate that the server has changed the state of a subscription.

Attributes:

a. `SubscriptionState` state (Mandatory)
The new state of the subscription.

b. `dateTime` lastUpdatedOn (Mandatory)
Timestamp of when the subscription was last updated.

c. `String` lastUpdateReason (Mandatory)

The description of the reason why the subscription status was changed by the server.

2.7.4 Complex Data Types

2.7.4.1 AIXMBasicMessage

Container for AIXM features returned from the service. Defined by AIXM 5.1.1.

2.7.4.2 Activation

The complete operational state of a specific airspace.

Attributes:

a. `AirspaceID` airspace (Mandatory)
The identifier of the airspace this activation applies to.
b. `ActivationBand[]` activationBands (Mandatory)
Bands describing the current pending and active levels or a single band describing the entire airspace as not active.

2.7.4.3 ActivationBand

The activation state for a vertical section of an airspace.

Attributes:

a. `ActivationStatus` activationStatus (Mandatory)
The activation status of this band.
b. `FLBlock` levels (Mandatory)
The levels of this band.
c. `dateTime` activationStart (Mandatory)
The time of the start of this activation state.
d. `dateTime` activationEnd (Optional)
The planned end time of this activation state. May be null in the case of the band representing not active or if the end time of the activation is not known, potentially due to a required user interaction.
e. `ActivityID` booking (Optional)
The booking identifier causing this activation state. May be null only in the case of the band representing not active.
f. `String` remark (Optional)
Remark associated with the booking.
g. `String[]` callsigns (Mandatory)
Array of callsigns associated with the booking. The array may be empty.
h. `ResponsibleUnit` responsibleUnit (Optional)
Unit responsible for the activation band. Can be null if the activation is inactive.

2.7.4.4 ActivityData

Additional system specific data associated with a specific airspace structure supporting the creation of valid bookings.

Attributes:

a. `AirspaceID` airspace (Mandatory)
The identifier of the airspace this data applies to.
b. `ResponsibleUnit[]` responsibleUnits (Mandatory)
The responsible units that can be responsible for bookings on the identified airspace.
c. `boolean` bookable (Mandatory)
Whether the identified airspace may be booked.

2.7.4.5 AirspaceRelease

A vertical block of airspace included in a booking to be temporarily released from use.

Attributes:

a. `AirspaceID[]` airspace (Mandatory)
The identifier(s) of the airspace to be released.
b. `FLBlock` levels (Mandatory)
The vertical level band that is being released from all airspace.
c. `dateTime` startTime (Mandatory)
The start date and time of the airspace release.
d. `dateTime` endTime (Optional)
The end date and time of the airspace release. This may be undefined if there is no known date and time for the airspace to be returned to use by the original booking.
e. `AirspaceReleaseState` state (Mandatory)
The current state of this airspace release.

2.7.4.6 AirspaceReservation

A vertical block of an airspace to be used.

Attributes:

a. `AirspaceID` airspace (Mandatory)
The identifier of the airspace that is being booked.
b. `FLBlock` levels (Mandatory)
The vertical level band that is being booked.
c. `dateTime` startTime (Mandatory)
The start date and time of the airspace usage.

d. `dateTime` endTime (Mandatory)
The end date and time of the airspace usage.

2.7.4.7 Altitude

A specific altitude value and unit pair.

Attributes:

a. `int` value (Mandatory)
The altitude value.

b. `AltitudeType` type (Mandatory)
The altitude unit.

2.7.4.8 Booking

An ARES as defined by Baseline requirements Part I. Defines the times and levels of airspace that are planned to be used.

Attributes:

a. `ActivityID` bookingID (Optional)
Uniquely identifies the Booking.

  I. Must be NULL when creating the Booking.
  II. Cannot be NULL any time after creation.

b. `BookingStatus` bookingStatus (Optional)
The current state of the booking.

  I. Must be NULL when creating the Booking.
  II. Cannot be NULL any time after creation.

c. `AirspaceReservation[]` airspaceReservations (Mandatory)
The airspace and levels at which they are booked. The array must not be empty.

d. `AirspaceRelease[]` airspaceReleases (Optional)
One or more airspace releases.

e. `MissionID[]` missions (Optional)
One or more references to associated missions.

f. `ResponsibleUnit` responsibleUnit (Mandatory)
The responsible unit who will be responsible for this booking during the activation phase.

g. `int` priority (Optional)
The priority level of this booking where 1 is the highest priority, reflecting the operational need for the booking.

h. `int` numberOfAircraft (Optional)
The number of aircraft associated with this booking.

i. `Event` event (Optional)
An optional reference to an associated Event.

j. `AircraftIdentification[]` callsigns (Optional)
The aircraft identifiers associated with this booking. The list may be empty but must be present.

k. `String` remarks (Optional)
Read-only field containing the existing remarks in this booking.

l. `ContactData` pointOfContact (Mandatory)
The contact information for this Booking.

m. `dateTime` lastChanged (Optional)
Provides a concurrency mechanism.

I. Must be NULL on creation.
II. Must match the value held by the service when updating otherwise a versioning error will be returned.

2.7.4.9 BookingAction

A modification that may be made to a specific booking.

Attributes:

a. `ActivityID` bookingID (Mandatory)
The system identifier of the Booking this action is for.

b. `BookingActionType` action (Mandatory)
The type of action that is allowed by this BookingAction.

2.7.4.10 BookingChange

An overview of a specific historical change made to a booking.

Attributes:

a. `dateTime` changeTime (Mandatory)
The time at which the change was made.

b. `String` user (Mandatory)
The identifier of the user that made the change.

c. `ChangeType` changeType (Mandatory)
The type/reason for the change that was made.

d. `String` changeDescription (Mandatory)
A textual description of the change.

2.7.4.11 BookingConflict

A pair of bookings which are unlikely to both gain approval due to geometric and time-based overlap of the booked airspace, or violation of other rules defined by the ASM Support System.

Values:

a. `ActivityID` bookingA (Mandatory)
The unique identifier of the first booking in the conflict.

b. `ActivityID` BookingB (Mandatory)
The unique identifier of the second booking in the conflict.

2.7.4.12 BookingHistory

A descriptive history of a specific booking.

Values:

a. `ActivityID` bookingID (Mandatory)
The unique identifier of the booking the history is for.

b. `BookingChange[]` history (Mandatory)

The changes made to the booking. The array may be empty.

2.7.4.13 ContactData

Contact information for a related entity.

Attributes:

a. `String` name (Mandatory)
   The contact name.
b. `String[]` phoneNumbers (Optional)
   Any available telephone numbers.
c. `String[]` emailAddresses (Optional)
   Any available email addresses.
d. `String[]` frequencies (Optional)
   Any available radio frequencies.

2.7.4.14 Event

A planned event which will have an impact on airspace requirements.

Attributes:

a. `ActivityID` eventID (Optional)
   Uniquely identifies the event.
   I. Must be NULL while creating.
   II. Cannot be NULL once the Mission has been created.
b. `String` title (Mandatory)
   A descriptive title of the event.
c. `EventSummary` summary (Mandatory)
   The summary details of the event defining the type of event, its status, period, location and a brief description.
d. `String` fullDescription (Optional)
   A detailed description of the event.
e. `String` expectedBenefits (Optional)
   The expected benefits of the event, when applicable.
f. `String` remarks (Optional)
   Free remarks/comments about the events.
g. `EventOriginator` originator (Optional)
   The originator of this Event.
h. `dateTime` lastChanged (Optional)
   Provides a concurrency mechanism.
   I. Must be NULL on creation.
   II. Must match the value held by the service when updating otherwise a versioning error will be returned.

2.7.4.15 EventSummary

The summary details of an Event.

Attributes:

a. `EventType` type (Mandatory)
   The type of the associated event.
b. `String` subtype (Optional)
   A free-text subtype of the event.
c. `EventStatus` status (Optional)
   The status of the event.
d. `dateTime` start (Mandatory)
   The start date and time of the event.
e. `dateTime` end (Mandatory)
   The end date and time of the event.
f. `String` location (Optional)
   The location of the event.
g. `String` shortDescription (Optional)
   A short, summary, description of the event.
h. `dateTime` creationTime (Optional)
   The date and time at which the event was created. Must be null on creation and cannot be null at any time afterwards.
i. `dateTime` cancellationTime (Optional)
   The date and time at which the event was cancelled. Set by the service. Must be null while the EventStatus is not 'CANCELLED', must not be null if the EventStatus is 'CANCELLED'.
j. `dateTime` lastPublished (Optional)
   The date and time at which the event was last published.

2.7.4.16 EventOriginator

The details of the originator of an Event.

Attributes:

a. `String` organisationType (Optional)
   The type of organisation the Event originated from.
b. `String` organisationName (Optional)
   The name of the organisation the Event originated from.

2.7.4.17 FLBlock

A block of flight levels between lower and upper limits.

Attributes:

a. `Altitude` lower (Mandatory)
   The lower limit of this block.
b. `Altitude` upper (Mandatory)
   The upper limit of this block.

2.7.4.18 Mission

Operation carried out by an aircraft or a group of aircraft used in military service.

Attributes:

a. `MissionID` missionID (Optional)
   Uniquely identifies the mission.
   I. Must be NULL while creating.
   II. Cannot be NULL once the Mission has been created.
b. `String` missionType (Optional)
   The type of mission this is.
c. `String` aircraftType (Optional)
   The type of aircraft.
d. `AirportCode` departureAirport (Optional)
   The airport code of the departure airport.
e. `AirportCode` arrivalAirport (Optional)
   The airport code of the arrival airport.
f. `MissionID` linkedMission (Optional)
   A link to another mission.
g. `dateTime` lastChanged (Optional)
   Provides a concurrency mechanism.
   I. Must be NULL on creation.
   II. Must match the value held by the service when updating otherwise a versioning error will be returned.

2.7.4.19 Proposal

A suggested change to the airspace, times and levels of an existing booking generally to improve airspace use.

Attributes:

a. `ActivityID` proposalID (Optional)
   Uniquely identifies the proposal.
   I. Must be NULL on creation of the proposal.
   II. Cannot be NULL any time after creation.
b. `ActivityID` bookingID (Mandatory)
   Unique ID of the booking this proposal is for.
c. `AirspaceReservation[]` airspaceReservations (Mandatory)
   Contains the proposed airspace usage. The array must not be empty.
d. `ContactData` pointOfContact (Mandatory)
   The point of contact for this proposal.
e. `String` remarks (Optional)
   Free text remarks about the proposal.

2.7.4.20 ResponsibleUnit

An organisation responsible for the activation phase of a booking.

Attributes:

a. `String` name (Mandatory)
   The user recognised name of the responsible unit.

b. `CivilMilitary` civilMilitary (Mandatory)
   Whether the responsible unit is civil or military.

2.7.4.21 Subscription

The details of a Subscription created to a specific Subscription Topic through the service.

Attributes:

a. `UUID` uuid (Mandatory)
   The unique identifier of the subscription.
b. `String` queueName (Mandatory)
   The name of the queue associated to this subscription.
c. `SubscriptionTopic` topic (Mandatory)
   The topic of the subscription.
d. `SubscriptionState` state (Mandatory)
   The current state of the subscription.
e. `boolean` heartbeatEnabled (Optional)
   True if heartbeat is enabled and subscription queue will also receive HearbeatTechnicalMessages. It must be true for ACTIVE subscriptions, but the heartbeats can be disabled for PAUSED subscriptions.

2.7.5 Simple Data Types

2.7.5.1 <<enumeration>> ActivationStatus

Enumerates the possible states an airspace block can be in.

Values:

a. `NOT_ACTIVE`
   The airspace is not in use and there a no approved planned uses in the near future.
b. `PENDING`
   The airspace is not currently in use but there is an approved use in the near future.
c. `ACTIVE`
   The airspace is currently in use.

2.7.5.2 <UUID> ActivityID

Unique ID of a Reservation or Proposal, allocated by the service.

2.7.5.3 <string> AircraftIdentification

A callsign to be associated with a booking. Defined as a String restricted to 15 characters.

2.7.5.4 <string> AirportCode

Unique 4-letter airport code provided by ICAO.

2.7.5.5 <UUID> AirspaceID

Unique ID of an Airspace, allocated by the service.

2.7.5.6 <<enumeration>> AirspaceReleaseState

Enumerates the possible states of an AirspaceRelease.

Values:

a. `RELEASE_SCHEDULED`
The initial state of a release before airspace has been released.
b. `RELEASED`
The state of an ‘active’ release while the associated airspace is released.
c. `REACTIVATION_SCHEDULED`
The state of an active release once reactivation of the associated airspace has started.
d. `COMPLETED`
The state reached once the associated airspace has been reactivated or the associated booking has ended.
e. `CANCELLED`
If the planned release has been cancelled before becoming RELEASED.

2.7.5.7 <<enumeration>> AltitudeType

The unit of an Altitude value.

Values:

a. `FLIGHT_LEVEL`
Values measured in FL.
b. `ALTITUDE`
Values measured in ft.

2.7.5.8 <<enumeration>> BookingActionType

Enumerates the possible types of BookingAction available.

Values:

a. `MAJOR_EDIT`
Allows a full edit of all fields of the booking.
b. `MINOR_EDIT`
Allows editing of non-essential fields, editing of times, airspace and levels is not allowed.
c. `ACTIVE_EDIT`
Allows editing of any Airspace Reservation while the Booking state is ACTIVE. Editing of the start and end times that are in the past are not allowed.
d. `HANDLE_PROPOSAL`
Allows accepting or rejecting the proposal associated with the booking.
e. `EDIT_REMARKS`

Allows addition of a new remark to the booking.

f. `CANCEL`
Allows the booking to be cancelled.

2.7.5.9 <<enumeration>> BookingStatus

Enumerates the possible states of a Booking

Values:

a. `INITIAL_REQUEST`
Initial state for a Booking while awaiting full approval.

b. `APPROVED`
The state reached by a Booking after approval by all involved parties.

c. `ALLOCATED`
The state reached when a Booking is included in a 'RELEASED' AUP.

d. `REFERENCE_ALLOCATED`
The state reached when a Booking enters into the 'Reference Allocation' buffer on the booked airspace. Typically, three hours prior the start time of the earliest Airspace Reservation.

e. `PRE_ACTIVE`
The state reached by a Booking before activation. Typically, thirty minutes prior activation.

f. `ACTIVE`
The state reached at the start time of the earliest Airspace Reservation.

g. `COMPLETED`
The state reached at the end time of the latest Airspace Reservation.

h. `CANCELLED`
The state a Booking is in once it has been cancelled.

2.7.5.10 <<enumeration>> ChangeType

Enumerates the possible types of change that can be made to a booking.

Values:

a. `APPROVED`
Identifying that the booking was given approval.

b. `CANCELLED`
Identifying that the booking was cancelled.

c. `CANCEL_REQUESTED`
Identifying that cancellation of the booking was requested.

d. `CREATED`
Identifying that the booking was created.

e. `EDITED`

Identifying that an edit was made to the booking.

f. `PROPOSAL_ACCEPTED`
Identifying that a proposal associated with this booking was accepted.

g. `PROPOSAL_REJECTED`
Identifying that a proposal associated with this booking was rejected.

h. `PROPOSED`
Identifying that a proposal was made on this booking.

i. `STATUS_CHANGED`
Identifying that the status of the booking changed.

2.7.5.11 <<enumeration>> CivilMilitary

Enumeration identifying either Civil or Military.

Values:

a. `CIVIL`
Identifies the associated element as Civil.
b. `MILITARY`
Identifies the associated element as Military.

2.7.5.12 <<enumeration>> EventStatus

Enumerates the possible states of an Event.

Values:

a. `PROPOSED`
The state of the Event when initially proposed.
b. `PLANNED`
The state of the Event once it is planned.
c. `CONFIRMED`
The state once the event has been confirmed.

d. `ONGOING`
The state once the event is in progress.

e. `ON_HOLD`
The state if the event is suspended.

f. `COMPLETED`
The state once the event is has completed.

g. `CANCELLED`
The state of an Event once it has been cancelled.

2.7.5.13 <<enumeration>> EventType

Enumerates the possible types of Event.

Values:

a. `MILITARY`

Type for Military events.

b. `SPECIAL`
Type for any non-military events.

2.7.5.14 <<enumeration>> FeatureType

Enumerates the possible types of airspace feature.

Values:

a. `AIRSPACE`
AIXM Airspace features.

b. `ROUTE`
AIXM Route features.

c. `ROUTE_SEGMENT`
AIXM Route Segment features.

d. `NAVAID`
AIXM Navaid features.

e. `DESIGNATED_POINT`
AIXM Designated Point features.

f. `FLIGHT_RESTRICTION`
AIXM Flight Restriction features.

g. `FLIGHT_RESTRICTION_GROUP`
ADR Flight Restriction Group features.

2.7.5.15 <UUID> MissionID

Unique ID of a Mission, allocated by the service.

2.7.5.16 <<enumeration>> ProposalActionType

Enumerates the possible ProposalActions.

Values:

a. `ACCEPT`
Accepts the proposal to be applied to the associated booking.

b. `REJECT`
Rejects the proposal such that no change is made to the booking.

2.7.5.17 <<enumeration>> ReplyStatus

Enumerates the possible ReplyStatus values, indicating the success or reason for failure of a request to the service.

Values:

a. `OK`
If the request completed successfully.

b. `INVALID_INPUT`
If the input was invalid for any reason.

c. `NOT_AUTHORISED`
If the user is not authorised to make the request.

d. `OBJECT_NOT_FOUND`
   If the subject of the request does not exist.
e. `OBJECT_OUT_OF_DATE`
   If a newer version of the subject of the request exists.
f. `INTERNAL_SERVER_ERROR`
   If an unexpected error occurred within the service.

2.7.5.18 <<enumeration>> SubscriptionState

Enumerates the possible subscription states.

Values:

a. `PAUSED`
   The subscription exists but is not being notified. It can be started.
b. `ACTIVE`
   The subscription exists and is being notified.
c. `DELETED`
   The subscription has been deleted. It is not being notified and cannot be started.

2.7.5.19 <<enumeration>> SubscriptionTopic

Enumerates the possible subscription topics.

Values:

a. `STATIC_DATA`
   Maps to the *queryAirspace* operation.
b. `ACTIVITY_DATA`
   Maps to the *queryActivityDataList* operation.
c. `BOOKINGS`
   Maps to the *queryBookingList* operation.
d. `CONFLICTS`
   Maps to the *queryConflictList* operation.
e. `ACTIONS`
   Maps to the *queryActionList* operation.
f. `MISSIONS`
   Maps to the *queryMissionList* operation.
g. `PROPOSALS`
   Maps to the *queryBookingProposalList* operation.
h. `EVENTS`
   Maps to the *queryEventList* operation.
i. `ACTIVATIONS`
   Maps to the *queryActivationList* operation.

2.7.5.20 <String> UUID

Universally Unique Identifier defined as a string of the pattern: [a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}.

2.7.6 Filters

2.7.6.1 Filter

Abstract filter that all implementations inherit from. To be used when requesting and subscribing for data.

Defines no values.

2.7.6.2 ActivityIDFilter

A Filter defining a series of ActivityIDs to be retained.

Inherits from Filter.

Attributes:

a. `ActivityID[]` activities (Mandatory)
The ActivityIDs to be retained.

2.7.6.3 AirspaceIDFilter

A Filter defining a series of AirspaceIDs to be retained.

Inherits from Filter.

Attributes:

a. `AirspaceID[]` airspaces (Mandatory)
The AirspaceIDs to be retained.

2.7.6.4 AirspaceNameFilter

A Filter defining a series of airspace names to be retained.

Inherits from Filter.

Attributes:

a. `String[]` names (Mandatory)
The names of the airspace features to be retained.

2.7.6.5 AirspaceTypeFilter

A Filter defining a series of airspace feature types to be retained.

Inherits from Filter.

Attributes:

a. `FeatureType[]` featureTypes (Mandatory)
The types of airspace feature to be retained.

2.7.6.6 AndFilter

A Filter defining a series of Filters to be applied with a logical AND.

Inherits from Filter.

Attributes:

a. `Filter[]` filters (Mandatory)
   The filters to AND together.

2.7.6.7 ChangePeriodFilter

A Filter defining a time window, any target data items that have changed in that time window shall be returned.

Inherits from Filter.

Values:

a. `dateTime` changePeriodStart (Mandatory)
   The date and time before which changes should not be retained.
b. `dateTime` changePeriodEnd (Mandatory)
   The date and time after which changes should not be retained.

2.7.6.8 GeometryFilter

A Filter defining a GML polygon. Airspace data that intersects this polygon or data relevant to such airspace shall be returned by this filter.

Inherits from Filter.

Values:

a. `gml:Polygon` region (Mandatory)
   The geometry describing the 2D filter area to be retained.

2.7.6.9 InterestedIntervalFilter

A Filter defining a time window, any target data items that have an applicability that intersects with the time window shall be returned.

Inherits from Filter.

Values:

a. `dateTime` timeInterestedStart (Mandatory)
   The earliest date and time that data items must applicable on or after in order to be retained.
b. `dateTime` timeInterestedEnd (Mandatory)
   The latest date and time that data items must be applicable on or before to be retained.

2.7.6.10 MissionIDFilter

A Filter defining a series of MissionIDs to be retained.

Inherits from Filter.

Values:

a. `MissionID[]` missions (Mandatory)
   The mission IDs to be retained.

2.8 Semantic correspondence of information definition

The Information Definition found in section 2.7 conforms with the semantics of the ATM Information Reference Model (AIRM) version 1.0.0.

The semantic correspondence report is available in Annex F in support of this statement. The report was created in accordance with the EUROROCONTROL SWIM Specification for Information Definition.

2.9 Error handling

Error handling between the two systems shall exchange comprehensive and unambiguous descriptions of problems, providing a complete description of the cause and consequence of issues. This approach will facilitate effective troubleshooting and problem resolution.

An “error” entity shall contain more than just the text which describes the issue which has occurred; a number of other features shall be available which will support effective management of, and reporting on, errors. Some considerations for what features an error entity might support such a strategy are described below.

2.9.1 Category

An error shall be associated with a “category” field. The category field will allow for easy classification for the interested party, and also allow for effective filtering and reporting on recorded errors. The available categories shall be subject to customisation, but a non-exhaustive proposed basic set is described here.

- Authorisation
- Connectivity
- Versioning
- Static Data
- Reservation
- Event
- Mission
- Conflict
- Proposal
- Other

2.9.2 Type

An error shall be associated with a “type” field. The type field will allow for ease of identification of a general cause of the error, and also allow for effective filtering and reporting.

The available categories shall be subject to customisation, but a non-exhaustive proposed basic set is described here.

- INVALID_DATA_TYPE
- INVALID_DATA_VALUE
- INVALID_COLLECTION_SIZE
- INVALID_FILTER
- CANNOT_BE_NULL
- MUST_BE_NULL
- VERSION_CONFLICT
- SESSION_EXPIRED
- OBJECT_NOT_FOUND
- OTHER

2.9.3 Relevant Data

If the error is data related, and the data size is below a configurable threshold, then the data shall be incorporated into the message.

If the error is caused by an invalid choice being submitted, then the list of possible values for the choice shall be included in the error message.

2.9.4 Platform Specification

A description of the platform on which the system reporting the error may optionally be provided. This may provide additional information to assist in troubleshooting activities. Such information might include some of the following:

- Build identifier
- Operating System name and version number
- Names and version numbers of dependencies

2.10 Service Quality - Non-functional Considerations

There are a number of non-functional requirements which need to be taken into account when considering the exchange of ASM data, as they are relevant to any system development. These type of requirements typically are concerned with characteristics such as availability and performance, and can be used as measurements of a system's efficiency, effectiveness and continued viability when comparing actual performance with expected performance.

In order for these non-functional requirements to be defined as concrete testable requirements, significant analysis of the entire environment would need to be carried out to ascertain the basic nature of the infrastructure and the predicted user activity.

The definition of these non-functional is out of scope of this document.

A non-exhaustive list of some of the features which might directly influence the non-functional aspects is given below.

- Predicted load (typical and extreme)
- Number of concurrent users (typical and extreme)
- Available network bandwidth
- Hardware specifications
- Security policy restrictions

Some of the associated non-functional requirements which need to consequently be considered are described in the following section.

2.10.1 Reliability: availability and recoverability

The reliability of the service is how it performs its functions under specified conditions for a specified period of time. Two main requirements should be agreed between all interested parties: availability and recoverability.

The expected availability of the service is the degree to which it is operational and accessible when required for use. . Whilst the ideal goal is a consistent 24/7/365 service provision, this is not always a realistic target due to the potential need for disaster recovery, maintenance or upgrades. As well as agreeing on the acceptable levels of (non)availability, contingency plans need also to be agreed in order that operations can continue seamlessly and safely when any downtime is required.

Recoverability concerns how the service is re-established in the event of an interruption or a failure.

Monitoring and measuring these characteristics provides a means to identify where improvements are required in terms of reliability.

2.10.2 Performance efficiency: capacity and time behaviour

The performance efficiency of the service is key to assessing the quality of the service. Performance requirements to be agreed between all interested parties cover the capacity of the service and its response times.

The capacity of the service, i.e., the expected maximum rate at which the service can process transactions and the maximum message size of responses must be specified. The requirement can include the number of items that can be stored, the number of concurrent users, the communication bandwidth, throughput of transactions, and size of messages.

The expected time bahaviour of the service, i.e., the processing/response time, must be specified. This can be expressed as an indication of a maximum time needed for the service provider to complete the request, measured from the time instant the service provider receives the request to the time instant the service provider sends the response or makes it available.

2.10.3 Security: confidentiality and integrity

Information being managed within an ASM Support System is highly sensitive and is to be managed with utmost care in terms of security.

The expected security mechanisms should be agreed between all interested parties. Requirements should be expressed for confidentiality and integrity.

Confidentiality ensures that information is accessible only to those authorized to have access. Integrity is an assurance that unauthorized access to, or modification of, an information service interface or information is prevented.

Mechanisms and strategies for achieving confidentiality and ensuring integrity are numerous and various in nature. Whilst considering how best to achieve maximum levels of security, it is important also to consider that technology choices will not be so bespoke or idiosyncratic that they prevent potential future users from taking up the system.

2.10.4 Extensibility

The ASM Support System is extensible. This means that new features can be added. This places requirements on the design. It must be adaptable over time and remain interoperable between two major versions. Incompatibility between software versions is inevitable as systems mature and improve over their lifetime, but it is a priority to minimise disruption between upgrades. Maintaining a stable interface between two major versions would allow the wider user community the opportunity to upgrade their software at the most suitable time, rather than enforcing the need for updating as soon as a new release becomes available.

2.11 Service behaviour

2.11.1 Subscription Behaviour

At initial connection:

- The Service Consumer authenticates itself and connects to the Service
- The Service Consumer subscribes to data applying the required filters
- The Service authorises the use of data by the Service Consumer
- The Service Consumer requests all data of interest from the ASM Support System
- As data changes the Service publishes data to the Service Consumer

It is down to the implementation as to whether or not subscriptions need to be remade at any other subsequent connection.

At any other subsequence connection (If the subscriptions need to be remade):

- The process follow the process at initial connection

At any other subsequence connections (If the subscriptions do not need to be remade)

- The Service Consumer authenticates itself and connects to the Service Provider
- The data subject to the Consumer's subscription is provided to the Service Consumer

2.11.2 Booking and Booking Conflict Behaviour

The Service Consumer is creating a booking in an airspace structure managed by the ASM Support System on the Service Provider side. The Service Consumer has subscribed to all data from the Booking and Booking Conflict interface:

- The Service Consumer creates a booking in the ASM Support System using the Service;
- Conflicts and all updates of the booking, in line with the local ASM process, are provided to the Service Consumer;
- All available actions to be taken by the Service Consumer are notified to the Service Consumer; (see 2.7.4.9 BookingAction)
- As the booking is approved in the ASM Support System the Service publishes the updates to the Booking to the Service Consumer.

The Service Consumer creates a booking without having subscribed to any data,– see 2.4.6.2 ASM Support System 1 and 2.4.7.2 ASM Support System 1 for conflicts, so the Service Consumer has to rely on the request-reply mechanism.

A User of the local ASM Support System creates a booking on the Service Provider side. The Service Consumer has subscribed to all data from the Booking and Booking Conflict interface:

- At the moment the booking is created by the User of the local ASM Support System and is within the Service Consumer filters, the booking information is provided by the Service Provider to the Service Consumer.

A User of the local ASM Support System creates a booking on the Service Provider side. The Service Consumer is not subscribed to any data – see 2.4.6.2 ASM Support System 3, so the Service Consumer has to rely on the request-reply mechanism.

2.11.3 Airspace Negotiation Behaviour

The Service Consumer creates a booking proposal on a booking managed by the Service Provider. The Service Consumer has subscribed to all data from the negotiation interface and has the booking ID and the action to propose:

- The Service Consumer creates a booking proposal in the ASM Support System on the Service Provider side;
- When the booking proposal is accepted or rejected the Service Provider publishes the update to the proposal and booking, if accepted, to the Service Consumer.

The Service Consumer creates a booking proposal on a booking managed by the Service Provider without having subscribed to any data – see 2.4.9.2 External User 1.

A User of the local ASM Support System creates a booking proposal, the Service Consumer rejects it. The Service consumer has subscribed to all data from the booking and negotiation interfaces:

- The Service Provider publishes the new booking proposal and handle proposal action to the Service Consumer;
- The Service Consumer rejects the booking proposal;
- The Service Provider publishes the booking proposal as handled to the Service Consumer. The booking is not changed and so not published.

A User of the local ASM Support System creates a booking proposal, the Service Consumer accepts it. The Service Consumer has subscribed to all data from the booking and negotiation interfaces:

- The Service Provider publishes the new booking proposal and handle proposal action to the Service Consumer;
- The Service Consumer accepts the booking proposal;
- The Service Provider publishes the booking proposal as handled to the Service Consumer. The booking is updated to reflect the proposal and published to the Service Consumer.

The Service Consumer handles a booking proposal on a booking they introduced via the Service Provider. The Service Consumer is not subscribed to any data – see 2.4.9.2 External User 2.

2.11.4 Local Airspace Structure Behaviour

Introduction of a new airspace structure in the ASM Support System on the Service Provider side. The Service Consumer has subscribed to all data from the airspace structure interface:

- At the moment the new airspace structure is created and is within the Service Consumer filters, the airspace structure data and activity data is provided by the Service Provider to the Service Consumer.

Introduction of a new airspace structure in the ASM Support System on the Service Provider side. The Service Consumer is not subscribed to data from the airspace structure interface – see 2.4.5.2.

2.11.5 Airspace Status Behaviour

Airspace status information. The Service Consumer has subscribed to all data from the Airspace Status interface.

- The current status of the airspace is immediately published to the Service Consumer when the subscription is started.
- The Service Consumer acknowledges each airspace status.
- All updates to the status of airspace structures within the filters of the Service Consumers are provided in real time to the Service Consumer by the Service Provider.
- As each update is received the Service Consumer acknowledges the changed airspace status.

Airspace status information. The Service Consumer is not subscribed to data from the Airspace Status interface – see 2.4.10.2.

2.11.6 Mission Behaviour

Creation and update of a new mission by the Service Consumer via the ASM Service:

- The Service Consumer creates a mission via the Service;
- The created mission is returned including the mission ID;
- The Service Consumer updates the previously created mission using the mission ID.

A User of the local ASM Support System creates a mission. The Service Consumer has subscribed to all data from the mission interface:

- The Service Provider publishes the new mission to the Service Consumer.

A User of the local ASM Support System creates a mission. The Service Consumer is not subscribed to any data from the mission interface – see 2.4.8.2 ASM Support System 2.

2.11.7 Long Term Planning Behaviour

The Service Consumer creates and updates a new long term Event – see 2.4.4.2 External User 1:

- The Service Consumer creates a long term Event via the Service Provider;
- The created Event is returned including the event ID by the Service Provider;
- The Service Provider updates the previously created event using the event ID.

A User of the local ASM Support System creates an event. The Service Consumer has subscribed to all data from the Long Term Planning interface:

- The Service Provider publishes the new event to the Service Consumer.

A User of the local ASM Support System creates an event. The Service Consumer is not subscribed to any data from the Long Term Planning interface – see 2.4.4.2 ASM Support System 1.

ANNEX A - Traceability to regulatory requirements

This annex contains traceability tables between relevant European legislation and the ASM Support Systems Interfaces Specification.

The first column identifies the individual regulation.

The second column identifies regulatory requirements where this specification's adoption can support compliance

The third column identifies requirements (document section) in the ASM Support Systems Interfaces Specification that can support compliance to regulation.

Legislation	Regulatory Requirement (reference)	Specification requirement (Section)
Commission Implementing Regulation (EU) No 2021/116 of 1 February 2021 (The CP1 Regulation)	Article 3, Paragraphs 1 (c) and 2 Annex, Section 3, Paragraph 3.1.1 Annex, Section 5	2.4, 2.5, 2.7, 2.9, 2.10, Annex G

Table 3: Traceability to regulatory requirements

ANNEX B – Traceability to operational requirements from Annex 11 / ASM Handbook Ed. [5.10]

The local/FAB ASM Support System requirements described in this specification are in line with the baseline process described in the Annex 11 of the ASM Handbook Chapter 2.2.

Focusing on the local/Fab ASM process the following operational requirements described in Annex 11 are relevant for this Specification.

The tables provides traceability to the operational requirements in Annex 11

1. ASM Reference database

Requirement ID	Description	Remarks	Specification Requirement Part I	Specification Requirement Part II
SSA-DB-006	ASM/ATFCM stakeholders SHALL be able to define ad-hoc areas in both local and NM systems.	Online creation of airspace structures.	ASM-DB-FUN-030	N/A
SSA-DB-007	ASM/ATFCM stakeholders SHALL be able to visualise 3D maps representing airspace data at a given effective date.		ASM-DB-FUN-100 ASM-DB-FUN-140	N/A
SSA-DB-008	ASM/ATFCM stakeholders SHOULD be able to define		ASM-DB-FUN-	N/A

Requirement ID	**Description**	**Remarks**	**Specification Requirement Part I**	**Specification Requirement Part II**
	business rules to be associated to airspace structures.		050	
SSA-DB-009	ASM/ATFCM stakeholders SHOULD be able to visualise pre-defined airspace scenarios in both local and NM systems.		ASM-DB-FUN-230; ASM-DB-FUN-260	N/A
SSA-DB-010	ASM/ATFCM stakeholders SHALL be able to manage complex FUA restrictions and FUA/RAD restrictions, composed by different sub-restrictions.		ASM-FRA-CON-030	N/A
SSA-DB-011	ASM/ATFCM stakeholders SHALL be able to express airspace availability and activation referring to either Sunrise or Sunset.		ASM-DB-FUN-010	N/A
SSA-DB-012	ASM/ATFCM stakeholders SHALL be able to manage the partial/total overlap of areas/FBZs reservations.		ASM-DB-FUN-200 ASM-DB-FUN-210 ASM-DB-FUN-215	N/A
SSA-DB-013	ASM/ATFCM stakeholders SHALL be able to retrieve/manage pre-defined airspace scenarios using a		**ASM-DB-FUN-130**	

Requirement ID	Description	Remarks	Specification Requirement Part I	Specification Requirement Part II
	common ASM Scenario Repository.		**ASM-DB-FUN-140**	
SSA-DB-014	ASM/ATFCM stakeholders SHALL be able to express airspace availability and activation referring to different UOM		ASM-DB-FUN-160	N/A
SSA-DB-015	ASM/ATFCM stakeholders SHALL be able to define FUA restrictions groups to process via AUP/UUP.		**ASM-DB-FUN-280**	

Table 4: Traceability to ASM reference database requirements

2. Planning

Req. ID	Description	Remarks	Specification Requirement Part I	Specification Requirement Part II
SSA-PL-001	AMCs SHOULD be able to retrieve by automated means information on the planned utilisation of airspace reservation (TRA, TSA, CBA) (e.g. 7 days in advance).	Rolling AUP.	ASM-DB-FUN-130 ASM-DB-FUN-140 ASM-DB-FUN-150	N/A
SSA-PL-002	Daily submission of airspace reservation and conditional route requests to AMCs by approved agencies SHOULD be made through a common interface using a data exchange format supported by the NM ENV database.	Publication of eAMI via B2B publish/ subscribe.	ASM-DB-FUN-285	N/A
SSA-PL-004	Authorised users/units SHOULD be able to book an ad-hoc area, according to common standards and coordination procedures.	Dynamic airspace allocation at levels 2 and 3.	ASM-DB-FUN-030; ASM-DB-FUN-060	N/A
SSA-PL-005	Authorised users/units SHALL have an assigned restricted access for planning purposes.	User authentication.	ASM-DB-FUN-040	N/A

SSA-PL-006	Authorised users/units SHALL be able to create/modify/delete area reservations within the lateral and vertical limits of an airspace structure (or a combination of).		ASM-DB-FUN-040;
			ASM-DB-FUN-060;
			ASM-DB-FUN-140;
			ASM-DB-FUN-150
				N/A

Req. ID	**Description**	**Remarks**	**Specification Requirement Part I**	**Specification Requirement Part II**
SSA-PL-007	Authorised users/units SHALL be able to create/modify/delete the following minimum set data for planning request into the local system: • Reference number; • Start/end date and time; • 3D airspace block (airspace ID, vertical extension); • Responsible unit; • Requestor (PoC).		ASM-DB-FUN-160	N/A

Req. ID	Description	Remarks	Specification Requirement Part I	Specification Requirement Part II
SSA-PL-008	Authorised users/units SHOULD be able to enter/modify/delete additional static and dynamic data for planning request into the local system: • ADEP; • ADES; • Aircraft type; • Number of aircraft; • Priority; • Call sign(s); • Mission ID; • Link to other missions; • Controlling Units; • Remarks.	FBZ and FUA/EU restrictions activation.	ASM-DB-FUN-180	N/A

Req. ID	**Description**	**Remarks**	**Specification Requirement Part I**	**Specification Requirement Part II**
SSA-PL-009	Authorised users/units, within their area of responsibility, MAY be able to create/modify/delete alternative options into the local system, in terms of: • Time shifting; • Different airspace blocks with or without FBZ/FUA restrictions; • Different FUA restrictions and/or FUA/RAD restrictions and/or FUA restrictions groups for the requested areas; • Alternate areas or feasible flying distance from the A/B and requested shape in order to identify alternate areas.		ASM-DB-FUN-220 **ASM-DB-FUN-280**	N/A
SSA-PL-016	Authorised users/units MAY be able to enter an airspace request (airspace/route booking, amendment, cancellation) into the local system, on behalf of other users/units, following relevant FAB rules if applicable.		ASM-FAB-FUN-010 ASM-FAB-FUN-020 ASM-FAB-FUN-030	N/A
SSA-PL-017	Authorised users/units SHALL be prompted with an acknowledgement message once an airspace request (airspace/route booking, amendment, cancellation) is successfully entered into the local system.		ASM-DB-FUN-190; ASM-DB-FUN-100	N/A

Req. ID	Description	Remarks	Specification Requirement Part I	Specification Requirement Part II
SSA-PL-018	Authorised users/units SHALL be able to retrieve a log of all the recorded airspace requests (airspace/route booking, amendment, cancellation) entered into the local system.		ASM-DB-OPS-060	N/A
SSA-PL-019	Authorised users/units MAY be able to visualise both the airspace requests (airspace/route booking, amendment, cancellation) entered into the local system either by themselves or other users/units.		ASM-DB-FUN-190; ASM-DB-FUN-100	N/A
SSA-PL-020	Authorised users/units SHALL be able to delete a request from the local system.		ASM-DB-FUN-130	N/A
SSA-PL-021	Authorised users/units SHALL be able to visualise any modifications occurred to any airspace requests.		ASM-DB-FUN-190; ASM-DB-FUN-100	N/A
SSA-PL-022	Authorised users/units SHALL be prompted with error messages in case of validation errors and database inconsistencies, related to airspace requests under their responsibility.		ASM-DB-OPS-160	N/A
SSA-PL-023	Authorised users/units SHALL be prompted with warning messages in case of conflicts detected related to airspace requests under their responsibility.		ASM-DB-FUN-210	N/A
SSA-PL-024	Authorised users/units SHALL be able to visualise 4D maps (3D representation + time) of any airspace requests and associated events.		ASM-DB-FUN-100	N/A

Req. ID	Description	Remarks	Specification Requirement Part I	Specification Requirement Part II
SSA-PL-025	Authorised users/units SHALL be able to receive feedbacks on the evolution of airspace requests.		ASM-DB-FUN-190	N/A
SSA-PL-026	Authorised users/units SHALL be able to create/visualise/accept/reject change proposals to airspace requests on their own responsibility.		ASM-DB-FUN-220	N/A
SSA-PL-028	Authorised users/units SHALL be able to generate NOTAM proposals associated to an airspace request to be published when required.		ASM-DB-FUN-300	N/A
SSA-PL-029	AMCs SHOULD be able to retrieve by automated means information on the foreseen establishment of specific airspaces required for major military exercises (e.g. 3 months in advance).		ASM-DB-FUN-110	N/A
SSA-PL-030	AMCs SHALL be able to publish AUP information from D-6 to D-2 whenever available.	Rolling AUP	ASM-DB-FUN-280 ASM-DB-FUN-285	N/A
SSA-PL-031	AMCs SHALL be able to activate simultaneously different sub-restrictions.	Complex FUA Restrictions	ASM-FRA-CON-030	N/A
SSA-PL-032	AMCs SHALL be able to make area reservations outside the AIP published times/vertical limits.		ASM-DB-FUN-070	N/A

Table 5: Planning requirements

3. Local / FAB impact assessment

Req. ID	Description	Remarks	Specification Requirement Part I	Specification Requirement Part II
SSA-LOC-001	AMCs involved in cross-border operations SHOULD be able to exchange airspace request information, including: • AIS static and dynamic data; • reservations and missions information; • alternative operational solutions proposals.		N/A	2.4.10 2.4.5 2.4.6 2.4.8 2.4.7 2.4.9 2.4.11
SSA-LOC-002	The Lead AMC SHOULD be able to manage airspace requests submitted by approved agencies from a different State.		ASM-FAB-FUN-020	N/A
SSA-LOC-003	AMCs WILL be able to propose alternative local/FAB airspace allocation options.		ASM-DB-FUN-230 ASM-DB-FUN-240 ASM-DB-FUN-250 ASM-DB-FUN-260	2.4.9

Req. ID	Description	Remarks	Specification Requirement Part I	Specification Requirement Part II
SSA-LOC-004	AMCs WILL be able to perform a local/FAB impact assessment based on what-if analysis of different airspace allocation options, highlighting and solving potential conflicting requests (both spatial and temporal overlaps).	Booking de-conflicting.	ASM-DB-FUN-230 ASM-DB-FUN-240 ASM-DB-FUN-250 ASM-DB-FUN-260	N/A
SSA-LOC-005	AMCs, ACCs/FMPs concerned, military agencies, SHALL be able to visualise any airspace requests.		ASM-FAB-FUN-010 ASM-FAB-FUN-020 ASM-FAB-FUN-030	N/A

Req. ID	Description	Remarks	Specification Requirement Part I	Specification Requirement Part II
SSA-LOC-010	AMCs SHALL be able to process the following data in order to support local/FAB impact assessments: • Route structures; • Manageable and not manageable Area Structures; • NPZs, as required; • CDR route status (past, current, proposed and future); • AIP data associated with each route; • Area status (past, current, proposed and future); • AIP data associated with each area; • NOTAMs; • Airport Information; • FUA Restrictions; • FUA/RAD Restrictions groups; • FBZ Activation; • NPZ Activation.		ASM-FAB-FUN-010 ASM-FAB-FUN-030 ASM-FAB-FUN-310 ASM-FRA-CON-020 ASM-FRA-CON-030 ASM-FRA-OPS-010 ASM-FRA-OPS-020 ASM-FRA-OPS-030 ASM-DB-FUN-340	N/A
SSA-LOC-012	The Lead AMC SHALL have unique access and right of managing assigned cross borders airspace structures.	Unique ASM Authority for cross border airspace structures	ASM-FAB-OPS-020	N/A

Table 6: Traceability Local / FAB impact assessment requirements

4. Airspace Status

Req. ID	Description	Remarks	Specification Requirement Part I	Specification Requirement Part II
SSA-AS-001	ASM/ATFCM stakeholders SHALL be able to exchange airspace status information through a B2B publish/subscribe mechanism.	Publication of eAMI via B2B publish/ subscribe mechanism.	N/A	2.4.11
SSA-AS-003	ASM/ATFCM stakeholders MAY be able to visualise 4D maps (3D representation + time) of the real time status of airspaces.		ASM-DB-FUN-340	N/A
SSA-AS-004	Local ASM units MAY be able to exchange real time airspace status data with ATC systems.		N/A	2.4.11
SSA-AS-005	ASM responsible units SHOULD be able to confirm and acknowledge any changes to the status of an airspace structure (i.e. activation, de-activation, cancellation, amendment).		ASM-DB-FUN-320; ASM-DB-FUN-330	2.4.12
SSA-AS-007	ASM responsible units SHALL be able to introduce any changes to the status of an airspace structure (i.e. activation, de-activation, cancellation, amendment).		ASM-DB-FUN-320	N/A

5. General

Req. ID	Description	Remarks	Specification Requirement Part I	Specification Requirement Part II
SSA-GEN-004	ASM/ATFCM stakeholders SHALL be able to manage any airspace structures and conditions (e.g. FUA restrictions) subject to ASM/ATFCM process.		ASM-DB-OPS-060 ASM-DB-OPS-070 ASM-DB-FUN-030 ASM-FRA-CON-030 ASM-FRA-OPS-020 ASM-FRA-OPS-030 ASM-FRA-FUN-020	N/A
SSA-GEN-006	ASM/ATFCM stakeholders SHALL be able to exchange airspace status information, ensuring a common situational awareness at all times.		ASM-DB-FUN-340 ASM-FAB-FUN-040	2.4.11
SSA-GEN-011	ASM/ATFCM stakeholders SHALL be prompted with error/warning messages in case of exceptions (e.g. business rules violation; process deviation).		ASM-DB-OPS-160	N/A
SSA-GEN-012	ASM/ATFCM stakeholders SHALL be able to monitor systems operational status.		ASM-DB-CON-060	N/A
SSA-GEN-013	ASM/ATFCM stakeholders SHALL be at any time able to unambiguously assign an airspace structure to only one ASM		ASM-FAB-OPS-020	N/A

Req. ID	Description	Remarks	Specification Requirement Part I	Specification Requirement Part II
	Authority.			
SSA-GEN-014	ASM/ATFCM as well as AO/CFSP stakeholders SHALL be able to support the different phases of ASM/ATFCM CDM processes (as described in ERNIP Part 3 - ASM Handbook) by exchanging and processing ASM/ATFCM data.		ASM-DB-OPS-060 ASM-FAB-CON-030 ASM-DB-FUN-220 ASM-DB-FUN-230 ASM-DB-FUN-240 ASM-DB-FUN-250 ASM-DB-FUN-260	2.4.6 2.4.7 2.4.9

Table 7: Traceability to General requirements

ANNEX C – CONFORMITY WITH SWIM REQUIREMENTS

EUROCONTROL-SPEC-168 SWIM SERV Ed 2.0			**CONFORMITY VERIFICATION**	
Identifier	**Title**	**Implementation**	**Conform**	**Justification (Section, paragraph number)**
General Service Description Requirements				
SWIM-SERV-010	Service description coverage	M	Yes	2.1.1
SWIM-SERV-020	Service description language	M	Yes	whole document
SWIM-SERV-030	Service description identification	M	Yes	2.1.2
General Service Information				
SWIM-SERV-040	Service identification	M	Yes	2.1.2
SWIM-SERV-050	Service abstract	M	Yes	2.1.1
Service Provider and Contact Information				
SWIM-SERV-060	Service provider	M	N/A	2.6
SWIM-SERV-070	Provider point of contact	R	N/A	2.6
SWIM-SERV-080	Support availability	R	N/A	2.6
Service Characteristics				
SWIM-SERV-090	Geographical extent of information	M	N/A	2.6
SWIM-SERV-100	Service categories	M	Yes	2.1.3
SWIM-SERV-110	Service lifecycle information	M	N/A	2.6
SWIM-SERV-120	Service standard reference	M Conditional	N/A	2.6
High-level Description of Service Offer				
SWIM-SERV-130	Operational environment	M	Yes	2.2, Annex B,
SWIM-SERV-140	Service functions	M	Yes	2.3
Limitations and Constraints on Using the Service				
SWIM-SERV-150	Service access and use conditions	M	Yes	2.6
SWIM-SERV-160	Security constraints	M	Yes	2.6
SWIM-SERV-170	Additional technical information for the service consumer	M Conditional	Yes	2.6
Quality Aspects				
SWIM-SERV-180	Quality of service	M	Yes	2.6
SWIM-SERV-190	Source of information	R	N/A	2.6

SWIM-SERV-200	Service validation information	M	N/A	2.6
Behaviour of the Service				
SWIM-SERV-210	Application message exchange pattern	M	Yes	2.5.1
SWIM-SERV-220	Service behaviour	M	Yes	2.11
SWIM-SERV-230	Service monitoring	M Conditional	Yes	2.10
Service Implementation and Structural Details				
SWIM-SERV-240	Service interfaces	M	Yes	2.4
SWIM-SERV-250	SWIM TI Profile and interface bindings	M	Yes	2.5.1
SWIM-SERV-260	Service interface protocols and data format	M	Yes	2.5.1
SWIM-SERV-270	Service operations	M	Yes	2.4.2.7, 2.4.3.4, 2.4.4.4, 2.4.5.4, 2.4.6.4, 2.4.7.4, 2.4.8.4, 2.4.9.4, 2.4.10.4.
SWIM-SERV-280	Service messages	M	Yes	2.7
Information Aspects of the Service				
SWIM-SERV-290	Information definition (minimum)	M	Yes	2.7
SWIM-SERV-300	Information definition (extended)	R	Yes	2.7
SWIM-SERV-310	Filter encoding	M Conditional	Yes	2.3.3.3, 2.7.6
Resources				
SWIM-SERV-320	Machine-readable service interface definition	M Conditional	Yes	2.5.2
SWIM-SERV-330	Model view	R	Yes	2.3, 2.4.1, 2.4.2, 2.4.3, 2.4.4, 2.4.5, 2.4.6, 2.4.7, 2.4.8, 2.4.9, 2.4.10, 2.4.11, 2.4.12
SWIM-SERV-340	Examples of Code	R	N/A	2.6
SWIM-SERV-350	Abbreviations and acronyms	M	Yes	1.8

Table 8: Conformity with EUROCONTROL-SPEC-168 SWIM SERV Ed 2.0

EUROCONTROL-SPEC-169 SWIM INFO Ed 1.0			**CONFORMITY VERIFICATION**	
Identifier	**Title**	**Implementation**	**Conform**	**Justification (Section, paragraph number)**
General Requirements				
SWIM-INFO-001	Need for information definitions	M	Yes	2.7
SWIM-INFO-002	Information definition language	M	Yes	whole document
SWIM-INFO-003	Information definition identification	M	Yes	2.7

SWIM-INFO-004	Information definition responsible party	M	Yes	2.7
SWIM-INFO-005	Information definition scope	M	Yes	2.7
SWIM-INFO-006	Information definition namespace	M	Yes	2.7
SWIM-INFO-007	Information definition concepts	M	Yes	2.7
SWIM-INFO-008	Unique identifiers for concepts	M	N/A	N/A
SWIM-INFO-009	Preservation of meaning	M Conditional	Yes	2.7
SWIM-INFO-010	Principles for definitions for concepts	R	N/A	N/A
SWIM-INFO-011	Semantics of metadata	R	Yes	2.7
SWIM-INFO-012	Use of data types	M	Yes	2.7
Requirements for semantic correspondence				
SWIM-INFO-013	Establish semantic correspondence	M	Yes	2.8, Annex F
SWIM-INFO-014	Forms of semantic correspondence	M	Yes	2.8, Annex F
SWIM-INFO-015	Out-of-scope and no semantic correspondence established declarations	M	Yes	2.8, Annex F
SWIM-INFO-016	Mapping of information concepts to the matching AIRM concept	M	Yes	2.8, Annex F
SWIM-INFO-017	Mapping of data concepts to the matching AIRM concepts	M	Yes	2.8, Annex F
SWIM-INFO-018	Additional traces to clarify the mapping of narrower concepts	M	Yes	2.8, Annex F
SWIM-INFO-019	Use of the AIRM's unique identifiers in traces	M	Yes	2.8, Annex F

Table 9: Conformity with EUROCONTROL-SPEC-169 SWIM INFO Ed 1.0

Conformity with EUROCONTROL-SPEC-170 SWIM TI Yellow Profile is achieved through the following Requirements:

- ASM-INTF-TIP-010
- ASM-INTF-TIP-020
- ASM-INTF-TIP-030
- ASM-INTF-TIP-040

which are set out is Section 2.5 of this Specification.

Page left blank intentionally

ANNEX D – CONFORMITY CHECKLIST

This annex summarises the requirements to be met when assessing conformity to this specification.

Table 10 lists each requirement in the specification using its identifier and title. It then states the level of implementation to be achieved (see Table 10). In some cases, the implementation is conditional which means that the requirement is to be implemented when the condition applies.

Level of Implementation	Operative verb used in the requirement
M = Mandatory	**shall**
R = Recommended	**should**
O = Optional	**may**

Table 10: Level of implementation

Identifier	Level of implementation
ASM-INTF-FIL-010	M
ASM-INTF-SUBS-010	M
ASM-INTF-SUBS-020	M
ASM-INTF-SUBS-030	M
ASM-INTF-SUBS-040	M
ASM-INTF-SUBS-050	M
ASM-INTF-SUBS-070	M
ASM-INTF-SUBS-080	M
ASM-INTF-SUBS-090	M
ASM-INTF-SUBS-100	M
ASM-INTF-SUBS-110	M
ASM-INTF-SUBS-115	R
ASM-INTF-SUBS-120	M
ASM-INTF-SUBS-130	M
ASM-INTF-SUBS-140	M
ASM-INTF-SUBS-150	M
ASM-INTF-SUBS-160	M
ASM-INTF-SUBS-163	R
ASM-INTF-SUBS-165	R
ASM-INTF-SUBS-167	R

Identifier	Level of implementation
ASM-INTF-SUBS-168	M
ASM-INTF-SUBS-170	M
ASM-INTF-SUBS-180	M
ASM-INTF-SUBS-190	M
ASM-INTF-SUBS-200	M
ASM-INTF-SUBS-210	M
ASM-INTF-SUBS-215	O
ASM-INTF-SUBS-217	R
ASM-INTF-SUBS-220	M
ASM-INTF-SUBS-230	M
ASM-INTF-SUBS-240	M
ASM-INTF-SUBS-250	M
ASM-INTF-SUBS-260	M
ASM-INTF-SUBS-270	M
ASM-INTF-SUBS-280	M
ASM-INTF-SUBS-290	M
ASM-INTF-PUB-010	M
ASM-INTF-PUB-020	M
ASM-INTF-PUB-025	R
ASM-INTF-PUB-030	M
ASM-INTF-PUB-040	M
ASM-INTF-PUB-050	M
ASM-INTF-PUB-060	M
ASM-INTF-PUB-070	M
ASM-INTF-PUB-080	M
ASM-INTF-PUB-090	M
ASM-INTF-PUB-100	M
ASM-INTF-PUB-110	M
ASM-INTF-PUB-120	M
ASM-INTF-PUB-130	M
ASM-INTF-LTPL-010	R
ASM-INTF-LTPL-020	M
ASM-INTF-LTPL-030	M
ASM-INTF-LTPL-040	M
ASM-INTF-LTPL-050	M

Identifier	Level of implementation
ASM-INTF-LTPL-060	M
ASM-INTF-LTPL-070	M
ASM-INTF-LTPL-080	M
ASM-INTF-LTPL-090	M
ASM-INTF-LTPL-100	M
ASM-INTF-LTPL-110	M
ASM-INTF-LTPL-120	M
ASM-INTF-LAS-010	M
ASM-INTF-LAS-020	M
ASM-INTF-LAS-030	M
ASM-INTF-LAS-040	M
ASM-INTF-LAS-050	M
ASM-INTF-LAS-055	M
ASM-INTF-LAS-060	M
ASM-INTF-LAS-070	M
ASM-INTF-LAS-080	M
ASM-INTF-LAS-085	M
ASM-INTF-ARES-010	M
ASM-INTF-ARES-020	M
ASM-INTF-ARES-030	M
ASM-INTF-ARES-040	M
ASM-INTF-ARES-050	M
ASM-INTF-ARES-060	M
ASM-INTF-ARES-070	M
ASM-INTF-ARES-080	M
ASM-INTF-ARES-090	M
ASM-INTF-ARES-100	M
ASM-INTF-ARES-110	M
ASM-INTF-ARES-120	R
ASM-INTF-ARES-130	M
ASM-INTF-ARES-140	M
ASM-INTF-ARES-150	M
ASM-INTF-ARES-160	M
ASM-INTF-ARES-170	M
ASM-INTF-ARES-180	R

Identifier	Level of implementation
ASM-INTF-ARES-190	M
ASM-INTF-ARES-200	M
ASM-INTF-ARES-210	M
ASM-INTF-ARES-220	M
ASM-INTF-ARES-230	M
ASM-INTF-ARES-240	M
ASM-INTF-ARES-250	M
ASM-INTF-ARES-260	M
ASM-INTF-ARES-270	M
ASM-INTF-ARES-280	M
ASM-INTF-CON-010	R
ASM-INTF-CON-020	M
ASM-INTF-CON-030	M
ASM-INTF-CON-040	M
ASM-INTF-CON-050	M
ASM-INTF-CON-060	M
ASM-INTF-CON-070	M
ASM-INTF-MIS-010	R
ASM-INTF-MIS-020	M
ASM-INTF-MIS-030	M
ASM-INTF-MIS-040	M
ASM-INTF-MIS-050	M
ASM-INTF-MIS-060	M
ASM-INTF-MIS-070	M
ASM-INTF-MIS-080	M
ASM-INTF-MIS-090	R
ASM-INTF-MIS-100	M
ASM-INTF-MIS-110	M
ASM-INTF-MIS-120	M
ASM-INTF-MIS-130	M
ASM-INTF-NEG-010	R
ASM-INTF-NEG-020	M
ASM-INTF-NEG-030	M
ASM-INTF-NEG-040	M
ASM-INTF-NEG-050	M

Identifier	Level of implementation
ASM-INTF-NEG-060	M
ASM-INTF-NEG-070	M
ASM-INTF-NEG-080	M
ASM-INTF-NEG-085	M
ASM-INTF-NEG-090	M
ASM-INTF-NEG-100	M
ASM-INTF-NEG-110	M
ASM-INTF-NEG-120	R
ASM-INTF-STAT-010	M
ASM-INTF-STAT-020	M
ASM-INTF-STAT-030	M
ASM-INTF-STAT-040	M
ASM-INTF-STAT-050	M
ASM-INTF-STAT-055	M
ASM-INTF-STAT-060	M
ASM-INTF-STAT-070	M
ASM-INTF-STAT-080	R
ASM-INTF-STAT-090	M
ASM-INTF-STAT-100	M
ASM-INTF-TIP-010	M
ASM-INTF-TIP-020	M
ASM-INTF-TIP-030	M
ASM-INTF-TIP-040	M
ASM-INTF-SDR-010	M
ASM-INTF-SDR-020	M
ASM-INTF-SDR-030	M

Table 11: Conformity checklist

Page left blank intentionally

ANNEX E - SPECIFICATION UPDATE PROCEDURES

It is necessary to periodically check this EUROCONTROL Specification for consistency with referenced material, notably ICAO SARPS and relevant Regulations. The Specification is also expected to evolve following real project and field experience, as well as advances in technology.

The main objectives of a regular review are:

a) to improve the quality of the requirements (e.g. clarity, testability, etc.);
b) to verify that the level of detail published is adequate;
c) to ensure that design-oriented requirements, imposing unnecessary constraints to technical solutions, have been avoided;
d) to ensure that advances in technology are properly reflected;
e) to make all stakeholders, including industry, aware of the latest developments.

The update process for this EUROCONTROL Specification may be summarised as follows:

Stakeholders may provide change proposals either through existing working arrangements (e.g. established working groups) or by sending a formal Change Request (CR) to the generic email address: standardisation@eurocontrol.int

The CR needs to provide following minimum elements:

- Originator information (name, Organisation, contact details)
- Specification title, number and edition date
- Page, chapter, section (subsection) where the issue appears
- Description of the issue and reason for change
- Specific change proposal text (incl. potential alternatives, if any).

Main steps towards a revised version:

- Agency (Standardisation unit) will assess each CR in coordination with content owners, classify the urgency and establish the CR impact category (major, minor or editorial).
- Agency will then prepare resolution proposal(s) and, if needed, discuss those with the originator and/or relevant working arrangements. Note: CR will be grouped into “change packages” to consider reasonable update cycles.
- Agreed changes will be integrated into a revised version “Proposed Issue” including a summarised list of changes.
- Consultation will be performed in accordance with the CR impact category identified:
    - Major changes require full formal stakeholder consultation (PC level)
    - Minor changes need consultation at working layers (e.g. working group or Team)
    - Editorial changes may be implemented directly at any stage though grouped with change packages.

Note: Identified errors which may cause potential problems when implementing, may be corrected directly via separate “Corrigendum”.

The Agency will apply this process in an objective and impartial way and will consult stakeholders as needed and in line with the formal Standards Development Process.

Page left blank intentionally

ANNEX F – SEMANTIC CORRESPONDENCE REPORT

Note: The content of this annex is provided in a separate file.

Page left blank intentionally

ANNEX G – WEB SERVICES DESCRIPTION LANGUAGE (WSDL)

Note: The content of this annex is provided in a separate file.

EUROCONTROL

SUPPORTING EUROPEAN AVIATION

www.eurocontrol.int