Federated Identity Management for the EUDAT Data e-Infrastructure

Principled promoting of persistent personal principals: particular practical perspectives

Jens Jensen, STFC
EUDAT AAI TF

DPConline workshop July 2013
Background – EUDAT *in nuce*

- EUDAT is building a *data e-infrastructure*
  - Support user communities (ESFRI)
    - CLARIN (linguistics, heterogeneous + long tail)
    - ENES (climate)
    - EPOS (Earth obs)
    - VPH (human physiology)
    - LifeWatch (biodiversity)
  - Move data in and out of EUDAT: PRACE, EGI (“data staging”)
  - Move data between sites (“safe replication”)
  - Data storage for individual users (“simplestore”)
Principles: AAI

• Authentication
  – Make use of *existing infrastructures*
  – SSO whenever possible
  – Make use of existing code - pragmatic

• Authorisation
  – Link to community rôles (users can be in more than one community)

• Delegation…
  – Even if it’s identity delegation

• Infrastructure
  – Like the grids, secure with IGTF+commercial
(Other) High Level Goals

• Usable… users are often non-technical
  – Can’t manage X.509 certificates
• Promote collaborations – interdisciplinary
• Work with what communities already have
  – Unless it’s rubbish (maybe)
  – So need multi-LoA support 😊
  – “The Facebook generation”
• Modular – SOA (use of standards, web services)
• Practical rather than perfect
“Federated” Identity

User ➔ Web server
Shibboleth

User

WAYF

Web server
Shibboleth

IdP

AA

WAYF

User

Web server
Federated Identity

IdP
AA

User

WAYF

Web server
Federated Identity

IdP
AA

WAYF

User

Web server
Proposal – the 36,000 feet view

• Use external identity providers
  – Used by communities: OpenID, Shib

• Internal SLCS: X.509
  – Credential managed by portal, not user
  – Support command line access
  – Support delegation

• Central federation database
  – Can be distributed, but is one DB
  – Handles attributes, too

• Infrastructure – accept IGTF (like EGI, PRACE)
Proposal – the 36,000 km view

contrail-project.eu

www.igtf.net

www.ogf.org
Different types of Identity Providers

IdP A

IdP B

IdP C

IdP D

AuthN

AtP 1

AtP 2

AtP 3

Identity credential conversion

EUDAT

consolidated credentials

Johannes Reetz, RZG

attribute based access control information

Attribute Provider AuthZ

either community-managed or (*) attributes provided by user’s home IdP are reused
Requirements

• Scalable (10**7 users)
• Easy enough to use for “non-technical” users
• Support long tail researchers (aka homeless)
• Portal and command line login
• Mature, robust, performant
• Standards-based
• Work with existing community practices (if pos.)
• Communities manage authorisation policies
Premise

- **Support existing user communities**
  - CLARIN already using Shib (note the ePTID problem)
  - ENES already use OpenID (in ESGF)
  - Provide “authentication services”

- **Federated identity management**
  - Must work with iRODS for data storage
  - Must work with GridFTP (and GlobusOnline) for data movement
  - Must work with Invenio (ORCID)
Attributes

• Shibboleth uses eduPerson
  – E.g., CN, email, telephonenumber, …

• Inconsistently published between federations
  – Attributes published,
  – Values of attributes

• Supporting diverse communities – lowest baseline

• Ought to have user-defined ARP…

• In my opinion, ought to negotiate according to ARP
Building the Infrastructure: Identifying static services

- **X.509 host certificates from trusted CAs**
  - Also trusted by PRACE, EGI, (EUDAT)

- **Browser facing**
  - Commercial or NREN (Terena)
  - Firefox/Windows:
    - Tools → Options → Advanced → Encryption → View Certificates → Authorities
  - IE/Windows: Tools → Internet Options → Content → Trusted Root…

- **Internal/static**
  - As above, or
  - IGTF (www.igtf.net) – covers most countries, or
  - From NRENs

- **Need distribution of used CAs to all hosts**
  - Federation package of trusted CA certificates (like Apache)
Evaluations – 2010

1. Standalone Shib (or SAML)
2. Work with a single community’s portal
3. Use SimpleSAMLPhp (alone)
4. EGI or GEMBUS STS
5. Contrail AAI code
6. Moonshot
Findings

• Code satisfying most requirements least mature
• Need X.509 – at least internally (GridFTP)
• Need good docs for integrators – and effort!
  – Need to be able to work with betas
• Technical collaborations: EGI, EUDAT, Contrail
• Supporting multiple communities:
  – Ends up being kludgy
  – MyProxy for GO, OAuth2 for ORCID, …
• Requirements change regularly
• Can spend $\infty$ time on evaluations
Picking code from Contrail

• Federated identity *login*
• Internally, uses OAuth2 for *delegation*
• Portal obtains an X.509 certificate (via delegation)
• X.509 certificate contains a SAML assertion
  – For authorisation
• Portal manages credential, not user
  – Not user’s browser, either
  – Except for command line access (later)
OAuth2-in-a-slide

Client

Authorisation Server

Owner

Authenticate, permit delegation

Resource

Access token

Access token
Contrail ConSec/Fed architecture

Federation layer:
- PIP
- Fed core
- AuzSvr/CA

Resource layer:
- Prov’ing mgr
- SLAMF
- SLAMP
- PEP
- VEP

contrail
core
Prov’ing mgr
SLAMF
SLAMP
AuzSvr/CA
Fed core
VEP
Web

open computing infrastructures for elastic services
OAuth roles

contrail

open computing infrastructures for elastic services

PIP

Web

Fed core

AuzSvr/CA

SLAMF

 contrail

“Owner”, Client

AuZ svr

Client, Resource

Client

Prov’ing mgr

PEP

VEP

SLAMP

Owner”, Client

Client

“Owner”, Client

Client, Resource

Owner”, Client

Client

Owner”, Client

Client, Resource

Owner”, Client

Client

Owner”, Client

Client, Resource

Owner”, Client

Client

Owner”, Client

Client, Resource

Owner”, Client

Client

Owner”, Client

Client, Resource

Owner”, Client

Client

Owner”, Client

Client, Resource

Owner”, Client

Client

Owner”, Client

Client, Resource

Owner”, Client

Client

Owner”, Client

Client, Resource

Owner”, Client

Client

Owner”, Client

Client, Resource
Plan: Community Portal Integration

• Plan A
  – EUDAT runs an IdP *authenticator*
  – Redirects to trusted external IdPs
  – Certificate service via Oauth
  – Adv: More secure, easier for communities

• Plan B
  – Community manages login
  – Certificate via trusted connections
  – Adv: Simpler than Plan A

• Plan B2: EUDAT separate portal (easier ∫)
• “Plan A = Plan B + OAuth2”
Different identities with ePTID

Home problem: identity changes if home IdP changes (ePPN

Persistent identifier (Australia, new eduPerson revision)
Portal view – GO Integration

Different identities with ePTID

Home problem: identity changes if home IdP changes (ePPN)

Persistent identifier (Australia, new eduPerson revision), Umbrella
WAYF

IdP

Google

Yahoo

Umbrella

WAYF

IdP Bridge

Auz Svr

DB

Account creation
LoA set
Attribute update (eg email)
Authorisation and Access Control

Federated Id

DB

Federation core

=attributes (SAML)

=attributes (SAML)

PIP

Subject.

OK

X reject

+ suspend
Plan: Community Authorisation

“Standard XACML infrastructure”

Community

- Policy ctrl / adm
- Access

Infrastructure

- Policies
- PIP / attrs
- PDP

Resource/site

PEP
Standards

- SAML (OASIS)
- XACML (OASIS)
- X.509 (ITU-T)
- OAuth2 (IETF)
- HTTP (IETF)
- TLS (IETF)
- REST – not a standard, a principle
Authentication workflow

CA

WEB

Fed API

Core

Contrail
IdP

AS

External
IdP
Experiences - Minor Issues

- Certificates (deployment)
  - Need for browser-friendly certificates on browser-facing services
  - Need for trusted certificates on infrastructure hosts
- LoA (1.4?)
- Signing AUP (maintained as federation attribute)
- Mobile access?
- Supporting command line login
  - And iRODS command line access (tickets)
- Portal integration HOWTO (documentation)
- Registration with existing (Shib) feds (deployment)
- Controlling the delegation – still needs user interaction
  - Preauthorise, authorise, or log
Major Issues

• Time/effort/skills needed for integration
  – Hungry student algorithm?

• Sustainability of components (SOA)
  – Use “standard” (open source) components when pos.
  – Maintain components
  – Replace components
  – Do without it
  – Pay someone to support it (or similar)
  – Live with the risk…
End to end demonstrator

1. User goes to community portal and logs in
2. User selects “EUDAT login”
3. Redirect to EUDAT portal (Plan B2)
4. Redirect to authorisation server (AS), which notices user is not logged in
5. AS redirects user to AuC bridge
6. AuC bridge asks user to select IdP and redirects
7. If user is logged in to other portal, home IdP remembers
8. But not the WAYF…?
End to end demonstrator

9. When authentication returns, AuC bridge updates database and creates its own SAML identity assertion for the user, and returns to AS

10. AS validates assertion, and sets up authorisation for the portal to access fed api (or whatever…!)

11. EUDAT Portal obtains access token

12. Portal generates key pair and obtains certificate

13. Now “logged in” to EUDAT

14. Display overview of resources
File access

- **Browser**
  - **HTTP(S)**
- **Portal**
- **Globus Online**
  - **GridFTP(?)**
- **MyProxy**
- **iRODS**
- **PRACE**
  - **GridFTP(?)**
  - **GridFTP(?)**
End to end demonstrator

a. User clicks file link
b. File points at remote file via EUDAT portal
c. Browser requests download of file via portal
d. Portal uses certificate to authenticate to iRODS
e. iRODS extracts SAML assertion and passes to PDP
f. PDP consults policies, PIP, to make decision
g. iRODS grants access (or not) to file, returning data to portal
h. Portal returns data to browser (pipe vs local copy)
Next Steps

• Workshop with EUDAT user communities?
• (More) things to do with EGI…
  – Getting the Contrail credential doing something Useful™
  – Controlling EGI FC resources
• Security evaluations & reporting
  – Security evaluations
• Possibly extensions stuff
  – Moonshot
  – WS-Fed/Trust, Microsoft
The future…?

- Moonshot – [www.project-moonshot.org](http://www.project-moonshot.org)
- Like eduRoam, but for higher level services
  - Carries attributes
- Based on IETF standards
  - RADIUS
  - EAP
  - And OASIS
    - SAML
- Has its own IETF working group (ABFAB-WG)
RADIUS-in-a-slide

EAP: extensible access protocol – *tunnel*
Routing servers can see anonymised credentials
E.g. “@stfc.ac.uk” instead of the tunnelled full identity
The Future (the other kind)

• Managing identities – user perspective
  – Remembering passwords
  – Remembering usernames!
  – Where to log in

• Service provider perspective
  – Accuracy of account information
  – Email addresses
  – Reuse of credentials
Conclusion … of sorts

• Lots of stuff…
• Use small components which know how to do things
• Need expertise in communities
• Spend time analysing, but not too much
• Do not underestimate integration
• Track and contribute to emerging technologies