SWIM-TI Blue Profile Technical Specification Corrigendum

Document Name	SWIM-TI Blue Profile Technical Specification Corrigendum
Document ID	D44-005.01
Edition	00.01.00
Edition Date	15/12/2016
Abstract	The present corrigendum document contains amendments to the SWIM Blue Profile Technical Specification (14.01.04 D44-005 [15]) to include improvements agreed by the IOP Anaysis experts.

Authoring & Approval

Prepared By		
Name & Company	**Position & Title**	**Date**
[redacted] LEONARDO-FINMECCANICA	[redacted]	15/12/2016
[redacted] THALES		15/12/2016
[redacted] THALES		15/12/2016

Reviewed By		
Name & Company	**Position & Title**	**Date**
[redacted] INDRA	[redacted]	09/12/2016
[redacted] THALES		09/12/2016
[redacted] INDRA		09/12/2016

Approved By		
Name & Company	**Position & Title**	**Date**
[redacted] LEONARDO-FINMECCANICA	[redacted]	15/12/2016
[redacted] THALES		15/12/2016
[redacted] INDRA		15/12/2016

Rejected By		
Name & Company	**Position & Title**	**Date**

Document History

Edition	Date	Status	Author	Justification
00.00.01	20/10/2016	Revised Draft	[redacted]	Revised draft based on SJU approved previous version (D44-005).
00.00.02	21/10/2016	Revised Draft		Implemented FO Identifier type update documented in F#9 Technical Note §3.5 (changes concerning the API have not been implemented yet). Implemented changes identified in the context of "Asynchronous notification of request failure" discussion documented in F#9 Technical Note §3.8 and §3.9 (changes concerning the API have not been implemented yet).
00.00.03	28/10/2016	Revised Draft		Implemented changes identified in the context of "Distribution Failure - Problem in payload such as

				version mismatch" discussion documented in F#9 Technical Note §3.17.3 (changes concerning the API have not been implemented yet). Implemented changes identified in the context of "Compression" discussion documented in F#9 Technical Note §3.18.
00.00.04	03/11/2016	Revised Draft	[redacted]	Implemented changes identified in the context of "Collisions or Concurrent updates of FO releases" discussion documented in F#9 Technical Note §3.11.
00.00.05	07/11/2016	Revised Draft		Updated §1 and §2. Updated §3.3.9.1. Updated §3.3.9.2.
00.00.06	22/11/2016	Revised Draft		Implemented improvements agreed during the first review cycle.
00.00.07	07/12/2016	Revised Draft		Added Table 2-1 in §2.2. Updated figures 3-1 and 3-2. Updated §3.3.9.2 (UML, new requirements, implemented agreed comments). Updated §3.3.9.1 (new requirements). Embedded final IDL and WSDL.
00.00.08	08/12/2016	Revised Draft		Open comments assessment.
00.01.00	15/12/2016	Final Version		Final Version.

Table of Contents

List of tables

List of figures

1 Introduction

1.1 Purpose of the document

The present corrigendum document contains amendments to the SWIM-TI Blue Profile Technical Specification (14.01.04 D44-005 [15]) to include improvements agreed by the IOP Anaysis experts.

The amendment contributes to SESAR Solution #28 "Initial ground-ground interoperability" and to the coming ED-133 speciciation update. It addresses Ground-Ground Interoperability between ATC-ATC and ATC-NM and constitutes a basis for further IOP prototyping and validations.

Section §3.3.9 of the present document replaces existing §3.3.9 section in the SWIM-TI Blue Profile Technical Specification. The introduction in §3 has been not updated and it is provided in this document for completeness. Improvements include, but are not limited to, a definition of logical interface between the SWIM technical infrastructure and IOP applications.

1.2 Structure of the document

This document is organized as follows:

Chapter 1: Purpose and scope, requirements guidelines, SWIM Technical Infrastructure high level overview.

Chapter 2: Overview of the context, inputs and results concerning the D44-005 corrigendum.

Chapter 3: SWIM-TI Blue Profile Interface Requirements updates.

Chapter 4: Assumptions.

Chapter 5: Referenced documents;

2 Scope of Contribution

The current work is performed in the scope of the SESAR1 IOP Analysis Team and includes contributions to Feature #9 “FO Mechanisms” and Feature #6 “FO Recovery” for the SWIM-related mechanisms.

2.1 Identified Input

The following input has been used for the analysis of current limitations and the preparation of improvements to the current SWIM Blue Profile Specigfication:

1. ED-133 requirements requiring clarification,
2. WG 59: ED-133 rev 1 open issues – 20151001.
3. Requirements not verified/validated within SESAR 1 and that will require further refinement/amendments for next ED-133 updates.
4. Issues raised during SESAR 1 verification and validation activities (VP-022 / 14.02.09 D11/D03, VP-711 / 10.02.05-D36).
5. Missing capabilities/External FDPS input.
6. SWIM TS 14.01.04 D44-005 (Blue Profile).

2.2 Changes to the SWIM Blue Profile specification

The following table details impact on the current SWIM TS introduced by the items addressed within Feature #9 and Feature #6 activities.

Item	SWIM Application Interface Layer	FlightObjectManagement Interface (WSDL)	FlightObjectDistribution Interface (IDL+QoS)	Missing Requirements	Update of existing requirements	Explanation Text/Tables/Figures to be updated/added
IOP-MFO-180 (-mdw) Requirement update	X			X	X	
API-MDW and API-APP Interfaces	X			X		
FlightObjectManagement Interface	X	X		X		
FO Recovery						X
Flight Object Identifier	X	X	X		X	X
Asynchronous notification of request failure	X	X			X	X

Collisions or Concurrent updates of FO releases	X		X	X	X	X
Distribution Failure - Problem in payload (such as version mismatch)	X	X			X	X
Avoid side-effects of not synchronized clocks						
Compression			X	X	X	X

In the table below a summary of the updated, deleted and new requirements is provided.

Table 2-1: New and Updated Requirements

Identifier	Type of assessment	Comment or Change rationale
REQ-14.01.04-TS-0901.0725	Requirement Text change	Added ReportFlightObjectServicesFailure operation to FlightObjectManagement interface.
REQ-14.01.04-TS-0901.0435	Requirement Text change	Updated operation signature, requirement title and rationale.
REQ-14.01.04-TS-0901.0900	New Requirement	Added ReportFlightObjectServicesFailure operation to FlightObjectManagement interface.
REQ-14.01.04-TS-0901.0440	Requirement Text change	Updated operation signature, requirement title and rationale.
REQ-14.01.04-TS-0901.0445	Requirement Text change	Updated operation signature, requirement title and rationale.
REQ-14.01.04-TS-0901.0796	Requirement Text change	Updated operation signature, requirement title and rationale.
REQ-14.01.04-TS-0901.0410	Requirement attribute(s) change	Updated requirement rationale.
REQ-14.01.04-TS-0901.0425	Requirement Text change	Typo in the requirement text. Improved rationale.
REQ-14.01.04-TS-0901.0910	New Requirement	Improved requirements concerning semantic and validity checks applicable to the FlightObjectManagement interface operations.
REQ-14.01.04-TS-0901.0920	New Requirement	Improved requirements concerning semantic and validity checks applicable to the FlightObjectManagement interface operations.
REQ-14.01.04-TS-0901.0930	New Requirement	Improved requirements concerning semantic and validity checks applicable to the FlightObjectManagement interface operations.
REQ-14.01.04-TS-0901.0940	New Requirement	Improved requirements concerning semantic and validity checks applicable to the FlightObjectManagement interface operations.
REQ-14.01.04-TS-0901.0950	New Requirement	Improved requirements concerning semantic and validity checks applicable to the FlightObjectManagement interface operations.

REQ-14.01.04-TS-0901.0960	New Requirement	Improved requirements concerning semantic and validity checks applicable to the FlightObjectManagement interface operations.
REQ-14.01.04-TS-0901.0970	New Requirement	Improved requirements concerning semantic and validity checks applicable to the FlightObjectManagement interface operations.
REQ-14.01.04-TS-0901.0980	New Requirement	Improved requirements concerning semantic and validity checks applicable to the FlightObjectManagement interface operations.
REQ-14.01.04-TS-0901.0990	New Requirement	Improved requirements concerning semantic and validity checks applicable to the FlightObjectManagement interface operations.
REQ-14.01.04-TS-0901.1000	New Requirement	Improved requirements concerning semantic and validity checks applicable to the FlightObjectManagement interface operations.
REQ-14.01.04-TS-0901.1010	New Requirement	Improved requirements concerning semantic and validity checks applicable to the FlightObjectManagement interface operations.
REQ-14.01.04-TS-0901.1020	New Requirement	Improved requirements concerning semantic and validity checks applicable to the FlightObjectManagement interface operations.
REQ-14.01.04-TS-0901.1030	New Requirement	Improved requirements concerning semantic and validity checks applicable to the FlightObjectManagement interface operations.
REQ-14.01.04-TS-0901.0815	Requirement Text change	Typo in the requirement text.
REQ-14.01.04-TS-0901.0460	Requirement Text change	Full name for the Flight Object identifier. Improved the rationale.
REQ-14.01.04-TS-0901.0465	Requirement Text change	Full name for the Flight Object identifier and added the stakeholder identifier. Improved the rationale.
REQ-14.01.04-TS-0901.0480	Requirement Text change	Typo.
REQ-14.01.04-TS-0901.0490	Deleted Requirement	Deletion reason: replaced by REQ-14.01.04-TS-0901.0490-UPD1, REQ-14.01.04-TS-0901.0490-UPD2 and REQ-14.01.04-TS-0901.0490-UPD3.
REQ-14.01.04-TS-0901.1040	New Requirement	Adoption of MIME type for Flight Object Cluster payload.
REQ-14.01.04-TS-0901.1050	New Requirement	Adoption of MIME type for Flight Object Cluster payload.
REQ-14.01.04-TS-0901.1060	New Requirement	Adoption of MIME type for Flight Object Cluster payload.
REQ-14.01.04-TS-0901.0826	Requirement Text change	Typo in the requirement text. Improved the rationale.
REQ-14.01.04-TS-0901.0792	Requirement Text	Typo in the requirement text.

	change	
REQ-14.01.04-TS-0901.1070	New Requirement	Added requirements concerning the Flight Object update release checking.
REQ-14.01.04-TS-0901.1080	New Requirement	Added requirements concerning the Flight Object update release checking.
REQ-14.01.04-TS-0901.1090	New Requirement	Added requirements concerning the Flight Object update release checking.
REQ-14.01.04-TS-0901.1100	New Requirement	Added requirements concerning the Flight Object update release checking.
REQ-14.01.04-TS-0901.1110	New Requirement	Added requirements concerning the Flight Object update release checking.
REQ-14.01.04-TS-0901.1120	New Requirement	Added requirements concerning the Flight Object update release checking.
REQ-14.01.04-TS-0901.0505	Requirement Text change	Typo in the requirement text.
REQ-14.01.04-TS-0901.0526	New Requirement	To clarify that the TI layer is expected to enforce the maximum sample size check (see also REQ-14.01.04-TS-0901.0525).
REQ-14.01.04-TS-0901.1130	New Requirement	To detail how the Flight Object deletion is implemented using the OMG DDS.
REQ-14.01.04-TS-0901.0545	Requirement Text change	Improved the requirement text.
REQ-14.01.04-TS-0901.0315	Requirement Text change	Updated the text to refer to SWIM TI - APP IDL. Improved the rationale.
REQ-14.01.04-TS-0901.1140	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1145	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1150	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1155	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1160	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1165	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1170	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1175	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1180	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1185	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer

		description improvement.
REQ-14.01.04-TS-0901.1190	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1195	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1200	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1205	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1210	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1215	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1220	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1225	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1230	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.0316	Requirement Text change	Updated the text to refer to SWIM TI - APP IDL. Improved the rationale.
REQ-14.01.04-TS-0901.1235	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1240	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1245	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1250	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1255	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1260	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1265	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1270	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1275	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.

REQ-14.01.04-TS-0901.1280	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1285	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1290	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1295	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1300	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1305	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1310	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1315	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1320	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1325	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1330	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1335	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1340	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1345	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1350	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1355	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1360	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1365	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.1370	New Requirement	New requirement identified in the context of the SWIM TI - APP interfacing layer description improvement.
REQ-14.01.04-TS-0901.0570	Requirement Text	Updated requirement text and rationale.

	change	

2.3 Proposal for future SWIM-TI Blue Profile improvements

The following points have beed envisaged in Feature #9; but the team did not converge on a solution yet:

1. DEADLINE QoS to IOP_STATUS and RECOVERY_STATUS
2. Performance Categories
3. Further elaborate the CHECKSUM/SUMMARY
4. Training/Testing Shadow Mode
5. (Prototyping) DDS Security
6. (Prototyping) Flight Object Overlay
7. Optimization of Flight Object distribution

The work may be resumed in the context of SESAR 2020 activities.

2.3.1 DEADLINE QoS to IOP_STATUS and RECOVERY_STATUS

ICD for FO_SUMMARY topic uses DEADLINE QoS Policy to contribute to the management of the periodic publication of Flight Object Summaries.

IOP_STATUS and RECOVERY_STATUS also support periodic publications and may also benefit from the use of the DEADLINE QoS Policy.

Furthermore, previous SESAR 1 work (P14.02.09-D11) states the following:

> **DEADLINE QoSPolicy**
>
> Use of Deadline for supervision (IOP_STATUS) is not appropriate here since we want to detect failure of the DataWriter of an instance. What is needed is the LIVELINESS QoS with MANUAL_BY_TOPIC setting and use the *SP-IOP-Max_StatusIdentification_WaitingTime* as *lease_duration*.
>
> When not publishing, the DataWriter will have to call *assert_liveliness()* to manually asserts its liveliness to indicate to the Service that it remains active. This way we are sure another DataWriter cannot republish the instance.

Further work will have to address the use or not of such QoS Policy and agree on any impact on existing ICD.

2.3.2 Performance Categories

IOP Analysis Team confirmed the use of performance categories without clear operational uses cases to support their use.

In order for SWIM-TI implementations to support performance categories for Flight Object distribution, the SWIM TAD section §I.1.6.3. TRANSPORT_PRIORITY recommends using for the WIRE-ICD the TRANSPORT_PRIORITY QoS Policy to correctly support the 3 defined categories (d_1, d_2, d_3) for Flight Object Publications.

This has not been verified in SESAR 1 and should be validated.

2.3.3 Flight Object Summary vs. Checksum

The SWIM-TI distributes the full Flight Objects to all IOP stakeholders in the distribution list and only summaries to other IOP stakeholders in the IOP Area. A Flight Object summary contains the operational key (Flight Key) of the flight and the Flight Object identifier and the Manager/Publisher of the Flight Object. Information in the Flight Object summaries allows any stakeholder to query the SWIM-TI for Flight Objects by Flight Key or a subset of a Flight Key.

Also coming from ED-133, the Flight Object summary is also used for supervision purposes as in the following requirements:

IOP-MFO-154-mdw

MM25,MM64

For supervision purposes, the IOP-MDW shall publish to all IOP stakeholders, periodically (at every SP-IOP-Checksum_Interval time) the summary (FO_SUMMARY) related to each of the flight-objects for which it is the publisher.

IOP-MFO-170-mdw

MM62,MM63,MM12,MM25,MM64

The IOP-MDW shall check that each ATSU has received the release identification (of the managed flight objects) at each SP-IOP-Checksum_Interval time.

Comment: A data distribution service policy can be used to achieve that (DEADLINE see section D.3.2.2.4.4).

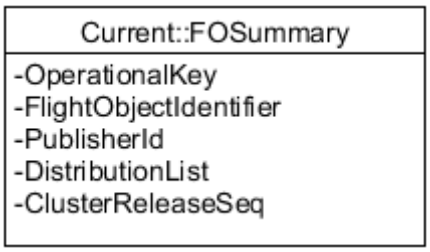
IOP-MFO-190-mdw

MM14,MM25

The IOP-MDW shall request to the FDMP if it does not receive the FO SUMMARY of the flight objects managed by the FDMP before the SP-IOP-Checksum_Interval expires.

Comment: A data distribution service policy can be used to achieve that (DEADLINE see section D.3.2.2.4.4). A system instance FDC/FDU will detect when the FO SUMMARY was not received as expected and will request it to the FDMP system (notice that it does not mean that this request will reach the Application layer of that system).

A Flight Object summary is thus published at least every SP-IOP-Checksum_Interval (30 seconds) for each Flight Object in the IOP Area. Given that the Operational Key of a Flight Object does not change very often, publishing it each 30 seconds is a waste of valuable network resources. This will even get worse once the Network Manager is publishing Flight Object summaries for flight in the planning phase (more Flight Objects with few updates).



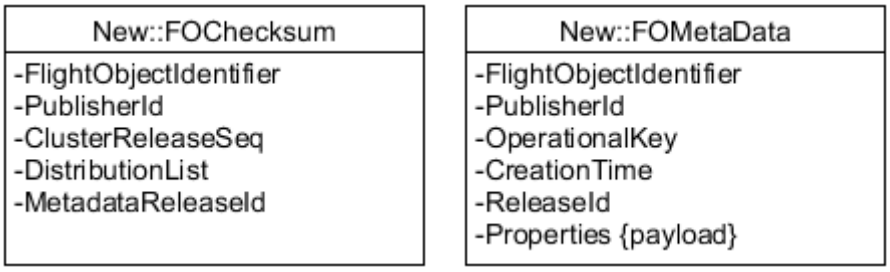
A candidate solution to minimize bandwidth consumption is to separate the checksum aspect {Flight ObjectIdentifier, ClusterReleases, PublisherId} from the ‘summary’ that can be published at a much lower frequency than for the checksum.
This will also allow distribution of additional keys such as IFPLID or GUFI for search purposes.
The following section is a proposal for a possible implementation for a solution.

2.3.3.1 Flight Object Summary

Split current FOSummary structure into 2 structures:

- FOCkecksum
  - Published with higher frequency (at every SP-IOP-Checksum_Interval time, same as current FOSummary).

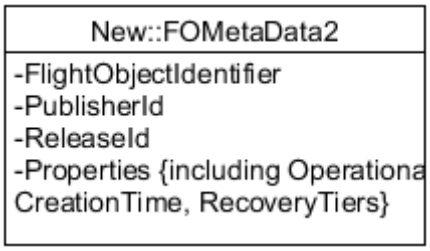
- FOMetaData
  - All FO metadata including OperationalKey
  - Used for SearchFo()
  - Published with lower frequency and on update.
  - Properties as key/value sequence
  - Placeholder for future keys such IFPL_ID, GUFI … etc.



The FOMetaData is not a cluster because it is not specific to DistributionList members. This will require addition of RestoreMetaData() to restore FOMetaData when needed.

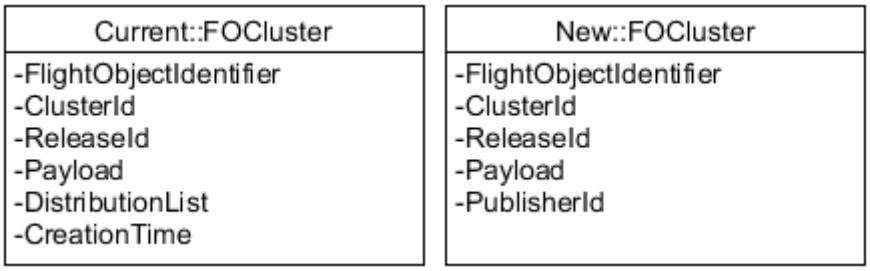
Options

- Include OperationalKey and CreationTime in properties?
- Convey Tiers in FOMetaData instead of DistributionList?
- Allow Application data in FoMetaData?



2.3.3.2 Flight Object Cluster

Move DistributionList and CreationTime to FOMetaData.



2.3.4 Training/Testing Shadow Mode

To support training mode the SWIM-TI will require a capability to support some ‘read-only’ mode where a SWIM Node will receive Flight Objects coming from other SWIM Nodes but with no write capabilities to avoid side-effects on other functioning SWIM Nodes.

SESAR 2020 PJ27 requires shadow mode with existing operational systems. In this context the SWIM Node will require read/write capabilities but constrained to the the nodes participating to the exercise; and share othe capabilities that are in use by the operational systems.

2.3.5 (Prototyping) DDS Security

P14.02.09 in the SESAR 1 carried out several tests to achieve interoperability between PrismTech and RTI DDS solutions in a secure mode, and has contributed to the finalisation of the DDS security specification by raising 35 issues to the OMG for correction in the DDS specification. As DDS security is a key piece in the interoperability scenario for Blue Profile, further work is needed to get proper interoperability in a scenario involving securited exchanges of fligh objects.

2.3.6 (Prototyping) Flight Object Overlay

SESAR 1 SWIM TAD defines an FO-Overlay network for a scalable Flight Object network. Apart from the necessary multicast (SSM), the Flight Object-overlay network has not been prototyped.

To prepare for future deployment and support a hih number of stakeholders, it is important to validate such architecture.

2.3.7 Optimization of Flight Object distribution

In order to optimize the network performance, it is possible to take advantage of the SESAR 1 results provided by P14.02.09-D86 WP1.3.3 Preliminary Solution Evaluations Report – 2016. Further work may be required to incorporate some alternative technological approaches in Blue Profile prototypes.

3 SWIM Blue Profile Functional and non-Functional Requirements

In this chapter functional, non-functional and interface requirements are provided. The chapter is organized in several sub-chapters. The first level of decomposition is between requirements that apply to all the technical functions (§3.1) – or in general to the SWIM Node at a whole - and those that are specific to a given technical function (§3.2, §3.3, etc.). The technical functions are from functional and technical views detailed in the SWIM-TI TAD [13].

The second level of decomposition is between functional, non-functional and interface requirements. In particular, each sub-chapter §3.X is structured as follows:

- Functional requirements (§3.X.1).
- Non-functional requirements, which include the following NFRs:
  - Adaptability (§3.X.2), which contains requirements related to growth and expandability.
  - Performance Characteristics (§3.X.3), which contains requirements concerning capacity, accuracy, timing performances, software resource usage, etc .
  - Safety and Security (§3.X.4), which contains security and privacy requirements, including access limitations, data protection and recovery methods; it also includes safety requirements(according to the safety analysis based on respective standards – when available).
  - Maintainability (§3.X.5), which contains quantitative maintainability requirements.
  - Reliability (§3.X.6) which contains requirements concerning the robustness to abnormal operating conditions.
  - Internal Data Requirements (§3.X.7).
  - Design and Construction Constraints (§3.X.8).
  - Interface requirements (§3.X.9), which contains the specification of the interfaces (including external, internal and network bindings).

If in one or more sub-sections of §3.2, §3.3, etc., no requirements concerning a given category (e.g. Design and Construction Constraints) are provided, all those (if any) included in the concerning §3.1 section (e.g. §3.1.8) are applicable. This approach has been adopted to avoid the duplication of (similar) requirements.

The third level of decomposition concerns the NFRs: all the sections have been organized according to NFR characteristics and sub-characteristics defined in the ISO/IEC 25010:2011. For instance, §3.X.3 (Performance Characteristics) has been traced to ISO/IEC 25010:2011 “Performance efficiency” NFR characteristic. According to that, §3.X.3 has been decomposed by providing a section for each ISO/IEC 25010:2011 “Performance efficiency” sub-characteristics (i.e. time behaviour, resource utilization and capacity requirements). The adoption of ISO/IEC 25010:2011 as reference is coherent and consistent with the SWIM Profiles definition [14].

In the TAD [13], the SWIM-TI Security functional and technical views are described. This specification includes all the identified requirements representing the “what” and the “how” concerning those views. This specification is then complemented by the SWIM-TI Identity Management Technical Specification [15]. In accordance with ISO/IEC 25010:2011, for each §3.X sub-chapters, a specific section concerning the security has been provided. Requirements included in those sections are security requirements applicable to the SWIM Node or/and to the specific technical function (e.g. Messaging). The same applies to the SWIM-TI Security for which security requirements have been identified. For instance, the access to SWIM-TI Security function configurations shall be restricted to authorized users only: this is an example of “security requirement” applicable to the SWIM-TI Security technical functions.

For additional details about SWIM-TI TSs requirements guidelines and the mapping between ISO/IEC 25010:2011 characteristics and TS table of content, refer to [15].

The interface requirements sections section (§3.X.9) has been decomposed according to interface binding kinds described in the TAD [13]. In particular, when applicable, following decomposition is adopted:

- Service Interface bindings, which contains the specifications concerning the "Service Binding". This kind of binding is external to the SWIM-TI and related to an ATM specific service only.
- Internal Service Interface bindings, which contains the specifications concerning the "Internal Service Binding". This kind of binding is internal to the SWIM-TI only and related to any such internal service (e.g. PKI services).
- Network Interface bindings, which contains the specifications concerning the "Network Binding". This kind of binding is external to the SWIM-TI and related to the Network only.
- External Service Interface bindings, which contains the specifications concerning the "External Service Binding". This kind of binding is external to the SWIM-TI and not a <Service binding> or a <Network binding> (e.g. Time Service).

A given binding of type "Service Binding" or "Internal Service Binding" or "External Service Binding" relies on one specific "Network Binding" (traced in the concerning REQ Trace table). Blue Profile "Service Binding" specifications are provided in §3.3.9.1. "Network Binding" specifications applicable to the Blue Profile "Service Binding" and "Internal Service Binding" are provided in D44-005 §3.1.9.1.

In the figure below an overview of technical functions and interfaces concerning this technical specification are provided. More precisely the figure and the text provided, concern both the Blue Profile FDD and core profile parts (D44-005 §2.4.1): SO functions, some advanced suspervision functions and Flight Object interfaces are only part of the FDD profile part.

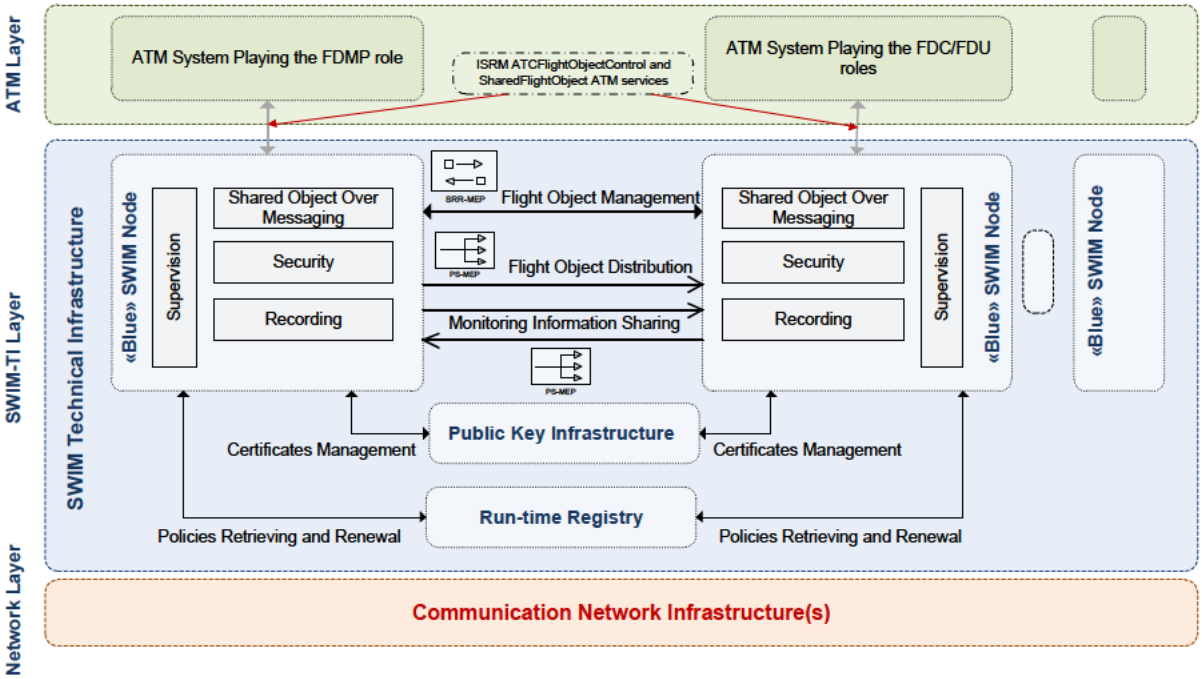


Figure 3-1: Blue Profile Layered Architecture and Interfaces

Currently the SO (and in general the Blue Profile FDD Profile part) is used to enable the consumption and the provisioning of two ATM Specific Services [10]: *ATCFlightObjectControl* and *SharedFlightObject*. The provisioning and consumption architecture specified in the ED-133 and adopted in the BP is based on the Service Virtualization Design Pattern and described in §3.3.9. The two SWIM-TI layer interfaces associated to those ATM Specific services are *FlightObjectManagement*

and *FlightObjectDistribution*. These interfaces are detailed in §3.3.9 and their relationship with SO is described in D44-005 §3.2.

Internal Service Interface bindings identified for the Blue Profile are:

- X.509 certificates management, which includes interfaces at SWIM-TI layer provided by SWIM-TI PKIs to allow Information Security related functions to retrieve, renewal, verify, etc. X.509 certificates used by security controls at transport (e.g. TLS/SSL) or message levels. This interface bindings – consumer role - are provided in D44-005 §3.4.9.1. The source of these bindings is the 14.01.04.D44-002 (SWIM-TI Identity Management Technical Specification).
- Policies Retrieving / Renewal, which is an interface at SWIM-TI layer provided by the SWIM-TI Run-Time Registry(ies) to allow the retrieving/renewal of several kind of policies (messaging policies, security policies, etc.). The source of these bindings is the 14.01.04.D44-003 (SWIM-TI Run-Time Registry Technical Specification).
- Monitoring Information Sharing, which is an interface at SWIM-TI layer aiming at sharing monitoring information between distributed instances of Supervision related functions. This interface is detailed in §3.5.

Furthermore, Communication Network Infrastructure(s) requirements (including interface requirements concerning the interface the Network provides to the SWIM-TI layer) are detailed in D44-005 *§*3.1.

It is anticipated that even if the Blue Profile is currently bound to those ATM Specific Services, the technologies (e.g. OMG DDS) and also the patterns (Service Virtualization) could be adopted for other ATM Specific Service instances.

3.3 Messaging Functional and non-Functional Requirements

3.3.9 Interface Requirements

In this chapter, Messaging Interface requirements identified for the Blue Profile are provided.

Blue Profile includes both generic and ATCFlightObjectControl and SharedFlightObject specific bindings. In particular <BP Core> includes requirements and bindings (refer to §3.3.9.1.1) enabling ATM information exchanges in SRR-MEP and PS-MEP modes based respectively on WS and DDS technologies.

The core part is then complemented but the <BP FDD> requirements and bindings (refer to §3.3.9.2) required to enable the consumption and the provisioning of two ATM Specific Services [10]: *ATCFlightObjectControl* and *SharedFlightObject*. The provisioning and consumption architecture specified in the ED-133 and adopted in the BP is depicted in the two figures above and based on the Service Virtualization Design Pattern.

In the figure below both logical and physical views concerning the *ATCFlightObjectControl* service are provided. The design pattern adopted consists of providing locally to ATM systems virtualized instances of the service (the ones between the SWIM-TI and ATM layers in the figure) demanding to the SWIM-TI layer the routing of the requests and responses to the right participants. At SWIM-TI layer the exchanges concerning this service are managed by the SO functionalities which uses the Messaging *FlightObjectManagement* SWIM-TI layer interface to properly interact with distributed peers.

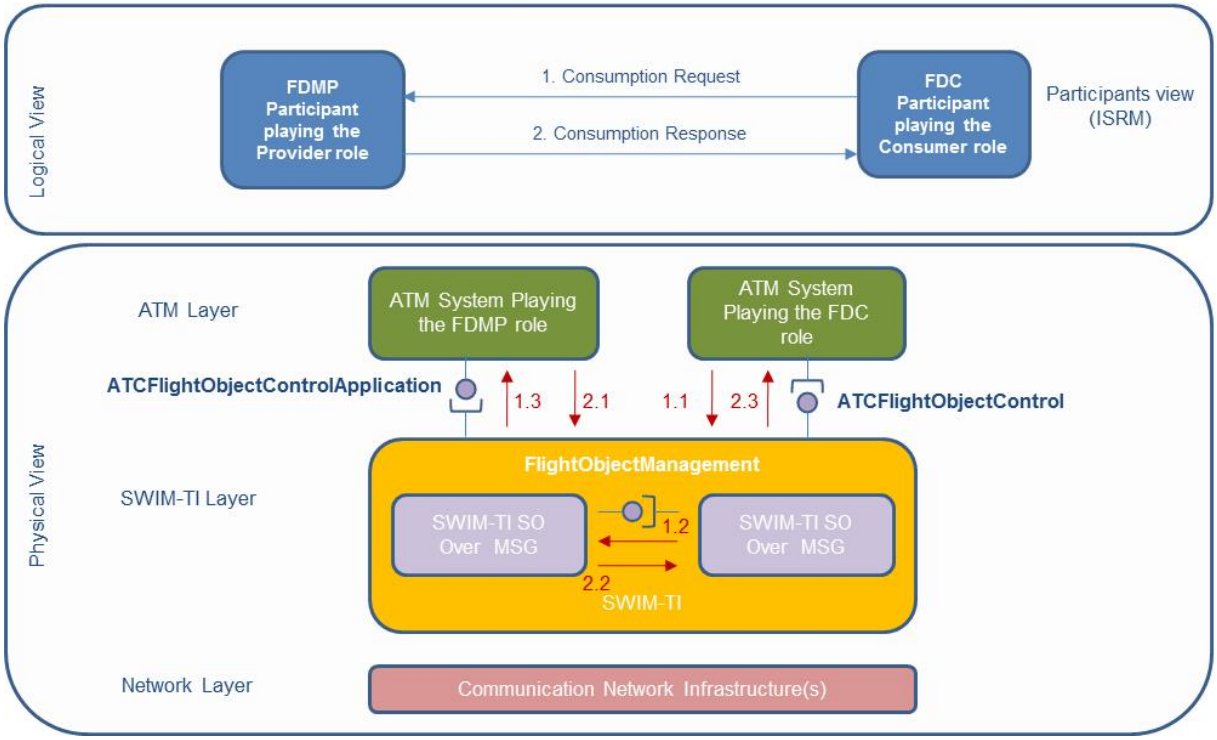


Figure 3-2: ATCFlightObjectControl Request/Response ATM Service Physical Provisioning and Consumption Schema

Summarizing three interfaces have been identified for the ISRM *ATCFlightObjectControl* logical service:

- *ATCFlightObjectControl* Service Technical interface,

- *ATCFlightObjectControlApplication Technical interface, and*
- *FlightObjectManagement* Internal SWIM Technical interface.

The same design pattern has been adopted for the *SharedFlightObject* logical service. In the figure below both logical and physical views concerning the service are provided. The design pattern adopted consists of providing locally to ATM systems virtualized instances of the service (the ones between the SWIM-TI and ATM layers in the figure) demanding to the SWIM-TI layer all the complexity to properly distribute the information. At SWIM-TI layer the exchanges concerning this service are managed by the SO functionalities which uses the Messaging *FlightObjectDistribution* SWIM-TI layer interface to properly interact with distributed peers. In addition to Flight Object information exchanges, the local interface also includes exchanges of IOP status of all he stakeholders of the IOP Area.

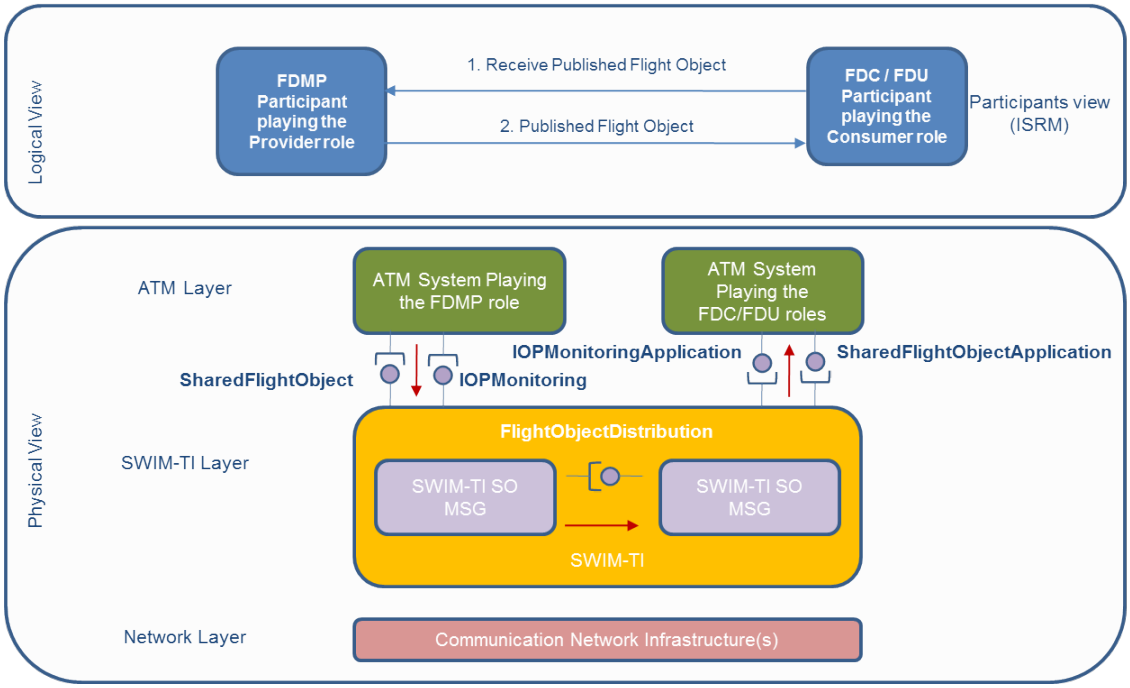


Figure 3-3: SharedFlightObject Publish/Subscribe ATM Service Physical Provisioning and Consumption Schema

Summarizing five interfaces have been identified for the ISRM *SharedFlightObject* logical service:

- *SharedFlightObject* Service Technical interface,
- *SharedFlightObjectApplication* Technical interface,
- *IOPMonitoring* Technical interface,
- *IOPMonitoringApplication* Technical interface, and
- *FlightObjectDistribution* Internal SWIM Technical interface.

The Flight Object Services Request operation enables the mechanism specified in ED-133 [6] used to allow an ATC application to request one or more Flight Object operations to another ATC application managing the target Flight Object. This operation represents a generic channel allowing the routing of the applicative requests concerning the execution of one of more Flight Object applicative operations.

The interaction scenario starts when the ATC application on consumer side identifies the need to require the execution of one or more Flight Object applicative operations according to operational needs (e.g. to add a new flight constraint). This logically consists to require to the ATC application on provider side (assuming that it is the current Flight Data Manager (FDMP) – refer to IOP roles defined in ED-133) to update accordingly the flight data. This logical interaction is implemented via the mediation of the two SWIM Nodes acting respectively as service consumer (SWIM Node consumer) and provider (SWIM Node provider) using the Flight Object Management Interface.

When the SWIM Node on the ATC application consumer side receives the request from the latter, it discovers the SWIM Node on the current FDMP ATC application side according to the provided target flight identifier. The discovered node represents the service provider for that specific “Flight Object Services Request” operation. In case it is not possible to discover the FDMP the request cannot be served.

On the SWIM Node provider side, to serve that request consists mainly in to forward it to the ATC application that can accept or not the request. The request status (accepted or not accepted) is then returned by the SWIM Node provider, as service return value, to the requesting SWIM Node. The latter notifies the “Flight Object Services Request” operation invocation results to the requesting ATC application.

The FlightObjectManagement request point is a port for consuming the FlightObjectManagement service on the SWIM Node FO Contributor side, the port requires the provider interface and provides the consumer interface.

According to the pattern defined in the ED-133, the FDMP accepting the Flight Object services request, shall asynchronously update and publish the Flight Object (Flight Object Distribution interface, §3.3.9.2.2). Because different problems (e.g. competitive constrains) may occur at the application layer during the asynchronous processing of the requested updates, the FlightObjectManagement interface includes the Flight Object Services Request processing failure notification operation used by the FDMP to notify the FDC with any failure concerning previously accepted requests.

The Flight Object Data Rejection operation shall enable the mechanism specified in ED-133 used to allow an ATC application to request the rejection of a given Flight Object which has been previously published by the corresponding FDMP ATC application. The requesting ATC application provides the reason for the rejection that is routed through “Flight Object Data Rejection” operation to the current FDMP ATC application.

The interaction scenario starts when the ATC application on consumer side receives a Flight Object data that, following an appropriate processing, is marked as to be rejected due to a specific reason. This event has to be notified to the current FDMP that published those Flight Object data. This logical interaction is implemented via the mediation of the two SWIM Nodes acting respectively as service consumer (SWIM Node consumer) and provider (SWIM Node provider) using the Flight Object Management interface.

When the SWIM Node on the ATC application consumer side receives the rejection request from the latter, it discovers the SWIM Node on the current FDMP ATC application side according to the provided target flight identifier. The discovered node represents the service provider for that specific “Flight Object Data Rejection” operation. In case it is not possible to discover the FDMP the request cannot be served. On the SWIM Node provider side, to serve that request consists mainly in to forward it to the ATC application and then in to return to the SWIM Node consumer the operation return value. Finally, this value is notified to the requesting ATC application.

The Flight Object Data Restoring operation shall enable the mechanism specified in ED-133 used to allow an ATC application to request the restoring of a given Flight Object data. This logically consists in to require to the current FDMP ATC application to distribute again the current version of the target Flight Object.

This logical interaction is implemented via the mediation of the two SWIM Nodes acting respectively as service consumer (SWIM Node consumer) and provider (SWIM Node provider) using the Flight Object Management Interface. When the SWIM Node on the ATC application consumer side receives

the restoring request from the latter, it discovers the SWIM Node on the current FDMP ATC application side according to the provided target flight identifier.

The discovered node represents the service provider for that specific “Flight Object Data Restoring” operation. In case it is not possible to discover the FDMP the request cannot be served. On the SWIM Node provider side, to serve that request consists mainly in to forward it to the ATC application and then in to return to the SWIM Node consumer the operation return value. Finally, this value is notified to the requesting ATC application.

Flight Objects Recovery Request shall enable the mechanism used to allow to request the recovery (re-publications) of one or more or all Flight Object in a given Tier. The requester of this operation is a recovering SWIM Node whereas the provider (properly discovered - see above) is the SWIM Node of the current FDMP of target Flight Objects.

As the ED-133 lacks a mature recovery process definition, an alternative recovery process has been designed and validated. Both FlightObjectManagement and FlightObjectDistribution internal SWIM Technical interfaces have been updated in order to implement the designed recovery mechanism.

The following assumptions have been considered while specifying the Blue Profile FDD profile part Recovery:

- Recovery process can be triggered by different ways:
  - On demand by IOP Application,
  - Automatically by the SWIM-TI layer if the automatic recovery is enabled/provided (it is an optional capability) and if certain criteria are met (e.g. at start-up or upon reconnection after temporarily isolation from the IOP network). A local “automatic recovery policy” will define the rules to drive the recovery process, with no input (configuration file) or limited input from the IOP Application. The Policy shall be defined by the application layer and SWIM-TI is supposed to enforce this policy (this profile already provides the enforcement for different kinds of policy).
- Recovery process aims at recuperating:
  - The most up-to-date version of the Flight Objects for which the SWIM Node is part of its Distribution List,
  - Summaries of all the Flight Objects in the SWIM Network.
- Efficiency issues are taken into account:
  - Minimize the possibility of a “storm of updates”,
  - Updates to ED-133 Flight Object Data Model and Services will be considered if necessary but kept to a minimum.
- The Recovery process for Blue Profile does not constitute an ATM Information Exchange as it doesn’t support the provision of any new ATM information that was not already available. Instead it should be seen as a technical feature of the SWIM-TI that allows its autonomous recovery in case a sub-set of the SWIM-TI Network falls down.

An overview of the designed recovery mechanism is provided hereafter.

The approach is based on “Recovery Tiers” (i.e. Recovery Tier 1, Recovery Tier 2 up to Recovery Tier n). Each SWIM Node in the Distribution List of a Flight Object is associated with a Tier.

This tiered approach allows to:

- perform the recovery process in sequential steps in order to prevent storm of updates on the recovering SWIM Node side,
- ensure that the most critical Flight Objects are recovered first.

Each Flight Object has an enriched Distribution List in which every stakeholder is assigned with a Tier according to its priority in the recovery process. It is important to note that a Tier is associated to each SWIM Node in the Distribution List for each Flight Object. Hence, a SWIM Node can be associated

with different Tiers for different Flight Objects (since it might be further downstream for some Flight Objects than others).

The number of Tiers can be configured to the optimal value that ensures, the most critical Flight Objects are received soon enough while mitigating a “storm of updates” on the receiving SWIM Node.

An example of assignement logic for the Tiers is provided below for a given Flight Object:

- Tier 0 is associated to the SWIM Node whose ATSU holds responsibility of the Flight.
- Tier 1 is associated to the SWIM Nodes whose ATSU are crossed next downstream.
- Tier 2 is associated to all the other SWIM Nodes in the Distribution List.

It is important to notice that the Tier approach is quite generic concept and does not depend on the particular definition of Tiers. It consists on a sequential recovery process together with a particular criterion to determine the sequence of recovery. The specific definition of Tiers is out of scope of the SWIM-TI, for which a Tier is only a priority of recovery associated to a Flight Object. This specification doesn’t intend to define the business logic to map stakeholders and Flight Objects to Tiers; the definition provided above should serve simply as an example providing guidance.

Two different options were analyzed to implement the tier approach: one pure SRR-MEP and one pure PS-MEP.

The adopted recovery mechanism takes the strengths of PS-MEP Recovery (*FlightObjectDistribution* interface) with the added flow control of SRR-MEP Recovery (*FlightObjectManagement* interface) as a back-up mechanism. This allows for an efficient approach while ensuring that the recovering SWIM Node can rely on a Request/Response mechanism in case any of the expected Flight Objects are not recovered during the process for unexpected reasons.

The Blue Profile FDD profile part Recovery mechanism is based on the following steps:

STEP.0: The recovery process is initiated either:

a. triggered by the IOP Application (application driven mode), or

b. automatically by the SWIM-TI layer if the automatic recovery is enabled/provided (it is an optional capability) and when certain conditions are met (e.g. reconnection after an isolation from the IOP network). In this mode the rules to operate the recovery are described in the ‘automatic recovery policy’.

STEP.1: The recovering SWIM Node receives a Summary periodical publication from all the SWIM Nodes in the network. This allows the recovering SWIM Node to identify which Flight Objects it needs to recover.

STEP.2: The recovering SWIM Node sets its IOP Recovery Status to “TRUE” and the recovering Tier(s) to the level(s) of the Flight Objects to recover first (let’s say “Tier T(s)” in this example) as specified by the IOP Application (application driven mode) or specified in the local automatic recovery policy (automatic mode). This is notified to the other SWIM Nodes through the periodic publication of an “IOP_RECOVERY” topic. Note: the IOP Status is updated by the IOP Application independently.

STEP.3: Every SWIM Node on the Network checks the Tier(s) associated to the recovering SWIM Node for each Flight Object it acts as FDMP. It is thus aware at which point of the recovering process it needs to publish each Flight Object.

STEP.4: SWIM Nodes that act as FDMP for Flight Objects identified as “Tier T(s)” for the recovering SWIM Node in the Distribution List proceed to publish them using the “FO_CLUSTER” Topic and by using the partition QoS in order to ensure that only the respective recovering nodes receive the FO clusters. In order to avoid the unnecessary re-publication of Flight Objects, the recovering node includes a ‘Recovery Context ID’ in the periodically published IOP_RECOVERY topic. The SWIM Node will re-publish the Flight Objects only upon receipt of the first IOP_RECOVERY containing the same context id.

STEP.5: The recovering SWIM Node receives all the Flight Objects for which it appeared as “Tier T(s)” in the Distribution List.

Since the IOP Application (application driven mode) or the SWIM Node (automatic mode) is aware (in STEP 1) of the entire list of Flight Objects and which ones it expects to receive during “Tier T(s)”, it checks the completion of the “Tier T(s)” Recovery process and optionally react if any expected Flight Objects is missing.

a. CONDITIONAL STEP: a Request/Response mechanism is used to recover the missing Flight Object(s). The request sent to the appropriate SWIM Node identifies the missing Flight Object Identifiers. The receiving Node will return first a Response (Boolean indicating the result of the process and a reason in case of failure) and then publish the requested Flight Object(s).

STEP.6: Upon completion of the “Tier T(s)” recovery, the recovering SWIM Node updates its RECOVERY_STATUS topic with the next Tier(s) to recover, as indicated by the IOP Application (application driven mode) or by the local automatic recovery policy (automatic mode). Steps 3 to 5 are re-iterated.

STEP.7: The process continues iteratively until the IOP Application (application driven mode) or the local automatic recovery policy (automatic mode) considers the recovery process completed. This can be either because all missing Flight Objects have been recovered or the still missing Flight Objects are considered not in interest.

STEP.8: Upon receiption of the indication from the IOP Application (application driven mode) or the local automatic recovery policy (automatic mode) that the recovery is completed, the recovering SWIM Node changes its IOP Recovery Status to “FALSE”. Note: the IOP status is updated by the IOP application independently.

All these interfaces (generic and Flight Object specific) are detailed hereafter. For what concerns the Flight Object specific interfaces, according to the interoperability needs assessment, interfaces concerning the service interface layer have been specified only at platform independent model abstraction level (not technology specific instantiation is provided). More precisely:

Service Interface layer (no technology specific instantiation, §3.3.9.1.2 to §3.3.9.1.7):

- ATCFlightObjectControl Service Technical interface.
- ATCFlightObjectControlApplication Technical interface.
- SharedFlightObject Service Technical interface.
- SharedFlightObjectApplication Technical interface.
- IOPMonitoring Technical interface.
- IOPMonitoringApplication Technical interface.

The above interfaces are described in §3.3.9.1. An illustrative IDL is provided to better support the analysis and the development of these interfaces:

Internal Service Interface layer (including technology specific instantiations, §3.3.9.2):

- FlightObjectManagement Internal SWIM Technical interface.
- FlightObjectDistribution Internal SWIM Technical interface.

The following requirement applies to all external service interfaces.

[IREQ]

Identifier	REQ-14.01.04-TS-0901.0840
Requirement	The interface binding contract shall reference the authoritative procedure that describes the versioning mechanisms, which are applicable to the contract and any of its constituents.
Title	The contract itself shall be versioned.
Status	<In Progress>
Rationale	A study on versioning of the service interface has revealed that there is not one size that fits all. The effective organisation of versioning is decided at service instantiation. This requirement ensures that whatever option is taken, that the option is known to all impacted Stakeholders. In the SWIM Profiles Technical Specifications “Interface Evolution Analysis” several rules and recommendations that ATM Service architects may adopt and/or complement are provided. Interface evolution analysis focus on evolution of only STDD (Service Technical Design Description) “Service Technical Interfaces” part because its relationship with SWIM-TI interface bindings specifications. Some of the rules/recommendations are SWIM-TI Profiles Interface Bindings independent whereas other are binding specific due to particular standards adopted in that binding. For instance rules on XSD modelling techniques to achieve minor version compatibility are only applicable to interface bindings using XML/XSD. Furthermore, ATM service implementations versioning is not addressed. In particular for a given version of the STDD, a stakeholder may plan different versions of the service implementation. According to the “Contract first” (STDD) approach, changes on service implementations are not expected to impact technical interoperability (the STDD version is the same) if what specified in the STDD is properly used as reference by both provider and consumer.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<YP Core><YP Security+><YP Messaging+><BP Core><BP FDD><PP Core><PP Messaging Bridging>
Domain of interest	<ICD><Governance><SLA>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer><Subscriber><Publisher><Publication consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<Yes>
Testability	<Conformance testable><Interoperability testable>

[IREQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	Yellow Profile	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<ALLOCATED_TO>	<Functional block>	Purple Profile	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0302	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0303	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0790	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0795	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0304	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0305	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0306	N/A

<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0760	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0308	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0765	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0307	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0770	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0309	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0775	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0317	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0780	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0318	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0785	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0751	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0752	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0705	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0710	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0715	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0720	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0325	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0800	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0801	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0802	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0803	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0804	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0805	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0806	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0807	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	A/C-57	<Full>
<SATISFIES>	<Enabler>	AGSWIM-34	<Full>
<SATISFIES>	<Enabler>	AGSWIM-41	<Full>
<SATISFIES>	<Enabler>	AGSWIM-43	<Full>
<SATISFIES>	<Enabler>	AGSWIM-44	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-06b	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1 Service Interface Bindings

This paragraph provides all the needed details concerning the identified Service Technical interfaces as introduced before. In particular interface bindings for both generic and <BP FDD> Profile Part specific are specified.

In D44-005 Appendix C Interface Evolution analysis, applicable to ATM services using interface bindings part of this Technical Specification, is provided.

3.3.9.1.1 Generic Interface Bindings

No changes with respect to D44-005.

3.3.9.1.2 *ATCFlightObjectControl* Interface Requirements

ATCFlightObjectControl endpoints are provided on top of SWIM-TI SO and MSG and in particular on top of the Internal Interface FlightObjectManagement which provides several operations used to serve properly ATCFlightObjectControl consumption/provisioning.

According to the design pattern adopted, this interface has been identified and described only at logical level because there are no interoperability needs. The *FlightObjectManagement* interface is impacting the interoperability whereas the way locally the SWIM-TI layer interacts with the ATM layer is kept flexible and open.

[REQ]

Identifier	REQ-14.01.04-TS-0901.0315
Requirement	ATCFlightObjectControl Interface shall be instantiated using the following binding: + Protocol stack: not standardised + MEPs: not standardised + Fault handling: not standardised + Encoding. not standardised + Security: - Confidentiality: not standardised - Integrity: not standardised - Authenticity: not standardised - Authorization: not standardised - Non-repudiation: not standardised + Contract: - formalism of contract description: UML, IDL, Text - minimum: not applicable - reference: Blue Profile Technical Specification, ISRM + Interoperability: not standardised
Title	ATCFlightObjectControl Interface binding
Status	<Validated>
Rationale	This binding is not subjected to standardisation and is implementation specific. ATCFlightObjectControl endpoints are provided on top of SWIM-TI SO and MSG and in particular on top of the Internal Interface FlightObjectManagement which provides several operations used to serve properly ATCFlightObjectControl consumption/provisioning. According to the design pattern adopted, this interface has been identified and described only at logical level because there are no interoperability needs. In the introduction of §3.3.9 an illustrative IDL is provided to support the analysis and the development of this interface. The FlightObjectManagement interface is impacting the interoperability whereas the way locally the SWIM-TI layer interacts with the ATM layer is kept flexible and open.
Category	<Interface>
Validation Method	
Verification Method	<Analysis>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Service binding>

Conformance	<No>
High Level	<Yes>
Testability	<Applicable but not testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

In the figure below, ATCFlightObjectControl service contract as specified in ISRM 2.0 is provided.

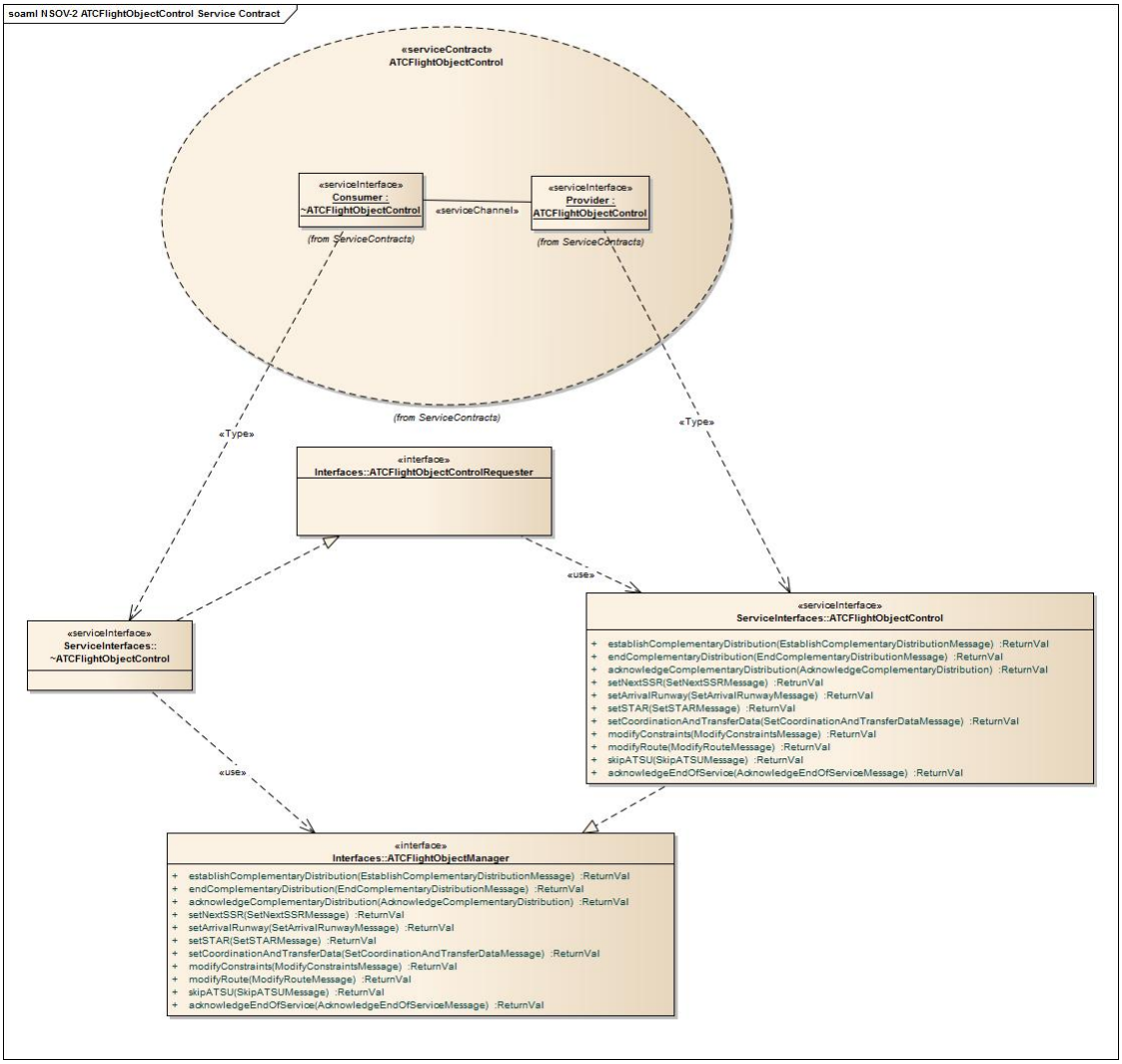


Figure 3-4: ISRM ATCFlightObjectControl service contract

All the operations provided by the service (e.g. *establishComplementaryDistribution*, *setNextSSR*, etc.) aim at submitting control messages that have been properly defined as message type (e.g. *establishComplementaryDistributionMessage*, *setNextSSRMessage*, etc.). At SWIM-TI layer each one of these operations is mapped to the *FlightObjectManagement* interface *RequestFOService* operation enveloping the corresponding control message type. FlightObjectManagement interface and concerning data types are specified in §3.3.9.2.1.

3.3.9.1.2.1 Operations

3.3.9.1.2.1.1 RequestFlightObjectServices

```
ComplexReport RequestFlightObjectServices (
          in IopStakeholderId requester_id,
          in RequestId request_id,
          in FlightObjectIdentifier fo_id,
          in ClusterReleaseIdSequence fo_release,
          in Requests fo_request,
          in Qos priority);
```

3.3.9.1.2.1.1.1 Parameters

Input:

- **requester_id : IopStakeholderId.** This parameter is the identifier of the IOP stakeholder requesting the service**.**
- **request_id** : **RequestId**. Request identification.
- **fo_id**: **FlightObjectIdentifier.** This parameter represents the unique identifier of Flight Object.
- **fo_release** : **ClusterReleaseIdSequence.** This parameter is the list of the cluster identifiers and related release numbers for the whole FO.
- **fo_request** : **Requests.** This parameter represents a set of application services to be requested to an application in another system instance.
- **priority: Qos.** This parameter represents the priority that is assigned to the request delivery. It is one of the three possible classes of services offered by the SWIM-TI (`d_1`, `d_2`, `d_3`).

Output:

ComplexReport: Return value for the acceptance or the rejection of the request.

In addition to ComplexReports that may be returned by the IOP application, the following ComplexReport may be generated locally by the SWIM-TI as a result of validity checking by the SWIM-TI.

return_code	return_value	Comments
TRUE		Field *return_value* is not applicable.
FALSE	fo_not_found	The Flight Object is not known (does not exist).
FALSE	syntax_error	Malformed parameter (requester_id).
FALSE	semantic_error	Clusters names are different from the locally stored clusters names. Cluster not found. Cluster release id is out of range.
FALSE	not_eligible	The requester id is the manager (FDMP) of the Flight Object.

FALSE	bad_fo_version	The provided Flight Object release is different from the locally stored release.
FALSE	timeout	No response from stakeholder (FDMP) within the predefined time duration
FALSE	middleware_failure	Internal middleware error (at FDC or at FDMP). Unknown Stakeholder (FDMP).
FALSE	isolated_stakeholder	Stakeholder (FDMP) is not IOP Enabled. Local middleware is not IOP Enabled.
FALSE	communication_failure	Communication with FDMP could not be established or is lost while a request is in progress, after the request was sent by FDC but before the reply from FDMP has been returned to application.

3.3.9.1.2.1.1.2 Validity Checking

- The fo_id must exist in the IOP area. In case it doesn't exist the report shall return "NOK", with the reason fo_not_found.
- The requester must not be the current manager of the FO.
- Let Rcurrent be the locally stored FOReleaseId, Rnew be the new FOReleaseId as in fo_release, where R[i] is the release version of cluster i in the FOReleaseId R.

  IF (for each cluster i Rnew[i] = Rcurrent[i]) THEN OK

  ELSE NOK with reason bad_fo_version.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1140
Requirement	While serving an ATCFlightObjectControl RequestFlightObjectServices operation call, the SWIM-TI layer shall enforce the following validity checks: - Target Flight Object Identifier must exist in the IOP area. In case this check is not completed successfully, the "fo_not_found" error code shall be returned. - The requester must not be the current manager of the Flight Object. In case this check is not completed successfully, the "not_eligible" error code shall be returned. - The provided Flight Object release must be equal to the release stored locally. In case this check is not completed successfully, the "bad_fo_version" error code shall be returned. - The provided Flight Object Service requests must be not null and not empty. In case this check is not completed successfully, the "syntax_error" error code shall be returned. - The provided Flight Object Service request identifier must be not null and not empty. In case this check is not completed successfully, the "syntax_error" error code shall be returned. - The provided requester stakeholder identifier must be not null, not empty and well formed syntax. In case this check is not completed successfully, the "syntax_error" error code shall be returned. - The provided Flight Object Cluster identifiers must be valid. In case this check is not completed successfully, the "semantic_error" error code shall be returned. - The provided Flight Object Cluster releases must be valid (within the admissible range). In case this check is not completed successfully, the "semantic error" error code shall be returned.
Title	ATCFlightObjectControl RequestFlightObjectServices operation validity

	checks
Status	<In Progress>
Rationale	In addition to ComplexReport that may be returned by the FDMP, the SWIM-TI, as a result of validity checking, may generate locally additional error codes (ComplexReport with report code FALSE).
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1145
Requirement	While serving an ATCFlightObjectControl RequestFlightObjectServices operation call, the SWIM-TI layer shall return “middleware_failure” error code when the FDMP’s stakeholder is not known, or when errors occur at the SWIM-TI layer.
Title	ATCFlightObjectControl RequestFlightObjectServices operation “middleware failure” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the FDMP layer and by the SWIM-TI layer (validity checks), the requester is also informed when internal errors occur at the SWIM-TI layer.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1150
Requirement	While serving an ATCFlightObjectControl RequestFlightObjectServices operation call, the SWIM-TI layer shall return “isolated_stakeholder” error code to application when neither the local SWIM-TI layer nor the FDMP stakeholder is IOP Enabled.
Title	ATCFlightObjectControl RequestFlightObjectServices operation “isolated stakeholder” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the FDMP’s application layer and by the SWIM-TI layer (validity checks), the requester is also informed when neither the FDMP not the local SWIM-TI layer is IOP Enabled.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.2.1.1.3 Semantic

Once validity checking is OK, the SWIM-TI will invoke FlightObjectManagement RequestFlightObjectServices up to IOP-Max_Contrib_Requests times. If no response is available within *SP-IOP-Max_Contrib_Waiting_Time * IOP-Max_Contrib_Requests*, a corresponding ComplexReport shall be reported to the application with negative acknowledgement, and the content of the response to each of the request is undefined.

The SWIM-TI shall reuse the same parameters (with no change of the request identifier) for each of the retries.

The calling system is blocked until the reception of the return value (a ComplexReport), which contains the set of acknowledgements returned from the receiving application.

The request_id must be unique in the context of the application. The application is responsible for updating value of request_id between successive calls to the service.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1155
Requirement	Once successfully completed the validity checks applicable to the ATCFlightObjectControl RequestFlightObjectServices operation, the SWIM-TI layer shall invoke FlightObjectManagement RequestFlightObjectServices on the FDMP and: - Return back to the requester with the invocation return, or - In case the communication with FDMP cannot be established or is lost after request is sent to FDMP and before reply from FDMP is received by the SWIM-TI layer, the “communication_failure” error shall be returned. - If no response is availablefrom FDMP within S*P-IOP-Max_Contrib_Waiting_Time_** time duration*,* the SWIM-TI shall reinvoke again FlightObjectManagement RequestFlightObjectServices on the FDMP using the same input parameters. - If no response is available after SP-IOP-Max_Contrib_Requests_* reivocations, the “timeout” error code shall be returned.
Title	ATCFlightObjectControl RequestFlightObjectServices invocation of FlightObjectManagement RequestFlightObjectServices operation.
Status	<In Progress>
Rationale	Once validity checking is OK, the SWIM-TI will invoke FlightObjectManagement RequestFlightObjectServices up to IOP-Max_Contrib_Requests times. If no response is available within SP-IOP-Max_Contrib_Waiting_Time * IOP-Max_Contrib_Requests, a corresponding ComplexReport shall be reported to the application with negative acknowledgement, and the content of the response to each of the request is undefined. To allow the FDMP to detect the retries, the SWIM-TI will the same request identifier for each of the retries. The calling system is blocked until the reception of the return value (a ComplexReport), which contains the set of acknowledgements returned from the receiving application. The request_id must be unique in the context of the application. The application is responsible for updating value of request_id between successive calls to the service. SP-IOP-Max_Contrib_Requests_* value depends on the value of input priority.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>

Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.2.1.2 RestoreFlightObject

```
Report RestoreFlightObject(in IopStakeholderId requester id,
                           in FlightObjectIdentifier flight iop id);
```

3.3.9.1.2.1.2.1 Parameters

Input:

- **requester_id : IopStakeholderId.** This parameter is the identifier of the IOP stakeholder requesting restoring of the Flight Object.
- **flight_iop_id: FlightObjectIdentifier.** This parameter represents the Flight Object identifier of the flight to restore.

Output:

Report: Return value for the acceptance or the rejection of the request.

return_code	**return_value**	**Comments**
TRUE		Field *return_value* is not applicable.
FALSE	fo_not_found	The Flight Object is not known (does not exist).
FALSE	syntax_error	Malformed parameter (requester_id).
FALSE	middleware_failure	Internal middleware error. Unknown Stakeholder (FDMP).
FALSE	not_eligible	The requester id is the manager (FDMP) of the Flight Object.
FALSE	isolated_stakeholder	Stakeholder (FDMP) is not IOP Enabled. Local middleware is not IOP Enabled.
FALSE	communication_failure	Communication with FDMP could not be established or is lost while a request is in progress, after the request was sent by FDC but before the reply from FDMP has been returned to application.
FALSE	timeout	No response from stakeholder (FDMP) within the predefined time duration

3.3.9.1.2.1.2.2 Validity Checking

The flight_iop_id must match one of the identifiers stored in the Flight Object summaries in the local SWIM-TI instance. Otherwise a fo_not_found exception shall be reported to the application.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1160
Requirement	While serving an ATCFlightObjectControl RestoreFlightObject operation call, the SWIM-TI layer shall enforce the following validity checks: - Target Flight Object Identifier must exist in the IOP area. In case this check is not completed successfully, the “fo_not_found” error code shall be returned. - The requester must not be the current manager of the Flight Object. In case this check is not completed successfully, the “not_eligible” error code shall be returned. - The provided requester stakeholder identifier must be not null, not empty and well formed syntax. In case this check is not completed successfully, the “syntax_error” error code shall be returned.
Title	ATCFlightObjectControl RestoreFlightObject operation validity checks
Status	<In Progress>
Rationale	In addition to Report that may be returned by the FDMP, the SWIM-TI, as a result of validity checking, may generate locally additional error codes (Report with report code FALSE).
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1165
Requirement	While serving an ATCFlightObjectControl RestoreFlightObject operation call, the SWIM-TI layer shall return “middleware_failure” error code when the FDMP’s stakeholder is not known, or when errors occur at the SWIM-TI layer.

Title	ATCFlightObjectControl RestoreFlightObject operation “middleware_failure” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the FDMP layer and by the SWIM-TI layer (validity checks), the requester is also informed when internal errors occur at the SWIM-TI layer.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1170
Requirement	While serving an ATCFlightObjectControl RestoreFlightObject operation call, the SWIM-TI layer shall return “isolated_stakeholder” error code to application when neither the local SWIM-TI layer nor the FDMP stakeholder is IOP Enabled.
Title	ATCFlightObjectControl RestoreFlightObject operation “isolated_stakeholder” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the FDMP’s application layer and by the SWIM-TI layer (validity checks), the requester is also informed when neither the FDMP not the local SWIM-TI layer is IOP Enabled.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.2.1.2.3 Semantic

This operation allows an application to restore the latest version of a specific Flight Object. This triggers the invocation of the FlightObjectManagement RestoreFlightObject operation to the FDMP of the Flight Object to request its republication.

This operation is also used by a stakeholder that is not in the distribution list to request content of a Flight Object.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1175
Requirement	Once successfully completed the validity checks applicable to the ATCFlightObjectControl RestoreFlightObject operation, the SWIM-TI layer shall invoke FlightObjectManagement RestoreFlightObject on the FDMP and: - Return back to the requester with the invocation return, or - In case the communication with FDMP cannot be established or is lost after request is sent to FDMP and before reply from FDMP is received by the SWIM-TI layer, the “communication_failure” error shall be returned. - If no response is availablefrom FDMP within predefined time duration, the “timeout” error code shall be returned.
Title	ATCFlightObjectControl RestoreFlightObject invocation of FlightObjectManagement RestoreFlightObject operation.
Status	<In Progress>
Rationale	This operation allows an application to restore the latest version of a specific Flight Object. This triggers the invocation of the FlightObjectManagement RestoreFlightObject operation to the FDMP of the Flight Object to request its republication. This operation is also used by a stakeholder that is not in the distribution list to request content of a Flight Object.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>

Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.2.1.3 RejectFlightObject

```
Report RejectFlightObject(
      in IopStakeholderId requester id,
      in FlightObjectIdentifier fo id,
      in ClusterReleaseIdSequence fo release,
      in string reject_reason);
```

3.3.9.1.2.1.3.1 Parameters

Input:

- **requester_id: IopStakeholderId.** This parameter is the identifier of the IOP stakeholder rejecting the Flight Object.
- **fo_id: FlightObjectIdentifier.** This parameter is the identifier of the Flight Object to be rejected.
- **fo_release: ClusterReleaseIdSequence.** This parameter is the FO release of the Flight Object to be rejected. It contains all cluster identifiers and release numbers of the Flight Object.
- **reject_reason: string.** Reason for rejection.

The SWIM-TI will not interpret the reason of the rejection. It is assumed that IOP Application ICD will define all possible values for this parameter (possibly XML).

Output:

Report: Return value for the acceptance or the rejection of the request.

In addition to Report that may be returned by the IOP apploication, the following Report may be generated locally by the SWIM-TI as a result of validity checking by the SWIM-TI.

return_code	**return_value**	**Comments**
TRUE		Field *return_value* is not applicable.
FALSE	fo_not_found	The Flight Object is not known (does not exist).
FALSE	syntax_error	Malformed parameter (requester_id).
FALSE	middleware_failure	Internal middleware error. Unknown Stakeholder (FDMP).
FALSE	not_eligible	The requester id is the manager (FDMP) of the Flight Object.
FALSE	semantic_error	Empty rejection reason.

		Wrong/Unknown requester_id. Cluster release id is out of range. Invalid Cluster identifiers.
FALSE	isolated_stakeholder	Stakeholder (FDMP) is not IOP Enabled. Local middleware is not IOP Enabled.
FALSE	communication_failure	Communication with FDMP could not be established or is lost while a request is in progress, after the request was sent by FDC but before the reply from FDMP has been returned to application.
FALSE	timeout	No response from stakeholder (FDMP) within the predefined time duration

3.3.9.1.2.1.3.2 Validity Checking

The fo_id must be known locally from the stored Flight Object summaries. Otherwise a Report value FALSE with return_value fo_not_found shall be returned.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1180
Requirement	While serving an ATCFlightObjectControl RejectFlightObject operation call, the SWIM-TI layer shall enforce the following validity checks: - Target Flight Object Identifier must exist in the IOP area. In case this check is not completed successfully, the “fo_not_found” error code shall be returned. - The requester must not be the current manager of the Flight Object. In case this check is not completed successfully, the “not_eligible” error code shall be returned. - The provided requester stakeholder identifier must be not null, not empty and well formed syntax. In case this check is not completed successfully, the “syntax_error” error code shall be returned. - The provided rejection reason must be not null and not empty. In case this check is not completed successfully, the “syntax_error” error code shall be returned. - The provided requester stakeholder identifier must be known by the SWIM-TI layer. In case this check is not completed successfully, the “semantic_error” error code shall be returned. - The provided Flight Object Cluster identifiers must be valid. In case this check is not completed successfully, the “semantic_error” error code shall be returned. - The provided Flight Object Cluster releases must be valid (within the admissible range). In case this check is not completed successfully, the “semantic_error” error code shall be returned.
Title	ATCFlightObjectControl RejectFlightObject operation validity checks
Status	<In Progress>
Rationale	In addition to Report that may be returned by the FDMP, the SWIM-TI, as a result of validity checking, may generate locally additional error codes (Report with report code FALSE).
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>

Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1185
Requirement	While serving an ATCFlightObjectControl RejectFlightObject operation call, the SWIM-TI layer shall return “middleware_failure” error code when the FDMP’s stakeholder is not known, or when errors occur at the SWIM-TI layer.
Title	ATCFlightObjectControl RejectFlightObject operation “middleware_failure” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by FDMP and by the SWIM-TI layer (validity checks), the requester is also informed when internal errors occur at the SWIM-TI layer.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1190
Requirement	While serving an ATCFlightObjectControl RejectFlightObject operation call, the SWIM-TI layer shall return “isolated_stakeholder” error code to application when neither the local SWIM-TI layer nor the FDMP stakeholder is IOP Enabled.
Title	ATCFlightObjectControl RejectFlightObject operation “isolated_stakeholder” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the FDMP’s application layer and by the SWIM-TI layer (validity checks), the requester is also informed when neither the FDMP not the local SWIM-TI layer is IOP Enabled.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.2.1.3.3 Semantic

This operation requests the rejection of a given Flight Object, reporting the reason for rejection to the current FDMP (from the locally stored Flight Object summaries). This triggers a calling of the FlightObjectManagement RejectFlightObject operation to the FDMP of the Flight Object to be rejected.

The SWIM-TI will not accept a request from an application that is already FDMP.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1195
Requirement	Once successfully completed the validity checks applicable to the ATCFlightObjectControl RejectFlightObject operation, the SWIM-TI layer shall invoke FlightObjectManagement RejectFlightObject on the FDMP and: - Return back to the requester with the invocation return, or

	- In case the communication with FDMP cannot be established or is lost after request is sent to FDMP and before reply from FDMP is received by the SWIM-TI layer, the “communication_failure” error shall be returned. - If no response is availablefrom FDMP within predefined time duration, the “timeout” error code shall be returned.
Title	ATCFlightObjectControl RejectFlightObject invocation of FlightObjectManagement RejectFlightObject operation.
Status	<In Progress>
Rationale	This operation requests the rejection of a given Flight Object, reporting the reason for rejection to the current FDMP (from the locally stored Flight Object summaries). This triggers a calling of the FlightObjectManagement RejectFlightObject operation to the FDMP of the Flight Object to be rejected. The SWIM-TI will not accept a request from an application that is already FDMP.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.2.1.4 RequestFlightObjectsRecovery

```
struct FoRecoveryException {
    FlightObjectIdentifier flight_object_id;
    ExceptionKind error;
  };

  typedef sequence<FoRecoveryException> FoRecoveryExceptions;

  struct FoRecoveryReport {
    Report report value;
    FoRecoveryExceptions return value;
  };

FoRecoveryReportRequestFlightObjectsRecovery(
```

```
        in IopStakeholderId requester_id,
        in Tier recovery tier,
        in FlightObjectIdentifiers missing_fos);
```

3.3.9.1.2.1.4.1 Parameters

Input:

- **requester_id: IopStakeholderId.** This parameter is the identifier of the IOP stakeholder requesting the Flight Object recovery (the recovering SWIM Node).
- **recovery_tier: Tier.** This parameter consists of an Integer providing the Recovery Tier of the Recovering SWIM Node identifier of the Flight Object to be rejected.
- **missing_fos: FlightObjectIdentifiers.** This parameter consists of the Flight Object identifiers of the Flight Objects to be recovered.

Output:

FoRecoveryReport: Return value for the acceptance or the rejection of the request together with any failure to publish a missing Flight Object.

return_code	return_value	Comments
TRUE		Check FoRecoveryExceptions for per Flight Object report.
FALSE	syntax_error	Malformed parameter (requester_id).
FALSE	semantic_error	Tier value is not within a valid range.
FALSE	middleware_failure	Internal middleware error.
FALSE	isolated_stakeholder	Local middleware is not IOP Enabled.

For ***each missing Flight Object*** the following recovery exception may be returned:

return_value	Comments
fo_not_found	The Flight Object is not known (does not exist).
not_eligible	The requester is already the manager of the Flight Object (cannot call oneself). The Flight Object does not belong to the indicated tier. The recovering SWIM Node is not in the distribution list of the Flight Object.
isolated_stakeholder	Stakeholder (FDMP) is not IOP Enabled.
communication_failure	Communication with FDMP could not be established or is lost while a request is in progress, after the request was sent by FDC but before the reply from FDMP has been returned to application.
timeout	No response from stakeholder (FDMP) within the predefined time duration

3.3.9.1.2.1.4.2 Validity Checking

When missing_fos is empty, the entire list of Flight Objects for which the requester is in specified recovery tier. Otherwise, all missing Flight Objects shall be known to the SWIM-TI and correspond to the indicated recovery tier.

The requester shall not be a manager of any Flight Object from the missing_fos.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1200

Requirement	While serving an ATCFlightObjectControl RequestFlightObjectsRecovery operation call, the SWIM-TI layer shall enforce the following validity checks: - The provided requester stakeholder identifier must be not null, not empty and well formed syntax. In case this check is not completed successfully, the “syntax error” error code shall be returned. - The provided Tiers must be valid (within the admissible range). In case this check is not successfully completed, the “semantic error” error code shall be returned. - The local SWIM-TI layer must be IOP Enabeld. In case this check is not successfully completed, the “isolated_stakeholder” error code shall be returned. -When provided, each of the missing Flight Objects to recover, the SWIM-TI shall enforce the following checks: - Target Flight Object Identifier must exist in the IOP area. In case this check is not completed successfully, the Flight Object Indentifier and “fo_not_found” error code shall be added to the recovery exception list. - The requester must not be the current manager of the Flight Object. In case this check is not completed successfully, the Flight Object Indentifier and “not_eligible” error code shall be added to the recovery exception list. - The Flight Object must belong to the provided tier. In case this check is not completed successfully, the Flight Object Indentifier and “not_eligible” error code shall be added to the recovery exception list. - The requester must be in the distribution list of the Flight Object. In case this check is not completed successfully, the Flight Object Indentifier and “not_eligible” error code shall be added to the recovery exception list. - The manager of the Flight Object must be IOP Enabeld. In case this check is not completed successfully, the Flight Object Indentifier and “isolated_stakeholder” error code shall be added to the recovery exception list.
Title	ATCFlightObjectControl RequestFlightObjectsRecovery operation validity checks
Status	<In Progress>
Rationale	When missing_fos is empty, the entire list of Flight Objects for which the requester is in specified recovery tier. Otherwise, all missing Flight Objects shall be known to the SWIM-TI and correspond to the indicated recovery tier. The requester shall not be a manager of any Flight Object from the missing_fos
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A

<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1205
Requirement	While serving an ATCFlightObjectControl RequestFlightObjectsRecovery operation call, the SWIM-TI layer shall return “middleware_failure” error code when errors occur at the SWIM-TI layer.
Title	ATCFlightObjectControl RequestFlightObjectsRecovery operation “middleware_failure” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the SWIM-TI layer (validity checks), the requester is also informed when internal errors occur at the SWIM-TI layer.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.2.1.4.3 Semantic

This operation allows an application to request recovery of a subset or all of the Flight Objects for which it is in a specific recovery tier.

Once validity checking is OK, the SWIM-TI will initiate calls to FlightObjectManagement interface RequestFlightObjectsRecovery operation to all the managers of the Flight Objects to recover.

This implies that one ATCFlightObjectControl.RequestFlightObjectsRecovery request might trigger several FlightObjectManagement. RequestFlightObjectsRecovery requests directed to different stakeholders. A return_code of TRUE is set if all the FlightObjectManagement. RequestFlightObjectsRecovery requests are accepted by the respective FDMPs.

An ATCFlightObjectControl.RequestFlightObjectsRecovery call returns immediately when all FlightObjectManagement. RequestFlightObjectsRecovery requests finish. A FlightObjectManagement. RequestFlightObjectsRecovery request finishes either when a response is received or when the request fails due to conditions corresponding to communication_failure or timeout. The actual SharedFlightObject publications of relevant FO data might take place after the ATCFlightObjectControl.RequestFlightObjectsRecovery returned.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1210
Requirement	Once successfully completed the validity checks applicable to the ATCFlightObjectControl RequestFlightObjectsRecovery operation, the SWIM-TI layer shall invoke FlightObjectManagement RequestFlightObjectsRecovery operations on all the managers of the Flight Objects to recover (that have passed validity check) and: - Return back to the requester a union of all the recovery exceptions received from the managers. - In case communication with a stakeholder cannot be established or is lost after request is sent to the stakeholder and before reply from the stakeholder is received by the SWIM-TI layer, each Flight Object Indentifier of the Flight Objects to recover and for which the stakeholder is the manager and “communication_failure” error code will be added to the recovery exception list. - In case no response is available from astakeholder within predefined time duration, each Flight Object Indentifier of the Flight Objects to recover and for which the stakeholder is the manager and “timeout” error code will be added to the recovery exception list.
Title	ATCFlightObjectControl RequestFlightObjectsRecovery invocation of FlightObjectManagement RequestFlightObjectsRecovery operation.
Status	<In Progress>
Rationale	This operation allows an application to request recovery of a subset or all of the Flight Objects for which it is in a specific recovery tier. Once validity checking is OK, the SWIM-TI will initiate calls to FlightObjectManagement interface RequestFORecovery operation to all the managers of the Flight Objects to recover. This implies that one ATCFlightObjectControl.RequestFlightObjectsRecovery request might trigger several FlightObjectManagement.RequestFlightObjectsRecovery requests directed to different stakeholders. A return_code of TRUE is set if all the FlightObjectManagement. RequestFlightObjectsRecovery requests are accepted by the respective FDMPs. An ATCFlightObjectControl.RequestFlightObjectsRecovery call returns immediately when all FlightObjectManagement.RequestFlightObjectsRecovery requests finish. A FlightObjectManagement. RequestFlightObjectsRecovery request finishes either when a response is received or when the request fails due to conditions corresponding to communication_failure or timeout. The actual SharedFlightObject publications of relevant FO data might take place after the ATCFlightObjectControl.RequestFlightObjectsRecovery returned.

Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.2.1.5 ReportFlightObjectServicesFailure

```
Report ReportFlightObjectServicesFailure(
           in IopStakeholderId fdmp requester id,
           in IopStakeholderId fdc requester id,
           in RequestId request id,
           in Responses services_reports);
```

3.3.9.1.2.1.5.1 Parameters

Input:

- **fdmp_requester_id: IopStakeholderId.** This parameter is the identifier of the IOP stakeholder reporting failure of services implementation, i.e. FDMP.
- **fdc_requester_id: IopStakeholderId.** This parameter is the identifier of the IOP stakeholder for which to report the failure, i.e. the FDC that have previous performed the service request.
- **request_id: RequestId.** Unique request identification generated by the FDC that have previous performed the service request that has failed.
- **services_reports: Responses.** Response to each of the FO service requests which implementation failed.

Output:

Report: Return value for the acceptance or the rejection of the request.

In addition to Report that may be returned by the IOP apploication, the following Report may be generated locally by the SWIM-TI as a result of validity checking by the SWIM-TI.

return_code	return_value	Comments
TRUE		Field *return_value* is not applicable.
FALSE	syntax_error	Malformed stakeholder identifier. Malformed service responses. Request Id is empty.
FALSE	semantic_error	Request Id not known by FDC. fdmp_requester_id and fdc_requester_id shall be different.
FALSE	timeout	No response from stakeholder (FDC) within the predefined time duration.
FALSE	middleware_failure	Internal middleware error (at FDC or at FDMP). Unknown Stakeholder (FDC).
FALSE	isolated_stakeholder	Stakeholder (FDC) is not IOP Enabled. Local middleware is not IOP Enabled.
FALSE	communication_failure	Communication with FDMP could not be established or is lost while a request is in progress, after the request was sent by FDC but before the reply from FDMP has been returned to application.

3.3.9.1.2.1.5.2 Validity Checking

The SWIM-TI shall be IOP enabled.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1215
Requirement	While serving an ATCFlightObjectControl ReportFlightObjectServicesFailure operation call, the SWIM-TI layer shall enforce the following validity checks: - The provided requester stakeholder identifiers must be not null, not empty and well formed syntax. In case this check is not completed successfully, the “syntax_error” error code shall be returned. - The provided services reports must be not null and not empty. In case this check is not completed successfully, the “syntax_error” error code shall be returned. - The provided request identifier must be not null and not empty. In case this check is not completed successfully, the “syntax_error” error code shall be returned. - The provided requester stakeholder identifiers must not be identical. In case this check is not completed successfully, the “semantic_error” error code shall be returned.
Title	ATCFlightObjectControl ReportFlightObjectServicesFailure operation validity checks
Status	<In Progress>
Rationale	In addition to Report that may be returned by the FDC, the SWIM-TI, as a result of validity checking, may generate locally additional error codes (Report with report code FALSE).
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>

Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1220
Requirement	While serving an ATCFlightObjectControl ReportFlightObjectServicesFailure operation call, the SWIM-TI layer shall return “middleware_failure” error code when the FDC’s stakeholder is not known, or when errors occur at the SWIM-TI layer.
Title	ATCFlightObjectControl ReportFlightObjectServicesFailure operation “middleware failure” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the FDC and by the SWIM-TI layer (validity checks), the requester is also informed when internal errors occur at the SWIM-TI layer.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1225
Requirement	While serving an ATCFlightObjectControl ReportFlightObjectServicesFailure operation call, the SWIM-TI layer shall return “isolated_stakeholder” error code to application when neither the local SWIM-TI layer nor the FDC stakeholder is IOP Enabled.
Title	ATCFlightObjectControl ReportFlightObjectServicesFailure operation “isolated stakeholder” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the FDC’s application layer and by the SWIM-TI layer (validity checks), the requester is also informed when neither the FDC not the local SWIM-TI layer is IOP Enabled.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.2.1.5.3 Semantic

This operation reports to an FDC failure of an already accepted ATCFlightObjectControl RequestFlightObjectServices operation call. This will trigger a call to the FlightObjectManagement ReportFlightObjectServicesFailure operation on the FDC.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1230
Requirement	Once successfully completed the validity checks applicable to the ATCFlightObjectControl ReportFlightObjectServicesFailure operation, the SWIM-TI layer shall invoke FlightObjectManagement RequestFlightObjectServices on the FDC and: - Return back to the requester with the invocation return, or - In case the communication with FDC cannot be established or is lost after request is sent to FDC and before reply from FDC is received by the SWIM-TI layer, the “communication_failure” error shall be returned. - If no response is availablefrom FDC within predefined time duration, the

	“timeout” error code shall be returned.
Title	ATCFlightObjectControl RequestFlightObjectServices invocation of FlightObjectManagement RequestFlightObjectServices operation.
Status	<In Progress>
Rationale	This operation reports to an FDC failure of an already accepted ATCFlightObjectControl RequestFlightObjectServices operation call. This will trigger a call to the FlightObjectManagement ReportFlightObjectServicesFailure operation on the FDC.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.2.2 Functional Requirements

Functional requirements provided in §3.3.9.1.2.1 are applicable.

3.3.9.1.2.3 Performance Requirements

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.2.4 Security Requirements

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.2.5 Data Transfer

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.2.6 Transaction

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.3 *SharedFlightObject* Interface Requirements

SharedFlightObject endpoints are provided on top of SWIM-TI SO and MSG and in particular on top of the Internal Interface FlightObjectDistribution which provides several operations used to serve properly the distribution of Flight Object data.

According to the design pattern adopted, this interface has been identified and described only at logical level because there are not interoperability needs. The *FlightObjectDistribution* interface is impacting the interoperability whereas the way locally the SWIM-TI layer interacts with the ATM layer is kept flexible and open.

[REQ]

Identifier	REQ-14.01.04-TS-0901.0316
Requirement	SharedFlightObject Interface shall be instantiated using the following binding: + Protocol stack: not standardised + MEPs: not standardised + Fault handling: not standardised + Encoding. not standardised + Security: - Confidentiality: not standardised - Integrity: not standardised - Authenticity: not standardised - Authorization: not standardised - Non-repudiation: not standardised + Contract: - formalism of contract description: UML, IDL, Text - minimum: not applicable - reference: Blue Profile Technical Specification, ISRM + Interoperability: not standardised
Title	SharedFlightObject Interface binding
Status	<Validated>
Rationale	This binding is not subjected to standardisation and is implementation specific. SharedFlightObject endpoints are provided on top of SWIM-TI SO and MSG and in particular on top of the Internal Interface FlightObjectDistribution which provides several operations used to serve properly the distribution of Flight Object data. According to the design pattern adopted, this interface has been identified and described only at logical level because there are no interoperability needs. In the introduction of §3.3.9 an illustrative IDL is provided to support the analysis and the development of this interface. The FlightObjectDistribution interface is impacting the interoperability whereas the way locally the SWIM-TI layer interacts with the ATM layer is kept flexible and open.
Category	<Interface>
Validation Method	
Verification Method	<Analysis>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Publisher>
Selfstanding set	<Service binding>
Conformance	<No>
High Level	<Yes>
Testability	<Applicable but not testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance

<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

In the figure below, SharedFlightObject service message types as specified in ISRM 2.0 are provided.

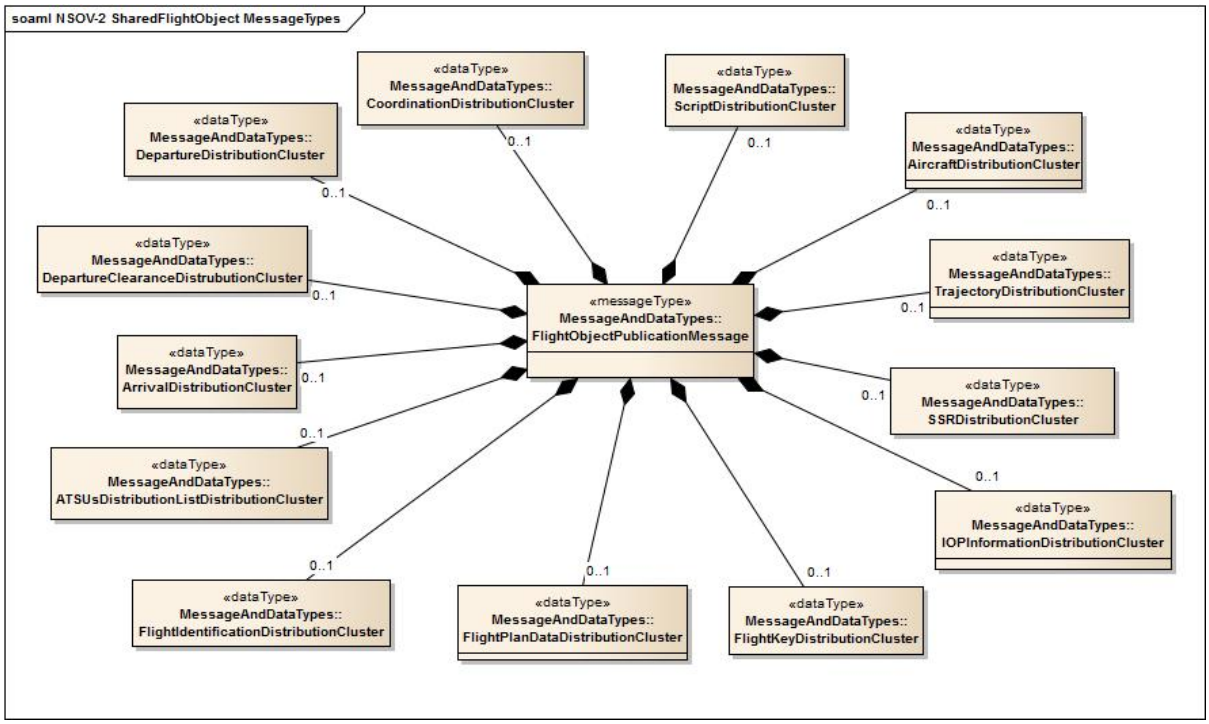


Figure 3-5: ISRM SharedFlightObject service message types

All the message types (e.g. FlightPlanDataDistributionCluster, TrajectoryDistributionCluster, etc.) represent clustered information concerning the Flight Object. At SWIM-TI layer each one of these message types are distributed via the *FlightObjectDistribution* interface using the same data type which envelops such ATM information into the Flight Object Cluster. FlightObjectDistribution interface and concerning data types are specified in §3.3.9.2.2. This specification supports the what-if concept introduced in Shared Object requirements chapter. In order to agree on Flight Object changes, two or more participants exchange What-if Flight Object (WIFO). A WIFO is an alternative Flight Object that is generated from a real Flight Object and contains the modifications needed to propose an alternative to the real one. A WIFO may include less clusters with respect to the real FO (e.g. just clusters the what-if proposal applies to).

3.3.9.1.3.1 Operations

3.3.9.1.3.1.1 CreateFlightObject

```
CreationReport CreateFlightObject(
      in FlightObjectIdentifier fo_id,
      in FlightKey flight id,
      in FoClusters fo clusters,
      in DistributionList distribution list);
```

3.3.9.1.3.1.1.1 Parameters

Input:

- **fo_id: FlightObjectIdentifier**. This parameter represents the IOP identifier of a Flight Object.
- **flight_id: FlightKey.** This parameter is the operational key of the flight. When all elements of the operational key are defined, it should identify uniquely the Flight Object.
- **fo_clusters: FoClusters.** This parameter is the set of all the clusters of the Flight Object and their identifiers.
- **distribution_list: DistributionList.** This parameter is the distribution list of the Flight Object with associated recovery tiers.

Output:

```
struct CreationReport {
      boolean report_code;
      ExceptionKind return value;
      FlightObjectIdentifier fo id;
};
```

CreationReport: Return value for the acceptance or the rejection of the request.
In case of duplicate_fo, the IOP identifier of the already existent FO will be returned.

return_code	return_value	Comments
TRUE		Field *return_value* is not applicable.
FALSE	fo_not_found	In case of What-If Flight Object the unique Flight Object Identifier does not refer to an existing real Flight Object.
FALSE	duplicated_fo	The Flight Object exists in the middleware and cannot be created again.
FALSE	syntax_error	Malformed stakeholder in distribution list. Malformed Flight Key.
FALSE	semantic_error	Missing distribution list. Unknown stakeholder in distribution list. Missing Flight Key. Missing cluster id. No Cluster provided.
FALSE	middleware_failure	Internal middleware error

3.3.9.1.3.1.1.2 Validity Checking

The fo_id must not exist in the IOP area (checking locally stored FO summaries). The flight_id must not exist in the IOP area when all elements in the flight key are present (including context_id for a WIFO).

The distribution list must be provided and members of the distribution list are known to the stakeholder. The recovery tiers must be provided.

The fo_clusters must be present and must contain all the FO clusters making up a Flight Object.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1235
Requirement	While serving a SharedFlightObject CreateFlightObject operation call, the SWIM-TI layer shall enforce the following validity checks: - The provided Flight Object Identifier must not exist in the IOP area. In case this check is not completed successfully, the “duplicated_fo” error code shall be returned. - In case of What-If Flight Object the unique Flight Object Identifier must refer to an existing real Flight Object. In case this check is not completed successfully, the “fo_not_found” error code shall be returned. - When the provided Flight Key has all attributes present, the Flight Key must not exist in the IOP area. In case this check is not completed successfully, the “duplicated_fo” error code shall be returned. - The provided stakeholder identifiers in the distribution list must be well formed syntax. In case this check is not completed successfully, the “syntax_error” error code shall be returned. - Provided attributes in the Flight Key must be well formed. In case this check is not completed successfully, the “syntax_error” error code shall be returned. - The provided distribution list must be not null and not empty. In case this check is not completed successfully, the “semantic_error” error code shall be returned. - All stakeholder identifiers in the provided distribution list must be known to the SWIM-TI layer. In case this check is not completed successfully, the “semantic_error” error code shall be returned. - The provided Flight Key must be not null and not empty. In case this check is not completed successfully, the “semantic_error” error code shall be returned. - The provided set of Flight Object Clusters must be not null and not empty. In case this check is not completed successfully, the “semantic_error” error code shall be returned. - The provided Flight Object Cluster identifiers must be valid. In case this check is not completed successfully, the “semantic_error” error code shall be returned. - The provided Flight Object Cluster releases must be valid (within the admissible range). In case this check is not completed successfully, the “semantic_error” error code shall be returned.
Title	SharedFlightObject CreateFlightObject operation validity checks
Status	<In Progress>
Rationale	The fo_id must not exist in the IOP area (checking locally stored FO summaries). The flight_id must not exist in the IOP area when all elements in the flight key are present (including context_id for a what-if flight objec). In case of what-if Flight Object the unique FO identifier must refer to an existing real Flight Object and the What-If context identifier must not exist for the same real Flight Object. The distribution list must be provided and members of the distribution list are known to the stakeholder. The recovery tiers must be provided. The fo_clusters must be present and must contain all the FO clusters making up a Flight Object.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>

Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1240
Requirement	While serving a SharedFlightObject CreateFlightObject operation call, the SWIM-TI layer shall return “middleware_failure” error code when errors occur at the SWIM-TI layer.
Title	SharedFlightObject CreateFlightObject operation “middleware_failure” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the SWIM-TI layer (validity checks), the requester is also informed when internal errors occur at the SWIM-TI layer.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>

<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.3.1.1.3 Semantic

This service operation allows the creation of a Flight Object for a flight for which there is no Flight Object already present in the IOP Area.

The fo_id and the flight_id must not already exist in the IOP area. The application is responsible for the provision of a unique FO identifier. When a Flight Object is created, an attribute creation_time shall be added by the SWIM-TI to the Flight Object to discriminate duplicated Flight Objects. A SUMMARY is created and published and then it is periodically published in the IOP Area via the FlightObjectDistribution interface (and received by the systems instances on the distribution list). In case of what-if Flight Object the unique FO identifier must refer to an existing real Flight Object and the What-If context identifier must not exist for the same real Flight Object. In case the Flight Object already exists in the IOP area, the return code is FALSE and the return value is duplicated_fo, and the FO Identifier of the already existent FO will be returned. In that case no new Flight Object is published. The flight_key parameter is used to update the operational key of the flight.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1245
Requirement	Once successfully completed the validity checks applicable to the SharedFlightObject CreateFlightObject operation, the SWIM-TI layer shall perform the following: - create the Flight Object, and - add a creation time attribute to the Flight Object, and - set own stakeholder identifier as the publisher for the Flight Object, and - store a local copy of the Flight Object, and - call FlightObjectDistribution to publish the Flight Object (summary and clusters) according to PSPUSH-MEP message exchange pattern to the members of the distribution list.
Title	SharedFlightObject CreateFlightObject operation implementation.
Status	<In Progress>
Rationale	This service operation allows the creation of a Flight Object for a flight for which there is no Flight Object already present in the IOP Area. The fo_id and the flight_id must not already exist in the IOP area. The application is responsible for the provision of a unique FO identifier. When a Flight Object is created, an attribute creation_time shall be added by the SWIM-TI to the Flight Object to discriminate duplicated Flight Objects. A SUMMARY is created and published and then it is periodically published in the IOP Area via the FlightObjectDistribution interface (and received by the systems instances on the distribution list). In case of what-if Flight Object the unique FO identifier must refer to an existing real Flight Object and the What-If context identifier must not exist for the same real Flight Object. In case the Flight Object already exists in the IOP area, the return code is FALSE and the return value is duplicated_fo, and the FO Identifier of the already existent FO will be returned. In that case no new Flight Object is published. The flight_key parameter is used to update the operational key of the flight.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>

Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.3.1.2 PublishFlightObject

```
Report PublishFlightObject(
      in FlightObjectIdentifier fo id,
       in FoClusters fo clusters,
       in FlightKey flight id,
       in DistributionList distribution_list,
       in Qos priority);
```

3.3.9.1.3.1.2.1 Parameters

Input:

- **fo_id : FlightObjectIdentifier**. This parameter represents the IOP identifier of a Flight Object.
- **fo_clusters : FoClusters.** This parameter represents the subset of the Flight Object clusters that are modified and their identifiers.
- **flight_id: FlightKey.** This parameter is the flight key of the FO. It is set to the special value null when it is not updated or contains the new version of the flight key when it has been updated.
- **distribution_list: DistributionList.** This parameter is the distribution list of the FO with associated recovery tiers. This input parameter is passed when the distribution list of the FO has to be modified.
- **priority: Qos:** This parameter represents the priority that is assigned to the publication. It is one of the three possible quality of services offered by the IOP-MDW (`d_1, d_2, d_3`)

Each cluster that composes the FO has a release Id. It is incremented by the Application upon update of that cluster.

Output:

Report: Return value for the acceptance or the rejection of the request.

return_code	return_value	Comments
TRUE		Field *return_value* is not applicable.

FALSE	fo_not_found	The Flight Object is not known (does not exist).
FALSE	syntax_error	Malformed parameter (flight_id and/or distribution_list contain a malformed element).
FALSE	semantic_error	Incorrect stakeholder in distribution list. Name or release id cluster is different with summary information. Invalid range for Release (cluster release id is out of range). Unknown Cluster (Known clusters are set at FO creation). Input clusters list should not be null.
FALSE	middleware_failure	Internal middleware error.
FALSE	old_fo_version	Release id is older than or equal to locally stored one.
FALSE	duplicated_fo	FlightKey with all the fields present already exists and already used for another FO.
FALSE	fo_version_collision	ReleaseId collides with the release of the locally stored FO.

3.3.9.1.3.1.2.2 Validity Checking

- The fo_id must exist in the list of FOs stored in the local system instance. In case it doesn't exist the Report shall return "FALSE", with the reason fo_not_found.
- When provided (not null), the distribution list must contain a valid list of stakeholders, and the recovery tiers must be provided.
- The cluster release ids has to fulfil the following checks for each of the clusters to be updated:

  Let Rcurrent be the locally stored FOReleaseId, Rnew be the new FOReleaseId from fo_clusters, where R[i] is the release version of cluster i in the FOReleaseId R.

  IF (for each cluster i Rcurrent[i] ≤ Rnew[i] and there is at least one cluster j where Rcurrent[j] < Rnew[j]) THEN OK.

  IF (for each cluster i Rnew[i] ≤ Rcurrent[i] and there is at least one cluster j where Rnew[j] < Rcurrent[j]) THEN NOK with reason old_fo_version.

  IF (for each cluster i Rnew[i] = Rcurrent[i] and Rcurrent was published locally) THEN NOK with reason old_fo_version.

  OTHERWISE NOK with fo_version_collision.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1250
Requirement	While serving a SharedFlightObject PublishFlightObject operation call, the SWIM-TI layer shall enforce the following validity checks: - The provided Flight Object Identifier must exist in the IOP area. In case this check is not completed successfully, the "fo_not_found" error code shall be returned. - When provided (not null and not empty), the Distribution List must contain well formed stakeholder identifiers. In case this check is not completed

	successfully, the “syntax_error” error code shall be returned. - When provided (not null and not empty), the Flight Key must contain well formed attributes. In case this check is not completed successfully, the “syntax_error” error code shall be returned. - When provided (not null and not empty), the Distribution List must contain stakeholder identifiers know by the SWIM-TI layer. In case this check is not completed successfully, the “semantic_error” error code shall be returned. - The provided set of Flight Object Clusters must be not null and not empty. In case this check is not completed successfully, the “semantic_error” error code shall be returned. - The provided Flight Object Cluster identifiers must be valid. In case this check is not completed successfully, the “semantic_error” error code shall be returned. - The provided Flight Object Cluster releases must be valid (within the admissible range). In case this check is not completed successfully, the “semantic_error” error code shall be returned. - The Flight Object Release obtained by updating the locally storead Flight Object with provided Cluster release must not generate a collision with locally stored Flight Object Release. In case this check is not completed successfully, the “fo_version_collision” error code shall be returned. - For each provided Flight Object Cluster, its release number must be higher that the locally stored cluster release number for the target Flight Object. In case this check is not completed successfully, the “old_fo_version” error code shall be returned.
Title	SharedFlightObject PublishFlightObject operation validity checks
Status	<In Progress>

Rationale	The fo_id must exist in the list of FOs stored in the local system instance. In case it doesn’t exist the Report shall return “FALSE”, with the reason fo_not_found.
	When provided (not null), the distribution list must contain a valid list of stakeholders, and the recovery tiers must be provided.
	The cluster release ids has to fulfil the following checks for each of the clusters to be updated:
	Let Rcurrent be the locally stored FOReleaseId, Rnew be the new FOReleaseId from fo_clusters, where R[i] is the release version of cluster i in the FOReleaseId R.
	IF (for each cluster i Rcurrent[i] ≤ Rnew[i] and there is at least one cluster j where Rcurrent[j] < Rnew[j]) THEN OK.
	IF (for each cluster i Rnew[i] ≤ Rcurrent[i] and there is at least one cluster j where Rnew[j] < Rcurrent[j]) THEN NOK with reason old_fo_version.
	IF (for each cluster i Rnew[i] = Rcurrent[i] and Rcurrent was published locally) THEN NOK with reason old_fo_version.
	OTHERWISE NOK with fo_version_collision.

Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1255
Requirement	While serving a SharedFlightObject PublishFlightObject operation call, the SWIM-TI layer shall return “middleware_failure” error code when errors occur at the SWIM-TI layer.
Title	SharedFlightObject PublishFlightObject operation “middleware_failure” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the SWIM-TI layer (validity checks), the requester is also informed when internal errors occur at the SWIM-TI layer.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A

<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.3.1.2.3 Semantic

In case the validity checking is OK, the SWIM-TI will update its locally stored clusters then automatically identify itself as the FDMP of the FO. The modified clusters will be sent to the distribution list. The publication will take into account the required priority of the modification provided in the invocation. The flight_id parameter is used to update the operational key of the flight. This parameter is the flight key of the FO. It contains the new value of the flight key when it is updated, or a null value when there is no change.

The distribution_list parameter is used to update the distribution list of the Flight Object. It contains the new value of the distribution list with associated recovery tiers allocation when it is updated. An empty distributuion list is used when there is no change to the current distribution list.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1260
Requirement	Once successfully completed the validity checks applicable to the SharedFlightObject PublishFlightObject operation, the SWIM-TI layer shall perform the following: - When the Flight Key is provided, the SWIM-TI updates the flight key of the Flight Object. When all the flight key attributes are present and correspond to an already existant Flight Object, the SWIM-TI will not update the flight key of the Flight Object and will return “duplicated_fo” error code. - When the Distribution List is provided, the SWIM-TI updates the distribution list of the Flight Object. - Set own stakeholder identifier as the publisher for the Flight Object. - Update local copy of the Flight Object, and - Call FlightObjectDistribution to publish the Flight Object updates (summary and clusters) according to PSPUSH-MEP message exchange pattern to the members of the distribution list.
Title	SharedFlightObject PublishFlightObject implementation.
Status	<In Progress>
Rationale	In case the validity checking is OK, the SWIM-TI will update its locally stored clusters then automatically identify itself as the FDMP of the FO. The modified clusters will be sent to the distribution list. The publication will take into account the required priority of the modification provided in the invocation. The flight_id parameter is used to update the operational key of the flight. This parameter is the flight key of the FO. It contains the new value of the flight key when it is updated or a null value when there is no change. The distribution_list parameter is used to update the distribution list of the Flight Object. It contains the new value of the distribution list with associated recovery tiers allocation when it is updated. An empty distributuion list is used when there is no change to the current distribution list.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>

Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.3.1.3 SearchFlightObject

```
SearchReport SearchFo(in FlightKey flight id);
```

3.3.9.1.3.1.3.1 Parameters

Input:

- **flight_id : FlightKey.** This parameter represents either a subset or the entire operational key of the flight to be searched.

Output:

```
struct SearchReport{
      Report report_value;
      FlightKeySequence flight_ids;
      FlightObjectIdentifiers fo_ids;
    };
```

- **SearchReport:** Return value including the result of the search operation. The first field is the report value (acceptance/rejection and reason). The second field reports an array of operational keys matching the input `flight_id`. The third is an array of the Flight Object identifiers corresponding to the reported operational keys. In case of rejection both arrays are empty.

SearchReport: Return value for the acceptance or the rejection of the request.

return_code	**return_value**	**Comments**
TRUE		Field *return_value* is not applicable.
FALSE	fo_not_found	The Flight Object is not known (does not exist).
FALSE	syntax_error	Malformed key (at least one of the key elements is malformed).
FALSE	semantic_error	Missing Flight Key.

FALSE	middleware_failure	Internal middleware error

3.3.9.1.3.1.3.2 Validity Checking

The *flight_id* must match one of the identifiers stored in the Flight Object summaries in the local SWIM-TI instance.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1265
Requirement	While serving a SharedFlightObject SearchFlightObject operation call, the SWIM-TI layer shall enforce the following validity checks: - Provided attributes in the Flight Key must be well formed. In case this check is not completed successfully, the “syntax_error” error code shall be returned. - The provided Flight Key must be not null and not empty. In case this check is not completed successfully, the “semantic_error” error code shall be returned.
Title	SharedFlightObject SearchFlightObject operation validity checks
Status	<In Progress>
Rationale	Searching for Flight Objects requires a valid operational key (Flight Key).
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1270
Requirement	While serving a SharedFlightObject SearchFlightObject operation call, the SWIM-TI layer shall return “middleware_failure” error code when errors occur at the SWIM-TI layer.
Title	SharedFlightObject SearchFlightObject operation “middleware_failure” error code

Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the SWIM-TI layer (validity checks), the requester is also informed when internal errors occur at the SWIM-TI layer.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.3.1.3.3 Semantic

This operation is to identify the FO(s) in the local SWIM-TI instance that match a given operational key (FlightKey) or a subset of its attributes. If a subset of the parameters is used, the other parameters shall be set as “null”. If many Flight Objects match the request, their FO identifiers and Operational Keys will be inserted in the return value.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1275
Requirement	Once successfully completed the validity checks applicable to the SharedFlightObject SearchFlightObject operation, the SWIM-TI layer shall identify the Flight Objects in the local SWIM-TI instance that match the attributes present in the provided flight key (search criteria). - If many Flight Objects match the search criteria, the SWIM-TI shall return their Flight Object Identifiers and Operational Keys. - If no Flight Object matches the search criteria, the SWIM-TI shall return “fo_not_found” error code.
Title	SharedFlightObject SearchFlightObject operation implementation.
Status	<In Progress>
Rationale	This operation is to identify the FO(s) in the local SWIM-TI instance that match a given operational key (FlightKey) or a subset of its attributes. If a subset of the parameters is used, the other parameters shall be set as “null”. If many Flight Objects match the request, their FO identifiers and Operational Keys will be inserted in the return value.

Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.3.1.4 DeleteFlightObject

```
Report DeleteFlightObject(
       in FlightObjectIdentifier fo_id);
```

3.3.9.1.3.1.4.1 Parameters

Input:

- **fo_id : FlightObjectIdentifier.** This parameter is the Flight Object identifier of the Flight Object to delete.

Output:

Report: Return value for the acceptance or the rejection of the request.

return_code	return_value	Comments
TRUE		Field *return_value* is not applicable.
FALSE	fo_not_found	The Flight Object is not known (does not exist).
FALSE	middleware_failure	Internal middleware error.
FALSE	not_eligible	The delete operation is not invoked from the Flight Object manager. Only locally managed FOs can be deleted.

3.3.9.1.3.1.4.2 Validity Checking

The fo_id must exist and the Flight Object clusters must be locally available in the SWIM-TI. The SWIM-TI must already be the publisher of Flight Object to allow its deletion.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1280
Requirement	While serving a SharedFlightObject DeleteFlightObject operation call, the SWIM-TI layer shall enforce the following validity checks: - The provided Flight Object Identifier must exist in the IOP area. In case this check is not completed successfully, the “fo_not_found” error code shall be returned. - The requester must be the current manager of the Flight Object. In case this check is not completed successfully, the “not_eligible” error code shall be returned.
Title	SharedFlightObject DeleteFlightObject operation validity checks
Status	<In Progress>
Rationale	The fo_id must exist and the Flight Object clusters must be locally available in the SWIM-TI. The SWIM-TI must already be the publisher of Flight Object to allow its deletion.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1285
Requirement	While serving a SharedFlightObject DeleteFlightObject operation call, the SWIM-TI layer shall return “middleware_failure” error code when errors occur at the SWIM-TI layer.
Title	SharedFlightObject DeleteFlightObject operation “middleware_failure” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the SWIM-TI layer (validity checks), the requester is also informed when internal errors occur at the SWIM-TI layer.
Category	<Interface>

Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.3.1.4.3 Semantic

This operation allows the application to request the deletion of a Flight Object by the SWIM-TI. The SWIM-TI will stop the publication of the FO and the related FO summary.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1290
Requirement	Once successfully completed the validity checks applicable to the SharedFlightObject SearchFlightObject operation, the SWIM-TI layer shall request FlightObjectDistribution to delete the Flight Object from the IOP area.
Title	SharedFlightObject DeleteFlightObject operation implementation.
Status	<In Progress>
Rationale	This operation allows the application to request the deletion of a Flight Object by the SWIM-TI. The SWIM-TI will stop the publication of the FO and the related FO summary.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.3.2 Functional Requerements

Functional requirements provided in §3.3.9.1.3.1 are applicable.

3.3.9.1.3.3 Performance Requirements

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.3.4 Security Requirements

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.3.5 Data Transfer

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.3.6 Transaction

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.4 *IOPMonitoring* Interface Requirements

This interface is provided by the SWIM-TI later and used by the application layer.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1295
Requirement	IOPMonitoring Interface shall be instantiated using the following binding: + Protocol stack: not standardised + MEPs: not standardised + Fault handling: not standardised + Encoding. not standardised + Security: - Confidentiality: not standardised - Integrity: not standardised - Authenticity: not standardised - Authorization: not standardised - Non-repudiation: not standardised + Contract: - formalism of contract description: UML, IDL, Text - minimum: not applicable - reference: Blue Profile Technical Specification, ISRM + Interoperability: not standardised
Title	IOPMonitoring Interface binding
Status	<Validated>
Rationale	This binding is not subjected to standardisation and is implementation specific. IOPMonitoring endpoints are provided on top of SWIM-TI SO and MSG and in particular on top of the Internal Interface FlightObjectDistribution which provides several operations used to serve properly IOPMonitoring consumption/provisioning. According to the design pattern adopted, this interface has been identified and described only at logical level because there are no interoperability needs. In the introduction of §3.3.9 an illustrative IDL is provided to support the analysis and the development of this interface. The FlightObjectDistribution interface is impacting the interoperability whereas the way locally the SWIM-TI layer interacts with the ATM layer is kept flexible and open.
Category	<Interface>
Validation Method	
Verification Method	<Analysis>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Service binding>
Conformance	<No>
High Level	<Yes>
Testability	<Applicable but not testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>

<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.4.1 Operations

3.3.9.1.4.1.1 UpdateApplicationStatus

```
Report UpdateApplicationStatus(
        in IopStakeholderId stakeholder_id,
        in IopStatus status);
```

3.3.9.1.4.1.1.1 Parameters

Input:

- **stakeholder_id**: **IopStakeholderId**. It represents the stakeholder identifier calling the operation.
- **status: IopStatus.** This parameter represents the status of the application (enabled or not_enabled)

Output:

Report: Return value for the acceptance or the rejection of the request.

return_code	**return_value**	**Comments**
TRUE		Field *return_value* is not applicable.
FALSE	syntax_error	Malformed parameter (requester_id).
FALSE	middleware_failure	Internal middleware error
FALSE	isolated_stakeholder	Local middleware is not IOP Enabled.

3.3.9.1.4.1.1.2 Validity Checking

The SWIM-TI must be 'enabled', otherwise it must reply with an "*isolated_stakeholder*" error code.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1300
Requirement	While serving a SharedFlightObject UpdateApplicationStatus operation call, the SWIM-TI layer shall enforce the following validity check: - The provided requester stakeholder identifier must be not null, not empty and well formed syntax. In case this check is not completed successfully, the "syntax_error" error code shall be returned.
Title	SharedFlightObject UpdateApplicationStatus operation validity checks
Status	<In Progress>
Rationale	The requester identifier must be present and well formed.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>

Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1305
Requirement	While serving a SharedFlightObject UpdateApplicationStatus operation call, the SWIM-TI layer shall return “middleware_failure” error code when errors occur at the SWIM-TI layer.
Title	SharedFlightObject UpdateApplicationStatus operation “middleware_failure” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the SWIM-TI layer (validity checks), the requester is also informed when internal errors occur at the SWIM-TI layer.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>

<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1310
Requirement	While serving an ATCFlightObjectControl UpdateApplicationStatus operation call, the SWIM-TI layer shall return “isolated_stakeholder” error code to application when the local SWIM-TI layer is not IOP Enabled.
Title	ATCFlightObjectControl UpdateApplicationStatus operation “isolated stakeholder” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the FDMP’s application layer and by the SWIM-TI layer (validity checks), the requester is also informed when the local SWIM-TI layer is not IOP Enabled.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.4.1.1.3 Semantic

The application layer calls this operation to notify the SWIM-TI of changes of its status (enabled or not_enabled). This operation triggers the publication of the IOP Status to the IOP Area.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1315
Requirement	Once successfully completed the validity checks applicable to the SharedFlightObject SearchFlightObject operation, the SWIM-TI layer shall request FlightObjectDistribution to publish the local IOP Status to IOP area.
Title	SharedFlightObject UpdateApplicationStatus operation implementation.
Status	<In Progress>
Rationale	The application layer calls this operation to notify the SWIM-TI of changes of its status (enabled or not enabled). This operation triggers the publication of

	the IOP Status to the IOP Area.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.4.1.2 UpdateRecoveryStatus

```
Report UpdateRecoveryStatus(
              in RecoveryStatus recovery status,
              in RecoveryContext context id,
              in Tiers recovery tiers);
```

3.3.9.1.4.1.2.1 Parameters

Input:

- **recovery_status: RecoveryStatus.** This parameter indicates that the SWIM node is currently performing a recovery or that the recovery process is completed.
- **context_id: RecoveryContext.** This parameter consists of a unique identifier for the on-going recovery.
- **recovery_tiers: Tiers.** Tier sequence specifying the current recovery Tier(s) of the SWIM Node when the SWIM Node recovery status is true.

Output:

Report: Return value for the acceptance or the rejection of the request.

return_code	**return_value**	**Comments**
TRUE		Field *return_value* is not applicable.
FALSE	semantic_error	Tier value is not within a valid range.
FALSE	isolated_stakeholder	Local middleware is not IOP Enabled.
FALSE	middleware_failure	Internal middleware error.

3.3.9.1.4.1.2.2 Validity Checking

The local SWIM Node must be IOP enabled.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1320
Requirement	While serving a SharedFlightObject UpdateRecoveryStatus operation call, the SWIM-TI layer shall enforce the following validity check: - Each provider recovery Tier must be a valid range. In case this check is not completed successfully, the “semantic_error” error code shall be returned.
Title	SharedFlightObject UpdateRecoveryStatus operation validity checks
Status	<In Progress>
Rationale	The requester identifier must be present and well formed.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1325
Requirement	While serving a SharedFlightObject UpdateRecoveryStatus operation call, the SWIM-TI layer shall return “middleware_failure” error code when errors occur at the SWIM-TI layer.
Title	SharedFlightObject UpdateRecoveryStatus operation “middleware_failure” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the SWIM-TI layer (validity checks), the requester is also informed when internal errors occur at the SWIM-TI layer.
Category	<Interface>

Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1330
Requirement	While serving an ATCFlightObjectControl UpdateRecoveryStatus operation call, the SWIM-TI layer shall return “isolated_stakeholder” error code to application when the local SWIM-TI layer is not IOP Enabled.
Title	ATCFlightObjectControl UpdateRecoveryStatus operation “isolated_stakeholder” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the FDMP’s application layer and by the SWIM-TI layer (validity checks), the requester is also informed when the local SWIM-TI layer is not IOP Enabled.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>

<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.4.1.2.3 Semantic

This operation enables the application to initiate, maintain, and terminate a recovery for specific recovery tiers.

A TRUE value for recovery_status will initiate and/or maintain a recovery, while a FALSE value will terminate an on-going recovery.

Value of parameter recovery_status is only relevant when recovery_status is TRUE.

The recovery_context is unique per recovering SWIM node and is required to avoid unnecessary republications from an FDMP following reception of a TRUE recovery_status.

Once a recovery is initiated a RECOVERY_STATUS publication is performed periodically thanks to PS-MEP Recovery (FlightObjectDistribution interface).

[REQ]

Identifier	REQ-14.01.04-TS-0901.1335
Requirement	Once successfully completed the validity checks applicable to the SharedFlightObject SearchFlightObject operation, the SWIM-TI layer shall the SWIM-TI layer shall perform the following: - If the provided recovery status is TRUE and no recovery is in progress, the SWIM-TI shall request FlightObjectDistribution to start recovery using the provided recovery context. - If the provided recovery status is TRUE and a recovery is in progress, the SWIM-TI shall update the value of the recovery context to the provided recovery context and maintain the on-going recovery. - If the provided recovery status is FALSE, the SWIM-TI shall request FlightObjectDistribution to stop any ongoing recovery.
Title	SharedFlightObject UpdateRecoveryStatus operation implementation.
Status	<In Progress>
Rationale	This operation enables the application to initiate, maintain, and terminate a recovery for specific recovery tiers. A TRUE value for recovery_status will initiate and/or maintain a recovery (REQ-14.01.04-TS-0901.0826), while a FALSE value will terminate an on-going recovery (REQ-14.01.04-TS-0901.0792). Value of parameter recovery_status is only relevant when recovery_status is TRUE. The recovery_context is unique per recovering SWIM node and is required to avoid unnecessary republications from an FDMP following reception of a TRUE recovery_status. Once a recovery is initiated a RECOVERY_STATUS publication is performed periodically thanks to PS-MEP Recovery (FlightObjectDistribution interface).
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>

Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.4.2 Functional Requerements

Functional requirements provided in §3.3.9.1.4.1 are applicable.

3.3.9.1.4.3 Performance Requirements

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.4.4 Security Requirements

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.4.5 Data Transfer

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.4.6 Transaction

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.5 IOPMonitoringApplication Interface Requirements

This interface is provided by the application layer and used by the SWIM-TI layer.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1340
Requirement	IOPMonitoringApplication Interface shall be instantiated using the following binding: + Protocol stack: not standardised + MEPs: not standardised + Fault handling: not standardised + Encoding. not standardised + Security: - Confidentiality: not standardised - Integrity: not standardised - Authenticity: not standardised - Authorization: not standardised - Non-repudiation: not standardised + Contract: - formalism of contract description: UML, IDL, Text - minimum: not applicable - reference: Blue Profile Technical Specification, ISRM + Interoperability: not standardised
Title	IOPMonitoringApplication Interface binding
Status	<Validated>
Rationale	This binding is not subjected to standardisation and is implementation specific. IOPMonitoringApplication endpoints are provided on top of SWIM-TI SO and MSG and in particular on top of the Internal Interface FlightObjectDistribution which provides several operations used to serve properly IOPMonitoringApplication consumption/provisioning. According to the design pattern adopted, this interface has been identified and described only at logical level because there are no interoperability needs. In the introduction of §3.3.9 an illustrative IDL is provided to support the analysis and the development of this interface. The FlightObjectDistribution interface is impacting the interoperability whereas the way locally the SWIM-TI layer interacts with the ATM layer is kept flexible and open.
Category	<Interface>
Validation Method	
Verification Method	<Analysis>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Service binding>
Conformance	<No>
High Level	<Yes>
Testability	<Applicable but not testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>

<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.5.1 Operations

3.3.9.1.5.1.1 NotifyMiddlewareStatus

```
Report NotifyMiddlewareStatus(
                    in IopStakeholderId stakeholder,
                    in IopStatus status);
```

3.3.9.1.5.1.1.1 Parameters

Input:

- **stakeholder: IopStakeholderId**: Identifier of the stakeholder
- **status: IopStatus.** This parameter is the SWIM-TI status (iop enabled or not_enabled).

Output:

Report: Return value for the acceptance or the rejection of the request.
SWIM-TI is not required to process return code and value.

return_code	return_value	Comments
TRUE		SWIM-TI is not required to process return code and value.
FALSE		SWIM-TI is not required to process return code and value.

3.3.9.1.5.1.1.2 Validity Checking

- The parameter stakeholder must represent a valid stakeholder identifier in the IOP area.
- The parameter status must be one of the valid statuses.

3.3.9.1.5.1.1.3 Semantic

The SWIM-TI notifies the application of its own status changes.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1345
Requirement	The SWIM-TI shall notify the application of the SWIM-TI status changes.
Title	IOPMonitoringApplication NotifyMiddlewareStatus operation invocation.
Status	<In Progress>
Rationale	The SWIM-TI notifies the application of its own status changes.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>

Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.5.1.2 NotifyIOPAreaStatus

```
Report NotifyIOPAreaStatus(
                    in StatusSequence status);
```

3.3.9.1.5.1.2.1 Parameters

Input:

- **stakeholders_status: StatusSequence.** This parameter identifies the IOP Stakeholders in the whole IOP Area and their IOP Status.

Output:
Report: Return value for the acceptance or the rejection of the request.
SWIM-TI is not required to process return code and value.

return_code	return_value	Comments
TRUE		SWIM-TI is not required to process return code and value.
FALSE		SWIM-TI is not required to process return code and value.

3.3.9.1.5.1.2.2 Validity Checking

The parameter IopStakeholderId of stakeholder_status must represent stakeholder identifiers in the IOP area.

3.3.9.1.5.1.2.3 Semantic

This operation is used to notify the application of the status of the IOP Stakeholders in the whole IOP Area as received via IOP_STATUS topic.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1350
Requirement	The SWIM-TI shall inform the IOP application when it detects another IOP stakeholder that is not IOP Enabled becoming IOP Enabled or conversely an IOP stakeholder that is IOP Enabled becoming not IOP Enabled.
Title	IOPMonitoringApplication NotifyMiddlewareStatus operation invocation.
Status	<In Progress>
Rationale	This operation is used to notify the application of the status of the IOP Stakeholders in the whole IOP Area as received via FlightObjectDistribution.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.5.2 Functional Requerements

Functional requirements provided in §3.3.9.1.5.1 are applicable.

3.3.9.1.5.3 Performance Requirements

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.5.4 Security Requirements

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.5.5 Data Transfer

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.5.6 Transaction

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.6 SharedFlightObjectApplication Interface Requirements

This interface is provided by the application layer and used by the SWIM-TI layer.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1355
Requirement	SharedFlightObjectApplication Interface shall be instantiated using the following binding: + Protocol stack: not standardised + MEPs: not standardised + Fault handling: not standardised + Encoding. not standardised + Security: - Confidentiality: not standardised - Integrity: not standardised - Authenticity: not standardised - Authorization: not standardised - Non-repudiation: not standardised + Contract: - formalism of contract description: UML, IDL, Text - minimum: not applicable - reference: Blue Profile Technical Specification, ISRM + Interoperability: not standardised
Title	SharedFlightObjectApplication Interface binding
Status	<Validated>
Rationale	This binding is not subjected to standardisation and is implementation specific. SharedFlightObject endpoints are provided on top of SWIM-TI SO and MSG and in particular on top of the Internal Interface FlightObjectDistribution which provides several operations used to serve properly the distribution of Flight Object data. According to the design pattern adopted, this interface has been identified and described only at logical level because there are no interoperability needs. In the introduction of §3.3.9 an illustrative IDL is provided to support the analysis and the development of this interface. The FlightObjectDistribution interface is impacting the interoperability whereas the way locally the SWIM-TI layer interacts with the ATM layer is kept flexible and open.
Category	<Interface>
Validation Method	
Verification Method	<Analysis>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Publisher>
Selfstanding set	<Service binding>
Conformance	<No>
High Level	<Yes>
Testability	<Applicable but not testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>

<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.6.1 Operations

3.3.9.1.6.1.1 NotifyFlightObject

```
Report NotifyFlightObject(
      in FlightObjectIdentifier fo_id,
      in FoClusters clusters,
      in Qos priority);
```

3.3.9.1.6.1.1.1 Parameters

Input:

- **fo_id** : **FlightObjectIdentifier**. Identifier of the Flight Object in the IOP area.
- **clusters : FoClusters.** This parameter represents a set of unparsed clusters.
- **priority : Qos**. This parameter represents the priority of the notification.

Output:

Report: Return value for the acceptance or the rejection of the request.
SWIM-TI is not required to process return code and value.

return_code	return_value	Comments
TRUE		SWIM-TI is not required to process return code and value.
FALSE		SWIM-TI is not required to process return code and value.

3.3.9.1.6.1.1.2 Validity Checking

A Flight Object update is valid when the following conditions are ALL OK:

- Received summary and clusters are from the same publisher.
- For each cluster with ClusterReleaseId is the FOReleaseId indicated in the summary, a payload is available (received or locally stored).
- Let Rcurrent be the locally stored FOReleaseId, and Rnew be the newly received FOReleaseId.

  For each cluster i Rcurrent[i] ≤ Rnew[i] and there is at least one cluster j where Rcurrent[j] < Rnew[j]

  Where R[i] is the release version of cluster i in the FOReleaseId R.

3.3.9.1.6.1.1.3 Semantic

The SWIM-TI calls this operation when it receives a valid Flight Object update (set of FO_CLUSTERs and associated FO_SUMMARY data) to pass it on to the application.

Refer to REQ-14.01.04-TS-0901.1090.

3.3.9.1.6.1.2 NotifyException

```
Report NotifyException(
         in FlightObjectIdentifier fo_id,
         in IopStakeholderId publisher,
         in ClusterReleaseIdSequence fo release,
         in ExceptionKind reason);
```

3.3.9.1.6.1.2.1 Parameters

Input:

- **fo_id** : **FlightObjectIdentifier**. Identifier of the Flight Object.
  - **publisher : IopStakeholderId.** This publisher of the Flight Object.
  - **fo_release : ClusterReleaseIdSequence.** This parameter is the list of the cluster identifiers and related release numbers for the whole FO.
  - **reason : ExceptionKind.** This parameter is the **SWIM-TI** exception.

Output:

Report: Return value for the acceptance or the rejection of the request.
SWIM-TI is not required to process return code and value.

return_code	return_value	Comments
TRUE		SWIM-TI is not required to process return code and value.
FALSE		SWIM-TI is not required to process return code and value.

3.3.9.1.6.1.2.2 Validity Checking

The parameter fo_id when used must represent a valid Flight Object identifier in the IOP area.

3.3.9.1.6.1.2.3 Semantic

This operation is used to notify the application layer of any exception raised by the SWIM-TI instance. Possible exceptions are listed in the table below.

It is also called to notify to the application, the reception of a Flight Object that is not aligned with the last release identifier, stored in the Flight Object summary. The flight_iop_identifier, the publisher and the fo release are reported to the application when needed. If not needed a special value “null” is to be used.

- old_fo_version : raised when the release id of a Flight Object is different from the one reported in the FO summary.
- fo_version_collision: raised when a concurrent update has been detected based on received FO summary.
- middleware_failure: raised when an internal SWIM-TI exception is raised.
- critical_error: raised for all the other cases.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1360
Requirement	The SWIM-TI shall inform the IOP application when the following exceptional conditions occur: - FlightObjectDistribution receives a Flight Object update with “Invalid Flight Object release” colliding with the locally stored Flight Object’s release. - FlightObjectDistribution receives a Flight Object update with “Invalid Flight Object release” that is prior to the locall stored Flight Object’s release. - An error internal to the SWIM-TI is detected.
Title	SharedFlightObjectApplication NotifyException operation invocation.
Status	<In Progress>
Rationale	This operation is used to notify the application layer of any exception raised by the SWIM-TI instance. Possible exceptions are listed in the table below. It is also called to notify to the application, the reception of a Flight Object that is not aligned with the last release identifier, stored in the Flight Object summary. The flight_iop_identifier, the publisher and the fo release are reported to the application when needed. If not needed a special value “null” is to be used. - old_fo_version: raised when the release id of a Flight Object is different from the one reported in the FO summary. - fo_version_collision: raised when a concurrent update has been detected based on received FO summary. - middleware_failure: raised when an internal SWIM-TI exception is raised. - critical_error: raised for all the other cases. Additional checks provided in REQ-14.01.04-TS-0901.1100 are also applicable.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.6.1.3 NotifyFlightObjectDeletion

```
Report NotifyFlightObjectDeletion(
      in FlightObjectIdentifier fo_id);
```

3.3.9.1.6.1.3.1 Parameters

Input:

- **fo_id: FlightObjectIdentifier.** This parameter is the identifier of the deleted Flight Object.

Output:
Report: Return value for the acceptance or the rejection of the request.
SWIM-TI is not required to process return code and value.

return_code	return_value	Comments
TRUE		SWIM-TI is not required to process return code and value.
FALSE		SWIM-TI is not required to process return code and value.

3.3.9.1.6.1.3.2 Validity Checking

The parameter fo_id must refer to valid Flight Object Identifier for which the SWIM-TI has already received the cluster payloads.

3.3.9.1.6.1.3.3 Semantic

This operation is invoked when a Flight Object has been deleted from the IOP Area. This is triggered by reception of disposed FO clusters and summary in the underlying FlightObjectDistribution interface.

The application will only be notified when the SWIM-TI has already received cluster payloads for the Flight Object (it does not make sense if only summaries are received locally).

[REQ]

Identifier	REQ-14.01.04-TS-0901.1365
Requirement	The SWIM-TI shall inform the IOP application when the FlightObjectDistribution detects that a locally stored Flight Object has been deleted from the IOP Area by another stakeholder via SharedFlightObject DeleteFlightObject operation.
Title	SharedFlightObjectApplication NotifyFlightObjectDeletion operation invocation.
Status	<In Progress>
Rationale	This operation is invoked when a Flight Object has been deleted from the IOP Area. This is triggered by reception of disposed FO clusters and summary in the underlying FlightObjectDistribution interface. The application will only be notified when the SWIM-TI has already received cluster payloads for the Flight Object (it does not make sense if only summaries are received locally).
Category	<Interface>

Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.6.2 Functional Requerements

Functional requirements provided in §3.3.9.1.6.1 are applicable.

3.3.9.1.6.3 Performance Requirements

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.6.4 Security Requirements

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.6.5 Data Transfer

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.6.6 Transaction

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.7 ATCFlightObjectControlApplication Interface Requirements

This interface is provided by the application layer and used by the SWIM-TI layer.

[REQ]

Identifier	REQ-14.01.04-TS-0901.1370
Requirement	ATCFlightObjectControlApplication Interface shall be instantiated using the following binding: + Protocol stack: not standardised + MEPs: not standardised + Fault handling: not standardised + Encoding. not standardised + Security: - Confidentiality: not standardised - Integrity: not standardised - Authenticity: not standardised - Authorization: not standardised - Non-repudiation: not standardised + Contract: - formalism of contract description: UML, IDL, Text - minimum: not applicable - reference: Blue Profile Technical Specification, ISRM + Interoperability: not standardised
Title	ATCFlightObjectControlApplication Interface binding
Status	<Validated>
Rationale	This binding is not subjected to standardisation and is implementation specific. ATCFlightObjectControlApplication endpoints are provided on top of SWIM-TI SO and MSG and in particular on top of the Internal Interface FlightObjectManagement which provides several operations used to serve properly ATCFlightObjectControlApplication consumption/provisioning. According to the design pattern adopted, this interface has been identified and described only at logical level because there are no interoperability needs. In the introduction of §3.3.9 an illustrative IDL is provided to support the analysis and the development of this interface. The FlightObjectManagement interface is impacting the interoperability whereas the way locally the SWIM-TI layer interacts with the ATM layer is kept flexible and open.
Category	<Interface>
Validation Method	
Verification Method	<Analysis>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Service binding>
Conformance	<No>
High Level	<Yes>
Testability	<Applicable but not testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>

<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.1.7.1 Operations

3.3.9.1.7.1.1 RequestFlightObjectServices

```
ComplexReport RequestFlightObjectServices (
      in IopStakeholderId requester_id,
      in RequestId request_id,
      in FlightObjectIdentifier fo_id,
      in ClusterReleaseIdSequence fo_release,
      in Requests fo_request,
      in Qos priority);
```

3.3.9.1.7.1.1.1 Parameters

Input:

- **requester_id : IopStakeholderId.** This parameter is the identifier of the IOP stakeholder requesting the service**.**
- **request_id** : **RequestId**. Request identification.
- **fo_id**: **FlightObjectIdentifier.** This parameter represents the unique identifier of Flight Object.
- **fo_release** : **ClusterReleaseIdSequence.** This parameter is the list of the cluster identifiers and related release numbers for the whole FO.
- **fo_request** : **Requests.** This parameter represents a set of application services to be requested to an application in another system instance.
- **priority: Qos.** This parameter represents the priority that is assigned to the request delivery. It is one of the three possible classes of services offered by the SWIM-TI (`d_1, d_2, d_3`).

Output:

ComplexReport: Return value for the acceptance or the rejection of the request. It contains the result of the operation(s) and the Report attribute.

3.3.9.1.7.1.1.2 Validity Checking

Validity checking is performed upon reception of FlightObjectManagement RequestFlightObjectServices operation.

3.3.9.1.7.1.1.3 Semantic

This operation is called on reception of a FlightObjectManagement RequestFlightObjectServices operation request and is used to deliver the request(s) to the application.

The SWIM-TI will report Application's response as a return to received FlightObjectManagement RequestFlightObjectServices operation request.

Refer to REQ-14.01.04-TS-0901.0930.

3.3.9.1.7.1.2 RejectFlightObject

```
Report RejectFlightObject(
      in IopStakeholderId stakeholder_id,
      in FlightObjectIdentifier fo id,
      in ClusterReleaseIdSequence fo release,
      in string reject_reason);
```

3.3.9.1.7.1.2.1 Parameters

Input:

- **requester_id : IopStakeholderId.** This parameter is the identifier of the IOP stakeholder rejecting the Flight Object.
- **fo_id : FlightObjectIdentifier.** This parameter is the identifier of the Flight Object to be rejected.
- **fo_release : ClusterReleaseIdSequence.** This parameter is the FO release of the Flight Object to be rejected. It contains all cluster identifiers and release numbers of the Flight Object.
- **reject_reason : string.** Reason for rejection.

The SWIM-TI will not interpret the reason of the rejection. It is assumed that IOP Application ICD will define all possible values for this parameter (possibly XML).

Output:
Report: Return value for the acceptance or the rejection of the request.

3.3.9.1.7.1.2.2 Validity Checking

The parameter requester_id must represent a stakeholder identifier in the IOP area. The parameter fo_id must refer to a Flight Object of which the receiving application is the FDMP.

3.3.9.1.7.1.2.3 Semantic

This operation allows the requesting application to notify to the target receiving application about a request to reject a Flight Object. This operation is called on reception of a FlightObjectManagement RejectFlightObject operation request and is used to deliver the request(s) to the application.

The SWIM-TI will report Application's response as a return to received FlightObjectManagement RejectFlightObject operation request.

Refer to REQ-14.01.04-TS-0901.0990.

3.3.9.1.7.1.3 ReportFlightObjectServicesFailure

```
Report ReportFlightObjectServicesFailure(
            in IopStakeholderId requester_id,
            in RequestId request_id,
            in Responses services_reports);
```

3.3.9.1.7.1.3.1 Parameters

Input:

- **requester_id : IopStakeholderId.** This parameter is the identifier of the IOP stakeholder reporting failure of services implementation, i.e. FDMP.

- **request_id : RequestId.** Unique request identification generated by the FDC that have previous performed the service request that has failed.
- **services_reports : Responses.** Response to each of the FO service requests which implementation failed.

Output:

Report: Return value for the acceptance or the rejection of the request.

3.3.9.1.7.1.3.2 Validity Checking

Validity checking is performed upon reception of FlightObjectManagement ReportFlightObjectServicesFailure operation.

3.3.9.1.7.1.3.3 Semantic

This operation reports to the FDC application failure of an already accepted API_APP_RequestFoService call. It is called on reception of a FlightObjectManagement ReportFlightObjectServicesFailure operation and is used to report to the application.

The SWIM-TI will report Application's response as a return to received FlightObjectManagement ReportFlightObjectServicesFailure operation request.

Refer to REQ-14.01.04-TS-0901.0960.

3.3.9.1.7.2 Functional Requerements

Functional requirements provided in §3.3.9.1.7.1 are applicable.

3.3.9.1.7.3 Performance Requirements

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.7.4 Security Requirements

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.7.5 Data Transfer

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.1.7.6 Transaction

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.2 Internal Service Interface Bindings

This paragraph provides all the needed details concerning the identified Internal SWIM Technical interfaces as introduced before.

3.3.9.2.1 *FlightObjectManagement* Interface Requirements

The FlightObjectManagement interface, as part of the ATCFlightObjectControl service physical architecture (refer to Figure 3-2) provides the following operations:

- *RequestFOService*, Flight Object Services Request operation (SRR-MEP),
- *ReportFOServiceFailure*, Flight Object Services Request processing failure (SRR-MEP),
- *RejectFO* , Flight Object Rejection operation (SRR-MEP),
- *RestoreFO*, Flight Object Restoring operation (SRR-MEP),
- *RequestFORecovery*, Flight Objects Recovery operation (SRR-MEP);

These operations are the remaining part of this chapter. In the remaining part of this paragraph the business model is described using SoaML diagrams, mainly capturing the Business Architecture Model level of definition.

The *ATCFlightObjectControl* services technical architecture provides a high level view about how the participants collaborate by providing and using the service.

Depicted in the services architecture are:

- Flight Object Management interface that provides above operations,
- Participants that are stakeholders in the service interface.

The participants have the common purpose to establish the information services to support information exchange depicted in the BPMN.

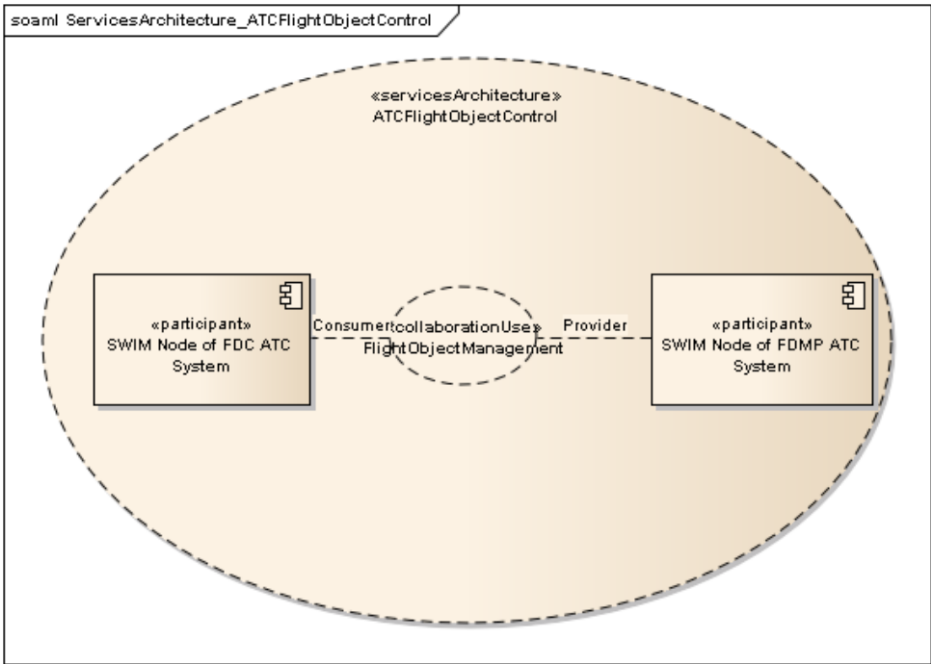


Figure 3-6: Flight Object Management Architecture

The participants represent logical or real organizational units taking part in the information exchange. The SoaML Flight Object Management participants include:

- SWIM Node of Flight Data Contributor System on the consumer side of the SWIM-TI which may call a particular service on another SWIM Node of Flight Data Manager/Publisher System;
- SWIM Node of Flight Data FO Manager/Publisher is a participant on the provider side of the SWIM-TI.

The information exchange between, the SWIM nodes is performed by the service operations exchange description provided in §3.3.9.2.1.1. Technology specific instantiation of the interface is provided in §3.3.9.2.1.7.

3.3.9.2.1.1 Operations

The operations included in the Flight Object Management Interface are the following:

- Flight Object Services Request
- Flight Object Services Request processing failure notification
- Flight Object Data Rejection
- Flight Object Data Restoring
- Flight Objects Recovery Request

The service contract for the FlightObjectManagement interface is depicted in the SoaML diagram below. The participants from the services architecture are shown in their consumer and provider roles.

The FlightObjectManagement service operations are synchronous and with a response that does not require interface implementation on the consumer side (the service contract includes only the provider interface specification). The interface for the provider is named FlightObjectManagement. The service contract is identified in the following SoaML diagram.

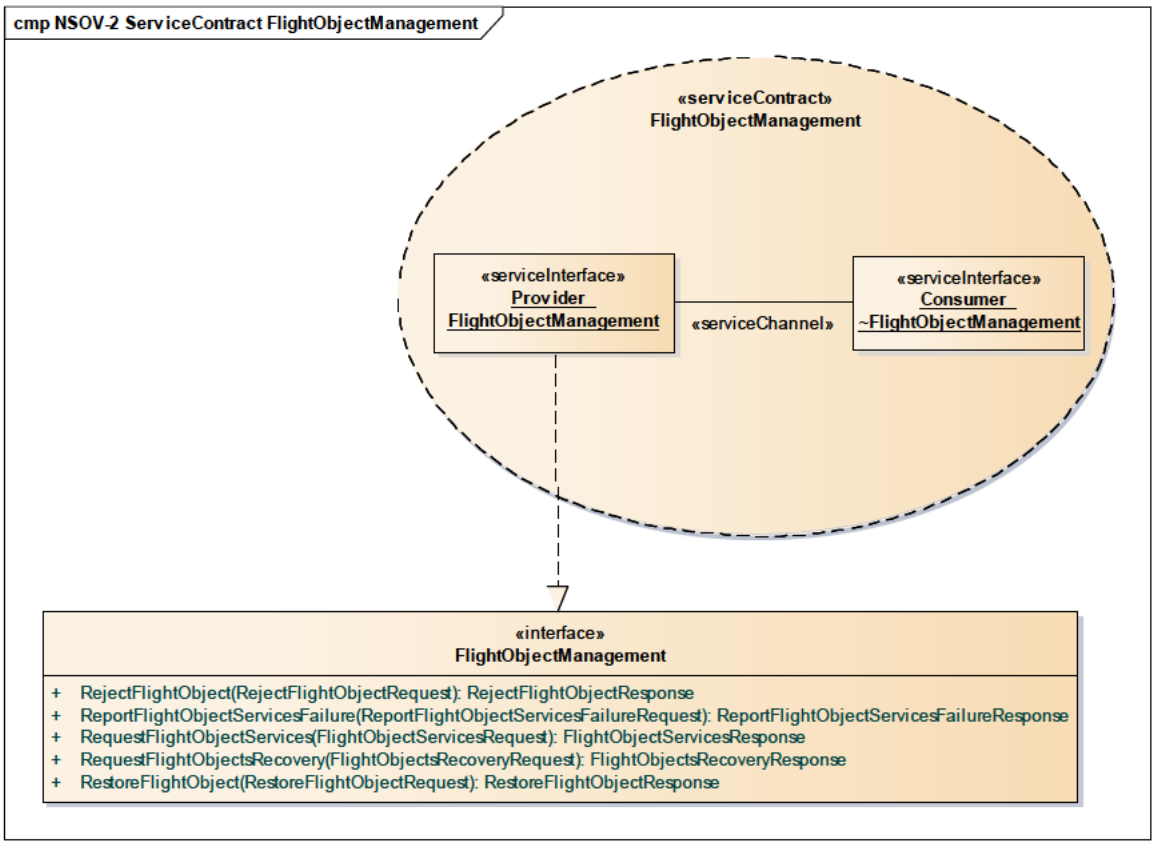


Figure 3-7: FlightObjectManagement Interface UML Contract

The FlightObjectManagement <<service point>> is a port for providing the service on the provider side (the port provides the provider interface). The FlightObjectManagement <<request point>> is a port for consuming the FlightObjectManagement service on the SWIM Node FO Contributor side (the port requires the provider interface).

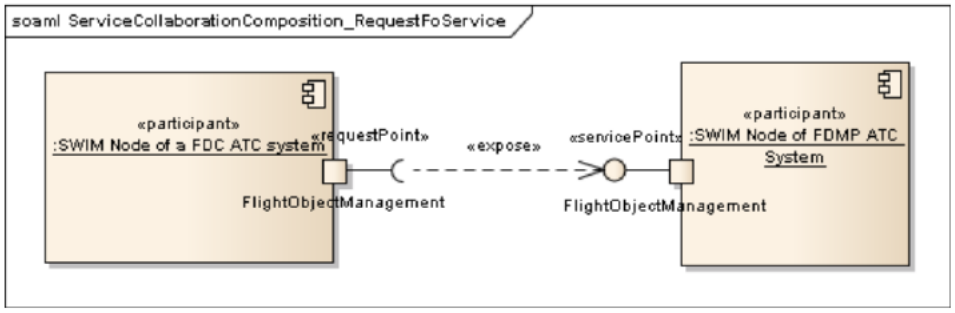


Figure 3-8: FlightObjectManagement Collaboration Composition

[REQ]

Identifier	REQ-14.01.04-TS-0901.0725
Requirement	The FlightObjectManagement Interface contract shall include the following operations: - Flight Object Services Request. - Flight Object Services Request processing failure notification. - Flight Object Data Rejection. - Flight Object Data Restoring.

	- Flight Objects Recovery Request.
Title	FlightObjectManagement Interface contract operations
Status	<Validated>
Rationale	To ensure that the FlightObjectManagement Interface contract provides the needed and only the identified operations.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Analysis>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<Yes>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0405
Requirement	The FlightObjectManagement Interface contract shall be specified according to the SRR-MEP.
Title	FlightObjectManagement Interface contract Message Exchange Pattern
Status	<Validated>
Rationale	To ensure that the FlightObjectManagement Interface contract is specified according to the expected Message Exchange Pattern identified for that interface.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Analysis>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<Yes>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance

<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.2.1.1.1 RequestFlightObjectServices

The Message Types for the `RequestFlightObjectServices` operation are a description of information exchanged between the service consumer (FDC) and provider (FDMP).

The Messages Types are described in the following table, also refer to the SoaML diagram.

Table 3-1: FlightObjectManagement Interface RequestFlightObjectServices Operation Message Types

Message Type	Brief Description
Input	
IopStakeholderId	It contains a complex data type representing the stakeholder identifier (defined as string)
RequestIdentifier	It contains a complex data type representing a unique identifier of the Flight Object services request being provided through the operation.
FlightObjectIdentifier	It contains a complex data type representing the unique flight object identifier. It allows also to distinguish between a real and a what-if Flight Object (default value for whatIfContextId is the empty string "" that denotes a real FO). At SWIM Technical layer, the IOP wide unique identifier of a real Flight Object is made of a unique universal identifier for the Flight Object and an empty What-If Context Identifier. For a What-If Flight Object, the IOP wide unique identifier is made of the unique universal identifier for the real Flight Object and a unique What-If Context Identifier.
ClusterReleaseId	It is a complex type representing the release of a given Flight Object cluster.
ClusterReleaseIdSequence	It is a sequence of ClusterReleaseId and it represents the Flight Object release
RequestSequence	It represents a container for an array of strings each one of them contains the XML representation of the operation (or application level service) requested to the stakeholder: array of the real operative intentions
QoS	It simply contains a complex data type value representing the quality of service expected

Output	
`ComplexReport`	It contains a complex data type which defines a container for: - an array of strings (defining the invocation result for each service request), - the “Report” data.
`Report`	It contains a complex data type (defining the report as two main attributes, (i) report code and (ii) report exception) representing error information whether the request is correctly checked and accepted (report code TRUE) or not (report code FALSE). Possible exceptions (report code FALSE) that may be generated by the SWIM-TI layer are described in dedicated requirements.

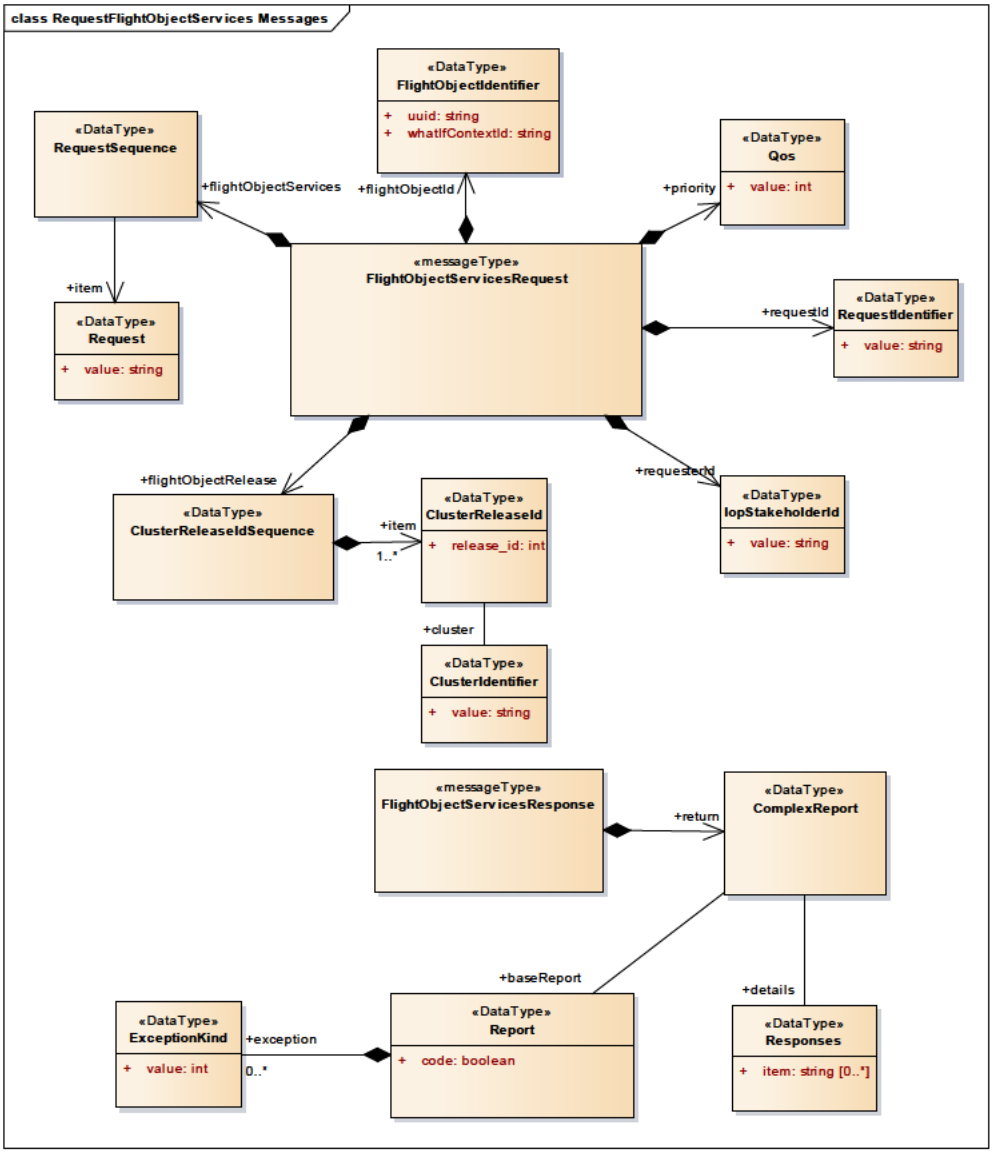


Figure 3-9: FlightObjectManagement Interface RequestFlightObjectServices Operation Message Type

[REQ]

Identifier	REQ-14.01.04-TS-0901.0435
Requirement	FlightObjectManagement RequestFlightObjectServices operation signature shall be: + Input Message: - IopStakeholderId - RequestIdentifier - FlightObjectIdentifier - ClusterReleaseIdSequence - RequestsSequence - QoS + Output Message: - ComplexReport + Fault Message: - : non specified
Title	FlightObjectManagement RequestFlightObjectServices operation signature
Status	<Validated>
Rationale	FlightObjectManagement RequestFlightObjectServices operation signature consists of: Input Message: - IopStakeholderId, which contains a complex data type representing the stakeholder identifier (defined as string). It is the identifier of the IOP stakeholder requesting the service. - RequestIdentifier, which contains a complex data type representing a unique identifier of the Flight Object services request being provided through the operation. - FlightObjectIdentifier, which contains a complex data type representing the unique Flight Object identifier. It allows also to distinguish between a real and a what-if Flight Object (default value for whatIfContextId is the empty string "" that denotes a real FO). At SWIM Technical layer, the IOP wide unique identifier of a real Flight Object is made of a unique universal identifier for the Flight Object and an empty What-If Context Identifier. For a What-If Flight Object, the IOP wide unique identifier is made of the unique universal identifier for the real Flight Object and a unique What-If Context Identifier. It identifies the Flight Object to which the services request applies to. - ClusterReleaseIdSequence, which represents the Flight Object release consisting of a sequence of a complex type representing the release of a given Flight Object cluster (cluster name-cluster release pair). It identifies the release of the Flight Object to which the services request applies to. - RequestsSequence, which represents a container for an array of strings each one of them contains the XML representation of the application services (real operative intentions) requested for a specific Flight Object. - QoS, which represents the priority that is assigned to the request delivery. It is one of the three possible classes of services offered by the SWIM-TI (d_1, d_2, d_3). Output Message: ComplexReport, which defines a container for - an array of strings (defining the invocation result for each service request), - a complex data type (defining the report as two main attributes, (i) report code and (ii) report exception) representing error information whether the request is correctly checked and accepted (report code TRUE) or not (report code FALSE). Possible exceptions (report code FALSE) that may be generated by the SWIM-TI layer are described in dedicated requirements. Semantic: The SWIM-TI on the receiving side enforces first the validity checks and, in

	case no error codes are generated, that layer triggers in synchronous way (blocking) a call to ATCFlightObjectControlApplication interface_RequestFlightObjectServices operation and responds with application return.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<Yes>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.2.1.1.2 ReportFlightObjectServicesFailure

The message types for the `ReportFlightObjectServicesFailure` operation is a description of information exchanged between the service consumer and provider.

The Messages Types are described in the following table, also refer to the SoaML diagram.

Table 3-2: FlightObjectManagement Interface ReportFlightObjectServicesFailure Operation Message Types

Message Type	Brief Description
Input	
`IopStakeholderId`	It contains a complex data type representing the stakeholder identifier (defined as string)
`RequestIdentifier`	It contains a complex data type representing a unique identifier of the Flight Object services request. It uniquely identifies the previously accepted RequestFOService request.
`FailureDescription`	It is a textual (string) description of the failure occurred while processing previously accepted RequestFOService request.

Output	
Report	It contains a complex data type (defining the report as two main attributes, (i) report code and (ii) report exception) representing error information whether the request is correctly checked and accepted (report code TRUE) or not (report code FALSE). Possible exceptions (report code FALSE) that may be generated by the SWIM-TI layer are described in dedicated requirements.

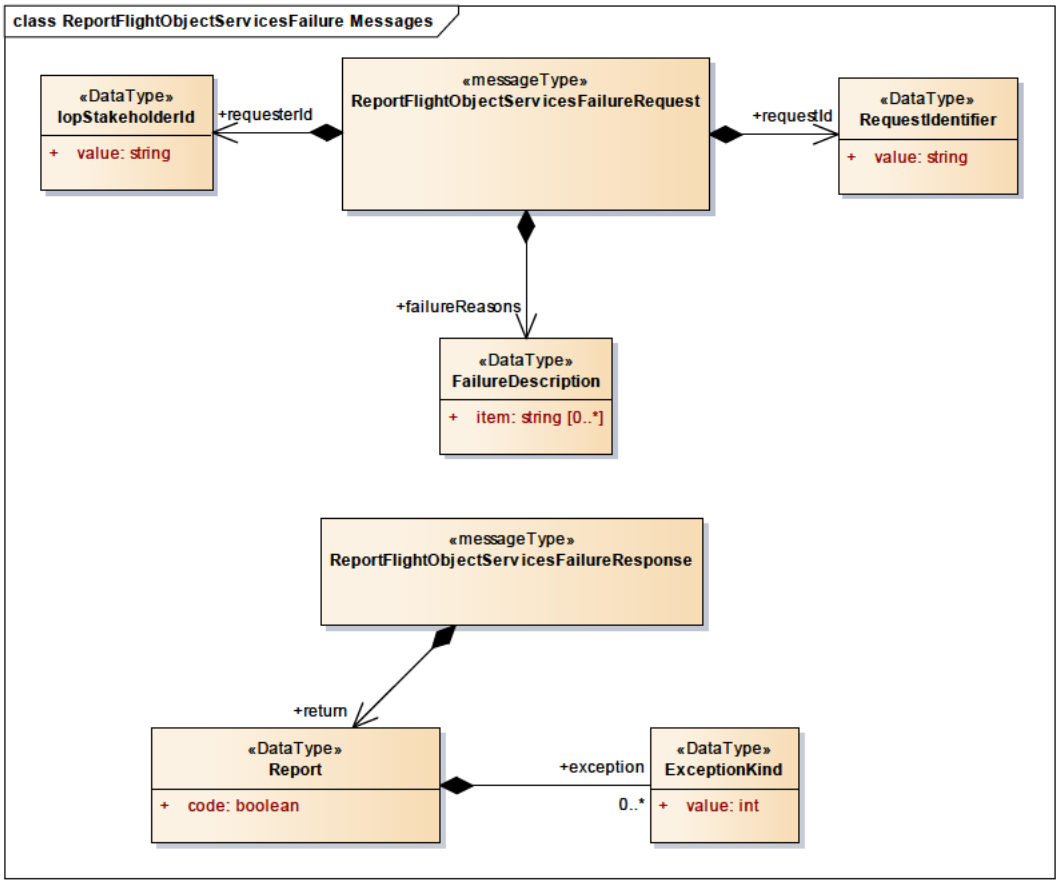


Figure 3-10: FlightObjectManagement Interface ReportFlightObjectServicesFailure Operation Message Types

[REQ]

Identifier	REQ-14.01.04-TS-0901.0900
Requirement	FlightObjectManagement ReportFlightObjectServicesFailure operation signature shall be: + Input Message: - IopStakeholderId - RequestIdentifier - FailureDescription + Output Message: - Report + Fault Message: - : non specified
Title	FlightObjectManagement ReportFlightObjectServicesFailure operation

	signature
Status	<In Progress>
Rationale	FlightObjectManagement ReportFlightObjectServicesFailure operation signature is as follows: - IopStakeholderId, which contains a complex data type representing the stakeholder identifier (defined as string). This parameter is the identifier of the IOP stakeholder reporting failure of services implementation (i.e. FDMP). - RequestIdentifier, which contains a complex data type representing a unique identifier of the Flight Object services request. It uniquely identifies the previously accepted RequestFlightObjectServices request. - FailureDescription, which contains a textual (string) description of the failure occurred while processing previously accepted RequestFlightObjectServices requests (one for each of the FO service which implementation failed). The SWIM-TI does not interpret this information (all possible values/format (e.g. XML) for this parameter are in scope of application layer specification). Output Message: Report, which is a complex data type (defining the report as two main attributes, (i) report code and (ii) report exception) representing error information whether the request is correctly checked and accepted (report code TRUE) or not (report code FALSE). Possible exceptions (report code FALSE) that may be generated by the SWIM-TI layer are described in dedicated requirements. Semantic: This operation allows to the FDMP to report to the application (FDC) the failure of an already accepted Flight Object Service request. The SWIM-TI on the receiving side enforces first the validity checks and, in case no error codes are generated, that layer triggers in synchronous way (blocking) a call to ATCFlightObjectControlApplication interface_ ReportFlightObjectServicesFailure (FDC) operation and responds with application return.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<Yes>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>

<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.2.1.1.3 RejectFlightObject

The message types for the `RejectFlightObject` operation is a description of information exchanged between the service consumer and provider.

The Messages Types are described in the following table, also refer to the SoaML diagram.

Table 3-3: FlightObjectManagement Interface RejectFlightObject Operation Message Types

Message Type	Brief Description
Input	
IopStakeholderId	It contains a complex data type representing the stakeholder identifier (defined as string)
FlightObjectIdentifier	It contains a complex data type representing the unique Flight Object identifier. It allows also to distinguish between a real and a what-if Flight Object (default value for whatIfContextId is the empty string "" that denotes a real FO). At SWIM Technical layer, the IOP wide unique identifier of a real Flight Object is made of a unique universal identifier for the Flight Object and an empty What-If Context Identifier. For a What-If Flight Object, the IOP wide unique identifier is made of the unique universal identifier for the real Flight Object and a unique What-If Context Identifier.
ClusterReleaseId	It is a complex type representing the release of a given Flight Object cluster.
ClusterReleaseIdSequence	It is a sequence of ClusterReleaseId and it represents the Flight Object release
RejectReason	It contains a complex type representing the rejection reason. Values for the rejection reasons are non-negative integers. The possible integer values and related semantic are specified and maintained at application level.
Output	
Report	It contains a complex data type (defining the report as two main attributes, (i) report code and (ii) report exception) representing error information whether the request is correctly checked and accepted (report code TRUE) or not (report code FALSE). Possible exceptions (report code FALSE) that may be generated by the SWIM-TI layer are described in dedicated requirements.

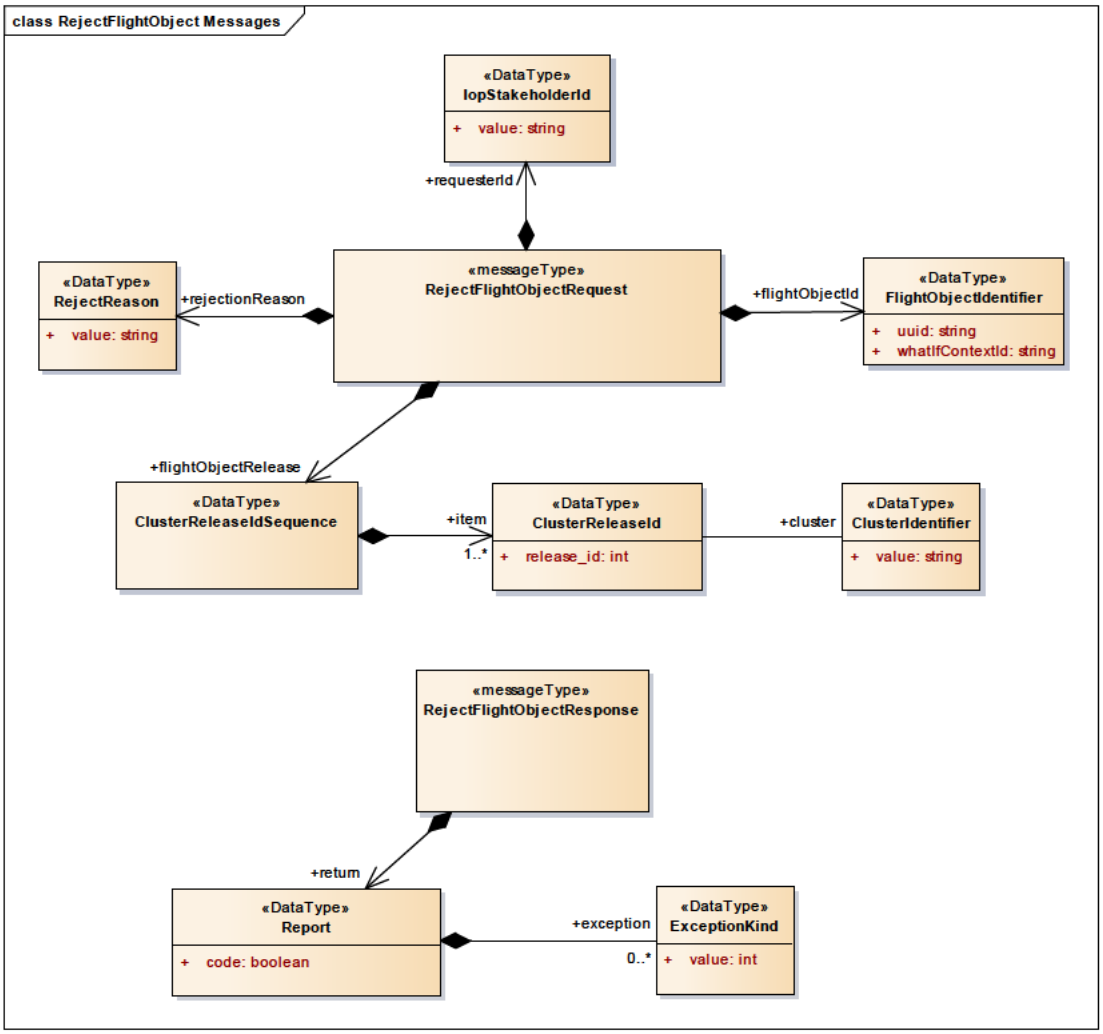


Figure 3-11: FlightObjectManagement Interface RejectFlightObject Operation Message Types

[REQ]

Identifier	REQ-14.01.04-TS-0901.0440
Requirement	FlightObjectManagement RejectFlightObject operation signature shall be: + Input Message: - IopStakeholderId - FlightObjectIdentifier - ClusterReleaseIdSequence - RejectReason + Output Message: - Report + Fault Message: - : non specified
Title	FlightObjectManagement RejectFlightObject operation signature
Status	<Validated>
Rationale	FlightObjectManagement RejectFlightObject operation signature consists of: Input Message: - IopStakeholderId, which contains a complex data type representing the stakeholder identifier (defined as string). It identifies the IOP stakeholder rejecting the Flight Object. - FlightObjectIdentifier, which contains a complex data type representing the

	unique Flight Object identifier. It allows also to distinguish between a real and a what-if Flight Object (default value for whatIfContextId is the empty string "" that denotes a real FO). At SWIM Technical layer, the IOP wide unique identifier of a real Flight Object is made of a unique universal identifier for the Flight Object and an empty What-If Context Identifier. For a What-If Flight Object, the IOP wide unique identifier is made of the unique universal identifier for the real Flight Object and a unique What-If Context Identifier. It identifies the Flight Object to be rejected. - ClusterReleaseIdSequence, which represents the Flight Object release consisting of a sequence of a complex type representing the release of a given Flight Object cluster (cluster name-cluster release pair). It identifies the release of the Flight Object to be rejected. - RejectReason, which contains a string representing the reject reason(s). The SWIM-TI does not interpret this information (all possible values/format (e.g. XML) for this parameter are in scope of application layer specification). Output Message: Report, which is a complex data type (defining the report as two main attributes, (i) report code and (ii) report exception) representing error information whether the request is correctly checked and accepted (report code TRUE) or not (report code FALSE). Possible exceptions (report code FALSE) that may be generated by the SWIM-TI layer are described in dedicated requirements. Semantic: This operation allows to a FDC to request to the FDMP the rejection of a given Flight Object reporting the reason for rejection. The SWIM-TI on the receiving side enforces first the validity checks and, in case no error codes are generated, that layer triggers in synchronous way (blocking) a call to ATCFlightObjectControlApplication interface_ RejectFlightObject (FDMP) operation and responds with application return.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<Yes>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.2.1.1.4 RestoreFlightObject

The message types for the `RestoreFlightObject` operation is a description of information exchanged between the service consumer and provider.

The messages types are described in the following table, also refer to the SoaML diagram.

Table 3-4: FlightObjectManagement Interface RestoreFlightObject Operation Message Types

Message Type	Brief Description
Input	
`IopStakeholderId`	It contains a complex data type representing the stakeholder identifier (defined as string)
`FlightObjectIdentifier`	It contains a complex data type representing the unique Flight Object identifier. It allows also to distinguish between a real and a what-if Flight Object (default value for whatIfContextId is the empty string "" that denotes a real FO). At SWIM Technical layer, the IOP wide unique identifier of a real Flight Object is made of a unique universal identifier for the Flight Object and an empty What-If Context Identifier. For a What-If Flight Object, the IOP wide unique identifier is made of the unique universal identifier for the real Flight Object and a unique What-If Context Identifier.
Output	
`Report`	It contains a complex data type (defining the report as two main attributes, (i) report code and (ii) report exception) representing error information whether the request is correctly checked and accepted (report code TRUE) or not (report code FALSE). Possible exceptions (report code FALSE) that may be generated by the SWIM-TI layer are described in dedicated requirements.

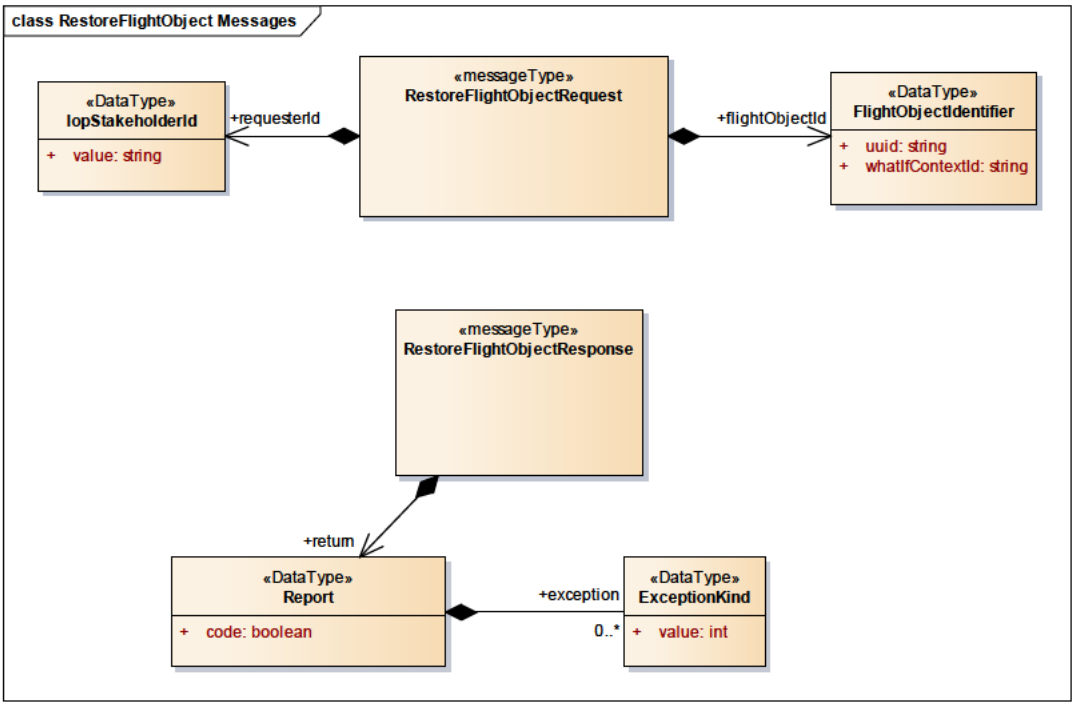


Figure 3-12: FlightObjectManagement RestoreFlightObject Operation Message Types

[REQ]

Identifier	REQ-14.01.04-TS-0901.0445
Requirement	FlightObjectManagement RestoreFlightObject operation signature shall be: + Input Message: - IopStakeholderId - FlightObjectIdentifier + Output Message: - Report + Fault Message: - : non specified
Title	FlightObjectManagement RestoreFlightObject operation signature
Status	<Validated>
Rationale	FlightObjectManagement RestoreFlightObject operation signature consists of: Input Message: - IopStakeholderId, which contains a complex data type representing the stakeholder identifier (defined as string). It identifies the IOP stakeholder requesting the service. - FlightObjectIdentifier, which contains a complex data type representing the unique Flight Object identifier. It allows also to distinguish between a real and a what-if Flight Object (default value for whatIfContextId is the empty string "" that denotes a real FO). At SWIM Technical layer, the IOP wide unique identifier of a real Flight Object is made of a unique universal identifier for the Flight Object and an empty What-If Context Identifier. For a What-If Flight Object, the IOP wide unique identifier is made of the unique universal identifier for the real Flight Object and a unique What-If Context Identifier. It identifies Flight Object to restore. Output Message: Report, which is a complex data type (defining the report as two main

	attributes, (i) report code and (ii) report exception) representing error information whether the request is correctly checked and accepted (report code TRUE) or not (report code FALSE). Possible exceptions (report code FALSE) that may be generated by the SWIM-TI layer are described in dedicated requirements. Semantic: This operation allows to a FDC to request to the FDMP the restoring (re-publish) of a given Flight Object. This could be required when, e.g., an IOP stakeholder in not part of the current distribution list of a given Flight Object. The SWIM-TI on the receiving side (FDMP) enforces first the validity checks and, in case no error codes are generated, that layer re-publishes the full content of the target Flight Object (summary and all clusters).
Category	<Interface>
Validation Method	
Verification Method	<Review of Design>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<Yes>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.2.1.1.5 RequestFlightObjectsRecovery

The message types for the `RequestFlightObjectsRecovery` operation is a description of information exchanged between the service consumer and provider.

The Messages Types are described in the following table, also refer to the SoaML diagram.

Table 3-5: FlightObjectManagement Interface RequestFlightObjectsRecovery Operation Message Types

Message Type	Brief Description
Input	
`IopStakeholderId`	It contains a complex data type representing the stakeholder identifier (defined as string)

FlightObjectIdentifierSequence	It defines a container for sequence of zero or more FlightObjectIdentifier.
FlightObjectIdentifier	It contains a complex data type representing the unique Flight Object identifier. It allows also to distinguish between a real and a what-if Flight Object (default value for whatIfContextId is the empty string "" that denotes a real FO). At SWIM Technical layer, the IOP wide unique identifier of a real Flight Object is made of a unique universal identifier for the Flight Object and an empty What-If Context Identifier. For a What-If Flight Object, the IOP wide unique identifier is made of the unique universal identifier for the real Flight Object and a unique What-If Context Identifier.
Tier	It contains a complex data type representing the recovery Tier (defined as and Integer)
Output	
FlightObjectRecoveryReport	It contains a complex data type which defines a container for: - an array of possible exeptions (one for each Flight Object to be recovered), - the “Report” data.
Report	It contains a complex data type (defining the report as two main attributes, (i) report code and (ii) report exception) representing error information whether the request is correctly checked and accepted (report code TRUE) or not (report code FALSE). Possible exceptions (report code FALSE) that may be generated by the SWIM-TI layer are described in dedicated requirements.

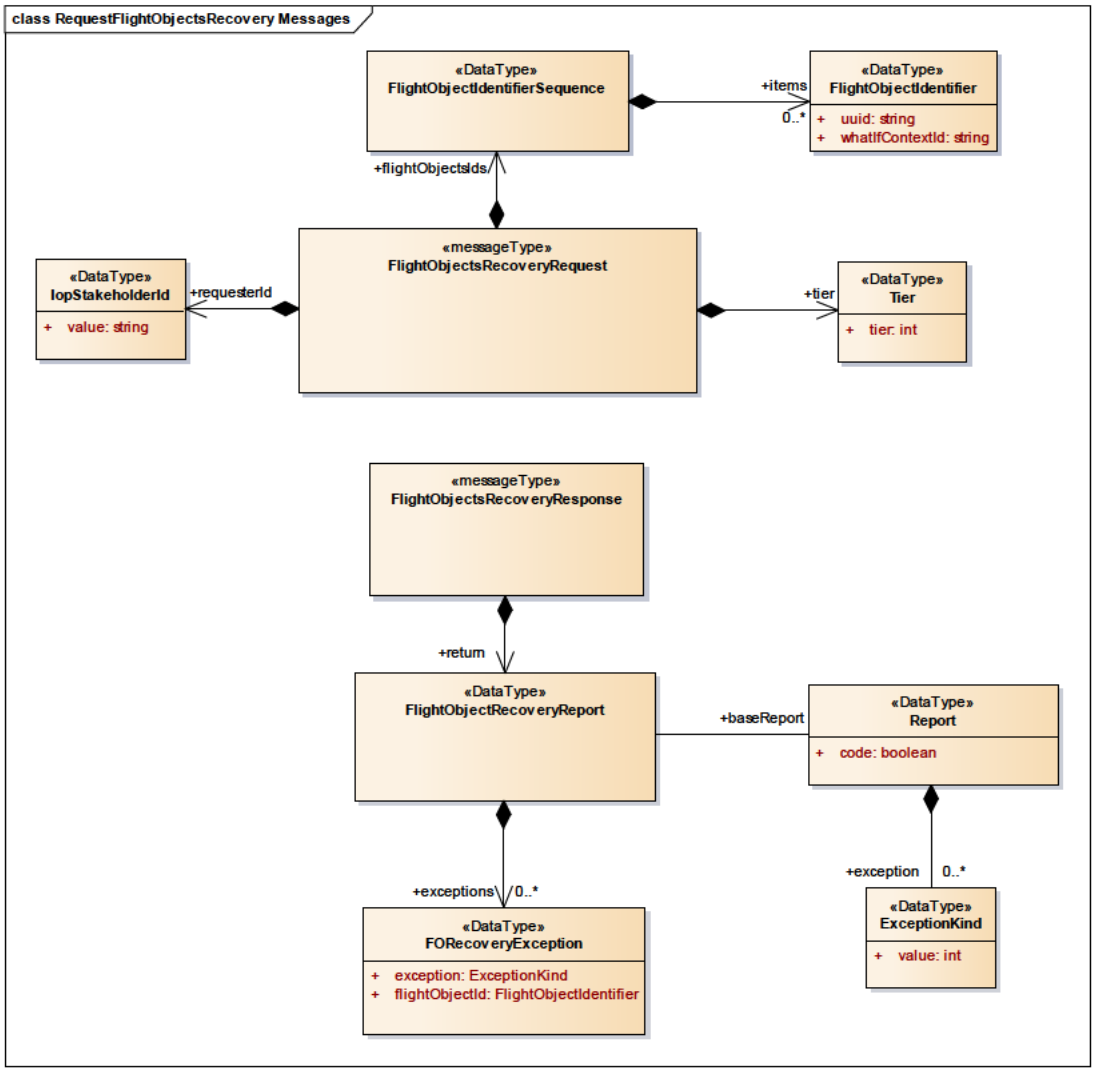


Figure 3-13: FlightObjectManagement RequestFlightObjectsRecovery Operation Message Types

[REQ]

Identifier	REQ-14.01.04-TS-0901.0796
Requirement	FlightObjectManagement RequestFlightObjectsRecovery operation signature shall be: + Input Message: - IopStakeholderId - Tier - FlightObjectIdentifier [0..*] + Output Message: - FlightObjectRecoveryReport + Fault Message: - : unspecified
Title	FlightObjectManagement RequestFlightObjectsRecovery operation signature
Status	<Validated>
Rationale	FlightObjectManagement RequestFlightObjectsRecovery operation signature is:

	Input Message: - IopStakeholderId, which contains a complex data type representing the stakeholder identifier (defined as string) the Recovering SWIM Node is serving. This parameter is the identifier of the IOP stakeholder requesting the Flight Object recovery (the recovering SWIM Node). - Tier, an Integer providing the Recovery Tier of the Recovering SWIM Node. - A sequence of FlightObjectIdentifier, a list of a complex data type representing the unique Flight Object identifier. It allows also to distinguish between a real and a what-if Flight Object (default value for whatIfContextId is the empty string "" that denotes a real FO). At SWIM Technical layer, the IOP wide unique identifier of a real Flight Object is made of a unique universal identifier for the Flight Object and an empty What-If Context Identifier. For a What-If Flight Object, the IOP wide unique identifier is made of the unique universal identifier for the real Flight Object and a unique What-If Context Identifier. This parameter consists of the Flight Object identifiers of the Flight Objects to be recovered. When the attribute is left empty, it is to be understood as the entire list of Flight Objects pertaining to the specified Tier. Output Message: FlightObjectRecoveryReport, which defines a container for - an array of possible exceptions (one for each Flight Object to be recovered), - a complex data type (defining the report as two main attributes, (i) report code and (ii) report exception) representing error information whether the request is correctly checked and accepted (report code TRUE) or not (report code FALSE). Possible exceptions (report code FALSE) that may be generated by the SWIM-TI layer are described in dedicated requirements. Semantic: This operation allows to a FDC to request the recovery of Flight Objects (re-publish). The SWIM-TI on the receiving side (FDMPs) enforces first the validity checks and, in case no error codes are generated, that layer re-publishes the the target (depending on the Tier) Flight Objects (summary and all clusters). This requirement covers the following NIST security controls: CP-7 a, CP-10.
Category	<Interface><Security>
Validation Method	
Verification Method	<Review of Design>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<Yes>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>

<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.2.1.2 Functional Requirements

This section provides specific functional requirements identified for the FlightObjectManagement Interface.

[REQ]

Identifier	REQ-14.01.04-TS-0901.0410
Requirement	For a given FlightObject, the physical endpoint to be consumed by a SWIM Node acting as consumer for FlightObjectManagement interface shall be the one provided by the SWIM Node on the current FDMP side.
Title	FlightObjectManagement provider endpoint
Status	<Validated>
Rationale	Only the SWIM Node fulfilling the role of manager owns the responsibility to accept the request of a service coming from another node involved in the collaborative scenario. The Flight Object operation is executed by the ATC application in order to fulfil the role of FDMP for the target Flight Object; on the same Flight Object, the ATC requestor application fulfils the role of FDC.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0415
Requirement	A SWIM Node acting as consumer for FlightObjectManagement interface shall be able to discover the current FlightObjectManagement provider

	endpoint for the target FlightObject.
Title	FlightObjectManagement provider endpoint discovery
Status	<Validated>
Rationale	Only the SWIM Node fulfilling the role of manager owns the responsibility to accept the request of a service coming from another node involved in the collaborative scenario. The Flight Object operation is executed by the ATC application in order to fulfil the role of FDMP for the target flight; on the same flight, the ATC requestor application fulfils the role of FDC. Requestor SWIM Node (service consumer) has to discover the manager (service provider), in order to forward to it the requested service(s).
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0420
Requirement	FlightObjectManagement provider endpoint for the target FlightObject shall be discovered using the following metadata: - FlightObject Unique identifier. - ATC System Unique Identifier. - Available FlightObjectManagement physical endpoints for each uniquely identified ATM System expected to play the FDMP role.
Title	FlightObjectManagement provider endpoint discovery metadata
Status	<Validated>
Rationale	Only the SWIM Node fulfilling the role of manager owns the responsibility to accept the request of a service coming from another node involved in the collaborative scenario. The Flight Object operation is executed by the ATC application in order to fulfil the role of FDMP for the target flight; on the same flight, the ATC requestor application fulfils the role of FDC. Requestor SWIM Node (service consumer) has to discover the manager (service provider), in order to forward to it the requested service(s). The right physical endpoint is looked up knowing which ATC system is currently owning the target FlightObject and therefore retrieving the corresponding physical endpoint.

Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0425
Requirement	Metadata enabling FlightObjectManagement provider endpoint discovery shall be managed at configuration phase providing for each uniquely identified ATM system its corresponding physical endpoint.
Title	FlightObjectManagement provider endpoint discovery metadata configuration
Status	<Validated>
Rationale	Only the SWIM Node fulfilling the role of manager owns the responsibility to accept the request of a service coming from another node involved in the collaborative scenario. The Flight Object operation is executed by the ATC application in order to fulfil the role of FDMP for the target Flight Object; on the same Flight Object, the ATC requestor application fulfils the role of FDC. Requestor SWIM Node (service consumer) has to discover the manager (service provider), in order to forward to it the requested service(s). The right physical endpoint is looked up knowing which ATC system is currently owning the target FlightObject and therefore retrieving the corresponding physical endpoint. Even if all the needed metadata could be dynamically discovered and exchanged at SWIM-TI layer, currently it is required to manage that information as configuration artefact. This configuration may include two columns: in the first one is provided the ATC System identifier (the key) and in the second column the physical endpoint concerning that specific ATC system.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>

Domain of interest	<ICD><SLA>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0910
Requirement	While serving a FlightObjectManagement RequestFlightObjectServices operation call, the SWIM-TI layer shall enforce the following validity checks: - Target Flight Object Identifier must exist in the IOP area. In case this check is not completed successfully, the “fo_not_found” error code shall be returned. - The SWIM-TI layer processing the request is also serving the application currently managing the Flight Object. In case this check is not completed successfully, the “not_eligible” error code shall be returned. - The provided Flight Object release must be equal to the release stored locally. In case this check is not completed successfully, the “bad_fo_version” error code shall be returned. - The provided Flight Object Services request must be not null and not empty. In case this check is not completed successfully, the “syntax_error” error code shall be returned. - The provided Flight Object Services request identifier must be not null and not empty. In case this check is not completed successfully, the “syntax_error” error code shall be returned. - The provided requester stakeholder identifier must be not null and not empty. In case this check is not completed successfully, the “syntax_error” error code shall be returned. - The provided requester stakeholder identifier must be known to the SWIM-TI. In case this check is not completed successfully, the “semantic_error” error code shall be returned. - The provided Flight Object Cluster identifiers must be valid. In case this check is not completed successfully, the “semantic_error” error code shall be returned. - The provided Flight Object Cluster releases must be valid (within the admissible range). In case this check is not completed successfully, the “semantic error” error code shall be returned.
Title	FlightObjectManagement RequestFlightObjectServices operation validity checks
Status	<In Progress>

Rationale	In addition to ComplexReports that may be returned by the IOP application, the SWIM-TI, as a result of validity checking, may generate locally additional error codes (ComplexReport with report code FALSE).
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0325	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.0920
Requirement	In case errors occur at the SWIM-TI layer while serving a FlightObjectManagement RequestFlightObjectServices operation call, the “middleware_failure” error code shall be returned.
Title	FlightObjectManagement RequestFlightObjectServices operation “middleware_failure” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the application layer and by the SWIM-TI layer (validity checks), the requester is also informed when internal errors occur at the SWIM-TI layer.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A

<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0325	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.0930
Requirement	Once completed successfully the validity checks applicable to the FlightObjectManagement RequestFlightObjectServices operation, the SWIM-TI layer shall forward the request to the application layer (ATCFlightObjectControlApplication interface RequestFlightObjectServices operation) and: - Return back to the requester with the application return, or - In case errors occur at the application layer, the “application_failure” error code shall be returned. - In case the application layer is not reachable, the “application_failure” error code shall be returned. - In case the application layer is not responding within a configurable timeout, the “application failure” error code shall be returned.
Title	FlightObjectManagement RequestFlightObjectServices operation “application_failure” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the application layer and by the SWIM-TI layer (validity checks, internal errors), the requester is also informed when errors occur while interacting with the application layer.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>

<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0325	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.0940
Requirement	While serving a FlightObjectManagement ReportFlightObjectServicesFailure operation call, the SWIM-TI layer shall enforce the following validity checks: - The provided Flight Object Services failure description must be not null and not empty. In case this check is not completed successfully, the "syntax_error" error code shall be returned. - The provided requester stakeholder identifier must be not null and not empty. In case this check is not completed successfully, the "syntax_error" error code shall be returned. - The provided requester stakeholder identifier must be known to the SWIM-TI. In case this check is not completed successfully, the "semantic_error" error code shall be returned.
Title	FlightObjectManagement ReportFlightObjectServicesFailure operation validity checks
Status	<In Progress>
Rationale	In addition to reports that may be returned by the IOP application, the SWIM-TI, as a result of validity checking, may generate locally additional error codes (Report with report code FALSE).
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0325	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.0950
Requirement	In case errors occur at the SWIM-TI layer while serving a

	FlightObjectManagement ReportFlightObjectServicesFailure operation call, the “middleware_failure” error code shall be returned.
Title	FlightObjectManagement ReportFlightObjectServicesFailure operation “middleware_failure” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the application layer and by the SWIM-TI layer (validity checks), the requester is also informed when internal errors occur at the SWIM-TI layer.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0325	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.0960
Requirement	Once completed successfully the validity checks applicable to the FlightObjectManagement ReportFlightObjectServicesFailure operation, the SWIM-TI layer shall forward the request to the application layer (ATCFlightObjectControlApplication interface ReportFlightObjectServicesFailure operation) and: - Return back to the requester with the application return, or - In case errors occur at the application layer, the “application_failure” error code shall be returned. - In case the application layer is not reachable, the “application_failure” error code shall be returned. - In case the application layer is not responding within a configurable timeout, the “application_failure” error code shall be returned.
Title	FlightObjectManagement ReportFlightObjectServicesFailure operation “application_failure” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the application layer and by the SWIM-TI layer (validity checks, internal errors), the requester is also informed when errors occur while interacting with the application layer.

Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0325	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.0970
Requirement	While serving a FlightObjectManagement RejectFlightObject operation call, the SWIM-TI layer shall enforce the following validity checks: - Target Flight Object Identifier must exist in the IOP area. In case this check is not completed successfully, the “fo_not_found” error code shall be returned. - The SWIM-TI layer processing the request is also serving the application currently managing the Flight Object. In case this check is not completed successfully, the “not_eligible” error code shall be returned. - The provided Flight Object release must be equal to the release stored locally. In case this check is not completed successfully, the “bad_fo_version” error code shall be returned. - The provided Flight Object rejection reason must be not null and not empty. In case this check is not completed successfully, the “syntax_error” error code shall be returned. - The provided requester stakeholder identifier must be not null and not empty. In case this check is not completed successfully, the “syntax_error” error code shall be returned. - The provided requester stakeholder identifier must be known to the SWIM-TI. In case this check is not completed successfully, the “semantic_error” error code shall be returned. - The provided Flight Object Cluster identifiers must be valid. In case this check is not completed successfully, the “semantic_error” error code shall be returned. - The provided Flight Object Cluster releases must be valid (within the admissible range). In case this check is not completed successfully, the “semantic_error” error code shall be returned.
Title	FlightObjectManagement RejectFlightObject operation validity checks

Status	<In Progress>
Rationale	In addition to reports that may be returned by the IOP application, the SWIM-TI, as a result of validity checking, may generate locally additional error codes (Report with report code FALSE).
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0325	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.0980
Requirement	In case errors occur at the SWIM-TI layer while serving a FlightObjectManagement RejectFlightObject operation call, the “middleware_failure” error code shall be returned.
Title	FlightObjectManagement RejectFlightObject operation “middleware_failure” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the application layer and by the SWIM-TI layer (validity checks), the requester is also informed when internal errors occur at the SWIM-TI layer.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance

<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0325	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.0990
Requirement	Once completed successfully the validity checks applicable to the FlightObjectManagement RejectFlightObject operation, the SWIM-TI layer shall forward the request to the application layer (ATCFlightObjectControlApplication interface RejectFlightObject operation) and: - Return back to the requester with the application return, or - In case errors occur at the application layer, the “application_failure” error code shall be returned. - In case the application layer is not reachable, the “application_failure” error code shall be returned. - In case the application layer is not responding within a configurable timeout, the “application failure” error code shall be returned.
Title	FlightObjectManagement RejectFlightObject operation “application_failure” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the application layer and by the SWIM-TI layer (validity checks, internal errors), the requester is also informed when errors occur while interacting with the application layer.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>

<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0325	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.1000
Requirement	While serving a FlightObjectManagement RestoreFlightObject operation call, the SWIM-TI layer shall enforce the following validity checks: - Target Flight Object Identifier must exist in the IOP area. In case this check is not completed successfully, the “fo_not_found” error code shall be returned. - The SWIM-TI layer processing the request is also serving the application currently managing the Flight Object. In case this check is not completed successfully, the “not_eligible” error code shall be returned. - The provided requester stakeholder identifier must be not null and not empty. In case this check is not completed successfully, the “syntax_error” error code shall be returned. - The provided requester stakeholder identifier must be known to the SWIM-TI. In case this check is not completed successfully, the “semantic_error” error code shall be returned.
Title	FlightObjectManagement RestoreFlightObject operation validity checks
Status	<In Progress>
Rationale	The SWIM-TI, as a result of validity checking, may generate locally error codes (Report with report code FALSE).
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0325	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.1010

Requirement	Once completed successfully the validity checks applicable to the FlightObjectManagement RestoreFlightObject operation, the SWIM-TI layer shall re-publishes the full content of the target Flight Object (summary and all clusters) or, in case errors occur at the SWIM-TI layer, “middleware failure” error code shall be returned.
Title	FlightObjectManagement RestoreFlightObject operation “middleware_failure” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the SWIM-TI layer during validity checks, the requester is also informed when internal errors occur at the SWIM-TI layer.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0325	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.1020
Requirement	While serving a FlightObjectManagement RequestFlightObjectsRecovery operation call, the SWIM-TI layer shall enforce the following validity checks: - The provided requester stakeholder identifier must be not null, not empty and well formed syntax. In case this check is not completed successfully, the “syntax_error” error code shall be returned. - The provided Tiers must be valid (within the admissible range). In case this check is not successfully completed, the “semantic_error” error code shall be returned. - The local SWIM-TI layer must be IOP Enabeld. In case this check is not successfully completed, the “isolated_stakeholder” error code shall be returned. - (when provided, for each of the missing Flight Objects to recover) Target Flight Object Identifier must exist in the IOP area. In case this check is not completed successfully, the Flight Object Indentifier and “fo_not_found” error code shall be added to the recovery exception list. - (when provided, for each of the missing Flight Objects to recover) The

	requester must not be the current manager of the Flight Object. In case this check is not completed successfully, the Flight Object Indentifier and “not_eligible” error code shall be added to the recovery exception list. - (when provided, for each of the missing Flight Objects to recover) The Flight Object must belong to the provided tier. In case this check is not completed successfully, the Flight Object Indentifier and “not_eligible” error code shall be added to the recovery exception list. - (when provided, for each of the missing Flight Objects to recover) The requester must be in the distribution list of the Flight Object. In case this check is not completed successfully, the Flight Object Indentifier and “not_eligible” error code shall be added to the recovery exception list. - (when provided, for each of the missing Flight Objects to recover) The manager of the Flight Object must be IOP Enabeld. In case this check is not completed successfully, the Flight Object Indentifier and “isolated_stakeholder” error code shall be added to the recovery exception list.
Title	FlightObjectManagement RequestFlightObjectsRecovery operation validity checks
Status	<In Progress>
Rationale	The SWIM-TI, as a result of validity checking, may generate locally error codes (Report with report code FALSE).
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0325	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.1030
Requirement	Once completed successfully the validity checks applicable to the FlightObjectManagement RequestFlightObjectsRecovery operation, the SWIM-TI layer shall publish the the target (depending on the Tier) Flight Objects (summary and all clusters) or: - In case the SWIM-TI layer is not IOP Enabled, the “isolated_stakeholder” error code shall be returned.

	- In case the application layer is not IOP Enabled, the “isolated_stakeholder” error code shall be returned.
Title	FlightObjectManagement RequestFlightObjectsRecovery operation “isolated_stakeholder” error code
Status	<In Progress>
Rationale	To make sure that in addition to error codes generated by the application layer and by the SWIM-TI layer (validity checks, internal errors), the requester is also informed when errors occur while interacting with the application layer.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0325	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.0815
Requirement	The SWIM Node acting as FDMP and accepting a RequestFlightObjectsRecovery request should publish all locally-managed Flight Objects for which the requesting node is in requested Tier within a maximum time of ‘SP-IOP-FOsRePublication_Recovery_Time’.
Title	Tiered Recovery of Flight Objects: re-publication maximum time
Status	<Validated>
Rationale	In order to allow Flight Objects to be recovered sequentially and mitigate a storm of updates, a Tier approach is used to recover Flight Objects. Flight Objects are received by the recovering SWIM Node according to a defined priority given by the Tier information. This recomendation aims at limiting as much as possible the time by when, the receiving SWIM Node at FDMP side, has to republish requested Flight Objects. Note that the Node might be in a situation that it prefers to allocate resources to its normal operation rather than to the recovering activities. This may cause missing timing restriction. In that case it is the application layer on the receiver side responsible to handle the event (timeout expired). When only a set of missing Flight Objects is requested to recover, the FDMP SWIM Node will only address the requested Flight Objects. If the set

	is empty then it will have to address all locally-managed Flight Objects for which the requesting node is in Tier 'recovery_tier'.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Subscription handler><Publisher><Publication consumer><Publication mediator><Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.2.1.3 Performance Requirements

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.2.1.4 Security and Integrity Requirements

This section provides specific security and integrity requirements identified for the FlightObjectManagement Interface.

Currently the information exchanged through this interface is protected and the participants are authorized and authenticated adopting transport level security solutions.

[REQ]

Identifier	REQ-14.01.04-TS-0901.0430
Requirement	Transport Level Security shall be applied for the FlightObjectManagement interface endpoint.
Title	Transport Level Security for FlightObjectManagement provider endpoint
Status	<Validated>
Rationale	Transport level security applied to protect data exchanged through this interface and to properly authenticate and authorize interfacing entities. This requirement covers NIST security controls IA-2 and IA-8.
Category	<Interface><Security>
Validation Method	

Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.2.1.5 Data Transfer

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.2.1.6 Transactions

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.2.1.7 Interface Instantiation

In this section the SOAP Web-Service specific model or instantiation of the FlightObjectManagement interface is provided. The binding and the contract have been defined according to requirements, UML model and interface descriptions are provided in the previous sections. In particular the binding includes HTTPS to enable transport level security mechanisms.

[REQ]

Identifier	REQ-14.01.04-TS-0901.0325
Requirement	FlightObjectManagement interface shall be instantiated using the following binding: + Protocol stack: - SOAP 1.1 over HTTPS POST over TCP + MEPs: - SRR-MEP

	+ Fault handling: - the service shall be able to determine the content of the HTTP status code and HTTP reason phrase + Encoding: - Text encoding + Security: - Confidentiality: transport - Integrity: transport - Authenticity: transport mutual - Authorization: transport - Non-repudiation: none + Contract: - formalism of contract description: WSDL 1.1 - minimum: not applicable - reference: Blue Profile Technical Specification, ISRM + Interoperability: WS-I Basic Profile 1.2
Title	FlightObjectManagement Interface binding
Status	<Validated>
Rationale	Flight Object Management requires a specific Interface configuration. Security controls are all at transport (HTTP over TLS) level. Authenticity (or Authentication) at transport level has not to be confused with HTTP Basic and Digest Access Authentication that are not supported by this binding.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Service binding>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0101	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0115	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0116	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0124	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0125	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0129	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0010	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0114	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0275	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0123	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0111	N/A
<SATISFIES>	<ATMS Requirement>	REQ-14.01.04-TS-0001.0350	<Full>
<SATISFIES>	<ATMS Requirement>	REQ-14.01.04-TS-0001.0302	<Full>
<SATISFIES>	<ATMS Requirement>	REQ-14.01.04-TS-0001.0303	<Full>
<SATISFIES>	<ATMS Requirement>	REQ-14.01.04-TS-0001.0304	<Full>

<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0006.0001	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0006.0010	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0845	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0002.0600	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0002.0630	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0002.0031	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0121	N/A
<SATISFIES>	<ATMS Requirement>	REQ-14.01.04-TS-0001.0302	<Full>
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

Latest version of WSDL for the FlightObjectManagement interface contract is provided here after (using WSDL standard version 1.1).

The following table presents a mapping between ExceptionKind values used to communicate with the application and the numerical values used between SWIM-TI nodes instances.

ExceptionKind value	**Value in the WSDL for the FlightObjectManagement ("ExceptionKind" data type)**
duplicated_fo	0
fo_not_found	1
syntax_error	2
not_eligible	3
semantic_error	4
old_fo_version	5
isolated_stakeholder	6
timeout	7
application_failure	8
middleware_failure	9
critical_error	10

communication_failure	11
fo_version_collision	12
bad_fo_version	13

Furthermore here below also the mapping of QoS values is provided.

QoS value	**Value in the WSDL for the FlightObjectManagement (“QoS” data type)**
d_1	0
d_2	1
d_3	2

[REQ]

Identifier	REQ-14.01.04-TS-0901.0450
Requirement	The FlightObjectManagement interface shall be instantiated according to the WSDL contract available in the latest 14.01.04 Blue Profile Technical specification.
Title	FlightObjectManagement Interface Binging Contract
Status	<Validated>
Rationale	The WSDL is available in the 14.01.04 Technical Specification.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0325	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.2.2 *FlightObjectDistribution* Interface Requirements

This interface allows the FDMP ATC application to logically distribute Flight Object data to the other relevant ATC application. This logical communication between ATC applications is implemented via the mediation of the SWIM Nodes (refer to Figure 3-3) acting as Flight Object data publishers (on FDMP ATC application side) and/or subscribers (on FDC/FDU ATC Application side). The technology specific instantiation of the FlightObjectDistribution interface (§3.3.9.2.2.7) consists of a set of well-defined OMG DDS (Data Distribution Service) topics, data types and QoS. The data samples of each topic are distributed across SWIM Nodes using the OMG DDS capabilities and according to defined QoS.

[REQ]

Identifier	REQ-14.01.04-TS-0901.0455
Requirement	The FlightObjectDistribution Interface contract shall be specified according to the PSPULL-MEP and PSPUSH-MEP.
Title	FlightObjectDistribution Interface contract Message Exchange Patterns
Status	<Validated>
Rationale	To ensure that the FlightObjectDistribution Interface contract is specified according to the expected Message Exchange Patterns identified for that interface.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Analysis>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Subscription handler><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<Yes>
Testability	<Applicable but not testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

In the following sections the requirements concerning this interface are provided.

3.3.9.2.2.1 Operations

The FlightObjectDistribution interface provides all the typical operations needed to enable entities (publishers and subscribers) to exchange information according to the PSPUSH-MEP and PSPULL-MEP message exchanges patterns. Furthermore, this interface provides also mechanisms to inform interested entities (subscribers) that a specific information instance has been deleted.

This paragraph provides requirements about message structure, type and size concerning the FlightObjectDistribution interface.

The message types describe here below represent the message exchanged between entities interaction through the FlightObjectDistribution interface. These message types are then used to specify OMG DDS data types concerning OMG DDS topics identified for this interface (refer to §3.3.9.2.2.7). The messages types (as many events to be transferred on this interface) are described in the table and SoaML diagram here below.

Table 3-6: FlightObjectDistribution Interface Message Types

Message Type	Message Type Description
`FOCluster`	It acts as a container of complex data and it is used as a single unit of information transferred between involved participants. Its main objective consists in transferring clustered data representing FlightObject data information. Any updated cluster is identified by the attribute 'cluster_id' and its value is stored in the ClusterPayload attribute. FOCluster message type is detailed in Figure 3-15.
`FOSummary`	It acts as a container of complex data and it is used as a single unit of information transferred between involved participants. Its main objective consists in transferring data representing the description of a specific flight defined by its flight key. FOSummary message type is detailed in Figure 3-14.
`IOPStatus`	It acts as a container of complex data and it is used as a single unit of information transferred between involved participants. Its main objective consists in transferring data representing the status of both ATC application and SWIM Node. IOPStatus message type is detailed in Figure 3-16.
`IOPRecoveryStatus`	It acts as a container of complex data and is used as a single unit of information transferred between involved participants. Its main objective consists in transferring data representing the status of the recovering process and the current level(s) of recovering. IOPRecoveryStatus message type is detailed in Figure 3-16.

Table 3-7: FlightObjectDistribution Interface Message Types data elements

Message Components	Description
`DistributionClusterKind`	It defines an enumerative data type describing all the possible types of transferred clusters
`DistributionList`	Is a collection of the information related to the rules of distribution defined for the flight and in particular it defines the list of the stakeholders to which the Flight Object has to be distributed. For each stakeholder the following information are included: - IopStakeholderId, - an integer identifying the Recovery Tier associated to the stakeholder for that specific Flight Object.
`FlightObjectIdentifier`	It represents the unique Flight Object identifier. It allows also to distinguish between a real and a what-if Flight Object (default value for whatIfContextId is the empty string "" that denotes a real FO). At SWIM Technical layer, the IOP wide unique identifier of a real Flight

	Object is made of a unique universal identifier for the Flight Object and an empty What-If Context Identifier. For a What-If Flight Object, the IOP wide unique identifier is made of the unique universal identifier for the real Flight Object and a unique What-If Context Identifier.
ClusterPayload	It is a complex data type representing the unparsed XML content of the cluster which is transferred between the involved participants. It is composed by: - sequence of bytes (RawData data type) containing the compressed or uncompressed XML data. - MIME type (MimeType data type) representing the registered IANA MIME Type name identifying the content type of the sequence of bytes (RawData). For RawData containing uncompressed XML data, the MIME Type is "application/xml" or "text/xml" (RFC7303). For RawData containing compressed XML data, the MIME Type is the IANA registered name assigned to the adopted compression algorithm (e.g. "application/gzip" when GZIP is used - RFC6713).
MimeType	It defines a data type representing an IANA registered MIME Type.
RawData	It defines a data type representing a sequence of bytes.
EnabledStatus	It defines a boolean data type representing the status as with the following options: FALSE: DISABLED TRUE: ENABLED.
RecoveryStatus	It defines a boolean data type representing a recovery status as with the following options: TRUE: recovering in progress. FALSE: recovery is completed.
Time	It defines a data type representing a time instant (using a specific representation)
ReleaseId	It defines a data type identifying a release version of a distributed object
FlightKey	It defines a structured representation of the operational key uniquely identifying a Flight Object. It is composed by: -adep: A string (4 characters) specifying the departure airport of the flight. -ades: A string (4 characters) specifying the arrival airport of the flight. -arcid: A string (7 characters) specifying the identification of the aircraft (registration marking or ICAO designator followed by flight identifier). -eobd: A Date specifying the date of the flight. -eobt: A string (4 characters) specifying the hours and minutes of the estimated of block time. -what-if_context: A string (20 characters) that logically groups a series of FOs and WIFOs by their what-if context. This attribute is relevant and has to be used only when referring to a WIFO.
IopStakeholderId	It defines a data type representing the unique identifier of an ATC application.
ClusterReleaseIdSequence	It defines a complex data type representing a sequence of data identifying the particular Flight Object release: the composition is defined as a sequence of release identification of each cluster
Tier	It defines a complex data type representing the Recovery Tier.

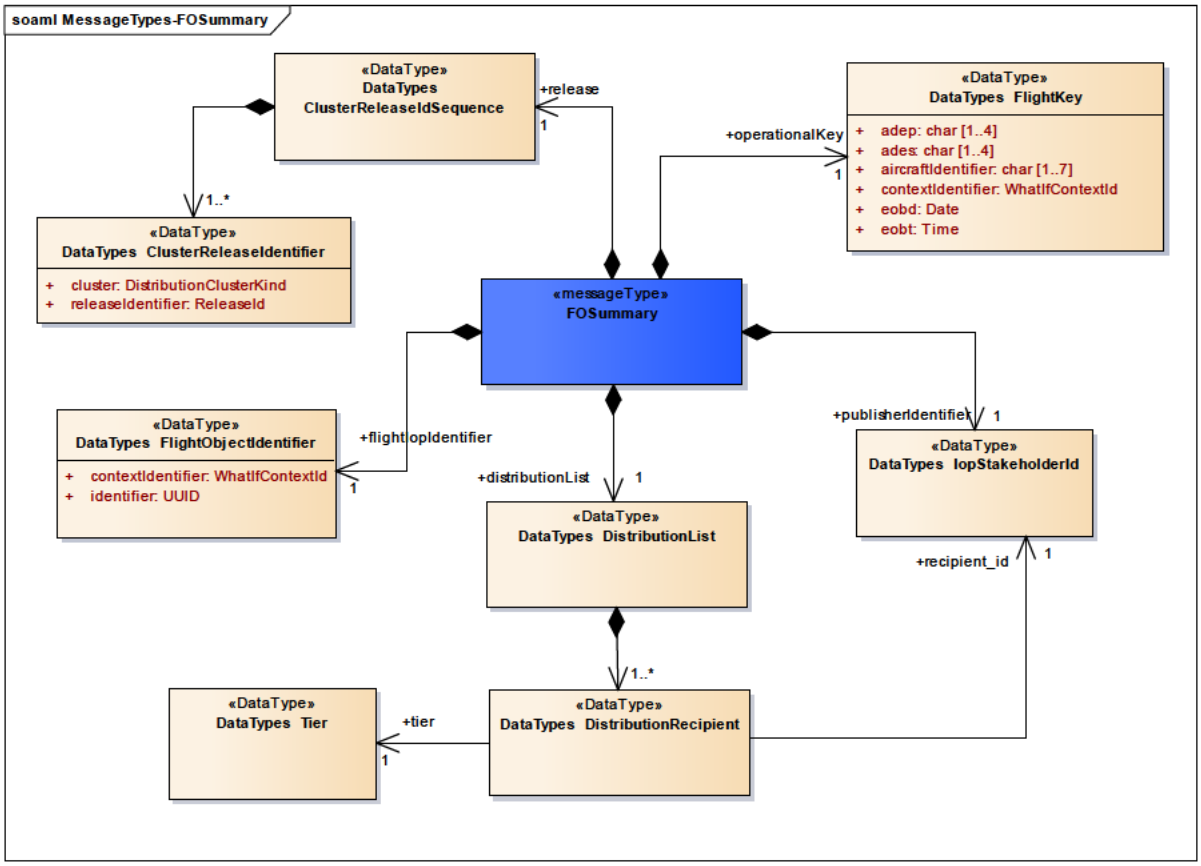


Figure 3-14: FlightObjectDistribution Interface FOSummary Message Type

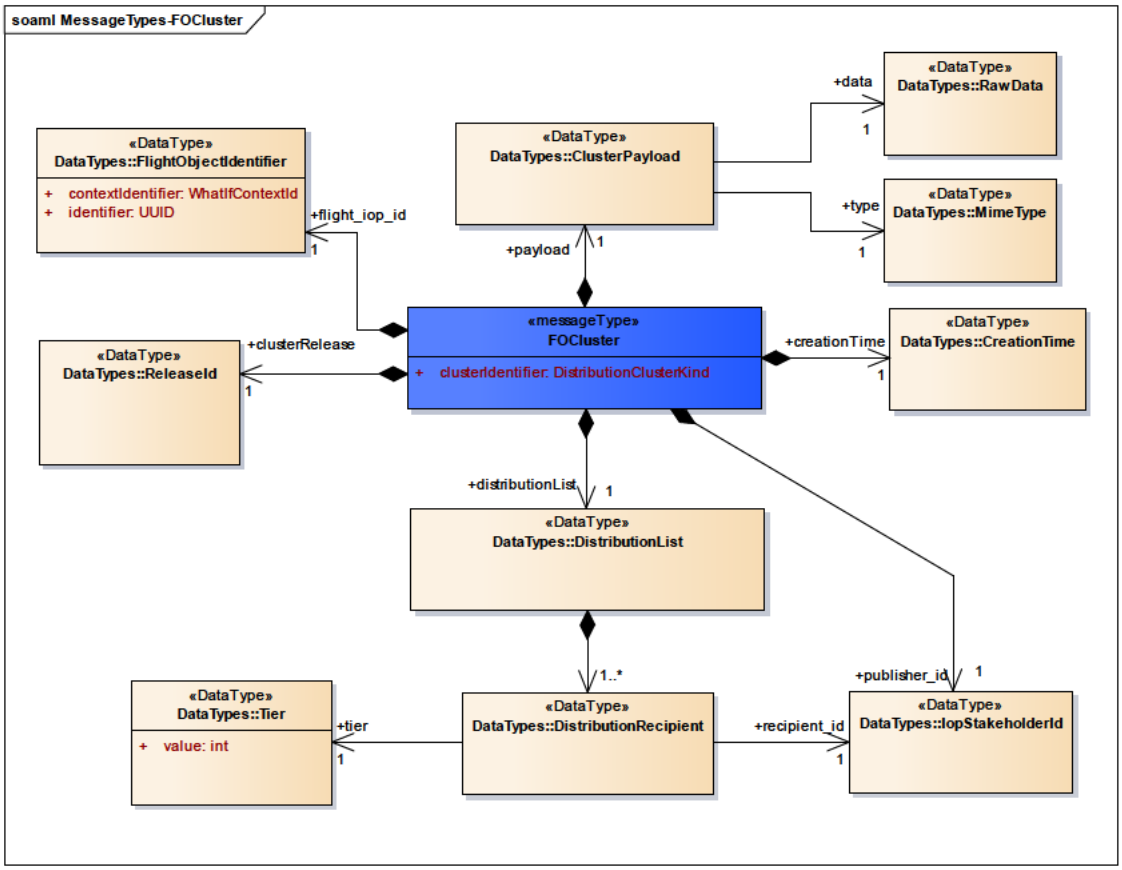


Figure 3-15: FlightObjectDistribution Interface FOCluster Message Type

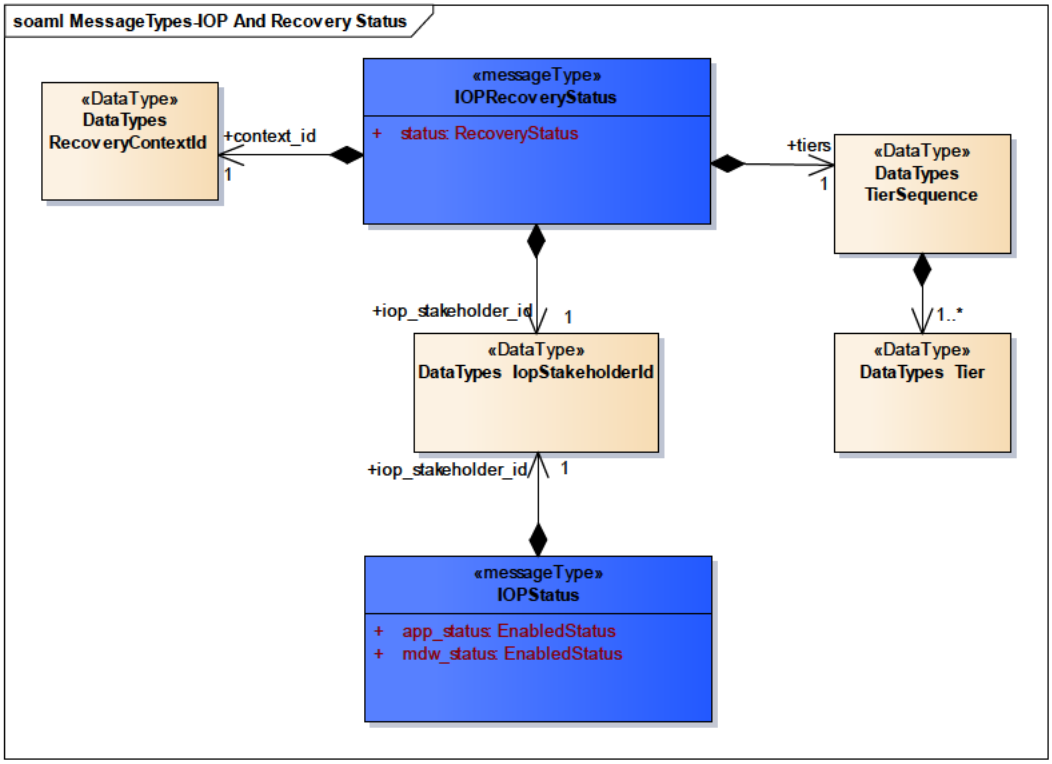


Figure 3-16: FlightObjectDistribution Interface IOPStatus and IOP Recovery Status Message Typs

[REQ]

Identifier	REQ-14.01.04-TS-0901.0460
Requirement	The FOSummary message type shall contain the following information: - FlightObjectIdentifier. - DistributionList. - ClusterReleaseIdSequence. - IopStakeholderId. - FlightKey.
Title	FlightObjectDistribution Interface FOSummary Message Type Structure
Status	<Validated>
Rationale	FOSummary message type structure consists of: - FlightObjectIdentifier, which is the unique Flight Object identifier. It allows also to distinguish between a real and a what-if Flight Object (default value for whatIfContextId is the empty string "" that denotes a real FO). At SWIM Technical layer, the IOP wide unique identifier of a real Flight Object is made of a unique universal identifier for the Flight Object and an empty What-If Context Identifier. For a What-If Flight Object, the IOP wide unique identifier is made of the unique universal identifier for the real Flight Object and a unique What-If Context Identifier. - DistributionList, which is the list of unique identifiers of ATC applications interested to that FlightObject and the associated Tier of the Flight Object for that ATC application. - ClusterReleaseIdSequence, which contains the current releases of each cluster - IopStakeholderId, which is the unique identifier of the ATC application currently managing that FlightObject. - FlightKey, which includes key information such departure airport, arrival airport, off block time, etc. This requirement covers the following NIST security controls: CP-7 a, CP-10.
Category	<Interface><Security>
Validation Method	
Verification Method	<Review of Design>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0465
Requirement	The FOCluster message type shall contain the following information: - FlightObjectIdentifier. - DistributionClusterKind. - ClusterRelease. - ClusterPayload. - DistributionList. - CreationTime. - IopStakeholderId.
Title	FlightObjectDistribution Interface FOCluster Message Type Structure
Status	<Validated>
Rationale	FOCluster message type structure consists of: - FlightObjectIdentifier, which is the unique Flight Object identifier. It allows also to distinguish between a real and a what-if Flight Object (default value for whatIfContextId is the empty string "" that denotes a real FO). At SWIM Technical layer, the IOP wide unique identifier of a real Flight Object is made of a unique universal identifier for the Flight Object and an empty What-If Context Identifier. For a What-If Flight Object, the IOP wide unique identifier is made of the unique universal identifier for the real Flight Object and a unique What-If Context Identifier. - DistributionClusterKind, which is the name/kind of the cluster - ClusterRelease, which is the release of the cluster - ClusterPayload, which encapsulates the cluster content. - DistributionList, which is the list of unique identifiers of ATC applications interested to that FlightObject. - CreationTime, which is the creation time of the FlightObject. - IopStakeholderId, IopStakeholderId, which is the unique identifier of the ATC application publishing that FlightObject. The ClusterPayload data type represents the unparsed XML content of the cluster which is transferred between the involved participants. It is composed by: - sequence of bytes (RawData data type) containing the compressed or uncompressed XML data. - MIME type (MimeType data type) representing the registered IANA MIME Type name identifying the content type of the sequence of bytes (RawData). For RawData containing uncompressed XML data, the MIME Type is “application/xml” or “text/xml” (RFC7303). For RawData containing compressed XML data, the MIME Type is the IANA registered name assigned to the adopted compression algorithm (e.g. “application/gzip” when GZIP is used - RFC6713).
Category	<Interface>
Validation Method	
Verification Method	<Review of Design>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A

<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0472
Requirement	The IOPStatus message type shall contain the following information: - IopStakeholderId. - SWIM Node status. - ATC application status.
Title	FlightObjectDistribution Interface IOPStatus Message Type Structure
Status	<Validated>
Rationale	IOPStatus message type structure consists of: - IopStakeholderId, which is the unique identifier of the ATC application sharing that status - SWIM Node status, which is the status of the SWIM Node serving that ATC application - ATC application status, which is the status of that ATC application. This requirement covers the following NIST security controls: CP-7 a, CP-10.
Category	<Interface><Security>
Validation Method	
Verification Method	<Review of Design>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0473

Requirement	The IOPRecoveryStatus message type shall contain the following information: - IopStakeholderId. - SWIM Node recovery status. - Recovery ContextId. - Tier sequence.
Title	FlightObjectDistribution Interface IOPRecoveryStatus Message Type Structure
Status	<Validated>
Rationale	IOPRecoveryStatus message type structure consists of: - IopStakeholderId, which is the unique identifier of the ATC application sharing that status - SWIM Node recovery status which indicates that the SWIM node is currently performing a recovery or that the recovery process is completed. - Tier sequence, which specifies the current recovery Tier(s) (the tiers are provided as ordered sequence starting from highest priority tier) of that SWIM Node when the SWIM Node recovery status is true. This attribute only needs to be fulfilled when the SWIM Node is performing a recovery. This requirement covers the following NIST security controls: CP-7 a, CP-10.
Category	<Interface><Security>
Validation Method	
Verification Method	<Review of Design>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0475
Requirement	FOSummary shall be published every 60 seconds.
Title	FlightObjectDistribution Interface FOSummary publication frequency
Status	<Validated>
Rationale	FOSummary is periodically published (even if the FlightObject has not been updated) to provide all the involved entities with latest key information concerning the FlightObject. This requirement applies to FO Summaries for both real and what-if (WI)

	Flight Objects. For WIFO the distribution of the summaries is limited only to the specific Flight Object distribution list (see REQ-14.01.04-TS-0901.0370 and REQ-14.01.04-TS-0901.0375).
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><SLA>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.0370
Requirement	FOSummary for real Flight Objects shall be published to all the partitions of the IOP Area Stakeholders.
Title	FlightObjectDistribution Interface - Expected recipients of FOSummary publication for real Flight Objects
Status	<Validated>
Rationale	FOSummary is periodically published (even if the FlightObject has not been updated) to provide all the involved entities with latest key information concerning the FlightObject. For real Flight Objects all the IOP Area stakeholders have to receive this information. For further details refer to Flight Object Summary Distribution QoS, Topic name and structure table.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><SLA>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance

<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.0375
Requirement	FOSummary for what-if Flight Objects shall be published to all the partitions of the IOP Area Stakeholders part of the Flight Object distribution list.
Title	FlightObjectDistribution Interface - Expected recipients of FOSummary publication for what-if Flight Objects
Status	<Validated>
Rationale	FOSummary is periodically published (even if the FlightObject has not been updated) to provide all the involved entities with latest key information concerning the FlightObject. For what-if Flight Objects ONLY the IOP Area stakeholders part of the distribution list have to receive this information.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><SLA>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.0480
Requirement	IOPStatus shall be published every 30 seconds.

Title	FlightObjectDistribution Interface IOPStatus publication frequency
Status	<Validated>
Rationale	IOPStatus information is periodically distributed by all participating entities to share their status with all the others.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><SLA>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.0490
Requirement	The FOCluster message type ClusterPayload shall be compressed using GZIP.
Title	FlightObjectDistribution Interface FOCluster ClusterPayload attribute compression.
Status	<Deleted>
Rationale	To reduce ClusterPayload size. Taking into account that ClusterPayload is a String, its content will be Base64 String built from bytes arrays as resulting from the GZIP compression. It is a local decision to always apply compression or only if the FO Cluster data exceeds a given threshold. The receiver can check if the FO Cluster being received is Base64 encoded (compressed data) or not. There could be additional mechanisms (e.g. DDS User Data QoS) to enrich published data with metadata such as compression applied or not. Currently, the mechanism has not been specified. Furthermore, a detailed analysis and testing it is recommended in order to evaluate the possibility to use as data type for the FO Cluster payload (see IDL) a sequence of bytes instead of a string. Deletion reason: replaced by REQ-14.01.04-TS-0901.1040, REQ-14.01.04-TS-0901.1050 and REQ-14.01.04-TS-0901.1060.
Category	<Interface>
Validation Method	

Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1040
Requirement	When a Flight Object Cluster XML payload is compressed, the ClusterPayload MimeType attribute shall indicate the IANA registered MIME Type name of the adopted compression algorithm.
Title	FlightObjectDistribution Interface FOCluster message type: compressed XML payload.
Status	<In Progress>
Rationale	This requirement allows consumers to discover a) if the Flight Object cluster being received is compressed or not and b) in case it is compressed, the compression algorithm used.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>

<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.1050
Requirement	When a Flight Object Cluster XML payload is not compressed, the ClusterPayload MimeType attribute shall be “application/xml” or “text/xml”.
Title	FlightObjectDistribution Interface FOCluster message type: uncompressed XML payload (RFC7303).
Status	<In Progress>
Rationale	This requirement allows consumers to discover in a standard based way when a Flight Object cluster being received is not compressed.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.1060
Requirement	Compression algorithms used for Flight Object Cluster XML payload compression shall be: - GZIP [RFC6713]. - Additional algorithms documented and agreed as part of the offline-defined IOP partner’s Letter of Agreement.
Title	FlightObjectDistribution Interface FOCluster message type: compression algorithms.
Status	<In Progress>
Rationale	This requirement allows consumers to discover in a standard based way the

	compression algorithm used to compress the Flight Object being received. The minimum interoperability requirement is to support GZIP. Additional compression algorithms (and related MIME types) maybe agreed by the stakeholders and added to the offline-defined IOP partner's Letter of Agreement.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.0495
Requirement	The FOCluster message type CreationTime shall be expressed in milliseconds since the epoch time 00.00 hours, January, 1st, 1970 UTC.
Title	FlightObjectDistribution Interface FOCluster CreationTime attribute format.
Status	<Validated>
Rationale	To ensure interoperability.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A

<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A

3.3.9.2.2.2 Functional Requirements

[REQ]

Identifier	REQ-14.01.04-TS-0901.0826
Requirement	When requested by the IOP Application to start recovery for a given Tier(s) or when the automatic recovery is provided (optional capability) and local triggering conditions apply, the recovering SWIM Node shall publish a RECOVERY_STATUS item at least every 30 seconds with: - the Recovery Status set to TRUE, - a new Recovery Contex ID generated for the SWIM Node, - the Tier(s) set as provided by the IOP Application or specified in the local automatic recovery policy, and store the Recovery context Id for the current recovery iteration.
Title	Tiered Recovery of Flight Objects: Recovery Process initiation and retries (recovering node)
Status	<Validated>
Rationale	Upon reception a request from the IOP Application or automatically triggered, the IOP Recovery Status information is periodically distributed by any recovering entities to trigger relevant FOs re-publication for a given Tier(s). This requirement covers both the case of the first iteration of the recovery process as well as any subsequent re-try. Triggered periodic RECOVERY_STATUS publication, is interrupted according to the conditions in REQ-14.01.04-TS-0901.0792. If the Recoverer (node publishing FOs in response of processing of received RECOVERY_STATUS) receives a RECOVERY_STATUS with status FALSE and context id ‘ctxid’, the recoverer shall instantly stop any further publications belonging to ‘ctxid’. and any internal resources assoziated with ‘ctxid’ can be released. It is a local decision to either stop the publication of the RECOVERY_STATUS item or to continue to publish it with the status set to FALSE. This requirement covers the following NIST security controls: CP-7 a, CP-10.
Category	<Interface><Security>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Subscription handler><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>

Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0777
Requirement	A SWIM Node with IOP Status “ENABLED” receiving a RECOVERY_STATUS from a SWIM Node with a Recovery Status set to “TRUE” for a specified Tier(s) and containing a Recovery Context ID not locally stored for the recovering SWIM Node, shall: - publish all locally-managed Flight Objects for which the recovering Node (‘iop_stakeholder_id’) is in Tier(s) ‘TierSequence’, and - store the Recovery Context ID for the recovering SWIM Node.
Title	Tiered Recovery of Flight Objects: Initial Detection of a Recovering Node
Status	<Validated>
Rationale	This requirement triggers the publication of the relevant Flight Objects by a SWIM Node when it detects that a remote SWIM Node is entering a specific Recovering tier(s). The check on the Recovery Context ID allows to trigger the publication of the Flight Objects only once (see next requirement). This requirement covers the following NIST security controls: CP-7 a, CP-10.
Category	<Interface><Security>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Subscription handler><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A

<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0778
Requirement	Upon reception of a RECOVERY_STATUS from a SWIM Node with a Recovery Status set to “TRUE” and containing an already locally stored Recovery Context ID for the recovering SWIM Node, the receiving SWIM Node shall ignore the message.
Title	Tiered Recovery of Flight Objects: Subsequent Detection of a Recovering Node (same iteration).
Status	<Validated>
Rationale	In order to allow Flight Objects to be recovered sequentially and mitigate a storm of updates, a Tier approach is used for Recovery where Flight Objects are received by the recovering SWIM Node according to a defined priority given by the Tier information. This requirement allows to detect situations where the recovering is in progress but a new periodic RECOVERY_STATUS is received from the recovering Node for the same recovering iteration. In that case, the RECOVERY STATUS is ignored, thus avoiding multiple publications of the same Flight Objects. This requirement covers the following NIST security controls: CP-7 a, CP-10.
Category	<Interface><Security>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Subscription handler><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0779
Requirement	Upon reception of a RECOVERY_STATUS from a SWIM Node with a Recovery Status set to “TRUE” and containing a different Recovery Context ID than the already locally stored Recovery Context ID for the recovering SWIM Node, the receiving SWIM Node shall: - cancel the current recovering activities with the SWIM Node, - publish all locally-managed Flight Objects for which the recovering Node (‘iop_stakeholder_id’) is in Tier(s) ‘TierSequence’, and - store the new received Recovery Context ID for the recovering SWIM Node.
Title	Tiered Recovery of Flight Objects: Subsequent Detection of a Recovering Node (new iteration).
Status	<Validated>
Rationale	This requirement allows to detect situations where the recovering is in progress but a new periodic RECOVERY_STATUS is received from the recovering Node indicating a re-start of the recovering process. This might occur for instance when the recovering node fails again during the recovery. The current recovering activities are cancelled and a new recovery process is re-started. This requirement covers the following NIST security controls: CP-7 a, CP-10.
Category	<Interface><Security>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Subscription handler><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0831
Requirement	Upon reception of a RECOVERY_STATUS from a SWIM Node with a Recovery Status set to “FALSE”, the receiving SWIM Node shall delete the Recovery Context ID stored for the recovering SWIM Node and considers the recovery completed for the SWIM Node.
Title	Tiered Recovery of Flight Objects: Recovery Completion Indication

Status	<Validated>
Rationale	This requirement allows to free the resources set for the recovery of a remote Node when that recovering SWIM Node indicates that it has completed its recovery process. The Recovery Context ID that was stored previously and associated with the recovering Node is freed. This requirement covers the following NIST security controls: CP-7 a, CP-10.
Category	<Interface><Security>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Subscription handler><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0811
Requirement	Upon request of the IOP Application or when the automatic recovery is provided (optional capability) and local triggering conditions apply, the recovering SWIM Node shall request (RequestFORecovery operation) Flight Objects re-publication to the concerning FDMP(s).
Title	Tiered Recovery of Flight Objects: RequestFORecovery operation (recovering node)
Status	<Validated>
Rationale	It could happen that not all the tiers or/and not all the Flight Object within a given tier have been completed successfully. In that case, the Application IOP (or the SWIM Node in automatic mode) can request the recovering SWIM Node to request missing FOs re-publication. This requirement covers the following NIST security controls: CP-7 a, CP-10.
Category	<Interface><Security>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>

Roles	<Subscriber><Subscription handler><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0792
Requirement	Upon reception of a request from the IOP Application to stop recovery or when the automatic recovery is provided (optional capability) and local stopping conditions apply, the SWIM Node shall: - publish a RECOVERY_STATUS with the RecoveryStatus set to FALSE with the RecoveryContextId associated with the current recovery iteration, - either stop or continue publishing periodically the RECOVERY_STATUS.
Title	Tiered Recovery of Flight Objects: Recovery Process Completion (recovering node)
Status	<Validated>
Rationale	On request of the IOP Application or automatically triggered (conditions are specified in the automatic recovery policy), the IOP Recovery Status information is sent once to inform other SWIM Nodes that the Recovery process is completed. It is a local decision to stop or continue sending periodically the RECOVERY_STATUS. The IOP Status is managed independently by the IOP Application. The SWIM Node can be declared ENABLED while the recovery process is still on-going. This requirement covers the following NIST security controls: CP-7 a, CP-10.
Category	<Interface><Security>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Subscription handler><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1070
Requirement	The SWIM Technical Infrastructure shall enforce Flight Object update release checking by means of processing: - Flight Object release from latest Flight Object summary. - Flight Object publisher identifier from latest Flight Object summary. - Flight Object Cluster release of the Flight Object Clusters being received. - Flight Object Cluster publisher identifier of the Flight Object Clusters being received.
Title	Flight Object update release checking capability
Status	<In Progress>
Rationale	An FO release is a version of a Flight Object and is a sequence of all the releases of the clusters of the Flight Object. This is basically a version vector (https://en.wikipedia.org/wiki/Version_vector) for tracking changes to the clusters of a Flight Object. At Flight Object creation, all the clusters releases are equal to zero and each time a Flight Object is updated, the releases of the updated clusters are incremented and the FDMP publishes, a part of the updated FOCluster, the FOSummary containing the FO release. When a Flight Object is updated independently at multiple locations, the FO release information may not follow the correct ordering that allows the receiving FDCs to order the updates and discard old FO releases. This conflicting situation is referred to as a collision of FO updates. Given two FO Releases Rx and Ry: - Rx < Ry, if for each cluster i Rx[i] ≤ Ry[i] and there is at least one cluster j where Rx[j] < Ry[j] (Rx is prior to Ry). - Rx = Ry, if for each cluster i Rx[i]=Ry[i] (Rx is identical to Ry) A collision happens when Rx ≠ Ry and neither Rx < Ry nor Ry < Rx. A collision is also when Rx = Ry but the updates are from different publishers. The SWIM-TI enforces Flight Object update release checking in order to detect possible collisions and to react accordingly. This requirement provides the means used to enforce the checking. The latest Flight Object summary is the one locally stored at the moment when the update being processed is received.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>

Roles	<Subscriber><Subscription handler><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.1080
Requirement	The SWIM Technical Infrastructure Flight Object update release checking shall result in one of the following status: - Valid Flight Object release. - Invalid Flight Object release.
Title	Flight Object update release checking result status
Status	<In Progress>
Rationale	An FO release is a version of a Flight Object and is a sequence of all the releases of the clusters of the Flight Object. This is basically a version vector (https://en.wikipedia.org/wiki/Version_vector) for tracking changes to the clusters of a Flight Object. At Flight Object creation, all the clusters releases are equal to zero and each time a Flight Object is updated, the releases of the updated clusters are incremented and the FDMP publishes, a part of the updated FOCluster, the FOSummary containing the FO release. When a Flight Object is updated independently at multiple locations, the FO release information may not follow the correct ordering that allows the receiving FDCs to order the updates and discard old FO releases. This conflicting situation is referred to as a collision of FO updates. Given two FO Releases Rx and Ry: - Rx < Ry, if for each cluster i Rx[i] ≤ Ry[i] and there is at least one cluster j where Rx[j] < Ry[j] (Rx is prior to Ry). - Rx = Ry, if for each cluster i Rx[i]=Ry[i] (Rx is identical to Ry) A collision happens when Rx ≠ Ry and neither Rx < Ry nor Ry < Rx. A collision is also when Rx = Ry but the updates are from different publishers. The SWIM-TI enforces Flight Object update release checking in order to detect possible collisions and to react accordingly. This requirement provides the status resulting from the release checking enforcement: - Valid Flight Object release status: the Flight Object update release successfully passed the checking.

	- Invalid Flight Object release status: collisions/inconsistencies have been detected during the Flight Object update release checking.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Subscription handler><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.1090
Requirement	In case the Flight Object release checking results in “Valid Flight Object release”, the SWIM Technical Infrastructure shall store the updated Flight Object and shall notify the application layer with the Flight Object Identifier and the whole set of Clusters and release identifiers corresponding to the update of the Flight Object..
Title	Flight Object update release checking: behaviour in case of “Valid Flight Object release”
Status	<In Progress>
Rationale	An FO release is a version of a Flight Object and is a sequence of all the releases of the clusters of the Flight Object. This is basically a version vector (https://en.wikipedia.org/wiki/Version_vector) for tracking changes to the clusters of a Flight Object. At Flight Object creation, all the clusters releases are equal to zero and each time a Flight Object is updated, the releases of the updated clusters are incremented and the FDMP publishes, a part of the updated FOCluster, the FOSummary containing the FO release. When a Flight Object is updated independently at multiple locations, the FO release information may not follow the correct ordering that allows the receiving FDCs to order the updates and discard old FO releases. This conflicting situation is referred to as a collision of FO updates. Given two FO Releases Rx and Ry: - Rx < Ry, if for each cluster i Rx[i] ≤ Ry[i] and there is at least one cluster j where Rx[j] < Ry[j] (Rx is prior to Ry). - Rx = Ry, if for each cluster i Rx[i]=Ry[i] (Rx is identical to Ry)

	A collision happens when Rx ≠ Ry and neither Rx < Ry nor Ry < Rx. A collision is also when Rx = Ry but the updates are from different publishers. The SWIM-TI enforces Flight Object update release checking in order to detect possible collisions and to react accordingly. This requirement provides the expected behaviour in case of the Flight Object update release checking results in “Valid Flight Object release”.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Subscription handler><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.1100
Requirement	In case the Flight Object release checking results in “Invalid Flight Object release”, the SWIM Technical Infrastructure shall discard the Flight Object update and shall notify the application layer with discarded Flight Object release, the identifier of the Flight Object update publisher and the reason for discarding the update.
Title	Flight Object update release checking: behaviour in case of “Invalid Flight Object release”
Status	<In Progress>
Rationale	An FO release is a version of a Flight Object and is a sequence of all the releases of the clusters of the Flight Object. This is basically a version vector (https://en.wikipedia.org/wiki/Version_vector) for tracking changes to the clusters of a Flight Object. At Flight Object creation, all the clusters releases are equal to zero and each time a Flight Object is updated, the releases of the updated clusters are incremented and the FDMP publishes, a part of the updated FOCluster, the FOSummary containing the FO release. When a Flight Object is updated independently at multiple locations, the FO release information may not follow the correct ordering that allows the

	receiving FDCs to order the updates and discard old FO releases. This conflicting situation is referred to as a collision of FO updates. Given two FO Releases Rx and Ry: - Rx < Ry, if for each cluster i Rx[i] ≤ Ry[i] and there is at least one cluster j where Rx[j] < Ry[j] (Rx is prior to Ry). - Rx = Ry, if for each cluster i Rx[i]=Ry[i] (Rx is identical to Ry) A collision happens when Rx ≠ Ry and neither Rx < Ry nor Ry < Rx. A collision is also when Rx = Ry but the updates are from different publishers. The SWIM-TI enforces Flight Object update release checking in order to detect possible collisions and to react accordingly. This requirement provides the expected behaviour in case of the Flight Object update release checking results in “Invalid Flight Object release”.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Subscription handler><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.1110
Requirement	Flight Object update release checking shall result in “Valid Flight Object release” when the following conditions are met: - there is at least one cluster whose release is greater than the corresponding release stored locally, AND - none of the locally stored clusters have releases greater than the received releases.
Title	Flight Object update release checking: conditions for “Valid Flight Object release”
Status	<In Progress>
Rationale	An FO release is a version of a Flight Object and is a sequence of all the releases of the clusters of the Flight Object. This is basically a version vector (https://en.wikipedia.org/wiki/Version_vector) for tracking changes to

	the clusters of a Flight Object. At Flight Object creation, all the clusters releases are equal to zero and each time a Flight Object is updated, the releases of the updated clusters are incremented and the FDMP publishes, a part of the updated FOCluster, the FOSummary containing the FO release. When a Flight Object is updated independently at multiple locations, the FO release information may not follow the correct ordering that allows the receiving FDCs to order the updates and discard old FO releases. This conflicting situation is referred to as a collision of FO updates. Given two FO Releases Rx and Ry: - Rx < Ry, if for each cluster i Rx[i] ≤ Ry[i] and there is at least one cluster j where Rx[j] < Ry[j] (Rx is prior to Ry). - Rx = Ry, if for each cluster i Rx[i]=Ry[i] (Rx is identical to Ry) A collision happens when Rx ≠ Ry and neither Rx < Ry nor Ry < Rx. A collision is also when Rx = Ry but the updates are from different publishers. The SWIM-TI enforces Flight Object update release checking in order to detect possible collisions and to react accordingly. This requirement provides conditions when the Flight Object update release checking output status is “Valid Flight Object release”.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Subscription handler><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A

[REQ]

Identifier	REQ-14.01.04-TS-0901.1120
Requirement	Flight Object update release checking shall result in “Invalid Flight Object release” when the following conditions are met: - there is not at least one cluster whose release is greater than the corresponding cluster release stored locally, OR - there is at least one cluster whose release is less than the corresponding

	cluster release stored locally, OR - there is not at least one cluster whose release is not equal to the corresponding cluster release stored locally AND the publisher identifier is different from the one corresponding to the Flight Object release stored locally.
Title	Flight Object update release checking: conditions for “Invalid Flight Object release”
Status	<In Progress>
Rationale	An FO release is a version of a Flight Object and is a sequence of all the releases of the clusters of the Flight Object. This is basically a version vector (https://en.wikipedia.org/wiki/Version_vector) for tracking changes to the clusters of a Flight Object. At Flight Object creation, all the clusters releases are equal to zero and each time a Flight Object is updated, the releases of the updated clusters are incremented and the FDMP publishes, a part of the updated FOCluster, the FOSummary containing the FO release. When a Flight Object is updated independently at multiple locations, the FO release information may not follow the correct ordering that allows the receiving FDCs to order the updates and discard old FO releases. This conflicting situation is referred to as a collision of FO updates. Given two FO Releases Rx and Ry: - Rx < Ry, if for each cluster i Rx[i] ≤ Ry[i] and there is at least one cluster j where Rx[j] < Ry[j] (Rx is prior to Ry). - Rx = Ry, if for each cluster i Rx[i]=Ry[i] (Rx is identical to Ry) A collision happens when Rx ≠ Ry and neither Rx < Ry nor Ry < Rx. A collision is also when Rx = Ry but the updates are from different publishers. The SWIM-TI enforces Flight Object update release checking in order to detect possible collisions and to react accordingly. This requirement provides conditions when the Flight Object update release checking output status is “Invalid Flight Object release”.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><Function/Behaviour>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Subscription handler><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>

<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A

3.3.9.2.2.3 Performance Requirements

Technology specific performance requirements have been captured as OMG DDS QoSs and specified in §3.3.9.2.2.7.

3.3.9.2.2.4 Security and Integrity Requirements

In this section security requirements concerning the FlightObjectDistribution interface are provided.

[REQ]

Identifier	REQ-14.01.04-TS-0901.0505
Requirement	For all Data distributed through the FlightObjectDistribution interface confidentiality, integrity and authenticity shall be ensured.
Title	Flight Object Data confidentiality, integrity and authenticity.
Status	<In Progress>
Rationale	The SWIM infrastructure providing / consuming Flight Object data requires trust between all participant nodes. It is important to protect the SWIM infrastructure and Data from compromised SWIM nodes and malicious eavesdroppers. Security solution adopted to cover that needs shall be based of interoperable COTS. Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN).For architectural aspects and terminology refer to latest 14.01.03 TAD.
Category	<Interface><Security>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<SLA><ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Subscription handler><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>

<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0510
Requirement	All the entities exchanging Flight Object data through the Flight Object Distribution interface shall be authenticated and Authorized.
Title	Flight Object Data consumers and providers entities Authentication and Authorization
Status	<In Progress>
Rationale	The SWIM infrastructure providing / consuming Flight Object data requires trust between all participant nodes. It is important to protect the SWIM infrastructure and Data from compromised SWIM nodes and malicious eavesdroppers. Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN). For architectural aspects and terminology refer to latest 14.01.03 TAD. This requirement covers NIST security control IA-8.
Category	<Interface><Security>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<SLA><ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer><Subscriber><Subscription handler><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0525
Requirement	For safety reasons maximum sample size shall be properly specified, documented and configured.
Title	Protection against Overload Maximum Sample Size
Status	<In Progress>
Rationale	To protect SWIM-TI from large samples that may corrupt and/or break DDS applications, it is important to define a maximum size for data samples for a safe deployment of DDS-based applications. This will also improve security of the SWIM nodes as buffer overflow techniques may be used to get

	privileged access to remote nodes. This requirement covers NIST security controls SC-5.
Category	<Interface><Security>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<SLA><ICD><Governance>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer><Subscriber><Subscription handler><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0526
Requirement	For safety reasons the SWIM-TI layer shall enforce maximum sample size policy.
Title	Protection against Overload Maximum Sample Size (enforcement)
Status	<In Progress>
Rationale	To protect SWIM-TI from large samples that may corrupt and/or break DDS applications, it is important to define a maximum size for data samples for a safe deployment of DDS-based applications. This will also improve security of the SWIM nodes as buffer overflow techniques may be used to get privileged access to remote nodes. At SWIM-TI layer, the maximum sample size is configured (REQ-14.01.04-TS-0901.0525) and enforced accordingly. This requirement covers NIST security controls SC-5.
Category	<Interface><Security>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<SLA><ICD><Governance>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Service provider><Service consumer><Subscriber><Subscription handler><Publisher><Publication consumer><Publication mediator>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

3.3.9.2.2.5 Data Transfer

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications

3.3.9.2.2.6 Transactions

Requirements concerning this category have not been identified during SESAR 1 programme. This requirement category may be further investigated according to the evolution of the SWIM-TI Technical Specifications.

3.3.9.2.2.7 Interface Instantiation

In this section the OMG DDS specific model or instantiation of the FlightObjectDistribution interface is provided. The binding and the contract have been defined according to requirements, UML model and interface descriptions provided in the previous sections. In particular the contract consists of OMG DDS IDL, OMG DDS Topics and QoS as described here below.

As anticipated above, this interface allows to share also IOP-Status information. The latter is a data shared for supervision purposes and it is provided here and not in the SPV requirements chapter just because its contract is strongly related to Flight Object distribution.

[REQ]

Identifier	REQ-14.01.04-TS-0901.0700
Requirement	FlightObjectDistribution interface shall be instantiated using the following binding: + Protocol stack: - DDS 1.2 over DDS Security 1.0, DDSI 2.1 over UDP + MEPs: - PSPUSH-MEP, PSPULL-MEP + Fault handling: - As defined per standard + Encoding: - As defined per standard + Security:

	- Confidentiality: message level as defined per DDS Security - Integrity: message level as defined per DDS Security - Authenticity: mutual, message level as defined per DDS Security - Authorization: message level as defined per DDS Security - Non-repudiation: message level as defined per DDS Security + Contract: - formalism of contract description: OMG IDL, QoS Configuration, DDS Security Configuration - minimum: not applicable - reference: Blue Profile Technical Specification, ISRM + Interoperability: as for OMG DDSI 2.1 and DDS Security 1.0
Title	FlightObjectDistribution Interface binding
Status	<In Progress>
Rationale	FlightObjectDistribution requires a specific Interface configuration. At the time of writing this TS (May 2016), OMG DDS Security is an adopted OMG BETA specification being standardized. BP TS just identifies which DDS Security plugins have to be used and how. Further evolutions of DDS Security BETA, until it will be considered standard, are only expected to fix specification issues that may be raised during the one-year finalization task force . This limits the impact on the BP TS. Verification of this requirement has as precondition the verification of requirements REQ-14.01.04-TS-0901.0500 and REQ-14.01.04-TS-0901.0515, This requirement can be fully verified only by interoperable DDS Security implementations. If not available, network level or message level (application) mechanisms may be used to fill the gap.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Service binding>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0191	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0102	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0117	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0146	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0147	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0153	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0164	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0189	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0010	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0109	N/A

<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0184	N/A
<SATISFIES>	<ATMS Requirement>	REQ-14.01.04-TS-0001.0400	<Full>
<SATISFIES>	<ATMS Requirement>	REQ-14.01.04-TS-0001.0410	<Full>
<SATISFIES>	<ATMS Requirement>	REQ-14.01.04-TS-0001.0420	<Full>
<SATISFIES>	<ATMS Requirement>	REQ-14.01.04-TS-0001.0430	<Full>
<SATISFIES>	<ATMS Requirement>	REQ-14.01.04-TS-0001.0500	<Full>
<SATISFIES>	<ATMS Requirement>	REQ-14.01.04-TS-0001.0510	<Full>
<SATISFIES>	<ATMS Requirement>	REQ-14.01.04-TS-0001.0520	<Full>
<SATISFIES>	<ATMS Requirement>	REQ-14.01.04-TS-0001.0530	<Full>
<SATISFIES>	<ATMS Requirement>	REQ-14.01.04-TS-0001.0621	<Full>
<SATISFIES>	<ATMS Requirement>	REQ-14.01.04-TS-0001.0631	<Full>
<SATISFIES>	<ATMS Requirement>	REQ-14.01.04-TS-0001.0052	<Full>
<SATISFIES>	<ATMS Requirement>	REQ-14.01.04-TS-0001.0091	<Full>
<SATISFIES>	<ATMS Requirement>	REQ-14.01.04-TS-0001.0302	<Full>
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0102	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0183	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0106	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0107	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0108	N/A
<INCLUDES>	<ATMS Requirement>	REQ-14.01.04-TS-0811.0185	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

For each one of the three message types introduced before a corresponding OMG DDS topic has been identified providing topic name, topic structure and QoS. In the following tables the DDS QoS are reported pointing out the values for each policy and referring to the specific entity.

Table 3-8: Flight Object Summary Distribution QoS, Topic name and structure

Flight Object Summary Distribution QoS			
OMG DDS Topic Name	FO_SUMMARY	**OMG DDS Data Type**	IOP::FO::FOSummary
OMG DDS Topic QoS			
QoS Policy	**QoS Policy Attribute**	**Value**	**Note**
DURABILITY	kind	VOLATILE	
OWNERSHIP	kind	SHARED	DDS DEFAULT value
RELIABILITY	kind	RELIABLE	
DESTINATION_ORDER	kind	BY_SOURCE_TIMESTAMP	This requires time synchronization
DEADLINE	period	SP-IOP-Checksum_Interval x factor	SP-IOP-Checksum_Interval determines how often each summary instance is sent - a multiplication factor must be applied to it to determine the deadline interval

HISTORY	kind	KEEP_LAST	DDS DEFAULT value
HISTORY	depth	1	DDS DEFAULT value
OMG DDS Subscriber QoS			
QoS Policy	**QoS Policy Attribute**	**Value**	**Note**
PARTITION	name	Name of the partition/s the local system instance belongs to	
OMG DDS Data Reader QoS			
QoS Policy	**QoS Policy Attribute**	**Value**	**Note**
DURABILITY	kind	VOLATILE	
OWNERSHIP	kind	SHARED	DDS DEFAULT value
RELIABILITY	kind	RELIABLE	
DESTINATION_ORDER	kind	BY_SOURCE_TIMESTAMP	This requires time synchronization
HISTORY	kind	KEEP_LAST	DDS DEFAULT value
HISTORY	depth	1	DDS DEFAULT value
RESOURCE_LIMITS	max_instances	> total_flight_objects	
RESOURCE_LIMITS	max_samples	> total_flight_objects	
RESOURCE_LIMITS	max_samples_per_instance	1	
DEADLINE	period	SP-IOP-Checksum_Interval x factor	SP-IOP-Checksum_Interval determines how often each summary instance is sent - a multiplication factor must be applied to it to determine the deadline interval
OMG DDS Publisher QoS			
QoS Policy	**QoS Policy Attribute**	**Value**	**Note**
PARTITION	name	All the partitions of the IOP Area Stakeholders	
OMG DDS Data Writer QoS			
QoS Policy	**QoS Policy Attribute**	**Value**	**Note**
DURABILITY	kind	VOLATILE	
OWNERSHIP	kind	SHARED	DDS DEFAULT value
RELIABILITY	kind	RELIABLE	
DESTINATION_ORDER	kind	BY_SOURCE_TIMESTAMP	This requires time

			synchronization
HISTORY	kind	KEEP_LAST	DDS DEFAULT value
HISTORY	depth	1	DDS DEFAULT value
RESOURCE_LIMITS	max_instances	>(SP-IOP-Max_FO_Managed) x (total_number_of_publishers)	
RESOURCE_LIMITS	max_samples	>(SP-IOP-Max_FO_Managed) x (total_number_of_publishers)	
RESOURCE_LIMITS	max_samples_per_instance	1	
WRITER_DATA_LIFECYCLE	autodispose_unregistered_instances	FALSE	
DEADLINE	period	SP-IOP-Checksum_Interval x factor	SP-IOP-Checksum_Interval determines how often each summary instance is sent - a multiplication factor must be applied to it to determine the deadline interval. The "factor" must be greater than 1.

Table 3-9: Flight Object Cluster Distribution QoS, Topic name and structure

Flight Object Cluster Distribution QoS			
OMG DDS Topic Name	FO_CLUSTER	**OMG DDS Data Type**	IOP::FO::FOCluster
OMG DDS Topic QoS			
QoS Policy	**QoS Policy Attribute**	**Value**	**Note**
DURABILITY	kind	VOLATILE	
OWNERSHIP	kind	SHARED	DDS DEFAULT value
RELIABILITY	kind	RELIABLE	
DESTINATION_ORDER	kind	BY_SOURCE_TIMESTAMP	This requires time synchronization
HISTORY	kind	KEEP_LAST	DDS DEFAULT value
HISTORY	depth	1	DDS DEFAULT value
OMG DDS Subscriber QoS			
QoS Policy	**QoS Policy Attribute**	**Value**	**Note**
PARTITION	name	Name of the partition/s the local system instance belongs to	
PRESENTATION	coherent_access	FALSE	

OMG DDS Data Reader QoS			
QoS Policy	**QoS Policy Attribute**	**Value**	**Note**
DURABILITY	kind	VOLATILE	
OWNERSHIP	kind	SHARED	DDS DEFAULT value
RELIABILITY	kind	RELIABLE	
DESTINATION_ORDER	kind	BY_SOURCE_TIMESTAMP	This requires time synchronization
HISTORY	kind	KEEP_LAST	DDS DEFAULT value
HISTORY	depth	1	DDS DEFAULT value
RESOURCE_LIMITS	max_instances	> SP-IOP-Max_FO_Stored x number_of_clusters	
RESOURCE_LIMITS	max_samples	> SP-IOP-Max_FO_Stored x number_of_clusters	
RESOURCE_LIMITS	max_samples_per_instance	1	
OMG DDS Publisher QoS			
QoS Policy	**QoS Policy Attribute**	**Value**	**Note**
PARTITION	name	All and only those partitions of the IOP area stakeholders part of the distr bution list of the Flight Object being published. Recovery phase publications: All and only those partitions of the IOP area stakeholders part of the distr bution list AND recovering the Flight Object being published.	
PRESENTATION	coherent_access	FALSE	
OMG DDS Data Writer QoS			
QoS Policy	**QoS Policy Attribute**	**Value**	**Note**
DURABILITY	kind	VOLATILE	
OWNERSHIP	kind	SHARED	DDS DEFAULT value
RELIABILITY	kind	RELIABLE	
DESTINATION_ORDER	kind	BY_SOURCE_TIMESTAMP	This requires time synchronization
HISTORY	kind	KEEP_LAST	DDS DEFAULT value

HISTORY	depth	1	DDS DEFAULT value
RESOURCE_LIMITS	max_instances	> SP-IOP-Max_FO_Stored x number_of_clusters	
RESOURCE_LIMITS	max_samples	> SP-IOP-Max_FO_Stored x number_of_clusters	
RESOURCE_LIMITS	max_samples_per_instance	1	
WRITER_DATA_LIFECYCLE	autodispose_unregistered_instances	FALSE	

Table 3-10: IOP and Recovery Status Distribution QoS, Topic name and structure

IOP Status Distribution QoS			
OMG DDS Topic Name	IOP_STATUS	**OMG DDS Data Type**	IOP::FO::IOPStatus
OMG DDS Topic Name	RECOVERY_STATUS	**OMG DDS Data Type**	IOP::FO::IOPRecoveryStatus
OMG DDS Topic QoS			
QoS Policy	**QoS Policy Attribute**	**Value**	**Note**
DURABILITY	kind	VOLATILE	
OWNERSHIP	kind	SHARED	DDS DEFAULT value
RELIABILITY	kind	RELIABLE	
DESTINATION_ORDER	kind	BY_SOURCE_TIMESTAMP	This requires time synchronization
HISTORY	kind	KEEP_LAST	DDS DEFAULT value
HISTORY	depth	1	DDS DEFAULT value
OMG DDS Subscriber QoS			
QoS Policy	**QoS Policy Attribute**	**Value**	**Note**
PARTITION	name	Name of the partition/s the local system instance belongs to	
OMG DDS Data Reader QoS			
QoS Policy	**QoS Policy Attribute**	**Value**	**Note**
DURABILITY	kind	VOLATILE	
OWNERSHIP	kind	SHARED	DDS DEFAULT value
RELIABILITY	kind	RELIABLE	
DESTINATION_ORDER	kind	BY_SOURCE_TIMESTAMP	This requires time synchronization

HISTORY	kind	KEEP_LAST	DDS DEFAULT value
HISTORY	depth	1	DDS DEFAULT value
RESOURCE_LIMITS	max_instances	LENGTH_UNLIMITED	
RESOURCE_LIMITS	max_samples	LENGTH_UNLIMITED	
RESOURCE_LIMITS	max_samples_per_instance	1	
OMG DDS Publisher QoS			
QoS Policy	**QoS Policy Attribute**	**Value**	**Note**
PARTITION	name	All the partitions of the IOP Area Stakeholders.	
OMG DDS Data Writer QoS			
QoS Policy	**QoS Policy Attribute**	**Value**	**Note**
DURABILITY	kind	VOLATILE	
OWNERSHIP	kind	SHARED	DDS DEFAULT value
RELIABILITY	kind	RELIABLE	
DESTINATION_ORDER	kind	BY_SOURCE_TIMESTAMP	This requires time synchronization
HISTORY	kind	KEEP_LAST	DDS DEFAULT value
HISTORY	depth	1	DDS DEFAULT value
RESOURCE_LIMITS	max_instances	LENGTH_UNLIMITED	
RESOURCE_LIMITS	max_samples	LENGTH_UNLIMITED	
RESOURCE_LIMITS	max_samples_per_instance	1	
WRITER_DATA_LIFECYCLE	autodispose_unregistered_instances	FALSE	

Table 3-11: FlightObjectDistribution Interface OMG DDS Domain Participant and related UDP ports

OMG Domain Participant	
OMG DDS Domain Participant Identifier	**0**
DDSI Discovery Traffic UDP Ports	
Kind Of Traffic	**UDP Port**
MULTICAST	7400
UNICAST	7410
DDSI User Traffic UDP Ports	

Kind Of Traffic	UDP Port
MULTICAST	7401
UNICAST	7411

Table 3-12: FlightObjectDistribution Interface UDP Fragmentation and DDSI configuration

DDSI configuration related to UDP Fragmentation Information	
Configuration To Avoid Fragmentation (e.g. needed when IPSec and Firewalls are used)	
DDSI	**Size (bytes)**
Max Message Size	1228
Fragment Size	1032 (minimum default DDSI value)
Configuration With Fragmentation	
DDSI	**Size (bytes)**
Max Message Size	1228 < Value <= 65536
Fragment Size	1032 (minimum default DDSI value)

Latest version of the IDL for the FlightObjectDistribution interface contract is provided here after.

BPFDD-FlightObjectDistribution.zip

[REQ]

Identifier	REQ-14.01.04-TS-0901.0485
Requirement	The FlightObjectDistribution interface shall be instantiated according to the contract available in the latest 14.01.04 Blue Profile Technical specification.
Title	FlightObjectDistribution Interface Binding Contract
Status	<Validated>
Rationale	FlightObjectDistribution contract consists of the IDL, Topic names, Topics structures and QoS available in the 14.01.04 Technical Specification. DDS Security configuration:REQ-14.01.04-TS-0901.0685, REQ-14.01.04-TS-0901.0690, REQ-14.01.04-TS-0901.0691, REQ-14.01.04-TS-0901.0695, REQ-14.01.04-TS-0901.0698.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>

Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.1130
Requirement	Once completed successfully the validity checks applicable to the SharedFlightObject interface DeleteFlightObject operation, the SWIM-TI layer shall delete the target Flight Object via the FlightObjectDistribution interface by stopping the publication of related information and by performing OMG DDS dispose operation on the concerning Flight Object Cluster instances and Flight Object Summary instance.
Title	FlightObjectDistribution Interface Binging Contract – Flight Object deletion
Status	<In Progress>
Rationale	This requirement specifies how the Flight Object deletion is performed via OMG DDS technology. In particular all the Flight Object Cluster instances and the Flight Object Summary related to the target (to be deleted) Flight Object are disposed.
Category	<Interface><Interoperability>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>

<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0685
Requirement	The FlightObjectDistribution interface instantiation shall use DDS:Auth:PKI-RSA/DSA-DH DDS Security Builtin Authentication plugin.
Title	FlightObjectDistribution Interface Binging Contract DDS Security Authentication plugin
Status	<In Progress>
Rationale	Compliance with security interoperable protocol for DDS and in particular with Builtin Authentication plugin specified in DDS Security FTF Beta 1 clause 9.3. As per DDS Security clause 9.3 three things have to be configured to enable the plug-in: 1. The X.509 Certificate that defines the Shared CA (this certificate contains the 2048-bit RSA Public Key of the CA); 2. The 2048-bit RSA Private Key of the DomainParticipant; 3. An X.509 Certificate that chains up to the Shared CA , that binds the 2048-bit RSA Public Key of the DomainParticipant to the Distinguished Name (subject name) for the DomainParticipant and any intermediate CA certificates required to build the chain. DDS Security does not mandate how to configure this plugin (refer to 9.3.1) but it is important to clarify that the plugin uses a shared Certification Authority (CA) and that all the participants are pre-configured with shared-CA.
Category	<Interface><Interoperability><Security>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0690
Requirement	The FligthObjectDistribution interface instantiation shall use DDS:Access:PKI-Signed-XML-Permissions DDS Security Builtin Access Control plugin.
Title	FlightObjectDistribution Interface Binging Contract DDS Security Access Control plugin
Status	<In Progress>
Rationale	Compliance with security interoperable protocol for DDS and in particular with Builtin Access Control plugin specified in DDS Security FTF Beta 1 clause 9.4. As per DDS Security clause 9.4, three things have to be configured to enable the plug-in: 1. the Permissions CA X.509 certificate; 2. the Domain governance signed by the shared Permissions CA; 3. the DomainParticipant permissions signed by the Permissions CA. In particular Domain governance configuration includes: - which topics shall be secured and how. - Whether discovery is secured and how. DomainParticipant permissions configuration includes: - what DDS domaiin Id can be joined. - which topics can be read/write for each domain. - ties of the SubjectName matching the one on IdentityCertificate.
Category	<Interface><Interoperability><Security>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0691
Requirement	The DDS:Access:PKI-Signed-XML-Permissions DDS Security Builtin Access Control plugin configuration shall be as follows: - grant access only to authenticated DDS entity; - DDS metadata, reliability traffic and discovery shall be protected using MAC; - Payload data sent on all the three <BP

	FDD> topics shall be protected with Encrypt+MAC.
Title	FlightObjectDistribution Interface Binging Contract DDS Security Access Control plugin configuration
Status	<In Progress>
Rationale	DDS:Access:PKI-Signed-XML-Permissions DDS Security Builtin Access Control plugin configuration for binding REQ-14.01.04-TS-0901.0700.
Category	<Interface><Interoperability><Security>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0695
Requirement	The FlightObjectDistribution interface instantiation shall use DDS:Crypto:AES-CTR-HMAC-RSA/DSA-DH DDS Security Builtin Cryptography plugin.
Title	FlightObjectDistribution Interface Binging Contract DDS Security Cryptography plugin
Status	<In Progress>
Rationale	Compliance with security interoperable protocol for DDS and in particular with Builtin Cryptography plugin specified in DDS Security FTF Beta 1 clause 9.5. In particular the plugin provides the following modes of operation: - AES128 and AES256 for encryption; - SHA1 and SHA256 for digest; - HMAC-SHA1 and HMAC-SHA256 for MAC. The <BP FDD> Domain governance and DomainParticipant documents shall specifies the applicable modes for application data (Topic), metadata, reliability traffic and discovery.
Category	<Interface><Interoperability><Security>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD>

Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0698
Requirement	The FlightObjectDistribution interface instantiation shall use Logging DDS Security Builtin plugin.
Title	FlightObjectDistribution Interface Binging Contract DDS Security Logging plugin
Status	<In Progress>
Rationale	Compliance with security interoperable protocol for DDS and in particular with Builtin Logging plugin specified in DDS Security FTF Beta 1 clause 9.6.
Category	<Interface><Interoperability><Security>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD>
Point of view	<ATM service><SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Interoperability testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<APPLIES_TO>	<ATMS Requirement>	REQ-14.01.04-TS-0901.0700	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>

<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

The following set of requirements have been identified in the context of the "Flight Object Overlay" described in the SWIM-TI TAD and aiming at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN).

[REQ]

Identifier	REQ-14.01.04-TS-0901.0500
Requirement	The DDS implementation shall ensure the confidentiality and integrity of the data samples.
Title	DDS Implementation data confidentiality and integrity support
Status	<In Progress>
Rationale	The SWIM infrastructure requires trust between all participant nodes. It is important to protect the SWIM infrastructure from compromised SWIM nodes and malicious eavesdroppers. Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN). For architectural aspects and terminology refer to latest 14.01.03 TAD. This requirement covers NIST security controls SC-8.
Category	<Interface><Security>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP Core>
Domain of interest	<SLA>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0515
Requirement	The DDS implementation shall ensure the authentication and authorization of the DDS writers and readers.
Title	DDS Implementation Authentication and Authorization of DDS Writer and

	Readers
Status	<In Progress>
Rationale	The SWIM infrastructure providing / consuming Flight Object data requires trust between all participant nodes. It is important to protect the SWIM infrastructure and Data from compromised SWIM nodes and malicious eavesdroppers.In particular the authorization and the authentication of publishing/receiving entities are needed. In vase of DDS this consists of the authentication and authorization of DDS Data Writers and Data Reader. Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN). For architectural aspects and terminology refer to latest 14.01.03 TAD. This requirement covers NIST security controls AC-3, IA-5 a and SC-8.
Category	<Interface><Security>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP Core>
Domain of interest	<SLA>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0520
Requirement	The DDS implementation shall use a pre-defined set port numbers for its communication transport.
Title	Firewall Traversal
Status	<Validated>
Rationale	Define a fixed set of port numbers for firewall traversal. Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN). For architectural aspects and terminology refer to latest 14.01.03 TAD. This requirement covers NIST security controls SA-9 (2)
Category	<Interface><Security>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP Core>
Domain of interest	<ICD>

Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0535
Requirement	The DDS implementation shall include a standard Path MTU discovery protocol.
Title	Dynamic Path MTU Discovery
Status	<In Progress>
Rationale	Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN). For architectural aspects and terminology refer to latest 14.01.03 TAD.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP Core>
Domain of interest	<ICD><SLA>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>

<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0540
Requirement	The DDS implementation shall only convey externally visible IP addresses within Participant and Endpoint discovery messages.
Title	WAN DDS Deployment and Natting
Status	<In Progress>
Rationale	When Network Address Translation is used, devices analyse IP packet and translate locally visible only addresses to/from external public addresses. This is usually only done by analysing IP headers and/or TCP/UDP headers only so any local addressing within the payload of the messages will not be translated by the devices. DDS discovery protocols will have to make sure any addressing information exchanged with external participants is publically accessible from outside. Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN). For architectural aspects and terminology refer to latest 14.01.03 TAD.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP Core>
Domain of interest	<ICD><SLA>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0545
Requirement	The DDS implementation shall be able to never send UDP datagrams larger than the path MTU.
Title	WAN DDS Deployment No IP Fragmentation
Status	<Validated>

Rationale	For efficient use of the network, it is important to limit the loss rate of data samples because lost data samples are resent what results in higher bandwidth usage and, therefore, in higher costs. This requires control of Maximum Transmission Unit (MTU) within IP based networks and avoid IP fragmentation (for security reasons, many firewalls block IP fragments, losing one single IP fragment on the WAN results in resending the entire UDP datagram). Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN). For architectural aspects and terminology refer to latest 14.01.03 TAD.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP Core>
Domain of interest	<ICD><SLA>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0550
Requirement	The DDS implementation shall support compression of data samples in an interoperable way.
Title	Interoperable Compression at DDS level
Status	<In Progress>
Rationale	Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN). For architectural aspects and terminology refer to latest 14.01.03 TAD.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP Core>
Domain of interest	<ICD><SLA>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>

Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0555
Requirement	When compression is enabled, the DDS implementation shall compress data samples before any DDS fragmentation.
Title	Compression before DDS level fragmentation
Status	<In Progress>
Rationale	It is more efficient to compress a data sample then fragment it for transfer on the network. The reverse may generate too many small packets on the network. Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN). For architectural aspects and terminology refer to latest 14.01.03 TAD.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP Core>
Domain of interest	<ICD><SLA>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>

<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0560
Requirement	The DDS implementation shall be able to adapt the publication rate to the bandwidth of the Wide Area Network.
Title	DDS Publication rate adapted to the WAN bandwidth
Status	<In Progress>
Rationale	It is required to adapt publication rate to the bandwidth of the Wide Area Network which avoids bursts and subsequent resends of lost data samples. Some DDS vendors already provide some support for bandwidth limitation at the DDS level; but this is not very common. Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN). For architectural aspects and terminology refer to latest 14.01.03 TAD.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP Core>
Domain of interest	<ICD><SLA>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0565
Requirement	When underlying network supports multiple classes of service, the DDS implementation shall be able to publish data samples according to selected classes of services.
Title	Publishing data with priority
Status	<In Progress>
Rationale	Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN).

	Candidate DDS QoS: TRANSPORT_PRIORITY For architectural aspects and terminology refer to latest 14.01.03 TAD.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><SLA>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0570
Requirement	Adding a DataReader replica in a SWIM Node shall minimize additional communication and data transfer on the network.
Title	Limit impact of local redundancy
Status	<Validated>
Rationale	Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN). Adding redundancy at a specific location (or SWIM node) introduces new DDS Readers/Writers to the overall DDS domain(s). Starting-up of a DataReader replica will result in all DataWriters publishing their latest data samples (combination of reliability, durability and destination order QoS) to align the new DataReader. This will result in a high volume of data at start-up to be transmitted over the network (WAN). Adding a local replica will impact performance of all nodes with DataWriters. In order to implement the replica mechanism, data must be available locally (at the master/primary server). Feeding the new data reader with existing data shall be done locally using some Node/Local-level State Transfer. For architectural aspects and terminology refer to latest 14.01.03 TAD.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP Core>

Domain of interest	<ICD><SLA>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0575
Requirement	The SWIM-TI Messaging providing the FlightObjectDistribution shall ensure atomicity of FO updates.
Title	Atomic Flight Object Update
Status	<Validated>
Rationale	One update is performed completely or not at all. Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN). For architectural aspects and terminology refer to latest 14.01.03 TAD.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><SLA>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A

<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0580
Requirement	The DDS implementation shall provide, scalable, efficient and interoperable discovery protocol supporting PIM-SSM and minimising discovery traffic.
Title	Efficient DDS Discovery
Status	<In Progress>
Rationale	For scalability, a hierarchical architecture is preferable in order to decrease the exchange of discovery or heartbeat messages between all DDS participants. Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN). For architectural aspects and terminology refer to latest 14.01.03 TAD.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP Core>
Domain of interest	<ICD><SLA>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0585
Requirement	The DDS multicast locators shall support multicast addresses in the ranges of SSM allocated addresses as defined in RFC 4607.

Title	SSM multicast locators
Status	<Validated>
Rationale	IP version 4 (IPv4) addresses are in the 232/8 (232.0.0.0 to 232.255.255.255) range. For IP version 6 (IPv6), addresses are in the FF3x::/32 range. Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN). For architectural aspects and terminology refer to latest 14.01.03 TAD. Prototyping feedback: SSM testing campaign showed that current DDS implementations did not fall back transparently and 'smoothly' to unicast when configuration problems prevented correct working of SSM. Peers were configured to also support DDS unicast locators and DDS uses unicast of ACK/NACK.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP Core>
Domain of interest	<ICD><SLA>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0590
Requirement	The DDS implementation shall provide per data instance subscription in an interoperable way.
Title	Instance-level subscriptions support
Status	<In Progress>
Rationale	Limit visibility over DDS Topics. Solutions shall perform any required filtering at the source level (Writer side) to minimise publications over the network. Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN).

	For architectural aspects and terminology refer to latest 14.01.03 TAD.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP Core>
Domain of interest	<ICD><SLA>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0595
Requirement	The DDS implementation shall allow mapping of a DDS partition to a multicast address.
Title	Support for network partitions
Status	<In Progress>
Rationale	Mapping of DDS partitions to SSM multicast addresses allows finer control of network communication paths. Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN). The term multicast is intended as IP Multicast. For architectural aspects and terminology refer to latest 14.01.03 TAD.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP Core>
Domain of interest	<ICD><SLA>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0581
Requirement	The DDS heartbeat data shall not cross FO routers.
Title	Hierarchical Discovery
Status	<In Progress>
Rationale	For scalability, a hierarchical architecture is preferable in order to decrease the exchange of discovery or heartbeat messages between all (DDS) participants. Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN). For architectural aspects and terminology refer to latest 14.01.03 TAD.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><SLA>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

[REQ]

Identifier	REQ-14.01.04-TS-0901.0582
Requirement	The FO Router shall support DDS Simple Endpoint and Discovery Protocol for discovering local FO Nodes.
Title	SEDP support
Status	<In Progress>
Rationale	Current discovery protocol can be used as is behind FO Routers. Requirement identified in the context of the "Flight Object Overlay" that aims at providing an efficient and effective Flight Object distribution over Wide Area Network (WAN). For architectural aspects and terminology refer to latest 14.01.03 TAD.
Category	<Interface>
Validation Method	
Verification Method	<Review of Design><Test>
Profile Part	<BP FDD>
Domain of interest	<ICD><SLA><Function/Behaviour>
Point of view	<SWIM-TI provider>
Roles	<Subscriber><Publisher><Publication consumer><Publication mediator><Subscription handler>
Selfstanding set	<Not applicable>
Conformance	<No>
High Level	<No>
Testability	<Conformance testable>

[REQ Trace]

Relationship	Linked Element Type	Identifier	Compliance
<APPLIES_TO>	<Operational Focus Area>	ENB02.01.01	N/A
<ALLOCATED_TO>	<Project>	14.02.09	N/A
<ALLOCATED_TO>	<Functional block>	MSG	N/A
<ALLOCATED_TO>	<Functional block>	SO	N/A
<ALLOCATED_TO>	<Functional block>	Blue Profile	N/A
<SATISFIES>	<Enabler>	GGSWIM-51c	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-APS-05b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-01b	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05a	<Full>
<SATISFIES>	<Enabler>	SWIM-INFR-05b	<Full>
<SATISFIES>	<Enabler>	ER APP ATC 160	<Full>

4 Assumptions

1. The definition of a mapping that associates a Tier for each stakeholder in the Distribution List has to be provided by the IOP Application (P10.02.05).

5 References

[1] Template Toolbox 03.01.01
https://extranet.sesarju.eu/Programme%20Library/SESAR%20Template%20Toolbox.dot

[2] Requirements and V&V Guidelines 03.01.00
https://extranet.sesarju.eu/Programme%20Library/Requirements%20and%20VV%20Guidelines.doc

[3] Templates and Toolbox User Manual 03.01.00
https://extranet.sesarju.eu/Programme%20Library/Templates%20and%20Toolbox%20User%20Manual.doc

[4] EUROCONTROL ATM Lexicon
https://extranet.eurocontrol.int/http://atmlexicon.eurocontrol.int/en/index.php/SESAR

[5] SOA Service Interaction Security Patterns, http://soapatterns.org/masterlist_c.php

[6] Eurocae WG59, ED-133 Flight Object interoperability specification, June 2009.

[7] 14.02.09 D36-001, SWIM TI V3.0.0 Verification Report, November 2015.

[8] 14.02.09 D36-002, SWIM TI V3.0.1 Verification Report, April 2016.

[9] 14.02.09 D85. WP1.3.1 Preliminary Solution Evaluations Report - V2, March 2016.

[10] 08.03.10 D65 ISRM 2.0, June 2016.

[11] P14.01.04 Interface Evolution Analysis, December 2015.

[12] 14.01.04.D43-001, SWIM-TI Technical Specifications Catalogue 3.1, Edition 00.01.00, December 2015.

[13] 14.01.03 D30, SWIM Architectural Definition Final, Edition 00.01.00, June 2016.

[14] 14.01.03 D39 SWIM Profiles Final, Edition 00.01.00, June 2016.

[15] 14.01.04.D44-001, SWIM-TI Technical Specifications Catalogue, Edition 00.01.00, July 2016

[16] OMG The Real-time Publish-Subscribe Wire Protocol DDS Interoperability Wire Protocol Specification January 2009 http://www.omg.org/spec/DDSI/2.1/

[17] OMG Data Distribution Services, v1.2, January 2007 http://www.omg.org/spec/DDS/1.2/

[18] OMG-IDL: Interface Definition Language (IDL) version 3.5 http://www.omg.org/spec/IDL35/

[19] OMG DDS Security Specification V1.0 June 2014 http://www.omg.org/spec/DDS-SECURITY/1.0/Beta1

[20] W3C Recommendation SOAP Message Transmission Optimization Mechanism 25 January 2005 http://www.w3.org/TR/2005/REC-soap12-mtom-20050125/

[21] W3C Note, Simple Object Access Protocol (SOAP) 1.1 08 May 2000 http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[22] W3C Member Submission SOAP 1.1 Binding for MTOM 1.0 05 April 2006 http://www.w3.org/Submission/soap11mtom10/

[23] W3C Recommendation SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) 27 April 2007 http://www.w3.org/TR/soap12-part1/

[24] W3C Recommendation SOAP Version 1.2 Part 2: Adjuncts (Second Edition) 27 April 2007 http://www.w3.org/TR/2007/REC-soap12-part2-20070427/

[25] W3C Note Web Services Description Language (WSDL) 1.1 15 March 2001 http://www.w3.org/TR/wsdl

[26] W3C Recommendation Extensible Markup Language (XML) 1.0 (Fifth Edition) 26 November 2008 http://www.w3.org/TR/xml/

[27] W3C Recommendation Extensible Markup Language (XML) 1.0 (Fourth Edition) 16 August 2006, edited in place 29 September 2006 http://www.w3.org/TR/2006/REC-xml-20060816/

[28] W3C Recommendation XML Schema Part 1: Structures Second Edition 28 October 2004 http://www.w3.org/TR/xmlschema-1/

[29] W3C Recommendation XML Schema Part 2: Datatypes Second Edition 28 October 2004 http://www.w3.org/TR/xmlschema-2/

[30] IETF RFC 4510 Proposed Standard, Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map, June 2006 http://www.rfc-editor.org/rfc/rfc4510.txt

[31] IETF RFC 5280 Proposed Standard, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, May 2008 http://www.rfc-editor.org/rfc/rfc5280.txt

[32] IETF RFC 4523 Proposed Standard, Lightweight Directory Access Protocol (LDAP) Schema Definitions for X.509 Certificates, June 2006 http://www.rfc-editor.org/rfc/rfc4523.txt

[33] IETF RFC 5055 Proposed Standard, Server-Based Certificate Validation Protocol (SCVP), December 2007 http://www.rfc-editor.org/rfc/rfc5055.txt

[34]IETF RFC 2236 Internet Group Management Protocol, Version 2 November 1997 http://tools.ietf.org/html/rfc2236

[35]IETF RFC 3376 Internet Group Management Protocol, Version 3 October 2002 http://tools.ietf.org/html/rfc3376

[36]IETF RFC 2710 Multicast Listener Discovery (MLD) for IPv6, October 1999 http://tools.ietf.org/html/rfc2710

[37]IETF RFC 3810 Multicast Listener Discovery Version 2 (MLDv2) for IPv6, June 2004 http://tools.ietf.org/html/rfc3810

[38]IETF Proposed Standard Source-Specific Multicast for IP August 2006 http://tools.ietf.org/html/rfc4607

[39]IETF Draft Standard IP Version 6 Addressing Architecture February 2006 http://tools.ietf.org/html/rfc4291

[40]IETF Proposed Standard Unicast-Prefix-based IPv6 Multicast Addresses August 2002 http://tools.ietf.org/html/rfc3306

[41]IETF RFC 6176 Prohibiting Secure Sockets Layer (SSL) Version 2.0 March 2011 http://tools.ietf.org/html/rfc6176

[42]IETF RFC 793 Transmission Control Protocol September 1981 http://tools.ietf.org/html/rfc793

[43]IETF RFC 768 User Datagram Protocol 28 August 1980 http://tools.ietf.org/html/rfc768

[44]IETF RFC 791 Internet Protocol September 1981 http://tools.ietf.org/html/rfc791

[45]IETF RFC 2460 Internet Protocol, Version 6 (IPv6) Specification December 1998 http://tools.ietf.org/html/rfc2460

[46]IETF RFC 2246 The TLS Protocol Version 1.0 January 1999 http://tools.ietf.org/html/rfc2246

[47]IETF 2616 Hypertext Transfer Protocol -- HTTP/1.1 June 1999 http://tools.ietf.org/html/rfc2616

[48]IETF informational RFC 2818 HTTP Over TLS May 2000 http://tools.ietf.org/html/rfc2818

[49]OASIS WSI Basic Profile Version 1.2, Final Material, 2010-11-09 http://ws-i.org/profiles/basicprofile-1.2-2010-11-09.html

[50]IETF RFC 6960 X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP June 2013 http://tools.ietf.org/html/rfc6960

[51]IETF RFC 4158 Internet X.509 Public Key Infrastructure: Certification Path Building September 2005 http://tools.ietf.org/html/rfc4158

[52]IETF RFC 6434, Memo, IPv6 Node Requirements, December 2011 http://tools.ietf.org/html/rfc6434

[53]IETF RFC 792 Internet Standard, INTERNET CONTROL MESSAGE PROTOCOL, September 1981 http://tools.ietf.org/html/rfc792

[54]IETF RFC 950, Internet Standard, Internet Standard Subnetting Procedure, August 1985 http://tools.ietf.org/html/rfc950

[55]IETF RFC 6918 Proposed Standard, Formally Deprecating Some ICMPv4 Message Types, April 2013 http://tools.ietf.org/html/rfc6918

[56]IETF 5905 Proposed Standard, Network Time Protocol Version 4: Protocol and Algorithms Specification https://tools.ietf.org/html/rfc5905.

[57]OMG Standard Extensible and Dynamic Topic Types for DDS (DDS-XTypes) V1.1 November 2014 http://www.omg.org/spec/DDS-XTypes/1.1.

[58]IETF RFC 4033 Domain Name System Security Extensions (DNSSEC), March 2005 https://tools.ietf.org/html/rfc4033.

[59]Security Requirements for Cryptographic Modules US Federal Information Processing Standard (FIPS 140-2), May 2001 http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf.

[60]Efficient XML Interchange (EXI) Format 1.0 (Second Edition), Recommendation, 11 February 2014. http://www.w3.org/TR/2014/REC-exi-20140211/.

[61] IETF RFC 7568 Deprecating Secure Sockets Layer Version 3.0,June 2015 https://tools.ietf.org/html/rfc7568.

-END OF DOCUMENT-