eIDAS Interoperability Architecture

Version 1.3

31 August 2019

eIDAS Technical Specifications

Document created by:

eIDAS eID Technical Subgroup

Document adopted by:

eIDAS Cooperation Network

DOCUMENT HISTORY

Version	Date	Adopted by	Short Description of Changes
1.0	26 January 2016	eIDAS Cooperation Network	/
1.1	16 December 2016	eIDAS Cooperation Network	/
1.2	27 September 2019	eIDAS Cooperation Network	• Use of non-notified eID schemes • Identification of Relying parties • Fixes in attributes profiles and cryptographic requirements • Improvements in metadata handling

NOTE: As specified in the *Opinion No. 3/2023 of the Cooperation Network on version 1.3 of the eIDAS Technical specifications* **version 1.3** of the eIDAS technical specifications is limited to the documents included in **version 1.2 plus eIDAS Common Attributes v1**

Table of Contents

1. Introduction

This document specifies the interoperability components of the eIDAS-Network, i.e. the components necessary to achieve interoperability of notified eID schemes according to the eIDAS Regulation [eIDAS]. This specifications are based on the requirements laid down in the Implementing Act [eIDAS IF].
Stakeholder of the eIDAS-Network are:

- the relying party:
    - requires authenticity/integrity of the received person identification data
    - in order to fulfill his data protection obligations, requires also confidentiality of the received personal identification data
- the citizen:
    - expects confidentiality of his person identification data
    - expects that the eIDAS-Network respects his privacy
- the operators of components of the eIDAS-Network
    - requirements derived from the requirements of the relying party and the citizens

To fulfill these requirements, and to provide accountability / liability mandated by the regulation, a chain of responsibility / trust is needed throughout the complete authentication process.
Therefore the framework for cross-border interoperability must provide:

- confidentiality of the person identification data;
- authenticity/integrity of the person identification data;
- secure identification/authentication of communication end-points.

The framework must not put requirements on the eID scheme or the systems of the MS where the relying party is established. It is assumed that the national systems provide adequate measures to provide confidentiality, authenticity/integrity and communication end-point identification for their systems.

Note: Relying party and citizen may also require availability of the eIDAS-Network. Requirements on operators concerning availability are out of scope of the Interoperability Framework and therefore not part of this technical specifications.

Note: The eIDAS Interoperability Framework is the European framework for trusted cross-border identification. Beyond that, Member States may reuse the Interoperability Framework also for sector-specific applications. Such sectors might make use of additional sector-specific functionalities, such as

sector-specific attributes or acceptance of additional non-notified eID means, that go beyond the scope of the eIDAS Regulation. Such optional functionalities do not impose any obligation to other Member States and may require bilateral agreements like on protocol elements or semantics.

Hence, these Technical Specifications contain means to integrate sector-specific functionalities, while the actual integration and usage are out of scope.

1.1. DEFINITIONS

The following terms are used throughout this document:

- *MS*: State covered by the eIDAS regulation, i.e. a Member State of the European Union and/or the European Economic Area
- *Person*: A natural or legal person, or a natural person representing a legal person.
- *Sending MS*: the MS whose eID scheme is used in the authentication process, and sending authenticated ID data to the receiving MS.
- *Receiving MS*: the MS where the relying party requesting an authentication of a person is established.
- *eIDAS-Node*: an operational entity involved in cross-border authentication of persons. A Node can have different roles, which are distinguished in this specification (eIDASConnector/eIDAS-Service, see below).
    - *eIDAS-Connector*: an eIDAS-Node requesting a cross-border authentication.
    - *eIDAS-Service*: an eIDAS-Node providing cross-border authentication.
        - *eIDAS-Proxy-Service*: an eIDAS-Service operated by the Sending MS and providing personal identification data.
        - *eIDAS-Middleware-Service*: an eIDAS-Service running Middleware provided by the Sending MS, operated by the Receiving MS and providing personal identification data.
- *Proxy based scheme*: a (notified) eID scheme which provides cross-border authentication via an eIDAS-Proxy-Service.
- *Middleware based scheme*: a (notified) eID scheme which provides cross-border authentication via eIDAS-Middleware-Services.
- *Middleware*: Software provided by a MS notifying a Middleware based scheme which is used by Receiving MSs to operate eIDAS-Middleware-Services.

1.2. KEY WORDS

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”,

“RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119]. The key word "CONDITIONAL" is to be interpreted as follows:

CONDITIONAL: The usage of an item is dependent on the usage of other items. It is therefore further qualified under which conditions the item is REQUIRED or RECOMMENDED.

1.3. ROBUSTNESS PRINCIPLE

Implementations according to this specification SHALL following the robustness principle, also known as Postel's Law:

“Implementations should follow a general principle of robustness:
be conservative in what you do, be liberal in what you accept from others.”

2. eIDAS-Nodes

Interoperability between different eID-schemes is achieved via defining the technical interfaces between *eIDAS-Connectors* and *eIDAS-Services*, collectively *eIDAS-Nodes*. The interfaces between the eIDAS-Connector and relying parties, and between the eIDAS-Services and the eID-scheme are part of the national system of the Receiving MS and the Sending MS, respectively, and therefore out of scope of this specification.

2.1. Sending MS

Sending MS can choose between two integration scenarios for their eID-scheme.

1. *Proxy*-based: The Sending MS operates an *eIDAS-Proxy-Service*, relaying authentication requests and authentication assertions between an eIDAS-Connector operated by the Receiving MS and the eID scheme of the Sending MS.
2. *Middleware*-based: In this scenario the Sending MS does not operate a Proxy for the purpose of authentication of persons to relying parties of other MS. The Sending MS provides a *Middleware* to other MS, which is operated by the operator(s) of the eIDConnector(s) of the Receiving MS.

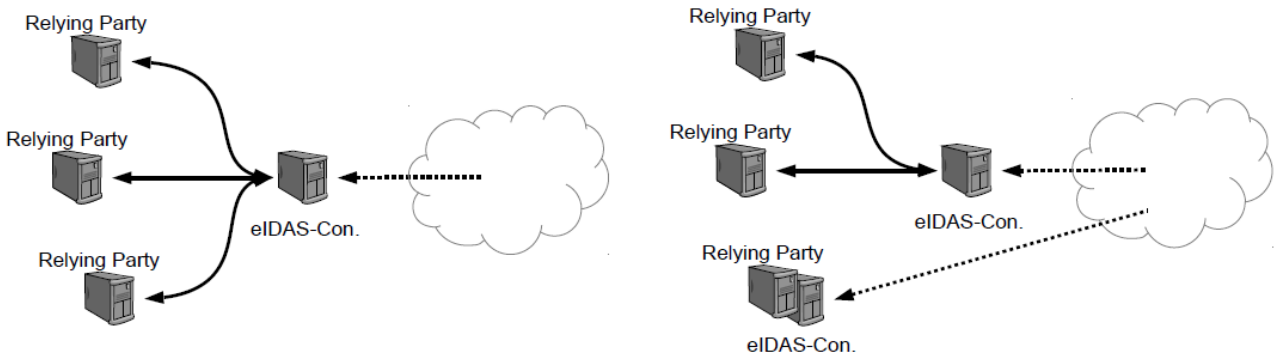
A MS notifying their eID scheme as a Middleware-based scheme MUST provide the necessary Middleware to Receiving MSs (see section 7.3).

2.2. Receiving MS

Each Receiving MS SHALL operate one or more *eIDAS-Connectors*. It is up to the Receiving MS to decide the national deployment of Connectors. Connectors need not to be operated by the MS itself, but can also be operated by public and/or private relying parties established in that MS.

In the following, MSs operating exactly one Connector are called *Centralized MSs*, while MSs operating several Connectors are called *Decentralized MSs*.

An eIDAS-Connector is operated together with eIDAS-Middleware-Services for communication with middleware-based eID schemes.



Centralized MS Decentralized MS

Figure 1: Centralized and decentralized deployment

The internal structure of the eIDAS-Connector is out of scope of this document. An eIDASConnector MAY provide additional services, (e.g. signature services), which are also out of scope of this specification.

Note: Centralized MS notifying their own eID-scheme as a Proxy-based scheme MAY operate their eIDAS-Connector and their eIDAS-Proxy-Service as an integrated deployment.

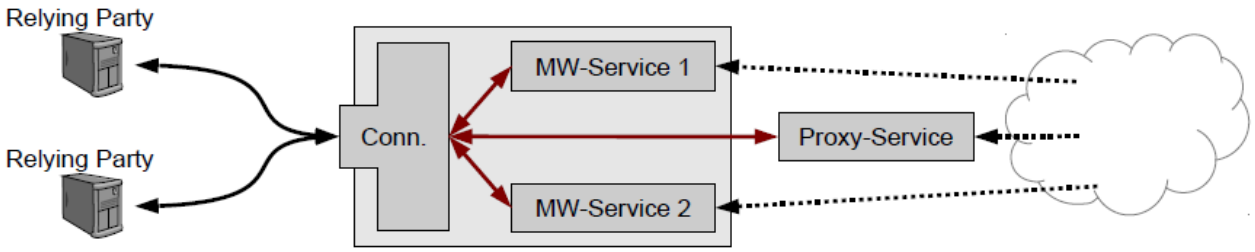


Figure 2: Components

Receiving MSs MUST ensure that personal identification data received via an eIDAS-Connector is processed according to applicable data protection legislation. This includes that data MUST NOT be forwarded to unidentified peers.

2.3. COMMUNICATION BETWEEN EIDAS-NODES

The eIDAS-Nodes SHALL use SAML (see section 3.2) for communication.

2.3.1.Attributes

eIDAS-Nodes MUST be able to handle the attributes of the Minimum Data Set [eIDAS IF] and MAY support additional attributes, i.e.

- eIDAS-Connectors MUST be able to request and process all mandatory and optional attributes of the Minimum Data Set as required by the corresponding relying parties of the Receiving MS.
- eIDAS-Services MUST be able to process and respond the requested attributes that are provided by the Minimum Data Set of the corresponding eID scheme of the Sending MS.
- eIDAS-Nodes MAY support additional attributes beyond the scope of the Minimum Data Set.
- eIDAS-Nodes MUST ignore attributes in a received request or response SAML message that are not supported.

Note: Specification and mutual recognition of additional sector-specific attributes is out of scope of the eIDAS Technical Specifications.

2.3.2.Levels of Assurance

eIDAS-Nodes MUST be able to process the eIDAS Level of Assurance. To support reuse of the eIDAS Interoperability Framework for sector-specific applications, eIDAS-Nodes MAY support non-notified eID schemes. The corresponding process flows are specified in chapter 5 of this specification.

Note: In cases where recognition of eID scheme not required by the eIDAS Regulation, recognition arrangements are out of scope of the eIDAS Technical Specifications.

2.4. IDENTIFICATION OF EIDAS-NODES

To provide an uninterrupted chain of trust for authentications, as well as an uninterrupted chain of responsibility for integrity/authenticity and confidentiality for personal identification data, eIDASNodes MUST be securely identified before transmitting data to them/accepting data from them.

2.4.1.Proxy based Schemes

Certificates for SAML signing and encryption of messages between eIDAS-Connector and eIDASProxy-Services are exchanged via signed SAML Metadata, see section 6.

2.4.2.Middleware based Schemes

Certificates for SAML signing and encryption of messages between eIDAS-Connector and eIDASMiddleware-Services are exchanged directly between the entities, since there is a one-to-one correspondence between Connector and Middleware-Service. To facilitate exchange, Middleware-Services SHALL provide (unsigned) SAML Metadata containing the certificate of the Service, see section 6.

2.5. IDENTIFICATION OF RELYING PARTIES

To enhance trust and transparency for users of the eIDAS-Network, it is advisable to enable identification of relying parties via the eIDAS nodes.

According to article 7.f of the eIDAS Regulation, a Sending MS may define terms of access for non-public sector relying parties so that there MAY be the need to identify this type of relying parties.

Identification of relying parties is realised through a RequesterID in the SAML request message (see [eIDAS SAML] for specification of the SAML field).

If an eIDAS Service requires the RequesterID for identification of non-public relying parties, it SHALL indicate this via a flag in the SAML metadata (see [eIDAS SAML]).

- eIDAS-Connectors MAY use this information from the flag in the SAML Metadata for an automatic processing (i.e., checking the necessity to identify non-public sector relying parties).
- Otherwise, eIDAS Connectors SHALL ignore this flag in the Metadata. In this case, implementers of eIDAS Connectors SHALL consult the separate specification document for non-public sector relying parties to make sure that identification requirements are fulfilled by other means.

3. Interfaces

An operator of an eIDAS-Connector operates one instance of the eIDAS-Connector, and one eIDAS-Middleware-Service for each type of (notified) middleware based eID schemes.

Note: MSs notifying middleware based eID schemes based on the same technology MAY share a middleware, i.e. only one instance of the middleware is necessary for those eID schemes.

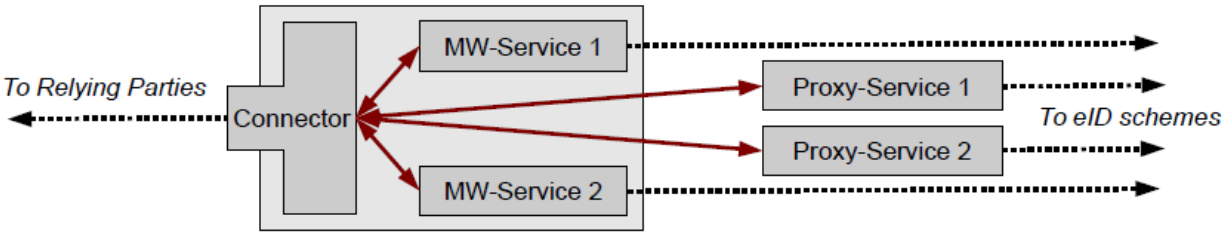


Figure 3: Interfaces

The eIDAS-Connector provides the following interfaces:

3.1. Interface between eIDAS-Connector and Relying Party

This interface is up to the Receiving MS and out of scope of this specification.

Note: This interface must be secure enough to meet the requirements of the maximum level of assurance transmitted by the Connector, in order to ensure the authenticity and confidentiality of the transmitted personal identification data.

3.2. Interface between eIDAS-Connector and eIDAS-Service

This section describes the interface between eIDAS-Connector and eIDAS-Service. This covers eIDAS-Proxy-Services as well as eIDAS-Middleware-Services.

The eIDAS-Nodes SHALL use SAML, including error handling, for communication, as specified in [SAML-Core], as profiled in [eIDAS SAML].

Messages MUST NOT be sent to eIDAS-Nodes not identified and MUST NOT be accepted from eIDAS-Nodes not identified (see section 2.4). The necessary information for communication, e.g. URLs, certificates for encryption/signature and information about the capabilities of nodes are contained in SAML Metadata objects of the Nodes (see section 6.3). Metadata objects of peers can be retrieved during the authentication process or ahead of time (see section 6.4). Metadata objects MUST NOT be used if not successfully (explicitly or implicitly) verified (see section 6.5).

For all SAML communication, Transport Layer Security (see section 7.4 and [eIDAS Crypto]) SHALL be enforced by all Nodes.

3.2.1.Request Messages

SAML Request Messages MUST be signed (see [SAML-Core]). The requirements from [eIDAS Crypto] apply.

The Request Message Format is defined in [eIDAS SAML]. The data format of the eIDAS attributes is defined in [eIDAS Attributes].

3.2.1.1. Binding

HTTP Redirect binding or HTTP POST binding (see [SAML-Binding]) SHALL be used for transmitting SAML Request messages. Nodes SHALL support both bindings for Request messages.

It is RECOMMENDED for the requesting Node to use HTTP Redirect binding if the (signed) request is short enough to be processed via a redirect[1].

3.2.1.2. Verification

Each eIDAS-Service MUST verify the integrity/authenticity of a SAML Request message before processing the request.

This compromises the following steps:

1. Extract the signature certificate of the Connector from the verified SAML Metadata object of the Connector (see sections 6.4/6.5)
2. Verify the signature of the SAML Request message.

Note: Steps 1 MAY be performed ahead of time, if the SAML Metadata object is validated and imported decoupled from the receiving of the SAML message.

Request messages which cannot be verified via this procedure MUST be rejected.

3.2.2.Response Messages

Response Messages MUST be signed (see [SAML-Core]). Additionally, an Assertion contained in the Response Message MAY be signed.

SAML Assertions MUST be encrypted; the encryption certificate SHALL be retrieved from the verified SAML Metadata object of the requesting Connector.

---

[1] This recommendation is aimed at reducing latency (redirect processing is usually faster than POST-processing) and enhancing usability in environments where java-script is not available, e.g. corporate networks.

Assuming a successful response, the Response message MUST contain exactly one EncryptedAssertion-element. The encrypted Assertion MUST contain exactly one AuthnStatement-element and one AttributeStatement-element.

The requirements from [eIDAS Crypto] apply.

The Response Message Format is defined in [eIDAS SAML].

3.2.2.1. Binding

HTTP POST binding (see [SAML-Binding]) SHALL be used for transmitting SAML Response messages. The Response MUST NOT be transmitted to a URL not contained as AssertionConsumerService in the metadata of the Connector.

3.2.2.2. Verification

Each eIDAS-Connector MUST verify the authenticity a SAML Response message before processing the included assertion.

This comprises the following steps:

1. Extract the signature certificate of the Connector from the verified SAML Metadata object of the Connector (see sections 6.4/6.5).
2. Verify the signature of the SAML Response message.
3. If the SAML Assertion is signed, its signature MAY be verified.

Assertion messages which cannot be verified via this procedure MUST be rejected.

Unsolicited Response Messages MUST NOT be accepted.

3.3. INTERFACE BETWEEN EIDAS-SERVICE AND EID SCHEME

This interface is up to the Sending MS and out of scope of this specification.

4. MS Selection

The eIDAS-Connector SHALL offer the user the possibility to select the MS, whose (notified)eID scheme is to be used for authentication, if the MS was not already pre-selected by the requesting relying party.

The eIDAS-Connector SHOULD only offer those MSs which are capable of fulfilling the request of the relying party (minimum Level of Assurance, type of Data Set to be authenticated), if these information are available to the Connector.

5. Process Flow

This section describes the process flow to authenticate a person, enrolled in the eID-scheme of the Sending MS, to a relying party established in the Receiving MS.

The request MAY contain an identifier identifying the MS, whose eID scheme is to be used for the authentication, if this is already known to the relying party.

1. The process is started by the relying party, which sends an authentication request to the eIDAS-Connector responsible for it. The eIDAS-Connector can be directly attached to the relying party (Decentralized MS) or operated by a separate entity (Centralized MS).

   The request MAY contain an identifier identifying the MS, whose eID scheme is to be used for the authentication, if this is already known to the relying party.

2. The eIDAS-Connector SHALL request the MS, whose eID scheme is to be used for the authentication from the user, if this information was not already contained in the request of the relying party (see section 4).

3. The eIDAS-Connector SHALL send a SAML-Request to the eIDAS-Service corresponding to the selected MS (see section 3.2.1).

   - The request MUST include the type of the relying party and SHOULD include a RequesterID identifying the relying party (irrespective of the type of the relying party).
   - If the requesting relying party is a non-public entity and the eIDAS-Service requires the identification of non-public sector relying parties, the request MUST include the RequesterID (see section 2.5).
   - The request MUST include exactly one RequestedAuthContext element specifying the required minimum eIDAS LoA (see [eIDAS SAML]).

     If non-notified schemes are supported by the eIDAS-Connector, the RequestedAuthContext element MAY contain a reference to one or more accepted authentication context classes. The authentication context classes may be sector-specific. (Note that for non-notified schemes exact matching is used as defined in the Message Format specification).

4. The eIDAS-Service MUST verify the authenticity of the Request (see section 3.2.1.2).

   - If the eIDAS-Service serves several eID schemes, the Service SHOULD provide a scheme selection interface for the user.
   - If the requesting relying party is a non-public entity, the eIDAS-Service MAY reject the Request if the terms of access of the eID scheme (see Article 7(f) of [eIDAS]) are not fulfilled.

- If the requested (or higher) Level of Assurance cannot be fulfilled by the eIDAS-Service, the Request MUST be rejected.
    - For non-notified schemes exact matching applies. If the requested sector specific authentication context cannot be fulfilled by the eIDAS-Service, the Request MUST be rejected.

5. The eIDAS-Service SHALL perform the authentication of the person according to the selected eID scheme at least on the requested Level of Assurance.
6. The eIDAS-Service SHALL send a SAML Response to the requesting eIDAS-Connector containing an encrypted SAML Assertion (see section 3.2.2). The SAML response SHALL include information on the used Level of Assurance (or, if supported, the alternative authentication context class) of the identification means used for authentication.
7. The eIDAS-Connector SHALL verify the authenticity (see section 3.2.2.2) of the received SAML Response message and decrypt the Assertion. The Connector MUST verify that the Level of Assurance (or, if supported, the sector-specific authentication context class) indicated in the Assertion matches or exceeds the requested Level of Assurance, and send the received authenticated person identification data to the requesting relying party.
    - For non-notified authentication context classes exact matching applies as defined in the Message Format specification. The authentication context classes may be sector specific.
8. If any of the checks (e.g. signature verification, Level of Assurance matching) fails, the procedure MUST be aborted. Error handling SHALL follow the SAML specification (see [SAML-Core]).

Note: Definition of additional context classes for non-notified eID schemes is sector specific and out of scope of this specification.

6. Metadata exchange

To provide an uninterrupted chain of trust for authentications, as well as an uninterrupted chain of responsibility for integrity/authenticity and confidentiality for personal identification data, Nodes must be securely identified. The Architecture defines two communication relationships:

1. Communication between eIDAS Connectors and Proxy-Services.
2. Communication between eIDAS Connectors and Middleware-Services.

Metadata exchange is based on the following principles:

- The trust anchors for all Nodes are the MSs. No central trust anchor (e.g. via the Commission) is provided. Trust anchors are exchanged bilaterally between MSs (see section 6.2).
- Metadata are distributed in the form of SAML Metadata [SAML-Meta]. Metadata objects of Connectors and Proxy-Services are signed by the trust anchor or by another entity (e.g. the operator of the Node) authorized via a certificate chain starting from the trust anchor. (see section 6.3). Metadata objects of Middleware-Services are unsigned.

SAML Metadata SHALL be instantiated per Node.

This model allows different national deployment scenarios (arrows denote signatures):

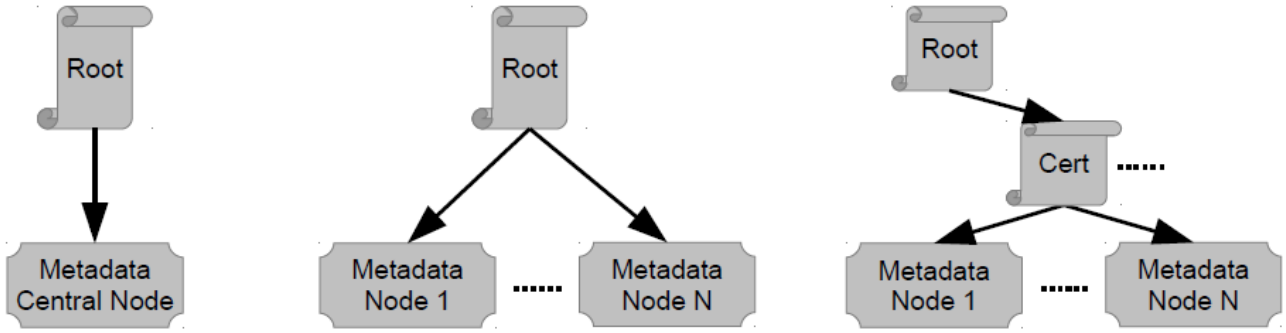


Figure 4: Metadata trust management

Note: This allows to separate the trust anchor from the actual SAML end points (Nodes). This implies that the entity providing the trust (and holding the "root key") is not necessarily the same providing the SAML metadata, i.e. the Node operator. Since the trust anchors are exchanged bilaterally, and all trust (including Node certificates) is derived from these anchors, it is not necessary to use certificates from public CAs for these certificates.

6.1. PRE-EXCHANGED DATA

Some data MUST be made available by each MS and exchanged bilaterally in order to convey trust between MSs:

- the trust anchor of each MS (see section 6.2);
- for Proxy-based notification the URL of the metadata of the corresponding Proxy-Service;
- for MSs deploying a single Connector, the URL of the metadata of that Connector, or, for MSs deploying several Connectors, the URL of the list of metadata locations (see section 6.4.2).

6.2. TRUST ANCHOR

All MS MUST bilaterally exchange trust anchors in the form of certificates, each certifying a signing key held by the MS (a“Root”). Such signing key can either be used:

- to directly sign SAML metadata objects, or
- as root certificate of a PKI used to sign SAML metadata objects.

Certificates of root keys and subordinate certificates SHALL follow [RFC5280].

MSs MUST ensure that all SAML Metadata objects signed directly or indirectly under such Root describe valid eIDAS-Nodes established in that MS. This implies:

- The described node must be a sending node (eIDAS-Service) of the (notified)eID scheme or a receiving node (eIDAS-Connector) entitled to receive authenticated personal data according to [eIDAS].
- No longer entitled (e.g. due to compromise) nodes must be revoked. This can be for example done by revoking the metadata signing certificate (which revokes all nodes whose metadata are signed with that certificate) or by using short lifetimes of metadata and not resigning them in case of revocation (see also section 6.4.1).

In addition, a bilaterally exchanged Trust Anchor can be used to sign an optional MetadataServiceList (see appendix A of the Message Format specification), which may contain information about national metadata resources and how to validate them.

6.3. SAML METADATA

Each eIDAS-Connector and each eIDAS-Service MUST provide metadata about the Connector / Service in the form of SAML Metadata (see [SAML-Meta]).

The SAML Metadata objects of Connectors and Proxy-Services MUST be signed and MUST include a certificate chain starting at a trust anchor (see section 6.2) and terminating with a certificate certifying the key used to

sign the Metadata object. Requirements for SAML from [eIDAS Crypto] apply. Metadata objects of Middleware-Services are exchanged directly with the corresponding Connector and therefore do not need to be signed.

The SAML Metadata Interoperability Profile Version MUST be followed (see [SAML-MetaIOP]). Certificates MUST be encapsulated in X509Certificate-elements. The Metadata Format for Connectors and Services is defined in [eIDAS SAML].

6.4. METADATA LOCATION

The SAML Metadata of Connectors and Proxy-Services MUST be publicly available under a HTTPS URL. The SAML Metadata of Middleware-Services SHALL be exported as a file.

SAML Requests and Responses originating at a Connector or a Proxy-Service MUST contain a HTTPS URL in the <Issuer> element pointing to the SAML Metadata object of the Issuer of the request/assertion (see section 4.1.1 of [SAML-Meta] "Well-Known Location" method).

Requirements for TLS from [eIDAS Crypto] apply.

6.4.1.Caching of Metadata

Retrieving and validating metadata objects during the authentication process can have an impact on reliability and performance of the authentication process, e.g. if the metadata URL is not reachable due to network problems or verification is not possible due to unavailability of revocation information for certificates.

To mitigate this, eIDAS Nodes MAY cache metadata objects (see section 4.3.1 of [SAML-Meta]).

Note: If the node regularly retrieves, verifies and caches new metadata objects after having learned the corresponding URL from a SAML message, retrieving metadata during the authentication process is only necessary for the first contact to a new node or if the metadata URL has changed.

6.4.2.Pre-fetching of Metadata

To avoid the need to learn metadata URLs from SAML messages and/or the need to validate metadata objects during the authentication process, metadata objects can be pre-fetched.

To facilitate pre-fetching, each MSs SHOULD publish a structured list (i.e. MetadataServiceList) of metadata locations. The list MUST be published via a https URL. The format of the MetadataServiceList is defined in [eIDAS SAML].

6.5. METADATA VERIFICATION

For verification of signed SAML metadata, verifiers MUST build and verify a Certification Path according to [RFC5280] starting from a trusted trust anchor (see section 6.2) and ending at the signer of the metadata object. Revocation check MUST be performed for all certificates containing revocation information.

SAML metadata files directly exchanged between Connectors and Middleware-Services do not need to be explicitly verified, trust is conveyed implicitly via the direct exchange.

All restrictions contained in the Metadata object (e.g. validity period) MUST be honoured by the verifier.

7. Operational and Security Requirements

7.1. Nodes

Nodes MUST NOT store any transaction data containing personal data beyond as required by Article 9(3) of [eIDAS IF]:

> *The node operator shall store data which, in the event of an incident, enable reconstruction of the sequence of the message exchange for establishing the place and the nature of the incident. The data shall be stored for a period of time in accordance with national requirements and, as a minimum, consist of the following elements:*
>
> *a) node's identification;*
>
> *b) message identification;*
>
> *c) message date and time.*

7.2. Node Operators

Information assurance requirements for eIDAS-Services are laid down in Article 10 of [eIDAS IF]:

> *(1) Node operators of nodes providing authentication shall prove that, in respect of the nodes participating in the interoperability framework, the node fulfils the requirements of standard ISO/IEC 27001 by certification, or by equivalent methods of assessment, or by complying with national legislation.*
>
> *(2) Node operators shall deploy security critical updates without undue delay.*

The vendors SHALL provide necessary security updates in a timely manner. This includes the eIDAS-Connector as well as eIDAS-Services.

7.3. Middleware Provisioning

The MS notifying a middleware-based eID scheme is responsible to provide the necessary Middleware. The Middleware SHALL meet the following requirements:

- Middleware SHALL be provisioned as pre-configured virtual machine in Open Virtualization Format. Configuration of the virtual machine by the operator SHOULD be supported by scripts or similar provided by the notifying MS.
- The virtual machine SHALL be provided for or SHALL be configurable for test and production environments.

- The virtual machine SHALL expose logging information via a syslog-interface and health information via an SNMP-interface.
- Generation of keys SHALL be performed under control of the hoster of the Middleware.

The Middleware SHALL be provided to DG DIGIT for bundling with the CEF-implementation (see section 8.1) and to the other Member States (if requested).

The Middleware SHALL be accompanied by documentation, providing at least:

- Instructions for how to access an administrative or service account on the operating system needed for configuration and troubleshooting.
- Instructions on how to install keys and certificates and other security-relevant topics.
- Requirements for the operational environment, including network configuration, DNS configuration, firewall ports to open and expected storage capacity needed.
- Instructions on how to monitor the MW e.g. for heartbeat, crashes, resource utilization, response times and security events.
- Instructions on how to install new versions and patches in a way that does not erase logdata, configuration etc.
- Instructions on how to backup the machines (what files), and when logs files can be erased.
- Instructions on how to scale in case additional processing power is needed and how to implement and handle fail-over.
- Instructions how to restart the machines and troubleshoot common problems.

Notifying MSs SHALL provide service and support for the Middleware.

7.4. TLS

Communication between eIDAS-Connectors and eIDAS-Services via the browser of the user SHALL be protected by TLS. The requirements for TLS and TLS certificates from [eIDAS Crypto] apply.

7.5. SAML

Communication via SAML MUST be cryptographically protected according to [eIDAS Crypto]. This includes encryption of SAML Assertions and signing all SAML protocol messages.

Before verifying a signature, the verifier MUST perform a XML schema validation of the signed object. For encrypted parts of a SAML protocol message, the decrypted content MUST be additionally validated after decryption.

It is RECOMMENDED to carefully consider the well-known security aspects of SAML-based systems (see [SAML-Sec]) and the Best Practices for XML Signatures (see [XMLSig BP]).

7.6. KEY STORAGE

Private cryptographic keys MUST be securely stored.

Public keys used to authenticate SAML assertions MUST be stored protected against manipulation.

8. Implementations (informative)

This section provides information on the currently known implementations.

Note: This list is informative only, being listed in this section does not imply compliance to the specification.

8.1. CEF

An implementation of the eIDAS-Connector and the eIDAS-Proxy-Service as a single package licensed under the EUPL is provided by DG DIGIT under the CEF (Connecting Europe Facility) program. Requirements on the implementation, including provisioning and service/support are managed by the CEF Management Board.

This implementation is provided bundled with the Middlewares provided by the Member States having notified a middleware based scheme.

8.2. MOA

The MOA (Modules for Online Applications) software components enabling and supporting Austrian eID identification and authentication can be downloaded from https://joinup.ec.europa.eu/software/moa-idspss/.

8.3. Middleware-Service for eIDAS-Token

The Middleware-Service for an eIDAS-Token [eIDAS-Token] based eID scheme is defined in [TR- 03130-3].

References

[eIDAS-Token]	BSI: TR-03110 Advanced Security Mechanisms for Machine Readable Travel Documents and eIDAS Token
[TR-03130-3]	BSI: TR-03130-3 eIDAS-Middleware-Service for eIDAS-Token, https://www.bsi.bund.de/EN/eID-TR
[eIDAS Attributes]	eIDAS: SAML Attribute Profile
[eIDAS SAML]	eIDAS: SAML Message Format
[eIDAS Crypto]	eIDAS: Security Requirements for TLS and SAML
[eIDAS IF]	EU: COMMISSION IMPLEMENTING REGULATION (EU) 2015/1501 of 8 September 2015 on the interoperability framework pursuant to Article 12(8) of Regulation (EU) No 910/2014 of the European Parliament and of the Council on electronic identification and trust services for electronic transactions in the internal market
[eIDAS]	EU: REGULATION (EU) No 910/2014 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC
[RFC2119]	IETF: RFC 2119: S. Bradner: Key words for use in RFCs to Indicate Requirement Levels
[RFC5280]	IETF: RFC 5280: D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk: RFC 5280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
[SAML-Core]	OASIS: Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0
[SAML-Binding]	OASIS: Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0
[SAML-Sec]	OASIS: F. Hirsch, R. Philpott, E. Maler: Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0
[SAML-Meta]	OASIS: Metadata for the OASIS Security Assertion Markup Language (SAML) v2.0
[SAML-MetaIOP]	OASIS: SAML V2.0 Metadata Interoperability Profile Version 1.0
[XMLSig BP]	W3C: XML Signature Best Practices, www.w3.org/TR/xmldsig-bestpractices

eIDAS Interoperability Architecture - Version 1.2

2019 – 21 pages