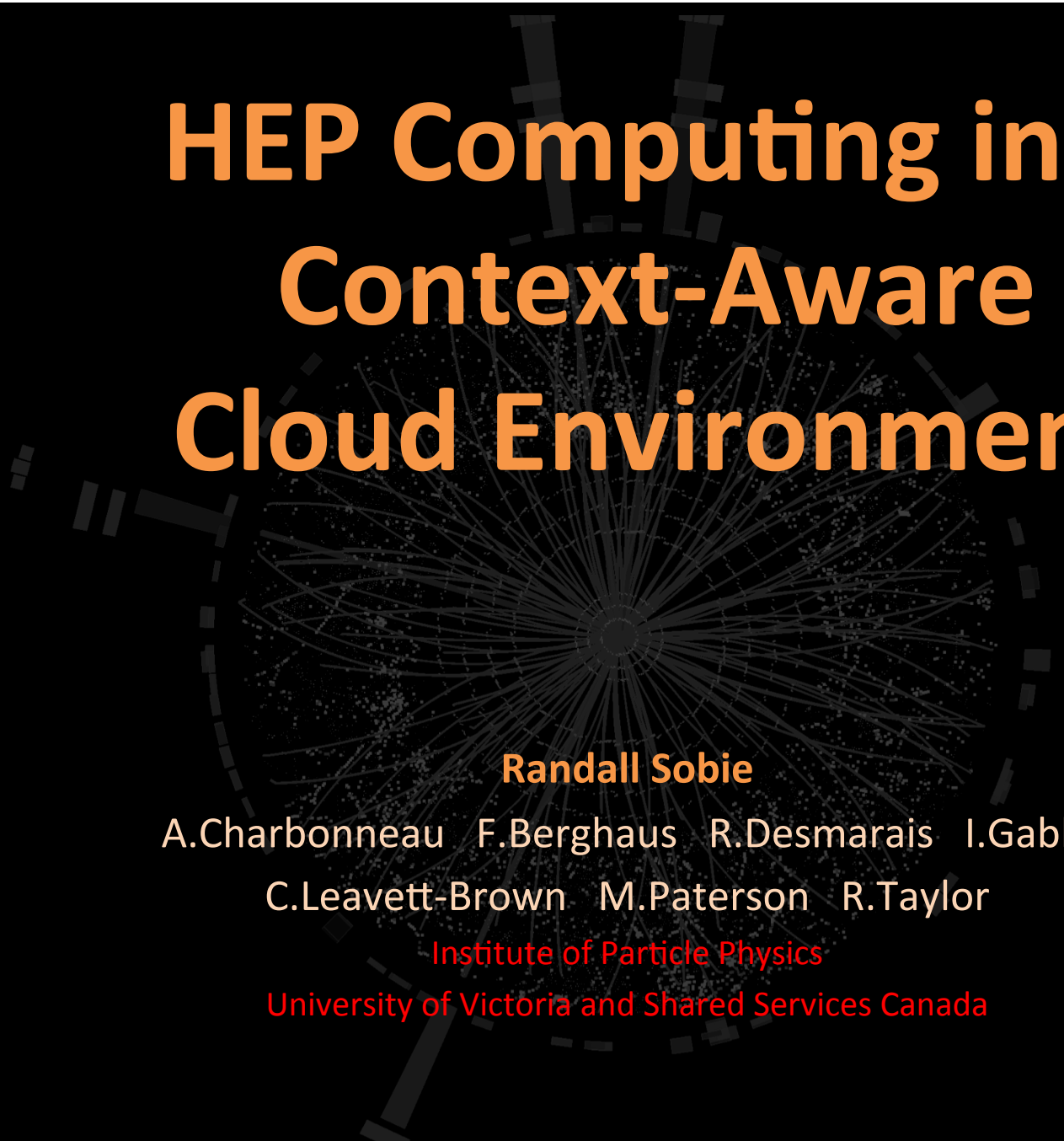
HEP Computing in a Context-Aware Cloud Environment



Randall Sobie

A.Charbonneau F.Berghaus R.Desmarais I.Gable

C.Leavett-Brown M.Paterson R.Taylor

Institute of Particle Physics

University of Victoria and Shared Services Canada

Motivation

Science is becoming a global endeavour

Particle physics (HEP) requires large machines that can not be funded by a single country

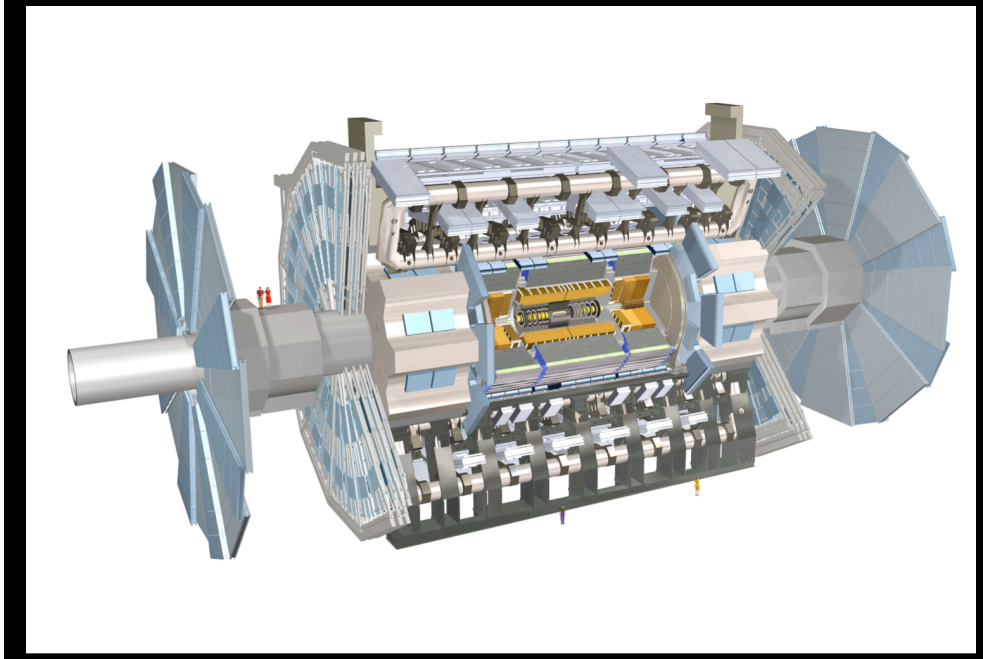
Use computing resources distributed around the world to analyze the vast data sets

CERN Laboratory in Geneva

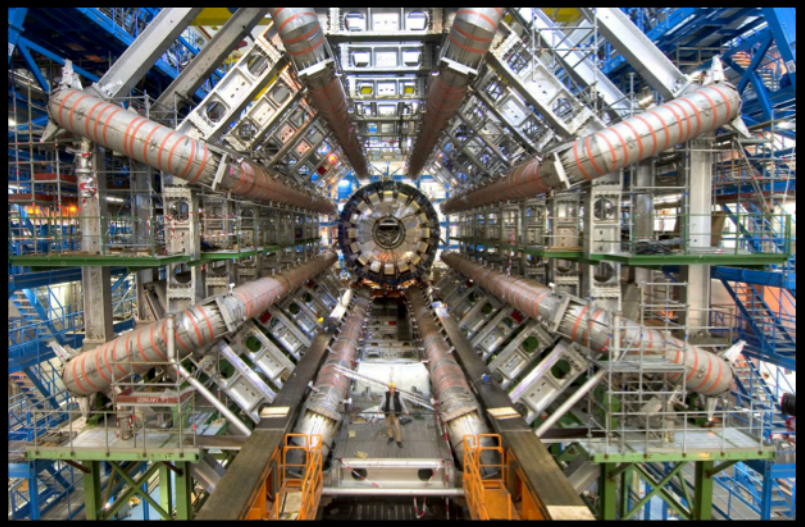
Largest man-made machine in the world
27 km circumference
Colliding protons-on-protons at very high energy

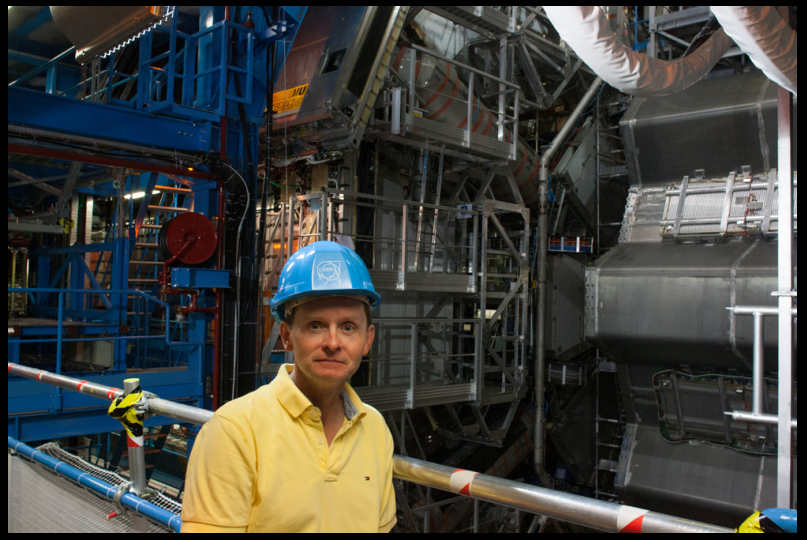
Goal is to find the Higgs boson, dark matter or other new physics particles

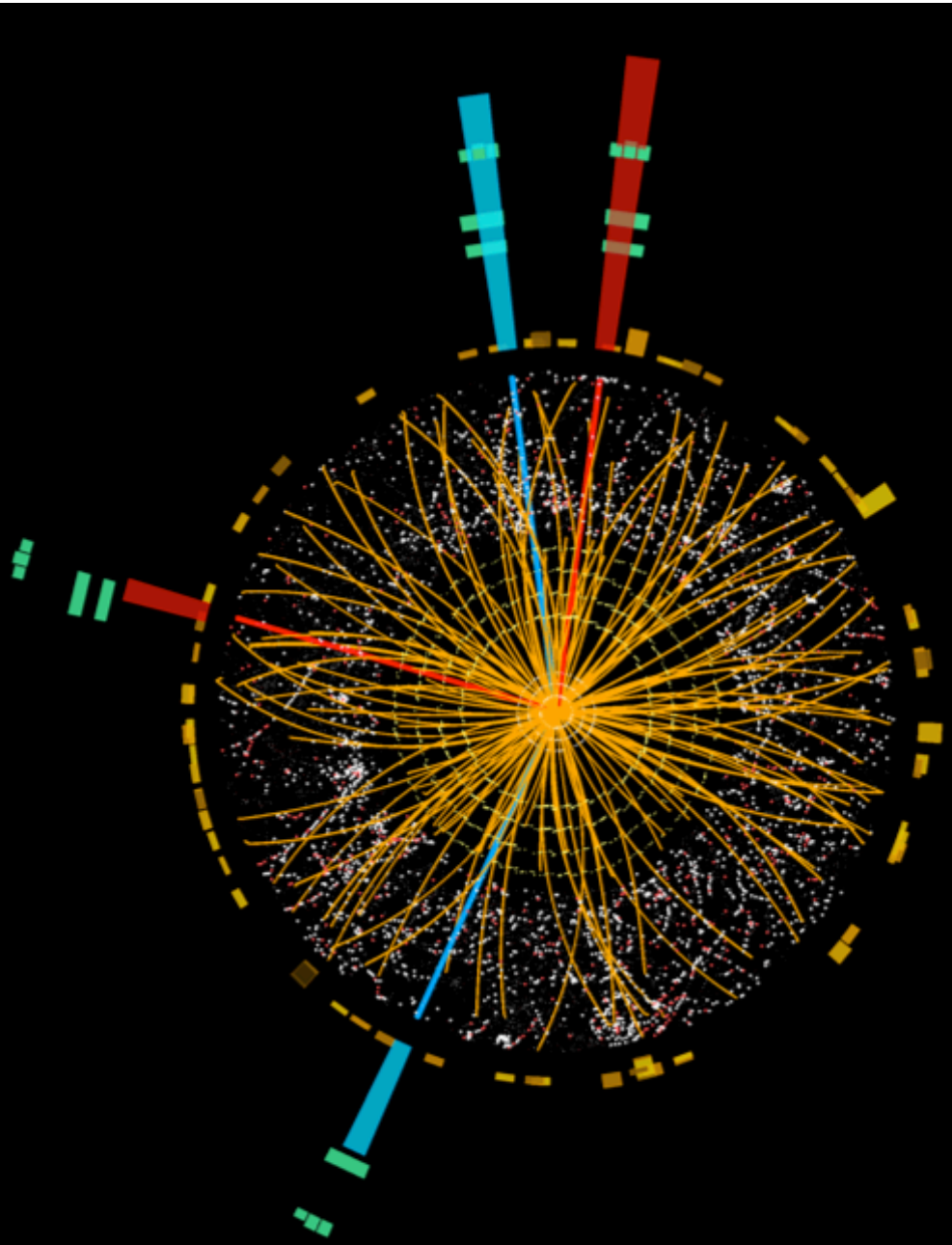
ATLAS Detector



44 m wide 25 m diameter







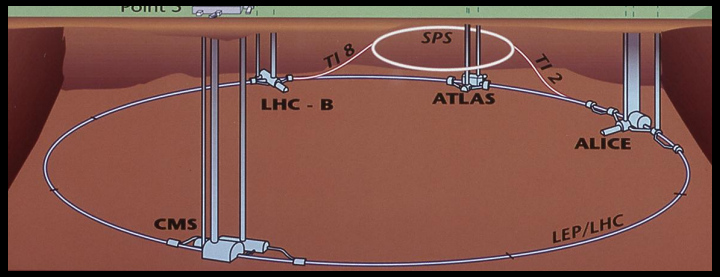
Event image (Higgs candidate)

In the plane perpendicular to the proton beams

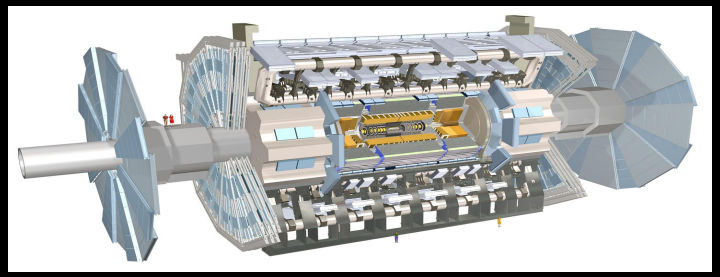
Yellow tracks are charged particles
Curvature indicates sign of he charge

Coloured boxes indicate energy deposition

Each "event" is about 2 MB
Collect 200 events per second



40 million collisions per second



100,000 collisions selected

200 events per second

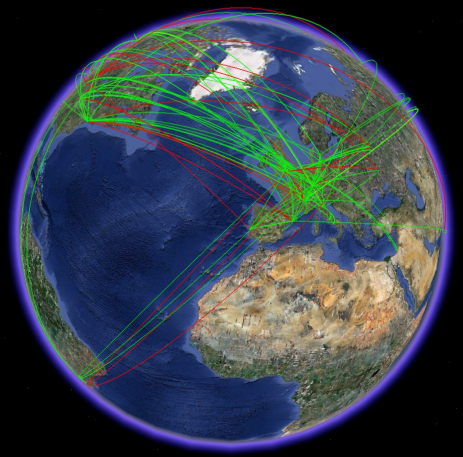
WLCG Computing Grid

CERN Tier 0

10 Tier1 sites

60+ Tier2 sites

150 PB data



Clouds in High Energy Physics

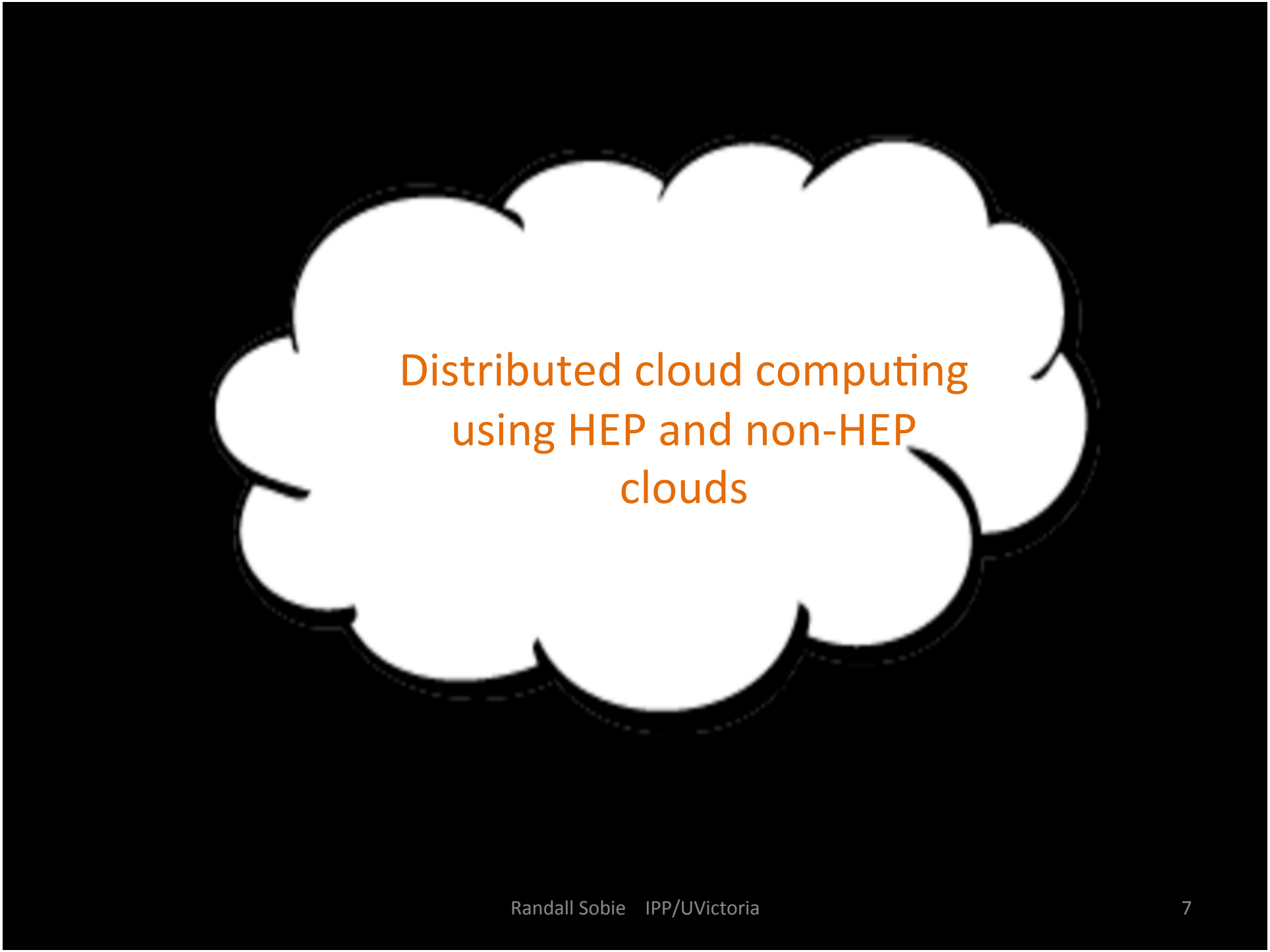
Long-term preservation of software and data of HEP experiments

Distributed cloud computing using HEP and non-HEP clouds

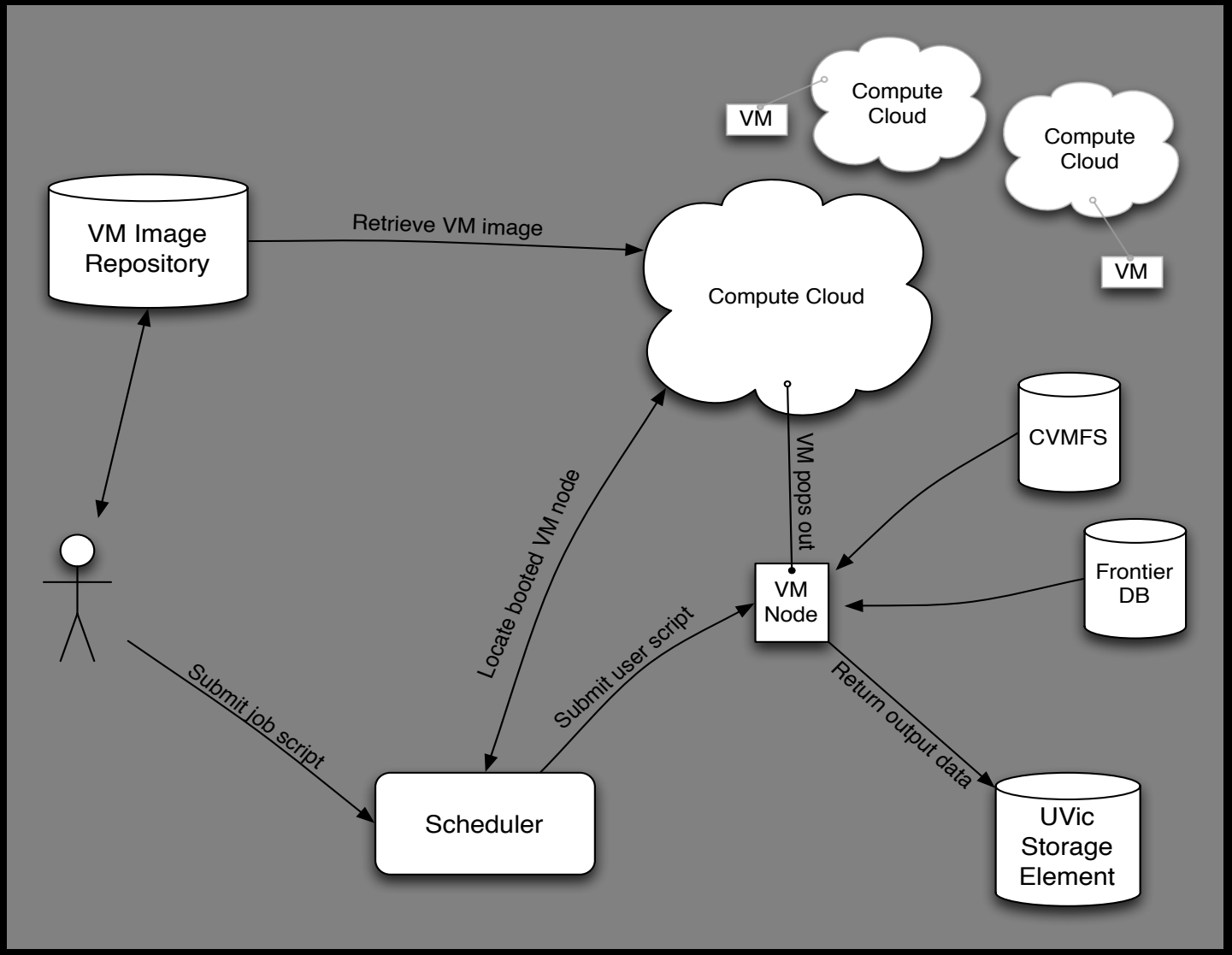
Utilize special computing resources attached to the detectors

Use commercial clouds for exceptional computing demands

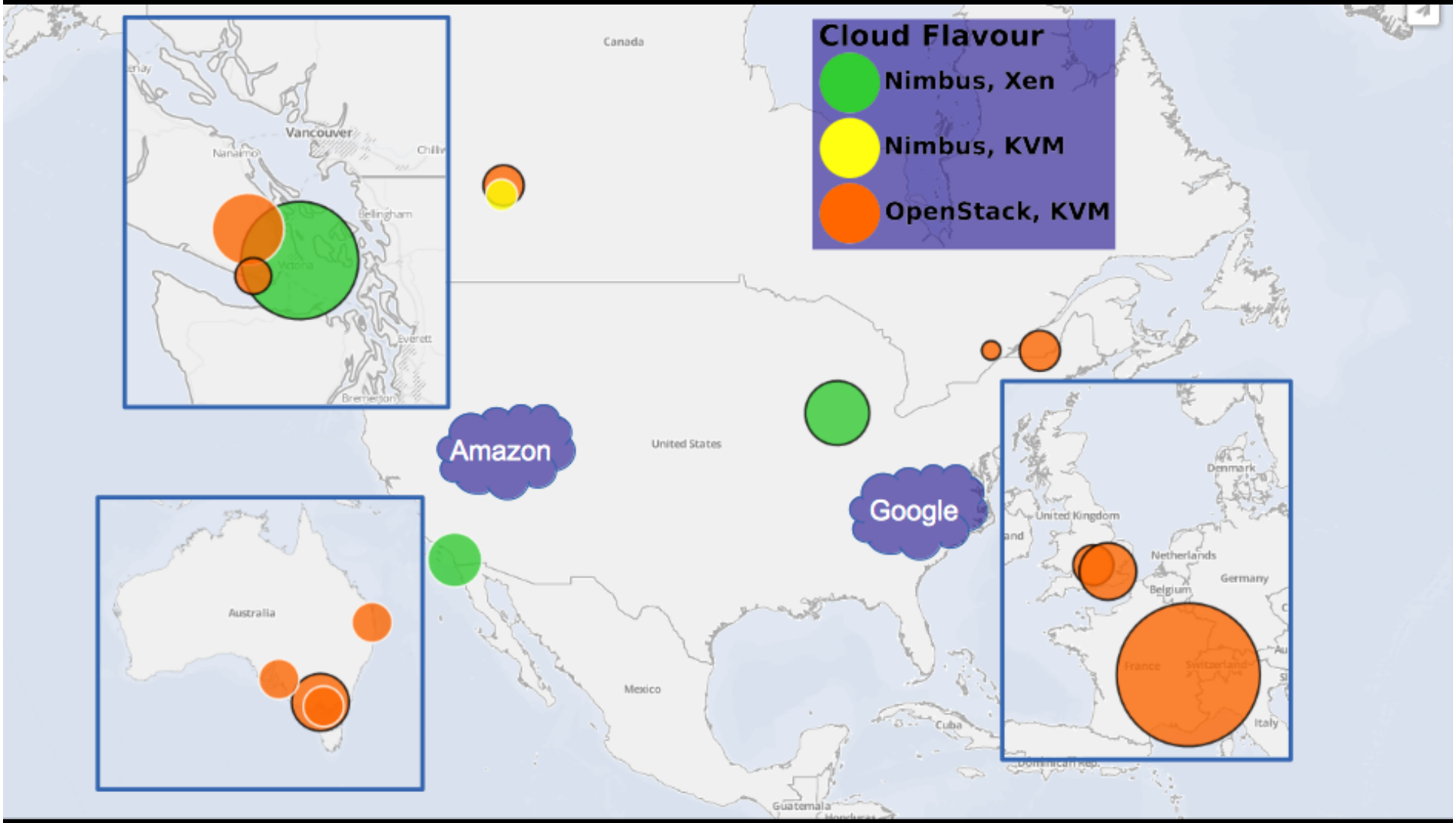
Simplify the management of heterogeneous in-house resources


Distributed cloud computing
using HEP and non-HEP
clouds

Overview of distributed compute cloud



Clouds


Cloud Flavour
Nimbus, Xen
Nimbus, KVM
OpenStack, KVM
Amazon
Google
Canada
United States
Mexico
Vancouver
Australia
United Kingdom
Netherlands
Germany
Belgium
France
Switzerland
Italy
Denmark

Operate three HEP HTCondor/CS Instances

ATLAS

North America (2.5 years)
Europe (6 months)

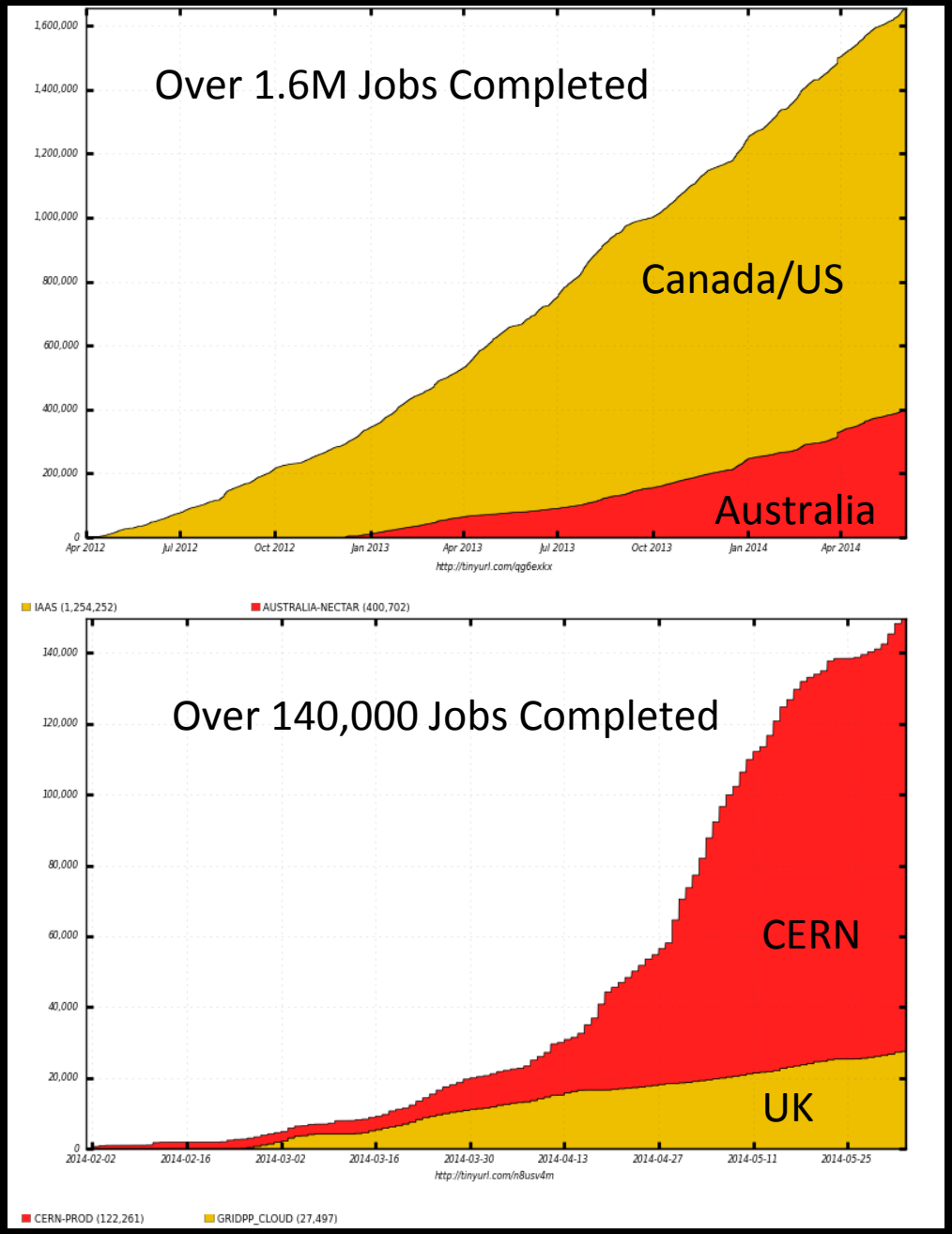
BelleII

North America (6 months)

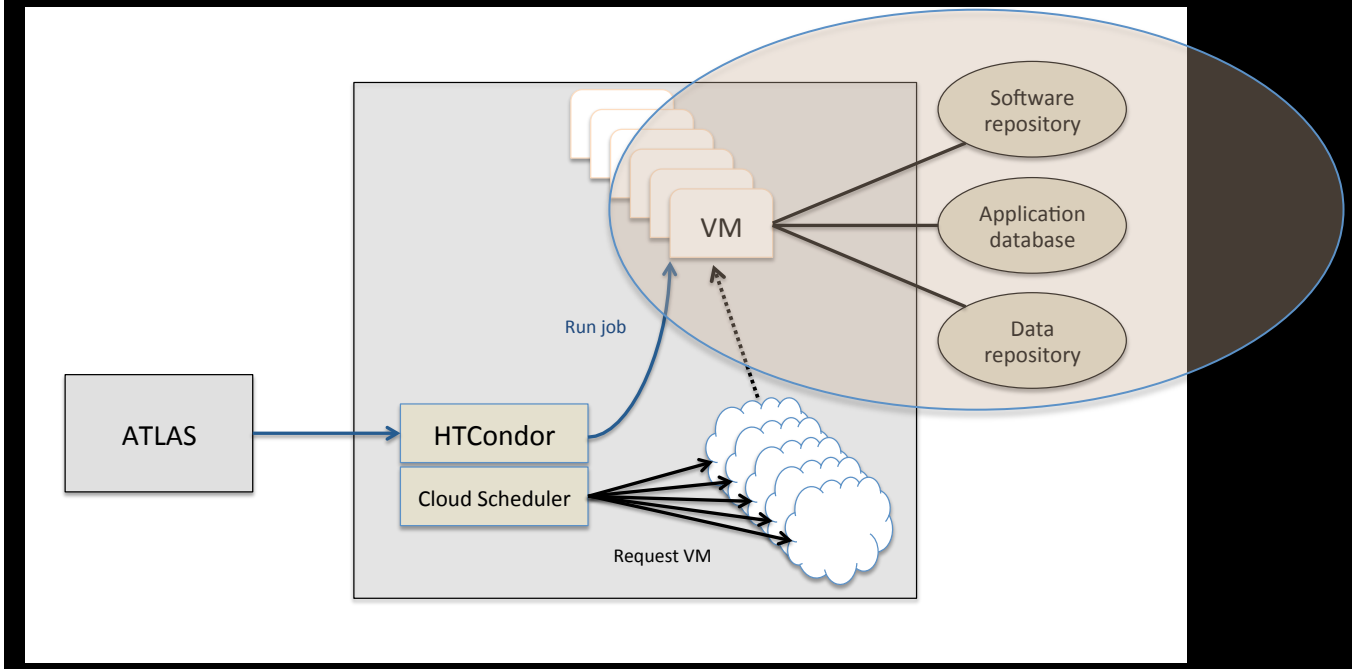
Total work loads average 1500 cores
Peak work loads 2500 cores

1000 cores on Amazon EC2

Working on API to Google GCE



Need for a *context-aware* cloud



VMs can appear anywhere in the world

Software repository and application database servers are fixed nodes

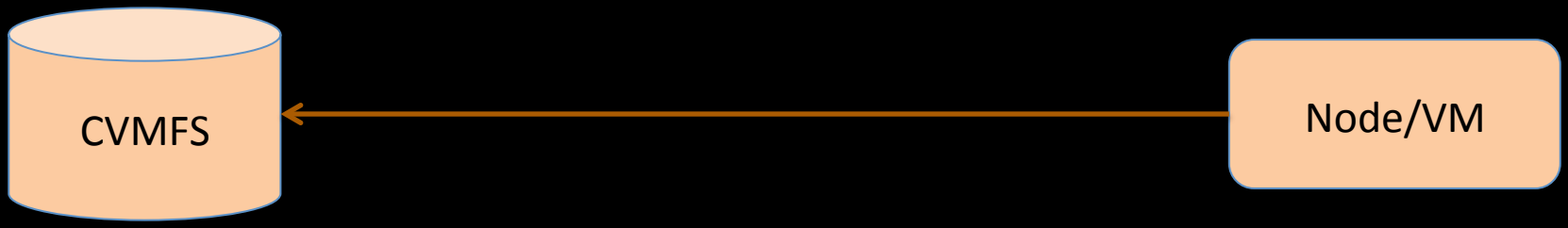
No input data repository

Single output data repository

We also have access to repositories distributed around the world

How can we make the system aware of these repositories?

Software distribution

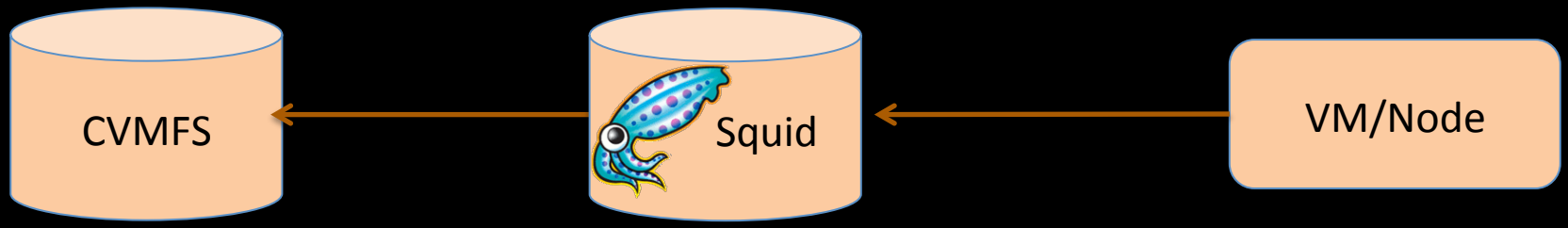


CernVM file system (CVMFS)

A network file system based on HTTP and optimized to deliver experiment software in a fast, scalable, and reliable way.

Files and file metadata are aggressively cached and downloaded on demand.

Software distribution



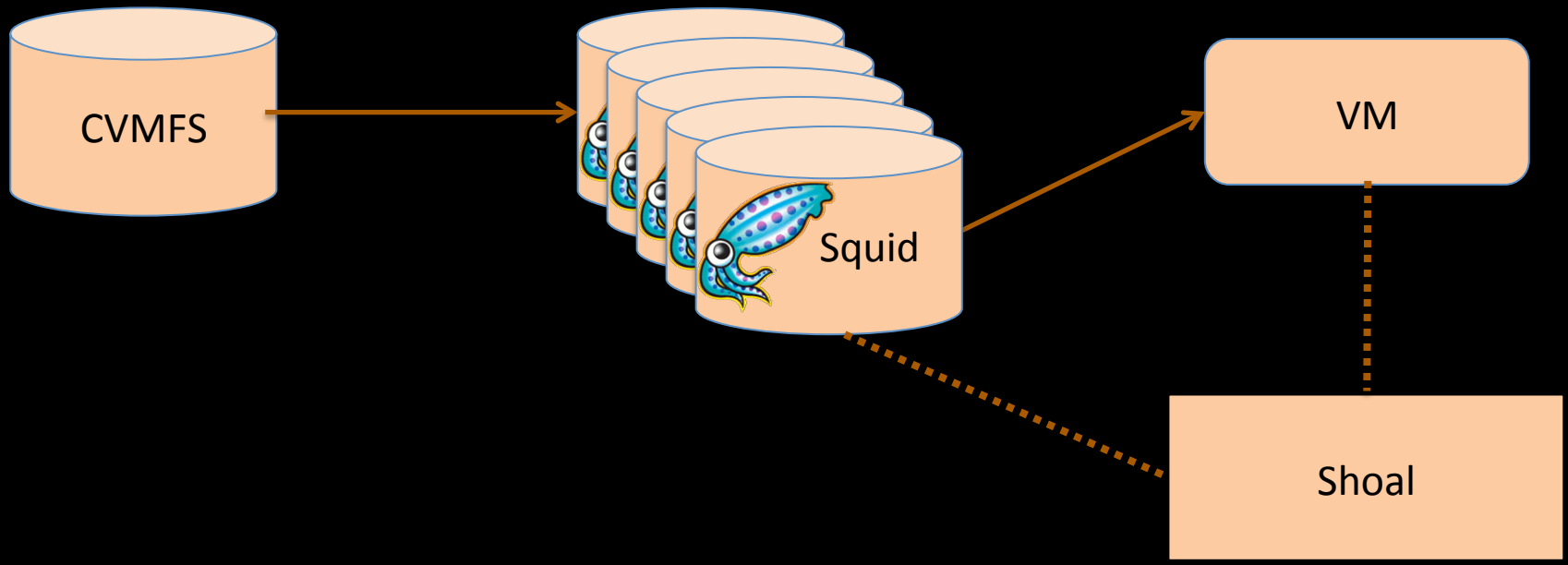
Squid HTTP Cache

Rather than query the CVMFS master repository we use Squid HTTP caches

Approximately 100 Squid caches distributed around the world

Squids distribute the world load but we still use a single Squid (per HTCondor/CS instance)

Software distribution in a context-aware cloud

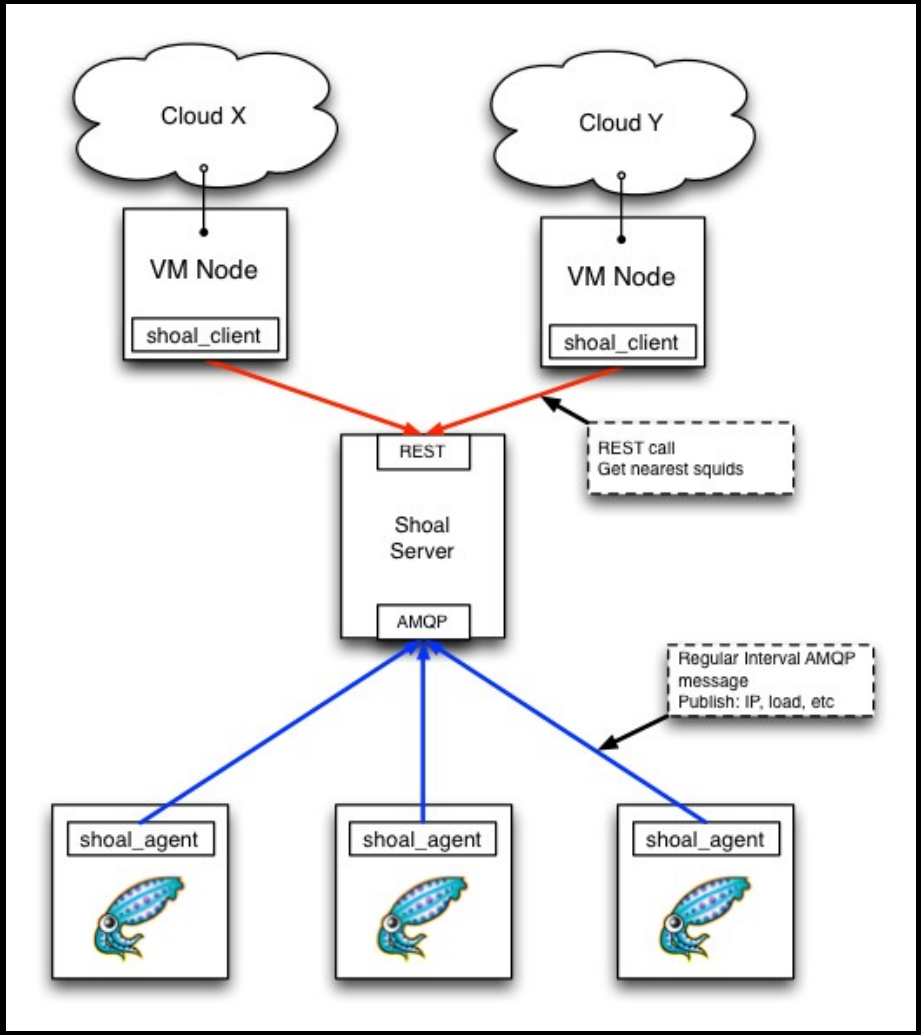


Shoal – a group of Squids

Shoal keeps a list of active Squid caches

Discovers new Squids and removes inactive Squids

VM queries Shoal for the location of the nearest squid (fastest network, lowest load)



Shoal

http://arxiv.org/pdf/1311.0058v1.pdf

github.com/hep-gc/shoal

Shoal used in production operation

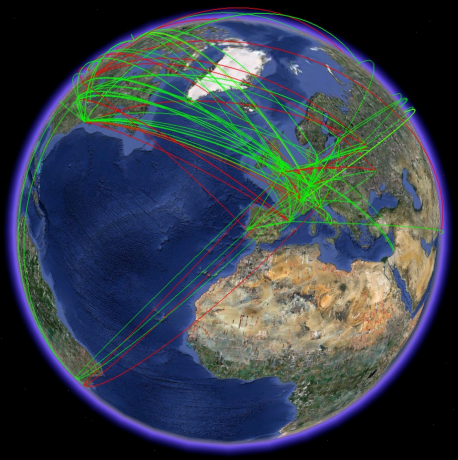
Allows for dynamic creation of Squids

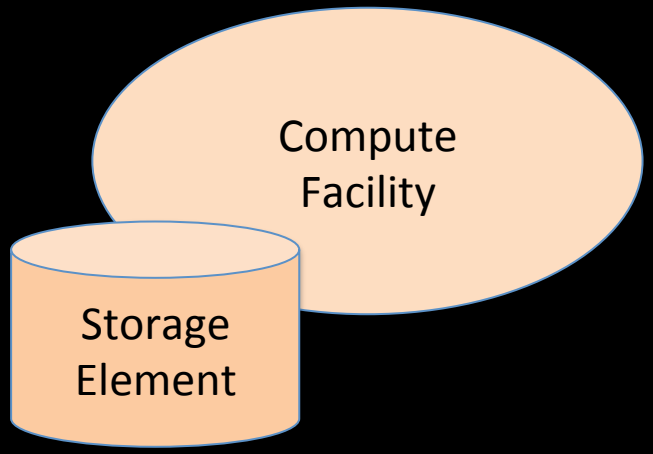
A cloud could boot a Squid-VM

Data Management

Approximately 100 sites (150 PB of data)

Each site has CPU and storage (SE)
10-100G network





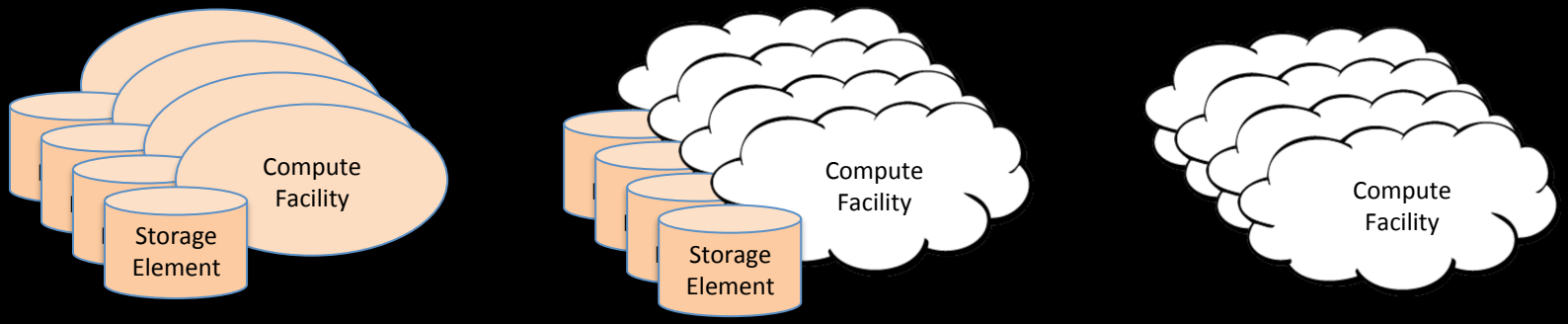
Two types of applications:

1. Analysis jobs are sent to the site with the data
2. Production jobs often have little input data
   Primary use of cloud resources

We have three types of facilities

Two of those types of facilities host the data samples

The samples at each SE are not identical



Non-cloud HEP

Cloud HEP

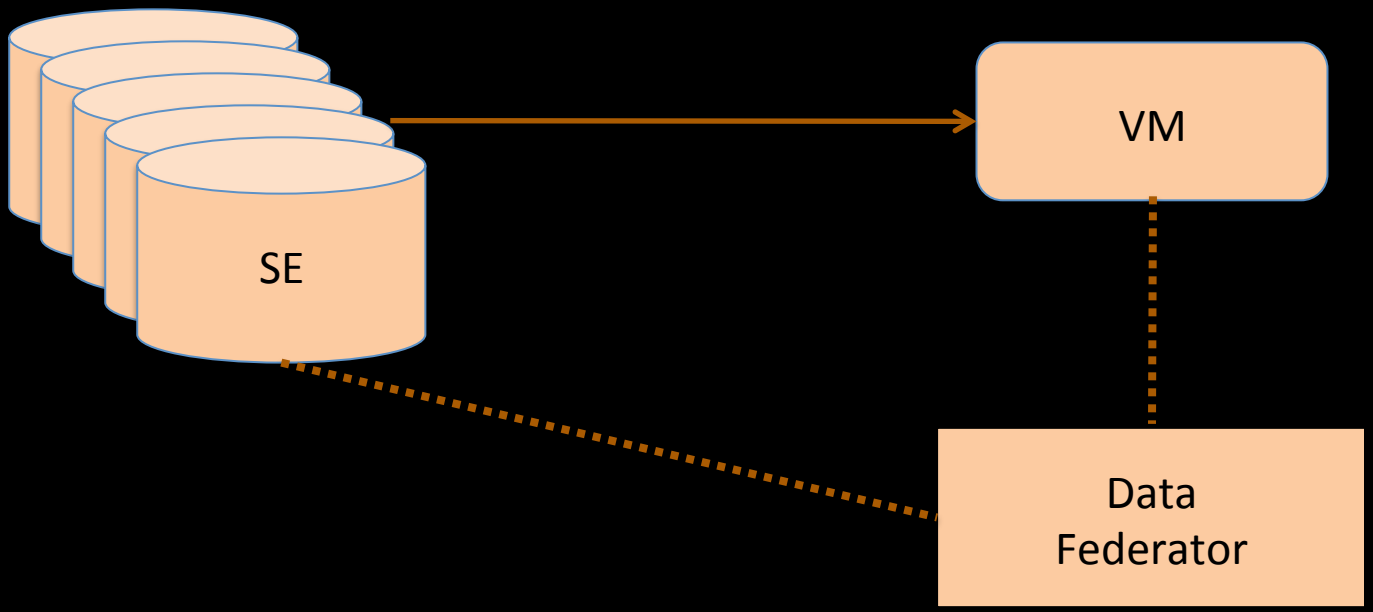
Non-HEP cloud

We want to run jobs with data on the cloud sites

The data is retrieved from the nearest SE

We need a “data federation”

HEP Data Federation



The LHC experiments operate an HEP-specific data federation – FAX

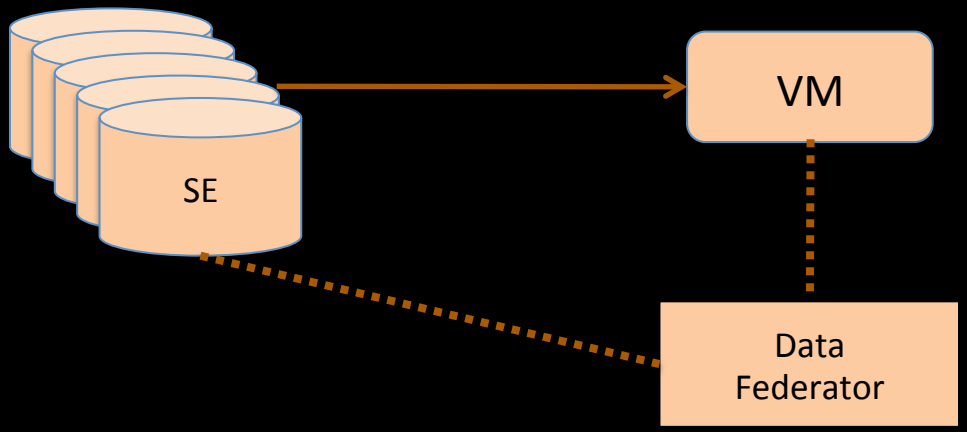
Primarily used for failover purposes on traditional computing facilities

We are testing a more general data federation

Targeted at clouds at both HEP and non-HEP sites

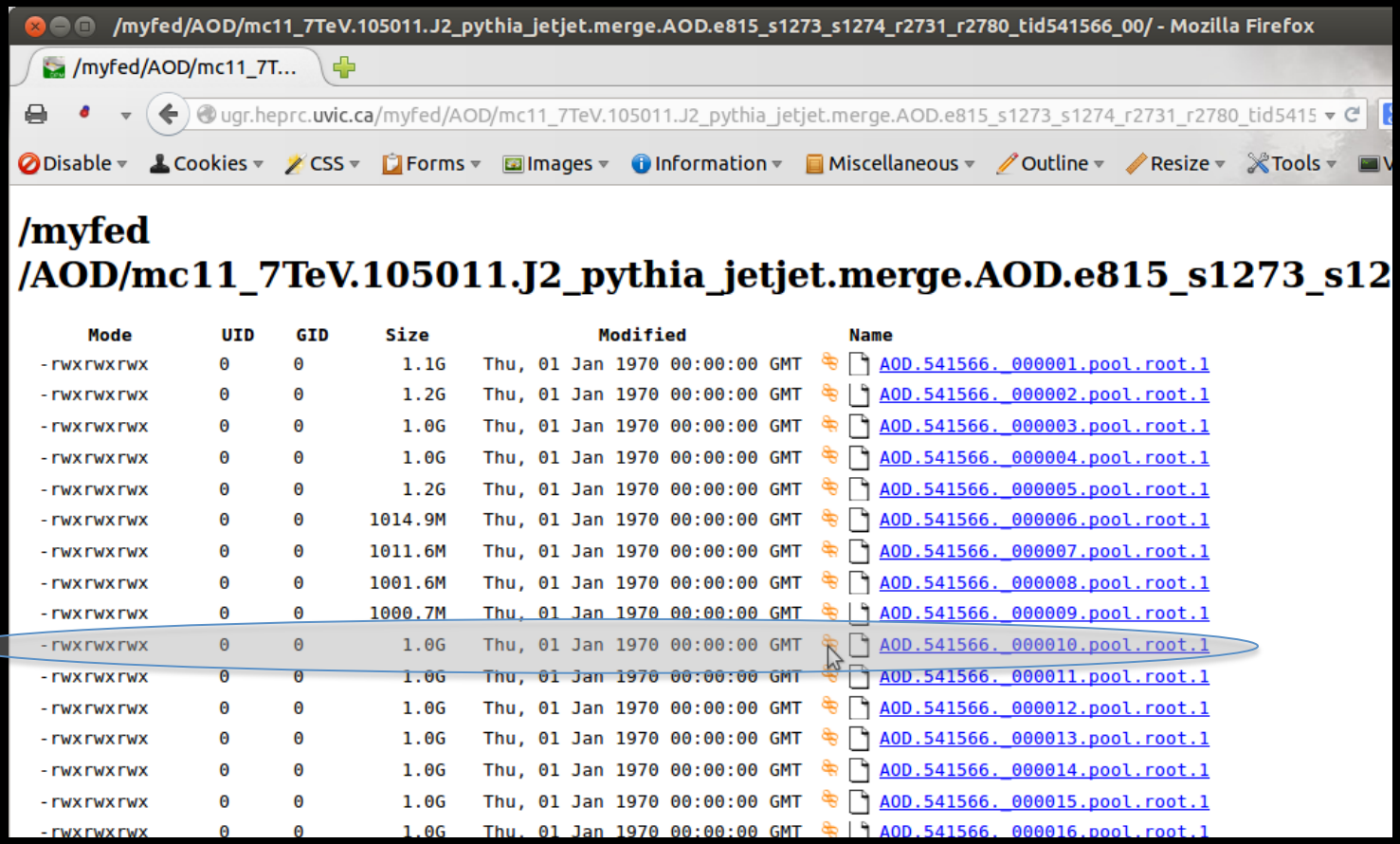
We have adopted WebDAV for file access

Open-source HTTP protocol standard

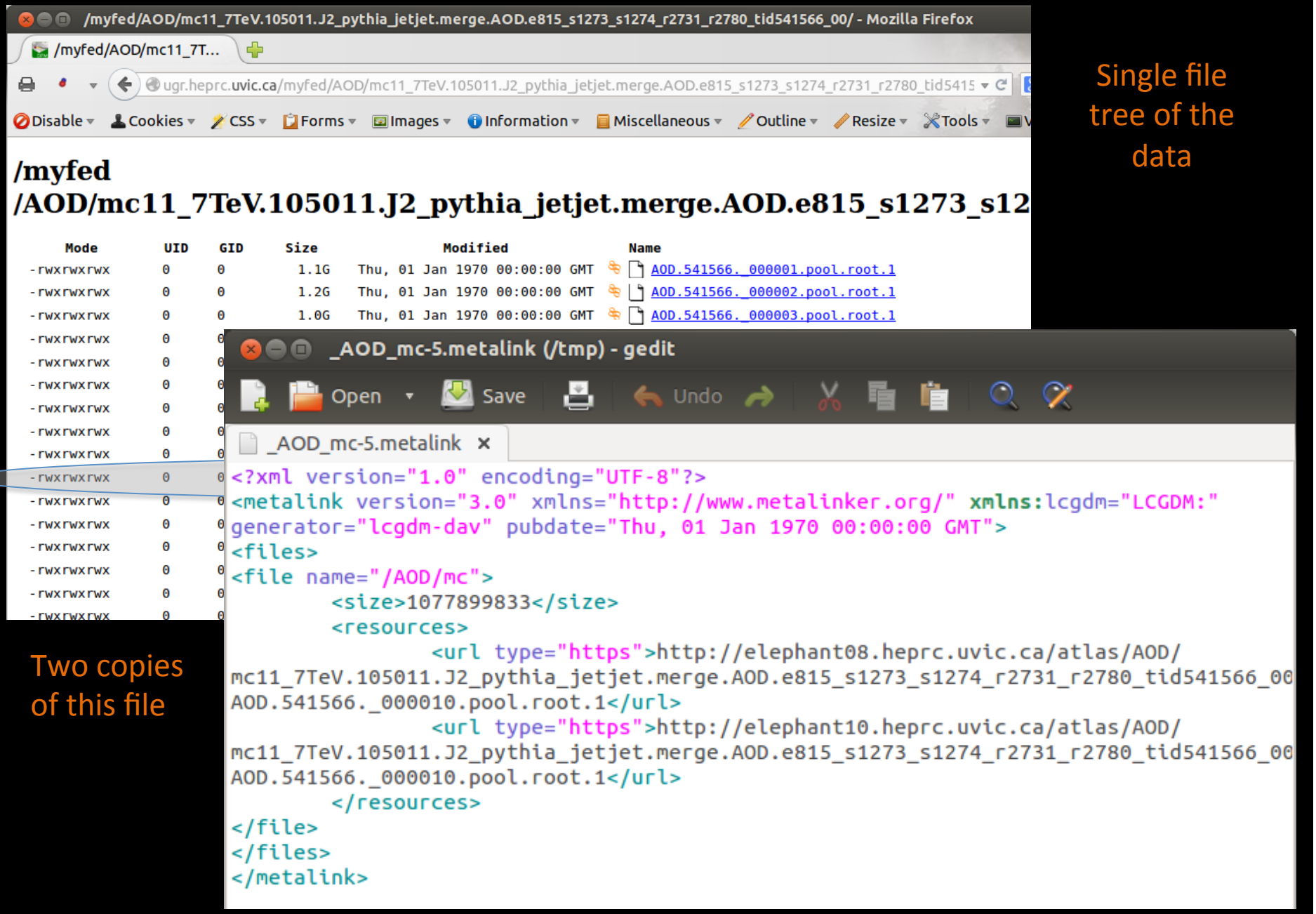


Custom (CERN) Data Federator

Presents a common view of the data file tree to the application
Directs the VM to the nearest SE with the input data
Uses GeoIP information to pick the closest site



Single file tree of the data



Single file tree of the data

Two copies of this file

Data Federation Status

Currently testing the data federation with our two test clouds

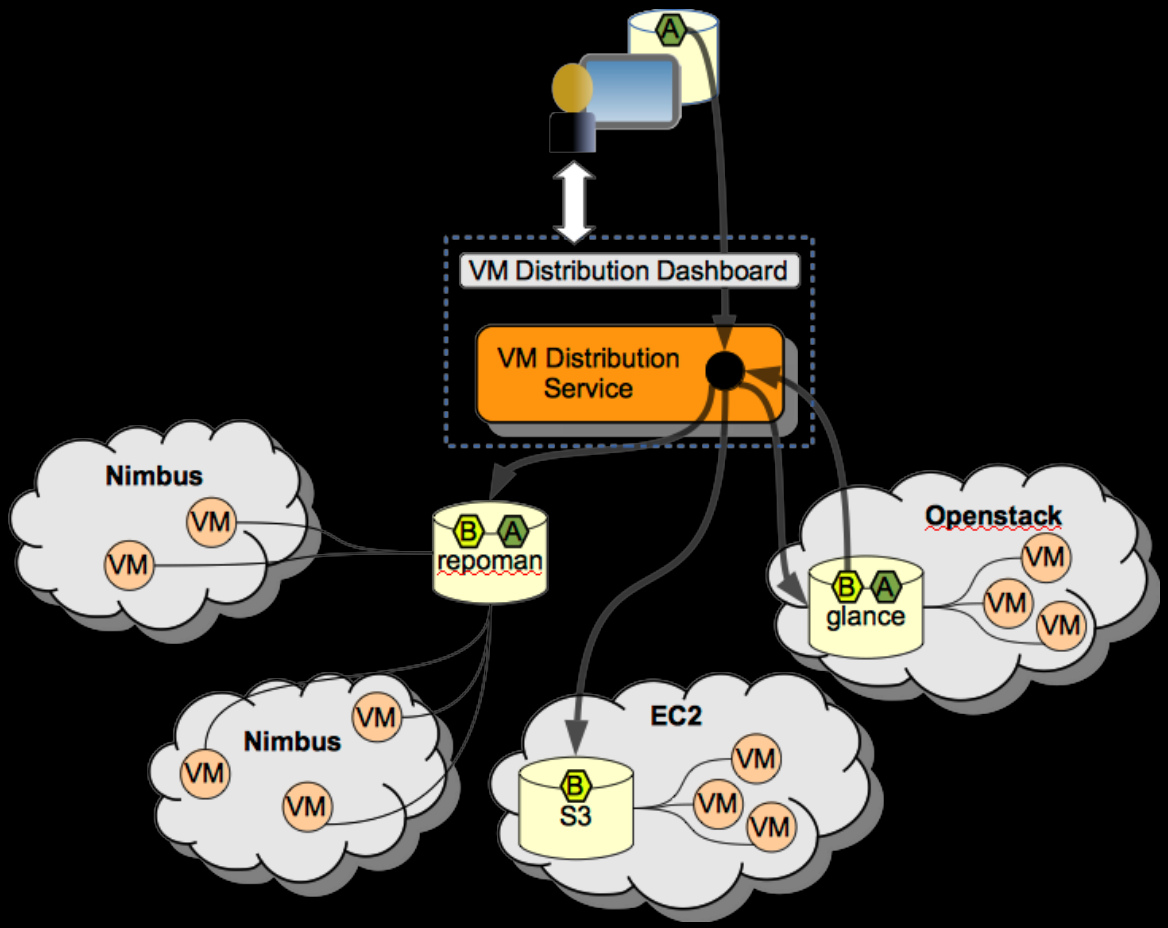
Sorted out some issues

Denial-of-Service vulnerability fixed

Disabled web crawling using “robots.txt”

Next step is to use it with the ATLAS application

Image distribution system



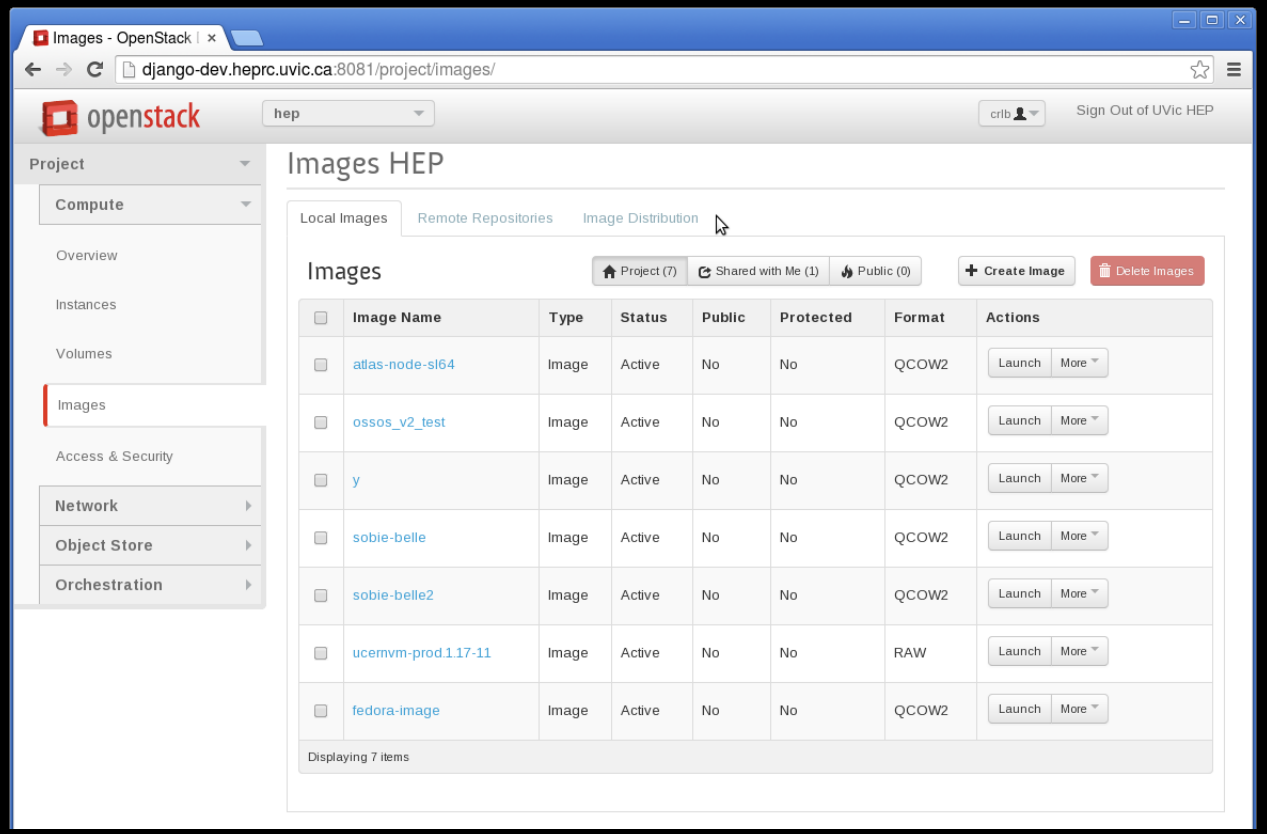
We need to manage VM images over multiple clouds

Nimbus clouds can access images from external repositories (such as our Repoman repository)

Openstack and **EC2** clouds require that we manually upload images to each cloud

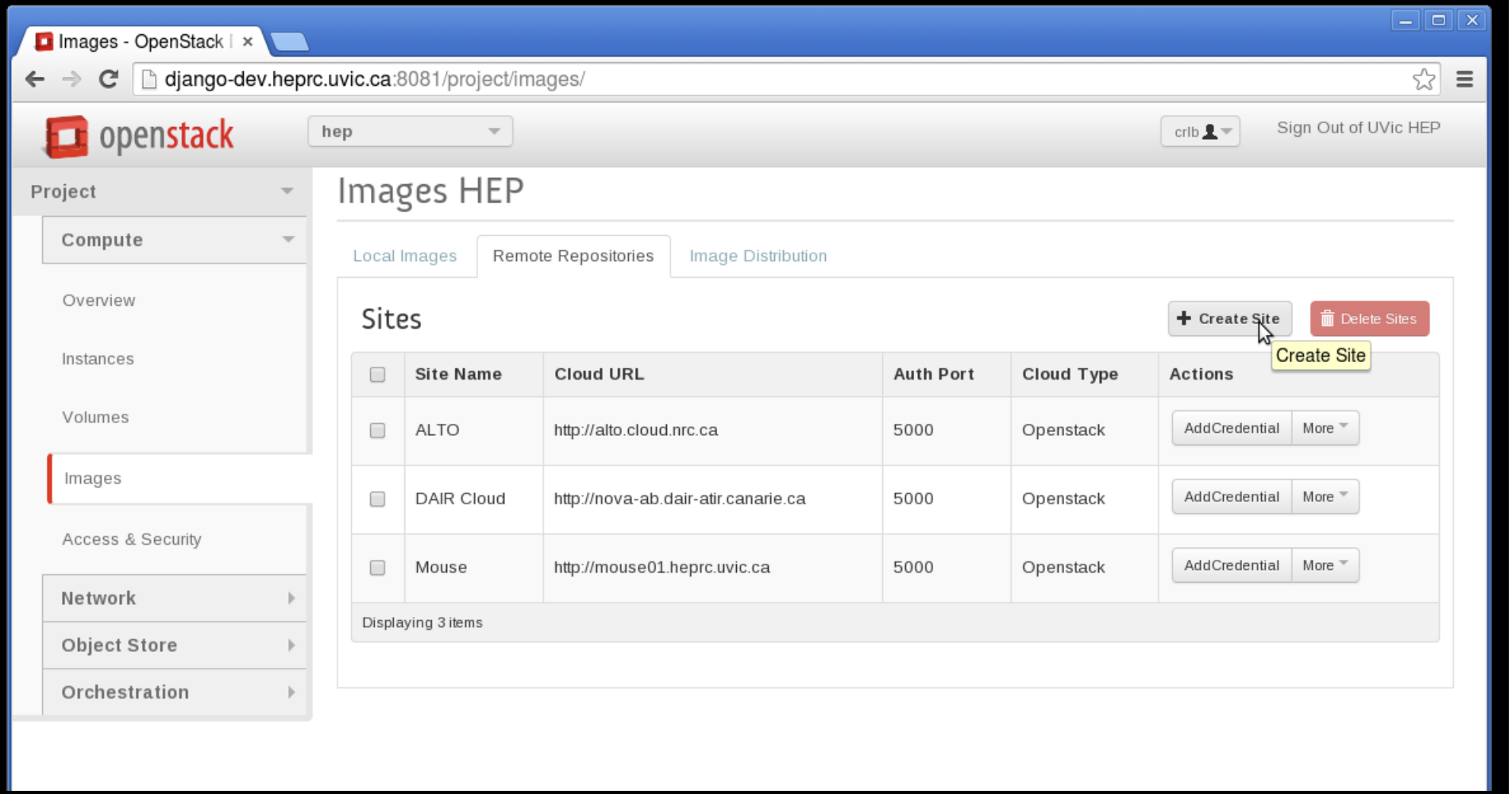
Glint

Created ***Glint*** to integrate with the individual *Glance* repositories of each OpenStack cloud

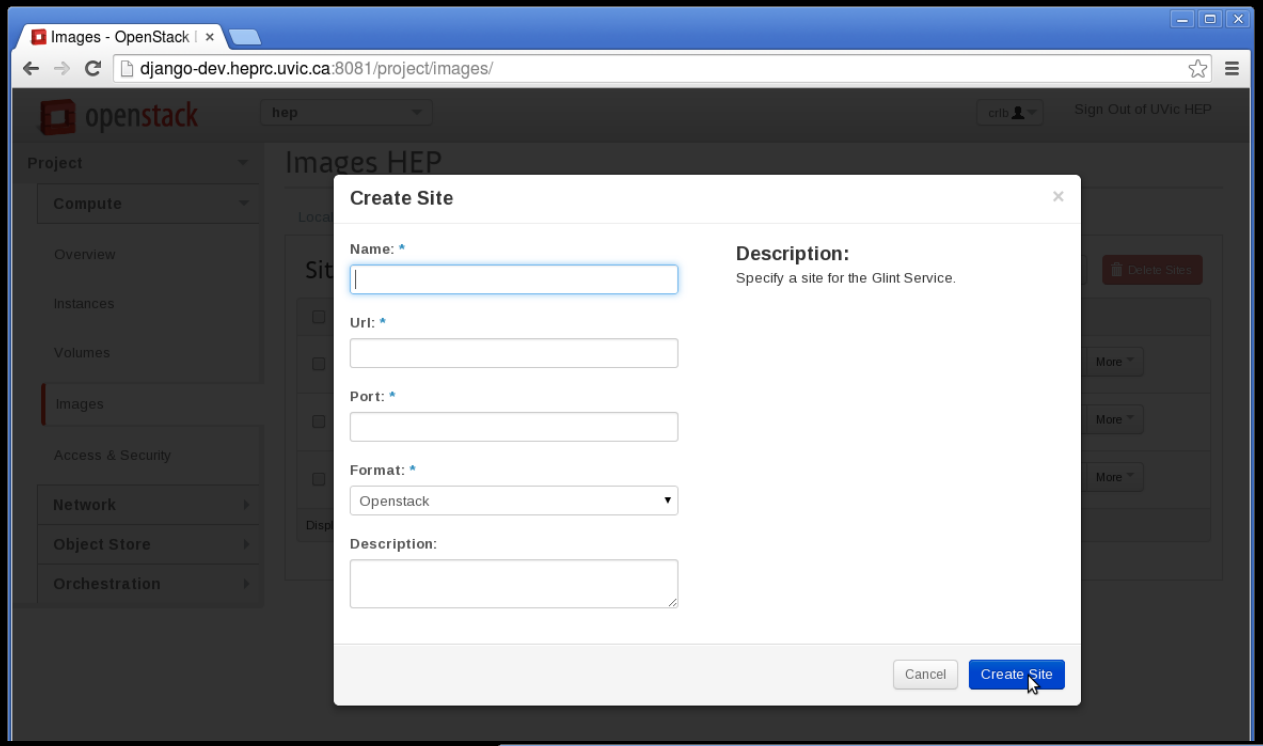


Glint is designed to be part of the OpenStack GUI

Glint manages images on multiple OpenStack clouds

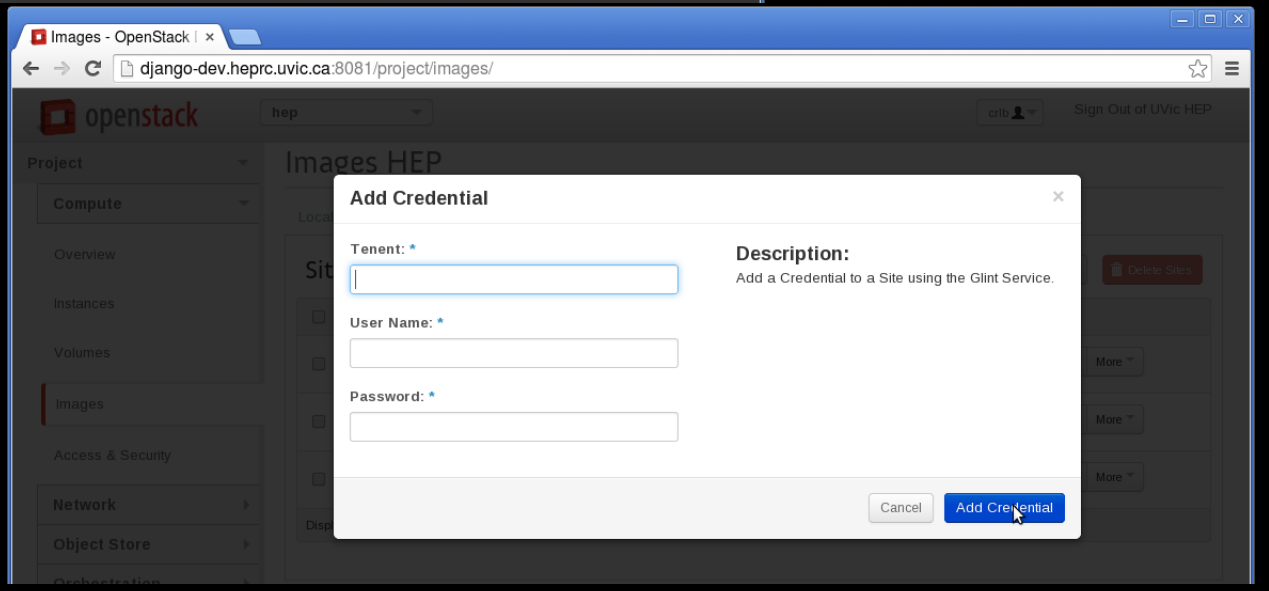


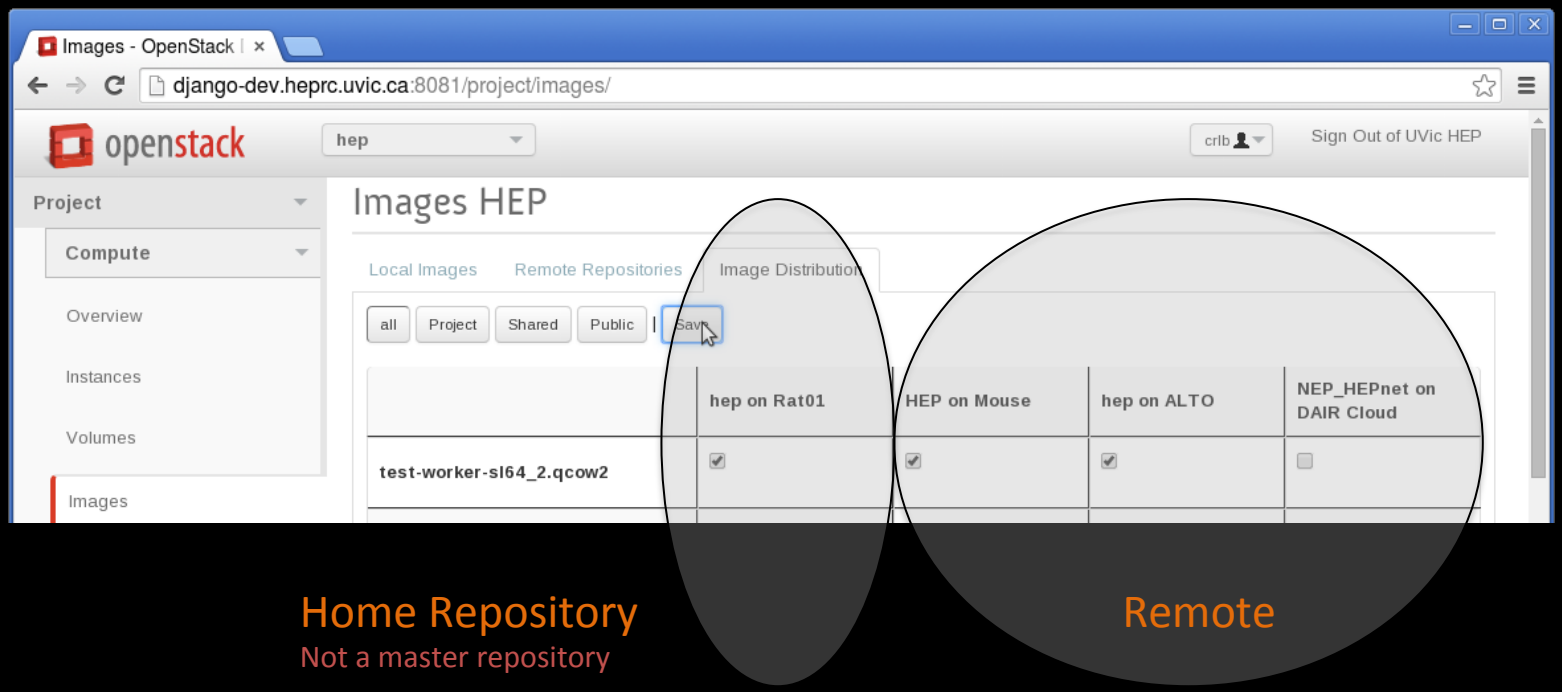
This image shows the remote repositories



And add other clouds

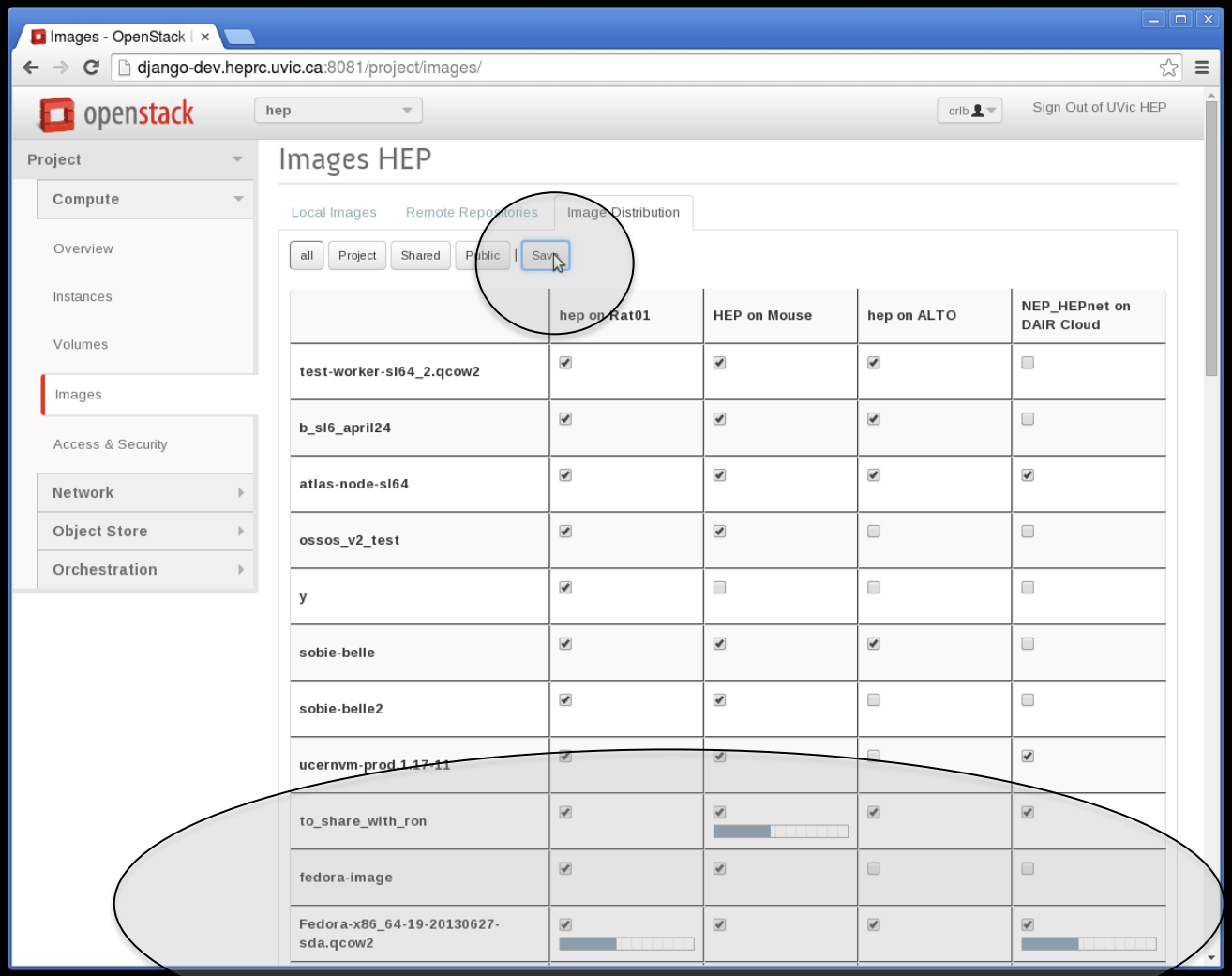
Add user credentials





The images on the Home Repository can be different than those on the Remote Repositories

Potential for different images to use the same name



Glint is under beta-test and will in production before September 2014

Integration into OpenStack is a key goal

Summary

We have been operating a distributed cloud computing system for scientific applications for many years

Used for ATLAS and BelleII HEP experiments and Canadian astronomy community

We are extending the functionality and reliability of the system by making it a context-aware distributed cloud

Utilizing distributed software caches

Accessing distributed data repositories

Managing VM images over multiple clouds

We expect the system to group in scale to 5000 cores by 2015

Providing these new services will further enhance its capabilities

Acknowledgements

Contact information: rsobie@uvic.ca
Web sites: http://rjs.phys.uvic.ca/
http://heprc.phys.uvic.ca/home/