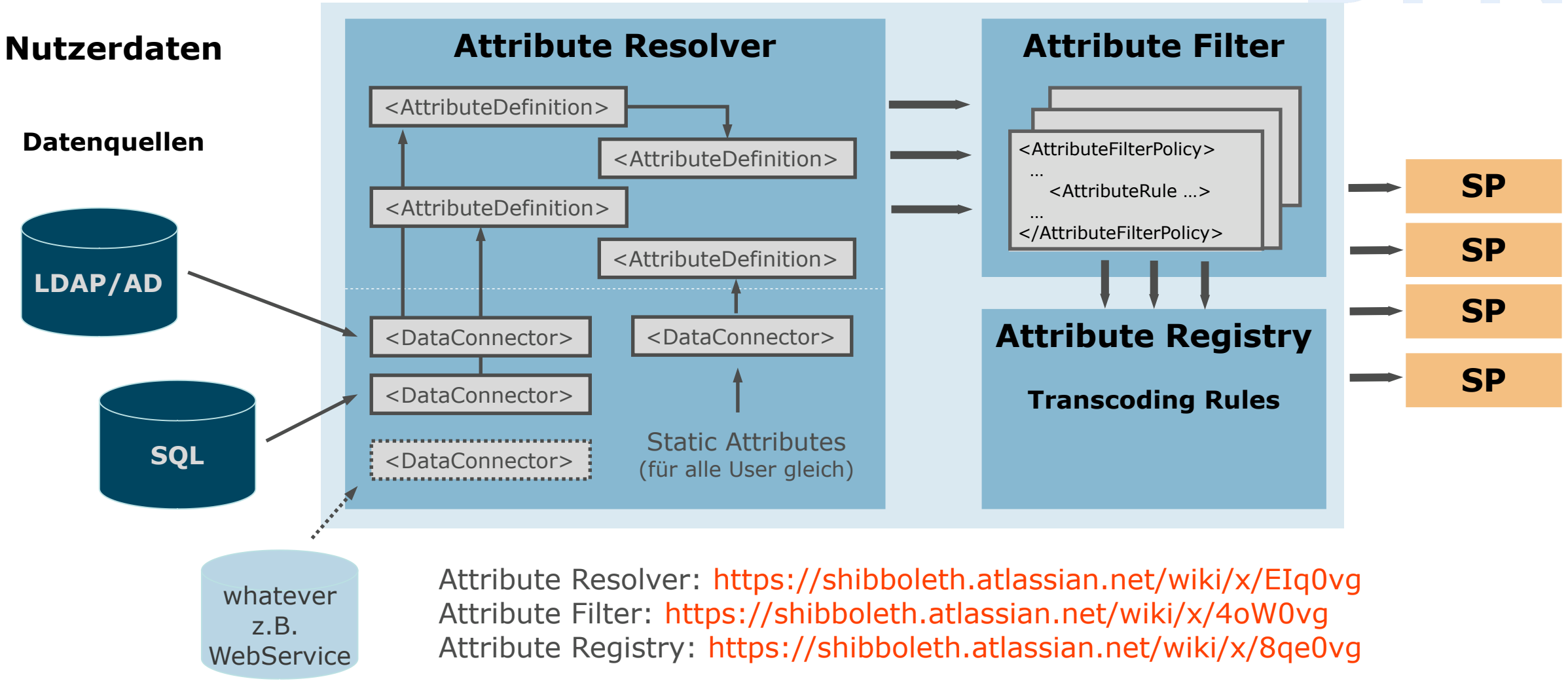
Attribute

IdP Workshop, Teil 3

Andreas Borm , Doreen Liebenau

IdP – Attribute Resolver, Filter und Encoding



Attribute Resolver: https://shibboleth.atlassian.net/wiki/x/EIq0vg
Attribute Filter: https://shibboleth.atlassian.net/wiki/x/4oW0vg
Attribute Registry: https://shibboleth.atlassian.net/wiki/x/8qe0vg

Attributschemata

- Schema: Definition von Attribut-Mengen, den zulässigen Werten und deren Bedeutungen
- im Föderationsumfeld verbreitete Schemata:
    - eduPerson (international)
    - SCHAC (Schema for Academia, international)
    - dfnEduPerson (e-Learning, Deutschland)
    - weitere, z.B. inetOrgPerson
- Unsere Dokumentation
- Attribute für SAML-Kommunikation müssen nicht im IdM vorhanden sein!

Attribut-Handling im IdP

- IdP liest Attribute aus IdM/Nutzerverzeichnis
- Splitten, Zusammenfügen, Umschreiben von Attributen im IdP möglich
- Generierung neuer Attribute in Abhängigkeit von bereits definierten Attributen / Werten
- Attribute Filter Policies entscheiden über Weitergabe von Attributen an SPs
- Encoding der Attribute
- Einholung der Zustimmung zur Attributfreigabe (User Consent Modul)
- IdP verpackt Attribute in SAML-Assertion
- Versand der SAML-Assertion per HTTP-POST an Assertion Consumer Service (ACS) des anfragenden SPs (URL aus Föderationsmetadaten)

IdP: Auslesen der Attribute aus dem IdM

- Attribute Resolver:
  - Abschnitt „Data Connectors“
- Konfiguration:
  - `./conf/attribute-resolver.xml`
  - `./conf/ldap.properties`
- Unsere Doku

```
<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}"
    responseTimeout="%{idp.attribute.resolver.LDAP.responseTimeout}"
exportAttributes="uid givenName sn mail displayName">
    <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}" />
</DataConnector>
```

IdP: Attributdefinitionen im IdP

- Umschreiben und Neugenerierung von Attributen wird ebenfalls in `./conf/attribute-resolver.xml` konfiguriert
- Abschnitt: „Attribute Definitions“
- Typen von Attribute Definitions (Auswahl)
  - Simple
  - Mapped
  - ScriptedAttribute
  - Scoped
- Shibboleth-Dokumentation

```
<AttributeDefinition xsi:type="Simple" id="ou">
    <InputDataConnector ref="myLDAP" attributeNames="ou"/>
</AttributeDefinition>
```

IdP: Attribute Encoding

- IdP wandelt seine Attribute um, damit der empfangende SP sie versteht.
- Konfiguration in der Attribute Registry, unterhalb von `./conf/attributes`
  - `z.B. “ou” in inetOrgPerson.xml`

```
<bean parent="shibboleth.TranscodingProperties">
    <property name="properties">
        <props merge="true">
            <prop key="id">ou</prop>
            <prop key="transcoder">SAML2StringTranscoder SAML1StringTranscoder</prop>
            <prop key="saml2.name">urn:oid:2.5.4.11</prop>
            <prop key="saml1.name">urn:mace:dir:attribute-def:ou</prop>
            <prop key="displayName.en">Organizational unit</prop>
            <prop key="displayName.de">Organisationseinheit</prop>
            <prop key="displayName.fr">Unité organisationnelle</prop>
            ….
        </props>
    </property>
</bean>
```

- Doku im Shib-Wiki

IdP: Attribut-Filter

- legt fest, welche Attribute an einen SP oder eine Gruppe von SPs versendet werden
- Attribute werden anhand der ID aus Attribute Resolver referenziert
- sehr flexibel, Regeln anhand vieler Kriterien möglich, z.B.
  - SP (EntityID)
  - Föderation
  - Entity Attribute
  - User
  - Attribut-Werte
- Doku im Shib-Wiki

```
<AttributeFilterPolicy id="dfn_test_sps">
    <PolicyRequirementRule xsi:type="OR">
        <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="ou"                     permitAny="true"/>
    <AttributeRule attributeID="sn"                     permitAny="true"/>
    <AttributeRule attributeID="givenName"              permitAny="true"/>
</AttributeFilterPolicy>
```

IdP: Attributfreigabe

- User Consent-Modul
- **Nachweispflicht!**
- Informationen zur Einwilligung der Endnutzenden zur Attributfreigabe werden im Logfile `idp-consent-audit.log` abgelegt
  → gut aufheben

User Identifier

Identifikation von Usern (Subjects)

- Entwicklung Identifier:
  - SAML1
    - <NameIdentitfier>- Element
  - SAML2.0
    - <NameID>-Element
    - Mehrere NameID-Formate
      - <saml2:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent" NameQualifier="https://idp.local/idp/shibboleth" SPNameQualifier="https://sp1.local/shibboleth" xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">HFgOn7tH2rKS1pmtzVARvEDOUUk=</saml2:NameID>
  - SAML2.0 (neu)
    - <Attribute>- Element
    - Subject Identifier Attributes Profile
      - <saml2:Attribute FriendlyName="samlPairwiseID" Name="urn:oasis:names:tc:SAML:attribute:pairwise-id" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"> <saml2:AttributeValue>M22SVWRXLZIVUONQ55M522POYCQRIGJ4@local</saml2:AttributeValue></saml2:Attribute>

Übersicht der Entwicklung

Phase	Mechanismus	Typischer Identifier	**Kommentar** (siehe Best Practice)
SAML 1.x	NameIdentifier (im Subject)	E-Mail, lokale User-ID	- Wenig Datenschutz, - keine Pairwise-IDs
SAML 1.1	SAML 1.1 Attribut (als Attribut)	eduPersonTargetedID	- pairwise-like - aus Kompatibilitätsgründen noch verbreitet - komplexes XML-Konstrukt
SAML 2.0	NameID (im Subject)	persistent, transient, emailAddress, unspecified	- Mehr Formate, - komplex, - Datenschutzprobleme
SAML 2.0 (neu)	Subject Identifier Attribute (als Attribut)	subject-id, pairwise-id	-Klare Semantik, - Datenschutz, - Interoperabilität