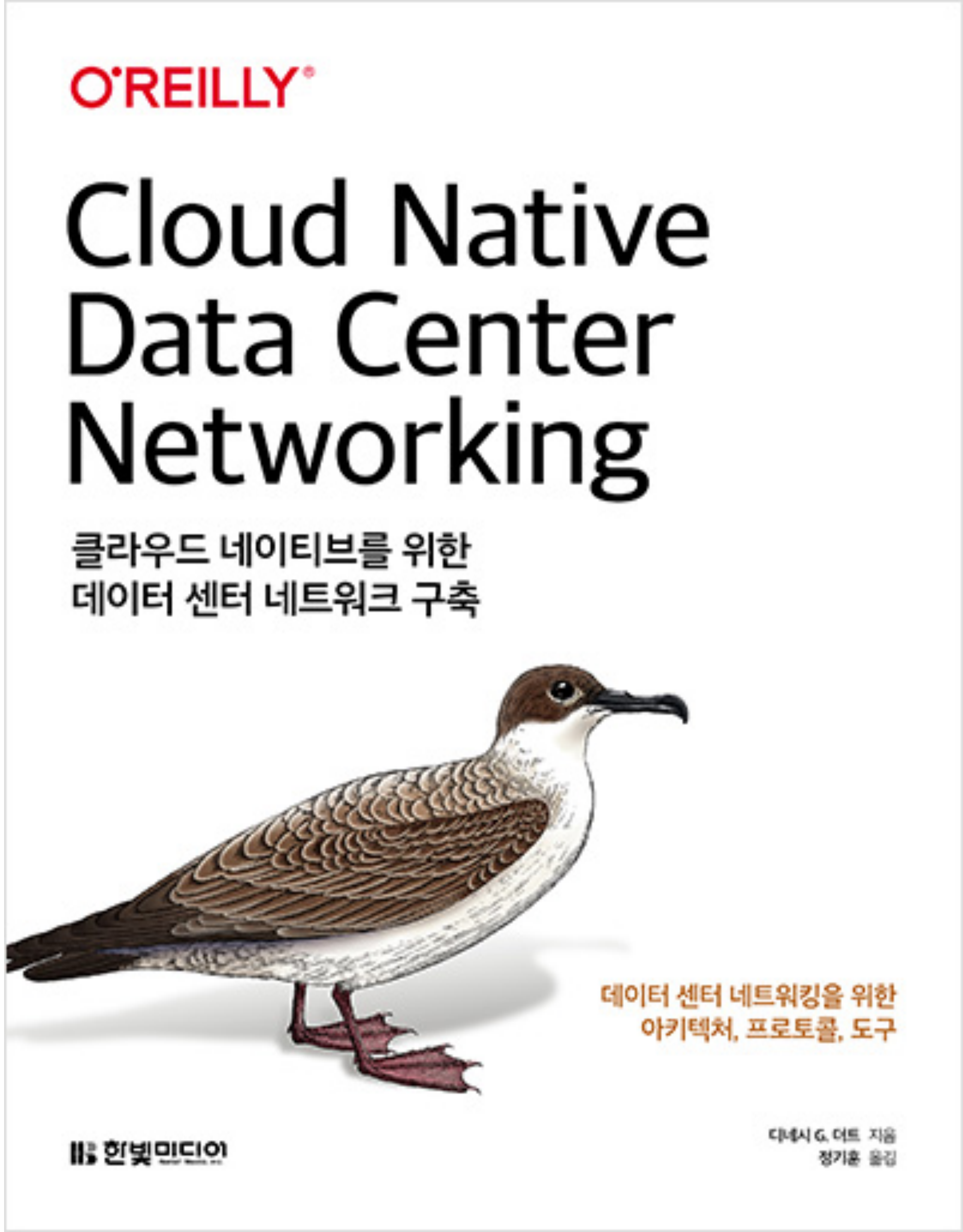
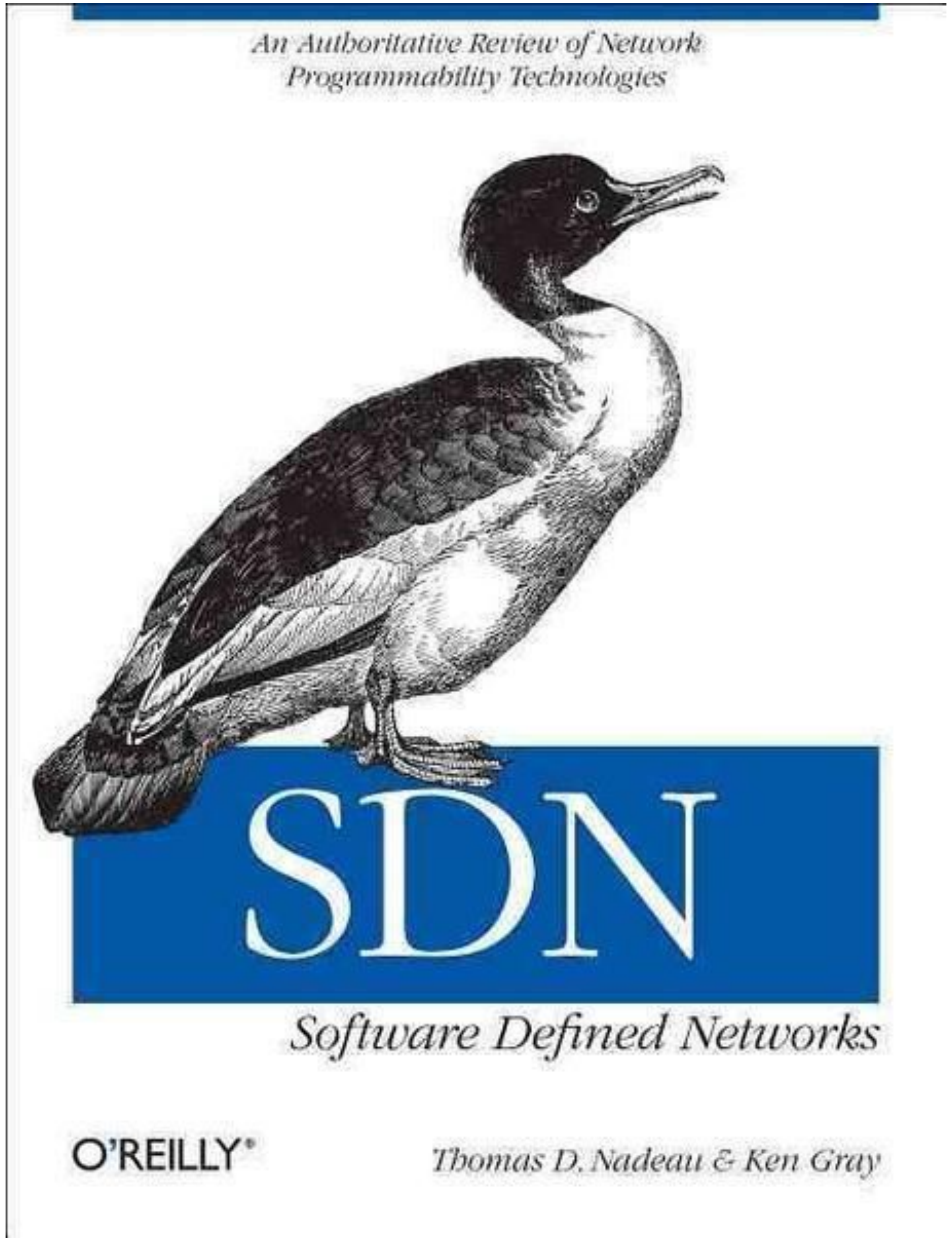
Open Virtual Network

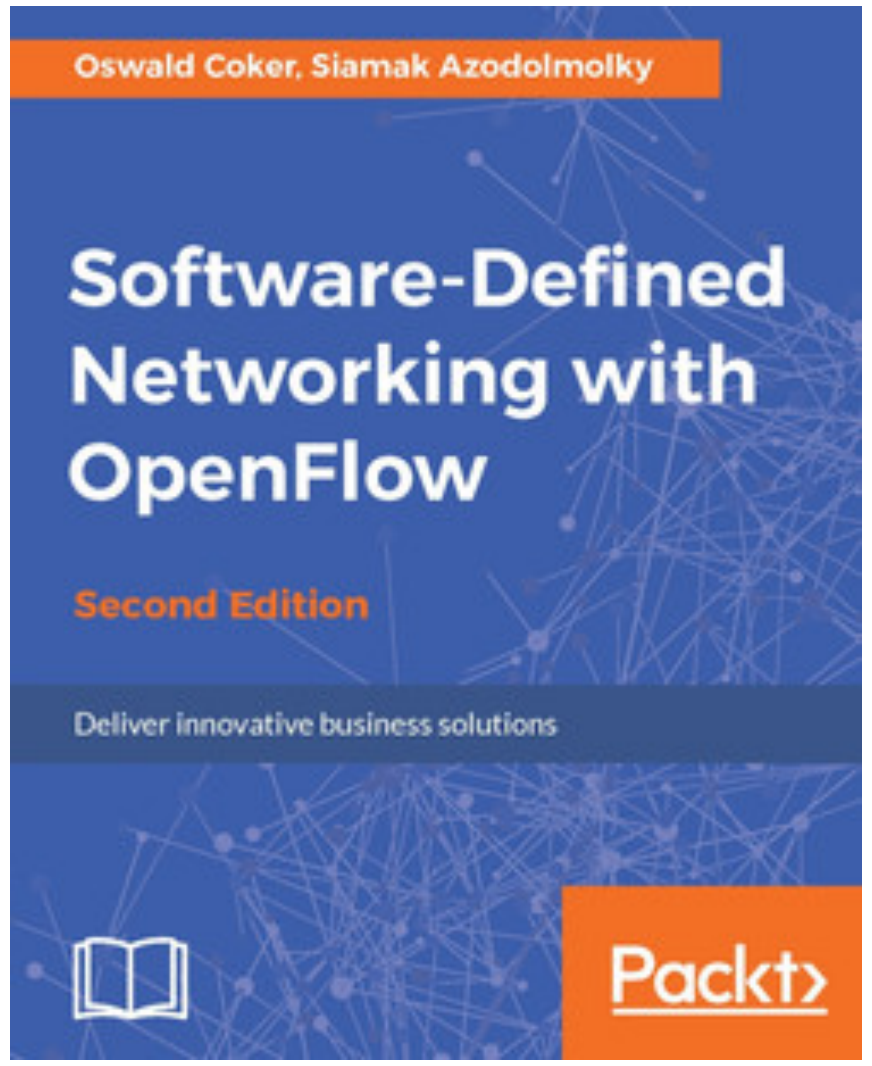
Software Defined Network

양성연 2025-05-25

추후 읽어볼 책들 ...


O'REILLY
Cloud Native Data Center Networking
클라우드 네이티브를 위한 데이터 센터 네트워크 구축
데이터 센터 네트워킹을 위한 아키텍처, 프로토콜, 도구
한빛미디어

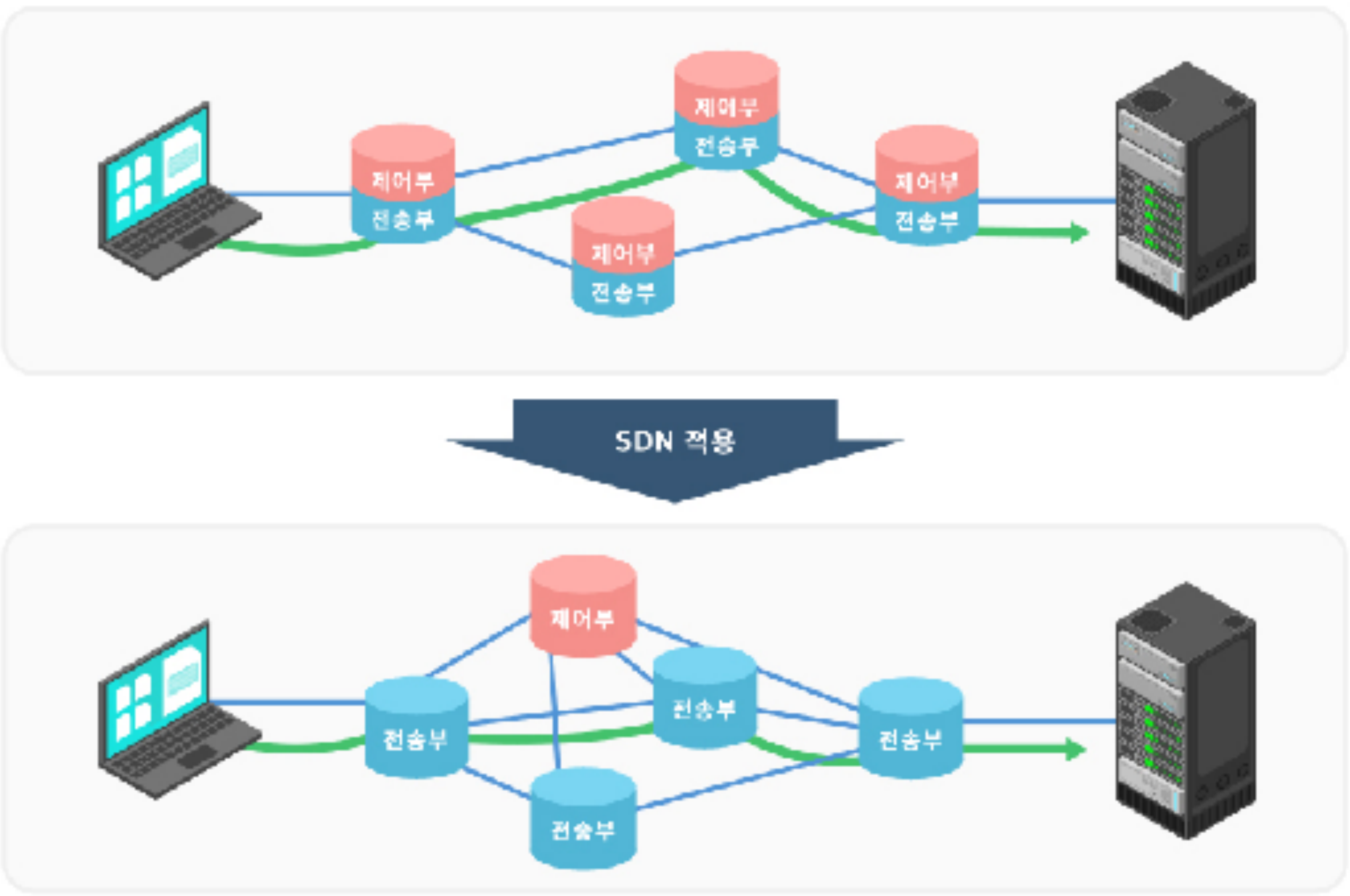

An Authoritative Review of Network Programmability Technologies
SDN
Software Defined Networks
O'REILLY
Thomas D. Nadeau & Ken Gray

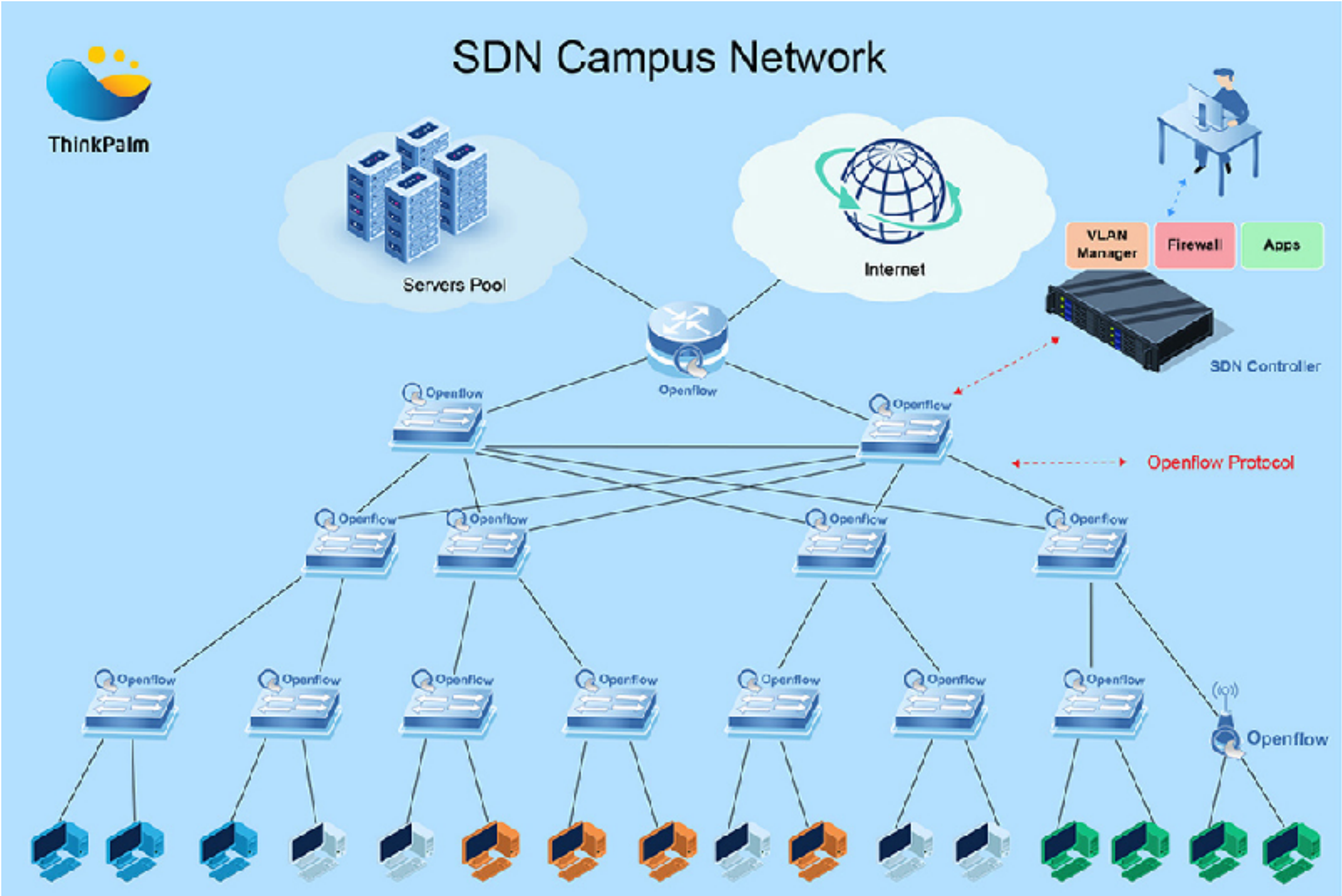

Oswald Coker, Siamak Azodolmolky
Software-Defined Networking with OpenFlow
Second Edition
Deliver innovative business solutions
Packt


Sriram Subramanian, Sreenivas Voruganti
Software-Defined Networking (SDN) with OpenStack
Leverage the best SDN technologies for your OpenStack-based cloud infrastructure
Packt

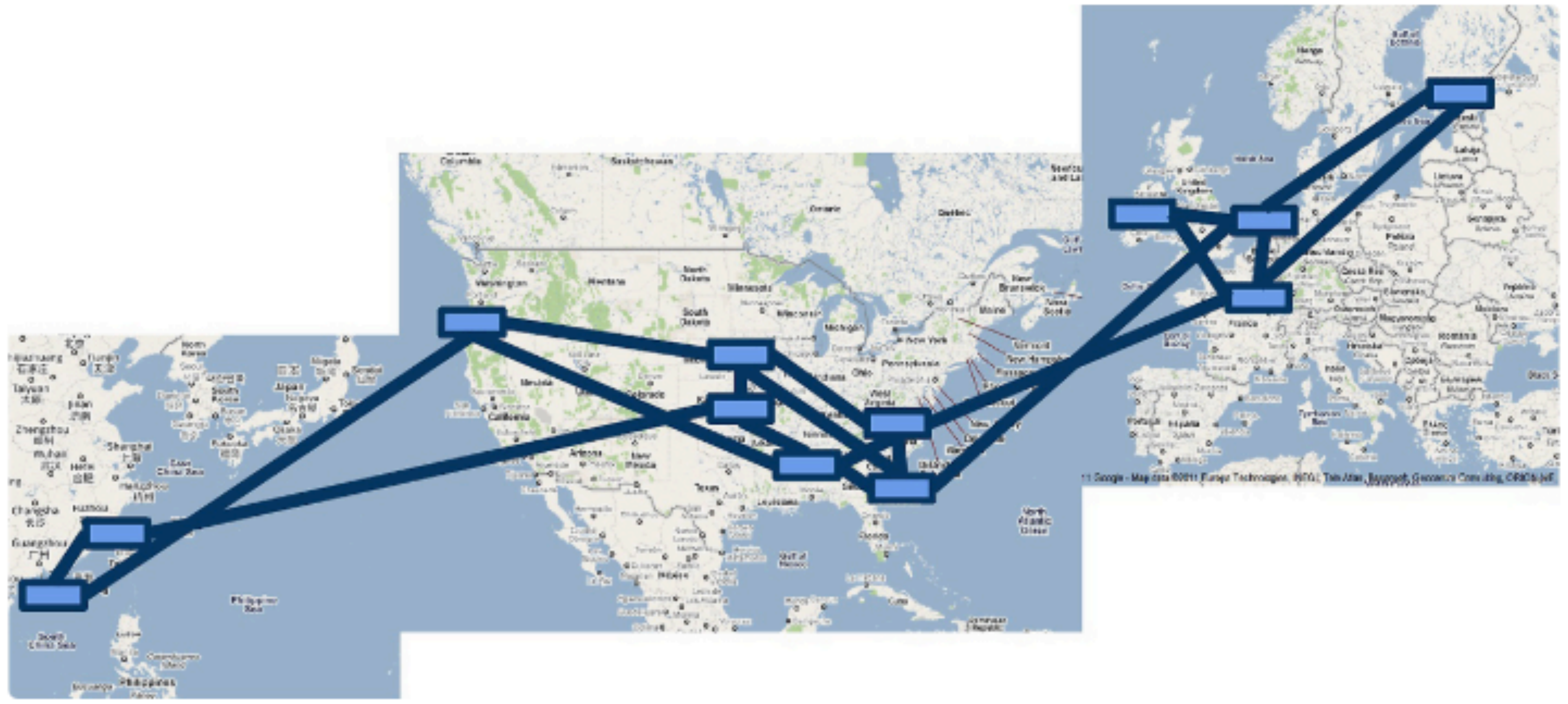
SDN (Software Defined Network)

SDN은 네트워크 장치의 제어부 (Control Plane)와 전송부 (Data Plane)를 분리하는 개념

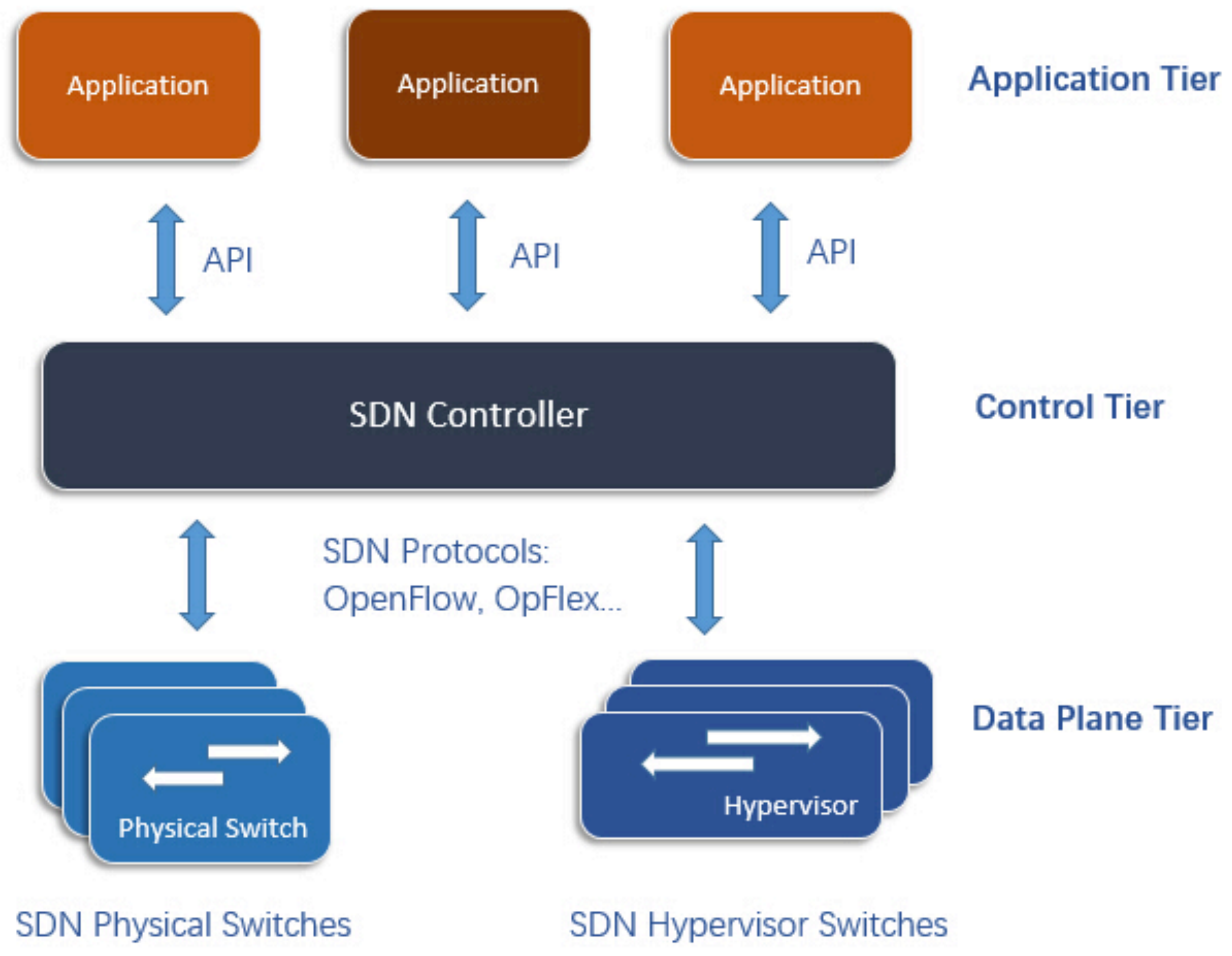




G-scale 2010년에 시작한 OpenFlow 프로젝트로 전 세계에 흩어져 있는 구글 데이터센터 백본 구간을 SDN 기반으로 전환하는 프로젝트



1. 50% -> 100% 리소스 활용률
2. WAN 구간에서의 경로 최적화
3. 비용 절감
   - SDN 컨트롤러와 화이트박스 스위치(기존 레거시 장비의 제어방식이 공개가 되지 않아 '블랙박스'라고 불린 것의 반대 개념)의 조합



출처 : Unveil the Myths About SDN Switch FS Community

SDN 컨트롤러 : 전체 네트워크 자원에 대한 중앙 집중적 제어를 담당하는 네트워크 운영체제 (NOS) 역할. 대표적으로 ONF 주도의 ONOS 가 존재

Data Plane : SDN 컨트롤러의 지시에 따라 패킷 전송을 수행하는 네트워크 장치를 통칭

Application Layer : Routing, Loadbalance, ACL 등 네트워크 제어하는 데 사용되는 소프트웨어 어플리케이션

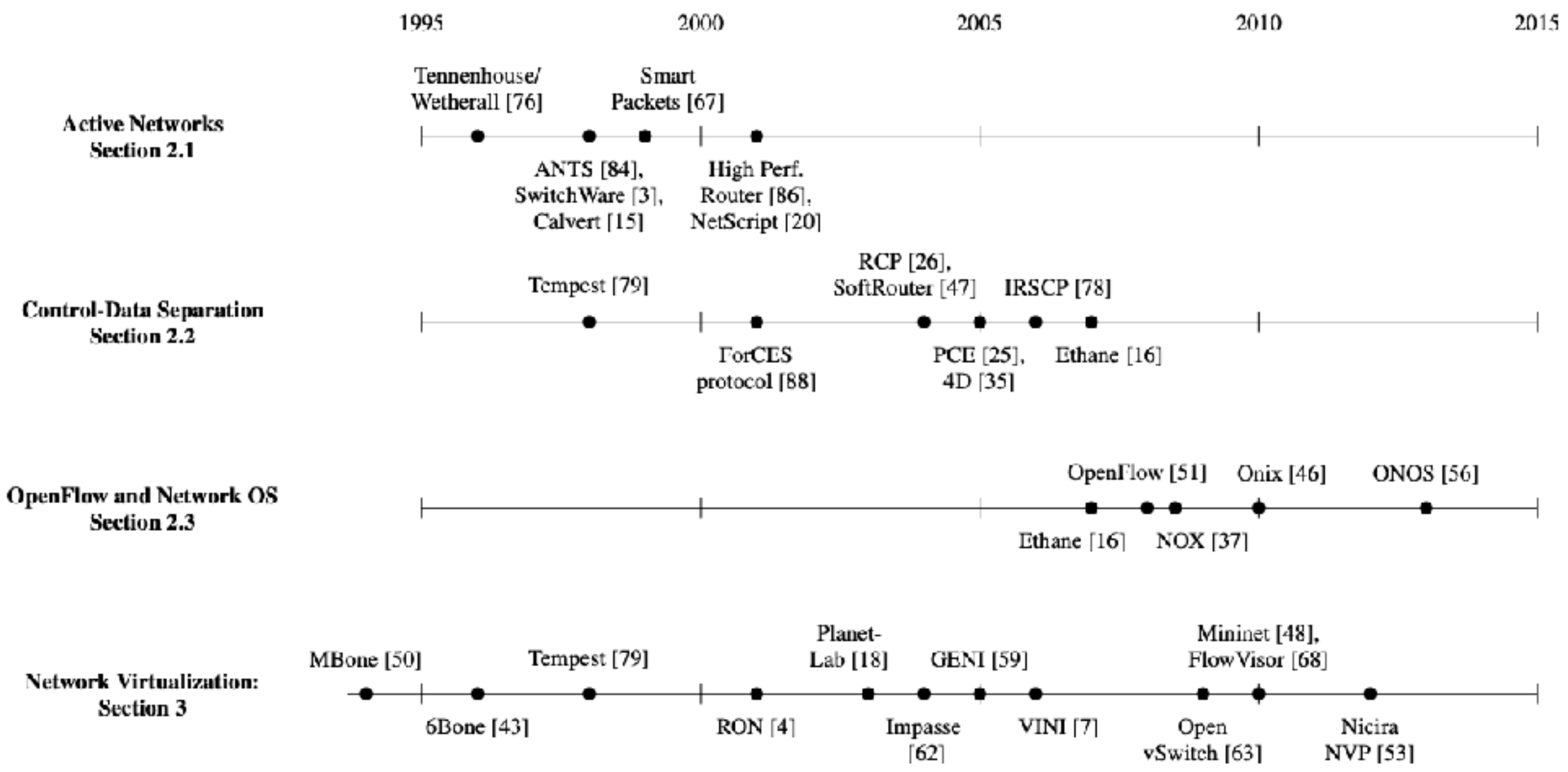
The Road to SDN: An Intellectual History of Programmable Networks

Nick Feamster
Georgia Tech
feamster@cc.gatech.edu

Jennifer Rexford
Princeton University
jrex@cs.princeton.edu

Ellen Zegura
Georgia Tech
ewz@cc.gatech.edu

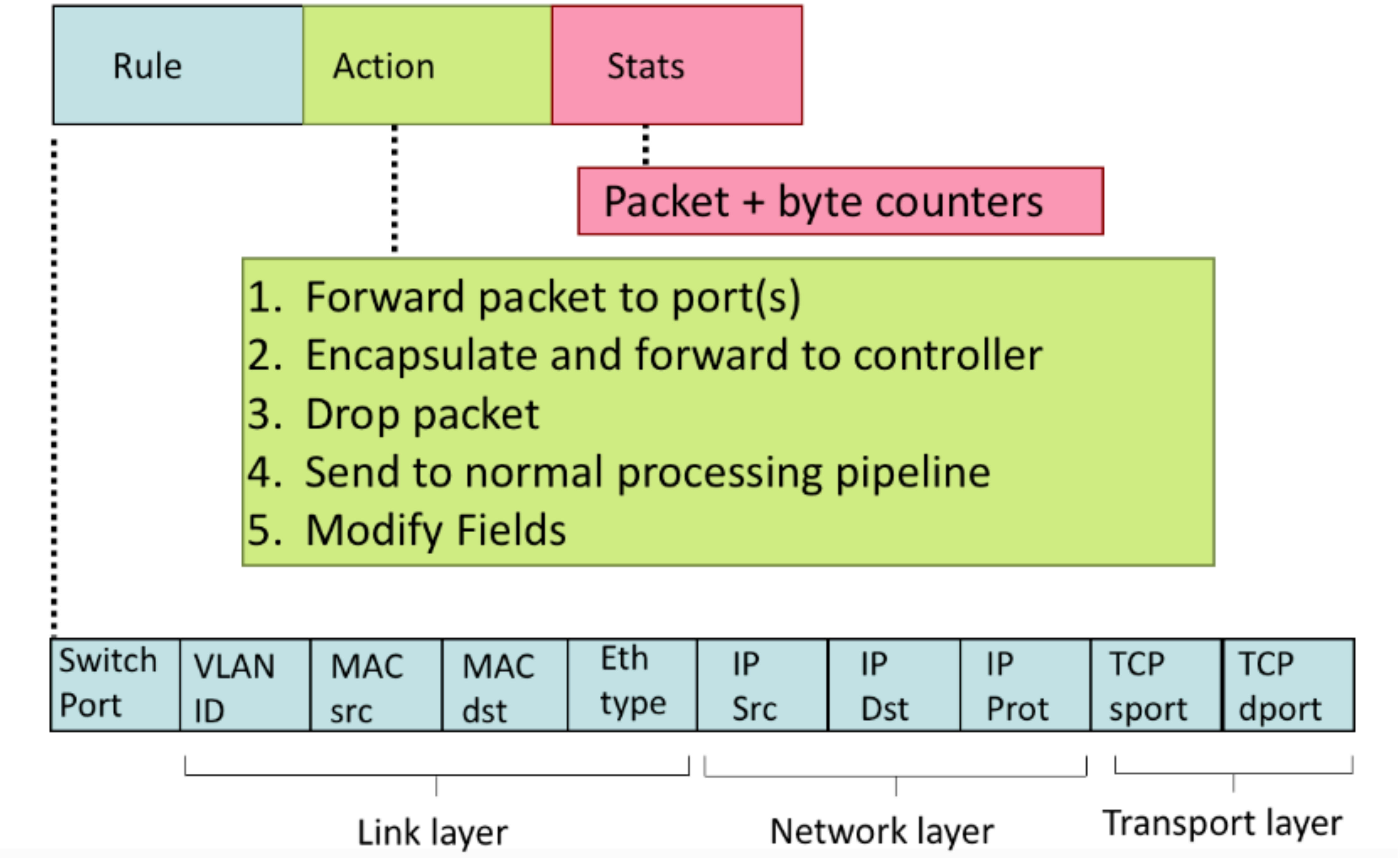
https://www.cs.princeton.edu/courses/archive/fall13/cos597E/papers/sdnhistory.pdf

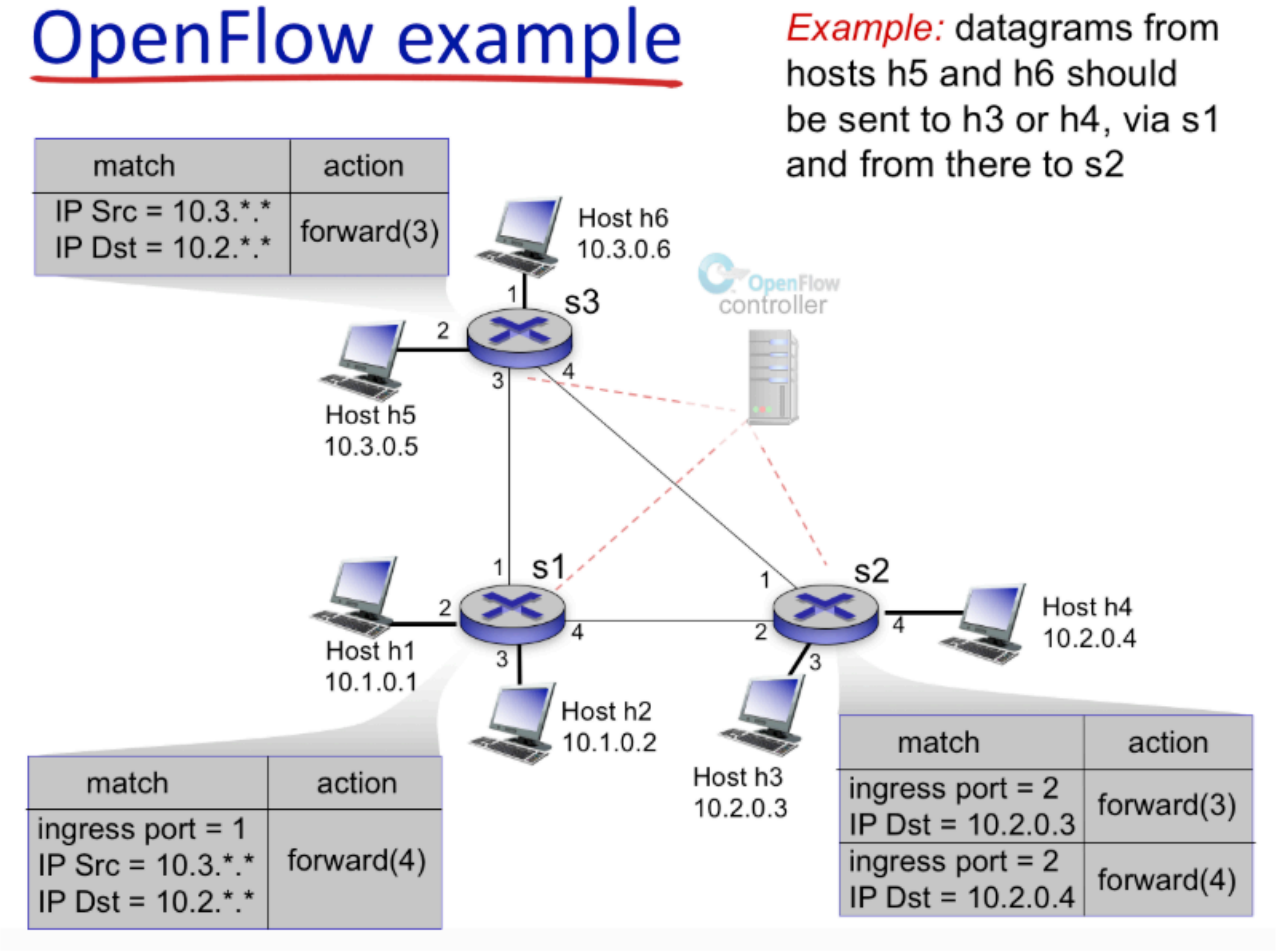


패킷을 프로그래밍 가능하게 하자는 초기 실험 연구들 -> 제어/데이터 평면을 분리하여 관리 효율화 시도 -> OpenFlow 기반 SDN + 이를 운영하는 네트워크 운영체제 -> 실 사용 예 ...

OpenFlow

OpenFlow Protocol is a multivendor standard defined by ONF for implementing SDN in networking equipment.





Destination-based forwarding:

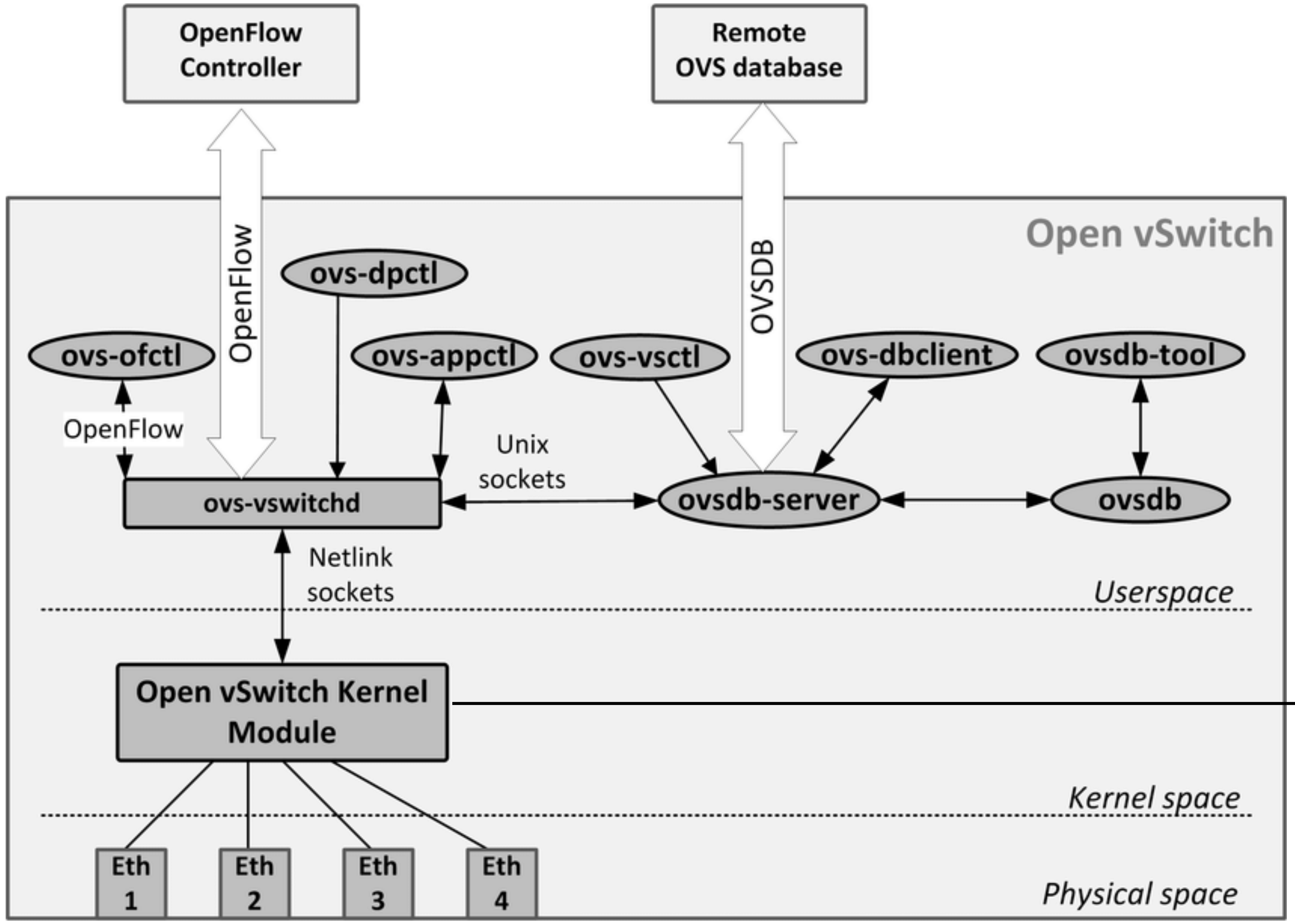
Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	51.6.0.8	*	*	*	port6

IP datagrams destined to IP address 51.6.0.8 should be forwarded to router output port 6

OpenFlow Switch Specification

https://opennetworking.org/wp-content/uploads/2013/04/openflow-spec-v1.0.0.pdf

Open vSwitch (https://github.com/openvswitch/ovs)



https://blog.naver.com/ww3w3/220063418331

- **ovsdb-client:** Open vSwitch database JSON-RPC client
- **ovsdb-tool:** Open vSwitch database management utility
- **ovs-ofctl:** a utility for querying and controlling OpenFlow switches and controllers
- **ovs-dpctl:** a tool for configuring the switch kernel module
- **ovs-vsctl:** a utility for querying and updating the configuration of ovs-vswitchd
- **ovs-appctl:** a utility that sends commands to running Open vSwitch daemons
- **ovs-pki:** a utility for creating and managing the public-key infrastructure for OpenFlow switches

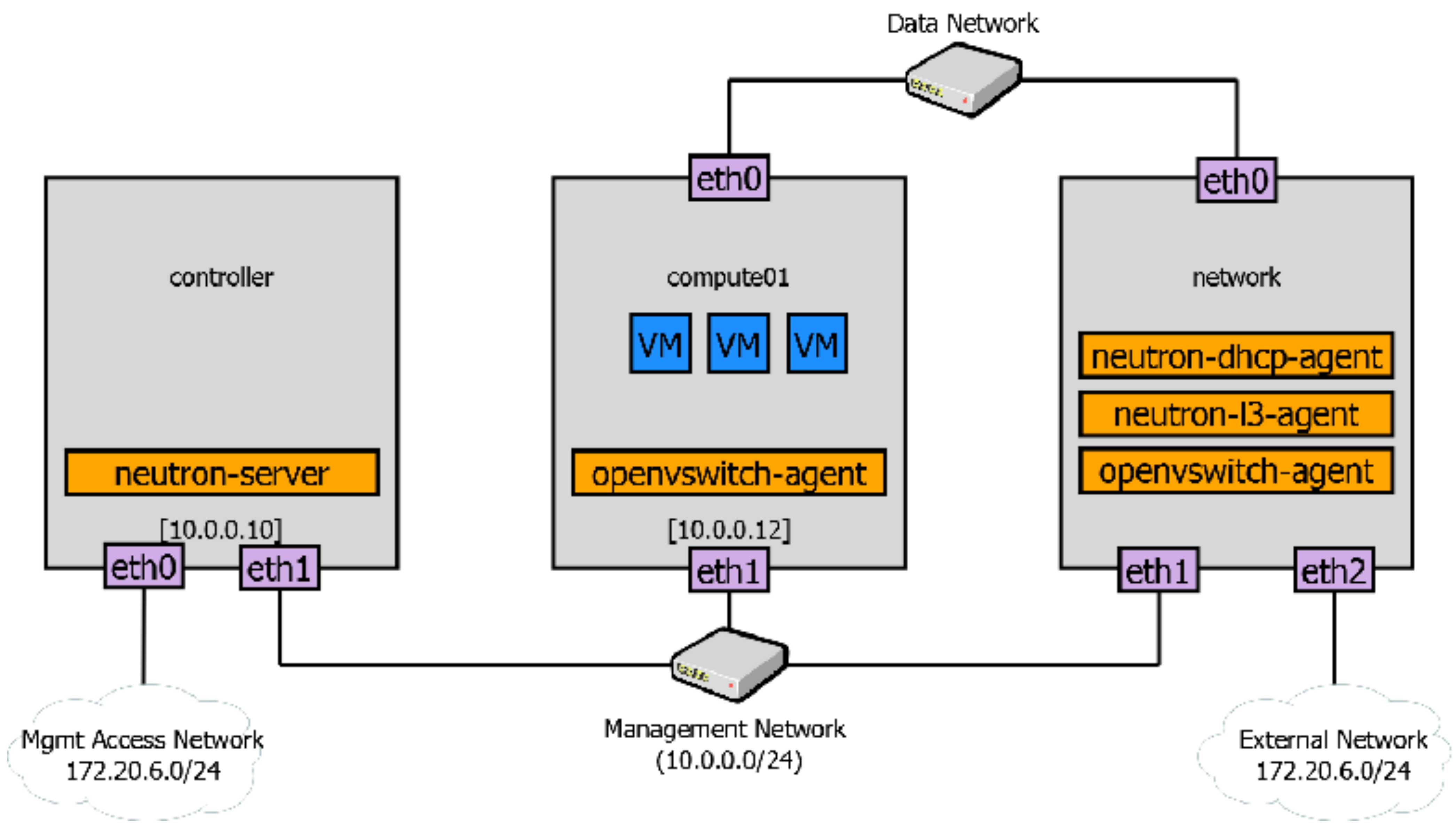
https://arthurchiao.art/blog/ovs-deep-dive-3-datapath/

```
/** struct datapath - datapath for flow-based packet switching */
struct datapath {
        struct rcu_head rcu;
        struct list_head list_node;

        struct flow_table table;
        struct hlist_head *ports; /* Switch ports. */
        struct dp_stats_percpu __percpu *stats_percpu;
        possible_net_t net; /* Network namespace ref. */

        u32 user_features;
        u32 max_headroom;
};
```

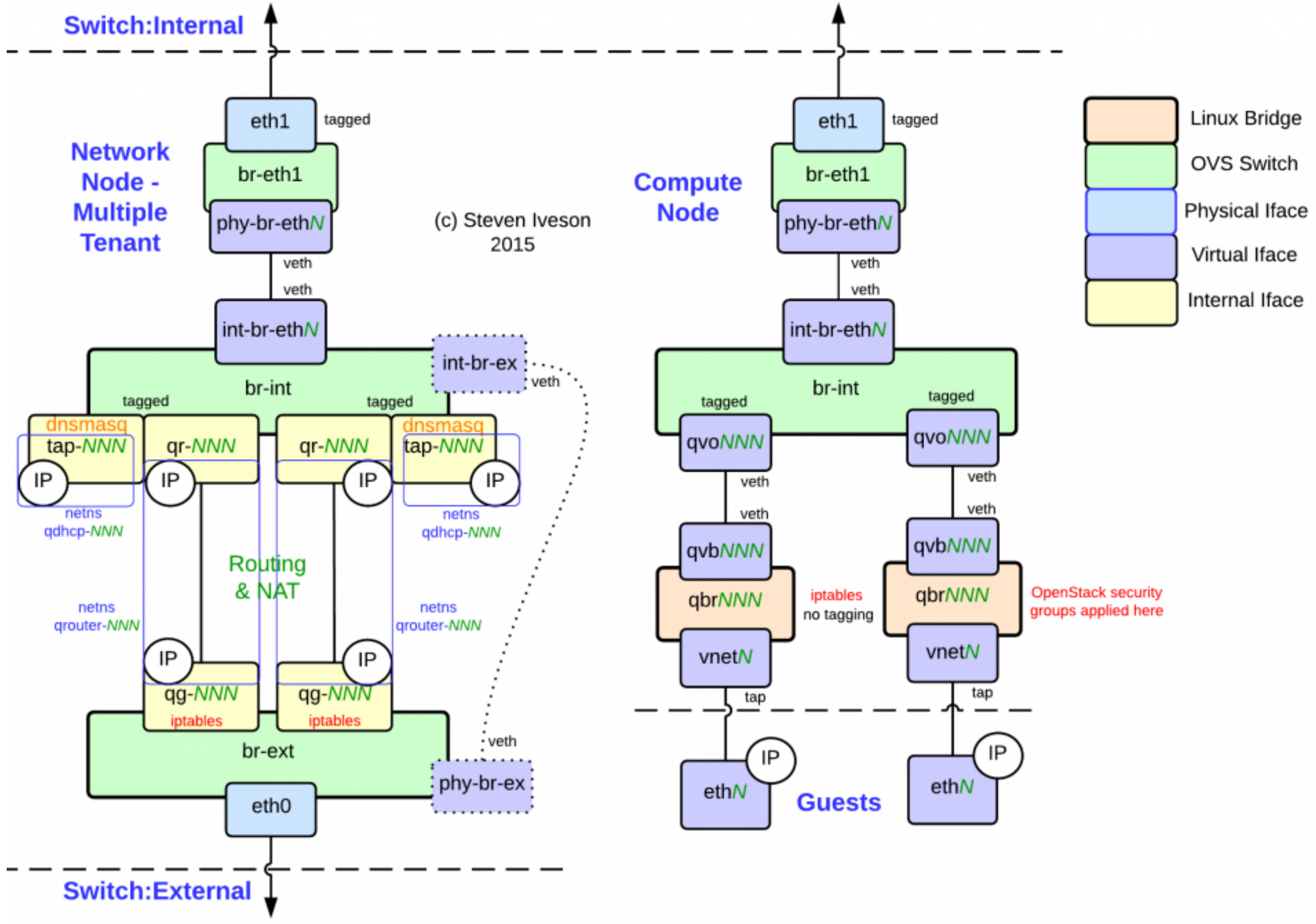
1) **ovs-vswitchd:** 커널과 netlink 로 통신을 하며 ovs의 주 업무를 담당한다.(controller와 openflow 통신, switch configuration 저장 및 변경(with ovsdb-server,-ovsdb))
2) **ovsdb-server:** OpenVSwitch의 데이터베이스 서버로, ovs-vswitchd로 부터 받은 정보들을 ovsdb에 작성하고 그 외에 components로 부터 들어온 정보들을 db에 써준다.
3) **ovs-brcompatd:** keeps the compatibility with the traditional bridges

OVS with Neutron

https://docs.openstack.org/neutron/zed/admin/deploy-ovs-provider.html
https://docs.openstack.org/neutron/latest/admin/deploy-ovs-selfservice.html




```
[ml2]
type_drivers = flat,vlan,vxlan
tenant_network_types = vxlan
mechanism_drivers = openvswitch,l2population
extension_drivers = port_security
```

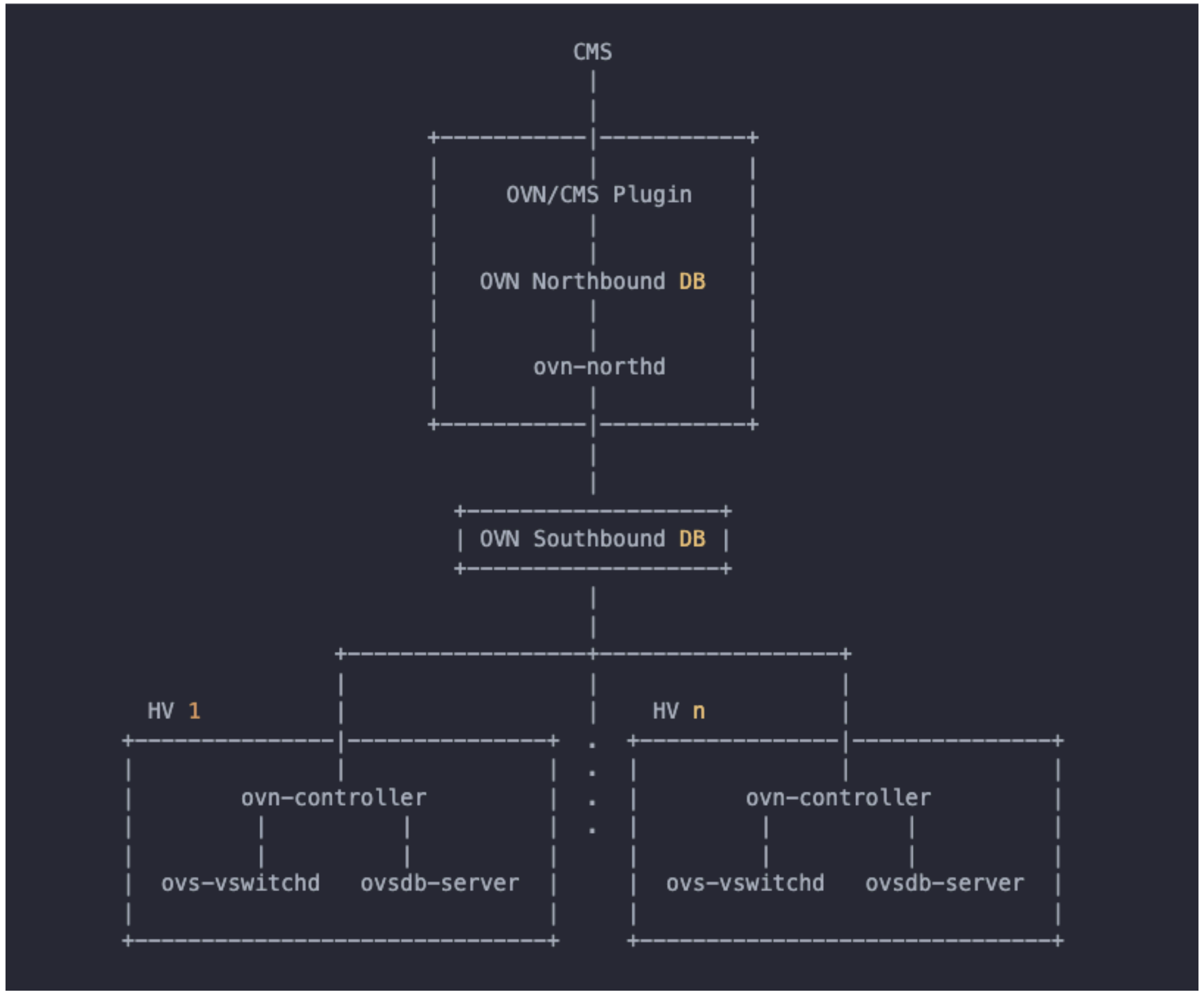


Modular Layer 2 (ML2) 플러그인은 OpenStack 네트워킹에서 현실 세계의 복잡한 데이터 센터에서 사용되는 다양한 L2(레이어 2) 네트워크 기술을 동시에 사용할 수 있도록 하는 프레임워크입니다.

Open Virtual Network (https://github.com/ovn-org/ovn)

OVN (Open Virtual Network) is a series of daemons for the Open vSwitch that translate virtual network configurations into OpenFlow.

OVN provides a higher-layer of abstraction than Open vSwitch, working with logical routers and logical switches, rather than flows. OVN is intended to be used by cloud management software (CMS). For details about the architecture of OVN, see the ovn-architecture manpage. Some high-level features offered by OVN include:



2015년 프로젝트 발표 관련 포스트

https://networkheresy.wordpress.com/category/network-virtualization/
https://docs.openstack.org/neutron/latest/admin/ovn/ovn.html

아키텍처

- Cloud Management System (CMS): integrates OVN into a physical network by managing the OVN logical network elements and connecting the OVN logical network infrastructure to physical network elements. Exmaples include OpenStack Neutron’s ml2/ovn plugin and the ovn-kubernetes project.

- OVN Databases: stores data representing the OVN logical and physical networks.

- Hypervisors: run Open vSwitch and translate the OVN logical network into OpenFlow on a physical or virtual machine.

- Gateways: extends a tunnel-based OVN logical network into a physical network by forwarding packets between tunnels and the physical network infrastructure.

An Example

Logical_Switch

Name	Ports
LS1	LP1,LP2

Logical_Port

Name	MAC
LP1	AA
LP2	BB

Chassis (ovn-controller)

Name	Encap	IP
HV1	Geneve	10.0.0.10
HV2	Geneve	10.0.0.11

Bindings (ovn-controller)

Name	Chassis
LP1	HV1

Pipeline (ovn-northd)

Datapath	Match	Action
LS1	eth.dst = AA	LP1
LS1	eth.dst = BB	LP2
LS1	eth.dst = <broadcast>	LP1,LP2

LP2 Arrives on HV2

Logical_Switch

Name	Ports
LS1	LP1,LP2

Logical_Port

Name	MAC
LP1	AA
LP2	BB

Chassis (ovn-controller)

Name	Encap	IP
HV1	Geneve	10.0.0.10
HV2	Geneve	10.0.0.11

Bindings (ovn-controller)

Name	Chassis
LP1	HV1
LP2	HV2

Pipeline (ovn-northd)

Datapath	Match	Action
LS1	eth.dst = AA	LP1
LS1	eth.dst = BB	LP2
LS1	eth.dst = <broadcast>	LP1,LP2

LP1, LP2는 각각의 MAC 주소를 가진 논리 포트.

OVN은 이 포트들이 어떤 노드에 있는지(Chassis)를 기억하고,

MAC 주소 기반으로 패킷이 어느 포트로 가야 할지를 판단하여

Overlay 터널(Geneve)을 통해 해당 호스트로 패킷을 보냄

OVN Database

- **ovn-northbound**
  - OpenStack/CMS integration point
  - High-level, desired state
    - Logical ports -> logical switches -> logical routers

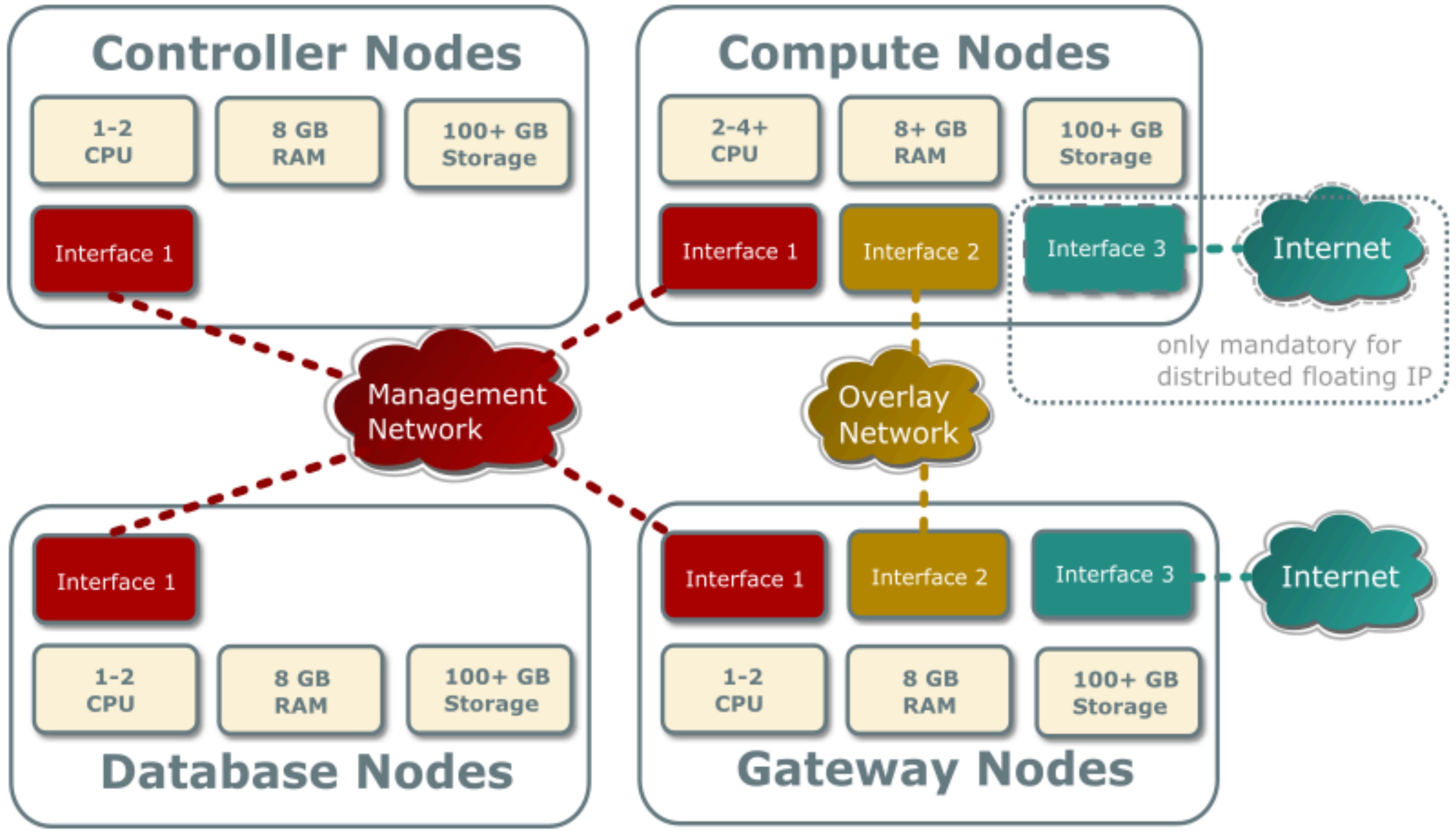
- **ovn-southbound**
  - Run-time state
    - Location of logical ports
    - Location of physical endpoints
    - Logical pipeline generated based on configured and run-time state

OVN Daemons

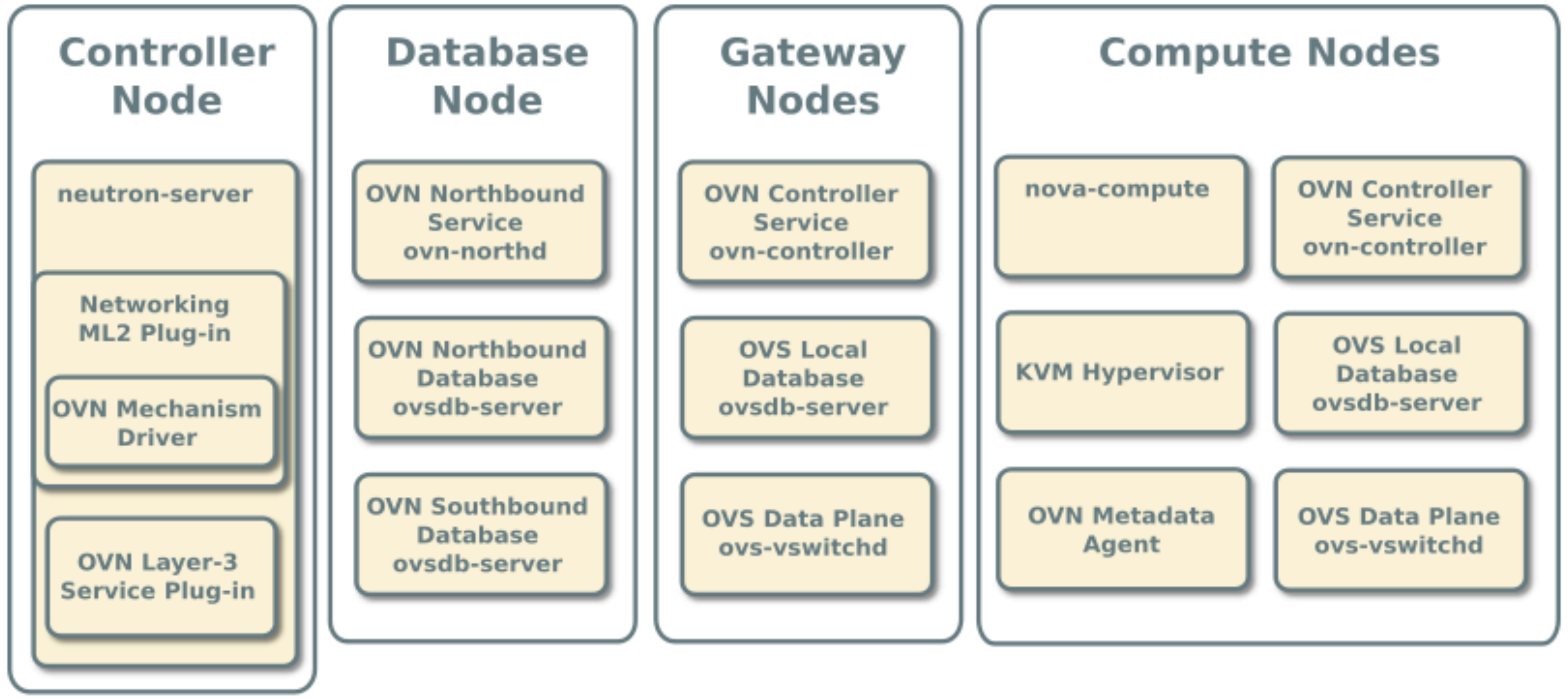
- **ovn-northd**
  - Converts from the high-level northbound DB to the run-time southbound DB
  - Generates logical flows based on high-level configuration
- **ovn-controller**
  - Registers chassis and VIFs to southbound DB
  - Converts logical flows into physical flows (ie, VIF UUIDs to OpenFlow ports)
  - Pushes physical configuration to local OVS instance through OVSDB and OpenFlow

OVN with Neutron

Hardware layout

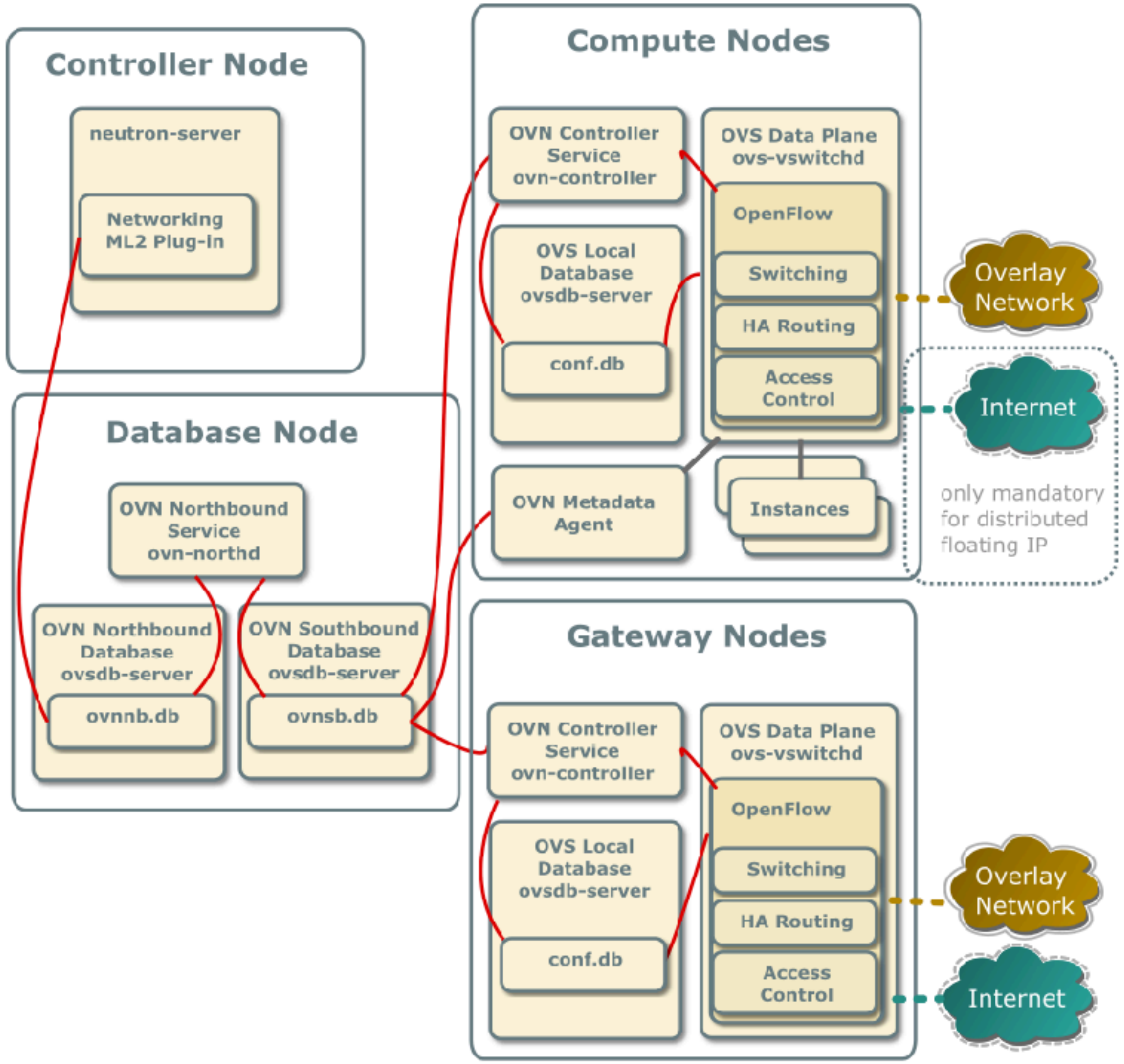


Service layout



Networking service with OVN integration

The reference architecture deploys the Networking service with OVN integration as described in the following scenarios:



OVN vs OVS

Q: What are the key differences between ML2/ovs and ML2/ovn?

Detail	ml2/ovs	ml2/ovn
agent/server communication	rabbit mq messaging + RPC.	ovsdb protocol on the NorthBound and SouthBound databases.
l3ha API	routers expose an "ha" field that can be disabled or enabled by admin with a deployment default.	routers don't expose an "ha" field, and will make use of HA as soon as there is more than one network node available.
l3ha dataplane	qrouter namespace with keepalive process and an internal ha network for VRRP traffic.	ovn-controller configures specific OpenFlow rules, and enables BFD protocol over tunnel endpoints to detect connectivity issues to nodes.
DVR API	exposes the "distributed" flag on routers only modifiable by admin.	exposes the "distributed" flag based on the configuration option enable_distributed_floating_ip
DVR dataplane	uses namespaces, veths, ip routing, ip rules and iptables on the compute nodes.	Uses OpenFlow rules on the compute nodes.
E/W traffic	goes through network nodes when the router is not distributed (DVR).	completely distributed in all cases.
Metadata Service	Metadata service is provided by the qrouters or dhcp namespaces in the network nodes.	Metadata is completely distributed across compute nodes, and served from the ovnmeta-xxxxx-xxxx namespace.
DHCP Service	DHCP is provided via qdhcp-xxxxx-xxx namespaces which run dnsmasq inside.	DHCP is provided by OpenFlow and ovn-controller, being distributed across computes.
Trunk Ports	Trunk ports are built by creating br-trunk-xxx bridges and patch ports.	Trunk ports live in br-int as OpenFlow rules, while subports are directly attached to br-int.

성능 관련 게시글

1. https://thesaitech.wordpress.com/2019/02/15/a-comparative-study-of-openstack-networking-architectures/
2. https://blog.russellbryant.net/post/2016/12/2016-12-19-comparing-openstack-neutron-ml2ovs-and-ovn-control-plane/
3. https://developers.redhat.com/blog/2019/01/02/performance-improvements-in-ovn-past-and-future

왜 Geneve 인가?

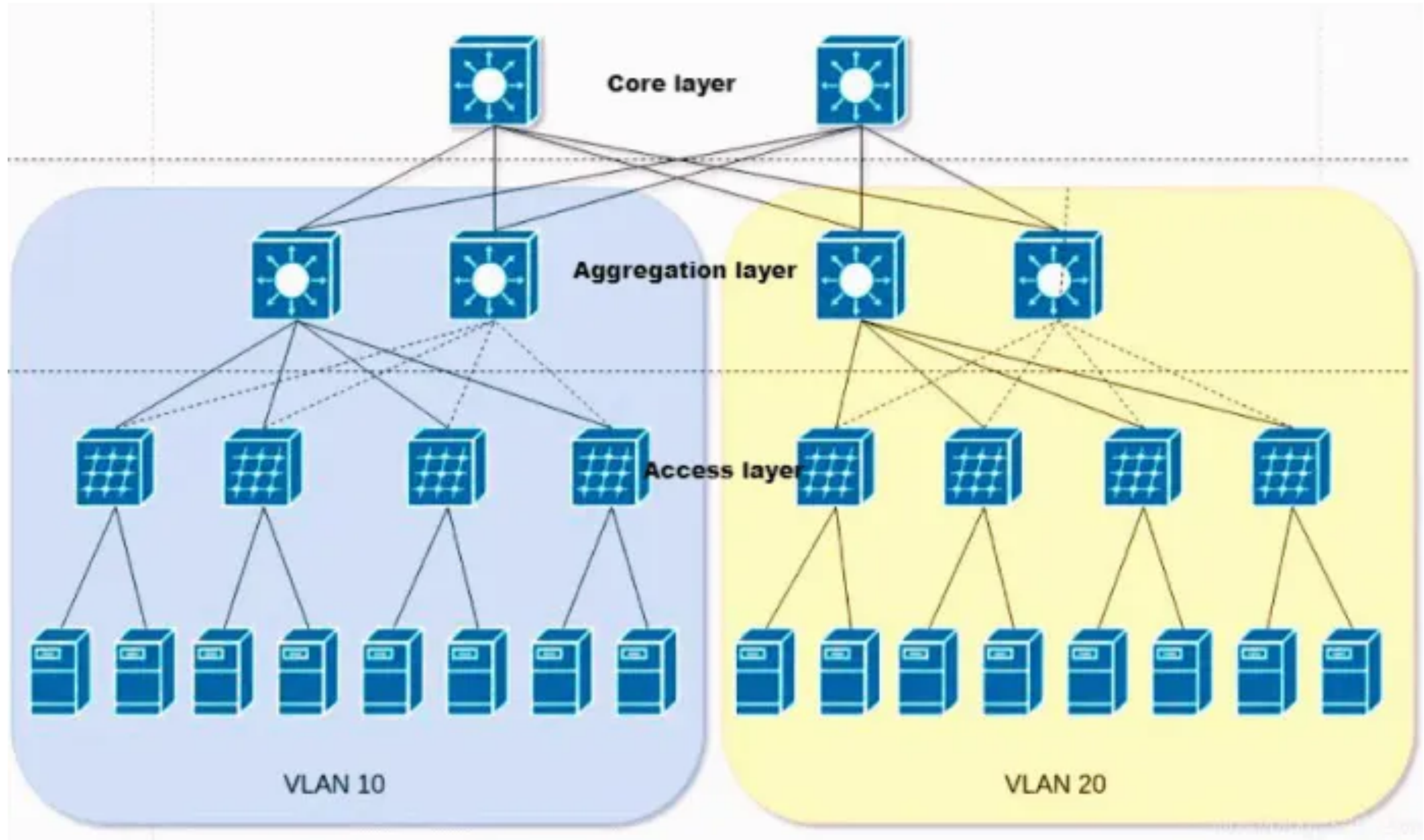
https://blog.russellbryant.net/post/2017/05/2017-05-30-ovn-geneve-vs-vxlan-does-it-matter/

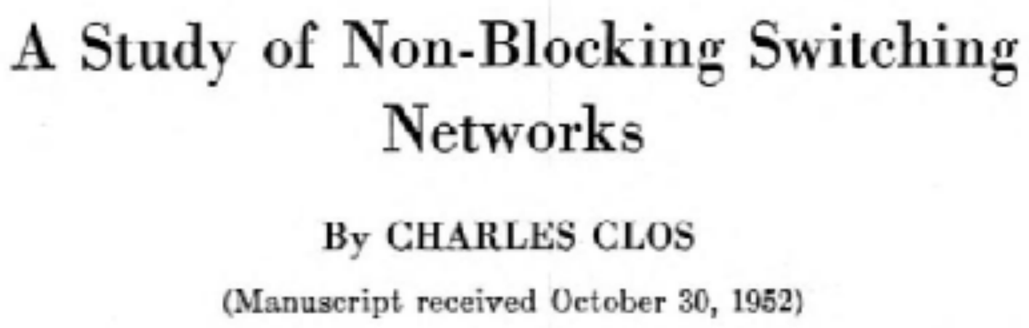
Why use Leaf-Spine Architecture ?

https://medium.com/@Asterfusion/what-is-leaf-spine-architecture-850952bcf7d0

기존 3-Tier (Core-Aggregation-Access) 구조의 한계

- Aggregation 스위치는 트래픽 병목을 유발하며 수직 확장 필요
- 고정 경로 기반 구조로 동적 부하 분산이 어려움
- Access-Aggregation 간의 L2 스패닝트리 필요 → 느림 & 비효율
- 상위 계층 스위치에서 복잡한 VLAN 트렁킹 필요
- 전용 벤더 장비 (Core, Agg) 스케일/비용 문제



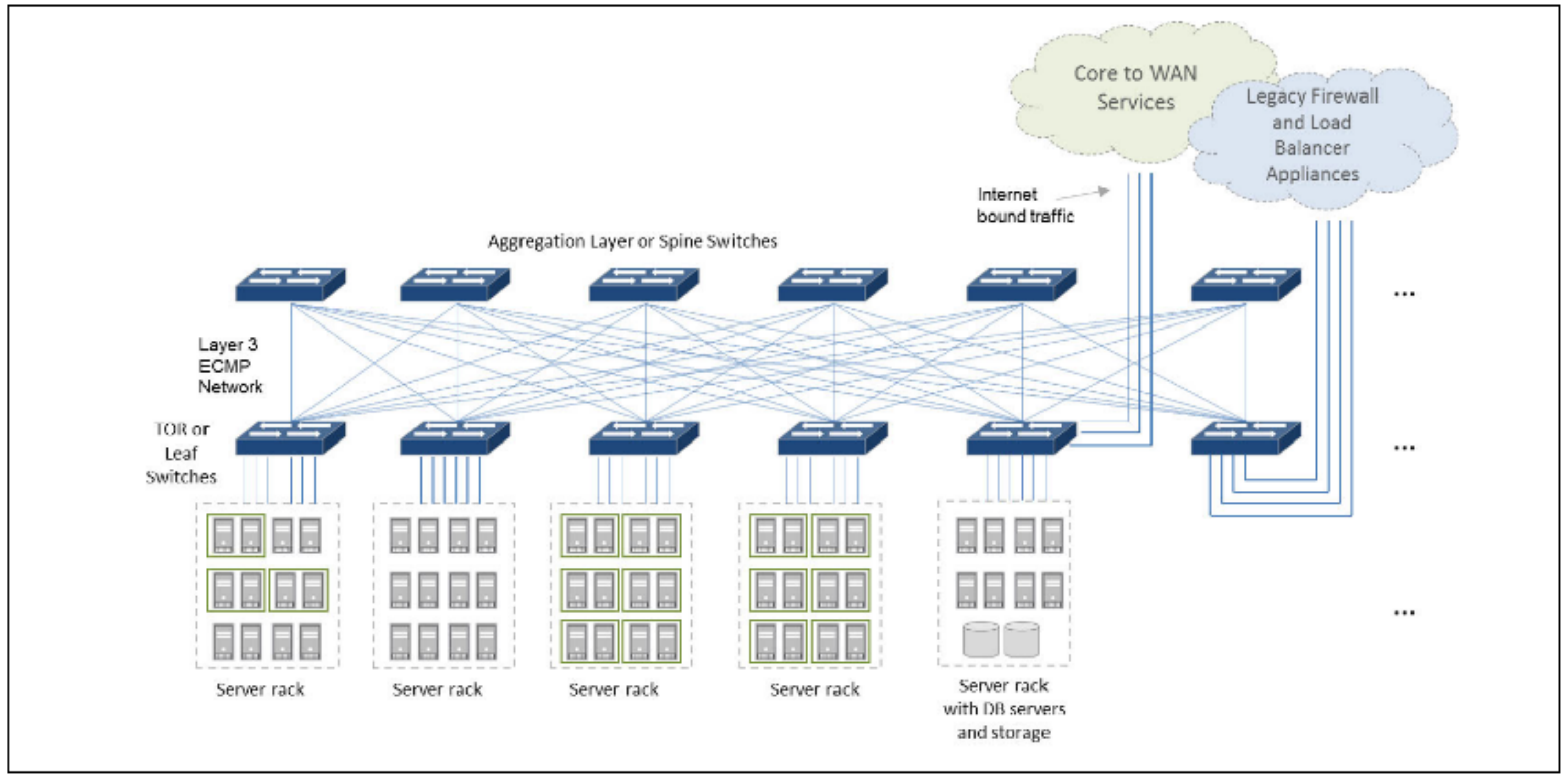

A Study of Non-Blocking Switching Networks

By CHARLES CLOS

(Manuscript received October 30, 1952)

Leaf-Spine 구조 (Clos Topology) 도입

- Leaf와 Spine만 추가하면 대규모 트래픽 처리 가능
- 모든 Leaf–Spine 간 연결이 동등한 비용 → BGP 기반 부하 분산
- 모든 Leaf ↔ Spine 간 연결이 평등하여 예측 가능한 지연
- L2 브로드캐스트 도메인 최소화 + SDN 오버레이와 궁합이 좋음
- BGP, EVPN, VLAN-Free 구조로 컨트롤 플레인 자동화 용이
- 일반적인 whitebox 스위치 사용 가능 (Cumulus, SONiC 등)

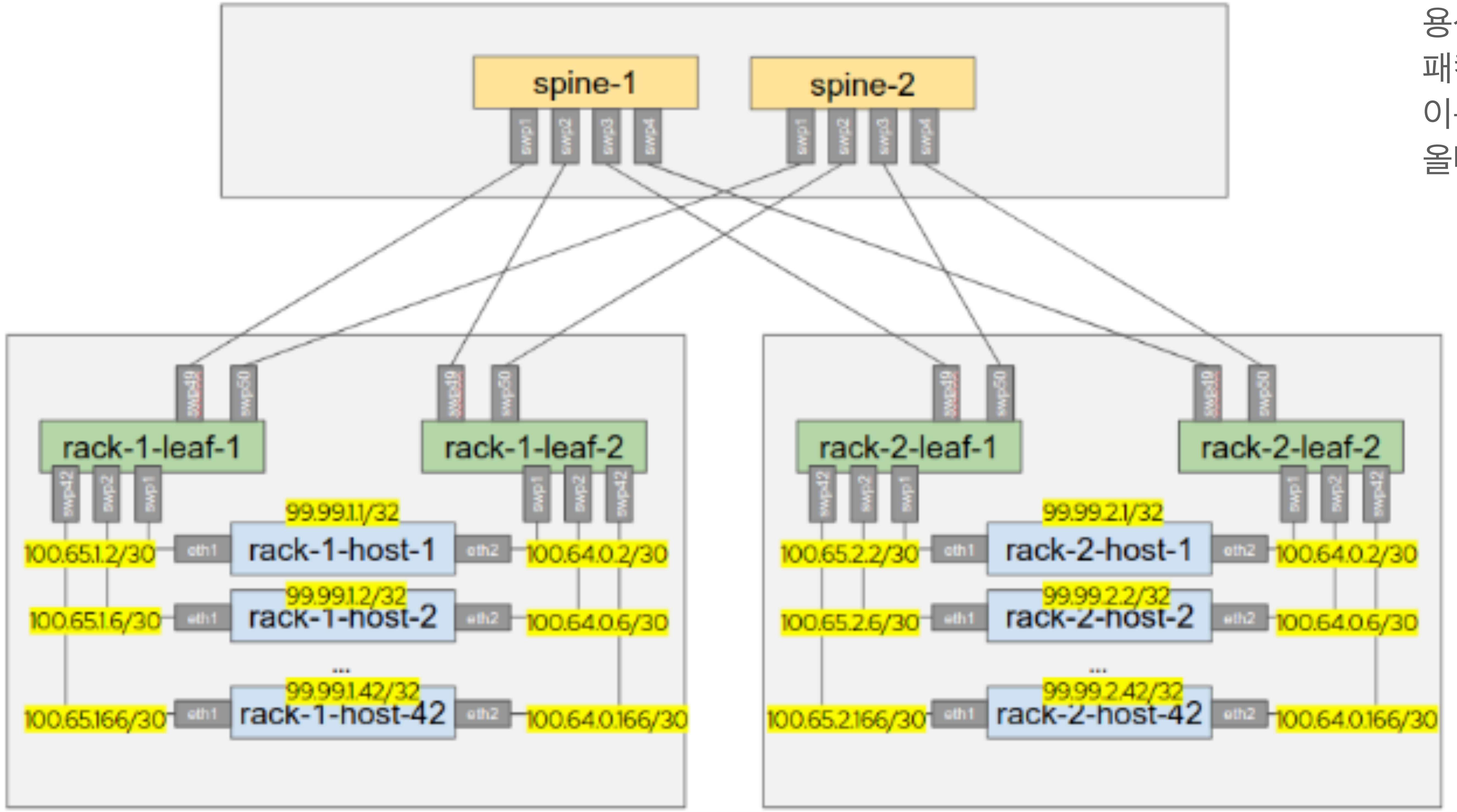


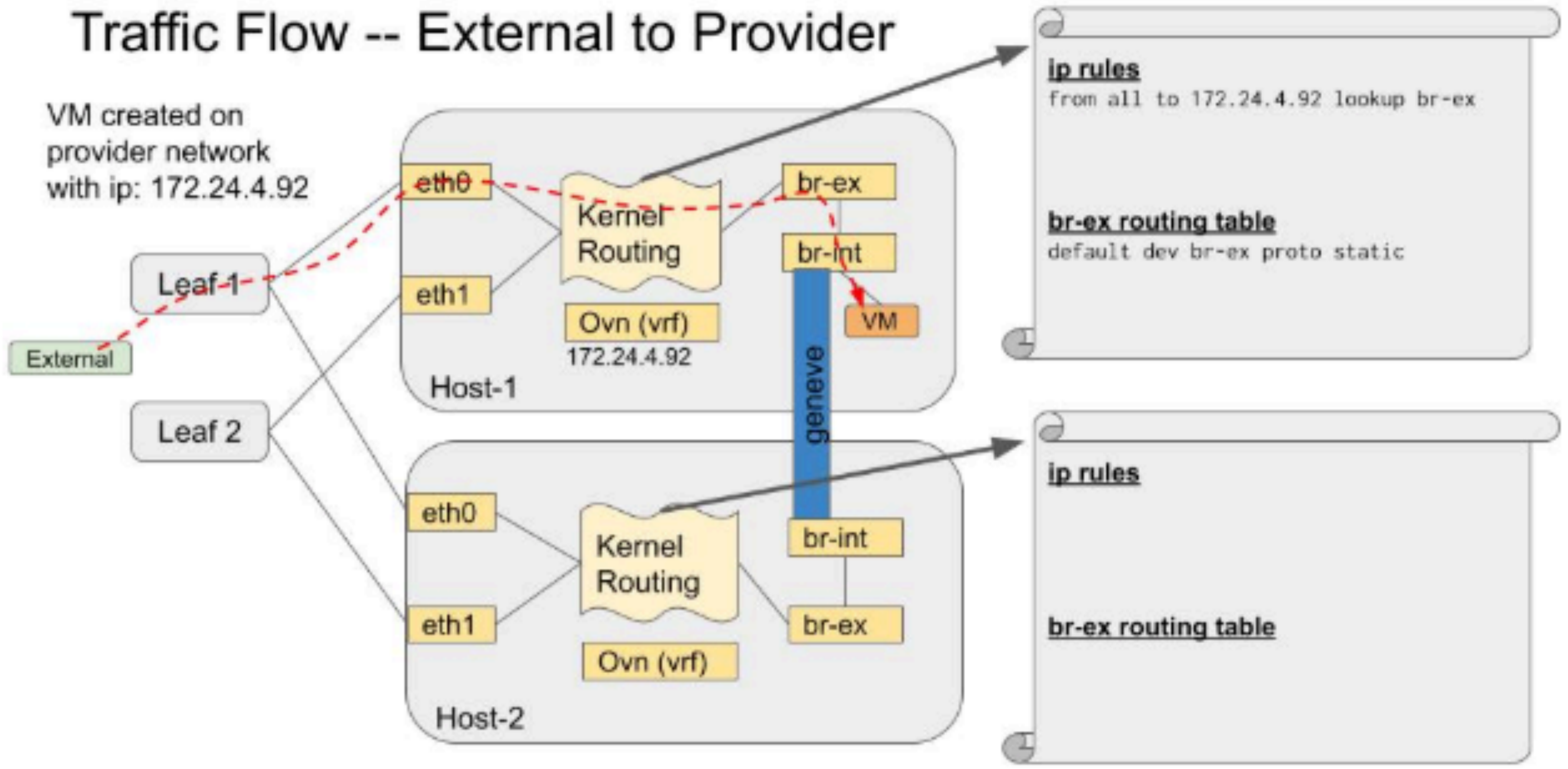
프로토콜(라우팅/정책/경로 제어)의 처리 주체를 Leaf에 두고, Spine은 요청을 가교하는 역할로만 사용

OVN-BGP Agent

https://ltomasbo.wordpress.com/2021/02/04/openstack-networking-with-bgp/
https://ltomasbo.wordpress.com/2021/02/04/ovn-bgp-agent-testing-setup/
https://ltomasbo.wordpress.com/2021/02/04/ovn-bgp-agent-in-depth-traffic-flow-inspection/

물리적 토폴로지는 각 서버가 연결된 랙의 리프에 레이어 2 연결을 갖는 스파인-리프 아키텍처로 구성됩니다. BGP는 패브릭, 즉 서버, 리프, 스파인에서 실행됩니다. 각 서버는 고가용성과 부하 분산을 위해 두 개의 리프에 연결됩니다. 이는 동일한 비용의 여러 경로를 통해 패킷을 라우팅하는 라우팅 기술인 ECMP(Equal-Cost Multi-Path)를 통해 구현됩니다. 이는 엄격한 요구 사항은 아니며, 기존에 본딩된 NIC를 사용하더라도 직접 연결된 경로가 올바른 네트워크를 통해 광고되도록 보장하는 데는 문제가 없습니다.





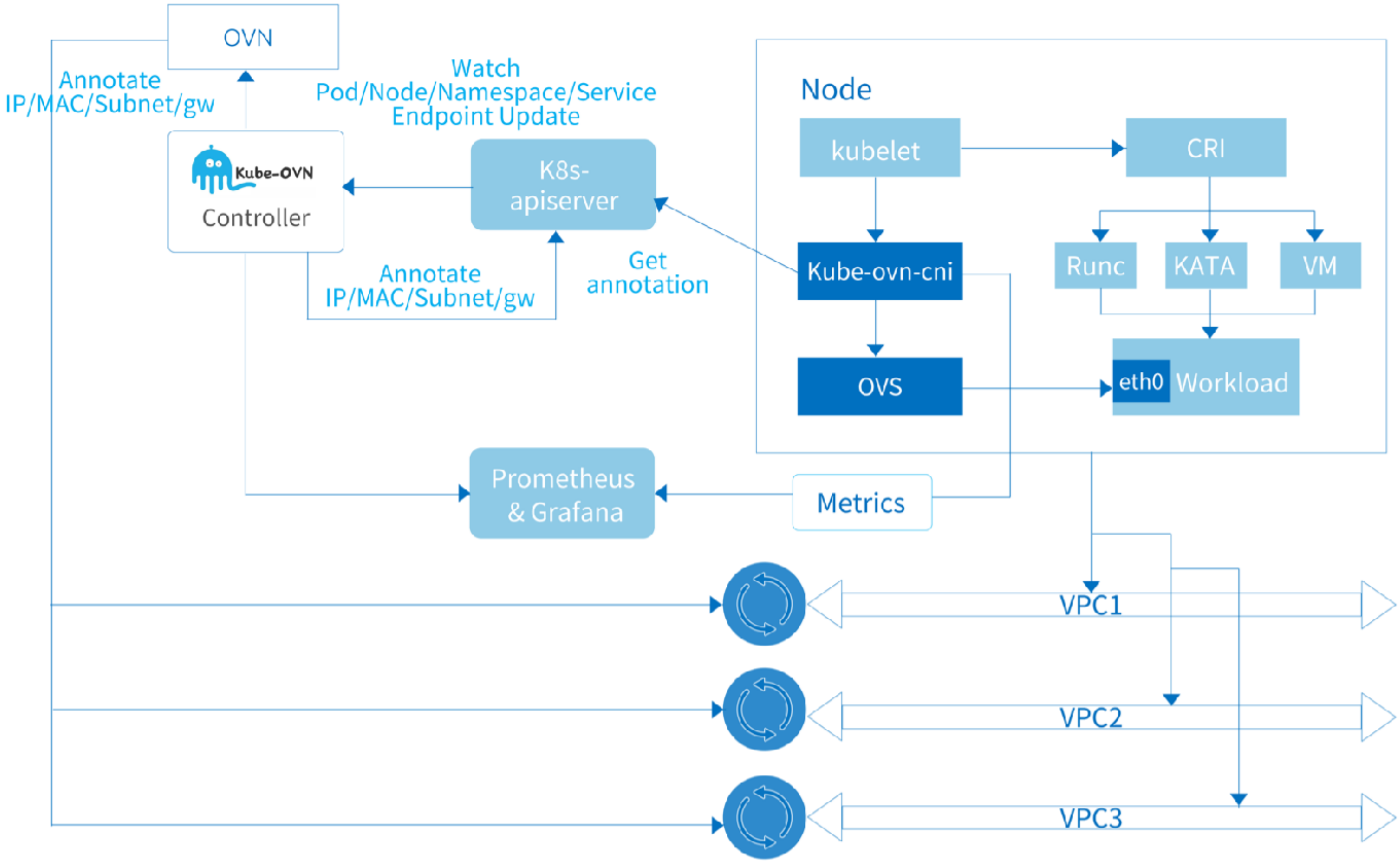
OVN-BGP 에이전트는 각 노드(OpenStack 컨트롤러/컴퓨팅 노드)에서 실행되는 Python 기반 데몬으로, 모든 노드가 BGP 피어(리프)에 연결되어 있다고 가정합니다. OVN Southbound 데이터베이스를 모니터링하여 VM 부팅/종료 시점과 FIP 연결/해제를 감지합니다. 이 경우, 커널 네트워킹 기능을 활용하여 해당 VM과 연결된 IP 주소가 BGP를 통해 광고되도록 필요한 재구성을 수행하고, 외부 트래픽을 OVN 오버레이로 라우팅하는 데 필요한 모든 작업을 수행합니다.

Design of OVN BGP Agent with EVPN

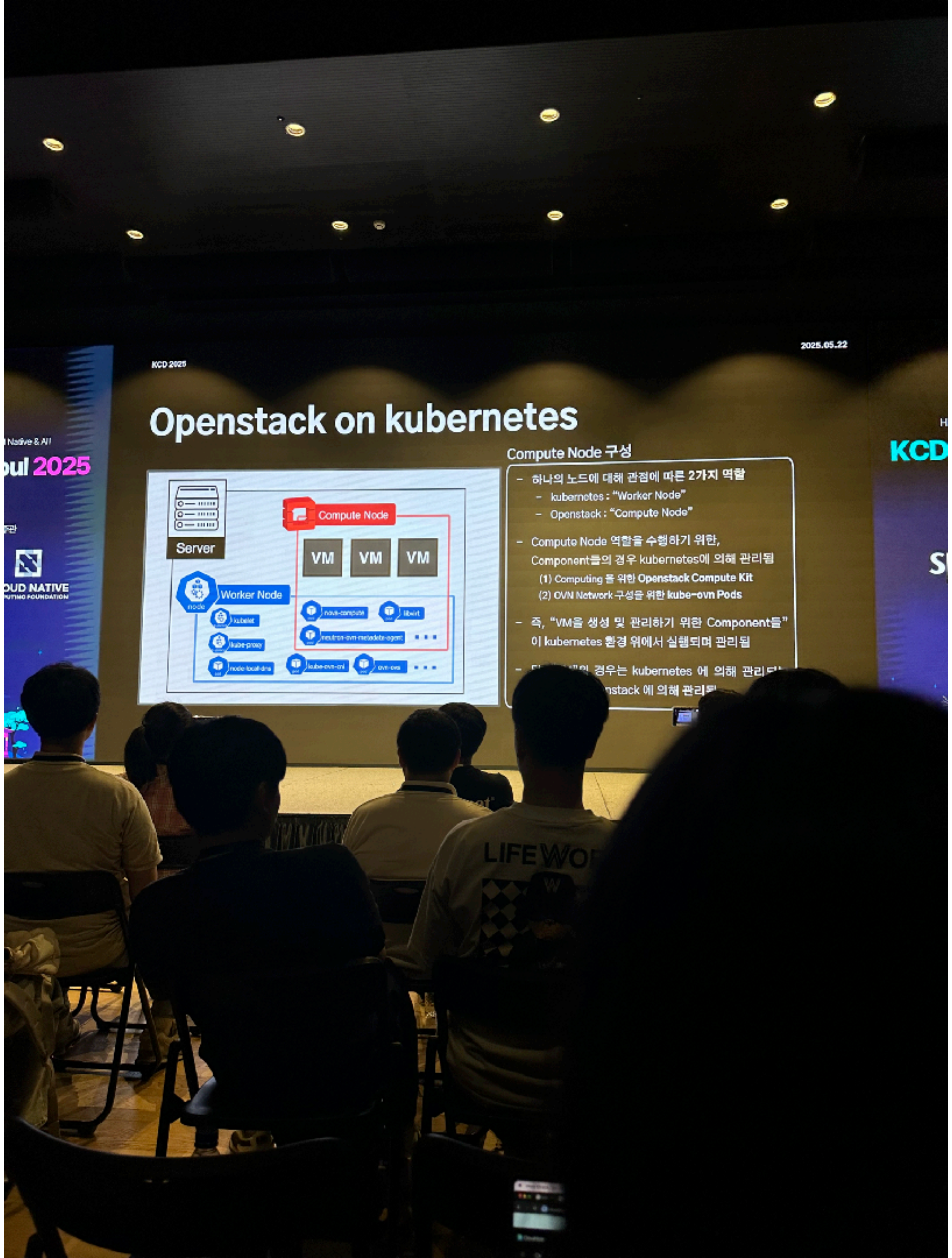
https://docs.openstack.org/ovn-bgp-agent/latest/contributor/drivers/evpn_mode_design.html

Kube-OVN

https://www.kube-ovn.io/



KCD 2025
2025.05.22
Pod Network 구현
Pod Network의 구현
– "ovn-ovs" Pod 내 동작중인 "ovn-controller" Process 는 ovn-central 과 통신하며, 해당 노드의 OVS 브릿지 내 Flow 들을 설정 및 관리하면서 실질적으로 가상 네트워크를 구현함
Pod to Pod/External 통신
– Pod 간 통신의 경우 Pod에서, OVS 브릿지를 거쳐서 Geneve 기반의 터널링을 통해 상호 통신
LIFEWORK



KCD 2025
2025.05.22
Openstack on kubernetes
Server
Compute Node
VM
VM
VM
Worker Node
Compute Node 구성
– 하나의 노드에 대해 관점에 따른 2가지 역할
– kubernetes : "Worker Node"
– Openstack : "Compute Node"
– Compute Node 역할을 수행하기 위한, Component들의 경우 kubernetes에 의해 관리됨
(1) Computing 을 위한 Openstack Compute Kit
(2) OVN Network 구성을 위한 kube-ovn Pods
– 즉, "VM을 생성 및 관리하기 위한 Component들" 이 kubernetes 환경 위에서 실행되며 관리됨