Federating Clouds for High Energy Physics

Ian Gable
University of Victoria

Andre Charbonneau, Martin Conklin, Ronald Desmarais, Colson Driemel,
Colin Leavett-Brown, Randall Sobie, Michael Paterson, Ryan Taylor

with significant assistance and support from the
ATLAS and Belle II Collaborations, and CERN IT

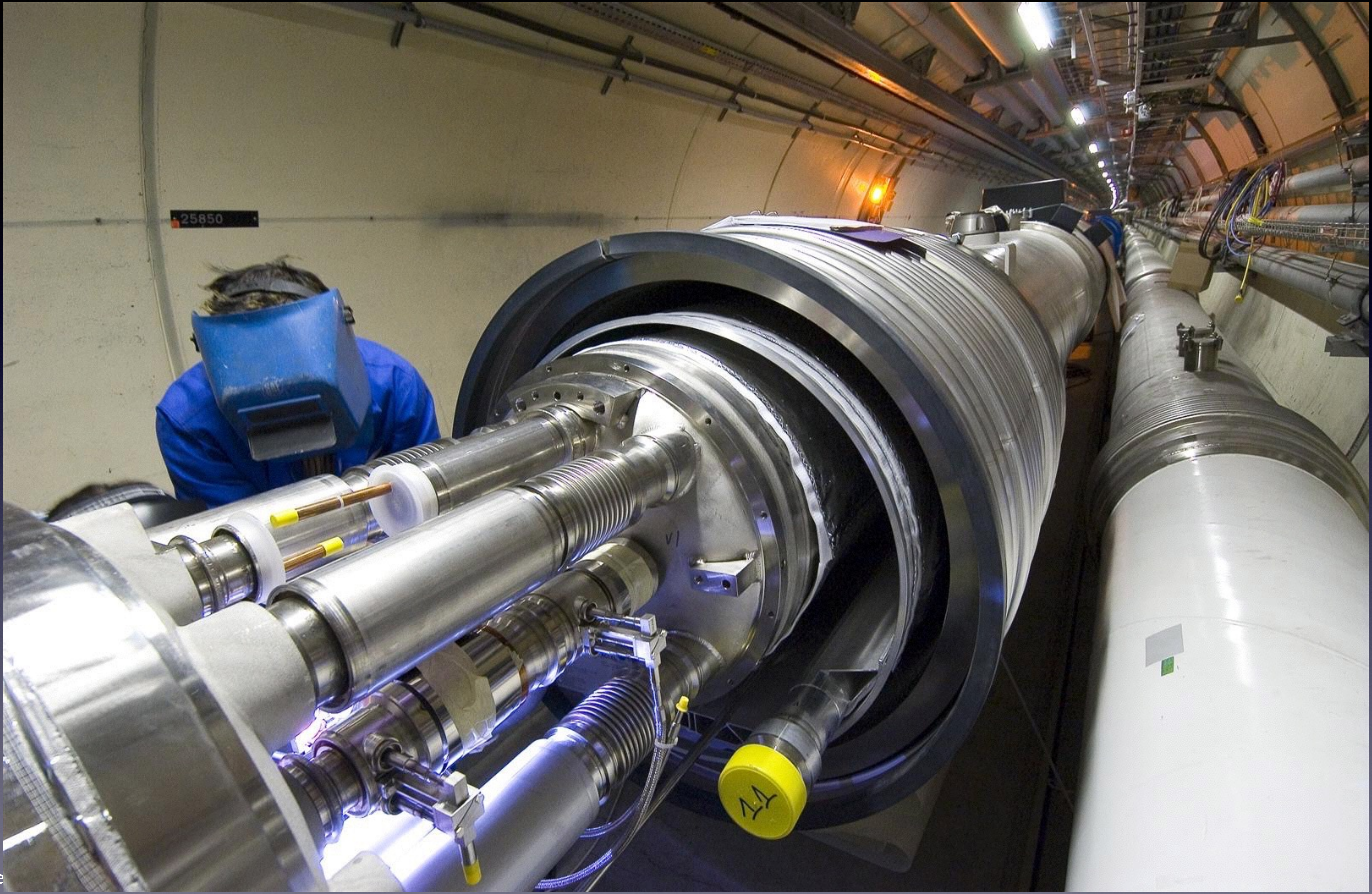
OpenStack Summit , May 18-22, 2015

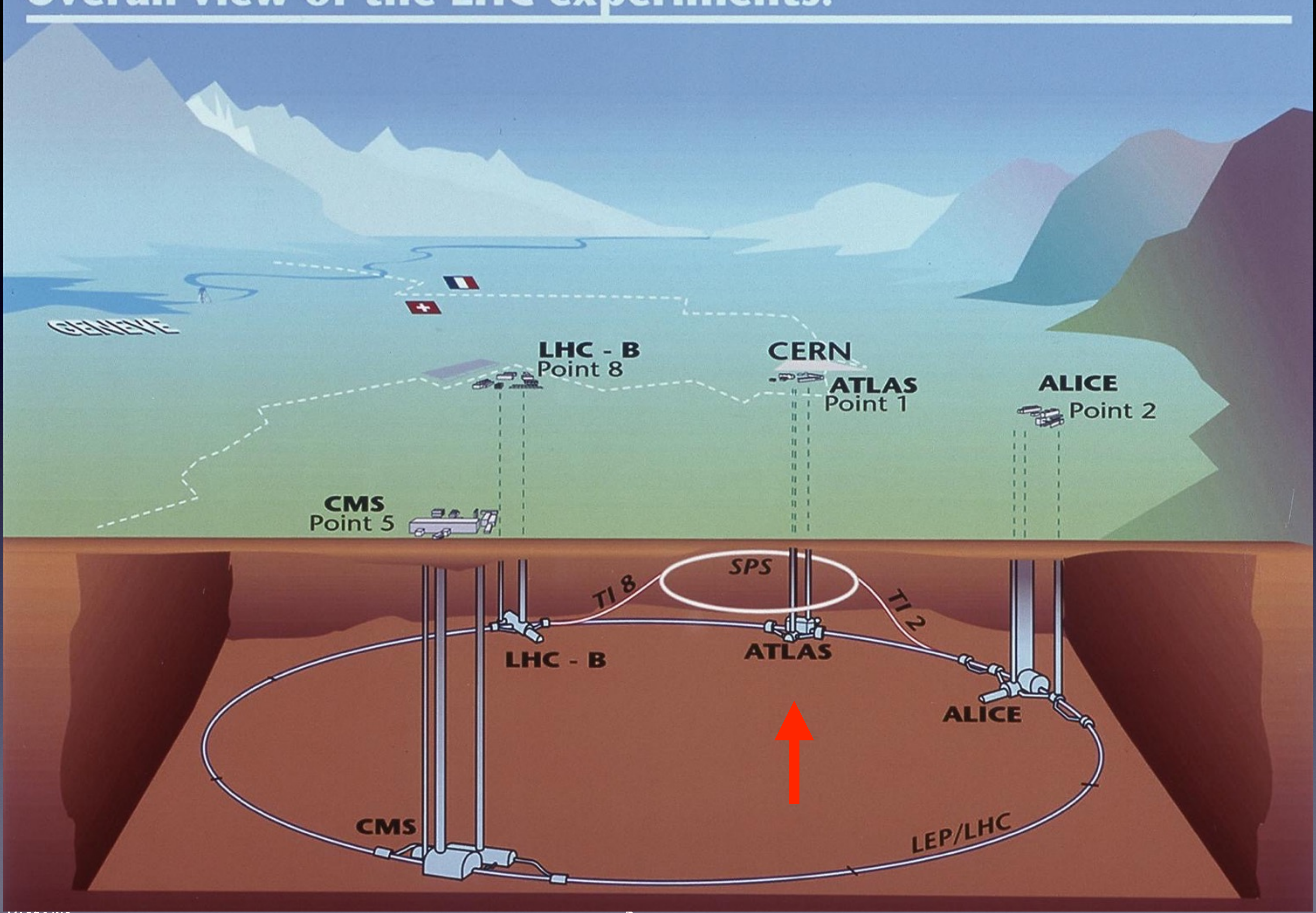
Outline

- What is experimental High Energy Physics?
- What our computing workloads look like?
- Components of our Distributed Cloud
  - Cloud Scheduler: Batch Job Management
  - Glint: VM image distribution
  - Shoal: Squid cache discovery
- Some results

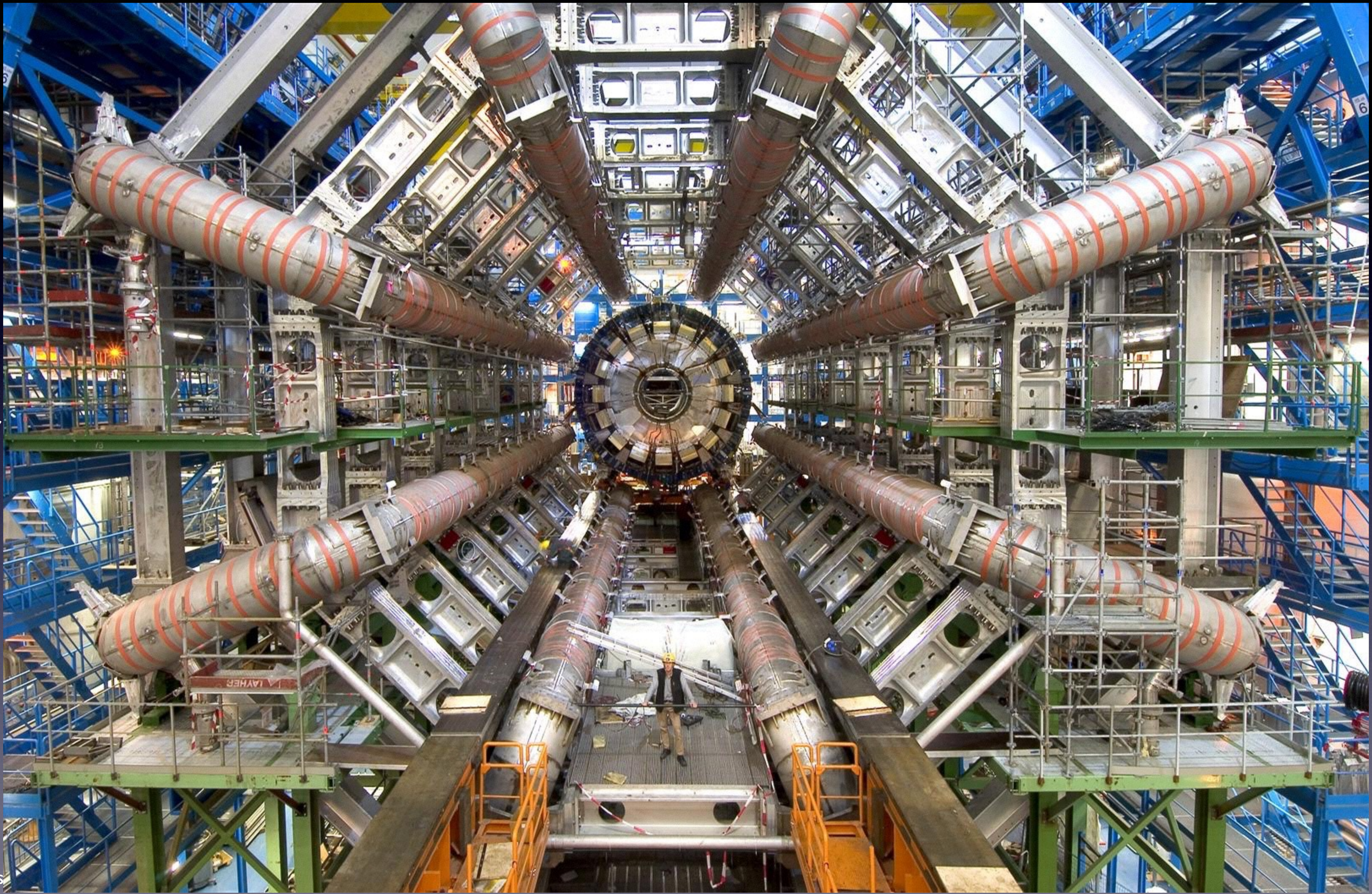
Large Hadron Collider

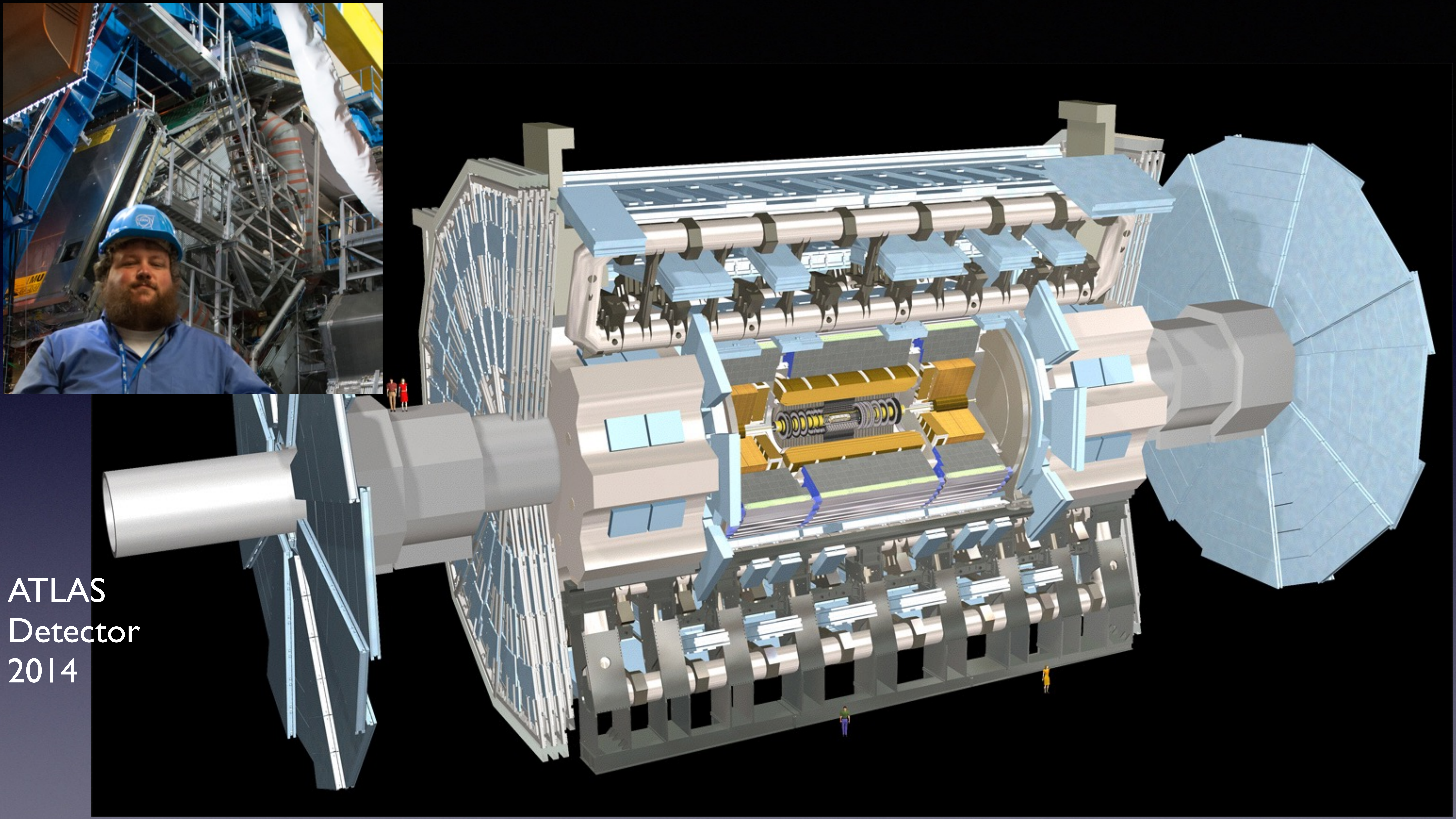
27 km ring


25850

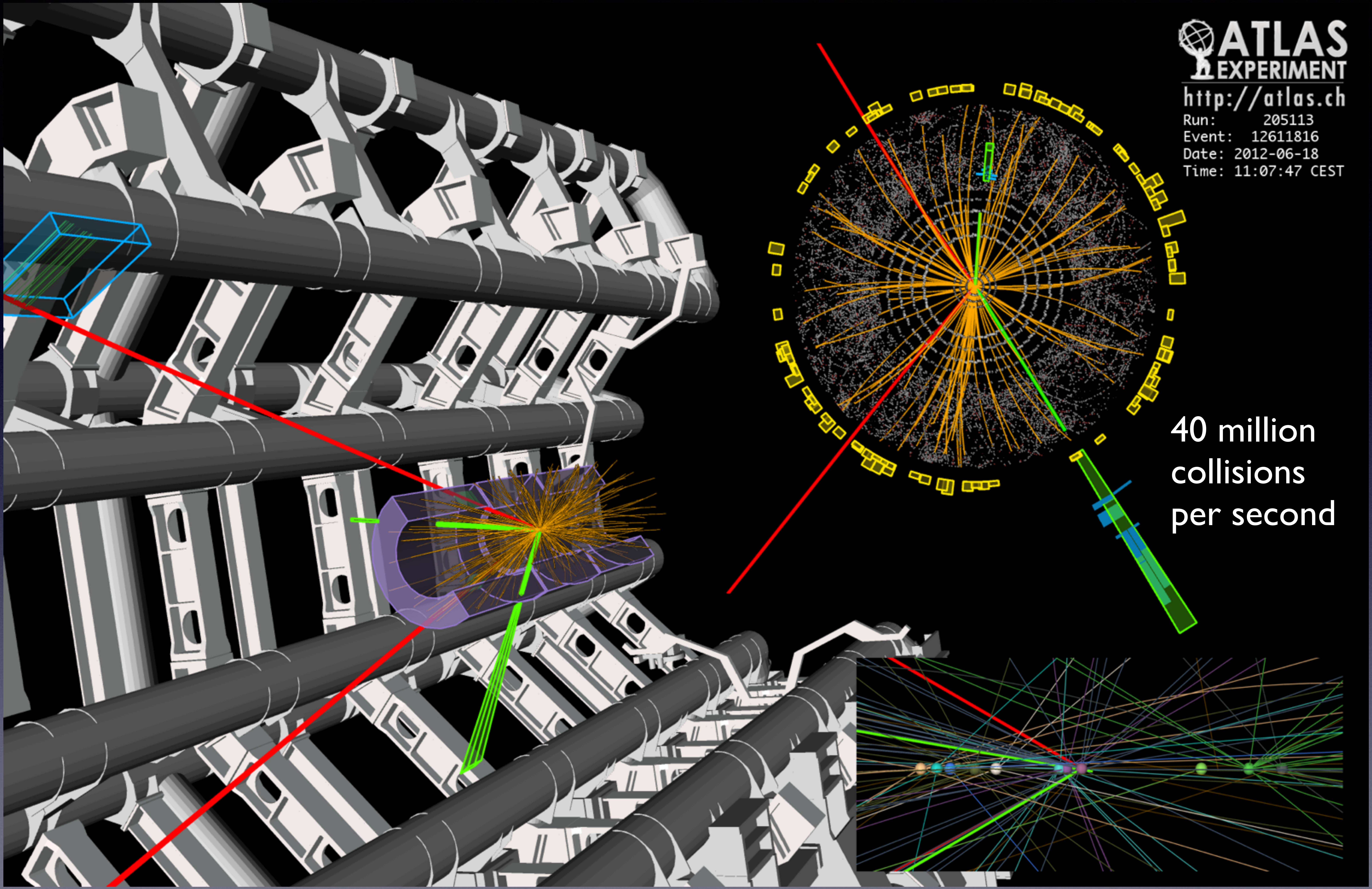

GENEVE
LHC - B
Point 8
CERN
ATLAS
Point 1
ALICE
Point 2
CMS
Point 5
SPS
TI 8
TI 2
LHC - B
ATLAS
ALICE
CMS
LEP/LHC

ATLAS
Detector
2005





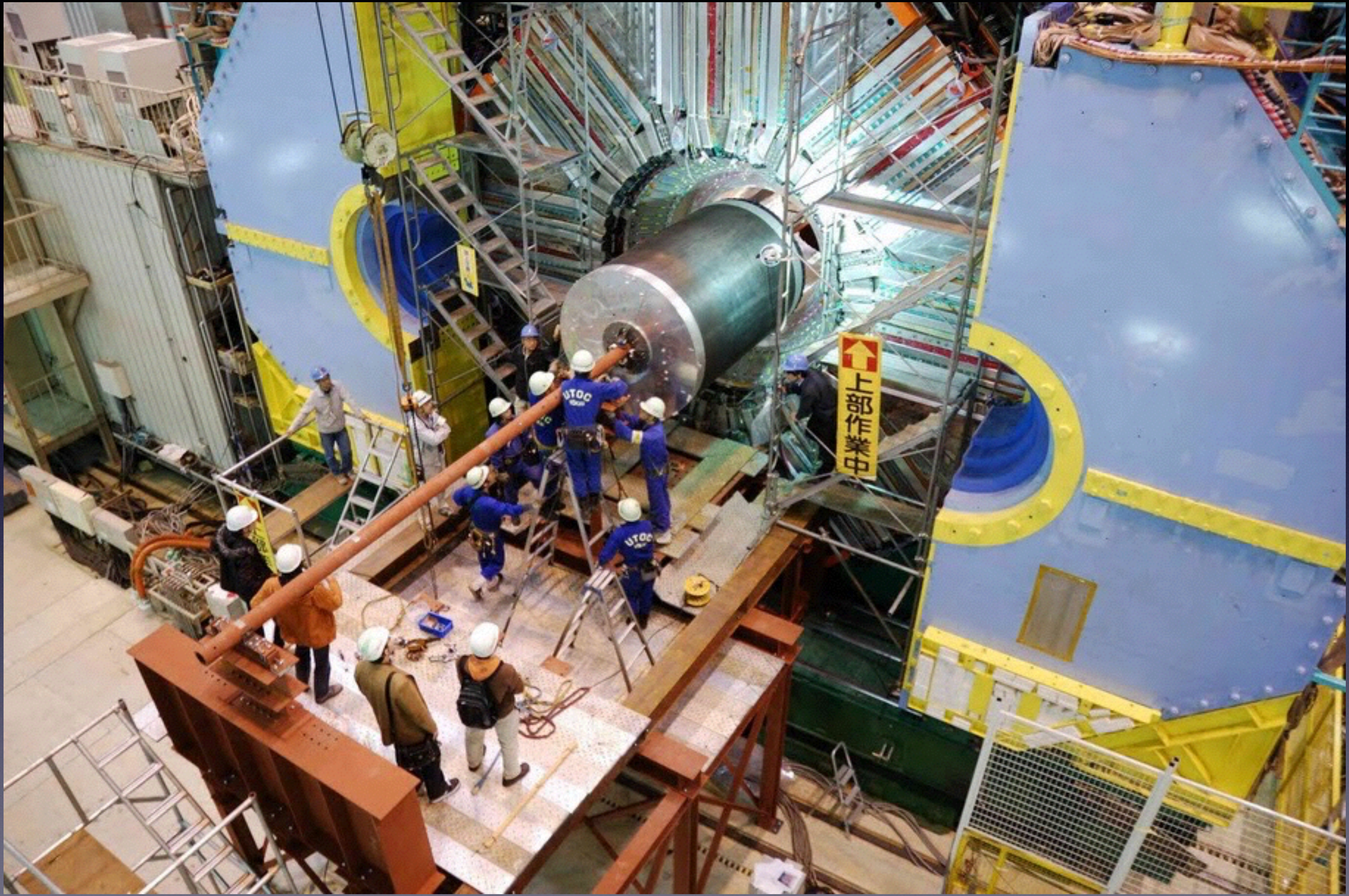
ATLAS
Detector
2014



40 million collisions per second

Belle II
Detector

KEK
Laboratory



Scale and other experiments

Each interesting ‘event’ stored on disk

ATLAS experiment roughly 170 PB on disk today, now growing all the time

LHC Experiments and other High Energy Physics experiments sure to grow to exascale in coming years.

Now down to the details.

High Energy Physics Computing workloads

- High Throughput Computing workload composed of mostly embarrassingly parallel tasks (jobs).
- Jobs for HEP are usually 1-24 hours in length and can be done single core, or multi core jobs (memory saving)
- Jobs are either Monte Carlo simulation of collisions or analysis of real collision data from the detector readout
- Most of the workload today is run on ethernet connected Linux clusters from 500 - 10000 cores at Research and Education institutions around the world
- On any given day there is roughly ~300K cores running HEP jobs for the Worldwide LHC Computing Grid (collection of non-cloud federated Linux clusters)

Our IaaS timeline

Today's Problem and Opportunity

We wish to be able to run across multiple clouds without having any 'special' relationship with those cloud providers. In other words we can't impose any requirements on them.

Components of the Solution

manage Jobs:

HTCondor
High Throughput Computing

manage VM instances:

Cloud Scheduler

Discover Web Caches:

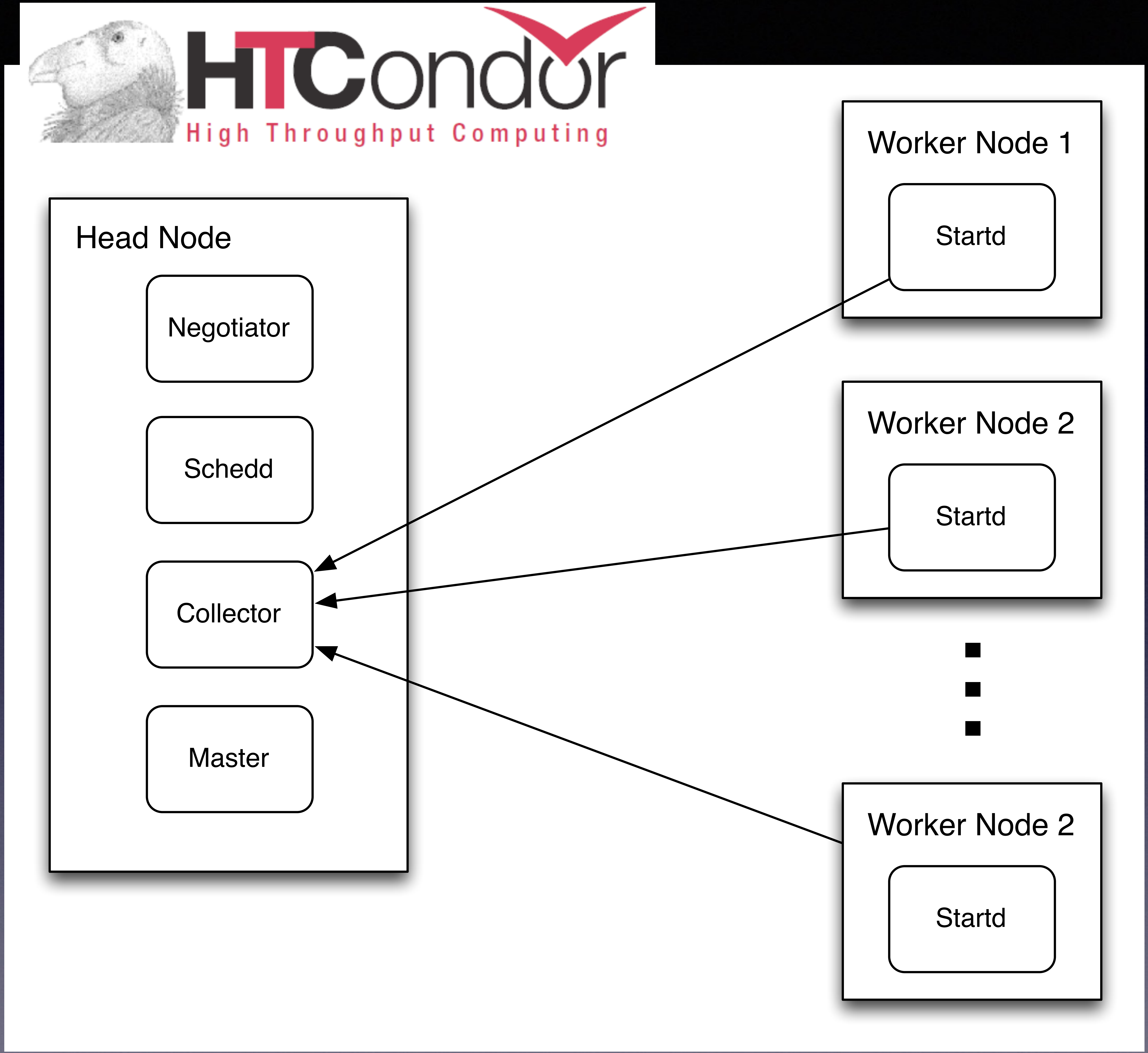
Shoal

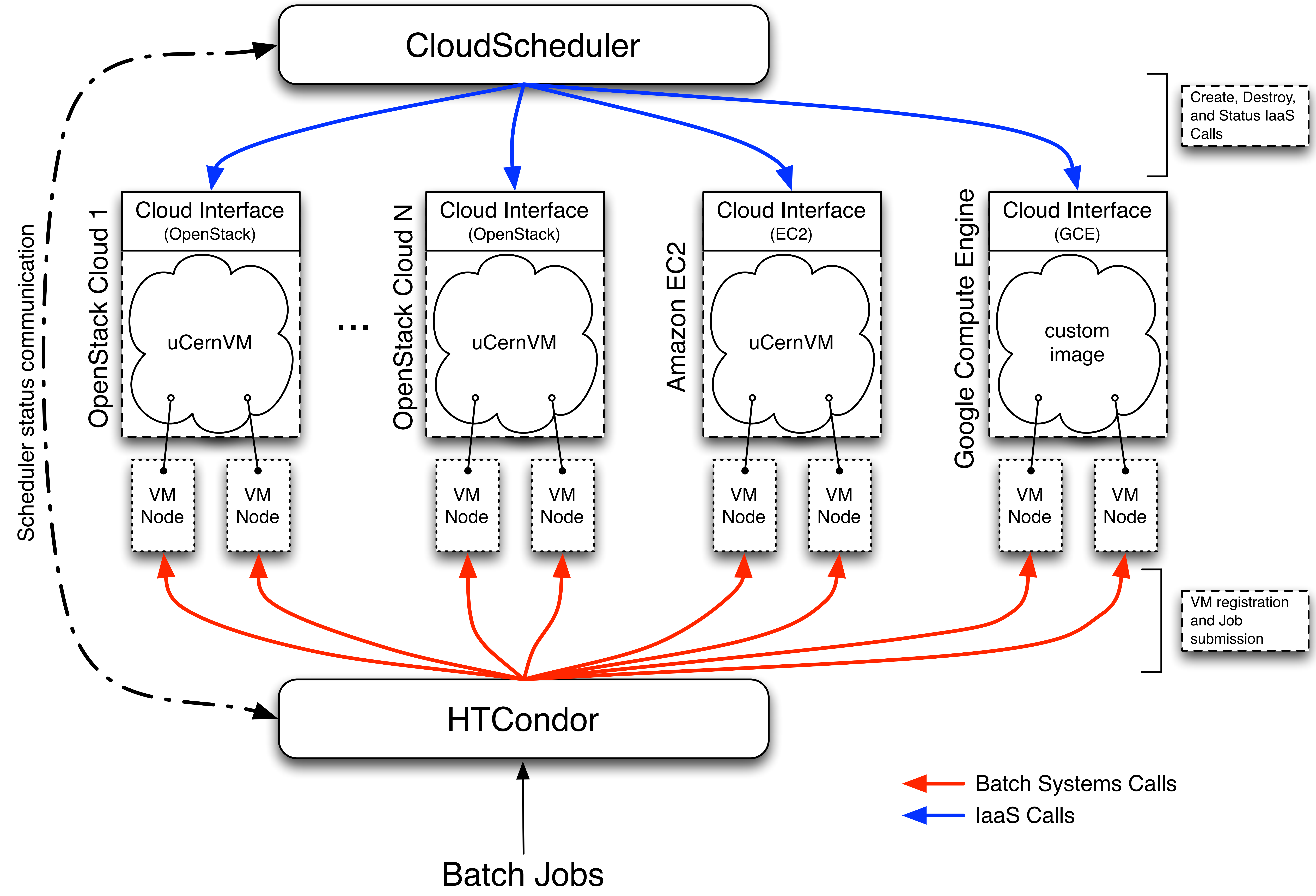
Manage VM images:

Glint

The VM itself:

CernVM +CVMFS


HTCondor
High Throughput Computing
Head Node
Negotiator
Schedd
Collector
Master
Worker Node 1
Startd
Worker Node 2
Startd
Worker Node 2
Startd


CloudScheduler
Create, Destroy, and Status IaaS Calls
Scheduler status communication
OpenStack Cloud 1
Cloud Interface
(OpenStack)
uCernVM
VM Node
VM Node
...
OpenStack Cloud N
Cloud Interface
(OpenStack)
uCernVM
VM Node
VM Node
Amazon EC2
Cloud Interface
(EC2)
uCernVM
VM Node
VM Node
Google Compute Engine
Cloud Interface
(GCE)
custom image
VM Node
VM Node
VM registration and Job submission
HTCondor
Batch Jobs
Batch Systems Calls
IaaS Calls

$ condor_submit atlas-sub.sub

```
universe = vanilla

# === job parameters ===
dir            = $ENV(HOME)/logs/analy
output         = $(Dir)/$(Cluster).$(Process).out
error          = $(Dir)/$(Cluster).$(Process).err
log            = $(Dir)/$(Cluster).$(Process).log
executable     = runpilot3-wrapper.sh
arguments      = -s ANALY_IAAS -h ANALY_IAAS -p 25443 -w https://pandaserver.cern.ch -u user
environment    = "ATLAS_SITE_NAME=IAAS APF_PYTHON26=1 RUCIO_ACCOUNT=pilot"
request_cpus   = 1
request_memory = 2000
request_disk   = 10000000
requirements   = VMType =?= "atlas-worker" && Target.Arch == "x86_64"
x509userproxy = $ENV(HOME)/atlaspt.proxy

# === job behaviour ===
stream_output           = False
stream_error            = False
notification            = Error
should_transfer_files   = YES
when_to_transfer_output = ON_EXIT_OR_EVICT

# === VM configuration for cloud scheduler ===
+VMName         = "PandaCern"
+VMAMI          = "ucernvm-prod.1.18-13"
+VMInstanceType = "c8-30gb-430"
+VMKeepAlive    = "30"
+VMJobPerCore   = "True"
+TargetClouds   = "IAAS"
+VMAMIConfig    = "/srv/userdata/IAAS.yaml:cloud-config,/srv/userdata/cernvm-data.txt:ucernvm-config"
+VMUseCloudInit = "True"
+VMInjectCA     = "False"
```

Cloud Scheduler Define Resources Available
Define resources available:

/etc/cloudscheduler/cloud_resources.conf

```
[chameleon]
auth_url:          https://proxy.chameleon.tacc.utexas.edu:5000/v2.0
cloud_type:        OpenStackNative
regions:           regionOne
tenant_name:       FG-54
vm_domain_name: .novalocal
key_name:          rd_key
networks:          FG-54-HEP-NET
security_group:    default
username:          *******
password:          ********
secure_connection: true
enabled:           false

[cc-east]
auth_url:          https://east.cloud.computecanada.ca:5000/v2.0
cloud_type:        OpenStackNative
regions:
tenant_name:       Belle
networks:          Belle_network
key_name:          rd_key
vm_domain_name:    .openstacklocal
security_group:    default
username:          ******
password:          ********
secure_connection: true
enabled:           false
```

Example Operational Task

Email on Friday:

“Hey Mike,

We are taking cloud-x down Tuesday at 9:00 central time can you make sure you aren't running anything important.

Cheers,

The friendly OpenStack Admins”

https://github.com/hep-gc/cloud-scheduler

State of the System on Monday Morning:

- 1000 Cores of Belle-II jobs running
- Each job is roughly 12 hours long and each job is in a different state of completion
- There are several thousands jobs waiting in the Condor job queue

Goal:

No users jobs are killed and all VMs are shutdown cleanly before 9:00 Tuesday

Operations on Monday Morning:

Prevent any more VMs from being booted:

$ cloud_admin -d cloud-x

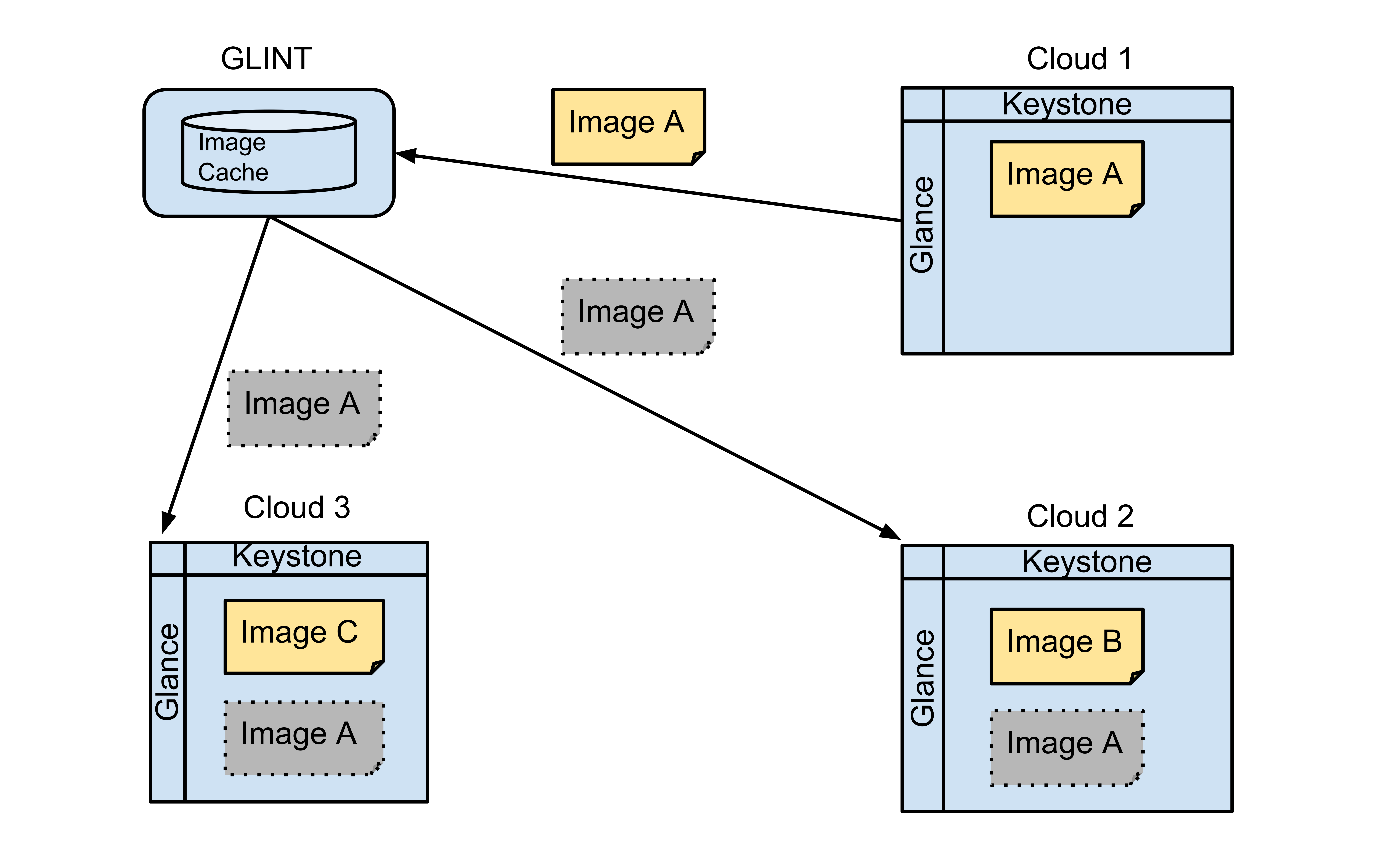
Stop submitting new jobs to running VM and shutdown the VM once all jobs are complete:

$ cloud_admin -o -c cloud-x -a https://github.com/hep-gc/cloud-scheduler

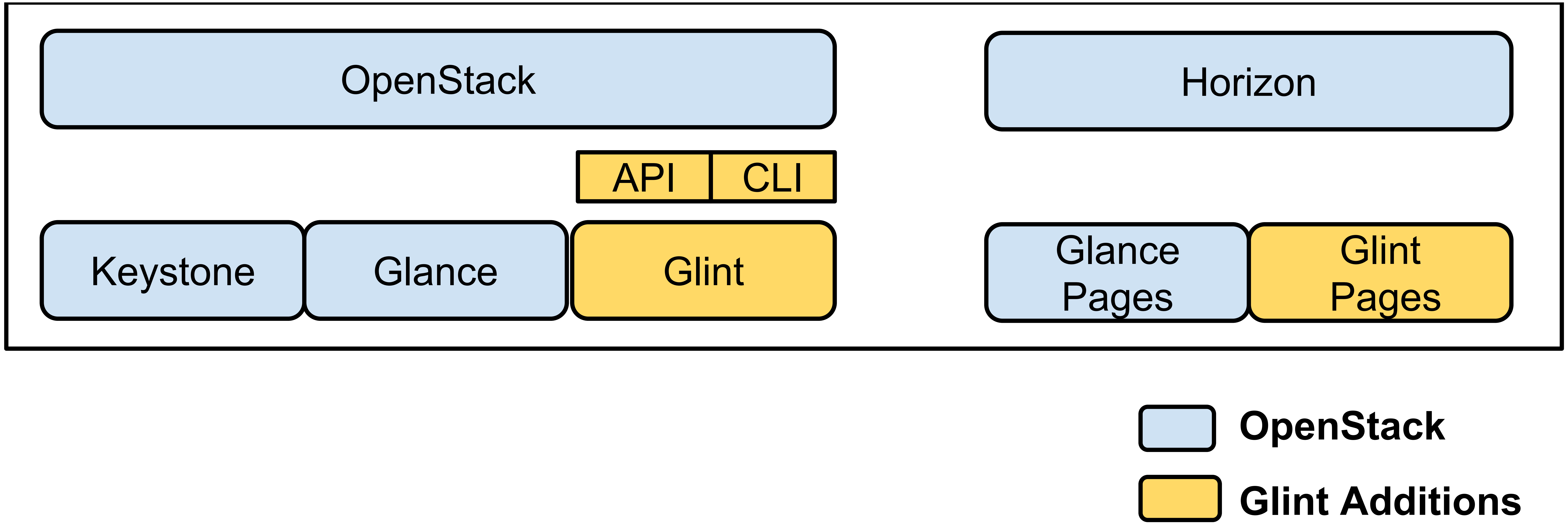
Problem: Too many clouds to manage VM images manually

Solution: Glint Image Distribution Service

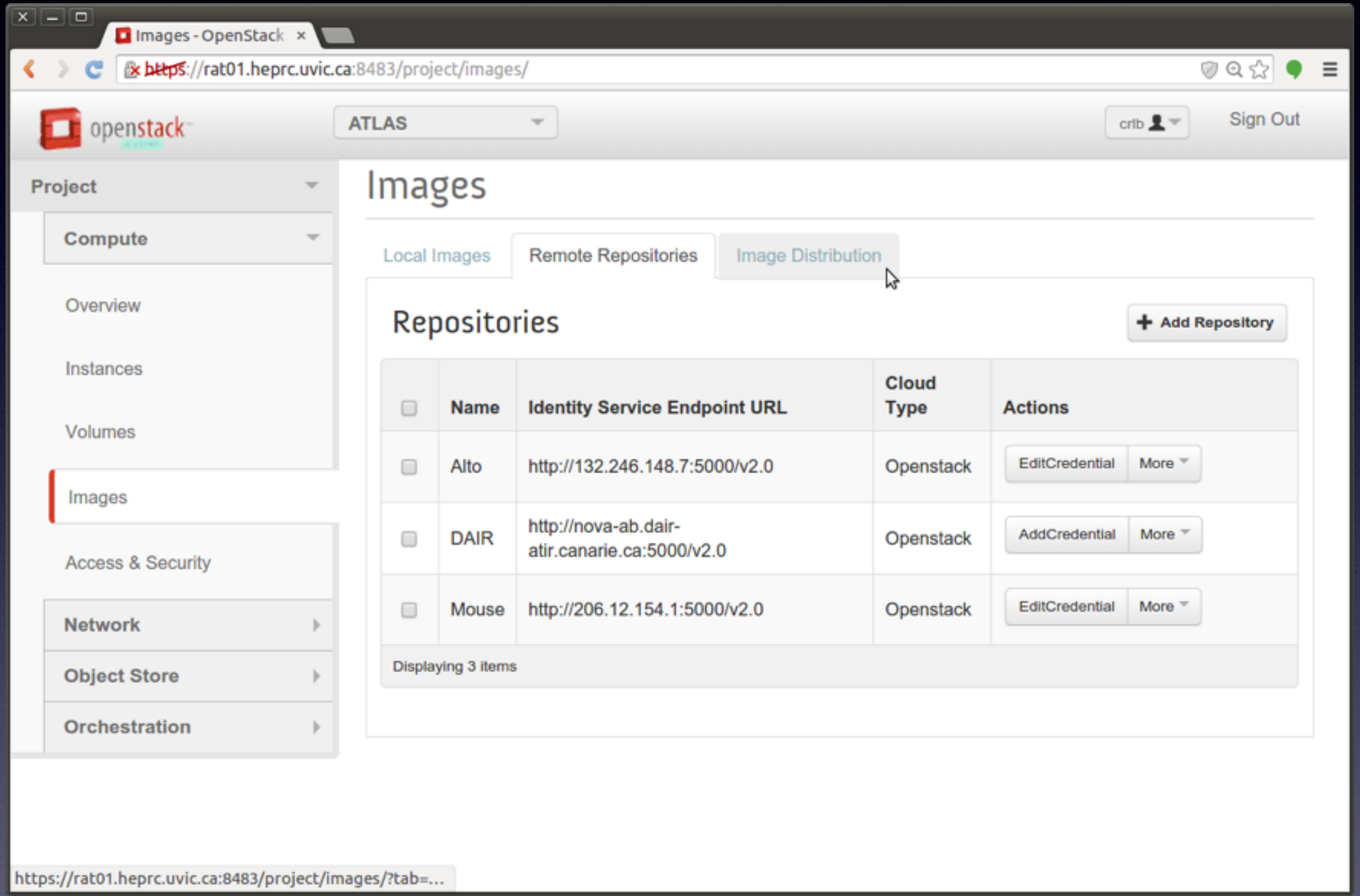
Glint

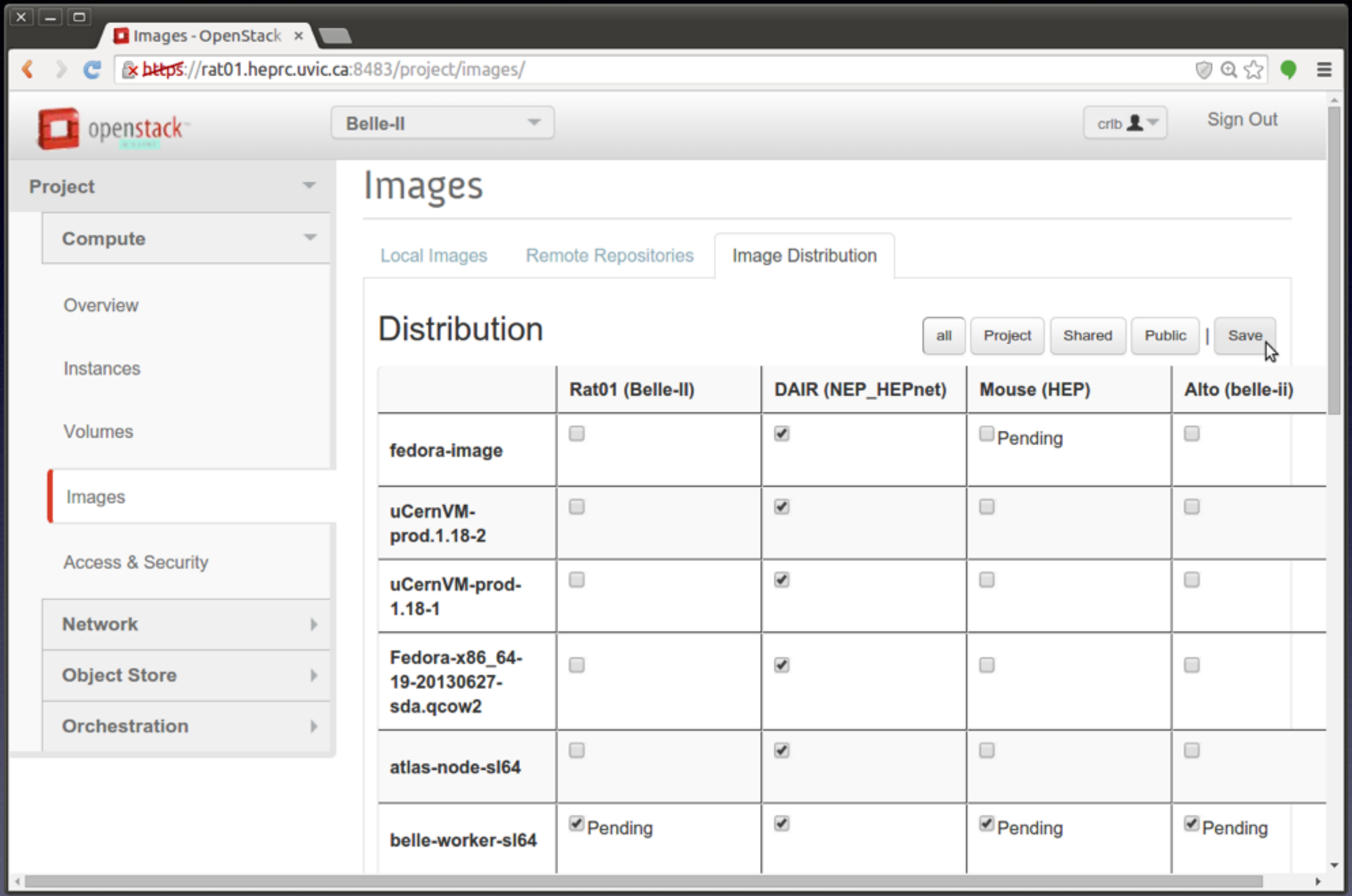

GLINT
Image
Cache
Image A
Cloud 1
Keystone
Glance
Image A
Image A
Image A
Cloud 3
Keystone
Glance
Image C
Image A
Cloud 2
Keystone
Glance
Image B
Image A

OpenStack with Glint



Horizon Interface with Glint Pages





Goals for Glint

We have learned a lot this week.

Take advantage of keystone federation. User won't have to provide creds for multiple clouds

Take Advantage of Glance Tasks

Ultimate goal to have the functionality in Glint available as a part of Keystone and Glance

PyPI:

https://pypi.python.org/pypi/glint-service/

launchpad:

https://launchpad.net/python-glint

Github:

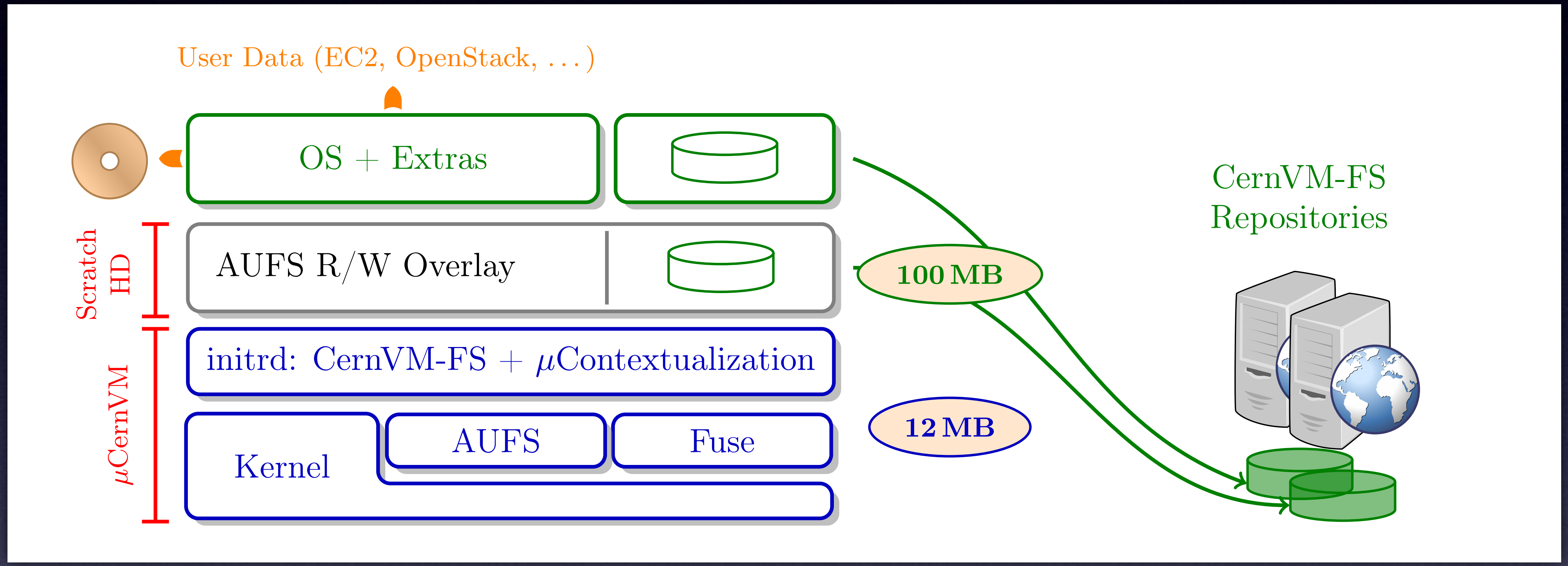
https://github.com/hep-gc/glint-service

more details contact Ron Demarais <rd@uvic.ca>

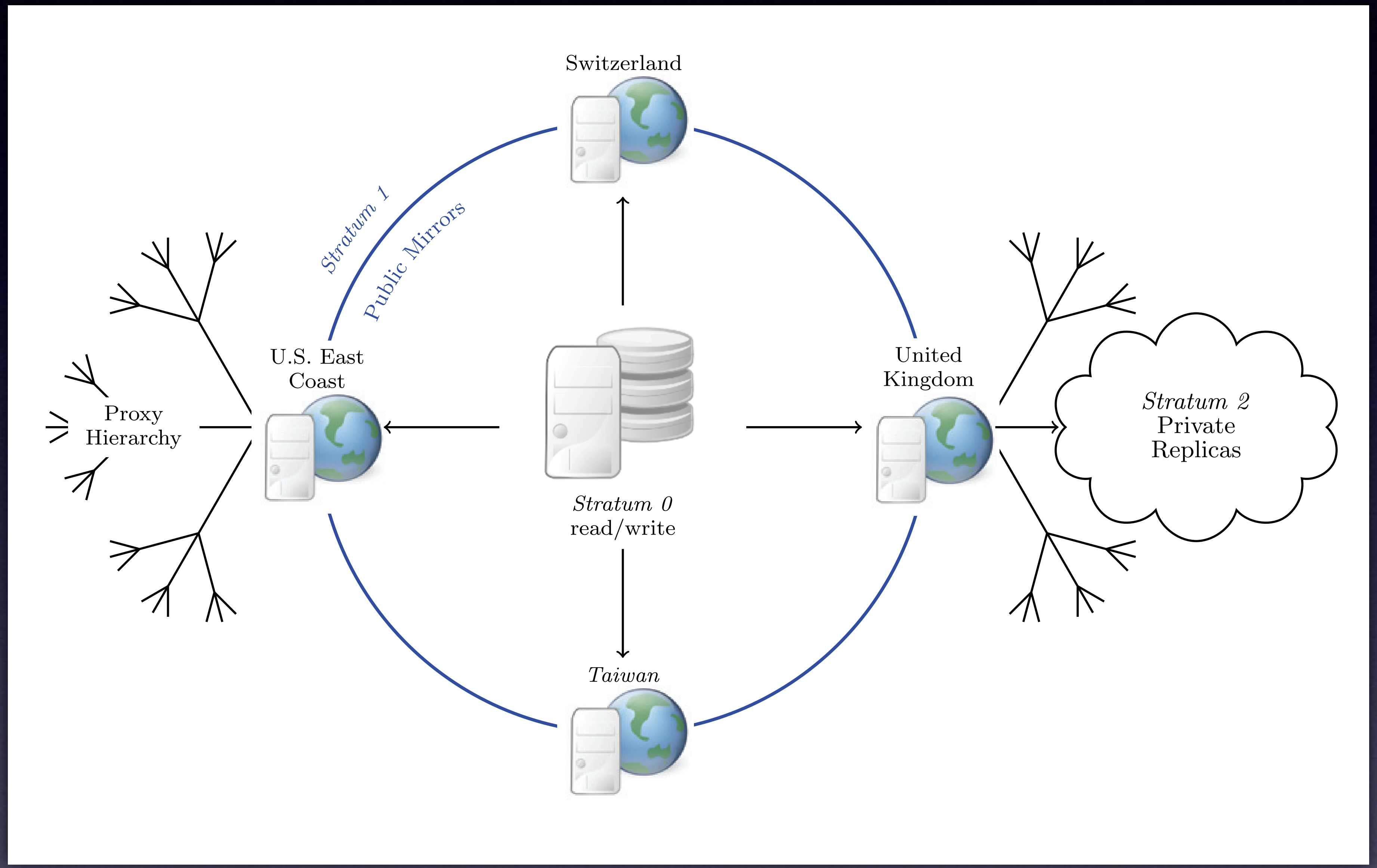
The Virtual Machine Image

CernVM and CVMFS

CernVM is RHEL compatible HEP software appliance in only 20 MB



CVMFS is a caching network file system based on HTTP and optimized for software, i.e. millions of small files



comes with it's own CDN!

http://cernvm.cern.ch

Requires fast and near HTTP cache

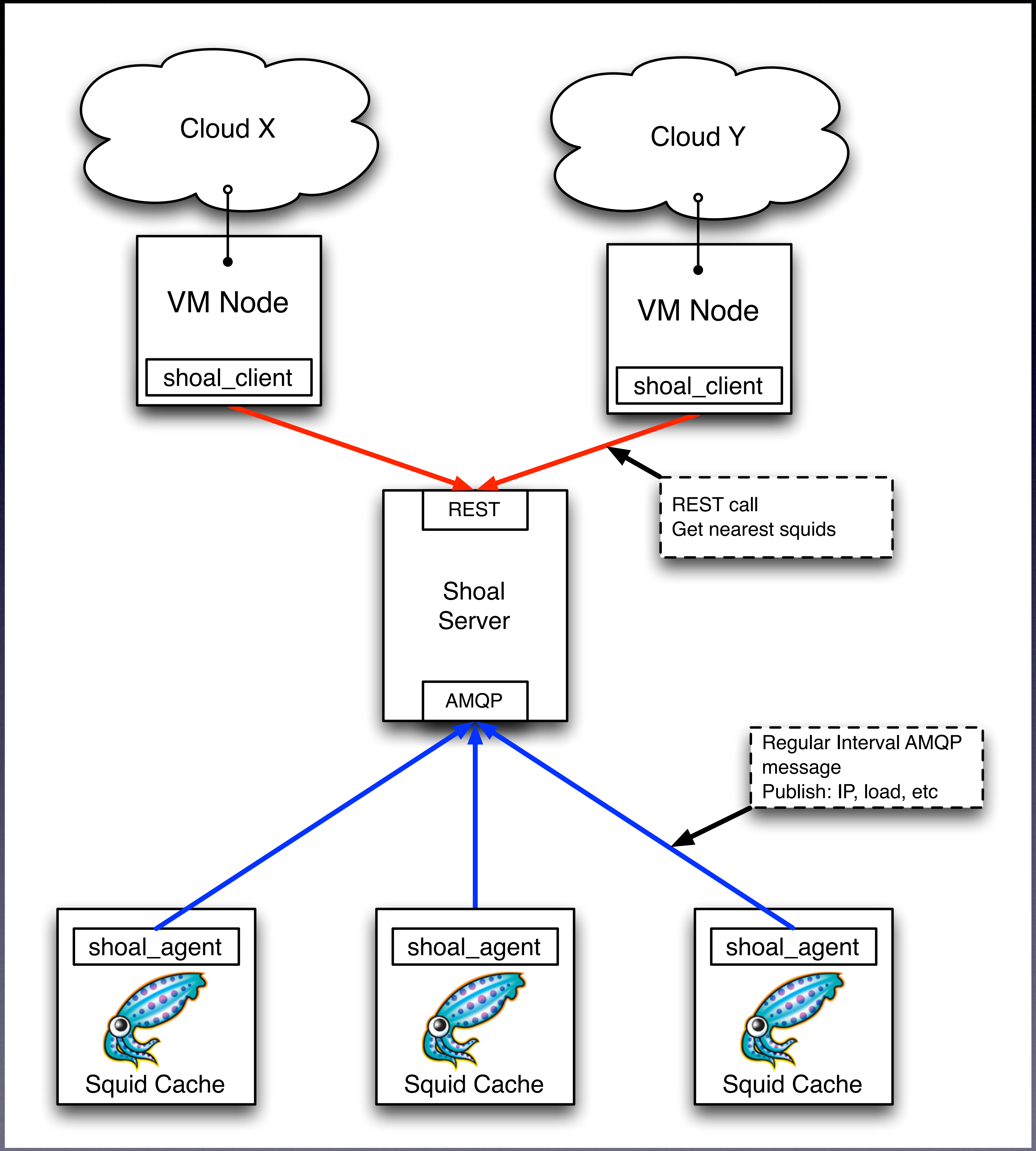
The caching challenge on IaaS cloud

When booting VMs on different arbitrary clouds they don't know which squid they should use

In order to work well, VMs need to able to access a local web cache (squid) to be able to efficiently download all the experiment software and now OS libraries they need to run

If a VM is statically configured to access a particular cache it can be slow (Geneva → Vancouver for example) and it can get overloaded

Shoal



uses the highly Scalable AMQP protocol to advertise Squid servers to Shoal

uses GeoIP information to determine which is the closest to each VM

Squids advertise every 30 seconds, server verifies if the squid is functional

https://github.com/hep-gc/shoal


Total Number of Jobs by Site
9 Weeks from Week 11 of 2015 to Week 20 of 2015
DIRAC.UVic.ca
LCG.Napoli.it
LCG.DESY.de
20.7%
Cloud
500K
jobs
LCG.KIT.de
LCG.Pisa.it
LCG.SIGNET.si
LCG.DESY.de 1149081.0
DIRAC.UVic.ca 542687.2
LCG.Napoli.it 495470.3
LCG.KEK2.jp 437108.0
LCG.MPPMU.de 282603.1
LCG.KMI.jp 220146.9
LCG.Cosenza.it 201754.5
LCG.KIT.de 193038.2
LCG.Pisa.it 187461.1
LCG.SIGNET.si 185874.2
LCG.Frascati.it 149875.8
LCG.CNAF.it 133398.2
DIRAC.PNNL.us 131282.3
LCG.CESNET.cz 131141.1
DIRAC.BINP.ru 129298.9
LCG.UA-ISMA.ua 111533.7
LCG.HEPHY.at 108294.0
LCG.CYFRONET.pl 105986.6
LCG.NCHC.tw 84017.3
LCG.McGill.ca 71695.1
CLOUD.CC1_Krakow.pl 65714.2
CLOUD.AWS_Sydney.au 59909.0
LCG.KISTI.kr 52582.5
LCG.Torino.it 51849.1
CLOUD.AWS_Tokyo.jp 46835.6
LCG.ULAKBIM.tr 33997.4
LCG.Melbourne.au 31961.6
LCG.NTU.tw 20650.9
DIRAC.Nara-WU.jp 19279.2
DIRAC.Yonsei.kr 17442.9
LCG.Legnaro.it 17131.3
DIRAC.TIFR.in 16769.0
SSH.KMI.jp 16397.7
DIRAC.Osaka-CU.jp 15627.5
DIRAC.Niigata.jp 14199.1
DIRAC.Tokyo.jp 10612.3
DIRAC.CINVESTAV.mx 10336.9
DIRAC.TMU.jp 3333.0
DIRAC.Yamagata.jp 2963.4
DIRAC.Beihang.cn 2495.4
CLOUD.AWS_Singapore.sg 1057.0
ANY 829.0
... plus 3 more
Generated on 2015-05-20 06:36:54 UTC
Virtual Machines
s+Australia)
ERN
nada
UK
May 2015
Total: 3,160,227 , Average Rate: 0.07 /s
15 16 17
cc-east
nefos
dair-qc

Summary

CloudScheduler/HTCondor flexible way to run Batch Jobs on Clouds.

Key enabling technologies for this:

- CVMFS + CernVM
- Shoal: dynamic Squid cache Publishing
- Glint: VM Image Distribution

Current users ATLAS, Belle II, CANFAR, Compute Canada HPC consortium

Acknowledgements

University of Victoria Systems

canarie

Imperial College London

Chameleon

CERN

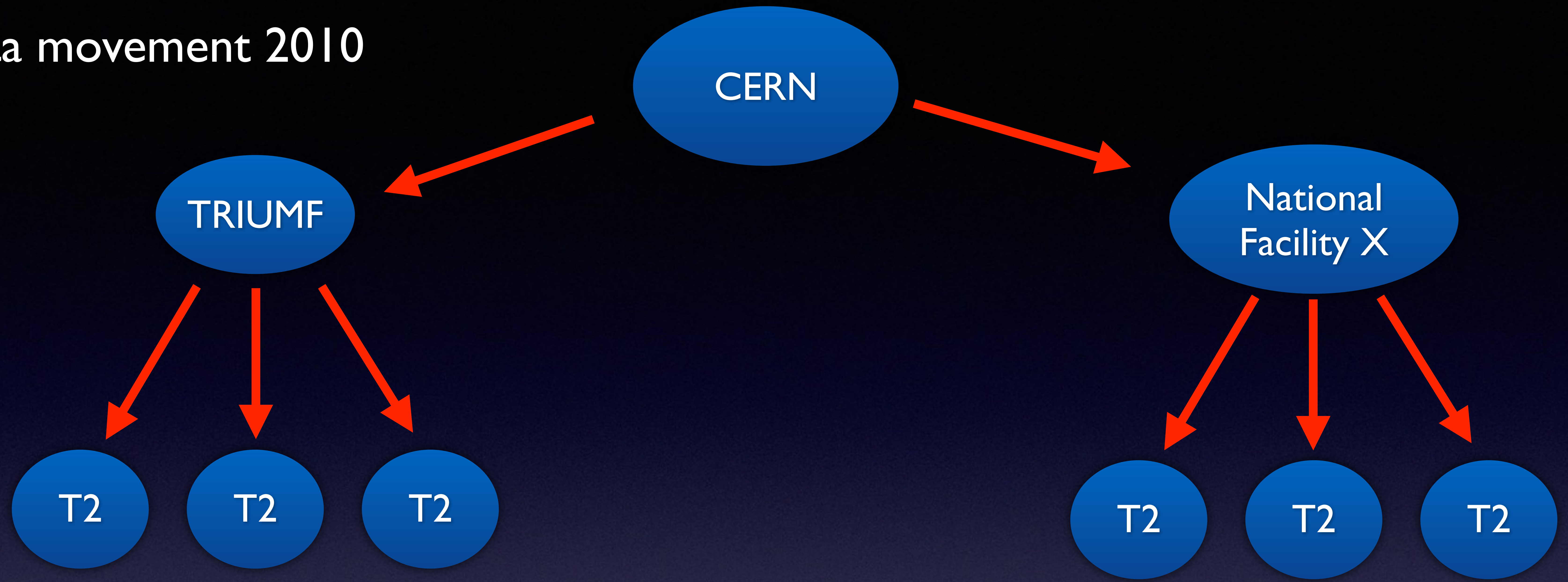
cybera

compute canada | calcul canada

amazon.com

nectar

Data movement 2010



Data movement 2015