Federated Grid Access Using EMI STS

Henri Mikkonen <henri.mikkonen@hip.fi>
Helsinki Institute of Physics (UH.HIP)

ISGC 2013, March 17-22, 2013
Academia Sinica, Taipei, Taiwan
Contents

• Problem definition
• Related work
• Security Token Service (STS) overview
• Use cases for federated identity
• (Short) Demonstration
Federated Grid Access?

• Identity federations
  – National and international (eduGAIN)
  – De-facto standard is Security Assertion Markup Language (SAML)
  – Mostly Web-based
    • *Non-browser is coming up (e.g. Project Moonshot)*

• Grid middlewares
  – EMI: ARC, dCache, gLite and UNICORE
  – User authentication is based on X.509 certificates
  – Command-line and other non-browser clients
Related work

• Multiple solutions to X.509 issuance based on federated identity exist
  —Compatible IGTF profiles: Member Integrated Credential Services (MICS) and Short-Lived (SLCS)
• Most of them are at least partly Web-based
  —Either directly, or command-line client might be emulating a Web browser behind the scenes
• Proprietary protocols often used between the client and the service
• Grid identity is not just the X.509 certificate
  – gLite, ARC and dCache use *Grid proxies*
  – Grid proxy = End-Entity Certificate (EEC) + (short-lived) proxy certificate + private key
    • *Proxy certificate contains VO attributes*

• Grid proxy initialization (*voms-proxy-init*)
  – Request VO attributes from VOMS
  – Attach them into the proxy certificate which is signed by the private key corresponding to EEC
  – Default lifetime is 12 hours
What is STS?

• WS-Trust: “STS is a Web service used to issue, renew, validate and cancel security tokens”
  – Security Token: “A collection of statements (claims) about a user or resource” (WS-Security)
    • Any digital identity that can be attached into a SOAP message
  – STS establishes a trust relationship between different application / security domains

• EMI STS implements the ISSUE operation for the supported token formats
STS Architecture

SECURITY TOKEN SERVICE

- SOAP Client
  - Request Dispatcher
  - WS-Trust Handler
    - AuthN Engine
    - Attribute Decoder
    - Attribute Resolver
    - Token Generator
      - Plugin
      - Plugin
      - Plugin
      - Plugin

- External Source
- External Source
- External Source

19/03/2013
Henri Mikkonen @ ISGC 2013
SAML to Proxy - Components

- User knows username/password at his home institute & wants to access a Grid resource

<table>
<thead>
<tr>
<th>Component</th>
<th>Credential input</th>
<th>Credential output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid resource</td>
<td>Grid Proxy</td>
<td></td>
</tr>
</tbody>
</table>
SAML to Proxy - Components

• User knows username/password at his home institute & wants to access a Grid resource

<table>
<thead>
<tr>
<th>Component</th>
<th>Credential input</th>
<th>Credential output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid resource</td>
<td>Grid Proxy</td>
<td></td>
</tr>
<tr>
<td>VOMS</td>
<td>X.509 certificate</td>
<td>VOMS attributes</td>
</tr>
</tbody>
</table>
User knows username/password at his home institute & wants to access a Grid resource

<table>
<thead>
<tr>
<th>Component</th>
<th>Credential input</th>
<th>Credential output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid resource</td>
<td>Grid Proxy</td>
<td></td>
</tr>
<tr>
<td>VOMS</td>
<td>X.509 certificate</td>
<td>VOMS attributes</td>
</tr>
<tr>
<td>Online CA</td>
<td>Certificate Signing Request (CSR)</td>
<td>X.509 certificate</td>
</tr>
</tbody>
</table>
SAML to Proxy - Components

• User knows username/password at his home institute & wants to access a Grid resource

<table>
<thead>
<tr>
<th>Component</th>
<th>Credential input</th>
<th>Credential output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid resource</td>
<td>Grid Proxy</td>
<td></td>
</tr>
<tr>
<td>VOMS</td>
<td>X.509 certificate</td>
<td>VOMS attributes</td>
</tr>
<tr>
<td>Online CA</td>
<td>Certificate Signing Request (CSR)</td>
<td>X.509 certificate</td>
</tr>
<tr>
<td>STS</td>
<td>SAML assertion</td>
<td>Grid Proxy</td>
</tr>
</tbody>
</table>
**SAML to Proxy - Components**

- User knows username/password at his home institute & wants to access a Grid resource

<table>
<thead>
<tr>
<th>Component</th>
<th>Credential input</th>
<th>Credential output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid resource</td>
<td>Grid Proxy</td>
<td></td>
</tr>
<tr>
<td>VOMS</td>
<td>X.509 certificate</td>
<td>VOMS attributes</td>
</tr>
<tr>
<td>Online CA</td>
<td>Certificate Signing Request (CSR)</td>
<td>X.509 certificate</td>
</tr>
<tr>
<td>STS</td>
<td>SAML assertion</td>
<td>Grid Proxy</td>
</tr>
<tr>
<td>Home Institute</td>
<td>Username/password</td>
<td>SAML assertion</td>
</tr>
</tbody>
</table>
SAML to Proxy - Components

Username/password

SAML assertion

STS SOAP Client

Home Institute

SAML TRUST DOMAIN

Proxy

RST(SAML):Proxy

STS

X.509 TRUST DOMAIN

CSR

X.509 certificate

Request for attributes

VOMS attributes

Online CA

VOMS

Username/password

SAML assertion

RST(SAML):Proxy

Proxy

STS SOAP Client

Home Institute

SAML TRUST DOMAIN

Online CA

VOMS

X.509 TRUST DOMAIN
Use case 1: Web portal

Browser access

Web portal

STS SOAP Client

ST S SOAP Client

SAML AuthnRequest

Username/password

SAML Assertion

SAML TRUST DOMAIN

Home Institute

STS

RST(SAML):Proxy

Proxy

X.509 TRUST DOMAIN

Online CA

CSR

X.509 certificate

Request for attributes

V O M S attributes

V O M S
Use case 1: Web portal

- Relatively simple
- Uses SAML 2.0 Web SSO Profile
- Certificates nor private keys are never stored locally by the user
- Requires browser (to some users)
- If STS has different SAML entity than portal (as it should), SAML delegation is required
Use case 2: CLI with ECP profile

Client tool

STS SOAP Client

Local access

3.

5.

STS

SAML TRUST DOMAIN

Home Institute

1.

2.

4.

6.

7.

X.509 TRUST DOMAIN

Online CA

CSR

X.509 certificate

Request for attributes

VOMS attributes

RST(SAML):Proxy

Proxy

Client tool tool

1.

2.

3.

4.

5.

6.

7.

19/03/2013

Henri Mikkonen @ ISGC 2013

16
Use case 2: CLI with ECP profile

1. Client tool initiates the ECP profile
2. STS issues a SAML AuthnRequest
3. User chooses home institute from the list
4. Client tool sends the AuthnRequest to the chosen home institute
5. User writes his username/password
6. Client authenticates the user to his home institute
7. Home Institute Issues SAML assertion
Use case 2: CLI with ECP profile

1. Client tool initiates the ECP profile

- No Web-browser required (to some users)
- Supported by Shibboleth Identity Provider (as of version 2.3)

- Not widely supported worldwide (a small fraction of the users need the ECP profile)

19/03/2013
Henri Mikkonen @ ISGC 2013
Use case 3: CLI with another STS

Client tool

STS SOAP Client

Home Institute

SAML assertion

RST(Username&Passwd):SAML

variant

STS

Proxy

RST(SAML):Proxy

X.509 TRUST DOMAIN

Online CA

CSR

X.509 certificate

Request for attributes

VOMS attributes

VOMS

SAML TRUST DOMAIN

Home Institute

RST(Username&Passwd):SAML

19/03/2013

Henri Mikkonen @ ISGC 2013
Use case 3: CLI with another STS

- Simpler than the ECP profile to be implemented

- Not widely supported by the open source SAML Identity Provider softwares
Summary

• STS is a general purpose service used for transforming security tokens
  – It can be used in both Web browser and non-Web use cases for enabling Grid access using federated identity
    – SAML to X.509 & Grid proxy conversion
      • Proxy initialization can be outsourced to STS
  
• STS is relatively simple to be integrated
  – One SOAP message exchange between the client and the STS
Thank you!

STS is a part of the EMI-3 release:
http://www.eu-emi.eu/emi-3-montebianco