



Underneath OpenStack Quantum: Software Defined Networking with Open vSwitch

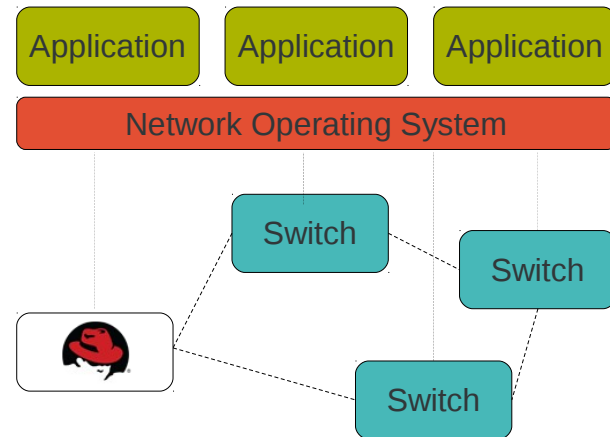
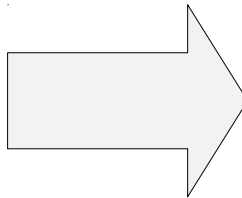
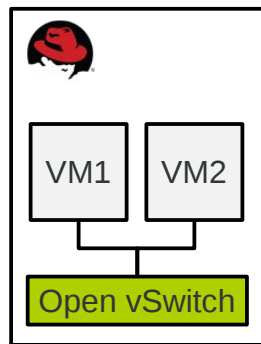
Thomas Graf <tgraf@redhat.com>
Principal Software Engineer
Red Hat, Inc.

April 24, 2013

Part One

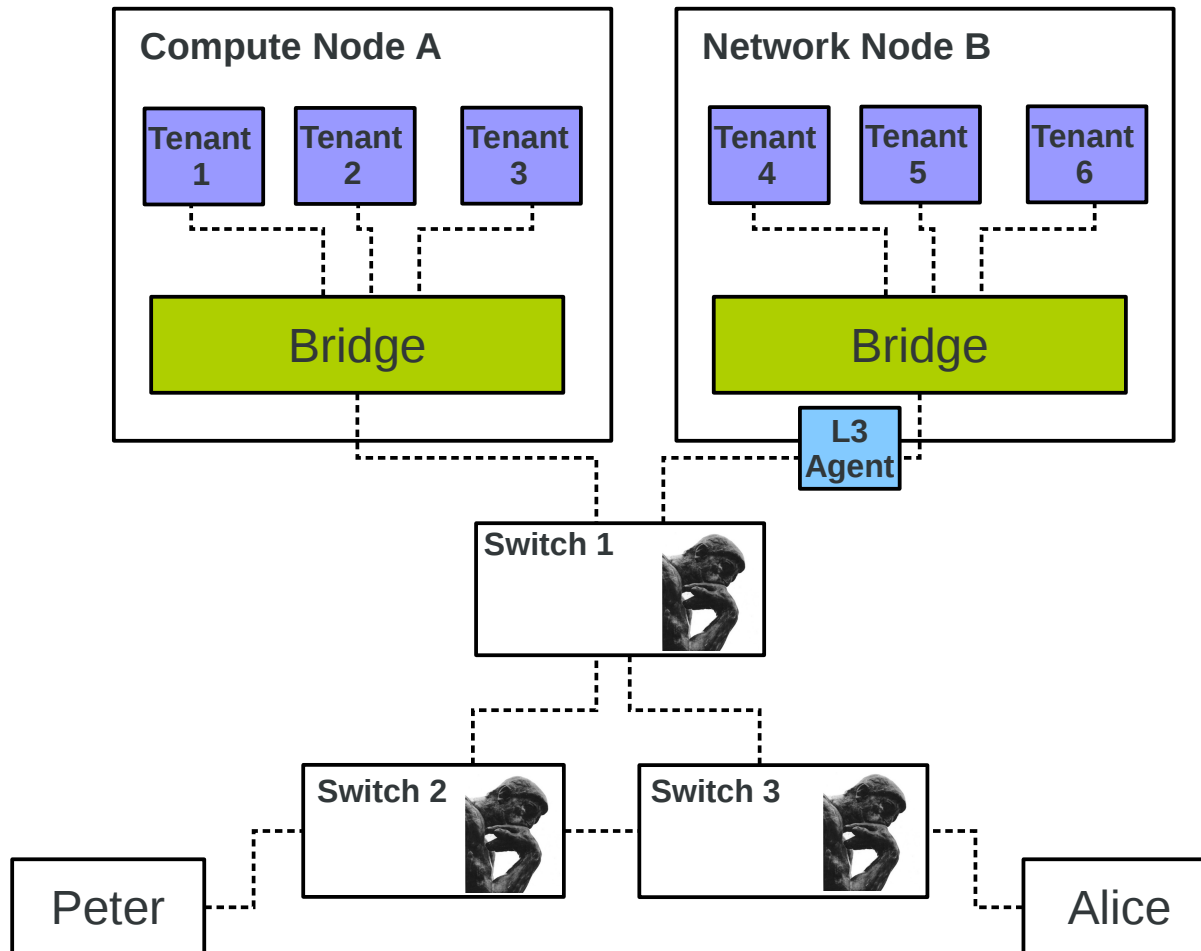
Why Open vSwitch?

Open vSwitch enables Linux to become part of a **Software Defined Networking** architecture.



Switched Networks

Switches **learn** from the network traffic they observe and **decide independently**.



Dynamically update flow tables in a universal language.

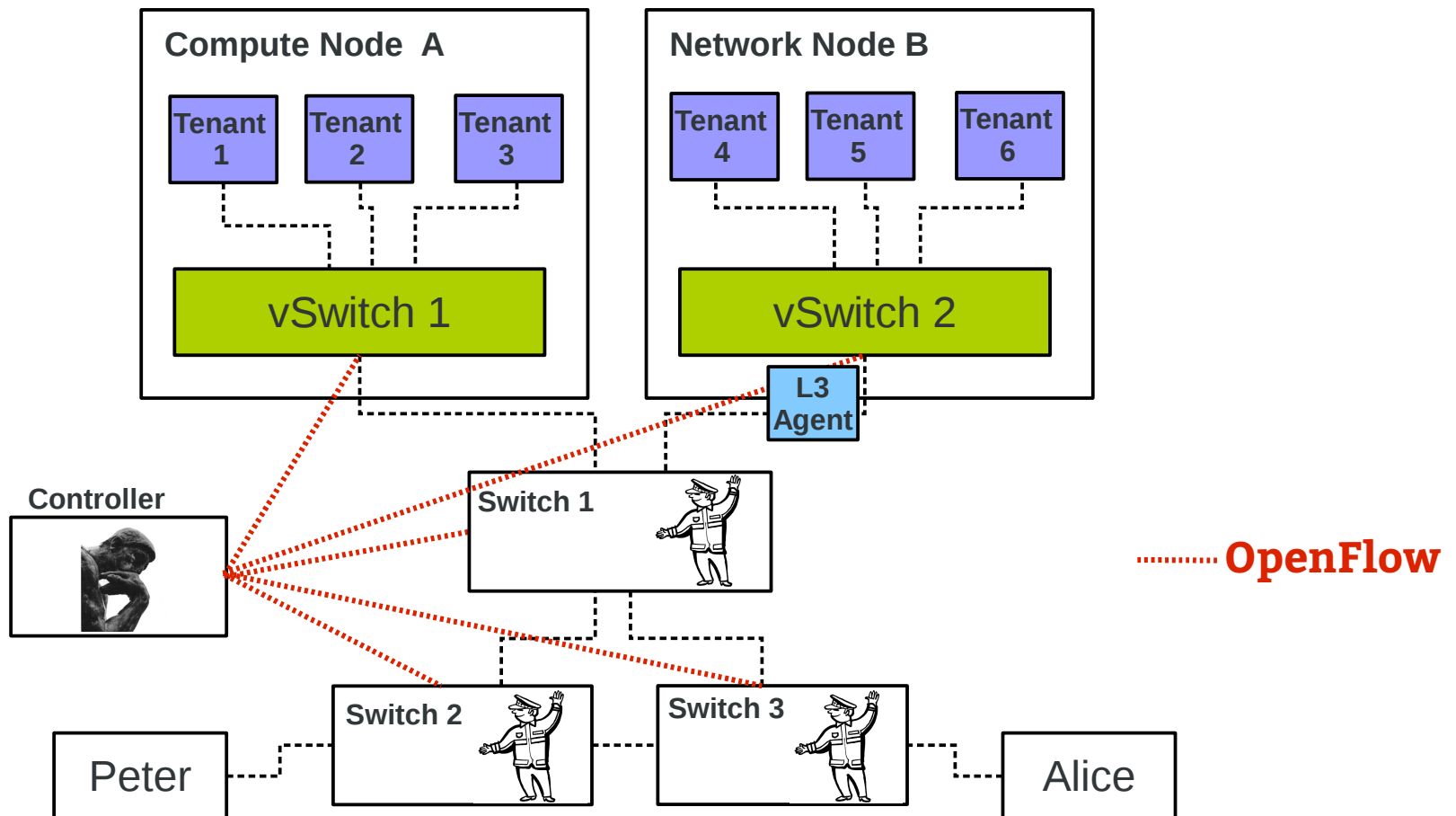
In the Software Defined Networking architecture, the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications.

Software-Defined Networking:
The New Norm for Networks
ONF White Paper
April 13, 2012



Software Defined Networking

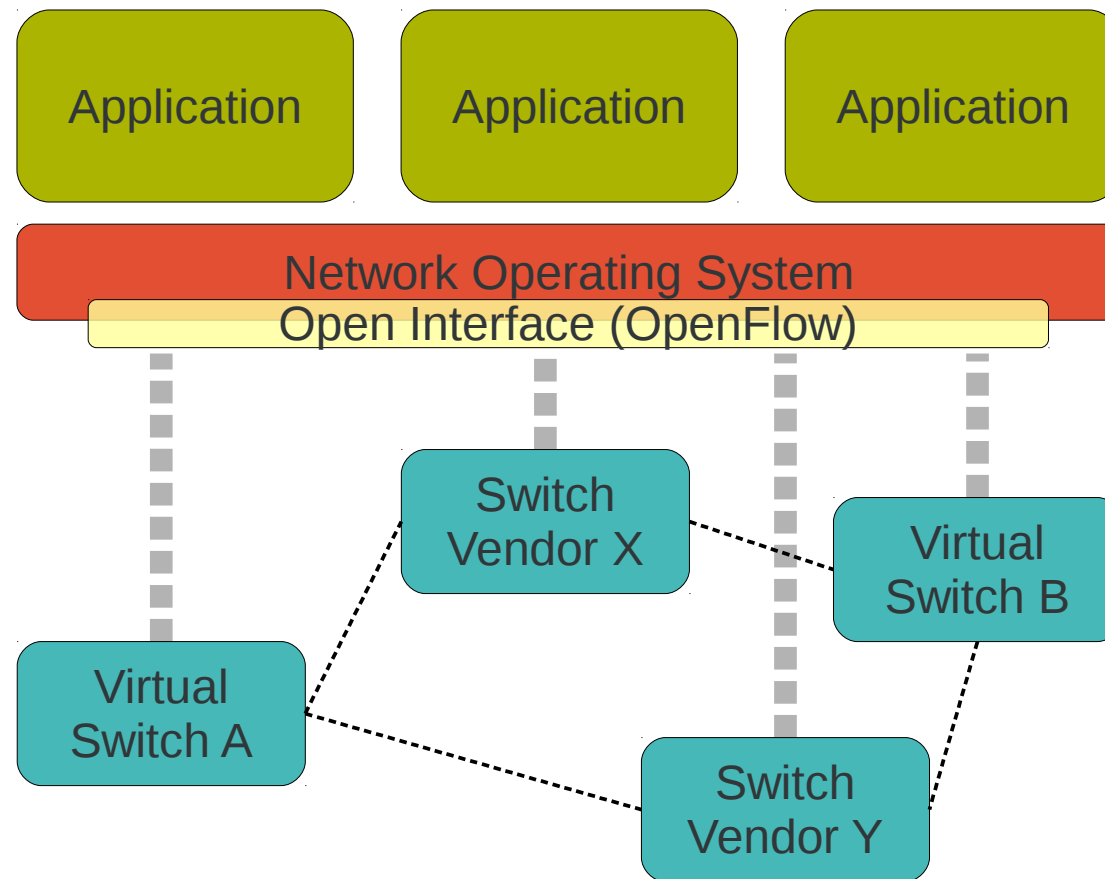
A **logically centralized** controller decides what is best for the network based on a **global view** of the network.



Software Defined Networking

An attempt to create a well-known API for applications of the Network that did not succeed **yet**.

OpenDaylight on its way to make this happen.



OpenFlow



The **Open Standard** behind it.

1. Match on arbitrary bits in packet (header)

2. Execute actions

- Forward to port
- Drop
- Send to controller
- Mangle packet

OpenFlow enables networks to evolve, by giving a remote controller the power to modify the behavior of network devices, through a well-defined "forwarding instruction set". The growing OpenFlow ecosystem now includes routers, switches, virtual switches, and access points from a range of vendors.



OpenFlow Capable Devices

- Software Switches
 - Open vSwitch, Cisco Nexus 1000V
 - VMware vSphere, NEC Hyper-V, ...
- Hardware Switches
 - Brocade, Cisco, HP, IBM, Juniper Networks, NEC, ...
- Switching ASICs
 - Indigo – Open source firmware leveraging Ethernet switch ASICs to support up to 48x 10G ports
 - Mellanox SwitchX-2 chip



Is it production ready?

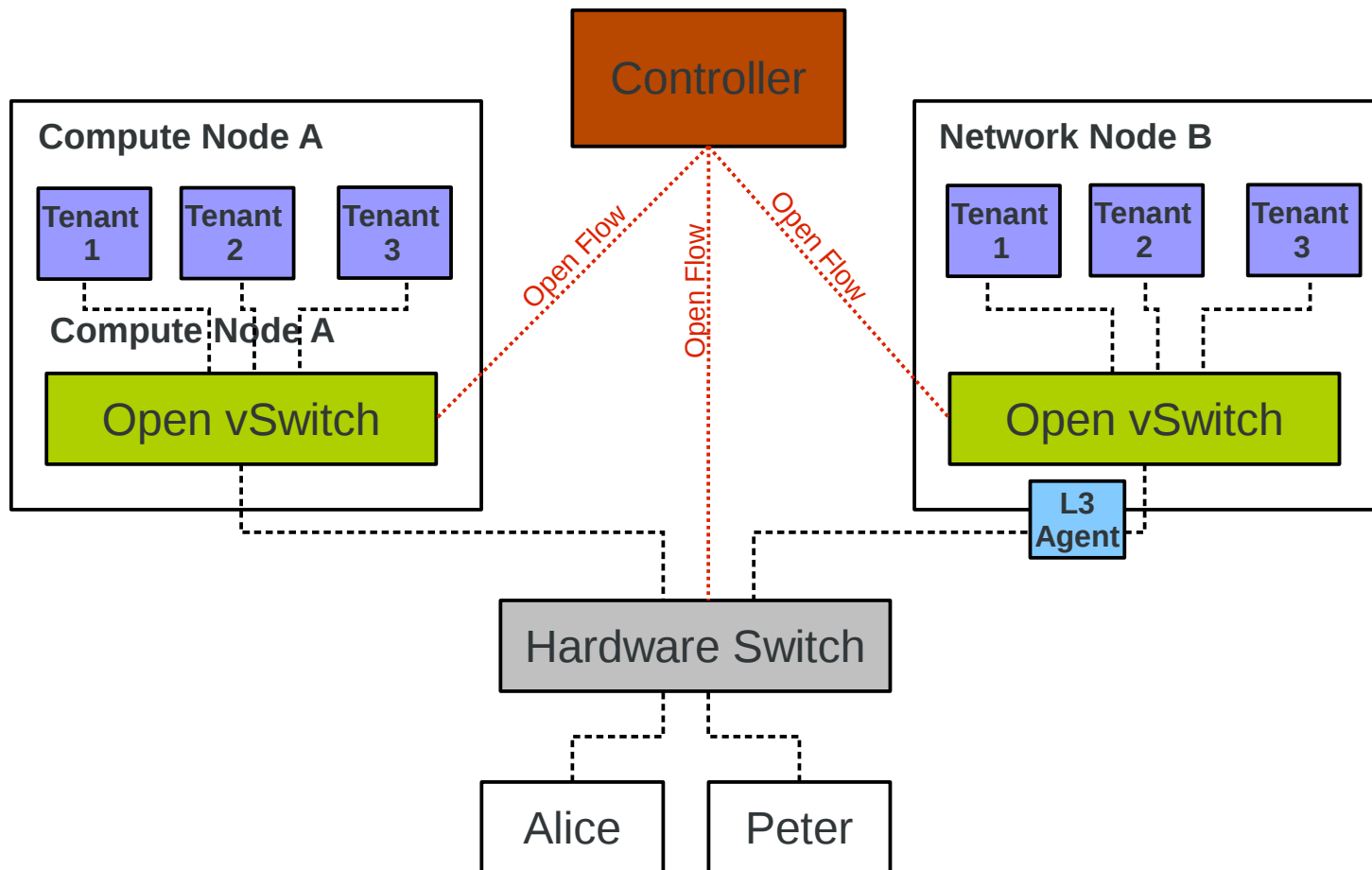
Google's Software Defined WAN



Part Two

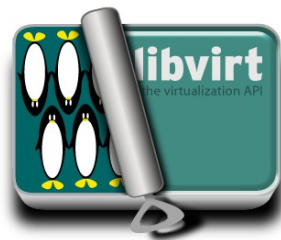
Open vSwitch

Open vSwitch is a **virtual switch** for hypervisors providing network connectivity to virtual machines.



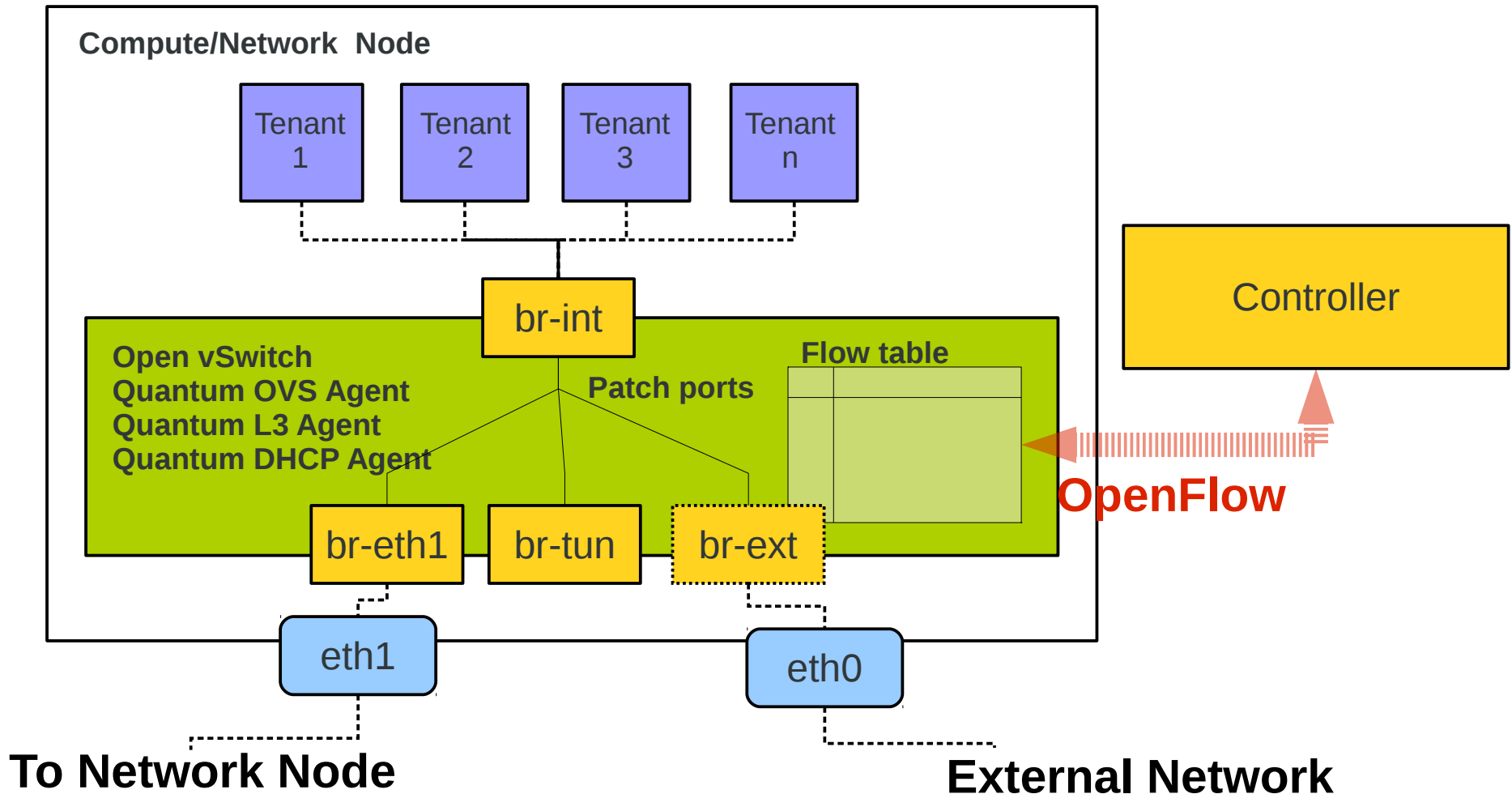
Open vSwitch Project

- Primarily used as a virtual switch for VMs
- Multi Platform (Linux, Microsoft, and Silicon)
- Developed by Nicira & Community
- Apache License (User Space), GPL (Kernel)
- OpenFlow 1.1 + extensions
- Any netdevice (physical/virtual) can be added as uplink port



How does it work?

Open vSwitch maintains a **flow table** that defines what to do with each flow.



Feature

Fine Grained Flow Table Control

- Extensive flow matching capabilities
 - Layer 1 – Tunnel ID, In Port, QoS priority, skb mark
 - Layer 2 – MAC address, VLAN ID, Ethernet type
 - Layer 3 – IPv4/IPv6 fields, ARP
 - Layer 4 – TCP/UDP, ICMP, ND
- Possible chain of actions
 - Output to port (port range, flood, mirror)
 - Discard, Resubmit to table x
 - Packet Mangling (Push/Pop VLAN header, TOS, ...)
 - Send to controller, Learn

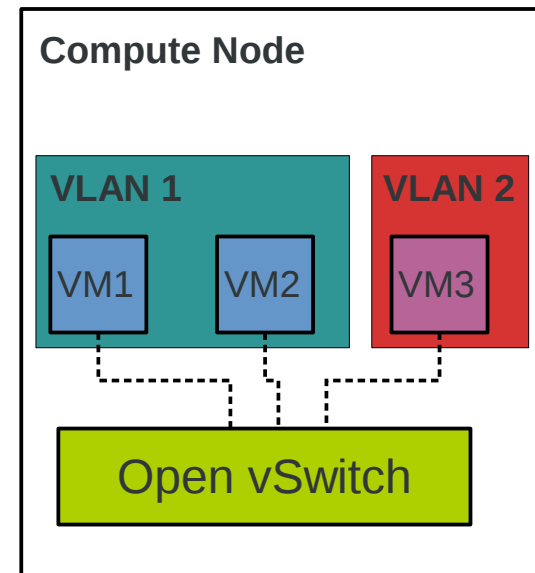


Feature

Security / L2 Segregation

VLAN isolation enforces **VLAN membership** of a VM without the knowledge of the guest itself.

```
# ovs-vsctl add-port ovsbr port2 tag=10
```



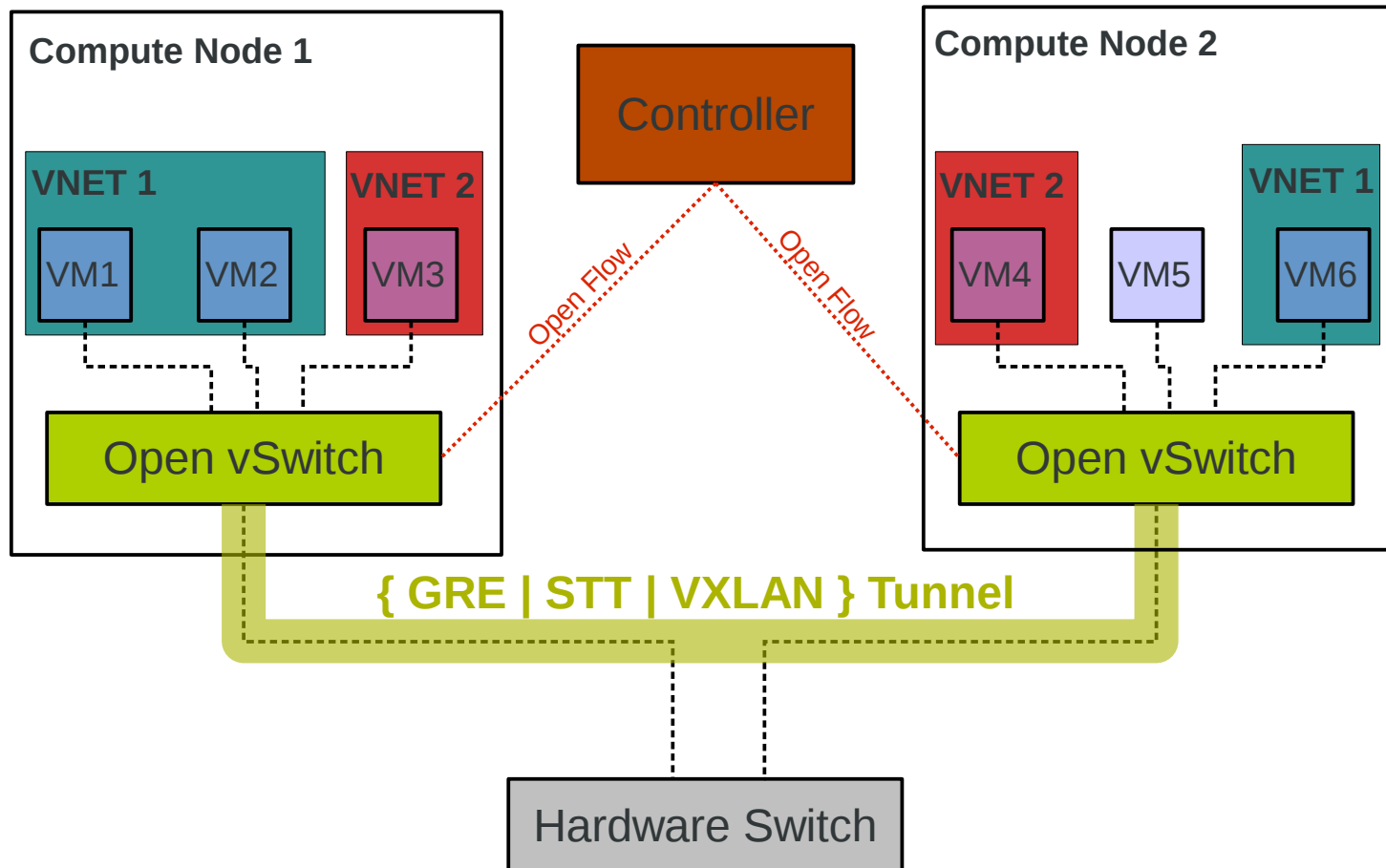
Caveat: MAX(VLAN_ID) limited



Feature

Tunneling

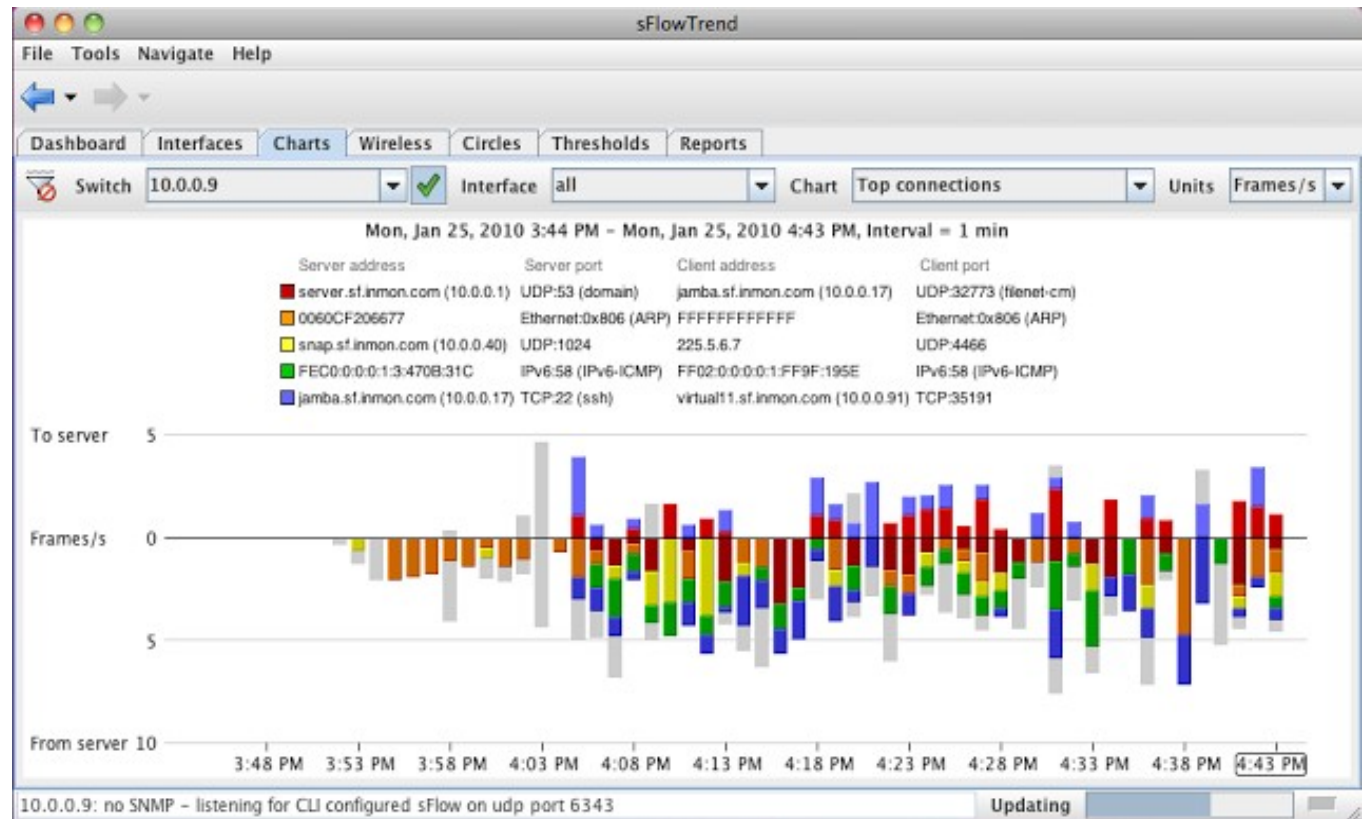
Tunneling provides **isolation** and reduces **dependencies** on the physical network.



Feature Visibility

Supports **industry standard technology** to monitor the use of a network.

- sFlow
- NetFlow
- Port Mirroring
 - SPAN
 - RSPAN
 - ERSPAN



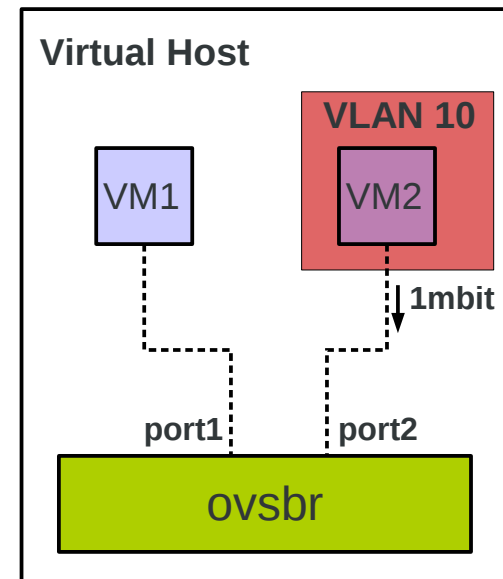
Feature

Quality of Service

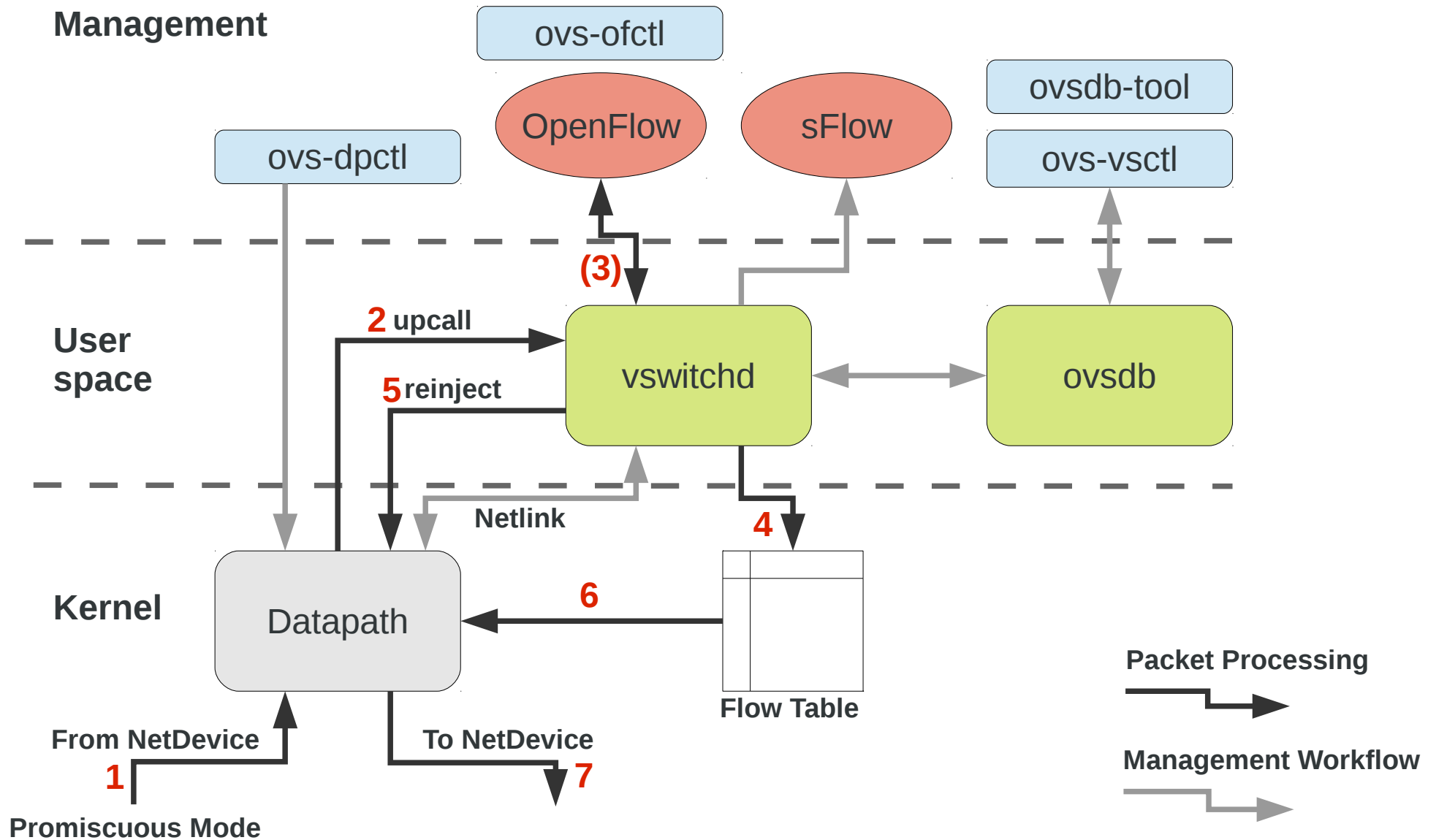


- Uses **existing Traffic Control Layer**
 - Policer (Ingress rate limiter)
 - HTB, HFSC (Egress traffic classes)
- Controller (Open Flow) can select Traffic Class

```
# ovs-vsctl set Interface port2 \  
ingress_policing_rate=1000
```



Architecture



Modifying the Flow Table

Strip VLAN header of all packets from MAC address 11:22:33:44:55:66 and forward packet to port 1.

```
# ovs-ofctl add-flow ovsbr \  
  dl_src=11:22:33:44:55:66,actions=strip_vlan,output:1  
  
# ovs-ofctl dump-flows ovsbr  
[...]  
cookie=0x0, duration=36.24s, table=0, n_packets=0,  
n_bytes=0, idle_age=36, dl_src=11:22:33:44:55:66  
actions=strip_vlan,output:1
```



Questions?

- Open vSwitch
 - <http://www.openvswitch.org/>
- OpenFlow
 - <http://www.openflow.org/>
- Open Networking Foundation
 - <http://www.opennetworking.org/>
- sFlow
 - <http://www.sflow.org/>
- Going with the Flow: Google's Secret Switch to the Next Wave of Networking
 - <http://www.wired.com/wiredenterprise/2012/04/going-with-the-flow-google/>

