Plugging Cloud and VM-based resources into WLCG

Andrew McNab
University of Manchester
LHCb and GridPP

Overview

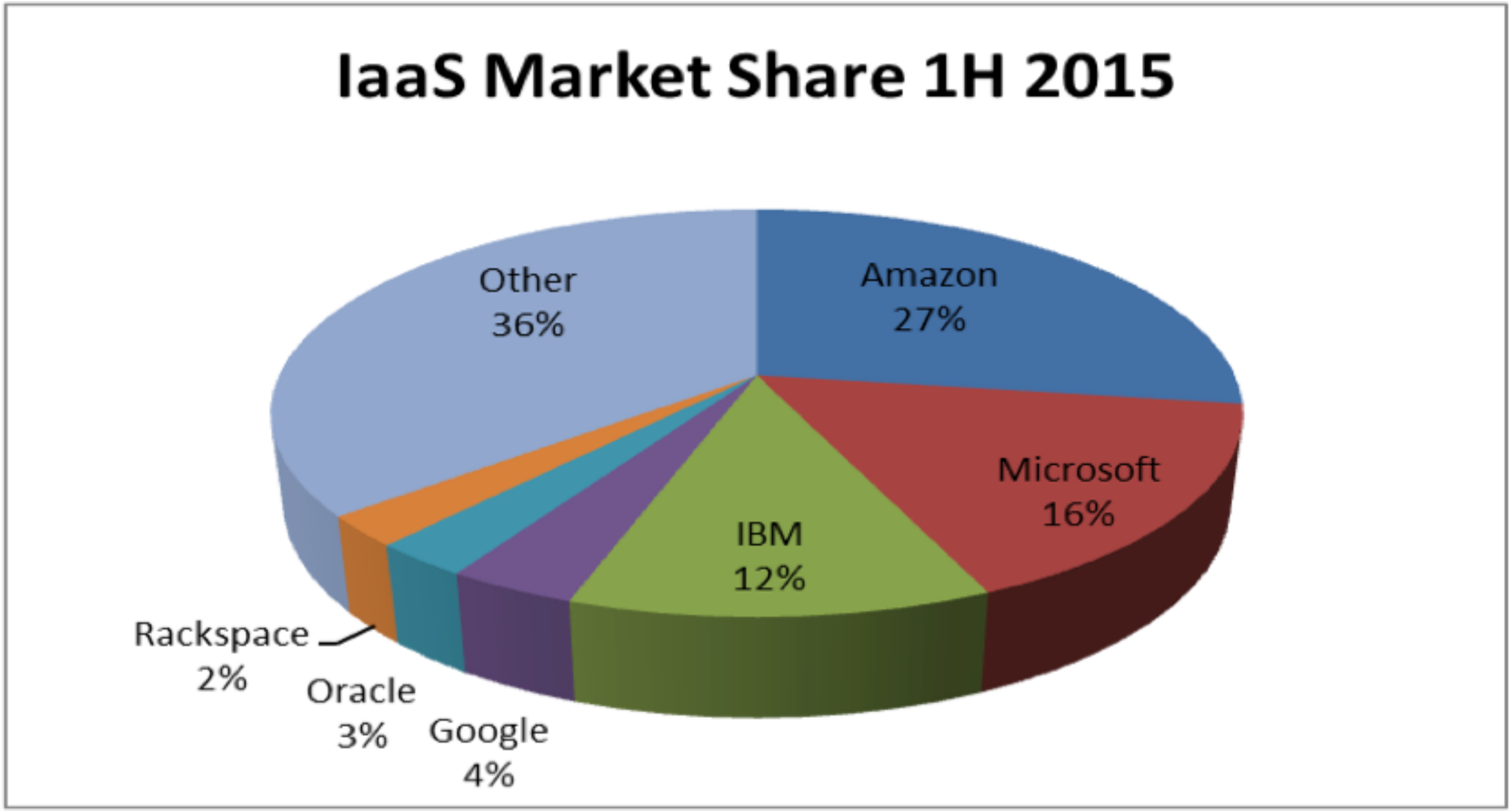
- Medium term: to end of LS2
  - "What will things be like in 2020?"
  - Commercial landscape
  - Academic/research landscape
- Different ways of using Clouds
- Clouds/VMs to simplify site operation
- Experiments status and plans
- Other types of logical machine

What do I mean by “Cloud”

- In HEP we tend to say “Cloud” and just mean Infrastructure-as-a-Service (IaaS) Clouds
    - Arguably our Grid(s) are a kind of Platform-as-a-Service cloud
- IaaS Clouds give you a programmatic way to manage and use
    - Virtual Machines
    - Virtualised storage
    - Virtualised networking
- All this at a remote service
- “Cloud” terminology originally promoted by Amazon Elastic Compute Cloud in 2005

Commercial providers

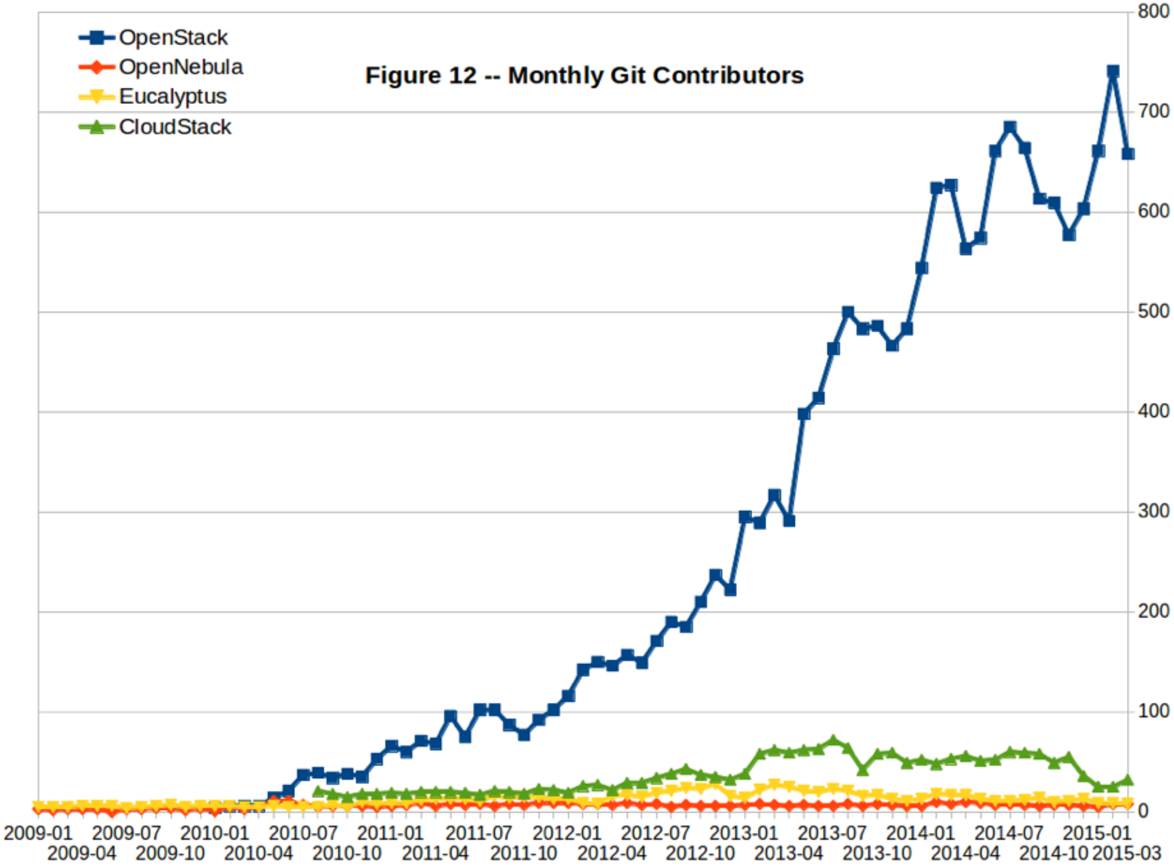
- Amazon (EC2), Microsoft (Azure), IBM (OpenStack,…), Google Cloud Platform, Rackspace (OpenStack/Azure/VMware), …



- Ralph Finos, http://wikibon.com/public-cloud-market-shares-2014-and-2015/

Open source cloud implementations

- OpenStack is way ahead of other open source clouds in terms of commercial and academic adoption and number of developers
- Compare Linux taking off around 1995
- Particularly important for us as academic / research institutes tend to prefer open source
- OpenStack structure means institutes can contribute to features they need
- In particular, CERN has adopted OpenStack



Guesses about the future

- Amazon and Microsoft will continue to dominate commercial provision, with closed source solutions
  - Only Amazon provides Amazon EC2; you can buy MS Azure software to run your own cloud service
    - Compare Apple MacOS vs Microsoft Windows
  - Maybe Google will catch up (Android did for phones)
  - Other players do not have the money/agility to catch up, but won’t go away
- Academic/research providers will standardise on OpenStack for cloud services as they have on Linux
  - National/regional/institution HPC/HTC centres will increasingly offer cloud rather than batch
  - Big projects, e.g. SKA, looking at HPC on OpenStack, and at contributing required features to the codebase

Scenarios for running jobs on Clouds/VMs

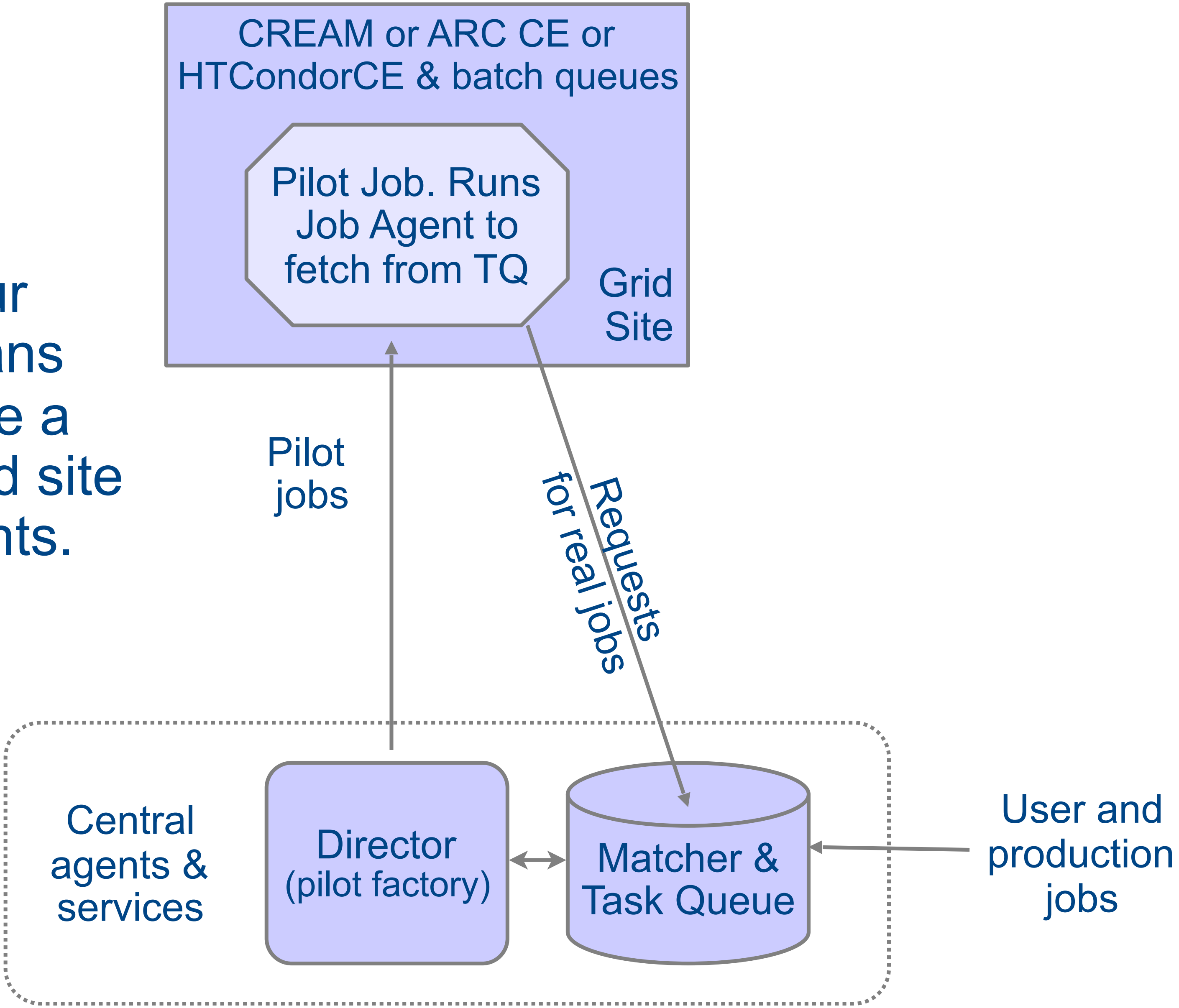
- “Just extend your batch service” to external cloud VMs
- Experiment creates batch worker VMs in response to pilot or payload job pressure
- Resource provider creates experiments’ VMs in response to observed demand (“Vacuum”)

Extending batch farms onto clouds

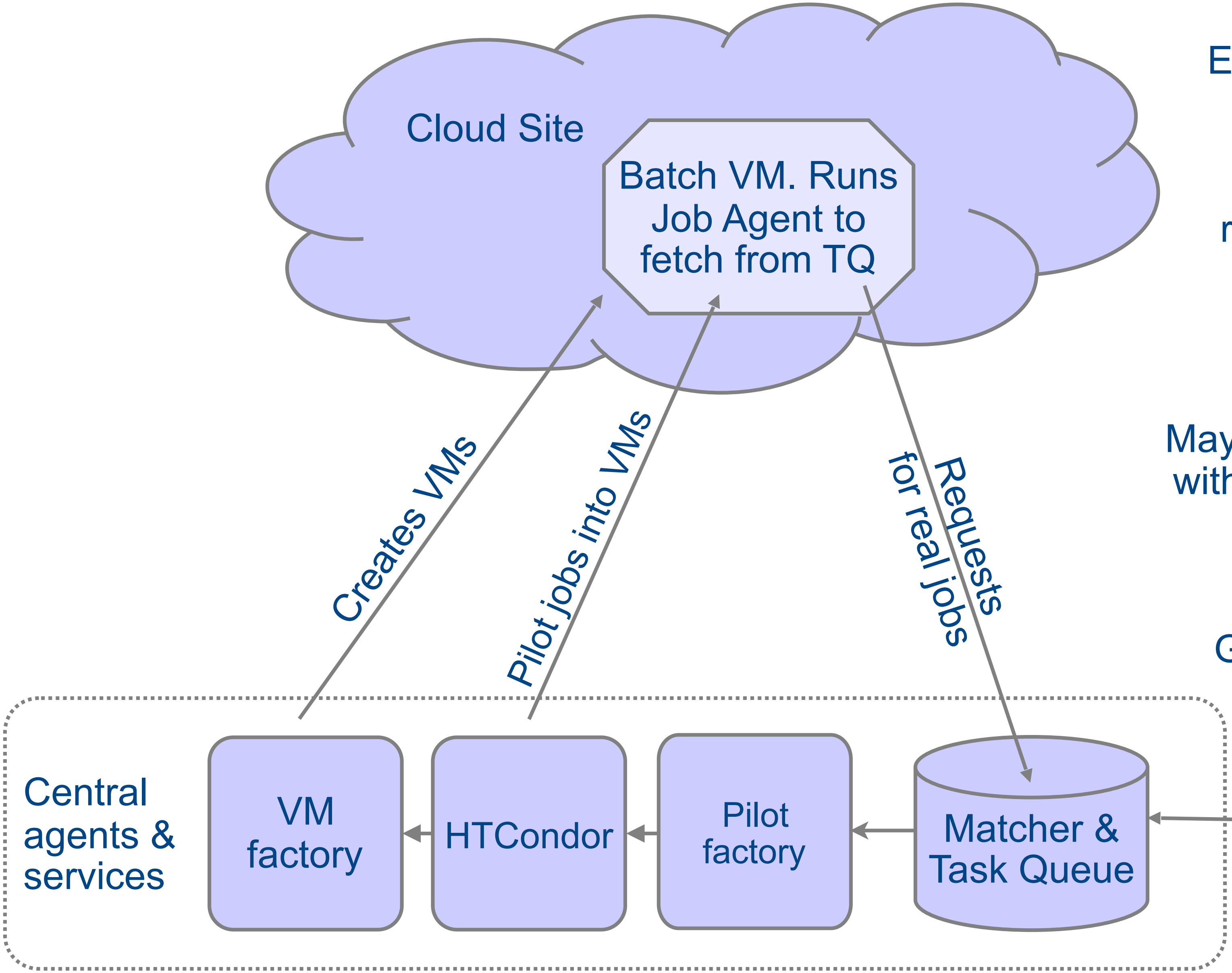
- Cloud systems like OpenStack allow virtualization of fabric
- Resource provider might already have a batch farm + gatekeeper
    - LSF/CREAM, HTCondor/ARC, HTCondorCE
- Resource provider might already run WNs in Cloud VMs (eg CERN lxbatch)
- So one option is to
    - Just buy in capacity from commercial providers or a collaborating institute (eg Wigner)
    - Set the VMs up as batch WNs
    - CERN is favouring this approach for future cloud procurements
- Not very elastic out of the box
    - But may be possible to drain WN VMs so they can be released
- Big question is whether this kind of baseload capacity is cheaper in-house or not

The Grid with Pilot Jobs

“Just extend your batch farm” means the site looks like a conventional grid site to the experiments.


CREAM or ARC CE or HTCondorCE & batch queues
Pilot Job. Runs Job Agent to fetch from TQ
Grid Site
Pilot jobs
Requests for real jobs
Central agents & services
Director (pilot factory)
Matcher & Task Queue
User and production jobs

Experiment creates VMs according to demand



Experiment manages cloud resources.

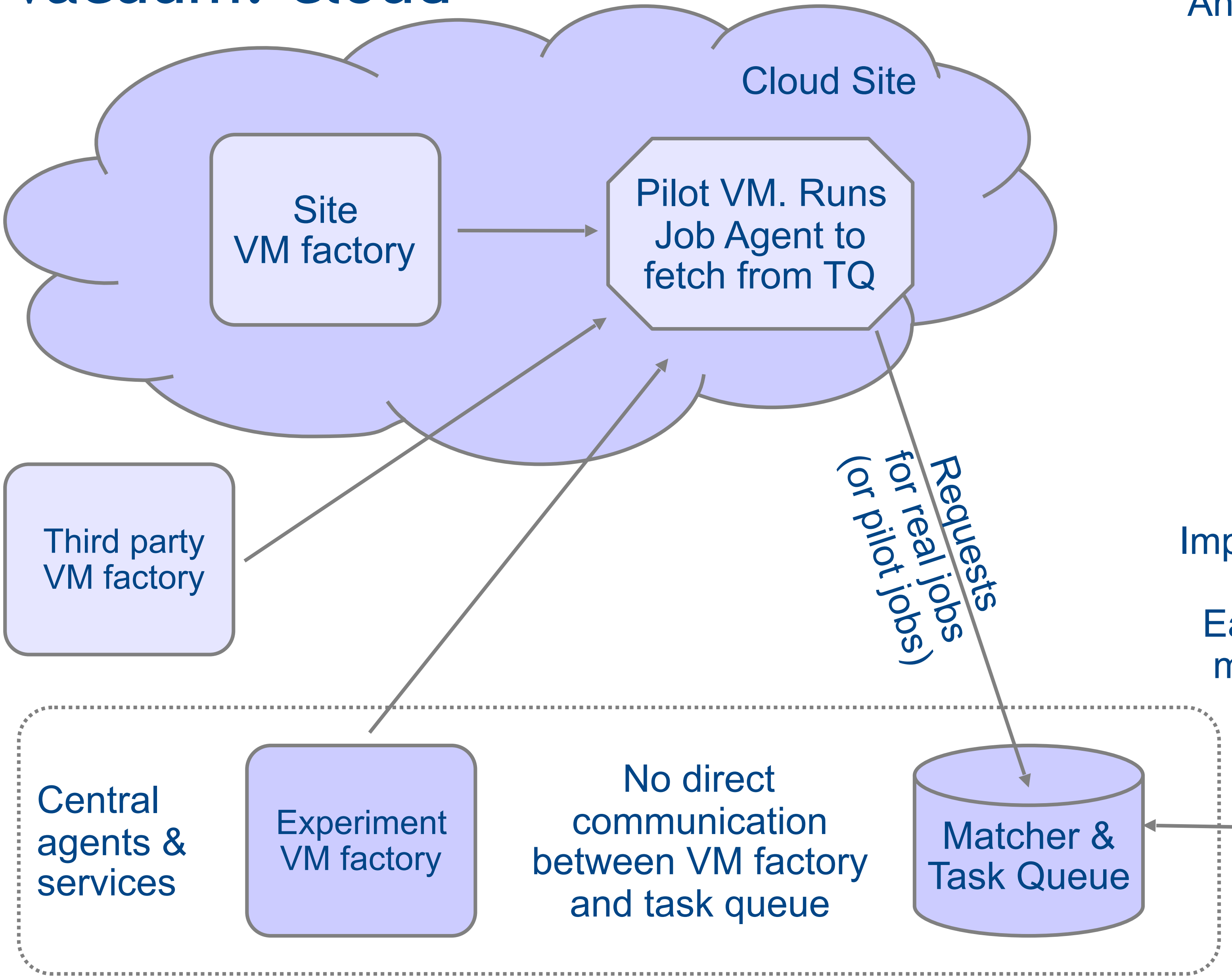
Creates VMs in response to demand from pilot jobs.

Very elastic.

May be difficult to share with other experiments.

CloudScheduler, AutoPyFactory, and GlideinWMS use this approach.

Vacuum: cloud


Cloud Site
Site VM factory
Pilot VM. Runs Job Agent to fetch from TQ
Third party VM factory
Requests for real jobs (or pilot jobs)
Central agents & services
Experiment VM factory
No direct communication between VM factory and task queue
Matcher & Task Queue
User and production jobs

An external VM factory that manages VMs.

Can be run centrally by experiment or by site itself or by a third party

VMs started and monitored, but not managed in detail ("black boxes")

Implemented by Vcycle

Easy to mix VMs from multiple experiments.

Vacuum: autonomous hypervisors

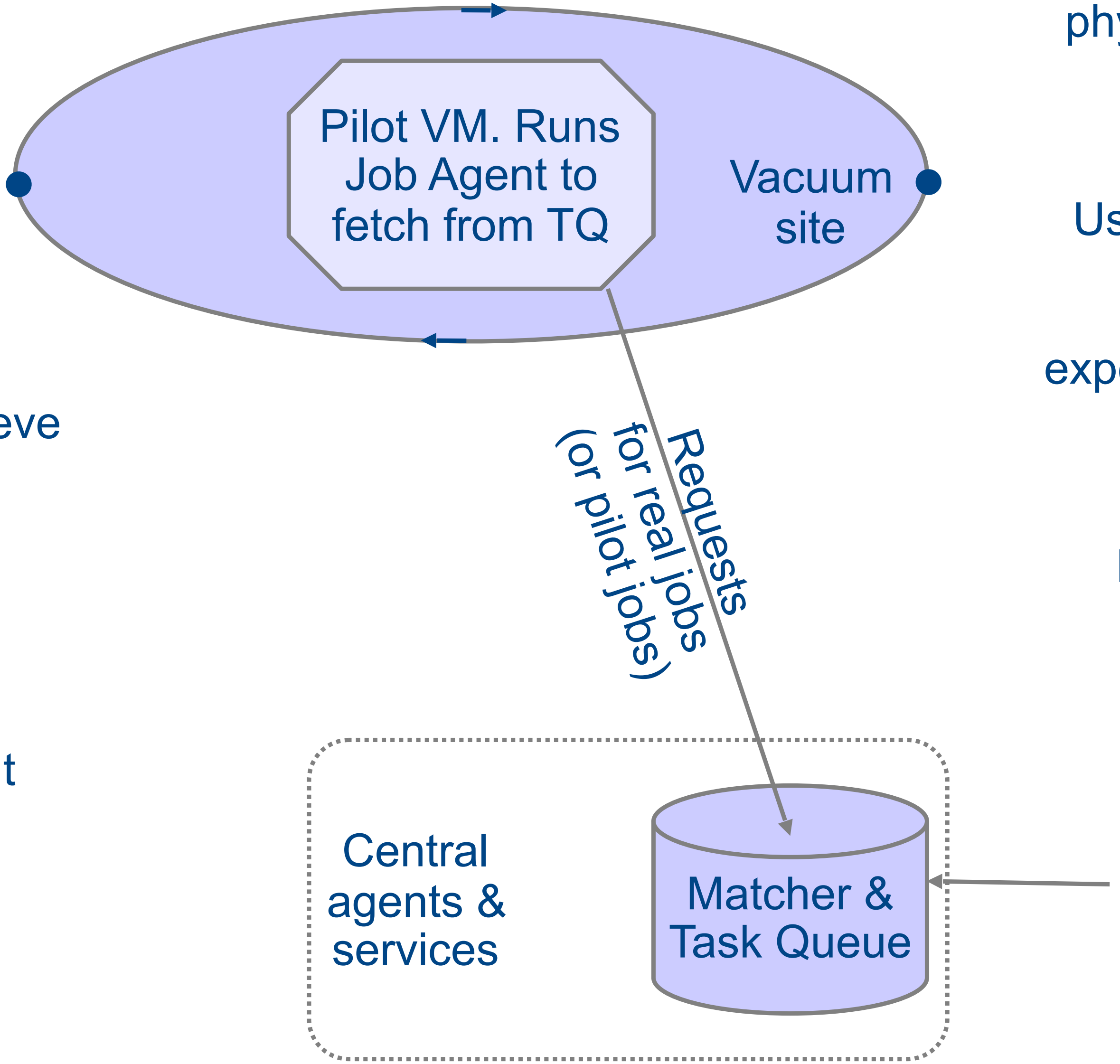
Strip the system right down and have each physical host at the site create the VMs itself.

Use feedback from VM outcomes to decide which experiments' VMs to create as slots become free.

Easy to mix VMs from multiple experiments

Vac works this way, with inter-hypervisor communication to achieve desired targets shares between experiments.

BOINC with VMs is effectively using this model too, but with completely independent hypervisors.


Pilot VM. Runs Job Agent to fetch from TQ
Vacuum site
Requests for real jobs (or pilot jobs)
Central agents & services
Matcher & Task Queue
User and production jobs

Clouds/VMs in UK Tier-2 Evolution

- Motivated by continued tightening of funding and need to operate with less staff
- For compute, sites offered VM-based solutions which reduce the number of service types required at the site to ~1
  - Shift complexity inside write-once, run-everywhere experiment VMs
- For sites with university or department OpenStack (etc)
  - Vcycle: multiple experiments’ VMs in a shared tenancy
- For sites managing their own hardware
  - Vac: autonomous hypervisors running VMs, no headnodes etc
  - Vac-in-a-Box website generates per-site SL6 kickstart scripts
- Vac/Vcycle very simple “glue”: about 3500 lines of Python each

ALICE

- ALICE uses cloud resources internally
  - eg for interactive analysis, build farms, …
- Have done some tests on HLT
- However, they are not keen on having to be system managers of VMs on cloud sites running jobs
- ALICE had positive experience with CERN second cloud procurement on DBCE
  - All four LHC experiments had VMs, run by Vcycle
  - For ALICE, CernVM + HTCondor; managed by CERN/IT
  - ALICE just had to submit batch jobs
- If necessary, ALICE could run in VMs started in any way by the resource provider (even Vac)

ATLAS

- Cloud computing accounts for approximately 5% of ATLAS's workload
- Variety of facilities
  - Static clouds, distributed cloud systems, high-level trigger (HLT) , opportunistic research (private) clouds, commercial clouds
- Mostly CernVM-based, but not exclusively
- Different VM provisioning mechanisms: natively or via cloud plugins
  - HTCondor, Vac/Vcycle, CloudScheduler, AutoPyFactory, NECTAR-MOAB
  - Similar functionality but designed to integrate into existing infrastructures

- Plan to expand utilization, enhance reliability, and ease of operations
  - Better and more responsive monitoring
  - Utilization of discovery services in a context-aware environment
- HLT Farm (Sim@P1)
  - Further testing for rapid (automated) switching during beam periods
- Further evaluation of commercial options
  - Primarily Amazon EC2 but also cloud-brokered solutions

CMS

- CMS relies on GlideinWMS and either VMs from a Glidein factory or from a third-party (eg ~static configuration or Vcycle etc)
- CMS T0 at CERN in Run 2 is CMS's share of T0 expressed as VM capacity in CMS's project on central OpenStack service
    - ie CMS manages the VMs itself rather than use lxbatch
- Heavy use of HLT: aim for 50% usage, by running in shutdowns/stops, between fills, and on unused capacity during runs.
- Using cloud resources at several other sites (RAL, Imperial, Bologna, CNAF, and in Finland)
- For next CERN cloud procurements, leading to Helix Nebula Science Cloud
    - Plan to move from RAL test infrastructure to global pool
    - More data-intensive work than MC production in next round of tests
- Move from custom images to CernVM
- More sites may move to cloud-only
- May see large-scale (real) production running on commercial clouds occasionally

LHCb

- Uses the same VMs everywhere, on Vcycle/OpenStack and Vac sites
  - CernVM + DIRAC pilot client (no pilot job involved)
  - “Get it right once, should work everywhere”
- Peaks at around 1200 mostly single-processor VMs (~3% of total)
- Mostly Monte Carlo but also some production jobs
- Plan to request some pledged resources at Tier-1s as VMs
  - Will validate user analysis jobs with data access
  - Multiprocessor user jobs (in VMs, LHCb can test with cgroups etc)
- 500 VM LHCb tenancy at CERN runs with very little intervention
  - Only breaks when new VM or Vcycle versions scaled up there first
- LHCb’s plan is to follow sites’ preferences for how the majority of resources are presented

Themes from the experiments

- Notable variety in experiments’ preferences/investments about internals of VMs
- All refer to possibility (hope) of running same VMs everywhere
- All standardising on CernVM
  - Very convenient: small images, “free” software/security updates
  - WLCG could encourage WLCG cloud sites to make CernVM available among default images?
- Should continue adding modules to CernVM contextualization for common requirements, to further simplify user_data files
- WLCG could act to promote commonality between VMs and interoperation with different VM lifecycle managers (CloudScheduler, GlideinWMS, Vcycle, Vac, …)
  - Machine/Job Features specification (draft HSF technical note) a prototype for doing this?
  - Also logging for security? Live benchmarking? Monitoring?

Other types of “logical machine”

- Containers
  - Explosive growth of Docker bringing attention to this
  - “Heavyweight containers” approximate VMs, but share host’s kernel and native performance
  - Can be run by cloud systems (eg OpenStack)
  - Limitations of privileges model means need cvmfs from host too, or run something in userspace
- Unikernels
  - Build OS components into application, running directly on hypervisor or bare metal
    - Hypervisor takes care of hardware drivers, so one unikernel application can run everywhere
  - Application and kernel share address space so no context switching within unikernel logical machine

Summary

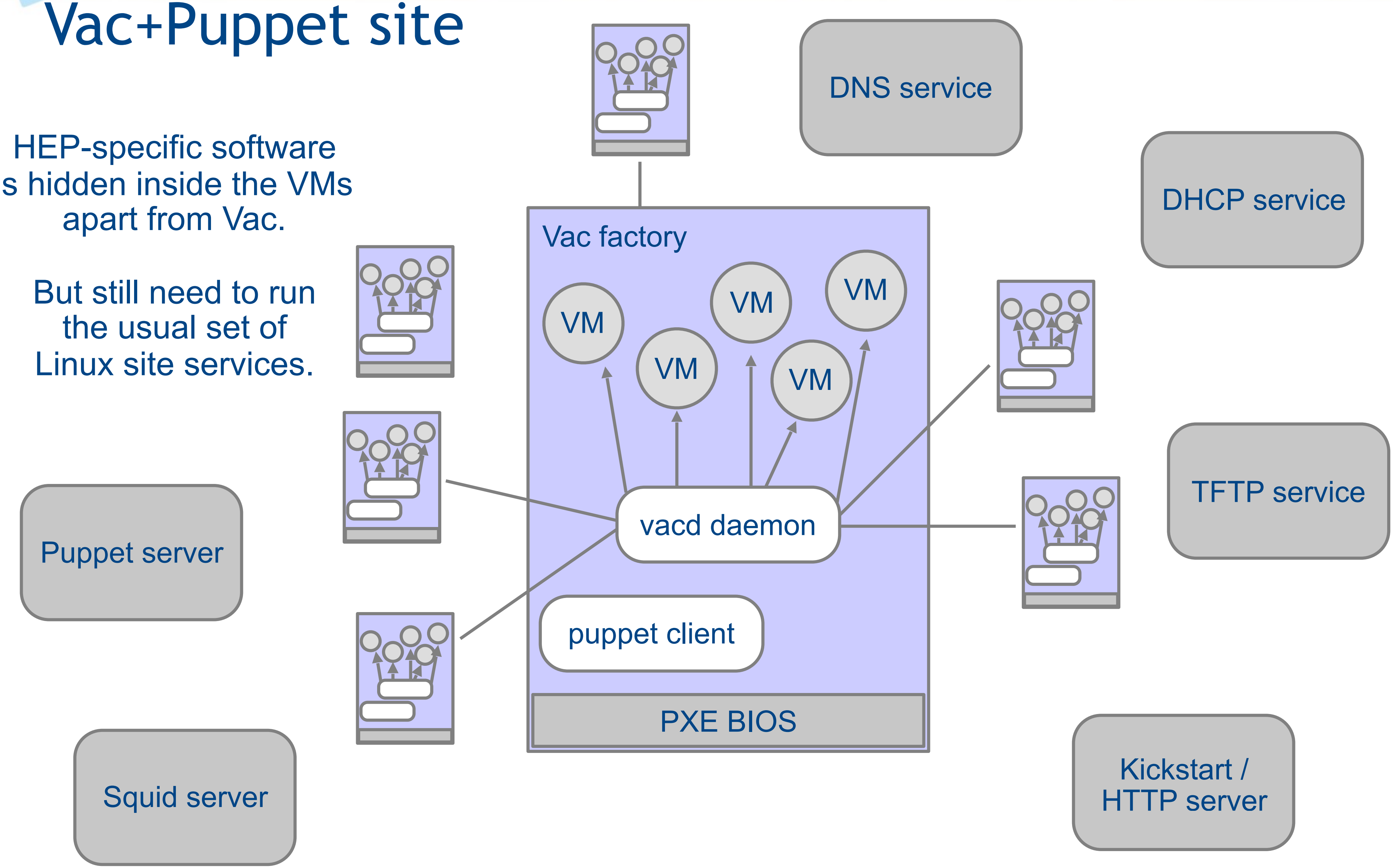
- Clouds are a significant part of WLCG now
  - CMS T0 use alone would justify that without everything else that is going on
- Lots of software solutions
- But some commonalities already: eg CernVM
- Several opportunities for WLCG to help
  - Push common modules into CernVM to simplify contextualization
  - Agree APIs between VMs, sites, and experiments
  - Ideally any VM lifecycle manager would be able to run any VM
- Evolution of commercial and research computing landscapes strongly suggests that more resources will naturally be as Cloud/VM
  - Although might be other types of Logical Machine not VMs as such

Extra slides

Vac+Puppet site

HEP-specific software
is hidden inside the VMs
apart from Vac.

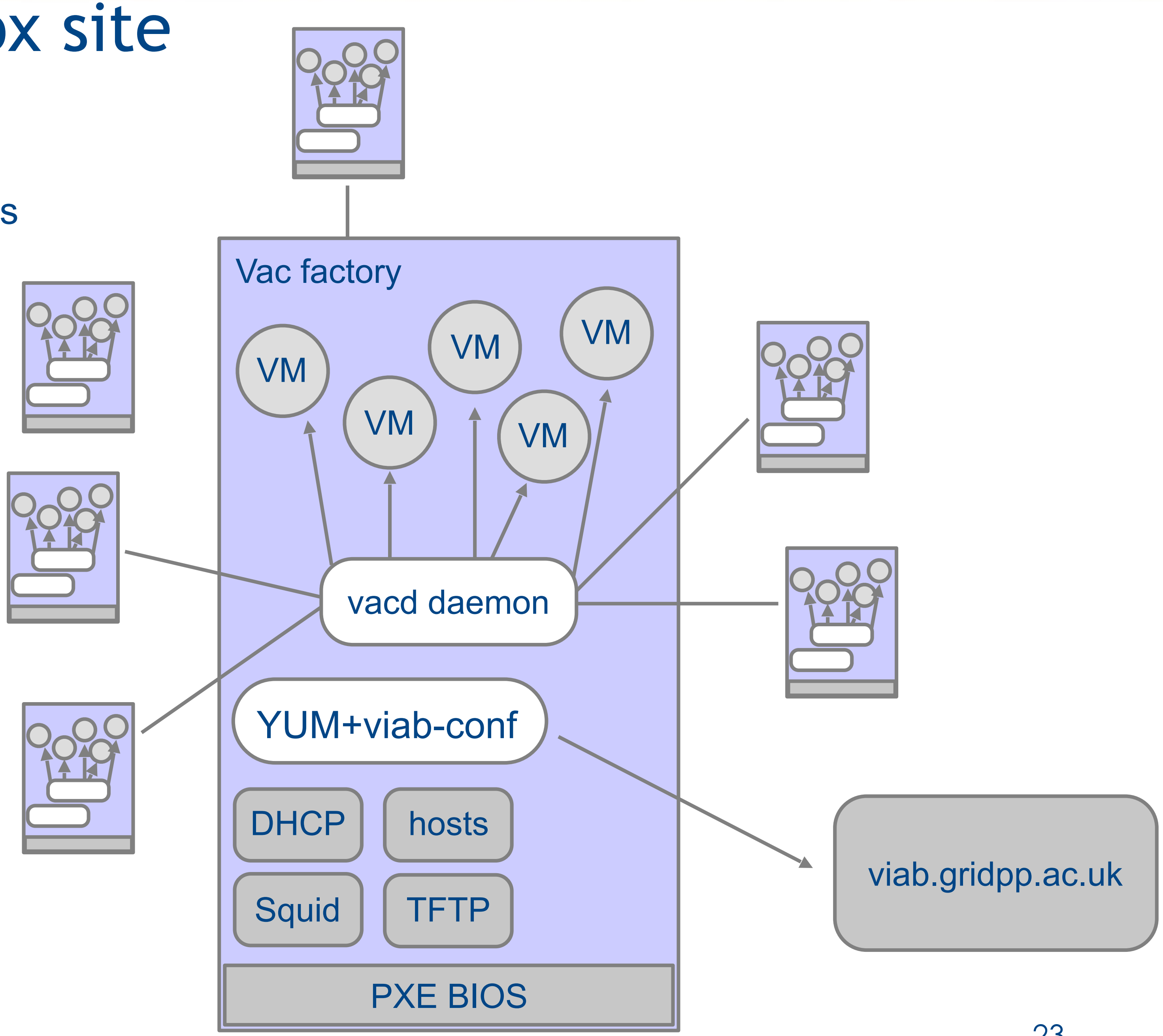
But still need to run
the usual set of
Linux site services.



Vac-in-a-Box site

HEP-specific software is hidden inside the VMs apart from Vac.

Site services are hidden inside the Vac factory machines.


Vac factory
VM
VM
VM
VM
VM
vacd daemon
YUM+viab-conf
DHCP
hosts
Squid
TFTP
PXE BIOS
viab.gridpp.ac.uk