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 call identifier: CIP-ICT-PSP-2011-5
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

Deliverable Id :	
Deliverable Name :	
Status :	
Dissemination Level :	**<PU,CO,RE,PP>**
Due date of deliverable :	
Actual submission date :	
Work Package :	**WP 4**
Organization name of lead contractor for this deliverable :	
Author(s):	**Bojan Suzic**
Partner(s) contributing :	**AT-TUG**

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

Project co-funded by the European Community under the ICT Policy Support Programme

History

Version	*Date*	*Modification reason*	*Modified by*
0.1	30.3.2014	Initial documentation	Bojan Suzic
0.2	10.6.2014	Extending sections	Bojan Suzic

Table of contents

List of Figures

List of Tables

List of abbreviations

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

Executive summary

This document is targeting system administrators and operators, as well as the users of V-IDP system who will deploy or harden V-IDP systems at their local premises.

It provides quick reference on installation and configuration of V-IDP system. It covers the reference installation of preconfigured V-IDP system provided by Graz University of Technology and A-SIT Austria. Furthermore, the document explains the structure of V-IDP deployment archive, provides the steps necessary to execute during its separate deployment and configuration.

When you finish reading this document you should be able to install and configure V-IDP system, to gain better understating of its function and provide primary support in the case of troubleshooting.

1 Introduction

This document consists of two main parts.

The first part explains the structure and configuration of provided V-IDP reference system installation in the form of Debian Wheezy virtual machine.

The second part of the document provides the information on manual deployment of V-IDP system based on accompanying V-IDP deployment archive. It explains main steps involved in that process and provides hints and links to the settings already done and explained a the reference system.

For further questions, suggestions and comments please contact Austrian V-IDP team. Detailed contact informations are available at STORK 2 Wiki pages.

2 Configuration of V-IDP Virtual Machine

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

The virtual machine is based on Debian 7.4 distribution and contains optimal set of packages necessary 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

The client HTTP(S) request coming to the V-IDP system are served by Apache httpd front-end server, which is installed and configured in its standard Debian flavor. It contains additional mod_jk module, which interfaces with backend Apache Tomcat server. The Apache Tomcat server instance included in this reference system contains all relevant modules belonging to the V-IDP. Installed MySQL Server instance is used to store configuration, session and statistics related data. The installation is based on the one shipped with default Debian distribution, configured according to standard Debian practices.

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

The credentials used to login to the virtual machine are:

User: `stork`

Password: `st12knn`

2.1 Apache httpd instance

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

Path	Description
`/etc/apache2/`	Base configuration directory
`/etc/apache2/mods-enabled`	Enabled modules, links to configuration of modules
`/etc/apache2/ssl`	Location for SSL certificates
`/var/log/apache2`	Apache log files

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
```

The following figure provides the excerpt of configuration of **mod_jk** module.

```
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 **Apache Tomcat 7** instance containing the MOA applications. Table 2 provides the list and description of the most relevant paths of Tomcat installation.

Path	Description
/usr/share/tomcat7	Contains Tomcat binaries an libraries
/var/lib/tomcat7	Contains shared and common files, deployment and directory for endorsed files
/var/lib/tomcat7/webapps	Contains Austrian authentication middleware
/etc/tomat7	Configuration directory
/var/log/tomcat7	Log files

Table 2: Relevant Apache Tomcat paths

The Tomcat server contains web modules **ConfigurationInterface**, **moa-id-auth** and **bkuonline**, that basically build up the V-IDP system. They are by default deployed under **webapps** directory.

The following list briefly summarizes 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 provides overview on deployed applications:

Path	URL
bkuonline	https://testvidp.buergerkarte.at/bkuonline
moa-id-auth	https://testvidp.buergerkarte.at/moa-id-auth
moa-id-configuration	https://testvidp.buergerkarte.at/moa-id-configuration

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**.

Variable	Description
LOGS_DIR	Destination of log data
CATALINA_OUT	Log file containing Tomcat's standard output
LOGS_BASE	Base directory for other logs
LOGGING_OPT	Points to file containing logging configuration
FILE_ENCODING	Sets file encoding

MOA_ID_CONFIG	Location of base MOA-ID configuration file
MOA_ID_CONFIG_TOOL	Location of base MOA-ID web interface configuration file
MOA_ID_STORK	Location of STORK related files such as SAMLEngine configurations
MOA_SPSS_CONFIG	Points to MOA-SPSS configuration file
MOA_SPSS_TSL_HASHCACHE	Caching directory used by MOA-SPSS
JAVA_ENDORSED_DIRS	Path to endorsed directory

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.allowUnsafeRenegotiation=true $RAND_FILE
-XX:PermSize=64m -XX:MaxPermSize=786m $MOA_ID_STORK"

JVM_OPTS="$FILE_ENCODING;$LOGGING_OPT;$MOA_ID_CONFIG;$MOA_SPSS_CONFIG;$MOA_ID_CON
FIG_TOOL;-Dsun.security.ssl.allowUnsafeRenegotiation=true;$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_TRUST_STORE_TYPE $PARAM_SSL_DEBUG $MOA_ID_CONFIG_TOOL -
Dsun.security.ssl.allowUnsafeRenegotiation=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`**.

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

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 should not be normally changed.

Path	Description
`/etc/mysql`	The base directory for configuration of MySQL server
`/etc/mysql/debian.cnf`	Automatically generated configuration by Debian – should not be changed directly
`/etc/mysql/my.cnf`	MySQL server main configuration file
`/var/log/mysql`	MySQL server log location

Table 6: MySQL relevant paths

Both the username and password used to connect to MySQL server are `vidp`. The users are advised to change the password, as well to reset the root MySQL password. The later can be done by executing the following set of commands:

```
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 in Debian command line. The first part of the code is used to start MySQL server in safe mode. The middle part is then used to log in MySQL instance and set the passwords for **`root`** and **`vidp`** users. At the end, the instance is restarted in normal mode.

3 Configuration of V-IDP and its components

This section presents the configuration of V-IDP software components.

All references to specific files or directories in this section are based on reference installation provided in the form of V-IDP virtual machine, as introduced in Section 2: Configuration of V-IDP Virtual Machine. Therefore, these paths should be adjusted correctly in the case of specific configuration based on manual deployment and installation of modules.

3.1 MySQL configuration

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

Database	Description
`moa-id-config`	Various configuration settings
`moa-id-session`	Temporary session data
`moa-id-statistics`	Statistics on MOA-ID usage

Table 7: Preconfigured MySQL databases

Both the username and password used to connect to MySQL server are **`vidp`.**

Users are advised to change MySQL credentials and adjust server and access settings to conform to local deployment environment and security requirements.

3.2 Preparing basic MOA configuration

Before MOA instance is started, it is necessary to review and configure its configuration parameters available under **`/etc/tomat7/moa-id/moa-id.properties`**. This file provides 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 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:

Parameter		`configuration.moasession.key`
	Example	`MyKey123`
	Description	Passphrase used to encrypt the session data in database. Optional.
Parameter		`Configuration.monitoring.active`

	Example	true
	Description	Defines the status of monitoring servlet
Parameter		Configuration.monitoring.message.success
	Example	True
	Description	Success message sent in case of passing the tests
Parameter		Configuration.monitoring.test.identitylink.url
	Example	/test/idl/test_identitylink.xml
	Description	URL of IdentityLink used for testing
Parameter		configuration.advancedlogging.active
	Example	true
	Description	Defines the status of additional logging facilities (in database)

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.

Parameter		moasession.hibernate.dialect
	Example	org.hibernate.dialect.MySQLDialect
	Description	Dialect used to communicate with MOA-session data store
Parameter		moasession.hibernate.connection.url
	Example	jdbc:mysql://localhost/moa-id-session?charSet=utf-8&autoReconnect=true
	Description	URL to MOA-session database schema
Parameter		moasession.hibernate.connection.driver_class
	Example	com.mysql.jdbc.Driver
	Description	Driver used to access session database
Parameter		moasession.hibernate.connection.username
	Example	moasession
	Description	The user name used to access the session database
Parameter		moasession.hibernate.connection.password
	Example	moapassword
	Description	The password used to access the session database

Parameter	configuration.hibernate.dialect
Example	org.hibernate.dialect.MySQLDialect
Description	Dialect used to communicate with MOA-configuration data store
Parameter	configuration.hibernate.connection.url
Example	jdbc:mysql://localhost/moa-id-config?charSet=utf-8&autoReconnect=true
Description	URL to MOA-config database schema
Parameter	configuration.hibernate.connection.driver_class
Example	com.mysql.jdbc.Driver
Description	Driver used to access config database
Parameter	configuration.hibernate.connection.username
Example	moaconfiguser
Description	The user name used to access the database
Parameter	moasession.hibernate.connection.password
Example	moaconfigpassword
Description	The password used to access the database
Parameter	advancedlogging.hibernate.dialect
Example	org.hibernate.dialect.MySQLDialect
Description	Dialect used to communicate with data store for MOA advanced logging
Parameter	advancedlogging.hibernate.connection.url
Example	jdbc:mysql://localhost/moa-id-statistics?charSet=utf-8&autoReconnect=true
Description	URL to advanced logging database schema
Parameter	advancedlogging.hibernate.connection.driver_class
Example	com.mysql.jdbc.Driver
Description	Driver used to access advanced logging database
Parameter	advancedlogging.hibernate.connection.username
Example	moaloguser
Description	The user name used to access the advanced statistics database
Parameter	moasession.hibernate.connection.password
Example	moalogpassword
Description	The password used to access the advanced statistics database

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 under **`/etc/tomcat7/moa-id-configiguration/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.

Parameter		`general.login.deaktivate`
	Example	`false`
	Description	With this option login using web interface can be deactivated.
Parameter		`general.publicURLContext`
	Example	`https://testvidp.buergerkarte.at/ConfigurationInterface`
	Description	This is the URL of web configuration interface
Parameter		`general.defaultlanguage`
	Example	`de`
	Description	Defines default language for web configuration interface
Parameter		`general.mail.host`
	Example	`mail.testvidp.buergerkarte.at`
	Description	The mail server used to send the emails
Parameter		`general.mail.from.name`
	Example	`TestVIDP System`
	Description	The name of the sender of emails
Parameter		`general.mail.from.address`
	Example	`admin@testvidp.buergerkarte.at`
	Description	The *from* address used when sending emails from the host
Parameter		`general.mail.useraccountrequest.verification.subject`
	Example	`MOA-ID User verification`
	Description	Subject of verification email sent when opening account
Parameter		`general.mail.useraccountrequest.verification.template`
	Example	`mail/verification_template.html`
	Description	Location of the verification's email template
Parameter		`general.mail.useraccountrequest.isactive.subject`

	Example	MOA-ID User Activation
	Description	Subject of activation mail sent after the account is activated
Parameter		general.mail.useraccountrequest.isactive.template
	Example	mail/activation_template.html
	Description	Location of the activation's email template
Parameter		general.mail.useraccountrequest.rejected.template
	Example	mail/rejected_template.html
	Description	Location of the template used to inform the user about rejection or removal of its requested account.
Parameter		general.mail.createOArequest.isactive.subject
	Example	MOA-ID Service Provider Activation
	Description	Subject of activation mail for service provider (online application)
Parameter		general.mail.createOArequest.isactive.template
	Example	mail/oa_activation_template.html
	Description	Location to the template used for service provider activation email
Parameter		general.mail.admin.adress
	Example	admin@testvidp.buergerkarte.at
	Description	Email address of system admin
Parameter		general.mail.admin.subject
	Example	MOA-ID Status information
	Description	Subject of status information message
Parameter		general.mail.admin.adresses.template
	Example	mail/admin_template.html
	Description	Location of status information/change template sent to the admin

Table 10: Web interface configuration for MOA-ID

In the following table described are database specific configuration parameters:

Parameter		hibernate.dialect
	Example	org.hibernate.dialect.MySQLDialect
	Description	The SQL dialect used to store configuration
Parameter		hibernate.connection.url

	Example	jdbc:mysql://localhost/moa-id-config?charSet=utf-8&autoReconnect=true
	Description	URL used to access the database (configuration store)
Parameter		hibernate.connection.driver_class
	Example	com.mysql.jdbc.Driver
	Description	Driver class used to access the database
Parameter		hibernate.connection.username
	Example	Vidp
	Description	The username used to access the database
Parameter		hibernate.connection.password
	Example	Vidppass
	Description	The password used to access the database

Table 11: MOA-ID web configuration interface database configuration

3.4 Configuring specific Stork support for MOA

Common SAMLEngine developed in the course of STORK project is used for the signing and verification of STORK SAML messages. Its specific configuration files can be found under **/etc/tomcat7/moa-id/stork**.

The main file for defining sub-configurations located in this directory is **SamlEngine.xml**. This file defines and references instance configurations for each profile used in the application.

The following is snippet excerpt from **SamlEngine.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**. Each configuration in the system requires such instance designation and further definition of particular **SamlEngine** and **SignatureEngine** subsections. In SamlEngine configuration contained are the attribute definitions and other settings – there is

generally no need to adjust those files. On the other hand, **SignatureEngine** contains settings necessary to access keystore used to sign and verify SAML messages.

The following is snippet excerpt from that file:

```
<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

Generally nearly all the lines from the previous configuration snippet should be adjusted in order to reflect local configuration.

Referenced keystore should contain trust certificates from other trusted parties, such as PEPSes, and private key used to sign message delivered to the peers.

3.5 General MOA configuration

General configuration in MOA is accessed through web configuration interface of MOA-ID. By default, this interface is located at **https://vidp-domain.tld/moa-id-configuration**.

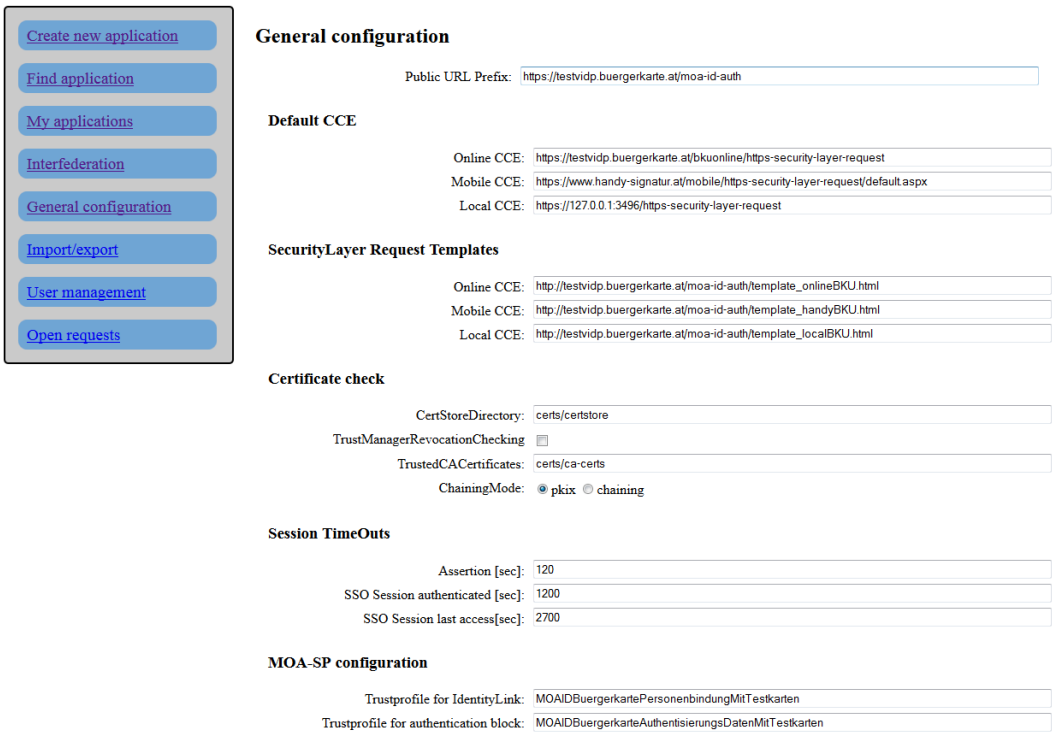

Figure 7: Example general configuration interface screen

Figure 7 shows the web page of configuration interface, located in general configuration section of MOA-ID.

The default login data for MOA-ID configuration interface are:

Username: `stork`

Password: `st12knn`

In order to access MOA-ID configuration it might be necessary to temporarily disable login functionality and create a new user. The more details are provided in Section 4.15.

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

3.5.1 Public URL Prefix

Parameter		`Public URL Prefix`
	Example	`https://testvidp.buergerkarte.at/moa-id-auth`
	Description	URL-Prefix of MOA-ID instance. This setting is used for automatic generation of forms and information and must be defined.

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.

Parameter		`Online CCE`
	Example	`https://testvidp.buergerkarte.at/bkuonline/https-security-layer-request`
	Description	URL to the Online-CCE instance
Parameter		`Mobile CCE`
	Example	`https://www.handy-signatur.at/mobile/https-security-layer-request/default.aspx`
	Description	URL to the Mobile-CCE instance
Parameter		`Local CCE`
	Example	`https://127.0.0.1:3496/https-security-layer-request`
	Description	URL pointing to the locally installed and executed CCE instance

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 out of scope of this setup.

Parameter		`Online CCE`
	Example	`http://testvidp.buergerkarte.at/moa-id-auth/template_onlineBKU.html`
	Description	SL Template for communication with Online-CCE
Parameter		`Mobile CCE`
	Example	`http://testvidp.buergerkarte.at/moa-id-auth/template_handyBKU.html`
	Description	SL Template for communication with Mobile-CCE
Parameter		`Local CCE`
	Example	`http://testvidp.buergerkarte.at/moa-id-auth/template_localBKU.html`
	Description	SL Template for communication with local CCE instance

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`**.

Parameter		`CertStoreDirectory`
	Example	`certs/certstore`
	Description	Path name to the directory containing trust store used in the course of TLS server based certificate check.
Parameter		`TrustManagerRevocationChecking`
	Example	`false`
	Description	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`.
Parameter		`TrustedCACertificates`
	Example	`certs/ca-certs`

	Description	Path to the directory containing trusted CA certificates. In the course of TLS server based certificate check these CA would be considered as trusted.
Parameter		`Chaining Mode`
	Example	`pkix`
	Description	This option defines if the certificate check method would be based on `chaining` or on RFC 3280 `pkix.`

Table 15: Configuration of trusted certificates

3.5.4 Session timeouts

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

Parameter		`Assertion`
	Example	`120`
	Description	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.
Parameter		`SSO Session authenticated`
	Example	`1200`
	Description	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.
Parameter		`SSO Session last access`
	Example	`2700`
	Description	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.

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.

Parameter		`Trustprofile for Identity Link`
	Example	`MOAIDBuergerkartePersonenbindungMitTestkarten`

	Description	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.
Parameter		`Trustprofile for authentication block`
	Example	`MOAIDBuergerkarteAuthentisierungsDatenMitTestkarten`
	Description	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.
Parameter		`Transformations for authentication block`
	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 Service Name:
	Example	
	Description	
Parameter		SSO Service Target
	Example	
	Description	
Parameter		SSO Auth Block Text
	Example	
	Description	

Table 19: Configuration of Single Sign-On parameters

3.5.8 Secure Identity Accross Borders Linked

The parameters from this section specify the settings directly relevant for Stork deployment.

In this section it is possible to define **standard QAA-Level** of requests, which sets minimal requirements for QAA.

Further, for each relevant **country** the appropriate **C-PEPS URL** should be entered.

The next available setting is **configuration of attributes**. In this subsection defined are generally supported STORK attributes, such as `eIdentifier`, `givenName` and `others`. The field **mandatory** is used to mark the attributes which must be included in the response from PEPS. Figure 8 shows the example configuration of attributes.

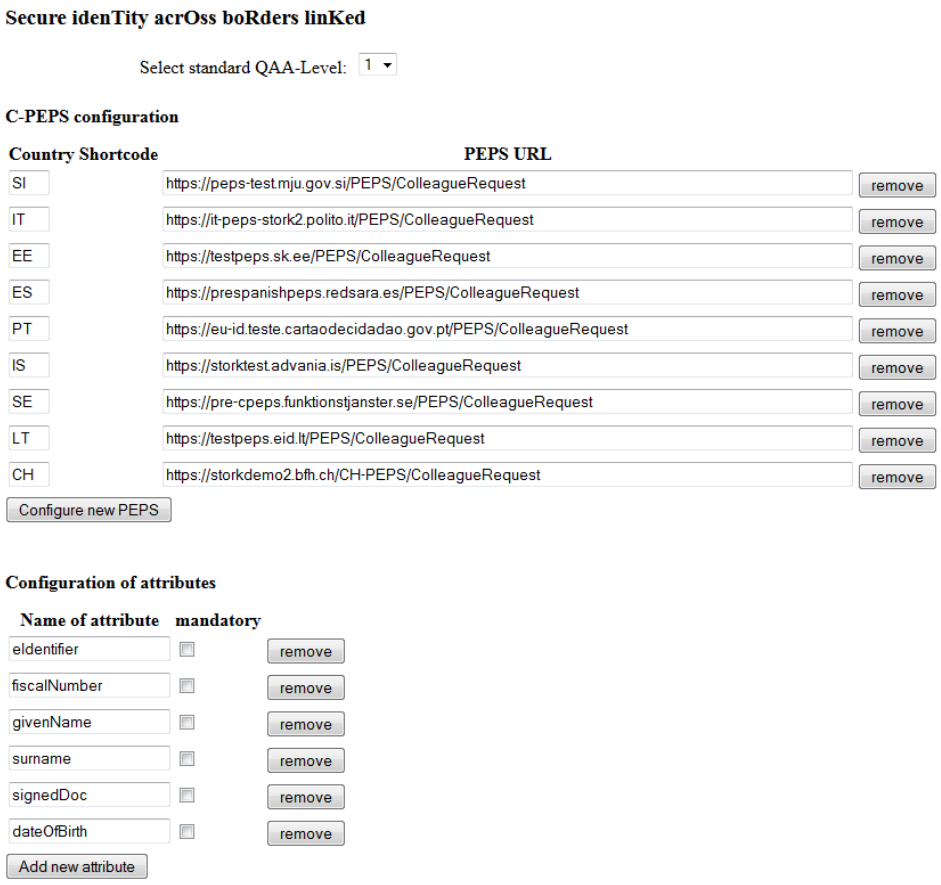

Figure 8: Example configuration of C-PEPS URLs and attributes

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 under:

`/etc/tomcat/moa-id/transforms/TransformsInfoAuthBlockTable_DE_new.xml`.

3.6 Setting up particular service provider

The configuration of service provider[1] is initiated from the main web-interface menu, under the option **`Interfederation > STORK VIDP`**. The list of available applications can be invoked through the option **`Interfederation`**. Furthermore, the service provider configuration is

[1] In this document referred also as an *online application*

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.

Parameter		`Online-Application is activated`
	Example	`true`
	Description	Allows to activate and deactivate service provider application.
Parameter		`Unique identifier (PublicURLPrefix):`
	Example	`https://peps-test.gov.tld/SP/ReturnPage`
	Description	This parameter is used to distinguish service provider applications and find/retrieve the specific one. The identificator should contain at least the URL prefix of externally visible and reachable service provider application
Parameter		`Name of the Online-Application:`
	Example	`Demo SP`
	Description	User defined name of service provider. It is shown during user authentication process.
Parameter		`Private sector application`
	Example	`true`
	Description	This parameter should be set to true for STORK service providers
Parameter		`Identification number:`
	Example	`STORK IT`
	Description	This parameter is mandatory and should be used to to set the SP country.

Table 20: Basic settings for service provider application

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 4.17.

Parameter		`Online CCE`
	Example	`https://testvidp.buergerkarte.at/bkuonline/https-security-layer-request`

	Description	URL to the application specific Online-CCE instance. If no instance is configured, the default one from the MOA-ID general configuration is applied.
Parameter		`Mobile CCE`
	Example	`https://www.handy-signatur.at/mobile/https-security-layer-request/default.aspx`
	Description	URL to the application specific Mobile-CCE instance. If no instance is configured, the default one from the MOA-ID general configuration is applied.
Parameter		`Local CCE`
	Example	`https://127.0.0.1:3496/https-security-layer-request`
	Description	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.
Parameter		`KeyBoxIdentifier`
	Example	`SecureSignatureKeypair`
	Description	Configures keypair used to sign the authentication block. Default value is SecureSignatureKeypair.
Parameter		`SecurityLayerTemplates (Legacy Request)`
	Example	`false`
	Description	Using this option it is possible to define additional three SecurityLayer templates. In STORK related deployments this selector should be left unchecked.

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.

Parameter		`Mandates`
	Example	`true`
	Description	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.
Parameter		`Profile`
	Example	`Gesundheit,Zustellung,ERsB,GeneralvollmachtBilateral`
	Description	This element contains the comma-separated list of mandating profiles. The full list is available under https://vollmachten.stammzahlenregister.gv.at/mis.
Parameter		`Allow mandated login only`

	Example	`false`
	Description	Allows setting the possibility to accept authentication based on powers only.

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 service provider application.

`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:

Plugin Name		`EHvdAttributeProvider`
	Example	`http://ehvdtest.buergerkarte.at/RetreveAttribute`
	Description	Used in the health domain.
Parameter		`SignedDocAttributeRequestProvider`
	Example	`http://testvidp.buergerkarte.at/RetreveAttribute`
	Description	Used to support document signing using `signDoc` attribute.
Parameter		`MISAttributeRequestProvider`
	Example	`http://mistest.buergerkarte.at/moa-id-auth/MISProvider`
	Description	Used to enable support of authentication in powers.
Parameter		`StorkAttributeRequestProvider`
	Example	`http://testvidp.buergerkarte.at/RetreveAttribute`
	Description	General plugin, used for other cases.

Table 23: Attribute providers settings

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 und 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.
```

Figure 9: Example AuthBlockText configuration

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:

Parameter		CCE-Selection Template
	Example	<uploaded file>

[2] https://www.eid-stork2.eu/wiki/index.php/AT

	Description	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).
Parameter		`Send-Assertion Template`
	Example	`<uploaded file>`
	Description	This option allows setting of application specific template for additional authentication request in the case of single sign-on login.

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.

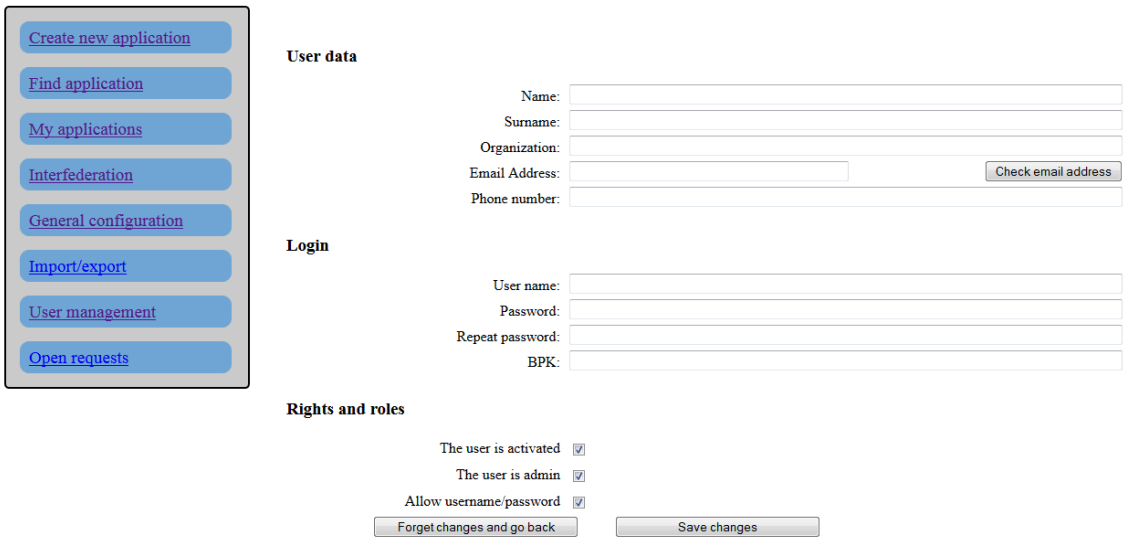

Figure 12: Creating new users

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 communication with 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 basic configuration elements necessary to be reviewed before the BKUonline is used in productive system.

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

```
<Whitelist>
  https://www\.formularservice\.gv\.at/BKU/.*,
  https?://127\.0\.0\.1/.*,
  https?://localhost/.*,
  https://testvidp.buergerkarte.at/.*
</Whitelist>
```

Figure 13: BKUOnline whitelist hosts

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 **/etc/tomcat7/webapps/bkuonline/WEB-INF/classes/at/gv/egiz/bku/certs**.

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

In directory **certStore/toBeAdded** copied are all the certificates used in the process of certificate chain building and verification, including the SSL certificate of BKUOnline and MOA-ID host system. Therefore, if the server SSL certificate does not contain complete chain, the certificates from that chain should be copied to this directory separately. After the restart, the application will check this directory, read and move its contents.

Should some certificates be left in **toBeAdded** after the application's or server restart, the server administrator should check the logs for the errors. One of the reasons might be incorrect certificate format. In this case, the file should be converted and again copied to **toBeAdded** directory.

3.8.3 Logging configuration

The file **/etc/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 higher logging level (e.g. **DEBUG** instead of **INFO**) and by adjusting file paths to fit the local path and configuration needs.

4 Deployment of V-IDP package

In this section the system requirements and contents of deployment archive are described. They are followed with the list of necessary steps in the deployment process. When necessary, the list contains also additional brief explanations.

For the reference to specific configuration please follow the descriptions from Section 3: Configuration of V-IDP and its components.

4.1 System requirements

Please ensure that you have **`Tomcat 6`** or **`7`** installed on the system, as well as updated **`Java SE 1.6`** or newer. The **`Apache Tomcat 7.0.50`** and **`Java SE Update SE 7`** (current update) are recommended. The location of Java SE will be further referred as **`$JAVA_HOME`**.

This manual assumes that all included `.war` files are to be deployed on the same `Tomcat` instance. Should specific deployment scenario require separate `Tomcat` instances, the steps described in this manual should be reproduced on each instance, where applicable.

4.2 Package contents

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

Path	Description
`conf`	Configuration directories of MOA-ID and its components
`doc`	Documentation, including this manual
`endorsed`	Additional libraries
`ext`	Additional external libraries
`source`	Source code of Demo Online Application
`tomcat`	Apache Tomcat startup scripts

Table 25: Contents of deployment archive

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

Path	Description
`moa-id`	Configuration of MOA-ID
`moa-id-configuration`	Configuration of MOA-ID web interface
`moa-spss`	Configuration of MOA-SPSS module

Table 26: Contents of configuration directory in deployment archive

4.3 Create MySQL databases

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

In order to deploy MOA-ID it is necessary to configure 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 carry further actions in the terms of database schema initialization. Such data will be automatically populated during MOA-ID deployment and initialization.

4.4 Copy and set up MOA-ID configuration files

The contents of **conf** directory, described in Table 25: Contents of deployment archive and Table 26: Contents of configuration directory in deployment archive should be copied to appropriate location, e.g. Tomcat configuration directory. Furthermore the contained files should be adjusted to reflect 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 under Debian is described 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 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 website[3] of other PEPS/VIDP interacting parties and updated certificate information can be consulted for further details.

Additionally, **KeyStore Explorer**[4] 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 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 *Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files* from the Java web site[5].

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 three war files contained in the archive should be deployed:

- **bkuonline.war**
- **moa-id-auth.war**
- **moa-id-configuration.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 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 have SSL check enabled.

4.12 Install production SSL certificate

The production SSL certificate should be installed and enabled in `Apache httpd` and `Apache Tomcat` daemons.

4.13 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.

4.14 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.

4.15 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 login option in configuration file **`moa-id-configtool.properties`** under **`moa-id-configuration`** directory. For such purpose the option **`general.login.deaktivate`** should be set to **`true`**. After the application is restarted the 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 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 get through general MOA-ID setup, create web interface users and create service provider application.

4.16 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.

4.17 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:

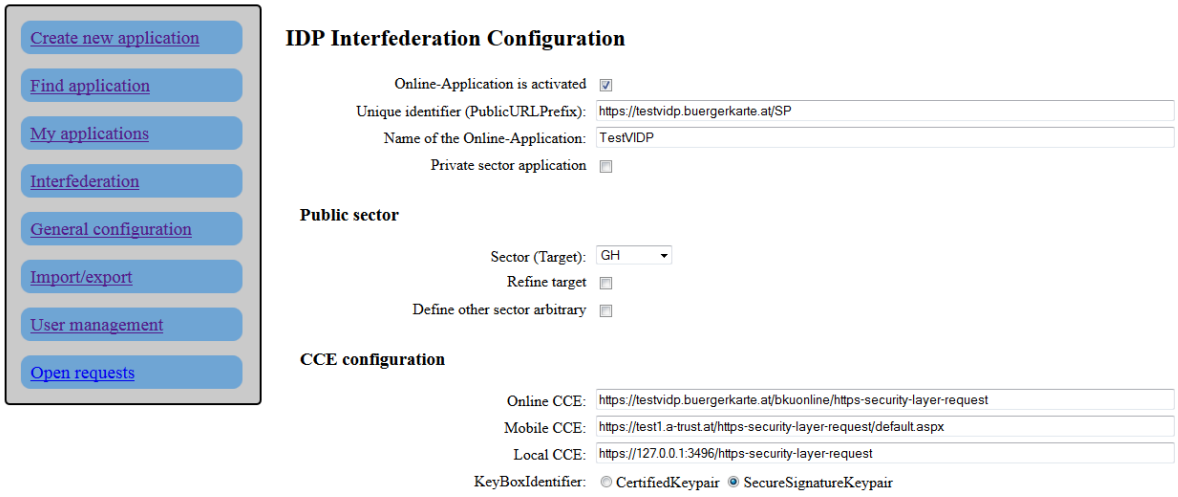

Figure 15: Application settings

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 References

[1] http://en.wikipedia.org/wiki/Executive_summary

[2] References are marked using a numbered list style referred to as References. This style has the following specifications: Calibri, 11pt., single line spacing, 6 pt. space from the previous paragraph.

[3] References within the text are in the form:

[4] [Az07] for a source with one author: A (capital) stands for the first letter of the author's family name, z for the second letter of the family name, 07 stands for the year 2007 of publication,

[5] [Az07a], [Az07b], if a number of works by the same author exist from the same year,

[6] [XY00] or [XYZ01] for a source with two or three authors: X, Y and Z stand for the first letters (in capitals) of the authors in the sequence in which they are listed in the source.

[7] in the case of more than three authors, only the first author mentioned in the source is listed, by analogy with (1), e.g. [Az07].