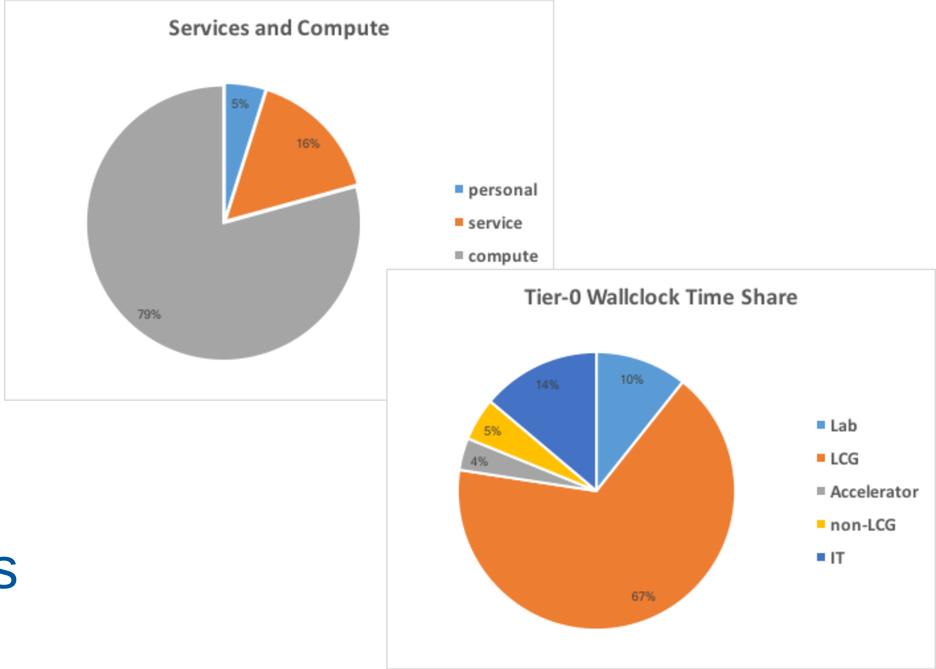
CERN

CERN Batch in the HNSciCloud

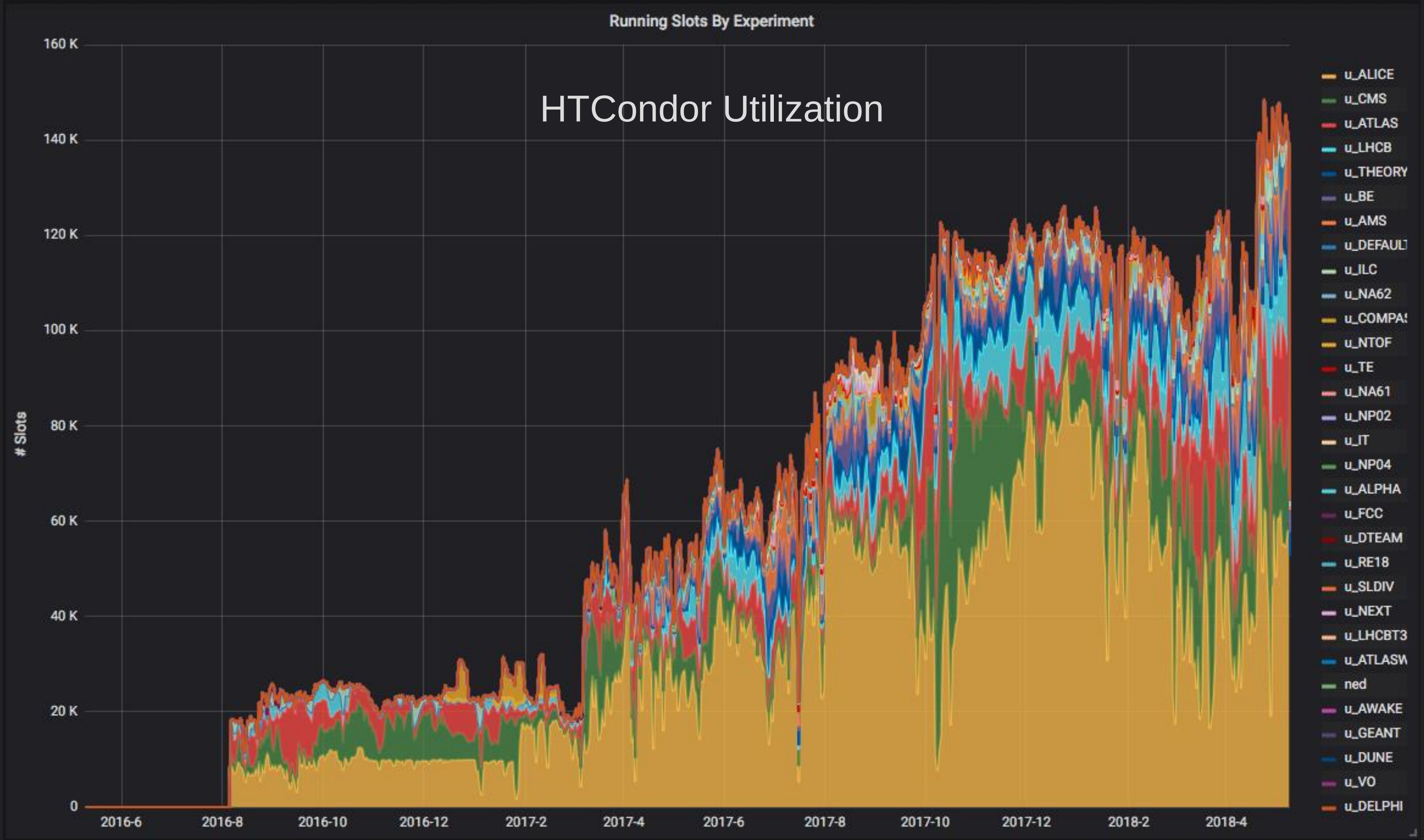
Ben Jones IT-CM

CERN Batch Service

- 230k cores of compute provided via HTCondor (or LSF) for LHC Grid and local users
- Large increase in capacity over last couple of years to support Run 2
- Batch service has been able to run on public cloud and other opportunistic resources – in particular grid workflows



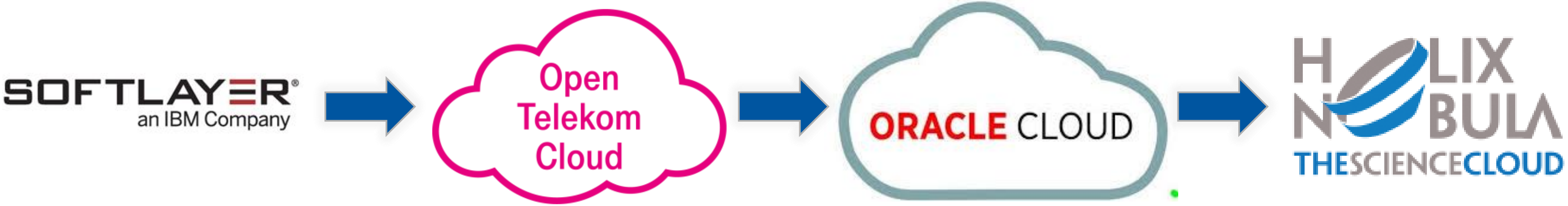
HTCondor Utilization



Grid vs Local

	Grid	Local
Authentication	X509 Proxy	Kerberos
Submitters	LHC experiments, COMPASS, NA62, ILC, DUNE…	Local users of experiments, Beams, Theorists, AMS, ATLAS Tier-0
Submission method	Submission frameworks: GlideinWMS, Dirac, PanDA, AliEn	From condor_submit by hand, to complicated DAGs, to Tier-0 submit frameworks.
Storage	Grid protocols. SRM, XRootD…	AFS, EOS

XBatch Cloud experience



2016

“XBatch” integrate cloud resources into standard pool with standard tools

2017

Use Docker to abstract external cloud platform

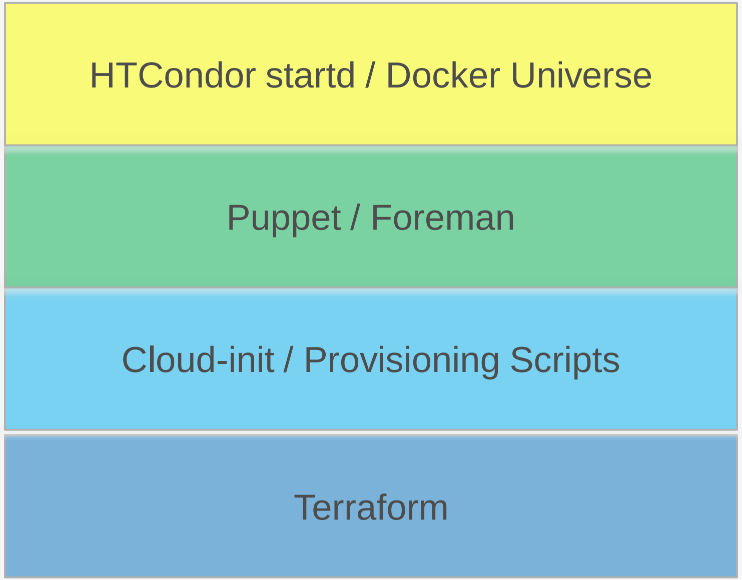
2018

Batch on Cloud resources

- Grid jobs better candidate to run in cloud as already designed to be location agnostic, with sophisticated job management & monitoring
- Use same provisioning & orchestration for public cloud and local cloud, where possible
- We generally have flat capacity & more jobs than resources
- The machine / container running job the job lives longer than the job
- Limited infrastructure required in cloud (proxies…)

xBatch Stack

Job of each layer is just to bootstrap the next



Application — HTCondor startd / Docker Universe

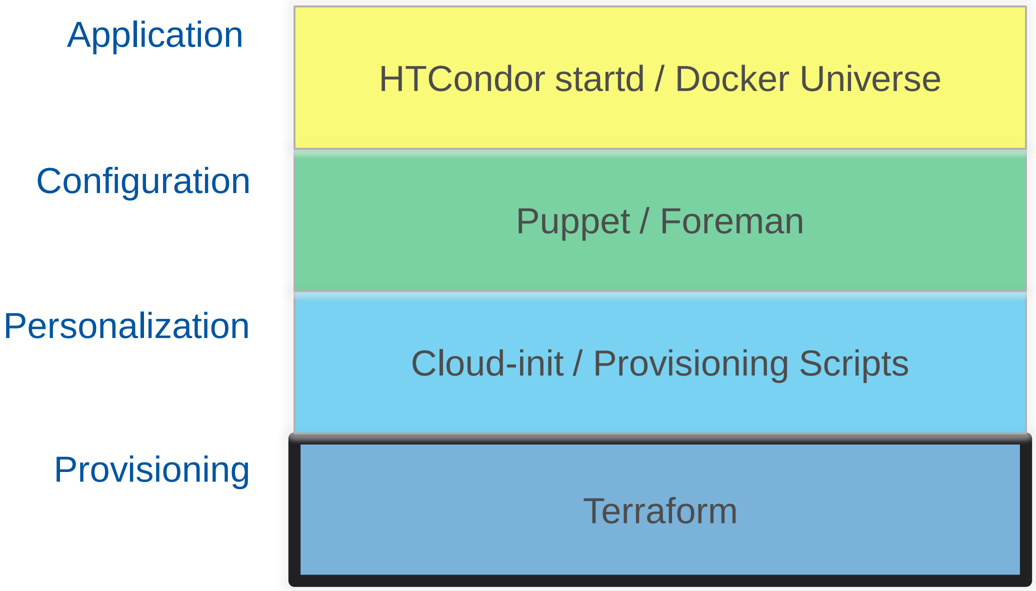
Configuration — Puppet / Foreman

Personalization — Cloud-init / Provisioning Scripts

Provisioning — Terraform

xBatch Stack

Job of each layer is just to bootstrap the next



Application

HTCondor startd / Docker Universe

Configuration

Puppet / Foreman

Personalization

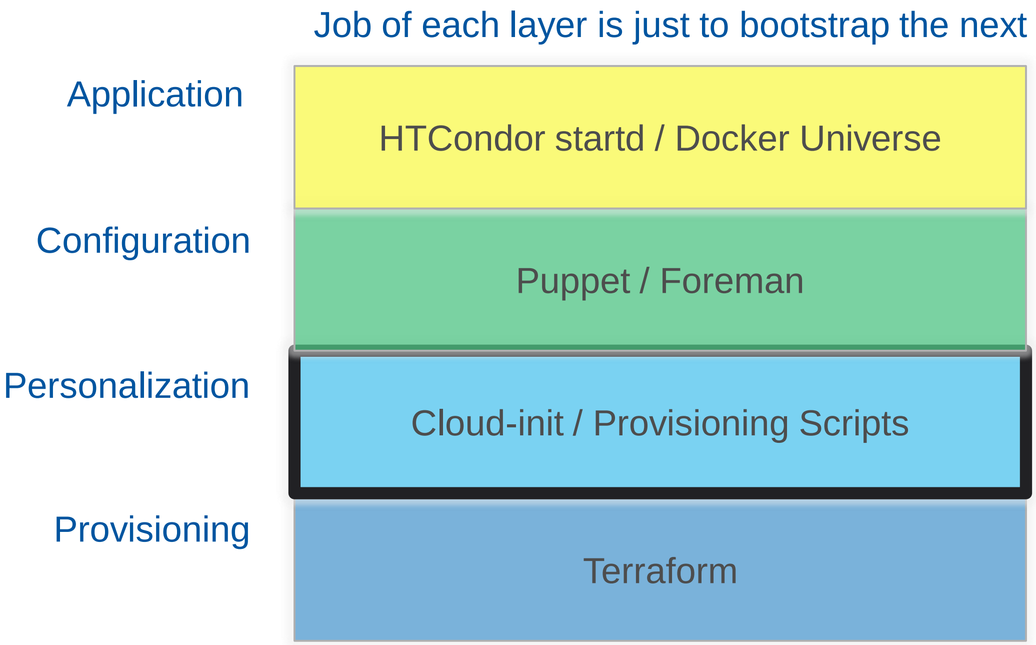
Cloud-init / Provisioning Scripts

Provisioning

Terraform

- Terraform selected as industry standard tool to abstract APIs
- Support out of the box for OpenStack (ie T-Systems, CERN) or CloudStack (Exoscale)
- Issues if used to expand / shrink regularly, but ideal for our purposes

xBatch Stack



Job of each layer is just to bootstrap the next

Application — HTCondor startd / Docker Universe

Configuration — Puppet / Foreman

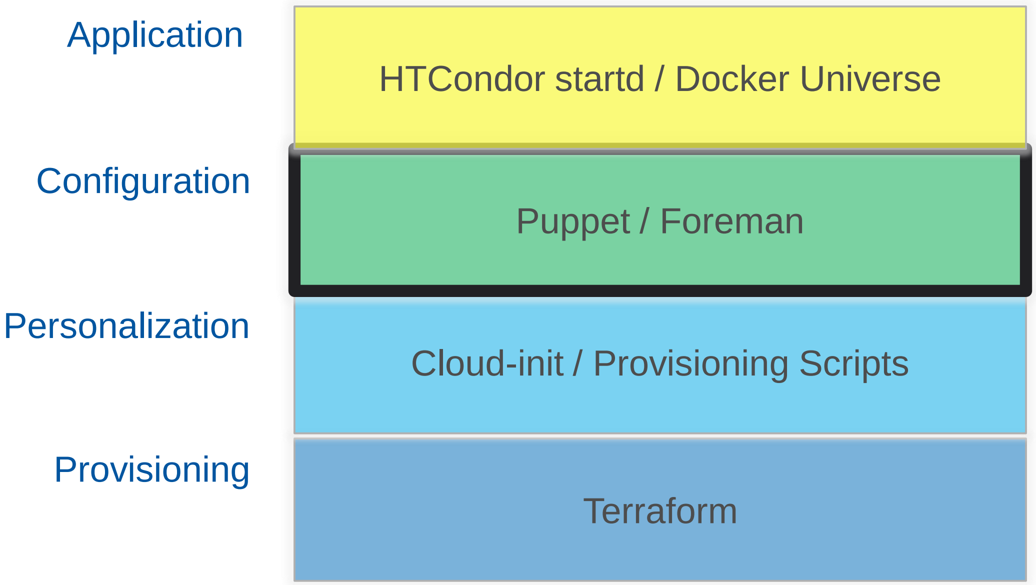
Personalization — Cloud-init / Provisioning Scripts

Provisioning — Terraform

- Cloud-init personalizes machine
- Install and configure puppet client
- Use one-shot time limited secret, unique to each machine, to sign off x509 certificate needed for puppet & condor
- Terraform post-exec where cloud-init support lacking

xBatch Stack

Job of each layer is just to bootstrap the next



Application — HTCondor startd / Docker Universe

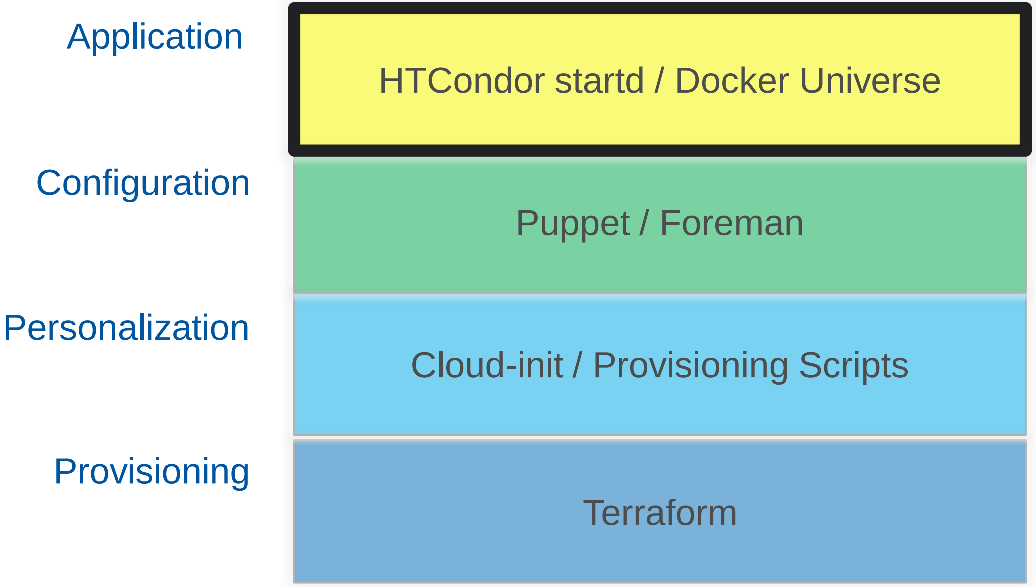
Configuration — Puppet / Foreman

Personalization — Cloud-init / Provisioning Scripts

Provisioning — Terraform

- As with internal machines, foreman used for classification & inventory, puppet used for configuration
- Some differences with internal machines, but re-use components & configuration where possible

xBatch Stack



Job of each layer is just to bootstrap the next

Application — HTCondor startd / Docker Universe

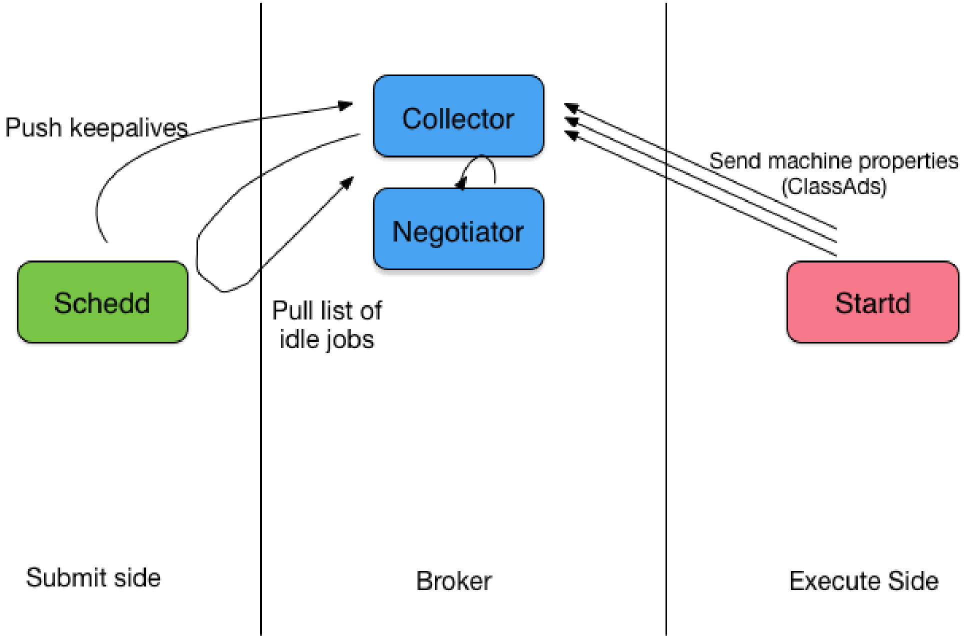
Configuration — Puppet / Foreman

Personalization — Cloud-init / Provisioning Scripts

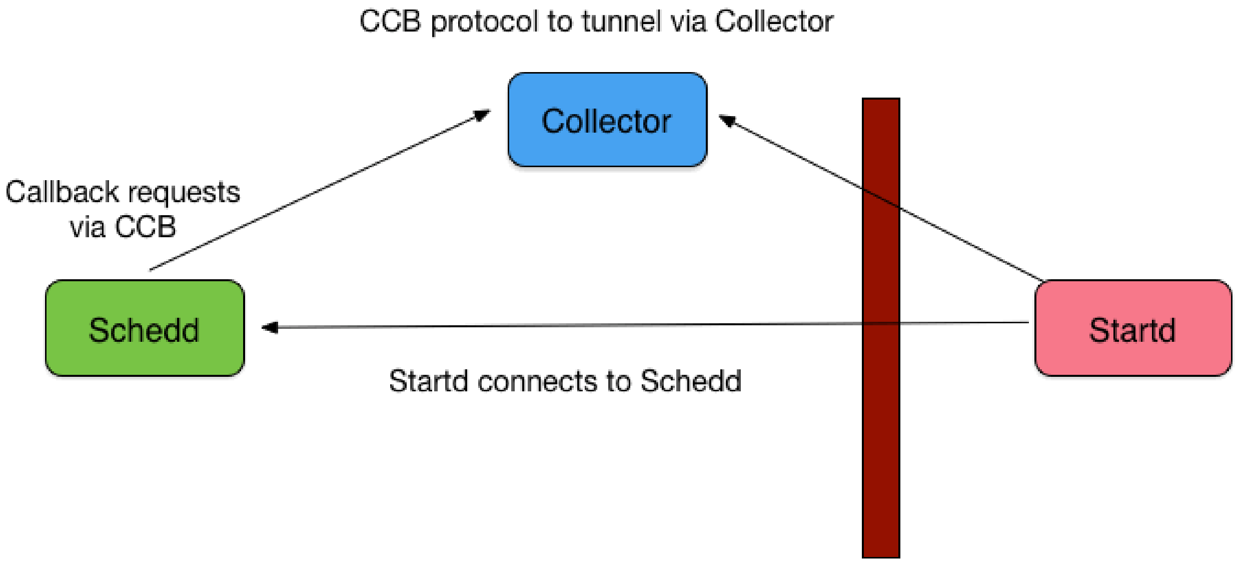
Provisioning — Terraform

- HTCondor with Docker ”universe” to abstract cloud machine from wlcg worker node environment
- Host provides CVMFS and HTCondor but can use Cloud-provided CentOS images
- HTCondor can be configured to work across firewalls & NATs

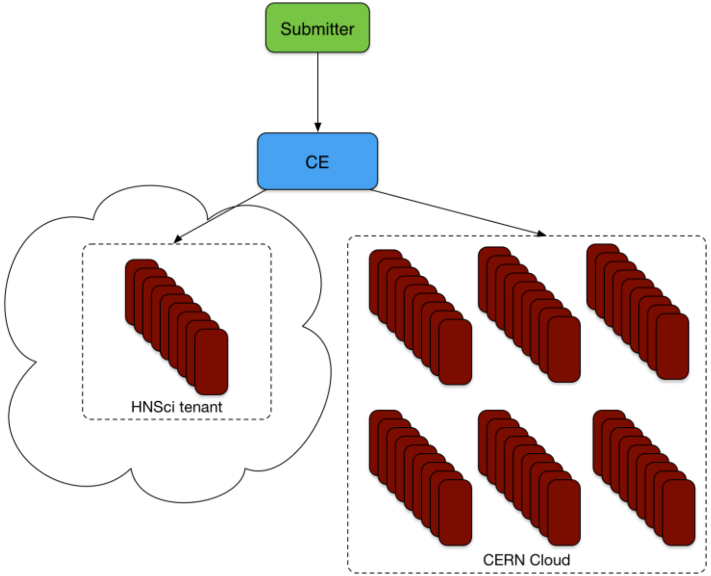
HTCondor Communication



Communication via firewall


CCB protocol to tunnel via Collector
Collector
Callback requests via CCB
Schedd
Startd
Startd connects to Schedd

CERN HTCondor to HNSci Cloud



- HTCondor works with symmetric matching of Host Properties with Job Requirements
- We route jobs explicitly asking for cloud resources (ie “WantHNSciRHEA” “WantHNSciTSys”) to machines in those clouds
- For experiments, they can be monitored as specific sites. For HTCondor, they are separate routes

T-Systems Challenges

- T-Systems primarily RFC1918 addresses, requiring NAT.
- No issue for HTCondor, but additional issue for managing the network resource
- Initial deployment used self-managed SNAT server
  - Single point of failure, that in fact failed during Availability Zone outage
- Migrated to new T-Systems managed SNAT service
  - Different flavours of service by # of connections difficult to provision for
- Both solutions require manual intervention from T-Systems to set ports to 10Gb
  - Actual bandwidth has varied under testing
- Still unexplained network issues leading to expired jobs

RHEA Challenges

- Prefer consistent use of provisioning via terraform against Cloud APIs rather than intermediate PaaS
- Necessary to re-provision resources after contract changes
- Exoscale flavours provide more RAM than strictly required
  - 8 core/32GB RAM (4GB/Core), WLCG standard is 2GB/core
- Terraform CloudStack provider sets disk in GiB, but gives status in KiB
  - Leads to circular provisioning for any change
  - Required to “ignore” Disk in terraform resource definition
  - We learn that we should just migrate to terraform-exoscale :)

Positives

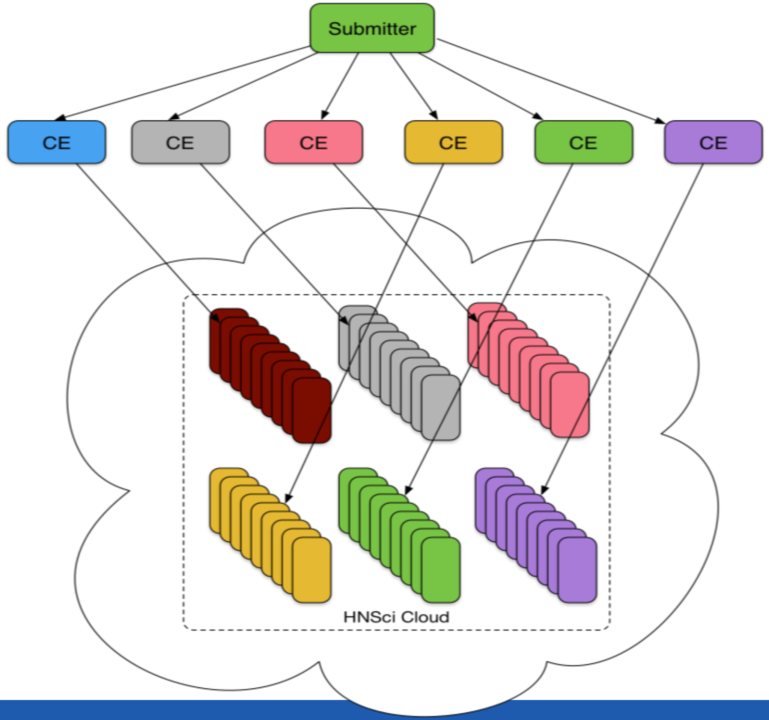
- TSystems API speed now much improved from previous engagement
  - No bottlenecks for terraform provisioning
- Exoscale performance very good, and useful features like online resizing of instances

WLCG Cloud Consolidation

- 8 of the 10 members of the Buyers Group have WLCG workload
- Agreement to consolidate to a shared WLCG tenant
- Reduce effort at procurer sites to support WLCG workload in Commercial Cloud
- Reduce network traffic between each procurer site and commercial clouds to support WLCG workload
- Simplification for LHC experiments to exploit commercial cloud resources

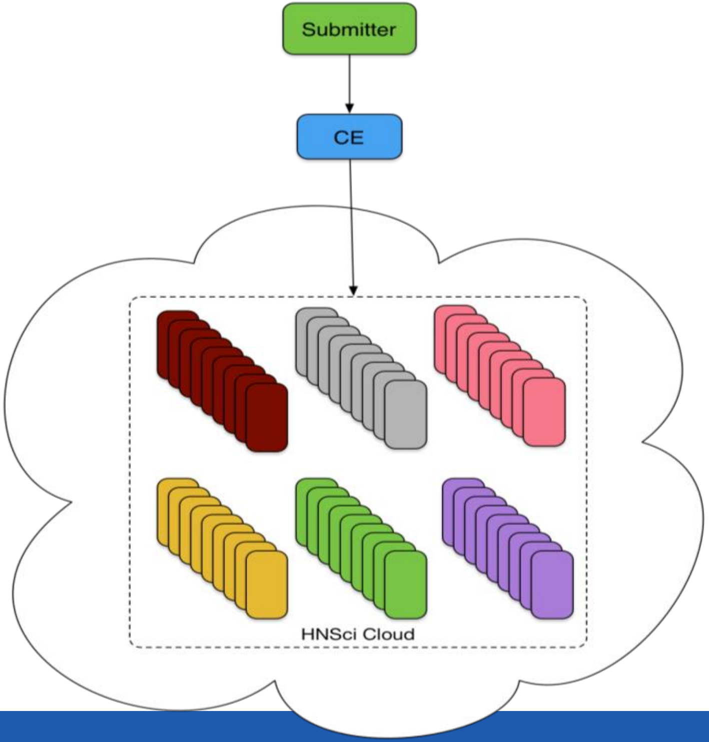
Unconsolidated

- In the current setup, an experiment could have to define 8 additional sites to send jobs to the same resources



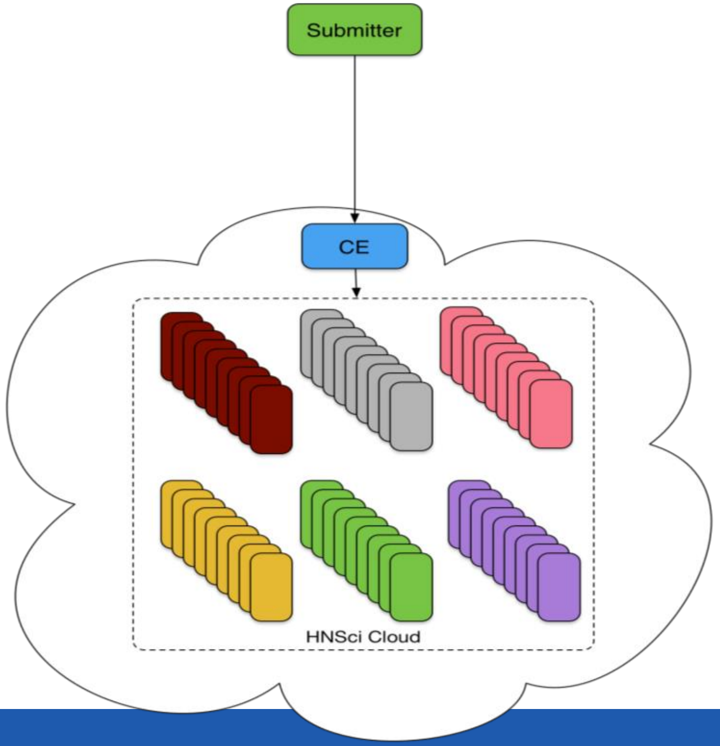
Shared tenant

- With a shared tenant, and a single entry point (CE), only one site has to be set up, per cloud vendor



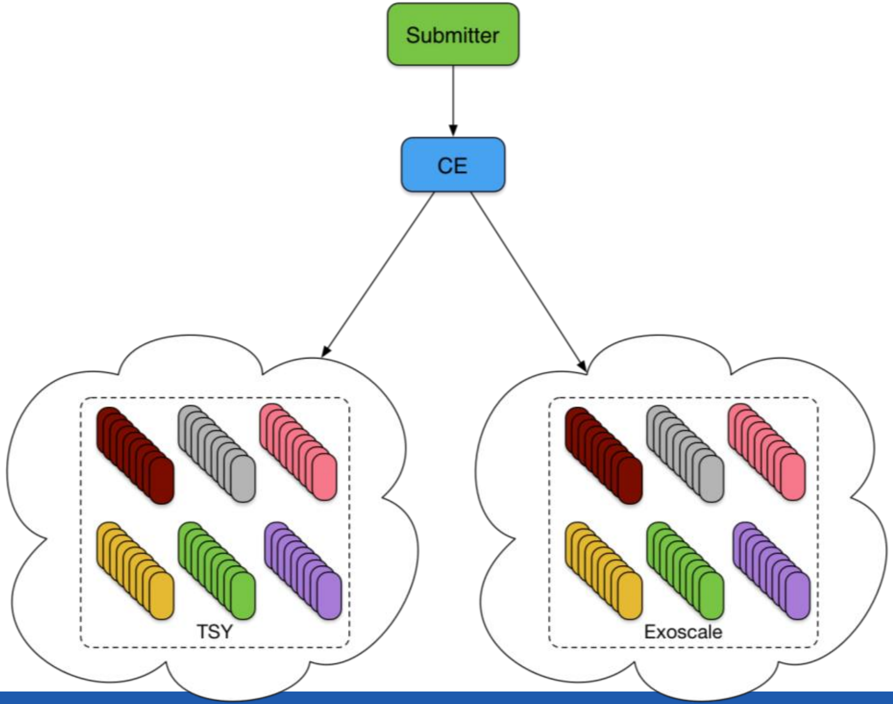
Vendor batch service

- If Cloud vendors provide metered batch services (ie HTCondor) then there's the possibility of further simplifying



Multicloud

- If we have to manage multiple clouds, having entry points / routes onsite may remain the best solution.



Questions?