Ryu: Network Operating System

Kazutaka Morita
NTT Software Innovation Center

Isaku Yamahata
VA Linux

June 6, 2012

Agenda

- Overview
- How it works
- Demo
- Summary

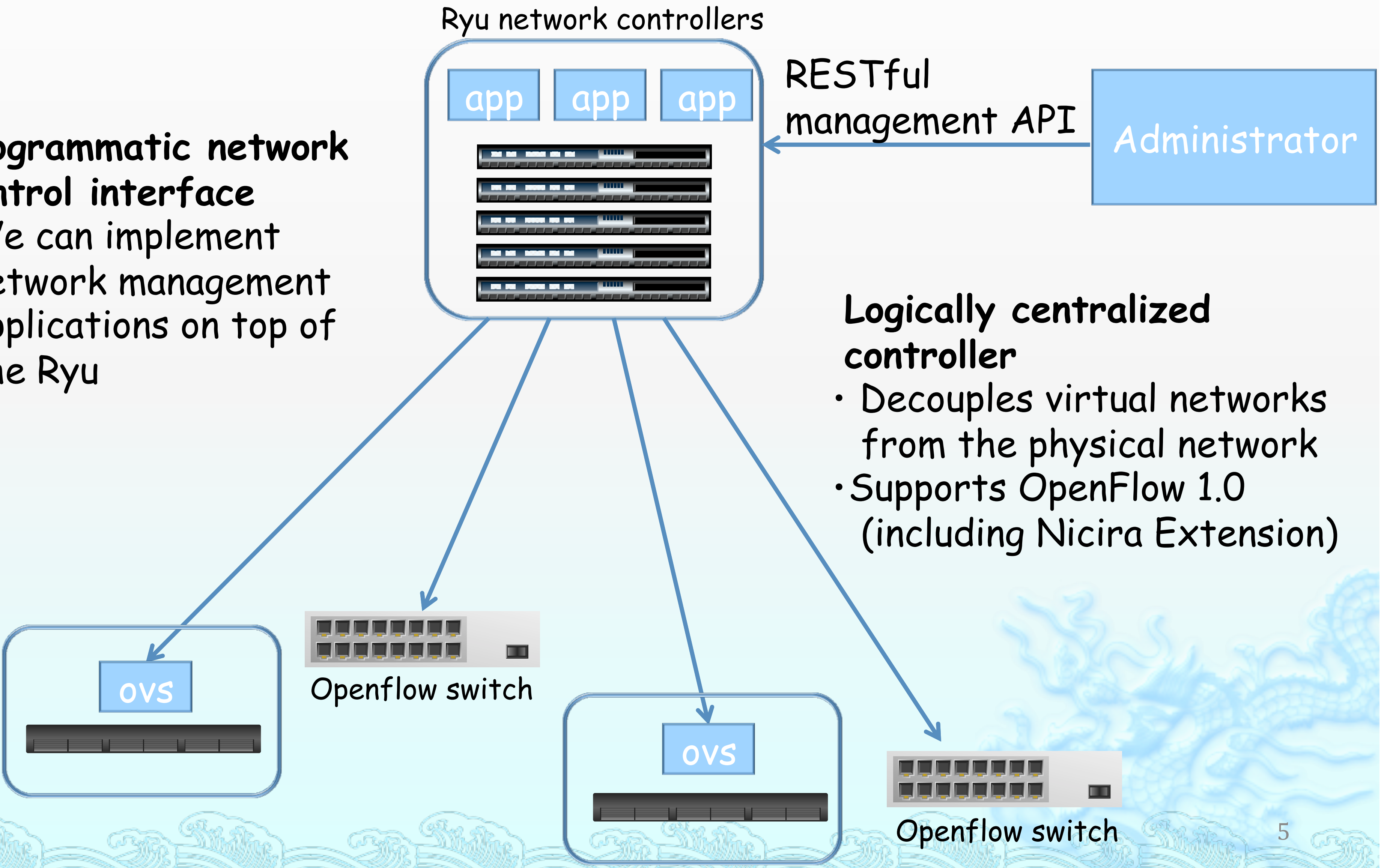
Overview

What is Ryu?

- Open-sourced network operating system
  - Network operating system
    - Programmatic network control interface
    - Logically centralized controller for thousands of switches (OVS, openflow switch)
  - Open source software (Apache v2)
    - Fully written in Python
    - Project site: http://www.osrg.net/ryu/

- Ryu stands for
  - 流　- Means "flow" in Japanese
  - 龍　- Means "Japanese dragon", one of water gods

Overview



Programmatic network control interface

- We can implement network management applications on top of the Ryu

Logically centralized controller

- Decouples virtual networks from the physical network
- Supports OpenFlow 1.0 (including Nicira Extension)

Goals

- De facto OSS network operating system
- High quality enough for use in large production environment
  - Code quality
  - Functionality
  - Usability
- Become the standard network controller of cloud software (e.g. OpenStack)
  - Ryu plugin is merged into OpenStack Essex
- Default Controller for fedora/debian/ubuntu

What does Ryu provide?

- Ryu applications
  - GRE tunneling
  - VLAN support
  - Topology discovery
  - MAC based segregation
- We can use these features with only commodity hardware

How it works

Integrate with OpenStack

- OpenStack
  - Open source software for building private and public clouds
- What does Ryu bring to OpenStack?
  - Flat L2 networks regardless of the underlying physical network
  - Scalable multi-tenant isolations
    - Ryu provides tunneling based isolations
    - VLAN doesn't scale larger than 4096
    - We don't need high-end switches

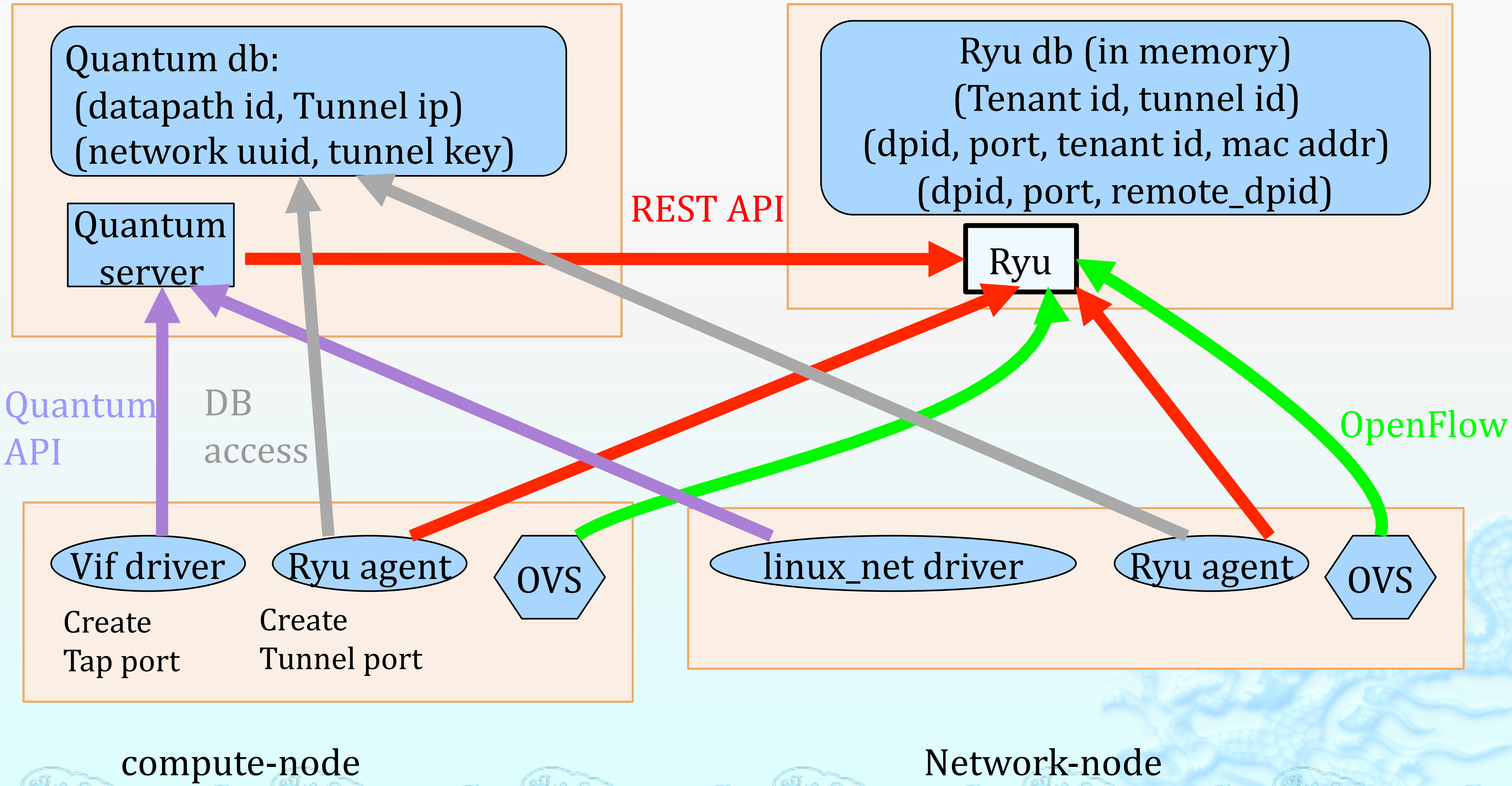
OpenStack

- Nova: cloud management system
  - Nova compute node
    - Physical machine that runs guest VM instances
  - Nova network node
    - Physical machine that runs networks gateway to the outside network
- Quantum: network management system
  - Quantum server
    - Manages network configuration
    - Nova requests quantum-server for network configuration
  - Quantum agent
    - It runs on nova compute/network node
  - Quantum plugin
    - Plugin for each network technology
    - Ryu plugin

How Ryu works with OpenStack

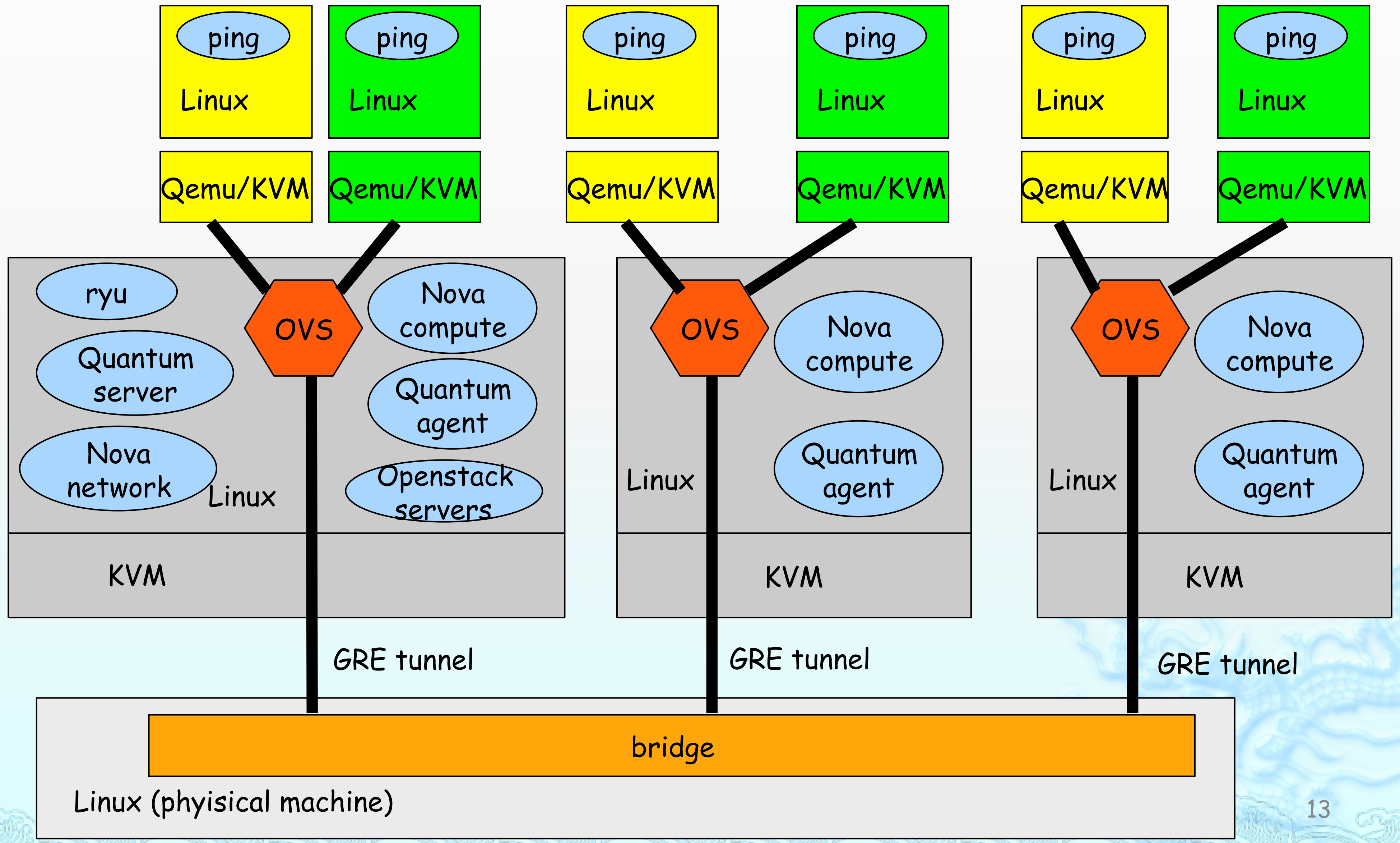
Quantum-node: somewhere where compute/network can communicate. Typically on network-node

Ryu-node: somewhere where compute/network/quantum can communicate Typically on network-node

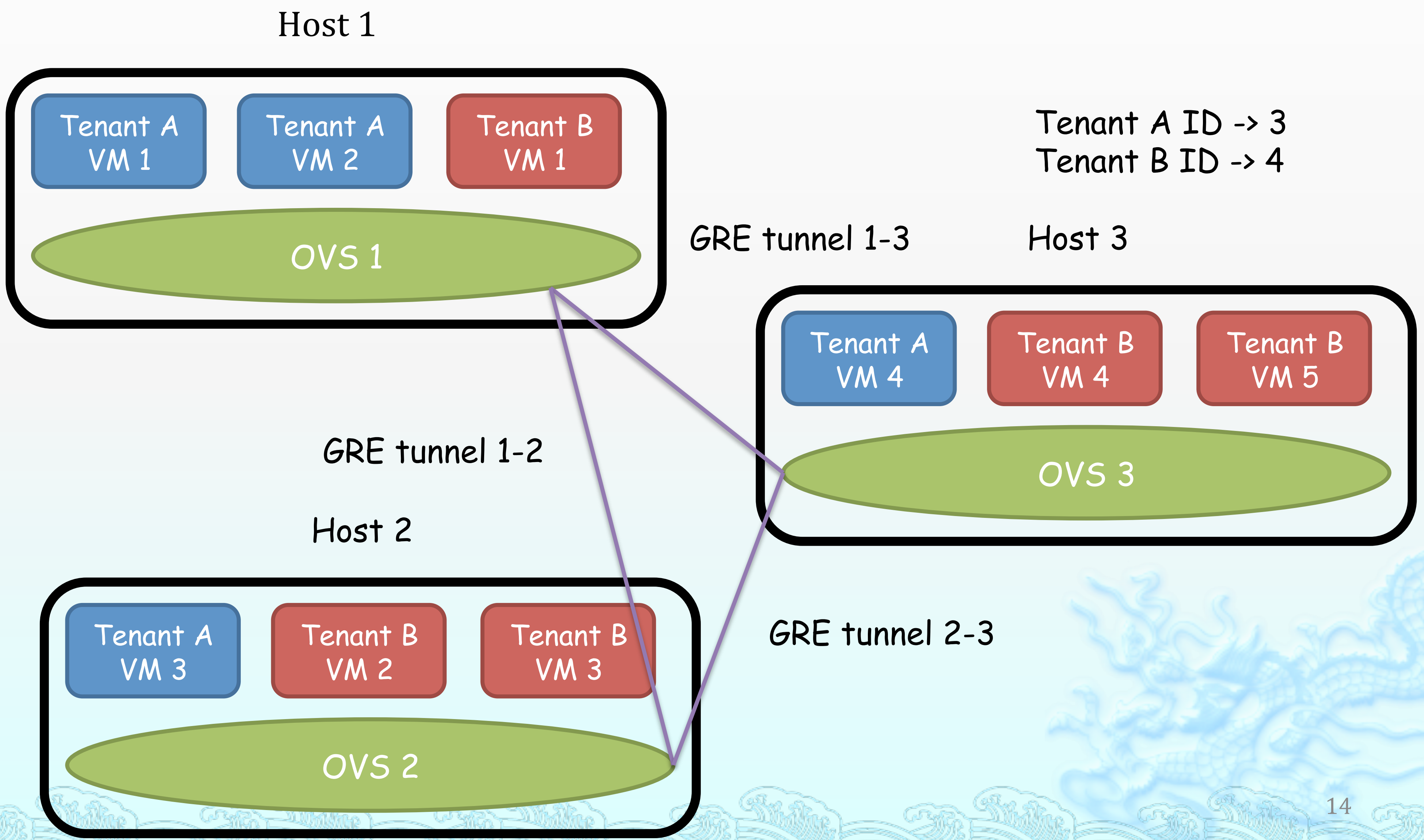


Demo

Ryu demo (GRE tunneling)



Ryu demo (GRE tunneling)



Summary

Future items

- Integration with Quantum IPAM and L3 API
- Firewall
- Virtual network to physical network, and vice versa
  - Convert among GRE key, VLAN tag, …
- Distributed controllers
  - No single point of failure
  - Datacenter-wide scalability

Summary

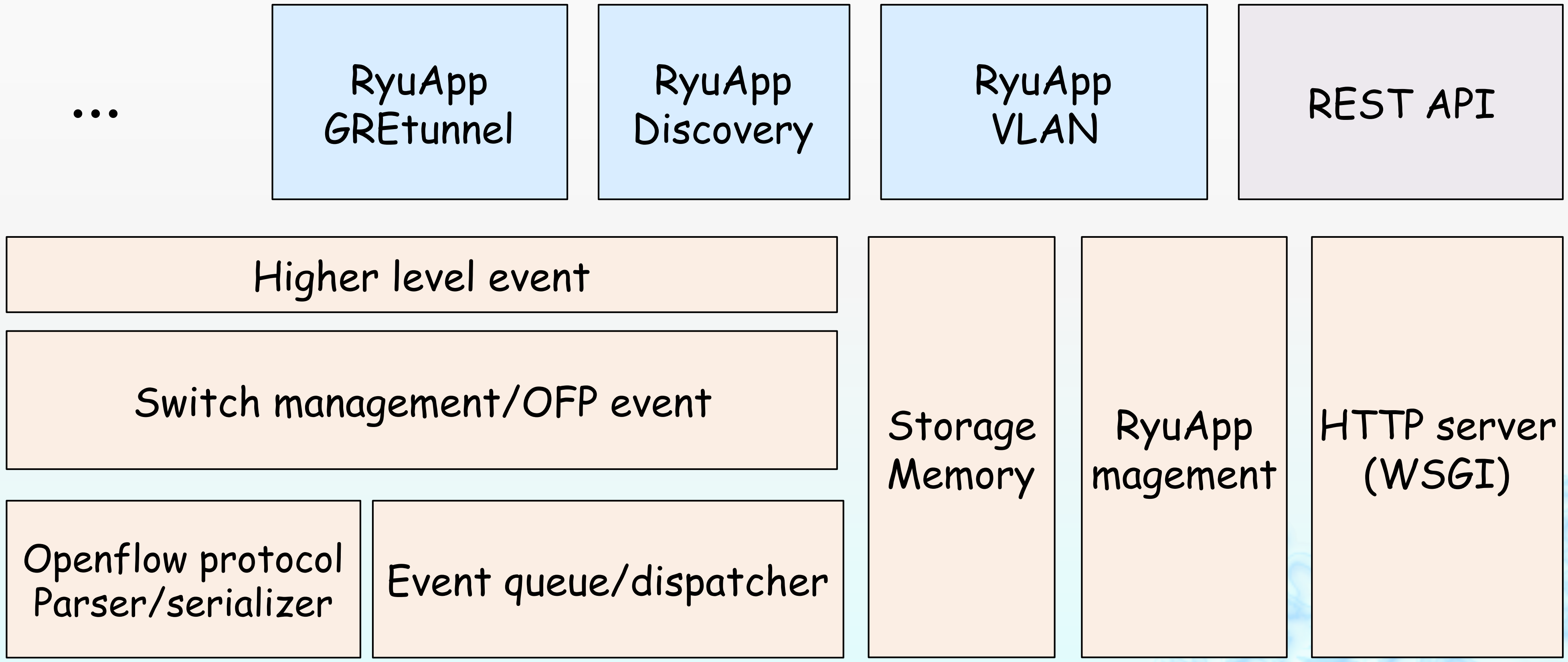
- Ryu is an open-sourced network operating system licensed under Apache License v2.
  - Site: http://www.osrg.net/ryu/
  - ML: ryu-devel@lists.sourceforge.net

- Set up Ryu environment with VM images
  - https://github.com/osrg/ryu/wiki/RYU-OpenStack-environment-VM-image-file-HOWTO

Thank you! Any questions?

Appendix

Block diagram of Ryu



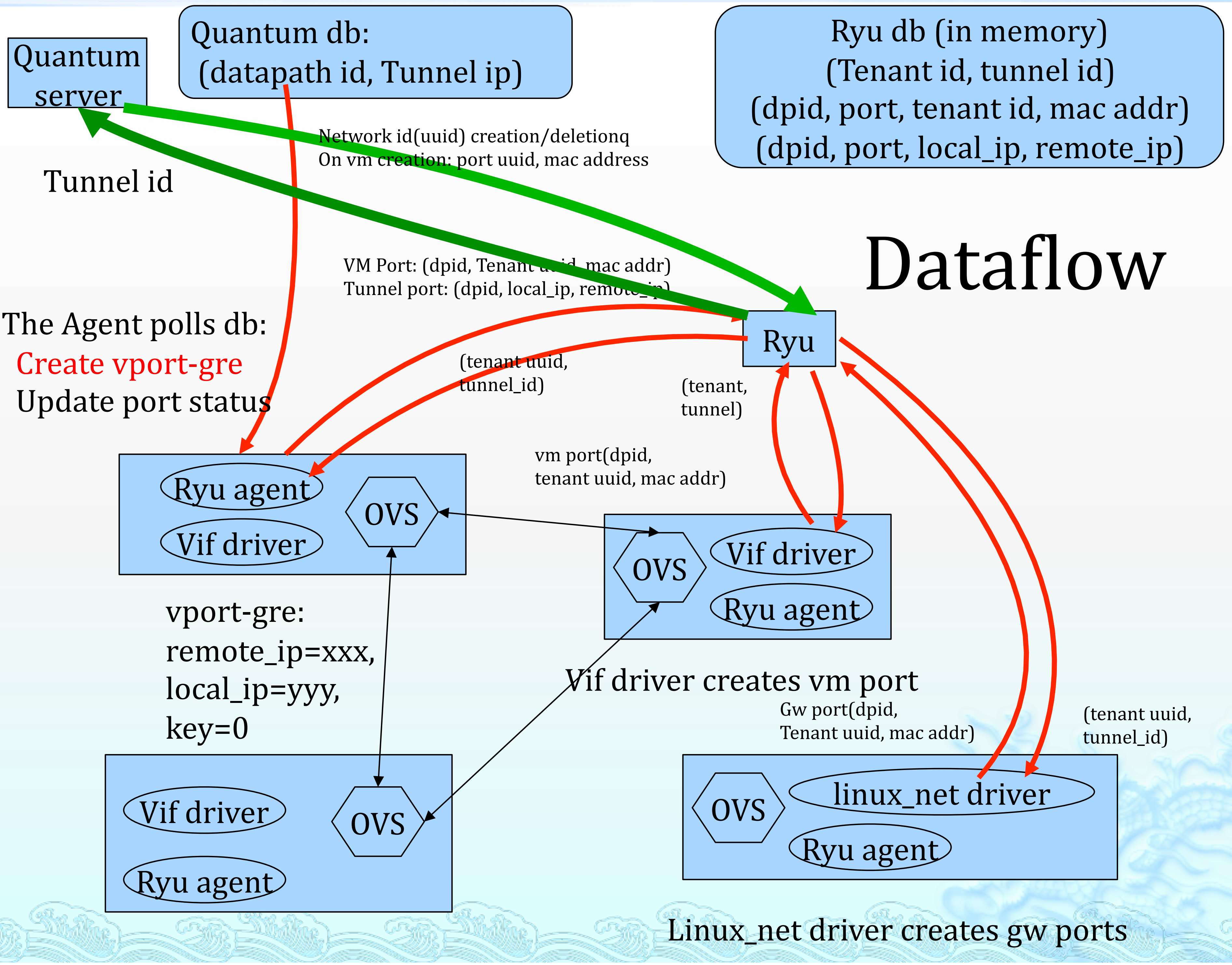
OpenStack basics

- OpenStack
  - Nova: cloud management system
  - Quantum: network management system
- Nova compute node
  - Physical machine that runs guest VM instances
- Nova network node
  - Physical machine that runs networks gateway to the outside network
- Quantum server
  - Manages network configuration
  - Nova requests quantum-server for network configuration
- Quantum agent
  - It runs on nova compute/network node
- Quantum plugin
  - Plugin for each network technology
  - Ryu plugin

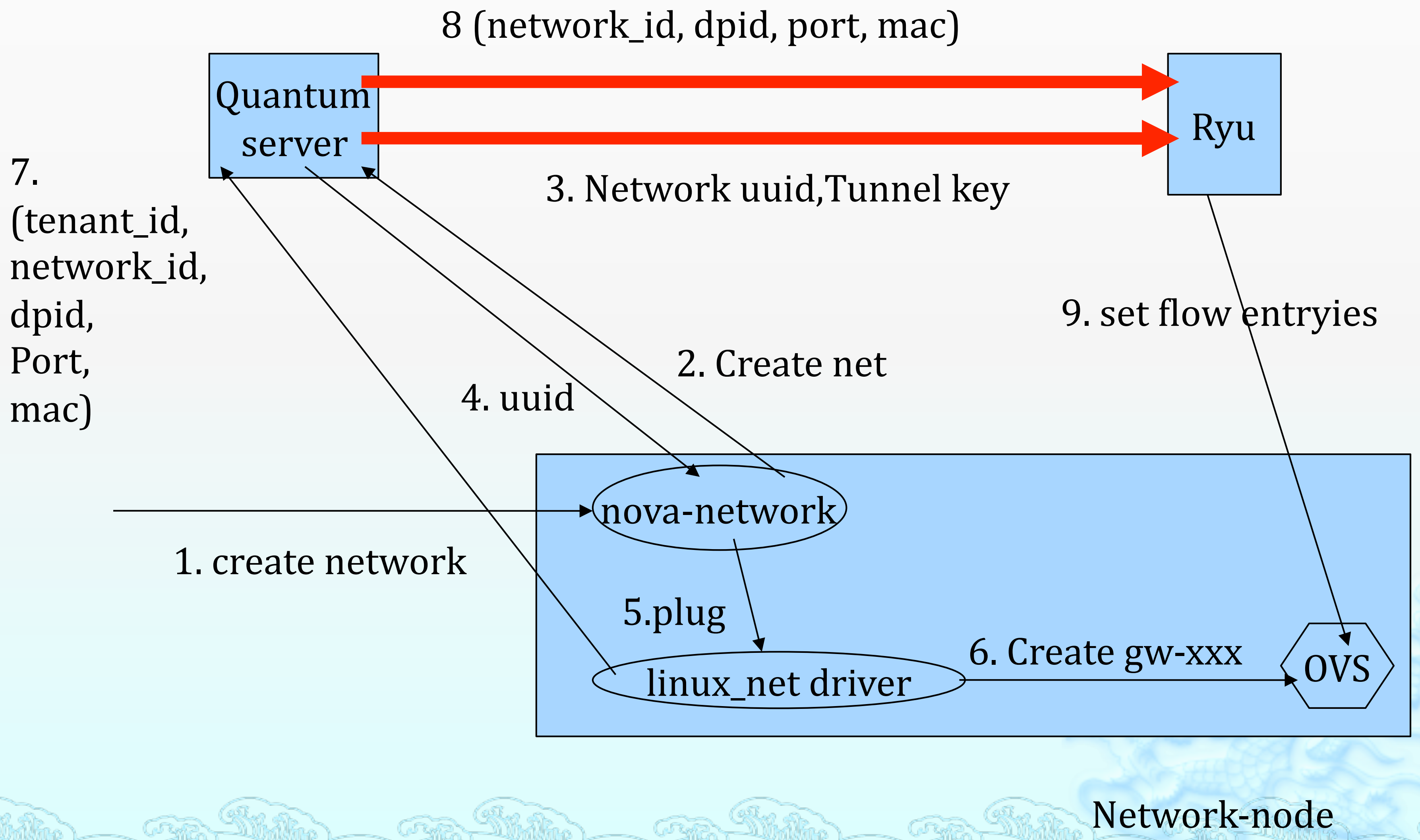
GRE tunneling with openstack

- Network Tenant creation
  - GRE key assignment
  - Gateway creation
- Guest VM instance creation
  - Port creation
    - tenant ↔ key ↔ port relationship
  - Setting flow to the VM port
- Tunnel port management
  - Tunnel port creation/deletion
    - Track physical compute node
  - Setting flow to the tunnel port

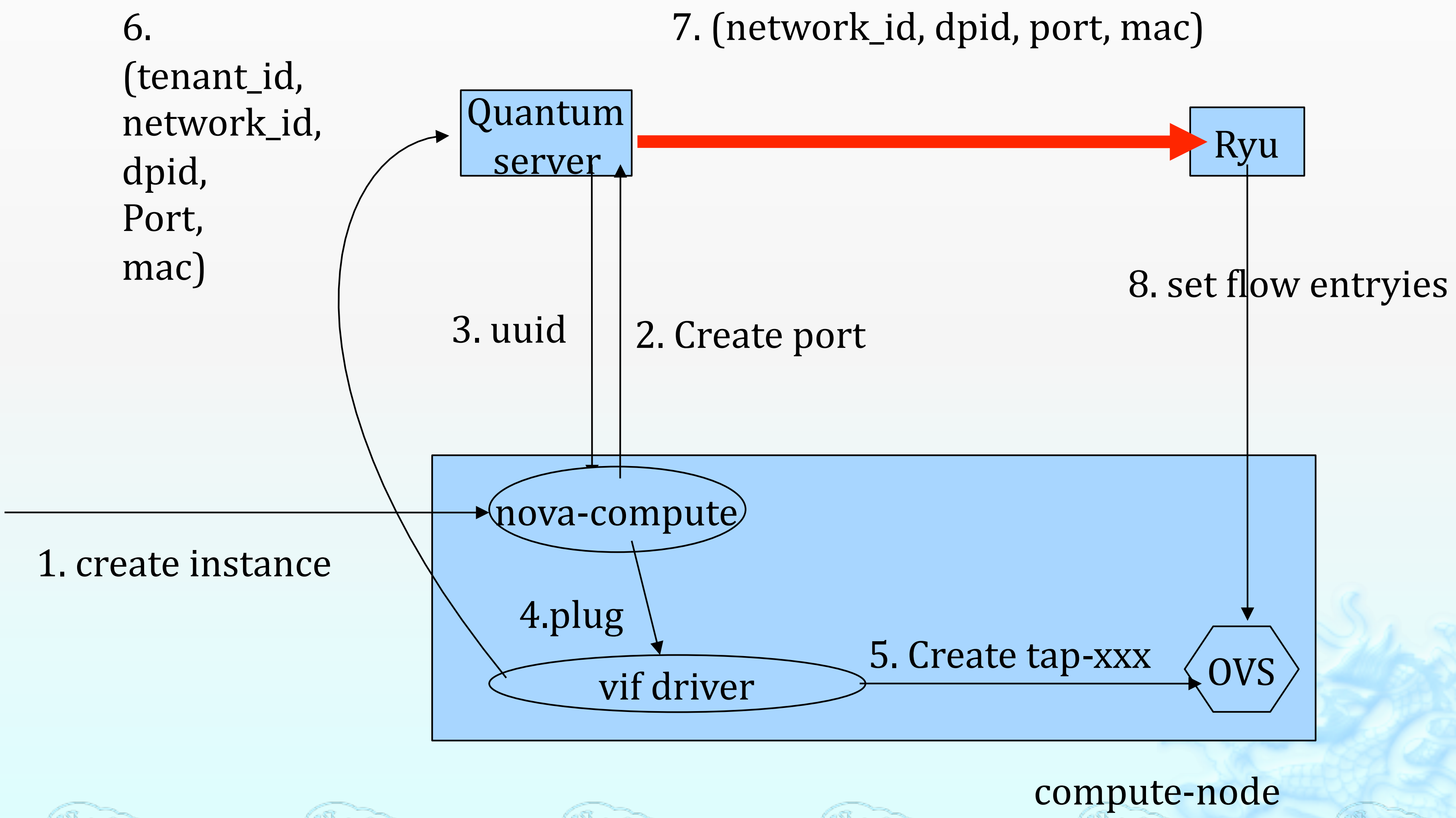
Dataflow


Quantum server
Quantum db:
(datapath id, Tunnel ip)
Ryu db (in memory)
(Tenant id, tunnel id)
(dpid, port, tenant id, mac addr)
(dpid, port, local_ip, remote_ip)
Network id(uuid) creation/deletionq
On vm creation: port uuid, mac address
Tunnel id
VM Port: (dpid, Tenant uuid, mac addr)
Tunnel port: (dpid, local_ip, remote_ip)
Ryu
The Agent polls db:
Create vport-gre
Update port status
(tenant uuid,
tunnel_id)
(tenant,
tunnel)
vm port(dpid,
tenant uuid, mac addr)
Ryu agent
Vif driver
OVS
vport-gre:
remote_ip=xxx,
local_ip=yyy,
key=0
Vif driver creates vm port
Gw port(dpid,
Tenant uuid, mac addr)
(tenant uuid,
tunnel_id)
linux_net driver
Linux_net driver creates gw ports

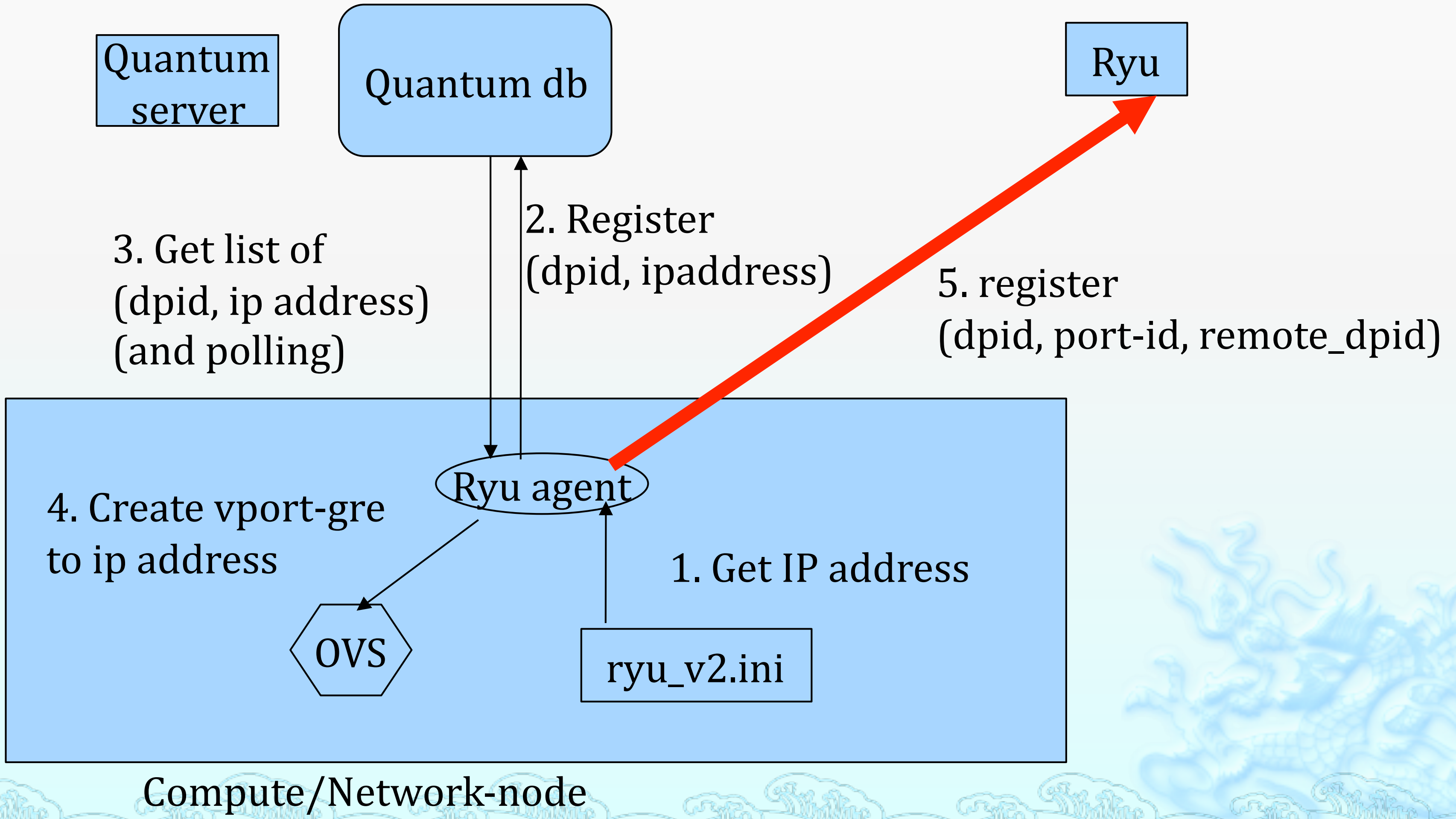
Network Creation



Instance Creation

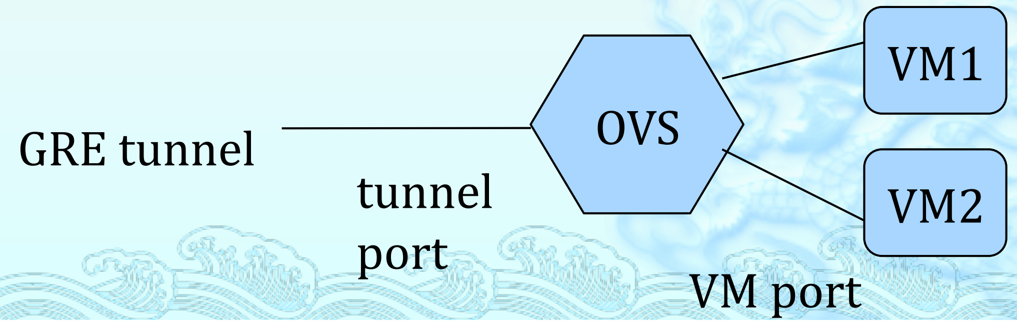


Node boot up


Quantum server
Quantum db
Ryu
2. Register (dpid, ipaddress)
3. Get list of (dpid, ip address) (and polling)
5. register (dpid, port-id, remote_dpid)
Ryu agent
4. Create vport-gre to ip address
1. Get IP address
OVS
ryu_v2.ini
Compute/Network-node

Flow Table Usage

	Src table		Tunnel out		Local out	
	Table 0		Table 1		Table 2	
In port	match	action	match	action	match	action
VM port	in_port src mac	set_tunnel goto table 1	tunnel_id dst mac	output(tunnel) goto table 2		
	in_port	drop	tunnel_id	goto table 2	tunnel_id dst mac	output(vm)
Tunnel port	in_port tunnel_id	goto table 2			tunnel_id	drop
	in_port	drop				


GRE tunnel
tunnel port
OVS
VM1
VM2
VM port