Vacuum Platform

Andrew McNab
University of Manchester
GridPP and LHCb

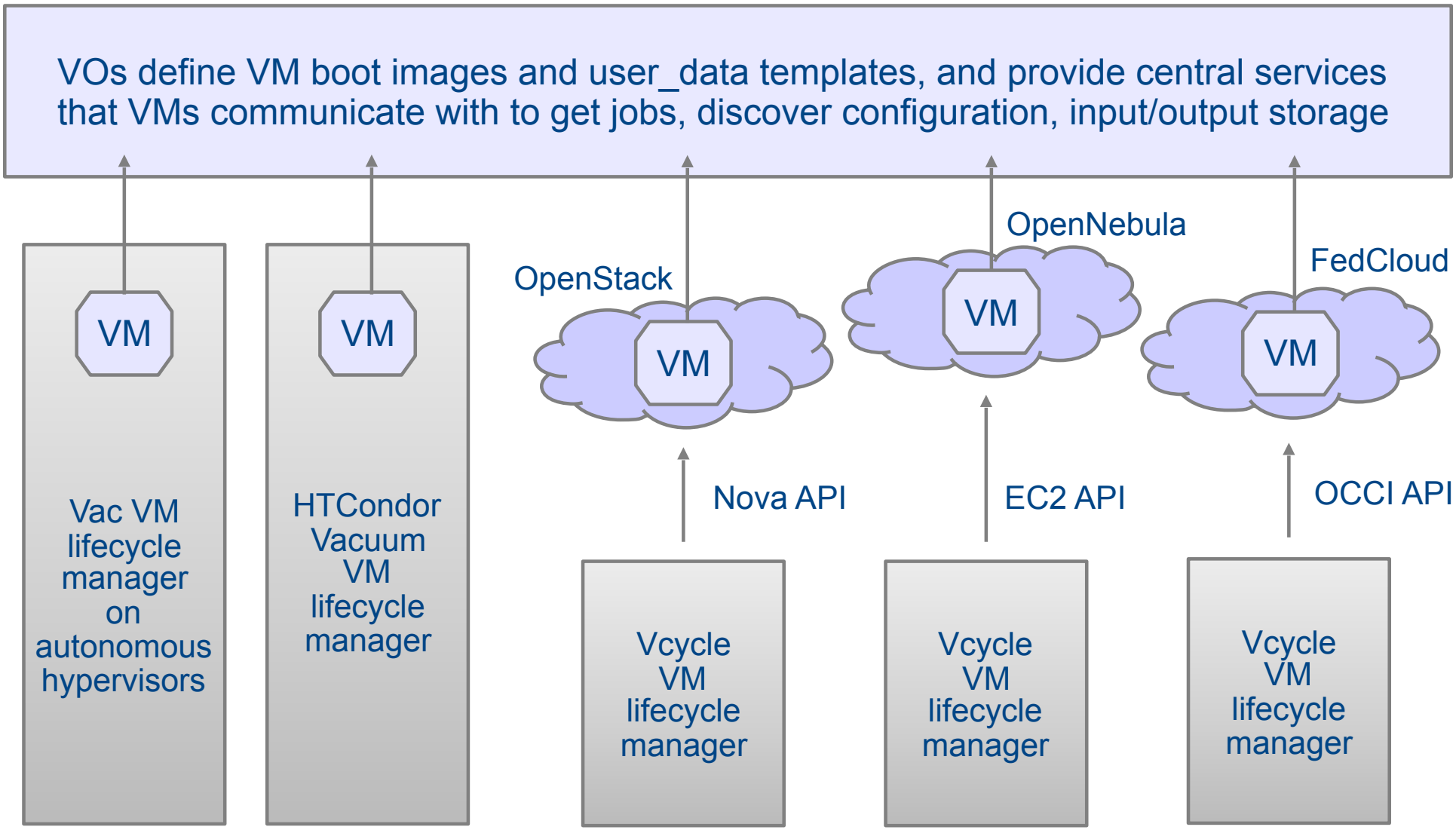
Background

- GridPP in the UK has an effort to simplify the operation of sites
  - To cope with constraints on staff funding
  - Easier to recruit new sites (eg outside High Energy Physics)
- For job execution, this has focused on using virtualisation
- Basic idea is to put complicated, HEP-specific software into VMs provided by the virtual organizations
  - Simplified by use of cvmfs and VOs' pilot job frameworks
  - Almost completely transparent to the jobs running inside VMs
- Sites need a way of creating/managing VMs
  - Vcycle (Nova/EC2/OCCI), Vac (autonomous hypervisors), HTCondor Vacuum
- Now in a position to support VOs outside of HEP, and would like to pursue being a EGI Community Platform as part of this

VM lifecycle

- Vacuum model assumes that VMs
  - Have a well-defined lifecycle
  - Which the VM can largely control itself
- In particular, that if a VM has no work to do for its VO, then it will shut itself down
- Allows VM lifecycle manager software to treat VMs as “black boxes”
  - Create a VM
  - It finds some work: create more VMs of that type
  - It doesn’t find any work: try VMs of another type (another VO)
  - Apply target shares between types of VM / VOs when doing this
- Simple enough to be run by resource provider (~site)
  - Makes it easy to have multiple VOs’ VMs running
  - Do not, for instance, have to create one OpenStack tenancy per VO

Vacuum platform



Pros and cons

- Pros
  - Vacuum model allows very simple VM lifecycle managers
    - eg Vac and Vcycle ~3500 lines of Python each
  - Common interface across multiple VM-creation platforms
  - Readily scales up to more sites
- Cons
  - Scaling not as good for increasing number of VOs
    - Currently requires manual effort per site to set up each VO
  - VMs must be able to operate autonomously
    - May be harder to use for applications where VM creation is directly integrated into workflow

Current sites and maximal VO support

		ATLAS	CMS	LHCb	GridPP DIRAC
Vac	Manchester	✔	✔	✔	✔
	Oxford	✔	✔	✔	✔
	Lancaster	✔		✔	✔
	Liverpool			✔	✔
	UCL	✔		✔	✔
Vcycle	CERN (LHCb)			✔	
	CERN (Dev)	✔	✔	✔	✔
	Imperial	✔	✔	✔	✔
	CC-IN2P3			✔	
HTCondor Vacuum at RAL (STFC)		✔	✔	✔	✔

Platform components (1)

- Standard procedures for VO to specify how to
  - fetch VM boot image from VO's URL
  - fetch user_data template to contextualize VMs from VO's URL
- WLCG Machine/Job Features (MJF) mechanism to get VM metadata
  - eg maximum VM lifetime
- $JOBOUTPUTS extension to MJF to allow VMs to supply log files to site and return message about why the VM stopped
- Aim to provide OpenStack-like environment where possible
  - e.g. HTTP metadata, ssh keys, user_data on 169.254.169.254
  - But user_data template mechanism can work round this on other platforms
- Support Cloud Init contextualization

Platform components (2)

Some reliance already on EGI services:

- APEL accounting using “job message” usage records
  - Message files generated by VM lifecycle managers directly
  - Sent to central APEL by running ssmsend
  - In production for Vac sites
- Register resources in GOCDB, for discovery, downtimes etc
  - uk.ac.gridpp.vac and uk.ac.gridpp.vcycle Service Types
- GGUS for operations tickets (none yet)
  - Vac/Vcycle Support Unit
- Drafting technical note describing interfaces and requirements:
  - https://github.com/HEP-SF/documents/tree/master/HSF-TN/draft-2016-VACPLAT

“Missing” EGI pieces

- Not using BDII
  - Would really like to avoid doing this
  - Believe that GOCDB is sufficient for discovery
  - Vcycle has support for GLUE 2.0 in JSON published via HTTPS
    - Will be added to Vac
  - Intend to publish GLUE/JSON URLs in GOCDB
- No SAM tests yet
  - VOs could do this within their VMs
  - Some fraction of all VM starts? Submit SAM jobs to VO pilot job framework?
  - Alternatively create dedicated VM definitions that run SAM tests that resource providers would run periodically

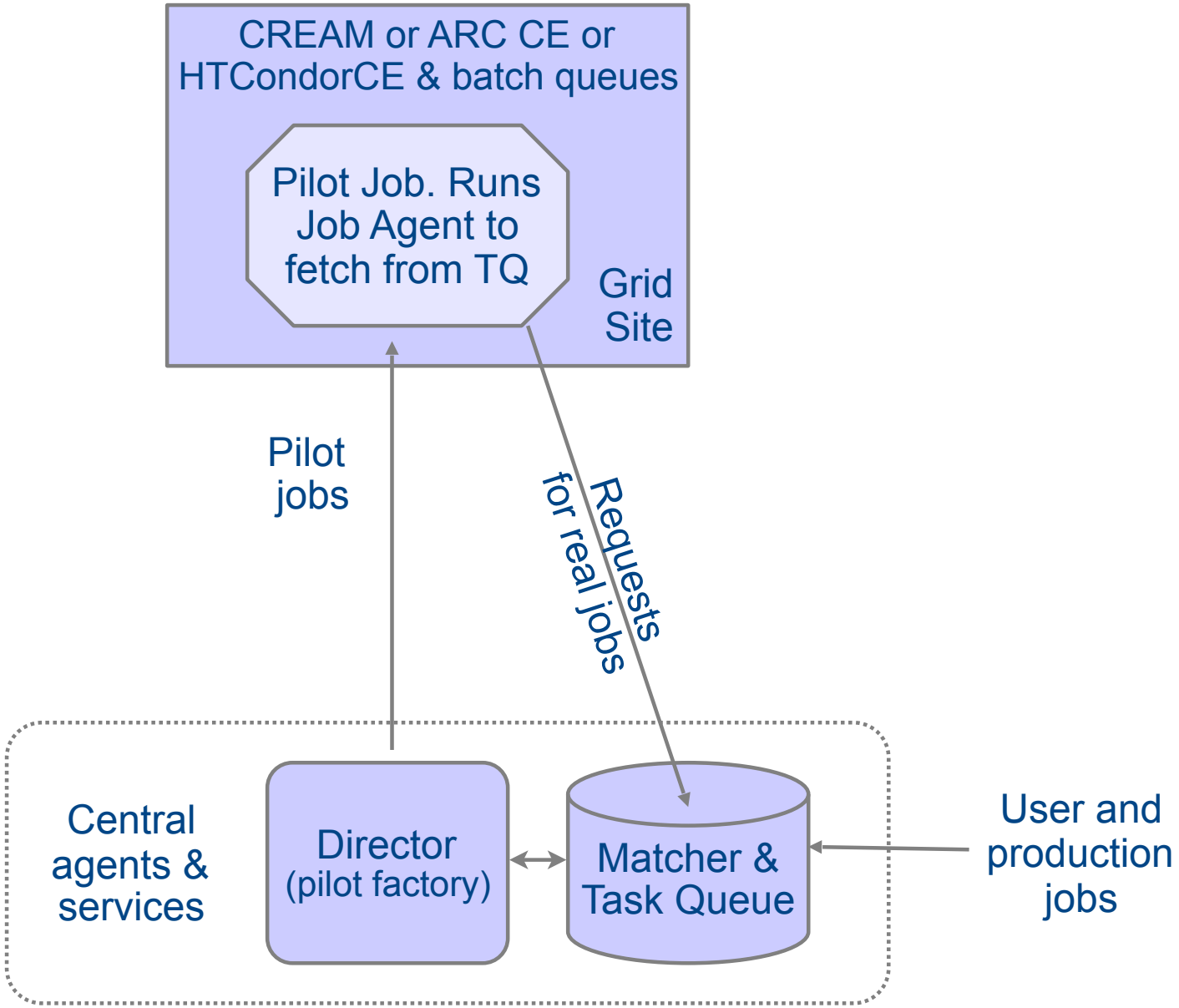
Summary / next steps

- GridPP has developed the “Vacuum Platform” to address evolving requirements / constraints
- Running production work for ATLAS, LHCb, CMS, GridPP DIRAC VOs
- Running at sites in the UK and at CERN and CC-IN2P3
- Already set up on EGI GOCDB, APEL and GGUS.
- Currently HEP/WLCG focused but looking at expanding beyond this

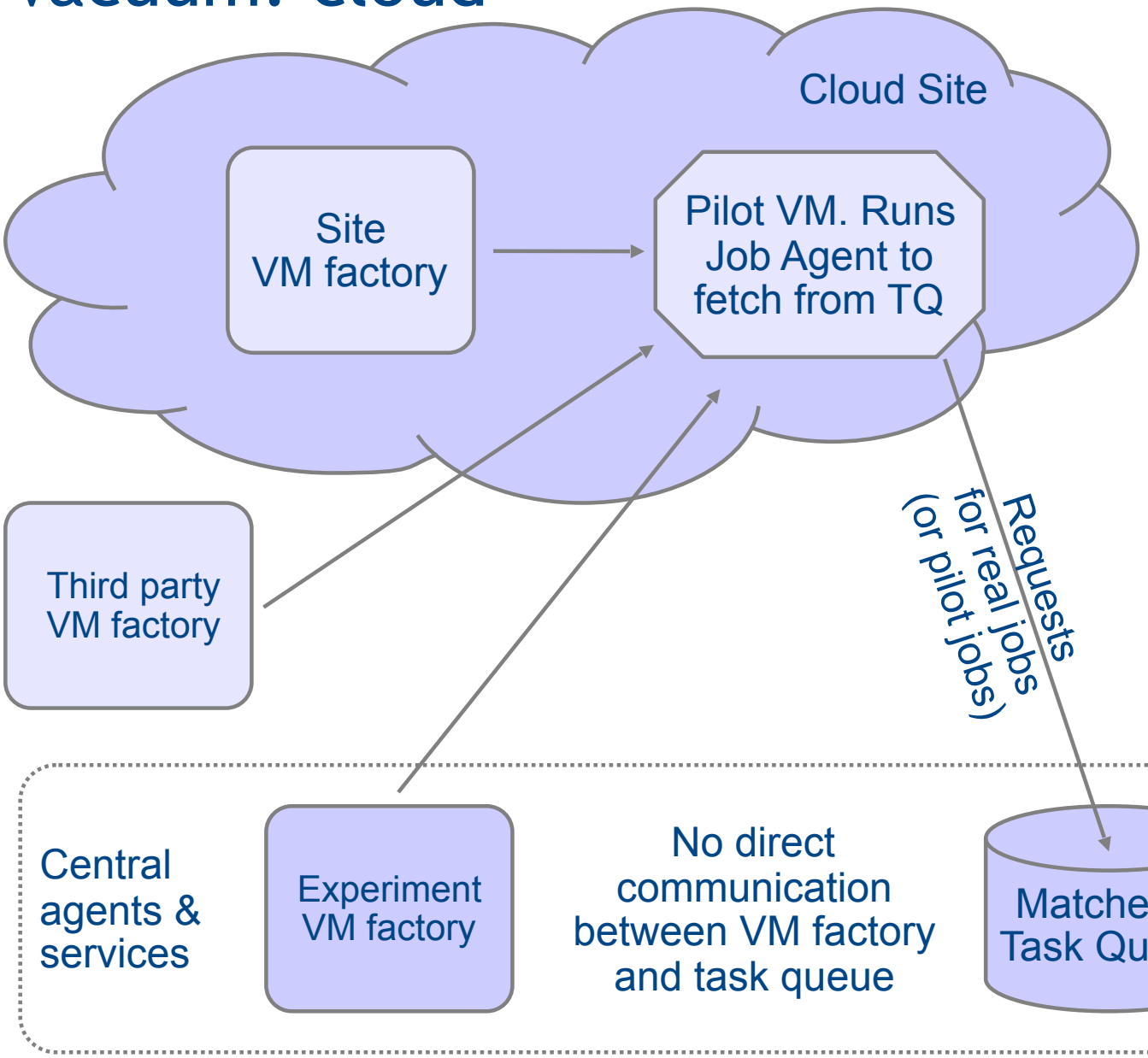
- Next steps include looking at “missing” components
- Have submitted first section of the Service Design Transition Package template with view to becoming an EGI Community Platform
- Looking for feedback on how to proceed

Extra slides

The Grid with Pilot Jobs



Vacuum: cloud



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.