

# Développement logiciel pour le Cloud (TLC)

## 6. Infrastructure-as-a-Service

Guillaume Pierre

Université de Rennes 1

Fall 2012

<http://www.globule.org/~gpierre/>



## Table of Contents

- 1 Introduction
- 2 OpenStack
- 3 OpenStack services
- 4 Conclusion

## Table of Contents

- 1 Introduction
- 2 OpenStack
- 3 OpenStack services
- 4 Conclusion

## The landscape of IaaS solutions

| Cloud system        | License     | API                |
|---------------------|-------------|--------------------|
| Amazon Web services | Proprietary | Proprietary        |
| (Windows Azure)     | Proprietary | Proprietary        |
| OpenNebula          | Open-source | Proprietary + OCCl |
| OpenStack           | Open-source | AWS + OCCl         |

- OpenStack was created by Rackspace and NASA
- Many others joined: AMD, Intel, Canonical, SUSE Linux, Red Hat, Cisco, Dell, HP, IBM, Yahoo etc.

# Table of Contents

- 1 Introduction
- 2 OpenStack
- 3 OpenStack services
- 4 Conclusion

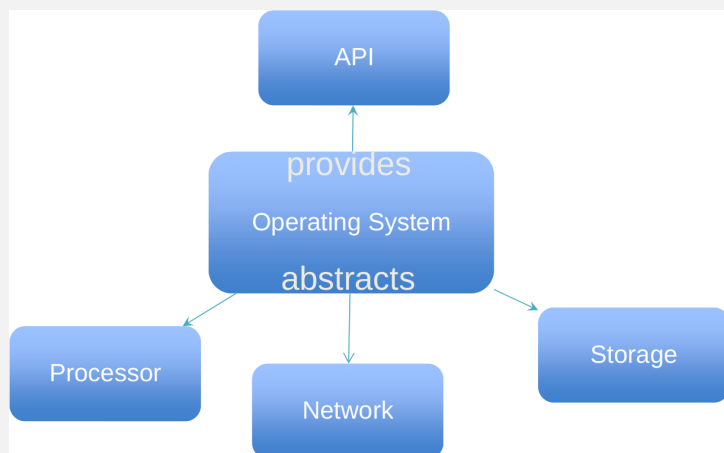
# OpenStack

- “OpenStack is a community of open source developers, participating organizations and users building and running the open source cloud operating system.”
- “OpenStack is a Cloud Operating System”
- “OpenStack is a Cloud Orchestration layer”



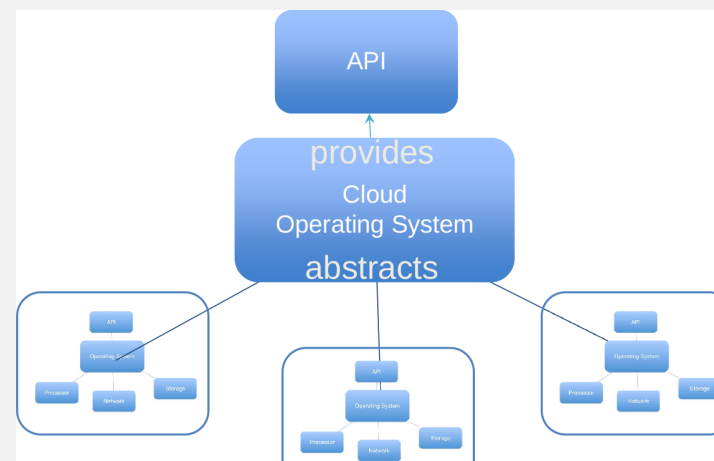
# Traditional operating system

- Provides APIs
- Abstracts access to hardware resources
- Controls access to these resources



# Cloud operating system

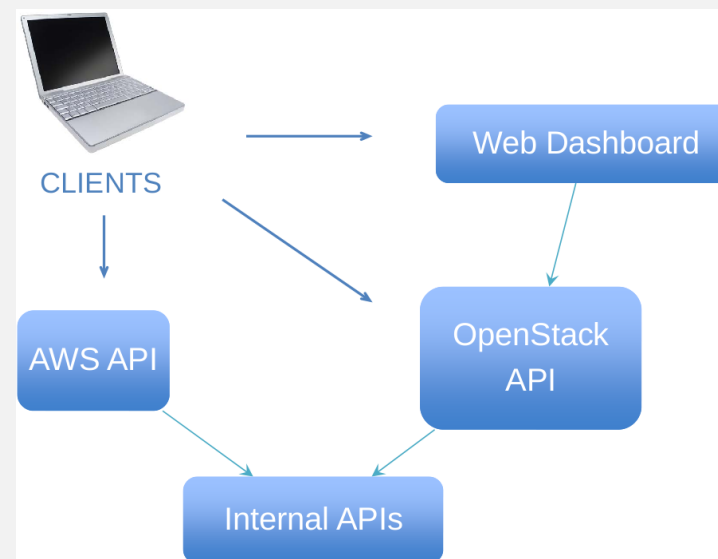
- Provides APIs
- Abstracts access to hardware **virtualized** resources
- Controls access to these resources



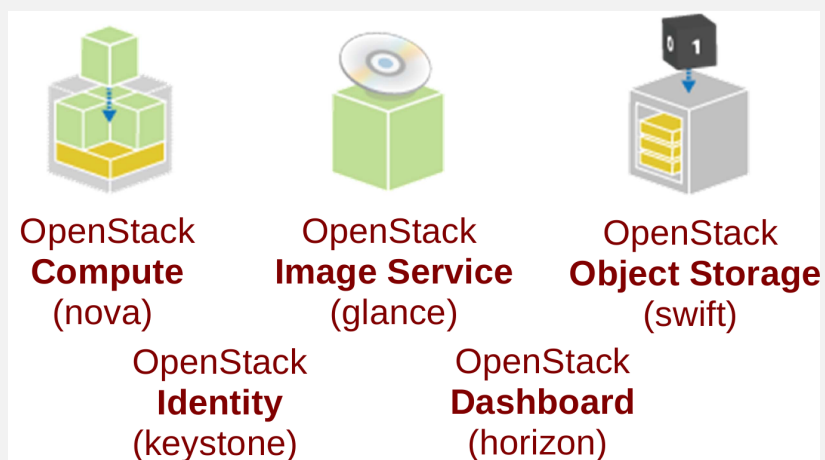
## What's the difference?

- The **type of resource** controlled
  - ▶ Hardware components vs. whole systems
- The **scale** of resources controlled
  - ▶ One machine vs. a data center
- The **location** of the resources
  - ▶ Local vs. remote

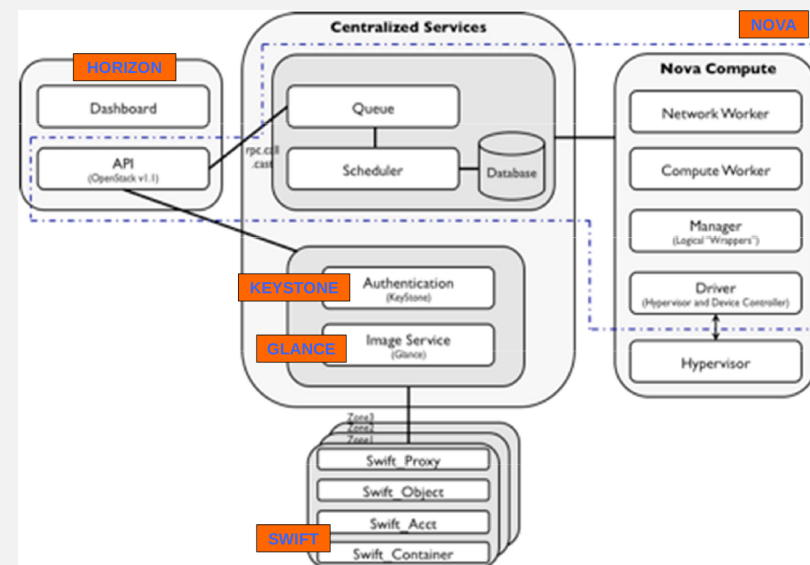
## Interacting with OpenStack



## Main services

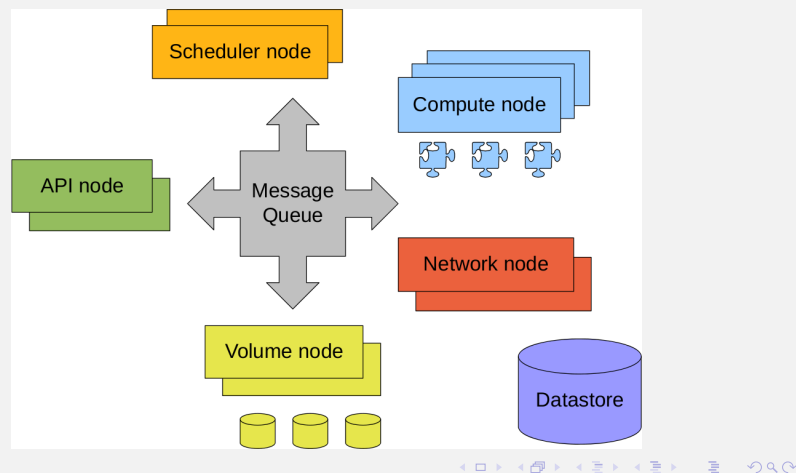


## Mapping services onto the underlying system architecture



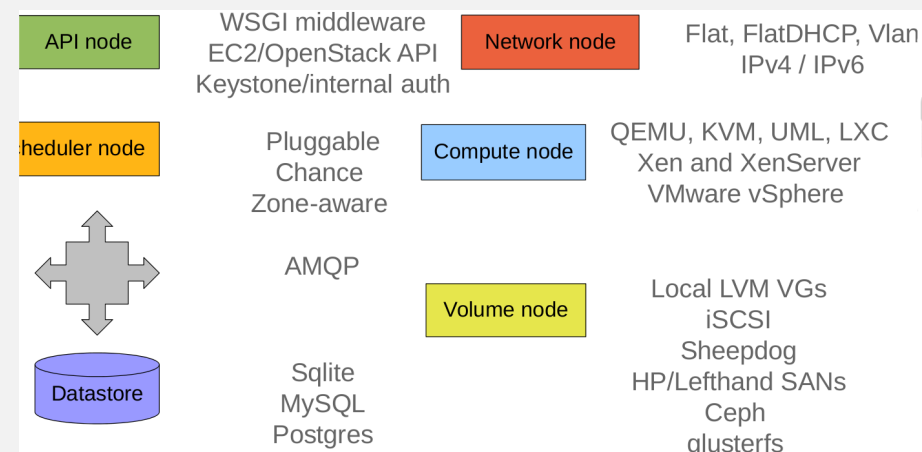
## OpenStack Nova == the computation service

- Nova is the heart of OpenStack
- It manages all **resources, networking, authorization, and scalability needs** for the cloud



## Challenge: flexibility

- Different customers want to deploy OpenStack over different types of systems



## Advanced features

- Cloud federation
  - ▶ Merge multiple clusters into a single cloud
  - ▶ Possibly across multiple administration domains
- Cloud bursting
  - ▶ When one cloud approaches full capacity it starts buying resources from another cloud
  - ▶ And extends seamlessly to these new resources
  - ▶ Keep things transparent to the clients

## Table of Contents

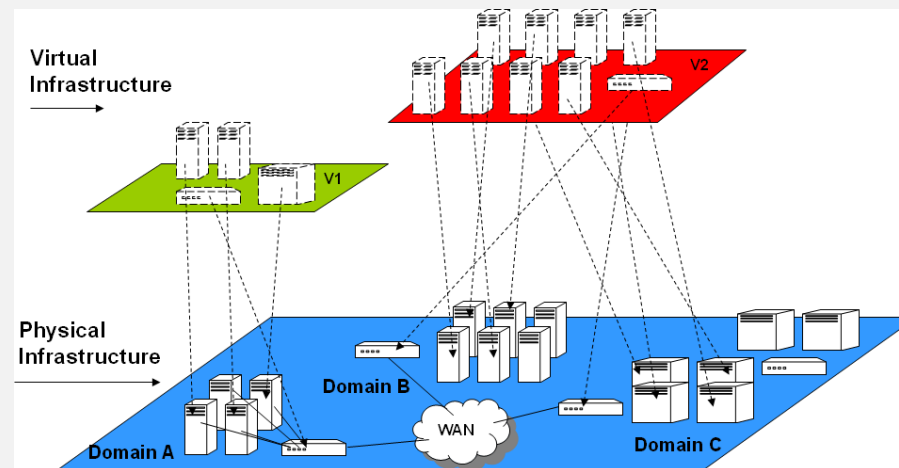
- 1 Introduction
- 2 OpenStack
- 3 OpenStack services
- 4 Conclusion

## Network virtualization

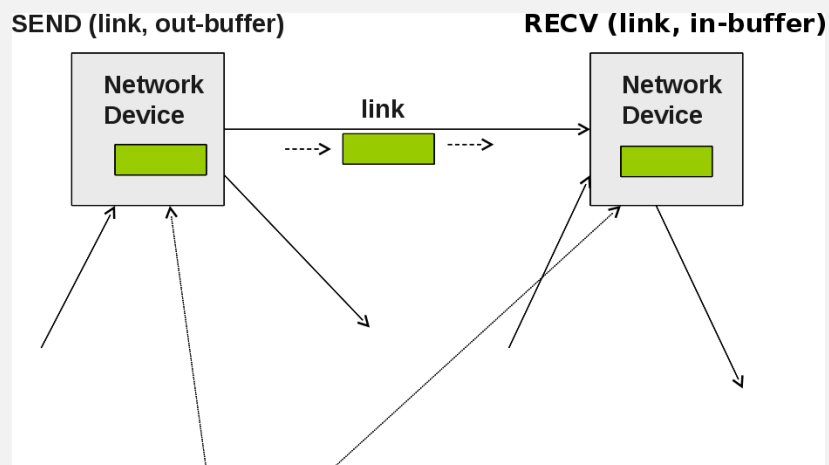
- Goal: **hide the complexity of the real networks** behind simple and secure abstractions
- In the physical world:
  - ▶ Many applications belonging to many customers
  - ▶ Machines are located in different clusters or data centers
  - ▶ Each data center has its own firewall/NAT/address space/policies
- In the virtualized world:
  - ▶ Many virtual networks (e.g., one per user or project)
  - ▶ Each isolated and independently configured (address allocation, protocols used etc.)
  - ▶ Multiplexing physical network resources



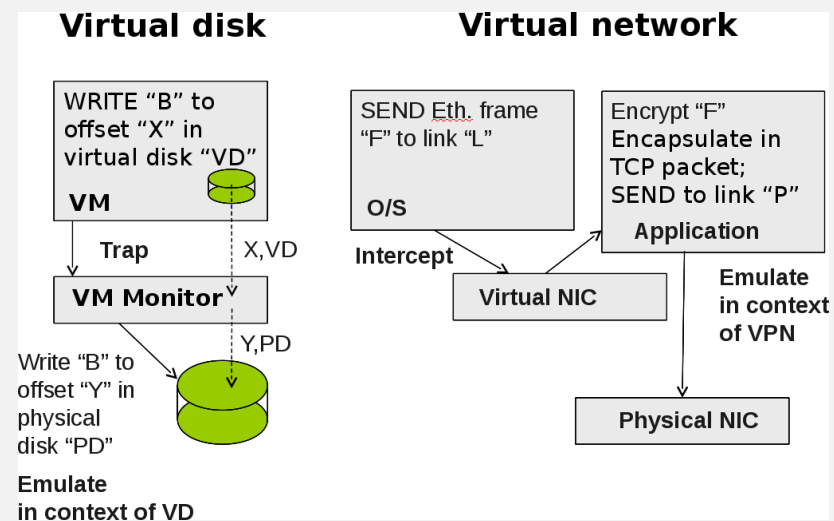
## Network virtualization



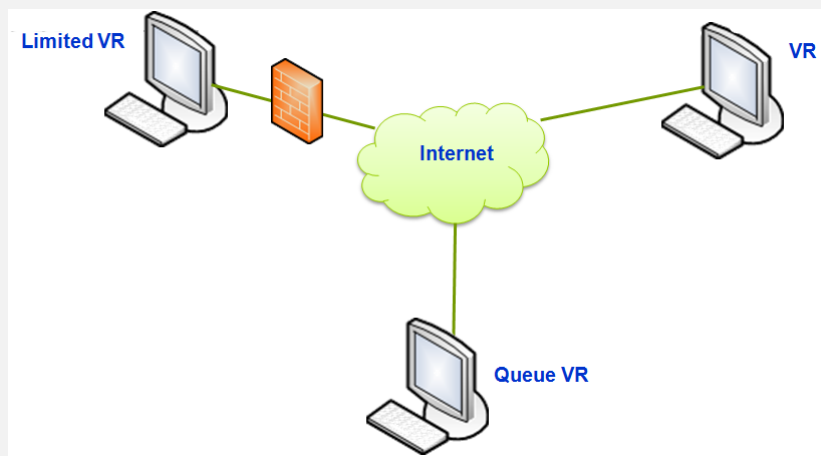
## Network abstractions



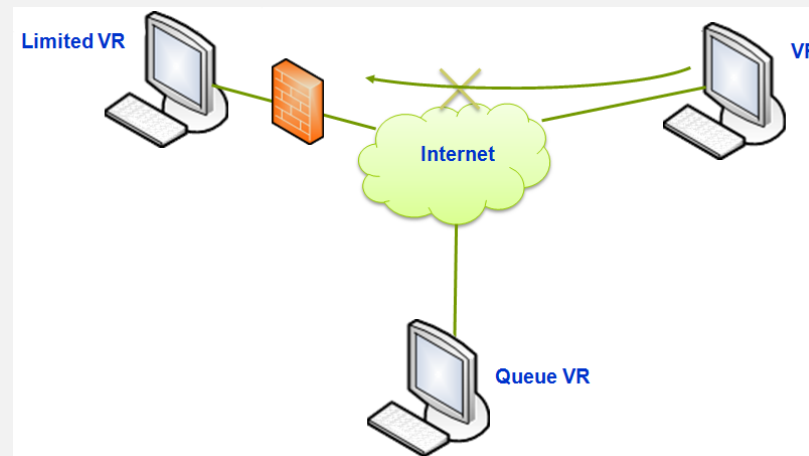
## Virtualizing network primitives



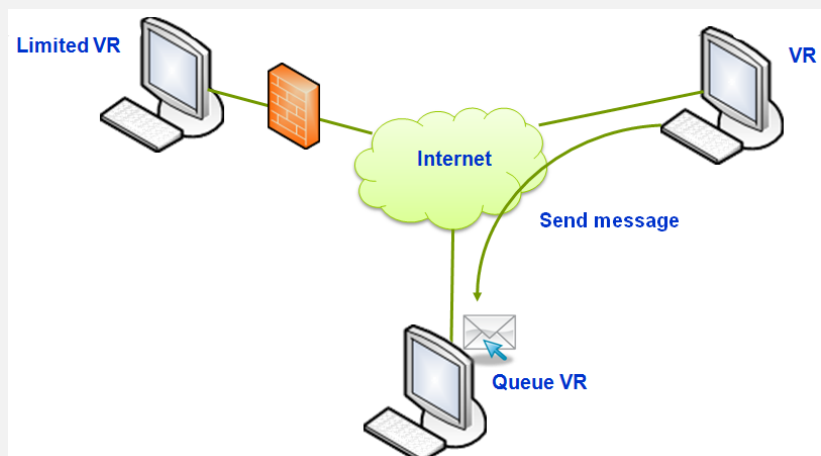
## Example: firewall traversal



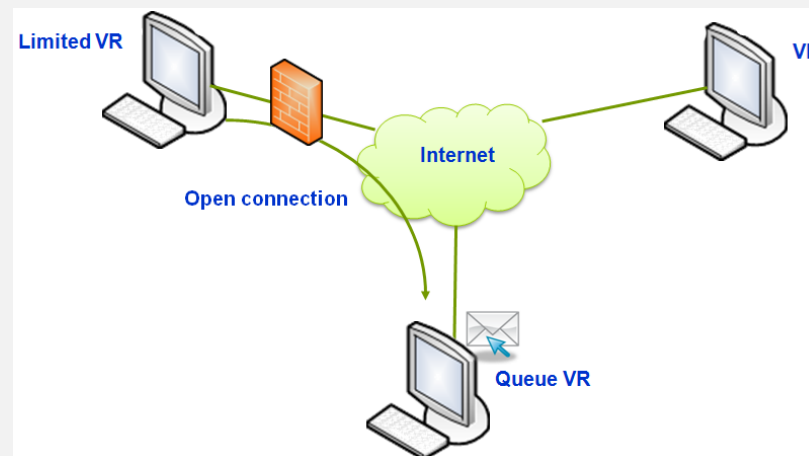
## Example: firewall traversal



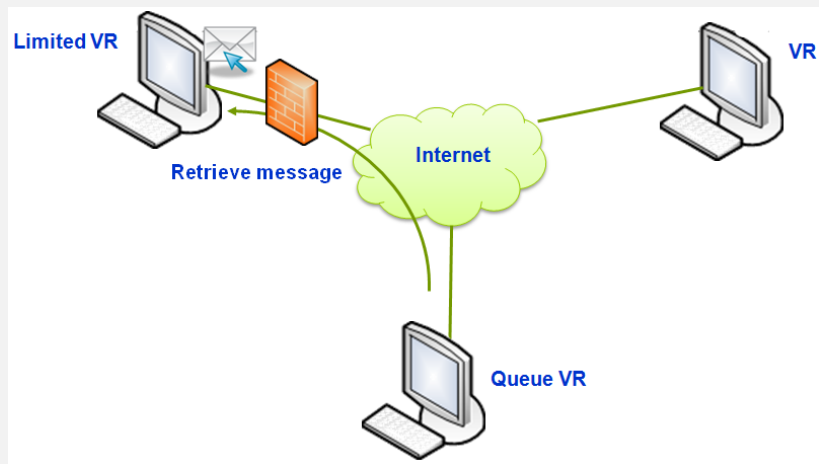
## Example: firewall traversal



## Example: firewall traversal



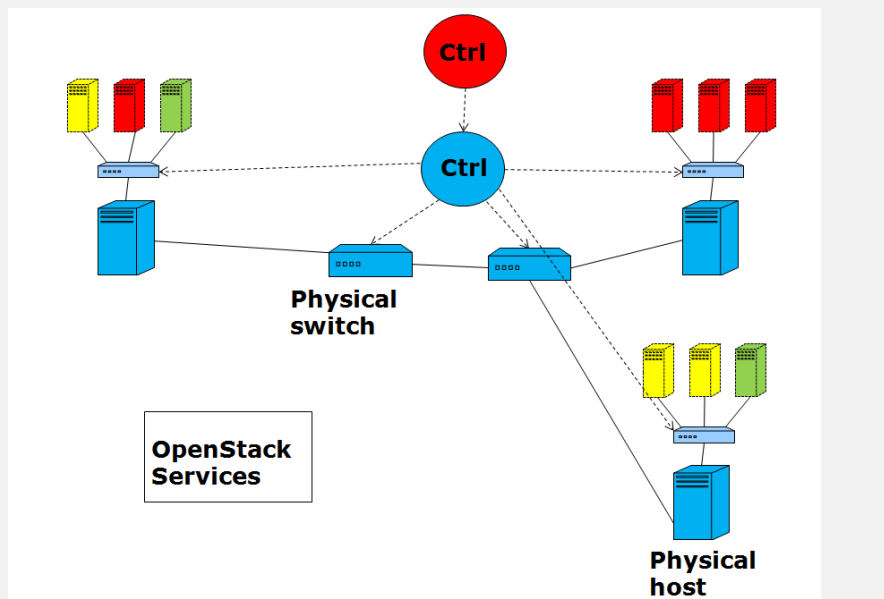
## Example: firewall traversal



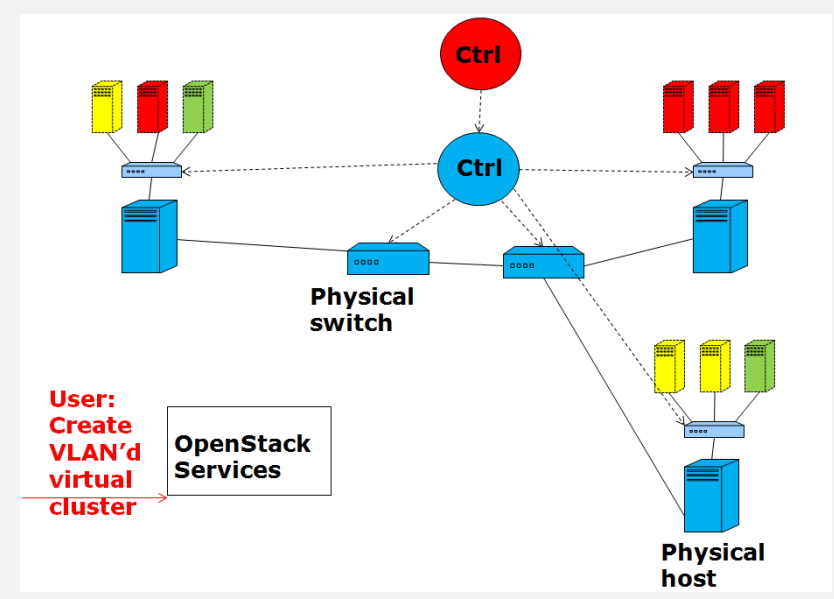
## OpenStack Quantum

- Multiple, virtual, isolated networks per tenant
- Create ports on networks and attach VMs
- Control your own private IP address space
- Access via CLI or GUI (horizon)
- Support different underlying technologies (VLANs, L2/L3 tunnels etc)
- Extend through plugins

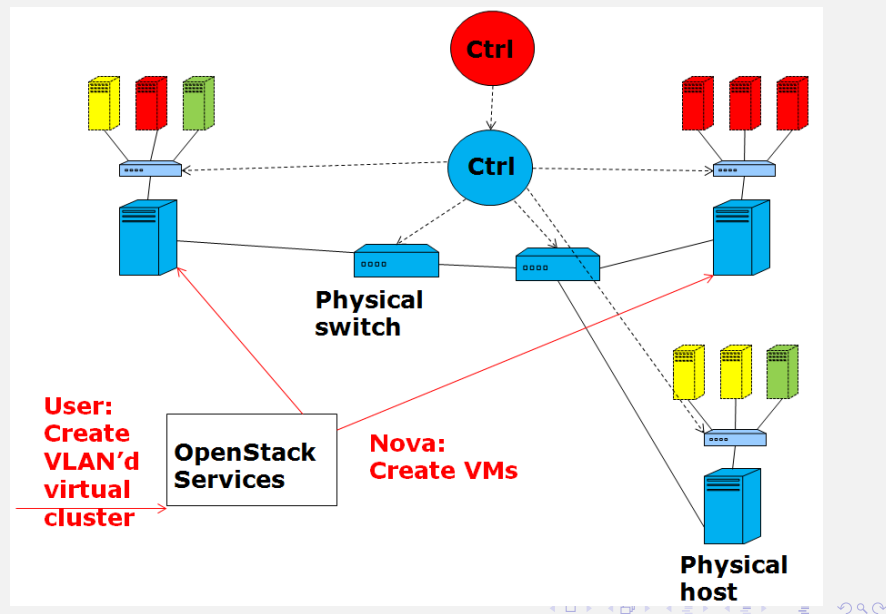
## Example: OpenStack Nova + Quantum



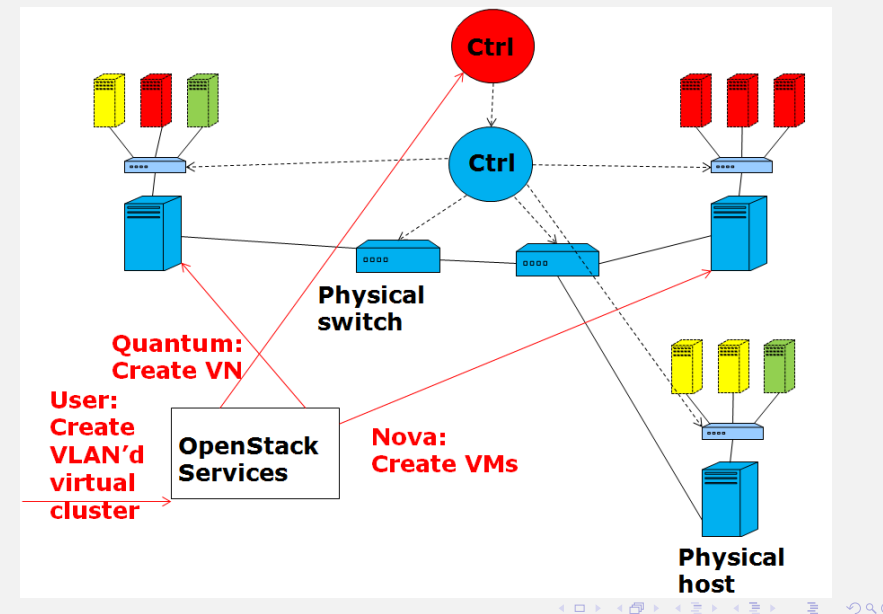
## Example: OpenStack Nova + Quantum



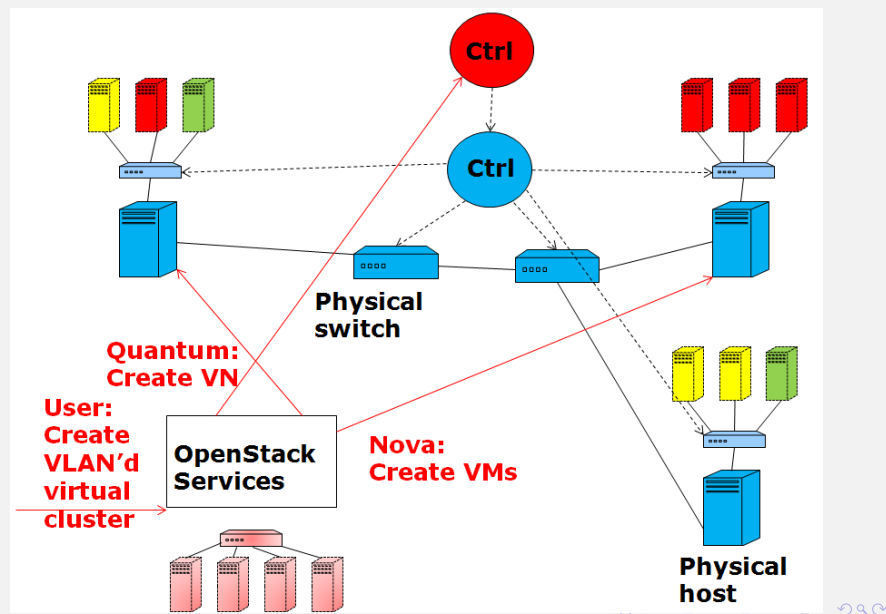
## Example: OpenStack Nova + Quantum



## Example: OpenStack Nova + Quantum



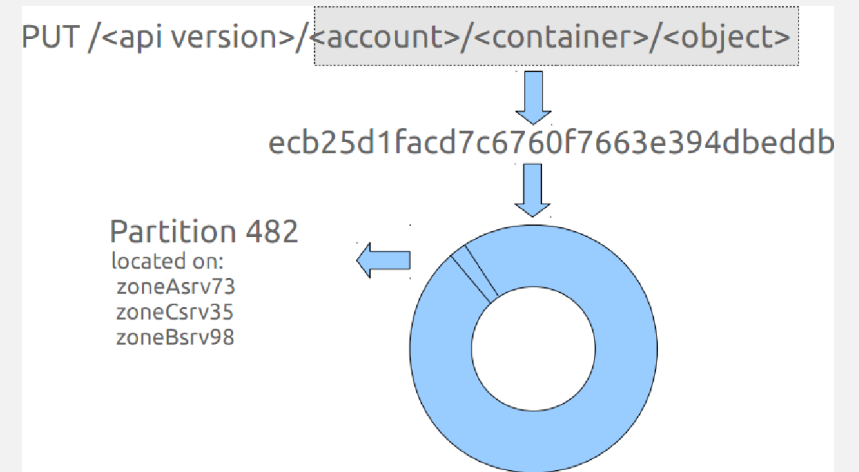
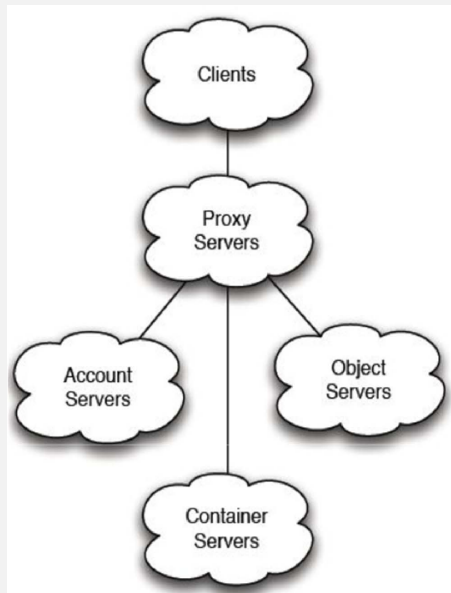
## Example: OpenStack Nova + Quantum



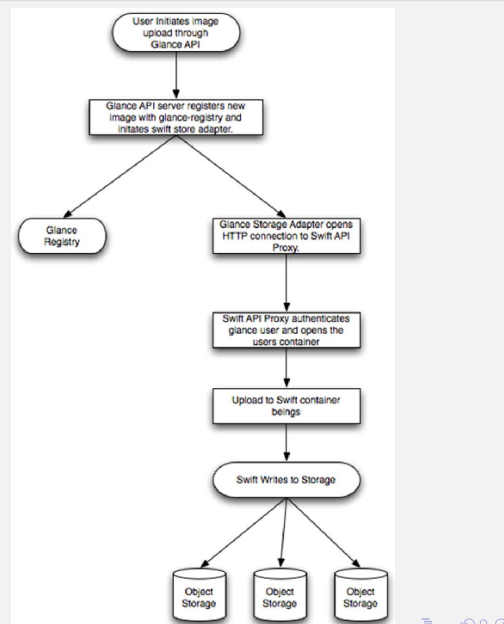
## OpenStack Swift == data storage service

- Goal: **redundant, scalable object storage** using clusters of commodity servers
  - ▶ Scale: terabytes to petabytes of accessible data
- Data storage services are **not a filesystem**
  - ▶ Rather: **long term storage** for permanent, static data
  - ▶ Data can be updated if necessary but clouds assume this is infrequent
- Usually: a RESTful API
  - ▶ GET /api-version/username/container/object
  - ▶ PUT /api-version/username/container/object





- Goal: **discovering, registering, and retrieving virtual machine images**
- Glance is implemented as a layer on top of Swift



- Goal: **single source of authentication and authorization**
  - ▶ Same account and credentials for starting a vm and accessing a container in object storage
  - ▶ Means of expressing API endpoints
- Authorization is usually handled using **capabilities**
  - ▶ Authenticate with the Keystone service
  - ▶ Receive a token back
    - ★ The token encapsulates an identity and a scope
    - ★ Signed by the keystone service
  - ▶ Show the token for all requests to other services
    - ★ `curl -H "X-Auth-Token:999888777666" http://my.keystone.server:35357`

- 1 Introduction
- 2 OpenStack
- 3 OpenStack services
- 4 Conclusion**

- Infrastructure-as-a-Service is **more than a simple interface to virtualization**
  - ▶ Machine virtualization
  - ▶ Network virtualization
  - ▶ Data storage services
  - ▶ Disk image repositories
  - ▶ Identity management
  - ▶ (Load balancing)
  - ▶ (DNS)
  - ▶ (Monitoring)
  - ▶ etc.
- A good IaaS system should have:
  - ▶ A rich set of infrastructure services
  - ▶ Good APIs + user-friendly GUI
  - ▶ Strong integration between these services
- IaaS == **the operating system of a cloud platform**