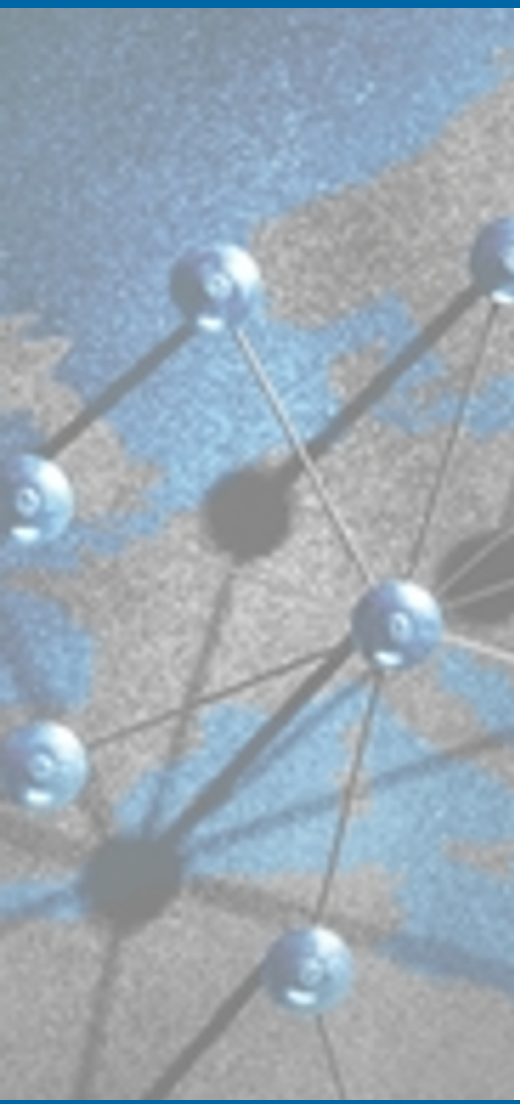


VOMS-aware identity service for Openstack

A. López García, *E. Fernández, M. Puel*

Instituto de Física de Cantabria (CSIC-UC) — Centre de Calcul IN2P3 (CNRS)

e-infrastructure

Outline

Outline

Introduction to Openstack Architecture

Openstack is based on several components

- compute service (nova)
- store service (swift)
- image service (glance)
- identity service (keystone)
- block storage service (cinder)
- network service (quantum)
- dashboard service (horizon)

Introduction to Openstack Architecture

Openstack is based on several components

compute service (nova) Spawns and manages the instances.

store service (swift)

image service (glance)

identity service (keystone)

block storage service (cinder)

network service (quantum)

dashboard service (horizon)

Introduction to Openstack Architecture

Openstack is based on several components

compute service (nova) Spawns and manages the instances.

store service (swift) Object storage.

image service (glance)

identity service (keystone)

block storage service (cinder)

network service (quantum)

dashboard service (horizon)

Introduction to Openstack Architecture

Openstack is based on several components

compute service (nova) Spawns and manages the instances.

store service (swift) Object storage.

image service (glance) Image catalog, store and retrieval.

identity service (keystone)

block storage service (cinder)

network service (quantum)

dashboard service (horizon)

Introduction to Openstack Architecture

Openstack is based on several components

compute service (nova) Spawns and manages the instances.

store service (swift) Object storage.

image service (glance) Image catalog, store and retrieval.

identity service (keystone) Authentication, Authorization, etc.

block storage service (cinder)

network service (quantum)

dashboard service (horizon)

Introduction to Openstack Architecture

Openstack is based on several components

compute service (nova) Spawns and manages the instances.

store service (swift) Object storage.

image service (glance) Image catalog, store and retrieval.

identity service (keystone) Authentication, Authorization, etc.

block storage service (cinder) Provides block device storage (a-la EBS).

network service (quantum)

dashboard service (horizon)

Introduction to Openstack Architecture

Openstack is based on several components

compute service (nova) Spawns and manages the instances.

store service (swift) Object storage.

image service (glance) Image catalog, store and retrieval.

identity service (keystone) Authentication, Authorization, etc.

block storage service (cinder) Provides block device storage (a-la EBS).

network service (quantum) Provides network connectivity.

dashboard service (horizon)

Introduction to Openstack Architecture

Openstack is based on several components

compute service (nova) Spawns and manages the instances.

store service (swift) Object storage.

image service (glance) Image catalog, store and retrieval.

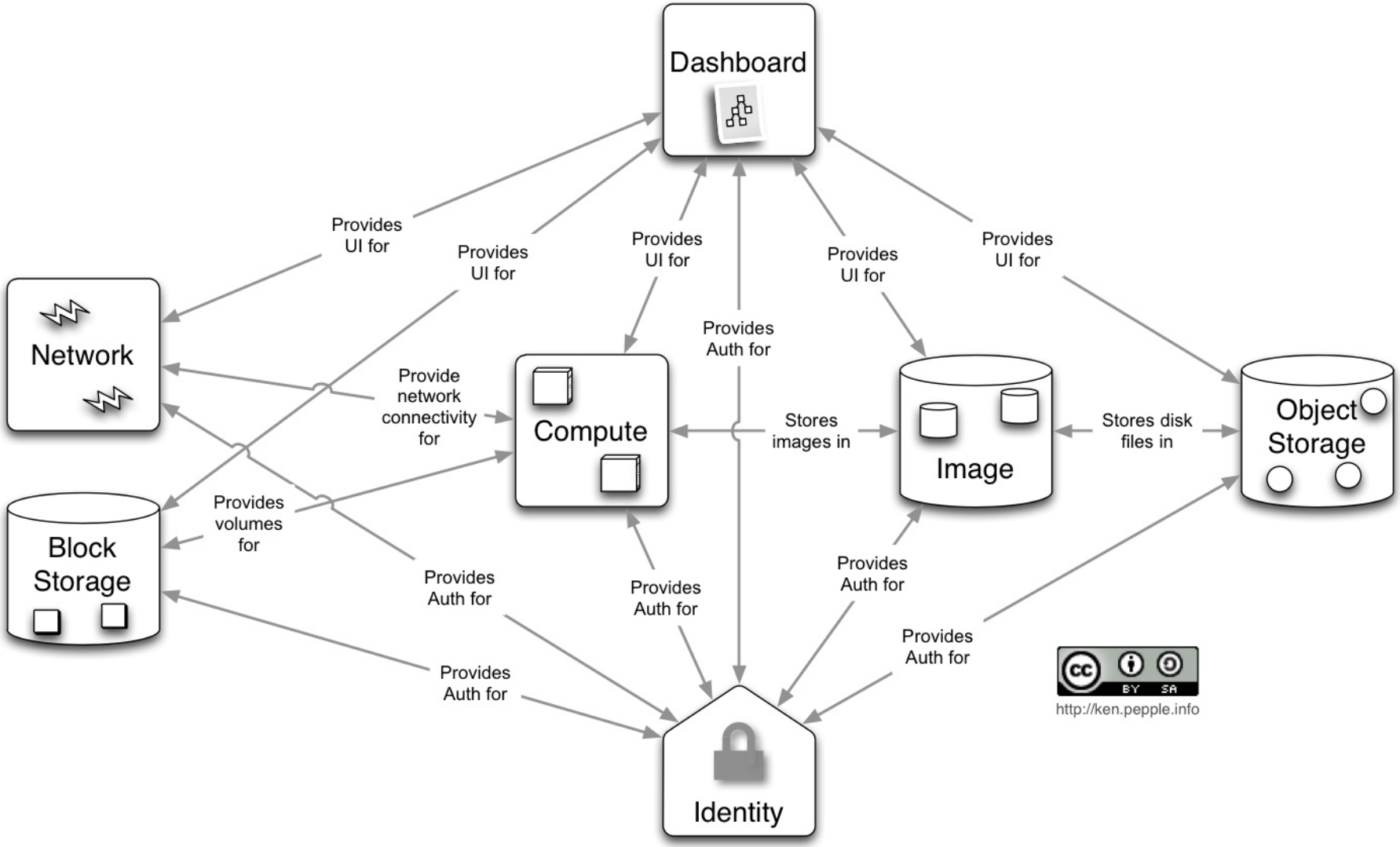
identity service (keystone) Authentication, Authorization, etc.

block storage service (cinder) Provides block device storage (a-la EBS).

network service (quantum) Provides network connectivity.

dashboard service (horizon) Web interface.

Openstack Architecture



AuthN and AuthZ in Openstack

- Authentication and Authorization is orchestrated around the identity service *Keystone*.
- Auth is based on users, tenants, domains, roles and tokens.
  - A user is member of 1 or more tenants.
  - A tenant (group, project) is part of 1 or more domains.
  - A user may have specific roles within a tenant or globally within a Keystone domain.
  - A token may be associated with a tenant or not:
    - Unscoped tokens are not associated with a tenant. Used for discovery (available tenants, endponts) and are only understood by keystone
    - Scoped tokens are associated within a tenant and are required to interact with any other component.
  - A token can be unsigned (UUID) or signed (PKI based).

AuthN and AuthZ in Openstack

- Authentication in Keystone is a 2 part mechanism.
  - 1st phase: A user initiates authentication against Keystone and a token is issued.
  - 2nd phase: The token is used to authenticate against all the other Openstack services.
- All authenticated requests require a scoped token.
- A token has a limited validity.
  - Valid within only one tenant.
  - Fixed expiration time.
- The token is verified with each of the requests by all of the Openstack components.
  - UUID tokens are validated online: it requires a call back to the Keystone server.
  - PKI tokens can be verified offline: signed message.
- Role based authorization (RBAC).

Federation Efforts

- EGI and the NGIs are already a federation of resources providers.
- Many of the resource providers have some virtualized and/or cloud resources.
  - Many different software stacks.
  - Different interfaces.
  - Different capabilities.
- Many users are interested in accessing those resources.
  - Profit from flexibility.
  - Deploy its own software environment.
- Authentication is the cornerstone of such a Federation

Federated Identity

Cloud federation across several providers has some consequences for RPs:

- Manage large numbers of users, in different (and sometimes overlapping) groups.
- Populate the user base to each of the sites involved in the federation.
- Each site is free to use a different cloud middleware with its own AuthN/AuthZ mechanism.
- EGI Fedcloud requires that it is X.509 and not user/password based.
- Same situation as in the Grid $\rightarrow$ VO based authentication might be the answer.

For the users and communities

- Manage different credentials: user/password based, X.509 proxies, VOMS...
- Readapting the existing tools.
- Difficult to manage users for the communities.

Outline

VOMS Authentication in the Cloud

Apply VOs to the Cloud using VOMS-based authentication.

- Widely used in the Grid. Infrastructure already in place (PKI, VOMS servers, portals, etc.)
- User communities are familiar with it.
  - No extra credentials for users
  - No extra effort for managers.
  - No transition effort.
- Resource providers are familiar with it.
  - No extra effort for configuration.
  - No extra effort on their side to allow a VO to execute.
- Grid tools can be easily adapted to interact with cloud testbeds
- Integrated (or possible integration) with other operational tools.
- Extensible (for example it is possible to move towards SAML).

Keystone VOMS AuthN middleware

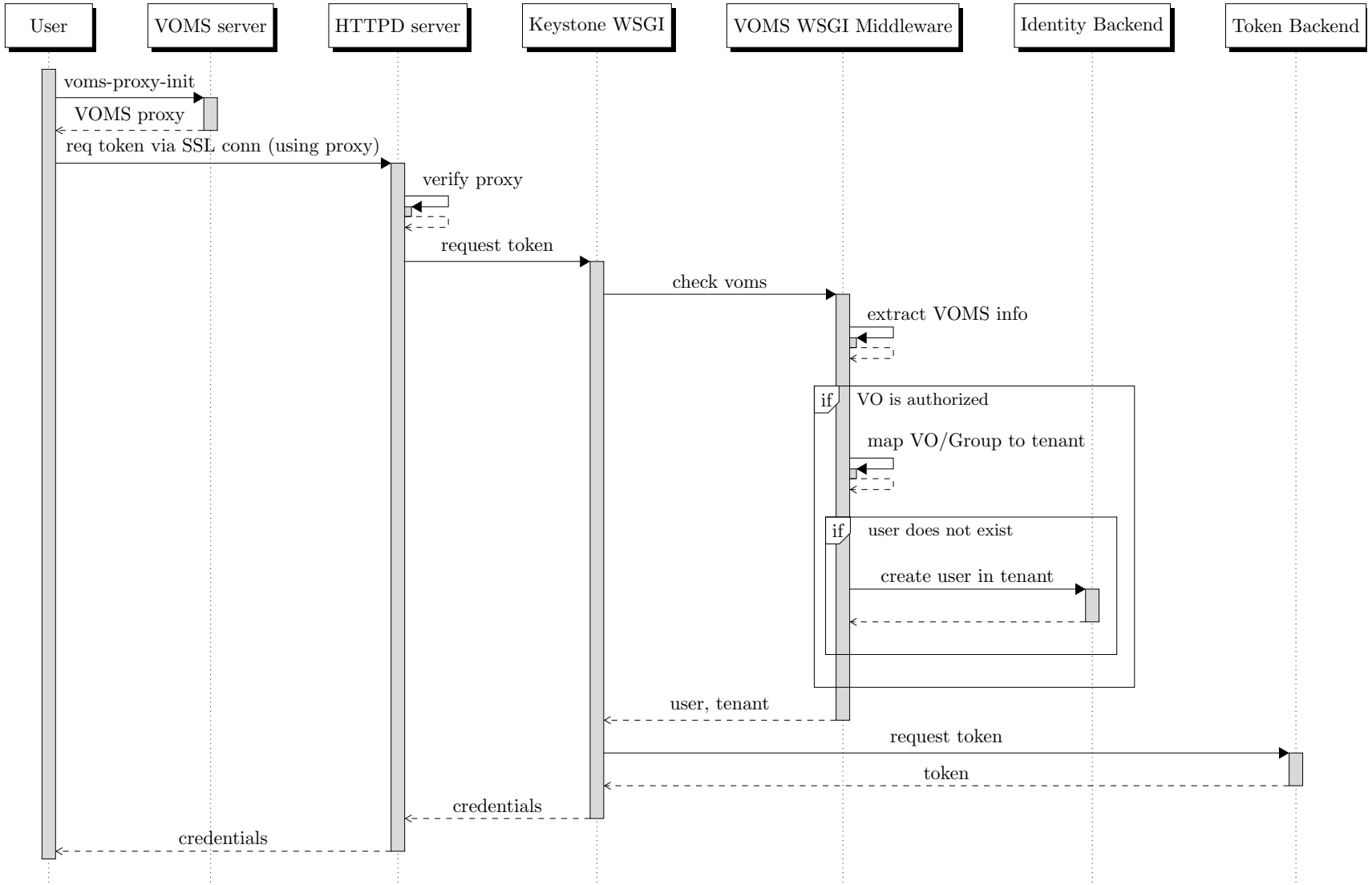
Deployment.

- Keystone is a WSGI application.
- Keystone is deployed behind Apache (or other HTTPD server).
- The HTTPD server verificates the X.509 proxy: validity, CA, CRLs.

VOMS module.

- WSGI middleware filter.
- Add-on to the Keystone server, no need for patch or modification.
- The VOMS proxy should be authenticated upstream (by the HTTPD server).
- Extracts the VO info from the VOMS proxy and maps it to a user, internal tenant and domain.

VO support in Openstack



Openstack Client AuthN module

- Pluggable authentication mechanism has been contributed to the mainline from version 2.13.0.
- VOMS auth module available for novaclient.

```
$ git clone https://github.com/IFCA/voms-auth-system-openstack
$ cd voms-auth-system-openstack
$ python setup.py install
$ voms-proxy-init -voms VONAME -rfc
$ nova --os-auth-system voms --x509-user-proxy /tmp/proxy credentials
```

Outline

Perspective and Future work

Several Openstack sites using it so far.

- CC-IN2P3 (France), IFCA (Spain), IISAS (Slovakia), Jüelich Supercomputing Center (Germany).

Perspective and Future work

Several Openstack sites using it so far.

- CC-IN2P3 (France), IFCA (Spain), IISAS (Slovakia), Jüelich Supercomputing Center (Germany).

Integration with applications.

- VOMS modules for Openstack clients.
- libcloud (Apache) fork to support VOMS auth.
- Dirac module.

Perspective and Future work

Several Openstack sites using it so far.

- CC-IN2P3 (France), IFCA (Spain), IISAS (Slovakia), Jüelich Supercomputing Center (Germany).

Integration with applications.

- VOMS modules for Openstack clients.
- libcloud (Apache) fork to support VOMS auth.
- Dirac module.

Future steps.

- Exploit VOMS roles for RBAC in Keystone.
- Study the possibility of SAML assertions instead of proxies.
- Integration with Openstack dashboard (horizon).

Thanks!

Talk is cheap. Show me the code.

- Keystone module:
  https://github.com/IFCA/keystone-voms
  - Documentation:
    https://keystone-voms.readthedocs.org/en/latest/
- Client module:
  https://github.com/IFCA/voms-auth-system-openstack

mailto:aloga@ifca.unican.es