eIDAS-Node Migration Guide

Version 2.0

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

Table of contents

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 v1.4 to eIDAS-Node v2.0.

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 2.0.
- Chapter 3 — *Changes*: Contains detailed information about the changes that should be taken into consideration when migrating to eIDAS-Node version 2.0.

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 2.0, including:

- configuration changes; and
- changes to code.

2. Prerequisites

Before starting your migration to eIDAS-Node version 2.0 you should have:

- already implemented eIDAS-Node version 1.4;
- downloaded the eIDAS-Node v2.0 Integration Package; and
- downloaded the latest documentation.

3. Changes

3.1. Summary of changes

In eIDAS-Node version 2.0 there are several changes that affect your implementation. The main changes are:

- Split of Specific into Specific Connector and Specific Proxy Service: different application context, Build process as WAR/JAR (see section 3.3);
- New eIDAS metadata module (see section 3.4);
- Migration from OpenSAML 2 to OpenSAML 3 (see section 3.5);
- TLS cipher suites support (see section 3.6);
- Generic-Specific communication interface (see section 3.7);
- Improvements to LightResponse interface regarding Subject (see section 3.8);
- Improvements to LightResponse interface regarding RelayState (see section 3.9);
- NameIDFormat changed to optional (see section 3.10);
- Upgrade of javax.servlet-api (see section 3.11);
- Wildfly 11.0.0 support (see section 3.12);
- GlassFish Open Source Edition 5.0 (full profile) support (see section 3.13);
- End of GlassFish 3 support (see section 3.14);
- WebSphere Application Server Liberty Core v9/17.0.0.4 support(see section 3.15);
- WebSphere configuration change (see section 3.16);
- Look and feel change (see section 3.17);
- Removal of incorrect legal person attributes (see section 3.18); and
- Cleanup of external configuration entries (see section 3.19).

For other fixes and improvements requiring no action, please see section 3.20.

3.2. Citizen consent logic moved to Specific Proxy Service (changed the external configuration entries)

The code implementing the user consent for request or response has been moved to, and refactored in, the `Eidas-SpecificProxyService` module. Please refer to the *eIDAS-Node Demo Tools Installation and Configuration Guide* (when published) for more details on the description of these entries.

3.2.1. Code changes

Consent-related code has been removed from classes `ColleagueRequestServlet.java` and `ServiceControllerService`. `CitizenConsentServlet` has been removed. Method `processIdpSpecificResponse` signature in `ISERVICESAMLService` has been changed.

Views citizenConsent.jsp and presentConsent.jsp have been removed. Constants have been removed from EidasErrorKey, NodeParameterNames, NodeViewNames. EidasParameterKeys.

Methods processCitizenConsent, getCitizenConsent have been removed from Interface ISERVICECitizenService.

3.2.2. Configuration changes

The following entries have been removed from eidas.xml

- service.specificidpredirect.url
- service.citizenConsentUr
- service.askconsent.type
- service.askconsent.value
- service.askconsent.all.attributes
- service.askconsent.attribute.names.only

Also, since the consent functionality has been removed from the node the related error codes have also been removed:

- 202006={0} - Citizen consent is malformed.
- 202007={0} - Consent not given for a mandatory attribute
- 202012={0} - Citizen consent not given.

3.3. Split of Specific into Specific Connector and Specific Proxy Service: different application context, Build process as WAR/JAR

In the previous version an Eidas-Specific module was included in eIDAS-Node that was responsible for converting the MS-Specific Request and Reponses into eIDAS Requests and Responses.

In the current version two modules exist:

- EIDAS-SpecificConnector which implements a demo protocol conversion between MS Specific module and eIDAS-Node Connector; and
- EIDAS-SpecificProxyService which implements a demo protocol conversion between eIDAS-Node Proxy Service and MS Specific module.

3.3.1. Code changes

Eidas-Specific module has been completely removed. Two new modules EIDAS-SpecificConnector and EIDAS-SpecificProxyService have been created.

The pom.xml of the EidasNode has changed to handle the Standard Deployment where these two new modules are built as WARs and the Monolithic Deployment where the two modules are build as JARs and included in the Eidas-Node.war.

3.3.2. Configuration changes

New entries in `eidas.xml` have been added:

- `specific.proxyservice.request.receiver`
- `specific.connector.response.receiver`

Changes have been made regarding environment variables. The environment variable `SPECIFIC_CONFIG_REPOSITORY` that existed in previous versions has been replaced by: `SPECIFIC_CONNECTOR_CONFIG_REPOSITORY` and `SPECIFIC_PROXY_SERVICE_CONFIG_REPOSITORY` environment variables. These will be used by the `EIDAS-SpecificCommunicationDefinition` module therefore must be configured.

3.4. New eIDAS metadata module

A new module has been created with the purpose of concentrating all the Metadata related classes and functionalities that were scattered in `EIDAS-SAML-Engine` and `Eidas-Node` modules. Therefore to access metadata related functionalities, you need to use these new module classes.

3.4.1. Code changes

The classes that now compose the several classes have been moved or refactored. The EIDAS-Metadata module is now a dependency of `EIDAS-SAMLEngine`.

The following describes some major code changes related to this module.

A new interface `EidasMetadataParametersI` has been introduced, a value object interface responsible for present ALL data items for an eIDAS Metadata, including technical and business information. It aggregates Role Descriptor data. `EidasMetadataParameters` class implements this interface.

Due to these changes several classes have also been changed. `AbstractMetadataCaching` has been refactored, methods signatures were changed in order to replace retrieving and caching `EntityDescriptor` instanced by `EidasMetadataParametersI` instances. `DistributedMetadataCaching` has also been refactored to handle `EidasMetadataParametersI` instead of `EntityDescriptor`. Deprecated class `SerializedEntityDescriptor` has been removed and its usage in the code replaced by `EidasMetadataParametersI`. `SimpleMetadataCaching` has also been refactored in order to handle `EidasMetadataParametersI`.

A new interface `EidasMetadataRoleParametersI` has been introduced, which has the responsibility to represent either SP or IDP role descriptors, which is determined by `getRole()` method. This is implemented by `EidasMetadataRoleParameters` using `MetadataConfiguration`, a new factory class that simple creates and returns instances of `EidasMetadataParameters` and of `EidasMetadataRoleParameters`.

Due to change in the signature of methods of `MetadataFetcherI` the class `WrappedMetadataFetcher` overridden method `getEntityDescriptor` has been replaced by `getEidasMetadata`.

AUCONNECTORSAML methods now throw EIDASMetadataException instead of EIDASSAMLEngineException and also overrides method getSamlEngine() which has been added to interface ICONNECTORSAMLService.

ConnectorMetadataGeneratorServlet code has been refactored so that it only generates the metadata for /ConnectorMetadata URL. As /ConnectorResponderMetadata has been removed, there is no need to have dual behaviour on this servlet regarding metadata generation.

ProxyServiceMetadataGeneratorServlet has been refactored in a similar way to ConnectorMetadataGeneratorServlet. There is now no need for this servlet to produce metadata for both /ConnectorMetadata and /ConnectorResponderMetadata URLs, since /ConnectorResponderMetadata has been removed.

EidasNodeMetadataGenerator methods were refactored in terms of field names and related getters and setters and also the method generateMetadata signature has been changed so the check on spEngineName and idpEngineName to decide which metadata to generate has been replaced by idpRole Boolean. Adaptation to throw EIDASMetadataException has also been introduced. Method addBindingLocation has been removed. A new class NodeMetadataUtil, to be used by EidasNodeMetadataGenerator, has been introduced in the EidasNode module. It contains methods to create the ContactData and OrganizationData related to Connector and ProxyService.

The signature of method configureProtocolProcessor in DOMConfigurator class has changed, as well as other parts of the code due to the addition of MetadataClockI as a parameter.

In EidasExtensionConfiguration registering of elements related to RequestedAttribute, RequestedAttributes, SigningMethod, DigestMethod, SPtype, SPCountry and XSAny have been removed. These configurations have moved to EidasExtensionConfiguration.

EidasProtocolProcessor has been refactored to adapt to the introduction of MetadataClockI and to adapt to EIDASMetadataException, metadataFetcher related methods were renamed. SPSSODescriptor class has been replaced by EidasMetadataRoleParametersI class.

In AttributeUtil, new methods have been introduced that perform the safecopy of attribute definitions into another set (clone) and to perform attribute definition sets comparison.

Exception classes: EIDASMetadataException and EIDASMetadataProviderException are defined in the EIDAS-Metadata module which implied several adaptations in classes from other modules.

ContactData and OrganizationData have been moved from EIDAS-SAMLEngine to EIDAS-Commons and CertificateAliasPair has been moved from EIDAS-SAMLEngine to EIDAS-Encryption. SPType related classes have been moved from EIDAS-SAMLEngine to EIDAS-Metadata.

Due to the amount of refactoring, Junit tests EidasNodeMetadataGeneratorTest and TestEidasNodeMetadataLoader have been changed accordingly.

The `applicationContext.xml`, more specifically beans `serviceMetadataGeneratorIDP` and `connectorMetadataGeneratorSP` have been changed to adapt to changes in `EidasNodeMetadataGenerator` and `EidasNodeMetadataGenerator` fields.

3.4.2. Configuration changes

Removal of entries from `eidas.xml`:

- `ssos.connectorMetadataGeneratorIDP.redirect.location`
- `ssos.connectorMetadataGeneratorIDP.post.location`
- `connector.responder.metadata.url`
- `service.requester.metadata.url`

3.5. Migration from OpenSAML 2 to OpenSAML 3

The code of previous eID versions used OpenSAML 2.6.5 based dependencies. This version no longer uses it. Now OpenSAML 3.3.0 dependencies are used. This change has also implied a change in the http client dependency previously used.

3.5.1. Code changes

The internal repository (`EIDASprojectrepo`) which included the OpenSAML 2.6.5 dependency has been removed from EIDAS-Parent and also the entire `/EIDAS-Encryption/src/main/lib` folder has been removed.

The following dependencies have been removed:

- `org.opensaml:opensaml` version 2.6.5-eidas_1
- `org.apache.santuario:xmlsec` version 2.0.5
- `commons-httpclient:commons-httpclient` version 3.1

New dependencies have been added:

- `org.opensaml:opensaml-saml-api` version 3.3.0
- `org.opensaml:opensaml-saml-impl` version 3.3.0
- `org.opensaml:opensaml-xmlsec-impl` version 3.3.0
- `org.opensaml:opensaml-core` version 3.3.0
- `org.opensaml:opensaml-xmlsec-api` version 3.3.0
- `net.shibboleth.utilities:java-support` version 7.3.0
- `org.apache.httpcomponents:httpclient` version 4.5.5
- `org.apache.httpcomponents:httpcore` version 4.4.9

In `BaseMetadataFetcher` the SSL connection factory has been re-implemented because of the new `HttpClient` and `HttpClientBuilder` classes now being used. Also, `TLSProtocolSocketFactory` has been replaced by `TLSSocketFactory`.

The way to initialise OpenSAML has changed, instead of using `SAMLBootstrap` `OpenSamlHelper` class is now used, more specifically calling the initialise method. The Configuration `registerObjectProvider` method has been replaced by `XMLObjectProviderRegistrySupport` class (opensaml-core) method `registerObjectProvider`. This change impacted classes `AbstractProtocolEncrypter` and `AbstractProtocolEngine`.

In `AbstractProtocolSigner` some changes have been made and several methods and classes related to security configuration and validation for signatures have been changed.

The `ValidatorSuites` featured in OpenSAML 2.6.5 do not exist in Opensaml 3.3.0 so `Validator` classes in the code in the `EIDAS-SAML-Engine` module have been implemented. These validation classes are executed from `ProtocolEngine` from methods `validateRequestWithValidatorSuite` and `validateResponseWithValidatorSuite`..

Other changes have been the removal of `validateAttributeValueFormat` method from `AUCONNECTORSAML` and the removal of `SAMLBootstrap`.

3.5.2. Configuration changes

None

3.6. TLS cipher suites support

The following show the code and configuration changes for TLS cipher suites support.

3.6.1. Code changes

`BaseMetadataFetcher` code has changed to retrieve the TLS enabled ciphers from entry `tls.enabled.ciphers`. A new class `TLSSocketFactory` is used.

3.6.2. Configuration changes

A new entry `tls.enabled.ciphers` is now in `eidas.xml` to define the TLS cipher suites to be used by the `HttpClient` when accessing HTTPS URLs, such as, when requesting metadata pages. Its value is the concatenation of TLS cipher suites to be supported, separated by commas.

3.7. Generic-Specific communication interface

A new interface has been implemented to allow the communication between the Node and the MS Specifics (Specific Connector and Specific Proxy Service). SAML communication between the MS Specific modules and the Node has been removed, as has metadata generation dedicated to SP and IdP. The URLs `/EidasNode/ConnectorResponderMetadata` and `/EidasNode/ServiceRequesterMetadata` have also been removed.

In version 2.0 the Light Request/Response are now exchanged between the MS Specific modules and the eIDAS-Node. The Standard Deployment uses the Hazelcast share map to store a representation of XML Light Request and Light Response and sends/receives

BinaryLightToken which is used as a key to store/retrieve the Light Request and Light Response.

3.7.1. Code changes

The EIDAS-SpecificCommunicationDefinition module has been completely re-implemented and its purpose is to provide the communication functionality between Eidas-Node and MS-Specifics. Eidas-Node, Specific Connector and Specific ProxyService modules have as a dependency the EIDAS-SpecificCommunicationDefinition module. The interface SpecificCommunicationService defines all the methods to store and retrieve the XML representation of ILightRequest/ILightResponse.
It has been implemented so that depending on the type of build/deployment: Standard Deployment or Monolithic Deployment, either distributed maps provided by Hazelcast maps or a local ConcurrentMap instance are used to provide the communication storage for LightRequest/LightResponse between the node and specific modules.

This module now has an independent application context based on specificCommunicationDefinitionApplicationContext.xml, where the definition of the maps instances is done.

The servlets SpecificConnectorRequestServlet and SpecificProxyServiceResponse are responsible for receiving the BinaryLightToken, using it to retrieve the ILightRequest or ILightResponse, respectively.

The ColleagueRequestServlet and ColleagueResponseServlet were modified so that they store the ILightRequest/ILightResponse respectively and create the BinaryLightToken to be used by the receiver to retrieve the ILightRequest/ILightResponse.

In this process two new tokenRedirect.jsp were introduced to send the token to the destination, either MS-Specific Connector or MS Specific Proxy Service.

3.7.2. Configuration changes

There are new configuration files:

- specificCommunicationDefinitionConnector.xml
- specificCommunicationDefinitionProxyservice.xml

These hold the entries needed to configure the issuer name, secret and algorithm to be used by the BinaryLightToken used in the exchange of Light Request/Light Response.

New maps have been added to hazelcast.xml.

3.8. Improvements to LightResponse interface regarding Subject

Two new fields: Subject and SubjectNameIdFormat have been added. Both are mandatory if the Response status is not a failure.

`Subject` is used in the `NameID` field of the assertion. The Node no longer automatically copies any attribute values here. The RECOMMENDED value is however a person identifier of [`eIDAS-Attr-Profile`].

`SubjectNameIdFormat` according to eIDAS Message Format 3.2, but must be the same as the `NameIdPolicy` specified in the Request (if there is one).

3.9. Improvements to LightResponse interface regarding RelayState

`RelayState` has been added to `LightRequest` and `LightResponse`. It is the MS-Specific part responsible to set the values through this interface, so it has nothing to do with incoming web requests. The default behaviour according to `SAML2Binding` specification is to relay it back so it is recommended in the MS-Specific Proxy to set `lightResponse.relayState(lightRequest.getRelayState())`.

3.10. NameIDFormat changed to optional

It is no longer necessary to specify the `NameIDFormat` of the Request as it is no longer validated by the Proxy Service. The Proxy Service still validates the `NameIdFormat` sent by the IdP and the error message for the existing code 202013 has been changed to "Invalid Response" (previously "Invalid SAML Response").

3.11. Upgrade of javax.servlet-api

The servlet version of `javax.servlet-api` has been upgraded from 2.5 to 3.0.1

This allows the cookie configuration in `web.xml` and functionally similar code to be removed and provide protection against XSS.

3.12. Wildfly 11.0.0 support

Wildfly 11.0.0 is now supported. The build process is similar to that for JBoss7. Please refer to the *eIDAS-Node Demo Tools Installation and Configuration Guide* (when published) for more details.

3.13. GlassFish Open Source Edition 5.0 (full profile) support

GlassFish 5 is now supported. The build process is similar to that for GlassFish Open Source Edition 4.1 (full profile). Please refer to the *eIDAS-Node Demo Tools Installation and Configuration Guide* for more details.

3.14. End of GlassFish 3 support

GlassFish 3 is no longer supported. It has been replaced by GlassFish 5.

3.15. WebSphere Application Server Liberty Core v9/17.0.0.4 support

The new version of WebSphere Application Server, Liberty Core v9/17.0.0.4 is now supported.

3.16. WebSphere configuration change

In order for eIDAS error messages to be properly displayed on WebSphere, the following property `<webContainer com.ibm.ws.webcontainer.enableErrorExceptionTypeFirst="true"/>` must be added to `$SERVER_HOME/usr/servers/defaultServer/server.xml file`. This is because of the way WebSphere deals with error page handling by first giving preference to HTTP error codes rather than exceptions resulting in an error page without eIDAS error code/message.

3.16.1. Configuration changes

The dependency from `xml-apis` 1.4 no longer needs manual intervention.

3.17. Look and feel change

The look and feel of the eIDAS Node has changed therefore you will need to incorporate these changes into your implementation.

3.17.1. Code changes

The following eIDAS-Node files have changed:

- `custom.css`
- `leftColumn.jsp`
- `leftColumnNoAnim.jsp`
- `connectorRedirect.jsp`

3.18. Removal of incorrect legal person attributes

The following incorrect legal person attributes were removed:

- http://eidas.europa.eu/attributes/legalperson/VATRegistration
- http://eidas.europa.eu/attributes/legalperson/LegalAddress
- http://eidas.europa.eu/attributes/legalperson/representative/VATRegistration
- http://eidas.europa.eu/attributes/legalperson/representative/LegalAddress

3.18.1. Code changes

Obsolete legacy code related to EID-423 has been removed:

- `AUCONNECTOR.java`

- AUSERVICE.java
- ColleagueRequestServlet.java
- ColleagueResponseServlet.java
- EidasProtocolProcessor.java
- EidasSpec.java
- LegalPersonSpec.java
- RepresentativeLegalPersonSpec.java

3.19. Cleanup of external configuration entries

Several unused/obsolete configuration entries were removed:

- DEMO-SP.qaalevel
- sp.default.parameters
- connector.id
- connector.destination.url
- connector.spSector
- connector.spApplication
- connector.spCountry
- connector.spInstitution
- saml.sp
- saml.idp
- connector.node.url
- service.node.url
- service.LoA
- eidas.supportFramingSamlRequest
- eidasNodeOnly

3.20. Other fixes/improvements requiring no action

3.20.1. Removal of BouncyCastle dependency

The BouncyCastle dependency has been removed from EIDAS-Encryption.

3.20.2. Improved parsing of metadata

In the convertExtensions method of MetadataUtil class a fix allows parsing of metadata produced by the German Middleware.

3.20.3. Added missing "AddressID" property to PostalAddress

`AddressID` property has been added to `PostalAddress` as in CORA ISA Vocabulary v0.3.