Installation and configuration of uApprove Jet Pack 2.5.1

This document contains the uApprove Jet Pack (in short, uApprove JP) deployment guide and the general manual.

uApprove JP is an extension for the Shibboleth Identity Provider 2.x. It also includes unique modification to the original uApprove. It allows to make users authenticating at an Identity Provider to accept terms of use and to release attribute selectively. More information about the concept of uApprove JP.

Notes about this guide:

- This guide assumes that uApprove JP is installed on a Linux system. It's also possible to install it on a different operating system, like Windows. In this case, you may need to adapt some paths and commands accordingly.
- The guide shows paths and commands using variables like, e.g. `$IDP_INSTALL$`, `$IDP_HOME$` or `$UAPPROVE_INSTALL$`. You need to substitute these variables by the real paths, except where it is explicitly stated that you don't need to substitute them.

Table of Contents

Assumptions

- The Shibboleth Identity Provider is unpacked at `$IDP_INSTALL$` (e.g., `/usr/local/src/shibboleth-identity-provider-#version#`).
- The Shibboleth Identity Provider is installed at `$IDP_HOME$` (e.g., `/opt/shibboleth-idp`).
- Tomcat is installed at `$CATALINA_HOME$` (e.g., `/usr/java/tomcat`).
- uApprove JP is downloaded and unpacked at `$UAPPROVE_INSTALL$` (e.g., `/usr/local/src/uApproveJP-#version#`).

1 Installation

1.1 Prerequisites

- Shibboleth Identity Provider 2.4.0 or later.
- MySQL 5.1 or later.

1.2 Library Installation

Copying the libraries to the IdPs library directory:

```
# cp $UAPPROVE_INSTALL$/lib/*.jar $IDP_INSTALL$/lib
# cp $UAPPROVE_INSTALL$/lib/jdbc/*.jar $IDP_INSTALL$/lib
```

Provide the JDBC connector for your database to the classpath of the IdP. You might use one of the provided MySQL or HSQL JDBC connector:

```
# cp $UAPPROVE_INSTALL$/lib/jdbc/optional/#jdbc-connector#.jar $IDP_INSTALL$/lib
```

⚠ Assure that only one version of each library is present in `$IDP_INSTALL$/lib`.

1.3 Configuration Template

Copying the configuration template to the IdPs configuration directory:

```
# cp $UAPPROVE_INSTALL$/manual/configuration/uApprove.properties $IDP_HOME$/conf
# cp $UAPPROVE_INSTALL$/manual/configuration/uApprove.xml $IDP_HOME$/conf
```

1.4 Webapp files

Copying of the web application files like the JSPs, CSS files and images to the IdPs webapp directory:

```
# mkdir $IDP_INSTALL$/src/main/webapp/uApprove
# cp $UAPPROVE_INSTALL$/webapp/* $IDP_INSTALL$/src/main/webapp/uApprove
```

1.5 Database Preparation

⚠ The following database parameters are examples. Adapt the values as required. Especially, choose a secure password.

- Create a database with the name “uApprove”.
- Create a database user with the username “uApprove” and password “secret”.
- Grant `INSERT, SELECT, UPDATE, DELETE` rights for the user.
- Create the initial table structures using the schemas:
    - `$UAPPROVE_INSTALL$/manual/storage/terms-of-use-schema.sql`
    - `$UAPPROVE_INSTALL$/manual/storage/attribute-release-schema.sql`
    - `$UAPPROVE_INSTALL$/manual/storage/service-access-data-schema.sql`

2 Basic Deployment

2.1 Web Application Deployment Descriptor

Extend the IdP web application deployment descriptor (`$IDP_INSTALL$/src/main/webapp/WEB-INF/web.xml`). Adapt your existing file as shown below.

- Add `$IDP_HOME$/conf/uApprove.xml` to the `contextConfigLocation` context parameter. Keep the `$IDP_HOME$` variables. They will be replaced during the re-deployment of the IdP in a later step.
- Add the required filters and servlets as shown.

```
<web-app ...>

    <context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>$IDP_HOME$/conf/internal.xml; $IDP_HOME$/conf/service.xml; $IDP_HOME$/conf/uApprove.xml;</param-value>
    </context-param>

    <!-- IdP Listeners, Filters and Servlets -->
    <!-- ...                                 -->

    <!-- uApprove Filter and Servlets -->

    <filter>
        <filter-name>uApprove</filter-name>
        <filter-class>ch.SWITCH.aai.uApprove.Intercepter</filter-class>
    </filter>

    <filter-mapping>
        <filter-name>uApprove</filter-name>
        <url-pattern>/profile/Shibboleth/SSO</url-pattern>
        <url-pattern>/profile/SAML1/SOAP/AttributeQuery</url-pattern>
        <url-pattern>/profile/SAML1/SOAP/ArtifactResolution</url-pattern>
        <url-pattern>/profile/SAML2/POST/SSO</url-pattern>
        <url-pattern>/profile/SAML2/POST-SimpleSign/SSO</url-pattern>
        <url-pattern>/profile/SAML2/Redirect/SSO</url-pattern>
        <url-pattern>/profile/SAML2/Unsolicited/SSO</url-pattern>
        <url-pattern>/Authn/UserPassword</url-pattern>
    </filter-mapping>

    <servlet>
        <servlet-name>uApprove - Terms Of Use</servlet-name>
        <servlet-class>ch.SWITCH.aai.uApprove.tou.ToUServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>uApprove - Terms Of Use</servlet-name>
        <url-pattern>/uApprove/TermsOfUse</url-pattern>
    </servlet-mapping>

    <servlet>
        <servlet-name>uApprove - Attribute Release</servlet-name>
        <servlet-class>ch.SWITCH.aai.uApprove.ar.AttributeReleaseServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>uApprove - Attribute Release</servlet-name>
        <url-pattern>/uApprove/AttributeRelease</url-pattern>
    </servlet-mapping>

</web-app>
```

2.2 Custom of Configuration

Modify IdP at IDP_HOME/conf/internal.xml. Add id="uajpMetadataExtensions" bean definition in <constructor-arg> element in id="shibboleth.OpensamlConfig" bean definition as below:

```
<bean id="shibboleth.OpensamlConfig" class="edu.internet2.middleware.shibboleth.common.config.OpensamlConfigBean"
         depends-on="shibboleth.LogbackLogging">
    ...
    <constructor-arg>
        <list>
            <bean id="shibMetadataExtensions" class="org.opensaml.util.resource.ClasspathResource">
                <constructor-arg value="/shibboleth-saml-ext-config.xml"/>
            </bean>
            <bean id="uajpMetadataExtensions" class="org.opensaml.util.resource.ClasspathResource">
                <constructor-arg value="/uApprove-jp-metadata-config.xml"/>
            </bean>
        </list>
    </constructor-arg>
    <property name="parserPool" ref="shibboleth.ParserPool"/>
    ...
```

In IDP_HOME/conf/uApprove.xml change:

```
<context:property-placeholder location="classpath:/configuration/uApprove.properties" />
```

to:

```
<context:property-placeholder location="file:$IDP_HOME$/conf/uApprove.properties" />
```

Replace the variable IDP_HOME by the correct path, e.g. /opt/shibboleth-idp (so that the whole value looks like file:/opt/shibboleth-idp/conf/uApprove.properties for example).

Put the correct metadata of the IdP as IDP_HOME/conf/relying-party.xml. And, edit the IDP_HOME/conf/relying-party.xml to read IDP_HOME/metadata/idp-metadata.xml:

```
<!-- ========================================== -->
<!--      Metadata Configuration                -->
<!-- ========================================== -->
<!-- MetadataProvider the combining other MetadataProviders -->
<metadata:MetadataProvider id="ShibbolethMetadata" xsi:type="metadata:ChainingMetadataProvider">

    <!-- Load the IdP's own metadata.  This is necessary for artifact support. -->
    <metadata:MetadataProvider id="IdPMD" xsi:type="metadata:FilesystemMetadataProvider"
                               metadataFile="$IDP_HOME$/metadata/idp-metadata.xml"
                               maxRefreshDelay="P1D" />
```

Customize IDP_HOME/conf/uApprove.properties according your database environment and required features. See inline documentation of uApprove.properties for configuration options.

In case you enable the 'Terms of Use' module (enabled by default), you need to provide an appropriate text suitable for your organization.

An example 'Terms Of Use' HTML file can be found in $UAPPROVE_INSTALL$/manual/examples/terms-of-use.html.

- Copy $UAPPROVE_INSTALL$/manual/examples/terms-of-use.html to IDP_HOME/conf/terms-of-use.html:

  ```
  # cp $UAPPROVE_INSTALL$/manual/examples/terms-of-use.html $IDP_HOME$/conf/terms-of-use.html
  ```

- Adapt IDP_HOME/conf/terms-of-use.html as required.
- Adapt the value of tou.resource in IDP_HOME/conf/uApprove.properties accordingly:

  ```
  tou.resource = file:$IDP_HOME$/conf/terms-of-use.html
  ```

2.3 Custom Templates

In case you want to customize the templates, follow section Custom View Templates.

At least, you should copy your organization's logo to the file $IDP_INSTALL$/src/main/webapp/uApprove/logo.png, since a placeholder logo is installed by default.
You may also want to put your federation's logo to the file $IDP_INSTALL$/src/main/webapp/uApprove/federation-logo.png (which is an empty placeholder logo by default).

- For the SWITCHaai federation, the logo is available at http://www.switch.ch/aai/design/images/switchaai-logo.png.
- For the GakuNin, the logo is available at https://www.gakunin.jp/info/logo/.

The logo is very large so that you may adjust the size using height, width in <img> tag at attribute-release.jsp and attribute-check.jsp that are copied into $IDP_INSTALL$/src/main/webapp/uApprove/.

```
<img src="<%= request.getContextPath()%>/uApprove/federation-logo.png" alt="" class="federation_logo" height="32" width="162"
>
```

2.4 Deployment

To activate uApprove JP the IdP must be re-deployed:

```
# cd $IDP_INSTALL$
# ./install.sh
```

Copy idp.war to $CATALINA_HOME$/webapps:

```
# cp $IDP_HOME$/war/idp.war $CATALINA_HOME$/webapps/
```

Restart Tomcat:

```
# service tomcat6 restart
```

3 Upgrade

3.1 Upgrade from uApprove.jp 2.2.1

You have to completely remove this old version first and perform a clean install.

Please have a look at 1 Installation above for further information on how to install uApprove.

- Legacy uApprove.jp-2.2.1 deployment: Please remove it.
- Configuration files: Please use the new configuration file and tailor them to your needs.

  The new ToU is an HTML file, which you can customize, just copy the content from your legacy ToU XML to a plain HTML file.

- uajpmf:AttributeUapprove matching rules: You cannot use anymore. Please rewrite rules using uajpmf:AttributeInMetadata instead. Please refer to 4.7 Attribute In Attribute Requester's Metadata Plugin for further information.

1. According to isRequired attribute of <RequestedAttribute> element and you want to set desired if it is set to true, optional if it is set to false, and not displayed if the attribute has no <RequestedAttribute> element and the metadata has no <AttributeConsumingService> element, the rule is rewritable as below:

uApprove.jp 2.2.1	uApprove JP 2.5.1
<PermitValueRule xsi:type="uajpmf:AttributeUapprove" *isApproved="true"* requestedOnly="true" />	<PermitValueRule xsi:type="uajpmf:AttributeInMetadata" onlyIfChecked="true" onlyIfRequired="false" *matchIfMetadataSilent="false"* />

2. According to isRequired attribute of <RequestedAttribute> element and you want to set desired if it is set to true, optional if it is set to false, and also optional if the metadata has no <AttributeConsumingServie> element, the rule is rewritable as below. But, in uajpmf:AttributeUapprove the attribute which has no <RequestedAttribute> element in <AttributeConsumingService> element it becomes optional, in uajpmf:AttributeInMetadata, it is not displayed:

uApprove.jp 2.2.1	uApprove JP 2.5.1
<PermitValueRule xsi:type="uajpmf:AttributeUapprove" *isApproved="true" requestedOnly="false"* />	<PermitValueRule xsi:type="uajpmf:AttributeInMetadata" onlyIfChecked="true" onlyIfRequired="false" matchIfMetadataSilent="true" />

3. The policy which has <AttributeConsumingService> element in metadata, it is not rewritable straight forward, <PolicyRequirementRule> element should be set to <basic:ANY> element and set onlyIfRequired="true" matchIfMetadataSilent="false" at each rule:

uApprove.jp 2.2.1	uApprove JP 2.5.1
<PolicyRequirementRule xsi:type="uajpmf:AttributeUapprove" />	<PolicyRequirementRule xsi:type="basic:ANY" />
<PermitValueRule xsi:type="uajpmf:AttributeUapprove" *isApproved="true" requestedOnly="false"* />	<PermitValueRule xsi:type="uajpmf:AttributeInMetadata" onlyIfChecked="true" onlyIfRequired="false" *matchIfMetadataSilent="false"* />

In the table above, attributes written in italic is default value so that may be omitted in the actual configuration.

- JSP: Please use the new provided JSP files. If you use 4.1 Reset Attribute Release Consent, please adjust your login.jsp.
- Database Migration: It is impossible to migrate.

4 Advanced Deployment

This section contains advanced configuration topics.

4.1 Reset Attribute Release Consent

For providing the feature that a user is able to clear [1] her attribute release consent during the login flow, add a checkbox to $IDP_INSTALL$/src/main/webapp/login.jsp:

```
<form action="<%=request.getAttribute("actionUrl")%>" method="post">
  ...
  <input id="uApprove.consent-revocation" type="checkbox" name="uApprove.consent-revocation" value="true"/>
  <label for="uApprove.consent-revocation">Clear my attribute release consent</label>
  ...
</form>
```

In default login page of Shibboleth IdP 2.4.0 or later, label is made invisible using CSS. So you need to change `<label>` using `style` attribute.

```
<form action="<%=request.getAttribute("actionUrl")%>" method="post">
  ...
  <section>
    <input id="uApprove.consent-revocation" type="checkbox" name="uApprove.consent-revocation" value="true"/>
    <label for="uApprove.consent-revocation" style="position: relative; left: 0px;">Clear my attribute release consent<
/label>
  </section>
  ...
</form>
```

[1] Clear means to delete general consent if it was given as well delete as all attribute release consents for the accessed relying party.

4.2 Storage

File-only

For a simple deployment a file only database can be used. HSQL provides such an option.
Define the according database properties in `uApprove.properties`:

```
database.driver             = org.hsqldb.jdbcDriver
database.url                = jdbc:hsqldb:file:/var/opt/uApprove/hsql.db
database.username           = SA
database.password           =
```

Initializing the database with the provided schemas:

```
echo "SHUTDOWN;" > /tmp/shutdown
java -jar $HSQLDB_HOME$/lib/sqltool.jar ¥
  --inlineRC=url=jdbc:hsqldb:file:/var/opt/uApprove/hsql.db,user=SA,password= ¥
  $UAPPROVE_INSTALL$/manual/storage/terms-of-use-schema.sql /tmp/shutdown
java -jar $HSQLDB_HOME$/lib/sqltool.jar ¥
  --inlineRC=url=jdbc:hsqldb:file:/var/opt/uApprove/hsql.db,user=SA,password= ¥
  $UAPPROVE_INSTALL$/manual/storage/attribute-relase-schema.sql /tmp/shutdown
java -jar $HSQLDB_HOME$/lib/sqltool.jar ¥
  --inlineRC=url=jdbc:hsqldb:file:/var/opt/uApprove/hsql.db,user=SA,password= ¥
  $UAPPROVE_INSTALL$/manual/storage/service-access-data-schema.sql /tmp/shutdown
```

⚠ `$HSQLDB_HOME$` defines the location where the downloaded HSQL distribution is extracted.

⚠ Assure that the user running the container (e.g., Jetty) has write permission to the db directory.

Custom SQL Statements

1. Copy the provided `$UAPPROVE_INSTALL$/manual/storage/sql-statements.properties` to `$IDP_HOME$/conf/uApprove.sql-statements.properties`.
2. Adjust `$IDP_HOME$/conf/uApprove.sql-statements.properties` according your needs.
3. Enable your custom `sql-statements.properties` in `$IDP_HOME$/conf/uApprove.xml`:

```
<bean id="uApprove.touModule" class="ch.SWITCH.aai.uApprove.tou.ToUModule" ...>
    <!-- ... -->
    <property name="storage">
        <bean class="ch.SWITCH.aai.uApprove.tou.storage.JDBCStorage" ...
            p:sqlStatements="file:/$IDP_HOME$/conf/uApprove.sql-statements.properties" ... />
    </property>
</bean>

<!-- ... -->

<bean id="uApprove.attributeReleaseModule" class="ch.SWITCH.aai.uApprove.ar.AttributeReleaseModule" ...>
    <!-- ... -->
    <property name="storage">
        <bean class="ch.SWITCH.aai.uApprove.ar.storage.JDBCStorage" ...
           p:sqlStatements="file:/$IDP_HOME$/conf/uApprove.sql-statements.properties" ... />
    </property>
</bean>
```

Graceful JDBC Connection Handling

The JDBC storage can be configured to act graceful in case of a temporary database issue (e.g., communication link is not available). Instead throwing exceptions and display the error page, no persistent actions are applied.
This implies that the users, independently of former ToU acceptances and/or attribute release consents have to accept/consent again (if they already had) or have to do it during the next login (if it was the first time).

It might make sense to set checkoutTimeout to an appropriate low value – so as the user has not bothersome latency.

The settings are defined in `$IDP_HOME$/conf/uApprove.xml`:

```
<bean id="uApprove.touModule" class="ch.SWITCH.aai.uApprove.tou.ToUModule" ...>
    <!-- ... -->
    <property name="storage">
        <bean class="ch.SWITCH.aai.uApprove.tou.storage.JDBCStorage" ...
            p:graceful="true" ... />
    </property>
</bean>

<!-- ... -->

<bean id="uApprove.attributeReleaseModule" class="ch.SWITCH.aai.uApprove.ar.AttributeReleaseModule" ...>
    <!-- ... -->
    <property name="storage">
        <bean class="ch.SWITCH.aai.uApprove.ar.storage.JDBCStorage" ...
           p:graceful="true" ... />
    </property>
</bean>
```

JDBC Connection Pool Tuning

c3p0 configuration for further details on configuration options.

The settings are defined in `$IDP_HOME$/conf/uApprove.xml`:

```
<bean id="uApprove.dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource" ...
    ...
    p:idleConnectionTestPeriod="300" ... />
```

You are free to use another JDBC connection pooling library (e.g., BoneCP). Just provide the right data source class name in the bean definition as well the required libraries in the classpath.

4.3 Templates

Custom View Templates

Feel free to customize the JSP, CSS and image files located in $IDP_INSTALL$/src/main/webapp/uApprove/.
For convenience the JSTL is used, cf. JSTL reference.

4.4 Localization

Custom Messages

You might adjust/extend the provided resource bundles in $UAPPROVE_INSTALL$/manual/examples/messages and copy them into the IdPs classpath (e.g., $IDP_INSTALL$/src/main/webapp/WEB-INF/classes/uApprove/messages).
Specify your bundles base in IDP_HOME/conf/uApprove.xml:

```
    <bean id="uApprove.viewHelper" class="ch.SWITCH.aai.uApprove.ViewHelper" ...
        p:messagesBase="uApprove.messages" />
```

If you customize Japanese messages, edit file #view#_ja-UTF8.properties which character code is UTF-8. After editing the file, convert the file using native2ascii command and copy it to $IDP_INSTALL$/src/main/webapp/WEB-INF/classes/uApprove/messages.
Customizing message attribute-release.jsp procedure is:

```
# cd $UAPPROVE_INSTALL$/manual/examples/messages
(Edit attribute-release_ja-UTF8.properties)
# native2ascii attribute-release_ja-UTF8.properties attribute-release_ja.properties
# cp attribute-release_ja.properties $IDP_INSTALL$/src/main/webapp/WEB-INF/classes/uApprove/messages/
```

Attribute Names and Descriptions

See $UAPPROVE_INSTALL$/manual/examples/attribute-descriptions.xml for an example how to configure the localized attribute names and descriptions.

Relying Party Names and Descriptions

Currently only <AttributeConsumingService> element in metadata is supported to retrieve localized relying party names and descriptions.
For providing such names and descriptions extend the metadata for the SP like:

```
<EntityDescriptor entityID="https://sp.example.org/shibboleth">
    <!-- ... -->
    <SPSSODescriptor>
        <!-- ... -->
        <AttributeConsumingService index="1">
            <ServiceName xml:lang="en">Example SP</ServiceName>
            <!-- Service names in other languages -->
            <ServiceDescription xml:lang="en">Some description of Example SP</ServiceDescription>
            <!-- Service descriptions in other languages -->
        </AttributeConsumingService>
    </SPSSODescriptor>
</EntityDescriptor>
```

The metadata of the SWITCHaai federation contain these information.

4.5 Strict Comparison

Terms Of Use Content Comparison

In the default configuration, only the ToU version is compared to evaluate if a user accepted the ToU. You might enable that the ToU content is compared too in IDP_HOME/conf/uApprove.xml:

```
<bean id="uApprove.touModule" ... p:compareContent="true" ... />
```

4.6 Audit Logging

uApprove JP allows (facilitating the IdP's audit log) to enable audit logging to IDP_HOME/logs/idp-audit.log.

Enable Terms Of Use Audit Log

You might enable ToU audit log in IDP_HOME/conf/uApprove.xml:

```
<bean id="uApprove.touModule" ... p:auditLogEnabled="true" ... />
```

Example:

```
20120101T010000Z|ch.SWITCH.aai.uApprove|||tou.acceptance|null|null|null|student1||1.0,5
b2ee897c08c79a09cd57e8602d605bf8c52db17de9793677c36b5c78644b2b3,|
```

Enable Attribute Release Audit Log

You might enable attribute release audit log in IDP_HOME/conf/uApprove.xml:

```
<bean id="uApprove.attributeReleaseModule" ... p:auditLogEnabled="true" ... />
```

Examples:

```
20120101T010000Z|ch.SWITCH.aai.uApprove||https://sp.example.org/shibboleth|ar.consent|null|null|null|student1||uid,surname,givenName,|
20120101T010000Z|ch.SWITCH.aai.uApprove||https://sp.example.org/shibboleth|ar.clearConsent|null|null|null|student1|||
20120101T010000Z|ch.SWITCH.aai.uApprove|||ar.generalConsent|null|null|null|student1|||
```

4.7 Attribute In Attribute Requester's Metadata Plugin

Configuration Attribute In Requester's Metadata Matching Rule

This rule allows the release of an attribute if, via its metadata, the SP indicates a need/desire for the attribute. The attributes are indicated by means of <AttributeConsumingService> element within the<SPSSODescriptor> element. Attributes with isRequired='true' at <RequestedAttribute> is marked as required, and with isRequired='false' is marked as optional. See SAML metadata for more information.

Please be aware of the following:

- This filter requires the attribute requester's metadata be loaded and available.
- The requester's metadata must have an <SPSSODescriptor> role since that is the role that contains the listed attributes.
- This matching function only operates as a value rule and only really makes sense as a permit value rule.

Define the Namespace

In your attribute filter policy file you'll need to add the namespace declaration for this plugin. To do this:

- Add the attribute xmlns:uajpmf="http://www.gakunin.jp/ns/uapprove-jp/afp/mf" before the xmlns:xsi attribute on the root <AttributeFilterPolicyGroup> element.
- Add the following at the end of the whitespace delimited list of values for the xsi:schemaLocation attribute:
  http://www.gakunin.jp/ns/uapprove-jp/afp/mf classpath:/schema/shibboleth-2.0-afp-mf-uApprovejp.xsd

Define the Rule

This rule is defined by the <PermitValueRule xsi:type="uajpmf:AttributeInMetadata"> element with the following optional attribute:

onlyIfRequired	Boolean flag indicated that only those attributes which are marked as required should be released, those marked as optional will not be. Default value: **true**.
matchIfMetadataSilent	Boolean flag indicated that is marked as optional when the metadata has no <AttributeConsumingService> element. Default value: **false**.
onlyIfChecked	Boolean flag indicated that only those attributes which are marked as optional and user has permitted should be released. Default value: **false**. When set to false, its behavior is as same as saml:AttributeInMetadata of Shibboleth IdP 2.4.0 or later.

How to write Permit Value Rule using the AttributeInMetadata Match Function:

```
<PermitValueRule xsi:type="uajpmf:AttributeInMetadata" onlyIfRequired="false" onlyIfChecked="true">
```

Attributes marked as optional will be displayed with checkbox. It is released when checkbox is checked only.

Example Permit Value Rule using the AttributeInMetadata Match Function:

```
<!--    ================================================================================
case 1: rule which compares metadata definitions with attributes mail,
     eduPersonPrincipalName, eduPersonAffiliation.

     Metadata which is marked as required, Everything is required information
     and always released.

     Metadate which is marked as optional:
     * mail attribute is required information and always released.
     * eduPersonPrincipalName attribute is optional information. In attribute
       selection window, it is displayed with checkbox. If the user checked the
       checkbox, it is released.
     * eduPersonAffiliation attribute is not released.

     No attributes are released when SP has no <AttributeConsumingService>
     element in metadata.
        ================================================================================ -->
<afp:AttributeFilterPolicy id="PolicyforSPwithAttributeConsumingService">
    <afp:PolicyRequirementRule xsi:type="basic:ANY" />

    <afp:AttributeRule attributeID="mail">
        <afp:PermitValueRule xsi:type="uajpmf:AttributeInMetadata"
                             onlyIfRequired="false" />
    </afp:AttributeRule>

    <afp:AttributeRule attributeID="eduPersonPrincipalName">
        <afp:PermitValueRule xsi:type="uajpmf:AttributeInMetadata"
                             onlyIfRequired="false"
                             onlyIfChecked="true" />
    </afp:AttributeRule>

    <afp:AttributeRule attributeID="eduPersonAffiliation">
        <afp:PermitValueRule xsi:type="uajpmf:AttributeInMetadata" />
    </afp:AttributeRule>

</afp:AttributeFilterPolicy>

<!--    ================================================================================
case 2: Example rule to add rule to SP which has no <AttributeConsumingService>
     element in metadata.

     When SP has no <AttributeConsumingService> element:
     * mail attribute is required information and always released.
     * eduPersonPrincipalName attribute is optional information. In attribute
       selection window, it is displayed with checkbox. If the user checked the
       checkbox, it is released.
     * eduPersonAffiliation attribute is not released.

     When SP has <AttributeConsumingService> element, it is the same as case 1.
```

```
        ============================================================================ -->
<afp:AttributeFilterPolicy id="PolicyforSPwithoutAttributeConsumingService">
    <afp:PolicyRequirementRule xsi:type="basic:ANY" />

    <afp:AttributeRule attributeID="mail">
        <afp:PermitValueRule xsi:type="uajpmf:AttributeInMetadata"
                             matchIfMetadataSilent="true"
                             onlyIfRequired="false" />
    </afp:AttributeRule>

    <afp:AttributeRule attributeID="eduPersonPrincipalName">
        <afp:PermitValueRule xsi:type="uajpmf:AttributeInMetadata"
                             matchIfMetadataSilent="true"
                             onlyIfRequired="false"
                             onlyIfChecked="true" />
    </afp:AttributeRule>

    <afp:AttributeRule attributeID="eduPersonAffiliation">
        <afp:PermitValueRule xsi:type="uajpmf:AttributeInMetadata" />
    </afp:AttributeRule>

</afp:AttributeFilterPolicy>
```

4.8 Modification of AttributeQuery profile handler

uApprove JP generates a list of attributes of SAML response message in Attribute Query according to the content of user's acceptance.

Configuration of profile handler

Define the name space

You need to add the definition of name space for this plugin to profile handler file(ex. IDP_HOME/conf/handler.xml) like,

1. add xmlns:uajpph="http://www.gakunin.jp/ns/uapprove-jp/profile-handler" attribute before xmlns:xsi attribute in the root element.
2. add below to the list of xsi:schemaLocation attribute values.
   http://www.gakunin.jp/ns/uapprove-jp/profile-handler classpath:/schema/shibboleth-2.0-idp-profile-handler-uapprovejp.xsd

```
...
<ph:ProfileHandlerGroup
        xmlns:ph="urn:mace:shibboleth:2.0:idp:profile-handler"
        xmlns:uajpph="http://www.gakunin.jp/ns/uapprove-jp/profile-handler"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:mace:shibboleth:2.0:idp:profile-handler classpath:/schema/shibboleth-2.0-idp-profile-handler.xsd
                            http://www.gakunin.jp/ns/uapprove-jp/profile-handler classpath:/schema/shibboleth-2.0-idp-profile-handler-
uapprovejp.xsd">
...
```

Change of Profile Handler definition

You need to change AttributeQuery profile handler like:

1. change value of xsi:type attribute from ph:SAML1AttributeQuery to uajpph:SAML1AttributeQueryUApprove.
2. change value of xsi:type attribute from ph:SAML2AttributeQuery to uajpph:SAML2AttributeQueryUApprove.

```
...
    <ph:ProfileHandler xsi:type="uajpph:SAML1AttributeQueryUApprove" inboundBinding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-
binding"
                       outboundBindingEnumeration="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding">
        <ph:RequestPath>/SAML1/SOAP/AttributeQuery</ph:RequestPath>
    </ph:ProfileHandler>
...
    <ph:ProfileHandler xsi:type="uajpph:SAML2AttributeQueryUApprove" inboundBinding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
                       outboundBindingEnumeration="urn:oasis:names:tc:SAML:2.0:bindings:SOAP">
        <ph:RequestPath>/SAML2/SOAP/AttributeQuery</ph:RequestPath>
    </ph:ProfileHandler>
...
```

Behavior when the SP is blacklisted.

If `services.blacklist` property of `uApprove.properties` is set true, `services` property becomes to the black list of the SP. The SP that matches the entity ID with a regular expression in the `services` property connect without using uApprove. In this case, attributes that has been send are not saved to the storage.

If the AttribureQuery profile handler of uApprove JP is set, this handler will not respond the attribute information to the query from SP that are registered in the blacklist. When a user had been consented before the SP has been blacklisted, except the case consented by "This time I agree to send this data. I will check the data again at next login", this handler responds the consented attributes.

> ⚠ Do not blacklist a SAML1 SP when the IdP has been configured to use the AttribureQuery profile handler of uApprove JP. This IdP can not send the attributes to SAML1 SP that has blacklisted.

4.9 List of attribute approved SP

Installation and Configuration of Shibboleth SP

The URL of the list of attribute approved SP must be protected at the Shibboleth SP which provides REMOTE_USER.

Install Shibboleth SP to your IdP server and configure as below.

Configuration of REMOTE_USER

Add `uid` to `REMOTE_USER` at `/etc/shibboleth/shibboleth2.xml`:

```
...
    <!-- The ApplicationDefaults element is where most of Shibboleth's SAML bits are defined. -->
    <ApplicationDefaults entityID="https://idp.example.ac.jp/shibboleth-sp"
                         REMOTE_USER="uid">
...
```

Configuration of SSO

Delete the configuration for the DS from the <SSO> element at `/etc/shibboleth/shibboleth2.xml` because of accessing the IdP on the same machine:

```
...
           <SSO entityID="https://idp.example.ac.jp/idp/shibboleth">
             SAML2 SAML1
           </SSO>
...
```

Configuration IdP's metadata

Add configuration to read the IdP's metadata at `/etc/shibboleth/shibboleth2.xml`:

```
...
       <!-- Example of locally maintained metadata. -->
       <MetadataProvider type="XML" file="$IDP_HOME$/metadata/idp-metadata.xml" />
...
```

Configuration of receiving uid

Add configuration to receive uid from IdP at /etc/shibboleth/attribute-map.xml:

```
 ...
   <Attribute name="urn:mace:dir:attribute-def:uid" id="uid"/>
   <Attribute name="urn:oid:0.9.2342.19200300.100.1.1" id="uid"/>

</Attributes>
```

After configuration, restart shibd.

Configuration of IdP

Configuration of SP's metadata

Add configuration to read the SP's metadata at IDP_HOME/conf/relying-party.xml:

```
...
       <!-- Load the local SP's metadata. -->
       <metadata:MetadataProvider id="LocalSPMD" xsi:type="metadata:FilesystemMetadataProvider"
                                  metadataFile="$IDP_HOME$/metadata/sp-metadata.xml"
                                  maxRefreshDelay="P1D" />
...
```

You will get the template of the SP's metadata from the below URL:

https://idp.example.ac.jp/Shibboleth.sso/Metadata

Configuration of attribute-resolver.xml

Add definition of uid at IDP_HOME/conf/attribute-resolver.xml:

```
   <resolver:AttributeDefinition xsi:type="ad:Simple" id="uid" sourceAttributeID="uid">
       <resolver:Dependency ref="myLDAP" />
       <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-def:uid" />
       <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:0.9.2342.19200300.100.1.1" friendlyName="uid" />
   </resolver:AttributeDefinition>
```

Configuration of attribute-filter.xml

Configure only uid is released at IDP_HOME/conf/attribute-filter.xml:

```
   <afp:AttributeFilterPolicy id="...">
       <afp:PolicyRequirementRule xsi:type="basic:AttributeRequesterString"
                                  value="https://idp.example.ac.jp/shibboleth-sp" />
       <afp:AttributeRule attributeID="uid">
           <afp:PermitValueRule xsi:type="basic:ANY"/>
       </afp:AttributeRule>
   </afp:AttributeFilterPolicy
```

Configuration of uApprove.properties

Configure to skip consent by uApprove because the SP to work at the IdP host.

- services: Add the entity ID by regular expression.

- services.blacklist: set true.

```
services                    = ^https://idp¥.example¥.ac¥.jp/shibboleth-sp$
services.blacklist          = true
```

Add Configuration of ListApprovalsServlet to Web application deployment descriptor:

```
<web-app ...>

    ...
    <servlet>
        <servlet-name>ListConsentedSP - List of consented SP</servlet-name>
        <servlet-class>jp.gakunin.uApprovejp.lcsp.ListApprovalsServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>ListConsentedSP - List of consented SP</servlet-name>
        <url-pattern>/uApprove/ListConsentedSP</url-pattern>
    </servlet-mapping>

</web-app>
```

You need to re-deploy IdP. See 2.4 Deployment above.

Configuration of httpd

Configure httpd at /etc/httpd/conf.d/ssl.conf as below:

```
...
<Location /idp/uApprove/ListConsentedSP>
  AuthType shibboleth
  ShibRequestSetting requireSession 1
  require valid-user
</Location>
...
```

After configuration, restart httpd.

URL of the list of attribute approved SP

The URL of the list of attribute approved SP is as below:

https://idp.example.ac.jp/idp/uApprove/ListConsentedSP

Configure Exit page

When Exit button pressed, redirects to `https://idp.example.ac.jp/idp/uApprove/list-approvals-exit.html`. You can change the location by setting the URL to `lcsp.returnURL` of the `uApprove.properties`.

Example:

```
lcsp.returnURL = https://idp.exmaple.ac.jp/your-made-page.html
```

5 Troubleshooting

5.1 Troubleshooting

- Check IDP_HOME/logs/idp-process.log for ERROR or WARN messages.
- Check Tomcat’s log files located at $CATALINA_HOME$/logs for error messages.

5.2 Detailed logging

Enabling DEBUG or TRACE log level for uApprove JP in IDP_HOME/conf/logging.xml:

```
<logger name="ch.SWITCH.aai.uApprove" level="DEBUG"/>
<logger name="jp.gakunin.shibboleth" level="DEBUG"/>
<logger name="jp.gakunin.uApprovejp" level="DEBUG"/>
```

A Notification of the using purpose of attributes on SP

A SP administrator can notify users of the using purpose (ex, use as initial value of user's profile) of attributes on uApporve JP when they add the using purpose of attributes to their SP metadata.

A.1 Configuration

The notification of the using purpose of attributes on SP can be used to add uajpmd:description to <RequestedAttribute> element, or add <uajpmd:RequestedAttributeExtension> element in <Extensions> element in <SPSSODescriptor> element.

<uajpmd:RequestedAttributeExtension> element can describe by multiple languages. If both are set to one attribute, <uajpmd:RequestedAttributeExtension> element takes precedence.

uajpmd:description

This attribute is defined by the <ResquestedAttribute> element:

uajpmd:description	String indicated of the using purpose of attributes on SP

Example the <RequestedAttribute> element with uajpmd:descrption:

```
      <md:RequestedAttribute FriendlyName="mail"
               Name="urn:oid:0.9.2342.19200300.100.1.3"
               NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
               uajpmd:description="The mail attribute is used as the initial value of the mail address field of the registration
form."/>
```

<uajpmd:RequestedAttributeExtension>

This element defines with the following attributes and one and more <uajpmd:Description> elements:

uajpmd:FriendlyName	The value of FriendlyName of the <RequestedAttribute> element to associate <uajpmd:RequestedAttributeExtension> element.

The <uajpmd:Description> element describes the using purpose of attributes on SP and defines with the following attributes:

xml:lang	The language used in the using purpose of attributes on SP

Example the <uajpmd:RequestedAttributeExtension>:

```
<md:EntitiesDescriptor Name="uapprovejp-dev-metadata.xml"
                    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
                    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
                    xmlns:shibmd="urn:mace:shibboleth:metadata:1.0"
                    xmlns:uajpmd="http://www.gakunin.jp/ns/uapprove-jp/metadata"
                    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  ...
  <md:EntityDescriptor entityID="...">
    <md:SPSSODescriptor>
      ...

      <md:Extensions>
        ...
        <uajpmd:RequestedAttributeExtension FriendlyName="mail">
          <uajpmd:Description xml:lang="en">The mail attribute is used as the initial value of the mail address field of the
registration form.</uajpmd:Description>
          <uajpmd:Description xml:lang="ja">mail 属性を登録ページのメールアドレス欄の初期値として使用します</uajpmd:Description>
        </uajpmd:RequestedAttributeExtension>
        ...
      </md:Extensions>
      ...

      <md:AttributeConsumingService index="1">
              <md:ServiceName xml:lang="en">Sample Service</md:ServiceName>

        <md:RequestedAttribute FriendlyName="eduPersonPrincipalName"
                               Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
                               NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                               isRequired="true"/>
        <md:RequestedAttribute FriendlyName="mail"
                               Name="urn:oid:0.9.2342.19200300.100.1.3"
                               NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
      </md:AttributeConsumingService>
      ...
  </md:SPSSODescriptor>
```