Fermilab

Managed by Fermi Research Alliance, LLC for the U.S. Department of Energy Office of Science

X.509 Authentication and Authorization in FermiCloud

Hyunwoo KIM, Steven C TIMM,
Fermilab Scientific Computing Division
UCC(Utility/Cloud Computing), Cloud Federation Management
London, 08 December 2014

Contents

- ✓ FermiCloud general introduction

- ✓ FermiCloud with OpenNebula (ONe)
  - FermiCloud Contribution: X.509 Authentication Module
  - New Development: Prototype of role-based X.509 Authorization based on Open Science Grid (OSG)

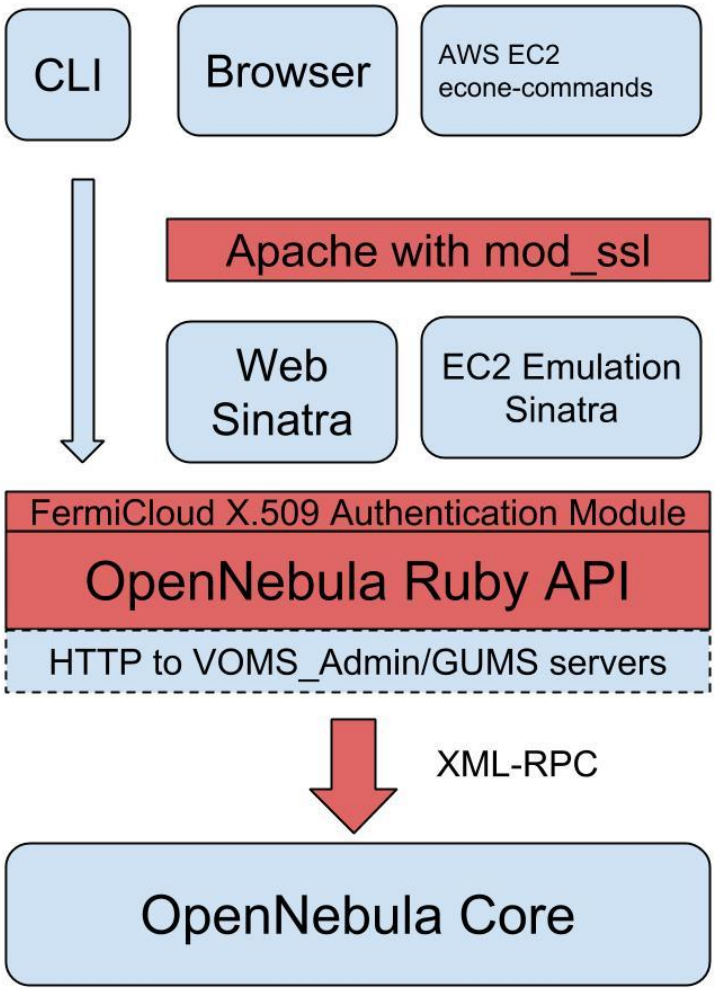
- ✓ Survey for Federation
  - One of FermiCloud future agenda: Federation
  - Use of VOMS by EGI FederatedCloud with OpenNebula
  - OpenStack has 2 extensions: VOMS and Shibboleth

FermiCloud Project

- ✓ Small-Medium private cloud, in service since 2010
  - We serve scientific stake-holders: e.g. OSG and Fermilab experiments users
  - production/pre-production clusters: 29 physical nodes
    - 16-core Intel Xeon CPU with 48 GB memory
  - test-bed: ~100 nodes for a demonstration of 1000 virtual machines
- ✓ We use OpenNebula.
  - ONe 3.2's been running 3 years successfully => Upgrade to 4.8
  - OpenStack is being evaluated (authN/authZ)
- ✓ Fermilab is using Kerberos for general authentication
  - Only short-lived KX.509 certificates are allowed for FermiCloud
- ✓ Part of job provisioning system for Grid/Cloud
  - Jobs can be deployed to FermiCloud when Grid resources run short
  - Jobs can also go to commercial cloud resources(AWS EC2)
  - Mass-launching of on-demand worker nodes in EC2
  - Also looking into launching on-demand batch system in EC2

Overview: Authentication/Authorization in OpenNebula

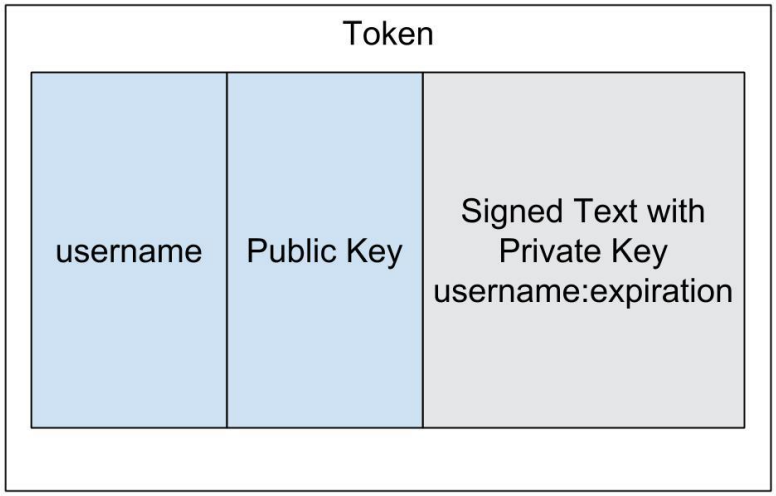
- ✓ All interfaces go through Ruby API
- ✓ Authentication
- ❖ FermiCloud contribution:
  - In ~2010, most cloud services were using long-lived password
    - short-lived credentials (KX.509) allowed
  - X.509 authentication module developed as new module in ONe Ruby API
  - CLI: directly uses this
- ✓ Web/EC2 interfaces: Ruby Web Server going through Apache/SSL
  - internally same X.509 authN module used
- ✓ Authorization
- ✓ We need more than ACL
  - ✓ fine-grained, external authZ service
- ❖ We developed a prototype based on OSG authorization model
  - Use extended attributes service (VOMS)
  - Use XACML PDP and ID mapping



X.509 Authentication Module Details

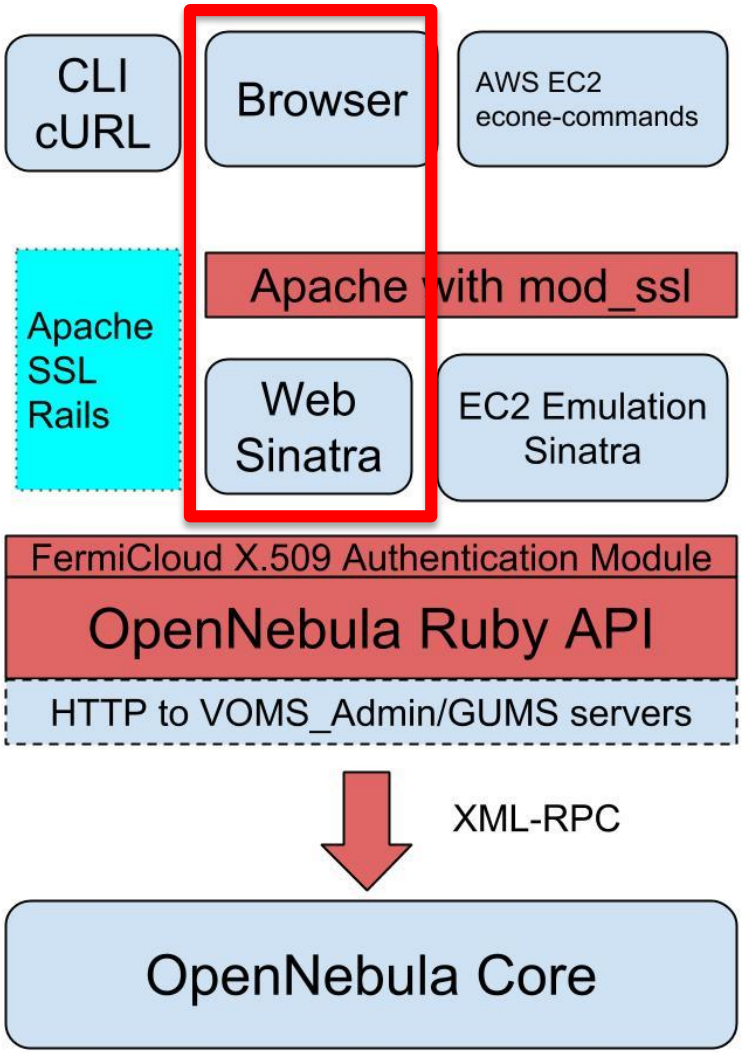
- ✓ login command
  use private key to encrypt
  (username:expiration)
  attach the public key and username

- ✓ the rest of the commands:
  transmit the token to the server
  server uses the public key to decrypt
    the cypher-text and compare with
    the plain username
  the server makes sure that the token
    has not expired



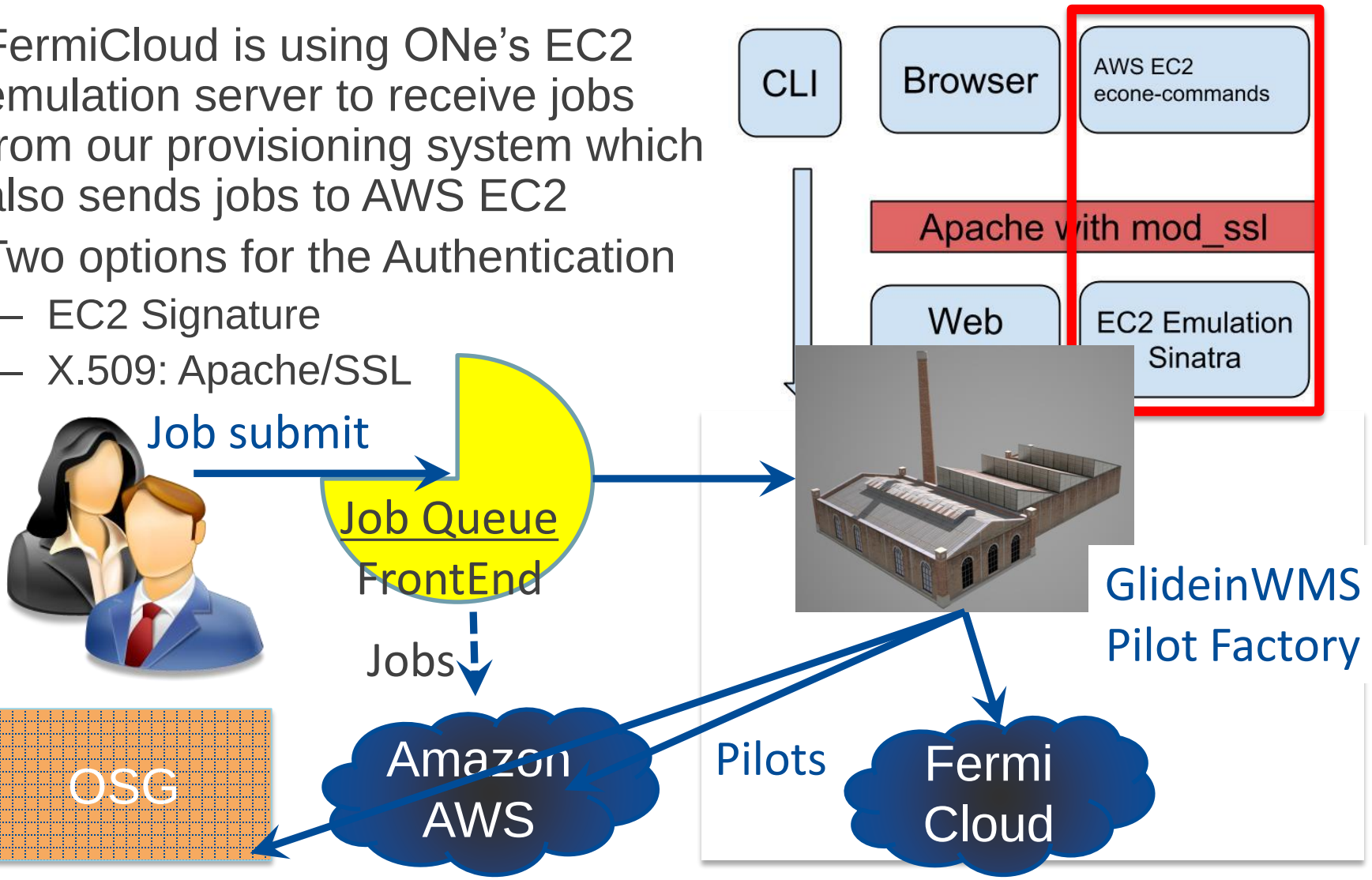
Web Interface: Ruby Web Servers/Frameworks

- ✓ Web interface is Ruby Web server/framework(Sinatra)
  - behind Apache/SSL
- ✓ Can we use cURL command for OpenNebula CLI?
- ✓ The rOCCI places Apache/Rails in front of Ruby API for CLI
  - ONe Web/EC2 interfaces have Sinatra/Rack/Thin in front of Ruby API
- ✓ Either way, it would be possible to place REST server to enable cURL command with the same X.509-based token



OpenNebula EC2 Emulation Interface

- ✓ FermiCloud is using ONe's EC2 emulation server to receive jobs from our provisioning system which also sends jobs to AWS EC2
- ✓ Two options for the Authentication
  - EC2 Signature
  - X.509: Apache/SSL



FermiCloud AuthZ Prototype based on OSG AuthZ Model

- ✓ Open Science Grid Authorization Model
- ✓ Use identities extended with attributes(Fully-Qualified Attributes Names)
  - Virtual Organization Membership Service(VOMS)
  - Users interact with VOMS to get attribute-enhanced credentials
- ✓ Use XACML for authorization based on user attributes
  - Grid User Management system (as XACML PDP)

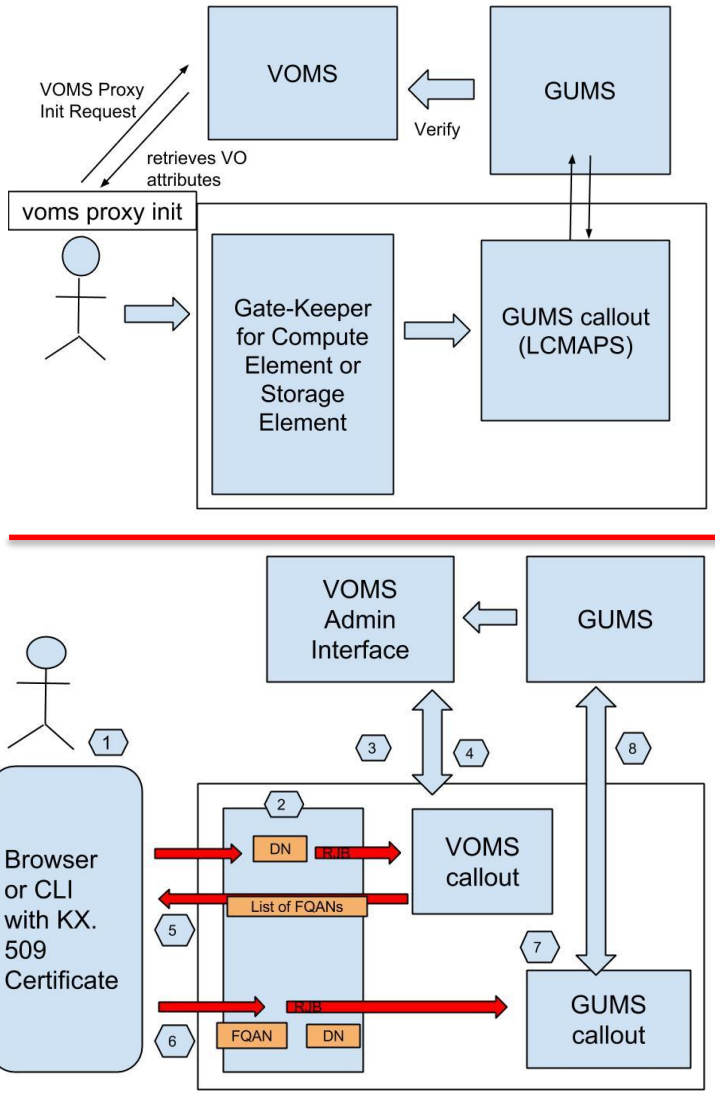
- ✓ New features in FermiCloud Authorization
  - Web Browser constraint
    - fail to transmit both voms-signed proxy & personal certificates
  - Instead, use new call-out(Java), needs FQAN, not the whole certificate. How do we acquire FQAN?
    - use voms-proxy-init and GridSite will extract FQAN
    - use KX.509 certificate with VOMS Admin interface

- ❖ We have developed a prototype for this:

1. User using KX.509 certificate with browser/CLI
2. User Distinguished Name extracted
3. Callout via RJB to VOMS_Admin interface
4. returns a list of user FQANs
5. Prompt for user selection of one FQAN
6. Now DN and FQAN both available)
7. Callout via RJB to GUMS
8. Decision is made by GUMS

GUMS returns user's group id(VO), can label VM (with GID/VO)

RJB=Ruby Java Bridge



Fermilab

VO and Beyond

- ✓ VOMS is a good method for identity federation
  - We might need to alter security policy to support non-Kerberos-based certificates access to cloud resources at Fermilab
  - We are interested in finding use cases
- ✓ VOMS + X.509 in EGI FederatedCloud
  - rOCCI(Apache/SSL/Rails) in front of ONe Ruby API
    - GridSite for FQAN and contact VOMS to verify voms proxy
- ✓ OpenStack appears to provide
  - Regular: curl keystone-url -d JSON(username/password)
  - VOMS: curl keystone-url --cert -d JSON(voms:true)
  - Shibboleth: curl shibboleth-url, redirect to SAML IdP

Summary

- ✓ FermiCloud developed and contributed new authentication module using X.509-based short-lived token to OpenNebula
  - there's an emerging pathway in all of these different clouds towards short-lived tokens.
  - authentication via SSL for Web and EC2 Servers
- ✓ FermiCloud has developed a prototype authorization module
  - based on Open Science Grid authorization model
  - role-based fine-grained authorization
    - VOMS: identity and attributes management
    - GUMS: policy/decision service
  - VO is used for identity federation and accounting purposes
- ✓ We will continue to monitor federation technologies/trends

VOMS and SAML in OpenStack Keystone

regular Keystone

curl http://LH:5000/v2.0/tokens -d

{auth:

{ tenantName: admin,

passwordCredentials:

{username:admin, password: secret}

}}

KS-VOMS module

voms-proxy-init -voms fermilab:/fermilab

curl https://LH:5000/v2.0/tokens --cert proxy -d {auth:{voms: true}}

voms.json maps incoming VO to the user

Is this interesting to us?

OS-FEDERATION extension

curl http://LH:5000/v3/OS-FEDERATION/identity_providers/A/protocols/B/auth

identity provider will be contacted via SAML(assertions) or OIDC(claims)

service provider issue unscoped token

curl http://LH:5000/v3/auth/tokens

{auth:

{

identity:{saml2:{id:unscopedtoken} },

scope:{}

}

}

OAuth 2.0 in Google Cloud Platform (as part of survey)

- OAuth 2.0 appears to be not so relevant to us
  - AuthN/AuthZ for applications on behalf of users
- The following is all happening inside my Python application
  - I only need to acquire a credential(client_secret) and let my app have it
- My App contacts the OAuth AuthZ Server with client_secret
  - https://accounts.google.com/o/oauth2/auth?client_id=4&scope=openid
    - OpenID Connect prompts for my google password (user authN)
  - https://accounts.google.com/o/oauth2/auth?client_id=4&scope=bquery
    - one-time authentication code is returned(app authN with client_secret)
- My App contacts AuthZ Server to obtain Access Token
  - POST /o/oauth2/token HTTP/1.1(HTTP Header, request line)
  - Host: accounts.google.com (HTTP Header, MIME field)
  - Returns ID Token, Access Token, Refresh Token
    - Use ID Token for identification
    - Refresh Token can be used to simply renew Access Token alone
- Use Access Token for the actual service

OpenNebula EC2 Emulation Interface

- ✓ FermiCloud is using ONe's EC2 emulation server to receive jobs from our provisioning system which also sends jobs to AWS EC2
- ✓ EC2 emulation server responds to EC2 Query Request
  - It finds EC2 Actions embedded in the HTTP messages
  - then it maps EC2 Action to OpenNebula internal requests
- ✓ Two options for the Authentication
  - EC2 Signature
  - X.509: Apache/SSL