Adding Federated Identity Management to Openstack

David Chadwick

d.w.chadwick@kent.ac.uk

OpenStack

- Large open source project to develop cloud software
- Latest statistics: 15770 people from 136 countries (3 April 2014)
- Supported by most large IT companies e.g. Dell, HP, RedHat, IBM, Ubuntu, AT&T, Cisco, Hitachi, NEC, Huawei, Intel, Juniper, Yahoo, Ericsson etc. etc.

OpenStack Services (**Project** names)

- Compute – **Nova**, IaaS fabric controller
- Storage – **Swift**, object storage
- Network – **Neutron**, network and IP management
- Dashboard – **Horizon**, user web based interface to OpenStack services
- Identity – **Keystone**, user identity service
- Image – **Glance**, for discovering, registering, and retrieving virtual machine images
- Orchestration – **Heat**, orchestration engine to launch multiple cloud applications based on templates

Authentication in OpenStack

User

Keystone

User database

User Authenticates to

Gets Keystone's token in return

Trust Relationship

Provides Keystone's token to

OpenStack Services
Swift/Nova etc.

Authorisation in OpenStack

- Keystone token contains user's ID and roles
- Services then use either user's roles and RBAC to grant access to resources, or user's ID and DAC

- In order to add FIM to OpenStack we do not need to change any of the OpenStack services provided Keystone still returns the same token as in the non-federated case
  - Services will be ignorant of federation

Why Do Federation?

- Makes it easier for users
  - Less credentials to remember/manage
  - Provides single sign on
- Makes it easier for system developers
  - Don't need to develop secure credential storage or authentication mechanisms and protocols
- Provides much more flexibility
  - Allows any type of authentication mechanism to be easily incorporated since it is "out of scope" of the federation protocol and OpenStack code
- Can make it more secure
  - Users can have one set of strong credentials, so less likely to share them, forget them etc.
  - No longer a honeypot of credentials to be stolen by attackers
- Makes it easier for operations/administration staff
  - No need to register new users, replace lost or forgotten credentials, remove old users

What is Federated Identity Management?

- FIM is primarily about trust management
- But the UK Access Management Federation was purposefully created to require zero trust in each member who joined
  - All that was required was “technical trust” in each member i.e. implemented SAML correctly ☺
  - There is an optional clause concerning trust in identity, but no technical way of validating this initially
- When we added FIM to OpenStack some of the commercial developers thought FIM was about implementing SAML and disregarded trust management

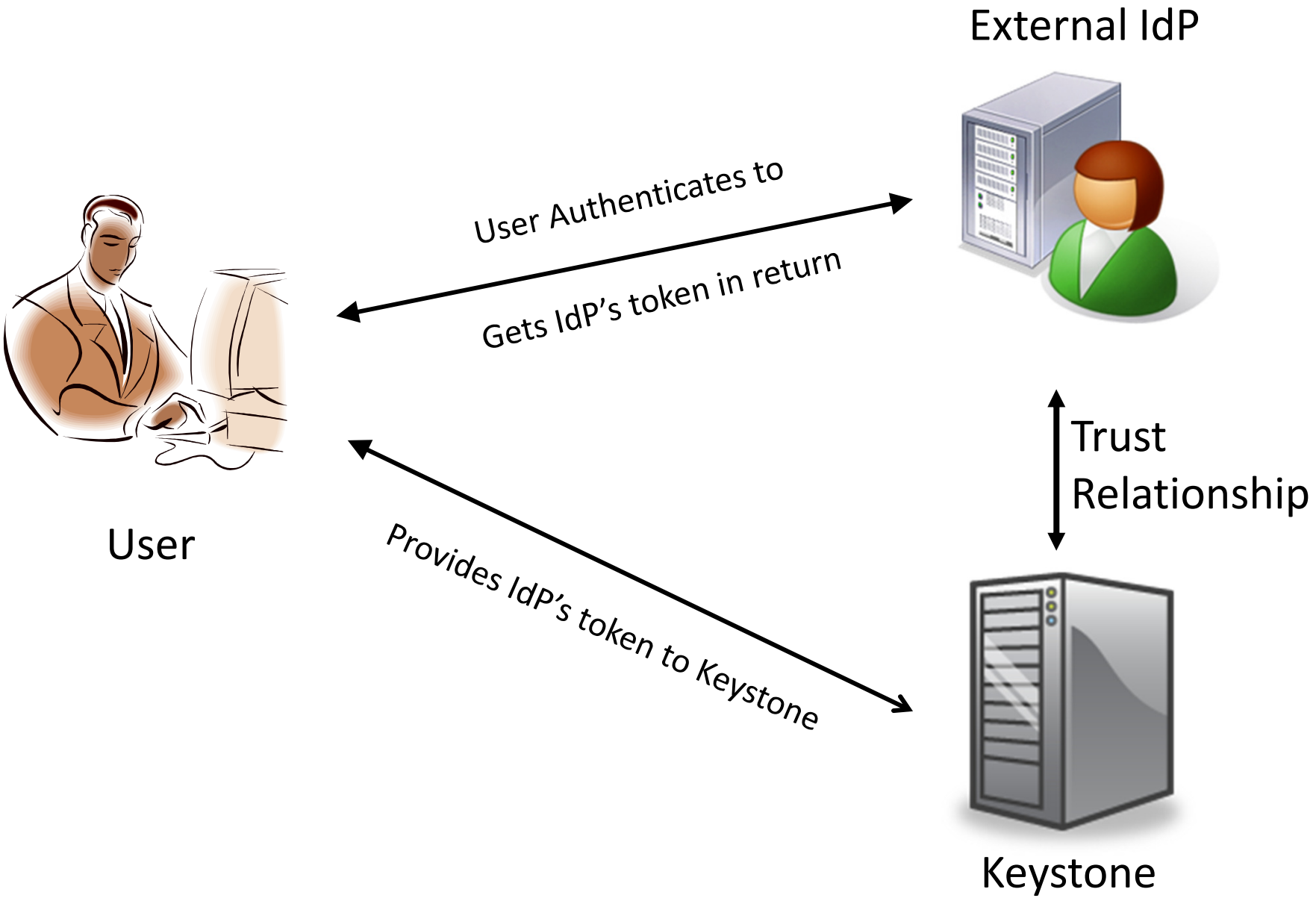
Guiding Principles

- Keep it simple for OpenStack Service Providers (CSPs)
  - All FIM should be done by Keystone (trust management, protocol validation etc.)
- Backwards Compatibility
  - Each OpenStack service should keeps it existing tenants and roles for authz and trusts Keystone to correctly issue them to users
- Keystone must manage all trust relationships with the external IdPs and AAs
- No Religion Please
  - We believe in federation protocols
  - But we don't care what it is called: OAuthv1, OAuthv2, SAMLv1, SAMLv2, X.509, eduroam, OpenID, IETF ABFAB……<enter the name of the next federation protocol here>
  - Federation protocol should be a plugin module

FIM in OpenStack

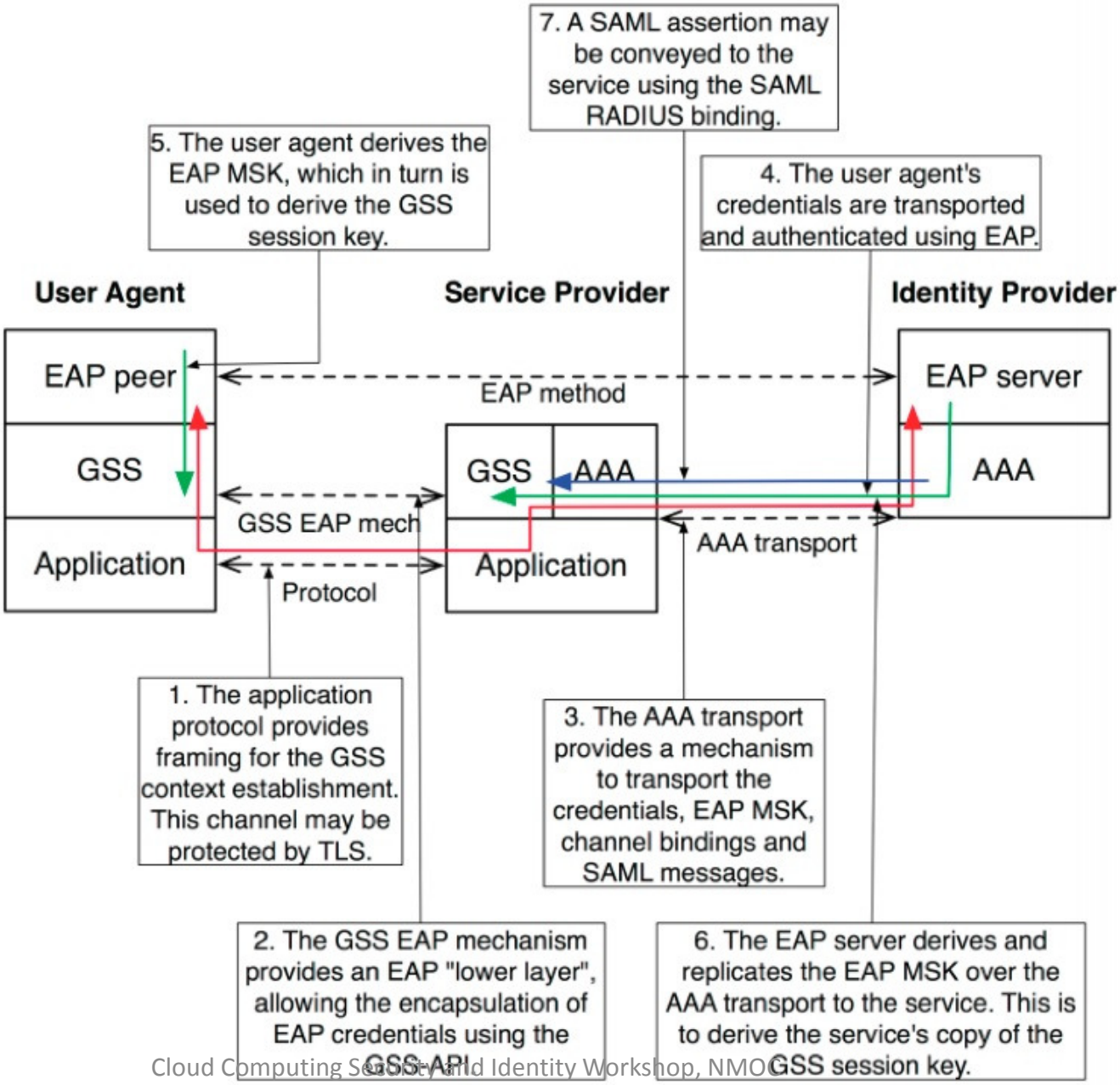
- We implemented FIM in OpenStack to be all about trust management and be independent of the federation protocol (SAML, OpenID, ABFAB etc.)
- 3 trust components to our initial implementation:
- Manage the IdPs that are trusted to authenticate users – AUTHENTICATION TRUST
- Manage the identity attributes that an IdP is trusted to assert – IDENTIFICATION TRUST
- Manage the mappings of trusted identity attributes into OpenStack authz attributes – AUTHORISATION TRUST
- We also need to manage the LoA in each IdP and in each attribute, but this is currently a step too far for the OpenStack developers (though LoA in an IdP can be fudged by converting it into an identity attribute)

Federated Authn with External IdPs



IETF ABFAB – SAML EAP Profile

Picture courtesy of BeSTGRID University of Auckland, NZ



One Protocol Specific Plugin Module

- Three methods
- Get IdP Request – protocol specific request to IdP
- [Negotiate Parameters – optional for those protocols that need it, like ABFAB]
- Validate IdP Response – protocol specific way of validating IdP's response
- Common output at the end

Protocol Output

- Federation wide Unique ID of end user e.g. useruid@idp
- Set of {Set of user identity attributes and name of IdP that asserted them}
  - Caters for attribute aggregation
  - LoA can be included as an identity attribute
- Validity time of asserted identity

Getting our Implementation Accepted in Core OpenStack Release

- Very difficult, has taken 2 years since we started our work
- In April 2014 the OpenStack IceHouse release will have basic support for FIM, but using an Apache front end to handle the FIM protocol and OpenStack as a backend service, and attribute mapping based on our design
- Apache will pass the ID and identity attributes of the user to OpenStack via Http header attributes

Next Steps

- Integral support for FIM in Openstack
  - Apache does not support some FIM protocols e.g. IETF ABFAB and SAML ECP profile
  - End to end security from IdP to Keystone
- Support for Virtual Organisations

Problem 1 – User Access Rights

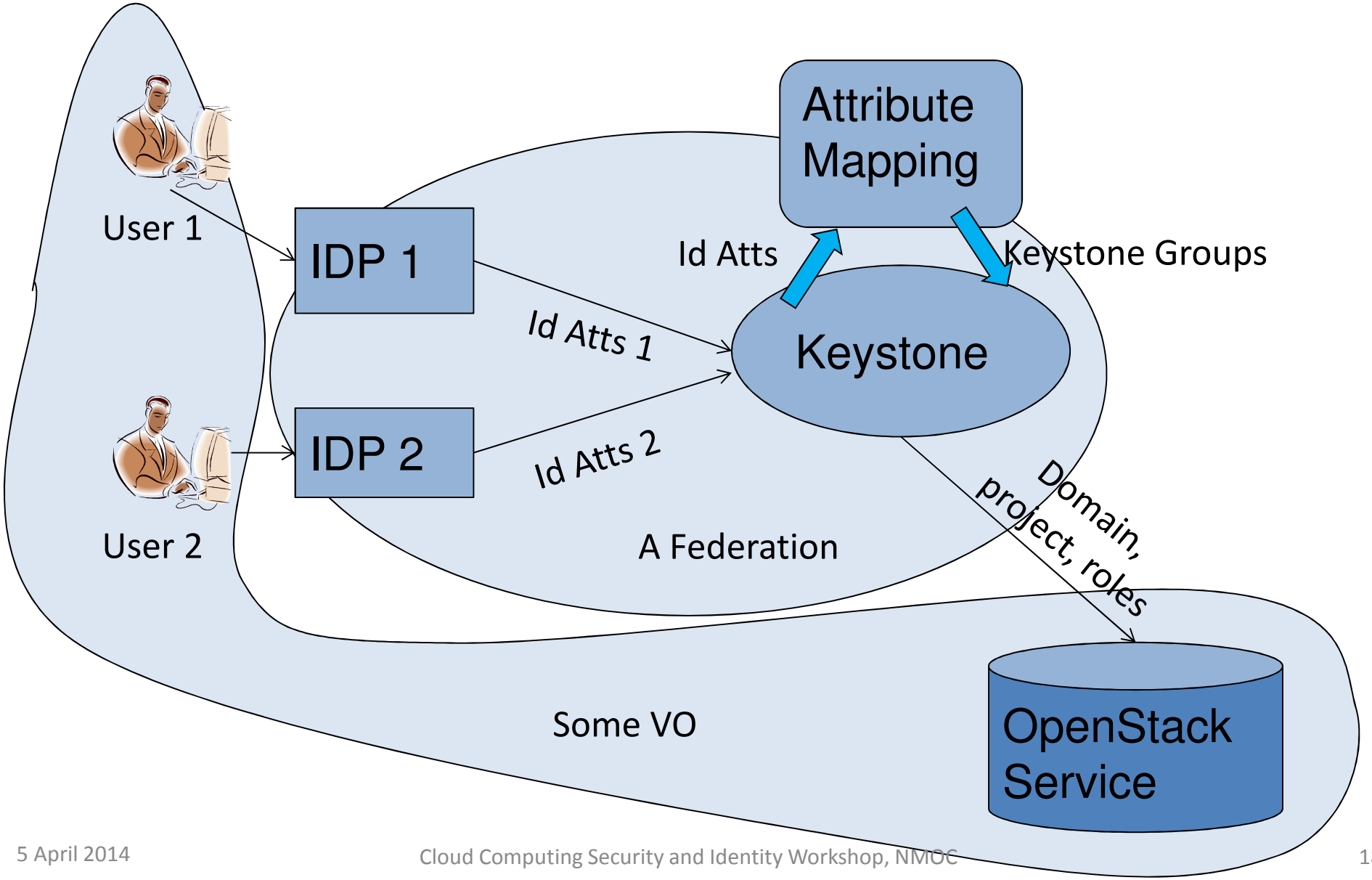
- Not all users from one IdP should have the same access rights at an SP (e.g. OpenStack cloud)
- Different users from different IdPs may have the same access rights at an SP
- Solution 1. Give different users different identity attributes
  - By updating the IdP
- Solution 2. Create a Virtual Organisation (VO) where different users from different IdPs may have the same identity attributes (or roles)
  - Requires updating the IdPs or creating a VO service
- Solution 3. Add attribute mappings at the SP based on the user's unique federation ID and identity attributes
  - Requires updating the SP

Problem 2 – Updating the IdP Asserted Attributes

- IdPs will not assert the attributes that SPs need
- Organisational IdPs typically store user's attributes in corporate LDAP servers, and these are fixed for use by the organisation
- Its very difficult (almost impossible?) for SPs to get the specific attributes they need for authz to be added to corporate LDAP servers
- Federated Solution – standardise a set of user identity attributes that all IdPs will issue and all SPs will consume e.g. EduPerson schema, but this is too restrictive
- Grid Solution - the Virtual Organisation Management Service (VOMS) that allows a VO administrator to give tailored roles to VO users (from different organisations)
  - Need attribute aggregation or VOMS to become an IdP

Integrating VO Management into Keystone

- Use the federation attribute mapping service to form VO roles/groups



Mapping Issue

- VO Administrator typically does not know what attributes or unique ID the IdPs will assert for each VO member
- Neither does the user
  - unless it is a globally unique name like ABFAB Network Access Identifier - username@realm
  - but still may not know which attributes are being asserted
- Solution – VO Role Registration by Invitation

Invite Users to Register to a VO Role

- VO Administrator creates a Keystone group (VO role) and gives it permissions (domains, projects, roles)
- Sends group name and PIN to invited VO members out of band
- VO member authenticates to Keystone via his/her IdP
- Asks to join group and provides PIN
- Keystone automatically creates a mapping rule from user's unique IdP asserted ID to group name (or can be held in pending queue for admin to OK it first)

Publication

David W. Chadwick, Kristy Siu, Craig Lee, Yann Fouillat, Damien Germonville. “Adding Federated Identity Management to OpenStack”. Journal of Grid Computing. Dec 2013. http://dx.doi.org/10.1007/s10723-013-9283-2

Any Questions