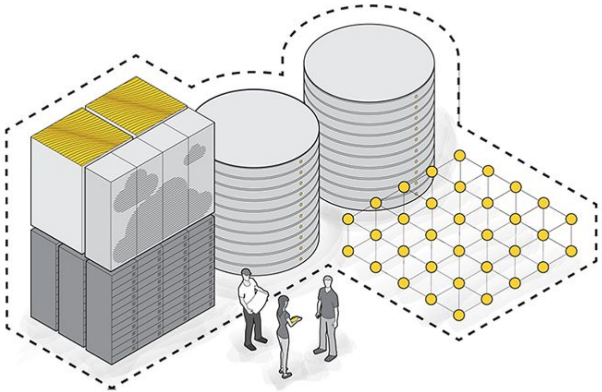
VC3

Virtual Clusters for Community Computation

DOE NGNS PI Meeting
September 27–28, 2017

Douglas Thain, University of Notre Dame
Rob Gardner, University of Chicago
John Hover, Brookhaven National Lab

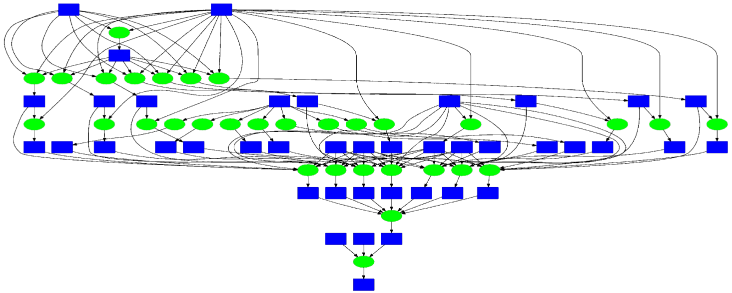


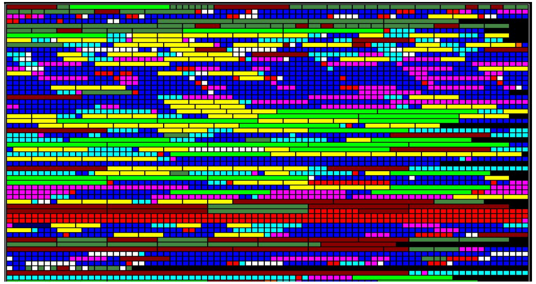
A platform for provisioning cluster frameworks over heterogenous resources for collaborative science teams

Project & People

- Joint project started in 2016 between University of Chicago, Notre Dame, and BNL
- Funded for three years by DOE Office of Advanced Scientific Computing Research (ASCR) and Next Generation Networking Services (NGNS)
- co-PIs: Rob Gardner, David Miller (UC), Douglas Thain, Kevin Lannon, Mike Hildreth, Paul Brenner (ND), and John Hover (BNL)
- Dev & Ops Team: Lincoln Bryant, Jeremy Van (UC), Ben Tovar, Kenyi Hurtado Anampa (ND), Jose Caballero (BNL)
- Testing and app on-boarding: Suchandra Thapa (UC/OSG), Ben Benedikt Riedel (UC/OSG)

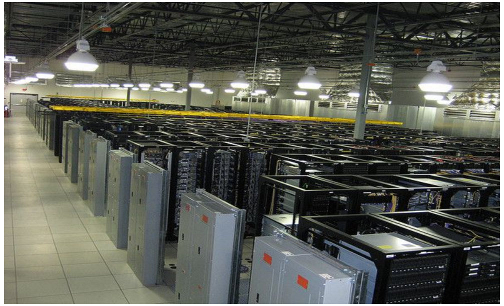
You have developed a complex workload which runs successfully at one site, perhaps your home university.





Now, you want to migrate and expand that application to national-scale infrastructure.
And allow others to easily access and run similar workloads.

Open Science Grid

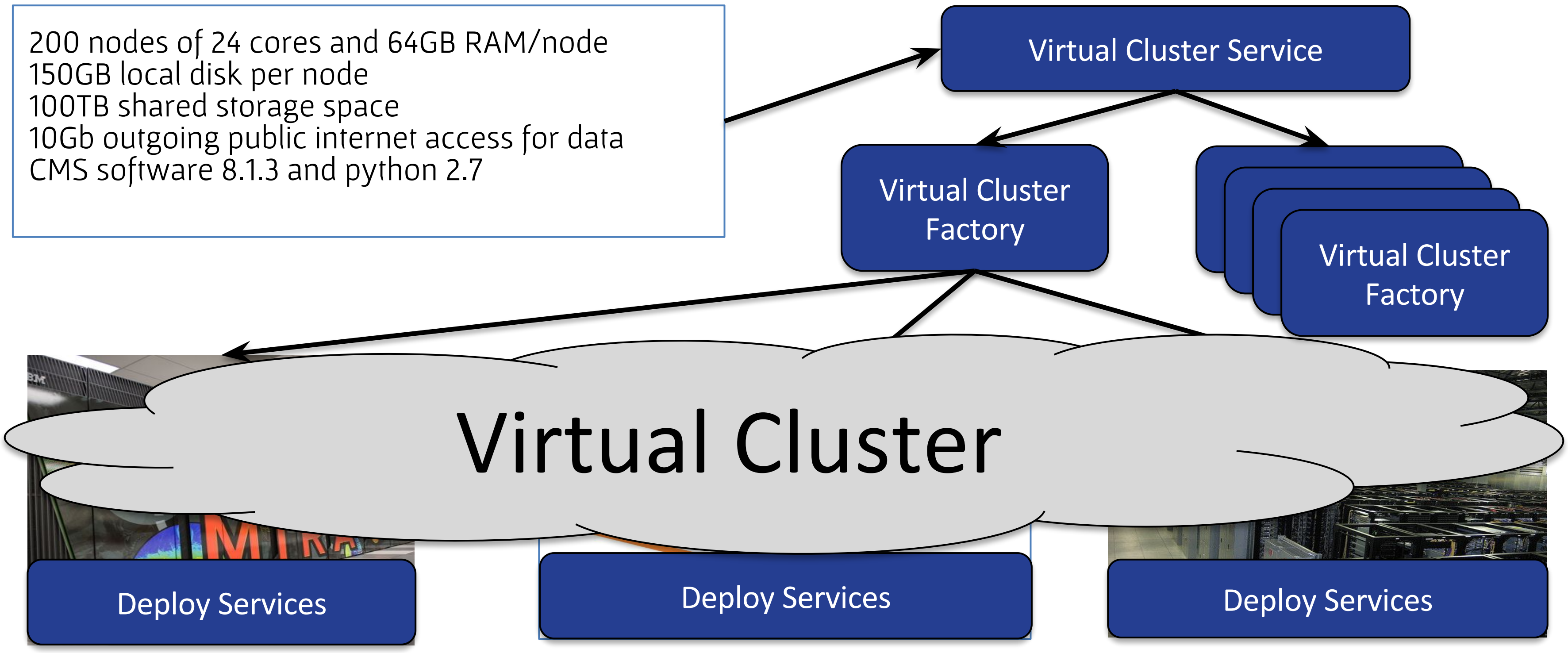


Traditional HPC Facility

Distributed HTC Facility

Commercial Cloud

Concept: Virtual Cluster



Traditional HPC Facility | Distributed HTC Facility | Commercial Cloud

What is it?

VC3 aggregates allocation-based resources, dynamically constructing homogeneous virtual clusters (middleware) as a service. Key features:

- **Automated**: Clusters are requested, built, used, and torn down by the system, driven by a user-facing web portal.
- Utilizes **dynamic infrastructure**. Factories and other central services are spawned as needed (and destroyed when finished). Static components relatively lightweight.
- **Application (middleware) agnostic**: Cluster middleware can be HTCondor, WorkQueue. Extensible, e.g. Apache Spark or Kubernetes
- **VC3 Builder** satisfies all dependencies specified in cluster definition, *as needed*.
- **User driven**: Oriented toward aggregating individual or small group allocations, e.g. campus clusters, academic clouds, university HPCs for **federated teams**

What it's not

- VC3 is **not** a workload management system. It doesn't run **jobs**, it provisions a customized cluster for your chosen middleware.
- It isn't oriented toward creating large scale, global clusters
- Clusters are short-lived, for individuals or small groups, purpose-built for a workflow/task.
- Doesn't currently handle data. Globus integration is forseen.
- Not expected to be picked up and deployed by a VO; it will be a service. But all the code is open, packages and dependencies are published, so in theory someone *could.*
- Not developed from scratch. Integrates existing technologies and combines them into a fully automated, user-oriented service.

Representative Use Case

- A university researcher has a small group cluster, a campus batch cluster, and access to OSG.
- A colleague from another institution has an allocation on an academic cloud and an Amazon credit.
- They can create a VC3 ***project*** and each assign their resources (or a portion) to it.
- For a particular workflow, they define a virtual cluster, e.g.
  - 50 nodes of 2GB RAM, 25GB disk.
  - RHEL6
  - HTCondor cluster (managed with dynamic CM, schedd).
  - CVMFS
  - GCC 4.3 (an older version)
  - Other arbitrary package dependencies.

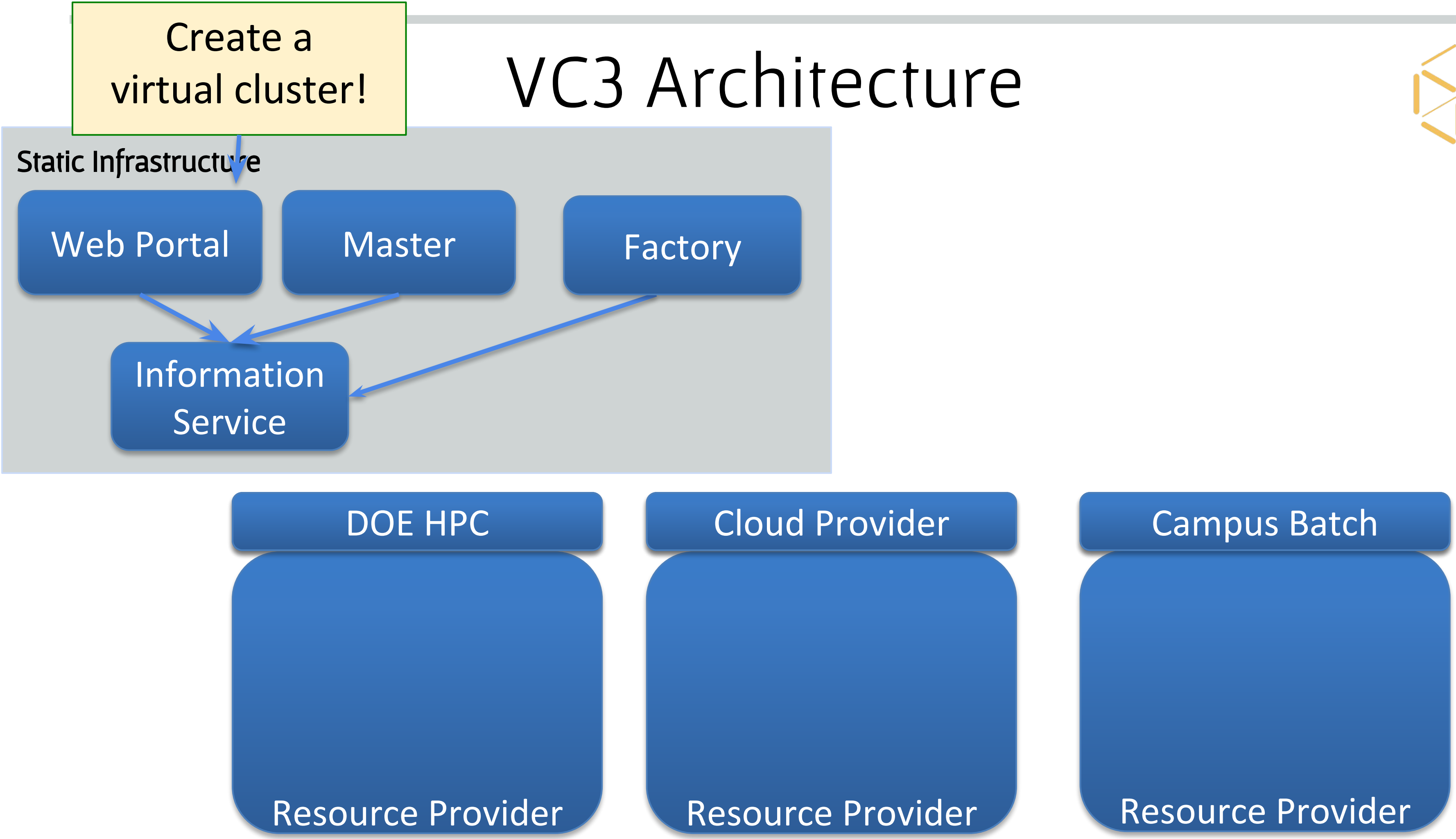
And request it, specifying a usage policy.

Representative Use Case (cont.)

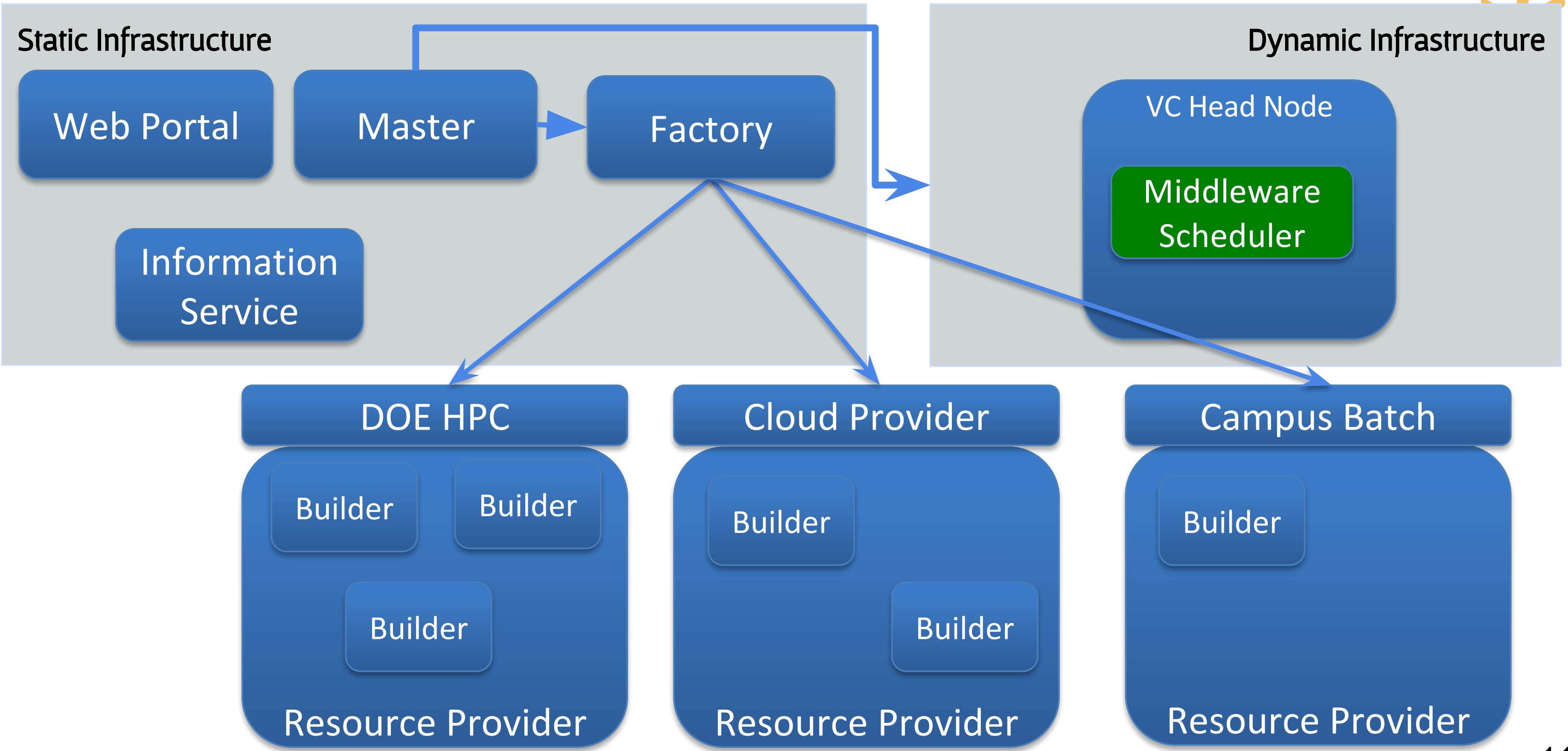
The VC3 system then...

- Sorts out which resources *can* service the request.
- Spawns a dynamic master to manage this single cluster.
- Controller launches/configures a provisioning factory and central middleware infrastructure (e.g. the vc3-factory + HTCondor central manager and schedd.)
- The factory then submits vc3-builders to resources, re-submitting as needed.
- vc3-builder, for each worker, satisfies all dependencies *however needed on that node.*
  - If GCC 4.3 is not present, installs it.
  - If fuse not present for CVMFS, sets up Parrot.
- User then either loads workflow and data on managed infrastructure, or triggers remote submission into it.
- User triggers cluster teardown when done, (staging out data if not handled out of band).
- When all workers are gone, central infrastructure shut down. Request completed.

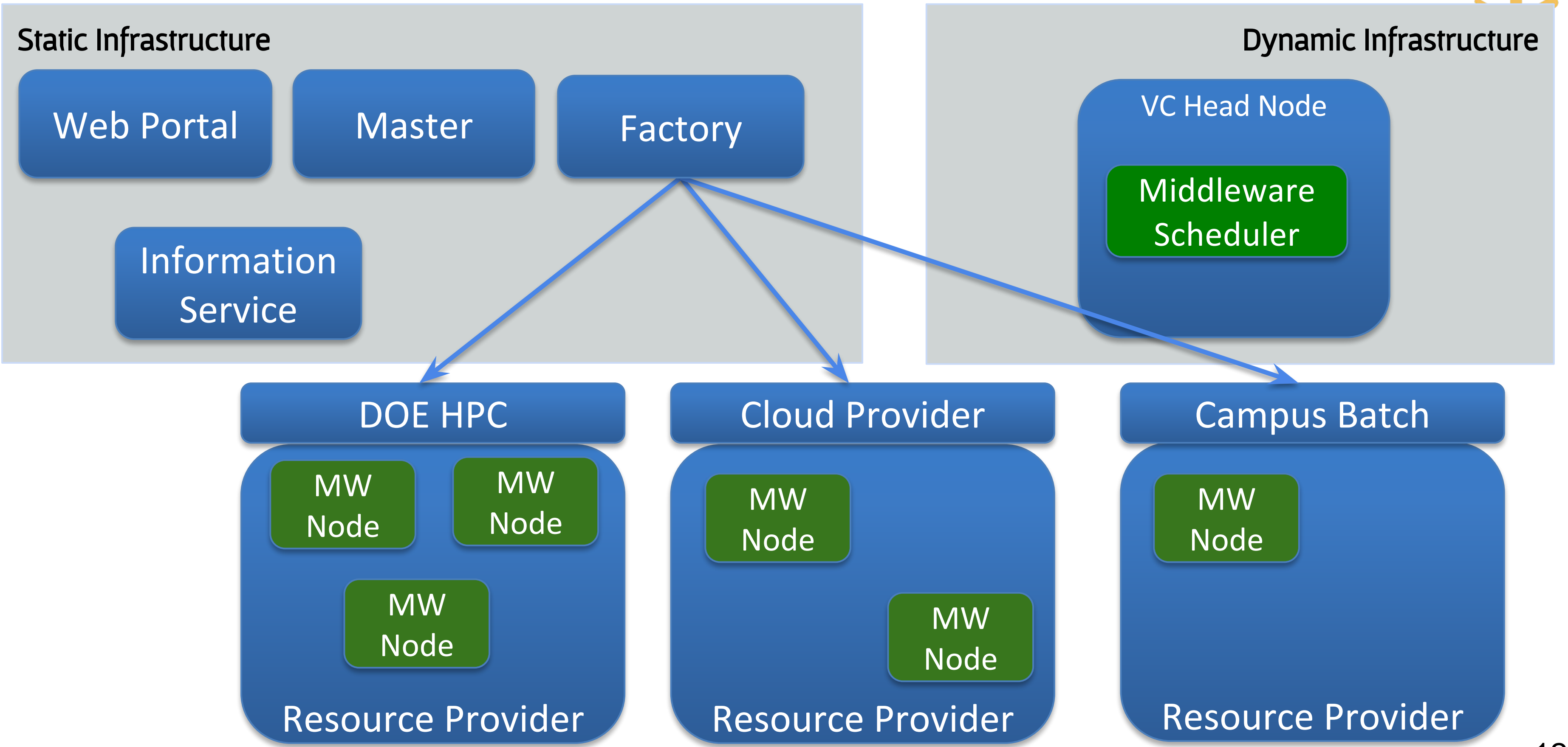
VC3 Architecture



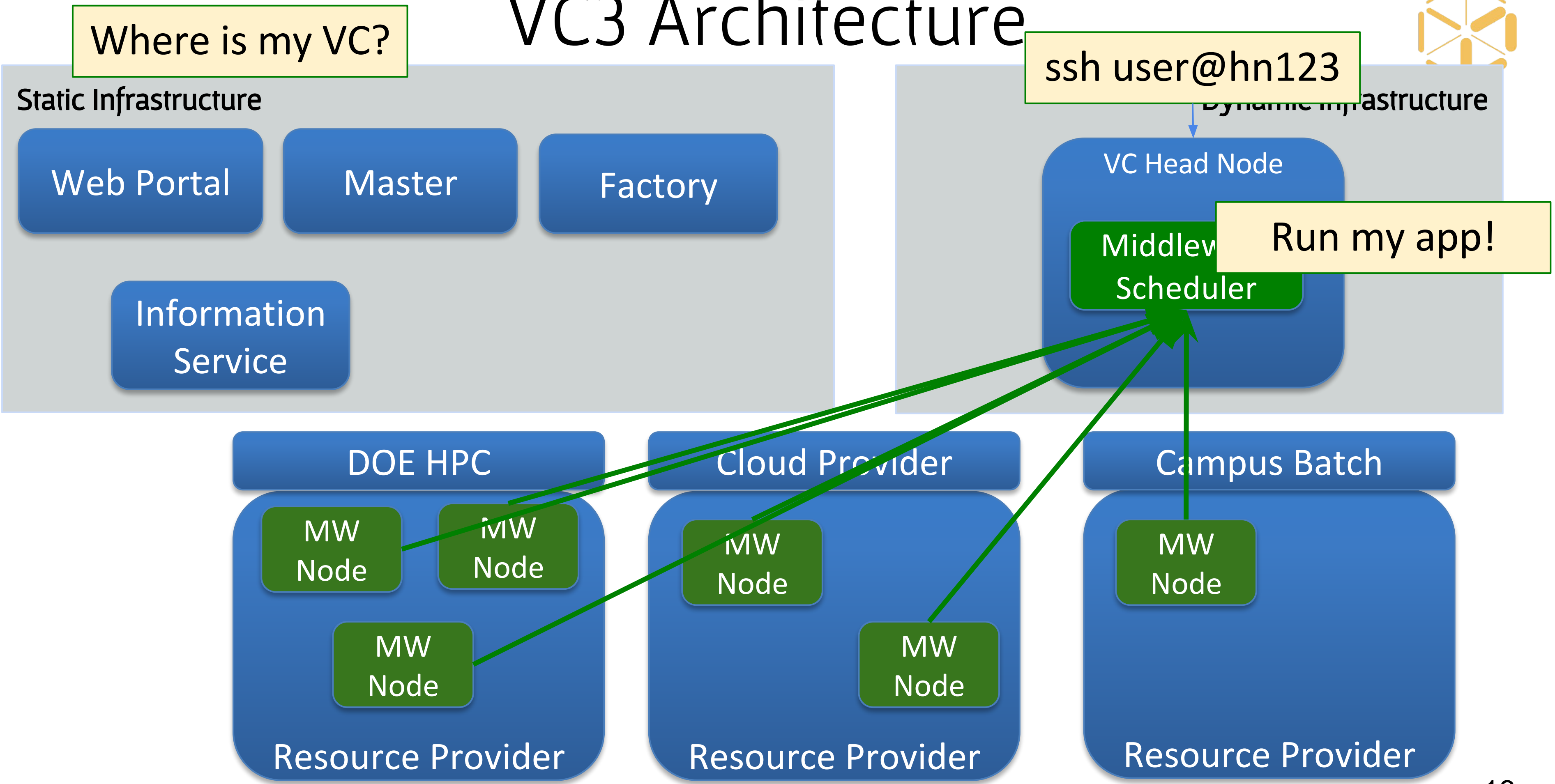
VC3 Architecture



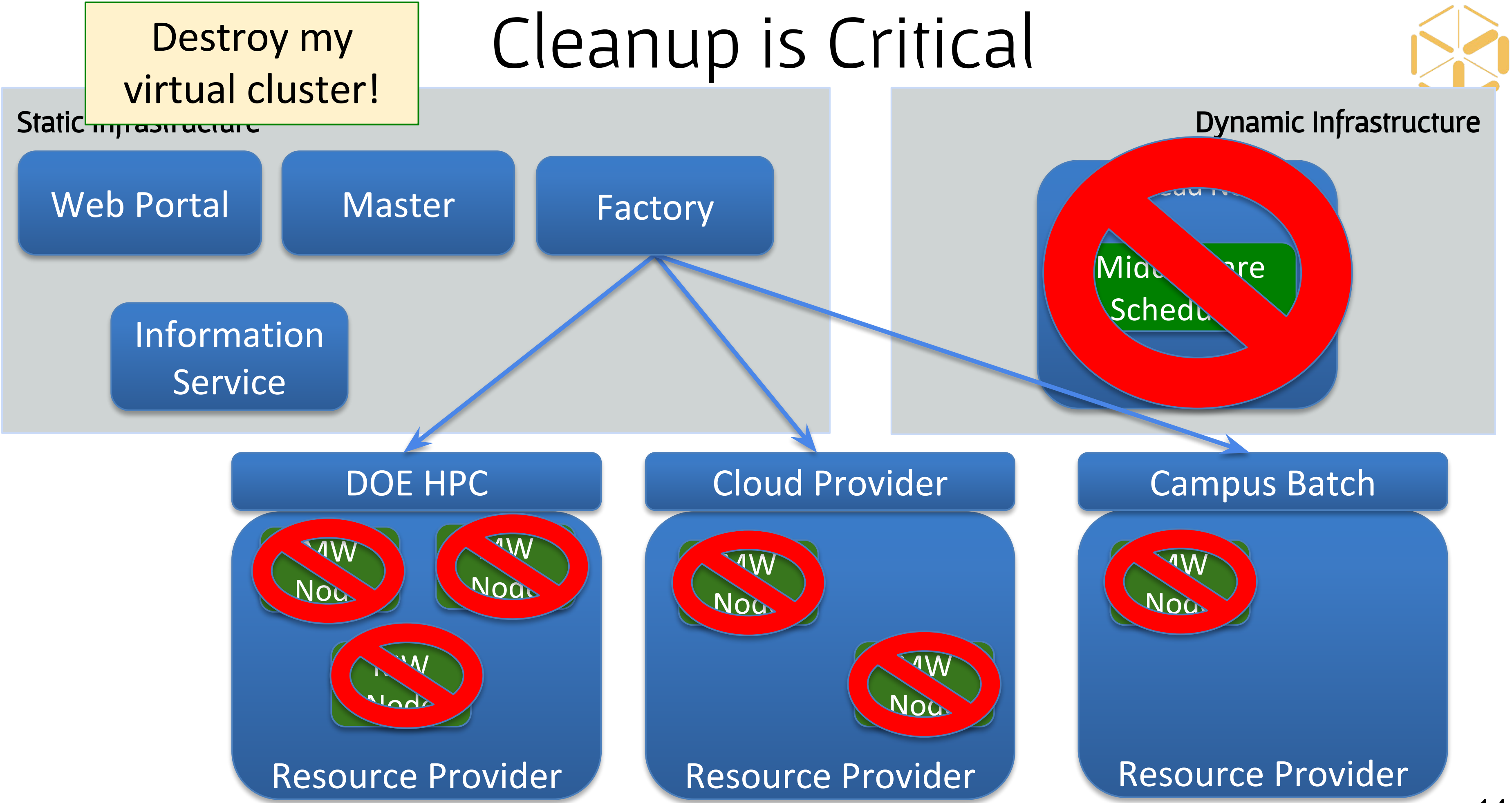
VC3 Architecture



VC3 Architecture



Cleanup is Critical



Everything Cleaned Up

Static Infrastructure

Web Portal

Master

Factory

Information Service

DOE HPC

Resource Provider

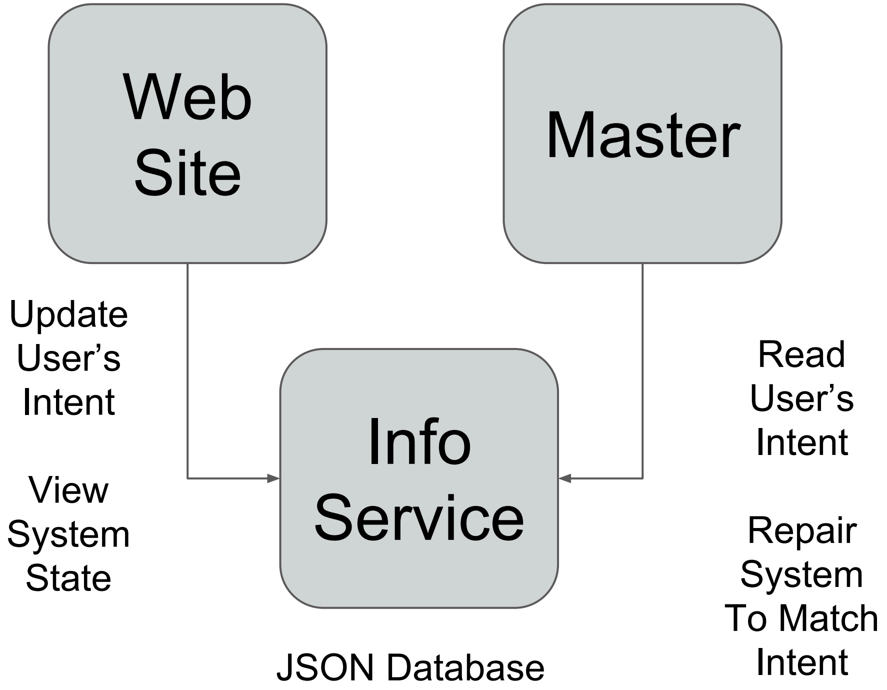
Cloud Provider

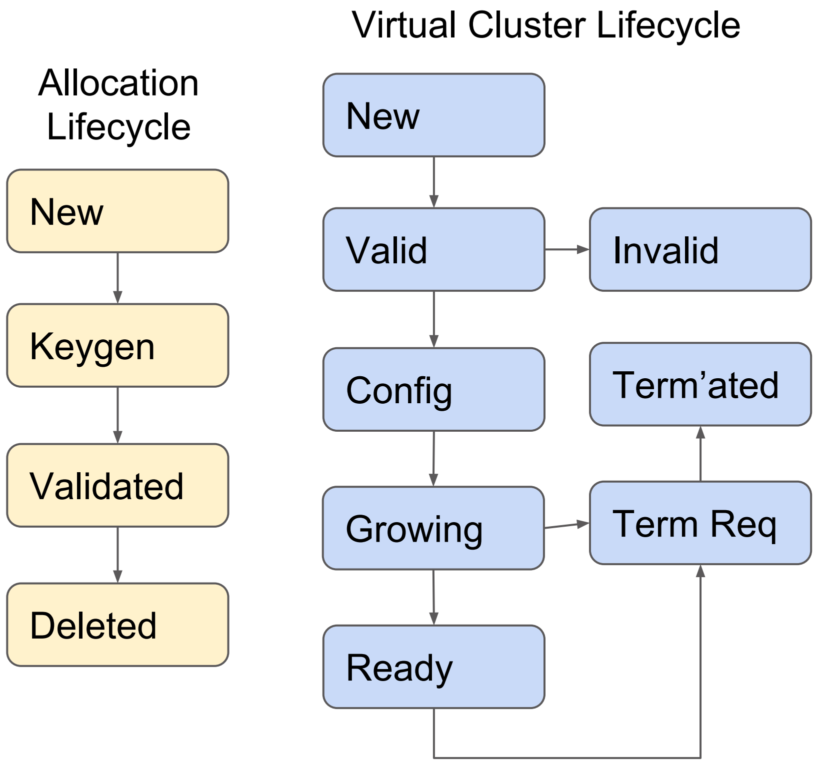
Resource Provider

Campus Batch

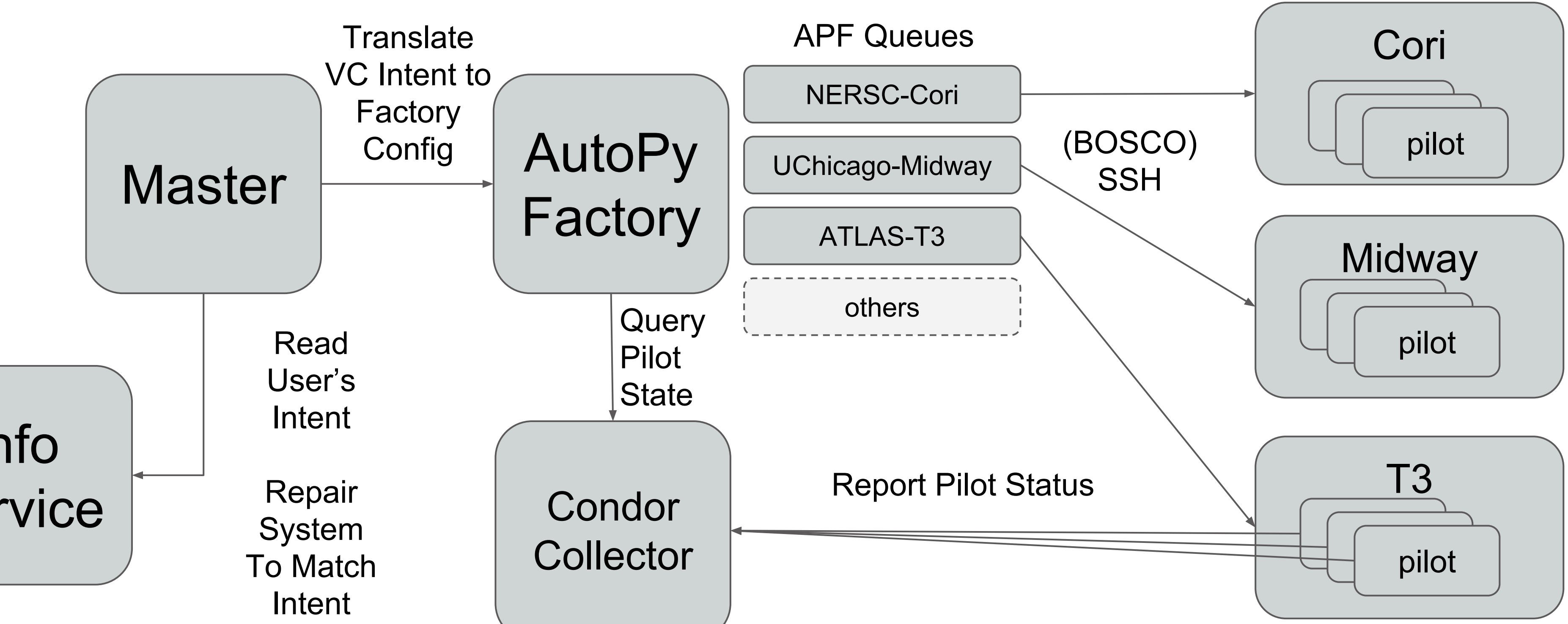
Resource Provider

Details: Cluster Lifecycle



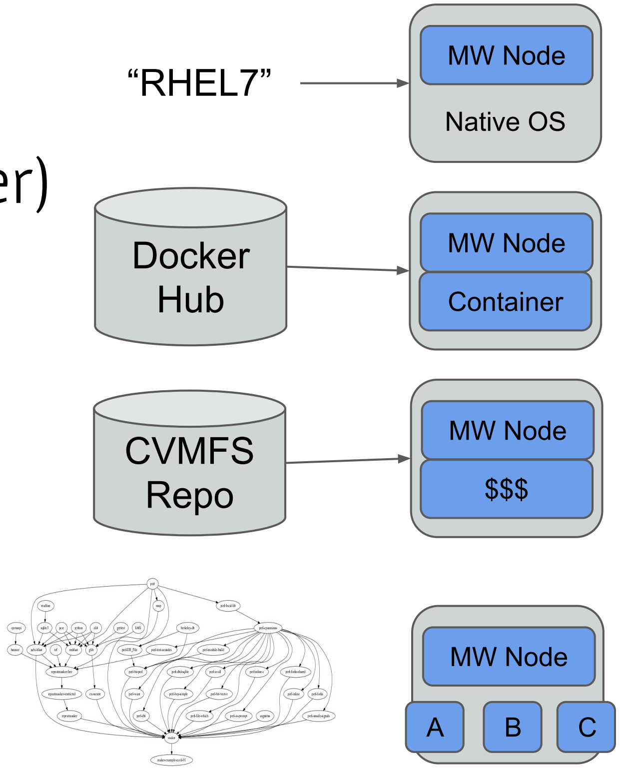


Details: Cluster Provisioning



Details: Software Environments

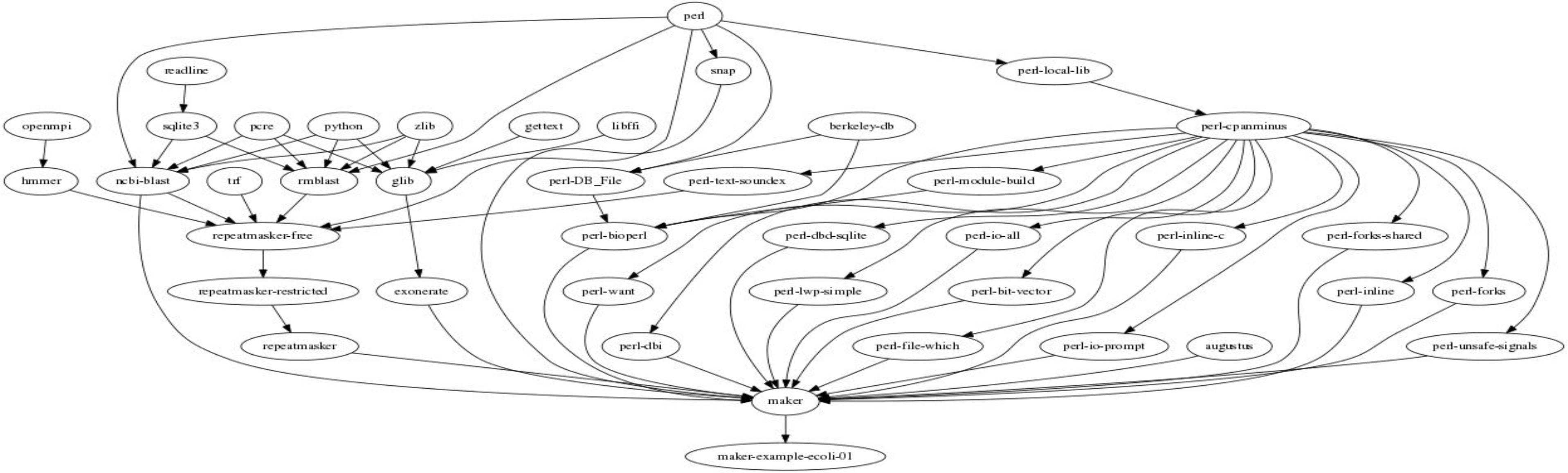
- Native Availability
  - Specify desired OS from list.
- Containers (Docker/Singularity/Shifter)
  - Specify image to pull from Docker Hub.
- On-Demand Deployment (CVMFS)
  - Specify CVMFS repo, system mounts it.
  - CVMFS via FUSE (kernel) or Parrot (user)
- Build on Site
  - Specify list of software packages needed.
  - VC3-Builder downloads and installs.



The MAKER Genomics Pipeline

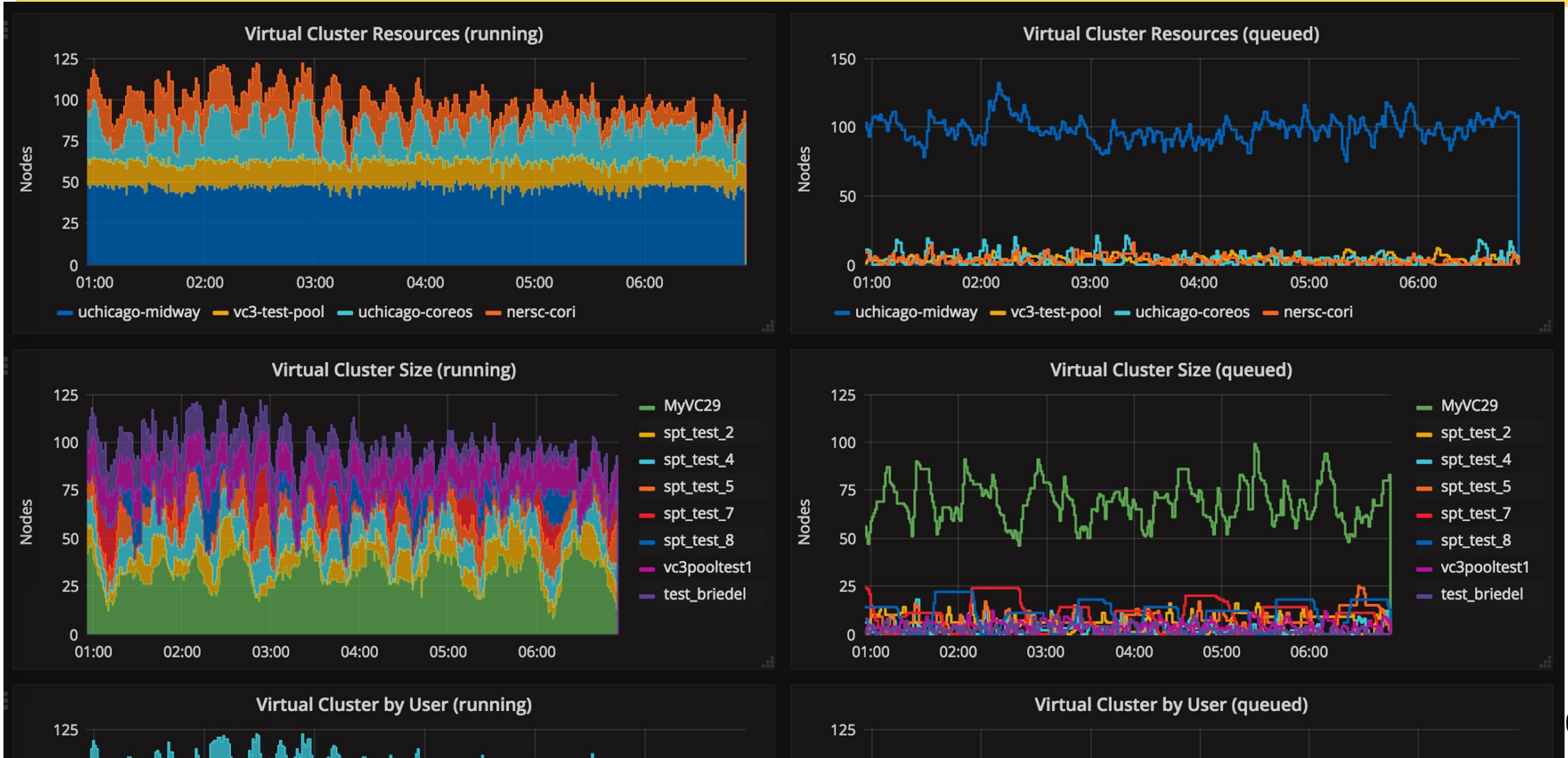
http://www.yandell-lab.org/software/maker.html

vc3-pilot **–require maker** maker -BIO



Custom docker container in Jetstream took weeks to install pieces by hand.
Converted to vc3-builder, successfully ported to Stampede in a single automated install.

Details: System Monitoring

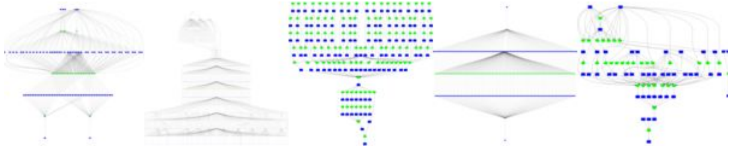


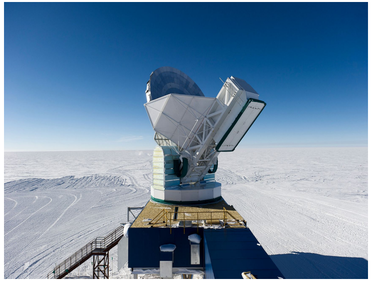
Current System Status

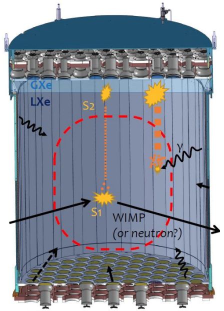
- Basic functionality is up and running and used daily by project members.
  - Users can create projects, define cluster templates, attach allocations, create virtual clusters, monitor status, and tear them down.
  - Scale: O(100) VCs running concurrently
- Resources Connected:
  - NERSC – Cori (SLURM + Docker)
  - UChicago Midway (SLURM), ATLAS T3 (HTCondor + CVMFS)
  - OSG Testbed (HTCondor)
  - Syracuse (HTCondor), Stampede2 (PBS), Notre Dame (SGE) – Testing Stages
- Middleware Selectable: HTCondor or Work Queue
- Authentication Mechanisms:
  - Globus Auth for User –> Portal, SSH Key for Portal –> Resource

Applications Working Under VC3

- Various Bioinformatics Workflows
  - Makeflow + HTCondor + BWA, Shrimp, BLAST…
- Lobster CMS Data Analysis
  - Work Queue + Builder + CVMFS
- South Pole Telescope (SPT-3G)
  - HTCondor Jobs + Docker + CVMFS
- XENON1T
  - Pegasus + HTCondor + CVMFS
- MAKER Bioinformatics Pipeline
  - Work Queue + Builder
- IceCube Simulation Framework







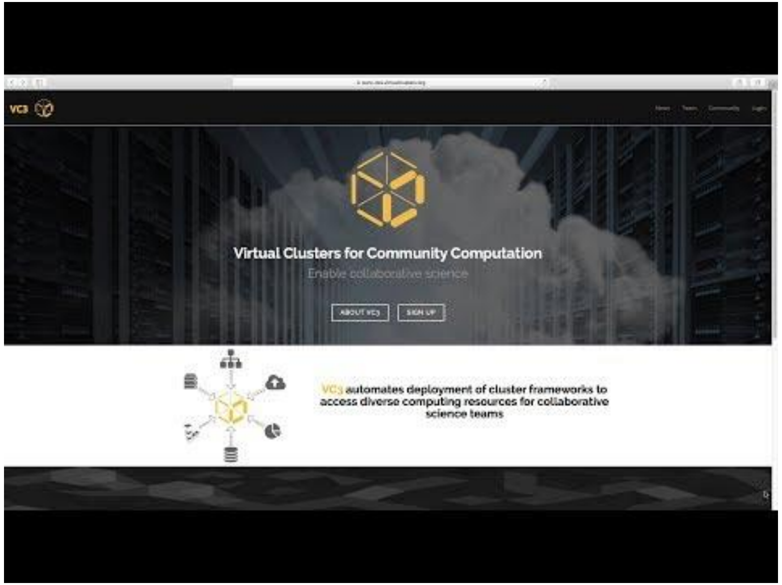
VC3 Live Demo

Connect allocations, create virtual clusters with both HTCondor and Work Queue to work workflows with Pegasus and Makeflow:

(Live Demo!)

VC3 Recorded Demo

Connect allocations, create virtual clusters with both HTCondor and Work Queue to work workflows with Pegasus and Makeflow:



Challenges and Considerations

- Software: Diversity or Consistency?
  - What do users really want? Global container names vs local site installs vs CVMFS mount vs on-demand installs? No method applies globally.
- Authentication Complexity
  - 2FA: pass to user, or argue that site is a “factor”?
  - SSH Keys: auth file / auth db / Kerberos
  - Alternate approach: Provide pull-mode “recipe” for user to invoke.
- Punching Through Layers
  - Ex: NERSC Shift Docker name goes in the job script header.
  - Must modify Condor-G BOSCO, APF, resource description, web portal, ...

Challenges and Considerations (2)

- Concurrency Management at Master
  - Basic idea is simple, but corner cases are challenging.
  - Too many ssh connections: tarpitted!
  - Remote systems become less responsive as queues get longer.
  - Small scales: event-driven; large scales: periodic bulk behavior.
- Capturing Failure Modes
  - Knowing, detecting, reporting, reacting. "Unknown unknowns"
- Dimensions of Scalability / Performance
  - # of nodes in virtual cluster is interesting but not the main concern!
  - # of concurrent virtual clusters
  - # of concurrent allocations usable by one cluster
  - Overhead to setup / tear down a virtual cluster

Plans Going Ahead

- Completing the VC3 Vision:
  - Fully dynamic deployment of cluster head nodes.
  - Select (or recommend) sites based on requirements.
  - Parameterize software environment from interface.
- Expanding Coverage:
  - Sites: Campus Clusters + DOE Facilities
  - Middleware (Spark)   Applications (LHC, HEP, Bio)
- Serving Users:
  - Closed Beta (late 2017), Open Beta (2018)

Collaborators and Connections

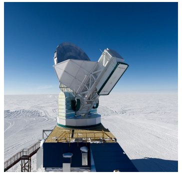
NSF S2I2 Software Infrastructure

U.S. DEPARTMENT OF ENERGY | Office of Science

HTCondor High Throughput Computing

CILogon

Open Science Grid



CMS Compact Muon Solenoid

Xe XENON Dark Matter Project

Pegasus 4.8.0

Makeflow

globus

CernVM File system

ATLAS EXPERIMENT

ci connect

Flask web development, one drop at a time

SGCI Science Gateways Community Institute

openstack.

AutoPyFactory

Grafana

VC3

Virtual Clusters for Community Computation

UNIVERSITY OF NOTRE DAME

THE UNIVERSITY OF CHICAGO

BROOKHAVEN NATIONAL LABORATORY

http://www.virtualclusters.org

https://github.com/vc3-project