UbuntuNet Alliance- Identity Federation Training

Identity Federation key components

www.geant.org

Content

- What is **Security Assertion Markup Language (SAML)** ?
  - SAML V2.0 Deployment Profile for Federation Interoperability
  - SAML Components
- Example of **Authentication flow**
- SAML Metadata
  - Federation Metadata
  - Federation Metadata Structure
- Add Trust to Metadata
  - Certificates usage between IdPs and SPs
  - Certificates usage at the Federation level
- SAML2 IdP and SP Implementations
  - SAML2 IdP and SP Selection criteria
- Introduction to Shibboleth
  - Shibboleth Service Provider
  - Shibboleth Identity Provider
- Attributes exchange
  - Attributes Specification
  - Attribute Schemas
- Recommended Attributes in eduGAIN
- Identifiers Attributes

What is Security Assertion Markup Language (SAML) ?

The Security Assertion Markup Language is an **XML-based, open- standard data format for exchanging authentication and authorization data between parties**, in particular, between an Identity Provider and a Service Provider.

SAML - Security Assertion Markup Language

- OASIS standard describing the XML messages exchanged
- between Identity Provider (IdP) and Service Provider (SP)
- Its purpose is to enable the **authentication and secure exchange trusted identity information (attributes)** between IdP and SP

SAML 2.0 is the de facto standard for academic identity federations

SAML 2.0 Web Browser Single Sign On Profile

- Profile that describes how to use SAML in order to achieve SSO
- Main use of SAML within Identity Federations

SAML V2.0 Deployment Profile for Federation Interoperability

This profile specifies behavior and options that deployments of the **SAML V2.0 Web Browser SSO profile**, and related profiles, are required or permitted to rely on.

SAML2int Interoperable SAML 2.0 Profile

- A deployment profile for SAML2.0 Web Browser Single Sign On profile
- Aims to influence how a SAML entity should be implemented
- Aims to influence how a SAML entity implementation should be configured

SAML Components

SAML Metadata

An XML document describing SAML Entities, both in technical as well as non-technical terms.

Valid SAML Metadata MUST meet the requirements defined in the SAML Metadata Specification

SAML Metadata Consumer

An entity or organization that downloads, processes and uses SAML V2.0 Metadata.

SAML Metadata Producer

An organization that produces and publishes SAML V2.0 Metadata.
An XML document describing SAML Entities

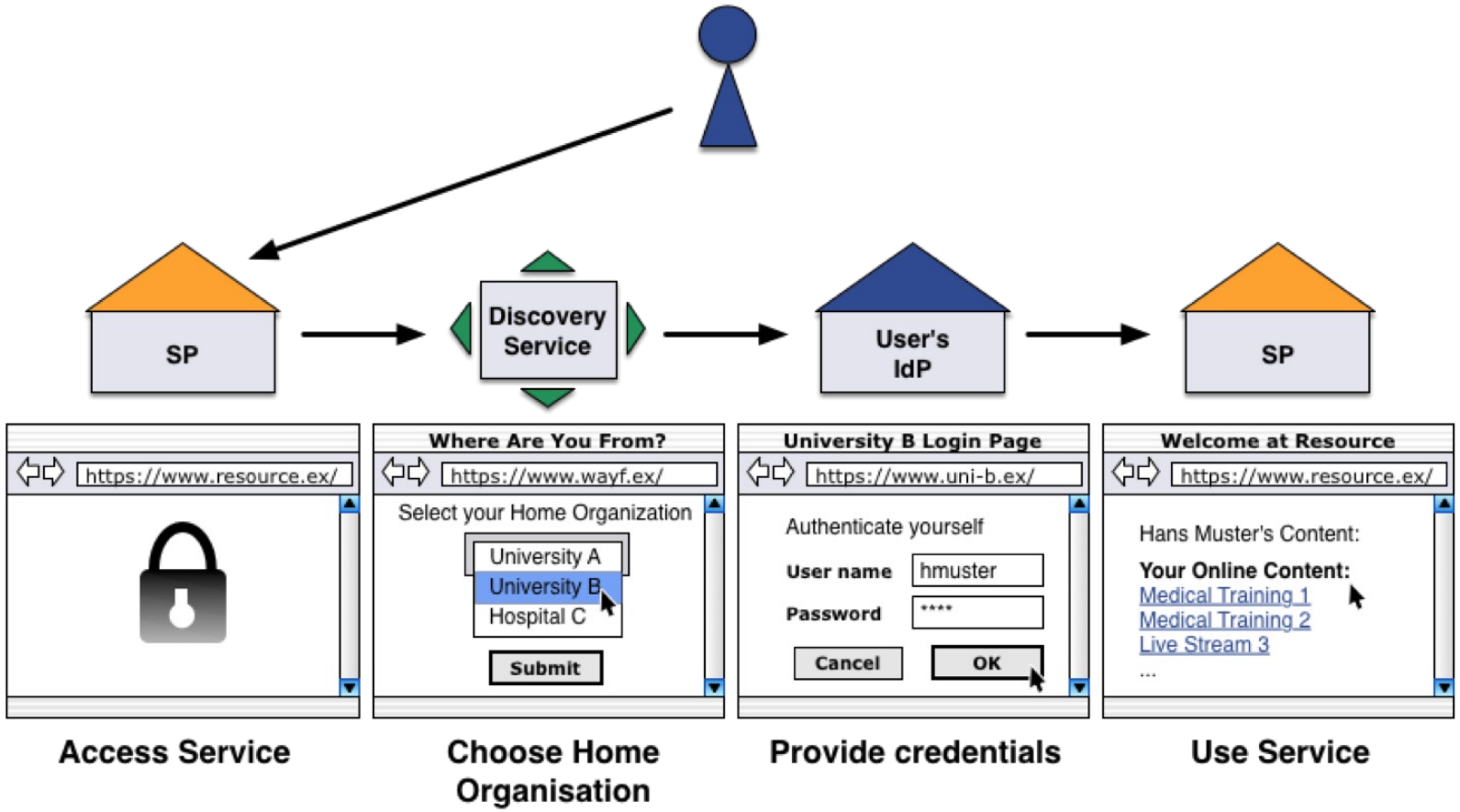
SAML Metadata Consumers/Producers

At the institution level	
	• A **SAML authority that authenticates users** against a user repository • Retrieves information for the users in the form of attributes • Transfers the authentication event along with the attributes to a SAML Service Provider
	• **A SAML consumer** that acts as a middleware in order to protect Web Applications • Consumes SAML messages from the Identity Provider and deduces authentication events and attributes
Discovery Service	• The Discovery Service service, also known as "Where Are You From (WAYF)" service, **lets the user choose his home institution from a list and then redirects the user to the login page** of the selected institution for authentication.

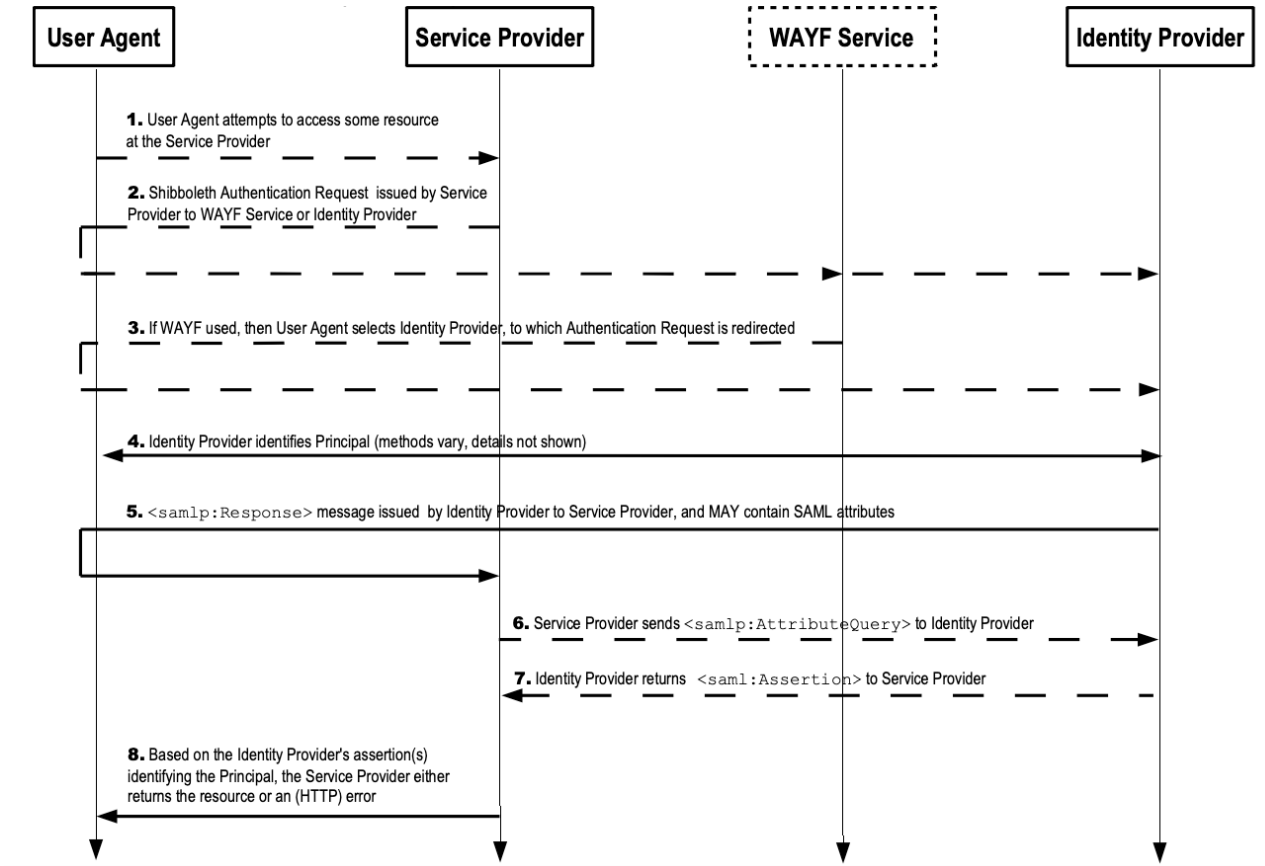
SAML Metadata Consumers/Producers

At the NREN level	
Federation Registry	The system component that helps Federation Operators to register and manage entities. It could be uses for collecting, processing and republishing federation metadata.
Central Discovery Service (optional)	A central Discovery Service, operated by Federation Operators to support and help their institutions to deploy services without having to deploy a discovery service at their level.

Example of Authentication flow



SAML Authentication Flow



SAML Metadata

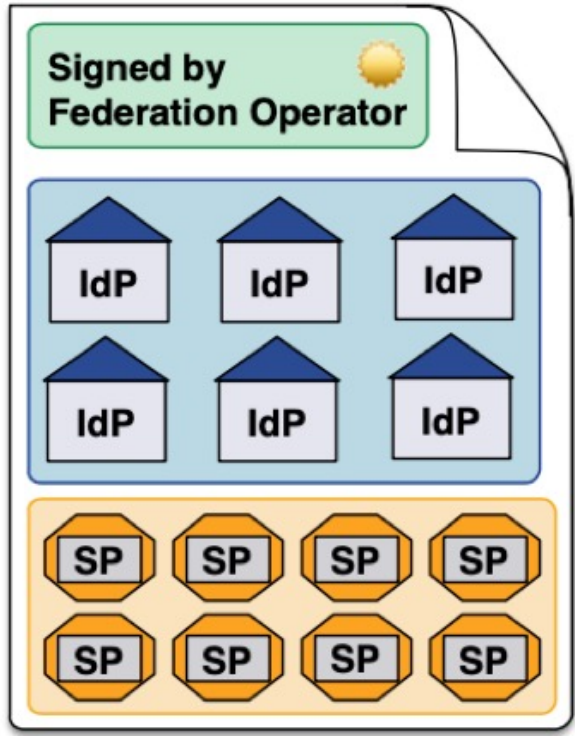
- Entities in a federation trust each other, but which entities are in federation?
- Entities in a federation must be known to trust them. Therefore, a standard way to list and describe entities is needed.
- (SAML2) Metadata provides such a standard way. Standardized format to describe entities
- Metadata typically is signed by a trusted third party, in our case the federation operator which should be trusted by all participants in the federation.

Federation Metadata

- **The federation Metadata provides the technical trust in the federation**
  - XML Documents defined by the SAML 2.0 standards
  - Generated by the Federation operators
  - Cryptographically signed by the Federations operators
  - Optionally transported over the internet using SSL
  - Contains technical information on all participating entities

Federation Metadata Structure

- To trust the metadata,
  it should be **protected** properly.
- No defined order of IdPs and SPs.
- Other entities could be described too.
  But mostly IdPs and SPs.



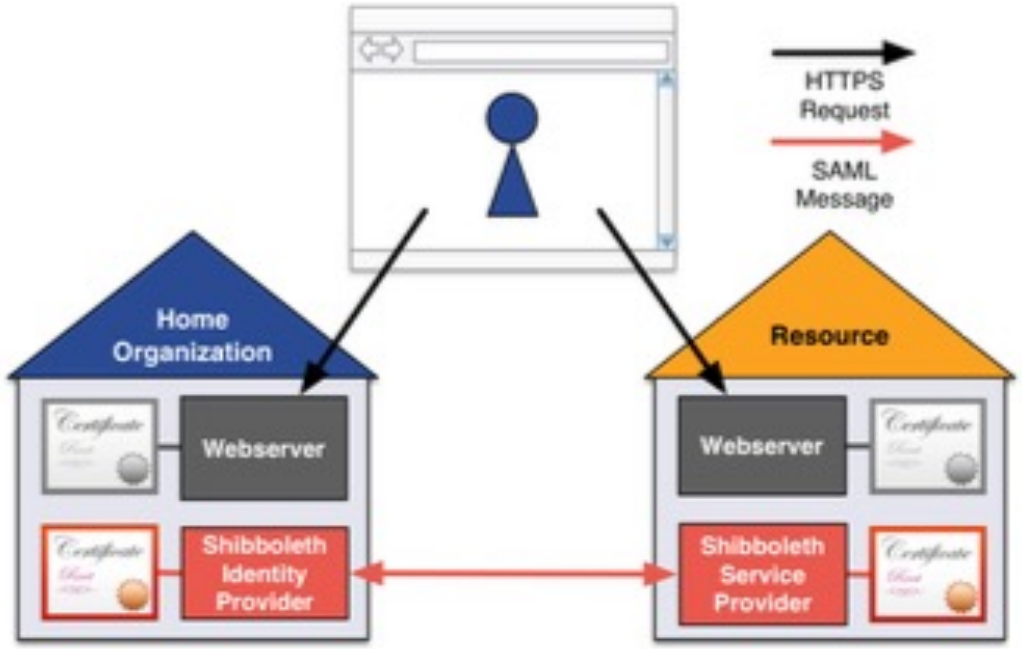
Add Trust to Metadata

- **Consumers** of metadata must be sure that the metadata was really created by Federation Operator
- Therefore, **metadata must be secured**
- Two methods to secure metadata:
  A. **Recommended:** Add an **XML signature** on metadata and publish public signing key Metadata can be served via http in this case.
  B. Serve plain metadata via a **secure HTTPS URL**.
  Make web server use a certificate issued by a well-known CA

Certificates usage between IdPs and SPs

- **SSL/TLS between the user's browser and the Web server:**
  - indicates that the Web Service protects user data and ensures that the user is connected to an authentic site.
- **Self-signed certificate (and private key) for signing/encrypting, with long lifetime (> 10y)**
  - **The signing of SAML assertions** allows the recipient of a SAML assertion to verify the identity of the issuer and to verify its integrity. This operation is carried out by the two parties involved in the exchange (IDP and SP).
  - **Encryption of SAML assertions** by using the recipient's certificate to encrypt the assertion, the recipient will be able to access the content of the assertion.

Certificates usage between IdPs and SPs



Certificates usage at the Federation level

- **Federation operator signing certificate (the metadata file published by the federation operator is signed):**
  - eduGAIN **collects the metadata** of all the participating federations, **re-signs** and **publishes the aggregated metadata** for the interfederation so that it can be consumed by all the participating Federations.
  - In order to be able to validate the integrity and authenticity of the Federation's metadata, **the eduGAIN Operations Team needs to receive the certificate with which the Federation signs it's locally aggregated metadata.**
  - This certificate is also used by IdPs and SPs to ensure that the integrity and authenticity of the Federation's metadata.

SAML2 IdP and SP Implementations

- **Shibboleth** (most common in academic environment, most comprehensive set of features)
- **SimpleSAMLPHP** (second most common, mostly suitable for PHP applications)
- **Microsoft ADFS** (limited SAML2 support, much handwork to get it running in a federation)
- **pySAML2** (python implementation)
- **mod_mellon** (Apache module, small user base)
- **……**

SAML2 IdP and SP Selection criteria

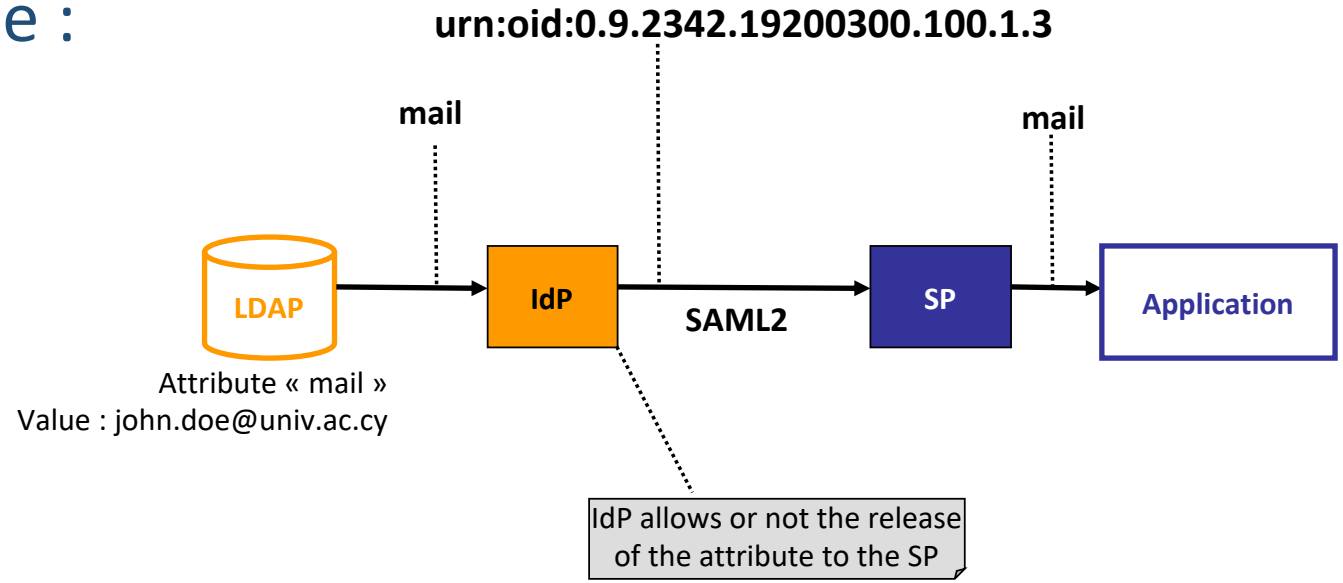
- Can it consume SAML2 metadata?
  - Containing > 9k entities
  - More than 80MB in size
  - Can metadata be refreshed automatically

- Does it support the Web SSO profile (saml2int.org)?
  - E.g. can it process signed and encrypted SAML assertions from IdPs with self-signed X.509 certificates?

- Is it a **secure, well adopted implementation backed by a strong community or vendor** ?

Attributes exchange

- When a user attempts to access a Service Provider-protected site, it usually asks the user's Identity Provider to provide one or more specific ***identity attributes*.**
- Example :



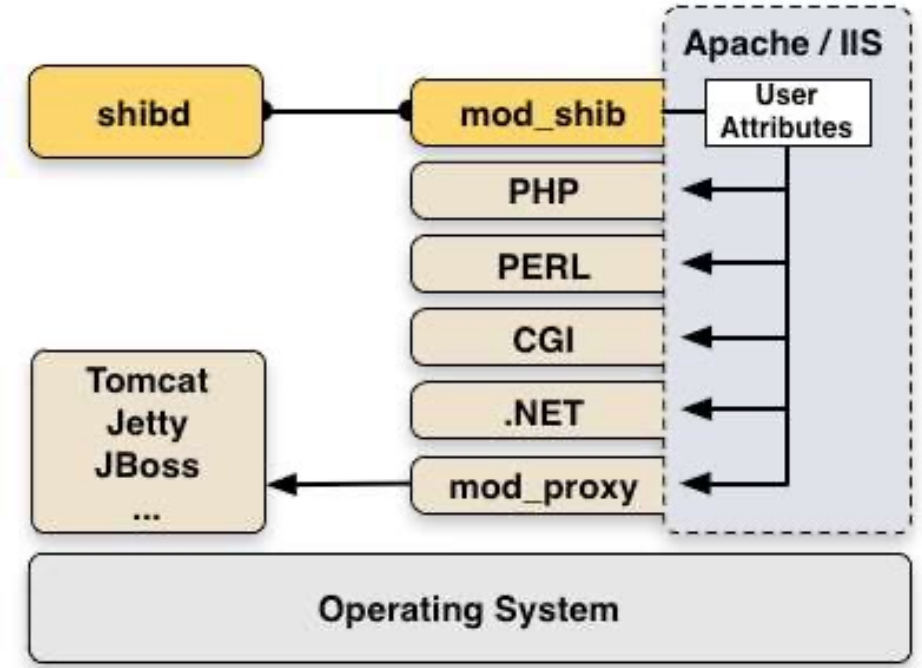
Introduction to Shibboleth

- **The Origin**
  - Internet2 in the US launched the open source project in 2000
- **The name**
  - Word Shibboleth was used to identify members of a group
- **The standard**
  - Based on Security Assertion Markup Language (SAML)
- **The Consortium**
  - The new home for Shibboleth development
  - Collect financial contributions from deployers worldwide

- The Shibboleth software is the most widely used in the research and education environment
- Website: https://shibboleth.net/

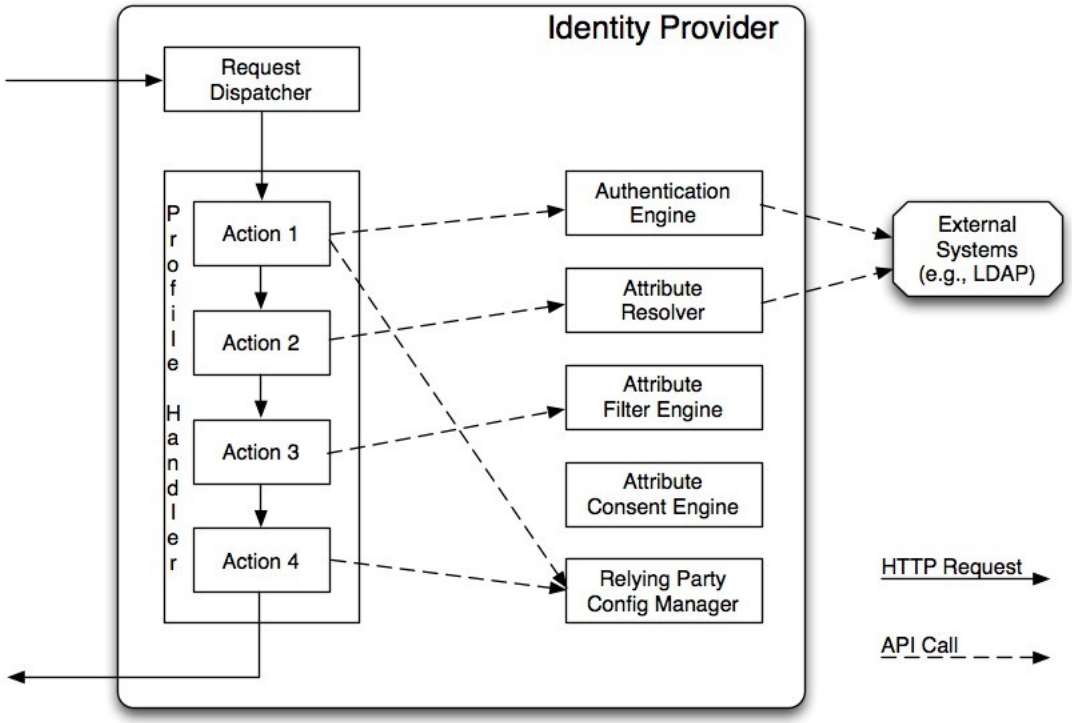
Shibboleth Service Provider

- Runs on: Linux, Solaris, Windows, Mac OS X, FreeBSD, ...
- **Protects web applications**
- shibd processes attributes
- **Can authorize users** with
  - Apache directives
  - Shibboleth XML
  - Access rule
- **Provides attributes to applications** via web server environment variables or headers



Shibboleth Identity Provider

- Runs on: Linux, Solaris, Windows, Mac OS X, FreeBSD, ...
- **Authentication**
- **Attribute Resolution**
  - **pulling in data** from external systems (e.g., LDAP directories and relational databases)
  - **Creating attributes** from the pulled in data
  - **Associating protocol-specific encoders with the created attributes.**
- **Attribute Filter**
  - Determines what information are allowed to be sent to a requester.
- Relying party
  - Define SAML profiles for SPs



Authentication

- Based on Spring Web Flow

- Login flows
  - **Password**
  - RemoteUser
  - RemoteUserInternal
  - X509
  - X509Internal
  - SPNEGO / Kerberos
  - IPAddress
  - External
  - Multi-Factor
  - Function
  - SAML

Most popular

Where to specify which authentication flow to use ?
/opt/shibboleth-idp/conf/idp.properties :
« idp.authn.flows = Password »

Where can I find available flows ?
/opt/shibboleth-idp/auth/….

Attribute Resolver

- Sample files provided by default
  - **/opt/shibboleth-idp/conf/attribute-resolver.xml**
  - **/opt/shibboleth-idp/conf/attribute-resolver-ldap.xml**
- Where to choose the right file to take into account ?
  - ➔ /opt/shibboleth-idp/conf/services.xml
- It contains :
  - **DataConnectors**
  - **Attribute definition**

DataConnectors

Defines connections to sources of data which provide input to attribute definitions.

DataConnector Plugin Types	
Static	+++
ScriptedDataConnector	+++
ComputedId	++
StoredId	+++
PairwiseId	
RelationalDatabase	++
LDAPDirectory	+++
HTTP	
Subject	
StorageService	++
EntityAttributes	

Data Connector (example: LDAPDirectory)

```
<!-- ========================================== -->
<!--      Data Connectors                       -->
<!-- ========================================== -->

<!--
Example LDAP Connector

The connectivity details can be specified in ldap.properties to
share them with your authentication settings if desired.
-->
<DataConnector id="myLDAP" xsi:type="LDAPDirectory"
    ldapURL="%{idp.attribute.resolver.LDAP.ldapURL}"
    baseDN="%{idp.attribute.resolver.LDAP.baseDN}"
    principal="%{idp.attribute.resolver.LDAP.bindDN}"
    principalCredential="%{idp.attribute.resolver.LDAP.bindDNCredential}"
    useStartTLS="%{idp.attribute.resolver.LDAP.useStartTLS:true}"
    connectTimeout="%{idp.attribute.resolver.LDAP.connectTimeout}"
    trustFile="%{idp.attribute.resolver.LDAP.trustCertificates}"
    responseTimeout="%{idp.attribute.resolver.LDAP.responseTimeout}">
    <FilterTemplate>
        <![CDATA[
            %{idp.attribute.resolver.LDAP.searchFilter}
        ]]>
    </FilterTemplate>
        <ConnectionPool
        minPoolSize="%{idp.pool.LDAP.minSize:3}"
        maxPoolSize="%{idp.pool.LDAP.maxSize:10}"
        blockWaitTime="%{idp.pool.LDAP.blockWaitTime:PT3S}"
        validatePeriodically="%{idp.pool.LDAP.validatePeriodically:true}"
        validateTimerPeriod="%{idp.pool.LDAP.validatePeriod:PT5M}"
        expirationTime="%{idp.pool.LDAP.idleTime:PT10M}"
        failFastInitialize="%{idp.pool.LDAP.failFastInitialize:false}" />
</DataConnector>
```

Data Connector (example: Static)

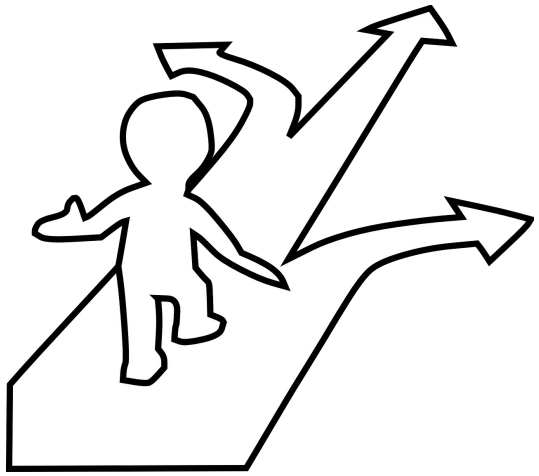
```
<!-- ========================================== -->
<!--      Data Connectors                       -->
<!-- ========================================== -->

<DataConnector id="staticAttributes" xsi:type="Static">
    <Attribute id="affiliation">
        <Value>member</Value>
    </Attribute>
</DataConnector>
```

Attribute Resolver

Attribute Definition	
Simple	+++
PrincipalName	
Scoped	+++
Prescoped	+++
RegexSplit	
ScriptedAttribute	+++
Mapped	+++
Template	
SubjectDerived	
ContextDerived	
Decrypted	

Which Attribute Definition to chooses ?

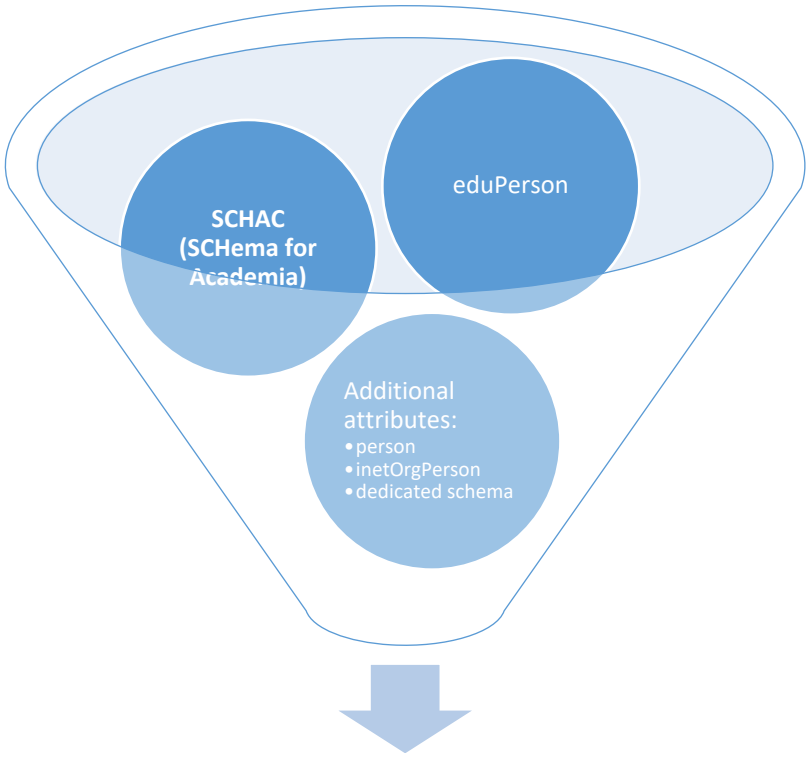


Attributes Specification

- **Attribute Specification** is crucial for the **data exchange within a federation** or eduGAIN, it provides the common basis on which two communicating entities are able to share information they know to interpret identically.

- **Federation participants must support a defined set of user attributes** and requires or recommends the use or release of user attributes to enable **interoperability**, good user experience, and to help protect personal privacy.
  A supported attribute might or might not be made available for use by an Identity Provider.

Attribute Schemas



MyEduPerson

https://wiki.refeds.org/display/STAN/SCHAC+Releases
https://wiki.refeds.org/display/STAN/eduPerson+2020-01

Recommended Attributes in eduGAIN: 1/2

Attribute	Description
commonName *OID:* 2.5.4.3	Name and Surname of the user. *Example: John Doe*
displayName *OID:* 2.16.840.1.113730.3.1.241	Name and Surname of the user. *Example: John Doe*
mail *OID:* 0.9.2342.19200300.100.1.3	Holds Internet mail address *Example: john.doe@example.org*
schacHomeOrganization *OID:* 1.3.6.1.4.1.25178.1.2.9	Specifies a person´s home organization using the domain name of the organization. *Example: example.org*
schacHomeOrganizationType *OID:* 1.3.6.1.4.1.25178.1.2.10	Type of a Home Organization *Example: urn:schac:homeOrganizationType:int:university*

Recommended Attributes in eduGAIN: 2/2

Attribute	Description
eduPersonAffiliation *OID:* 1.3.6.1.4.1.5923.1.1.1.1	Specifies the person's relationship(s) to the institution in broad categories. Example: such as student, faculty, staff, alum, etc.
eduPersonPrincipalName (ePPN) *OID:* 1.3.6.1.4.1.5923.1.1.1.6	Unique, persistent identifier of the user. Example: *jdoe@example.org*
eduPersonScopedAffiliation *OID:* 1.3.6.1.4.1.5923.1.1.1.9	Specifies the person's affiliation within a particular security domain. Example: *staff@example.org;member@example.org*
eduPersonTargetedID/persistentID (ePTID) *OID:* 1.3.6.1.4.1.5923.1.1.1.10	Unique, persistent, opaque and targeted identifier of the user. (Serialized) Example: https://idp.example1.org/idp/shibboleth!https://SP.example.org!yrdfefohZY+cdGvqu/Dubc=

Identifiers Attributes

- Web services often need to **re-identify users** in order to present them their profile, and should receive only the absolutely needed information about the users (data minimization principle).

- Properties of identifiers:
  - **Uniqueness:** as the name implies an identifier should identify a user without doubt, no two users should have the same identifier
  - **Reassignability**: user identifiers can be reassigned to another user after the first user with this identifier has left the organization. Reassigned identifiers can cause access control and traceability problems.
  - **Opacity**: an identifier that gives no clue (i.e. it only contains random data and no names) about the identity of the user is called opaque.
  - **Persistency:** an identifier that stays identical over time is called persistent.
  - **Targetedness**: an identifier that is intended to be used for a single service only and is specific to that service is called targeted.
  - **Transientness:** a transient identifier stays the same for a login session, but changes when the user logs in again.

Overview of identifier attributes and their characteristics

Property/Identifier	ePPN	ePTID	ePUID	mail
Unique	X	X	X	X
Non-reassignable		X	X	
Opaque		X	X	
Persistent	X	X	X	X
Targeted		X		
Transient				
Availability in eduGAIN	Very good	Good	Poor	Very good

Attribute Filter

- The Shibboleth IdP allows filtering outgoing user attributes
  - To limit the distribution of nominative attributes to services
- /opt/shibboleth-idp/conf/attribute-filter.xml
  - Defines rules for each SP
  - Defines rules for each attribute
- By default, this file is poor in filtering rules
  - no attribute sent

Side effect of attribute filtering at IdP level ➔ If the IdP is too restrictive, users will not have access to services (SP)

Question ?

But how IdPs knows what attributes are requested by each service ?

From the attribute filter, IdP administrators are defining which attributes to send to each service

eduGAIN contains more than 3000 services, could be time consuming to define attribute release for each one.

- **Use of Dynamic Filter configuration:**
  - **Requested attributes could be shared in the metadata (SP or federation metadata)**
  - **IdPs can be configured to release automatically these attributes if available in metadata**

GÉANT
Networks • Services • People

Thank you

Any questions?

www.geant.org

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 856726 (GN4-3).