Open vSwitch
2023 Conference

# Linux Networking & OVS as a Pipeline

Speaker: Dan Daly, Intel

**Team:** Namrata Limaye, Venkata Suresh Kumar, Sandeep Nagapattinam, Nupur Uttarwar, Intel

# Flow vs. Pipeline

## Flow

Exact Match

↓

Megaflow Match

Issues:

- Combo Explosions
- Butterfly Effects

## Pipeline

Fixed Function

↓

Programmable

Issues:

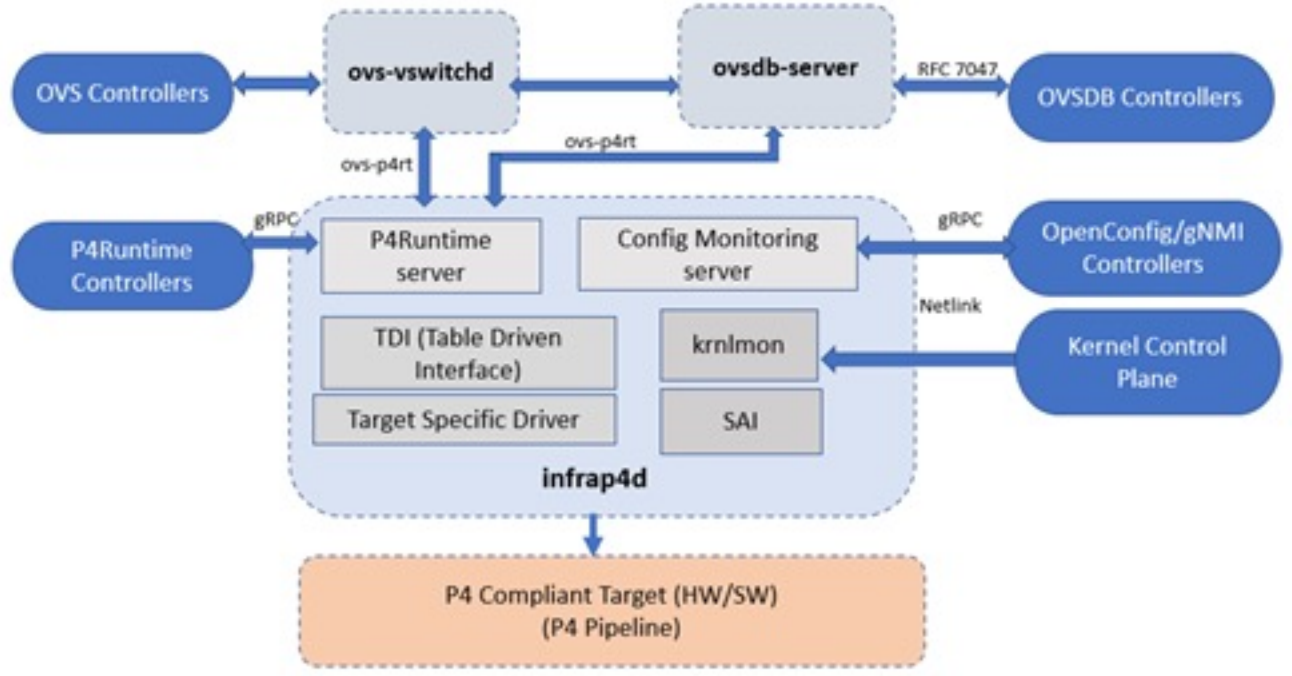
- A priori knowledge
- Wildcards in Software

# P4 Control Plane

Open vSwitch

2023 Conference

https://ipdk.io/p4cp-userguide/



Re-implemented ‘P4 OVS’ as part of **Linux Networking as a Pipeline**

# Golden Rule

Linux Behavior == Linux as a Pipeline Behavior

Differences must be called out / disallowed

Rule:

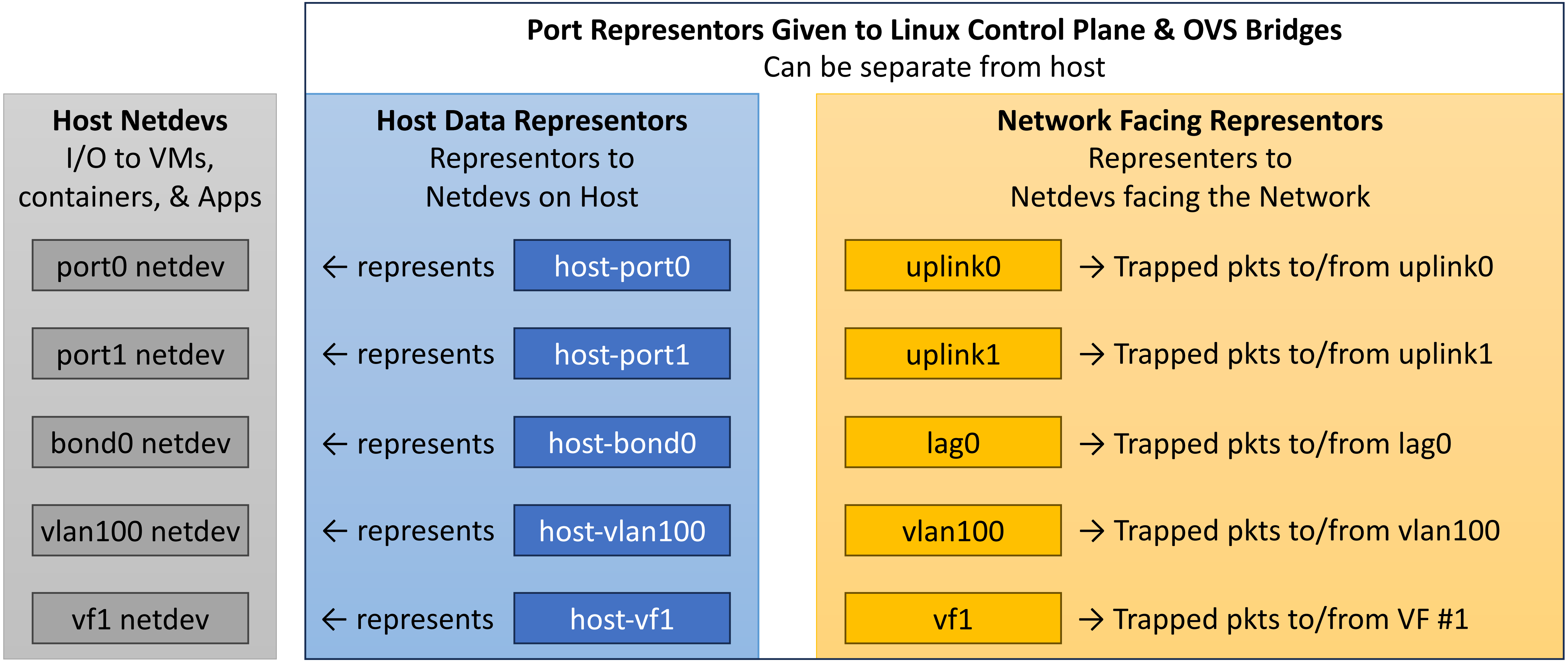
Linux Kernel is the Golden Model

# Port Representors

## Pipeline acts as a co-processor

Open vSwitch

2023 Conference



**Port Representors Given to Linux Control Plane & OVS Bridges**

Can be separate from host

| **Host Netdevs**<br>I/O to VMs, containers, & Apps | | **Host Data Representors**<br>Representors to Netdevs on Host | **Network Facing Representors**<br>Representers to Netdevs facing the Network | |
|---|---|---|---|---|
| port0 netdev | ← represents | host-port0 | uplink0 | → Trapped pkts to/from uplink0 |
| port1 netdev | ← represents | host-port1 | uplink1 | → Trapped pkts to/from uplink1 |
| bond0 netdev | ← represents | host-bond0 | lag0 | → Trapped pkts to/from lag0 |
| vlan100 netdev | ← represents | host-vlan100 | vlan100 | → Trapped pkts to/from vlan100 |
| vf1 netdev | ← represents | host-vf1 | vf1 | → Trapped pkts to/from VF #1 |

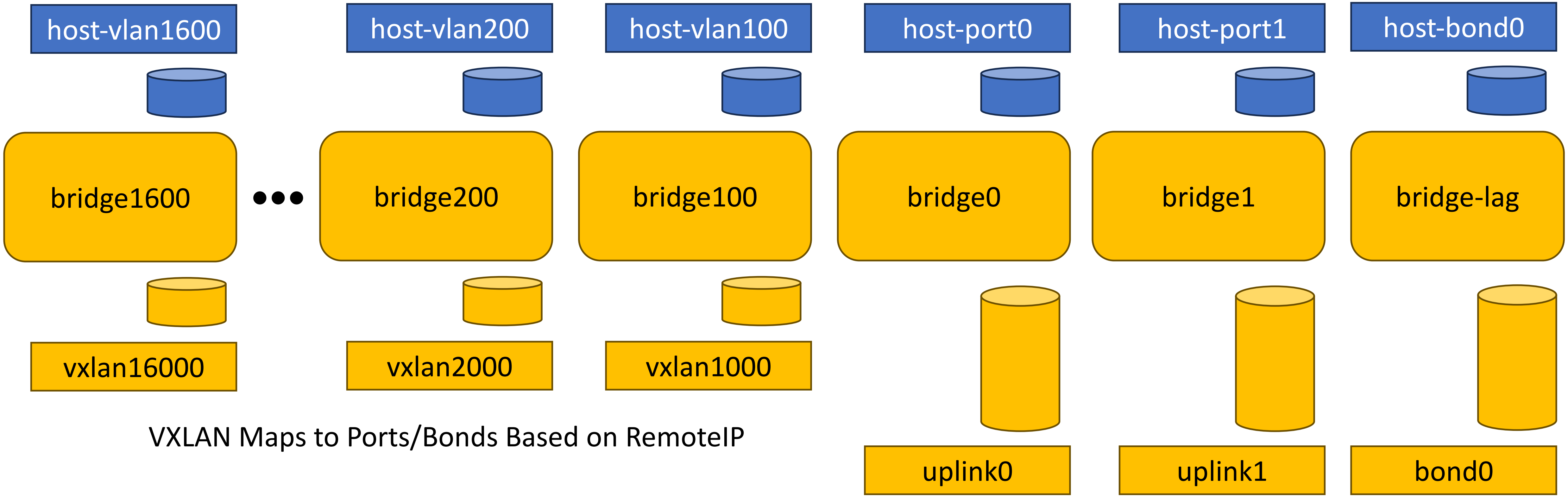
# Example: Map Tunnels → VLAN

Example as seen from OVS

Open vSwitch

2023 Conference

VLAN Maps to Ports/Bonds Based on Host Config



host-vlan1600

host-vlan200

host-vlan100

host-port0

host-port1

host-bond0

bridge1600 ••• bridge200

bridge100

bridge0

bridge1

bridge-lag

vxlan16000

vxlan2000

vxlan1000

uplink0

uplink1

bond0

VXLAN Maps to Ports/Bonds Based on RemoteIP

# Example: Map Tunnels → VLAN

## Example as seen from OVS

```
$ ovs-vsctl add-br bridge0
$ ovs-vsctl add-port bridge0 host-port0
$ ovs-vsctl add-port bridge0 uplink0
$ ovs-vsctl add-br bridge1
$ ovs-vsctl add-port bridge1 host-port1
$ ovs-vsctl add-port bridge1 uplink1
```

```
$ ovs-vsctl add-br bridge-lag
$ ovs-vsctl add-port bridge1 host-bond0
$ ovs-vsctl add-port bridge1 lag0
```

```
$ ovs-vsctl add-br bridge100
$ ovs-vsctl add-port bridge100 VXLAN1000 -- set interface VXLAN1000 type=VXLAN
options:remote_ip=10.100.0.1 options:key=1000
$ ovs-vsctl add-port bridge100 host-bond0 tag=100
$ ovs-vsctl add-br bridge200
$ ovs-vsctl add-port bridg200 VXLAN2000 -- set interface VXLAN2000 type=VXLAN
options:remote_ip=10.100.0.15 options:key=2000
$ ovs-vsctl add-port bridge200 host-bond0 tag=200
…
$ ovs-vsctl add-br bridge1600
$ ovs-vsctl add-port bridge1600 VXLAN16000 -- set interface VXLAN16000 type=VXLAN
options:remote_ip=10.100.0.1 ip options:key=16000
$ ovs-vsctl add-port bridge1600 host-bond0 tag=1600
```

# Configuration Example

Open vSwitch
2023 Conference

Bond config ”create lag” → Linux Create LAG → netlink → …
… → infrap4d-krnlmon → SAI → TDI → Pipeline Target

OVS config ”create tunnel” → Linux Create Tunnel → netlink → …
… → infrap4d-krnlmon → SAI → TDI → Pipeline Target

Also → Linux Maps ‘RemoteIP’ → MAC using ARP/ND table

Patch to OVS

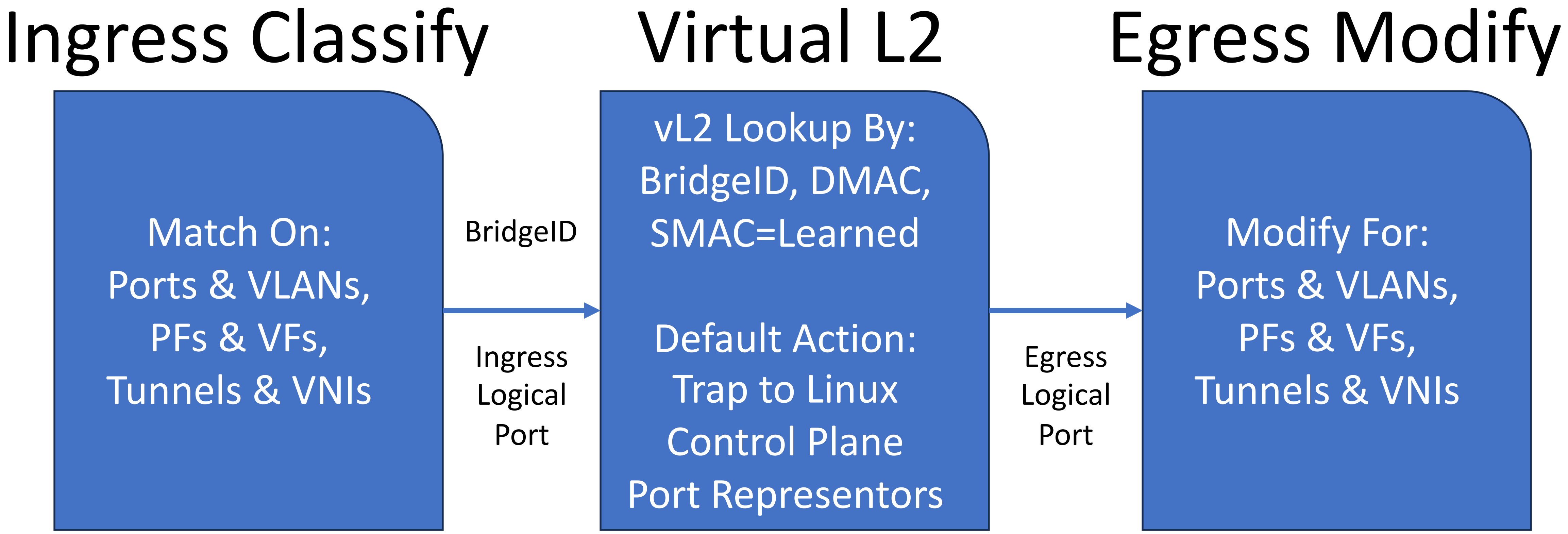
OVS ”Learns MAC” → P4Runtime Call → …

… → infrap4d-p4runtime → TDI → Pipeline Target

# Virtual L2 Pipeline

Open vSwitch

2023 Conference



Ingress Classify

Match On:
Ports & VLANs,
PFs & VFs,
Tunnels & VNIs

BridgeID

Ingress
Logical
Port

Virtual L2

vL2 Lookup By:
BridgeID, DMAC,
SMAC=Learned

Default Action:
Trap to Linux
Control Plane
Port Representors

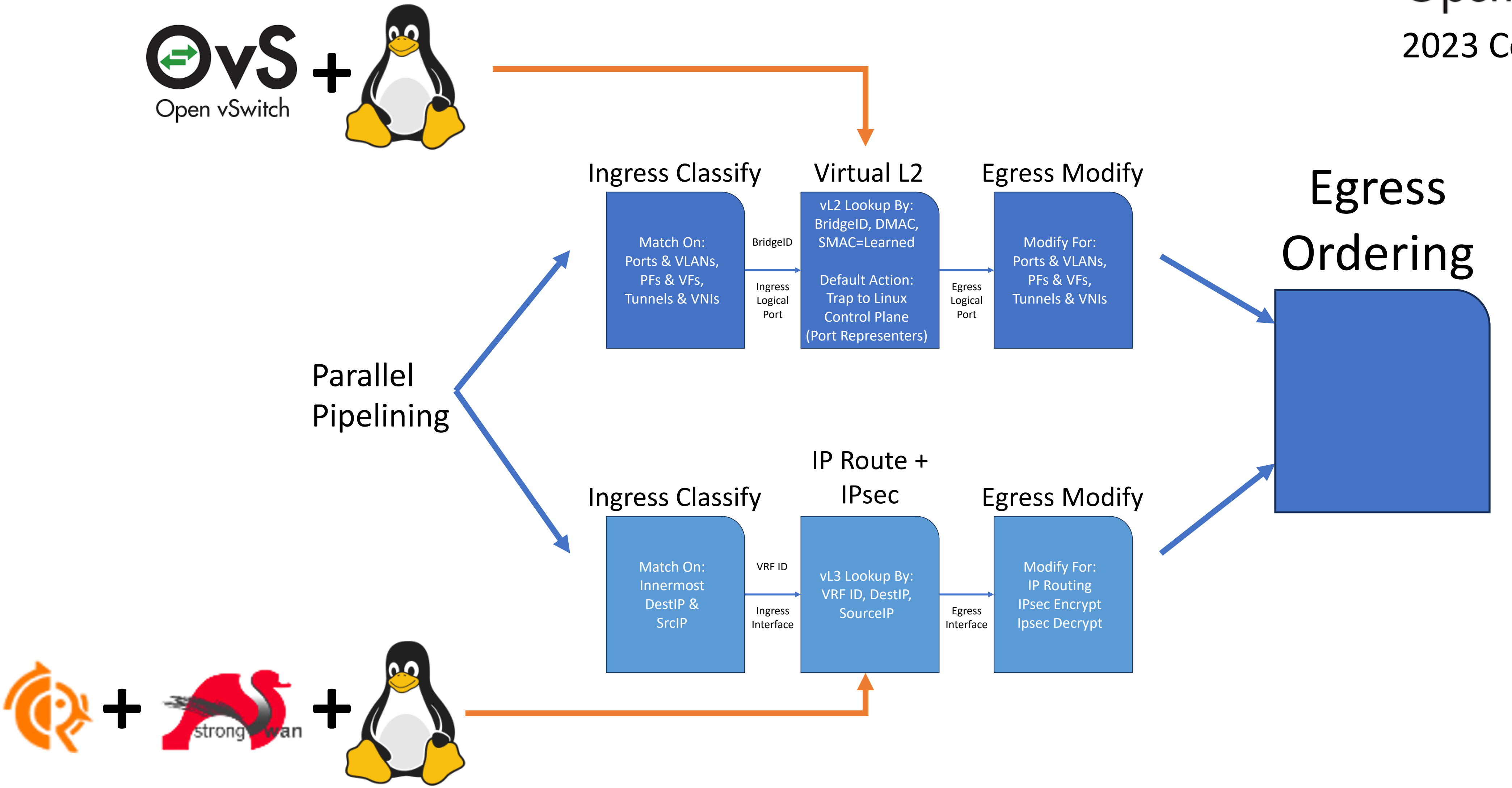
Egress
Logical
Port

Egress Modify

Modify For:
Ports & VLANs,
PFs & VFs,
Tunnels & VNIs

# Make it a Combo!

Open vSwitch
2023 Conference



Open vSwitch +

Parallel Pipelining

Ingress Classify

Match On:
Ports & VLANs,
PFs & VFs,
Tunnels & VNIs

BridgeID

Ingress Logical Port

Virtual L2

vL2 Lookup By:
BridgeID, DMAC,
SMAC=Learned

Default Action:
Trap to Linux
Control Plane
(Port Representers)

Egress Logical Port

Egress Modify

Modify For:
Ports & VLANs,
PFs & VFs,
Tunnels & VNIs

Ingress Classify

Match On:
Innermost
DestIP &
SrcIP

VRF ID

Ingress Interface

IP Route + IPsec

vL3 Lookup By:
VRF ID, DestIP,
SourceIP

Egress Interface

Egress Modify

Modify For:
IP Routing
IPsec Encrypt
Ipsec Decrypt

Egress Ordering

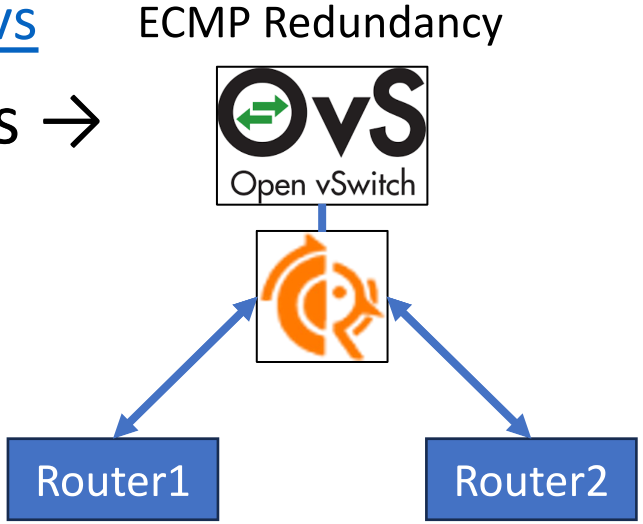
+ strongSwan +

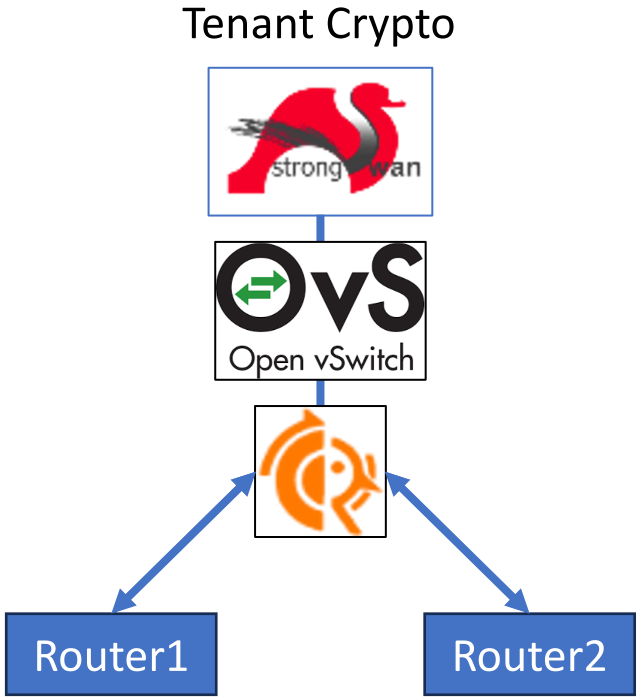
# Summary

Open vSwitch
2023 Conference

Linux as a Pipeline

- Full Feature Fidelity
- Enables Optimizations & Offloading
- New! Patch to Open vSwitch
  - https://github.com/ipdk-io/ovs
- Supports Real World Combos →



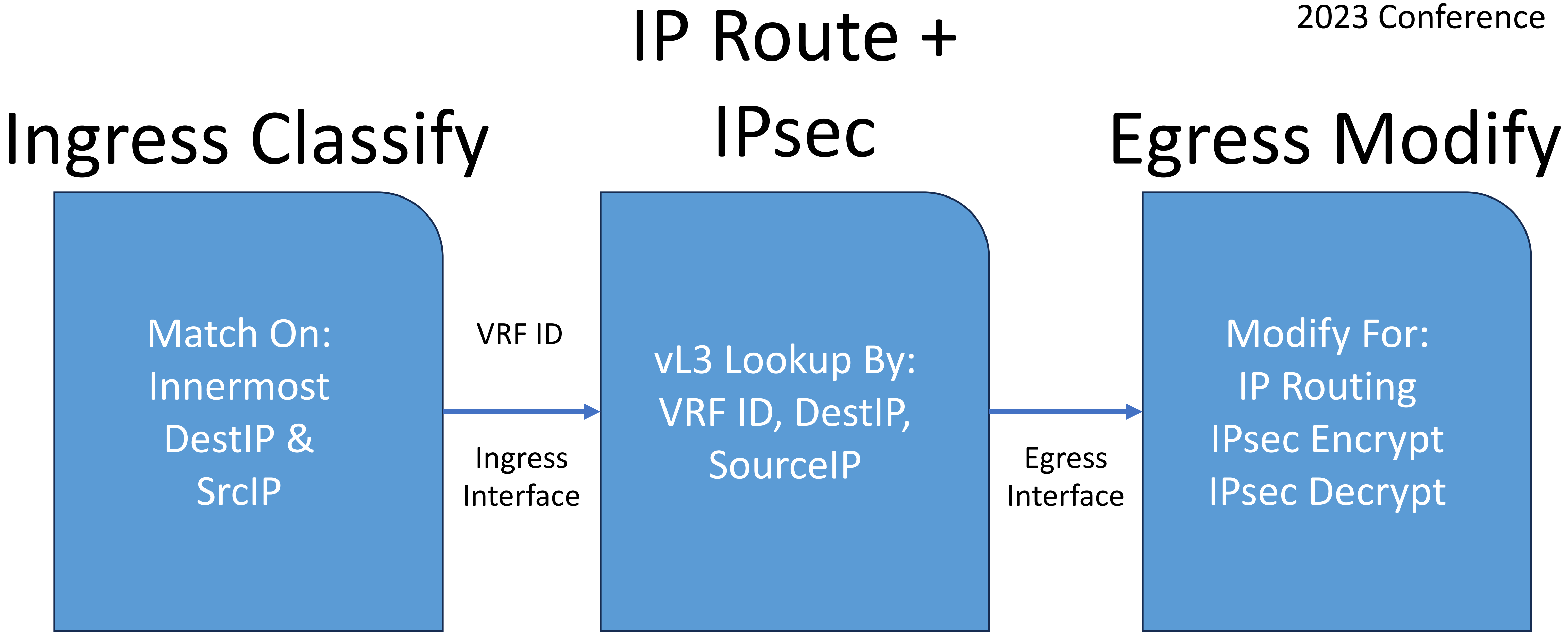


Open vSwitch
2023 Conference

# Thank You!

OvS
Open vSwitch
2023 Conference

# Virtual L3 Pipeline



Ingress Classify

Match On:
Innermost
DestIP &
SrcIP

VRF ID

Ingress
Interface

IP Route +
IPsec

vL3 Lookup By:
VRF ID, DestIP,
SourceIP

Egress
Interface

Egress Modify

Modify For:
IP Routing
IPsec Encrypt
IPsec Decrypt