Tutorial on OpenStack and the GARR Federated Cloud

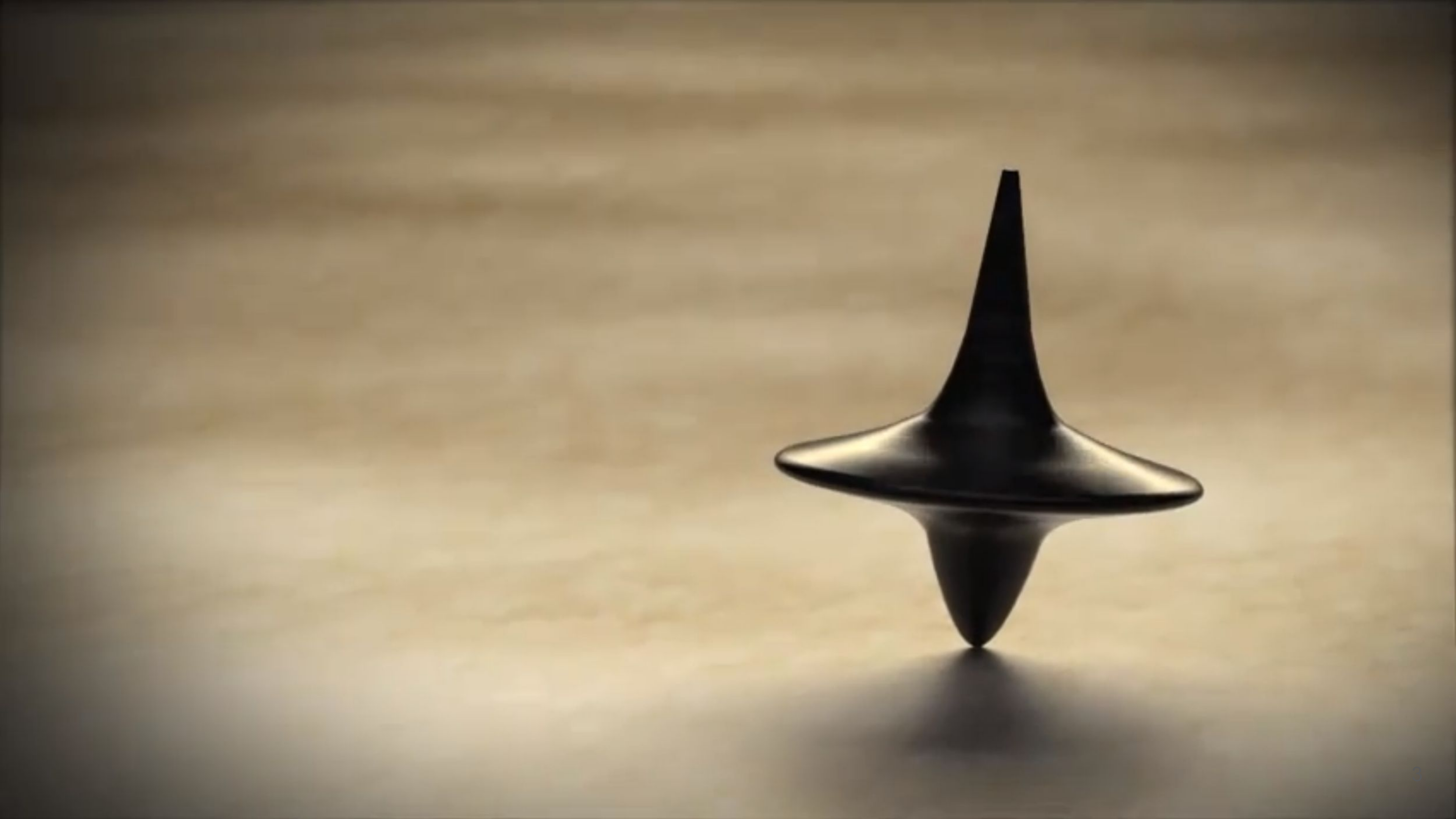
Tutors: A. Barchiesi, A. Colla, G. Marzulli

Workshop GARR - Roma 2017

Alex Barchiesi (alex.barchiesi@garr.it, GARR dipartimento CSD)

Outline

- Introduction to Cloud Computing
- Requirements and Goals
- OpenStack
- GARR Cloud Infrastructure
- GARR Federated Cloud:
  - Architecture
  - How we started building it
  - How you can build it
- Status
- Demo & Hands on sessions
  - Link to Hands-on guide: https://goo.gl/qAqjAq
  - Link to this presentation: https://goo.gl/DMfrwp



The big switch

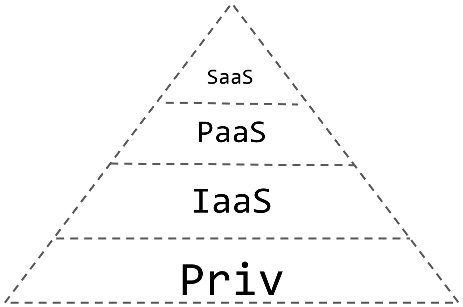
"You can never change things by fighting the existing reality.

To change something, build a new model that makes the existing obsolete."

[R. Buckminster Fuller]

JP Rangaswami (Corporate Eco Forum) comments over on his blog, advocating **Open Source as the antidote to Cloud Monopolies**: "I have always had this sense that there is no longer any room for artificial monopolies, that the market will provide a self-correcting mechanism. But I have always been wrong on this. We can argue about why this is so, but not about the fact. **Microsoft, Google and Apple are facts**. **Open standards, open platforms and open source are ways to prevent this happening**. Ways to guarantee that history won't repeat itself. But this needs coherent communal action, something that is hard to achieve in emergent environments."

Types of Cloud Services

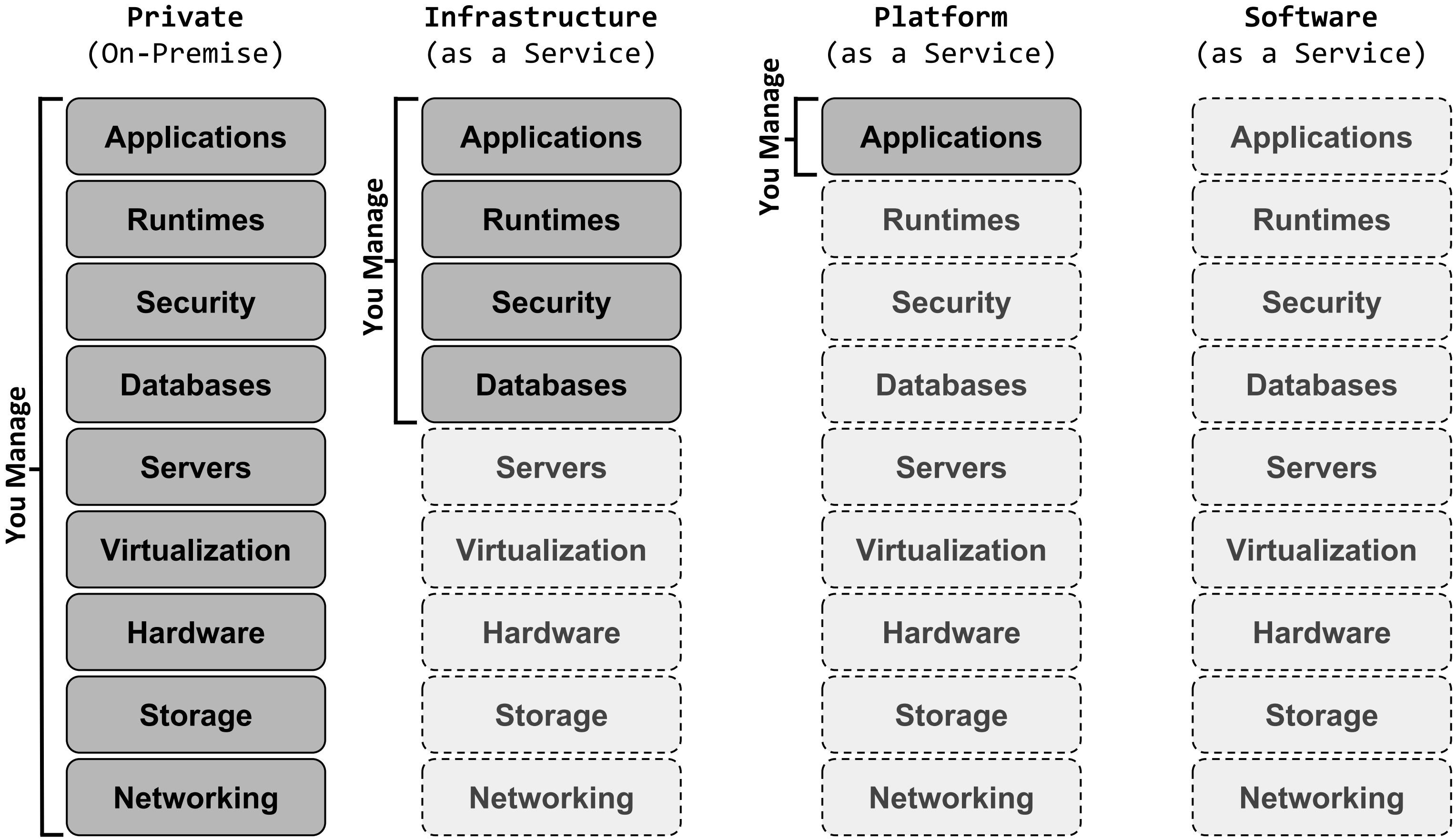


1. **Utility computing.** Amazon’s success in providing virtual machine instances, storage, and computation at pay-as-you-go utility pricing was the breakthrough. Developers, not end-users, are the target of this kind of cloud computing. (**IaaS**)

2. **Platform as a Service.** One step up from pure utility computing are platforms (like Google AppEngine and Salesforce’s force.com), which hide machine instances behind higher-level APIs. (**PaaS**)

3. **Cloud-based end-user applications.** Any web application is a cloud application in the sense that it resides in the cloud. Google, Amazon, Facebook, twitter, flickr, and virtually every other Web 2.0 application is a cloud application in this sense. (**SaaS**)

Cloud Service Models



Private
(On-Premise)

You Manage

Applications
Runtimes
Security
Databases
Servers
Virtualization
Hardware
Storage
Networking

Infrastructure
(as a Service)

You Manage

Applications
Runtimes
Security
Databases
Servers
Virtualization
Hardware
Storage
Networking

Platform
(as a Service)

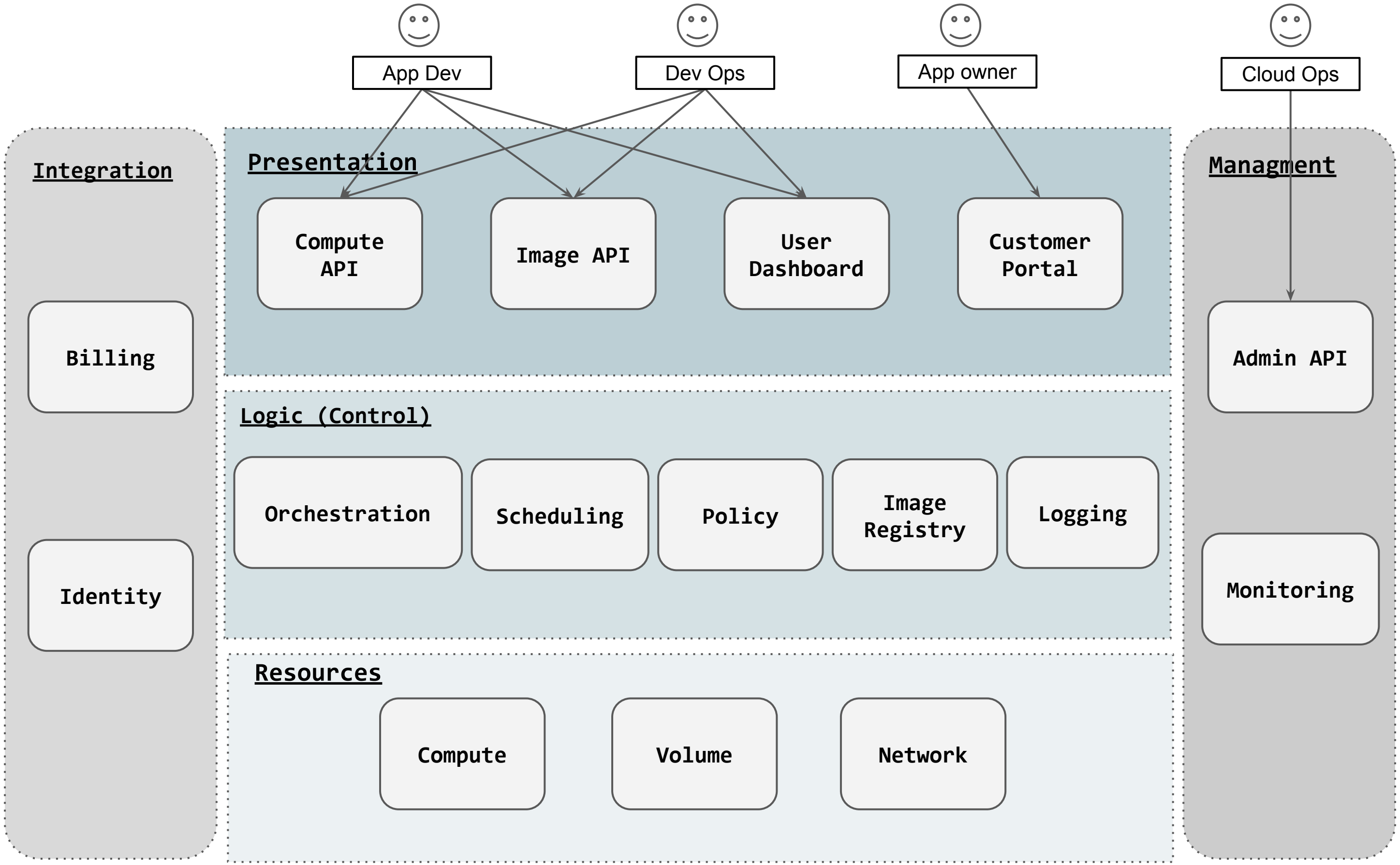
You Manage

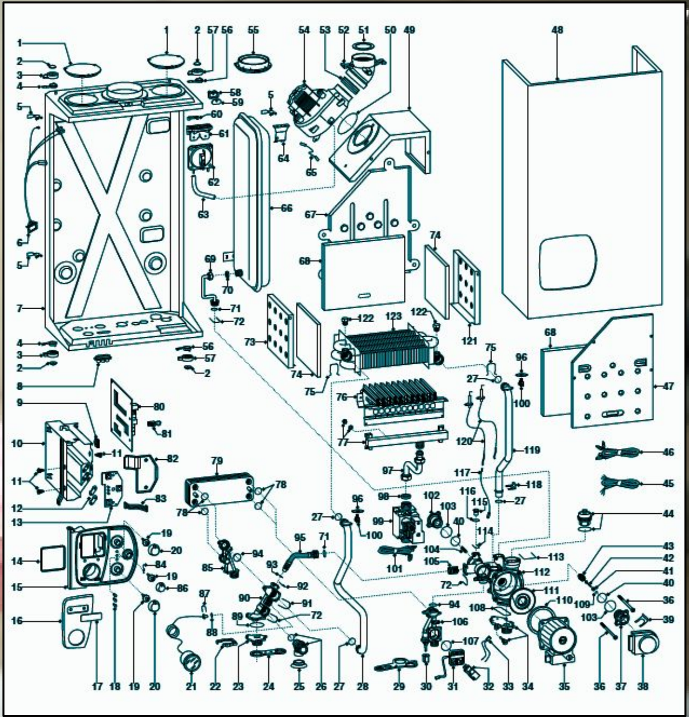
Applications
Runtimes
Security
Databases
Servers
Virtualization
Hardware
Storage
Networking

Software
(as a Service)

Applications
Runtimes
Security
Databases
Servers
Virtualization
Hardware
Storage
Networking

Conceptual Cloud Architecture



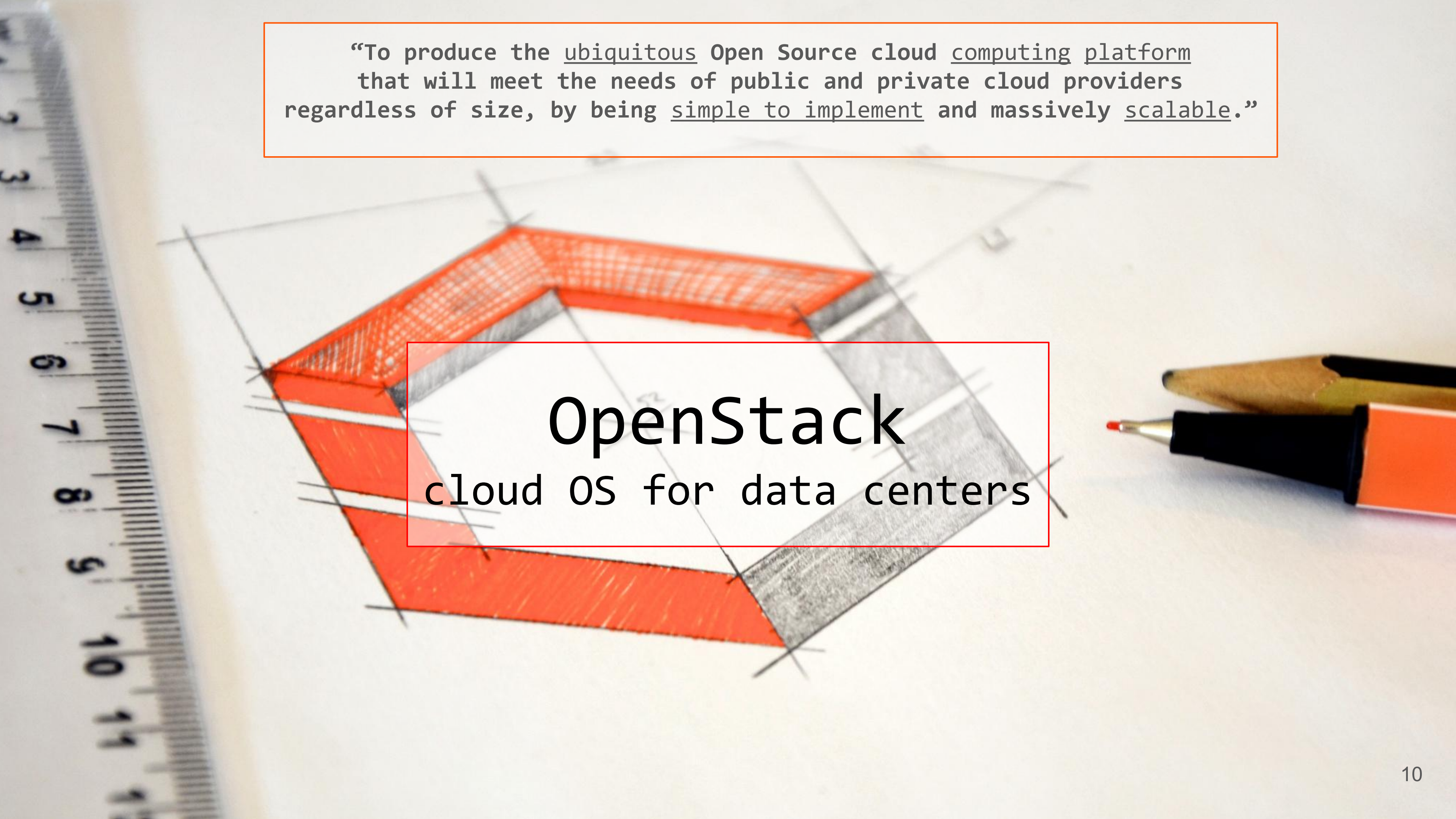


BETTY GILLIS

(our) goals and requirements

- open-source
- reduced manpower *efforts*
- sharing resources
- simplify provisioning of storage and computing services
- different organizations
- unified access (SSO)
- always on
- replicable and scalable
- *self* deploying and *self* healing
- elastic
- separation / flexible security policies
- Empower users with something more than a PAAS and something easier than a IAAS

> “To produce the ubiquitous Open Source cloud computing platform that will meet the needs of public and private cloud providers regardless of size, by being simple to implement and massively scalable.”

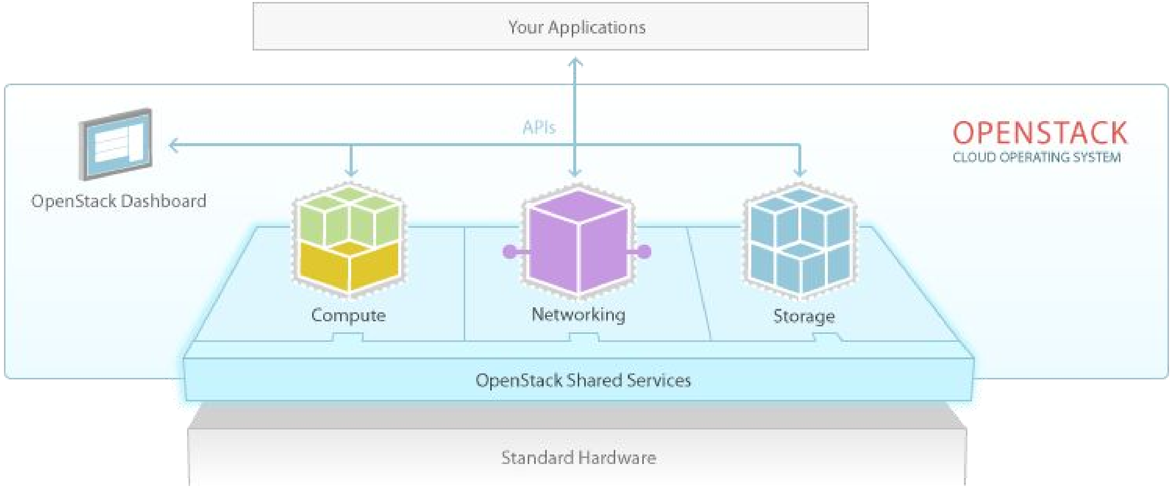


OpenStack

cloud OS for data centers

What is OpenStack?

- Apache 2.0 license (OSI), no paid enterprise version
- Open design process, 2x year public Design Summits
- Publicly available open source code repository
- Open community processes documented and transparent
- Commitment to drive and adopt open standards
- Modular design for deployment flexibility via APIs

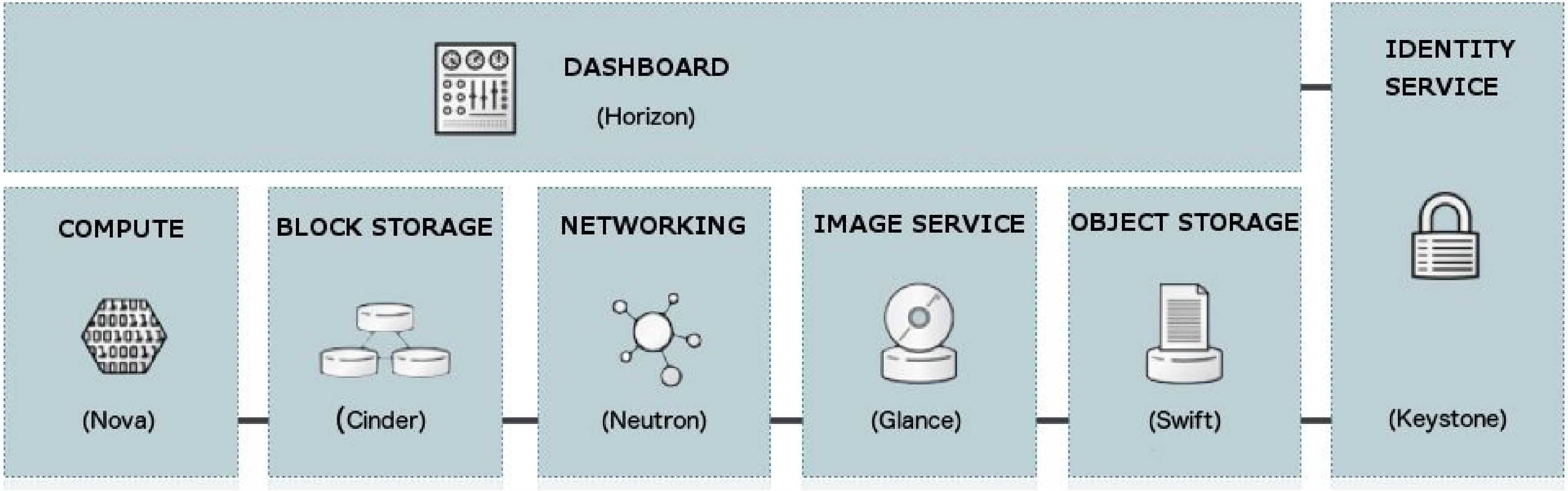


Is a community

creating public software to build private and public clouds

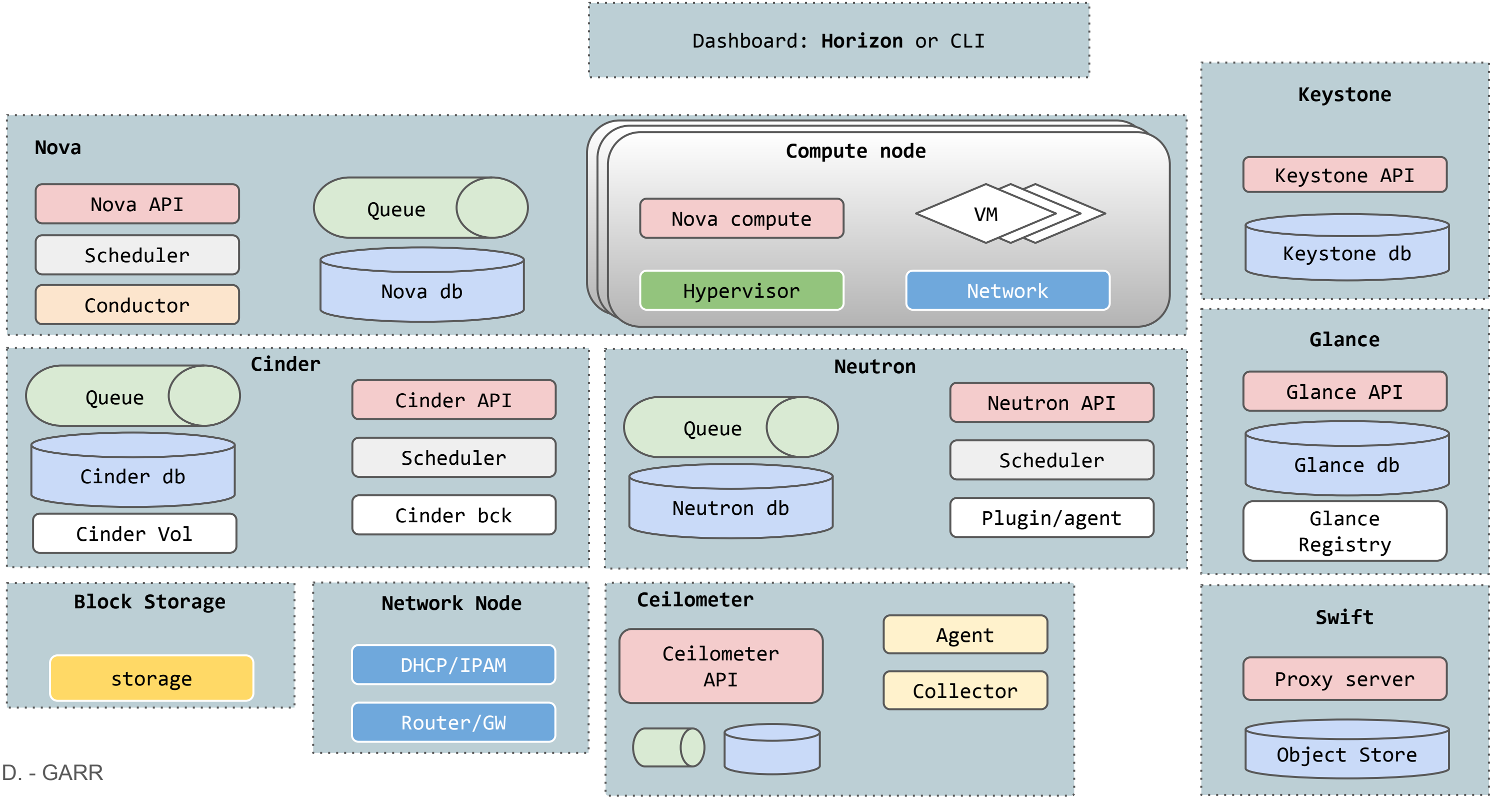
Components (umbrella project for)

- **Horizon** (Dashboard)
- **Keystone** (Identity Management)
- **Nova** (Compute, where VMs are run)
- **Glance** (Image Service, where templates are)
- **Cinder** (Block Storage, persistent storage for VMs)
- **Swift** (Object Storage, snapshots and not frequently updated data)
- **Neutron** (Networking and SDN)
- **Ceilometer** (Telemetry)



OpenStack through VM provisioning

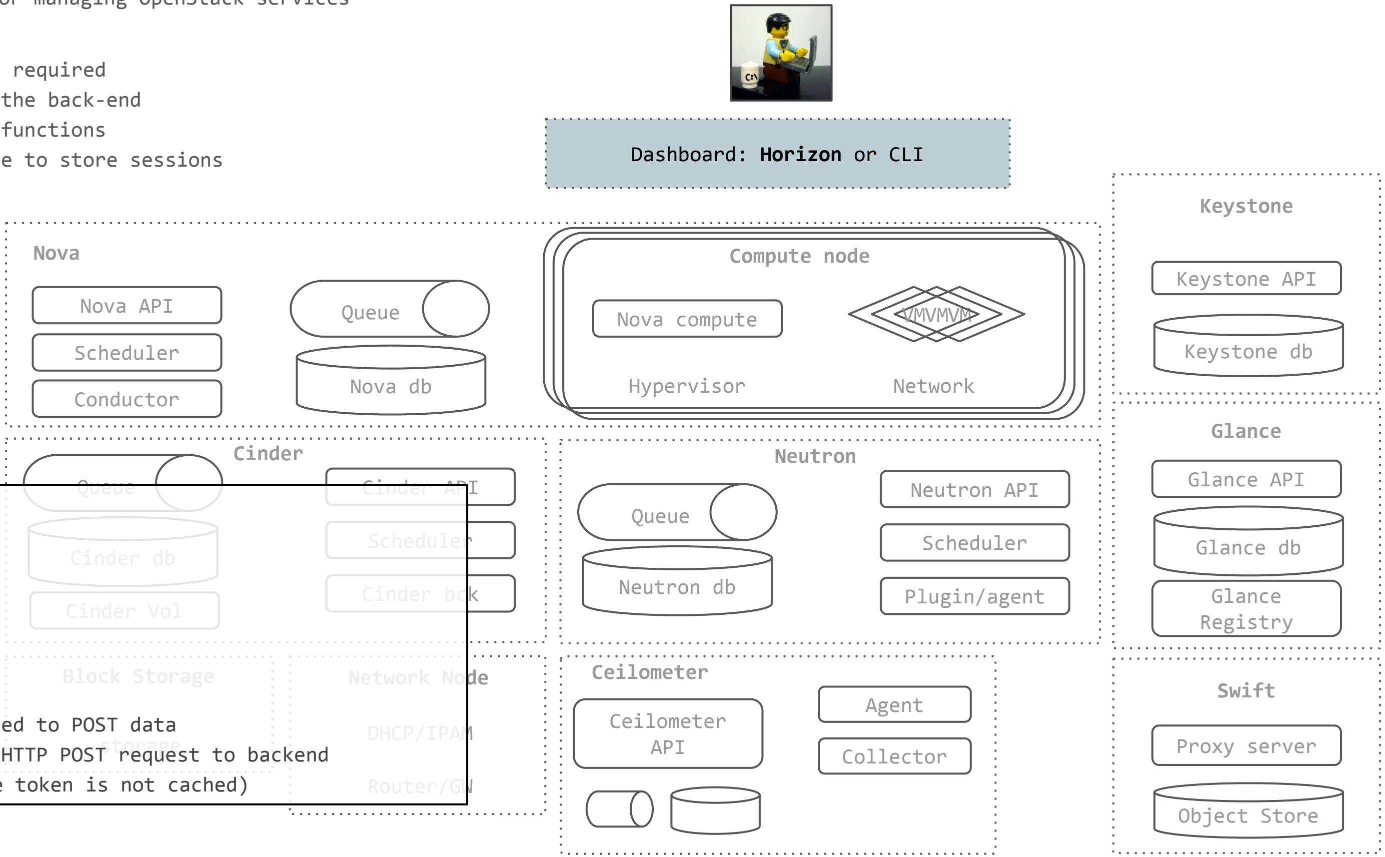
- Most common and complex process
- Involves interaction of most components



Horizon: the OpenStack dashboard

Provides a baseline user interface for managing OpenStack services

- Is “stateless” - no database required
- Delegates error handling to the back-end
- Doesn’t support all the API functions
- Can use memcached or database to store sessions



STEP 1:
User logs in to UI
Specifies VM parameters:

- Name
- Flavor
- Keys
- …

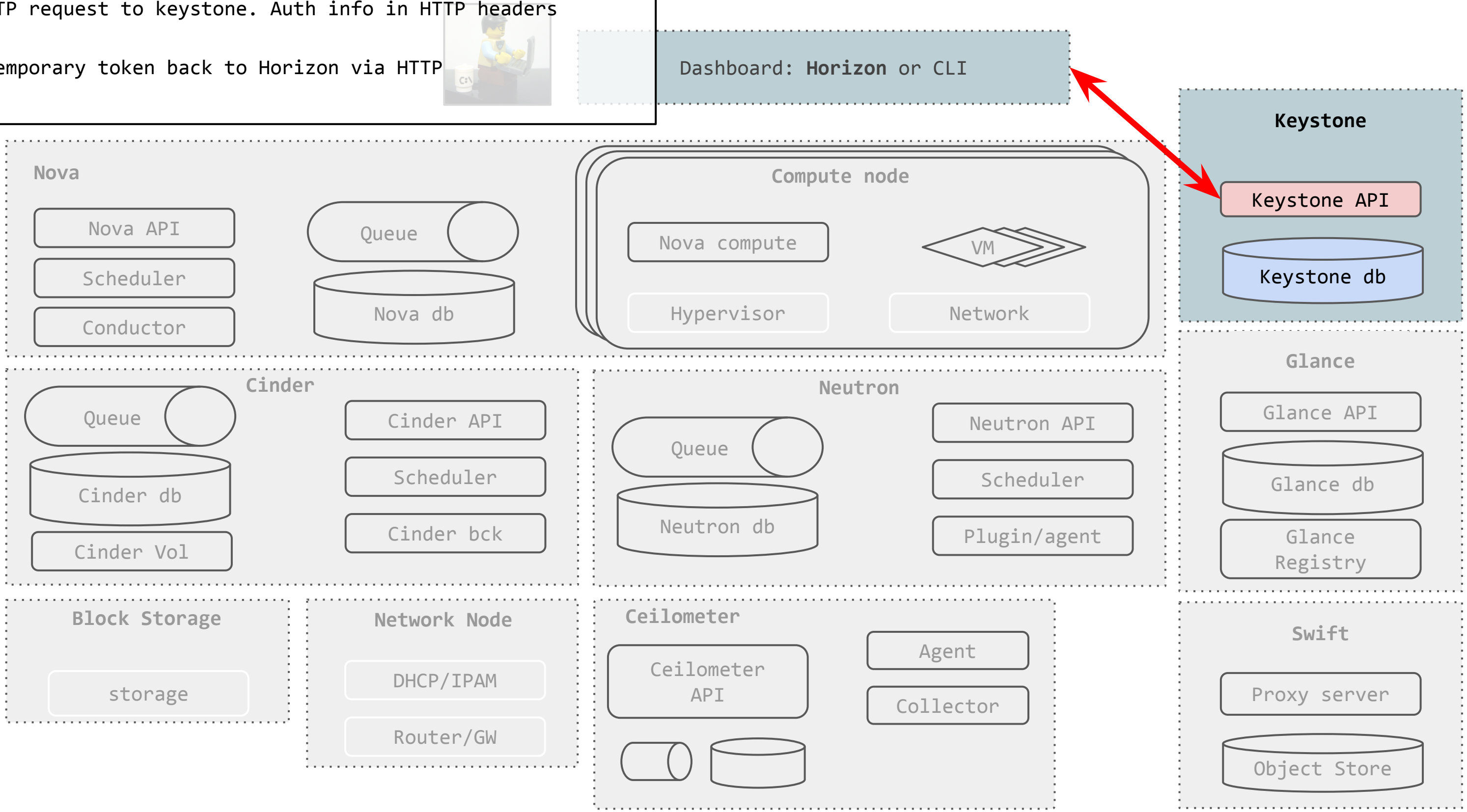
Hits Create button

- Form parameters are converted to POST data
- *“Create”* request initiates HTTP POST request to backend
  - To keystone (if the token is not cached)

STEP 2:

Validate Auth Data:

- Horizon sends HTTP request to keystone. Auth info in HTTP headers
- Keystone sends temporary token back to Horizon via HTTP



K
33
8
mm
22
16
11
8
5.6
4
2.8
KEYSTONE-ELGEET ½"
Keystone
OLYMPIC
BOSTON 24, MASS. U.S.A.
24
12
16
48

Keystone: the OpenStack Identity Service

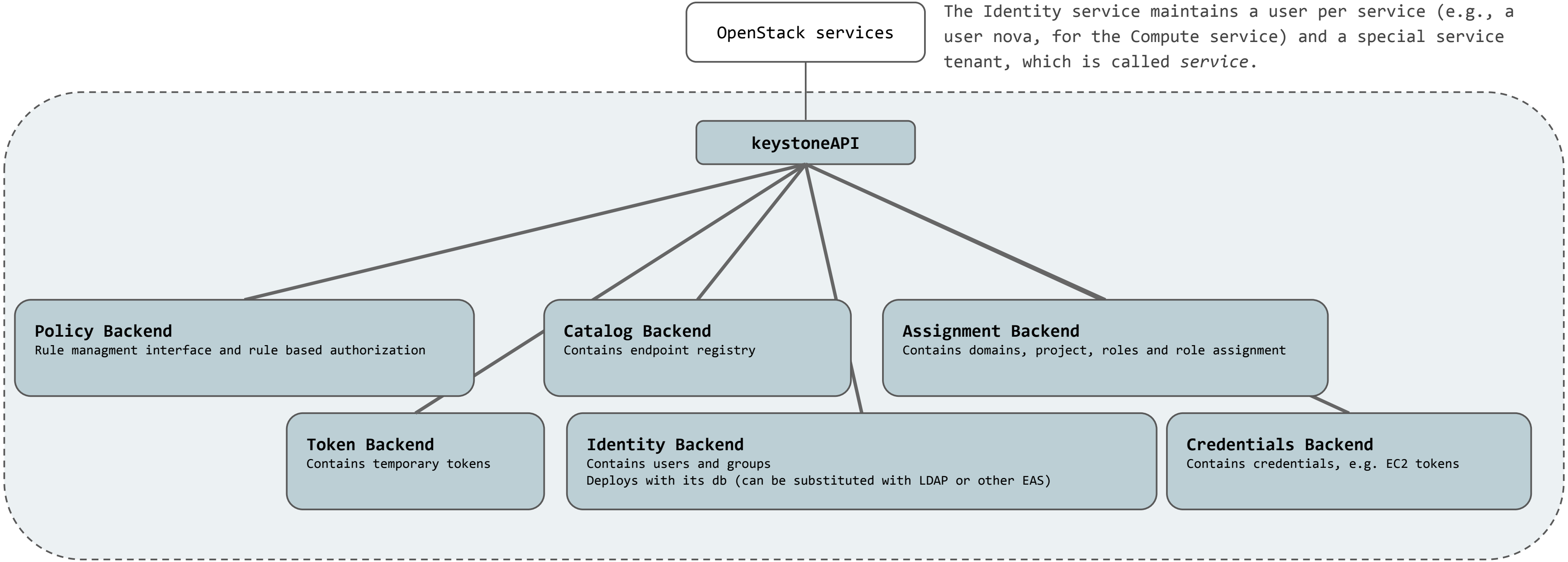
Provides **identity, token, catalog and policy services**

for use specifically by projects in the OpenStack family.

Provides **service catalog** to let other OpenStack systems know where relevant API endpoints for Services.
Two main concepts of Identity service management are:

- **Services**
- **Endpoints**

The Identity service maintains a user per service (e.g., a user nova, for the Compute service) and a special service tenant, which is called *service*.

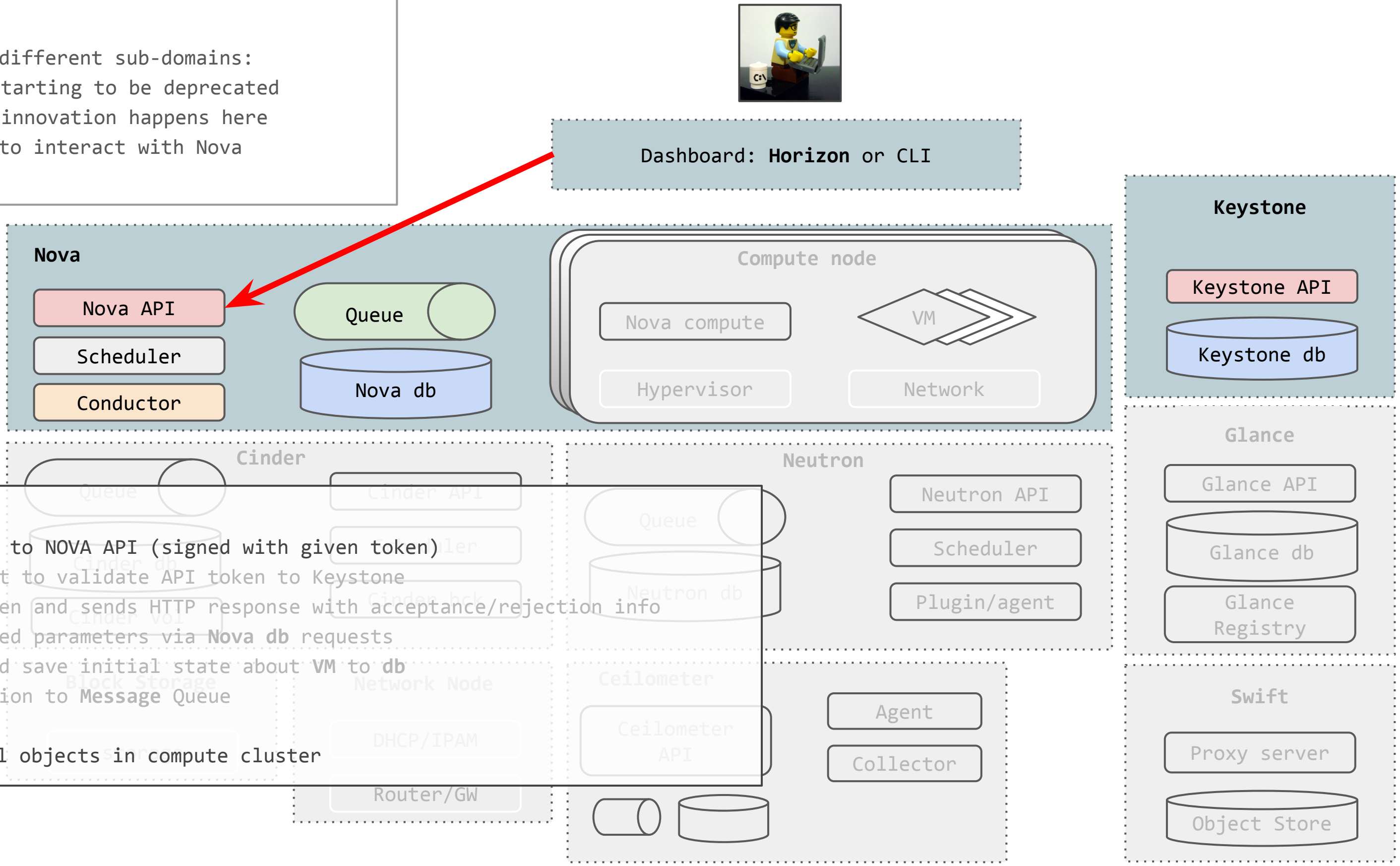

OpenStack services
keystoneAPI
Policy Backend
Rule managment interface and rule based authorization
Catalog Backend
Contains endpoint registry
Assignment Backend
Contains domains, project, roles and role assignment
Token Backend
Contains temporary tokens
Identity Backend
Contains users and groups
Deploys with its db (can be substituted with LDAP or other EAS)
Credentials Backend
Contains credentials, e.g. EC2 tokens

Nova
RADIAL GT

OpenStack Compute API (Nova API)

Nova API is a **RESTful** API web service used to interact with Nova

- Exposes **REST API** via HTTP
- Provides multiple APIs on different sub-domains:
  - **EC2-compatible** - starting to be deprecated
  - **Compute API** - all innovation happens here
- Is the only “allowed” way to interact with Nova
- Is “**stateless**”



STEP3:

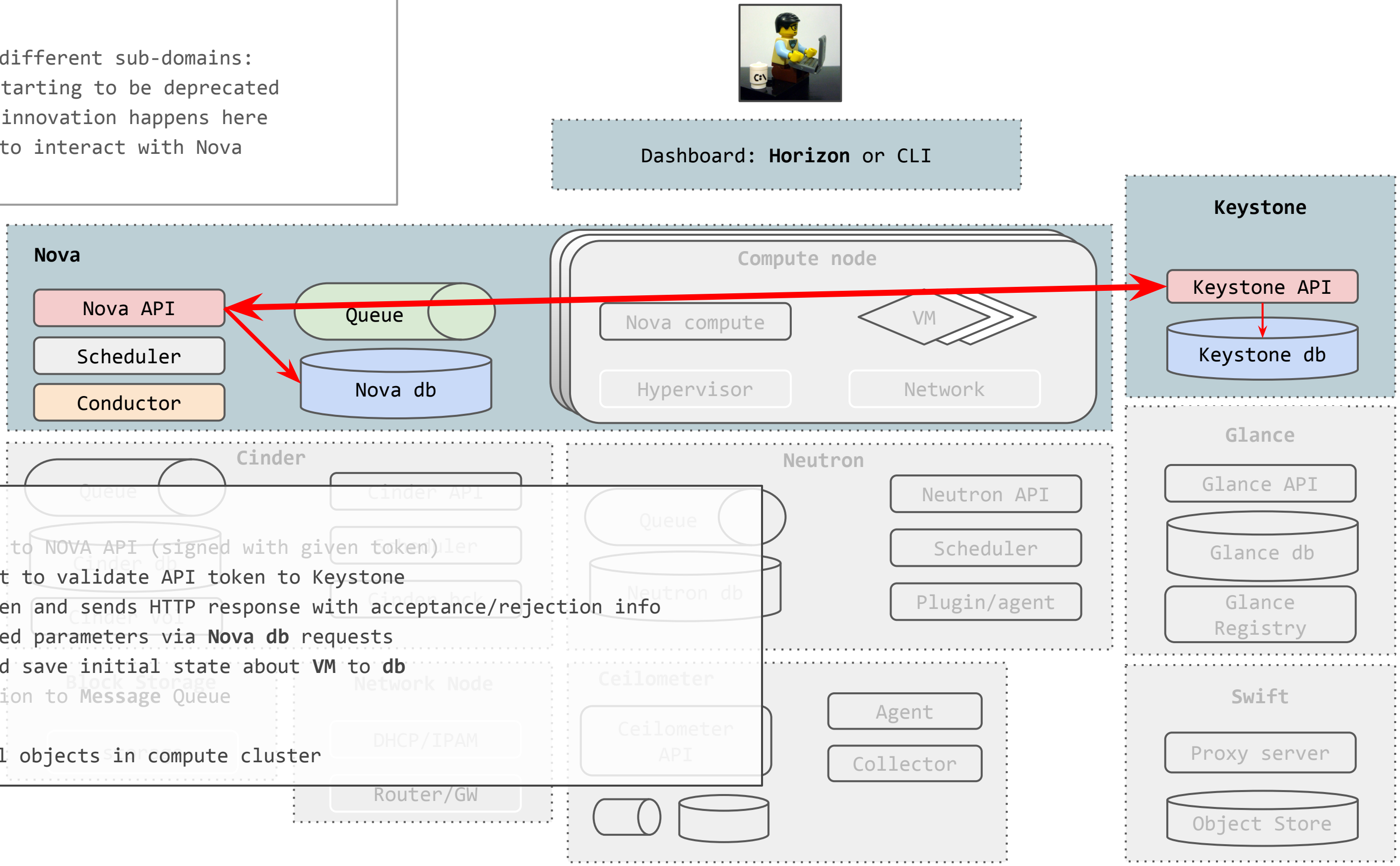
- **Horizon** sends POST request to NOVA API (signed with given token)
- **Nova API** sends HTTP request to validate API token to Keystone
- **Keystone** validates API token and sends HTTP response with acceptance/rejection info
- **Nova** validates cloud-related parameters via **Nova db** requests
- If request can be processed save initial state about **VM** to **db**
- **Send** message with next action to **Message** Queue

Nova db stores current state of all objects in compute cluster

OpenStack Compute API (Nova API)

Nova API is a **RESTful** API web service used to interact with Nova

- Exposes **REST API** via HTTP
- Provides multiple APIs on different sub-domains:
  - **EC2-compatible** - starting to be deprecated
  - **Compute API** - all innovation happens here
- Is the only “allowed” way to interact with Nova
- Is “**stateless**”

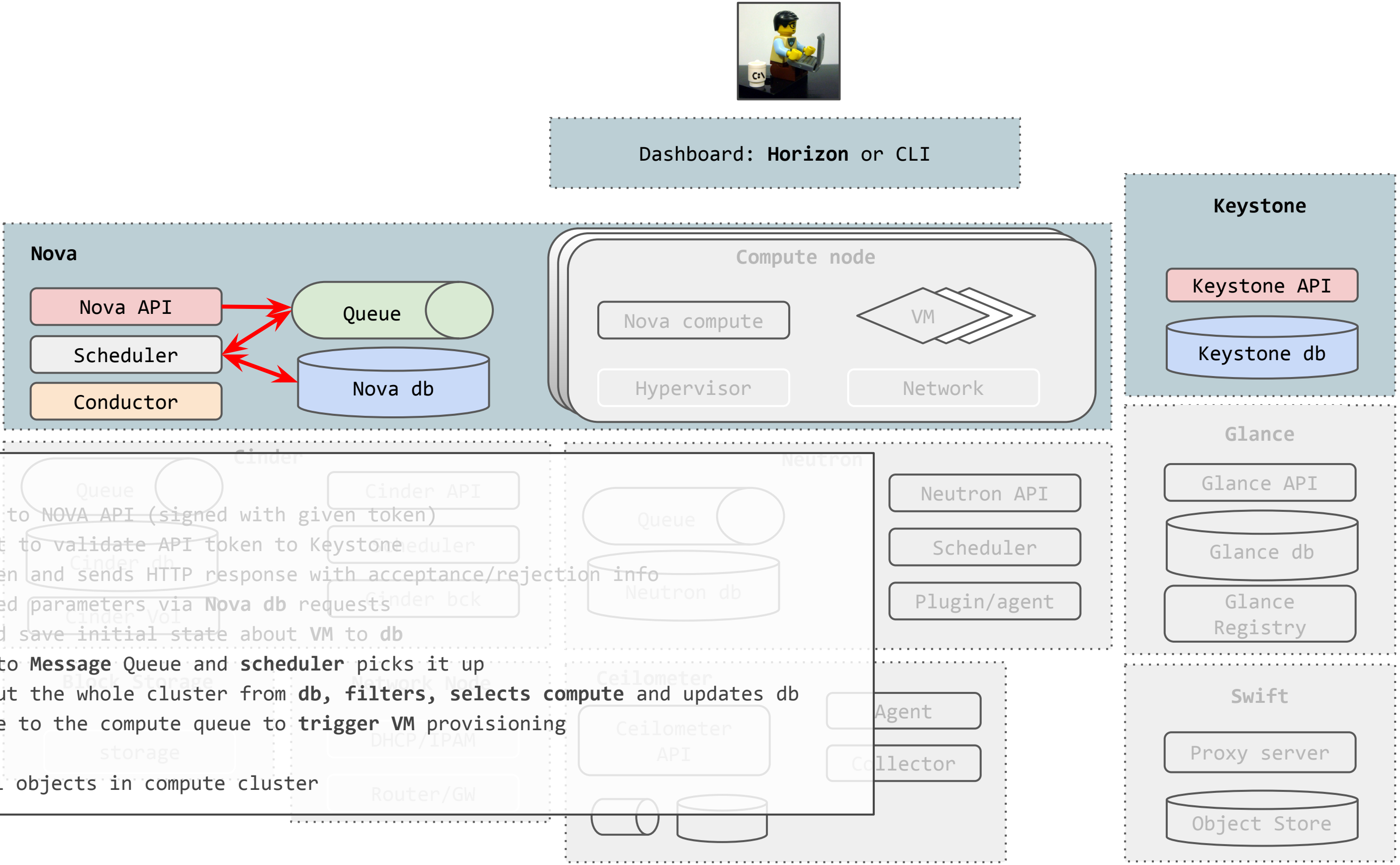


STEP4:

- **Horizon** sends POST request to NOVA API (signed with given token)
- **Nova API** sends HTTP request to validate API token to Keystone
- **Keystone** validates API token and sends HTTP response with acceptance/rejection info
- **Nova** validates cloud-related parameters via **Nova db** requests
- If request can be processed save initial state about **VM** to **db**
- Send message with next action to **Message** Queue

Nova db stores current state of all objects in compute cluster

OpenStack Nova Scheduler



Dashboard: **Horizon** or CLI

Nova

Nova API

Scheduler

Conductor

Queue

Nova db

Compute node

Nova compute

VM

Hypervisor

Network

Keystone

Keystone API

Keystone db

Cinder

Queue

Cinder API

Scheduler

Cinder db

Cinder bck

Cinder Vol

Block Storage

storage

Neutron

Queue

Neutron db

Neutron API

Scheduler

Plugin/agent

Network Node

DHCP/IPAM

Router/GW

Ceilometer

Ceilometer API

Agent

Collector

Glance

Glance API

Glance db

Glance Registry

Swift

Proxy server

Object Store

STEP5:

- **Horizon** sends POST request to NOVA API (signed with given token)
- **Nova API** sends HTTP request to validate API token to Keystone
- **Keystone** validates API token and sends HTTP response with acceptance/rejection info
- **Nova** validates cloud-related parameters via **Nova db** requests
- If request can be processed save initial state about VM to **db**
- **Send** message with **VM info** to **Message** Queue and **scheduler** picks it up
- **Scheduler** fetches info about the whole cluster from **db, filters, selects compute** and updates db
- **Scheduler** publishes message to the compute queue to **trigger** VM provisioning

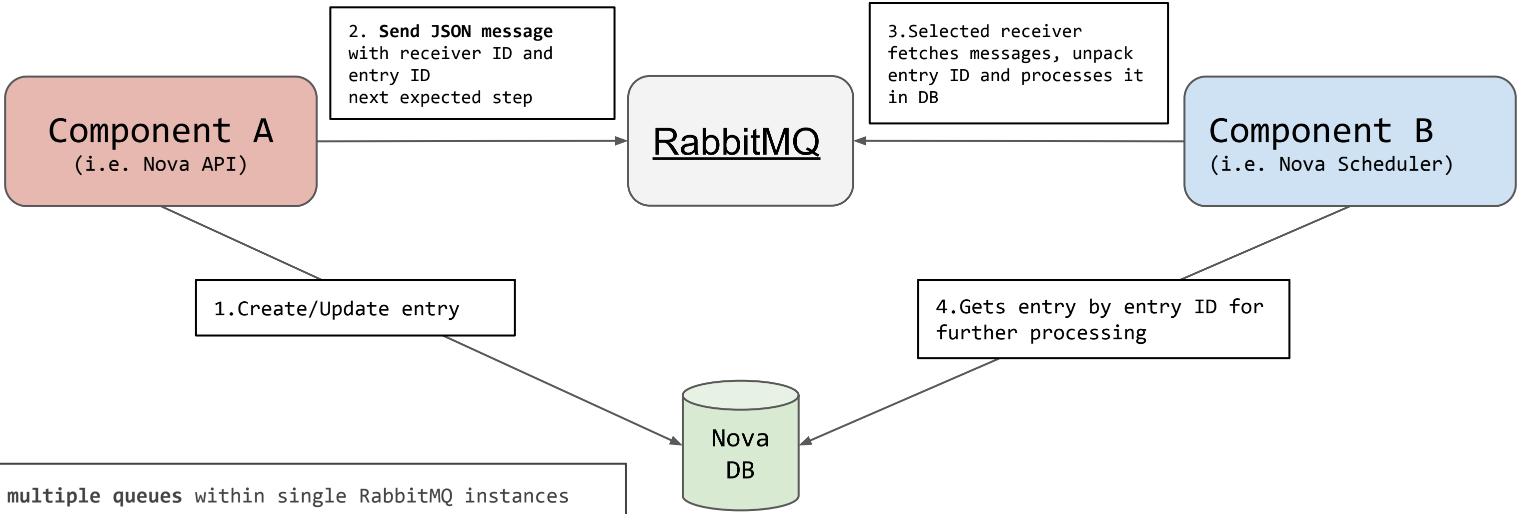
Nova db stores current state of all objects in compute cluster

RabbitMQ

© Michelle Middleton

Message Queue

Is a unified way for collaboration between sub-components


Component A
(i.e. Nova API)
2. **Send JSON message** with receiver ID and entry ID
next expected step
RabbitMQ
3.Selected receiver fetches messages, unpack entry ID and processes it in DB
Component B
(i.e. Nova Scheduler)
1.Create/Update entry
4.Gets entry by entry ID for further processing
Nova DB

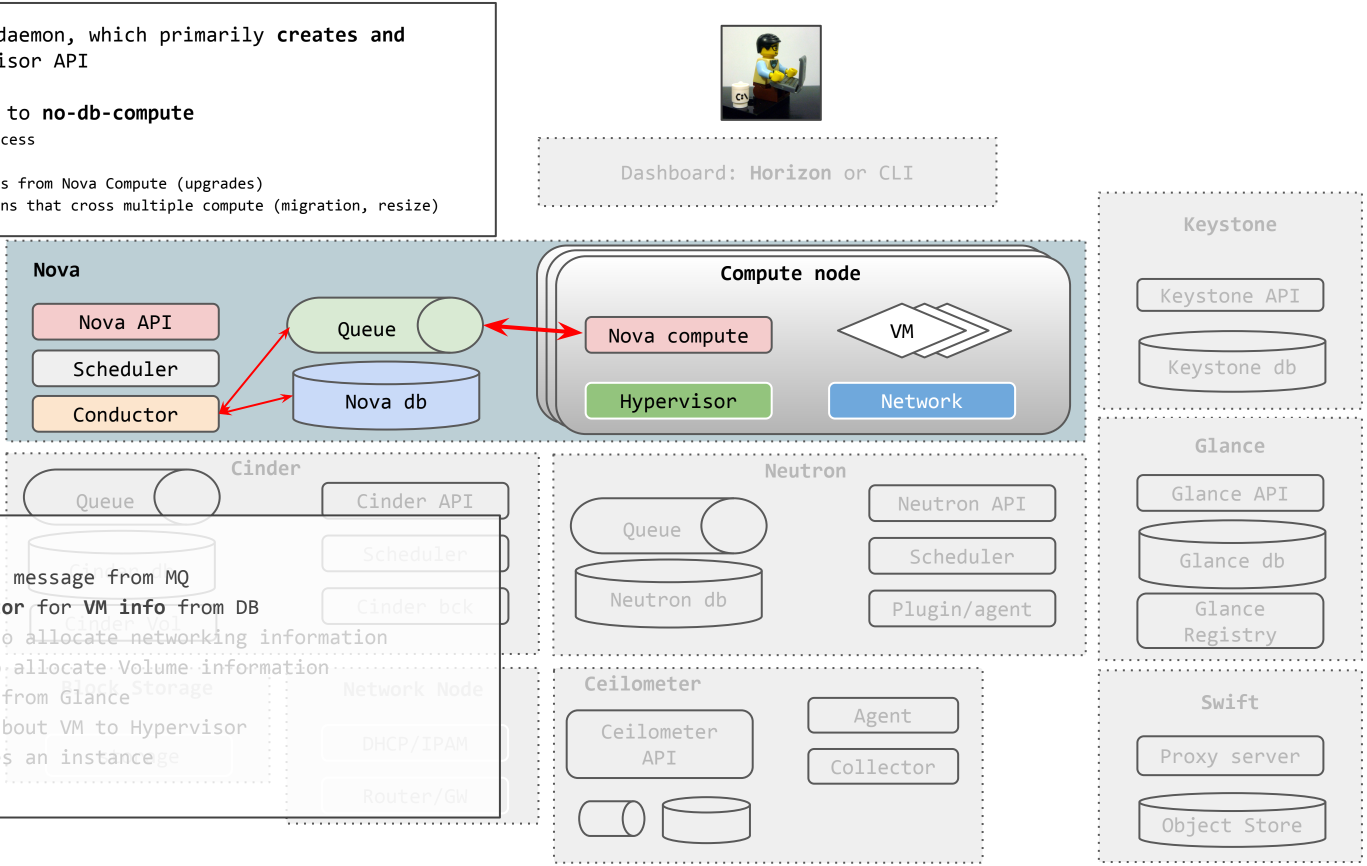
- Uses **multiple queues** within single RabbitMQ instances
  - Used by services to build machine state
  - Each compute node has a queue
- Message traffic is **not intensive**
- **No broadcast** messages
  - Monitoring uses API polling
- **HA** should be be configured **separately**
  - Mirror queues not handled by OpenStack

Nova Compute

Nova Compute is a worker daemon, which primarily **creates and terminates VMs** via Hypervisor API

Nova Conductor is the key to **no-db-compute**

- Eliminates remote db access
- Horizontal scalability
- Hides db implementations from Nova Compute (upgrades)
- Beneficial for operations that cross multiple compute (migration, resize)



STEP6:

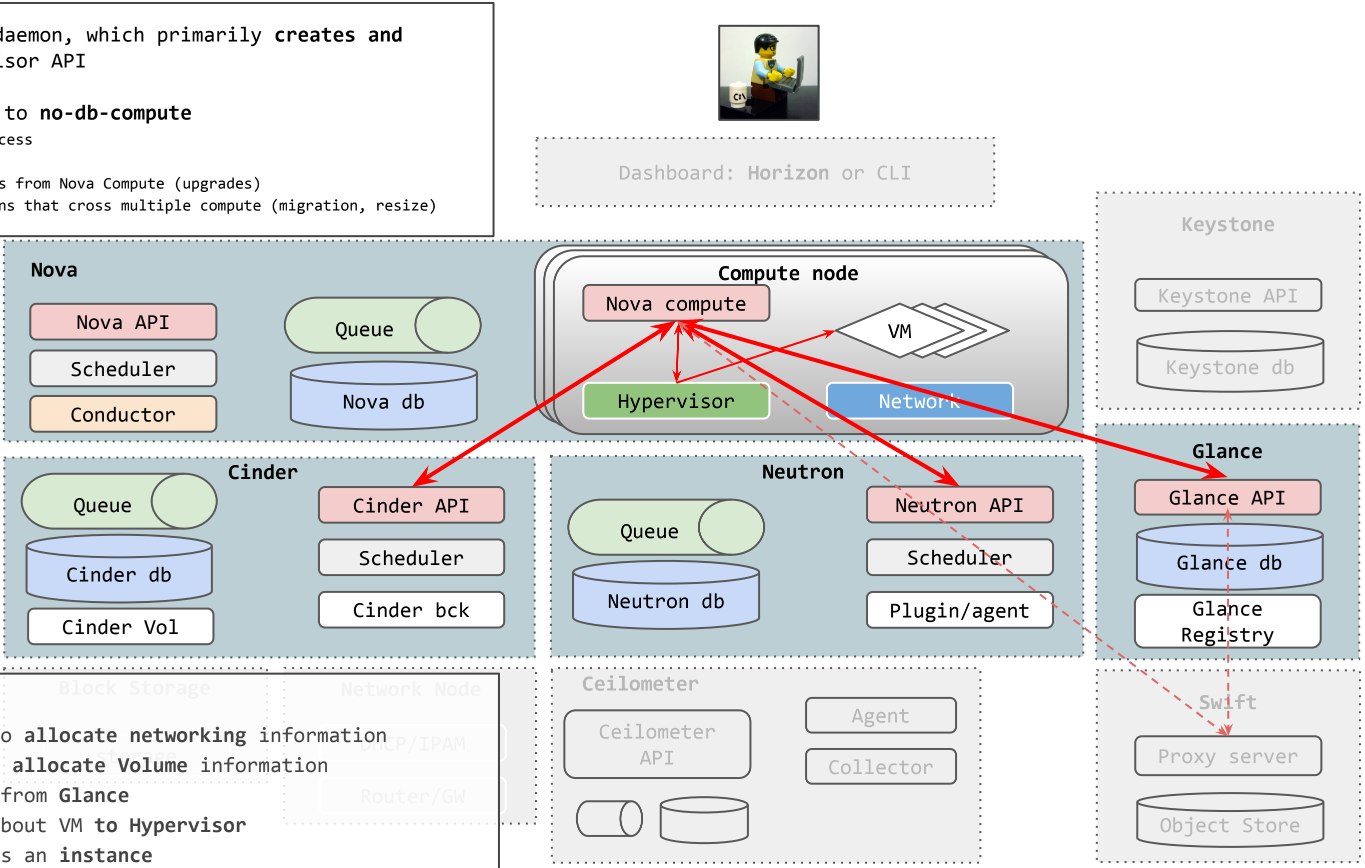
- **Nova Compute** gets message from MQ
- Asks **Nova Conductor** for **VM info** from DB
- Queries Neutron to allocate networking information
- Queries Cinder to allocate Volume information
- Fetches VM image from Glance
- Passes all info about VM to Hypervisor
- Hypervisor creates an instance

Nova Compute

Nova Compute is a worker daemon, which primarily **creates and terminates VMs** via Hypervisor API

Nova Conductor is the key to **no-db-compute**

- Eliminates remote db access
- Horizontal scalability
- Hides db implementations from Nova Compute (upgrades)
- Beneficial for operations that cross multiple compute (migration, resize)



STEP7:

- **Queries** Neutron to **allocate networking** information
- **Queries** Cinder to **allocate Volume** information
- Fetches VM **image** from **Glance**
- Passes all info about VM **to Hypervisor**
- **Hypervisor** creates an **instance**

Nova Compute drivers (for reference)

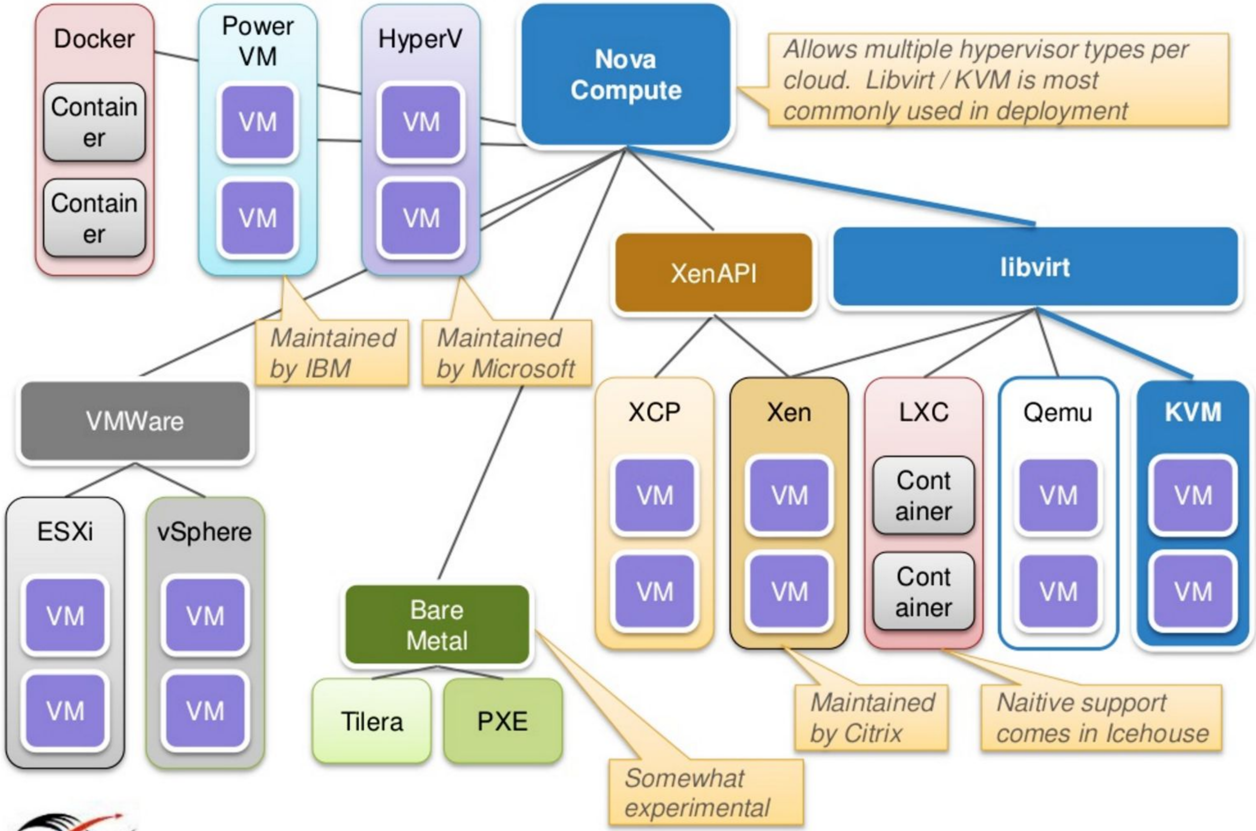


Image Courtesy of MIRANTIS

glance

Glance: OpenStack image service

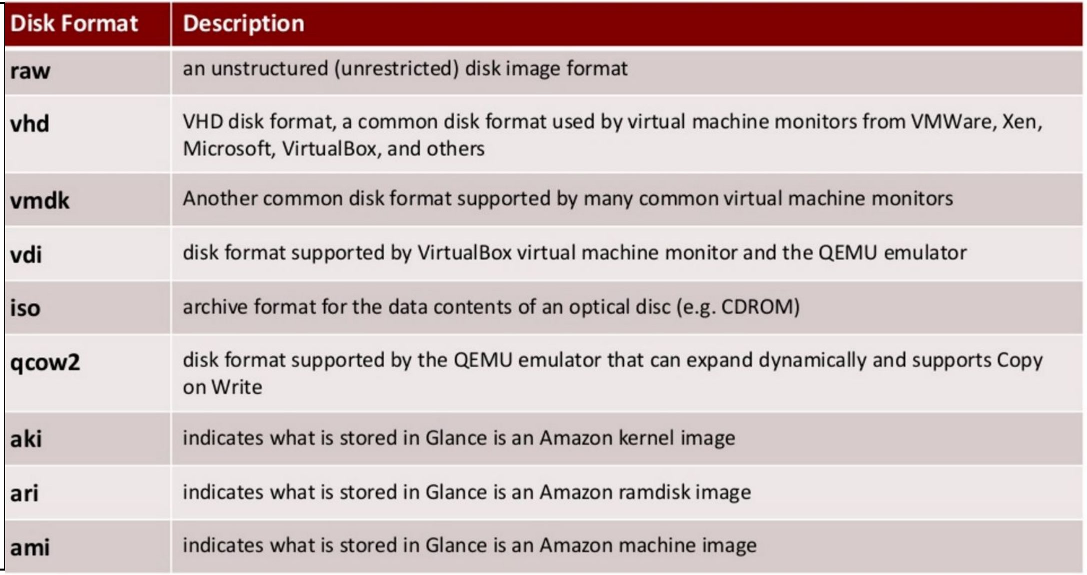
Provides services for:

- Discovering
- Registering
- Retrieving virtual machine images

May use **multiple backends** for image storage

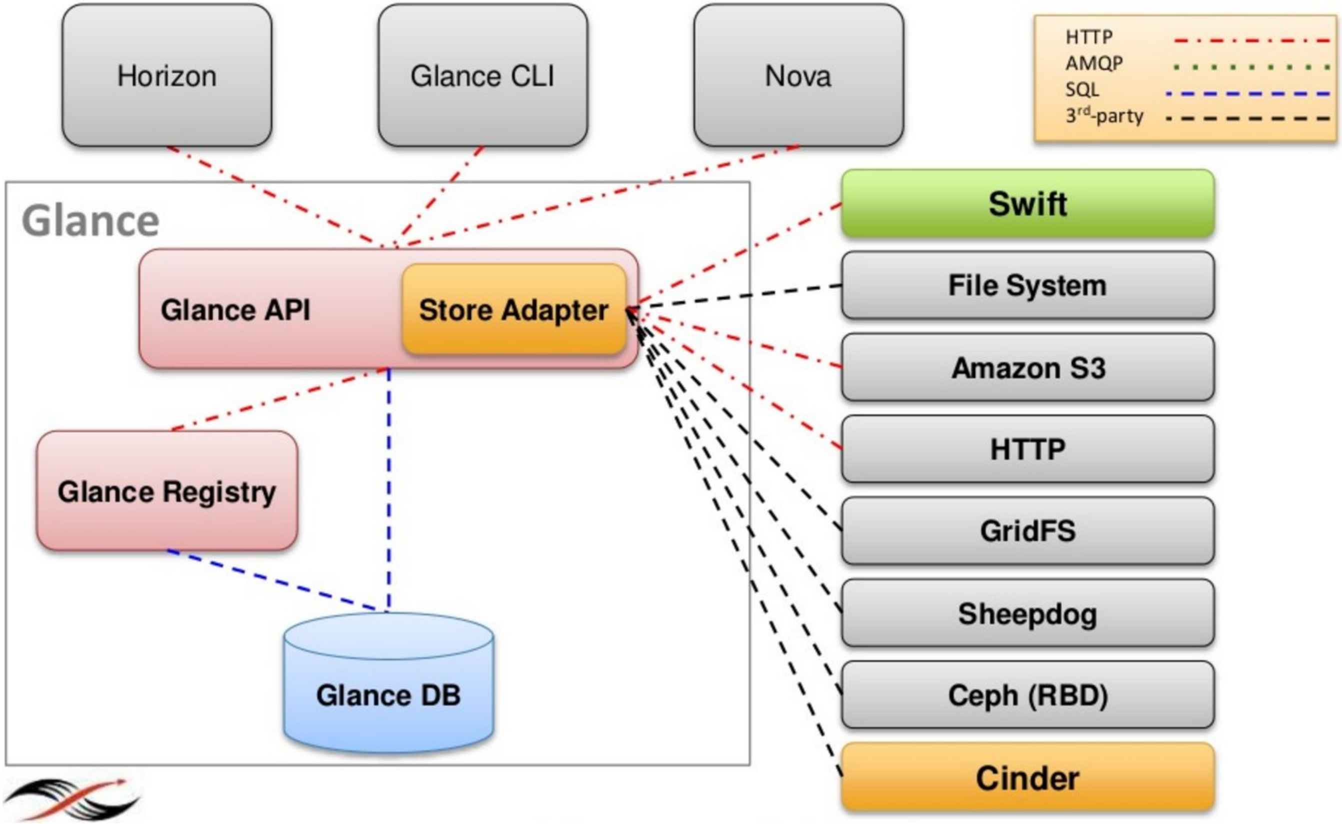
May store the same image in multiple locations

Supports **multiple** image **formats**



Disk Format	Description
raw	an unstructured (unrestricted) disk image format
vhd	VHD disk format, a common disk format used by virtual machine monitors from VMWare, Xen, Microsoft, VirtualBox, and others
vmdk	Another common disk format supported by many common virtual machine monitors
vdi	disk format supported by VirtualBox virtual machine monitor and the QEMU emulator
iso	archive format for the data contents of an optical disc (e.g. CDROM)
qcow2	disk format supported by the QEMU emulator that can expand dynamically and supports Copy on Write
aki	indicates what is stored in Glance is an Amazon kernel image
ari	indicates what is stored in Glance is an Amazon ramdisk image
ami	indicates what is stored in Glance is an Amazon machine image

Glance architecture



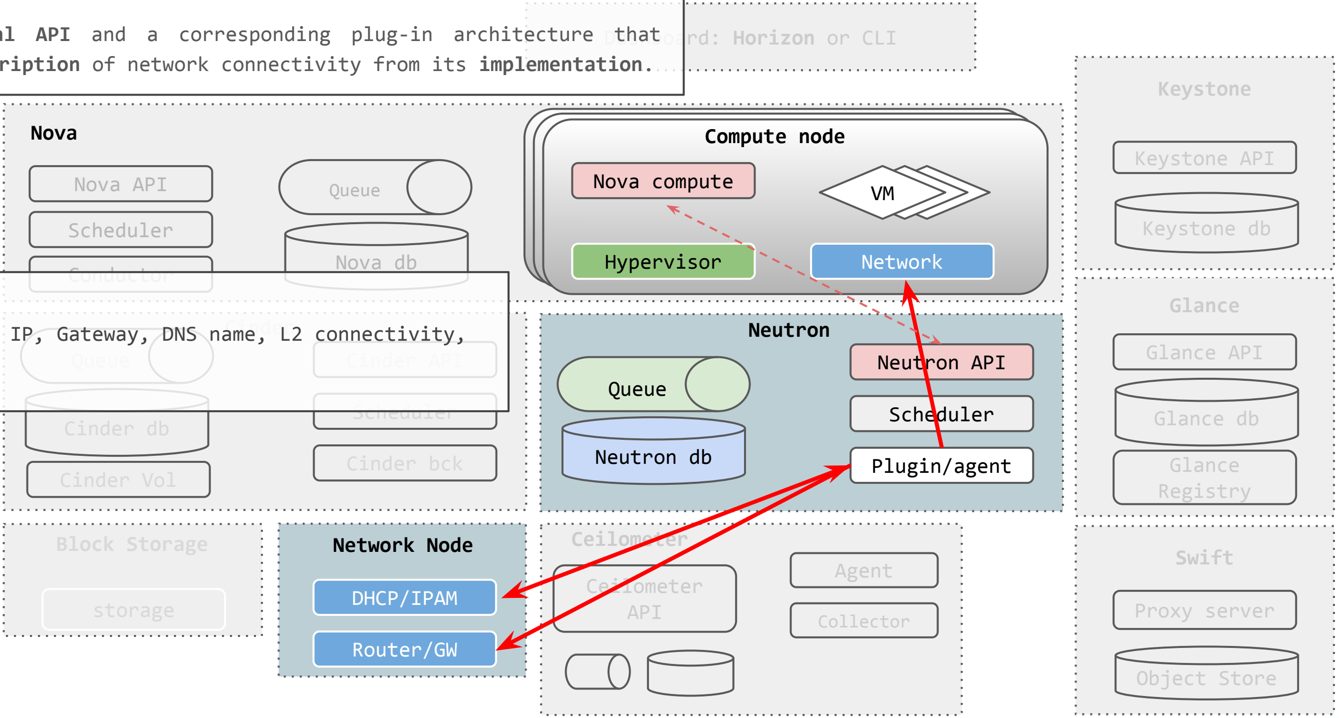
neutron

Openstack networking:Neutron (configure Network)

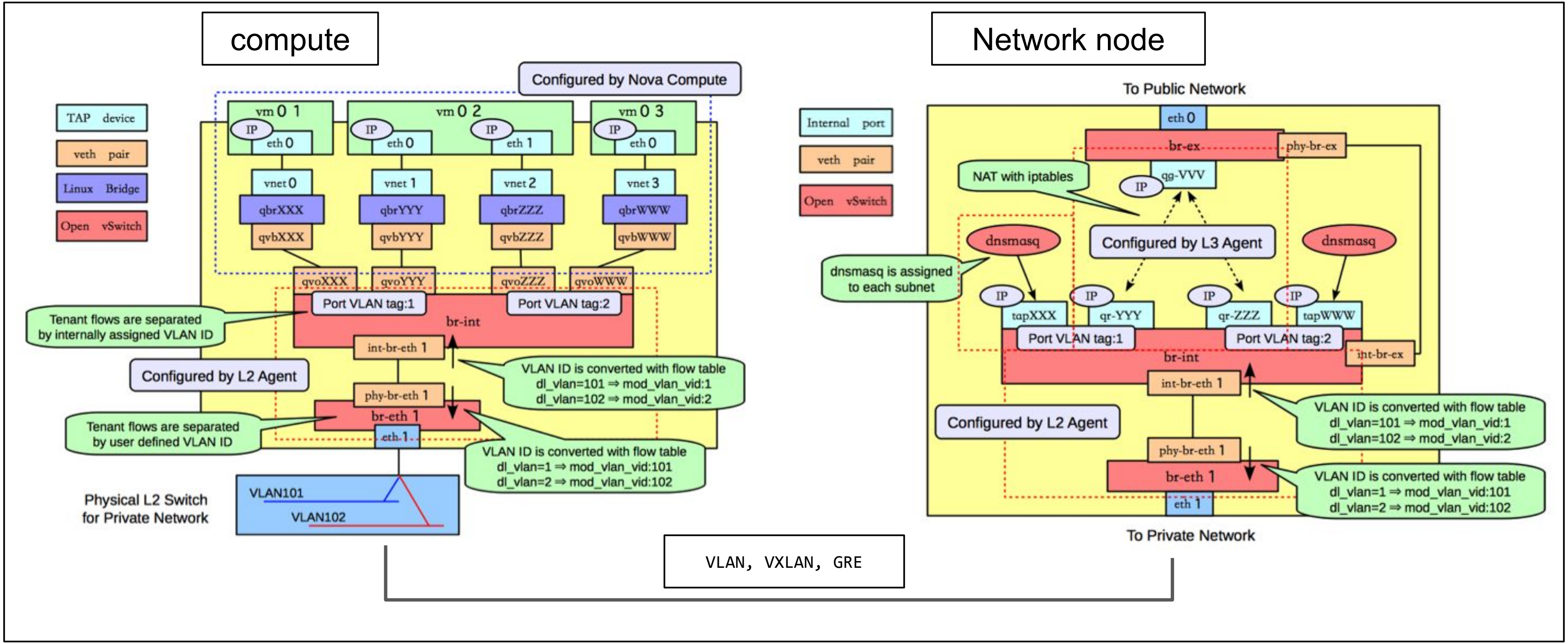
- Provides a flexible **API (POST/GET)** for service providers or their tenants **to manage OpenStack network topologies**.
    - **Create networks, associate VMs, set routers**, etc.
- Presents a **logical API** and a corresponding plug-in architecture that **separates** the **description** of network connectivity from its **implementation**.

STEP9:

- Neutron configures IP, Gateway, DNS name, L2 connectivity, etc



Networking in (too many) details



compute

Network node

VLAN, VXLAN, GRE

Configured by Nova Compute

Tenant flows are separated by internally assigned VLAN ID

Configured by L2 Agent

Tenant flows are separated by user defined VLAN ID

VLAN ID is converted with flow table
dl_vlan=101 ⇒ mod_vlan_vid:1
dl_vlan=102 ⇒ mod_vlan_vid:2

VLAN ID is converted with flow table
dl_vlan=1 ⇒ mod_vlan_vid:101
dl_vlan=2 ⇒ mod_vlan_vid:102

Physical L2 Switch for Private Network

To Public Network

NAT with iptables

Configured by L3 Agent

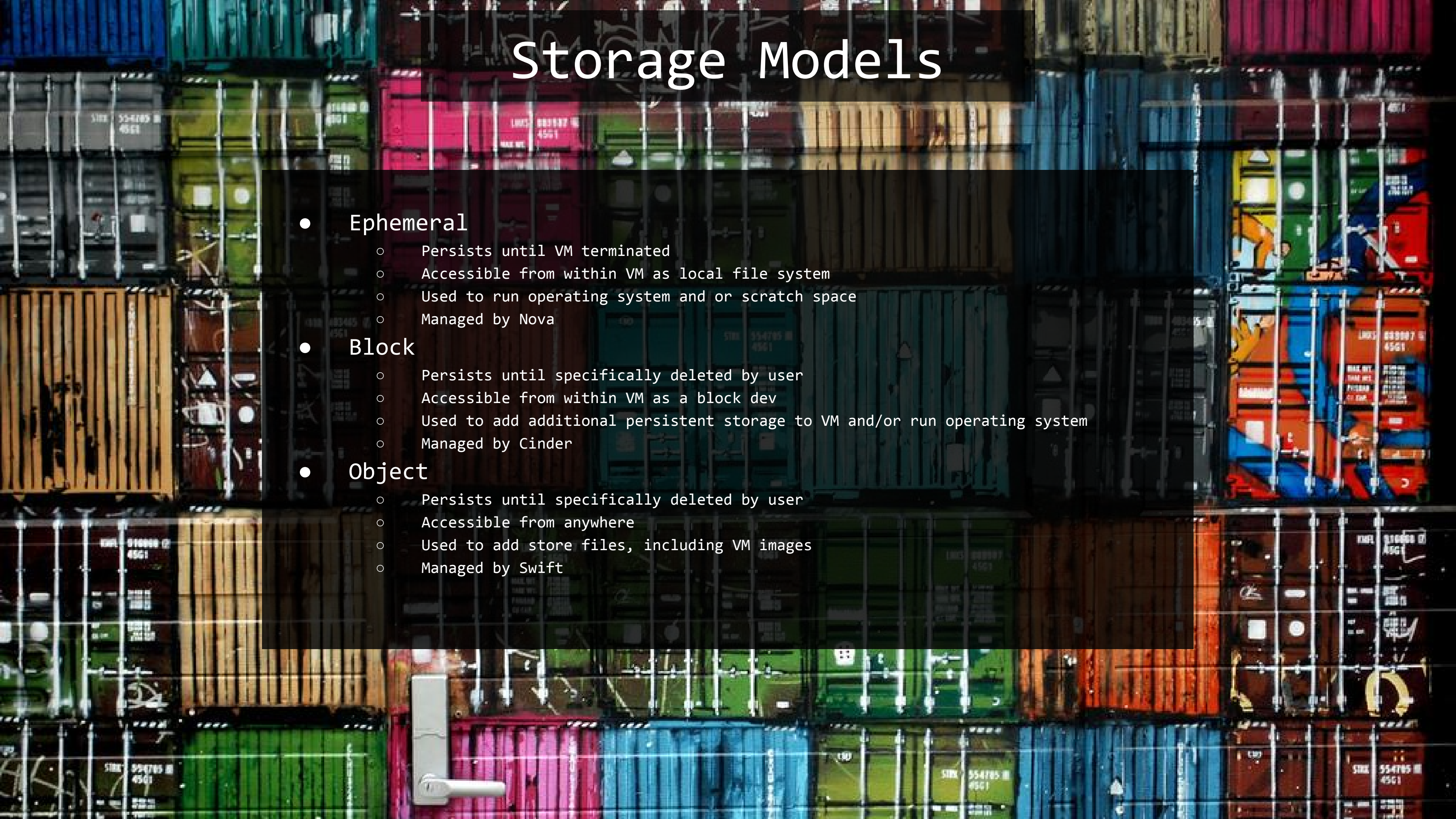
dnsmasq is assigned to each subnet

Configured by L2 Agent

VLAN ID is converted with flow table
dl_vlan=101 ⇒ mod_vlan_vid:1
dl_vlan=102 ⇒ mod_vlan_vid:2

VLAN ID is converted with flow table
dl_vlan=1 ⇒ mod_vlan_vid:101
dl_vlan=2 ⇒ mod_vlan_vid:102

To Private Network



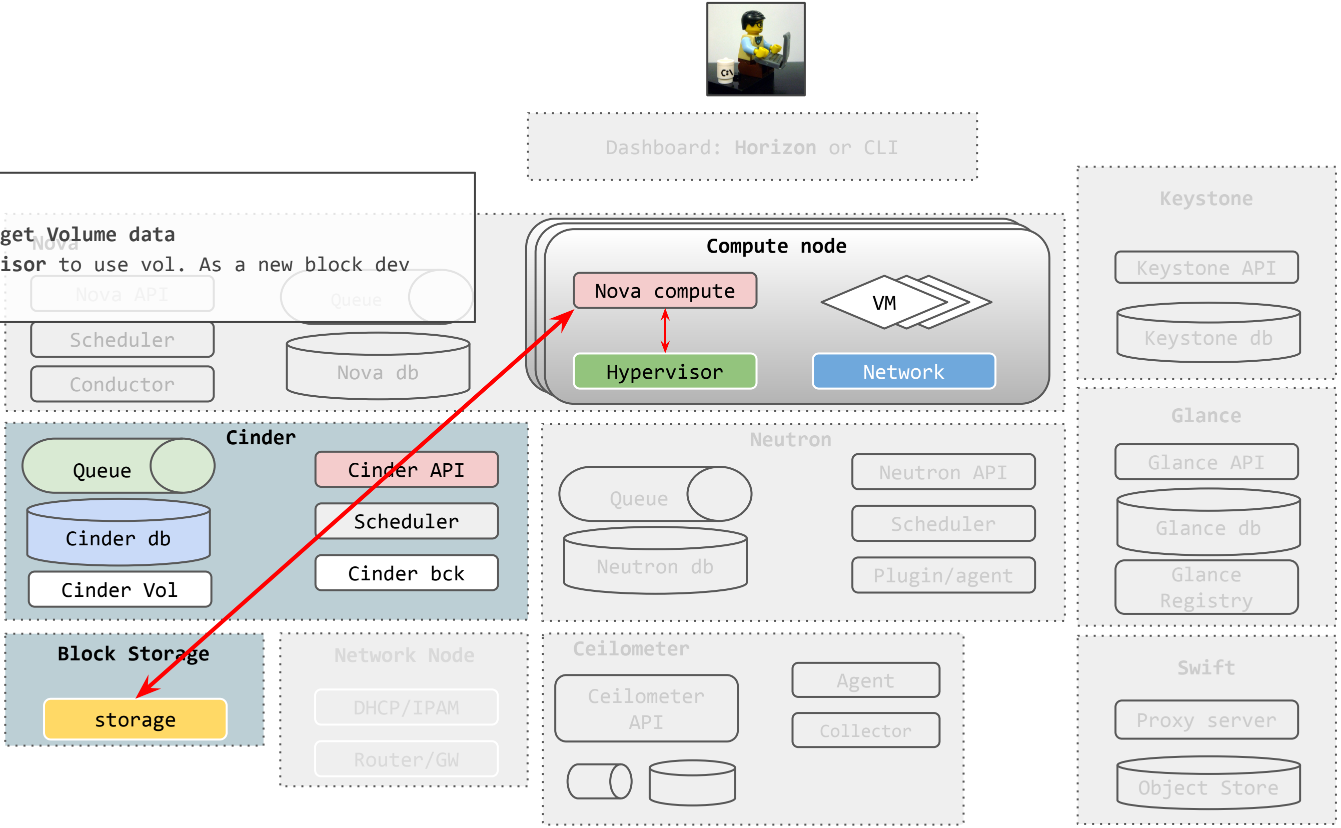
Storage Models

- Ephemeral
  - Persists until VM terminated
  - Accessible from within VM as local file system
  - Used to run operating system and or scratch space
  - Managed by Nova
- Block
  - Persists until specifically deleted by user
  - Accessible from within VM as a block dev
  - Used to add additional persistent storage to VM and/or run operating system
  - Managed by Cinder
- Object
  - Persists until specifically deleted by user
  - Accessible from anywhere
  - Used to add store files, including VM images
  - Managed by Swift

Nova Compute (Requests Volume)

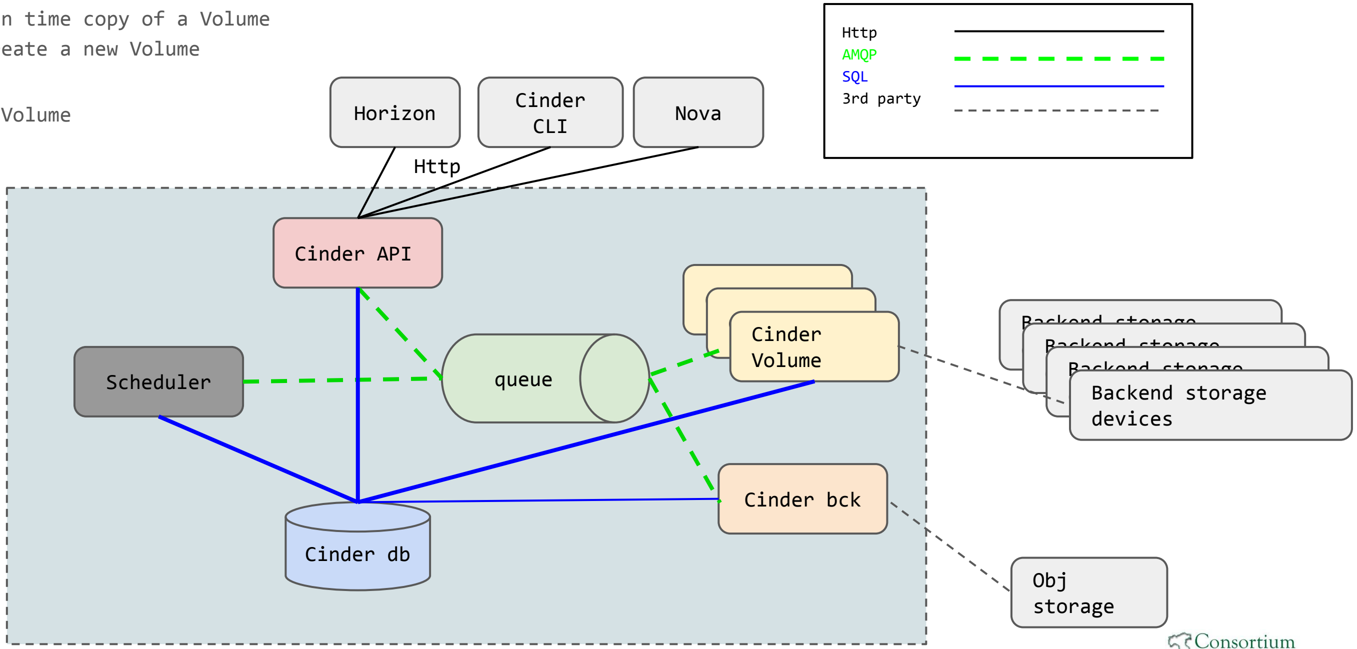
STEP8:

- contacts **Cinder** to **get Volume data**
- **Instruct** the **Hypervisor** to use vol. As a new block dev

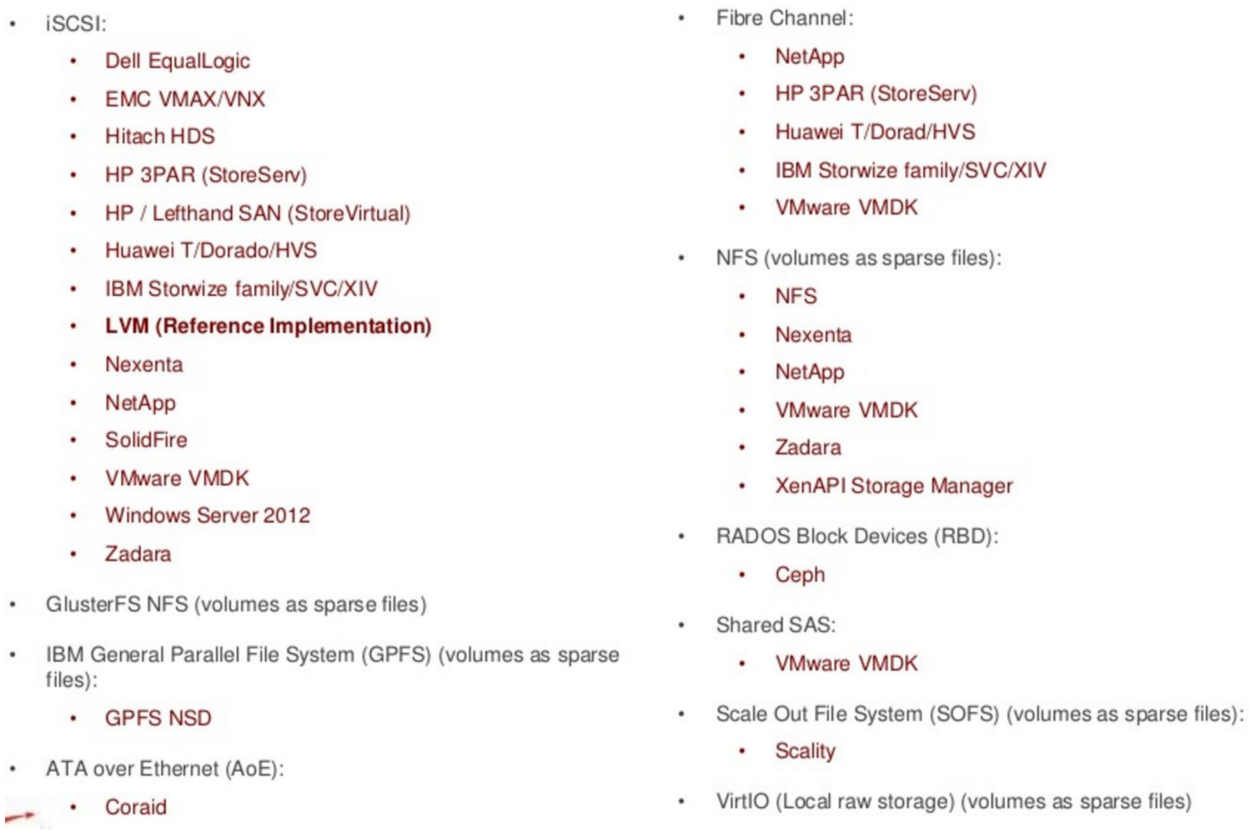


Cinder resources: OpenStack block storage

- Volume
  - Is a persistent R/W block storage device
  - Can be attached to VMs as a secondary storage
  - Can be root store to boot VMs
  - Can be attached only to one instance at a time
  - Keeps its state independent of an instance
- Snapshot
  - Is a read only point in time copy of a Volume
  - Can then be used to create a new Volume
- Backup
  - An archived copy of a Volume

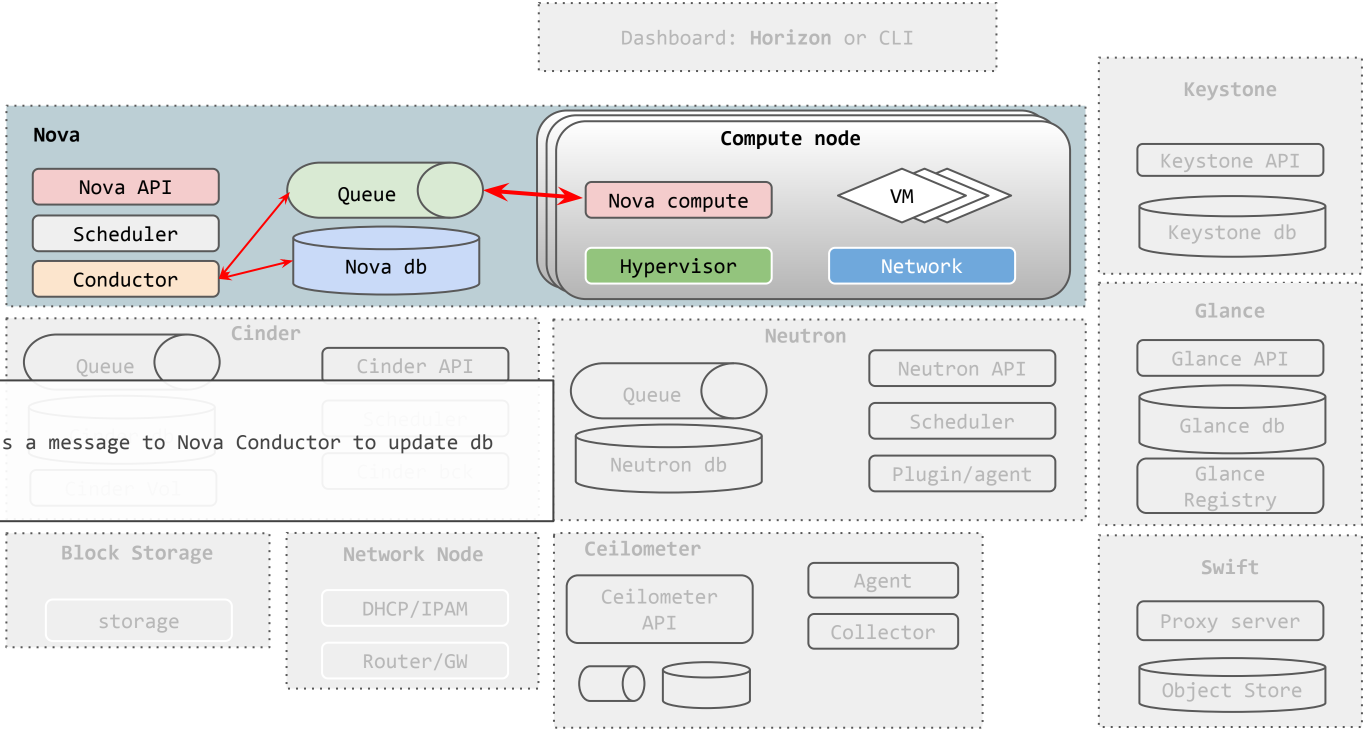


Cinder Volume driver (for reference)



- iSCSI:
  - Dell EqualLogic
  - EMC VMAX/VNX
  - Hitach HDS
  - HP 3PAR (StoreServ)
  - HP / Lefthand SAN (StoreVirtual)
  - Huawei T/Dorado/HVS
  - IBM Storwize family/SVC/XIV
  - **LVM (Reference Implementation)**
  - Nexenta
  - NetApp
  - SolidFire
  - VMware VMDK
  - Windows Server 2012
  - Zadara
- GlusterFS NFS (volumes as sparse files)
- IBM General Parallel File System (GPFS) (volumes as sparse files):
  - GPFS NSD
- ATA over Ethernet (AoE):
  - Coraid
- Fibre Channel:
  - NetApp
  - HP 3PAR (StoreServ)
  - Huawei T/Dorad/HVS
  - IBM Storwize family/SVC/XIV
  - VMware VMDK
- NFS (volumes as sparse files):
  - NFS
  - Nexenta
  - NetApp
  - VMware VMDK
  - Zadara
  - XenAPI Storage Manager
- RADOS Block Devices (RBD):
  - Ceph
- Shared SAS:
  - VMware VMDK
- Scale Out File System (SOFS) (volumes as sparse files):
  - Scality
- VirtIO (Local raw storage) (volumes as sparse files)

VM is up



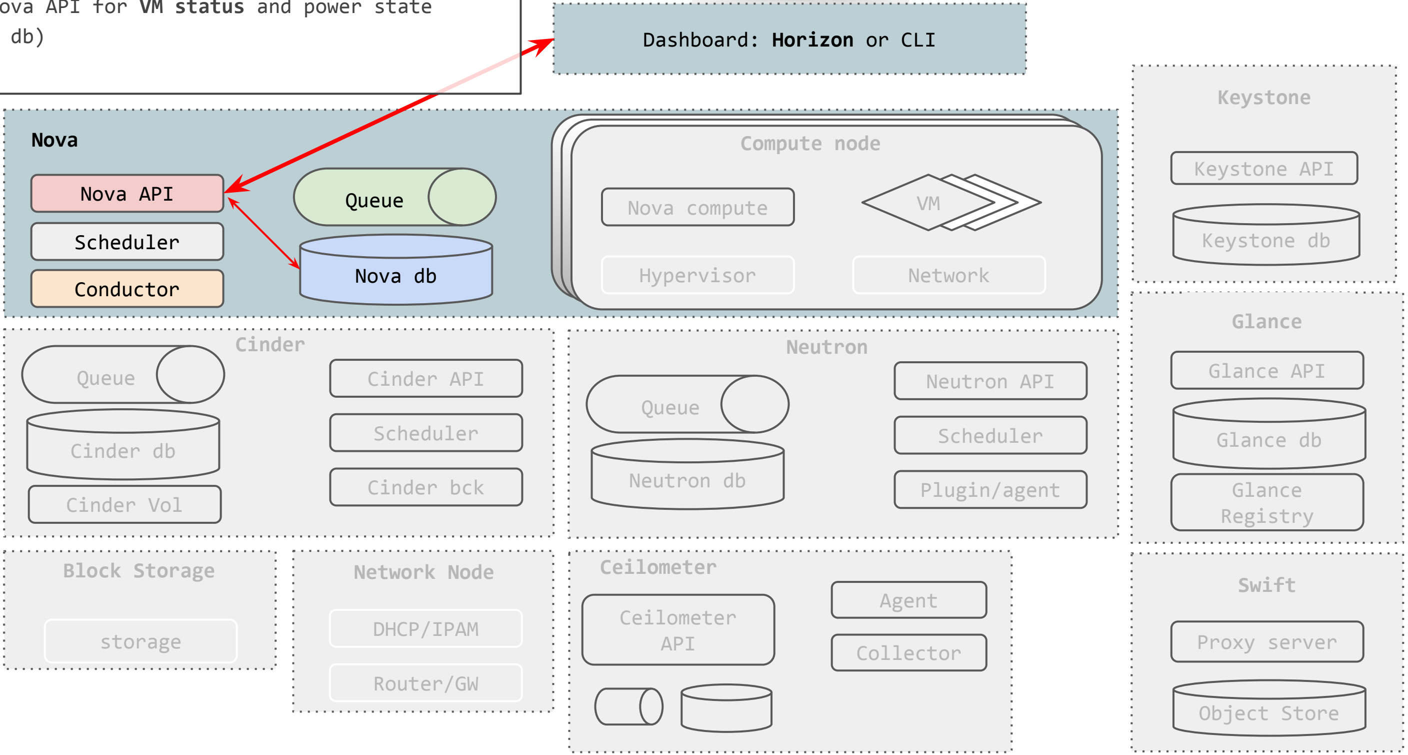
STEP10:

- **Nova Compute** sends a message to Nova Conductor to update db with VM state

User is happy

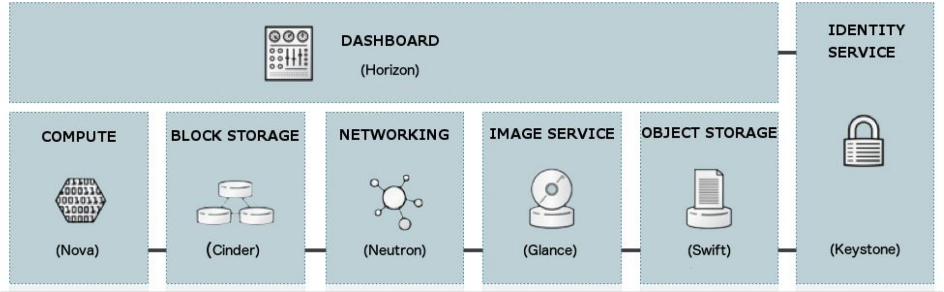
STEP 11:

- Horizon polls Nova API for **VM status** and power state (taken from the db)



OpenStack Architecture recap

- Users log into **Horizon** and initiate a VM create
- **Keystone** authorizes
- **Nova** initiates provisioning and saves state to DB
- **Nova Scheduler** finds appropriate host
- **Neutron** configures networking
- **Cinder** provides block device
- Image URI is looked up through **Glance**
- Image is retrieved via **Swift**
- VM is rendered by Hypervisor



GARR Cloud Infrastructure

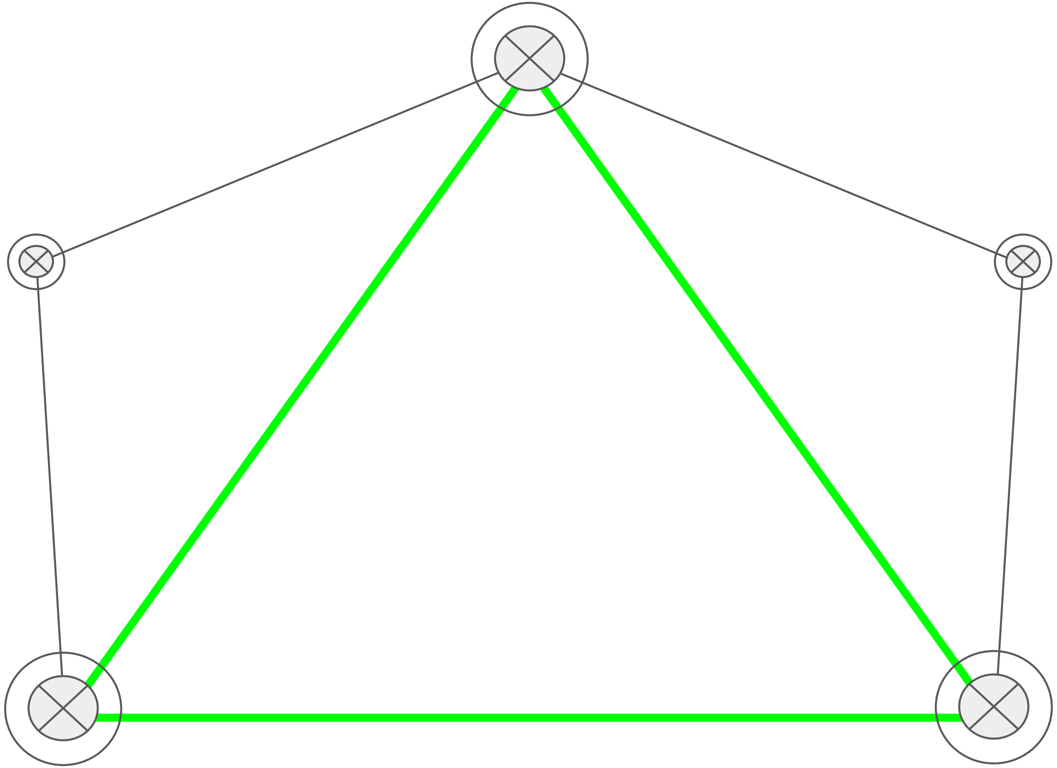
A.Marfia

8500 core
10 PB

... 11 rack/CSD-modules

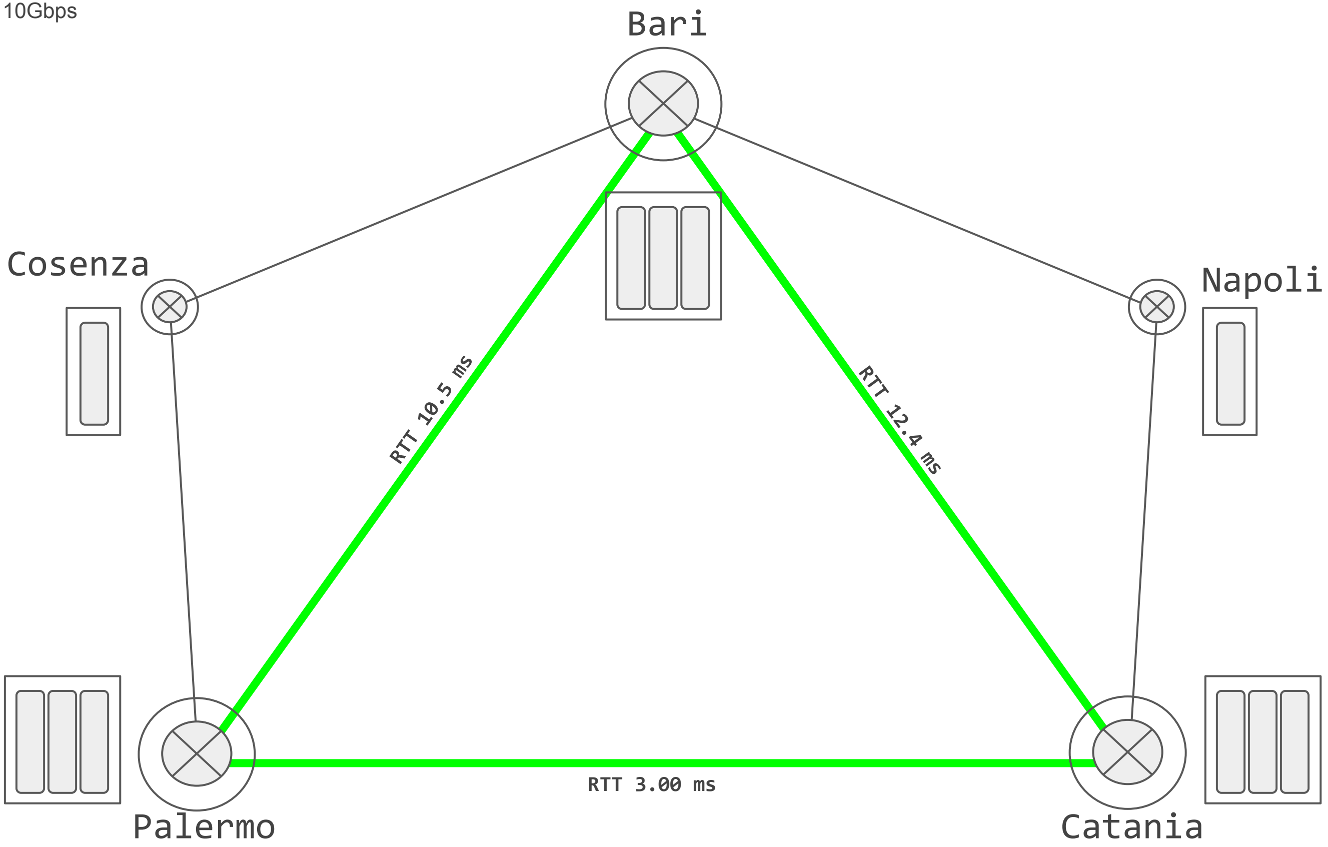
Network

40Gbps
10Gbps

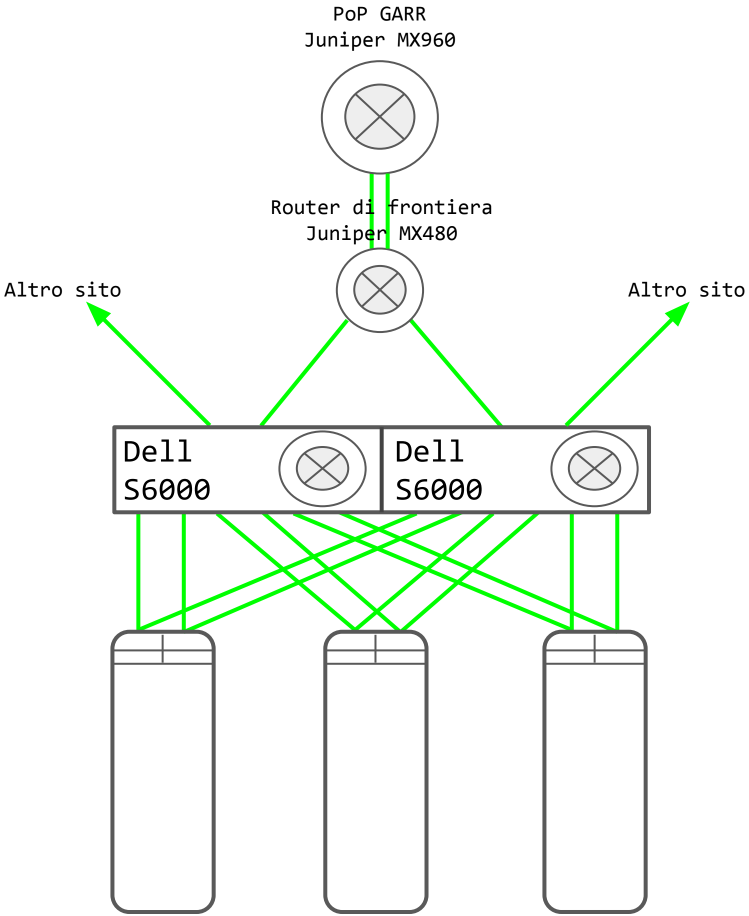


Network

40Gbps
10Gbps



Sito



Rete Dati 40 Gbps

- 2 Switch ToR Dell MXL in ciascun modulo-CSD
- 2 Router/Switch centro stella Dell S6000
  - 32 porte x 40 Gbps

Rete di gestione (“ILO”) separata da rete Dati

- 2 Switch management ToR Dell S55 in ciascun modulo-CSD
- 2 Switch management centro stella Dell S4810
  - 48 porte x 1 Gbps

Chassis Blade **Dell M1000e:**

- 16 server (lame) Dell Poweredge M620
- 2 switch integrati Ethernet (Dell MXL)
  - 2x16 porte 10 Gbps -> server
  - 4 uplink 40 Gbps -> centro stella;
- 2 switch Fibre Channel (Brocade M6505)
  - 16 porte a 16 Gbps verso i server
  - 8 uplink a 16 Gbps verso gli storage controller;

2 Storage Array **MD3860f FC:**

- Dischi SAS 116x4TB + 4xSSD 1.6TB
- FiberChannel brocade controller 2x16 Gbps (2x4 porte)

Modulo CSD

-Sistema operativo: **ubuntu** Xenial

-4 schede di rete - link aggregation **40Gbps**

-vlan + **Bridge** linux

-lama fisica fa da **GW per LXC**

-iptables su lama per:

- -fwd, nat + sicurezza LXC
- -indistinguibilità lame x LXC

LinuxContainer (LXD)

-opensource

-supportati da kernel moderni

-Uso efficiente HW (rispetto VM)

-Near Bare Metal runtime performance

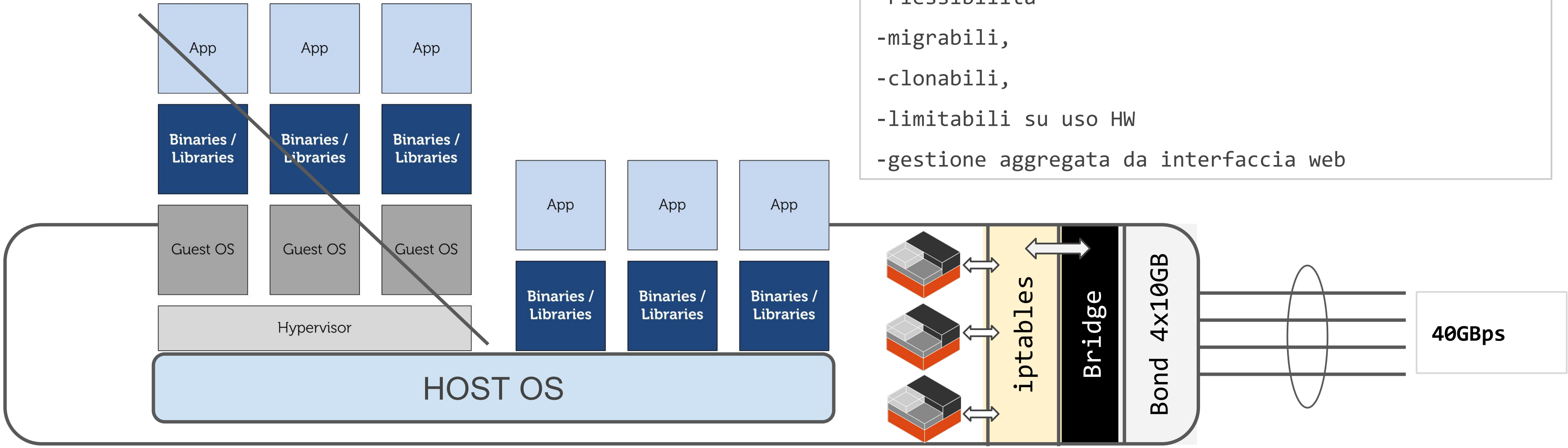
-Flessibilità

-migrabili,

-clonabili,

-limitabili su uso HW

-gestione aggregata da interfaccia web



Lama

Federated Cloud Architecture

multi-region (OpenStack) model

Region

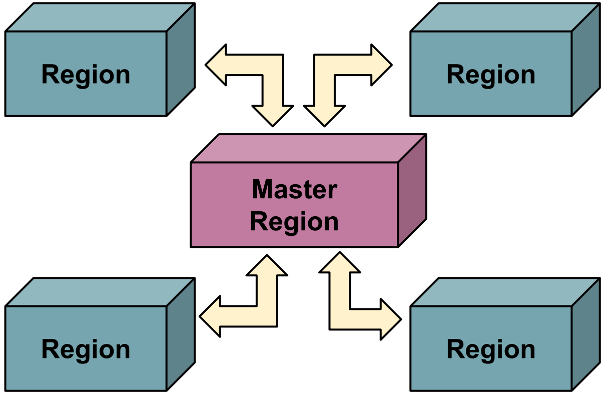
has its own deployment of OpenStack, is linked to other regions using Identity and dashboard.

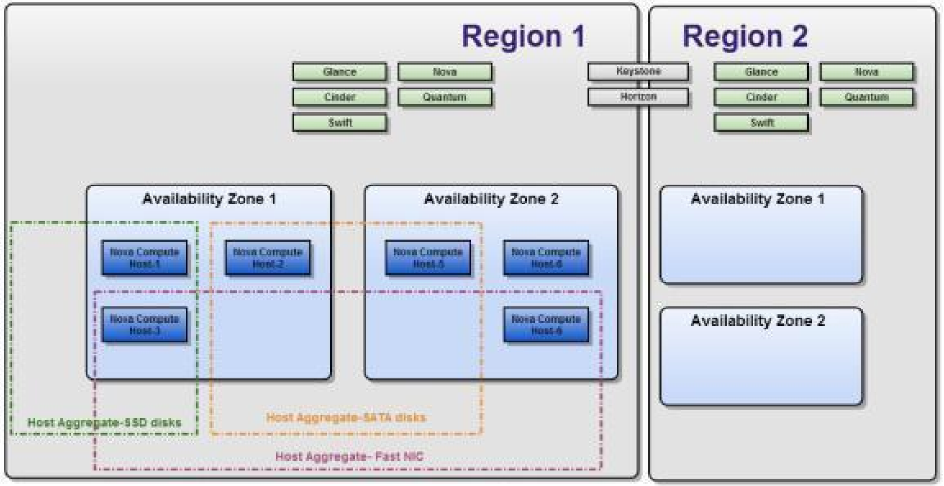
Availability Zone

Within each Region, nodes can be logically grouped into Availability Zones (AZ)

Host Aggregate

Within a Region machines can be grouped into Host aggregates. A machine may belong to multiple Host aggregates.



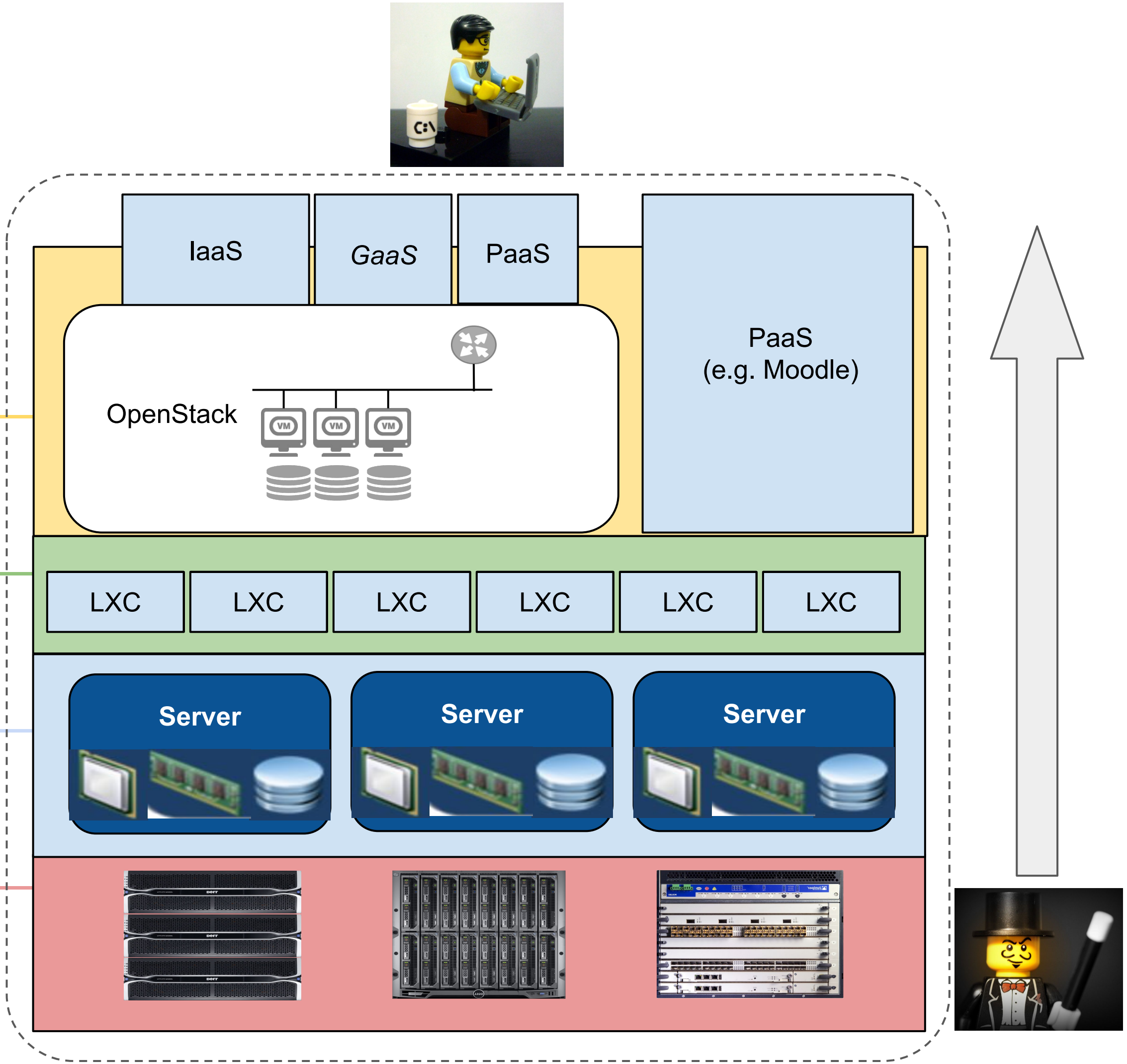


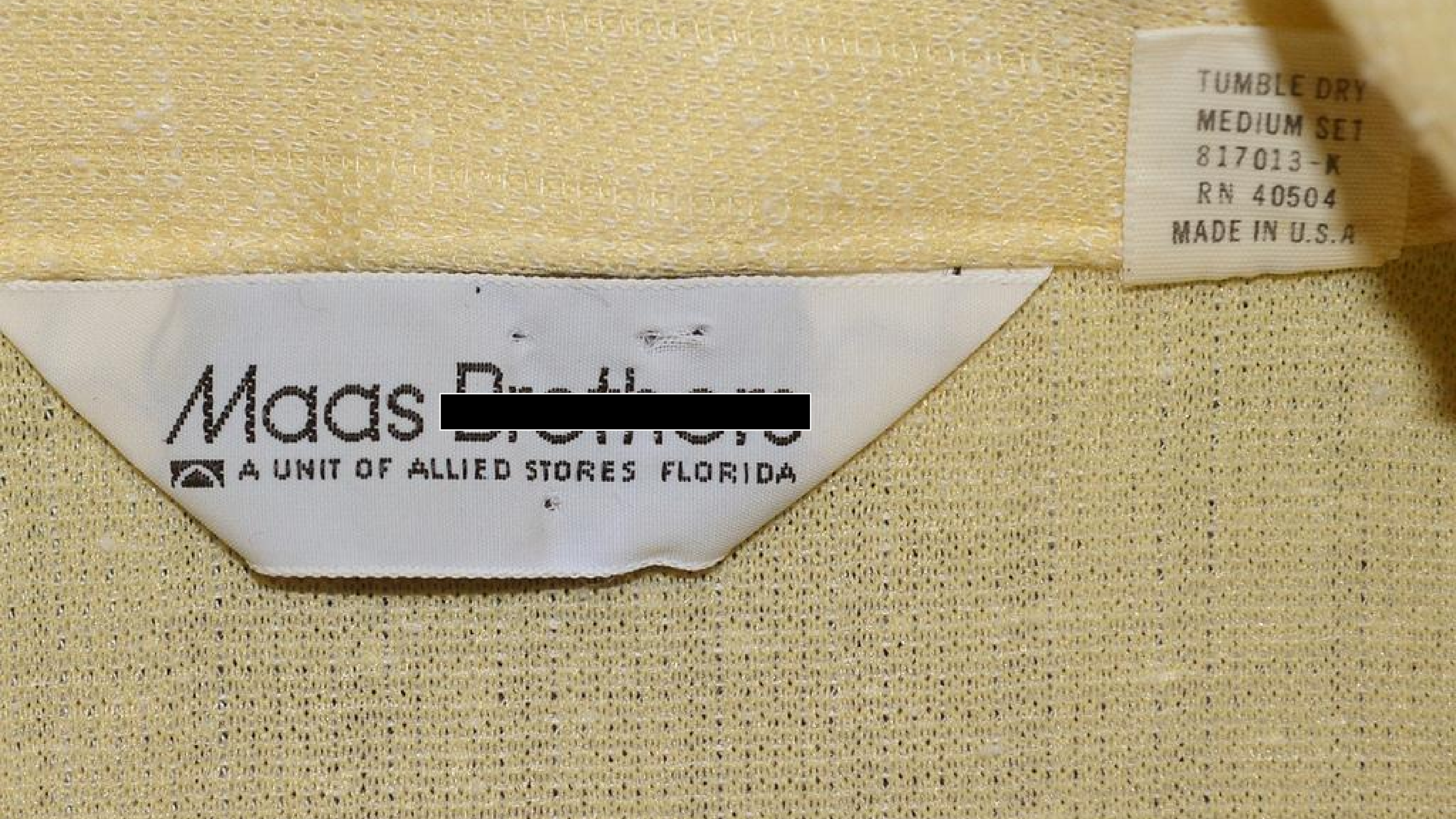
Building the GARR federated cloud...

...being an extremely busy team...

4 Layers

1. Application Services
2. Infrastructure *Virtualization*
3. Operating System
4. Physical resources





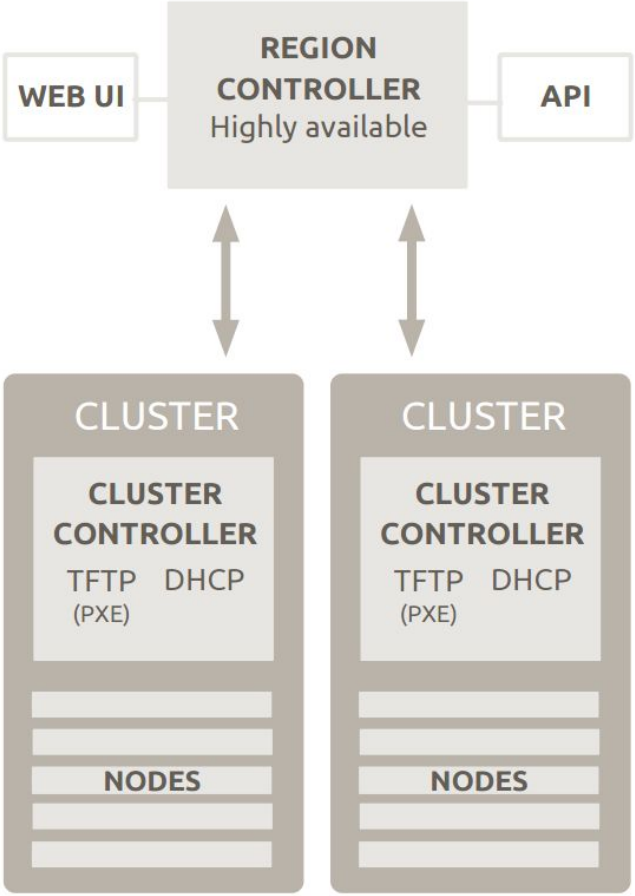
Maas [redacted]

A UNIT OF ALLIED STORES FLORIDA

TUMBLE DRY
MEDIUM SET
817013-K
RN 40504
MADE IN U.S.A

Metal As A Service

- Discover, commission and deploy **physical servers**
- **Allocate** physical resources to match workload requirements.
- **Retire servers when they are no longer needed** and make them available for new workloads as required.
- **Cross datacenters** provisioning



Images Courtesy of CANONICAL

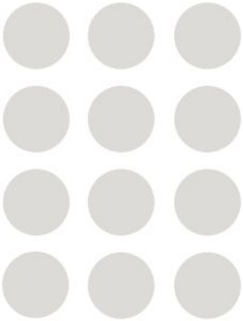
Rapid provisioning at cloud scale

3-step provisioning process

1

Install MAAS on first server

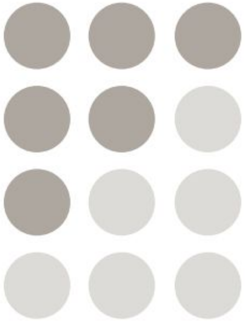
2



Discover Nodes

Automatically discover nodes
Enlist nodes via PXE boot
or manually enter MAC addresses

3

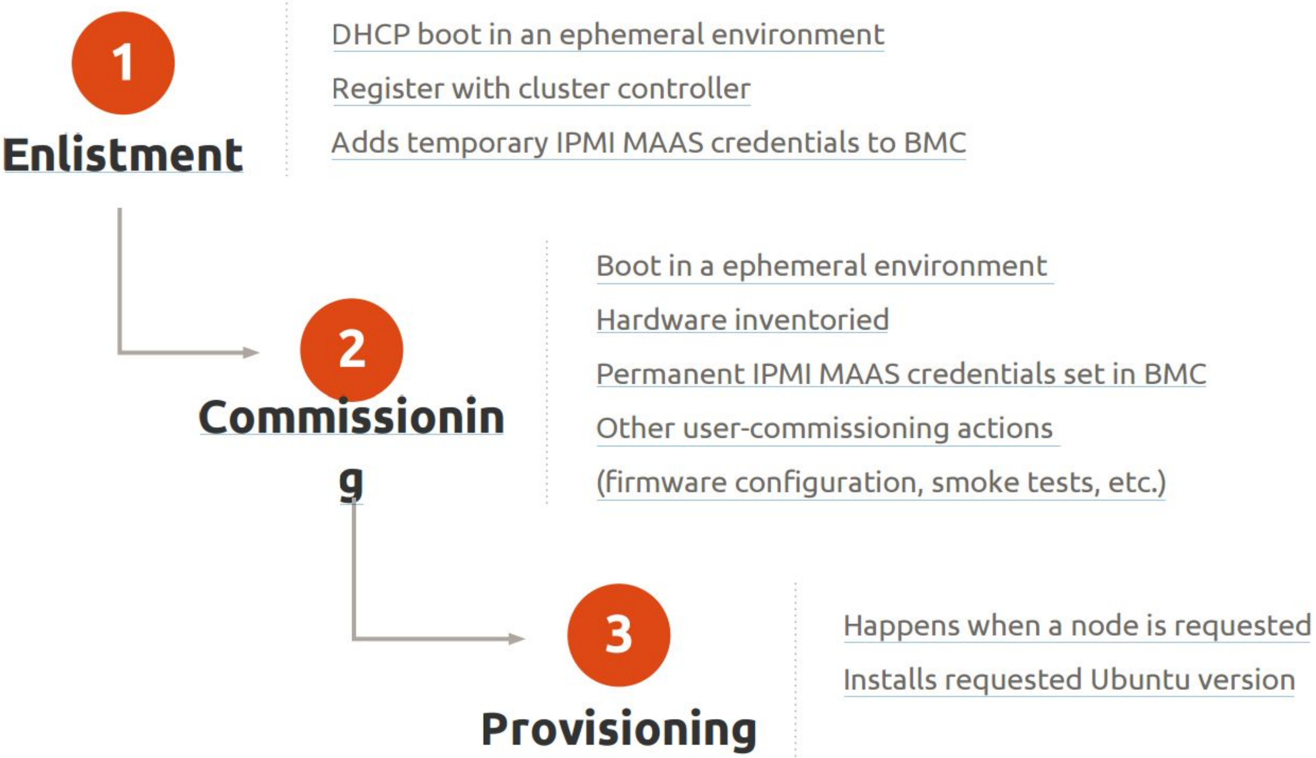


Power on Nodes

Hypervisor or OS
provisioned automatically

Hardware provisioning workflow

Courtesy of
CANONICAL



1

Enlistment

- DHCP boot in an ephemeral environment
- Register with cluster controller
- Adds temporary IPMI MAAS credentials to BMC

2

Commissionin
g

- Boot in a ephemeral environment
- Hardware inventoried
- Permanent IPMI MAAS credentials set in BMC
- Other user-commissioning actions
- (firmware configuration, smoke tests, etc.)

3

Provisioning

- Happens when a node is requested
- Installs requested Ubuntu version

Juju allows **configuring, managing, maintaining, deploying and scaling** cloud services (workloads) quickly and efficiently on multiple providers:

- private or public clouds
- bare metal, **leveraging MAAS to control the hardware.**

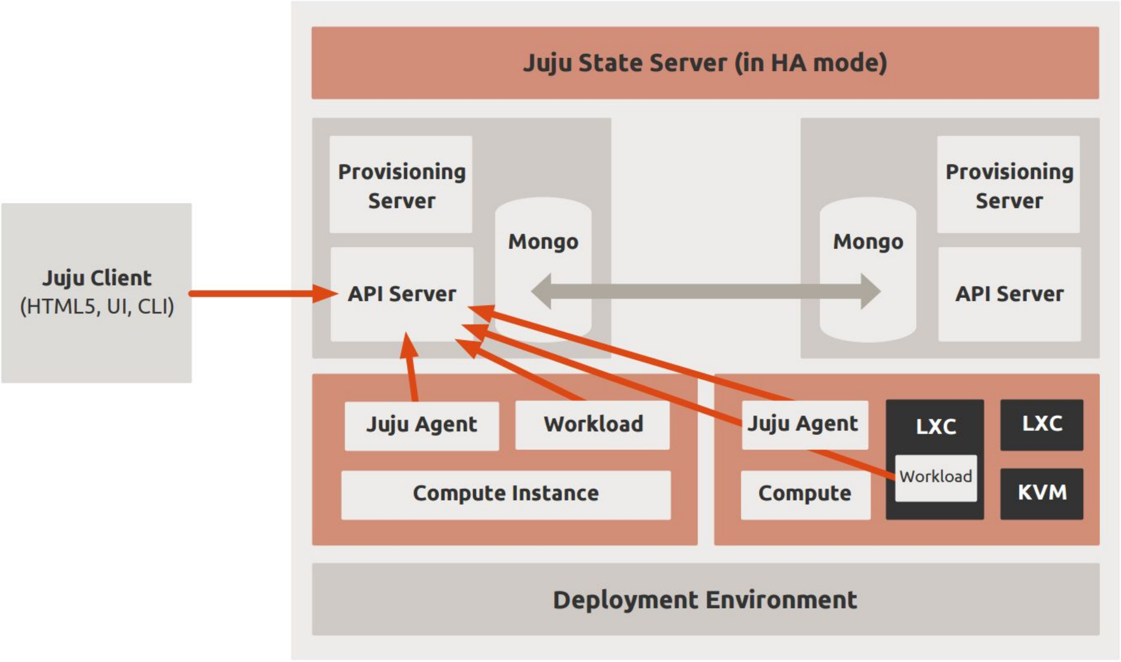
Juju uses descriptions of services called **Charms** which specify how to deploy a service.

Juju can manage and scale models consisting of many charms, creating complex architectures like OpenStack.

Juju can be controlled via a **web GUI, the command line, or API.**

architecture

- **Ease of provisioning:**
  from local machines to large clouds
- **Event-based**
  Reacts to changes in environment, self configuring
- **Scalable Templates**
  designed to scale by adding more units
- **Language independence** Hooks can be written in any language
- **In our env**: MAAS cloud to deploy O~S and O~S cloud to be available as a service



Courtesy of
CANONICAL

Anatomy of a Charm

Create charms and deploy your services

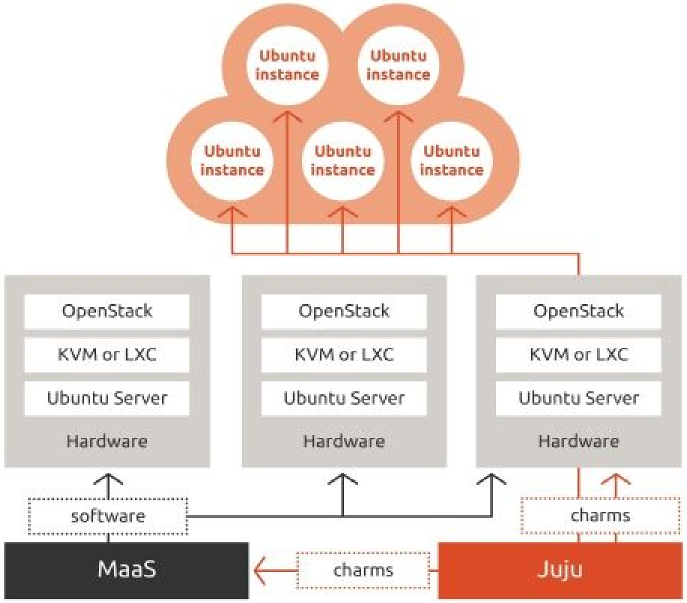
Charm Tools

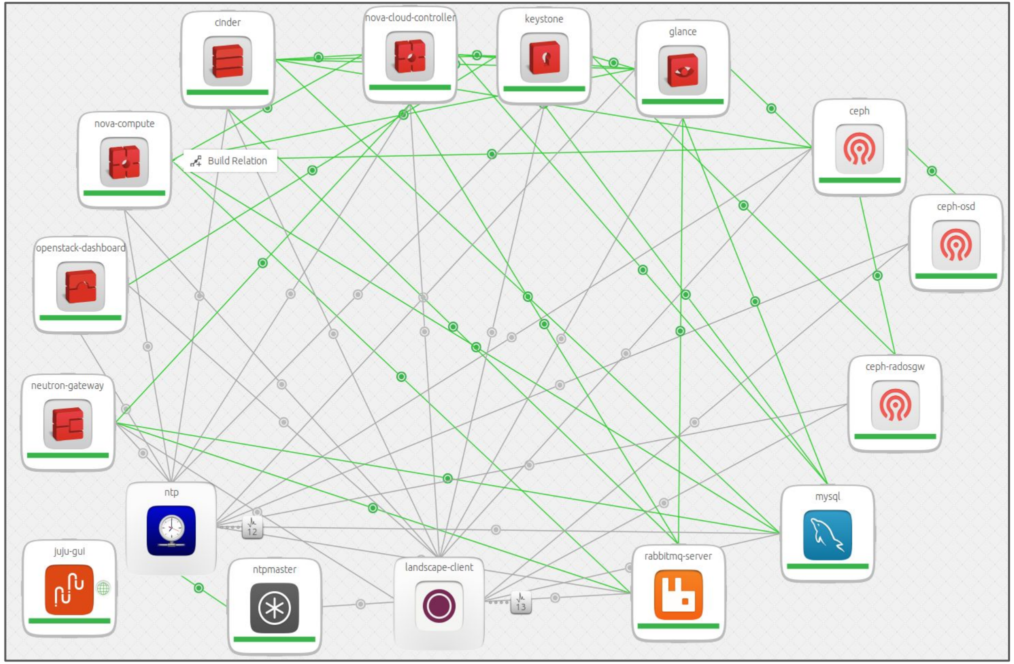
```
$ sudo add-apt-repository ppa:juju/stable
$ sudo apt-get update
$ sudo apt-get install charm-tools

$ juju charm create my-charm
```

Instant deployment

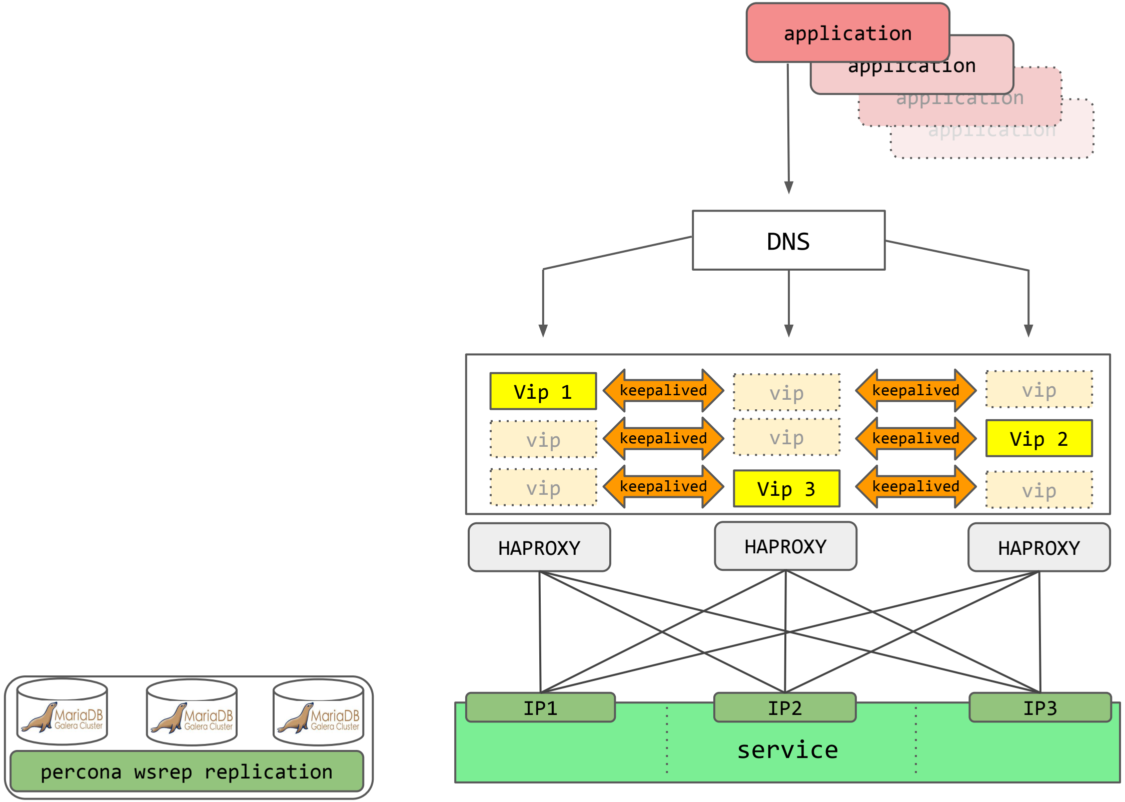
```
my-charm
 ├── config.yaml
 ├── hooks
 │    ├── config-changed
 │    ├── install
 │    ├── relation-name-relation-broken
 │    ├── relation-name-relation-changed
 │    ├── relation-name-relation-
departed
 │    ├── relation-name-relation-joined
 │    ├── start
 │    ├── stop
 │    └── upgrade-charm
 ├── icon.svg
 ├── metadata.yaml
 ├── README.ex
 └── revision
```

OpenStack as (one) orchestrated service





Service requests workflow



Criteri implementativi

No vendor lock in

- **Openstack** per la piattaforma virtuale
  - Release Mitaka
- **Ceph** (block) e **Swift** (object) via radosGW per la fornitura di storage

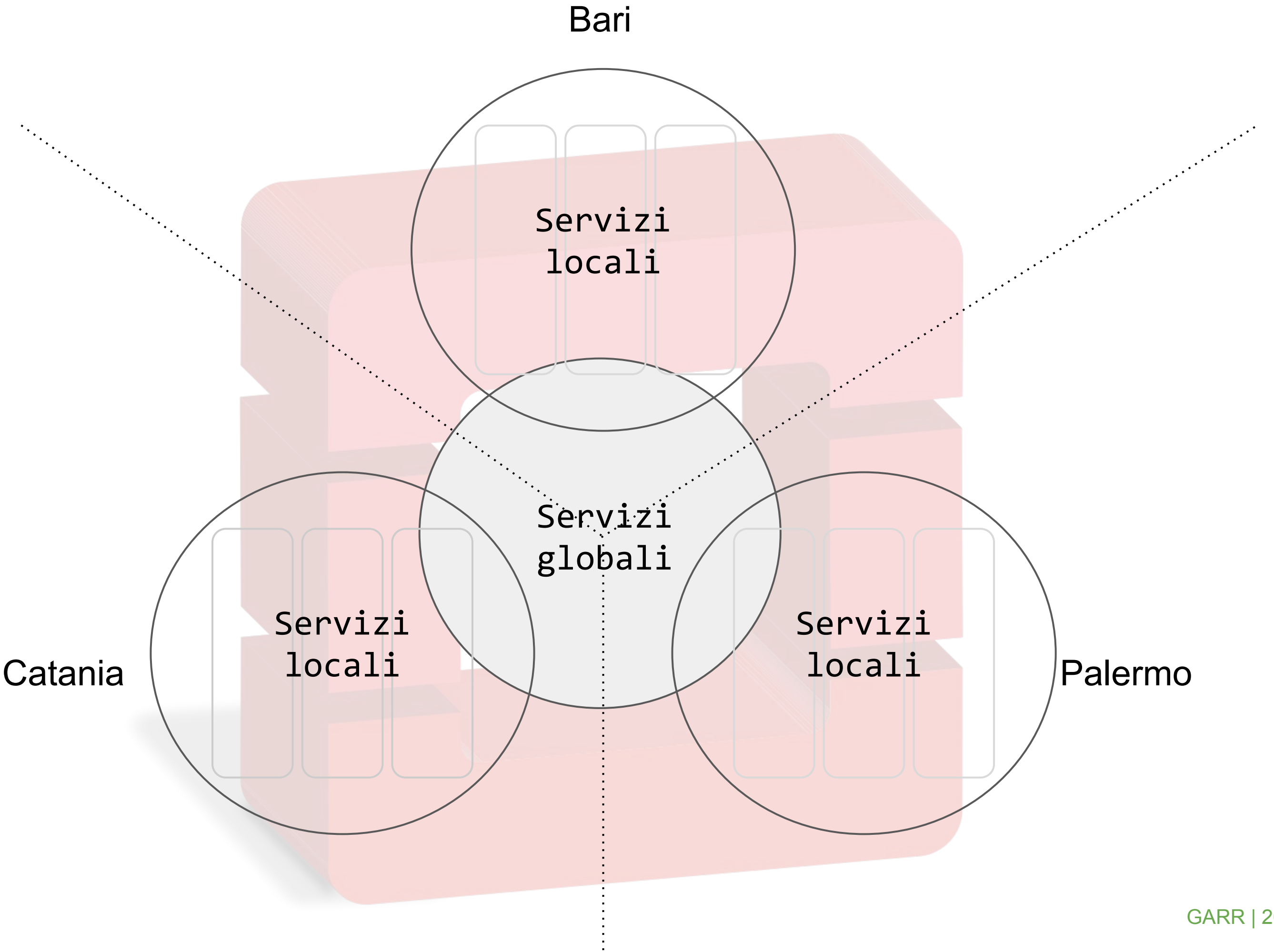
Suddivisione dei servizi di base:

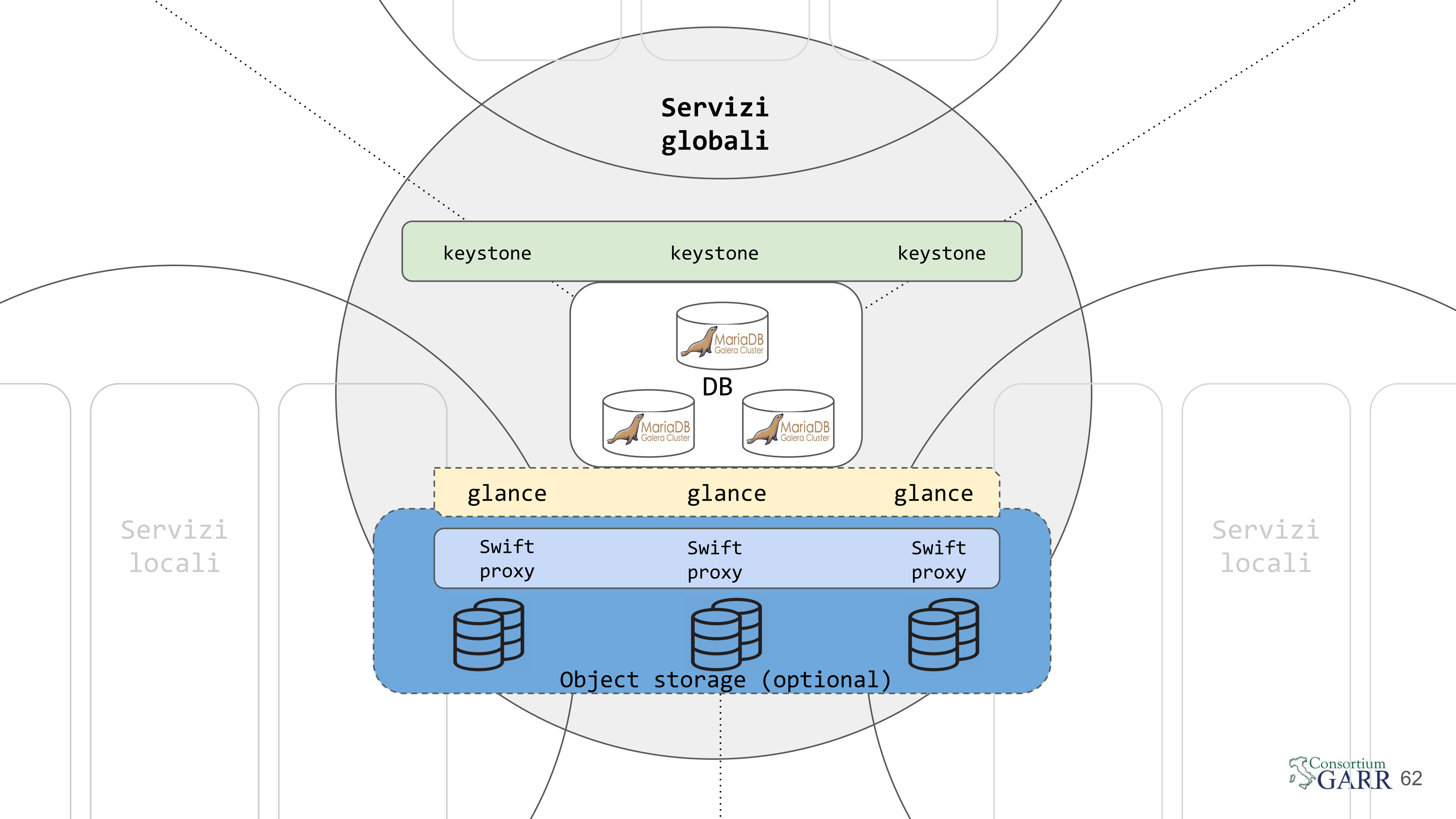
- **Globali** (unici sull'intero cluster - ridondati su **3 siti**)
  - Identity service / Keystone
  - Image service / Glance
  - Object Storage / rados gw
- **Locali** (individuali su ciascun sito - ridondati su **3 rack**)
  - Controller service / Nova
  - Network service / Neutron
  - Block Storage / Ceph

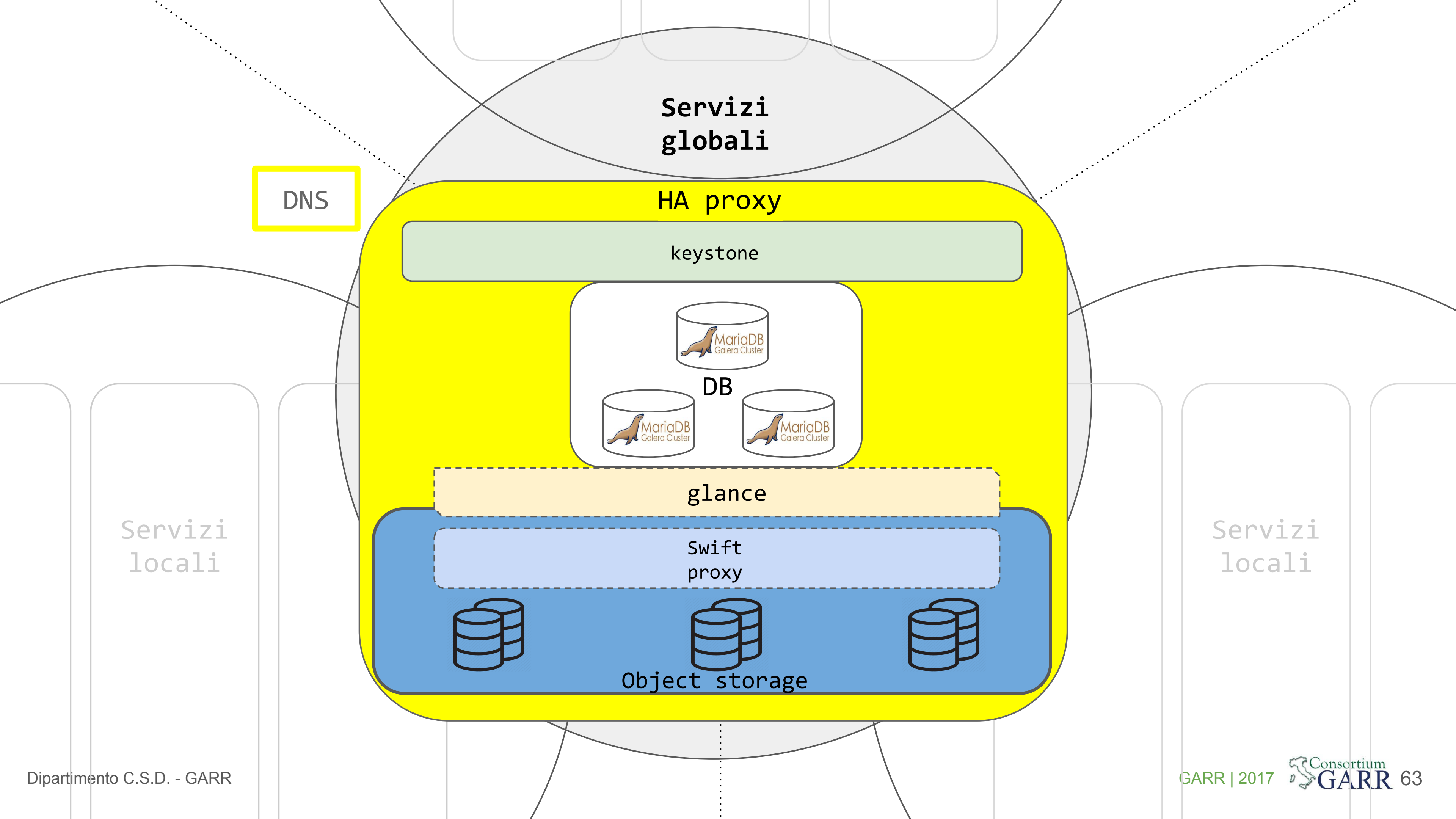
Ciascun sito individua una Openstack **Region**

OFFload trasparente vs **Amazon**

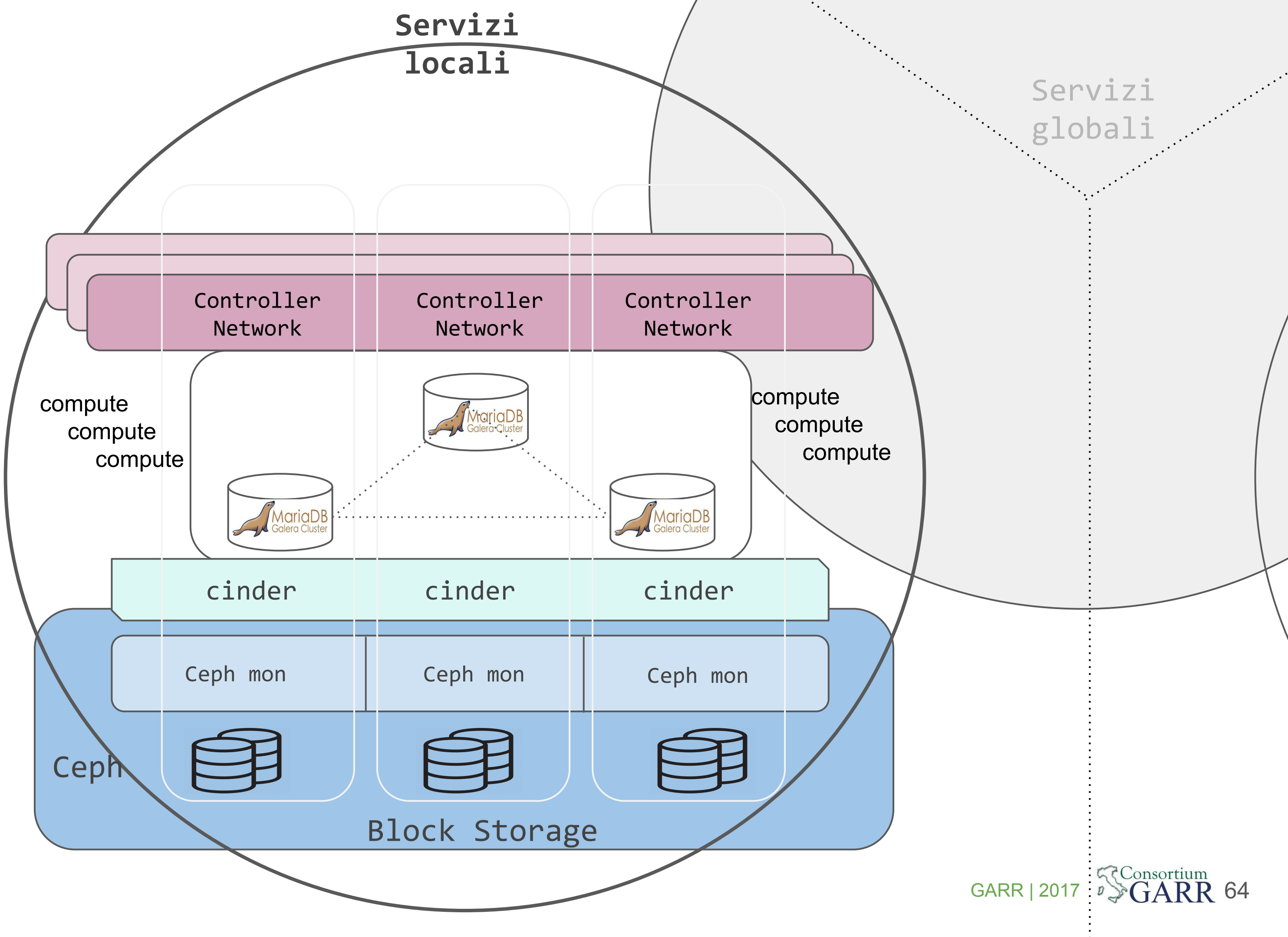
dns


Bari
Servizi locali
Servizi globali
Catania
Servizi locali
Servizi locali
Palermo

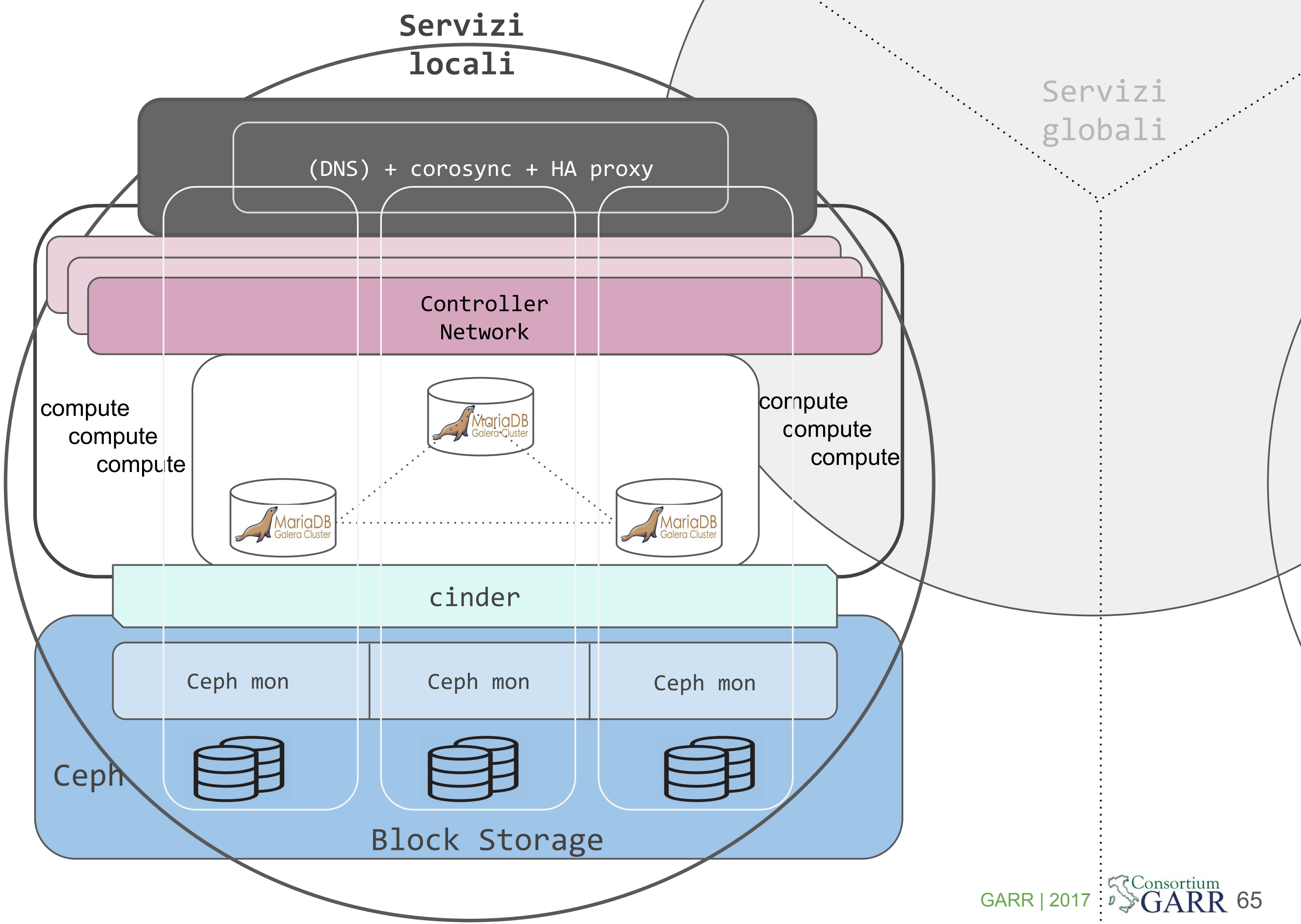

Servizi
globali
keystone
keystone
keystone
MariaDB Galera Cluster
DB
MariaDB Galera Cluster
MariaDB Galera Cluster
glance
glance
glance
Swift
proxy
Swift
proxy
Swift
proxy
Object storage (optional)
Servizi
locali
Servizi
locali


Servizi
globali
DNS
HA proxy
keystone
MariaDB Galera Cluster
DB
MariaDB Galera Cluster
MariaDB Galera Cluster
glance
Swift
proxy
Object storage
Servizi
locali
Servizi
locali

3x


Servizi locali
Servizi globali
Controller Network
Controller Network
Controller Network
compute
compute
compute
compute
compute
compute
MariaDB Galera Cluster
MariaDB Galera Cluster
MariaDB Galera Cluster
cinder
cinder
cinder
Ceph mon
Ceph mon
Ceph mon
Ceph
Block Storage

3x


Servizi locali
Servizi globali
(DNS) + corosync + HA proxy
Controller Network
compute
compute
compute
compute
compute
compute
MariaDB Galera Cluster
MariaDB Galera Cluster
MariaDB Galera Cluster
cinder
Ceph mon
Ceph mon
Ceph mon
Ceph
Block Storage

Users, projects, domain and roles

- **Users:** the basic entity who is allowed to log into Openstack
- **Roles:** define which actions users can perform
  - Default OpenStack roles: Cloud-admin, Admin, Member
  - Policies are written in json format files (policy.json)
- **Projects:** Organizational units in the cloud to which users are assigned to
  - Users can be members of one and more projects
  - Projects define resource quotas (CPUs, RAM, storage...)
  - Projects can have sub-projects associated
  - Nested quotas can be activated to limit the total resources assigned to a project tree
- **Domains:** higher level containers for projects and users
  - new with Keystone API v3!

Virtual Data Center on the GARR Cloud

Aim: delegate administration workload to vDC admins

- **Cloud admins** create “parent” project with agreed total resources (CPU, RAM, storage…)
- **vDC admins**
  - **create “child”** projects (limited by the quotas set on parent)
  - **assign users** to child projects
  - can **delegate administration** of parts of the project tree

Highlight

Modular and compact

- **Core** services Openstack on Linux Containers
  - Local components on **3 blades** each on a different **Rack**
  - Global components on **3 sites**

Throughput

- Networking: 4 link aggregation up to **40 Gbps**

Highlight

Resilienza/Load balancing

- servizi globali: ridondanza via **DNS**
  - 1 hostname globale risolto da più record-A
  - Resilienza servizi globali verificata contro:
    - Shutdown processo sul container
    - Shutdown container
    - Breakdown networking intero sito
- servizi locali: ridondanza e load balancing via **DNS + corosync + HAproxy**
  - tempi di risposta uguali anche in caso di perdita di un membro del cluster
- **percona multi-master** per i database
- HA Keystone via **Fernet tokens driver**
- **Rabbit cluster** (3 membri) locale

Highlight

Networking

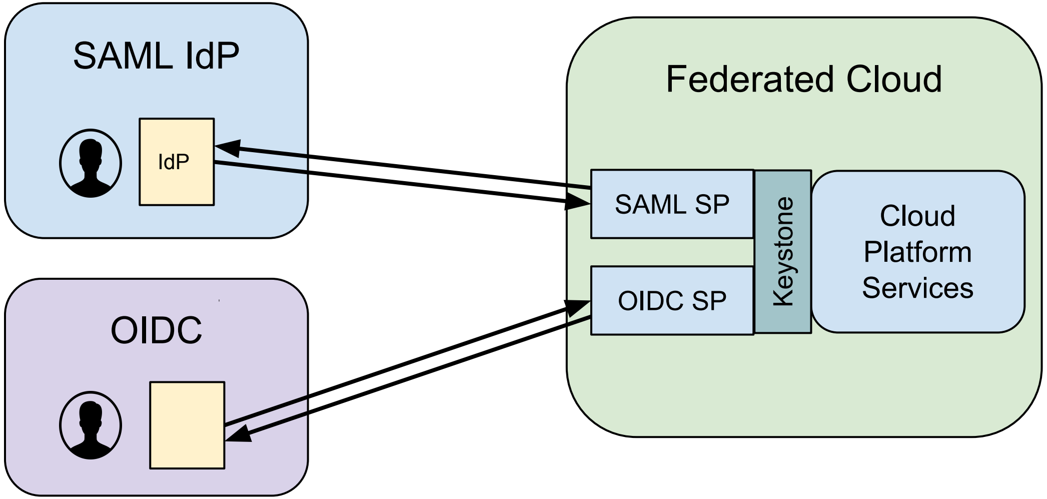
- Separazione L2 (VLAN) delle reti server e delle reti di Openstack
- Separazione reti tenant via GRE
- Networking inter-sito servizi openstack e tenant via IP su link dedicato

Sicurezza

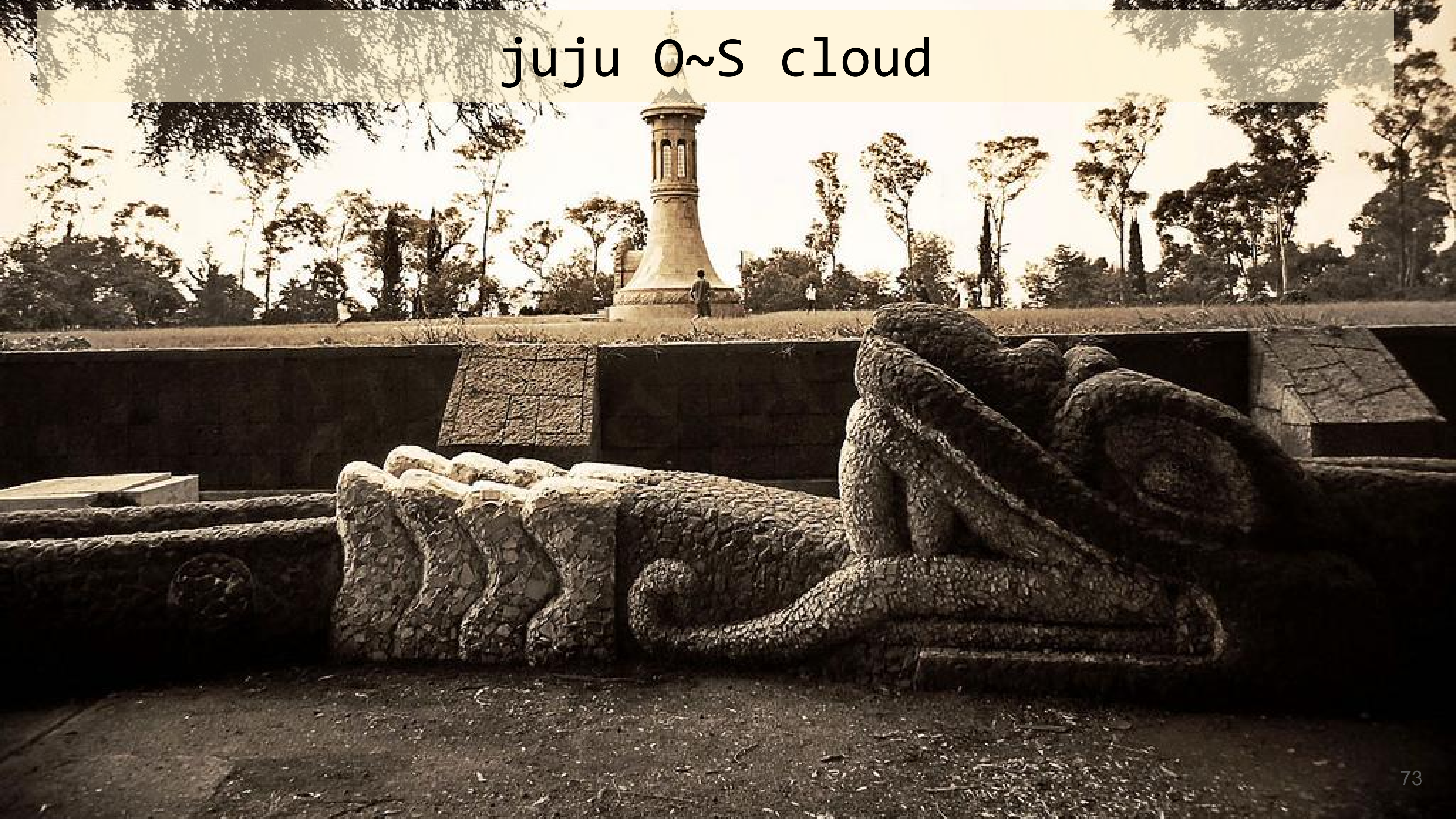
- Servizi Openstack su LXC: iptables sui server fisici ospiti
- VM Openstack sui compute: Neutron Security groups (iptables)
- ACL sul router di frontiera

Federated authentication/authorization

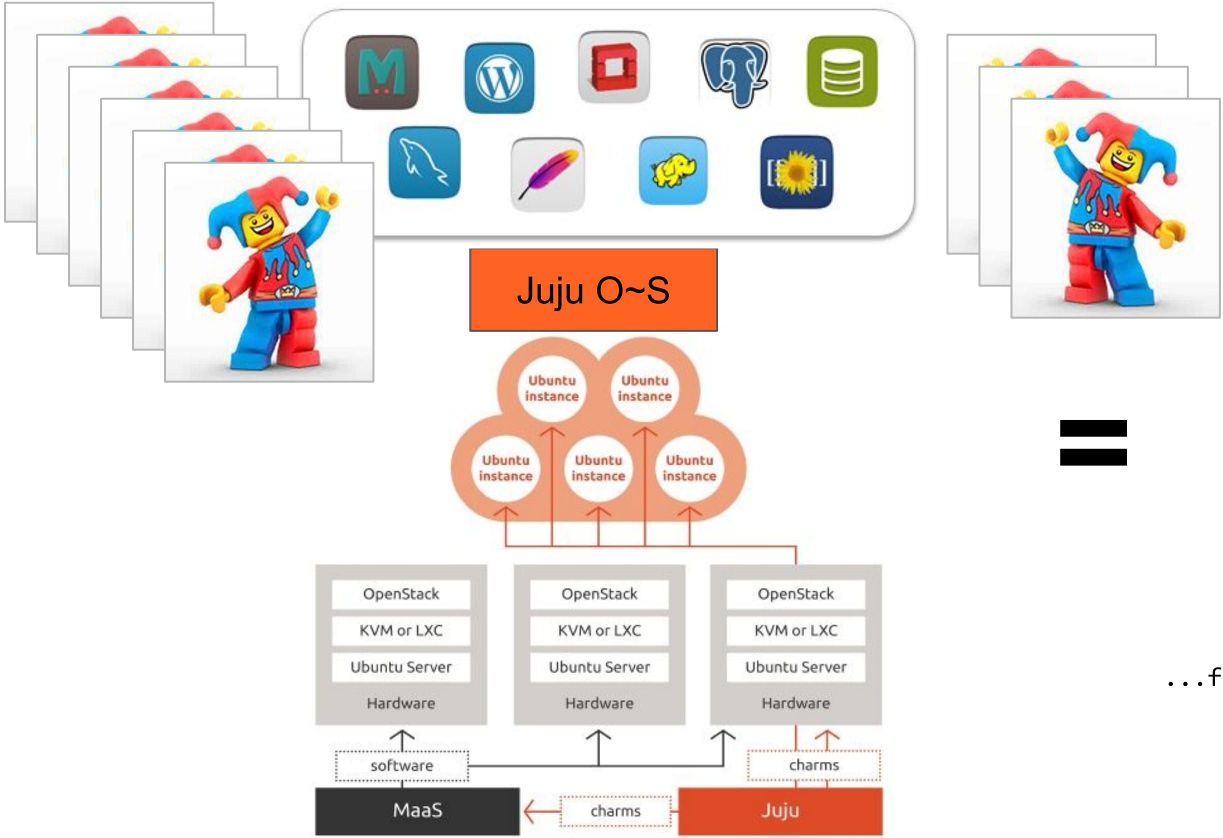
1. **Separation of roles**: cloud administrator and the domain administrators.
2. The federated Identity providers are **delegated** only for **authentication**
3. **No authorization stored outside of keystone**, in order to avoid:
    a. Having to check reliability and consistency of such information
    b. Having to map it to internal keystone entities
    c. Force users to act on an IdP not under their personal control
4. **Users can be granted rights on any project** of the federation, irrespective of their affiliation and under the sole control of the administrator for that project
5. Deploy the simplest solution, relying **as much as possible on native OpenStack** capabilities avoiding any extra non necessary component.



juju O~S cloud



More powerful than a PAAS, easier than a IAAS



Juju O~S

Ubuntu instance
Ubuntu instance
Ubuntu instance
Ubuntu instance
Ubuntu instance

OpenStack	OpenStack	OpenStack
KVM or LXC	KVM or LXC	KVM or LXC
Ubuntu Server	Ubuntu Server	Ubuntu Server
Hardware	Hardware	Hardware

software
charms
MaaS
charms
Juju

=

GaaS :)

...for friends, don’t tell the CSD *boss*

status

What is available:

- Complete automatic deployment of openstack from bare metal to full region up and running in a few hours (you’ll see it in a moment)
- 2 regions up and running (we’ll setup a 3rd in moment)
- 4 deployments openstack mitaka for a total of about 20.000 vcpu (and counting till 100k - 120k)
- Virtual Data Centers available to users in few minutes on demand
- PaaS services (i.e. Moodle, Hadoop, Spark...)
- *GaaS* a GARR version of advanced PaaS or simplified IaaS (via juju with O~S cloud backend)
- Federated access (SAML-idem and OIDC-google login available)
- Multiple region Federation *recipe* (git and knowledge base available)

Gestione failover: scenari

Indisponibilità container / servizio su container

- Resilienza garantita da DNS + Keepalived + HAproxy
- Juju will take care of keeping the state consistent scaling the services horizontally in case needed

Perdita di un modulo-CSD

- Servizi core openstack in HA cross modulo-CSD
- VM istanziate su volumi Ceph *evacuabili* (*nova evacuate*) e riattivabili su altri moduli senza perdita dati
    - Possibilità di gestione automatica failure (nrpe nagios)

Perdita di un sito

- Servizi globali openstack resilienti (sperimentato)
- VM ephemeral disk:
    - Snapshot periodica su Glance / Swift -> immagini disponibili sull’intero cluster
    - Respawn da snapshot su altro sito (richiede replica reti)
- VM Ceph: in progress

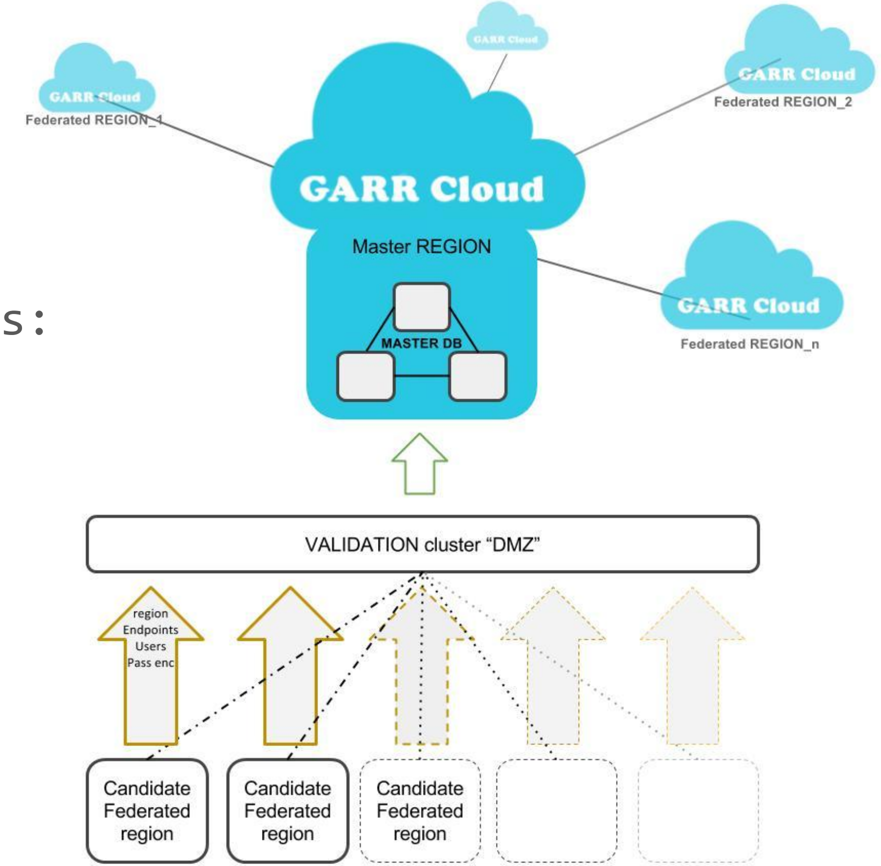
Join the Federation

Procedure of inclusion

- Bundle O~S attaches to validation cluster
- Validation in “DMZ” cluster
- No cleartext credentials exchange

Different contribution options:

1. You own HW, but have no manpower/knowledge
2. You already have an O~S deployment
3. None of the previous, but you have (wo)men



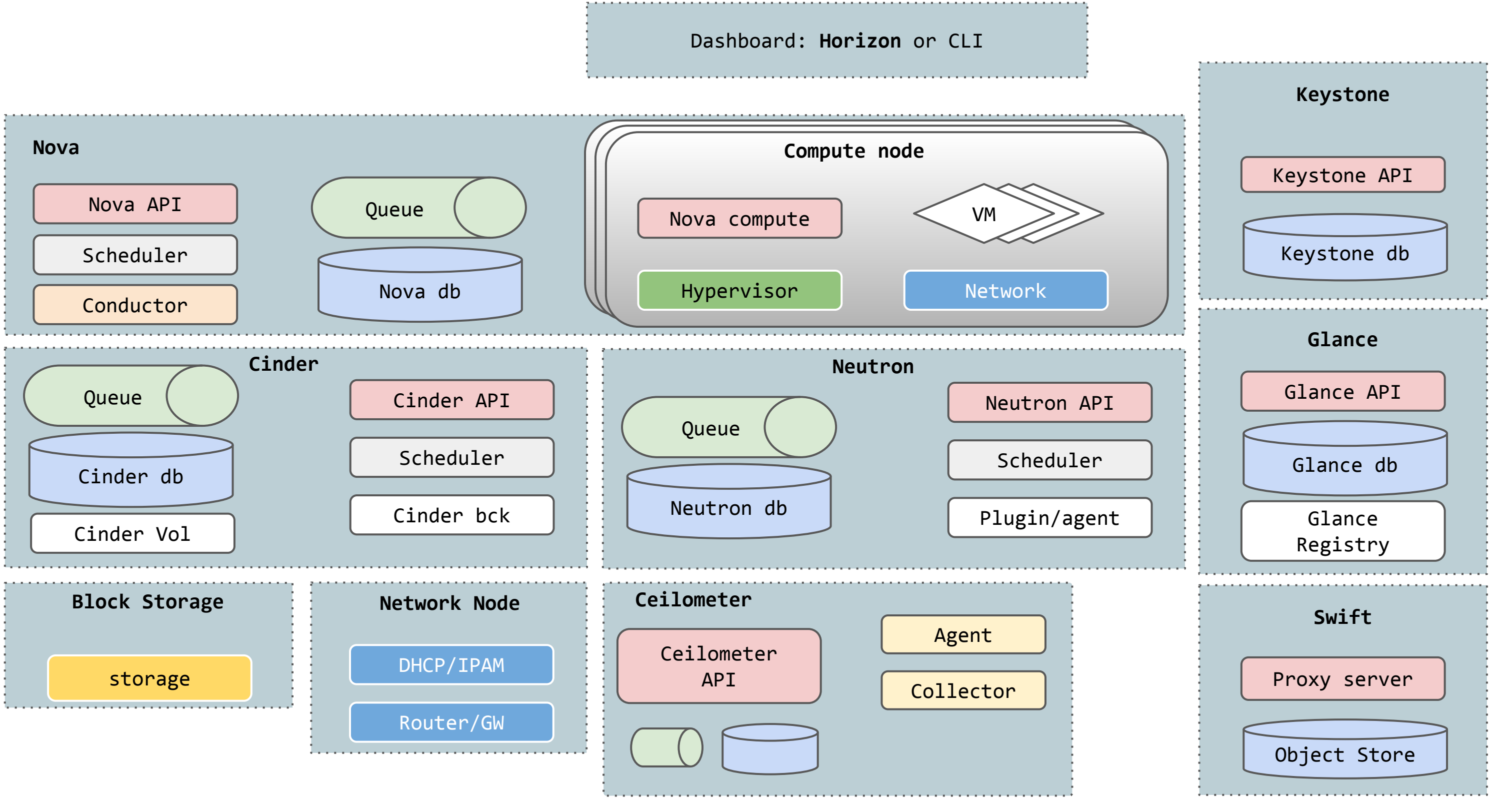
https://cloud.garr.it/forms/register/

Useful and Suggested readings

- About monopoly debate in cloud http://shirky.com/writings/powerlaw_weblog.html
- “The Big switch” N.Carr (yes it’s a paper book!)
- OpenStack Cloud Administrator Guide http://docs.openstack.org/admin-guide-cloud/content/index.html
- OpenStack keystone developer documentation http://docs.openstack.org/developer/keystone/
- OpenStack Identity Administration documentation http://docs.openstack.org/trunk/openstack-compute/install/content/ch_installingopenstack-identity-service.html
- Deploying openstack - Ken Pepple (O’Really)
- About GARR cloud http://cloud.garr.it
- OPENSTACK NETWORKING GUIDE (ask google for the latest)
- MAAS (CANONICAL) on www
- Juju (CANONICAL) on www
- Blockchain and O~S (for future developments) https://www.openstack.org/videos/barcelona-2016/blockchain-and-openstack-building-trusted-chains

hands on

bck



Dashboard: **Horizon** or CLI

Nova

Nova API

Scheduler

Conductor

Queue

Nova db

Compute node

Nova compute

VM

Hypervisor

Network

Keystone

Keystone API

Keystone db

Cinder

Queue

Cinder db

Cinder Vol

Cinder API

Scheduler

Cinder bck

Neutron

Queue

Neutron db

Neutron API

Scheduler

Plugin/agent

Glance

Glance API

Glance db

Glance Registry

Block Storage

storage

Network Node

DHCP/IPAM

Router/GW

Ceilometer

Ceilometer API

Agent

Collector

Swift

Proxy server

Object Store