

# OpenStack GDL

## Dragonflow SDN + Other cool networking stuff in OpenStack



October 17, 2017

Pino de Candia @ Huawei



# Background

Dragonflow gives OpenStack the first network solution that:

- Is truly open source
- Scales
- Just works, without babysitting
- Is small and easy to extend
- Gives you choices:
  - to optimize for your use-case
  - and use tools you already know



# Open Source

- Dragonflow was developed in OpenStack from day 1
- Led by Huawei, but intended for the community, not a commercial product
  - Not the case for OpenContrail, MidoNet, Calico and others.
- Full control with OpenStack.org
  - Not the case with OVN, OpenContrail and others.

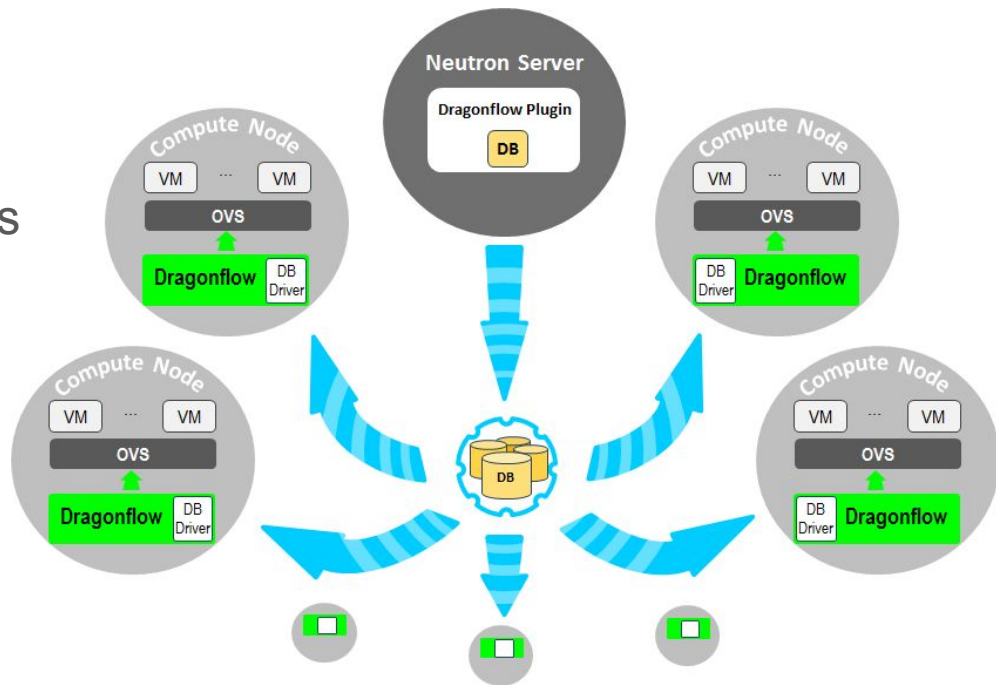


# Scale

Requires distributing everything.

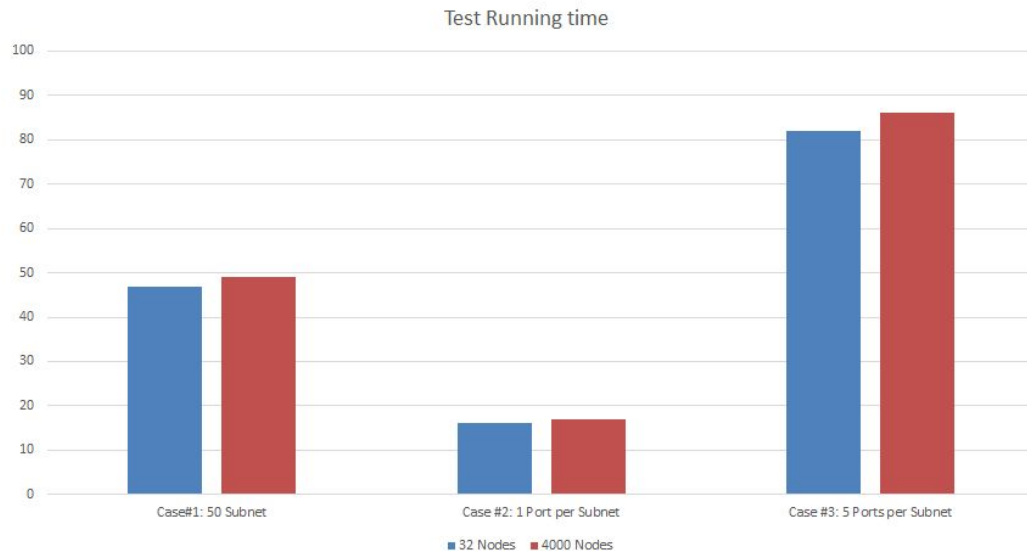
So the only centralized component is the Database cluster used as the source of truth for intent.

Push policy all the way to the edge.



# Tested scale to thousands of nodes

Deliver consistent network performance up to 4000 compute nodes (tests indicate significant room for improvement ).

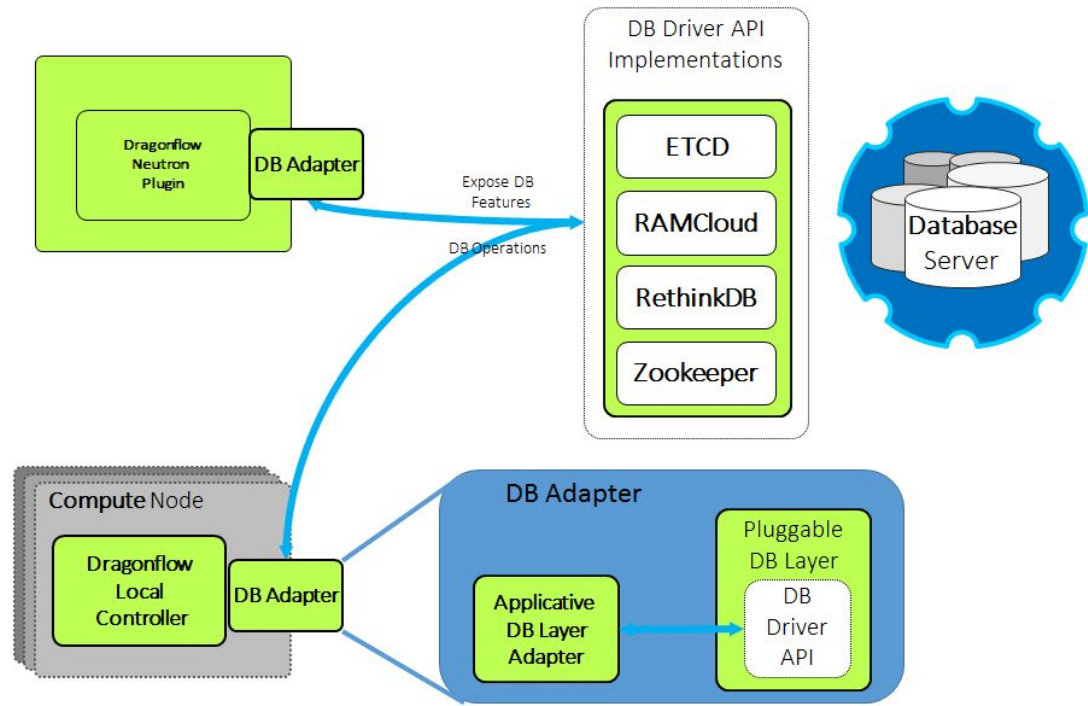


openstack



# Choose your own DB and PubSub

- Reduce the skill burden by using DB and PubSub technologies you already know or use elsewhere.
- Reduce the operational burden by reusing already deployed DB and PubSub clusters.



# Choose vlan, overlay, or flat networking

- There are good use-cases for each connectivity model.
- Dragonflow lets you choose per network (not per deployment).
- A workload can benefit from multiple networks, each optimized for different outcomes (performance vs. security or L2 isolation).
- This also allows flexibility in designing and evolving your physical network.

# No babysitting

Other solutions have too many centralized components that easily become bottlenecks or become overwhelmed.

- Dragonflow has all intelligence at the edge, except for the Database.

Other solutions have centralized datapath components, middleboxes.

- Dragonflow pushes more functionality to the datapath pipeline at the edge.  
Less code, more stability.





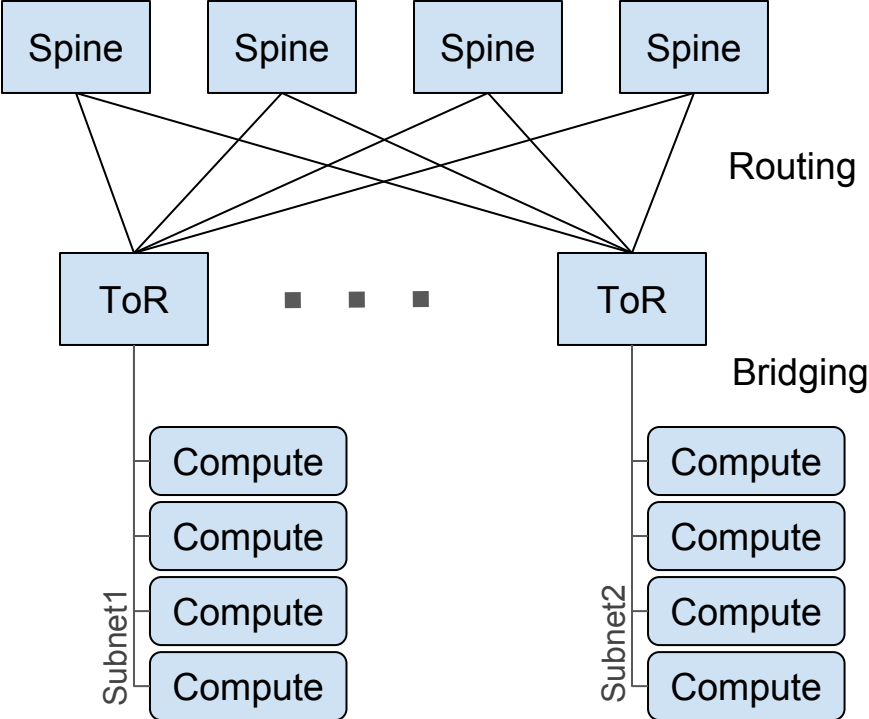
# Neutron without Network Nodes

DHCP

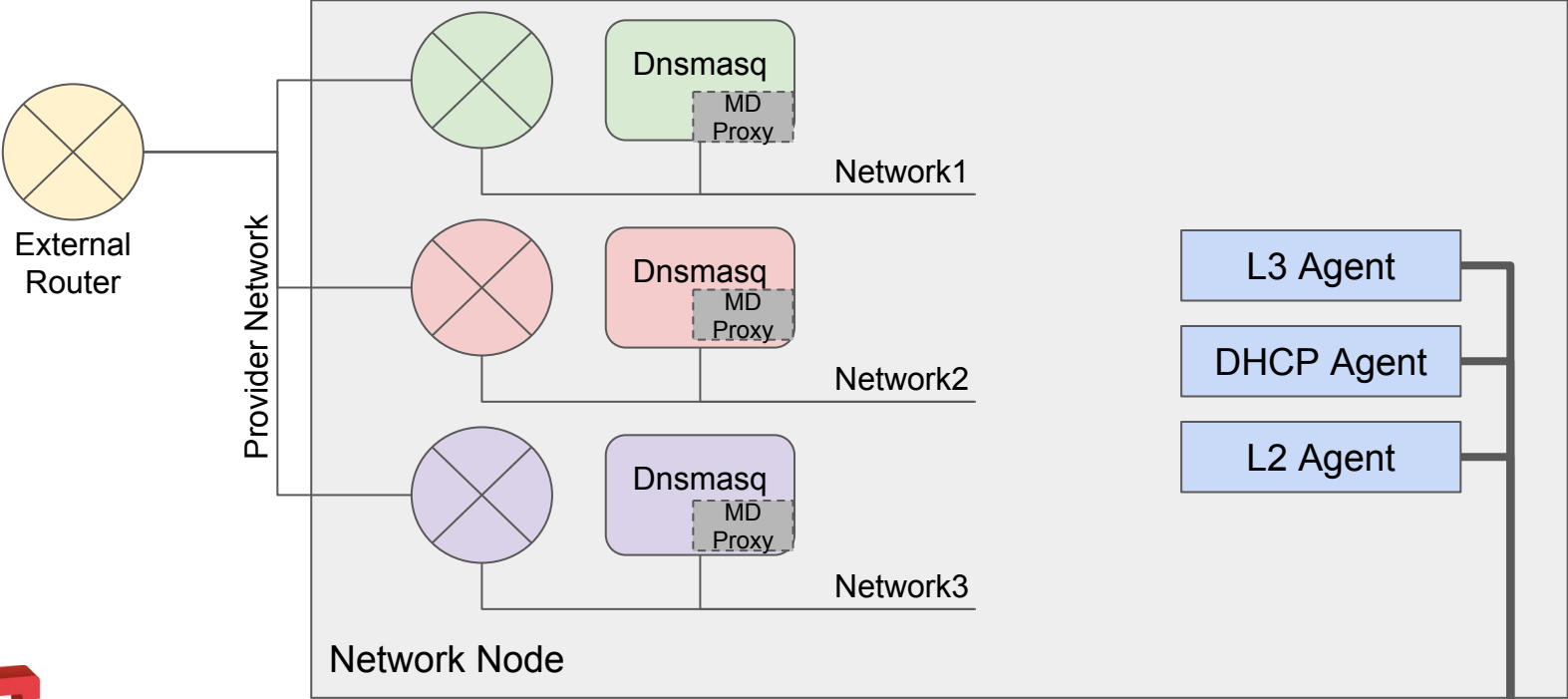
MetaData API proxy

FloatingIP

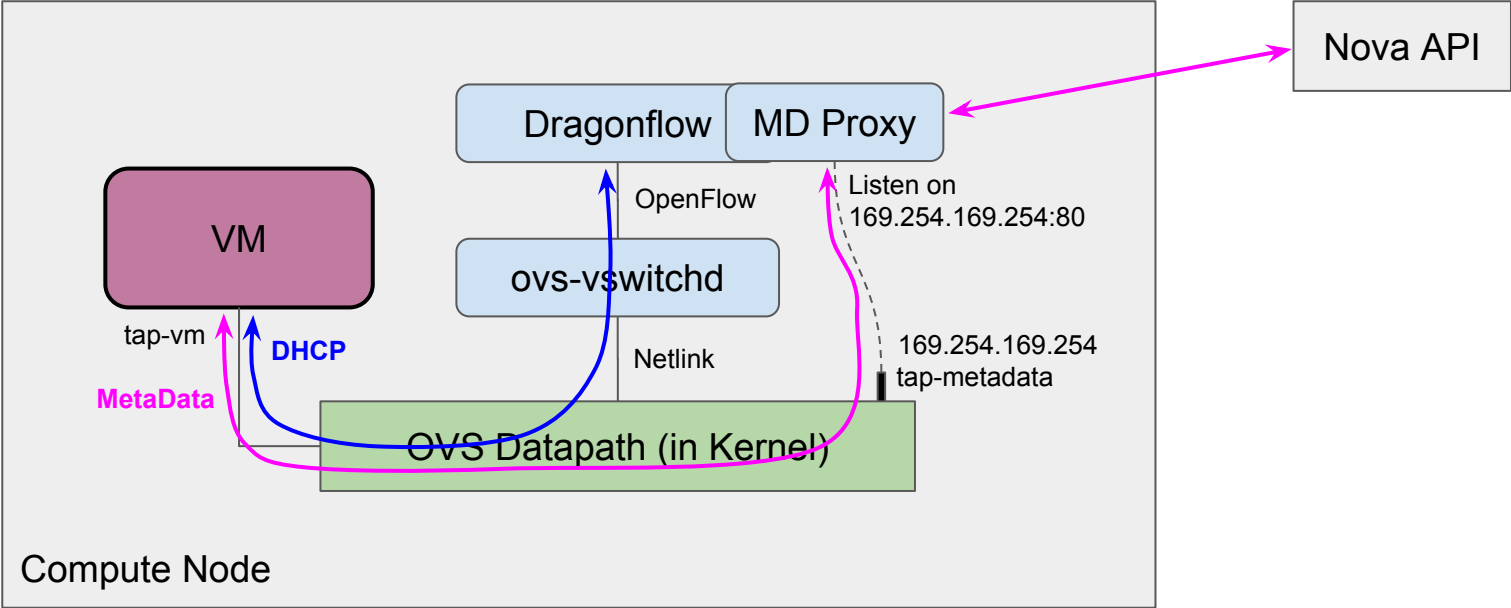
SNAT



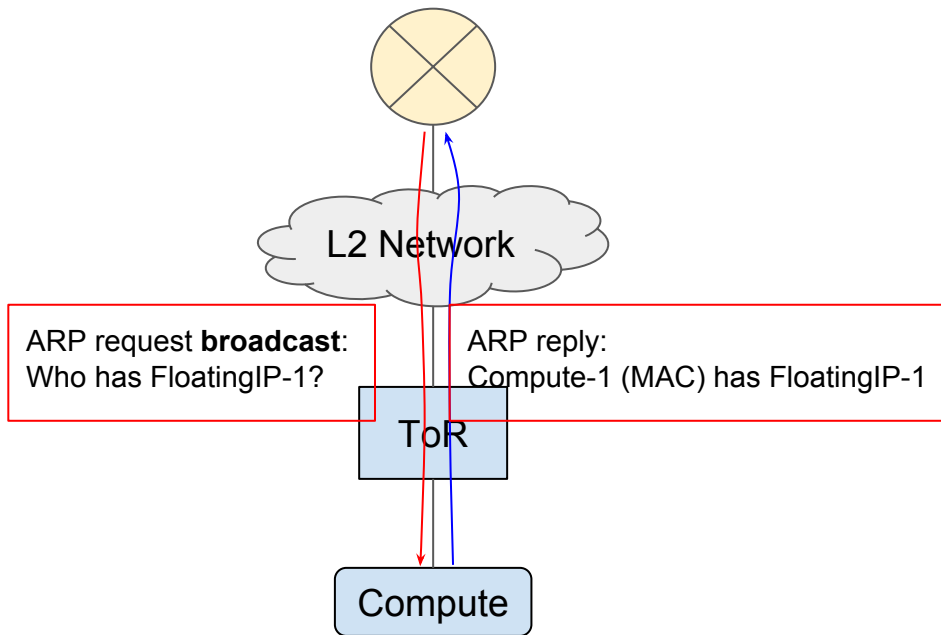
# Recap: what are Network nodes?



# DHCP and MD Proxy without Network Nodes

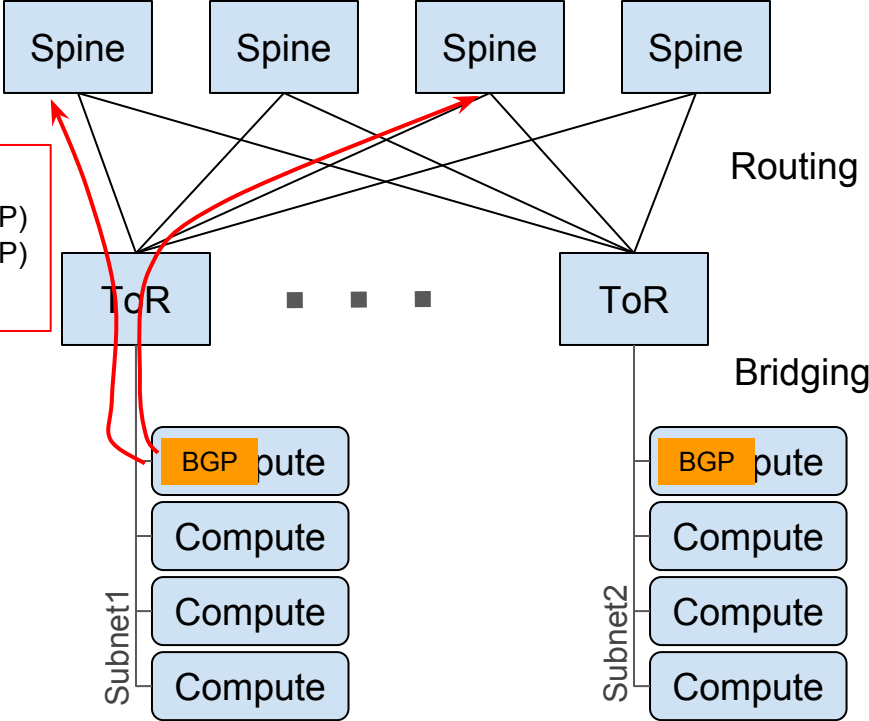


# Floating IP without Network Nodes (on L2)

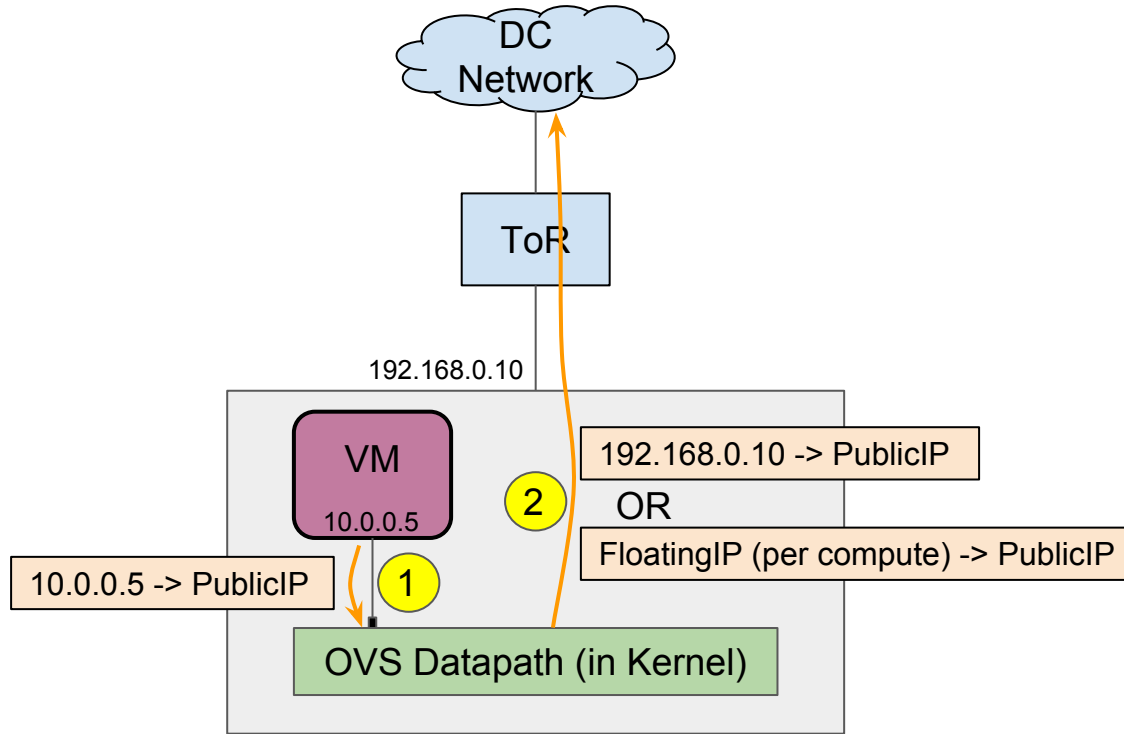


# Floating IP without Network Nodes (on L3)

BGP Advertisements:  
- FloatingIP-1 is at Compute-1 (IP)  
- FloatingIP-2 is at Compute-2 (IP)  
...etc



# SNAT without Network Nodes



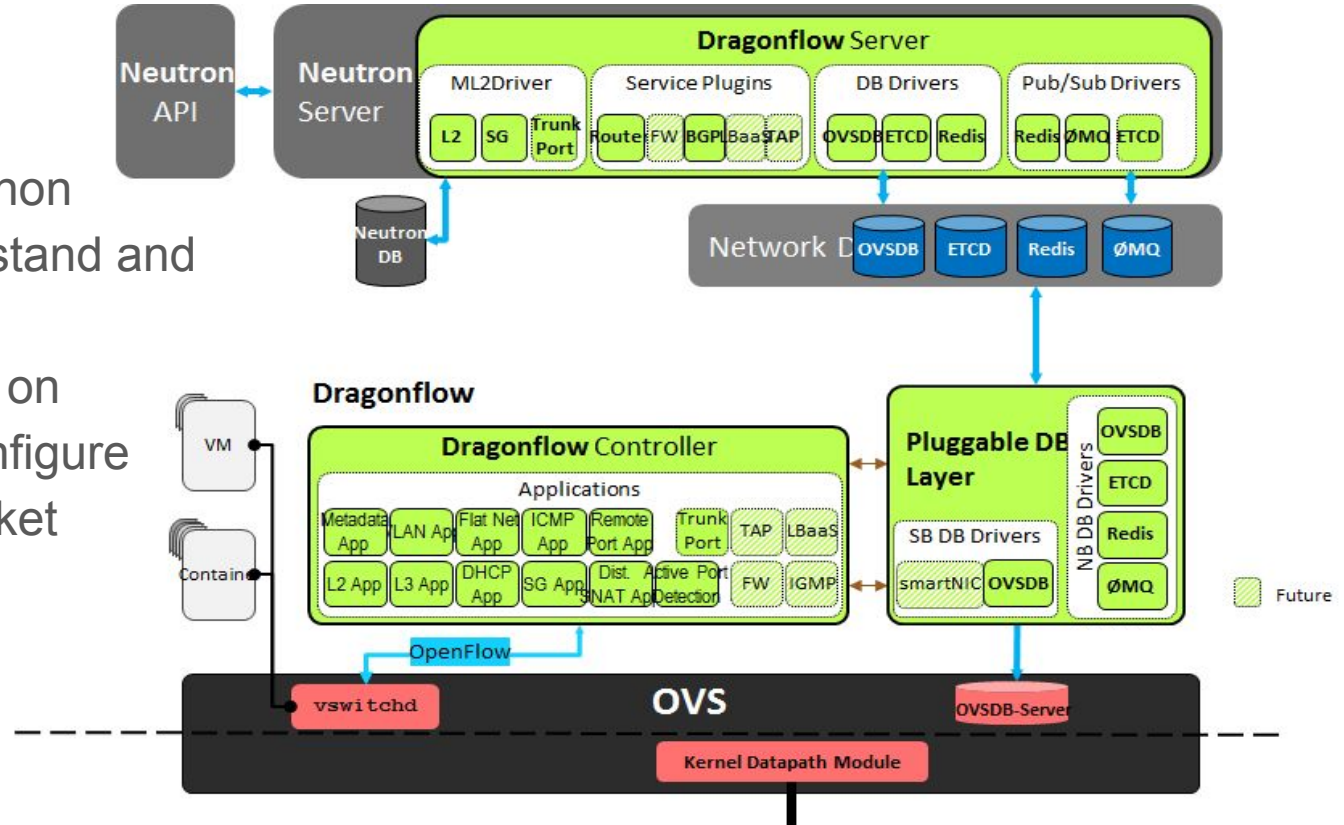
# Easy to add minor features without forking

- Don't wait for upstream for minor changes.
- Dragonflow modules let you experiment with new functionality.
- Many useful features can be added with tens of lines of code.
- Take back control of your network.



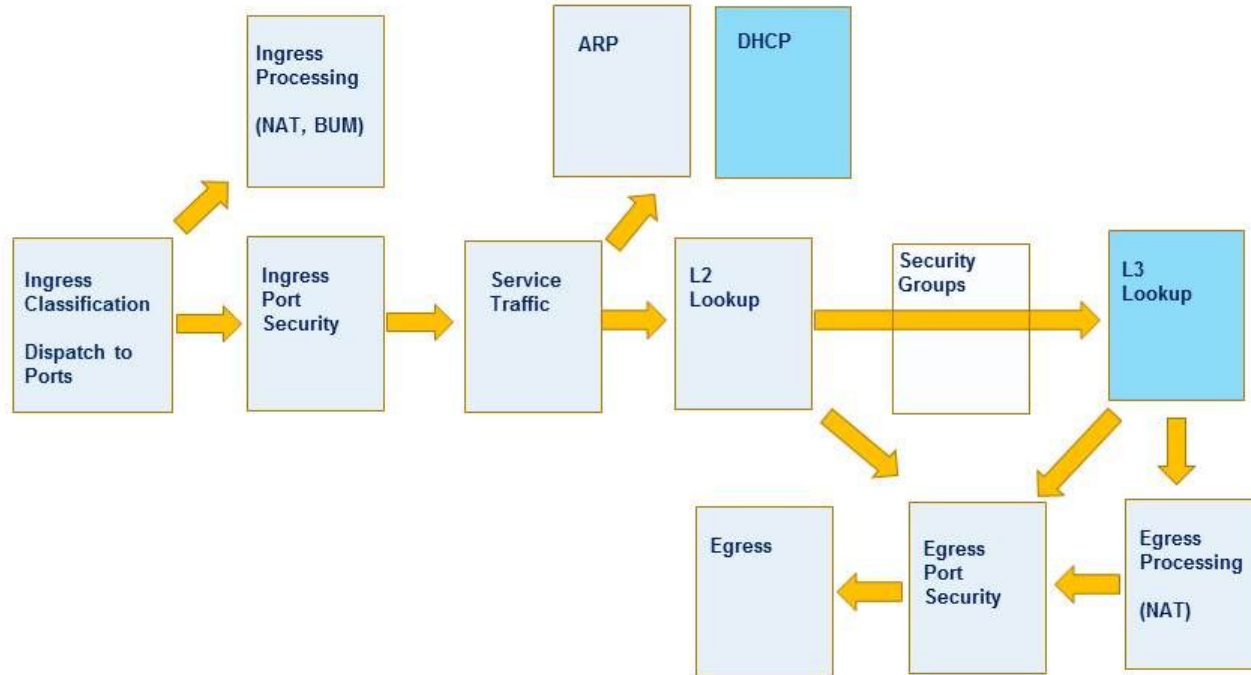
# Lightweight

- ~10K LoC in Python
- Is easy to understand and troubleshoot
- Design centered on modules that configure tables in the packet pipeline.





# DragonFlow Pipeline



# Dragonflow recently added features

<http://www.dragonflow.net/2017/10/pike-release-what-have-we-done-of-late.html>

- Hierarchical port binding
- IPv6 support
- Trunk ports
- Service function chaining
- Service health reports (Guru meditation reports)
- BGP Dynamic Routing
- Distributed SNAT (SNAT without network nodes)

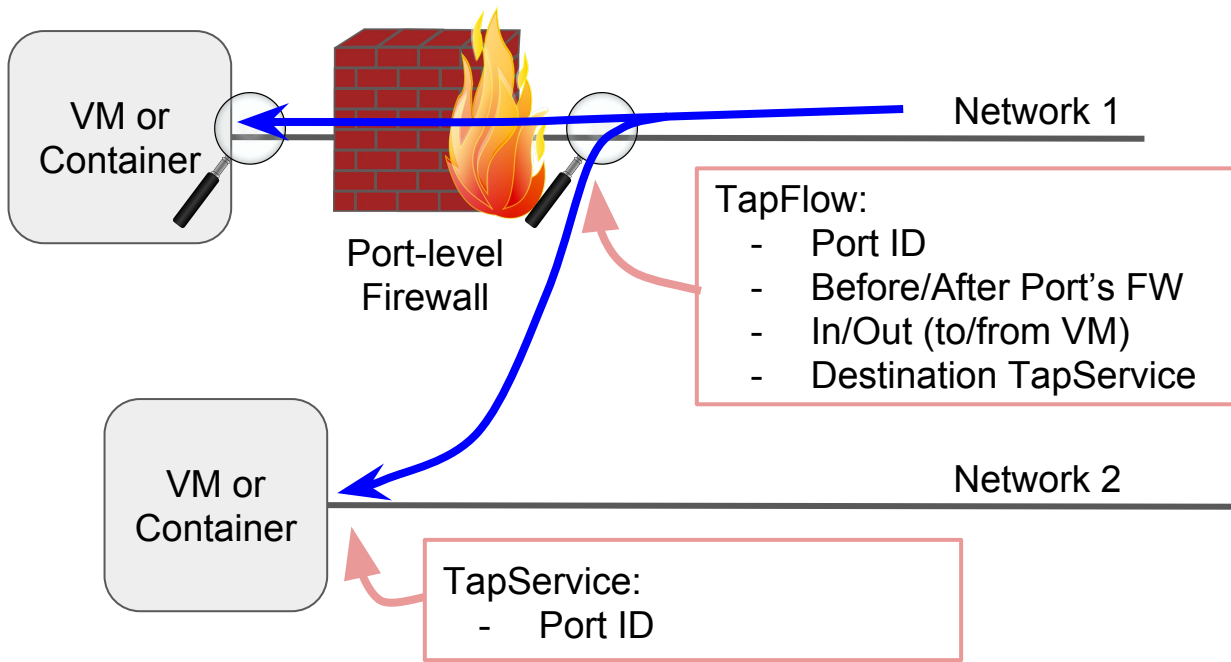


# Dragonflow upcoming features

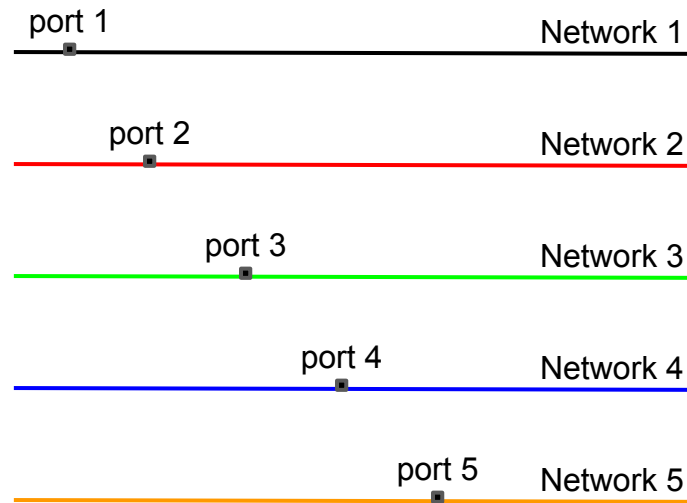
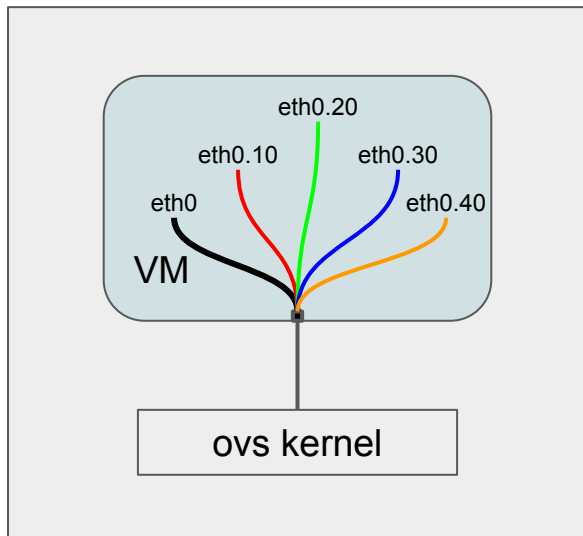
- Flat networking that supports overlapping IPs (using NAT)
  - Research code written but not yet upstream
- BGP MPLS eVPN and L3VPN
  - Research code written but not yet upstream
- Kubernetes support (with Kuryr and Kolla)
- Native LBaaSv2 API (without Octavia)
- TAPaaS
- FWaaS
- Native Ironic (Bare-metal) support



# Port Mirroring (TAP-as-a-Service)

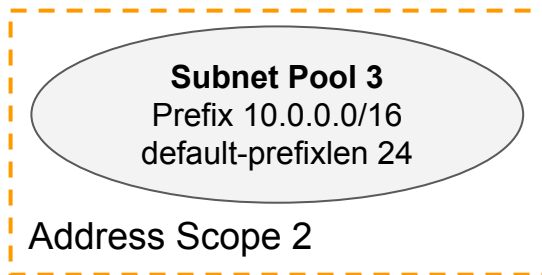
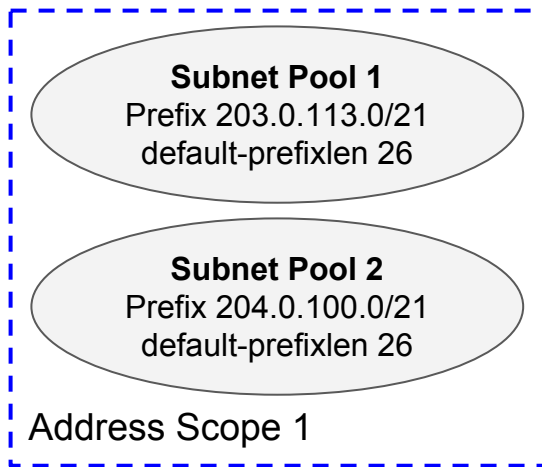


# Neutron trunk ports (a.k.a. vlan-aware VMs)

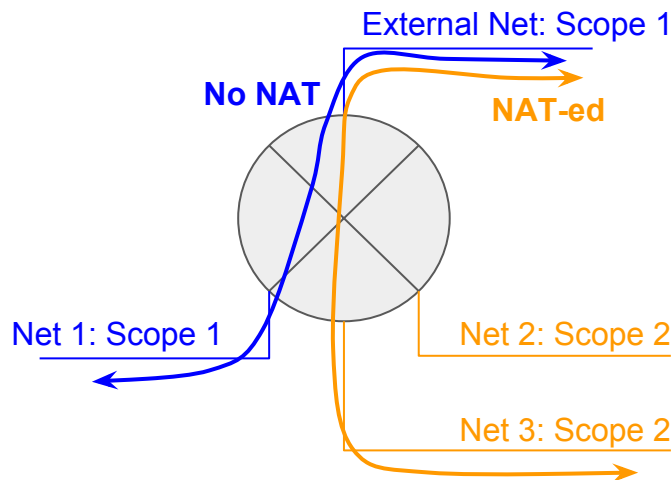


Example: port 1 is the trunk port;  
ports 2-5 are sub-ports of port 1.

# Subnet Pools and Address Scopes



Enable prefix delegation and no-NAT networking



# DPDK



- Out-of-kernel (user-space) packet processing
- Using dedicated CPUs, huge pages, ring buffers
- No interrupts

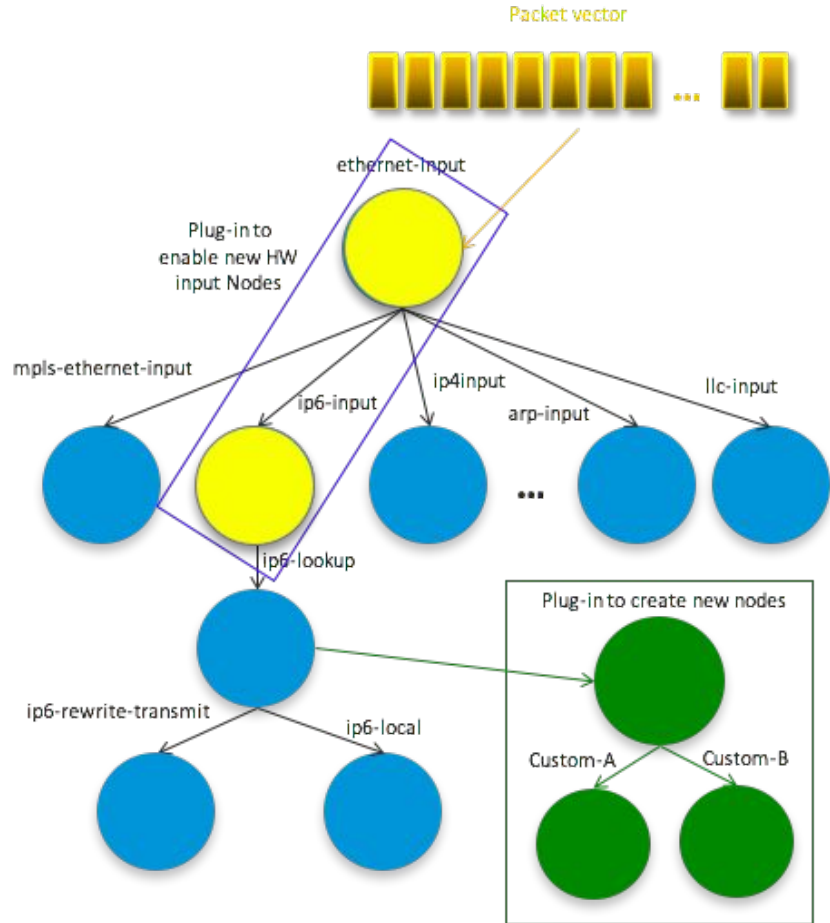
Very low level, all about moving data on/off an interface. NOT a switch, doesn't understand network protocols.

Open vSwitch has a mode where the datapath runs in the same process as vswitchd and uses DPDK.



# FD.io and VPP

- Vector Packet Processing
- Donated by Cisco
- Uses DPDK and AF\_PACKET





# eBPF

- Allow Linux user-space programs (in assembly-like language)
- To run safely in the kernel (JIT compiled and safety checked)
- Multiple hook points (notably TC - traffic control)

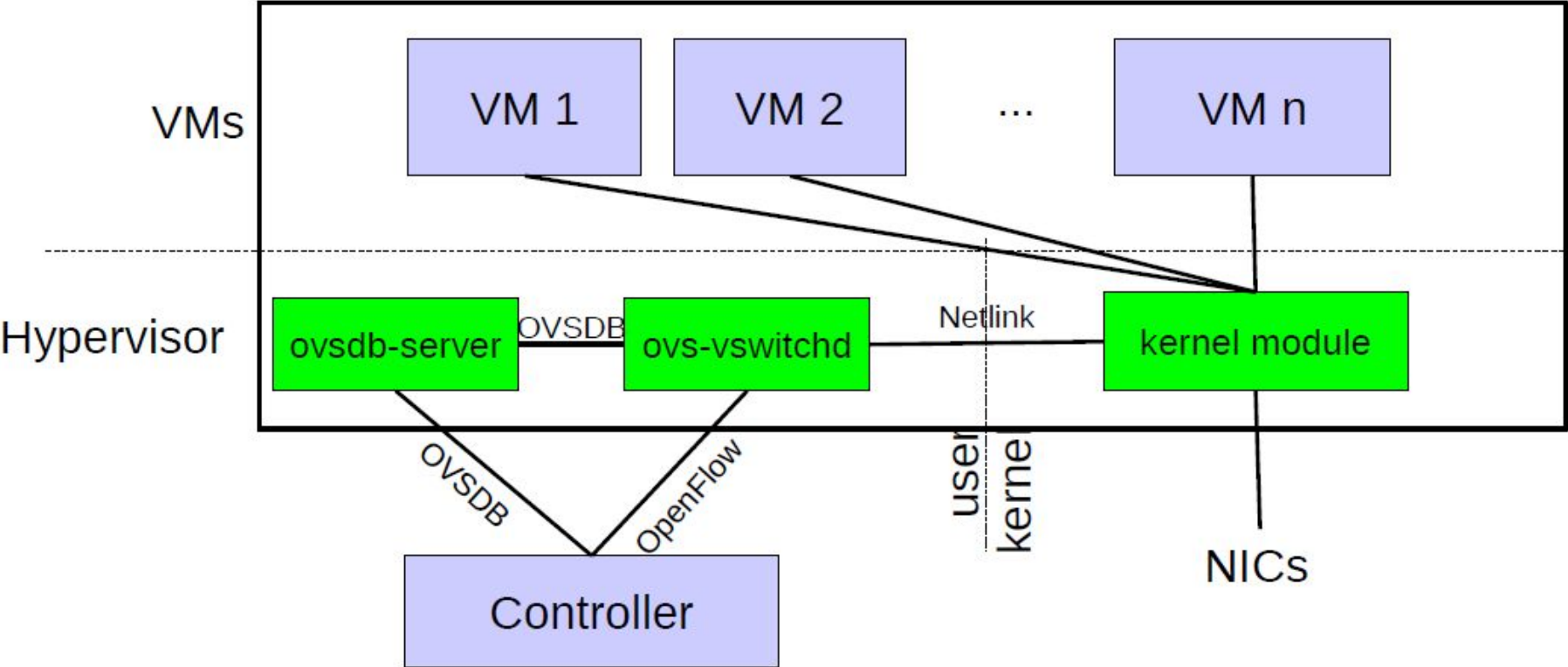
Innovate faster outside the kernel.

Unlock better network performance in-kernel.

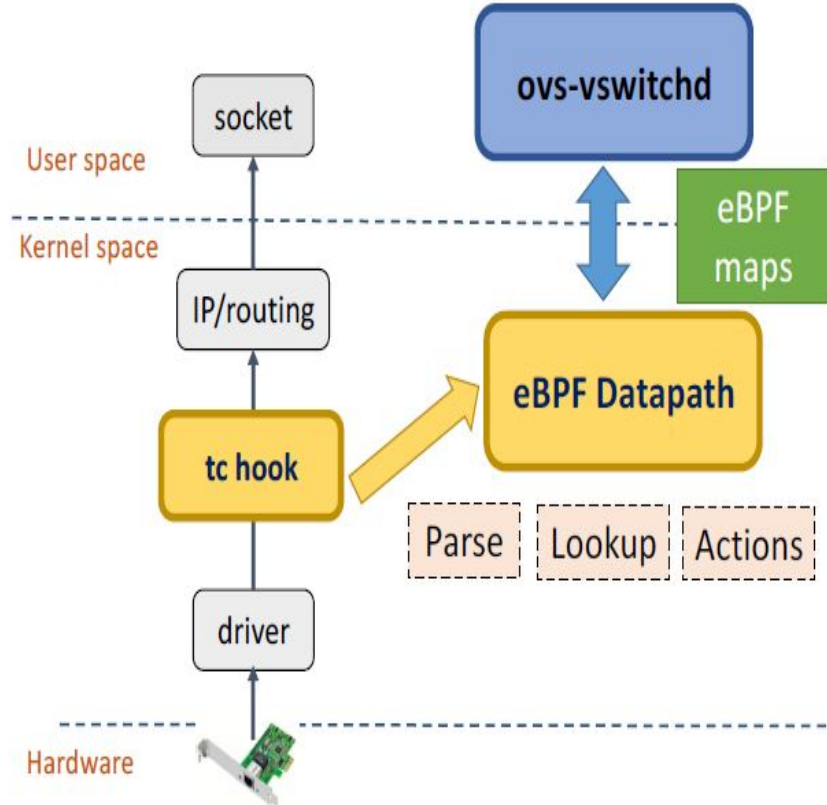
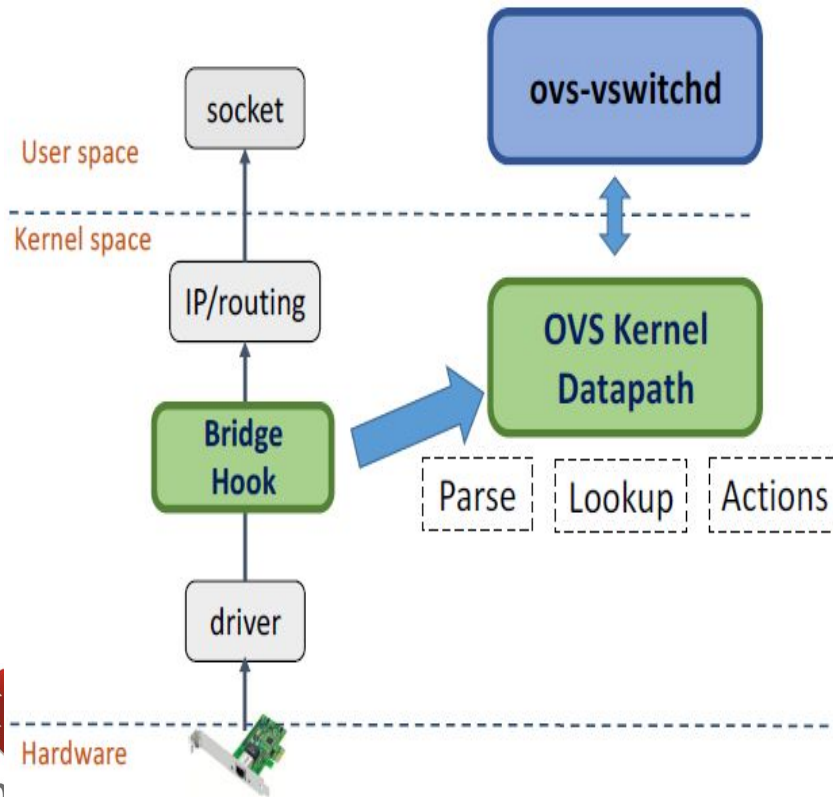
Multiple offshoots: IOVisor, XDP, Cilium



# Review: Open vSwitch Architecture

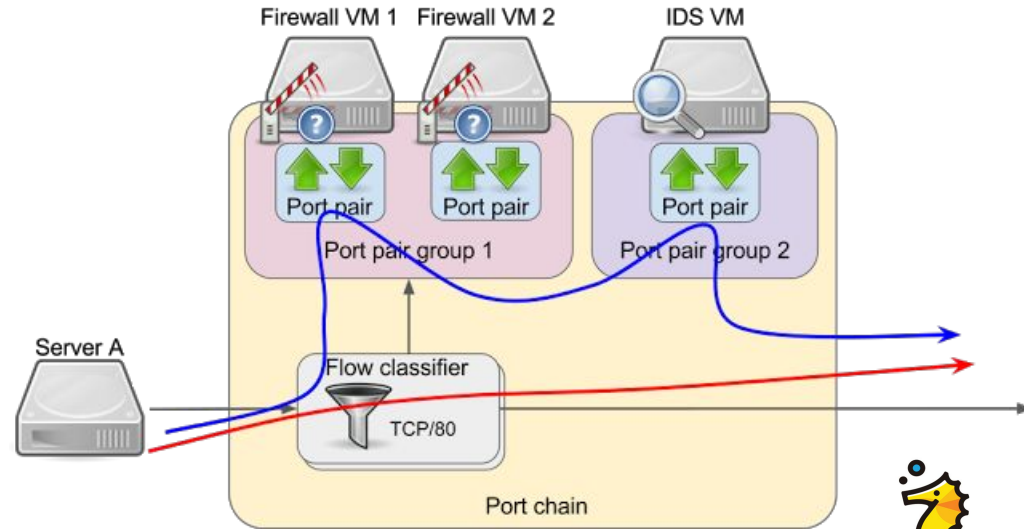
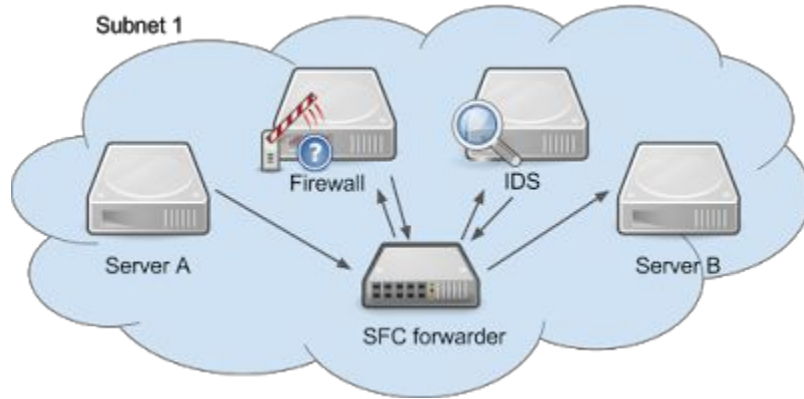


# eBPF - enhance OVS actions or replace datapath?



# Service function chaining (SFC)

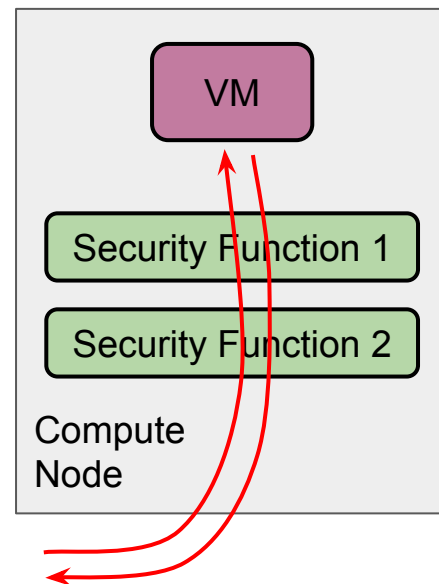
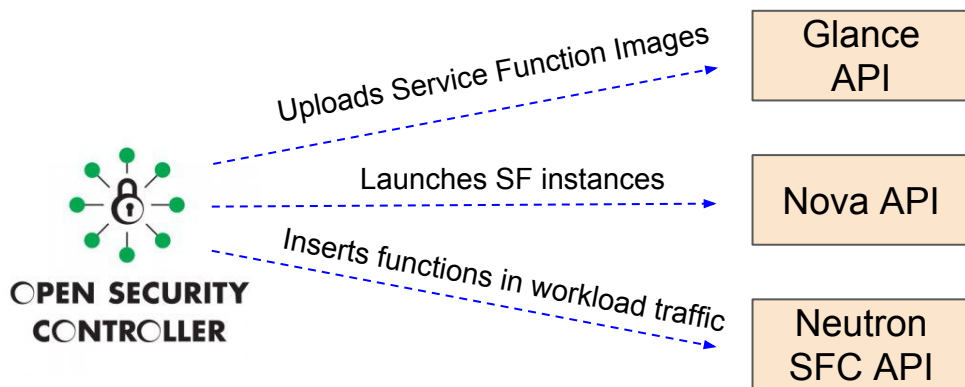
<http://www.dragonflow.net/2017/08/policy-based-routing-with-sfc-in.html>



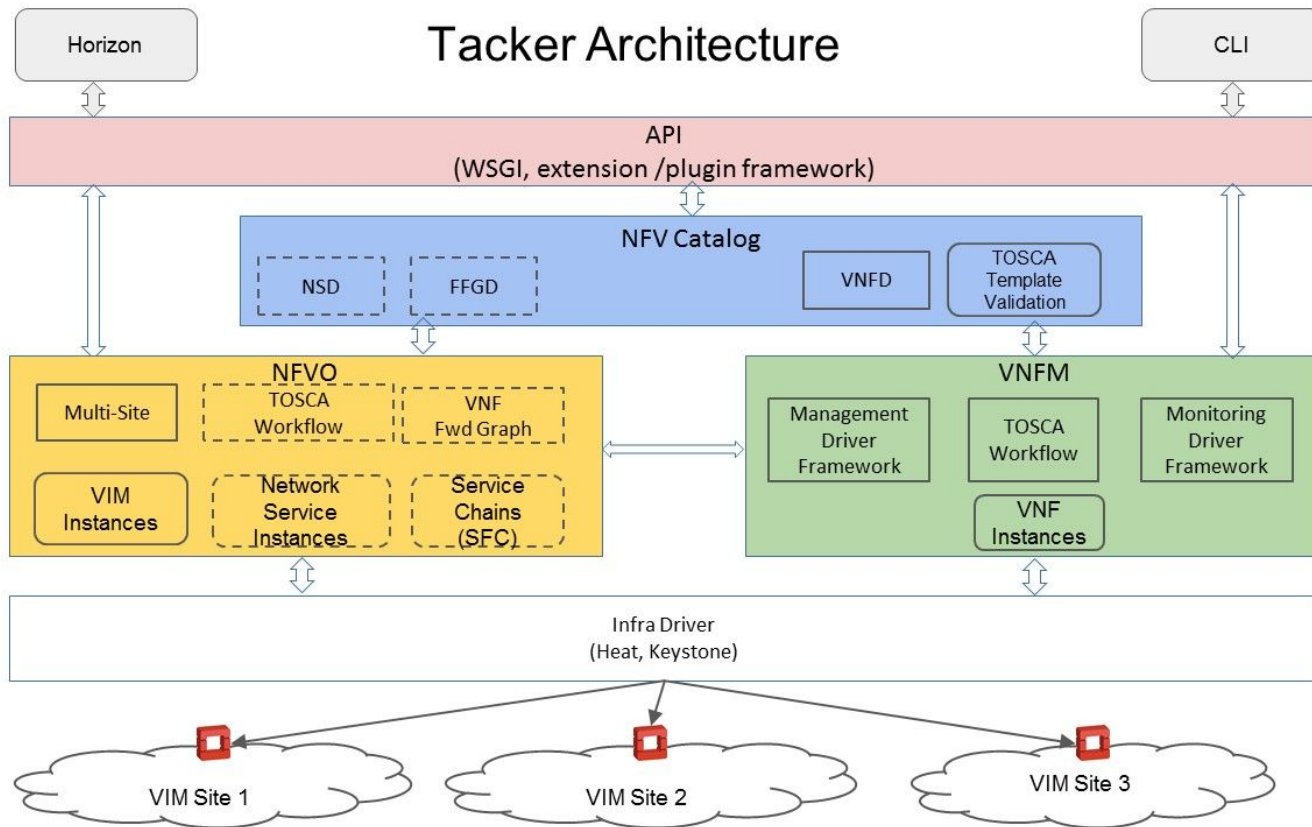
# Open Security Controller



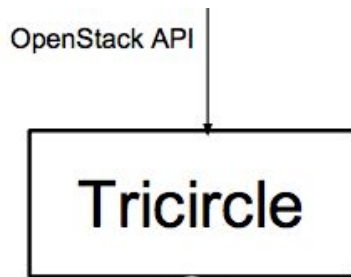
- Started by Intel, now part of Linux Foundation
- Enabled by SFC (Service Function Chaining) API
- Supports multiple security vendors, OpenStack, Kubernetes, VMWare



# Tacker - OpenStack NFV Orchestration

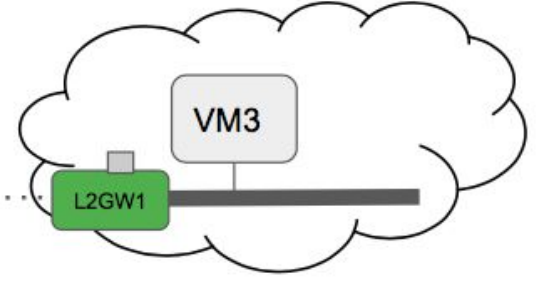
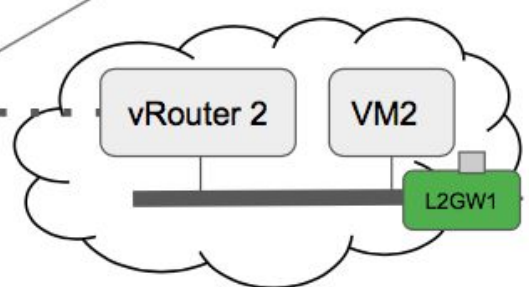
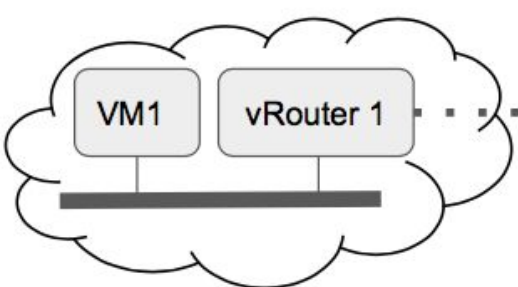
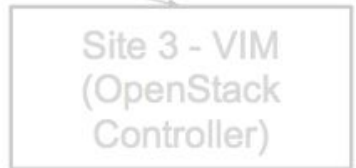
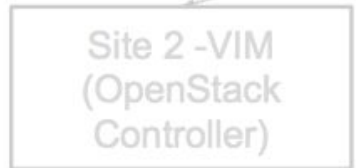


# TriCircle (Federation)

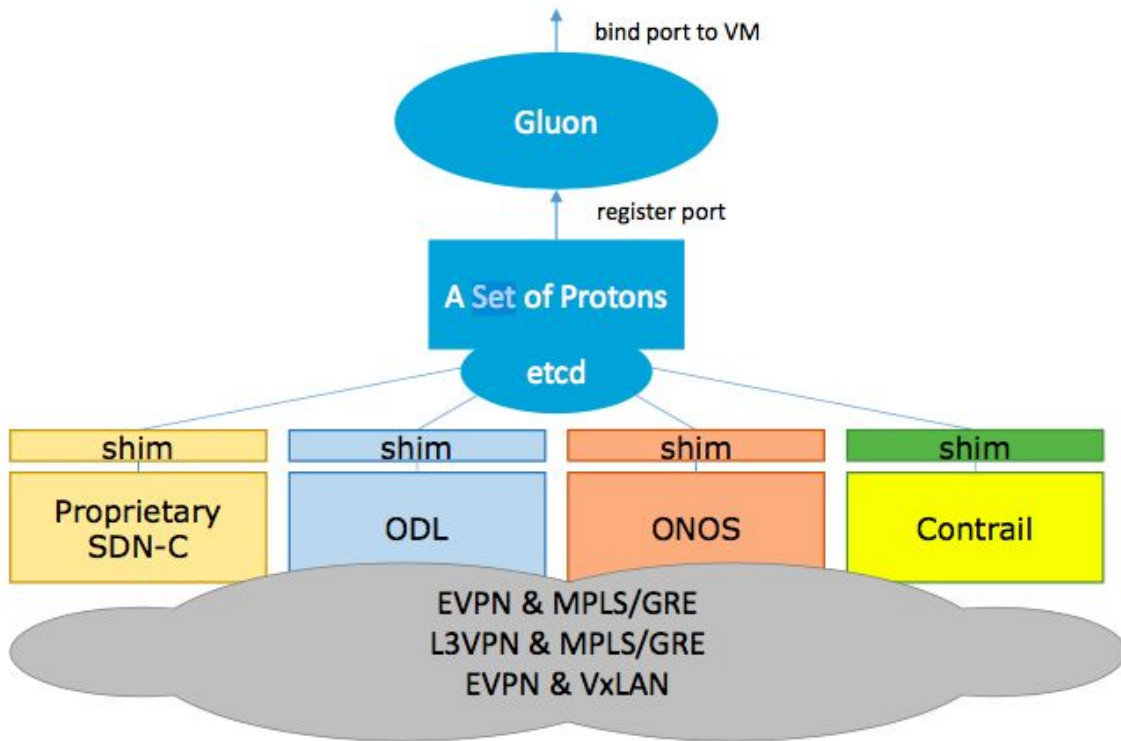


**OpenStack API Gateway:**

- Tenant level L2/L3 networking automation for E-W traffic isolation



# Glunon (when Neutron doesn't fit)



- 10,000 ft View

- Glunon – a Port Arbiter that

- Maintains a mapping of ports of different networking back-ends
    - Forwards port-related requests to the correct back-end

- Proton – a set of APIs of a particular NFV Networking Service (i.e. standard NBI)

- Proton Server – an API server that hosts multiple Protons

- Shim – the adaptor of Proton to native SDN-C API





# Dragonflow - Additional Resources

- Blogs: <http://www.dragonflow.net/>
- User documentation: <https://docs.openstack.org/dragonflow/latest/>
- Code: <https://github.com/openstack/dragonflow>
- Specs and Bugs on <https://launchpad.net/dragonflow>
- IRC channel #openstack-dragonflow (chat.freenode.net)
- Developers: <https://wiki.openstack.org/wiki/Dragonflow>



pino.decandia at huawei dot com  
giuseppe.decandia at gmail dot com  
@pino\_deca on Twitter

# Thanks!

## Time for Q&A



# Topics I didn't have time for...

- Get-me-a-network
- FWaaS v2
- Common classification framework
- QoS and bandwidth limiting
- DNS (Neutron internal and Designate)
- Octavia and LBaaS v2
- (IPSec) VPNaaS
- Routed Provider Networks
- Hierarchical port binding

