
CERN

# Deploying to T-Systems

Ben Jones

# Strategy

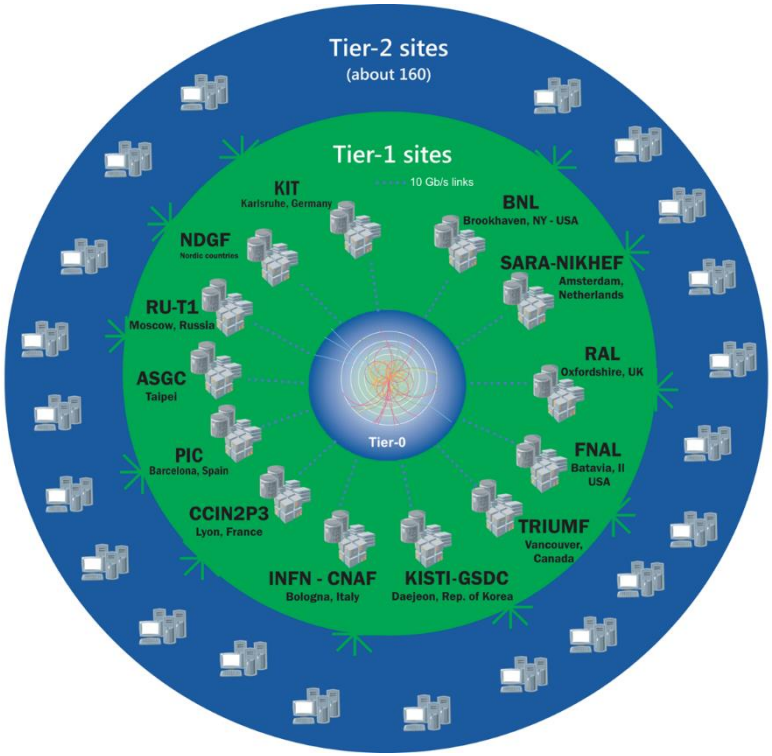
- Can we provision, manage and send workload to external batch machines using same toolset and entry points as internal machines?
- Workload should be constant for length of engagement
- Business as usual

# Worldwide LHC Computing Grid

**TIER-0 (CERN):**
**data recording, reconstruction and distribution**

**TIER-1:**
**permanent storage, re-processing, analysis**

**TIER-2:**
**Simulation, end-user analysis**




**nearly 170 sites, 40 countries**

**~350'000 cores**

**500 PB of storage**

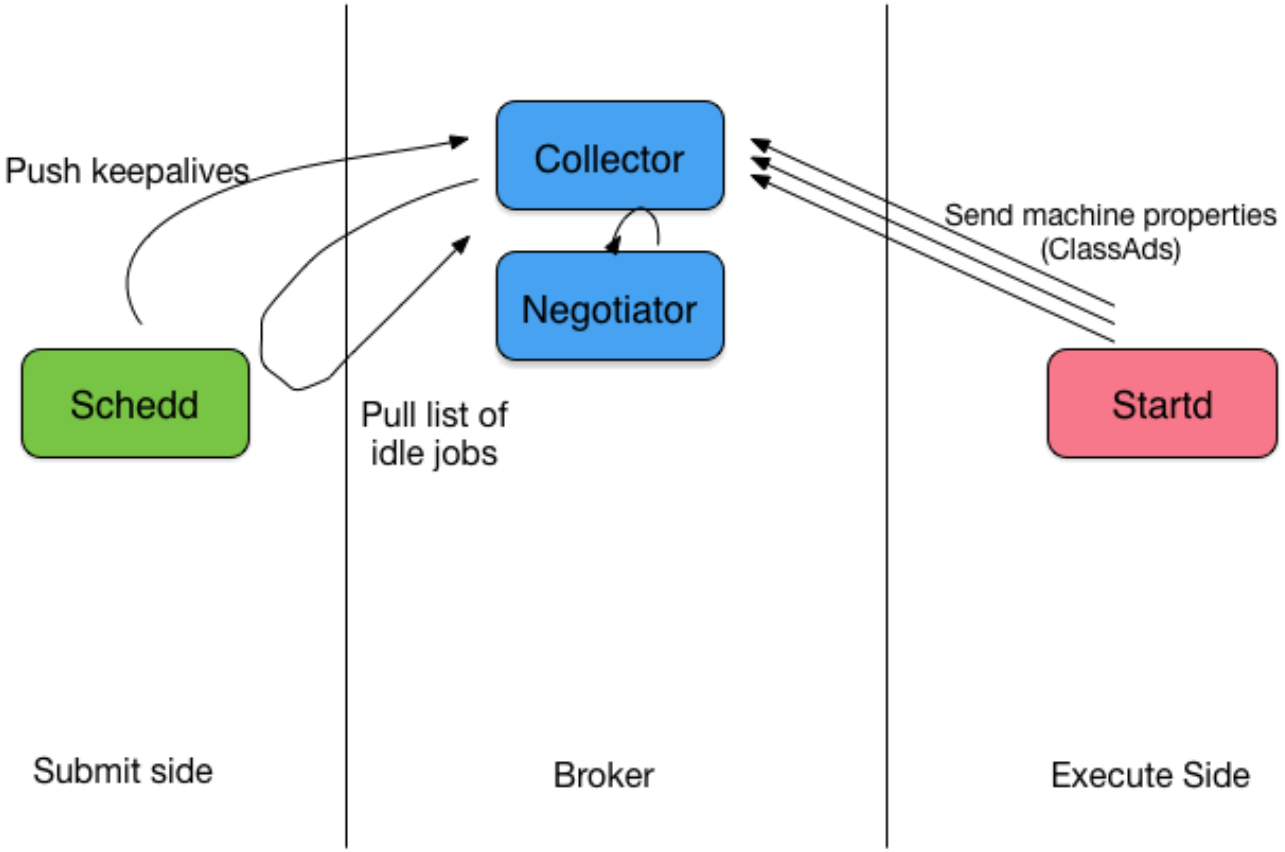
**> 2 million jobs/day**

**10-100 Gb links**

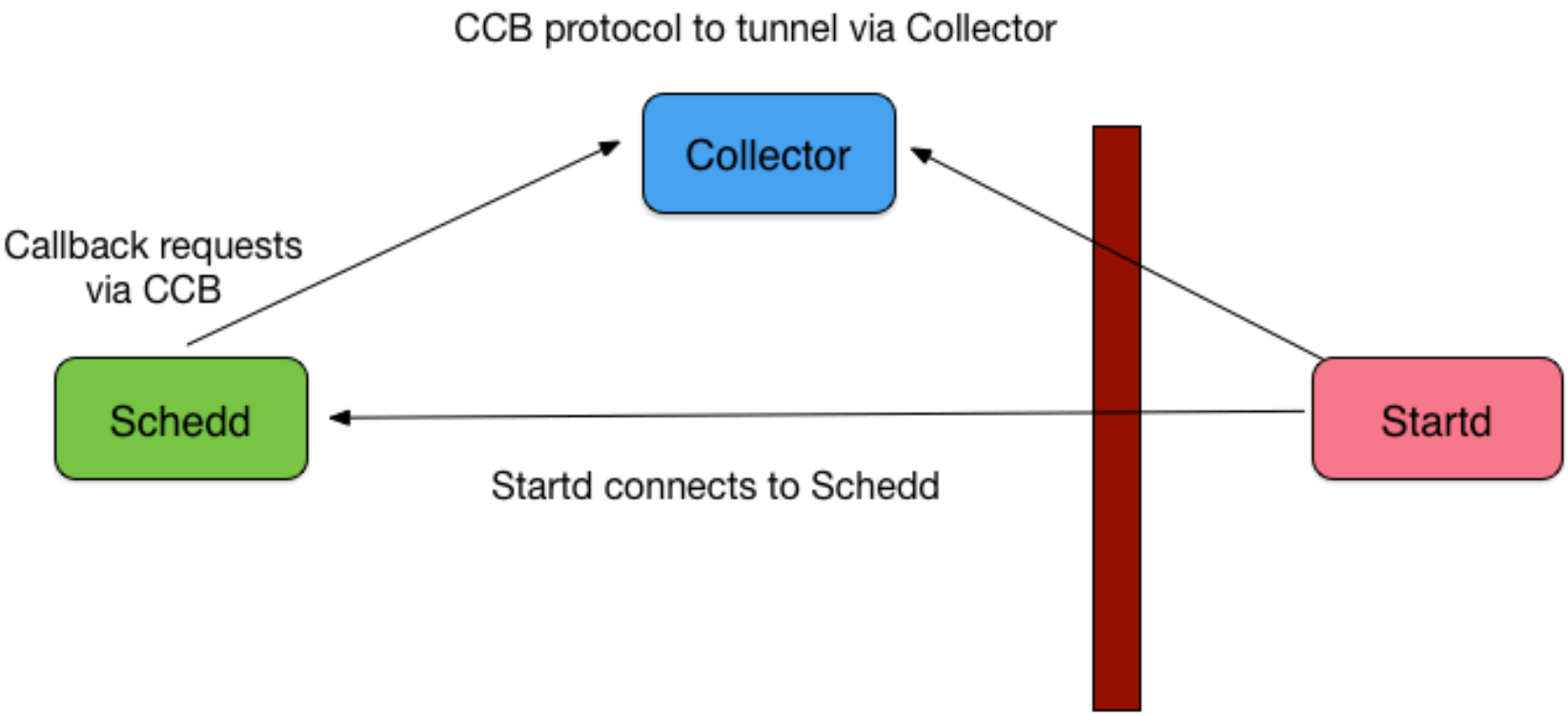
# CERN Batch Service

- Large part of Batch workload is Grid
- Moving to HTCondor
  - All Cloud activity (inc SoftLayer) using HTCondor
- Grid jobs accepted on special schedds called “CEs”
- Jobs submitted to schedds are matched to compute resources using HTCondor as a broker
- Machines can advertise particular properties and have workload routed to them appropriately

# HTCondor communication



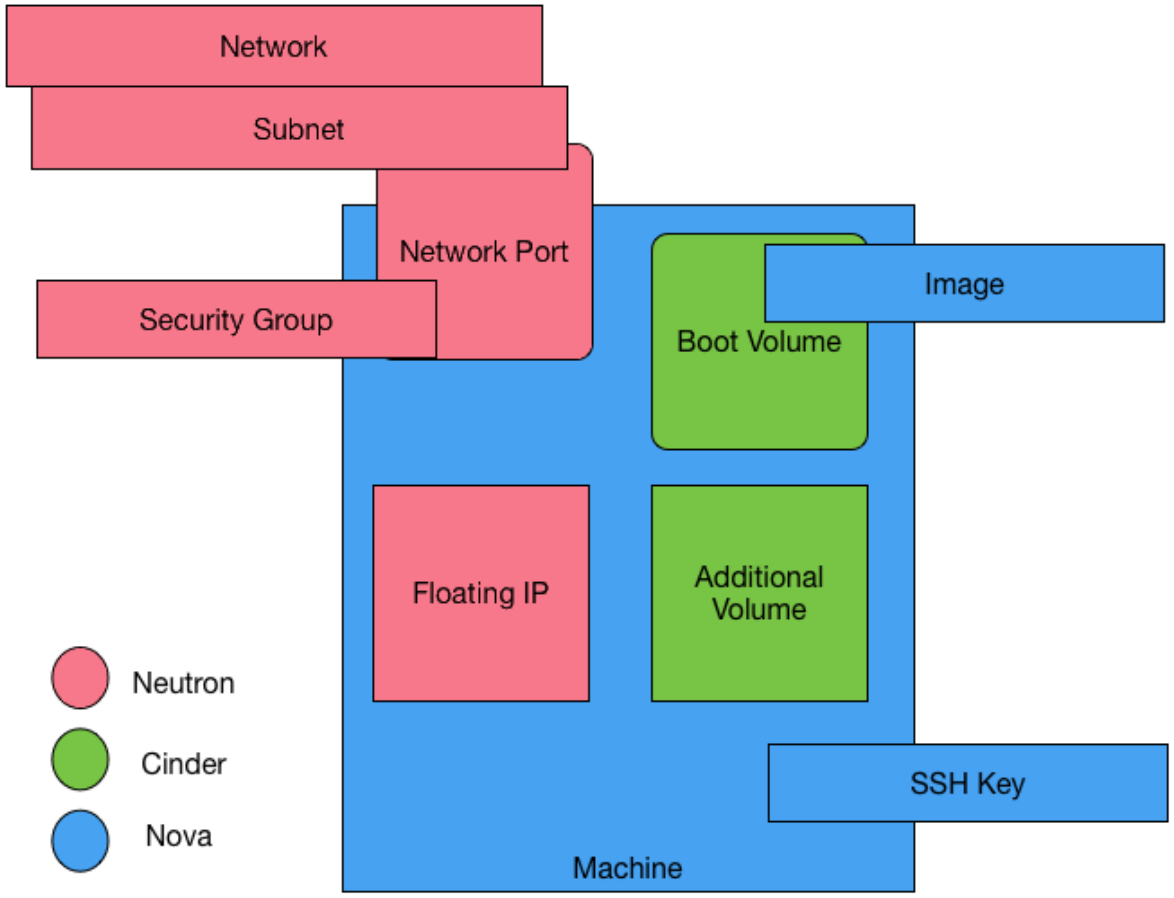
# Communication via firewall



# Provisioning

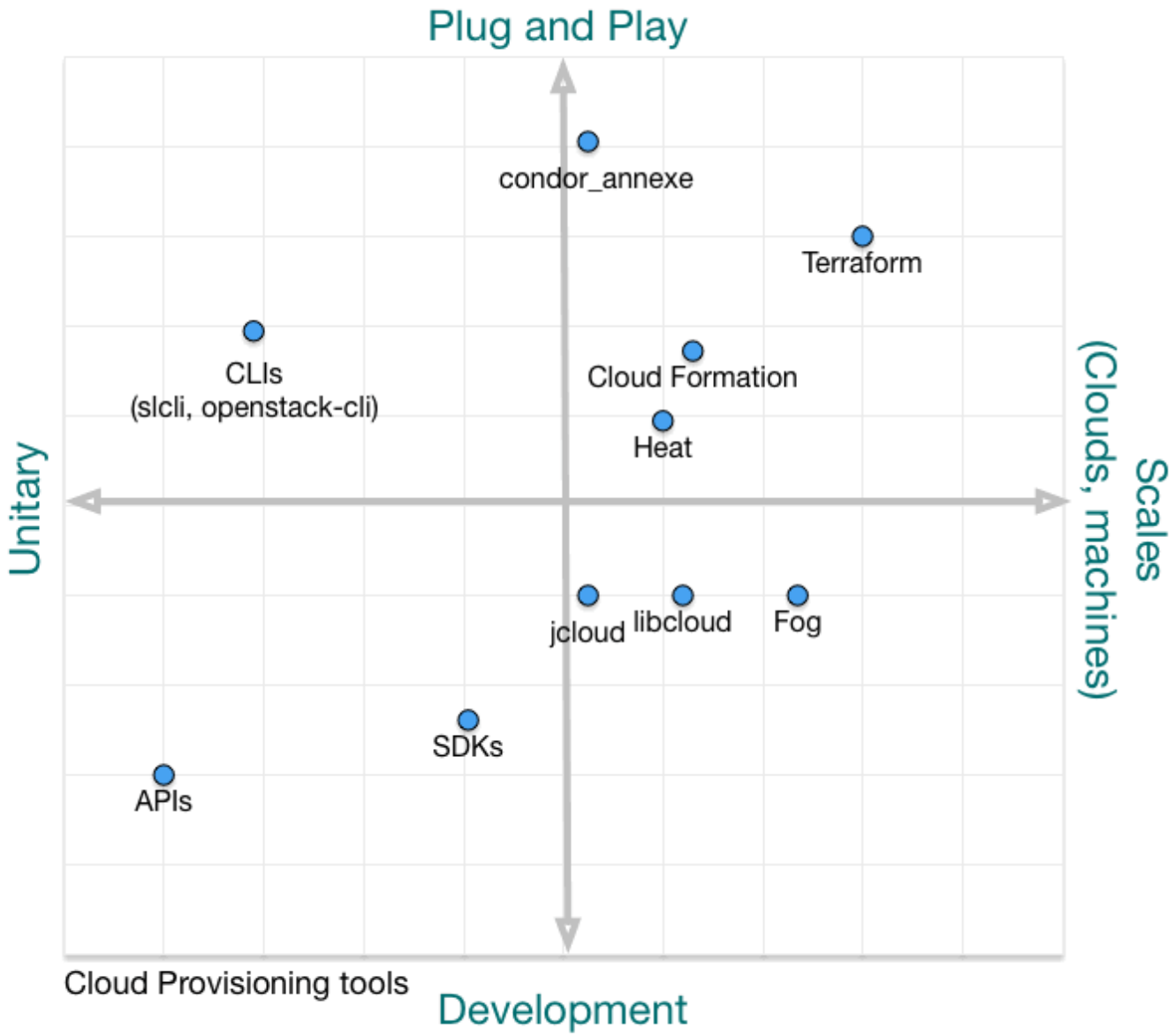
- Job of provisioning is to get node to the point that it can run puppet
- Limited amount of personalisation required
  - Name, puppet config, some rpms
- Internally we use cloud-init
  - Not (yet) an option in T-Systems
- We have more complex requirements for provisioning in T-Systems than previous activity
  - Deploying storage & provider requirements

# Building a T-Systems node



# Building a machine the hard way

- 12 CLI commands
  - Most CLI commands require UUIDs from other commands.
  - Subnet needs network ID, Port needs subnet ID and Security Group ID, Machine needs IDs from Port, Disks, SSH keys
- More API calls required, including Authentication, catalog discovery
- Automation needed for scale and agility


Plug and Play
condor_annexe
Terraform
CLIs
(slcli, openstack-cli)
Cloud Formation
Heat
Unitary
Scales
(Clouds, machines)
jcloud
libcloud
Fog
SDKs
APIs
Cloud Provisioning tools
Development

# terraform

- Nice templating approach
- Declarative
- Lots of support for different cloud providers
- Works with internal cloud too
- OpenStack support there via RackSpace “gophercloud” go SDK

# APIs

- Some API issues early on with T-Systems
- Bespoke “T-Systems” API versus “OpenStack”
- Addition of newer Neutron API smoothed problems for tooling
- Some investment in ecosystem tooling necessary if bespoke APIs to be exploited

# Configuration

- Initial contextualization needed to install puppet, and defined puppet configuration
- For now contextualisation not possible via userdata
  - DHCP options could be used in future
- Means only current option is to push to machine (via ssh)
  - Easy to do in terraform with remote-exec provider, but means we rely on public IP

# Current Status

- Can build end-to-end with terraform
- Service nodes created
- Incremental improvements can be made to process, but no showstoppers
- Next phase: adding compute to HTCondor pool

# Questions?