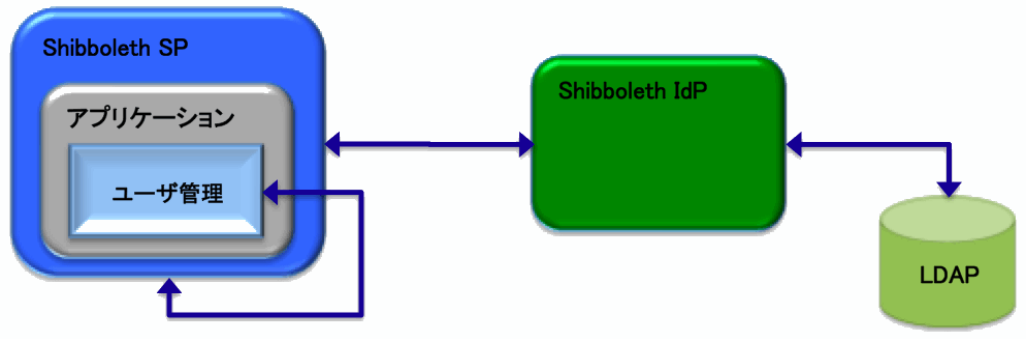
Shibbolethのセッションをもとにユーザ管理を行う

Shibbolethのセッションをもとにユーザ管理を行う

ユーザ管理機能を新たに構築し、アプリケーションへのアクセスをユーザの持つ権限により制御したい場合のパターンです。

概要図



ユーザ管理には、Shibbolethのセッション情報をもとにアクセスされたアプリケーションに対してユーザがアクセス権限を有しているかをチェックする機能が必要となります。ユーザ認証用のログイン画面は、Shibbolethが提供している認証画面を利用します。

Shibboleth SP側の設定

Apacheの設定ファイルhttpd.conf、.htaccessあるいは、shib.conf（rpmでインストールした場合のみ）の何れかにLocationを追加することで行います。

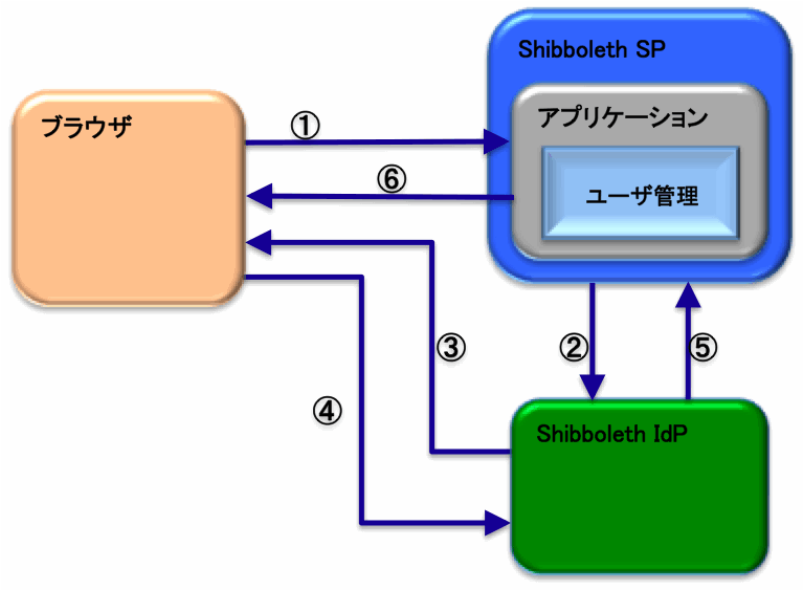
※/etc/shibboleth/shibboleth2.xmlファイルのRequestMapper要素にtype="Native"が設定されている場合に有効です。

設定例）　「App」をShibboleth化するための設定例

```
<Location /App>
  AuthType shibboleth
  ShibCompatWith24 On
  ShibRequestSetting requireSession true
  Require shib-session
</Location>
```

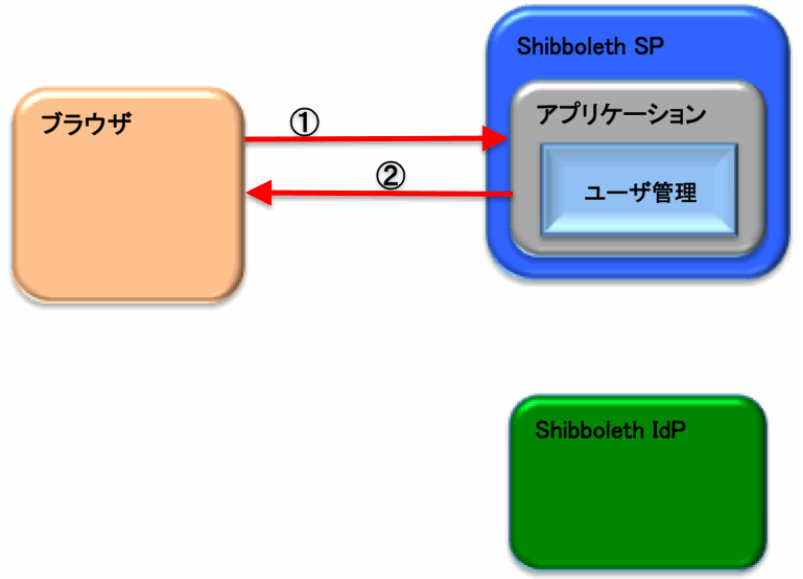
この設定により、App下の全リソースは、Shibbolethにより認証されます。

処理イメージ図：初回起動時



①ブラウザからアプリケーションのURLをアクセスします。
②Shibbolethのセッション情報がないため、Shibboleth SPからShibboleth IdPの認証画面にリダイレクトされます。
③ブラウザに認証画面を表示します。
④認証画面にユーザ／パスワードを入力し、Shibboleth IdPで認証を行います。
⑤認証結果をShibboleth SPに返します。
⑥認証が成功した場合は、ユーザ管理機能でアクセスされたアプリケーションに対するユーザのアクセス権限
をチェックします。
権限がある場合、アプリケーションを実行し、結果をブラウザに返します。ブラウザには、Shibbolethのセッション情報
を含むcookieが返されます。
権限がない場合、エラー画面を表示します。
認証が失敗した場合は、認証失敗画面を表示します。

処理イメージ図：SSO認証セッションが存在する場合



①ブラウザからアプリケーションのURLをアクセスします。
②既にShibboleth認証されているユーザからのアクセスであるため、ユーザ認証は行わずにアプリケーションに対する
　アクセス権限をチェックします。
　権限がある場合、アプリケーションを実行し、結果をブラウザに返します。
　権限がない場合、エラー画面を表示します。

サンプルコード

サンプルコードとして、シボレス認証＋アプリケーション独自のユーザー管理をする場合の対応例を示します。

アプリケーションのリソースをSPの保護下に置きます。
アプリケーションはシボレス認証情報から、アプリケーション独自のユーザー情報を取得します。
ユーザー情報のロールを検証することにより、アクセスできるコンテンツを制御します。

例では、Ruby on Rails を利用したアプリケーションを想定します。

ユーザーの権限確認は before_filter として実現し、ApplicationController のメソッドとして実装します。
ここでは、二種類の権限レベルを想定し、それらは eduPersonEntitlement 属性によって識別できるものとします。

```
class ApplicationController < ActionController::Base
  # 一般ユーザー権限の検証
  def require_member
    # Shibbolethの認証状態の確認
    #  このアクションが実行された時点で、Shibbolethのセッションは確立されています。
    #  必要に応じてSPの返却したパラメータを参照して権限の確認等を行います。
    #  SPの返却したパラメータは、HTTP環境変数に追加されています。
    unless /nii-member/ =~ request.env['entitlement']
      # eduPersonEntitlement に既定の値が無ければ認証エラー
      render :file => 'error/auth_error', :use_full_path => true, :status => 403
    end
  end

  # 管理者ユーザー権限の検証
  def require_admin
    # Shibbolethの認証状態の確認
    #  このアクションが実行された時点で、Shibbolethのセッションは確立されています。
    #  必要に応じてSPの返却したパラメータを参照して権限の確認等を行います。
    #  SPの返却したパラメータは、HTTP環境変数に追加されています。
    unless /nii-admin/ =~ request.env['entitlement']
      # eduPersonEntitlement に既定の値が無ければ認証エラー
      render :file => 'error/auth_error', :use_full_path => true, :status => 403
    end
  end
end
```

認証エラー画面のviewファイルは app/views/error/auth_error.html.erb に配置してください。
各コントローラでは、必要に応じて before_filter を設定します。

一般ユーザ権限でアクセスを許すコンテンツでは、require_memberメソッドを利用します。

```
class CommonPageController < ApplicationController
  before_filter :require_member

  # 以下はアクションの定義
  def some_action

  end
end
```

管理者ユーザ権限を要求するページでは、require_adminメソッドを利用します。

```
class SystemPageController < ApplicationController
  before_filter :require_admin

  # 以下はアクションの定義
  def some_action

  end
end
```

また、eduPersonEntitlement 属性を送信してこないユーザーに対してもアクセスを許可するコンテンツを用意する場合は、これらの before_filter を設定しません。

```
class OpenAccessController < ApplicationController
  def some_action

  end
end
```

Apache設定ファイルの編集（httpd.confに設定する場合）

/etc/httpd/conf.d/httpd.conf に下記コードを追加します。

```
<Location /App>
  AuthType shibboleth
  ShibCompatWith24 On
  ShibRequestSetting requireSession true
  Require shib-session
</Location>
```