European Aviation Common PKI

Guidance to SWIM Providers and Consumers

Edition: 0.3
Edition date: 26-02-2024
Classification: Green

DOCUMENT CONTROL

Document Title	European Aviation Common PKI
Document Subtitle	Guidance to SWIM Providers and Consumers
Document Reference	This field is automatically updated
Edition Number	0.3
Edition Validity Date	26-02-2024
Classification	Green
Status	Draft
Author(s)	Abdel Youssouf
Contact Person(s)	Patrick Mana

APPROVAL TABLE

Authority	Date	Signature
Prepared by:		
Reviewed and endorsed by:		
Approved by:		

EDITION HISTORY

Edition No.	Validity Date	Author(s)	Reason
0.1	26-02-2024	Abdel Youssouf	Initial Draft After Review
0.2	05-04-2024	Abdel Youssouf	Incorporate Patrick Mana's comments
0.3	22-05-2024	Abdel Youssouf	Incorporate Dario DI Crescenzo's comments
0.3-1	27-05-2024	Abdel Youssouf	Add Guidance to SWIM Service Consumers

TABLE OF CONTENT

TABLE OF FIGURES

2 INTRODUCTION

2.1 General

Most of European civil aviation stakeholders have embarked with digitalisation era and are increasingly dependent on Information Systems and Information Technology. This new culture of electronic transaction and networking laid out to a greater threat than ever before of fraud, information eavesdropping, data theft, … for the aviation organizations and their customers. Hence, an effective, robust means of securing electronic communications and transactions has become increasingly important.

The European Aviation Common Public Key Infrastructure (EACP) is a service specifically tailored to aviation's specific needs in terms of cybersecurity and will support aviation stakeholders in complying with Common Pilot Project (CP1) Implementing Rule (EC N° 2021/116).

The CP1 mandates the creation of "a Common Public Key Infrastructure (PKI), which is used in signing, emitting and maintaining certificates and revocation lists used in inter-stakeholder communication for operational purposes, and for providing interoperability between eligible stakeholders for those who have their Local PKI".

EACP is aimed at providing digital certificates and interoperability services to all European aviation stakeholders, from air navigation service providers (ANSPs) to airspace users, airports, civil aviation authorities, military organisations, manufacturers, aviation meteorological service providers or businesses providing aviation services in any of EUROCONTROL's 41 Member States and 2 Comprehensive Agreement States.

EACP does not mandate the sole use of its digital certificates but offers the possibility to European Aviation stakeholders to use their local PKI (once assessed and qualified as eligible by the EACP) and to benefit from the interoperability feature of the EACP, such that European Aviation stakeholders can exchange secure messages between them without ambiguity and with confidence that the counterpart local PKI has reached an acceptable level of maturity.

2.2 Guidance to SWIM Providers and Consumers

The aim of this document is to provide a guidance to SWIM Service Providers and Consumers to support the use and integration of the European Aviation Common PKI into their systems, to raise their awareness about challenges and constraints they may encounter and to inform them about the potential changes that need to be made to their systems to use EACP.

This document is presented as follows:

- Section 3: Scope your business case. It is important to elaborate the case for which SWIM based information exchange will be elaborated. An important topic to take in consideration is the assurance level required to protect the information exchange. The assurance level for SWIM based Flight Plan may be different form the one elaborated for MET use case.
- Section 4: Conduct a security risk assessment and safety assessment. Every use case has its own threats and per consequent, need a specific set of security controls to mitigates the risks. If the same SWIM environment is used to enable

the information exchange, then the common denominator requiring the highest security controls and assurance level needs to be used.

- Section 5: The Use of EACP. This section addresses the Use of EACP Family 5.1.1 and Family 5.2.1. The aim is to raise awareness of using one or other Family and to highlight challenges that system administrators need to take into account when using one or other Family.
- Section 6: Prepare your servers. This section provides guidance in phased approach for both SWIM Service Providers and Consumers.

3 Scope your business case

The aim of this section is to help SWIM Providers and Consumers to scope a use case. It is assumed that the information and data that will be exchanged in the context of this use case will be under the SWIM framework.

3.1 Operational Scope

The first step ever is to provide an overview of the business environment and operational context for the use case, the following items may help defining the context:

- Provide AS-IS high level architecture and precise the back end zone (secure zone where the data is created, stored, processed, etc) and the front end zone (DMZ) with the connection (entry) points.
- Give an overview of the network architecture. Describe the segregation of the VLANs and explain protection zones by firewalls, etc.
- List all agents (actors) that act in the data flow (ingress or egress) and the role that they would play within it, including the data exchanged between SWIM Providers to SWIM Consumers.
- Get acquainted with the interfaces and protocols (e.g. AMQP 1.0) used to exchange the data.
- Define the environment used to exchange the information and data. If it is private environment (e.g. PENS/NewPENS), get the required credentials to access it.

3.2 Data and Information Security and Safety Scope

Once the operational scope has been completed, it is important to tackle the security aspect of the use case. the following items need to be in consideration.

- Security:
    - To keep the level of security commensurate to the degree of risk, and to minimize the impact of security incidents, it is recommended for an EACP PKI user to conduct a Security Risk Assessment for each system using EACP services and apply the appropriate controls to mitigate the risks. These controls can rely on EACP services.
    - Define the security objectives in term of Confidentiality, Integrity and Availability of the information and data being exchanged between SWIM Providers and SWIM Consumers.
    - Describe threats that may compromise the information and that could result in a breach. Threats will be compiled in the security risk assessment see section 4.1
    - It is expected that the result of the section 4.1 will provide a set of security controls to achieve the security objectives and to indicate the data protection level.
- Safety:
    - As for the security risk management, it is needed that each organisation conducts a Safety (Support) Assessment in case relevant safety related systems are affected by the use of EACP certificates or services.
    - Define the safety objectives of the information and data being exchanged between SWIM Providers and SWIM Consumers.

- Describe hazards that may affect safety. Hazards will be compiled in the safety assessment see section 4.2.
- It is expected that the result of the section 4.2 will provide a set of safety requirements to achieve the safety objectives.

3.3 Assurance Level Scope

The protection level derived from the security risk assessment would indicate the assurance level required to protect the data and information in transit. One way of accomplishing this is to use digital certificates.

The assurance level reflects the level of trust and confidence that a relying party can place on a digital certificate and refers to the strength of binding between the public key and the subject of the certificate, as well as about the various aspects of its lifecycle management.

4 Security Risk Assessment and Safety Assessment

Having scoped your business case, and fixed your security and safety objectives, there is a need to determine any potential threat, vulnerability and hazard, failure to your Business Case and then run a Security Risk Assessment and Safety (support) Assessment.

4.1 Security Risk Assessment

The Security Risk Assessment would determine a set of security controls that would achieve your security objectives to protect your data and information in both rest and transit and to mitigate the risks.

In addition, an important output of the security risk assessment is the identity assurance level (IAL). IAL will provide users and relying parties with the confidence that data and information to be protected are treated with the level of security commensurate with the degree of risks.

The IAL must be mapped to ones defined in the EACP Certificate Policy document, and therefore be expressed in terms of EACP functionalities.

Many standards already provide guidelines for conducting a security risk assessment, e.g.:

- ISO/IEC 27001,
- NIST Risk Management Framework (RMF),
- EUROCAE ED-201A and ED-205 (A).

The main European applicable regulation is:

- EU Regulation N. 2023/203.

4.2 Safety Assessment

As for the Security Risk Assessment, it is needed that you conduct a Safety (Support) Assessment in case the functionality of your system augmented with the use of EACP, and its associated services may be affected.

International and European regional regulations require to conduct a Safety (Support) Assessment in case a minor or major change to a safety related functional system is applied. That would be the case of e. g. a flight data management server or a surveillance system where a digital certificate provided by EACP will be implemented.

Main European applicable regulations are:

- EU Regulation N. 2017/373
- EASA - AMC3 ATS.OR.205(a)(2) Safety assessment and assurance of changes to the functional system.

5 The Use of EACP

SWIM Providers and Consumers must follow some steps to implement digital certificates in their systems as well as update their systems to ensure appropriate use and validation of the PKI services.

5.1 Who can Use EACP?

The following list specifies stakeholders that can benefit from EACP services:

- European Aviation Stakeholders: includes Air Navigation Service Providers, Airspace Users, Airport Operators, Civil Aviation Authorities or Military Organisations, aviation manufacturers of a EUROCONTROL Member State or Comprehensive Agreement State.
- European Aviation Extended Stakeholders: includes entities based in a EUROCONTROL Member State or Comprehensive Agreement State, which are not European Aviation Stakeholders, but which main business is aviation, or which is providing services to aviation.
- External Aviation Stakeholders, includes:
    - Air Navigation Service Provider, Airspace User, Airport Operator, Civil Aviation Authority, Military Organisation or aviation manufacturer, Entities, which main business is aviation and to Entities delivering product and services to EACP Participants,
    - and which are legally established in a Member State different than EUROCONTROL Member States or Comprehensive Agreement States.
- All EUROCONTROL service's users

5.2 Step 1: Installation of cryptographic libraries

The first step is to install the required cryptographic libraries to achieve the required level of protection of their data and information.

The cryptographic libraries depend on the Operating System (OS) in use, as well as on the IAL derived from the SRA:

- Case A: If the IAL has settle for the use of software based cryptographic keys:
    - For Windows OS, in most cases, Windows provides the native libraries based on the Microsoft Cryptographic API (CAPI) or Cryptographic Next Generation (CNG) to perform the key generation and other cryptographic operations. Other custom libraries such as OPENSSL, JAVA, .Net or Python can also perform the same cryptographic operations.
    - In Linux OS environment, the most popular libraries are OPENSSL, JAVA or Python, etc.
- Case B: If the IAL has required the use of hardware based cryptographic keys:

- In this case, SWIM Providers and Consumers must install the driver(s) associated with the hardware that will be in use and follow the vendor's instructions on how to initiate the hardware. Particular attention is given to the case where the hardware is a Hardware Security Module (HSM). In this specific case, there is a need to develop a proper Key Management Policy and Procedures.

5.3 Step 2: Follow the steps to acquire your digital certificate(s)

5.3.1 Understand your PKI practices!

SWIM Providers and Consumers need to understand if they are constrained to use their Local PKI (EACP – Family 5.2.1) or if they can use the common PKI (EACP – Family 5.1.1).

In both cases, the PKI providers must publish their policies and practice statements such that users and relying parties can be aware of the lifecycle management of the digital certificate of interest. In addition, PKI providers should guide users on how to purchase a particular type of a digital certificate following one or other protocol. Examples are purchasing TLS Server certificate based on ACME protocol, or a TLS client Certificate based on Http protocol, or on REST API, etc.

5.3.2 Which Digital Certificate to purchase?

The Yellow Profile has mandated to secure the communication channel between SWIM Providers and Consumers. This can be achieved by the unilateral server authentication. However, for obvious security reasons it is recommended to use mutual authentication.

One of the ways to achieve this requirement is to use TLS (Transport Layer Security) certificates. These digital certificates support the TLS protocols that are popular and widely used to secure communications over a network. TLS protocols have been standardized. The TLS version in use today is TLS version 1.2, and the latest version is version 1.3 (rfc 8446).

TLS protocols are based on a communication between clients and servers over a network. During the negotiation phase, an exchange of digital certificate occurs between a client and a server. Figure 1depicts a snapshot of a TLS protocol. Ample information can be found in the rfc 8446.

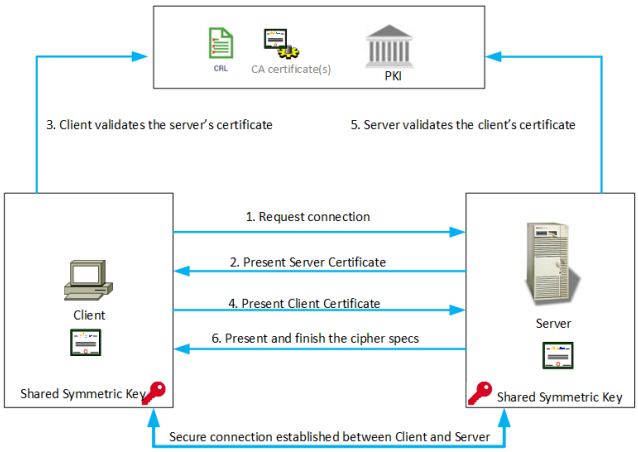

Figure 1: TLS protocol snapshot.

There are two brands of TLS certificates,

- TLS Server certificates: Those certificates must be installed by SWIM Providers. Figure 2 shows an example of TLS Server certificate.

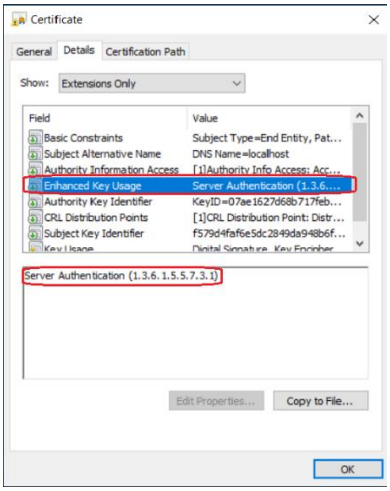

Figure 2: Example of a TLS Server certificate

- TLS Client certificates: Those certificates must be used by SWIM Consumers when connecting to SWIM Providers. Figure 3 shows an example of TLS Client certificate.

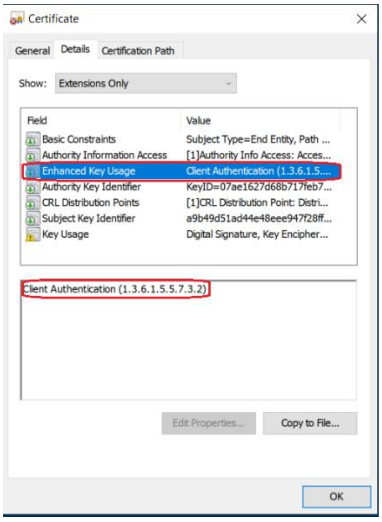

Figure 3: Example of a TLS Client certificate.

Good to know: Yellow Profile requires the use of TLS certificates but does not mandate signing SWIM messages for the integrity purpose. If your SRA expresses the need to digitally signing SWIM messages, then another digital certificate with the purpose of signing messages needs also to be purchased from EACP.

5.3.3 How to purchase Digital certificates?

5.3.3.1 Case of EACP – Family 5.1.1

At the time of writing this document, the EACP PKI provider has not been selected yet. In the following version of this document, and once the EACP provider will be known,

EUROCONTROL will provide ample information on how to purchase different digital certificates from common PKI (Family 5.1.1).

5.3.3.2 Case of EACP – Family 5.2.1

In case SWIM Providers and Consumers have decided to use their Local PKI, then they are conveyed to follow their local PKI process and procedures for purchasing/retrieving, using and maintaining their TLS digital certificates.

5.4 Step 3: Understanding the Validation Service

5.4.1 Why do we need to validate digital certificates?

Cryptographic operations are useless if the involved digital certificates are not validated.

In digital world, a digital certificate can be seen as a passport to identify and authenticate a citizen in real life. Digital certificates are then those authentication tokens distributed to different entities in the digital world, aiming to ultimately authenticate them whenever required.

Like in real life, passports are verified and validated at every trip by the control border, digital certificates need also to be verified and validated at the beginning of any transaction.

Validating digital certificates guarantees that entities involved in a transaction are genuine and that their digital certificates are generated by a trusted entity, valid in time, not revoked and used for the purpose for which they have been generated.

5.4.2 What to validate?

Therefore, the validation process can be based either on the basic validation process or on advanced validation process:

- The basic validation process is based on:
    - Certificate expiry check: Every digital certificate has a validity period. The validity period of a digital certificate is defined in rfc 5280 as being the time interval during which the CA warrants that it will maintain information about the status of the certificate. the validity period is made of two dates:
        - Notbefore: which indicate the date on which the validity period of the certificate begins.
        - NoAfter: which indicates the date on which the validity period of the certificate ends.
    - Certificate’s digital signature check: Every digital certificate contains information such as the identity on the certificate holder and the issuer, the validity period, the hashing algorithm, the public key, the "digital signature", and some extensions. Since this certificate is signed by its issuer’s CA, the value of this signature is stored in the field of "digital signature".
      The validation of a digital signature is summarised in the steps blow:
        - Extract the signature value from the certificate.

- Retrieve the Issuer's CA certificate either from the certificate itself (see section 5.5.2) or from trusted source such as CP/CPS.
- Use the CA public key to decrypt the signature value, this will result in a hash value.
- Compute the hash value of the certificate and compare it with the value obtained in last step, if the two values match, one can conclude that the digital signature is valid.

- Revocation check: Every CA must provide information about the status of the certificates it has issued. The status of the certificates can be provided either via the CRL file or via the OCSP responder. Both information can be obtained from the certificate itself (see section 5.5.2) or from trusted source such as CP/CPS.
  Validating a revocation means accessing the CRL file or the OCSP responder and ensuring that the certificate has not been declared as revoked by the CA.
- Path Validation: The trust is inherited from a top root CA. The path validation starts with the end entity certificate and proceeds through intermediate certificates up to the top root CA which is considered as trusted root certificate.
  Usually, the trusted root certificate can be loaded to a local trust named trust-store file. Intermediate CAs may or may not be loaded to that trust-store file. See section 5.5.1 and section 5.5.2 for more information.

Good to know: some APIs provide tools that perform Path validation and at the same time verify the digital signature and check the revocation at each layer of the certification hierarchy up to the Root CA.

Figure 4 gives an example of path validation of an end entity certificate.

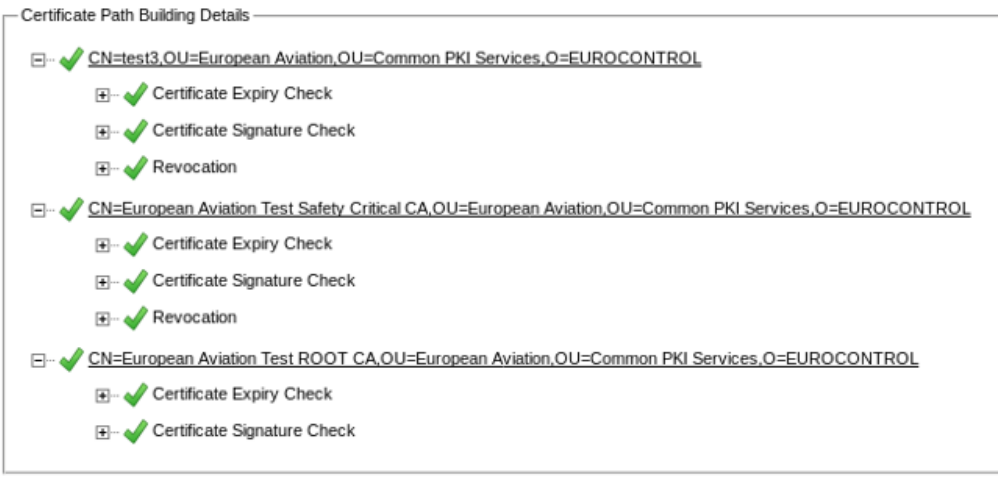

Figure 4: Example of Path Validation of an End Entity Certificate

- Key usage and Extended key usage are two certificate extensions that express the purpose of the use of digital certificate. See Figure 2 and Figure 3 for TLS certificates extended key usage.

- Advanced validation: Validation of digital certificate can be stretched beyond the basic one described above and verifies other attributes in a digital certificate such as:
    - Certificate Policy OID check
    - Name validation, policy mapping and other related checks

Good to know: Users and relying parties may delegate the validation themes to a specific server named "Server Certificate Validation Protocol (SCVP)" implementing the rfc 5055.

5.4.3 How to validate a digital certificate?

The essential of validating a digital certificate consist in parsing a digital certificate and looking to its specific attributes. As such:

1) To validate its validity period, the algorithm needs to read its notBefore and notAfter attributes and compare them to the current time. An example of such algorithm is:

Parse a TLS digital certificate and:

Get notBefore-date and notAfter-date from the certificate.

Get current-date from your server.

If current-date < notBefore-date then throw an exception (Certificate not yet valid exception)

Else If current-date > notAfter-date then throw an exception (Expired certificate exception)

Else … continue processing

2) To validate the certificate path up to a trust anchor:

1. Get the trust-Store-file
    - Extract the trust anchor from the trustStore-file
    - If you have loaded all intermediate certificates to the trust-Store file, then form a list of certificates contained in that file.
    - If not, then ensure that every certificate in path has AIA extension to indicate and access their issuer ‘s CA certificate.
2. Provide the Revocation information to the Path Validation engine,
    - Provide the cache of the CRL file, or
    - Ensure that every certificate in the path has the CDP (Certificate Distribution Point) extension to indicate the location of CRL file, or
    - If relying on the OCSP protocol, ensure that every certificate in path has AIA extension to indicate and access OCSP responder.
3. The validation engine proceeds with the validation by starting with end entity certificate and go above to next level or the hierarchy up the Root CA.
    - At every layer, the validation engine will validate a digital certificate by checking its digital signature and its revocation, if the signature is wrong and/or the certificate is revoked, then throw an exception, if not, continue processing.

PS: many libraries allow doing this path validation in one package.

3) Validate the purpose of using digital certificate:

Parse the TLS certificates:

The validation here will consist in checking the Enhanced Key usage extension for the TLS certificates.

- For the Server certificate:
  Parse the certificate and check if the ExtendKeyUsage-OID (object Identifier)= 1.3.6.1.5.5.7.3.1[1], If yes, continue processing, if not, throw an exception and drop the communication.
- For the Client certificate:
  Parse the certificate and check if the ExtendKeyUsage-OID (object Identifier)= 1.3.6.1.5.5.7.3.2, If yes, continue processing, if not, throw an exception and drop the communication.

If no exceptions are thrown, continue processing.

5.5 Step 4: Understanding your environment!

5.5.1 Environmental constraints

It is worth noting that EACP (with both Family 5.1.1 and Family 5.2.1) will be running on the internet network. However, some SWIM services may be running on private network (such as NM B2B running PENS/NewPENS) or on a mix mode.

When SWIM services are running on private network, then a direct connection to the PKI providers becomes problematic, per consequent provisioning SWIM Providers and Consumers with PKI services becomes challenging. In addition, some protocols to request digital certificates may become inappropriate.

Example of PKI services include an **automatic** provisioning with digital certificates (such as ACME protocol) as well as with the PKI validation features (CA certificates, CRL files or OCSP responses) required to accomplish the validation and verification tasks listed in section 5.3.

SWIM Providers and Consumers need to think about different tricks to provision their servers and clients with the required validation inputs such as caching the CRLs and the CA certificates and/or using OCSP stappling.

5.5.1.1 Case1 – Using EACP Family 5.1.1

When both SWIM Providers (servers) and Consumers (clients) are running on private network and are using the common PKI (ECAP family 5.1.1) which is running on public network, the problem of provisioning SWIM servers and clients with different PKI services becomes complex. The following considerations need to be taken into account:

1. Digital Certificate: When purchasing a digital certificate, some protocols may not be appropriate in a constrained environment. A typical example is ACME protocol that helps automating the certificate lifecycle management once properly configured. SWIM users relying on that protocol need then to accommodate an appropriate mean for purchasing and renewing their digital certificates accordingly.

[1] An object Identifier is a sequence of numbers that uniquely and permanently references an object. In X.509 digital certificate, each field is designated by an object identifier (see rfc 5280 for more details). In the Extended Key Usage extension, the flag "Server certificate" is identified by 1.3.6.1.5.5.7.3.1, and the flag "Client certificate" is identified by 1.3.6.1.5.5.7.3.2.

Good to know: Both SWIM servers and clients need to configure what is commonly known as "Trust Store File". This trustStorFile need to host at minimum the Root CA certificate, and depending on the operational constraint (see section 5.4.2), other subordinate CAs.

2. Access to EACP Repository: SWIM servers and clients will be restricted from accessing the EACP repository, and then from accessing CA certificates and CRL files. To overcome this situation, one may need to cache EACP CA hierarchy. EACP CA hierarchy is made at least of Root CA (considered to be as trust anchor), and an Issuing CA that has issued the end entity certificates. In some specific case, an Intermediate CA may also be implemented.

   The validity period of CAs lasts for few years. This means that once this file has been created, it will last for years provided that none of its CA certificate has been revoked. If one of the CAs has been revoked, System Administrators of SWIM servers and clients need to consult the EACP Certificate Practices Statement (CPS) to understand the steps they need to undertake. The minimum minimorum is to remove the revoked certificate and its associated issued certificates in the hierarchy and install the newly created ones.

3. SWIM servers and clients may need to cache the CRL files.
   In most cases, CRL files need to be available to the validation engine configured at the SWIM server's and client's side. Every CA in the hierarchy generates its associated CRL file.

Good to know: Both SWIM servers and clients need to create a file that holds the CRL files in a concatenated way.

Unlike the trustStoreFile, the CRL file needs to be updated frequently.

In general, and based on good practices, the validity period of a Root CRL file can last for 6 months, of an Intermediate CRL file for 3 months. But the validity period of the Issuing CRL depends on how the Issuing CA has configured its policies to generate its CRL file. In general, an Issuing CRL file is generated every week, or every 72 hours, or every 24 hours, or upon a revocation of an end entity certificate. System administrators are then invited to reach the EACP CPS document to be acquainted with the frequency of change of the Issuing CRL file.

Figure 5 shows an example of Using EACP Common backbone in a constrained environment.

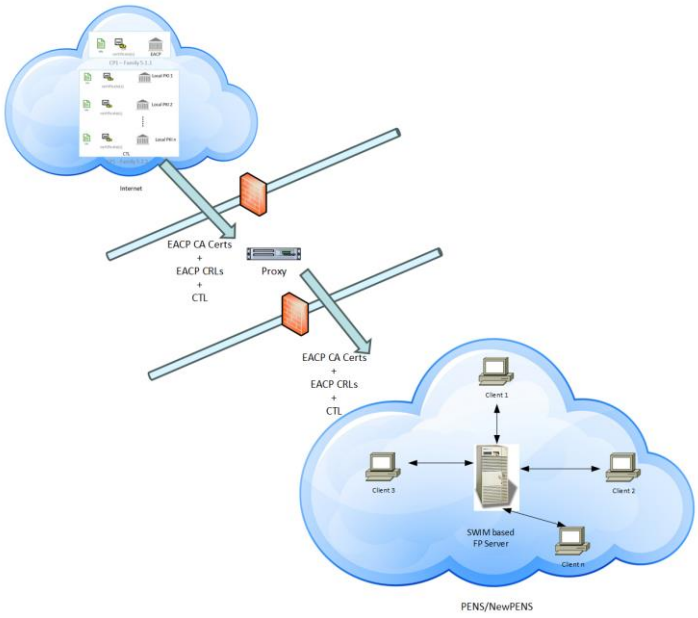

Figure 5: Example of Use of EACP - Family 5.1.1 - in constrained environment.

Creating a Proxy.

An example of overcoming the problem of provisioning the SWIM servers and clients with timely updated CA certificates and associated CRL files is to configure a proxy between the EACP servers, and the Private network see Figure 5. The proxy will have to handle the two critical files:

Configuring the trustStoreFile

As said before, this file does not need to be updated frequently. Therefore, System Administrators may proceed to manual load of the different CA certificates and create the trustStoreFile.

Script for updating CRL file.

A script should be running on the proxy to update the concatenated CRL file with the most up to date CRL file. As said before, focus is done on the Issuing CRL file. The frequency of running the script is obviously based on the EACP CRL policy. The script should forward the newly downloaded CRL file to the SWIM Servers and clients.

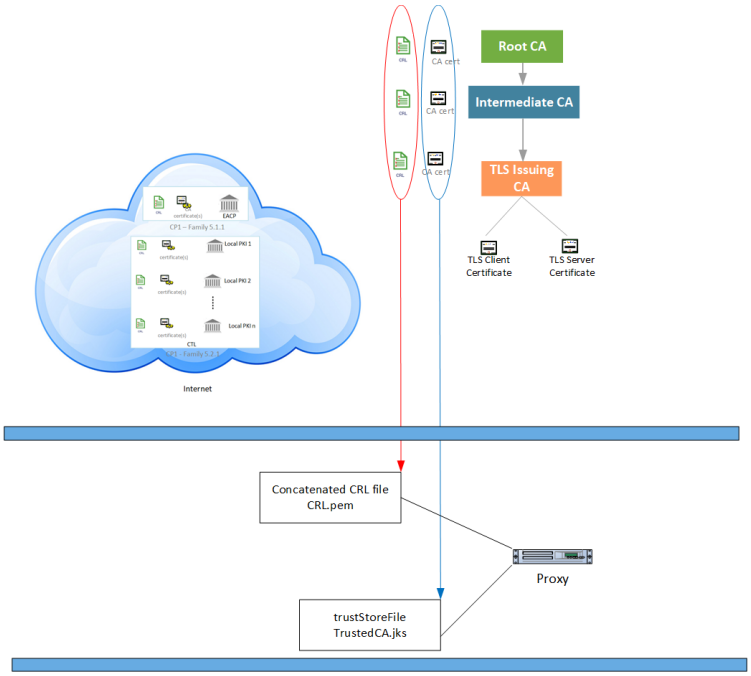

Figure 6: EACP Family 5.1.1 – Building Validation inputs.

5.5.1.2 Case of EACP - Family 5.2.1

If SWIM users are compelled to use EACP – family 5.2.1, then the first ever task to do is to procure the CTL file which is published on the EACP repository or the EUROCONTROL website. Once downloaded, SWIM users must validate the digital signature of the CTL file by following the instructions that will be given on the specified websites.

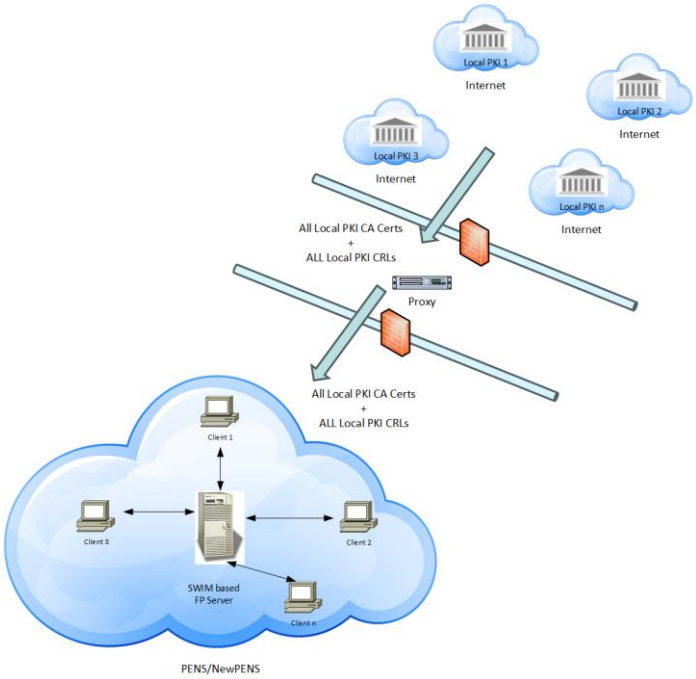

Figure 7: Example of Use of EACP - Family 5.2.1 - in constrained environment.

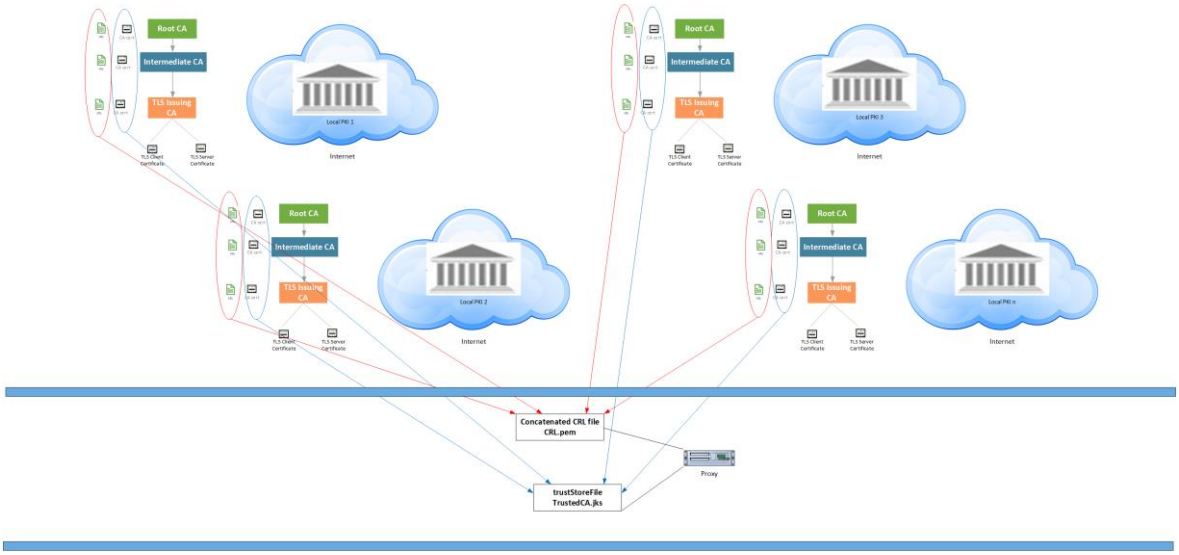

Figure 8: EACP Family 5.2.1 – Building Validation inputs.

5.5.2 Operational constraints

The RFC 5280 specifies that version 3 of digital certificates can be formatted with a set of fields named extensions, that gives ample information about the certificate, its issuer and its location, the purpose of using the certificate, the location of the CRL file and/or the OCSP URL to validate a certificate, etc. However,

this standard does not mandate the systematic inclusion of these features in a digital certificate.

As some of these features are important to the validation process, and in case they are missing, System Administrators need to find a way on how to provide the missing information to the validation engine.

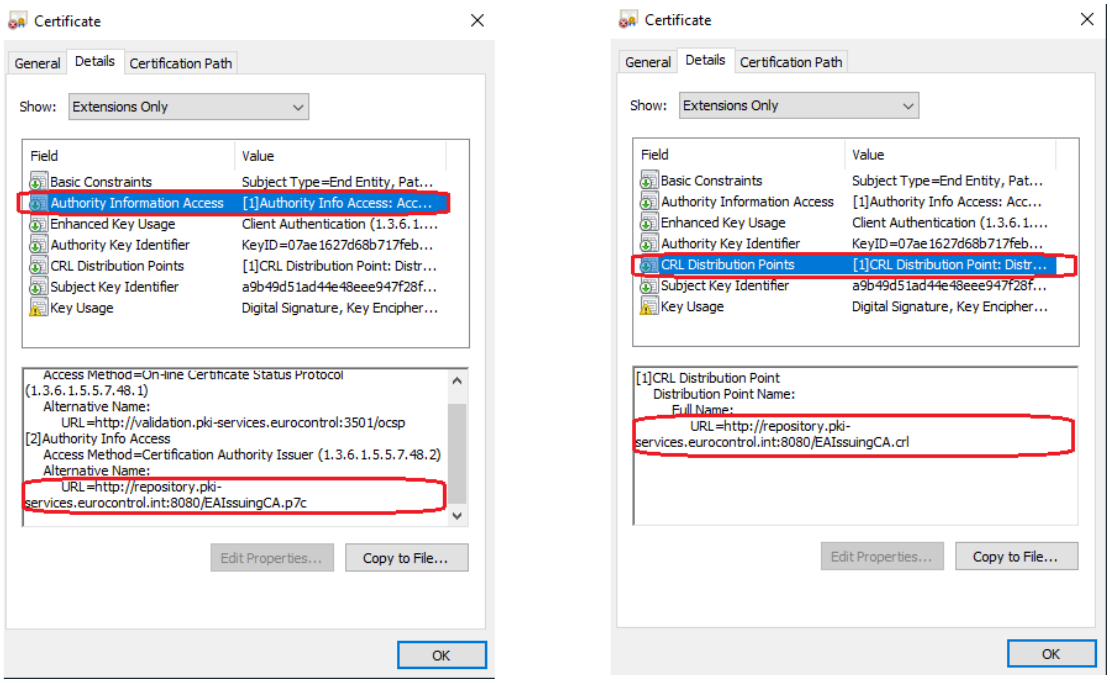

Figure 9: Digital Certificate showing URLs for downloading the Issuer certificate and the associated CRL file.

Good to know: System Administrators may seek the missing information in the PKI documentation (CP &/or CPS). By the missing information, we meant the different URLs pointing out to the different repositories where the CA certificates and CRLs are hosted.

System Administrators may need to hardcode the URLs in the Validation engine (some specification file) or use the caching tricks as described above.

6 Prepare your Servers

This section highlights the steps needed to prepare your environment to support the SWIM services. To illustrate the SWIM situation, we will use the following SWIM architecture to help nailing down this guidance:

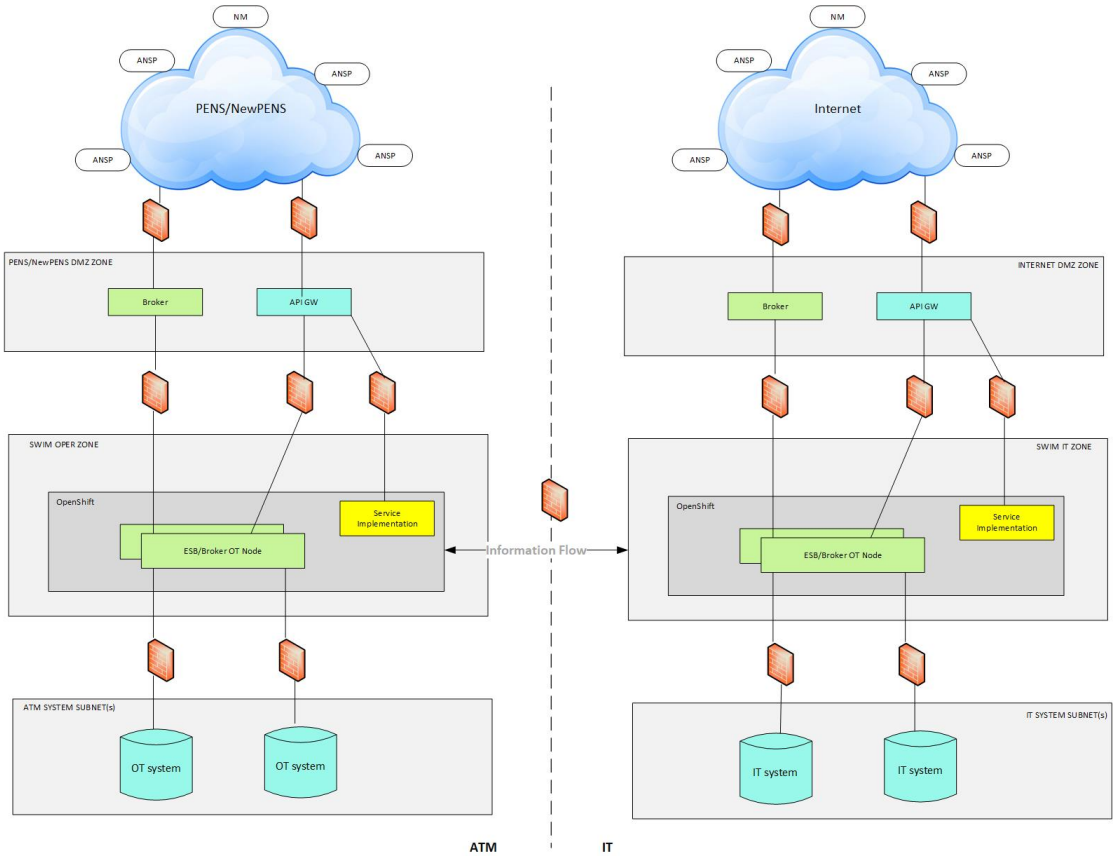

Figure 10: Example illustrating SWIM Architecture

In most cases, SWIM services can be provided with two technologies so called publish/subscribe method based on broker, or in request/response method based on some APIs. The two methods use the TLS certificates to secure the connections.

6.1 Guidance to SWIM Service Providers

This section highlights the steps required by SWIM Providers to prepare their servers accordingly.

1. Install the required cryptographic libraries accordingly see section 5.2

2. Develop the solution that will support your SWIM use case and ensure to have considered the validation module. Some solutions use open source. Typical examples of such open source solutions are Nginx, Httpd and Apache tomcat for supporting **Request/Response** mode and Apache Activemq for supporting **Publish / Subscribe** mode.

> **Good to know**: System Architects and developers must be aware that some open source solutions do not validate digital certificates systematically.

3. Purchase digital certificates, following the EACP provider instructions or your local PKI procedures. See section 5.3

4. Decide on how to provide validation inputs to the server. Validation inputs are made of CA certificates and CRL files: Prepare the script for caching the CRL or configure the server to access the validation inputs directly from the URLs indicated in the certificates. See section 5.5.1 for more information whether you are using EACP Family 5.1.1 or EACP Family 5.2.1.

5. Configure your server to use the validation inputs. Ensure to validate both TLS server and TLS client certificate. If you have opted for open source solution, ensure to use the latest stable solution, and to configure your server following the solution's instructions.

> **Good to know**: Some open source solutions have integrated the client certificates validation in their configuration, but do not validate TLS server certificates. This means that it is possible to start and maintain a service with an expired and/or revoked TLS server certificate. If such open sources are used in their solutions, developers must then develop a script that validate the server certificates also.

5.1. Figure 11 gives an illustrative example of configuring Tomcat with the validation inputs provided from EACP Family 5.1.1. The Trust store file is made of the branch of the CA hierarchy leading to the end entity certificate, while the Revocation file is made of a concatenated file containing all the CRL files of that branch.

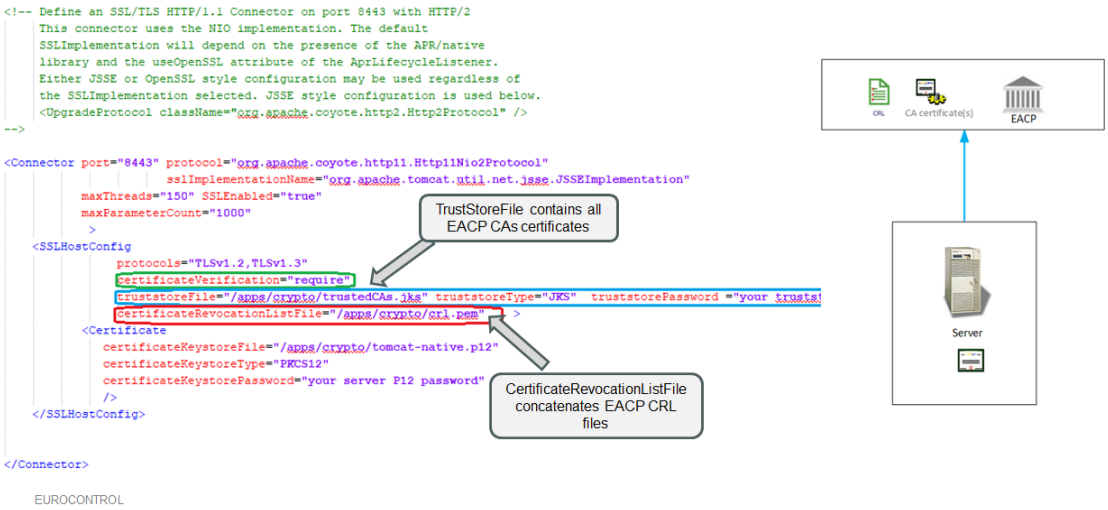

Figure 11: Example of Apache Tomcat with EACP Family 5.1.1

5.2. Figure 12 gives an example of configuring Apache Tomcat with EACP Family 5.2.1. System administrator must first download and validate the CTL integrity following the instructions that will be given to them, and then populate the trustStorefile with each local PKI CA contained in the CTL.

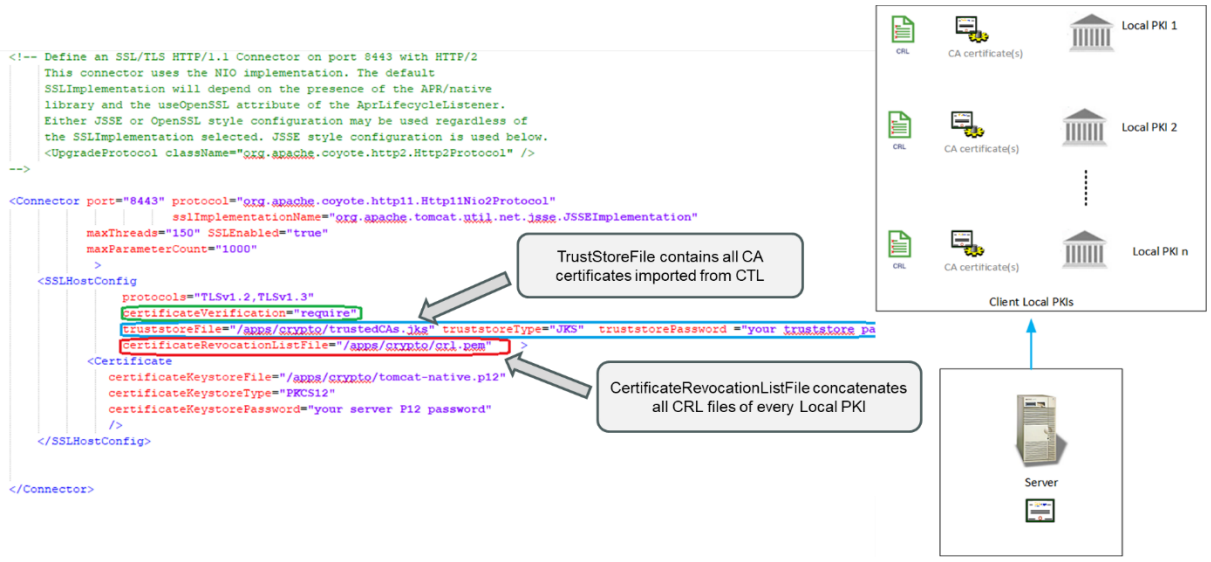

Figure 12: Example of Apache Tomcat with EACP Family 5.2.1

5.3. System architects and developers must add the validation module as described in section 5.4.2 in their solution.

> **Good to know**: Some open source solutions allow customising their native libraries for specific purpose (e.g. for validation). A typical example is Apache tomcat allowing to customise its connectors. Hence, developers may customise the existing connectors by adding a validation module.

6.2 Guidance to SWIM Service Consumers

SWIM Consumers use their TLS client certificates to connect to the SWIM service Providers. Mutual TLS connection consists in exchanging and validating the TLS certificates between Providers and Consumers. Since we have emphasized in section 6.1 the need to validate TLS client certificates and have also recommended to validate TLS server certificates, we will focus in this section on the need to validate the TLS server certificates at the SWIM Consumer's side. As outlined in section 6.1, there is need to make a distinction between TLS server certificate provided by the EACP Family 5.1.1 and the one provided by EACP Family 5.2.1, i.e. provided by one of the Local PKIs, member of the EACP CTL.

6.2.1 TLS Server certificate provided by Family 5.1.1

1. Install the required cryptographic libraries accordingly see section 5.2.

2. Prepare a trustStorefile that contains at least the EACP Root CA's branch of Server certificate,

Good to know: If SWIM Consumers do have environmental constraints (see section 5.5.1) or operational constraints (see section 5.5.2), then they may input the whole EACP CA branch of the Server certificate in the trustStoreFile.

3. Decide on how the server revocation will be checked? Via the CRL file or via the OCSP.
    a. If the revocation by CRL is selected, SWIM Consumers must ensure they can access the repository to get the CRL file (no environmental or operational constraints), if not then they can find a way on how to cache the CRL.
    b. If the OCSP is selected, then SWIM Consumers must ensure they can access the OCSP Responder.

Good to know: If SWIM Consumers do have environmental constraints (see section 5.5.1) or operational constraints (see section 5.5.2), then they may cache the CRL file, or use the OCSP stapling.

4. Develop a script that will check at every layer:
    a. the expiration of the certificate
    b. The revocation of the certificate
    c. The validation of the digital signature of the certificate
    d. The flag of the enhanced key usage of "Server authentication" of the server certificate. See Figure 2 for more details.

6.2.2 TLS Server certificate provided by a local PKI, member of EACP CTL

1. Install the required cryptographic libraries accordingly see section 5.2

2. Download the CTL and validate its digital signature first. Extract the concerned Local PKI's Root CA.

3. Gather the necessary information on how to access and store the validation inputs, i.e. CA certificates and associated CRL files or OCSP responders or OCSP stapling.

4. Prepare a trustStorefile that contain the Root CA's branch of Server certificate.

Good to know: **Good to know**: If SWIM Consumers do have environmental constraints (see section 5.5.1) or operational constraints (see section 5.5.2), then they may input the whole Local PKI CA branch of the Server certificate in the trustStoreFile.

5. Decide on how the server revocation will be checked? Via the CRL file or via the OCSP.
    a. If the revocation by CRL is selected, SWIM Consumers must ensure they can access the Local PKI repository to get the CRL file.
    b. If the OCSP is selected, then SWIM Consumers must ensure they can access the Local PKI OCSP Responder.

Good to know: If SWIM Consumers do have environmental constraints (see section 5.5.1) or operational constraints (see section 5.5.2), then they may cache the CRL file, or use the OCSP stapling.

6. Develop a script that will check at every layer:
    a. the expiration of the certificate
    b. The revocation of the certificate
    c. The validation of the digital signature of the certificate
    d. The flag of the enhanced key usage of "Server authentication" of the server certificate. See Figure 2 for more details.

SUPPORTING
EUROPEAN
AVIATION

www.eurocontrol.int