eIDAS-Node Installation Quick Start Guide v2.6

eIDAS eID Implementation

Exported on 04/15/2022

Table of Contents

Document history

Version	Date	Modification reason	Modified by
1.0	26/11/2015	Modifications to align with the eIDAS technical specifications.	DIGIT
1.1.0	29/06/2016	Modifications due to installation changes related to architectural and stability improvements Update of the deployments configuration and related libraries	DIGIT
1.2.0	20/01/2017	Configuration and stability improvements, please see Version 1.2.0 Release Notes.	DIGIT
1.3.0	05/05/2017	Modifications to align with changes in Technical Specifications version 1.1. For details, please see the Version 1.3.0 Release Notes.	DIGIT
1.4. Pre-Release	31/08/2017	Modifications to remove support for JBoss6. Support WebLogic 12.2 family of servers. Amend filename conventions to change '\' to '/'.	DIGIT
1.4. Official release	06/10/2017	Error corrections and improvements	DIGIT
2.0	28/03/2018	Changes in supported application servers.Configuration and stability improvements. Architectural changes (separation of Specific Connector and Specific Proxy Service), please see Version 2.0 Release Notes and the *eIDAS-Node Migration Guide* for detail.	DIGIT
2.1.	07/06/2018	Reuse of document policy updated and version changed to match the corresponding Release. Minor changes made to file references describing the release.	DIGIT
2.2	14/09/2018	Minor changes made to file references describing the release.	DIGIT

Version	Date	Modification reason	Modified by
2.3	20/06/2019	Document updated to reflect current installation and configuration	DIGIT
2.4	06/12/2019	Document updated to reflect current installation and configuration	DIGIT
2.5	11/12/2021	eIDAS-Node 2.5 release	DIGIT
2.6	04/2022	eIDAS-Node 2.6 release	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 a brief overview of technical nature and is not supplementing or amending terms and conditions of any procurement procedure; therefore, no compensation claim can be based on the contents of the present document.

List of abbreviations

The following abbreviations are used within this document.

Abbreviation	Meaning
eIDAS	electronic Identification and Signature. The Regulation (EU) N°910/2014[1] governs electronic identification and trust services for electronic transactions in the internal market to enable secure and seamless electronic interactions between businesses, citizens and public authorities.
IdP	Identity Provider. An institution that verifies the citizen's identity and issues an electronic ID.
LoA	Level of Assurance (LoA) is a term used to describe the degree of certainty that an individual is who they say they are at the time they present a digital credential.
MS	Member State.
SAML	Security Assertion Markup Language
SP	Service Provider

[1] http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2014.257.01.0073.01.ENG

List of definitions

The following definitions are used within this document.

Term	Meaning
eIDAS-Node	An eIDAS-Node is an application component that can assume two different roles depending on the origin of a received request. See eIDAS-Node Connector and eIDAS-Node Proxy Service.
eIDAS-Node Connector	The eIDAS-Node assumes this role when it is located in the **Service Provider's** (i.e. the receiving) Member State. In a scenario with a Service Provider asking for authentication, the eIDAS-Node Connector receives the authentication request from the Service Provider and forwards it to the eIDAS-Node of the citizen's country.
eIDAS-Node Proxy Service	The eIDAS-Node assumes this role when it is located in the **citizen's** (i.e. the sending) Member State. The eIDAS-Node Proxy Service receives authentication requests from an eIDAS-Node of another MS (their eIDAS-Node Connector). The eIDAS-Node Proxy-Service also has an interface with the national eID infrastructure and triggers the identification and authentication for a citizen at an identity and/or attribute provider.

1 Introduction

This document describes how to quickly install a Service Provider, eIDAS-Node Connector, eIDAS-Node Proxy Service and IdP from the distributions in this release package. The distributions provide preconfigured eIDAS-Node modules for running on each of the supported application servers (Tomcat, WildFly, WebLogic and WebSphere).

Detailed information on the setup and configuration of the sample eIDAS-Nodes, is included in the *eIDAS-Node Installation and Configuration Guide*.

Detailed information on integration of the eIDAS-Node into your national infrastructure is included in the *eIDAS-Node National IdP and SP Integration Guide*.

1.1 Document structure

This document is divided into the following sections:

- Section 1 − *Introduction:* this section;
- Section 2 − *Release content:* lists the files delivered with this release and describes their contents;
- Section 3 − *Overview of the preconfigured demo eIDAS-Node packages:* illustrates the setup of the configurations provided with this distribution;
- Section 4 − *Demo eIDAS-Node set up and configuration:* describes step-by-step how to install the demo configuration;
- Section 5 − *Specific configuration:* provides information on how the setup can be changed to suit your needs;
- Section 6 − *Compiling the modules from the source:* describes how to rebuild the Maven project if necessary;
- Section *7 — Enabling logging:* describes how to enable audit logging of the communications between eIDAS-Node Proxy Service and Connector.

1.2 Document aims

Describes how to quickly install demonstration versions of an eIDAS-Node Connector, eIDAS-Node Proxy Service, Service Provider (SP) and Identity Provider (IdP) from the distributions in this release package to enable familiarity with the eID software.

1.3 Other technical reference documentation

We recommend that you also familiarise yourself with the following eID technical reference documents, which are available on **Digital Home > eID** :

- *eIDAS-Node Installation and Configuration Guide* describes the steps involved when implementing a Basic Setup and goes on to provide detailed information required for customisation and deployment.
- *eIDAS-Node National IdP and SP Integration Guide* provides guidance by recommending one way in which eID can be integrated into your national eID infrastructure.
- *eIDAS-Node Demo Tools Installation and Configuration Guide* describes the installation and configuration settings for Demo Tools (SP and IdP) supplied with the package for basic testing.
- *eIDAS-Node and SAML* describes the W3C recommendations and how SAML XML encryption is implemented and integrated in eID. Encryption of the sensitive data carried in SAML 2.0 Requests and Assertions is discussed alongside the use of AEAD algorithms as essential building blocks.
- *eIDAS-Node Error and Event Logging* provides information on the eID implementation of error and event logging as a building block for generating an audit trail of activity on the eIDAS Network. It describes the files that are generated, the file format, the components that are monitored and the events that are recorded.
- *eIDAS-Node Security Considerations* describes the security considerations that should be taken into account when implementing and operating your eIDAS-Node scheme.

- *eIDAS-Node Error Codes* contains tables showing the error codes that could be generated by components along with a description of the error, specific behaviour and, where relevant, possible operator actions to remedy the error.

Disclaimer: The users of the eIDAS-Node sample implementation remain fully responsible for its integration with back-end systems (Service Providers and Identity Providers), testing, deployment and operation. The support and maintenance of the sample implementation, as well as any other auxiliary services, are provided by the European Commission according to the terms defined in the European Union Public License (EUPL) at https://joinup.ec.europa.eu/sites/default/files/custom-page/attachment/eupl_v1.2_en.pdf

2 Release content

For information on the changes in this release, please see the current Release Notes.
The deliverable consists of the following zip files:

Deliverable	Description
EIDAS-2.6.0.zip	Distribution version 2.6.0 of the sample eIDAS-Node
EIDAS-Sources-2.6.0.zip	Source files (Maven project) of the sample eIDAS-Node including an example of implementation of the eIDAS-Node Specific Connector, the eIDAS-Node Specific Proxy Service, demonstration Service Provider (SP) and IdP (Identity Provider).
EIDAS-Binaries-Wildfly-2.6.0.zip	Deployable war files of a preconfigured eIDAS-Node for a WildFly server (including IdP.war, EidasNode.war, SP.war SpecificConnector.war, SpecificProxyService.war)
EIDAS-Binaries-Tomcat-2.6.0.zip	Deployable war files of a preconfigured eIDAS-Node for a Tomcat server (including IdP.war, EidasNode.war, SP.war, SpecificConnector.war, SpecificProxyService.war)
EIDAS-Binaries-Was-2.6.0.zip	Deployable war files of a preconfigured eIDAS-Node for a WebSphere server (including IdP.war, EidasNode.war, SP.war, SpecificConnector.war, SpecificProxyService.war)
EIDAS-Binaries-Wls-2.6.0.zip	Deployable war files of a preconfigured eIDAS-Node for a WebLogic server (including IdP.war, EidasNode.war, SP.war, SpecificConnector.war, SpecificProxyService.war)

3 Overview of the preconfigured demo eIDAS-Node packages

This distribution provides an example configuration in which each supported server represents one country providing an eID service. For the purpose of this demo, fictitious countries are used (CA, CB, CC, CD, CF).

The following table illustrates the setup of the configurations provided with this distribution.

Application Server	Supported version(s)	Default host	Default port	Country	Description
Tomcat	9.0.58	localhost	8080	CA	Country A
GlassFish	No version currently supported	localhost	8081	CB	Country B
WildFly	23.0.2 Final (Servlet-Only Distribution)	localhost	8085	CC	Country C
WebLogic	14.1.1.0.0 (14c)	localhost	7001	CD	Country D
WebSphere Liberty Profile	Liberty 21.0.0.5 (Web Profile 8)	localhost	9080	CF	Country F

*Default build server provided with the binaries

4 Demo eIDAS-Node set up and configuration

Each example eIDAS-Node package is preconfigured to use 'localhost' as hostname and a default http listening port; see the table in section 3.
The http listening port of your application server must be adapted according to these default values.

If you need to change these default values, refer to section *5.1 — Changing the default hostname or http port* for details.

4.1 To set up and configure the demo, perform the following steps:

4.1.1 Configuring the JVM

The project is built by default using the **Java SDK** version **11.0.12**.

4.1.2 Oracle Java JCE Unlimited Strength Jurisdiction Policy

In java 11, to verify the JCE security policy, the "crypto.policy" configuration in "JAVA_HOME/conf/security/java.security" file can be checked. By default, the configuration is "unlimited".

4.1.3 Preparing the environment

1. Copy the server configuration files and keystores provided for testing purposes into the local directories: Open the zip file (config.zip in the EIDAS-Binaries-xxx-yyy.zip[2]) and copy the directory keystore and the directory of the application server as required (i.e. tomcat, wildfly, wls, was) into the configuration directory.
2. Local directory or directories must be defined in order to store the configuration files and the test keystores. These directories need to be defined either as OS/AS environment variables or command-line parameters:
   EIDAS_CONFIG_REPOSITORY for EidasNode
   SPECIFIC_CONNECTOR_CONFIG_REPOSITORY for Specific Connector
   SPECIFIC_PROXY_SERVICE_CONFIG_REPOSITORY for Specific Proxy Service
   SP_CONFIG_REPOSITORY for SP
   IDP_CONFIG_REPOSITORY for IdP
   It is also possible to use only one common directory for all the modules. JVM command line example:

```
 -DEIDAS_CONFIG_REPOSITORY=c: /Pgm/projects/configEidas/tomcat/
 -DSPECIFIC_CONNECTOR_CONFIG_REPOSITORY=c: /Pgm/projects/configEidas/
tomcat/specificConnector/
 -DSPECIFIC_PROXY_SERVICE_CONFIG_REPOSITORY=c: /Pgm/projects/
configEidas/tomcat/specificProxyService/
 -DSP_CONFIG_REPOSITORY=c: /Pgm/projects/configEidas/tomcat/sp/
 -DIDP_CONFIG_REPOSITORY=c: /Pgm/projects/configEidas/tomcat/idp/
```

   By default, the configuration file structure (e.g. tomcat) must be as follows (it is also possible to use only one common directory for all the modules):

[2] http://vs-cis-k2:8088/job/STORK%20PEPS%20Release/lastSuccessfulBuild/artifact/STORK-Peps-Binaries-Tomcat123.zip

```
tomcat/eidas.xml
tomcat/encryptionConf.xml
tomcat/EncryptModule_Connector.xml
tomcat/EncryptModule_Connector_p12.xml
tomcat/EncryptModule_Service.xml
tomcat/EncryptModule_Service_p12.xml
tomcat/saml-engine-additional-attributes.xml
tomcat/igniteNode.xml
tomcat/igniteSpecificCommunication.xml
tomcat/SamlEngine.xml
tomcat/SamlEngine_Connector.xml
tomcat/SamlEngine_Service.xml
tomcat/SignModule_Connector.xml
tomcat/SignModule_Connector_p12.xml
tomcat/SignModule_Connector_EC_p12.xml
tomcat/SignModule_Service.xml
tomcat/SignModule_Service_p12.xml
tomcat/SignModule_Service_EC_p12.xml
tomcat/idp/additional-attributes.xml
tomcat/idp/idp.properties
tomcat/idp/user.properties
tomcat/metadata/MetadataFetcher_Connector.properties
tomcat/metadata/MetadataFetcher_Service.properties
tomcat/sp/additional-attributes.xml
tomcat/sp/sp.properties
tomcat/specificConnector/additional-attributes.xml
tomcat/specificConnector/eidas-attributes.xml
tomcat/specificConnector/specificCommunicationDefinitionConnector.xml
tomcat/specificConnector/specificConnector.xml
tomcat/specificProxyService/additional-attributes.xml
tomcat/specificProxyService/eidas-attributes.xml
tomcat/specificProxyService/
specificCommunicationDefinitionProxyservice.xml
tomcat/specificProxyService/specificProxyService.xml
keystore/eidasKeyStore.jks
keystore/eidasKeyStore_Connector_CA.jks
keystore/eidasKeyStore_Connector_CB.jks
keystore/eidasKeyStore_Connector_CC.jks
keystore/eidasKeyStore_Connector_CD.jks
keystore/eidasKeyStore_Connector_CF.jks
keystore/eidasKeyStore_IDP_CA.jks
keystore/eidasKeyStore_IDP_CB.jks
```

```
keystore/eidasKeyStore_IDP_CC.jks
keystore/eidasKeyStore_IDP_CD.jks
keystore/eidasKeyStore_IDP_CF.jks
keystore/eidasKeyStore_METADATA.jks
keystore/eidasKeyStore_Service_CA.jks
keystore/eidasKeyStore_Service_CB.jks
keystore/eidasKeyStore_Service_CC.jks
keystore/eidasKeyStore_Service_CD.jks
keystore/eidasKeyStore_Service_CF.jks
keystore/eidasKeyStore_SP_CA.jks
keystore/eidasKeyStore_SP_CB.jks
keystore/eidasKeyStore_SP_CC.jks
keystore/eidasKeyStore_SP_CD.jks
keystore/eidasKeyStore_SP_CF.jks
keystore/eidasKeyStore.p12
keystore/eidasKeyStore_METADATA.p12
keystore/eidasKeyStore_METADATA_EC.p12
keystore/eidasKeyStore_METADATA_TC.p12
keystore/eidasKeyStore_METADATA_TC_EC.p12
```

Please note: all components in the binary distribution are preconfigured for the file system layout indicated above. Deviating from this layout will require changes to the configurations of the individual modules. Please refer to the *eIDAS-Node Installation and Configuration Guide* for more details

4.1.4 Installing a security provider

In order for the Node to be able to sign, verify and encrypt/decrypt data, one or more security providers need to be installed/configured that can perform those operations. In most cases this means that Bouncy Castle provider will have to be always installed/configured, since it provides operations others do not. However, this might be avoided if, for some reason, another security provider, that is installed/configured, performs all necessary cryptographic functionalities required by eIDAS technical specifications. Therefore, BouncyCastle is not installed directly through the code to allow this flexibility.

4.1.4.1 PKCS12 or JKS Software Keystores with BouncyCastle

When using classic software keystores such as "PKCS12" or "JKS", the BouncyCastle provider will be needed, and to install the BouncyCastle provider via the `java.security` file (which can be found in the `$JAVA_HOME/conf/security` folder).

Note that in Java 11, additional to defining the provider in the java.security file, we have to define two java options (`JAVA_OPTS`) in order to actually use the provider :

- *--module-path* to indicate the location/path of the provider jar
- *--add-modules* to add the module corresponding to the provider (defined by java 9 standards in module-info file present in the jar)

For example, to install the BouncyCastle provider in Java 11, we would have to :

1. Locate and open in a text editor the file in the `$JAVA_HOME/conf/security` folder

2. Add a line to the `java.security` file:
   security.provider.N=org.bouncycastle.jce.provider.BouncyCastleProvider
   (you should set N according to your config, to the next available index in the list of providers).
3. Add the following java options:

```
export JAVA_OPTS= "$JAVA_OPTS --module-path /path/to/library/bcprov-jdk15on-170.jar"
export JAVA_OPTS= "$JAVA_OPTS --add-modules org.bouncycastle.provider"
```

4.1.4.2 PKCS11

Please refer to the *eIDAS-Node Installation and Configuration Guide* for more details in the sections : "*3.1.4.2. PKCS11*"

4.1.5 Configuring the application server

It is necessary to increase the default JVM memory settings. Set the following JVM parameter in the startup script of your application server :
-Xms512m -Xmx1024m

4.1.5.1 On Wildfly 23.0.2 Final Server (Servlet-Only Distribution)

In order for the node, to work correctly with Wildfly, a module for BouncyCastle has to configured, therefore :

- Replace if not done before all files and folders contained in ${WILDFLY_SERVER_HOME}/modules/system/layers/base/org/bouncycastle/main/ files by the ones in /AdditionalFiles/Wildfly23/org/bouncycastle/main/ found in the release.

(Note the bcprov-jdk15on-170.jar is a signed version obtained from https://www.bouncycastle.org/download/bcprov-jdk15on-170.jar)

4.1.5.2 On WebSphere Liberty Profile the following features should be enabled:

Websphere server feature

Webprofile feature must be disabled for Websphere Liberty Profile from server.xml to avoid error "Exception occurred during processing request".

Therefore, in the file "${SERVER_HOME}/usr/servers/${SERVER}/server.xml", comment the "webProfile-8.0" feature line, and add the "jsp-2.3" feature as in the example below :

```
< featureManager >
    < feature >jsp-2.3</ feature >
</ featureManager >
```

Websphere additional configuration

1. Add property <webContainer com.ibm.ws.webcontainer.enableErrorExceptionTypeFirst="true"/> to $SERVER_HOME/usr/servers/defaultServer/server.xml file.
2. Add this property in server.xml file of websphere:
   <webContainer com.ibm.ws.webcontainer.redirectcontextroot="true"/>
   If set to true, and a request is made to the context root of an application with a missing trailing slash, the WebContainer appends the trailing slash. The WebContainer redirects to the URL with the appended slash before it applies any servlet filters defined in the application.

4.1.5.3 Deploy the applications according to your application server.

- EidasNode.war
- SP.war
- IdP.war
- SpecificConnector.war
- SpecificProxyService.war

In order to deploy the project, after the build is complete, copy the artefact (EIDAS-Node/target/EidasNode.war) to the deploy folder of the Server.

Deployment of EidasNode on WebSphere

Copy the war files in ${WEBSPHERE_SERVER_HOME}/usr/servers/${SERVER}/dropins/ and start the server.

4.1.5.4 Run and validate the installation

You now have a Service Provider, Specific-Connector ,eIDAS-Node Connector, eIDAS-Node Proxy Service , Specific-ProxyService and an IdP configured to run on localhost:

- Tomcat: http://localhost:8080/SP
- Wildfly: http://localhost:8085/SP
- WebLogic: http://localhost:7001/SP
- WebSphere, WebSphere Liberty Profile: http://localhost:9080/SP

To validate the installation, a first test can be performed simulating that a citizen from a country accesses services in the same country.

1. Open the Service Provider URL : http://localhost:defaultport/SP[3]
2. Choose for both the SP and citizen country the fictitious country for which your application server has been configured (CA, CB, CC, CD or CF).
3. The generated Simple Protocol Request is displayed. Submit the form.
4. Click **Next** to give your consent to attributes being transferred.
5. Enter the user credentials. Type 'xavi' as **Username** and 'creus' as **Password** and submit the page.
6. Click **Submit** to validate the values to transfer. The Simple Protocol Response is displayed.
7. **Submit** the form.

You should see **Login Succeeded**.

[3] http://localhostdefaultport

5 Specific configuration

5.1 Changing the default hostname or http port

The parameters below can be adapted to reflect your configuration.

Note: The application server must be restarted after changes have been made.

5.1.1 eIDAS-Node hostname and port

1. Edit the file eidas.xml located in the configuration directory as shown below.

Property	**Value**
connector.assertion.url	http://<connector.*yourHostname*>:<connector.*yourPort*>/ EidasNode/ColleagueResponse
connector.metadata.url	http://<connector.*yourHostname*>:<connector.*yourPort*>/ EidasNode/ConnectorMetadata
service.metadata.url	http://<service.*yourHostname*>:<servicer.*yourPort*>/ EidasNode/ServiceMetadata
ssos.serviceMetadataGenerator.post.location	The URL for the metadata <md:SingleSignOnService> location attribute of the SingleSignOnService related to Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST". e.g. http:// <service.*yourHostname*>:<service.*yourPort*>/ EidasNode/ ColleagueRequest/
ssos.serviceMetadataGeneratorIDP.redirect.location	The URL for the metadata <md:SingleSignOnService> location attribute of the SingleSignOnService related to Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect". e. g. http:// <*service.yourHostname*>:<*service.yourPort*>/EidasNode/ ColleagueRequest/
specific.proxyservice.request.receiver	The URL for specific-proxyService requests receiver only used when specific proxy service is build/deployed as WAR http://<specific ProxyService.*yourHostname*>:<specific ProxyService.*yourPort*>/SpecificProxyService/ ProxyServiceRequest

Property	**Value**
specific.connector.response.receiver	The URL for specific-connector response receiver used when specific connector is build/deployed as WAR http://<*specific Connector.ourHostname*>:<*specific Connector.yourPort*>/SpecificConnector/ ConnectorResponse
security.header.CSP.report.uri	Prefix for **report_uri** header populated by the node http://<service.*yourHostname*>*:*<service.*yourPort*>/ EidasNode

2. Open and edit the file sp.properties as shown below.

Property	**Value**
country1.url	http://<connector.*yourHostname*>:<connector.yourPort>/ EidasNode/ServiceProvider

5.1.2 SP hostname and port

Open and edit the file sp.properties as shown below.

Property	**Value**
sp.return	http://<sp.*yourHostname*>:<sp.*yourPort*>/SP/ReturnPage

Open and edit the file /specificConnector/specificConnector.xml as shown below.

Property	**Value**
specific.connector.request.url	http://<connector.*yourHostname*>:< connector.*yourPort*>/EidasNode/ SpecificConnectorRequest

5.1.3 Identiy provider (IdP) hostname and port

Edit the file /specificProxyService/specificProxyService.xml located in the configuration folder as shown below.

Property	**Value**
idp.url	http://<idp.*yourHostname*>:<idp.*yourPort*>/IdP/ AuthenticateCitizen

Property	**Value**
specific.proxyservice.idp.response.service.url	http://<specific ProxyService.*yourHostname*>:<specific ProxyService.*yourPort*>/SpecificProxyService/IdpResponse
specific.proxyservice.response.url	http://<service.*yourHostname*>:< service.*yourPort*>/EidasNode/SpecificProxyServiceResponse

5.2 Changing the keystore location

By default, the test keystores are located in the directory 'keystore' in the same directory as the configuration directory. You can change these values by editing the files below to reflect your configuration. All filenames and path information are relative to the configuration directory for the given module.

Keystore	**Files**
eIDAS-Node	SignModule_Service.xml SignModule_Connector.xml EncryptModule_Service.xml EncryptModule_Connector.xml

5.3 Changing keystore configuration

By default, the preconfigured eIDAS components use the following extended configuration.

Since the 2.6 release, the JKS keystores are using a proprietary format. It was decided to migrate the keystores to the PKCS12 industry format since it is an standard format, both formats keystores "JKS" and "PKCS12" will be provided in the same directory.

5.3.1 Extended configuration

In this configuration, all stakeholders (Connector /Proxy Service) use their own certificate for the signing and encrypting of SAML messages.

This setup is close to a real-life scenario, where the components are distributed across servers and Member States.

Example for country 'CA':

	Keystore		Certificate	Country
Connector	eidasKeyStore_Connector_CA.jks (SignModule_Connector.xml, EncryptModule_Connector.xml)	Key Pair	Connector-ca-demo-certificate	CA
		Trusted	Metadata (signing certificate)	CA
Proxy Service	eidasKeyStore_Service_CA.jks (SignModule_Service.xml, EncryptModule_Service.xml)	Key Pair	Service-ca-demo-certificate	CA
		Trusted	Metadata (signing certificate)	CA
Metadata	eidasKeyStore_METADATA.jks	Key Pair	Metadata (signing certificate)	CA

5.3.2 Basic configuration

In this configuration, all stakeholders share the same certificate.

This setup is a simplified scenario for a lab environment, but corresponds less to a real-life situation.

In order to set up the basic scenario, all SignModule configuration files should be adapted to reference the common test keystore, eidasKeyStore "JKS" or "PKCS12" format.

5.3.2.1 Setting up your Keystore "PKCS12"

Copy your *eidasKeystore.p12* (the key store with your eIDAS-Node keys, alternatively you can use the example key store provided with the application) into a directory of your own choice, and make sure that:

- the property *keyStorePath* on file:$EIDAS_CONFIG_REPOSITORY/SignModule_Service.xml reflects the relative location of your Proxy Service *eidasKeyStore.p12*.
- the property *keyStorePath* on file:$EIDAS_CONFIG_REPOSITORY/SignModule_Connector.xml reflects the relative location of your eIDAS-Node Connector eidasKeyStore.p12.

If the eIDAS-Node is configured to use encryption (essential in the production environment), also ensure that:

- the property *keyStorePath* on file:$EIDAS_CONFIG_REPOSITORY/EncryptModule_Service.xml reflects the relative location of your Proxy Service eidasKeyStore.p12.
- the property *keyStorePath* on file:$EIDAS_CONFIG_REPOSITORY/EncryptModule_Connector.xml reflects the relative location of your eIDAS-Node Connector eidasKeyStore.p12.

For more information see the *eID eIDAS-Node and SAML* manual.

5.4 eIDAS-Node compliance and Recommended configuration

For validation purposes, the demo eIDAS Nodes do not use HTTPS and the configuration parameters are set as shown below.

Property	**Test Value**	**Recommended value (EIDAS-CONFIG)**	
metadata.restrict.http	false	default(true)	Metadata must be only available via HTTPS
check.certificate.validity.period	false	default(true)	Do not allow expired certificates
keyStoreType	JKS	PKCS12	Cx certificates (perimed)
signature.algorithm	http://www.w3.org/2007/05/xmldsig-more#sha512-rsa-MGF1	http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha512	
metadata.signature.algorithm	http://www.w3.org/2007/05/xmldsig-more#sha512-rsa-MGF1	http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha512	

For more information see the paragraphe *7.4. eIDAS-Node compliance* in the *eIDAS Node Installation and configuration Guide*.

6 Compiling the modules from the source

If you need to rebuild the Maven project, open EIDAS-Parent and execute the Maven commands described in the table below according to your application server.
At a command prompt, first navigate to the folder indicated (EIDAS-Parent) and then enter the corresponding command line.

Recommended versions of Maven are 3.5.4 and above. Lower versions can result in exceptions.

Folder	**Command line**	
EIDAS-Parent	Tomcat/ WebSphere	mvn clean install -PNodeOnly[,DemoToolsOnly] –PnodeJcacheIgnite[,specificCommunicationJcacheIgnite]
	WebLogic	mvn clean install -P weblogic -PNodeOnly[,DemoToolsOnly] –PnodeJcacheIgnite[,specificCommunicationJcacheIgnite]
	Wildfly	mvn clean install -P wildfly -PNodeOnly[,DemoToolsOnly] –PnodeJcacheIgnite[,specificCommunicationJcacheIgnite]

In order to deploy the project, after the build is complete, copy the artifacts needed IdP.war, SP.war, SpecificConnector.war and SpecificProxyService.war to the deploy folder of the Server. The EidasNode.war may also be needed check eIDAS-Node Installation and configuration guide for more information.

7 Enabling logging

The locations of the audit files are by default configured to use a Java system properties variable called *LOG_HOME*.

A value can be assigned to this variable by using: *-DLOG_HOME="<myDirectoryName>"* at server start-up.

Additionally, the logging of the exchanged messages within the eIDAS Node and between eIDAS Node and the Specific could be enabled by setting the property saml.audit from eidas.xml configuration file to true.

Note: The eIDAS-Node logs may contain person identification data, hence these logs should be handled and protected appropriately in accordance with the European privacy regulations [Dir. 95/46/EC] and [Reg. 2016/679].

[Reg. 2016/679] REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC.

[Dir. 95/46/EC] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data.