Universitätsbibliothek Freiburg

Anwendungen schützen mit Shibboleth 2

7. Shibboleth-Workshop

Karlsruhe, 13. November 2008

Bernd Oberknapp
Universitätsbibliothek Freiburg
E-Mail: bo@ub.uni-freiburg.de

Übersicht

Universitätsbibliothek Freiburg

- Unterstützte Plattformen
- Komponenten des Shibboleth SP
- Access Control
- SessionInitiator
- WAYFs und Discovery Services
- Beispiel: ReDI
- Attribute
- Logout, Virtual Hosts, Clustering
- Umstellung von Anwendungen

Unterstützte Plattformen

Universitätsbibliothek Freiburg

- Shibboleth ermöglich den Schutz von Anwendungen mit dem Shibboleth Service Provider (SP).
- Der SP ist in C++ implementiert, aktuell ist Version 2.1.
- Folgende Webserver werden unterstützt:
  - Apache (mod_shib, shibd)
  - IIS (ISAPI-Filter, shibd)
  - weitere Webserver (z.B. lighttpd) über FastCGI
- Die Installation kann erfolgen über:
  - Binärpakete (Red Hat 4/5, Windows, Solaris 8, MacOS X)
  - SRPMS (z.B. für openSUSE zu empfehlen)
  - Sources
- Achtung: Die Pakete für Debian 5.0 sind nicht aktuell!

Andere SAML2-Implementierungen

- Es gibt keinen Shibboleth Java- oder .NET-SP.
- Java-Anwendungen können damit nur über einen (vorgeschalteten) Apache/IIS geschützt werden.
- Alternativ könnte eine andere SAML2-Implementierung verwendet werden:
  - OIOSAML.JAVA
  - OIOSAML.NET
  - simpleSAMLphp
- Im Prinzip sollte jede SAML2-Implementierung mit Shibboleth 2 kompatibel sein…

Komponenten des SP

Universitätsbibliothek Freiburg

SessionInitiator	LogoutInitiator
Services: • AssertionConsumer • ArtifactResolution • SingleLogout • ManageNameID	Handler: • Metadata • Status • Session
Attribute • Resolver • Mapper • Filter	Access Control: • XML • Apache • (Lazy Session)

Access Control

- Der SP stellt die Informationen über den Nutzer (Attribute), die der IdP übermittelt hat, als Umgebungsvariablen (optional als HTTP-Header) zur Verfügung.
- Es gibt drei Möglichkeiten, Inhalte mit dem SP zu schützen:
  - XML Access Control (Apache und IIS)
  - Apache Access Control
  - Zugriffskontrolle durch die Anwendung (Lazy Session)
- Empfehlung:
  - XML und Apache Access Control nicht mischen!
  - Wenn möglich die Apache Access Control verwenden.

XML Access Control

Universitätsbibliothek Freiburg

- Die Konfiguration erfolgt über shibboleth2.xml:
  - <RequestMapper type="XML">
  - entsprechende <Host>- und <Path>-Einträge
  - inline oder in externe XML-Datei ausgelagert
- Die XML Access Control ermöglicht die Definition praktisch beliebig komplexer Regeln.
- Einfaches Beispiel:

```
<RequestMap applicationId="default">
  <Host name="www.example.org" authType="shibboleth"
        requireSession="true">
    <Path name="restricted">
      <AccessControl>
        <Rule require="affiliation">member@example.org</Rule>
      </AccessControl>
    </Path>
  </Host>
</RequestMap>
```

Apache Access Control

Universitätsbibliothek Freiburg

- Die Konfiguration erfolgt hauptsächlich über Apache httpd.conf und .htaccess.
- Eine AND- und OR-Verknüpfung von Regeln und eine Kombination mit IP-Kontrolle ist möglich.
- Beispiel:
  - Basiskonfiguration in shibboleth2.xml:

```
<RequestMapper type="Native">
  <RequestMap applicationId="default">
    <Host name="www.example.org" />
  </RequestMap>
</RequestMapper>
```

  - Access Control-Konfiguration in httpd.conf:

```
<Location /restricted>
  AuthType shibboleth
  ShibRequireSession On
  Require affiliation member@example.org
</Location>
```

Beispiele Apache Access Control

Universitätsbibliothek Freiburg

- Viele Anwendungen können sehr einfach z.B. von Apache Basic Auth auf Shibboleth umgestellt werden.
- In der UB Freiburg schützen wir damit u.a. den Zugang zu internen Anwendungen wie Nagios, BackupPC und WebSVN.
- Auch die internen Webseiten der UB Freiburg sind durch Shibboleth geschützt, Zugriff haben nur Mitarbeiter der UB mit Kostenstelle UFR-003000:

```
<Location /intern>
  AuthType shibboleth
  ShibRequireSessionWith mylogin
  ShibRequireAll On
  Require affiliation member@uni-freiburg.de
  Require deptnum UFR-003000
</Location>
```

Lazy Session

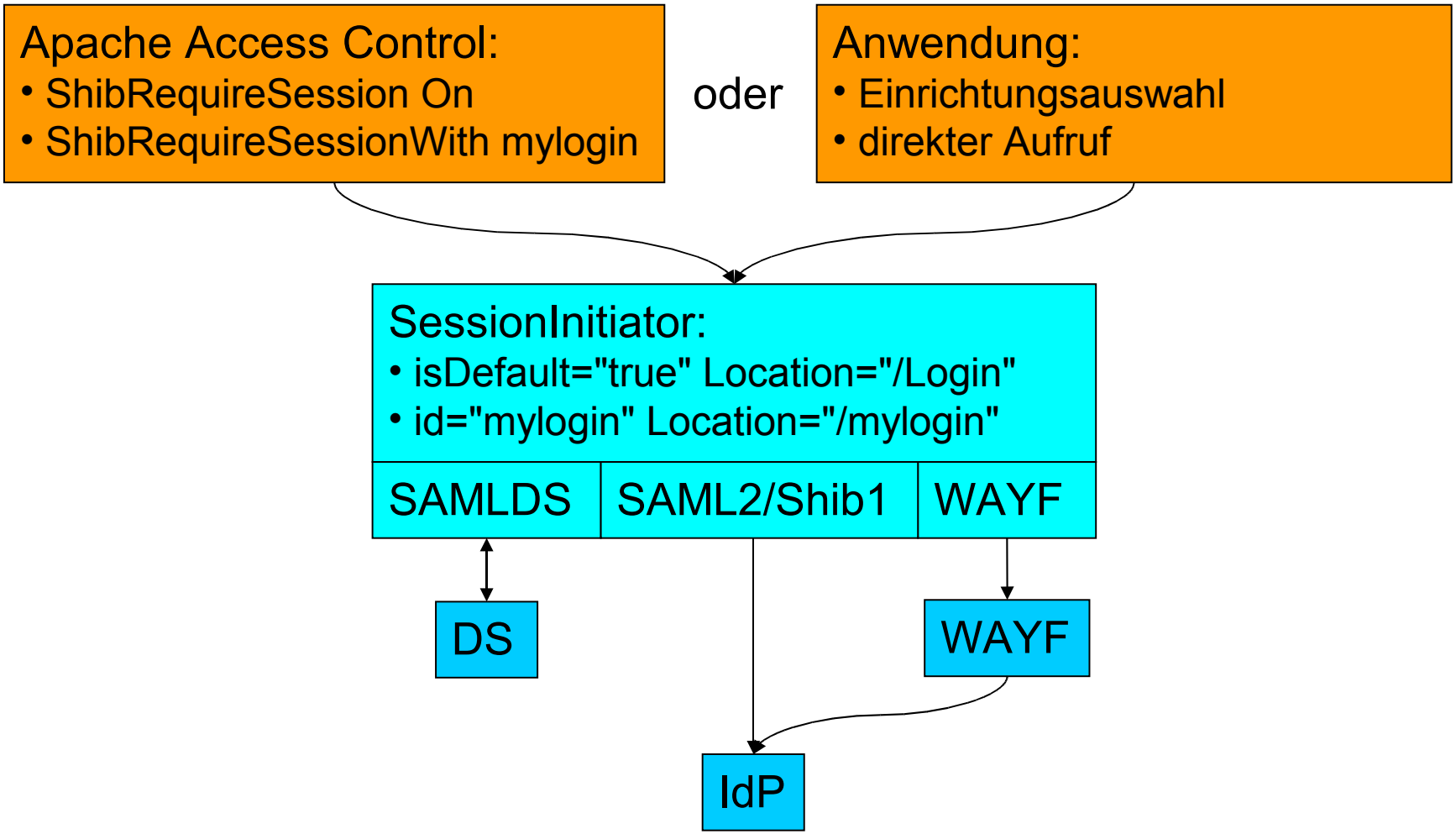
- Bei Lazy Session erfolgt die Zugriffskontrolle durch die Anwendung selbst.
- Der SP muss in diesem Fall nur angewiesen werden, die Attribute zur Verfügung zu stellen:

```
<Location /lazy>
  AuthType shibboleth
  Require shibboleth
</Location>
```

- **Achtung:** Beim Zugriff auf /lazy
  - wird **keine Authentifizierung ausgelöst** und
  - es werden **keine Attribute geprüft**!

  Das muss die Anwendung selbst übernehmen.

Aufbau einer Shibboleth-Session



SessionInitiator

Universitätsbibliothek Freiburg

- Ein SessionInitiator
  - initiiert beim Aufruf eine Shibboleth SP-Session
  - wird von der Apache/XML Access Control oder von einer Anwendung aufgerufen
  - leitet den Nutzer entweder zu einem DS/WAYF oder direkt zu einem Identity Provider (IdP) weiter
  - kann über seine ID oder URL angesprochen werden.
- Bei Shibboleth 2 sind die SessionInitiator deutlich komplexer als bei Shibboleth 1.3, da mehr Fälle abgedeckt werden müssen.
- Ein Shibboleth 2 SessionInitiator besteht üblicherweise aus einer Kette mehrerer SessionInitiator mit verschiedenen Typen.

Beispiel SessionInitiator

Universitätsbibliothek Freiburg

- Der SessionInitiator leitet den Nutzer per Default zum DFN-AAI-Test DS/WAYF weiter, wobei dieser als SAML2 Discovery Service (DS) angesprochen wird:

```
<SessionInitiator type="Chaining"
    Location="/DS" isDefault="true"
    id="DS" relayState="cookie">
  <SessionInitiator type="SAML2"
      defaultACSIndex="1" acsByIndex="false"
      template="bindingTemplate.html" />
  <SessionInitiator type="Shib1"
      defaultACSIndex="5" />
  <SessionInitiator type="SAMLDS"
      URL="https://wayf.aai.dfn.de/DFN-AAI-Test/wayf" />
</SessionInitiator>
```

Beispiel SessionInitiator

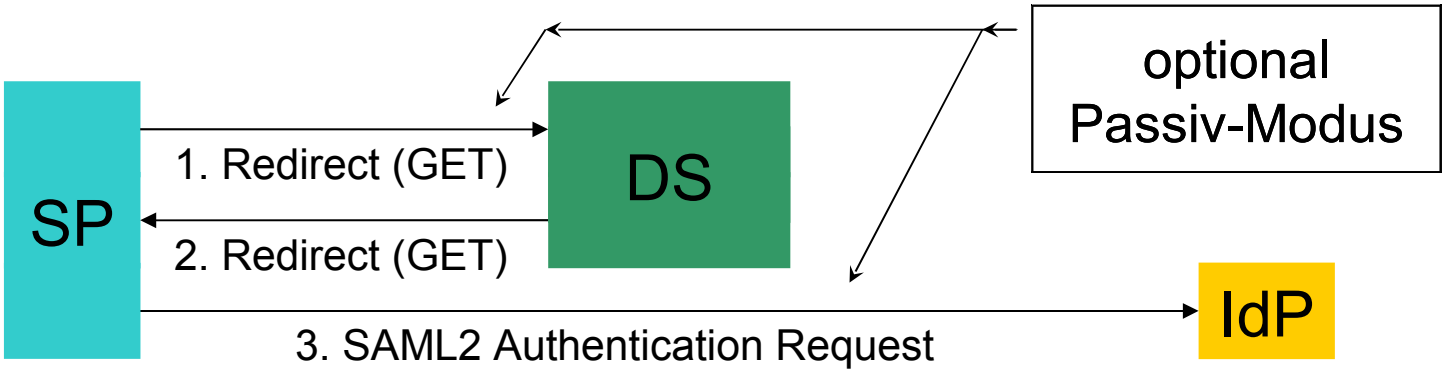
Universitätsbibliothek Freiburg

- Wird der SessionInitiator (von einer Anwendung) mit Parameter providerId=<Entity-ID eines bekannten IdP> aufgerufen, so wird der Nutzer zum IdP weitergeleitet.
- Dabei wird vorzugsweise ein SAML2-Profil verwendet (weil der Typ SAML2 vor Shib1 angegeben ist).
- defaultACSIndex verweist auf den vom SP bevorzugten SAML2 bzw. Shibboleth 1.x AssertionConsumerService.
- Wird keine bekannte Entity-ID übergeben, wird der Nutzer zum DFN-AAI-Test DS/WAYF weitergeleitet.
- Soll ein Shibboleth 1.x WAYF statt eines DS verwendet werden, muss als Typ WAYF und zusätzlich ein defaultACSIndex angegeben werden, der auf einen Shibboleth 1.x AssertionConsumerService verweist.

Shib1 WAYF und SAML2 DS

- Bei einem Shibboleth 1.x WAYF wird der Nutzer vom SP über den WAYF zum IdP geleitet:

- Bei einem SAML2 DS hat der SP mehr Kontrolle über den IdP Discovery Prozess – der SP erfährt vom DS, über welchen IdP die Authentifizierung erfolgen soll:



Zentraler DS/WAYF

Universitätsbibliothek Freiburg

- Wenn eine Anwendung von mehreren Einrichtungen genutzt wird, müssen die Nutzer Ihre Einrichtung irgendwie auswählen können.
- Wenn nur Einrichtungen aus einer Föderation die Anwendung nutzen, kann der zentrale DS/WAYF der Föderation verwendet werden.
- Vorteile:
  - sehr einfach, kein Implementierungsaufwand
  - Der Nutzer muss die Einrichtung bei einem zentralen DS/WAYF nur einmal (für mehrere Anwendungen) auswählen.
- Nachteil:
  - Ein zentraler DS/WAYF bietet meist auch Einrichtungen zur Auswahl an, die die Anwendung nicht nutzen dürfen.

Eigene Einrichtungsauswahl

Universitätsbibliothek Freiburg

- Eine eigene Einrichtungsauswahl ist zu empfehlen
  - wenn Einrichtungen aus mehreren Föderationen die Anwendung nutzen
  - oder nur Einrichtungen zur Auswahl angeboten werden sollen, die die Anwendung tatsächlich nutzen können.
- Der Internet2 (Java) oder SWITCH (PHP) DS/WAYF könnte als Komponente verwendet werden.
- Eine Einrichtungsauswahl selbst zu implementieren ist aber meistens einfacher:
  - Der Nutzer muss seine Einrichtung auswählen können.
  - Die Entity-ID der Einrichtung muss bestimmt werden.
  - Der SessionInitiator des SP muss mit providerId=<Entity-ID> und target=<Rückkehr-URL> aufgerufen werden.
- Dies wird häufig mit Lazy Session kombiniert.
- Beispiel

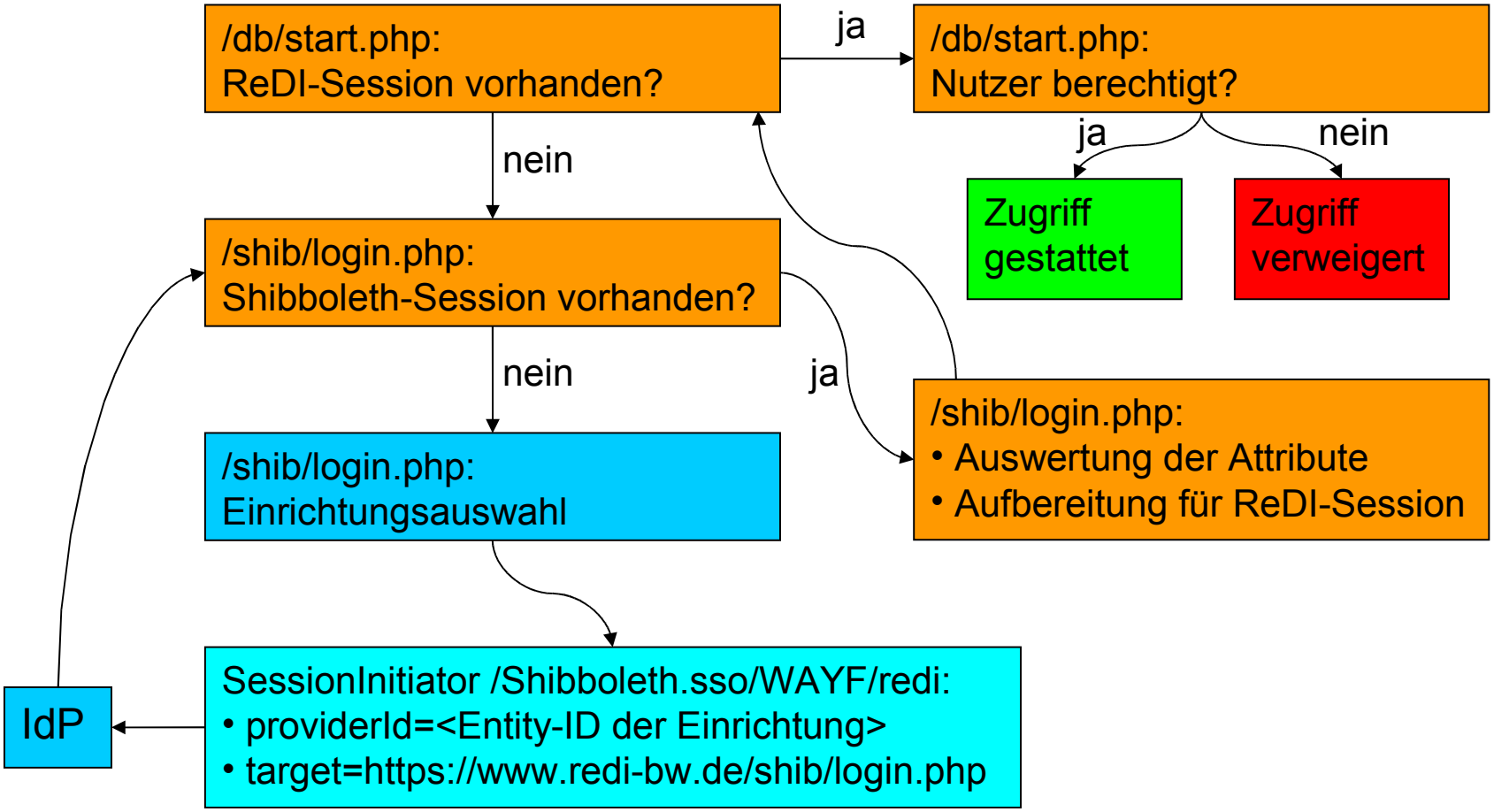
Beispiel ReDI

Universitätsbibliothek Freiburg

- In der Regionalen Datenbank-Information Baden-Württemberg (ReDI) sind mehr als 600 lizenzpflichtige Datenbanken eingebunden.
- Der Aufruf von Datenbanken erfolgt über /db/start.php.
- Das Skript /shib/login.php ist zuständig für
  - die Einrichtungsauswahl,
  - die Auswertung der vom IdP gelieferten Attribute und
  - die Bereitstellung der Informationen in der ReDI-Session.
- Für /shib wird Lazy Session verwendet:

```
<Location /shib>
  AuthType shibboleth
  Require shibboleth
</Location>
```

Beispiel ReDI



Attribute

Universitätsbibliothek Freiburg

- Die Attribut-Verarbeitung erfolgt im Shibboleth SP 2 in mehreren Stufen:
  - AttributeResolver: Fragt Attribute per SOAP-Request beim IdP ab, wenn diese nicht per Attribute Push geliefert wurden.
  - AttributeExtractor (attribute-map.xml): Legt fest, wie die gelieferten Attribute dekodiert und auf Umgebungsvariablen (oder HTTP-Header) abgebildet werden. REMOTE_USER ist ein Sonderfall, die Konfiguration erfolgt in shibboleth2.xml.
  - AttributeFilter (attribute-policy.xml): Legt fest, welche Attribute und Attributwerte von welchen IdPs akzeptiert werden.
- Shibboleth 2 bietet deutlich mehr Möglichkeiten als Shibboleth 1, die Konfiguration ist aber auch entsprechend komplexer.

Beispiel Attribute

- Das Attribut eduPersonEntitlement mit dem Wert urn:mace:dir:entitlement:common-lib-terms soll nur vom IdP der Universität Freiburg (Entity-ID https://mylogin.uni-freiburg.de/shibboleth) akzeptiert und als Umgebungsvariable entitlement bereitgestellt werden:
- attribute-map.xml:

```
<Attributes xmlns="urn:mace:shibboleth:2.0:attribute-map"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Attribute name="urn:mace:dir:attribute-def:eduPersonEntitlement"
      id="entitlement" />
  <Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7"
      id="entitlement" />
</Attributes>
```

Beispiel Attribute

Universitätsbibliothek Freiburg

- attribute-filter.xml:

```
<afp:AttributeFilterPolicyGroup
    xmlns="urn:mace:shibboleth:2.0:afp:mf:basic"
    xmlns:afp="urn:mace:shibboleth:2.0:afp"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <afp:AttributeFilterPolicy>
  <afp:PolicyRequirementRule xsi:type="AttributeIssuerString"
      value="https://mylogin.uni-freiburg.de/shibboleth" />
  <afp:AttributeRule attributeID="entitlement">
   <afp:PermitValueRule xsi:type="AttributeValueString"
       value="urn:mace:dir:entitlement:common-lib-terms" />
  </afp:AttributeRule>
 </afp:AttributeFilterPolicy>
</afp:AttributeFilterPolicyGroup>
```

(Kein) Single Logout

Universitätsbibliothek Freiburg

- Der Shibboleth SP 2 unterstützt Single Logout (SLO), der Shibboleth IdP 2 bisher aber noch nicht!
- SLO könnte momentan also nur mit einem nicht Shibboleth IdP implementiert werden.
- Warum SLO so schwierig umzusetzen ist, ist im Shibboleth Wiki beschrieben (siehe SLOIssues).
- SLO soll voraussichtlich im Shibboleth IdP 2.2 (teilweise) implementiert werden (siehe RoadMap).
- Anwendungen mit eigener Session-Verwaltung müssen für SLO angepasst werden – die Anwendungssession muss auch beendet werden!
- Noch kann die Empfehlung nur lauten: "Zum Beenden der Sitzung den Browser schließen".

Virtual Hosts, Clustering

Universitätsbibliothek Freiburg

- Mehrere Anwendungen auf einem Server sollten durch IP basierte Virtual Hosts getrennt werden, wenn sie unterschiedliche (Attribut-) Anforderungen haben.

  Pro Virtual Host sind dafür eine entityID und ein Metadaten-Eintrag sowie je ein RequestMap- und ApplicationOverride-Eintrag notwendig. Der SP sollte für alle Virtual Hosts dasselbe Zertifikat verwenden.
- Der SP 2 ist über den ODBC-StorageService Cluster fähig. Wenn die Anwendung schon Custer fähig ist und eine eigene Session-Verwaltung besitzt, ist eine Clusterung des SP normalerweise nicht notwendig – es genügt meistens, eine Anwendungssession über Shibboleth aufzubauen.

Umstellung von Anwendungen

Universitätsbibliothek Freiburg

- Für die Umstellung von Anwendungen auf Shibboleth gibt es kein „Kochrezept“!
- Folgende Punkte sind u.a. für die Umstellung relevant:
  - Wie werden die Ressourcen bisher geschützt (Apache, Tomcat, eigenes Verfahren, ...)?
  - Existiert ein eigenes Session-Management?
  - Kann dieses weiter verwendet werden, z.B. indem eine Sitzung über Shibboleth aufgebaut wird?
  - Existiert eine eigene Rechteverwaltung?
  - Können die dafür notwendigen Informationen per Shibboleth über Attribute bereitgestellt werden?
  - Können die Identity-Provider die Attribute liefern?

Vielen Dank für Ihre Aufmerksamkeit!

Fragen?

DFN-AAI-Webseite: https://www.aai.dfn.de
DFN-AAI Hotline: hotline@aai.dfn.de
AAI-Users Mailingliste: aai-users@aai.dfn.de