

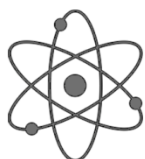


DEPARTMENT OF
COMPUTER SCIENCE
THAMMASAT UNIVERSITY



vasabiLab
VIRTUALIZATION ARCHITECTURE AND
SCALABLE INFRASTRUCTURE LABORATORY

Thammasat University

Science  **Cloud**



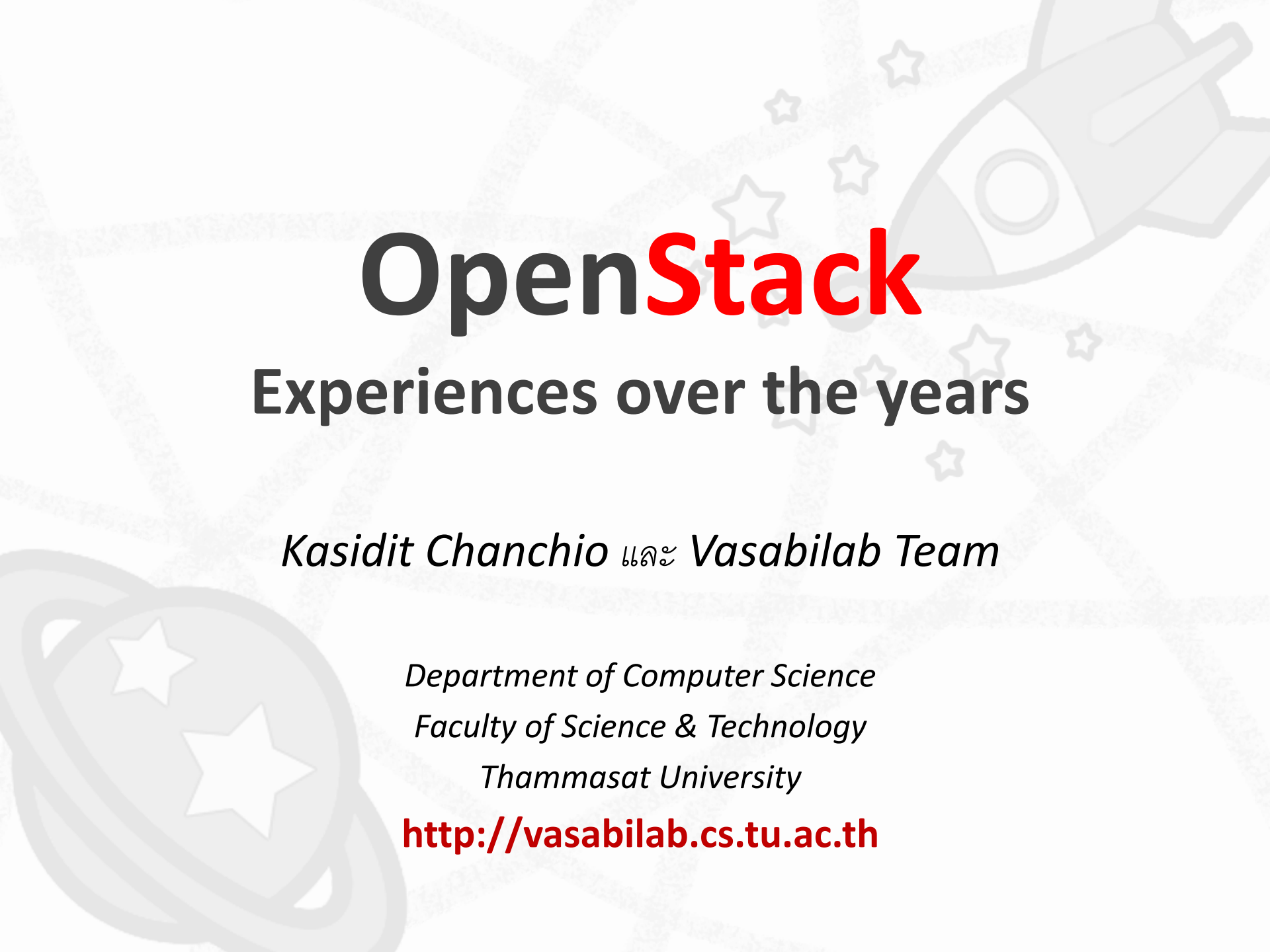
OpenStack

Experiences over the years

vasabilab.cs.tu.ac.th

Thammasat University





OpenStack

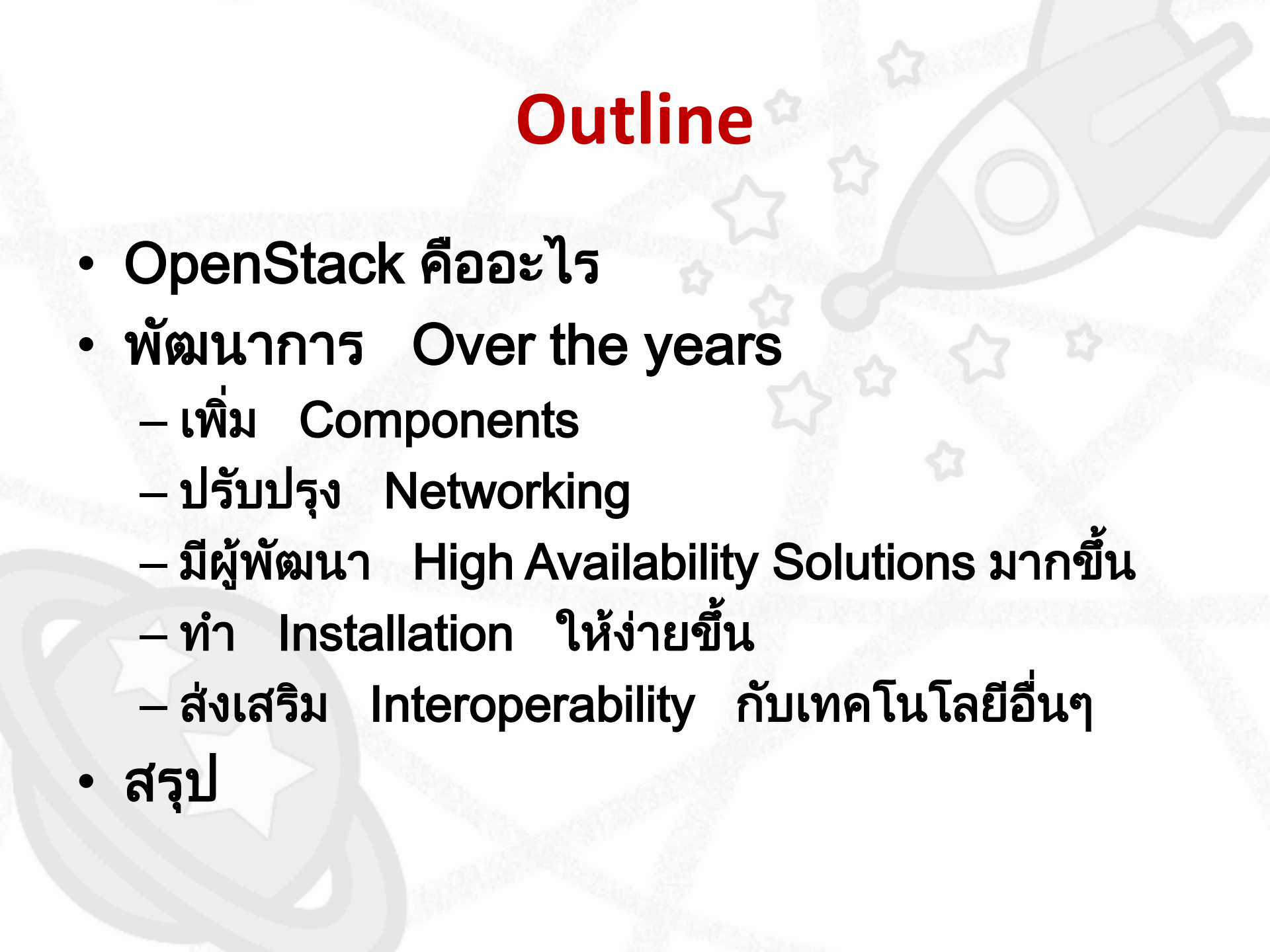
Experiences over the years

Kasidit Chanchio และ Vasabilab Team

*Department of Computer Science
Faculty of Science & Technology
Thammasat University*

<http://vasabilab.cs.tu.ac.th>

Outline



- OpenStack คืออะไร
- พัฒนาการ Over the years
 - เพิ่ม Components
 - ปรับปรุง Networking
 - มีผู้พัฒนา High Availability Solutions มากขึ้น
 - ทำ Installation ให้ง่ายขึ้น
 - ส่งเสริม Interoperability กับเทคโนโลยีอื่นๆ
- สรุป

The background features a stylized illustration of a rocket ship in the upper right corner, pointing towards the top right. It is surrounded by several five-pointed stars of varying sizes. In the lower left corner, there is a planet with a ring system and two large stars on its surface. The entire scene is set against a light gray background with faint, intersecting lines forming a grid-like pattern.

OpenStack คืออะไร

แนะนำ OpenStack

- ระบบ OpenStack เป็นซอฟต์แวร์บริหารจัดการทรัพยากรประมวลผลแบบกลุ่มเมฆ Cloud Management Software แบบ Open Source ที่ได้รับความนิยมเป็นอย่างมาก

<http://www.openstack.org>

- Version ปัจจุบันชื่อว่า OpenStack Mitaka (ปีที่ 6)
- เริ่มต้นเป็น IaaS แต่ในปัจจุบันเป็นมากกว่านั้น
- อนุญาตให้ผู้ใช้สร้าง Public Cloud เพื่อให้บริการ Cloud ในวงกว้าง
- อนุญาตให้ผู้ใช้สร้าง Private Cloud ใน

แนะนำ OpenStack

- มีผู้เข้าร่วม contribute codes เป็นจำนวนมาก รวมทั้งบริษัทใหญ่ๆ เช่น vmware IBM, Red-hat, Dell, Mirantis, HP, Google อื่นๆ
- มี OpenStack Foundation เป็นผู้บริหารจัดการ
- มีการจัด Event เพื่อเผยแพร่ Software อย่างต่อเนื่อง
- มีบริษัทที่นำไปใช้งานจริงในระดับ Production เพิ่มขึ้น
- ในปัจจุบันมี Vendor หลายแห่ง Offer OpenStack services และระบบอื่นที่เกี่ยวข้อง

Features

- ให้บริการจัดสรรทรัพยากรหน่วยประมวลผล เช่น virtual machines และ container และเครื่องจริง
- ให้บริการจัดการระบบเครือข่าย virtual networks
- ให้บริการจัดการระบบจัดเก็บข้อมูล virtual storages
- ให้บริการประสานการปฏิบัติงานของ ส่วนประกอบที่หลากหลาย
- ในให้บริการทั้งแบบ IAAS PAAS และ

The OpenStack Community 6 years in



who uses openstack?



Bloomberg



at&t



VISA



ebay

Walmart 









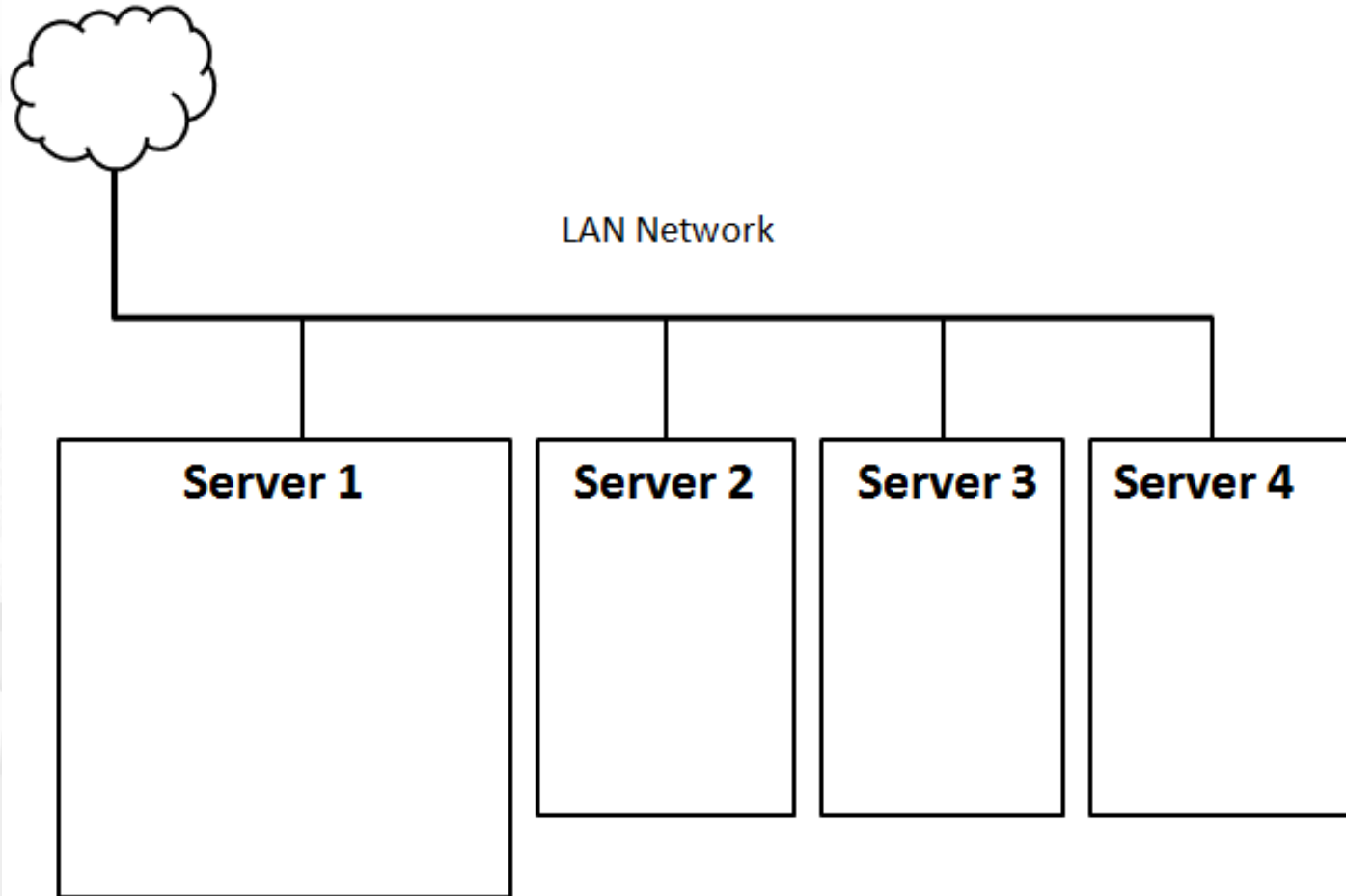


97% OF USERS SAY

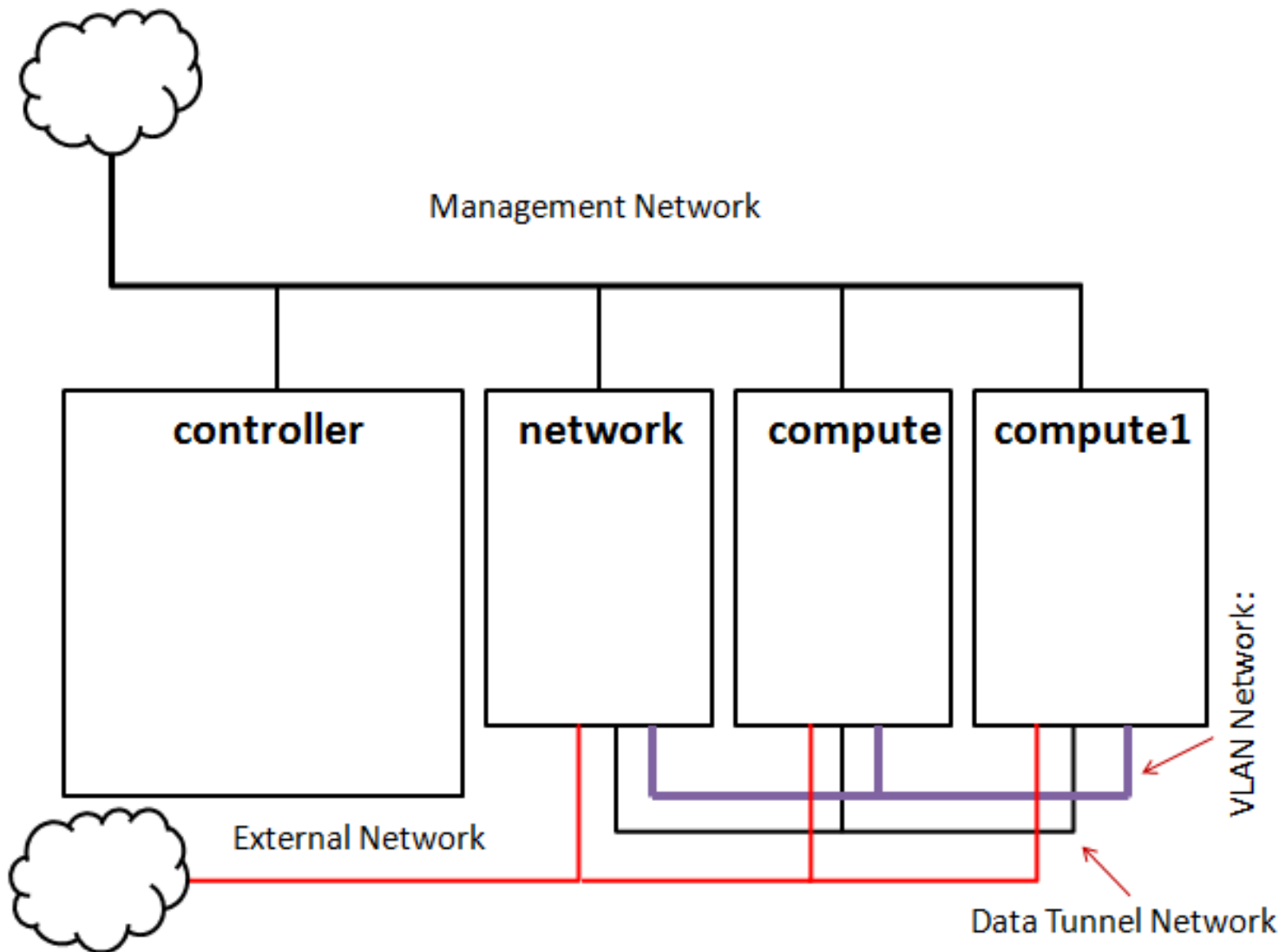
**“STANDARDIZING ON THE SAME
PLATFORM AND APIS THAT POWER
A GLOBAL NETWORK OF PUBLIC
AND PRIVATE CLOUDS”**

IS A TOP 5 REASON FOR ADOPTING

เริ่มต้นจาก Servers และ LAN



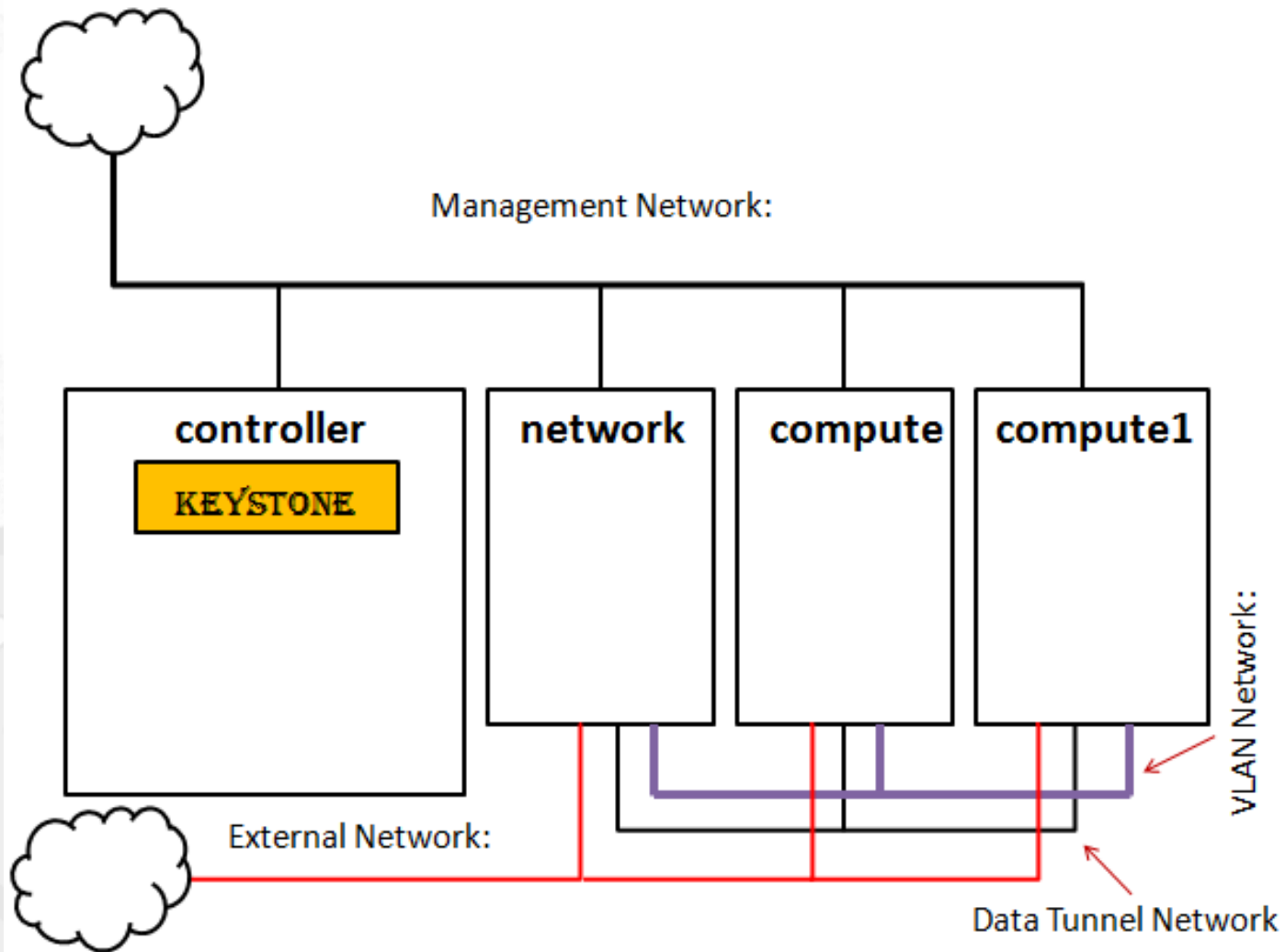
สร้าง Physical Layer



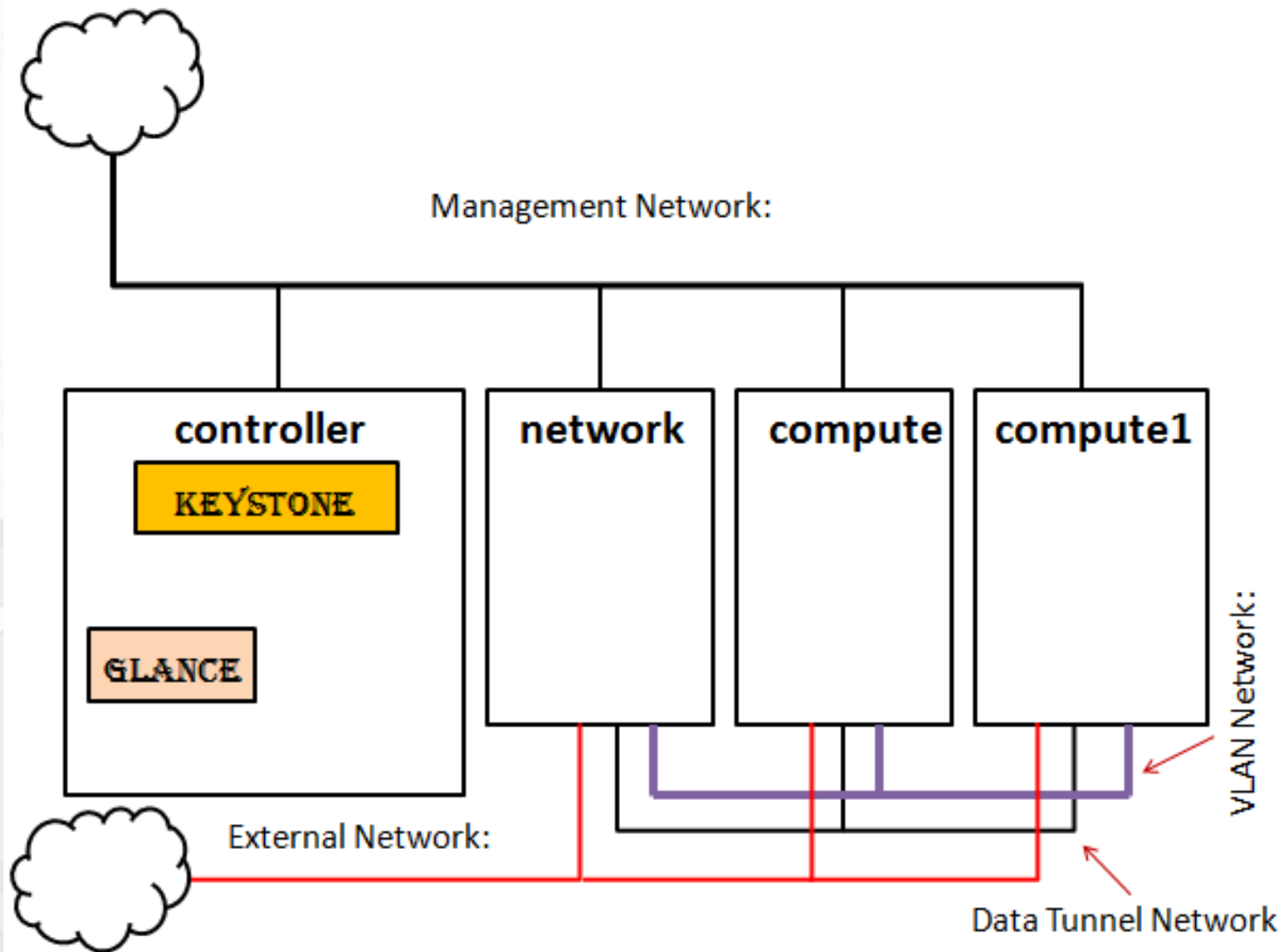
Core Components

- Keystone: ให้บริการ authentication
- Glance: ให้บริการจัดการ virtual machine image & snapshots
- Nova: ให้บริการจัดการ virtual machine
- Neutron: ให้บริการ virtual network
- Horizon: ให้บริการ Web-based User Interface
- Cielometer: ให้บริการวัดปริมาณการใช้ resources และส่ง Notify ตามเงื่อนไขที่ผู้ใช้งานกำหนด
- Swift: ให้บริการ object storages

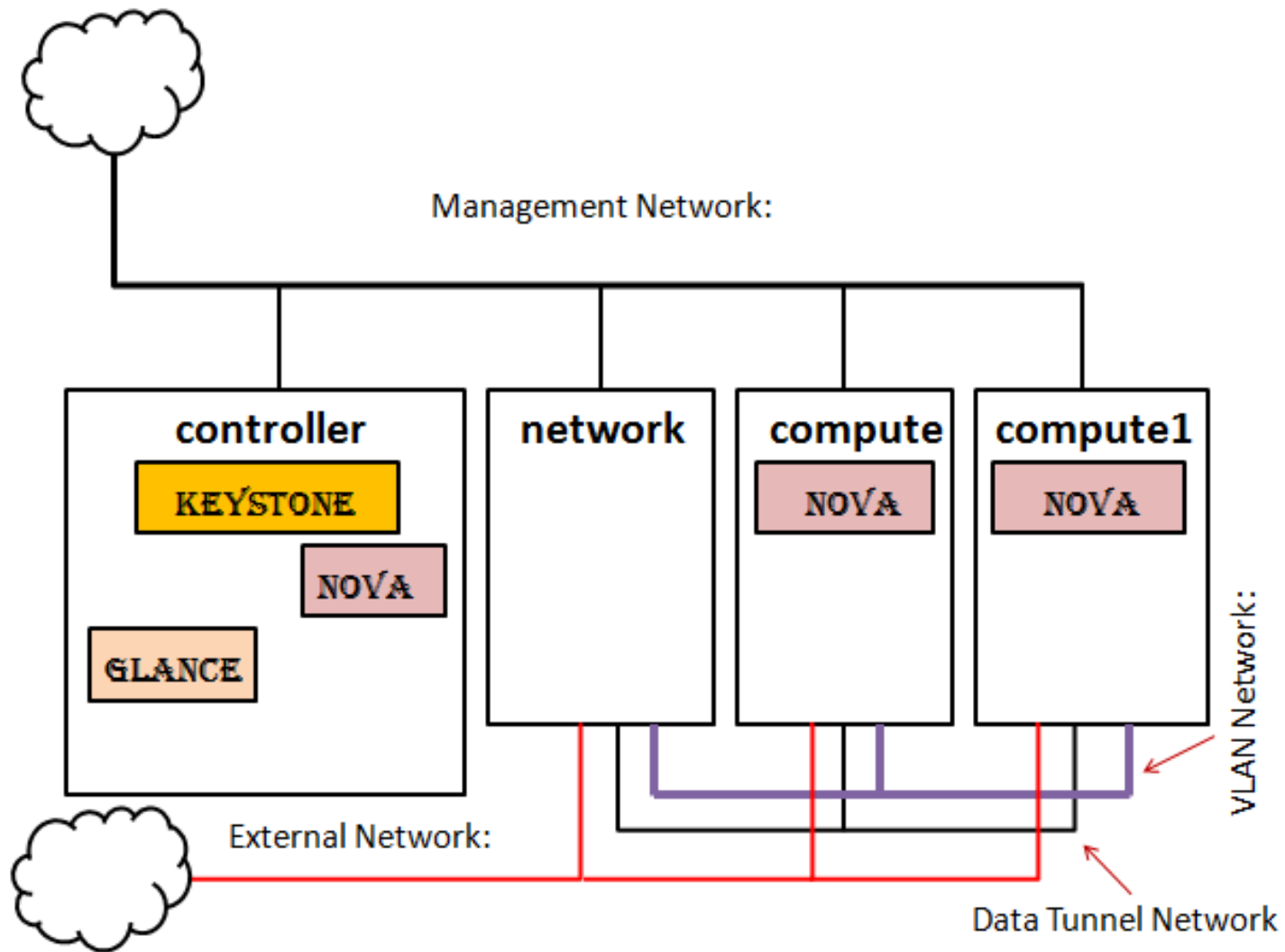
KEYSTONE



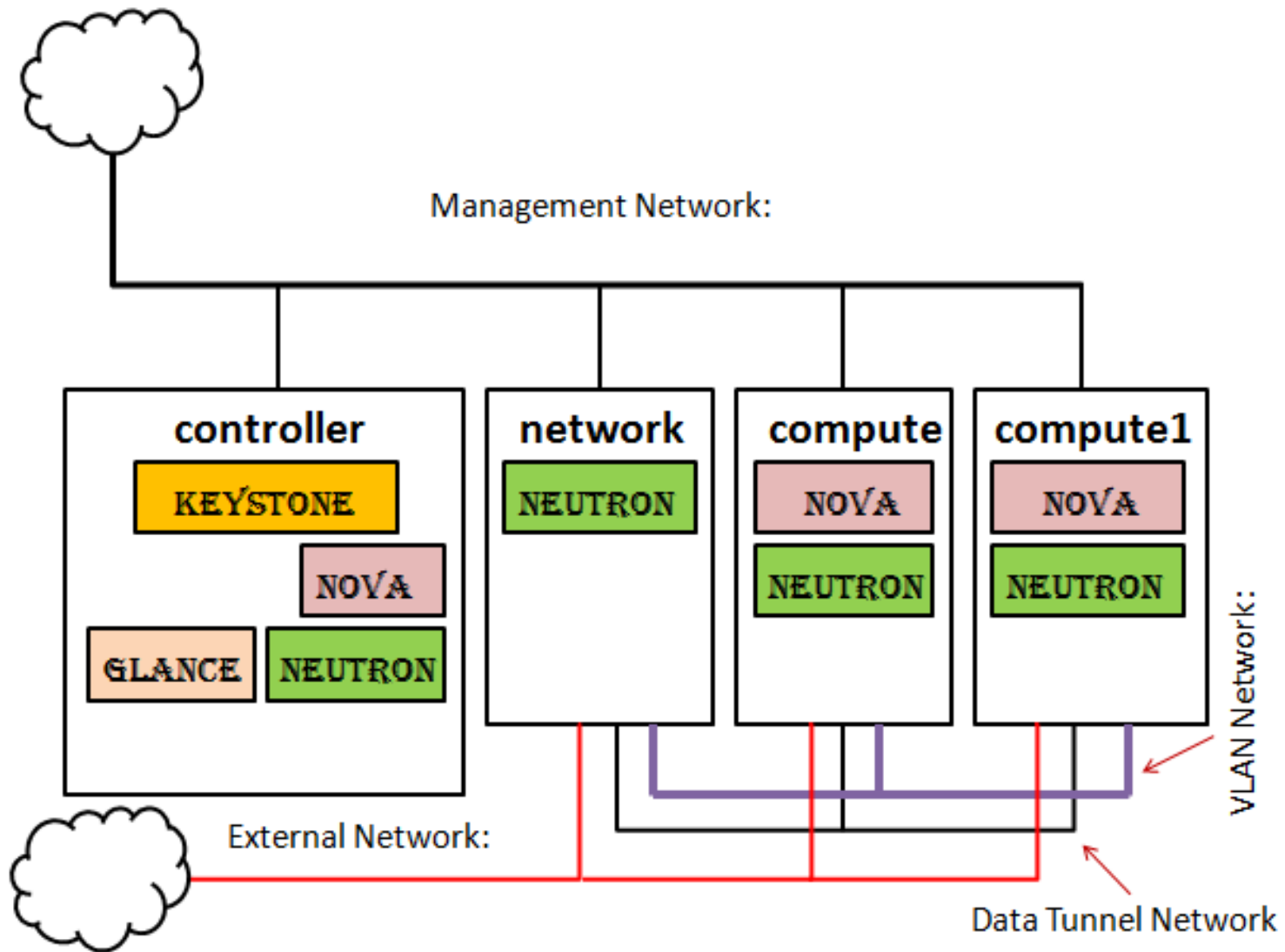
GLANCE



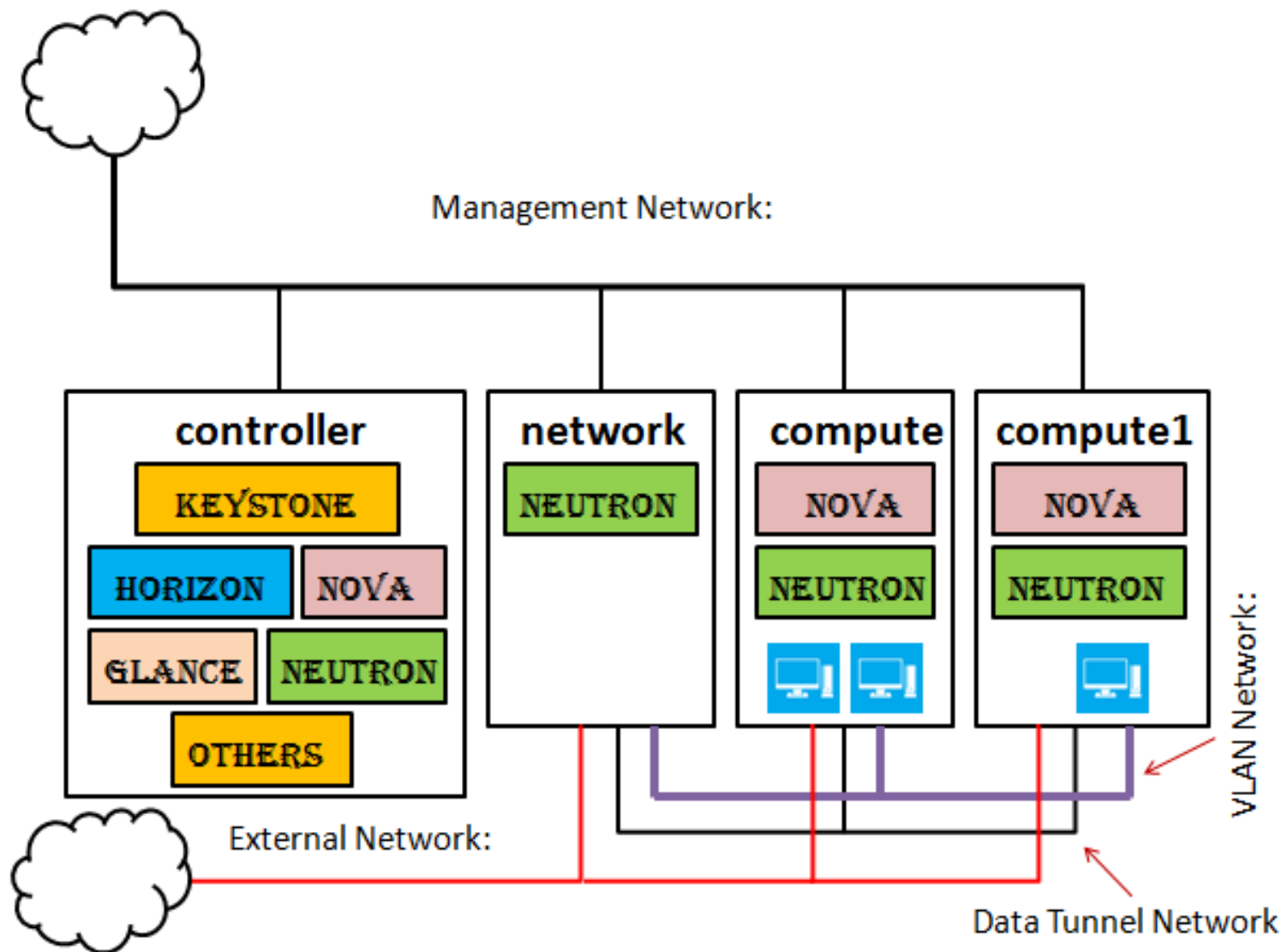
NOVA



NEUTRON



HORIZON AND OTHERS



หลักการของ OpenStack

- ความปลอดภัยที่สำคัญที่สุด
 - การใช้บริการทุก component ต้องผ่านการ authentication โดย keystone
 - มี security group หรือ Firewall rules สำหรับทุก virtual machines
- การสั่งงานทุก component ต้องมีมาตรฐานเดียวกัน
 - ทุก component ต้องมี API ที่ชัดเจน
 - การสั่งงานและประสานงานระหว่าง component ทำผ่าน API เท่านั้น

หลักการของ OpenStack (cont)

- ระบบต้องทนทานต่อความผิดพลาด
 - การสื่อสารทำผ่าน Persistent Message Queue
 - Component กลับมาทำงานได้ตามปกติในภายหลัง
 - มีข้อเสนอแนะในการทำ High Availability
- ปฏิบัติงานร่วมกับ software อื่นได้
 - มี plugin drivers สำหรับติดต่อกับ Hardware และ software แบบต่างๆ
 - เชื่อมต่อและทำงานร่วมกันกับเทคโนโลยีอื่นๆได้ เช่น SDN

Over the years

- สร้างบริการใหม่จากบริการที่มีอยู่
 - ให้บริการ IaaS PaaS SaaS และอื่นๆด้วย
- เพิ่มความสามารถและความน่าเชื่อถือของบริการที่มีอยู่
- เพิ่มการทำ High Availability ให้ OpenStack Services
- ทำให้ Document ดีขึ้นและตรงกับความเป็นจริง
- ทำให้การ Installation ง่ายขึ้น
- ส่งเสริมการ Integration กับเทคโนโลยีอื่นๆ

เพิ่ม **Components** ใหม่

มี **Components** ให้บริการหลากหลาย

- ในปัจจุบันมี Project สำหรับพัฒนา Component ใหม่ ๆ กว่า 19 Projects (เข้าชม list ได้ที่ OpenStack Web site)
- บน OpenStack web site แสดง Maturity ของแต่ละ Project
- Components ที่มีอยู่ได้รับการ Bug Fix อย่างต่อเนื่อง
- Basic Component ที่ทุกๆ component ใช้ได้แก่
 - Keystone, Horizon, Ceilometer

ไม่ใช่แค่ IaaS อีกต่อไป

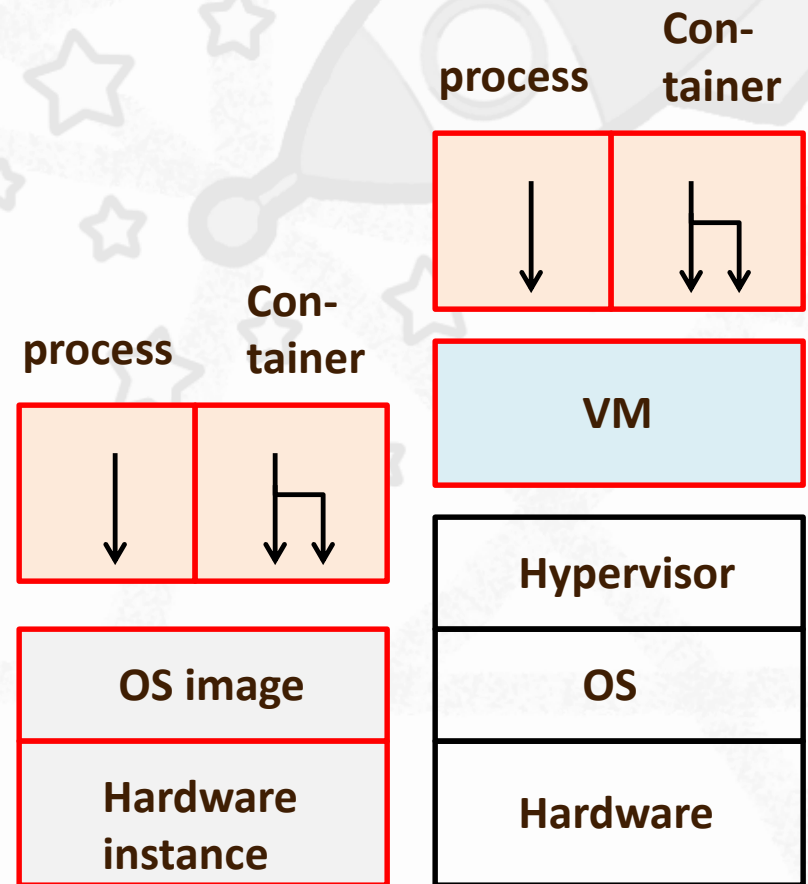
- Heat Orchestration: จัดการการประสานงานระหว่างสิ่งต่างๆ
- IroniC: ให้บริการสร้างและบริหารจัดการเครื่องจริง
- Sahara: ให้บริการ map reduce processing
- Magnum: ให้บริการ บริหารจัดการ container
- Trove: ให้บริการ database
- Manila: ให้บริการ shared file system
- Murano: ให้บริการ Application Catalog

Heat: ประสานทุกส่วน

- ให้บริการให้ผู้ใช้สามารถกำหนดโครงสร้าง Infrastructure ต่างๆที่จำเป็นสำหรับรัน Application รวมทั้งการเพิ่มลดแบบ auto-scaling
- ใช้บริการของ Keystone สำหรับ authentication และ Horizon แสดงภาพของ Template
- ผู้ใช้สร้าง Template (HOT หรือ CloudFormation format) เพื่อกำหนดสิ่งที่ตนต้องการในการรัน Applications
- กำหนด VMs และ Networks สำหรับรัน

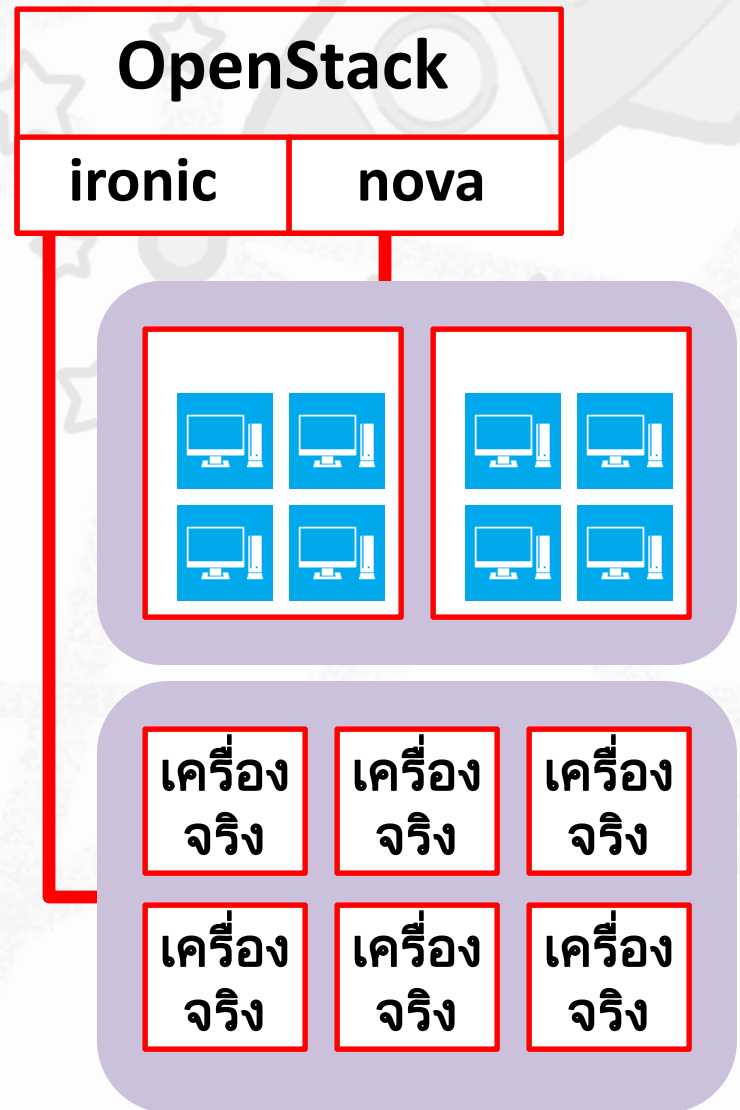
ขยายชนิดของ **Compute Instances**

- เริ่มจาก VM instance
- ขยายชนิดของ Instance เป็น เครื่องจริง
- รองรับ Instance ที่เป็น Process สำหรับประมวลผล เฉพาะทางเช่น MapReduce
- รองรับ Instance ที่เป็น Container (สามารถมองได้ เป็น ครอบครัวของ Processes ที่แยก cgroup, name space และ images)
 - รันบน VM



Ironic: ใช้งานเครื่องจริง

- ให้บริการบริหารจัดการเครื่องจริงสำหรับใช้งาน (แทนที่จะเป็น VM) ในระบบ Cloud
 - ใช้ Preboot Execution Environment (PXE) เพื่อ load OS image ให้กับเครื่องจริงผ่าน Network
 - ใช้ Intelligent Platform Management Interface (IPMI) จัดการเปิดปิดเครื่องจัดการ BIOS และกำหนดค่าพารามิเตอร์ต่างๆ
- ในปัจจุบันใช้ Flat Network

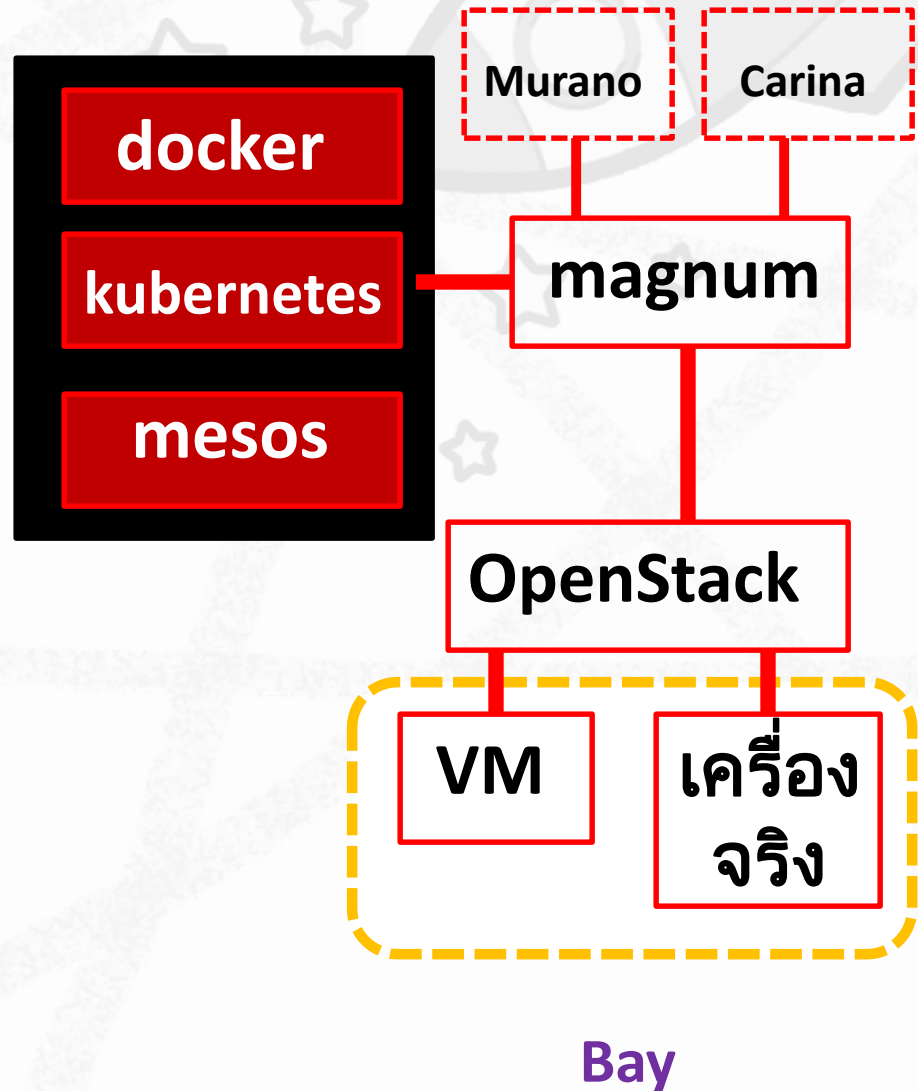


Sahara: รันโปรแกรม

- ให้บริการกำหนด Image และ Cluster รวมทั้ง input/output และ job สำหรับรัน Hadoop Map-reduce Applications
- ใช้บริการของ Keystone สำหรับ authentication และ Horizon สำหรับสั่งงาน และดู progress
- ใช้ Glance สำหรับ register Hadoop image ใช้ Nova สำหรับรัน Hadoop Cluster ใช้ Swift สำหรับเก็บ Job binary และ Input/Output ใช้ Ironic สำหรับรัน job บนเครื่องจริง

Magnum: รันหลากหลาย Containers

- ให้บริการ Container as a Service (CaaS) ที่อนุญาตให้ผู้ใช้รัน Container Technology ที่หลากหลาย ไม่ว่าจะเป็น Docker, Swarm, Kubernetes, Mesos บน OpenStack VM หรือ เครื่องจริง
- ใช้ Heat, Nova, Ironic, Neutron etc.
- Bay = กลุ่มของ Instances ที่ใช้รัน Container (รันหลาย Bay พร้อมกันได้)



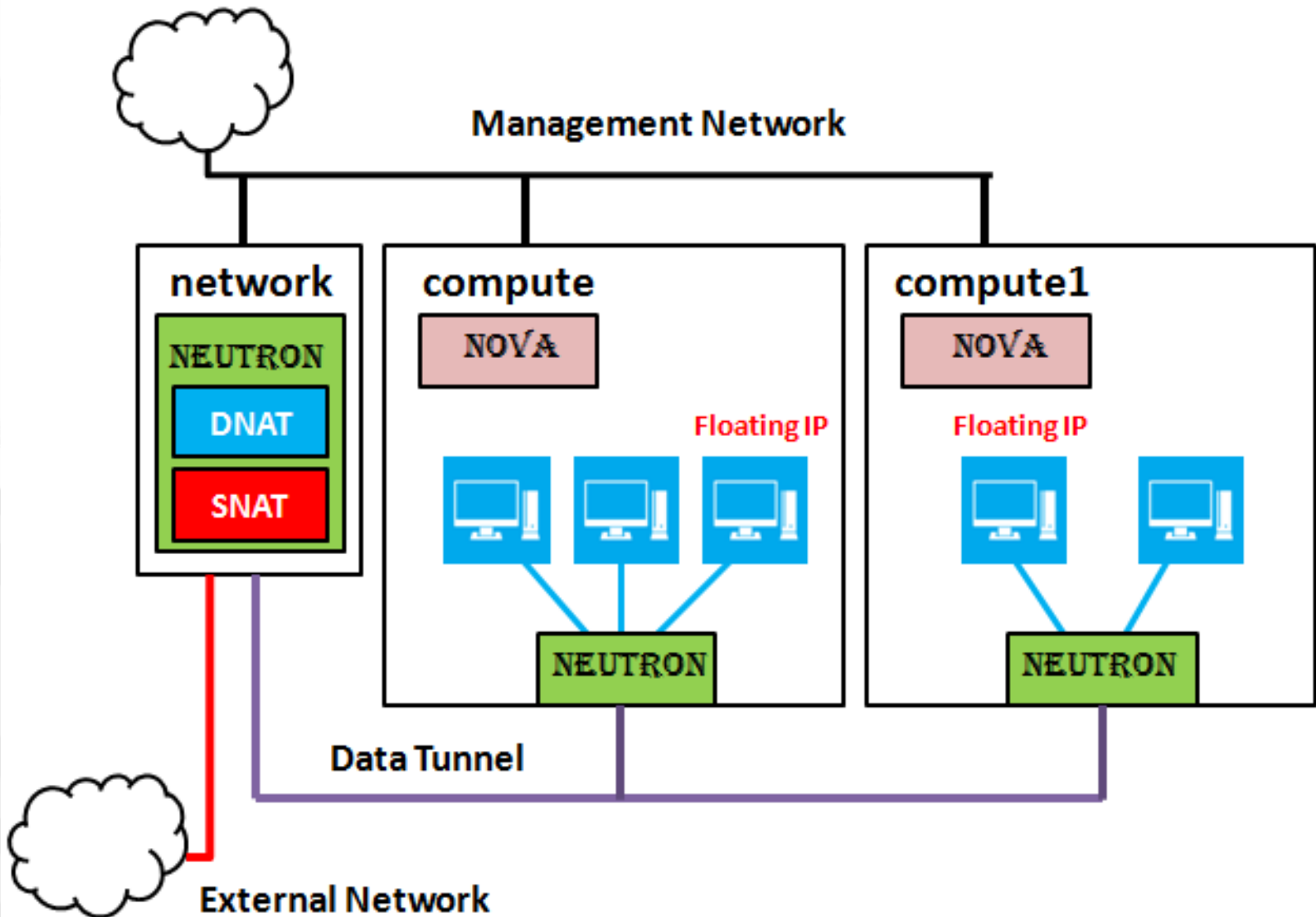


เพิ่มความสามารถของ **Core Component**
และเพิ่มบริการ **High Availability**

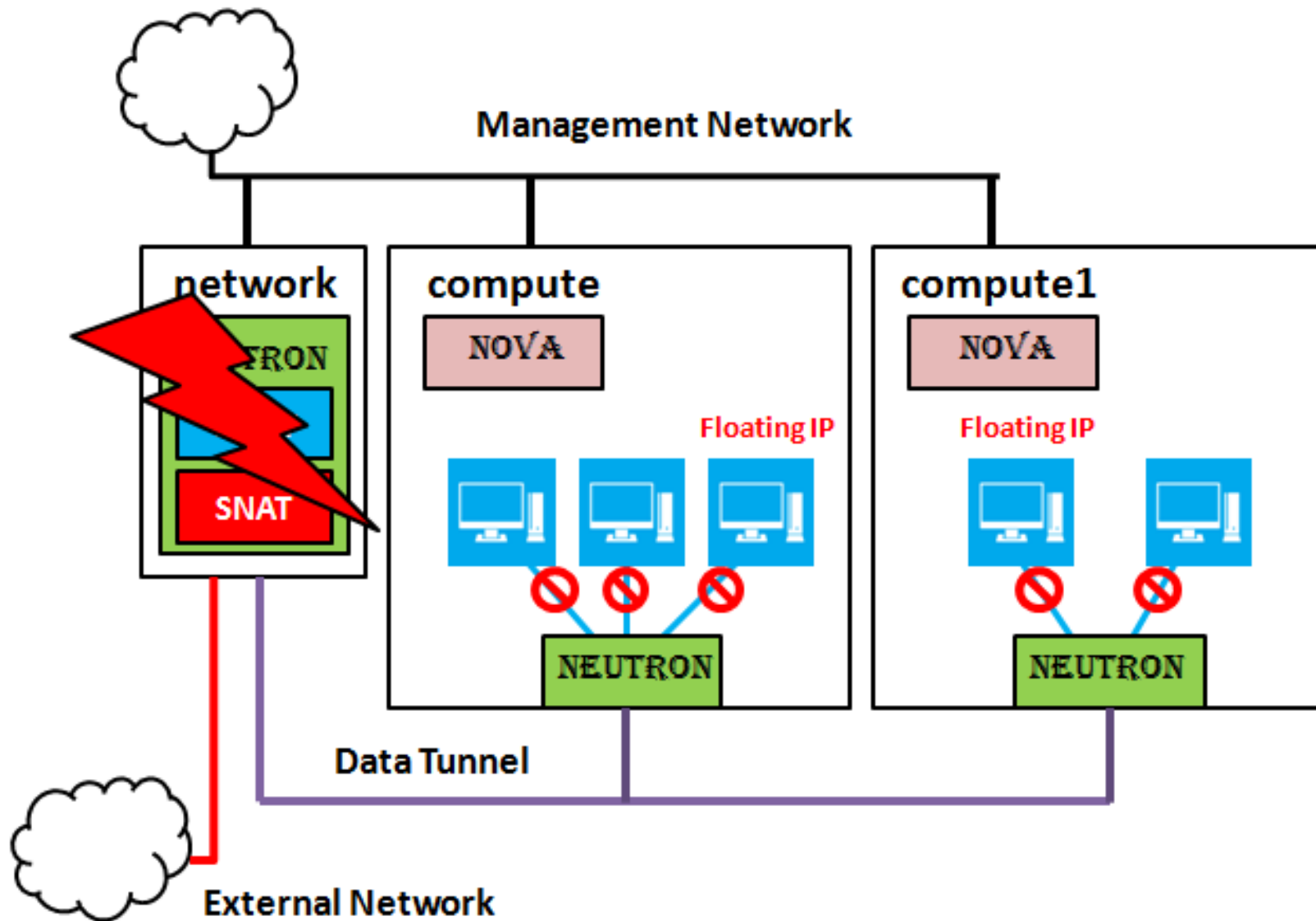
ความสามารถที่มากขึ้นของ **neutron**

- ปีที่แล้วมีบริการ self-service network, DNAT, SNAT
- OpenStack Liberty + Mitaka เพิ่มบริการ
 - Provider network + Self-service network,
 - VLAN network,
 - DVR (High Availability)
- เฉพาะด้าน LBaaS, QoS, BGP routing, High Availability DHCP, DNS, IPv6, RBAC, SR-IOV, SDN SFC, etc.
- ทุกๆ Component มีพัฒนาการอย่างต่อเนื่อง

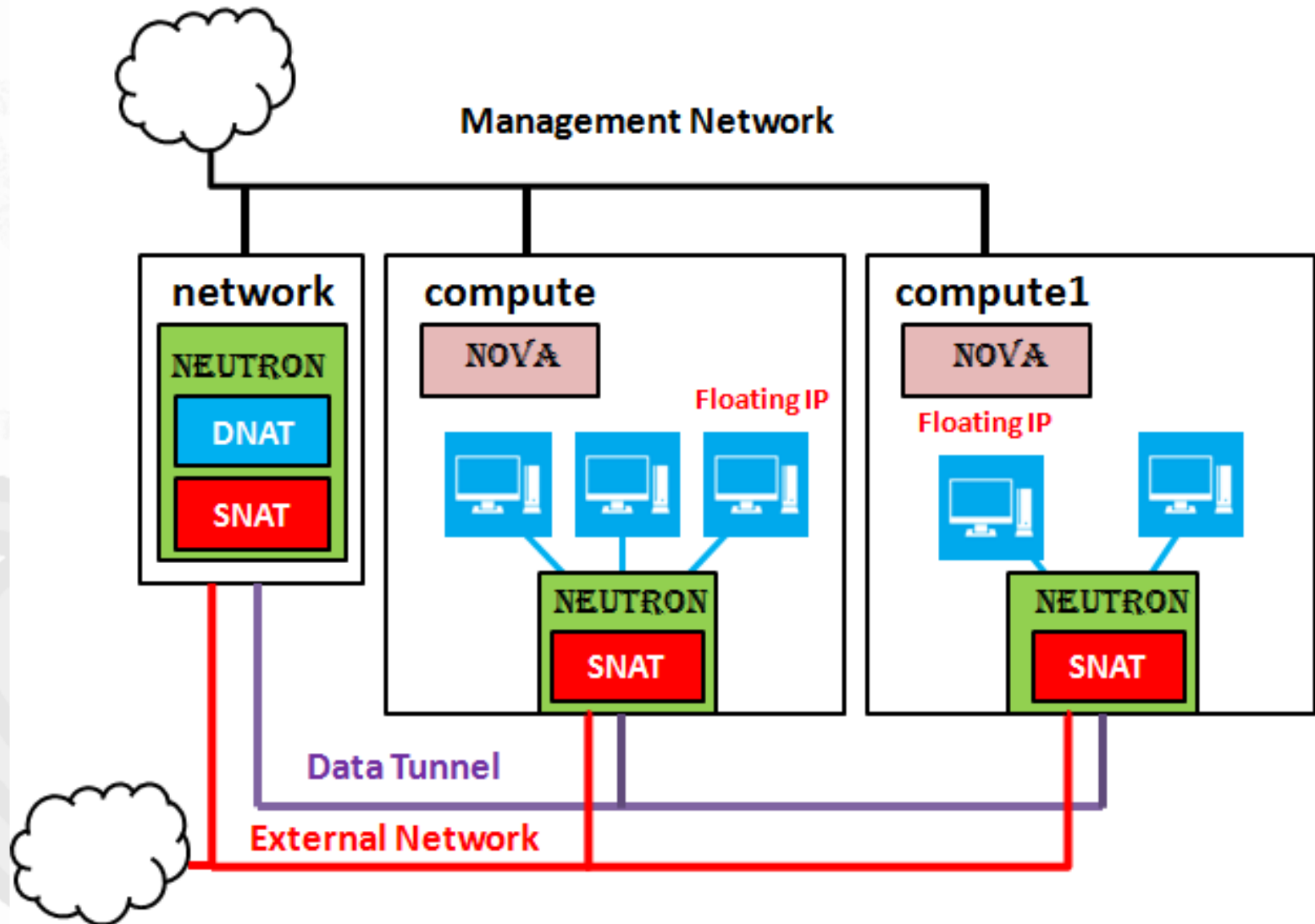
CLASSIC OVS NET



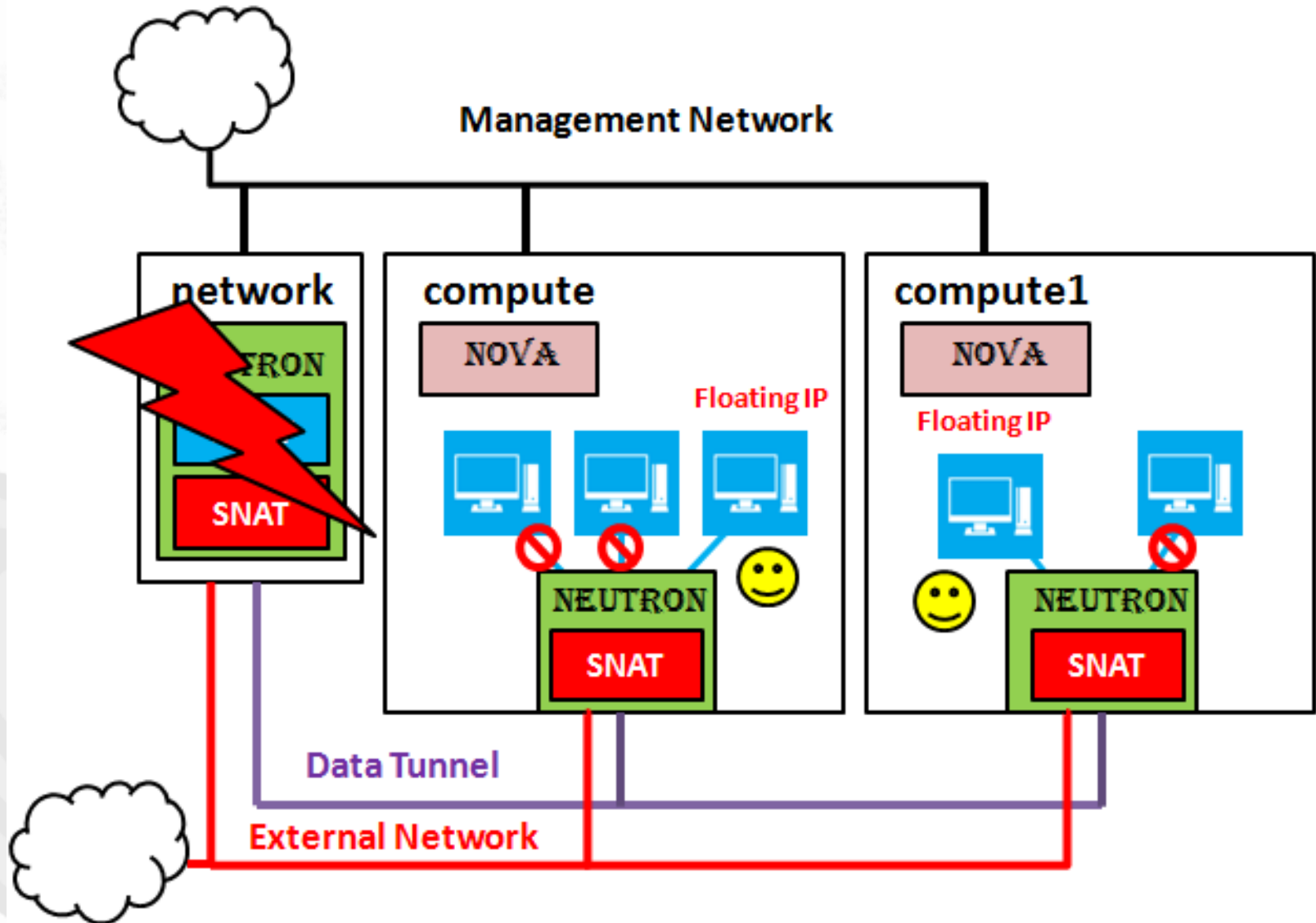
CLASSIC OVS NET



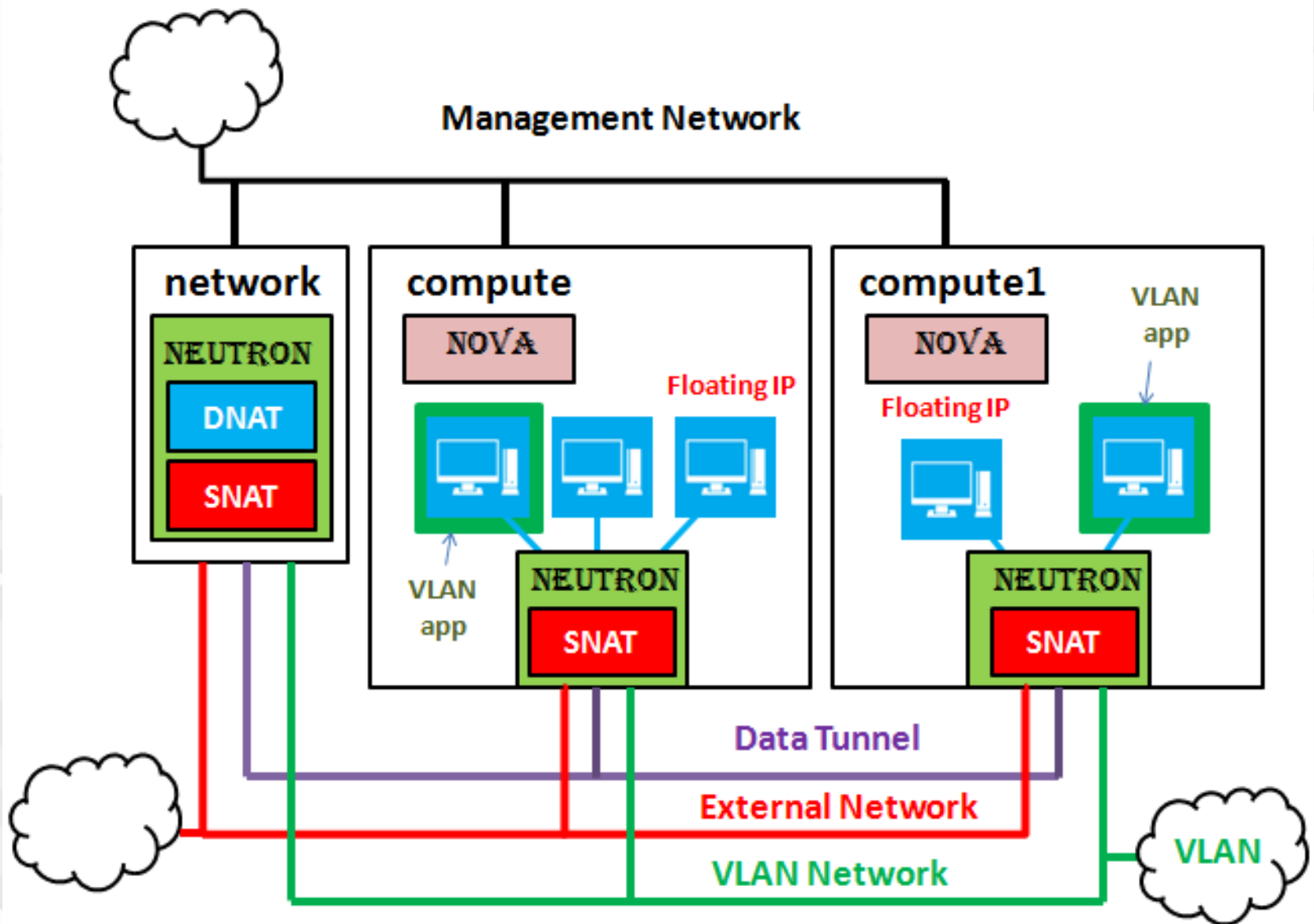
DVR NETWORK



DVR NETWORK



VLAN NETWORK





มีแนวทางให้บริการ **High Availability** ที่ชัดเจนขึ้น

หลักการ **High Availability**

- HA มีเพื่อลดเวลา Downtime และ ลดการสูญหายของข้อมูล (Data Loss): ยกตัวอย่างเช่น HA 99% หมายถึงมี Downtime ได้ 3.65 วันต่อปี ในขณะที่ HA 99.99% มี Downtime ได้ 52.56 นาทีต่อปี (สามารถทำได้จริงบน OpenStack อ้างอิง Official HA Guide)
- HA เป็นความพยายามขจัด Single-point of Failure ของระบบ
- Active/Passive เป็น HA แบบที่ในขณะหนึ่งมีเพียง Master Service เดียวให้บริการและจะ Activate Standby Service เมื่อ Master ดัน

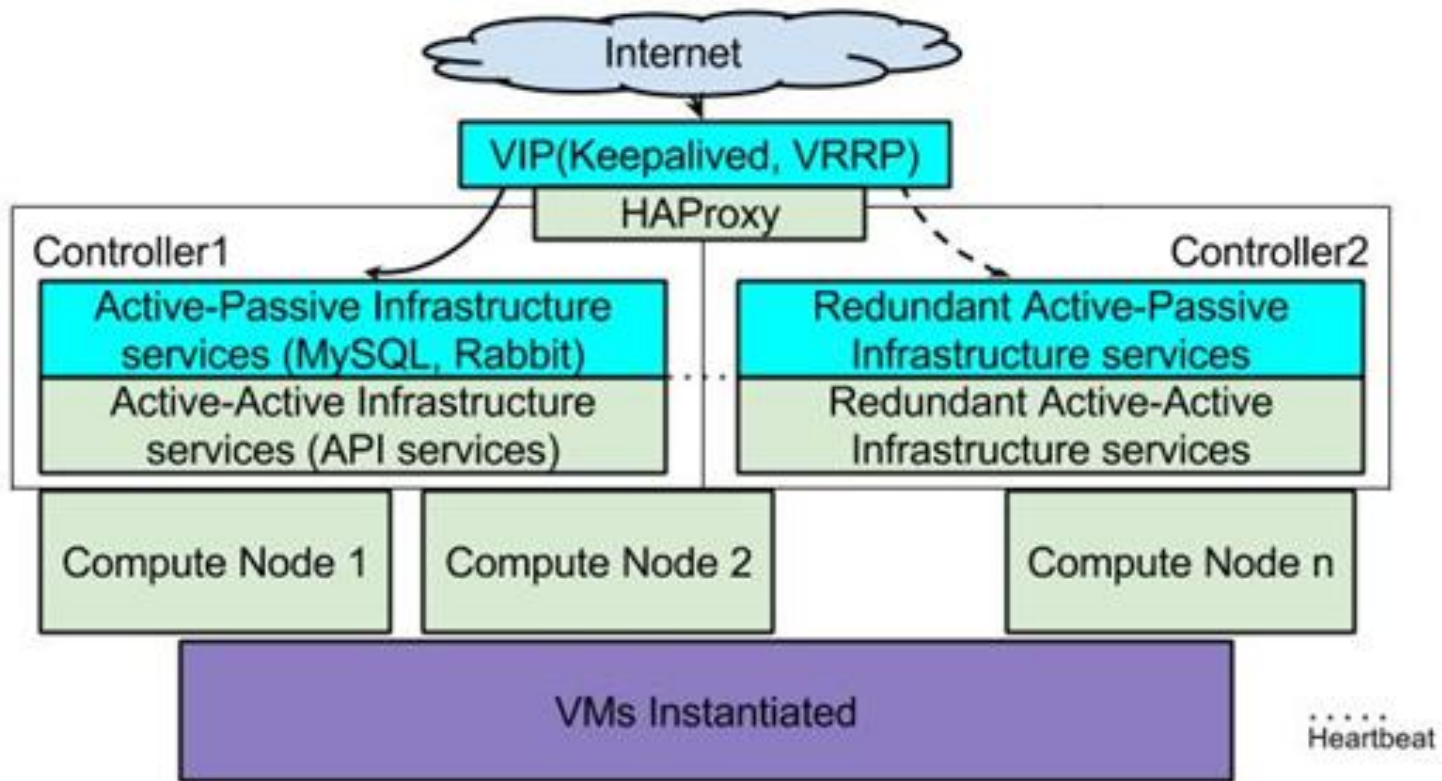
บริการ **High Availability**

- ได้รับประสบการณ์จริงจากการสร้าง High Availability ใน Production Deployment ของ Vendors ต่างๆใน 5 ปีที่ผ่านมา
- จาก official web site, OpenStack ระบุแนวทางในการให้บริการ HA สำหรับ OpenStack components แต่ไม่ได้ครอบคลุม guest vm (ผู้ใช้ต้องทำ image snapshot หรือ backup data หรือ task checkpointing หรือ VM checkpointing เอง)
- Controller Node และ Network Node: ระบุแนวทางติดตั้ง Active/Passive และ Active/Active สำหรับ service

HA สำหรับ Controller Services

- แยกพิจารณาจากลักษณะของ OpenStack Services ว่าเป็น Stateless หรือ Stateful ดังนี้
 - Stateless: Web Server, OpenStack API, Nova scheduler, etc.
 - Stateful: MySQL, RabbitMQ, etc.
- และเลือกให้บริการแบบ Active/Passive หรือ Active/Active สำหรับ Services แต่ละอย่าง
- ในปัจจุบันมี Solutions จากหลาย Vendors อาทิเช่น Rackspace, Mirantis, Cisco, RedHat, etc.

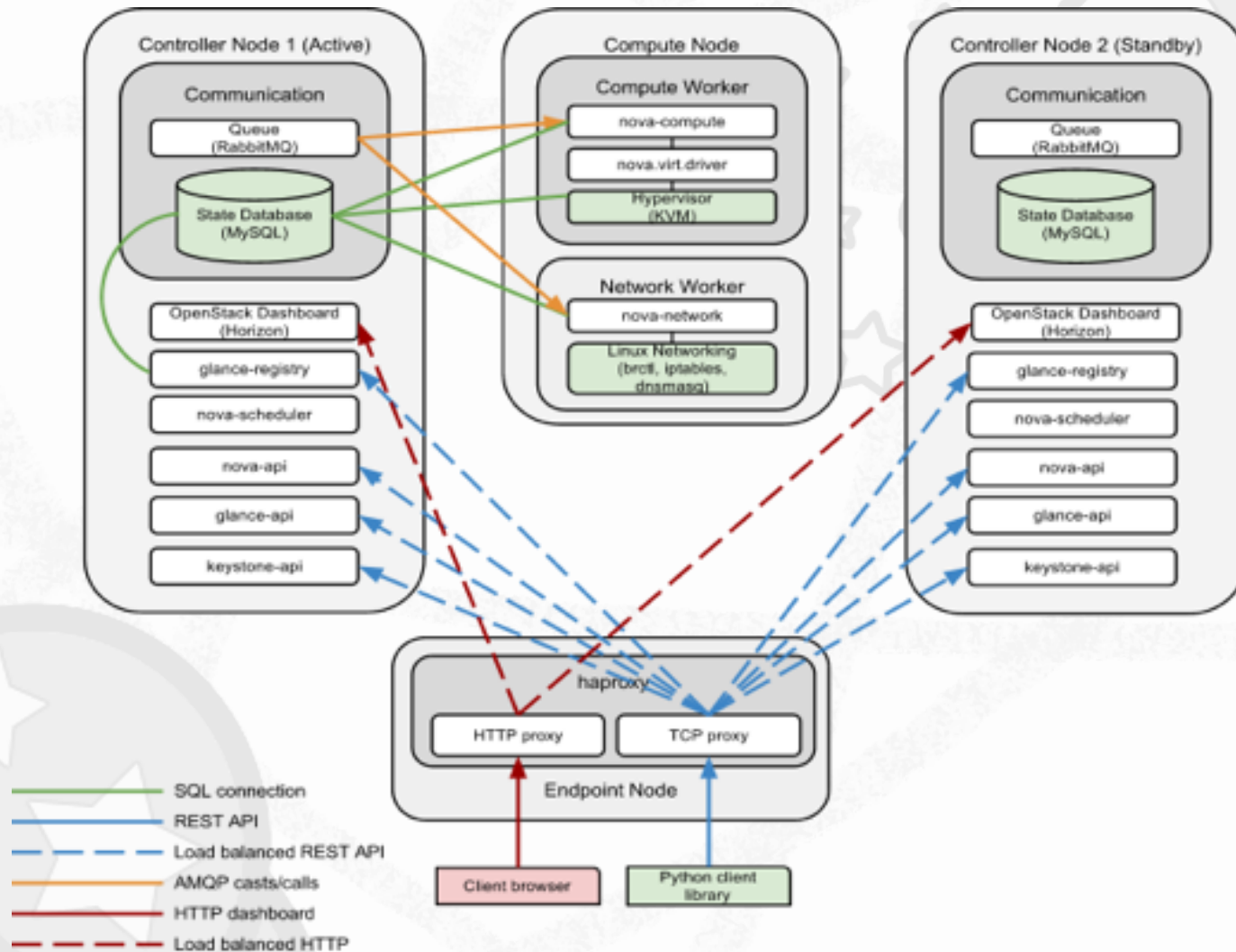
Rackspace HA solution



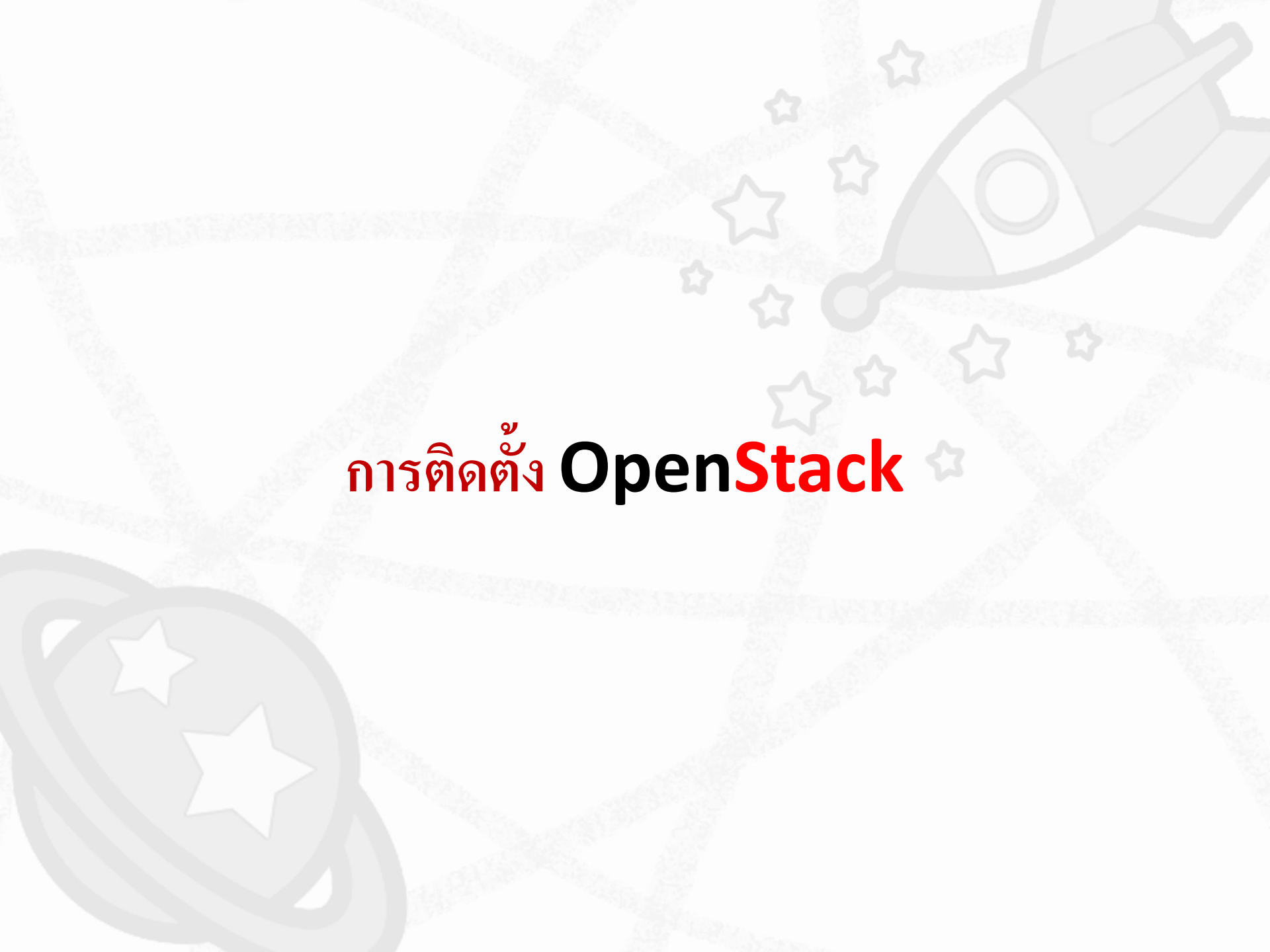
อ้างอิงจาก :

<http://blog.rackspace.com/implementing-high-availability-ha-for-rackspace-private-cloud>

Mirantis HA solution



อ้างอิงจาก: <https://www.mirantis.com/blog/understanding-options-deployment-topologies-high-availability-ha-openstack/>



การติดตั้ง **OpenStack** ☆

ภาพรวมของ **OpenStack** Installation

- ในปัจจุบันมีหลายวิธีในการติดตั้ง OpenStack
 - Red Hat RDO, Mirantis Fuel, OpenStack Ansible
- แต่ทุกวิธีตั้งอยู่บนพื้นฐานของการติดตั้ง Core Component ที่ระบุใน OpenStack Official Installation Guide
- พื้นฐานที่สำคัญที่สุดคือ ความรู้เกี่ยวกับ Core Components ของระบบ และที่ไปที่มาของ component เหล่านั้น
 - ทำให้สามารถปรับเพิ่มลด Features ได้ตาม

OpenStack Installation Guide

- ในปัจจุบัน OpenStack Installation Guide ดีและชัดเจนขึ้นมาก
- เน้นตัวอย่างการติดตั้งระบบขนาดเล็กที่ประกอบด้วย 2 nodes
- แต่กลับขาดขั้นตอนที่ชัดเจนในการติดตั้งแบบ Classic OVS network และแบบ DVR network และแบบอื่นๆ
 - ต้องหาข้อมูลจาก document ของ OpenStack KILO
 - หาจาก Contribute ของผู้ใช้

การ Installation ที่ง่ายขึ้น

- นำ container มาใช้ทำให้การติดตั้งง่ายขึ้น
 - OpenStack Ansible: นำ Docker Technology มาทำให้การติดตั้ง OpenStack Component ง่ายขึ้น
- นำ OpenStack Ironic มาใช้ทำให้การติดตั้งง่ายขึ้น
 - Triple O: เริ่มจาก OpenStack เล็กๆ และใช้ OpenStack Ironic ติดตั้ง OpenStack images บน Physical Nodes และได้ OpenStack ใหญ่



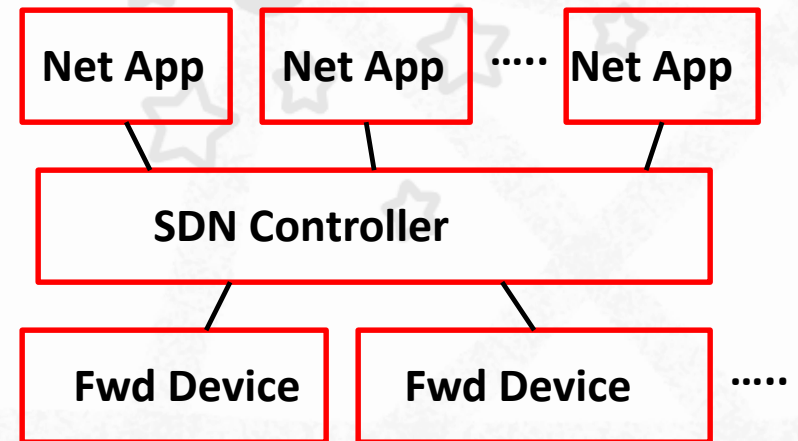
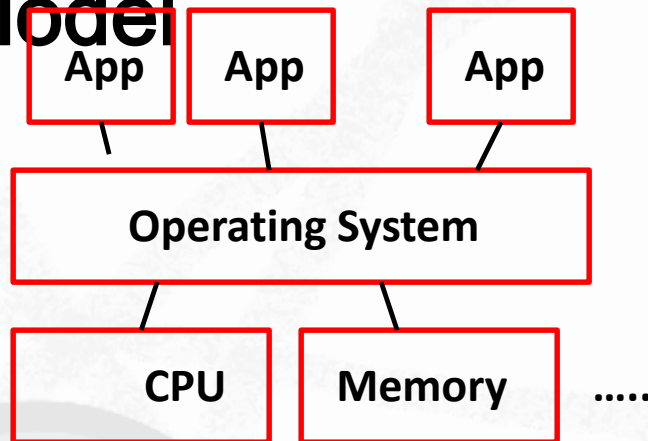
Integration กับเทคโนโลยีอื่น

ประสานการทำงานกับเทคโนโลยีอื่น

- **OpenStack และ SDN integration**
 - Neutron Plugin to OpenDayLight
 - Neutron Plugin for ONOS
- **OpenStack และ container interoperability/integration**
 - รัน Container ร่วมกับ OpenStack: Mesos สามารถใช้ OpenStack VM สำหรับรัน Container Orchestration
 - รัน Container ใน OpenStack: เช่นใน Magnum, Murano, และ Carina
 - รัน OpenStack บน Container เช่น

SDN Integration

- SDN เปรียบเทียบกับ Computer Layer Model



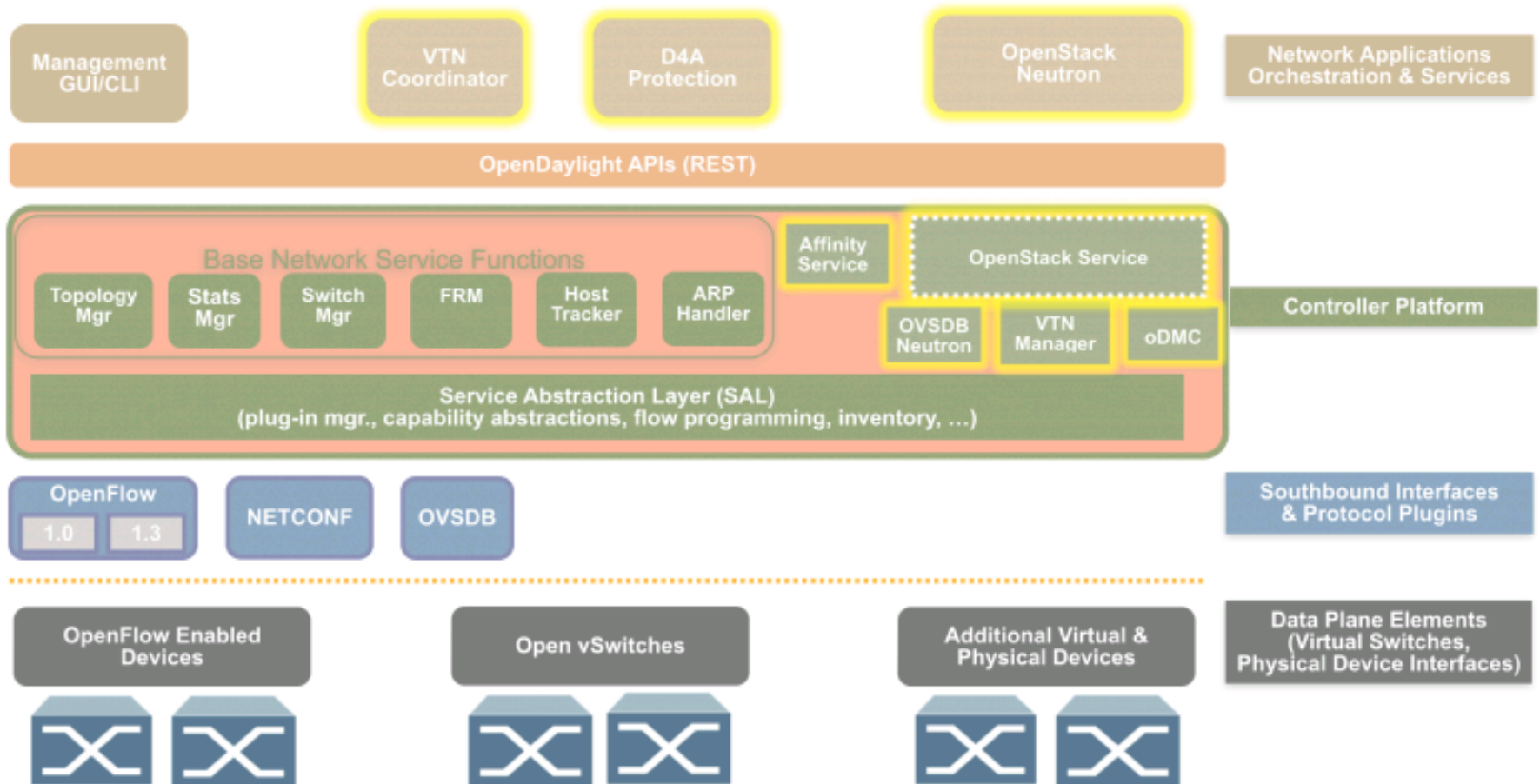
D. Kreutz et al. "Software-Defined Networking: A comprehensive Survey."

- OpenDayLight คือ SDN controller หนึ่งที่เป็นความร่วมมือกันของกว่า 36 บริษัทเพื่อสร้าง



OPEN
DAYLIGHT

Virtualization Edition

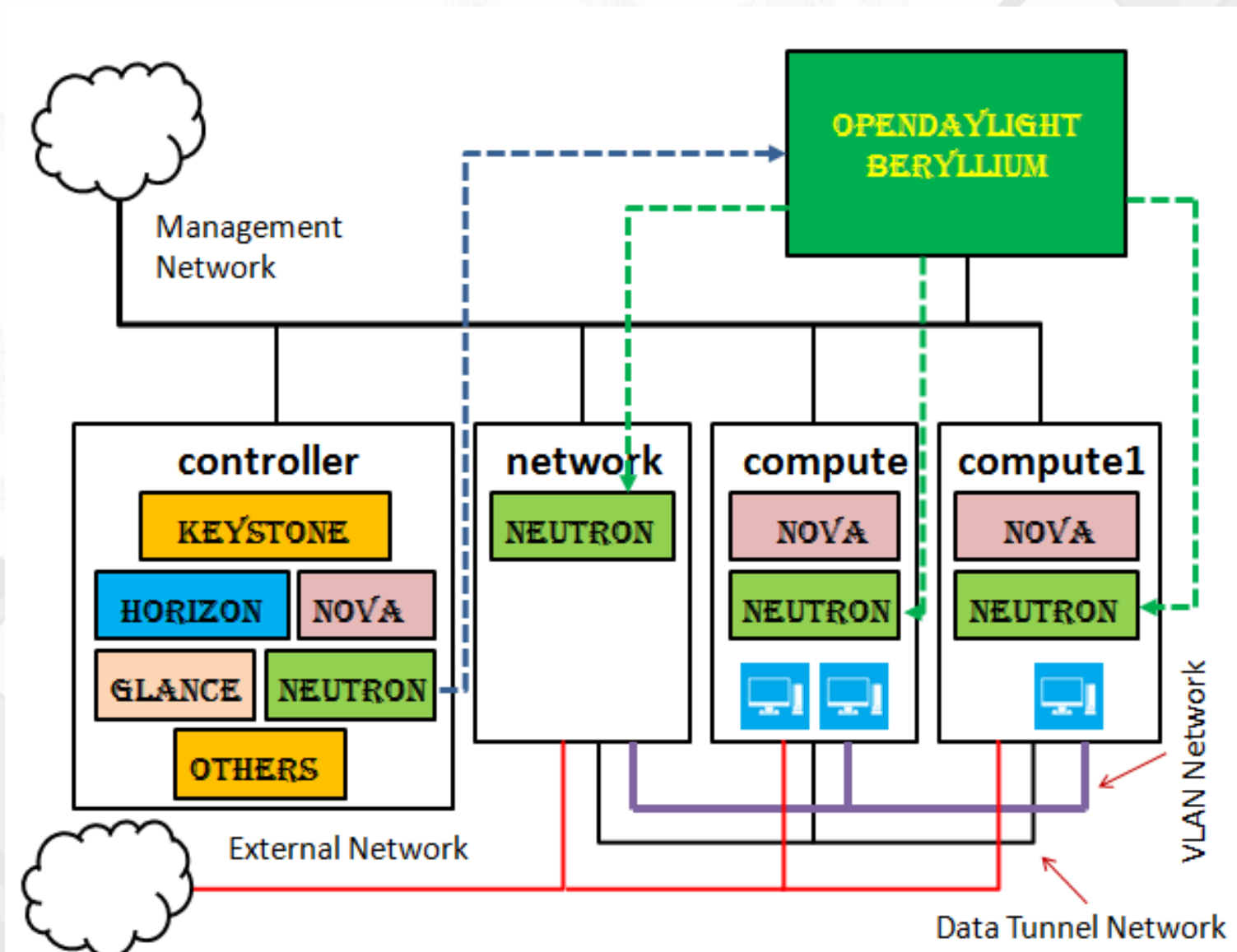


ที่มา: https://screeninet.files.wordpress.com/2014/04/odl_virt1.png

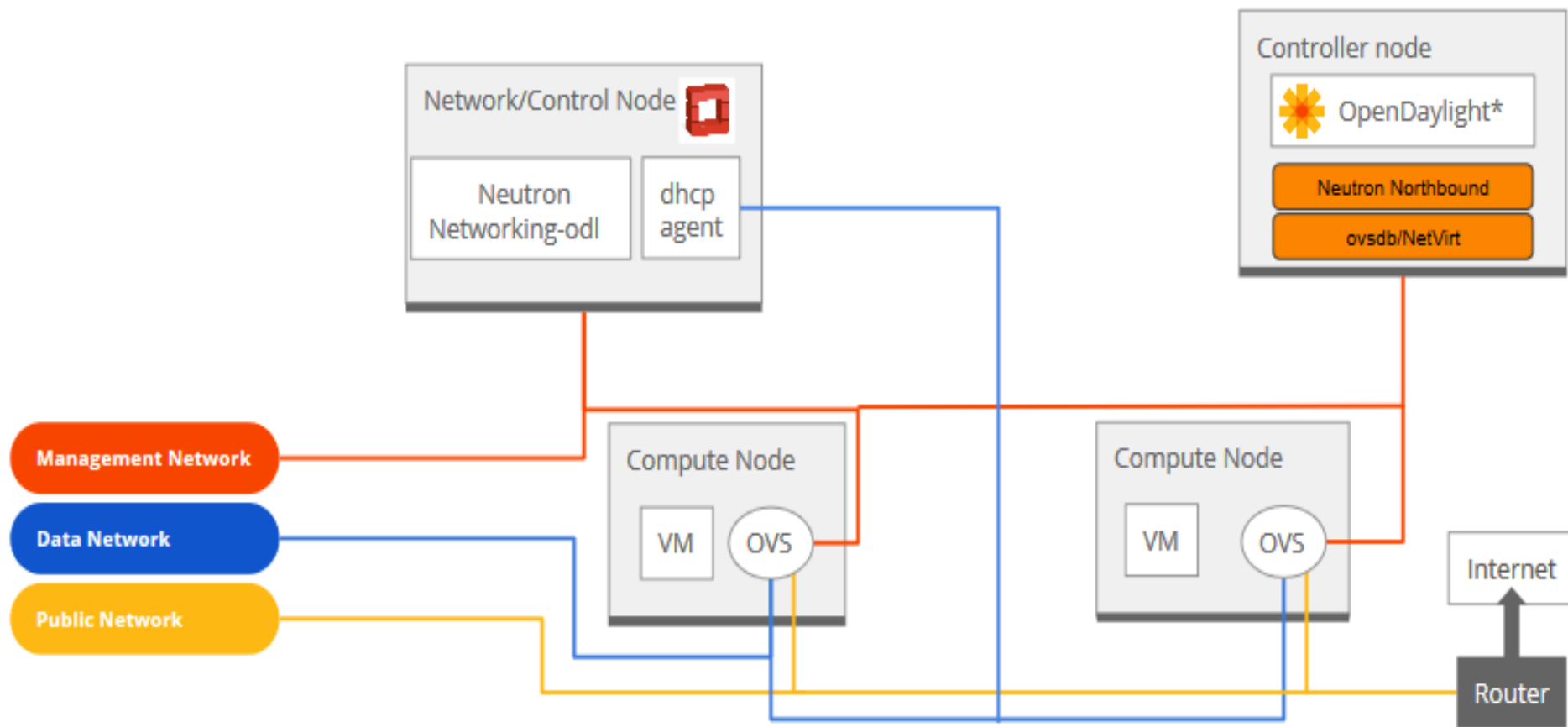
OpenDayLight Integration

- Neutron สามารถใช้ OpenDayLight เป็น Network Controller
- ทำให้ Datacenter สามารถควบคุม Virtual Switch และ OpenFlow switch (hardware) ได้โดยใช้ OpenDayLight Controller และ Network Applications ต่างๆ
 - Network Virtualization,
 - QoS,
 - Traffic Engineering,
 - Other Net Apps (e.g. Security Apps)

ODL INTEGRATION



Openstack and OpenDayLight Integration



OpenDayLight Integration

- OpenDayLight มี Features ที่สนับสนุนการทำงานของ Neutron เกือบทุกความสามารถ ยกเว้น Security Group (Firewall)
 - OpenDayLight มี Network Application ที่จัดการเรื่อง Firewall อยู่แล้วได้แก่ Virtual Tenant Network (VTN)
- OpenDayLight ทำให้ลดจำนวน Bridge ภายใน neutron ลง
- ที่ vasabilab เราเขียนวิธีการ Integrate OpenStack Mitaka แบบ Classic OVS network และ OpenDayLight Beryllium ได้

สรุป

- ในปัจจุบัน OpenStack เป็น Software ที่ Features พร้อมสำหรับใช้งานเป็น Production Public และ Private Cloud
- OpenStack คือกลุ่มคนที่ทำงานร่วมกันเพื่อประโยชน์ส่วนรวม
- เป็น Software ที่ปรับตัวทำงานร่วมกับเทคโนโลยีอื่นๆได้ แต่ในขณะเดียวกันก็รักษาแนวทางของตนเอง
- ความเข้าใจใน Installation Guide ของ OpenStack ยังเป็นพื้นฐานที่สำคัญสำหรับการ Admin หรือพัฒนาต่อในอนาคต