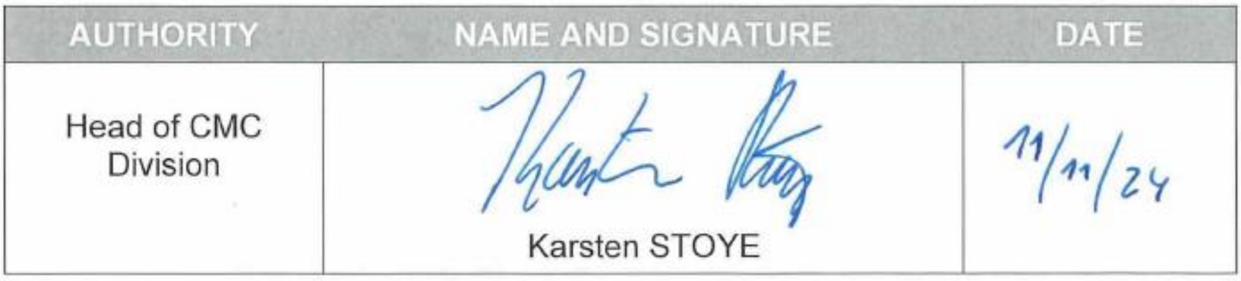
EUROCONTROL

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

Edition Number	:	1.0
Edition Date	:	11/11/2024
Status	:	Released issue
Intended for	:	LARA users

DOCUMENT APPROVAL

The following table identifies the authority who has approved the present issue of this document.



AUTHORITY	NAME AND SIGNATURE	DATE
Head of CMC Division	Karsten STOYE	11/11/24

DOCUMENT CHANGE RECORD

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

EDITION NUMBER	EDITION DATE	REASON FOR CHANGE	PAGES AFFECTED
0.1	31/01/2024	Initial Draft	All
0.2	15/04/2024	Update after internal revision	All
0.3	24/09/2024	Update after implementation findings	20, 25, 26, 27, 29, 34
1.0	11/11/2024	Released issue	

CONTENTS

LIST OF FIGURES

EXECUTIVE SUMMARY

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

LARA Extension to this Specification introduces technical interface requirements enabling retrieval of national ASM plan and facilitating user-system interaction during the tactical phase of ASM.

The requirements are stemming from the LARA Users operational deployment needs.

1. Introduction

1.1 Background

This document is the LARA Extension to EUROCONTROL Specification for Airspace Management (ASM) Support System Requirements supporting the ASM processes at local and FAB level - Part II: ASM Systems Interface requirements (*ASM Specification - Part II*). It has been developed in collaboration with LARA users.

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

LARA Extension to this Specification introduces technical interface requirements enabling retrieval of national ASM plan and facilitating system-user interaction during the tactical phase of airspace management.

1.2 Purpose of the document

The purpose of this document is to introduce, additional to the ASM Service' ones, interfaces providing the planning and activation data as required.

These interface requirements present the LARA extension to the ASM Service. The implementation of this extension will facilitate the deployment of the ASM Service.

1.3 Scope of the document

The LARA Extension to the ASM Specification - Part II covers the technical requirements for a standardised interface between LARA and External Users, which includes ATC systems, in the retrieval of the National Airspace Use Plan and involvement in the tactical phase of airspace management processes.

The requirements are stemming from the LARA Users operational deployment needs.

1.4 Applicability

The LARA Extension to the ASM Specification - Part II shall be considered optional to the ASM Service depending on the operational relevance in the deployment of the Service.

1.5 Conventions

The conventions used in the ASM Specification - Part II are applicable to this document:

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

b. “**Should**” – indicates a recommendation or best practice.

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

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

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

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

where:

ASM stands for ASM Support System requirement;

[yy]: Is a sequence of 2 to 4 characters to identify the environment to which the requirement is referring to;

*Note: In this document this sequence of characters will be **INTF***

[Fn]: Is a sequence of 2 to 4 characters to identify the functionality of ASM Support System to which the requirement applies *(e.g. **TACT** – tactical activation)*;

[nnn]: Is a numeric identifier, for a sequence of requirements with the same **[Fn]** identifier[1].

1.6 Target Audience

The target audience for the LARA Extension to the ASM Specification - Part II includes, but is not limited to:

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

1.7 Abbreviations and Acronyms

All abbreviations and acronyms specified in the ASM Specification - Part II

1.8 Definitions

All definitions specified in the ASM Specification - Part II

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

1.9 Reference material

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

1.10 Document structure

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

  The requirements in this section have been defined taking a service approach. Information definition requirements are also included.

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

Annex B provides Web Services Description Language (WSDL).

2. Interface extension to the ASM Service

2.1 Airspace Planning Interface

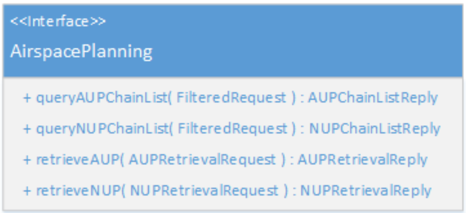

<<Interface>>
AirspacePlanning
+ queryAUPChainList(FilteredRequest) : AUPChainListReply
+ queryNUPChainList(FilteredRequest) : NUPChainListReply
+ retrieveAUP(AUPRetrievalRequest) : AUPRetrievalReply
+ retrieveNUP(NUPRetrievalRequest) : NUPRetrievalReply

Figure 1: Airspace planning interface operations

2.1.1 Interface role

The Airspace Planning interface enables the retrieval of Airspace Use Plans and/or National Airspace Use Plans held within LARA by External Users.

The data that can be managed by this interface is as follows:

- Airspace Use Plans (AUPs) - AUPs are a daily notification to interested parties of the Airspace Management status in a particular airspace management cell.

- Updated airspace Use Plans (UUPs) – UUPs are an update to the daily AUP that supersede the previous plan when published.

- National Use Plans (NUPs) – NUPs – similar to AUPs or UUPs except that the plans contain a different set (normally a superset) of airspace including that which is only of interest at the national level, and therefore has not been included in AUPs/UUPs. National Use Plans do not have a strict progression in the way AUPs are updated by UUPs. They are available through this interface via a ‘chain’ which provides access to the NUPs of a given date in order of creation but there is no equivalent of an ‘originating plan’ as there is with a UUP.

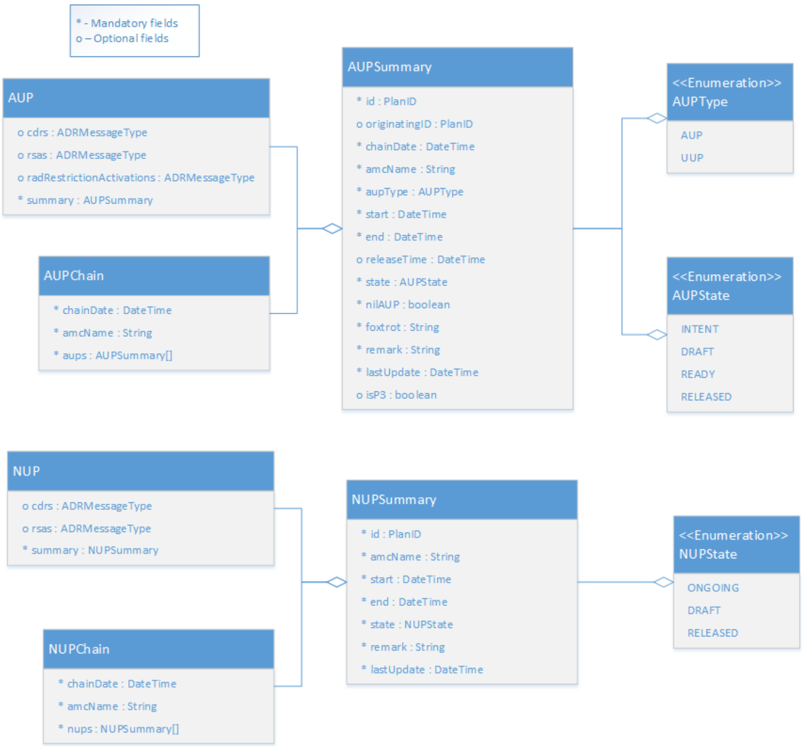


Figure 2: Airspace planning interface overview

The elements in the data model are described in full detail in section 2.4.2 Interface Messages.

2.1.2 Information Exchange Flow

The following diagrams present the information exchange flow for the Airspace Planning interface.

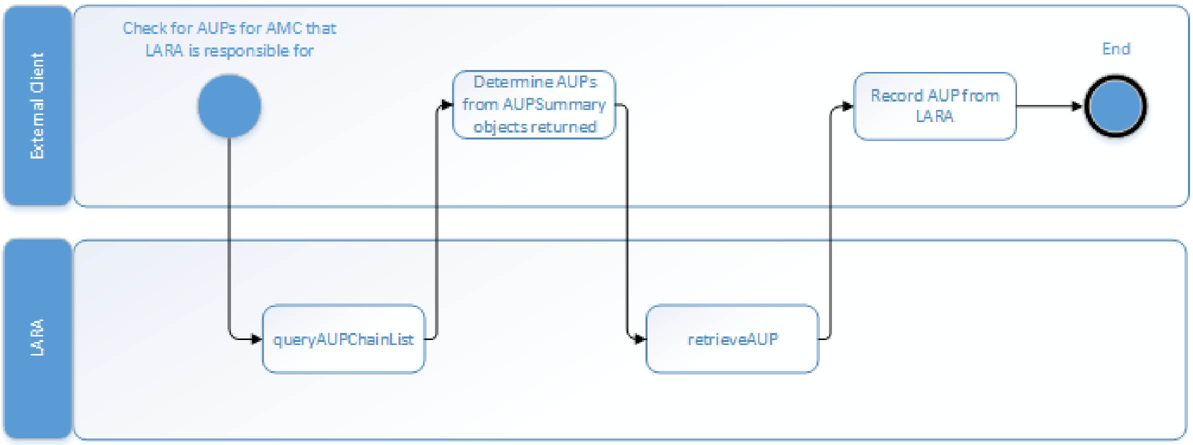

External Client
LARA
Check for AUPs for AMC that LARA is responsible for
queryAUPChainList
Determine AUPs from AUPSummary objects returned
retrieveAUP
Record AUP from LARA
End

Figure 3: Airspace planning interface - AUP information exchange flow

The diagram shows the exchange flow between LARA and an External Client, where the External Client is requesting AUP/UUPs from within LARA.

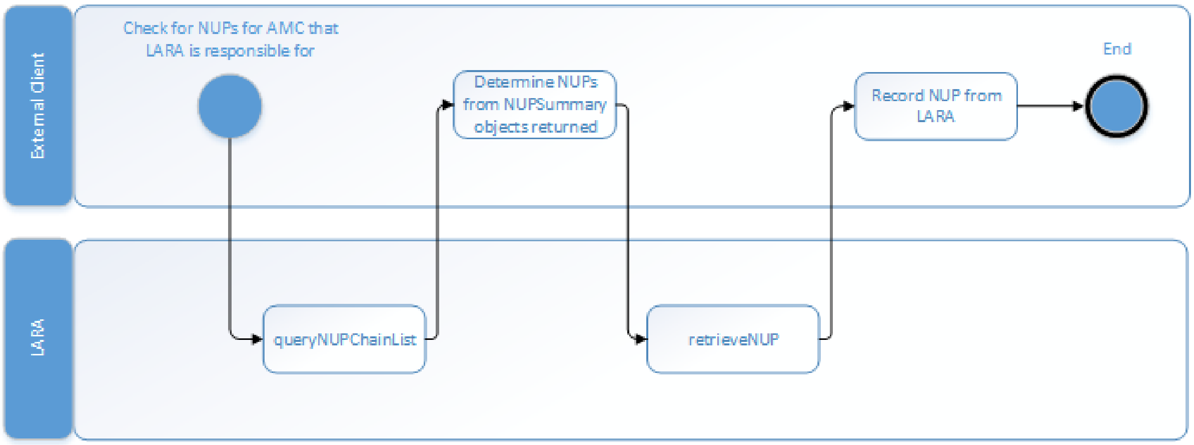

External Client
LARA
Check for NUPs for AMC that LARA is responsible for
Determine NUPs from NUPSummary objects returned
Record NUP from LARA
End
queryNUPChainList
retrieveNUP

Figure 4: Airspace planning interface - NUP information exchange flow

The diagram shows the exchange flow between LARA and an External Client, where the External Client is requesting NUPs (National Airspace Use Plans) from within LARA.

2.1.3 Interface Functions

The interface performs the following functions:

- AUP chain query - Retrieve the summaries and UUIDs of AUPs and UUPs based on filter criteria, for example, all the plans for a particular AMC and time period.
- AUP retrieval - Retrieve the full details of an AUP or UUP given the UUID of the AUP/UUP.
- NUP chain query - Retrieve the summaries and UUIDs of NUPs based on filter criteria, for example, all the NUPs for a particular AMC and time period.
- NUP retrieval - Retrieve the full details of an NUP given the UUID of the NUP.

ASM-INTF-PLAN-010: **ASM** Service **should** be supported by the Airspace Planning interface to manage access to local airspace planning data.

ASM-INTF-PLAN-020: The Airspace Planning interface **shall** support the following operations:

- queryAUPChainList,
- queryNUPChainList,
- retrieveAUP,
- retrieveNUP

2.1.4 Service Operations and Associated Messages

queryAUPChainList

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

ASM-INTF-PLAN-040: If the request is valid, the queryAUPChainList operation **shall** transmit the list of AUPSummary definitions that meet the supplied filter criteria in the AUPChainListReply message to the requesting user.

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

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

- AMCNameFilter
- ChangePeriodFilter
- InterestedIntervalFilter
- AndFilter

queryNUPChainList

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

ASM-INTF-PLAN-070: If the request is valid, the queryNUPChainList operation **shall** transmit the list of NUP Summaries that meet the supplied filter criteria in the NUPChainListReply message to the requesting user.

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

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

- AMCNameFilter
- ChangePeriodFilter
- InterestedIntervalFilter
- AndFilter

retrieveAUP

ASM-INTF-PLAN-090: The retrieveAUP operation **shall** receive and process the AUPRetrievalRequest message from an External User.

ASM-INTF-PLAN-100: If the request is valid, the retrieveAUP operation **shall** transmit the AUP/UUP matching the given AUPRetrievalRequest in an AUPRetrievalReply message to the requesting user.

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

retrieveNUP

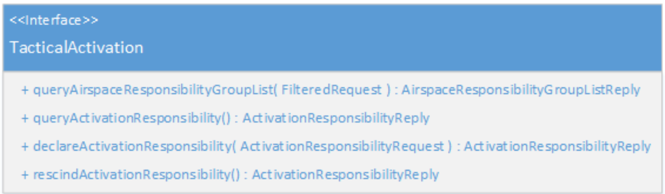
ASM-INTF-PLAN-120: The retrieveNUP operation **shall** receive and process the NUPRetrievalRequest message from an External User.

ASM-INTF-PLAN-130: If the request is valid, the retrieveNUP operation **shall** transmit the NUP matching the given NUPRetrievalRequest in the NUPRetrievalReply message response to the requesting user.

ASM-INTF-PLAN-140: If for any reason the request is not valid, the retrieveNUP operation **shall** transmit an appropriate error in the NUPRetrievalReply message to the requesting External User.

Note: The definition of these messages can be found in section 2.4.2 Interface Messages*.*

2.2 Tactical Activation Interface



```
<<Interface>>
TacticalActivation

+ queryAirspaceResponsibilityGroupList( FilteredRequest ) : AirspaceResponsibilityGroupListReply
+ queryActivationResponsibility() : ActivationResponsibilityReply
+ declareActivationResponsibility( ActivationResponsibilityRequest ) : ActivationResponsibilityReply
+ rescindActivationResponsibility() : ActivationResponsibilityReply
```

Figure 5: Tactical Activation interface operations

2.2.1 Interface role

Activation/deactivation of airspace structures in LARA may require manual actions from the responsible users. These actions are completed using specific functionalities of LARA i.e. users responsible for the activation/deactivation of the airspace structures shall use the HMI of LARA to perform their tasks. If these users are air traffic/defence controllers, the use of the LARA HMI adds complexity to their working environment.

The Tactical Activation interface provides an option for exchanging these activation/confirmation actions from the ATC position to LARA. This provides for the air traffic/defence controllers to use their ATC HMI to coordinate the activation/deactivation of the airspace structures with the process still managed by LARA.

The activation/deactivation process follows two phases, a confirmation stage followed by an acknowledgement phase. The user responsible for the airspace activation will confirm the activation or deactivation of an airspace before that status change goes ahead. Once confirmed other users that can be responsible for or may be responsible for other ongoing activities in the same airspace will acknowledge that they are aware of the change in status of the airspace. Once all confirmations and acknowledgements have been performed the airspace status change will proceed.

The users responsible for an activation/deactivation are typically well defined within LARA. An activation will be the responsibility of a given unit and so users acting for that unit will be responsible for the confirmation of the activation. All other units which may be responsible for the same airspace structure will be required to acknowledge the activation. Should a confirmation or acknowledgement not be dealt with in a timely manner, LARA will generate warnings which are distributed to all users involved in the process. This notifies the users of a potential delay in the process and allows for any external procedures to be initiated.

In addition to a simple, preconfigured LARA setup where the users responsible for the activation/deactivation are known ahead of time and constant throughout the process there is also an option for a ‘dynamic’ mode. In a dynamic activation mode users can declare and rescind responsibility for pre-defined groups of airspace at any time. When they declare responsibility, they will immediately be included in any ongoing activation/deactivation processes as well as any future processes, rescinding control will similarly remove them immediately from any ongoing processes potentially requiring other users to step in and perform any outstanding confirmation or potentially allowing an airspace to activate. This interface provides for access to the airspace groups and to allow the External Client to declare responsibility for a set of these groups.

LARA allows for significant flexibility in the airspace management process including the activation and deactivation of airspace. It is possible that after activation of a booking has started, the confirmation and some acknowledgements have been given, the originator of the booking could then edit the booking to, for example, change the booked levels. In this situation the received confirmations and acknowledgements become invalid and are discarded by LARA, LARA will then notify the users who had provided the confirmations and acknowledgements that the confirmation process has been reset via a warning.

This interface allows users of the ATC System to take part in the confirmation and acknowledgement processes. By declaring their own responsibility through the interface the ATC System can see in real time the status of the confirmations and acknowledgements that LARA is waiting for them to perform, before changing the status of the airspace. Users of the ATC System may then perform the required actions to change the status of airspace within LARA, by using the ATC System HMI. The users or system may rescind control to notify LARA that they will not be actively participating in confirmation and acknowledgement, but it should be noted that in some situations a user rescinding control will not remove the requirement from LARA that they perform the action to allow the airspace activation state change to proceed.

In order to provide this information in real time this interface is used to configure the responsibilities of the consumer with all confirmation, acknowledgement and warning notifications being sent to the consumer via an AMQP ‘read’ queue and requests performed by the consumer via an AMQP ‘write’ queue. Both of these queues are automatically set up the first time the user declares responsibility for airspace and are only deleted if the user rescinds control. The ‘read’ queue is treated similarly to a subscription queue created through the Subscription Management interface providing HeartbeatTechnicalMessages to the consumer, allowing them to be confident that LARA is active.

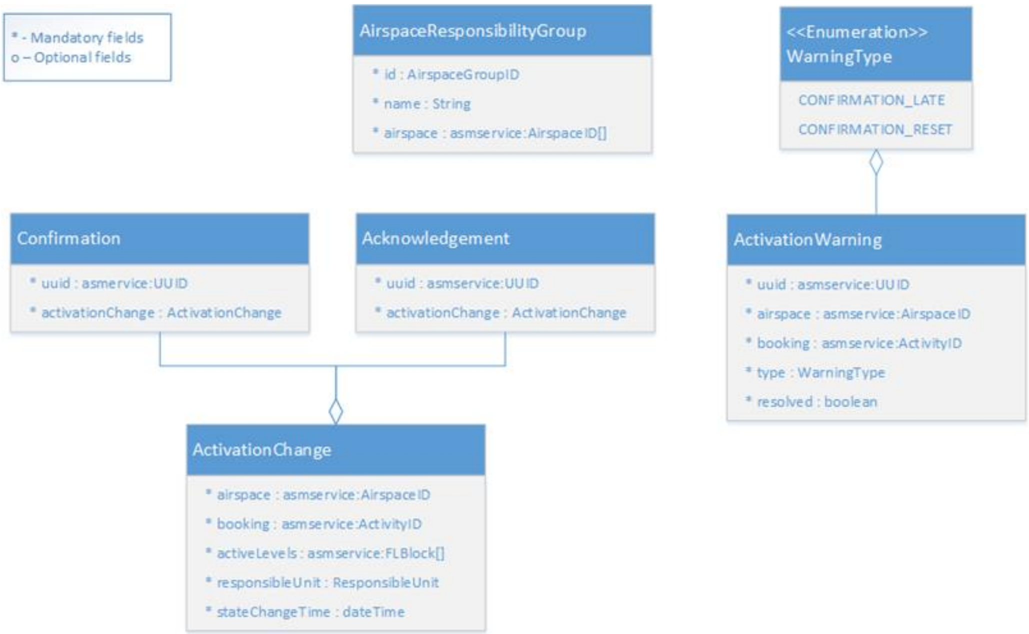

* - Mandatory fields
o – Optional fields
AirspaceResponsibilityGroup
* id : AirspaceGroupID
* name : String
* airspace : asmservice:AirspaceID[]
<<Enumeration>>
WarningType
CONFIRMATION_LATE
CONFIRMATION_RESET
Confirmation
* uuid : asmervice:UUID
* activationChange : ActivationChange
Acknowledgement
* uuid : asmservice:UUID
* activationChange : ActivationChange
ActivationWarning
* uuid : asmservice:UUID
* airspace : asmservice:AirspaceID
* booking : asmservice:ActivityID
* type : WarningType
* resolved : boolean
ActivationChange
* airspace : asmservice:AirspaceID
* booking : asmservice:ActivityID
* activeLevels : asmservice:FLBlock[]
* responsibleUnit : ResponsibleUnit
* stateChangeTime : dateTime

Figure 6: Tactical activation interface overview

The elements in the data model are described in full detail in section 2.4.2 Interface Messages.

Basic Use Case - An ATC System communicating with LARA

An ATC system is a service consumer of LARA. Assuming that the military controller is responsible for the activation of the airspace and the Civil ATCO shall confirm it. Both are working on their military and civil ATCO working positions.

Shortly before the scheduled activation of an airspace structure, LARA requires user confirmation. It sends an activation message to the military ATC/defence system as the military user is responsible for this activation.

> The Military controller receives the activation message on his/her working position, coordinates/decides on the activation, and confirms the activation by performing an action using the ATC/defence HMI (e.g. pressing a button, selecting an option, etc.). The exact nature of the action is down to the development of functionality on the ATC/defence HMI but performing the action should send a reply via the interface to confirm the activation.

LARA receives the confirmation and sends a message to the civil ATC system so that the responsible civil ATCO(s) acknowledges the activation.

> The civil ATCO is triggered with an acknowledgement message on his working position, does the required coordination/decides on the activation, and acknowledges the activation via the ATC HMI (e.g., pressing a button, selecting an option, etc.). The exact nature of the action is down to the development of functionality on the ATC HMI but performing the action should send a reply via the interface to acknowledge the activation.

Now that both civil and military controllers have approved the airspace activation, LARA activates the airspace structure at the scheduled time.

Airspace deactivation follows the same steps.

2.2.2 Information Exchange Flow

The following diagrams present the information exchange flow for the Tactical Activation interface. This interface makes use of a Request/Reply interface to define the parameters of the airspace responsibility the External User wants to establish and then makes use of a Publish/Subscribe interface to reliably inform the External User of new activations requiring attention. External Client interactions to confirm or acknowledge an airspace activation are also performed via Publish/Subscribe, notifying LARA of their confirmation or acknowledgement request.

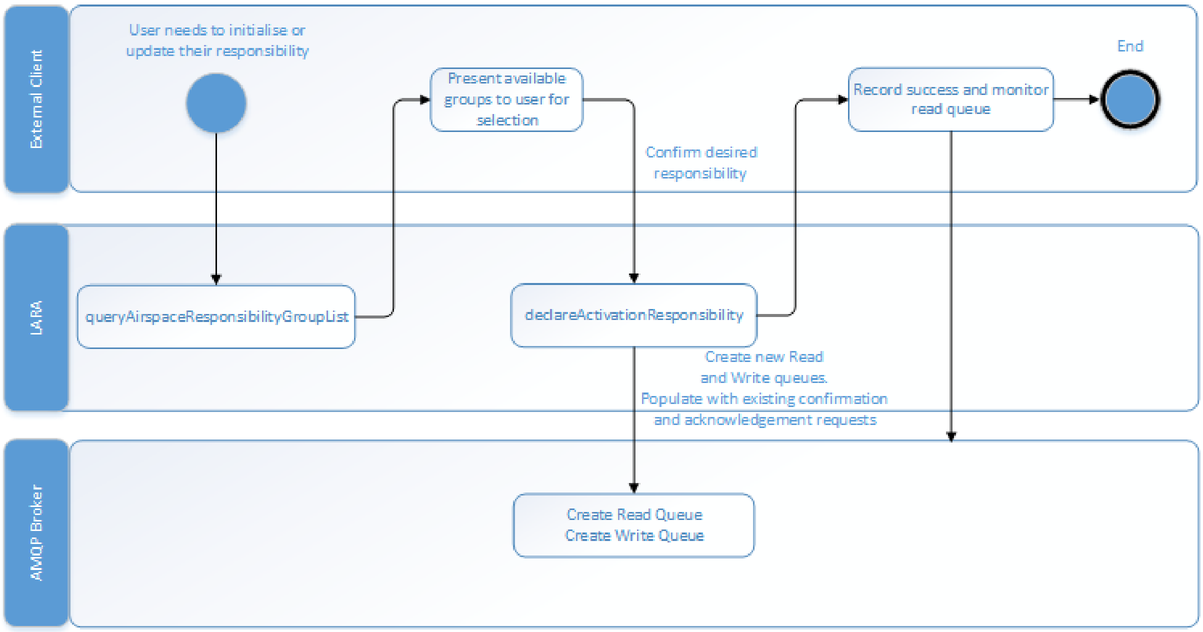

External Client
LARA
AMQP Broker
User needs to initialise or update their responsibility
queryAirspaceResponsibilityGroupList
Present available groups to user for selection
Confirm desired responsibility
declareActivationResponsibility
Create new Read and Write queues.
Populate with existing confirmation and acknowledgement requests
Create Read Queue
Create Write Queue
Record success and monitor read queue
End

Figure 7: Tactical activation interface information exchange initialization

This diagram shows the initial Request/Reply interaction being used to identify the available airspace groups, these groups are then presented for selection. Once the required groups, if any, are known then the External Client calls the interface to declare responsibility. In response LARA creates two new queues, one for the client to read from and one to write to, any existing confirmation or acknowledgement requests in the system that are applicable to this External Client are immediately added to the read queue in the form of a ConfirmationAcknowledgementSynchronizationNotification message. The names of the new queues are returned along with confirmation that the request was successful to the client. The External Client begins to monitor the read queue.

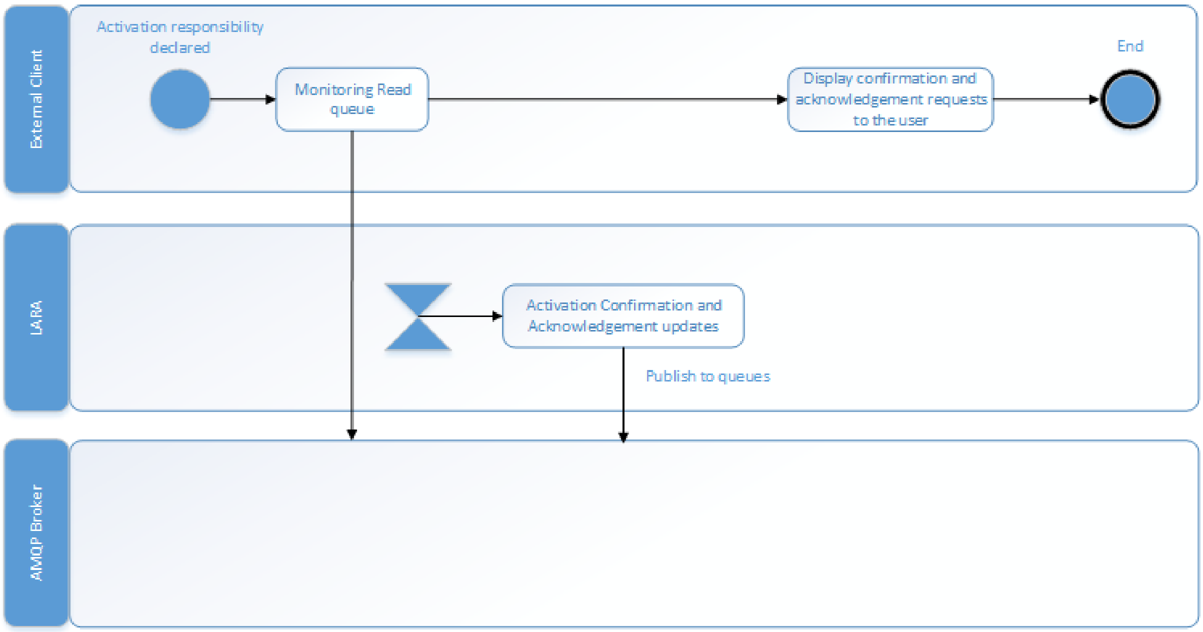

External Client
LARA
AMQP Broker
Activation responsibility declared
Monitoring Read queue
Display confirmation and acknowledgement requests to the user
End
Activation Confirmation and Acknowledgement updates
Publish to queues

Figure 8: Tactical activation interface information publication

This diagram shows an External Client which has declared responsibility for managing activations and is currently monitoring a read queue. Internally to LARA there is a change which generates new confirmation and acknowledgements, these are published to any relevant activation read queues including that of the External Client. The External Client reads the new confirmation and acknowledgement updates and presents the updates to the user.

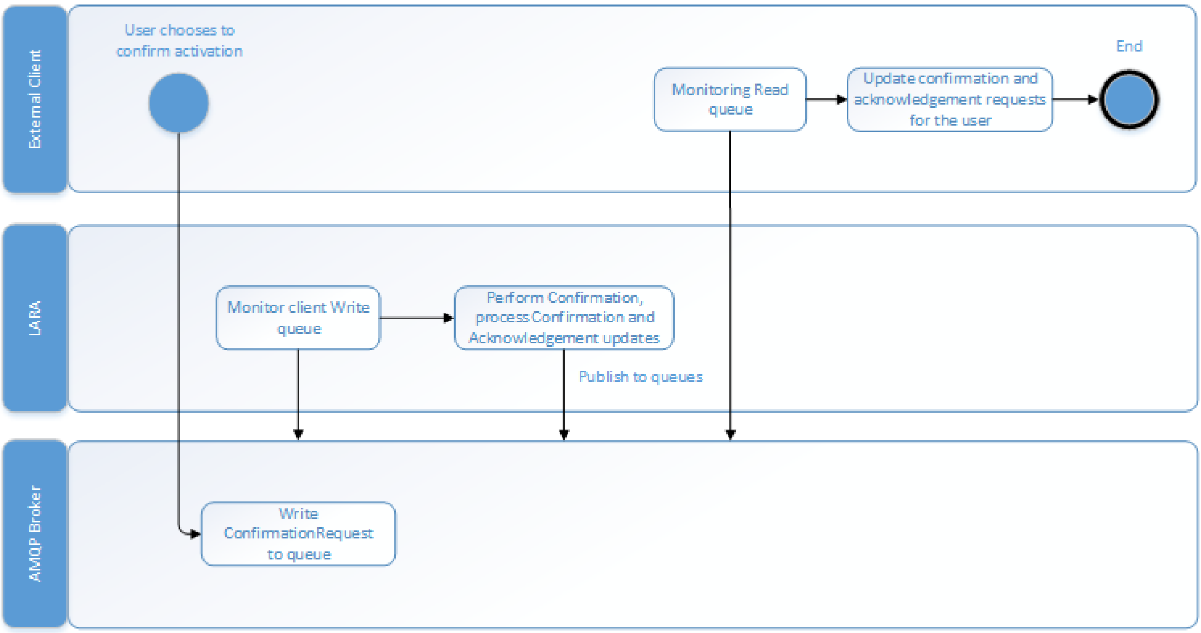

External Client
LARA
AMQP Broker
User chooses to confirm activation
Monitoring Read queue
Update confirmation and acknowledgement requests for the user
End
Monitor client Write queue
Perform Confirmation, process Confirmation and Acknowledgement updates
Publish to queues
Write ConfirmationRequest to queue

Figure 9: Tactical activation interface client confirmation

This diagram shows an External Client with a user who has previously been presented with a confirmation. The user chooses to confirm the activation. The External Client writes a ConfirmationRequest containing the relevant confirmation ID to its write queue. LARA, already monitoring the write queue, reads the message and processes the ConfirmationRequest, performing the confirmation and updating the confirmations and acknowledgements, removing the confirmation requirement from the External Client as that has been performed. The updates generated by LARA are published to any relevant read queues including that of the External Client. The External Client reads the ConfirmationNotification which includes the existing confirmation as removed and so removes the confirmation option from the user.

2.2.3 Queue Management

Client Disconnection

In general, the queue patterns defined in the ASM System Requirements Part 2 include the use of a Time-To-Live (TTL) function to detect unresponsive or disconnected consumers. In such a situation the consumer has their existing subscriptions paused and is effectively disconnected from the ASM system.

This general approach poses a potential problem with this interface. A consumer, who is required to provide the final acknowledgement of a dynamic activation, may become disconnected from the message queue. If this disconnection was reported back to LARA, this would effectively remove the responsibility of the consumer from the dynamic group. Subsequently, the pending airspace would no longer require any more acknowledgements and activate either immediately or at the start time of the booking.

To cater for such a scenario TTL will be applied differently on the tactical activation read queues. No TTL will be applied to tactical activation messages and consequently no message expiry, heartbeat messages sent to these queues will have the standard TTL set and so will expire but will not cause the consumer to be disconnected. As a result, no connection loss detection will be performed by LARA. Following an uncontrolled disconnection from the AMQP broker a consumer must either re-establish the connection and continue to take part in the confirmation and acknowledgement process or they must actively rescind their responsibility accepting that this may cause dynamically managed airspace to activate. As LARA is not detecting any disconnections on these queues it will continue to write heartbeat, acknowledgement and confirmation messages, these messages must be read and processed by the consumer when reconnecting to establish the correct state.

Should a consumer be disconnected for an extended period while holding responsibility for confirmation and acknowledgement the existing LARA super user functionality may be used to continue the activation and deactivation of airspace.

LARA Disconnection

There may equally be occasions when LARA may lose connection to the AMQP broker. In such a situation the consumer should recognise that HeartbeatTechnicalMessages are not being placed on the queue and consequently that there is an issue with LARA.

In this situation it is possible that LARA continues to function while being unable to notify external parties of updates to the required confirmations and acknowledgements. To ensure that consumers have the correct, up-to-date information, as soon as LARA reconnects to the AMQP broker it will place a new ConfirmationAcknowledgementSynchronizationNotification message on the queue which will define the complete data for the connected consumer.

LARA Errors

It is possible that LARA may fail in other ways to communicate a message to the consumer. In such a scenario a SubscriptionTechnicalMessage will be placed on the AMQP read queue indicating an error and no further messages will be placed on the queue. Should a SubscriptionTechnicalMessage be encountered on this queue the consumer must fully disconnect and re-establish their connection by re-declaring their responsibility.

2.2.4 Interface Functions

The interface performs the following functions:

- Requesting Airspace Responsibility Groups – Gives the consumer access to the airspace groups defined within LARA, allowing the consumer to declare a responsibility for a set of the groups.
- Requesting Activation Responsibility – Gives the consumer access to their current responsibility state, identifying whether they are generally responsible for confirmation and acknowledgement, and whether they are responsible for any airspace groups.
- Declaring Activation Responsibility – Makes the consumer responsible for general confirmation and acknowledgements, allows the consumer to define airspace groups they are responsible for.
- Rescind Activation Responsibility – Removes responsibility for any confirmation and acknowledgement, including for any groups that there was a responsibility for.

ASM-INTF-TACT-010: ASM Service **should** be supported by the Tactical Activation interface to manage tactical airspace structures activations.

ASM-INTF-TACT-020: The Tactical Activation interface **shall** support the following operations:

- queryAirspaceResponsibilityGroupList
- queryActivationResponsibility
- declareActivationResponsibility
- rescindActivationResponsibility

2.2.5 Service Operations and Associated Messages

queryAirspaceResponsibilityGroupList

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

ASM-INTF-TACT-040: If the request is valid, the queryAirspaceResponsibilityGroupList operation **shall** transmit the matching airspace responsibility group data in the AirspaceResponsibilityGroupListReply message to the requesting External User.

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

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

- AirspaceIDFilter
- AirspaceNameFilter
- AirspaceGroupIDFilter
- GeometryFilter
- AndFilter

queryActivationResponsibility

ASM-INTF-TACT-060: The queryActivationResponsibility operation **shall** transmit the responsibility data relevant to the calling user in the ActivationResponsibilityReply message to the requesting External User.

ASM-INTF-TACT-070: If for any reason the request is not successful, the queryActivationResponsibility operation **shall** transmit an appropriate error in the ActivationResponsibilityReply message to the requesting External User.

declareActivationResponsibility

ASM-INTF-TACT-080: The declareActivationResponsibility operation **shall** receive and process the ActivationResponsibilityRequest message from an External User.

ASM-INTF-TACT-090: If the activation responsibility declaration is valid, the declareActivationResponsibility operation **shall** transmit the ‘OK’ ReplyStatus response along with the names of the AMQP queues to be used and the updated responsibilities inside an ActivationResponsibilityReply message to the requesting External User.

ASM-INTF-TACT-091: If the activation responsibility declaration is valid, any existing responsibilities for the consumer **shall** be replaced with those from the ActivationResponsibilityRequest.

ASM-INTF-TACT-092: If updating the activation responsibility existing AMQP queues **shall** be reused for the requesting External User.

ASM-INTF-TACT-100: If for any reason the request is not valid, the declareActivationResponsibility operation **shall** transmit an appropriate error in the ActivationResponsibilityReply message to the requesting External User.

ASM-INTF-TACT-110: If the activation responsibility declaration is valid, all existing Confirmations and Acknowledgements and Warnings applicable to the requesting user **shall** be placed on the named AMQP queue as a Confirmation AcknowledgementSynchronizationNotification message.

ASM-INTF-TACT-120: Once activation responsibility is declared all subsequent changes to Confirmation and Acknowledgements applicable to the requesting user **shall** be placed on the named AMQP queue as ConfirmationNotification and AcknowledgementNotification messages.

ASM-INTF-TACT-140: The External User **shall** confirm an activation by publishing a ConfirmRequest message onto the AMQP write queue referenced in the ActivationResponsibilityRequest.

ASM-INTF-TACT-150: The External User **shall** acknowledge an activation by publishing a AcknowledgeRequest message onto the AMQP write queue referenced in the ActivationResponsibilityRequest.

ASM-INTF-TACT-160: Once a ConfirmRequest or an AcknowledgeRequest is successfully processed then a corresponding ConfirmationNotification or AcknowledgementNotification **shall** be published identifying the confirmation or acknowledgement as removed.

ASM-INTF-TACT-170: If any confirmation or acknowledgement, forming part of an activation or deactivation process which the External User is part of, is not completed in a timely manner then a warning of type CONFIRMATION_LATE **shall** be placed on the named AMQP queue as a WarningNotification.

ASM-INTF-TACT-171: Once all confirmations and acknowledgments for an activation or deactivation are complete, any previously sent warning **shall** be updated to resolved through a WarningNotification placed on the named AMQP queue.

ASM-INTF-TACT-180: If a previously actioned confirmation or acknowledgement is invalidated and so requires further action, a WarningNotification containing a warning of type CONFIRMATION_RESET **shall** be placed on the named AMQP queue.

rescindActivationResponsibility

ASM-INTF-TACT-200: The rescindActivationResponsibility operation **shall** remove all responsibilities from the requesting user, delete the associated AMQP queues and transmit the updated state in an ActivationResponsibilityReply to the requesting External User.

ASM-INTF-TACT-210: If for any reason the request is not valid, the rescindActivationResponsibility operation **shall** transmit an appropriate error in the ActivationResponsibilityReply message to the requesting External User.

Note: The definition of these messages can be found in section 2.4.2 Interface Messages*.*

2.3 Add-ons to the Subscription Management Interface

2.3.1 Interface role

The Subscription Management interface, defined in the ASM System Requirements Part 2, allows for data to be subscribed for to be pushed to External Users. LARA extends this interface to introduce new data types to be subscribed for along with corresponding filters.

All other service behaviour and requirements from the original specification are met by this extended interface.

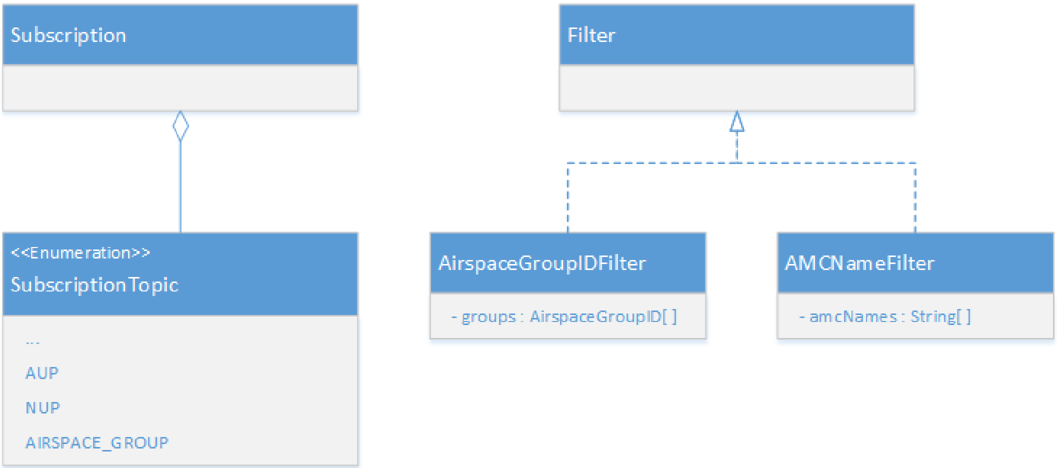


Figure 10: Extended subscription management interface overview

The new and modified elements in the data model are described in full detail in section 2.4.

2.3.2 Information Exchange Flow

The information exchange flow for the extended Subscription Management interface follows that defined by the ASM System Requirements Part 2.

Message timeouts, heartbeats and technical messages all follow the same flow and patterns.

2.3.3 Interface Functions

No new functions are introduced by this extension.

2.3.4 Service Operations and Associated Messages

No new service operations or messages are introduced by this extension.

2.4 Information Definition

2.4.1 Scope

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

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

2.4.2 Interface Messages

AcknowledgementRequest

Request to acknowledge that a change in the activation state of airspace should go ahead.

Attributes:

a. `asmservice:UUID` uuid (Mandatory)
The UUID of the Acknowledgement to be acknowledged.

AcknowledgementNotification

Message notified to a responsible External Client as the result of a change in their required acknowledgements.

Inherits from TacticalActivationNotification.

Attributes:

a. `Acknowledgement[]` acknowledgements (Mandatory)
New or modified acknowledgements. The array may be empty.
b. `Acknowledgement[]` removedAcknowledgements (Mandatory)
Acknowledgements which are no longer required, can no longer be acted on. The array may be empty.

ActivationResponsibilityReply

Inherits from Reply.

Reply to an ActivationResponsibilityRequest and the queryActivationResponsibility and rescindActivationResponsibility operations.

Attributes:

a. `boolean` isResponsible (Mandatory)
Indicates whether the requestor is currently considered actively responsible for confirmation/acknowledgement within LARA.
b. `AirspaceGroupID[]` responsibleGroups (Optional)
Array containing the IDs of any airspace responsibility groups the requestor is responsible for. Must be empty or unset if the request is not currently responsible for any confirmation/acknowledgements.
c. `String` readQueue (Optional)
The name of the AMQP queue the requestor must read from. Must be set if the isReponsible flag of this reply is true.
d. `String` writeQueue (Optional)

The name of the AMQP queue the request must write to. Must be set if the isResponsible flag of this reply is true.

ActivationResponsibilityRequest

Message to request general responsibility for airspace activation along with responsibility for the specified groups.

Attributes:

a. `AirspaceGroupID[]` groups (Optional)
   The IDs of the airspace responsibility groups the requestor should be made responsible for. No groups need to be specified for a general responsibility to be added.

ActivationWarningNotification

Message notified to a responsible External User as a result of a change or delay in an activation/deactivation process that they are participating in.

Inherits from TacticalActivationNotification.

Attributes:

a. `ActivationWarning` warning (Mandatory)
   The ActivationWarning itself.

AirspaceResponsibilityGroupListReply

Inherits from Reply.

Reply to a FilteredRequest.

Attributes:

a. `AirspaceResponsibilityGroup[]` groups (Mandatory)
   The retrieved airspace groups. The array can be empty.

AirspaceResponsibilityGroupNotification

Message notified to a responsible External User as a result of a change to the airspace responsibility groups available in LARA.

Attributes:

a. `AirspaceResponsibilityGroup[]` groups (Mandatory)
   New or modified airspace responsibility groups. The array can be empty.
b. `AirspaceResponsibilityGroup[]` deletedGroups (Mandatory)
   Deleted airspace responsibility groups. The array can be empty.

AUPChainListReply

Inherits from Reply.

Reply to a FilteredRequest.

Attributes:

a. `AUPChain[]` chains (Mandatory)
   The retrieved AUP chains. The array can be empty.

AUPNotification

Message notified to an AUP subscription as the result of a change to an AUP or the creation of a new AUP.

Attributes:

a. `AUP` aup (Mandatory)
   The changed AUP.
b. `boolean` deleted (Mandatory)
   If the AUP has been deleted. May only be true if the AUP is not released.

AUPRetrievalReply

Inherits from Reply.

Reply to an AUPRetrievalRequest.

Attributes:

a. `AUP` aup (Optional)
   The retrieved AUP. The AUP may be null if the request does not match an existing AUP.

AUPRetrievalRequest

Request for a specific AUP/UUP.

Attributes:

a. `PlanID` planID (Mandatory)
   The identifier of the plan to be retrieved.

ConfirmationAcknowledgementSynchronizationNotification

Message notified to a responsible External User defining all of their current confirmation and acknowledgements, replacing any previous messages.

Attributes:

a. `Confirmation[]` confirmations (Mandatory)
   All existing confirmations for the subscribed External User. The array may be empty.
b. `Acknowledgement[]` acknowledgements (Mandatory)
   All existing acknowledgements for the subscribed External User. The array may be empty.
c. `ActivationWarning[]` warnings (Mandatory)
   All existing activation warnings for the subscribed External User. The array may be empty.

ConfirmationNotification

Message notified to a responsible External User as a result of a change in their required Confirmations.

Inherits from TacticalActivationNotification.

Attributes:

a. `Confirmation[]` confirmations (Mandatory)
   New or modified confirmations. The array may be empty.
b. `Confirmation[]` removedConfirmations (Mandatory)
   Confirmations which are no longer required, can no longer be acted on. The array may be empty.

ConfirmationRequest

Request to confirm that a change in the activation state of airspace should go ahead.

Attributes:

a. `asmservice:UUID` uuid (Mandatory)
   The UUID of the Confirmation to be confirmed.

NUPChainListReply

Inherits from Reply.

Reply to a FilteredRequest.

Attributes:

a. `NUPChain[]` chains (Mandatory)
   Retrieved NUP chains.

NUPRetrievalReply

Inherits from Reply.

Reply to an NUPRetrievalRequest.

Attributes:

a. `NUP` nup (Optional)
   The retrieved NUP. The NUP may be null if the request does not match an existing NUP.

NUPRetrievalRequest

Request for a specific NUP.

Attributes:

a. `PlanID` nupID (Mandatory)
   The identifier of the NUP to be retrieved.

NUPNotification

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

Attributes:

a. `NUP` nup (Mandatory)
   The changed NUP.
b. `boolean` deleted (Mandatory)
   If the NUP has been deleted. May only be true if the NUP is in draft state.

<<abstract>> TacticalActivationNotification

Abstract ancestor of all tactical activation related notification messages.

Attributes:

a. `TacticalNotificationType` type (Mandatory)
   The type of this notification.

2.4.3 Complex Data Types

Acknowledgement

Represents a change in the activation state of an airspace which needs to be acknowledged.

Attributes:

a. `asmservice:UUID` uuid (Mandatory)
   The UUID identifying this acknowledgement.
b. `ActivationChange` activationChange (Mandatory)
   The details of the change in activation which need to be acknowledged.

ActivationChange

A new activation state for a specific airspace structure pending full confirmation and acknowledgement.

Attributes:

a. `asmservice:AirspaceID` airspace (Mandatory)
   The unique ID of the airspace this change is applicable to.
b. `asmservice:ActivityID` booking (Mandatory)
   The unique ID of the booking causing this activation change.
c. `asmservice:FLBlock[]` activeLevels (Mandatory)
   The flight level blocks which will be active as a result of the associated booking once this change has been confirmed and acknowledged.
d. `asmservice:ResponsibleUnit` responsibleUnit (Mandatory)
   The unit responsible for the booking and the associated active airspace.
e. `dateTime` stateChangeTime (Mandatory)
   The earliest expected time this state change will take effect.

ActivationWarning

An information warning related to the pending activation or deactivation of an airspace.

Attributes:

a. `asmservice:UUID` uuid (Mandatory)
   The unique ID of this warning.
b. `asmservice:AirspaceID` airspace (Mandatory)
   The unique ID of the airspace this warning is applicable to.
c. `asmservice:ActivityID` booking (Mandatory)
   The unique ID of the booking this warning is applicable to.
d. `WarningType` type (Mandatory)
   The type of Warning.
e. `boolean` resolved (Mandatory)

Whether this warning is resolved. An unresolved warning will be resolved in the future. Some warning types may be one-time notifications where resolved will be true when the warning is first received, indicating that there will be no further updates.

AirspaceResponsibilityGroup

A group of airspace for which activation and deactivation can be managed.

Attributes:

a. `AirspaceGroupID` id (Mandatory)
The unique ID of this airspace group.

b. `String` name (Mandatory)
The name of this airspace group.

c. `asmsevice:AirspaceID[]` airspace (Mandatory)
The unique IDs of the airspace making up this group.

AUP

An airspace use plan.

Attributes:

a. `ADRMessageType` cdrs (Optional)
The list of explicit CDR openings/closures for this AUP defined as a series of AIXM RouteSegmentTypes in an ADRMessage.

b. `ADRMessageType` rsas (Optional)
The list of explicit area allocations for this AUP defined as a series of AIXM AirspaceTypes in an ADRMessage.

c. `ADRMessageType` radRestrictionActivations (Optional)
The list of explicit RAD restriction activations for this AUP defined as a series of AIXM FlightRestrictionTypes in an ADRMessage.

d. `AUPSummary` summary (Mandatory)
The summary information relating to this AUP.

AUPChain

Represents an AUP chain, i.e. the AUP baseline of a (AMC, day) pair and its subsequent versions (UUPs) in the day.

Attributes:

a. `dateTime` chainDate (Mandatory)
Date and time of the AUP chain.

b. `String` amcName (Mandatory)
Name of the Airspace Management Cell.

c. `AUPSummary[]` aups (Mandatory)
The ordered list of AUP summaries in the chain.

AUPSummary

Represents a general summary of an AUP/UUP but without its main contents (i.e. CDR openings, closures or area allocations).

Attributes:

a. `PlanID` id (Mandatory)
   Unique ID of the AUP.
b. `PlanID` orignatingID (Optional)
   Unique ID of the plan that this plan is based off.
c. `dateTime` chainDate (Mandatory)
   Date and time of the AUP chain.
d. `String` amcName (Mandatory)
   Name of the Airspace Management Cell.
e. `AUPType` aupType (Mandatory)
   Whether AUP or UUP.
f. `dateTime` start (Mandatory)
   Start time of plan period.
g. `dateTime` end (Mandatory)
   End time of plan period.
h. `dateTime` releaseTime (Optional)
   When the AUP/UUP was released.
i. `AUPState` state (Mandatory)
   The current state of the AUP.
j. `boolean` nilAUP (Mandatory)
   Whether AUP/UUP is nil and user selected to not include any area allocations.
k. `String` foxtrot (Mandatory).
   Remarks field from foxtrot section of the plan.
l. `String` remark (Mandatory)
   Short remark.
m. `dateTime` lastUpdate (Mandatory)
   When the plan was last updated.
n. `boolean` isP3 (Optional)
   Identifies whether procedure 3 is active.

Confirmation

Represents a change in the activation state of an airspace which needs to be confirmed.

Attributes:

a. `asmservice:UUID` uuid (Mandatory)
   The UUID identifying this confirmation.
b. `ActivationChange` activationChange (Mandatory)
   The details of the change in activation which need to be confirmed.

NUP

A national airspace use plan.

Attributes:

a. `ADRMessageType` cdrs (Optional)
   The list of explicit CDR openings/closures for this AUP defined as a series of AIXM RouteSegmentTypes in an ADRMessage.
b. `ADRMessageType` rsas (Optional)

The list of explicit area allocations for this AUP defined as a series of AIXMAirspaceTypes in an ADRMessage.

c. `NUPSummary` summary (Mandatory)
   The summary information relating to this NUP.

NUPChain

Represents an NUP chain, i.e. the NUP baseline of a (AMC, day) pair.

Attributes:

a. `dateTime` chainDate (Mandatory)
   Date and time of the NUP chain.
b. `String` amcName (Mandatory)
   Name of the Airspace Management Cell
c. `NUPSummary[]` nups (Mandatory)
   Array of NUP summaries in the chain.

NUPSummary

Represents a general summary of an NUP but without its main contents (i.e. CDR opening closures or area allocations).

Attributes:

a. `PlanID` id (Mandatory)
   Unique ID of the NUP.
b. `String` amcName (Mandatory)
   Name of the Airspace Management Cell.
c. `dateTime` start (Mandatory)
   Start time of plan period.
d. `dateTime` end (Mandatory)
   End time of plan period.
e. `NUPState` state (Mandatory)
   The current state of this NUP.
f. `String` remark (Mandatory)
   Short remark.
g. `dateTime` lastUpdate (Mandatory)
   When the plan was last updated.

2.4.4 Simple Data Types

<UUID> AirspaceGroupID

Unique ID of an airspace responsibility group.

<<enumeration>> AUPState

Enumerates the possible states of an AUP.

Values:

a. `INTENT`
   Initial state when the plan has not yet been uploaded to NM.
b. `DRAFT`

After initial upload to NM with a successful validation.

c. `READY`
State of the plan once coordination with adjacent AMCs is completed the AUP is promoted to READY status by the AMC.

d. `RELEASED`
AUPs in the READY state are promoted to RELEASED by the NM CADF position.

<<enumeration>> AUPType

Enumerates type of AUP

Values:

a. `AUP`
Initial airspace plan.

b. `UUP`
Updated airspace plan.

<UUID> PlanID

Unique ID of an airspace plan (AUP, UUP, NUP), allocated by the service.

<<enumeration>> SubscriptionTopic

Enumerates the possible subscription topics, introducing LARA specific topics in addition to the standard topics defined by the ASM Service.

Values:

a. `STATIC_DATA`
Maps to the ASM Service *queryAirspace* operation.

b. `ACTIVITY_DATA`
Maps to the ASM Service *queryActivityDataList* operation.

c. `BOOKINGS`
Maps to the ASM Service *queryBookingList* operation.

d. `CONFLICTS`
Maps to the ASM Service *queryConflictList* operation.

e. `ACTIONS`
Maps to the ASM Service *queryActionList* operation.

f. `MISSIONS`
Maps to the ASM Service *queryMissionList* operation.

g. `PROPOSALS`
Maps to the ASM Service *queryBookingProposalList* operation.

h. `EVENTS`
Maps to the ASM Service *queryEventList* operation.

i. `ACTIVATIONS`
Maps to the ASM Service *queryActivationList* operation.

j. `AUP`
Maps to the *queryAUPChainList* operation, providing the changed AUP in the notification.

k. `NUP`

Maps to the *queryNUPChainList* operation, providing the changed NUP in the notification.

l. `AIRSPACE_GROUP`
Maps to the *queryAirspaceResponsibilityGroupList* operation.

<<enumeration>> TacticalNotificationType

Enumerates types of TacticalActivationNotificaiton.

Values:

a. `ACKNOWLEDGEMENT`
To identify notifications of acknowledgements.

b. `CONFIRMATION`
To identify notifications of confirmations.

c. `WARNING`
To identify notifications of warnings.

<<enumeration>> NUPState

Enumerates the possible states of an NUP.

Values:

a. `ONGOING`
Initial state when the NUP has not been drafted by AMC action.

b. `DRAFT`
After successful validation following the AMC drafting action.

c. `RELEASED`
Once released by AMC action.

<<enumeration>> WarningType

Enumerates the possible warning types.

Values:

a. `CONFIRMATION_LATE`
If one or more confirmations or acknowledgements are outstanding for an activation or deactivation to proceed.

b. `CONFIRMATION_RESET`
If a previously given confirmation or acknowledgement has been invalidated.

2.4.5 Filters

AirspaceGroupIDFilter

A filter defining a series of AirspaceGroupIDs to be retained.

Inherits from asmservice:Filter.

Values:

a. `AirspaceGroupID[]` groups (Mandatory)
The IDs of the airspace groups to be retained.

AMCNameFilter

A filter defining a series of AMC names to be retained.

Inherits from asmservice:Filter.

Values

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

2.5 Semantic correspondence of information definition

The Information Definition found in section 2.4 conforms with the semantics of the ATM Information Reference Model (AIRM) version 1.0.0.

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

ANNEX A – Semantic Correspondence Report

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

ANNEX B – Web Services Description Language (WSDL)

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