Co-financed by the European Union
Connecting Europe Facility

Project full title

ATM Data "as a service"

Project acronym

ADaaS

Grant Agreement no.

INEA/CEF/TRAN/M2014/1026723

D2.4 Annex : OpenCWP Service Yellow SWIM Profile

Editor:	Kristof SCHIPPERS
Version:	1.0 (Released)
Dissemination Level[1]: PU	CO
Author(s):	Kristof SCHIPPERS Peter HENDRICKX
Due date of deliverable:	N/A
Actual submission date:	30.11.2017
Start date of project:	26.03.2015
Duration:	31.12.2017
Organisation name of lead contractor for this deliverable:	MUAC

Abstract: This document contains a service-provider independent definition of the OpenCWP service compliant with the Yellow Profile SWIM Specifications.

[1] **Project co-funded by the European Commission within the:**

PU – Public; PP - Restricted to other programme participants (including the Commission Services); RE - Restricted to a group specified by the consortium (including the Commission Services); CO - Confidential, only for members of the consortium (including the Commission Services).

Document Approbations

Authorship Level

Function	Name	Signature	Date
Author MUAC	Peter HENDRICKX		
Author MUAC	Kristof SCHIPPERS		

Co-Authorship

Function	Name
ATM/STR/SWM	Walter VAN HAMME
ATM/STR/SWM	Sofiia MIKOLENKO
ATM/STR/SWM	Francisco GRACIANI HIGUERO

Authorisation Level

Function	Name	Signature	Date
Head of International Affairs SCL	Žiga OGRIZEK		
Head of ENG MUAC	Peter NAETS		

History

Version	*Date*	*Modification reason*	*Modified by*
0.1.	17.11.2017	First version for Review	Authors/Co-Authors
0.2	01.12.2017	Modifications from Review	Peter HENDRICKX
0.3	07.12.2017	Added AIRM conformance information	Peter HENDRICKX
1.0	08.12.2017	Initial Released Version	Kristof SCHIPPERS

Table of contents

Executive Summary 6
1 Service Definition Identification 7
2 General Service Information 7
2.1 Service Identification 7
2.2 Service Provider 7
2.3 Service Categories 7
2.4 Service Standard Reference 8
2.5 Operational Needs 8
2.6 Service Functionality 10
2.7 Usage and Access Conditions 12
2.8 Non-functional requirements 12
2.9 Technical Constraint 13
2.10 Concepts 14
3 Service Interfaces 17
3.1 General Interfaces Information 17
3.2 SubscriptionManagement interface 23
3.3 DataDistribution interface 26
3.4 HeartbeatDistribution interface 42
3.5 FlightPlanDataManagement interface 44
3.6 SectorSpecificDataManagement interface 52
3.7 CorrelationManagement interface 57
3.8 FlightBrightManagement interface 59
3.9 SectorisationManagement interface 62
3.10 SsrCodeManagement interface 64
4 Exchanged Information 66
4.1 AIRM Conformance 66
4.2 General Message Information 66
4.3 SubscriptionManagement Messages 71
4.4 DataDistribution Messages 73
4.5 HeartbeatDistribution Messages 98
4.6 FlightPlanDataManagement messages 98
4.7 SectorSpecificDataManagement messages 105
4.8 CorrelationManagement messages 111
4.9 FlightBrightManagement messages 112
4.10 SectorisationManagement message 113
4.11 SsrCodeManagement message 114
5 Appendix: technical details 115
6 Abbreviations & Acronyms 122

7 Annex A: SWIM Specifications Compliance Matrix 125
8 Annex B: AIRM Mapping Spreadsheet 127

Executive Summary

This document contains a service-provider independent definition of the OpenCWP service.

The OpenCWP service decouples the FDPS from an ATM data service provider (ADSPs) from the CWPs from air traffic service units (ATSUs) through open and standardised service interfaces to foster ADSP/ATSU interoperability.

Its aim is to provide ATSUs with an interface that allows its own CWP working positions (or other systems that require ATM data as a service) to connect with a remotely located FDPS.

The OpenCWP service includes correlation, flight data distribution and flight data management, and supports sectorisation.

It is conformant with the (draft) EUROCONTROL SWIM specifications (compliance matrix in Annex)

- EUROCONTROL specification for SWIM Technical Infrastructure Yellow Profile
- EUROCONTROL specification for SWIM Information Definition
- EUROCONTROL specification for SWIM Service Description

The OpenCWP Service was successfully demonstrated within the ADaaS Workpackage 2 where Slovenia Control CWPs consumed MUAC FDPS Services.

1 Service Definition Identification

Title	OpenCWP Service Definition
Edition	1.0
Reference date	07 Dec 2017

2 General Service Information

2.1 Service Identification

Service Name	OpenCWP
Service Version	1.0
Service Abstract	The OpenCWP service decouples the FDPS from an ATM data service provider (ADSPs) from the CWPs from air traffic service units (ATSUs) through open and standardised service interfaces to foster ADSP/ATSU interoperability. Its aim is to provide ATSUs with an interface that allows its own CWP working positions (or other systems that require ATM data as a service) to connect with a remotely located FDPS. The OpenCWP service includes correlation, flight data distribution, flight data management, etc..

2.2 Service Provider

This document contains a service-provider independent definition of the OpenCWP service.

2.3 Service Categories

Categorisation of the service according to the PCP information exchange areas.

Information exchange area	Flight information exchange

Categorisation of the service according to the SESAR 1 Service Categorisation, see ISRM Portfolio, Appendix B. Only a subset of the facets have been used.

Facet	*Concept*	*Comment*
ATM Information	flight information air traffic operations information	
Flight Phase	execution planning	
Intended Provider	Air Navigation Service Provider	ATM Data Service Provider

Facet	*Concept*	*Comment*
Intended Consumer	Air Navigation Service Provider	Controller Working Position
ATM Capability	Service Delivery Management	

2.4 Service Standard Reference

The OpenCWP service does not adhere to an existing service standard.

The OpenCWP service definition is expected to become a reference for concrete implementations.

2.5 Operational Needs

2.5.1 Operational Needs

ADaaS (ATM Data as a Service) refers to the concept of building a cross-vendor interoperable system architecture, where air traffic service units (ATSUs) are decoupled from ATM data service providers (ADSPs) through open and standardised interfaces. The concept is built upon the principle that data is accessed and/or updated by multiple (remotely located) users while ensuring a single point for updates, and as such eliminates redundancy and streamlines costs by centralising data in one entity. It paves the way for further harmonisation in air traffic management and helps alleviate the de-fragmentation of the European network, as required by the Single European Sky.
The key to such an approach is the utilisation of system-wide information management (SWIM) compliant data exchange protocols and open service oriented architecture (SOA) as vital elements, in order to facilitate the seamless integration of future applications.

In order to facilitate an accelerated development and deployment of the ADaaS concept, MUAC offers the OpenCWP service definition describing an open service interface that can be implemented in various multi-vendor scenarios. The current operational MUAC FDPS-CWP interface has been used as a starting point; based on good engineering practices its implementation already accommodates an interface between a data centre and an OPS room using different technologies and manufactured by different suppliers. Additionally, its interface design rationale has not changed since its inception, but has proven in practice that a continuous and flexible extension of its payload contents is possible; thereby not only demonstrating the feasibility to define services, but also demonstrate its maturity and the many aspects needed for commissioning of such services (e.g. performance, safety, security, etc.).

2.5.2 Information Exchange Requirements

The identification of the needs of the service is not based on Information Exchange Requirements (IER). See next section for an overview of the process being followed.

2.5.3 Service identification and modelling

The main objective of the OpenCWP service definition document is to represent services offered by one or more ADSPs that are consumed by remotely located CWPs for operational usage.
In order to establish such standard interface, the current operational MUAC FDPS-CWP interface has been used as a starting point. Its implementation is already proven in practice, and accommodates an interface between a data provider and CWP that are both using different technologies, different concept of operations, and were manufactured by different providers based on good engineering practices.
The latter interface can already be considered as service oriented, although it does not provide the fine-course granularity in terms of the number of interfaces offered. For the ADaaS project, the MUAC interface principle is proposed to be further split-up in order to allow a wide number of

interfaces and operations, tailor the messages and their content, while ensuring loose coupling to the furthest extend possible between provider (FDPS) and consumer (CWP). Within this context, the connector box acts as access point for both entities, serves as a message broker, and ensures consistency in terms of exchange protocols and message formats.
All required interfaces and operations have been identified during several face-to-face meetings between operational and technical representatives of MUAC and Slovenia Control. Identification started from a logical architecture point of view, followed by a transposition into a set of operations and messages (section 3.1.3 Interface overview diagram), and finally the definition of the actual message payload (section 4 Exchanged Information). The SESAR B.04.04 Workstation, Operational and service modelling Report [D07] served here as an important guidance in the service modelling of the different interfaces.
Based on this exercise, the functionalities provided by the MUAC operational FDPS, and the operational requirements [URD] set by Slovenia Control, a first set of essential information elements that are required for the OpenCWP service were identified:

- Flight plan data
- Trajectory information, including sector sequence information
- Coordination & transfer information
- Correlation information
- Safety net (STCA) information
- Application monitoring (heartbeat mechanism)
- QNH and transition level information
- Map data information

-

2.6 Service Functionality

Functionality	*Real world effect*
Subscription Management allows service consumers to subscribe or unsubscribe to the distribution of a series of information areas. The following information areas can be subscribed to: • flightplan-data • sector-specific-data • safety-net • flightplan-monitoring • datalink • correlation • heartbeat • flight-bright • mtcd-service • system-information • strategic-constraint • sectorisation-data • ssr-code • meteo-data • runway-info • special-area • map-data	Not applicable.
Data Distribution ensures that each service consumer is provided with the latest up-to-date information relative to the subscribed information areas.	Not applicable.
Heartbeat Distribution supports the service provider and the service consumer to monitor each other's status.	Not applicable.
Management of Flight Plan Data allows service consumers to provide the correct & latest up-to-date controller information, create, modifiy, udate ASPL or SFPL, Submit requests about instructions/clearances given to the flight crew (e.g. DCT, CFL, NFL, speed, heading, …), Change status information regarding the flight's airframe (e.g. no FSSA, RVSM status, …), Etc.	Any instruction communicated to the pilot (either via voice or datalink) are expected to be executed by the aircrew. Furthermore, such information is communicated towards downstream partners and the network manager.

Functionality	*Real world effect*
Management of Sector Specific Data allows service consumers to provide the correct & latest up-to-date controller information regarding sector-specific information and coordination & transfer information, such as taking control of a flight (or proposing hand-over, request-on-frequency, etc.), Change the coordinated entry and exit levels, Deliver departure clearance for an flight departing from an internal aerodrome, Skip and cancel-skip of an internal sector, Bypass and cancel-bypass of the 1st downstream internal sector, Delegate the flight to another internal sector, Change the next downstream internal sector into the preferred one, Change the entry/exit frequency of sectors, Etc.	Upon communication of the SSR code to the aircrew, the aircraft transponder will emit that code. Furthermore, the present/next SSR codes allocated to the flight-plan are communicated to downstream partners interested in the flight.
Management of Correlation allows service consumers to provide inputs related to the linkage of flight plans with tracks, such as link a flight plan with and unlink a flight plan from a specific track, set the present, next or downstream SSR code a flight.	Correlation establishment, and the resulting identification of the current aircraft position related to the flight plan are communicated towards the network manager.
Management of Flight Bright allows service consumers to provide inputs related to highlight of a track or flight plan, such as adding or removing SSR codes or Mode S callsigns to / from the Bright function for his OPS sector; adding or removing a Flight to / from the Bright function for his OPS sector;or another internal OPS Sector; point a flight to an external flight sector / centre	The management of flight bright does not have a real-world effect.
Management of Sectorisation allows service consumers to perform a re-sectorisation change, such as verifying a new sectorisation change (would it be valid if performed), performing a sectorisation change.	Re-sectorisation changes in the configuration of the air traffic control unit result in the real-world effect that air traffic is handled in a more efficient way (in accordance with the flow of traffic).
Management of SSR code allows service consumers to reserve an SSR code for manual assignment later on (i.e. manual assignment by using the Operation setSsr, see section 3.7.4), and to clear such code from display in the whole OPS sector. An SSR code reserved for manual assignment is not available for automatic assignment. The SSR code will be reserved during a design parameter time and then released if not manually assigned to any flight plan or released according to the standard release rules if manually assigned to a flight plan during this design parameter time.	The management of SSR code does not have a direct real-world effect. Its effect is reflected via the aforementioned management of correlation when the SSR code is communicated to the aircrew.

2.7 Usage and Access Conditions

Usage and access conditions have to be agreed between the service provider and the service consumer.

Example of restrictions in usage and access of the OpenCWP service:

- Identified users are granted permission to access and use the OpenCWP service according the Terms of Use, provided that:
  - they agree not to distribute any part of the delivered data received from the OpenCwp service, without prior written authorization from the service provider.
  - they agree not to use the OpenCWP service for any commercial use unrelated to ther service provider's business interests without the prior written authorization of the service provider.
    - Prohibited commercial uses includes any of the following actions taken without the service provider's express approval:
    - sale of access to the OpenCWP service;
    - sale of the data delivered via the OpenCWP service.
  - Prohibited commercial uses do not include any use that the service provider expressly authorizes in writing.

2.8 Non-functional requirements

Maximum time of Delivery	Maximum time of delivery depends on the service interface type: • All functions receiving or producing traffic or flight related data. All functions producing system warnings or system errors. Less than 350ms. • Functions sending and/or receiving management information and/or maps including cartographical and aeronautical data. Less than 500ms. • Functions sending and/or receiving statistical information and/or predicted airspace visual representations. Less than 1000ms.
Reliable messaging	Supported through configuration of Apache AMQ Broker.
Authentication	XYZ certificate used for mutual authentication.
Authorisation	Based on identities retrieved from XYZ certificates. Multiple clients per organization are authorized to consume the service at a given point in time. If a connection to the broker (rather than login request) is attempted by a client application while another one (from the same organization) is already established, the connection (or login request) will be refused.
Service Availability	The underlying infrastructure to accommodate the OpenCWP service is developed for continuous operational use (24 hours per day, 7 days per week).
Service Performance	Service restart time is required to be less than 5 minutes.
Subscription persistency	Organizations will be considered as subscribed to the service until: • a "logout" is performed,

	• a design parameter of number of heartbeats are missed from the consumer. Upon these events, the subscription will be terminated. The client is expected to re-subscribe at broker level upon broker subscription expiration.
Transport level integrity and confidentiality	Transport level integrity and confidentiality is achieved via IPsec (*AES256*).
Message persistency	OpenCWP service does not support message persistency. Consumers are ensured that latest available and up-to-date data is received.

2.9 Technical Constraint

The service instance is expected to be running at the service consumer site in a fully secure environment.

Other possible constraints would be part of a service level agreement to be established and agreed between the service provider and the service consumer.

2.10 Concepts

2.10.1 Initialisation & distribution of messages

As a general principle, the Connector Box is the system instance that governs the initialisation towards the remote clients. The MUAC FDPS initialisation capabilities are thus de-coupled from the remote connecting clients.

Initialisation is performed by requesting a full transmission of all data on a dedicated channel for the requestor only so as not to be disturbed by initialisations of other clients at the same time[2]. The connector box is arranging the initialisation, and the client is not expected to include any logic (e.g. sequence number checking). Figure 1 represents the general algorithm inside the connector box to provide initialisation data to its customers. Whenever a client subscribes to an interface, the internal flight plan data store is copied (indicated in blue). All required data messages (supported by the requested interface(s)) for the initialisation are then emitted towards the client. Any update which is arriving in the meantime is distributed instantaneously to the up-and-running clients (i.e. the ones not initialising), while the updates for initialising clients are buffered until all initialisation data has been sent. After the initialisation is completed, the buffered updates are sent.

For the updates, there are two types of data distribution: cyclic updates and updates on create/update/cancel for the non-cyclic distribution. Chapters 3 and 4 describe all interfaces and messages and their type of distribution supported by the OpenCwp service. It's important to note that as a general principle, the majority of the data is sent only when changed (i.e. only the "delta" with respect to the previous message of the same type is distributed – see section 5.1.9).

2 Bear in mind this is so at the logical level, at the network level bandwidth is shared.

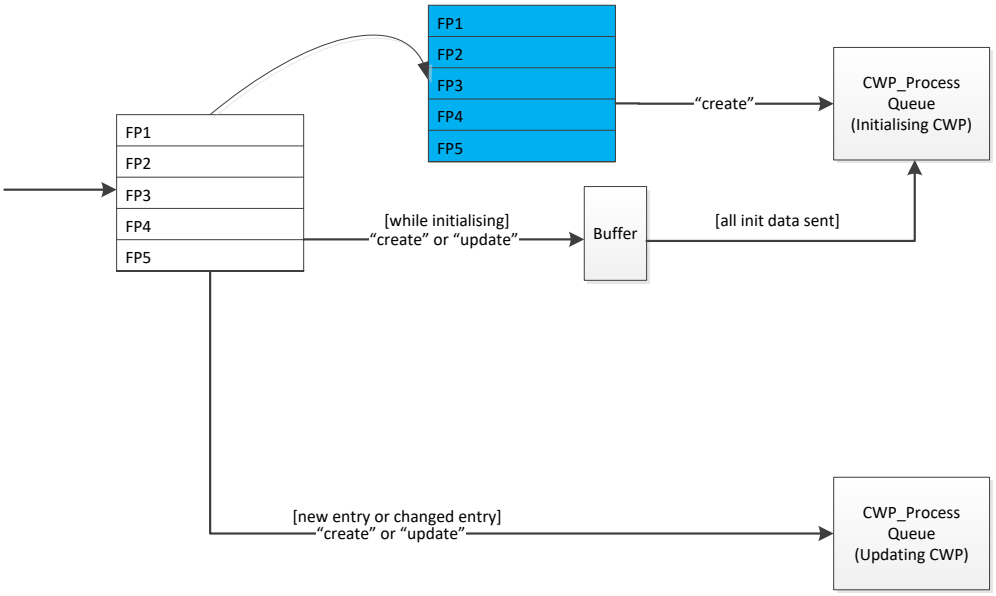


Figure 1: Initialisation logic (internal) of the connector box

Note that some of the offered OpenCwp interfaces are inter-dependent: the provision of information by one interface is required prior to the usage of data offered by another interface. Therefore, when the client is requesting subscription to multiple interfaces at the same time, a clear sequence is established from the server side (connector box) towards the clients in order to allow the latter to properly process the messages/operations in one-go (without requiring to "buffer" messages).
In case the client would request interfaces without requesting the dependant interface(s), if any, then the operation will be rejected. As such, a client should always consume the dependent interfaces (i.e. the consumer is responsible to request the applicable interfaces that are required for the client to construct its own reliable picture). Figure 2 represents the service tree of the OpenCWP service.

The table below list all messages to which all identified information elements are allocated as well as the sequence in which they must be sent in the initialisation cycle and following the creation of a new flight plan.

Message(s)	Sequence
FlightPlanDataMessage	1
SectorSpecificDataMessage	2
FlightPlanMonitoringMessage	3
MtcdConflictMessage	4
MtcdInformationMessage	4
DatalinkMessage	5
FlightBrightMessage	6

Table 1: Priority order upon initialisation

Additionally, processing of messages may involve updates of information sent before. Such information is identified by message items called keys. The following table identifies the messages that are keyed and lists the corresponding key(s):

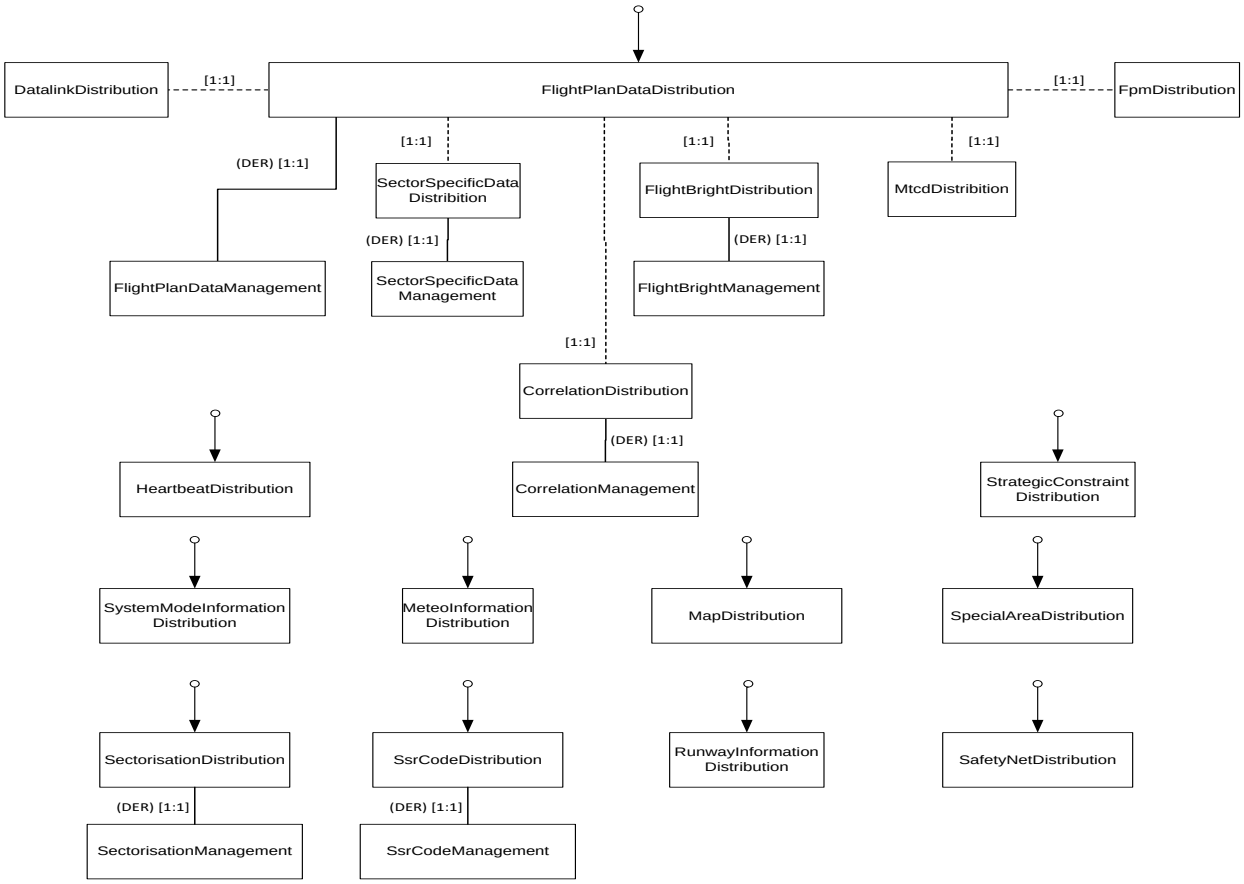


Figure 2: Service Tree

Message(s)	Key(s)
FlightPlanDataMessage	plan-identifier
SectorSpecificDataMessage	plan-identifier
	range-id
FlightPlanMonitoringMessage	plan-identifier
DatalinkMessage	plan-identifier
CorrelationMessage	plan-identifier
	track-number
FlightBrightMessage	plan-identifier
MtcdConflictMessage	mtcd-conflict-identifier
MtcdInformationMessage	plan-identifier
	mtcd-conflict-identifier

Table 2: Messages and their key(s)

Please note that the plan-identifier consists of at least a unique plan-number, flight-identifier or GUFI (or a combination of these). Further explanation of the subscription mechanism is provided in section 3.2 of this document.

3 Service Interfaces

3.1 General Interfaces Information

The operations exposed by the service have been grouped in 9 interfaces (see diagram below). These interfaces have been grouped in 2 groups: the Publish Subscribe interfaces and the Management interfaces.

3.1.1 Message Exchange Pattern

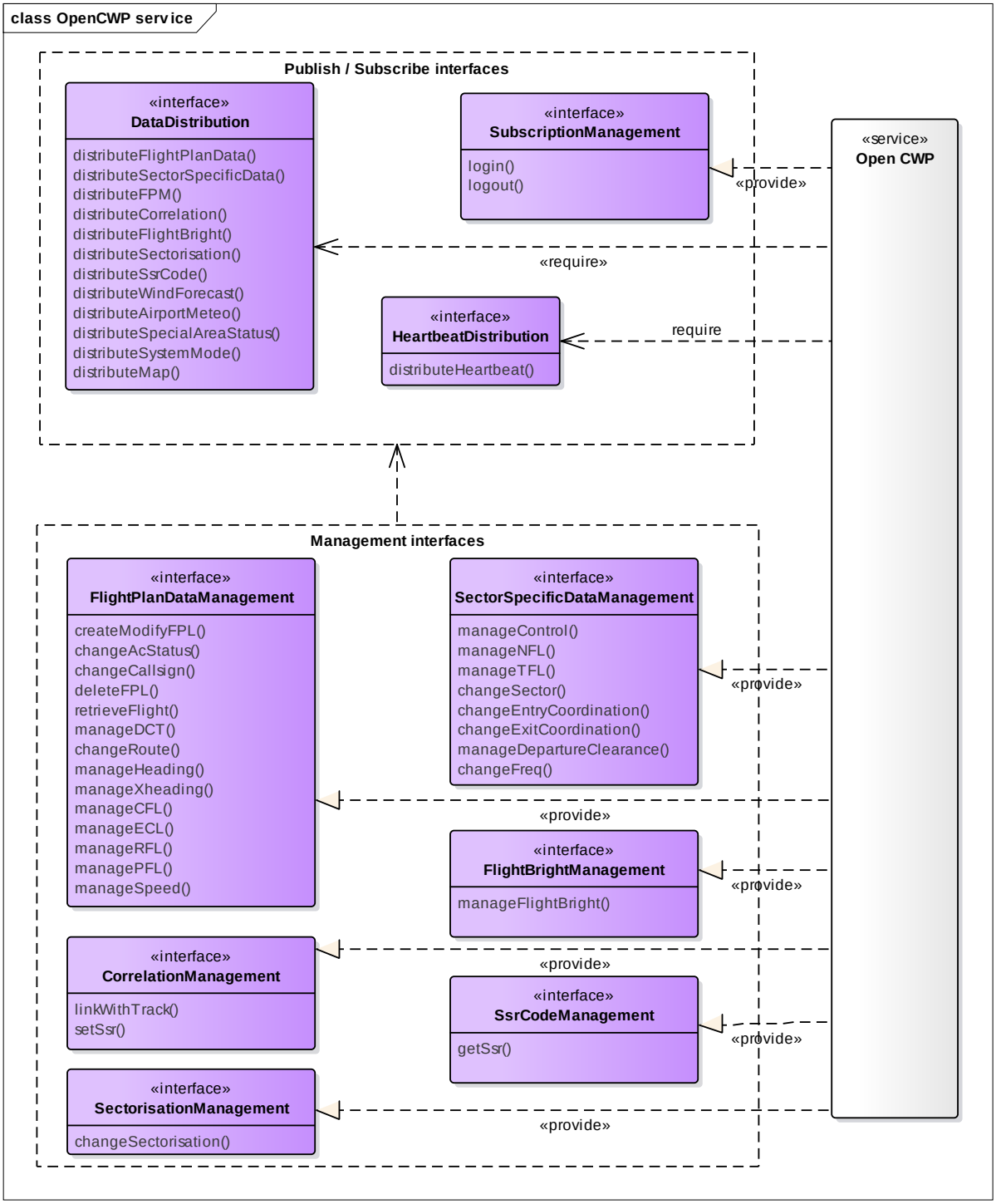
The service uses 2 message exchange patterns (MEP)

- the Publish Subscribe interfaces use Publish/Subscribe MEP. The publication is of type Push
- the Management interfaces use Request/Reply MEP .

3.1.2 Provider side and consumer side interfaces

All interfaces are provider side interfaces, except for the DataDistribution and HeartbeatDistribution interfaces which are consumer side interfaces.

3.1.3 Interface overview diagram



3.1.4 Dependency between interfaces

The interfaces have several dependencies.

The dependencies are described within the pre-requisite sections of each interface.

An overview of the dependencies can also be found in section “initialisation and distribution of messages).

3.1.5 Service Interface Binding

3.1.5.1 Selected binding

Profile Name	SWIM TI Yellow Profile
Profile Version	1.0
Selected Binding	AMQP Messaging
Additionally supported TI functional requirements	-

3.1.5.2 Network Connection

The network connection between the provider and consumer shall support IPv4 protocol, with encoding.

Depending on security demands, the connection can be either established over:

- Private VPN with SLA agreement of the network provider
- IPSec tunnelling (over internet)

3.1.5.3 Time Synchronisation

In order for the service provider and the service consumer to be time-synchronised, the NTP protocol is used.
Note: MUAC components use a MNTS server, while SCL components use their own time server.

3.1.5.4 Transport protocol

ActiveMQ supports multiple protocols:

AMQP 1.0
TCP

For the initial phase of the project, the OpenWire protocol has been selected as the transport protocol for all services. In the next step and once sufficient knowledge has been obtained on ActiveMQ, the AMQP (v1.0) protocol will be used for all services.

3.1.5.5 Technical supported protocols

Transport/Messaging Protocols	AMQP 1.0, TCP
Protocol Configuration	AMQP 1.0 ***content-type*** header to specify Media Type values.
Security	Authentication: Application level
Exception Handling	TBD

3.1.6 Machine Readable Service Interface

The ASN.1 and XSD definition files are embedded in the following packages:

3.1.7 Model View

No service model is available for this service.

The diagrams used to illustrate this document have been produced using modelling tools

- Service behaviour diagrams have been modelled using Visio
- Interface diagrams have been modelled using Sparx Enterprise Architect.

These are available as separate files

3.1.8 Service Behaviour

3.1.8.1 Nominal behaviour

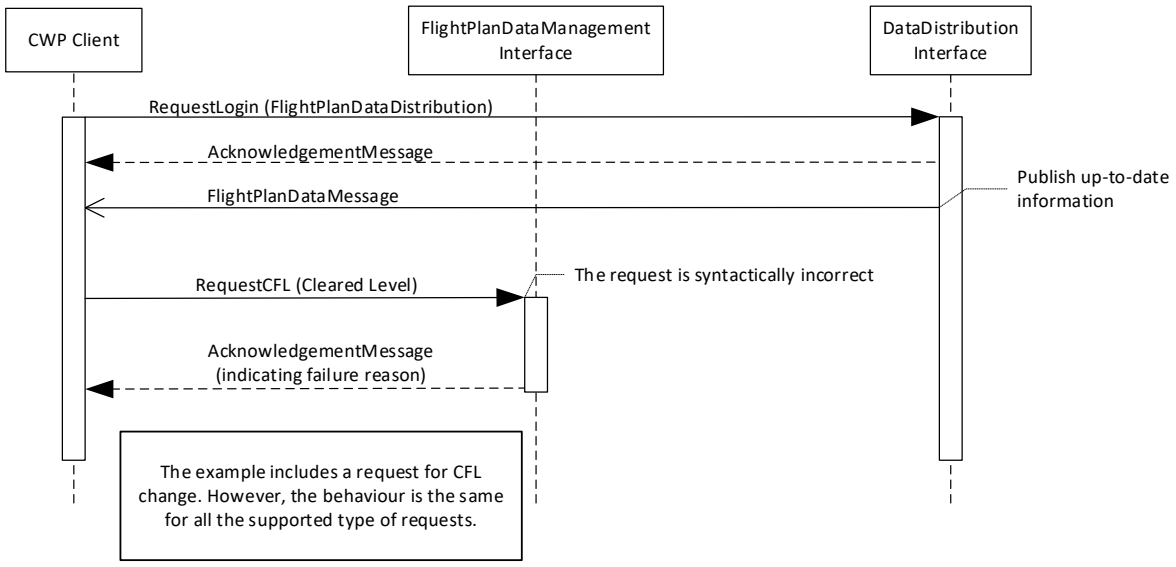
The service nominal behaviour is described in the Services Dynamic Behaviour sub-sections within each interface section.

3.1.8.2 Error cases

This section describes the behaviour of the OpenCWP service in terms of error handling.

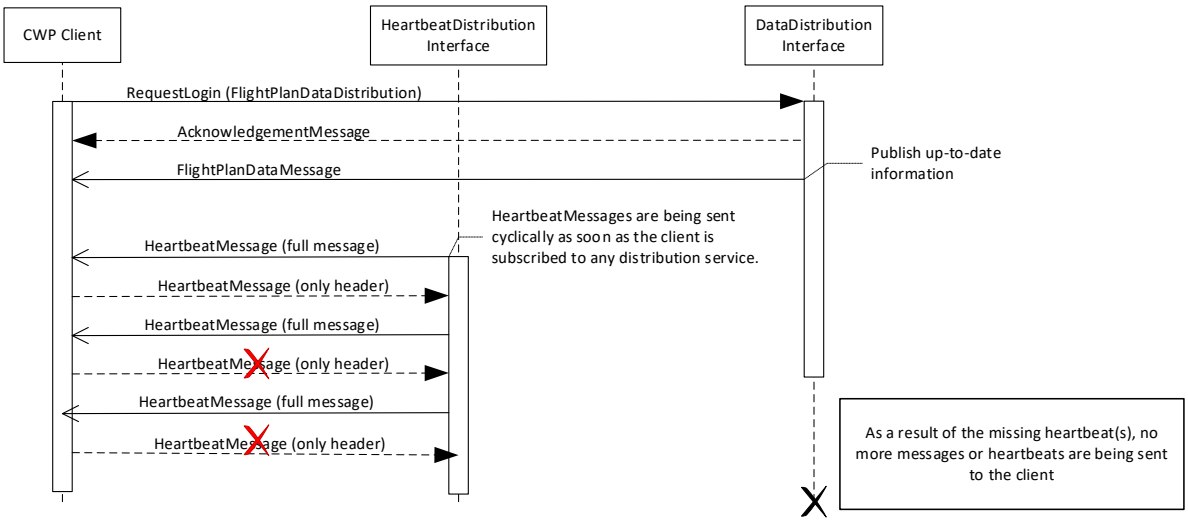
Following error use cases have been considered:

- A request is received from the consumer, which is not syntactically correct (and as such cannot be processed). An AcknowledgementMessage is returned with failure reason (see section XXX – AcknowledgementMessage).



Each input is first replied with the AcknowledgementMessage to indicate the acceptance or rejection of the request. The client is expected to start an internal timer in order to capture those cases where there would be no reply. In case of the latter, the consumer is expected to trigger a new input.

- Heartbeat messages are being lost between the consumer and provider (see section XXX – Operation distributeHeartbeat).



When more than a design parameter of subsequent heartbeat messages is missed, the connector box will remove the client from its configuration and no more message will be sent (the TCP/IP connection remains established). There is no pre-notification from the server-side to the customer-side for this event: the client is expected to perform heartbeat monitoring as well and a new RequestLogin is required in order to trigger a new initialisation.

- The physical connection between consumer and provider is lost.

  In case the physical connection is lost, the connector box will remove the corresponding consumer(s) from its configurations when not receiving a response in due time similar to the process as described in the previous point, however in this latter case also obviously the TCP/IP connection is lost.

3.1.9 Service Validation

The Maastricht Upper Area Control Centre (MUAC) established in 2013 the Shared ATS System (SAS), where a virtual centre network solution has been put into operational use, with one air navigation service provider offering shared ATM data services for the benefit of another ATSU in the core area of Europe. With the Shared ATS System the safety, efficiency and cost-effectiveness of a data service solution has been proven

The ADaaS Demonstrator, designed and developed in cooperation between MUAC and Slovenia Control, is composed of 3 phases:

- Phase 1: An ATM infrastructure is setup between MUAC and Slovenia Control where MUAC Controller Working Positions (CWP) installed in Slovenia Control are remotely connected to an FDPS instance in MUAC. The communication between the MUAC FDPS and CWPs is using the legacy interface. It is currently implemented and successful shadow operations have been conducted in June 2016.

- Phase2: The interface between FDPS and CWP is changed to an open interface and the Slovenia Control CWP is connected to the MUAC FDPS via this interface. The OpenCWP interface decouples the ATM data service provider (ADSPs) from the air traffic service units (ATSUs) through an open and standardised service interfaces to foster

ADSP/ATSU cross-vendor interoperability. Services include correlation, flight data distribution, flight data management, etc. and will be demonstrated in shadow operations in February 2017.

- Phase 3: The distributed architecture that allows remotely located data service providers to be completely synchronised is established. The identified solution(s) within the Target ADaaS Architecture will be experimentally established, in order to validate the assumptions and uncertainties of such architecture. Its feasibility will be demonstrated in shadow operations in November 2017.

3.1.10 Service Monitoring

See HeartbeatDistribution Interface.

3.1.11 Code Examples

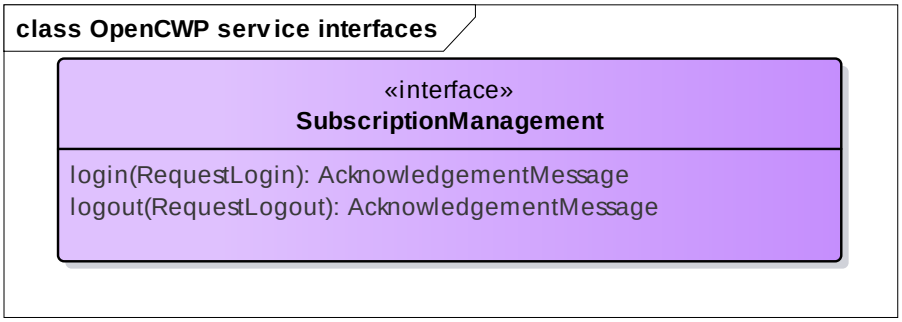
Find below a nice little program that explains the service consumer step-by-step on how to connect.

cb_example_01.tar

3.2 SubscriptionManagement interface

The following section provides further details on the general (un)subscription mechanism, as introduced in section 2.10.1 (Initialisation & distribution of messages).
The mechanism guarantees that any connecting CWP client receives the correct & latest up-to-date information for the service(s) it subscribes to. Additionally, it provides the means to unsubscribe to services.
As explained in section 2.10.1, all services implemented on the connector box are distributed over the CWP Process Queue (i.e. 1 queue per connected client). On the queue, only those messages related to the requested services are distributed. This is done to allow connecting clients to choose whichever service they want, and limit the amount of data traversed over the network.
The initialisation service is implemented by means of the following message:

- RequestLogin message (client requests subscription and initialisation to the connector box for one or more services)
- RequestLogout message (client requests to unsubscribe from all services)



3.2.1 Pre-requisites

The CWP Process Queue shall be unique per consumer. It is the consumer's responsibility to ensure this uniqueness.

3.2.2 Transmission Events

Subscription requests and corresponding reaction(s) are expected whenever a client requests for subscription to a service[3] and as such its frequency and event depends on the client's implementation (i.e. when does the connecting system request for subscription & initialisation). Initialisation dynamics are further explained in section 3.2.5.

3.2.3 Operation login

Description	The operation allows to: • Request client subscription and initialisation for one or more services, • Unsubscribe to one or more services.
Input	RequestLogin message
Output	AcknowledgementMessage

[3] Initialisation messages are not expected to be sent by the server side, without a prior initialisation request. In the case this would happen, the client is expected to ignore the messages.

3.2.4 Operation logout

Description	The operation allows requesting the un-subscription from all services.
Input	RequestLogout message
Output	AcknowledgementMessage

3.2.5 Service dynamic behaviour

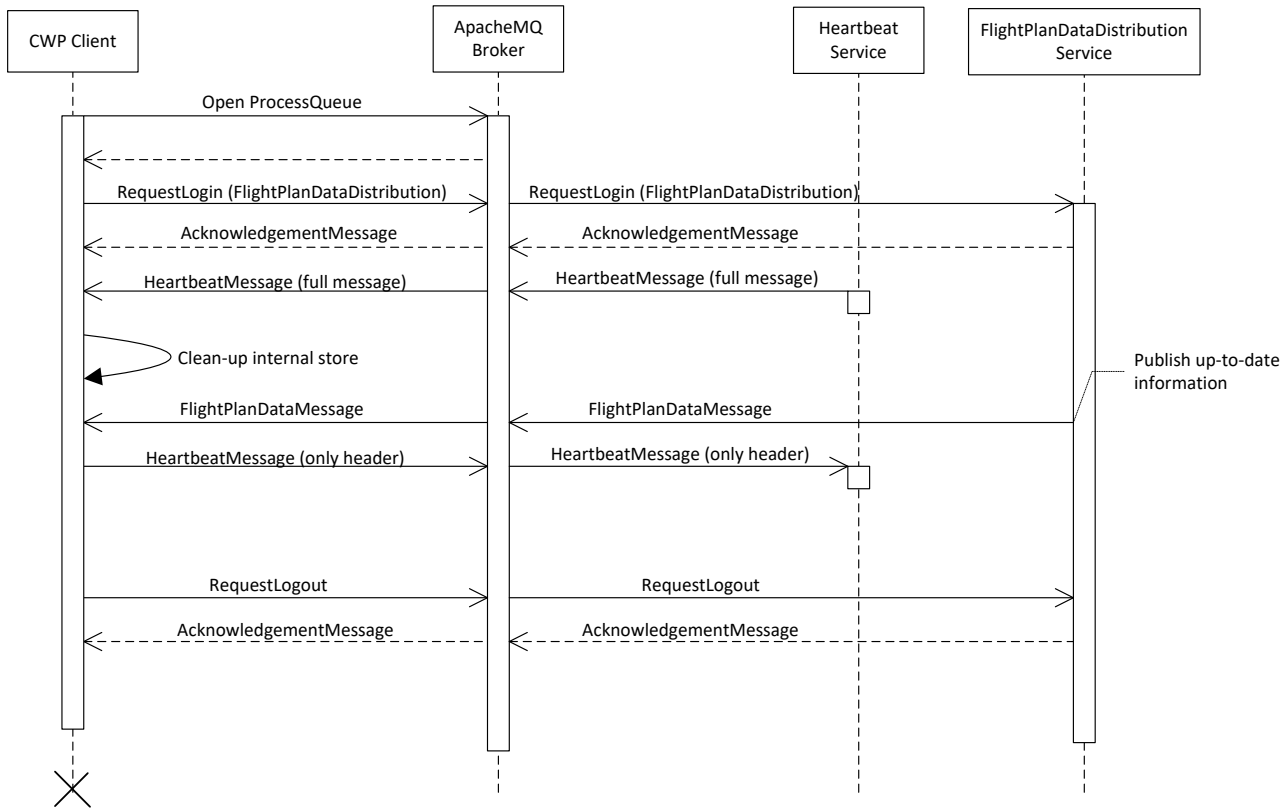


Figure 3: Sequence diagram for subscription to a service

Initialisation dynamics are explained in Figure 3 for the case where the client requests subscription and initialisation (Flight Plan Data Distribution service). In summary:

1. The client opens his dedicated CWP ProcessQueue.
2. The client sends a RequestLogin message to the dedicated CwpServer InputQueue residing at the connector box (and indicates its expected reply queue – i.e. CWP Process Queue). This establishes at the same time a subscription to the fixed CwpServer InputQueue in the broker.
3. The connector box transmits an acknowledgement message to the client over the CWP ProcessQueue (point-to-point), indicating the acceptance or rejection of the request.
4. The connector box starts sending cyclically heartbeat messages to the client, a reply is expected in due time from the client. At this stage, it is expected that the client has cleaned his internal stores[4].

[4] The client may trigger the clean-up of its internal stores at an earlier stage.

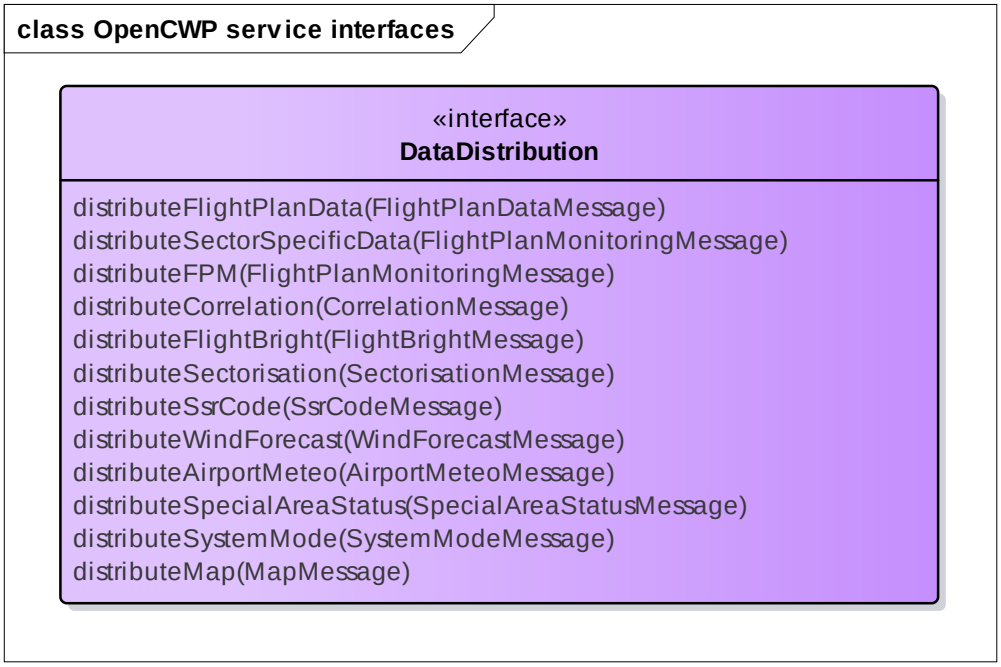
5. Provided the request for initialisation is accepted, the connector box sends all related messages associated to the services that the client requested data for via the CWP ProcessQueue.

To unsubscribe from all services with a single message, a RequestLogout message is expected (as explained in section 3.2 of this document).
Note 1: when a login request is sent to the connector box to request a service, the client is expected to clean all of its internal stores (this to avoid any mismatch between residing and newly received data at the client).
Note 2: In case there is another RequestLogin from a client (after having already correctly processed one), the connector box will accept the input, and treat it as a new initialisation request message. If in the new request the service which was previously requested is no longer appearing, the client will be unsubscribed from that service. As such, the RequestLogin can serve as well as a mechanism to unsubscribe to one or more services.
Note 3: If the RequestLogin message contains an unsubscription request for a service on which other services are dependant (see Figure 1 for the service tree), the client will also be unsubscribed from these dependant services.

3.3 DataDistribution interface

3.3.1 Pre-requisites

The client is subscribed to the corresponding data distribution (see login operation).



3.3.2 Operation distributeFlightPlanData

Description	The flight plan data distribution ensures that each client is provided with the latest up-to-date flight plan information detailing the following information: • General flight plan information • Clearances • 4D-Trajectory • Airspace crossing & sector sequence information • Entry & Exit coordination data for the current leg • Current (under-control or first) and next sector conditions • Basic correlation information Note that the current implementation of the service only includes the coordination information related to the current legs[5]. Downstream leg information may be provided at the same time by another service or by extending the definition of the current data type. For the 4D-Trajectory, airspace crossing and sector

[5] A leg is the part between the entry and exit of a flight in the area of interest of the considered airspace. In case of multiple entries and exits, then the service only provides relevant information for the 1st leg, until that part is no longer relevant.

	sequence information, the complete flight within the considered system's AoI will be covered.
Pre-requisites	None.
Transmission Events	Flight plan related information is distributed as soon as the flight plan data is available on the connector box (i.e. when FDPS has distributed it). Upon any event that causes an update (e.g. controller input reception of OLDI message, etc.) to the related flight, the data shall be distributed to the connected clients. In case of a cancellation/deletion of the flight, the clients shall be informed instantaneously. During initialisation, all flight plan data information covered by the flight plan data distribution service is distributed.
Input	FlightPlanDataMessage
Output	None.

3.3.2.1 Service dynamic behaviour

Figure 4 shows a sequence diagram to reflect the dynamic behaviour of the Flight Plan Data Distribution service to support a CWP to:

- Subscribe to the service,
- Receive up-to-date flight plan data information,
- Unsubscribe from the service.

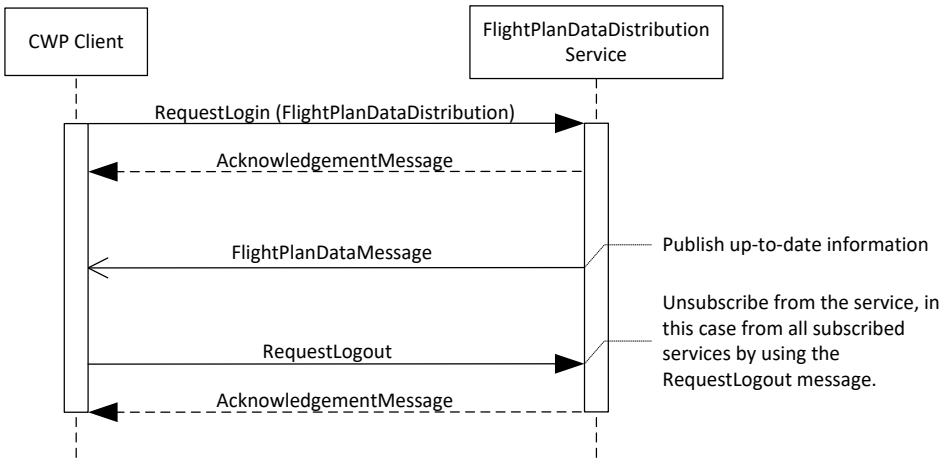


Figure 4: Flight Data Distribution service dynamics

3.3.3 Operation distributeSectorSpecificData

Description	The SectorSpecific data distribution ensures that each client is provided with the latest up-to-date OPS sector-specific information for a flight detailing the following information: • Sector status (normal, skipped, bypassed, etc.) • Entry, Internal & Exit coordination data for the OPS sector for which information is distributed.

	• Intra-sector dialogue information • Event trigger information (for display purposes) for the OPS sector t Important: the service will emit all sector-specific data for all flights over the service. It is the clients responsibility to filter out the relevant messages for him. The rationale of using such an approach is that it allows the usage of topics (multicast) at a later stage in the project without changing the client side functionality.
Pre-requisites	The client is subscribed to the Flight Plan Data Distribution service.
Transmission Events	The Sector Specific Data message is sent for each OPS sector, a time parameter prior the entry into the sector. Typically, the first message is received 30 minutes in advance, provided the FlightPlanData message has been distributed. Upon any event that causes an update (e.g. controller input reception of OLDI message, etc.) to the related flight, the data shall be distributed to the connected clients. In case of a cancellation/deletion of the flight, the clients shall be informed instantaneously. During initialisation, all sector specific data information covered by the sector specific data distribution service is broadcasted.
Input	SectorSpecificDataMessage
Output	None

Service dynamic behaviour

Figure 5 shows a sequence diagram to reflect the dynamic behaviour of the Sector Specific Data Distribution service to support a CWP to:

- Subscribe to the service,
- Unsubscribe from the service,
- Receive up-to-date OPS sector-specific information.

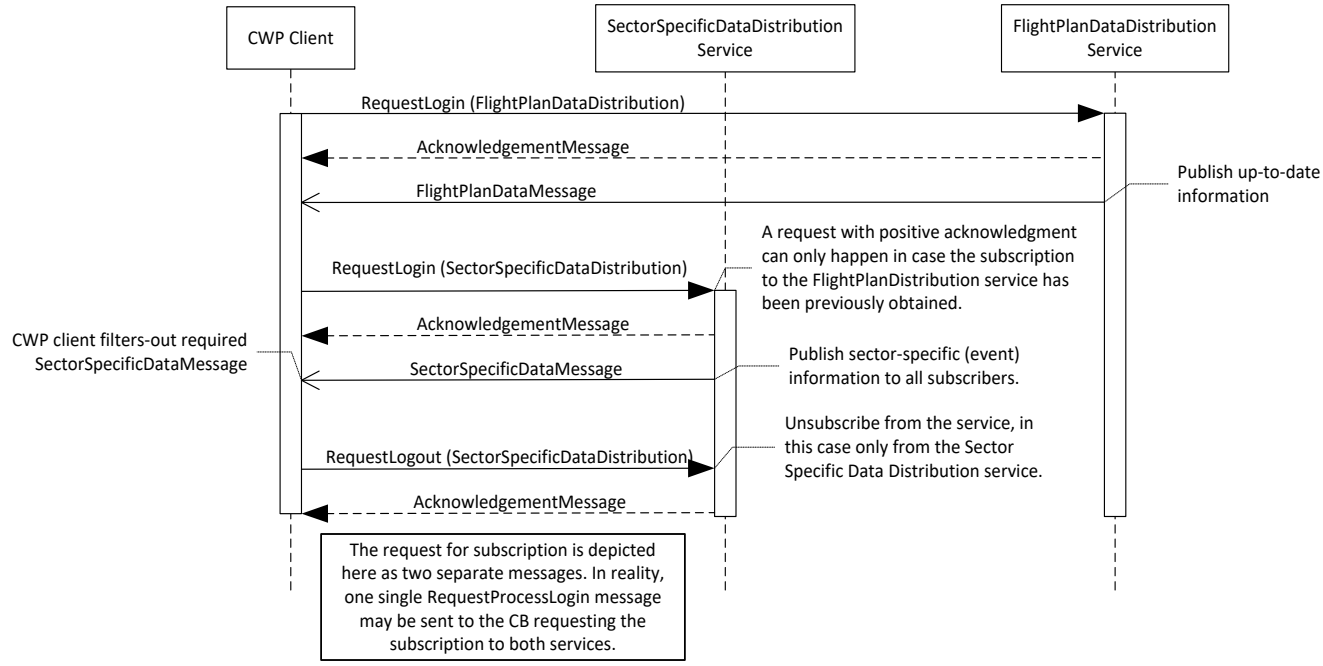


Figure 5: Sector Specific Data Distribution service dynamics

3.3.4 Operation distributeFlightPlanMonitoring

Description	This service provides all subscribed clients with all detected deviations (Lateral, Vertical and Longitudinal) between the trajectories cleared data and the track. Additionally, it provides an automatic re-routing proposal when available. The service is implemented by means of the FpmMessage. Conformance monitoring information is only provided for SFPLs, which are under-control and not in MANUAL sub-state. Re-routing information may already be received for SFPLs, which have been coordinated at entry (e.g. ACT received at the server, or manual coordination performed) and are not yet-under-control of the local centre.
Pre-requisites	The client is subscribed to the Flight Plan Data Distribution service.
Transmission Events	The FpmMessage provides lateral, vertical and longitudinal conformance information, and as well re-routing proposals in case of lateral un-conformance. The FpmMessage is not a cyclic message so it is only sent when any of its fields change or upon initialisation.
Input	FlightPlanMonitoringMessage
Output	None

Service dynamic behaviour

Figure 6 shows a sequence diagram to reflect the dynamic behaviour of the Flight Plan Monitoring Distribution service to support a CWP to:

- Subscribe to the service,

- Unsubscribe from the service,
- Receive up-to-date flight plan monitoring information.

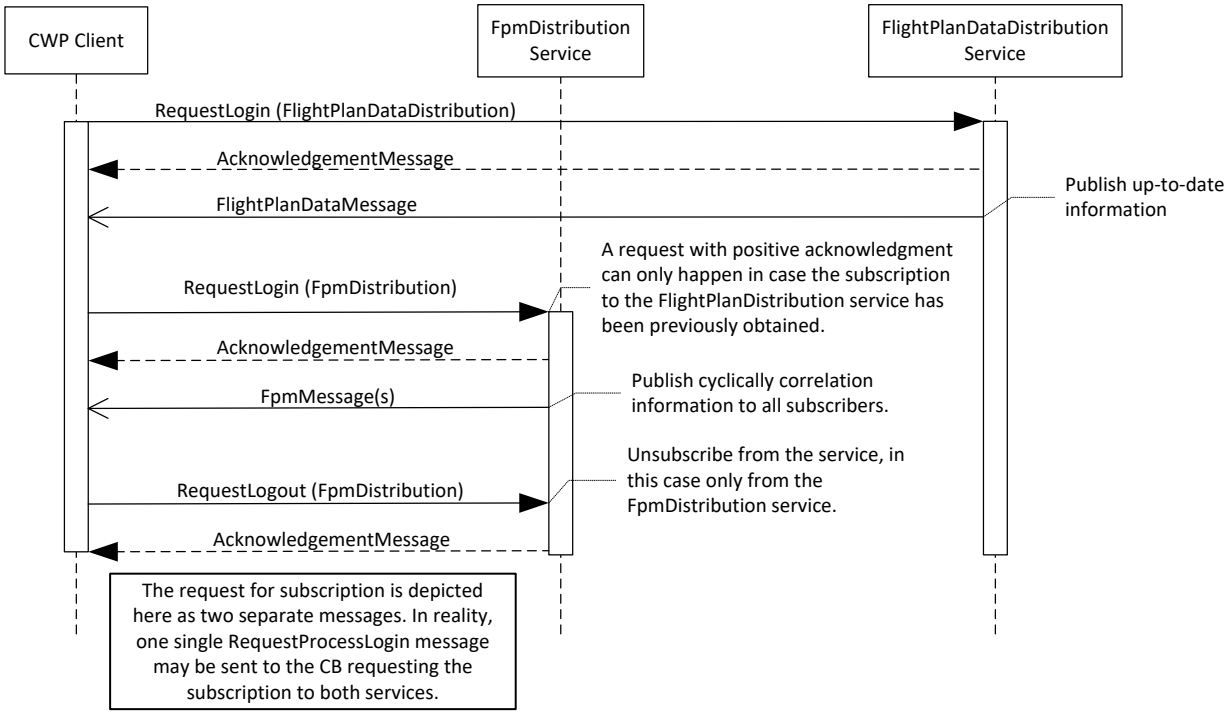


Figure 6: Flight Plan Monitoring Distribution service dynamics

3.3.5 Operation distributeCorrelation

Description	This service provides all subscribed clients with all detected deviations (Lateral, Vertical and Longitudinal) between the trajectories cleared data and the track. Additionally, it provides an automatic re-routing proposal when available. The service is implemented by means of the FpmMessage. Conformance monitoring information is only provided for SFPLs, which are under-control and not in MANUAL sub-state. Re-routing information may already be received for SFPLs, which have been coordinated at entry (e.g. ACT received at the server, or manual coordination performed) and are not yet-under-control of the local centre.
Pre-requisites	The client is subscribed to the Flight Plan Data Distribution service.
Transmission Events	Correlation information is provided cyclically by means of one or more CorrelationMessages, of which each of them can contain the linkage information for a certain amount (150) of correlated and uncorrelated flight plans. The data is provided cyclically (governed by the corr_cyclic_time design parameter). However, this parameter may be overridden by giving it as input parameter when performing the RequestLogin message when subscribing to the Correlation Service. CorrelationMessages are distributed for each correlated SFPL that has the flight plan state “ACTUAL” or “INFORMATION” and for ASPLs that are associated to the track. The distribution of these correlation messages for “ACTUAL” SFPLs starts when the flight is eligible for linkage, while they are only distributed for “INFORMATION” SFPLs when they are actually correlated. For SFPLs that are not correlated, but have the “ACTUAL” flight plan state, a CorrelationMessage is distributed from a design parameter before the entry in the AoR until the exit AoR.
Input	CorrelationMessage
Output	None

Service dynamic behaviour

Figure 7 shows a sequence diagram to reflect the dynamic behaviour of the Correlation Distribution service to support a CWP to:

- Subscribe to the service,
- Unsubscribe from the service,
- Receive up-to-date extended correlation information.

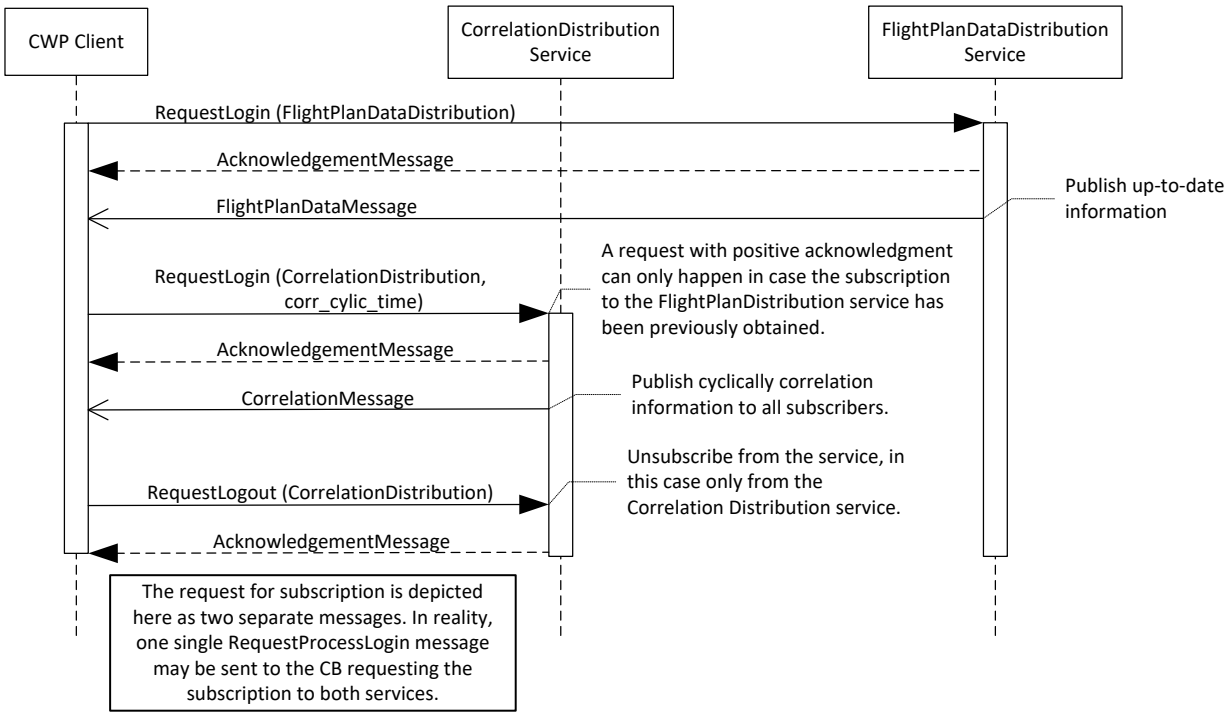


Figure 7: Correlation Distribution service dynamics

3.3.6 Operation distributeFlightBright

Description	The flight bright distribution service provides the means to highlight a flight within for the own OPS sector based on SSR Code(s) “SSR Bright” and/or ModeS callsign(s) “ModeS bright”, and within own or to other OPS sectors (internal/external) based on callsign(s) “flight-plan bright”. The service is implemented by means of the FlightBrightMessage. The message is distributed to the own OPS sector (case of bright for SSR code, ModeS code, or flightplan bright for the own OPS sector) or to another OPS sector (case of flightplan bright to another OPS sector); in other words message distribution is “OPS sector oriented”. At initialisation, actual bright information for the OPS sector is distributed. Per OPS sector there can be a maximum of 20 requests for flight bright.
Pre-requisites	The client is subscribed to the Flight Plan Data Distribution service.
Transmission Events	Flight Bright information is distributed towards interested clients in the following cases: • Distribution to OPS sector requesting SSR bright or ModeS bright (see Flight Bright Management Service dynamics, section 3.8)

	• Distribution to OPS sector for which flight plan bright is intended (either own sector or other sector) • Distribution to pointing OPS Sector upon receipt of external PNT message at the server • Distribution to OPS sector requesting an external flight plan bright (i.e. PNT message) in order to monitor the protocol state of the external point (i.e. ensure LAM message is received).
Input	FlightBrightMessage
Output	None

Note that bright information is removed upon manual request (see Flight Bright Management service dynamics, section 3.8) or when the related flight is deleted.

Service dynamic behaviour

Figure xx shows a sequence diagram to reflect the dynamic behaviour of the Flight Bright Distribution service to support a CWP to:

- Subscribe to the service,
- Unsubscribe from the service,
- Receive up-to-date flight bright information upon initialisation.

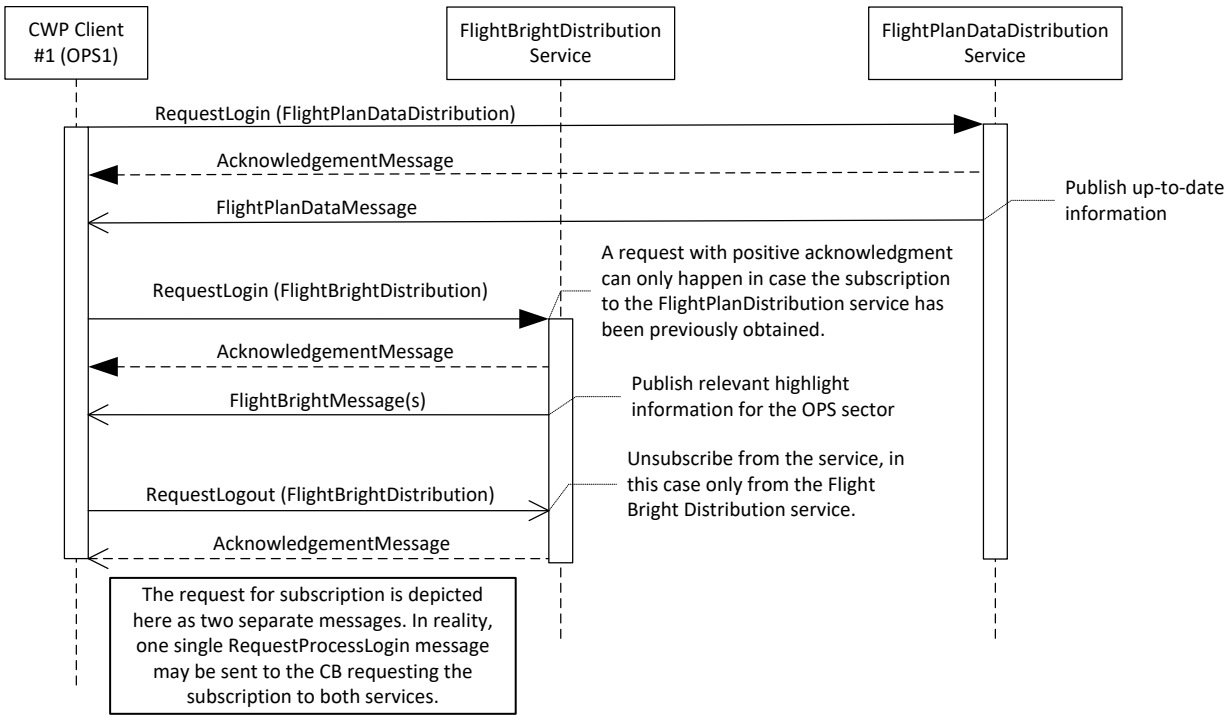


Figure 8: Flight Bright Distribution service dynamics

For a more complex dynamic overview in conjunction with a request to bright a flight, the reader is directed to section 3.8 of this document (Flight Bright Management Service dynamics)

3.3.7 Operation distributeSectorisation

Description	The sectorisation distribution service ensures that each client is provided with the latest up-to-date sectorisation information residing at the server. Basically, the following information is distributed: • Sectorisation pattern: contains the sectorisation pattern (i.e. unique identifier for each sectorisation) selected for each sector group. • Sector composition: composition contains the list of airspace volumes and basic sectors composing each flight sector (see section 5.1.4 for more information on the concept). • Sector consolidation: contains the list of flight sectors consolidated into each OPS sector (see section 5.1.4 for more information on the concept). • Sector allocation: contains the list of flight sectors allocated to each internal/external centre.
Pre-requisites	None.
Transmission Events	The sectorisation message is sent upon any sectorisation change triggered at the server side, either by manual action from a dedicated FDO position, or triggered by a request from the Sectorisation Management Service.
Input	SectorisationMessage
Output	None

Service dynamic behaviour

Figure 9 shows a sequence diagram to reflect the dynamic behaviour of the Sectorisation Distribution service to support a CWP to:

- Subscribe to the service,
- Unsubscribe from the service,
- Receive up-to-date sectorisation information.

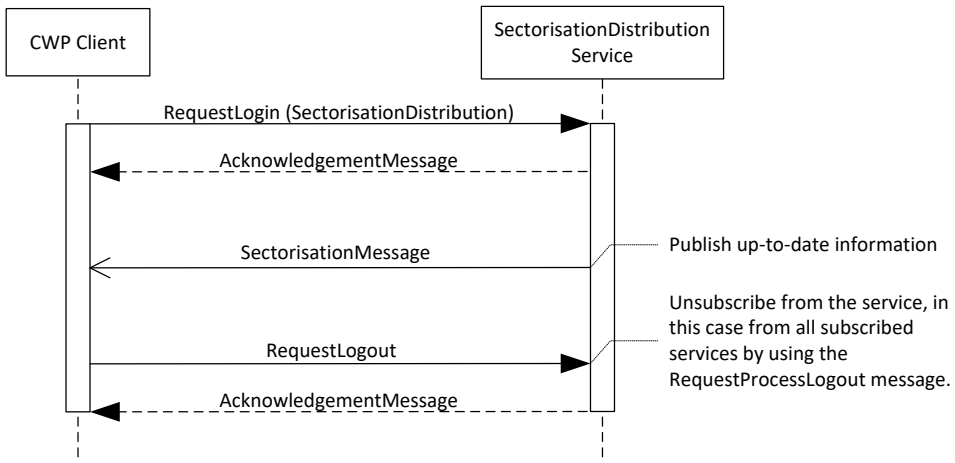


Figure 9: Sectorisation Distribution service dynamics

3.3.8 Operation distributeSsrCode

Description	The service provides the OPS sector with the information related to reservation of a free manual-assignable SSR code. The service is implemented by means of the SsrCodeMessage. The message is distributed to the own OPS sector only.
Pre-requisites	None.
Input	SsrCodeMessage
Output	None

Service dynamic behaviour

Figure 10 shows a sequence diagram to reflect the dynamic behaviour of the SsrCode Distribution service to support a CWP to:

- Subscribe to the service,
- Unsubscribe from the service,
- Receive an up-to-date SSR code (or reset of previous requested code) upon initialisation.

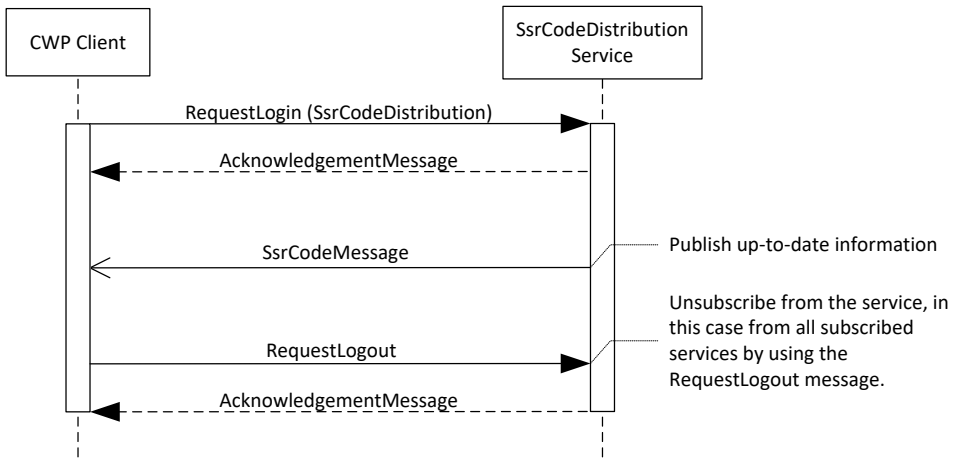


Figure 10: SsrCode Distribution service dynamics

3.3.9 Operation distributeWindForecast

Description	The meteorological information distribution service aims to provide all operational sectors with the latest: • Wind & temperature forecast information extracted from the external world for a range of different levels, • Airport related meteorological information received from the external world from the following sources: o METAR and METAR COR o SPECI and SPECI COR • QNH, transition level and transition altitude information for airports This service is implemented by means of two messages: • WindForecastMessage, containing the wind & temperature forecast information. • AirportMeteoMessage, containing the airport related information.
Pre-requisites	None.
Transmission Events	Meteorological information data is distributed upon any change of such data stored at the server side, either automatically by reception of new external data, or upon manual action by the supervisor.
Input	WindForecastMessage
Output	None

Service dynamic behaviour

Figure 11 shows a sequence diagram to reflect the dynamic behaviour of the Meteorological Information Distribution service to support a CWP to:

- Subscribe to the service,

- Unsubscribe from the service,
- Receive up-to-date meteorological forecast & nowcast information.

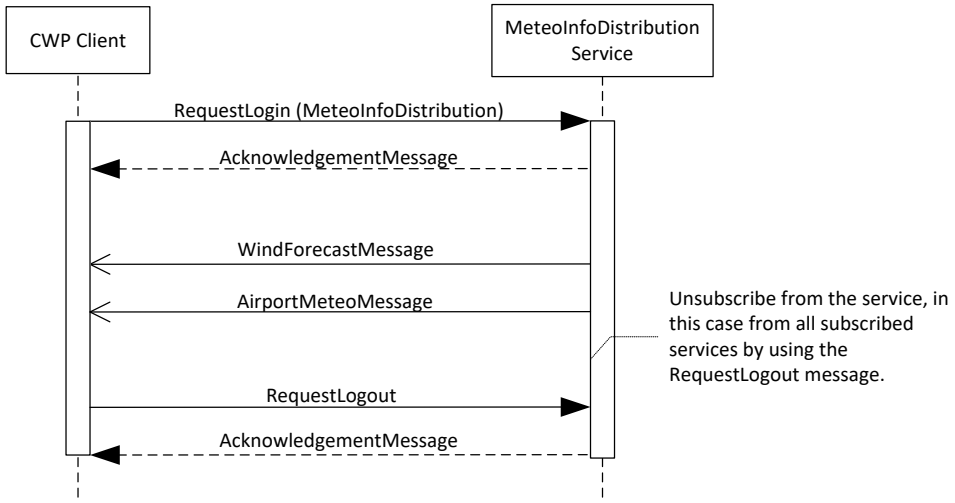


Figure 11: Meteorological Information Distribution service dynamics

3.3.10 Operation distributeMap

Description	The map distribution service aims to provide requesting clients with the latest information of map data. A distinction is made between static, semi-dynamic and dynamic maps: • Static: It is defined offline, and their contours cannot be modified on-line. Textual elements and/or other display elements may be modified. • Semi-dynamic: It is defined offline, and their contours can be modified on-line. Textual elements and/or other display elements may be modified. • Dynamic: They are created on-line. All of their contents may be modified.
Pre-requisites	None.
Transmission Events	Map data is distributed towards interested clients upon the following changes: • Activation, de-activation, creation or deletion of the related map • Modification of the map contents (e.g. addition of textual elements, change of contour, etc.) • Inclusion or exclusion of sectors or sector groups to which the map is required for display
Input	MapMessage
Output	None

Every map is distributed in a single MapMessage. For a static map, there is currently no

information on its graphical content sent (only textual elements are provided) and it's expected that the client has the graphical definition stored locally. For semi-dynamic and dynamic maps, both the graphical and textual elements are provided. At initialisation, all static, semi-dynamic and dynamic maps are provided. A maximum of 100 MapMessages can be transmitted.

Service dynamic behaviour

Figure 12 shows a sequence diagram to reflect the dynamic behaviour of the Map Distribution service to support a CWP to:

- Subscribe to the service,
- Unsubscribe from the service,
- Receive up-to-date map data.

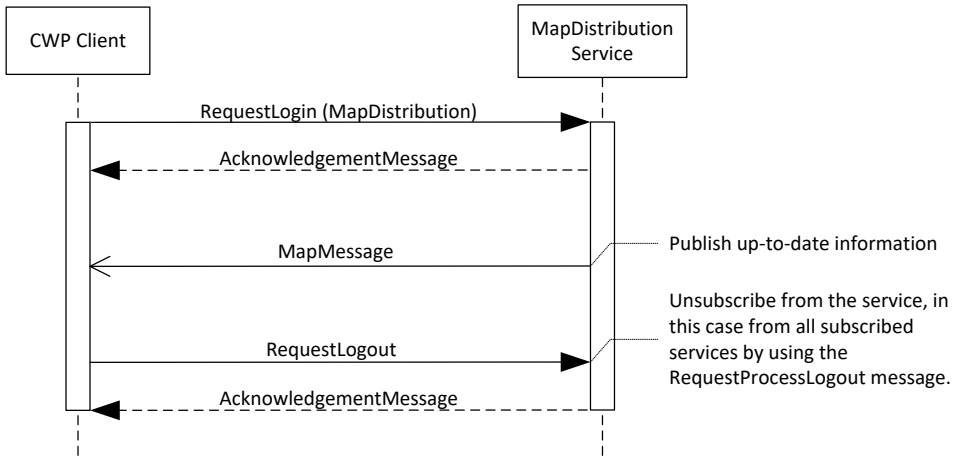


Figure 12: Map Distribution service dynamics

3.3.11 Operation distributeSystemMode

Description	The system mode distribution service provides to all connected & subscribed clients the following FDPS system information: • System: Primary or Fallback • System mode: Operational or Test mode • System sub-mode: Authorised or unauthorised • Link status: ON or OFF (link between Primary and Fallback) • MTCD Status: ON or OFF • MTCD Time Horizon • FDPS degradation level • FDPS SW and adaptation data version FDPS coordinate projection parameters
Pre-requisites	None.
Transmission Events	FDPS system mode information is addressed to all connected & subscribed clients when at least of the aforementioned items change.
Input	SystemModeMessage
Output	None

The service is implemented by means of the SystemMode message.

Service dynamic behaviour

Figure 13 shows a sequence diagram to reflect the dynamic behaviour of the System Mode Distribution service to support a CWP to:

- Subscribe to the service,
- Unsubscribe from the service,
- Receive up-to-date map data.

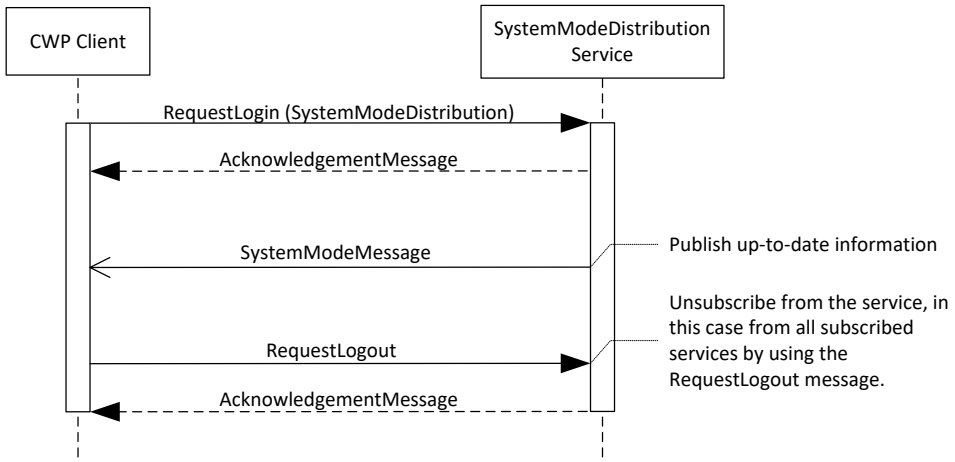


Figure 13: System Mode Distribution service dynamics

3.3.12 Operation distributeSpecialArea

Description	The special area distribution service provides all subscribed clients with the latest status of all special areas (e.g. TSA, TRA, etc.) residing at the server side. For each area the following information is provided: • Airspace status: o pre-warning (pending to be active within 15 minutes) o active o inactive • Operation Mode: o Manual (i.e. the special area activation/de-activation is triggered by manual supervisor action) o Scheduled (i.e. the special area activation/de-activation is triggered automatically by following an activation/de-activation schedule) • The applicable lower and upper level related to the special area activation • The start and end time related to the special area activation.

	• In case of booking via LARA, additional LARA booking information, being: o Unique LARA reservation identifier o LARA activation status o List of callsigns involved in the mission o Mission type o Permeable or non-permeable indicator
Pre-requisites	None.
Transmission Events	Special area status information data is distributed towards interested clients in the following cases: • A pre-defined time before (15 minutes), and just at, status change (i.e. active or inactive) of the special area. • Modification of any of the elements (e.g. levels, activation time, etc.) applying to the special area booking provided the status information is eligible for distribution (see previous bullet).
Input	SpecialAreaMessage
Output	None

The service is implemented by means of the SpecialAreaStatus message. The SpecialAreaStatus message contains the list of special areas next-to-change or currently changing their activity (active/inactive) or covered flight levels.
For each special are, there is currently no information on its graphical content (i.e. contour definition) provided and it's expected that the client has the graphical definition stored locally.
At initialisation, actual information on all special areas residing at the server side are distributed.
Information for maximum 350 special areas can be contained per SpecialAreaStatus message.

Service dynamic behaviour

Figure 14 shows a sequence diagram to reflect the dynamic behaviour of the Special Area Distribution service to support a CWP to:

- Subscribe to the service,
- Unsubscribe from the service,
- Receive up-to-date special area information.

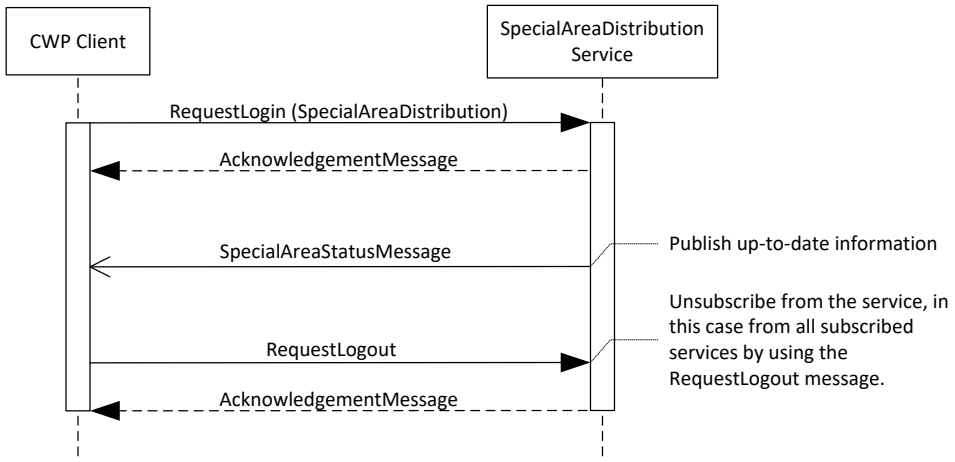
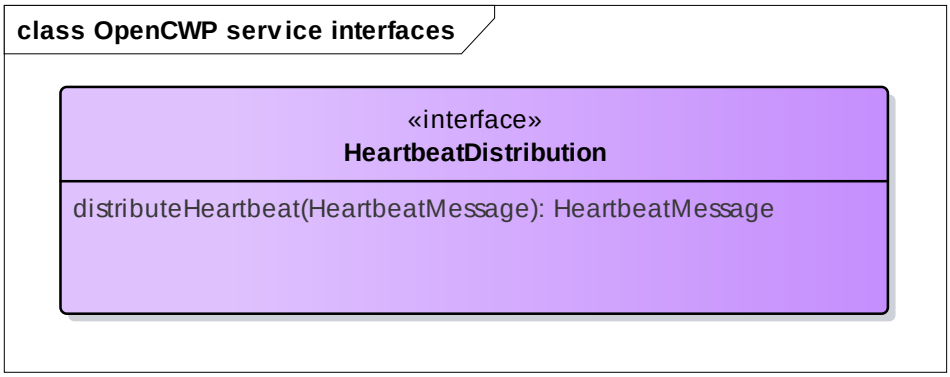


Figure 14: Special Area Distribution service dynamics

3.4 HeartbeatDistribution interface



3.4.1 Operation distributeHeartbeat

Description	The Hearbeat distribution supports the service provider and the service consumer to monitor each other's status.
Pre-requisites	Client has to perform an initialisation/subscription request for any service to enable the Heartbeat Distribution. In return the client is expected to always send a HeartbeatMessage back in reply to each heartbeat from the server side. Heartbeat message monitoring is no longer applied at provider side from the moment a RequestLogout message is sent or time-out is achieved (i.e. no heartbeat messages received anymore).
Transmission Events	Server application heartbeat information (i.e. from service towards clients) is provided cyclically. The initial cycle is set to 5 seconds (governed by a design parameter).
Input	HeartbeatMessage (full message)
Output	HeartbeatMessage (header only)

The Heartbeat distribution is indicating cyclically the status of the connection between the FDPS and the CB, and the status of the connector box connection with the client. In the first case, it considers the link with FDPS established whenever heartbeats are received from the FDPS application. In the latter case, the connector box does implement a heartbeat service (indicating the status of its processes) to its clients.

Involved subsystems and link status								Heartbeat message content		Client behaviour	
FDPS	EN	FDPS Client (CB)	EN	CWP Server (CB)	EN	CWP	==>	upstream-server-alive T server-alive T	==>	1	No action (normal reception)
	DI		EN		EN			upstream-server-alive F server-alive T		2	Wait for connection with upstream-server or server
	EN		DI		EN		==>	upstream-server-alive F server-alive F	==>		
	DI		DI		EN						
	-		-		DI		==>	No heartbeat	==>	3	Periodic retry to login

Table 3: Description of CB heartbeat contents related to connection status with involved subsystems

Table 3 above gives a schematic explanation of the different statuses that are emitted in the heartbeat message by the Connector box, depending on the link status between the involved subsystems. In the first table the different subsystems and their respective links are depicted. In

the second table, the heartbeat contents as emitted by the connector box are explained, while the third table describes the expected client (CWP) behaviour.
The Heartbeat Distribution service is enabled by default[6]. As soon as a client subscribes the first[7] time to a service, and the request is accepted, the connector box will provide cyclically a heartbeat. The message flow is considered to be bi-directional; once the server sends the 1st heartbeat message, the client is expected to send a HeartbeatMessage in reply, but only the header file (protocol and application level) should be populated. This allows monitoring the client at the same time. For the time being, the heartbeat message from the client to the connector box is expected via the InputQueue of the CWP server process, however no AcknowledgementMessage will be sent in response.
Note that although the aim of the Heartbeat Distribution service is to indicate server or client presence information; it does also support more detailed monitoring information, like type of assigned role on the client for example (this is because the header definition supports such information via the source-function-id field).
When more than a design parameter of subsequent heartbeat messages is missed, the connector box will remove the client from its configuration and no more message will be sent (the TCP/IP connection remains established). There is no pre-notification from the server-side to the customer-side for this event: the client is expected to perform heartbeat monitoring as well and a new RequestLogin is required in order to trigger a new initialisation (i.e. client behaviour #3 as indicated in Table 3 above).
On the other hand, when there are heartbeats indicating that the connection with the upstream-server is lost (i.e. when the connection with FDPS is considered disabled or the connection between the FDPS Client and CWP Server process is considered disabled at the connector box), the client is expected to wait for the connection with the FDPS to be re-established (i.e. client behaviour #2 as indicated in Table 3 above).

3.4.1.1 Service dynamic behaviour

Figure 15 shows a sequence diagram to reflect the dynamic behaviour of the Heartbeat Distribution to support the service provider and the service consumer (CWP client) to monitor each other's status.

[6] The RequestLogin message does allocate for a specific request for the Heartbeat Distribution service from the client towards the server. However, this is not implemented in this phase of the project.

[7] When a request is made for an additional service, no additional heartbeat cycle will be started.

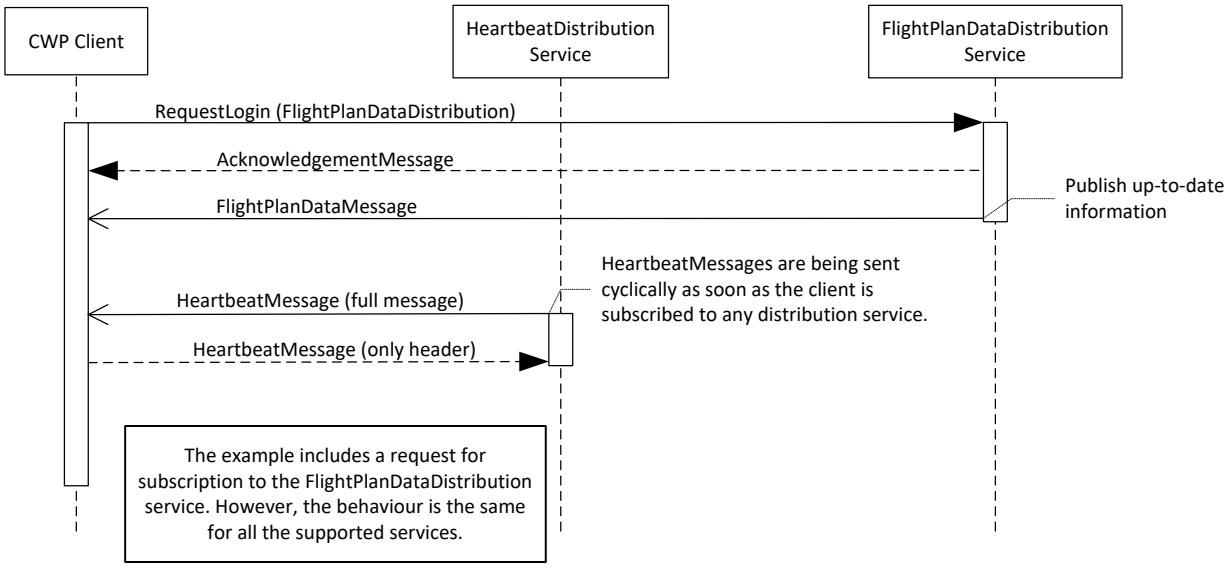


Figure 15: Heartbeat Distribution service dynamics

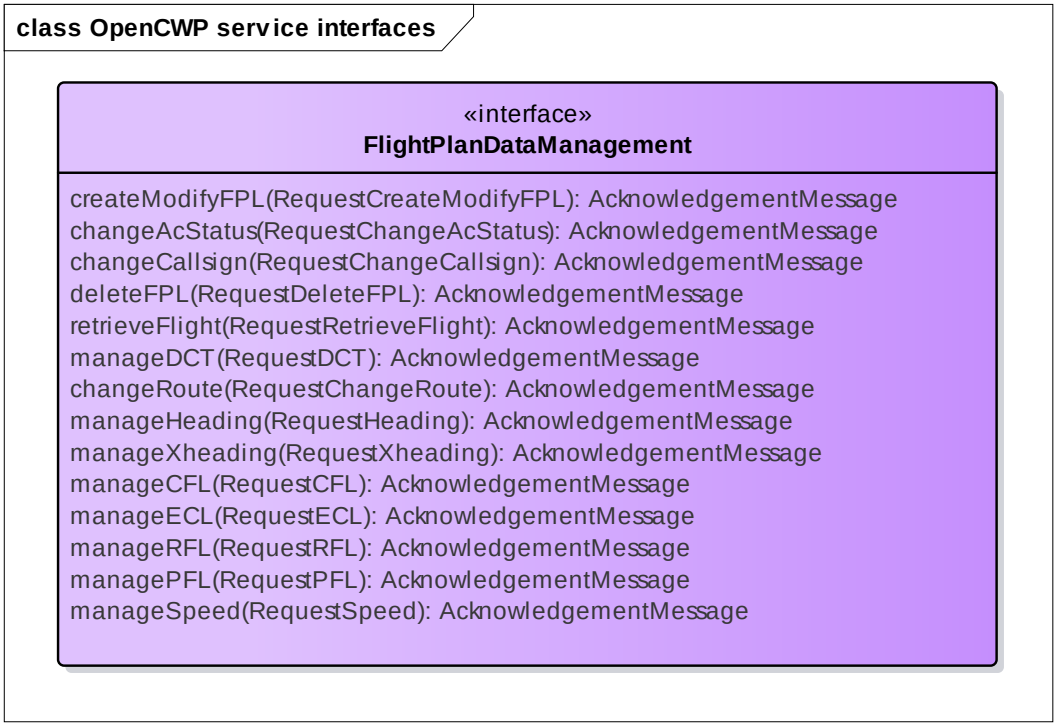
3.5 FlightPlanDataManagement interface

The Flight Plan Data Management service supports any connecting CWP client to send certain inputs in order to trigger the correct & latest up-to-date controller information, more specifically:

- Create ASPL or SFPL,
- Modify an ASPL/SFPL or upgrade an SFPL,
- Downgrade an SFPL into an ASPL,
- Delete an existing ASPL/SFPL,
- Submit requests about instructions/clearances given to the flight crew (e.g. DCT, CFL, NFL, speed, heading, …)
- Change status information regarding the flight's airframe (e.g. no FSSA, RVSM status, …)
- Etc.

When an input is made and successfully processed the response to the request is delivered in two parts:

- Each input is first replied with the AcknowledgementMessage to indicate the acceptance or rejection of the request. The client is expected to start an internal timer in order to capture those cases where there would be no reply. In case of the latter, the client is expected to trigger a new input.
- Secondly, provided the input was accepted, the updated information (as delivered by the Flight Plan Distribution service) on the flight is sent. As such, subscription to the Flight Plan Distribution service is mandatory prior the user requesting flight data modifications.



3.5.1 Pre-requisites

The client is subscribed to the Flight Plan Data Distribution service.

3.5.2 Transmission Events

Not applicable.

3.5.3 Operation createModifyFPL

Description	The operation allows to: • Create ASPL or SFPL, • Modify an ASPL/SFPL, • Upgrade an ASPL into SFPL, • Downgrade an SFPL into an ASPL.
Input	RequestCreateModifyFPL message
Output	AcknowledgementMessage

3.5.4 Operation changeAcStatus

Description	The operation allows modifying: • number of aircraft related to a flight plan • aircraft type • flight's wake-turbulence category • flight's deviation status

	• RVSM capability • 8.33 kHz capability • UHF equipment • BRNAV/PRNAV equipment • ModeS capability • Number of people on board • FSSA capability • Avoiding weather indicator • Fuel dumping indicator
Input	RequestChangeAcStatus message
Output	AcknowledgementMessage

3.5.5 Operation changeCallsign

Description	The operation allows to: • Modify the callsign of an existing ASPL or SFPL.
Input	RequestChangeCallsign message
Output	AcknowledgementMessage

3.5.6 Operation deleteFPL

Description	The operation allows to manually deleting an existing ASPL or SFPL from the system.
Input	RequestDeleteFPL message
Output	AcknowledgementMessage

3.5.7 Operation retrieveFlight

Description	The operation allows retrieving manually the flight plan data (Flight Plan Data Message) for an existing ASPL or SFPL from the system.
Input	RequestRetrieveFlight message
Output	AcknowledgementMessage

3.5.8 Operation manageDCT

Description	From an operational perspective, a DCT action can represent multiple cases. The interface definition has been considered to fulfil the operational scenarios listed below. For completeness reasons the

required options to be passed in the RequestDCT input are given below:

- DCT to a (intermediate) Point
  - to

    Allows the sending of a single route point name (or coordinate) with optional intermediate point. In this case, the CB will automatically determine the most likely type of DCT instruction given (i.e. make the distinction between “to-original-route” and “to-trajectory” as given below). From an interface and client implementation perspective this option allows to perform DCT actions without the requirement for the client to reference to index in the trajectory. Also, this option allows also performing a DCT instruction for an ASPL[8] for which no trajectory is available.

- DCT to Point on the Trajectory (Uplink = UM74)
  - to-trajectory
    - intermediate-point not sent
    - end-point represents the index point and position in the trajectory
- DCT to Point on the Original Route (Uplink = UM74 + UM72)
  - to-original-route
    - intermediate-point not sent
    - end-point represents the point name (located on the original route)
- DCT to (Intermediate) Point not located on the trajectory; in this case the end-point is on the trajectory (Uplink = UM79: "CLEARED TO [end-point] VIA [intermediate-point])
  - to-trajectory
    - intermediate-point represents the route point name or position of the intermediate point
    - end-point represents the index point and position in the trajectory
- DCT to (Intermediate) Point not located on the trajectory; in this case the end-point is on the original route (Uplink = UM79: "CLEARED TO [end-point] VIA [intermediate-point] + UM72)
  - to-original-route
    - intermediate-point represents the route point name or position of the intermediate point

[8] DCT instruction for ASPL is currently not yet supported at the server side, except for the case of cancelling a previous heading instruction (i.e. DCT action with XHDG as point). In all other cases, the DCT instruction will be returned hence if the instruction is made for an ASPL, it will be returned with an error.

	▪ end-point represents the point name (located on the original route) • DCT to (Intermediate) Point not located on the trajectory, and no end-point specified. In this case, the DCT is considered to be a point completely off-route, which is not to be re-joined. A typical example is a DCT to a point outside the AoI, which is not located on the trajectory (Uplink = UM74). o to-trajectory ▪ intermediate-point represents the route point name or position of the intermediate point ▪ end-point represents the null index point Note 1: A current heading restriction will be removed upon receiving a DCT input. Note 2: For ASPLs, a DCT to an Intermediate OR End Point is allowed (with whatever DctData option) provided any reference to a trajectory point is populated with the null-TrajectoryPointNumber. Inputs combining both Intermediate AND End Point will not be processed.
Input	RequestDCT message
Output	AcknowledgementMessage

3.5.9 Operation changeRoute

Description	From an operational perspective, a change route action represents a re-routing of the flight across multiple waypoints (unlike a DCT where the flight is instructed to go to only one waypoint). Note 1: A current heading restriction will be removed upon receiving Operation changeRoute. Note 2: Only available for SFPLs.
Input	RequestChangeRoute message
Output	AcknowledgementMessage

3.5.10 Operation manageHeading

Description	The interface definition has been compiled taking into account the operational cases listed below. For completeness reasons the possible parameters, for such situations, to be passed in the RequestHeading input are indicated as well: • Present Heading (uplink possible) • Assigned Heading – relative or absolute true/magnetic heading (uplink possible) • Non-specified Heading; use track direction as heading and update trajectory with this value (no uplink possible)

	• Heading to avoid weather, for example CBs; use track direction as heading an generate an observed tactical trajectory with it (no uplink possible) In the first three cases above, the type of heading closure is mandatory in the request. The following cases have been distinguished in case the application limit and trajectory resuming point are included in the request: • No application limit and re-joining point specified in the input: a default application limit will be applied and the trajectory will be closed based on pre-defined closure rules. • Application limit specified as distance and re-joining point present in the request: the specified distance and re-joining-point will be applied. • Application limit specified as "Indefinite": the system will auto-close the trajectory at AoR exit. In all cases, the input heading is considered as magnetic heading.
Input	RequestHeading message
Output	AcknowledgementMessage

3.5.11 Operation manageXheading

Description	The command allows resetting a previous heading instruction for ASPLs only. In case the command is requested for an SFPL, the result will be returned with an error.
Input	RequestXheading message
Output	AcknowledgementMessage

3.5.12 Operation manageCFL

Description	To process the input of a cleared flight level, with application time/distance and/or ROCD restriction.
Input	RequestCFL message
Output	AcknowledgementMessage

3.5.13 Operation manageECL

Description	To process the input of an en-route cruising level. The ECL is applied until the exit of the AoI (i.e. propagated all the way), unless the application-limit is present in the request.
Input	RequestECL message
Output	AcknowledgementMessage

3.5.14 Operation manageRFL

Description	To process the input of a requested flight-level as requested by the pilot or filed by the airline company.
Input	RequestRFL message
Output	AcknowledgementMessage

3.5.15 Operation managePFL

Description	To process the input of a planned flight level (PFL). The PFL is best described as an ECL, but only applied until the requesting sector's exit.
Input	RequestPFL message
Output	AcknowledgementMessage

3.5.16 Operation manageSpeed

Description	The interface definition has been compiled taking into account the operational cases listed below: • Present speed (uplink possible) • Assigned speed, including route point until where the speed restriction should apply (uplink possible) • Speed range, including route point until where the speed restriction should apply (between a lower and upper speed) • Minimum or Maximum speed instruction, including route point until where the speed restriction should apply (uplink possible) • Keyword, to represent a pure textual speed instruction to the ATCO (e.g. CLEAN speed) without affecting the flight's calculated profile • Resume speed, to cancel any previous speed restriction (uplink possible)
Input	RequestSpeed message
Output	AcknowledgementMessage

3.5.17 Service dynamic behaviour

In Figure 16 a typical input sequence for a CFL input is presented. Any other type of operation supported by the flight plan data management service exploits the same behaviour. As can be seen, there is a dependency between the Flight Plan Distribution and Management service. Upon any input (delivered by the Flight Plan Management service), the operational reply (i.e. update of flight plan information) will be performed by the Flight Plan Distribution service. Prior requesting any input, the requestor need to be subscribed to the Flight Plan Distribution service, otherwise the request will be rejected with a negative acknowledgment.

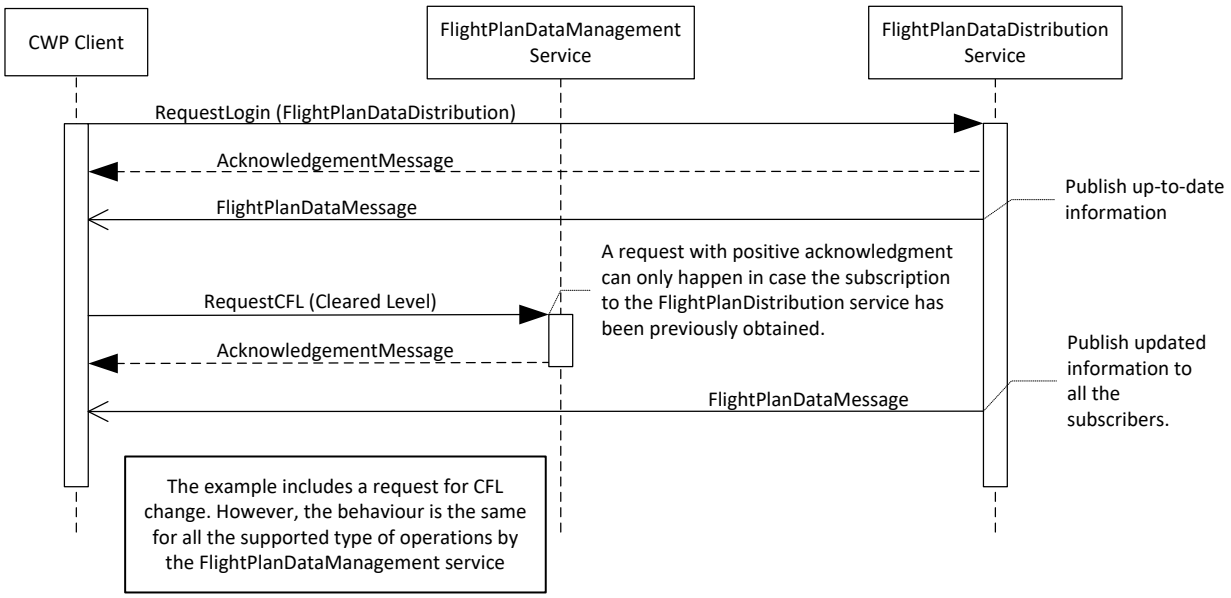


Figure 16: Flight Plan Data Management service dynamics

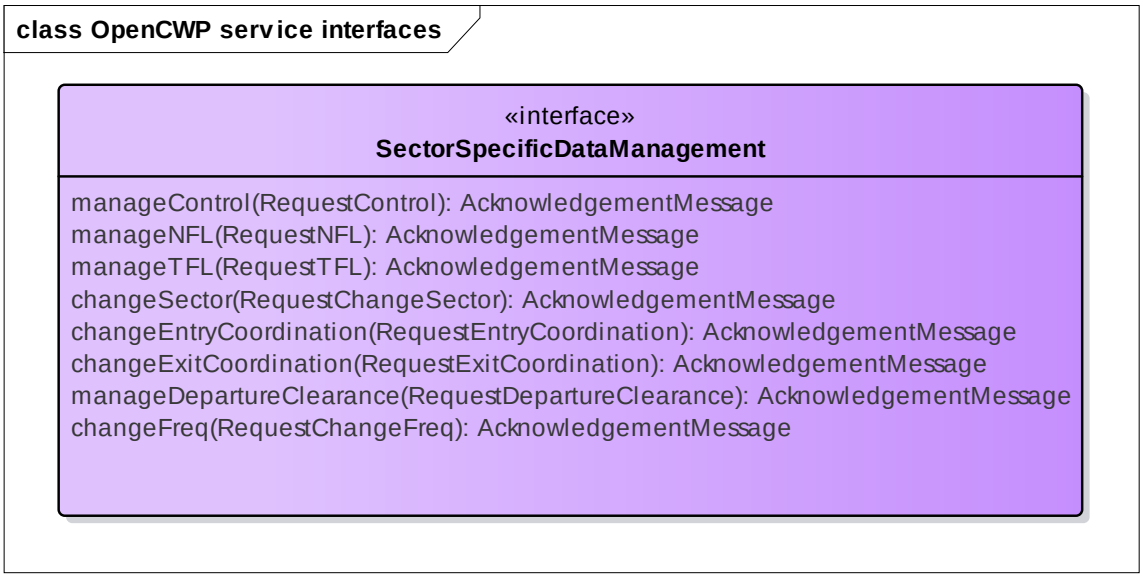
3.6 SectorSpecificDataManagement interface

The Sector Specific Data Management service supports any connecting CWP client to send certain inputs in order to trigger the correct & latest up-to-date controller information regarding sector-specific information and coordination & transfer information, more specifically:

- Allow taking control of a flight (or proposing hand-over, request-on-frequency, etc.),
- Change the coordinated entry and exit levels,
- Deliver departure clearance for an flight departing from an internal aerodrome,
- Skip and cancel-skip of an internal sector,
- Bypass and cancel-bypass of the 1st downstream internal sector,
- Delegate the flight to another internal sector,
- Change the next downstream internal sector into the preferred one,
- Change the entry/exit frequency of sectors,
- Etc.

When an input is made and successfully processed the response to the request is delivered in two parts:

- Each input is first replied with the AcknowledgementMessage to indicate the acceptance or rejection of the request. The client is expected to start an internal timer in order to capture those cases where there would be no reply. In case of the latter, the client is expected to trigger a new input.
- Secondly, provided the input was accepted, the updated information (as delivered by the Flight Plan Data Distribution & Sector Specific Data Distribution service) on the flight is sent. As such, subscription to the both aforementioned services is mandatory prior the user requesting sector specific data modifications.



3.6.1 Pre-requisites

The client is subscribed to the Flight Plan Data Distribution & Sector Specific Data Distribution service.

3.6.2 Transmission Events

Not applicable.

3.6.3 Operation manageControl

Description	To process the input of a control input command, to allow either: • Take control of the flight, • Transfer control of the flight, • Delegate control of the flight to the sector where the flight is geographically located, • Hand-over propose/accept.
Input	RequestControl message
Output	AcknowledgementMessage

3.6.4 Operation manageNFL

Description	To process the input of an entry flight-level for the coordination between internal sectors or with the external upstream partner.
Input	RequestNFL message
Output	AcknowledgementMessage

3.6.5 Operation manageTFL

Description	To process the input of a transfer flight-level for the coordination between internal sectors or with the external downstream partner.
Input	RequestTFL message
Output	AcknowledgementMessage

3.6.6 Operation changeSectors

Description	The operation allows to: • Skip and cancel-skip of an internal sector, • Bypass and cancel-bypass of the 1st downstream internal sector, • Delegate the flight to another internal sector, • Change the next downstream internal sector into the preferred one.
Input	RequestChangeSectors message
Output	AcknowledgementMessage

3.6.7 Operation changeEntryCoordination

Description	The operation allows performing coordination changes related to the entry of the sector performing the request. The operation applies to both internal and external entry coordination. Changes to the following items can be requested, or negotiated: • NFL & NSFL (as also possible with requestNFL – see section 3.6.4) • Departure level (indicates the initial cleared level for departure flights out of an internal aerodrome) • ETO (Estimated Time Over) • COP (Coordination Point) • PSSR (Present SSR Code) • DCT-to point (including intermediate point if required) • Accepting Frequency • Speed • Heading
Input	RequestEntryCoordination message
Output	AcknowledgementMessage

3.6.8 Operation changeExitCoordination

Description	The operation allows performing coordination changes related to the exit of the sector performing the request. The operation applies to both internal and external exit coordination. Changes to the following items can be requested, or negotiated: • TFL & TSFL (as also possible with requestTFL – see section 3.6.5) • ETO (Estimated Time Over) • COP (Coordination Point) • PSSR (Present SSR Code) • DCT-to point (including intermediate point if required) • Transferring Frequency • Speed • Heading
Input	RequestExitCoordination message
Output	AcknowledgementMessage

3.6.9 Operation manageDepartureClearance

Description	The operation allows performing a departure clearance (for a flight departing from an internal aerodrome). For example, a TWR sector can perform this action, or eventually a higher sector that has to deliver the departure clearance.

	Items available in the departure clearance are kept limited for the first phase and include: • Departure level (indicates the initial cleared level for departure flights out of an internal aerodrome) • Take-off time • PSSR (Present SSR Code) • Accepting Frequency • SID • Departure runway
Input	RequestDepartureClearance message
Output	AcknowledgementMessage

3.6.10 Operation changeFreq

Description	The operation allows an OPS sector: • To select the frequency to be sent to the transferring previous adjacent center or internal sector, • To select the default for the frequency to be sent to the transferring previous adjacent center or internal sector, • To change the exit frequency with the next partner or internal sector, • To reset the exit frequency.
Input	RequestChangeFreq message
Output	AcknowledgementMessage

3.6.11 Service dynamic behaviour

In Figure 17 a typical input sequence for a request to change the sector sequence of a flight is presented. Any other type of operation supported by the Sector Specific Data Management service exploits the same behaviour. There is a dependency between the Sector Specific Data Disitribution/Management and Flight Plan Data Distribution service.

Upon any input (delivered by the Sector Specific Data Management service), the operational reply will (i.e. update of sector specific information) will be performed by the Sector Specific Distribution service and also reflected in an update of the Flight Plan Data Message (supported by the Flight Plan Data Distribution service).

Prior requesting any input, the requestor needs to be subscribed to both the Flight Plan Distribution and Sector Specific Data Distribution service; otherwise the request will be rejected with a negative acknowledgment.

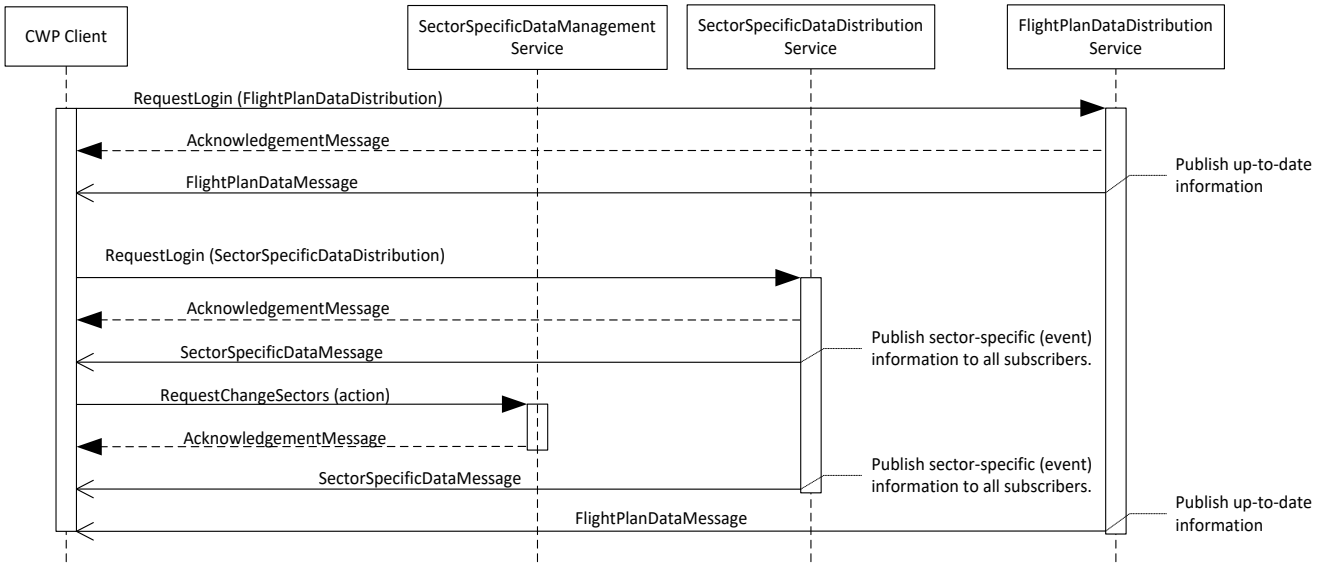


Figure 17: Sector Specific Data Management service dynamics

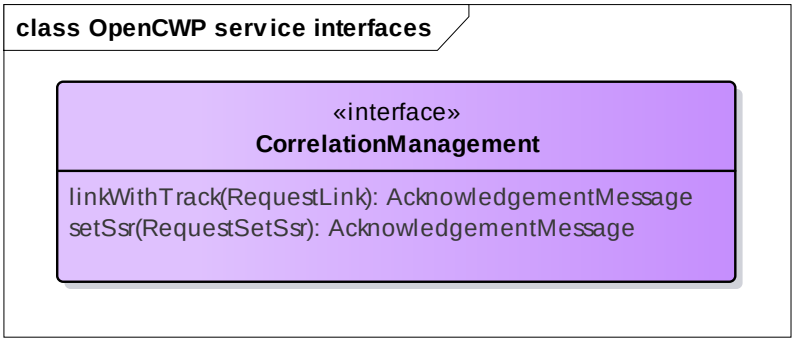
3.7 CorrelationManagement interface

The Correlation Management service allows any connecting CWP client to perform inputs related to the linkage or unlinkage of flight plans with tracks, more specifically:

- Link a flight plan with a specific track,
- Unlink a flight plan from a specific track,
- Set the present, next or downstream SSR code a flight.

When an input is made and successfully processed the response to the request is delivered in two parts:

- Each input is first replied with the AcknowledgementMessage to indicate the acceptance or rejection of the request. The client is expected to start an internal timer in order to capture those cases where there would be no reply. In case of the latter, the client is expected to trigger a new input.
- Secondly, provided the input was accepted, the updated information (as delivered by the Flight Plan Distribution service) on the flight is sent. As such, subscription to the flight plan distribution service is mandatory prior the user requesting flight data modifications. Additionally, if the user is subscribed to the Correlation Distribution service, extended correlation information will be sent.



3.7.1 Pre-requisites

The client is subscribed to the Flight Plan Data Distribution service and optionally the Correlation Distribution service.

3.7.2 Transmission Events

Not applicable.

3.7.3 Operation linkWithTrack

Description	To process the input of a manual linkage/un-linkage request of a flight plan with a track.
Input	RequestLink message
Output	AcknowledgementMessage

3.7.4 Operation setSsr

Description	To process the input of setting the present, next or downstream SSR code of a flight plan.

Input	RequestSetSsr message
Output	AcknowledgementMessage

3.7.5 Service dynamic behaviour

In Figure 18 a typical input sequence for a un-linkage request is presented. Any other type of operation supported by the Correlation Management service exploits the same behaviour. As can be seen, there is a dependency between the Correlation Management and Flight Plan Data Distribution service (if subscribed to the both). Upon any input (delivered by the Correlation Management service), the updated linkage information will be performed by the Correlation Distribution and Flight Plan Data Distribution service (if subscribed to both). Prior requesting any input, the requestor needs to be subscribed to Flight Plan Data Distribution service (and optionally the Correlation Management service); otherwise the request will be rejected with a negative acknowledgment.

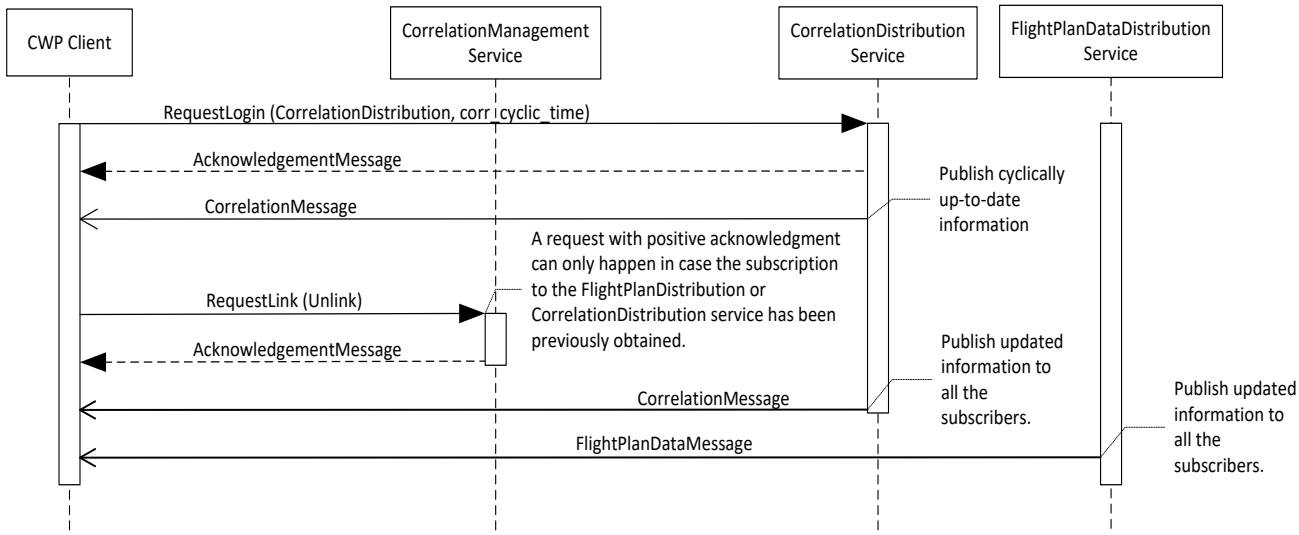


Figure 18: Correlation Management service dynamics

3.8 FlightBrightManagement interface

The Flight Bright Management service allows any connecting CWP client to perform inputs related to highlight of a track or flight plan, more specifically:

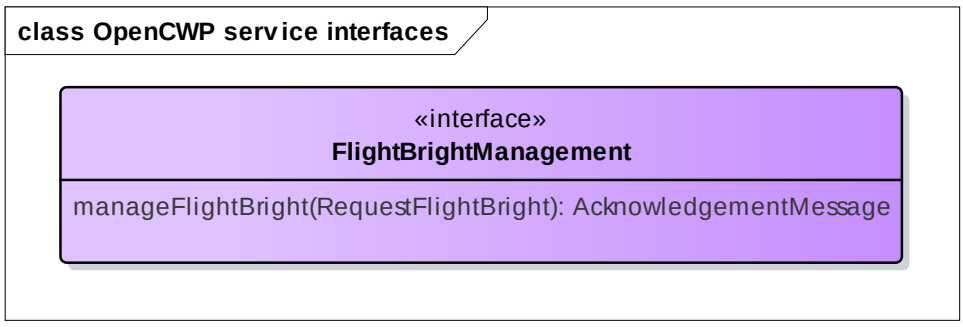
- SSR Bright:
  - Add an SSR Code to the SSR Bright function for his OPS sector
  - Cancel all SSR Codes from the SSR Bright function for his OPS sector
  - Delete one SSR Code from the SSR Bright function for his OPS sector

- ModeS Bright:
  - Add an ModeS callsign to the ModeS Bright function for his OPS sector
  - Cancel all ModeS callsign from the ModeS Bright function for his OPS sector
  - Delete one ModeS callsign from the ModeS Bright function for his OPS sector

- FPL Bright:
  - Add a flight to the FPL Bright function for his OPS sector
  - Delete a flight from the FPL Bright function for his OPS sector
  - Add a flight to the FPL Bright function of another internal OPS Sector (by specifying an internal flight sector)
  - Point a flight to an external flight sector / centre

When an input is made and successfully processed the response to the request is delivered in two parts:

- Each input is first replied with the AcknowledgementMessage to indicate the acceptance or rejection of the request. The client is expected to start an internal timer in order to capture those cases where there would be no reply. In case of the latter, the client is expected to trigger a new input.
- Secondly, provided the input was accepted, the updated information (as delivered by the Flight Bright Distribution service) on the flight is sent. As such, subscription to the Flight Bright distribution service is mandatory prior the user requesting modifications. Please do note that the distribution of the FlightBright message is OPS sector oriented.



3.8.1 Pre-requisites

The client is subscribed to the Flight Bright Distribution service.

3.8.2 Transmission Events

Not applicable.

3.8.3 Operation manageFlightBright

Description	To process the input of a manual SSR Code track bright, ModeS track bright or flight plan bright.

Input	RequestFlightBright message
Output	AcknowledgementMessage

3.8.4 Service dynamic behaviour

In Figure 19 a typical input sequence for an SSR bright and flight plan request is presented. As can be seen, there is a dependency between the Flight Bright Management and Flight Plan Data Distribution service; the sole reason for this is in case of a flight plan bright. Upon any input (delivered by the Flight Bright Management service), the highlight information will be performed by the Flight Bright Distribution.

It is important to note that in this example both connecting clients are having a different OPS sector allocated. In the first request, client #1 which has OPS1 sector, requests the highlight of a track with a certain SSR1 code. Since the distribution of the FlightBrightMessage is OPS sector oriented, the message will only be distributed to the connected clients sharing the same OPS sector (i.e. client #2 which has OPS2 sector will not receive this information).

After this, client #1 now performs a flight plan bright action, highlighting the FP1 to OPS2 sector (client #2). Again, since the distribution is OPS sector oriented, only OPS2 (client #2) will receive a FlightBrightMessage. When now client #2 request an SSR bright for the same track (i.e. SSR1) as client #1 previously requested, the FlightBrightMessage will only be updated towards the OPS sector requesting it (OPS2, client #2 in this case); the FlightBrightMessage for client #2 now contains both the SSR1 code highlight, and FP1 highlight.

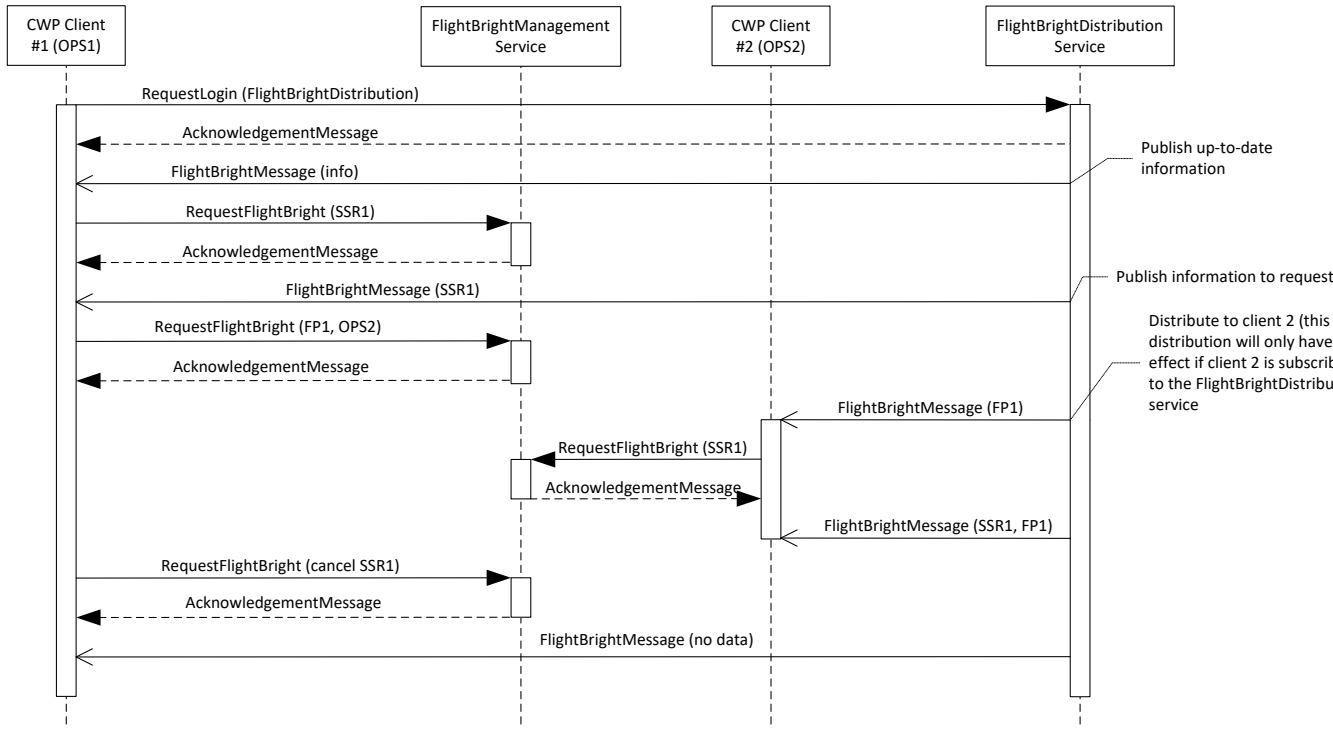


Figure 19: Flight Bright Management service dynamics #1

To further explain the OPS oriented distribution of the FlightBright message, another example is presented in Figure 20, where again two clients are connected, however this time they share both the same OPS sector. Similar to the previous example client #1 (OPS1 sector) requests an SSR bright. This time, the FlightBrightMessage is distributed to both client #1 and client #2 since they both have the same OPS sector. Similarly, when a flight plan bright is requested by client #1 to distribute to its own OPS sector, both client #1 and #2 will receive it because they both have the

same OPS sector allocated.

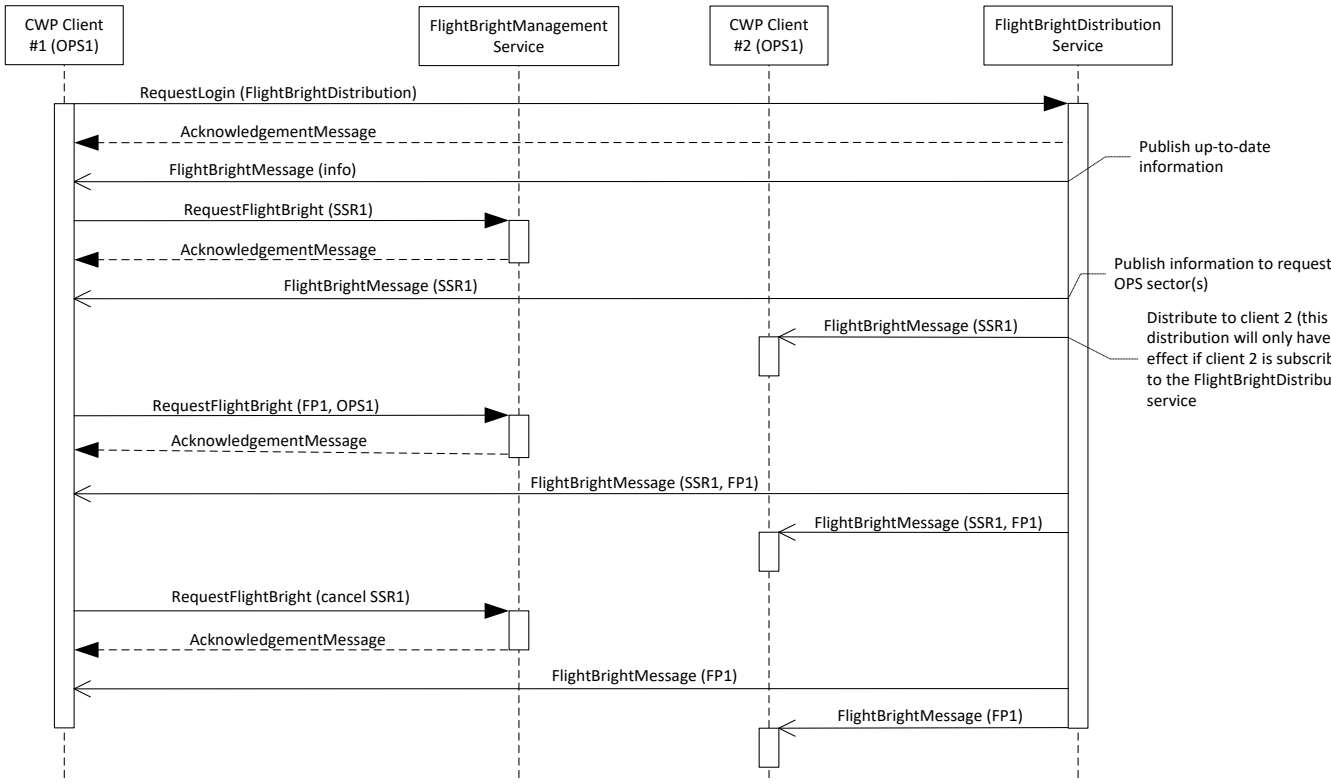


Figure 20: Flight Bright Management service dynamics #2

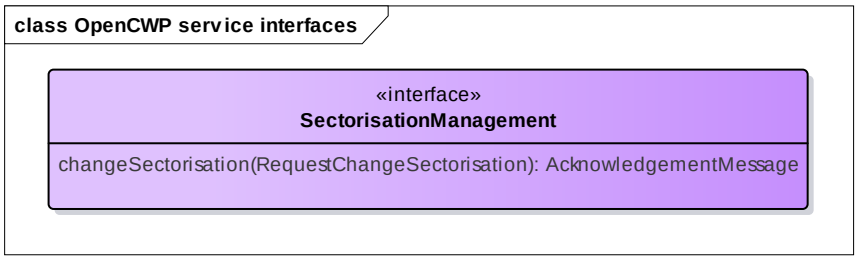
3.9 SectorisationManagement interface

The sectorisation management service provides any connecting client with the means to perform a re-sectorisation change. Two kind of requests can be made, being:

- A request to verify a new sectorisation change (i.e. would the request once executed be valid for the server?),
- A request to perform a sectorisation change.

When an input is made and successfully processed the response to the request is delivered in two parts:

- Each input is first replied with the AcknowledgementMessage to indicate the acceptance or rejection of the request. The client is expected to start an internal timer in order to capture those cases where there would be no reply. In case of the latter, the client is expected to trigger a new input.
- Secondly, provided the input was accepted, the updated information (as delivered by the Sectorisation Distribution service) is sent. As such, subscription to the Sectorisation distribution service is mandatory prior the user requesting sectorisation modifications.



3.9.1 Pre-requisites

The client is subscribed to the Sectorisation Distribution service.

3.9.2 Transmission Events

Not applicable.

3.9.3 Operation changeSectorisation

Description	To process the input of a manual sectorisation request.
Input	RequestChangeSectorisation message
Output	AcknowledgementMessage

3.9.4 Service dynamic behaviour

In Figure 21 a typical input sequence for a request to change the sectorisation is presented. There is a dependency between the Sectorisation Disitribution and Management service.
Upon any input (delivered by the Sectorisation Management service), the operational reply will (i.e. update of the applied sectorisation) will be performed by the Sectorisation Distribution service and also reflected in an update of the Sectorisation Message.
In case of a request to “verify” a sectorisation change, an acknowledgement message is returned indicating if the verification is successful or not. To actually implement the sectorisation change, the client is expected to send the RequestChangeSectorisation message with option “execute” (when he previous verification was successful) or “force” (case where the previous verification

failed). The main reason of returning a failed verification is that another connected client is in the course of requesting a sectorisation change (i.e. it has the “token” for sectorisation at the server). Prior requesting any input, the requestor needs to be subscribed to the Sectorisation Distribution service; otherwise the request will be rejected with a negative acknowledgment.

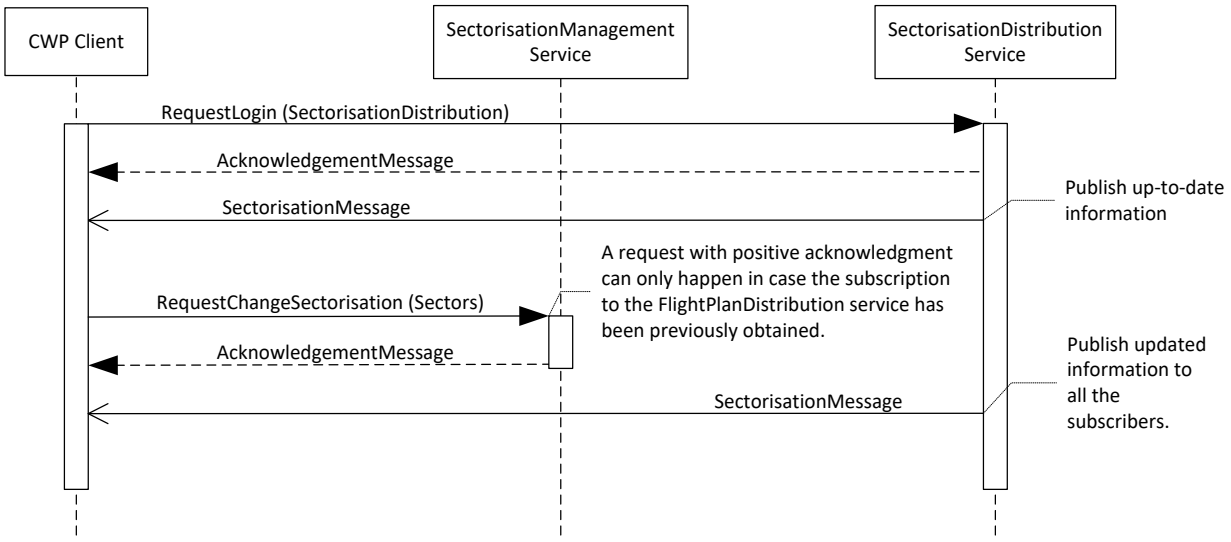
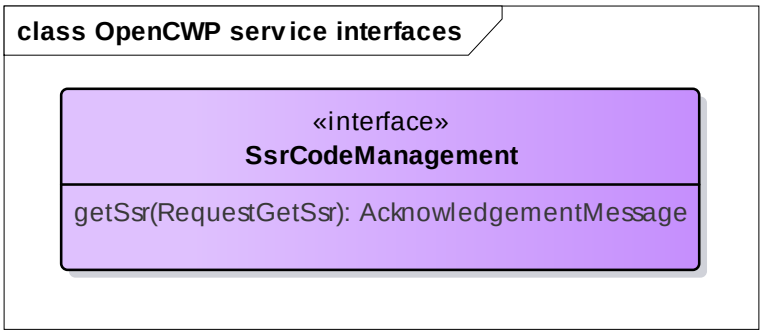


Figure 21: Sectorisation Management service dynamics

3.10 SsrCodeManagement interface

The SSR code management service allows a CWP client to reserve an SSR code for manual assignment later on (i.e. manual assignment by using the Operation setSsr service call, see section 3.7.4), and to clear such code from display in the whole OPS sector.
An SSR code reserved for manual assignment is not available for automatic assignment.
The SSR code will be reserved during a design parameter time and then released if not manually assigned to any flight plan or released according to the standard release rules if manually assigned to a flight plan during this design parameter time.
When an input is made and successfully processed the response to the request is delivered in two parts:

- Each input is first replied with the AcknowledgementMessage to indicate the acceptance or rejection of the request. The client is expected to start an internal timer in order to capture those cases where there would be no reply. In case of the latter, the client is expected to trigger a new input.
- Secondly, provided the input was accepted, the updated information (as delivered by the SsrCode Distribution service) is sent. As such, subscription to the SsrCode distribution service is mandatory prior the user requesting modifications. Please do note that the distribution of the FlightBright message is OPS sector oriented.



3.10.1 Pre-requisites

The client is subscribed to the SsrCode Distribution service.

3.10.2 Transmission Events

Not applicable.
OperationsThe SsrCodeManagement interface consists of one operation. This section describes the possible request and its data content.

3.10.3 Operation getSsr

Description	The manual input allows: • to reserve a SSR code • to clear the SSR code on all CWP units within the OPS sector
Input	RequestGetSsr message
Output	AcknowledgementMessage

3.10.4 Service dynamic behaviour

The dynamic behaviour of this interface follows the same concept as previously explained in the FlightBright management interface; the SsrCodeMessage is OPS sector oriented and the message will only be distributed to the connected clients sharing the same OPS sector.
As such, Figure 22 only represents the nominal case for the SsrCode management service. Here, two clients are connected sharing the same OPS sector. Client #1 (OPS1 sector) requests an SSR code, for which the reply is distributed to all connected clients sharing the same OPS sector (client #2 in this case). Similary, a request to reset the code (i.e. remove from display) implements the same distribution mechanism.

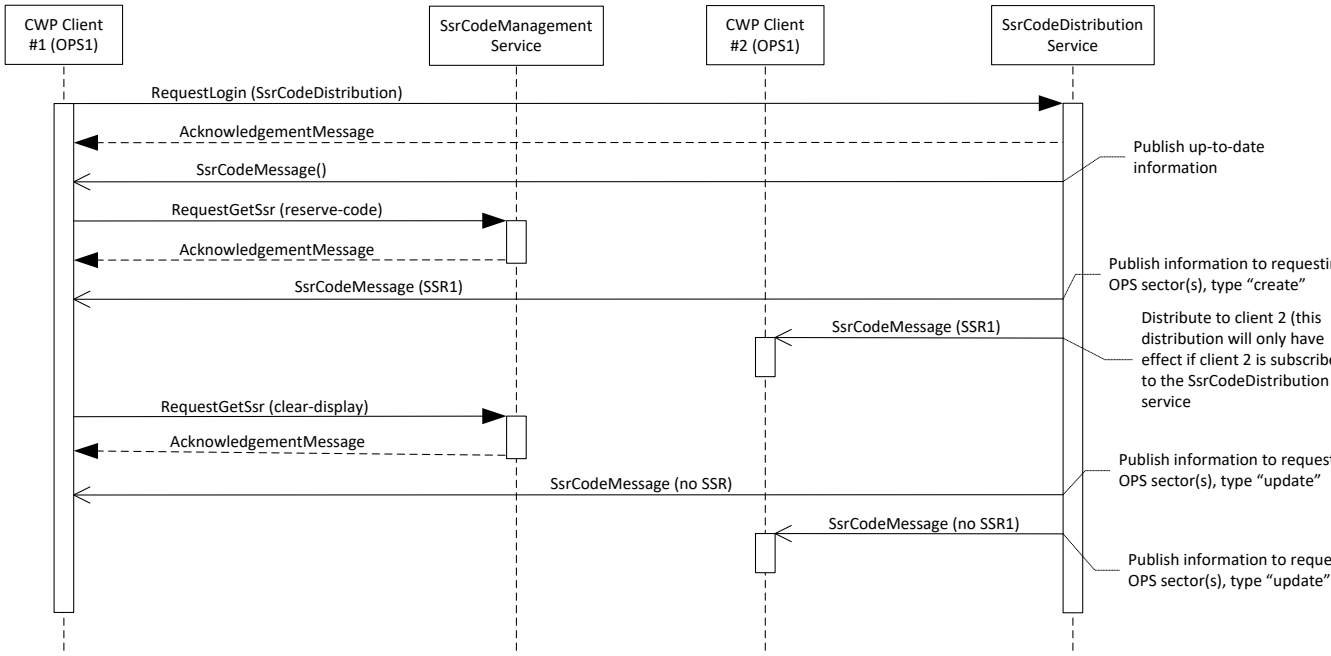


Figure 22: SsrCode Management service dynamics

4 Exchanged Information

4.1 AIRM Conformance

The attached AIRM mapping spreadsheet constitutes an evidence of the effort undertaken to find a semantic correspondence in the AIRM version 4.1.0. for each of the concepts in the ADaaS interface.

Due the innovative nature of the ADaaS project, a significant number of concepts in its interface fall beyond the current scope of the AIRM. This fact has been reflected in the mapping, as indicated by the requirement SWIM-INFO-014 of the EUROCONTROL Specification For SWIM Information Definition. A change request (AIRM Change Request #37) has been created to initiate the process of adapting the AIRM.

The AIRM tracing of the exchanged data elements can be found in Annex B.

4.2 General Message Information

4.2.1 Common Message header definition

Each message will be defined by a series of attributes and properties. This section aims to describe the common header describing each message and service.

Table 4 describes these properties which are common and shared between all messages, and indicate whether they are available at the protocol and/or application level. These properties are intended to allow the correct management of message exchange using ActiveMQ, the connector box application and the FDPS application (i.e. compare it to the global “Header” of a message). Message header properties are expected[9] to be set by the system instance sending the message.

It has to be understood that the message properties listed below are an abstraction of the actual implementation; as such properties are supported by a multitude of messaging standards.

Message properties	Meaning	Usage	Defined at	Comment
version-id	Identifier of the interface definition version used in the message.	Mandatory in all messages.	Protocol level	Interface version number used in the message. It allows the receiving system to check whether the message is coherent with the interface level used at the sender.
request-id	application message identifier	Mandatory in request messages (optional in all other messages).	Application level	Unique message identifier
reply-id	application correlation identifier	Mandatory in reply messages (not used in any other messages).	Application level	Identical to the request-id of the request message. Note: in case of no reply on the request, the client is expected to emit a new request.

[9] Any property not listed is either not set or is used with the defaults settings

Message properties	Meaning	Usage	De-fined at	Comment
reply-to	node to send replies to	Mandatory in request mes-sages.	Proto-col level	Indicates the reply queue name for a request message. Note: This queue name shall be unique for each consumer: it is considered the consumer's re-sponsibility to comply with this re-quirement.
enc-type	Encoding rule to be used (in order to extract payload from the message)	Mandatory for all messages.	Proto-col level	Indicates the encoding type (e.g. XER, BER, etc.) to allow the re-ceiver to apply the correct encod-ing rules for message payload ex-traction.
msg-type	message type	Mandatory for all messages.	Proto-col level	Identifies the message type (e.g. FlightPlanDataMessage). It allows quick checking of the type of message of any AMQP application without decoding/en-coding the message.
service-type	service type re-lated to the message	Mandatory for for messages originated from the connector box.	Proto-col level	Identifies the service type (e.g. FlightPlanDataDistribution ser-vice). It allows quick checking of the type of service of any AMQP ap-plication without decoding/encod-ing the message.
source-id[10]	creating user id	Mandatory for all messages.	Proto-col level	Originator of the message indicat-ing the type of source and unit identifier. Shall be one of the fol-lowing: SERVER.1, CLIENT.1, CLIENT.2, CLIENT.x, …
destina-tion-id[10]	destination id	Mandatory for all messages.	Proto-col level	Destination of the message, indi-cating the type of destination and unit identifier. Shall be one of the following: SERVER.1, CLIENT.1, CLIENT.2, CLIENT.x, … Unit identifier can be "all" so iden-tify all units of the same type (e.g. "CLIENT.all" in case the message is intended for all clients).

10 The source-id and destination-id are identifier for logical machines. It allows targeting specific machines, which can be useful when there is another ADSP server targeted (e.g. MUAC FDPS#2 instead of MUAC FDPS#1).

Message properties	Meaning	Usage	Defined at	Comment
source-function-id	source function identifier	Only relevant for messages originated from the clients connected to the connector box. Mandatory field.	Application & Protocol level	Originator, indicating the source operational sector, of the message. sector.role.mode.unit format (e.g. “LOW.SEC.NOR.1” for operational sector LOW, group EXECUTIVE, NORmal mode, unit 1) Sector, role, mode and unit can be “all” to identify all of the same type (e.g. “LOW.all.all.all” to target all roles, mode & units of sector LOW).
destination-function-id	destination function identifier	Only relevant for messages originated from the connector box. Mandatory field.	Application & Protocol level	Destination, indicating the target operational sector, of the message. sector.role.mode.unit format (e.g. “LOW.SEC.NOR.1” for operational sector LOW, group EXECUTIVE, NORmal mode, unit 1) Sector, role, mode and unit can be “all” to identify all of the same type (e.g. “LOW.all.all.all” to target all roles, mode & units of sector LOW).
timestamp	Indicates the timestamp at which the message was sent by the originator.	Mandatory for all messages.	Application & Protocol level	Available at AMQP level and at application level. The latter can be used to check when an operational input is made, for example.
sequence-number	sequence number	Mandatory for all messages originated from the connector box.	Application & Protocol level	AMQP Sequence number of message (represents the time-wise order in which the connector box is processing and delivering the message). ASN1 sequence number represents the feeding application (FDPS) sequence number. Note: Sequence numbers can be checked by the receiver who can then take appropriate actions. For the 1st stage, sequence number checks are not to be implemented by the client.

Message properties	Meaning	Usage	De-fined at	Comment
cycle-num-ber	Cycle number	Contains the identifier of the forthcoming ini-tialisation cycle.	Appli-cation & Pro-tocol level	Note: this is only required in case of multicast data (i.e. usage of topics) and the functionality is cur-rently not implemented.

Table 4 – Common attributes across all exchange messages

The allowed format for both the source-function-id and destination-function-id fields can be found below:

<OPS>.<Role>.<Mode>.<Unit>, where

- <OPS>: all | Operational Sector Identity
  - 1 to 6 chars to identify the operational sector identity
- <Role>: all | Role Identity
  - "sec" for executive controller
  - "sea" for executive assistance controller
  - "spc" for coordinating controller
- <Mode>: all | Mode Identity
  - "nor" for normal roles
  - "mon" for monitoring roles
- <Unit>: all | one | two

4.2.2 AcknowledgementMessage

The Acknowledgement message provides the application reply from the service provider towards service consumers for any of the operations for which a reply is sent.

The generic reply structure contains the following information:

- Success or failure of the message request
- Reason indication

Detailed information can be found directly in the ASN.1/XSD definition provided in section 3.1.6 of this document.

4.2.3 Filtering Capabilities

4.2.3.1 Message filtering by the service consumer

All messages distributed by the service to its consumers have the source-id, destination-id and destination-function-id fields populated in the header of the message (see Table 4 in section 4.2.1 of this document). The latter two items are expected to be used for message filtering purposes at the consumer side.

The destination-id and destination-function-id are both available in the protocol header, and provide a quick means to filter out any message that is not expected for the service consumer, without any explicit decoding of the complete message at application level. For example, in the

awkward case where there would be a message targeted to the wrong service consumer, the latter can easily discard the message based on the destination-id and/or destination-function-id. Additionally, the destination-function-id is also populated within the “application layer” of the header. This provides the same filtering possibilities as mentioned in the previous paragraph, but only after decoding the message contents.
A message is expected to be processed if and only if each of the sector, role, mode and unit items within the aforementioned fields is either “all” or identical to the sector, role, mode and unit as defined by the role allocated at the local client (see section 5.1.5 for more information on role allocation).

4.2.4 Message encoding (XER or BER encoding)

In order to define the messages, the ASN.1 definition language is used as a formal notation to define the OpenCWP interface. For the OpenCWP service, usage can be be made of the XER encoding (XML message format) and BER encoding. The usage of either message format is supported and can be either switched by using a configuration file, or alternatively by starting another CWP server process). For further reference, the items below aim to identify the major benefits and drawbacks of both encoding rules:

- BER encoding:
    - BER specifies how data should be encoded for transmission, independently of machine type, programming language, or representation within an application program.
    - BER is highly structured, prefixing all values with a tag and a length
    - Decoding a binary stream improves performance. Another benefit is size: a binary encoding may save up to 80% or even more relative to corresponding XML text.
- XER encoding[11,12,13]:
    - Just like BER and PER, XER also specifies how data should be encoded for transmission, independently of machine type, programming language, or representation within an application program.
    - Unlike the more compact encoding rules, XER is immediately readable.
    - XER’s reason for existence is ease of legibility (no tools are needed), but...
        - XER uses significantly more bandwidth.
        - Response times are impacted.

11 The rules to convert ASN.1 BER to XER (https://en.wikipedia.org/wiki/XML_Encoding_Rules) are standardised in X.680 and X.693.

12 The online website (http://www.obj-sys.com/webtools/asn2xsd.php) allows generating the respective XSD schema according to the standard.

13 For the project, the OSS Nokalva (XML/ASN1 toolset) is recommended.

4.3 SubscriptionManagement Messages

This section describes the Request messages. The Reply messages are described in section 4.2.2 AcknowledgementMessage.

ASN.1/XSD definition – See section 3.1.6 of this document.

4.3.1 RequestLogin message

header	See section 4.2.1
serv ice-group	See definition below
ServiceGroup ::= SEQUENCE	
Type: ServiceGroup	
flightplan-data-service	Each service field is defined by the following items: • Boolean value indicating the service is requested or not. • Additionally, optional input parameters can be set to request specific information (or trigger specific service functionality).
sector-specific-data-service	
safety-net-service	
flightplan-monitoring-service	
datalink-service	
correlation-service	
heartbeat-service	
flight-bright-service	
mtcd-service	
system-information-service	
strategic-constraint-service	
sectorisation-data-service	
ssr-code-service	
meteo-data-service	
runway-info-service	
special-area-service	
map-data-service	

When sending the RequestLogin message, the connector box knows on which queue to send the point-to-point reply and subsequent initialisation & update data (by means of the “reply-to” property in the header). At this point the role which is configured at the client is also known, as this is indicated via the header.
Based on the service-group (which identifies the services which the client is interested in) in the request, the connector box has all the required information to know which messages are to be sent over the queue of the connecting client.
Additional login parameters may be provided in the request in order to request data for a specific flight example. For this phase of the project, filtering options are limited to the combination of callsign, ADEP and ADES for the FlightPlan Data Distribution and Arrival Management Service.
An example of a login message[14] data contents is given below for the case of CWP#1 configured with LOW (executive) OPS role, requesting FlightPlanData Distribution Service:
<protocol.header field name=”VERID” value=”1”>
<protocol.header field name=”JMSReplyTo” value=”CWP1ProcessQueue”>
<protocol.header field name=”ENCTYPE” value=”XER”>
<protocol.header field name=”MSGTYPE” value=”RequestLogin”>
<protocol.header field name=”SRCID” value=”CLIENT.1”>
<protocol.header field name=”DESTID” value=”SERVER.1”>
<protocol.header field name=”SRCFUNCID” value=”LOW.SEC.all.all”>

[14] Exact example of message will be included once available.

<protocol.header field name=”DSTFUNCID” value=”all.all.all.all”>
<protocol.header field name=”JMSTimestamp” value=”20160711115814.460”>
<application.requestLogin.header field name=”request-id” value=”1”>
<application.requestLogin.header field name=”sourceFunction” value=”LOW.SEC.all.all”>
<application.requestLogin.header field name=”destinationFunction” value=”all.all.all.all”>
<application.requestLogin.header field name=”timeStamp” value=”20160711115814.320”>
<application.requestLogin.flightplan-data-service field name=”subscribe” value=”TRUE”>
If the client would wish to subscribe to only updates for flights it can provide on ore more of the optional login parameters in the request. Below example would only provide information for flights leaving from Ljubljana and going to Brussels:
<application.requestLogin.flightplan-data-service field name=”login-parameters” value=”(ADEP=LJLJ),(ADES=EBBR)”>

4.3.2 RequestLogout message

header	See section 4.2.1

When sending the RequestLogout message, the connector box will unsubscribe the client from all the services.
An example of a logout message[15] data contents is given below for the case of CWP#1 configured with LOW (executive) OPS role, requesting to unsubscribe from all services:

<protocol.header field name=”VERID” value=”1”>
<protocol.header field name=”JMSReplyTo” value=”CWP1ProcessQueue”>
<protocol.header field name=”ENCTYPE” value=”XER”>
<protocol.header field name=”MSGTYPE” value=”RequestLogout”>
<protocol.header field name=”SRCID” value=”CLIENT.1”>
<protocol.header field name=”DESTID” value=”SERVER.1”>
<protocol.header field name=”SRCFUNCID” value=”LOW.SEC.all.all”>
<protocol.header field name=”DSTFUNCID” value=”all.all.all.all”>
<protocol.header field name=”JMSTimestamp” value=”20160711115914.232”>
<application.requestLogin.header field name=”request-id” value=”2”>
<application.requestLogin.header field name=”sourceFunction” value=”LOW.SEC.all.all”>
<application.requestLogin.header field name=”destinationFunction” value=”all.all.all.all”>
<application.requestLogin.header field name=”timeStamp” value=”20160711115914.125”>

ASN.1/XSD definition – See section 3.1.6 of this document.

15 Exact example of message will be included once available.

4.4 DataDistribution Messages

This section describes the messages distributed to the service consumers. No reply expected.

ASN.1/XSD definition – See section 3.1.6 of this document.

4.4.1 FlightPlanDataMessage

The following section describes the data items provided in the FlightPlanDataMessage. Items that are strikethrough are part of the definition, but will not be distributed in the 1st implementation of the connector box for the ADaaS project, because they will either not be triggered (i.e. the corresponding request action is not in scope of the project) or they are not operationally used (i.e. not required for Slovenia Control operations).

header	See section 4.2.1
message-type	Create, update or cancel message.
FlightPlanIdentification	
FlightPlanIdentification.PlanIdentifier	
plan-number	Unique plan identifier (used to refer to the flight plan in all other messages). Assigned by the server.
~~flight-identifier~~	~~Unique flight identifier: it can be an IFPL identifier, aircraft registration, a flight number, etc.~~
~~global-flight-identifier~~	~~GUFI~~
what-if	Boolean indicating whether the distributed flightplan data refers to a what-if action performed on the flight or not.
FlightLeg state information	
FlightLegStateinformation.FlightLegStatus	
Current	Set when the relevant information in the Flight-PlanDataMessage is sent for the current leg in the system.
Next	Set when the relevant information in the Flight-PlanDataMessage is sent for the next leg in the system (case of multiple legs) or when the 1st leg is not yet under-control.
in-sequence	Set when the relevant information in the Flight-PlanDataMessage is sent for a one (or more legs) located behind the next leg in the system.
FlightLegStateinformation.FlightLegState	
actual-live	Value set in case the related flightplan is activated (either by the reception of an OLDI coordination message, or by manual action, for example Operation manageNFL or Operation changeEntryCoordination)
actual-left	Value set in case the related flightplan has been handed-over and is a time-parameter after the center's AoR (or when the flightplan leg has been manually cancelled upon input, for example by Operation deleteFPL).
terminated	Value set in case the related flightplan has been excluded for processing (i.e. at the end of the flight), and is soon to be deleted from the system. The value is set when the flight moves from actual-left to terminated state.
pre-warning	Set for flightplans which are distributed: • prior being activated (i.e. actual-live), • VFR/IFR transition flights for which the IFR portion is not yet activated.
GeneralInfo	
GeneralInfo.FlightPlanKeys	

callsign	Aircraft identifier
adep	ICAO identifier of the departure airport
ades	ICAO identifier of the destination airport
eobd	Estimated off-block date
eobt	Estimated off-block time
GeneralInfo.FlightPlanInitialData	
flight-rules	Flight rules as indicated in the flightplan.
flight-type	Flight type as indicated in the flightplan.
number-of-aircraft	Number of aircraft in the flight
aircraft-type	ICAO identifier of the aircraft type
wake-turbulence-category	ICAO (RECAT) wake turbulence category identifier
nca-equipment	Navigation and Communication information (as filed in the flightplan)
ssr-ads-equipment	SSR and ADS equipment as filed in the flightplan
true-airspeed	True airspeed as filed in the flightplan (F15a)
requested-flight-level	Requested flight level (RFL) as filed in the flightplan (F15b)
field-18	Contents of the field 18 as filed in the flightplan (F18)
free-text	Free text that can be manually included as a result of a controller (Operation createModifyFPL) or flight data operator action.
rvsm-status	Indicating RVSM status: exempt, equipped and/or capable
status-833	Indicating 833KHz status: equipped or exempt
uhf-equipped	Indicating whether flight is UHF equipped or not
brnav-equipped	Boolean indicating if aircraft is B-RNAV equipped or not.
prnav-equipped	Boolean indicating if aircraft is P-RNAV equipped or not.
modes-capability	Indicating ModeS capability: equipped, not or unknown.
field-18-rte	String of information as indicated in the F18 RTE subfield.
field-18-rmk	String of information as indicated in the F18 RMK subfield.
field-18-msg	String of information as indicated in the F18 MSG subfield.
field-18-mdcn	String of information as indicated in the F18 MDCN subfield.
field-18-sts	String of information as indicated in the F18 STS subfield.
field-18-rfp	String of information as indicated in the F18 RFP subfield.
field-18-ifp	String of information as indicated in the F18 IFP subfield.
field-18-typ	String of information as indicated in the F18 TYP subfield.
field-18-dep	String of information as indicated in the F18 DEP subfield.
field-18-dest	String of information as indicated in the F18 DEST subfield.
field-18-altn	String of information as indicated in the F18 ALTN subfield.
field-18-dof	String of information as indicated in the F18 DOF subfield.
field-18-reg	String of information as indicated in the F18 REG subfield.
field-18-cod	String of information as indicated in the F18 CODE subfield.
stay-info	Information structure related to DLE and STAYINFO information.
~~eet~~	~~Estimated elapsted time of the flight~~
~~eet-data~~	~~String of information indicating EET data~~
~~route~~	~~Route as indicated in the F15c of the flightplan~~
GeneralInfo.FlightPlanGroundData	
~~is-transferred-to-tower~~	~~Boolean indicating if a flight is passed from GROUND control to TWR control. Set to TRUE upon manual action.~~
~~departure-notification-initiator~~	~~Identifies the OPS position that initiates the request for departure clearance process.~~

~~departure-notification-receiver~~	~~Identifies the OPS position that is expected to provide the departure clearance. Entered manually by the notification initiator.~~
~~clearance-passed-indicator~~	~~Indicates if the departure clearance has been passed to the aircraft or not. Entered by manual input.~~
estimated-start-movement-time	The estimated time & date the aircraft is expected to start movement associated with departure.
actual-start-movement-time	The actual time & date the aircraft has vacated the parking position.
~~ground-status~~	~~Indicates the Aircraft Flight Status (related to A-CDM) of the aircraft on the ground.~~
GeneralInfo.FlightPlanDepartureData	
~~estimated-time-departure~~	~~The estimated time & date of departure (from the runway), for departure flights, indicated by the TWR ATCO.~~
actual-time-departure	The actual time & date of departure, set upon reception of the DEP message.
~~departure-runway~~	~~The departure runway allocated by the ATCO.~~
sid	The SID allocated by default by the system or filed in the flightplan.
~~allocated-sid~~	~~The SID allocated by the ATCO when giving the departure clearance.~~
departure-clearance	Indicating the type of clearnance (e.g. radar release), departure clearance level and clearance status (e.g. pending). For the initial phase of the project, only the departure clearance level will be populated.
~~takeoff-clearance~~	~~Indicating whether take-off clearance has been verbally passed to the pilot or not.~~
~~departure-gate~~	~~Gate (or parking position) and terminal at which the passengers access for boarding.~~
~~deicing-position~~	~~Aircraft stand position at which the de-icing procedure takes place.~~
~~departure-status~~	~~Indicates the Aircraft Flight Status (related to A-CDM) of the aircraft relevant for departure.~~
GeneralInfo.FlightPlanArrivalData	
~~arrival-runway~~	~~The arrival runway allocated by the ATCO.~~
alternate-aerodrome-1	The 1st alternate aerodrome of the flight.
alternate-aerodrome-2	The 2nd alternate aerodrome of the flight.
estimated-time-arrival	The estimated time & date of arrival of the flight at the destination airport.
actual-time-arrival	The actual time & date of arrival of the flight.
star	The STAR allocated by default by the system or filed in the flightplan.
~~allocated-star~~	~~The STAR allocated by the ATCO when determining the STAR.~~
~~approach-type~~	~~The type of approach (e.g. ILS, NDB, etc.) that the aircraft will perform. Entered by manual input.~~
people-on-board	The number of people on board. Entered by Operation changeAcStatus.
~~airport-action~~	~~The action(s) to be performed at the airport (e.g. touch-and-go, full stop, etc.). Entered by manual input.~~

~~weather-passed-indicator~~	~~Indicating the letter designation (e.g. *bravo*), from the ICAO spelling alphabet, of the ATIS confirmed to be received by the pilot (or indicated to the pilot).~~
~~arrival-coor-twr~~	~~Indicating whether verbal coordination between approach and TWR has been performed or not.~~
~~circling~~	~~Indicates if the aircraft performs a circling approach. Entered by manual input.~~
~~cleared-to-land~~	~~Boolean indicating whether the aircraft is cleared to land.~~
~~arrival-gate~~	~~Gate (or parking position) and terminal at which the passengers disembark.~~
~~arrival-status~~	~~Indicates the Aircraft Flight Status (related to A-CDM) of the aircraft relevant for arrival.~~
GeneralInfo.FlightPlanInternalData	
flight-plan-type	SFPL (System Flight Plan) or ASPL (Abbreviated flight plan). For the latter limited information will be distributed.
current-flight-rules	VFR or IFR. Dynamically switches along the course of the flight.
current-traffic-class	GAT, OAT or unknown. Dynamically switches along the course of the flight.
flight-plan-state	Status to categorize a flightplan from the point of view from the system. Useful to be able to provide different functionality according to the rules specified by each state.
present-ssr-code	Present assigned code (PSSR)
next-ssr-code	Next assigned code (NSSR)
downstream-ssr-code	Downstream SSR code (DSSR)
route-encroachment-SSR-alert	When set to TRUE, indicates that two flights with the same SSR code are too close.
radio-callsign	Radio Callsign of the Airline Company identified from the flightplan callsign.
GeneralInfo.FlightPlanAdditionalData	
corrupt-fssa	Indicating that the ModeS FSSA read-out is unreliable. Set to TRUE upon manual Operation changeAcStatus.
is-avoiding-weather	Boolean indicating a flight is circumnavigating due to weather. Set to TRUE upon manual Operation changeAcStatus.
is-fuel-dumping	Boolean indicating a flight is dumping fuel (case of emergency). Set to TRUE upon manual Operation changeAcStatus.
~~qnh-passed~~	~~The QNH value as passed verbally to the pilot. Entered upon manual input.~~
non-deviation-status	Indicates that the controller responsible for the flight guarantees that the aircraft will stay on its assigned pattern and at its assigned level and will not deviate. Set to TRUE upon manual Operation changeAcStatus.
~~fp-creation-time~~	~~The time at which the flightplan is created in the local system.~~
~~flight-category~~	~~Category of the flight: inbound/outbound/overflight/domestic~~
~~GeneralInfo.FlightPlanMilitaryData~~	

~~flight-accepted~~	~~Indicating whether a flight is accepted or not. (used for Military operations)~~
~~area-monitor~~	~~Indicating whether a flight is monitored or not. (used for Military operations)~~
~~accept-sector~~	~~The sector that accepts the flight (ref. flight-accepted field).~~
~~arr-rapcon-accepted~~	~~Indicating whether a flight is accepted by arrival or rapcon sectors. (used for Military operations)~~
~~civ-notification~~	~~Identifier of the civil center that is informed verbally by the military controller about a certain flight. (used for Military operations)~~
~~informed-centres~~	~~Identifier of the additional civil or military centers that are informed verbally by the military controller about a certain flight. (used for Military operations)~~
~~tma-crossing-approval~~	~~Indicating whether a flight can traverse a TMA (while not being controlled by the TMA ATCO).~~
~~twr-distance~~	~~The distance to go until reaching the threshold. Entered by manual input. (used for Military operations)~~
~~formation~~	~~Indicates the type of formation. Entered by manual input. (used for Military operations)~~
~~military-approval~~	~~Indicates whether military approval for such flight has been granted or not (placeholder structure for future Military operations).~~
GeneralInfo.EmergencyData	
~~communication-defect~~	~~Attribute to represent an issue in the communication between aircraft and ground.~~
~~emergency-description~~	~~Attribute to represent additional textual information describing the emergency situation & reason.~~
~~emergency-phase~~	~~Enumerated value indicating:~~ ~~• Uncertainty phase~~ ~~• Alert phase~~ ~~• Distress phase~~
~~emergency-status~~	~~Enumerated value indicating the emergency is:~~ ~~• None (not declared)~~ ~~• Declared~~ ~~• Cancelled~~
~~last-contact-frequency~~	~~Transmitting or receiving frequency of the last two-way contact with the aircraft.~~
~~time-last-contact-frequency~~	~~Time & date of the last two-way contact with the aircraft.~~
~~last-reported-position~~	~~Last reported aircraft position.~~
~~time-last-reported-position~~	~~Time & date of last reported aircraft position.~~
~~last-track-position~~	~~Last observed aircraft position~~
~~time-last-track-position~~	~~Time & date of last observed aircraft position.~~
ControlData	
ControlData.InternalClearances	
level	Cleared flight level information (entered by Operation manageCFL). Only available in case the flight is actually under control (and is not operating in a blocked level band). Includes the time at which the operational instruction was entered.

blocked-level-band	Indicates the lower and upper level (block) in which the aircraft operates (entered by Operation manageCFL). Can only be populated in case there is no cleared flight level and if manually input. Includes the time at which the operational instruction was entered.
~~attitude~~	~~Attitude indicator. Only available after manual input (i.e. case where the attitude is manually entered in case of flight without mode C).~~ ~~Includes the time at which the operational instruction was entered.~~
heading	Cleared heading information (entered by Operation manageHeading). Includes the time at which the operational instruction was entered.
~~holding~~	~~Holding information.~~ ~~Includes the time at which the operational instruction was entered.~~
speed	Cleared speed information (entered by Operation manageSpeed). Includes the time at which the operational instruction was entered.
~~offset~~	~~Offset route information.~~ ~~Includes the time at which the operational instruction was entered.~~
~~dct~~	~~DCT information.~~ ~~Includes the time at which the operational instruction was entered.~~
ControlData.CoordinationControlData	
under-local-control	Indicator whether or not the flight is under control of the local center. Set to TRUE upon first assume of the flight (Operation manageControl). Set to FALSE upon last sector cancel assume.
inconsistent-under-control-sector	Indicator whether or not the flight's position is no longer in the under-control sector area of responsibility (after re-sectorisation).
current-frequency	Indicates the current frequency on which the flight is.
inconsistent-frequency	Indicates that the flight is probably on the incorrect frequency (after a re-sectorisation).
previous-sector-id	Name of the operational sector (internal) that was previously under control of the flight.
~~release-indication~~	~~Specify whether the flight is released in the transferring sector (climbing, descending, turning or full), optionally including the release level.~~
next-sector-constraints	Information available once the flight is assumed by the local center. Contains the constraints requested by the next internal sector: • Requested point, • requested heading, • requested vertical rate climb/descend, • Requested speed.

~~ControlData.MilitaryData~~	
~~target-level~~	~~The target-level is defined looking to the level where another intention (climbing/descending) starts limited to current and next sector.~~
~~cleared-target-level~~	~~After timer expiry (2min), the target-level is copied into the cleared-target-level. Only set when the flightplan state is actual-live.~~
TrajectoryData	
TrajectoryData.RoutePointList	
name	Name of the point
type	Indicator of the point type: bearing/distance, point, lat/lon, airport, etc.
position	Position of the point expressed in x/y-coordinates
quality	Bit String indicating: 0. entry in the AoR 1. bottom-of-climb 2. exit in the AoR 3. bottom-of-descend 4. manoeuvre start 5. manoeuvre end 6. fdm-point: whether a point is to be considered as FDM is defined in adaptation data. It allows to display the point or not at the client side. 7. dct-intermediate-point: set for the intermediate point(s) received in either the Operation manageDCT or Operation changeRoute 8. dct-end-point: set for the end point received in either the Operation manageDCT or Operation changeRoute 9. top-of-climb 10. top-of-descend 11. special-area-entry-point 12. special-area-exit-point 13. strategic-constraint-application-point 14. gat-traffic 15. outside-aor 16. significant-point: whether a point is to be considered as significant is defined in adaptation data. It allows to display the point or not at the client side. 17. on-route 18. preferred: the automatic CDR change or alternative route is not allowed when the route is flagged as preferred. 19. cdr-substituted-by-alternative: set when the CDR sequence will be replaced by the alternative route. 20. cdr-not-useable: set when a non-available CDR is considered as non-replaceable when there is no alternative route defined in the adaptation data. 21. cdr-substituted-by-shortening-route: set when a CDR that has been classified as insertable CDR will be automatically inserted in place of the corresponding longer portion of route if it has not yet been flown over.

	22. non-available-cdr-over-preferred: set when a non-available CDR in which at least one segment has been declared as preferred is found in the route. 23. cleared-indicator 24. aoi-entry-point 25. aoi-exit-point 26. transition-route-point 27. sid-end-point 28. star-initial-point 29. ops-sector-crossing 30. ifr-vfr-transition 31. vfr-ifr-transition 32. oat-gat-transition 33. gat-oat-transition 34. stay-point 35. dle-point
calculated-level	Level as calculated by the server at the position
aircraft-mass	Mass as calculated by the server at position
calculated-heading	Expected aircraft heading as calculated by the server at position
tas	TAS value as calculated by the server
pfl	The planned flight level at the point location
pfl-status	Indicates if the pfl was entered manually (Operation managePFL) or automatically calculated.
rfl	The requested flight level at the point location
reference-leg	Identifies whether the point is located in the current, next or in-sequence leg.
TrajectoryData.RoutePointETOs	
time	The calculated time over the point location
TrajectoryData.AdditionalData	
next-route-point-not-overflown	Index in the route-point-list indicating the next point that is to be overflown
after-bpx-xcop	The first fixpoint after the boundary exit point
auto-accepted-modification	Index to the trajectory, indicating the point where the system has automatically re-routed the flight to.
TrajectoryData.TSACrossingList	
tsa-crossing-list	List of TSAs entered by the trajectory, including the entry and exit reference to the trajectory.
TrajectoryData.RapconCrossingList	
rapcon-crossing-list	List of RAPCON areas entered by the trajectory, including the entry and exit reference to the trajectory.
TrajectoryData.OriginalRouteData	
original-route-data	2D trajectory list of points, indicating the start and end of diversion. Indicated the part of the route that changed due to a manual route change input (e.g. Operation manageDCT or Operation changeRoute).
~~route-change-indicator~~	~~Is a flag indicating that a part of the route has changed upon GRM input. This flag is filled searching through the points on the route and checking that the flag "cleared-indicator" has~~

	~~changed. The route-change-indicator is from the whole flight plan, and the cleared-indicator is from each point.~~
TacticalTrajectory	
ObservedTrajectoryData.ObservedTrajectoryPoint	
position	Position expressed in x/y-coordinates
level	Level as calculated by the server at the position
eto	The calculated time over the point location
~~UserPreferredTrajectoryData~~	
~~Represents the trajectory as preferred by the airspace user.~~ ~~Same structure as for TrajectoryData, however the OriginalRouteData will never be provided.~~	
SectorSequenceInfo	
SectorSequenceInfo.FlightSectors	
identity	Identifier of the crossed flight sector.
frequency	Frequency information of the related flight sector.
entry	Reference to the trajectory route-point-list where the flight sector is entered.
exit	Reference to the trajectory route-point-list where the flight sector is no longer crossed.
SectorSequenceInfo.ControllerOperationalSectors	
identity	Identifier of the crossed controller operational sector.
range-id	Unique identifier that identifies the operational sector in the total sector sequence (i.e. it allows to identify the operational sector's instance in case that sector is crossed multiple times).
entry	Reference to the flight sector (SectorSequenceData.FlightSectors) where the operational sector is starting.
exit	Reference to the flight sector (SectorSequenceData.FlightSectors) where the operational sector is ending.
SectorSequenceInfo.PlannerOperationalSectors	
identity	Identifier of the crossed planner operational sector.
range-id	Unique identifier that identifies the operational sector in the total sector sequence (i.e. it allows to identify the operational sector's instance in case that sector is crossed multiple times).
entry	Reference to the flight sector (SectorSequenceData.FlightSectors) where the operational sector is starting.
exit	Reference to the flight sector (SectorSequenceData.FlightSectors) where the operational sector is ending.
CoordinationDataList List for current, next and in-sequence leg (max. 3) containing detailed information on the current status of coordination between two internal controllers (upstream and downstream controllers) or an external centre and the first internal controller or the last internal controller and an external centre.	
CoordinationData .reference-leg	Identifies for which leg the coordination data is provided. In the demonstrator of this project, this value will always be set to "current".
CoordinationData.aor-entry-data Detailed information on the current status of coordination between the upstream external partner and first internal controller.	
expiration-time	Not applicable for external entry coordination.

coordination-partner	This information refers to the upstream center. Coordination partner structure: o center identity, o internal/external dialogue classification, o data source: derived from trajectory, derived from the LoA or manually entered (e.g. Operation manageNFL).
coordination-evolution	Summary of the status and actions pending to the controller for this coordination: o Verbal-coordination-done: o When TRUE, protocol-state does not apply. o Flag is reset when coordination progresses (e.g. a new dialogue is opened). o Requires-verbal-action is set when: o Electronic means to perform coordination are not available (i.e. protocol problems) o Electronic dialogue to get the initial agreed data failed (i.e. rejection by transferring/receiving sector, operational timeout in pre-agree sub-phase)
coordination-phase	See description for aor-exit-data.
coordination-subphase	See description for aor-exit-data.
dialogue	See description for aor-exit-data.
protocol-state	See description for aor-exit-data.
agreed-data	The agreed-data during the notification phase represents the data as sent in the ABI message (they are removed when the Coordination phase starts). The change-indicators at entry serve to show which agreed-data has changed through external coordination (e.g. REV). This means that upon initial coordination no change-indicators apply. All changed items (COP, ETO, TFL, TSFL, RVSM, SSR, UHF, 8.33kHz & Aircraft Type) are identified through the change-indicators. RVSM, SSR, UHF, 8.33kHz and Aircraft Type (for which no negotiating-data structure exists) are updated in the flight-plan regardless if the external message leads to a dialogue for the other items. As such, their change-indicators are set upon receipt on any external message (or manual input) received after coordination has started. The change-indicators related to the AoR entry are never reset.
data-being-negotiated	Negotiating-data represents the coordination data that is being negotiated.

	In the case the dialogue is closed unsuccessfully (i.e. rejected or operational time-out); the negotiating-data is kept. This negotiating-data is cleared when: o agreed-data is updated (i.e. new agreed data) o new negotiating-data is to be created (i.e. new dialogue) o verbal-coordination is done (i.e. manual action like Operation changeExitCoordination)
calculated-coordination-data	These are the coordination data derived from the calculated trajectory taking considering Manual input or LoA data. For the external entry coordination, the data is maintained until the applicable boundary is overflown.
trajectory-derived-data	These are the coordination data derived from the calculated trajectory without considering Manual input or LoA data. For the external entry coordination, the data is maintained until the applicable boundary is overflown.
tentative-data	The tentative data for external entry coordination is only used in the case where an entry revision (RRV) triggers a different entry OPS sector (than the one previously determined and agreed). The related new negotiating-data is sent in the data-being-negotiated field.
cancel-coordination-data	Not applicable for external entry coordination.
frequency	The coordination entry frequency is determined as follows: o Initially set to main frequency of geographically determined External Flight Sector, o Can be modified by the controllers during the coordination process using RequestCoordination or RequestFrequency message. o Can be overwritten by the frequency received in the OLDI COF message.
CoordinationData.aor-exit-data Detailed information on the current status of coordination between the downstream external partner and last internal controller.	
expiration-time	This information is applicable when there is an open dialogue. It is the time left (in 10 seconds granularity) when open coordination dialogue will expire (operational time-out).
coordination-partner	This information refers to the downstream center. Coordination partner structure: o center identity, o internal/external dialogue classification, o data source: derived from trajectory, derived from the LoA or manually entered.
coordination-evolution	Summary of the status and actions pending to the controller for this coordination: o Number of minutes to send the ACT (this countdown starts from 10 minutes prior ACT transmission. o Verbal-coordination-done:

	○ When TRUE, protocol-state does not apply. ○ Flag is reset when coordination progresses (e.g. a new dialogue is opened). o Requires-verbal-action is set when: ○ Electronic means to perform coordination are not available (i.e. protocol problems) ○ Electronic dialogue to get the initial agreed data failed (i.e. rejection by transferring/receiving sector, operational timeout in pre-agree sub-phase) o Expedite indicator o Suspended indicator
coordination-phase	Information on the current coordination phase: Notification, Coordination, Transfer or Transferred.
coordination-subphase	This information is only relevant when the coordination-phase is Coordination. Following information is distributed: o None o Pre-agree: there is an open dialogue and there is not yet agreed data. o Agree: there is an agreement between the upstream and downstream center. o Post-agree: there is an open dialogue and there is previously agreed data.
dialogue	This information is only relevant when the coordination-phase is Coordination. Following information is distributed: o dialogue-is-active: there is an open dialogue o dialogue-opened-by: who opened the dialogue (upstream/downstream centre) o centre-with-pending-action: who has pending action (upstream/downstream controller/centre) o post-agree-cause: cause for closing the dialogue. When a dialogue is closed, the last three data items remain as they are for history.
protocol-state	This information is only relevant when the coordination-phase is Coordination. Following information is distributed: o None o Ack-awaiting: coordination message has been sent out, but no reply has been received yet and operational timer has not yet expired. o Ack-received: coordination message has been sent out, and reply has been received in time. o Ack-timed-out: coordination message has been sent out, but no reply has been received in due time.

agreed-data	The agreed-data during the notification phase represents the data as sent in the ABI message (they are removed when the Coordination phase starts). For items COP, ETO, TFL & TSFL the change-indicators serve to show which agreed data has changed through external coordination (e.g. REV). For items SSR, RVSM, 8.33kHz, UHF & Aircraft Type the change-indicators refer to the fact that verbal coordination of these items is required (i.e. verbal coordination is required and the data is changed in comparison to what was sent earlier to the partner). This means that if electronic coordination of changes to these items (i.e. changes since the start of exit coordination) is successful, they are not set (because the exit partner received the changed data). Change indicators related to the exit coordination can be reset upon making a RequestCoordination and upon accepting a counterproposal from the downstream partner.
data-being-negotiated	Negotiating-data represents the coordination data that is being negotiated. In the case the dialogue is closed unsuccessfully (i.e. rejected or operational time-out); the negotiating-data is kept. This negotiating-data is cleared when: o agreed-data is updated (i.e. new agreed data) o new negotiating-data is to be created (i.e. new dialogue) o verbal-coordination is done (i.e. manual action like Operation changeExitCoordination)
calculated-coordination-data	These are the coordination data derived from the calculated trajectory taking considering Manual input or LoA data. For the external exit coordination, the data is maintained until the flight is transferred.
trajectory-derived-data	These are the coordination data derived from the calculated trajectory without considering Manual input or LoA data. For the external exit coordination, the data is maintained until the flight is transferred.
tentative-data	Not applicable for external exit coordination.
cancel-coordination-data	This data is created when exit coordination was already done with one external partner, but now the partner has changed. It refers to the old coordination data including MAC (if supported).
frequency	Initially set to main frequency of geographically determined External Flight Sector; can be modified manually (see Operation).
CoordinationData.first-current-internal-data	

Detailed information on the current under-control sector (or the first internal sector if not yet under-control) internal coordination data.	
range-id	RangeID has to be an increasing number along the progress of the flight (see section 5.1.8 for more details). Discontinuities within the RangeID sequence are allowed. The sector RangeId, before assume is the RangeId of the entry sector, after assume it is the RangeId of the under-control sector.
pfl	Refers to the first PFL in the sector. When the application point is overflown, refers to the next PFL (when there are multiple PFLs). When the flight is leveled in the whole sector or there is a CFL, this field is null. The PFL only comes from an ECL (automatic) or from Operation managePFL (manual).
pfl-status	Indicates if the PFL is due to manual or automatic action.
manual-ecl	Value of the ECL (introduced by Operation manageECL). Refers to the first ECL in the sector. When the application point is overflown refers to the next ECL.
multiple-pfls	"True" when there are more than one PFL in the sector. When there is only one PFL left in the sector, the flag will NOT be set anymore.
controller-id	Executive Controller Logical Position Display Name
sector-id	OPS Sector Display Name
internal-coordination-data	Contains specific information on the coordination for the sector. The following information is distributed: o sector-substate: o null o normal: flight is in NORMAL mode (i.e. default state) o manual: flight is put in MANUAL mode (i.e. no more conformance monitoring) o hold: flight is in holding pattern o monitored: used for flights which are given advisory or information services (but not under-control), for example VFR flights. o suspended: used for flights which are leaving the area for a while, but are known to return later o internal-hand-over-state: o null o HOP: Hand-over is proposed o HOA: Hand-over is accepted o XASM: flight is transferred to next sector o time-of-assume: o null: not assumed by sector o value: time + date when assume was done by the sector o sector-control-status:

	o current: flight is under-control of sector o next: flight is not yet under-control of sector, but the sector is the next in sequence o in-sequence: flight is not yet under-control of sector, but the sector is one of the downstream sectors (but not the next) o sector-boundary-status: o ES (Entry Sector): previous sector is external, next sector is internal o SS (Sector Sector): previous and next sector are internal o SX (Sector Exit): previous sector is internal, next sector is internal o EX (Exit Sector): previous and next sector are external
CoordinationData.first-current-exit-data Detailed information on the current under-control sector (or the first internal sector if not yet under-control) exit coordination data.	
expiration-time	
coordination-partner	This information refers to the downstream center or internal sector (whichever applicable). Coordination partner structure: o center or sector identity, o external/internal dialogue classification, o data source: derived from trajectory, derived from the LoA or manually entered.
coordination-evolution	Summary of the status and actions pending to the controller for this coordination: o Verbal-coordination-done: o When TRUE, protocol-state does not apply. o Flag is reset when coordination progresses (e.g. a new dialogue is opened). o Requires-verbal-action is set when: o Electronic means to perform coordination are not available (i.e. protocol problems) o Electronic dialogue to get the initial agreed data failed (i.e. rejection by transferring/receiving sector, operational timeout in pre-agree sub-phase)
coordination-phase	Information on the current coordination phase: Notification, Coordination, Transfer or Transferred.
coordination-subphase	This information is only relevant when the coordination-phase is Coordination. Following information is distributed: o None o Pre-agree: there is an open dialogue and there is not yet agreed data. o Agree: there is an agreement between the upstream internal and downstream internal/external sector.

	o Post-agree: there is an open dialogue and there is previously agreed data.
dialogue	This information is only relevant when the coordination-phase is Coordination. Following information is distributed: o dialogue-is-active: there is an open dialogue o dialogue-opened-by: who opened the dialogue (upstream/downstream centre/controller) o centre-with-pending-action: who has pending action (upstream/downstream controller/centre) o post-agree-cause: cause for closing the dialogue. When a dialogue is closed, the last three data items remain as they are for history.
protocol-state	This information is only relevant when the coordination-phase is Coordination. Following information is distributed: o None o Ack-awaiting: coordination message has been sent out, but no reply has been received yet and operational timer has not yet expired. o Ack-received: coordination message has been sent out, and reply has been received in time. o Ack-timed-out: coordination message has been sent out, but no reply has been received in due time.
agreed-data	The agreed-data during the notification phase represents the coordination data as determined by the FDPS system between internal sectors. In case the next sector is external the same logic as described in aor-exit-data field is used. For internal sectors, the change-indicators serve to show which agreed-data has changed through internal inputs. These items are identified through the change-indicator for the following items: COP, ETO, TFL & TSFL. Change-indicators related to internal entry coordination are never reset. Change-indicators related to internal exit coordination are reset upon making an RequestCoordination (with option COORD) action.
data-being-negotiated	The data-being-negotiated structure represents any coordination data that is currently being negotiated, either proposed/counter-proposed or to be verbally agreed. In case that a dialogue is closed by a rejection or operational time-out, the negotiating-data is kept. The presence of the negotiating data will help the ATCO in subsequent verbal coordination. This “old” negotiating data is cleared when the

	coordination progresses after the dialogue was closed unsuccessfully. This is when the agreed-data is updated, when new negotiating-data is to be created, or following any manual request with COORD indication (meaning that verbal coordination was performed).
calculated-coordination-data	The calculated-coordination-data structure represents the coordination data derived from the trajectory, but with considering applicable LoA constraints.
trajectory-derived-data	The trajectory-derived-data structure represents the coordination data derived from the trajectory, but without considering any manual request or applicable LoA constraints.
cancel-coordination-data	The cancel-coordination-data structure is created when an external coordination was already performed, but now the partner has changed. This to represent a typical MAC case.
frequency	Initially set to main frequency of geographically determined External/Internal Flight Sector; can be modified manually (see Operation).
CoordinationData.next-sector-internal-data Detailed information on the next under-control sector internal coordination data.	
Same description as for first-current-internal-data.	
CoordinationData.next-sector-exit-data Detailed information on the next under-control sector exit coordination data.	
Same description as for first-current-exit-data.	
ArrivalManagerData Identifies the information received from downstream arrival manager.	
control-center-id	Identifier of the ATC unit providing the arrival management service for the destination aerodrome.
metering-fix	The metering fix (MFX) for which the arrival data applies.
time-over-metering-fix	The proposed time at which the flight should arrive at the MFX as calculated by the arrival management process.
total-time-to-lose	The time adjustment (lose) required at the metering fix.
total-time-to-gain	The time adjustment (gain) required at the metering fix.
cop	Coordination point (COP)
aman-time	The proposed time at which the flight should arrive at the COP as calculated by the arrival management process.
assigned-speed	The speed advisory as proposed by the arrival management service unit.
route	The requested routing as proposed by the arrival management service unit.
change-indicators	Indicates which fields have changed compared to the previous AMA reception.
InformationTaskData Structure for military information received from military partners.	
~~executive~~	~~Indicator of the external military executive sector currently under control of the flight~~
~~external-level~~	~~Indicator of the target level (i.e. level to go to) as instructed by the external military executive controller~~
~~release-field~~	~~Indicates if the flight was released by the last external military executive controller.~~
CorrelationInfo	

Track number	Track identifier (-1 means the flightplan is not correlated)
Linkage status	Enumerated indicating the linkage status of the flight plan: • uncorrelated: flight plan has not yet been correlated • correlated: flight plan correlated with flight plan • manually-decorrelated: flight plan was manually de-correlated (Operation link); • correlation loss due to loss of track information; • correlation loss due to Mode-A mismatch; • correlation loss due to Mode-S mismatch.
ambiguity	Indicates ambiguities resulting from the correlation/de-correlation (e.g. mode A mismatch, track ambiguity, etc.)

4.4.2 SectorSpecificDataMessage

The following section describes the data items provided in the SectorSpecificData message.

header	See section 4.2.1
message-type	Create, update or cancel message.
FlightPlanIdentification	
FlightPlanIdentification.PlanIdentifier	
plan-number	Unique plan number (used to refer to the flightplan in all other messages). Assigned by the server.
~~flight-identifier~~	~~Unique flight identifier: it can be an IFPL identifier, aircraft registration, a flight number, etc.~~
~~global-flight-identifier~~	~~GUFI~~
what-if	Boolean indicating whether the distributed flightplan data refers to a what-if action performed on the flight or not.
RangeId	
range-id	Unique identifier that identifies the operational sector in the total sector sequence (i.e. it allows to identify the operational sector's instance in case that sector is crossed multiple times).
SectorStatus	
sector-status	Indicates the status of the OPS sector for which the information is distributed: • normal • skipped (this OPS sector has been skipped) • bypassed (the upstream OPS sector bypassed this OPS sector) • short-crossed (this OPS sector is crossed less than a relevant distance) • handing-over (this OPS sector is handing-over the flight) • receiving-hand-over (this OPS sector is receiving the hand-over of the flight) • mac-back-to-pending (MAC processed for this sector and its state moves to LEFT) • mac-to-left (MAC processed for this sector and its state moves to LEFT) • tentative (used when the current OPS sector hand-over proposes the flight to a sector not in the sequence. For

	the current sector, the tentative coordination then represents the coordination data with the new involved sector) • normal-bypass-done (this OPS sector is bypassed by the upstream OPS sector) • normal-next-skipped (the downstream OPS sector has been skipped) • normal-next-short-crossed (the downstream OPS sector is crossed less than a relevant distance)
Planner Accessibility	
entry	OPS sector identification of the planner upstream the current OPS executive sector.
internal	OPS sector identification of the planner related to the current OPS executive sector.
exit	OPS sector identification of the planner downstream the current OPS executive sector.
Event Trigger	
executive	Display trigger for controller OPS Sector
planner	Display trigger for planner OPS Sector
entry-passed	Display trigger for to indicate controller OPS sector entry has been passed
holding	Display trigger to indicate flight is in holding in all distributed SectorSpecific data messages (not only to one distributed to the OPS sector where the flight is in hold)
SectorEntryData Detailed information on the sector upstream internal or external coordination data.	
Same description as for aor-entry-data (Flight Plan Data Distribution Service).	
SectorInternalData Detailed information on the sector internal coordination data.	
Same description as for first-current-internal-data (Flight Plan Data Distribution Service).	
SectorExitData Detailed information on the sector downstream internal or external coordination data.	
Same description as for aor-exit-data (Flight Plan Data Distribution Service).	
TentativeData Detailed information on the tentative coordination data.	
Same description as for tentative-data (Flight Plan Data Distribution Service).	
SectorDialogueData Detailed information on the intra-sector coordination data (i.e. dialogue between executive and coordinator of the same OPS sector)	
~~distributed-sector-constraint~~	~~Indicates the dialogue type to negotiate on.~~
~~distributed-responsibility-state~~	~~Indicates the dialogue type status.~~
SectorNonCpdlcStandbyData Detailed information on STANDBY data (i.e. when an aircraft is requested to standby on the frequency)	
~~entry-noncpdlc-standby-data~~	~~Possible values:~~ • ~~Transfer-standby-acceptable: means that the receiving controller has indicated that the transferring controller can transfer the flight in standby frequency.~~

	• ~~Transferred-in-standby is an additional boolean set when the transferring controller has indicated that the XASM was done in standby mode.~~ • ~~Standby-alert is set by the server if no Assume is made by the receiving controller x seconds after the transferring controller has made the XASM in standby mode.~~
~~exit-noncpdlc-standby-data~~	~~Same as for entry-noncpdlc-standby data but then for exit.~~
FlightLeg state information	
Same description as for FlightLegStateinformation (Flight Plan Data Distribution Service).	

4.4.3 FlightPlanMonitoringMessage

The following section describes the data items provided in the FpmMessage.

header	See section 4.2.1
MonitoringList List containing the flight plan monitoring information for up to 150 SFPLs.	
MonitoringList.MonitoringInfo	
message-type	Create, update or cancel message.
plan-identifier	Unique plan identifier (used to refer to the flight plan in all other messages). Assigned by the server.
fp-auto-excluded	Set to TRUE in case the flight plan is automatically exluced for flight plan monitoring by the server.
~~fp-manual-excluder~~	~~Populated with the OPS sector name of the position that has manually excluded the flight for flight plan monitoring. Otherwise it is filled with the NULL value.~~ ~~Manual FPM exclusion is currently not supported.~~
monitoring-alerts	List of monitoring alerts existing for the flight plan. TRUE means the alert is triggered. FALSE means no alert. Multiple alerts may be set simultaneously. Following alerts are possible: • lateral: in case of lateral deviation of the track from the expected route. • turning-too-far: in case the track is expected to turn too far in a turn. • turned-too-far: in case the track had turned too far (i.e. flying not the correct heading after an expected turn). • heading: in case of un-conformance with the manual constraint set by the operator (see section 3.5.10 Operation manageHeading) • cfl: in case the track behaviour is considered incorrect in order to achieve the CFL/NFL. • level-bust: in case the track has overshot the applicable CFL/NFL • potential-level-bust: in case the track is expected to overshoot the applicable CFL/NFL • unexpected-holding-exit: in case the track is exiting the holding volume (currently not available in the demonstrator)

	• vertical-rate-of-climb: in case of un-conformance with the manual constraint set by the operator (see section 3.5.12 Operation manageCFL)
vertical-progress-status	Track vertical behavior, being: • unknown • levelled • climbing/descending
MonitoringList.MonitoringInfo.rerouting-proposal-info Information regarding a re-routing proposal by the FPM function	
status	Status of the rerouting: • none: no rerouting proposal • pending: rerouting proposal available, pending to be accepted (the acceptance of such proposal is currently out of scope of the demonstrator) • impossible: no rerouting proposal is possible
Intermediate-point	The coordinate position of the intermediate-point provided in the rerouting proposal.
rejoining-point	Identification for a trajectory point to join the intermediate point with the route. The identification is composed by a position point in coordinates and time indicating the correct point.
track-heading	Track heading at the moment of the rerouting proposal.

4.4.4 CorrelationMessage

The following section describes the data items provided in the CorrelationMessage.

header	See section 4.2.1
Correlation List List containing the correlation information for up to 150 correlated and uncorrelated flight plans.	
CorrelationList.CorrelationInfo	
Track number	Track identifier (-1 means the flightplan is not correlated)
~~sac~~	~~System Area Code (SAC)~~
~~sic~~	~~System Identification Code (SIC)~~
Linkage status	Enumerated indicating the linkage status of the flight plan: • uncorrelated: flight plan has not yet been correlated • correlated: flight plan correlated with flight plan • manually-decorrelated: flight plan was manually de-correlated; • correlation loss due to loss of track information; • correlation loss due to Mode-A mismatch; • correlation loss due to Mode-S mismatch.
ambiguity	Indicates ambiguities resulting from the correlation/de-correlation (e.g. mode A mismatch, track ambiguity, etc.)
CorrelationList.PositionInfo	
plan-number	Unique plan identifier.
plan-track-position	Provides the planned or estimated position of the flight. The Null position is provided in case of: • ASPL • Correlated SFPL

	• Uncorrelated SFPL with o flight-plan state “information” o flight-plan state “actual” until a design parameter before the entry in the AoR o flight-plan state “actual” as from exit AoR
~~plan-track-time~~	~~The time associated to the current plan-track-position. The Null time is provided in case the plan-track-position has the Null position value.~~
projected-position	The Projected Point for correlated flights under local control in lateral conformance.
~~last-track-position~~	~~Last known track position (in case of uncorrelated flight which was previously correlated).~~

4.4.5 FlightBrightMessage

The following section describes the data items provided in the FlightBrightMessage.

header	See section 4.2.1
message-type	Create, update or cancel message.
SSR Code Bright List List of SSR Codes which are subject to highlight within the OPS Sector	
ssr-code	SSR code
ModeS Bright List List of ModeS callsign which are subject to highlight within the OPS Sector	
callsign	ModeS callsign
FlightPlan Bright List List of flightplans for which the OPS sector has received a FPL bright (internal/external) and the list of flight plans for which the OPS sector has performed an external bright.	
FlightPlanBrightList.pointed-list	
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made.
last-initiator	Last initiator of the FPL bright function, being • OPS sector (if internal initiator) • External center identifier & flight sector (if external initiator)
FlightPlanBrightList.pointing-sent-list (only in case FPL bright is targeted to external entity)	
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made.
centre-id	Centre identifier to which the external flight plan bright (PNT) is sent.
flight-sector	External flight sector to which the flight plan bright (PNT) is sent.
protocol-state	See description for the FlightPlanDataMessage

4.4.6 SectorisationMessage

The following section describes the data items provided in the Sectorisation message.

header	See section 4.2.1
message-type	Create, update or cancel message.

application-timestamp	The application timestamp indicates when the new sectorisation was changed at the server.
sectorisation-pattern	The sectorisation pattern contains the sectorisation pattern (i.e. unique identifier for each sectorisation) currently applied for each sector group.
sectorisation-composition	The sector composition contains the list of airspace volumes and basic sectors composing each flight sector. Additionally, the primary and secondary voice frequency for each flight sector is included.
sectorisation-consolidation	The sector consolidation contains the list of flight sectors consolidated into each OPS sector.
sectors-allocation	The sector allocation contains the list of flight sectors allocated to each internal/external centre.

4.4.7 SsrCodeMessage

The following section describes the data items provided in the SsrCodeMessage.

header	See section 4.2.1
message-type	Create or update message. A “cancel” type will normally not happen for the SsrCode, as a reset of the SsrCode is performed by sending the NULL value for the SsrCode with message-type “update”.
code	The value of the sector code (case of reply to a request to reserve an SSR code) or the NULL value (case of reset for display of an SSR code).

4.4.8 WindForecastMessage

The following section describes the data items provided in the WindForecastMessage.

header	See section 4.2.1
message-type	Create, update or cancel message.
validity-period	The start and final validity times for the application of the forecast meteorological information.
wind-zones-setting	The identity names of, and the wind speed, wind heading, and air temperature, at each identified layer inside, all Meteorological (Wind) Zones. Up to 60 wind zones, and 40 wind layers can be distributed.

4.4.9 AirportMeteoMessage

The following section describes the data items provided in the AirportMeteoMessage.

header	See section 4.2.1
message-type	Create, update or cancel message.
aerodrome	The ICAO identifier of the related airport.
metar-text	The contents of the METAR message.
metar-flags	An indicator that is flagged whenever the received METAR information is outdated.
runway-data	Relevant runway information extracted from the METAR message (e.g. RVR for a certain runway).

meteo-info	The QNH, transition-level and transition-altitude

4.4.10 MapMessage

The following section describes the data items provided in the MapMessage.

header	See section 4.2.1
message-type	Create, update or cancel message.
Map Object Structure containing the map information.	
map-name	Name of the map.
map-reference	Unique identifier of the map.
activity-mode	Indicator representing the mode of the map, either activated manual or scheduled.
displayed	Boolean indicating if the map is to be displayed currently or not.
display-period	Corresponds to the start & end time of current (when displayed is true) OR next (when displayed is false) display period.
sector-group	List of sector group identifiers for which the map is to be displayed for.
~~flight-sectors~~	~~List of flight-sectors identifiers for which the map is to be displayed for. Only optionally provided.~~
map-definition	Defines the graphical elements constructing the map. Options include line type and color. Different types of contours are allowed, being circle, polygons, polylines and eight figures.
text-definition	List of textual elements associated to the map. Includes text, font size, color and display position.

4.4.11 SysytemModeMessage

The following section describes the data items provided in the SystemModeMessage.

header	See section 4.2.1
system	FDPS running as Primary or Fallback
system-mode	FDPS running in operational mode or test mode
system-submode	FDPS running in AUTHORISED (i.e. will accept inputs) or
link-status-on	TRUE when the Primary and Fallback system are connected (i.e. synchronized). FALSE otherwise.
system-degradation-level	Indicates the level of functionality provided by the FDPS, ranging from full-performance (no degradation) to degradation-level-4 (least functionality).
mtcd-is-on	TRUE (FALSE) if automatic MTCD is enabled (disabled).
mtcd-time-horizon	Look ahead time of the MTCD function.
fdps-version	Current applicable SW and adaptation data version running at the active FDPS kernel.
projection-data	Coordinate projection parameters (projection method, projection point & tangency point).

4.4.12 SpecialAreaMessage

The following section describes the data items provided in the SpecialAreaStatusMessage.

header	See section 4.2.1
message-type	Create, update or cancel message.
TSA Status List Structure containing the information related to maximum 350 special areas, all contained in a TSA Status Record	
TsaStatusList.TsaStatusRecord	
identity-name	The name of the special area
operation-mode	Operation mode of the current applicable booking: • Manual: the special area activation/de-activation is triggered by manual supervisor action. • Scheduled: the special area activation/de-activation is triggered automatically by following an activation/de-activation schedule.
instances-list	Per special arae, there may be one or more level bands booked, and even with different activation/de-activation times. This structure provides the means to contain such information. See below for more details on its content.
TsaStatusList.TsaStatusRecord.TsaInstancesList	
activity-status	Indicates if the TSA instance is currently active or inactive.
pre-warning-flag	Boolean value set 15 minutes prior activation or de-activation. It allows triggering specific display in case the ATCO would require being informed prior the activation or de-activation of a special area.
operation-block	Structure containing the following information regarding the TSA instance: • time-period: start and end time of the booking • lower-level: lower level of the booking (always provided in FL unit) • lower-source: identifies on which QNH information the lower-level is calculated, being: o FL: originally entered as Flight Level o AC: corrected with current QNH o AO: corrected with outdated QNH o AD: corrected with default (ISA) QNH • upper-level: upper level of the booking (always provided in FL unit) • upper-source: identifies on which QNH information the upper-level is calculated, being: o FL: originally entered as Flight Level o AC: corrected with current QNH o AO: corrected with outdated QNH o AD: corrected with default (ISA) QNH • entry-source: identifies the source of the booking, being: o ADAP: loaded from static adaptation data o FDO: entered from FDO (supervisor) o LARA: updated from LARA o AUP: updated from AUP message

	o UUP: updated from UUP message o B2B: updated from NM B2B services • lara-information: in case the entry-source is LARA, additional LARA information is stored here, being: o lactstat: LARA activation status o unit: unit making the booking o callsigns: list of involved callsigns in the mission o mission-type: type of mission o permeability: Indicates whether the area is permeable for other traffic or not o reservation-id: unique LARA reservation identifier

4.5 HeartbeatDistribution Messages

This section describes the messages involved for heartbeat.

ASN.1/XSD definition – See section 3.1.6 of this document.

4.5.1 HeartbeatMessage

4.5.1.1 Data items

Note: for simplicity reasons, it's currently assumed there is only one instance feeding the connector box (i.e. only one FDPS).

Note: The full message is distributed to the service consumer. The reply message sent back by the service consumer is limited to the header.

The following section describes the data items provided in the HeartbeatMessage.

header	See section 4.2.1
upstream-server-alive	Boolean indicating that the instance feeding the connector box with data is up-and-running. Optional field.
server-alive	Boolean indicating the connector box services are up-and-running. Optional field.

4.6 FlightPlanDataManagement messages

This section describes the Request messages. The Reply messages are described in section 4.2.2 AcknowledgementMessage.

4.6.1 RequestCreateModifyFPL message

The following section describes the data items provided in the RequestCreateModifyFPL message.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made. Contains additionally the indicator to reflect if the input refers to a what-if action or not.
action	Indicates the requested action, either: • create-sfpl (to request the creation of an SFPL). Additional parameters can be given (e.g. Callsign,

	ADEP, ADES, etc.). The following parameters are mandatory for SFPL creation: Callsign, Aircraft Type, ADEP, ADES, EOBT, TAS, RFL & Route. • create-aspl (to request the creation of an ASPL). Additional parameters can be given (as for the "create-sfpl" case). Callsign is mandatory. • modify (to request the modification of an ASPL/SFPL). Any parameter available in the asn1 definition may be requested for change. The same parameters as indicated in the "create-sfpl" case are mandatory. • upgrade-aspl (to request the upgrade of an ASPL into an SFPL). The same parameters as indicated in the "create-sfpl" case are required. • downgrade-sfpl (to request the downgrade of an SFPL into an ASPL).

4.6.2 RequestChangeAcStatusmessage

The following section describes the data items provided in the RequestChangeAcStatus message.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made.
number-of-aircraft	Number of aircraft in the flight
aircraft-type	ICAO identifier of the aircraft type
wake-turbulence-category	ICAO (RECAT) wake turbulence category identifier
non-deviation-status	Indicates that the controller responsible for the flight guarantees that the aircraft will stay on its assigned pattern and at its assigned level and will not deviate.
rvsm-status	Indicating RVSM status: exempt, equipped and/or capable
status-833	Indicating 833KHz status: equipped or exempt
~~uhf-equipped~~	~~Indicating whether flight is UHF equipped or not~~
~~brnav-equipped~~	~~Boolean indicating if aircraft is B-RNAV equipped or not.~~
~~prnav-equipped~~	~~Boolean indicating if aircraft is P-RNAV equipped or not.~~
~~modes-capability~~	~~Indicating ModeS capability: equipped, not or unknown.~~
people-on-board	The number of people on board.
corrupt-fssa	Indicating that the ModeS FSSA read-out is unreliable. Set to TRUE when unreliable.
is-avoiding-weather	Boolean indicating TRUE if a flight is circumnavigating due to weather.
is-fuel-dumping	Boolean indicating TRUE if a flight is dumping fuel (case of emergency).

4.6.3 RequestCangeCallsign message

The following section describes the data items provided in the RequestChangeCallsign message.

Header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made. Contains additionally the

	indicator to reflect if the input refers to a what-if action or not.
callsign	Indicates the new callsign.

4.6.4 RequestDeleteFPL message

The following section describes the data items provided in the RequestDeleteFPL message.

Header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made.
range-id	Number indicating the unique position of the controller position in the sector sequence.

4.6.5 RequestRetreiveFlight message

The following section describes the data items provided in the RequestRetrieveFlight message.

Header	See section 4.2.1
flight-id	Identification of the flight to be retrieved, can be either: • Internal flight plan number • IFPL ID • GUFI

4.6.6 RequestDCT message

The following section describes the data items provided in the RequestDCT (Direct To Position) message.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made. Contains additionally the indicator to reflect if the input refers to a what-if action or not.
range-id	Number indicating the unique position of the controller position in the sector sequence.
uplink	Boolean indicating if the DCT action is requested to be uplinked or not.
DctOption Any input is specified with one of the following coordination conditions: • exec: initiates dialogue as required • coord: execute input without initiating dialogue (regardless of changes to coordination) • consult: always initiates a dialogue	
DctOption.<exec | consult>.DctData	
to	Allows the sending of a single route point name (or coordinate) with optional intermediate point. In this case, the CB will automatically determine the most likely type of DCT instruction given (i.e. make the distinction between “to-original-route” and “to-trajectory” as given below). From an interface and client implementation perspective this option

	allows to perform DCT actions without the requirement for the client to reference to index in the trajectory.
to-original-route	Allows the sending of an intermediate-point (not on the trajectory/route) and an end-point on the original route.
to-trajectory	Allows the sending of an intermediate-point (not on the trajectory/route) and an end-point on the trajectory.
DctOption.<coord>.DctWithCfl	
DctData	See above.
cfl-data	Contains the actual cleared flight level. Optionally the CFL can be specified with a vertical rate of climb/descend and/or application time or distance.

4.6.7 RequestChangeRoute message

The following section describes the data items provided in the RequestChangeRoute message.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made. Contains additionally the indicator to reflect if the input refers to a what-if action or not.
range-id	Number indicating the unique position of the controller position in the sector sequence.
start-point	Start point of the re-routing. Needs to be located on a point of the trajectory. In case it's not provided, the current track position will be considered by the server as start-point.
intermediate-point	List of intermediate-points (located off-route) over which the flight is going to be re-routed.
end-point	Last point of the re-routing. Needs to be located on a point of the trajectory (i.e. this is the rejoining point)
uplink	Boolean indicating if the DCT action is requested to be uplinked or not.

4.6.8 RequestHeading message

The following section describes the data items provided in the RequestHeading (Heading) message.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made. Contains additionally the indicator to reflect if the input refers to a what-if action or not.
range-id	Number indicating the unique position of the controller position in the sector sequence.
uplink	Boolean indicating if the heading action is requested to be uplinked or not.
HeadingOption Any input is specified with one of the following coordination conditions: • exec: initiates dialogue as required • coord: execute input without initiating dialogue (regardless of changes to coordination)	

• consult: always initiates a dialogue	
HeadingOption.<exec | consult>.HeadingData	
present-heading	Present heading. Additionally requires type of closure (standard, indefinite or distance) as part of the request.
assigned-value	Assigned heading, relative or absolute true/magnetic heading. Additionally requires type of closure (standard, indefinite or distance) as part of the request.
non-specified-heading	Non-specified heading. Additionally requires type of closure (standard, indefinite or distance) as part of the request.
cb-manual	Heading due to weather avoidance.
HeadingOption.<coord>.HeadingWithCfl	
HeadingData	See above.
cfl-data	Contains the actual cleared flight level. Optionally the CFL can be specified with a vertical rate of climb/descend and/or application time or distance.

4.6.9 RequestXheading message

The following section describes the data items provided in the RequestXheading (cancel heading) message.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made. Contains additionally the indicator to reflect if the input refers to a what-if action or not.
range-id	Number indicating the unique position of the controller position in the sector sequence.

4.6.10 RequestCFL message

The following section describes the data items provided in the RequestCFL (cleared flight level) message.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made. Contains additionally the indicator to reflect if the input refers to a what-if action or not.
range-id	Number indicating the unique position of the controller position in the sector sequence.
cfl-data	Contains the actual cleared flight level. Optionally the CFL can be specified with a vertical rate of climb/descend and/or application time or distance.
application-limit	Optionally specified with a level until which the vertical rate of climb/descend restriction applies.
uplink	Boolean indicating if the CFL is requested to be uplinked or not.

4.6.11 RequestECL message

The following section describes the data items provided in the RequestECL (En-route cruising

level) message.

Header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made. Contains additionally the indicator to reflect if the input refers to a what-if action or not.
range-id	Number indicating the unique position of the controller position in the sector sequence.
ecl	En-route cruising level value.
application-limit	Optionally specified with a level from where the ECL should apply. The ECL is always propagated in the trajectory until exit AoI.

4.6.12 RequestRFL message

The following section describes the data items provided in the RequestRFL (Requested Flight Level) message.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made. Contains additionally the indicator to reflect if the input refers to a what-if action or not.
range-id	Number indicating the unique position of the controller position in the sector sequence.
rfl	Requested flight level value.

4.6.13 RequestPFL message

The following section describes the data items provided in the RequestPFL (Planned Flight Level) message.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made. Contains additionally the indicator to reflect if the input refers to a what-if action or not.
range-id	Number indicating the unique position of the controller position in the sector sequence.
pfl	Planned flight level value.

4.6.14 RequestSpeed message

The following section describes the data items provided in the RequestSpeed (speed instruction) message.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made. Contains additionally the indicator to reflect if the input refers to a what-if action or not.

range-id	Number indicating the unique position of the controller position in the sector sequence.
speed-restriction	Value and type of speed restriction. Options depend on type of restriction, being: • Present • Min • Max • Range • Assigned • Keyword • Resume
uplink	Boolean indicating if the speed request action is to be uplinked or not.

4.7 SectorSpecificDataManagement messages

This section describes the Request messages. The Reply messages are described in section 4.2.2 AcknowledgementMessage.

4.7.1 RequestControl message

The following section describes the data items provided in the RequestControl message

To process the input of a control input command, to allow either:

- Take control of the flight,
- Transfer control of the flight,
- Delegate control of the flight to the sector where the flight is geographically located,
- Hand-over propose/accept.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made.
range-id	Number indicating the unique position of the controller position in the sector sequence.
control-action	Indicates the action that is requested. Allows for the following actions: • Assume (i.e. take control of the flight by the next in sequence) • Force-assume (i.e. take control of the flight by any other controlling sector than the next one in sequence) • X-Assume (i.e. transfer the flight) • Collapse (i.e. give control to the sector where the flight is geographically located) • Re-contact (i.e. set new under-control frequency) • Hand-over-propose (i.e. HOP) • Hand-over-accept (i.e. HOA)

4.7.2 RequestNFL message

The following section describes the data items provided in the RequestNFL (Entry Flight Level) message.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made. Contains additionally the indicator to reflect if the input refers to a what-if action or not.
range-id	Number indicating the unique position of the controller position in the sector sequence.
NFL Data Any input is specified with one of the following coordination conditions: • exec: initiates dialogue as required • coord: execute input without initiating dialogue (regardless of changes to coordination) • consult: always initiates a dialogue • accept : accepts the proposed data in the open dialogue • reject : rejects the proposed data in the open dialogue	

• counter-propose: make a counterproposal to the proposed data in the open dialogue
NflData.<exec|coord|consult|counter-propose>.NflProposal

4.7.3 RequestTFL message

The following section describes the data items provided in the RequestTFL (Transfer Flight Level) message.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made. Contains additionally the indicator to reflect if the input refers to a what-if action or not.
range-id	Number indicating the unique position of the controller position in the sector sequence.
TFL Data Any input is specified with one of the following coordination conditions: • exec: initiates dialogue as required • coord: execute input without initiating dialogue (regardless of changes to coordination) • consult: always initiates a dialogue • accept : accepts the proposed data in the open dialogue • reject : rejects the proposed data in the open dialogue • counter-propose: make a counterproposal to the proposed data in the open dialogue	
TflData.< exec|coord|consult|counter-propose>.TflProposal	
tfl	The requested transfer flight level.
ecl-application	When set to TRUE, an ECL will be applied at the requested TFL level prior performing the actual change of TFL (default is FALSE).
tsfl	The requested supplementary transfer flight level.
pfl	A planned Flight Level can optionally be specified with a start point and/or a choice of end-point, time or distance.
coordination-partner	The external coordination partner to perform the transfer flight level change with. If not provided, the calculated one by the system will be used.
TflData.<accept>.TflProposal	
frequency	Frequency of the sector accepting the coordination proposal.
TflData.<reject>.TflProposal	
NULL	Not applicable

4.7.4 RequestChangeSectors message

The following section describes the data items provided in the RequestChangeSectors message

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made. Contains additionally the indicator to reflect if the input refers to a what-if action or not.
range-id	Number indicating the unique position of the controller position in the sector sequence.

action	Indicates the requested action, either: • skip: requested by an internal in-sequence sector to skip itself. • x-skip: requested by an internal in-sequence sector to unskip itself (provided skip action has been implemented before). • bypass: requested by current or in-sequence internal sector to bypass the next downstream internal sector. • x-bypass: requested by current or in-sequence internal sector to undo the previously requested bypass action. • delegate: requested by current or in-sequence internal sector to delegate the flight to another sector. In case the action is performed by o current sector, the flight's sector specific data will be distributed to the delegated sector, but the current sector will be maintained in the sector sequence until the delegated sector takes control of the flight. o in-sequence sector, that sector will be replaced by the delegated sector. • set-next: requested by current or in-sequence internal sector to change its next downstream internal sector into the desired one.

4.7.5 RequestEntryCoordination message

The following section describes the data items provided in the RequestEntryCoordination message.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made. Contains additionally the indicator to reflect if the input refers to a what-if action or not.
range-id	Number indicating the unique position of the controller position in the sector sequence.
Entry Data Any input is specified with one of the following coordination conditions: • exec: initiates dialogue as required • coord: execute input without initiating dialogue (regardless of changes to coordination) • consult: always initiates a dialogue • accept : accepts the proposed data in the open dialogue • reject : rejects the proposed data in the open dialogue • counter-propose: make a counterproposal to the proposed data in the open dialogue	
EntryData.< exec|coord|consult|counter-propose>.TflProposal	
nfl	The requested entry flight level. Not to be requested in case it concerns a departure flight out of internal aerodrome.

nsfl	The requested supplementary entry flight level. Not to be requested in case it concerns a departure flight out of internal aerodrome.
departure-level	Indicates the initial cleared level. Only applicable for departure flights from an internal airport.
accept-sector	Indicates the sector that delivered the departure clearance or accepted the coordination.
eto	The estimated time over the entry coordination point.
cop	The entry coordination point.
dct-point	The DCT-to point to which the flight may proceed.
pssr	The requested SSR code of the flight.
coordination-partner	The external coordination partner to perform the entry flight level change with. If not provided, the calculated one by the system will be used.
frequency	Frequency of the sector accepting the coordination proposal.
speed	The Speed on which the flight may proceed.
heading	The heading on which the flight may proceed.
EntryData.<accept>.EntryProposal	
frequency	Frequency of the sector accepting the coordination proposal.
EntryData.<reject>.EntryProposal	
NULL	Not applicable

4.7.6 RequestExitCoordination message

The following section describes the data items provided in the RequestExitCoordination message.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made. Contains additionally the indicator to reflect if the input refers to a what-if action or not.
range-id	Number indicating the unique position of the controller position in the sector sequence.
Exit Data Any input is specified with one of the following coordination conditions: • act-out : force early transmission of ACT or RAP (prior the automatic event) • pac-out : trigger sending of PAC message to downstream partner • rap-out : force early tranmission of RAP • exec: initiates dialogue as required • coord: execute input without initiating dialogue (regardless of changes to coordination) • consult: always initiates a dialogue • verbal-coord : when coordination has been done verbally with downstream partner (it also clears any indication of outstanding coordination) • xcoord : when downstream partner changes, allows to cancel previous coordination (and resets verbal indicator) • accept : accepts the proposed data in the open dialogue • reject : rejects the proposed data in the open dialogue • counter-propose: make a counterproposal to the proposed data in the open dialogue	
ExitData.< act-out|pac-out|rap-out>.TflProposal	

NULL	Not applicable
ExitData.< exec|coord|consult|counter-propose>.TflProposal	
nfl	The requested entry flight level.
nsfl	The requested supplementary entry flight level.
eto	The estimated time over the entry coordination point.
cop	The entry coordination point.
dct-point	The DCT-to point to which the flight may proceed.
pssr	The requested SSR code of the flight.
coordination-partner	The external coordination partner to perform the entry flight level change with. If not provided, the calculated one by the system will be used.
frequency	Frequency of the sector accepting the coordination proposal.
speed	The Speed on which the flight may proceed.
heading	The heading on which the flight may proceed.
ExitData.< verbal-coord>.TflProposal	
NULL	Not applicable
ExitData.< xcoord>.TflProposal	
NULL	Not applicable
ExitData.<accept>.EntryProposal	
frequency	Frequency of the sector transferring the flight (and accepting the coordination proposal in case it was initiated by the downstream).
ExitData.<reject>.EntryProposal	
NULL	Not applicable

4.7.7 RequestDepartureClearance message

The following section describes the data items provided in the RequestDepartureClearance message.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made. Contains additionally the indicator to reflect if the input refers to a what-if action or not.
range-id	Number indicating the unique position of the controller position in the sector sequence.
Entry Departure Data	
departure-level	Indicates the initial cleared level.
accept-sector	Indicates the sector that delivered the departure clearance.
eto	The take-off time.
pssr	The requested SSR code of the flight.
frequency	Frequency of the sector accepting the coordination proposal.
~~sid~~	~~The SID for the departure.~~
~~departure-runway~~	~~The departure runway.~~

4.7.8 RequestChangeFreq message

The following section describes the data items provided in the RequestChangeFreq message.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made.
range-id	Number indicating the unique position of the controller position in the sector sequence.
action	Frequency action, either: • reset-exit: erases the exit sector frequency of requesting OPS sector and entry frequency of the next OPS sector. • set-exit: sets the exit frequency of the requesting OPS sector and entry frequency of the next OPS sector. • set-entry: sets the entry frequency of the requesting OPS sector and exit frequency of the previous OPS sector.

4.8 CorrelationManagement messages

This section describes the Request messages. The Reply messages are described in section 4.2.2 AcknowledgementMessage.

4.8.1 RequestLink message

The following section describes the data items provided in the RequestLink message.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made.
link-action	Link or unlink specified flight plan with specified track.

4.8.2 RequestSetSsr message

The following section describes the data items provided in the RequestSetSsr message.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made.
ssr-data	SSR code to be set, either present/next or downstream.

4.9 FlightBrightManagement messages

This section describes the Request messages. The Reply messages are described in section 4.2.2 AcknowledgementMessage.

4.9.1 RequestFlightBright message

The following section describes the data items provided in the RequestFlightBright message.

header	See section 4.2.1
plan-identifier	Unique plan-number, flight-identifier or GUFI of the flight plan for which the input is made. Mandatory in case the action is flight plan bright.
action	Indicates whether the request concerns an SSR code bright, ModeS bright or flight plan bright.
BrightAction.Ssr	
ssr-bright	Contains the SSR code of the track that requires high-light.
cancel-all	Cancels all highlighted tracks (which have been previously requested for SSR bright by this OPS sector).
cancel-code	Contains the SSR code of the track that requires removal of highlight.
BrightAction.ModeS	
modes-bright	Contains the ModeS callsign of the track that requires highlight.
cancel-all	Cancels all highlighted tracks (which have been previously requested for ModeS bright by this OPS sector).
cancel-modes	Contains the ModeS callsign code of the track that requires removal of highlight.
BrightAction.Fpl	
fpl-bright	Contains the internal flight sector to point the flight to, or in case of external point the external center identifier and optionally the external flight sector.
cancel-fpl	This option is required when the flight plan bright is to be removed (only accepted when the OPS sector has actually received a flight plan bright).

4.10 SectorisationManagement message

This section describes the Request messages. The Reply messages are described in section 4.2.2 AcknowledgementMessage.

4.10.1 RequestChangeSectorisation message

The following section describes the data items provided in the RequestChangeSectorisation message.

header	See section 4.2.1
mode	Type of input requested, either: • verify: check the correctness of the input. It does not perform any sectorisation changes. • execute: request to implement the sectorisation change. • force: identical to "execute" but forces in case the FDPS/Server token is blocked.
sectors-combination	Contains the list of Basic Sector Sets composing each flight sector, requested for change of sectorisation.
sectors-consolidation	Contains the list of flight sectors consolidated into each OPS Sector, requested for change of sectorisation.

4.11 SsrCodeManagement message

This section describes the Request messages. The Reply messages are described in section 4.2.2 AcknowledgementMessage.

4.11.1 RequestGetSsr message

The following section describes the data items provided in the RequestGetSsr message.

header	See section 4.2.1
action	Indicates whether the request concerns a request to reserve an SSR code or to clear a requested one from display.
ReserveCodeAction.reserve-code	
primary	Request to reserve a code out of the primary SSR code pool.
secondary	Request to reserve a code out of the secondaryry SSR code pool.
ReserveCodeAction.clear-display	
Null	Request to erase the SsrCode previously requested from display (i.e. emit new SsrCode message to the requesting OPS sector containing the NULL value for the SSR code).

5 Appendix: technical details

The following sections provide more detailed information on the technical details of the OpenCWP service. They are intented to provide the developer with a more detailed understanding on the specifics of the interface.

5.1.1 TCP/IP connection

Typically, standard multicast services like UDP are not reliable by design; cyclic retransmissions have to be performed to protect from sporadic loss of messages. Especially, in case of flight data distribution, the burden of cyclic data transmissions placed on both the sender and the receiver are too high.
While certain Apache ActiveMQ implementations (e.g. ActiveBlaze, etc.) seem to support the reliable multicast protocol (RMP), the option has been chosen to only make use of ActiveMQ queues (and no topics on which messages are replicated) to establish "point-to-point" like connections[16]. This, together with the fact that the client is expected to establish a TCP/IP connection is for the time being considered to provide already sufficient data integrity and delivery guarantee (by retransmitting until the receiver acknowledges the reception of the packet which is standard on the TCP/IP stack).

5.1.2 Interface version

The connector box is both logically and physically located between a server (FDPS) and client. One of its main requirements is that it is able to translate certain information from one side into the applicable format at the other side, while using different protocols. This also implies that it is possible to asynchronously update a system instance interface at one side without degrading the behaviour of any other system instance interface as long as this update only impacts one or more of the following:

- New optional data fields are added,
- New distribution events are added,
- New optional service arguments are added,
- New services are added.

Obviously, the use of the new added items in the system instances that use it should not affect the agreed OpenCWP semantics & dynamics of the interface. Provided this is the case, this implies that it is possible for two (or more) systems with different interface version to exchange information at the same time, allowing for smooth transitions & asynchronous updates once in operation. In order for each system to be aware of the other's interface level, an interface version identifier is sent in each exchanged message (see section 4.2.1 for its definition).

5.1.3 Roles and Role Addressing

In section 5.1.1, it was explained that only usage is made of ActiveMQ queues. Through these point-to-point channels, a client will send request operations to the connector box, and acknowledgments of such requests are returned as well on this point-to-point channel. In order to allow full flexibility for future applications, the interface definition is not restricted to the point-to-point application. Rather the interface is designed such that for example, client unit designated data, can be sent to one unit using the point-to-point mechanism (i.e. ActiveMQ Queue) or in the future the multicast channel (i.e. ActiveMQ Topic) addressing one single unit by using the destination-function-id field in the header (see Table 4 in section 4.2.1 of this document).
The destination-function-id is the element constructing the "roles" concept. When addressing individual CWP units, sectors, or all CWPs, the connector box indicates the intended destination

[16] The MUAC operationally used reliable multicast protocol (RMP) is implemented between the MUAC FDPS and the Connector Box.

of the message using the destination-function-id. When receiving data the client units are expected to decode the header ids and process only those messages addressed to the “role” configured at the CWP unit. The “roles” concept also applies the other way around, where a client requests an input. In order for the server application to know “who” is requesting the input, the source-function-id is expected to be populated for each input.
Additionally, the header includes a source-id and destination-id. These mandatory items allow targeting of specific machines (e.g. “SERVER.2”). This is especially relevant for later phases, where multiple data servers will be installed. The keyword “all” can be used to indicate that all servers are targeted, for example “SERVER.all”. See Table 4 in section 4.2.1 of this document for more details.

5.1.4 Flight Sectors, OPS sectors, Basic Sectors and Control Conditions

In order to understand further concepts in this document, the concept of Flight Sectors (FS), Operational Sector (OPS) and Control Condition is required. While the MUAC FDPS has many more concepts relating to sectorisation, this section details only those relevant aspects in order to understand the (re-)sectorisation concept.

A Flight Sector (FS) is considered as the combination of one or more volumes of airspace and one or more control conditions. A flight sector can have up to two frequencies associated. Flight Sectors are predefined in the FDPS adaptation database and their names cannot be changed dynamically. The assignment of airspace volumes to Flight Sectors (i.e. sectorisation) can be changed dynamically.

The traffic within an airspace volume can be distributed among different flight sectors working on the same piece of airspace (but interested in different types of traffic). The type of traffic assigned to a flight sector is performed by making use of one or more Control Conditions (CC). A Basic Sector (BS) is the combination of an airspace volume and a type of traffic (i.e. control condition) to be assigned.

An Operational Sector (OPS) is a name statically defined in the FDPS adaptation database corresponding to either a Controller or a Planner Team. One or more Flight Sectors may be assigned to an OPS Sector (and can be dynamically changed). An OPS Sector is responsible for the traffic of all Flight Sectors assigned to it.

Per OPS Sector, several functions can be off-line defined and stored in static adaptation data as sector-role pairs. For the ADaaS project, the following roles are considered:

- SEC: Executive Controller
- SEA: Assistant Controller
- SPC: Co-ordinating Controller

The following table shows an example of Flight Sectors and Operational Sectors definition.

Basic Sector		Flight Sector	OPS Sector & Function (day)		OPS Sector & Function (night)	
1	AV1 + Condition(s)	DLSK_LOW	LOW	SEC & SPC	LOW	SEC & SPC
2	AV1 + Condition(s)	DLSK_LOW	LOW		LOW	
3	AV2 + Condition(s)	LJ_APP	CLA	SEC & SPC	LOW	
4	AV2 + Condition(s)	LJ_APP	CLA		LOW	
5	AV2 + Condition(s)	LJ_APP	CLA		LOW	

Table 5: Example of FS, OPS & BS definition

The example shows how several airspace volumes are assigned to Flight Sectors, which are then combined into OPS Sectors. In our example flight sector “DLSK_LOW” (Dolsko Lower) consists out of 2 basic sectors (i.e. one airspace volume and two control conditions), while flight sector “LJ_APP” (Ljubljana Approach) consists out of 3 basic sectors (i.e. one airspace volumes and three control conditions). During normal operations, those flight sectors are allocated to OPS sectors LOW (1 executive & 1 planner) and CLA respectively (1 executive & 1 planner), while during night operations the “LJ_APP” flight sectors are collapsed onto the LOW Operational Sector (1 executive & 1 planner).

5.1.5 Roles allocation

In the concept of “roles”, as explained in section 5.1.3, it is expected that the client units provide themselves with means to assign roles to each unit. After each re-assignment of roles at the client unit, a new initialisation request (indicating the source-id, for example “CLIENT.1”, and source-function-id, for example “LOW.SEC.all.all” as operational executive sector) is expected. In the ADaaS WP2 demonstrator, it is assumed that each CWP position is composed of a unique and single role.

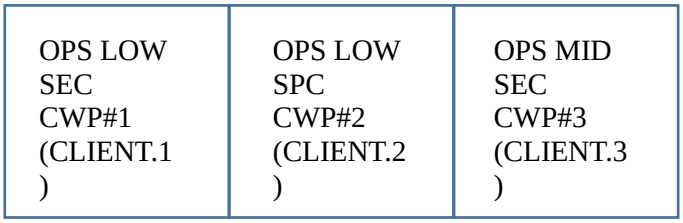


Figure 23: Example of a role allocation use case

Figure 23 presents a graphical representation of a typical use case where initially the same piece of airspace is controlled by operational sector LOW, more specifically the executive and coordinating/planner controller from two different but adjacent physical CWP positions. CWP#3 is currently manned by the executive controller for operational sector MID. Because of an on-the job training for an controller for OPS sector LOW, an additional executive assistant position is to be opened at physical position CWP#3.
CWP#1 and CWP#2 are not subject to a role re-allocation (i.e. the executive and planning controllers remain at the same position). CWP#3 is expected to make a role re-allocation: it is expected to de-assign its role (i.e. MID.SEC.all.all) and request initialisation to the connector box with source-id = “CLIENT.3” and source-function-id = “LOW.SEA.all.all” such that the server side can provide all required data for the additional assistant controller.

5.1.6 CWP role de-allocation & stop

In section 5.1.5 the concept of role-allocation was explained. This section describes the concept of role de-allocation and the process of notifying the connector box when the client’s application stops.
In case of de-allocation, the client can choose out of the following options:

a. A RequestLogout message is sent, followed by a new RequestLogin message to re-trigger initialisation.
b. Send a RequestLogin message (without a former RequestLogout message). Note that in between the time that the de-allocation starts and the RequestLogin is sent, sufficient heartbeat messages are still required to be sent from the client to the connector box in order to not let the latter time-out.

In case of stopping the client, the following methods are accepted by the connector box:

a. In case of a normal stop, the client can send a RequestLogout message after which the connector box will send a reply message and remove the client from its configuration and no more messages will be sent. The TCP/IP connection remains established. The CWP ProcessQueue opened by the client and InputQueue (CWP Server) remain applicable (i.e. they are not deleted). As non-durable queues are configured at the broker, it is ensured that upon new login the queues are providing only the relevant new data.
b. Even for the normal stop case, the client can choose to not send anything. As the connector box is constantly monitoring the response of the clients on heartbeats sent, it will remove the client from its configurations when not receiving a response in due time similar to the process as described in the previous point. This is also applied in case of an unplanned stop of the client; however in the latter case also the TCP/IP connection is lost.

5.1.7 Message reception by the service consumer (create, update and cancel types)

Within the interface definition, the clients are expected to follow certain general rules in order to correctly process the data received from the connector box:

1. The asn1 *FlightPlanIdentification* field is used to reference flight plan information.

2. The asn1 *TrackNumber* field is used to reference track information.

3. The asn1 *RangeId* field is used to reference sector specific information within a flight plan (see section 5.1.8 for more details)

4. In the context of internal reference storage (client side), the reception of messages with *MessageType* "create" or "update" is expected to result in the creation of a new internal reference at the client unit, provided there is no previous reference. In case there is already an internal reference then the creation shall not be triggered but an update process shall take place.

5. Some of the offered services are inter-dependent: the provision of information by one service is required prior to the usage of data offered by another service. Therefore, at initialisation, a clear sequence must be established by the server (connector box) towards the clients in order to allow the latter to properly process the messages/services in one-go (without requiring to "buffer" messages). More information on the ordering can be found section 2.10.1.

6. In the context of flight plan data the term "cancellation" has different meanings:
   a. Setting individual flight plan data items to the value "UNKNOWN" (see section 5.1.9),
   b. Removal of sector flight plan data (service SectorSpecificData) belonging to the same sector. This happens when the SectorSpecificDataMessage is sent with MessageType "cancel",
   c. Removal of complete flight plan data from the client. This happens when a FlightPlanDataMessage with message-type "cancel" is sent. In such case the client is expected to delete all related internal information for the flight.

All message fields with fixed length are sent by the connector box with right padding (left aligned) enabled.

5.1.8 Range identifiers

Range identifiers are unique identifiers that pertain to one overflight of the flight within the addressed operational sector. For each sector overflight the range identifiers increase with the number of sectors that have been overflown. This specific identifier is implemented for the case where a given sector is overflown more than once by the flight plan. Figure 24 presents an example of a sector sequence where a flight is crossing the same operational sector “A” twice. Whenever a client is performing an input (e.g. CWP#1) the server side cannot know only based on the identifier (e.g. CWP#1) or the role allocated (e.g. sector LOW.SEC.all.all) which instance is performing the input and therefore the concept of range identifiers is introduced.
In the interface definition, the usage of range identifiers by the client(s) has been made optional. In case they are not provided, the connector box assumes that the input is performed from the first instance of the operational sector requesting the input.

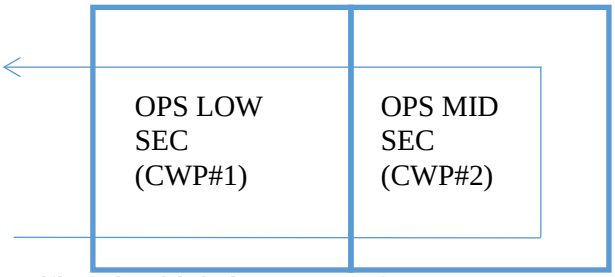


Figure 24: Range identifiers (multiple instances of one sector crossed)

5.1.9 Change, Reset and Unknown values (ASN1 definition)

The following sections contain general rules and restrictions for the use of the ASN1 syntax within the OpenCWP service.

No-Change and Reset Indications

Many message items sent from the connector box to the client (and other way around) are defined as OPTIONAL. Such categorisation is done in order to avoid unnecessary repetition of data to be sent: OPTIONAL items which are not included in a message are considered as not changed since the reception of the previous message. Therefore, absence of an optional message item is not to be considered as a reset of the data item. Resetting of individual message items is achieved by sending them with the “UNKNOWN” value, while multiple items may be reset at the same time by using the “RESET” indication on a CHOICE construct. Both methods are explained below.

The UNKNOWN Value

Special message item values have been defined to indicate absence of information to the user. These values can be used by the client when OPTIONAL items (in the ASN.1 sense) are not transmitted in the message.
The UNKNOWN value of a type can be defined explicitly in the ASN.1 syntax as follows:

```
TrackNumber          ::= INTEGER (-1..4095)
null-TrackNumber TrackNumber ::= -1
```

When not explicitly defined, the following UNKNOWN values apply:

- For IA5Strings, the “UNKNOWN” value is the smallest string of space (0x20) characters.
- For SEQUENCE OF items the “UNKNOWN” value is a list of size 0.

The RESET indication

In order to reset multiple items in one chunk, message items are reset to their “UNKNOWN” value by sending a CHOICE construct with a “delete“ item. An example definition can be found below:

```
Clearances ::= CHOICE
```

```
{
    delete NULL,
    items  SEQUENCE
    {
           level           [0] ClearedLevel         OPTIONAL,
           …
    }
}
```

5.1.10 Coordinate system

Coordinates are used within the MUAC FDPS and its input data sources (e.g. tracker). Obviously, it's vital that all systems using the OpenCwp service use the same coordinate system and projection method. In the OpenCwp service (and its corresponding message and operation definitions), an x/y-coordinate system has been used with dimension of 1/4096 nautical mile, origin and projection point at 460000N/0143000E. The projection method is a stereographic projection (WGS84).

5.1.11 Adaptation data

Using the OpenCWP's interface, both the client and connector box are expecting the agreed syntax and semantics described in the interface definition. However, for the FDPS to understand some requests (or for the client to understand some output), the actual payload needs to be aligned as well. Therefore, both client and server need to have, for some items, the same adaptation data level.

For example, a DCT request to a certain point unknown to the FDPS would not be accepted.

Following items are required to be aligned. For each item an example is given to demonstrate its necessity:

- OLDI capability (per partner)
  If a command to send a specific message for a partner (e.g. MAC message) is triggered which is not supported by the FDPS for that partner, the request will be rejected by the FDPS.
- Airports
  A request to create a flightplan to an unknown airport may be rejected.
- Airspace Volumes
  As part of the Sectorisation Distribution Service, the current operational configuration will be provided to the CWP. The exact location of each airspace component will not be distributed, and is considered to be locally available at the client side.
- Airways & CDRs
  A request to create a flightplan with an unknown airway may be rejected.
- Fixpoints & Navaids
  A request to create a flightplan with an unknown fixpoint may be rejected.
- Special Areas & Maps
  As part of the Special Area & Map Distribution Service, the list of "active" special areas and maps will be provided to the CWP. The exact location of each area will not be distributed, and is considered to be locally available at the client side.
- Strategic Constraints
  Strategic constraints, used to model the letter of agreements in the trajectory, are distributed towards the CWP. However, its location is not included, and is considered to be locally available at the client side.

- OPS Sector Names
  An input provided from a CWP client that represents an operational sector (see section 5.1.4) which is unknown at the FDPS will not be accepted.
- Flight Sector Names
  An input provided from a CWP client to delegate a flight to a certain flight sector which is unknown at the FDPS will not be accepted.
- SID/STAR information
  A request to create a flightplan with an unknown SID or STAR will be rejected.

The exact definition of each of the aforementioned items is outside the scope of this document (as it is considered as a dependency between the FDPS and clients, not between the connector box and its clients).

6 Abbreviations & Acronyms

Abbreviation	Term
AMQ	ActiveMQ
AMQP	ActiveMQ Protocol
ANSP	Air Navigation Service Provider
ATM	Air Traffic Management
ATSU	Air Traffic Service Unit
AoI	Area of Interest
AoR	Area of Responsibility
ASPL	Abbreviated System Plan
ATS	Air Traffic Services
BER	Basic Encoding Rules
BS	Basic Sector
CB	Connector Box
CC	Control Condition
CFL	Cleared Flight Level
COP	Coordination Point
CWP	Controller Working Position
DCT	Direct
ECL	En-route Cruising Level
EET	Estimated Elapsed Times
ETO	Estimated Time-Over
FDPS	Flight Data Processing System
FPM	Flight Plan Monitoring
FS	Flight Sector
GUFI	Globally Unique Flight Identifier
HDG	Heading

Abbreviation	Term
JMS	Java Messaging System
ICD	Interface Control Document
LoA	Letter of Agreement
MFX	Metering Fix
MTCD	Medium-Term Conflict Detection
MUAC	Maastricht Upper Area Control Centre
NFL	Entry Flight Level
NSFL	Supplementary Entry Flight Level
OPS	Operational Sector
PER	Packed Encoding Rules
PFL	Planned Flight Level
RFL	Requested Flight Level
RMP	Reliable Multicast Protocol
ROCD	Rate of Climb/Descend
SAC	System Area Code
SAS	Shared ATS System
SCL	Slovenia Control
SIC	System Identification Code
SFPL	System Flight Plan
SOA	Service-Oriented Architecture
STCA	Short-Term Conflict Alert
SWIM	System Wide Information Management
TFL	Transfer Flight Level
TSFL	Supplementary Transfer Flight Level
VRC	Vertical Rate of Climb

Abbreviation	Term
XER	XML Encoding Rules
XML	Extensible Markup Language

7 Annex A: SWIM Specifications Compliance Matrix

Identifier	Title	Level	Status	Ref / Comment
	General Requirements			
SWIM-SERV-001	Service description coverage	M	Ok	A single service is being described
SWIM-SERV-002	Service description language	M	Ok	It's in UK English
SWIM-SERV-003	Define abbreviations and acronyms	M	Ok	Present
SWIM-SERV-004	Use standard abbreviations and acronyms	R	Ok	Checked
SWIM-SERV-005	Service description identification	M	Ok	Present
SWIM-SERV-006	Service identification	M	Ok	Present
SWIM-SERV-007	Service abstract	M	Ok	Present
SWIM-SERV-008	Service provider	M	Ok	Empty: Provider-independent service definition
SWIM-SERV-009	Service categories	M	Ok	PCP info area + several ISRM facets
SWIM-SERV-010	Service standard reference	M	Ok	Empty: Not applicable for a service definition
SWIM-SERV-011	Operational needs	M	Ok	Present
SWIM-SERV-012	Service functionality	M	Ok	Functions & Real World Effects
SWIM-SERV-013	Service access and usage conditions	M	Ok	Example provided
SWIM-SERV-014	Quality of service	M	Ok	NFR instead of QoS for a service definition
SWIM-SERV-015	Technical constraint	R	Ok	Some provided
	Service Interface Requirements			
SWIM-SERV-016	Service interfaces	M	Ok	9 interfaces
SWIM-SERV-017	Message exchange pattern	M	Ok	P/S and R/R.
SWIM-SERV-018	Service profile and interface bindings	M	Ok	AMQP Messaging

SWIM-SERV-019	Service interface protocols	M	Ok	Present
SWIM-SERV-020	Machine-readable service interface definition	M Cond	Ok	XSD and ANS.1
SWIM-SERV-021	Service operations	M	Ok	Clearly described.
SWIM-SERV-022	Information Definition	M	Ok	Clear and detailed.
SWIM-SERV-023	AIRM conformance statement	M	Ok	Annex B: AIRM 4.1.0. Mapping Spreadsheet
SWIM-SERV-024	Filter capabilities	M	Ok	
SWIM-SERV-025	Service behaviour	M	Ok	Nominal behaviour plus error cases.
SWIM-SERV-026	Model view	R	-	Models only for illustrative diagrams
	Other Requirements			
SWIM-SERV-027	Service validation	M	Ok	Par. 3.1.9. Phase 3.
SWIM-SERV-028	Service monitoring	M Cond	Ok	Refers to HeartbeatDistribution interface
SWIM-SERV-029	Examples of Code	R	Ok	Step-by-step on how to connect example in Par. 3.1.11

8 Annex B: AIRM Mapping Spreadsheet

Spreadsheet embedded in Word Document.

AdaaS_AIRM_mapping.xlsx