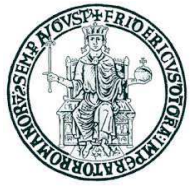
Cloud e Datacenter Networking

Università degli Studi di Napoli Federico II
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione DIETI
Laurea Magistrale in Ingegneria Informatica

Prof. Roberto Canonico

OpenStack: an introduction

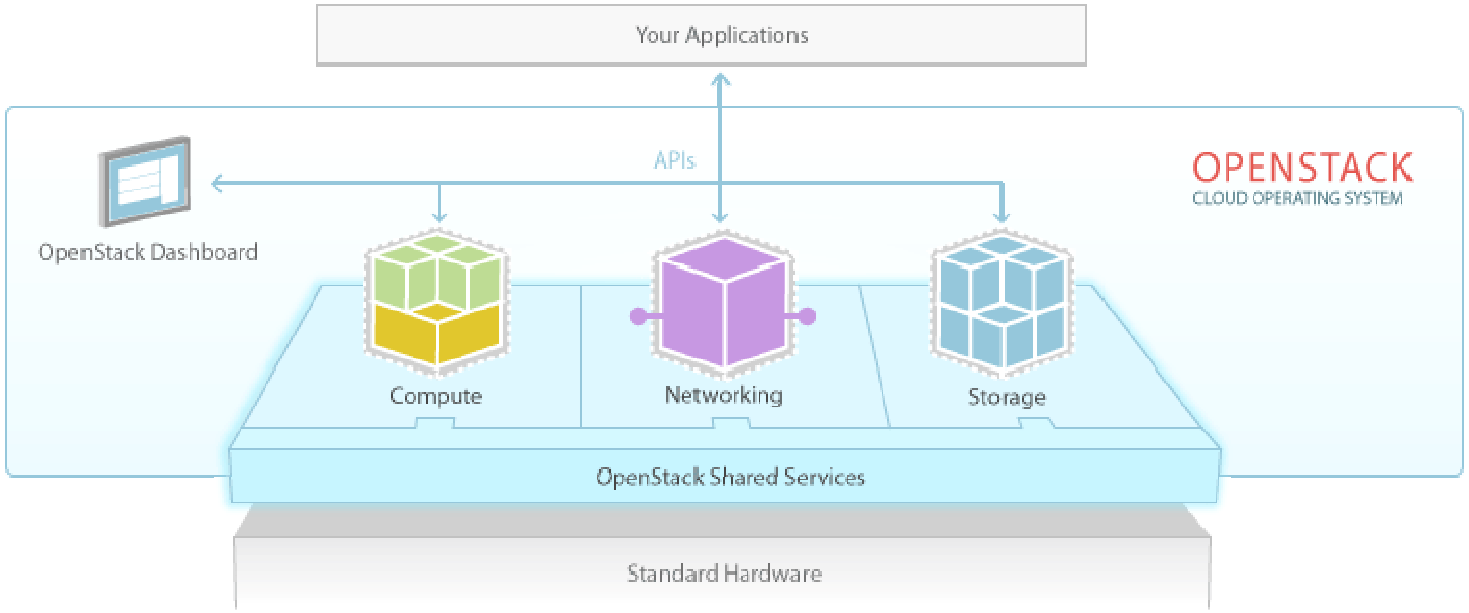


Lesson outline

- OpenStack Architecture
- Presentation of core OpenStack services

OpenStack

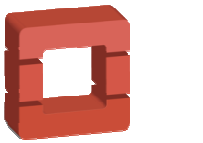
- OpenStack is a cloud management system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface
- Apache 2.0 license (OSI), open development process
- Publically available open source code repository
- Modular design for deployment flexibility via APIs



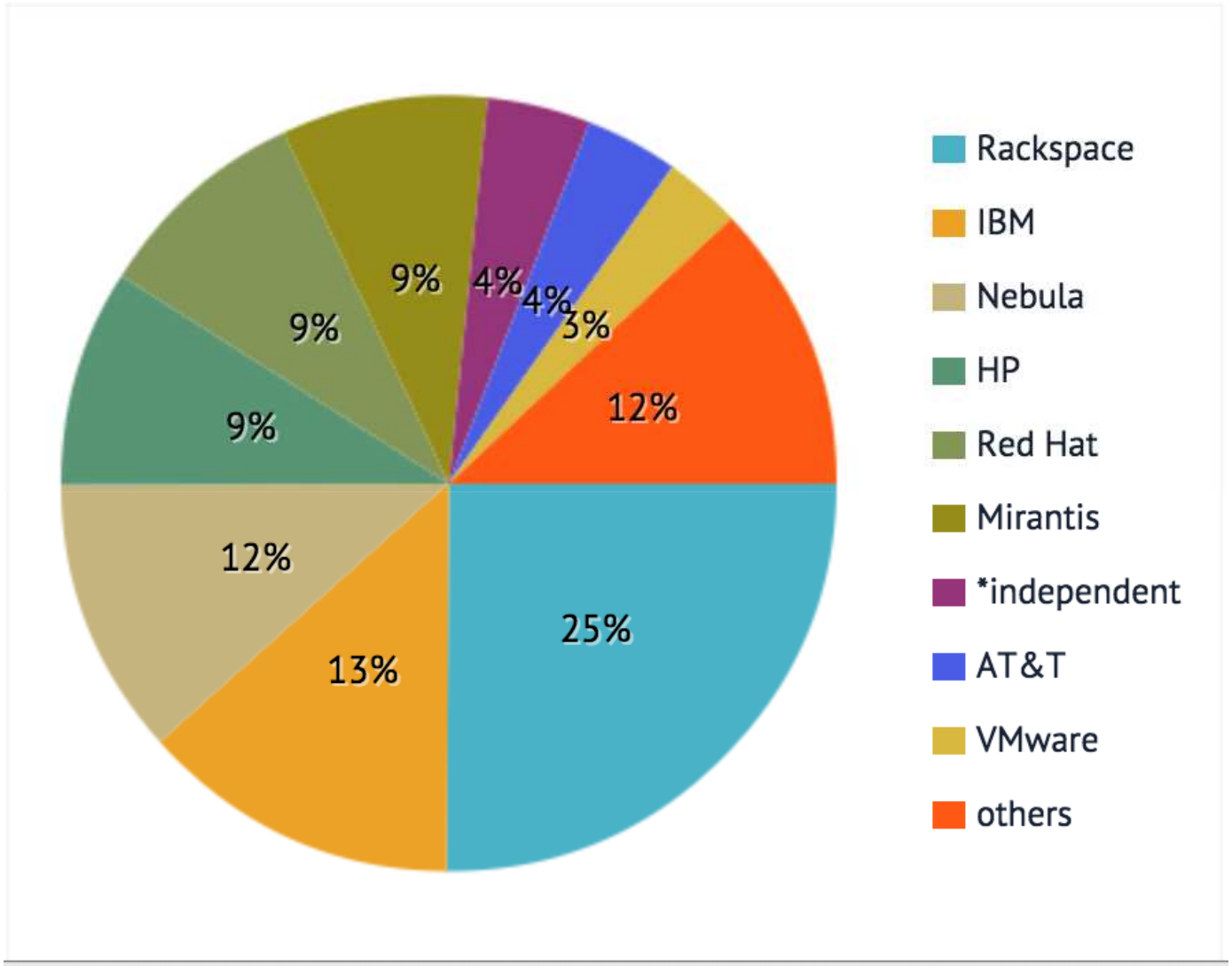
openstack
CLOUD SOFTWARE

OpenStack: A Brief History

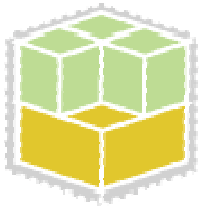
- September 2009: NASA Launches Nebula
  - One of the first cloud computing platforms built for Federal Government Private Cloud
- March 2010: Rackspace Open Sources Cloud Files software, aka Swift
- May 2010: NASA open sources compute software, aka “Nova”
- June 2010: OpenStack is formed
- July 2010: The inaugural Design Summit
- April 2012: OpenStack Foundation
- April 2013: Grizzly Release (7th)
- October 2013: Havana Release (8th)
  - Quantum service renamed to Neutron
- April 2014: Icehouse Release (9th)
- October 2014: Juno Release (10th)
- April 2015: Kilo Release (11th)
- October 2015: Liberty Release (12th)
- April 2016: Mitaka Release (13th)



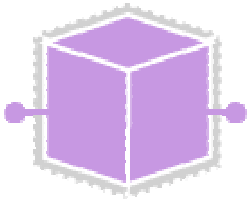
OpenStack top contributors



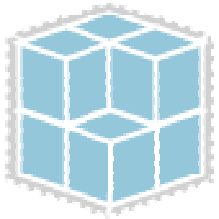
OpenStack Core Services



- Compute (“Nova”) provides virtual servers upon demand
  - Compute resources are accessible via APIs for developers building cloud applications and via web interfaces for administrators and users
  - The compute architecture is designed to scale horizontally on standard hardware



- Network (“Neutron” formerly known as “Quantum”) is a pluggable, scalable and API-driven system for managing networks and IP addresses
  - Replaced at some point the old Nova-Network service



- Block Storage (“Cinder”) provides persistent block storage to guest VMs
  - This project was born from code originally in Nova

- Dashboard (“Horizon”) provides a modular web-based user interface for all the OpenStack services

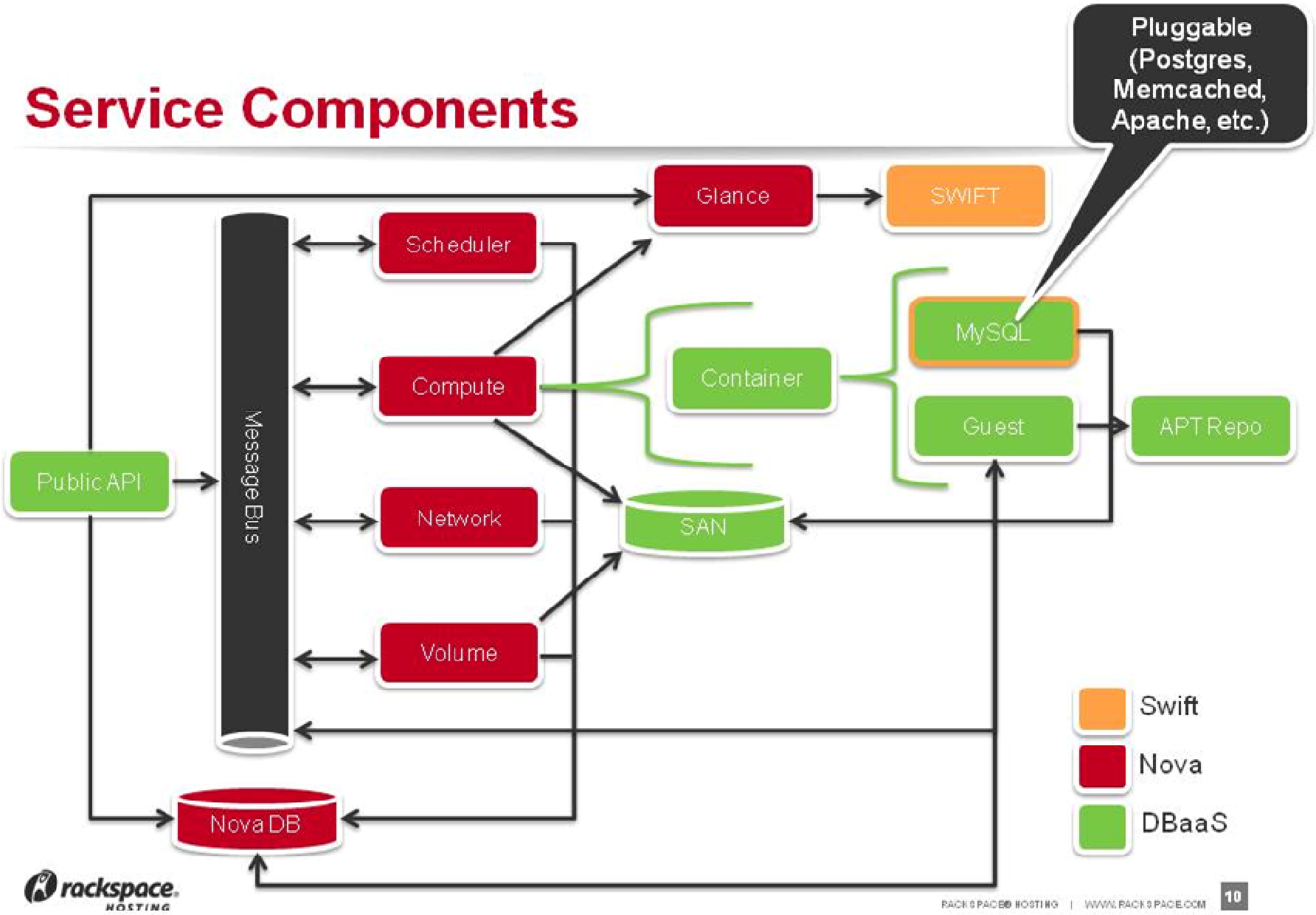
OpenStack Core Services

- Object Store (“Swift”) provides object storage
  - It allows you to store or retrieve files (but not mount directories)
- Image (“Glance”) provides a catalog and repository for virtual disk images
  - These disk images are most commonly used in OpenStack Compute

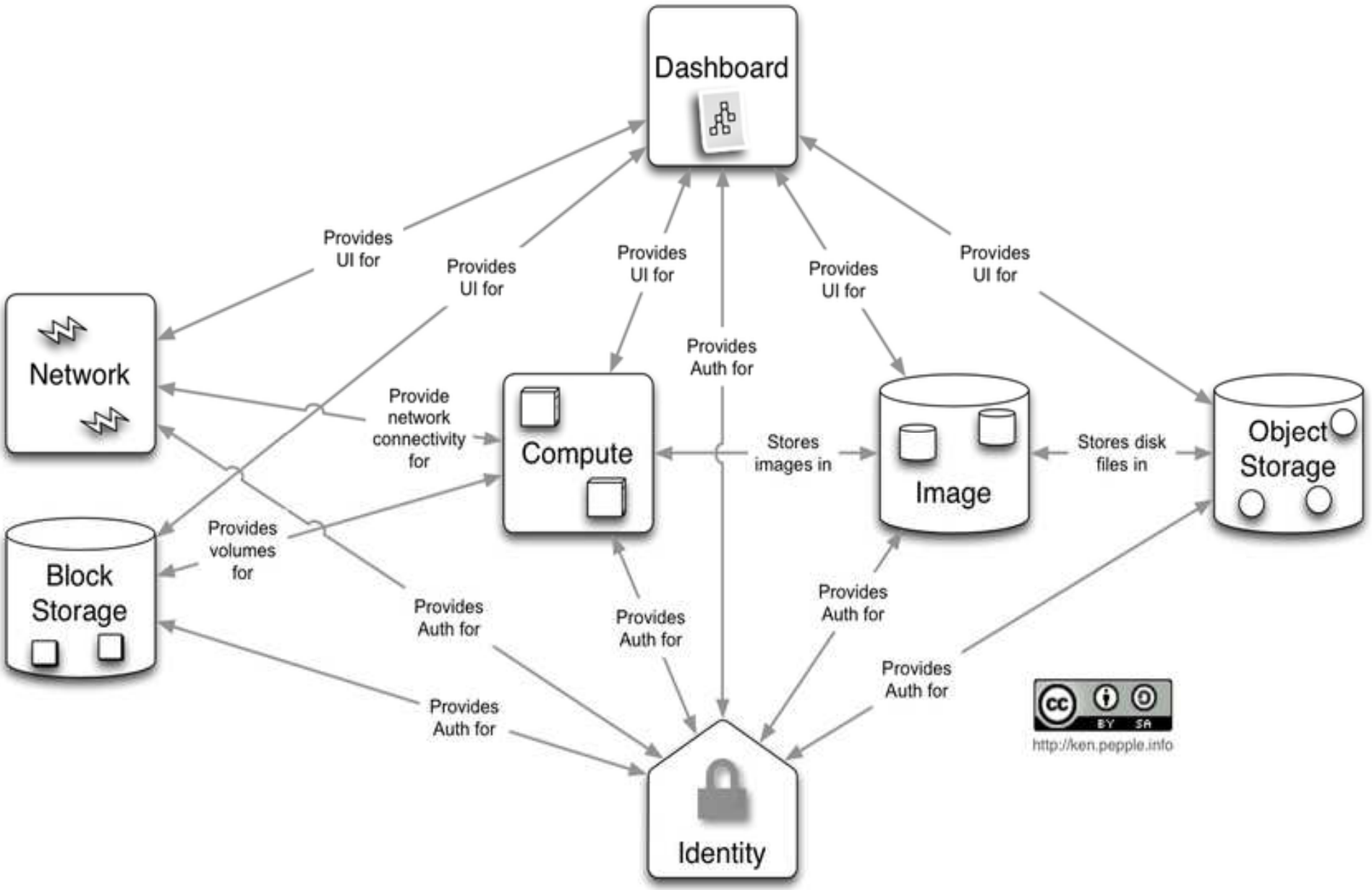
- Identity (“Keystone”) provides authentication and authorization for all the OpenStack services

- Orchestration (“Heat”) orchestrates multiple cloud applications using the AWS CloudFormation template format, through both an OpenStack-native REST API and a CloudFormation-compatible Query API
- Metering (“Ceilometer”) monitoring and metering framework

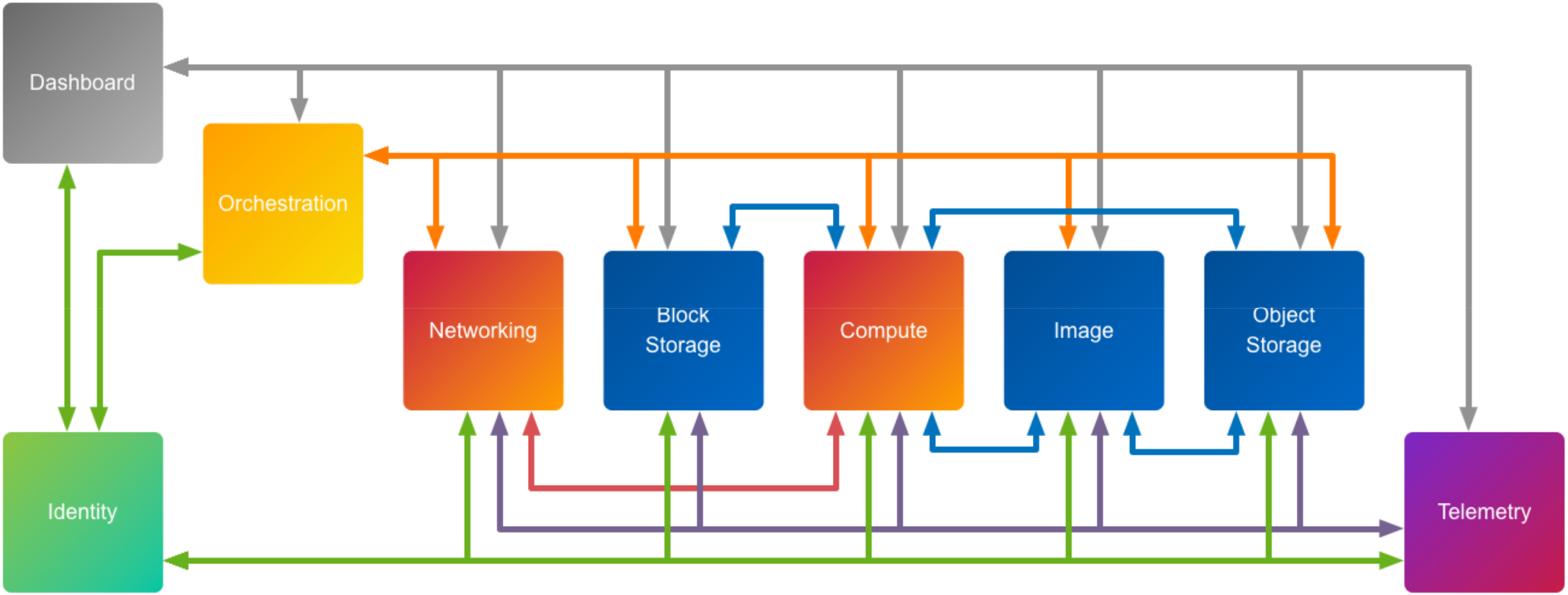
OpenStack architecture



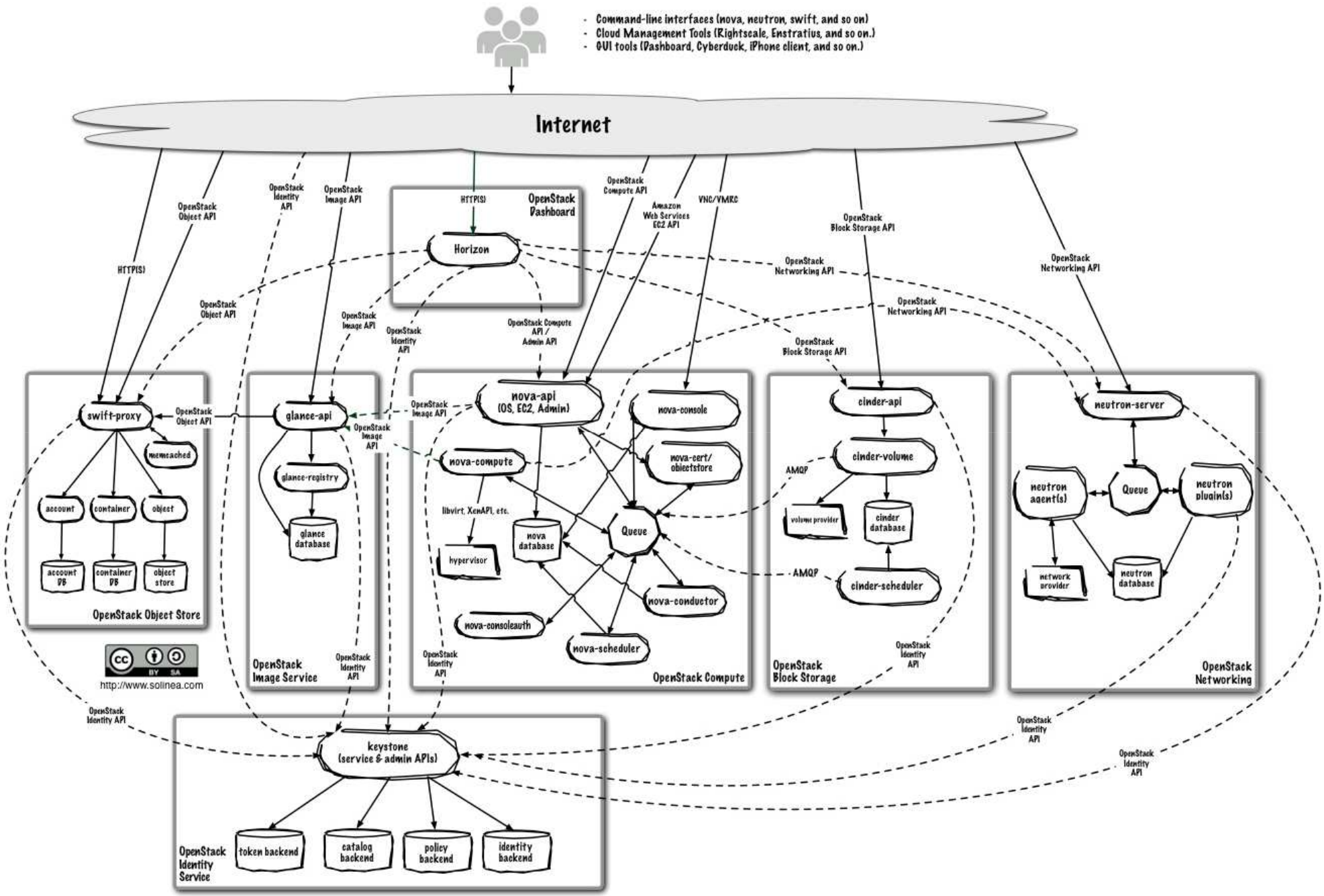
OpenStack Core Services: relationships



OpenStack Core Services: interactions (1)



OpenStack Core Services: interactions (2)



Common approach to OpenStack services design

- Each OpenStack core service exposes all its capabilities over a RESTful API
- Services interoperate through RESTful API calls, so when a service requires resources from another services, it makes a RESTful API call to query services' capabilities, list its resources or call for a certain action
- Each Openstack service consists of several **components**
- Components use a message broker server for inner service communication
  - RabbitMQ in most cases
- Components save persistent data and objects' states into a database

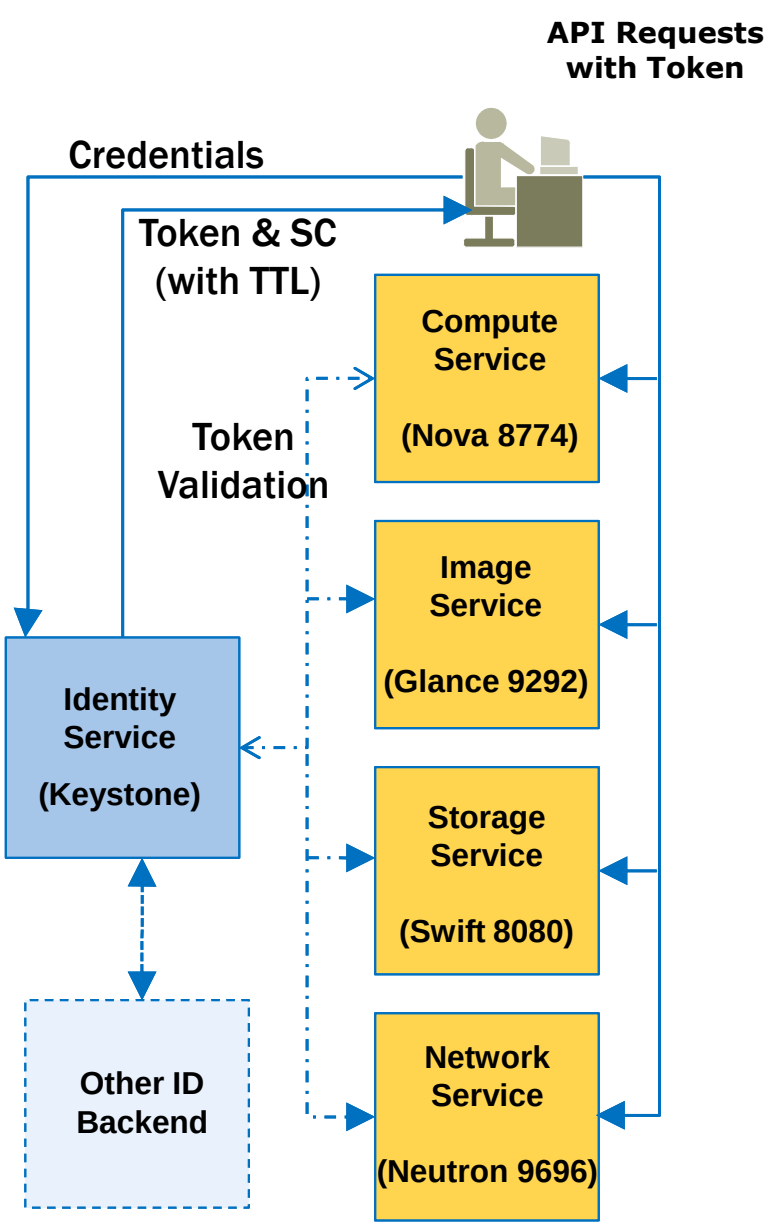
Message bus

- Communication among OpenStack components happens through a message bus
- Message routing between services
- Generic API to send messages
- Multiple drivers supported
  - RabbitMQ
  - ZeroMQ
  - Qpid

Nova database

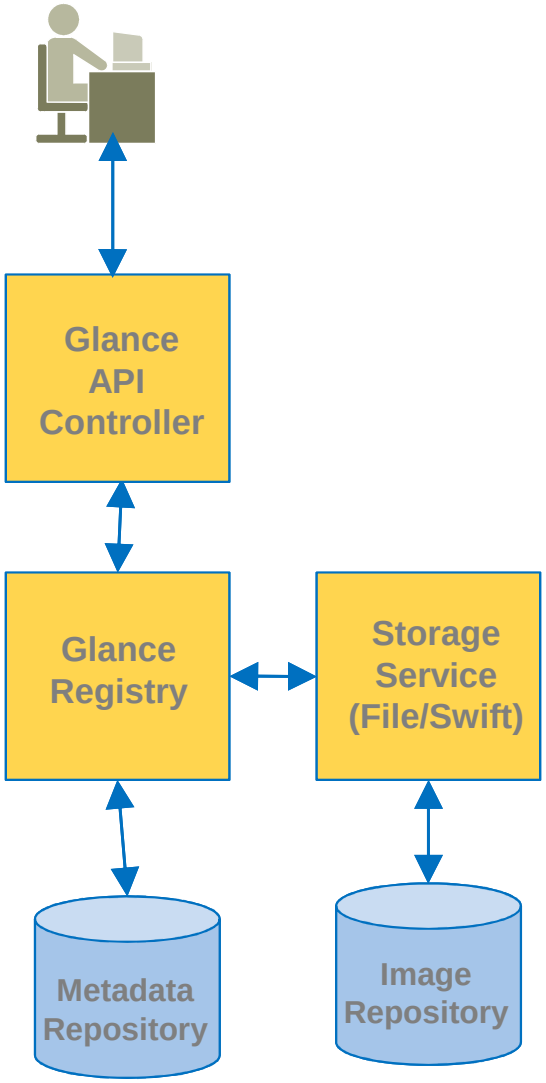
- All system data are stored in a MySQL Server
  - Instance info
  - Network info
  - Node info
- Python library SQL-Alchemy ORM
- SQLite for unit testing
- Other relational databases

Keystone Overview

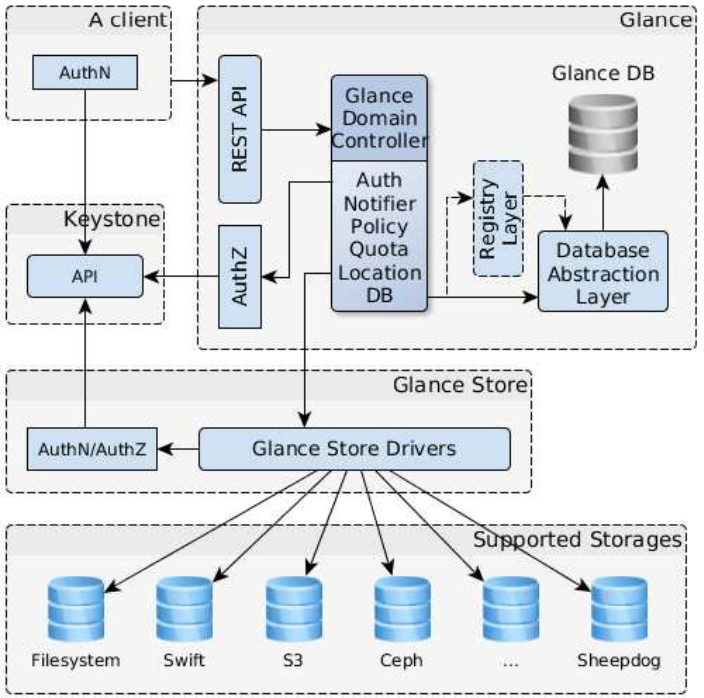


- Keystone acts as front-ends to various OpenStack services (compute, storage, etc.) for authentication and authorization (AA)
- Can function as an ID service on its own with SQLite or MySQL as ID server
  - Provides capabilities to create users and roles
- Supports multiple forms of authentication including user name and password credentials, token-based systems, and Amazon Web Services style logins
- Other ID services can be interfaced
- Can function as Service Catalogue (SC) to any client (users, applications, GUI)
  - SC is returned along with the token in response to an authentication request
  - SC contains following information
    - ***Service end-point*** (EP):
      `<service http address>:<port>/<service API version>/<tenant ID>`
    - ***Region*** in which service has been deployed

Image Service (Glance) Overview



- Meta-data about a [VM] image can be stored or updated in Glance Registry
- For actual storage of images, Glance registry can interface with
  - Swift, S3, Ceph or a File System
  - Can also interface with any web server (HTTP) for read-only data
- Meta-data stored in SQLlite or MySQL
- Glance does not scan the image to identify image parameters

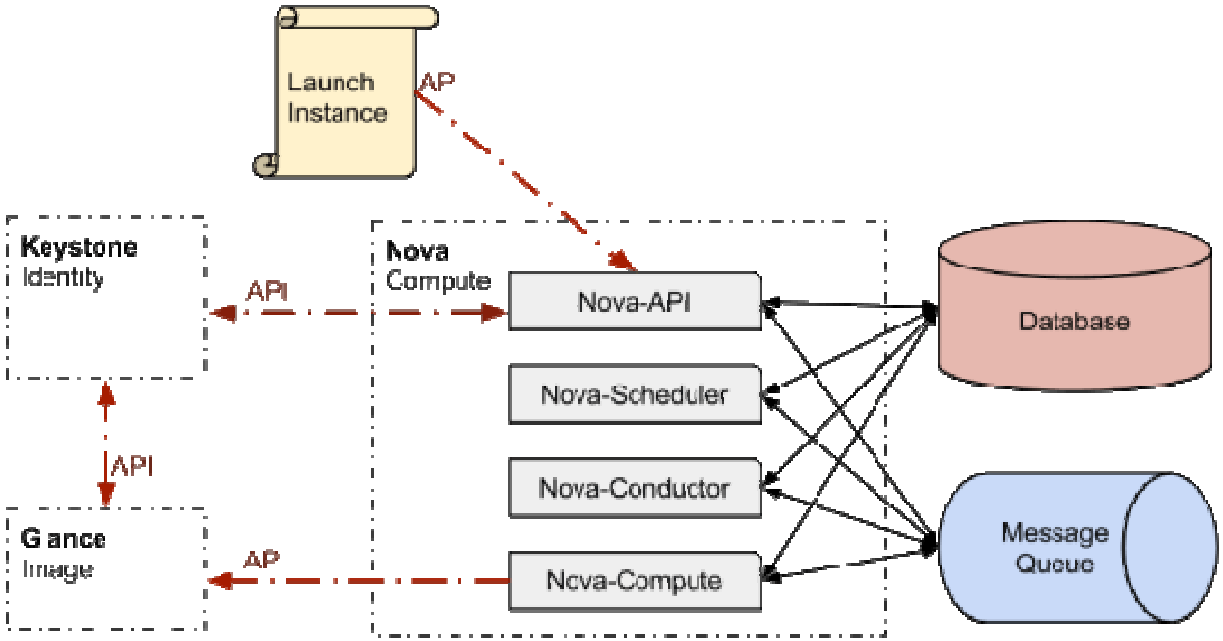


Nova Compute service

- Nova Compute service supports:
  - On-demand CRUD (Create / Read / Update / Delete) of instances (VMs)
  - On-demand attachment/detachment of VM to a network via Nova-Network
    - Nova-Network has been replaced by the Neutron service
  - On-demand attachment/detachment of block storage (“volume”) to/from VM
- Supports a number of different hypervisors
  - KVM
  - VMWare ESX/ESXi
  - XenServer, Xen Cloud Platform (XCP)
  - Hyper-V
- … but also lightweight container-based virtualization solutions
  - LXC Linux Containers
  - UML User Mode Linux
- … but also instances directly instantiated on bare-metal hardware (no virtualization)

Nova Compute service

- Nova interacts with Keystone for authentication, Glance for images and Horizon for web UI



Compute Instances

- Servers
  - An abstraction of running VM instances or virtual servers
  - A compute instance is associated to a set of resources
    - Flavor
    - Image
    - IPv4/6 addresses
    - Metadata
      - user specified, such as server name
- Flavors
  - Templates of hardware resources associated to a running instance
  - Example:
    - m1.medium:
      Memory: 4096MB,
      VCPUS: 2,
      Storage: 40GB,
      Swap: 0GB,
      RXTX Quota: 0GB,
      RXTX Cap: 0MB
  - Admin can create new flavors:

```
nova-manage instance_type create m1.mega 32768 16 320 0 0 0
```

- Image
  - Images can be used as templates when setting up new servers
  - OS image
  - VM disk
  - Other files

Nova-Volume Service (Cinder)

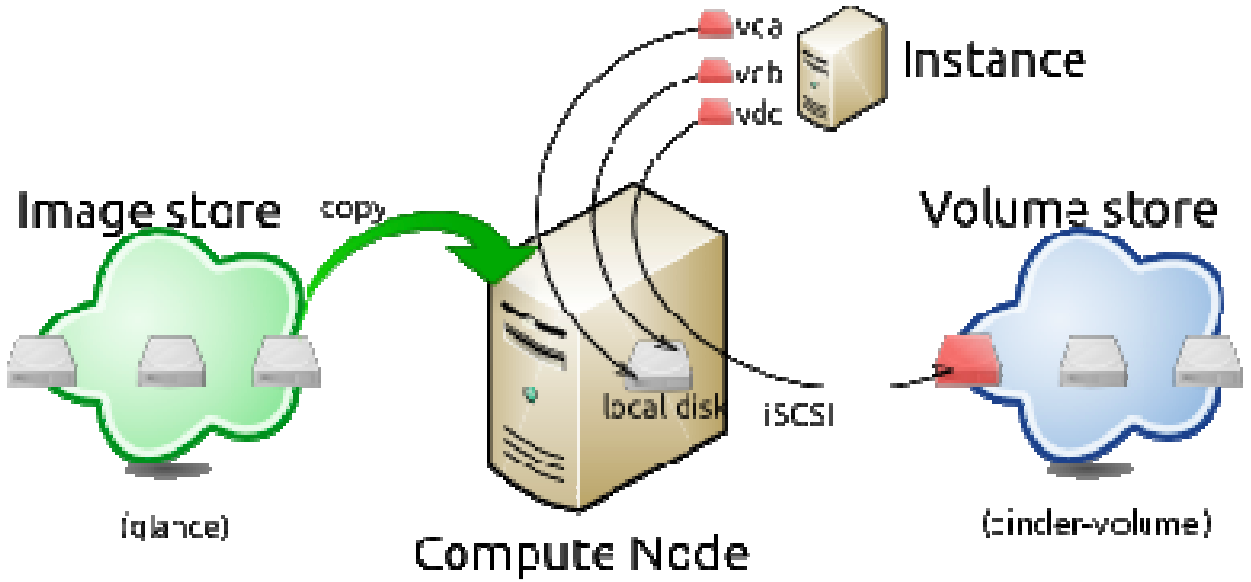
- Provides a persistent Block Storage Service for the instances running in Nova
- Create / Delete / Connect volumes to running instances via iSCSI
- Snapshots can be taken to create backups or to create new block storage volumes (e.g. to clone an instance)
- Different drivers available to physically connect to different storage systems
  - LVM / iSCSI
  - SAN drivers
  - Ceph

Nova-Scheduler Service

- Determines the placement of new resources requested via the API
- Modular architecture to allow for optimization
- Base Schedulers include
  - Round Robin
  - Filter Scheduler
  - Spread First
  - Fill First
  - Chance (random)

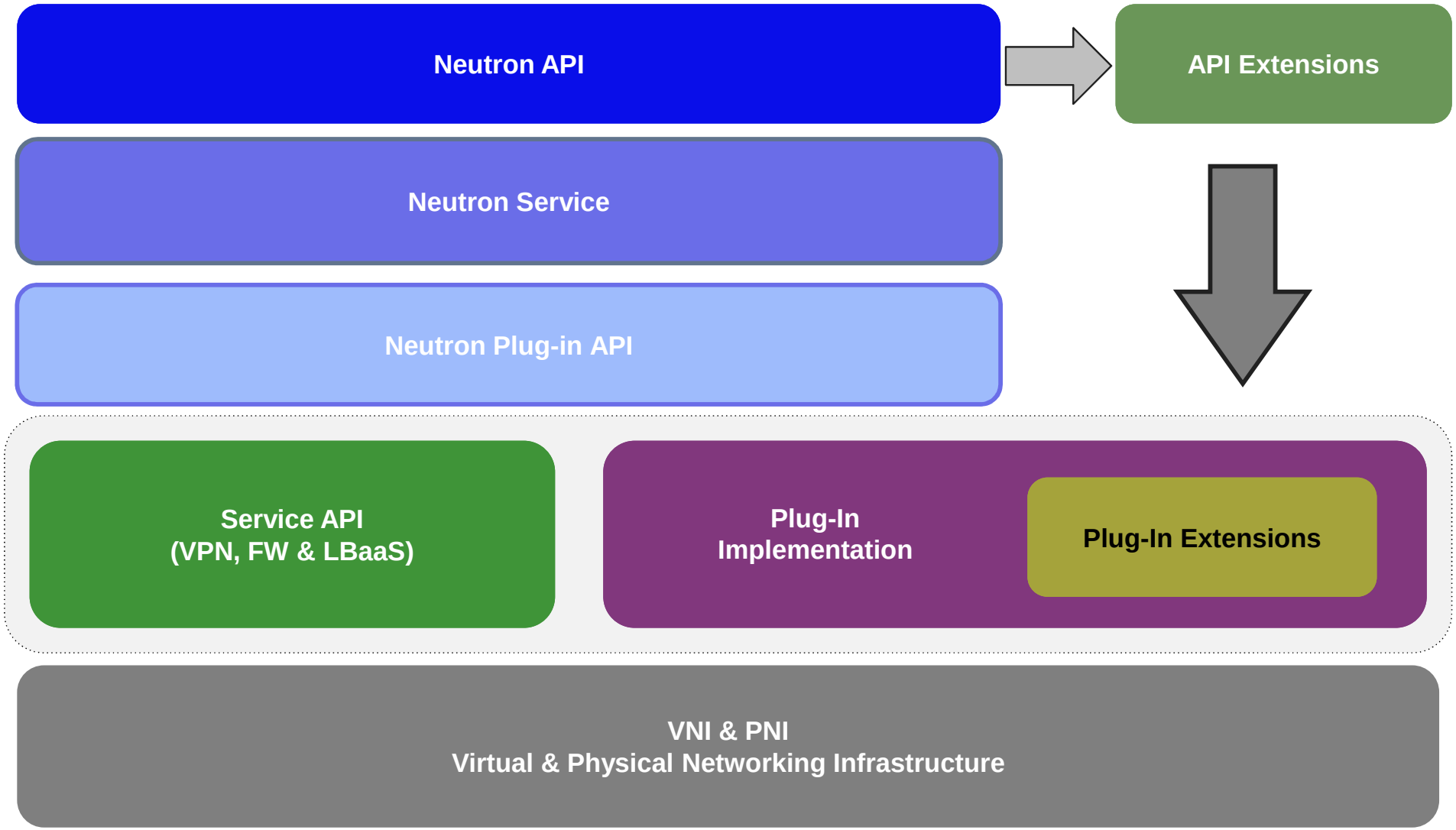
Nova compute: instance creation and storage

1. Image is copied from the Image store to the Compute node
2. A volume is made available to the VM from the Volume store through the Cinder service
3. The VM is activated in the Compute node
   - Some storage volumes live in the instance local storage
     - Destroyed when the instance is terminated (*ephemeral storage*)
   - Others are accessed through iSCSI (requires initiator sw in the VM)
     - Survive the instance termination (*persistent storage*)
     - Can be attached to another instance after instance termination



Neutron architecture

- Provides REST APIs to manage network connections for the resources managed by other services
- Modular design: API specifies service, vendors provide their implementation
  - Extensions for vendor-specific features



OpenStack deployments

- Deploying an OpenStack Cloud is a difficult task, as many alternative choices are possible
  - if one has enough hardware resources …
- A typical real-worls deployment of OpenStack relies on
  - N nodes acting as Controller and API nodes (N>1 for *High Availability*, HA)
  - K nodes acting as Network node
  - M nodes acting as Compute nodes
- To automatically install and configure the OpenStack services on a cluster of servers, several OpenStack distributions have been developed over the years
  - E.g. Mirantis Fuel, Red Hat Enterprise Linux OpenStack Platform, Ubuntu OpenStack , Cisco Metapod HP Helion OpenStack , Rackspace Private Cloud, IBM Cloud Manager, Oracle OpenStack , …
- For testing purposes, one can install all the core services in a single VM using DevStack
  - See tutorial