COMPETITIVENESS AND INNOVATION FRAMEWORK PROGRAMME
ICT PSP Fifth Call for proposals 2011 - Pilot Type A
Towards a single European electronic identification and authentication area

ICT PSP Theme/objective identifier: 4.2
Project acronym: STORK 2.0
Project full title: Secure idenTity acrOss boRders linKed 2.0
Grant agreement no.: 297263

VIDP 2.0 Installation Manual

Abstract: This document describes the installation, configuration and administration of a VIDP.
# History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modification reason</th>
<th>Modified by</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>30.3.2014</td>
<td>Initial documentation</td>
<td>Bojan Suzic</td>
</tr>
<tr>
<td>0.2</td>
<td>10.6.2014</td>
<td>Extending sections</td>
<td>Bojan Suzic</td>
</tr>
<tr>
<td>0.3</td>
<td>07.07.2014</td>
<td>Added setup for signature module</td>
<td>Sandra Kreuzhuber</td>
</tr>
<tr>
<td>0.4</td>
<td>24.02.2015</td>
<td>Updated signature deployment documentation</td>
<td>Sandra Kreuzhuber</td>
</tr>
<tr>
<td>0.5</td>
<td>05.05.2015</td>
<td>General refactoring</td>
<td>Johannes Feichtner</td>
</tr>
</tbody>
</table>
# Table of contents

History ......................................................................................................................... 2
Table of contents ........................................................................................................ 3
List of Figures .............................................................................................................. 5
List of Tables ............................................................................................................... 6
List of abbreviations ................................................................................................. 7
Executive summary ..................................................................................................... 8

1 Outline ................................................................................................................... 9

2 V-IDP Installation using the Virtual Appliance ......................................................... 10
   2.1 Apache httpd instance ....................................................................................... 11
   2.2 Apache Tomcat instance .................................................................................... 12
   2.3 Setting up and running Apache Tomcat instance ........................................... 13
   2.4 MOA-ID configuration files .............................................................................. 14
   2.5 MySQL Configuration files .............................................................................. 14

3 Configuration of V-IDP and its components ............................................................... 16
   3.1 MySQL configuration ......................................................................................... 16
   3.2 Preparing basic MOA configuration .................................................................. 16
   3.3 Preparing MOA configuration interface .......................................................... 19
   3.4 Configuring specific Stork support for MOA .................................................. 21
   3.5 General MOA configuration .............................................................................. 22
      3.5.1 Public URL Prefix ....................................................................................... 23
      3.5.2 SecurityLayer Request Templates ........................................................... 23
      3.5.3 Certificate check ......................................................................................... 24
      3.5.4 Session timeouts ......................................................................................... 25
      3.5.5 MOA-SPSS Configuration .......................................................................... 25
      3.5.6 External Services ......................................................................................... 26
      3.5.7 Single Sign-On ............................................................................................ 26
      3.5.8 Secure Identity Across Borders Linked ..................................................... 27
      3.5.9 Logging ....................................................................................................... 28
      3.5.10 Security Layer Transformations ............................................................... 28
   3.6 Setting up particular service provider .............................................................. 28
      3.6.1 Basic Settings ............................................................................................. 28
      3.6.2 CCE Configuration ...................................................................................... 29
      3.6.3 Mandates .................................................................................................... 30
      3.6.4 Secure Identity Across Borders Linked ..................................................... 30
      3.6.5 Additional general settings ........................................................................ 31
      3.6.6 CCE-Selection and Send-Assertion Templates ........................................ 32
   3.7 Other functionality of MOA-ID web configuration interface .............................. 34
   3.8 Configuring BKUOnline .................................................................................... 34
      3.8.1 Adjusting configuration file ....................................................................... 34
3.8.2 Configuring SSL certificates .............................................................. 35
3.8.3 Logging configuration ...................................................................... 35
3.8.4 DocumentService configuration ....................................................... 35

4 Manual V-1DP Installation ..................................................................... 36
4.1 System requirements .......................................................................... 36
4.2 Package contents ................................................................................ 36
4.3 Create MySQL databases ..................................................................... 37
4.4 Copy and set up MOA-ID configuration files ......................................... 37
4.5 Create and populate MOA-ID Keystore ................................................ 38
4.6 Configure environment variables ........................................................ 38
4.7 Copy endorsed files to Tomcat’s endorsed directory ................................ 38
4.8 Install JCE Policy files ........................................................................ 38
4.9 Copy ext files to ext directory of Java distribution ................................. 38
4.10 Deploy included WAR files .............................................................. 39
4.11 Configure DocumentService ............................................................ 39
4.12 Configure BKUOnline and disable support for Austrian Test Cards ........ 39
4.13 Install a production SSL certificate .................................................... 39
4.14 Setup general settings of MOA-ID ...................................................... 39
4.15 Initialize and setup MOA-ID web configuration interface ....................... 39
4.16 Setup MOA-ID and service provider application via web interface ........ 39
4.17 Security ............................................................................................ 40
4.18 Testing credentials for mobile citizen card environment ....................... 40

5 Setting up the signature module ............................................................. 41
5.1 Prerequisites ...................................................................................... 41
5.2 OASIS DSS ....................................................................................... 41
  5.2.1 Building the OASIS DSS module ................................................... 41
  5.2.2 Configuring the database ............................................................. 42
  5.2.3 Configuring the Oasis module ....................................................... 42
  5.2.4 Deployment in Tomcat .................................................................. 43
5.3 Austrian SPI Implementation .............................................................. 44
  5.3.1 Building the Austrian signature module ........................................ 45
  5.3.2 Configuring the SPI implementation ............................................ 45
  5.3.3 Configuring the web service ........................................................ 46
  5.3.4 Deployment in Tomcat ............................................................... 47

6 References ............................................................................................ 49
List of Figures

Figure 1: Mod_jk configuration (excerpt) ................................................................. 11
Figure 2: Definition of variables (excerpt) ............................................................... 14
Figure 3: Passing variables to Tomcat (excerpt) ....................................................... 14
Figure 4: Resetting MySQL password ..................................................................... 15
Figure 5: SamlEngine general configuration ............................................................ 21
Figure 6: Saml Signature Engine configuration ....................................................... 22
Figure 7: Example general configuration interface screen .................................... 22
Figure 8: Example configuration of C-PEPS URLs and attributes ....................... 27
Figure 9: Example AuthBlockText configuration .................................................... 31
Figure 10: Example transformation result of customized AuthTextBlock .......... 32
Figure 11: Example configuration of service provider application ..................... 33
Figure 12: Creating new users ............................................................................... 34
Figure 13: BkUOnline whitelist hosts ................................................................. 34
Figure 14: Setting up MySQL.............................................................................. 37
Figure 15: Application settings ........................................................................... 40
Figure 16: Modules required for Austrian SPI implementation ......................... 41
Figure 17: Modules required for reference SPI implementation ......................... 41
Figure 18: Creating the database ....................................................................... 42
Figure 19: Configuring oasis-dss ...................................................................... 43
Figure 20: Extending Tomcat context.xml ......................................................... 43
Figure 21: Extending Tomcat server.xml ............................................................ 44
Figure 22: Extending Tomcat variable definition ................................................. 44
Figure 23: Configuring signature-app-at ............................................................ 45
Figure 24: Configuring signature-web-at ........................................................... 46
Figure 25: Extending Tomcat variable definition ................................................. 47
List of Tables

Table 1: Relevant Apache httpd server and mod_jk paths ........................................ 11
Table 2: Relevant Apache Tomcat paths ................................................................. 12
Table 3: Paths and URLs of deployed applications ................................................ 12
Table 4: Relevant variables ...................................................................................... 13
Table 5: Configuration of MOA components ............................................................ 14
Table 6: MySQL relevant paths ............................................................................... 15
Table 7: Preconfigured MySQL databases ............................................................... 16
Table 8: MOA-ID basic configuration ..................................................................... 17
Table 9: MOA-ID database configuration ................................................................. 18
Table 10: Web interface configuration for MOA-ID .................................................. 20
Table 11: MOA-ID web configuration interface database configuration ................. 21
Table 12: MOA-ID general configuration .................................................................. 23
Table 13: Citizen Card Environments used by MOA-ID globally ......................... 23
Table 14: Security Layer templates used by Citizen Card Environment ................. 24
Table 15: Configuration of trusted certificates ....................................................... 24
Table 16: Session timeouts configuration ................................................................. 25
Table 17: MOA-SPSS configuration parameters ....................................................... 26
Table 18: Configuration of external services ............................................................ 26
Table 19: Configuration of Single Sign-On parameters ............................................ 27
Table 20: Basic settings for service provider application ......................................... 29
Table 21: Citizen Card Environment settings for service provider application ........ 29
Table 22: Mandates supported by service provider application ............................. 30
Table 23: Attribute providers settings ..................................................................... 31
Table 24: CCE-Selection and Send-Assertion Templates .......................................... 32
Table 25: Contents of deployment archive ............................................................... 36
Table 26: Contents of configuration directory in deployment archive .................... 36
## List of abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AQAA</td>
<td>Attribute Quality Authentication Assurance</td>
</tr>
<tr>
<td>AT</td>
<td>Austria</td>
</tr>
<tr>
<td>AP</td>
<td>Attribute Provider</td>
</tr>
<tr>
<td>BE</td>
<td>Belgium</td>
</tr>
<tr>
<td>CH</td>
<td>Switzerland</td>
</tr>
<tr>
<td>CZ</td>
<td>Czech Republic</td>
</tr>
<tr>
<td>EE</td>
<td>Estonia</td>
</tr>
<tr>
<td>eID</td>
<td>Electronic Identity</td>
</tr>
<tr>
<td>ES</td>
<td>Spain</td>
</tr>
<tr>
<td>EU</td>
<td>European Union</td>
</tr>
<tr>
<td>FR</td>
<td>France</td>
</tr>
<tr>
<td>GR</td>
<td>Greece</td>
</tr>
<tr>
<td>IS</td>
<td>Iceland</td>
</tr>
<tr>
<td>IT</td>
<td>Italy</td>
</tr>
<tr>
<td>LT</td>
<td>Lithuania</td>
</tr>
<tr>
<td>LU</td>
<td>Luxembourg</td>
</tr>
<tr>
<td>MS</td>
<td>STORK2.0 Member State</td>
</tr>
<tr>
<td>MW</td>
<td>MiddleWare</td>
</tr>
<tr>
<td>NL</td>
<td>The Netherlands</td>
</tr>
<tr>
<td>PEPS</td>
<td>Pan European Proxy Server</td>
</tr>
<tr>
<td>PT</td>
<td>Portugal</td>
</tr>
<tr>
<td>QAA</td>
<td>Quality Authentication Assurance</td>
</tr>
<tr>
<td>SAML</td>
<td>Security Assertion Markup Language</td>
</tr>
<tr>
<td>SE</td>
<td>Sweden</td>
</tr>
<tr>
<td>SI</td>
<td>Slovenia</td>
</tr>
<tr>
<td>SK</td>
<td>Slovakia</td>
</tr>
<tr>
<td>SP</td>
<td>Service Provider</td>
</tr>
<tr>
<td>STORK 2.0</td>
<td>Secure idenTity acrOss boRders linKed 2.0</td>
</tr>
<tr>
<td>TR</td>
<td>Turkey</td>
</tr>
<tr>
<td>UK</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>V-IDP</td>
<td>Virtual Identity Provider</td>
</tr>
<tr>
<td>WP</td>
<td>Work Package</td>
</tr>
</tbody>
</table>
Executive summary

This document is intended for system administrators, operators, and users of V-IDP system who want to deploy or harden a V-IDP system as a local installation.

It provides a quick reference on the installation and configuration of a V-IDP system. The document covers the reference installation of a preconfigured V-IDP system provided by Graz University of Technology and A-SIT Austria. Aside from explaining the structure of the V-IDP deployment archive, the steps necessary to perform a manual installation are listed.

By the last page of this tutorial, the reader should be able to install and configure a V-IDP system, should have acquired a better understanding of its mode of operation and be able to give primary support in the case of troubleshooting.
1 Outline

This document consists of multiple parts and is structured as follows:

- **Installation:**
  - Chapter 2 explains the deployment using the provided Virtual Alliance
  - Alternatively, Chapter 4 describes the manual installation process

- **Configuration:**
  - Chapter 3 highlights the most relevant configuration parameters and their purpose. These options have to be adjusted regardless of whether the V-IDP is deployed via the Virtual Machine or manually.
  - Chapter 5 describes how to install the OASIS signature module.

The Virtual Appliance, as presented in Chapter 2, includes all components which are necessary to run the V-IDP system. Although the system comes preconfigured, it is still necessary to adjust settings like passwords or URLs to the individual requirements. The relevant options are presented in Chapter 3.

Alternatively, if the preconfigured system is not employed, Chapter 4 lists all components and settings needed to run the V-IDP on a custom system. Basically, this chapter explains the steps that are already done when using the provided Virtual Appliance. Having installed the needed modules, the final configuration refers to the description provided in Chapter 3.

The last part covers the installation of the OASIS DSS module and the Austrian signature modules in order to support signature creation on the V-IDP system.

For further questions, suggestions and comments please contact the Austrian V-IDP team. Detailed contact information is available at the STORK 2 Wiki pages.
2 V-IDP Installation using the Virtual Appliance

Austria provides all PEPS countries with a reference virtual machine that can be used to easily deploy the V-IDP system.

The virtual machine is based on Debian and contains an optimal set of packages needed to run the V-IDP services, including the following set of components:

- Linux kernel 3.2.0-4- amd64
- Sun Java SDK 1.7.0.51-b13
- Apache httpd 2.2.22-13
- Apache Tomcat 7.0.28-4
- Apache 2 mod_jk 1.2.37
- MySQL Server 5.5.35

Minimal hardware requirements:
- 1 vCPU
- 2 GB RAM
- 5 GB HDD

Networking requirements:
- public IP or NAT
- ports-incoming: 80, 443
- ports-outgoing: 80, 443, 389, 636

Incoming HTTP(S) requests (from a client) are first handled by the Apache httpd front-end webserver, which comes installed and configured in its standard Debian flavor. The installation package also includes the mod_jk module, which acts as a bridge between the backend Apache Tomcat server and the Apache webserver. The Apache Tomcat server instance provided by this reference system contains all relevant modules belonging to the V-IDP. Furthermore, a MySQL server used to store the configuration, session and statistics.

Table 2 provides the list of configuration locations relevant for configured Apache Tomcat instance.

The credentials used to login to the virtual machine are:

**Superuser:** root  
**Password:** st12knn

**User:** stork  
**Password:** st12knn

**Note:** For security reasons, users are urged to change the passwords of both accounts.
2.1 Apache httpd instance

The reference system runs an Apache 2 instance including the mod_jk module. It acts as a frontend on standard HTTP(s) ports and using mod_jk connects to the local Tomcat service running a MOA-ID application.

<table>
<thead>
<tr>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/etc/apache2/</td>
<td>Base configuration directory</td>
</tr>
<tr>
<td>/etc/apache2/mods-enabled</td>
<td>Enabled modules, links to configuration of modules</td>
</tr>
<tr>
<td>/etc/apache2/ssl</td>
<td>Location for SSL certificates</td>
</tr>
<tr>
<td>/var/log/apache2</td>
<td>Apache log files</td>
</tr>
</tbody>
</table>

*Table 1: Relevant Apache httpd server and mod_jk paths*

The Apache2 instance can be started and stopped by issuing the following commands:

```
service apache2 stop
service apache2 start
```

For information purposes, the following figure depicts the default configuration of the mod_jk module. Basically, it is not necessary to personalize this configuration.

```
JkWorkersFile /etc/libapache2-mod-jk/workers.properties
JkLogFile /var/log/apache2/mod_jk.log
JkLogLevel info
JkShmFile /var/log/apache2/jk-runtime-status
JkWatchdogInterval 60
<Location /jk-status>
    JkMount jk-status
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
</Location>
<Location /jk-manager>
    JkMount jk-manager
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
</Location>
```

*Figure 1: Mod_jk configuration (excerpt)*
2.2 Apache Tomcat instance

The reference system runs an **Apache Tomcat 7** instance containing the MOA applications. Table 2 shows the most relevant paths and a descriptions of their purpose.

<table>
<thead>
<tr>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/share/tomcat7</td>
<td>Contains Tomcat binaries and libraries</td>
</tr>
<tr>
<td>/var/lib/tomcat7</td>
<td>Contains shared and common files, deployment and directory for endorsed files</td>
</tr>
<tr>
<td>/var/lib/tomcat7/webapps</td>
<td>Contains Austrian authentication middleware</td>
</tr>
<tr>
<td>/etc/tomat7</td>
<td>Configuration directory</td>
</tr>
<tr>
<td>/var/log/tomcat7</td>
<td>Log files</td>
</tr>
</tbody>
</table>

*Table 2: Relevant Apache Tomcat paths*

The Tomcat server contains the web apps **ConfigurationInterface**, **moa-id-auth** and **bkuonline**, which basically build up the V-IDP system. By default, they are all found in the **webapps** directory.

The following list briefly outlines their role:

- **ConfigurationInterface**
  
  Provides the web-based configuration interface for MOA. This interface is used to configure general MOA options, as well as to add and configure individual online applications (service providers).

- **moa-id-auth**
  
  Set of software tools which provide implementation and integration support for the functions and procedures mandate by the Austrian eGov strategy. Contains integrated support for Stork 2 functionalities.

- **bkuonline**
  
  Modular Austrian citizen-card environment. Provides online module and interface interacting the user’s citizen card.

The following table exemplifies how these webapps map to callable URLs:

<table>
<thead>
<tr>
<th>Path</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>bkuonline</td>
<td><a href="https://testvidp.buergerkarte.at/bkuonline">https://testvidp.buergerkarte.at/bkuonline</a></td>
</tr>
<tr>
<td>moa-id-auth</td>
<td><a href="https://testvidp.buergerkarte.at/moa-id-auth">https://testvidp.buergerkarte.at/moa-id-auth</a></td>
</tr>
<tr>
<td>moa-id-configuration</td>
<td><a href="https://testvidp.buergerkarte.at/moa-id-configuration">https://testvidp.buergerkarte.at/moa-id-configuration</a></td>
</tr>
</tbody>
</table>

*Table 3: Paths and URLs of deployed applications*
2.3 Setting up and running Apache Tomcat instance

Before the Tomcat instance is initialized, the necessary environment variables are passed to the server and applications.

The following table contains the list of included variables. In reference installation they are found in the file `/etc/tomcat7/set-variables.sh`.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGS_DIR</td>
<td>Destination of log data</td>
</tr>
<tr>
<td>CATALINA_OUT</td>
<td>Log file containing Tomcat's standard output</td>
</tr>
<tr>
<td>LOGS_BASE</td>
<td>Base directory for other logs</td>
</tr>
<tr>
<td>Logging_Opt</td>
<td>Points to file containing logging configuration</td>
</tr>
<tr>
<td>FILE_ENCODING</td>
<td>Sets file encoding</td>
</tr>
<tr>
<td>MOA_ID_CONFIG</td>
<td>Location of base MOA-ID configuration file</td>
</tr>
<tr>
<td>MOA_ID_CONFIG_TOOL</td>
<td>Location of base MOA-ID web interface configuration file</td>
</tr>
<tr>
<td>MOA_ID_STORK</td>
<td>Location of STORK related files such as SAMLEngine configurations</td>
</tr>
<tr>
<td>MOA_SPSS_CONFIG</td>
<td>Points to MOA-SPSS configuration file</td>
</tr>
<tr>
<td>MOA_SPSS_TSL_HASHCACHE</td>
<td>Caching directory used by MOA-SPSS</td>
</tr>
<tr>
<td>JAVA_ENDORSED_DIRS</td>
<td>Path to endorsed directory</td>
</tr>
</tbody>
</table>

*Table 4: Relevant variables*

The following two figures contain the excerpts from `/etc/tomcat7/set-variables.sh`. The parameters in this file are used to set and pass MOA-ID and other configuration options to MOA-ID running under Tomcat.

```
JAVA_HOME=/usr/lib/jvm/java
LOGS_DIR=/var/log/tomcat7/moaid-2.0
CATALINA_OUT=$LOGS_DIR/catalina.out
LOGS_BASE=-Dlogs.base=$LOGS_DIR
LOGGING_Opt=-Dlog4j.configuration=file:$CATALINA_BASE/conf/log4j.properties
FILE_ENCODING=-Dfile.encoding=UTF-8
RAND_FILE=-Djava.security.egd=file:/dev/urandom
MOA_ID_CONFIG=-Dmoa.id.configuration=$CATALINA_BASE/conf/moa-id/moa-id.properties
MOA_ID_CONFIG_TOOL=-Dmoa.id.webconfig=$CATALINA_BASE/conf/moa-id-configuration/moa-id-configtool.properties
MOA_ID_STORK=-Deu.stork.samlengine.config.location=$CATALINA_BASE/conf/moa-id/stork/
MOA_SPSS_CONFIG=-Dmoa.spss.server.configuration=$CATALINA_BASE/conf/moa-spss/MOASPSSConfiguration.xml
MOA_SPSS_TSL_HASHCACHE=-Diaik.xml.crypto.tsl.BinaryHashCache.DIR=$CATALINA_BASE/conf/moa-spss/tslworking/hashcache/
```
JAVA_ENDORSED_DIRS=${CATALINA_BASE/endorsed}

Figure 2: Definition of variables (excerpt)

JAVA_OPTS="$LOGS_BASE $FILE_ENCODING $LOGGING_OPT $MOA_ID_CONFIG $MOA_SPSS_CONFIG $MOA_ID_CONFIG_TOOL -Dsun.security.ssl allowableRenegotiation=true $RAND_FILE -XX:PermSize=64m -XX:MaxPermSize=786m $MOA_ID_STORK"

JVM_OPTS="$FILE_ENCODING;$LOGGING_OPT;$MOA_ID_CONFIG;$MOA_SPSS_CONFIG;$MOA_ID_CONFIG_TOOL; -Dsun.security.ssl allowableRenegotiation=true;$RAND_FILE; -XX:PermSize=64m; -XX:MaxPermSize=786m;$MOA_ID_STORK"

CATALINA_OPTS="$LOGGING_OPT $FILE_ENCODING $MOA_ID_CONFIG $MOA_SPSS_CONFIG $MOA_SPSS_TSL_HASHCACHE $PARAM_TRUST_STORE $PARAM_TRUST_STORE_PASS $PARAM_SSL_DEBUG $MOA_ID_CONFIG_TOOL -Dsun.security.ssl allowableRenegotiation=true $RAND_FILE -XX:MaxPermSize=786m $MOA_ID_STORK"

Figure 3: Passing variables to Tomcat (excerpt)

The Tomcat 7 instance can be started and stopped by issuing the following commands:

service tomcat7 stop

service tomcat7 start

2.4 MOA-ID configuration files

Table 5 presents the list of MOA-ID and related paths, as well as brief explanations on their contents. The referenced directories are available under Tomcat configuration directory – in the case of reference system it is /etc/tomcat7.

<table>
<thead>
<tr>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>moa-id</td>
<td>The base directory for MOA-ID configuration</td>
</tr>
<tr>
<td>moa-id/keys</td>
<td>Contains the keystore used by SAMLEngine</td>
</tr>
<tr>
<td>moa-id/stork</td>
<td>Contains STORK related SAMLEngine configuration files</td>
</tr>
<tr>
<td>moa-id-configuration</td>
<td>Configuration of MOA-ID web configuration interface, as well as templates and transformations</td>
</tr>
<tr>
<td>moa-spss</td>
<td>MOA-SPSS directory containing MOA-SPSS configuration, trust and cert store as well as transformation profiles</td>
</tr>
</tbody>
</table>

Table 5: Configuration of MOA components

2.5 MySQL Configuration files

Table 6 contains the list of the relevant MySQL server paths. The files contained there are already preconfigured and normally do not have to be changed.
Both the username and password used to connect to MySQL server are vidp. Users are advised to change the password, as well to reset the root MySQL password. The following snippet highlights the commands required to achieve this:

```bash
sudo /etc/init.d/mysql stop
sudo mysqld_safe --skip-grant-tables &
mysql -u root
use mysql;
update user set password=PASSWORD("mynewrootpassword") where User='root';
update user set password=PASSWORD("mynewvidppassword") where User='vidp';
flush privileges;
quit

sudo /etc/init.d/mysql stop
sudo /etc/init.d/mysql start
```

Figure 4: Resetting MySQL password

The code shown in Figure 4 should be entered at the Debian command line. The first part of the code is used to start the MySQL server in safe mode. The middle part is then used to log in to MySQL instance and set the passwords for root and vidp users anew. At the end, the instance is restarted in normal mode.

Please note that changing the MySQL password of the vidp User also requires you to adapt all references where it is used. Practically, this affects the file `/etc/tomcat7/moa-id/moa-id.properties` in which the parameters moasession.hibernate.connection.password and advancedlogging.hibernate.connection.password have to be reset. Likewise, the parameter hibernate.connection.password has to be adjusted in the file `/etc/tomcat7/moa-id-configuration/moa-id-configtool.properties`.

<table>
<thead>
<tr>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/etc/mysql</code></td>
<td>The base directory for configuration of MySQL server</td>
</tr>
<tr>
<td><code>/etc/mysql/debian.cnf</code></td>
<td>Automatically generated configuration by Debian – should not be changed directly</td>
</tr>
<tr>
<td><code>/etc/mysql/my.cnf</code></td>
<td>MySQL server main configuration file</td>
</tr>
<tr>
<td><code>/var/log/mysql</code></td>
<td>MySQL server log location</td>
</tr>
</tbody>
</table>

Table 6: MySQL relevant paths
3 Configuration of V-IDP and its components

This section explains all available configuration options for the V-IDP system. For deployment, it is required to personalize option values, such as the MySQL password or constant URLs.

All references to specific files or directories in this section are based on the reference installation provided in the form of V-IDP virtual machine, as introduced in Section 2: V-IDP Installation using the Virtual . Hence, if deployed individually without a preconfigured appliance, the paths of the configuration files and modules might vary.

3.1 MySQL configuration

The V-IDP requires a storage backend in order to store the necessary configuration, session and statistics data. For this purpose, it comes with a preinstalled MySQL server and three preconfigured databases, as shown in Table 7: Preconfigured MySQL databases.

<table>
<thead>
<tr>
<th>Database</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>moa-id-config</td>
<td>Various configuration settings</td>
</tr>
<tr>
<td>moa-id-session</td>
<td>Temporary session data</td>
</tr>
<tr>
<td>moa-id-statistics</td>
<td>Statistics on MOA-ID usage</td>
</tr>
</tbody>
</table>

Table 7: Preconfigured MySQL databases

Unless changed, both the username and password used to connect to MySQL server are vidp. Note: For security reasons, users are strongly encouraged to change the MySQL credentials and access settings to conform to local deployment environment and security requirements.

3.2 Preparing basic MOA configuration

Before a MOA instance should be started, it is necessary to review and configure its configuration parameters available under /etc/tomcat7/moa-id/moa-id.properties. This file provides the basic configuration which is necessary to start MOA service.

This file contains some parameters which are used in specific scenarios relevant for providers using other types of interfaces or authentication, internally in Austria. In the context of the Stork 2 deployment, most of these parameters are not used and there for not relevant. The following table lists the most relevant parameters for deployment case in the terms of Stork 2 scenario:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>configuration.moasession.key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>MyKey123</td>
</tr>
<tr>
<td>Description</td>
<td>Passphrase used to encrypt the session data in database. Optional.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Configuration.monitoring.active</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Example</td>
<td>true</td>
</tr>
<tr>
<td>Description</td>
<td>Defines the status of the monitoring servlet</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Configuration.monitoring.message.success</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>All Tests passed!</td>
</tr>
<tr>
<td>Description</td>
<td>Success message sent when the tests ended successfully</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Configuration.monitoring.test.identitylink.url</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>/test/idl/test_identitylink.xml</td>
</tr>
<tr>
<td>Description</td>
<td>URL of IdentityLink used for testing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>configuration.advancedlogging.active</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>true</td>
</tr>
<tr>
<td>Description</td>
<td>Defines the status of additional logging facilities (in database)</td>
</tr>
</tbody>
</table>

**Table 8: MOA-ID basic configuration**

The next table contains description of database configuration parameters on MOA-ID related to session data store, configuration and statistics data store.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>mosa.session.hibernate.dialect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>org.hibernate.dialect.MySQLDialect</td>
</tr>
<tr>
<td>Description</td>
<td>Dialect used to communicate with MOA-session data store</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>mosa.session.hibernate.connection.url</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>jdbc:mysql://localhost/moa-id-session?charSet=utf-8&amp;autoReconnect=true</td>
</tr>
<tr>
<td>Description</td>
<td>URL to the MOA-session database schema</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>mosa.session.hibernate.connection.driver_class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>com.mysql.jdbc.Driver</td>
</tr>
<tr>
<td>Description</td>
<td>Driver used to access session database</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>mosa.session.hibernate.connection.username</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>vidp</td>
</tr>
<tr>
<td>Description</td>
<td>The user name used to access the session database</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>mosa.session.hibernate.connection.password</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>vidp</td>
</tr>
<tr>
<td>Description</td>
<td>The password used to access the session database</td>
</tr>
<tr>
<td>Parameter</td>
<td>Example</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>configuration.hibernate.dialect</td>
<td>org.hibernate.dialect.MySQLDialect</td>
</tr>
<tr>
<td>configuration.hibernate.connection.url</td>
<td>jdbc:mysql://localhost/moa-id-config?charset=utf-8&amp;autoReconnect=true</td>
</tr>
<tr>
<td>configuration.hibernate.connection.driver_class</td>
<td>com.mysql.jdbc.Driver</td>
</tr>
<tr>
<td>configuration.hibernate.connection.username</td>
<td>vidp</td>
</tr>
<tr>
<td>moasession.hibernate.connection.password</td>
<td>vidp</td>
</tr>
<tr>
<td>advancedlogging.hibernate.dialect</td>
<td>org.hibernate.dialect.MySQLDialect</td>
</tr>
<tr>
<td>advancedlogging.hibernate.connection.url</td>
<td>jdbc:mysql://localhost/moa-id-statistics?charset=utf-8&amp;autoReconnect=true</td>
</tr>
<tr>
<td>advancedlogging.hibernate.connection.driver_class</td>
<td>com.mysql.jdbc.Driver</td>
</tr>
<tr>
<td>advancedlogging.hibernate.connection.username</td>
<td>vidp</td>
</tr>
<tr>
<td>advancedlogging.hibernate.connection.password</td>
<td>vidp</td>
</tr>
</tbody>
</table>

*Table 9: MOA-ID database configuration*
3.3 Preparing MOA configuration interface

The module for MOA web configuration interface is started separately. Its basic configuration can be adjusted at the file `/etc/tomcat7/moa-id-configuration/moa-id-configtool.properties`.

All other application relevant parameters are further defined in this configuration interface. The following table provides description of most relevant parameters in this file.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>general.login.deaktivate</td>
<td>With this option login using web interface can be deactivated.</td>
</tr>
<tr>
<td>general.publicURLContext</td>
<td>This is the URL of web configuration interface</td>
</tr>
<tr>
<td>general.defaultlanguage</td>
<td>Defines default language for web configuration interface</td>
</tr>
<tr>
<td>general.mail.host</td>
<td>The mail server used to send the emails</td>
</tr>
<tr>
<td>general.mail.from.name</td>
<td>The name of the sender of emails</td>
</tr>
<tr>
<td>general.mail.from.address</td>
<td>The <code>from</code> address used when sending emails from the host</td>
</tr>
<tr>
<td>general.mail.useraccountrequest.verification.subject</td>
<td>Subject of verification email sent when opening account</td>
</tr>
<tr>
<td>general.mail.useraccountrequest.verification.template</td>
<td>Location of the verification’s email template</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>general.mail.useraccountrequest.isactive.subject</td>
<td>Subject of activation mail sent after the account is activated</td>
</tr>
<tr>
<td>mail/activation_template.html</td>
<td>Location of the activation’s email template</td>
</tr>
<tr>
<td>mail/rejected_template.html</td>
<td>Location of the template used to inform the user about rejection or removal of its requested account.</td>
</tr>
<tr>
<td>mail/oa_activation_template.html</td>
<td>Location to the template used for service provider activation email</td>
</tr>
<tr>
<td><a href="mailto:admin@testvidp.buergerkarte.at">admin@testvidp.buergerkarte.at</a></td>
<td>Email address of system admin</td>
</tr>
<tr>
<td>MOA-ID Status information</td>
<td>Subject of status information message</td>
</tr>
<tr>
<td>mail/admin_template.html</td>
<td>Location of status information/change template sent to the admin</td>
</tr>
</tbody>
</table>

**Table 10: Web interface configuration for MOA-ID**

In the following table described are database specific configuration parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>org.hibernate.dialect.MySQLDialect</td>
<td>The SQL dialect used to store configuration</td>
</tr>
</tbody>
</table>
### 3.4 Configuring specific Stork support for MOA

The SAMLEngine, developed in the course of the STORK project, is used for the signing and verification of STORK SAML messages. The specific configuration files can be found in the folder `/etc/tomcat7/moa-id/stork/`.

The main file for defining sub-configurations located in this directory is `/etc/tomcat7/moa-id/stork/SamlEngine.xml`. This file defines and references instance configurations for each profile used in the application.

The following is snippet excerpt from `SamlEngine.xml`:

```xml
<instance name="VIDP">
<!-- Configurations parameters StorkSamlEngine -->
<configuration name="SamlEngineConf">
<parameter name="fileConfiguration" value="StorkSamlEngine_VIDP.xml" />
</configuration>
<!-- Settings module signature -->
<configuration name="SignatureConf">
<!-- Specific signature module -->
<parameter name="class" value="eu.stork.peps.auth.engine.core.impl.SignSW" />
<!-- Settings specific module -->
<parameter name="fileConfiguration" value="SignModule_VIDP.xml" />
</configuration>
</instance>
```

*Figure 5: SamlEngine general configuration*
In this example, the instance references two additional files, `StorkSamlEngine_VIDP.xml` and `SignModule_VIDP.xml`. First, the Sam1Engine configuration contains attribute definitions and other settings. Generally, there is no need to adjust this file. The second configuration `SignatureEngine` contains settings necessary to access the keystore used to sign and verify SAML messages.

The following is a snippet excerpt from that file:

```xml
<properties>
  <comment>SwModule sign with JKS.</comment>
  <entry key="keystorePath">/etc/tomcat7/moa-id/keys/storkDemoKeys.jks</entry>
  <entry key="keyStorePassword">change-me</entry>
  <entry key="keyPassword">change-me</entry>
  <entry key="issuer">C=AT, L=Graz, OU=Test Institute, CN=EuroPKI CA</entry>
  <entry key="serialNumber">2FCAA2CDB1522</entry>
  <entry key="keystoreType">JKS</entry>
</properties>
```

**Figure 6: Saml Signature Engine configuration**

**Note:** Nearly all lines of this configuration snippet should be adjusted in order to reflect the individual, local configuration!

The referenced keystore should contain trusted certificates from other trusted parties, such as PEPses, and private keys used to sign message delivered to the peers.

### 3.5 General MOA configuration

The general configuration is accessed through the web configuration interface of MOA-ID. By default, this interface is located at `https://vidp-domain.tld/moa-id-configuration`.

**Figure 7: Example general configuration interface screen**
The default login data for the MOA-ID configuration interface are:

Username: stork
Password: st12knn

The following subsections contain most important options for the Stork-based system setup.

### 3.5.1 Public URL Prefix

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Public URL Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td><a href="https://testvidp.buergerkarte.at/moa-id-auth">https://testvidp.buergerkarte.at/moa-id-auth</a></td>
</tr>
<tr>
<td>Description</td>
<td>URL-Prefix of MOA-ID instance. This setting is used for automatic generation of forms and information and must be defined.</td>
</tr>
</tbody>
</table>

*Table 12: MOA-ID general configuration*

In this section defined are the parameters for standard Citizen Card Environment. They are used by MOA-ID in the authentication process in the case there is no specific CCE defined for particular service provider.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Online CCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td><a href="https://testvidp.buergerkarte.at/bkuonline/https-security-layer-request">https://testvidp.buergerkarte.at/bkuonline/https-security-layer-request</a></td>
</tr>
<tr>
<td>Description</td>
<td>URL to the Online-CCE instance</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mobile CCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>URL to the Mobile-CCE instance</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Local CCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td><a href="https://127.0.0.1:3496/https-security-layer-request">https://127.0.0.1:3496/https-security-layer-request</a></td>
</tr>
<tr>
<td>Description</td>
<td>URL pointing to the locally installed and executed CCE instance</td>
</tr>
</tbody>
</table>

*Table 13: Citizen Card Environments used by MOA-ID globally*

### 3.5.2 SecurityLayer Request Templates

Security Layer Templates are used in the communication with previously configured Citizen Card Environment. The communication is based on HTTP Form, which is sent to the Citizen Card Environment through HTTP POST Request.

Necessary templates are included as a part of V-IDP distribution. They can be further customized and configured, however that is beyond the scope of this setup.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online CCE</td>
<td>SL Template for communication with Online-CCE</td>
</tr>
<tr>
<td><a href="http://testvidp.buergerkarte.at/moa-id-auth/template_onlineBKU.html">http://testvidp.buergerkarte.at/moa-id-auth/template_onlineBKU.html</a></td>
<td></td>
</tr>
<tr>
<td>Mobile CCE</td>
<td>SL Template for communication with Mobile-CCE</td>
</tr>
<tr>
<td><a href="http://testvidp.buergerkarte.at/moa-id-auth/template_handyBKU.html">http://testvidp.buergerkarte.at/moa-id-auth/template_handyBKU.html</a></td>
<td></td>
</tr>
<tr>
<td>Local CCE</td>
<td>SL Template for communication with local CCE instance</td>
</tr>
<tr>
<td><a href="http://testvidp.buergerkarte.at/moa-id-auth/template_localBKU.html">http://testvidp.buergerkarte.at/moa-id-auth/template_localBKU.html</a></td>
<td></td>
</tr>
</tbody>
</table>

**Table 14: Security Layer templates used by Citizen Card Environment**

### 3.5.3 Certificate check

In this section considered are general settings for certificate check and the configuration of trusted certificates. Referenced directories are relative to MOA-ID configuration directory – in the case of this reference installation it is `/etc/tomcat7/moa-id`.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CertStoreDirectory</td>
<td>Path name to the directory containing trust store used in the course of TLS server based certificate check.</td>
</tr>
<tr>
<td>certs/certstore</td>
<td></td>
</tr>
<tr>
<td>TrustManagerRevocationChecking</td>
<td>For TLS server based authentication there can be used only server certificates containing CRLDP extension. If there is no CRLDP extension contained, MOA-ID would not be able to perform CRL based check. If RevocationChecking is disabled, this setting should be set to false.</td>
</tr>
<tr>
<td>TrustedCACertificates</td>
<td>Path to the directory containing trusted CA certificates. In the course of TLS server based certificate check these CA would be considered as trusted.</td>
</tr>
<tr>
<td>certs/ca-certs</td>
<td></td>
</tr>
<tr>
<td>Chaining Mode</td>
<td>This option defines if the certificate check method would be based on chaining or on RFC 3280 pkix.</td>
</tr>
<tr>
<td>pkix</td>
<td></td>
</tr>
</tbody>
</table>

**Table 15: Configuration of trusted certificates**
### 3.5.4 Session timeouts

This subsection defines general timeouts used in the system, expressed in seconds.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Assertion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>120</td>
</tr>
<tr>
<td>Description</td>
<td>Defines the time window for authentication data, session data or assertions contained in the system to be considered as valid. After this time is passed the data will be removed or the authentication process canceled. This setting influences the maximal time allowed to collect additional STORK attributes.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>SSO Session authenticated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>1200</td>
</tr>
<tr>
<td>Description</td>
<td>Defines time window for validity of Single Sign-On session, starting from the time point of authentication. After the expiration of this period, the user will have to authenticate again.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>SSO Session last access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>2700</td>
</tr>
<tr>
<td>Description</td>
<td>Defines time window for validity of Single Sign-On session, starting from the time point of the last access. After the expiration of this period, the user will have to authenticate again.</td>
</tr>
</tbody>
</table>

Table 16: Session timeouts configuration

---

### 3.5.5 MOA-SPSS Configuration

This sections defines relevant parameters for configuring the options of MOA-SP module, which is used for checking of signature of identity link and authentication block. This module is shipped a part of MOA-ID and in Stork based scenario should generally not be adjusted.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Trustprofile for Identity Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>MOAIDBuergerkartePersonenbindungMitTestkarten</td>
</tr>
<tr>
<td>Description</td>
<td>This element specifies TrustProfileID used during verification of signature of the identity link, a part of signature request. The same TrustProfileID must be configured in the corresponding MOA-SP module.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Trustprofile for authentication block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>MOAIDBuergerkarteAuthentisierungsDatenMitTestkarten</td>
</tr>
<tr>
<td>Description</td>
<td>This element specifies TrustProfileID used during verification of signature of the authentication block, a part of signature request. The same TrustProfileID must be configured in the corresponding MOA-SP module.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Transformations for authentication block</th>
</tr>
</thead>
</table>
Example | MOAIDTransformAuthBlockTable_DE_new  
---|---
Description | This option defines ID of transformations profile, used during the verification of the signature of authentication block. The same transformation profile must be configured in the corresponding MOA-SP module.

| Parameter | URL for MOA-SP service  
---|---
| Example | http://testinstance.tld/moaspss  
| Description | URL of the used MOA-SP service. Optional. If this option is not configured, the MOA-ID will use the integrated version of MOA-SP. The MOA-SP configuration can be found under /etc/tomcat7/moa-spss/MOASPSSConfiguration.xml.

**Table 17: MOA-SPSS configuration parameters**

### 3.5.6 External Services

The following parameters are used for communication with external services such as online mandating service and source-PIN registry. These parameters should be configured only in special cases.

| Parameter | Online-Mandate Service URL  
---|---
| Example | https://vollmachten.egiz.gv.at/mis-test/MandateIssueRequest  
| Description | URL of online mandating service  

| Parameter | SZR Gateway Service URL  
---|---
| Example | https://szrgw.egiz.gv.at/services_2.0/IdentityLinkCreation  
| Description | URL of source-PIN registry  

**Table 18: Configuration of external services**

### 3.5.7 Single Sign-On

The options in this section define the parameters necessary for implementation of single sign-on functionality. This functionality is currently not relevant for STORK 2 application.
Parameter | SSO Auth Block Text
---|---
Example | 
Description | 

| Table 19: Configuration of Single Sign-On parameters |

### 3.5.8 Secure Identity Across Borders Linked

The parameters in this section specify the settings directly relevant for the Stork deployment. Hereby, it is possible to define the **standard QAA-Level** of requests, which sets minimal requirements for QAA. Moreover, for each relevant country the appropriate **C-PEPS URL** should be entered.

The next available setting is **Configuration of attributes**. This setting enables the definition of generally supported STORK attributes, such as eIdentifier, givenName and others. The field **mandatory** is used to mark attributes which must be included in the response from PEPS. Figure 8 shows the example configuration of attributes.

![Figure 8: Example configuration of C-PEPS URLs and attributes](image-url)
3.5.9 Logging

This subsection allows specific settings for logging configuration. These are not relevant for STORK based deployments.

3.5.10 Security Layer Transformations

The SL-Transformations are used by MOA-ID to create the signature of authentication block. The transformation file should be uploaded separately through this interface. The standard configuration is available at:

/etc/tomcat7/moa-id/transforms/TransformsInfoAuthBlockTable_DE_new.xml

3.6 Setting up particular service provider

The configuration of a service provider is initiated from the main web-interface menu, found under the option Interfederation > STORK VIDP. The list of available applications can be invoked through the option Interfederation. Furthermore, the service provider configuration is generally done by the users with administrator rights, but the basic settings for particular applications can be also done by other users registered in the system.

3.6.1 Basic Settings

This section defines the basic and parameters of service provider, such as its name and URL.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online-Application is activated</td>
<td>Allows to activate and deactivate service provider application.</td>
</tr>
<tr>
<td>Example: true</td>
<td></td>
</tr>
<tr>
<td>Unique identifier (PublicURLPrefix):</td>
<td>This parameter is used to distinguish service provider applications and find/retrieve the specific one. The identifier should contain at least the URL prefix of externally visible and reachable service provider application</td>
</tr>
<tr>
<td>Example: <a href="https://peps-test.gov.tld/SP/ReturnPage">https://peps-test.gov.tld/SP/ReturnPage</a></td>
<td></td>
</tr>
<tr>
<td>Name of the Online-Application:</td>
<td>User defined name of service provider. It is shown during user authentication process.</td>
</tr>
<tr>
<td>Example: Demo SP</td>
<td></td>
</tr>
<tr>
<td>Private sector application</td>
<td>This parameter should be set to true for STORK service providers</td>
</tr>
<tr>
<td>Example: true</td>
<td></td>
</tr>
</tbody>
</table>

1 In this document referred also as an online application
### Table 20: Basic settings for service provider application

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Identification number:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>STORK IT</td>
</tr>
<tr>
<td>Description</td>
<td>This parameter is mandatory and should be used to set the SP country.</td>
</tr>
</tbody>
</table>

#### 3.6.2 CCE Configuration

This section defines the citizen card environment which can be used to authenticate users. Please note that **Online CCE** should point to **BKUOnline** instance deployed at local premises. Details about that are provided in Section 3.8. Additionally, if you want to use test credentials for **Mobile CCE**, further details are provided in Section 4.18.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Online CCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td><a href="https://testvidp.buergerkarte.at/bkuonline/https-security-layer-request">https://testvidp.buergerkarte.at/bkuonline/https-security-layer-request</a></td>
</tr>
<tr>
<td>Description</td>
<td>URL to the application specific Online-CCE instance. If no instance is configured, the default one from the MOA-ID general configuration is applied.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mobile CCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>URL to the application specific Mobile-CCE instance. If no instance is configured, the default one from the MOA-ID general configuration is applied.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Local CCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td><a href="https://127.0.0.1:3496/https-security-layer-request">https://127.0.0.1:3496/https-security-layer-request</a></td>
</tr>
<tr>
<td>Description</td>
<td>URL pointing to the locally installed and executed CCE instance. If no parameter is set, the default one from the MOA-ID general configuration is used.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>KeyBoxIdentifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>SecureSignatureKeypair</td>
</tr>
<tr>
<td>Description</td>
<td>Configures keypair used to sign the authentication block. Default value is SecureSignatureKeypair.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>SecurityLayerTemplates (Legacy Request)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>false</td>
</tr>
<tr>
<td>Description</td>
<td>Using this option it is possible to define additional three SecurityLayer templates. In STORK related deployments this selector should be left unchecked.</td>
</tr>
</tbody>
</table>

*Table 21: Citizen Card Environment settings for service provider application*
3.6.3 Mandates

This subsection allows definition of mandating schemas used in the authentication process.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mandates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>true</td>
</tr>
<tr>
<td>Description</td>
<td>This setting defines if the service provider application supports online-mandates. If this box is checked, the option to log in as representative will be shown during authentication process at the CCE selection step.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>Gesundheit,Zustellung,ERsB,GenerallvollmachtBilateral</td>
</tr>
<tr>
<td>Description</td>
<td>This element contains the comma-separated list of mandating profiles. The full list is available under <a href="https://vollmachten.stammzahlenregister.gv.at/mis">https://vollmachten.stammzahlenregister.gv.at/mis</a>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Allow mandated login only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>false</td>
</tr>
<tr>
<td>Description</td>
<td>Allows setting the possibility to accept authentication based on powers only.</td>
</tr>
</tbody>
</table>

Table 22: Mandates supported by service provider application

3.6.4 Secure Identity Across Borders Linked

The parameters from this section specify STORK protocol relevant settings for the service provider application.

The VIDP interface is active parameter is used when the service provider application needs to support the login of the user from other countries. This way, on the login web page the option Home Country Selection will be enabled.

The option Ask the user for attributes transfer consent is applied to activate the user consent request before the user attributes are transferred to peer entity such as PEPS in cross-border scenario.

The next available setting is configuration of List of configured attribute providers. Each entry of the list should contain attribute provider plugin which should handle the request, the URL of attribute provider as well as attributes supported by that provider separated by commas, for example: mandateContent,attribute2.

The following table contains the description of currently supported attribute providers:

<table>
<thead>
<tr>
<th>Plugin Name</th>
<th>EHvdAttributeProvider</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td><a href="http://ehvptest.buergerkarte.at/RetreveAttribute">http://ehvptest.buergerkarte.at/RetreveAttribute</a></td>
</tr>
<tr>
<td>Description</td>
<td>Used in the health domain.</td>
</tr>
<tr>
<td>Parameter</td>
<td>SignedDocAttributeRequestProvider</td>
</tr>
</tbody>
</table>
Please note that for the exact URLs of attribute providers covering particular attributes and use cases can be found on Member State Wiki Page for Austria at the STORK 2 web site\(^2\).

This page contains other information such as SAML signing certificates of these attribute providers, which should be installed at local instance.

### 3.6.5 Additional general settings

This optional section enables service providers to provide customized, service provider application specific authentication block and citizen card environment selection form.

The option **Hide bPK/wbPK from AuthBlock** enables exclusion of bPK and wbPK from the authentication block presented to the user to sign.

The field **AuthblockText** should contain the customized service provider specific text. This text will be signed from the user in the course of authentication. The text may consist of characters, numbers and punctuation marks. It can contain the following keywords too:

- #NAME# - will be replaced with first and family name of the user
- #DATE# - will be replaced with the actual date
- #TIME# - will be replaced with the actual time

The example **AuthblockText** is represented with the following figure:

![With this action I, #NAME#, confirm at #DATE# on #TIME# o’clock the reception of the information package.](https://www.eid-stork2.eu/wiki/index.php/AT)
This text setting will be replaced into the following text and signed by the user in the course of authentication:

| With this action I, John Doe, confirm at 05.01.2014 on 10:00 o’clock the reception of the information package. |

*Figure 10: Example transformation result of customized AuthTextBlock*

### 3.6.6 CCE-Selection and Send-Assertion Templates

The following table contains the description of these parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>CCE-Selection Template</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&lt;uploaded file&gt;</td>
</tr>
<tr>
<td>Description</td>
<td>This parameter allows setting of the application specific CCE selection template. This template may be additionally customized and uploaded using this box. If applied, this template should be additionally checked for security (e.g. XSS attack).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Send-Assertion Template</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>&lt;uploaded file&gt;</td>
</tr>
<tr>
<td>Description</td>
<td>This option allows setting of application specific template for additional authentication request in the case of single sign-on login.</td>
</tr>
</tbody>
</table>

*Table 24: CCE-Selection and Send-Assertion Templates*

Summarized, Figure 11 shows the example configuration of service provider application.
Figure 11: Example configuration of service provider application
3.7 Other functionality of MOA-ID web configuration interface

The option **User management** is used to set up new users of MOA-ID web configuration interface. These users will be able to login to the configuration interface and administer applications and settings. Figure 12 shows the interface used to create new users.

![Figure 12: Creating new users](image)

3.8 Configuring BKUOnline

BKUOnline is separate application serving as an online Citizen Card Environment. It is used in the authentication process as a component loaded in the user’s browser as a Java applet. Based on the request initiated by service provider’s application it establishes the communication with a user’s citizen card (smart card environment) and performs all authentication and signature related tasks.

3.8.1 Adjusting configuration file

The file `/var/lib/tomcat7/webapps/bkuonline/WEB-INF/conf/configuration.xml` contains the basic configuration elements that have to be reviewed before the BKUOnline is used in a productive environment.

**Whitelist** section of this file contains the list of allowed **DataURLs**, e.g. the hosts which are allowed to call this instance of BKUOnline.

```xml
<Whitelist>
  https://www.formularservice.gv.at/BKU/.*,
  https://127.0.0.1/.*,
  https://localhost/.*,
  https://testvidp.buergerkarte.at/.*
</Whitelist>
```

![Figure 13: BKUOnline whitelist hosts](image)
Figure 13 shows the example configuration of whitelist section. The entries can be based on regular expressions.

Option **SSL/disableAllChecks** is used to deactivate all SSL certificate checks of calling instance. In production systems should be set to **false**.

### 3.8.2 Configuring SSL certificates

The relevant directories here are `certStore/toBeAdded` and `trustStore`, located under `/var/lib/tomcat7/webapps/bkuonline/WEB-INF/classes/at/gv/egiz/bku/certs`

The directory `trustStore` contains the trusted certificates of Certification Authorities. The certificates of additional CAs should be copied to this location.

The directory `certStore/toBeAdded` is a container for all certificates that should be used in the process of building and verifying a certificate chain, including the SSL certificate of BKUOnline and MOA-ID host system. Therefore, if the server SSL certificate fails to complete a chain, the missing certificates should separately be copied to this directory. After the restart, the application will check this directory, read and move its contents.

If some certificates remain in the `toBeAdded` folder after the application or server are restarted, the server administrator should check the logs for the errors. One of the reasons might be an incorrect certificate format. In this case, the file should be converted and again copied to `toBeAdded` directory.

### 3.8.3 Logging configuration

The file `/var/lib/tomcat7/webapps/bkuonline/WEB-INF/classes/log4j.properties` contains logging configuration for BKUOnline module. Relevant for the configuration are the sections `STDOUT appender` and `FILE appender`, as well as `log4j.rootLogger` setting. The logging can be customized by enabling a higher logging level (e.g. `DEBUG` instead of `INFO`) and by adjusting file paths to fit the local path and configuration needs.

### 3.8.4 DocumentService configuration

In the file `/var/lib/tomcat7/conf/moa-id/moa-id.properties` the property `stork.documentservice.url` is set which provides the URL to the DocumentService instance on V-IDP. For example:

```
```
4 Manual V-IDP Installation

This section describes the manual deployment and configuration of a V-IDP system, which is relevant if the provided Virtual Appliance is not used. The following steps summarize all needed components and settings that have to be adjusted.

For a reference to specific configuration options, please consider the description given in Section 3: Configuration of V-IDP and its components.

4.1 System requirements

Please ensure that you have Apache Tomcat 6 or 7 installed on the system, as well as an updated Java SE 1.6 or newer. The location of Java SE will be further referred as $JAVA_HOME.

For this tutorial, we assume that all included .war files are deployed in a single Tomcat instance. In case an individual deployment scenario requires separate instances, we advise to reproduce the following steps on each involved instance.

4.2 Package contents

The deployment package of MOA-ID contains the following directories, as described in Table 25: Contents of the deployment archive.

<table>
<thead>
<tr>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conf</td>
<td>Configuration directories of MOA-ID and its components</td>
</tr>
<tr>
<td>doc</td>
<td>Documentation, including this manual</td>
</tr>
<tr>
<td>endorsed</td>
<td>Additional libraries</td>
</tr>
<tr>
<td>ext</td>
<td>Additional external libraries</td>
</tr>
<tr>
<td>source</td>
<td>Source code of Demo Online Application</td>
</tr>
<tr>
<td>tomcat</td>
<td>Apache Tomcat startup scripts</td>
</tr>
</tbody>
</table>

*Table 25: Contents of the deployment archive*

The layout of conf directory is as describe in Table 26: Contents of the configuration directory in the deployment archive.

<table>
<thead>
<tr>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>moa-id</td>
<td>Configuration of MOA-ID</td>
</tr>
<tr>
<td>moa-id-configuration</td>
<td>Configuration of MOA-ID web interface</td>
</tr>
<tr>
<td>moa-spss</td>
<td>Configuration of MOA-SPSS module</td>
</tr>
</tbody>
</table>

*Table 26: Contents of the configuration directory in the deployment archive*
4.3 Create MySQL databases

MySQL is the preferred way to store data of the V-IDP. Other RDMBS might be used as a backend storage, however, only MySQL is tested and supported for our purpose.

In order to deploy MOA-ID it is necessary to configure a MySQL database user and to create the databases, as shown in Table 7: Preconfigured MySQL databases.

```
mysql -u root
create database `moa-id-config`;
create database `moa-id-session`;
create database `moa-id-statistics`;
use mysql;
create user 'newuser'@'localhost' identified by 'password';
grant all privileges on `moa-id-config`.* TO 'newuser'@'localhost';
grant all privileges on `moa-id-session`.* TO 'newuser'@'localhost';
grant all privileges on `moa-id-statistics`.* TO 'newuser'@'localhost';
flush privileges;
quit
sudo /etc/init.d/mysql stop
sudo /etc/init.d/mysql start
```

*Figure 14: Setting up MySQL*

Figure 14: Setting up MySQL illustrates how to create necessary databases and assign the database user with appropriate rights to use that databases. The username, password and database names are used further in MOA-ID configuration, as shown in the following subsections.

It should be noted that it is not necessary to execute further actions in the terms of database schema initialization. The databases will be automatically populated during the MOA-ID deployment and initialization.

4.4 Copy and set up MOA-ID configuration files

The contents of the `conf` directory, described in Table 25: Contents of the deployment archive and Table 26: Contents of the configuration directory in the deployment archive should be copied to an appropriate location, e.g. the Tomcat configuration directory. Furthermore, the contained files should be adjusted to reflect the local configuration.

The example configuration is described in “Section 2.4: MOA-ID configuration files, Section 3.2: Preparing basic MOA configuration” as well as in “Section 3.3: Preparing MOA configuration interface”. The Tomcat directory layout in Debian is summarized in “Section 2.2 Apache Tomcat instance”
4.5 Create and populate MOA-ID Keystore

The Java keystore (JKS) should be created and populated with trusted certificates of other interacting peers (PEPS, SP etc). It should contain the private key used to sign outgoing messages. The interacting peers should also install the certificate used to sign the messages.

The example configurations of this step are given in “Section 2.4: MOA-ID configuration files and Section 3.4: Configuring specific Stork support for MOA”.

The list at STORK 2 website³ of other PEPS/VIDP interacting parties and updated certificate information can be consulted for further details.

Additionally, KeyStore Explorer⁴ can be used for manipulation over Java Key Store files.

4.6 Configure environment variables

As demonstrated in “Section 2.3: Setting up and running Apache Tomcat instance”, the configuration of Tomcat instance should be set up to contain reflect the local configuration generated in previous steps.

4.7 Copy endorsed files to Tomcat’s endorsed directory

The files contained in directory endorsed should be copied to the Tomcat’s endorsed directory. These are libraries used by MOA-ID and its components.

4.8 Install JCE Policy files

In order to be able to support cryptographic configuration and options of IAIK libraries, it is also necessary to install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files from the Java web site⁵.

These files should be installed under $JAVA_HOME/lib/security, or in the location specified in README.TXT file distributed with the JCE archive.

4.9 Copy ext files to ext directory of Java distribution

The files from ext directory should be copied to $JAVA_HOME/jre/lib/ext directory of Java distribution. These are the IAIK libraries used for cryptographic functions.

---
³ https://www.eid-stork2.eu/wiki/index.php/WP4_-_MS_technical_information
⁴ http://keystore-explorer.sourceforge.net/
⁵ http://www.oracle.com/technetwork/java/javase/downloads
4.10 Deploy included WAR files

The following .war files contained in the archive should be deployed:

- bkuonline.war
- moa-id-auth.war
- moa-id-configuration.war
- DocumentService.war

The deployment can be done by copying the files to Apache Tomcat’s webapps folder. The structure of Tomcat installation under Debian is described in “Table 2: Relevant Apache Tomcat paths”.

Please adjust the naming schema of these files to reflect your local preferences. As reference you might use “Table 3: Paths and URLs of deployed applications”.

4.11 Configure DocumentService

Configure the DocumentService provided by Iceland as described here: https://www.eid-stork2.eu/wiki/images/9/9f/DTL_deployment.docx

Note: In some cases it is not sufficient to add the MySQL connector to your Tomcats lib folder. Instead, it also has to be added to Tomcat’s endorsed folder.

4.12 Configure BKUOnline and disable support for Austrian Test Cards

Based on the example given in “Section 3.8: Configuring BKUOnline you can configure BKUOnline”. Production systems must enable the SSL check.

4.13 Install a production SSL certificate

The production SSL certificate should be installed and enabled in the Apache Webserver and Apache Tomcat daemons.

4.14 Setup general settings of MOA-ID

As demonstrated in “Section 3.2: Preparing basic MOA configuration, it is necessary to setup basic configuration of MOA-ID”.

4.15 Initialize and setup MOA-ID web configuration interface

As demonstrated in “Section 3.3: Preparing MOA configuration interface, it is necessary to configure and enable web access for MOA-ID configuration interface”.

4.16 Setup MOA-ID and service provider application via web interface

Before the configuration of MOA-ID applications and general settings is approached, it is necessary to establish local MOA-ID configuration interface user. This is done by deactivating the login option in the configuration file moa-id-configtool.properties within the moa-id-configuration directory. For this purpose, the option general.login.deaktivate should be set to true. After the application has been restarted, users are able to access the configuration application without providing user credentials.

“Figure 12: Creating new users” from “Section 3.7 shows the interface used to create new web user”. In the default configuration and for the standard usage, all the fields under User data
should be filled in, all checkboxes under subsection Rights and roles enabled and in the Login section BPK field might be left blank.

After the users are created, general.login.deaktivate should be set again to false and the application restarted.

As demonstrated in “Section 3.5: General MOA configuration” and “Section 3.6: Setting up particular service provider”, it is furthermore necessary to perform the general MOA-ID setup, create web interface users and create service provider application.

4.17 Security

Last, but not least, you should check you internal organizational security guidelines and requirements, the recommendations of operating system and software platform used and harden your system.

4.18 Testing credentials for mobile citizen card environment

In order to be able to test authentication by using the mobile phone authentication credentials, the configuration specific to the application need to be adjusted.

This is done by selecting Interfederation on the left menu, then clicking to the particular application, and changing the content of its Mobile CCE field to https://test1.a-trust.at/https-security-layer-request/default.aspx, as shown in the following figure:

![Application settings](image)

This way the users will be able to use the following credentials:

- Phone number: 103011122334455
- Password: 123456789
- TAN: 123456

Please note that this will disable usage of real mobile phone credentials. If you want to accept the real credentials under Austrian HandySignatur, the field Mobile CCE should be set to https://www.handy-signatur.at/mobile/https-security-layer-request/default.aspx.
5 Setting up the signature module

In order to enable Austrian citizens to create signatures on a V-IDP system you have to deploy two .war packages:

- oasis-dss.war
- signature-web-at.war

5.1 Prerequisites

- Installed Apache Tomcat, this tutorial refers to Tomcat version 7.0.54
- Installed Java >= 1.6
- Installed MySQL

5.2 OASIS DSS

This section describes the installation and configuration of the OASIS-module. The module receives OASIS-requests and forwards them to the OASIS module of a different country or the local Service Provider Interface (SPI). The module needs a database and an application server to run. It is required that the oasis-dss module and signature-web-at access the same MySQL database.

5.2.1 Building the OASIS DSS module

This project is a Maven 3 project. Each OASIS DSS module is associated to a service implementing the SPI. In order to add the Austrian SPI implementation to the OASIS DSS module, it is necessary to add/remove the respective SPI implementation modules to the /path/to/svn/STORK2/branches/wp4_signature/signature/oasis-dss/pom.xml.

The Austrian solution requires the following modules:

```xml
<module>../SPI-Implementations/Common/signature.verification.soap.client</module>
<module>../SPI-Implementations/at/SigningApp-AT</module>
```

*Figure 16: Modules required for Austrian SPI implementation*

If present, please remove the following module, as it is only required when deploying the Reference signature solution:

```xml
<module>../SPI-Implementations/ref/spi-reference</module>
```

*Figure 17: Modules required for reference SPI implementation*

Open a console and go to /path/to/svn/STORK2/branches/wp4_signature/signature/oasis-dss/. Run the command “mvn install”. If you are using Glassfish instead of Tomcat run “mvn install -P glassfish”.
5.2.2 Configuring the database

It is recommend to use a MYSQL database. First a database has to be created. This document refers to "stork2_signatures" as database name, and uses the MySQL user "stork" with password "stork". Instructions on how to create a new database can be taken from the figure below. A new database can be created by using the command "create database stork2_signatures". The creation of the required tables is done automatically.

![Figure 18: Creating the database](image)

The name of the database also needs to be configured on the application server. The required steps will be explained in 5.2.4.1. Further the project /path/to/svn/STORK2/branches/wp4_signature/signature/SPI-Implementations/Common/stork-database handles the database communication. In case of using a different database name the file stork-database/src/main/resources/META-INF/persistence.xml needs to be updated.

5.2.3 Configuring the Oasis module

Each OASIS module is associated with one service that implements the SPI given in /path/to/svn/STORK2/branches/wp4_signature/signature/oasis-dss-api/src/main/java/eu/stork/signature/spi/ISignature.java. So for every signature service you want to deploy, you have to deploy an own oasis module (eg. in case of deploying the V-IDP you have to deploy an OASIS module using the Austrian SPI, if you also choose to use the SD-DSS reference applet for signing you have to deploy a second OASIS module). Instructions on how to configure which SPI is used within the oasis module can be found in 5.2.1.

The following figure shows an example configuration file with only one other OASIS module (or country) configured. In this tutorial the configuration file is saved as $CATALINA_BASE/conf/oasis-dss.properties.

**Important:** You also have to configure a forward_dss_<Your home country> URL for your home country.
# Configure the URL to the document transfer interface

download_servlet_url=http://<yourserver>/<your oasiscntext>/DSSService

userds_servlet_url=http://<yourserver>/<your oasiscntext>/DSSUserServlet

# Define the home country the DSS interface is running in
home_country=AT

# Define forward_dss_COUNTRYCODE=URL to forward requests to
# COUNTRYCODE to the specified address
forward_dss_AT=http://<yourserver>/<your oasiscntext>/DSSService
forward_dss_<another country code>=http://<another oasiscntext>/DSSService

# Define forward_userdss_COUNTRYCODE=URL to forward requests to
# COUNTRYCODE to the specified address
forward_userdss_AT=http://<yourserver>/<your oasiscntext>/DSSUserServlet?requestId=
forward_userdss_<another country code>=http://<another oasiscntext>/DSSUserServlet

# Behaviour if a request with an already existing request_id is received (FAIL | OVERWRITE)
duplicate_request_policy=FAIL

# Define the timeout for DSSService connections (in ms)
dss_service_request_timeout=120000

# timeout for storing SignRequests and corresponding #SignResponses in database (in seconds), default is 1800 (30 #minutes)
request_storage_timeout=1800

Figure 19: Configuring oasis-dss

5.2.4 Deployment in Tomcat

5.2.4.1 Configuring Tomcat

Please add the following block to $CATALINA_BASE/conf/context.xml (with adapted values for "username", "password" and in case you chose a different database name also "url"):

```xml
<Resource name="jdbc/STORK2_signatures" auth="Container" type="javax.sql.DataSource" maxActive="100" maxIdle="30" maxWait="10000"
 username="stork" password="stork"
driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost:3306/stork2_signatures" />
```

Figure 20: Extending Tomcat context.xml
Please add the following block to `<GlobalNamingResources>` in $CATALINA_BASE/conf/server.xml (with adapted values):

```xml
<Resource auth="Container"
    driverClassName="com.mysql.jdbc.Driver"
    maxActive="100" maxIdle="30" maxWait="10000"
    name="jdbc/STORK2_signatures" password="stork"
    type="javax.sql.DataSource"
    url="jdbc:mysql://localhost:3306/stork2_signatures"
    username="stork"/>
```

*Figure 21: Extending Tomcat server.xml*

When starting Tomcat pass the following configuration parameter:

```
-Dconfig_oasis_dss="path/to/configuration/file/oasis-dss.properties"
```

You can do this by extending the variables definition given in Figure 3. In the example below `oasis-dss.properties` is located in $CATALINA_BASE/conf folder.

```bash
OASIS_CONFIG=-Dconfig_oasis_dss=$CATALINA_BASE/conf/oasis-dss.properties
JAVA_OPTS="$LOGS_BASE $FILE_ENCODING $LOGGING_OPT $MOA_ID_CONFIG $MOA_SPSS_CONFIG $MOA_ID_CONFIG_TOOL -Dsun.security.ssl.allowUnsafeRenegotiation=true $RAND_FILE -XX:PermSize=64m -XX:MaxPermSize=786m $MOA_ID_STORK $OASIS_CONFIG "
```

*Figure 22: Extending Tomcat variable definition*

### 5.2.4.2 Application Deployment in Tomcat

Copy the `path/to/svn/STORK2/branches/wp4_signature/signature/oasis-dss/oasis-dss/target/oasis-dss.war` to `$CATALINA_BASE/webapps`. Do not forget that you also have to deploy the `.war` file for the SPI implementation as described in 5.3.4. Make sure that the context path matches with the context paths given in `oasis-dss.properties`.

### 5.3 Austrian SPI Implementation

The Austrian signature solution consists of the following projects:

- **signature-app-at** provides the Austrian implementation of the `eu.stork.signature.spi.ISignature` interface. It receives instances of `SignRequest`, temporarily stores them in a database and later retrieves the resulting signature and returns it back to the requesting VIdP. In case of signature verification, **signature-app-at** communicates with `signature.verification.soap.client`, converts the received verification results in a Stork 2.0 compliant `VerifyResponse` instance and returns it back to the calling VIdP.
• **signature-web-at** provides the web application that interacts with the user in order to sign a document. It provides a graphical user interface for selecting the preferred Citizen Card Environment and create the signature. It integrates the Mobile Phone Signature, Mocca online and Mocca local.

• **security-layer-mapping** converts instances of Stork 2.0 `SignRequest` to `CreateXMLSignatureRequestType`, vice versa it maps `CreateXMLSignatureResponseType` to instances of Stork 2.0 `SignResponse`. The *Security Layer* defines the interface, how applications can communicate with Austrian Citizen Card Environments.

• **signature.verification.soap.client** represents a SOAP client is based on AXIS. It is a client for the Austrian Signature Portal.

• **stork-database** handles the persistence storage required for the Austrian solution to work. Currently, a MySQL database is used. In order to access database functionality use the class `eu.stork.signature.database.RequestManagerSessionBeanLocal`.

• Furthermore, you will need a **BKUOnline** instance. You can reuse the instance in use with the V-IDP hosted at your premises.

The project **signature-app-at** has to be added as dependency to the OASIS DSS module as described in 5.2.1, while **signature-web-at** will be deployed separately.

### 5.3.1 Building the Austrian signature module

This project is a Maven 3 project. Open a console and go to `/path/to/svn/STORK2/branches/wp4_signature/signature/SPI-Implementations/at/`. Run the command “mvn install”. If you are using Glassfish instead of Tomcat run “mvn install -P glassfish”.

### 5.3.2 Configuring the SPI implementation

Create a configuration file for **signature-app-at**. In this document the configuration file is saved as `$CATALINA_BASE/conf/config_signature_app_at.properties`.

```
# Id of this SPI implementation
# REQUIRED
service_id=eu.stork.signature-app-at

# Url to the signature web service
# REQUIRED
web_service_url=http://<your server url>/signature-web-at

# Servlet and param name for signing
# REQUIRED
postfix_sign=/login?reqid=

# milliseconds that indicate how long the app should poll
# for a response from the signature web service
# OPTIONAL, default is 4 minutes
polling_duration=240000

# ms that app should pause before polling again
# OPTIONAL, default is 2000ms
polling_interval=2000

#SOAP endpoint for verify
SOAP_EndPoint=https://www.buergerkarte.at/test-signature-verification/services/VerificationService/
```

*Figure 23: Configuring signature-app-at*
5.3.3 Configuring the web service

Create the configuration file for signature-web-at. In this example the configuration file is saved as $CATALINA_BASE/conf/config_signature_webservice_at.properties.

```properties
# server url, should point to your server
# REQUIRED
app_url= https://<your server url>
# application context, change this field in case you changed the context where you deployed signature-web-at
# REQUIRED
app_context=/signature-web-at
# name of signature servlet
# REQUIRED
signature_context=/signature
# context and parameters where signature/attachment files can be accessed
# REQUIRED
filehoster_context=/getfile?reqid=
# string that should be written in front of the request id that represents the filename of the xml file with the signature,
# OPTIONAL, default is "signature_
signature_file_prefix=signature_
# Url where BKU returns Signature and IdentityLink response
# REQUIRED
dataurl= https://<your server url>/signature-web-at/dataurl
# Online BKU Url
# REQUIRED
url_bku_online=http://<your server url>/bkuonline/http-security-layer-request
# Mobile Phone Signature Url
# REQUIRED
# Local BKU Url
# REQUIRED
url_bku_local=http://127.0.0.1:3495/http-security-layer-request
# milliseconds that indicate how long the webservice should poll,
# in case a request has not yet been received from the SPI
# OPTIONAL, default is 1 minute
polling_duration=60000
# ms that webservice should pause before polling again
# OPTIONAL, default is 2000ms
polling_interval=2000
```

Figure 24: Configuring signature-web-at

Please note that url_bku_online should point to the BKUOnline instance deployed at local premises. Details about that are provided in Section 3.8. Additionally, if you want to use test credentials for the Mobile CCE, further details are provided in Section 4.18. In case of using test credentials for Mobile CCE, note that you have to configure url_bku_mobile to https://test1.a-trust.at/https-security-layer-request/default.aspx

**Note:** The url defined as dataurl has to use SSL, otherwise the Austrian Citizen Card won’t return a response. For using the Austrian Mobile Phone Signature Solution, please verify that your CA’s root certificate is supported by A-Trust (Information on the installation of the production SSL certificate can be found in section 4.13). A list of supported Root CA certificates is available here: https://labs.a-trust.at/developer/ShowArticle.aspx?id=106
5.3.3.1 Enabling PAdES support

To enable PAdES signatures the Austrian solution uses PDF-AS, which requires additional configuration. A basic configuration is located at /path/to/svn/STORK2/branches/wp4_signature/signature/SPI-Implementations/SigningApp-AT/src/main/resources/pdf-as-defaultconfig.zip. Extract the configuration to a location of your choice and specify the location using the “config_pdf_as” system property (see section 5.3.4.1). In this example the configuration files are extracted to $CATALINA_BASE/conf/.

5.3.4 Deployment in Tomcat

5.3.4.1 Configuring Tomcat

In order to provide Tomcat with the location of the two configuration files config_signature_webservice_at.properties and config_signature_app_at.properties the corresponding paths have to be passed as system property. Therefore, the following parameters can be appended to the Tomcat instance:

-Dconfig_signature_app_at= "path/to/configuration/file/config_signature_app_at.properties"

-Dconfig_signature_webservice_at= "path/to/configuration/file/config_signature_webservice_at.properties"

-Dconfig_pdf_as= "path/to/configuration/folder for extracted pdf-as-config"

Furthermore, you have to specify a temporary directory. You can configure a “stork.database.tmpdir” using -Dstork.database.tmpdir= "path/to/tmpdir" Alternatively, or if this value is not set, the “java.io.tmpdir” directory is used:

-Djava.io.tmpdir= "path/to/tmpdir"

You can do this by extending the variable definition given in Figure 3. In the example below the configuration files are located in $CATALINA_BASE/conf and the sample configuration for PDF-AS has been extracted to $CATALINA_BASE/conf/pdf-as-defaultconfig.

```
SIGNATURE_WEB_CONFIG=-Dsignature_webservice_at=$CATALINA_BASE/conf/config_signature_webservice_at.properties
SIGNATURE_APP_CONFIG=-Dconfig_signature_app_at=$CATALINA_BASE/conf/config_signature_app_at.properties
PDFAS_CONFIG=-Dconfig_pdf_as=$CATALINA_BASE/conf/pdf-as-defaultconfig
TEMP_DIR_PARAM=-Djava.io.tmpdir=$CATALINA_BASE/temp

JAVA_OPTS="$LOGS_BASE $FILE_ENCODING $LOGGING_OPT $MOA_ID_CONFIG $MOA_SPSS_CONFIG $MOA_ID_CONFIG_TOOL -Dsun.security.ssl.allowUnsafeRenegotiation=true $RAND_FILE -XX:PermSize=64m -XX:MaxPermSize=786m $MOA_ID_STORK $OASIS_CONFIG $SIGNATURE_WEB_CONFIG $SIGNATURE_APP_CONFIG $TEMP_DIR_PARAM $PDFAS_CONFIG"
```

Figure 25: Extending Tomcat variable definition
5.3.4.2 Application Deployment in Tomcat

Copy signature-web-at.war to $CATALINA_BASE/webapps. Make sure that the context path matches with the context paths given in config_signature_app_at.properties and config_signature_webservice_at.properties.
6 References

[1] D4.4 First version of Technical Specifications for the cross border Interface. WP4 core team, October 2013

[2] D4.3 First version of Technical Design. WP4 core team, October 2014
