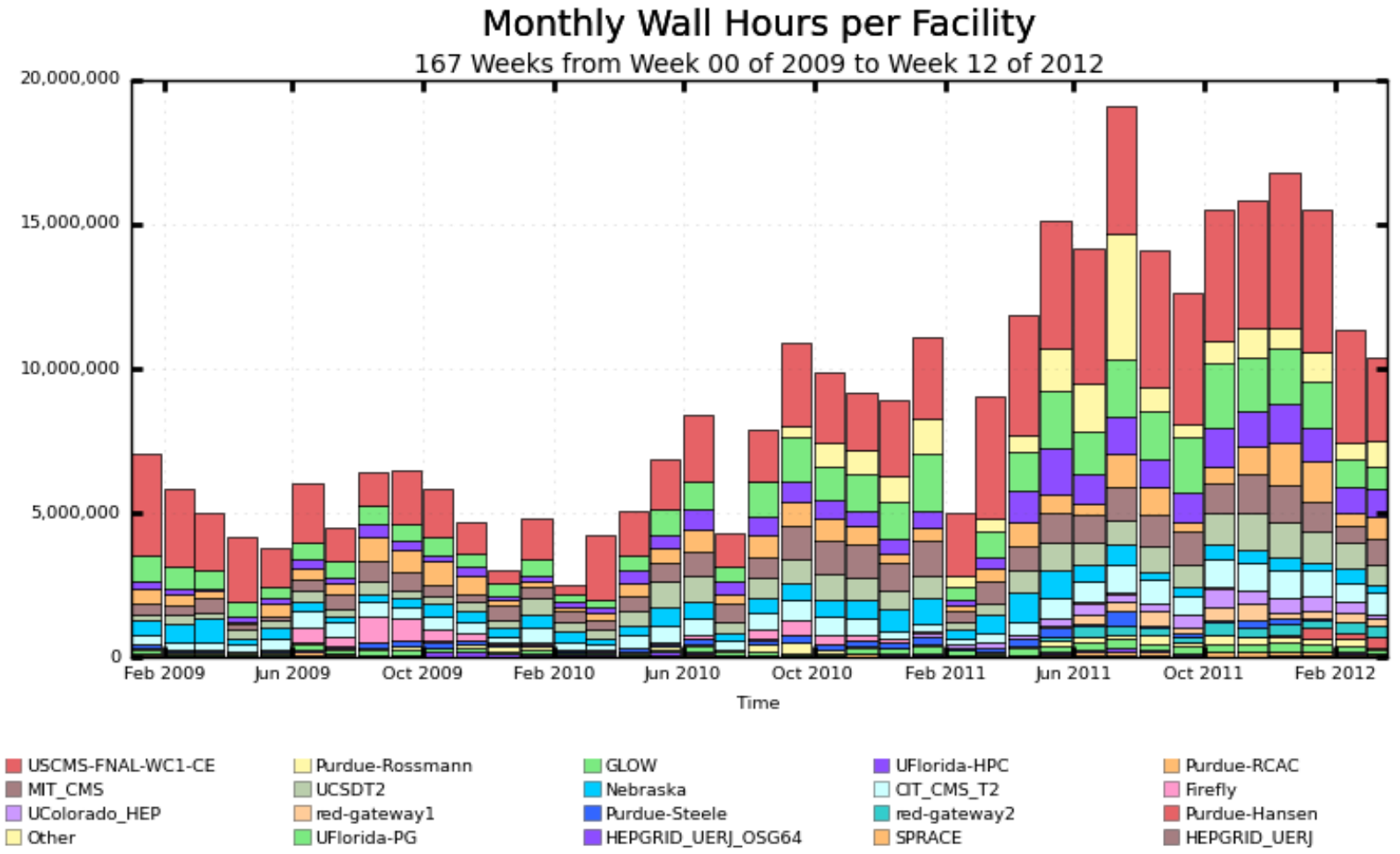
Glideins for CMS on OSG



Jeff Dost (UCSD)

Overview

- Architecture
- Concept of a Global Queue
- Operations

What are glideins?

- GlideinWMS is an implementation of a pilot Workload Management System
- A Pilot is simply a grid job that lands on a worker node and reserves a slot in advance for a user job.
  - When it gets there it calls home to retrieve the user job
- We call pilot jobs **glideins** in GlideinWMS

Why use glideins?

- Allows CMS to have a global queue to implement priorities
- Site failures are not seen by the end user
- Direct grid submission requires overhead.
  - If a pilot is already on a WN and not currently "claimed" when a user submits a job the startup overhead is greatly reduced.
  - Efficiency significantly increases on average if you have a continuous workflow of many jobs on sites for long periods of time (like CMS)

Overview

- **Architecture**
- Concept of a Global Queue
- Operations

Architecture

- **Components of WMS**
- Glidein Internals
- Topologies of Production Systems
- Support Teams

GlideinWMS Components

- User Pool
  - Implementation of global queue
- Glidein Frontend
  - Watch global queue, requests resources
- Glidein Factory
  - Submit glideins in response to resource requests

User Pool

- The user pool looks like any other **Condor** pool
  - Except that instead of on a local cluster, the pool slots are spread out on Sites all over the grid
- It has a condor queue that user jobs join on submission
  - This is what the Frontend checks periodically
- When new glideins start, the slots they reserve join the condor pool
  - **NOTE** This is independent of the underlying batch system the Site runs!

Glidein Frontend

- The Frontend is responsible for checking on waiting user jobs and sending requests to the Factory to submit glideins as needed
- User Pool / Frontend operators monitor user jobs and spot problem users

Glidein Factory

- The factory receives requests from the Frontend and submits glideins to requested Sites using **Condor-G**
- Knowledge about how to submit to various Sites is stored in the Factory configuration
- Factory Operators perform routine maintenance on the Factory as well as monitor glideins to ensure they are running on Sites without error.

Architecture

- Components of WMS
- **Glidein Internals**
- Topologies of Production Systems
- Support Teams

Startup Validation

- Users don't need to worry about Site problems
- Glideins do startup validation. If a WN does not have an adequate environment for a job to run the glidein terminates immediately and reports why.
- User jobs will never land on a node that fails validation
  - “Black hole nodes” do not affect the end user

Validation Examples

- Checks that CMSSW is available
- If gLExec is there, test if it works
- If Squid proxy cache is available glideins will try to use it
- Ensure pilot proxy has long enough lifetime
- Other internal GlideinWMS checks to ensure glidein can run before it starts
- In the future add validation similar to SAM Tests

Notes on gLExec

- If available on the WNs glideins will use it
- Two levels of protection:
  - Protects glidein itself from malicious user
  - Protects users from each other who run on the same glidein
- Additional benefit of running gLExec:
  - Admins can find the real user in the glexec logs

Glidein Lifetime

- Glideins don't reserve slots forever.
- If a glidein is idle with no user jobs to claim it for 20 minutes it terminates.
  - Factory Operators monitor global time wasted
- Otherwise the glidein lives as long as we define it to.
  - We typically set its lifetime to the **MaxWallClockTime** or **MaxCPUTime** (whichever is shorter) from BDII minus a small delta

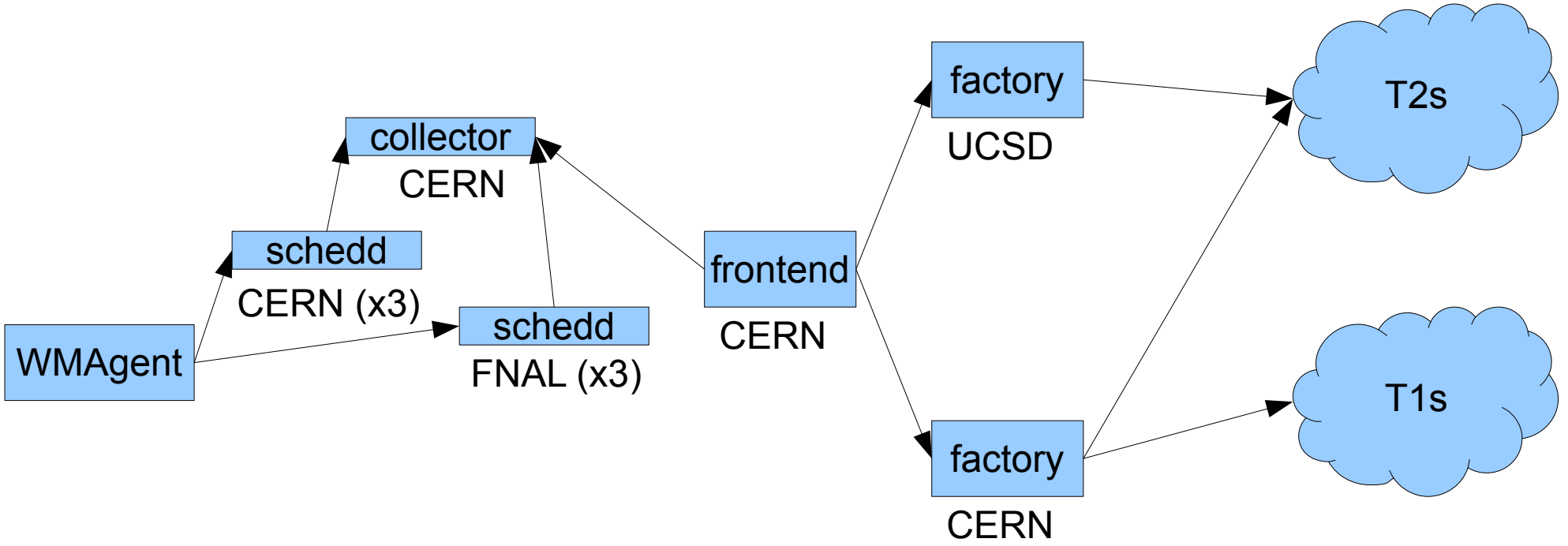
Glideins Protect User Jobs

- User jobs are not tied to the pilots they land on
  - If a pilot fails the user job will just restart on a new pilot somewhere else. It requires no user re-submission

Architecture

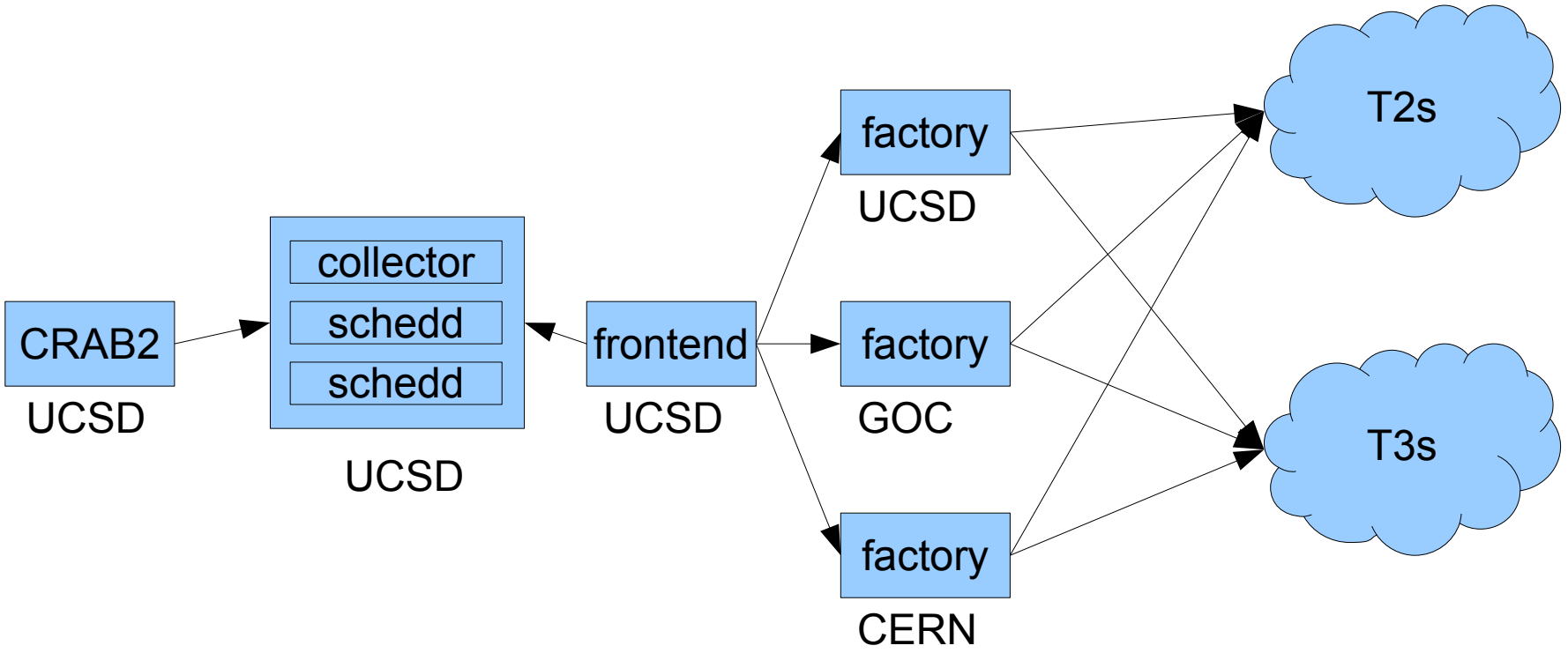
- Components of WMS
- Glidein Internals
- **Topologies of Production Systems**
- Support Teams

CMS Production + MC



Single User Pilots; DN with Role=production
* A T1 only gwms system also exists at FNAL
  - Not relevant to T2/T3; left out of this talk

CMS AnaOps



Multi-User Pilots; DN with Role=pilot

Architecture

- Glidein Internals
- Components of WMS
- Topologies of Production Systems
- **Support Teams**

Support Teams

- Cms-wms-support (funded by CMS)
  - cms-wms-support@physics.ucsd.edu
    - James Letts et. al
  - All complaints about Users go here
- Osg-gfactory-support (funded by OSG)
  - osg-gfactory-support@physics.ucsd.edu
    - Dost, Mortensen et. al
  - All complaints about glideins go here
- T1 Only Support
  - Not relevant to T2s / T3s thus left out of this talk

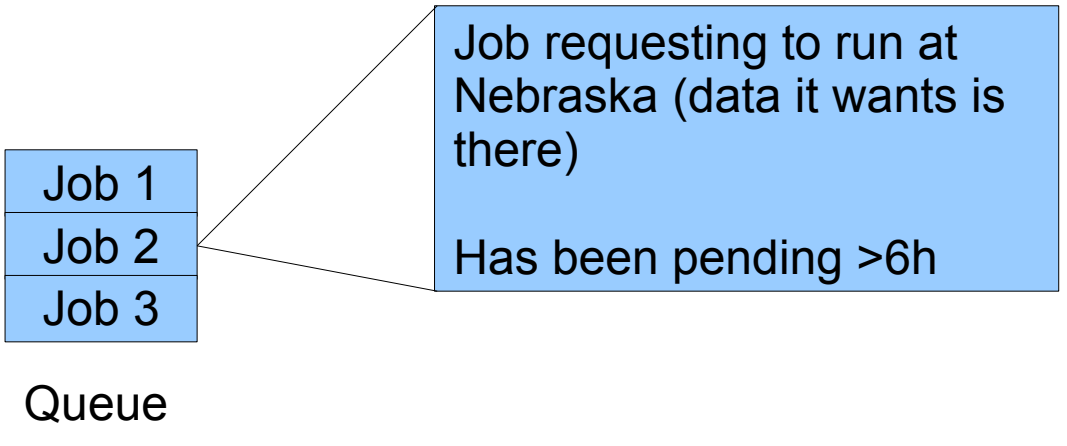
Overview

- Architecture
- **Concept of a Global Queue**
- Operations

Global Queue

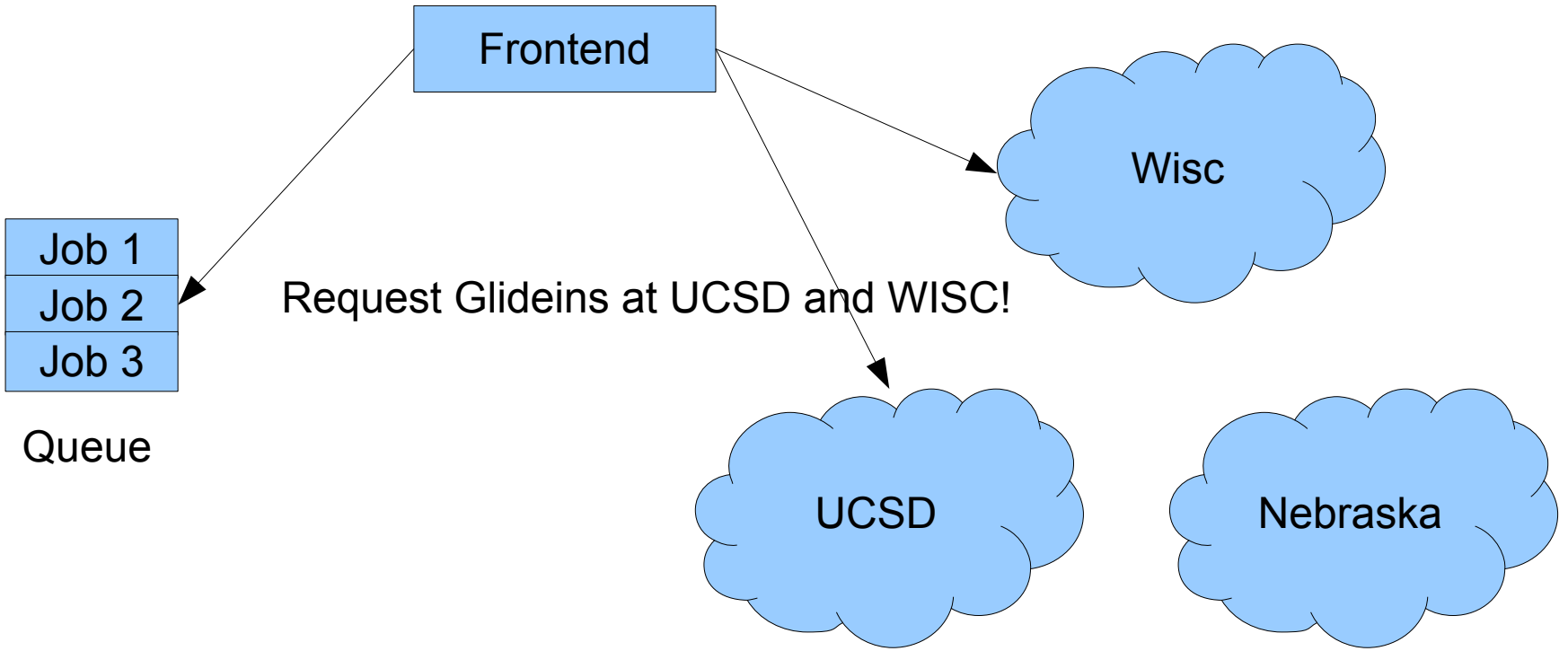
- User priority is no longer controlled at the Site level but Globally in the glideinWMS User Pool
- Exploring ways to make the Global Queue even more Site independent by exploiting Frontend matchmaking
- One such example is the **Overflow** setup

Overflow

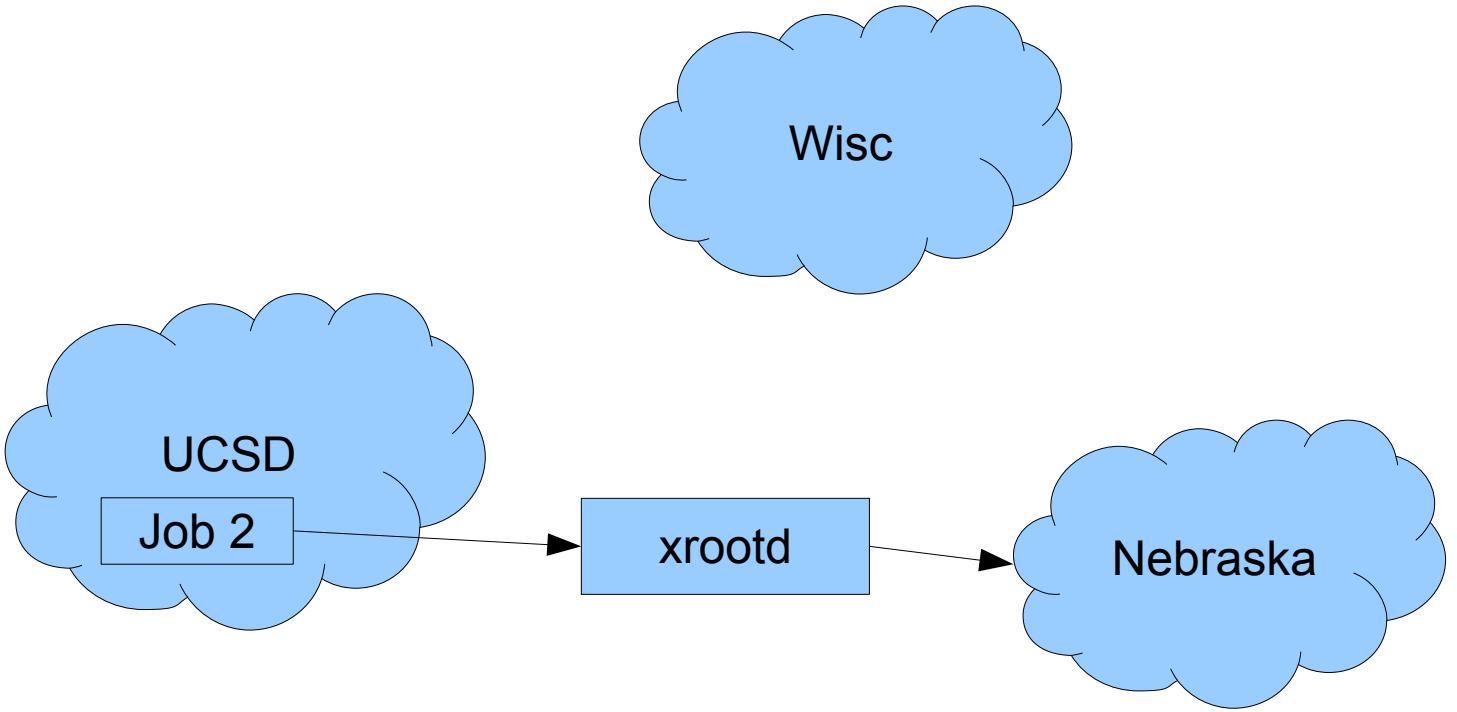


- If Jobs for a site are Pending in Global Queue for more than 6 hours, run the job elsewhere

Overflow


Frontend
Wisc
Job 1
Job 2
Job 3
Request Glideins at UCSD and WISC!
Queue
UCSD
Nebraska

Overflow



Job lands on glidein at UCSD but then uses xrootd to access Nebraska Storage!

Overview

- Architecture
- Concept of a Global Queue
- **Operations**

Role of cms-wms-support

- Control which sites to request to and what should run there
- Identify problematic user jobs
- Investigate held user jobs
- Monitor health of overflow
- Configure Global Queue
  - Configure special matchmaking such as overflow
  - In the future configure CMS overflow to opportunistic sites and even to clouds

Role of osg-gfactory-support

- Report Site issues through GOC and Savannah Ticketing systems
- Work closely with Site Admins to help debug problems
- Temporarily stop and resume submission as needed during site downtimes
- Configure Glidein Factory to submit to new resources
- Update Factory configuration to reflect Site changes (e.g. decommission / replace CEs)

Conclusion

- Glidein System jointly operated between CMS and OSG
  - People power at CERN, FNAL, and UCSD
  - Hardware at GOC, CERN, FNAL, UCSD
- CMS is one of ~12 Communities served by OSG Glidein Factory