Introduction to OpenStack

Carlo Vallati

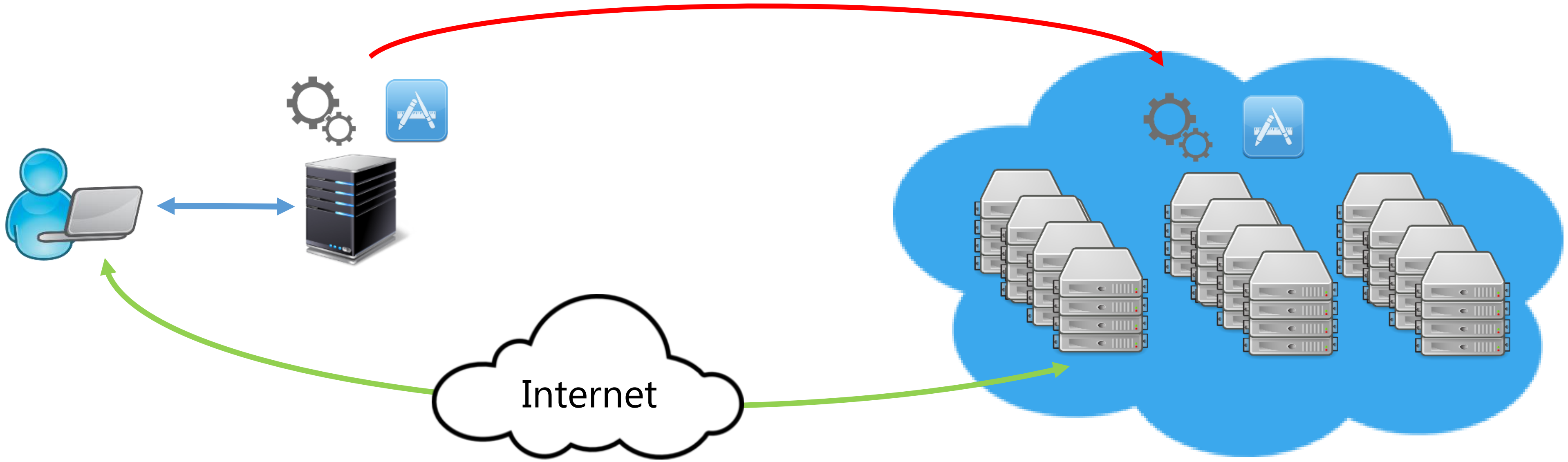
Assistant Professor

Dpt. Information Engineering – University of Pisa

carlo.vallati@iet.unipi.it

Cloud Computing - Definition

- Cloud Computing is a term coined to refer *application and services moved from local computing deployments to somewhere into the Internet "Cloud"*
- Outsourced services are accessed through the Internet using common protocols and networking standards
- "Clouds" are datacenters offering resources (computing and storage) on demand in a virtualized environment



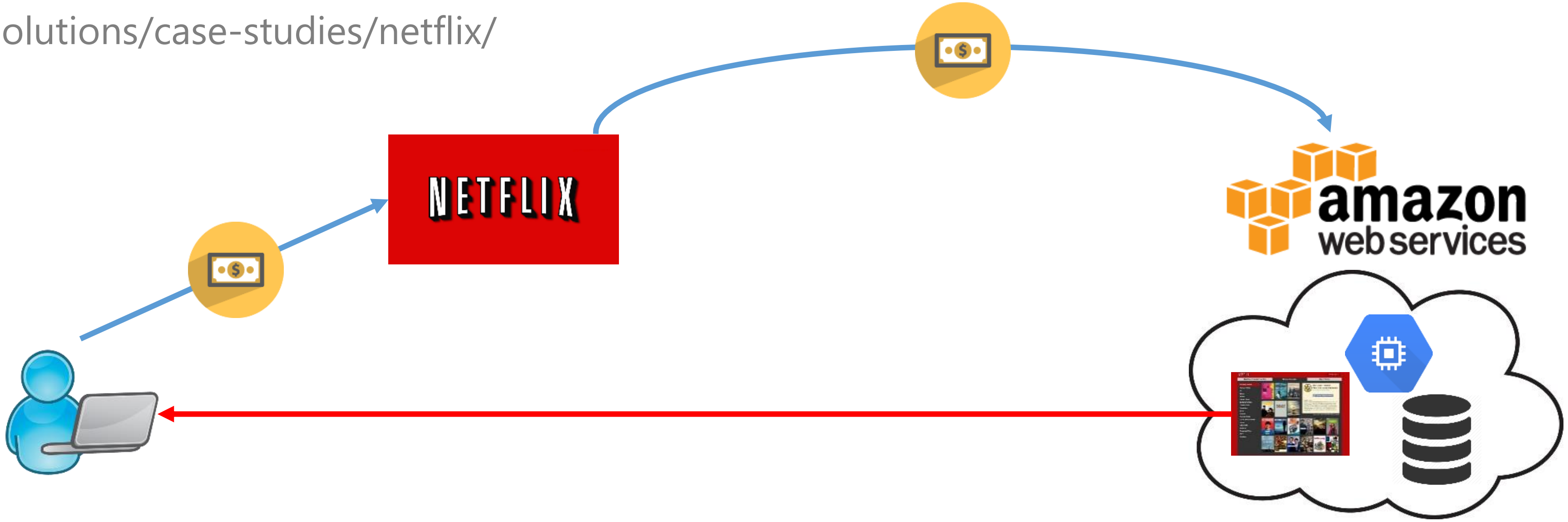
Cloud Computing – Business Model

Cloud computing *business model* is simple:

- Cloud computing (e.g. Amazon, Rackspace, Google) companies build large data-centers to sell **low-cost** and **scalable** storage and computing

- Other companies move their application and services to the cloud

Example: **Netflix**
https://aws.amazon.com/solutions/case-studies/netflix/



Cloud Computing - Advantages

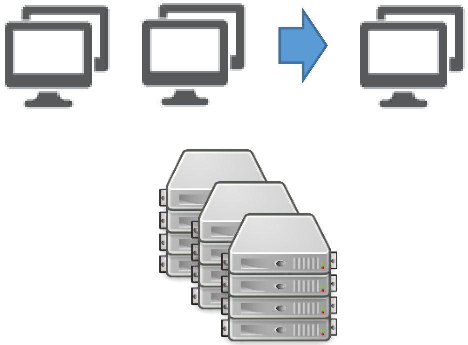
Cloud Computing paradigm in general offers a wide set of advantages for Cloud providers and end users:

1. *More efficient usage of resources*: **virtualization** enables sharing of physical services, storage and networking capabilities across different users. Such **shared infrastructure** enables multi-tenancy, making the most from the available infrastructure. This results in **lower costs** for computing and storage

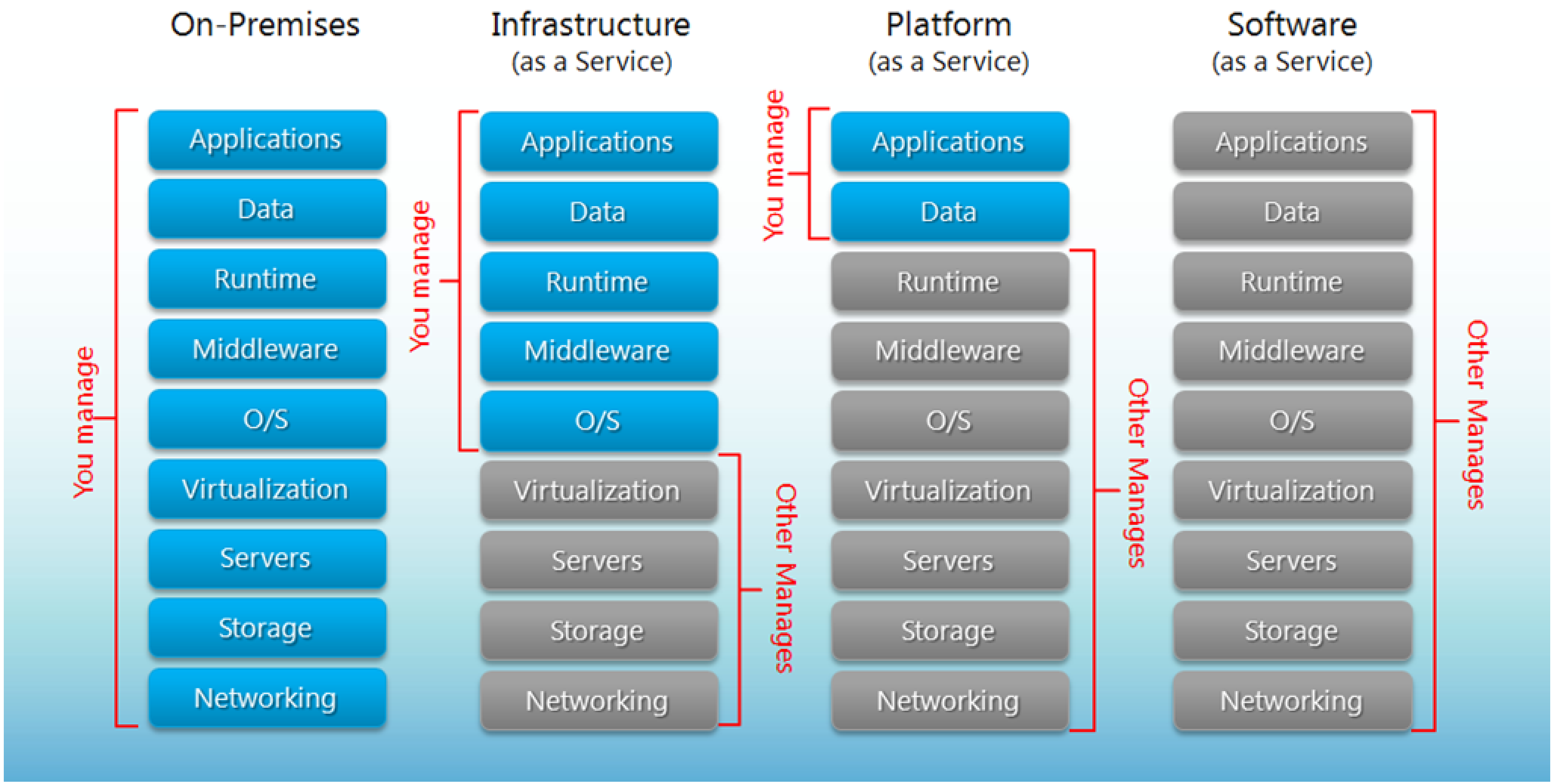
2. *High scalability*: provision of services can be based on **current demand requirements**. Such **dynamic provisioning** can be done *automatically* using software automation for dynamic scaling. This results in the possibility of dynamically expand/contract the required service

Shared infrastructure

Dynamic Provisioning



Cloud Service Models



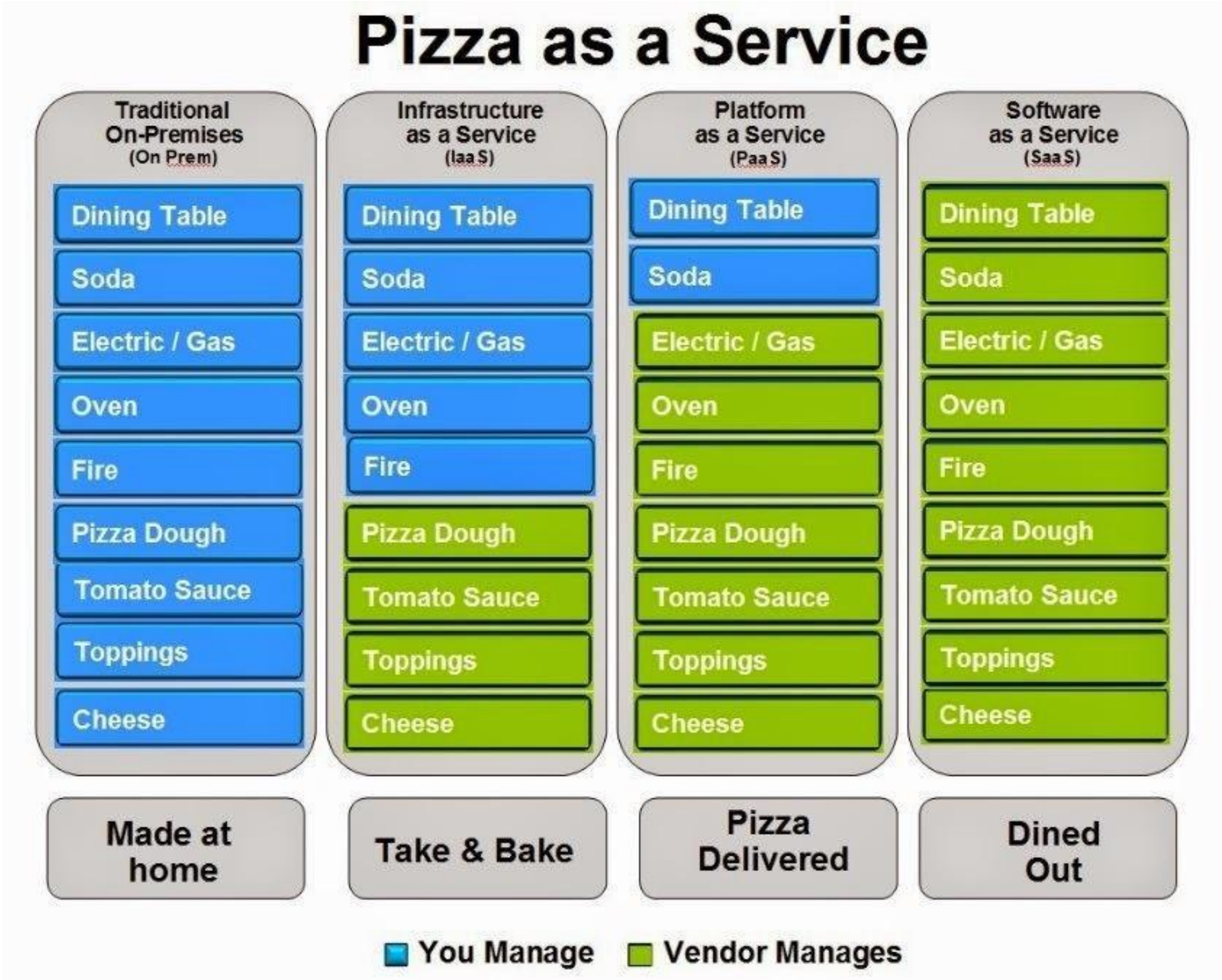
On-Premises		Infrastructure (as a Service)		Platform (as a Service)		Software (as a Service)	
You manage	Applications	You manage	Applications	You manage	Applications	Applications	Other Manages
	Data		Data		Data	Data	
	Runtime		Runtime	Runtime	Other Manages	Runtime	
	Middleware		Middleware	Middleware		Middleware	
	O/S		O/S	O/S		O/S	
	Virtualization	Virtualization	Other Manages	Virtualization		Virtualization	
	Servers	Servers		Servers		Servers	
	Storage	Storage		Storage		Storage	
	Networking	Networking		Networking		Networking	

Cloud Service Models – Definitions

- **Infrastructure as a Service**: IaaS provides virtual machines, virtual storage, virtual infrastructure, and other hardware assets as resources that clients can provision

- **Platform as a Service**: PaaS provides virtual machines, operating systems, applications, services, development frameworks, transactions, and control structures

- **Software as a Service**: SaaS is a complete operating environment with applications, management, and the user interface



Pizza as a Service

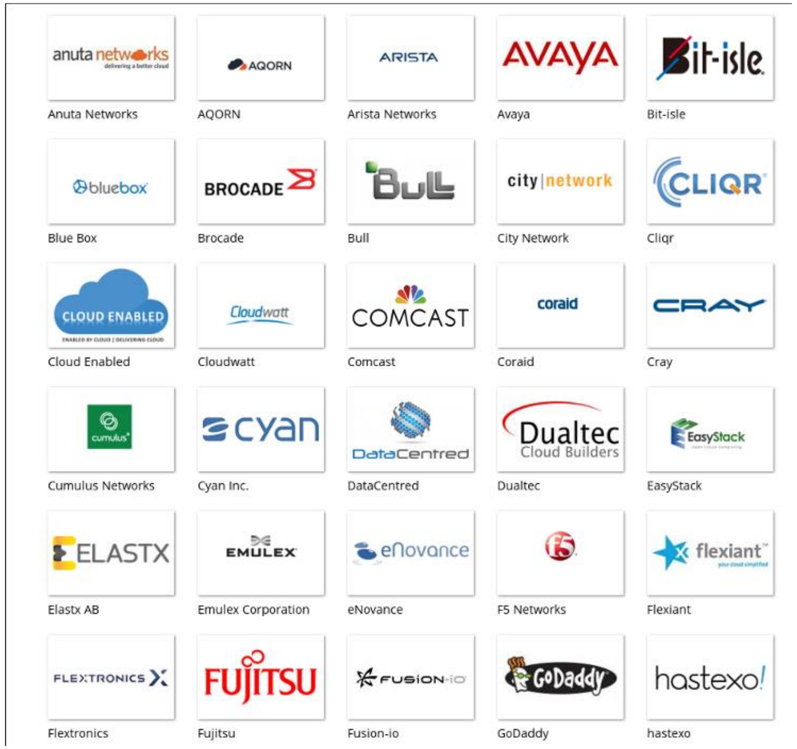
Traditional On-Premises (On Prem)	Infrastructure as a Service (Iaa S)	Platform as a Service (Paa S)	Software as a Service (Saa S)
🟦 Dining Table	🟦 Dining Table	🟦 Dining Table	🟩 Dining Table
🟦 Soda	🟦 Soda	🟦 Soda	🟩 Soda
🟦 Electric / Gas	🟦 Electric / Gas	🟩 Electric / Gas	🟩 Electric / Gas
🟦 Oven	🟦 Oven	🟩 Oven	🟩 Oven
🟦 Fire	🟦 Fire	🟩 Fire	🟩 Fire
🟦 Pizza Dough	🟩 Pizza Dough	🟩 Pizza Dough	🟩 Pizza Dough
🟦 Tomato Sauce	🟩 Tomato Sauce	🟩 Tomato Sauce	🟩 Tomato Sauce
🟦 Toppings	🟩 Toppings	🟩 Toppings	🟩 Toppings
🟦 Cheese	🟩 Cheese	🟩 Cheese	🟩 Cheese
Made at home	Take & Bake	Pizza Delivered	Dined Out

🟦 You Manage 🟩 Vendor Manages

What is OpenStack?

- Several cloud platforms are available today
- Some of them are also available as open-source
- **OpenStack** is a free open-source software platform for **IaaS cloud computing**
- Started as a joint project of *Rackspace Hosting* and of *NASA* in 2010
- Openstack today is supported and managed by the **OpenStack Foundation**, which composed by more than 500 companies (e.g. VMware, CISCO, Citrix, Oracle, Ericsson, IBM, etc)

openstack™ CLOUD SOFTWARE



Anuta Networks	AQORN	Arista Networks	Avaya	Bit-isle
Blue Box	Brocade	Bull	City Network	Cliqr
Cloud Enabled	Cloudwatt	Comcast	Coraid	Cray
Cumulus Networks	Cyan Inc.	DataCentred	Dualtec	EasyStack
Elastx AB	Emulex Corporation	eNovance	F5 Networks	Flexiant
Flextronics	Fujitsu	Fusion-io	GoDaddy	hastexo

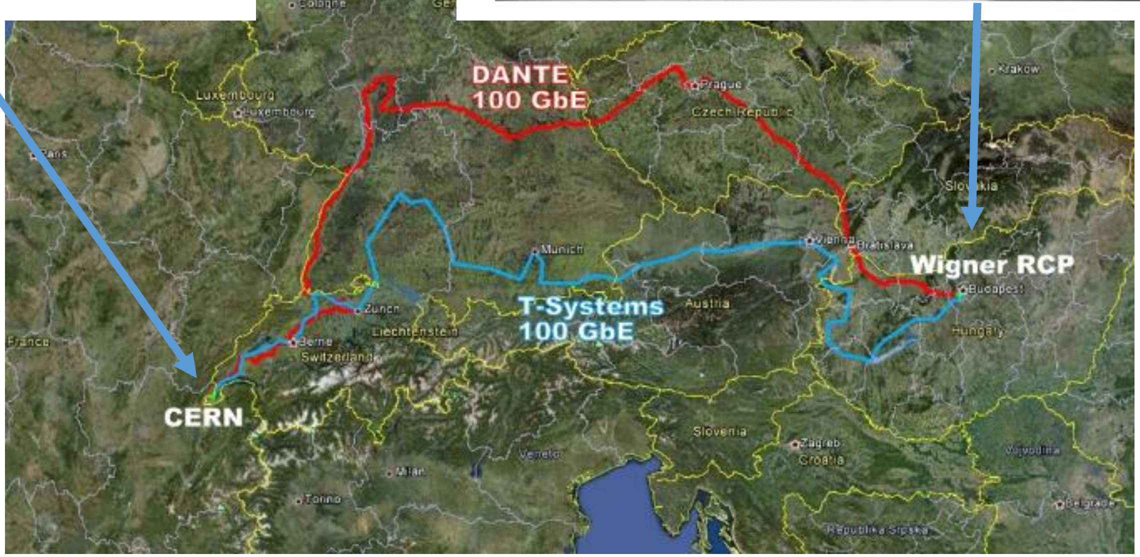
OpenStack @ CERN

CERN - Computer Center - Geneva, Switzerland

- 3.5 Mega Watts
- ~91000 cores
- ~120 PB HDD
- ~100 PB Tape
- ~310 TB Memory

CERN - Computer Center - Budapest, Hungary

- 2.5 Mega Watts
- ~20000 cores
- ~6 PB HDD

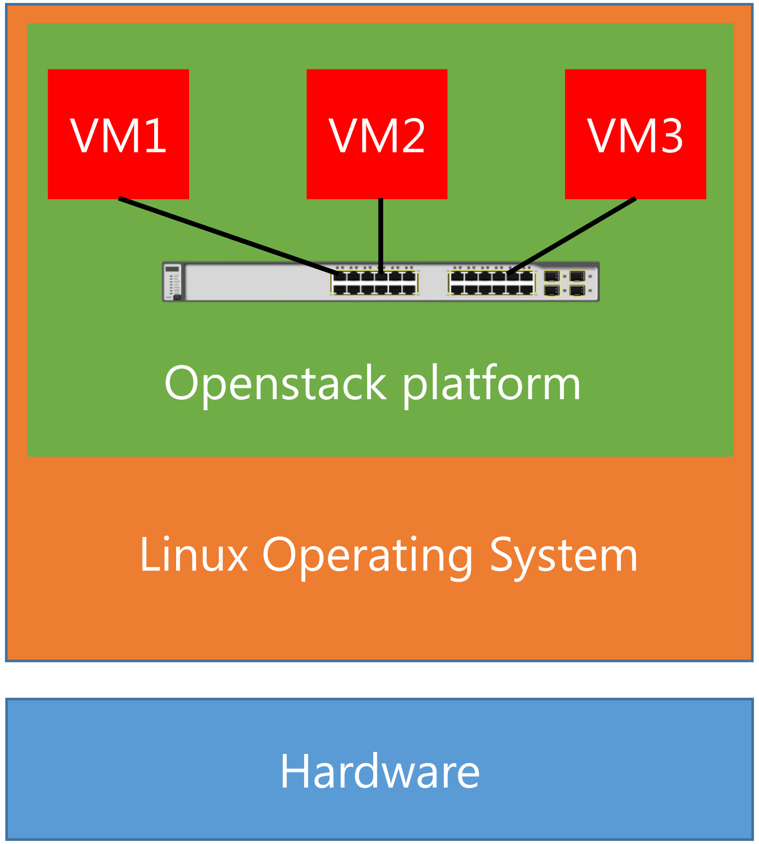


OpenStack is widely adopted today by companies to build large public/private cloud deployments.

Other User Stories:
https://www.openstack.org/user-stories/

OpenStack Software Platform

- OpenStack runs on top of commodity computers (no particular hardware is required)
- The software platform is installed and runs on top of the host operating system (e.g. Linux OS) in order to create a distributed "cloud operating system"
- Such cloud operating system support the creation of different *Virtual Machines* which can be connected through *Virtual Networks*



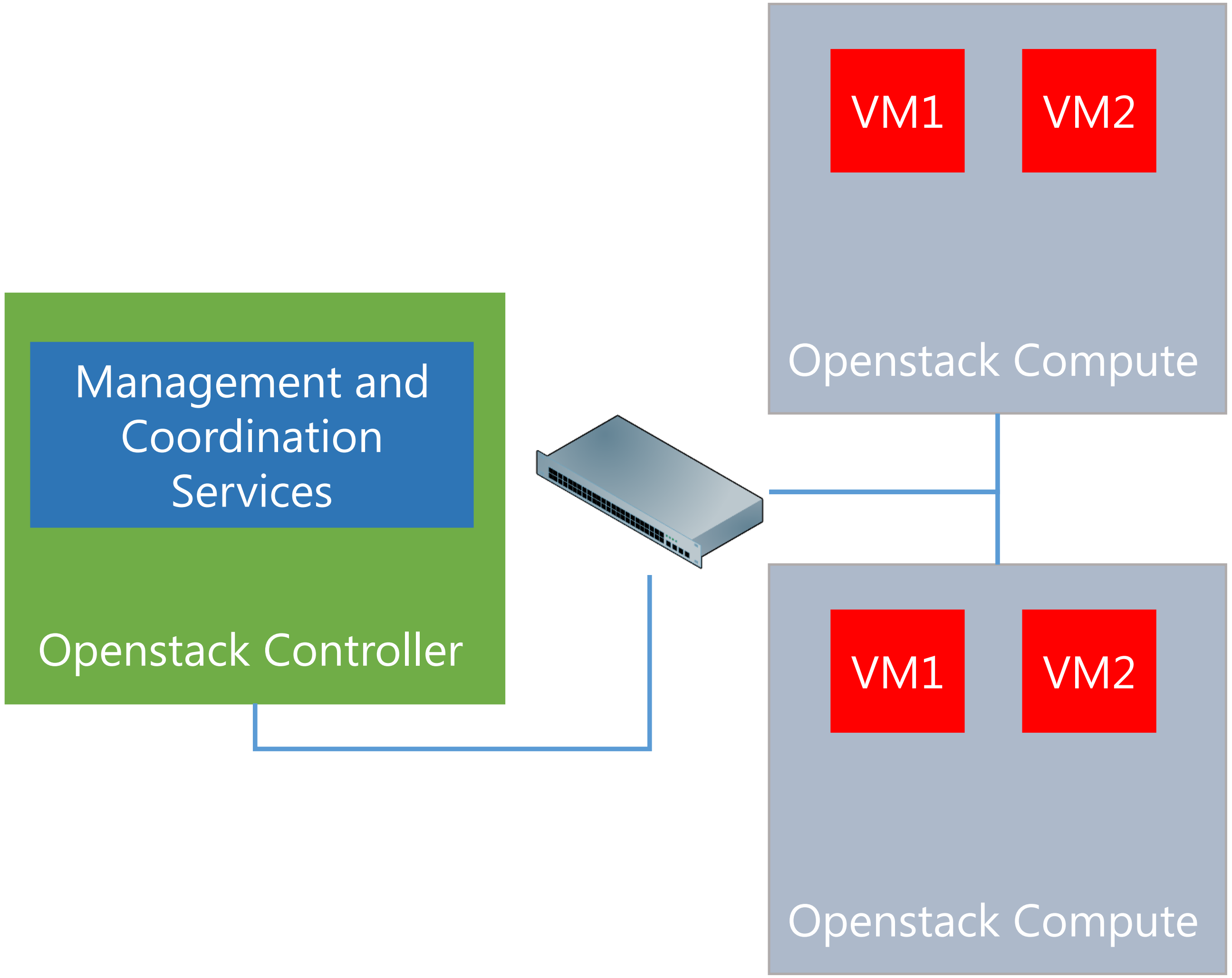
OpenStack Instance

- Nodes running the OpenStack software are configured to form a **single OpenStack instance** *combining together computing and storage*

- Nodes are usually connected through a high speed local area network

- In an instance at least a node is configure as ***controller*** which is in charge of coordinating Openstack functions and managing the resources available to the instance

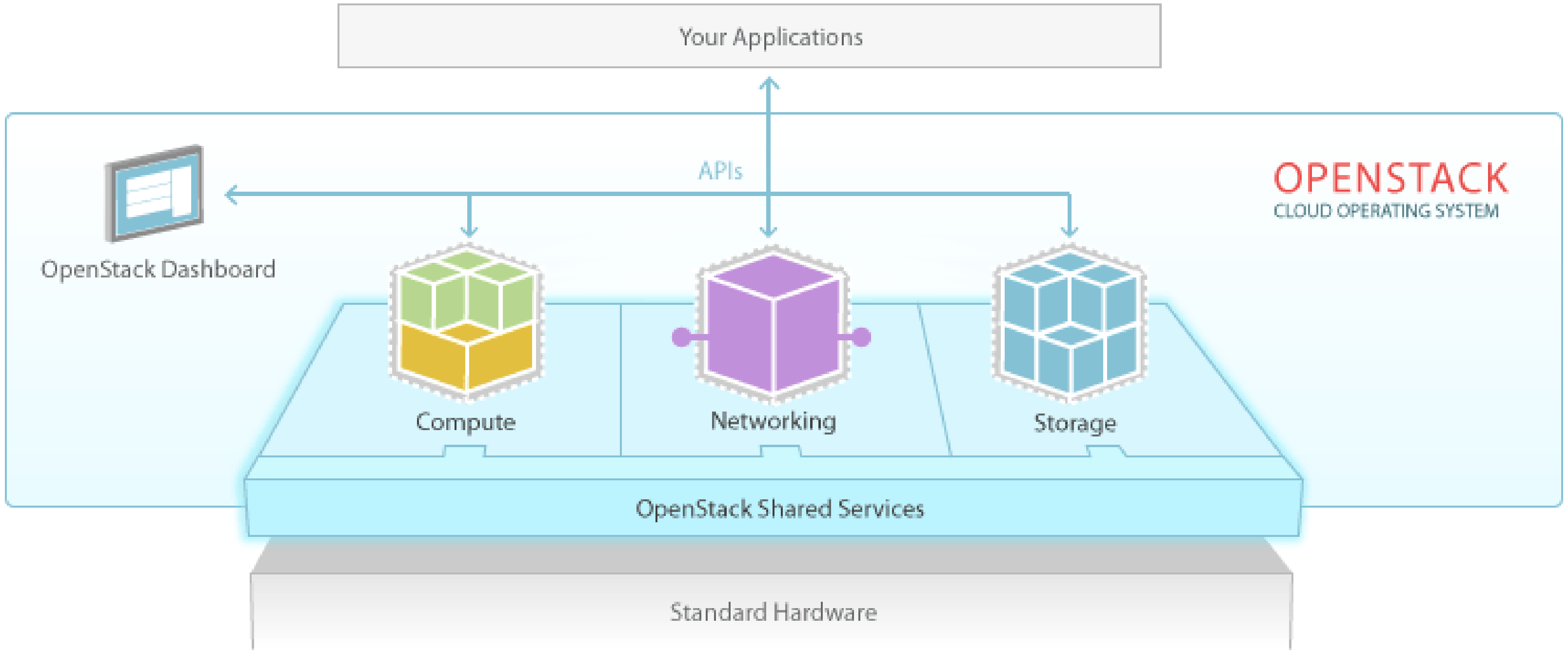
- Other nodes are configured as ***compute*** nodes that offer computation and storage resources to run virtual machines



OpenStack Architecture

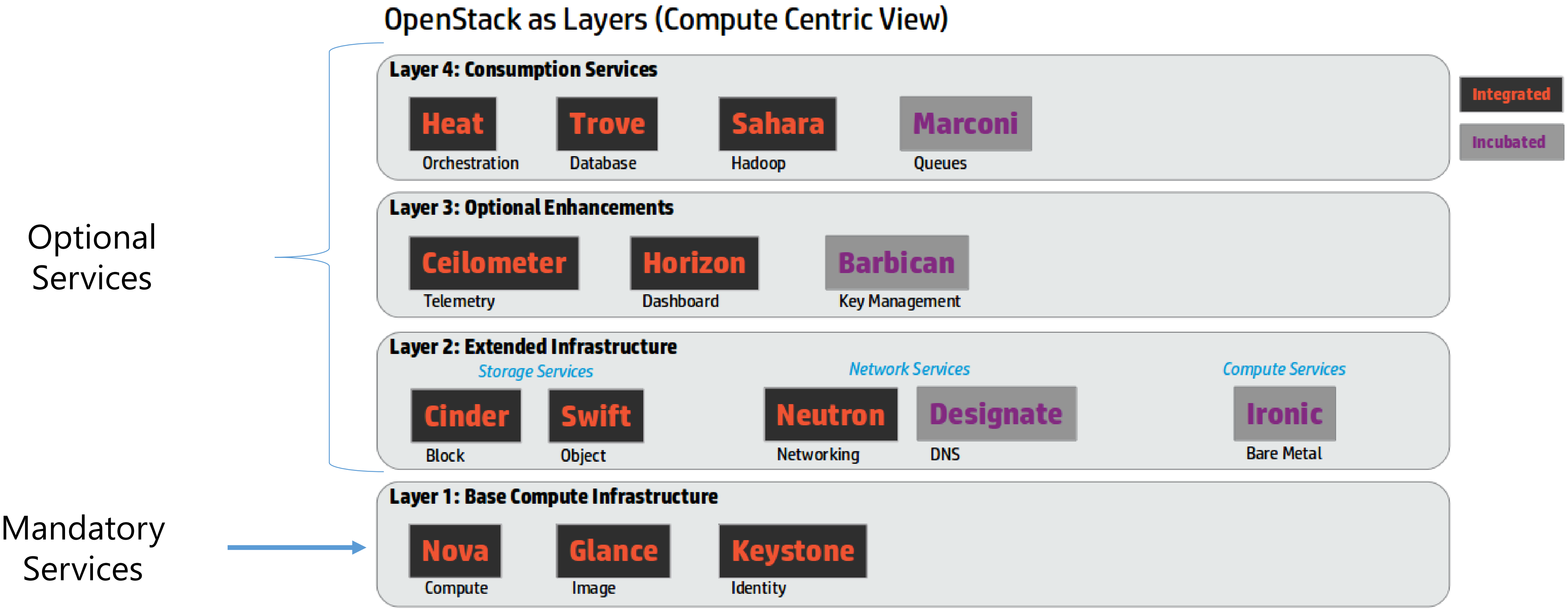
- The controller node exposes a web dashboard to allow users and administrators to manage Virtual Machines and allocate Compute, Storage and Networking to them

- Each service composing OpenStack exposes a set of REST APIs is exposed to allow automatic management directly from external applications



OpenStack Services

- OpenStack software is highly modular. Each service is provided by a different module, maintained as a separate project
- Apart from Core Services, mandatory on each installation, other services are optional and can be installed only if the provided functionalities are needed



OpenStack as Layers (Compute Centric View)

Optional Services

Mandatory Services

Integrated

Incubated

Layer 4: Consumption Services

Heat Orchestration | Trove Database | Sahara Hadoop | Marconi Queues

Layer 3: Optional Enhancements

Ceilometer Telemetry | Horizon Dashboard | Barbican Key Management

Layer 2: Extended Infrastructure

Storage Services: Cinder Block | Swift Object

Network Services: Neutron Networking | Designate DNS

Compute Services: Ironic Bare Metal

Layer 1: Base Compute Infrastructure

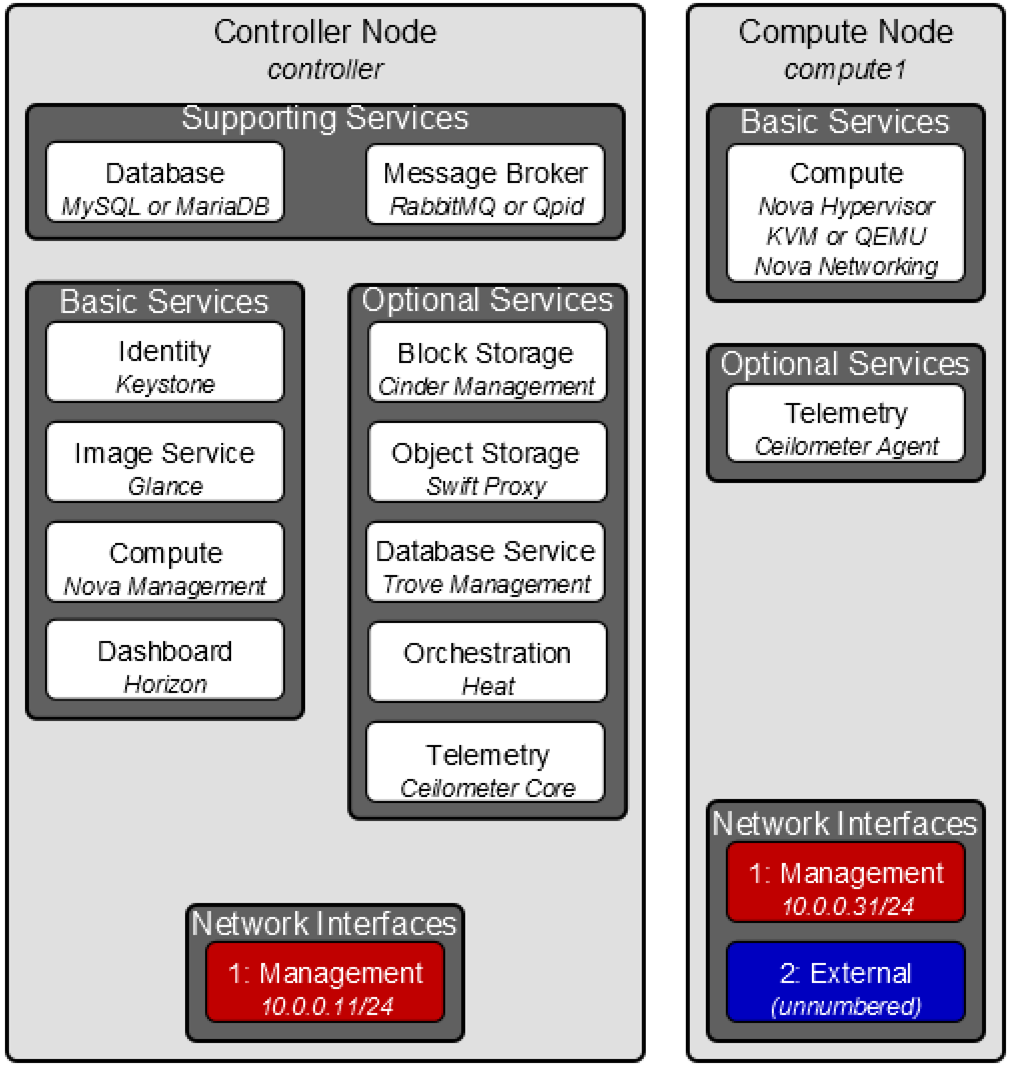
Nova Compute | Glance Image | Keystone Identity

OpenStack Services

- Services are installed on the controller node or in the compute nodes according to their functionalities

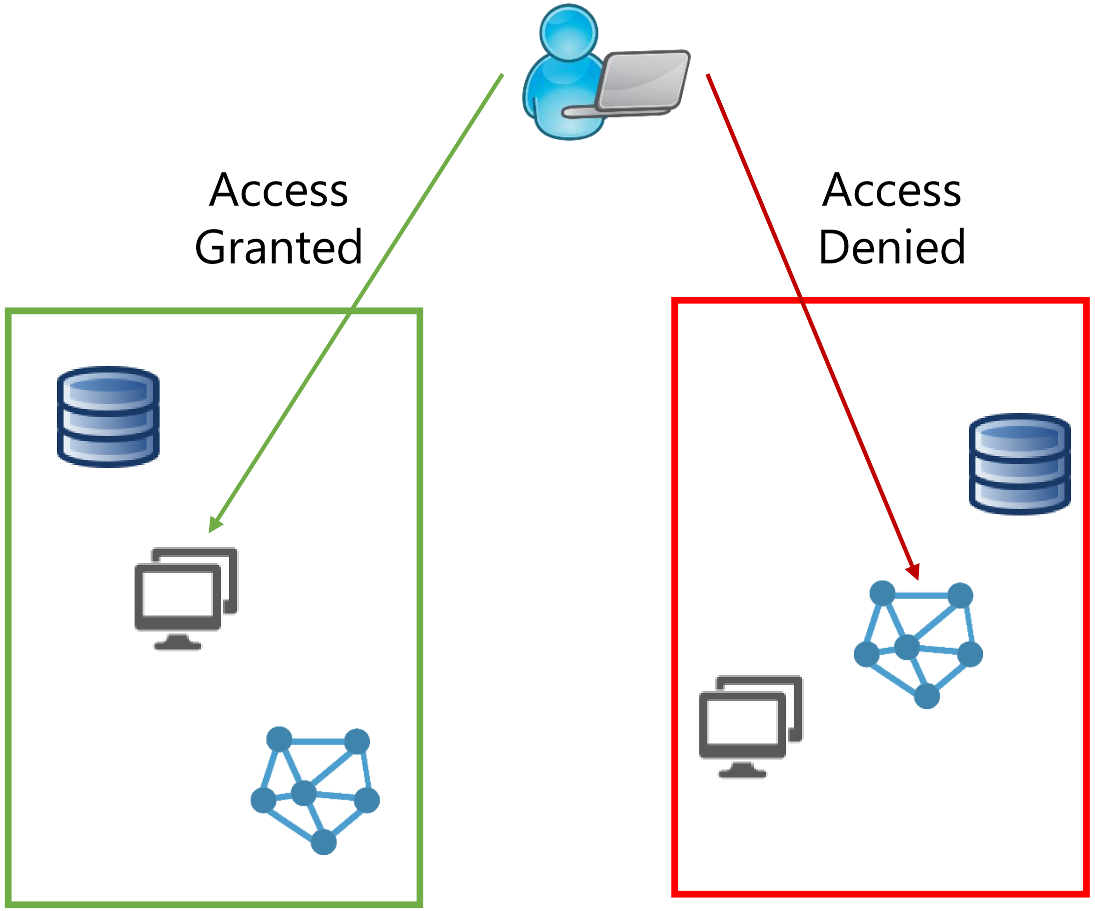
- Some services are required to be installed on both controller and compute nodes with *different configurations*

- All the services in the controller node leverage some *__supporting services__*, one **Database** (e.g. MySQL) for data storage and one **Message Broker** (e.g. Rabbit MQ) for message exchange



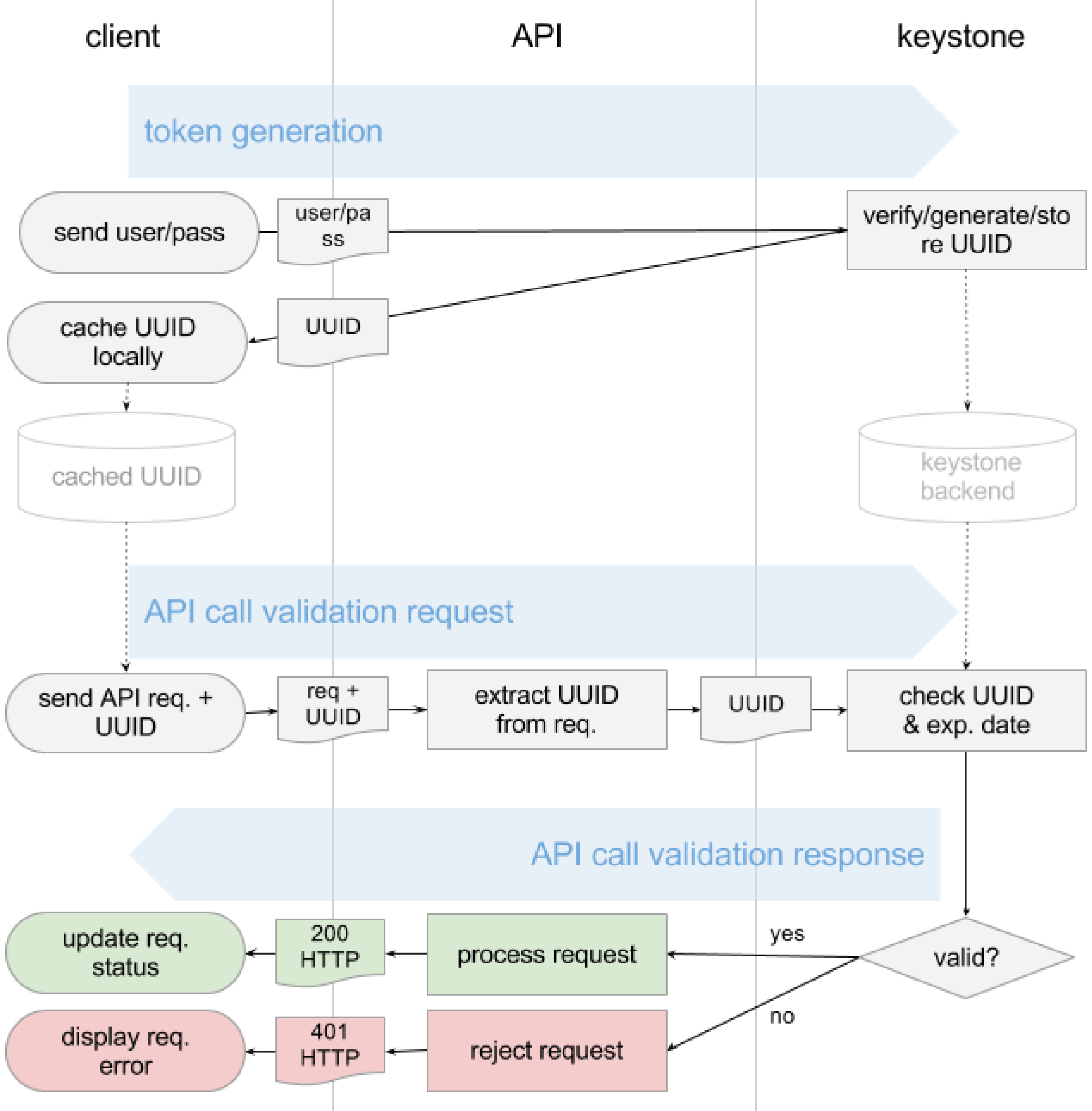
Keystone

- Keystone is the ***identity management component***
- Keystone is used by OpenStack for **authentication** and high-level **authorization**
- It ensures security by granting/denying access to objects (e.g. Virtual Machines or Virtual Networks) to different **Users**
- Objects are grouped into **projects**, authorizations can be granted per project
- Keystone is installed in the Controller node



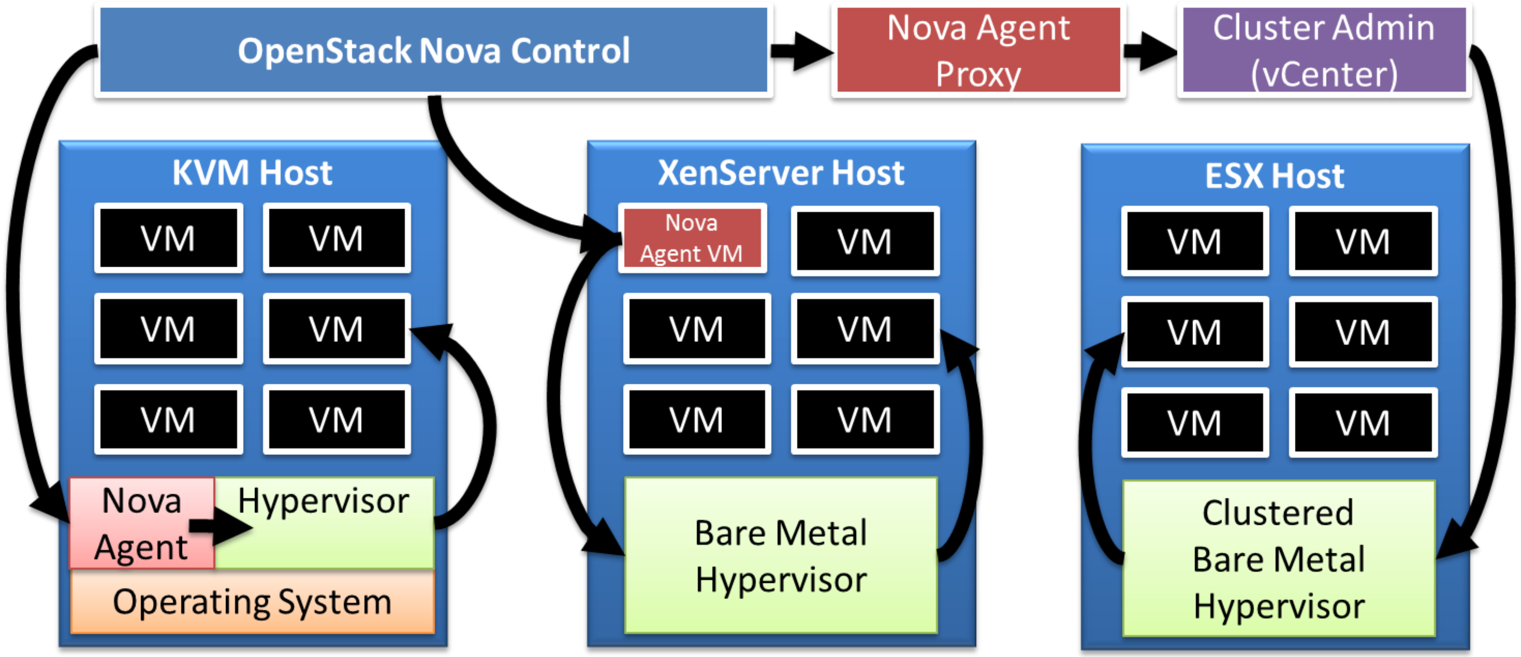
Keystone

- Keystone implements a token-based authorization
- An user first interacts with keystone using an user/pass based authentication
- If successful a token is received
- The token is used to access all OpenStack services
- Each service takes care of validating the token



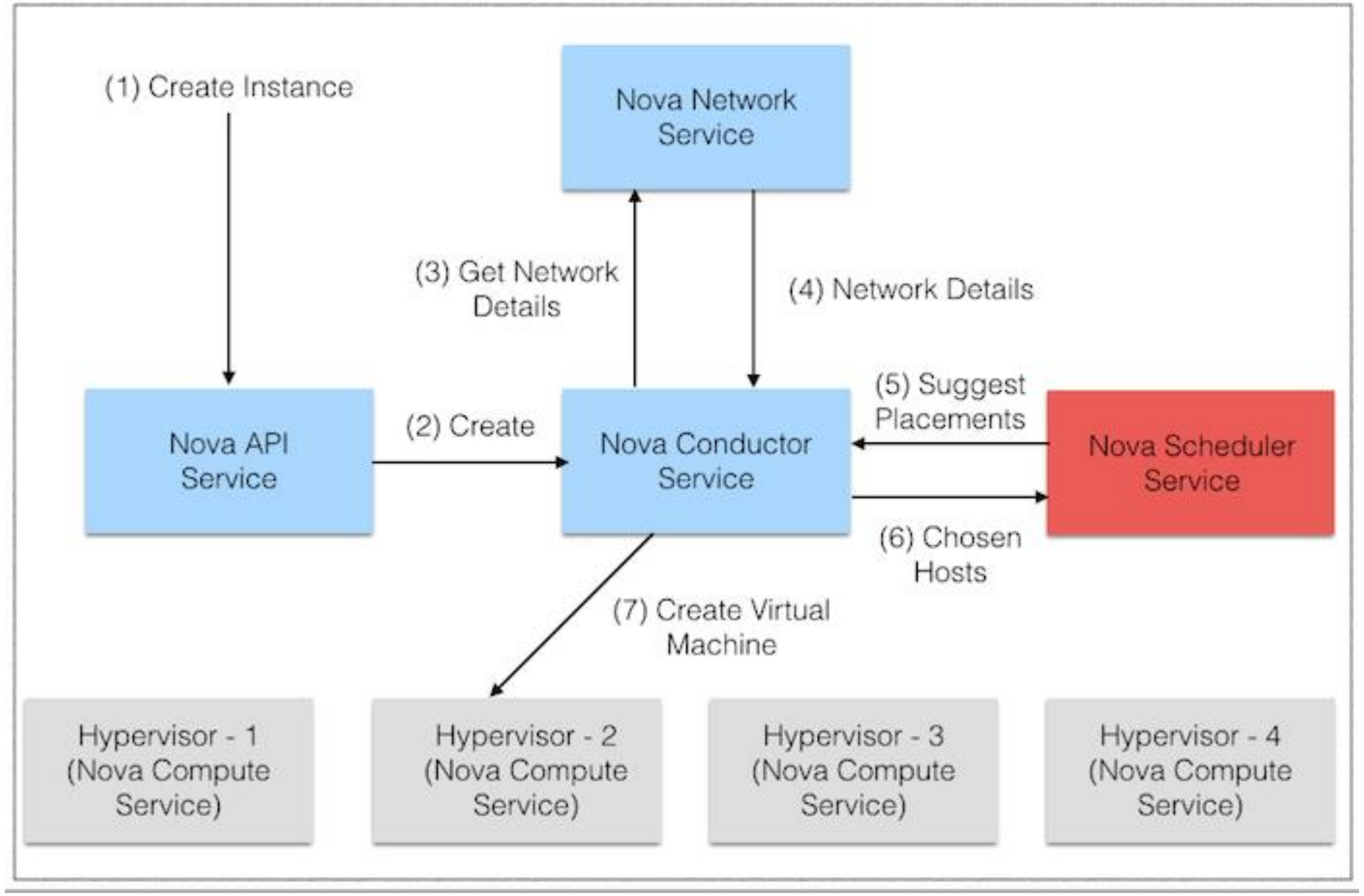
Nova

- Nova is the **instance management** component
- It is responsible for the *instantiation* and *management* of Virtual Machines
- Nova does not implement a new virtualization technology but leverage existing solutions interacting with the hypervisors
- *Different virtualization technologies*, including KVM, Xen, Vmware ESX, are supported



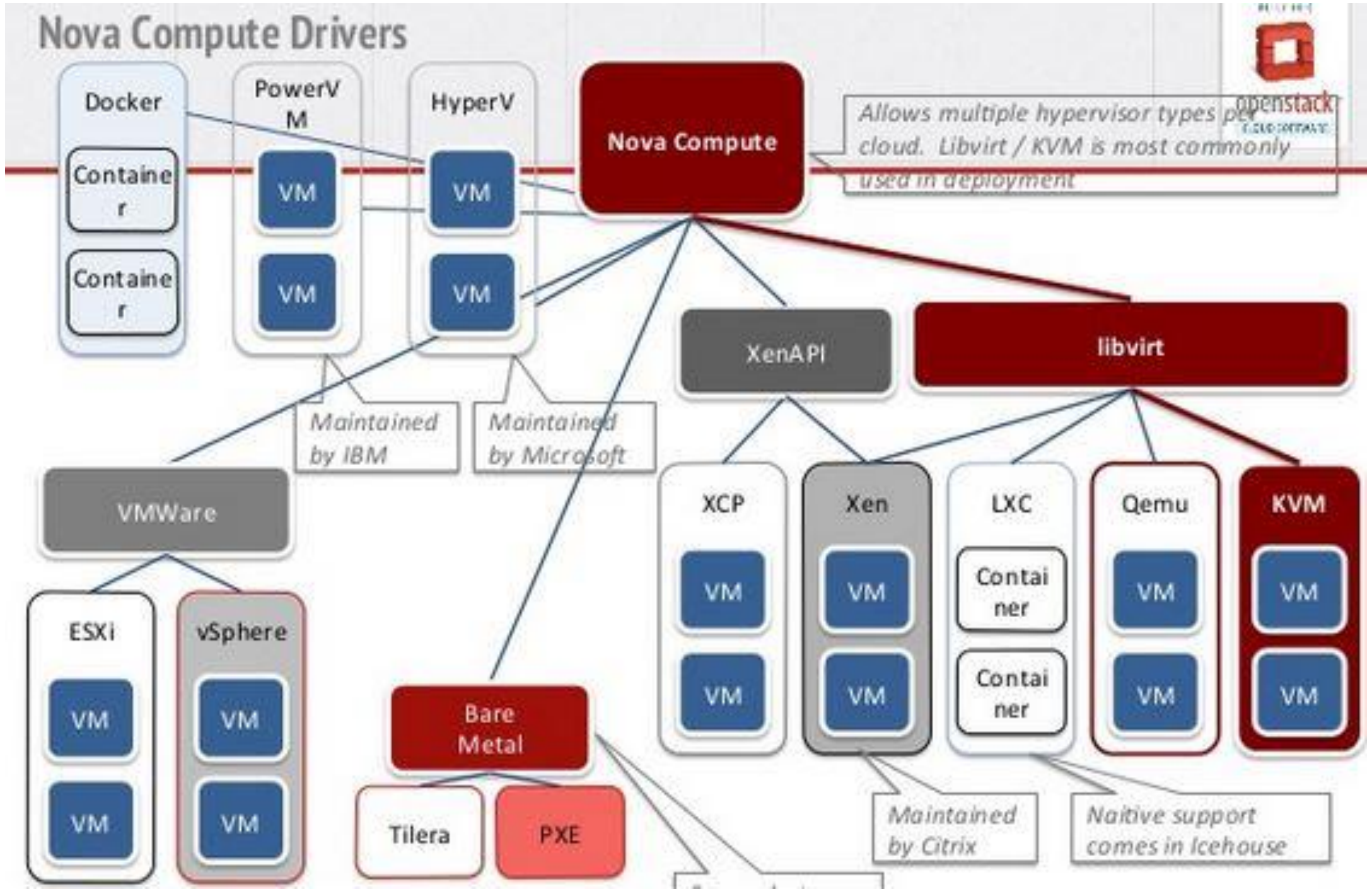
Nova – Controller Subcomponents

- The Nova module installed on the ***controller*** node is composed of the following sub-services: API service, scheduler service, conductor service and network service
- **API service**: exposes the external interface to users
- **Conductor**: manages all the control operations
- **Scheduler**: suggests placement of VMs in the instance according to the status of the compute nodes
- **Network**: implements basic networking services for VMs



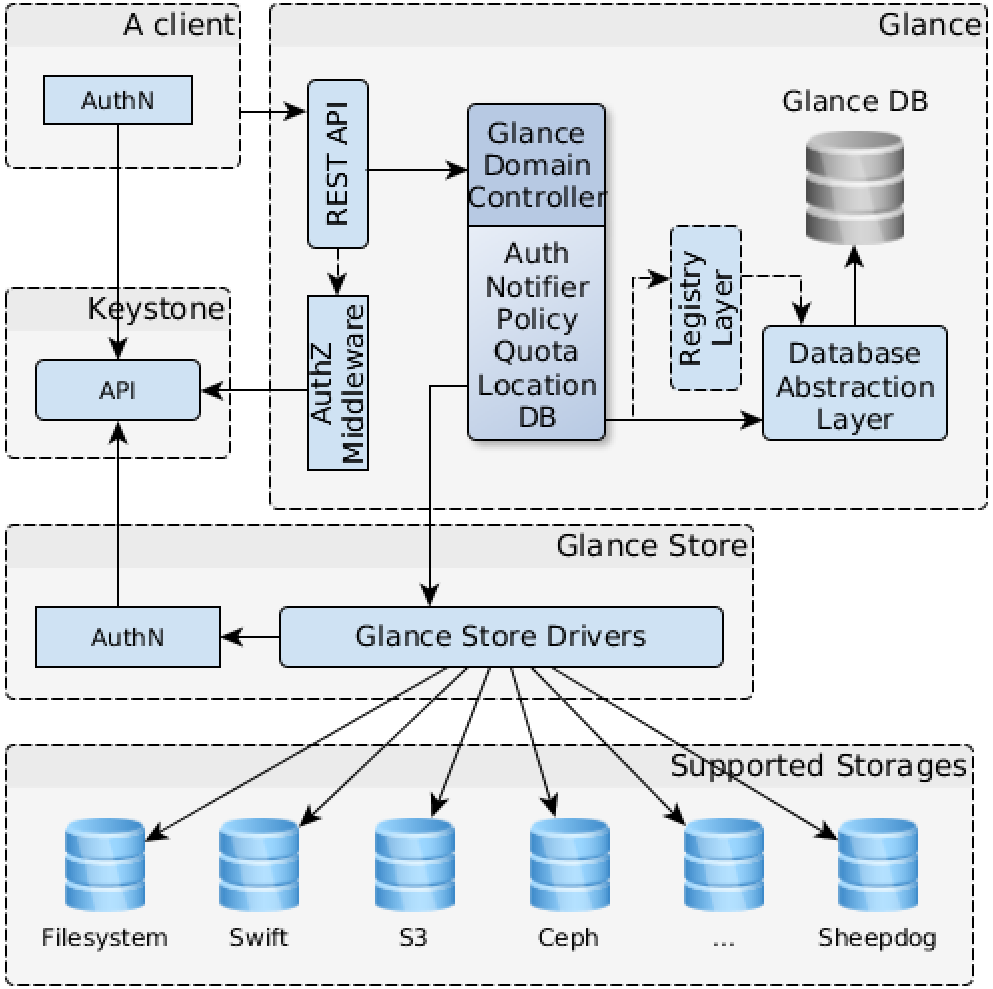
Nova – Compute Subcomponents

- On the ***compute node*** the Nova module is composed only of the ***compute service***
- The compute service receives commands from the controller (Conductor service) and instantiates/terminates VMs instances interacting with the **hypervisor**
- Drivers for different hypervisors are maintained to interface the compute service to different hypervisors
- Each driver exposes a common interface towards the specific APIs of each hypervisor



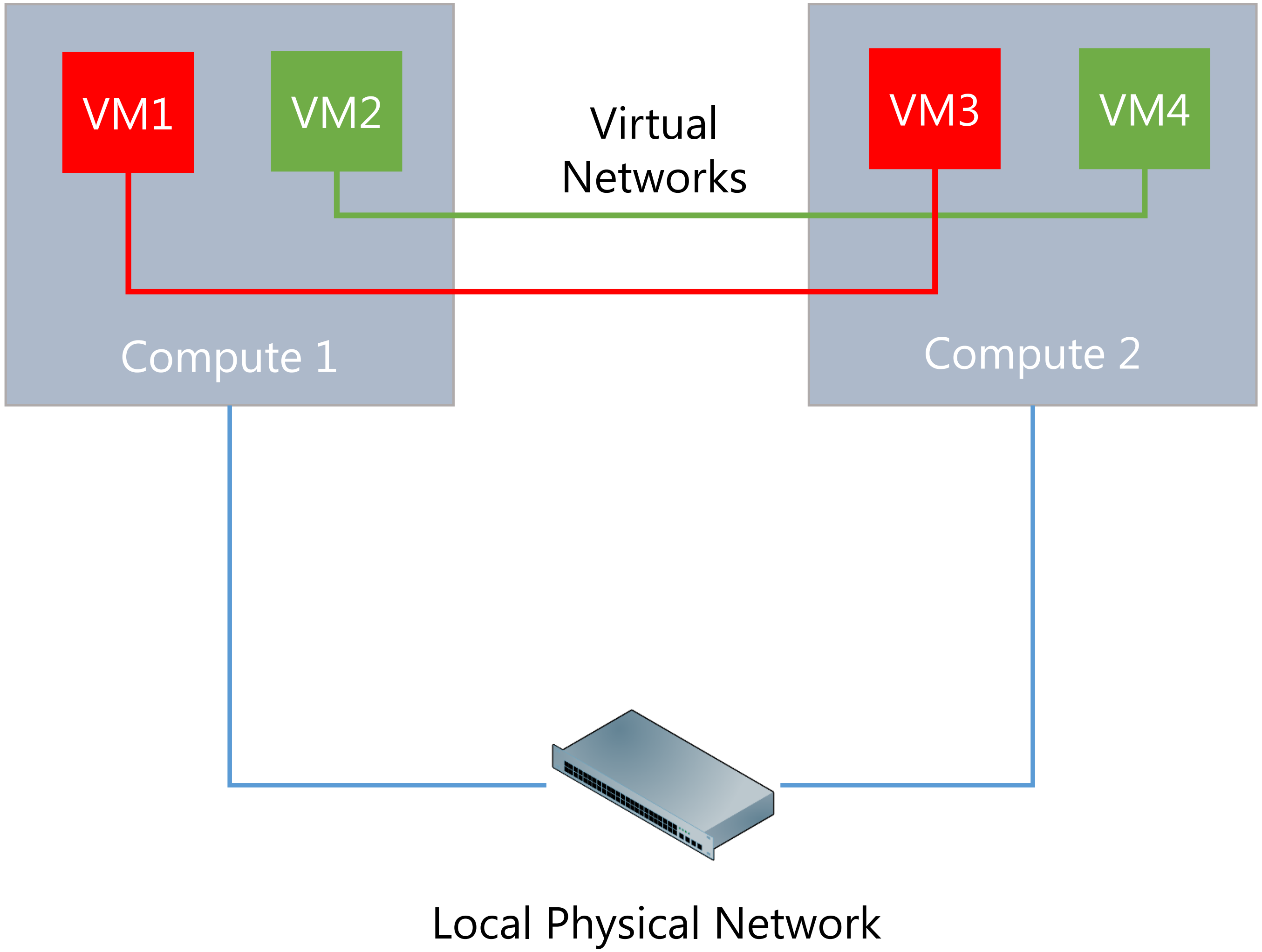
Glance

- Glance is the *image management service*
- Each VM is *instantiated from an image which* includes a specific *operating system pre-installed*
- Glance manages such collection of *VM templates*
- Images can be customized, e.g. a web server image has pre-installed a web server package
- Glance subcomponents are: **glance** (for image management) and **glance storage** (for storage management)
- Glance storage supports different storage options



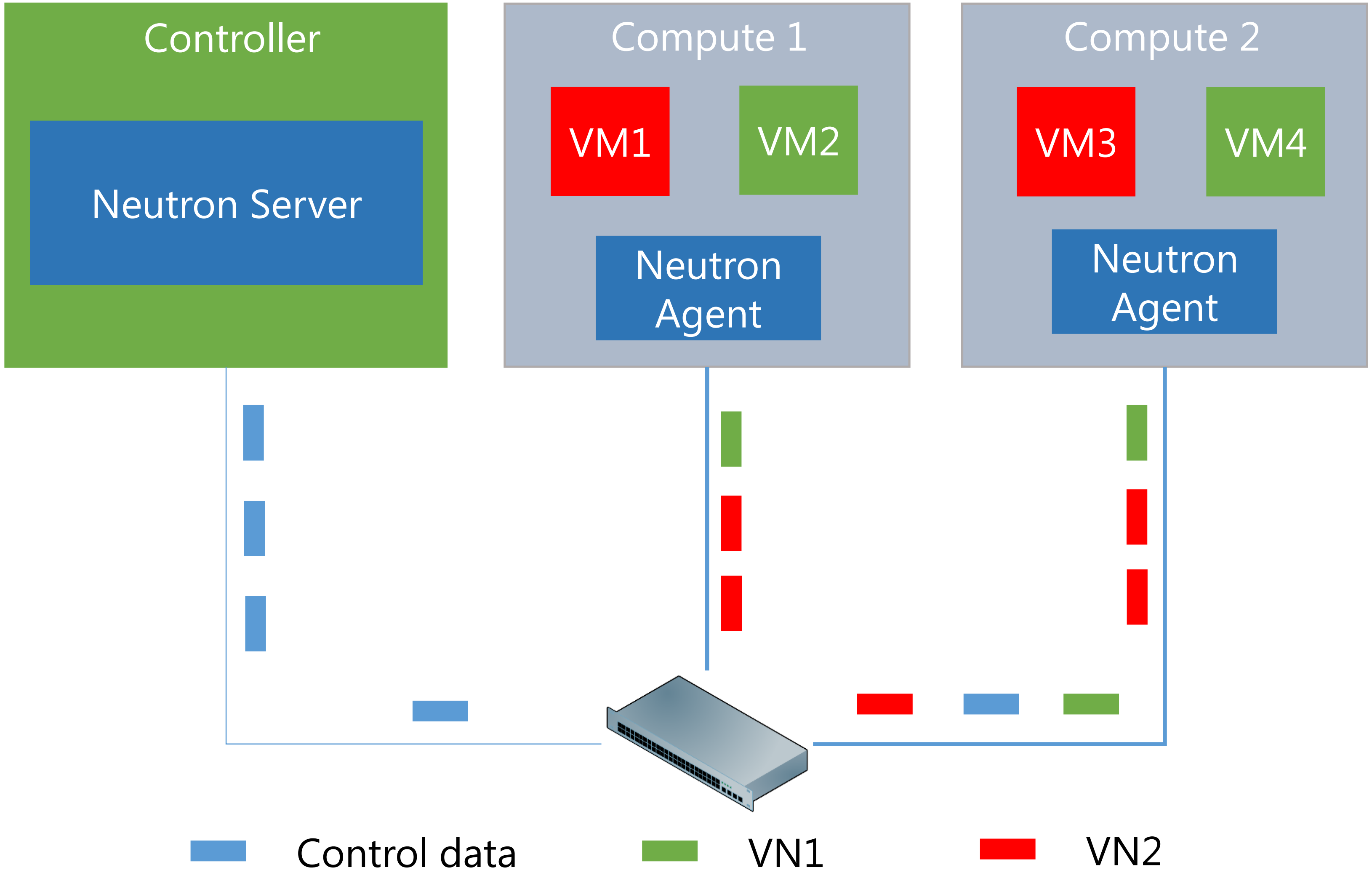
Neutron

- Neutron is the network management components
- When instantiated VMs require a virtual network for communication
- Neutron is responsible for managing infrastructure that allows the creation of Virtual Networks among VMs running on different Openstack compute nodes
- The Local Physical Network that interconnects Computing nodes is exploited to span such virtual networks over different compute nodes



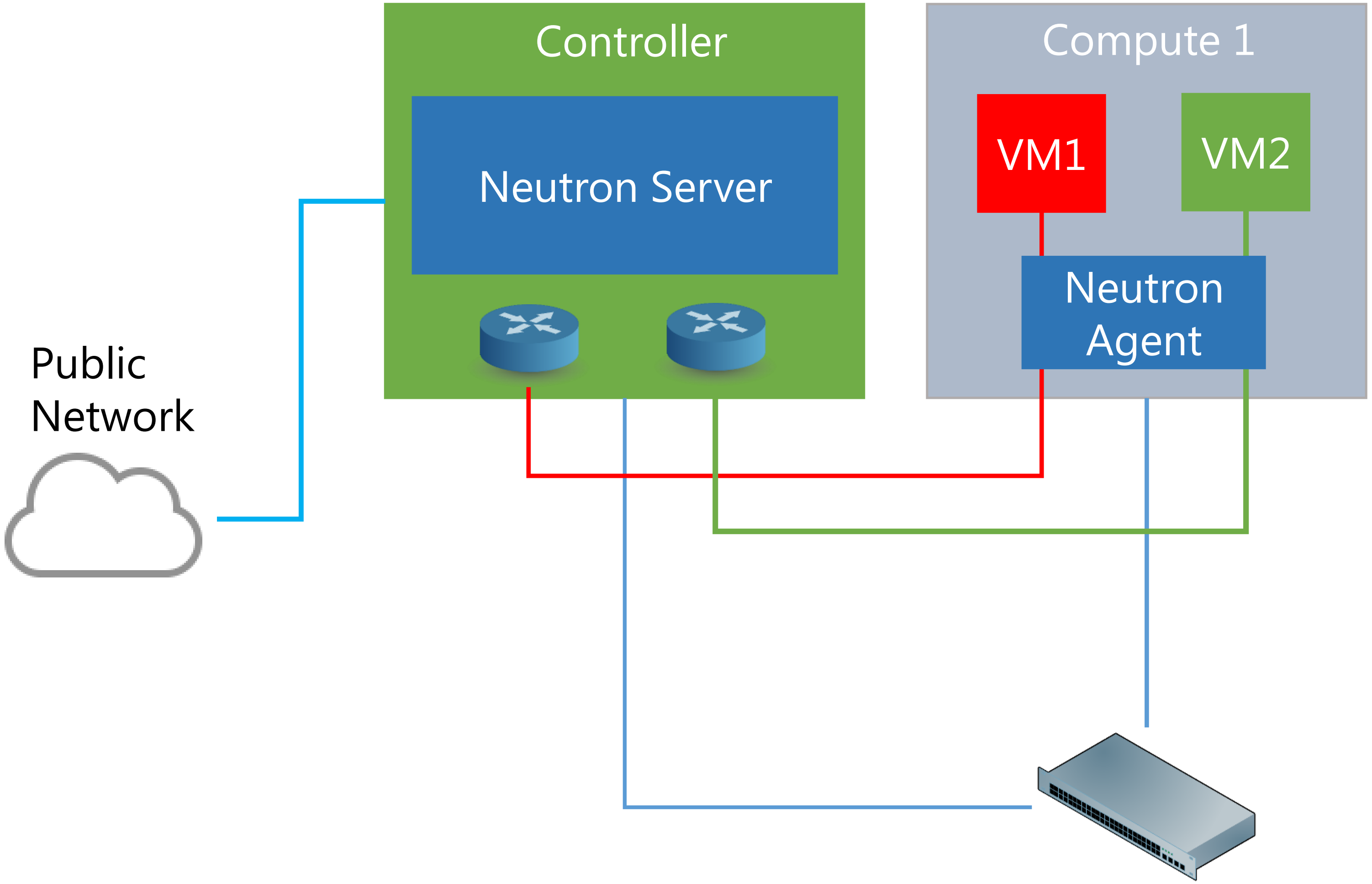
Neutron

- Neutron subcomponents are: *server* and *agent*
- **Neutron Agent**: supports the creation of virtual networks across different compute nodes managing dispatching of data on top of the local physical network
- **Neutron Server**: coordinates neutron agents of the computing nodes and exposes APIs for the management of Virtual Networks



Neutron

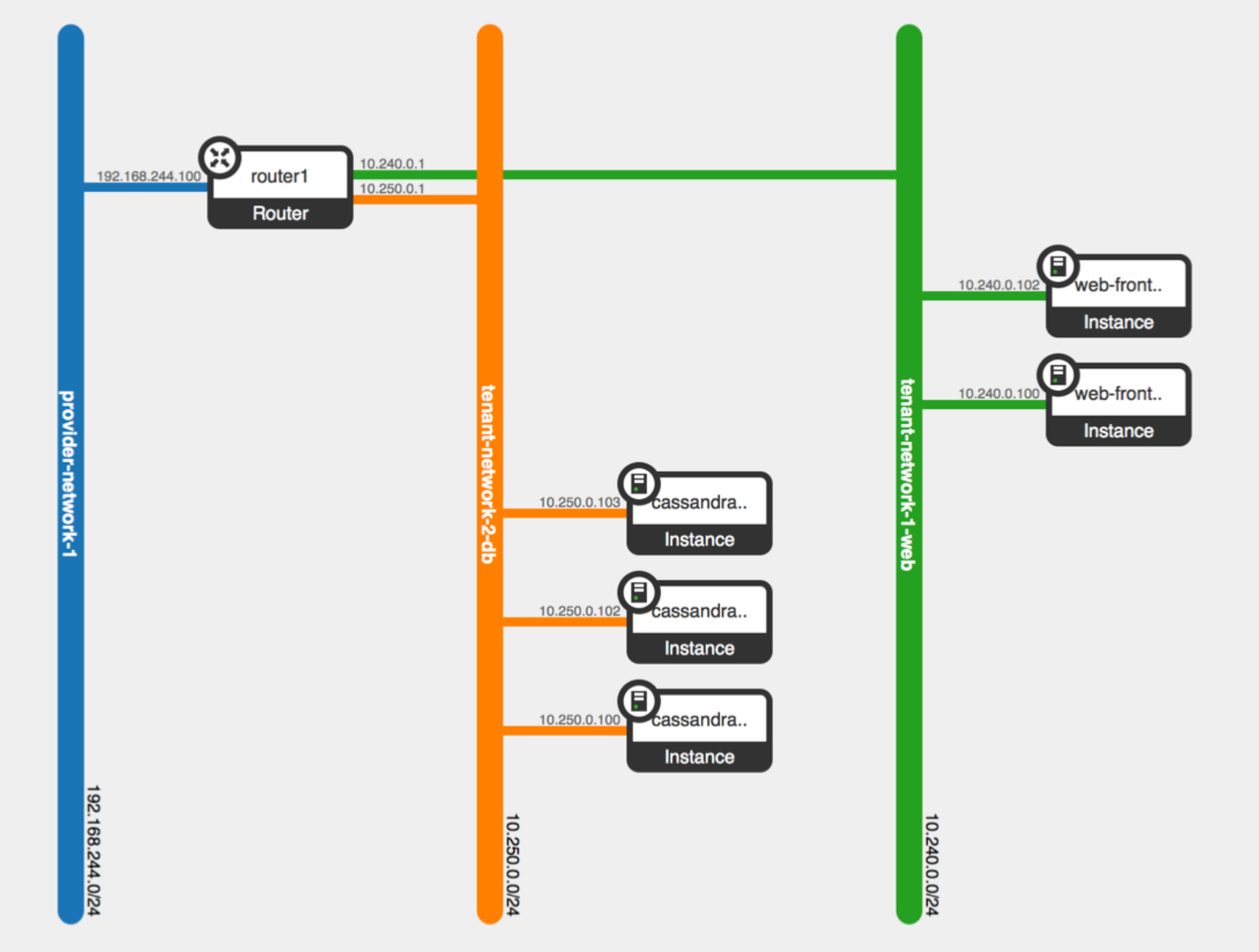
- Virtual Networks are usually private networks
- Neutron allows VMs to be connected to external networks, in order to allow VMs to be accessible from the internet
- To this aim a Network Node (usually the controller node) has to be included in the instance with a connection towards a public network
- This node will reroute traffic from/to the private VNs to/from the public networks Virtual Routers



Neutron

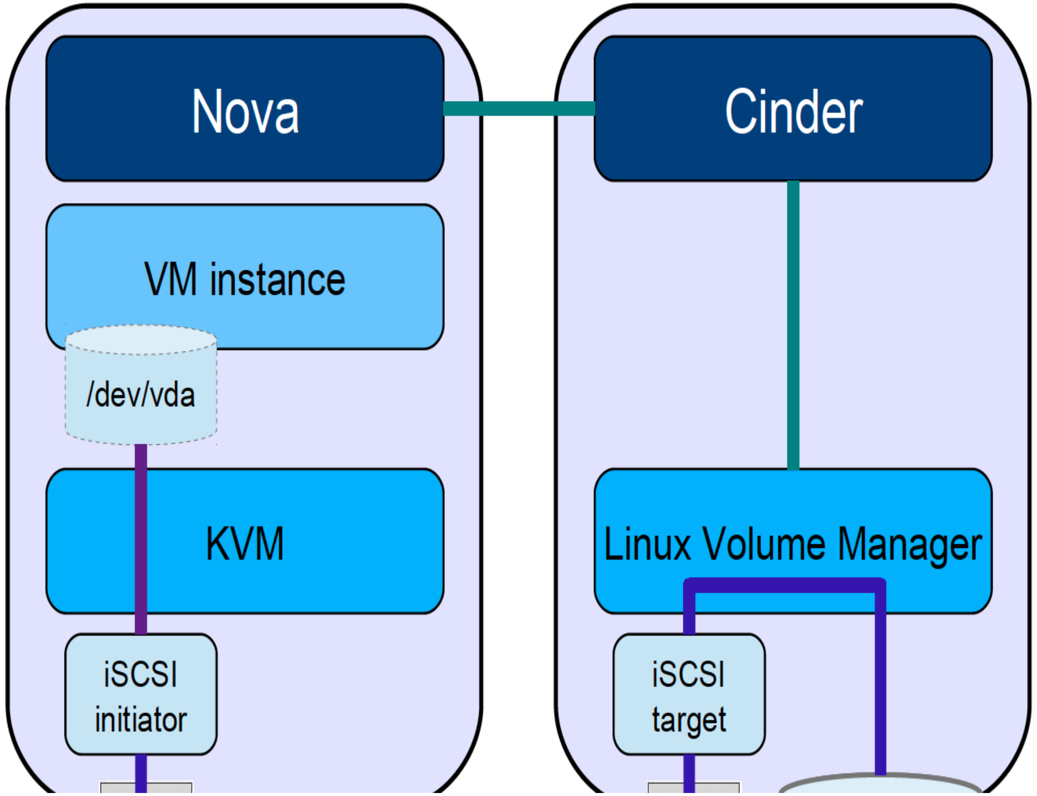
- Public IP addresses can be assigned to VMs

- Virtual Routers at the edge of each Virtual Network will take care of implementing Network Address Translation



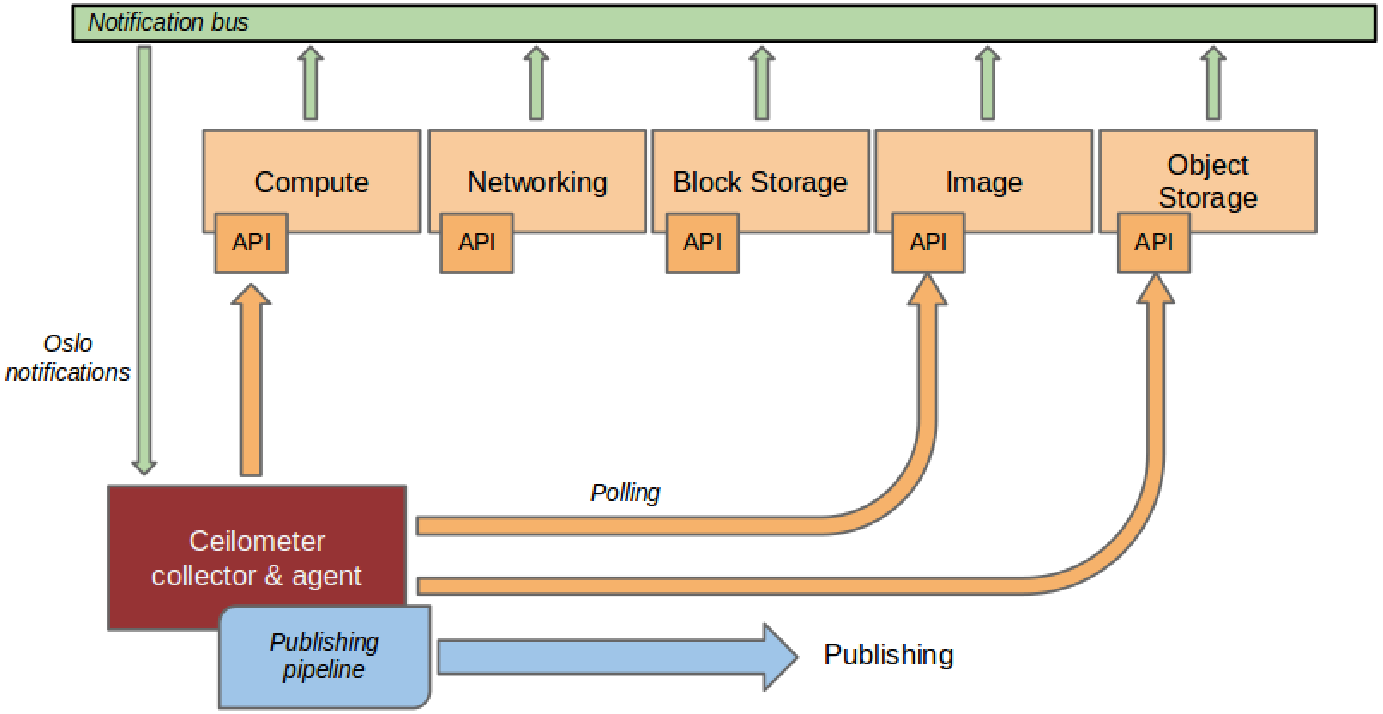
Cinder

- Cinder is the component responsible for managing **volumes**
- Each VM has a default volume which contains the operating system
- If a VM requires extra storage additional volumes can be dynamically created and attached to an instance
- Cinder can be configured to use **local storage** (e.g. Linux LVM) or **shared file systems** (e.g. NFS)



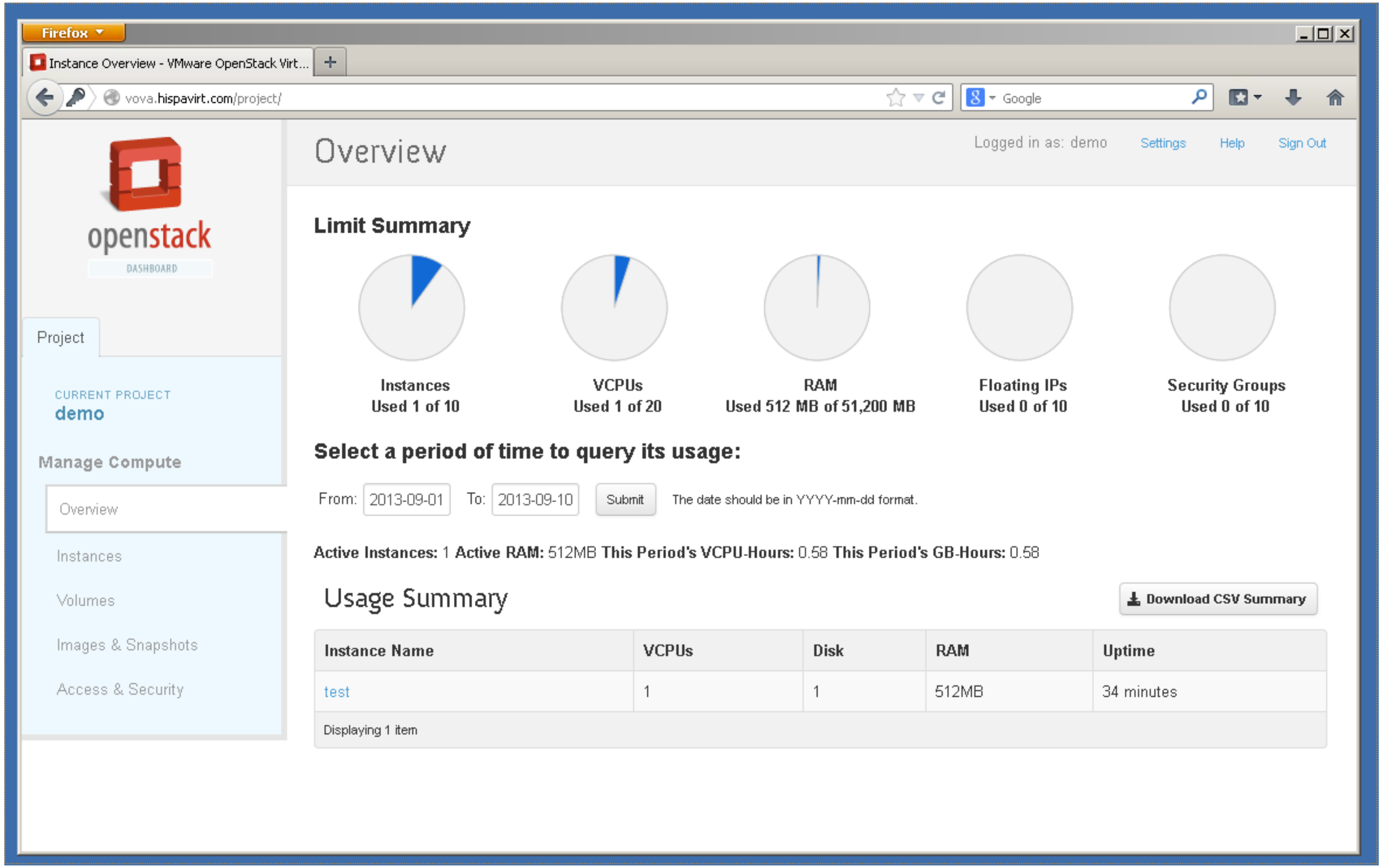
Ceilometer

- Ceilometer is the telemetry component
- It monitors all the component of the instance, **measuring the resource being used** by each User
- Data collected by Ceilometer can be used for **billing** purposes
- Ceilometer also collects **telemetry statistics** which can be used to check the status of the system



Horizon

- OpenStack functionalities are exposed to Users though a **web interface**
- The dashboard is usually exposed by the controller
- It allows management of all the instances aspects
- A set of command line tools are also included for backend management



Overview

Logged in as: demo Settings Help Sign Out

Limit Summary

Instances	VCPUs	RAM	Floating IPs	Security Groups
Used 1 of 10	Used 1 of 20	Used 512 MB of 51,200 MB	Used 0 of 10	Used 0 of 10

Select a period of time to query its usage:

From: 2013-09-01 To: 2013-09-10 Submit The date should be in YYYY-mm-dd format.

Active Instances: 1 **Active RAM:** 512MB **This Period's VCPU-Hours:** 0.58 **This Period's GB-Hours:** 0.58

Usage Summary

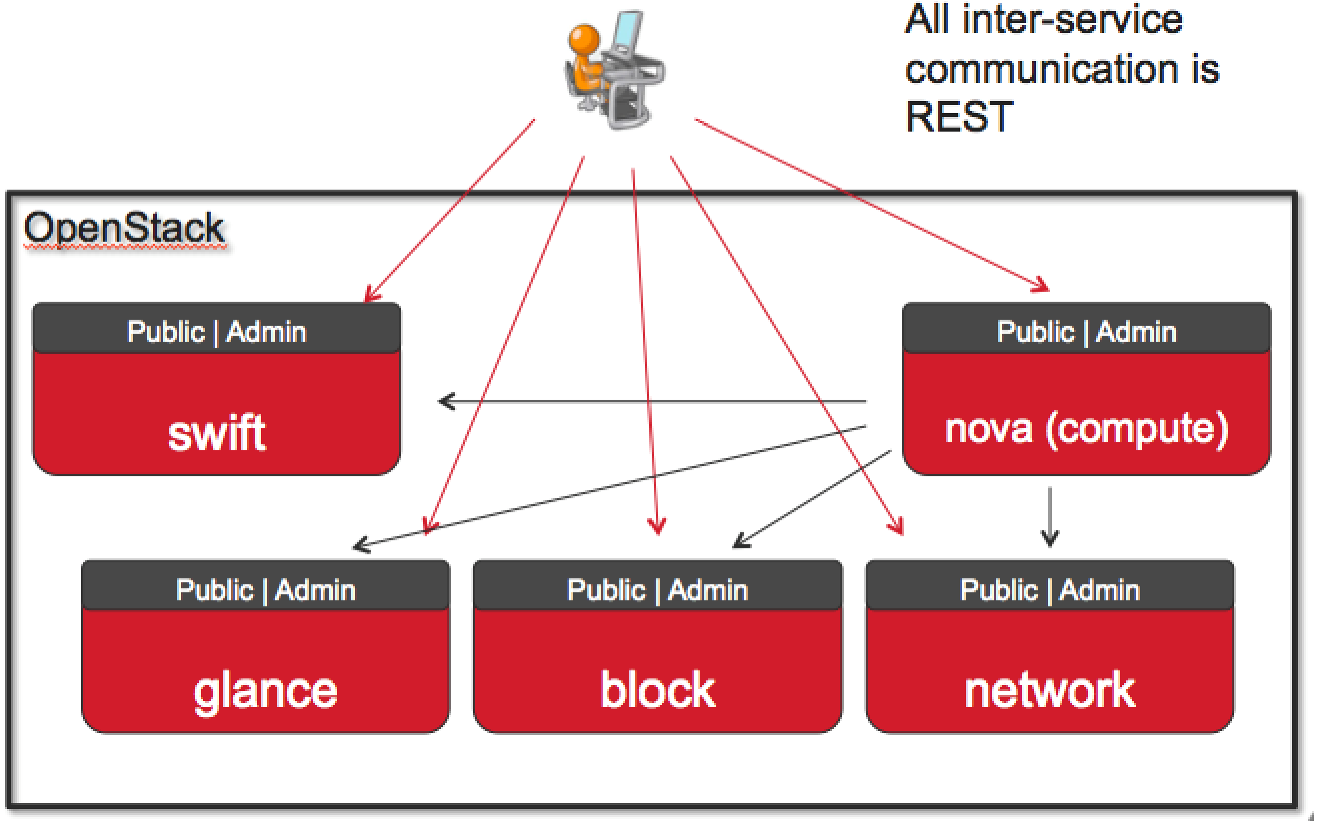
Download CSV Summary

Instance Name	VCPUs	Disk	RAM	Uptime
test	1	1	512MB	34 minutes

Displaying 1 item

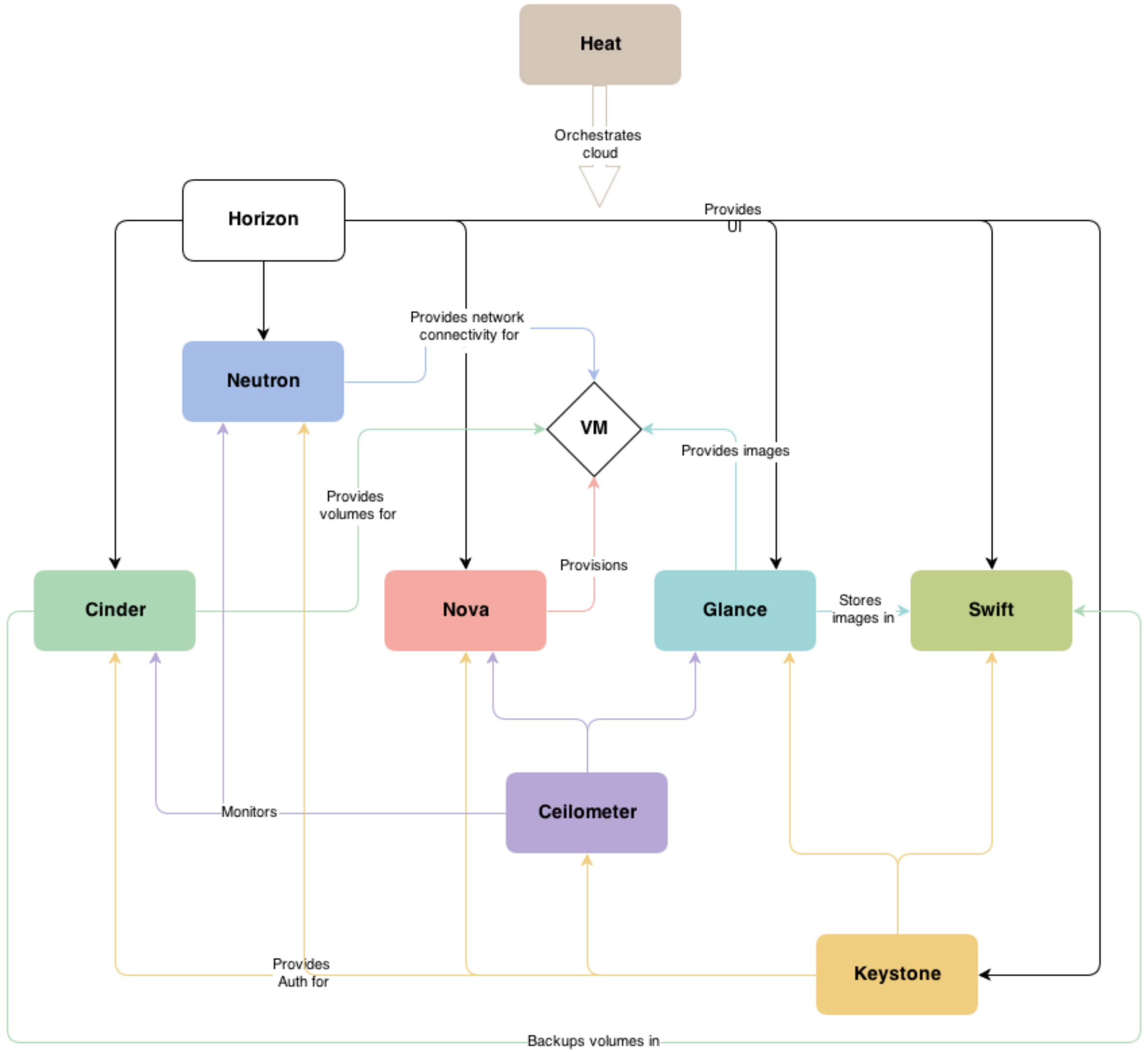
Service APIs

- Every OpenStack **service exposes a set of APIs**
- All APIs communication is **REST**
- APIs are exposed by each service for **inter-service** interaction and to expose a set of functionalities to Users
- APIs can be exploited by Users to embed automation process in external applications



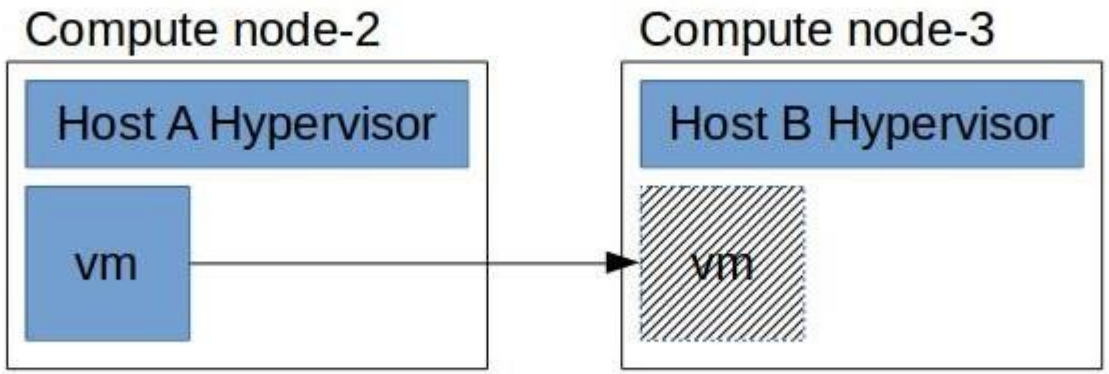
Documentation: http://developer.openstack.org/api-ref.html

OpenStack Service Interactions



Instances Live Migration

- Although Nova Scheduling automatically schedules VM execution on compute nodes based on resource status, **manual placement** of VM on a specific node is allowed for the **Instance Administrator**
- To this aim, OpenStack allows **Live Migration** of VMs among different compute node
- Live Migration allows administrators to move a VM from one host to another minimizing the down-time without turning the VM off



VM live migration requires Nova and Cinder to be configured with a **storage** which is **shared** among all the compute nodes, in order to allow VM transfer without downtime.
A shared storage can be implemented through **NFS** for example.

GlusterFS

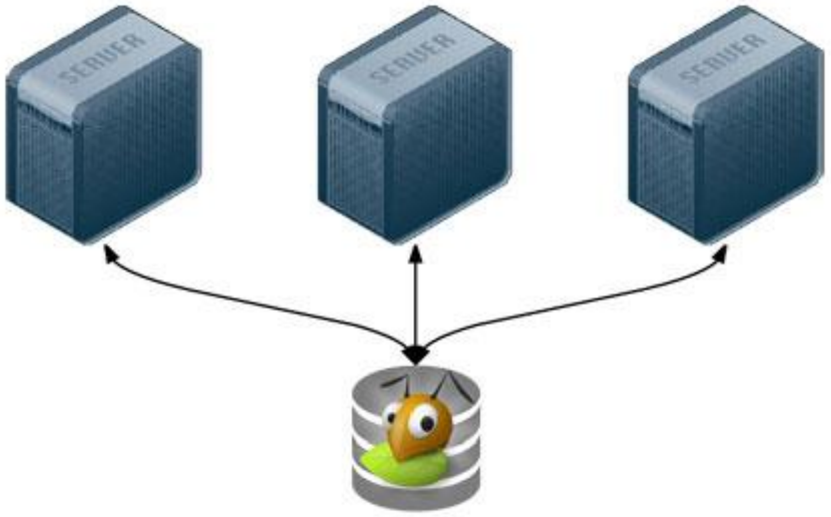
- Although NFS is a shared storage for volumes and VMs, its **centralized architecture** refrains its usage in practical deployments

- **Distributed** alternatives are usually adopted to increase resiliency to failure and guarantee scalability exploiting storage locally available to compute nodes

- *GlusterFS* is an example of network-attached storage file system usually adopted in OpenStack as shared storage point

- GlusterFS can be used locally in the same way is configured NFS

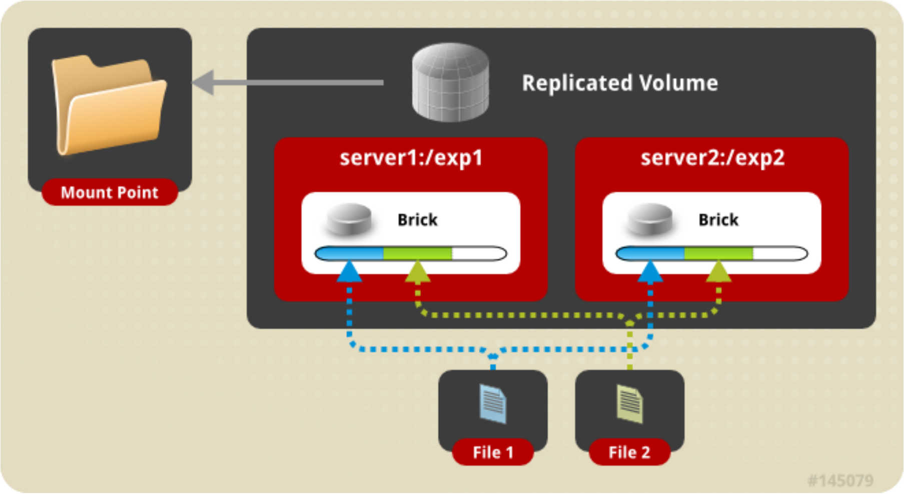
- There is no distinction between clients and server, *all the nodes participate offering some of the local storage*

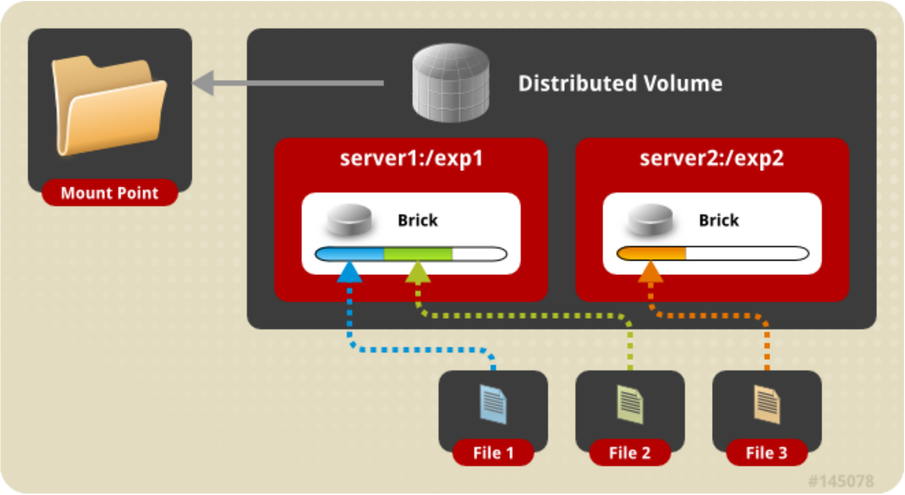


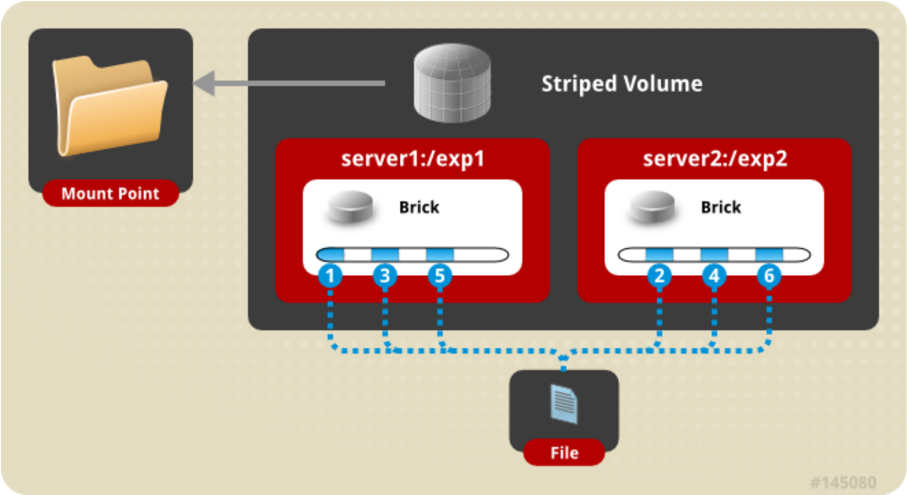
GlusterFS (Distributed, replicated volume)

GlusterFS – Basic Modes

- GlusterFS is *highly configurable*, with different levels of redundancy and replica
- Basic configuration includes: **replicated** volumes, **distributed** volumes and **striped** volumes

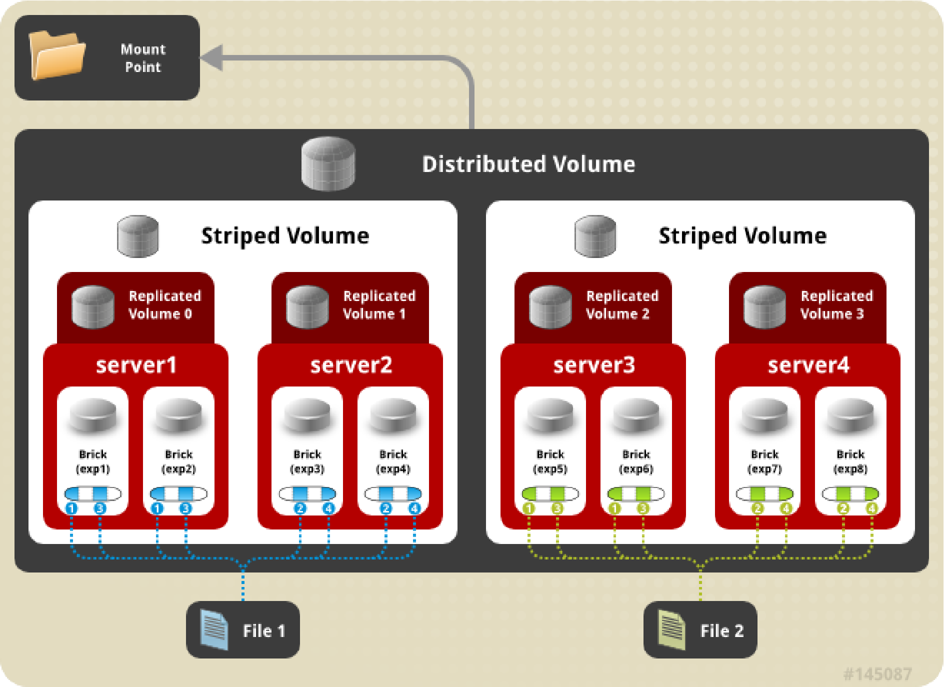


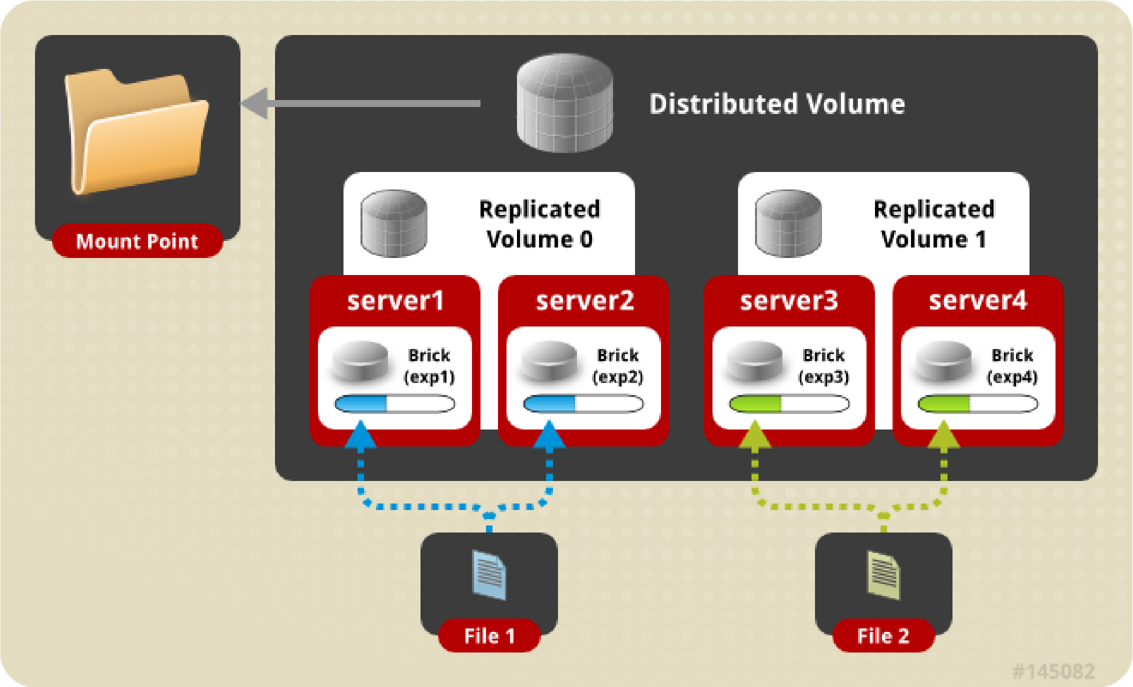




GlusterFS – Advanced Modes

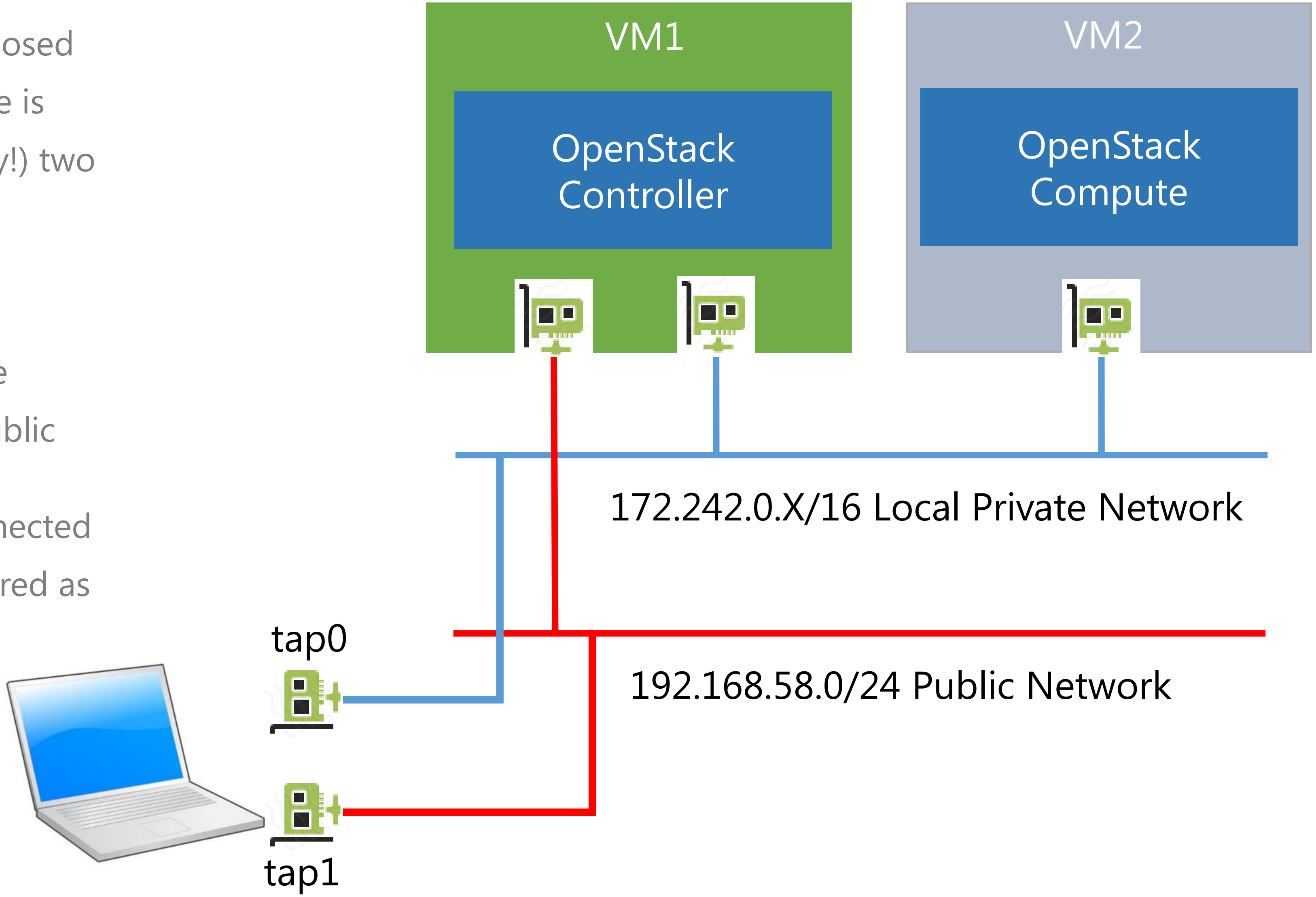
- In order to meet different requirements different combination of basic modes are allowed: **striped replicated** and **distributed replicated**





Demo Time!

- A simple OpenStack instance composed by a controller and a compute node is emulated by means of (oh the irony!) two virtual machines running on top of VirtualBox
- Two emulated Ethernet network are configured: one private and one public
- The controller node is the one connected to both the network as it is configured as network node


VM1
OpenStack Controller
VM2
OpenStack Compute
172.242.0.X/16 Local Private Network
192.168.58.0/24 Public Network
tap0
tap1

Demo

- VM images are available if some wants to replicate the test:
  - http://atlantis.iet.unipi.it/controller.ova
  - http://atlantis.iet.unipi.it/compute.ova
  - VM username: *root* password: *reverse*
  - OpenStack interface: http://172.242.0.100/ username: *admin* password: *7f937d60365440c4*

References

- https://www.openstack.org/
- https://wiki.openstack.org/wiki/Main_Page
- http://docs.openstack.org/openstack-ops/openstack-ops-manual.pdf
- http://www.gluster.org/community/documentation/index.php/Main_Page