



KubeCon



CloudNativeCon

Europe 2019

Network Machinery

A United-Front For Network Troubleshooting
with CRDs

Adel Zaalouk, SAP - @ZaNetworker





- The State of Network Troubleshooting in Kubernetes
- CRDs Are Not Just for Add-ons, they are for Networking Too
- Use-Case I: Network Reachability & Traffic Shaping CRDs
 - Demo
- Use-Case II: Kubernetesized-SDN CRDs
 - Demo

Networking Landscape in Kubernetes



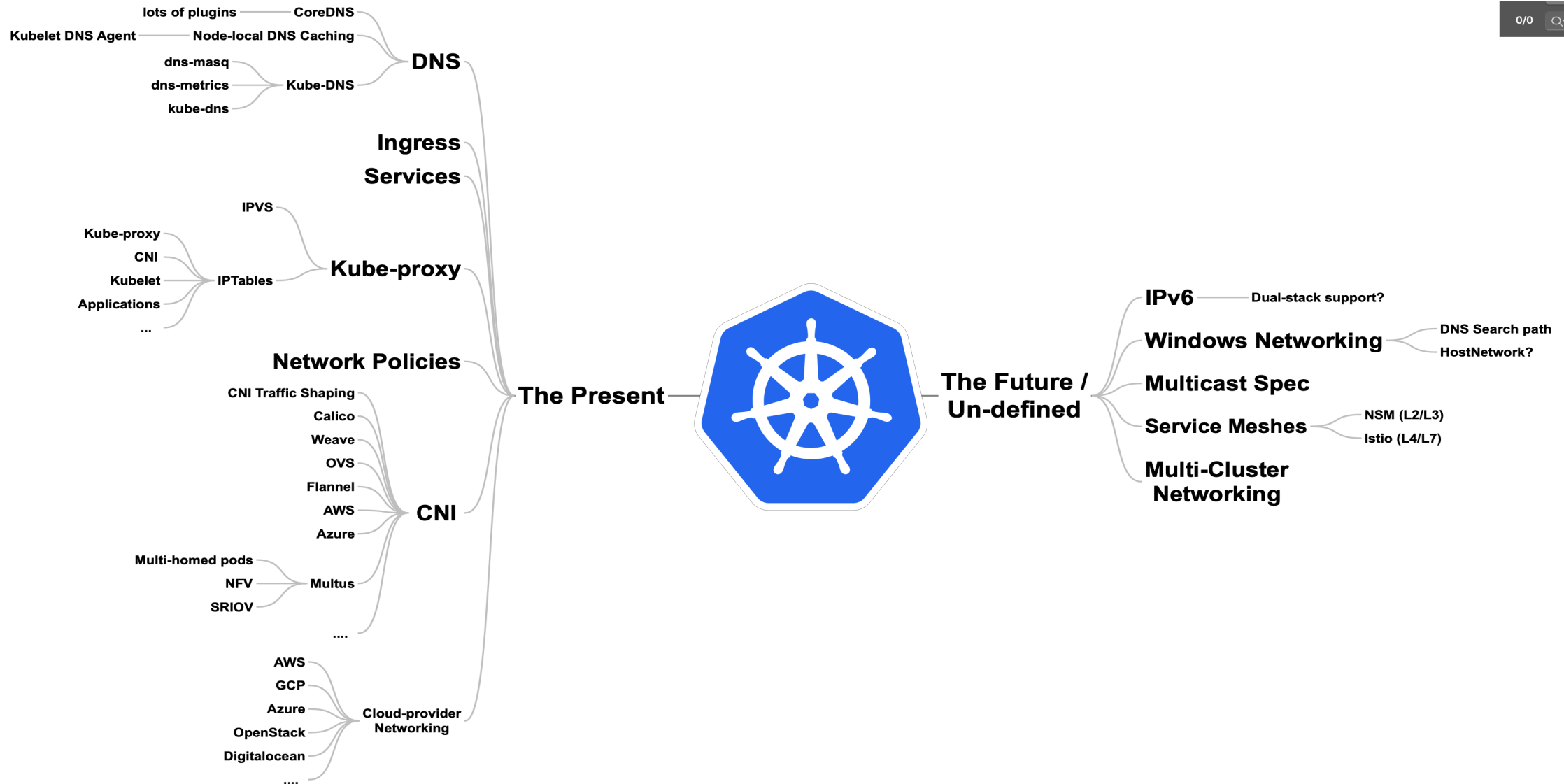
KubeCon



CloudNativeCon

Europe 2019

0/0 Q Search



Previous Troubleshooting Talks / Takes

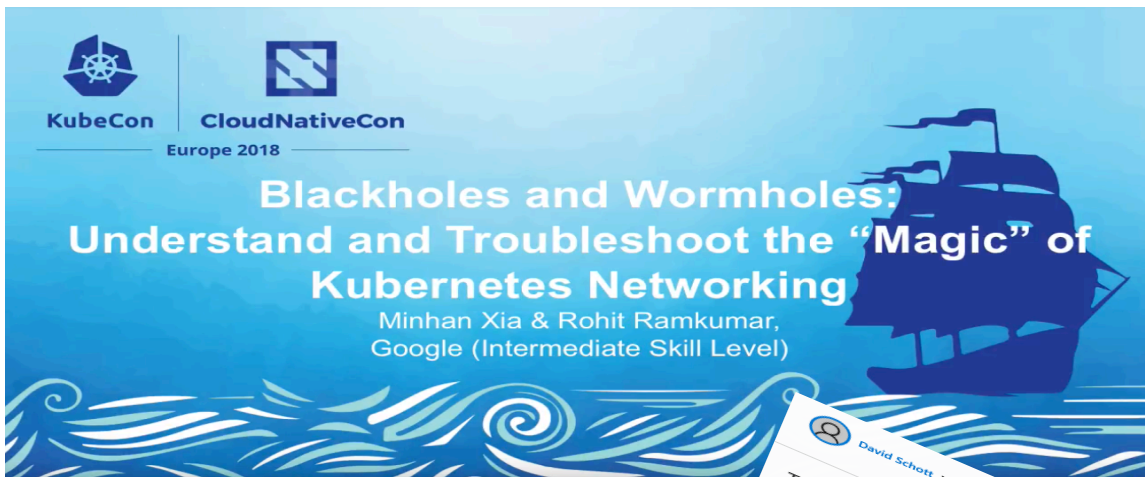


KubeCon



CloudNativeCon

Europe 2019



- Blackhole #1**
 - Conntrack entries are stale, traffic is being sent to ghost pods
- Blackhole #2**
 - `net.ipv4.conf.eth0.forwarding=0`
- Wormhole**
 - Conflicting iptable rules between Kubelet (HostPort) and Kube-proxy
- Packet loss**
 - tcpdump on pods / nodes in sender / receiver sides
- No tool just good practices**

No common place, no APIs.
Great tools just die out.
How to pool-in this knowledge in Kubernetes in a standard way?

How to pool-in this knowledge in Kubernetes Networking

- Symptoms of Kubernetes Network Failures**
 - Failed Create Network Sandbox
 - CNI Plugin / Config Failed**
 - Bad CNI: Pods Readiness failed (2/3)
- Identifying Network Interfaces**
- IPTables in Kubernetes**
- Troubleshooting (Packet Capture)**
- Tool Introduced (Kokotap: tap traffic over vxlan)**

CRDs Aren't Just For Addons



KubeCon



CloudNativeCon

Europe 2019

CRDs Aren't Just For Addons

KubeCon NA, Seattle
12/2018

Tim Hockin <thockin@google.com>
Principal Software Engineer
@thockin



Kube-style APIs are simple and powerful

We already have an API server (and Storage)

Birth of the Operator: Declarative APIs, actuated Asynchronously by Controllers

CRDs are no longer 2nd Class

Declarative Schema Validation with OpenAPI v3

Native-feeling APIs without changing Kubernetes code!

Admission (Mutation and Validation)

CRDs Are For Networking Too

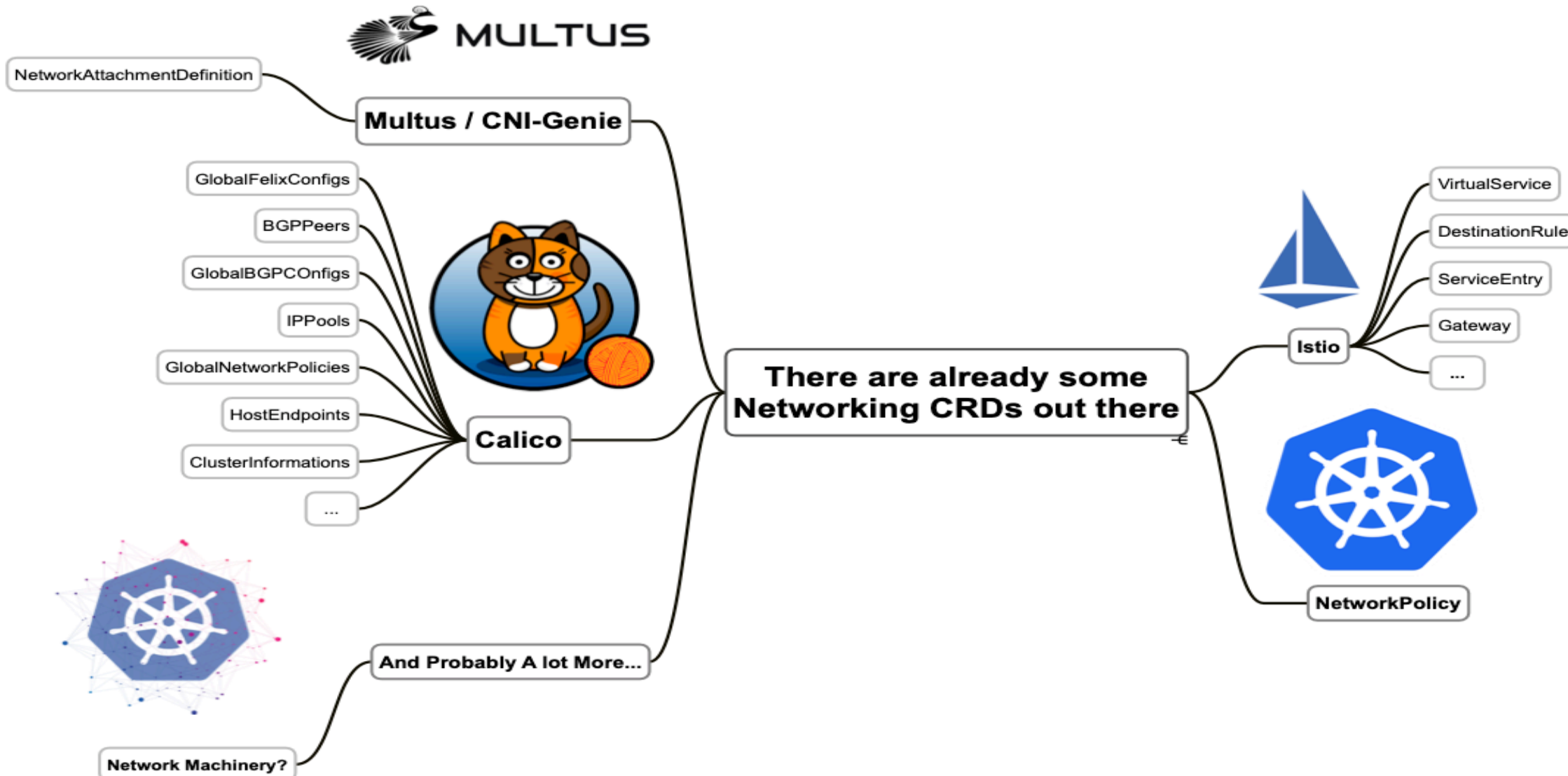


KubeCon



CloudNativeCon

Europe 2019



Network Machinery: The Idea



KubeCon



CloudNativeCon

Europe 2019

Utilize CRDs to build Network Troubleshooting Operators.

- Very familiar and widely accepted by the community.
- Many helper frameworks available.
- Declarative configuration for the resources.
- Out-of-the-Box feature-set such as:
 - Validating / Mutating / Conversion Webhooks
 - Versioned APIs with auto Code-Gen
 - ...

Network Machinery Collection

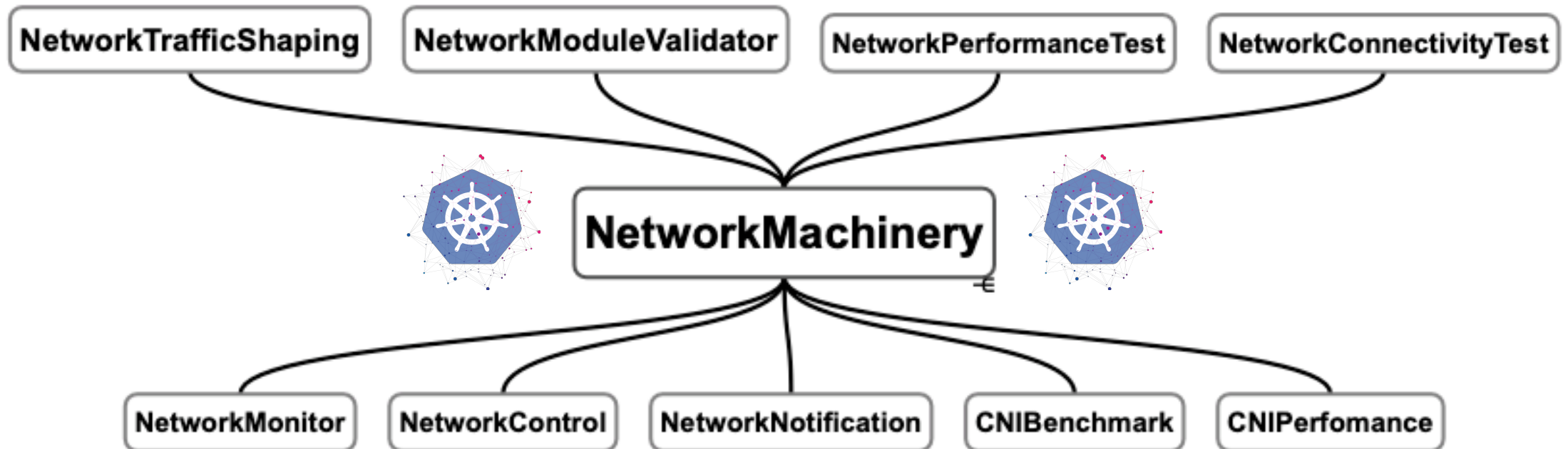


KubeCon



CloudNativeCon

Europe 2019



Network Machinery Collection

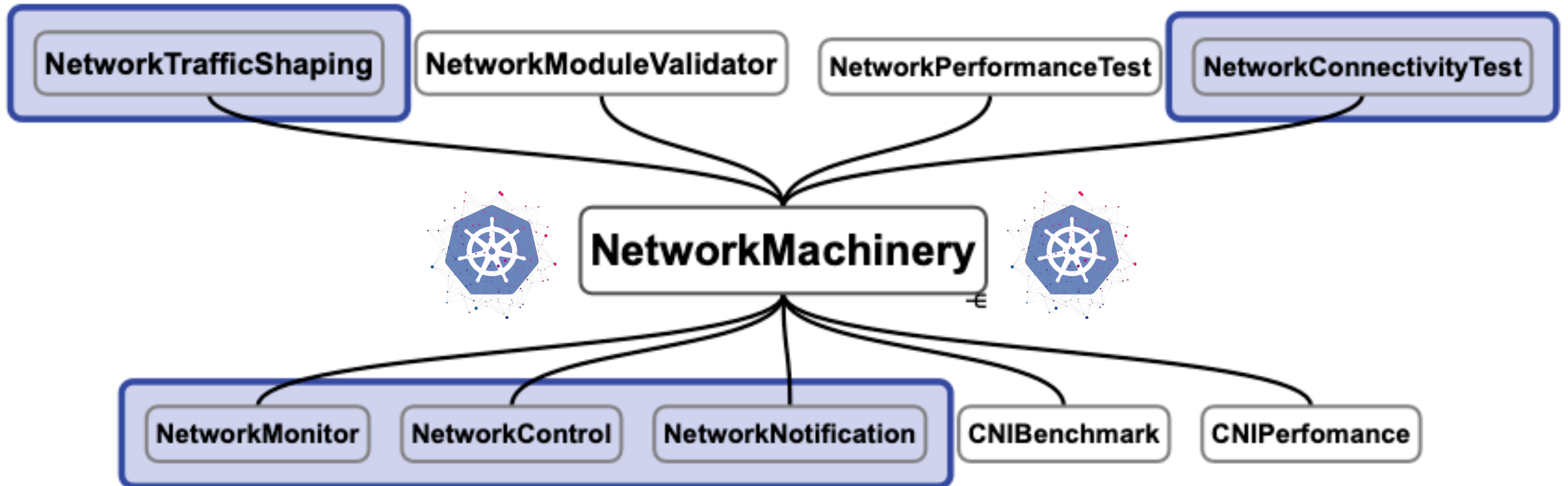


KubeCon



CloudNativeCon

Europe 2019



Reachability & Traffic Shaping

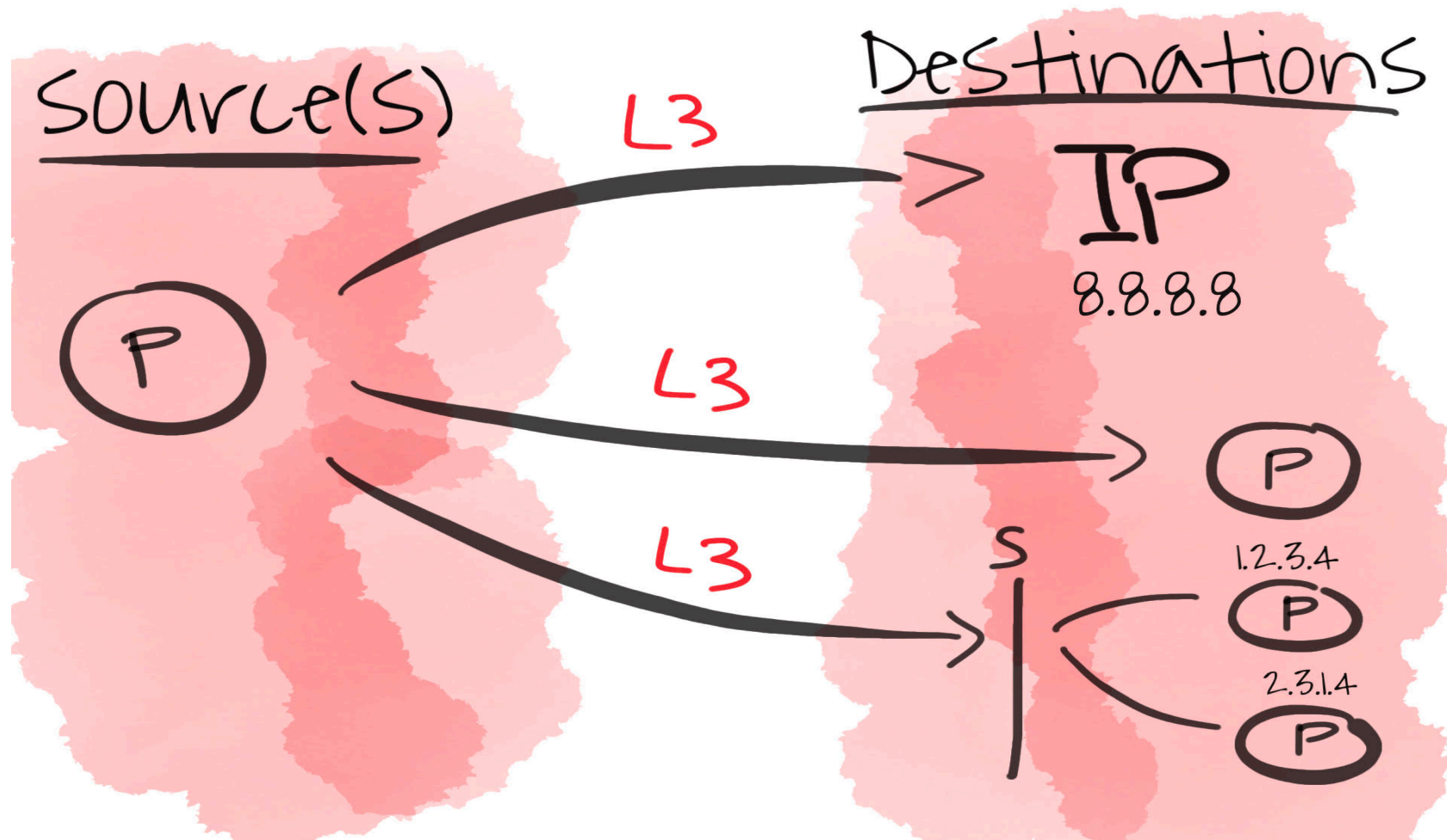


KubeCon



CloudNativeCon

Europe 2019



Reachability & Traffic Shaping

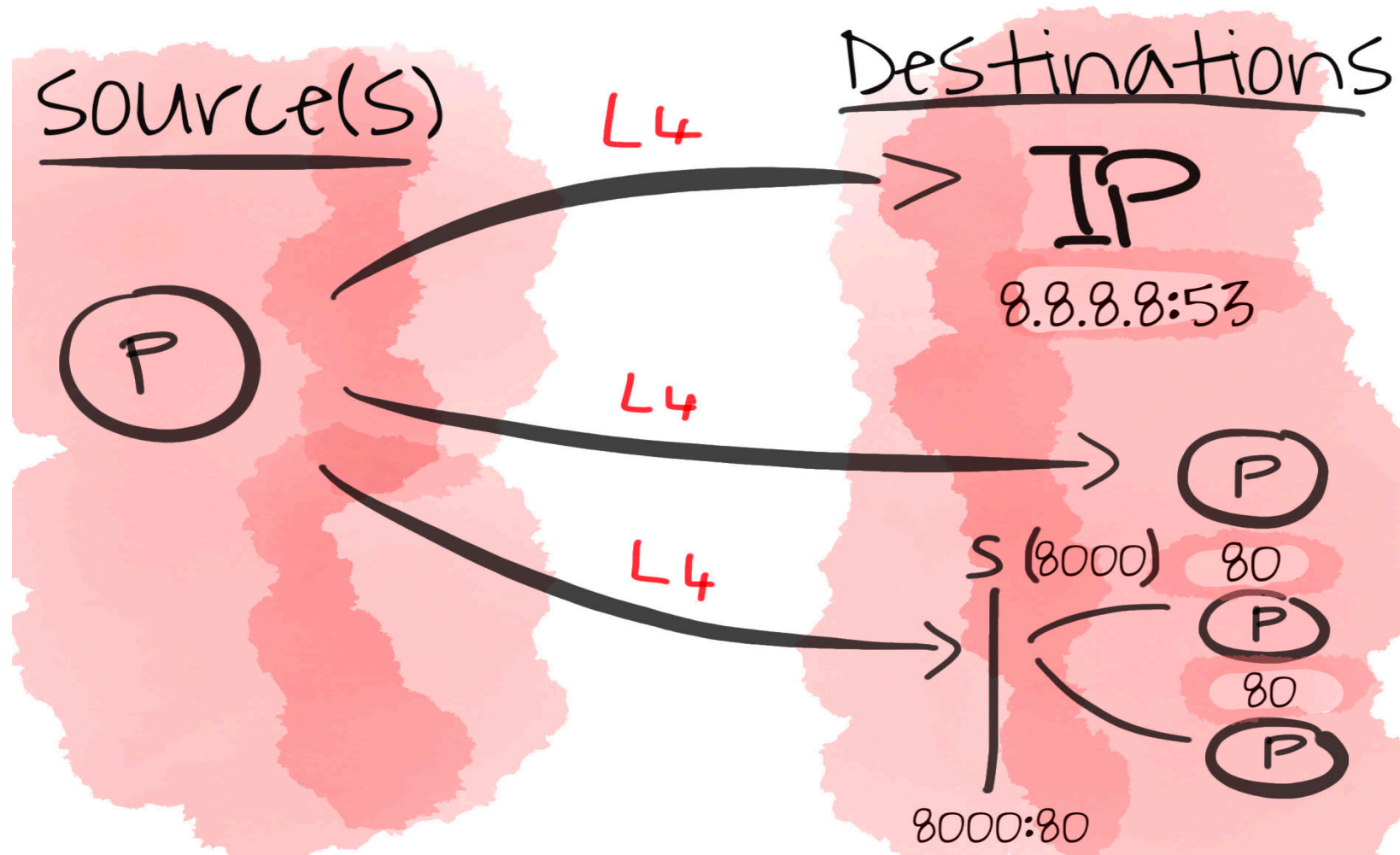


KubeCon



CloudNativeCon

Europe 2019



Reachability & Traffic Shaping

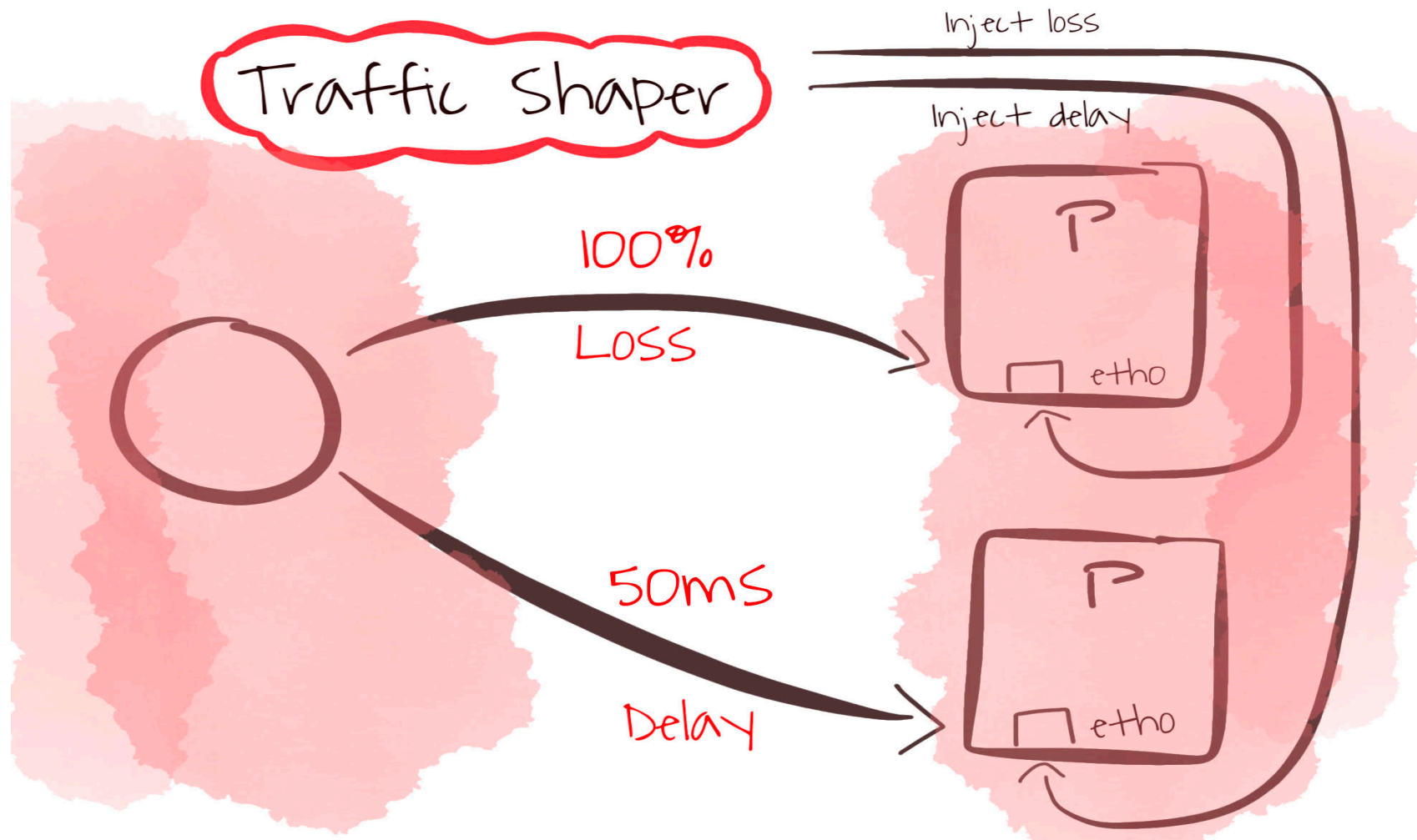


KubeCon



CloudNativeCon

Europe 2019





Demo time

Network Visibility & Control



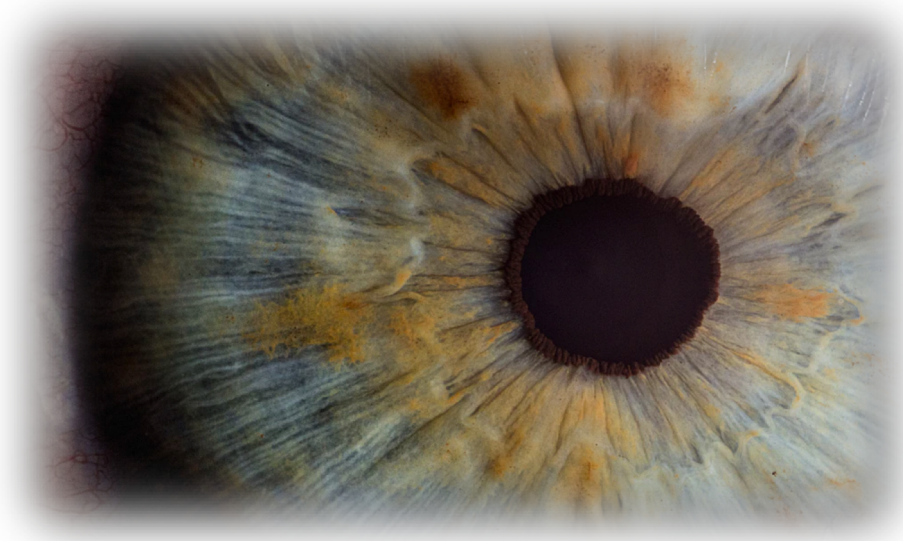
KubeCon



CloudNativeCon

Europe 2019

- **L3 & L4 connectivity and performance checks are not enough.**
- **We need more intel on what's happening in the network.**
 - Network Monitoring
 - Networking Control
- **We need to SEE and DO!**



SDN / OpenFlow / sFlow Capsule



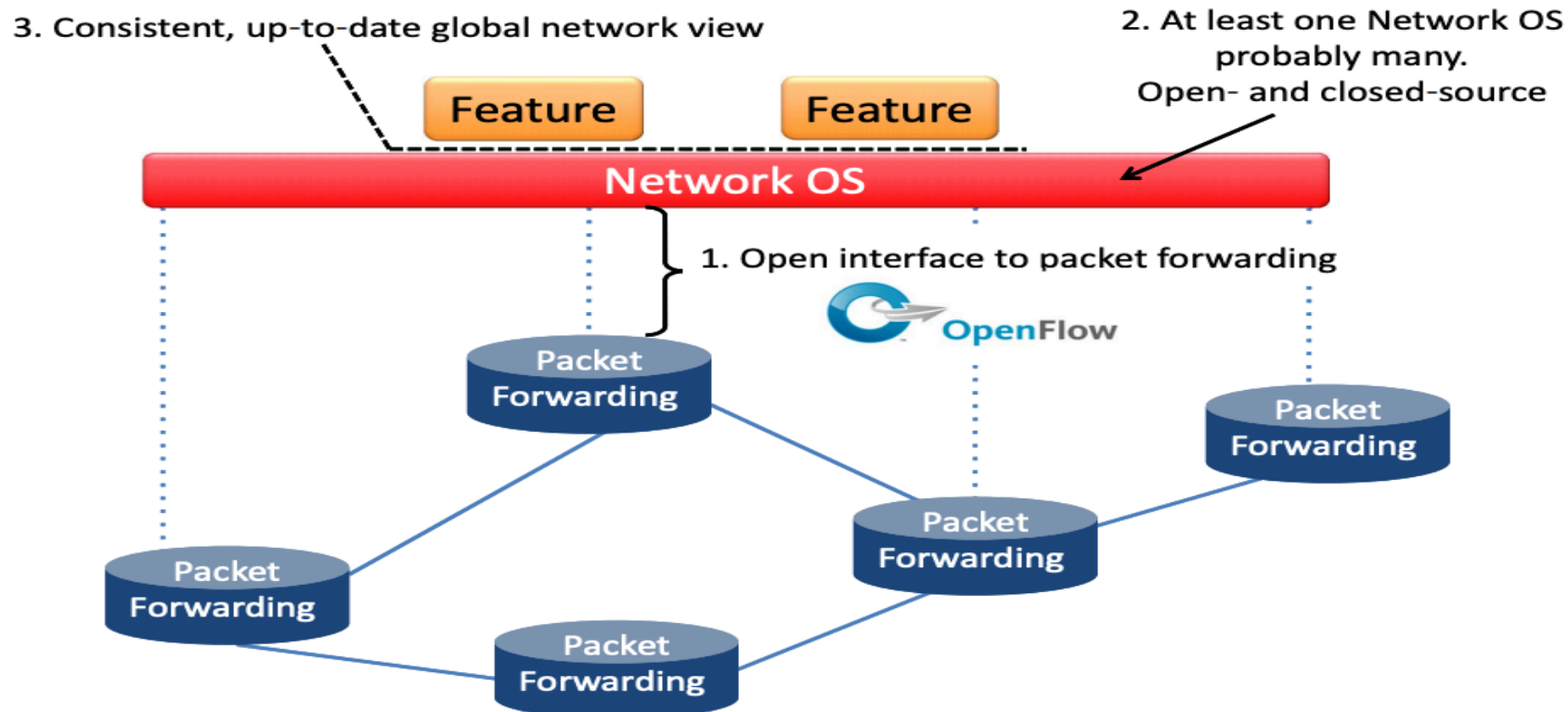
KubeCon



CloudNativeCon

Europe 2019

- SDN is about the Separation of the Control-Plane and Data-Plane
- **An early effort for programmable networks**



SDN / OpenFlow / sFlow Capsule



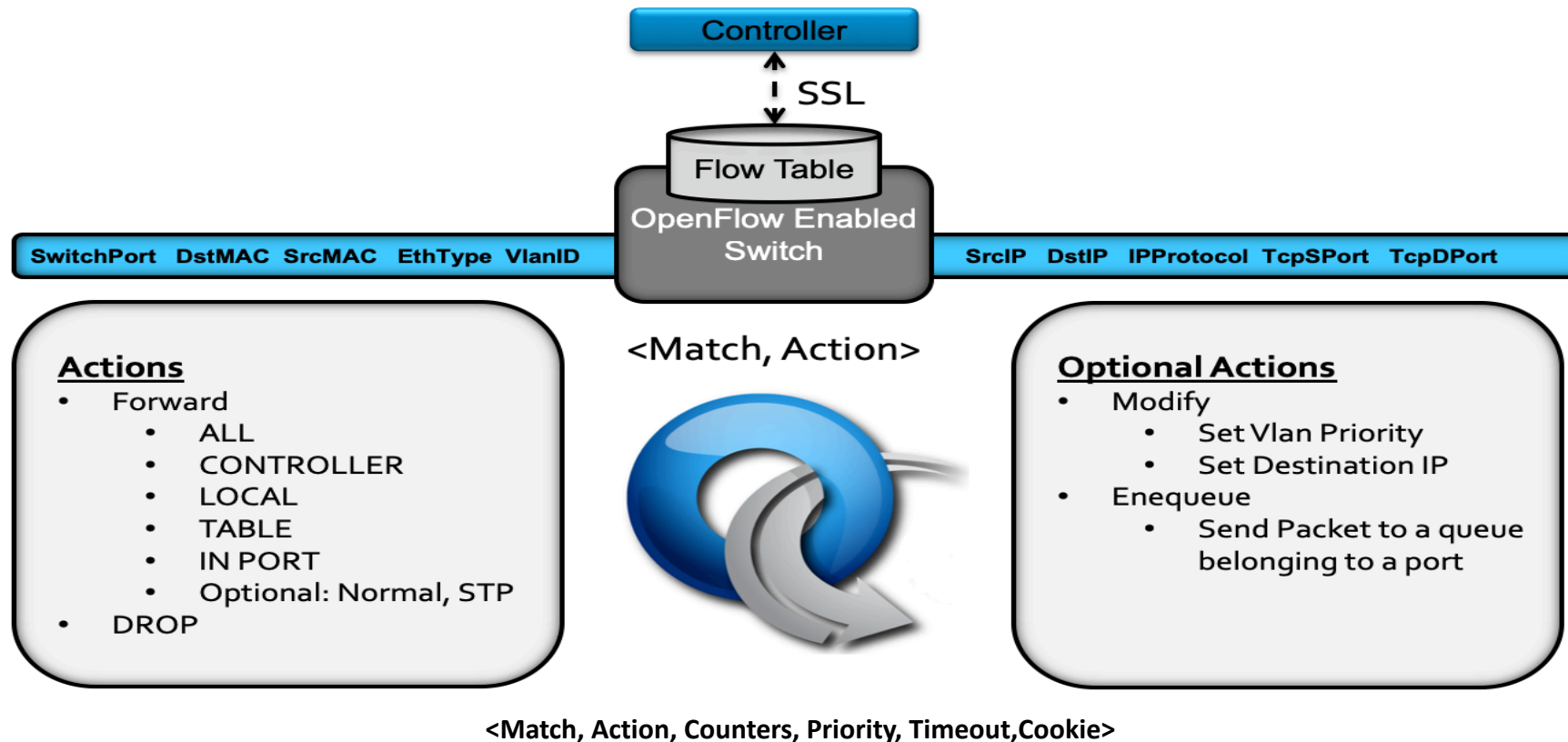
KubeCon



CloudNativeCon

Europe 2019

- SDN is about the Separation of the Control-Plane and Data-Plane
- An early effort for programmable networks



SDN / OpenFlow / sFlow Capsule



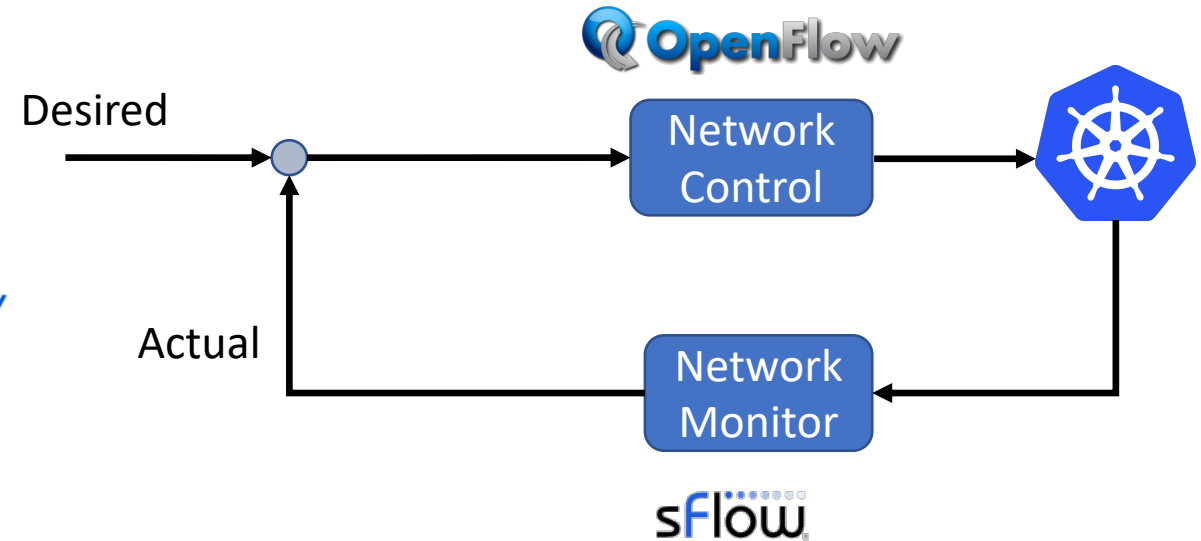
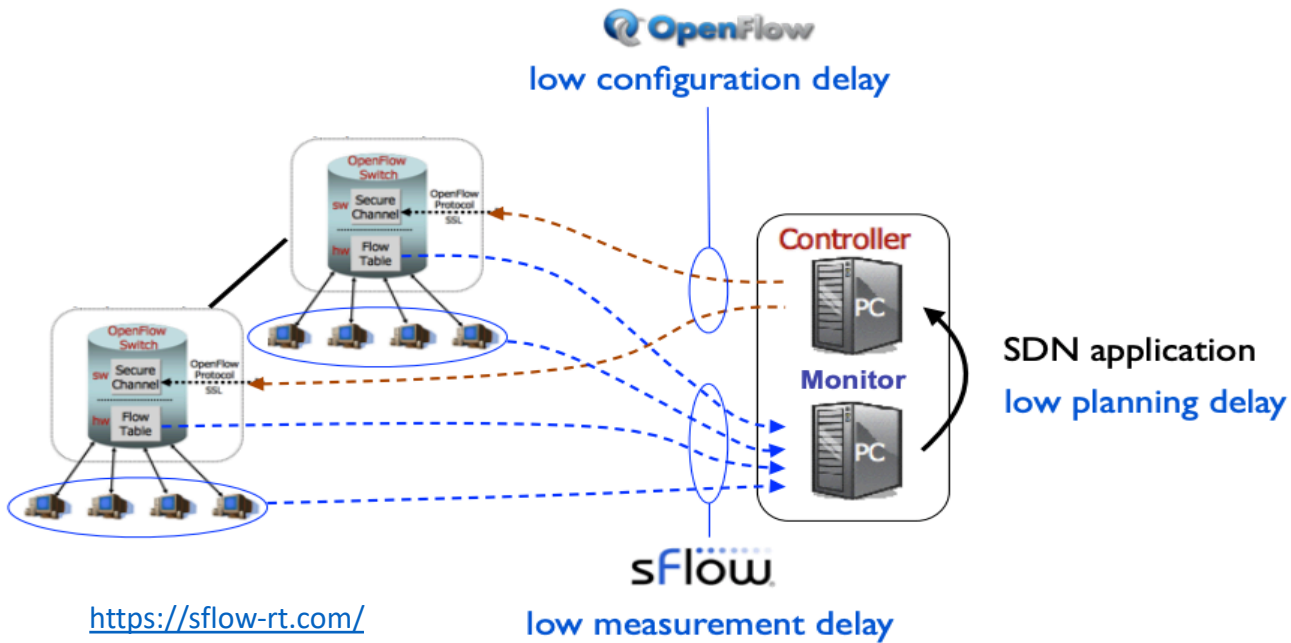
KubeCon



CloudNativeCon

Europe 2019

Our goal is to close the loop (Network Monitoring / Control)



SDN in Containers Context (OVS)

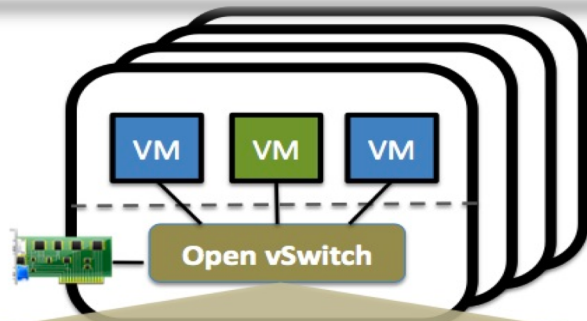



KubeCon




CloudNativeCon


Europe 2019



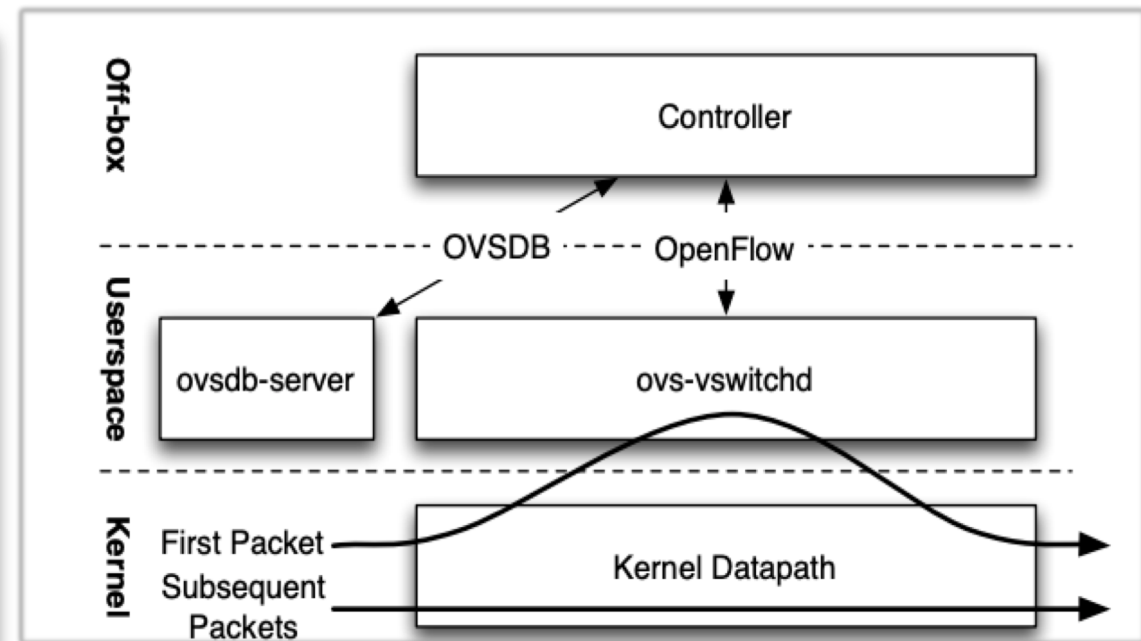
 **Security:** VLAN isolation, traffic filtering

 **Monitoring:** Netflow, sFlow, SPAN, RSPAN

 **QoS:** traffic queuing and traffic shaping

 **Automated Control:** OpenFlow, OVSDB mgmt. protocol

<https://www.openvswitch.org/>



<https://www.openvswitch.org/support/papers/nsdi2015.pdf>

Network Machinery Ingredients



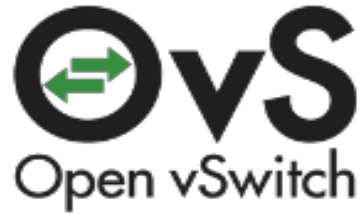
KubeCon



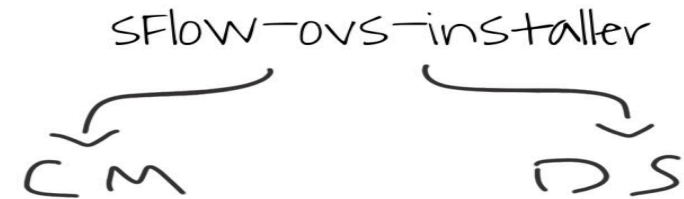
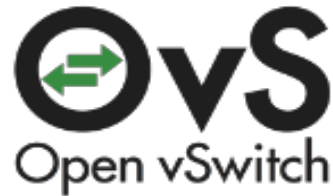
CloudNativeCon

Europe 2019

①



②



③



```
apiVersion: networkmachinery.io/v1alpha1
kind: NetworkMonitor
metadata:
  name: sflow-monitor
spec:
  monitoringEndpoint:
    ip: "10.0.0.10"
    port: "9909"
  flows:
    - name: "elephant-flow"
      keys: "ipsource,ipdestination,tcpsourceport,tcpdestinationport"
      value: "frames"
      log: "true"
    - name: "icmp-flow"
      keys: "ipsource,ipdestination"
      value: "frames"
      log: "true"
  thresholds:
    - name: "ddos"
      metric: "elephant-flow"
      value: 100
      flowName: "elephant-flow"
  eventsConfig:
    maxEvents: "5"
    timeout: "5s"
```

```
apiVersion: networkmachinery.io/v1alpha1
kind: NetworkNotification
metadata:
  name: network-notification-1
spec:
  networkEvent:
    flow:
      name: "some-flow"
      keys: "ipsource,ipdestination,tcpsourceport,tcpdestinationport"
      value: "frames"
    event:
      eventID: 1
      threshold: 20
      value: 1.20
      agent: "1.2.3.4"
      timestamp: "2019-05-21T11:15:00+00:00 in ISO 8601"
      name: "eventName"
      metric: "ddos"
      thresholdID: ""
      dataSource: "2"
```



Demo time

Other CRDs



KubeCon



CloudNativeCon

Europe 2019

```
---
apiVersion: networkmachinery.io/v1alpha1
kind: NetworkModulesValidator
metadata:
  name: module-validator-daemon
spec:
  nodes: all
  net:
    bridge:
      bridge-nf-call-iptables:
    ipv4:
      ip_forward: 1
      arp_proxy:
        interface: eth0
        value: 1
```

```
---
apiVersion: networkmachinery.io/v1alpha1
kind: NetworkPerformanceTest
metadata:
  name: perf-test
spec:
  type: iperf
  clients:
    - kind: pod | service
      name: podName | serviceName
      namespace: namespaceName
      configuration:
        protocol: tcp | udp
        bandwidth: 1000m #Mbps
        bidirectional: true | false
    - kind: node
      name: nodeName
      configuration:
        protocol: tcp | udp
        bandwidth: 1000m #Mbps
        bidirectional: true | false
  servers:
    - kind: ip
      ip: 1.2.3.4
    - kind: pod | service
      name: podName | serviceName
      namespace: namespaceName
```

Summary



KubeCon



CloudNativeCon

Europe 2019

- **Many tools and patterns but no API or common access point.**
- **CRDs enables us to describe and harmonize our APIs.**
- **Network Machinery utilizes CRDs for network troubleshooting**
 - First line of defense (Reachability / Performance / Traffic Shaping)
 - Second line of defense (Network Visibility / Management / Control)
 - Also, sanity checking and network modules validation

Finito / Owatta (終わった)



KubeCon



CloudNativeCon

Europe 2019



@ZaNetworker



@zanetworker

Network Machinery



<https://github.com/networkmachinery>

Gardener



<https://github.com/gardener>



KubeCon



CloudNativeCon

Europe 2019

Extras

CRDs Are For Networking Too



KubeCon



CloudNativeCon

Europe 2019



Multus / CNI-Genie

NetworkAttachmentDefinition

GlobalFelixConfigs

BGPPeers

GlobalBGPCOnfigs

IP Pools

GlobalNetworkPolicies

HostEndpoints

ClusterInformations

...

Calico

And Probably A lot More...

There are already some Networking CRDs out there



Istio

VirtualService

DestinationRule

ServiceEntry

Gateway

...



NetworkPolicy

Reachability & Traffic Shaping

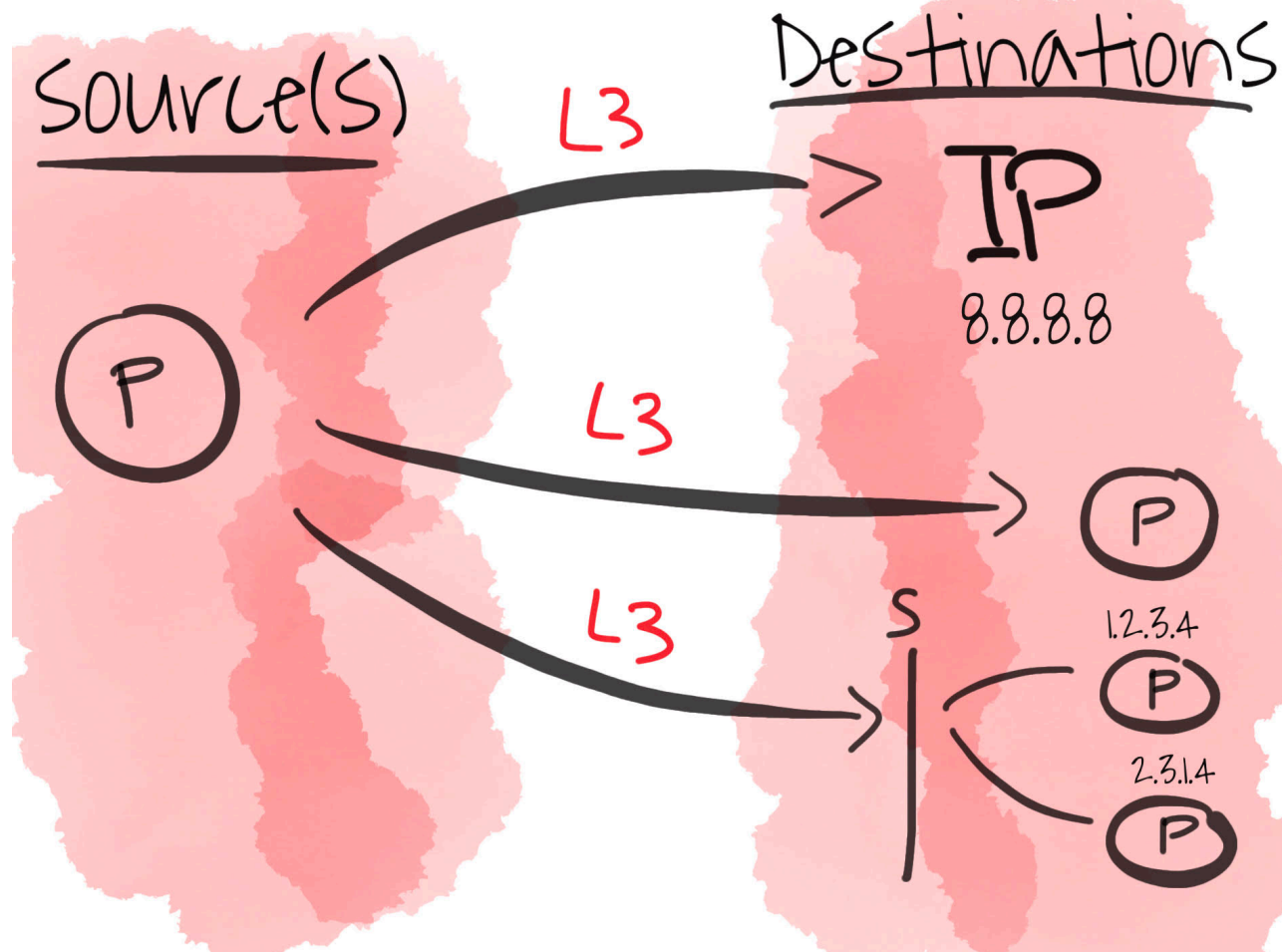


KubeCon



CloudNativeCon

Europe 2019



```
---
apiVersion: networkmachinery.io/v1alpha1
kind: NetworkConnectivityTest
metadata:
  name: smokeping
spec:
  layer: "3"
  source:
    name: "kube-apiserver-kind-kubecon2019-
control-plane"
    namespace: "kube-system"
    container: ""
  destinations:
    - kind: pod
      namespace: default
      name: somepod
    - kind: pod
      namespace: default
      name: kubecon-pod
    - kind: ip
      ip: "8.8.8.8"
    - kind: service
      namespace: default
      name: kubernetes
```

Reachability & Traffic Shaping

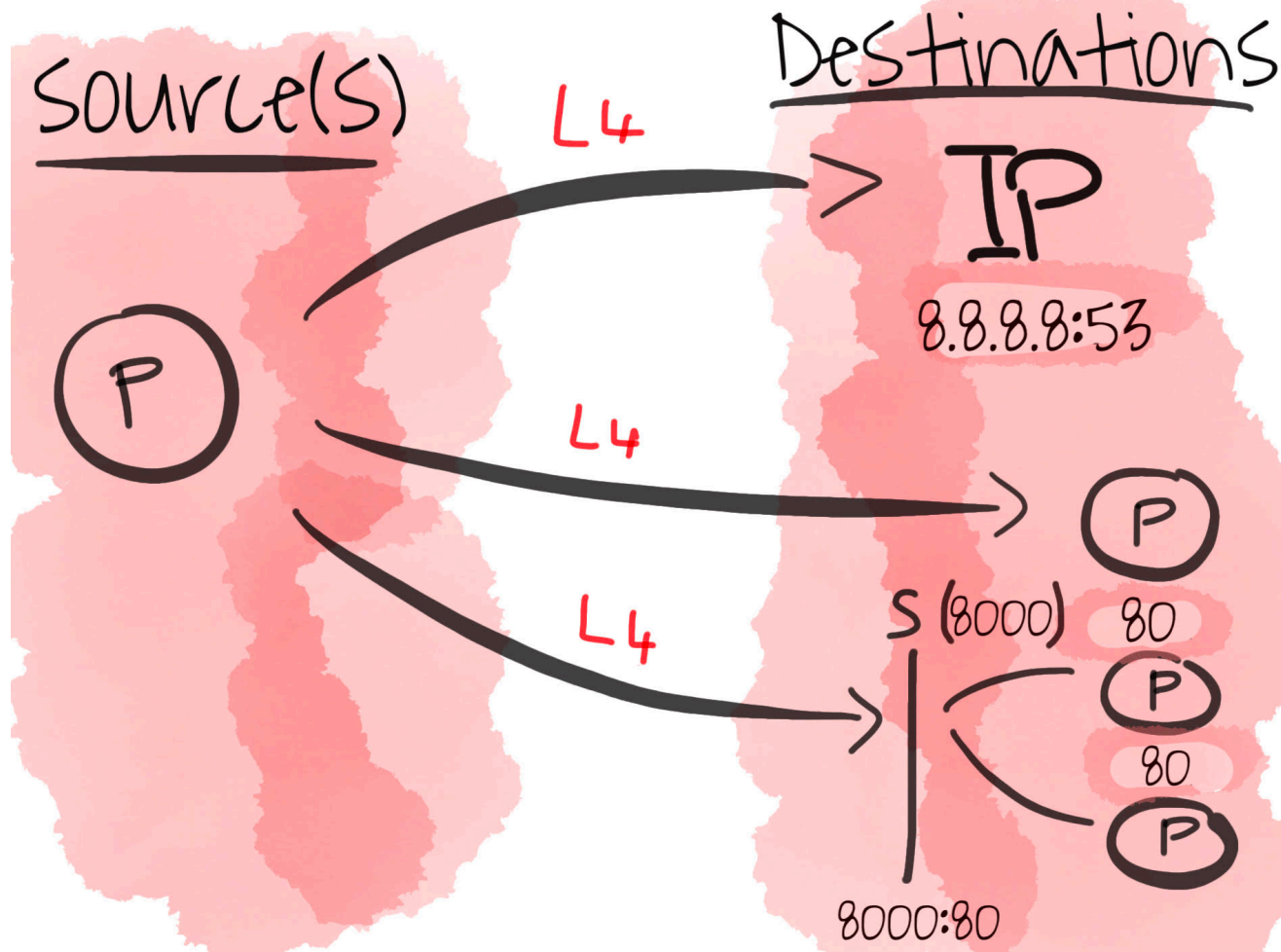


KubeCon



CloudNativeCon

Europe 2019



```
---
apiVersion: networkmachinery.io/v1alpha1
kind: NetworkConnectivityTest
metadata:
  name: port-test
spec:
  layer: "4"
  source:
    name: "kube-apiserver-kind-kubecon2019-
control-plane"
    namespace: "kube-system"
    container: ""
  destinations:
    - kind: pod
      namespace: kube-system
      name: kubecon-pod
      port: "51"
    - kind: service
      namespace: default
      name: kubernetes
      port: "443"
```

Reachability & Traffic Shaping

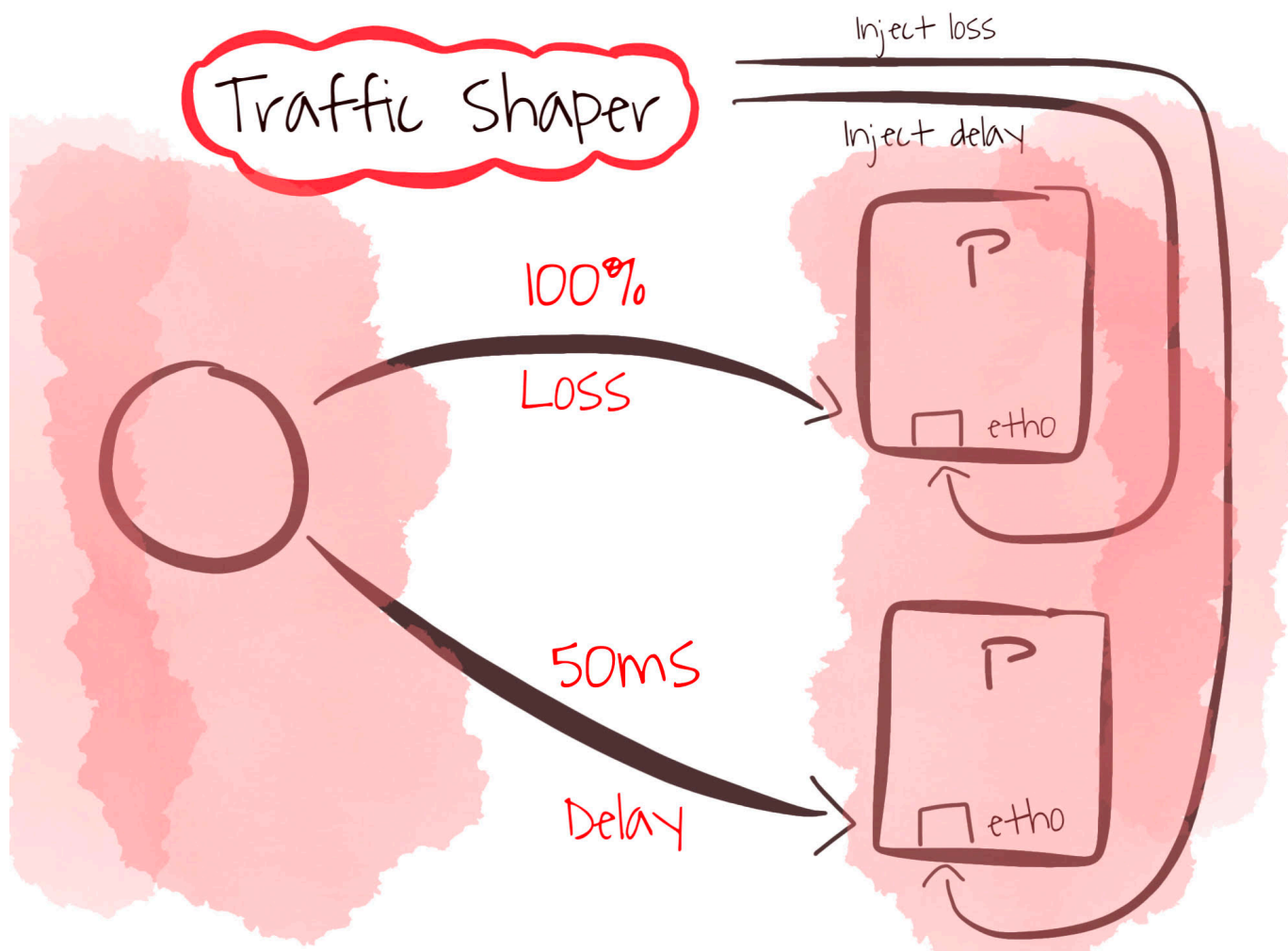


KubeCon



CloudNativeCon

Europe 2019



```
---
apiVersion:
networkmachinery.io/v1alpha1
kind: NetworkTrafficShaper
metadata:
  name: inject-delay | inject-loss
spec:
  targets:
    - kind: pod | selector
      name: podName
      namespace: namespaceName
      targetSelector:
        matchLabels:
          app: demo-kubecon
  configuration:
    type: delay | loss
    device: eth0
    value: 200ms | 90%
```

SDN / OpenFlow / sFlow Capsule



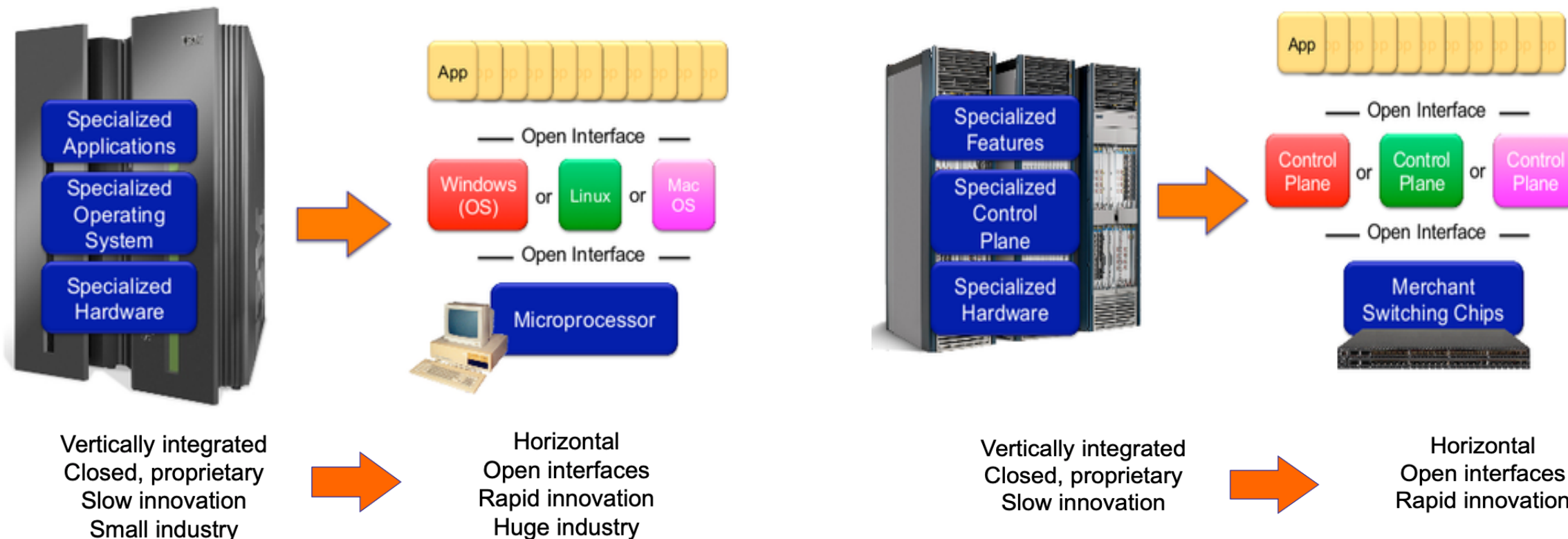
KubeCon



CloudNativeCon

Europe 2019

- SDN is about the Separation of the Control-Plane and Data-Plane
- An early effort for programmable networks



Network Machinery In Action



KubeCon



CloudNativeCon

Europe 2019



```
---
apiVersion: networkmachinery.io/v1alpha1
kind: NetworkMonitor
metadata:
  name: sflow-monitor
spec:
  monitoringEndpoint:
    ip: "10.0.0.10"
    port: "8008"
  flows:
    - name: "elephant-flow"
      keys: "ipsource,ipdestination,tcpsourceport,tcpdestinationport"
      value: "frames"
      log: "true"
    - name: "icmp-flow"
      keys: "ipsource,ipdestination"
      value: "frames"
      log: "true"
  thresholds:
    - name: "ddos"
      metric: "elephant-flow"
      value: 100
      flowName: "elephant-flow"
  eventsConfig:
    maxEvents: "5"
    timeout: "60"
```

sFlow



Network Machinery In Action



KubeCon



CloudNativeCon

Europe 2019



```
---
apiVersion: networkmachinery.io/v1alpha1
kind: NetworkNotification
metadata:
  name: network-notification-1
spec:
  networkEvent:
    flow:
      name: "some-flow"
      keys: "ipsource,ipdestination,tcpsourceport,tcpdestinationport"
      value: "frames"
    event:
      eventID: 1
      threshold: 20
      value: 1.20
      agent: "1.2.3.4"
      timestamp: "2019-05-21T11:15:00+00:00 in ISO 8601"
      name: "eventName"
      metric: "ddos"
      thresholdID: ""
      dataSource: "2"
```



Network Machinery In Action



KubeCon



CloudNativeCon

Europe 2019



sFlow

OpenFlow

