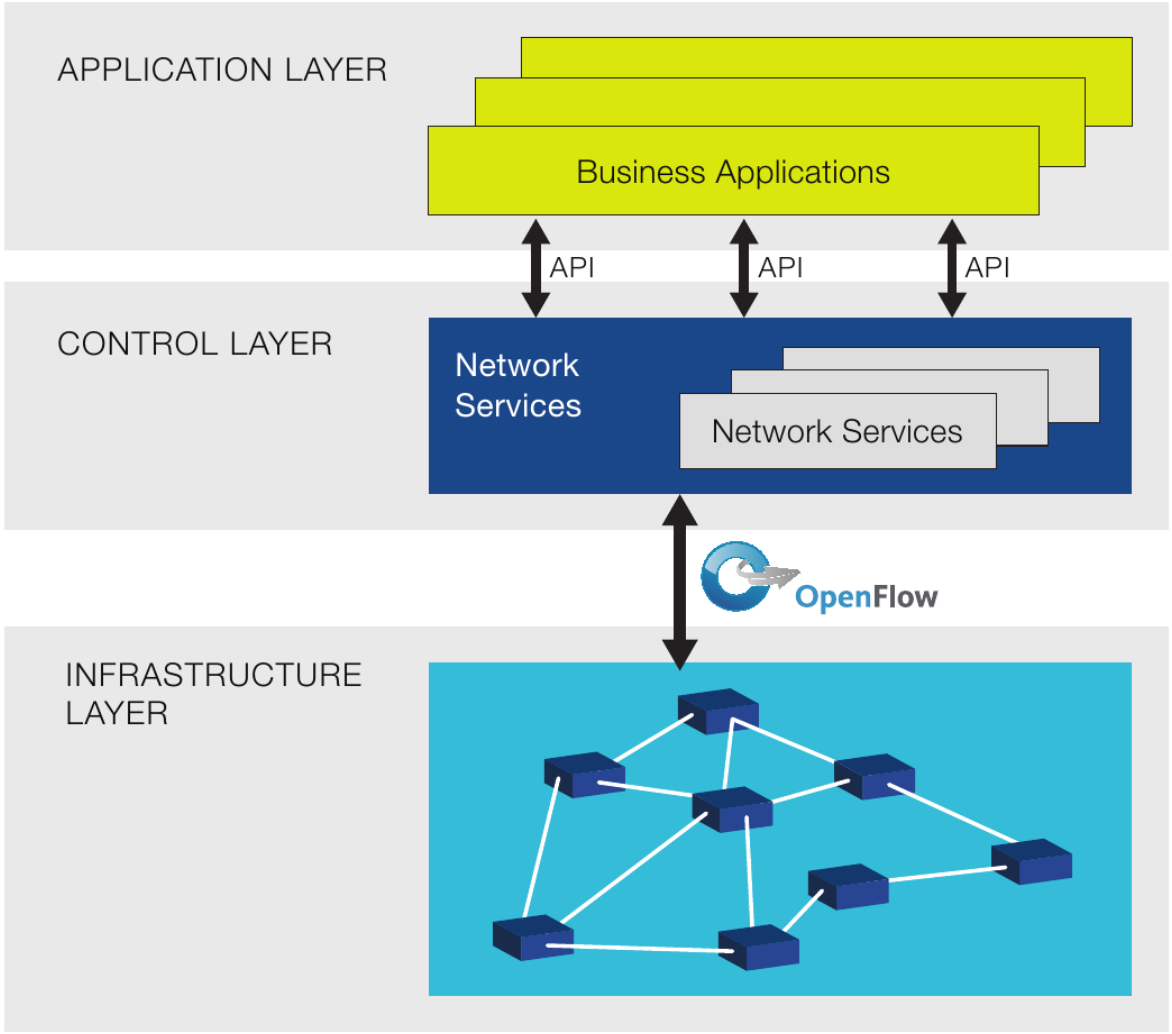
SDN in the Cloud

Thomas Michael Bohnert
Philipp Aeschlimann
{bohe,aepp}@zhaw.ch
ICCLab 2013

Agenda

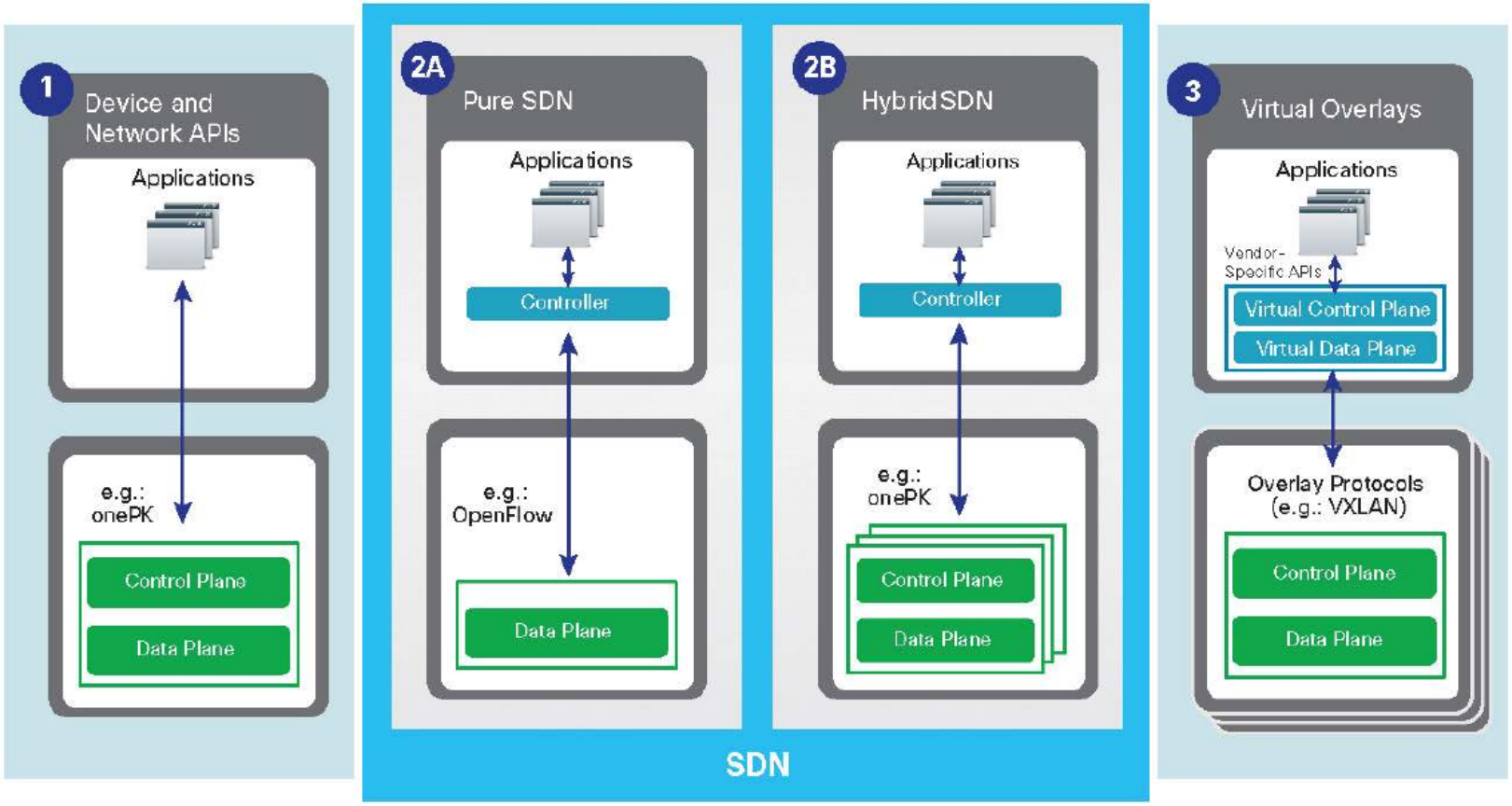
- SDN - What, Why, and How
- Cloud Frameworks, and SDN in Cloud Frameworks
- Available Control Plane’s
- Implementation in OpenStack

SDN - a paradigm



What is not SDN?

- CISCO



From the paradigm to implementation - SOUTHBOUND

- Every protocol that can connect to a network device
- SNMP
  - Can be used to:
  - Get hardware / software status
  - Configure hardware / software
- OVSDB
  - Configuration for the tables in Open vSwitch
- OpenFlow
  - the forwarding
  - the topology
  - the status of a device
  - simple QoS

From the paradigm to implementation - NORTHBOUND

- REST API
  - At the moment no specification for it
  - The specification is made by the available implementation - If at all
- Protocols
  - HTTP
  - JSON as data format
- Authentication and Authorisation
  - HTTP basic authentication mechanism
  - Can also use a backend (e.g. LDAP)
  - Use of certificates

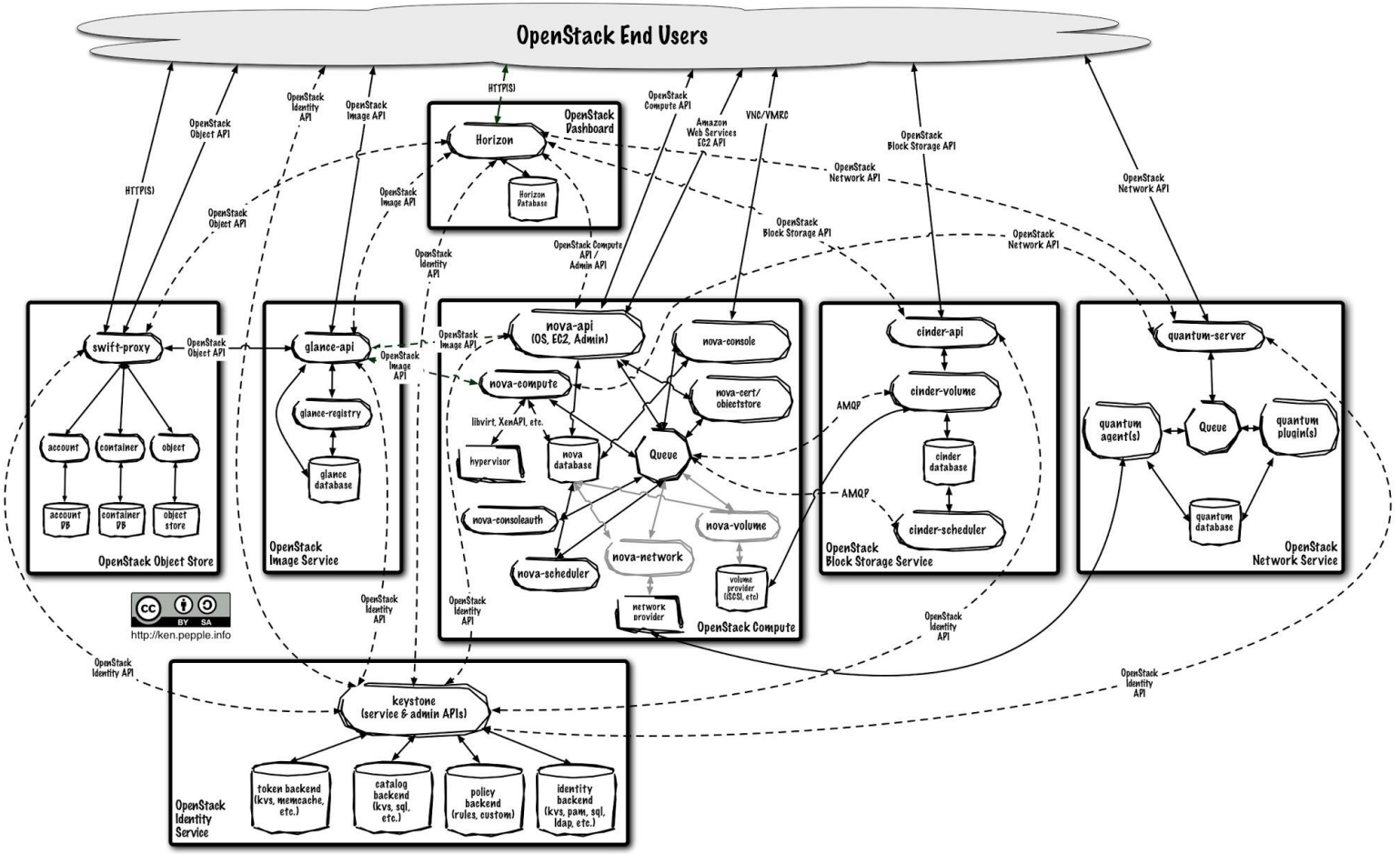
Clouds: A brief overview

- Available implementations of a “Cloud”
- Windows Azure
  - Provides IaaS and PaaS, released 2010
- Amazon Web Services AWS
  - Primarily IaaS (EC2, S3) but many more
- OpenStack
  - Provides IaaS, #1 OSS player
- CloudStack
  - Amazon API as well as self developed API
- Eucalyptus
  - Fully compatible with AWS
  - Good number of deployments
- OpenNebula
  - Research and Educational Institutions

Networking in Clouds

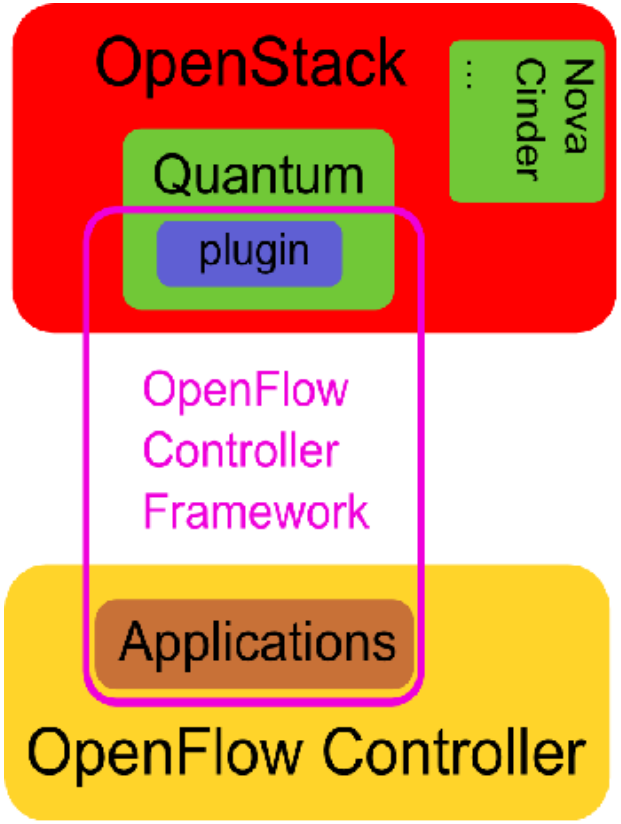
- Available implementations of a “Cloud”
- Amazon Web Services AWS
  - Virtual Private Cloud, mostly L3 control, VPN external
- OpenStack
  - From VLAN to SDN
- CloudStack
  - From VLAN to SDN
- Eucalyptus
  - From VLAN to SDN
- OpenNebula
  - From VLAN to SDN

OpenStack - Architecture



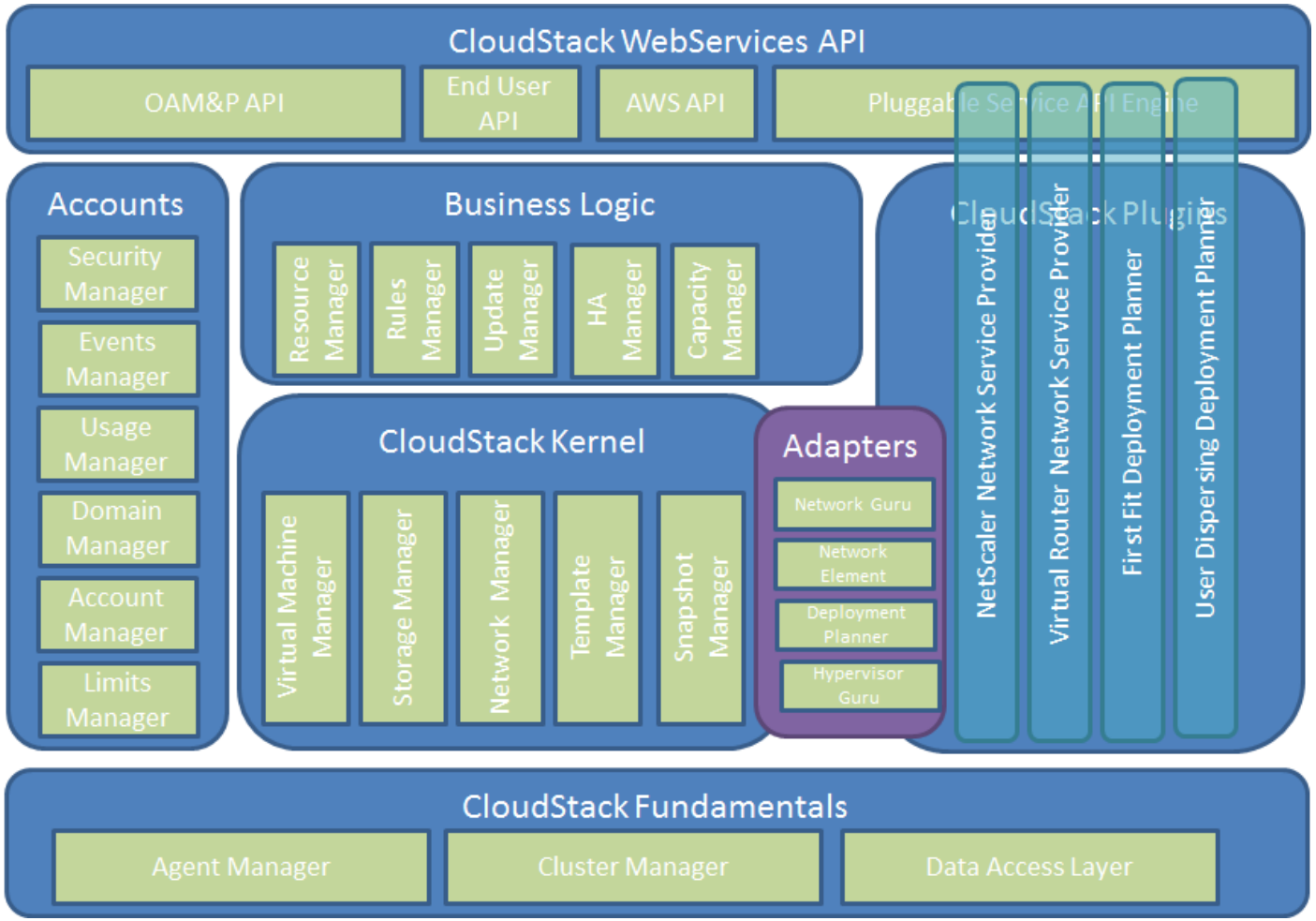
Mind: Starting with the Havana release, the OpenStack Networking project's code name is Neutron. Quantum is no longer used.

SDN in the Cloud - OpenStack

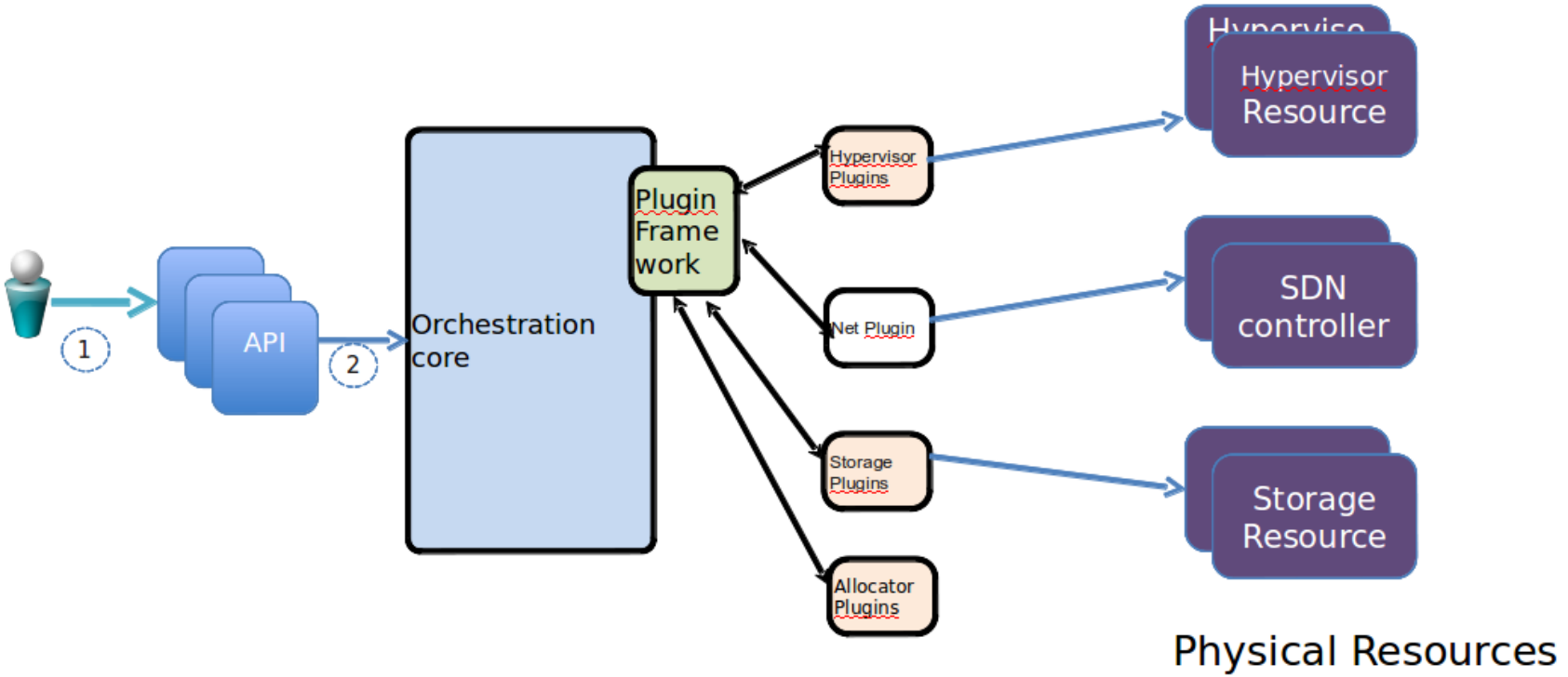


Mind: Starting with the Havana release, the OpenStack Networking project's code name is Neutron. Quantum is no longer used.

CloudStack - Architecture

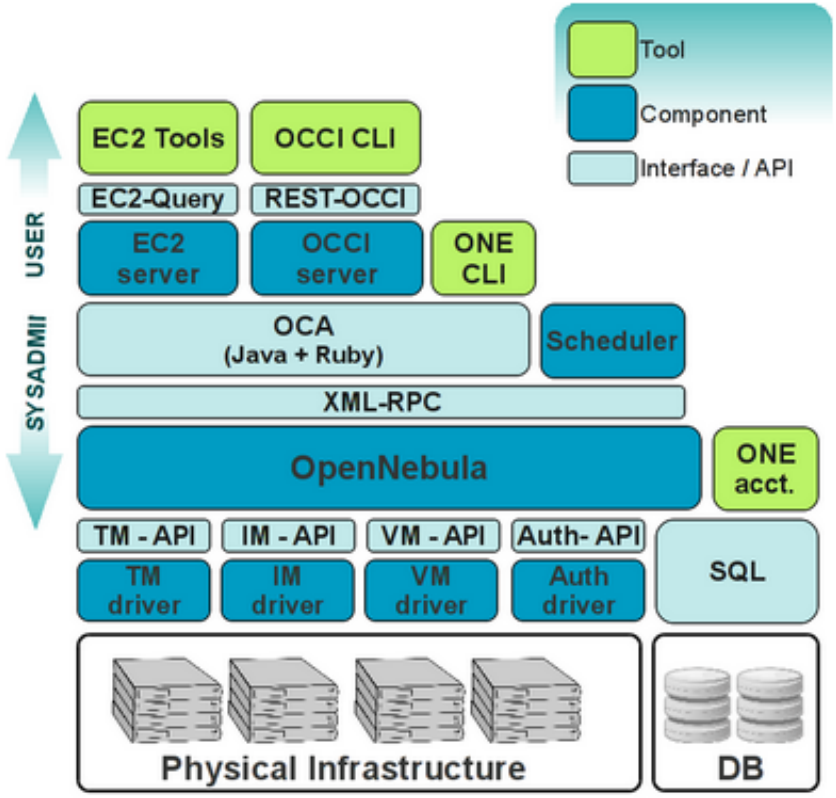


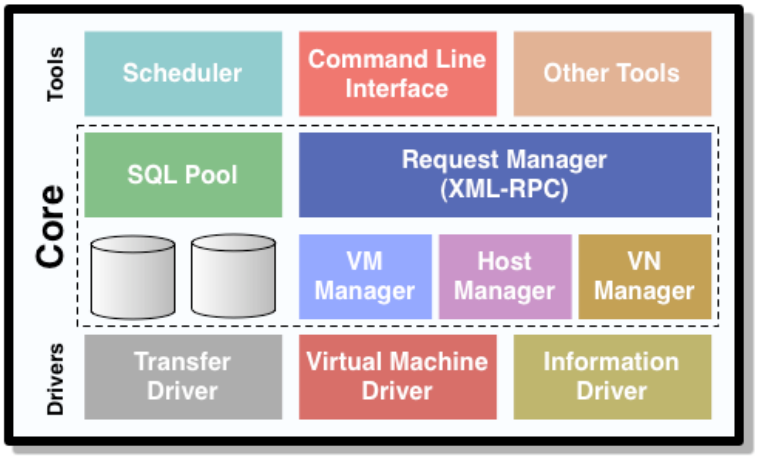
SDN in the Cloud - CloudStack



Source: Chiradeep Vittal, SDN in CloudStack

OpenNebula - Architecture





Virtual Network Manager

The Virtual Network Manager (VNM) is responsible for the handling of IP and MAC addresses, allowing the creation of virtual networks by keeping track of leases (a set form by one IP and one MAC valid on a particular network) and their association with virtual machines and the physical bridges the VM are using.

Cloudified Networking Services

	Nova	Quantum
*-as-a-service	Compute	Network
Major API abstractions	"virtual servers": represents a host with CPU, memory, disk, and NICs.	"virtual networks": A basic L2 network segment. "virtual ports": Attachment point for devices connecting to virtual networks.
Interactions with other OpenStack services.	virtual servers use "virtual images" from Glance.	virtual ports are linked to vNICs on "virtual servers".
Supports different back-end technologies	"virt-drivers" for KVM, XenServer, Hyper-V, VMWare ESX	"plugins" for Open vSwitch Cisco UCS, Linux Bridge, Nicira NVP, Ryu Controller.
API Extensibility for new or back-end specific features.	keypairs, instance rescue, volumes, etc.	quality-of-service, port statistics, security groups, etc.

Mind: Starting with the Havana release, the OpenStack Networking project's code name is Neutron. Quantum is no longer used.
Source: Dan Wendlandt – Quantum Hacker & PTL

Why SDN in the Cloud

- Overcome current problems
  - Restriction to 4096 VLAN ID’s
  - Dynamic creation of Network segments
  - Elastic implementation of the network
- The centralized approach of SDN
  - Avoid “box” configurations
  - Flexible monitoring in virtual and physical environment
  - Centralized management of the needs from the tenant
  - Testable Network for millions of tenants made easy
- Use Vendor independent hardware
  - Use of commodity hardware
  - Open Source Software available

Available control plane

What controllers are available

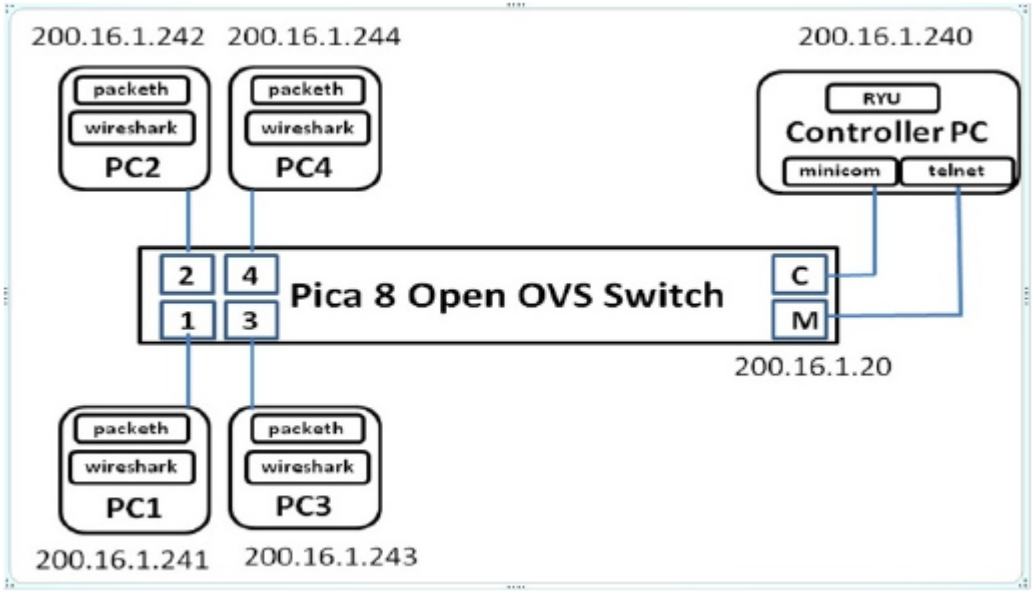
- Different controllers for different requirements
- OpenDaylight
  - A controller that supports not only OpenFlow
  - Not yet released
- NOX/POX
  - Reference Implementation from Stanford University
- RYU
  - The best choice for OpenStack
  - Implemented in python
- Trema
  - Implemented in ruby
  - Advanced development API
- Floodlight
  - Implemented in Java

RYU

- Ryu is an Operating System for Software Defined Network.
- Applications and server are written in python, as also lot of other parts in OpenStack.
- Ryu fully supports
  - OpenFlow v1.0 with Nicira Extensions
  - OpenFlow v1.2 and v1.3.
- All of the code is freely available under the Apache 2.0 license
- Ryu is developed openly
- NTT laboratories OSRG group started Ryu project.

RYU supported Hardware

- Reference controller for all Pica8 switches
- Compatible to OpenFlow Versions 1.0 1.2 and 1.3



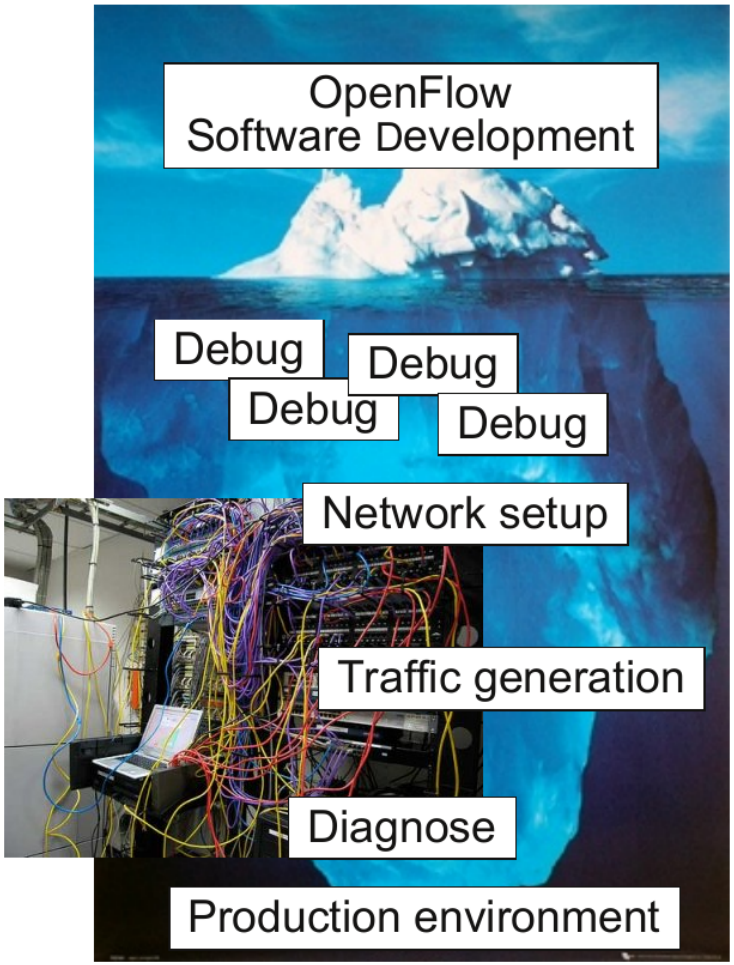
Trema

- “Trema is an OpenFlow controller framework that includes everything needed to create OpenFlow controllers in Ruby and C”*
- “Trema is not a simple OpenFlow controller, but targeting an all-in-one framework for OpenFlow development”*.
- Trema covers integrated network emulator, test framework, and debuggers
- Researchers can develop their own controllers not only for programming but also testing and debugging.”
- http://trema.github.com/trema/

* Source: Thomas Dietz

Trema

OpenFlow Iceberg



Scope of other OpenFlow controllers

Scope of Trema

Trema is an OpenFlow Platform for Entire Development Process (like Ruby on Rails)

- ***Shorter development cycle***
- ***Reduce labor cost***
- ***More and more research outputs :-)***

Trema

License

- Trema is released under the GNU General Public License version 2.0:
- http://www.gnu.org/licenses/gpl-2.0.html

It is Tested

- Automatic and periodical testing for all supported OSes
- Build test, unit test, acceptance test, test code coverage measurement

It is Supported

- Continuous
- Professional programmers at NEC support the community

Trema

- Trema supports GNU/Linux only.

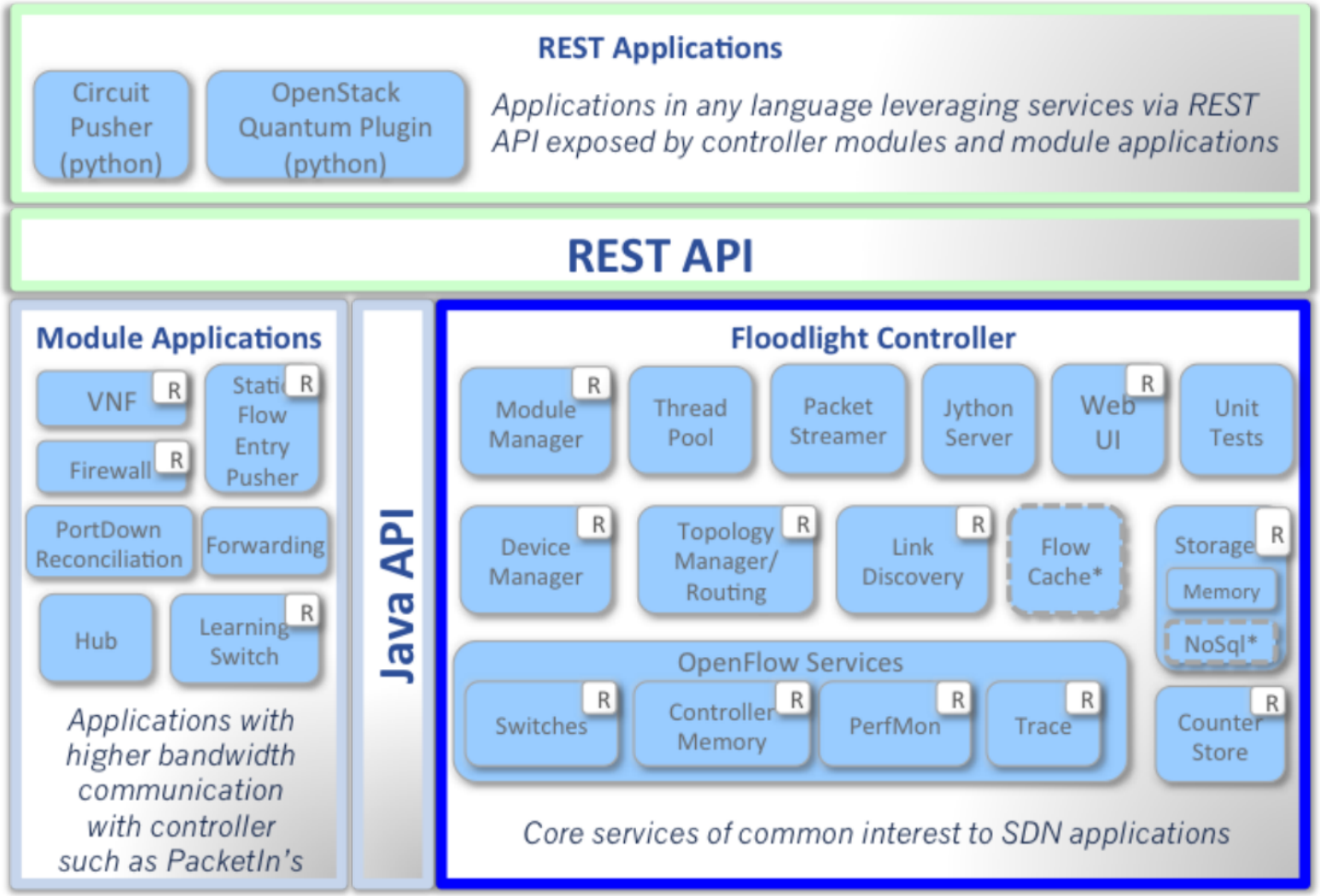
- It has been tested on the following environments:
  - Ruby 1.8.7 (1.9.x is NOT supported yet)
  - Ubuntu 12.10, 12.04, 11.10, and 10.04 (i386/amd64, Desktop Edition)
  - Debian GNU/Linux 6.0 (i386/amd64)
  - Fedora 16 (i386/x86_64)

- Trema currently supports OpenFlow version 1.0 only. (trema-edge - unstable release)

Floodlight

- Floodlight is the core of a commercial controller product from Big Switch Networks
- Is actively tested and improved by a community of professional developers
- Floodlight is an OpenFlow controller (the "Floodlight Controller") AND a collection of applications built on top the Floodlight Controller.”

Floodlight



* Interfaces defined only & not implemented: FlowCache, NoSql

Floodlight

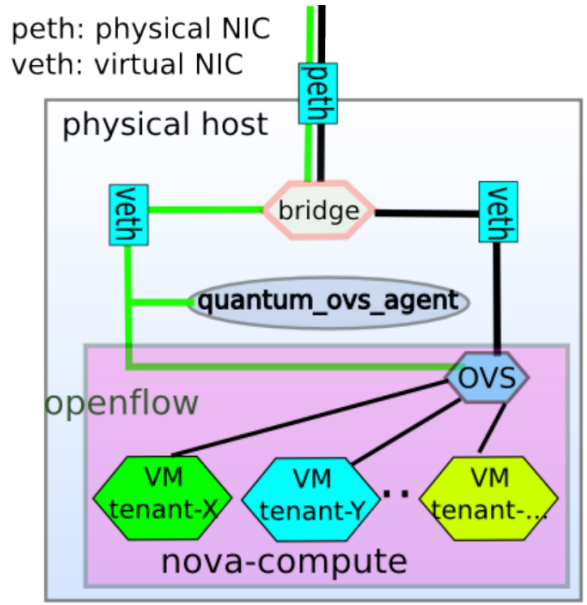
- OpenFlow Support
  - Currently supports the OpenFlow 1.0 specification.
  - Support for OpenFlow 1.2/1.3 was expected in March 2013 but it seems delayed.

- Programming Language
  - Java-based
  - Supports adding Java modules
  - Other languages can be used for application that are “above” Floodlight (using its APIs)

Implementation in OpenStack

RYU in OpenStack

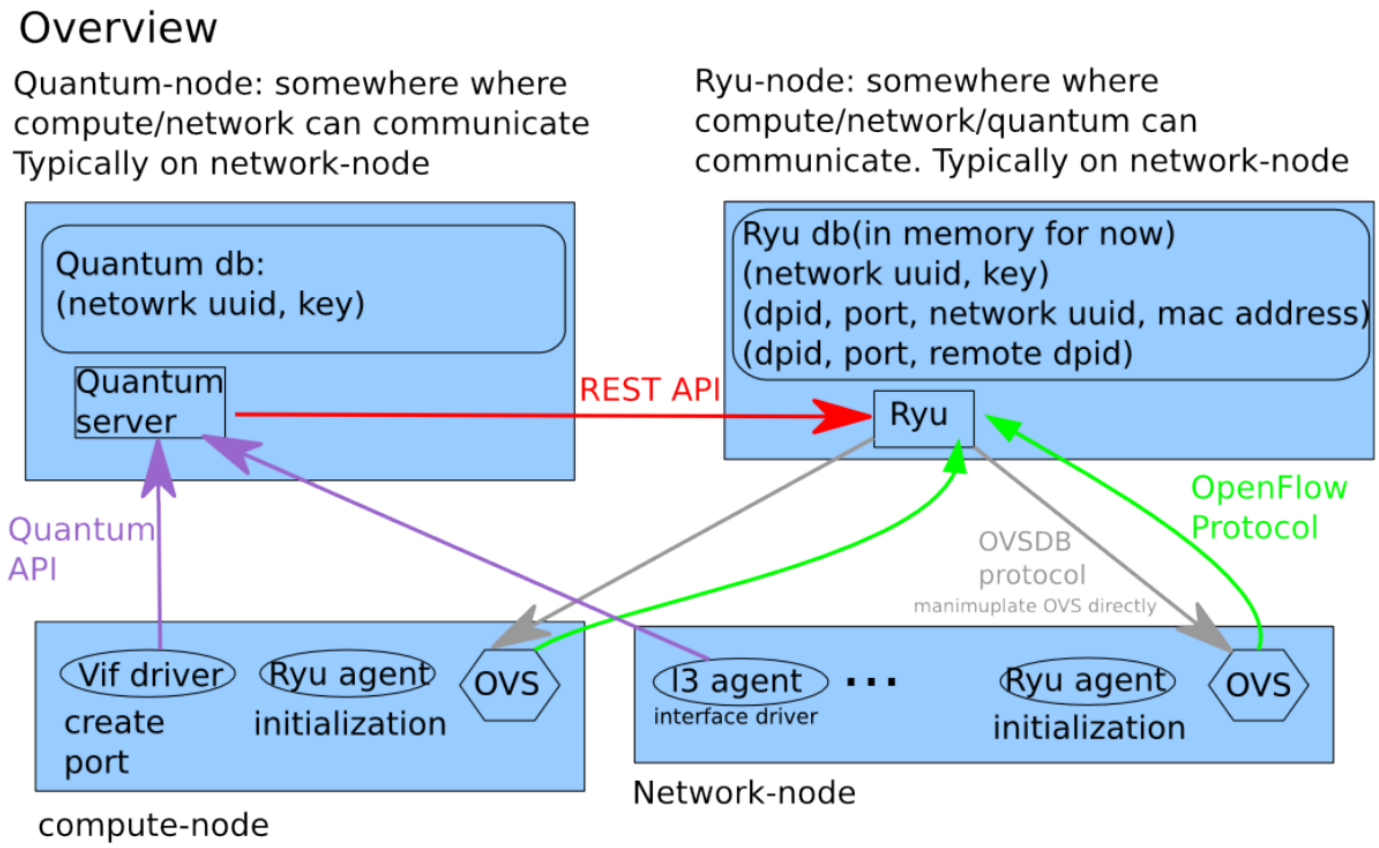
- Getting the nuts and bolts together
- A simple architecture on one node



compute node: single NIC

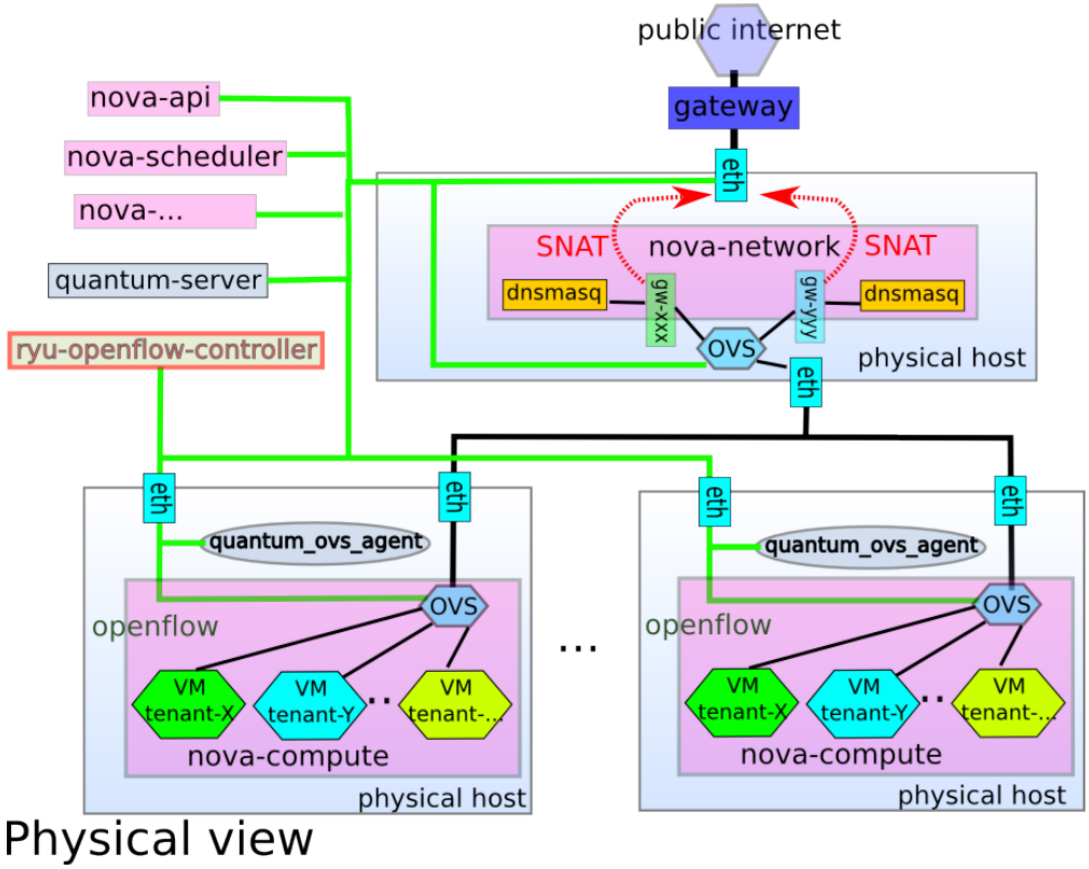
RYU in OpenStack

- L2-Isolation



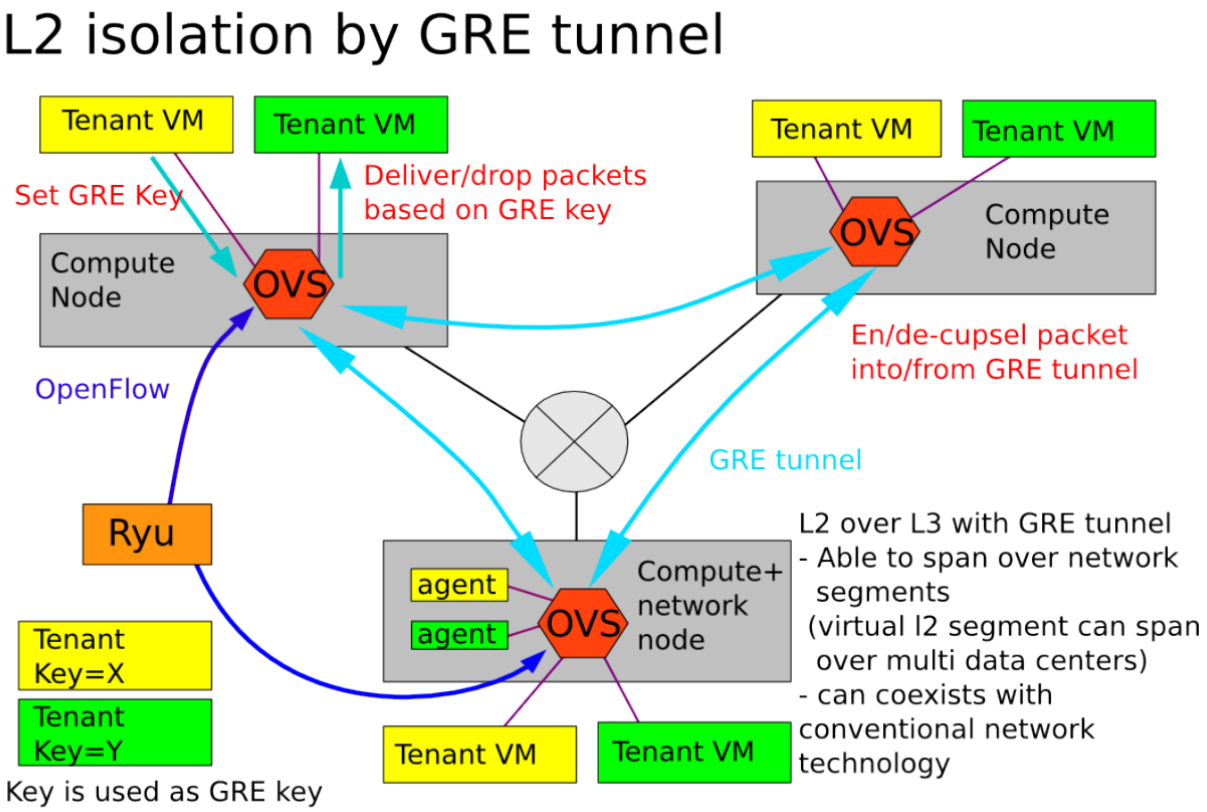
RYU in OpenStack

- A multi node deployment



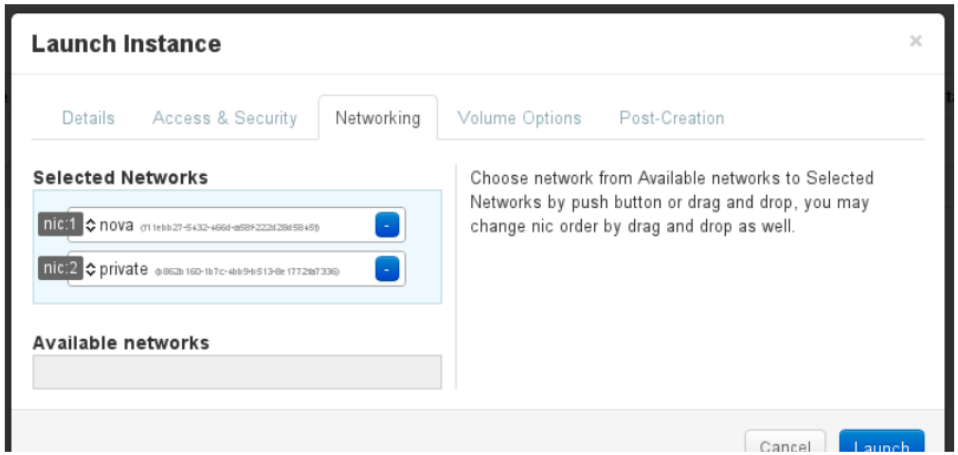
RYU in OpenStack

- RYU GRE Tunnel



Result of SDN in OpenStack

- NaaS in action!



Instances

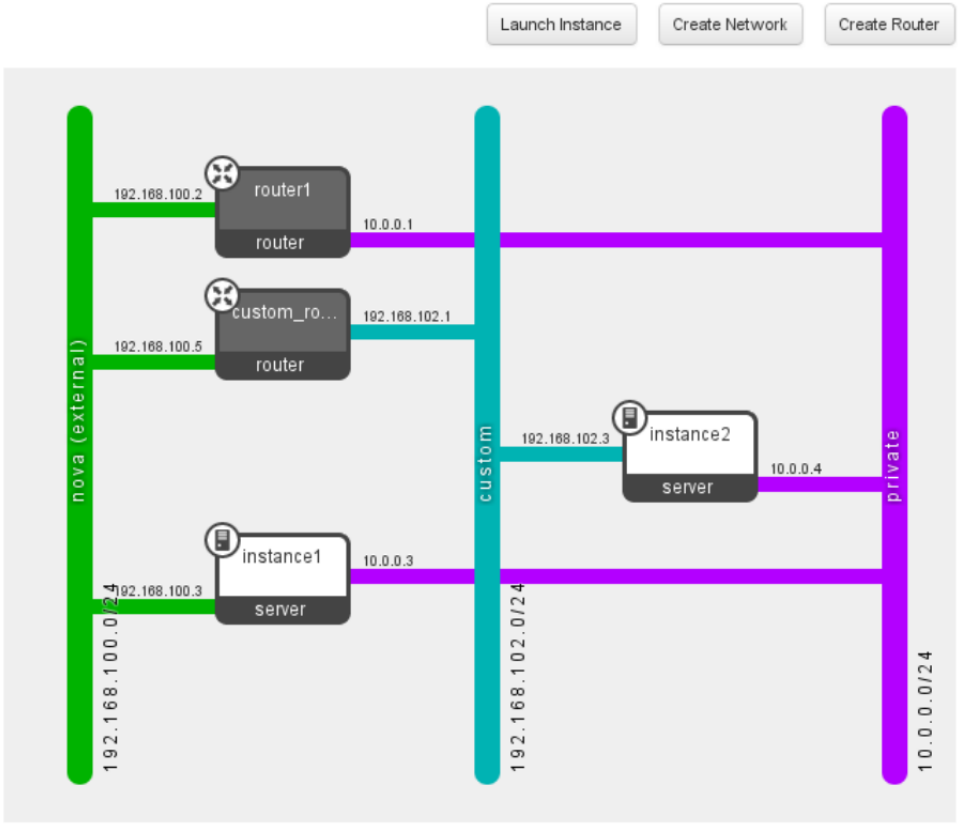
+ Launch Instance | Terminate Instances

Instance Name	IP Address	Size	Keypair	Status	Task	Power State	Actions
instance2	**private** 10.0.0.4 **custom** 192.168.102.3	m1.tiny | 512MB RAM | 1 VCPU | 0 Disk	-	Active	None	Running	Create Snapshot | More
instance1	**nova** 192.168.100.3 **private** 10.0.0.3	m1.tiny | 512MB RAM | 1 VCPU | 0 Disk	-	Active	None	Running	Create Snapshot | More

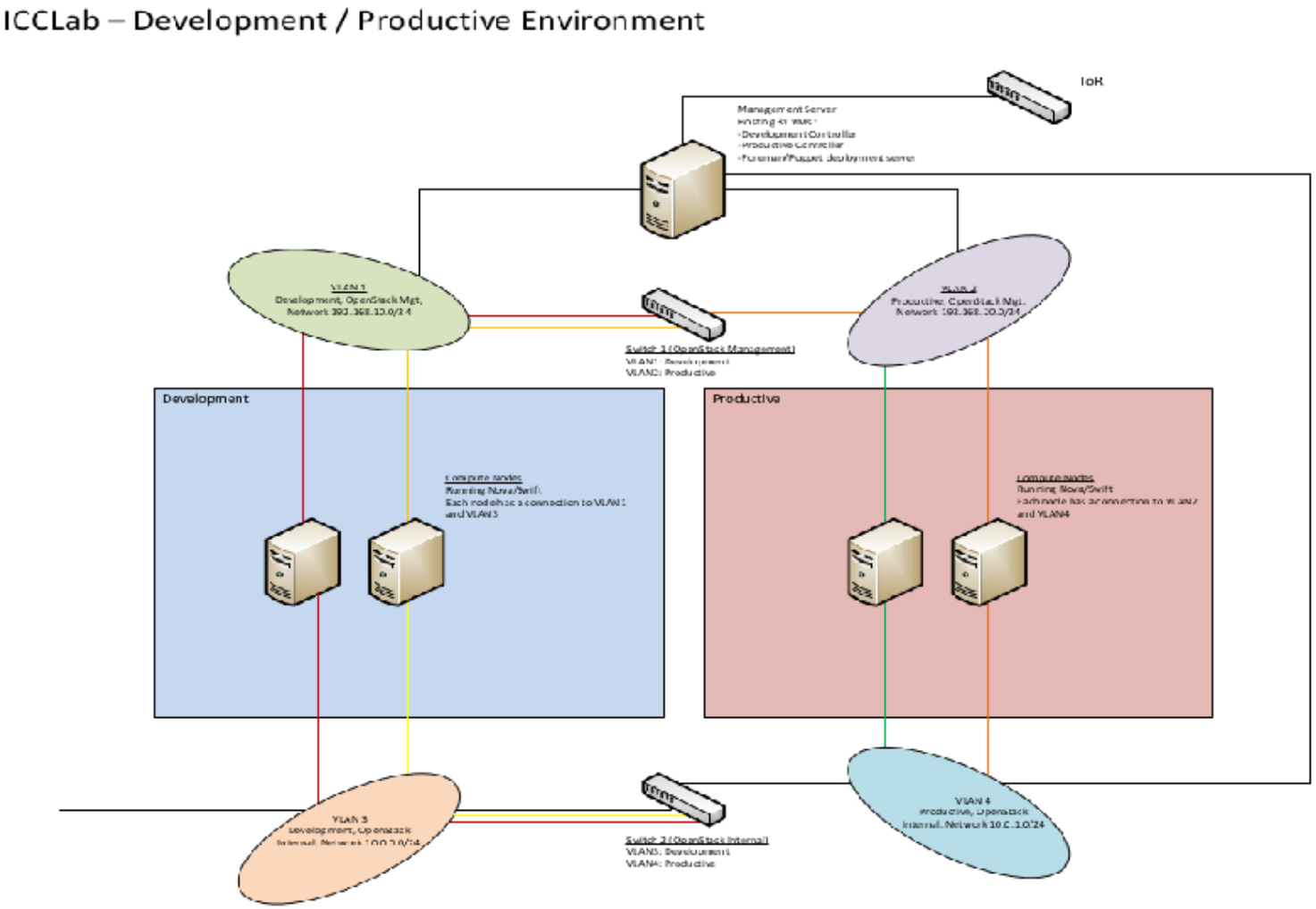
Displaying 2 items

Result of SDN in OpenStack

- NaaS in action!



ICCLab Current Deployments



Remarks

- (Our) Biggest challenge is to control both, virtual and physical networks
- Generally, networking research community very focused on OpenFlow development, not so much OpenFlow usage
- SDN means shift from Network Configuration to Network Programming
  - Software Development Best Practices!
  - SDN SDK

Thanks for your attention