eIDAS-Node Migration Guide

Version 2.1

Document history

Version	Date	Modification reason	Modified by
1.4 Pre release	31/08/2017	Origination	DIGIT
1.4 Official release	06/10/2017	Minor update to section 3.7	DIGIT
2.0	27/03/2018	New document for software version 2.0.0, please see content	DIGIT
2.1	12/07/2018	New document for software version 2.1, please see contents	DIGIT

Disclaimer

This document is for informational purposes only and the Commission cannot be held responsible for any use which may be made of the information contained therein. References to legal acts or documentation of the European Union (EU) cannot be perceived as amending legislation in force or other EU documentation.

The document contains information of a technical nature and does not supplement or amend the terms and conditions of any procurement procedure; therefore, no compensation claim can be based on the contents of this document.

Table of contents

DOCUMENT HISTORY 2
TABLE OF CONTENTS 4
1. INTRODUCTION 5
1.1. Document structure 5
1.2. Document aims 5
2. PREREQUISITES 6
3. CHANGES 7
3.1. Summary of changes 7
3.2. Remove xsi:type from LOA's metadata's attribute value 7
3.2.1. Code changes 7
3.2.2. Configuration changes 7
3.3. Use of SingleSignOnService instead of hardcoded URLs 7
3.3.1. Code changes 7
3.3.2. Configuration changes 8
3.4. Usage of simple DSI keys in SAML messages 8
3.4.1. Code changes 8
3.4.2. Configuration changes 9
3.5. eIDAS protocol versioning 9
3.5.1. Code changes 9
3.5.2. Configuration changes 9
3.6. Change in Gender allowed values 10
3.6.1. Code changes 10
3.6.2. Configuration changes 10
3.7. Check of metadata signing certificate against a Trust Chain 10
3.7.1. Code changes 10
3.7.2. Configuration changes 11
3.8. Corrected reference key name related to TLS cipher suites 12
3.8.1. Configuration changes 12
3.9. Other fixes/improvements requiring no action 12
3.9.1. Build separation between Demo and Node modules 12
3.9.2. Increase the preference of the relevant configurer using the order property 12
3.9.3. eIDAS flow with JavaScript off 12
3.9.4. Update copyright headers and remove authorship 12
3.9.5. Implemented CRLF protection for audit logs. Small correction in comment in external config file 13
3.9.6. Change of error code/message when metadata cannot be read 13
3.9.7. Change the order of validation when processing the SP request 13
3.9.8. Fix Junit failing tests due to validUntil in the past 13

1. Introduction

This document is intended for a technical audience consisting of developers, administrators and those requiring detailed technical information on how to configure, build and deploy the eIDAS-Node application.

The purpose of this document is to facilitate migration from eIDAS-Node v2.0 to eIDAS-Node v2.1.

1.1. Document structure

This document is divided into the following sections:

Chapter 1 — *Introduction*: this section.

Chapter 2 — *Prerequisites*: Identifies any prerequisites that are required before migrating your eIDAS-Node to version 2.1.

Chapter 3 — *Changes*: Contains detailed information about the changes that should be taken into consideration when migrating to eIDAS-Node version 2.1.

1.2. Document aims

The main aim of this document is to provide information on all the changes requiring your action when migrating to eIDAS-Node version 2.1, including:

configuration changes; and

changes to code.

2. Prerequisites

Before starting your migration to eIDAS-Node version 2.1 you should have:

already implemented eIDAS-Node version 2.0;
downloaded the eIDAS-Node v2.1 Integration Package; and
downloaded the latest documentation.

3. Changes

3.1. Summary of changes

In eIDAS-Node version 2.1 there are several changes that affect your implementation. The main changes are:

Remove xsi:type from LOA's metadata's attribute value(see section **Error! Reference source not found.**);

Use of SingleSignOnService instead of hardcoded URLs (see section **Error! Reference source not found.**);

Implement usage of simple DSI keys in SAML messages

For other fixes and improvements requiring no action, please see section 3.9.

3.2. Remove xsi:type from LOA's metadata's attribute value

Removed xsi:type from LOA's metadata's attribute value when the parameter metadata.hide.loatype from eidas.xml is set to true.

3.2.1. Code changes

Methods `setHideLoaType`, `isHideLoaType` have been added to Interface `EidasMetadataParametersI`.
Field `hideLoaType` has been added to class `EidasMetadataParameters`.
Field `hideLoaType` has been added to class `EidasMetadata`.

The field EIDAS_METADATA_HIDE_LOA("metadata.hide.loatype") has been added to EIDASValues enum.

3.2.2. Configuration changes

The following entry has been added to `eidas.xml`

- metadata.hide.loatype

3.3. Use of SingleSignOnService instead of hardcoded URLs

This has changed the behaviour of the Connector to always go to the URL fetched from SingleSignOnService.

3.3.1. Code changes

In `EidasNodeValidationUtil`, new methods have been introduced that perform the validation of the url from metadata against a hard coded list specified in eidas.xml.

In `AUCONNECTORSAML`, url is read from metadata instead of eidas.xml.

In `EIDASValues`, new fields were introduced:
EIDAS_CONNECTOR_REDIRECT_URIDEST("connector.url.redirect.location.whitelist"),

EIDAS_CONNECTOR_POST_URIDEST("connector.url.post.location.whitelist").

3.3.2. Configuration changes

Removal of entries from `eidas.xml`:

service**x**.url

New entries in `eidas.xml`:

connector.url.redirect.location.whitelist

connector.url.post.location.whitelist

3.4. Usage of simple DSI keys in SAML messages

Added the option to the eidas-node to sign the (and conversely verify the signature on) Authentication Request with a simple public RSAKey instead of the full X509Certificate. See EID-570.

3.4.1. Code changes

- EIDAS-SAMLEngine SignatureKey.java
- EIDAS-SAMLEngine KeyStoreSignatureConfigurator.java
- EIDAS-SAMLEngine SignatureConfiguration.java

Added required logic and variables to make effective the configuration flags `request.sign.with.key.value, response.sign.with.key.value`

- EIDAS-SAMLEngine ProtocolSignerI.java
- EIDAS-SAMLEngine AbstractProtocolSigner.java

Overloaded the existing

```
T sign(@Nonnull T signableObject) throws EIDASSAMLEngineException;
With
T sign(@Nonnull T signableObject, boolean onlyKeyInfoNoCert) throws
EIDASSAMLEngineException;

Added getter for the newly added flag of this feature
boolean isRequestSignWithKey();
boolean isResponseSignWithKey();


Added utility method to retrieve X509SecurityCerificate based on the incoming
RSAKeyValue
X509Credential getTrustedCertificateFromRSAKeyValue(RSAKeyValue
signatureRsaKeyValue) throws SecurityException, EIDASSAMLEngineException;
```

- EIDAS-SAMLEngine ProtocolEngine.java
- EIDAS-SAMLEngine AbstractProtocolEngine.java

Modified signing and verifying the signature of the Authentication request to implement the feature `protected AuthnRequest signRequest(@Nonnull AuthnRequest request) throws EIDASSAMLEngineException`
`public IAuthenticationRequest unmarshallRequestAndValidate(@Nonnull byte[] requestBytes, @Nonnull String citizenCountryCode)`

- EIDAS-Encryption CertificateUtil.java

overloaded the existing
`public static String getCountry(KeyInfo keyInfo) with versions accepting X509Cetificate as parameter.`

3.4.2. Configuration changes

New entries in `eidas.xml`:

request.sign.with.key.value set to true to have the eidas-node sign Authentication Requests it's originating , with the simple public RSAKey, as opposed to incorporating the full X509Certificate in the signature

response.sign.with.key.value set to true to have the eidas-node sign Authentication Responses it's originating , with the simple public RSAKey, as opposed to incorporating the full X509Certificate in the signature

3.5. eIDAS protocol versioning

In order to facilitate interoperability between different implementations of eIDAS-Nodes, the eIDAS-Node deployment version (aka. protocol versioning) was implemented. This implementation follows the definition in draft 1.6 of eIDAS message format 1.2. These values will appear in the eIDAS node's metadata.

3.5.1. Code changes

The interface EidasMetadataParametersI was modified, setters and getters related to protocol version and application identifier. EidasNodeMetadataGenerator was also modified to retrieve the mentioned entries from the Node Properties. The generation of EntityAttributes was also refactored and the code grouped inside method generateEntityAttributes.

Also in EidasMetadata, the code inside method generateExtensions was refactored: methods were created to contain the generation of child elements of Extensions element. Examples of newly added methods are: generateSigningMethods, generateSpType.

3.5.2. Configuration changes

Two new entries were introduced in eidas.xml:

- <entry key="eidas.protocol.version">1.1</entry>
- <entry key="eidas.application.identifier">CEF:eIDAS-ref:2.1</entry>

These properties allow configuring the values of protocol version and application identifier, which will be displayed in the eIDAS Node's metadata.

When none of these entries is present, none of the XML elements related to eidas deployment versioning (AKA protocol versioning) will be displayed in the eIDAS Node's Metadata.

3.6. Change in Gender allowed values

In version 2.0 a bug prevented to set the gender other that Male or Female. Therefore, neither values mentioned in the specification: "Unspecified" or " Not Specified" were possible. This issue was fixed.

At the present, the possible values for Gender were clarified in the eIDAS specifications meetings and only "Male", "Female" and "Unspecified" are possible values. However, to prevent interoperability issues the change regarding Gender was done to allow also "Not Specified".

3.6.1. Code changes

To allow the "Not Specified" value for gender, the Gender class was changed and " Not Specified" constant was added and the PATTERN_GENDER_EIDAS in EidasAttributeValidator class was adjusted. Schema files saml_eidas_natural_person.xsd and saml_eidas_representative_natural_person.xsd were also modified to allow this value.

3.6.2. Configuration changes

The file user.properties was changed and a new user genderu was added to test sending the "Unspecified" value.

3.7. Check of metadata signing certificate against a Trust Chain

The possibility to check a certificate against its trust chain was implemented. It is now possible to check the validity of the signed metadata using a certificate that belongs to the trust chain of the certificate that related to the key that signs the metadata. This change was done taking also in consideration compatibility with version 1.4.1 of the eIDAS Node.

3.7.1. Code changes

CertificateUtil was modified; the ExplicitKeyTrustEvaluator was replaced by the CertPathPKIXTrustEvaluator in the validation of the signing credential to allow validation with other certificates in the chain. Also method toCertificate(@Nonnull KeyInfo keyInfo) was changed so that, the signing certificate is assumed is the first appearing inside the KeyInfo element. The order how certificates in the chain are published was changed so that the one that relates to the signing key certificate is the first followed by his signer, etc., until it reaches the root of the chain. This was done to allow interoperability with 1.4.1 versions.

In the SignatureConfiguration class the signature of the constructor changed and ImmutableSet<X509Certificate> metadataKeystoreCertificates was introduced as a parameter. This parameter contains all the metadata certificates contained in the kesytore related to the metadata signature functionality. Related to this change, the corresponding getter was added, getMetadataKeystoreCertificates, and the hashCode and toString was also adjusted.

The KeyStoreSignatureConfigurator class was updated due to the changes in the SignatureConfiguration. Also, a new method was added so that the certificates from a keystore can be retrieved as an ImmutableSet<X509Certificate>.

The AbstractProtocolSigner constructors were also modified and the ImmutableSet<X509Certificate> metadataKeystoreCertificates parameter was added. The metadata signing credentials are now set with its certificate chain. The order is reversed so that it allows compatibility with 1.4.1 version.

In the AbstractProtocolSigner, a new method of checkValidTrust was introduced that depending on the certificate chain of the signature consumed, and on the existence of the metadata key signing certificate as a trusted entry, performs validation using the ExplicitKeyTrustEvaluator or performs validation of trust chain using CertPathPKIXTrustEvaluator. New methods of hasLeaf and getIssuerX509Certificate were also added to check the existence of the signing certificate as trusted and for retrieving the issuer certificate from the trusted entries.

New classes TestEidasNodeMetadataTrustChain and TestEidasNodeFileMetadataProcessorTrustChain were created that contain test cases related to trust chain possibilities.

3.7.2. Configuration changes

With the introduction of the trust chain support, it is necessary to have specific configuration for metadata. Therefore, the configuration related to the keystores in both SignModule_Connector.xml and SignModule_Service.xml need from now on to have the "metadata." entries defined. So that it references a specific keystore for metadata signing and trust chain publishing purposes only. A possible configuration is:

```
<entry
key="metadata.keyStorePath">../keystore/eidasKeyStore_METADATA_TC.jks</en
try>

<entry key="metadata.keyStorePassword">local-demo</entry>

<entry key="metadata.keyPassword">local-demo</entry>

<entry key="metadata.issuer">CN=metadataNode, OU=Node, O=eIDAS,
L=Brussels, ST=BRUSSELS, C=EU</entry>

<entry key="metadata.serialNumber">89e1841a35e10504</entry>

<entry key="metadata.keyStoreType">JKS</entry>
```

The keystore referred, in the example above, `eidasKeyStore_METADATA_TC.jks` must contain the private key that will be used to sign the metadata and all the certificates that compose its trust chain up to the root certificate or the sub ca that is intended to be trusted. The new provided demo keystore `eidasKeyStore_METADATA_TC.jks`, contains two trusted entries (alias intermediatecametadata and rootcametadata), and the private key entry that signs the metadata (alias metadatanode).

Note that this keystore will be used to get the metadata signing key and to retrieve the certificates to be published at the Connector and Proxy-Service metadata. Adding an extra non-related certificate will be published in the eIDAS Metadata URLs. This will give a wrong impression to the party that consumes the metadata, that it can trust that certificate. However, in the current version, the certification path will be validated. So trusting a certificate outside the trust chain, even if published, will result in error.

To validate the metadata signed with the above trust chain, it will be necessary to add as trusted entry one of the certificates that compose the trust chain. Taking as example `eidasKeyStore_METADATA_TC.jks`, one of certificates with alias rootcametadata, intermediatecametadata or metadatanode will need to be imported into the keystore used to trust/validate the metadata. In the provided configuration it will be the `eidasKeyStore.jks`.

3.8. Corrected reference key name related to TLS cipher suites

The key name in the EIDAS-Config/eidas.xml was corrected from tls.enabled.cipher.suites to tls.enabled.ciphers, which was used in the code and mentioned in other documenters e.g. eIDAS-Node Installation and Configuration Guide.

3.8.1. Configuration changes

If you used tls.enabled.cipher.suites in eidas.xml you will need to replace that key name by tls.enabled.ciphers.

3.9. Other fixes/improvements requiring no action

3.9.1. Build separation between Demo and Node modules

Added NodeOnly and DemoToolsOnly maven profile to EIDAS-Parent pom.xml to build only the respective modules.

3.9.2. Increase the preference of the relevant configurer using the order property

The Node applicationContext has some properties with default values. In order to make these default values work we modified the order of the relevant configurer placeholderConfig to a negative value at the applicationContext.xml.

```
<property name="order" value="-2147483648"/>
```

3.9.3. eIDAS flow with JavaScript off

Refactoring of js pages in order to make the eIDAS flow to work with Javascript turn off.

3.9.4. Update copyright headers and remove authorship

File types affected: java, xml, jsp, properties, and javascript js.

3.9.5. Implemented CRLF protection for audit logs. Small correction in comment in external config file

`LoggingSanitizer` was implemented that contains removeCRLFInjection method to prevent CRLF injection. This method is executed in `AUCONNECTORSAML.java` and `AUCONNECTORSAML.java` for some fields that are external to the node and are sent to logs potentially containing CRLF injection.

3.9.6. Change of error code/message when metadata cannot be read

Change done in AUCONNECTORSAML.java to use SAML_ENGINE_NO_METADATA instead of SPROVIDER_SELECTOR_INVALID_SAML when metadata can not be read.

3.9.7. Change the order of validation when processing the SP request

Moved the validation of country code before the validation of metadata url when processing the SP request by the Node's connector in `AUCONNECTORSAML`.

3.9.8. Fix Junit failing tests due to validUntil in the past

Updated ed.xml file used by Junit tests, containing metadata. The validUntil attribute was updated so it is now in the Future and does not fail validity on this value.