NM 23.0.0 - NOP/B2B Reference Manuals - Essentials

Edition No.	:	23.0.0.5.26	
Edition Issue Date	:	29/04/2019	
Author	:	NM NOP/B2B Team	
Reference	:	B2B/23.0.0/Essentials	
Copy No.	:		← stamp here

Document Control

Copyright Notice

Document Identification

Full Title:	NM 23.0.0 - NOP/B2B Reference Manuals - Essentials
Total Number of Pages:	80

Table of Contents

List of Figures

List of Tables

References

External

[1] *Web Service Contract Design and Versioning for SOA*. Thomas Erl. Prentice Hall. September 2008.

[2] AIXM: Aeronautical Information Exchange Model (http://www.aixm.aero)

[3] *A useful resource regarding PKI: http://www.oasis-pki.org*.

[4] *GlobalSign web site: https://www.globalsign.com/*.

[5] *Certificate Practices Statement (CPS) and the Certificate Policy can be consulted on https://www.globalsign.com/en/repository/*.

[6] On UUIDs: http://en.wikipedia.org/wiki/Universally_Unique_Identifier

Terminology

(1) This section provides definitions of terms that are not standard terms of the Web Services terminology. Definitions of these standard terms can be found in [1].

(2) Furthermore, this section clarifies a few terms that are used across this document:

a) Client application or client: an application that consumes/uses the NM NOP/B2B web services

b) Customer: an organisation that has an agreement with NM for the use of the NM NOP/B2B web services

c) User: this term is used in the context of authentication; it refers to the identity that is accessing the NM NOP/B2B web services; that identity is associated with the customer organisation

d) End user: it is a physical person who uses the client application

Main Abbreviations and Acronyms

AIXM (Aeronautical Information Exchange Model)
See [2].

UUID (Universally Unique Identifier)
See [6].

Chapter 1. Introduction

1.1. Identification

(1) This document forms part of the set of the NM 23.0.0 - NOP/B2B Reference Manuals, which all together form the NM 23.0.0 - NOP/B2B Documentation.

(2) Its reference is B2B/23.0.0/Essentials.

(3) Its title is NM 23.0.0 - NOP/B2B Reference Manuals - Essentials.

1.2. Purpose

(1) NM groups its B2B (web) services into logical (web) service groups, e.g. the "FlightServices" group contains `FlightPreparationService`, `FlightFilingService` and `FlightManagementService`.

(2) The granularity of the NOP/B2B Documentation is the service group and is called the "Reference Manual" for that service group. For example, in NM 23.0 where six service groups are defined ("CommonServices", "GeneralInformationServices", "AirspaceServices", "FlowServices", "FlightServices" and "PublishSubscribeServices"), six service group Reference Manuals are provided:

a) CommonServices NOP/B2B Reference Manual

b) GeneralInformationServices NOP/B2B Reference Manual

c) AirspaceServices NOP/B2B Reference Manual

d) FlowServices NOP/B2B Reference Manual

e) FlightServices NOP/B2B Reference Manual

f) PublishSubscribeServices NOP/B2B Reference Manual

(3) As the number of NM B2B Web Services increases, together with the number of Service Groups, it was deemed necessary to group in a separate document the general information not related to specific service groups. This need is anticipated via this document to provide all the information that is not specific to a specific service group, e.g. WS-* versions, WSDL and XML Schema organisation, authentication mechanism, and so forth.

1.3. Scope

(1) The intended readers of this document are customer-side governance specialists, business analysts, architects, designers and developers, involved in the development of client applications using NOP/B2B services.

1.4. Overview

(1) This version of the document focuses on the following topics:

a) Contextual Information

b) Protocols

c) Access to NOP/B2B Services

d) Web Services Lifecycle

e) Service Location

f) Request Prioritisation

g) Bandwith Protection

h) Exchange Model

i) Service Contract Organisation

j) Testing Facilities

k) Error and Warning Reporting

Chapter 2. Context

(1) NM provides two web-based channels for accessing its services:

a) The NOP/Portal (introduced in NM 12.2)

b) The NOP/B2B Services (introduced in NM 13.0)

(2) The NOP provides a consolidated view of the different aspects of the Network to support the planning process. The main goals of the NOP are the wide dissemination of Network information and collaboration between the Network partners.

(3) The NOP/B2B Services provide access to both data and services via system-to-system interface, allowing the NM customers to exploit and use the information in their own systems, according to their business needs.

(4) The more specific goal of the NOP/B2B Services within the global NOP context is to support the interoperability required in the future ATM systems, being the enabler for SWIM (System Wide Information Management) in SESAR.

(5) Supporting interoperability implies the use of standard technologies in the service provision. Therefore, as exposed in the next sections, the currently available standards (or *de factor* standards) are followed. NM will adapt to the evolution of these standards as they mature, and follow the standards further defined by SESAR.

Chapter 3. Protocols

3.1. General

(1) The NOP/B2B web services are based on JAX-WS 2.0 stack of operations.

(2) The NOP/B2B exposes its web services in two flavours:

a) XML requests and replies directly embedded into HTTP requests and responses (no SOAP envelope)

b) The same XML requests and replies embedded into SOAP messages, themselves embedded into HTTP requests and responses. Operation names are associated to SOAP requests.

(3) The SOAP NOP/B2B web services make use of WSDL 1.1 and SOAP 1.1.

(4) All NOP/B2B web services use HTTPS as transfer protocol.

(5) The details of the SOAP and non-SOAP messages are given in Service Contract Organisation & Conventions.

(6) All service group reference manuals come with a directory containing examples of SOAP and non-SOAP XML requests and responses. The provided examples are not exhaustive: not all requests of all services, nor all possible combinations of request attribute values are exemplified.

(7) SOAP and non-SOAP NOP web services are exposed at the same locations (same URLs).

3.2. Security Protocols

3.2.1. Data Encryption and Authentication

(1) The supported protocols are TLSv1, TLSv1.1, TLSv1.2.

3.2.2. Cryptographic Algorithms

(1) The supported cipher suites are in this order AES-256, AES-128, DES-168.

3.3. B2B Reply Compression

(1) If the client accepts compression from host, the server will compress the B2B replies.

(2) The server will not compress the B2B replies if the client does not accept compression from host.

(3) The processing time with or without compression is similar.

(4) The compression ratio is around 20%.

3.4. Protocol Errors

(1) See Error and Warning Reporting - Protocol Errors.

3.5. Access to Files

(1) Depending on authorisation, the NOP/B2B services give access to files via dedicated port types.

(2) Access to the file repository makes use of SOAP and non-SOAP protocols.

(3) The `AirspaceStructure` service provides access to files via the following service requests:

a) `CompleteAIXMDatasetRequest`

b) `IncrementalAIXMDatasetRequest`

(4) The above service requests `queryCompleteAIXMDatasets()` and `queryIncrementalAIXMDatasets()` do not return the content of each file, but only the file identifiers.

(5) `queryCompleteAIXMDatasets()` - REQUEST

```
<?xml version="1.0" encoding="UTF-8"?>
<airspace:CompleteAIXMDatasetRequest xmlns:airspace="eurocontrol/cfmu/b2b/AirspaceServices">
  <endUserId>tstcadf1</endUserId>
  <sendTime>2019-03-17 10:00:00</sendTime>
  <queryCriteria>
    <date>2019-03-17</date>
  </queryCriteria>
</airspace:CompleteAIXMDatasetRequest>
```

(6) `queryCompleteAIXMDatasets()` - REPLY

```
<?xml version="1.0" encoding="utf-8"?>
<as:CompleteAIXMDatasetReply xmlns:as="eurocontrol/cfmu/b2b/AirspaceServices"
                             xmlns:cm="eurocontrol/cfmu/b2b/CommonServices">
  <requestReceptionTime>2019-03-17 10:44:52</requestReceptionTime>
  <requestId>B2B_CUR:12787</requestId>
  <sendTime>2019-03-17 10:44:52</sendTime>
  <status>OK</status>
  <data>
    <datasetSummaries>
      <updateId>721430</updateId>
      <publicationDate>2019-03-17</publicationDate>
      <sourceAIRACs>
        <airacId>1903</airacId>
      </sourceAIRACs>
      <files>
        <id>AIXMFile/20190317/CDS_721430/23.0.0/AirTrafficManagementService.BASELINE.zip</id>
        <type>AIXMDataSets</type>
        <releaseTime>2019-03-17 03:59:18</releaseTime>
        <fileLength>97801</fileLength>
      </files>
      ...
      <files>
        <id>AIXMFile/20190317/CDS_721430/23.0.0/AirportHeliport.BASELINE.zip</id>
        <type>AIXMDataSets</type>
        <releaseTime>2019-03-17 03:57:17</releaseTime>
        <fileLength>787022</fileLength>
      </files>
```

```
    </datasetSummaries>
  </data>
</as:CompleteAIXMDatasetReply>
```

(7) queryIncrementalAIXMDatasets() - REQUEST

```
<?xml version="1.0" encoding="UTF-8"?>
<airspace:IncrementalAIXMDatasetRequest xmlns:airspace="eurocontrol/cfmu/b2b/AirspaceServices">
  <endUserId>tstcadf1</endUserId>
  <sendTime>2019-03-17 10:00:00</sendTime>
  <queryCriteria>
    <lastKnownUpdateId>721430</lastKnownUpdateId>
  </queryCriteria>
  <timesliceType>PERM_DELTA</timesliceType>
</airspace:IncrementalAIXMDatasetRequest>
```

(8) queryIncrementalAIXMDatasets() - REPLY

```
<?xml version="1.0" encoding="utf-8"?>
<as:IncrementalAIXMDatasetReply xmlns:as="eurocontrol/cfmu/b2b/AirspaceServices" xmlns:cm="eurocontrol/cfmu
  <requestReceptionTime>2019-03-17 13:53:20</requestReceptionTime>
  <requestId>B2B_CUR:13183</requestId>
  <sendTime>2019-03-17 13:53:20</sendTime>
  <status>OK</status>
  <data>
    <datasetSummaries>
      <updateId>721467</updateId>
      <previousUpdateId>721462</previousUpdateId>
      <publicationDate>2019-03-17</publicationDate>
      <sourceAIRACs>
        <airacId>1903</airacId>
      </sourceAIRACs>
      <affectedFeatures/>
    </datasetSummaries>
    <datasetSummaries>
      <updateId>721450</updateId>
      <previousUpdateId>721445</previousUpdateId>
      <publicationDate>2019-03-17</publicationDate>
      <sourceAIRACs>
        <airacId>1903</airacId>
      </sourceAIRACs>
      <affectedFeatures/>
    </datasetSummaries>
    <datasetSummaries>
      <updateId>721439</updateId>
      <previousUpdateId>721430</previousUpdateId>
      <publicationDate>2019-03-17</publicationDate>
      <sourceAIRACs>
        <airacId>1903</airacId>
      </sourceAIRACs>
      <affectedFeatures/>
    </datasetSummaries>
    <datasetSummaries>
      <updateId>721472</updateId>
      <previousUpdateId>721467</previousUpdateId>
```

```
      <publicationDate>2019-03-17</publicationDate>
      <sourceAIRACs>
        <airacId>1903</airacId>
      </sourceAIRACs>
      <files>
        <id>AIXMFile/20190317/IDS_721472/23.0.0/RouteSegment.PERMDELTA.zip</id>
        <type>AIXMDataSets</type>
        <releaseTime>2019-03-17 13:52:27</releaseTime>
        <fileLength>2029</fileLength>
      </files>
      <affectedFeatures>
        <item>
          <key>RouteSegment</key>
          <value>7</value>
        </item>
      </affectedFeatures>
    </datasetSummaries>
    <datasetSummaries>
      <updateId>721456</updateId>
      <previousUpdateId>721450</previousUpdateId>
      <publicationDate>2019-03-17</publicationDate>
      <sourceAIRACs>
        <airacId>1903</airacId>
      </sourceAIRACs>
      <affectedFeatures/>
    </datasetSummaries>
    <datasetSummaries>
      <updateId>721462</updateId>
      <previousUpdateId>721456</previousUpdateId>
      <publicationDate>2019-03-17</publicationDate>
      <sourceAIRACs>
        <airacId>1903</airacId>
      </sourceAIRACs>
      <affectedFeatures/>
    </datasetSummaries>
    <datasetSummaries>
      <updateId>721445</updateId>
      <previousUpdateId>721439</previousUpdateId>
      <publicationDate>2019-03-17</publicationDate>
      <sourceAIRACs>
        <airacId>1903</airacId>
      </sourceAIRACs>
      <affectedFeatures/>
    </datasetSummaries>
  </data>
</as:IncrementalAIXMDatasetReply>
```

(9) Each file must then be downloaded separately using an HTTP GET request.

(10) The HTTP GET to download a specific AIXM file for a given platform/release is:

```
https://<domain>:<port>/<platform_name>/gateway/spec/AIXMFile/<date>/<CDS | IDS id>/<major>.<minor>.<increm
```

where:

a) `<domain>` is:

i) For OPS platform in normal conditions - www.b2b.nm.eurocontrol.int

ii) For PREOPS platform in normal conditions - www.b2b.preops.nm.eurocontrol.int

iii) www.contingency.nm.eurocontrol.int in contingency situations (not available for B2B_PREOPS platforms).

b) `<port>` is 443

c) `<platform_name>` is:

i) `FILE_OPS` for the operational platform

ii) `FILE_PREOPS` for the pre-operational platform

d) `<major>` is the major NM release version (as defined here).

e) `<minor>` is the minor NM release version.

f) `<micro>` is the micro NM release version.

(11) In NM 23, the supported URLs are:

a) Pre-operational: `https://<domain>:<port>/FILE_PREOPS/>/gateway/spec/AIXMFile/<date>/<CDS or IDS id>/23.0.0/<filename>`

b) Operational: `https://<domain>:<port>/FILE_OPS/>/gateway/spec/AIXMFile/<date>/<CDS or IDS id>/23.0.0/<filename>`

(12) Retrieve a complete AIXM data file for Airspace - HTTP GET request:

a) OPS:
```
wget https://www.b2b.nm.eurocontrol.int/FILE_OPS/gateway/spec/AIXMFile/20190317/CDS_721430/23.0.0/Airspace.BASELINE.zip
```

b) PREOPS:
```
wget https://www.b2b.preops.nm.eurocontrol.int/FILE_PREOPS/gateway/spec/AIXMFile/20190317/CDS_721430/23.0.0/Airspace.BASELINE.zip
```

(13) Retrieve a complete AIXM data file for Airspace - File content:

```
M Filemode      Length  Date         Time      File
- ----------  --------  -----------  --------  -----------------
  -rw-r-----  92094816  17-Mar-2019  03:25:50  Airspace.BASELINE
- ----------  --------  -----------  --------  -----------------
             92094816                         1 file
```

(14) Retrieve incremental AIXM data file for RouteSegment - HTTP GET request:

a) OPS:

```
wget https://www.b2b.nm.eurocontrol.int/FILE_OPS/gateway/spec/AIXM-
File/20190317/CDS_721430/23.0.0/RouteSegment.PERMDELTA.zip
```

b) PREOPS:

```
wget https://www.b2b.preops.nm.eurocontrol.int/FILE_PREOPS/gate-
way/spec/AIXMFile/20190317/CDS_721430/23.0.0/RouteSegment.PERM-
DELTA.zip
```

(15) Retrieve incremental AIXM data file for RouteSegment - File content:

```
M Filemode      Length  Date         Time      File
- ----------  --------  -----------  --------  ----------------------
  -rw-r-----     30092  17-Mar-2019  13:51:10  RouteSegment.PERMDELTA
- ----------  --------  -----------  --------  ----------------------
                30092                         1 file
```

Chapter 4. Access to NM NOP/B2B Services

4.1. Client Authentication

(1) In order to gain access to the NM NOP/B2B services, customers need to have officially requested access to them and have signed an Agreement with NM. The associated process is described on the NM website. The reader of the present document is supposed to have already passed that step.

(2) The B2B customer must request access to specific B2B services on the pre-operational and/or the operational platform.

(3) Note: these are two different processes. At the end of the each of these processes the requester receives the technical details on how to access the requested services and the authentication credentials in case of a new customer.

(4) Authentication of the access to the NOP/B2B services is achieved through client-side certificates. For information on PKI, see [3]. The certificates provided by NM are "Enterprise PKI Lite For Personal Digital ID" created for EUROCONTROL by GlobalSign ([4]). For information on the Certificate Practices Statement (CPS) and the Certificate Policy (CP) of these certificates see [5].

(5) NM creates different accounts and issues distinct certificates to access the pre-operational and operational platforms. The certificates for accessing the pre-operational platform are called "test certificates". Access is denied if a client application attempts to use a test certificate to access operational services.

(6) As a result of the access request process, the customer receives from NM a PKCS#12 file that contains both the certificate and the private key. The certificate identifies uniquely the customer. A PIN is needed in order to activate the certificate; the customer obtains this PIN by calling the NM CSO Helpdesk.

(7) The user accounts and the corresponding certificates are independent from the NM release. As long as it is not revoked, an account created for NM release X can also be used for releases X+1, X+2, etc. However, a certificate has a validity period of 3 years. The customer must submit a request for a new certificate via the NM Services Request form at least 2 months before the end of this period.

(8) The customer needs to install the certificate on its application platform. The details on how to do so depend on the said platform. Additionally, the details on how to write/run the client application to authenticate using the certificate depends on the technologies used by the customer.

(9) The certificate is used during the establishment of the SSL connection between the client application and the NM access infrastructure. The data in the certificate must not be corrupted and the certificate itself must not have been revoked via a certificate revocation list.

4.2. Server Authentication

(1) For establishment of the SSL connection the Operational and Pre-operational B2B Web Services servers use an SSL server certificate, corresponding with the domain host name of the server.

(2) B2B client applications may want to store this certificate in their local certificate store and use it for authenticating the server at SSL session establishment time.

(3) The SSL server certificates used by NM typically have a lifetime of one year and are then replaced by a new certificate.

(4) In order to avoid connection problems due to a failing server authentication in the client application after replacement of an SSL server certificate, NM will announce installation of a new SSL server certificate a few days in advance to the B2B Web Services customers. The announcement will specify the date and time of the new certificate's installation and it will include a copy of the new certificate. This should allow customers, who are performing server authentication in their B2B client applications to prepare measures for updating their configurations and so avoid service interruptions.

4.3. Authorisation

4.3.1. Introduction

(1) Each NM B2B service request is subject to authorisation control. All authorisation rules are role-based, meaning that access is granted to a user if he has the requested role. Each request is associated to a use-case and an authorisation rule grants access to the use-case. This means that the same authorisation rule may control access to several service requests.

(2) There are two kinds of authorisation rules:

a) **service-level authorisation**: grants or denies access to the service requests (e.g. `RunwayConfigurationPlanRetrievalRequest`).

b) **business authorisation**: these are additional rules that may restrict access to some parameters or fields in the request and/or the reply and are specific to each operation (e.g. to restrict the possible datasets on which to apply a runway configuration update). These business authorisation rules can be seen as a second-level authorisation that applies after the service-level authorisation has granted access to the service request.

(3) Some service requests are only protected by service-level authorisation which means that access to the operation is either fully granted or fully denied.

(4) Other service requests are protected by both kinds of authorisation rules, which means that a user may have access to the operation but depending on its profile it may or may not have access to all paramaters in the request and/or all fields in the reply.

4.3.2. Authorisation rules

(1) The following tables show for each B2B service request:

- the use case needed for the service-level authorisation;
- the business authorisation rules (if applicable) asssociated with the operation.

4.3.2.1. Airspace Service Group

Service: `AirspaceAvailability`	
Service Request:	AUPChainRetrievalRequest
Authorisation Use Case:	retrieve AUP
Service Request:	AUPCreationRequest
Authorisation Use Case:	manage AUP
Service Request:	AUPDeletionRequest
Authorisation Use Case:	manage AUP
Service Request:	AUPGetManageableRouteSegmentsForAMCAndRouteRequest
Authorisation Use Case:	retrieve AUP
Service Request:	AUPGetManageableRoutesForAMCRequest
Authorisation Use Case:	retrieve AUP
Service Request:	AUPRSAAllocationExpansionRequest
Authorisation Use Case:	manage AUP
Service Request:	AUPRetrievalRequest
Authorisation Use Case:	retrieve AUP
Service Request:	AUPServiceConfigurationRequest
Authorisation Use Case:	manage AUP
Service Request:	AUPUpdateRequest
Authorisation Use Case:	manage AUP
Service Request:	AUPValidationRequest
Authorisation Use Case:	manage AUP
Service Request:	EAUPCDRCompareRequest
Authorisation Use Case:	retrieve EAUP/EUUP

Service: `AirspaceAvailability`	
Service Request:	EAUPCDRRequest
Authorisation Use Case:	retrieve EAUP/EUUP
Service Request:	EAUPChainRetrievalRequest
Authorisation Use Case:	retrieve EAUP/EUUP
Service Request:	EAUPRSACompareRequest
Authorisation Use Case:	retrieve EAUP/EUUP
Service Request:	EAUPRSARequest
Authorisation Use Case:	retrieve EAUP/EUUP

Table 4.1. Service `AirspaceAvailability`

Service: `AirspaceStructure`	
Service Request:	CompleteAIXMDatasetRequest
Authorisation Use Case:	retrieve AIXM Datasets
Service Request:	IncrementalAIXMDatasetRequest
Authorisation Use Case:	retrieve AIXM Datasets

Table 4.2. Service `AirspaceStructure`

4.3.2.2. Common Service Group

Service: `Files`	
Service Request:	FileListRequest
Authorisation Use Case:	File Access

Table 4.3. Service `Files`

4.3.2.3. Flight Service Group

Service: `FlightFiling`	
Service Request:	ExtendedFlightPlanCreationRequest
Authorisation Use Case:	extended Flight Filing

Service: `FlightFiling`	
Service Request:	ExtendedFlightPlanUpdateRequest
Authorisation Use Case:	extended Flight Filing
Service Request:	FilingStatusRequest
Authorisation Use Case:	flight Filing
Service Request:	FlightArrivalRequest
Authorisation Use Case:	flight Filing
Service Request:	FlightDelayRequest
Authorisation Use Case:	flight Filing
Service Request:	FlightDepartureRequest
Authorisation Use Case:	flight Filing
Service Request:	FlightPlanCancellationRequest
Authorisation Use Case:	flight Filing
Service Request:	FlightPlanCreationRequest
Authorisation Use Case:	flight Filing
Service Request:	FlightPlanUpdateRequest
Authorisation Use Case:	flight Filing

Table 4.4. Service `FlightFiling`

Service: `FlightManagement`	
Service Request:	ATCDPIRequest
Authorisation Use Case:	submit Departure Planning Information Message
Business Authorisation:	ANU_Id
Service Request:	CancelDPIRequest
Authorisation Use Case:	submit Departure Planning Information Message
Business Authorisation:	ANU_Id

Service: `FlightManagement`	
Service Request:	EarlyDPIRequest
Authorisation Use Case:	submit Departure Planning Information Message
Business Authorisation:	ANU_Id
Service Request:	TargetDPISequencedRequest
Authorisation Use Case:	submit Departure Planning Information Message
Business Authorisation:	ANU_Id
Service Request:	TargetDPITargetRequest
Authorisation Use Case:	submit Departure Planning Information Message
Business Authorisation:	ANU_Id
Service Request:	GeneralAPIRequest
Authorisation Use Case:	submit Arrival Planning Information Message
Business Authorisation:	ANU_Id
Service Request:	TargetTimeOverAPIRequest
Authorisation Use Case:	submit Arrival Planning Information Message
Business Authorisation:	ANU_Id
Service Request:	TargetTakeOffAPIRequest
Authorisation Use Case:	submit Arrival Planning Information Message
Business Authorisation:	ANU_Id
Service Request:	FlightListByAerodromeRequest
Authorisation Use Case:	retrieve Flight Information
Business Authorisation:	Dataset Proposal Flight MCDM Hotspot Traffic Counts Sensitive Flight TTOF CCAMS Scenario AvoidedRegulations
Service Request:	FlightListByAerodromeSetRequest
Authorisation Use Case:	retrieve Flight Information
Business Authorisation:	Dataset Proposal Flight MCDM Hotspot Traffic Counts Sensitive Flight TTOF CCAMS Scenario AvoidedRegulations

Service: `FlightManagement`	
thorisation:	
Service Request:	FlightListByAircraftOperatorRequest
Authorisation Use Case:	retrieve Flight Information
Business Authorisation:	Dataset Proposal Flight MCDM Hotspot Traffic Counts Sensitive Flight TTOF CCAMS Scenario AvoidedRegulations
Service Request:	FlightListByAirspaceRequest
Authorisation Use Case:	retrieve Flight Information
Business Authorisation:	Dataset Proposal Flight MCDM Hotspot Traffic Counts Sensitive Flight TTOF CCAMS Scenario AvoidedRegulations
Service Request:	FlightListByHotspotRequest
Authorisation Use Case:	retrieve Flight Information
Business Authorisation:	Dataset Proposal Flight MCDM Hotspot Traffic Counts Sensitive Flight TTOF CCAMS Scenario AvoidedRegulations
Service Request:	FlightListByAircraftRegistrationMarkRequest
Authorisation Use Case:	retrieve Flight Information
Business Authorisation:	Dataset Proposal Flight MCDM Hotspot Traffic Counts Sensitive Flight TTOF CCAMS Scenario AvoidedRegulations
Service Request:	FlightListByKeysRequest
Authorisation Use Case:	retrieve Flight Information
Business Authorisation:	Dataset Proposal Flight MCDM Hotspot Traffic Counts Sensitive Flight TTOF CCAMS Scenario AvoidedRegulations

Service: `FlightManagement`	
Service Request:	FlightListByMeasureRequest
Authorisation Use Case:	retrieve Flight Information
Business Authorisation:	Dataset, Proposal Flight, MCDM, Hotspot, Traffic Counts, Sensitive Flight, Flight List By Measure, TTOF, CCAMS, Scenario, AvoidedRegulations
Service Request:	FlightListByPointRequest
Authorisation Use Case:	retrieve Flight Information
Business Authorisation:	Dataset, Proposal Flight, MCDM, Hotspot, Traffic Counts, Sensitive Flight, TTOF, CCAMS, Scenario, AvoidedRegulations
Service Request:	FlightListByTrafficVolumeRequest
Authorisation Use Case:	retrieve Flight Information
Business Authorisation:	Dataset, Proposal Flight, MCDM, Hotspot, Traffic Counts, Sensitive Flight, TTOF, CCAMS, Scenario, AvoidedRegulations
Service Request:	FlightPlanListRequest
Authorisation Use Case:	retrieve Flight Information
Service Request:	FlightRetrievalRequest
Authorisation Use Case:	retrieve Flight Information
Business Authorisation:	Dataset, Proposal Flight, MCDM, Hotspot, Traffic Counts, Sensitive Flight, TTOF, CCAMS, Scenario, AvoidedRegulations
Service Request:	FlightUpdateRequest
Authorisation Use Case:	submit Flight Update Message
Business Authorisation:	FlightUpdateType

Table 4.5. Service `FlightManagement`

Service: `FlightPreparation`	
Service Request:	ExtendedFlightPlanValidationRequest
Authorisation Use Case:	extended Flight Preparation
Service Request:	FlightPlanValidationRequest
Authorisation Use Case:	flight Preparation
Service Request:	RoutingAssistanceRequest
Authorisation Use Case:	flight Preparation

Table 4.6. Service `FlightPreparation`

Service: `FlightSafety`	
Service Request:	ACC3AccreditationListReplacementRequest
Authorisation Use Case:	update ACC3 Information
Service Request:	TCOAuthorisationListReplacementRequest
Authorisation Use Case:	update TCO Information
Service Request:	TCOAuthorisationListUpdateRequest
Authorisation Use Case:	update TCO Information

Table 4.7. Service `FlightSafety`

4.3.2.4. Flow Service Group

Service: `MCDM`	
Service Request:	MCDMTopicListRequest
Authorisation Use Case:	manage MCDM
Business Authorisation:	Dataset
Service Request:	MCDMTopicMessageRetrievalRequest
Authorisation Use Case:	manage MCDM
Business Authorisation:	Dataset
Service Request:	MCDMTopicUpdateRequest

Service: MCDM	
Authorisation Use Case:	manage MCDM
Business Authorisation:	Dataset
Service Request:	MCDMStateUpdateRequest
Authorisation Use Case:	manage MCDM
Business Authorisation:	Dataset

Table 4.8. Service MCDM

Service: Simulations	
Service Request:	SimulationListRequest
Authorisation Use Case:	manage Simulations
Service Request:	SimulationAvailabilityRequest
Authorisation Use Case:	manage Simulations
Service Request:	SimulationStartRequest
Authorisation Use Case:	manage Simulations
Service Request:	SimulationStopRequest
Authorisation Use Case:	manage Simulations

Table 4.9. Service Simulations

Service: ScenarioRepository	
Service Request:	ScenarioRegulationRetrievalRequest
Authorisation Use Case:	manage Simulations
Service Request:	ScenarioReroutingRetrievalRequest
Authorisation Use Case:	manage Simulations
Service Request:	ScenarioMCDMOnlyRetrievalRequest
Authorisation Use Case:	manage Simulations
Service Request:	ScenarioListRequest

Service: `ScenarioRepository`	
Authorisation Use Case:	manage Simulations

Table 4.10. Service `ScenarioRepository`

Service: `Measures`			
Service Request:	ATFCMSituationRequest		
Authorisation Use Case:	retrieve ATFCMS situation		
Business Authorisation:	Dataset		
Service Request:	NetworkImpactAssessmentRetrievalRequest		
Authorisation Use Case:	retrieve the Network Impact Assessment		
Business Authorisation:	Dataset		
Service Request:	AddFlightsToMeasureRequest		
Authorisation Use Case:	force CTOT		
Business Authorisation:	Dataset		
Service Request:	MCDMOnlyCancelRequest		
Authorisation Use Case:	manage MCDM only Measure		
Business Authorisation:	Dataset	Cherry pick	Non cherry pick (*)
Service Request:	MCDMOnlyCreationRequest		
Authorisation Use Case:	manage MCDM only Measure		
Business Authorisation:	Dataset	Hotspot Cherry pick	Non cherry pick (*)
Service Request:	MCDMOnlyListRequest		
Authorisation Use Case:	retrieve MCDM only Measure		
Business Authorisation:	Dataset	Hotspot	
Service Request:	MCDMOnlyUpdateRequest		
Authorisation Use Case:	manage MCDM only Measure		
Business Authorisation:	Dataset	Hotspot Cherry pick	Non cherry pick (*)
Service Request:	MeasureOpLogRetrievalRequest		

Service: `Measures`	
Authorisation Use Case:	retrieve Measure Oplog Information
Business Authorisation:	Dataset Measure Id
Service Request:	RegulationCancelRequest
Authorisation Use Case:	manage Regulation Measure
Business Authorisation:	Dataset Cherry pick Non cherry pick
Service Request:	RegulationCreationRequest
Authorisation Use Case:	manage Regulation Measure
Business Authorisation:	Dataset Hotspot MCDM Cherry pick Non cherry pick
Service Request:	RegulationListRequest
Authorisation Use Case:	retrieve Regulation Information
Business Authorisation:	Dataset Hotspot MCDM
Service Request:	RegulationProposalFilingRequest
Authorisation Use Case:	manage Regulation Proposal Measure
Business Authorisation:	Dataset Hotspot MCDM
Service Request:	RegulationProposalListRequest
Authorisation Use Case:	retrieve Regulation Proposal Information
Business Authorisation:	Dataset Hotspot MCDM
Service Request:	RegulationProposalRevocationRequest
Authorisation Use Case:	manage Regulation Proposal Measure
Business Authorisation:	Dataset
Service Request:	RegulationProposalUpdateRequest
Authorisation Use Case:	manage Regulation Proposal Measure
Business Authorisation:	Dataset Hotspot MCDM
Service Request:	RegulationUpdateRequest
Authorisation Use Case:	manage Regulation Measure

Service: `Measures`	
Business Authorisation:	Dataset Hotspot MCDM Cherry pick Non cherry pick
Service Request:	RemoveFlightsFromMeasureRequest
Authorisation Use Case:	force CTOT
Business Authorisation:	Dataset
Service Request:	ReroutingCancelRequest
Authorisation Use Case:	manage Rerouting Measure
Business Authorisation:	Dataset Cherry pick (*) Non cherry pick Rerouting Apply Kind
Service Request:	ReroutingCreationRequest
Authorisation Use Case:	manage Rerouting Measure
Business Authorisation:	Dataset Hotspot MCDM Cherry pick (*) Non cherry pick Rerouting Apply Kind
Service Request:	ReroutingListRequest
Authorisation Use Case:	retrieve Rerouting Information
Business Authorisation:	Dataset Hotspot MCDM
Service Request:	ReroutingUpdateRequest
Authorisation Use Case:	manage Rerouting Measure
Business Authorisation:	Dataset Hotspot MCDM Cherry pick (*) Non cherry pick Rerouting Apply Kind

Table 4.11. Service `Measures`

Service: `TrafficCounts`	
Service Request:	TrafficCountsByAerodromeRequest
Authorisation Use Case:	retrieve Traffic Counts
Business Authorisation:	Dataset
Service Request:	TrafficCountsByAerodromeSetRequest
Authorisation Use Case:	retrieve Traffic Counts

Service: `TrafficCounts`	
Business Authorisation:	Dataset
Service Request:	TrafficCountsByAircraftOperatorRequest
Authorisation Use Case:	retrieve Traffic Counts
Business Authorisation:	Dataset
Service Request:	TrafficCountsByAirspaceRequest
Authorisation Use Case:	retrieve Traffic Counts
Business Authorisation:	Dataset
Service Request:	TrafficCountsByPointRequest
Authorisation Use Case:	retrieve Traffic Counts
Business Authorisation:	Dataset
Service Request:	TrafficCountsByTrafficVolumeRequest
Authorisation Use Case:	retrieve Traffic Counts
Business Authorisation:	Dataset Scenario

Table 4.12. Service `TrafficCounts`

Service: `TacticalUpdates`	
Service Request:	CapacityPlanRetrievalRequest
Authorisation Use Case:	retrieve Capacity Plan
Business Authorisation:	Dataset
Service Request:	CapacityPlanUpdateRequest
Authorisation Use Case:	update Capacity Plan
Business Authorisation:	Dataset
Service Request:	OTMVPlanRetrievalRequest
Authorisation Use Case:	retrieve OTMV Plan
Business Authorisation:	Dataset

Service: `TacticalUpdates`	
Service Request:	OTMVPlanUpdateRequest
Authorisation Use Case:	update OTMV Plan
Business Authorisation:	Dataset
Service Request:	RunwayConfigurationPlanRetrievalRequest
Authorisation Use Case:	retrieve Runway Configuration Plan
Business Authorisation:	Dataset
Service Request:	RunwayConfigurationPlanUpdateRequest
Authorisation Use Case:	update Runway Configuration Plan
Business Authorisation:	Dataset
Service Request:	SectorConfigurationPlanRetrievalRequest
Authorisation Use Case:	retrieve Sector Configuration Plan
Business Authorisation:	Dataset
Service Request:	SectorConfigurationPlanUpdateRequest
Authorisation Use Case:	update Sector Configuration Plan
Business Authorisation:	Dataset
Service Request:	TrafficVolumeActivationPlanRetrievalRequest
Authorisation Use Case:	retrieve Traffic Volume Activation Plan
Business Authorisation:	Dataset
Service Request:	TrafficVolumeActivationPlanUpdateRequest
Authorisation Use Case:	update Traffic Volume Activation Plan
Business Authorisation:	Dataset
Service Request:	HotspotListRequest
Authorisation Use Case:	retrieve Hotspot Information
Business Authorisation:	Dataset HotspotKind
Service Request:	HotspotPlanUpdateRequest

Service: `TacticalUpdates`		
Authorisation Use Case:	manage Hotspot	
Business Authorisation:	Dataset	HotspotKind

Table 4.13. Service `TacticalUpdates`

4.3.2.5. General Information Service Group

Service: `AIMs`	
Service Request:	AIMListRequest
Authorisation Use Case:	retrieve AIM
Service Request:	AIMRetrievalRequest
Authorisation Use Case:	retrieve AIM

Table 4.14. Service `AIMs`

Service: `NMB2BInfo`	
Service Request:	NMReleaseInformationRequest
Authorisation Use Case:	retrieve NM B2B Information
Service Request:	NMB2BReferenceManualsRequest
Authorisation Use Case:	retrieve NM B2B Information
Service Request:	NMB2BWSDLsRequest
Authorisation Use Case:	retrieve NM B2B Information
Service Request:	NMB2BScenariosRequest
Authorisation Use Case:	retrieve NM B2B Information
Service Request:	NMB2BAddendaErrataRequest
Authorisation Use Case:	retrieve NM B2B Information

Table 4.15. Service `NMB2BInfo`

4.3.2.6. Publish-Subscribe Service Group

Service: `Messaging`	
Service Request:	MessagePullRequest
Authorisation Use Case:	manage Subscription

Table 4.16. Service `Messaging`

Service: `SubscriptionManagement`		
Service Request:	SubscriptionCreationRequest	
Authorisation Use Case:	manage Subscription	
Business Authorisation:	Subscription Topic	Subscription To All Flights
Service Request:	SubscriptionDeletionRequest	
Authorisation Use Case:	manage Subscription	
Service Request:	SubscriptionHistoryRequest	
Authorisation Use Case:	manage Subscription	
Service Request:	SubscriptionListRequest	
Authorisation Use Case:	manage Subscription	
Service Request:	SubscriptionPauseRequest	
Authorisation Use Case:	manage Subscription	
Service Request:	SubscriptionResumeRequest	
Authorisation Use Case:	manage Subscription	
Service Request:	SubscriptionRetrievalRequest	
Authorisation Use Case:	manage Subscription	

Table 4.17. Service `SubscriptionManagement`

4.3.3. Business authorisation details description

(1) This paragraph provides details on the business authorisation rules mentioned in the tables above.

(2) For each business authorisation rule it gives a description of what it is and what it restricts.

(1) **Authorisation rule: Dataset**
Restricts access to the different types of dataset (e.g. OPERATIONAL, FORECAST, etc.) used in Flight, Flow and Airspace services.
Where used:

- The `dataset` attribute defined in B2B requests belonging to the `flight.FlightManagement`
- The `dataset` attribute defined in B2B requests belonging to the `flow.TrafficCounts` service.
- The `dataset` attribute defined in B2B requests belonging to the `flow.Measures`
- The `dataset` attribute defined in B2B requests belonging to the `flow.TacticalUpdates`
- The `dataset` attribute defined in B2B requests belonging to the `flow.Mcdm`

(1) **Authorisation rule: MeasureId**
Restricts access to the different types of measures (e.g. REGULATION, REROUTING, etc.).
Where used:

- `measureId` attribute defined in the `MeasureOpLogRetrievalRequest` B2B request.

(1) **Authorisation rule: ProposalFlight**
Restricts access to proposal flights in a flight list or in regulation proposal.

(1) **Authorisation rule: SensitiveFlight**
Restricts access to sensitive flights in a flight list. Only special profiles can retrieve sensitive flights. These profiles are defined by NM.

(1) **Authorisation rule: TargetTimeOverFix**
Restricts access to the target time over fix. The TTOF is the target time over the relevant flight profile point for the most penalizing regulation of the flight and the actual time over (according to the CTFM point profile)
Note that the TargetTime related fields are trial related (SESAR) fields: they are only accessible (authorized) during specific trials or on test platforms.

(1) **Authorisation rule: FlightListByMeasure**
Allows retrieving flights affected by a measure.

(1) **Authorisation rule: Hotspot**
Allows retrieving information associated to a Hotspot.
Where used:

- Access to the B2B request `FlightListByHotspotRequest`.
- The `sourceHotspot` attribute defined in B2B requests belonging to the `Measures`
- The `hotspots` request flight field defined in B2B requests belonging to the `FlightManagement` service.

- The `caughtInHotspots` request flight field defined in B2B requests belonging to the `FlightManagement` service.

(1) **Authorisation rule: MCDM**
Allows retrieving information associated to MCDM.
Where used:

- The `mcdmInfo` request flight field defined in B2B requests belonging to the `FlightManagement` service.
- The `mcdmRequired` attribute defined in B2B requests belonging to the `Measures` service.

(1) **Authorisation rule: ANU_Id**
This rule is used to verify that the data requested or provided by a user is associated to the Air Navigation Unit.
Where used:

- The departure aerodrome in all DPI-related requests.
- The arrival aerodrome in all API-related requests.

(1) **Authorisation rule: FlightUpdateType**
This rule is used to verify that the flight update message's type send by the user is authorised.
Where used:

- Submit Flight Update Message service request.

(2)

Flight Update's type	Authorisation Use Case
DepartureInformation	submit FSA departure
EnRouteInformation	submit FSA en route
LandingInformation	submit APR arrival
AircraftPositionReport	submit APR en route
OceanicInformation from GANDER's OAC	submit FNM
OceanicInformation from SHANWICK's OAC or SANTA_MARIA's OAC	submit MFS

Table 4.18. Flight Update authorisation detail

(1) **Authorisation rule: TrafficCounts**
Restricts access to information related to traffic counts.
Where used:

a) The `worstLoadStateAtReferenceLocation` request flight field defined in B2B requests belonging to the `FlightManagement` service.

(1) **Authorisation rule: CherryPick**
Allows management of Cherry-Pick measures (on selected flights).

Remark: If access to non-cherry-pick rerouting is allowed, then access to cherry-pick rerouting is automatically allowed.
Where used:

a) The `measureCherryPicked` attribute defined in B2B requests belonging to the `Measures` service.

(1) **Authorisation rule: NonCherryPick**
Allows management of Non-Cherry-Pick measures (on flights belonging to a flow).
Remark: MCDM-Only measures can only be cherry-pick.
Where used:

a) The `measureCherryPicked` attribute defined in B2B requests belonging to the `Measures` service.

(1) **Authorisation rule: ReroutingApplyKind**
Restricts management of different kinds of applied rerouting measures.
Where used:

a) The `reroutingApplyKind` attribute defined in B2B requests belonging to the `Measures` service.

(1) **Authorisation rule: AvoidedRegulations**
Restricts access to information related to zero-rate regulations that have been impacting a flight.
Where used:

a) The `avoidedRegulations` request flight field defined in B2B requests belonging to the `FlightManagement` service.

(1) **Authorisation rule: PS_Topic**
Subscriptions are subject to topic-based authorisation, which means that a user may or may not have access to some subscription topics depending on its role.
In addition to this, some subscriptions are also subject to business authorisation rules to check that the user has access to all the fields requested in the message payload configuration.
Where used:

a) The `topic` attribute defined in SubscriptionCreationRequest B2B requests belonging to the `SubscriptionManagement` service.

TOPIC	Authorisation Use Case
AIXM	retrieve AIXM Datasets
AIM	retrieve AIM
EAUP	retrieve EAUP/EUUP
FLIGHT_DATA	retrieve Flight Information
FLIGHT_PLANS	retrieve Flight Information
REGULATIONS	retrieve Regulation Information

TOPIC	Authorisation Use Case
FLIGHT_FILING_RESULT	flight Filing

Table 4.19. PS_Topic's Authorisation

b) The `regulationFields` attribute defined in `RegulationPayloadConfiguration` type belonging to the `SubscriptionManagement` service.

c) The `flightFields` attribute defined in `FlightDataPayloadConfiguration` type belonging to the `SubscriptionManagement` service.

(1) **Authorisation rule: CCAMS**
Allows retrieving information associated to CCAMS.
Where used:

a) The `ccamsSSRCode` request flight field defined in B2B requests belonging to the `FlightManagement` service.

(1) **Authorisation rule: SubscriptionToAllFlights**
Subscriptions to all flight plans or all flights are subject to topic-based authorisation.
When the subscription filter is empty and the business authorization to subscribe to all fligh plans and flights is granted, any modification on any flight plan or flight will be communicated.

(1) **Authorisation rule: HotspotKind**
Identifies the kind of hotspot.

a) LOCATION_OF_INTEREST
Location of interest hotspots are used raise awarness about potential hotspots or problems in the context of the daily plan: For example weather or special events (a.o. Soyuz rocket launches that imply a closure of some airspaces for a period of time).

b) PROBLEM
Problem hotspots are STAM related hotspots and are linked to a demand-capacity inbalance. Typically a Location of interest hotspot can evolve into a problem hotspot if the potential risk metrializes and introduces a real demand-capacity inbalance for a specific period.

(1) **Authorisation rule: Scenario**
Allows retrieving information associated to Scenario Repository.
Where used:

a) The `applicableScenarios` request flight field defined in B2B requests belonging to the `FlightManagement` service.

b) The `computeFlowCounts` equals to `SCENARIO` in B2B requests belonging to the `TrafficCounts` service.

4.3.4. Delegated Access authorisation control

4.3.4.1. Description

(1) Delegated Access Authorisation (DAA) allows the administrator at an organisation level to specify the operations that can be performed on their data and by whom.

(2) This feature is disabled by default, and the impacted B2B services do not require permissions until DAA is turned on for specific data type by the data owner via the NOP Portal Administration interface. From this moment, the Air Navigation Unit of the B2B User must be granted access in order to be able to perform updates on these data.

(3) In order to support possible rejections of requests because of DAA, an optional reason field was added to the Reply type to provide a reason for the returned status, especially when the status is not set to ReplyStatus.OK.

4.3.4.2. Impacted services

4.3.4.2.1. FlightServices

4.3.4.2.1.1. FlightFilingService

(1) An aircraft operator can decide to restrict updates performed on the flights it operates (Only applies to NM B2B and NOP Portal).

(2) The flight plan originator can always update its submitted flight plans, even if it is not the actual operator and it did not receive delegated access by the flight operator.

(3) In case a B2B request is not authorised because of DAA, the following status/reason is returned: **NOT_AUTHORISED** / **No delegation of access from the flight owner for that operation.**

(4) The requestor shall contact the Aircraft Operator and ask to be added to the list of delegated organisations

(5) The following operations are subject to DAA:

(6) Cancel Flight:

a) fileFlightPlanCancellation

(7) Delay Flight:

a) fileFlightDelay

(8) Update Flight:

a) fileFlightPlanUpdate

b) fileExtendedFlightPlanUpdate

(9) File Flight Plan:

a) fileNewFlightPlan

b) fileNewExtendedFlightPlan

(10) Request Flight Plan:

a) Not applicable

Chapter 5. Web Services Lifecycle

5.1. Introduction

(1) The chosen approach is based on the following facts:

a) The service contracts must be released some months before the service becomes operational (officially released on the NM operational platform).

b) It is convenient for the B2B service consumers to have access to the NM B2B services on a test platform to test client applications before accessing the NM operational services. NM calls this platform the "pre-operational" (PREOPS) platform and the services deployed on it are called the "pre-operational" services.

c) The pre-operational services must remain available a "long" time after they become operational, to allow customers to work on their client applications with the actual NM B2B services without affecting the operational platform.

d) It is convenient for the B2B service consumers to have early access to draft implementations of the NM B2B services, therefore NM will, when possible, release them as is on the pre-operational platform some months before the services becomes operational.

e) B2B service consumers do not release their client applications at the same time as NM releases these services in operations. Therefore, NM must support multiple versions of its web services at all times.

f) Agility commands that NM can release pre-operational and operational services within the lifetime of a NM release, as long as it does not break the two statements above.

(2) This section defines the NM B2B web services lifecycle in support to the aforementioned constraints.

(3) With the purpose of formally defining the lifecycle, this chapter starts with a short definition of relevant concepts and terminology related to versioning. These concepts and terminology are mostly borrowed from [1].

5.2. Concepts

5.2.1. Backwards Compatibility

(1) A new version for a service contract that continues to support consumer programs designed to work with the old version is said to be backwards-compatible. From a design perspective, this means that the new contract has not changed in such a way that it can impact existing consumer programs that are already using the contract.

(2) Obvious examples of backwards-compatible changes are additions of new services, new operations to existing services, or additions of optional elements in existing message definitions.

5.2.2. Forwards Compatibility

(1) When a service contract is designed in such a way that it can support a range of future consumer programs, it is considered to have an extent of forwards-compatibility. This means that the contract can essentially accommodate how consumer programs will evolve over time.

(2) The usual means by which forwards compatibility is attempted in message definitions is through the use of wildcards.

5.2.3. Compatible Changes

(1) When a change is made to a web service that does not negatively affect its existing consumers, the change itself is considered as a compatible change.

(2) Backwards and forwards compatibility can be obtained via compatible changes.

(3) The most common compatible changes are:

a) adding a new WSDL operation definition and associated message definitions

b) adding a new WSDL port type definition and associated operation definitions

c) adding a new WSDL binding and service definitions

d) adding a new optional XML Schema element or attribute declaration to a message definition

e) reducing the constraint granularity of an XML Schema element or attribute of a message definition type

f) adding a new XML Schema wildcard to a message definition type

5.2.4. Incompatible Changes

(1) If after a change, the contract is no longer compatible with consumers, it is considered to have received an incompatible change.

(2) These are the types of changes that can break an existing contract and therefore impose the most challenges when it comes to versioning.

(3) Note that an incompatible change indicates the absence of backwards compatibility but not necessarily forwards compatibility as the latter is loose by definition.

(4) Common incompatible changes are:

a) renaming an existing WSDL operation definition

b) removing an existing WDSL operation definition

c) adding a fault message to an existing WSDL operation definition

d) adding a new required XML Schema element or attribute declaration to a message definition

e) increasing the constraint granularity of an XML Schema element or attribute declaration of a message definition

f) renaming an optional or required XML Schema element or attribute declaration in a message definition

g) removing an optional or required XML Schema element or attribute declaration or wildcard from a message definition

5.2.5. Versioning Strategies

(1) Even though there is no de-facto versioning technique for the WSDL, XML Schema and WSPolicy that comprises web service contracts, some known strategies have emerged:

a) **Strict** - Any compatible or incompatible changes result in a new version of the service contract. This approach does not support backwards and forwards compatibility.

b) **Flexible** - Any incompatible change results in a new version of the service contract and the contract is designed to support backwards compatibility but not forwards compatibility.

c) **Loose** - Any incompatible change results in a new version of the service contract and the contract is designed to support backwards and forwards compatibility.

5.3. Lifecycle

(1) This section describes how NM implements its web services lifecycle.

5.3.1. Pre-operational (PREOPS) Platform

(1) NM offers a pre-operational (PREOPS) platform that service consumers use for testing purposes. The software to be released in the new NM releases is deployed on the PREOPS platform N months before it goes operational, so that consumers can start working on the new contract in advance. However, since NM does not impose to its consumers to be ready at the moment of its operational release, the PREOPS services remain available after the operational release.

(2) The PREOPS platform is not the operational platform. As said above, this PREOPS platform is made available to allow NM customers to test the interoperability of their applications with the NOP/B2B services. The quality of the PREOPS services may be lower than the quality of the operational services. In particular, availability is not guaranteed 24/7, as support is only available during office hours. Additionally, the quality of the data on the PREOPS platform depends on the service group and is therefore defined in the specific reference manuals.

(3) Furthermore, the quality of a new or modified service just after its release on the PREOPS platform may be lower, since the NM development and testing activities may not be entirely finished at that point in time. However, NM considers that this early release may still be useful to NM customers, as it allows for early access to the services and therefore early testing.

5.3.2. Releases

(1) NM classifies NM B2B web services releases as follows:

a) **Major release**

i) NM deploys new NM B2B Web Service major releases as many times as required by the business demand with a maximum number of major releases per year - as of today this number is fixed to two major releases per year.

ii) A major release is represented by a sequence of two tokens, e.g. 21.5.

iii) The strict versioning strategy is applied from one major NM release to the other, to all the NM B2B web services at the same time. The version identifier of any service artefact is therefore the version identifier of its major NM release, namely 21.0, 21.5, and so forth.

iv) Each major release of the NM B2B Web Services deployed on the NM OPS platform remains available on this platform a fixed duration after its deployment - as of today this duration is fixed to two calendar years.

b) **In case of backwards incompatible change**

i) In some cases, NM might have to upgrade a major release with backwards incompatible changes, before it completes its lifecycle (as of today, two calendar years after OPS deployment). Such a backwards incompatible upgrade is delivered via a minor release, will occur as rarely as possible and will be communicated to potentially impacted customers via special means and early enough for them to plan action.

ii) NM ensures that this release is planned and deployed on PREOPS platform sufficiently in advance so that customers have enough time to upgrade their code and prepare their migration.

5.4. Service Platforms

(1) NM Web Services are exposed on two platforms:

a) The pre-operational platform is the testing platform. It is aimed at supporting the customer in the development of its client applications

b) The operational platform

(2) NM deploys new Web Services version several times a year, typically twice a year.
The version identifier of any service artifact is therefore the version identifier of its NM release, e.g. 18.0.0, 18.5.0, and so forth.

(3) NM guaranties that each version of the Web Services deployed on the operational platform remains available during two years after the deployment.

(4) New NM Web Services versions are deployed on the pre-operational platform a few weeks before being deployed on the operational platform.

(5) NM applies the same deployment policy on the pre-operational and the operational platforms.

(6) The quality of the pre-operational services may be lower than the quality of the operational services. In particular, availability is not guaranteed 24/7, as support is only available during office hours. Additionally, the quality of the data on the pre-operational platform depends on the service group and is therefore defined in the specific reference manuals.

Chapter 6. Service Location

(1) As said above, the URL of a service conveys both the pre-operational/operational platform and the NM release.

(2) Besides, since a request type defines entirely the port type that handles it and the request type is part of the message, the location of the port type is actually independent of the port type: for a given platform and NM release, NM exposes all its B2B services via a single URL.

(3) Furthermore, to cope with contingency situations, the current deployment architecture of the NOP/B2B services includes two distinct sites. The B2B services are always served by a single site. In normal conditions it is the primary site, and in contingency situations the contingency site.

(4) The contingency procedure is activated in case of *major* catastrophe at the primary site and is announced by different means, including AIMs, contingency web site, press, etc. The access to the contingency site must only be done in case the NM contingency has been announced.

(5) The service location for a given platform/release is:

```
https://<domain>:<port>/<platform_name>/gateway/spec/<major>.<minor>.<increment>
```

where:

a) `<domain>` is:

i) For OPS platform in normal conditions - www.b2b.nm.eurocontrol.int

ii) For PREOPS platform in normal conditions - www.b2b.preops.nm.eurocontrol.int

iii) www.contingency.nm.eurocontrol.int in contingency situations (not available for B2B_PREOPS platforms).

b) `<port>` is 443

c) `<platform_name>` is:

i) `B2B_OPS` for the operational platform

ii) `B2B_PREOPS` for the pre-operational platform

d) `<major>` is the major NM release version (as defined here).

e) `<minor>` is the minor NM release version.

f) `<micro>` is the micro NM release version.

(6) In NM 23, the supported URLs are:

a) Pre-operational: `https://<domain>:<port>/B2B_PREOPS/>/gateway/spec/23.0.0`

b) Operational: `https://<domain>:<port>/B2B_OPS/gateway/spec/23.0.0`

(7) If any text follows the `gateway/spec` path in one of these URLs, the server returns the HTTP status 400 (Bad Request).

Chapter 7. Request Prioritisation

(1) All method calls have exactly the same priority, regardless of the method or the customer.

(2) As explained here, a `Reply` comes with a `ReplyStatus` value. In case of overload of any NM system component on the request processing path, the `ReplyStatus` value is `OVERLOAD`.

(3) On the one hand, NM does not wish to impose limitations on the number of calls that a given customer can issue: NM will process calls as long as its resources are not exhausted.

(4) On the other hand, NM implements a mechanism that precludes the situation where the heavy demand from a certificate owner would prevent another certificate owner to access the NM services.

(5) This implementation is reflected in the table below. Two parameters are associated to each B2B Service request:

a) Max Call Count Per Unit Of Time

b) Overload Window Size In Secs

(6) If, for a specific B2B request type, a user reaches the first parameter threshold value before the time window expires, he will get an overload exception for the remaining time of this window. Example: the threshold of the B2B request `RunwayConfigurationPlanUpdateRequest` is 30 calls per 60 seconds. If a user has requested 30 calls in 50 seconds, he will get overload exceptions during the remaining 10 seconds of the time window. The B2B request `RunwayConfigurationPlanUpdateRequest` will resume normally at the beginning of the next 60 seconds window.

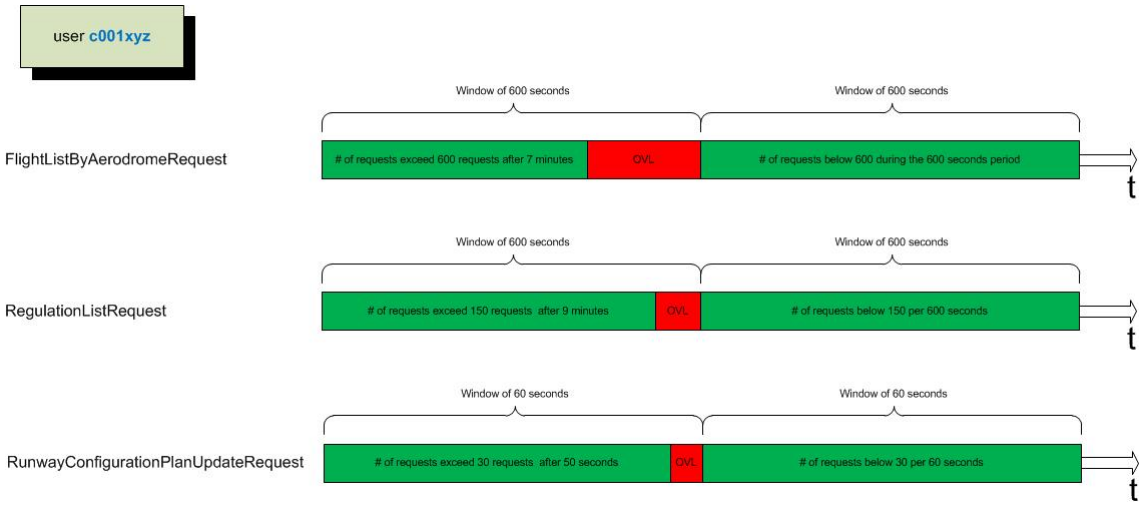

Figure 7.1. Number of Requests per Unit of Time

(7)

Disclaimer

The threshold values provided in the table below are subject to change at any given time. Communication about threshold value's change shall be done via an announcement on the NM B2B services OneSky Team site. This includes emails to all SPOCs having raised such an alert in the NM B2B services OneSky Team site. NM reserves the right to modify these threshold values in case critical operational services are jeopardised by heavy usage, misuse or abuse, in order to ensure the continuity of these essential services.

(8)

Service Group	Service	Service Request	Max Call Count Per Unit Of Time	Overload Window Size In Secs
common	Files	FileListRequest	300	600
generalinformation	AIMs	AIMListRequest	150	600
		AIMRetrievalRequest	555	600
generalinformation	NMB2BInfo	NMReleaseInformationRequest	5	60
		NMB2BReferenceManualsRequest	5	60
		NMB2BWSDLsRequest	5	60
		NMB2BScenariosRequest	5	60
		NMB2BAddendaErrataRequest	5	60
airspace	AirspaceAvailability	AUPChainRetrievalRequest	30	60
		AUPRetrievalRequest	30	60
		AUPCreationRequest	30	60
		AUPUpdateRequest	30	60
		AUPValidationRequest	30	60
		AUPDeletionRequest	30	60
		AUPRSAAllocationExpansionRequest	30	60

Service Group	Service	Service Request	Max Call Count Per Unit Of Time	Overload Window Size In Secs
		AUPServiceConfigurationRequest	30	60
		EAUPChainRetrievalRequest	375	600
		EAUPCDRRequest	225	600
		EAUPCDRCompareRequest	150	600
		EAUPRSARequest	125	600
		EAUPRSACompareRequest	150	600
		AUPGetManageableRoutesForAMCRequest	30	60
		AUPGetManageableRouteSegmentsForAMCAndRouteRequest	30	60
	AirspaceStructure	CompleteAIXMDatasetRequest	300	600
		IncrementalAIXMDatasetRequest	300	600
flight	FlightPreparation	FlightPlanValidationRequest	3450	600
		ExtendedFlightPlanValidationRequest	30	60
		RoutingAssistanceRequest	30	60
	FlightFiling	FilingStatusRequest	150	600
		FlightPlanCreationRequest	150	600
		FlightPlanUpdateRequest	150	600
		FlightPlanCancellationRequest	150	600
		FlightDelayRequest	150	600

Service Group	Service	Service Request	Max Call Count Per Unit Of Time	Overload Window Size In Secs
		FlightDeparture-Request	150	600
		FlightArrivalRequest	150	600
		FilingStatusRequest	150	600
		ExtendedFlightPlan-CreationRequest	30	60
		ExtendedFlight-PlanUpdateRequest	30	60
	FlightManagement	FlightPlanLis-tRequest	975	600
		FlightListByKey-sRequest	3750	600
		FlightListByAircraft-OperatorRequest	150	600
		FlightListByAero-dromeRequest	600	600
		FlightListByAero-dromeSetRequest	225	600
		FlightListByAirspace-Request	150	600
		FlightListByPointRe-quest	150	600
		FlightListByTraffic-VolumeRequest	600	600
		FlightListByMeasure-Request	300	600
		FlightListByHotspo-tRequest	300	600
		FlightListByAir-craftRegistration-MarkRequest	300	600
		FlightRetrievalRe-quest	5250	600
		PredictedDPIRe-quest	30	60
		EarlyDPIRequest	30	60

Service Group	Service	Service Request	Max Call Count Per Unit Of Time	Overload Window Size In Secs
		TargetDPITargetRequest	30	60
		TargetDPISequencedRequest	30	60
		ATCDPIRequest	30	60
		CancelDPIRequest	30	60
		FlightUpdateRequest	30	60
		GeneralAPIReques	30	60
		TargetTakeOffAPIRequest	30	60
		TargetTimeOverAPIRequest	30	60
		FlightSafety	ACC3AccreditationListReplacementRequest	30 60
	TCOAuthorisationListReplacementRequest		30	60
	TCOAuthorisationListUpdateRequest		30	60
flow	TrafficCounts	TrafficCountsByAircraftOperatorRequest	300	600
		TrafficCountsByAerodromeRequest	300	600
		TrafficCountsByAerodromeSetRequest	300	600
		TrafficCountsByAirspaceRequest	300	600
		TrafficCountsByPointRequest	300	600
		TrafficCountsByTrafficVolumeRequest	600	600

Service Group	Service	Service Request	Max Call Count Per Unit Of Time	Overload Window Size In Secs
	Measures	RegulationListRequest	150	600
		RegulationCreationRequest	30	60
		RegulationUpdateRequest	30	60
		RegulationCancelRequest	30	60
		RegulationProposalListRequest	30	60
		RegulationProposalFilingRequest	30	60
		RegulationProposalUpdateRequest	30	60
		RegulationProposalRevocationRequest	30	60
		ReroutingListRequest	150	600
		ReroutingCreationRequest	30	60
		ReroutingUpdateRequest	30	60
		ReroutingCancelRequest	30	60
		MCDMOnlyListRequest	150	600
		MCDMOnlyCreationRequest	30	60
		MCDMOnlyUpdateRequest	30	60
		MCDMOnlyCancelRequest	30	60
		MeasureOpLogRetrievalRequest	30	60
		RegulationForceDelayRequest	30	60

Service Group	Service	Service Request	Max Call Count Per Unit Of Time	Overload Window Size In Secs
		AddFlightsToMeasureRequest	30	60
		RemoveFlightsFromMeasureRequest	30	60
		ATFCMSituationRequest	30	60
	NetworkImpactAssessmentRetrievalRequest	30	60	
	Mcdm	MCDMTopicListRequest	400	600
		MCDMTopicUpdateRequest	30	60
		MCDMTopicMessageRetrievalRequest	300	600
		MCDMStateUpdateRequest	30	60
	TacticalUpdates	HotspotListRequest	300	600
		HotspotPlanUpdateRequest	30	60
		SectorConfigurationPlanRetrievalRequest	400	60
		SectorConfigurationPlanUpdateRequest	30	60
		CapacityPlanRetrievalRequest	400	600
		CapacityPlanUpdateRequest	30	60
		TrafficVolumeActivationPlanRetrievalRequest	500	600
		TrafficVolumeActivationPlanUpdateRequest	30	60
		OTMVPlanRetrievalRequest	400	600

Service Group	Service	Service Request	Max Call Count Per Unit Of Time	Overload Window Size In Secs
		OTMVPlanUpdateRequest	30	60
		RunwayConfigurationPlanRetrievalRequest	400	600
		RunwayConfigurationPlanUpdateRequest	30	60
	ScenarioRepository	ScenarioReroutingRetrievalRequest	30	60
		ScenarioRegulationRetrievalRequest	30	60
		ScenarioMCDMOnlyRetrievalRequest	30	60
		ScenarioListRequest	30	60
	Simulations	SimulationListRequest	30	60
		SimulationAvailabilityRequest	30	60
		SimulationStartRequest	30	60
SimulationStopRequest		30	60	
publishsubscribe	SubscriptionManagement	SubscriptionCreationRequest	30	60
		SubscriptionPauseRequest	30	60
		SubscriptionResumeRequest	30	60
		SubscriptionDeletionRequest	30	60
		SubscriptionListRequest	30	60
		SubscriptionRetrievalRequest	30	60

Service Group	Service	Service Request	Max Call Count Per Unit Of Time	Overload Window Size In Secs
		SubscriptionHistoryRequest	30	60
	Messaging	MessagePullRequest	30	60

Table 7.1. Overload thresholds

(9) In conclusion, NM implements a self-regulating mechanism such that:

a) Customers may consume the NM resources when they exist

b) When an overload condition is met, priority is given to calls issued by customers who are consuming fewer resources at that given time

(10) This mechanism also makes the NM services less sensitive to denial of service attacks.

(11) NM is however aware that some customers serve more end users than others, and therefore priorities might need to be adapted accordingly in the future. In addition, priorities may also depend on the service. These "refined" prioritisation requirements have not yet been assessed in the current NM release.

Chapter 8. Bandwith Protection

(1) The thresholds needed for the protection against very large replies are:

a) Defined for all the users.

b) The first threshold operates on a very small time window, typically 1 minute. This threshold value represents a PEAK overload.

c) The second threshold operates on a larger time window, typically 1 hours. This threshold value represents a BANDWITH overload.

(2) Protection against large replies:

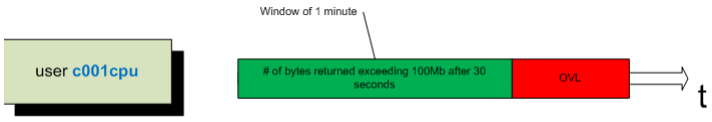

Figure 8.1. Threshold and Time Window for large replies protection

The PEAK threshold is 100Mb per 60 seconds. If a user has returned 100Mb in 30 seconds, he will get overload exceptions during the remaining 30 seconds of the time window. B2B request will resume normally at the beginning of the next 60 seconds window.

(3) Protection against a greedy usage of the bandwidth:

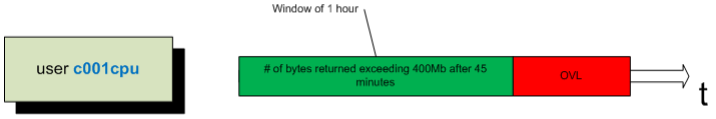

Figure 8.2. Threshold and Time Window for greedy usage of the bandwidth

The BANDWITH threshold is 400Mb per 1 hour. If a user has returned 400Mb in 45 minutes, he will get overload exceptions during the remaining 15 minutes of the time window. B2B request will resume normally at the beginning of the next 1 hour window.

Chapter 9. Exchange Model

9.1. Introduction

(1) Each service group defines its own exchange model: the set of requests, replies and data types that logically belong to the service group.

(2) Requests, replies and data types are defined and documented as classes, unions, enumerations and typedefs.

(3) Classes, unions and typedefs depend themselves on constrained strings and numbers.

(4) This chapter describes how exchange models are defined and how they are mapped to XML schemas (XSD).

9.2. Strings

9.2.1. Overview

(1) NM exchange models express string constraints in a formal manner.

(2) This section presents

a) The grammar that is used to describe string patterns

b) The mapping rules that are applied to translate this formal patterns into XSD patterns

9.2.2. Grammar

(1) A formal string validation language is defined: the EBNF (Extended Backus-Naur Form) of the grammar is defined below.

a) Spaces are not permitted as delimiters anywhere, the grammar considers them always as part of the rule.

b) As courtesy to the developer and not being part of the grammar, the multiplicity {1} can be omitted and the parser will automatically apply {1} if it cannot find a multiplicity of a `textItem`.

```
start                          = '' | listOrItems
listOrItems                    = orItem, { '|', orItem }
orItem                         = textItem, { textItem }
textItem                       = (groupOfTextDescriptors | textDescriptor), multiplicity
groupOfTextDescriptors         = '(', listOrItems, ')'
textDescriptor                 = ( charClass | charSequence | charLiteral )
charClass                      = 'ALPHA' | 'UALPHA' | 'LALPHA' | 'DIGIT' | 'HEXA' | 'TEXT' |
                                 'MULTILINE_TEXT'| 'WHITESPACE' | 'ANY'
charSequence                   = '"', charSequenceLiterals, { charSequenceLiterals }, '"'
charLiteral                    = charLiteralWithoutEscapees | allEscapedCharacters
multiplicity                   = '{', digit, [ ',', digit ], '}'
charSequenceLiterals           = charLiteralWoDoubleQuotesEscape | escapedDoubleQuote |
                                 escapedEscape
```

```
charLiteralWoDoubleQuotesEscape = charLiteralWithoutEscapees | '(' | ')' | '{' | '}' |
                                  '[' | ']' | '|'
charLiteralWithoutEscapees      = a-zA-Z0-9_ -+/*<>=,.;:?!'~@#$%^&
allEscapedCharacters            = escapedEscape | escapedDoubleQuote | '\(' | '\)' | '\{' |
                                  '\}' | '\[' | '])' | '\|'
escapedEscape                   = '\\'
escapedDoubleQuote              = '\"'
digit                           = '0'|'1'|'2'|'3'|'4'|'5'|'6'|'7'|'8'|'9'
```

(2) `charClass` represents a class of characters in the value to be validated. In the grammar, the `charClass` name is a terminal symbol.

(3) Available character classes are:

```
ALPHA:          all lower and upper case letters A to Z
UALPHA:         all upper case letters A to Z
LALPHA:         all lower case letters a to z
DIGIT:          all digits 0 to 9
HEXA:           all hexadecimal 0 to 9 A to F
TEXT:           the following characters: a-zA-Z0-9_ -+/*<>=,.;:?!'"~@#$%^&(){}[]\
MULTILINE_TEXT: the following characters: a-zA-Z0-9_ -+/*<>=,.;:?!'`"~@#$%^&(){}[]\ or any of new_line carr
WHITESPACE:     any of tab new_line carriage_return
ANY:            any character
```

9.2.3. XSD Mapping

(1) A `string` is mapped to:

```
<xs:simpleType name="typedef-name" >
  <xs:restriction base="xs:string"/>
    <xs:pattern value="{pattern}"
  </xs:restriction>
</xs:simpleType>
```

The `{pattern}` is evaluated according to the following character class mapping rules:

```
ALPHA:          [a-zA-Z]
UALPHA:         [A-Z]
LALPHA:         [a-z]
DIGIT:          [0-9]
HEXA:           [A-F0-9]
TEXT:           [a-zA-Z0-9_ \-\+\(&#47;)\*(&#60;)(&#62;)=,.;:?!'(&#34;)(&#126;)@#$%^(&#38;)
                 \(\)\{\}\[\]\\]
                all before mentioned characters
MULTILINE_TEXT: [a-zA-Z0-9_ \r\n\-\+\(&#47;)\*(&#60;)(&#62;)=,.;:?!'`(&#34;)(&#126;)@#$%^(&#38;)
                 \(\)\{\}\[\]\\]
                all before mentioned characters
WHITESPACE:     [ \t\r\n]
ANY:        .
```

9.2.4. Examples

(1) The `UALPHA{4}` pattern is mapped to the `<xs:pattern value="[A-Z]{4}"/>` XSD pattern.

(2) The `(ALPHA|DIGIT|.|_){1,500}` pattern is mapped to the `<xs:pattern value=""(([a-zA-Z]|[0-9])|(.))|(_)){1,500}""/>` XSD pattern.

9.3. Numbers

9.3.1. Overview

(1) NM exchange models supports the following concrete numbers: `byte, double, float, int, long, short`.

(2) Each number can be constrained to belong to a range [min, max[. By default, ranges are defined as right end open intervals. Each interval bound can however be included or excluded.

9.3.2. XSD Mapping

(1) A `{number}` is mapped to:

```
<xs:simpleType>
  <xs:restriction base="xs:{number}>">
    {number constraints}
  </xs:restriction>
</xs:simpleType>
```

(2) The `{number constraints}` are the usual XSD number bound constraints

```
<xs:minInclusive="{inclusive min}"/>
<xs:minExclusive="{exclusive min}"/>
<xs:maxInclusive="{inclusive max}"/>
<xs:maxExclusive="{exclusive max}"/>
```

9.4. Classes

9.4.1. Definition

(1) A class definition includes:

a) The name

b) The abstract property and/or inheritance relation

c) A textual description

d) The attribute definitions

e) Additional constraints

(2) An attribute definition includes:

a) The type

b) The name

c) The presence constraint (mandatory, optional or contextual)

d) A textual description

e) Additional type related constraints

9.4.2. XSD Mapping

(1) A class is mapped to a global XML schema complex type according to the following rules:

a) The schema type name is equal to the class name

b) If the class is abstract, the schema type is defined abstract

c) If the class extends a base class, the schema type extends the schema type that represents the base class

d) The list of attributes is mapped to a sequence of XML schema local elements which are defined according to the following rules:

i) The element name is equal to the attribute name

ii) The element multiplicity reflects the combination of the attribute presence and multiplicity:

A) 1..1 if the mapped attribute is mandatory and its type is not an array

B) 0..1 if the mapped attribute can be optional and its type is not a array

C) min..max if the mapped attribute is mandatory and its type is an array

D) 0..max if the mapped attribute can be optional and its type is an array

iii) The element type is evaluated according to the type of the mapped attribute

9.4.3. Examples

9.4.3.1. *Common Services* `<<class>> LastUpdate`

(1) The `LastUpdate` class is defined as follows:

a) Describes when and by whom an object has been last updated.

b) Attributes:

i) **`DateTimeSecond timestamp`** *(Mandatory)*
Time at which the object was created or last updated.

ii) **`UserId userId`** *(Mandatory)*
Id of the NOP user who created or last updated the object.

iii) **`AirNavigationUnitId airNavigationUnitId`** *(Optional)*
The ANU id associated to `userId`, if known by NM.

(2) The `LastUpdate` XSD Representation is:

```
<xs:complexType name="LastUpdate" abstract="false">
 <xs:sequence>
  <xs:element name="timestamp" type="common:DateTimeSecond" minOccurs="1" maxOccurs="1"/>
  <xs:element name="userId" type="common:UserId" minOccurs="1" maxOccurs="1"/>
  <xs:element name="airNavigationUnitId" type="common:AirNavigationUnitId" minOccurs="0"
                                                                           maxOccurs="1"/>
 </xs:sequence>
</xs:complexType>
```

9.4.3.2. *Airspace Services* `<<class>> ReferenceLocationAerodrome`

(1) The `ReferenceLocationAerodrome` class is defined as follows:

a) Reference to an aerodrome.

b) Inherits from: `ReferenceLocation`.

c) Attributes:

i) **`AerodromeICAOId id`** *(Mandatory)*
ICAO id of the referenced aerodrome.

(2) The `ReferenceLocationAerodrome` XSD Representation is:

```
<xs:complexType name="ReferenceLocationAerodrome" abstract="false">
 <xs:complexContent>
  <xs:extension base="airspace:ReferenceLocation">
   <xs:sequence>
    <xs:element name="id" type="airspace:AerodromeICAOId" minOccurs="1" maxOccurs="1"/>
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

9.4.3.3. *Flight Services* `<<class>> FlightOperationalLogEntry`

(1) The `FlightOperationalLogEntry` class is defined as follows:

a) Describes an entry in a flight operational log.

b) Attributes:

i) **`DateTimeSecond timestamp`** *(Optional)*
The date/time when the entry was recorded.

ii) **`FlightOperationalLogEntryType type`** *(Optional)*
The type of the operational log entry.

iii) **`int etfmsId`** *(Optional)*
The ETFMS unique id of the flight.
Constraint: Range: `[-2147483648,2147483647[`.

iv) **`IFPLId ifplId`** *(Optional)*
The IFPL id of the flight, if any in ETFMS.

v) **`string issuer`** *(Optional)*
The identification of the issuer of the message, if any.

vi) **`string message`** *(Optional)*
The detailed text of the message (if the entry is a detailed entry).

vii) **`string[] summaryFields`** *(Optional)*
The summary fields (if the entry is a summary entry).
Constraint: Size must be comprised between 0 and 2147483646.

(2) The `FlightOperationalLogEntry` XSD Representation is:

```
<xs:complexType name="FlightOperationalLogEntry" abstract="false">
 <xs:sequence>
  <xs:element name="timestamp" type="common:DateTimeSecond" minOccurs="0" maxOccurs="1"/>
  <xs:element name="type" type="flight:FlightOperationalLogEntryType" minOccurs="0"
                                                                      maxOccurs="1"/>
  <xs:element name="etfmsId" minOccurs="0" maxOccurs="1">
   <xs:simpleType>
    <xs:restriction base="xs:int"/>
   </xs:simpleType>
  </xs:element>
  <xs:element name="ifplId" type="flight:IFPLId" minOccurs="0" maxOccurs="1"/>
  <xs:element name="issuer" minOccurs="0" maxOccurs="1">
   <xs:simpleType>
    <xs:restriction base="xs:string"/>
   </xs:simpleType>
  </xs:element>
  <xs:element name="message" minOccurs="0" maxOccurs="1">
   <xs:simpleType>
    <xs:restriction base="xs:string"/>
   </xs:simpleType>
  </xs:element>
  <xs:element name="summaryFields" minOccurs="0" maxOccurs="unbounded">
   <xs:simpleType>
    <xs:restriction base="xs:string"/>
   </xs:simpleType>
  </xs:element>
 </xs:sequence>
</xs:complexType>
```

9.4.4. Special remark on the mapping of Classes to XSD Types

(1) A class is mapped to an XSD Complex Type Definition. Two aspects are considered to define the mapping strategy:

a) Inheritance relationships

b) Attributes

(2) The inheritance mapping has an equivalent in XSD: the complex type extension. An XSD complex type extends another XSD complex type, e.g.

```
<xs:complexType name="SpecialisedClass" abstract="false">
 <xs:complexContent>
  <xs:extension base="BaseClass">
   ...
  </xs:extension>
 </xs:complexContent>
</xs:complexType>
```

(3) To map the set of a class attributes, two options were considered:

a) Mapping to an XSD "sequence" model group:

i) Enforces the child element order

ii) Is compatible with XSD complex type extension

b) Mapping to an XSD "all" model group:

i) Does not enforce child element order

ii) Is not compatible with XSD complex type extension

(4) NM considered that losing the extension capability was of higher cost than the ordering constraint, and therefore adopted the XSD "sequence" mapping approach.

9.5. Unions

9.5.1. Definition

(1) A union is a polymorphic stereotype used in NM exchange models to model choices between different representation of the same concept, e.g. a flight plan can be defined as a structure or as a string that respects the ICAO format.

(2) A union definition includes:

a) The name

b) A textual description

c) The choice definitions

d) Additional constraints

(3) A choice definition includes:

a) The type

b) The name

c) The presence constraints (mandatory, optional or contextual)

d) A textual description

e) Additional type related constraints

9.5.2. XSD Mapping

(1) A union is mapped to a global XML schema complex type according to the following rules:

a) The schema type name is equal to the union name

b) The list of choices is mapped to a choice of XML schema local elements which are defined according to the following rules:

i) The element name is equal to the choice name

ii) The element multiplicity is 1..1

iii) The element type is evaluated according to the type of the mapped choice

9.5.3. Examples

9.5.3.1. *Flight Services* `<<union>> FlightPlanInput`

(1) The `FlightPlanInput` union is defined as follows:

a) Flight plan information expressed in either a structural or a textual format.

b) Choices:

i) **`FlightPlan structural`**
Object used when the flight plan data is input in a structured manner.

ii) **`string textual`**
FPL message text used when the flight plan data is input via a string.
Two message formats are accepted: ICAO and ADEXP.

(2) The `FlightPlanInput` XSD Representation is:

```
<xs:complexType name="FlightPlanInput">
 <xs:choice>
   <xs:element name="structural" type="flight:FlightPlan" minOccurs="1" maxOccurs="1"/>
   <xs:element name="textual" minOccurs="1" maxOccurs="1">
```

```
    <xs:simpleType>
      <xs:restriction base="xs:string" />
    </xs:simpleType>
  </xs:element>
 </xs:choice>
</xs:complexType>
```

9.6. Enumerations

9.6.1. Definition

(1) An enumeration definition includes

a) The name

b) The loose / strict property

c) The list of possible values

(2) NM distinguishes loose and strict enumerations. Loose enumerations accept values that are not strictly defined whereas strict enumerations do not. The purpose of loose enumerations is to support enumeration values that might appear in further NM releases without requiring an adaptation of the client applications.

(3) Enumerations are loose by default: the `<<enumeration>>` stereotype qualifies loose enumerations. If an enumeration is strict, it is qualified by the `<<strict enumeration>>` stereotype.

9.6.2. XSD Mapping

(1) An enumeration is mapped to a global XML schema simple type.

(2) The simple type definition depends on the nature of the enumeration: loose or strict.

(3) A strict `StrictEnum { SE1, SE2, SE3 }` enumeration is mapped to:

```
<xs:simpleType name="StrictEnum" >
  <xs:restriction base="xs:string">
    <xs:enumeration value="SE1"/>
    <xs:enumeration value="SE2"/>
    <xs:enumeration value="SE3"/>
  </xs:restriction>
</xs:simpleType>
```

(4) A loose `LooseEnum { LE1, LE2, LE3 }` enumeration is mapped to:

```
<xs:simpleType name="LooseEnum" >
  <xs:union>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="LE1"/>
```

```
        <xs:enumeration value="LE2"/>
        <xs:enumeration value="LE3"/>
      </xs:restriction>
    </xs:simpleType>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:pattern value="OTHER:[a-zA-Z][a-zA-Z0-9_]*"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:union>
</xs:simpleType>
```

9.7. Typedefs

9.7.1. Definition

(1) A typedef definition includes

a) A name

b) The underlying type (string or number)

9.7.2. XSD Mapping

(1) A typedef is mapped to a global XML schema simple type.

(2) The simple type definition depends on the nature of the typedef target type (`boolean`, `byte`, `double`, `float`, `int`, `long`, `short` or `string`).

(3) A `typedef<boolean>` is mapped to:

```
<xs:simpleType name="typedef-name" >
  <xs:restriction base="xs:boolean"/>
</xs:simpleType>
```

(4) A `typedef<{number}>` is mapped to:

```
<xs:simpleType name="typedef-name" >
  <xs:restriction base="xs:{number}>">
    {number constraints}
  </xs:restriction>
</xs:simpleType>
```

(5) A `typedef<string>` is mapped to:

```
<xs:simpleType name="typedef-name" >
  <xs:restriction base="xs:string"/>
    <xs:pattern value="{string pattern}"/>
  </xs:restriction>
</xs:simpleType>
```

9.7.3. Examples

9.7.3.1. *Airspace Services* <<typedef<string>>> AerodromeICAOId

(1) The `AerodromeICAOId` typedef is defined as follows:

a) ICAO id of an Aerodrome.

b) Pattern: `UALPHA{4}`.

(2) The `AerodromeICAOId` XSD Representation is:

```
  <xs:simpleType name="AerodromeICAOId">
    <xs:restriction base="xs:string">
      <xs:pattern value="[A-Z]{4}"/>
    </xs:restriction>
  </xs:simpleType>
```

9.8. Service Requests / Replies

9.8.1. Definition

(1) Requests / replies are defined as classes.

(2) A service request type is a concrete class that extends the abstract Request class.

(3) A service reply type is a concrete class that extends the abstract Reply class.

(4) By convention:

a) A request type name always ends with "Request"

b) The corresponding reply type name is always equal to the requested type name where "Request" is replaced with "Reply"

9.8.2. XSD Mapping

(1) Requests are mapped to the following pattern:

a) One XSD element

b) One complex type obtained according to the usual class XSD mapping

(2) Replies are mapped to the following pattern:

a) One XSD element

b) One reply complex type that:

i) Extends the *CommonServices* `Reply` type

ii) Wraps a *reply data* element that contains the reply specific data

c) One *reply data* complex type that is obtained by applying the usual class XSD mapping

9.8.3. Examples

9.8.3.1. *Flight Services* <<request>> FlightPlanValidationRequest

(1) The `FlightPlanValidationRequest` class is defined as follows:

a) Request to query the validation of an FPL according to the NM/IFPS validation rules.

b) The request provides the input flight plan information via a choice: either in string format or via a `FlightPlan` structure.

c) Inherits from: `Request`

d) Attributes:

i) **`FlightPlanInput flightPlan`** *(Mandatory)*
Flight plan to be validated.

(2) The `FlightPlanValidationRequest` XSD Representation is:

```
<!-- request element -->
<xs:element name="FlightPlanValidationRequest" type="flight:FlightPlanValidationRequest"/>

<!-- request complex type -->
<xs:complexType name="FlightPlanValidationRequest" abstract="false">
  <xs:complexContent>
    <xs:extension base="common:Request">
  <xs:sequence>
    <xs:element name="flightPlan" type="flight:FlightPlanInput" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

9.8.3.2. *Flight Services* <<reply>> FlightPlanValidationReply

(1) The `FlightPlanValidationReply` class is defined as follows:

a) Reply returned in response to FlightPlanValidationRequest.

b) Inherits from: `Reply`

c) Attributes:

i) **`IFPSError[] ifpsErrors`** *(Mandatory)*
Array of NM/IFPS errors in response to the flight plan validation.
Mandatory: the array is empty if there is no such error.

(2) The `FlightPlanValidationReply` XSD Representation is:

```
<!-- reply element -->
<xs:element name="FlightPlanValidationReply" type="flight:FlightPlanValidationReply"/>

<!-- reply complex type -->
<xs:complexType name="FlightPlanValidationReply" abstract="false">
 <xs:complexContent>
   <xs:extension base="common:Reply">
     <xs:sequence>
       <xs:element name="data" type="flight:FlightPlanValidationReplyData" minOccurs="0"
                                                                          maxOccurs="1"/>
     </xs:sequence>
   </xs:extension>
 </xs:complexContent>
</xs:complexType>

<!-- reply data complex type -->
<xs:complexType name="FlightPlanValidationReplyData" abstract="false">
 <xs:sequence>
   <xs:element name="ifpsErrors" type="flight:IFPSError" minOccurs="0" maxOccurs="unbounded"/>
 </xs:sequence>
</xs:complexType>
```

Chapter 10. Service Contract Organisation & Conventions

10.1. Overview

(1) As explained in Essentials - Purpose, NOP/B2B services are organised in service groups. The service group is logical: it groups logically the related services (port types) and the underlying exchange model (request/reply types and data types).

(2) For each NM service group NM provides:

a) One reference manual: `{ServiceGroup}.pdf`

b) One operational WSDL: `{ServiceGroup}_OPS_23.0.0.wsdl`

c) One pre-operational WSDL: `{ServiceGroup}_PREOPS_23.0.0.wsdl`

d) One XML schema (XSD): `{ServiceGroup}_23.0.0.xsd`

where `{ServiceGroup}` stands for the name of the service group following the CamelCase convention, e.g. `CommonServices`

10.2. Namespaces

(1) NM defines one namespace per service group. The service group namespace is defined as

```
eurocontrol/cfmu/b2b/{ServiceGroup}
```

(2) The target namespace of the operational WSDL, the pre-operational WSDL and the XML schema (XSD) is the service group namespace.

(3) Each time an NM namespace is indicated, it is meant to be relative to that root namespace. For example, when the `CommonServices` namespace is mentioned, the reader must understand `eurocontrol/cfmu/b2b/CommonServices`.

10.3. Naming Conventions

(1) By convention, all type names follow the CamelCase convention, including the service group name. It is sometimes necessary to derive a namespace name from a CamelCase type name: in order to do so, the CamelCase notation is kept in the namespace name, e.g. the `CommonServices` service group defines the `CommonServices` namespace name.

(2) Service request types are suffixed by "Request".

(3) Service reply types are suffixed by "Reply".

10.4. Service Group Reference Manual

(1) Each service group reference manual has:

a) A "Port Types" chapter

b) A "Data Types" chapter

(2) The "Port Types" chapter defines the services (port types) of the group. It is organised per service, and then per service request.

(3) Each service request section provides:

a) The SOAP operation name

b) The request definition

c) The reply definition

(4) The "Data Types" chapter defines all the types (classes, unions, enumerations and typedefs) that logically belong to the service group.

10.5. WSDL's and XSD

(1) In order to keep the XSD organisation as straightforward as possible, the following simplifying assumptions are followed:

a) All port types have a unique name in the scope of a NM version

b) All request and reply types have a unique name in the scope of a NM version

c) All data types have a unique name in the scope of a NM version

(2) NM provides one XSD file per service group: it contains all requests, replies and data types defined in the scope of the service group. The data XSD files are named:
`<service_group_name>_23.0.0.xsd`
The XSD namespace is `<service_group_name>`, e.g. `FlightServices` for the `FlightServices` service group.

(3) Clearly enough, the WSDL of a service group refers to the required XSD files consistently:

a) It always refers to the CommonServices XSD file

b) And to the XSD file associated to the service group requests, replies and data types

(4) Regarding SOAP message definitions, each request/reply pair is always supported via two messages:

a) The request message is named as the request type, e.g. if the request type name is `FlightPlanValidationRequest`, the request message definition in WSDL is:

```
<message name="FlightPlanValidationRequest">
    <part name="parameters" element="ns:FlightPlanValidationRequest"/>
</message>
```

b) The response message is named as the reply type, e.g. if the reply type name is `FlightPlanValidationReply`, the response message definition in WSDL is:

```
<message name="FlightPlanValidationReply">
    <part name="parameters" element="ns:FlightPlanValidationReply"/>
</message>
```

Chapter 11. PREOPS Testing

11.1. General

(1) NM defines "PREOPS" testing as the ability for the customer's development team to test its B2B client application before going OPS. PREOPS testing is supported via the PREOPS infrastructure, located at the PREOPS URLs indicated in ServiceLocation.

(2) PREOPS testing is achieved via the NM PREOPS infrastructure running in the NM premises, i.e. all customers use the same PREOPS web services deployed on the same PREOPS servers, using the same PREOPS data. Consequently, specific testing policies are described in the "PREOPS Testing" section of the various specific service group reference manuals, as these policies are not necessarily the same for testing e.g. airspace availability and flight filing services.

Chapter 12. Error and Warning Reporting

12.1. Principles

(1) All NOP/B2B services report errors in the same way some services report warnings, also in the same way.
The difference between an error and a warning is that an error prevents the transaction (read and write) from being committed, whereas a warning does not.

(2) The very existence of one or more errors resulting from a `Request` is expressed via the return ReplyStatus `reply.status`: if `reply.status` is `ReplyStatus.OK`, the `Request` was processed without error (but possibly with warnings).

(3) If the `reply.status` value is not `ReplyStatus.OK`:

a) The reply does not return any other data than the error data, i.e. no "business" data is returned

b) Unless the `reply.status` value is `ReplyStatus.OBJECT_OUTDATED`, in which case the latest version of the object to be updated is typically returned (see concrete `Replys` types)

(4) If the `reply.status` value is neither `ReplyStatus.OK` nor `ReplyStatus.INVALID_INPUT`, the `reply.status` value is self-explanatory and the returned `reply` has a null `inputValidationErrors` value; the `reply.iputValidationErrors` value is not null if and only if the `reply.status` value is `ReplyStatus.INVALID_INPUT`.

(5) Errors are typed using a three-dimensional path:

a) Service group: error first-level scope, namely the service group to which this error type belongs

b) Error category: within a service group, errors are further logically organised into categories

c) Error type: the final concrete error, within a service group and an error category

This way of expressing error types has been preferred to a simple error type id because it is in the customer's interest to take advantage of the service group/error category classification.

(6) The error type identification follows these conventions:

a) The `Error.group` attribute is of type `ServiceGroup`, where `ServiceGroup` enumerates (in Common) the existing service groups

b) The `Error.category` value is a `string<ErrorCategory>` where `ErrorCategory` belongs to the service group pointed out by `Error.group`. Each service group reference manual defines zero or one `ErrorCategory` enumeration that lists the error categories for that service group. It is because a port type in a service group may use an error category of another service group that the `Error.category` type is `string<ErrorCategory>`,

being conventionally understood that the `ErrorCategory` refers to the `ErrorCategory` enumeration of the `Error.group` service group.

c) The `Error.type` value is a `string<ErrorType>` where `ErrorType` is an enumeration defined in each service group that declares error types. Although the `ErrorType` is a real enumeration from the XSD perspectivem the `ServiceGroup`/`ErrorCategory`/`ErrorType` enumerations define together an "enumeration tree", where `ErrorTypes` can only be used within the context of a given `ErrorCategory`. The belonging of `ErrorType` enumerators to an `ErrorCategory` is expressed within the reference manuals (`ErrorType`), not in the XSD.
The Common `ErrorType` includes some general error types that can be used in many port types of potentially all service groups, like `MISSING_MANDATORY_ATTRIBUTE`.

(7) Warnings are represented using Error objects but are expressed within the `Reply.warnings` attribute.

12.2. Attribute Locations

(1) The Error class lists the attributes involved in the error via the `Set<string>` `Error.attributes`. If the error does not involve any attribute (e.g. for SLA errors), the `Error.attributes` set is left empty.

(2) The strings used in `Error.attributes` are either "attribute locations" or "attribute identifiers".

(3) "Attribute locations" are structured strings:

a) The attribute location is relative to the root `Request` object.

b) It uses attribute names separated by a "`:`" (colon) character.
For example the `FlightLevelRange` `flightLevelRange` of an `AbstractEAUPCDRRequest` (see Airspace Reference Manual) is indicated as "`flightLevelRange`".
The `flightLevelRange` `min` and `max` attributes are indicated as "`flightLevelRange:min`" and "`flightLevelRange:max`" respectively.

c) In order to locate an item in a collection/array attribute, the item in the collection is indicated with the attribute name, followed by the "#" (sharp) character, followed by the index of the object in the collection (the first collection item has index 0)
For example the 8th `UUID` in `UUID[]` `rsaUUIDs` of an `AbstractEAUPRSARequest` is indicated as "`rsaUUIDs#7`"; would the `UUID` class have an attribute called `someAttr`, this attribute would be located with "`rsaUUIDs#7:someAttr`".

(4) "Attribute identifiers" are also structured strings:

a) It is prefixed by the four characters "`@ID=`".

b) The value following the prefix is that of the XML ID of the entry producing the error as provided in the received message.

For example, within a received AIXM message, being the mandatory `gml:id` an attribute type XML ID, the timeslice with "`gml:id=ID_1`" would be identified with an `Error.attribute @ID=ID_1`.

12.3. Protocol Errors

12.3.1. HTTP 400 error - Bad Request

(1) The NM B2B web services make use of the standard HTTP 400 error [Bad Request] in any of the following cases:

- The request is for an unsupported NM release
- The request is not a well-formed XML
- The request is a well-formed XML but it is not valid with respect to the XSD (i.e. it does not conform to the type and attribute names defined in the XSD and documented in the reference manuals).
  Examples of causes for invalid XML documents are:
  a) Unexpected element or attribute
  b) Element order violation
  c) Incorrect primitive value
  d) Unexpected enum value

(2) To facilitate the B2B developer, in addition to the generic HTTP 400 error [Bad Request], a free text is returned indicating the cause of the error.
Below are some examples for each of the conditions listed above:

- Unsupported NM release

```
Error: 400 [Bad Request]
Version in root path '17.0.0' is not supported.; must be one of {18.0.0, 18.5.0, 19.0.0, 19.5.0}
```

- Malformed XML

```
Error: 400 [Bad Request]
NM 19.0.0 - Line:3 Col:49 - The end-tag for element type "sendTime" must end with a '>' delimiter.
```

```
Error: 400 [Bad Request]
NNM 19.0.0 - Line:50 Col:12 - The element type "registrationMark" must be terminated by the
matching end-tag "</registrationMark>".
```

- Invalid XML

```
Error: 400 [Bad Request]
NM 18.5.0 - Line:1 Col:234 - cvc-elt.1: Cannot find the declaration of element
'flight:FPLValidationRequest'.
```

-

```
Error: 400 [Bad Request]
NM 19.0.0 - Line:5 Col:35 - cvc-complex-type.2.4.a: Invalid content was found starting with
element 'aerodromeOfDeparture'. One of '{structural, textual}' is expected.
```

-

```
Error: 400 [Bad Request]
NM 18.5.0 - Line:23 Col:52 - cvc-datatype-valid.1.2.1: 'a1' is not a valid value for 'integer'.
```

-

```
Error: 400 [Bad Request]
NM 18.5.0 - Line:30 Col:44 - cvc-datatype-valid.1.2.3: 'VIFR' is not a valid value of union
type 'FlightRules'.
```

(3) **IMPORTANT**: B2B users should not use the B2B's reply to validate their requests. The textual error message is provided as a help for quickly identifying a problem. The users shall validate their requests against the provided XSD before sending them to the NOP B2B server.

12.3.2. HTTP 413 error - Request Entity Too Large

(1) If the server receives an XML message that is too large (500 KB in NM 16.0, where all input messages should be considerably smaller), the HTTP status 413 is returned.

DOCUMENT FINAL PAGE

To properly report any fault, or to propose a modification concerning the present document, please refer to:

- for faults, the Systems Incident Management Procedure, ref. STD-CM/PRO/SIMP
- for changes, the IT Change Management Process, ref. STD/ITSM/CHG