SP 構築・運用手順書
(Ver1.2)

１．概要

本書は SP の構築手順、および運用方法を説明したものです。

［１］SP の機能

まず、SP の動作について簡単に説明します。

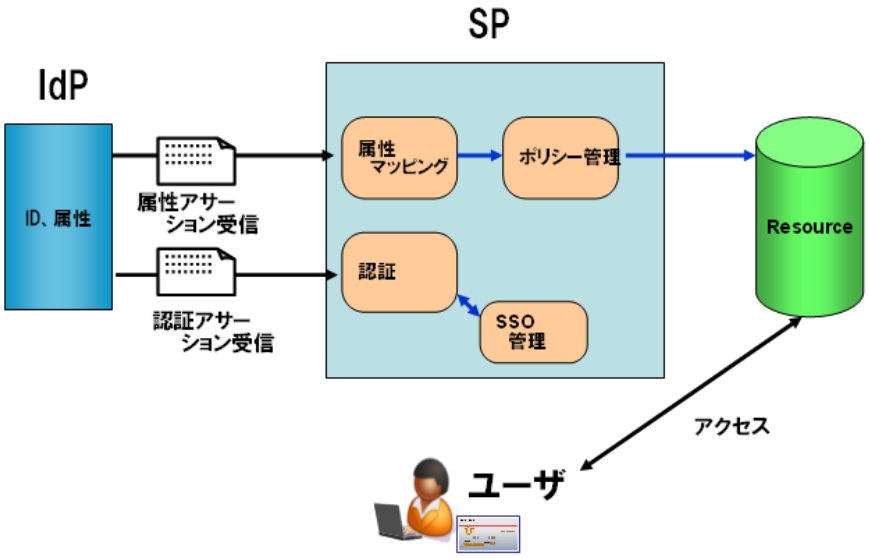


図１　SP の機能ブロック

図１　SP の機能ブロックは、SP の機能を単純化したブロックで示しています。SP は IdP と連携して、以下の２つの動作を行います。

- ユーザの認証を IdP に要求する
- ユーザの属性を安全に IdP から受信して、アプリケーションに渡す

■ 認証要求

ユーザが SP にアクセスすると、SP は IdP にリダイレクトを行い、IdP にユーザの認証を要求します。IdP はこれを受けてユーザの認証を行います。認証方式としては、ID／パスワード認証や、クライアント証明書による認証等の認証方式が設定可能です。

ユーザの認証が行われると、SP は IdP から認証アサーションを受信してユーザを認証したことを確認します。ただし、ここで受信するのはユーザを認証したという事実のみで、そのユーザが誰かという情報は渡されません。

■ 属性の安全な受信

SP は IdP に必要とする属性を要求します。IdP は要求された属性を属性アサーションに入れて SP に送信します。SP はこれを受信して、下記を行います。

- 属性アサーションから属性を取得して、属性の名称を IdP 間で利用する名称から、アプリケーションに渡すための名称に変換する。（図１の属性マッピング機能）
- アプリケーションへのアクセスを許可して良いかどうか、ポリシーを確認します。問題がない場合は、属性をアプリケーションに渡します。（図１のポリシー管理機能）
  アプリケーションでは属性を受け、この属性によりユーザに対する認可判断を行います。

以下の章では、SP の構築手順を示すとともに、上記機能の設定方法、および、これらの機能を用いて SP を運用するための方法について説明します。

［2］構築方式について

本書では、貴学にてサーバに OS から shibboleth(SP)までインストール・設定を行い、構築する方式について説明します。

また、SP として Plone を動作させる例も掲載していますので、合わせて参考にしてください。

２．貴学にて SP をインストールする場合の構築手順

［１］shibboleth (SP version2.0)の動作要件

・Apache HTTP Server 2.2 以上
・Apache Tomcat 5.5.25 以上
・Java 5 以上 (Plone を使用する場合には必要)
（ただし、CentOS に付属する Gnu Java は利用できません。 Sun の Java を利用してください。）

［２］OS をインストールする

①OS での設定

・OS：CentOS 5.1

インストーラでインストールするもの。

Web サーバー (HTTP のみ)
unixODBC

その他のパッケージがある場合は必要に応じてインストールしてください。
ただし、Java 開発と Tomcat は後の手順で別にインストールします。

・ネットワーク設定

環境に合わせ、ホスト名・ネットワーク・セキュリティを設定して下さい。
SP では shibd サービスが通信を行います。

②DNS へ登録する

新しいホスト名とｉｐアドレスを DNS に登録してください

③時刻同期を設定する

ntp サービスを用い、貴学環境の ntp サーバと時刻同期をしてください。
※Shibboleth では、通信するサーバ間の時刻のずれが約５分を越えるとエラーになります。

［3］jdk6、tomcat6 をインストールする

①tomcat5-5.5.17-8 の削除

tomcat5-5.5.25 以前のバージョンが入っている場合は、削除してください。

②jdk 6.0 のインストール

http://java.sun.com/javase/ja/6/download.html よりダウンロードした jdk-6u5-linux-i586-rpm.bin を適当なフォルダに置いて、以下のコマンドを実行してください。

```
chmod a+x jdk-6u5-linux-i586-rpm.bin
./jdk-6u5-linux-i586-rpm.bin
```

③tomcat6.0.16 のインストール

http://tomcat.apache.org/download-60.cgi よりダウンロードした apache-tomcat-6.0.16.tar.gz を /usr/java に置いて、以下のコマンドを実行してください。

```
cd /usr/java
  tar zxvf apache-tomcat-6.0.16.tar.gz
  ln -s apache-tomcat-6.0.16 /usr/java/tomcat
```

jsvc などを用いて、自動起動させると便利です。

ソースファイルを展開し、make で jsvc を作成した後、 $CATALINA_HOME/bin にコピーします。起動用スクリプトをコピーします。

```
  cd /usr/java/tomcat/bin
  tar xzvf jsvc.tar.gz
  cd jsvc-src
  ./configure
  make
  cp jsvc ..
  cp native/Tomcat5.sh /etc/rc.d/init.d/tomcat
```

/etc/rc.d/init.d/tomcat の先頭にコメントを追加することにより chkconfig コマンドが利用できます。

```
# chkconfig: - 86 15
# description: Tomcat
# processname: tomcat
```

また、ファイル中の以下の環境変数も変更が必要です。

JAVA_HOME

CATALINA_HOME

DAEMON_HOME

CATALINA_BASE

設定例

```
JAVA_HOME=/usr/java/default
CATALINA_HOME=/usr/local/tomcat
DAEMON_HOME=$CATALINA_HOME
CATALINA_BASE=$CATALINA_HOME
```

tomcat を実行するユーザ “tomcat” を作成した場合には

```
TOMCAT_USER=tomcat
```

も設定します。

他に、”tomcat”ユーザがログファイルを出力できるよう、ディレクトリの所有者を変更します。(シンボリックリンク先を変更するため最後の “/” が必要です)

```
chwon –R tomcat:   /usr/java/tomcat/
```

自動起動の設定 (オプションは マイナス ‘ー‘ が２つ必要です)

```
chkconfig --add tomcat
chkconfig --level 345 tomcat on
```

④/etc/profile に下記を追加してください。

```
JAVA_HOME=/usr/java/default
MANPATH=$MANPATH:$JAVA_HOME/man
CATALINA_HOME=/usr/java/tomcat
TOMCAT_HOME=$CATALINA_HOME
PATH=$JAVA_HOME/bin:$CATALINA_HOME/bin:$PATH
export PATH JAVA_HOME CATALINA_HOME
```

［3］shibboleth のインストール

① shibboleth-SP 関連のインストールファイルのダウンロード

http://shibboleth.internet2.edu/downloads.html から

shibboleth-SP 関連のインストールファイルをダウンロードします。

【対象ファイル】

log4shib-1.0-1.i386.rpm	shibboleth-docs-2.0-6.i386.rpm
log4shib-debuginfo-1.0-1.i386.rpm	xerces-c-2.8.0-1.i386.rpm
log4shib-devel-1.0-1.i386.rpm	xerces-c-debuginfo-2.8.0-1.i386.rpm
log4shib-doc-1.0-1.i386.rpm	xerces-c-devel-2.8.0-1.i386.rpm
opensaml-2.0-6.i386.rpm	xerces-c-doc-2.8.0-1.i386.rpm
opensaml-debuginfo-2.0-6.i386.rpm	xml-security-c-1.4.0-1.i386.rpm
opensaml-devel-2.0-6.i386.rpm	xml-security-c-debuginfo-1.4.0-1.i386.rpm
opensaml-docs-2.0-6.i386.rpm	xml-security-c-devel-1.4.0-1.i386.rpm
shibboleth-2.0-6.i386.rpm	xmltooling-1.0-6.i386.rpm
shibboleth-debuginfo-2.0-6.i386.rpm	xmltooling-debuginfo-1.0-6.i386.rpm
shibboleth-devel-2.0-6.i386.rpm	xmltooling-devel-1.0-6.i386.rpm
	xmltooling-docs-1.0-6.i386.rpm

②インストール

上記の shibboleth-SP 関連のファイルをインストールします。

```
# rpm -ivh log4shib-1.0-1.i386.rpm ¥
      xerces-c-2.8.0-1.i386.rpm ¥
      xml-security-c-1.4.0-1.i386.rpm ¥
      xmltooling-1.0-6.i386.rpm ¥
      opensaml-2.0-6.i386.rpm ¥
      shibboleth-2.0-6.i386.rpm
```

※依存関係上、上記の順番でインストールする必要があります。

その他のパッケージは必要に応じてインストールしてください。

unixODBC がインストールされていないと依存関係チェック時にエラーが表示されます。必要な場合は、以下のサイトからダウンロードしてください。

http://www.unixodbc.org/

③ httpd 設定

/etc/httpd/conf.d/ssl.conf にて、 ServerName を設定します。

```
ServerName upkishibSP.nii.ac.jp:443      ← ホスト名を設定
```

④ shibd 自動起動設定

shibd を OS 起動時に自動起動するには、以下のコマンドで設定します。
(オプションは マイナス ‘ー‘ が２つ必要です)

```
# chkconfig --add shibd
# chikconfig --level 345 shibd on
```

［4］shibboleth の設定

デフォルトでは shibboleth は /etc/shibboleth ディレクトリにインストールされます。変更する各設定ファイルも 同ディレクトリ配下にあります。
また、ログファイルは /var/log/shibboleth ディレクトリに出力されます。

・SP サーバのメタデータ

以下の UPKI イニシアティブのサイトの「SSO 実証実験のリポジトリ」から SP 用メタデータテンプレートをダウンロードし、必要な項目を変更します。完成した新しい SP 用のメタデータを、ヘルプデスクへ送付してください。
ヘルプデスクでは、送付していただいたファイルをもとに、DS に登録し、また共用メタデータを更新します。 更新完了後、共用メタデータをダウンロードしてファイルを配置します。
詳しくは、以下のサイトを参照してください。

https://upki-portal.nii.ac.jp/SSO/Repository

※ サイトへのアクセスには、UPKI イニシアティブ会員への登録が必要となります。

・shibboleth2.xml ファイル

/etc/shibboleth/shibboleth2.xml ファイルを以下の様に変更します。

※「<Host name="sp.example.org">」を検索し、場所を特定してください。（行番号は参考です）

```
62              <Host name="upkishibSP.nii.ac.jp">      ← ホスト名変更
63                  <Path name="secure" authType="shibboleth" requireSession="true"/>
64              </Host>
(省略)
77      <ApplicationDefaults id="default" policyId="default"
78          entityID=https://upkishibSP.nii.ac.jp/shibboleth-sp   ← ホスト名変更
79          homeURL="https://upkishibSP.nii.ac.jp/index.html"
80          REMOTE_USER="eppn persistent-id targeted-id"
81          signing="false" encryption="false"
82          >
```

※「Default example directs」を検索し、場所を特定してください。（行番号は参考です）

```
104             <!-- Default example directs to a specific IdP's SSO service
                                                    (favoring SAML 2 over Shib 1). -->
105             <SessionInitiator type="Chaining" Location="/Login"      isDefault="true"
                    id="Intranet"   relayState="cookie"
106                 entityID="https:// upkishibIdP.nii.ac.jp /idp/shibboleth">
                                  ↑
                    ※metadataに設定されているIdPのentityIDの内容を設定してください。

107                 <SessionInitiator type="SAML2" defaultACSIndex="1"
                                                    template="bindingTemplate.html"/>
108                 <SessionInitiator type="Shib1" defaultACSIndex="5"/>
109             </SessionInitiator>
(省略)                                                          ↓追加
129             <SessionInitiator type="Chaining" Location="/DS" isDefault="false" id="DS"
                                                                relayState="cookie">
130                 <SessionInitiator type="SAML2" defaultACSIndex="1" template=
                        "bindingTemplate.html" acsByIndex="false"/>
131                 <SessionInitiator type="Shib1" defaultACSIndex="5" acsByIndex="false"/>
132               <SessionInitiator type="SAMLDS" URL="https://upkishibDS.nii.ac.jp/ds/WAYF"/>
133             </SessionInitiator>                                 ↑DS 名
```

※「Example of locally maintained metadata」を検索し、場所を特定してください。（行番号は参考です）

```
217             <!-- Example of locally maintained metadata. -->
218             <!-- -->
219             <MetadataProvider type="XML" file=" /etc/shibboleth /partner-metadata.xml"/>
220             <!-- -->                          ↑コメントを外し、メタデータのパス書きます
221         </MetadataProvider>
```

［5］サービスを起動・停止方法

httpd の起動方法

service httpd start

tomcat の起動方法

service tomcat start　　(jsvc を利用した場合)

shibd の起動方法

service shibd stop

httpd の停止方法

service httpd stop

tomcat の停止方法

service tomcat start　　(jsvc を利用した場合)

shibd の停止方法

service shibd stop

［6］SP への接続確認

①httpd サービスと、shibd サービスを再起動する。

```
# service httpd restart
# service shibd restart
```

② SP にアクセスする

サーバ上のブラウザで、設定した SP にアクセスします。

https://localhost/Shibboleth.sso/Status

(サーバ名は必ず localhost として下さい)

画面上に ok が表示されれば SP に接続が確認出来ました。

［7］IdP との SP 接続確認

接続する IdP の設定変更も必要となります。設定変更は IdP の管理者に依頼して下さい。

① SP にテスト用のファイルを用意します。ファイルの内容は以下の１行です。

/var/www/html/secure/phpinfo.php

```
<?php phpinfo(); ?>
```

② SP のメタデータに IdP への接続設定を追加します。

直接リダイレクトする IdP のメタデータ

/opt/shibboleth-idp-2.0.0/metadata/idp-metadata.xml

に記述された、その IdP の<EntityDescriptor> ～ </EntityDescriptor>部分と同じ内容を全て、

SP の /etc/shibboleth/partner-metadata.xml に追加します。

```
<EntitiesDescriptor Name="urn:mace:shibboleth:testshib:two"
(省略)
        <EntityDescriptor entityID="https:// upkishibIdP.nii.ac.jp/idp/shibboleth">
(省略)                                          ↑ 直接リダイレクトする IdP
        </EntityDescriptor>
(省略)
</EntitiesDescriptor>
```

③ IdP に追加した SP の設定を追加します。

SP の/etc/shibboleth/partner-metadata.xml

に記載された この SP の<EntityDescriptor> ～ </EntityDescriptor>部分と同じ内容を全て、IdP の /opt/shibboleth-idp-2.0.0/metadata/idp-metadata.xml に追加します。

```
<EntitiesDescriptor Name="urn:mace:shibboleth:testshib:two"
(省略)
        <EntityDescriptor entityID="https:// upkishibSP.nii.ac.jp/idp/shibboleth">
(省略)                                          ↑ この SP のホスト名
        </EntityDescriptor>
(省略)
```

⑤ SP, IdP 共にサービスを再起動します。

⑥ ブラウザから SP の①で用意したファイルへアクセスします。

https:// upkishibSP /secure/phpinfo.php

⑦ IdP のログイン画面が表示され、ID, パスワードを入力してログインした後、表示される環境変数に、IdP で公開する設定とした値(LDAP に保存されている eduPersonPrincipalName など)が含まれていることを確認します。
これが、SSO により IdP から渡されたユーザの属性情報となります。

表示例）

PHP Variables

variable	value
_SERVER[“unscoped-affiliation”]	faculty

⑧ /etc/shibboleth/shibboleth2.xml ファイルに、接続する DS を設定します。
IdP へ直接リダイレクトせず、DS を用いる設定を行います。

※「Default example directs」を検索し、場所を特定してください。（行番号は参考です）

```
104          <!-- Default example directs to a specific IdP's SSO service
                                                (favoring SAML 2 over Shib 1). -->
105          <SessionInitiator type="Chaining" Location="/Login" isDefault="false"←false とする
                 id="Intranet"   relayState="cookie"
106                  entityID="https:// upkishibIdP.nii.ac.jp /shibboleth">
107              <SessionInitiator type="SAML2" defaultACSIndex="1"
                                                template="bindingTemplate.html"/>
108              <SessionInitiator type="Shib1" defaultACSIndex="5"/>
109          </SessionInitiator>
(省略)                                                          ↓true とする
129          <SessionInitiator type="Chaining" Location="/DS" isDefault="true" id="DS"
                                                                relayState="cookie">
130              <SessionInitiator type="SAML2" defaultACSIndex="1" template=
                   "bindingTemplate.html" acsByIndex="false"/>
131              <SessionInitiator type="Shib1" defaultACSIndex="5" acsByIndex="false"/>
132            <SessionInitiator type="SAMLDS" URL="https://upkishibDS.nii.ac.jp/ds/WAYF"/>
133          </SessionInitiator>
```

［8］サーバ証明書を申請、登録する

①「7．シングルサインオン実証実験用 サーバ証明書の取得方法について」を参考に、
サーバ証明書を申請します。
証明書の交付までには数日を要するので、お早めに申請してください。

②入手したサーバ証明書をもとに、以下のファイルに設定してください。

■/etc/httpd/conf.d/ssl.conf

```
（省略）
SSLCertificateFile /etc/shibboleth/cert/upkishibSP.crt ←サーバ証明書の格納先
SSLCertificateKeyFile  /etc/shibboleth/cert/upkishibSP.key ←サーバ秘密鍵の格納先
#SSLCACertificateFile /etc/pki/tls/certs/ca-bundle.crt ←コメントアウト
SSLCACertificatePath /etc/shibboleth/cert/CA      ←CA 証明書の格納先
（省略）
```

/etc/shibboleth/cert/CA ディレクトリが無い場合は作成してください。このディレクトリには、ファイル名をハッシュ値とした中間 CA 証明書を配置します。
詳しくは UPKI「サーバ証明書プロジェクト」 https://upki-portal.nii.ac.jp/cerpj
を参照してください。

■/etc/shibboleth/shibboleth2.xml

```
（省略）
        <!-- Simple file-based resolver for using a single keypair. -->
        <CredentialResolver type="File" key="cert/upkishibSP.key" certificate="cert/upkishibSP.crt"/>
                                                ←サーバ証明書，秘密鍵の格納先
```

③メタデータの更新
以下の UPKI イニシアティブのサイトの「SSO 実証実験のリポジトリ」から SP 用メタデータテンプレートをダウンロードし、必要な項目を変更します。

https://upki-portal.nii.ac.jp/SSO/Repository
※サイトへのアクセスには、UPKＩイニシアティブ会員への登録が必要となります。

ダウンロードしたメタデータテンプレートを下記のように変更してください。

（中略）

<EntityDescriptor entityID="https://upkishibSP.nii.ac.jp/idp/shibboleth">　←ホスト名

<IDPSSODescriptor protocolSupportEnumeration="urn:mace:shibboleth:1.0
urn:oasis:names:tc:SAML:2.0:protocol">

<KeyDescriptor>

<ds:KeyInfo>

<ds:X509Data>

<ds:X509Certificate>

MIIDOzCCAiOgAwIBAgIUdTJ6oiEccCjrtDyDaeBXTlRpfPcwDQYJKoZIhvcNAQEF
BQAwHzEdMBsGA1UEAxMUdXBraXNoaWIxMS5uaWkuYWMuanAwHhcNMDgwNzA4MTAz
NTQwWhcNMjgwNzA4MTAzNTQwWjAfMR0wGwYDVQQDExR1cGtpc2hpYjExLm5paS5h
Yy5qcDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAJlkO5kZI2Tz7tPg
1HVHwv7g7bCYNL7Zx11IeNwDyd1ZluRAfmSzTNP67YPSMPLaA4upkYM51JmHgskZ
GvbjNqfma+imNVo/R7J0LL6ucSgL++ax25XKJmViCspQpgYPiFsavoF6JwciRdwk
zeUkhJo0zZWZ1rSxeAevAuWJf9hDwelCyry5u2ZNljDNLFl6uzpPETv3DSMxwevs
tHggaq9917DSnH4ZhiBhXL3pd+g0qyw0ouuew9wtizZ6KpTtTll3InuTM6KiCG5M
Iyv7HVc9KtwkVAooF/LMPP9ofkZeuqzpc8T6Wg+zaUUsIKhEDhon4Zb/rt9tS3vB
JFpYBS8CAwEAAaNvMG0wTAYDVR0RBEUwQ4lUdXBraXNoaWIxMS5uaWkuYWMuanCG
K2h0dHBzOi8vdXBraXNoaWIxMS5uaWkuYWMuanAvaWRwL3NoaWJib2xldGgwHQYD
VR0OBBYEFCoPX1gOojzXlCTZT7l73KcHkJSHMA0GCSqGSIb3DQEBBQUAA4IBAQAW
GnudDV3eqTNLZPGH8zJWHCT8Az7CtG40aINRJzirbZI+r4X7Zuq5ZLv+n9EJ6rbd
xRWh6bIx9YTLKcLvzXx0ZM4fy6RFyJ+8qCDTXig0qDgVpng66xfJBi7ahjGIJgn6
xXwdYFE50zLC3qwrZ9kykXCy2ELLHfb3Z/g1o9fZZy7gjn77m1tDfWcs4M3NFCfL
zKbGNi+5a05w/wLkxpEaP8NPTHkbN3E+EXQDik7QQOqGJ0+JEUYLAPO6HTGGCs5i
YU+cTQ5QSgjfsSwcZQt6ljQUzlyhKOAWnazbrRGVfCVlwoYl0hkpmGMSb4Jjxo6E
61psWSAHlehx6L2F9Eat

</ds:X509Certificate>

入手した証明書に変更

```
                    </ds:X509Data>
                </ds:KeyInfo>
            </KeyDescriptor>
            <NameIDFormat>urn:mace:shibboleth:1.0:nameIdentifier</NameIDFormat>
            <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat>
            <SingleSignOnService Binding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"
                    Location="https://upkishibSP.nii.ac.jp/idp/profile/Shibboleth/SSO" />    ←ホスト名
            <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
                    Location="https://upkishibSP.nii.ac.jp/idp/profile/SAML2/POST/SSO" />    ←ホスト名
            <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
                    Location="https://upkishibSP.nii.ac.jp/idp/profile/SAML2/Redirect/SSO" />    ←ホスト名
        </IDPSSODescriptor>
        <Organization>
                <OrganizationName xml:lang="en">Test SP</OrganizationName>    ←組織名
                <OrganizationDisplayName xml:lang="en">Test SP</OrganizationDisplayName>    ←組織表示名
                <OrganizationURL xml:lang="en">http://YourHomePage/</OrganizationURL>    ←組織 URL
         </Organization>
         <ContactPerson contactType="technical">
                <GivenName>YourGivenName</GivenName>
                <SurName>YourSurName</SurName>    ←管理者名
                <EmailAddress>YourEmailAddress</EmailAddress>    ←管理者の e-mail アドレス
         </ContactPerson>
    </EntityDescriptor>
（中略）
```

完成した新しい SP 用のメタデータテンプレートを、ヘルプデスク（upki-sso-help@nii.ac.jp）へ送付してください。
ヘルプデスクでは、送付していただいたファイルをもとに、DS に登録するとともに共用メタデータを更新します。
詳しくは、以下のサイトを参照してください。

https://upki-portal.nii.ac.jp/SSO/Repository

※サイトへのアクセスには、UPKＩイニシアティブ会員への登録が必要となります。

④同様に以下の UPKI イニシアティブのサイトの「SSO 実証実験のリポジトリ」から upki-fed-metadata.xml をダウンロードし、ファイル名を partner-metadata.xml に変更して、/etc/shibboleth/partner-metadata.xml と差し替えてください。

https://upki-portal.nii.ac.jp/SSO/Repository

※サイトへのアクセスには、UPKI イニシアティブ会員への登録が必要となります。

⑤接続する IdP が、貴学にて構築した場合、メタデータを更新する必要があります。IdP 構築手順書を参照の上、設定をしてください。

３．Plone での SP 設定例

以下に、SP として Plone をインストールし、Shibbleth 連携させる例を記載します。

［１］plone（含む Zope）のダウンロード

以下の Web サイトからダウンロードしてください。

http://plone.org/products/plone

※NII では Plone-3.0.6-UnifiedInstaller.tgz を使用して検証しています。

［２］plone（含む Zope）のインストール

```
（ここでは/tmp/plone 配下にダウンロードしたファイルを置いています。）
#tar zxvf Plone-3.0.6-UnifiedInstaller.tgz　←解凍
#cd /tmp/plone/Plone-3.0.6-UnifiedInstaller
#./install.sh standalone　←インストール
```

次に Zope インスタンスを作成します。

```
# /opt/Plone-3.0.6/bin/mkzopeinstance.py
Directory: /opt/Plone-3.0.6/zinstance/Products
Username:admin　←ここではユーザ名を[admin]とした
Password:******　←パスワードを入力
Verify password::****** ←パスワードを再入力
```

［３］サービスの起動

```
# /opt/Plone-3.0.6/zinstance/bin/runzope
# /opt/Plone-3.0.6/zinstance/bin/zopectl start
```

［4］Plone Site の生成

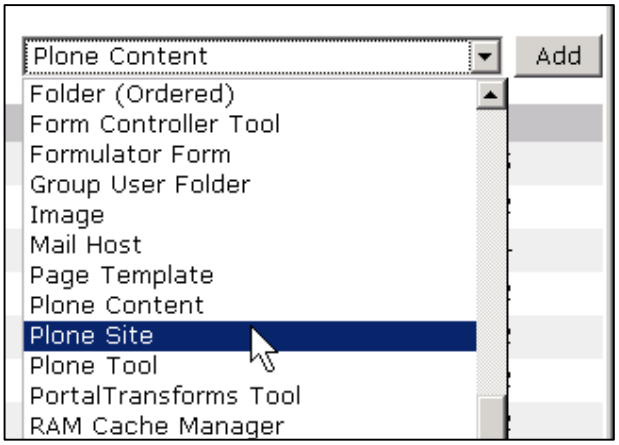
①以下の URL にアクセス

http://localhost:8080/manage

②ログイン

［2］で作成したアカウントでログインしてください。

③管理画面の右上のリストボックスから 「Plone Site」 を選択して「add」します。



④「Id」、「Title」、「Menbership source」、「Description」に入力し、「Add Plone Site」を押下します。

・「Id」：Plone Site の ID です。URL の一部となるので、英字のみとしてください。
ここでは Plone と入力しました。

・「Title」：Plone Site のタイトルです。Site を表す名前をつけてください。

・「Menbership source」：ユーザの設定をどこで行なうかを、下記の 2 種類から選択します。

「Create a new user folder in the portal」
こちらを選択すると、この Plone Site 用に新規にユーザフォルダを作成します。通常はこちらを選択することをおすすめします。

「I have an existing user folder and want to use that instead」
こちらは、既存のユーザフォルダを使用する設定です。Zope にすでに多数のユーザが登録されていたり、Zope 全体で統一してユーザ情報を扱いたい場合などにはこちらを選択してください。

・「Description」：その Plone Site がどのような Site なのかを説明する文章を記述してください。

⑤Web ブラウザで http://localhost:8080/plone/ にアクセスすると、下記のように Plone のトップページが表示されます。

［5］plone-shibbleth 接続設定

①以下の URL から、接続用プラグインをダウンロードします。

【URL】

http://tid.ithaka.org/software

【ダウンロードファイル】

・AutoUserMakerPASPlugin.zip
・ShibbolethLogin.zip
・ShibbolethPermissions.zip

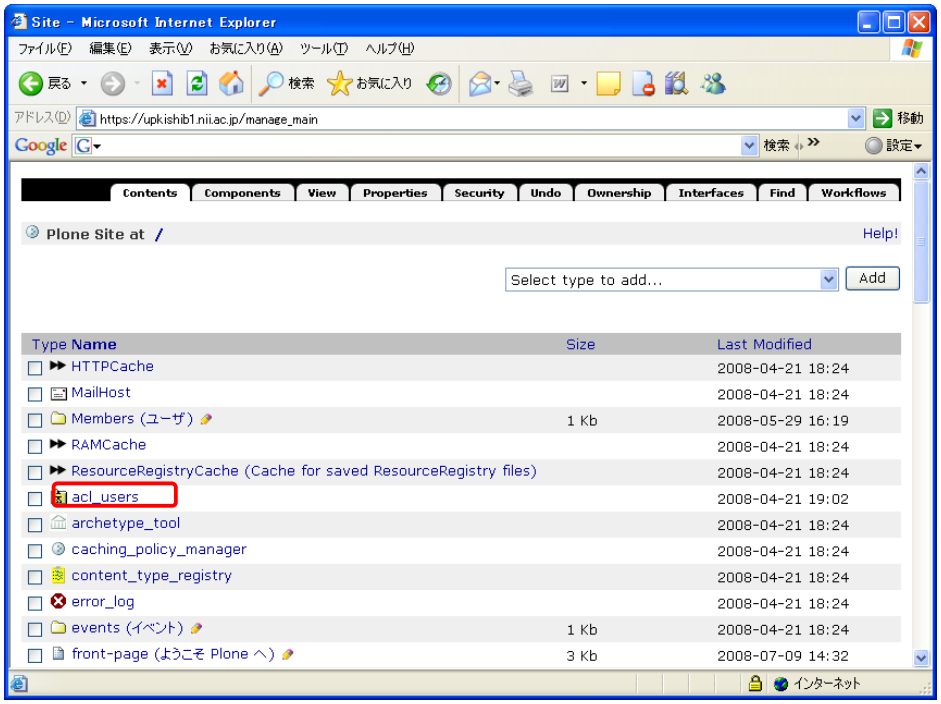
②上記のファイルを解凍し、/opt/Plone-3.0.6/zinstance/Products 配下に移動します。

③次に、Web ブラウザにて Plone を開き(http://localhost:8080/Plone)、管理者アカウントでログインし、[サイト設定] から [アドオンプロダクト] を開きます。

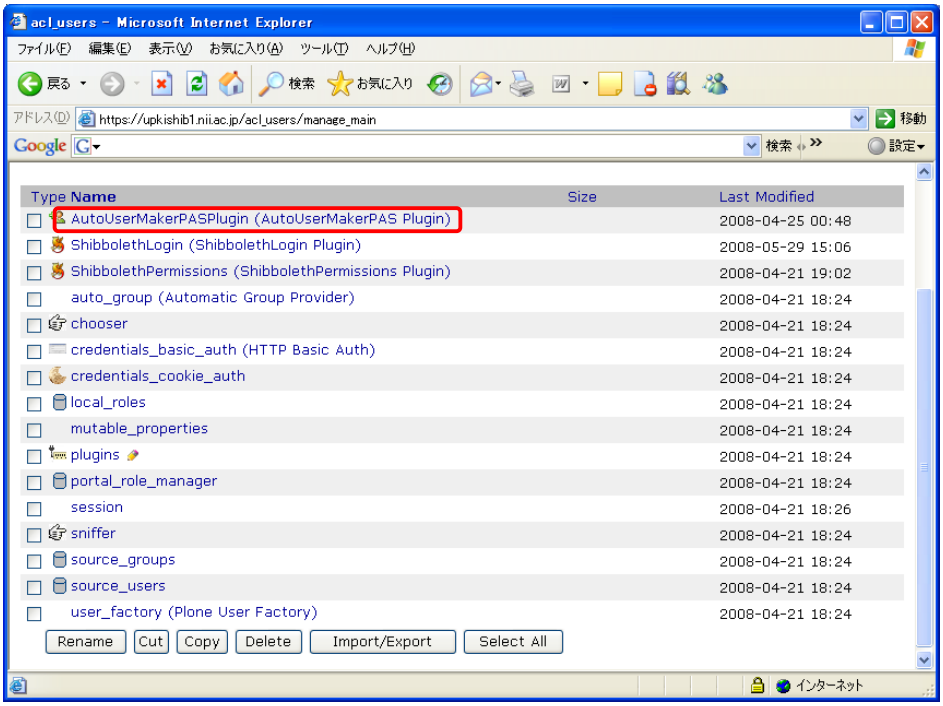
すると、[インストール可能なプロダクト] として、下記３つをチェック入れて、[インストール]ボタンを押してください。

・AutoUserMakerPASPlugin
・ShibbolethLogin
・ShibbolethPermissions

④続いて「Zope 管理画面(ZMI)」から「acl_users」を選択します。



⑤次に「AutoUserMakerPASPlugin」を選択します



⑥下記の値を設定し、"save"ボタン"を押下してください。

・User Setup Headers にて、下記を設定
- User ID: HTTP_EPPN
- user's full name: HTTP_GIVENNAME
- User's description: HTTP_SHIB_ORGPERSON_TITLE
- User's email: HTTP_MAIL
- User's locality: HTTP_SHIB_ORGPERSON_LOCALITY
- User's state: HTTP_SHIB_ORGPERSON_STATE
- User's country: HTTP_SHIB_ORGPERSON_C

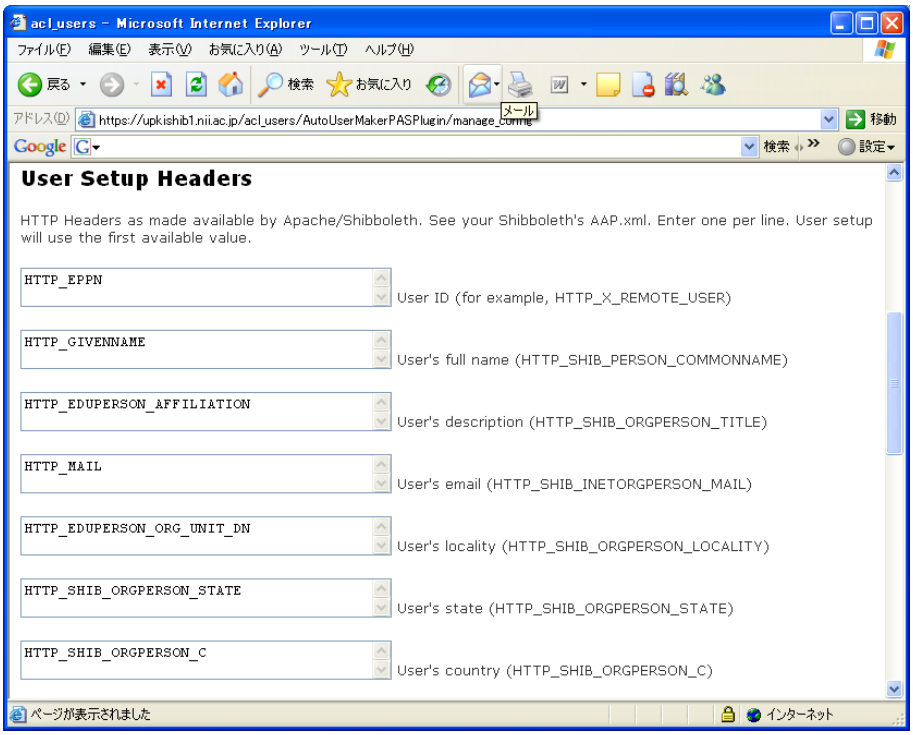
・User Mapping Headers に下記を設定
- HTTP_EPPN

・User Sharing Headers に下記を設定
（上のボックスに下記２つを設定）
- HTTP_EPPN

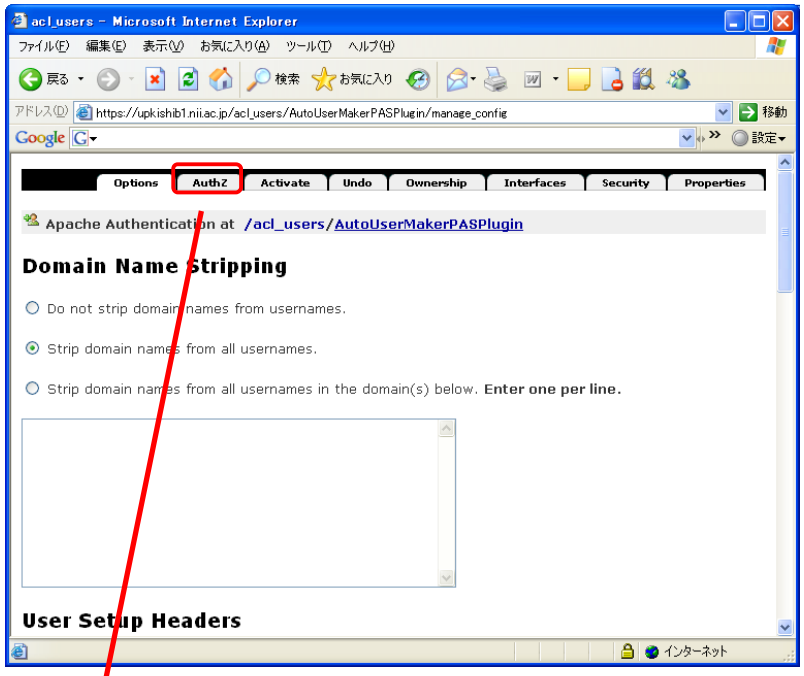
HTTP_EDUPERSON_ORG_UNIT_DN

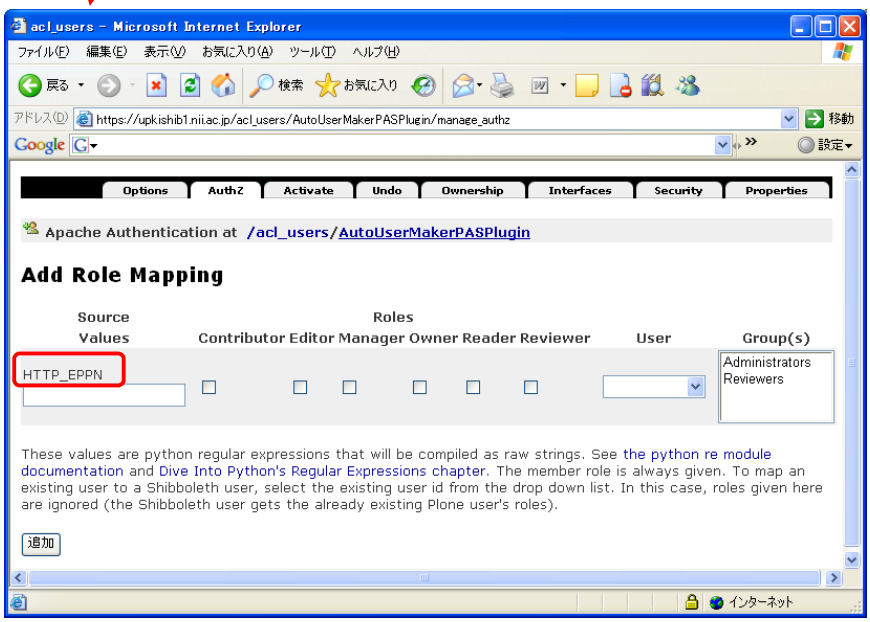
（下のボックスに下記２つを設定）
- User ID

Organization



⑦続いて「AuthZ」タブを選択し、「Values」を HTTP_EPPN に変更してください。





⑧⑤の画面から「Shibboleth Login」を選択し、以下の設定を行い"save"ボタン"をクリックします。

※下記のパスは、shibboleth2.xml で設定されたものです。

接続する IdP と DS は実績のある upkishib10.nii.ac.jp 、upkishib1.nii.ac.jp とします。

・The Shibboleth provider ID をブランクにする。

・The URL to use by the Identity Provider に下記を設定

- https://upkishibSP.nii.ac.jp/Shibboleth.sso/SAML2/POST

・labels for the URLs below に下記を設定

- Test IdP

Test DS

・URLs to redirect to for initiating a session.に下記を設定

- https://upkishibSP.nii.ac.jp/Shibboleth.sso/Login

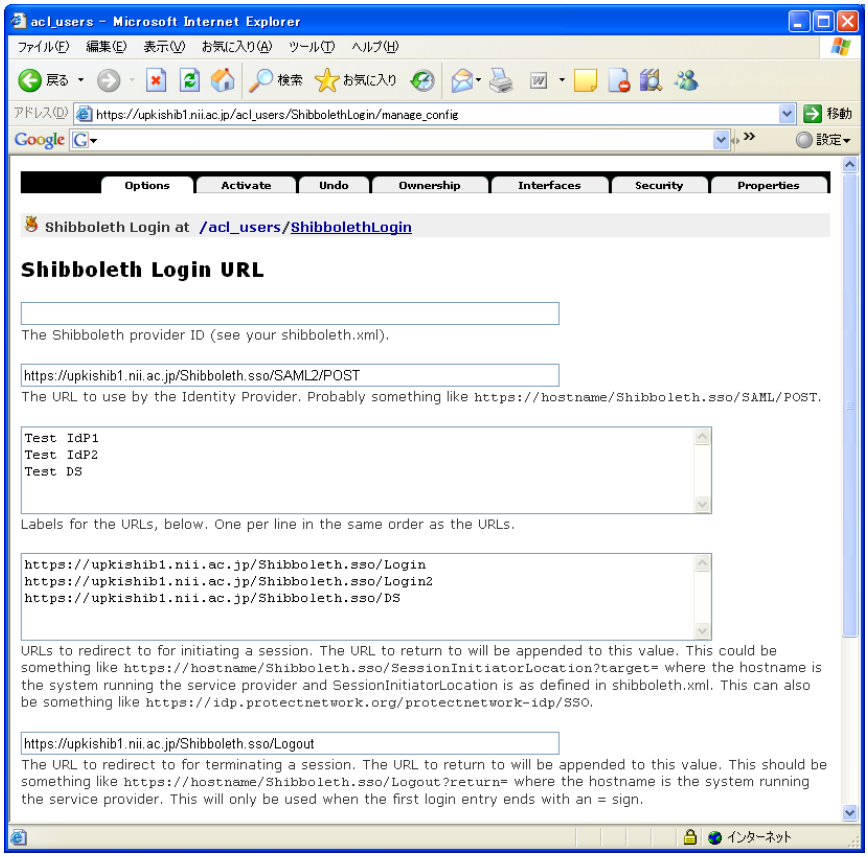
https://upkishibSP.nii.ac.jp/Shibboleth.sso/DS

/etc/shibboleth/shibboleth2.xml の設定と合わせる必要があります。

P.9 を参照してください。

・The URL to redirect to for terminating a session に下記を設定

- https://upkishib1.nii.ac.jp/Shibboleth.sso/Logout



⑨ssl.conf の修正

```
（中略）
<VirtualHost _default_:443>
（中略）
ProxyRequests Off
ProxyPass /Shibboleth.sso !
ProxyPass / http://127.0.0.1:8080/VirtualHostBase/https/upkishibSP.nii.ac.jp:443/plone/VirtualHostRoot/
  <Location />
        AuthType shibboleth
        ShibRequireSession Off
        ShibUseHeaders On
        Require shibboleth
  </Location>
（中略）
</VirtualHost>
```

⑩/etc/shibboleth/attribute-map.xml の編集

```
<!-- First some useful eduPerson attributes that many sites might use. -->
（中略）
<Attribute name="urn:mace:dir:attribute-def:eduPersonAffiliation" id="unscoped-affiliation">      ←コメント化
  ↓
<Attribute name="urn:mace:dir:attribute-def:eduPersonAffiliation" id="eduPerson_Affiliation">  ←置換
（中略）
<Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1" id="unscoped-affiliation">      ←コメント化
  ↓
<Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1" id="eduPerson_Affiliation">  ←置換
（中略）
<!-- Some more eduPerson attributes, uncomment these to use them... -->  ←このタグ配下のコメントを外す
（中略）
<!--Examples of LDAP-based attributes, uncomment to use these... -->      ←このタグ配下のコメントを外す
（中略）
```

⑪/etc/shibboleth/attribute-policy.xml の編集

```
（中略）
<!--          <afp:AttributeRule attributeID="affiliation">      ←コメント化
              <afp:PermitValueRule xsi:type="AND">
                  <RuleReference ref="eduPersonAffiliationValues"/>
                  <RuleReference ref="ScopingRules"/>
              </afp:PermitValueRule>
          </afp:AttributeRule>
          <afp:AttributeRule attributeID="unscoped-affiliation">
              <afp:PermitValueRuleReference ref="eduPersonAffiliationValues"/>
          </afp:AttributeRule>
          <afp:AttributeRule attributeID="eppn">
              <afp:PermitValueRuleReference ref="ScopingRules"/>
          </afp:AttributeRule>      -->
（中略）
```

⑫IdP 側の属性設定（IdP に upkishib10.nii.ac.jp を利用する場合は、設定済みです。）

■/opt/shibboleth-idp-2.0.0/conf/attribute-resolver.xml

```
（中略）
   <!-- Schema: Core schema attributes-->
（中略）
    <resolver:AttributeDefinition id="email" xsi:type="Simple" xmlns="urn:mace:shibboleth:2.0:resolver:ad"
        sourceAttributeID="mail">                          ←有効化
        <resolver:Dependency ref="myLDAP" />
        <resolver:AttributeEncoder xsi:type="SAML1String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:mace:dir:attribute-def:mail" />
        <resolver:AttributeEncoder xsi:type="SAML2String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:oid:0.9.2342.19200300.100.1.3" friendlyName="mail" />
    </resolver:AttributeDefinition>
    <resolver:AttributeDefinition id="givenName" xsi:type="Simple"       ←有効化
        xmlns="urn:mace:shibboleth:2.0:resolver:ad"      sourceAttributeID="givenName">
        <resolver:Dependency ref="myLDAP" />
        <resolver:AttributeEncoder xsi:type="SAML1String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:mace:dir:attribute-def:givenName" />
        <resolver:AttributeEncoder xsi:type="SAML2String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:oid:2.5.4.42" friendlyName="givenName" />
    </resolver:AttributeDefinition>
（中略）
<!-- Schema: eduPerson attributes -->
   <resolver:AttributeDefinition id="eduPersonAffiliation" xsi:type="Simple"       ←有効化
        xmlns="urn:mace:shibboleth:2.0:resolver:ad"    sourceAttributeID="eduPersonAffiliation">
        <resolver:Dependency ref="myLDAP" />
        <resolver:AttributeEncoder xsi:type="SAML1String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:mace:dir:attribute-def:eduPersonAffiliation" />
        <resolver:AttributeEncoder xsi:type="SAML2String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1" friendlyName="eduPersonAffiliation" />
    </resolver:AttributeDefinition>
```

```
    <resolver:AttributeDefinition id="eduPersonEntitlement" xsi:type="Simple"     ←有効化
        xmlns="urn:mace:shibboleth:2.0:resolver:ad"      sourceAttributeID="eduPersonEntitlement">
        <resolver:Dependency ref="myLDAP" />
        <resolver:AttributeEncoder xsi:type="SAML1String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:mace:dir:attribute-def:eduPersonEntitlement" />
        <resolver:AttributeEncoder xsi:type="SAML2String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7" friendlyName="eduPersonEntitlement" />
    </resolver:AttributeDefinition>
    <resolver:AttributeDefinition id="eduPersonOrgUnitDN" xsi:type="Simple"     ←有効化
        xmlns="urn:mace:shibboleth:2.0:resolver:ad"      sourceAttributeID="eduPersonOrgUnitDN">
        <resolver:Dependency ref="myLDAP" />
        <resolver:AttributeEncoder xsi:type="SAML1String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:mace:dir:attribute-def:eduPersonOrgUnitDN" />
        <resolver:AttributeEncoder xsi:type="SAML2String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
            name="urn:oid:1.3.6.1.4.1.5923.1.1.1.4" friendlyName="eduPersonOrgUnitDN" />
    </resolver:AttributeDefinition>
    <resolver:AttributeDefinition id="principalName" xsi:type="Scoped"     ←有効化
      xmlns="urn:mace:shibboleth:2.0:resolver:ad" scope="nii.ac.jp" sourceAttributeID="remoteUser">
        <resolver:Dependency ref="remoteUser" />
        <resolver:AttributeEncoder xsi:type="SAML1ScopedString"
          xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
          name="urn:mace:dir:attribute-def:eduPersonPrincipalName" />
        <resolver:AttributeEncoder xsi:type="SAML2ScopedString"
          xmlns="urn:mace:shibboleth:2.0:attribute:encoder" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
          friendlyName="eduPersonPrincipalName" />
    </resolver:AttributeDefinition>
    <resolver:AttributeDefinition xsi:type="PrincipalName" xmlns="urn:mace:shibboleth:2.0:resolver:ad"
     id="remoteUser" />   ←追加
(中略)
```

```
（中略）
  <!-- ========================================== -->
   <!--         Data Connectors                    -->
   <!-- ========================================== -->
   <!-- Example Static Connector -->        ←このタグにある定義をすべてコメント化
（中略）
   <!-- Example LDAP Connector -->
   <resolver:DataConnector id="myLDAP" xsi:type="LDAPDirectory"    ←有効化
       xmlns="urn:mace:shibboleth:2.0:resolver:dc"
       ldapURL="ldap://upkishib10.nii.ac.jp" baseDN="o=Test_University_A,dc=ac,c=JP"    ←LDAP に
       principal="cn=olmgr,o=Test_University_A,dc=ac,c=JP"                              合わせて修正
       principalCredential="csildap">
       <FilterTemplate>
           <![CDATA[
               (eduPersonPrincipalName=$requestContext.principalName)
           ]]>
       </FilterTemplate>
   </resolver:DataConnector>
（中略）
```

※接続する LDAP がない場合は、

１）<resolver:Dependency ref="myLDAP" /> を
<resolver:Dependency ref="staticAttributes" />に変更し、

２）<!-- Example Static Connector -->
<resolver:DataConnector id="staticAttributes" xsi:type="Static"
xmlns="urn:mace:shibboleth:2.0:resolver:dc">タグ内で
上記の項目を static に定義してください。

■/opt/shibboleth-idp-2.0.0/conf/attribute-filter.xml

```
（中略）
    <!-- Example Static Connector -->
    <!--   Release the transient ID to anyone -->
    <AttributeFilterPolicy id="releaseTransientIdToAnyone">
        <AttributeRule attributeID="transientId">      ←有効化
            <PermitValueRule xsi:type="basic:ANY" />
        </AttributeRule>
        <AttributeRule attributeID="principalName">      ←有効化
            <PermitValueRule xsi:type="basic:ANY" />
        </AttributeRule>
        <AttributeRule attributeID="givenName">      ←有効化
            <PermitValueRule xsi:type="basic:ANY" />
        </AttributeRule>
        <AttributeRule attributeID="eduPersonAffiliation">      ←有効化
            <PermitValueRule xsi:type="basic:ANY" />
        </AttributeRule>
        <AttributeRule attributeID="eduPersonOrgUnitDN">      ←有効化
            <PermitValueRule xsi:type="basic:ANY" />
        </AttributeRule>
（中略）
```

⑬サーバ証明書の設定

■接続する SP のサーバ証明書が、IdP 側の/opt/shibboleth-idp-2.0.0/metadata/idp-metadata.xml に登録されているか確認してください。

```
（中略）
        <EntityDescriptor entityID="https://upkishibSP.nii.ac.jp/idp/shibboleth">   ←ホスト名
        <IDPSSODescriptor protocolSupportEnumeration="urn:mace:shibboleth:1.0
         urn:oasis:names:tc:SAML:2.0:protocol">
           <KeyDescriptor>
              <ds:KeyInfo>
                 <ds:X509Data>
                    <ds:X509Certificate>
MIIDOzCCAiOgAwIBAgIUdTJ6oiEccCjrtDyDaeBXTlRpfPcwDQYJKoZIhvcNAQEF
BQAwHzEdMBsGA1UEAxMUdXBraXNoaWIxMS5uaWkuYWMuanAwHhcNMDgwNzA4MTAz
NTQwWhcNMjgwNzA4MTAzNTQwWjAfMR0wGwYDVQQDExR1cGtpc2hpYjExLm5paS5h
Yy5qcDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAJlkO5kZI2Tz7tPg
1HVHwv7g7bCYNL7Zx11IeNwDyd1ZluRAfmSzTNP67YPSMPLaA4upkYM51JmHgskZ
GvbjNqfma+imNVo/R7J0LL6ucSgL++ax25XKJmViCspQpgYPiFsavoF6JwciRdwk
zeUkhJo0zZWZ1rSxeAevAuWJf9hDwelCyry5u2ZNljDNLFl6uzpPETv3DSMxwevs
tHggaq9917DSnH4ZhiBhXL3pd+g0qyw0ouuew9wtizZ6KpTtTll3InuTM6KiCG5M
Iyv7HVc9KtwkVAooF/LMPP9ofkZeuqzpc8T6Wg+zaUUsIKhEDhon4Zb/rt9tS3vB
JFpYBS8CAwEAAaNvMG0wTAYDVR0RBEUwQ4IUdXBraXNoaWIxMS5uaWkuYWMuanCG
K2h0dHBzOi8vdXBraXNoaWIxMS5uaWkuYWMuanAvaWRwL3NoaWJib2xldGgwHQYD
VR0OBBYEFCoPX1gOojzXlCTZT7l73KcHkJSHMA0GCSqGSIb3DQEBBQUAA4IBAQAW
GnudDV3eqTNLZPGH8zJWHCT8Az7CtG40aINRJzirbZI+r4X7Zuq5ZLv+n9EJ6rbd
xRWh6bIx9YTLKcLvzXx0ZM4fy6RFyJ+8qCDTXig0qDgVpng66xfJBi7ahjGIJgn6
xXwdYFE50zLC3qwrZ9kykXCy2ELLHfb3Z/g1o9fZZy7gjn77m1tDfWcs4M3NFCfL
zKbGNi+5a05w/wLkxpEaP8NPTHkbN3E+EXQDik7QQOqGJ0+JEUYLAPO6HTGGCs5i
YU+cTQ5QSgjfsSwcZQt6ljQUzlyhKOAWnazbrRGVfCVlwoYl0hkpmGMSb4Jjxo6E
61psWSAHlehx6L2F9Eat
（中略）
```

入手した証明書

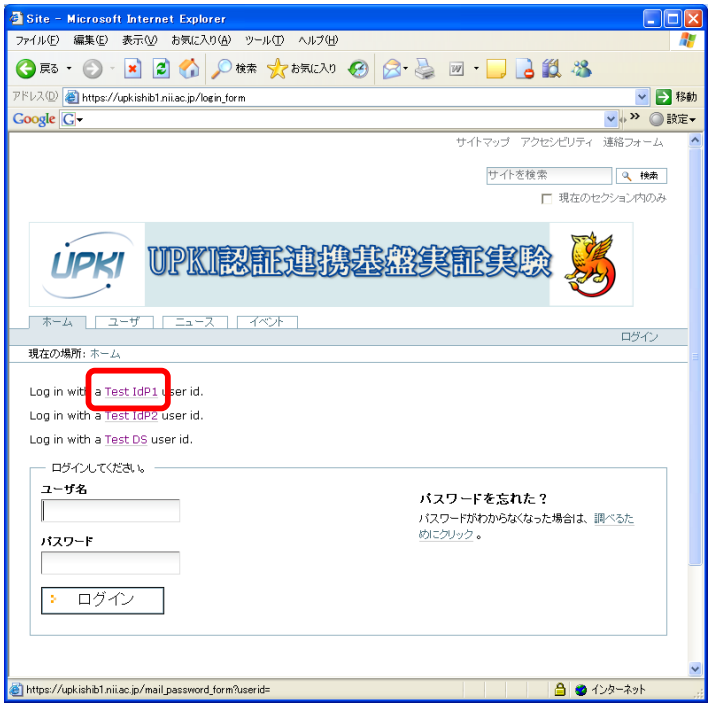
■同様に、接続する IdP のサーバ証明書が、SP 側の/etc/shibboleth/ partner-metadata .xml に登録されているか確認してください。

［5］接続確認

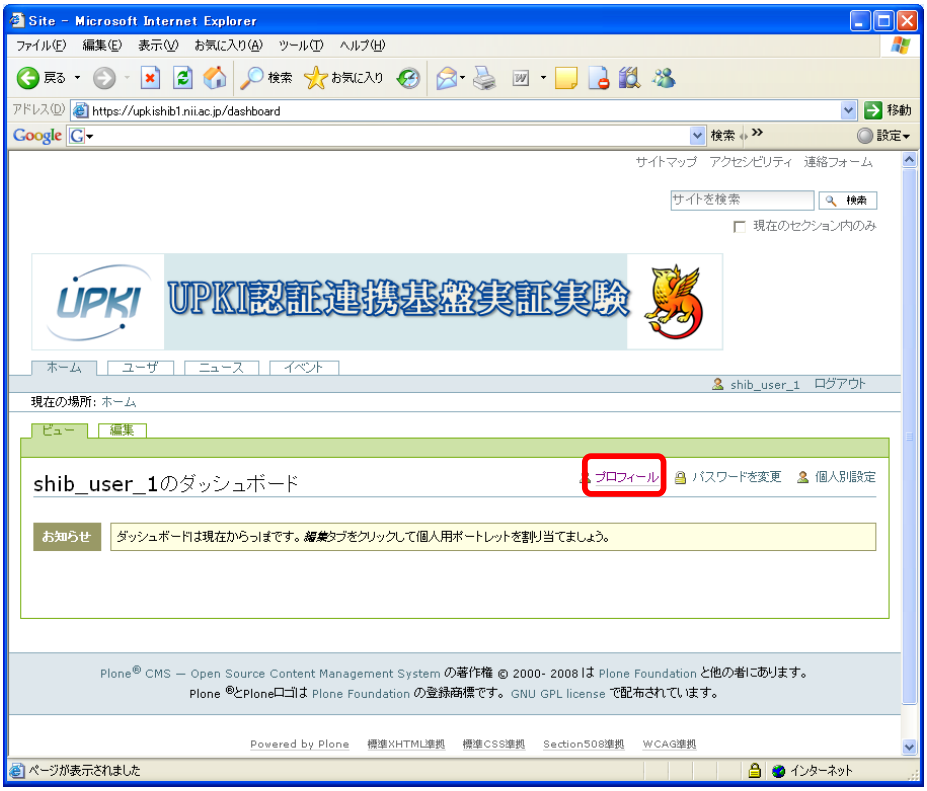
①Plone サイトにアクセスする

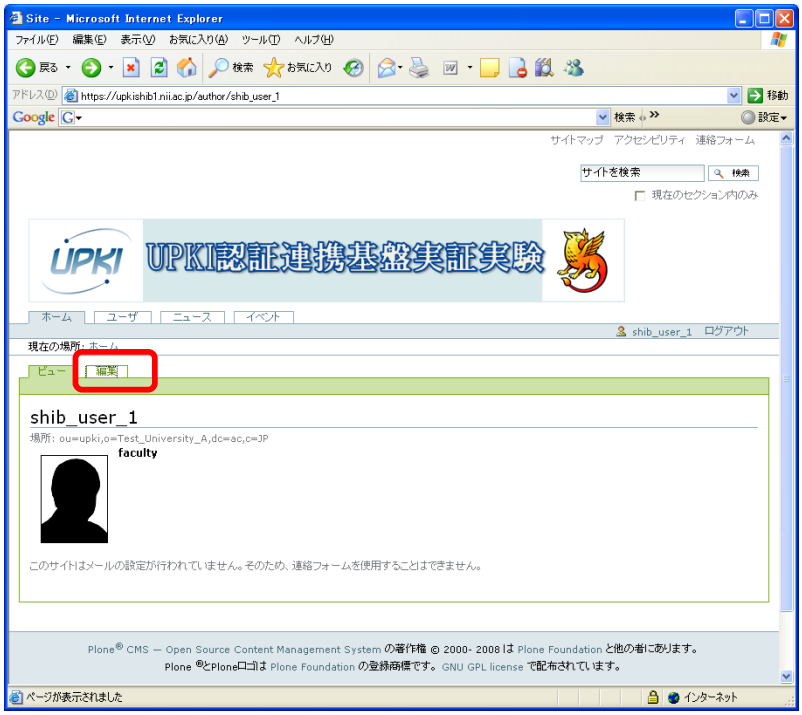
https://（貴学の SP サイト）/Plone

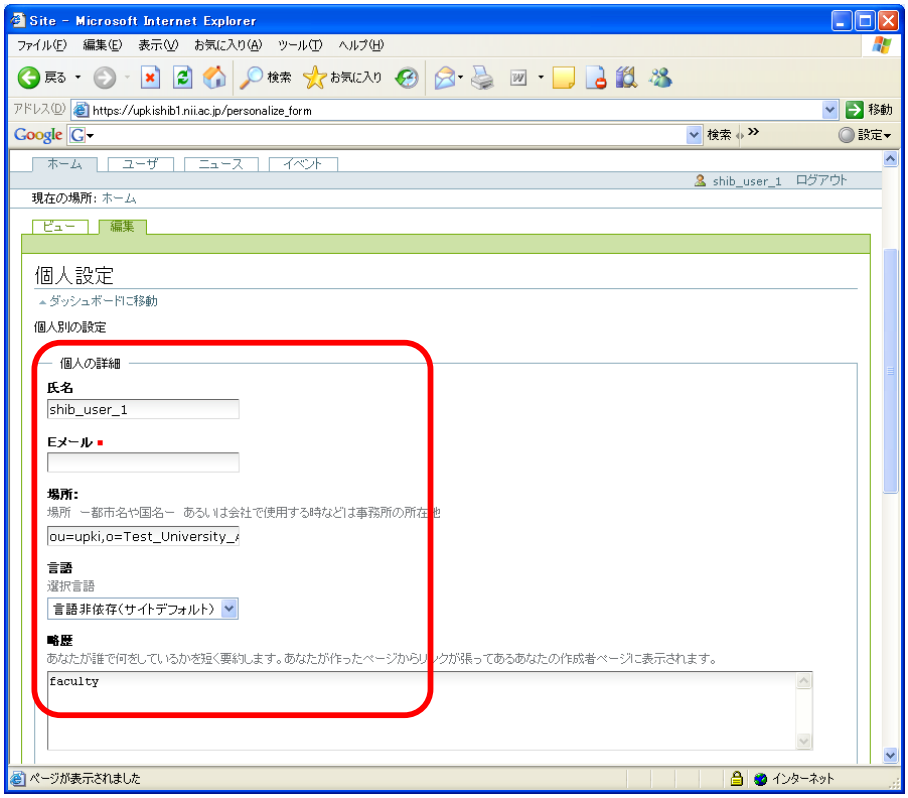
②ログインする



③ログインユーザの属性情報を確認する




Site - Microsoft Internet Explorer
https://upkishib1.nii.ac.jp/author/shib_user_1
UPKI認証連携基盤実証実験
shib_user_1
faculty
このサイトはメールの設定が行われていません。そのため、連絡フォームを使用することはできません。


Site - Microsoft Internet Explorer
https://upkishib1.nii.ac.jp/personalize_form
現在の場所: ホーム
個人設定
ダッシュボードに移動
個人別の設定
個人の詳細
氏名
shib_user_1
Eメール
場所:
言語
選択言語
言語非依存(サイトデフォルト)
略歴
faculty

４．構築後の基本的な設定、および運用方法

［１］metadata の管理方法

メタデータは新規 IdP、新規 SP の追加や、既存 IdP、既存 SP の証明書の更新等により、常に更新されます。そのため、定期的にリポジトリから最新の共通メタデータをダウンロードして IdP のメタデータを更新してください。
また、証明書の更新等、SP のメタデータに更新があった場合は、すみやかにヘルプデスクに送付してください。

［２］IdP-アプリケーション間で受け渡す属性の追加方法

/etc/shibboleth/attribute-map.xml 内に、該当する属性があるか確認してください。
※ほとんどの属性が attribute-map.xml にて定義されています。
attribute-map.xml で定義されている属性は、IdP がリリースすると、無変換でアプリケーションに送られます。

attribute-map.xml で定義されていない場合については、以下に「displayName」属性をマッピングする例で示します。

①スキーマの確認

・LDAP サーバ上の/etc/openldap/schema 配下にスキーマファイルがあります。
・「displayName」属性は、/etc/openldap/schema/inetorgperson.schema にて以下のように定義されています。

```
(中略)
# displayName
# When displaying an entry, especially within a one-line summary list, it # is useful to be able to identify a name to be used.  Since other attri- # bute types such as 'cn' are multivalued, an additional attribute type is # needed.  Display name is defined for this purpose.
attributetype ( 2.16.840.1.113730.3.1.241
        NAME 'displayName'
        DESC 'RFC2798: preferred name to be used when displaying entries'
        EQUALITY caseIgnoreMatch
        SUBSTR caseIgnoreSubstringsMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
        SINGLE-VALUE )
(中略)
```

②/etc/shibboleth/attribute-map.xml への登録

```
<Attributes xmlns="urn:mace:shibboleth:2.0:attribute-map"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
(中略)
      <Attribute name="urn:mace:dir:attribute-def:displayName" id="displayName"/>
    <Attribute name="urn:oid:2.16.840.1.113730.3.1.241" id="displayName"/>  ←①のoid
(中略)
</Attributes>
```

５．シングルサインオン実証実験用 サーバ証明書の取得方法について

実証実験で構築する IdP（認証サーバ）には，サーバ証明書の導入が必須となります。
また，本実証実験においては，UPKI「サーバ証明書発行・導入における啓発・評価研究プロジェクト」が発行するサーバ証明書を利用しますので，以下の手順に従い，サーバ証明書の申請を行ってください。

☆【サーバ証明書発行・導入における啓発・評価研究プロジェクト 参加予定機関の方】

(1)プロジェクトへの参加

最初に，本プロジェクトへの参加申請への参加申請が必要となります。詳細については，次のページをご覧ください。

〇サーバ証明書発行・導入における啓発・評価研究プロジェクト概要・参加要領等
https://upki-portal.nii.ac.jp/cerpj

(2)サーバ証明書の発行

プロジェクト参加完了後にサーバ証明書の発行を行います。次の手続きに従って，サーバ証明書の発行を申請してください。なお，CSR 作成にあたっては，次頁の「CSR プロファイル」を適用してください。

〇新規サーバ証明書発行手続き
https://upki-portal.nii.ac.jp/cerpj/request_new

(3)サーバ証明書インストール

次の手順に従って，サーバ証明書を IdP にインストールしてください。

〇サーバ証明書のインストール方法
https://upki-portal.nii.ac.jp/cerpj/niiodcamanual-v1-0.pdf

☆【サーバ証明書発行・導入における啓発・評価研究プロジェクト 参加機関の方】

次の手順に従って，新規に証明書の発行手続きおよびインストールを行ってください。

(1)新規サーバ証明書発行手続き
https://upki-portal.nii.ac.jp/cerpj/request_new
※なお，CSR 作成にあたっては，次頁の「CSR プロファイル」を適用してください。

(2)サーバ証明書インストール方法
https://upki-portal.nii.ac.jp/cerpj/niiodcamanual-v1-0.pdf

【CSR プロファイル】

本実証実験で使用するサーバ証明書の CSR は以下の形式で作成してください。

基本領域		設定内容	補
Version		Version 1(0)	-
Subject	Country	C=JP（固定値）	1
	Locality	L=Academe（固定値）	1
	Organization	O="主体者組織名" ＊ 機関毎に任意に指定 例) o= National Institute of Informatics	1
	Organizational Unit	OU="主体者組織単位名" ＊ 証明書毎に任意に指定 例) ou= NII Open Domain CA	1
	commonName	CN="サーバ FQDN" ＊証明書毎に任意に指定 例) cn=www.nii.ac.jp	1
SubjectPublicKeyInfo		主体者の公開鍵　1024 ビット以上 （ただし、例外を認める）	2
attrobites		原則 Null 値とする （ただし、例外を認める）	3
SignatureAlgorithm		SHA1 with RSAEncryption	

1. 上記指定以外の属性を利用する必要がある場合には事前相談すること。少なくとも ST（state or province name）属性は使用しないこと。また、例えば加入者メールアドレスなど本プロジェクトの確認項目対象外の情報を含めないこと。
2. RSA1024bit 以上とする。鍵長 1024bit 未満の場合には事前に登録局へ相談すること。
3. 任意の属性を含めても構わないが、必ずしも証明書に反映されるわけではない。また、含めた属性によっては受理不能とし、当該属性を除いて証明書発行要求の再生成を登録局から求める場合がある。少なくとも SubjectAltName.rfc822Name 属性は使用しないこと。

６．関連 URL

■ UPKI プロジェクト（UPKI イニシアティブ）
https://upki-portal.nii.ac.jp/

■ UPKI 認証連携基盤によるシングルサインオン実証実験
https://upki-portal.nii.ac.jp/SSO

■ UPKI 認証連携基盤リポジトリ
https://upki-portal.nii.ac.jp/SSO/Repository

■ Shibboleth プロジェクト
http://shibboleth.internet2.edu/

■ Shibboleth2.0 Wiki (Shibboleth2.0 の構築、設定に関する公式サイト)
https://spaces.internet2.edu/display/SHIB2/Home

■ Switch.aai （スイスのフェデレーション）
http://www.switch.ch/aai/

■ InCommon（米国のフェデレーション）
http://www.incommonfederation.org/