

Martin Lunze  
Zentrum für Informationsdienste und Hochleistungsrechnen  
Operative Prozesse und Systeme

# User-Deprovisionierung via Attribute-Query

Berlin, 70. DFN Betriebstagung // Dienstag 19. März 2019

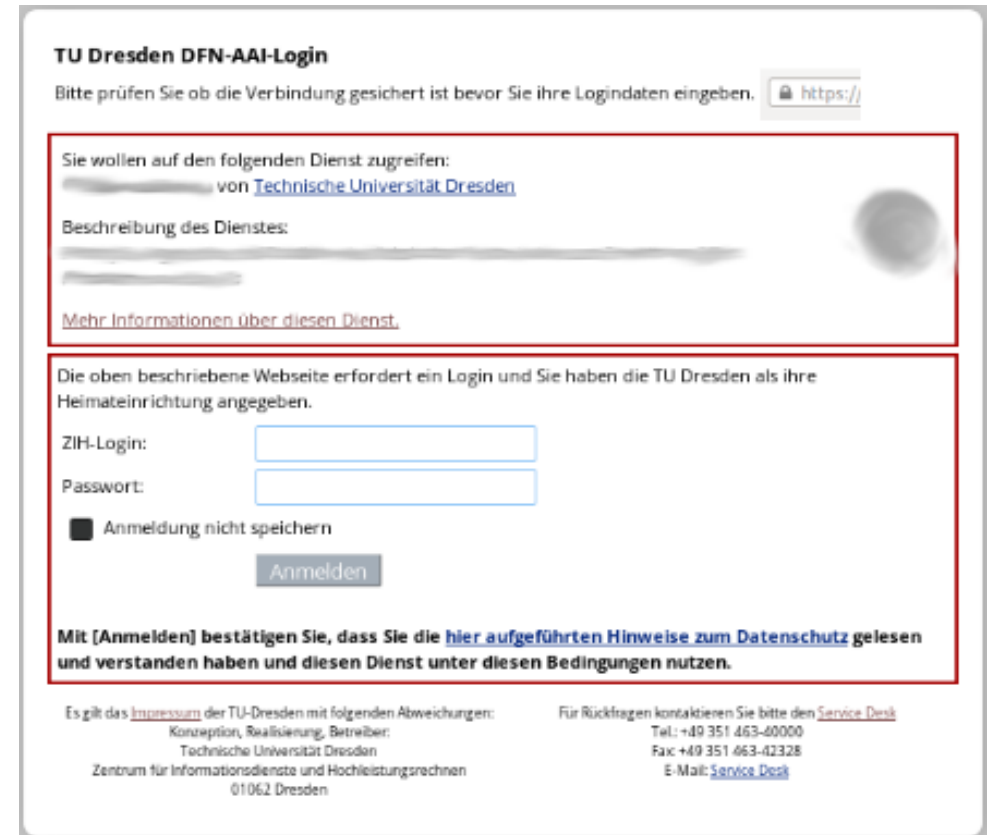
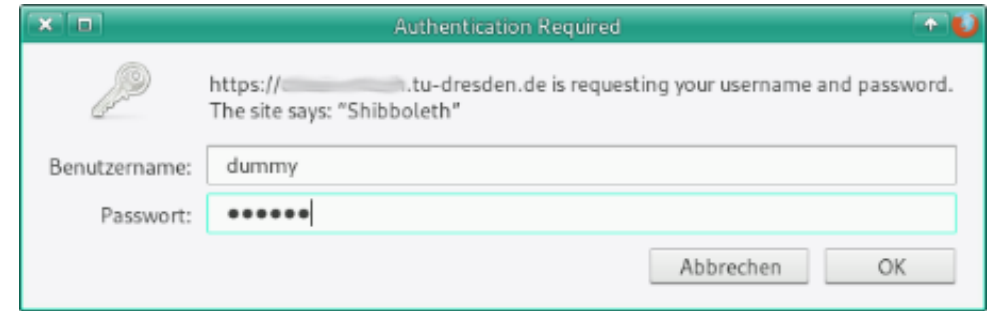
# Agenda

- Shibboleth an der TU Dresden
- Ausgangslage und Motivation
- Lösungsansätze
- SAML Attribute Query
- Voraussetzungen für Queries
- Queries stellen und auswerten
- Verlässlichkeit von Queries
- Datenschutzaspekte
- Wann und wie oft Queries stellen
- Fazit
- Links und Dokumentation

# Shibboleth an der TU Dresden

# Shibboleth an der TU Dresden

- **2007**
  - Einführung des Shibboleth Identity Provider (IdP) 1.x
  - 1 IdP Knoten, 1 Service Provider (SP)
  - Keine Benutzeroberfläche / Basic Authentifizierung
- **2012**
  - Umstellung auf IdP 2.x
  - 2 IdP Knoten (HA), aber weiterhin nur 1 SP
  - Benutzeroberfläche inklusive uApprove
  - Authentifizierung gegen openLDAP
  - Aufnahme in die DFN-AAI
- **Ab 2013**
  - Anbindung weiterer SPs (lokal + föderativ)



# Shibboleth an der TU Dresden

- **2017**
  - Umstellung auf IdP 3.x
  - Möglichkeit zum Single-Logout
- **2018**
  - Aufnahme in EduGain
  - Möglichkeit zur Deprovisionierung
- **Aktuell** (Stand 22.02.2019)
  - 50.000+ Nutzer
  - ~ 11.000 Logins pro Tag
  - > 40 lokale SPs
  - ~ 30 föderative SPs
  - Insgesamt bereits ~ 190 genutzte SPs
- **Zukünftig**
  - Speicherung des Attribute-Consent ;-)
  - 3 IdP Knoten

**TU Dresden DFN-AAI-Logout**

**Sie haben sich erfolgreich am Identity Provider ausgeloggt.**  
Es wird versucht Sie von folgenden Diensten abzumelden:

1. [redacted] dresden.de
2. [redacted] .tu-dresden.de
3. [redacted]
4. [redacted] .tu-dresden.de

Es gilt das Impressum der TU-Dresden mit folgenden Abweichungen:  
Konzeption, Realisierung, Betreiber:  
Technische Universität Dresden  
Zentrum für Informationsdienste und Hochleistungsrechnen  
01062 Dresden

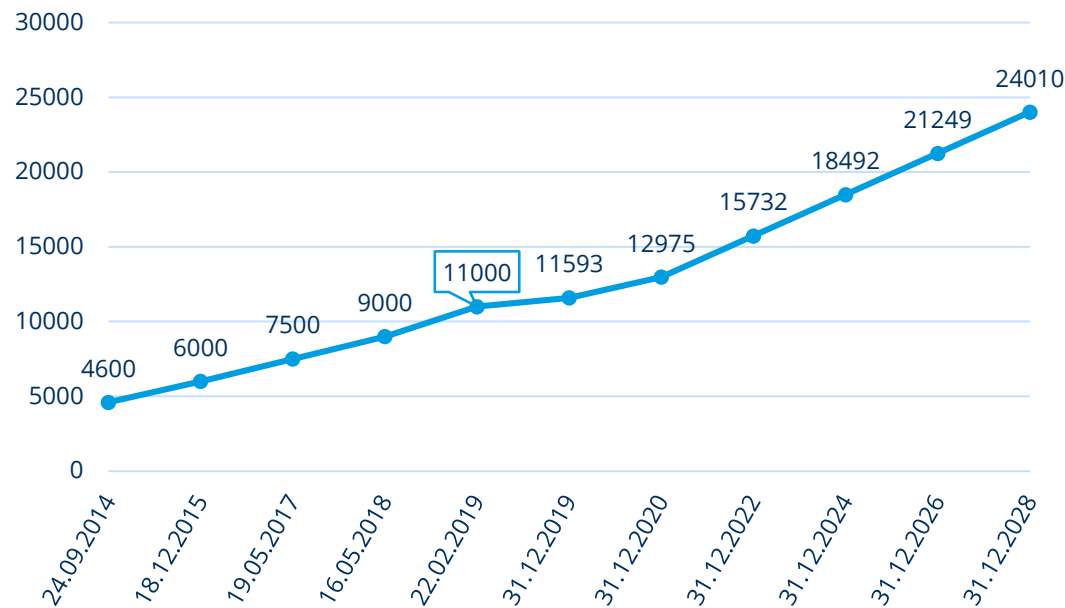
Für Rückfragen kontaktieren Sie bitte den Service Desk  
Tel.: +49 351 463-40000  
Fax: +49 351 463-02328  
E-Mail: Service Desk

# Ausgangslage und Motivation

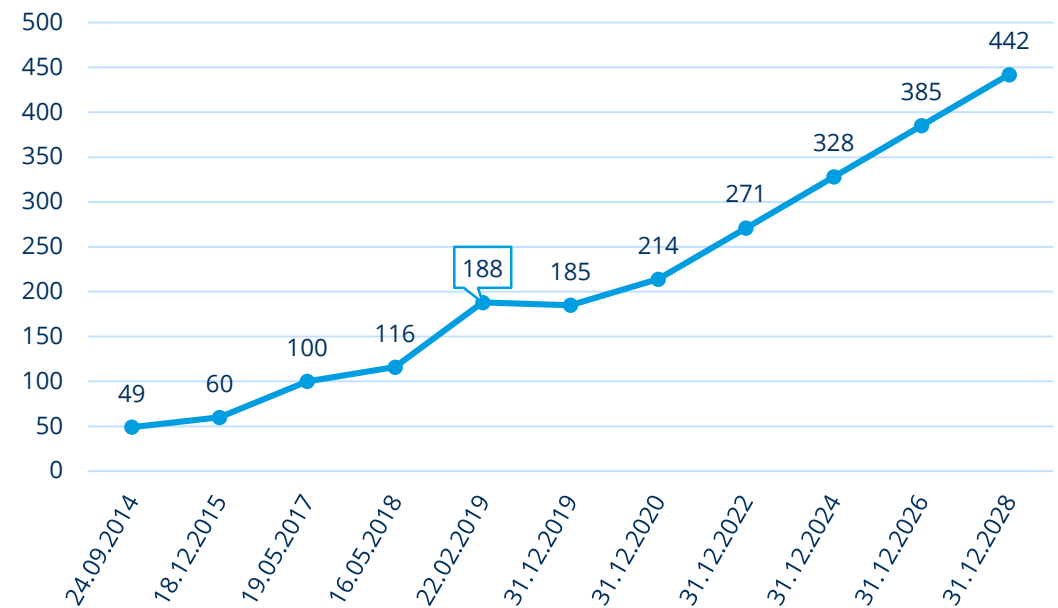
# Ausgangslage und Motivation

- Wachsende Anzahl der per Shibboleth angebundenen Webdienste
  - Zunehmende Verteilung von Nutzerdaten (pro Nutzer)
  - Wachsende Menge an Nutzerdaten (pro Dienst)
  - Steigende Streuung der Nutzerdaten (Einrichtungs- / Föderationsübergreifend)

Logins pro Tag

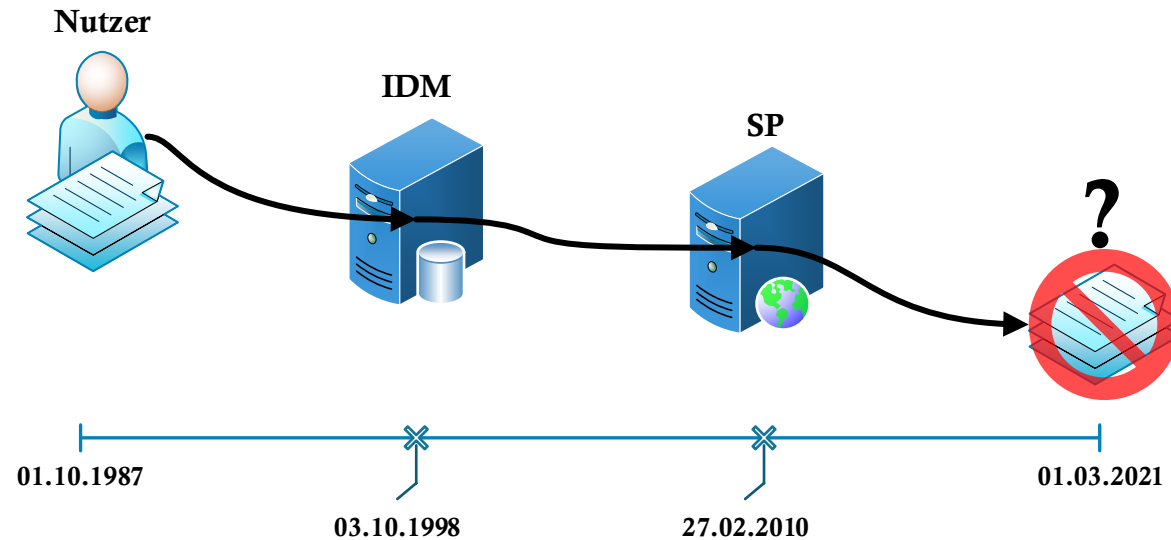


SPs



# Ausgangslage und Motivation

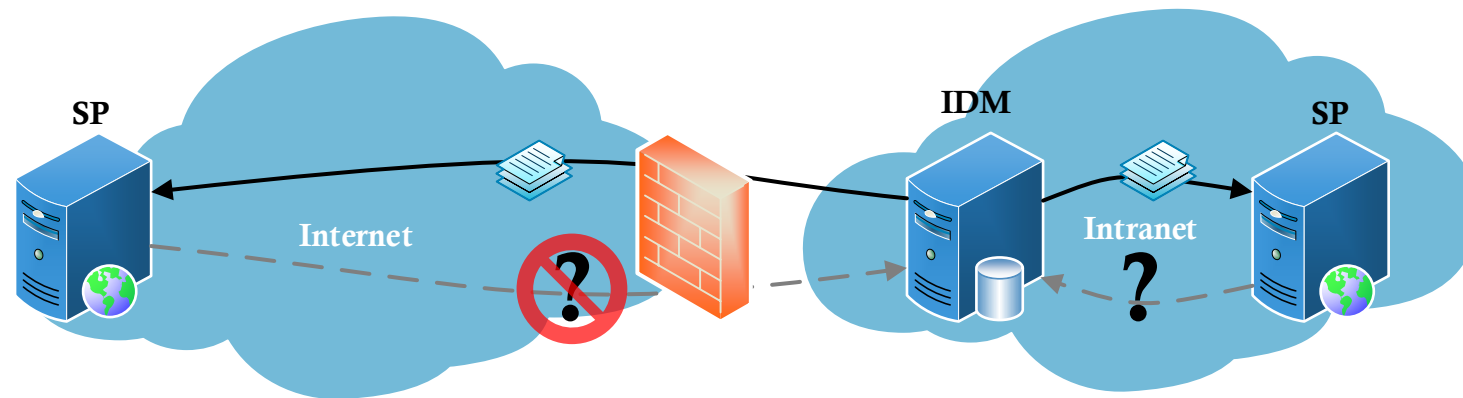
- Webdienste erhalten beim Login des Nutzers nur den aktuellen Zustand
  - Keine Rückmeldung bei Ausscheiden des Nutzers
  - Keine Synchronisierung der Nutzerdaten





# Ausgangslage und Motivation

- Keine Abfrage-Möglichkeit / andere Schnittstelle zum ursprünglichen Quellsystem
  - Zu komplex
  - Bei externen Diensten nicht erwünscht / statthaft



# Ausgangslage und Motivation

- Verschiedene Inselfösungen beim Thema Deprovisionierung
- Steigendes Interesse der SPs an einer verlässlichen / einheitlichen Abfragemöglichkeit
- Datenschutz verlangt zunehmend die Löschung von Nutzerdaten (DSGVO)
- Keine fertige Out-of-the-Box Lösung durch Shibboleth selbst



**=> Es bedarf einer generellen Best-Practice-Lösung!**

# Lösungsansätze

# Lösungsansätze

## 1) Sperren und Löschen der Nutzer nach definierter Inaktivität

	Nutzer	SP	IdP / IDM
Positiv	- ?	<ul style="list-style-type: none"><li>- Einfache Umsetzung</li><li>- Keine weiteren Abhängigkeiten</li></ul>	- Keine Anpassung notwendig
Negativ	<ul style="list-style-type: none"><li>- Aufwand durch regelmäßiges Login</li><li>- Unterschiedliche Löschezitpunkte je SP</li></ul>	<ul style="list-style-type: none"><li>- Frust der Nutzer bei ungewollter Löschung</li><li>- Keine aktive Abfragemöglichkeit des IdP / IDM</li></ul>	- ?

# Lösungsansätze

2) Abfrage des Quell-Systems (IDM) über eine weitere Schnittstelle

	Nutzer	SP	IdP / IDM
Positiv	<ul style="list-style-type: none"><li>- Wird nur gesperrt / gelöscht wenn wirklich nötig</li><li>- Einheitlicher / fest definierter Löschzeitpunkt</li></ul>	<ul style="list-style-type: none"><li>- Aktive Abfragemöglichkeit des IdP / IDM</li><li>- Weniger Frust-Mails der Nutzer</li></ul>	<ul style="list-style-type: none"><li>- ?</li></ul>
Negativ	<ul style="list-style-type: none"><li>- ?</li></ul>	<ul style="list-style-type: none"><li>- Großer Aufwand, teils komplex</li><li>- Keine einheitliche Lösung / Herangehensweise möglich</li><li>- Bei externen SPs nicht gewünscht</li></ul>	<ul style="list-style-type: none"><li>- Anpassungen am IDM nötig</li></ul>

# Lösungsansätze

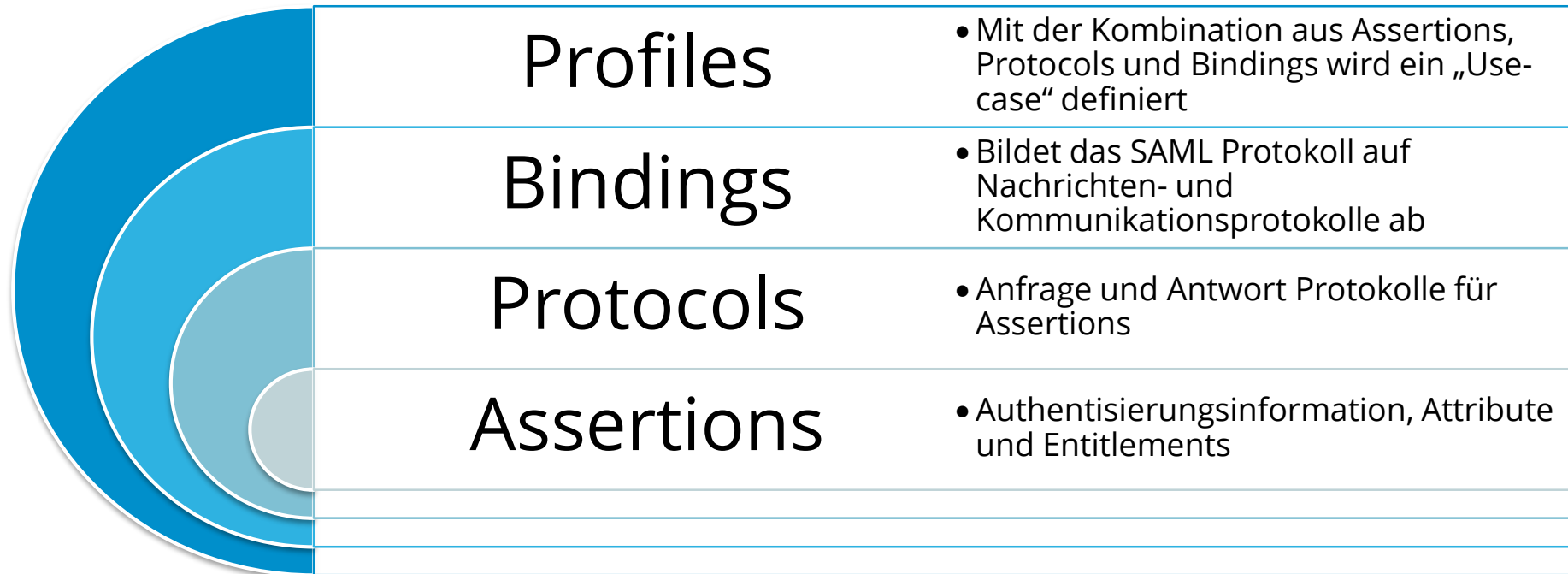
## 3) Abfrage des IdPs via Attribute-Query

	Nutzer	SP	IdP / IDM
Positiv	<ul style="list-style-type: none"><li>- Wird nur gesperrt / gelöscht wenn wirklich nötig</li><li>- Einheitlicher / fest definierter Löschzeitpunkt</li></ul>	<ul style="list-style-type: none"><li>- Aktive Abfragemöglichkeit des IdP / IDM</li><li>- Weniger Frust-Mails der Nutzer</li><li>- Nutzung einer bereits existierenden Schnittstelle</li><li>- Einheitliche Lösung / Herangehensweise</li><li>- Auch für externe SPs nutzbar</li></ul>	<ul style="list-style-type: none"><li>- Nutzung einer bereits existierenden Schnittstelle</li></ul>
Negativ	<ul style="list-style-type: none"><li>- ?</li></ul>	<ul style="list-style-type: none"><li>- ?</li></ul>	<ul style="list-style-type: none"><li>- Anpassungen am IdP / IDM nötig</li></ul>

=> **Entscheidung für Attribute-Query!** (Vorteile überwiegen, Aufwand ist überschaubar)

# SAML Attribute Query

# Auffrischung SAML



## Authentication Context

- Definiert Art und Weise der Authentifizierung

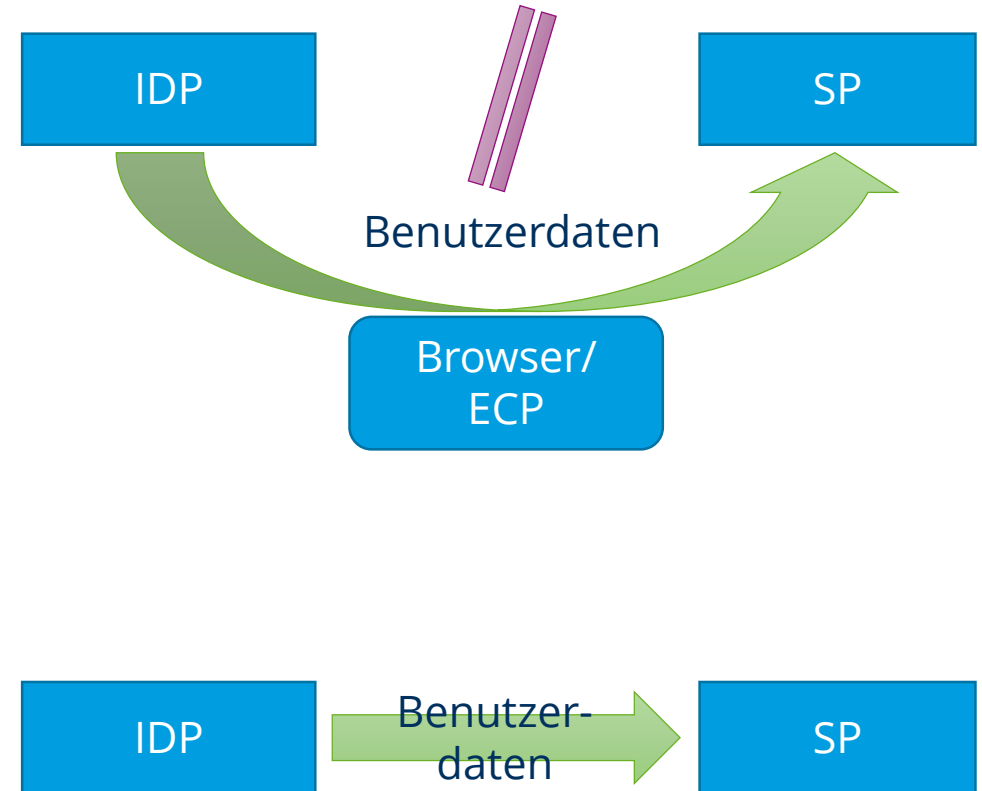
## Metadata

- Konfigurationsdaten für Service- und Identityprovider



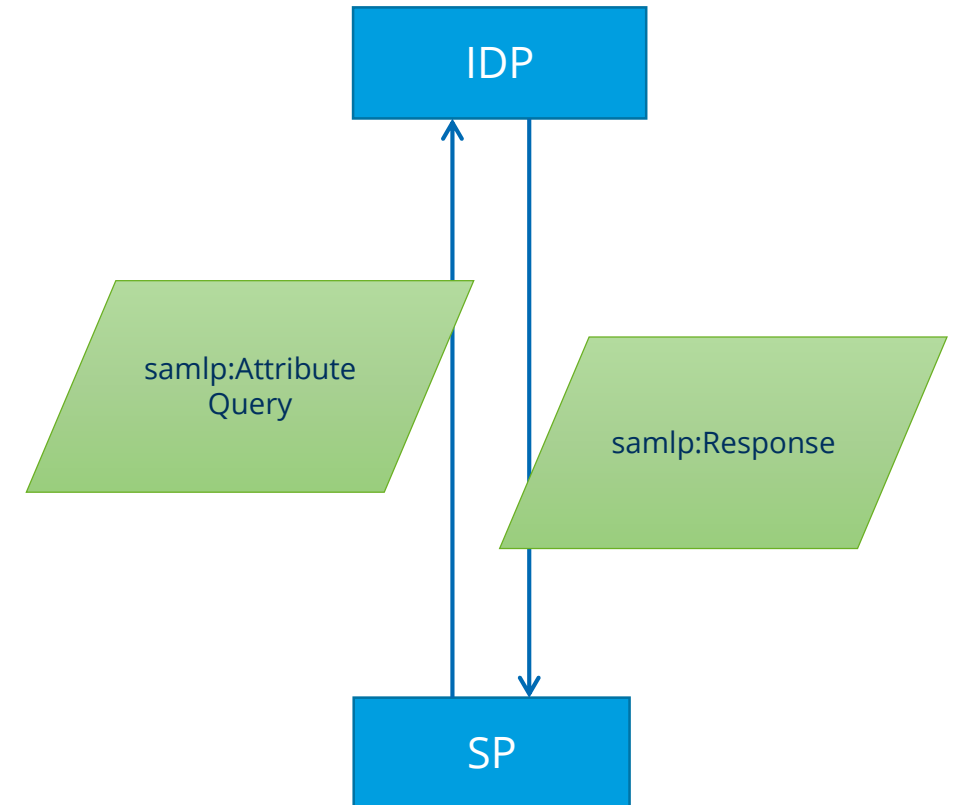
# SAML Profile

- WebSSO Profile
  - Web Single Sign On
  - Meist genutzt
  - Bekannt vom Login im Webbrowser
- ECP Profile
  - Enhanced Client or Proxy Profile
  - Für Fälle, wo kein Browser vorhanden ist
  - Einschränkungen bei den Login Methoden
  - Wenig verbreitet
- **Assertion Query/Request Profile**
  - Mehrere Protokolle sind in diesem Profil zusammengefasst
  - Direkte Kommunikation zwischen SP und IDP
  - Wichtig für Thematik Deprovisionierung -> `saml:AttributeQuery`

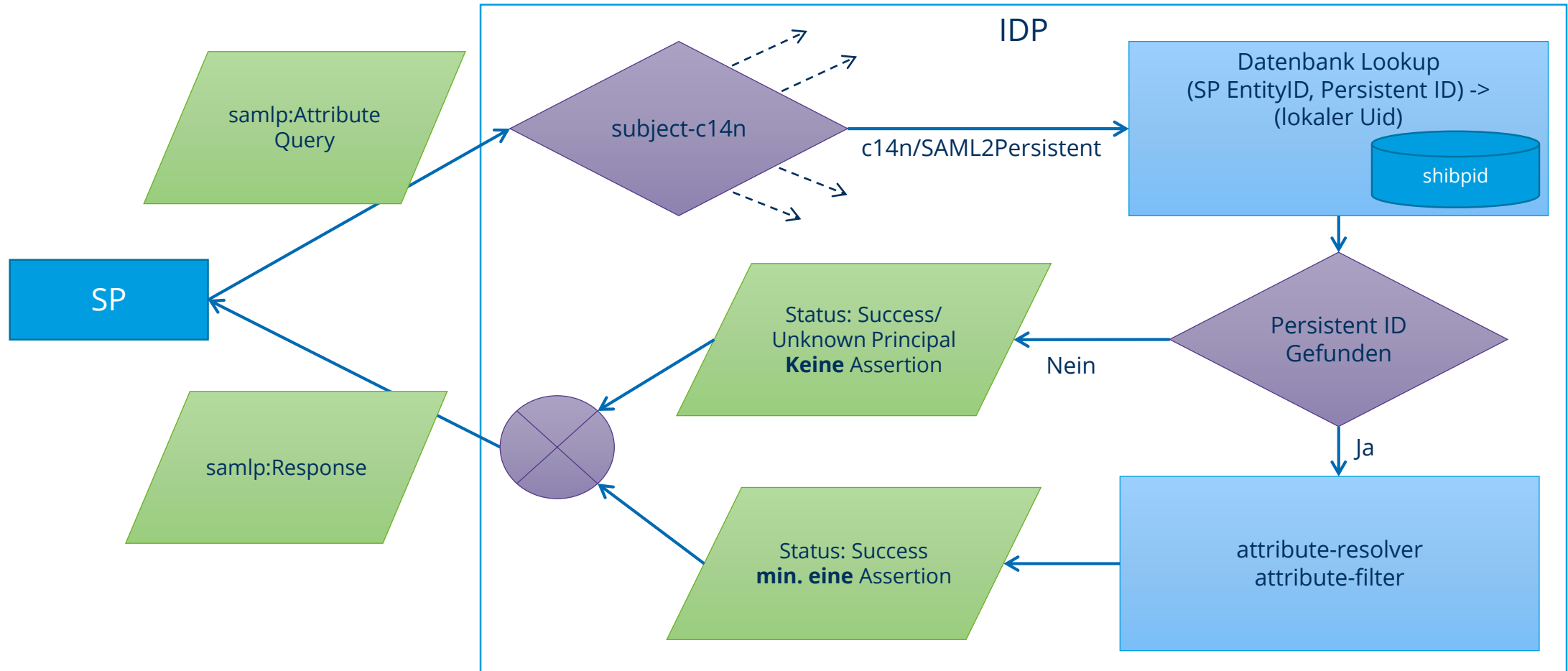


# Assertion Query/Request Profile

- Direkte Kommunikation zwischen SP und IDP
  - Abgesichert über HTTPS (Binding SOAP)
  - Authentisiert über TLS Zertifikat oder XML DSIG
- Benötigt ein Abfrageattribut
  - Typischerweise eine NameID
  - Transient ID (Session bezogen, wird ungültig)
  - Persistent ID (Mit [StoredPersistentIdGenerator](#))
  - Zukunft: Pairwise ID?
  - Im IDP erweiterbar auf andere Attribute
- Antwort IDP ist eine Response
  - Enthält einen Status
  - Enthält ein oder mehrere Assertions oder EncryptedAssertions



# Ablauf im IDP



# Shibboleth SP Handler

## Was passiert beim SP?

- Aufruf einer Handler-URL (/Shibboleth.sso/AttributeResolver)
- Übergabe des Namendefinier des Nutzers (stammt aus Authn-Response)
- Übergabe der entityID des IdPs
- Auslösen des Attribute Resolver Prozess
  - Backchannel-Anfrage an den IdP
  - Abfrage der Attribute des Nutzers

```
<samlp:AttributeQuery
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  ID="_41173d2246af001b8d09596f94130c46"
  IssueInstant="2019-02-21T08:58:22Z"
  Version="2.0">

  <saml:Issuer
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
    https://your.sp.entity.id/shibboleth
  </saml:Issuer>

  <saml:Subject
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
    <saml:NameID
      Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:persistent">
      UxAYyFbltqMj2F1irnEIRt1vg=
    </saml:NameID>
    </saml:Subject>
  </samlp:AttributeQuery>
```

# Shibboleth IDP

## Was passiert beim IdP?

- Aufruf der Attribute-Authority (/idp/profile/SAML2/SOAP/AttributeQuery)
- Auflösen des Namelidentifiers
  - Bei fehlgeschlagener Auflösung:
  - Übertragung einer Response an den SP
  - Status Unknown Principal oder Success
  - **Keine** Assertion
- Ermitteln der Nutzer-Attribute
- Übertragung einer Response an den SP
  - Status Success
  - **Mindestens eine** Assertion

```
<saml2p:Response
  xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol"
  ID="_a2ef652132173aace263fa1320d9798"
  InResponseTo="_41173d2246af001b8d09596f94130c46"
  IssueInstant="2019-02-21T08:58:23.392Z"
  Version="2.0">

  <saml2:Issuer
    xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
    https://your.idp.entity.id/idp/shibboleth
  </saml2:Issuer>
  ...
  <saml2:Assertion>
    ...
    <saml2:Subject>
      <saml2:NameID
        Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">
        _7d9be11d3a23f7740fce83da1f47b374
      </saml2:NameID>
      ...
    </saml2:Subject>
    ...
    <saml2:AttributeStatement>
      <saml2:Attribute
        FriendlyName="mail"
        Name="urn:oid:1.3.6.1.4.1.1466.115.121.26"
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
        <saml2:AttributeValue
          xmlns:xsd="http://www.w3.org/2001/XMLSchema"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:type="xsd:string">
          mustermann@domain.com
        </saml2:AttributeValue>
      </saml2:Attribute>
    </saml2:AttributeStatement>
  </saml2:Assertion>
</saml2p:Response>
```

# Name Identifier

## Hinweise zu Namelidentifiern:

Identifier	Persistent	Revocable	Reassignable	Opaque	Targeted	Portable	Global	Qualifier
Saml2 Transient NameID	No	N/A	N/A	Yes	N/A	N/A	Yes	N/A
Saml2 Persistent NameID	Yes	Yes	No	Yes	Yes	Yes	No	Issuer ID

- transientID
  - Nur innerhalb einzelner Session gültig
  - Somit keine Abfrage außerhalb der Login-Session des Nutzers möglich
- persistentID
  - Über alle Sessions hinweg gleich
  - Abfrage jeder Zeit möglich
- Pairwise ID?

# Voraussetzungen für Queries

# Voraussetzungen für Queries

- persistentId muss am IdP generiert werden
  - Aktivieren des nameID Generator

```
# saml-nameid.xml
<util:list id="shibboleth.SAML2NameIDGenerators">
  <ref bean="shibboleth.SAML2TransientGenerator" />

  <ref bean="shibboleth.SAML2PersistentGenerator" />
</util:list>
```

- Definieren des Quell-Attribut und des Salt  
(für hash-basierte Wert-Erzeugung beim ersten Login)

```
# saml-nameid.properties
idp.persistentId.sourceAttribute = lifetime_fixed_attribute (e.g. uid)
idp.persistentId.salt = %{persid.salt}
```



# Voraussetzungen für Queries

- persistentId muss am IdP gespeichert werden (Rückwärts-Auflösen)
  - Erstellen der Datenbank

```
CREATE TABLE shibpid (  
  localEntity VARCHAR(255) NOT NULL,  
  peerEntity VARCHAR(255) NOT NULL,  
  persistentId VARCHAR(50) NOT NULL,  
  principalName VARCHAR(50) NOT NULL,  
  localId VARCHAR(50) NOT NULL,  
  peerProvidedId VARCHAR(50) NULL,  
  creationDate TIMESTAMP NOT NULL,  
  deactivationDate TIMESTAMP NULL,  
  PRIMARY KEY (localEntity, peerEntity, persistentId)  
);
```

# Voraussetzungen für Queries

- Auswählen des persistentID Generator Typs
- Festlegen des Speicherziels für persistentID

```
# saml-nameid.properties  
idp.persistentId.generator = shibboleth.StoredPersistentIdGenerator  
idp.persistentId.dataSource = shibboleth.MySQLDataSource
```

- Definieren der Bean für den Zugriff auf das Speicherziel

```
# global.xml  
<bean id="shibboleth.MySQLDataSource"  
  class="org.apache.commons.dbcp2.BasicDataSource"  
  p:driverClassName="com.mysql.jdbc.Driver"  
  p:url="jdbc:mysql://127.0.0.1:3306/shibboleth"  
  p:username="shibboleth"  
  p:password="%{mysql.password}"  
  p:maxWaitMillis="15000"  
  p:validationQuery="select 1" />
```

# Voraussetzungen für Queries

- persistentId muss an SP ausgeliefert werden
- Query-Profil muss für SP freigeschalten sein

```
# relying-party.xml
<bean id="shibboleth.DefaultRelyingParty" parent="RelyingParty">
  <property name="profileConfigurations">
    <list>
      <bean parent="SAML2.SSO"
        p:postAuthenticationFlows="#{'attribute-release'}"
        p:nameIDFormatPrecedence="#{{
          'urn:oasis:names:tc:SAML:2.0:nameid-format:persistent',
          'urn:oasis:names:tc:SAML:2.0:nameid-format:transient',
        }}" />
      <ref bean="SAML2.Logout" />
      <ref bean="SAML2.AttributeQuery" />
      <ref bean="SAML2.ArtifactResolution" />
    </list>
  </property>
</bean>
```

# Voraussetzungen für Queries

- persistentId muss von SP gespeichert werden
- Nutzer muss sich einmal am SP angemeldet haben
- SP hat Möglichkeit einen Query zu stellen (Handler)
- SP erreicht den IdP direkt  
(ohne Redirect über den Browser des Nutzer)

# Queries stellen und auswerten

# Queries stellen und auswerten

- Initialen Test am SP mit dem „**resolvertest**“-Skript durchführen (~8 sec)

```
# resolvertest -n +9Blu1I8v96axDXHj01Gmpg36fM= -i https://your.idp.entity.id/idp/shibboleth -saml2 -f\
urn:oasis:names:tc:SAML:2.0:nameid-format:persistent

persistentId: https://your.idp.entity.id/idp/shibboleth! https://your.sp.entity.id/shibboleth!+9Blu1I8v96axDXHj01Gmpg36fM=
givenName: Max
surname: Mustermann
...
```

- Für den Produktiv-Einsatz jedoch „**zu langsam**“ da die ganze Config inklusive Metadaten geladen wird!
- Bei Zugriff zum IdP überprüfen der übermittelten Attribute auf Vollständigkeit mit dem „**resolvertest**“

```
# wget https://your.idp.de/idp/profile/admin/resolvertest?requester= https://your.sp.entity.id/shibboleth &principal=uid-des-test-nutzers

persistentId: https://your.idp.entity.id/idp/shibboleth! https://your.sp.entity.id/shibboleth!+9Blu1I8v96axDXHj01Gmpg36fM=
givenName: Max
surname: Mustermann
...
```

# Queries stellen und auswerten

- Konfiguration zum Stellen „**automatisierter**“ Anfragen vom SP via:
  - [Erweiterung der Gakunin-Föderation](#) (SP  $\geq$  2.5.0 und  $<$  2.6.0)
  - **Attribute Resolver Handler** (SP  $\geq$  2.6.0)

```
# shibboleth2.xml (SP 2.5)
<OutOfProcess>
  <Extensions>
    <Library path="attributequery-handler.so"/>
  </Extensions>
</OutOfProcess>

<InProcess>
  <Extensions>
    <Library path="attributequery-handler-lite.so"/>
  </Extensions>
</InProcess>

<ApplicationDefaults...>
  <Sessions...>
    ...
    <Handler type="AttributeQuery" Location="/AttributeQuery" acl="127.0.0.1 ::1" />
  </Sessions>
  ...
  <AttributeResolver type="Query" subjectMatch="true" />
</ApplicationDefaults>
```

```
# shibboleth2.xml (SP 3.0)
<OutOfProcess>
  <Extensions>
    <Library path="plugins.so"/>
  </Extensions>
</OutOfProcess>

<InProcess>
  <Extensions>
    <Library path="plugins-lite.so"/>
  </Extensions>
</InProcess>

<ApplicationDefaults...>
  <Sessions...>
    ...
    <Handler type="AttributeResolver" Location="/AttributeResolver" acl="127.0.0.1 ::1" />
  </Sessions>
  ...
  <AttributeResolver type="Query" subjectMatch="true" />
</ApplicationDefaults>
```

# Queries stellen und auswerten

- „shibd“ neustarten und mit CURL testen (~0.1 sec) (Faktor 80!!! schneller)

```
# curl --get --insecure "https://localhost/Shibboleth.sso/AttributeResolver"\
--data-urlencode "entityID= https://your.idp.entity.id/idp/shibboleth "\
--data-urlencode "nameId= +9Blu1I8v96axDXHj01Gmpg36fM= "\
--data-urlencode "format=urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"

{
  "persistent-id" : [
    " https://your.idp.entity.id/idp/shibboleth! https://your.sp.entity.id/shibboleth!+9Blu1I8v96axDXHj01Gmpg36fM= "
  ]
  "givenName" : [
    "Max"
  ]
  "surname" : [
    "Mustermann"
  ]
}
```

- **ACHTUNG:** Werte sollten „codiert“ übergeben werden, um Probleme bei **Sonderzeichen** zu vermeiden!
- Rückgabe ist ein **JSON**-Objekt mit einer Liste der Eigenschaften (Attributen) des Nutzers



# Verlässlichkeit von Queries

# Verlässlichkeit von Queries

## Wann können Nutzer nun gefahrlos gelöscht werden?

- Queries liefern in der Regel den normalen Attribut-Satz
- Erste Vermutung: Löschen bei einem **leeren** Ergebnis
- Aber folgende Fallstricke können **fälschlicherweise** zu einem leeren / nicht leeren Ergebnis führen:
  - **Leeres Ergebnis:**
    - persistentId wurde falsch übergeben / existiert auf Seiten des IdP nicht
    - persistentId wird generell beim IdP nicht gespeichert
    - Nutzer ist nicht mehr im LDAP (aber noch im IDM - Synchronisierungsproblem)
    - LDAP-Abfrage / Query ist fehlgeschlagen
  - **Nicht leeres Ergebnis:**
    - Statische Attribute im Resolver liefern immer Werte
    - Nutzer wurde im LDAP nicht entfernt (aber im IDM - Synchronisierungsproblem)
    - Query ist fehlgeschlagen (SP3.x zeigt z.B. im Resultat wenigstens die persistentID)

# Verlässlichkeit von Queries

=> **Expliziter Rückgabewert fürs Löschen erforderlich um Fehler bei der Abfrage auszuschließen!**

- 2 mögliche Lösungswege
- **Variante 1)**
  - Simpel für IdP
  - Setzen des „**deactivationDate**“ in der Datenbank
  - SP muss StatusCode auswerten (aktuell schwieriger)
  - Für z.B. externe SPs
- **Variante 2)**
  - Komplexer für IdP aber detaillierter
  - Ermitteln des Nutzerstatus und Weitergabe im Attribut „**schacUserStatus**“
  - SP muss Attribut auswerten (einfacher)
  - Für z.B. interne SPs
- **Parallelbetrieb** der Varianten möglich, indem man für einzelne SPs NICHT das deactivationDate setzt

# Verlässlichkeit von Queries

## Variante 1)

- Es muss in der Tabelle „**shibpid**“ ein Wert in der Spalte „**deactivationDate**“ gesetzt werden
  - Der Wert ist bisher egal (es wird nur geprüft ob „is NULL“)
  - Feature-Request wurde gestellt (DB-Schema anpassen zu „bool“ oder Wert auswerten)
- Die DB-Einträge dürfen nicht gelöscht werden (auch nicht, wenn der zugehörige Account gelöscht wurde)
- **Rückgabe:** statt einem Attribut muss der SAML2 Statuscode (und die Assertion) ausgewertet werden

```
<saml2p:Response>
...
<saml2p:Status>
  <saml2p:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
</saml2p:Status>
...
<saml2:Assertion>
...
</saml2:Assertion>
</saml2p:Response>
```

```
<saml2p:Response>
...
<saml2p:Status>
  <saml2p:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal"/>
</saml2p:Status>
...
</saml2p:Response>
```

# Verlässlichkeit von Queries

## Variante 1)

- **Problem:**

- StatusCode wird NICHT im JSON Objekt ausgegeben (lediglich in Logfile einsehbar)
- Bei falschem URL-Encoding bekommt der SP zu einem noch existierenden Account auch die "Unknown-Principal"-Meldung
- Der SP muss das Verfahren mit einem noch existierenden Account testen

**=> Aktuell bei Benutzung des Attribute Resolver Handler eines „normalen“ SP nicht nutzbar!**

**=> Feature-Request wird gestellt, so dass mehr Informationen ausgegeben werden!**

# Verlässlichkeit von Queries

## Variante 2)

- Einführung des Attributes [schacUserStatus](#) mit folgenden Werten:
    - Active\*
    - Blocked\*
    - Inactive\*
    - Deleted
- \* optional (zur feineren Unterscheidung bei noch existierenden Accounts)

Descr: Used to store a set of status of a person as user of services  
OID: urn:oid:1.3.6.1.4.1.25178.1.2.19  
Format: urn:schac:userStatus:<country-code>:<domain>:<iNSS>  
- The <country-code> must be a valid two-letter ISO 3166 country code identifier  
- <domain> is the institution domain name according to RFC 1035  
- <iNSS> is a Namespace Specific String as defined in RFC 2141 but case insensitive  
Examples:  
urn:schac:userStatus:de:einrichtung.de:affiliation:active  
urn:schac:userStatus:de:einrichtung.de:affiliation:blocked  
urn:schac:userStatus:de:einrichtung.de:affiliation:inactive  
urn:schac:userStatus:de:einrichtung.de:affiliation:deleted

# Verlässlichkeit von Queries

- Erweiterung des LDAP-Schema
  - Alle Nutzer erhalten „schacUserStatus“ Attribut
  - Anlegen eines Unter-Zweigs „archive“
  - Im Archiv landen alle ausgeschiedenen Nutzer
    - Somit vermeidet man „Leichen“ im produktiven Nutzer-Zweig
    - Im Archiv genügt es für „deleted“ Nutzer die „uid“ und den „schacUserStatus“ aufzuheben

ou=users,dc=einrichtung,dc=de

uid=user1,ou=users,dc=einrichtung,dc=de

uid: user1  
givenName: Max  
sn: Mustermann  
schacUserStatus: active

uid=user2,ou=users,dc=einrichtung,dc=de

uid: user2  
givenName: Tim  
sn: Maurer  
schacUserStatus: blocked

ou=archive,dc=einrichtung,dc=de

ou=users,ou=archive,dc=einrichtung,dc=de

uid=user3,ou=users,ou=archive,dc=einrichtung,dc=de

uid: user3  
givenName: Hans  
sn: Meier  
schacUserStatus: inactive

uid=user4,ou=users,ou=archive,dc=einrichtung,dc=de

uid: user4  
schacUserStatus: deleted

# Verlässlichkeit von Queries

- AttributeDefinition erstellen

```
# attribute-resolver.xml
<AttributeDefinition xsi:type="Simple" id="baseSchacUserStatus" dependencyOnly="true">
  <InputDataConnector ref="statusUserLDAP" attributeNames="schacUserStatus" />
  <InputDataConnector ref="statusArchiveLDAP" attributeNames="schacUserStatus" />
</AttributeDefinition>

<AttributeDefinition xsi:type="Template" id="schacUserStatus">
  <InputAttributeDefinition ref="baseSchacUserStatus" />
  <DisplayName xml:lang="de">Benutzerstatus</DisplayName>
  <DisplayName xml:lang="en">Userstatus</DisplayName>
  <DisplayDescription xml:lang="de">Status eines Benutzers für einen Dienst</DisplayDescription>
  <DisplayDescription xml:lang="en">set of status of a person as user of services</DisplayDescription>
  <AttributeEncoder xsi:type="SAML1String" name="urn:mace:terena.org:schac:attribut-def:schacUserStatus" />
  <AttributeEncoder xsi:type="SAML2String" name="urn:oid:1.3.6.1.4.1.25178.1.2.19" friendlyName="schacUserStatus" />
  <Template>
    <![CDATA[
      urn:schac:userStatus:de:einrichtung.de:${baseSchacUserStatus}
    ]]>
  </Template>
</AttributeDefinition>
```



# Verlässlichkeit von Queries

- DataConnector erstellen

```
# attribute-resolver.xml
<DataConnector id="statusArchiveLDAP" xsi:type="LDAPDirectory"
  ldapURL="%{idp.attribute.resolver.LDAP.ldapURL}"
  baseDN="%{idp.attribute.resolver.LDAP.archiveDN}"
  principal="%{idp.attribute.resolver.LDAP.bindDN}"
  principalCredential="%{idp.attribute.resolver.LDAP.bindDNCredential}"
  trustFile="%{idp.attribute.resolver.LDAP.trustCertificates}"
  searchScope="ONELEVEL">
  <FilterTemplate>
    <![CDATA[
      %{idp.attribute.resolver.LDAP.searchFilter}
    ]]>
  </FilterTemplate>
  <ReturnAttributes>schacUserStatus</ReturnAttributes>
</DataConnector>
```

- AttributeFilter erstellen

```
# attribute-filter.xml
<AttributeRule attributeID="schacUserStatus" permitAny="true" />
```

# Verlässlichkeit von Queries

- Ergebnis des Query

```
# curl --get --insecure "https://localhost/Shibboleth.sso/AttributeResolver"\
--data-urlencode "entityID= https://your.idp.entity.id/idp/shibboleth "\
--data-urlencode "nameId= +9Blu1I8v96axDXHj01Gmpg36fM= "\
--data-urlencode "format=urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"

{
  "persistent-id" : [
    " https://your.idp.entity.id/idp/shibboleth! https://your.sp.entity.id/shibboleth!+9Blu1I8v96axDXHj01Gmpg36fM= "
  ]
  "givenName" : [
    "Max"
  ]
  "surname" : [
    "Mustermann"
  ]
  "schacUserStatus" : [
    "urn:schac:userStatus:de:tu-dresden.de:affiliation:active"
  ]
}
```

**=> Bei schacUserStatus „deleted“ kann nun verlässlich gelöscht werden! :-)**

# Verlässlichkeit von Queries

- Optional Einführung eines weiteren zusätzlichen Attributes statusChanged mit Datum des letzten Change
  - Ermöglicht bessere Auswertung SP-seitig bezügl. Lösch-Fristen

```
ou=archive,dc=einrichtung,dc=de
```

```
ou=users,ou=archive,dc=einrichtung,dc=de
```

```
uid=user3,ou=users,ou=archive,dc=einrichtung,dc=de
```

```
uid: user3
```

```
givenName: Hans
```

```
sn: Meier
```

```
schacUserStatus: inactive
```

```
statusChanged: 20190223
```

```
uid=user4,ou=users,ou=archive,dc=einrichtung,dc=de
```

```
uid: user4
```

```
schacUserStatus: deleted
```

```
statusChanged: 20190118
```

# Datenschutzaspekte

# Datenschutzaspekte

- Standardmäßig kann **jeder** SP Queries stellen
- In der **RelyingParty** kann dies mit einer **ActivationCondition** für ausgewählte SPs eingeschränkt werden
- **ACHTUNG:** nur möglich, wenn kein SAML1 SP mehr bedient werden muss

```
# relying-party.xml
<bean id="shibboleth.DefaultRelyingParty" parent="RelyingParty">
  <property name="profileConfigurations">
    <list>
      <bean parent="SAML2.SSO"
        p:postAuthenticationFlows="#{'attribute-release'}"
        p:nameIDFormatPrecedence="#{
          'urn:oasis:names:tc:SAML:2.0:nameid-format:transient'
        }" />
      <ref bean="SAML2.Logout" />
      <ref bean="SAML2.ArtifactResolution" />
    </list>
  </property>
</bean>
```

```
# relying-party.xml
<util:list id="shibboleth.RelyingPartyOverrides">
  <bean parent="RelyingParty" p:activationCondition-ref="SP-consumes-persistentId">
    <property name="profileConfigurations">
      <list>
        <bean parent="SAML2.SSO"
          p:postAuthenticationFlows="#{'attribute-release'}"
          p:nameIDFormatPrecedence="#{
            'urn:oasis:names:tc:SAML:2.0:nameid-format:persistent',
            'urn:oasis:names:tc:SAML:2.0:nameid-format:transient'
          }" />
        <ref bean="SAML2.Logout" />
        <ref bean="SAML2.AttributeQuery" />
        <ref bean="SAML2.ArtifactResolution" />
      </list>
    </property>
  </bean>
</util:list>
```

# Datenschutzaspekte

- Erstellen der ActivationCondition

```
# activation-conditions.xml
<bean id="SP-consumes-persistentId" parent="shibboleth.Conditions.RelyingPartyId">
  <constructor-arg name="candidates">
    <list>
      <value>https://your.sp.entity.id/shibboleth</value>
      <value>https://another.sp.entity.id/shibboleth</value>
    </list>
  </constructor-arg>
</bean>
```

```
# services.xml
<util:list id="shibboleth.RelyingPartyResolverResources">
  <value>{%idp.home}/conf/relying-party.xml</value>
  <value>{%idp.home}/conf/credentials.xml</value>
  <value>{%idp.home}/conf/activation-conditions.xml</value>
  <value>{%idp.home}/system/conf/relying-party-system.xml</value>
</util:list>

<util:list id="shibboleth.AttributeResolverResources">
  <value>{%idp.home}/conf/attribute-resolver.xml</value>
  <value>{%idp.home}/conf/activation-conditions.xml</value>
</util:list>
```

# Datenschutzaspekte

- Standardmäßig werden **alle** freigegebenen Attribute per Query ausgeliefert
- Will man per Query eine **Synchronisierung** vornehmen ist nichts weiter zu tun
  - Bitte klären Sie dieses Vorgehen jedoch vorab mit Ihrem Datenschutzbeauftragten
  - Die Übermittlung personenbezogener Daten geschieht **ohne** Wissen des Nutzers
- Für eine reine **Deprovisionierung** sollte nur **schacUserStatus** per Query ausgeliefert werden
- Mit dem **Blockieren** der anderen Attribute erreicht man u.a.:
  - Minimierung der openLDAP Abfragen
  - Lastminimierung bei scripted Attributes
  - Datenschutz + Datensparsamkeit!
- Statt Blockieren per Conditions alternativ Aktivierung der Berücksichtigung des User Consent bei Queries

# Datenschutzaspekte

- Herunterladen folgender JAR-File [idp-predicate-impl-1.0.0.jar](#) nach „edit-webapp/WEB-INF/lib/“

- Anschließend den IdP neu bauen

```
# ./bin/build.sh
```

- Predicate nutzen um benötigte ActivationConditions zu erstellen

```
# activation-conditions.xml
<bean id="no-query" parent="shibboleth.Conditions.NOT">
  <constructor-arg>
    <list>
      <ref bean="RequestedAttributeQueryProfileIdPredicate" />
    </list>
  </constructor-arg>
</bean>

<bean id="SP-consumes-schacUserStatus" parent="shibboleth.Conditions.AND">
  <constructor-arg>
    <list>
      <ref bean="RequestedAttributeQueryProfileIdPredicate" />
      <ref bean="SP-consumes-persistentId" />
    </list>
  </constructor-arg>
</bean>
```



# Datenschutzaspekte

- Einschränken der DataConnectoren und der Attribute

```
<AttributeDefinition xsi:type="Simple" id="uid">
  <InputDataConnector ref="myLDAP" attributeNames="uid" />
  <!--...-->
</AttributeDefinition>

<DataConnector id="myLDAP" xsi:type="LDAPDirectory" activationConditionRef="no-query"
  <!--...-->
</DataConnector>

<DataConnector id="statusArchiveLDAP" xsi:type="LDAPDirectory" activationConditionRef="SP-consumes-schacUserStatus"
  <!--...-->
</DataConnector>

<AttributeDefinition xsi:type="ScriptedAttribute" id="specialAttribute" activationConditionRef="SP-consumes-specialAttribute">
  <!--...-->
</AttributeDefinition>
```

- Jeder DataConnector erhält eine Condition (um dessen Ausführung zu steuern)
- Attribute die nur von DataConnectoren abhängen brauchen keine weitere Condition
- Attribute die von keinem Connector oder von Attributen ohne Einschränkung abhängen benötigen eine separate Condition

# Wann und wie oft Queries stellen

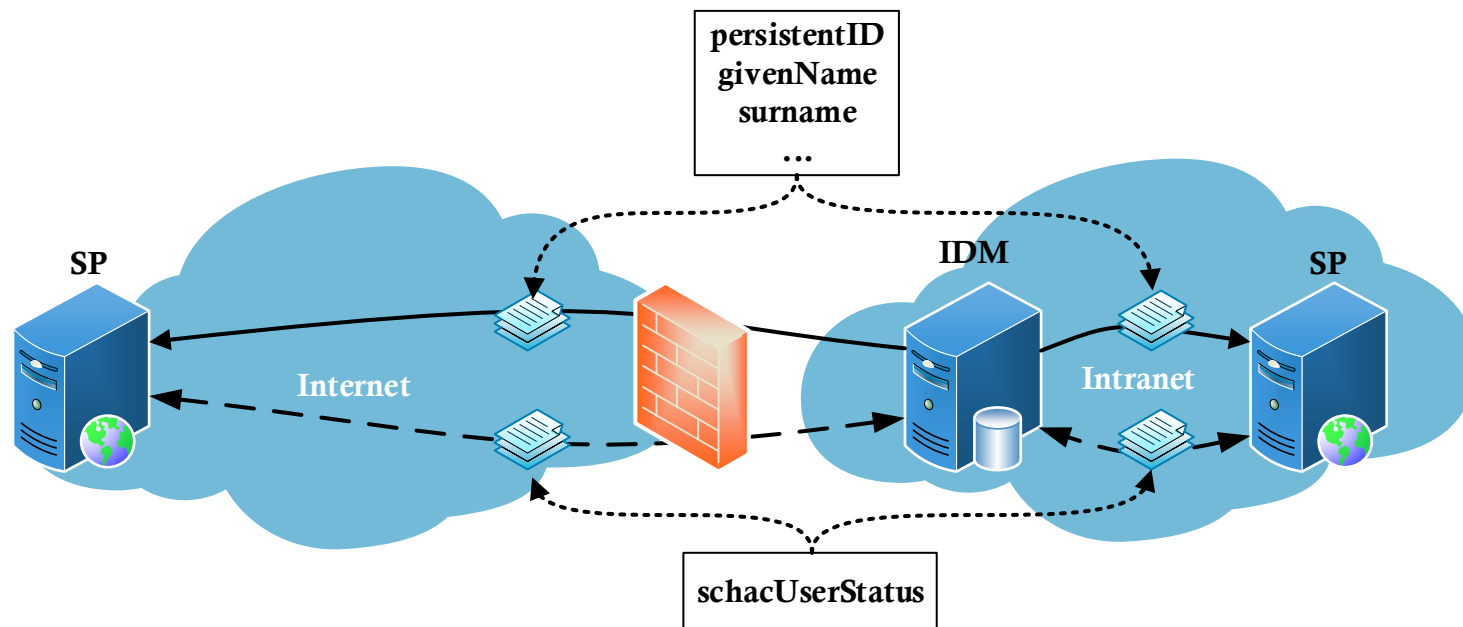
# Wann und wie oft Queries stellen

- Der **SP ist verantwortlich**, dass der IdP während seines Betriebes **nicht gestört** wird
- Vermeidung von unangemessen hoher Anzahl an Anfragen (**DOS-Attacke**)
  - Es ist nicht notwendig jeden Nutzer jeden Tag zu prüfen
  - Anlegen einer Cachefile mit Zeitstempel der letzten erfolgreichen Prüfung
- Abfrage des IdP z.B. nur wenn:
  - Letzter Login > x Tage
  - Letzte Prüfung > x Tage
- Am Besten **kleine Pausen** zwischen einzelnen Anfragen lassen
  - Gefahr durch Fail2Ban etc. ausgesperrt zu werden

# Fazit

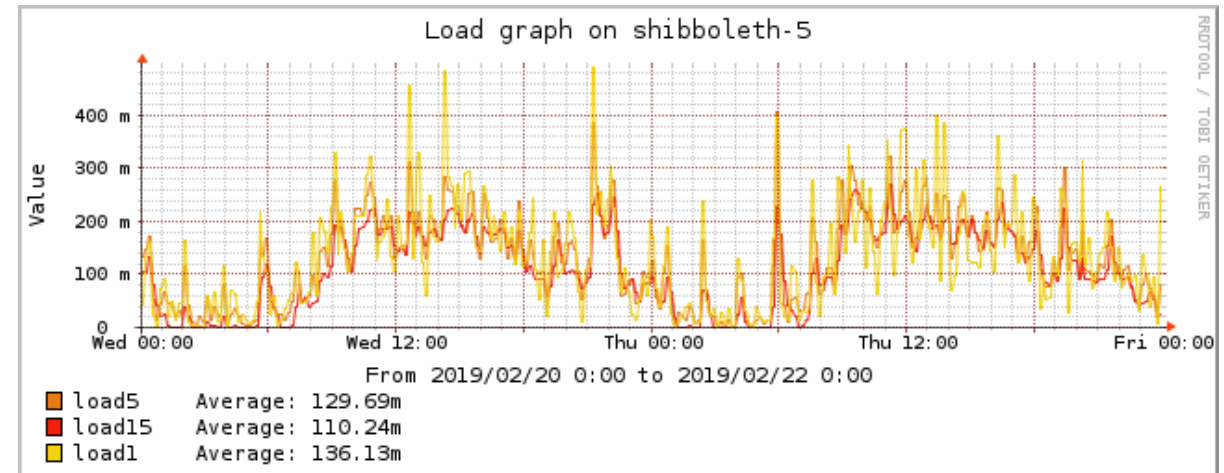
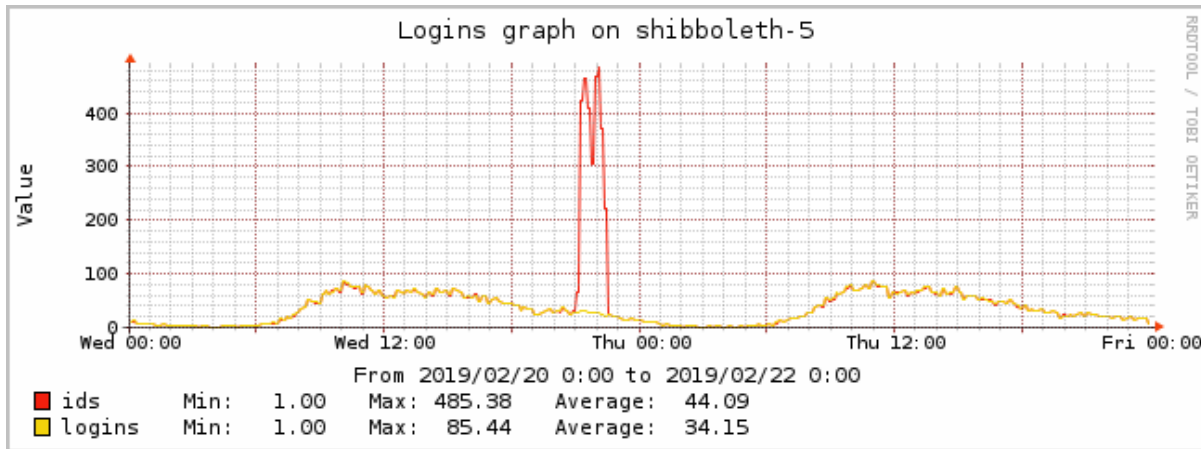
# Fazit

- Schaffung einer einheitlichen Schnittstelle und Namenskonvention zur Deprovisionierung
- Verbesserter Datenschutz durch Einschränkung von Queries
  - Nur von bestimmten SPs möglich
  - Nur bestimmte Daten werden übertragen



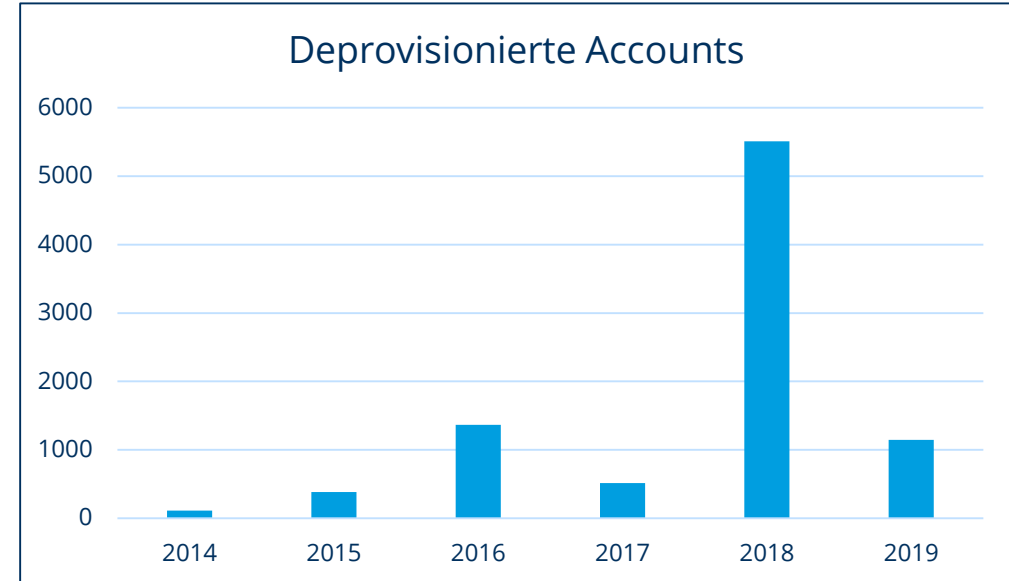
# Fazit TUD

- Anzahl bereits gelöschter Datensätze ~40.000 seit Mai '18
  - Abfrage von ~7.000 Nutzern pro Woche
  - Löschen von ~100 Nutzern pro Woche
  - Dauer eines Durchlauf ~1 Stunde (mit Pausen zw. einzelnen Anfragen)
- Kaum mehr Last auf System (Abfrage vergleichbar mit normalen Login)



# Fazit bwldm

- Deprovisionierung mit Variante 1
  - Nicht besonders vereinbart bislang
- 12 unterschiedliche IDPs mit >100 deprovisionierten Accounts
- Weitere 10 IDPs mit >10 und <100
- Weitere 7 IDPs mit <10
- Software reg-app nutzt direkt OpenSAML/java
  - Dadurch unbewusst Probleme mit dem Shibboleth SP vermieden



# Links und Dokumentation



# Links und Dokumentation

- [https://www.switch.ch/aai/support/presentations/opcom-201105/AAI-OpCom-Message\\_level\\_security.pdf](https://www.switch.ch/aai/support/presentations/opcom-201105/AAI-OpCom-Message_level_security.pdf)
- <https://bitbucket.org/PEOFIAMP/shibsp-plugin-attributequery-handler/>
- <https://wiki.shibboleth.net/confluence/display/SP3/Attribute+Resolver+Handler>
- <https://wiki.shibboleth.net/confluence/display/SP3/QueryAttributeResolver>
- <https://wiki.shibboleth.net/confluence/display/CONCEPT/NameIdentifiers>
- <https://wiki.shibboleth.net/confluence/display/IDP30/PersistentNameIDGenerationConfiguration>
- <https://wiki.shibboleth.net/confluence/display/IDP30/SecurityAndNetworking#SecurityAndNetworking-PortsandConnectors>
- <https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPAAccountChecking>
- <https://wiki.shibboleth.net/confluence/display/IDP30/AttributeDefinitionConfiguration>
- <https://wiki.shibboleth.net/confluence/display/IDP30/ActivationConditions>
- <https://wiki.refeds.org/download/attachments/1606048/SCHAC%2B1.5.0.pdf?version=3&modificationDate=1429195142624&api=v2>
- <https://doku.tid.dfn.de/de:shibidp3userdepro>
- <https://cloudstore.zih.tu-dresden.de/index.php/s/RrhJkwDwZOQoKRm>

# Danke für Ihre Aufmerksamkeit! :-)