Virtual Computing Laboratory
Institute for Digital Research and Education (IDRE)

OpenStack/Nova

Infrastructure as a Service (IaaS)

Prakashan Korambath
January 24, 2013
ppk@idre.ucl.a.edu

Essex Distribution

Ubuntu 12.04 LTS

Resources

Compute Nodes

- 16 IBM iDataplex 2U units – Intel Nehalem –E5530
    - 2 sockets per node
    - 4 cores per socket
    - 8 cores per node
    - 24 GB memory per node

- Total 256 cores and 768 GB memory

Controller Node

- AMD Opteron
- Six Core Opteron 2431
- 12 cores in total and 32 GB memory

Network Switches = 2

Design and Development Team

- All are welcome to participate!
  - ppk@idre.ucla.edu
  - Just attend the meeting and see if you can contribute some way.

- Meets every Tuesday between 9:30 and 10:30 am in room 3909 MSA (Monthly UC wide meetings are now organized by Paul Weakliem @ UCSB)
- Presentations on Cloud computing and storage, Grid Computing and Grid Data Transfer using Globus Online.
- Architecture design discussions, deployment and support
- Software development updates and issues

- Web Resources and Documentation
  - http://storage1.ats.ucla.edu/mediawiki/index.php/Main_Page
  - https://service.vcl.ucla.edu/dev/

  - https://sites.google.com/site/hpccloud/
  - https://sites.google.com/site/hpcconsultantguide/
  - https://sites.google.com/site/hpcconsultantguide/virtualization/kvm-on-ubuntu-12-04

Attributes of Cloud

- On-demand self-service - A cloud should enable self-service, so that users can provision servers and networks with little human intervention.
- Network access - Any computing capabilities are available over the network and you can use many different devices through standardized mechanisms.
- Resource pooling - Clouds can serve multiple consumers according to demand.
- Elasticity - Provisioning is rapid and scales out or in based on need.
- Metered or measured service - Just like utilities that are paid for by the hour, clouds should optimize resource use and control it for the level of service or type of servers such as storage or processing.

SaaS, PaaS, IaaS

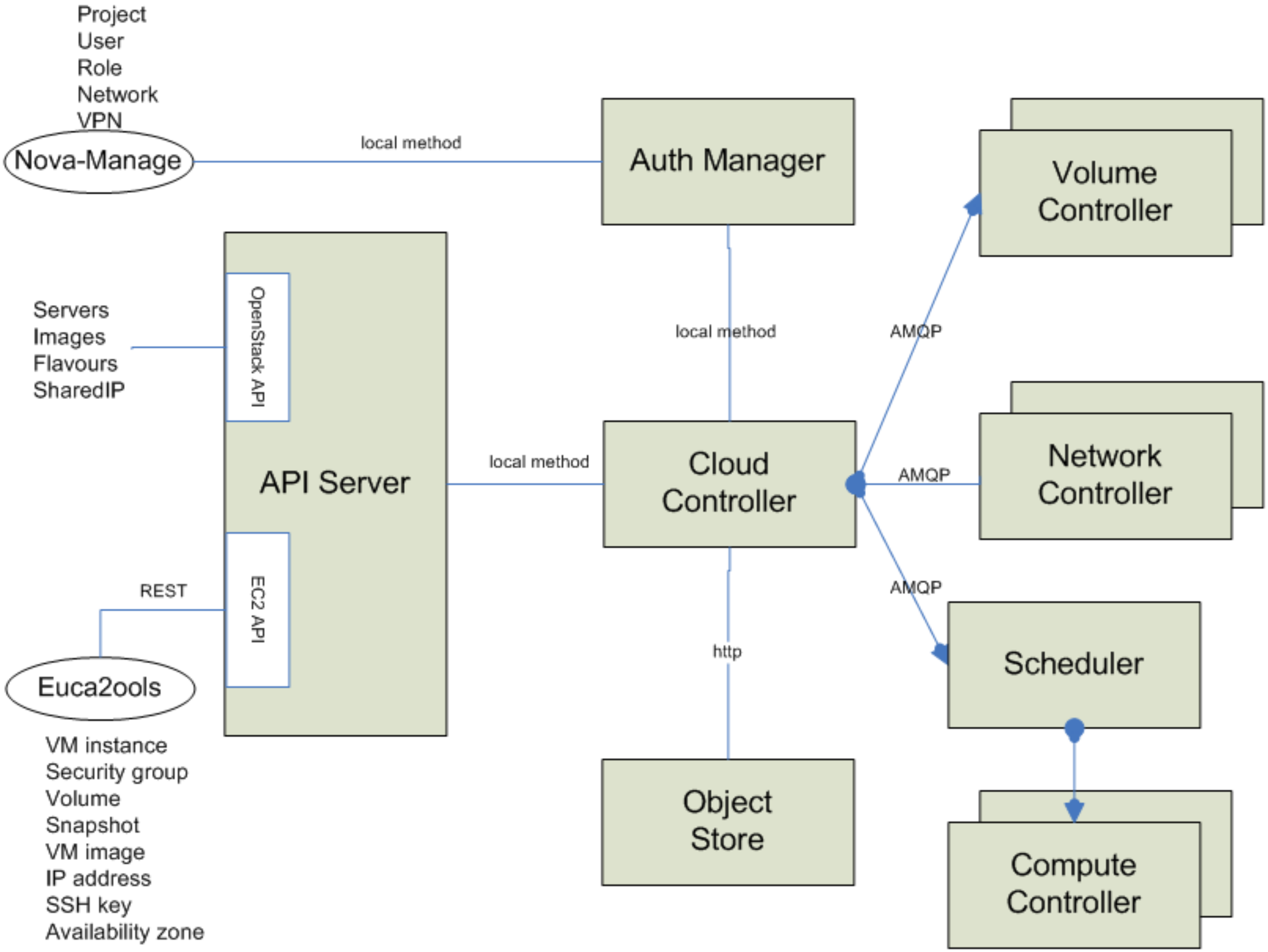
OpenStack

- Global collaboration of developers and cloud computing experts in producing an open source platform for private and public clouds.
- Technology
  - Web
  - REST
  - HTTP
  - Python
- Services
  - Compute (Nova)
  - Object Storage (Swift)
  - Image Service (Glance)
  - Identity (Keystone)
  - Dashboard (Horizon)
    - Each service is a webApp
    - REST API server (front end)
    - One or more back end servers
    - Messaging interface between them
  - API's use HTTP + json (or xml)
    - Use curl or wget or browser plugins
    - Any programming language via HTTP libraries
    - Use the Python novaclient library

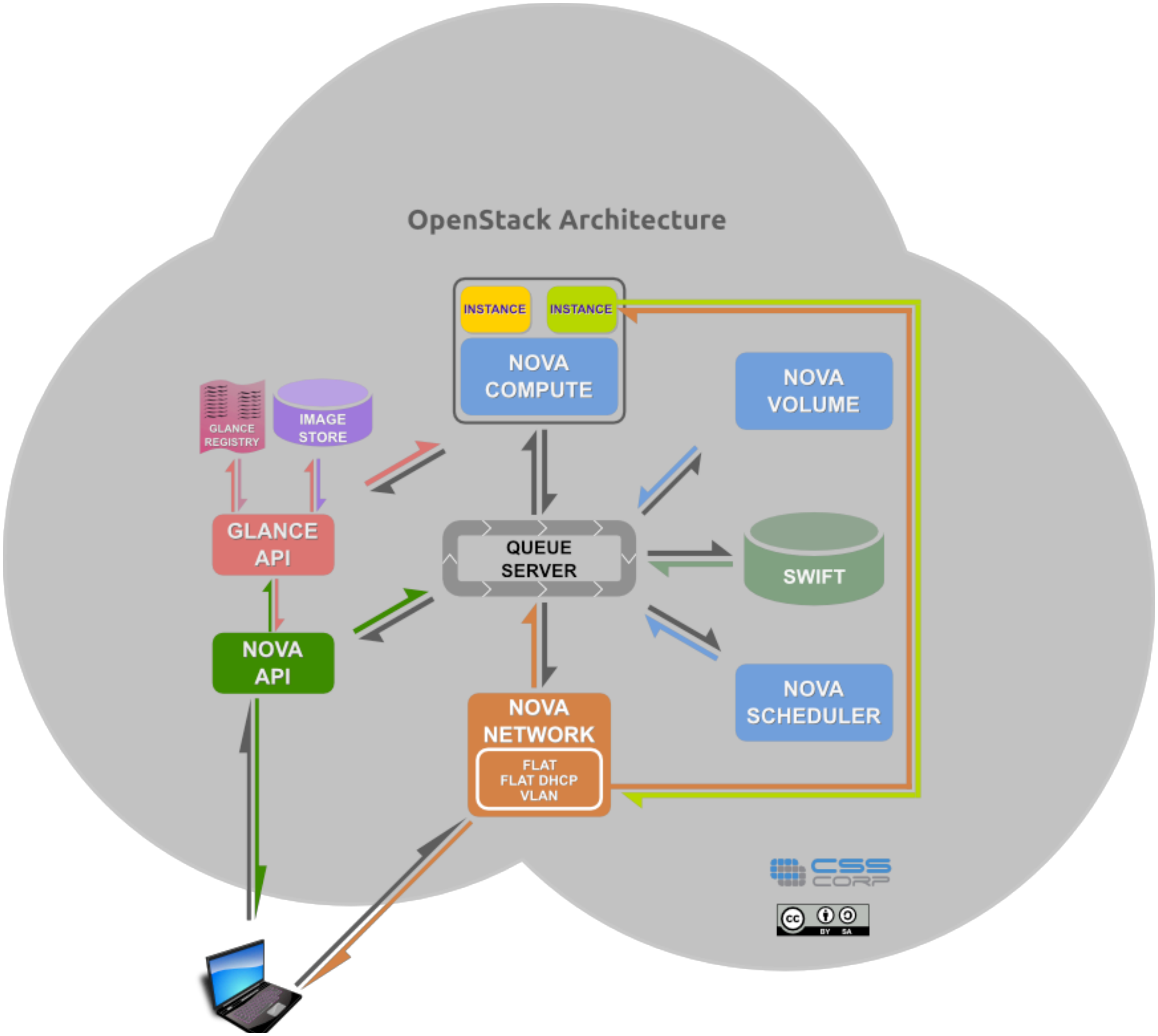
Concepts

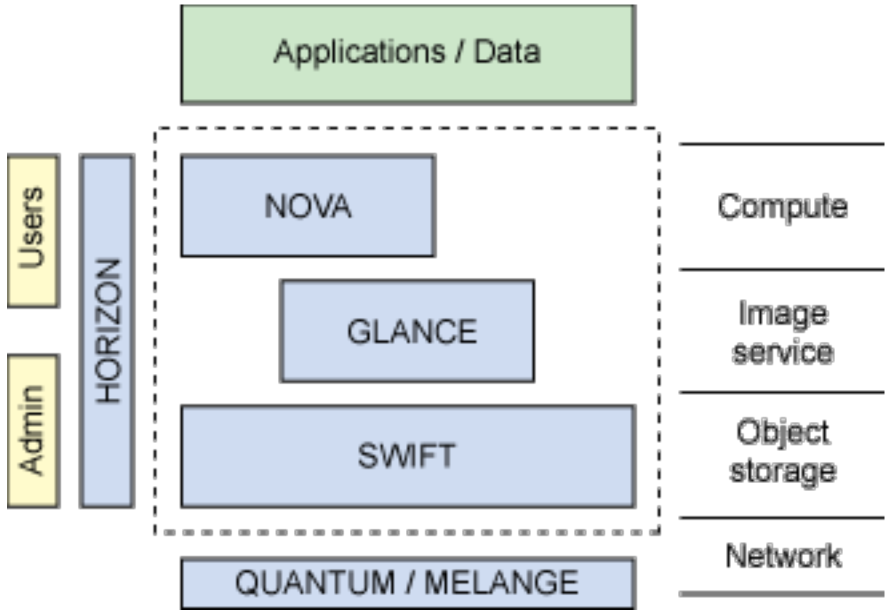
- Users and Projects, Cloud Admins, Project Manager etc. (Role Based Access)
  - access to images is limited by project
  - access/secret are per user
  - keypairs are per user
  - quotas are per project
- Virtualization - Hypervisors
  - KVM
  - Xen
  - HyperV
  - LXC
  - Qemu
- Images and Instances
  - A virtual machine running inside a cloud environment
- System Architecture – Shared Nothing, messaging
  - Cloud Controller
  - Compute Controller
  - Object Store
  - Volume Controller
  - Auth Manager
  - Network Controller
  - Scheduler
- Communication through HTTP GET POST

Architecture

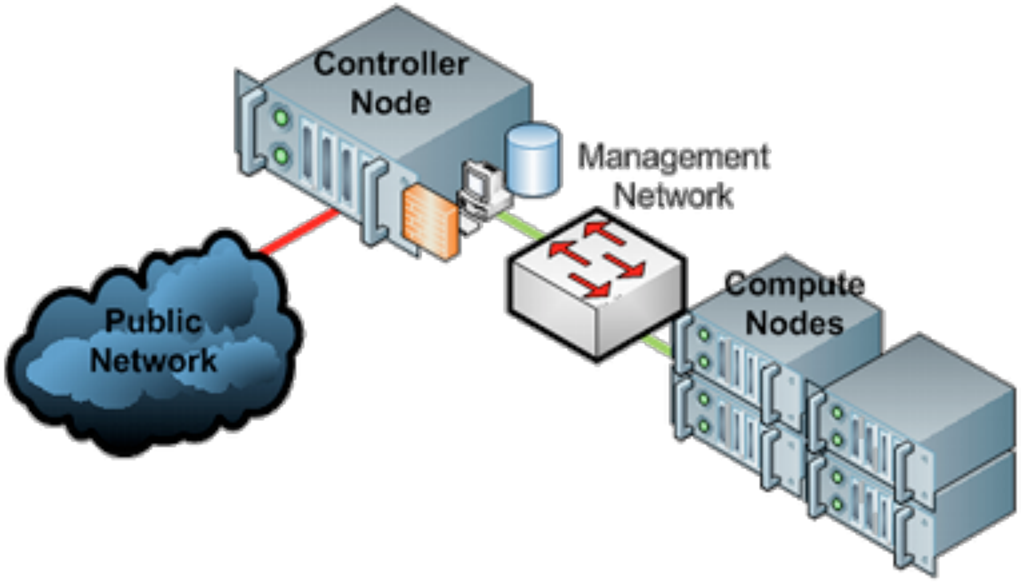


Components of an OpenStack Distribution

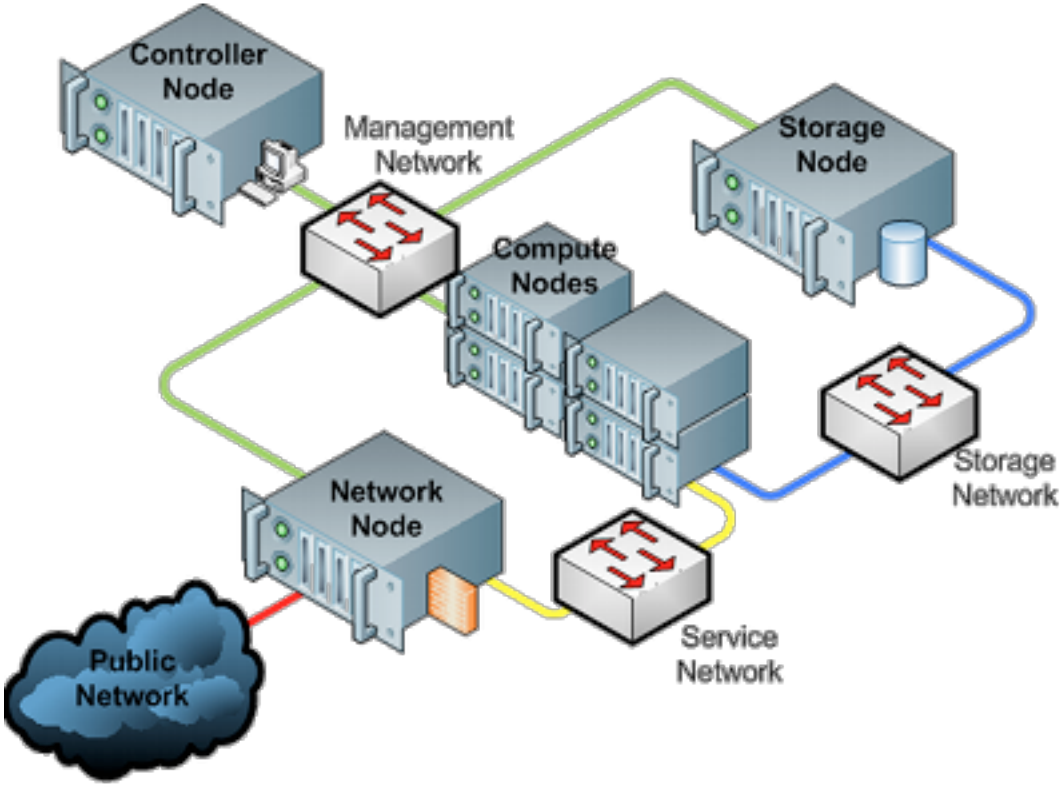



Applications / Data
Users
Admin
HORIZON
NOVA
GLANCE
SWIFT
QUANTUM / MELANGE
Compute
Image service
Object storage
Network

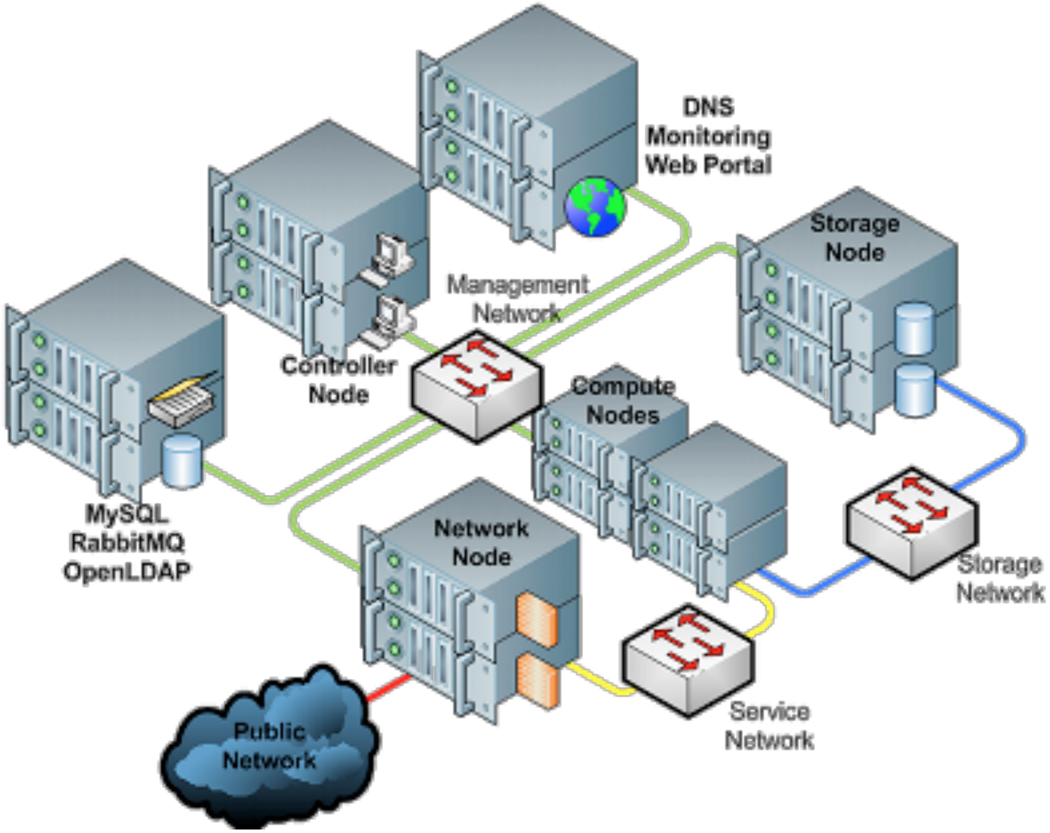
Hardware for Building a IaaS Cloud Platform I



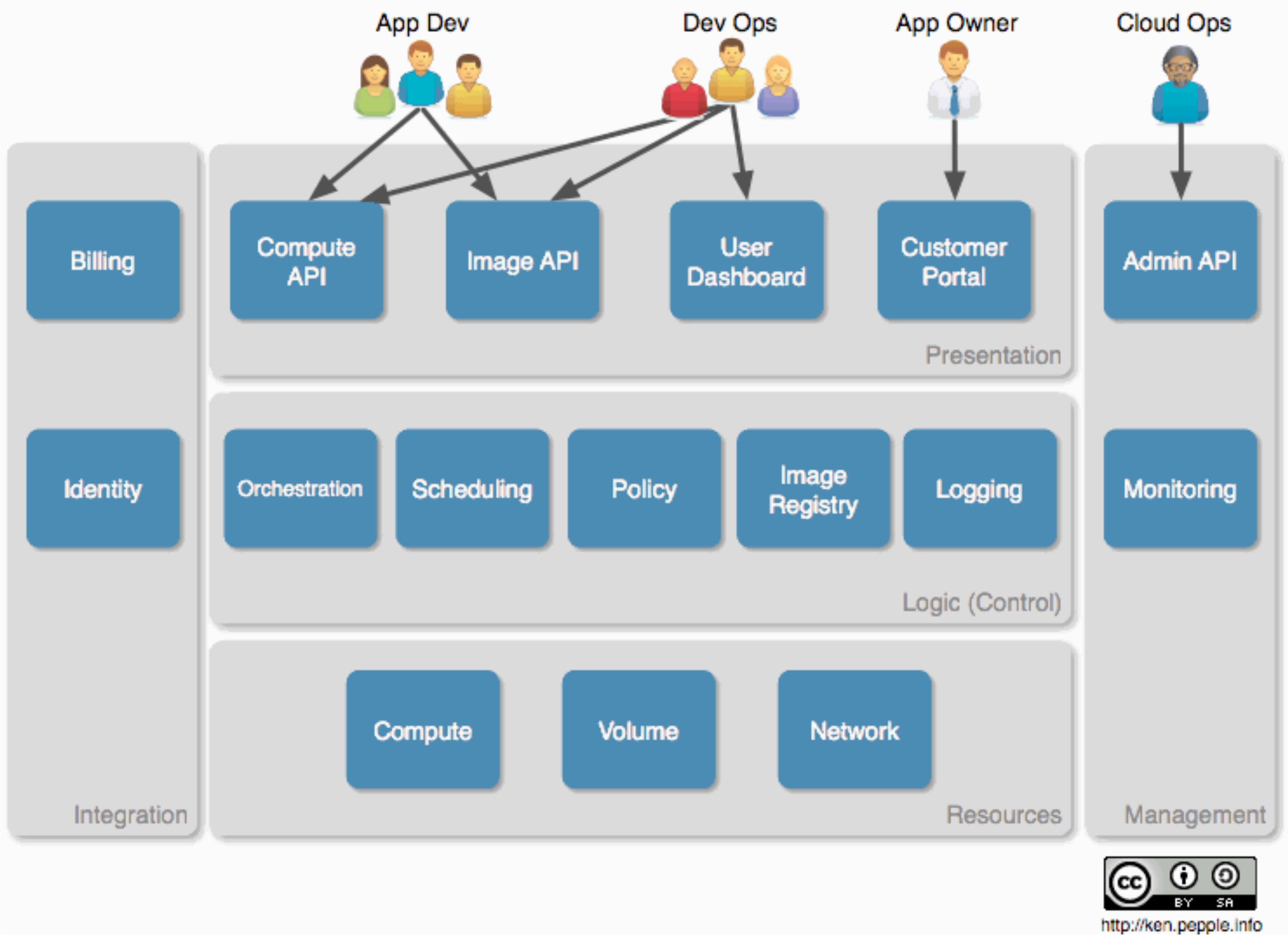
Hardware for Building a IaaS Cloud Platform I I



Hardware for Building a IaaS Cloud Platform III



Cloud Provider Conceptual Architecture



OpenStack Compute – Logical view

nova-api supports OpenStack API, EC2 API

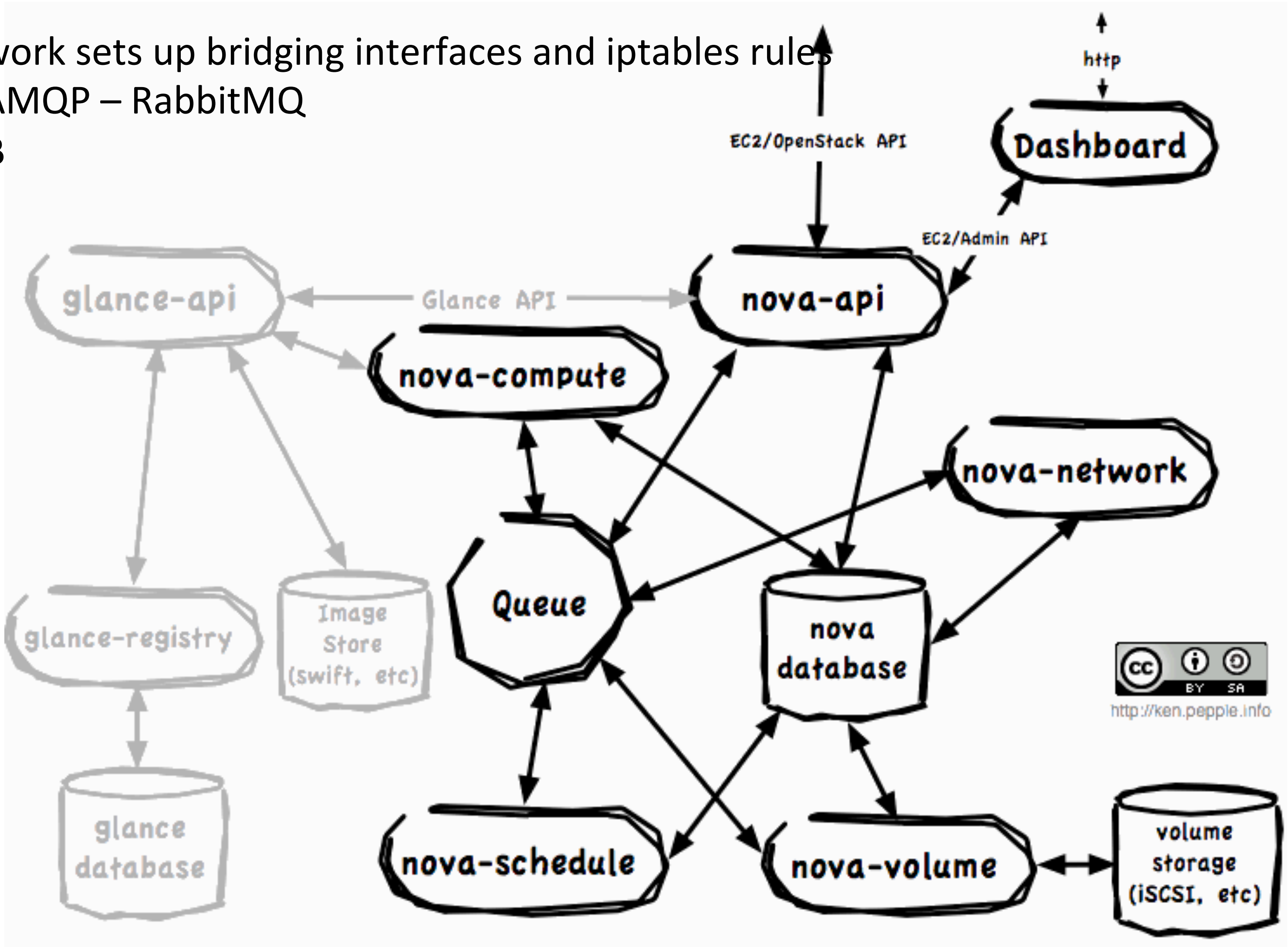
nova-compute creates and terminate virtual machine instances via libvirt for KVM and others

nova-volume manages the creation, attaching and detaching of persistent volumes to compute Instances

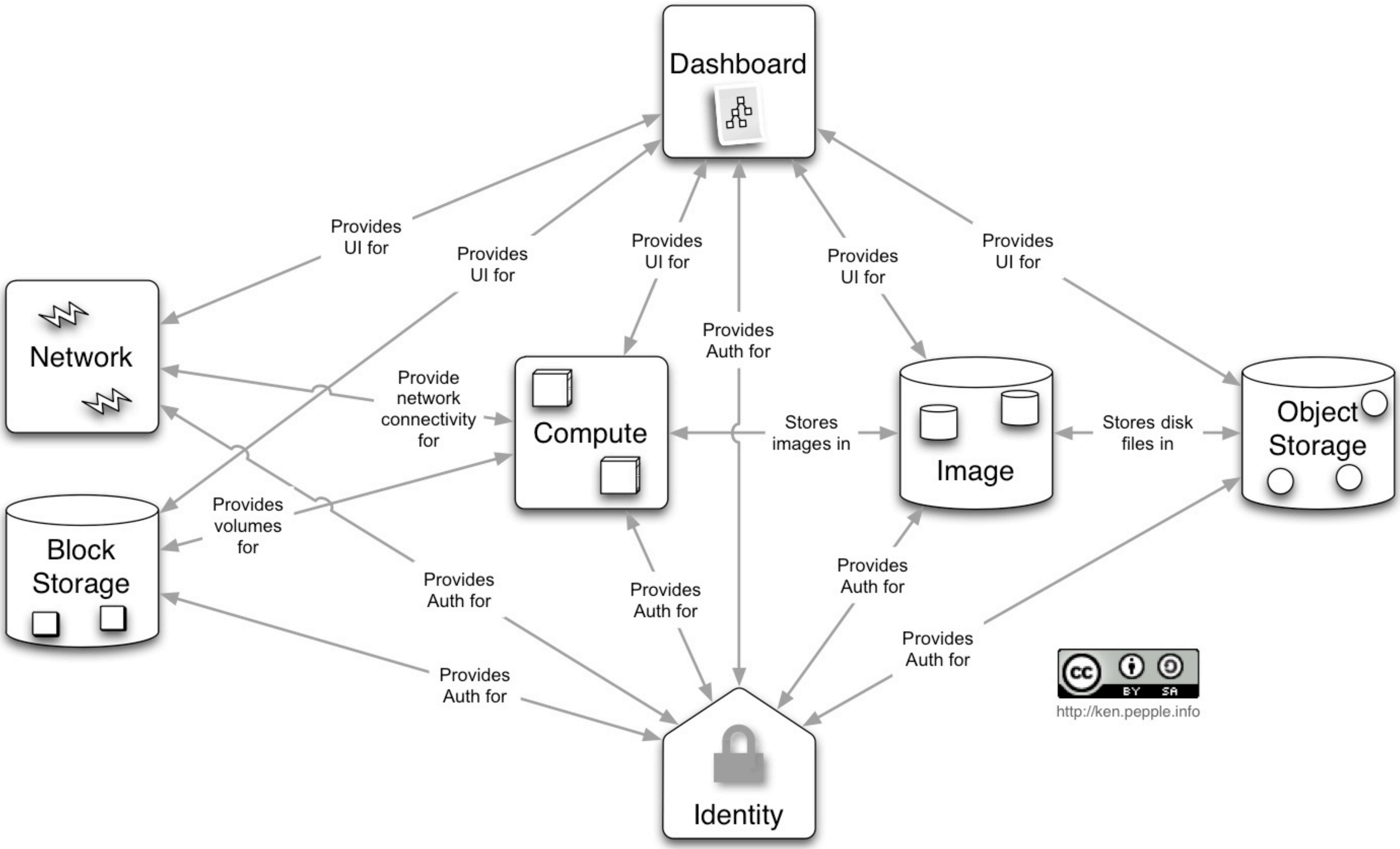
nova-network sets up bridging interfaces and iptables rules

Queue – AMQP – RabbitMQ

MySQL DB

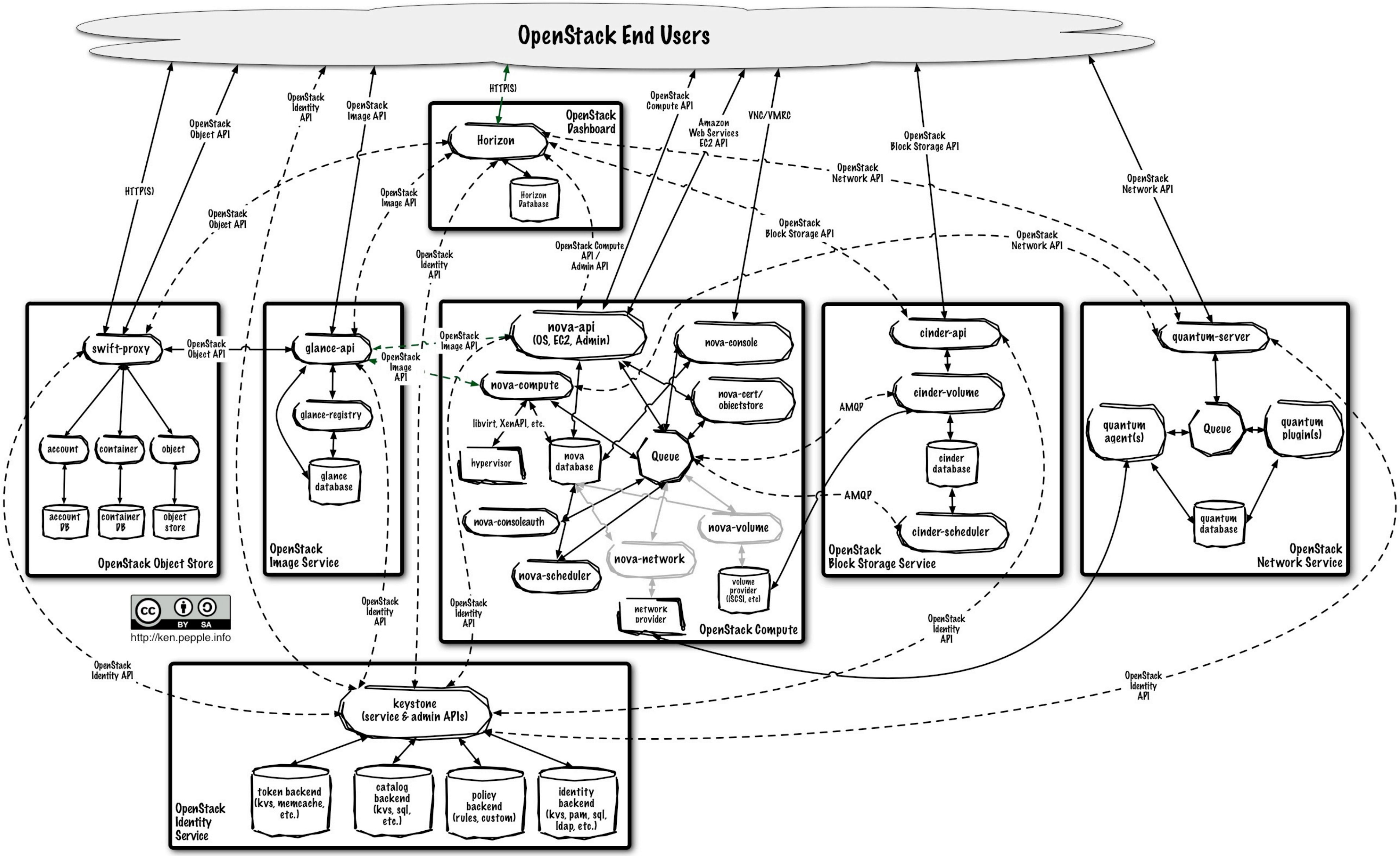


Conceptual Architecture

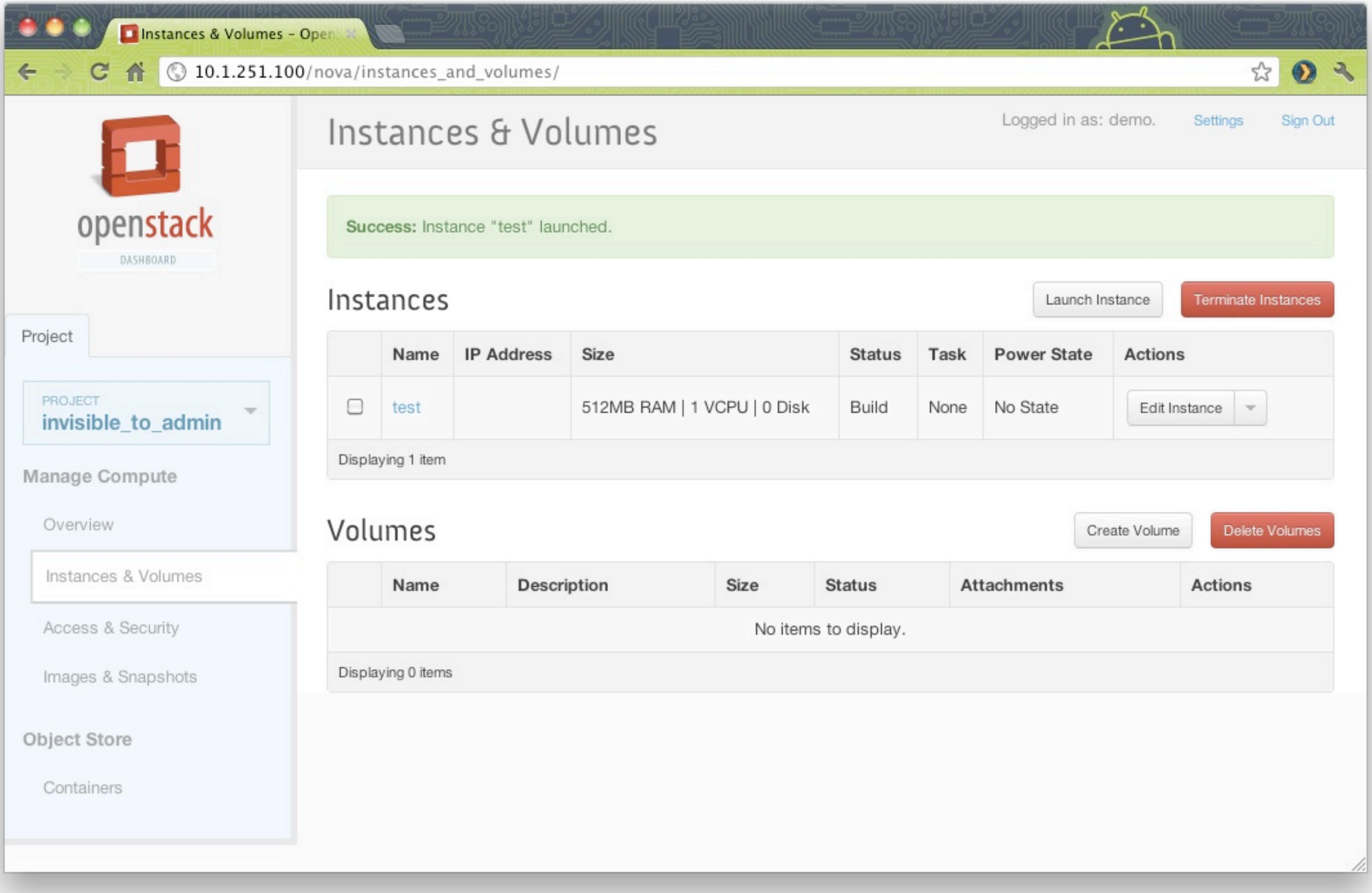


Logical Architecture

(quickly becoming messy and complicated)



Dashboard



Instances & Volumes

Success: Instance "test" launched.

Instances

	Name	IP Address	Size	Status	Task	Power State	Actions
□	test		512MB RAM | 1 VCPU | 0 Disk	Build	None	No State	Edit Instance

Displaying 1 item

Volumes

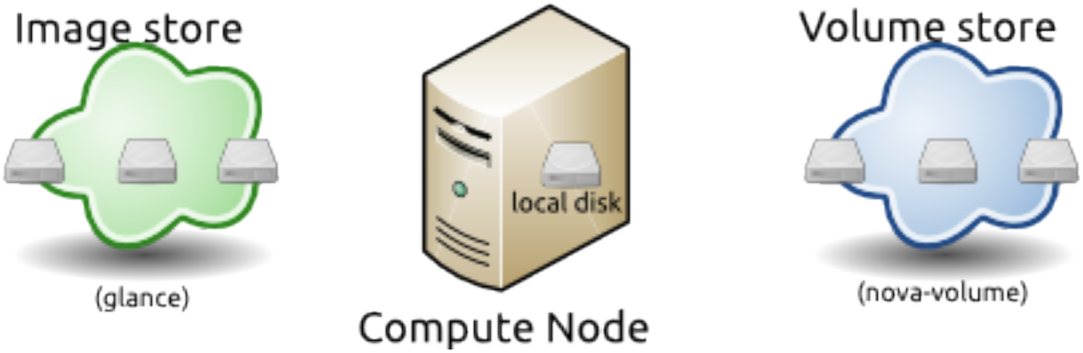
	Name	Description	Size	Status	Attachments	Actions
No items to display.						

Displaying 0 items

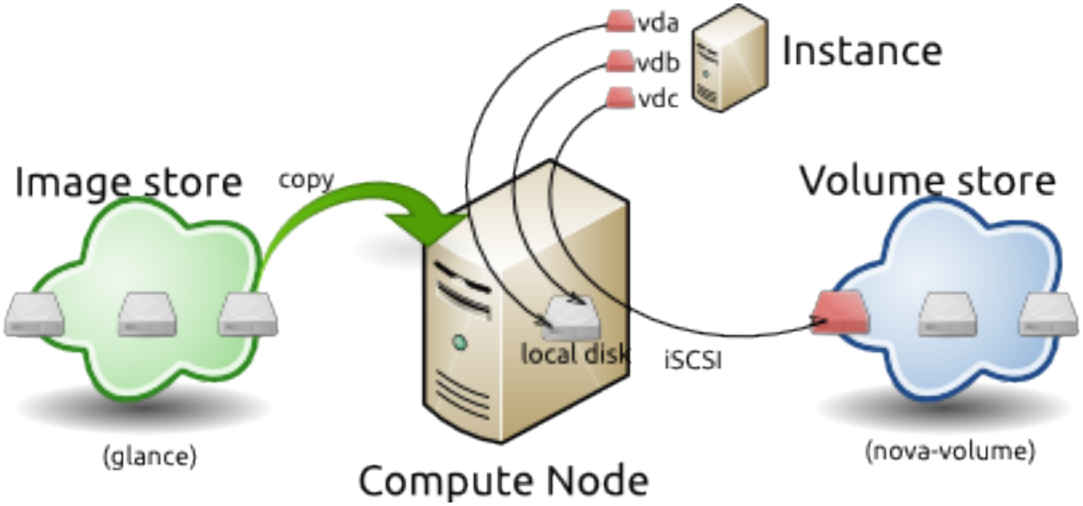
What is happening during Instance deployment?

(nova-volume has a similar function to EBS)

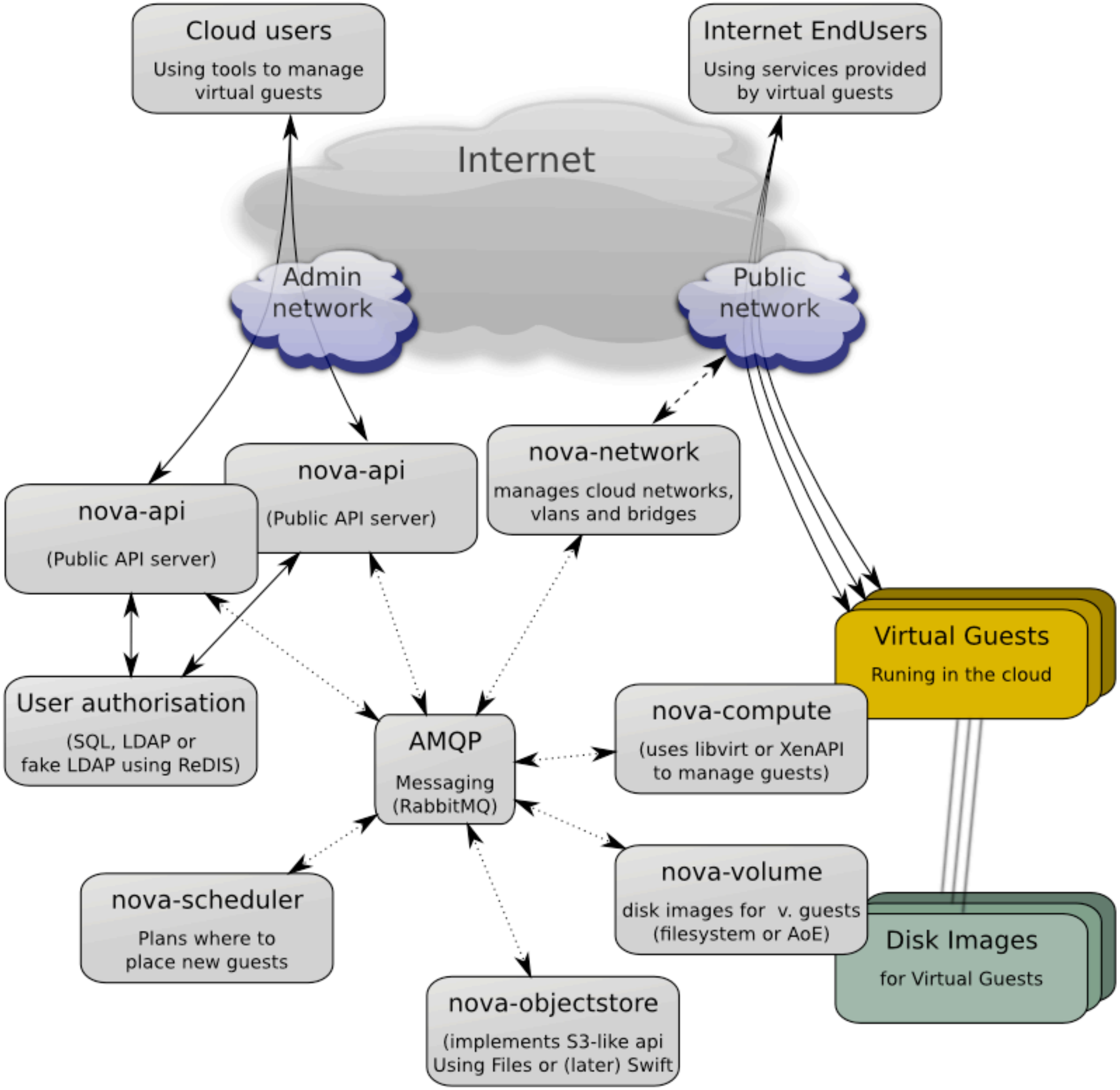
Initial State



Instance Launching



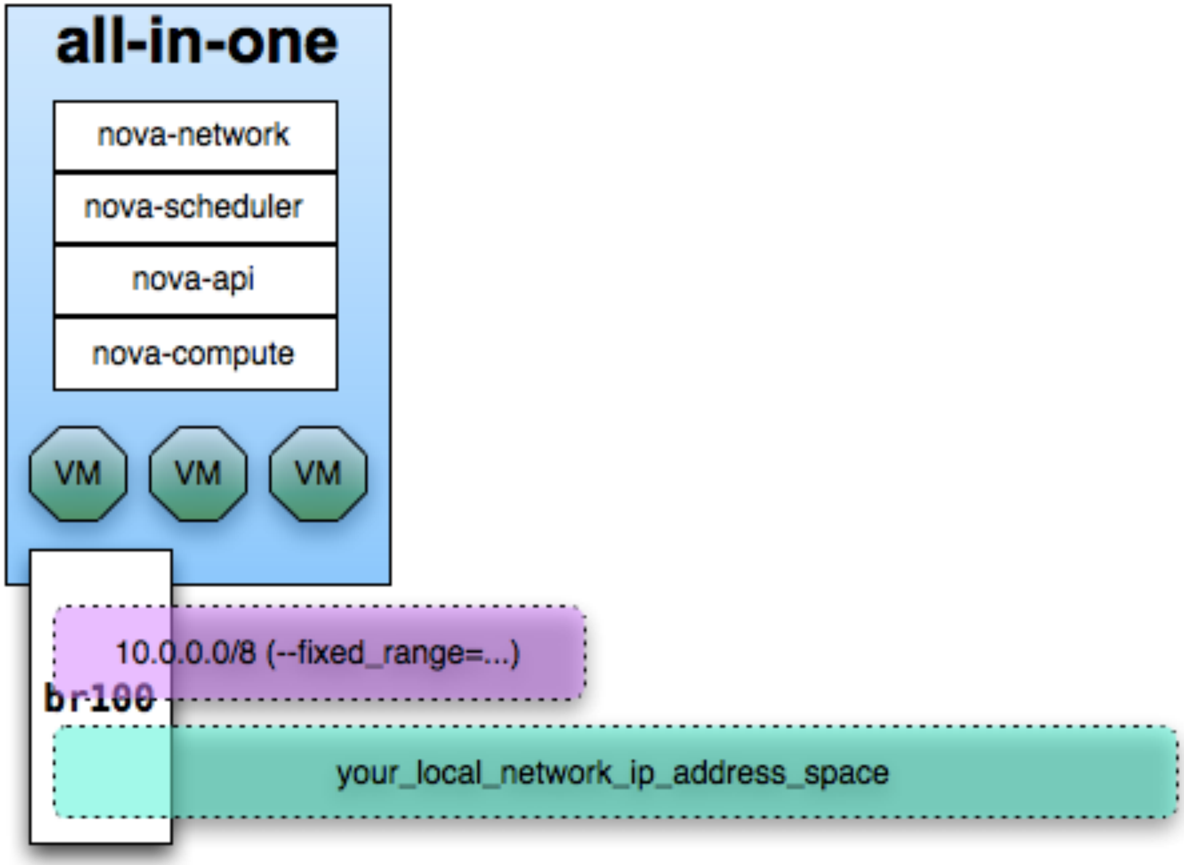
Service Architecture


Cloud users
Using tools to manage virtual guests
Internet EndUsers
Using services provided by virtual guests
Internet
Admin network
Public network
nova-api
(Public API server)
nova-api
(Public API server)
nova-network
manages cloud networks, vlans and bridges
Virtual Guests
Runing in the cloud
User authorisation
(SQL, LDAP or fake LDAP using ReDIS)
AMQP
Messaging (RabbitMQ)
nova-compute
(uses libvirt or XenAPI to manage guests)
nova-volume
disk images for v. guests (filesystem or AoE)
Disk Images
for Virtual Guests
nova-scheduler
Plans where to place new guests
nova-objectstore
(implements S3-like api Using Files or (later) Swift

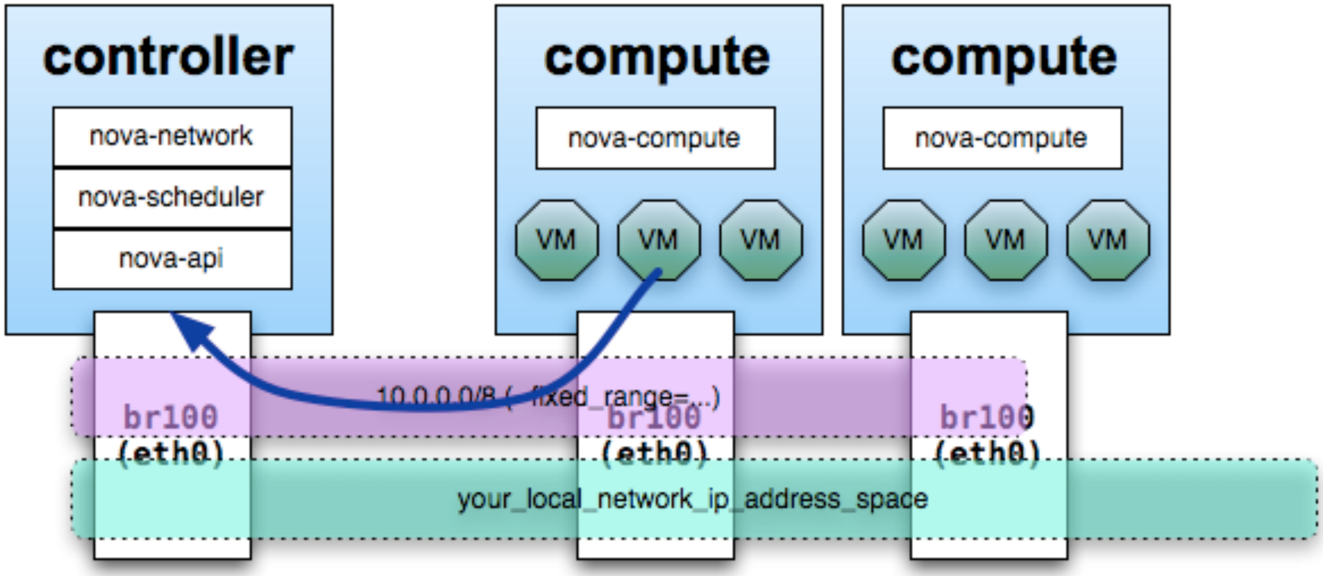
Networking

- Flat Mode
  - Fixed IB through the bridged network (br100) configured manually (/etc/network/interfaces)
- Flat DHCP
  - Runs dnsmasq as dhcp server listening on the bridge. Instances still get fixed IP addresses through dhcp.
- VLAN DHCP
  - Needs a switch that supports VLAN tagging. VLAN and bridge for each project. Private IPs are accessible only inside the VLAN

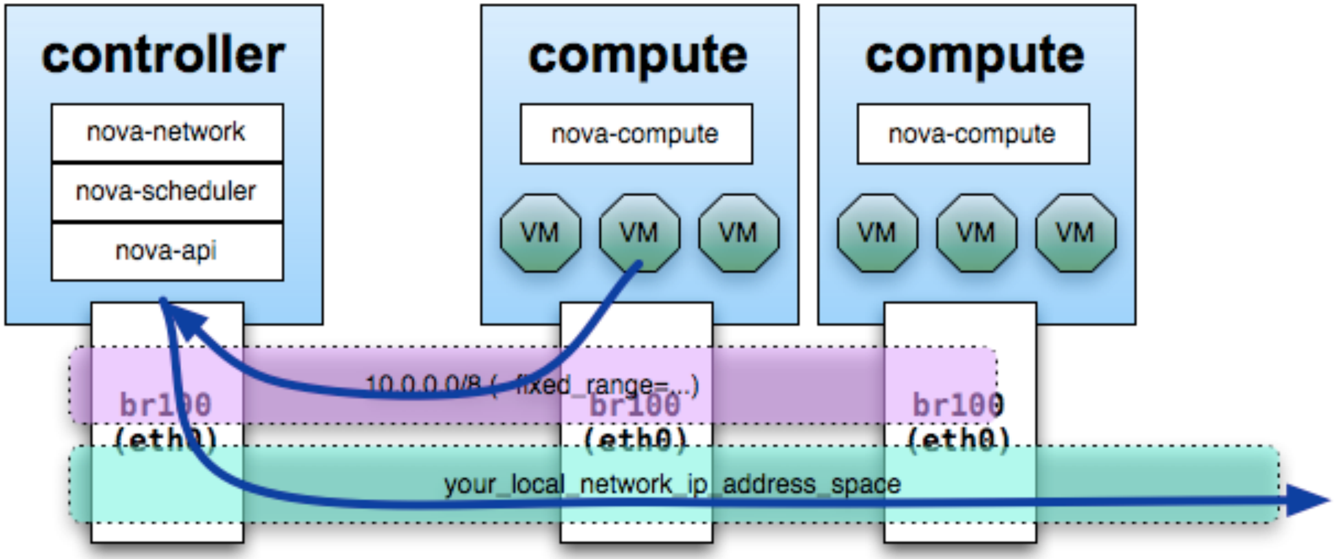
Flat network, all-in one server installation



Single adaptor hosts, first route

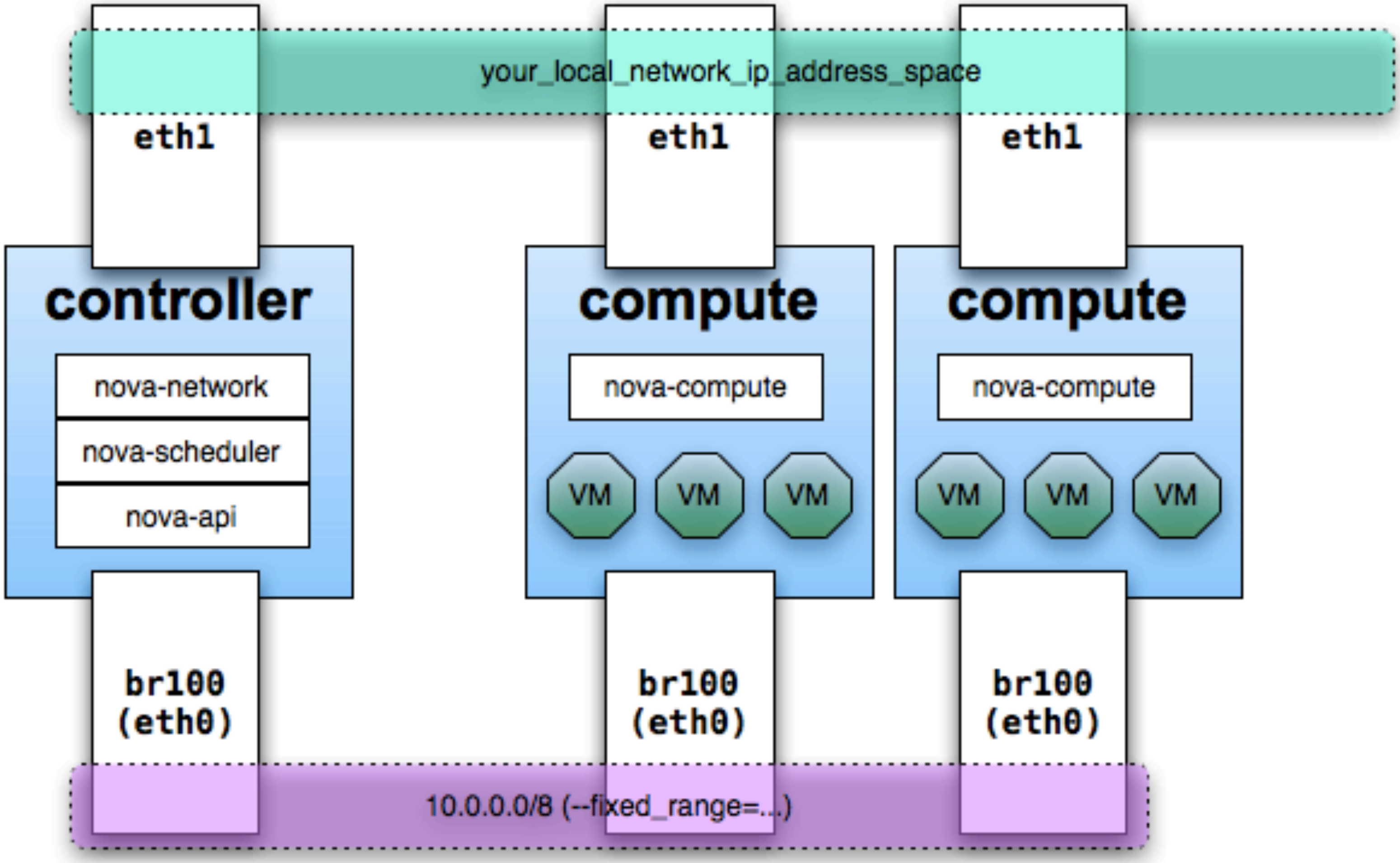


Single adaptor hosts, second route

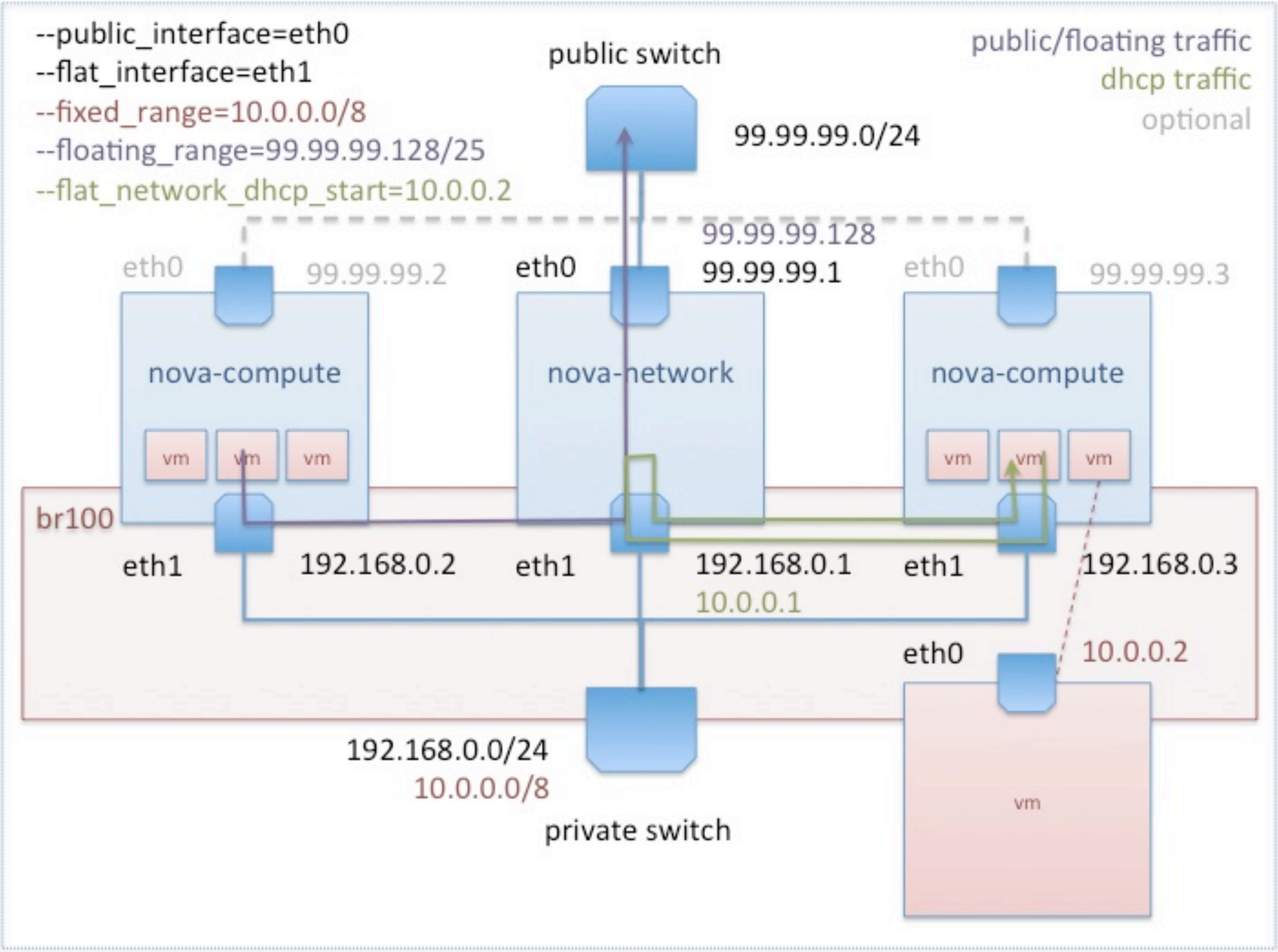


Flat network, multiple interfaces, multiple servers

- eth1 is used for physical maintenance of host operating system
- eth0 is used for deploying the virtual instances and network traffic to the instances

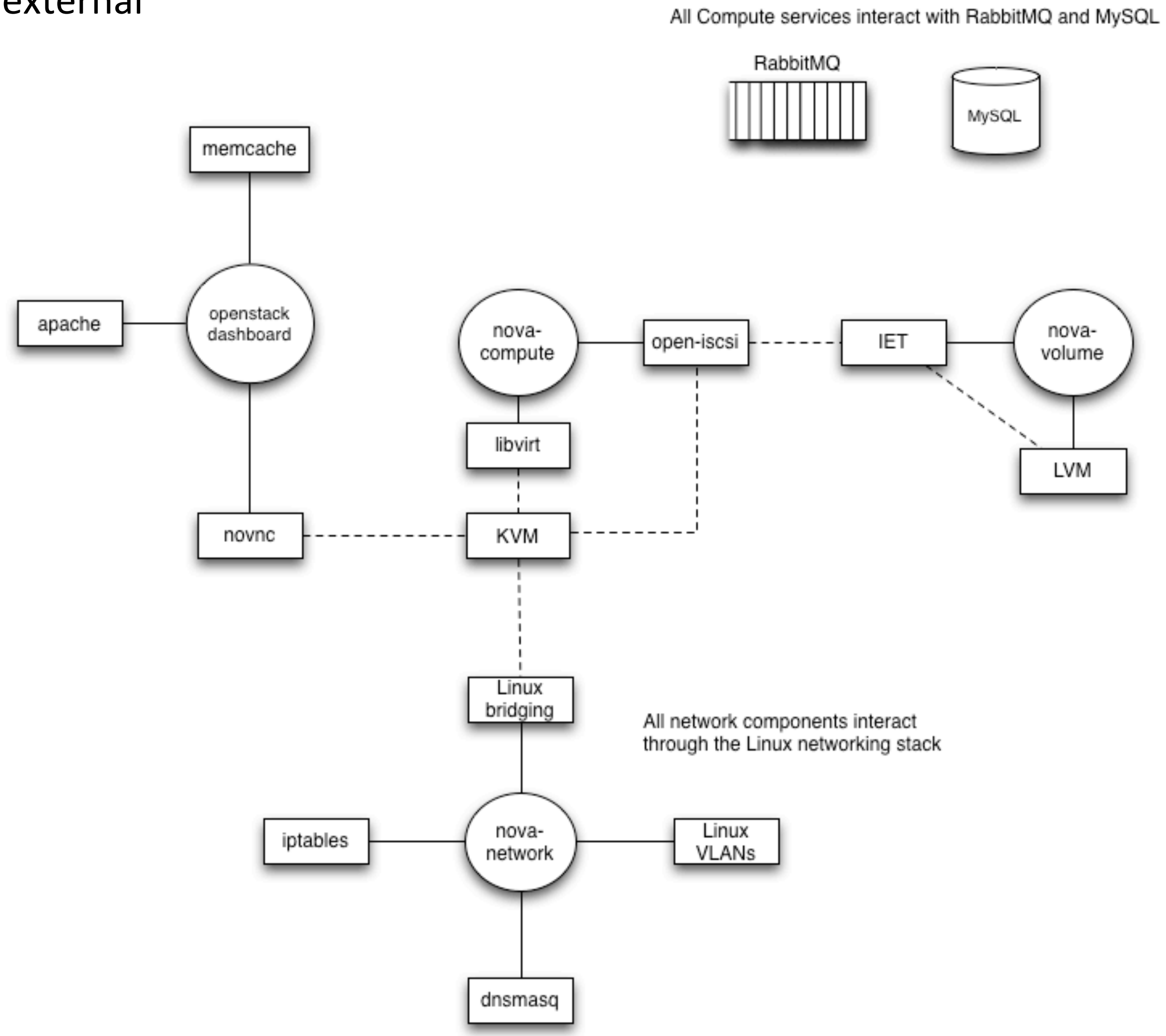


Flat DHCP network, multiple interfaces, multiple servers with libvirt driver



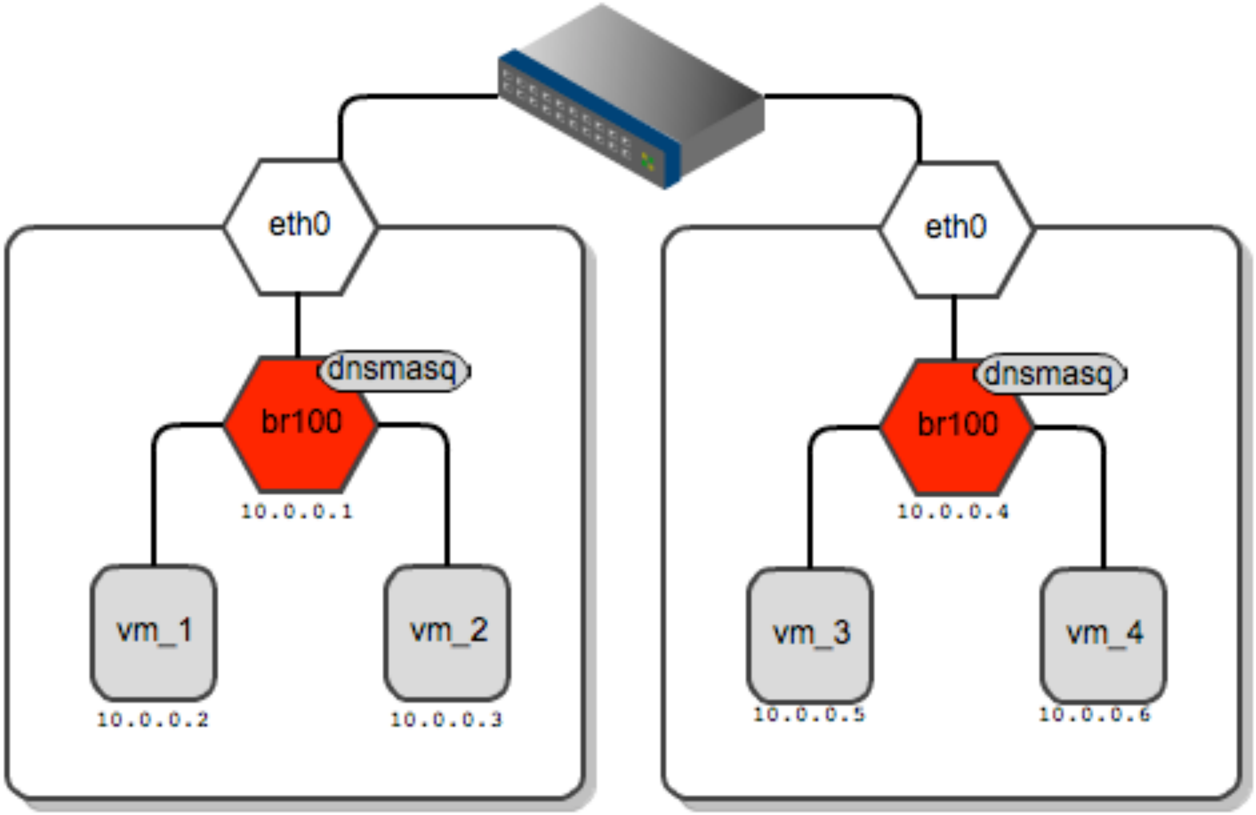
Underlying Technologies

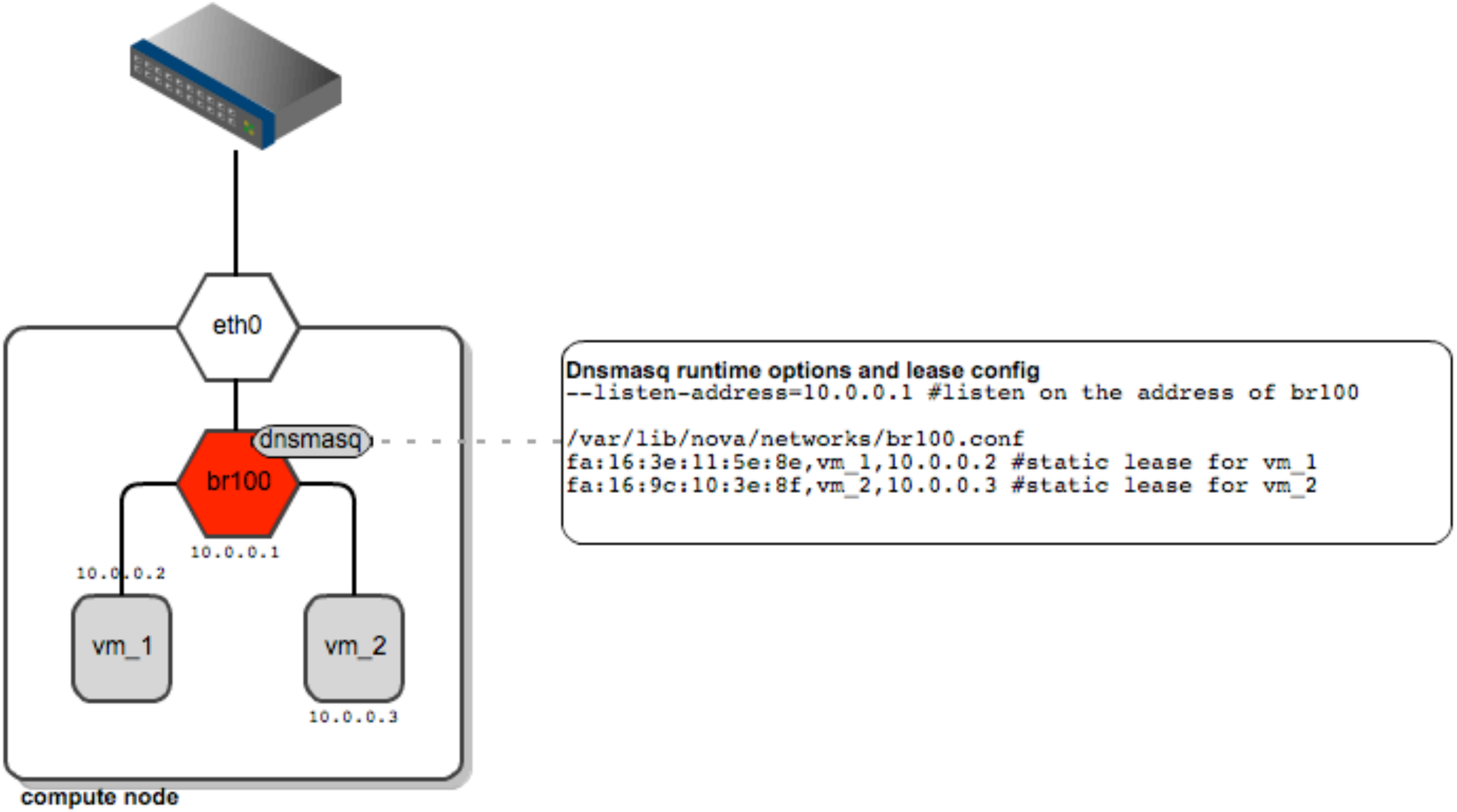
Circles are Linux services that are part of OpenStack compute
Rectangles are external



All Compute services interact with RabbitMQ and MySQL

All network components interact through the Linux networking stack

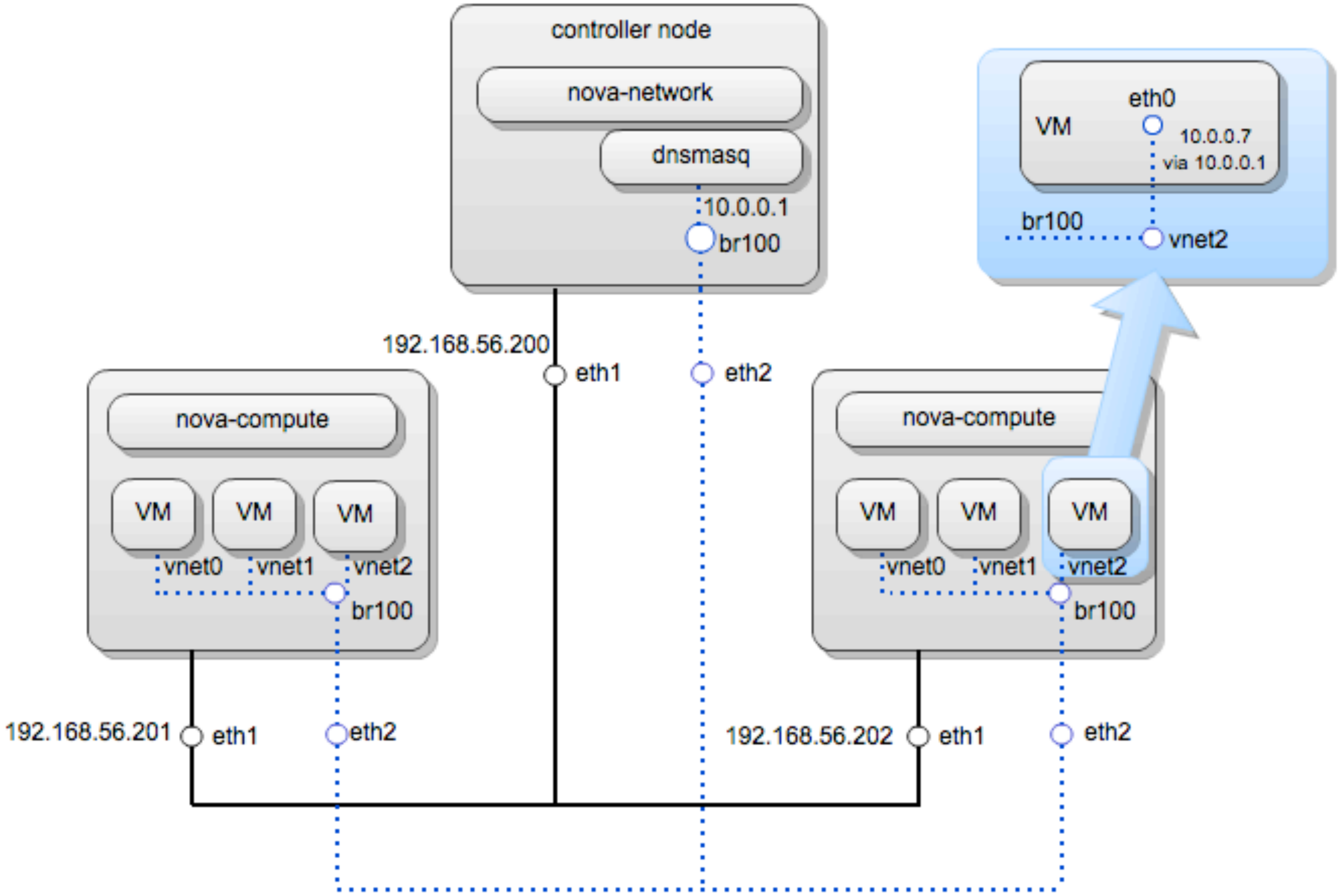

eth0
eth0
dnsmasq
br100
10.0.0.1
dnsmasq
br100
10.0.0.4
vm_1
10.0.0.2
vm_2
10.0.0.3
vm_3
10.0.0.5
vm_4
10.0.0.6



Dnsmasq runtime options and lease config

```
--listen-address=10.0.0.1 #listen on the address of br100

/var/lib/nova/networks/br100.conf
fa:16:3e:11:5e:8e,vm_1,10.0.0.2 #static lease for vm_1
fa:16:9c:10:3e:8f,vm_2,10.0.0.3 #static lease for vm_2
```


controller node
nova-network
dnsmasq
10.0.0.1
br100
192.168.56.200
eth1
eth2
VM
eth0
10.0.0.7
via 10.0.0.1
br100
vnet2
nova-compute
VM
VM
VM
vnet0
vnet1
vnet2
br100
192.168.56.201
eth1
eth2
nova-compute
VM
VM
VM
vnet0
vnet1
vnet2
br100
192.168.56.202
eth1
eth2

eth1 is the management network interface (controlled by --public_interface). The controller has address 192.168.56.200, and we have a default gateway on 192.168.56.101

eth2 is the VM network interface (controlled by --flat_interface). As said, it functions basically as an L2 switch; it doesn’t even have an IP address assigned.
It is bridged with br100 (controlled by --flat_network_bridge).

br100 usually doesn’t have any IP address assigned as well, but on the controller node it has dnsmasq listening on 10.0.0.1 (it is the DHCP server spawned by nova and used by VMs to get an IP address) because it’s the beginning of the flat network range (--fixed_range).

Quantum

- Multitenancy: Isolation, Abstraction, Full control over virtual networks
- Technology-agnostic: API specifies service, vendor provides its implementation
    - List, get, add, update, remove, plug, unplug
- Loose coupling: Standalone service, not specific to Openstack
- L2 networking
- Nova integration
- Plugins
    - Open vSwitch, Cisco UCS, Linux Bridge, NTT-Data Ryu, Nicira NVP

For the Developers

- Paste HTTP server
  - HTTP protocol + networking
    - (http://pythonpaste.org)
- WebOb requests and responses
  - Wrappers for HTTP requests and responses
    - http://www.webob.org
- OpenStack code for Nova, Glance Keystone etc.
    - http://www.openstack.org
- Web Service Gateway Interface (WSGI)
  - http://www.python.org/dev/peps/pep-3333/
  - http://www.wsgi.org
- RESTful Web Services
  - Book by L. Richardson and S. Ruby

Other Campuses

- http://prodigal.nic.uoregon.edu/~hoge/cis607/

Things to Do

- Install on CentOS/Fedora
- Configure network with Ipv6
- Test Folsom / Grizzly
- Test Cinder and Quantum
- Hybrid cloud
- Migrations using NFS server
- Test MooseFS
- EC2 API
- Management using Puppet