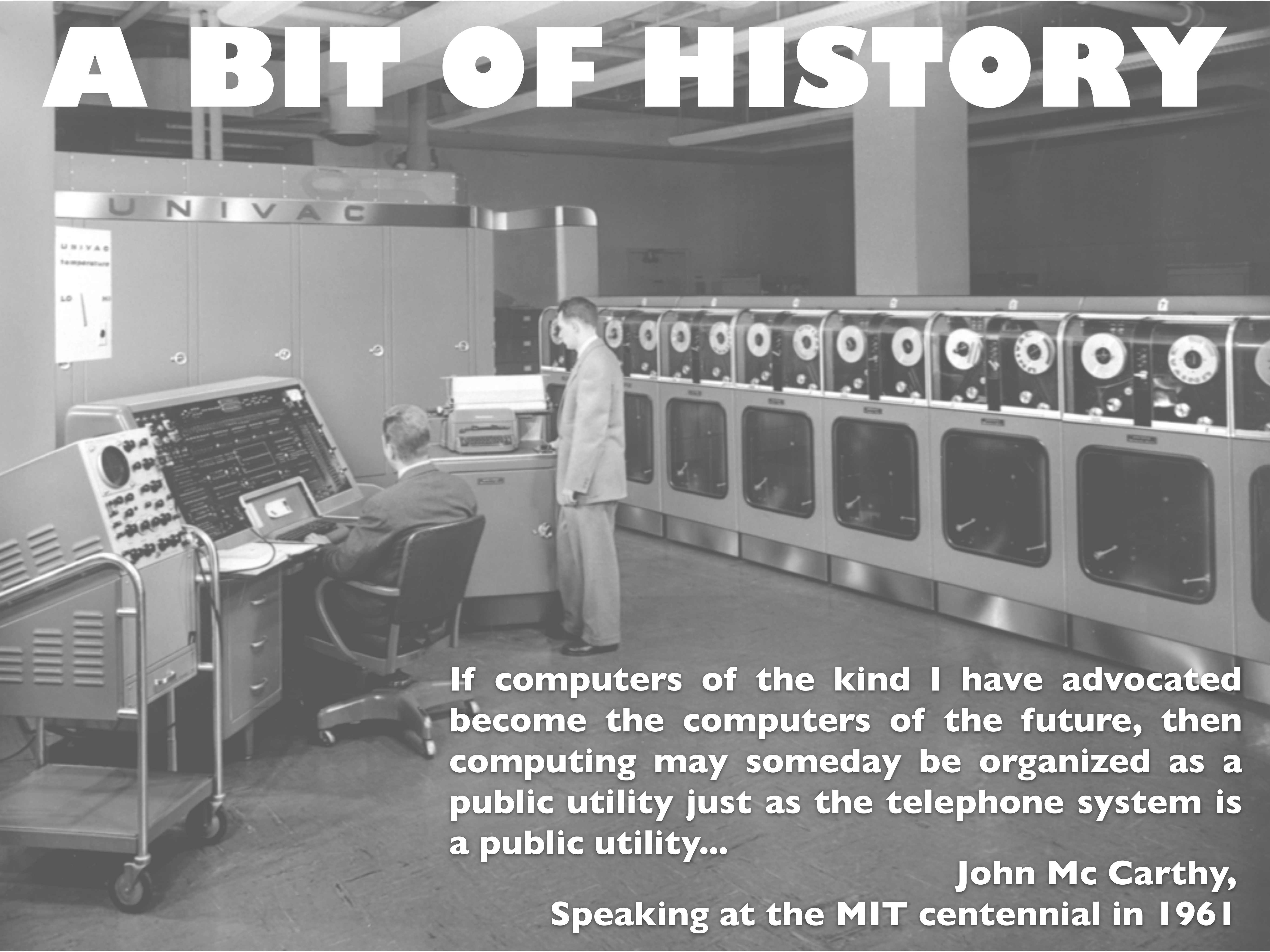
Infrastructure-as-a-Service
OpenStack

Adrien Lebre
STACK Research Group

A BIT OF HISTORY



If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility...

John Mc Carthy,
Speaking at the MIT centennial in 1961

Looking back…

- xxx Computing
  Meta / Cluster / Grid / Desktop / "Hive" / Cloud / Sky …

  $\Rightarrow$ xxx as Utility Computing

- A common objective: provide computing resources (both hardware and software) in a flexible, transparent, secure, reliable, ... way

- Challenges

  Software/Hardware heterogeneity

  Security (Isolation between applications, ...)

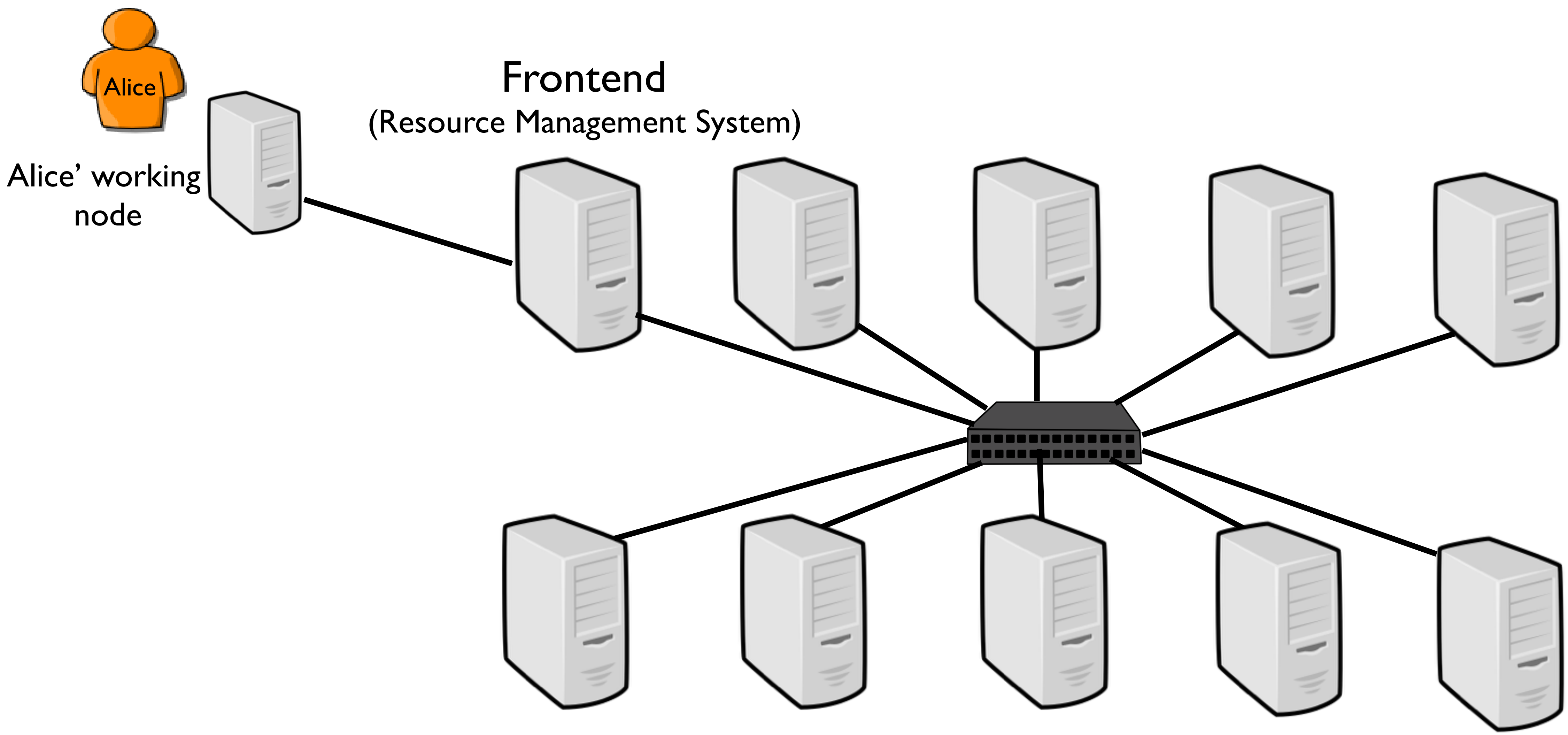
  Reliability / Resiliency

  Data Sharing

  Performance guarantees...

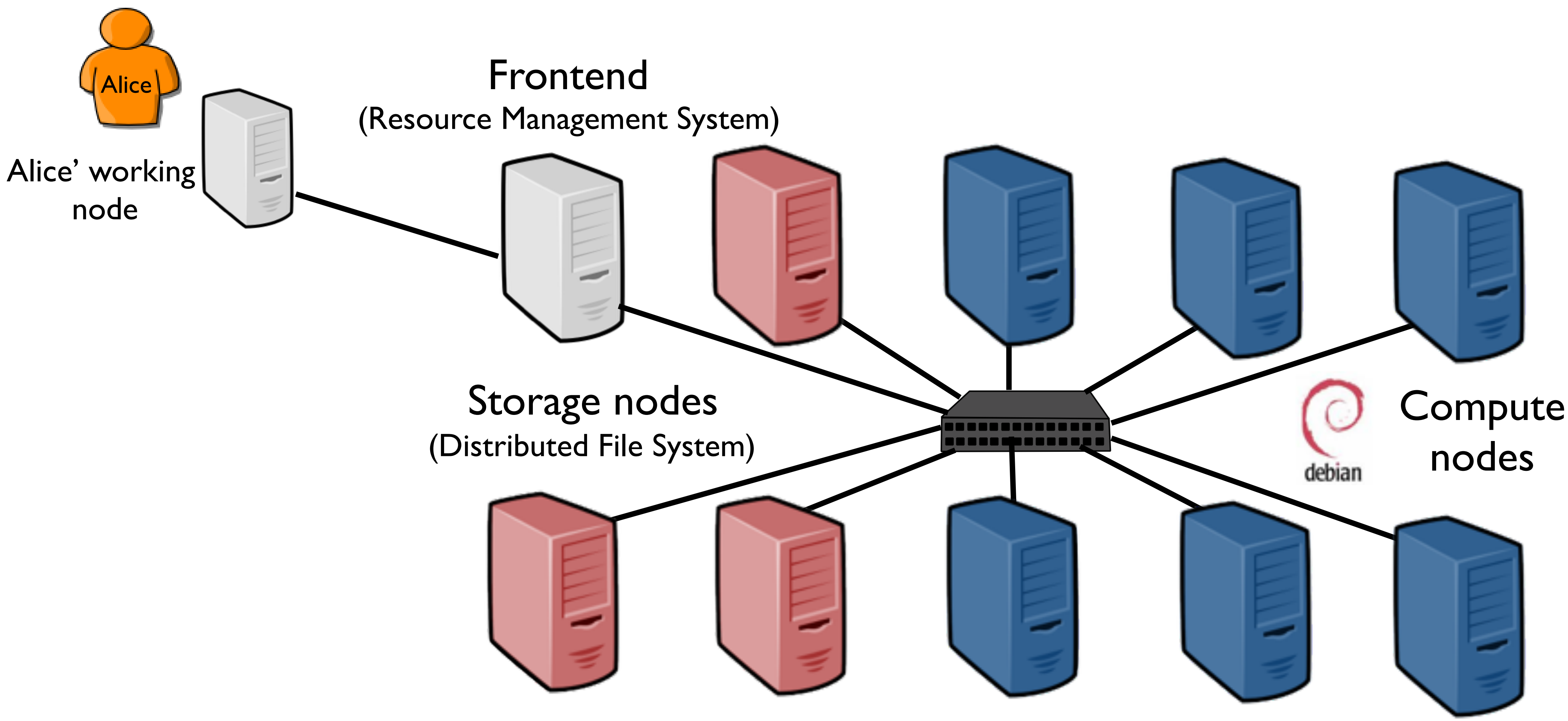
Looking back…

- Network of Workstations 1990 / 20**xx**



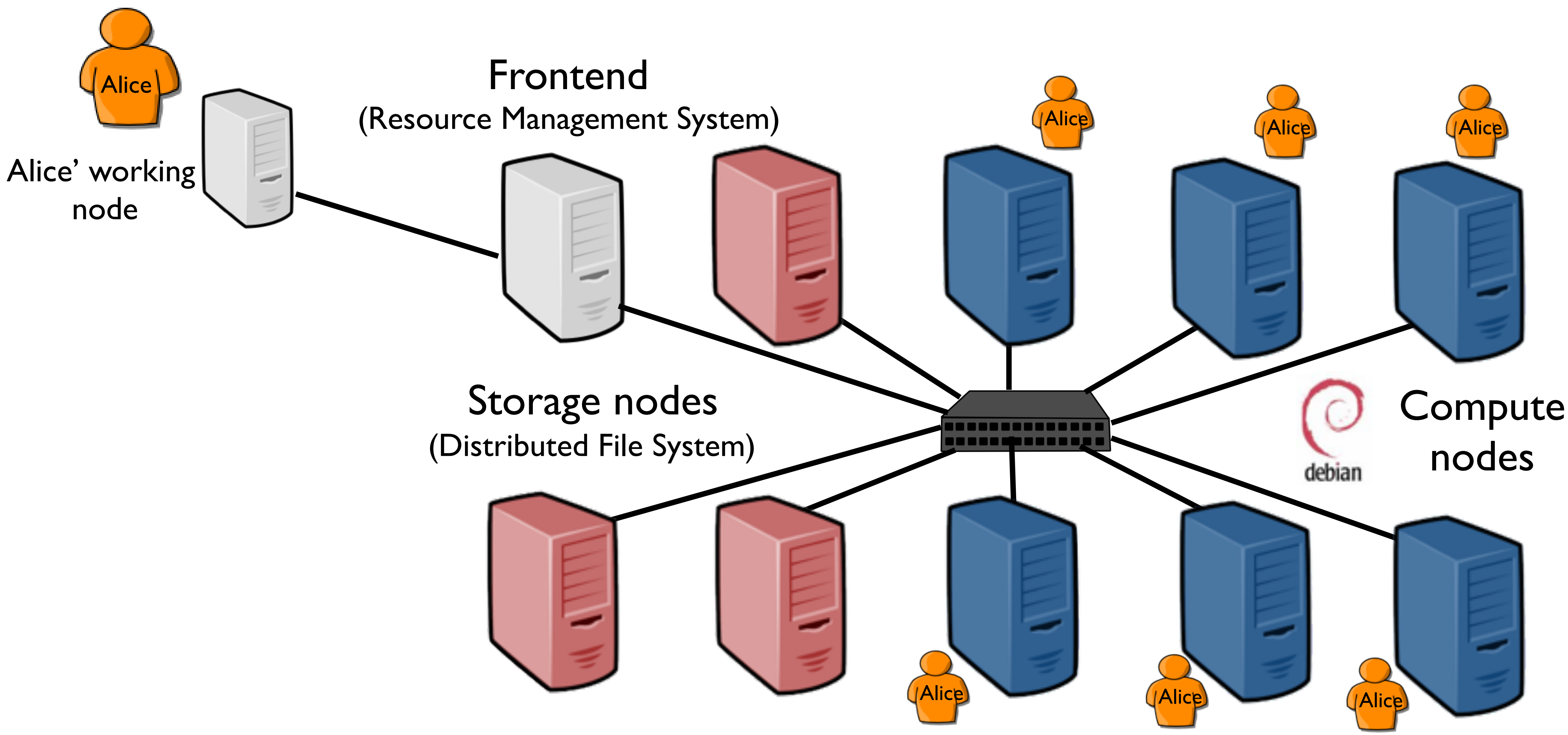
Looking back…

- Network of Workstations 1990 / 20xx



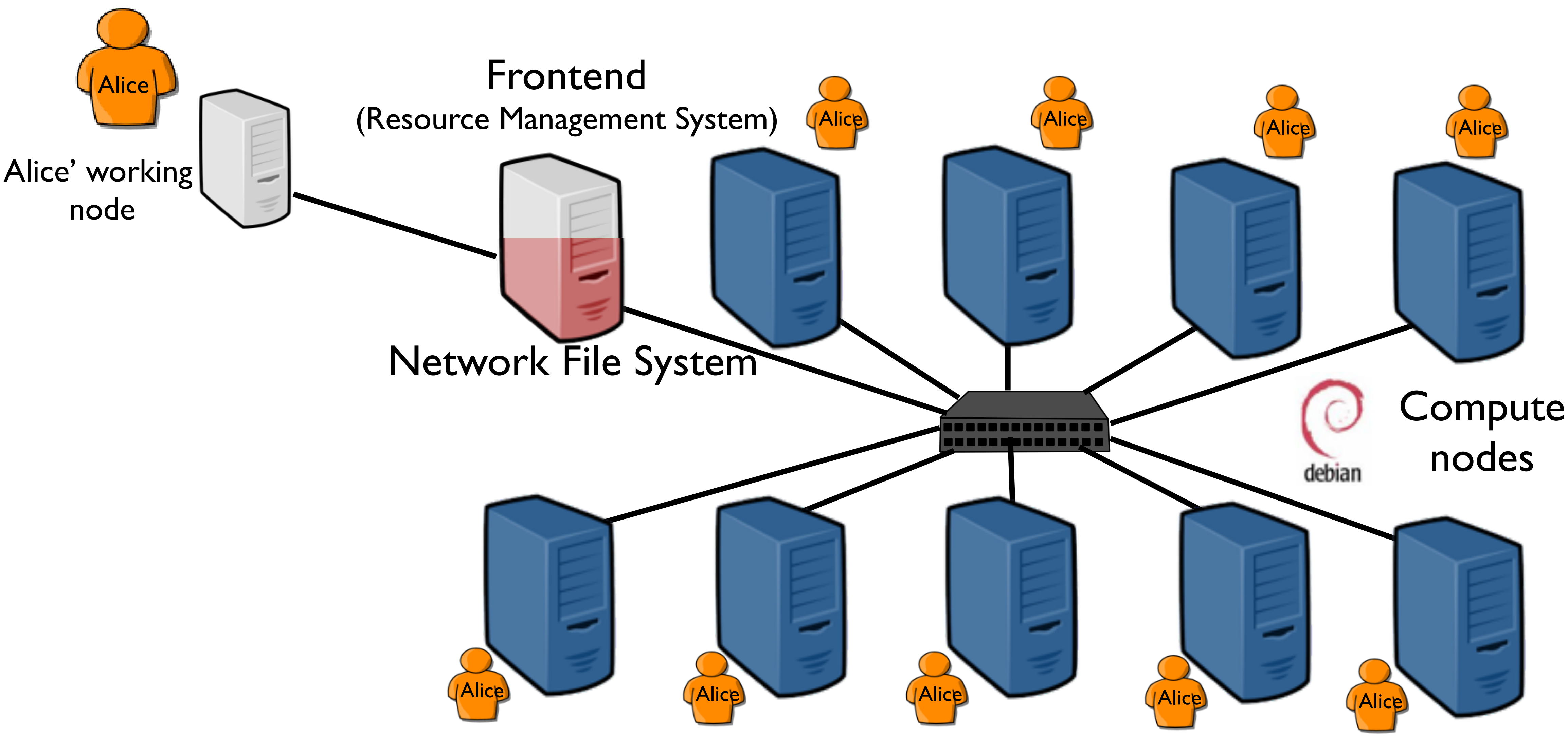
Looking back…

- Network of Workstations 1990 / 20xx



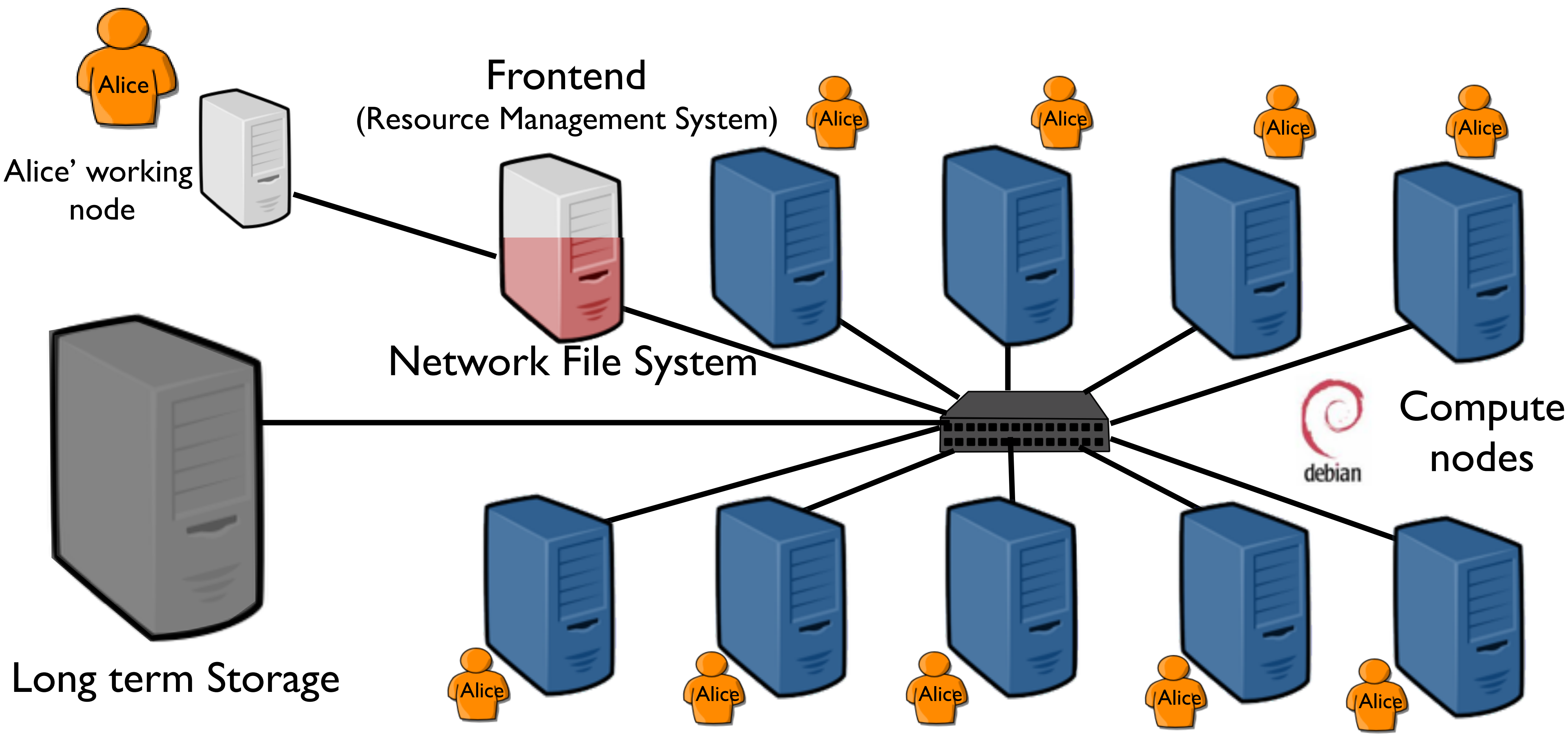
Looking back…

- Network of Workstations 1990 / 20xx



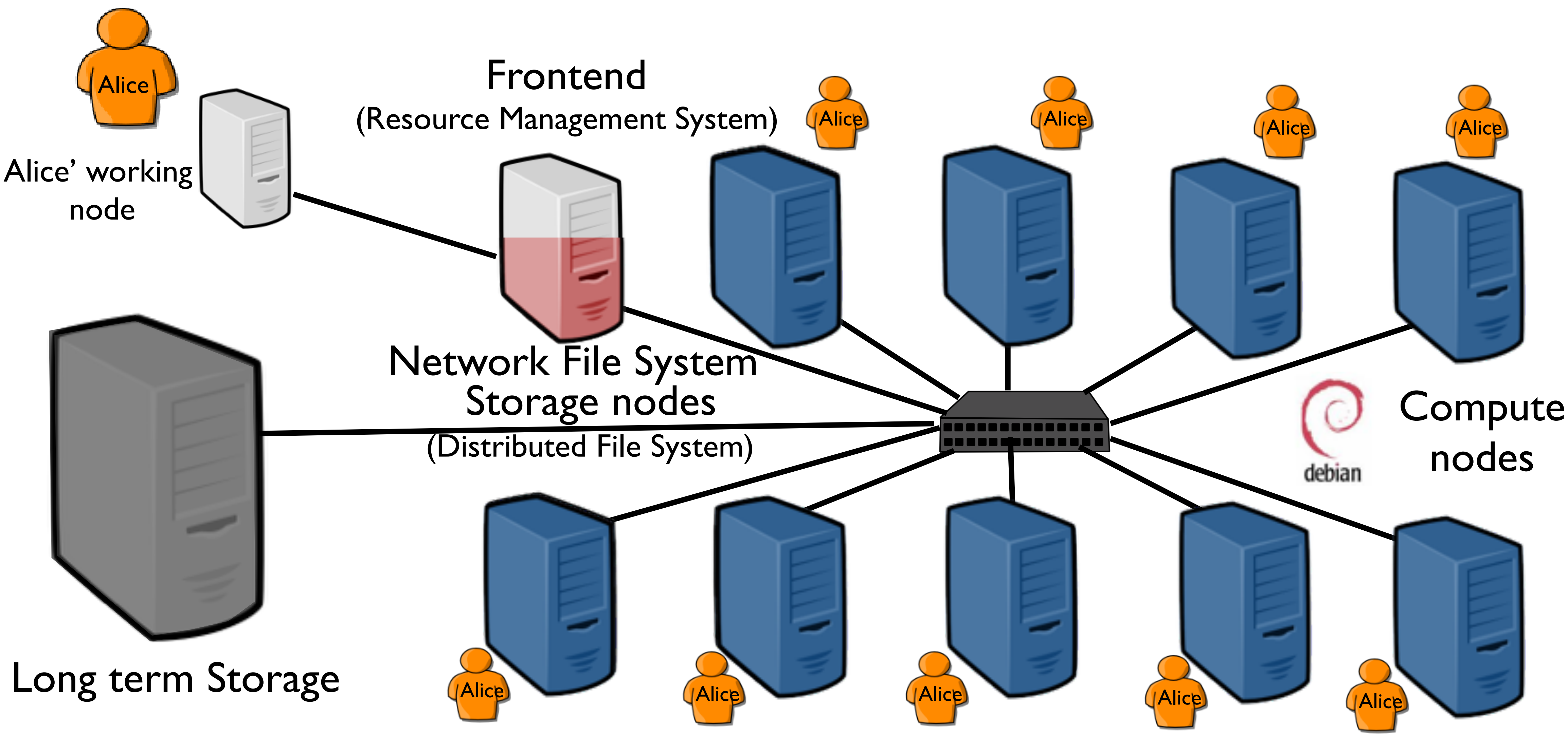
Looking back…

- Network of Workstations 1990 / 20**xx**



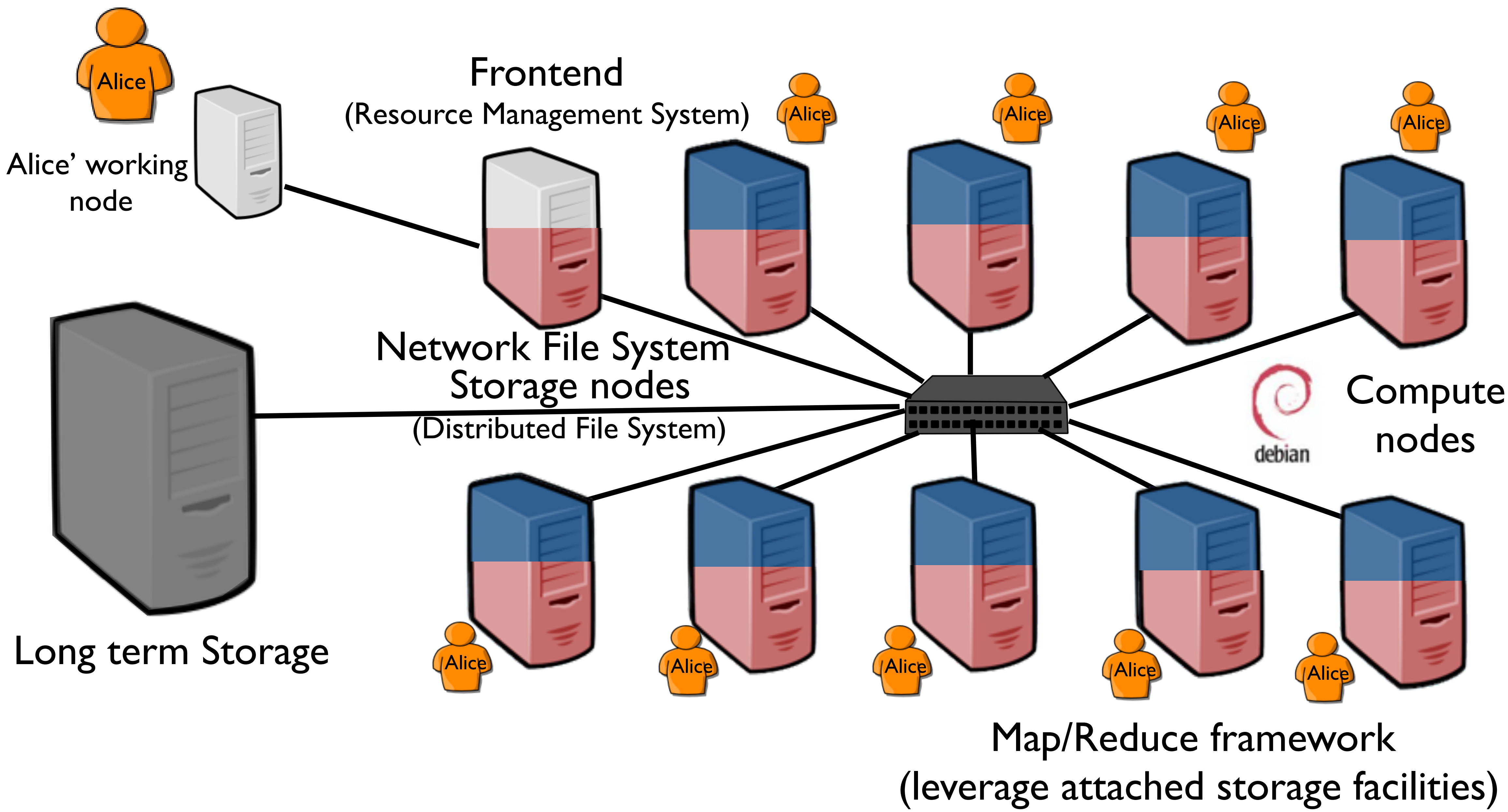
Looking back…

- Network of Workstations 1990 / 20**xx**



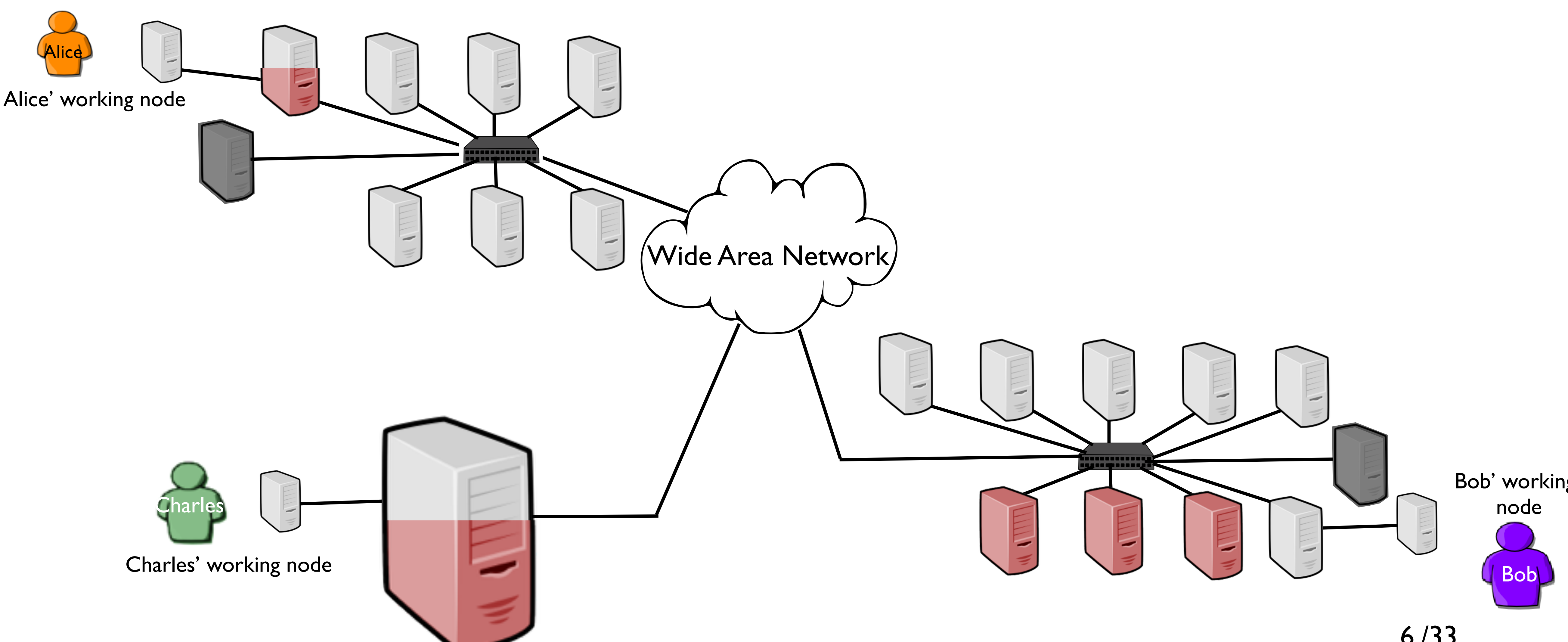
Looking back…

- Network of Workstations 1990 / 20**xx**



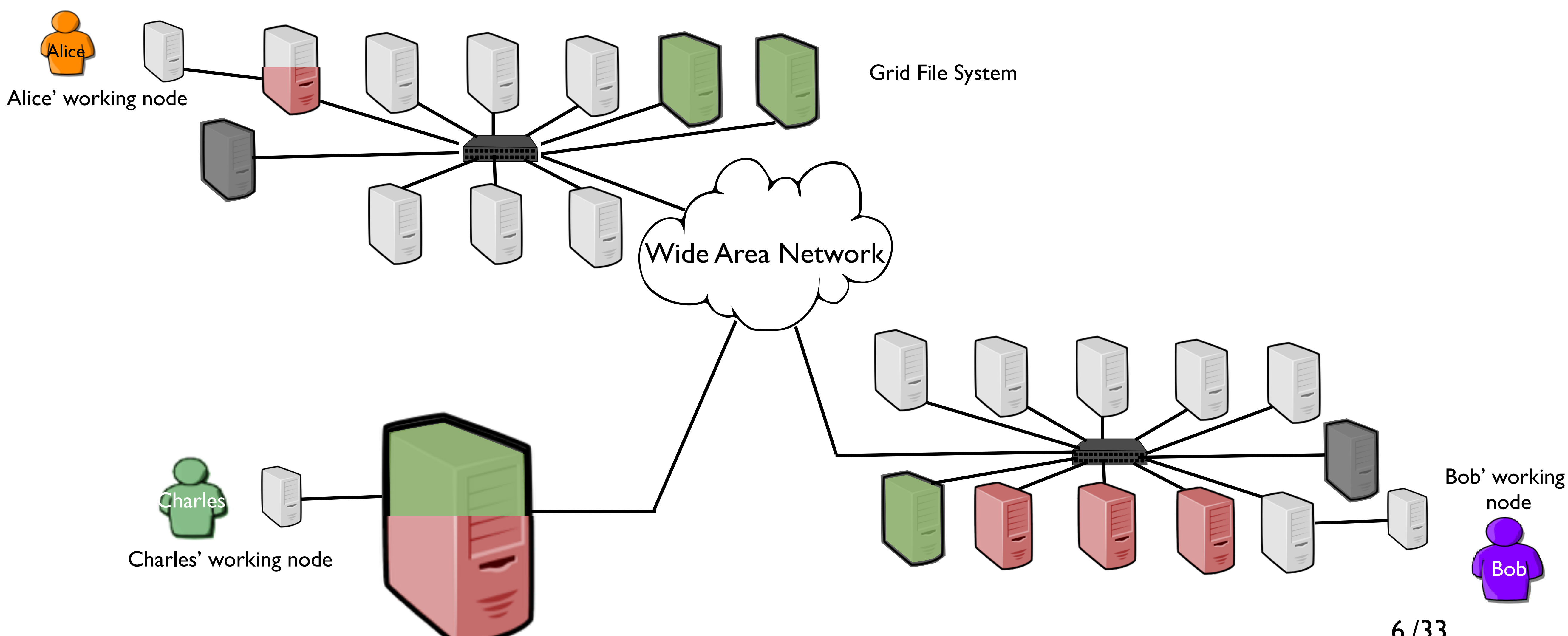
Looking back…

- Network of Workstations 1990 / 20**xx**
- Desktop 1998 / 201**x**
- Grid 1998 / 201**x**



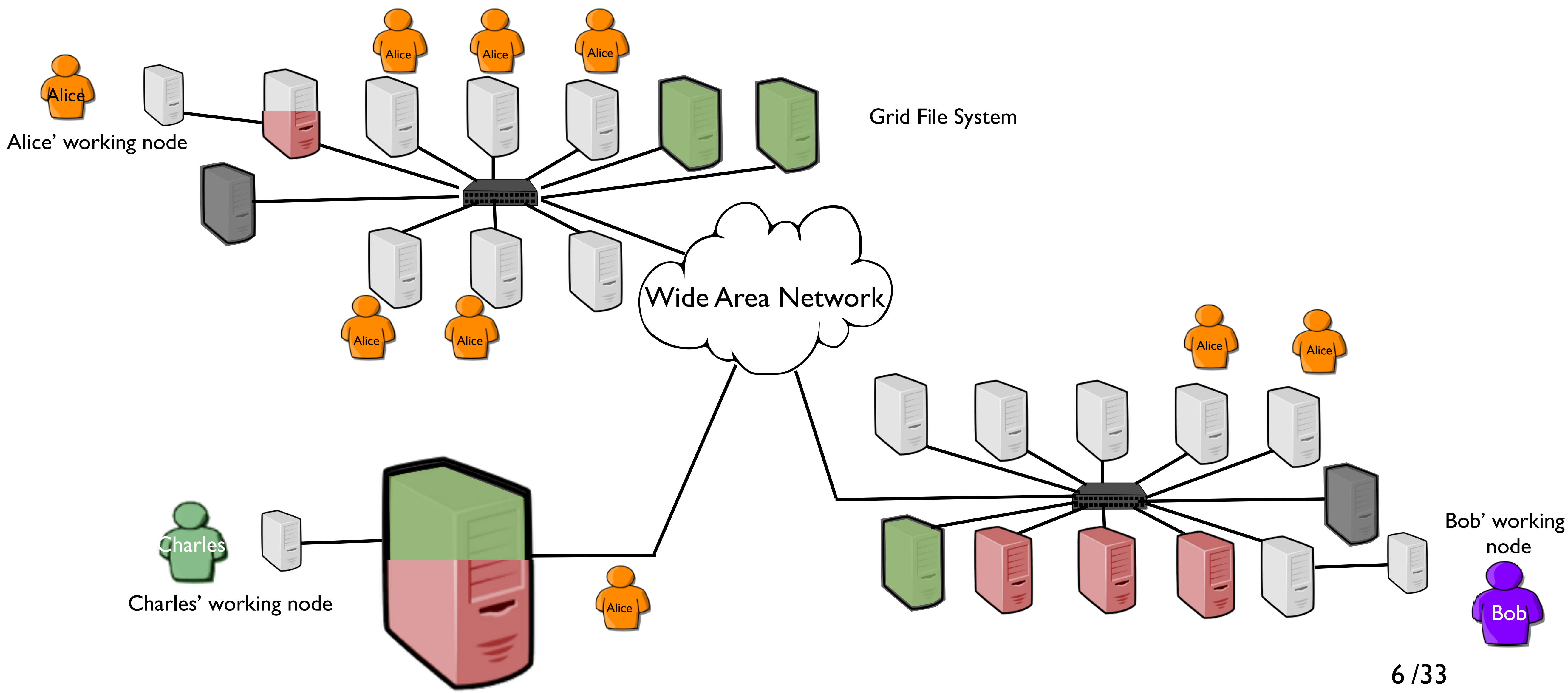
Looking back…

- Network of Workstations 1990 / 20xx
- Desktop 1998 / 201x
- Grid 1998 / 201x



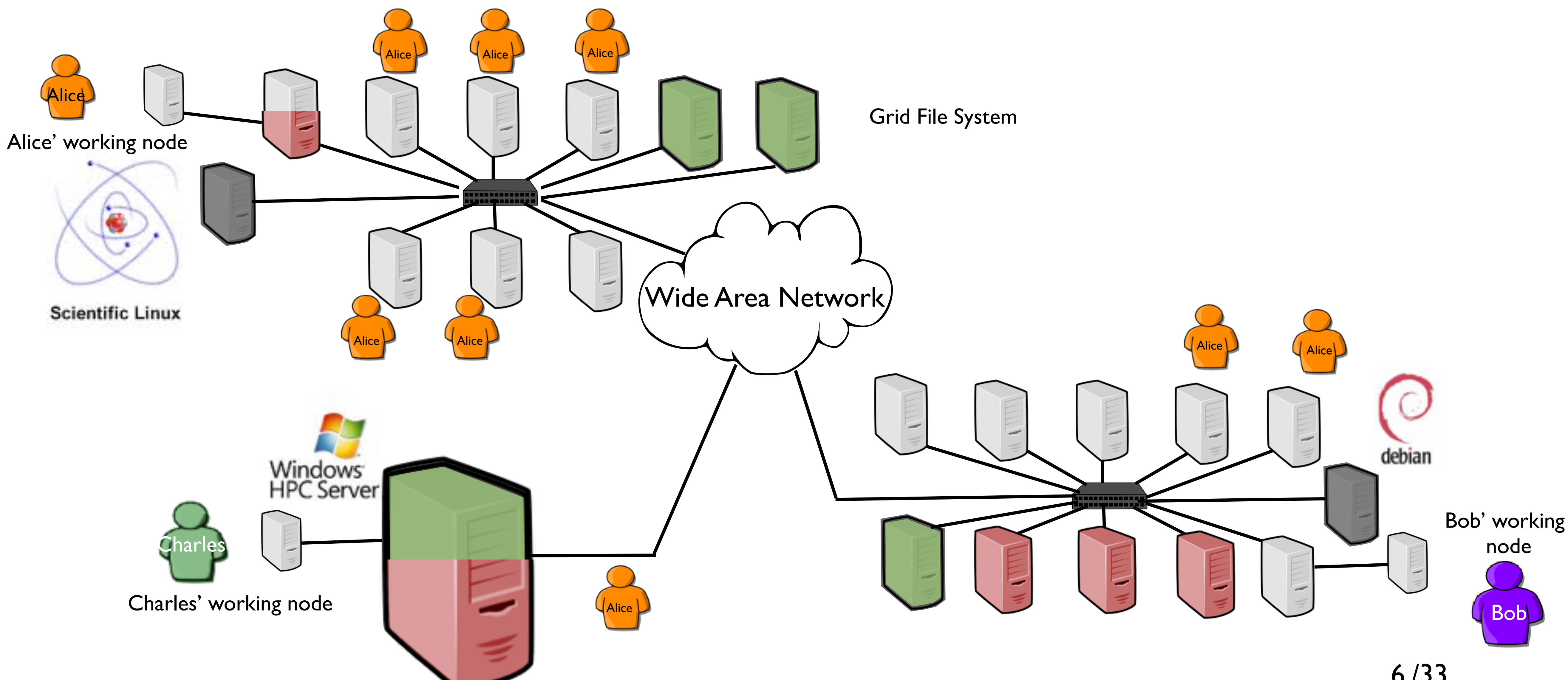
Looking back…

- Network of Workstations 1990 / 20xx
- Desktop 1998 / 201x
- Grid 1998 / 201x



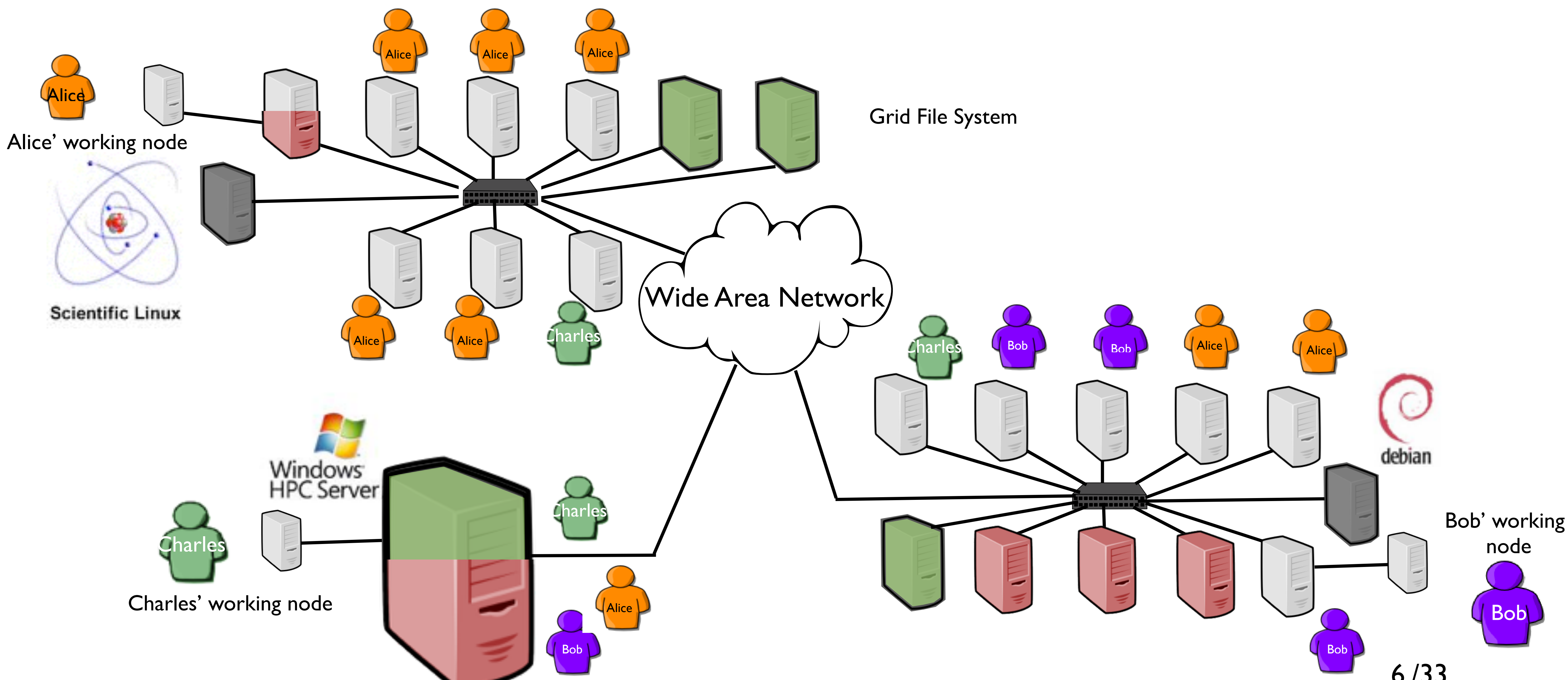
Looking back…

- Network of Workstations 1990 / 20xx
- Desktop 1998 / 201x
- Grid 1998 / 201x



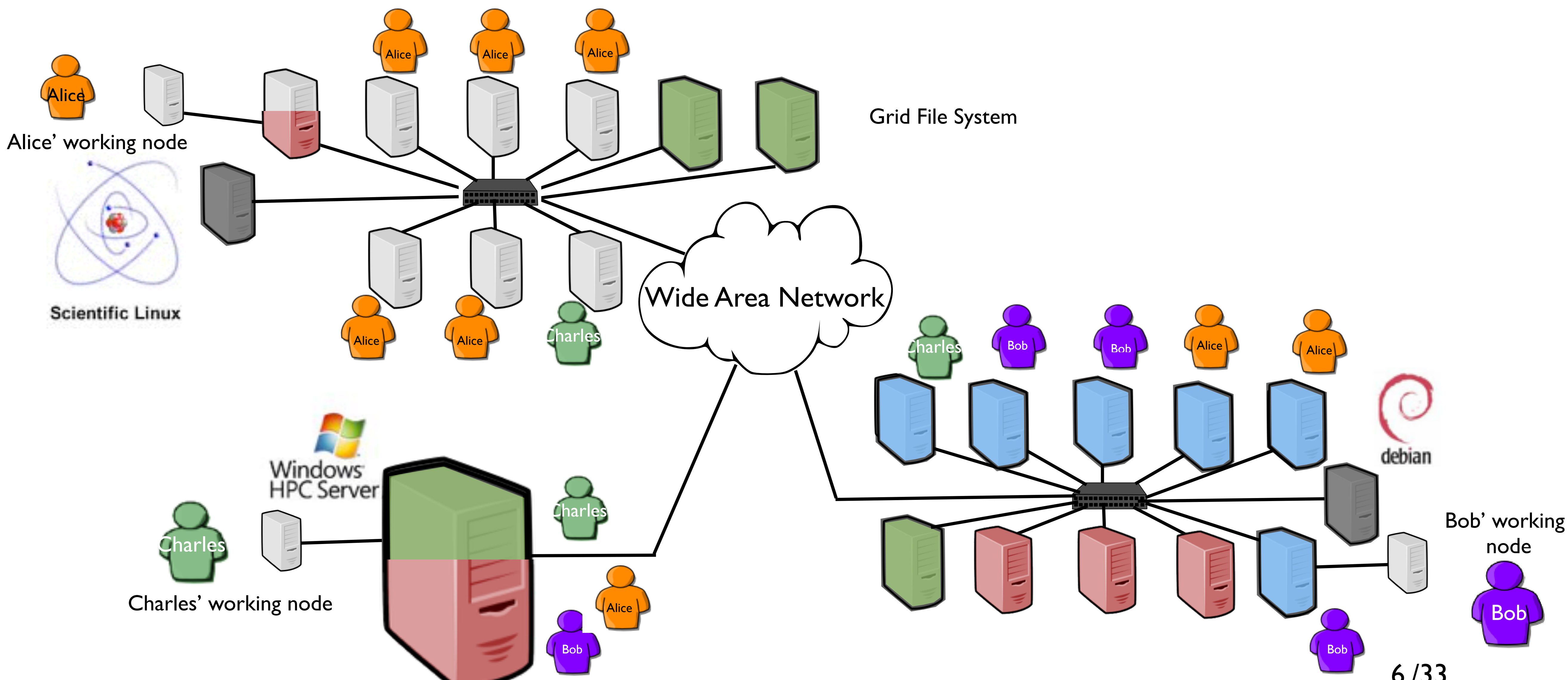
Looking back…

- Network of Workstations 1990 / 20xx
- Desktop 1998 / 201x
- Grid 1998 / 201x



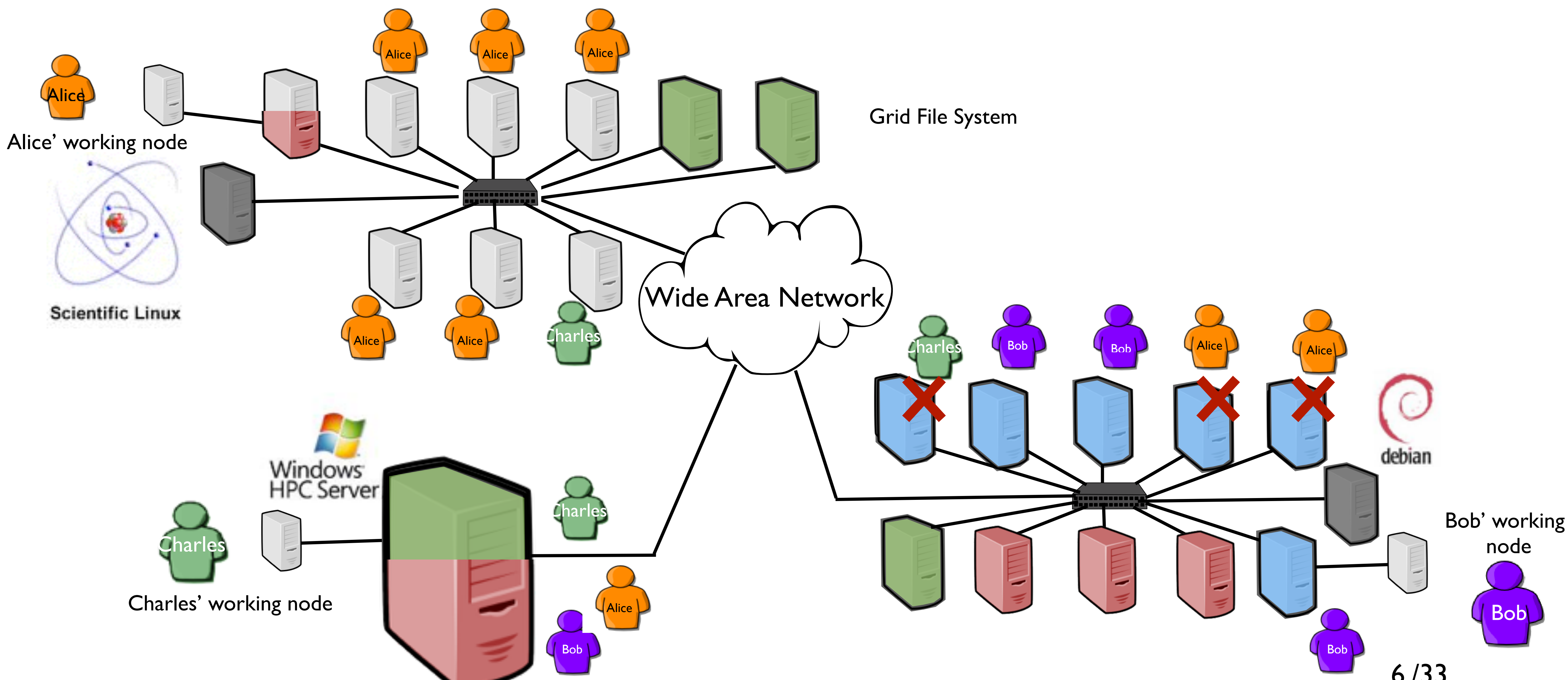
Looking back…

- Network of Workstations 1990 / 20xx
- Desktop 1998 / 201x
- Grid 1998 / 201x



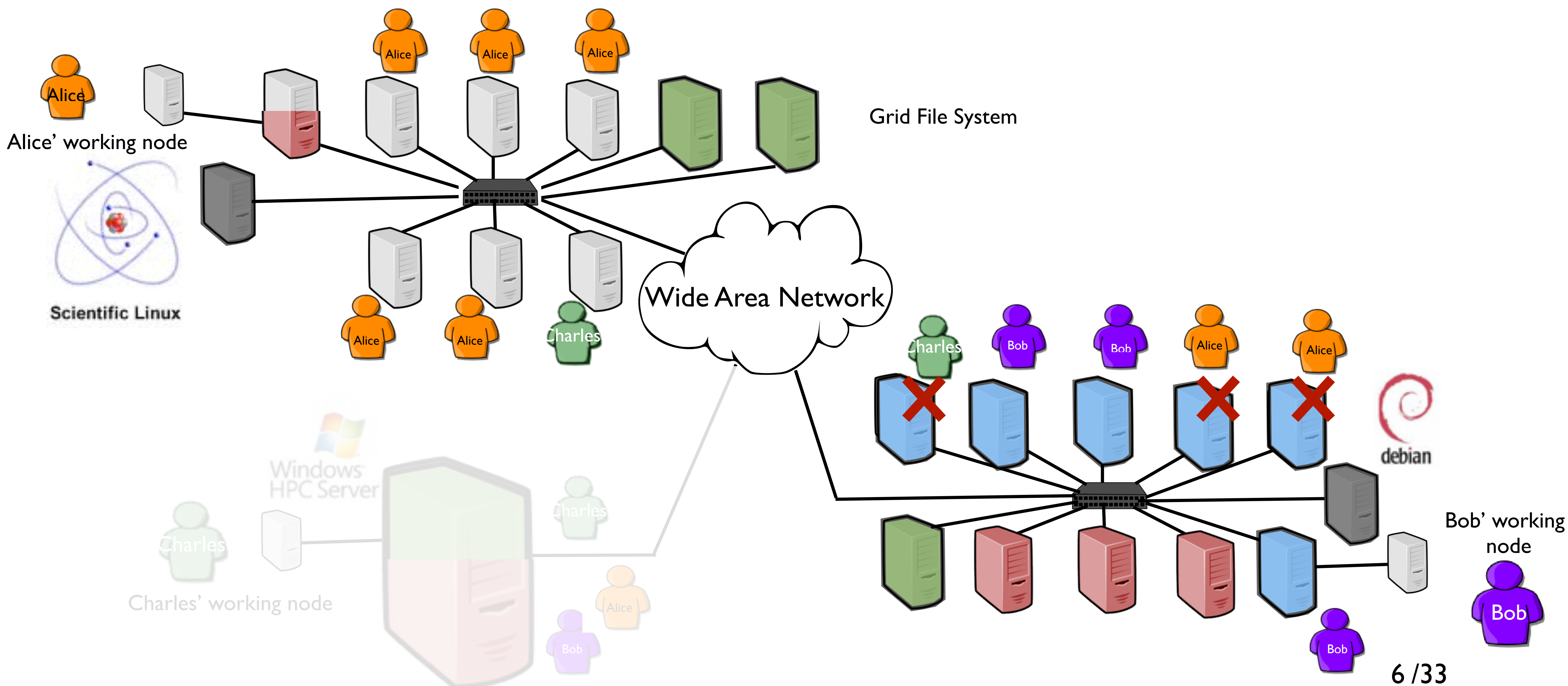
Looking back…

- Network of Workstations 1990 / 20xx
- Desktop 1998 / 201x
- Grid 1998 / 201x



Looking back…

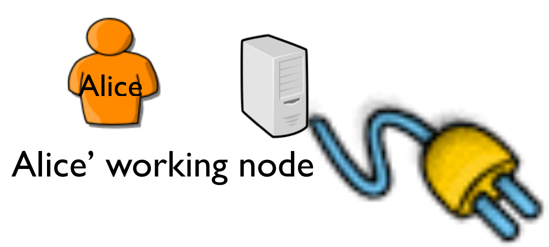
- Network of Workstations 1990 / 20xx
- Desktop 1998 / 201x
- Grid 1998 / 201x



Looking back…

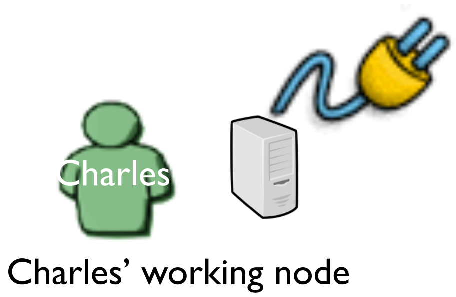
- Network of Workstations 1990 / 20xx
- Desktop 1998 / 201x
- Grid 1998 / 201x

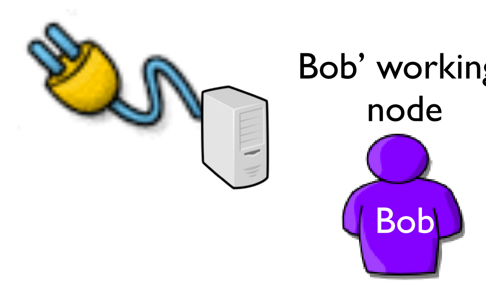
What a Grid ! ? !



Resource booking (based on user's estimates)
Security concerns (job isolation)
Heterogeneity concerns (hardware and software)
Scheduling limitations (a job cannot be easily relocated)
Fault tolerance issues

…

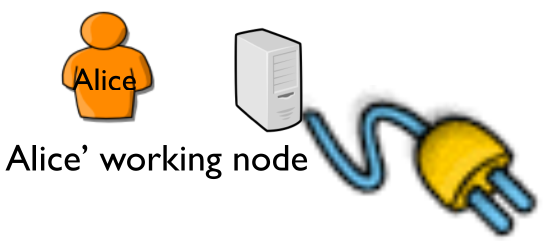




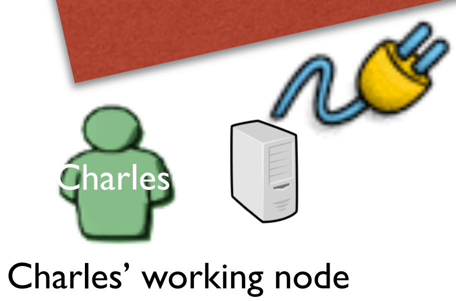
Looking back…

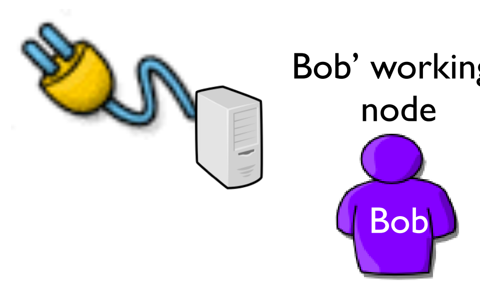
- Network of Workstations 1990 / 20xx
- Desktop 1998 / 201x
- Grid 1998 / 201x

What a Grid ! ? !



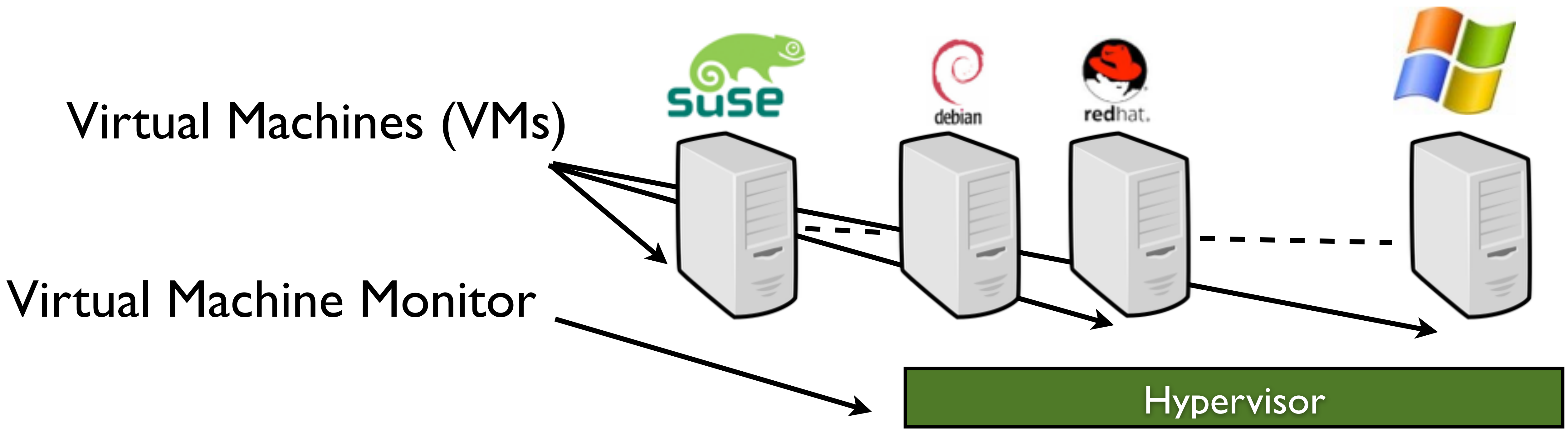
A lot of progress has been done since the 90's and several proposals partially addressed these concerns.





Looking back…

- System virtualization: One to multiple OSes on a physical node thanks to a hypervisor (an operating system of OSes)

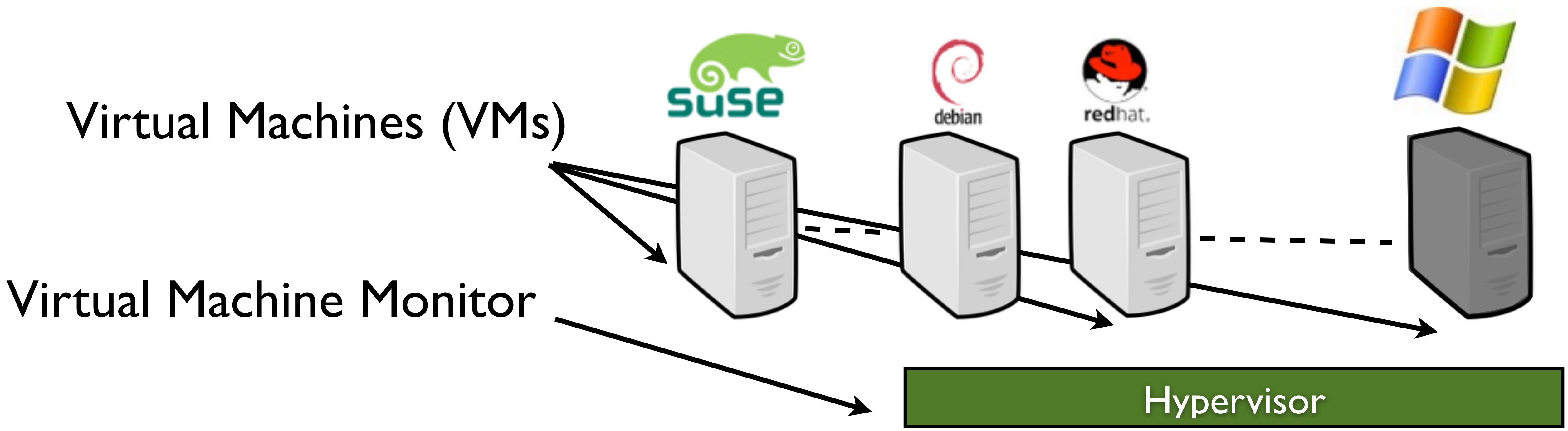


"A ***virtual machine*** (VM) provides a faithful implementation of a physical processor's hardware running in a protected and isolated environment.
Virtual machines are created by a software layer called the ***virtual machine monitor*** (VMM) that runs as a privileged task on a physical processor."

Physical Machine (PM)

Looking back…

- System virtualization: One to multiple OSes on a physical node thanks to a hypervisor (an operating system of OSes)

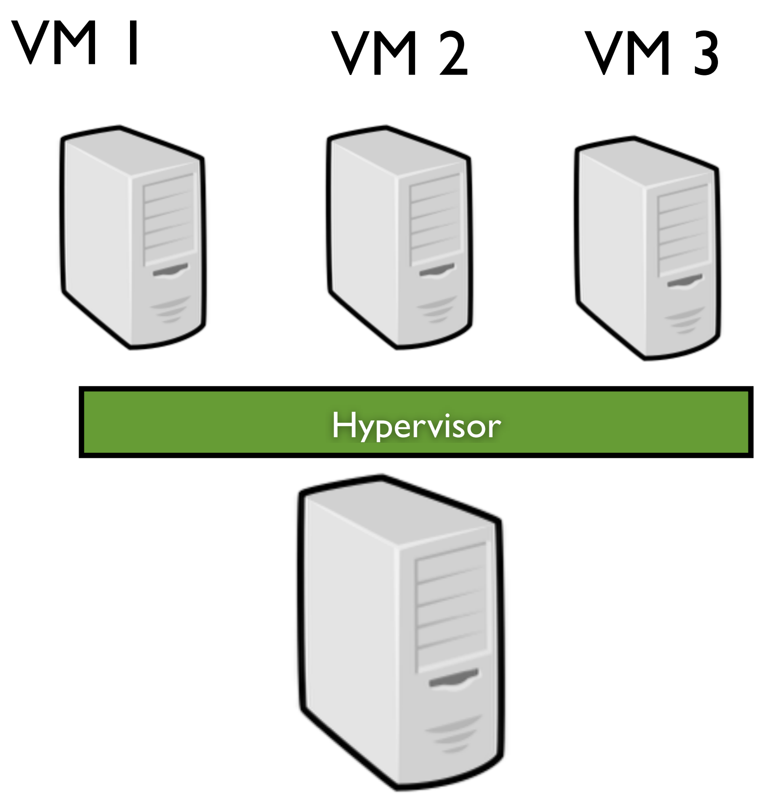


"A ***virtual machine*** (VM) provides a faithful implementation of a physical processor's hardware running in a protected and isolated environment.
Virtual machines are created by a software layer called the ***virtual machine monitor*** (VMM) that runs as a privileged task on a physical processor."

Physical Machine (PM)

Looking back…

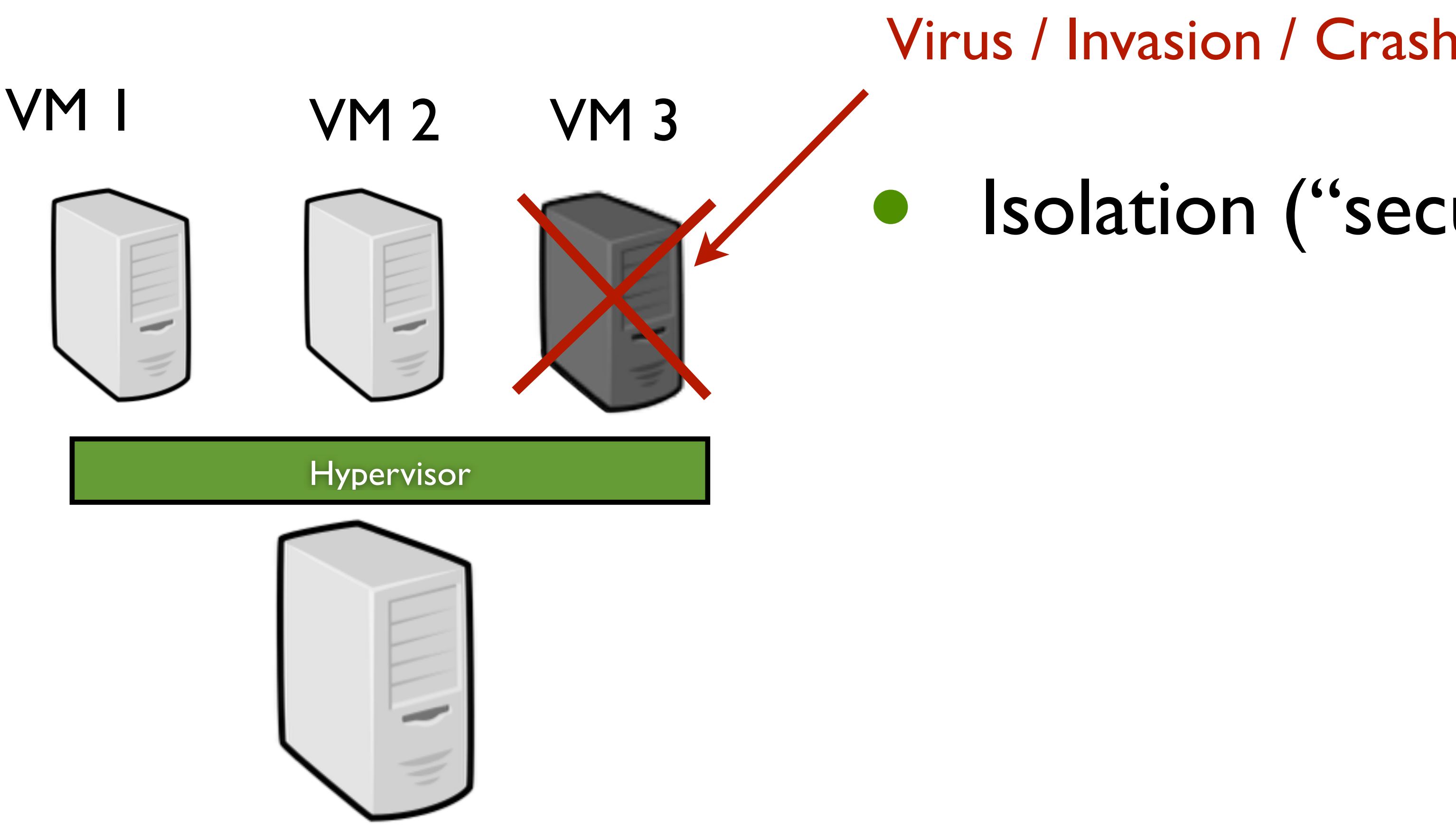
- System virtualization: a great sandbox



- Isolation ("security" between each VM)

Looking back…

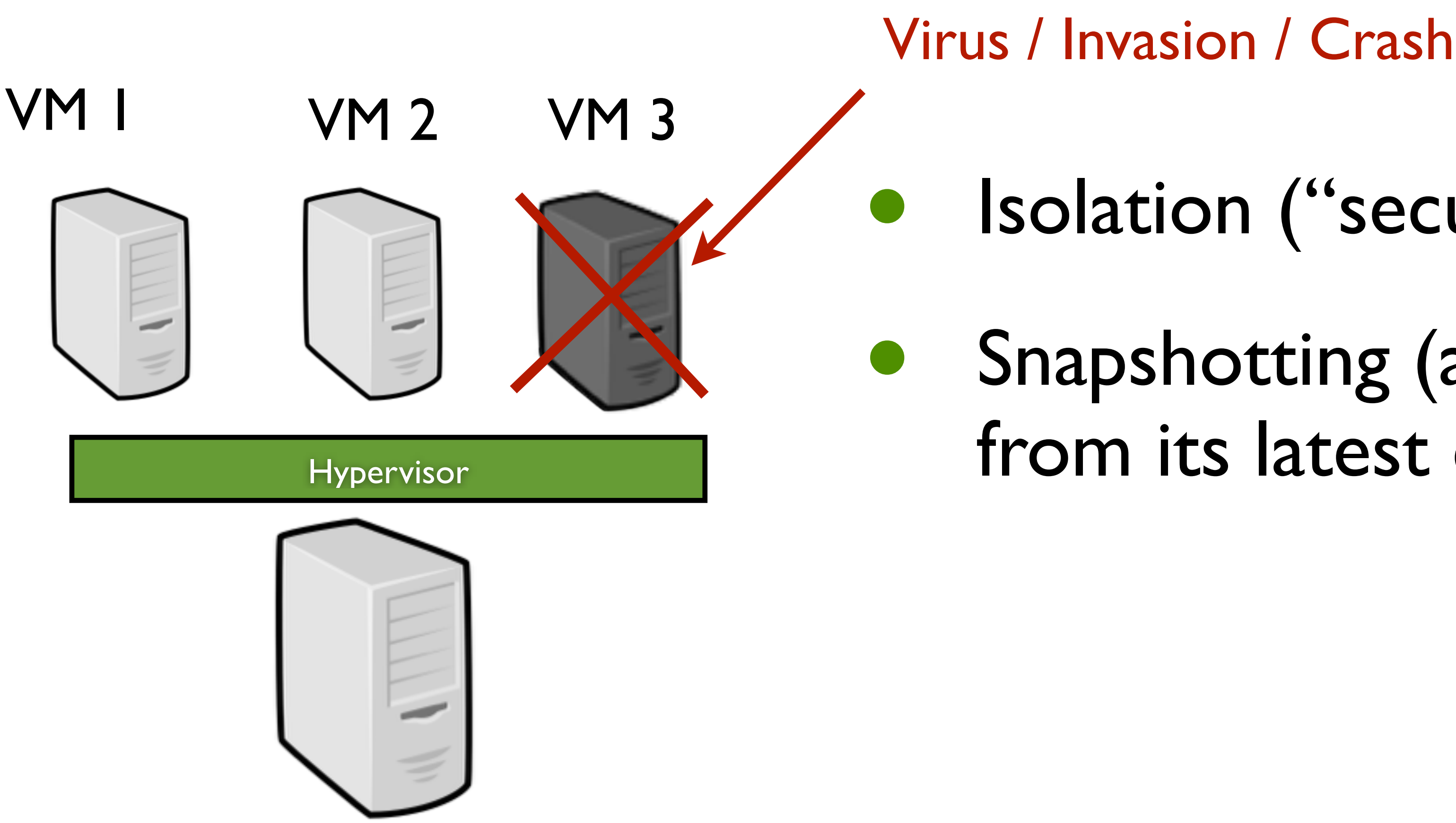
- System virtualization: a great sandbox



- Isolation ("security" between each VM)

Looking back…

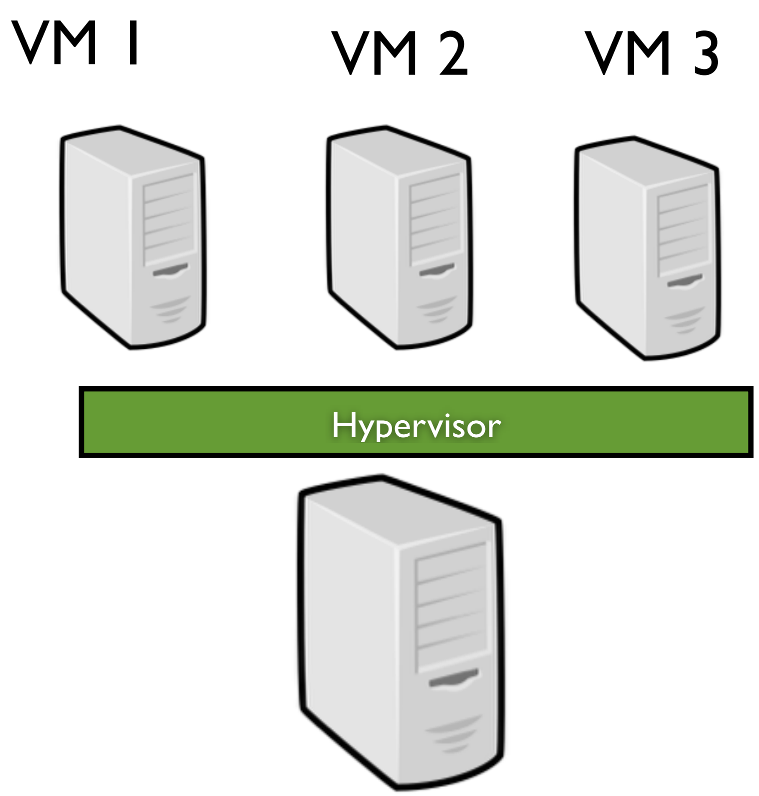
- System virtualization: a great sandbox



- Isolation ("security" between each VM)
- Snapshotting (a VM can be easily resume from its latest consistent state)

Looking back…

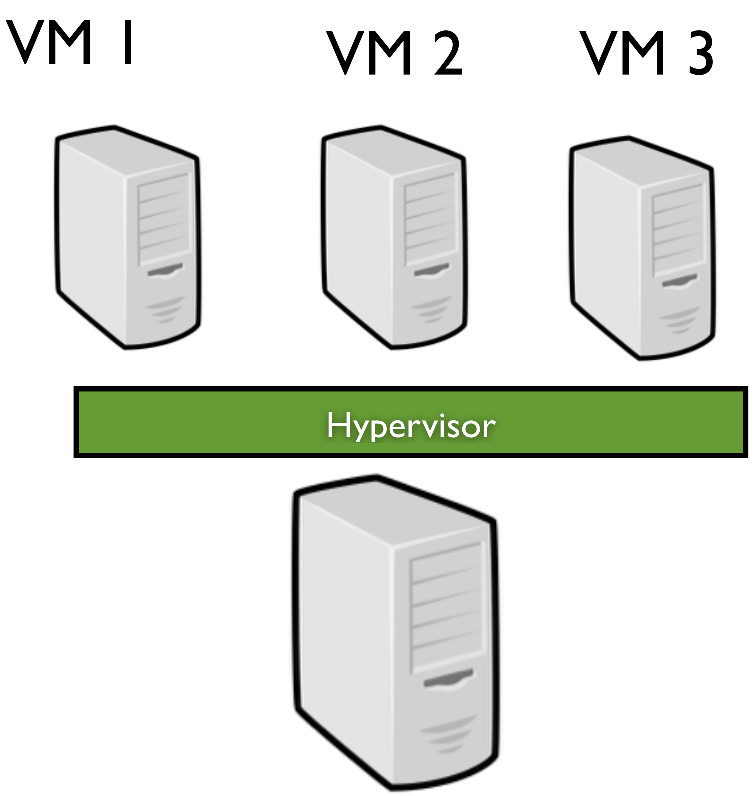
- System virtualization: a great sandbox



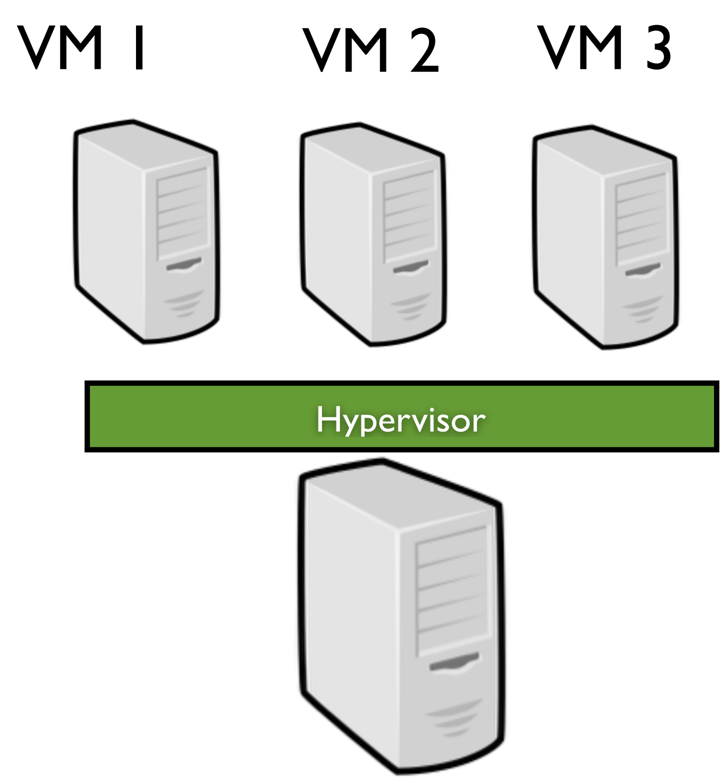
- Isolation ("security" between each VM)
- Snapshotting (a VM can be easily resume from its latest consistent state)

Looking back…

- System virtualization: a great sandbox



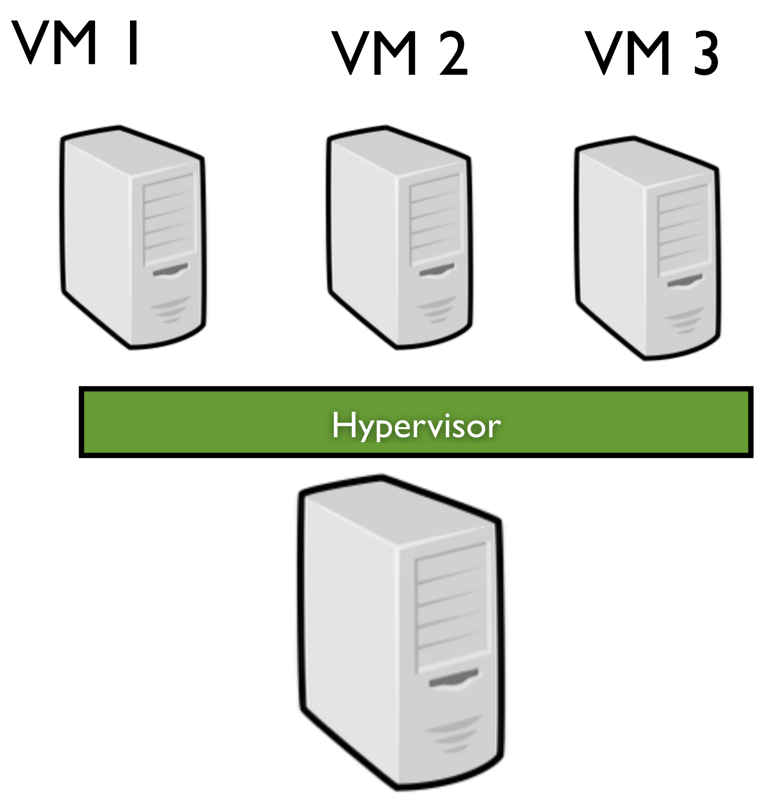
- Isolation (“security” between each VM)
- Snapshotting (a VM can be easily resume from its latest consistent state)



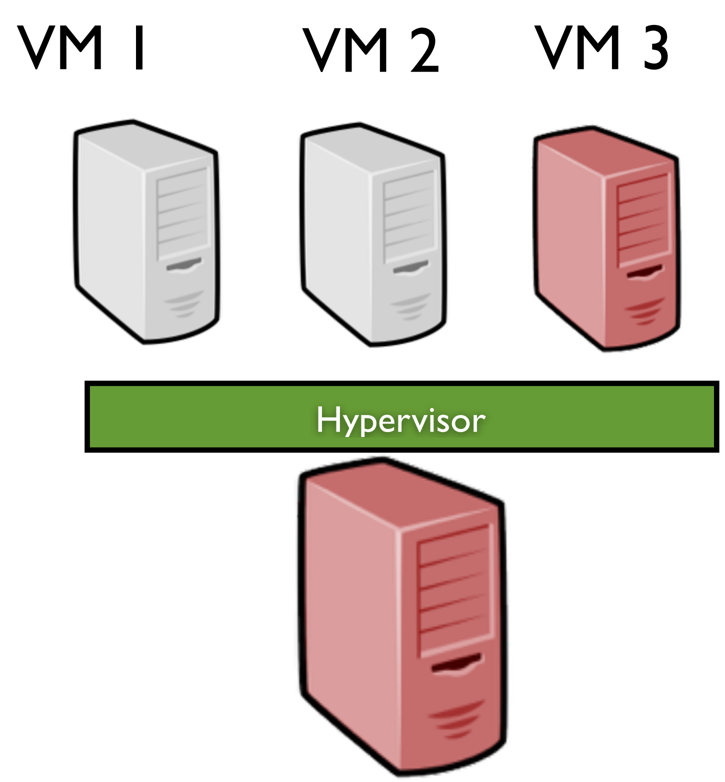
- Suspend/Resume

Looking back…

- System virtualization: a great sandbox



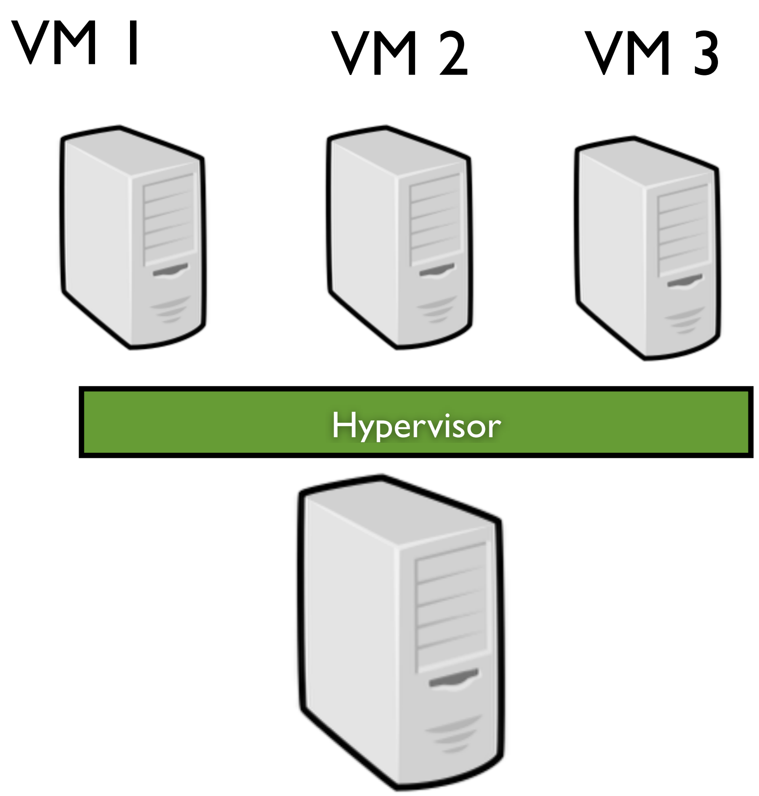
- Isolation ("security" between each VM)
- Snapshotting (a VM can be easily resume from its latest consistent state)



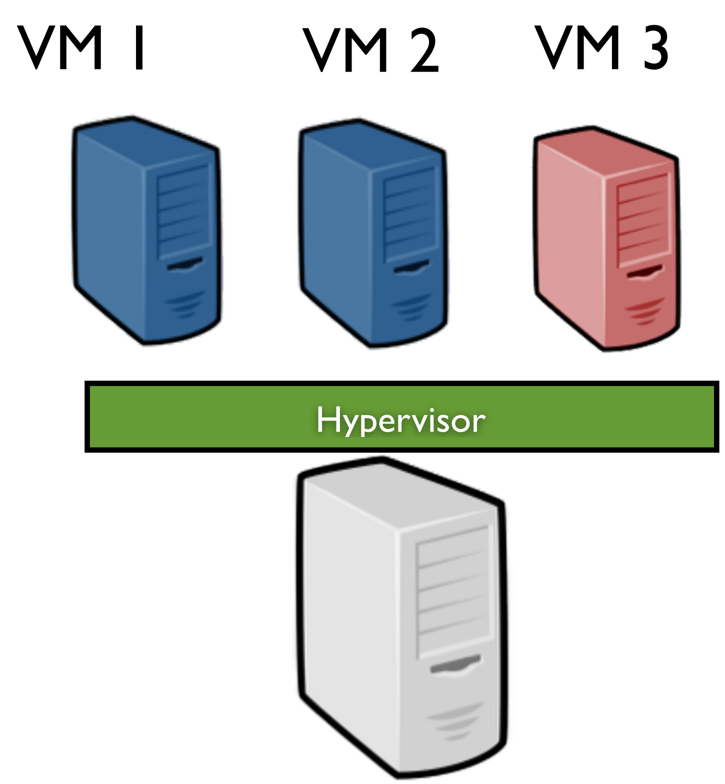
- Suspend/Resume

Looking back…

- System virtualization: a great sandbox



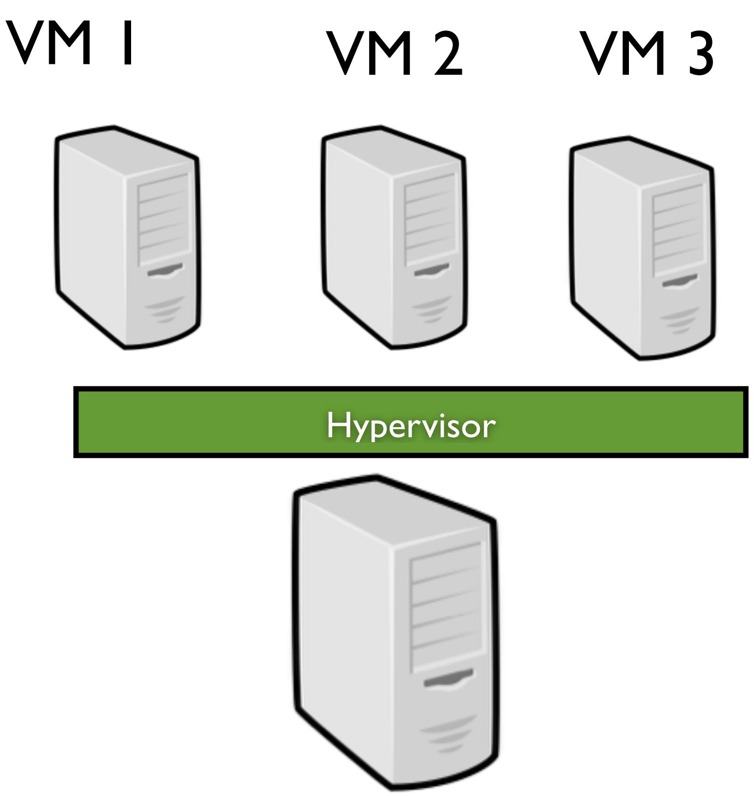
- Isolation (“security” between each VM)
- Snapshotting (a VM can be easily resume from its latest consistent state)



- Suspend/Resume

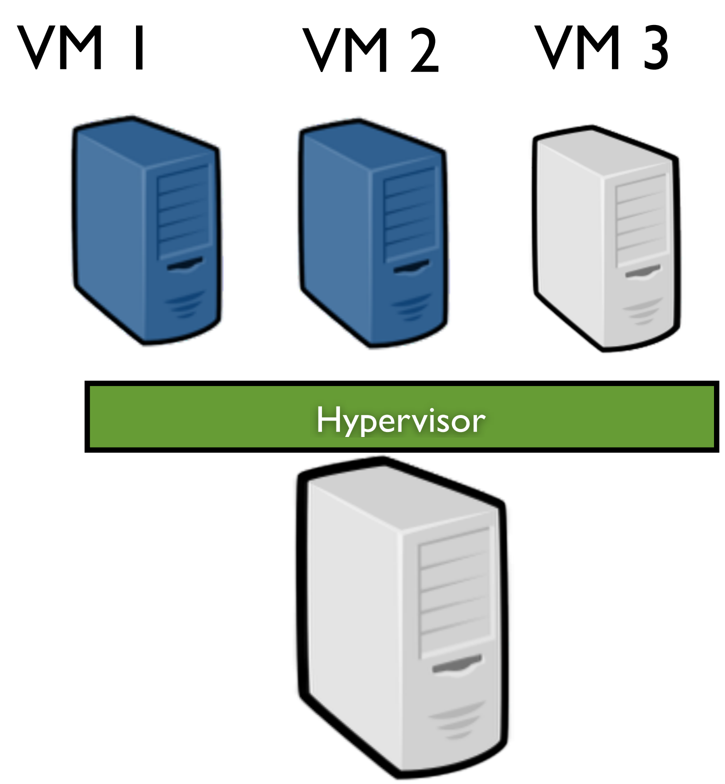
Looking back…

- System virtualization: a great sandbox



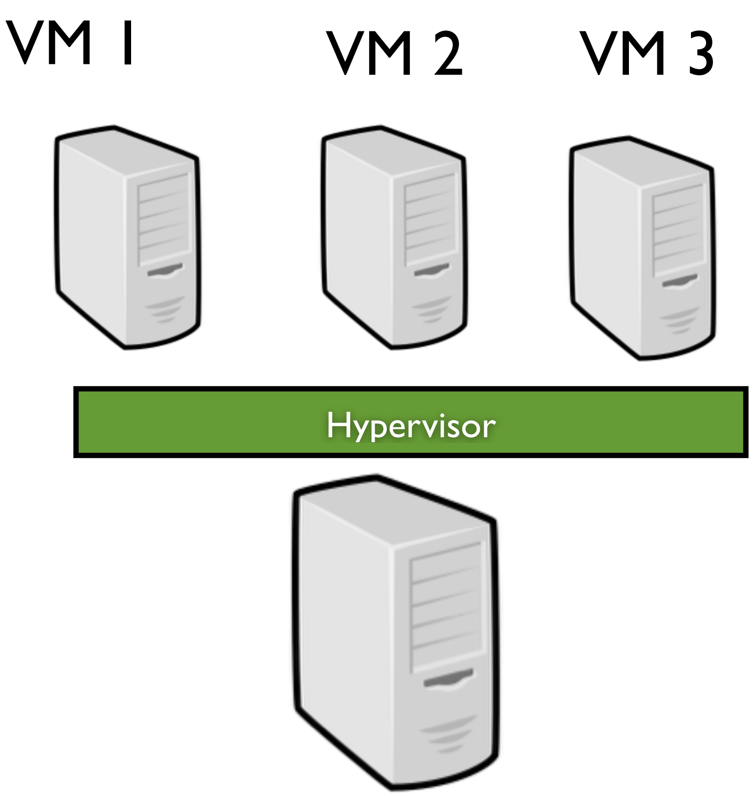
- Isolation (“security” between each VM)
- Snapshotting (a VM can be easily resume from its latest consistent state)

- Suspend/Resume



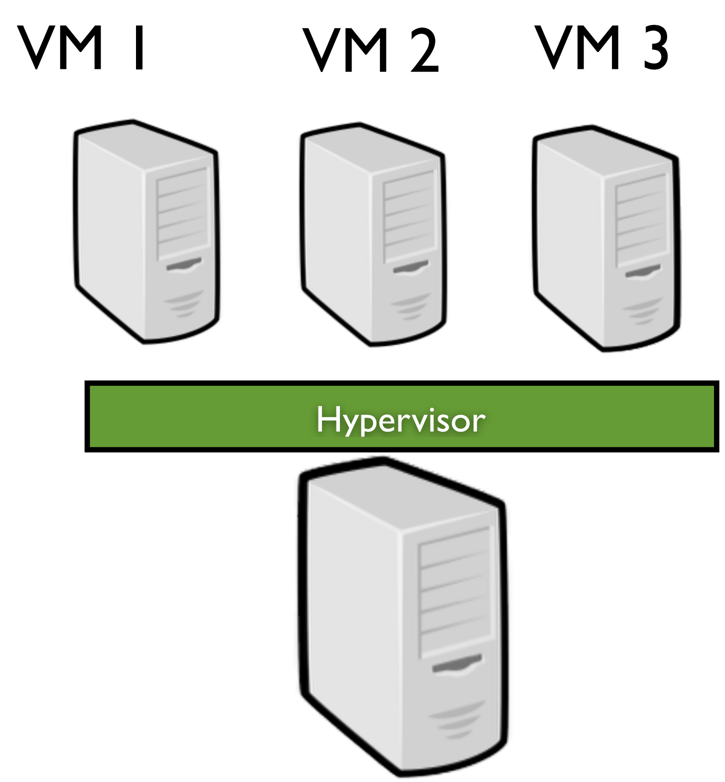
Looking back…

- System virtualization: a great sandbox



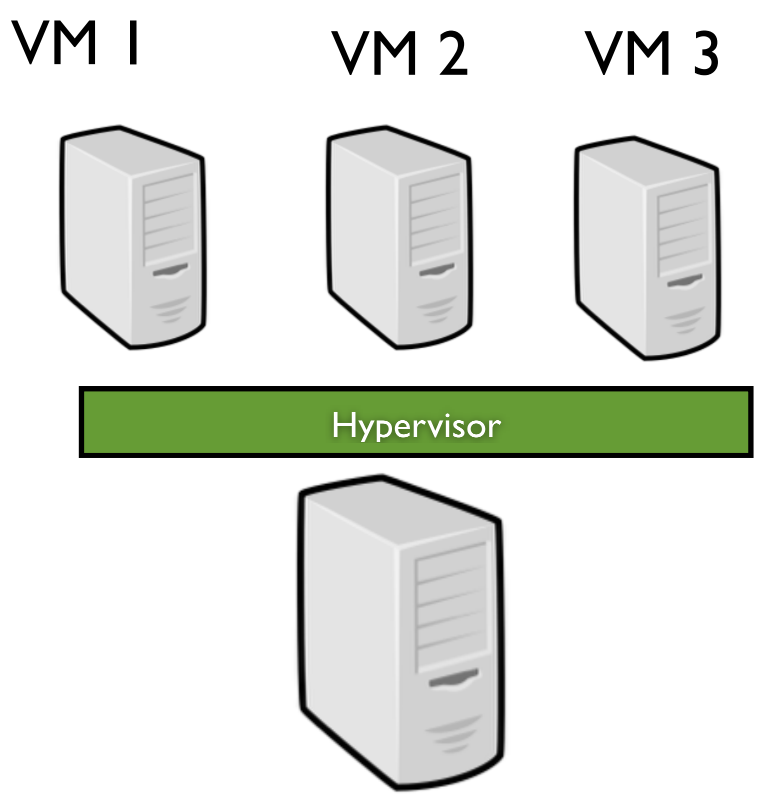
- Isolation ("security" between each VM)
- Snapshotting (a VM can be easily resume from its latest consistent state)

- Suspend/Resume

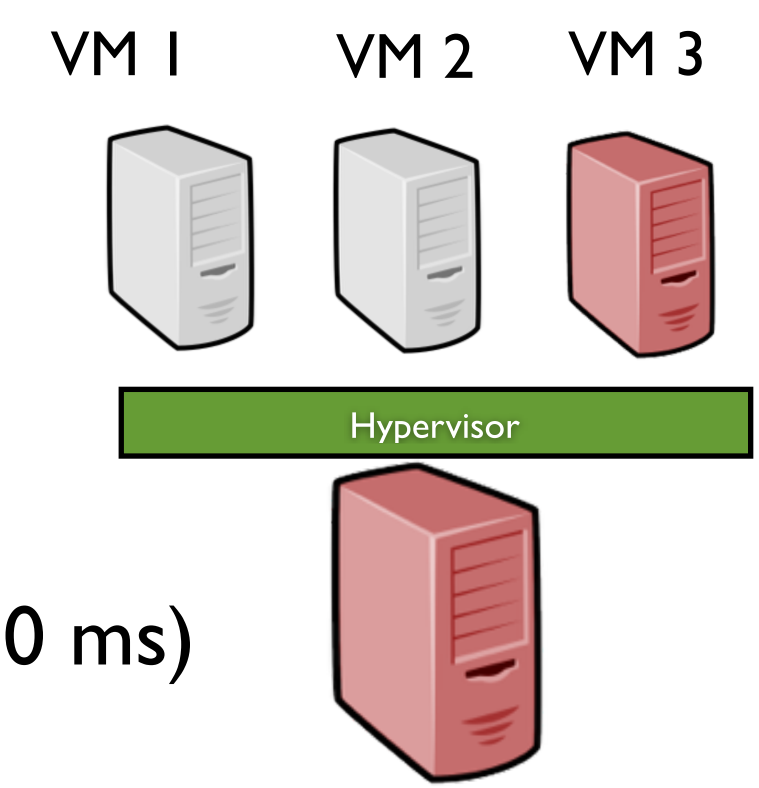


Looking back…

- System virtualization: a great sandbox



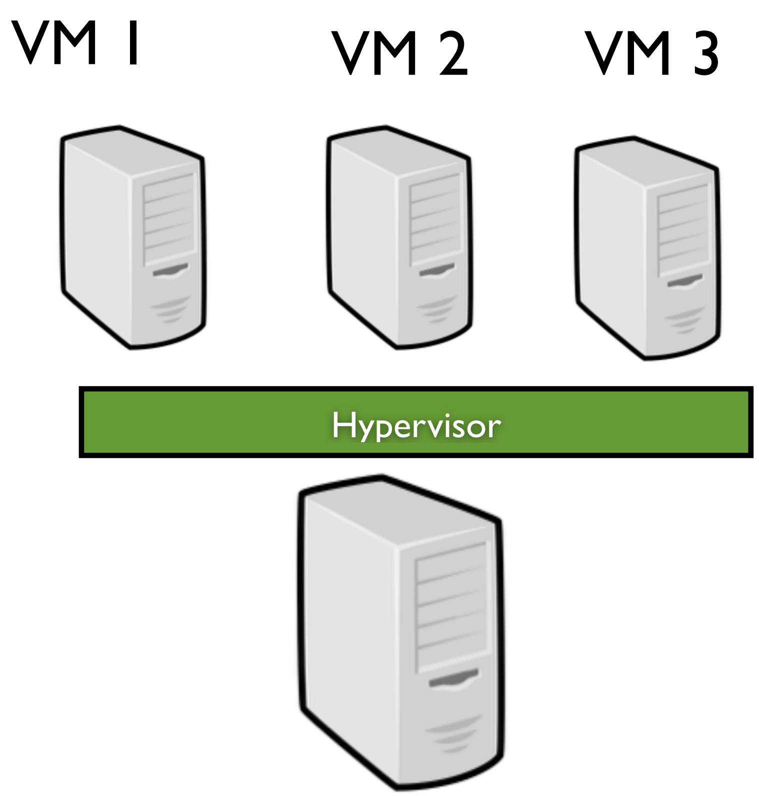
- Isolation (“security” between each VM)
- Snapshotting (a VM can be easily resume from its latest consistent state)



- Suspend/Resume
- Live migration
  (negligible downtime ~ 60 ms)
  Post/Pre Copy

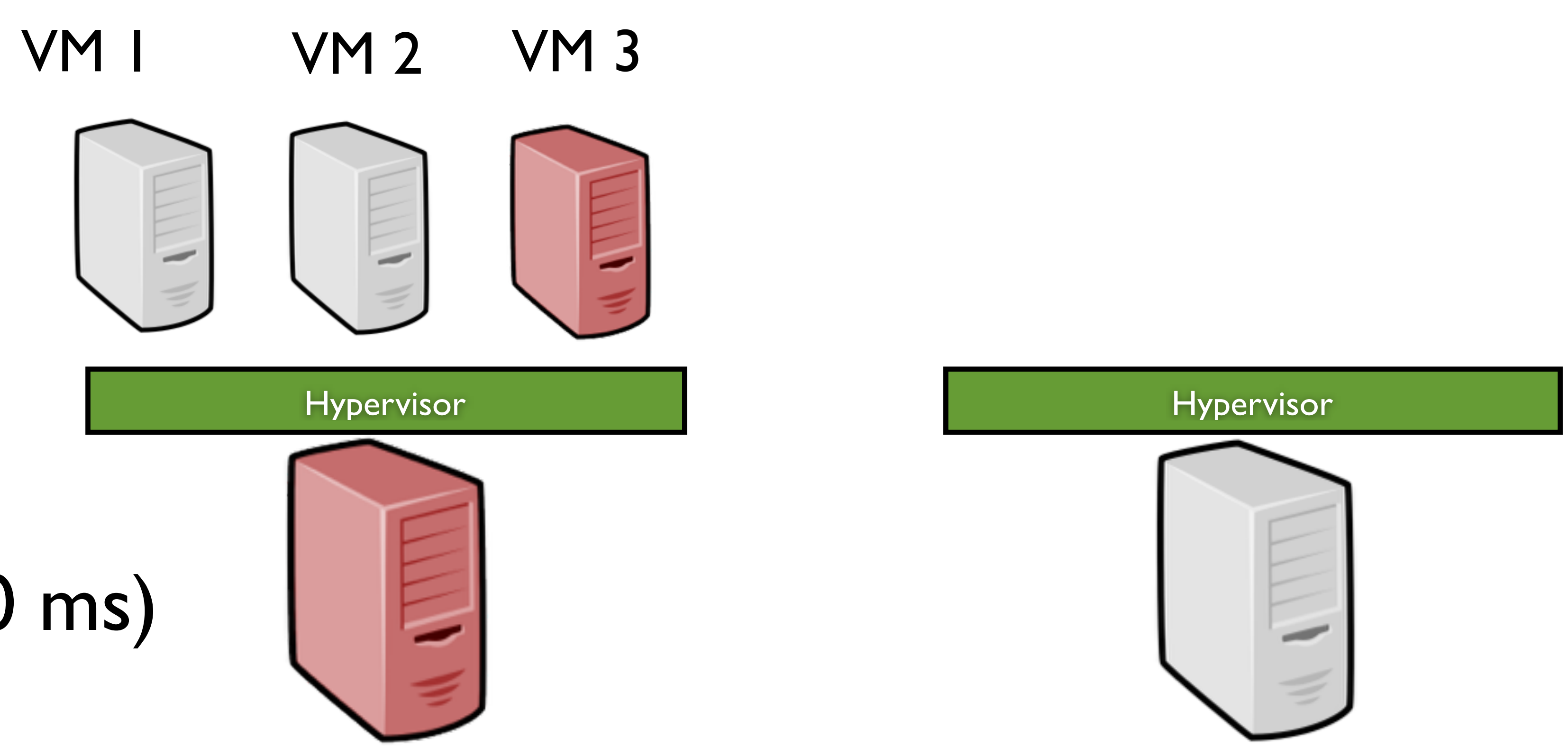
Looking back…

- System virtualization: a great sandbox



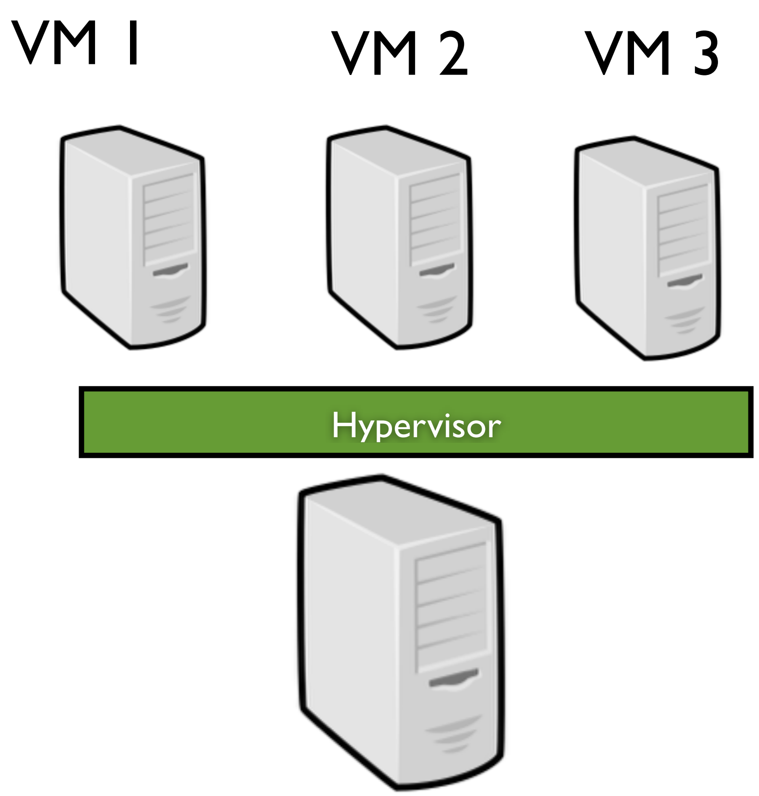
- Isolation (“security” between each VM)
- Snapshotting (a VM can be easily resume from its latest consistent state)

- Suspend/Resume
- Live migration
(negligible downtime ~ 60 ms)
Post/Pre Copy



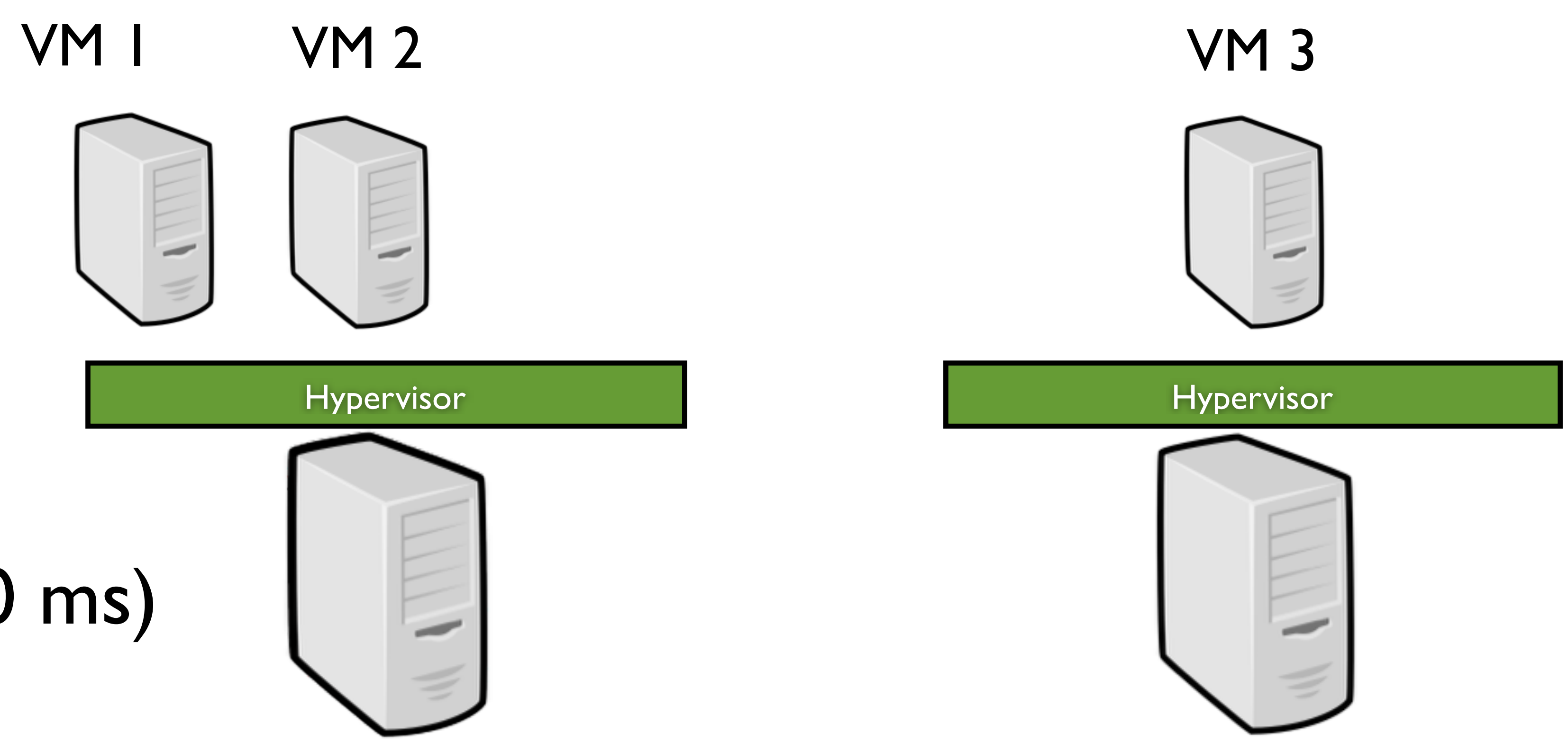
Looking back…

- System virtualization: a great sandbox



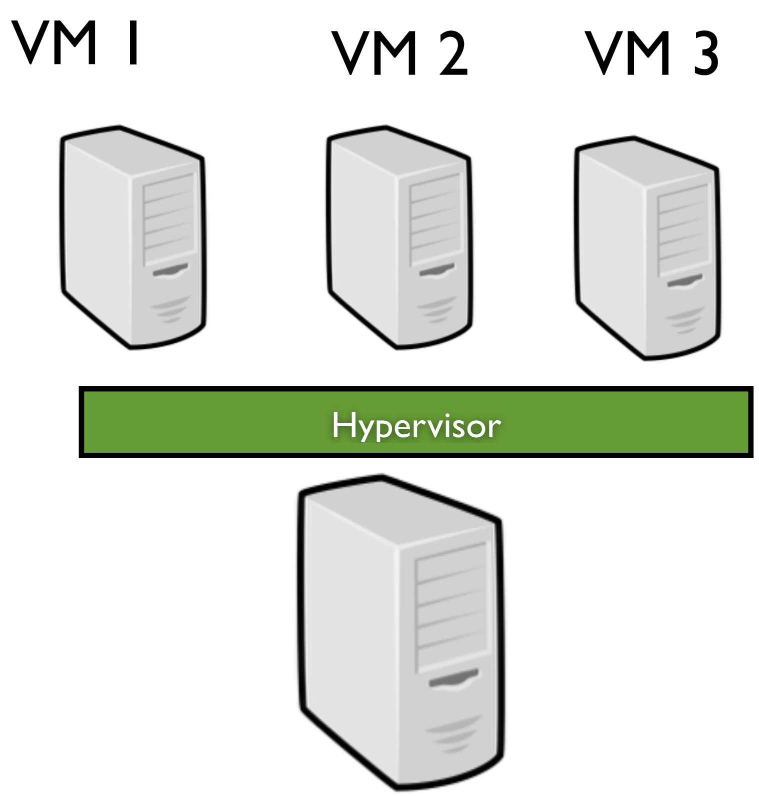
- Isolation ("security" between each VM)
- Snapshotting (a VM can be easily resume from its latest consistent state)

- Suspend/Resume
- Live migration
  (negligible downtime ~ 60 ms)
  Post/Pre Copy

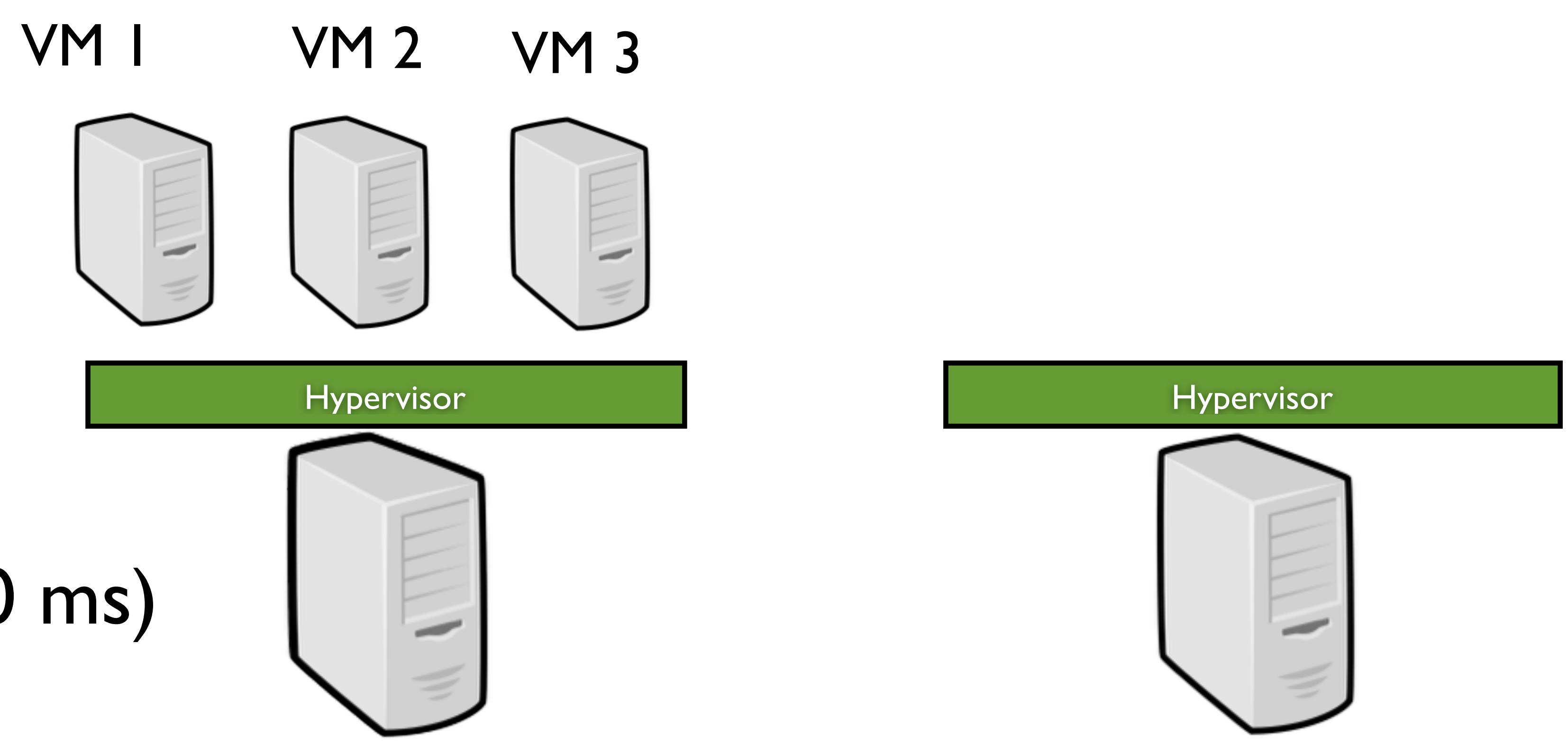


Looking back…

- System virtualization: a great sandbox



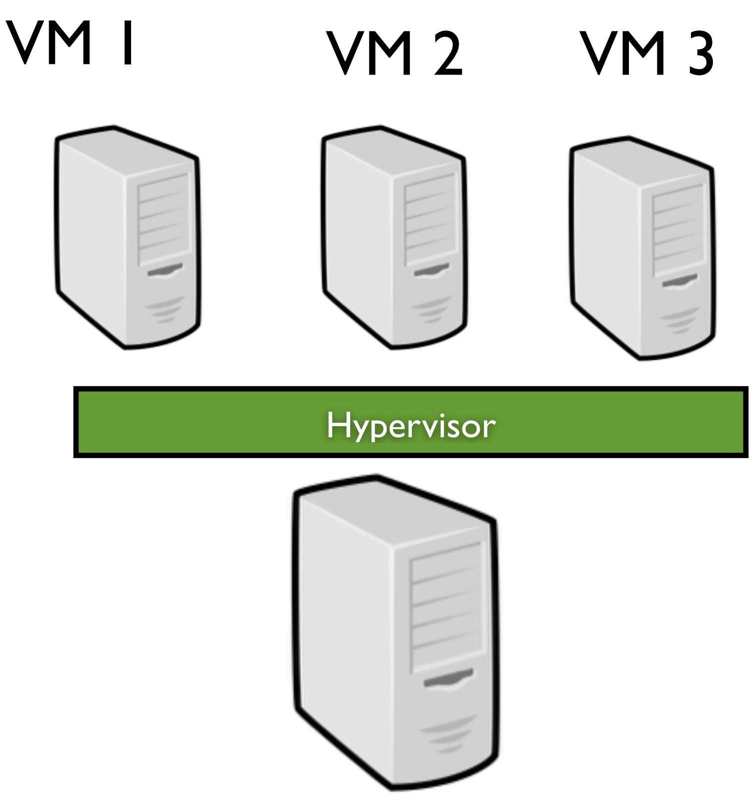
- Isolation ("security" between each VM)
- Snapshotting (a VM can be easily resume from its latest consistent state)



- Suspend/Resume
- Live migration
  (negligible downtime ~ 60 ms)
  Post/Pre Copy

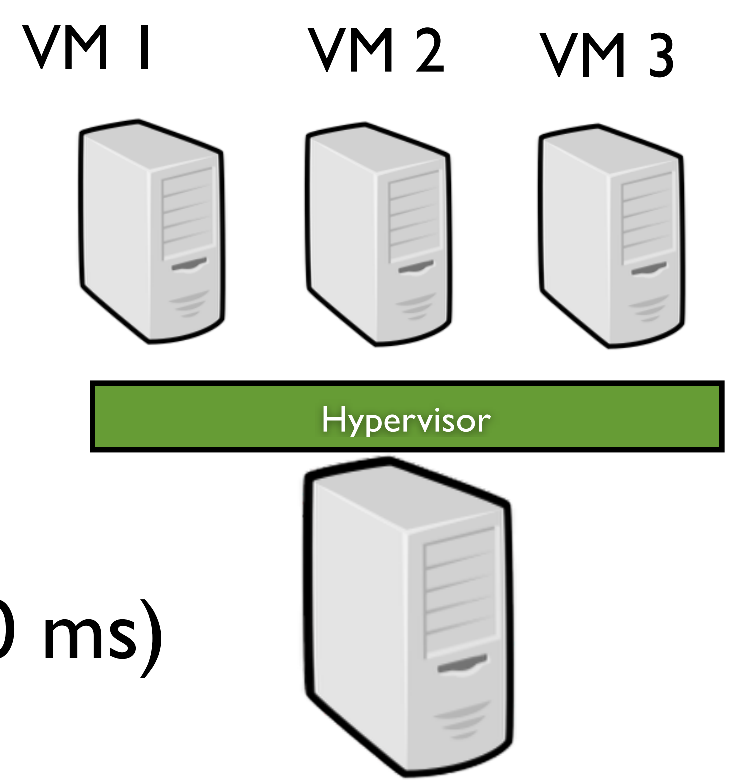
Looking back…

- System virtualization: a great sandbox



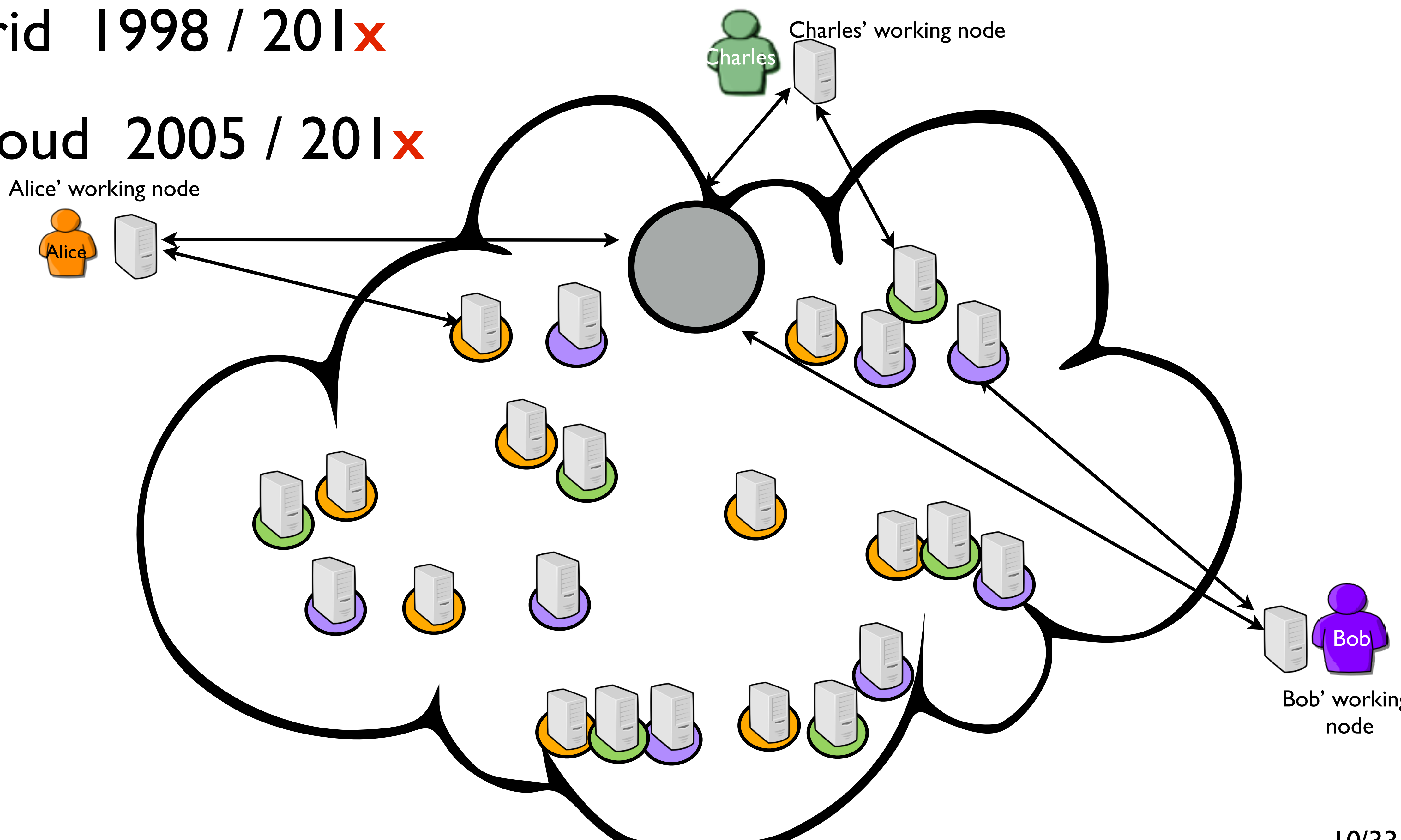
- Isolation ("security" between each VM)
- Snapshotting (a VM can be easily resume from its latest consistent state)

- Suspend/Resume
- Live migration
(negligible downtime ~ 60 ms)
Post/Pre Copy



Looking back ...

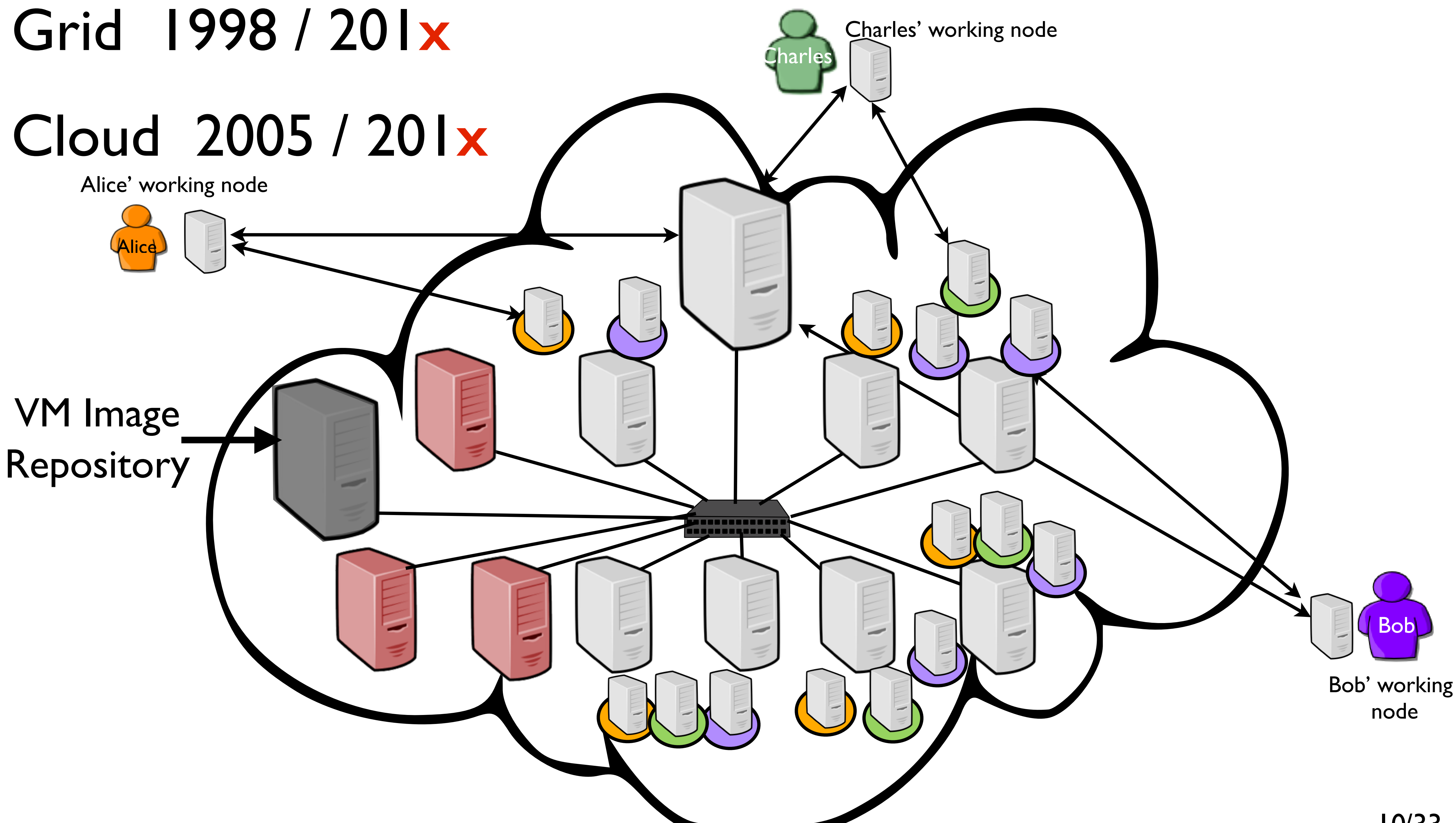
- Network of Workstations 1990 / 20xx
- Desktop Computing 1998/201x
- Grid 1998 / 201x
- Cloud 2005 / 201x



Infrastructure as a Service model

Looking back ...

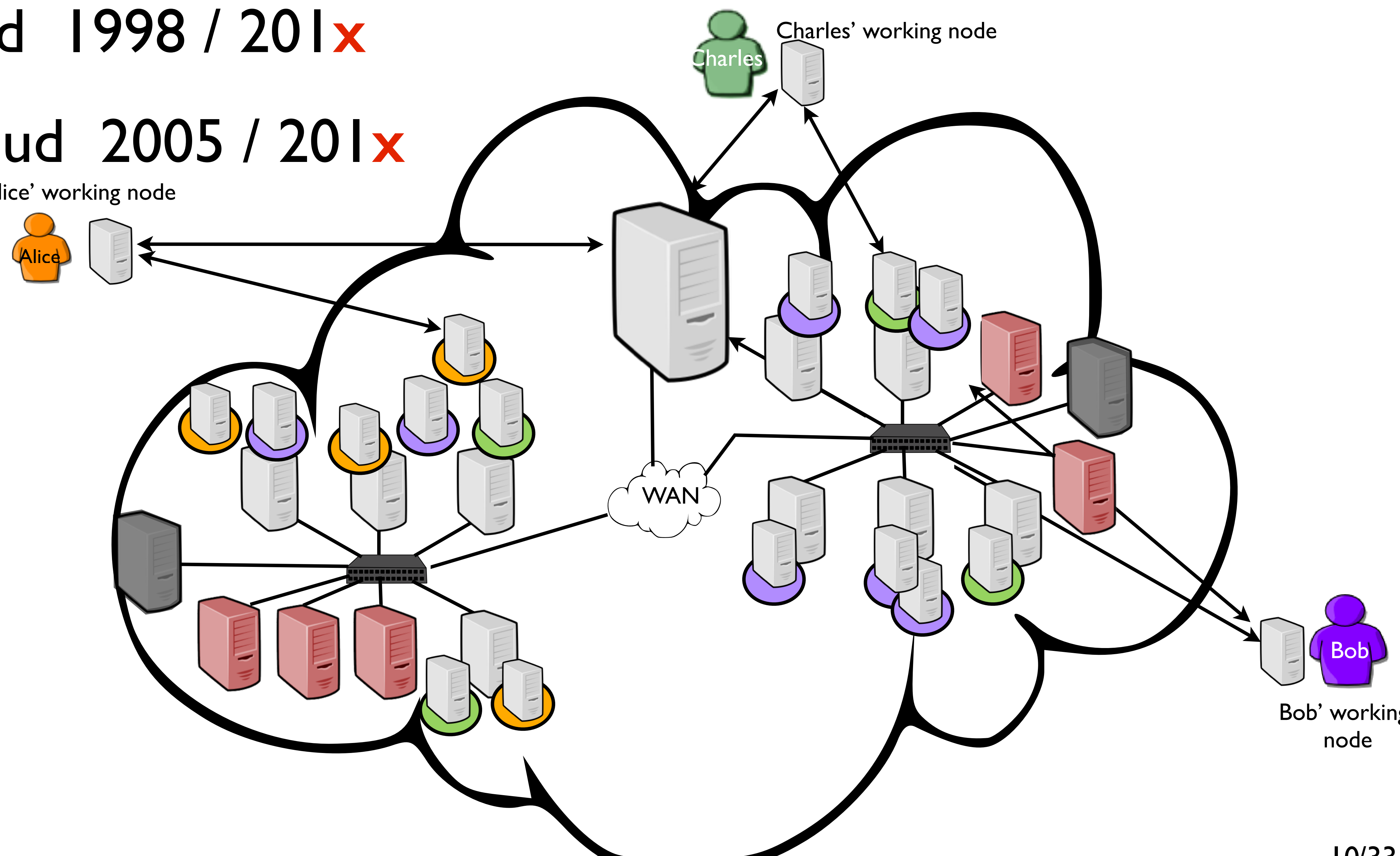
- Network of Workstations 1990 / 20xx
- Desktop Computing 1998/201x
- Grid 1998 / 201x
- Cloud 2005 / 201x



Infrastructure as a Service model

Looking back …

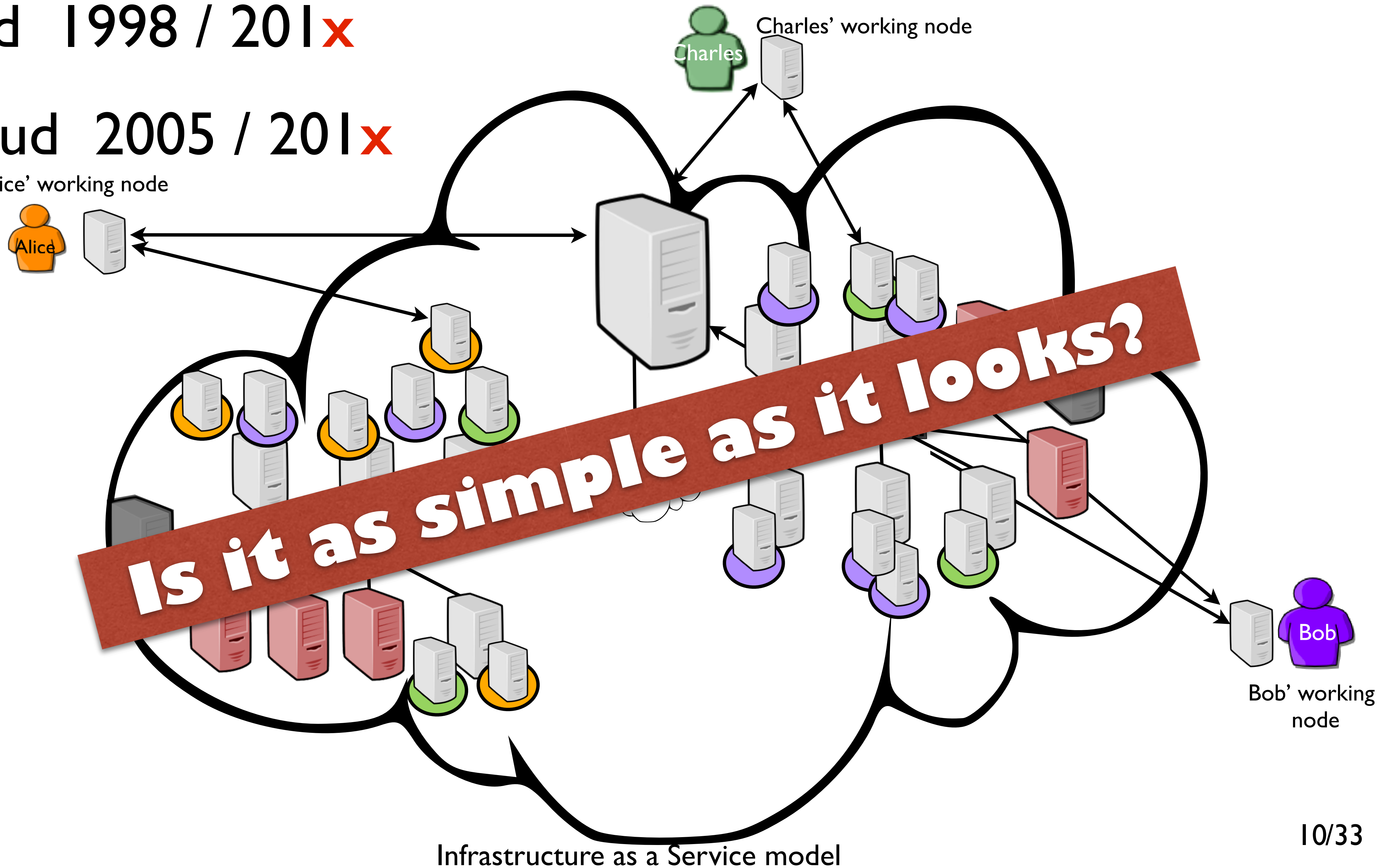
- Network of Workstations 1990 / 20**xx**
- Desktop Computing 1998/201**x**
- Grid 1998 / 201**x**
- Cloud 2005 / 201**x**



Infrastructure as a Service model

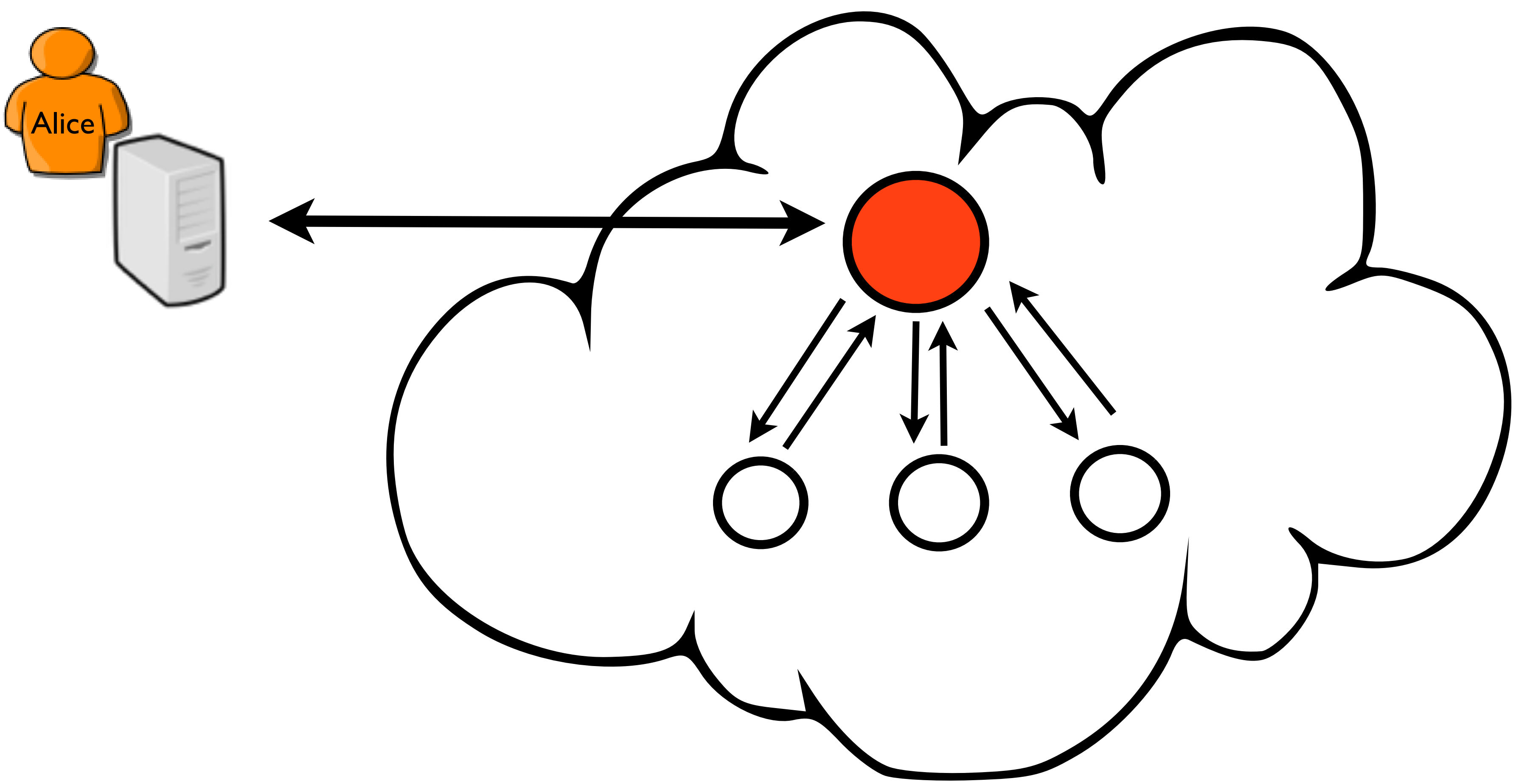
Looking back …

- Network of Workstations 1990 / 20xx
- Desktop Computing 1998/201x
- Grid 1998 / 201x
- Cloud 2005 / 201x



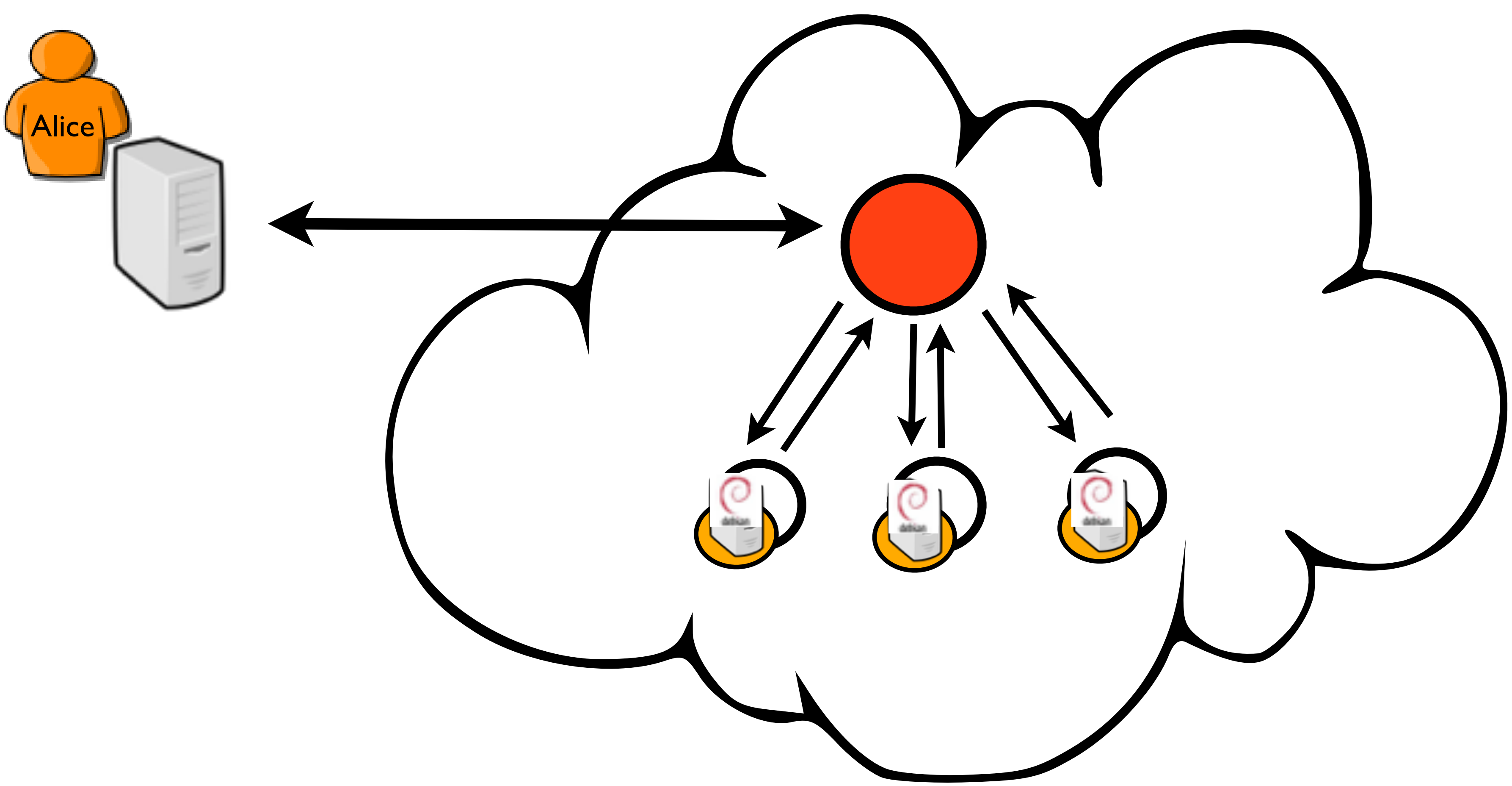
IaaS Resource Management Systems

- An Operating System for Cloud infrastructures (aka Cloudkits)
  - Configuration of Virtual Environments (VEs) (contextualization, network...)
  - Images management/deployment
  - “Secure” accesses to the VEs



IaaS Resource Management Systems

- An Operating System for Cloud infrastructures (aka Cloudkits)
  - Configuration of Virtual Environments (VEs) (contextualization, network...)
  - Images management/deployment
  - “Secure” accesses to the VEs



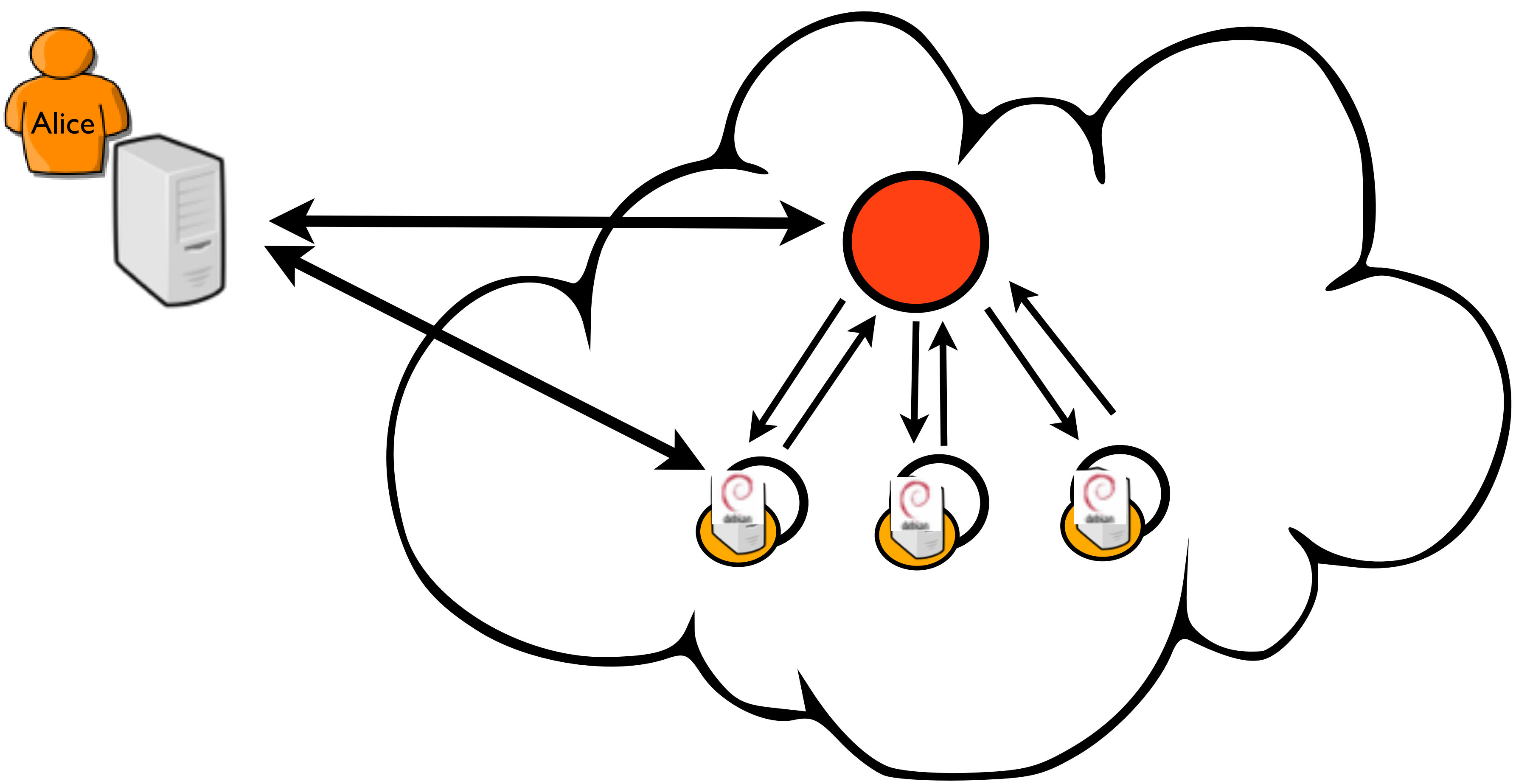
IaaS Resource Management Systems

- An Operating System for Cloud infrastructures (aka Cloudkits)

  Configuration of Virtual Environments (VEs) (contextualization, network...)

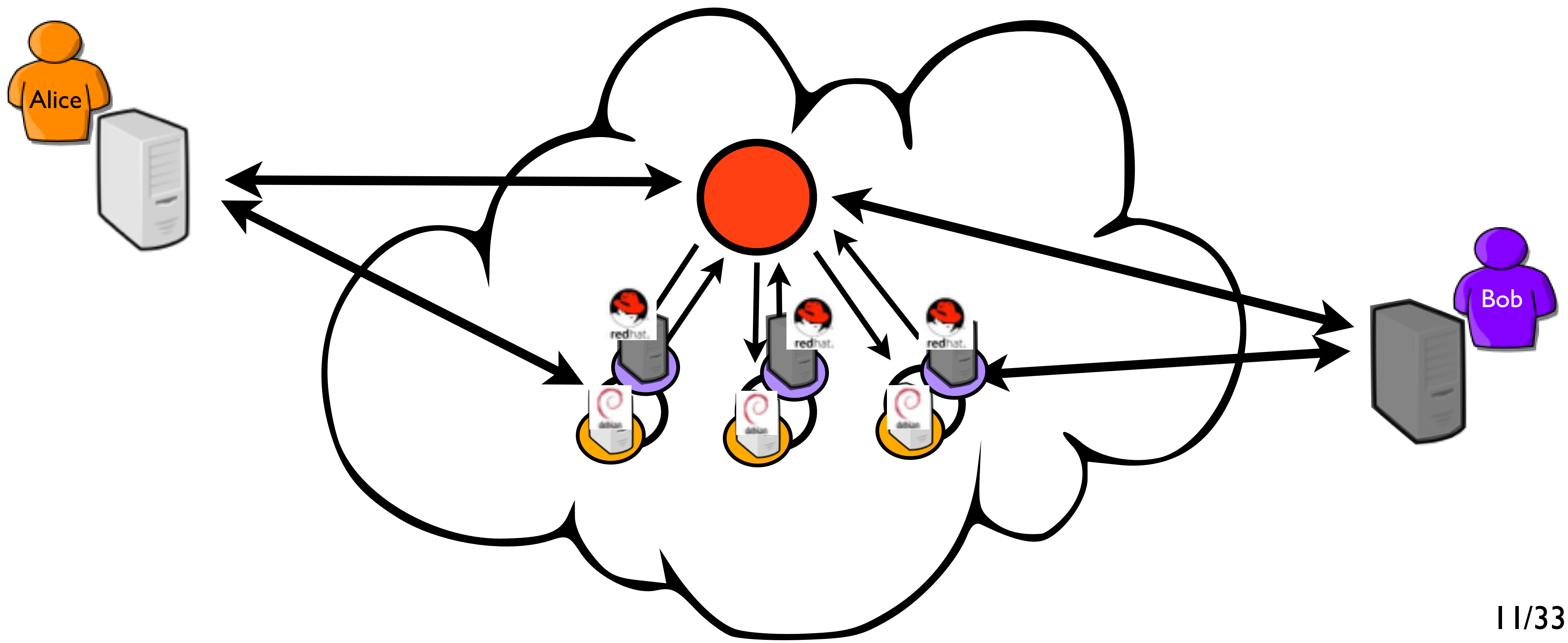
  Images management/deployment

  "Secure" accesses to the VEs

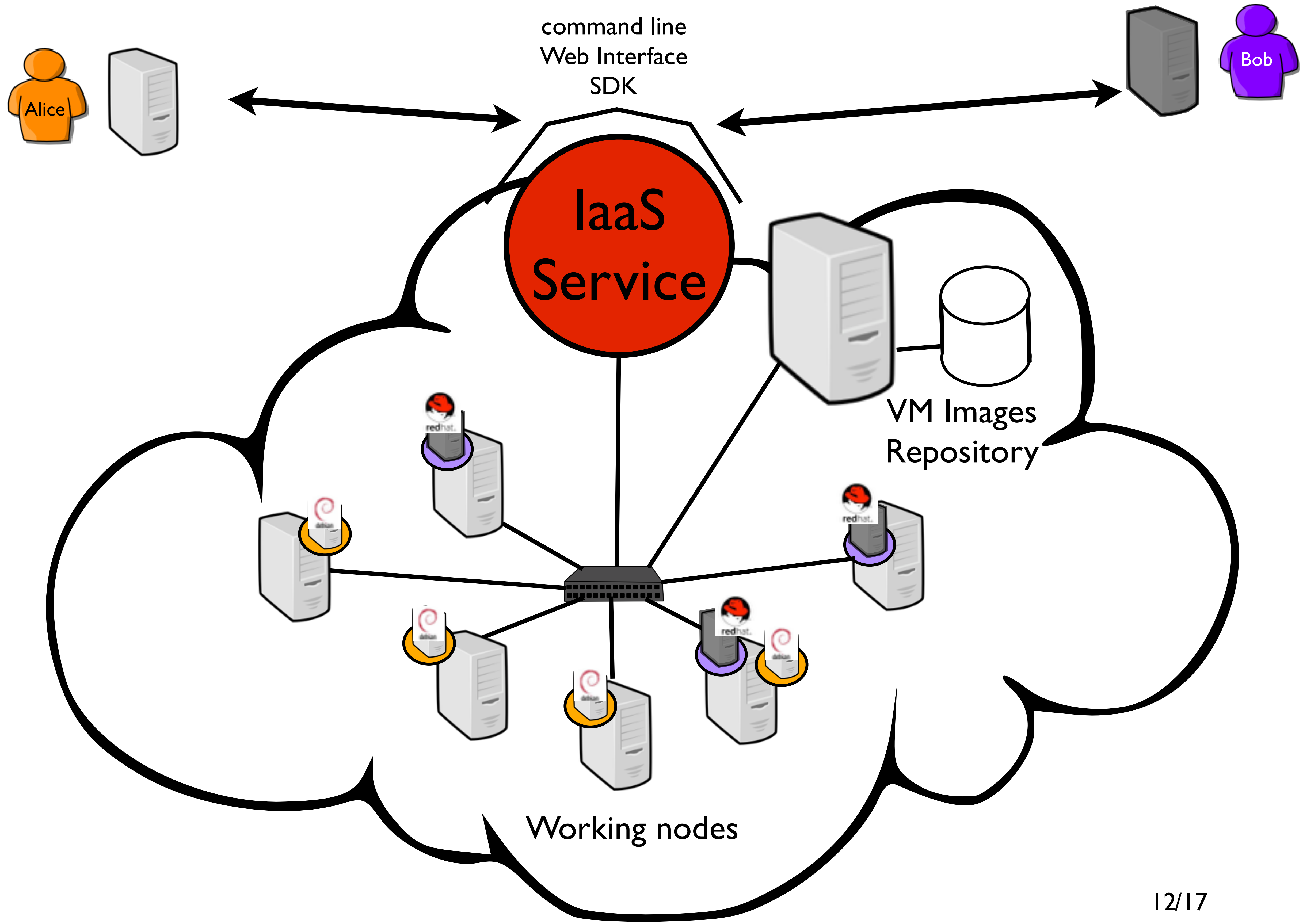


IaaS Resource Management Systems

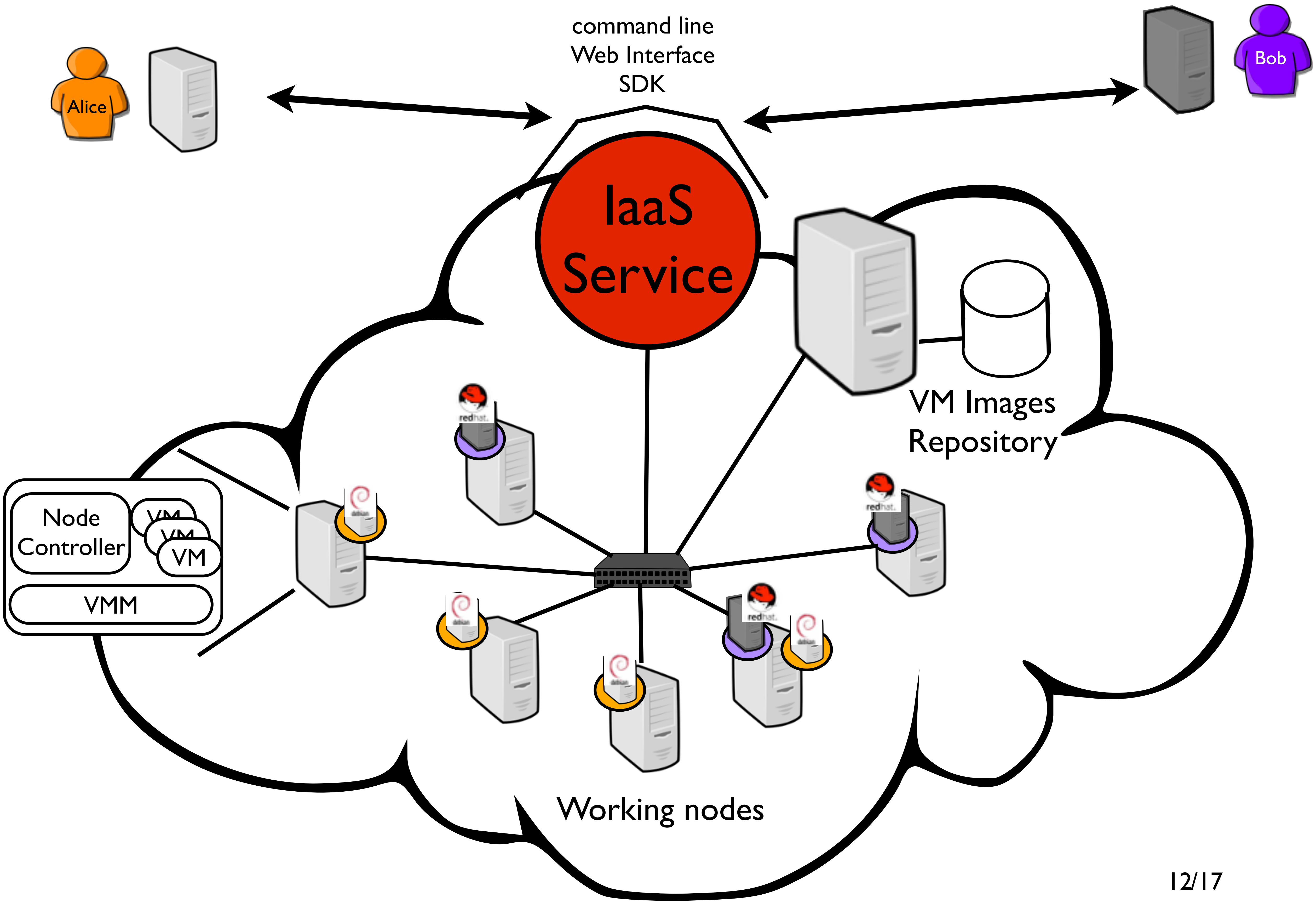
- An Operating System for Cloud infrastructures (aka Cloudkits)
  - Configuration of Virtual Environments (VEs) (contextualization, network...)
  - Images management/deployment
  - “Secure” accesses to the VEs



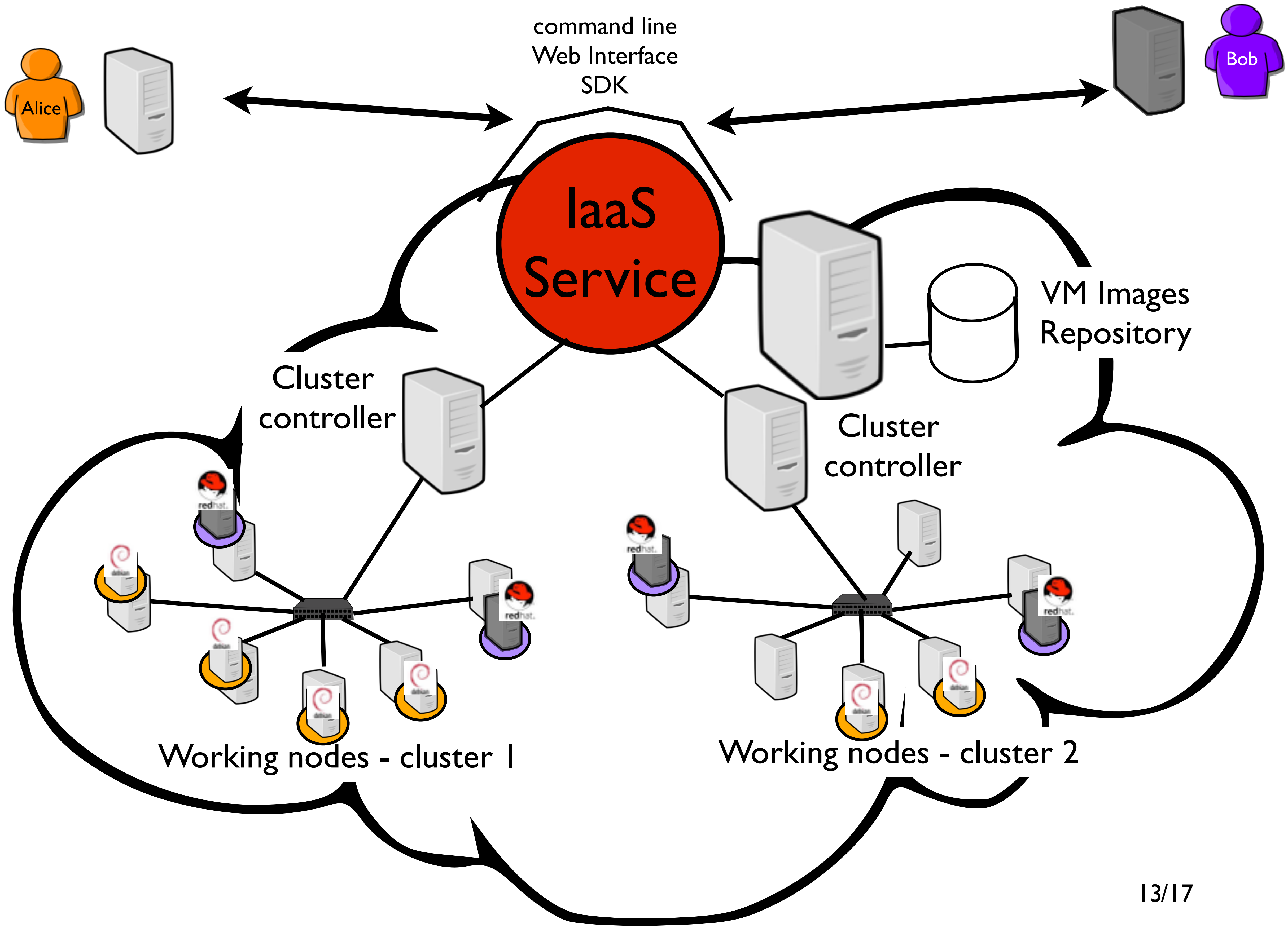
An Overview of the IaaS Internals



An Overview of the IaaS Internals



An Overview of the IaaS Internals



Managing IaaS - OpenSource solutions

- Open Nebula

  2008-20XX
  Results of the RESERVOIR project (mainly used in EU)
  Montero & Llorente, DSA-Research at UCM
  C++ / set of scripts

- CloudStack

  2010-20XX
  Apache project (in 2011)
  Java Based

- Open Stack

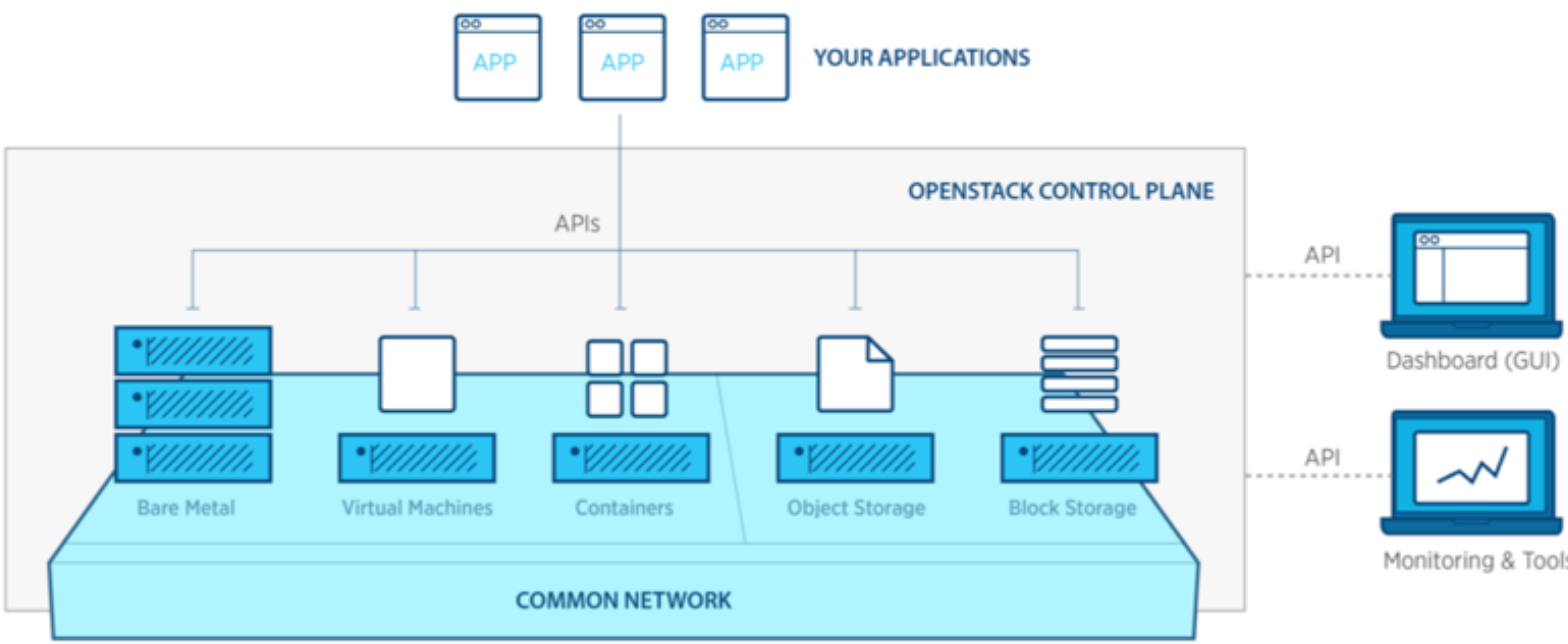
  2010-20XX
  Supported by several industrials
  The defacto open-source solution
  Python

You said OpenStack…

OpenStack

openstack.

One platform for bare metal, VMs and containers



OpenStack provides one platform to orchestrate bare metal, containers, and virtual machines on a single network, allowing private users to optimize for their application without creating more silos in their datacenters, and giving service providers more delivery options.

OpenStack


OPENSTACK MANAGES
5,000,000
CORES IN PRODUCTION


openstack.
50% of the FORTUNE 100
RUN ON OPENSTACK


What size organizations use OpenStack?
1% Do not know or not sure
9% 1 to 9 Employees
19% 10 to 99 Employees
21% 100 to 999 Employees
25% 1000 to 9,999 Employees
11% 10,000 to 99,999 Employees
14% 100,000 or more Employees

OpenStack

openstack.

More than 70,000 registered community members

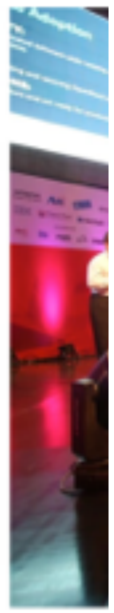
- 649 supporting organizations
- 181 countries represented
- 116 global user groups

OpenStack

openstack.

More than 70,000 registered community members

- 649 supporting organizations
- 181 countries represented
- 116 global user groups



openstack.

OpenStack users span industries

RETAIL/E-COMMERCE

Best Buy, Walmart, ebay, overstock

FINANCIAL

TD Bank, PayPal, FICO, American Express, Bloomberg

TELECOM

AT&T, NTT Communications, Deutsche Telekom, 中国移动 China Mobile, vodafone

ACADEMIC/RESEARCH

MIT Massachusetts Institute of Technology, CERN, Harvard University

ENERGY

State Grid Corporation of China

MANUFACTURING

VW, JFE

ENTERTAINMENT

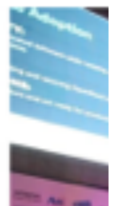
Comcast, betfair

OpenStack

openstack.

More than 70,000 registered community members

- 649 supporting organizations
- 181 countries represented



openstack.

OpenStack users span industries

COM

ACADEMIC/RESEARCH

AT&T

NTT Communications

Deutsche Telekom

中国移动 China Mobile

vodafone

Massachusetts Institute of Technology

CERN

HARVARD UNIVERSITY

ENTERTAINMENT

FE

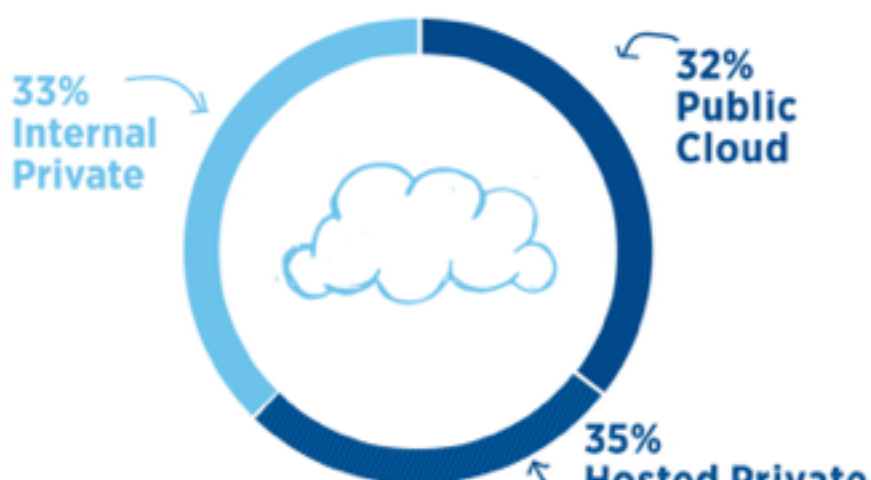
COMCAST

betfair

openstack.

Private clouds make up 2/3 of cloud platforms

Primary cloud computing platform:



Source: Forrester Research, 2017

OpenStack

openstack.

More than 70,000 registered community members

- 649 supporting organizations
- 181 countries represented

openstack.

OpenStack users span industries

COM

ACADEMIC/RESEARCH

AT&T

NTT Communications

Massachusetts Institute of Technology

CERN

openstack.

Private clouds make up 2/3 of cloud platforms

Primary cloud cc

Source: Forrester Research, 2017

openstack.

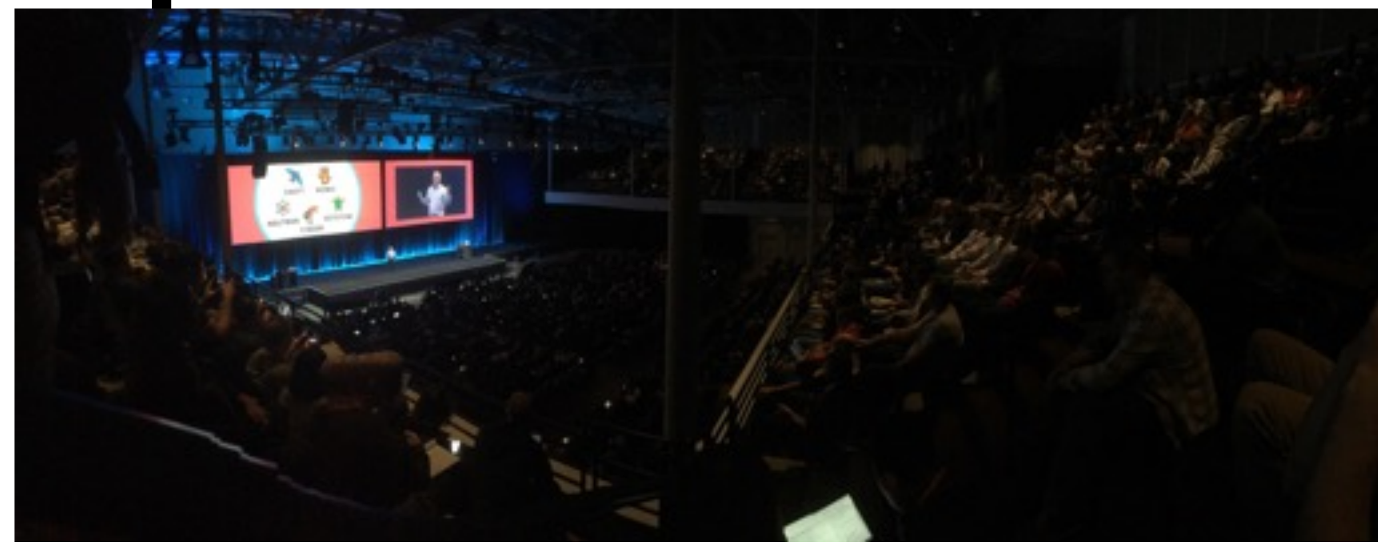
OpenStack
PROJECT
TEAMS
GATHERING

- The first OpenStack PTG was held in February in Atlanta, Georgia. Over 500 developers representing nearly 50 teams attended to collaborate and work on the OpenStack Pike release.
- The second PTG will be held in Denver, September 11-14. There are a few sponsorship opportunities available.

The OOO principles

- Open source
- Open development: access to each contribution/ logs of meetings,…
- Open design: the community is listened to set the direction of OpenStack
- Open Community: anyone can raise to leadership position

What is OpenStack?

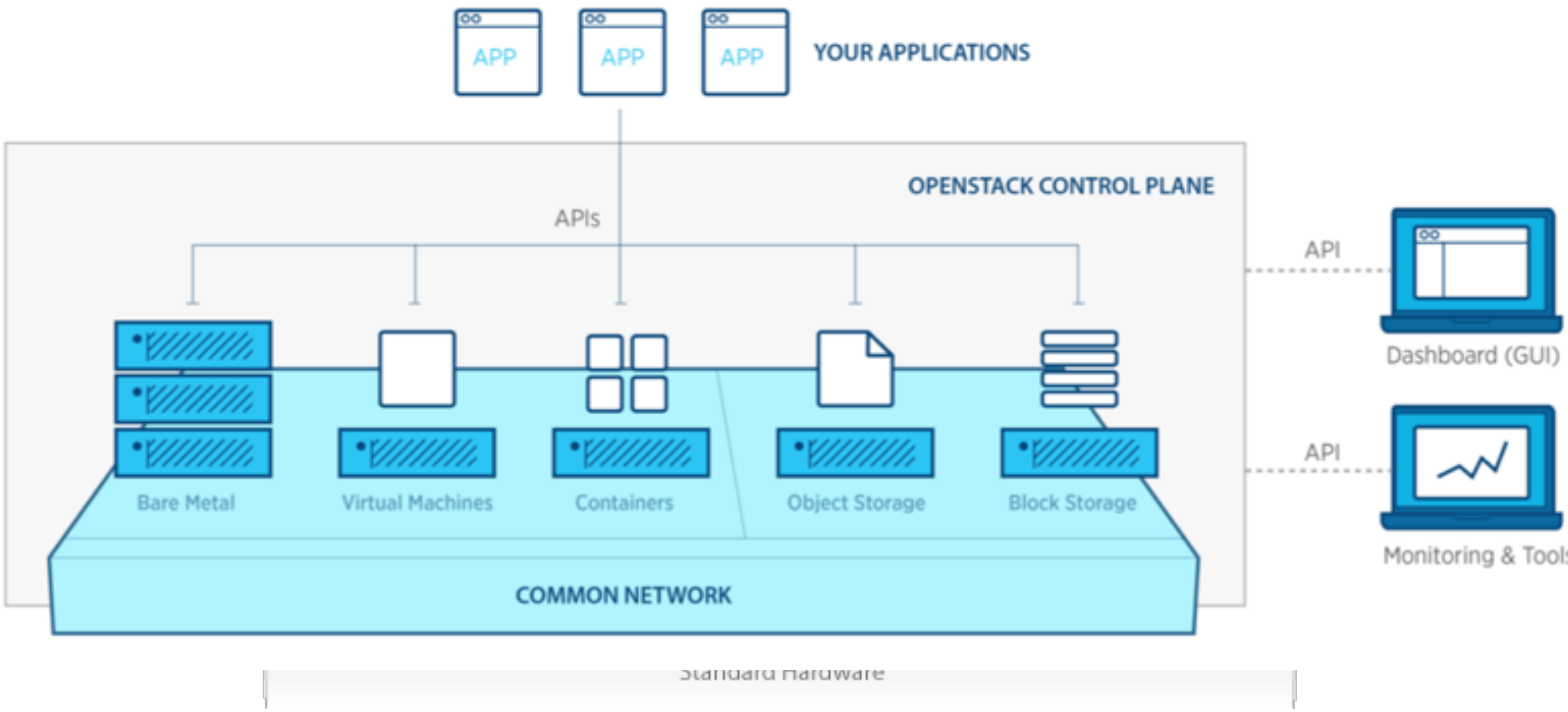


- an open-source project
- a common goal
- a coalition of organizations
- a foundation
- a trademark
- an interoperability standard
- a set of events (OpenStack Summit/PTG/OpenStack Days…)
- a governance model
- a job market
- a single project / a set of projects
- a set of principles
- a development community
- a big tent
- a bunch of python code
- a way to produce software
- a very active open-source project
- a success story
- ok so what it is….

A lot of Fun !

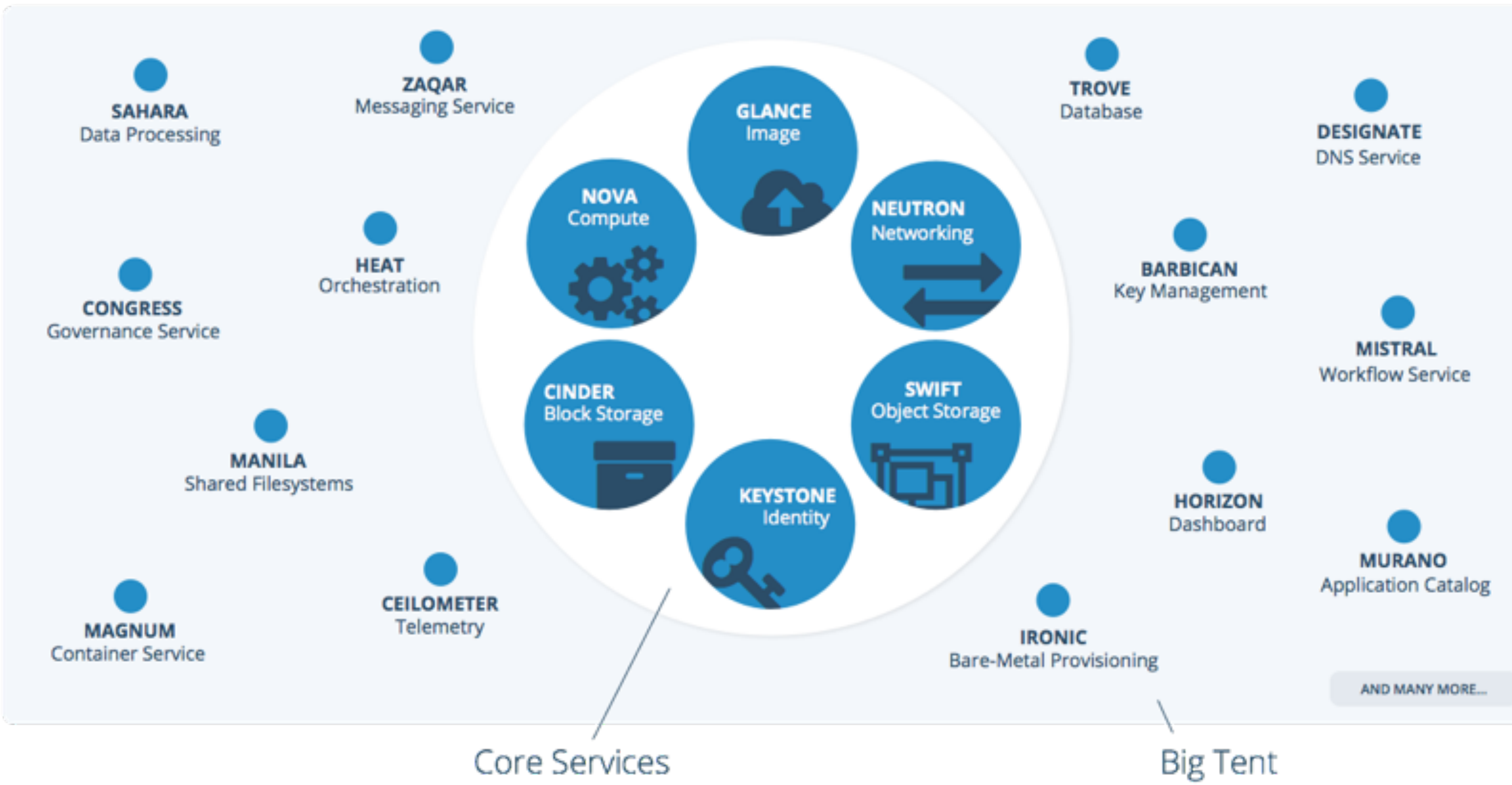
credits: D. Pertin FiCloud 2017 Presentation

A Rich (and Complex) Ecosystem



- **20 Millions of LoC**, 164 services, some services are composed of sub-services (e.g. nova-scheduler, nova-conductor, …)

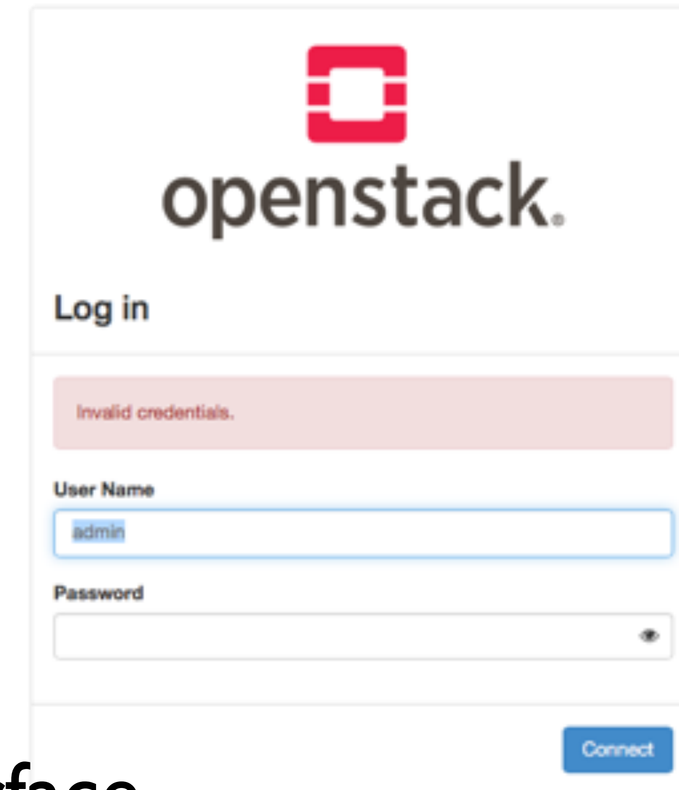
A Rich (and Complex) Ecosystem



- **20 Millions of LoC**, 164 services, some services are composed of sub-services (e.g. nova-scheduler, nova-conductor, …)

The User/Admin Viewpoints

- Everything goes through the API (including the HORIZON dashboard)



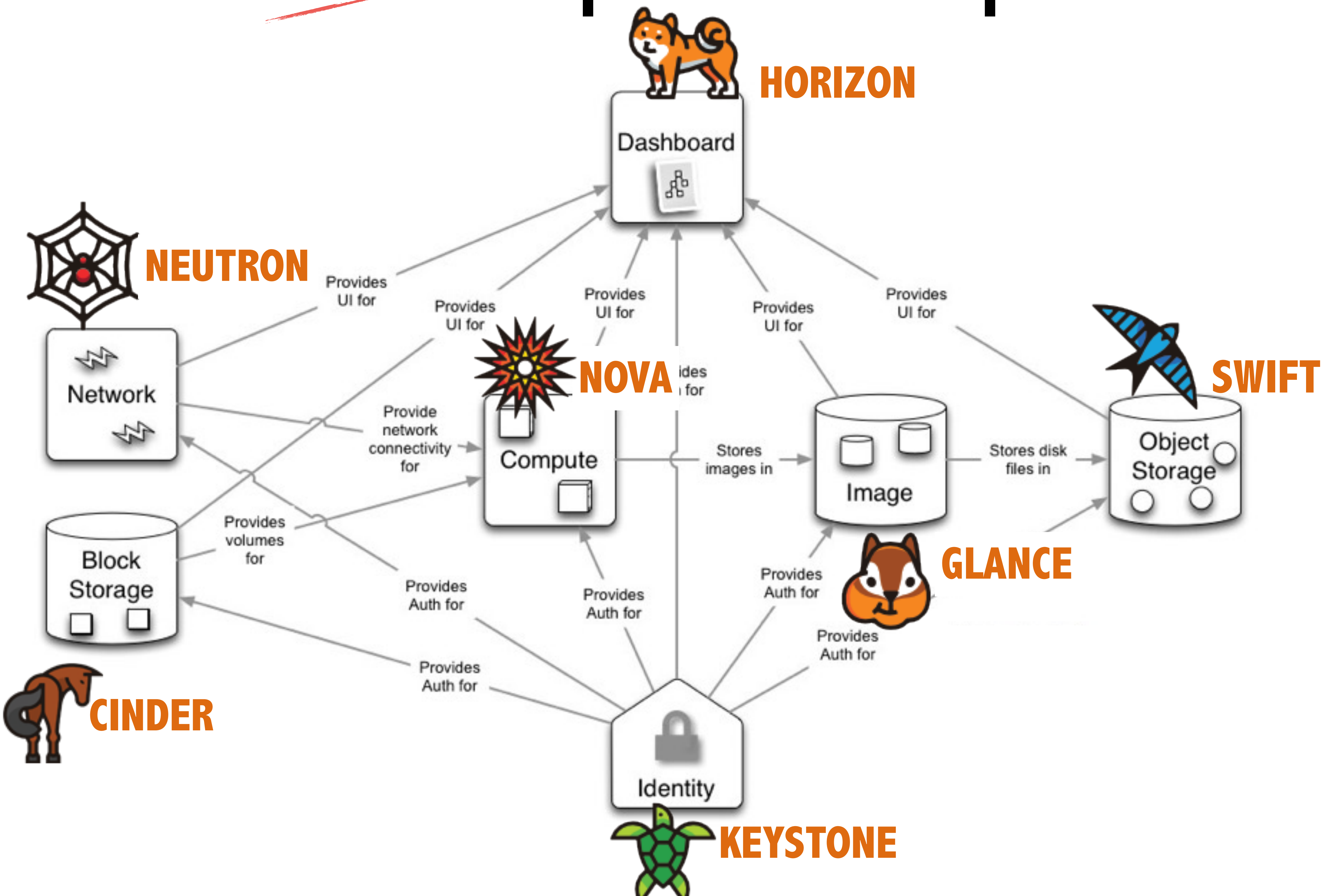
- APIs: REST / one per service

Through HTTP (curl)
Through SDKs and broker libraries
Through Horizon or the command line interface
Through HEAT

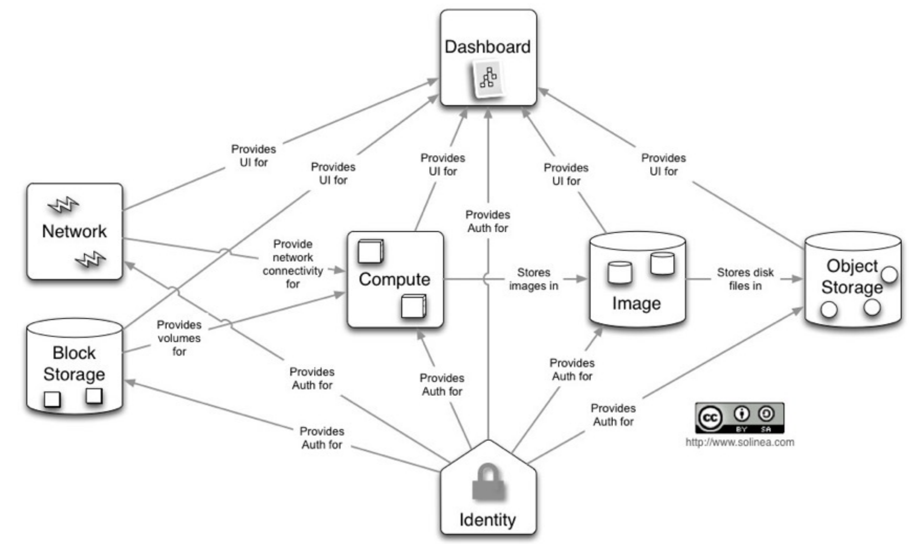
- You need specific credentials (delivered by KEYSTONE)

R-A Cherrueau / D. Pertin
Next Week !

The ~~Developer~~ Best Viewpoint



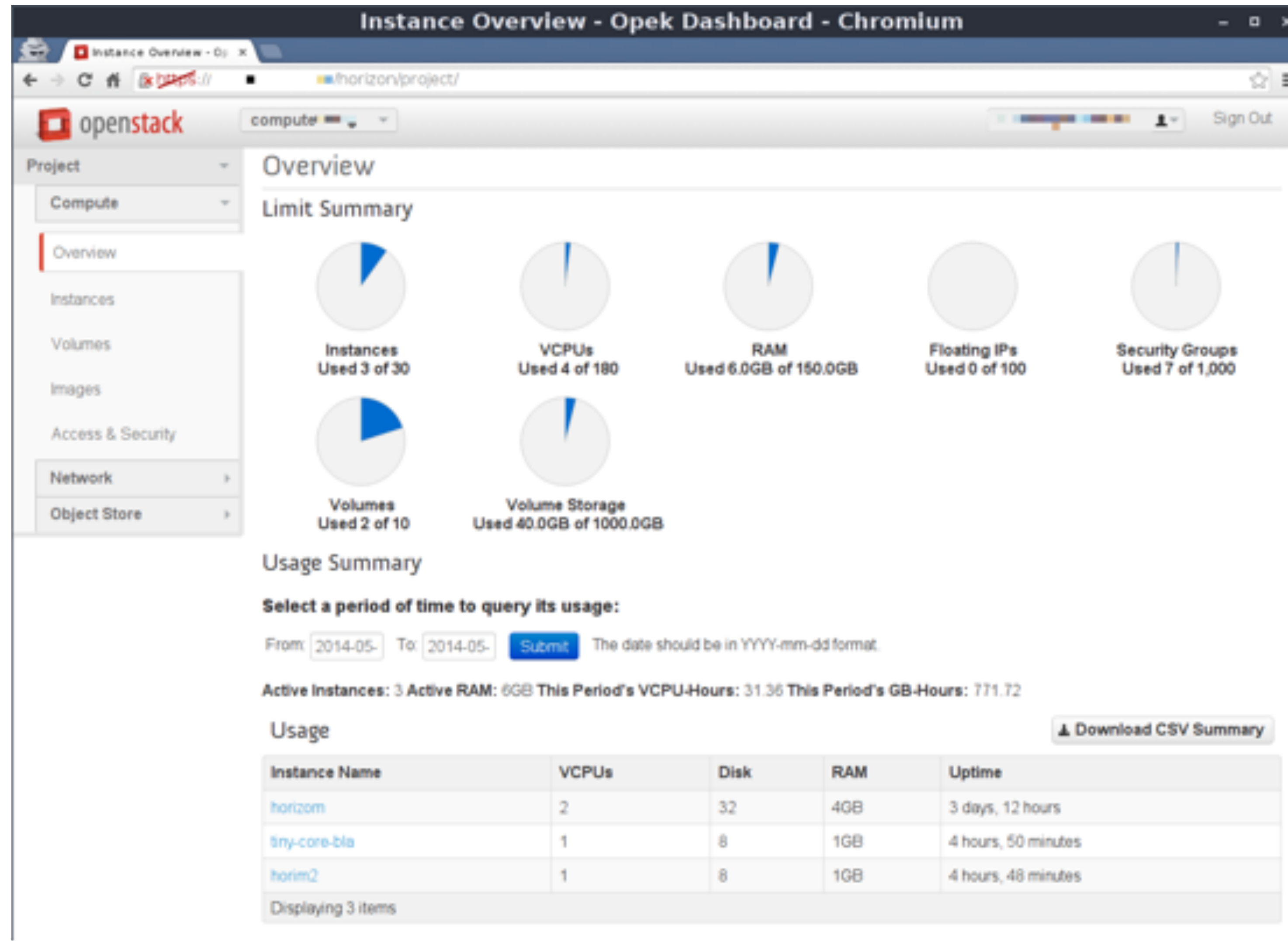
OpenStack core-services



- Each core-service is divided into several sub-services
- Services communicate through a communication bus (AMQP)
- System states are stored in a SQL DB (MySQL/MariaDB)
- Python for all projects
- APIs: OpenStack and AWS-like

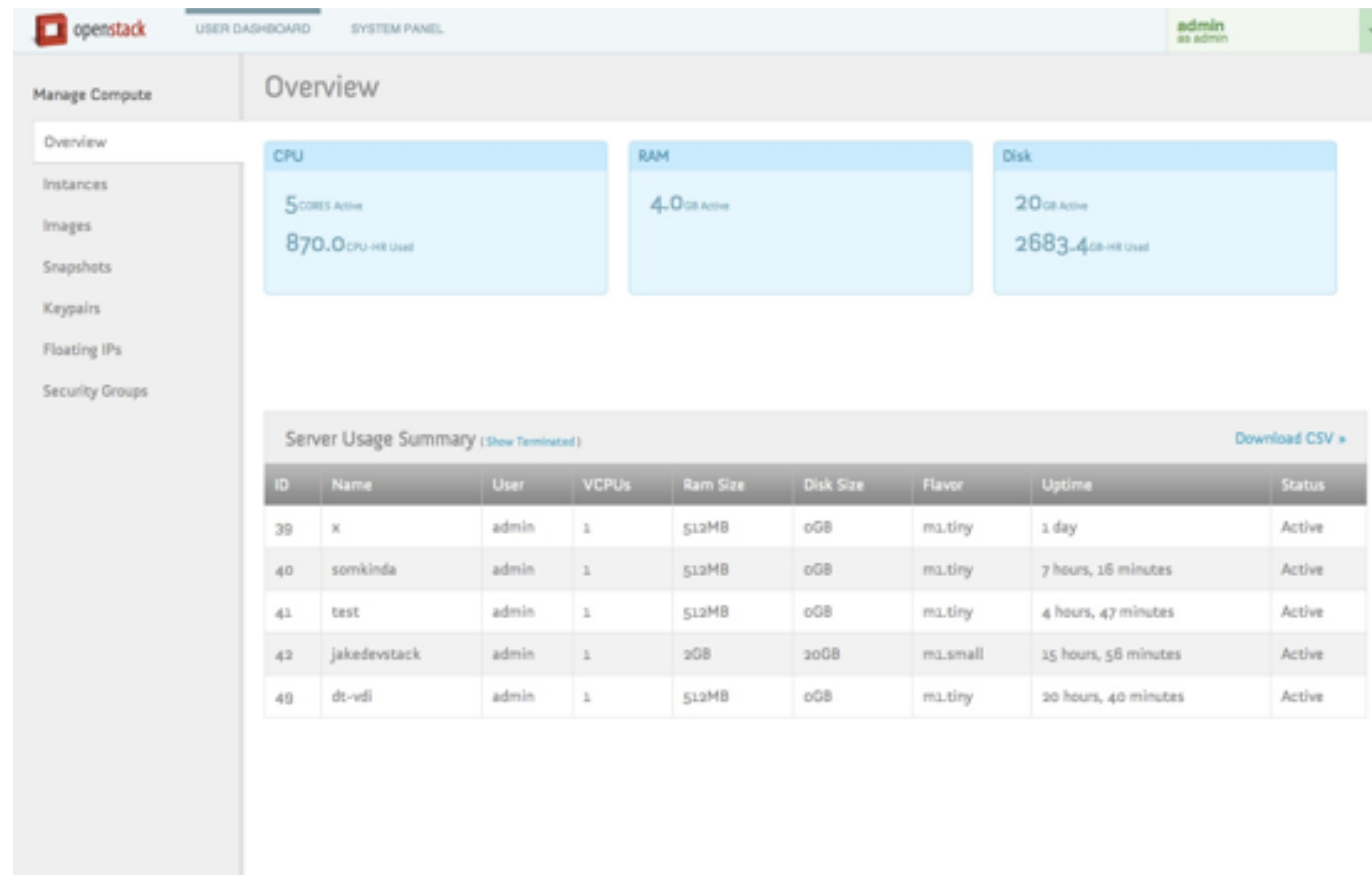
HORIZON Dashboard

- Provides a web based user interface to OpenStack services (a Django web application)
- Three central dashboards, a “User Dashboard”, a “System Dashboard”, and a “Settings” dashboard.



HORIZON Dashboard

- Provides a web based user interface to OpenStack services (a Django web application)
- Three central dashboards, a "User Dashboard", a "System Dashboard", and a "Settings" dashboard.



KEYSTONE Authentification

- Keystone provides API client authentication, service discovery, and distributed multi-tenant authorization by implementing OpenStack's Identity API.
- It supports LDAP, OAuth, OpenID Connect, SAML and SQL.PIs: REST / one per service

Nova Compute Service

- Implement services and associated libraries to provide massively scalable, on demand, self service access to compute instances
- Nova supports creating virtual machines and baremetal servers, through the use of IRONIC.

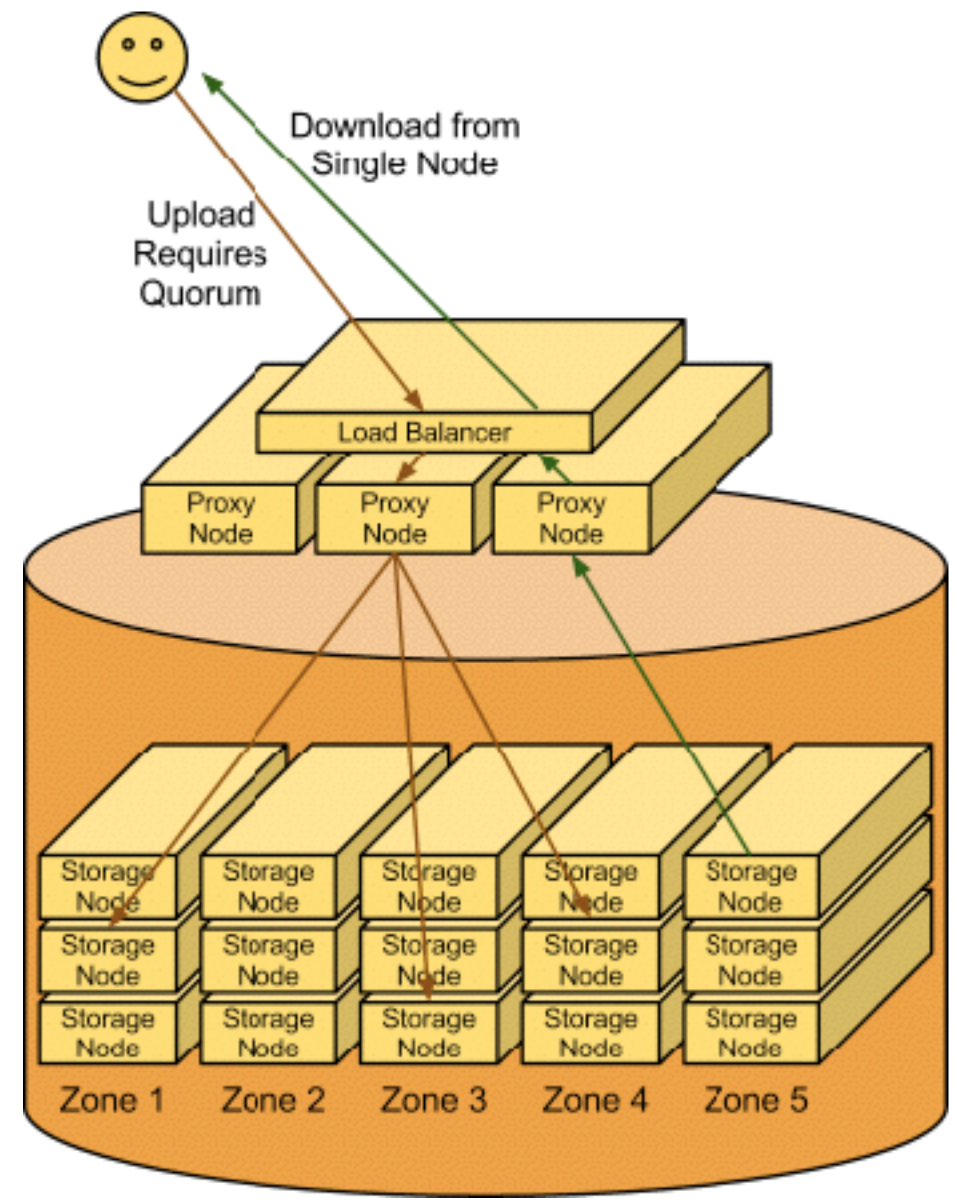
More details later…

Glance Image Service

- VM images contain a virtual disk that holds a bootable operating system on it.
- Each launched instance runs from a copy of the base image. Any changes made to the instance do not affect the base image. Snapshots capture the state of an instances running disk.
- Users can create a snapshot, and build a new image based on these snapshots
- Glance has a RESTful API that allows querying of VM image metadata as well as retrieval of the actual image.
- VM images made available through Glance can be stored in a variety of locations from simple filesystems to object-storage systems like the OpenStack Swift project.

Swift Object Store

- Swift is a highly available, distributed, eventually consistent object/blob store (a S3-like system)
- Organizations can use Swift to store lots of data efficiently, safely, and cheaply. It's built for scale and optimized for durability, availability, and concurrency across the entire data set.
- Swift is ideal for storing unstructured data that can grow without bound.
- The only service that does not leverage a central DB



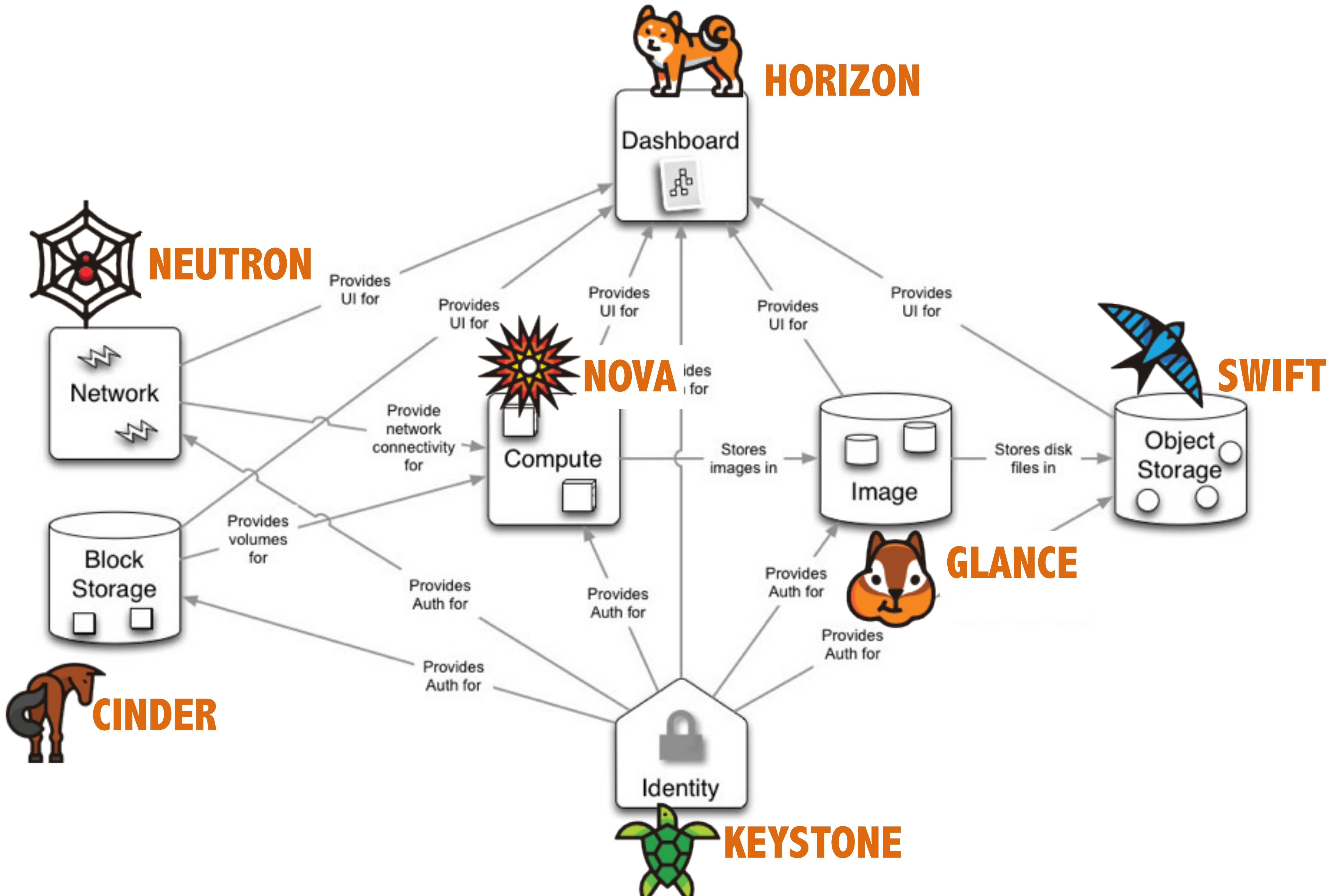
Cinder Block Storage

- You can add and remove additional resources from running instances, such as persistent volume storage
- The Cinder-volume service provides persistent block storage, instead of the ephemeral storage provided by the base image (i.e. the Glance one)
- Cinder virtualizes the management of block storage devices and provides end users with a self service API to request and consume those resources without requiring any knowledge of where their storage is actually deployed or on what type of device.
- This is done through the use of either a reference implementation (LVM) or plugin drivers for other storage.

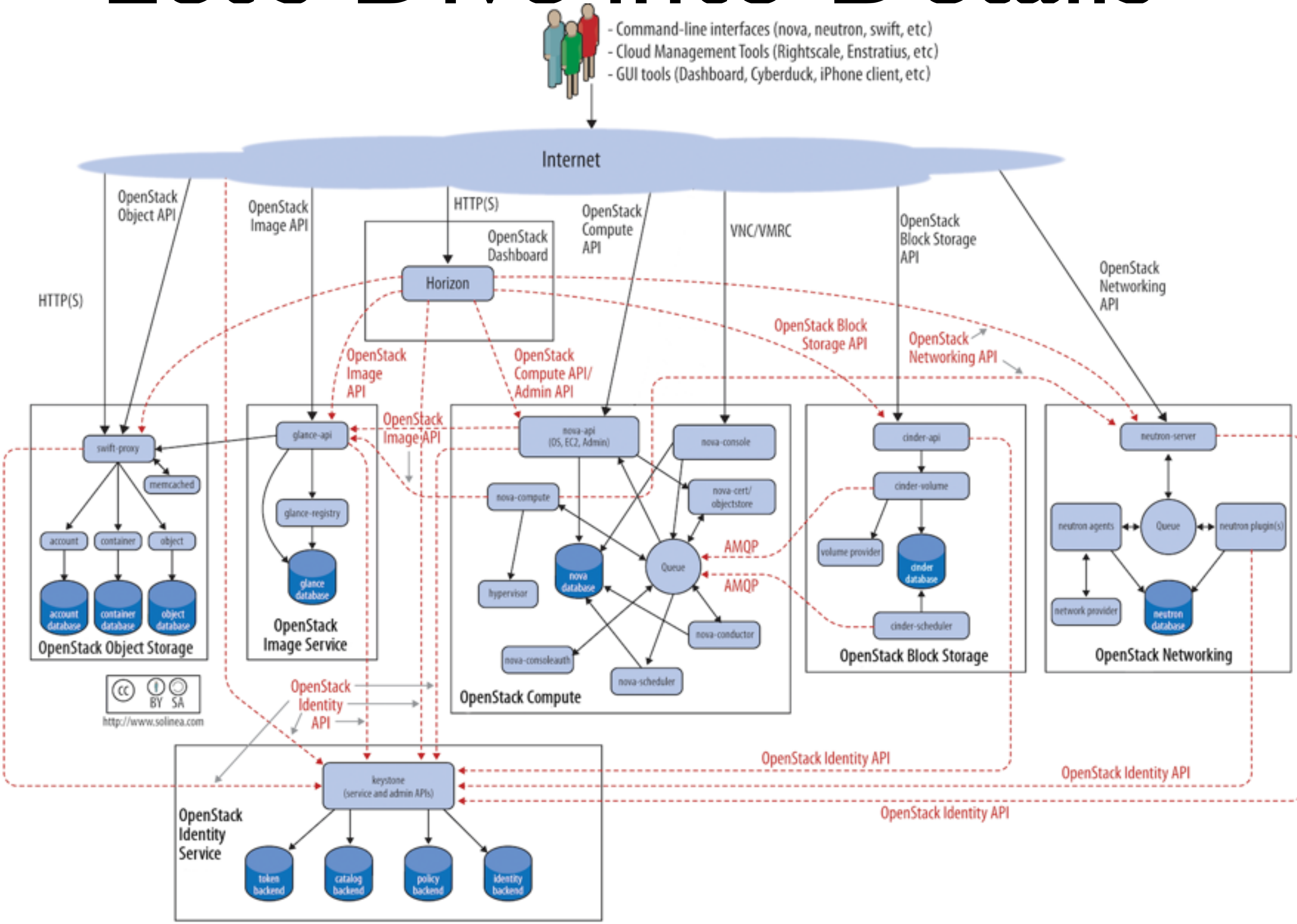
Neutron Networking

- OpenStack Neutron is an SDN networking project focused on delivering networking-as-a-service (NaaS) in virtual compute environments.
- Composed of several sub-services
  Neutron-server: API service
  Agent DHCP: DHCP service for instances
  Agent L3 : Routing service
- In addition to elementary services (L2/L3), Neutron provides additional mechanisms (load-balancing, firewalls, VPN…)
- A lot of plugins (LinuxBridge default one)

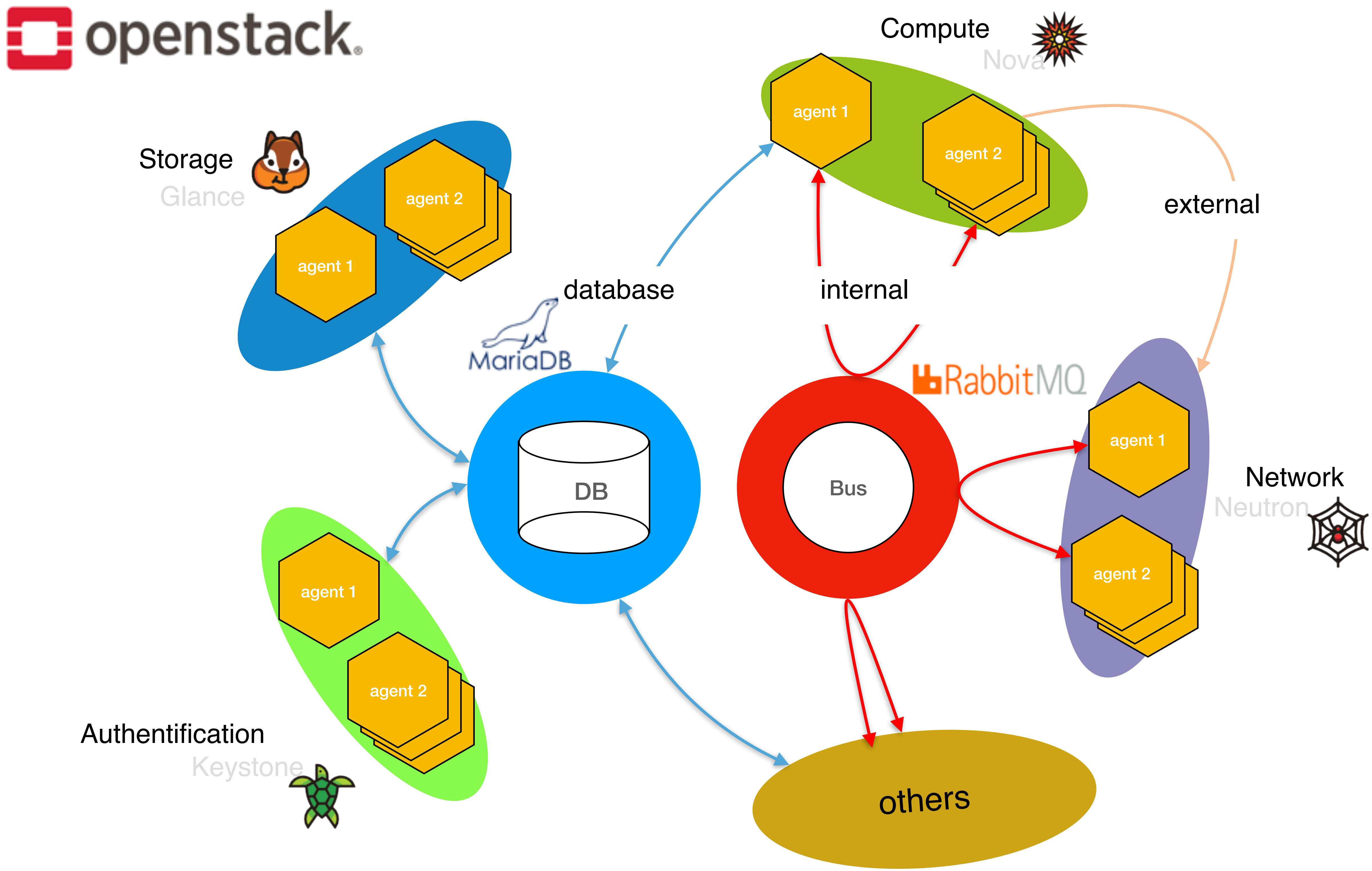
Let's Dive into Details



Let’s Dive into Details

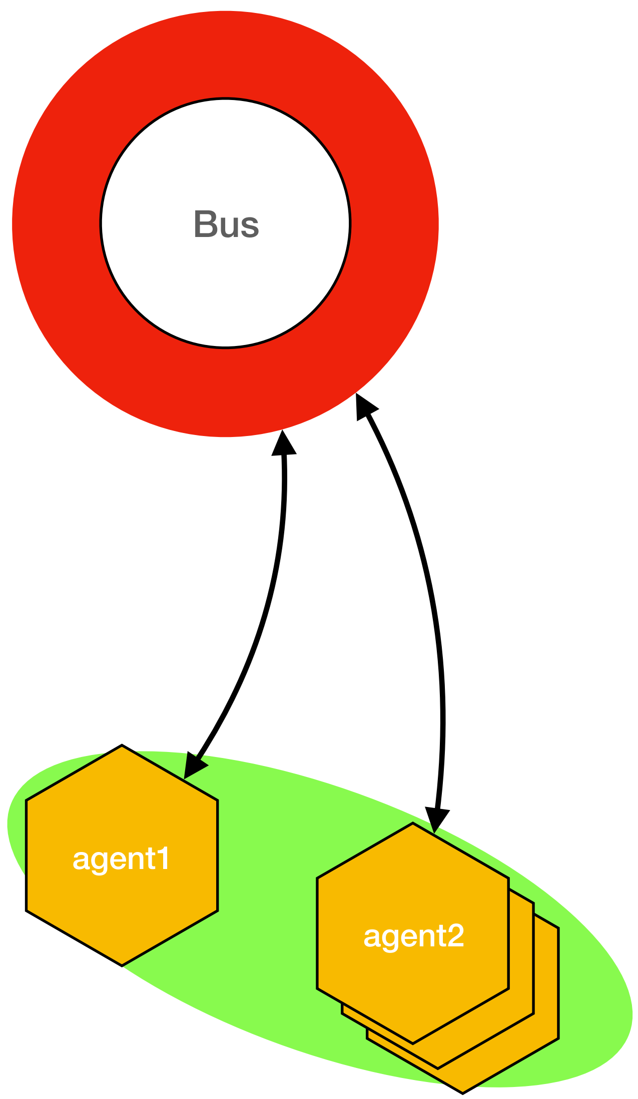


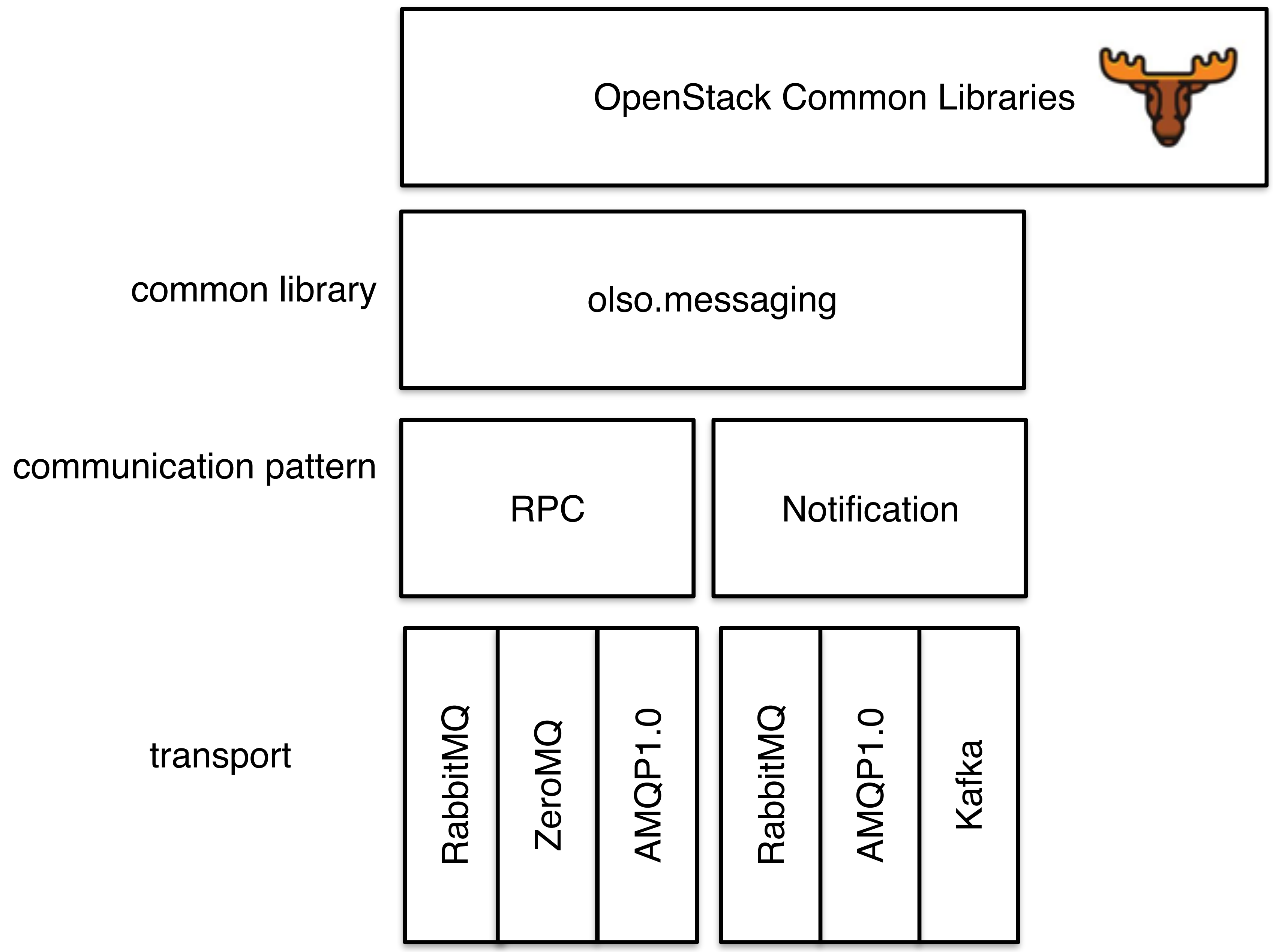
DBs/Communication Bus



credits: M. Simonin - IPL Discovery

DBs/Communication Bus

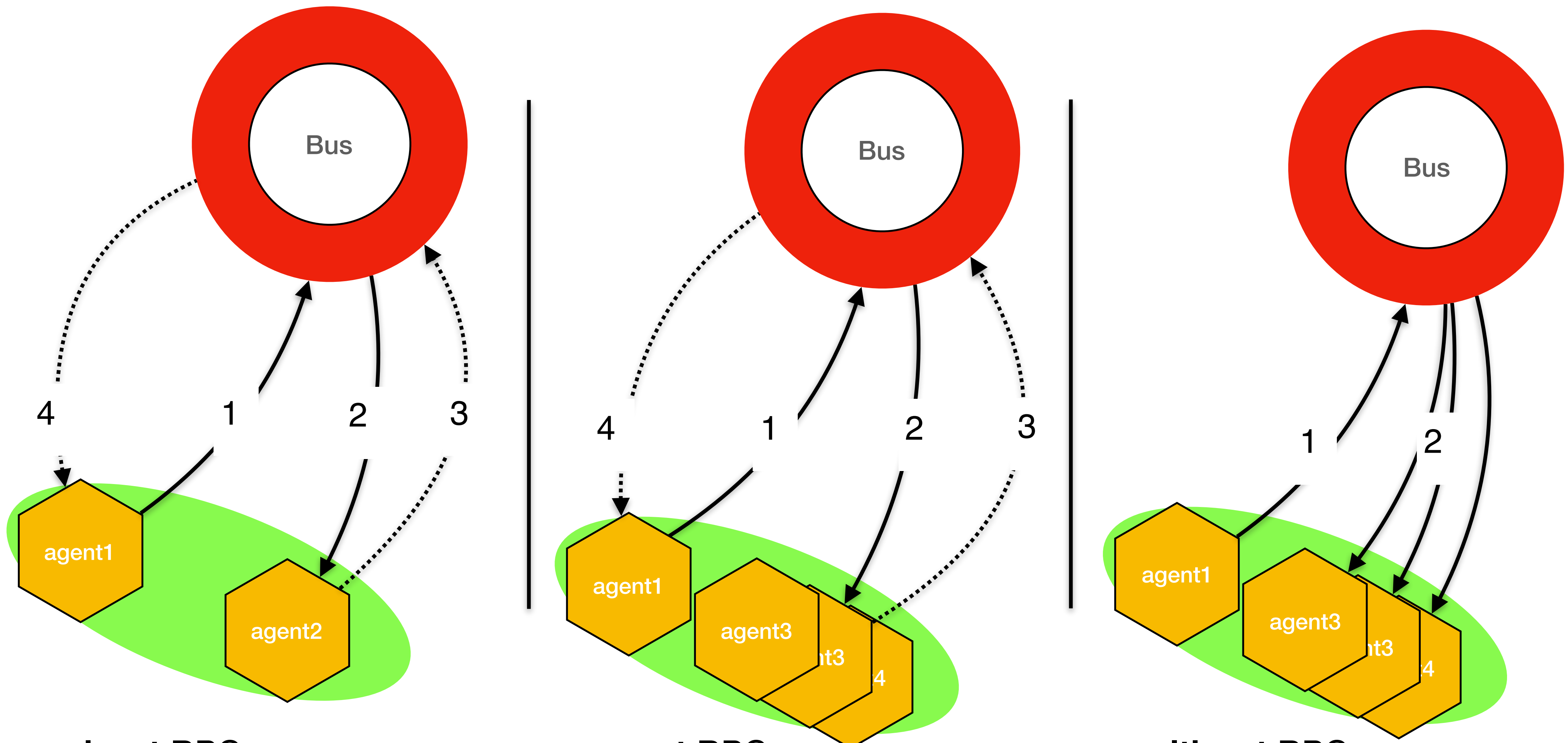




credits: M. Simonin - IPL Discovery

DBs/Communication Bus

RabbitMQ



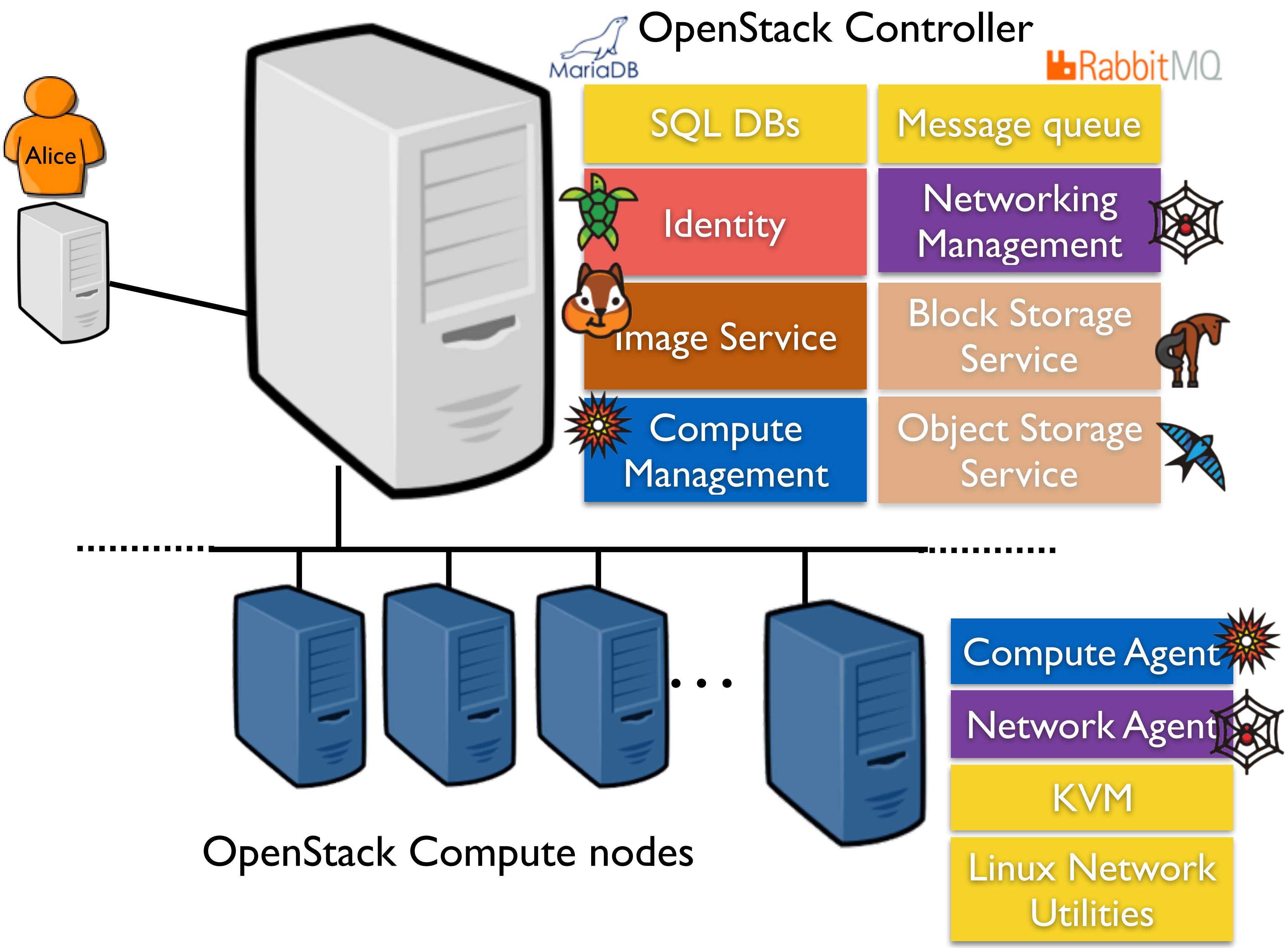
unicast RPC
blocking & non blocking

anycast RPC
blocking & non blocking

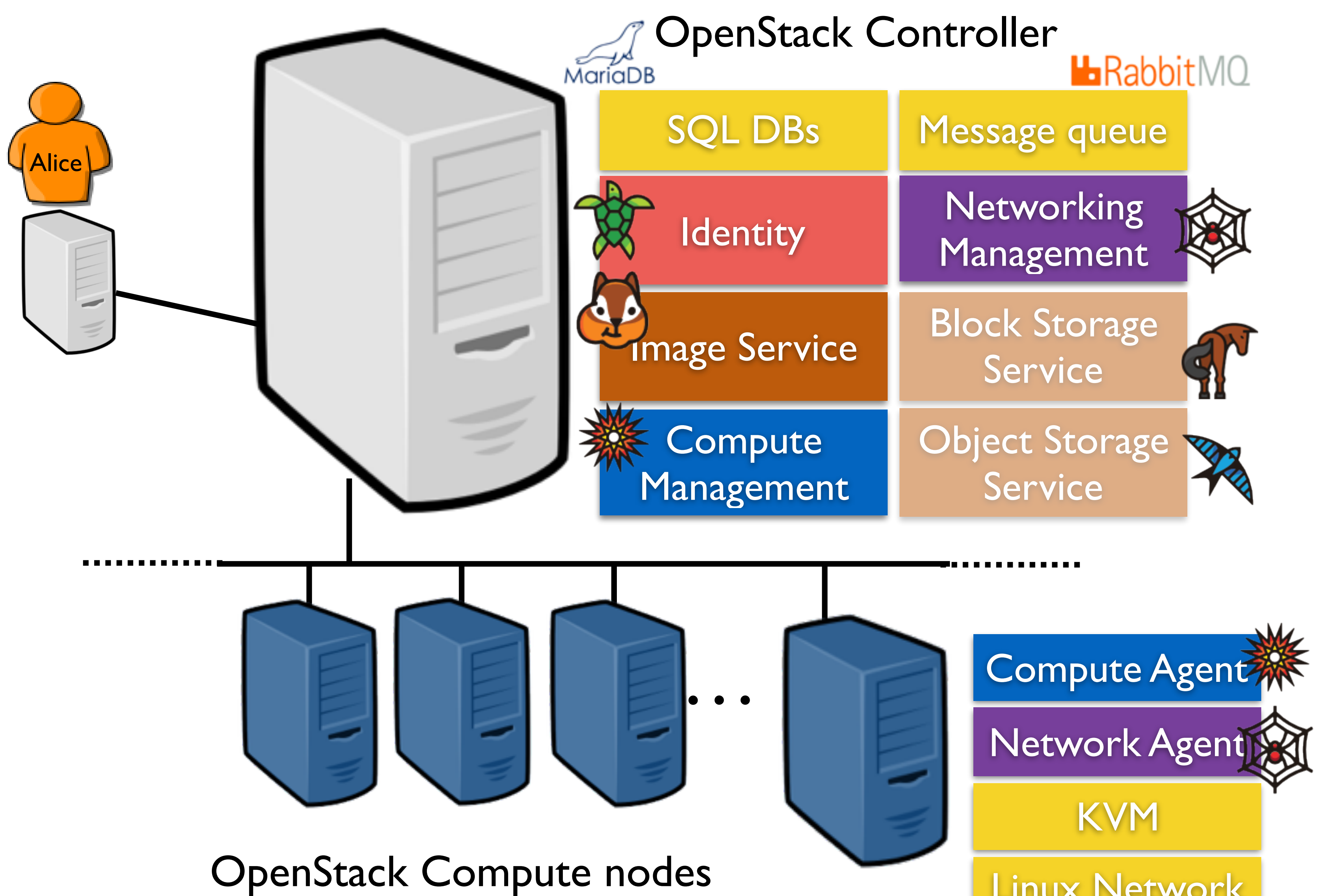
multicast RPC
non blocking only

credits: M. Simonin - IPL Discovery

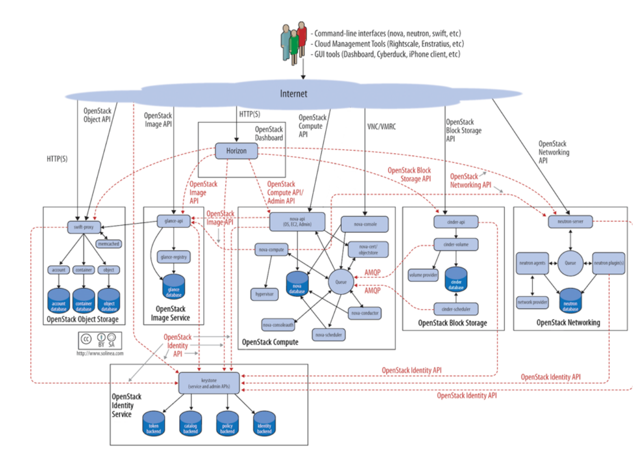
Physically, How This Looks Like?



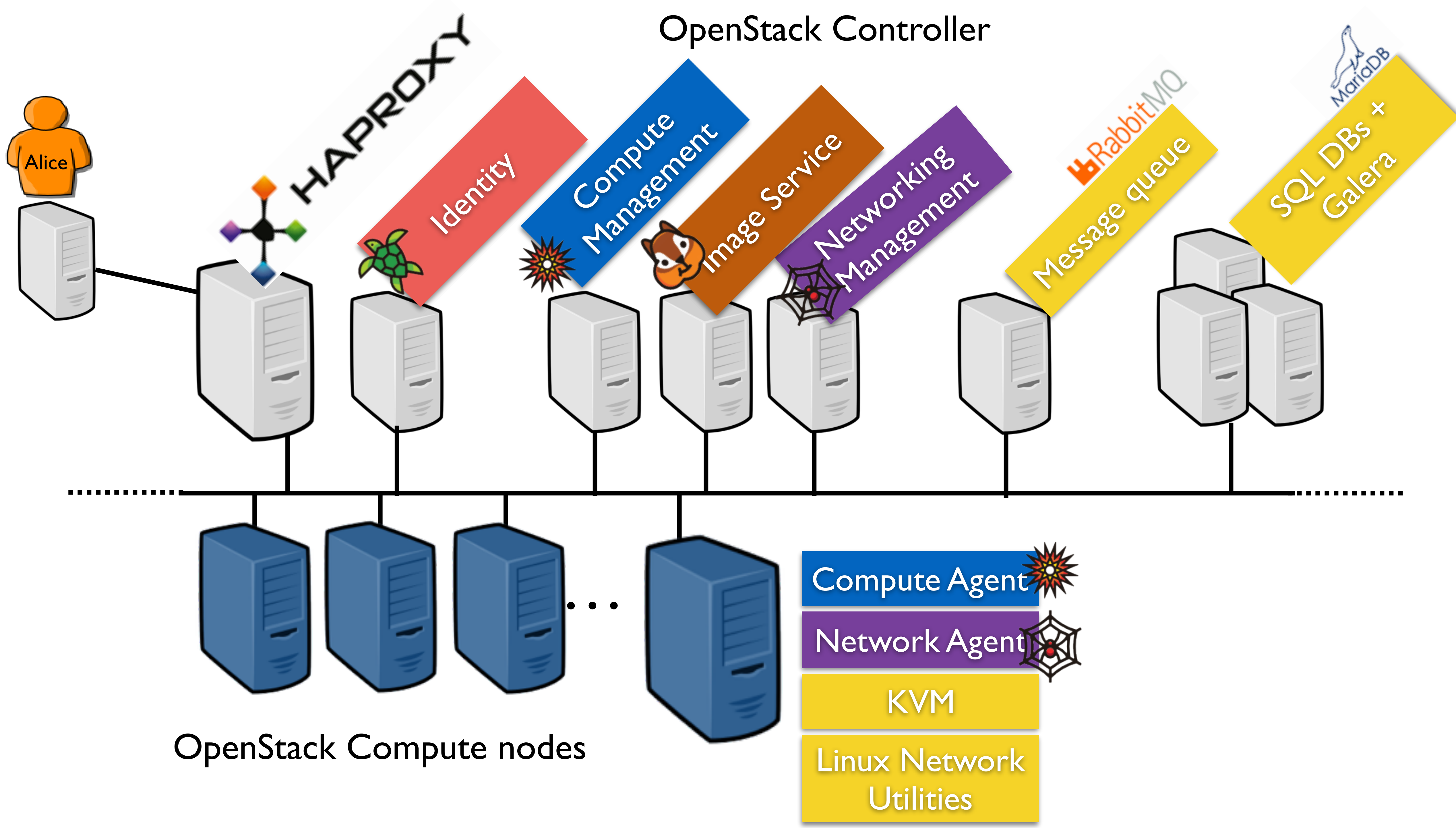
Physically, How This Looks Like?



A First Simplified View



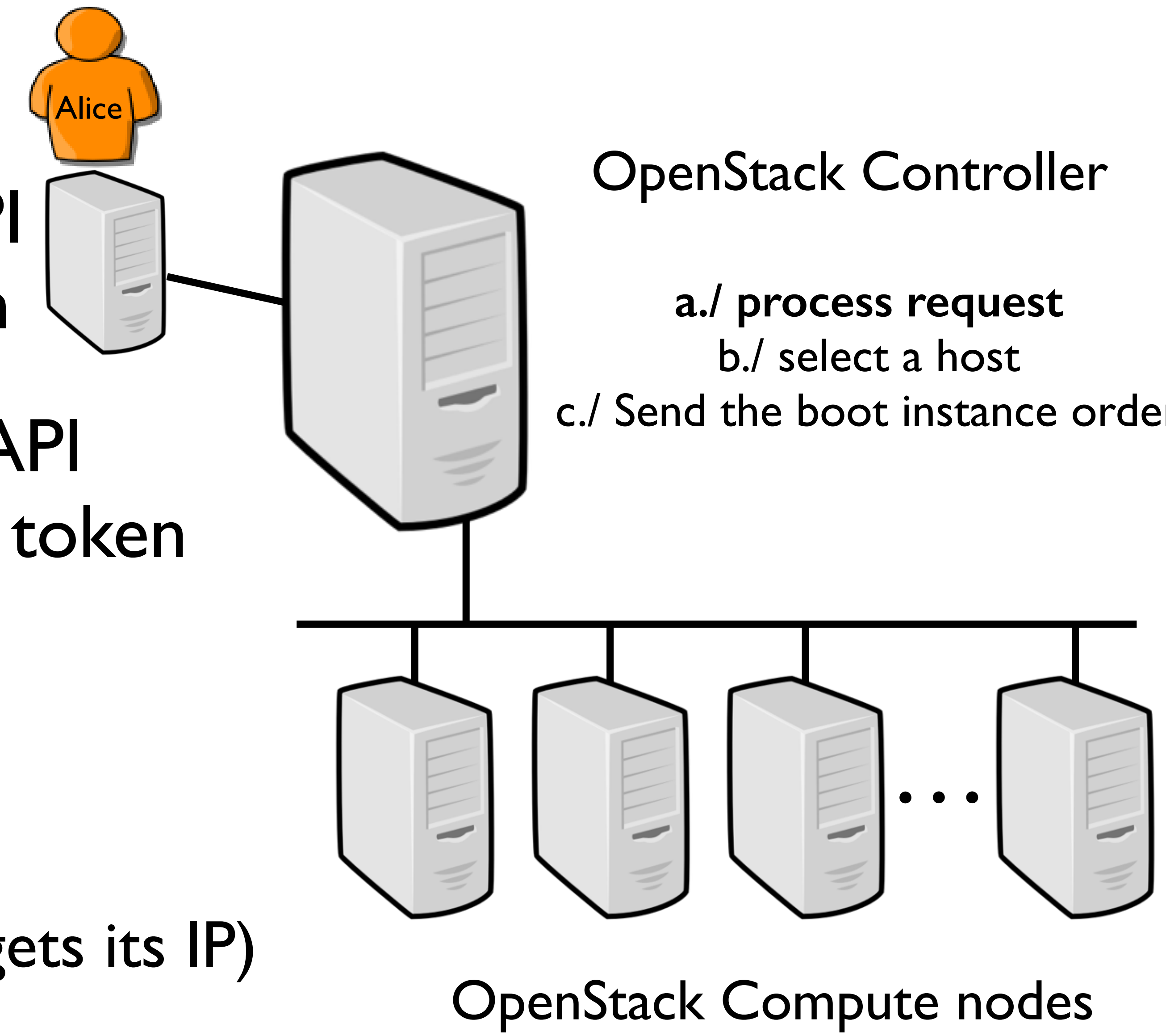
Physically, How This Looks Like?



Walkthrough of a typical Nova boot request

- Make a request to the Identity API (Keystone) to get an access token
- Make a request to the Compute API (Nova) using the aforementioned token and mentioning

  The flavor (VM properties)
  The image (a bootable image)
  The network ID (i.e. where the VM gets its IP)




```
vagrant@enos-node:/opt/enos$ openstack server create\
  --flavor m1.tiny\
  --image cirros.uec\
  --nic net-id=$(openstack network show private --column id --format value)\
  cli-vm
```

credits: J. Pipes - https://github.com/jaypipes/articles/blob/master/openstack/walkthrough-launch-instance-request.md

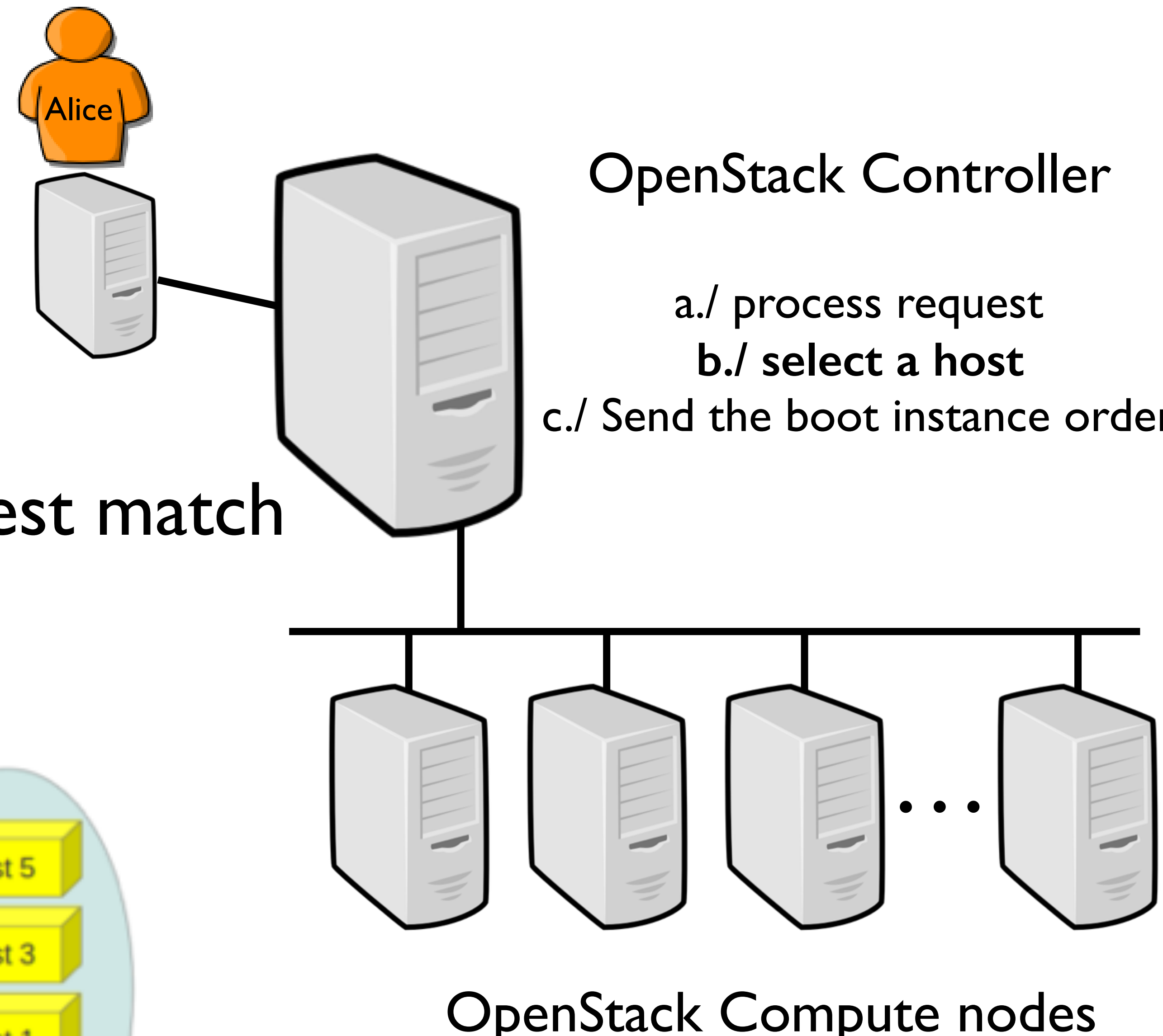
Walkthrough of a typical Nova boot request

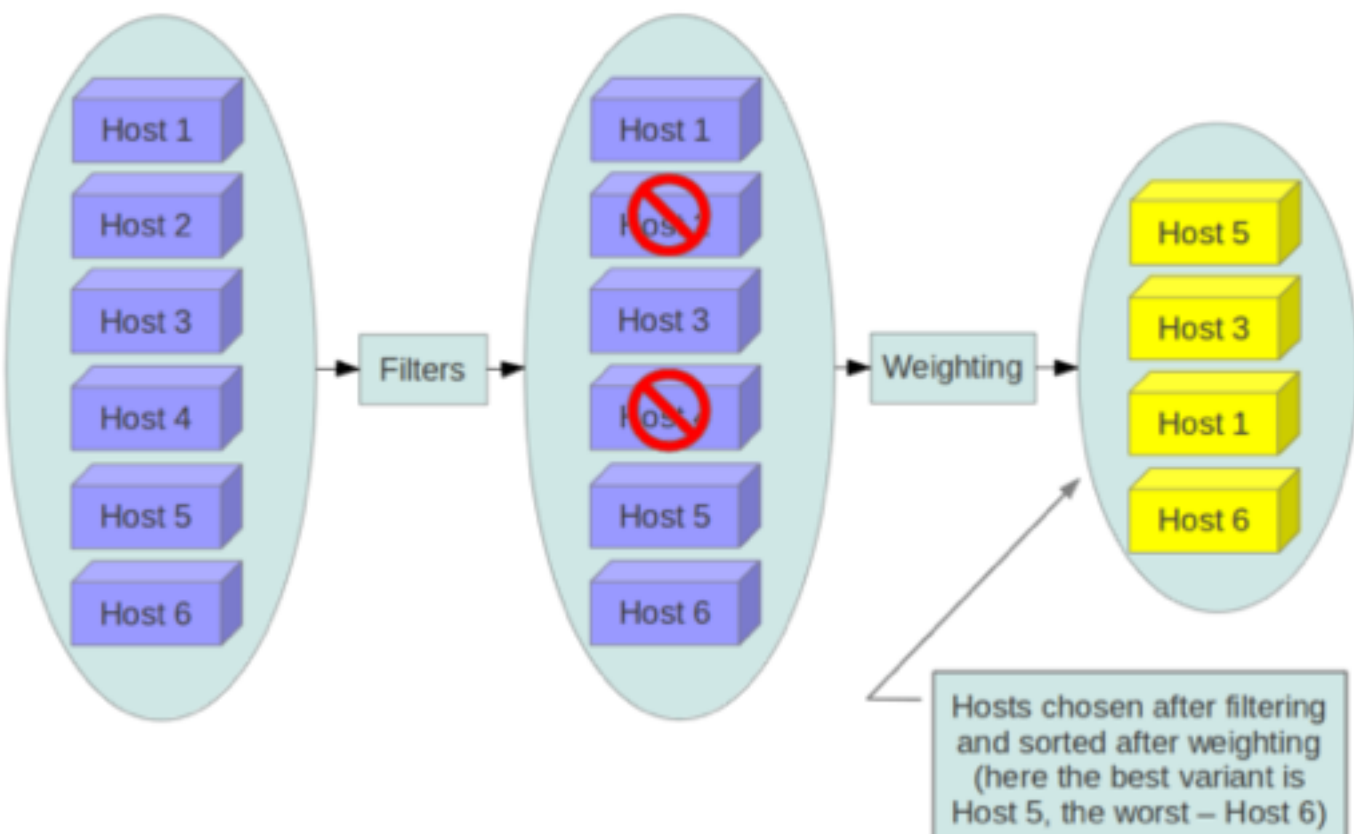
- Select a host (scheduler)

  Ask Placement for possible hosts (placement API)

  Apply filters

  Sort hosts with weighers and take best match



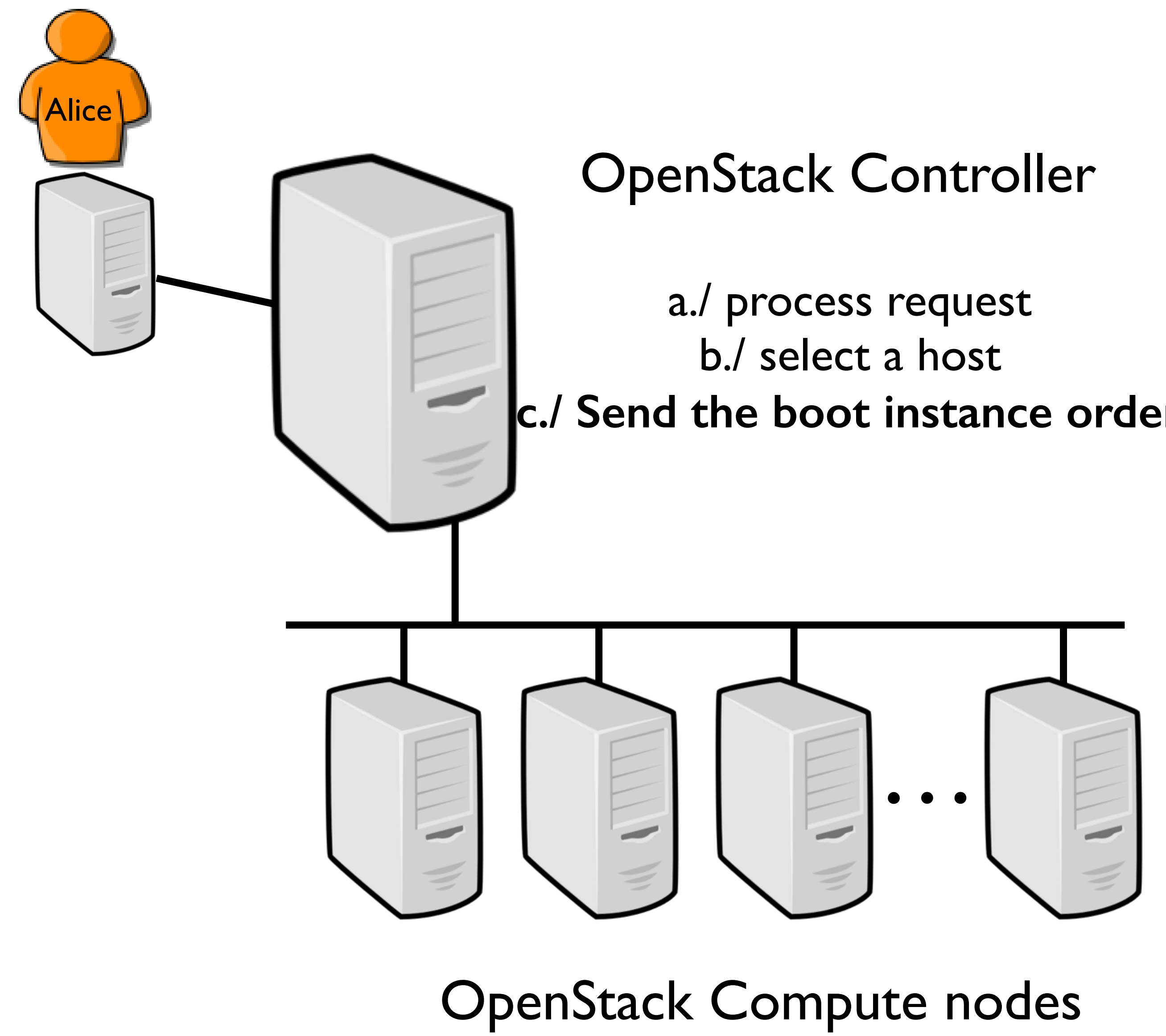


credits: J. Pipes - https://github.com/jaypipes/articles/blob/master/openstack/walkthrough-launch-instance-request.md

Walkthrough of a typical Nova boot request

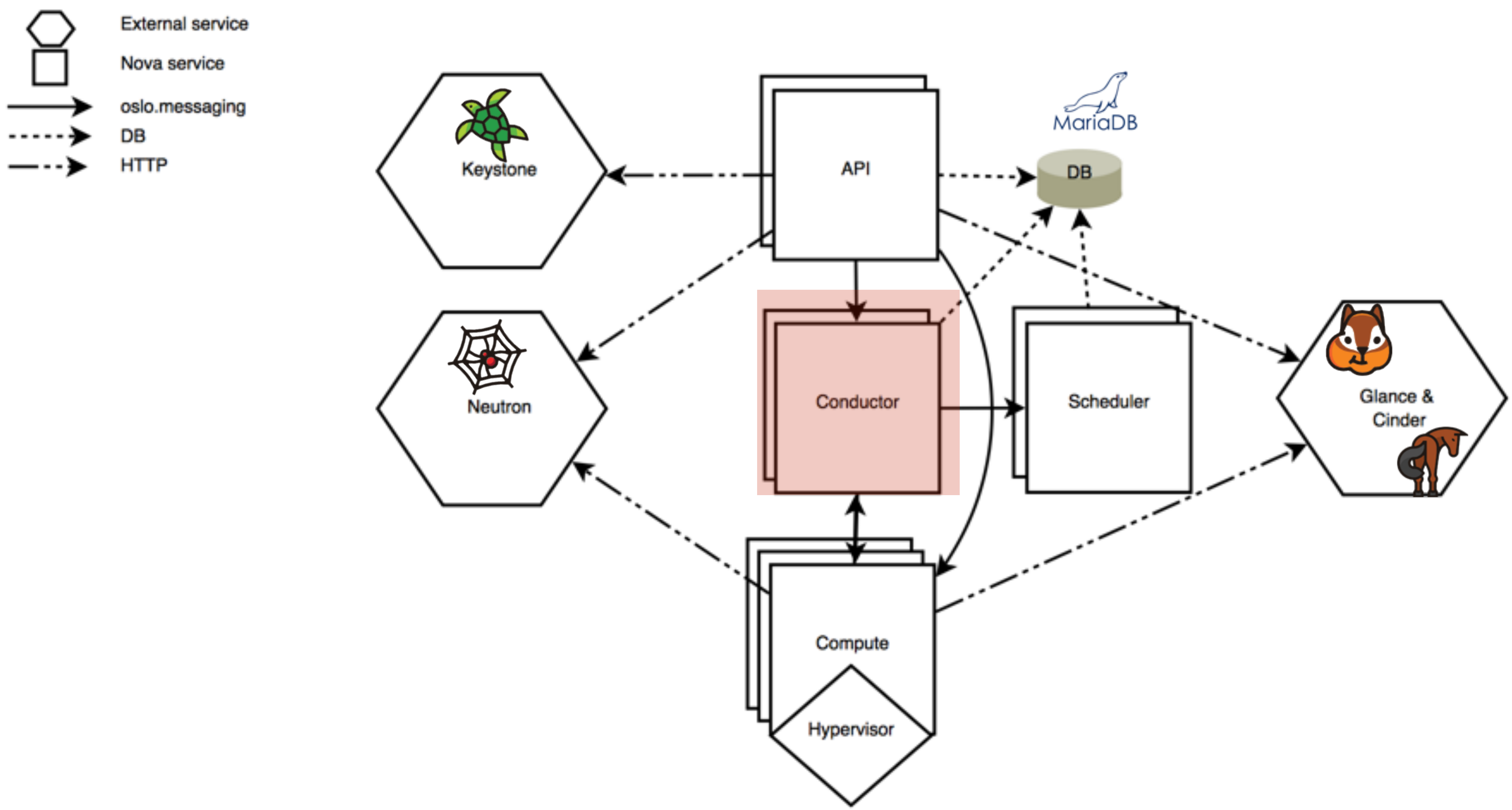
- When the compute node receives the request :

  Create a port from NEUTRON
  Prepare the corresponding block device
  Ask the hypervisor to boot the VM



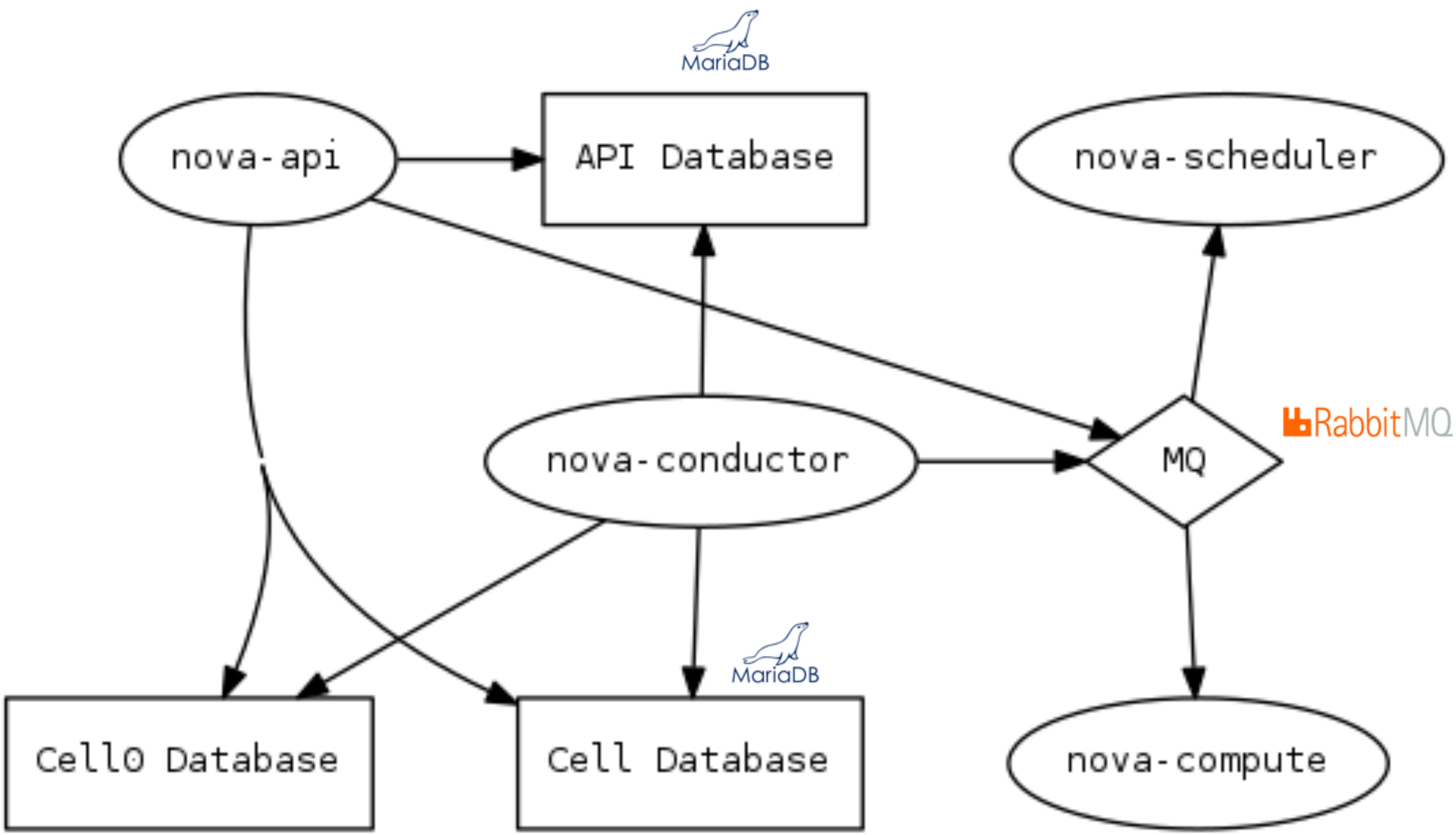
credits: J. Pipes - https://github.com/jaypipes/articles/blob/master/openstack/walkthrough-launch-instance-request.md

Focus on Nova Service



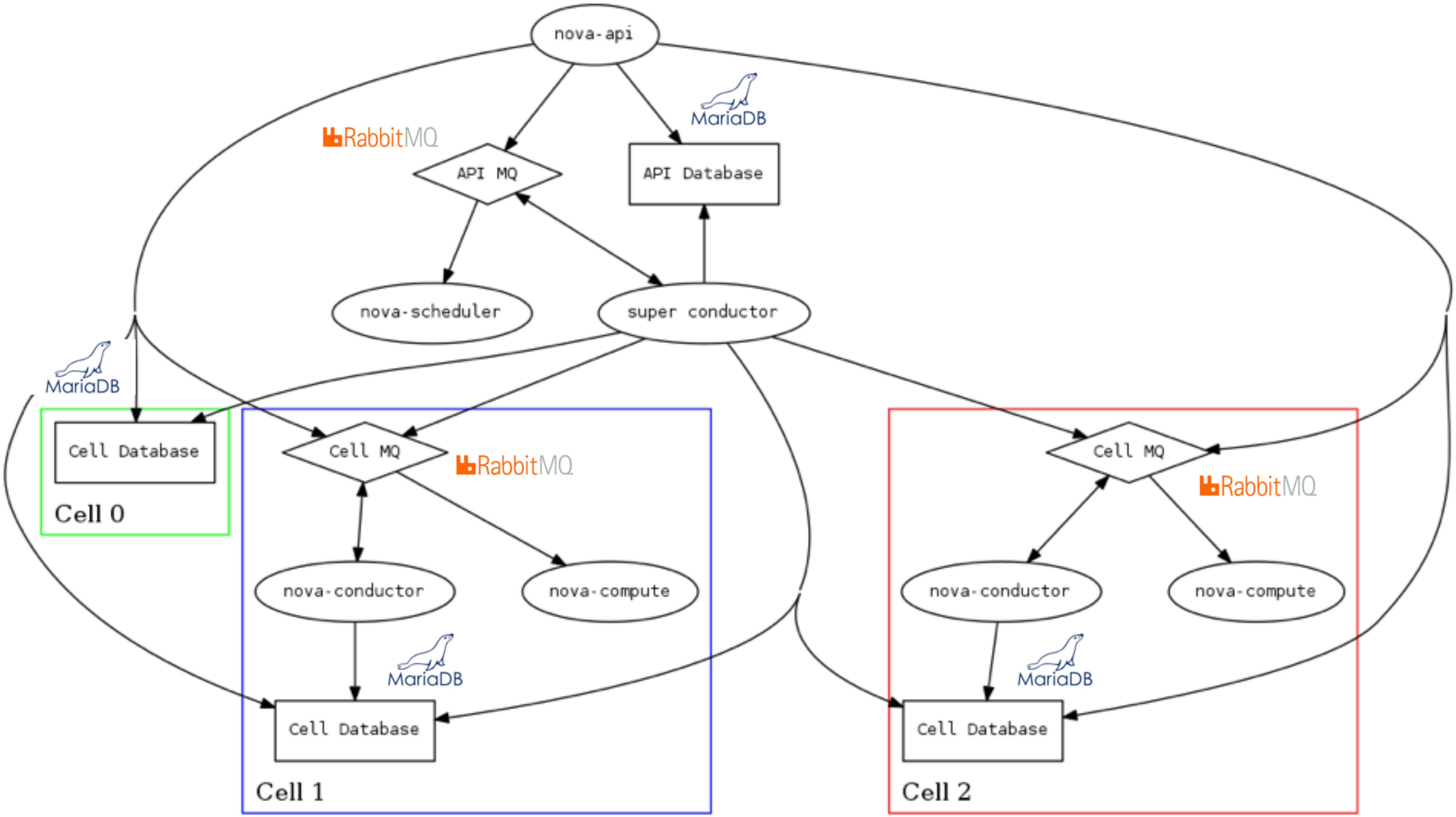
Nova Conductor: a key element

Focus on Nova Service



Cell... by default since Ocata

Focus on Nova Service



Cell...a way to segregate your infrastructure

Focus on Neutron

- Different kinds of networks

  Project networks provide connectivity to instances for a particular project/tenant (Private IPs)

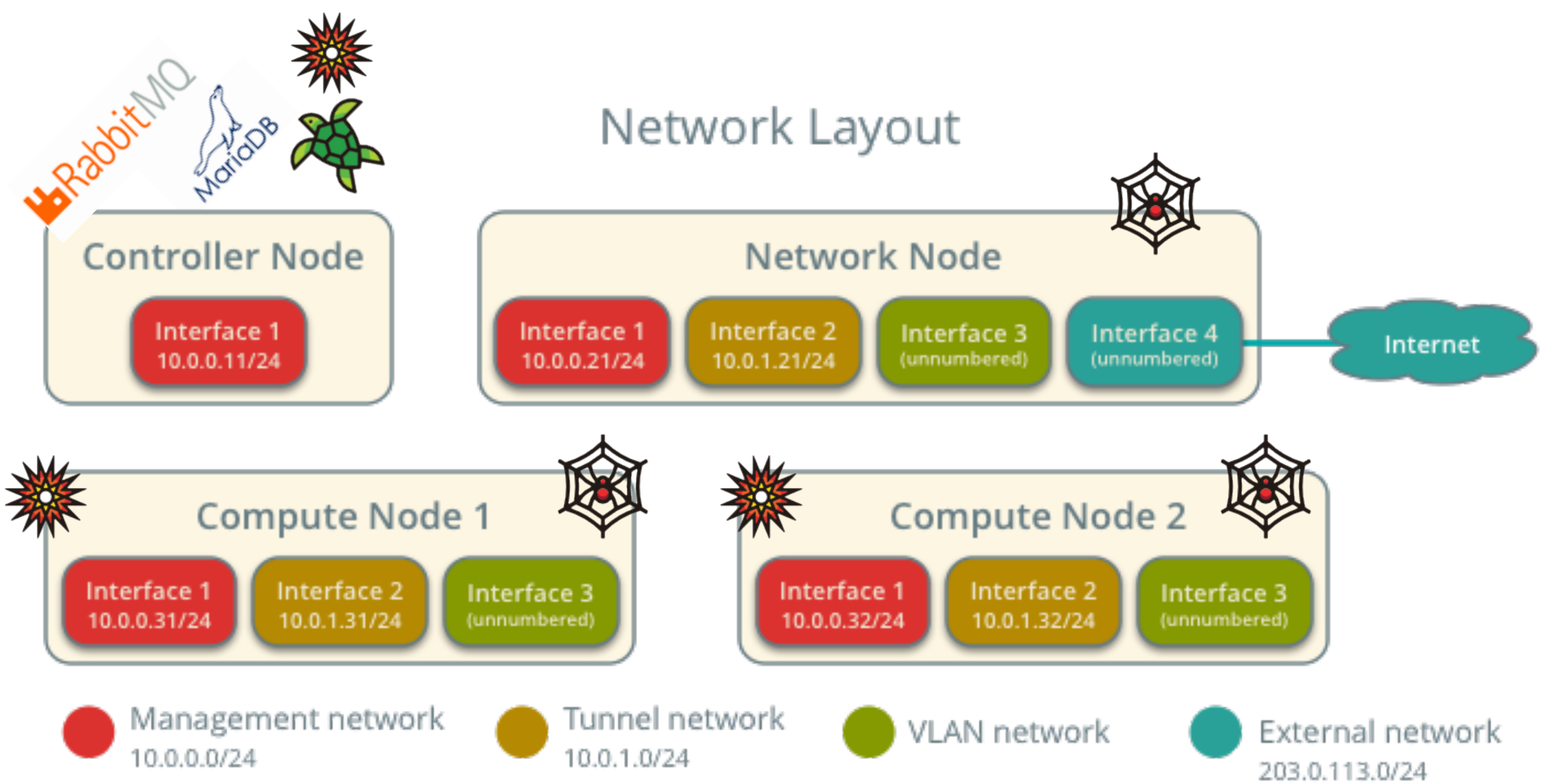
  External networks provide connectivity to external networks such as the Internet (Public IPs)

  Routers typically connect project and external networks.

- Other supporting services (DHCP, ssh keys, …)

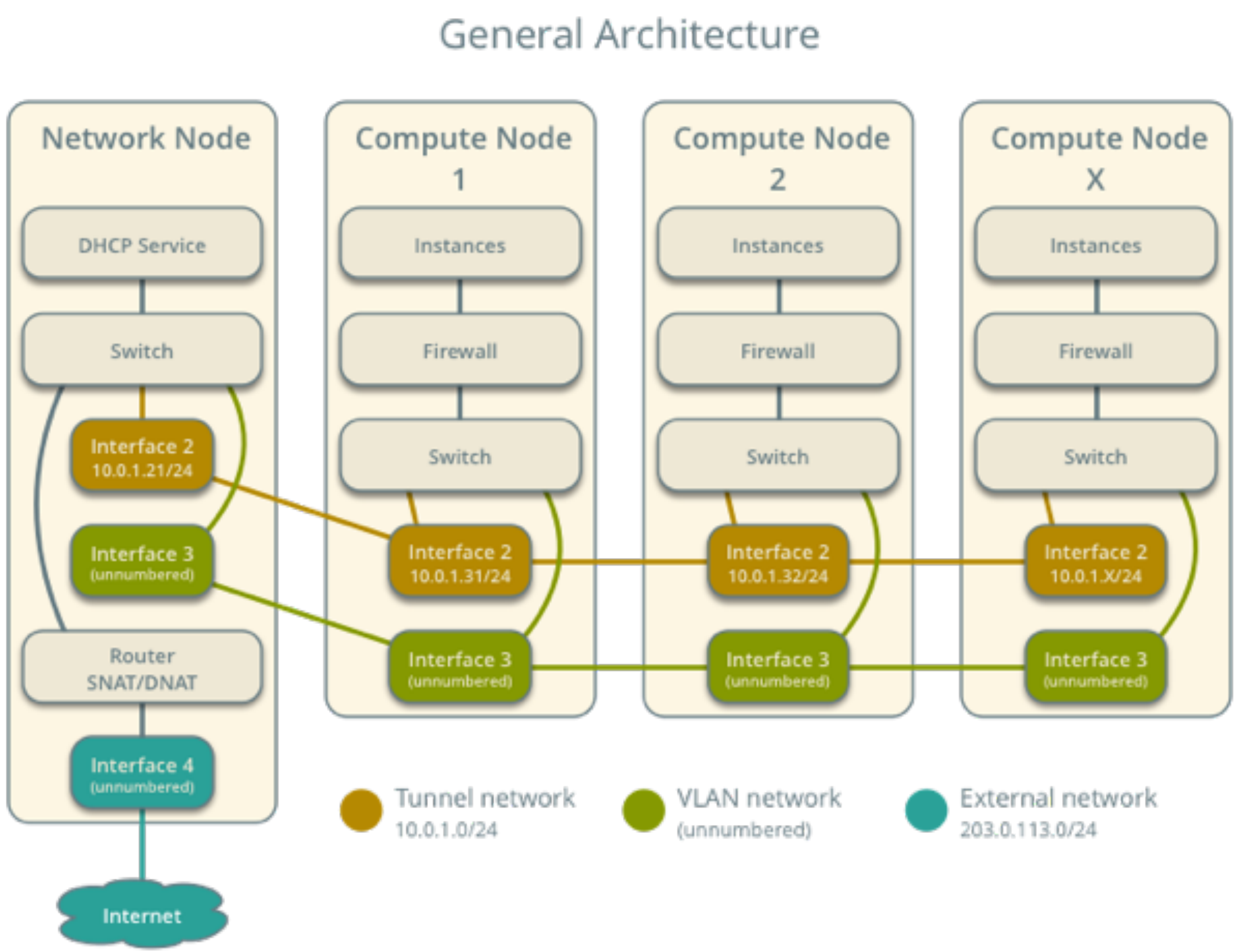
Understanding Neutron Flows

- In the example configuration, the management network uses 10.0.0.0/24, the tunnel network uses 10.0.1.0/24, and the external network uses 203.0.113.0/24. The VLAN network does not require an IP address range because it only handles layer-2 connectivity.



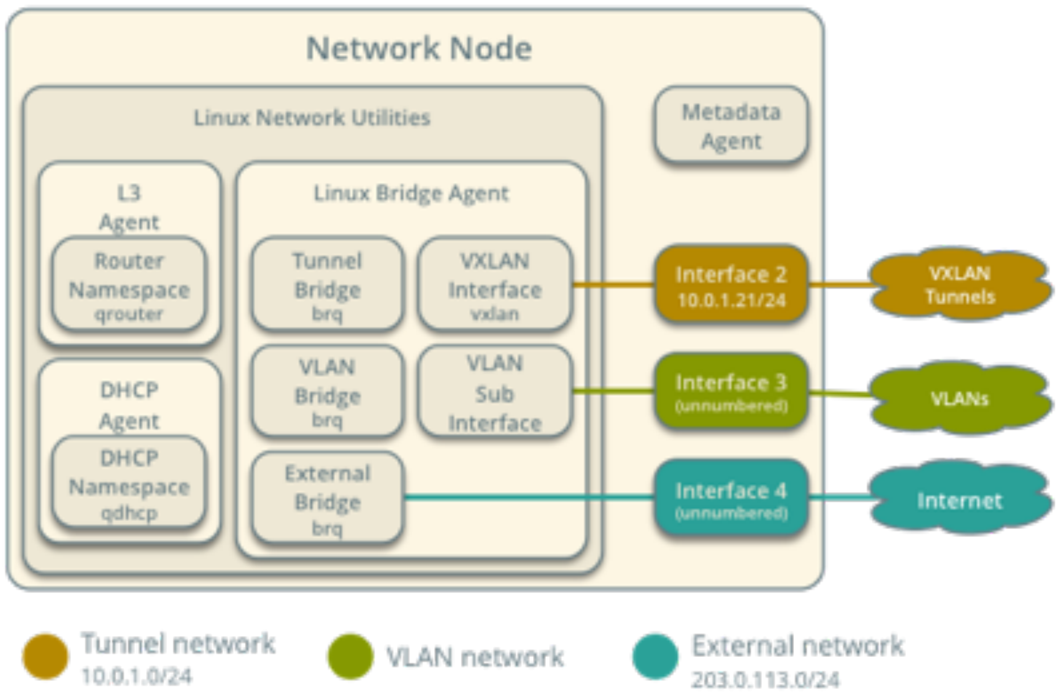
Understanding Neutron Flows

- Routing among project and external networks resides completely on the network node.
- While this makes the management of network flows easier, it may lead to SPOF issues

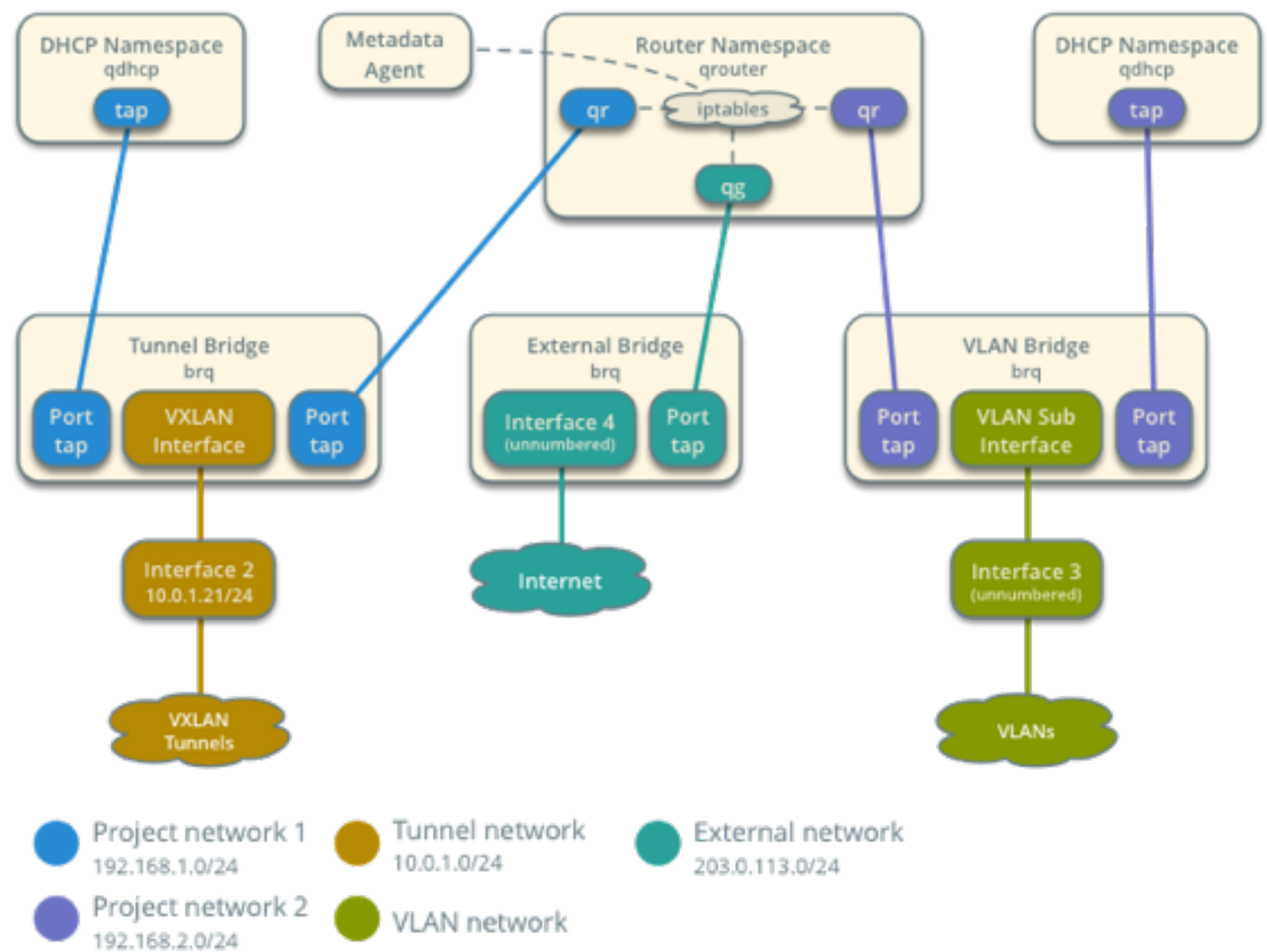


Understanding Neutron Flows

Network Node Overview

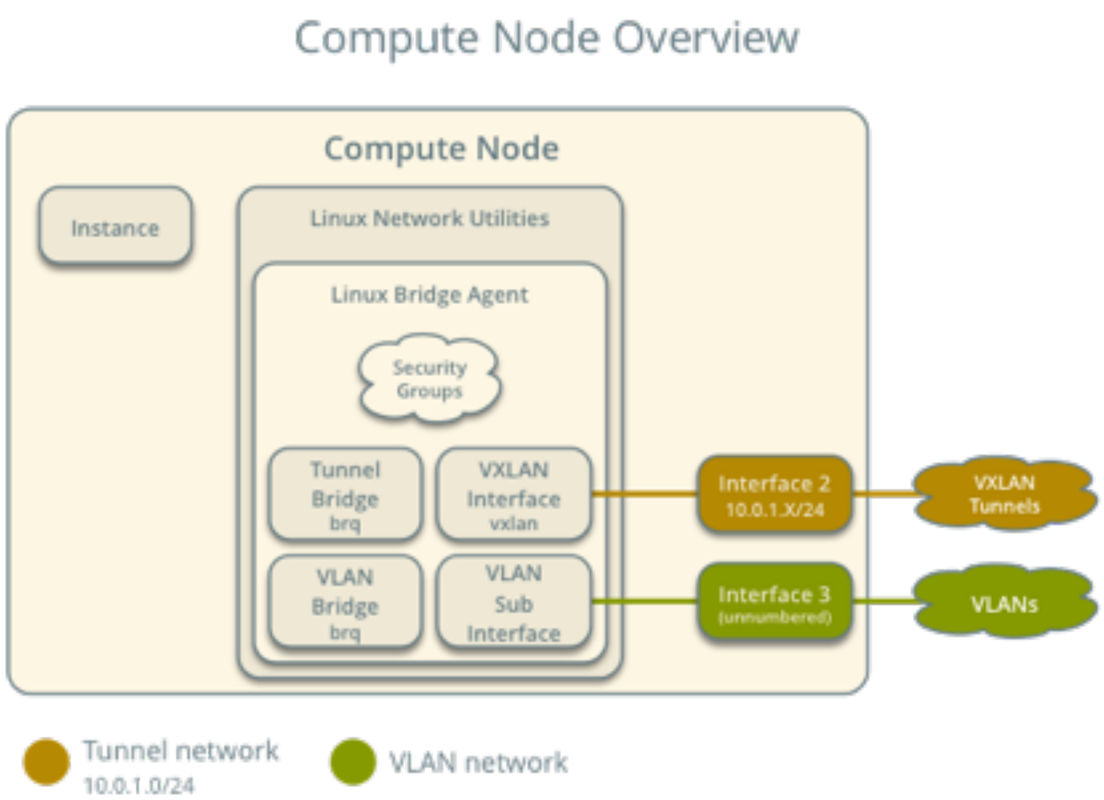


Network Node Components

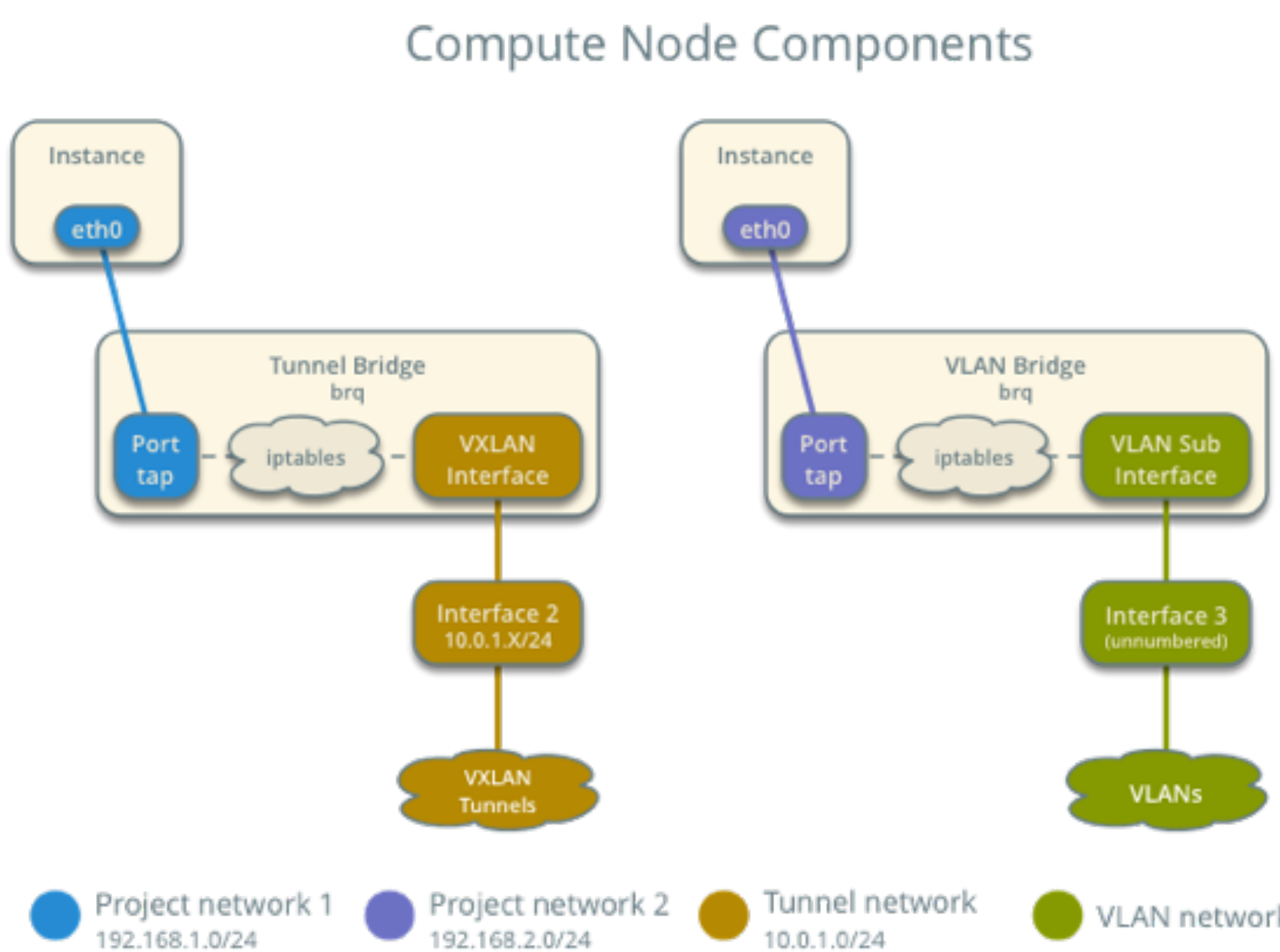


Inside the Network Node

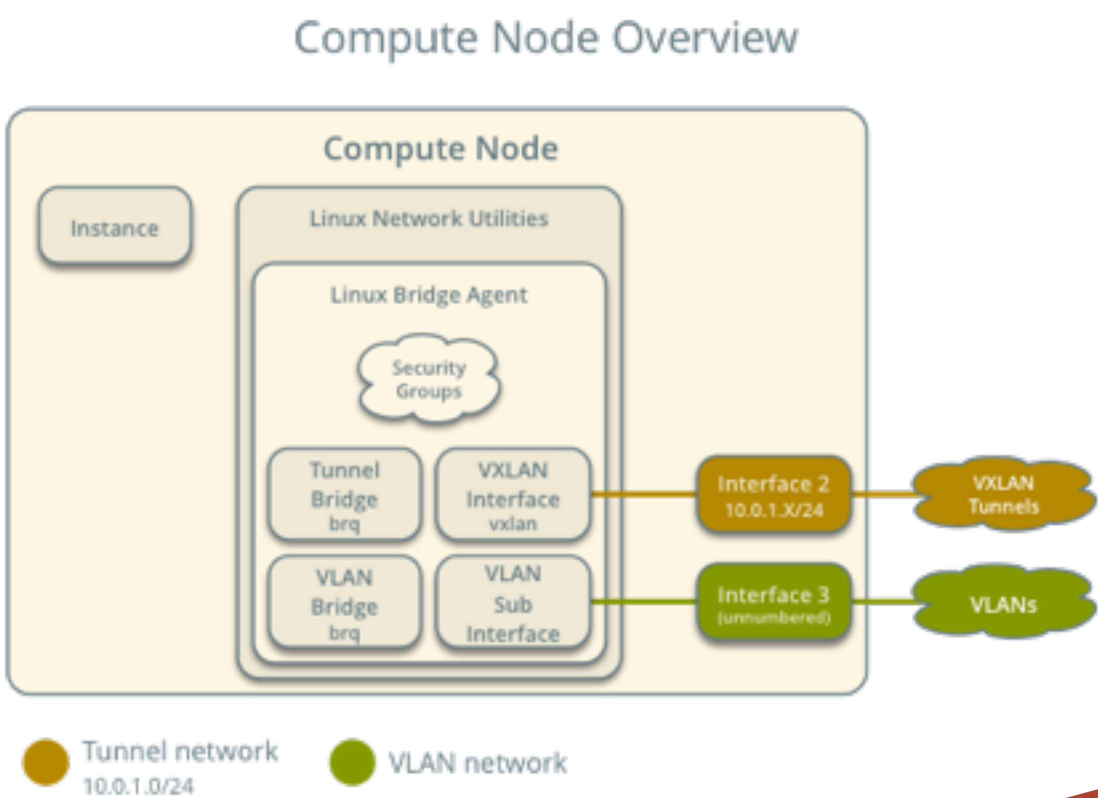
Understanding Neutron Flows

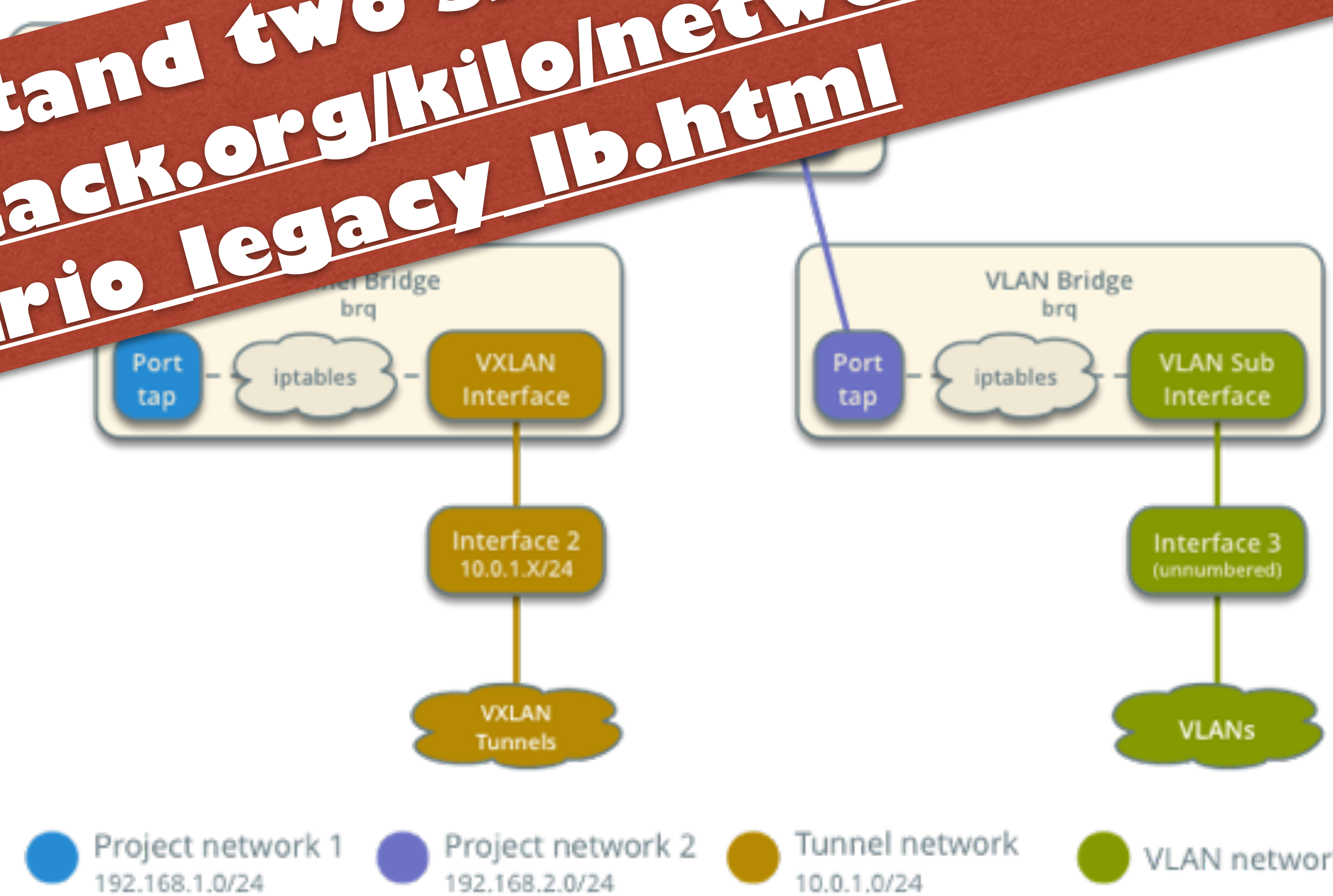


Inside the Compute Node



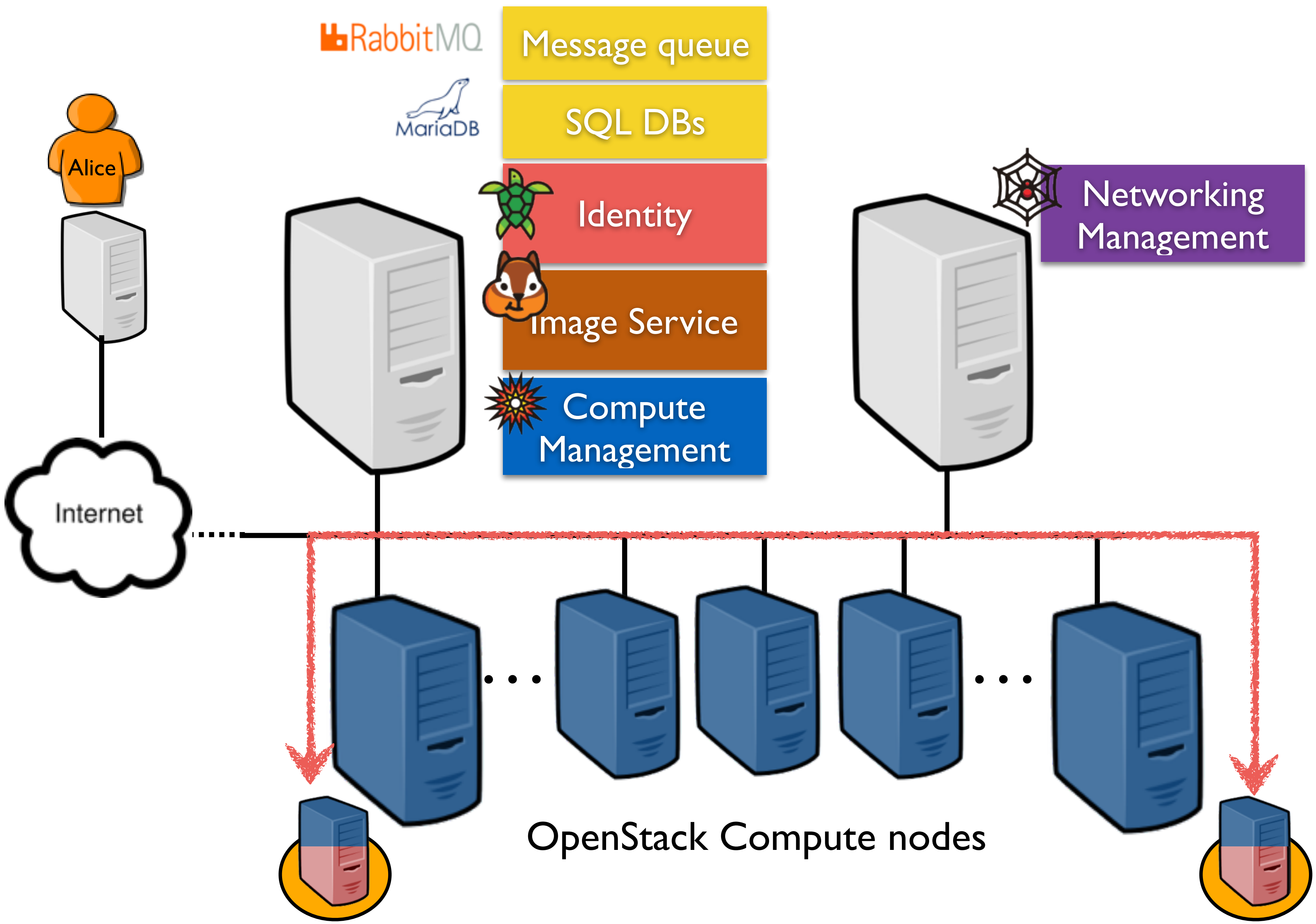
Understanding Neutron Flows





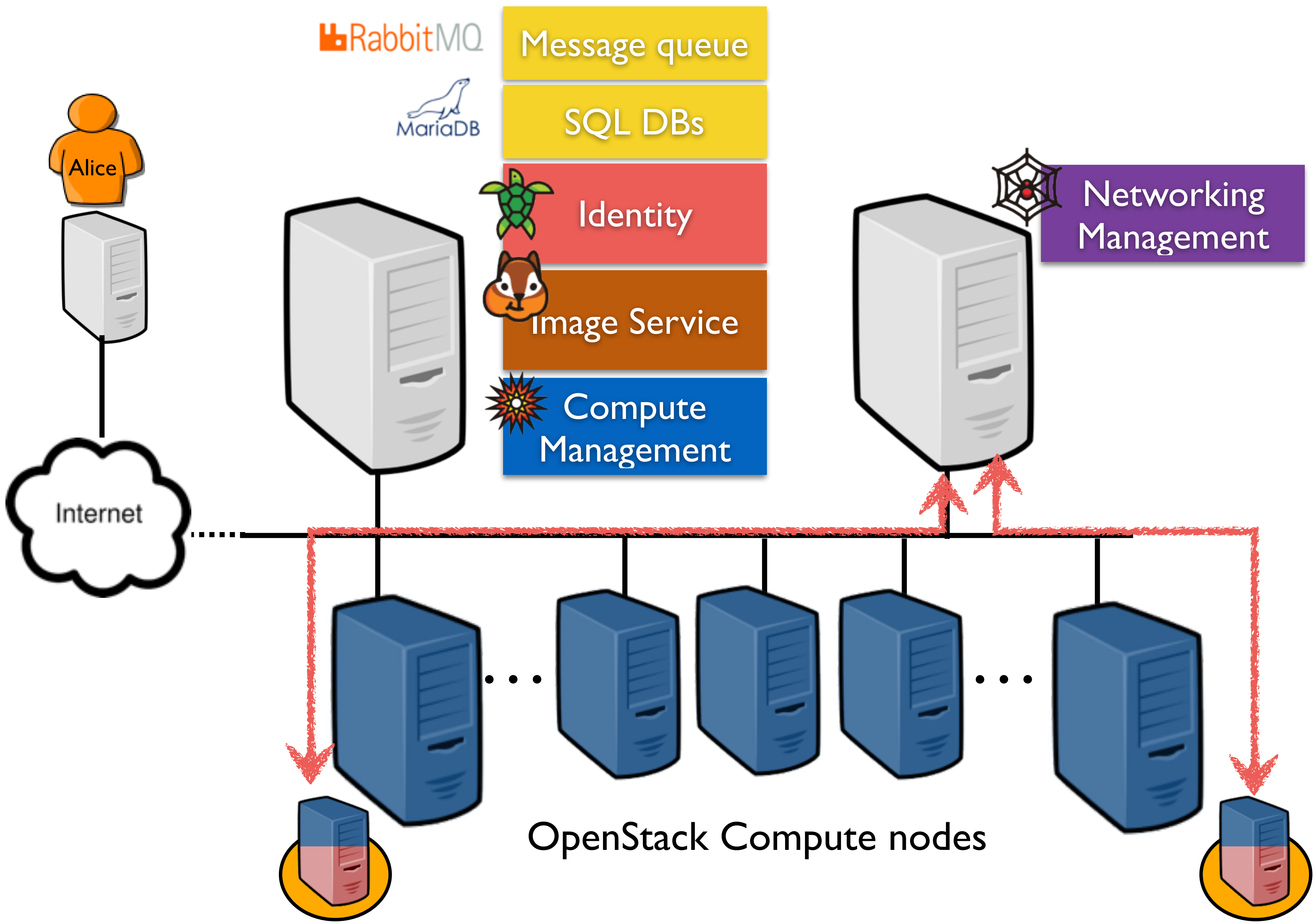
Interested to understand two slides, please see https://docs.openstack.org/kilo/networking-guide/scenario_legacy_lb.html

Neutron Flows in a Nutshell



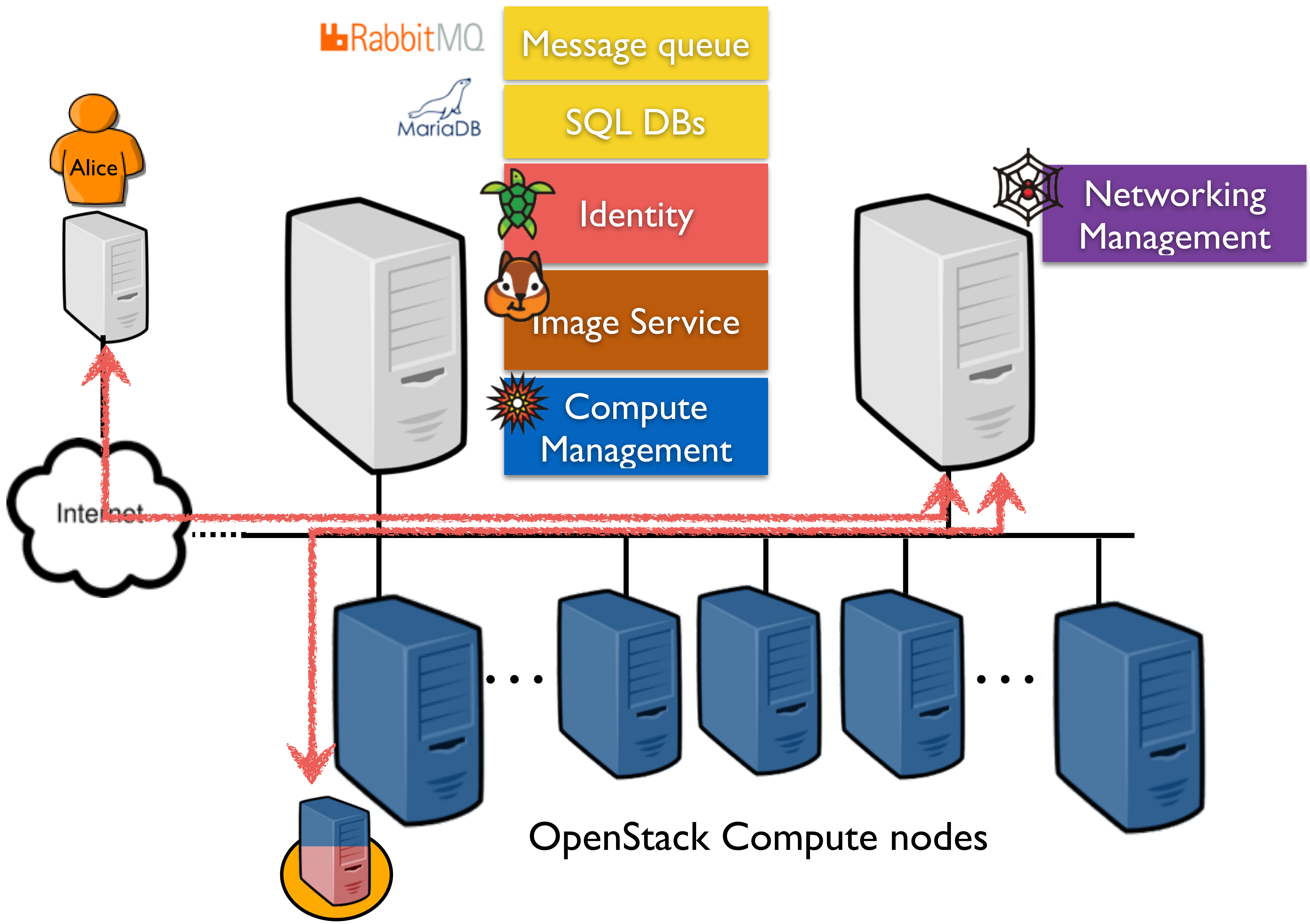
East/West - Same project network

Neutron Flows in a Nutshell



East/West - Different project networks

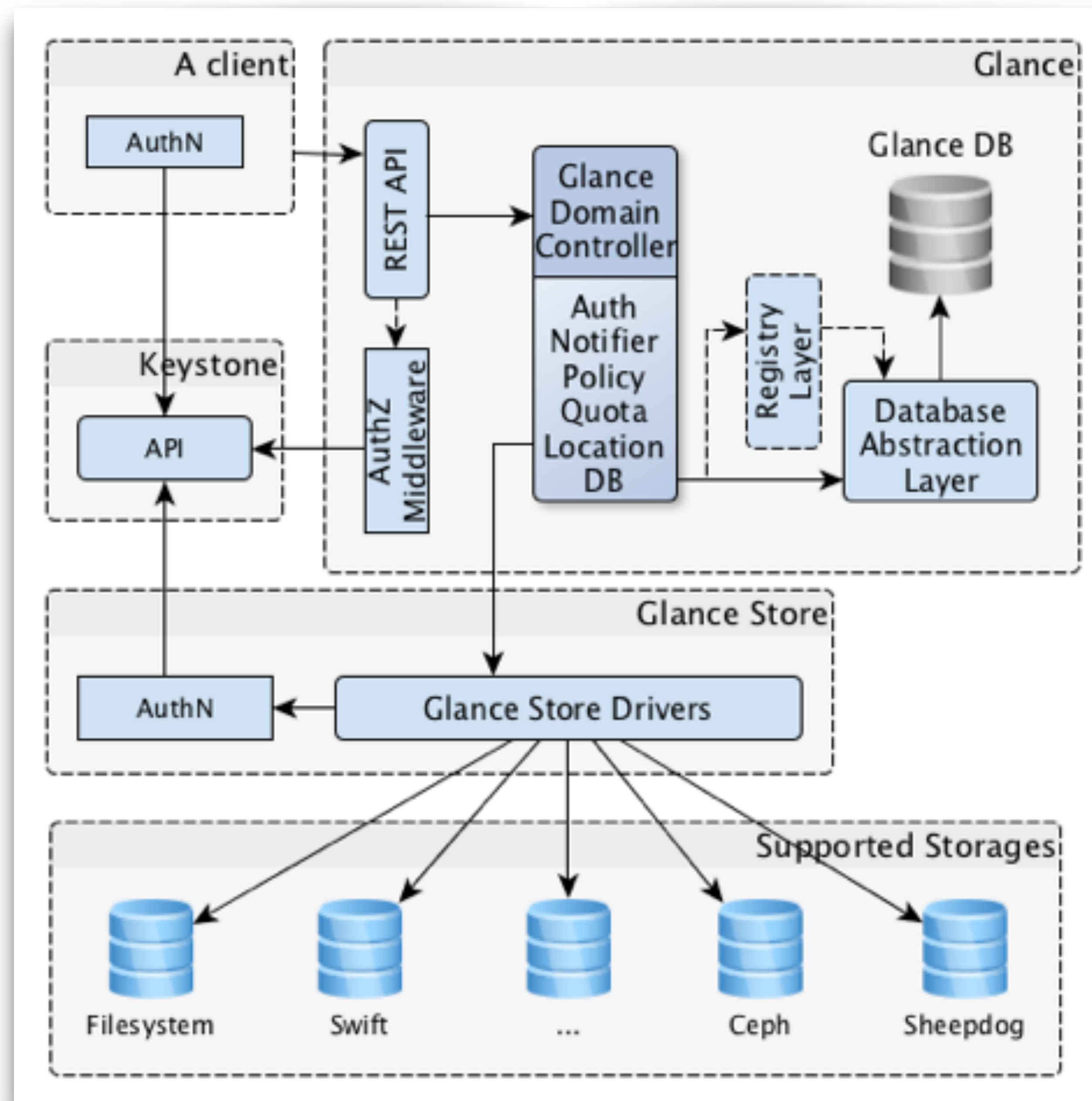
Neutron Flows in a Nutshell



North/South - Private or Public IPs

Focus on Glance

- For each image, one can specify properties
  - Image kind (raw, cqow2, vmdk, iso…)
  - Architecture
  - Distribution
  - Version
  - Storage space requirements
  - RAM minimal size
  - …
- Possible backends
  - Swift/S3
  - Ceph
  - HTTP
  - Local



Other Services IRONIC bare metal

- Ironic provisions bare metal (as opposed to virtual) machines.
- It may be used independently or as part of an OpenStack Cloud, and integrates with the OpenStack Identity (keystone), Compute (nova), Network (neutron), Image (glance) and Object (swift) services.
- When the Bare Metal service is appropriately configured with the Compute and Network services, it is possible to provision both virtual and physical machines through the Compute service's API.
- Although the project is mature, it is barely used…

Other Services…

- A lot…
- CEILOMETER: Metering & Data Collection Service
- DESIGNATES: DNS-as-a-Service
- OCTAVIA: load balancer
- TROVE: a Database-as-a-service (SQL and NoSQL)
- SAHARA: Big Data Processing Framework Provisioning
- ZUN: Container Management Service
- …
- and HEAT: Orchestration

Other Services HEAT orchestration

- Heat orchestrates the infrastructure resources for a cloud application based on templates (HOT) in the form of text files that can be treated like code.
- Heat provides both an OpenStack-native ReST API and a CloudFormation-compatible Query API.
- Heat also provides an autoscaling service that integrates with the OpenStack Telemetry services, so you can include a scaling group as a resource in a template.

Focus on HEAT

- OpenStack-core services hide lot of complexity by automating the deployment process of a new VM/PM
  - Nova contacts Neutron for network configuration
  - Nova contacts Glance to fetch an OS image
  - Nova is in charge of booting and managing the VM
- But starting an empty OS is not enough to deliver a service… The hard part is to put the VM in context:
  - Install the software stack (service + its dependencies)
  - Configure the service
  - Set a floating IP address to be reachable from Internet
- A Cloud application is made of multiple services
  - Multiple machines to boot, and services to deploy and configure to deliver the application

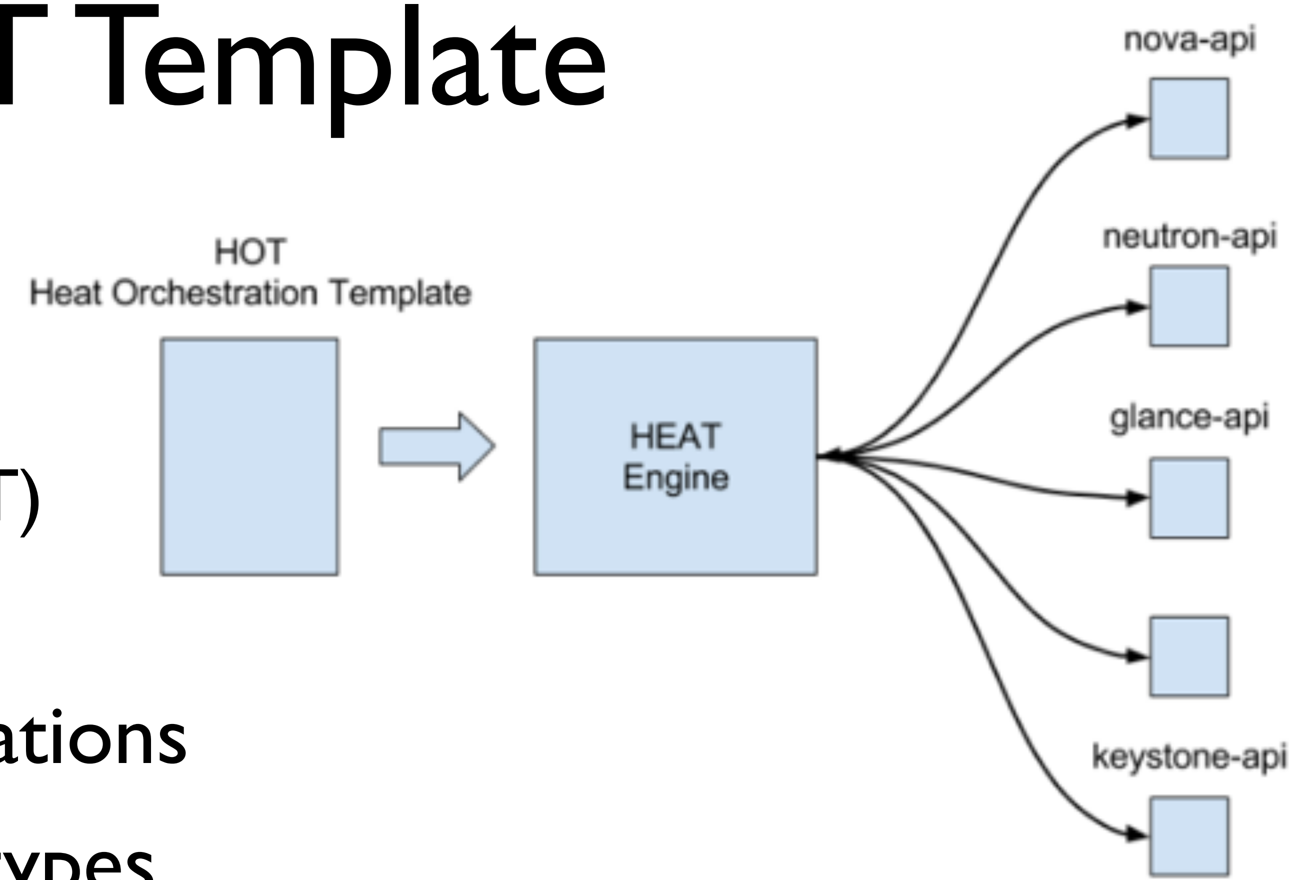
credits: D. Pertin - Introduction to Heat

Focus on HEAT

- DEVOPS Philosophy
- Infrastructure as Code
  Scale in/out (horizontal scaling)
  Automation
- Monitoring services/apps (instead of the infrastructure)
- Backup/Restart on demand

Manage Application Life-Cycle with code !

HEAT Template



- Readable format:

  HEAT Orchestration Template (HOT)

  YAML file
- Describe infrastructure and applications
- Declare any OpenStack resource types

  Instances, floating IPs, volumes, images, users, …
- Declare relationships between resources

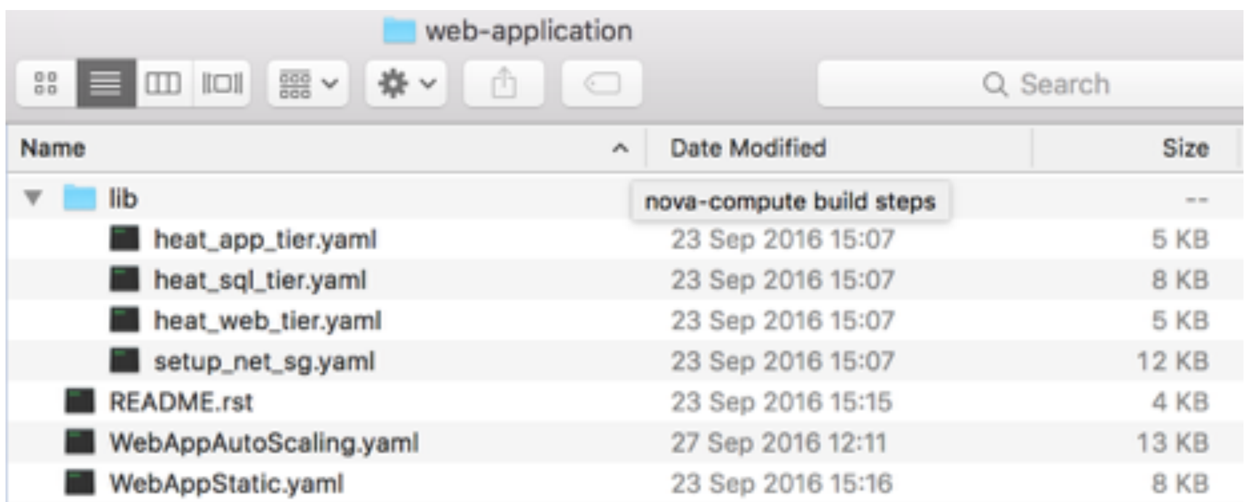
  e.g. a VM must be booted before installing software on it
- HEAT Engine

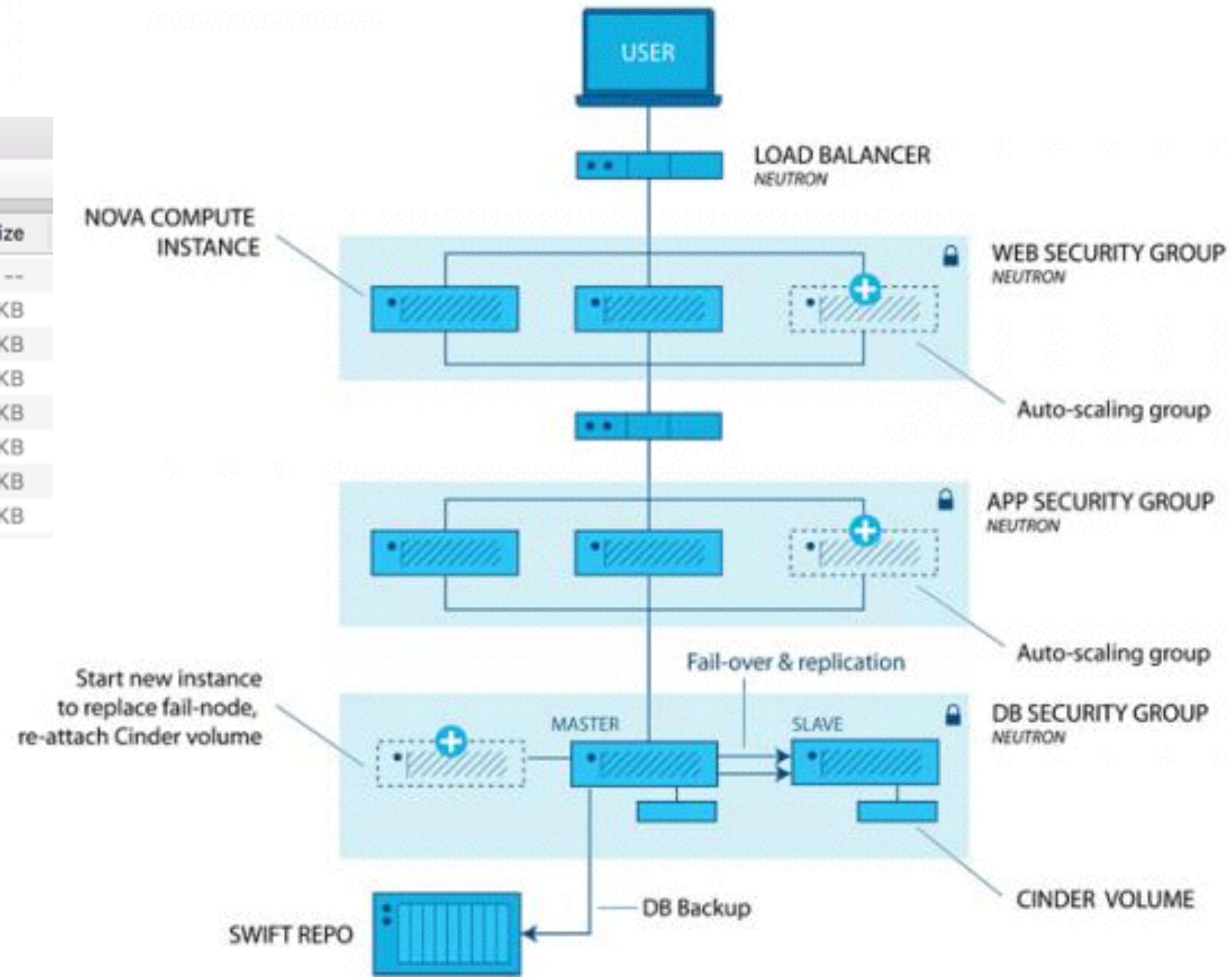
  Take a template as input
  Parse it
  Execute tasks through OpenStack API calls

credits: D. Pertin - Introduction to Heat

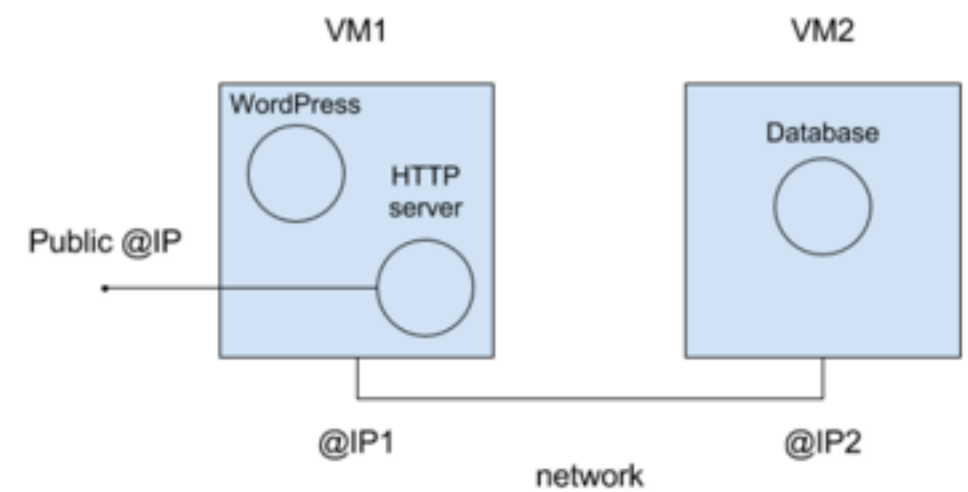
Understanding HEAT Through an Example

- Exemple: a 3Tier-Web Application : Wordpress





Understanding HEAT Through an Example



- Boot two virtual machines

  VM1 and VM2
  Ubuntu based OS
  Connect them to the private network

- Install the services

  Install a database server (mysql) on VM2
  Install a HTTP server (apache2) on VM1
  Download Wordpress on VM1

- Configure services
  Create an appropriate database in mysql for wordpress
  Configure wordpress to access this database
  Configure apache2 to serve wordpress

- Assign a floating IP to VM1

How can I automatise this process by leveraging HEAT?

credits: D. Pertin - Introduction to Heat

HEAT Template

- The Hello World example - boot a VM

```
$ cat boot.yaml
heat_template_version: 2015-04-30

description: Simple template to deploy a single compute instance

resources:                    # HEAT resources are declared here
  my_instance:                # Name of my resource
    type: OS::Nova::Server    # Type of my resource (this resources defines a VM)
    properties:               # Here we define the properties of this resource type
      key_name: my_key_name   # Name of an SSH key managed by Nova (or Barbican)
      image: ubuntu-trusty-x86_64      # Name of an image managed by Glance
      flavor: m1.small                 # Name of a flavor managed by Nova


$ openstack stack create my_stack -f boot.yaml
```

credits: D. Pertin - Introduction to Heat

HEAT Template

- Let's make it a bit more complex: boot a vm with parameters

```
$ cat boot_with_parameter.yaml
heat_template_version: 2015-04-30

description: Simple template to deploy a single compute instance with a parameter

parameters:                  # Parameters definition for this template
  key_name:                  # Name of the parameter
    type: string              # Type of the parameter
    description: Name of a KeyPair to enable SSH access to the instance

resources:
  my_instance:
    type: OS::Nova::Server
    properties:
      key_name: { get_param: key_name }    # Use an intrinsic function to get the
value of a parameter
      image: ubuntu-trusty-x86_64
      flavor: m1.small

$ openstack stack create my_stack -f boot_with_parameters.yaml --parameter
key_name=my_key
```

credits: D. Pertin - Introduction to Heat

HEAT Template

- Let's make it a bit more complex: boot a vm and get some outputs

```
$ cat boot_with_outputs.yaml
heat_template_version: 2015-04-30

description: Simple template to deploy a single compute instance, outputs its ip
address

resources:
  my_instance:
    type: OS::Nova::Server
    properties:
      key_name: my_key_name
      image: ubuntu-trusty-x86_64
      flavor: m1.small

outputs:                                  # Definition of the outputs of this template
 instance_ip:                             # Name of the output
  description: IP address of the deployed compute instance
    value: { get_attr: [my_instance, first_address] } # Set the IP address of the
                                                       # machine as the value of the
                                                       # output instance_ip

$ openstack stack create my_stack -f boot_with_outputs.yaml
```

credits: D. Pertin - Introduction to Heat

HEAT Template

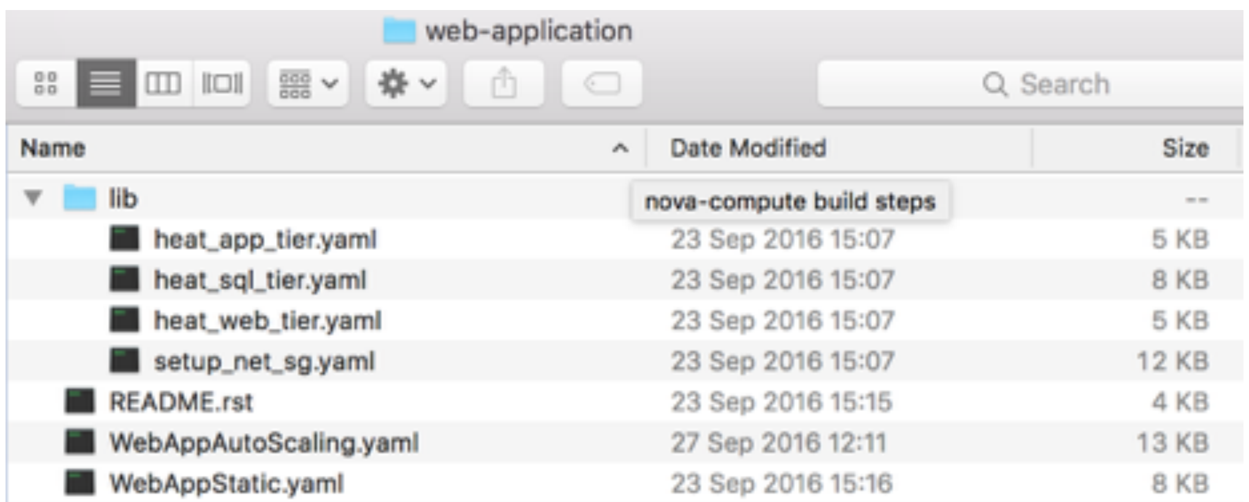
- Even more complex

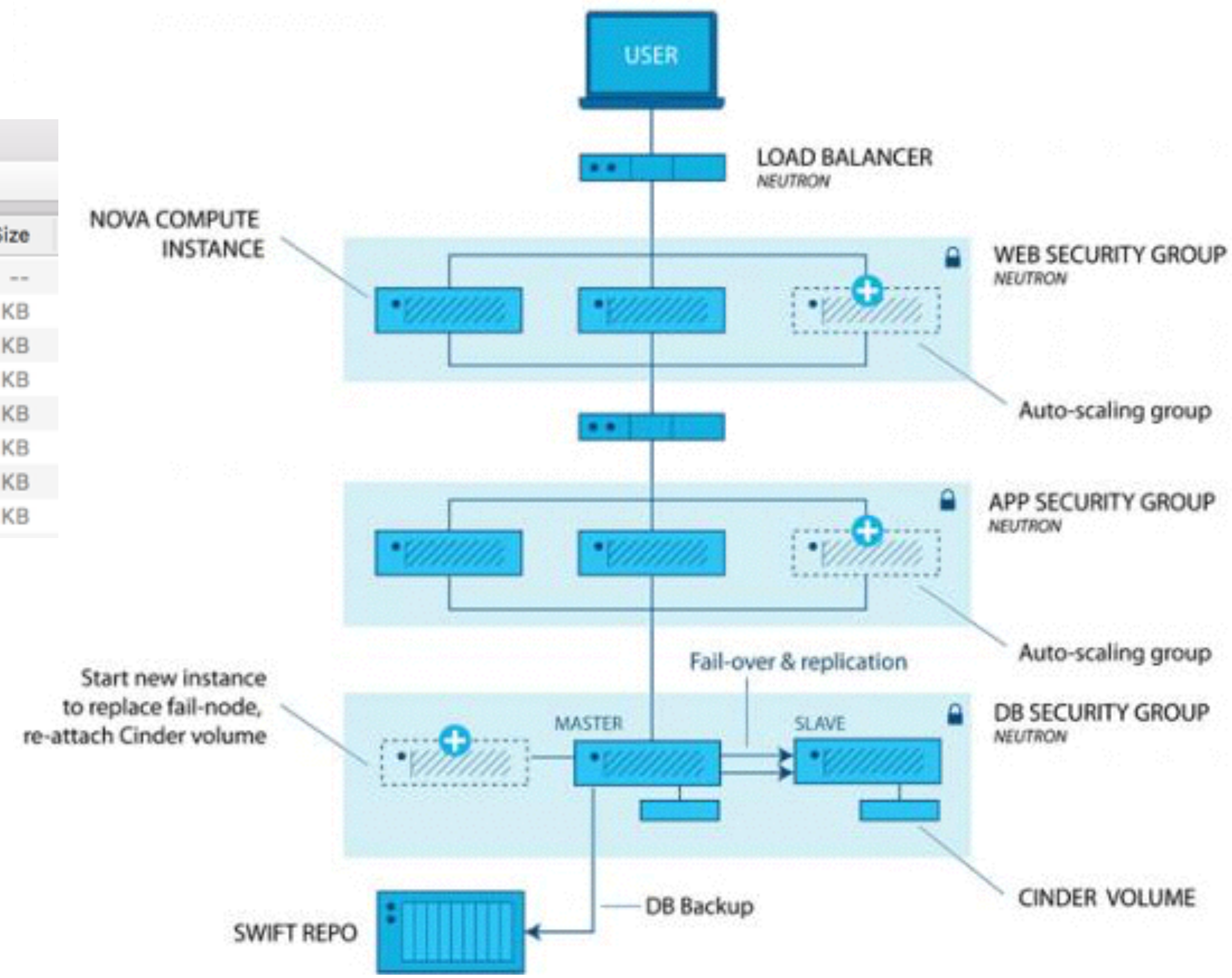
```
$ cat  boot_sql_server.yaml
heat_template_version: 2015-04-30
description: Template to deploy SQL server
parameters:
  DBRootPassword:
    type: string
resources:
  my_sql_instance:
    type: OS::Nova::Server
    properties:
      # general properties ...
      user_data:            # Definition of a boot script
        str_replace:        # Intrinsic function to replace
                            # strings in the script by parameters
          template: |       # Description of the script
            #!/bin/bash
            # do things like install mysql ...
            mysqladmin -u root password $db_rootpassword
            # do more things ...
          params:           # Description of the used parameters
            $db_rootpassword: { get_param: DBRootPassword }…

$ openstack stack create my_sql_server -f boot_sql_server.yaml --parameter
DBRootPassword=0p3nSt4cK
```

credits: D. Pertin - Introduction to Heat

Understanding HEAT Through an Example

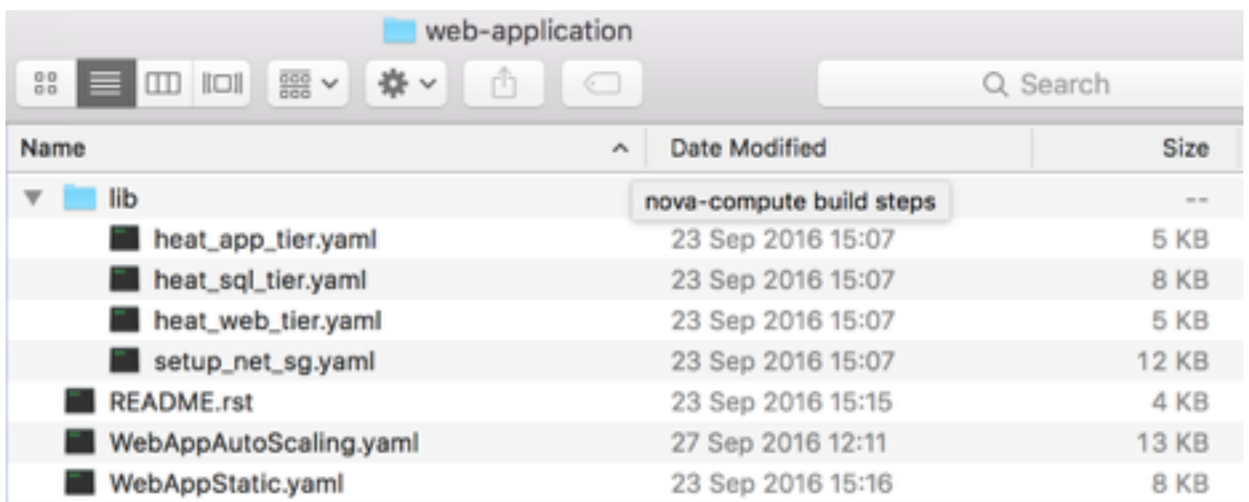
- The complete example during the practical session !

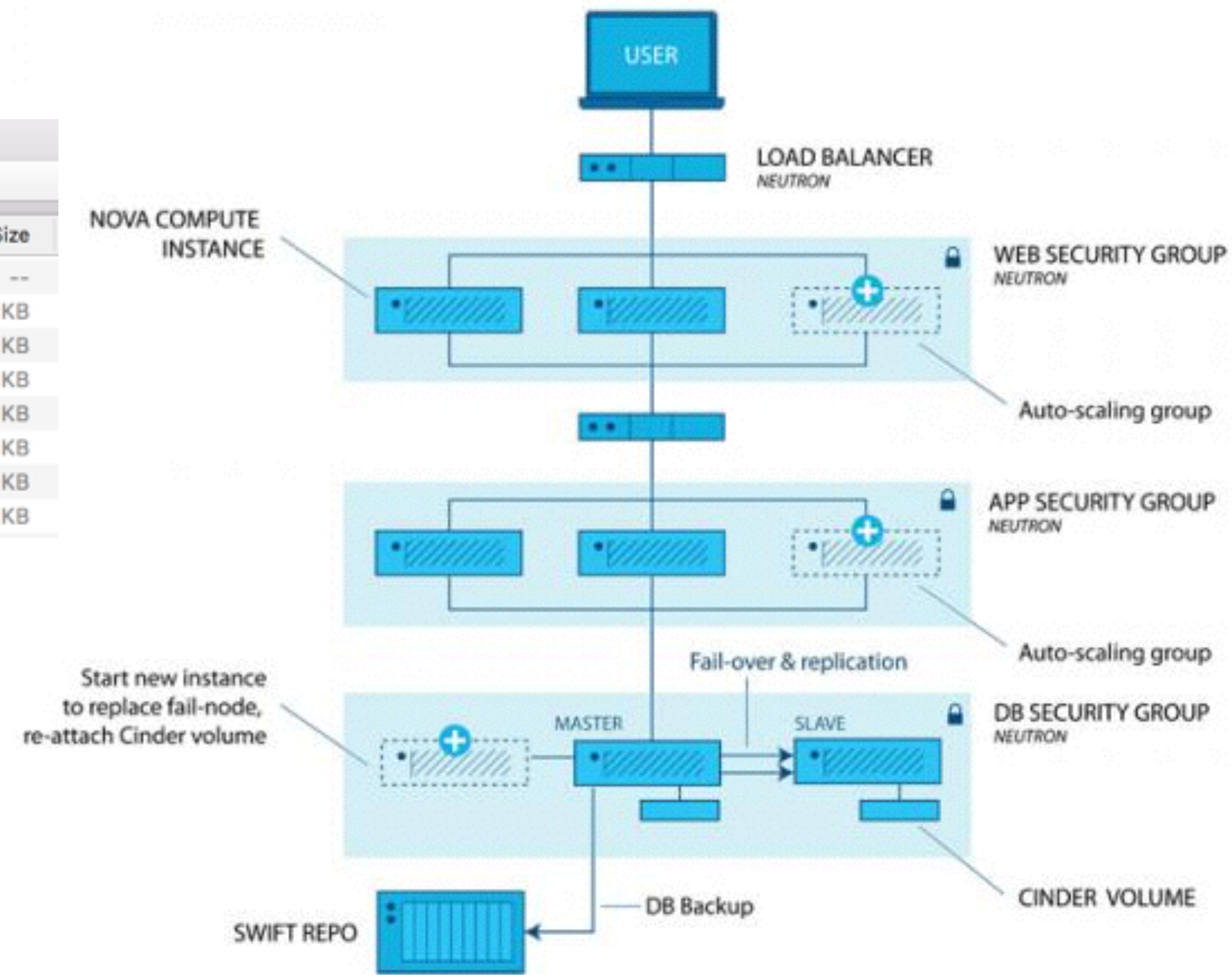




Understanding HEAT Through an Example

- The complete example during the practical session !





R-A Cherrueau
D. Pertin
Next Week !

Segregation Tools

- It is sometimes a key point to segregate a large infrastructure into several subsets. OpenStack provides several ways:

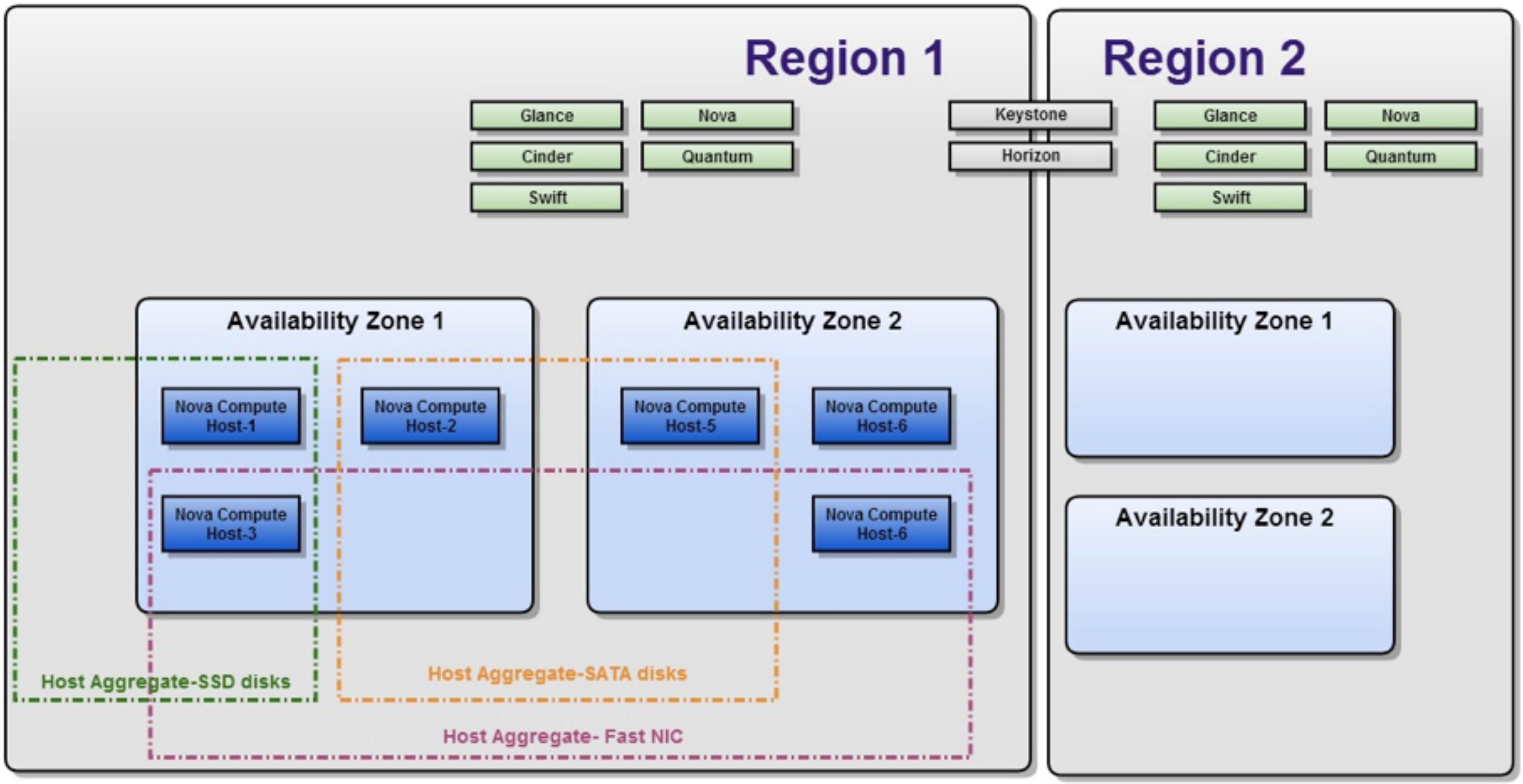
  Host aggregates - Nova: Classify compute nodes according to their specifics (Storage, GPU, …)

  Availability Zones - Nova, Cinder: Classify resources according to availability aspects (racks, data centers, …)

  Cells - Nova (Neutron soon !?): address segregation and scalability needs (one communication bus and one DB per cells)

  Regions - a federation like approach, each region has an almost complete OpenStack (keystone/horizone are shared). Equivalent to the AWS region.

Segregation Tools



credits: Kimi Zhang - https://kimizhang.wordpress.com/2013/08/26/openstack-zoning-regionavailability-zonehost-aggregate/

I want to test it !

DevStack

- A series of extensible scripts used to quickly bring up a complete OpenStack environment based on the latest versions of everything from git master
  `stack.sh + local.conf`
- A development environment and as the basis for much of the OpenStack project's functional testing.

DevStack

- A series of extensible scripts used to quickly bring up a complete OpenStack environment based on the latest versions of everything from git master
  `stack.sh + local.conf`
- A development environment and as the basis for much of the OpenStack project's functional testing.

WARNING: DevStack makes substantial changes to your system
Only launch it inside a VM

Experimental eNvironment for OpenStack

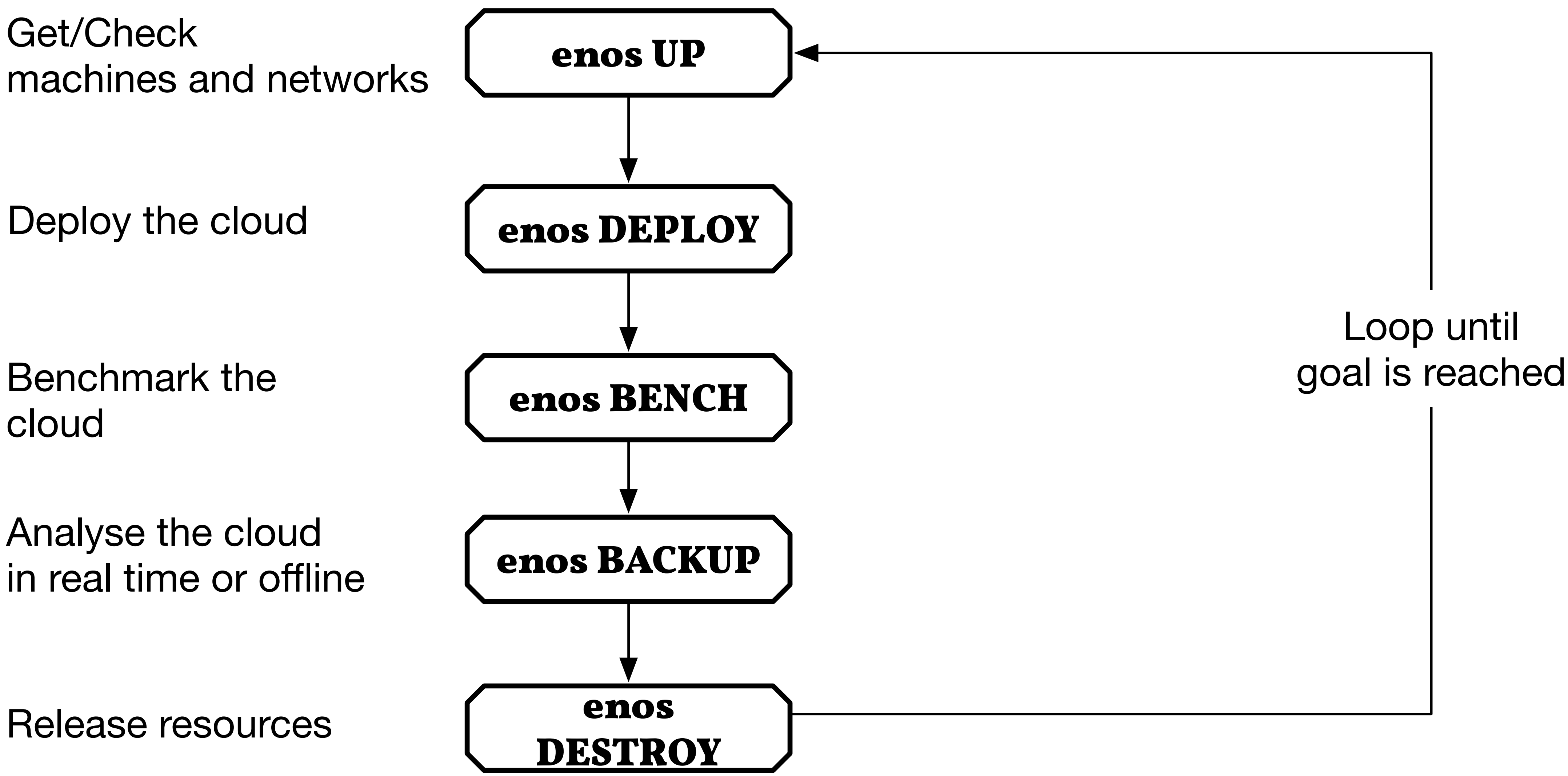
- A dedicated framework to conduct performance analyses of OpenStack at large-scale in a reproducible manner. The framework enables engineers/researchers to conduct experiments in an automize manner on top of different testbeds such as Grid'5000, Chameleon, OpenStack…
- Developed in the context of the Discovery Initiative
- Deploy a real production system by leveraging Kolla (i.e. not DevStack)

Experimental eNvironment for OpenStack

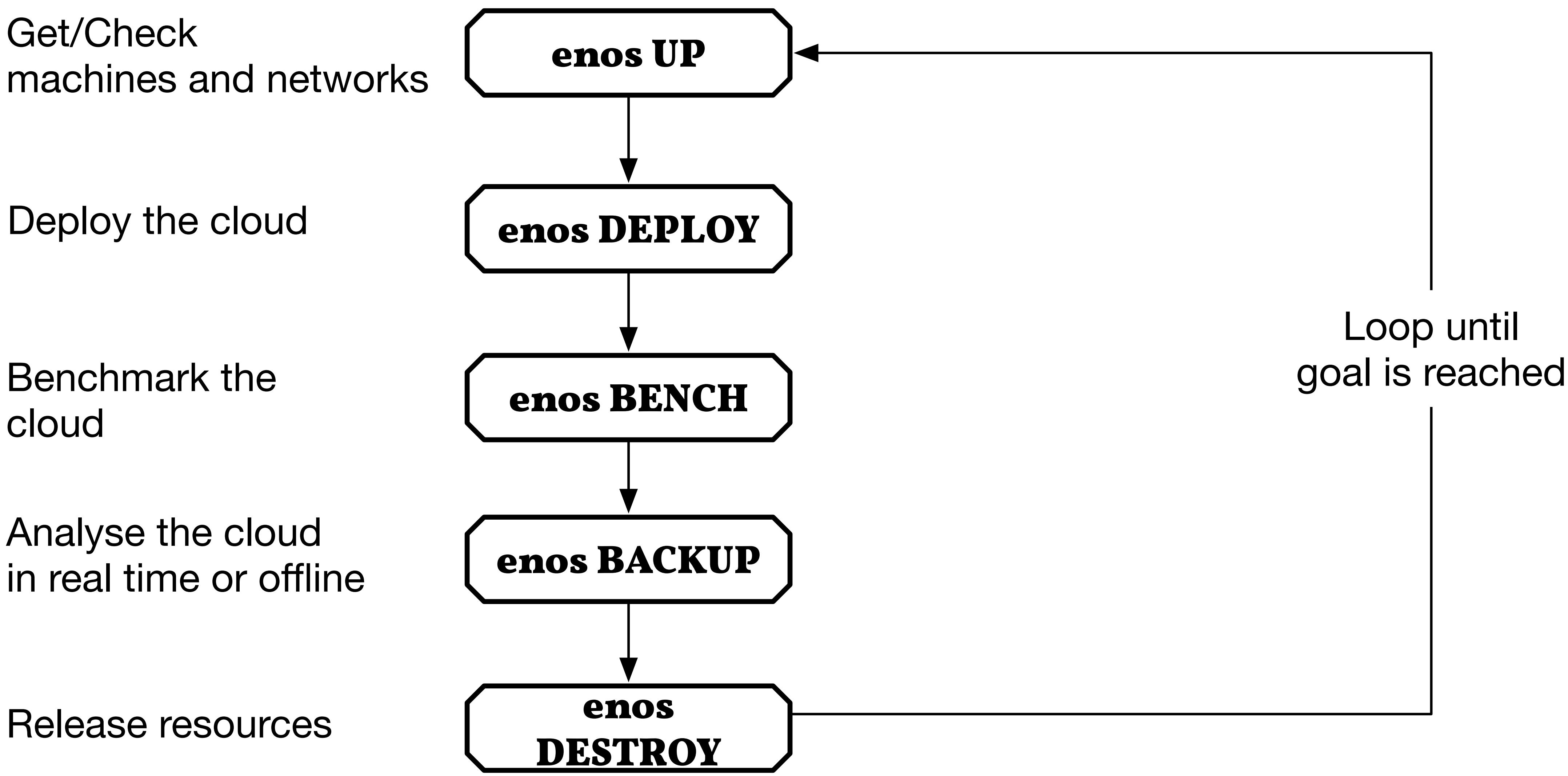
- A dedicated framework to conduct performance analyses of OpenStack at large-scale in a reproducible manner. The framework enables engineers/researchers to conduct experiments in an automize manner on top of different testbeds such as Grid'5000, Chameleon, OpenStack…
- Developed in the context of the Discovery Initiative
- Deploy a real production system by leveraging Kolla (i.e. not DevStack)

R-A Cherrueau / D. Pertin
Next Week !

Experimental eNvironment for OpenStack



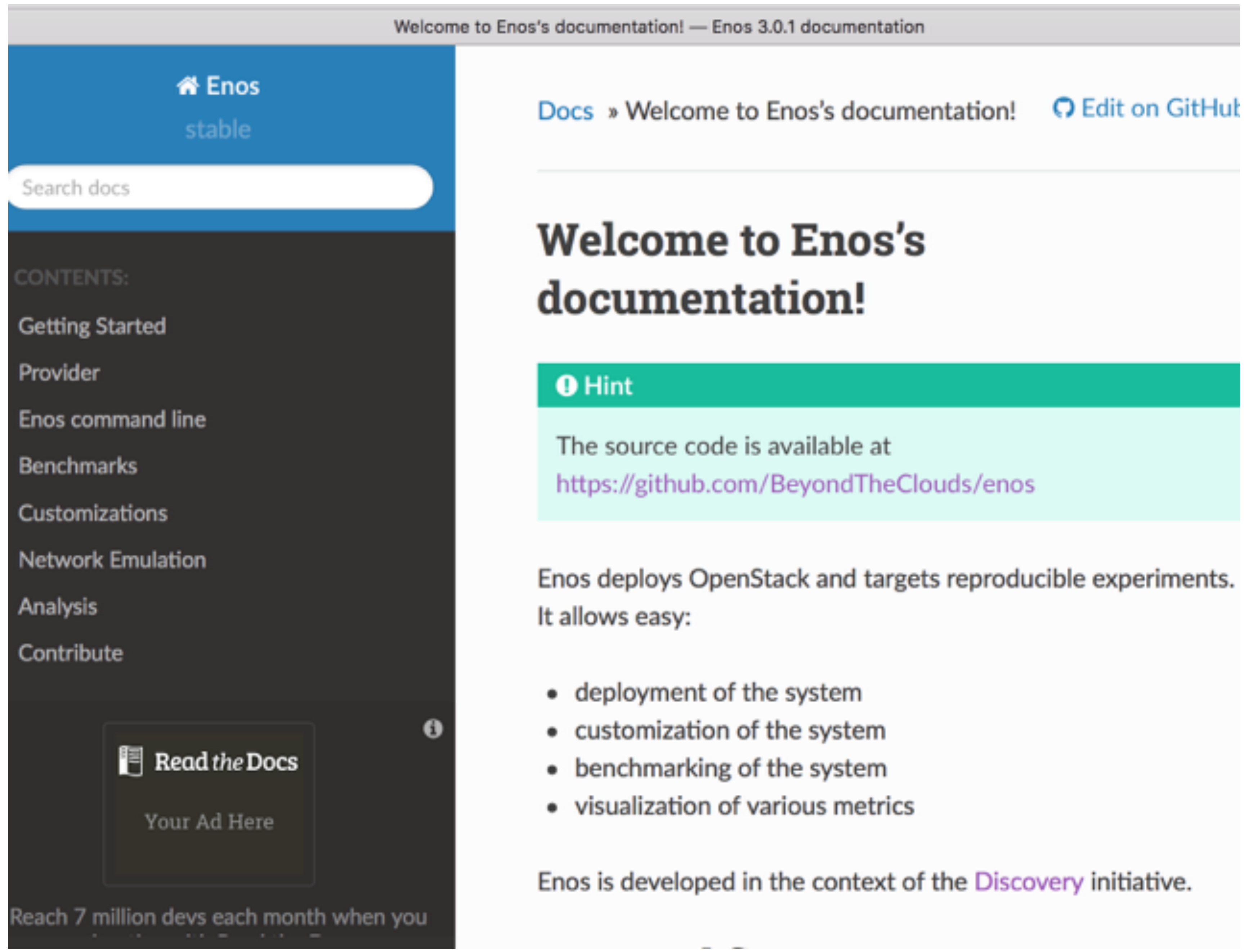
Experimental eNvironment for OpenStack



R-A Cherrueau / D. Pertin
Next Week !

You cannot Wait For the Practical Session

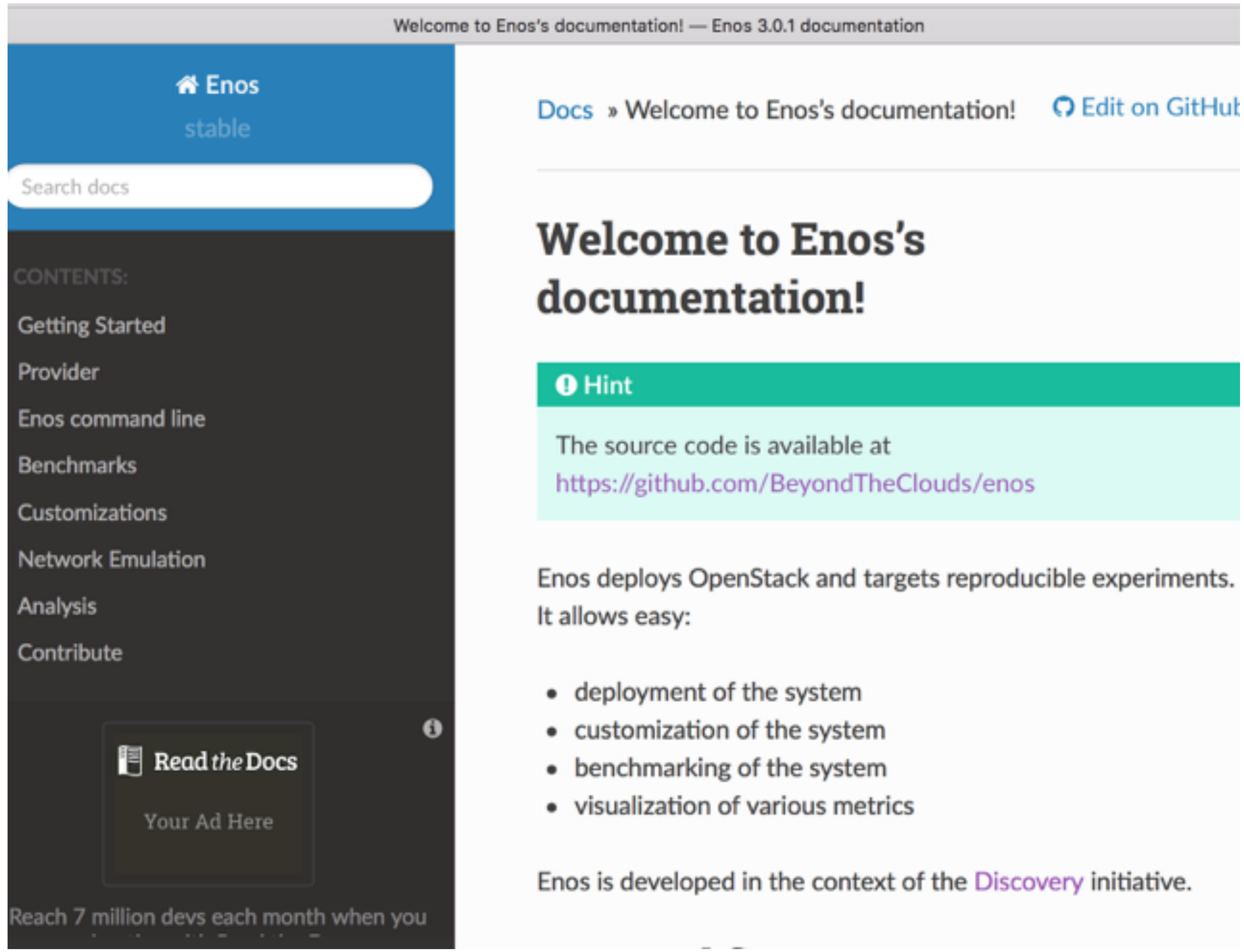
- Install
  VirtualBox
  Vagrant
  EnOS
- Prepare your RAM (at least 10GB ;-))



You cannot Wait For the Practical Session

- Install
  VirtualBox
  Vagrant
  EnOS

- Prepare your RAM (at least 10GB ;-))

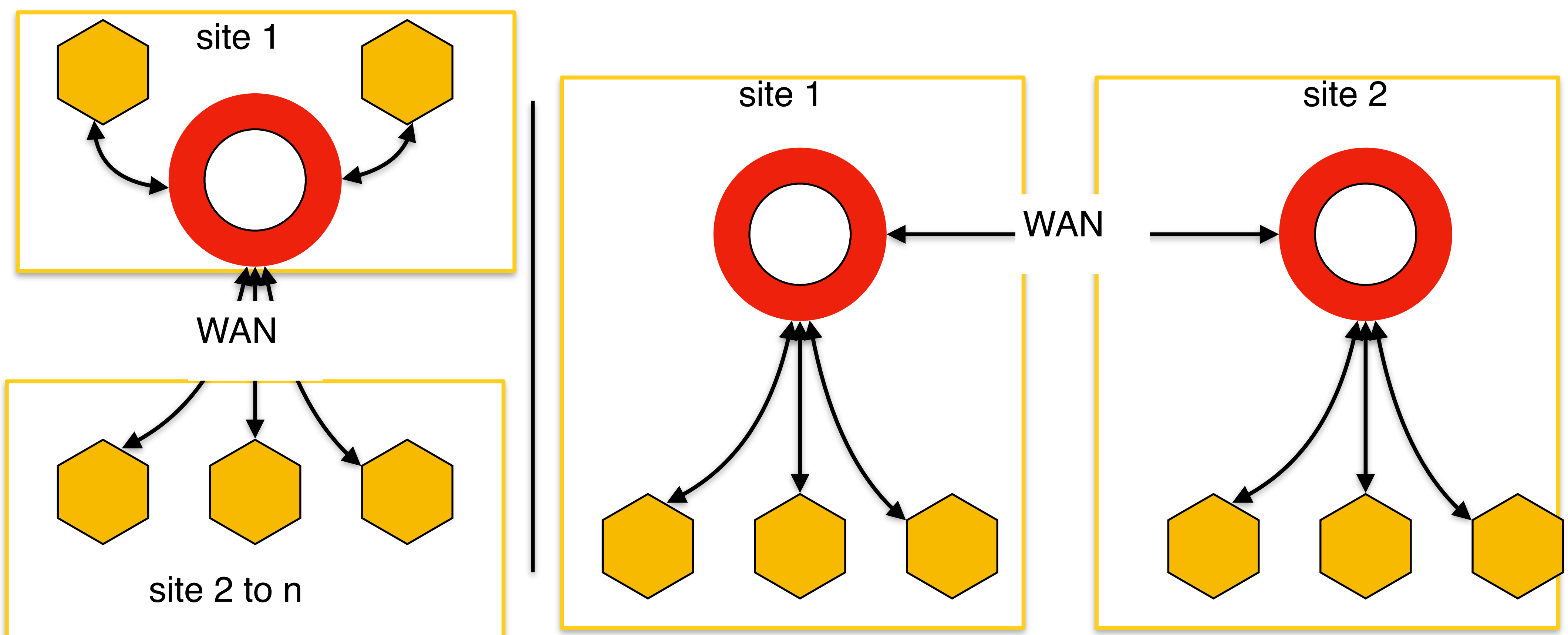


https://enos.readthedocs.io/en/stable/

Why diving in such a level of details?!

Fog/Edge Challenges

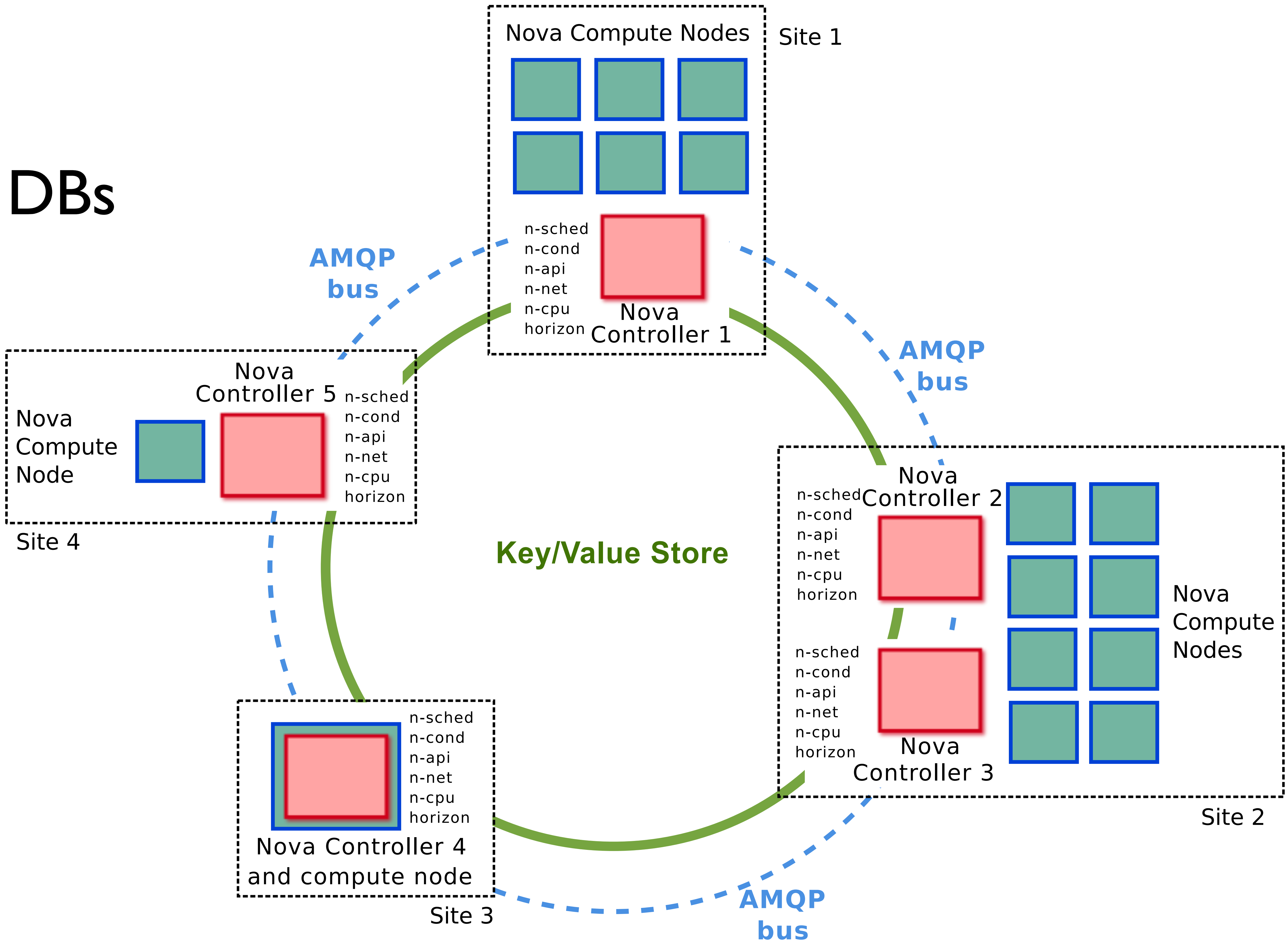
- Communication Bus
  - Central rabbitMQ and many edge servers
  - Distributed RabbitMQ through federations



credits: M. Simonin - IPL Discovery

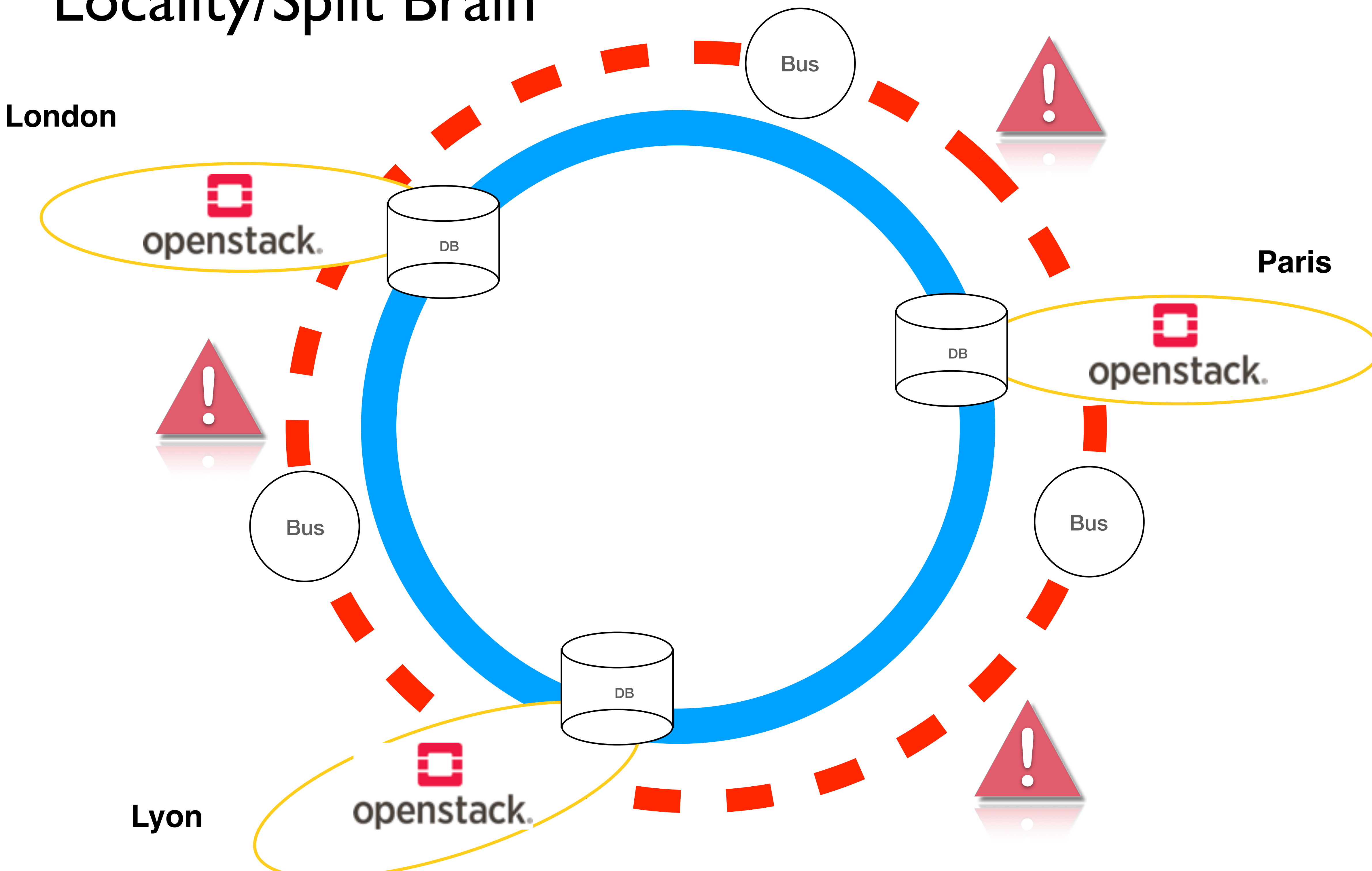
Fog/Edge Challenges

- DBs

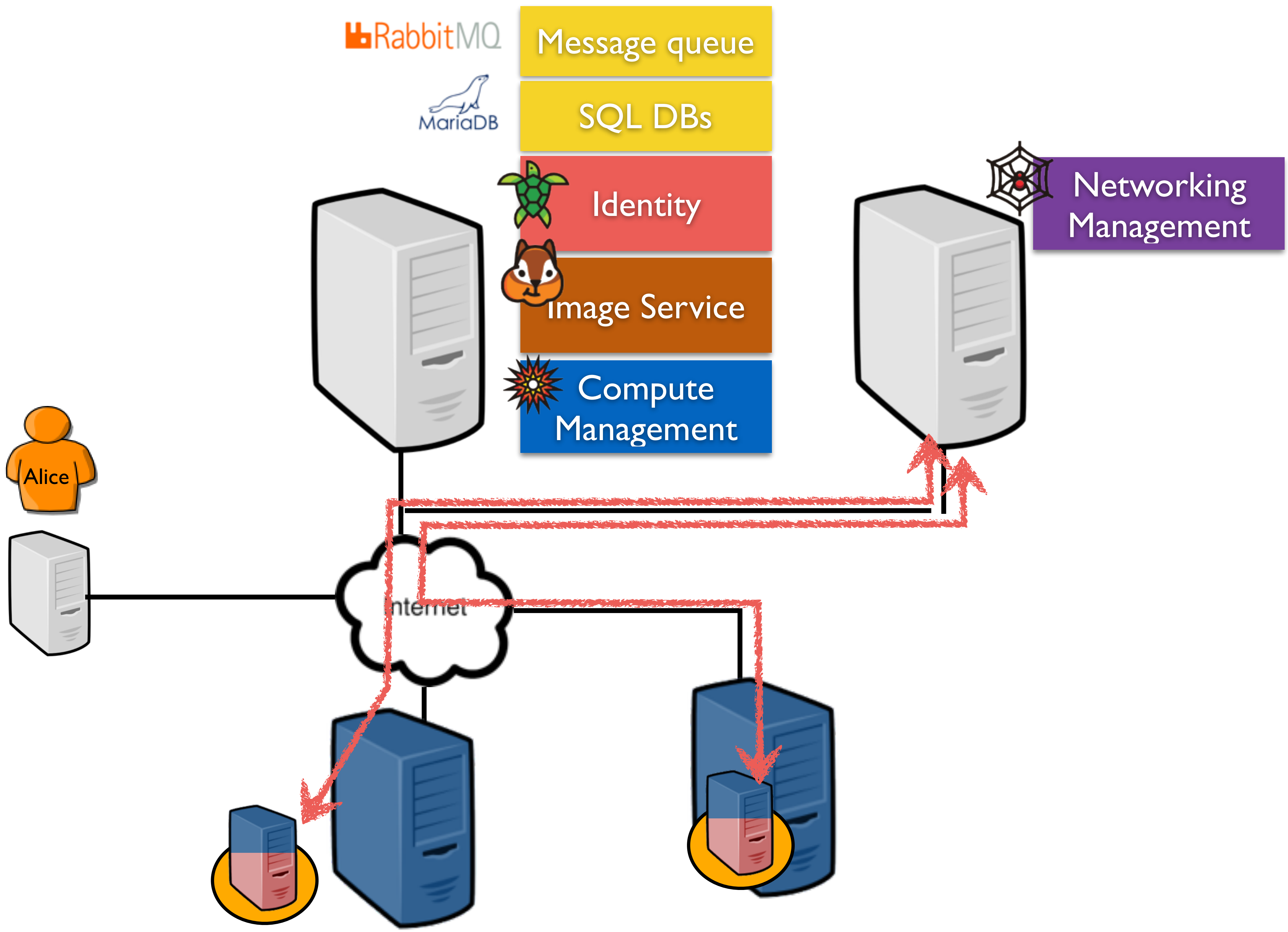


Fog/Edge Challenges

- Locality/Split Brain

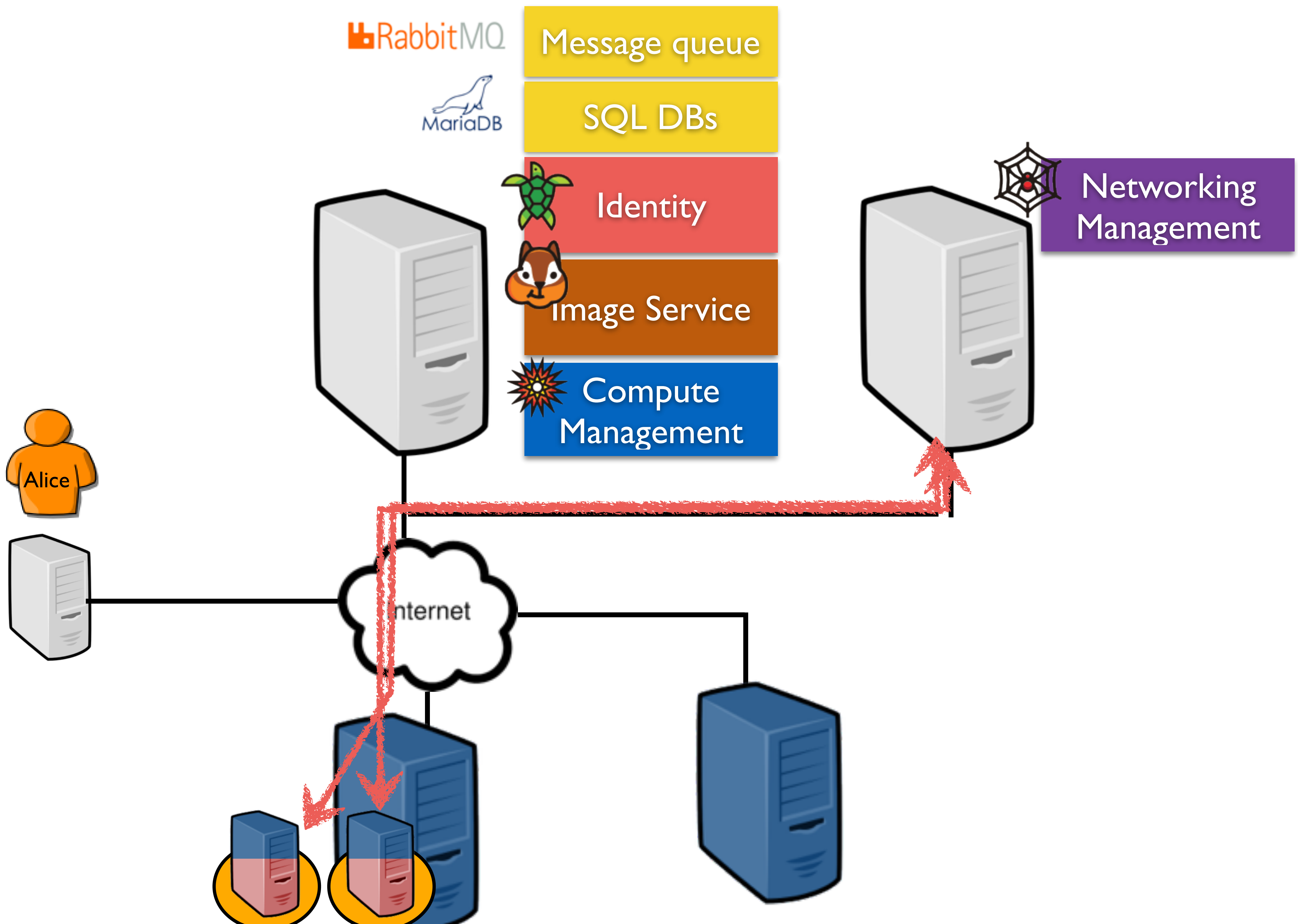


Neutron and Fog/Edge challenges



East/West - Different project networks

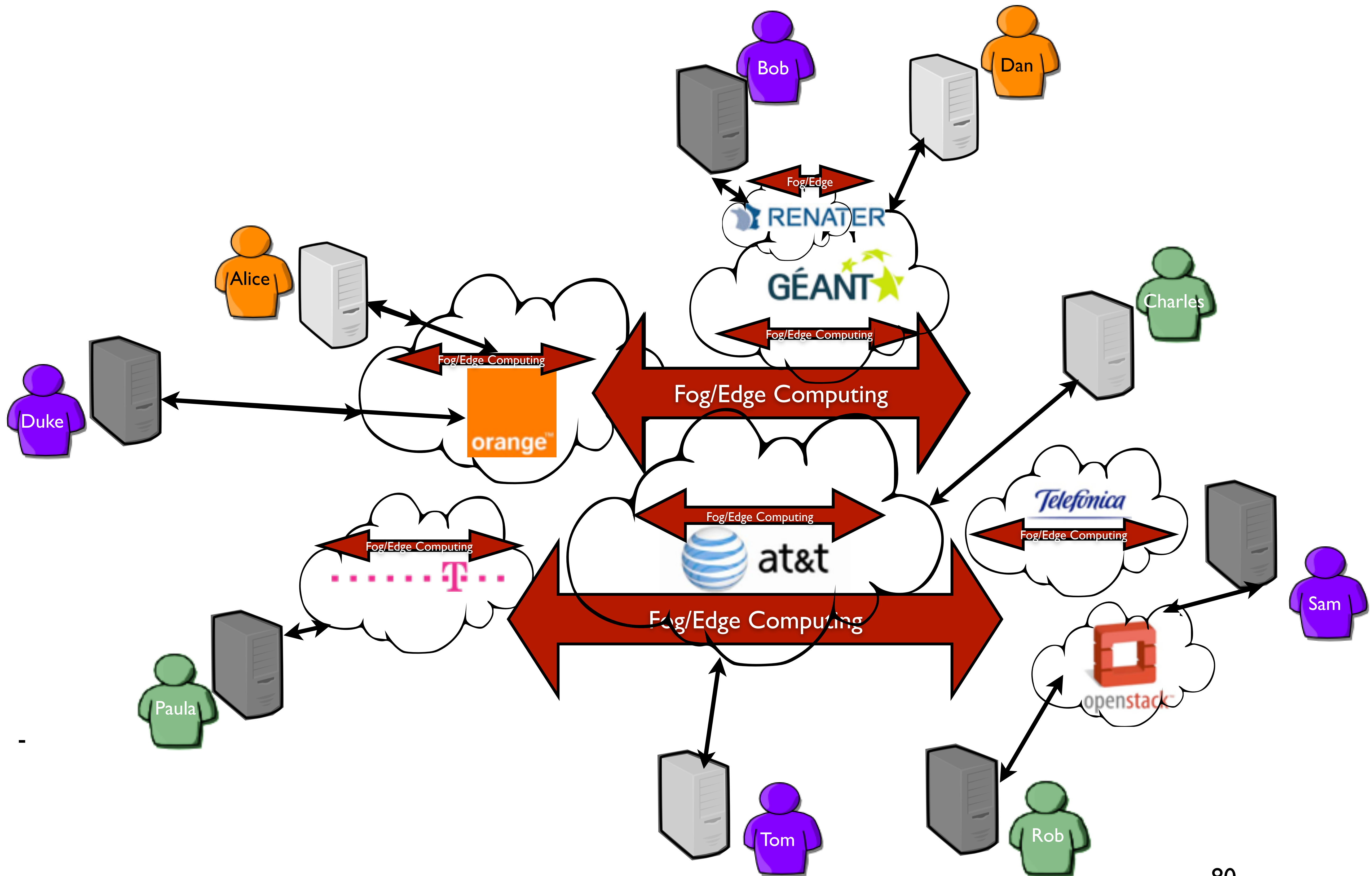
Neutron and Fog/Edge challenges



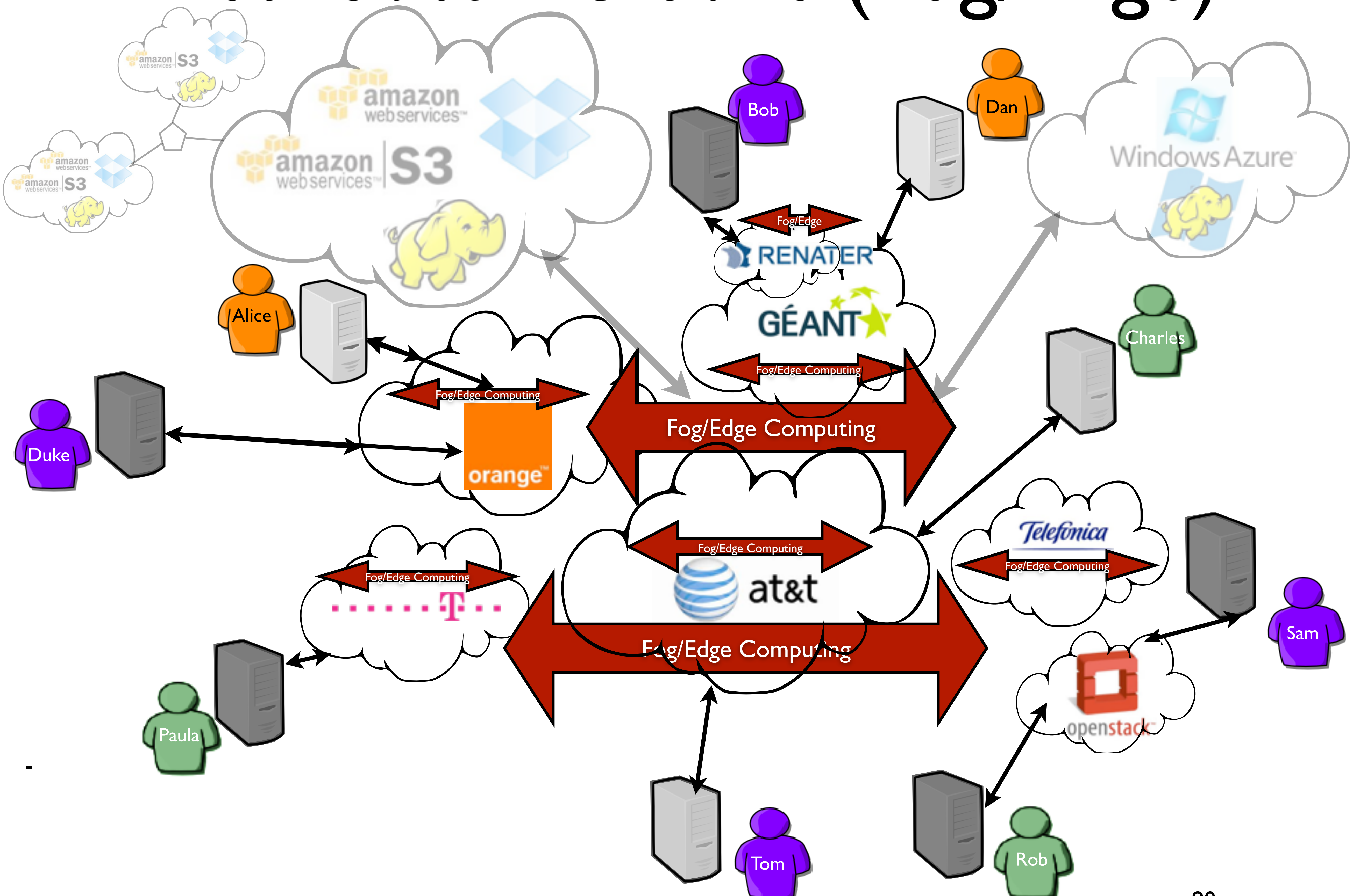
East/West - Different project networks

Takeaway Message

Distributed Clouds (Fog/Edge)



Distributed Clouds (Fog/Edge)



Clouds hide the infrastructure…
….by adding more layers !

and someone else's network

Clouds hide the infrastructure…
….by adding more layers !

Thanks

Utility

~~Cloud~~ Computing technology is changing every day

- How developers should develop new applications to benefit from geographically distributed infrastructures.
- How to locate hardware/software components?
- …

Do not hesitate to push the boundaries

http://beyondtheclouds.github.io/

We have Internship Positions

adrien.lebre@inria.fr

Bibliography

- In a Nutshell - How OpenStack Works
  A bit deprecated but good entry point to have a first idea in less than 10 minutes
  http://vmartinezdelacruz.com/in-a-nutshell-how-openstack-works/
- OSONES SLIDES
  Rich (almost up-to-date)
  https://github.com/Osones/formations
- OpenStack official Documentation
  Complete and up-to-date (at least yesterday ;-))
  https://www.openstack.org/software/