T2_US_VANDERBILT Overview

Eric Appelt and Andrew Melo

CMS Distributed Computing Model

Tier-0: T0_CH_CERN

- Store RAW detector data, backup to tape
- Prompt data reconstruction
- Distribute data to Tier-1 sites

Tier-1: T1_US_FNAL, etc…

- Typically national lab or other large facility
- Data reconstruction and tape backup
- Distribute analysis data to Tier-2 sites

Tier-2: T2_US_Vanderbilt, T2_US_MIT, et…

- Typically University sites
- Perform physics analysis on detector data
- Produce simulated data

CMS Computing Partners

- Open Science Grid (OSG) - NSF-funded project responsible for much of the software/infrastructure. Technically "experiment-agnostic" and works to support all US-based scientific computing, but CMS effectively runs the show (most of the management are USCMS)

- Worldwide LHC Computing Grid (WLCG) - Umbrella group responsible for all LHC experiments globally. Experiments estimate compute reqs. and provide them to the WLCG, member countries then contribute compute in proportion to the % of scientists in that country ("the pledge"). Runs the SAM test infrastructure.

What T2_US_Vanderbilt Must Provide

- Scheduled access to Physical CPU cores and memory
- CMS Software and environment on the server
- Storage of CMS datasets and access to files
- High-bandwidth transfers of data between sites

CMS Computing Hardware Managed By ACCRE

Dell R420 (older) 1U server

Dell C6420 (newer) 2U chassis housing 4 servers

- Mostly dual-socket Intel CPU or AMD CPU
- Accessible through slurm scheduler, shared with other cluster users
- Two partitions/environments:
  - “batch” - primary cluster partition for all ACCRE users
  - “nogpfs” - nodes that are not accessible to most users, not connected to GPFS filesystem. Includes older nodes past end-of-life retained opportunistically
- All running CentOS 7 (RHEL 7 variant) or Rocky 9 (RHEL 9 variant) linux with ACCRE configuration

Typical Job Submissions (local ACCRE users)

ACCRE User

User writes slurm script, runs sbatch to submit

Proxmox Hypervisor Cluster

Public Gateways

login.accre.vu

Job added to slurm queue

sched.vm.accre.vu

Home directory
Submission script
Job logging

When resources are available, slurm script launched on compute node

Compute Node

cn1523.vampire

PanFS Filesystem Appliance

Home directory
Submission script
Job logging

Typical CMS Grid Submissions

CMS User

User writes CRAB job specification runs crab submit

CERN

login.cern.ch

Submitted to CRAB server

CRAB Servers

SchedD process

glideinWMS system matches jobs with waiting processes on ACCRE compute nodes

Collector process

Negotiator process

OSG Servers

CMS Compute Elements

Proxmox Hypervisor Cluster

NFS

ce5.accre.vanderbilt.edu

NFS

ce6.accre.vanderbit.edu

Job payload
Logging
State information

StartD process

StartD process

Compute Node

cn1523.vampire

How are the StartD processes running on the Compute Nodes???

Pilot Job Submissions

glideinWMS Factory

Sends pilot jobs to compute elements

Keeps sending to ensure N jobs are always pending

CMS Compute Elements

NFS

ce5.accre.vanderbilt.edu

NFS

ce6.accre.vanderbit.edu

Translate pilot job specs into slurm submissions suitable for ACCRE

Virtualized Slurm Scheduler

sched.accre.vu

Launch jobs when resources available

Job payload
Logging
State information

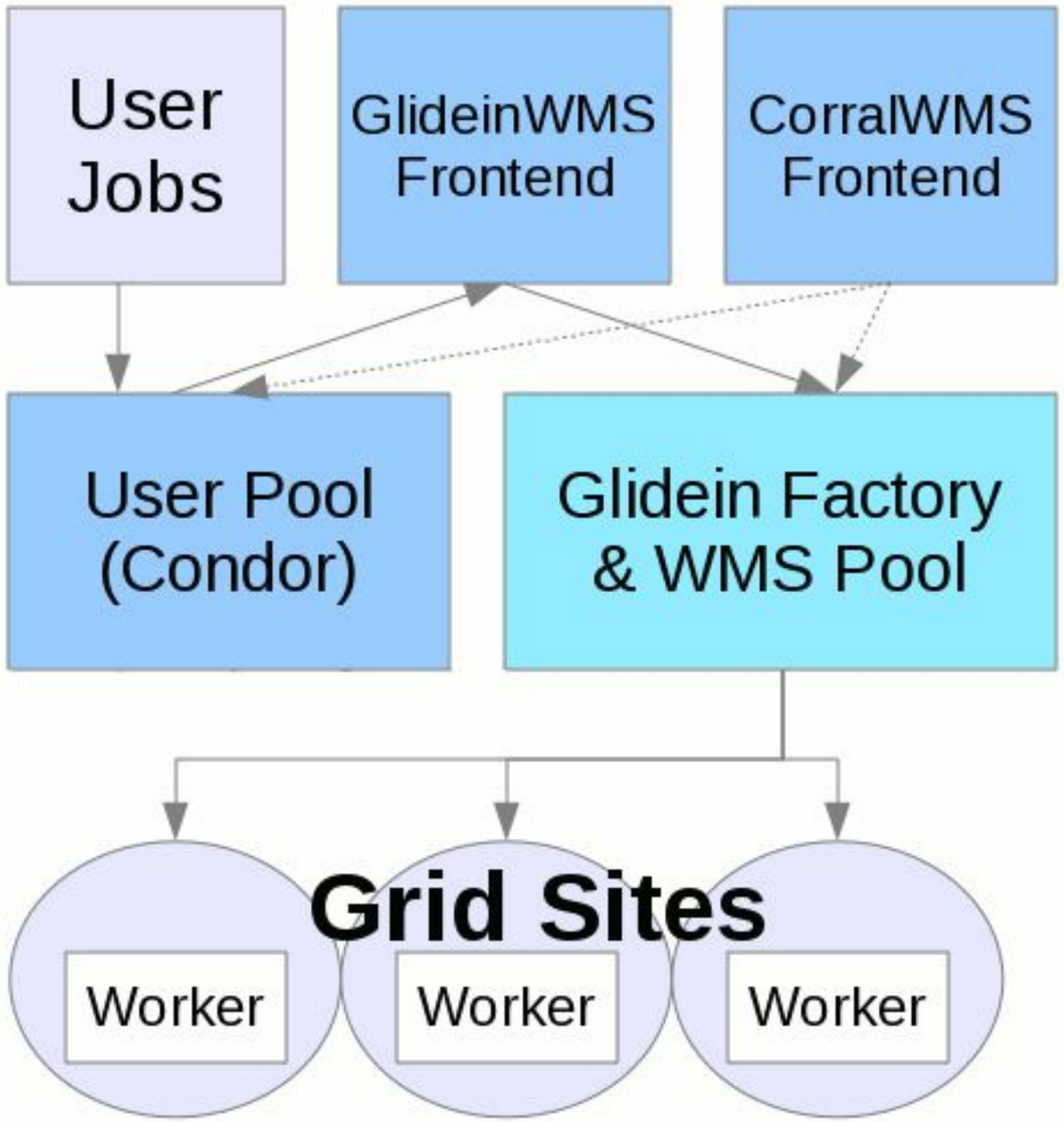
StartD process

Compute Node

cn1523.vampire

OSG Collectors And Negotiators

Pilot Job Submissions



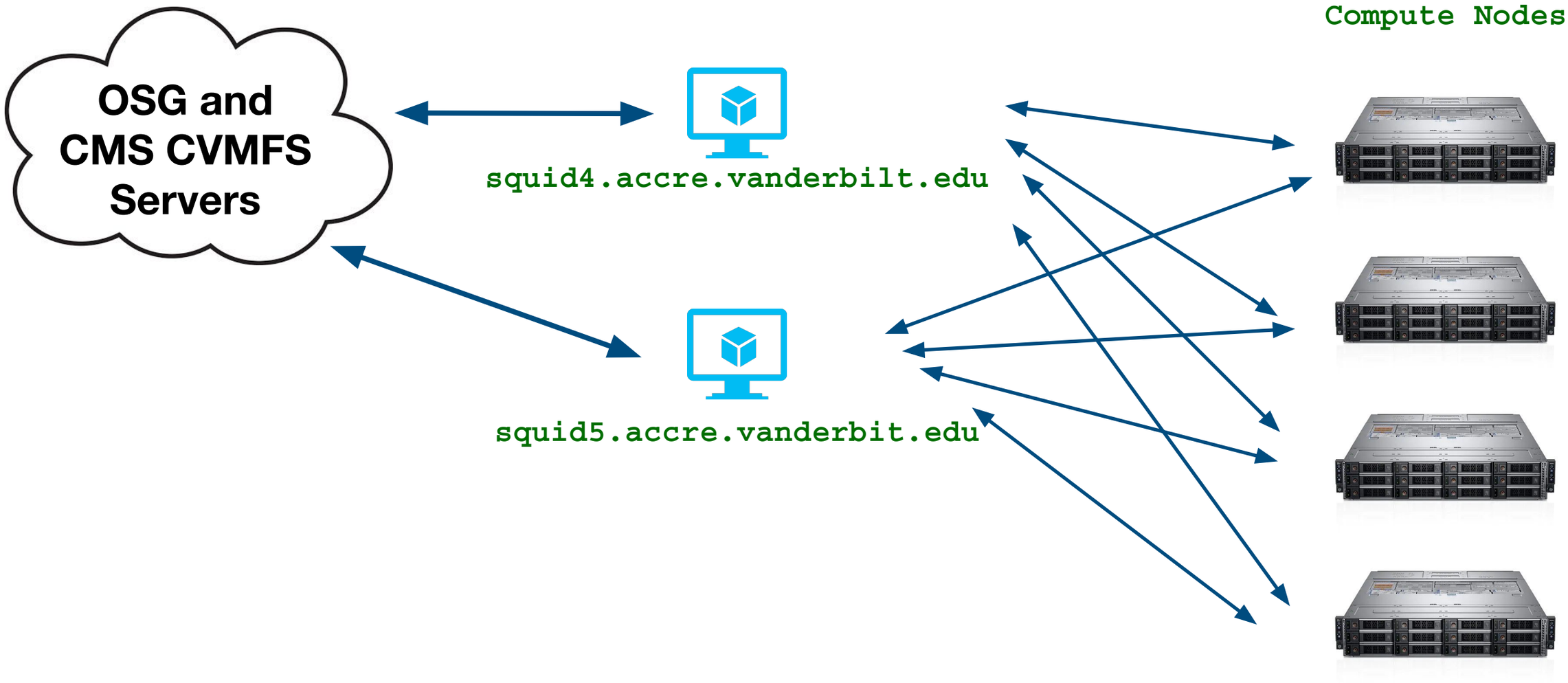
https://glideinwms.fnal.gov/doc.prd/images/simple_animation.gif

How do these jobs access CMS software and ensure a compatible environment???

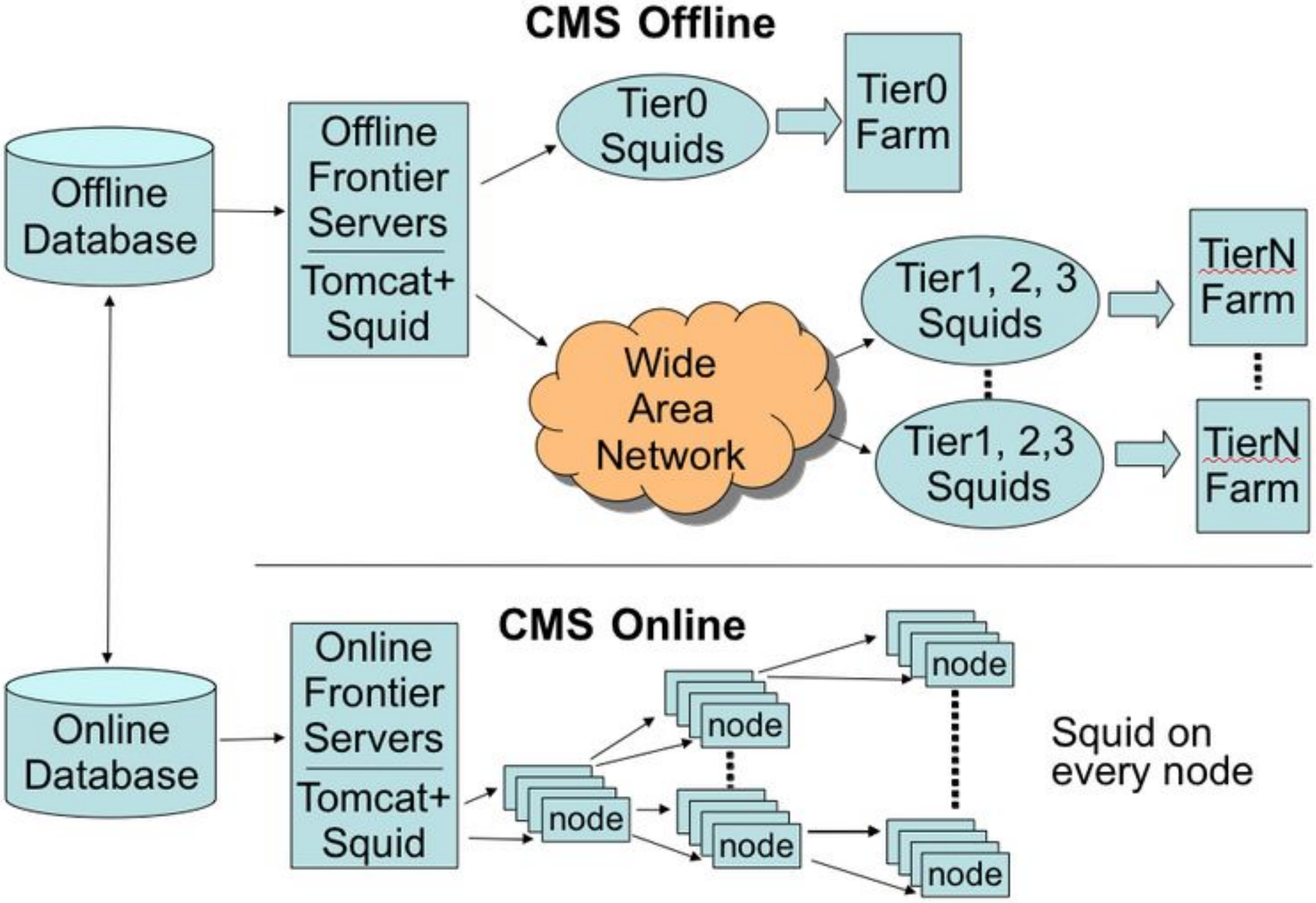
Software and Environment

- Standard Environment Using Apptainer (was Singularity) Containers
  - Containers provide an isolated environment mimicking running on a different linux OS with specified library versions
  - Similar to Docker without root privileges
  - CMS provides containers
  - ACCRE configures singularity on compute nodes
-
-
- CMS Software Distributed using CVMFS filesystem
  - Read-only immutable filesystem designed by CERN for software
  - Uses HTTP for data transfer
  - Sites use squid caching proxies to reduce load on central servers
  - Files available on all ACCRE compute nodes
  - `/cvmfs/cms.cern.ch`
  - `/cvmfs/singularity.opensciencegrid.org`

CVMFS Servers at ACCRE



CMS CVMFS Architecture



How is data stored and accessed???

Data Storage

- LStore Filesystem - Developed at Vanderbilt
- We house ~10PB currently
- Data can be accessed using special LStore commands or as a native linux filesystem using FUSE
  - CMS jobs access via "LFS FUSE mounts" which appears like a regular filesystem
- Single “LServer” machine coordinates communications between clients and storage servers, aka “depots”
- Data stored on “depots”
  - Linux servers with 10Gbit network connections
  - 36 bays for HDDs (currently mostly 12TB)
- Symlinks on cluster allow CMS data to be accessed via its “Logical File Name” i.e.
  - /store/hidata/HIRun2018/HIDoubleMuon/AOD/PromptReco-v1/000/325/…

How is data moved between sites?

XRootD

- File Transfer Protocol used by CMS
- CMS uses “anywhere, anything, anytime” model, so jobs running at Vanderbilt can access data at other sites, or other sites Vanderbilt.
- Authenticates using grid credentials
- Scales via redirection system
  - An XRootD server can provide requested files
  - …or redirect client to another server to provide files or further redirect

XRootD Servers at ACCRE

- We have two virtualized servers used as dedicated redirectors
  - xrootd1.accre.vanderbilt.edu
  - xrootd2.accre.vanderbilt.edu
  - The address “xrootd.a.v.e” points to both of them
- Additional eight physical servers for serving the actual data
  - se31, se32, se33, … , se37

Data Management

- Rucio sets up central rules for what CMS sites should store what data, manages bulk transfers between sites, ensures data exists at sites
- No on-site ACCRE servers or services are required for Rucio management for CMS data
- We do need to monitor Rucio and ensure our site is functioning

How do we identify CMS users/systems???

Authentication 1

- All CMS tools trust several "Certificate Authorities" (CA)
  - This is called the "trust root" - if the CA signs a message, we consider that message to be true

- User certificates are signed by the CERN CA for every user

```
└─• openssl x509 -in ~/.globus/usercert.pem -noout -issuer -subject
issuer= /DC=ch/DC=cern/CN=CERN Grid Certification Authority
subject= /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=meloam/CN=692113/CN=Andrew Malone Melo
```

- Host certificates are signed by a different CA in our deployment

```
[meloam@sec ~]$ openssl x509 -in /etc/grid-security/hostcert.pem -noout -issuer -subject
issuer= /C=US/O=DigiCert Grid/OU=www.digicert.com/CN=DigiCert Grid Trust CA G2
subject= /DC=com/DC=DigiCert-Grid/C=US/ST=TN/L=Nashville/O=Vanderbilt University/CN=sec.accre.vanderbilt.edu
```

Authentication 2

```
└─• openssl x509 -in ~/.globus/usercert.pem -noout -enddate
notAfter=Oct 14 14:44:20 2021 GMT
```

- We need to perform actions on behalf of a user at multiple places globally (e.g. write an output file) -- but all our software trusts any message signed by a CA
- If the secret leaks, then a rogue person can impersonate me until October 2021!
- Solution: Sign my own message but with a limited expire time

```
└─• grid-proxy-init -valid 24:0 ; grid-proxy-info -all
Your identity: /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=meloam/CN=692113/CN=Andrew Malone Melo
Creating proxy ........................................... Done
Your proxy is valid until: Tue Jan  5 10:34:53 2021
subject  : /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=meloam/CN=692113/CN=Andrew Malone Melo/CN=1623780931
issuer   : /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=meloam/CN=692113/CN=Andrew Malone Melo
identity : /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=meloam/CN=692113/CN=Andrew Malone Melo
type     : RFC 3820 compliant impersonation proxy
strength : 1024 bits
path     : /tmp/x509up_u112870
timeleft : 24:00:00
```

- The software follows the "chain of trust"

```
└─• ./showchain.sh /tmp/x509up_u112870
 0: subject= /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=meloam/CN=692113/CN=Andrew Malone Melo/CN=1623780931
issuer= /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=meloam/CN=692113/CN=Andrew Malone Melo
 1: subject= /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=meloam/CN=692113/CN=Andrew Malone Melo
issuer= /DC=ch/DC=cern/CN=CERN Grid Certification Authority
```

Authentication 3

- Certificates prove identity, but we also need to verify membership -- "Is this user a CMS member?"
  - People join and leave continually
  - Don't want a global SPOF

- Add an additional signature to the users proxy, this time

```
Your identity: /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=meloam/CN=692113/CN=Andrew Malone Melo
Creating temporary proxy ....................................... Done
Contacting  lcg-voms2.cern.ch:15002 [/DC=ch/DC=cern/OU=computers/CN=lcg-voms2.cern.ch] "cms" Done
Creating proxy ................................................. Done

Your proxy is valid until Mon Jan  4 22:52:46 2021
subject   : /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=meloam/CN=692113/CN=Andrew Malone Melo/CN=662591369
issuer    : /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=meloam/CN=692113/CN=Andrew Malone Melo
identity  : /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=meloam/CN=692113/CN=Andrew Malone Melo
type      : RFC compliant proxy
strength  : 1024 bits
path      : /tmp/x509up_u112870
timeleft  : 12:00:00
key usage : Digital Signature, Key Encipherment
=== VO cms extension information ===
VO        : cms
subject   : /DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=meloam/CN=692113/CN=Andrew Malone Melo
issuer    : /DC=ch/DC=cern/OU=computers/CN=lcg-voms2.cern.ch
attribute : /cms/Role=NULL/Capability=NULL
attribute : /cms/uscms/Role=NULL/Capability=NULL
timeleft  : 12:00:00
uri       : lcg-voms2.cern.ch:15002
```