

eIDAS Middleware

User Guide



eIDAS Middleware, Prototype

© 2016 Governikus GmbH & Co. KG

Table of contents

[1 Introduction 3](#_Toc453932521)

[1.1 Conventions in this document 3](#_Toc453932522)

[2 Prerequisites for the virtual machine 4](#_Toc453932523)

[2.1 Configuring the virtual machine 4](#_Toc453932524)

[2.2 RAM capacity 4](#_Toc453932525)

[2.3 CPU 4](#_Toc453932526)

[2.4 Disk / storage capacity 4](#_Toc453932527)

[3 Configuration of the operating system 5](#_Toc453932528)

[3.1 First time login via console 5](#_Toc453932529)

[3.2 Regenerate the ssh server key 5](#_Toc453932530)

[3.3 Setting up network access 5](#_Toc453932531)

[3.4 DNS configuration 6](#_Toc453932532)

[3.5 Firewall configuration 6](#_Toc453932533)

[4 Configuration of the middleware application 7](#_Toc453932534)

[4.1 Overview 7](#_Toc453932535)

[4.2 Prepare for installation 7](#_Toc453932536)

[4.3 Using the installer (recommended) 8](#_Toc453932537)

[4.3.1 Installation 8](#_Toc453932538)

[4.3.2 Installer example output 11](#_Toc453932539)

[4.3.3 After installation 12](#_Toc453932540)

[4.4 Manual installation 12](#_Toc453932541)

[4.4.1 Setup your JRE: Install the unlimited strength cryptography 12](#_Toc453932542)

[4.4.2 Prepare WildFly 13](#_Toc453932543)

[4.4.3 Setup the poseidas.xml 14](#_Toc453932544)

[4.4.4 Setting up the EU-Middleware 15](#_Toc453932545)

[5 Operating the server 17](#_Toc453932546)

[5.1 First time startup 17](#_Toc453932547)

[5.2 Startup and shutdown 17](#_Toc453932548)

[5.3 Monitoring 17](#_Toc453932549)

[5.3.1 Health Monitoring 17](#_Toc453932550)

[5.3.2 Log files 17](#_Toc453932551)

[5.3.3 Scaleability 17](#_Toc453932552)

[6 Common pitfalls 18](#_Toc453932553)

[6.1 Installation outside the virtual machine 18](#_Toc453932554)

[6.2 The AusweisApp2 is complaining 18](#_Toc453932555)

[6.3 Port 9444 is closed 18](#_Toc453932556)

[6.4 Keystore alias 18](#_Toc453932557)

[6.5 H2 DB cannot be created 18](#_Toc453932558)

[6.6 SubjectURL in the authorization certificate matches the URL for the tcToken. 18](#_Toc453932559)

[6.7 Private key / Certificate mismatch 18](#_Toc453932560)

[7 Glossary 20](#_Toc453932561)

# Introduction

You have received this virtual machine in order to setup the needed infrastructure to support the German ID card via your eIDAS connector. This virtual machine acts as an implementation of the middleware-based approach described in the technical specifications which can be found on the [collaborative platform created by the European Commission](https://joinup.ec.europa.eu/software/cefeid/document/eidas-technical-specifications-v10)

This virtual machine mainly consists of a debian operating system and a preconfigured WildFly application server in which the middleware web applications are deployed.

This document guides you through the installation and configuration process.

## Support and service

The Federal Office for Information Security (BSI) is accountable for support and service of the software. However, Governikus GmbH & Co. KG will provide support and service on behalf of the Federal Office for Information Security. You can email requests to [eidas-middleware@governikus.de](mailto:eidas-middleware@governikus.de)

Please note that the Federal Office for Information Security will be able to read those emails in order to take responsibility for their obligation.

## Conventions in this document

The following typographical conventions are used in this user guide:

**Italic**

Indicate URLs, email addresses, filenames, passwords and file extensions.

**Constant width**

Used for file and output listings, as well as within paragraphs to refer to file elements.

**Constant width bold**

Shows commands or other text that should be typed in literally by the user.

**Constant width italic**

Shows text that should be replaced with user-supplied values or by values determined by context.

**Syntax highlighted text**

This is mainly used for xml-file contents for better readability.

**Tables with icons**

|  |  |
| --- | --- |
| Note | This is a hint. Something you should be aware of. |

|  |  |
| --- | --- |
| Attention | This is a warning. You should really take this text into account. You might lose support or damage something if you ignore warnings in a table like this. |

# Prerequisites for the virtual machine

This section describes and recommends the setup of the virtual hardware that is required for operating the eIDAS middleware. You received an image of a virtual machine (Open Virtualization Format). This image can be booted in a compatible environment.

## Configuring the virtual machine

When operated on a vmware host please refer to [Best Practices Guide](http://www.vmware.com/files/pdf/techpaper/Enterprise-Java-Applications-on-VMware-Best-Practices-Guide.pdf).

## RAM capacity

A minimum of at least 2 GB RAM is required for the virtual machine but it is recommended to provide 8 GB.

## CPU

It is recommenced to equip the virtual machine with two CPU with at least 2 GHz each.

## Disk / storage capacity

The disk should provide at least 50 GB storage capacity.

# Configuration of the operating system

This section illustrates the requirements for the operational environment, including network configuration, DNS configuration and firewall.

## First time login via console

In order to configure the network you have to login via the console first. Use *governikus* as username and *Pleasechangeme!* as password. To change the system setting you will have to use the sudo command. Please reboot the system after configuration according to your environment and login via ssh.

## Regenerate the ssh server key

You should use an unique ssh server key. Please generate the server keys prior to first time use. To generate new server keys execute the following commands

**sudo dpkg-reconfigure openssh-server**

**sudo /etc/init.d/ssh restart**

## Setting up network access

The network configuration is done in this file: */etc/network/interfaces*

The default is configured to use dhcp. It is recommended to use a static configuration in your environment. The file looks like:

# This file describes the network interfaces available on your system

# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/\*

# The loopback network interface

auto lo

iface lo inet loopback

# The primary network interface

allow-hotplug eth0

iface eth0 inet dhcp

#iface eth0 inet static

# address 1.1.1.2

# netmask 255.255.255.0

# gateway 1.1.1.1

pre-up /bin/sh /etc/firewall/fire.sh

Change the following with a text editor like vi to your own values.

* **iface**: change the value dhcp to static.
* **address**: The IP address of this server.
* **netmask**: The netmask of the used network.
* **gateway**: The IP of the default gateway in this network segment.

## DNS configuration

The DNS configuration is done in this file */etc/resolv.conf*

The default values will probably not work in your environment! Change the following with a text editor like vi to your own values.

* **domain**: The name of your network domain, or if not used, comment it with “#” at the beginning of the line out.
* **search**: The name of your network domain, or if not used, comment it with “#” at the beginning of the line out.
* **nameserver**: The IP address of the first name server.
* **nameserver**: The IP address of the second name server.

## Firewall configuration

The firewall is preconfigured, all incoming connections, except the pre-configured, are denied. The settings can be found in this file */etc/firewall/firewall.sh*

Pre-configured Ports:

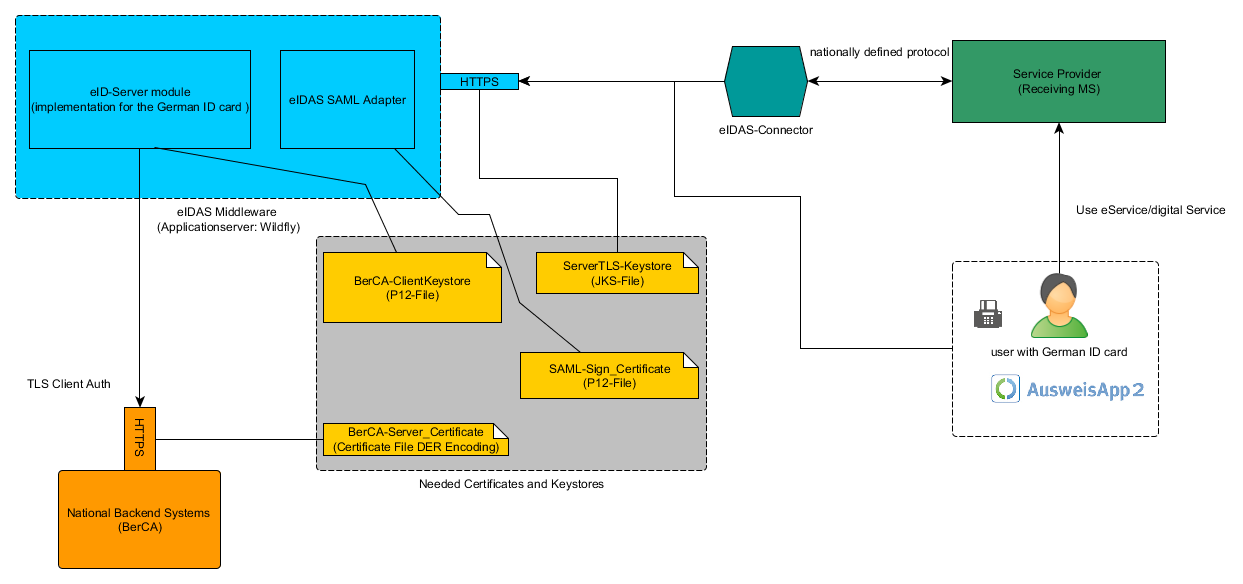
* **ssh**: (TCP 22)
* **https**: (TCP 443)
* **snmp**: (UDP 161)

|  |  |
| --- | --- |
| Note | **Hint**:  Any outgoing connection is allowed. |

# Configuration of the middleware application

## Overview

In order to setup the EU middleware you will need to provide some certificates. The following overview illustrates the required certificates and keystores.



## Prepare for installation

The installer will setup your JRE, WildFly, eID-Service and the EU-Middleware.

|  |  |
| --- | --- |
| Note | **Hint**:  Be sure to have the required certificates and keystores at hand! |

The following table describes the required *keystores*:

|  |  |  |
| --- | --- | --- |
| **Keystore-Name** | **Format** | **Desciption** |
| BerCA-Client-Keystore | PKCS#12 | This keystore is needed to access the German eID PKI. |
| ServerTLS | JKS | This keystore is used to setup the https-port of the server. |
| SAML-message | PKCS#12 | This keystore is used for signing the SAML-messages. |

Make sure to have a metadata.xml from your service provider.

|  |  |
| --- | --- |
| Note | **Note**: The certificate for the Server TLS must use a public key with a key length of at least 2048 Bit. A compatible certificate could be created with the command  *<PATH\_TO\_JAVA>/keytool -genkey -keystore /home/governikus/tls-keystore2.jks -keysize 2048 -sigalg SHA256withRSA -keyalg RSA* |

## Using the installer (recommended)

Stop your WildFly (see Startup and Shutdown ) then run the Installer.jar from the directory */home/governikus/Installer* with the command

**java -jar EU-Middleware-Installer.jar**

and follow the installation steps. After finishing the installer just copy the service provider *metadata.xml* to the *[wildfly]/standalone/configuration/serviceprovider* folder. After that start the WildFly and the EU middleware will be available. Now log in and follow these steps:

**cd Installer/**

**java -jar EU-Middleware-Installer.jar**

|  |  |
| --- | --- |
| Note | **Note**:  Please note that the installer will modify the *jaxb.properties* and security settings of your JRE if necessary. |

### Installation

The installer will ask you for input. This section describes the expected values.

Do you want to (re-)install via existing installation configs? (y/n)

The installer can save the input from a previous installation. Here you can answer y if you already created an installation configuration, otherwise answer n.

Please enter the URL to your eID-Server (Protocol, Hostname, Port. e.g.: https://hostname:8443)

The middleware will be available under a specific URL. Please type the designated address for the server.

Please enter a unique entity ID for the client

The unique entity is required for the SAML protocol please type a string (URI) which is unique to this middleware. This ID is used for the metadata.

Please enter an absolute path to your BerCA-Client-Keystore (in a PKCS #12 format)

The BerCA-Client-Keystore is required to access the German eID PKI. If you receive this keystore or if you can create it on your own depends on installation purpose. Please refer to the organisation you received this middleware from. In any case enter the full path to the keystore here.

File found. Please enter the PIN to the keystore.

Enter the PIN for this keystore.

Please enter the absolute path to your ServerTLS-Keystore (in JKS format)

The middleware will be available via an https-connection. Please enter the full path to the keystore file here. If you only have a pkcs#12 keystore you can convert it using the keytool command. See also converting keystores between JKS and P12.

File found. Please enter the PIN to the keystore

Enter the PIN for this keystore.

Please enter the alias of the certificate to install

Keystores can contain more private keys and certificates. Please enter the alias for the keypair here.

Please enter the absolute path to the keystore the SAML-message should be signed and encrypted with (in a PKCS #12 format)

The SAML response will be signed by the middleware. Please insert the full path to the keystore here.

File found. Please enter the PIN to the keystore

Enter the PIN for this keystore.

Please enter the alias of the certificate to install

Keystores can contain more private keys and certificates. Please enter the alias for the keypair here.

Please enter your company name

This is your company name. This value gets exposed via the metadata of the server (optional).

Please enter your organisation's name

Enter the name of the organization for which the server is operated. This value gets exposed via the metadata of the server (optional).

Please enter your organisation's web address

This value gets exposed via the metadata of the server (optional).

Please enter the two-letter abbreviation for your country (e.g.: DE)

This value gets exposed via the metadata of the server (optional).

Please enter the contact person's E-Mail address

This value gets exposed via the metadata of the server (optional).

Please enter the contact person's first name

This value gets exposed via the metadata of the server (optional).

Please enter the contact person's surname

This value gets exposed via the metadata of the server (optional).

Please enter the contact person's telephone number

This value gets exposed via the metadata of the server (optional).

Please enter the contact person's address

This value gets exposed via the metadata of the server (optional).

### Installer example output

java -jar EU-Middleware-Installer.jar

---------------------------------------

| Installation tool for EU-Middleware |

---------------------------------------

Type 'exit' at any time to abort the installation

Checking for Zulu JDK and JBoss Wildfly:

Zulu JDK found at /opt/middelware/java

JBoss Wildfly found at /opt/middelware/wildfly

Do you want to (re-)install via existing installation configs? (y/n)

n

Continuing without installation configs.

Please enter the URL to your eID-Server

(Protocol, Hostname, Port. e.g.: https://hostname:8443)

https://test.de:8443

Please enter a unique entity ID for the client

eidas-tst-de

Please enter an absolute path to your BerCA-Client-Keystore

(in a PKCS #12 format)

/home/governikus/demo\_1415.p12

File found. Please enter the PIN to the keystore.

Please enter the absolute path to your ServerTLS-Keystore

(in JKS format)

/home/governikus/demo\_123456.jks

File found. Please enter the PIN to the keystore

Please enter the alias of the certificate to install

mykey

Please enter the absolute path to the keystore the SAML-message

should be signed and encrypted with (in a PKCS #12 format)

/home/governikus/demo\_1415.p12

File found. Please enter the PIN to the keystore

Please enter the alias of the certificate to install

hartje bruns (governikus kg)

- Information about your company and the contact person -

Please enter your company name

Governikus GmbH

Please enter your organisation's name

Dito

Please enter your organisation's web address

www.governikus.de

Please enter the two-letter abbreviation for your country (e.g.: DE)

DE

Please enter the contact person's E-Mail address

hartje.bruns@governikus.de

Please enter the contact person's first name

Hartje

Please enter the contact person's surname

Bruns

Please enter the contact person's telephone number

Please enter the contact person's address

Installing...

Done

Do you want to create installation configs to reinstall

EU-Middleware with the same parameters? (y/n)

y

Please enter an absolute path to a folder to save the config files to

/home/governikus/

Saving configs to /home/governikus/

Please start JBOSS Wildfly as root now.

governikus@eu-middelware:~/Installer$

### After installation

After the middleware was configured successfully follow the steps described in section [First time startup](#_First_time_Startup).

## Manual installation

In the following sections you can see how the middleware is configured manually. These steps are automatically executed when you use the installer.

|  |  |
| --- | --- |
| Attention | **Attention**: It is highly recommended to use the installer. |

### Setup your JRE: Install the unlimited strength cryptography

1. Download the files from Orcale.com e.g <http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html>
2. Unzip the downloaded file and copy *local\_policy.jar* and *US\_export\_policy.jar* to the directory *$jre/lib/security*
3. Create or modify the *jaxb.properties*. The *jaxb.properties* have to include this entry: javax.xml.accessExternalSchema = all
4. Modify the security provider settings: Add to your *$jre/lib/security/java.security* file this entry security.provider.11=de.governikus.psksun.com.sun.net.ssl.internal.ssl.Provider
5. Add the *psk\_jsse.jar*, you find it in the res folder, to the directory *$jre/lib/ext*

### Prepare WildFly

1. Setup the Wildfly *standalone.conf* located in the *$wildfly/bin* folder:   
   Add this entry *-Djdk.tls.ephemeralDHKeySize=2048* to JAVA\_OPTS
2. Setup the WildFly *standalone.xml* located in the *$wildfly/standalone/configuration* folder. Add the SSL Keystores for HTTPS communication see the following example:

<security-realm name="https">

<server-identities>

<ssl>

<keystore path="ssl.jks"

relative-to="jboss.server.config.dir"

keystore-password="123456"

alias="localhost"

key-password="123456"/>

</ssl>

</server-identities>

</security-realm>

<security-realm name="httpsca">

<server-identities>

<ssl>

<keystore path="ssl.jks"

relative-to="jboss.server.config.dir"

keystore-password="123456"

alias="localhost"

key-password="123456"/>

</ssl>

</server-identities>

<authentication>

<truststore path="ssl.jks"

relative-to="jboss.server.config.dir"

keystore-password="123456"/>

</authentication>

</security-realm>

<security-realm name="httpspsk">

<server-identities>

<ssl protocol="TLSPSK">

<keystore path="ssl.jks"

relative-to="jboss.server.config.dir"

keystore-password="123456"

alias="localhost"

key-password="123456"/>

</ssl>

</server-identities>

</security-realm>

In the subsystem undertow add the https-listener:

<https-listener

name="https"

socket-binding="https"

security-realm="https"

enabled-cipher-suites="TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256,

TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256,

TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA384,

TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384,

TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256,

TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256,

TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA384,

TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384,

TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256,

TLS\_DHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256,

TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA256,

TLS\_DHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384"/>

<https-listener

name="httpsca"

socket-binding="httpsca"

security-realm="httpsca"

verify-client="REQUIRED"/>

<https-listener

name="httpspsk"

socket-binding="httpspsk"

security-realm="httpspsk" />

In the subsystem socket-binding-group add:

<socket-binding name="https" port="${jboss.https.port:8443}"/>

<socket-binding name="httpsca" port="${jboss.https.port:8444}"/>

<socket-binding name="httpspsk" port="${jboss.https.port:8445}"/>

In the subsystem data sources add this:

<datasource jta="true" jndi-name="java:/jdbc/eID\_Server"

pool-name="jdbc/eID\_Server" enabled="true" use-ccm="true">

<connection-url>

jdbc:h2:~/POSeIDAS-db.h2;DB\_CLOSE\_DELAY=-1;DB\_CLOSE\_ON\_EXIT=FALSE

</connection-url>

<driver-class>org.h2.Driver</driver-class>

<driver>h2</driver>

<security>

<user-name>sa</user-name>

<password>sa</password>

</security>

</datasource>

### Setup the poseidas.xml

Use this template and fill in the missing parts marked with []. Then copy the file to the TODO *$wildfly/standalone/configuration* folder.

**<?xml version="1.0" encoding="UTF-8" standalone="yes"?>**

<CoreConfiguration xmlns="http:/www.bos\_bremen.de/2009/06/eID-Server-CoreConfig">

<ServerUrl>

[SERVER URL e.g. https://localhost:443/eu-middleware]

</ServerUrl>

<sessionManagerUsesDatabase>true</sessionManagerUsesDatabase>

<sessionMaxPendingRequests>500</sessionMaxPendingRequests>

<certificateWarningMargin>200</certificateWarningMargin>

<TimerConfiguration>

<certRenewal length="2" unit="11"/>

<blacklistRenewal length="2" unit="11"/>

<masterAndDefectListRenewal length="2" unit="11"/>

</TimerConfiguration>

<ServiceProvider entityID="provider1" enabled="true">

<EPAConnectorConfiguration updateCVC="true">

<CVCRefID>provider1</CVCRefID>

<PkiConnectorConfiguration>

<blackListTrustAnchor>

for better readability this certificate is excluded.

See template on the virtual machine

</blackListTrustAnchor>

<masterListTrustAnchor>

for better readability this certificate is excluded.

See template on the virtual machine

</masterListTrustAnchor>

<defectListTrustAnchor>

for better readability this certificate is excluded.

See template on the virtual machine

</defectListTrustAnchor>

<policyImplementationId>govDvca</policyImplementationId>

<sslKeys id="testDVCA">

<serverCertificate>

for better readability this certificate is excluded.

See template on the virtual machine

</serverCertificate>

<clientCertificate>

[BERCA Client Key as BASE64]

</clientCertificate>

<clientKey>

[BERCA Client Cert as BASE64]

</clientKey>

</sslKeys>

<terminalAuthService sslKeysId="testDVCA">

<url>https://dev.governikus-eid.de:9444/gov\_dvca/ta-service</url>

</terminalAuthService>

<restrictedIdService sslKeysId="testDVCA">

<url>https://dev.governikus-eid.de:9444/gov\_dvca/ri-service</url>

</restrictedIdService>

<passiveAuthService sslKeysId="testDVCA">

<url>https://dev.governikus-eid.de:9444/gov\_dvca/pa-service</url>

</passiveAuthService>

<dvcaCertDescriptionService sslKeysId="testDVCA">

<url>https://dev.governikus-eid.de:9444/gov\_dvca/certDesc-service</url>

</dvcaCertDescriptionService>

</PkiConnectorConfiguration>

<PaosReceiverURL>

[PaosReceiverURL e.g. <https://localhost:443/eu-middleware/paosreceiver> - please note you need the port number]

</PaosReceiverURL>

<hoursRefreshCVCBeforeExpires>1</hoursRefreshCVCBeforeExpires>

</EPAConnectorConfiguration>

</ServiceProvider>

</CoreConfiguration>

Please note that the section <PkiConnectorConfiguration> must be filled in collaboration with the institution running the PKI of your choice. They provide the trust anchor certificates and the server certificate and might have conditions on what client certificate you can use. The example is configured with the values currently in use at the Governikus Test PKI.

### Setting up the EU-Middleware

1. Create a folder where the eIDAS connector *metadata.xml* is located.
2. Create a subfolder located in the *$wildfly/standalone/configuration* directory called *eu-connector-files*
3. Setup the *eu-middleware.properties* (see the following example)

Use this template and configure it with your specific values, then copy the file to the folder

TODO

*$wildfly/standalone/configuration*

#Please Note: Use absolute filepath

#PATH to the folder with the service provider configs

SERVICE\_PROVIDER\_CONFIG\_FOLDER=/home/governikus/serviceprovider\_configs

#PATH to the certificate the metadata is signed with

SERVICE\_PROVIDER\_METADATA\_SIGNATURE\_CERT=/home/governikus/metasigner.crt

#EntityID of the corresponding service provider in POSeIDAS.xml

ENTITYID\_INT=provider1

#Path to the keystore containing the middleware signature keypair

SERVICE\_PROVIDER\_SIGN\_KEY=/home/governikus/eu\_middleware\_proxy\_conf\_files/keystore.p12

#Pin of the signature key

SERVICE\_PROVIDER\_SIGN\_PIN=123456

#Alias of the signature key

SERVICE\_PROVIDER\_SIGN\_ALIAS=alias

#Path to the keystore containing the middleware crypto keypair

SERVICE\_PROVIDER\_CRYPT\_KEY=/home/governikus/eu\_middleware\_proxy\_conf\_files/keystore.p12

#Pin of the crypto key

SERVICE\_PROVIDER\_CRYPT\_PIN=123456

#Alias of the crypto key

SERVICE\_PROVIDER\_CRYPT\_ALIAS=alias

#the entityid of the eumiddleware service

ENTITYID=eumiddleware

#country where the MW is deployed

COUNTRYCODE=XY

#metadata contact person

CONTACT\_PERSON\_COMPANY=GOVERNIKUS

CONTACT\_PERSON\_EMAIL=andreas.hohnholt@governikus.de

CONTACT\_PERSON\_GIVENNAME=ANDREAS

CONTACT\_PERSON\_SURNAME=HOHNHOLT

CONTACT\_PERSON\_TEL=61

CONTACT\_PERSON\_ADDRESS=Am Fallturm 9

#metadata organization

ORGANIZATION\_DISPLAY\_NAME=GOVERNIKUS

ORGANIZATION\_NAME=GOVERNIKUS

ORGANIZATION\_URL=GOVERNIKUS.de

ORGANIZATION\_LANG=de

1. Add a metadata.xml to the configured service provider config folder (see eu-middleware.properties SERVICE\_PROVIDER\_CONFIG\_FOLDER properties)

# Operating the server

## First time startup

After the middleware was configured successfully (see [Installation](#_Using_the_Installer):) you can start the application server. Afterwards you can access the URL *https://<YOUR\_SERVERURL>/POSeIDAS/cvcList.jsf* to setup the initial connection to the German eID PKI. To enable the middleware at boot time execute the following command as root **update-rc.d wildfly enable**

## Startup and shutdown

The server can be started and rebooted using the standard commands. To start or restart the WildFly server use the following command: **/etc/init.d/wildfly restart** To stop the WildFly server use the following command: **/etc/init.d/wildfly stop**

## Monitoring

### Health Monitoring

SNMP is enabled by default and preconfigured. You can change the snmp setting by editing the file */etc/snmp/snmpd.conf*. You can monitor the health status of the server and the application using the the snmpwalk command. The OID to see the running processes is 1.3.6.1.2.1.25.4.2.1.4 (HOST-RESOURCES-MIB::hrSWRunPath). Check the output of the snmpwalk command if it contains string */opt/middelware/java/bin/java*. You can monitor CPU, network and memory consumption with the usual OID, see [SNMP Documentation](http://www.debianadmin.com/linux-snmp-oids-for-cpumemory-and-disk-statistics.html) for details.

|  |  |
| --- | --- |
| Note | **Note**:  If you want to use Nagios please refer to the [Nagios Website](https://www.nagios.com/solutions/snmp-monitoring/). |

### Log files

The WildFly Logfiles can be found in the following folder */opt/middelware/wildfly/standalone/log/*. The WildFly can also be configured to log to a syslog server. To use a syslog server change the values for *jboss.syslog.server=localhost* in the file */opt/middelware/wildfly/bin/standalone.conf*according to your needs.

### Scaleability

The virtual machine can be scaled reserving more memory (RAM) and/or faster CPU to the virtual machine. In case the memory configuration has changed, the server needs to be restarted. The configuration file */opt/middelware/wildfly/bin/standalone.conf* has to be changed according to the new memory capacity. The relevant java option for the memory configuration is *-Xmx*. The default is *1024m*, please change the amount according to your needs.

# Common pitfalls

This section describes some errors which are likely to happen and how to avoid them.

## Installation outside the virtual machine

This eIDAS middleware is released as a virtual machine. It might work when you copy the application server and the deployed application to another host, but that is not a supported scenario. Please use the software as supposed within the virtual machine.

## The AusweisApp2 is complaining

There are a lot of security measures built into the AusweisApp2. Here are some common issues.

* The hostname of the deployed eIDAS middleware has to match the corresponding subject URL in the authorization certificate.
* All URLs have to use the https scheme otherwise the AusweisApp2 will not accept the connection.

Please contact your eIDAS Middleware Provider for information.

## Port 9444 is closed

Port 9444 has to be open for outgoing messages. Symptoms for a closed Port 9444 are errors in the SSL communication.

## Keystore alias

A Keystore alias has to be defined in the eIDAS middleware configuration. Please check your Keystore for the correct alias.

## H2 DB cannot be created

Check if the WildFly user has a home directory.

## Private key / Certificate mismatch

In the eIDAS middleware configuration you have often to configure a private key and a matching certificate. Many signing/ssl communication errors appear because of a mismatch or wrong configuration.

# Glossary

C

CPU

Central Processing Unit/CPU · 4

D

DNS

Domain Name System · *see https://de.wikipedia.org/wiki/Domain\_Name\_System*

E

eIDAS Middleware

The middleware as described in the eIDAS technical specification. See collaborative platform created by the European Commission · 4

F

firewall

Is a network secutity system · *see https://en.wikipedia.org/wiki/Firewall\_(computing)*

G

german eid pki

The PKI for reading the german eid card · 7, 9, 17

J

JKS · *see https://en.wikipedia.org/wiki/Keystore*

JRE

Java Runtime Enviroment · 7

K

keystores

is a repository of security certificates and private keys · 7

O

Open Virtualization Forma

Open format for exchanging virtual machines · 4

P

PKCS#12 · *see https://en.wikipedia.org/wiki/PKCS\_12*

R

RAM

Random-Access Memory · 4

W

Wildfly · *See http://www.wildfly.org/*