deutsches forschungsnetz

Attribute

Attribut-Schemata, -Generierung, -Übertragung und Verarbeitung am SP

Shibboleth- und AAI Workshop für Helmholtz-Gemeinschaft

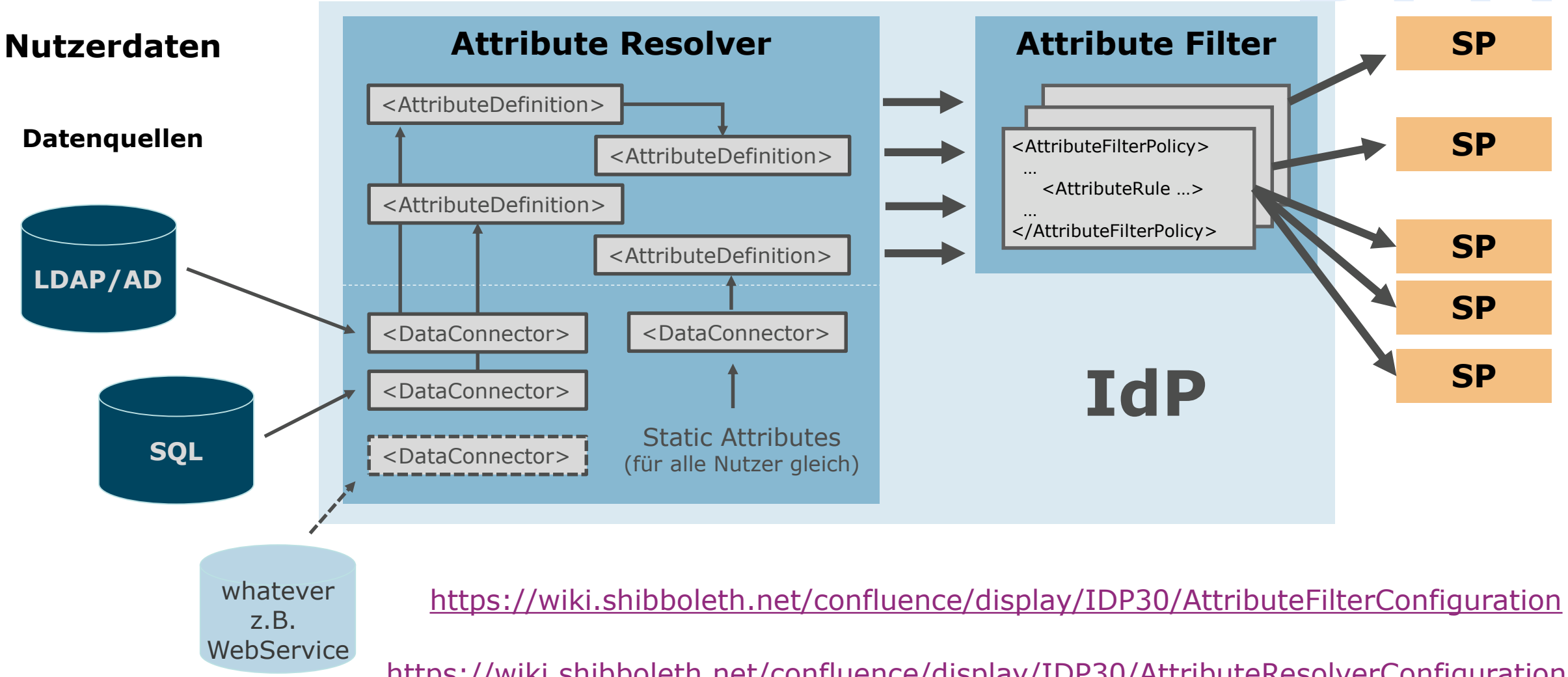
Berlin, 29. Januar 2019

Wolfgang Pempe (pempe@dfn.de)

Attribute: Generelles

- IdP stellt mithilfe von Attributen die (zur Nutzung eines Dienstes) notwendigen Informationen über den User zur Verfügung
- Attribute bilden u.a. die Grundlage für die Autorisierung (“was darf ich?”)
- Daneben gibt es Attribute, die der eindeutigen Identifizierung des Users und der Personalisierung des betreffenden Dienstes dienen (z.B. givenName, sn, eduPersonPrincipalName, mail)
- SP nimmt Attribute entgegen und stellt sie der jeweiligen Anwendung zur Verfügung
- SP und/oder Anwendung entscheiden anhand ihrer Regeln über den Zugriff.

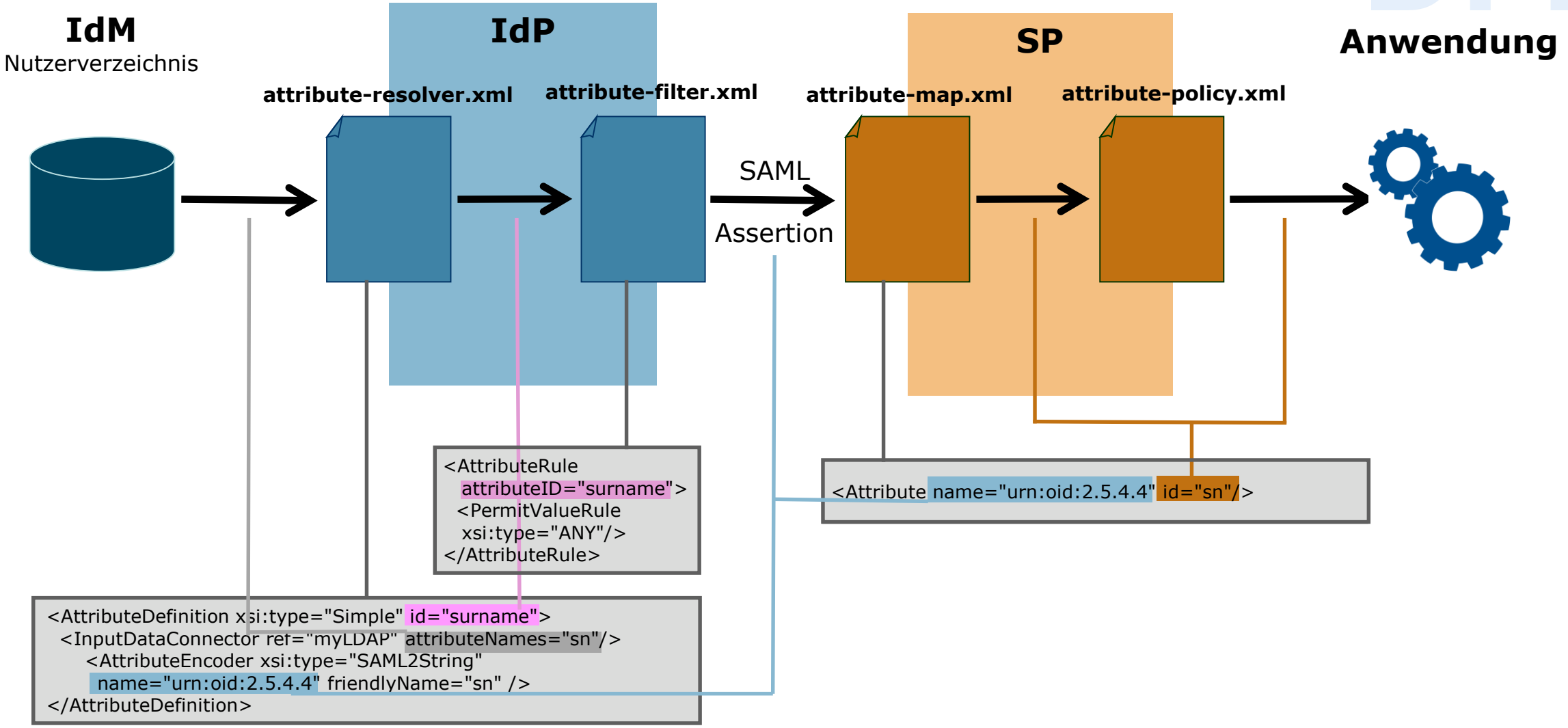
IdP – Attribute Resolver und Filter



https://wiki.shibboleth.net/confluence/display/IDP30/AttributeFilterConfiguration

https://wiki.shibboleth.net/confluence/display/IDP30/AttributeResolverConfiguration

Überblick


IdM
Nutzerverzeichnis
IdP
attribute-resolver.xml
attribute-filter.xml
SAML
Assertion
SP
attribute-map.xml
attribute-policy.xml
Anwendung
<AttributeRule
attributeID="surname">
<PermitValueRule
xsi:type="ANY"/>
</AttributeRule>
<Attribute name="urn:oid:2.5.4.4" id="sn"/>
<AttributeDefinition xsi:type="Simple" id="surname">
<InputDataConnector ref="myLDAP" attributeNames="sn"/>
<AttributeEncoder xsi:type="SAML2String"
name="urn:oid:2.5.4.4" friendlyName="sn" />
</AttributeDefinition>

Attribute: Schemata

- Schemata legen eine Menge von Attributen, die zulässigen Werte und deren Bedeutung fest.
- Im Föderationsumfeld hat sich etabliert:
  - eduPerson (international)
  - SCHAC (Schema for Academia, international)
  - dfnEduPerson (e-Learning, Deutschland)
  - Sowie weitere Attribute z.B. aus inetOrgPerson
- Doku: https://doku.tid.dfn.de/de:attributes
- Entscheidend für Kommunikation IdP ↔ SP, müssen aber nicht notwendigerweise im IdM vorhanden sein!

Attribute: Beispiele

- eduPerson:
  - eduPersonScopedAffiliation (Status innerhalb der Heimateinrichtung + Scope), z.B. `student@uni-musterstadt.de`
  - eduPersonEntitlement (Berechtigung), z.B. `urn:mace:dir:entitlement:common-lib-terms`
  - eduPersonPrincipalName (eindeutiger, nicht anonymer Username), z.B. `hugo123@uni-musterstadt.de`
  - eduPersonUniqueId (eindeutiger, anonymer bzw. pseudonymer Username) z.B. `28c5353b8bb34984a8bd4169ba94c606@uni-musterstadt.de`
- SCHAC:
  - schacPersonalUniqueCode (insbes. Matrikelnummer), z.B. `urn:schac:personalUniqueCode:de:lmu.de:Matrikelnummer:1234567`

Attribut-Handling im IdP

1. Liest alle “rohen” Attribute des Users aus dem IdM bzw. Nutzerverzeichnis (LDAP, AD, SQL) → DataConnector
2. Aus dem IdM gelesene Attribute können gesplittet, zusammengefügt und umgeschrieben werden → AttributeDefinition
3. Neue Attribute können in Abhängigkeit von anderen Attributen / Werten generiert werden → AttributeDefinition
4. Attribute Filter Policies entscheiden über die Weitergabe der Attribute / Attributwerte
5. User Consent: Zustimmung zur Attributfreigabe
6. Die transformierten Attribute werden in eine SAML-Assertion (“Zusicherung”) verpackt.
7. Assertion wird per HTTP-Post an den Assertion Consumer Service (ACS) des anfragenden SP geschickt (URL aus Föderationsmetadaten)

IdP: Attribute aus IDM auslesen

- Attribute Resolver:
  Data Connector
- Konfiguration:
  `./conf/attribute-resolver.xml`,
  Vorbelegung der Werte in
  `./conf/ldap.properties`
- Siehe auch unter
  https://doku.tid.dfn.de/de:shibidp3config-idm

```
<DataConnector id="myLDAP" xsi:type="LDAPDirectory"
    ldapURL="%{idp.attribute.resolver.LDAP.ldapURL}"
    baseDN="%{idp.attribute.resolver.LDAP.baseDN}"
    principal="%{idp.attribute.resolver.LDAP.bindDN}"
    principalCredential="%{idp.attribute.resolver.LDAP.bindDNCredential}"
    useStartTLS="%{idp.attribute.resolver.LDAP.useStartTLS:true}"
    connectTimeout="%{idp.attribute.resolver.LDAP.connectTimeout}"
    trustFile="%{idp.attribute.resolver.LDAP.trustCertificates}"
    responseTimeout="%{idp.attribute.resolver.LDAP.responseTimeout}">
    <FilterTemplate>
        <![CDATA[
            %{idp.attribute.resolver.LDAP.searchFilter}
        ]]>
    </FilterTemplate>
    <ConnectionPool
        minPoolSize="%{idp.pool.LDAP.minSize:3}"
        maxPoolSize="%{idp.pool.LDAP.maxSize:10}"
        blockWaitTime="%{idp.pool.LDAP.blockWaitTime:PT3S}"
        validatePeriodically="%{idp.pool.LDAP.validatePeriodically:true}"
        validateTimerPeriod="%{idp.pool.LDAP.validatePeriod:PT5M}"
        expirationTime="%{idp.pool.LDAP.idleTime:PT10M}"
        failFastInitialize="%{idp.pool.LDAP.failFastInitialize:false}" />
</DataConnector>
```

IdP: IDM-Attribute → SAML

- Aus dem IdM / Nutzerverzeichnis ausgelesene Attribute werden umgeschrieben auf eduPerson, SCHAC, dfnEduPerson, etc.
- Neue Attribute werden in Abhängigkeit von anderen Attributen bzw. -Werten generiert
- Mechanismus: “Attribute Definitions”
  - Simple
  - Mapped
  - ScriptedAttribute
  - u.a.m. → https://wiki.shibboleth.net/confluence/x/JgIUAQ
- Dies geschieht ebenfalls in attribute-resolver.xml

IdP: Simple Attribute Definition

```
<!--
The uid is the closest thing to a "standard" LDAP attribute
representing a local username, but you should generally *never*
expose uid to federated services, as it is rarely globally unique.
-->
<AttributeDefinition id="uid" xsi:type="Simple">
    <InputDataConnector ref="myLDAP" attributeNames="sAMAccountName"/>
    <AttributeEncoder xsi:type="SAML2String" name="urn:oid:0.9.2342.19200300.100.1.1"
                      friendlyName="uid" encodeType="false" />
</AttributeDefinition>

<!--
In the rest of the world, the email address is the standard identifier,
despite the problems with that practice. Consider making the EPPN value
the same as your official email addresses whenever possible.
-->
<AttributeDefinition id="mail" xsi:type="Simple">
    <InputDataConnector ref="myLDAP" attributeNames="mail"/>
    <AttributeEncoder xsi:type="SAML2String" name="urn:oid:0.9.2342.19200300.100.1.3"
                      friendlyName="mail" encodeType="false" />
</AttributeDefinition>
```

Shib Wiki: https://wiki.shibboleth.net/confluence/x/EgAcAQ

IdP: Mapped Attribute Definition

```
<AttributeDefinition xsi:type="Mapped" id="eduPersonAffiliation">
    <InputDataConnector ref="myLDAP" attributeNames="memberOf"/>
    <AttributeEncoder xsi:type="SAML2String" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
                      friendlyName="eduPersonAffiliation" />
    <DefaultValue>affiliate</DefaultValue>
    <ValueMap>
      <ReturnValue>staff</ReturnValue>
      <SourceValue>Mitarbeiter</SourceValue>
    </ValueMap>
    <ValueMap>
      <ReturnValue>employee</ReturnValue>
      <SourceValue>Mitarbeiter</SourceValue>
      <SourceValue>Professor</SourceValue>
      <SourceValue>Lehrbeauftragte</SourceValue>
    </ValueMap>
    <ValueMap>
      <ReturnValue>faculty</ReturnValue>
      <SourceValue>Professor</SourceValue>
      <SourceValue>Lehrbeauftragte</SourceValue>
    </ValueMap>
    <ValueMap>
      <ReturnValue>student</ReturnValue>
      <SourceValue>Studierende</SourceValue>
    </ValueMap>
    <ValueMap>
      <ReturnValue>member</ReturnValue>
      <SourceValue>Mitarbeiter</SourceValue>
      <SourceValue>Studierende</SourceValue>
      <SourceValue>Professor</SourceValue>
      <SourceValue>Lehrbeauftragte</SourceValue>
    </ValueMap>
</AttributeDefinition>
```

Shib Wiki:
https://wiki.shibboleth.net/confluence/x/HoAgAQ

IdP: Scripted Attribute Definition

```
<AttributeDefinition xsi:type="ScriptedAttribute" id="eduPersonEntitlement">
   <InputAttributeDefinition ref="eduPersonAffiliation" />
   <DisplayName xml:lang="en">Entitlement</DisplayName>
   <DisplayName xml:lang="de">Berechtigung</DisplayName>
   <AttributeEncoder xsi:type="SAML2String" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7"
                     friendlyName="eduPersonEntitlement" />
   <Script>
      <![CDATA[
           if (eduPersonAffiliation.getValues().contains("member")) {
              eduPersonEntitlement.getValues().add("urn:mace:dir:entitlement:common-lib-terms");
              eduPersonEntitlement.getValues().add("http://bwidm.de/entitlement/bwLSDF-SyncShare");
              eduPersonEntitlement.getValues().add("http://bwidm.de/entitlement/bwLSDF-FileService");
              eduPersonEntitlement.getValues().add("http://bwidm.de/entitlement/bwUniCluster");
           }
           if (eduPersonAffiliation.getValues().contains("staff")) {
              eduPersonEntitlement.getValues().add("http://bwidm.de/entitlement/bwLehrpool");
           }
      ]]>
   </Script>
</AttributeDefinition>
```

Shib Wiki: https://wiki.shibboleth.net/confluence/display/IDP30/ScriptedAttributeDefinition
Zu den Unterschieden der Scripting Engines unter Java 7 und 8 siehe auch
https://wiki.shibboleth.net/confluence/display/SHIB2/IdPJava1.8

IdP: Scoped Attribute Definition

```
<AttributeDefinition xsi:type="Scoped" id="eduPersonScopedAffiliation" scope="%{idp.scope}">
   <InputAttributeDefinition ref="eduPersonAffiliation" />
   <DisplayName xml:lang="en">Affiliation type (with institution)</DisplayName>
   <DisplayName xml:lang="de">Zugehörigkeit (+ Einrichtung)</DisplayName>
   <AttributeEncoder xsi:type="SAML2ScopedString" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.9"
                     friendlyName="eduPersonScopedAffiliation" encodeType="false" />
</AttributeDefinition>
```

- Scoped Attributes bestehen aus zwei Komponenten, die durch ‚@' getrennt sind:
  - **Value**, hier z.B. „student"
    und
  - **Scope**, der/ein Domain Name der Heimateinrichtung
- Beispiel: **`student@uni-freiburg.de`**

Shib Wiki:
https://wiki.shibboleth.net/confluence/display/IDP30/ScopedAttributeDefinition

IdP: Attribute Encoding

- Nach Umwandlung der Attribute werden diese in eine Form gebracht, die der empfangende SP versteht (Teil der AttributeDefinition):

```
<AttributeEncoder xsi:type="SAML1String" name="urn:mace:dir:attribute-def:mail"
                  encodeType="false" />
<AttributeEncoder xsi:type="SAML2String" name="urn:oid:0.9.2342.19200300.100.1.3"
                  friendlyName="mail" encodeType="false" />
```

- Wird auch in attribute-resolver.xml konfiguriert.
- Die gängigsten Attribute sind dort schon eingetragen (ggf. ent-kommentieren).
- Shibboleth Wiki: https://wiki.shibboleth.net/confluence/x/KQIUAQ

IdP: Attribut-Filter

- Legt fest, welche Attribute an einen SP oder eine Gruppe von SP versendet werden (Datenschutz!)
- Attribute werden anhand ihrer ID aus dem Attribute Resolver referenziert:
  `AttributeDefinition/@id`
- Sehr flexibel, Regeln anhand bestimmter Kriterien, z.B.:
  - Föderation
  - SP (Entity ID)
  - User
  - Attribut-Wert (oder in Abhängigkeit von anderen)
  - Entity Attribute
  - Boolesche Kombinationen daraus
- Shibboleth Wiki
  https://wiki.shibboleth.net/confluence/display/IDP30/AttributeFilterConfiguration
  https://wiki.shibboleth.net/confluence/display/IDP30/AttributeFilterPolicyConfiguration

IdP: Attribute Filter (Beispiel 1)

Konfiguration: `./conf/attribute-filter.xml`

```
<!-- Attribute an die DFN-Test-IdPs freigeben -->
<AttributeFilterPolicy id="dfn_test_sps">
     <PolicyRequirementRule xsi:type="OR">
         <Rule xsi:type="Requester" value="https://testsp.aai.dfn.de/shibboleth" />
         <Rule xsi:type="Requester" value="https://testsp2.aai.dfn.de/shibboleth" />
         <Rule xsi:type="Requester" value="https://testsp3.aai.dfn.de/shibboleth" />
     </PolicyRequirementRule>
     <AttributeRule attributeID="uid"                          permitAny="true"/>
     <AttributeRule attributeID="eduPersonPrincipalName"       permitAny="true"/>
     <AttributeRule attributeID="mail"                         permitAny="true"/>
     <AttributeRule attributeID="surname"                      permitAny="true"/>
     <AttributeRule attributeID="givenName"                    permitAny="true"/>
     <AttributeRule attributeID="o"                            permitAny="true"/>
     <AttributeRule attributeID="eduPersonScopedAffiliation"   permitAny="true"/>
     <AttributeRule attributeID="eduPersonAffiliation"         permitAny="true"/>
     <AttributeRule attributeID="eduPersonEntitlement"         permitAny="true"/>
     <AttributeRule attributeID="displayName"                  permitAny="true"/>
     <AttributeRule attributeID="schacHomeOrganization"        permitAny="true"/>
     <AttributeRule attributeID="schacHomeOrganizationType"    permitAny="true"/>
</AttributeFilterPolicy>
```

IdP: Attribute Filter (Beispiel 2)

Konfiguration: `./conf/attribute-filter.xml`

```
<AttributeFilterPolicy id="LibraryTermsToAnyone">
     <PolicyRequirementRule xsi:type="ANY" />

   <AttributeRule attributeID="eduPersonEntitlement">
     <PermitValueRule xsi:type="Value" value="urn:mace:dir:entitlement:common-lib-terms"/>
   </AttributeRule>

   <AttributeRule attributeID="eduPersonScopedAffiliation">
     <PermitValueRule xsi:type="OR">
         <Rule xsi:type="Value" value="member" />
         <Rule xsi:type="Value" value="library-walk-in" />
     </PermitValueRule>
   </AttributeRule>

</AttributeFilterPolicy>
```

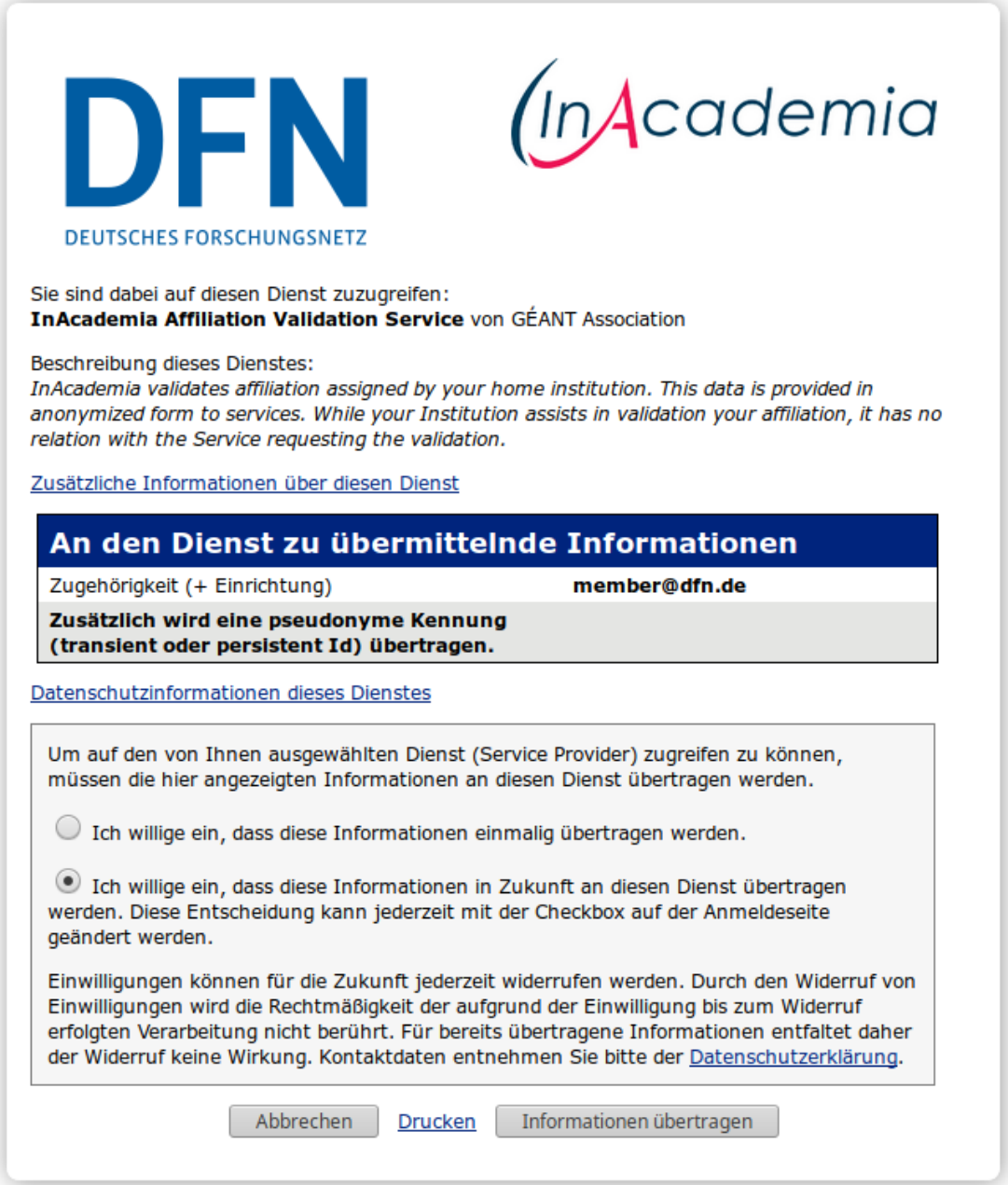
Weitere Beispiele im DFN-AAI Wiki unter https://doku.tid.dfn.de/de:shibidp3attributes

IdP:Attributfreigabe

User Consent-Modul

Hinweis:

Informationen zur Einwilligung der Endnutzer zur Attributfreigabe werden im Logfile `idp-consent-audit.log` abgelegt → gut aufheben, Nachweispflicht!



IdP: SAML Assertion → SP

Attribute werden im Attribute Statement übertragen:

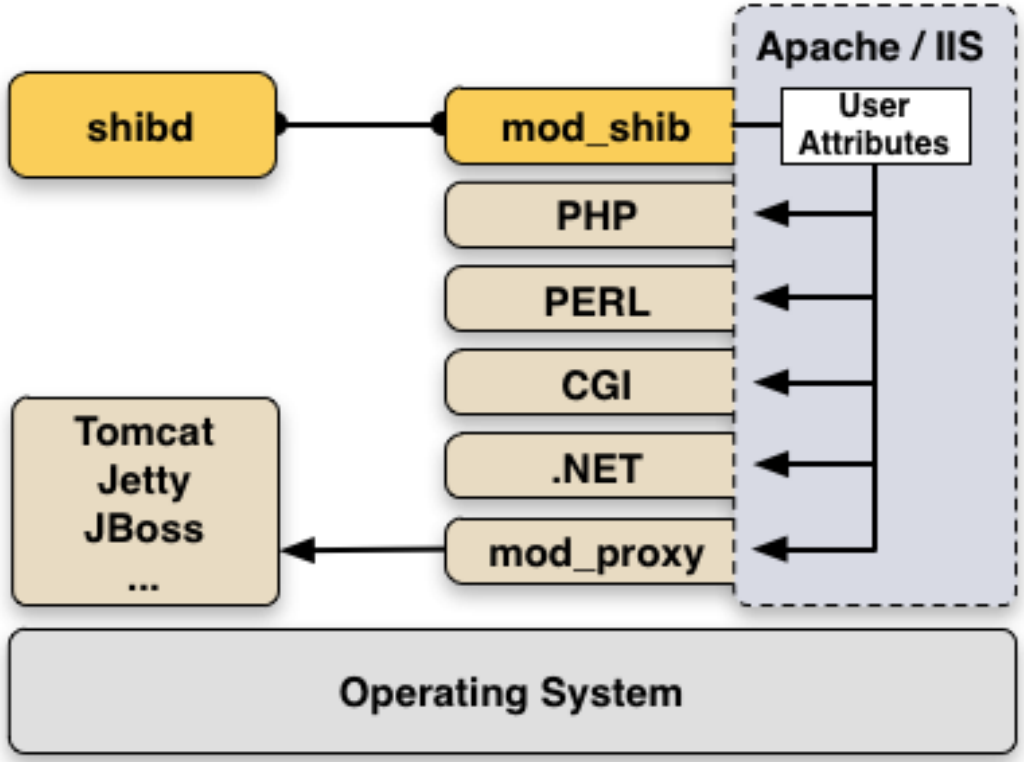
```
<saml2:AttributeStatement>
   <saml2:Attribute FriendlyName="eduPersonEntitlement"
      Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7"
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
     <saml2:AttributeValue>urn:mace:dir:entitlement:common-lib-terms</saml2:AttributeValue>
   </saml2:Attribute>
   <saml2:Attribute FriendlyName="eduPersonScopedAffiliation"
      Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.9"
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
     <saml2:AttributeValue>member@testscope.aai.dfn.de</saml2:AttributeValue>
   </saml2:Attribute>
</saml2:AttributeStatement>
```

Der empfangende SP identifiziert die Attribute anhand des URI-Wertes (meistens urn:oid) in Name

NB: `idp.loglevel.messages` und `idp.loglevel.encryption` jeweils auf `DEBUG` setzen, damit Assertions im IdP Log erscheinen.

SP: Assertion Consumer Service

- Empfängt die SAML-Assertion vom IdP
- Extrahiert die Attribute
- Identifiziert die Attribute anhand eindeutiger Kennung (URI in `saml2:Attribute/@Name`)
- Bildet Attribute auf interne Variablen ab (attribute-map.xml)
- Filtert Variablen (attribute-policy.xml)
- Exportiert Variablen per CGI-Interface



Quelle: https://www.switch.ch/aai/guides/sp/

SP: Attribute Map

- /etc/shibboleth/attribute-map.xml
- Bildet Attribute auf interne Variablen ab
- Variablenname wird in **id** definiert

```
<Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.9" id="affiliation">
    <AttributeDecoder xsi:type="ScopedAttributeDecoder" caseSensitive="false"/>
</Attribute>
<Attribute name="urn:mace:dir:attribute-def:eduPersonScopedAffiliation" id="affiliation">
    <AttributeDecoder xsi:type="ScopedAttributeDecoder" caseSensitive="false"/>
</Attribute>

<Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1" id="unscoped-affiliation">
    <AttributeDecoder xsi:type="StringAttributeDecoder" caseSensitive="false"/>
</Attribute>
<Attribute name="urn:mace:dir:attribute-def:eduPersonAffiliation" id="unscoped-affiliation">
    <AttributeDecoder xsi:type="StringAttributeDecoder" caseSensitive="false"/>
</Attribute>

<Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7" id="entitlement"/>
<Attribute name="urn:mace:dir:attribute-def:eduPersonEntitlement" id="entitlement"/>
```

SP: Attribute Policy

- /etc/shibboleth/attribute-policy.xml
- Filtert Variablen und deren Werte, i.d.R. genügen die Default-Einstellungen

```
<afp:PermitValueRule id="eduPersonAffiliationValues" xsi:type="OR">
    <Rule xsi:type="AttributeValueString" value="faculty"/>
    <Rule xsi:type="AttributeValueString" value="student"/>
    <Rule xsi:type="AttributeValueString" value="staff"/>
    <Rule xsi:type="AttributeValueString" value="alum"/>
    <Rule xsi:type="AttributeValueString" value="member"/>
    <Rule xsi:type="AttributeValueString" value="affiliate"/>
    <Rule xsi:type="AttributeValueString" value="employee"/>
    <Rule xsi:type="AttributeValueString" value="library-walk-in"/>
</afp:PermitValueRule>

<afp:PermitValueRule id="ScopingRules" xsi:type="AND">
    <Rule xsi:type="NOT">
        <Rule xsi:type="AttributeValueRegex" regex="@"/>
    </Rule>
    <Rule xsi:type="saml:AttributeScopeMatchesShibMDScope"/>
</afp:PermitValueRule>

<afp:AttributeFilterPolicy>
    <!-- This policy is in effect in all cases. -->
    <afp:PolicyRequirementRule xsi:type="ANY"/>
    <!-- Filter out undefined affiliations and ensure only one primary. -->
    <afp:AttributeRule attributeID="affiliation">
        <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>
    <!-- Catch-all that passes everything else through unmolested. -->
    <afp:AttributeRule attributeID="*" permitAny="true"/>
</afp:AttributeFilterPolicy>
```

SP: REMOTE_USER Variable

- Spezielle CGI-Variable, in der die Identität des Users transportiert wird.
- Eine oder mehrere der in attribute-map.xml definierten Variablen können verwendet werden
- wird gesetzt in shibboleth2.xml:

```
<ApplicationDefaults entityID="https://loa-check.aai.dfn.de/shibboleth"
                     REMOTE_USER="eppn persistent-id mail eduPersonUniqueId">
```

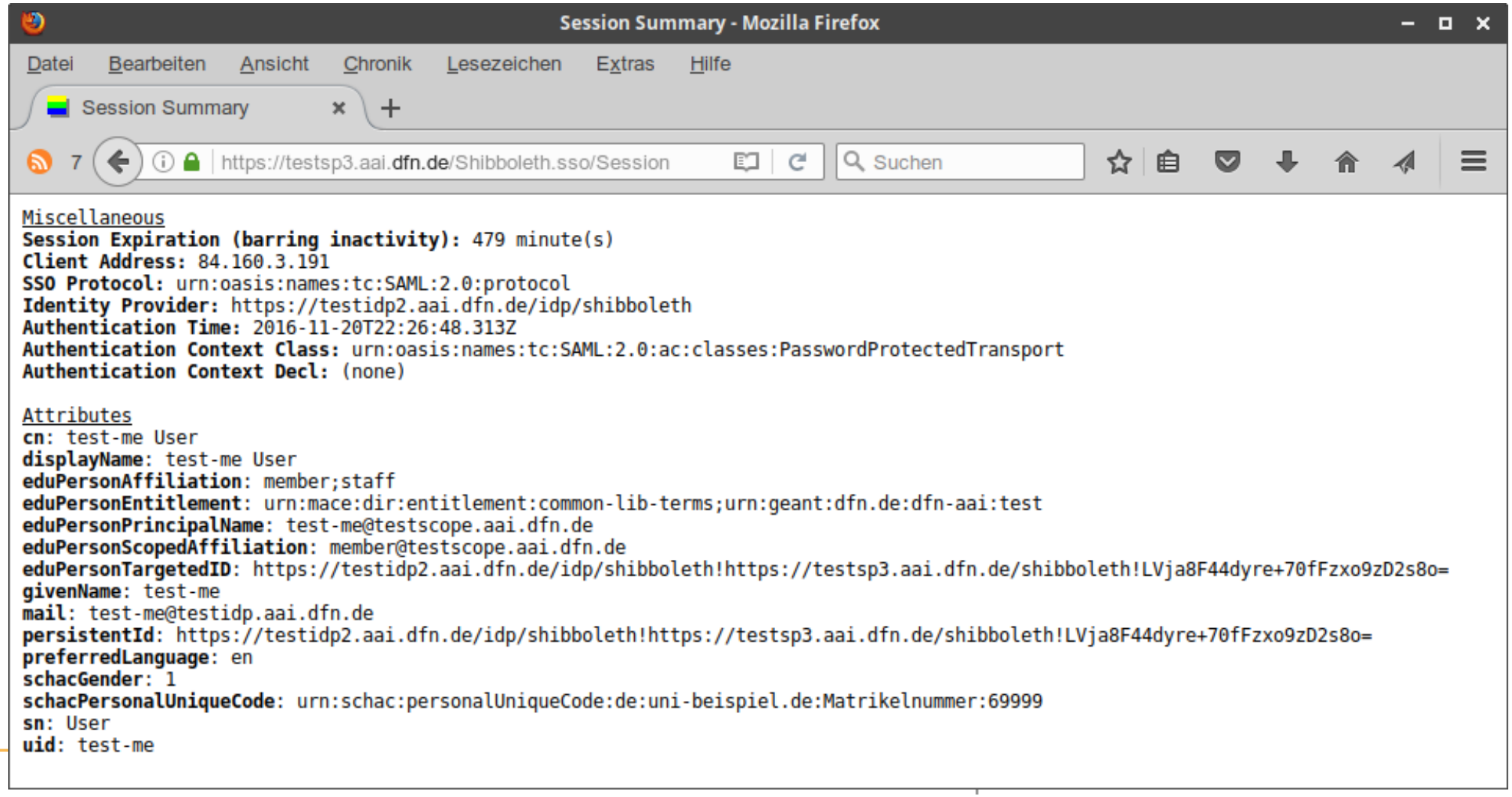
- Damit lassen sich auch Applikationen durch Shibboleth schützen, die keine direkte Shibboleth-(Attribut-)Unterstützung mitbringen → Alternative zu Basic Auth
- Siehe auch unter

  https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPApplication

  https://wiki.shibboleth.net/confluence/display/SP3/AttributeAccess#AttributeAccess-REMOTE_USER

SP: Session-Informationen

Session Handler URL Extension bei Shibboleth SP: /Shibboleth.sso/Session



Vielen Dank! Fragen? Kommentare?

- Kontakt
  - DFN-AAI Team

    E-Mail: aai@dfn.de
    Tel.: +49-30-884299-9124
    Fax: +49-30-884299-370

    Adresse:
    DFN-Verein, Geschäftsstelle
    Alexanderplatz 1
    D-10178 Berlin

Informationsquellen

- DFN-AAI Wiki:
  https://doku.tid.dfn.de/de:attributes
  https://doku.tid.dfn.de/de:shibidp3attributes
- eduPerson (u.a.m.)
  http://macedir.org/specs/eduperson/
- dfnEduPerson
  kommentierte Liste:
  https://doku.tid.dfn.de/de:elearning_attributes
- SCHAC (Schema for Academia)
  https://wiki.refeds.org/display/STAN/SCHAC+Releases
- inetOrgPerson
  https://tools.ietf.org/html/rfc2798
- Shibboleth Wiki:
  https://wiki.shibboleth.net/confluence/display/IDP30/AttributeResolverConfiguration
  https://wiki.shibboleth.net/confluence/display/IDP30/AttributeFilterConfiguration
  https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPAddAttribute