Shibboleth/Federation Operator Tutorial

TIIME Workshop 2018

Speaker: David Hübner,
DAASI International

Date: 6 Feb 2018

Agenda

1. Welcome and Introduction to the Workshop
2. Introduction to Shibboleth
3. Federation Tools from Shibboleth
4. PyFF as an Alternative
5. Shibboleth IdP Hosting
6. Plugin Interfaces of Shibboleth IdP V3
7. Hands-On Session

DAASI International

Introduction to Shibboleth

DAASI International

Shibboleth Project

*„Shibboleth is a **standards based**, **open source** software package for **web single sign-on across or within organizational boundaries**.“*

- Began as an Internet2 activity in 2000
- SAML (2.0)
- Maintained by the Shibboleth Consortium
- Widespread useage in research & education

DAASI International

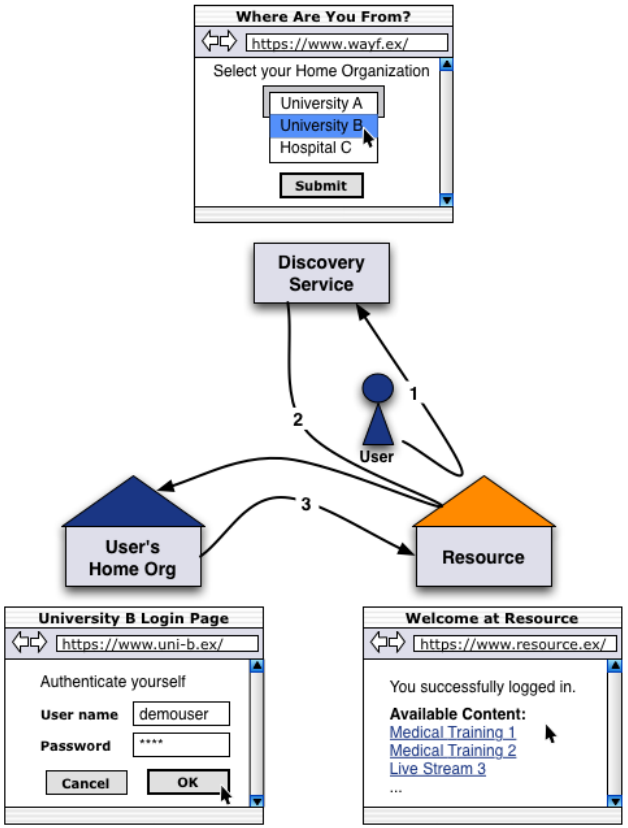
Shibboleth Project

- Several products
  - Identity Provider
  - Service Provider
  - OpenSAML Libraries (C++ & Java)
  - Metadata Aggregator
  - Centralized Discovery Service
  - Embedded Discovery Service

DAASI International

Short introduction to Web SSO...

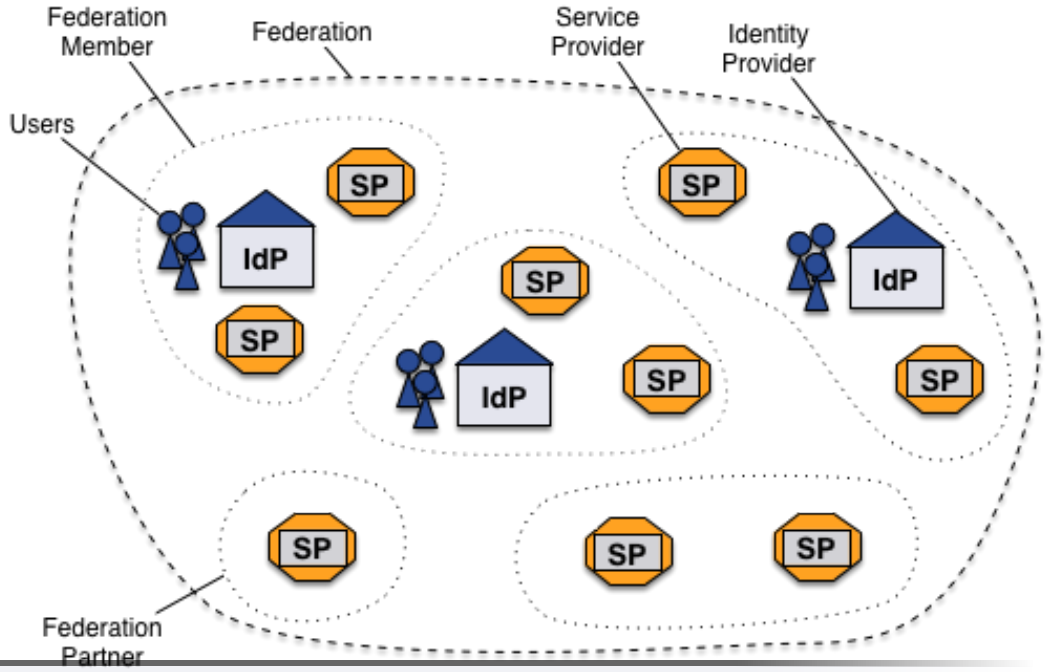
- Most important use case: Web Browser SSO
- Identity Provider (IdP) authenticates users and issues assertions
- Service Provider (SP) relies on assertions from IdO and uses this information to control access to protected services
- Discovery service (DS) lets users choose their home organization (and therefore IdP)



© Switch.ch

… and federations

- Federations as collection of organizations
- Federation operator as trusted third party
- Scalable way to allow SSO across organizational boundaries
- SAML Metadata to connect entities



DAASI International

Shibboleth IdP

- Implements the Identity Provider (IdP) Component
- AuthN of users from an organisation happens centrally at its IdP
- Assertions (SAML) to convey identity informations (AuthN result + attributes) to Relying Parties (RP)
- Shibboleth IdP is a Java application
- Protocol Support includes SAML 1.1, most SAML 2.0 profiles, CAS
- Based on Spring for configuration & wiring of components
- Current release: V3.3.2

DAASI International

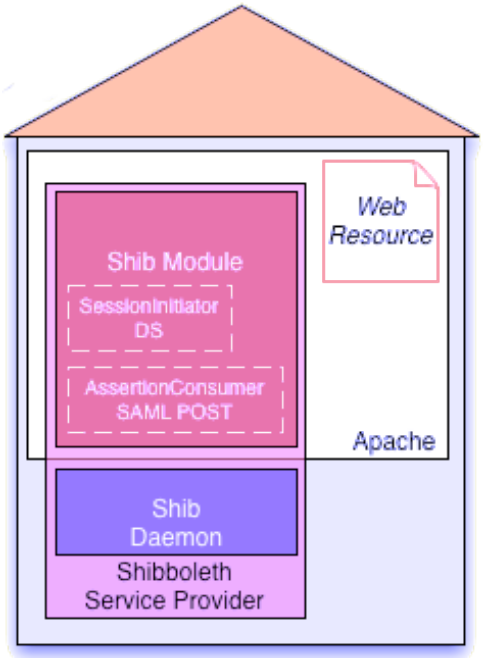
Shibboleth IdP

- Higly configurable
  - Authentication sources
  - Attribute Resolution
  - Attribute Filtering
  - Metadata Management
  - Per-RP Configuration
  - Policies
  - Backend / Storage

DAASI International

Shibboleth SP

- Implements the SP component
- There are two parts
  - C++ daemon (shibd)
    - Keeps states, evaluates protocol mesages
  - Webserver module (mod_shib)
    - Protects Locations/Directories, defines Access Rules
- Current release: V2.6
- Binaries for Windows, OSX, RPM-based Linux
- Switch Repository for Ubuntu/Debian



(c) Switch

Shibboleth SP Overview

- General configuration in /etc/shibboleth/shibboleth2.xml

- Applications as „containers“ to group common configuration
- Configuration of <Session> properties (e.g. duration)
  - <SessionInitiator> define how the SP should get a session (some shorthands for e.g. <SSO> and <Logout> exist)
  - <Handler> provide various endpoints

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

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

    <SSO discoveryProtocol="SAMLDS"
        discoveryURL="https://example.org/ds">
      SAML2 SAML1
    </SSO>

    <Handler type="MetadataGenerator" Location="/Metadata"
        signing="false"/>

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

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

  </Sessions>
```

Shibboleth SP Overview

- General configuration in /etc/shibboleth/shibboleth2.xml

- Load metadata (static or dynamic)
- Configure how the SP should provide attributes to applications
- Keys and Certificates
- Possible ApplicationOverrides to run multiple logical SPs with one installation

```
<MetadataProvider type="XML" validate="true"
    uri="http://example.org/fed-metadata.xml"
    backingFilePath="fed-metadata.xml" reloadInterval="7200">

    <MetadataFilter type="Signature"
        certificate="md_federation.pem"/>
</MetadataProvider>

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

<AttributeFilter type="XML" validate="true"
    path="attribute-policy.xml"/>

<CredentialResolver type="File"
    key="sp-key.pem" certificate="sp-cert.pem"/>

<!--<ApplicationOverride /> -->

</ApplicationDefaults>
```

DAASI International

Processing attributes

- /etc/shibboleth/attribute-map.xml to extract attributes and provide them internally
- Simple mapping from incoming name to internal id

```
<Attribute name="urn:oid:0.9.2342.19200300.100.1.3" id="mail"/>
```

- You can then use them as apache environment variables in protected applications, e.g. $_SERVER[‘mail‘]
- /etc/shibboleth/attribute-policy.xml to define processing policies
- Default configuration includes e.g. scope checking of ePPN, ePSA etc.

DAASI International

Processing attributes

- Attributes can be changed various times on their way to the application
  - IdP resolves attributes
  - IdP filters attributes on a per-RP basis
  - SP accepts or rejects attributes…
  - ...filters them accoring to policies…
  - ...and maps them to internal variables
  - Webserver provides variables as environment variables
- Be aware that names might change and attributes might get lost

DAASI International

Protecting Applications

- Three common options
  - Directly in the webserver via Apache Access Rules

```
<Location /application>
  AuthType shibboleth
  ShibRequestSetting requireSession 1
  require shib-session
</Location>
```

  - In shibboleth2.xml configuration via SPXMLAccessControl
  - Use application („lazy session“ or „passive protection“)

```
<Location /lazy>
  AuthType shibboleth
  ShibRequestSetting requireSession 0
  require Shibboleth
</Location>
```


DAASI International

Protecting Applications

- Usually Apache Access Rules is the easiest approach
- Require shib-attr allows to decide based on attributes

```
<Location /application>
  AuthType shibboleth
  ShibRequestSetting requireSession 1
  require shib-session
  require shib-attr affiliation student
</Location>
```

DAASI International

Shibboleth DS

- Two pieces still missing
- SP uses Discovery Service (DS) to allow users to choose their IdP

```
<SSO discoveryProtocol="SAMLDS"
    discoveryURL="https://example.org/ds">
  SAML2 SAML1
</SSO>
```

- Shibboleth *did* provide Centralized Discovery Service (CDS)
  - Central as in „not on every SP“
  - The federation operator might run this
  - Shibboleth CDS is no longer maintained
- Shibboleth does provide Embedded Discovery Service (EDS)
  - Embedded as in „runs on every SP host“
  - Can be combined with Metadata management of Shibboleth SP

DAASI International

Shibboleth Metadata Aggregator

- And the other missing piece:

```
<MetadataProvider type="XML" validate="true"
    uri="http://example.org/fed-metadata.xml"
    backingFilePath="fed-metadata.xml" reloadInterval="7200">

    <MetadataFilter type="Signature"
        certificate="md_federation.pem"/>
</MetadataProvider>
```

- Metadata Aggregation means:
  - –

DAASI International

Thanks!

https://www.daasi.de

DAASI International