deutsches forschungsnetz

SAML Service Provider

Einführung und Überblick

DFN-AAI Shibboleth Workshop Februar 2021| 22.2.2021

Wolfgang Pempe (pempe@dfn.de)

Workshop-Woche Service Provider

- Montag, 22.2.2021, 14:00-16:00:
  SAML Service Provider – Einführung und Überblick (Wolfgang Pempe, DFN)
- Dienstag, 23.2.2021, 9:00-12:00 und 13:00-16:00
  Praxis: Einführung in den Shibboleth SP
  (Silke Meyer, DFN und Steffen Hofmann, FU Berlin)
- Donnerstag, 25.2.2021, 10:00-12:00
  SimpleSAMLphp als Service Provider (Frank Tröger, FAU Erlangen-Nürnberg)
- Donnerstag, 25.2.2021, 14:30 bis 16:00
  Austausch zum Thema SP Onboarding und Wrap-Up (Mod.: Silke Meyer, DFN)

Inhalt

- Grundlagen
- Was ist und welche Aufgaben erfüllt ein SAML-fähiger Service Provider?
- Technische Implementierung

Pause

- Funktionsweise und Funktionsumfang Shibboleth SP
- Überlegungen zur Nachhaltigkeit

Grundlagen

Föderierte Identität …

User Id: `lmuster`
Passwd: ••••••

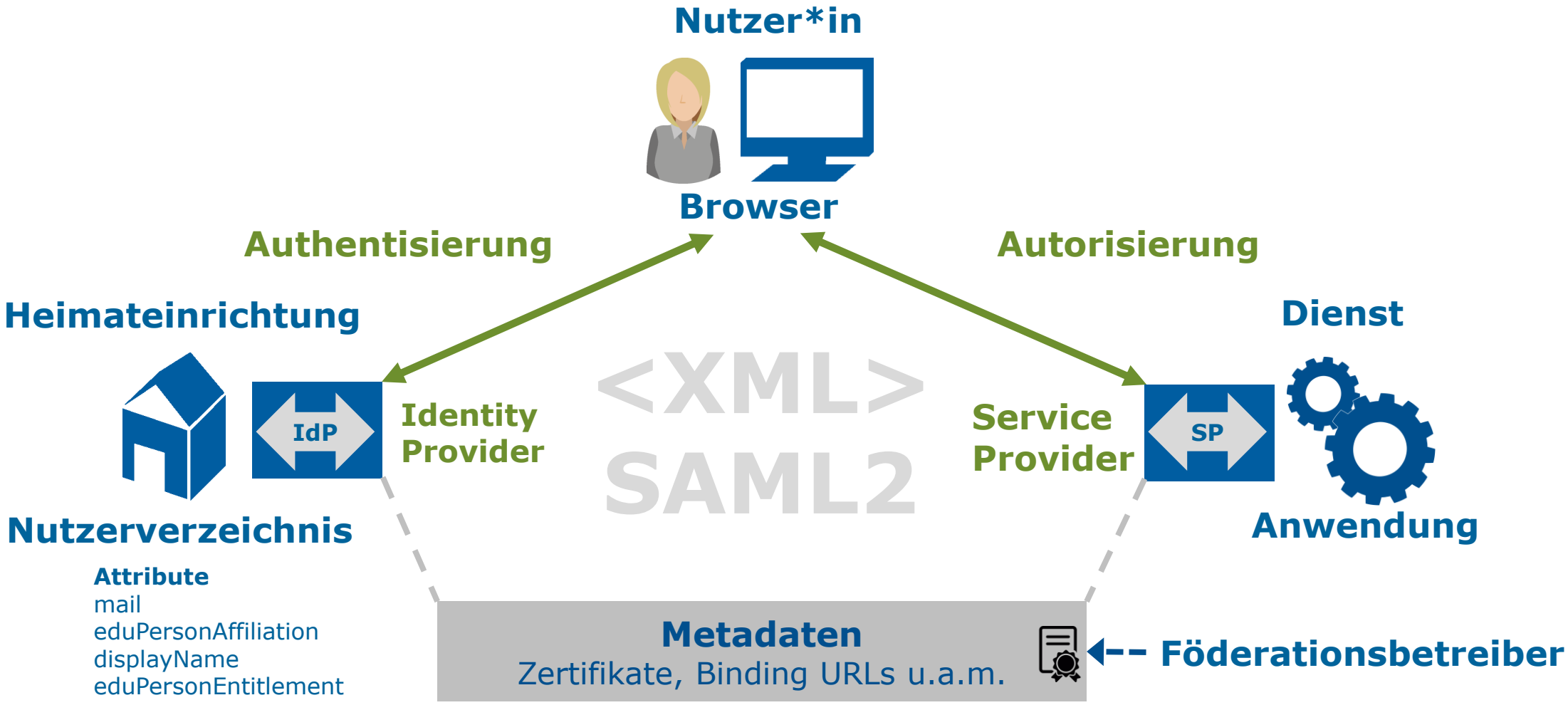
Universität Musterstadt

Online Shop XY

Datenbankanbieter YZ

Verlag WX

… im Zusammenspiel von IdP und SP

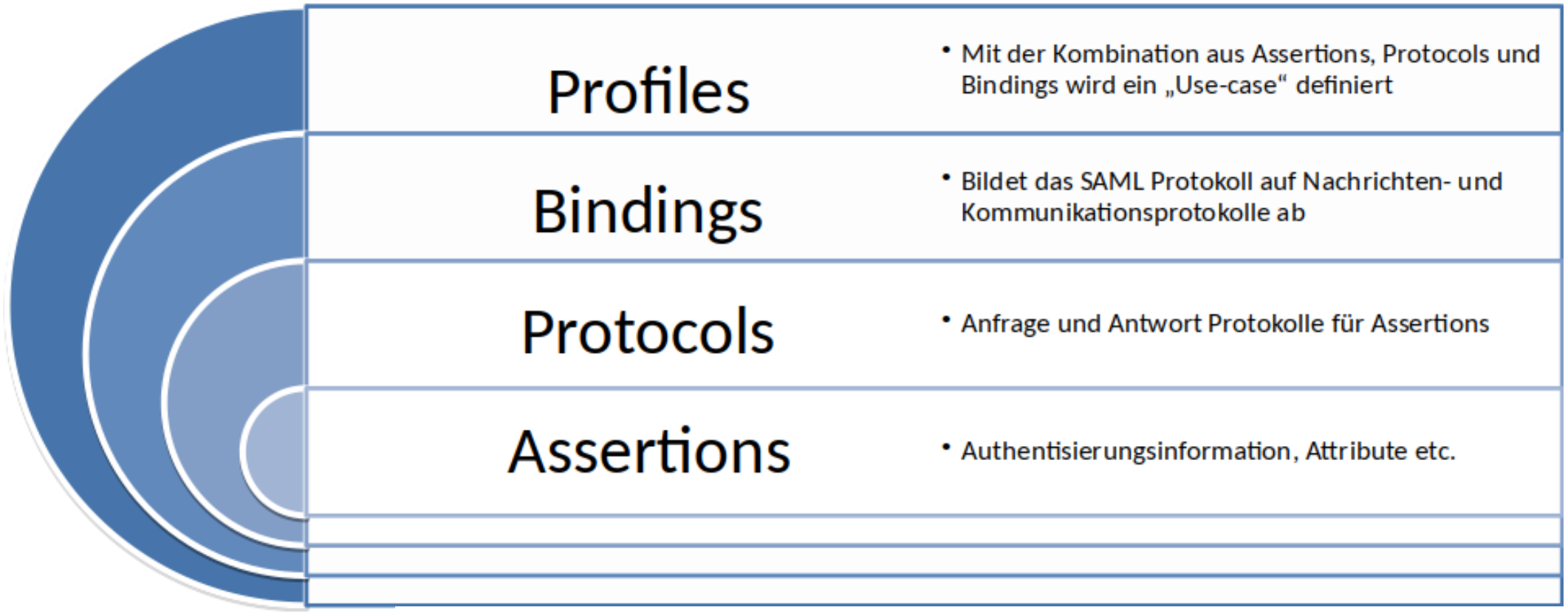


Siehe auch: https://wiki.shibboleth.net/confluence/display/CONCEPT/Home

SAML

- Steht für: **S**ecurity **A**ssertion **M**arkup **L**anguage
- XML-Framework (offener Standard bei OASIS), das aus mehreren Spezifikationen besteht, aktuell: SAML 2.0
- Die wichtigsten Komponenten:
  - **Metadata**
  - **Assertions** + Protocols
  - Bindings
  - Profiles
- Siehe https://docs.oasis-open.org/security/saml/v2.0/
- bzw. https://wiki.oasis-open.org/security

SAML – Komponenten



Authentication Context

- Definiert Art und Weise der Authentifizierung

Metadata

- Konfigurationsdaten für Service- und Identityprovider

Quelle: Michael Simon, KIT

Beispiel: Web Browser SSO Profile

- Bietet Single Sign-On für browser-basierte Webapplikationen
- Nutzer*in mit Browser will auf eine geschützte Resource beim Service Provider (SP) zugreifen
- Dabei kommen (z.B.) folgende Kombinationen zum Einsatz:
  - Protocol: Authentication Request Protocol
  - Binding: HTTP Redirect, HTTP POST, (HTTP Artifact)
- Details folgen

SAML Metadaten – typunabhängige Elemente

Wurzelelement

`<EntityDescriptor entityID="https://entity-xyz.de">`

Erweiterung gegenüber der ersten Fassung des Standards

`<Extensions>`

Informationen für User Interfaces

`<UIInfo>`

Zertifikate

`<KeyDescriptor>`

Benötigte / unterstützte Name Identifier

`<NameIDFormat>`

Kontaktdaten

`<Organization>`, `<ContactPerson>` (Typ: technical, administrative, support, security)

SAML Metadaten – Identity Provider

IdP Single Sign-On Descriptor

```
<IDPSSODescriptor>
```

„Scope" - Bezeichnung der Heimateinrichtung

```
<saml1md:Scope regexp="false">dfn.de</saml1md:Scope>
```

Bindings für SSO und SLO (Single Log-out)

```
<SingleSignOnService>, <SingleLogoutService>
```

Attribute Authority Descriptor (optional)

```
<AttributeAuthorityDescriptor>
```

Bindings für Attribute Queries (optional)

```
<AttributeService>
```

SAML Metadaten – Service Provider

SP Single Sign-On Descriptor

`<SPSSODescriptor>`

Bindings für die Entgegennahme von Assertions

`<AssertionConsumerService>`

Bindings für SLO (Single Log-out)

`<SingleLogoutService>`

Deklaration der vom SP benötigten Attribute

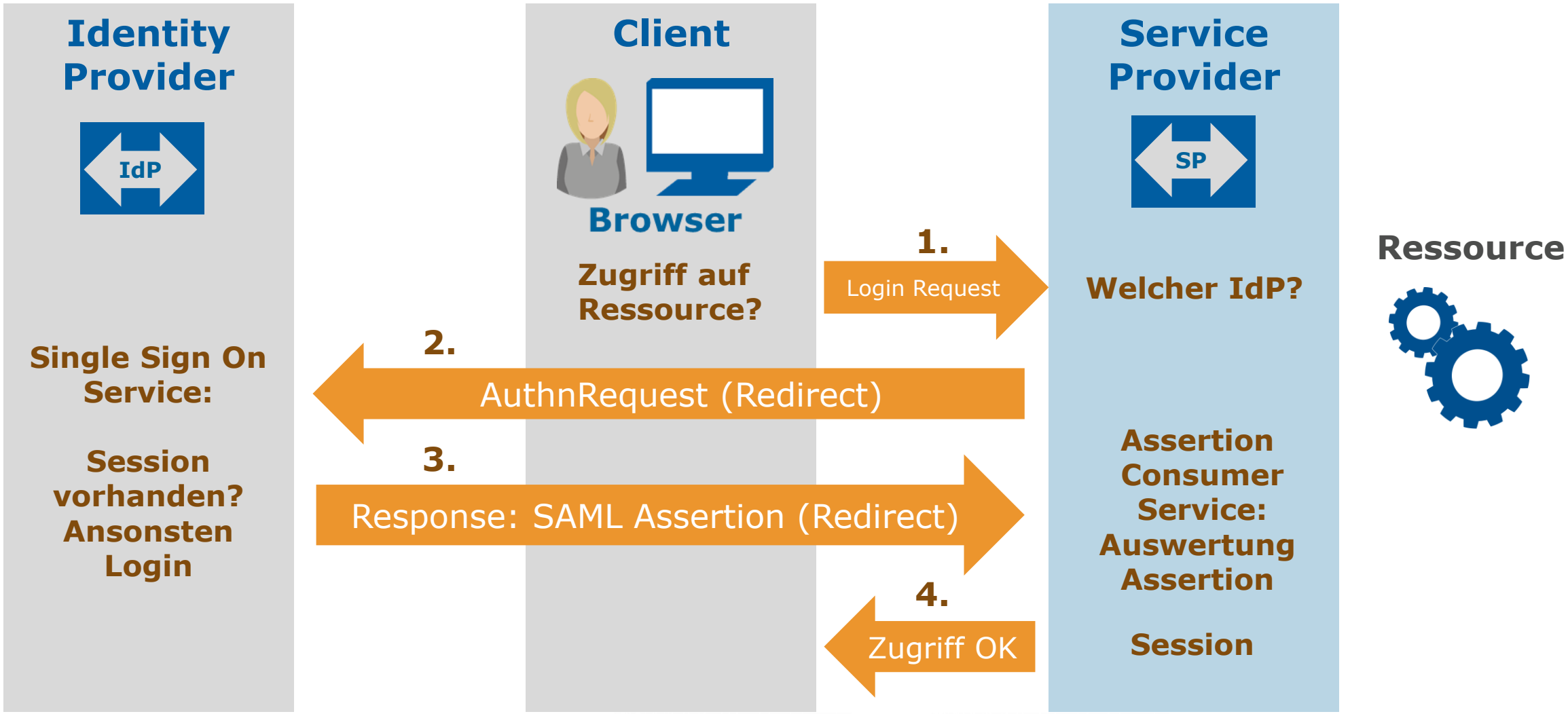
`<AttributeConsumingService>`

Was ist und welche Aufgaben erfüllt ein SAML-fähiger Service Provider?

Steckbrief Service Provider

- Schützt Ressourcen → Autorisierung
- Sitzt daher ‚irgendwo' zwischen Webserver und Ressource
- Discovery: Bietet den Nutzer*innen ggf. Möglichkeit zur Auswahl des Identity Providers der jeweiligen Heimateinrichtung
- Sendet Authentication Request zum Heimat-IdP des/der Nutzer*in
- Wertet Antwort des IdP (Assertion Consumer Service) aus und reicht bestimmte Daten (u.a. Attribute) an dahinterliegende Anwendung weiter
- Logout: Logout Request an IdP und ggf. Beenden der Anwendungs-Session

Ablauf Web Browser SSO



SP → IdP: AuthnRequest

- Service Provider bringt die Entity ID des betreffenden IdP in Erfahrung
- Anhand der Entity ID wird in den (Föderations)Metadaten nach einer `<SingleSignOnService>` Location bzw. einem URL gesucht, an den der AuthnRequest gesendet wird, bevorzugt:

  `Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"`
- AuthnRequest wird nicht verschlüsselt und i.d.R. auch nicht signiert

SP → IdP: AuthnRequest, Beispiel

SP issues an AuthnRequest to IdP

```
<?xml version="1.0" encoding="UTF-8"?>
<samlp:AuthnRequest
   AssertionConsumerServiceURL="https://testsp2.aai.dfn.de/Shibboleth.sso/SAML2/POST"
   Destination="https://testidp2.aai.dfn.de/idp/profile/SAML2/Redirect/SSO"
   ID="_ac9984cf86fbf3db7c3ff1aa769c2571"
   IssueInstant="2016-11-18T23:08:41Z"
   ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
   Version="2.0" xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
   <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">https://testsp2.aai.dfn.de/shibboleth</saml:Issuer>
   <samlp:NameIDPolicy AllowCreate="1"/>
</samlp:AuthnRequest>
```

1. reads SP's Entity ID…

2. performs a lookup in federation metadata…

Federation Metadata

```
<EntityDescriptor entityID="https://testsp2.aai.dfn.de/shibboleth">
  <Extensions><!-- . . . --></Extensions>
  <SPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <!-- . . . -->
    <AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
                              Location="https://testsp2.aai.dfn.de/Shibboleth.sso/SAML2/POST" index="1"/>
```

3. checks if any of the ACS URLs matches with the one in the AuthnRequest?

yes → **Continue**

no → **Abort [ERROR]**

IdP → SP: Response

```
<?xml version="1.0" encoding="UTF-8"?>
<saml2p:Response xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol"
  Destination="https://testsp2.aai.dfn.de/Shibboleth.sso/SAML2/POST"
  ID="_043991a786454dcfc0df45faeae5f033"
  InResponseTo="_ac9984cf86fbf3db7c3ff1aa769c2571"
  IssueInstant="2016-11-18T23:08:42.346Z" Version="2.0">
  <saml2:Issuer xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">https://testidp2.aai.dfn.de/idp/shibboleth</saml2:Issuer>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#"><!-- ... --></ds:Signature>
  <saml2p:Status>
    <saml2p:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
  </saml2p:Status>
  <saml2:Assertion ID="_03865718e927e18b30e37bd58ecbd45d"
    IssueInstant="2016-11-18T23:08:42.346Z" Version="2.0" xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
    <!-- ...-->
  </saml2:Assertion>
</saml2p:Response>
```

- IdP sendet Response an `AssertionConsumerServiceURL` aus AuthnRequest (s.o. `Destination`), `InResponseTo` enspricht `ID` aus AuthnRequest
- Assertion i.d.R. verschlüsselt, Response signiert

IdP → SP: Assertion

```
<saml2:Assertion ID="_03865718e927e18b30e37bd58ecbd45d"
  IssueInstant="2016-11-18T23:08:42.346Z" Version="2.0" xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
  <saml2:Issuer>https://testidp2.aai.dfn.de/idp/shibboleth</saml2:Issuer>
  <saml2:Subject>
    <saml2:NameID ...><!-- ... --></saml2:NameID>
    <saml2:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer"><!-- ... --></saml2:SubjectConfirmation>
  </saml2:Subject>
  <saml2:Conditions NotBefore="2016-11-18T23:08:42.346Z" NotOnOrAfter="2016-11-18T23:13:42.346Z">
    <saml2:AudienceRestriction>
      <saml2:Audience>https://testsp2.aai.dfn.de/shibboleth</saml2:Audience>
    </saml2:AudienceRestriction>
  </saml2:Conditions>
  <saml2:AuthnStatement AuthnInstant="2016-11-18T23:08:42.051Z" SessionIndex="_3d57770f4530c4e26756a3f1d8efc07b">
    <saml2:SubjectLocality Address="2001:638:d:c001::9"/>
    <saml2:AuthnContext>
      <saml2:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</saml2:AuthnContextClassRef>
    </saml2:AuthnContext>
  </saml2:AuthnStatement>
  <saml2:AttributeStatement>
     <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>
</saml2:Assertion>
```

Assertion Consumer Service

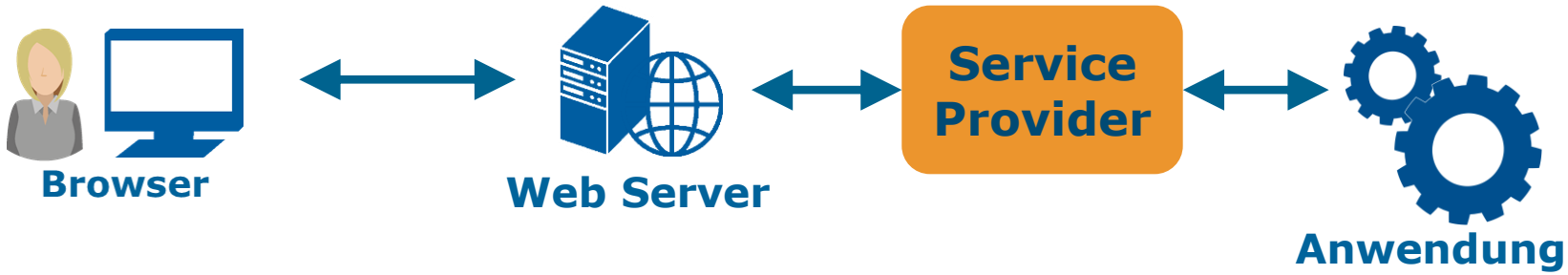
- Wertet Assertion aus, u.a.
  - Name IDs
  - Authentication Context
  - Attribute
- Anhand dieser Angaben
  - trifft der SP Autorisierungs-Entscheidung
  - konsultiert ggf. weitere Attributquellen, z.B. Attribute Query gegen Gruppenverwaltung (VO-Management) einer Forschungsinfrastruktur
  - Reicht Informationen an die geschützte Anwendung weiter

Technische Implementierung

Unterschiedliche Ansätze

1. Library

2. Standalone



3. Integration in Web Server

Beispiele Softwareprodukte

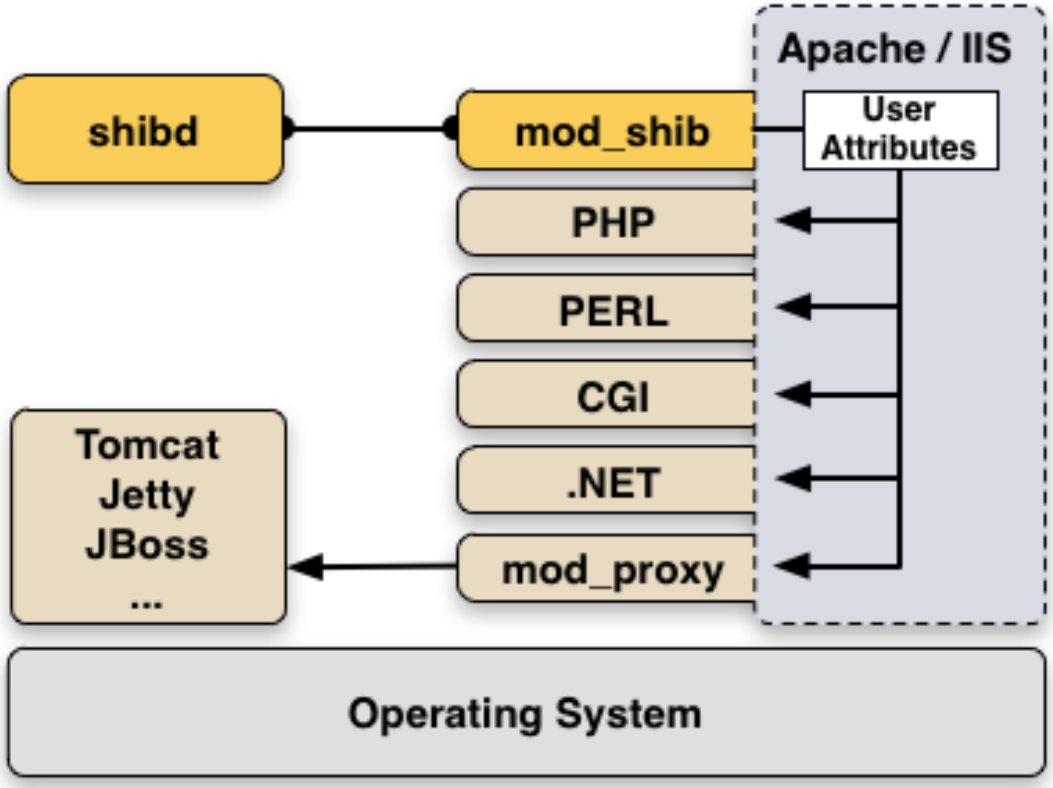
- Library
  - OpenSAML (C++, Java)
- Standalone
  - SimpleSAMLphp
- Integration Web Server
  - mod_auth_mellon
  - PySAML2 (+ mod_wsgi)
  - Shibboleth Service Provider
  - SimpleSAMLphp mit AuthMemCookie Apache Module
- IAM-Plattformen, z.B. Ping Identity, ForgeRock, Unity IdM, Keycloak, …

Open Source

Shibboleth Service Provider

Kurzübersicht Shibboleth SP

- Zwei Komponenten:
  - Web Server-Modul für Apache (`mod_shib`) und IIS Web Server (`iis7_shib.dll`): Schützt Files, Directories, Locations und erzwingt AAI-basierte Autorisierung
  - `shibd` Daemon: State Management, Verarbeitung XML, Security, „Logik“
- Kommunikation mit Anwendung:
  - Attribute werden auf Umgebungsvariablen abgelegt, auf die alle Anwendungen zugreifen können, die im Web Server laufen, z.B. PHP: $_SERVER['mail']
  - Handler (siehe folgende Folien)



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

Hinweise zu Installation und Konfiguration

- Zentrale Doku im Shibboleth Wiki: https://wiki.shibboleth.net/confluence/display/SP3/Home
- Repositories für Debian und Ubuntu werden von SWITCH gepflegt, die auch ausführliche Dokumentation zu Installation und Konfiguration für unterschiedliche Plattformen bereitstellen: https://www.switch.ch/aai/guides/sp/
- DFN-AAI Wiki: Basis-Konfiguration und Embedded Discovery Service: https://doku.tid.dfn.de/de:shibsp
- Schulungsmaterialien: https://doku.tid.dfn.de/de:aai:training:shibsp

Verzeichnisse unter Debian (Auswahl)

- /etc/shibboleth/
  - Zentrale Konfiguration in `shibboleth2.xml`
  - Attribute: `attribute-map.xml`, `attribute-policy.xml`
  - Logging: u.a. `native.logger` (mod_shib), `shibd.logger` (shibd)
  - HTML Templates (`*.html`)
  - Lokal generierte Zertifikate + Keys
- /var/cache/shibboleth/
  - Backup Remote Metadata
- /var/log/shibboleth/
  - u.a. `shibd.log` und `transaction.log` (`native.logger` per Default ins syslog)

Anpassung der Konfiguration (1)

In der Regel sind nur wenige Anpassungen in `shibboleth2.xml` erforderlich:

- Entity ID des SP und ggf. Belegung `REMOTE_USER`

```
<ApplicationDefaults entityID="https://sp.example.org/shibboleth"
REMOTE_USER="eppn subject-id pairwise-id persistent-id“>
```

- Security-Einstellungen für SP-Sessions (Cookies, https, Timeout, etc.)

```
<Sessions lifetime="28800" timeout="3600" relayState="ss:mem"
          checkAddress="false" consistentAddress="true"
          handlerSSL="true" cookieProps="https"
          redirectLimit="host">
```

Anpassung der Konfiguration (2)

- Session Initiator: definiert u.a., wie die Weiterleitung der Nutzer*innen zum IdP erfolgt (hart verdrahtet oder Discovery Service)

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

(Element `<SSO>` ist vereinfachte Version von `<SessionInitiator>`)

- Kontakt-Informationen für die Anzeige von Fehlerseiten

```
<Errors supportContact="helpdesk@example.org"
        helpLocation="/about.html"
        styleSheet="/shibboleth-sp/main.css"/>
```

Anpassung der Konfiguration (3)

- (Föderations-)Metadaten

```
<MetadataProvider type="XML"
 url="http://www.aai.dfn.de/fileadmin/metadata/dfn-aai-idp-metadata.xml"
 validate="true" backingFilePath="dfn-aai-idp-metadata.xml" reloadInterval="3600">
   <MetadataFilter type="Signature" certificate="/etc/ssl/aai/dfn-aai.pem"/>
</MetadataProvider>
```

- Zertifikat(e) und Private Key(s) für Signierung und Ver-/Entschlüsselung der SAML-basierten Kommunikation

```
<CredentialResolver type="File"
  key="/etc/ssl/private/myvhost.mydomain.de.key.pem"
 certificate="/etc/ssl/localcerts/myvhost.mydomain.de.crt.pem"/>
```

Überblick shibboleth2.xml

- Siehe Beispiel Shib SP 3.1 für Debian 10

Handler (1)

- Eine Art API, die verschiedene Funktionen über URLs zur Verfügung stellt: https://sp.uni-musterstadt.de/Shibboleth.sso/...
- Metadaten gemäß aktueller Konfiguration: `.../Shibboleth.sso/Metadata`
- Session-Informationen: `…/Shibboleth.sso/Session`

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

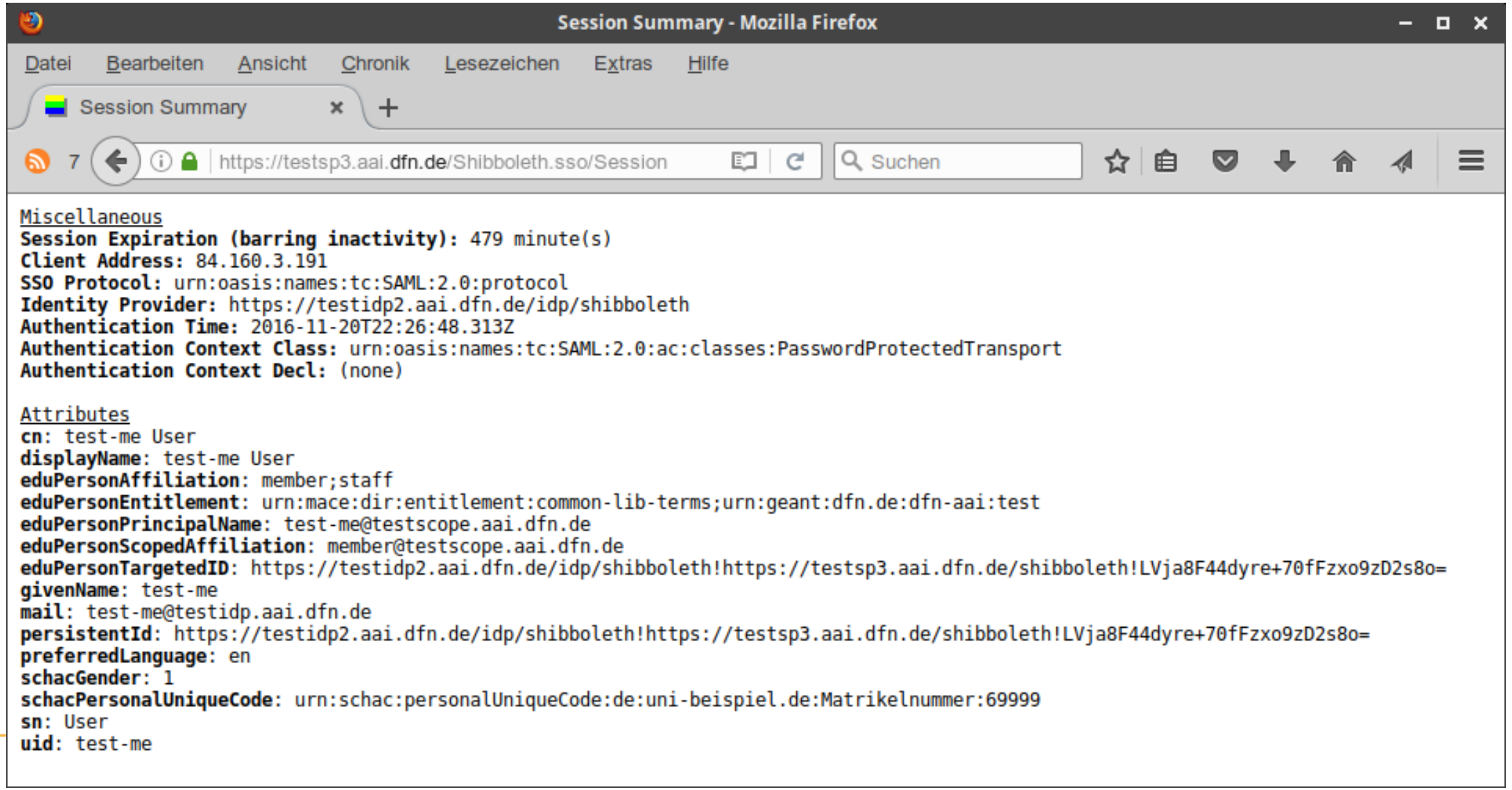
- Status-Informationen: `…/Shibboleth.sso/Status`

```
<Handler type="Status" Location="/Status" acl="127.0.0.1 ::1"/>
```

Handler (2)

- Discovery Feed – im JSON-Format aus den eingelesenen und ggf. gefilterten Metadaten generiert, als Basis für einen Embedded Discovery Service (EDS): `…/Shibboleth.sso/DiscoFeed`
- Login/Session Initiator, z.B. um Shib-Session seitens der Anwendung zu starten: `…/Shibboleth.sso/Login`
- Logout Initiator, z.B. um Shib-Session seitens der Anwendung zu beenden: `…/Shibboleth.sso/Logout`

Beispiel Session-Info

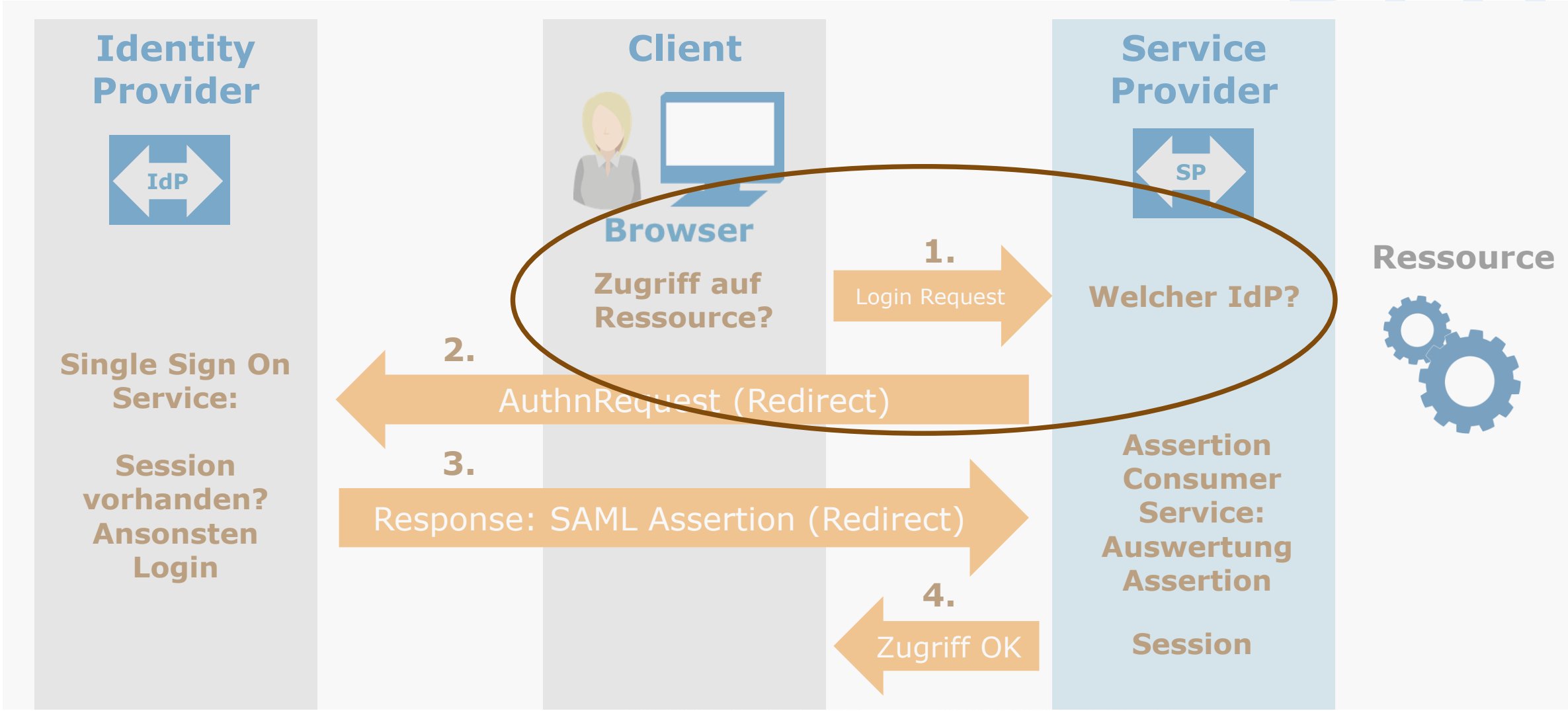


Attributquellen

- Von **IdPs** gesendete SAML Assertions
- Bei Bedarf zusätzliche Attributquellen: **Attribute Authorities** (via Attribute Query) (s.u. https://doku.tid.dfn.de/de:aai:attribute_authority)

```
<AttributeResolver type="SimpleAggregation" attributeId="subject-id"
 format="urn:oasis:names:tc:SAML:attribute:subject-id">
 <Entity>https://trusted-attribute-
authority.example.org/idp/shibboleth</Entity>
 <saml2:Attribute xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
  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"
  FriendlyName="eduPersonEntitlement"/>
</AttributeResolver>
```

Ablauf Web Browser SSO



Anwendungen schützen

- Die Regeln für die Zugriffskontrolle können auf unterschiedliche Arten definiert werden:
  - Web Server / Apache-Konfiguration: Apache Access Rules
  - Anwendung (SP: Lazy Session)
  - SP-Konfiguration: XML Access Control
  - SP-Handler: Attribute Checker (später im Ablauf, nach Attribut-Verarbeitung)
- Apache Access Rules ermöglichen einfache AND/OR-Verknüpfung von Bedingungen
- XML Access Control erlaubt komplexere Regeln

Apache Access Rules (1)

Schützen Files, Directories, Locations und triggern ggf. Session Initiator

```
<Location /protected>
    AuthType shibboleth
    ShibRequestSetting requireSession true
       <RequireAll>
          Require shib-attr affiliation staff@uni-xyz.de
          Require shib-attr mail .*
       </RequireAll>
</Location>
```

Die Bezeichnungen der Attribute bzw. Variablenname werden primär in `/etc/shibboleth/attribute-map.xml` definiert (siehe folgende Folien)

Apache Access Rules (2)

- Zwei Spezialfälle (können auch global in shibboleth2.xml, Element <SSO> gesetzt werden):
  - `ShibRequestSetting forceAuthn true` (erzwingt erneuten Login am IdP)
  - `ShibRequestSetting isPassive true` (wenn Heimat IdP bekannt und dort Session vorhanden, wird eine SP Session ohne weitere Interaktion gestartet, d.h. kein „Sign In“ Button o.ä. – siehe Doku im Shib Wiki)
- „Lazy Session“, d.h. mod_shib in Lauerstellung

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

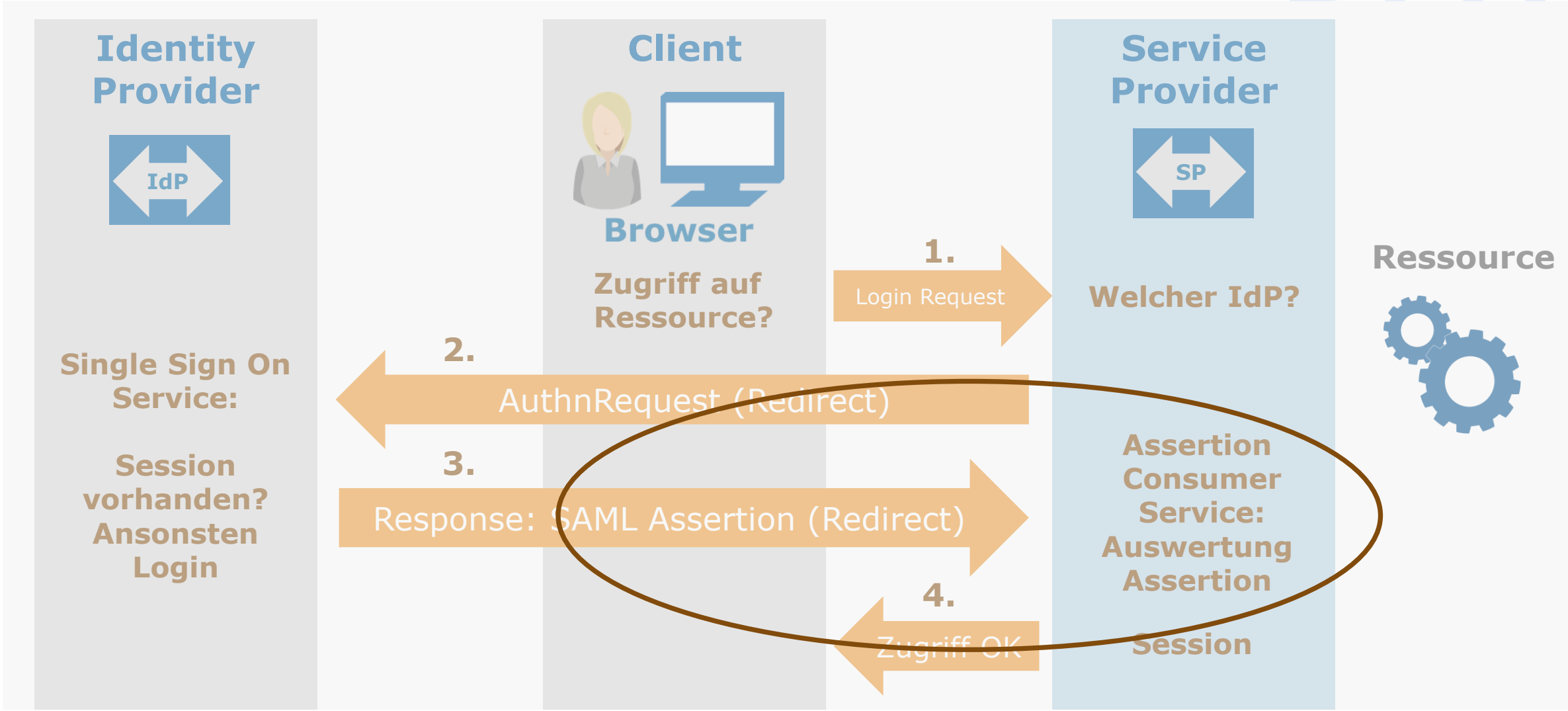
Anwendung muss Session starten → Login Handler (siehe vorherigen Folien)

XML Access Control

Zugriffsregeln in XML-Syntax außerhalb der Apache Konfiguration (Wichtig: In Apache muss `UseCanonicalName On` gesetzt sein!). Üblicherweise in `etc/shibboleth/shibboleth2.xml`:

```
<RequestMapper type="Native">
    <RequestMap applicationId="default">
      <Host name="sp.uni-musterstadt.de">
        <Path name="protected" authType="shibboleth" requireSession="true">
          <AccessControl>
            <AND>
              <Rule require="unscoped-affiliation">staff</Rule>
              <RuleRegex require="mail">.*</RuleRegex>
            </AND>
          </AccessControl>
        </Path>
      </Host>
    </RequestMap>
 </RequestMapper>
```

Ablauf Web Browser SSO



Verarbeitung von Attributen (1)

- Per Default extrahiert der Shib SP „Attribute“ aus SAML Assertions, d.h. aus `<saml2:AttributeStatement>` und `<saml2:Subject>` (transient-id, persistent-id)

  `<AttributeExtractor type="XML" validate="true" reloadChanges="false" path="attribute-map.xml"/>`

- Weitere Möglichkeiten (siehe hierzu Doku im Shib Wiki):

  `<AttributeExtractor type="Metadata">` (z.B. Kontaktdaten)

  `<AttributeExtractor type="Assertion">` (sonstige Elemente)

  u.a.m.

Verarbeitung von Attributen (2)

In **/etc/shibboleth/attribute-map.xml** werden Attribute aus Assertions auf Web Server-Variablen abgebildet (die anderen Attribute Extractors machen das direkt), Variablenname wird über „id“ definiert, z.B.:

```
<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: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:oasis:names:tc:SAML:2.0:nameid-format:persistent"
  id="persistent-id">
 <AttributeDecoder xsi:type="NameIDAttributeDecoder"
  formatter="$NameQualifier!$SPNameQualifier!$Name" defaultQualifiers="true"/>
</Attribute>
```

Verarbeitung von Attributen (3)

- Nächster Schritt: Welche Attribute bzw. Variablen und deren Werte werden weiterverarbeitet und an die Anwendung weitergereicht?
- Filter-Möglichkeiten über **`/etc/shibboleth/attribute-policy.xml`** (analog IdP), i.d.R. genügen die Voreinstellungen
- Beispiele
  - Syntax-Check: enthalten Scoped Attributes einen Scope? (@uni-xyz.de)
  - Kontrolliertes Vokabular: Zulässige Werte z.B. für (unscoped-)affiliation
  - Bestimmte Attribute, z.B. Entitlements nur von ausgewählten IdPs oder Attribute Authorities akzeptieren

Attribute Checker Handler

- Dieser Handler wird aktiv, bevor eine Weiterleitung auf eine geschützte Ressource erfolgt, Session Hook Mechanismus:

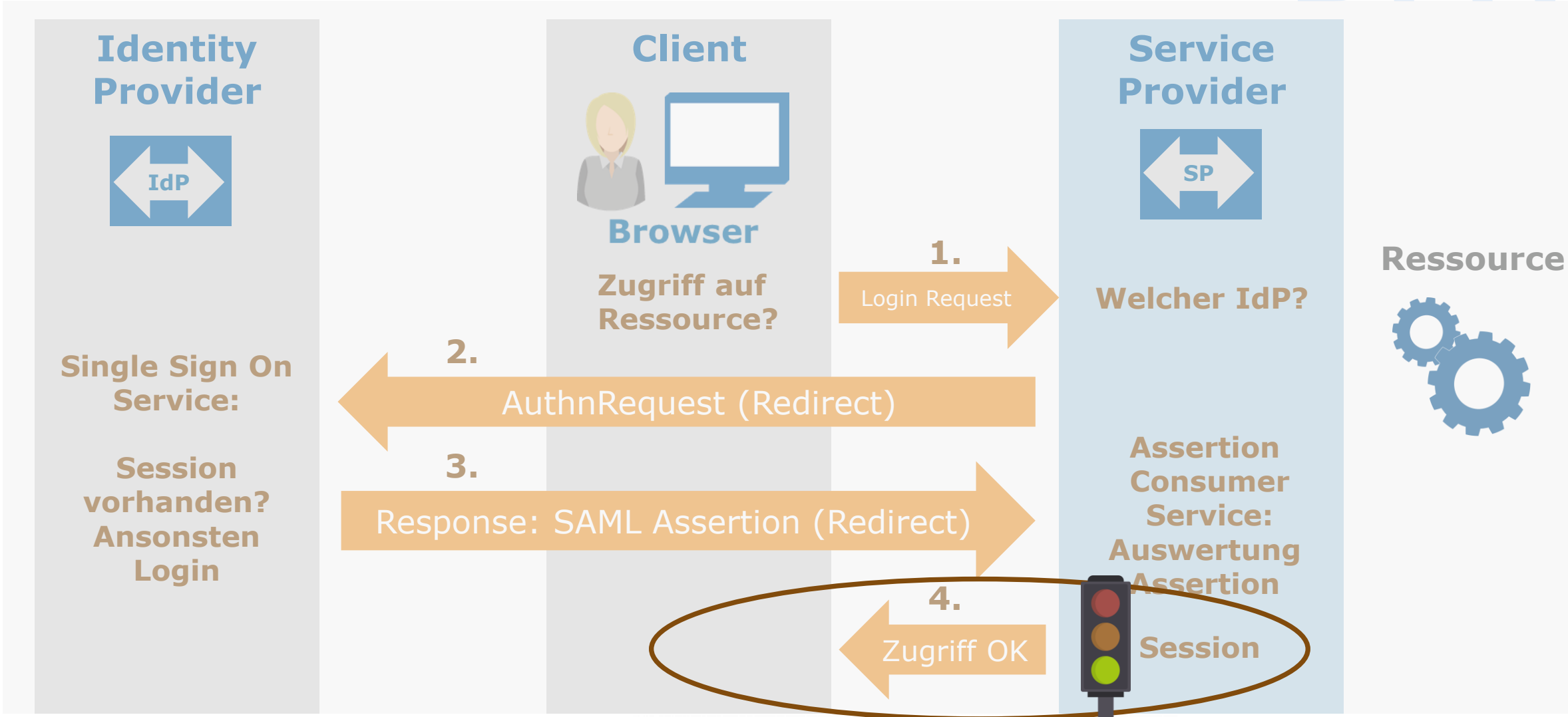
```
<ApplicationDefaults entityID="https://sp.example.org/shibboleth"
... sessionHook="/Shibboleth.sso/AttrChecker">
```

und

```
<Handler type="AttributeChecker" Location="/AttrChecker"
template="attrChecker.html" attributes="mail displayName"
flushSession="true"/>
```

- Ausführliches Beispiel im GÉANT Wiki: https://wiki.geant.org/display/eduGAIN/How+to+configure+Shibboleth+SP+attribute+checker

Fertig: Autorisierung



Weitere Themen

- Application Override
  - Mehrere VHosts (oder Pfade) auf einem Web Server, die jeweils SP-Funktionen benötigen, gekoppelt an eigene Entity ID und separate Konfiguration
  - Beispiele und Erklärungen im Praxisteil
  - https://wiki.shibboleth.net/confluence/display/SP3/ApplicationOverride
- REMOTE_USER
  - Spezielle CGI-Variable, in der die Identität des Users transportiert wird.
  - Eine oder mehrere der in attribute-map.xml definierten Variablen können verwendet werden, Konfiguration in `<ApplicationDefaults>` (siehe weiter oben)
  - Alternative zu Basic Auth: Damit lassen sich auch Applikationen durch Shibboleth schützen, die keine direkte Shibboleth-(Attribut-)Unterstützung mitbringen
- Logout: eigenes großes Thema

Überlegungen zur Nachhaltigkeit

Allgemeine Überlegungen

- Open Source vs. kommerzielle Produkte
  - Support vs. Vendor Lock-in
- Informationssicherheit
  - Schutz dienstlokaler Daten
- Datenschutz
  - User Deprovisionierung

Betriebliche/technische Aspekte

- Langfristige Pflege der Software (nächste Folien)
  - Regelmäßige Updates und Security Fixes
  - „Kümmern“ um Dependencies (externe Bibliotheken)
- Updates und Patches über Paketverwaltung des jeweiligen Betriebssystems oder analoge Mechanismen (also nicht aus Quellcode selber bauen)
- Unterstützung aktueller Sicherheits-Standards
- Nahtloser Key Rollover:
  gleichzeitige Unterstützung zweier Private Key - Zertifikat-Paare
- Umgang mit verschlüsselten Assertions/Responses

Status Open Source Produkte

- mod_auth_mellon
  - Community-Projekt? (vormals Uninett)
- PySAML2 / Identity Python
  - Community-Projekt, Unterstützung durch GÉANT 4-3 Projekt
- Shibboleth, OpenSAML
  - Shibboleth Consortium, finanziert aus Mitgliedsbeiträgen
  - Aktuell 9 Entwickler (m)
- SimpleSAMLphp
  - Community-Projekt (vormals Uninett)

Zukunft Shibboleth SP

- Aktuelle Software in C++ geschrieben, besteht weitgehend aus Dependencies, die z.T. nicht mehr maintained werden (insbes. XML-Libraries) oder bei denen nicht klar ist, welche Richtung deren Entwicklung nimmt
- Komplette Neuentwicklung geplant
- Details und Planungsstand Shibboleth SP 4: https://wiki.shibboleth.net/confluence/display/DEV/SP4Details
- Zukünftige Architektur API-basiert, soll auch andere Protokolle als SAML ermöglichen (→ OpenID Connect?)

Vielen Dank! Fragen? Kommentare?

- Kontakt
  - Wolfgang Pempe

    Teamleiter DFN-AAI
    E-Mail: pempe@dfn.de
    Tel.: +49-30-884299-308
    Fax: +49-30-884299-370

  - DFN-AAI Team

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

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