Anwendungen schützen mit dem Shibboleth Service Provider

23. Mai 2012, TU Kaiserslautern, RHRK

UNI FREIBURG

Albert-Ludwigs-Universität Freiburg

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

Übersicht

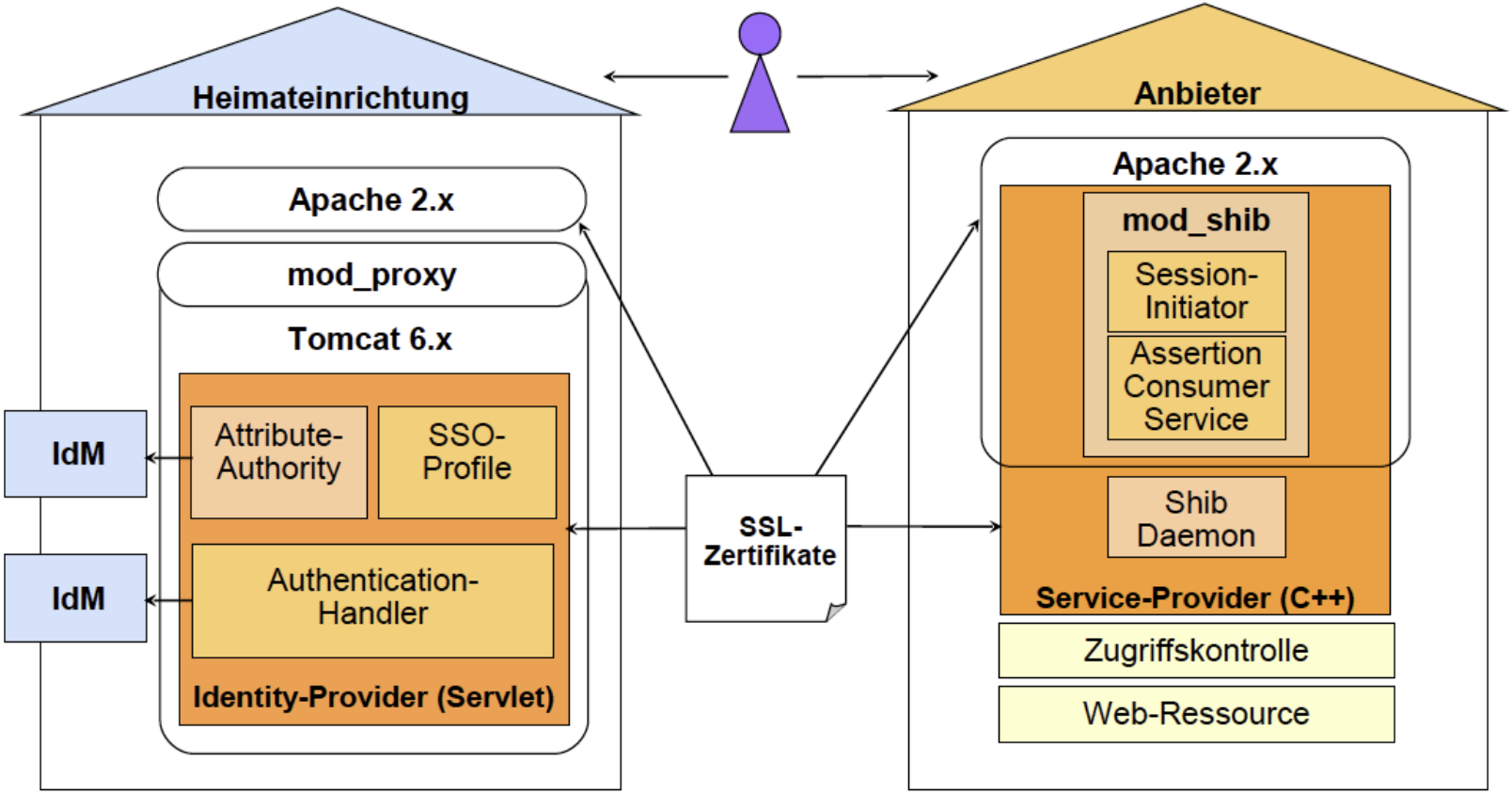
UNI FREIBURG

- Architektur des Shibboleth SPs
- Installation, Konfiguration und Test
- DFN-AAI-Metadatenverwaltung
- Discovery, Embedded Discovery Service
- Anwendungen schützen

UNI FREIBURG

Architektur des Shibboleth SPs

Shibboleth Architektur



Unterstützte Plattformen

UNI FREIBURG

- Shibboleth ermöglicht den Schutz von Anwendungen mit dem Shibboleth Service Provider (SP).
- Der SP ist in C++ implementiert.
- Der SP besteht aus einem Modul bzw. Filter und Extension für den Webserver und einem Daemon.
- Unterstützte Webserver:
  - Apache (2.2, Unterstützung für 2.4 ist für SP 2.5 geplant, 1.3 und 2.0 nur selbst kompiliert)
  - Internet Information Server (IIS)
  - FastCGI mit Authenticator-Support, z.B. lighttpd

Andere SAML2-Implementierungen

UNI FREIBURG

- Es gibt keinen Shibboleth Java- oder .NET-SP.
- Java-Anwendungen können damit nur über einen vorgeschalteten Webserver 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…

„API“ des Shibboleth SPs

UNI FREIBURG

- Der Shibboleth SP wird über URLs angesprochen:
  - Initiator: Session (SSO, Login) und Logout
  - Handler: Metadata, Status, Session und DiscoFeed zum Abruf bestimmter Informationen
  - Service: Endpunkte für die Kommunikation mit IdPs (oder „enhanced Clients“)
- Der Aufruf der URLs bzw. der entsprechenden Funktionen kann entweder direkt erfolgen oder durch Access Control Regeln ausgelöst werden.
- Informationen zum Status und zum Nutzer stellt der SP per Environment-Variablen oder HTTP-Header zur Verfügung.

UNI FREIBURG

Installation, Konfiguration und Test

Installation des Shibboleth SP

- Offizielle Binärpakete werden bereitgestellt (RPMs über den openSUSE Build Service) für
  - SLES/openSUSE
  - Red Hat/CentOS
  - OS X
  - Windows
- Andere Binärpakete sollten nur verwendet werden, wenn diese aus einer vertrauenswürdigen Quelle stammen und aktuell sind! Im Zweifelsfall ist es besser, die Sources selbst zu kompilieren.

Beispiel: Installation unter SLES

UNI FREIBURG

- Im Folgenden wird ein SLES11 SP2-Server als Beispiel verwendet. Es wird vorausgesetzt, dass Apache (Port 80 und 443) und andere Komponenten bereits installiert und grundsätzlich konfiguriert sind.
- Für die Installation des SP muss nur das offizielle Repository eingebunden und das Paket installiert werden:

```
# zypper ar -f http://download.opensuse.org/repositories/
security:/shibboleth/SLE_11_SP2/security:shibboleth.repo
# zypper in shibboleth
```

Konfiguration des Shibboleth SPs

UNI FREIBURG

- Die Konfiguration erfolgt über die Dateien unter `/etc/shibboleth`, die Einbindung in den Apache über `/etc/apache2/conf.d/shib.conf`.
- Konfigurationsdateien:
  - Hauptkonfigurationsdatei: `shibboleth2.xml`
  - Attribute (Mapping, Filter): `attribute-map.xml`, `attribute-policy.xml`
  - Logger: `*.logger`
  - Templates (für Produktion anpassen!): `*.html`
- Es gibt eine Vielzahl von Konfigurationsoptionen, die `shibboleth2.xml` ist aber (seit Version 2.4) trotzdem sehr übersichtlich.

Konfiguration des Shibboleth SPs

UNI FREIBURG

- Für einen funktionsfähigen Shibboleth SP sind nur wenige Anpassungen notwendig:
  - `entityID`: weltweit eindeutige Kennung des SP
  - `SSO` (SessionInitiator): legt fest, wie die Weiterleitung der Nutzer zum IdP erfolgt
  - `supportContact`: E-Mailadresse für Fehlermeldungen
  - `MetadataProvider`: Einbinden von Metadaten mit allen notwendigen Angaben für die Kommunikation mit IdPs
  - `CredentialResolver`: Zertifikat und Key
- Im Folgenden werden diese sowie einige nicht notwendige, aber empfehlenswerte Anpassungen anhand eines Beispiels veranschaulicht.

Konfiguration: entityID, SSO

UNI FREIBURG

- Die `entityID` muss für die DFN-AAI eine URL sein, sie sollte den Host- bzw. Dienstnamen und einem Pfad wie `/shibboleth` enthalten:

```
<ApplicationDefaults
  entityID="https://spt.ub.uni-freiburg.de/shibboleth"
  REMOTE_USER="eppn persistent-id targeted-id">
```

- Der SessionInitiator verweist im einfachsten Fall auf einen bestimmten IdP

```
<SSO entityID="https://testidp2.aai.dfn.de/idp/shibboleth">
    SAML2 SAML1
</SSO>
```

oder einen vorhandenen Discovery Service

```
<SSO discoveryProtocol="SAMLDS"
  discoveryURL="https://wayf.aai.dfn.de/DFN-AAI-Test/wayf">
    SAML2 SAML1
</SSO>
```

Konfiguration: Handler, Contact

UNI FREIBURG

- Zum Debuggen oder Monitoring muss der Status-Handler für die Rechner freigeschaltet werden, die darauf zugreifen können sollen:

```
<Handler type="Status" Location="/Status"
  acl="127.0.0.1 132.230.25.9 132.230.25.229"/>
```

- Der Session-Handler kann nicht nur die Namen der übermittelten Attribute anzeigen, sondern auch die Werte, was für die Fehlersuche sehr hilfreich ist:

```
<Handler type="Session" Location="/Session"
  showAttributeValues="true"/>
```

- `supportContact` muss geändert werden, sonst gehen ggf. E-Mails an den Betreiber des Mailservers:

```
<Errors supportContact="bo@ub.uni-freiburg.de" ...
```

Konfiguration: Metadaten

UNI FREIBURG

- Die notwendigen Informationen über die IdPs, mit denen der SP kommunizieren soll, müssen über `MetadataProvider` zur Verfügung gestellt werden.
- Für den Test wird die DFN-AAI-Test verwendet:

```
<MetadataProvider type="XML" uri="http://www.aai.dfn.de/
fileadmin/metadata/DFN-AAI-Test-metadata.xml"
  backingFilePath="DFN-AAI-Test-metadata.xml"
  reloadInterval="7200">
    <MetadataFilter type="RequireValidUntil"
      maxValidityInterval="2419200"/>
    <MetadataFilter type="Signature"
      certificate="dfn-aai.pem"/>
</MetadataProvider>
```

Konfiguration: Metadaten

UNI FREIBURG

- Die Metadaten sind die (technische) Basis des Vertrauens zwischen SP und IdP, daher muss die Authentizität und Integrität unbedingt sichergestellt werden. Dies erfolgt durch Prüfen der Signatur, das dafür notwendige Zertifikat wird von der DFN-AAI-Webseite heruntergeladen:

```
# cd /etc/shibboleth
# wget https://www.aai.dfn.de/fileadmin/metadata/dfn-aai.pem
```

- Es können beliebig viele `MetadataProvider` angegeben werden. Wie die Konfiguration aussehen muss ist jeweils auf der Webseite der Föderation, die die Metadaten bereitstellt, dokumentiert.

Konfiguration: Zertifikat und Key

UNI FREIBURG

- Für einen Test können beliebige Zertifikate verwendet werden, für den Produktionsbetrieb in der DFN-AAI sollten Zertifikate der DFN-PKI verwendet werden.
- Im Beispiel wird für den SP dasselbe unter `/etc/ssl` installierte Zertifikat wie für den Apache verwendet:

```
<CredentialResolver type="File"
  key="/etc/ssl/private/spt.ub.uni-freiburg.de.key"
  certificate="/etc/ssl/certs/spt.ub.uni-freiburg.de.crt"/>
```

- Die Basiskonfiguration ist damit abgeschlossen, die Konfiguration kann getestet und der SP gestartet werden:

```
# shibd -t
# rcshibd start
# rcapache2 restart
```

Test des Shibboleth SPs

- Zum Testen eines SPs können die Handler und der SessionInitiator direkt aufgerufen werden:
  - `https://spt.ub.uni-freiburg.de/Shibboleth.sso/Session`

    liefert Informationen zur Session oder „A valid session was not found“ falls noch keine Session besteht.
  - `https://spt.ub.uni-freiburg.de/Shibboleth.sso/Status`

    liefert Informationen zum Status des SPs – wenn die ACL des Handlers den Zugriff gestattet.
  - `https://spt.ub.uni-freiburg.de/Shibboleth.sso/Metadata`

    liefert die Metadaten des SP (in einfachen Fällen).
  - `https://spt.ub.uni-freiburg.de/Shibboleth.sso/Login`

    versucht eine Session zu starten – was scheitert, weil der IdP/Discovery Service den SP noch nicht kennt...

UNI FREIBURG

DFN-AAI-Metadatenverwaltung

Metadaten des SPs registrieren

- Da der SP mit IdPs aus der DFN-AAI-Testumgebung getestet werden soll, muss der SP für die DFN-AAI-Test registriert werden:
  - DFN-AAI-Metadatenverwaltung aufrufen
  - neuen SP anlegen
  - Metadaten über den Metadata-Handler abrufen (Zugriff muss in der Firewall freigeschaltet sein)
  - Ergebnis prüfen, fehlende Angaben ergänzen (eine ausführliche Beschreibung der Felder gibt es in der integrierten Hilfe)
  - unter Föderationen DFN-AAI-Test wählen
  - Angaben speichern

Metadaten des SPs registrieren

UNI FREIBURG

- Nicht jeder, der Shibboleth testen möchte oder einen SP betreibt, bekommt einen eigenen Account für die DFN-AAI-Metadatenverwaltung.
- Üblicherweise gibt es pro Einrichtung einen lokalen Administrator, der für die Verwaltung der Metadaten seiner Einrichtung zuständig ist und über den die Registrierung erfolgen muss.
- Die DFN-AAI-Metadatendateien werden jeweils zur vollen Stunde neu generiert, und danach müssen die IdPs noch ihr lokale Kopie aktualisieren bevor ein erneuter Test durchgeführt werden kann!

Auswahl der Föderation

- In der DFN-AAI-Metadatenverwaltung werden bei SPs folgende Föderationen zur Auswahl angeboten:
  - DFN-AAI-Test: Testumgebung, eine Nutzung dieser Föderation für den Produktionsbetrieb ist nicht zulässig!
  - DFN-AAI: IdPs müssen Verlässlichkeitsklasse Advanced haben (üblich bei lizenzierten Inhalten)
  - DFN-AAI-Basic + DFN-AAI: IdPs müssen entweder Verlässlichkeitsklasse Basic oder Advanced haben
  - lokale Metadaten: SP wird nur lokal, d.h. nur mit dem IdP der eigenen Einrichtung, genutzt
- Zukünftig wird es möglich sein, einen SP/IdP über eduGAIN auch in anderen Ländern in Europa bekannt zu machen (opt-in, momentan nur auf Anfrage).

UNI FREIBURG

Discovery, Embedded Discovery Service

Discovery

- Wie kommt der Nutzer vom SP zu seinem IdP?
- Wenn nur ein IdP genutzt wird, kann dieser einfach fest im `SessionInitiator` eingetragen werden.
- Wenn mehrere IdPs genutzt werden sollen, könnte eventuell ein zentraler Discovery Service (DS) der Föderation verwendet werden.
- Nachteil: Es werden alle IdPs der Föderation zur Auswahl angeboten, nicht nur diejenigen, mit denen der SP genutzt werden kann (lässt sich eventuell durch Einbetten des DS vermeiden).
- Empfehlung: Es sollte ein eigener DS bzw. eine eigene Einrichtungsauswahl verwendet werden (siehe auch die NISO EsPRESSO-Empfehlungen).

Einfache Einrichtungsauswahl

UNI FREIBURG

- Eine einfache Einrichtungsauswahl genügt im Prinzip, es ist nicht unbedingt eine Komponente notwendig, die das IdP Discovery Service Protocol beherrscht.
- Der Nutzer muss lediglich seine Einrichtung und damit die `entityID` eines IdP auswählen können, z.B. über eine Dropdown-Liste. Beispiel: ReDI
- Die `entityID` wird mit an den `SessionInitiator` übergeben, um den Nutzer direkt zum ausgewählten IdP weiterzuleiten. Beispiel:

```
https://spt.ub.uni-freiburg.de/Shibboleth.sso/Login?
entityID=https%3A%2F%2Fmylogintest.uni-freiburg.de%2Fshibboleth&
target=https%3A%2F%2Fspt.ub.uni-freiburg.de%2FShibboleth.sso%2FSession
```

- Nach dem Login wird der Nutzer zur `target` URL (im Beispiel der Session-Handler) weitergeleitet.

Embedded Discovery Service

UNI FREIBURG

- Der Embedded Discovery Service (EDS) ist eine elegante Möglichkeit, eine Einrichtungsauswahl in eigene Anwendungen zu integrieren.
- Der EDS ist komplett in JavaScript implementiert.
- Der EDS kann entweder über das DS Protokoll oder (bei entsprechender Konfiguration) auch direkt angesprochen werden.
- Die notwendigen Daten liefert der SP basierend auf den eingebundenen Metadaten im JSON-Format über den DiscoFeed-Handler. Beispiel:

```
https://spt.ub.uni-freiburg.de/Shibboleth.sso/DiscoFeed
```

Embedded Discovery Service

UNI FREIBURG

- Der EDS kann aus demselben Repository wie der SP installiert werden:

```
# zypper in shibboleth-embedded-ds
```

- Die Konfiguration erfolgt über `/etc/shibboleth-ds/idpselect_config.js` und den `MetadataProvider` in `shibboleth2.xml`, die Einbindung in den Apache über `/etc/apache2/conf.d/shibboleth-ds.conf`.
- Zur Einbettung des EDS in eine Webseite müssen lediglich eine CSS- und zwei JavaScript-Dateien eingebunden und ein `div` hinzugefügt werden, siehe das Beispiel `/etc/shibboleth-ds/index.html`.

Embedded Discovery Service

UNI FREIBURG

- Achtung: In der Beispiel `index.html` muss bei den URLs der Pfad `/shibboleth-ds/` ergänzt werden!
- Für einen ersten Test kann die (korrigierte) Beispiel `index.html` z.B. als `ds.html` in das DocumentRoot-Verzeichnis des Apache kopiert werden
- Die entsprechende URL wird in `shibboleth2.xml` beim `SessionInitiator` eingetragen:

```
<SSO discoveryProtocol="SAMLDS"
  discoveryURL="https://spt.ub.uni-freiburg.de/ds.html">
    SAML2 SAML1
</SSO>
```

- Zum Testen wird der `SessionInitiator` aufgerufen:

```
https://spt.ub.uni-freiburg.de/Shibboleth.sso/Login
```

Embedded Discovery Service

UNI FREIBURG

- Damit nicht nur die `entityIDs` angezeigt werden, muss bei Metadaten wie den DFN-AAI-Metadaten, die die MDUI-Extension noch nicht unterstützen, der `MetadataProvider` wie folgt ergänzt werden:

```
<MetadataProvider type="XML" legacyOrgNames="true" ...
```

- Über `MetadataFilter` kann die Auswahl auf bestimmte IdPs eingeschränkt werden (`Whitelist`), oder es können bestimmte IdPs ausgeschlossen werden (`Blacklist`). Beispiel:

```
<MetadataFilter type="Whitelist">
    <Include>https://mylogintest.uni-
freiburg.de/shibboleth</Include>
    <Include>https://idp.dfn.de/idp/shibboleth</Include>
</MetadataFilter>
```

UNI FREIBURG

Anwendungen schützen

Apache und XML Access Control

UNI FREIBURG

- Die Regeln für die Zugriffskontrolle können entweder in der Apache-Konfiguration (Apache Access Control) oder in der SP-Konfiguration (XML Access Control) angegeben werden.
- Apache Access Control ermöglicht die einfache UND- oder ODER-Verknüpfung von Bedingungen.
- Mit der XML Access Control sind praktisch beliebig komplexe Boolesche Regeln möglich.
- Empfehlung: Die beiden Varianten nicht mischen, wenn möglich Apache Access Control verwenden.

Aktiver Schutz

UNI FREIBURG

- Beim aktiven Schutz sorgt der Apache bzw. der SP dafür, dass der Zugriff nur möglich ist, wenn der Nutzer eine gültige Session hat und die angegebenen Regeln erfüllt sind.
- Im einfachsten Fall dürfen alle authentifizierten Nutzer auf ein Verzeichnis zugreifen:

```
<Location /protected>
    AuthType Shibboleth
    ShibRequestSetting requireSession true
    Require valid-user
</Location>
```

- Das tritt in der Praxis eher selten auf, typischerweise werden weitere Bedingungen an die Nutzer gestellt.

Aktiver Schutz

- Beispiel: Nur Mitarbeiter der UB Freiburg dürfen auf die internen Webseiten der UB zugreifen. Außerdem ist der Zugriff von den Mitarbeiternetzen der UB aus ohne Login gestattet:

```
<Location /intern>
    AuthType shibboleth
    ShibRequestSetting requireSession true
    ShibRequireAll On
    Require affiliation employee@uni-freiburg.de
    Require departmentNumber UFR-003000

    Order deny,allow
    Deny from all
    Allow from 132.230.25 132.230.225

    Satisfy any
</Location>
```

Aktiver Schutz

UNI FREIBURG

- Der SP setzt unter anderem den `REMOTE_USER`, wenn eines der in der `shibboleth2.xml` dafür angegeben Attribute zur Verfügung steht – damit kann eine Vielzahl von Anwendungen sehr einfach von der Apache Basic Auth auf Shibboleth umgestellt werden.
- In der UB Freiburg schützen wir so u.a. den Zugang zu internen Anwendungen wie Nagios, BackupPC, Gitweb und WebSVN.
- Bei komplexeren Anwendungen mit eigenem Session-Management wird häufig statt der ganzen Anwendung nur ein Skript geschützt, das die Anwendungssession aufbaut. Beispiel: Moodle

Passiver Schutz (Lazy Session)

UNI FREIBURG

- Bei Lazy Session ist die Anwendung selbst dafür verantwortlich,
  - bei Bedarf eine Authentifizierung auszulösen (durch Redirect zum `SessionInitiator` des SP),
  - die über Environment-Variablen oder HTTP-Header bereitgestellten Attribute (in PHP z.B. in $_SERVER) auszuwerten und
  - damit die Zugriffkontrolle durchzuführen.
- Der SP stellt der Anwendung lediglich die Attribute zur Verfügung (sofern vorhanden):

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

Mehrere VirtualHosts

UNI FREIBURG

- Auch bei mehreren (IP basierten) VirtualHosts gibt es nur eine SP-Instanz auf dem Server.
- Der SP bekommt aber pro VirtualHost eine eigene `entityID` und einen eigenen Eintrag in den Metadaten, so dass eine saubere Trennung der Attributfreigabe für die beiden VirtualHosts im IdP möglich ist
- Sprich: Zwei Anwendung mit unterschiedlichen Anforderungen an die Attribute lassen sich auf einem Server nur durch VirtualHosts sauber trennen.
- Der SP sollte für alle VirtualHosts dasselbe Zertifikat verwenden.

Mehrere VirtualHosts

- Es sind nur zwei Änderungen in `shibboleth2.xml` notwendig:
  - Eigene `applicationID` für den zweiten VirtualHost:

```
<RequestMapper type="Native">
    <RequestMap applicationId="default">
        <Host name="vh1.ub.uni-freiburg.de" />
        <Host name="vh2.ub.uni-freiburg.de"
          applicationId="vh2" />
    </RequestMap>
</RequestMapper>
```

  - `ApplicationOverride` mit eigener `entityID`:

```
<ApplicationOverride id="vh2"
  entityID="https://vh2.ub.uni-freiburg.de/shibboleth"
  homeURL="https://vh2.ub.uni-freiburg.de"/>
```

Personalisierung

- Um eine Personalisierung anbieten zu können, muss die Anwendung den Nutzer wiedererkennen können.
- Shibboleth/SAML bietet dafür persistent NameIDs/ eduPersonTargetedIDs, d.h. dauerhaft eindeutige Pseudonyme für die Nutzer, die für jeden SP unterschiedlich sind.
- Damit ist ein automatisches Login in die persönliche Umgebung möglich, ohne einen „Personal Account“ registrieren/verwenden zu müssen und ohne die Identität des Nutzers offenzulegen!
- Beispiel: Katalog plus der UB Freiburg

Anwendungen umstellen

- Für das Umstellen von Anwendungen auf Shibboleth gibt es kein „Kochrezept“!
- Folgende Punkte sind u.a. für die Umstellung relevant:
  - Wie wird die Anwendung 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 IdPs die Attribute liefern?

UNI FREIBURG

Vielen Dank für Ihre Aufmerksamkeit!

Fragen?

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