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

What‘s possible with Shibboleth IdP

What is this session about?

- We will look at various parts of Shibboleth IdP and see what‘s possible beyond just obvious configuration
- This session is about
  - More advanced configuration
  - Existing extensions that might be worth considering
  - Inspiration what can be done with your own extensions
- This session is **not** about
  - Learning to actually implement this stuff

Some general remarks

- Shibboleth IdP uses Spring to wire together various components
  - In general you can just write your own Java Beans that *do stuff* and just use the existing IdP structure to load them and let them *do stuff*
    - e.g. define a CustomViewContext in global.xml and use it in views
- Shibboleth IdP uses Spring Web Flow to define flows that involve user interaction
  - In general everything in system/ must not be touched
  - Bust usually you can just copy and write your own flow :)
- Most java classes are part of the public API

Authentication

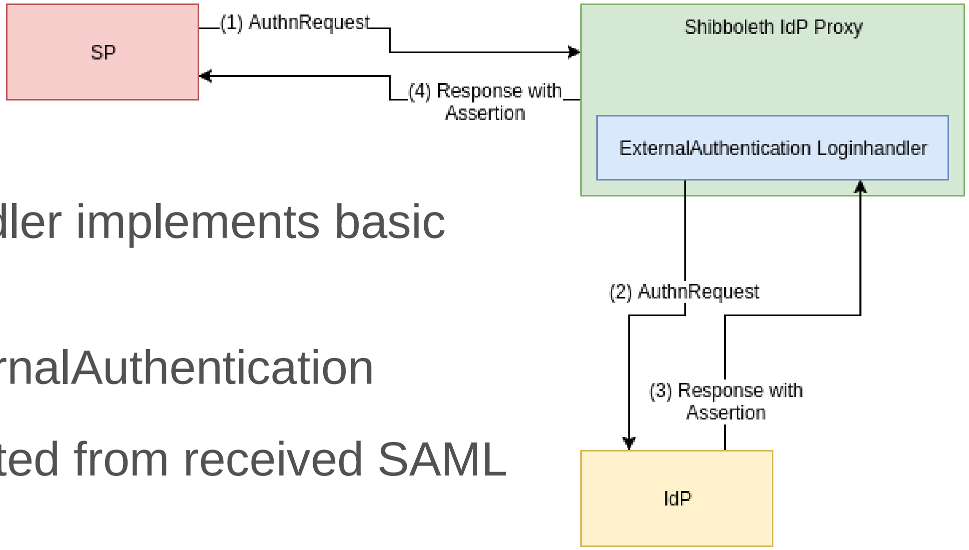
Authentication

- Common Login flows include Password, X509 and SPNEGO
- authn/Password provides a JAASAuthn module that can be used for various methods
- authn/External provides an interface to support arbitrary authentication sources
  - In general this involves writing your own code, that manages the external authentication process and connects back to the IdP
  - Extend net.shibboleth.idp.authn.ExternalAuthentication
  - You need to come up with a principal that the IdP can use to create a session, the rest is pretty much up to you
- authn/MFA or authn/Duo provides support for Multi Factor Authentication

Examples of External Authentication

- Shib-cas: https://github.com/Unicon/shib-cas-authn3
  - Uses an external CAS to obtain authentication
  - Registers its own flow (that is based on External Authentication)
  - The flow uses some properties that can be configured within the IdP to talk to the CAS server and validates the response

Examples of External Authentication


SP
(1) AuthnRequest
(4) Response with Assertion
Shibboleth IdP Proxy
ExternalAuthentication Loginhandler
(2) AuthnRequest
(3) Response with Assertion
IdP

- Proxy solution based on Shibboleth IdP
- ExternalAuthentication Loginhandler implements basic functionality of a SAML SP
- Once again based on authn/ExternalAuthentication
- After validation principal is extracted from received SAML assertion

Multi Factor Authentication

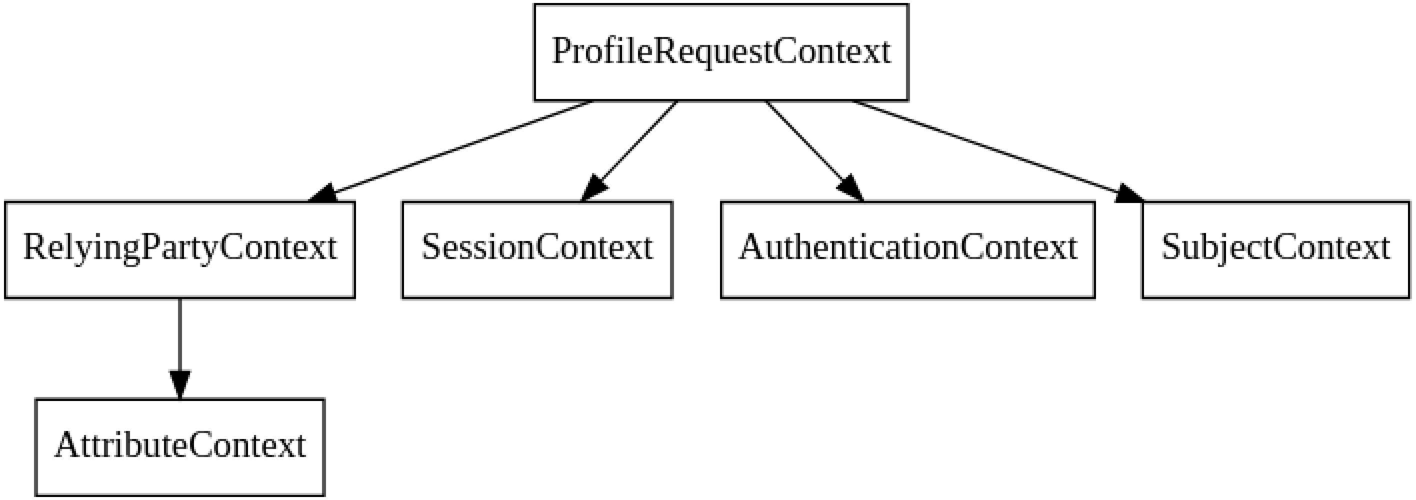
- Shibboleth IdP provides two login flows to support MFA
- authn/Duo
  - Implements the DuoWeb (3rd party) authentication interface into Shibboleth
  - Not free (>10 user)
- authn/MFA
  - Scriptable flow to multiple other flows
  - And define conditions when to use which conbination (e.g. based on AuthnContextClassRef)
  - The actual 2nd factor (i.e. TOTP) flow still needs to be implemented
  - There are extensions available (can‘t comment on them though)

Interceptors

Interceptors

- Defined as Spring Web Flows
- Can be used as „hooks“ at specific points in the profile flow
  - Inbound message
  - Post-authentication
  - Outbound message
- Often provide an easy way to modify behaviour, attributes or messages without too much implementation work
- Post-authentication interceptor has access to most variables, including authentication result and attributes
- Can be includes on a per-RP basis

Interceptors



Post-authentication interceptor

- Some flows are provided by Shibboleth IdP
  - Attribute Release Consent
  - Terms of Use Consent
  - Expiring Password
  - Impersonate
  - Context Check

Example of post-authentication intercept

- The context check interceptor can be easily extended
- Use case:
  - Terms of use are managed externally
  - IdP needs to evaluate if user can pass or should be blocked
  - This might also depend on the SP
- Most of this could also be implemented elsewhere
- Custom intercept flow provides an elegant way to do this

Example of post-authentication intercept

- Step 1: define bean for evaluation
  - Can be „scripted“ or provided by custom class
- Step 2: define flow
  - Evaluate using the custom bean
  - Positive result: proceed
  - Negative result:
    - Remove authentication result
    - Display view and inform the user
    - Redirect user to external application to accept terms of use
- Step 3: Include flow for relevant RPs

Proccessing attributes

Processing Attributes

- Resolving attributes consists of two parts
- Data Connector defines data sources to get attribute sets from (most commonly LDAP)
- Attribute Definitions extract the actual attributes from these sets
- Attribute Encoders define how to put these attributes into the assertion
- Shibboleth IdP provides commonly used types
- Data Connectors and Attribute Definitions can also be scripted
  - It‘s possible to inject external beans
  - Can get a bit confusing rather quickly ;)

Processing Attributes

- Shibboleth V3.4 will add the HTTPDataConnector that allows to query e.g. external APIs
- Currently you either need a custom Data Connector or write some pretty long scripts

Protocol Support

OIDC in Shibboleth

- shibboleth-oidc (University of Chicago)
  - https://github.com/uchicago/shibboleth-oidc
  - Based on MitreID Connect (Java)
  - Features include Authorization Code Flow, Implicit Flow, Dynamic Discovery, …
  - Configuration wired into Shibboleth
  - Not officially part of Shibboleth IdP and therefore no official support
- shibboleth-idp-oidc-extension (CSCfi)
  - https://github.com/CSCfi/shibboleth-idp-oidc-extension
  - Development as (part of) the GEANT GN4-2 project
  - Currently only Implicit Flow is implemented
  - There is some co-operation and „Code-Review“ from the Shibboleth developers
  - Goal is to eventually add this to core

Thanks!

https://www.daasi.de

DAASI International