OSG As A Partner

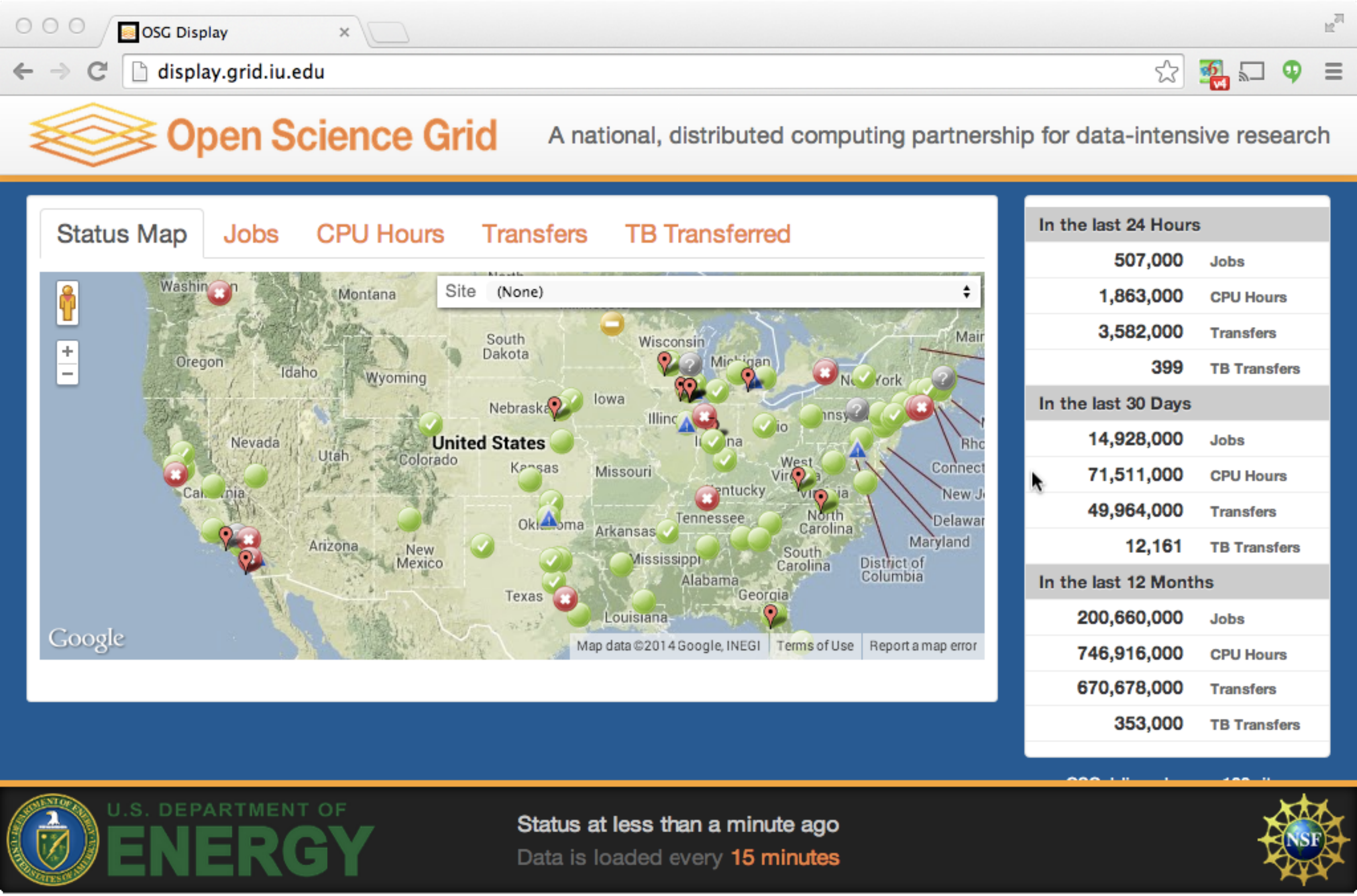
Brian Bockelman
OSG Technology Area Lead

Three Lessons for Today

- What OSG is, what OSG isn’t.
- How OSG partners with the LHC today.
- Basis vectors for an OSG / FIFE partnership.

The OSG - Overview

- “A national, distributed computing partnership for data-intensive research.”
  - The OSG aims to advance the science of **distributed high-throughput-computing**. We work to maximize the throughput of computing resources; FLOPS / Year, not FLOPS / Second.
  - We operate as a **partnership** between an NSF and DOE-funded core and a set of stakeholders trying to advance their domain science through DHTC.
  - We are a community of users, facilities, and organizations organized around DHTC.



OSG Display

display.grid.iu.edu

Open Science Grid

A national, distributed computing partnership for data-intensive research

Status Map | Jobs | CPU Hours | Transfers | TB Transferred

Site (None)

In the last 24 Hours	
507,000	Jobs
1,863,000	CPU Hours
3,582,000	Transfers
399	TB Transfers
In the last 30 Days	
14,928,000	Jobs
71,511,000	CPU Hours
49,964,000	Transfers
12,161	TB Transfers
In the last 12 Months	
200,660,000	Jobs
746,916,000	CPU Hours
670,678,000	Transfers
353,000	TB Transfers

U.S. DEPARTMENT OF ENERGY

Status at less than a minute ago

Data is loaded every 15 minutes

NSF

The OSG Fabric of Services

- The OSG consists of a fabric of services:
  - **Operational**: The *OSG Production Grid*, OSG-Connect services, glideinWMS factory, ticketing, OSG-CA.
  - **Software**: The OSG Software Stack, grid software maintenance.
  - **“Consulting”**: Grid software customization, help planning the use of OSG, porting domain software to the grid.

Resource Acquisition on the OSG

- Unlike the first 5 years of the OSG, we currently utilize a *resource acquisition*-based module.
    - Based on some conditions, a centralized factory acquires a resource on behalf of a VO (submits a job, launches a VM).
    - This resource joins a larger per-VO pool. Typically, this is a HTCondor pool or a PanDA install.
    - The user utilizes the resources in the pool; they see a simplified interface of running resources.

Transitive Trust Model

- OSG has a transitive trust model.
  - Sites trust VOs; VOs manage/trust their users. Transitively, sites establish trust with the users and their jobs.
  - This eliminates the need for a “user-level” trust relationship between sites.
- OSG facilitates this model through mechanisms such as audits of the VO infrastructure and site fire drills to verify they are following security procedures.
  - For example, we need to ensure we can identify who utilized a given resource.

Autonomy of Sites

- A core OSG principle is the *autonomy of sites*.
    - OSG does not own or control any site in the grid.
    - Sites are free to operate as they please, subject only to a minimal number of rules to keep the production grid functional.
    - OSG may advise or recommend, but we try to not require.
- Each site belongs to some organization (such as ATLAS, CMS, Nova, or the local campus computing center) that has its own set of policies the site *must* follow. By keeping the OSG requirements minimal, we can be as inclusive as possible.

The OSG - Opportunistic

- Organizations participating in the OSG Production Grid are encouraged to allow other OSG VOs to use their idle computing resources.

- OSG facilitates this:

  - **Indirectly**: the OSG-run glideinWMS factory can submit pilots to these resources *if* the resource enables the VO.

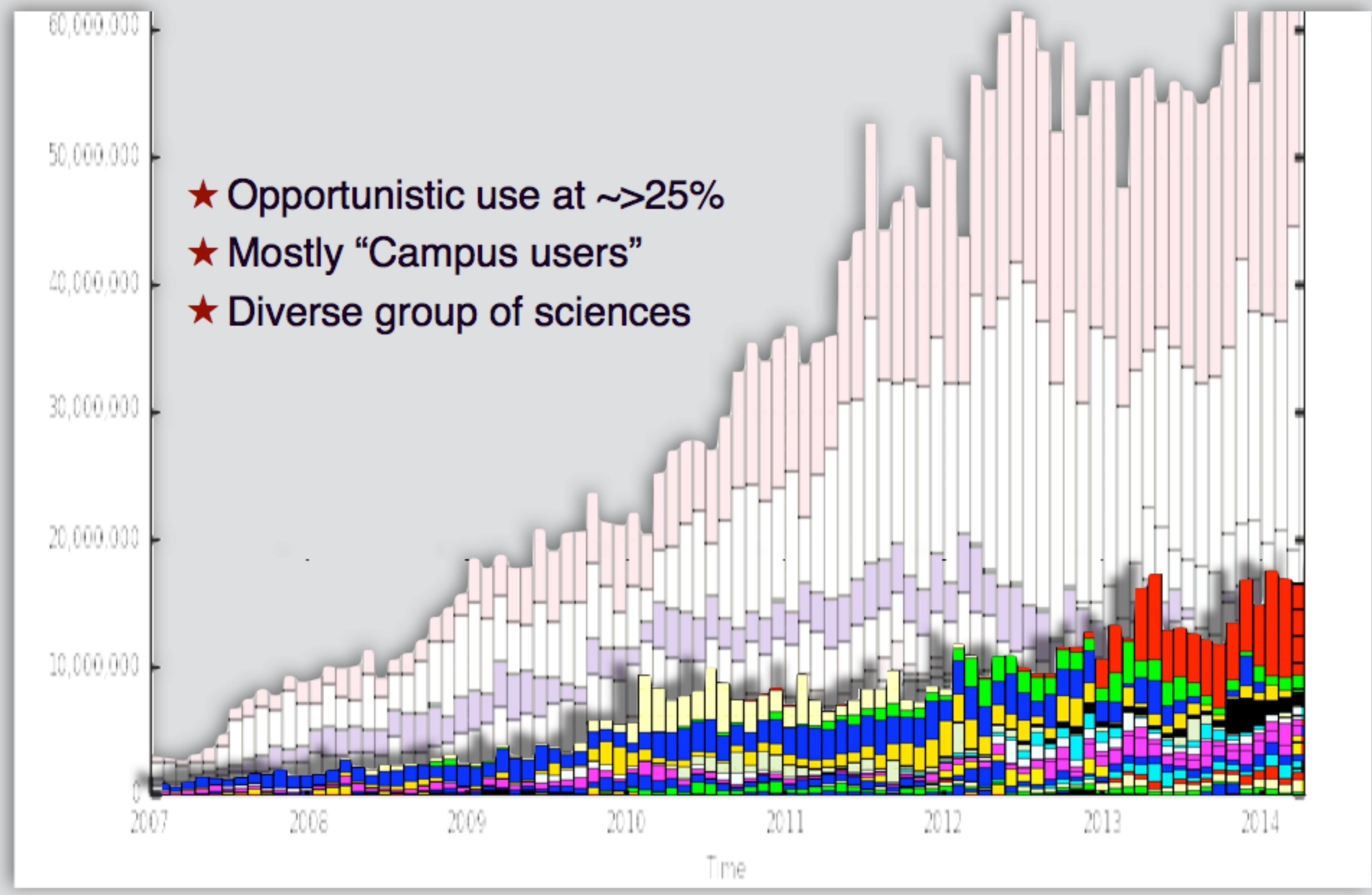
  - **Directly**: the OSG VO is enabled to these resources and can expose them as a HTCondor pool.

  - We encourage newcomers to use the direct method.

- The opportunistic pool averages about 12k cores around the clock.

Open Science Grid

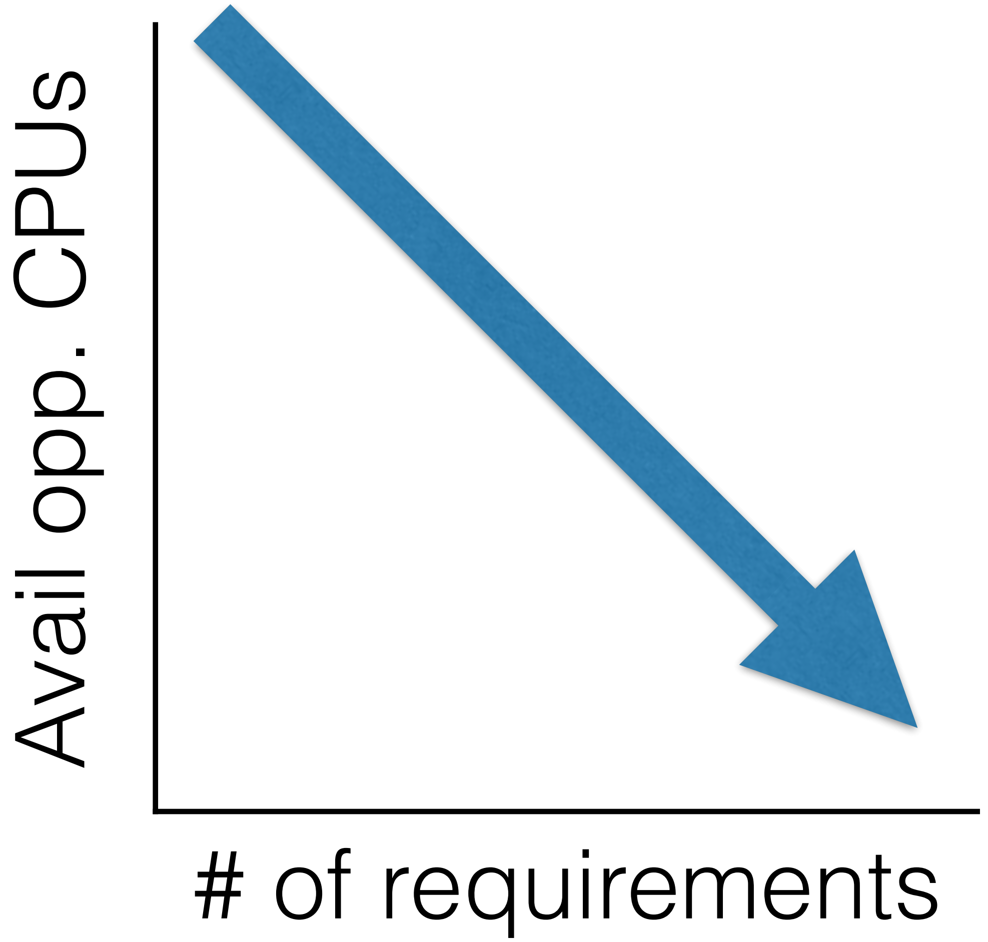
~proportional increase of opportunistic use



- ★ Opportunistic use at ~>25%
- ★ Mostly “Campus users”
- ★ Diverse group of sciences

The Cost of Requirements

- Any new requirement to running jobs limits the sites you can use opportunistically.
    - Sometimes the constraint is severe - decreasing the possible CPUs by a factor-10.
- You want your opportunistic jobs to look like a “normal” OSG job; what is normal evolves over the years.
- **AVOID** needing local site storage, multiple cores / job, more than 1GB / RAM core, worker node software requirements, more than 8 hours of job runtime.
- **DO** utilize CVMFS and the network.
    - Recall that US universities are making



Assume Opportunistic from Day One

If you can do opportunistic computing well, you can do computing on resources you own.

If you can do opportunistic computing, your owned resources will be more reliable. Opportunistic computing keeps your VO services simple and reliable.

OSG and the LHC

- The OSG runs a number of services for the LHC community. Highlights include:
    - Participation in WLCG activities such as accounting, security, monitoring, and information services.
    - Provides user and host certificates through a CA service.
    - Runs the pilot factories for CMS.
    - Software distribution through a CVMFS Stratum-1.
    - Software customization on an as-needed basis.
    - Keeping the production grid production-worthy.
- We try to capture *common requirements* across the USLHC organizations.

OSG and FIFE

- OSG is ready to listen and understand how we can better collaborate with FIFE.
- As I’m the OSG Technology Area coordinator, I’ll highlight two technologies which will be mutually beneficial.
- Do not overestimate technology! There can be more value in:
    - Attending the production meeting.
    - Leveraging the OSG Ticket service.
    - Utilizing the OSG glideinWMS service. Running and maintaining one at the OSG-scale costs at least 2 FTE / year.

Software Customization

- The available effort is very limited, but OSG can work to customize software for stakeholders.

- By default, we prefer to facilitate the work done in upstream.

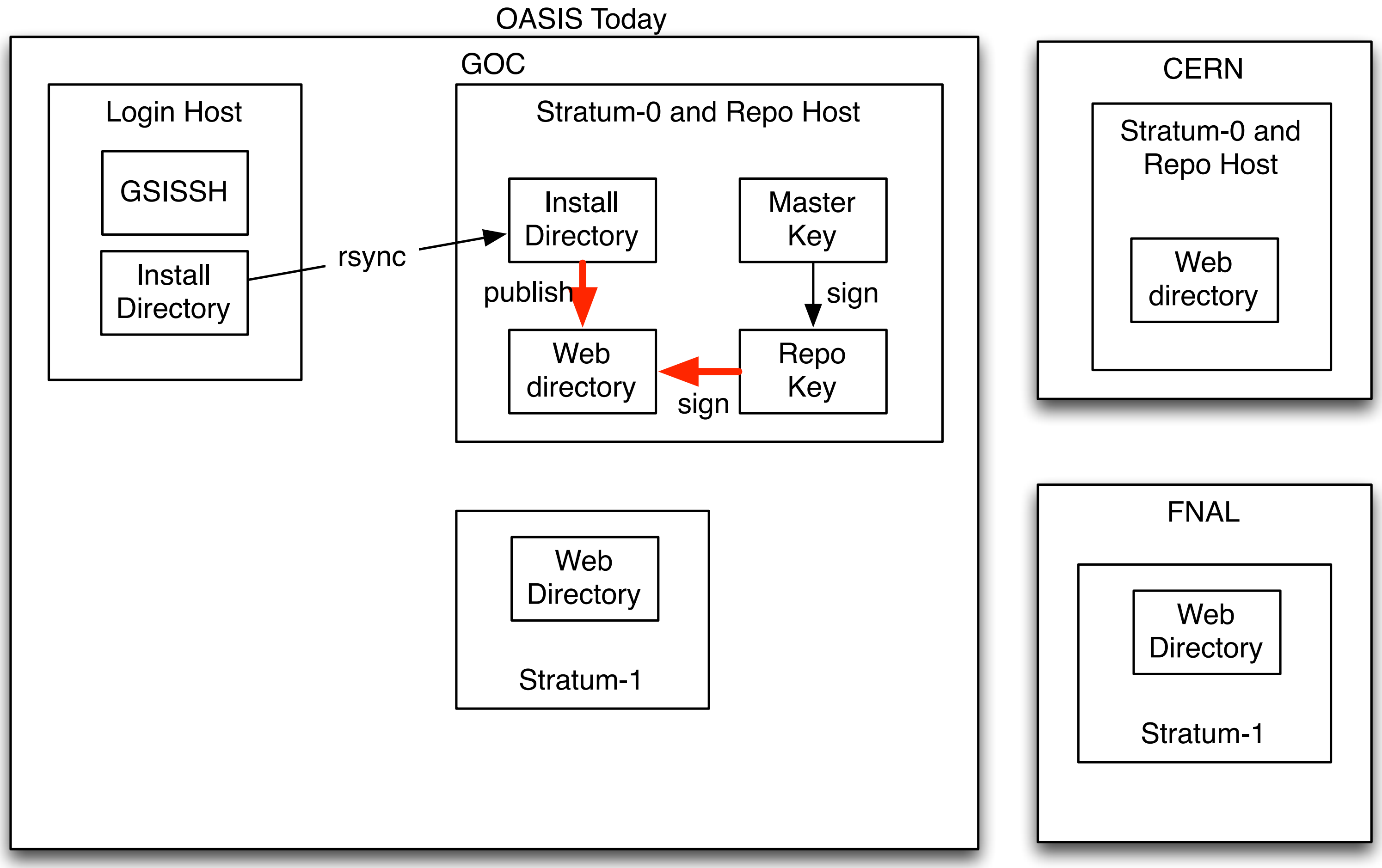
- If effort is unavailable, we contribute patches to upstream.

- If upstream is defunct, we will contribute and carry patches ourselves.

- **Example**: recently added some new features to GUMS specifically to support the FIFE use case on Fermigrid's dCache instance.

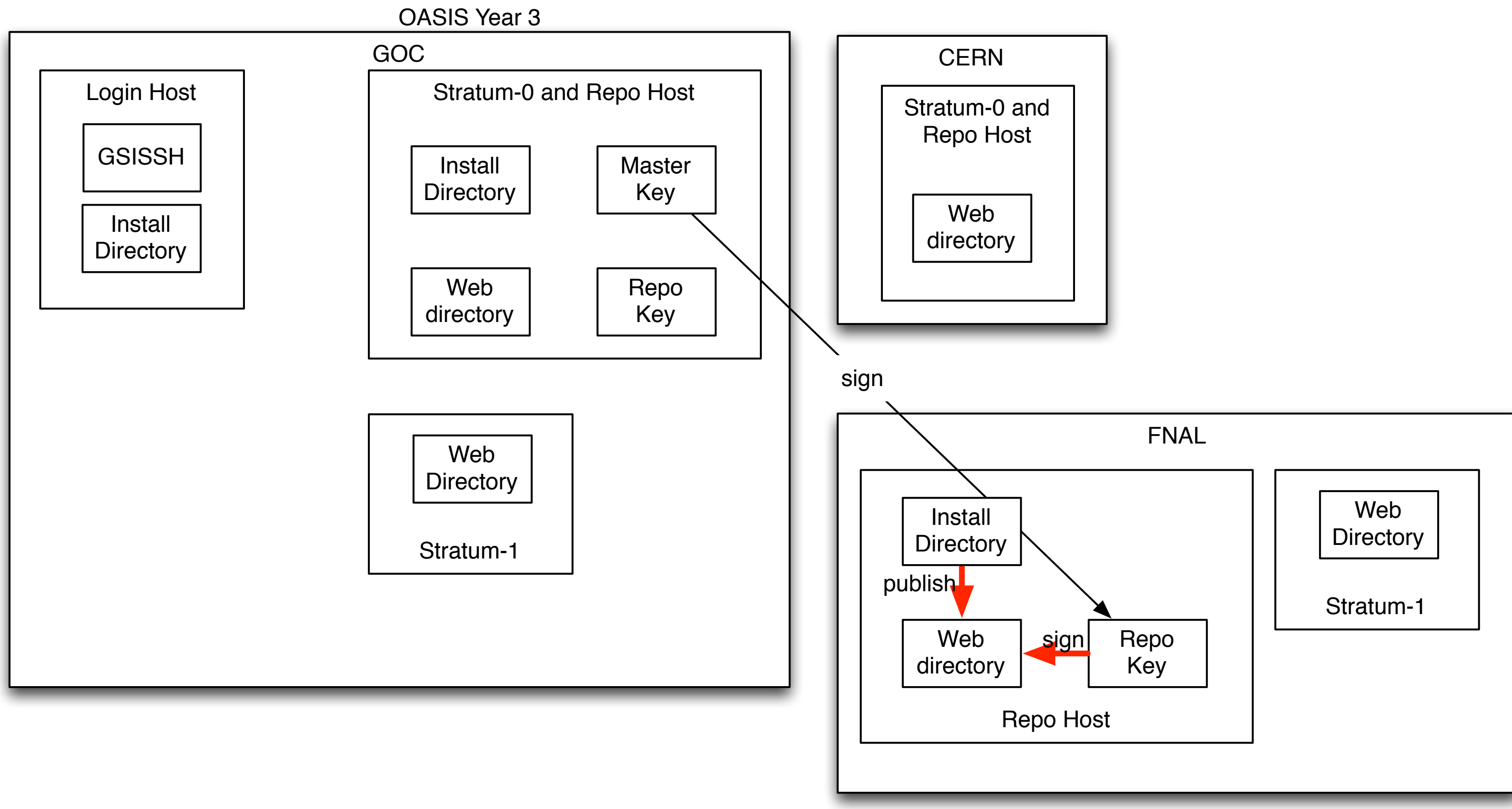
OASIS

- The OASIS service originally provided a all-in-one hosted CVMFS server for smaller VOs to distribute software.
- Shared login server, Stratum 0 / repo, and Stratum 1 infrastructure.



OASIS

- FIFE is quickly outgrowing the shared components.
- Working to deploy *external repos* - VO runs the repo host and manages software installs, but OSG still signs and runs remaining infrastructure.
- Still limited to what is possible with CVMFS.



Homework / Conclusions

- FIFE should be able to leverage DHTC methodology, the OSG Production Grid, and the OSG Opportunistic Facility to grow their computing scale.

- Distributed computing can grow expensive in terms of personnel; OSG and FIFE should identify and capitalize on overlapping needs.

- OSG and LHC “grew up together”; what’s the best mechanism to onboard FIFE?

Bonus Slide #1

- In addition to working with OSG, I've been contributing to CMS Computing for about a decade. Some thoughts and observations about running computing for HEP follow.

- Make sure your physics software can access files without accessing databases or web services.

- Either control or contribute to the physics software your users utilize; it'll make for a more nimble organization. The fact that CMS users use CMSSW and not ROOT on the grid allowed us to shave years off the deployment time of remote IO.

- NEVER rewrite working software.

- NEVER add requirements to the worker node for running sites. ALWAYS make sure you can run on resources you do not own!

- Developing your own workflow management system costs at least a FTE-decade. A mature one costs several FTE-decades.

- User interfaces cost several FTE-years to get ready for users.

Bonus Slide #2

- When possible, “make it a HTCondor problem”.
- Never let a user see a grid certificate.
- LHC-style data management - multiple custodial copies, moving files explicitly between sites - is extremely difficult and expensive to develop / operate.
    - There is no such thing as opportunistic storage.
- The network is 1-2 orders of magnitude more reliable than disk services.
    - If you don’t pay the people running the site, local storage will be unreliable but remote IO will work.
- Run full-scale computing exercises every year or two when you don’t have new data.