HTCondor on OSG

Brian Bockelman

HTCondor Week Europe 2017

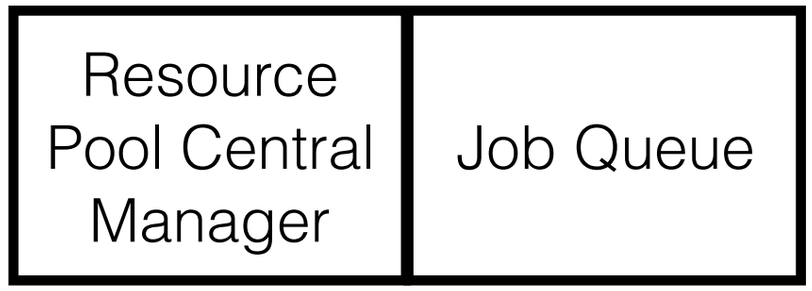
The Open Science Grid

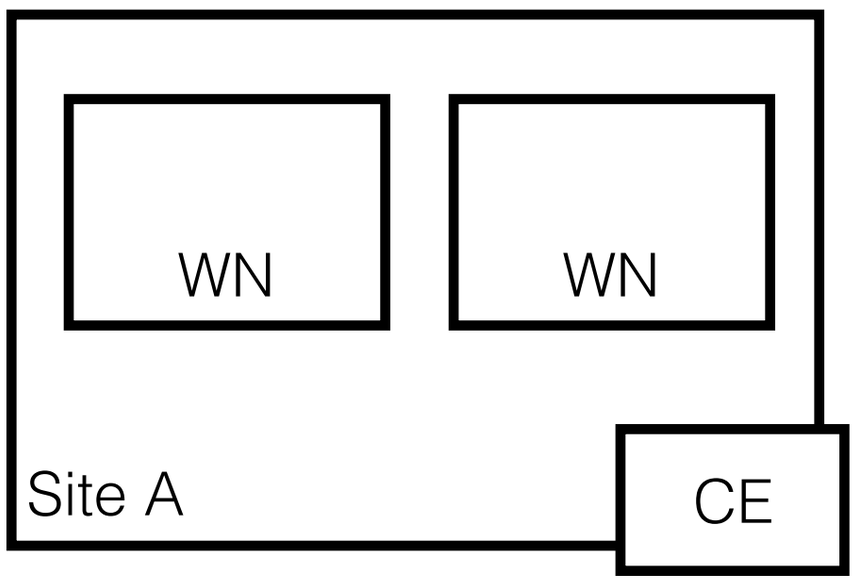
- The Open Science Grid (OSG) is a national, distributed computing partnership for data-intensive research.
- The OSG partnership includes a funded central project, participating sites, and experiments/VOs.
- The OSG provides a fabric of services (operations, software, knowledge base) that enables distributed High Throughput Computing (DHTC).
    - At the center of many of these services is HTCondor!

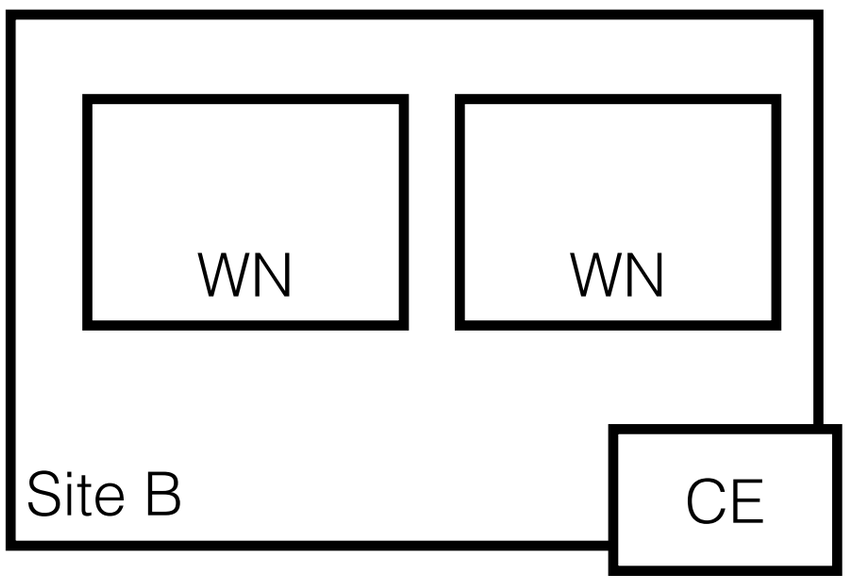
In the last 24 Hours	
302,000	Jobs
4,917,000	CPU Hours
5,917,000	Transfers
562	TB Transfers
In the last 30 Days	
9,631,000	Jobs
120,793,000	CPU Hours
194,037,000	Transfers
13,621	TB Transfers
In the last 12 Months	
132,408,000	Jobs
1,449,758,000	CPU Hours
1,992,412,000	Transfers
169,000	TB Transfers

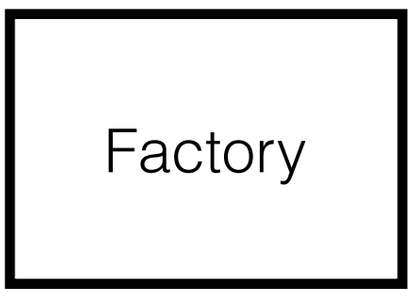
OSG Job Architecture

VO-specific services



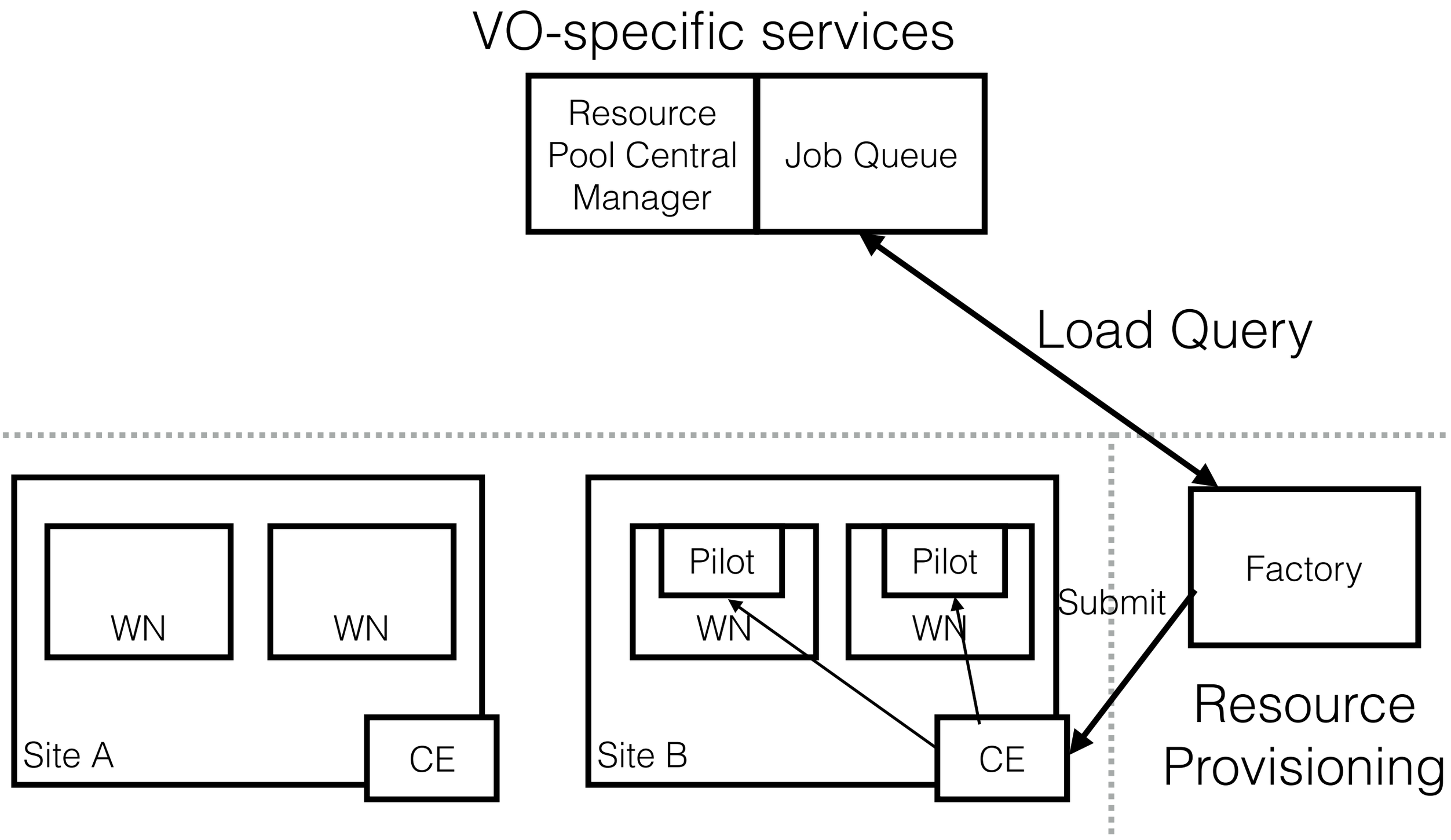




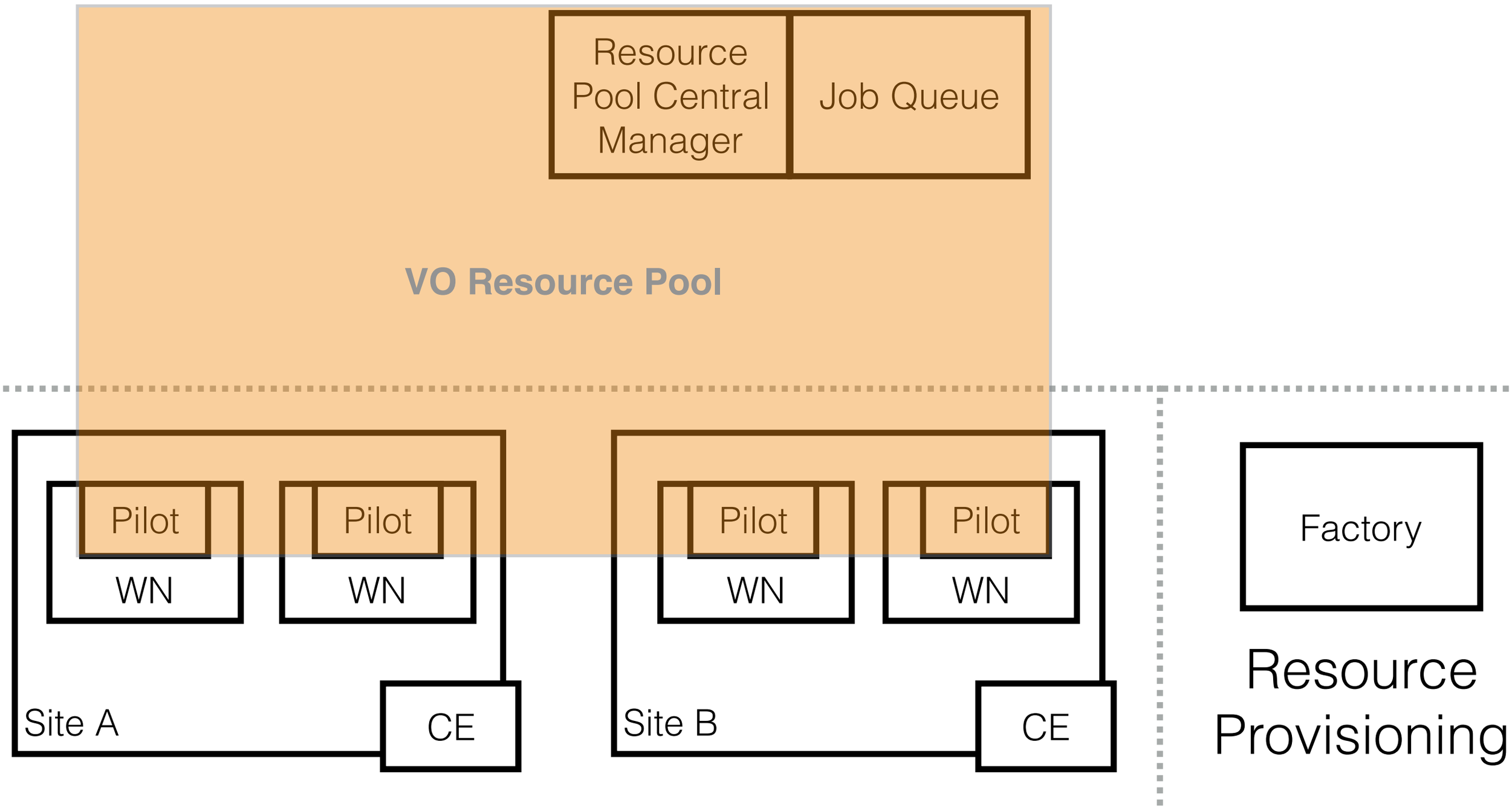


Resource
Provisioning

OSG Job Architecture



OSG Job Architecture



The Swiss Army Knife

- It's possible to use HTCondor for all "boxes" in the previous figure: many OSG VOs do!
- For OSG, HTCondor is often:
    - The Swiss Army knife used to meet a variety of challenges, or
    - The hammer used to hit all our problems.
- Why?
    - **Overlapping problem domains**. Both HTCondor and OSG focus on high thoughput computing (HTC) at their core; OSG emphasizes more on the distributed aspects (DHTC).
    - **Solid architecture**. HTCondor has its roots in cycle scavenging across the campus; similar challenges in terms of reliability between cycle scavenging and global distributed computing.
    - Leverage a huge pool of **in-house knowledge**.
    - Active, engaged **development team**.

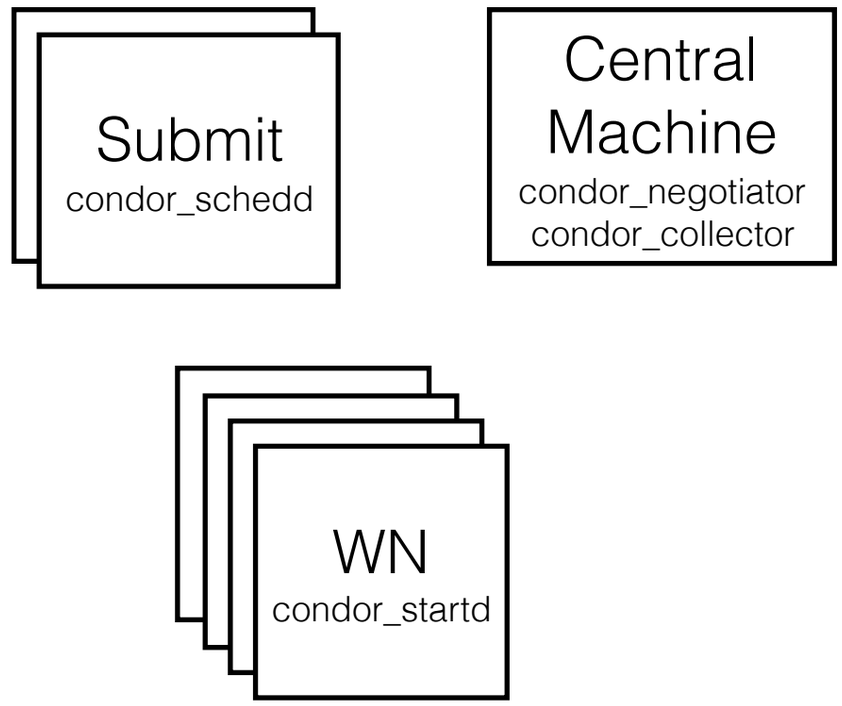
Goal for today:

Overview of the many ways HTCondor is utilized by OSG.

This is the “short version” of Matyas Selmeci’s in-depth presentation:
https://research.cs.wisc.edu/htcondor/HTCondorWeek2015/presentations/SelmeciM_UnorthodoxUses.pdf

HTCondor, the Batch System

- HTCondor is a popular batch system solution on OSG sites.
- Likely how everyone here uses HTCondor!
- Around 50% of OSG resources are inside HTCondor.

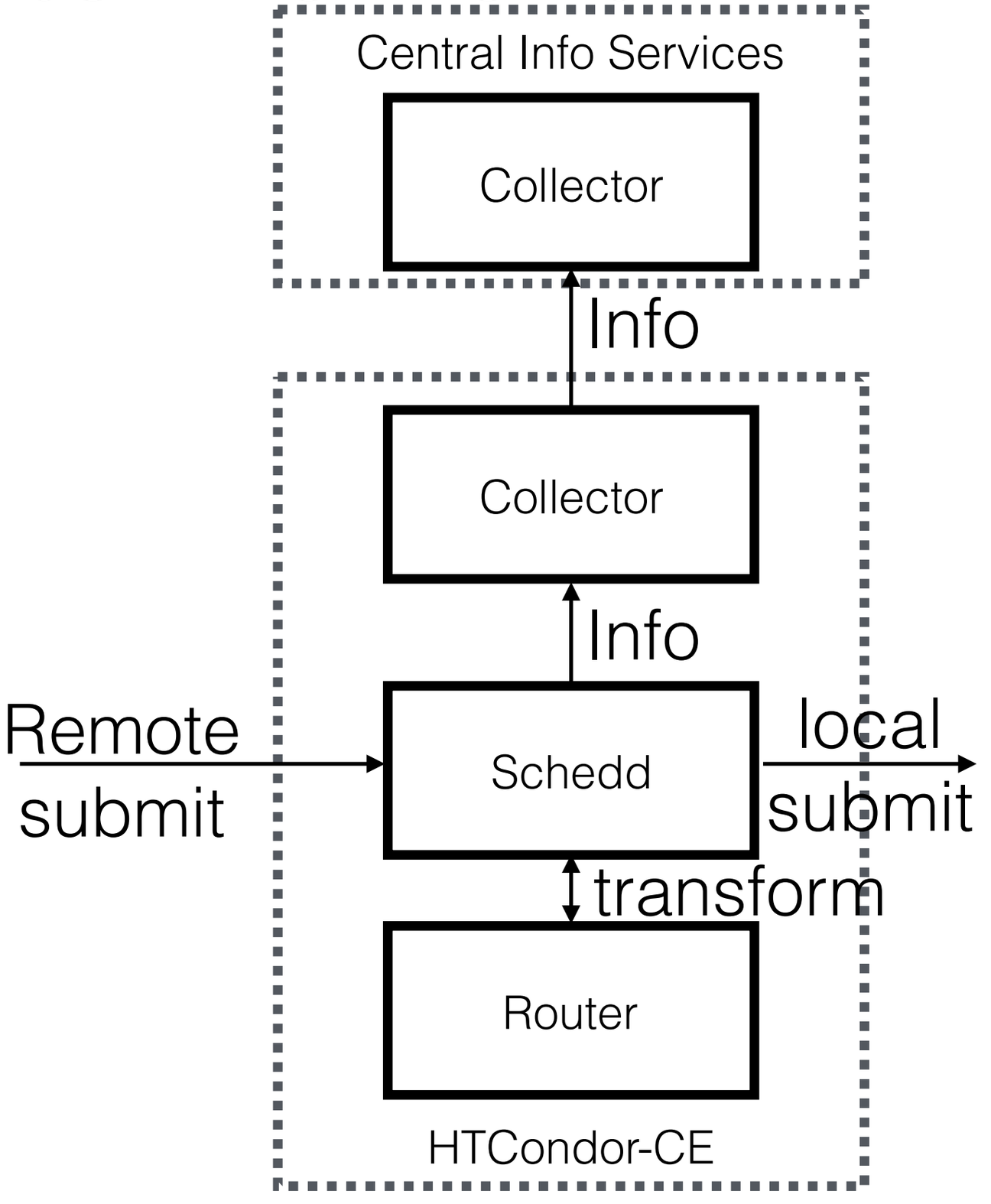


HTCondor, the Compute Element

- In this configuration, the `condor_schedd` is configured to accept remote submissions.

- Allows for a complex set of site-provided transforms before the job is then submitted to the local batch system.

- See later talk!



HTCondor, the Monitoring System

- OSG provides the “Resource and Service Validator” (RSV), a software package for simple Nagios-like monitoring of grid resources.

- Goal is to submit simple grid jobs and wait for them to finish: “simple” probes can take hours. Not a great fit for Nagios itself!

- Periodic task? Great, use cron! … Maybe not?

    - Difficult to run just one instance of a long-lived cron job.

    - Difficult to stagger jobs at random intervals.

    - cron does not do any process/resource management.

    - Jobs may be missed if machine is not running.

HTCondor, the Monitoring System

- OSG built a small `condor-cron` package that utilizes the “cron-like” scheduling available in HTCondor.
  - Like the CE, `condor-cron` can be installed alongside a “normal” HTCondor install.
- RSV is a set of tools that generate & manage `condor-cron` jobs:
  - Submit tests to condor (which run on the remote CE).
  - Regenerate the webpage.
  - Upload test results centrally.

```
OnExitRemove = false

# 7,27,47 * * * *
CronMinute = 7,27,47
CronHour = *
CronDayOfMonth = *
CronMonth = *
CronDayOfWeek = *

CronWindow = 99999999

Executable = ping-host
Arguments = ce.example.com
Queue
```

Information Services

- We tend to think of `condor_collector` as simply holding machine status - default output of `condor_status`. However, it also contains:
  - Submitter and fairshare information.
  - Performance statistics of the various daemons.
  - DNS-like location of each daemon.
- Basically, the collector can be used a generic message board!

Information Service

- For the schedd ad, HTCondor-CE injects information about:
  - Allowed VOs
  - Available resources.
  - How to allocate resources.
  - CE information (site name)
- All ClassAd and matchmaking based!
- The schedd ad is forwarded to a central collector. There, a process serves the information in several formats.
  - In 2017, we added an AGIS-specific JSON.

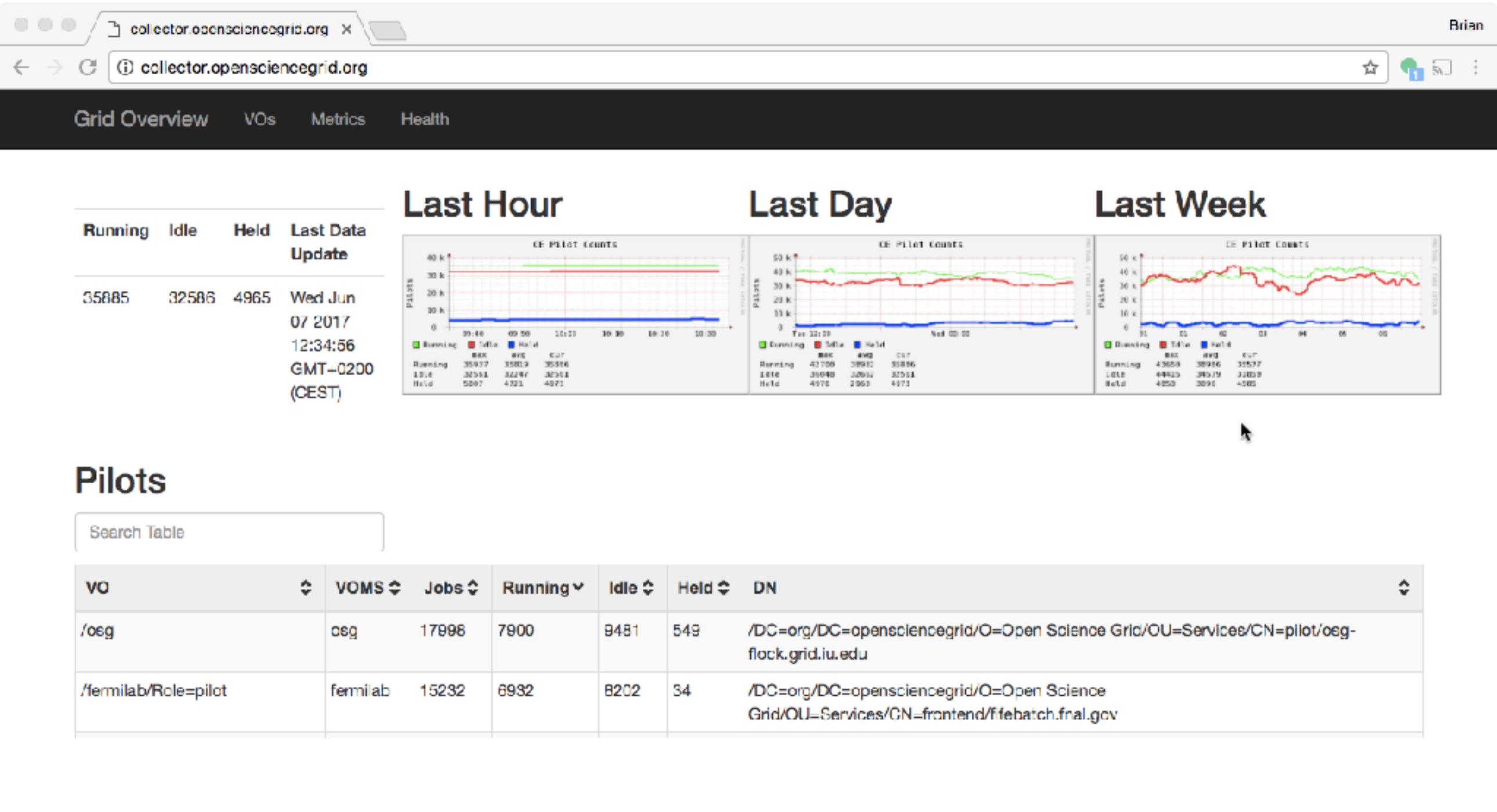
bbockelm — bbockelm@hcc-briantest7:~ — ssh hcc-briantest7.u...

```
[bbockelm@hcc-briantest7 ~]$ condor_ce_status -schedd -pool collector.openscienc
egrid.org
Name                 Machine              RunningJobs   IdleJobs   HeldJobs

CE01.CMSAF.MIT.EDU   CE01.CMSAF.MIT.EDU           264        799         14
CE02.CMSAF.MIT.EDU   CE02.CMSAF.MIT.EDU           126         16          0
CE03.CMSAF.MIT.EDU   CE03.CMSAF.MIT.EDU           147         29          0
atlas-ce.bu.edu      atlas-ce.bu.edu              654        380          0
bonner06.rice.edu    bonner06.rice.edu             11          2          4
ce.grid.unesp.br     ce.grid.unesp.br               0          0          0
ce01.brazos.tamu.edu ce01.brazos.tamu.edu           2          2        354
ce1.accre.vanderbilt.e ce1.accre.vanderbilt.e     586        152        321
cecc7test.hep.wisc.edu cecc7test.hep.wisc.edu       0          0          0
cit-gatekeeper.ultrali cit-gatekeeper.ultrali     632        219          7
```

bbockelm — bbockelm@hcc-briantest7:~ — ssh hcc-briantest7.unl.edu —

```
[bbockelm@hcc-briantest7 ~]$ condor_ce_info_status
Name                    CPUs   Memory  MaxWallTime AllowedVOs

T2_US_Nebraska Dell SC     4     8053         1440 cms, belle, cigi, des, fermilab, g
GridUNESP                  8    16384          720 cdf, gridunesp, mis, star, cms, dz
OrangeGrid                 1     4000         1440
gpgrid.fnal.gov            8    12010         1440 osg, minos, nova, cdf, minerva, ma
OU_OSCER_ATLAS_2650       20    32768         1440 atlas, dosar
OU_OSCER_ATLAS_2670       24    65536         1440 atlas, dosar
GridUNESP_CENTRAL          8    16384          720 gridunesp, cdf, fermilab, accelera
IceCube                   32    64000        43200 glow, icecube, osg
gpgrid.fnal.gov            8    12010         1440 osg, minos, nova, cdf, minerva, ma
GLOW CE                    8    16030         1440
GLOW g12                   8    16384         1440
GLOW g14                   8    16384         1440
GLOW g18                  16    48305         1440
GLOW g19                  24    48305         1440
GLOW g20                  16    32049         1440
GLOW g22                  24    64335         1440
GLOW g23                  24    64335         1440
GLOW g24                  24    64335         1440
GLOW g26                  32    96000         1440
GLOW g27                  40   128000         1440
GLOW g28                  40   128000         1440
GLOW g29                  40   128000         1440
GLOW g30                  40   128000         1440
```

Information Service



HTCondor, the Pilot Factory

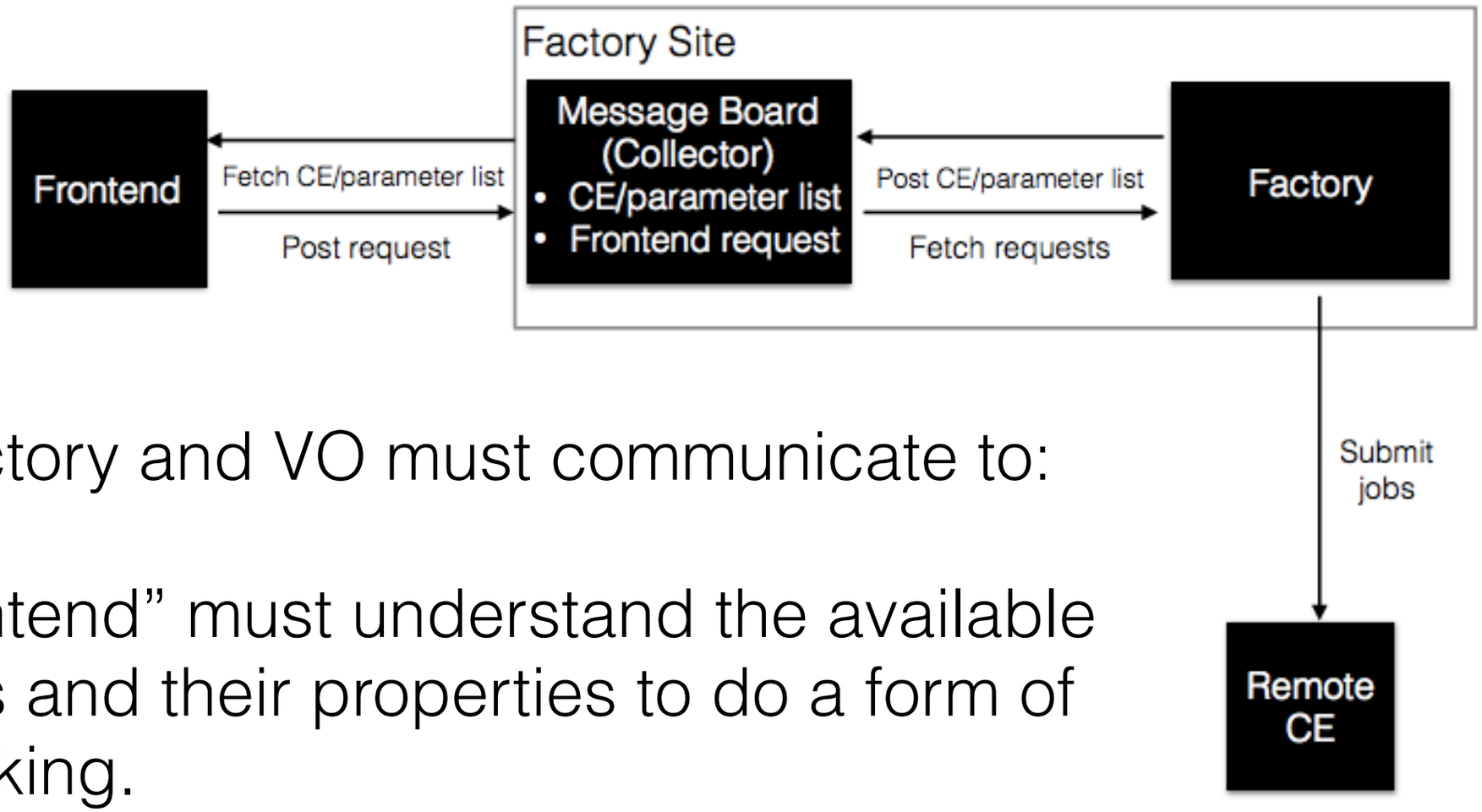
- HTCondor's "grid universe" has the ability to delegate the execution of jobs — pilot job — to a remote CE endpoint.

- The *factory* is in charge of determining the number of pilots to run, and where.

- The pilots are submitted to HTCondor, which manages the actual remote submission, job management, file management, etc.

- OSG's factory runs a single "glideinWMS" factory, which manages all the pilot submissions for multiple VOs.

HTCondor, the Information System



- The pilot factory and VO must communicate to:
  - VO's "frontend" must understand the available resources and their properties to do a form of matchmaking.
  - The frontend must tell the factory how many pilots to request and which types.

HTCondor, the Global Pool

- HTCondor is used to discover sites, submit pilots, and launch pilots. The pilots start … a HTCondor worker node!

- All the resources a VO can access are turned into a single, global HTCondor pool.
    - These range from “modest” (20k cores) to “massive” (200k cores).
    - See the presentation from CMS later in this session!
- To users, the global pool is simply a HTCondor pool. For *non*-data-intensive, everything looks homogeneous.
    - For data-intensive workflows, the site-to-site variations begin to leak through.

The Future

- It's possible that HTCondor has maximum saturation within the OSG!
  - Not looking at new use cases (currently!), but improving existing ones.
- **Information service**: we want to better automate collection of data and make the information more descriptive.
- **CE**: Improve distribution channels, further decouple from OSG (better integration with APEL accounting).
- **RSV monitoring**: Is "yet another" service sites have to run/ maintain. Looking at breaking this apart and embedding it in other services.