Introducing Shibboleth

Sebastian Rieger
sebastian.rieger@gwdg.de

Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen, Germany

CLARIN AAI Hands On Workshop, 25.02.2009, Oxford eResearch Center
Agenda

- Workshop will focus on best practices and lessons learned from the MPG-AAI
- Please ask anytime – wherever possible we can supply examples from the MPG-AAI
- to provide a working example a virtual machine was prepared for this workshop:
  - https://aai.mpg.de/shib-workshop-clarin.zip
    - OpenSuSE-based virtual machine to be used with VMware Player / Virtual Box …
    - Preconfigured Shibboleth IdP (using LDAP) and SP (config in /etc/shibboleth)
    - SP directly installed (logs in /var/log/shibboleth), IdP in /srv/idp
    - Little web page providing use cases, and rough documentation to get you started
Who are you? What are you allowed to do?

- traditionally authentication (Who are you?) was done in a centralized way (e.g. gatekeeper)
- authorization (also called AuthZ) was often implied by successful authentication (AuthN)
- to minimize the effort for AuthN and subsequent AuthZ (and Accounting) on the users’ and operators’ side several solutions evolved:
  - single password / username, Single Sign-On (SSO) → Identity Management (IdM)
  - formerly focused on central authentication systems or IdM (NIS, LDAP, …)
  - limited scalability (cooperation, guests, …), limited privacy (centralized data storage)
- increasingly decentralized AuthN (Kerberos, PKI, “federated identity”, …)
  - addressing high fluctuation of external users, guests, allowing cooperation over different sites that use different IdM systems
  - especially necessary for the increasing amount of Web (2.0) applications
  - balancing security and usability (no insecure IP based access nor use of client certificates…)

Who are you? What are you allowed to do?
Identity Management evolution

- **centralized directory services**: NIS, LDAP, Passport, …
- **synchronization**: script-based, Meta-Directory; **proxy**: RADIUS, Virtual Directory, CAS…
- **federated identities**: (Kerberos trusts), SAML: Shibboleth, simpleSAMLphp, ADFS, …
- **user-centric identities**: OpenID, OAuth, CardSpace, sxip, higgins, …
What means “federated identities”?

- Centralized IdM requires synchronization of identities, for distributed user groups this task can be tough.

- To allow secure identity management for distributed users, federated IdM has two parties:
  - Identity provider (IdP), holding the credentials and doing the AuthN.
  - Service provider (SP), holding the requested resource and doing the AuthZ.

- Trust between IdPs and SPs is established using digital signatures (XMLSig, X.509). SPs check digital signature of messages from IdPs, that authenticated the user, and vice versa.

- Multiple SPs and home organizations (their IdPs) join a federation. Users use their local account (IdP) to access all services, there’s no need to hold accounts at the SPs. Identities are federated; service locations, certificates etc. form the federation’s metadata.

- To enhance interoperability of identity federations, OASIS has released the Security Assertion Markup Language (SAML), current version: 2.0.
Components of the SAML 2.0 standard

- **SAML-Assertions** – Authentication, attribute and entitlement information (Issuer, Subject, Conditions, AuthnStatement, AttributeStatement, AuthzDecisionStatement)

- **SAML-Protocol** – Request/Response pairs e.g. obtaining assertions (Assertion Query and Request, Authentication Request, Artifact Resolution, Name Identifier Management, Name Identifier Mapping, Single Logout)

- **SAML-Bindings** – Mapping SAML protocols to communication standards (SAML SOAP, Reverse SOAP, HTTP Redirect, HTTP POST, HTTP Artifact, SAML URI)

- **SAML-Profiles** – Combining bindings, protocols, assertions to specific use-cases (SSO Profiles [IdP Discovery, Web Browser SSO, SLO ...], Attribute Profiles [XACML, LDAP, Basic, UUID, ...], Artifact Resolution, Name Identifier Mapping, Assertion Query / Request)
SAML Request Example

SAML AuthnRequest in the body of a SOAP message
Profile: Assertion Query/Request, Binding: SOAP, Protocol: Assertion Query/Request

```xml
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap/envelope/">
  <env:Body>
    <samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" ForceAuthn="true"
      AssertionConsumerServiceURL="http://www.example.com/
      AttributeConsumingServiceIndex="0"
      ProviderName="string" ID="sd23az" Version="2.0" IssueInstant="2009-01-31T12:00:00Z"
      Destination="http://www.example.com/
      Consent="http://www.example.com/">
      <saml:Subject xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
        <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
          mmuster@gwdg.de
        </saml:NameID>
      </saml:Subject>
    </samlp:AuthnRequest>
  </env:Body>
</env:Envelope>
```

several other bindings can be used instead of SOAP
(typically HTTP redirect, HTTP POST, HTTP Artifact)
SAML Response Example

SAML assertion, placed inside a <samlp:Response>

<saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" Version="2.0"
    IssueInstant="2009-01-31T12:00:00Z">
    <saml:Issuer>idp.gwdg.de</saml:Issuer>
    <saml:Subject>
        <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">mmuster@gwdg.de</saml:NameID>
    </saml:Subject>
    <saml:Conditions NotBefore="2009-01-31T12:00:00Z", NotOnOrAfter="2009-01-31T12:00:00Z"/>
    <saml:AuthnStatement AuthnInstant="2009-01-31T12:00:00Z" SessionIndex="2378942">
        <saml:AuthnContext>
        </saml:AuthnContext>
    </saml:AuthnStatement>
    <saml:AttributeStatement>
        <saml:Attribute Name="eduPersonAffiliation">
            <saml:AttributeValue>staff</saml:AttributeValue>
        </saml:Attribute>
    </saml:AttributeStatement>
</saml:Assertion>

other extensions like <Signature>, <AuthzDecisionStatement> are possible…
Building up federations...

- federation enables Single Sign-On over contained services (SPs) for all IdPs – it is a secure unified Authentication and Authorization Infrastructure (AAI) for all relying parties.
- multiple federations can be joined (e.g. to allow roaming)
  - IdPs or SPs can be placed in several different federations
  - Bridging elements (federation gateways) can be used (e.g. eduGAIN) allowing different protocols
- besides technical implementation trust in the federation is effectively controlled by contracts or the policy (e.g. ensuring account expiration, secure handling of certificates …)
User-centric identity management

- latest development in decentralized IdM: “user-centric”, “Identity 2.0”, …
  - enhanced usability and privacy, ubiquitous access, simplification
- self-determination and -service, the user manages his identities
  - Identity Selector: user selects his identity while accessing a resource – allows easy discovery of home organization
- Information Cards (I-Cards) can be used to present personal data
  - Managed (IdP), Personal (Self-Signed), Relationship (Social Networking)
- Implementations: OpenID, CardSpace, sxip, OAuth, higgins, Bandit, Ping…
- big players joined in last year: as Providers (≈ IdP), less Consumers (≈ SP)
  - OpenID (Provider): Google, IBM, Microsoft, VeriSign, Yahoo!
  - CardSpace (Provider): Microsoft Windows Vista, Higgins, …
OpenID

- Login uses URL (e.g. of a Provider, Blog, …), some use the mail address (Google)
- Discovery of the Provider can easily be done using the domain used as an OpenID
- Several extensions are available for OpenID 2.0 or sxip, e.g. multiple identities per user (personas) or trusted attribute authorities (digitally signed e.g. Card Space)
all solutions (federated and user-centric) are focussing web applications (with some exceptions like GridShib)

while interoperability has increased with SAML 2.0, some issues are still not addressed in the standard: privacy, accounting, integration of multiple federations → they are addressed by the specific implementations

user-centric IdM is simple to set up and use, enhances privacy, offers an intuitive discovery and globally unique ids

some of the user-centric solutions are platform or browser dependent, some still hold security flaws (Replay, MITM, XSS, CSRF, …)

users are bound to their Provider in user-centric IdM, if the Provider changes terms of use etc. identities can be lost, if users set up their own providers instead, who will trust them?

currently federated identity management solutions are state-of-the-art in scientific communities… integration of e.g. OpenID with federated IdM solutions is likely to happen!