The CMS openstack, opportunistic, overlay, online-cluster Cloud (CMSooooCloud)

J.A. Coarasa
CERN, Geneva, Switzerland

for the CMS TriDAS group.

CHEP2013, 14-18 October 2013,
Amsterdam, The Netherlands

Outline

- Introduction
  - The CMS Online Cluster
  - Opportunistic Usage
- The Architecture
  - Overlay in detail
  - Openvswitch-ed
  - OpenStack infrastructure
- Onset, Operation, Outlook and Conclusion

Introduction

- The CMS Online Cluster
- Opportunistic Usage

The CMS Online Cluster

A large

- More than 3000 computers
- More than 100 switches (>7000 ports)

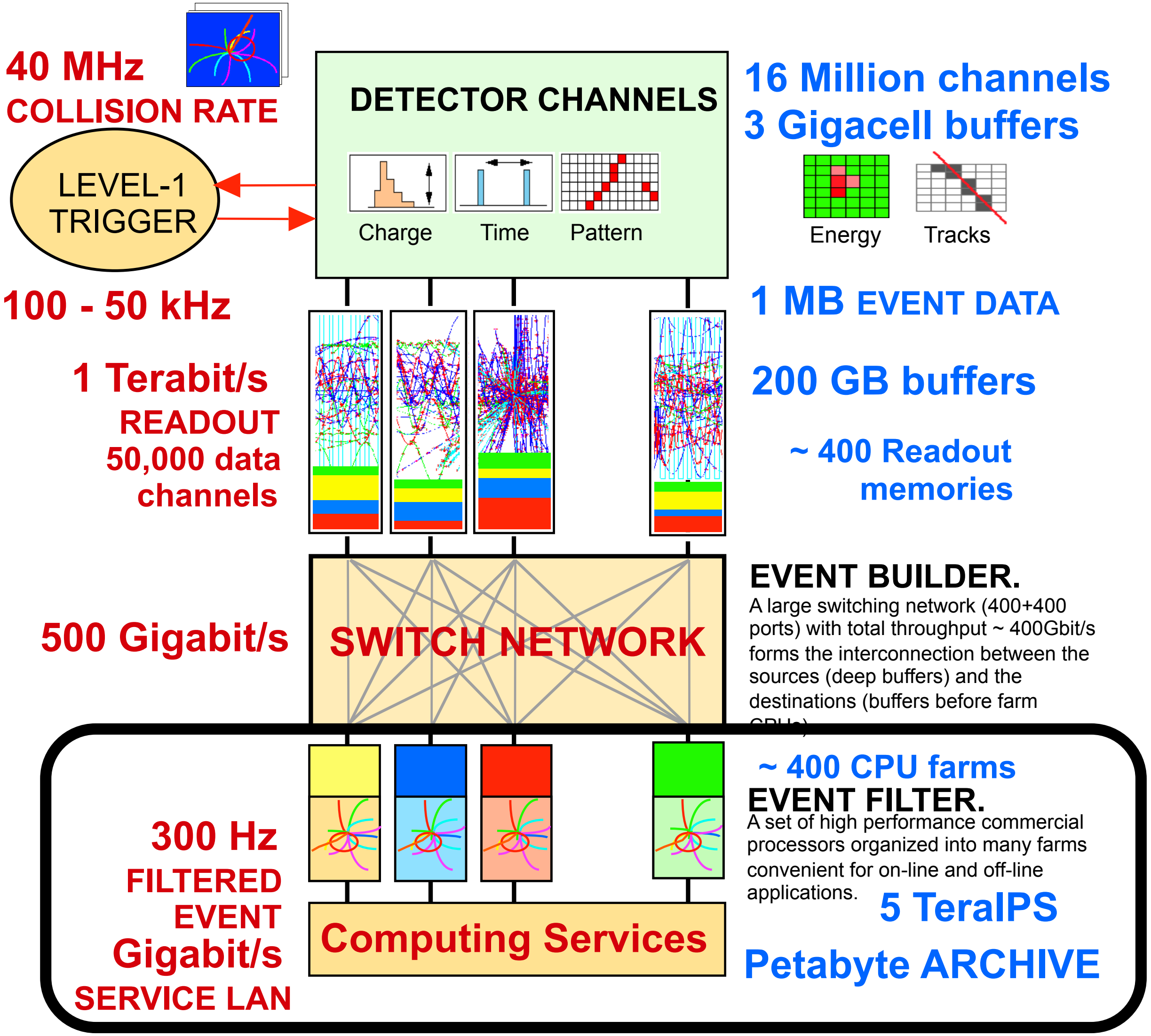
and complex cluster†

- Different kinds of hardware and ages
- Computers configured in more than 100 ways
- Set of segmented networks using sometimes VLANs
  - 1 network per rack
  - Up to 2 Additional networks in VLANs in some racks

designed as a data acquisition system to process data at 100GBytes/s and select and archive 20TBytes/day

†*The CMS online cluster: Setup, operation and maintenance of an evolving cluster.* **PoS ISGC2012 (2012) 023**.

The CMS Data Acquisition System



Cluster with ~1300 computers doing event filtering

The CMS High Level Trigger (HLT) Cluster

cluster	Nodes	cores (HT on)/ node	cores	Memory (Gbyte/node)	Disk (Gbytes/node)
(1)	720	8	5760	16	72
(2)	288	12 (24)	3456	24	225
(3)	256	16 (32)	4096	32	451
(1)+(2)+(3)	**1264**		**13312**	**26 Tbytes**	**227 Tbytes**

- Three generations of hardware, some with limited local storage
- Nice connectivity to CERN GRID Tier 0 were data is stored (~20Gbit/s, can be increased to 40Gbit without a large investment).

The HLT Clusters versus Tier[0,1,2]

CPU in HEP-SPEC06†

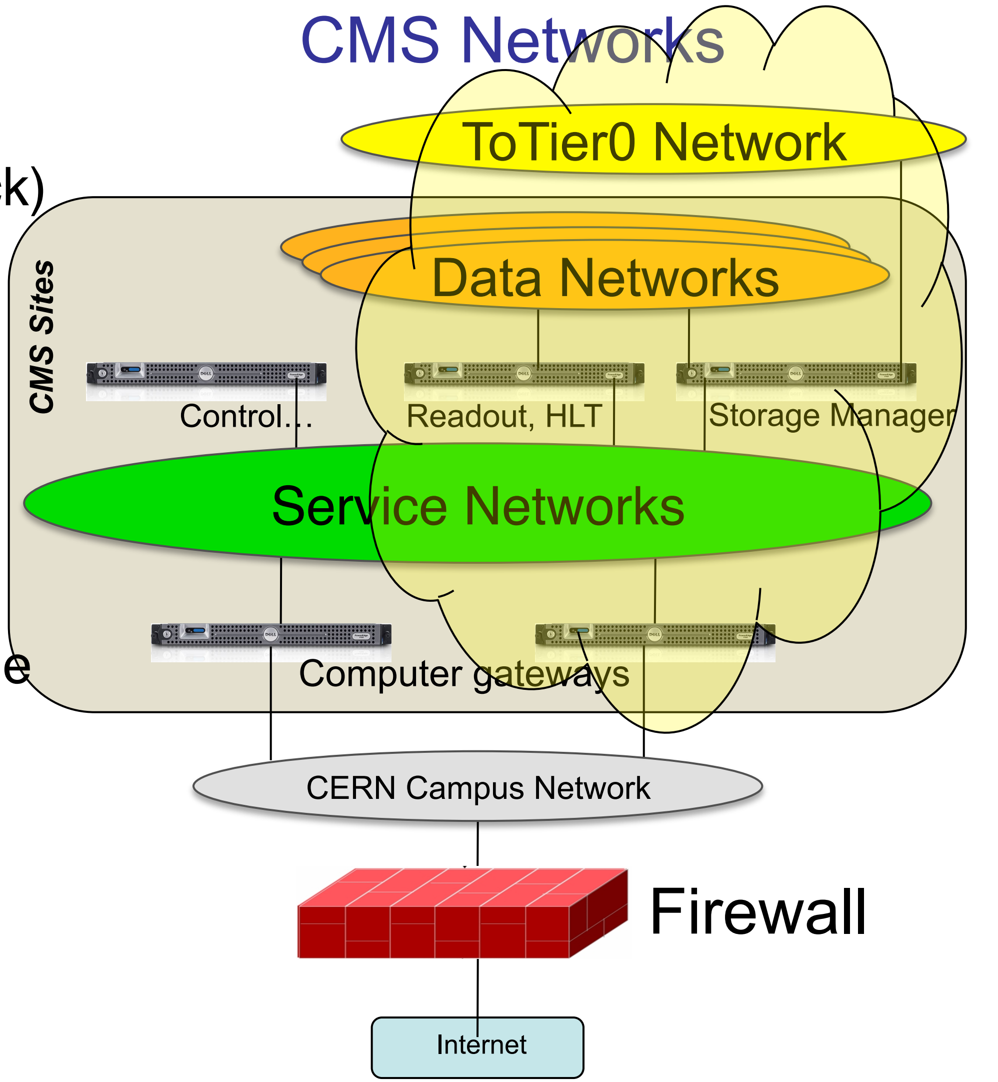
	HLT farm	Tier0	Tier1	Tier2
sum	602k + ALICE	356k	603k	985k
ATLAS	197k	111k	260k	396k
CMS	**195k**	**121k**	**150k**	**399k**
ALICE		90k	101k	143k
LHCb	210k	34k	92k	47k

† http://w3.hepix.org/benchmarks/doku.php/

The CMS Online Cluster: Network Details

CMS Networks:

- Private Networks:
  - Service Networks (per rack) (~3000 1 Gbit ports);
  - Data Networks (~4000 1Gbit ports)
    - Source routing on computers
    - VLANs on switches
  - To Tier 0 Network.
  - Private networks for Oracle RAC
  - Private networks for subdetectors
- Public CERN Campus Network



Opportunistic Usage

The cluster can be used when not 100% in use:

- During the technical stops (~1 week every 10):
  - These timeslots already used during data taking to set up the cloud infrastructure and test it;
- During the shutdown used to upgrade the accelerator (since Mid-February for more than a year)
- When the cluster is under-used:
  - Already used simultaneously while taking data (Heavy Ions, pPb 2013) as a proof of concept;
  - This needs cautious testing and deeper integration to allow dynamical resources reallocation;
  - Technically feasible but may arise concerns about putting in danger data taking.

The Architecture in the two scenarios (prove of concept and production)

- Requirements
- Overlay in detail
- OpenStack infrastructure

Requirements

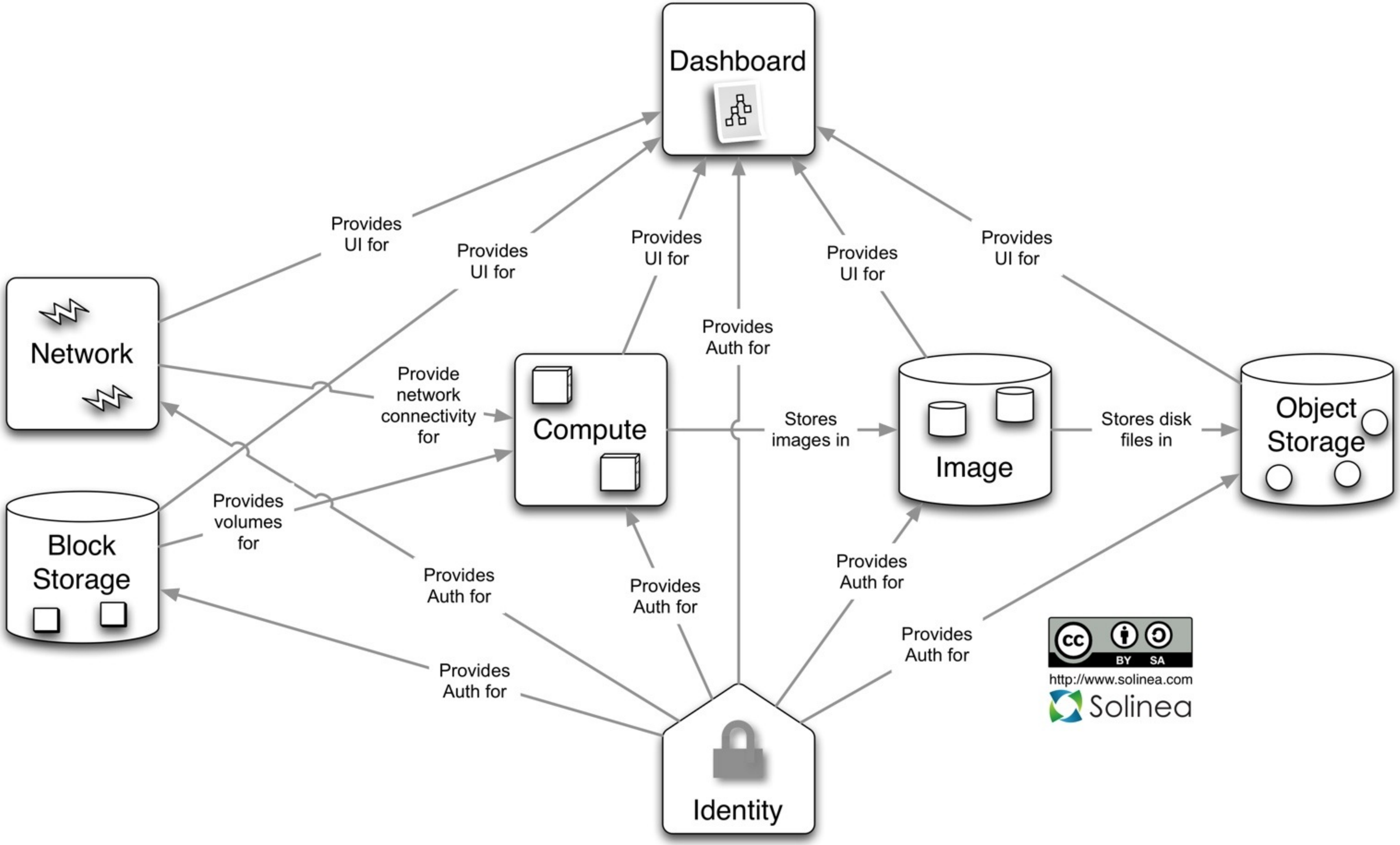
- Opportunistic usage (the motivation! Use the resources!).
- No impact on data taking.
- Online groups retain full control of resources.
- Clear responsibility boundaries among online/users (offline groups).

⇒Virtualization
⇒Overlay Cloud

Cloud Architecture with Openstack

- Auxiliary infrastructure
  - Database Backend (MySQL…)
  - Message passing system (RabbitMQ…)
- Openstack components
  - User frontends to control the cloud
    - Command line API (nova, EC2…)
    - Web interface (Dashboard)
  - VM disk image service (glance)
  - Network virtualization (nova-network, neutron…)
  - Compute virtualization (nova-compute)
  - Identity (keystone)
  - Storage virtualization

Openstack Components



Overview of Controlling architectures

Proof of Concept Phase

- Minimal changes implemented to test the cloud concept
  - Minimal software changes on computers
  - Untouched routing/network hardware

Production Phase

- Designed for
  - High availability
  - Easy scalability
- Allowed changes on network to allow overlay network and use of High Bandwidth Network

Overlay on HLT nodes

Overlay! **Not a dedicated cloud**. The HLT Nodes add software to be compute nodes.

- **Minimal changes** to convert HLT nodes in compute nodes participating in the cloud.
  - ⇒not to have any impact on data taking
    - Losing 1 min of data is wasting accelerator time (worth ~O(1000)CHF/min)
- Easy to quickly move resources between data taking and cloud usage.

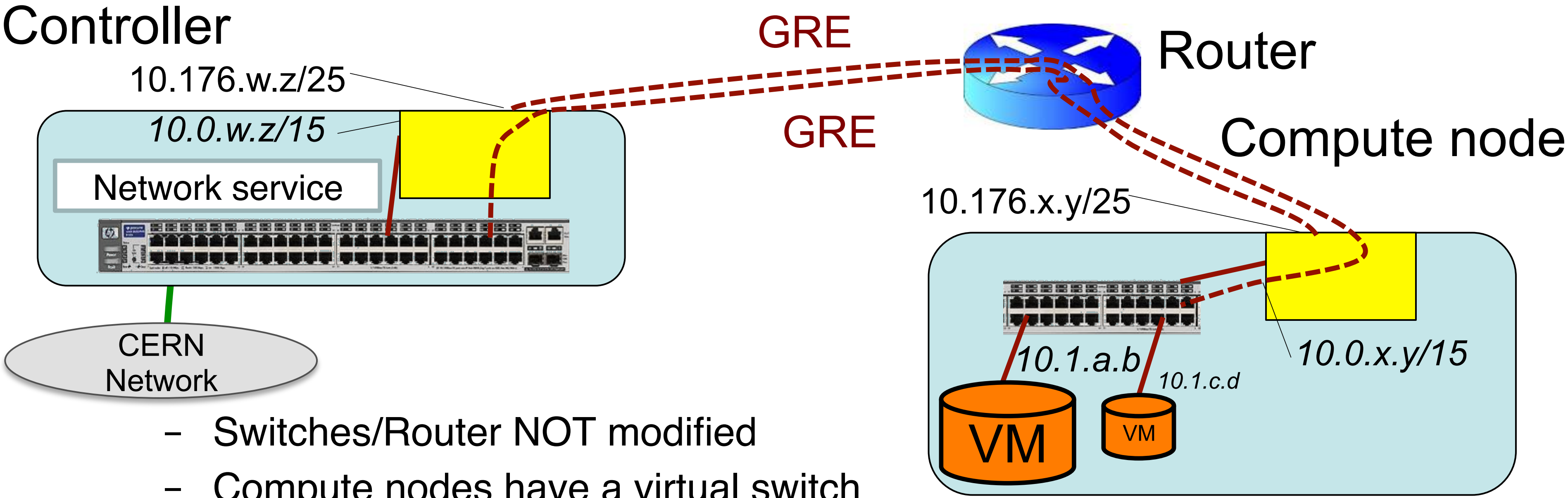
Overlay on HLT nodes. Details

- Networking
  - A virtual switch/VLAN added in the computer
- Software added
  - Libvirt, kvm
  - OpenStack[†1] compute (Essex/Grizzly from EPEL-RHEL 6)
  - OpenStack metadata-api and network (Grizzly only)
  - Open vSwitch[†2] (version 1.7.0-1) (NOT used with Grizzly)
  - RabbitMQ and MySQL clients
  - Home made scripts
    - Configure all components (and virtual network if necessary)
    - Clean up leftovers if necessary
      - » VMs, image files, hooks to bridge…

[†1] **http://www.openstack.org**

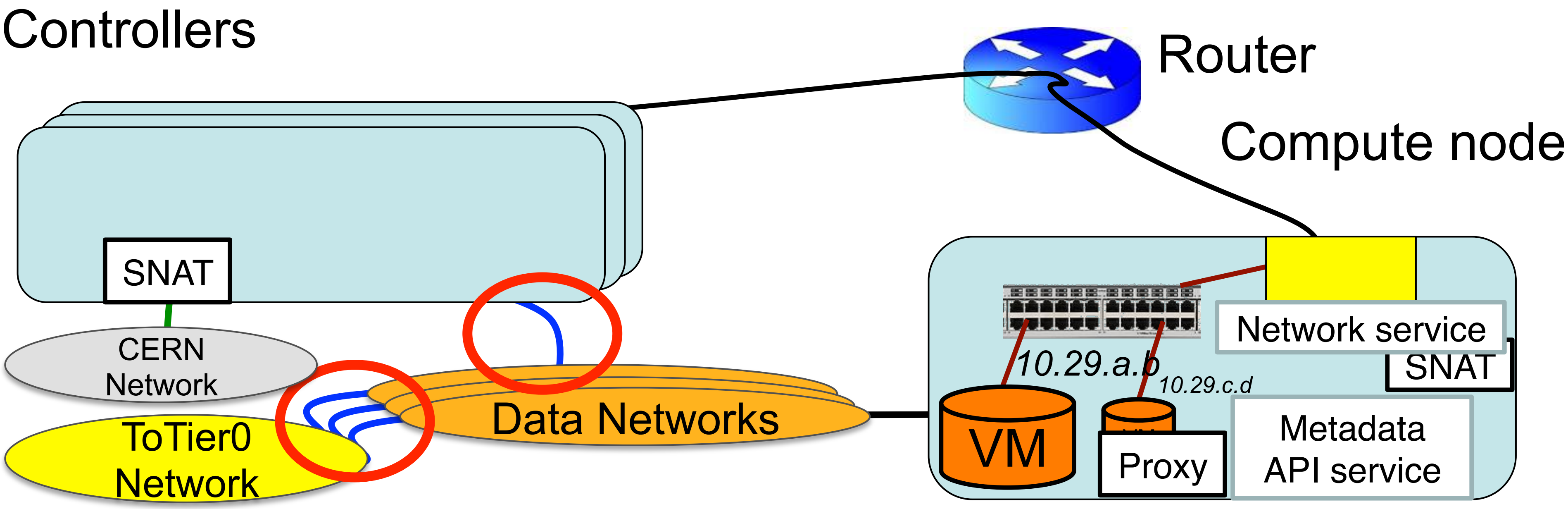
[†2] **http://openvswitch.org**

The Overlay/Virtual Network in Detail in the Proof of Concept Phase: Open vSwitch



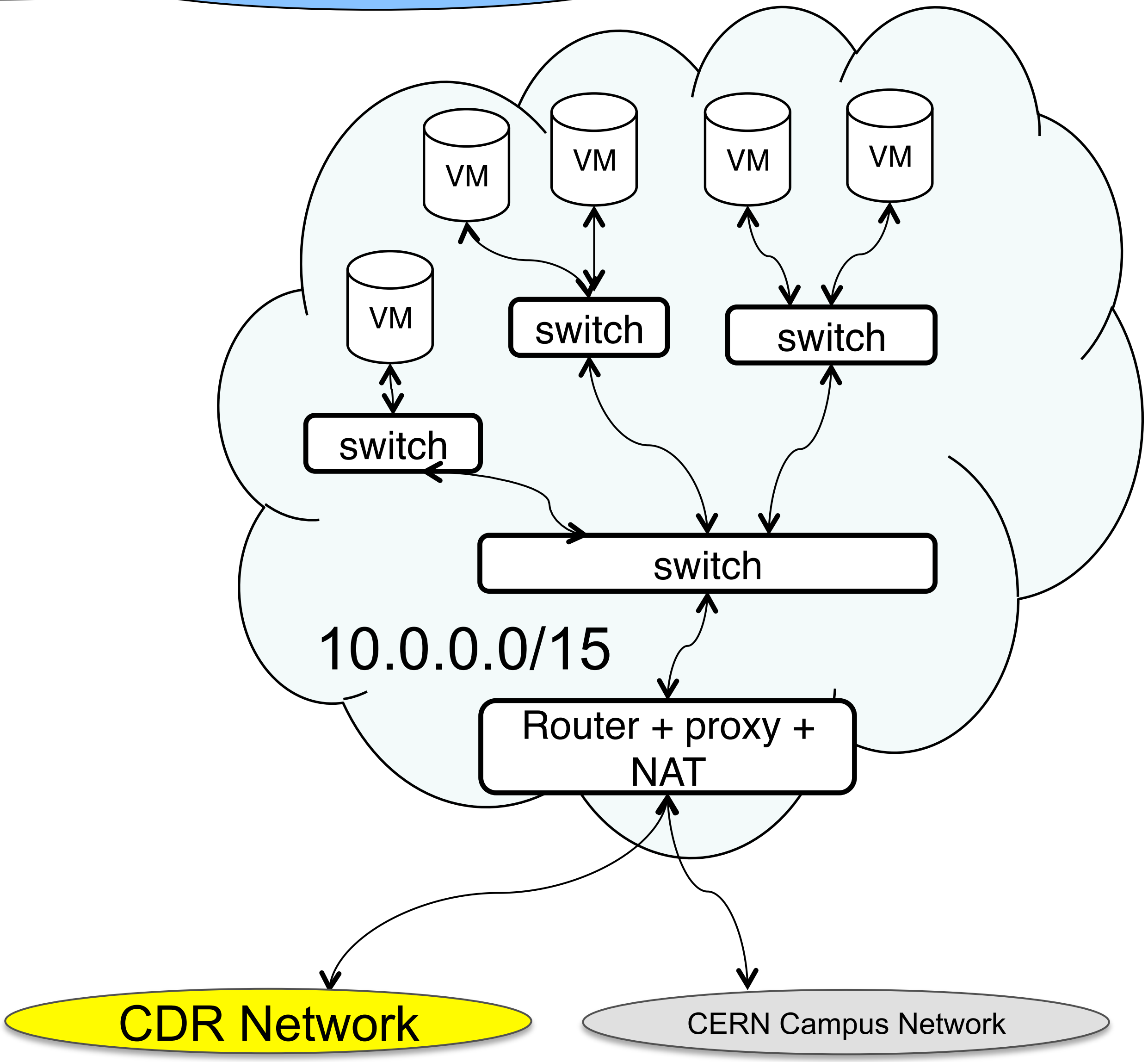
- Switches/Router NOT modified
- Compute nodes have a virtual switch
  - Where VMs hook up using 10.1.0.0/16 **(Flat network for VMs!)**
  - Where the compute node has a virtual interface (10.0.x.y/15) and traffic is routed to the control network (10.176.x.y/25)
  - And a port is connected to a central computer control network IP encapsulating with GRE
- A central computer acts as a big switch (potential bottleneck)
  - With *reciprocating GRE ports*
  - SNATs (OpenStack Nova Network service) and routes to the internet through the CERN Network

The Overlay/Virtual Network in Detail in the Production Phase: Virtual

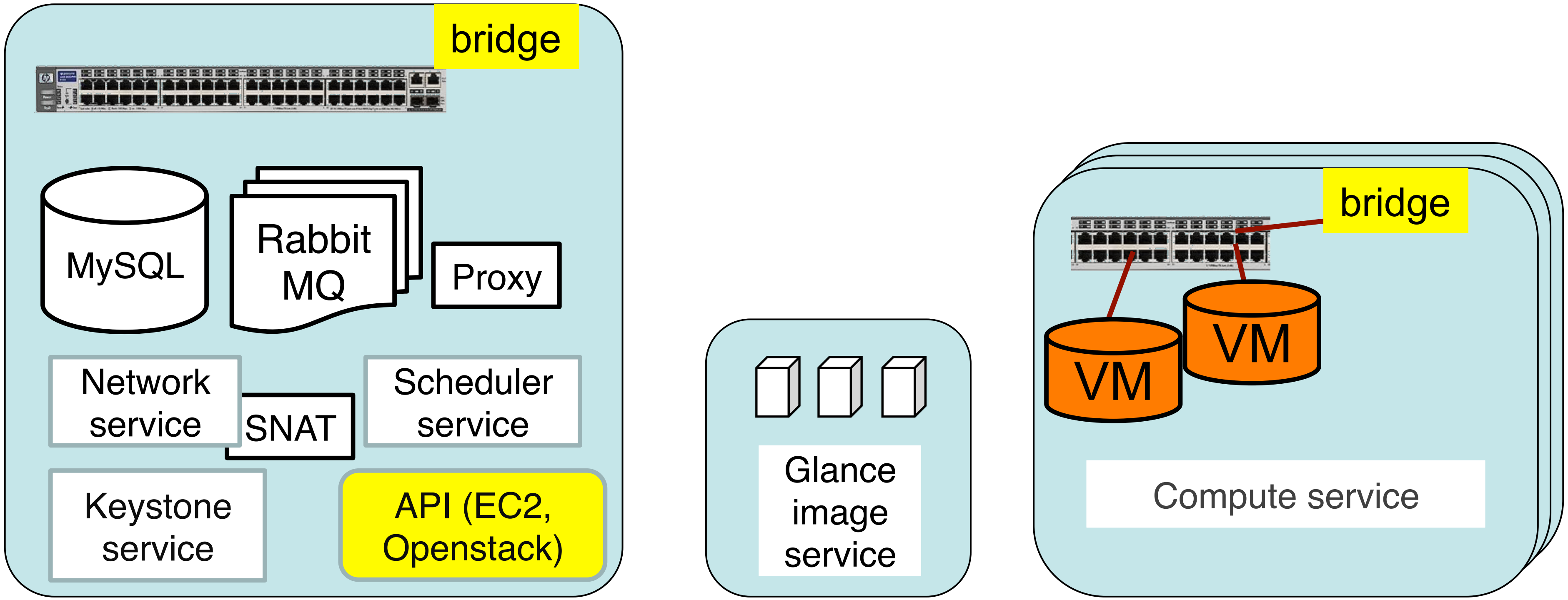


- Added cables and routing from Data Network switches to CDR and back (128.142.x.y…) and VLANs fencing the traffic of the VMs
- Added routing tables and iptables to hypervisor
  - VMs talking to 128.142.x.y will talk through the Data Networks
    - SNAT (OpenStack nova network provided with the *multihost with external gateway* configuration)
  - VMs talking to cmsproxy.cms:3128 will talk to the hypervisor:3128
    - DNAT (to make access to frontier scalable profiting from online infrastructure)
- The controllers in RR manner are default gateways to the other addresses
  - SNATs and routes to the internet through the CERN Network

CMS cloud: the Flat Network



The Cloud Controlling Layer in the Proof of Concept Phase

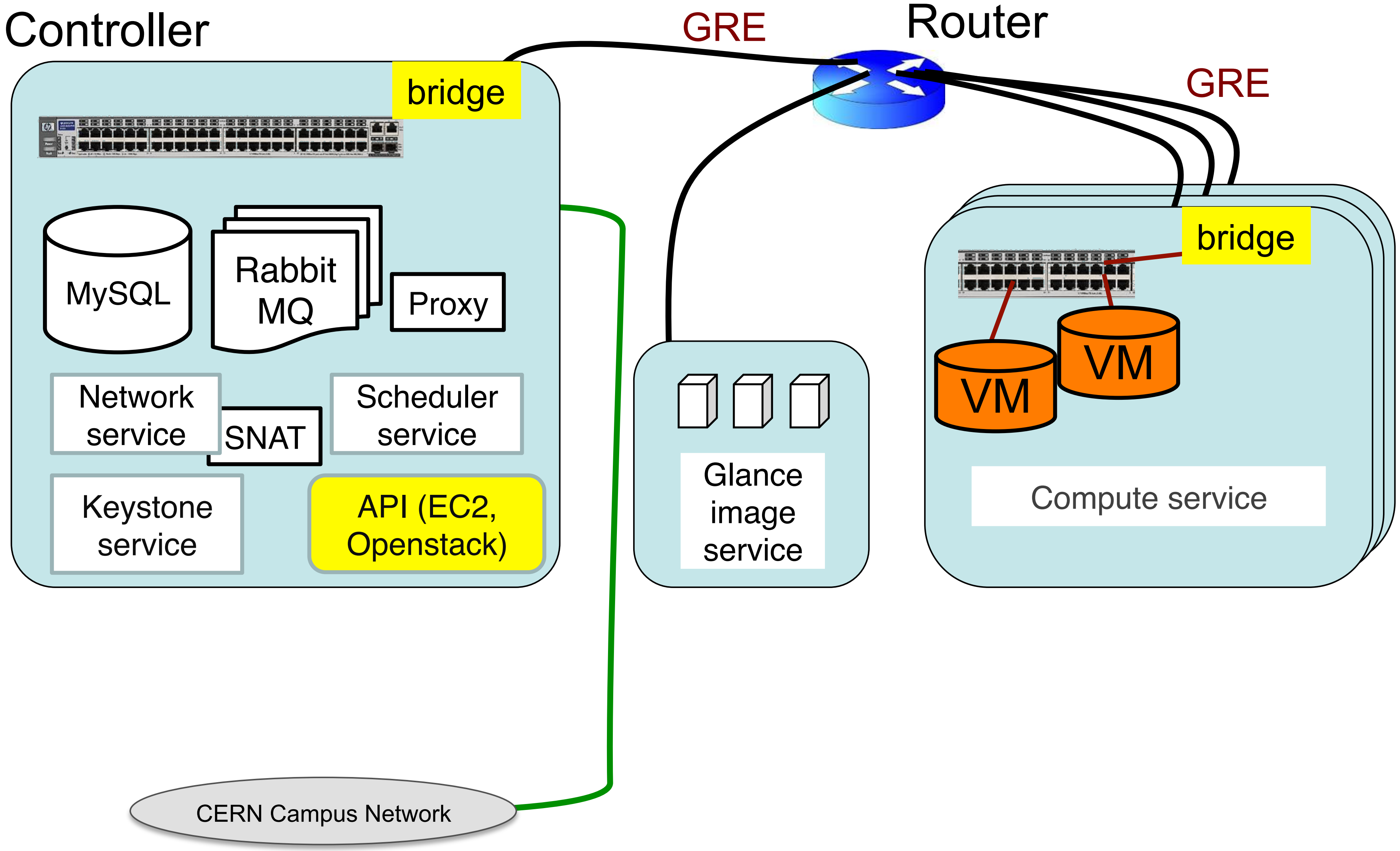


1xFat “controller” node
(Dell PowerEdge R610
48Gbytes, 8 CPU, 8x1Gbit
Ethernet)

1xVM image store

1300xCompute nodes

The CMSoCloud: Proof of Concept Phase



The Cloud Controlling Layer in the Production Phase

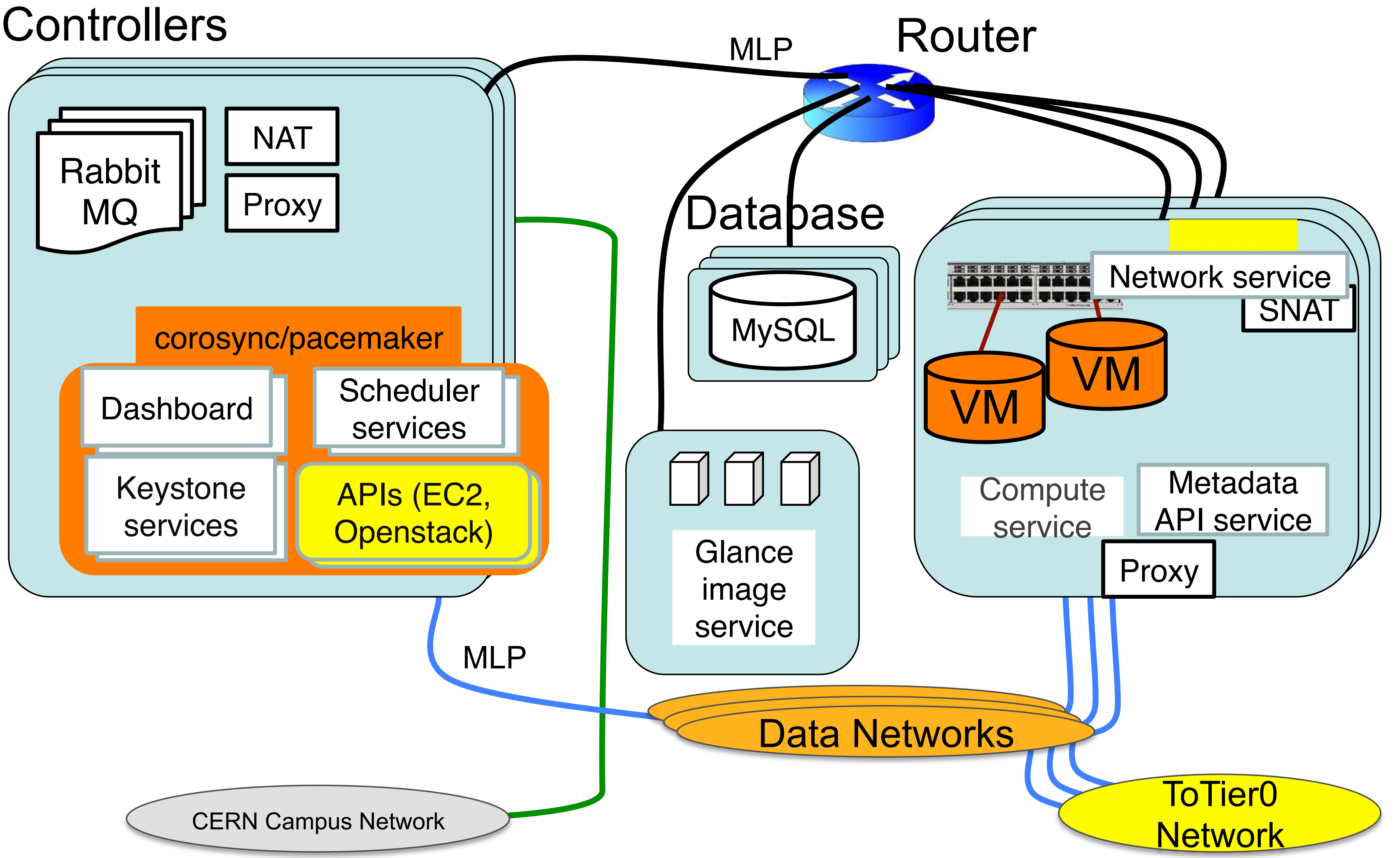
Purpose:

- Highly available
- Easy to scale out (services replicated)

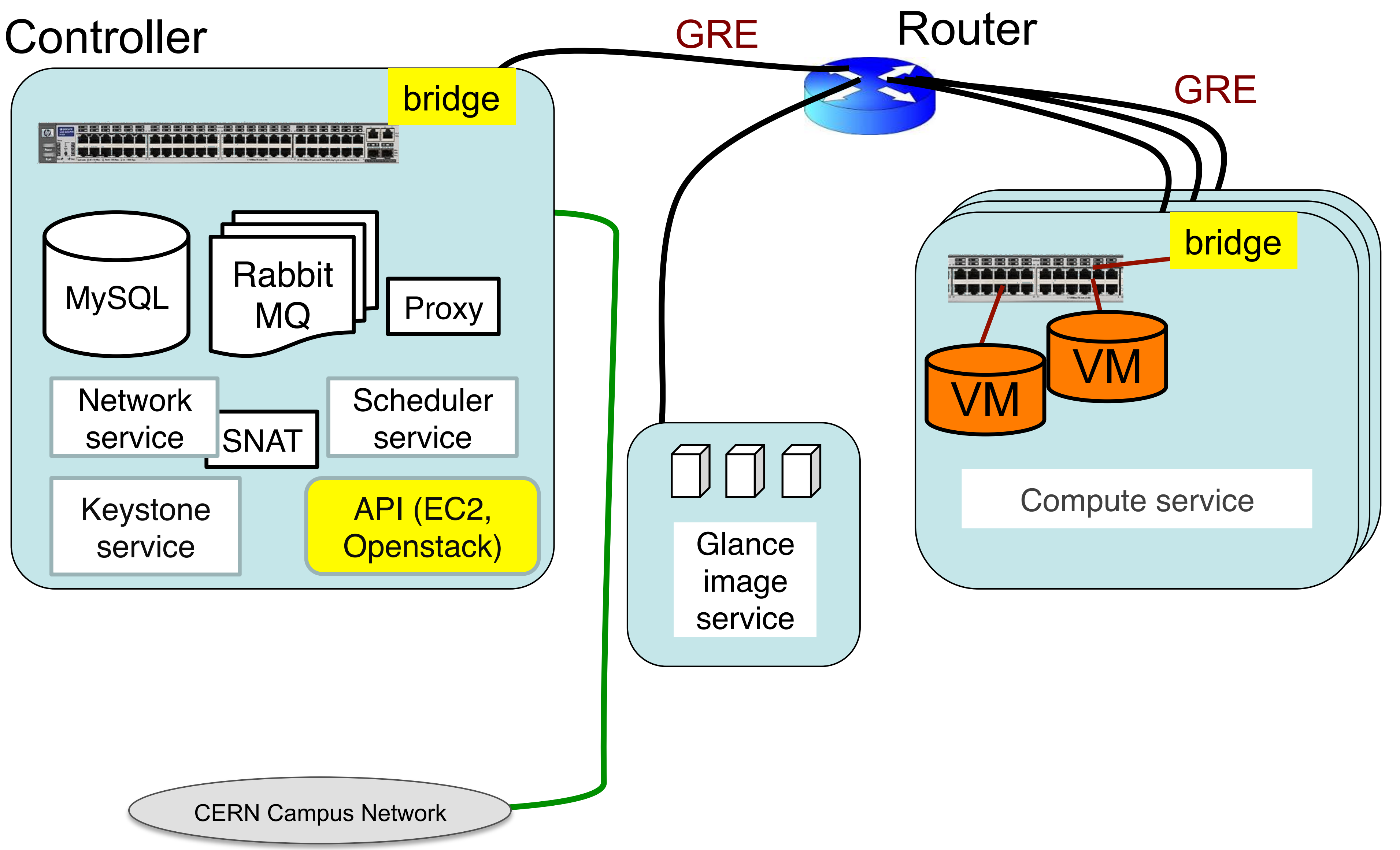
The Solution:

- MySQL cluster as a backend DB (N headnodes in Round Robin alias (RRa) and RAID 10 for the disk nodes)
  - Needed changes to the table definitions
- The rest of the services come in a *brick* (N instances)
  - RabbitMQ in cluster with replicated queues (RR order in conf. files)
  - Gateway virtual IP (VIP) for VM (RR conf. files)
  - SNAT to the outside world for the VMs
  - Corosync/pacemaker to provide OpenStack HA
    - APIs with VIP (RRa)
    - Keystones with VIP (RRa)
    - schedulers
    - Dashboard

The CMSoCloud: The Production Phase



The CMSoCloud: Proof of Concept Phase



Operating Architecture for The GRID, our first client as IaaS

The HLT clusters aim to run jobs as a GRID site

- A dedicated Factory in CERN IT instantiates VMs of the specific flavor from the specific VM image.
  - In CMS A dedicated VM image has been created.
  - The factory uses condor to control the life of the VM through ec2 commands.
- CVMfs is used:
  - To get the proper Workflow;
  - To get the proper cmssw software.
- Frontier is used. The controller is a frontier server.
- Xrootd is being used to stage in/out files of the cluster
  - A patched version due to bugs in staging out.

Onset, Operation, Outlook and Conclusion

Onset and Evolution of CMS online cloud: Growing on the Technical Stops

- July 2012: Deployment of first OpenStack infrastructure.
- 17-18/9/2012: HLT Cluster *cloudified*/migrated to SLC6.
- 8-12/10/2012: First tests of big scale VMs deployment. One of the Largest(?) OpenStack cloud in service.
  - We run the Folding@home project.
- Mid December 2012: First working image and revamped hardware for the controller/proxy.
  - We run the Folding@home project
  - We run the first cmssw workflows over Christmas.
- Cloud running since January 2013 (if conditions permit).
  - Also simultaneously when data taking on the heavy ions runs.
- June 2013. Cloud using the 20Gbit links.
- August 2013. Fully redundant scalable grizzly deployed.

CMSooooCloud Achievements

- No impact on data taking.
  - Nor during the setup phase.
  - Neither during data taking on Heavy Ion runs.
- Controlled ~1300 compute nodes (hypervisors).
- Deployed to simultaneously run ~3500 VMs in a stable manner.
- Deployed ~250 VMs (newest cluster) in ~5 min if previously deployed (cached image in hypervisor).
- Move resources to be used or not by the cloud in seconds.
- Able to run more than 6000 jobs simultaneously.
- The man power dedicated to *cloudify* the HLT cluster was low (~1.5 FTE or less for ~6 months) for the potential offered.

Outlook of CMSoooCloud

- Continue the operation as a GRID site.
- Integrate OpenStack with the online control to allow dynamic allocation of resources to the Cloud
- Interoperate with CERN IT's OpenStack Cloud, as a connected cell/zone.
- Use it for CMS online internal developments to benefit from the lower thresholds from idea to results.

CMSooooCloud: Conclusions

An overlay Cloud layer has been deployed on the CMS online High Level Trigger cluster with zero impact on data taking. One of the largest OpenStack clouds.

We shared the knowledge on how to deploy such an overlay layer to existing sites that may transition to cloud infrastructures (ATLAS online, Bologna, IFAE).

We gained experience on how to contextualize and deploy VMs on the cloud infrastructures, that are becoming commonplace, to run the GRID jobs.

We were able to run different kind of GRID jobs and non GRID jobs on our cloud.

Thank you. Questions?