Replacing iptables with eBPF in Kubernetes with Cilium

Cilium, eBPF, Envoy, Istio, Hubble

Michal Rostecki
Software Engineer
mrostecki@suse.com
mrostecki@opensuse.org

Swaminathan Vasudevan
Software Engineer
svasudevan@suse.com

What’s wrong with iptables?

What's wrong with legacy iptables?

IPtables runs into a couple of significant problems:

- Iptables updates must be made by recreating and updating all rules in a single transaction.
- Implements chains of rules as a linked list, so all operations are O(n).
- The standard practice of implementing access control lists (ACLs) as implemented by iptables was to use sequential list of rules.
- It's based on matching IPs and ports, not aware about L7 protocols.
- Every time you have a new IP or port to match, rules need to be added and the chain changed.
- Has high consumption of resources on Kubernetes.

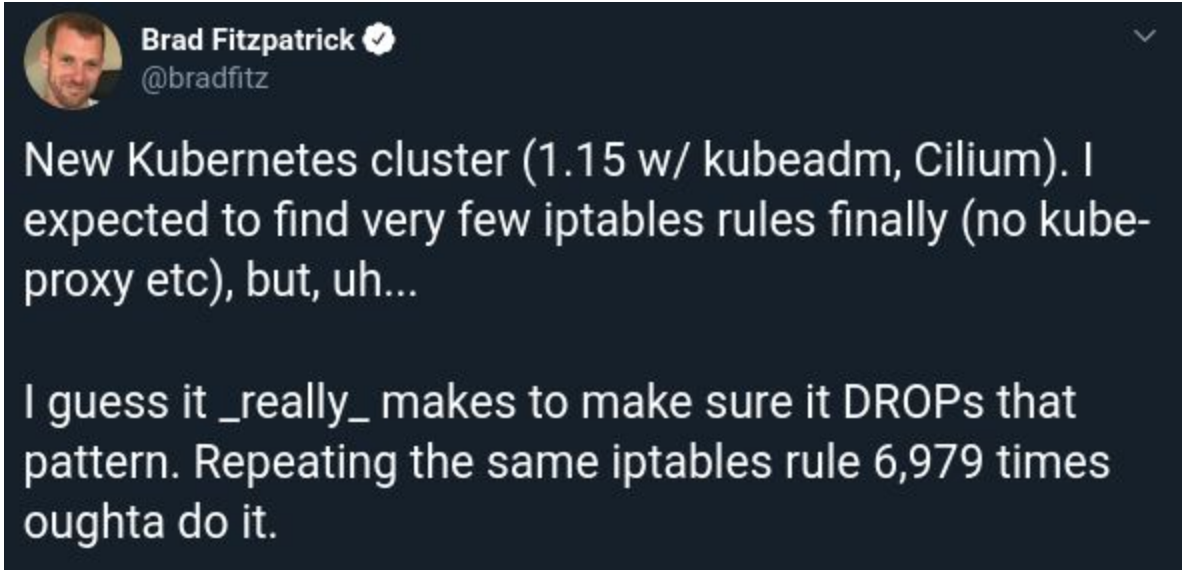
Based on the above mentioned issues under heavy traffic conditions or in a system that has a large number of changes to iptable rules the performance degrades.

Measurements show unpredictable latency and reduced performance as the number of services grows.

Kubernetes uses iptables for...

- kube-proxy - the component which implements Services and load balancing by DNAT iptables rules
- the most of CNI plugins are using iptables for Network Policies

And it ends up like that

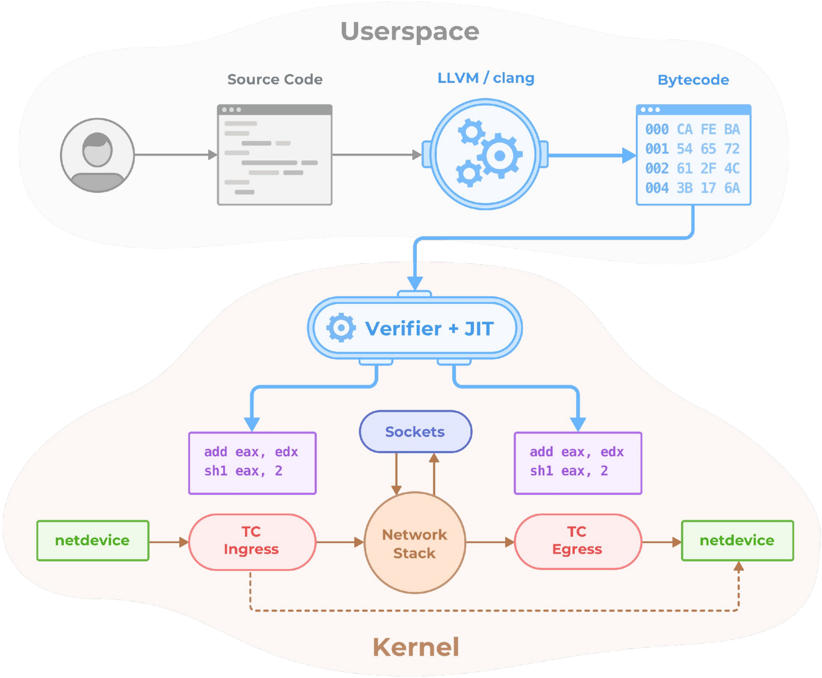

Brad Fitzpatrick
@bradfitz

New Kubernetes cluster (1.15 w/ kubeadm, Cilium). I expected to find very few iptables rules finally (no kube-proxy etc), but, uh...

I guess it _really_ makes to make sure it DROPs that pattern. Repeating the same iptables rule 6,979 times oughta do it.

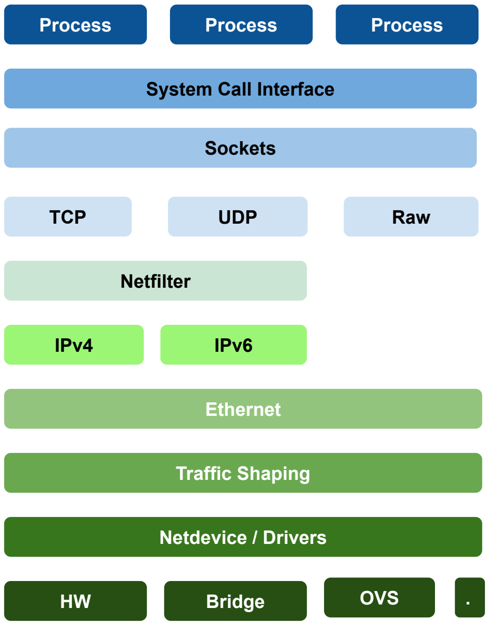
```
root@kc1b:~# iptables -L -v -n | more
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
  76M   12G KUBE-FIREWALL  all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
  76M   12G KUBE-FIREWALL  all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain KUBE-FIREWALL (2 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
    0     0 DROP       all  --  *      *       0.0.0.0/0            0.0.0.0/0            mark match 0x8000/0x8000 /* kubernetes firewall for dropping marked packets */
```

What is BPF?



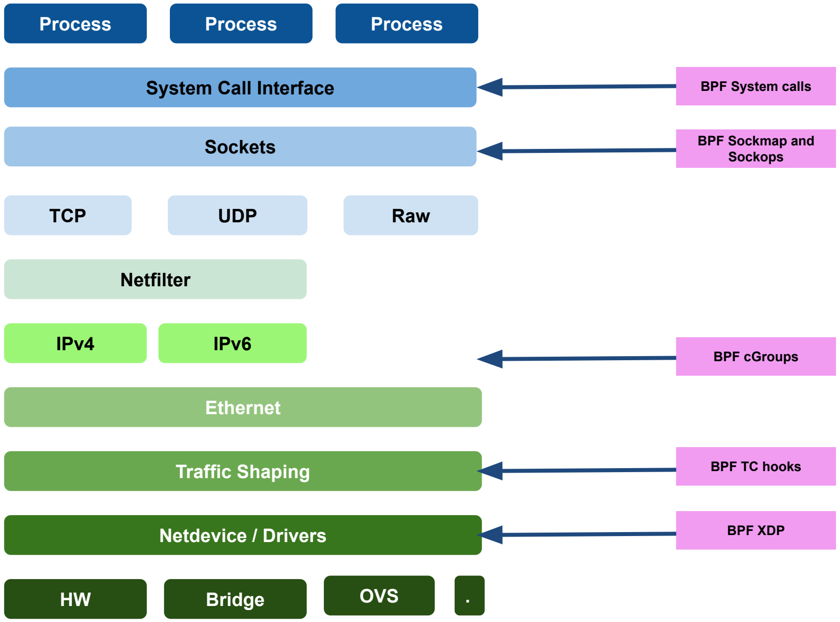
Userspace
Source Code
LLVM / clang
Bytecode
000 CA FE BA
001 54 65 72
002 61 2F 4C
004 3B 17 6A
Verifier + JIT
add eax, edx
shl eax, 2
Sockets
add eax, edx
shl eax, 2
netdevice
TC Ingress
Network Stack
TC Egress
netdevice
Kernel

Linux Network Stack

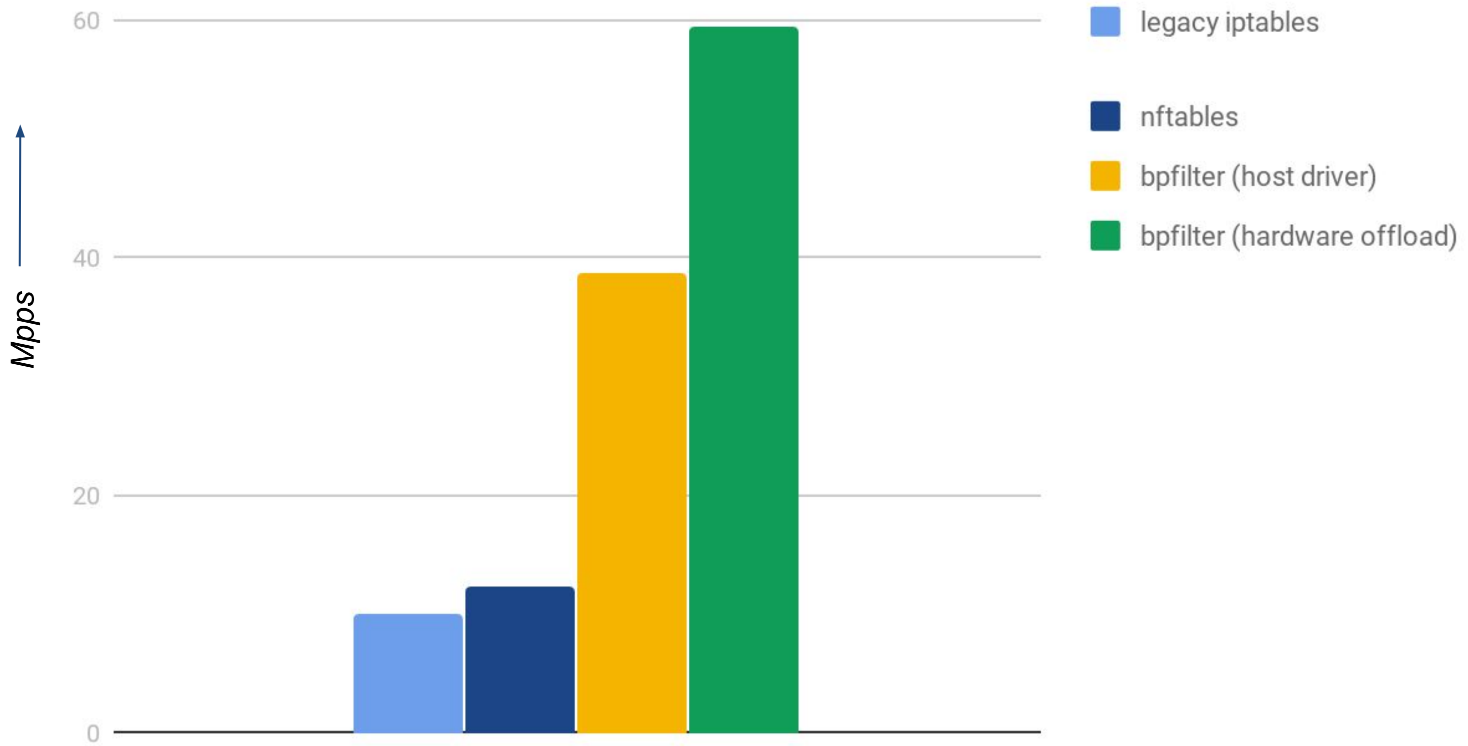


- The Linux kernel stack is split into multiple abstraction layers.
- Strong userspace API compatibility in Linux for years.
- This shows how complex the linux kernel is and its years of evolution.
- This cannot be replaced in a short term.
- Very hard to bypass the layers.
- Netfilter module has been supported by linux for more than two decades and packet filtering has to applied to packets that moves up and down the stack.

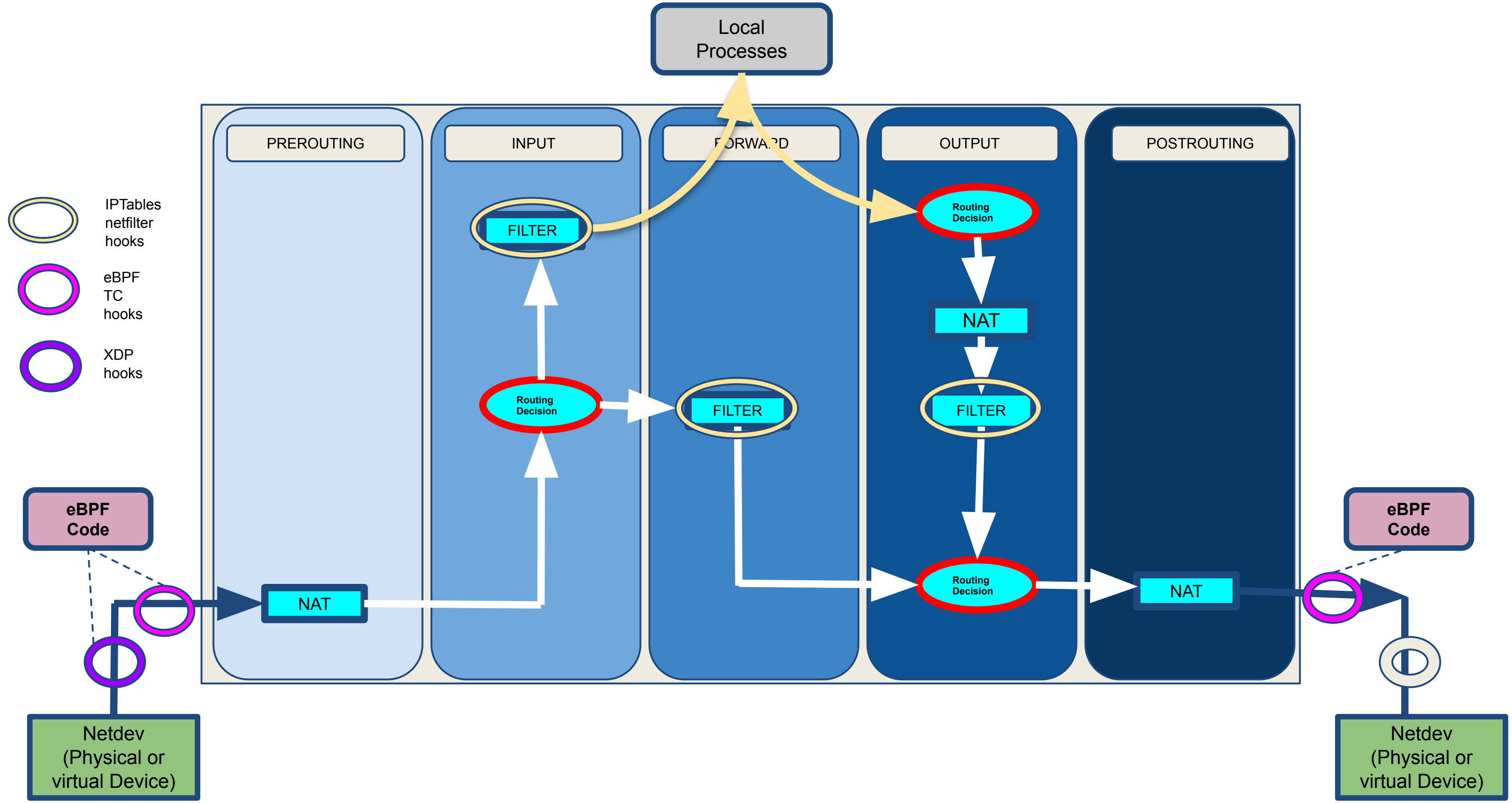
BPF kernel hooks



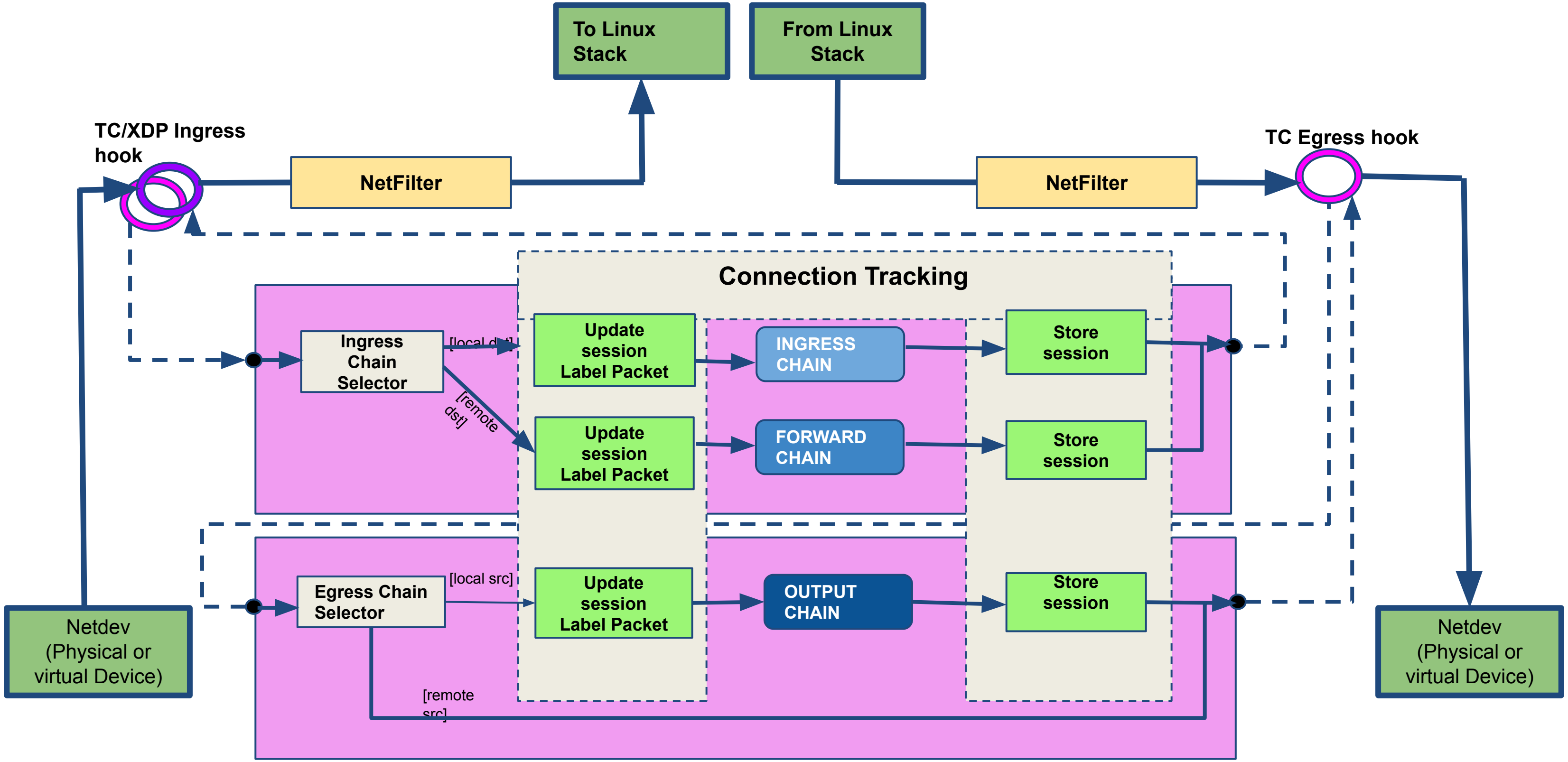
BPF goes into firewalls



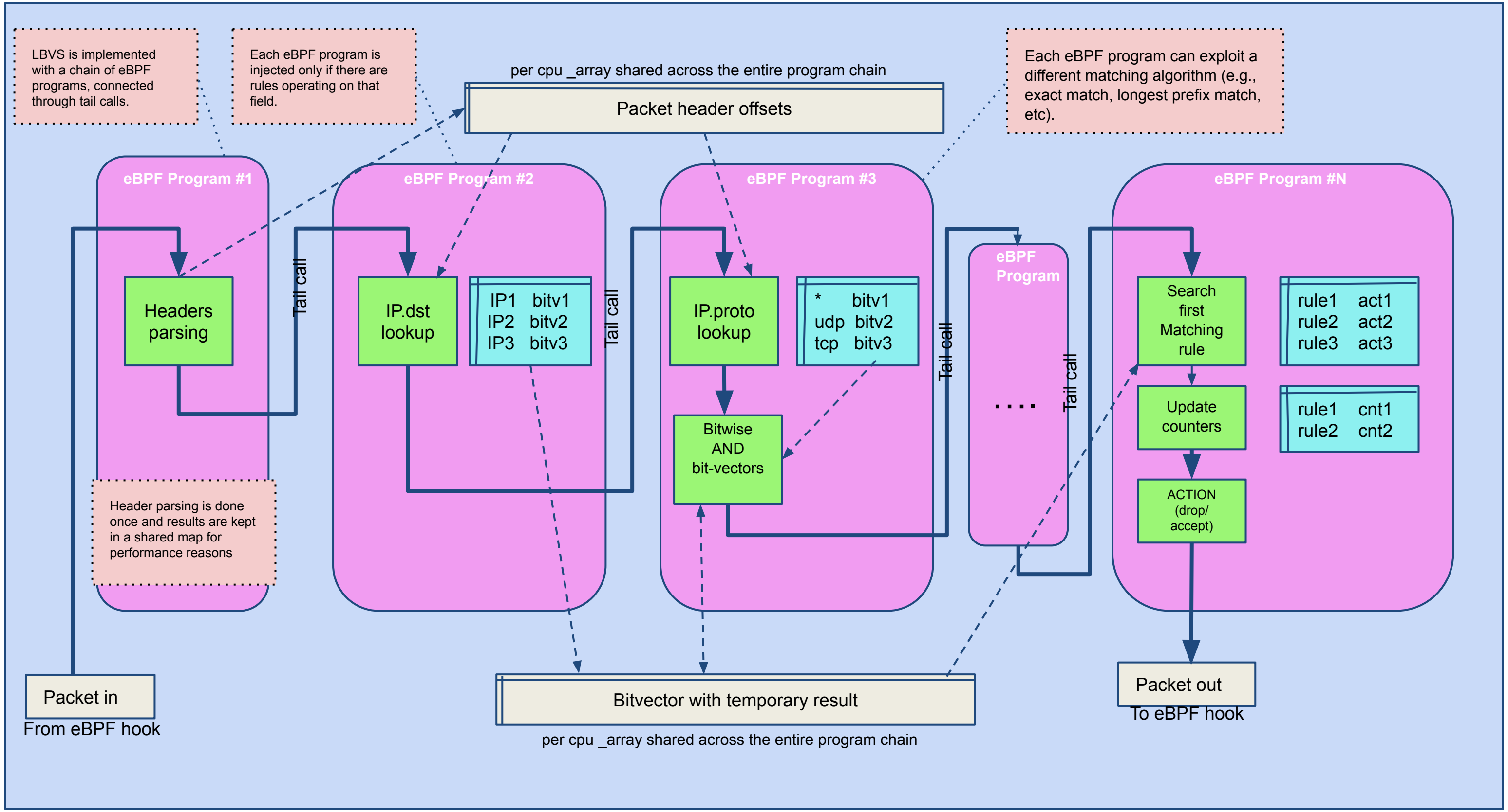
BPF replaces IPtables



BPF based filtering architecture



BPF based tail calls



BPF goes into...

- Load balancers - **katran**
- **perf**
- **systemd**
- **Suricata**
- **Open vSwitch** - AF_XDP
- And many many others

BPF is used by...

Google

NETFLIX

cilium

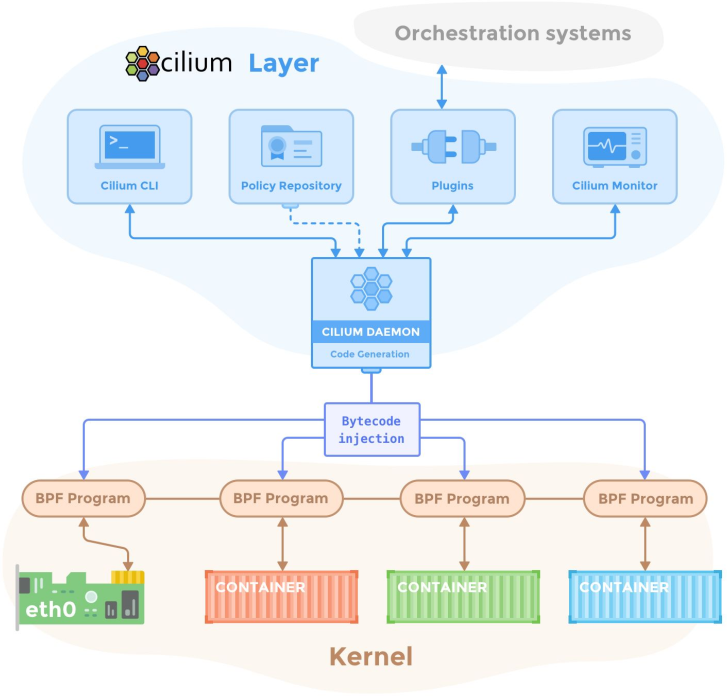
SUSE
CaaS
Platform

Falco

CLOUDFLARE®

Cilium

What is Cilium?



CNI Functionality

CNI is a CNCF (Cloud Native Computing Foundation) project for Linux Containers
It consists of specification and libraries for writing plugins.
Only care about networking connectivity of containers

- ADD/DEL

General container runtime considerations for CNI:

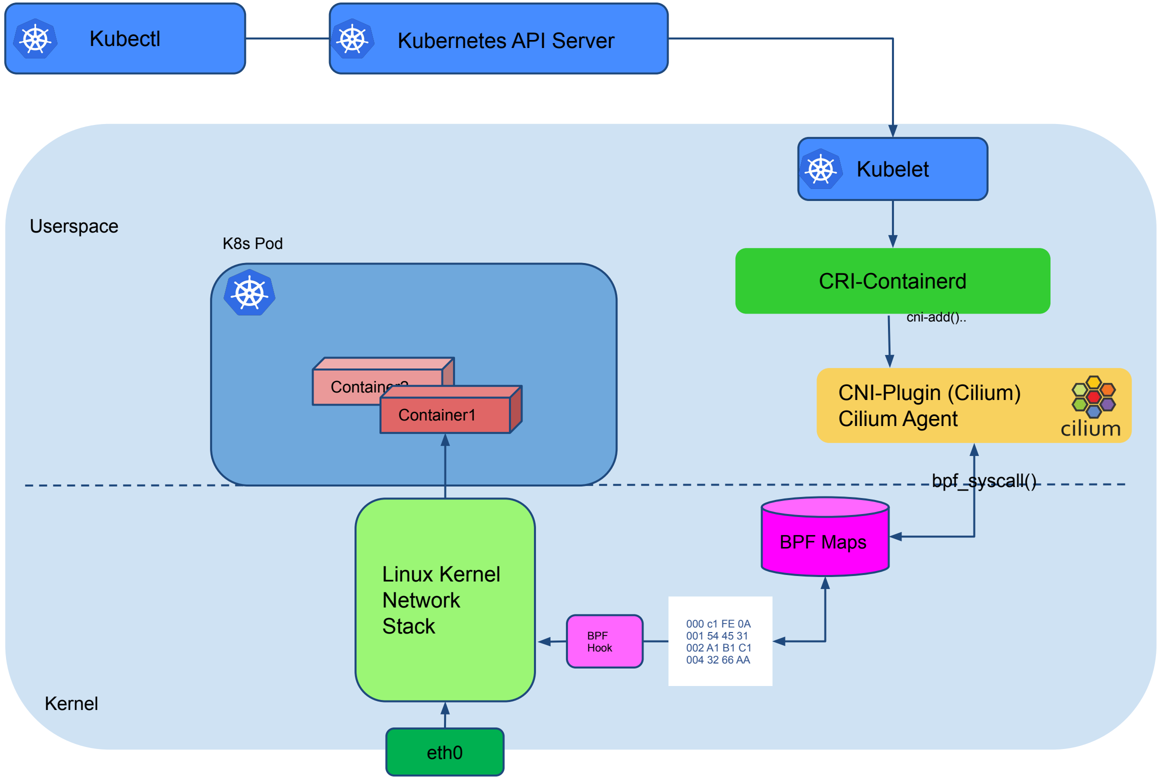
The container runtime must

- create a new network namespace for the container before invoking any plugins
- determine the network for the container and add the container to the each network by calling the corresponding plugins for each network
- not invoke parallel operations for the same container.
- order ADD and DEL operations for a container, such that ADD is always eventually followed by a corresponding DEL.
- not call ADD twice (without a corresponding DEL) for the same (network name, container id, name of the interface inside the container).

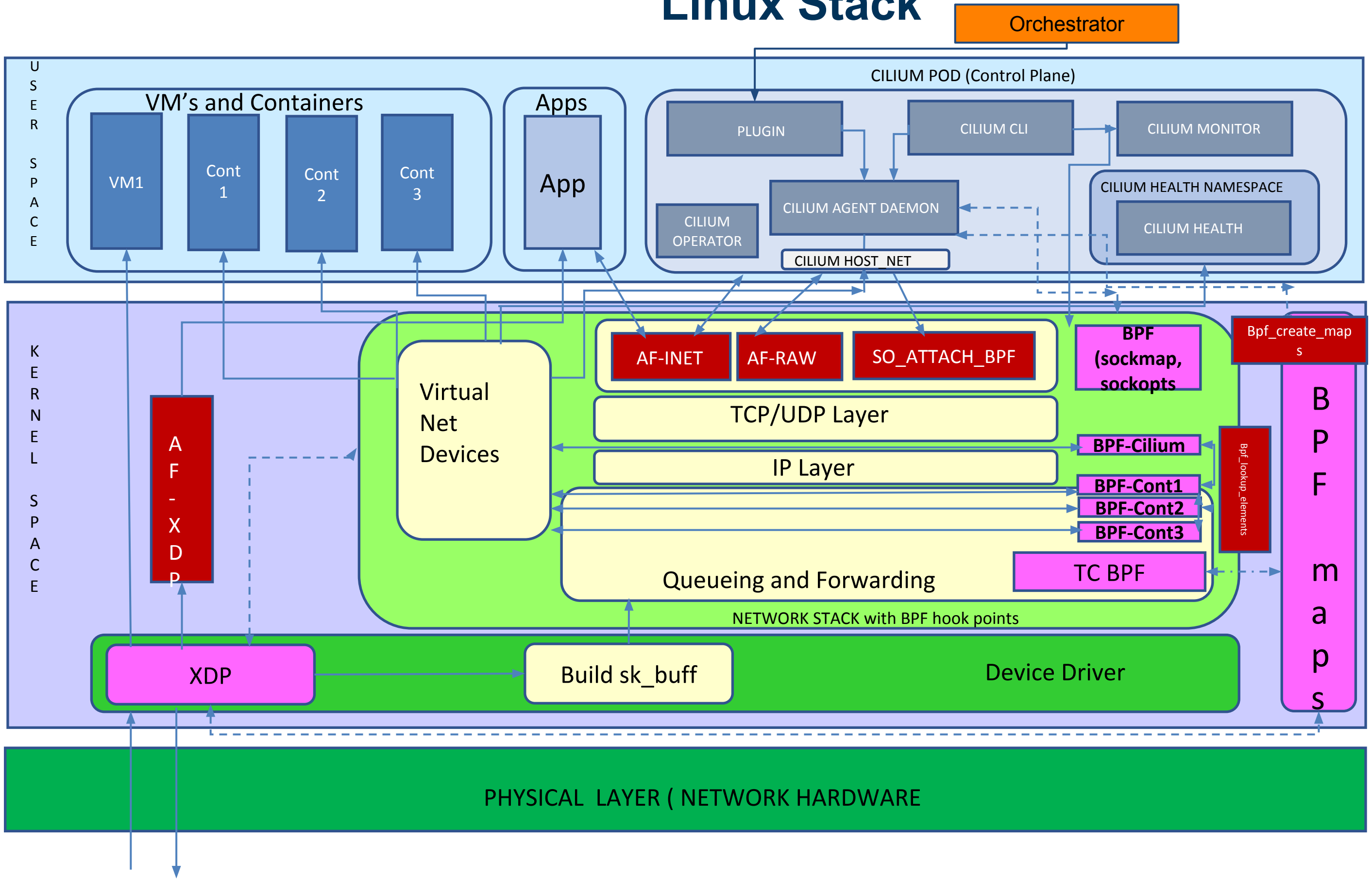
When CNI ADD call is invoked it tries to add the network to the container with respective veth pairs and assigning IP address from the respective IPAM Plugin or using the Host Scope.

When CNI DEL call is invoked it tries to remove the container network, release the IP Address to the IPAM Manager and cleans up the veth pairs.

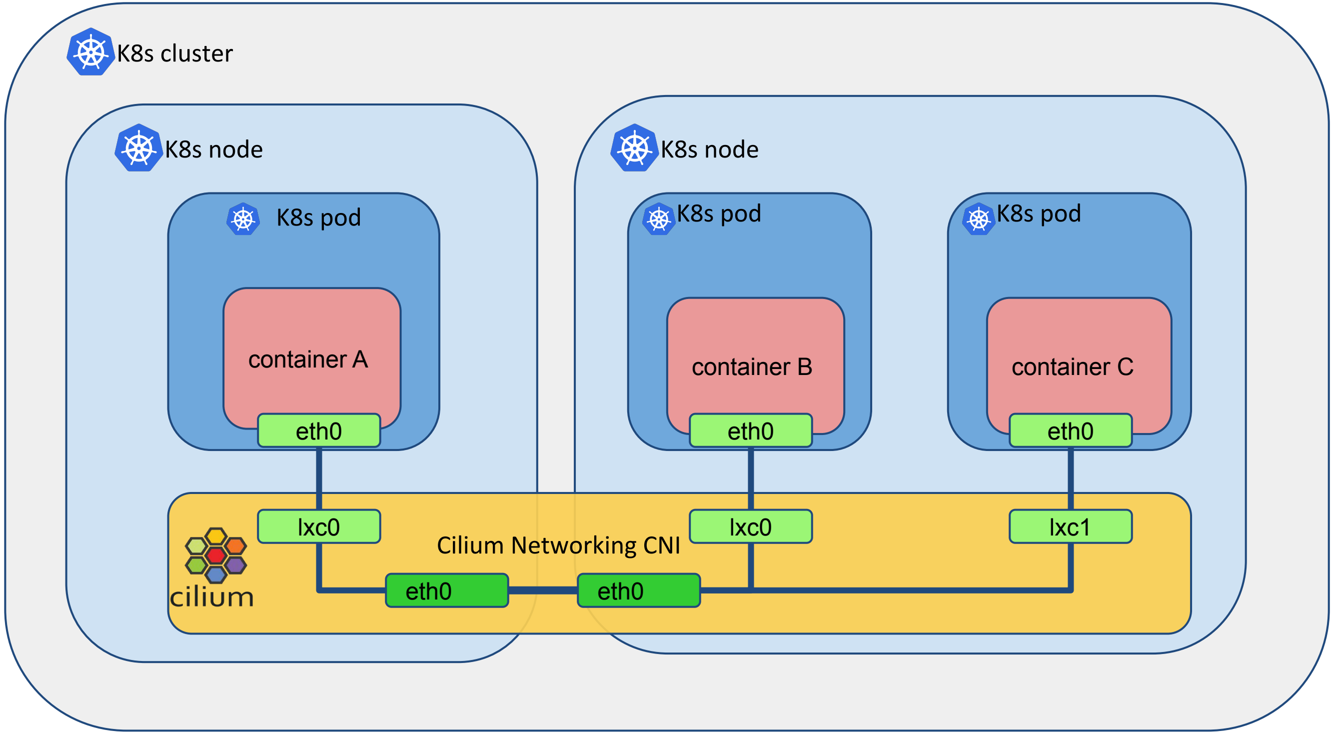
Cilium CNI Plugin control Flow



Cilium Components with BPF hook points and BPF maps shown in Linux Stack



Cilium as CNI Plugin

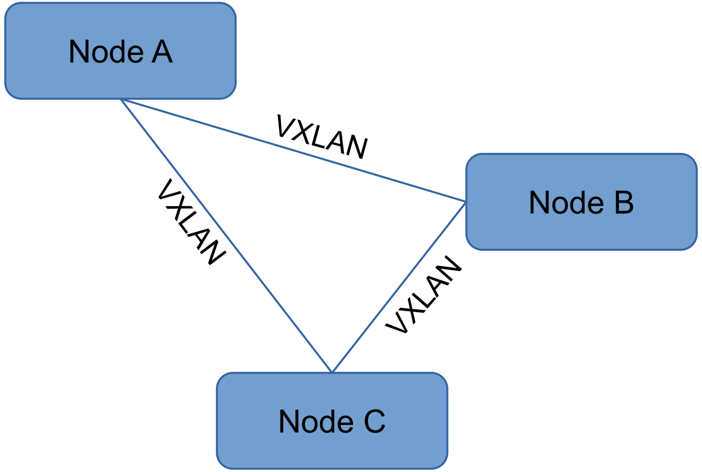


Networking modes

Encapsulation

Use case:

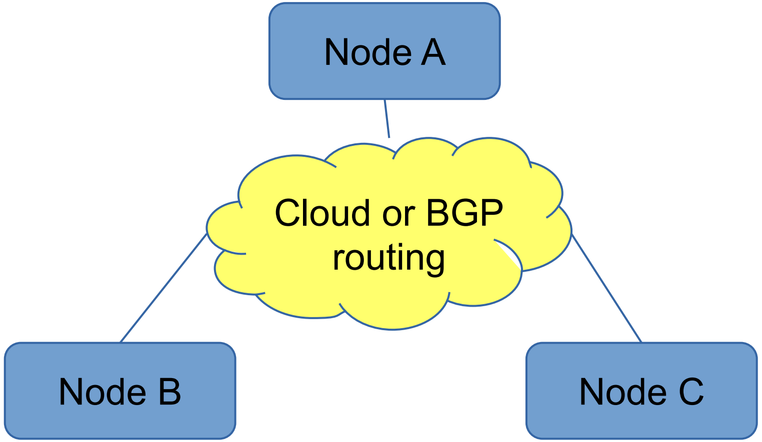
Cilium handling routing between nodes



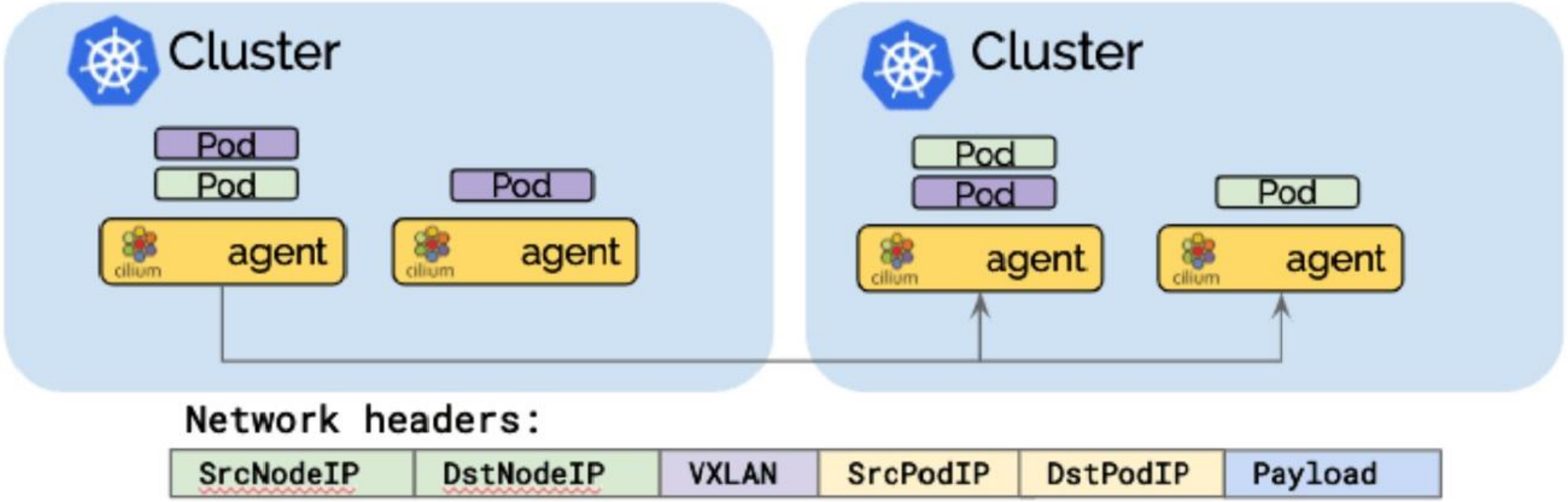
Direct routing

Use case:

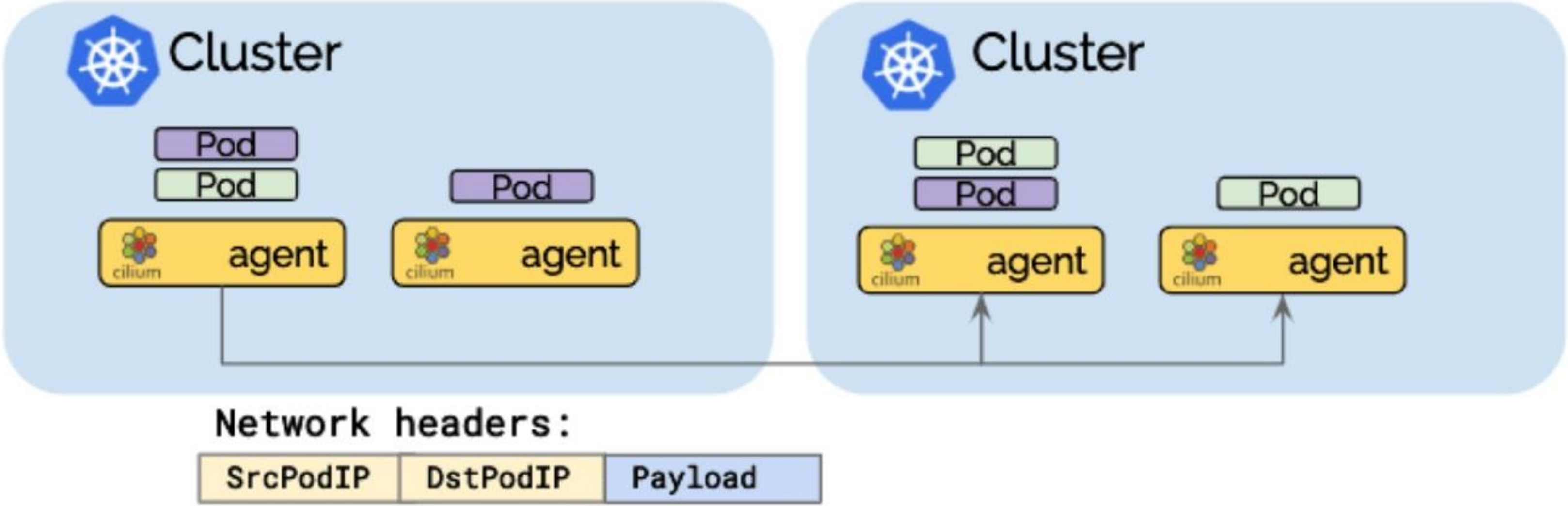
Using cloud provider routers, using BGP routing daemon



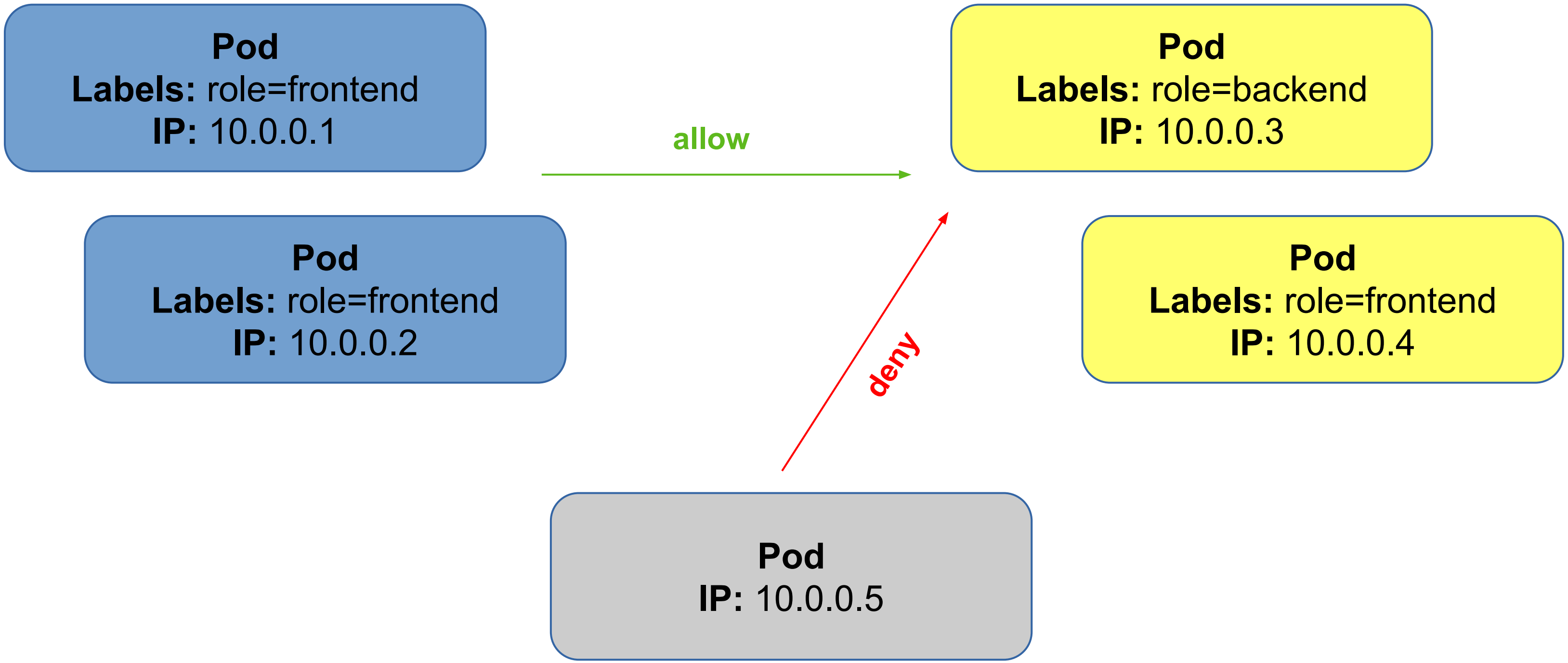
Pod IP Routing - Overlay Routing (Tunneling mode)



Pod IP Routing - Direct Routing Mode



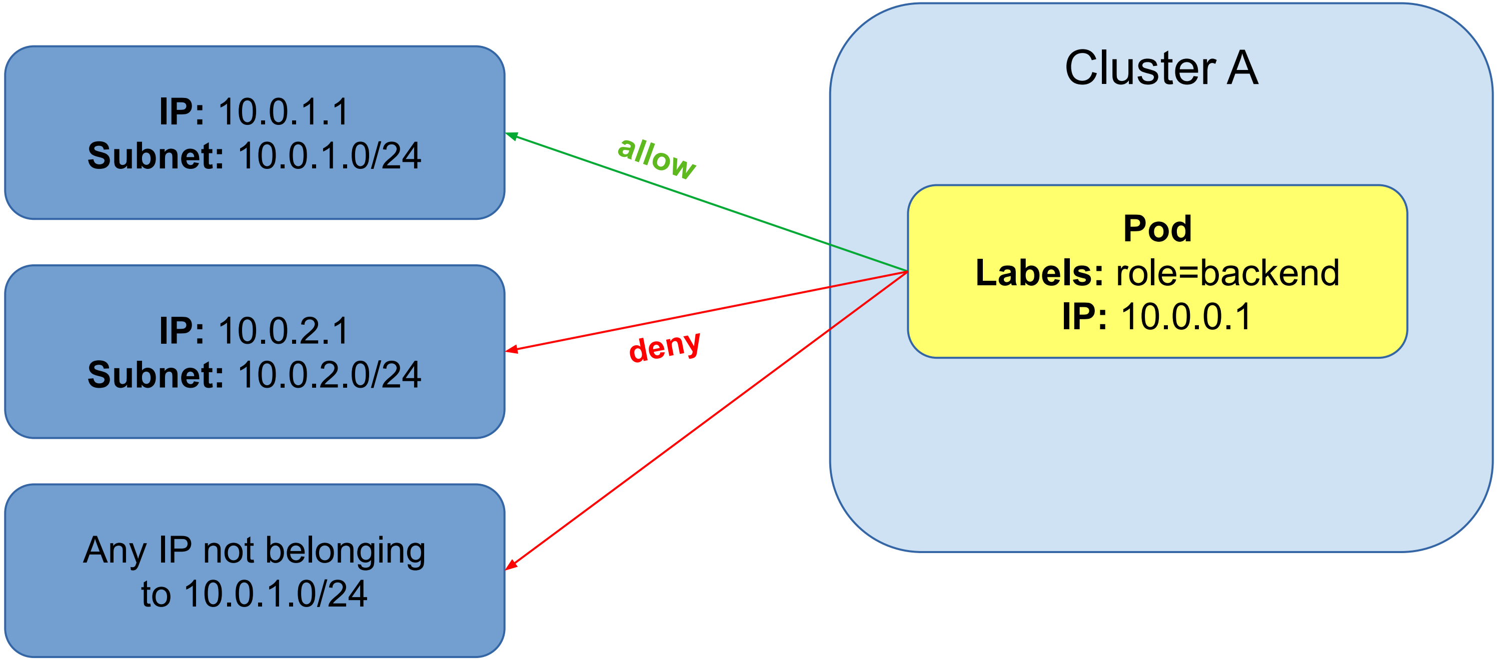
L3 filtering – label based, ingress



L3 filtering – label based, ingress

```yaml
apiVersion: "cilium.io/v2"
kind: CiliumNetworkPolicy
description: "Allow frontends to access backends"
metadata:
  name: "frontend-backend"
spec:
  endpointSelector:
    matchLabels:
      role: backend
  ingress:
  - fromEndpoints:
    - matchLabels:
        class: frontend
```

L3 filtering – CIDR based, egress



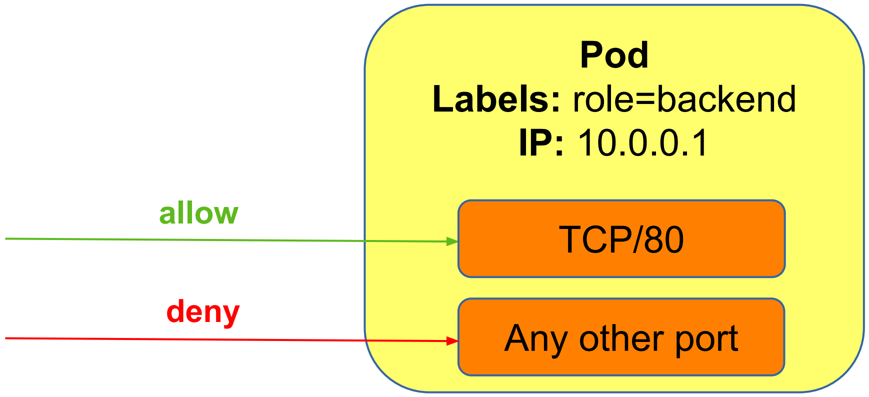
L3 filtering – CIDR based, egress

```
apiVersion: "cilium.io/v2"
kind: CiliumNetworkPolicy
description: "Allow backends to access 10.0.1.0/24"
metadata:
  name: "frontend-backend"
spec:
  endpointSelector:
    matchLabels:
      role: backend
  egress:
  - toCIDR:
    - IP: “10.0.1.0/24”
```

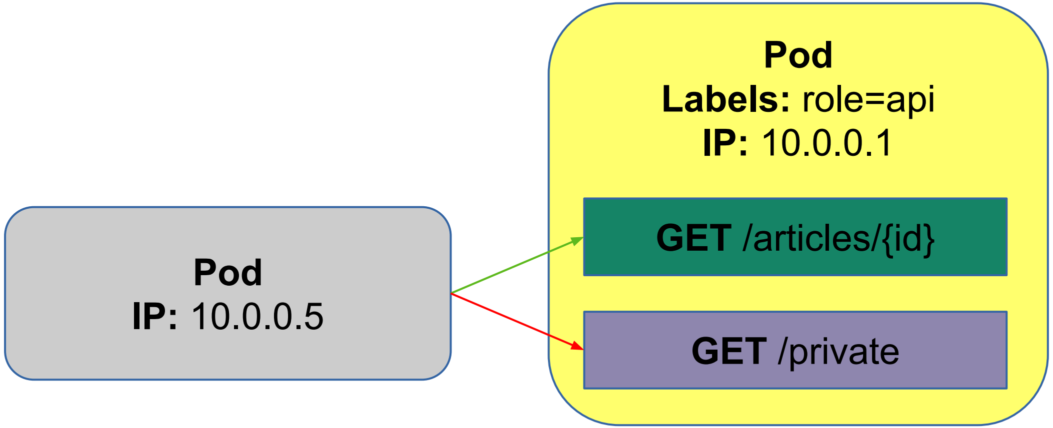
L4 filtering

```
apiVersion: "cilium.io/v2"
kind: CiliumNetworkPolicy
description: "Allow to access backends only on TCP/80"
metadata:
  name: "frontend-backend"
spec:
  endpointSelector:
    matchLabels:
      role: backend
  ingress:
  - toPorts:
    - ports:
      - port: "80"
        protocol: "TCP"
```

L4 filtering



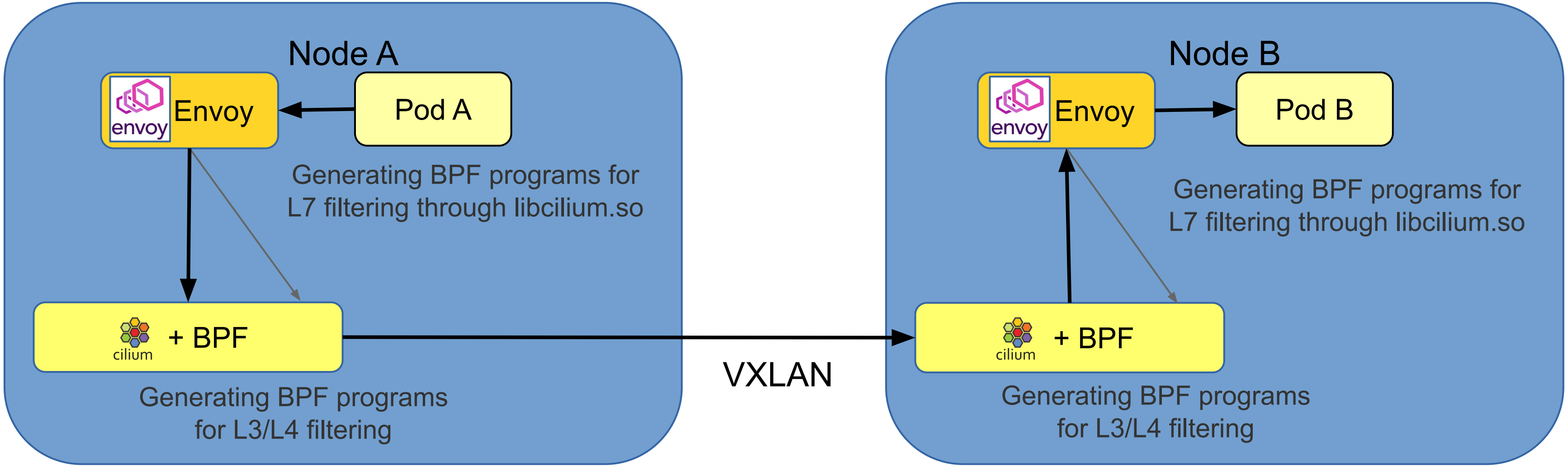
L7 filtering – API Aware Security



L7 filtering – API Aware Security

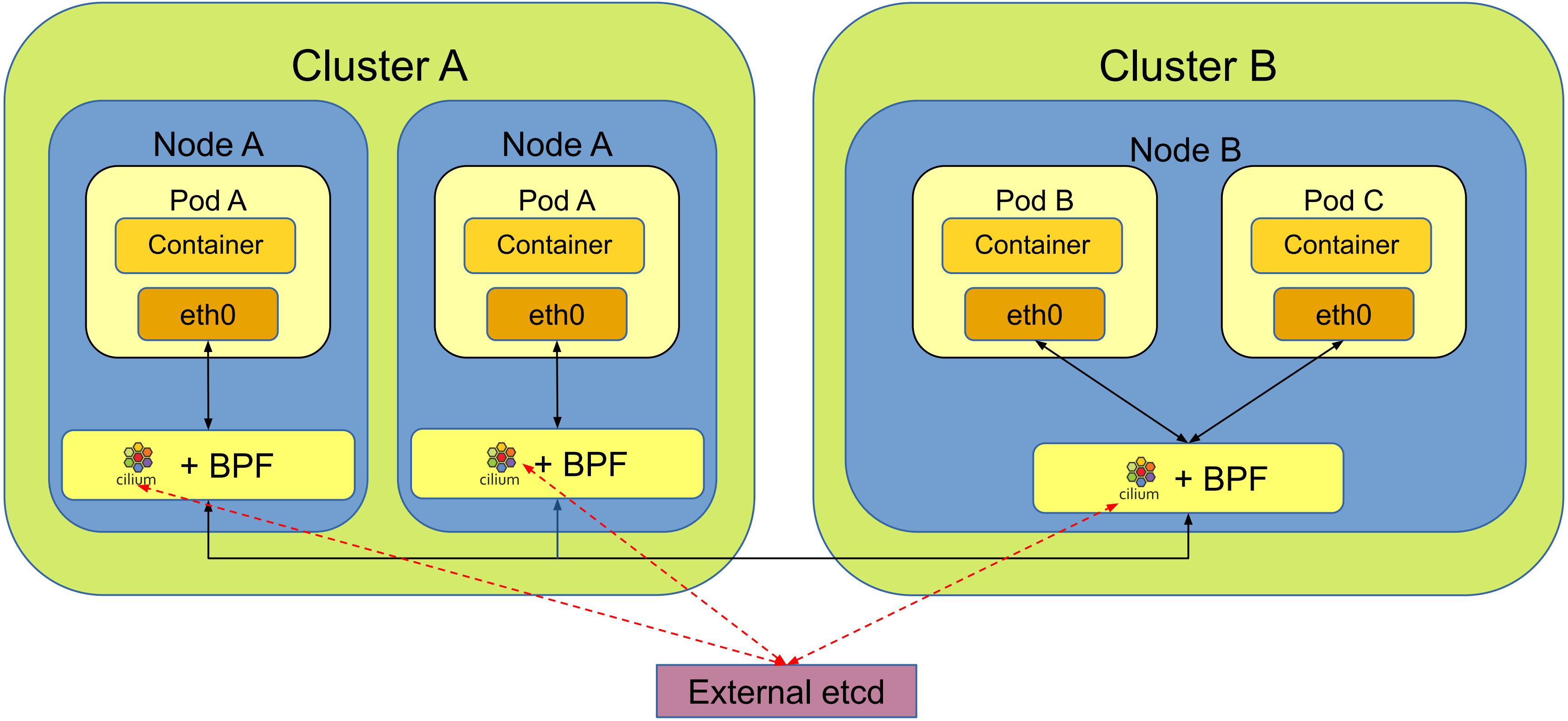
```
apiVersion: "cilium.io/v2"
kind: CiliumNetworkPolicy
description: "L7 policy to restict access to specific HTTP endpoints"
metadata:
  name: "frontend-backend"
  endpointSelector:
    matchLabels:
      role: backend
  ingress:
  - toPorts:
    - ports:
      - port: “80”
        protocol: “TCP”
      rules:
        http:
        - method: "GET"
          path: "/article/$"
```

Standalone proxy, L7 filtering

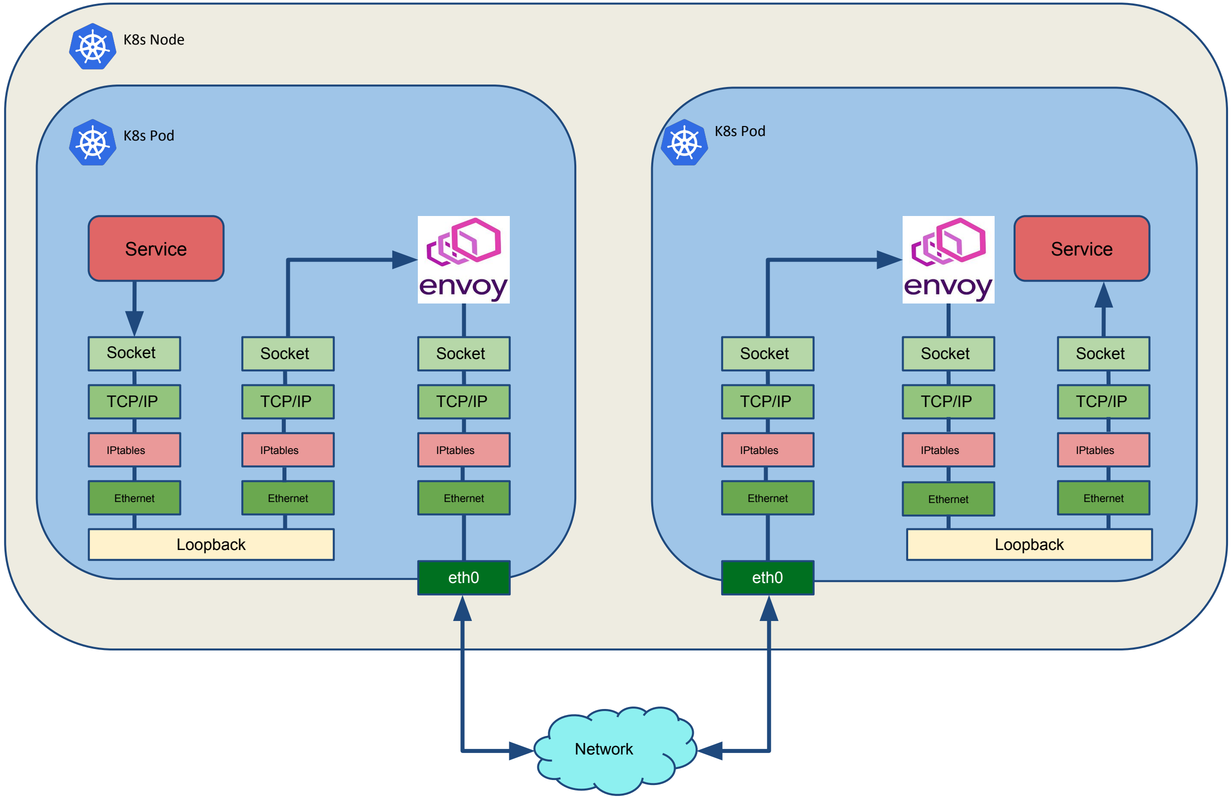


Features

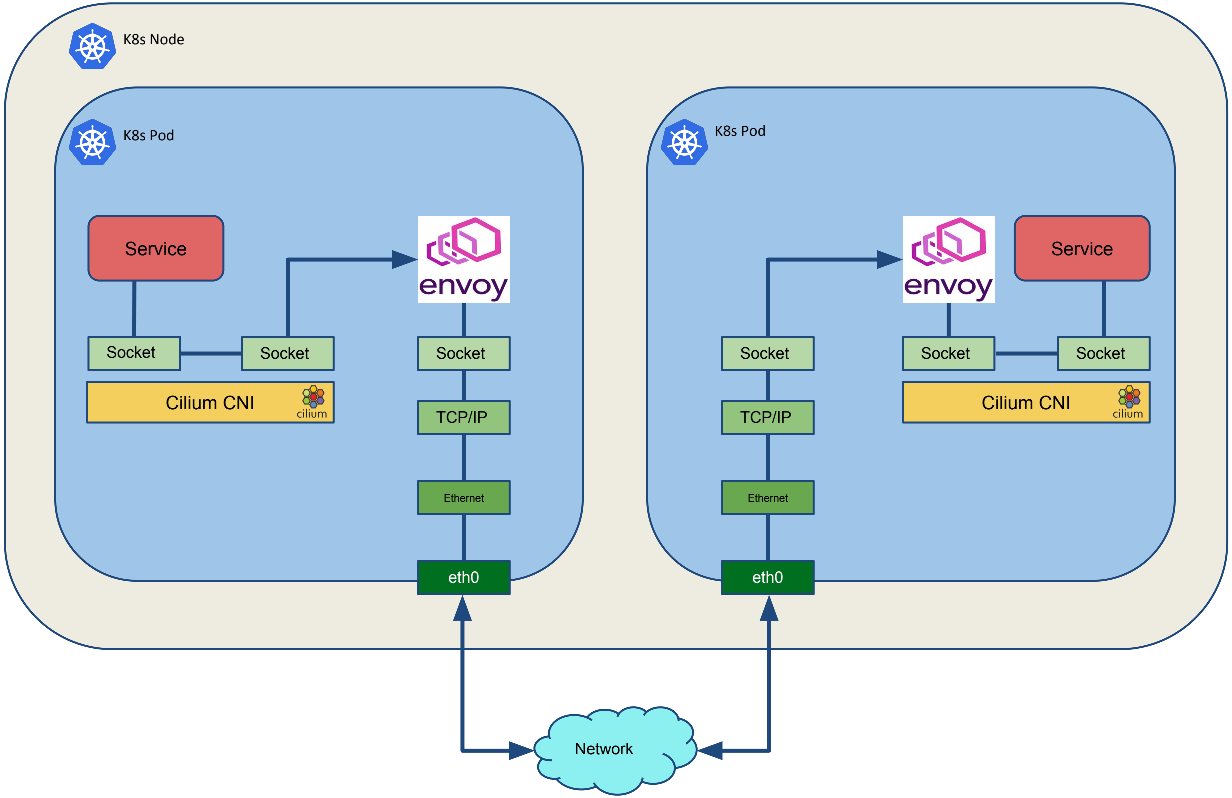
Cluster Mesh



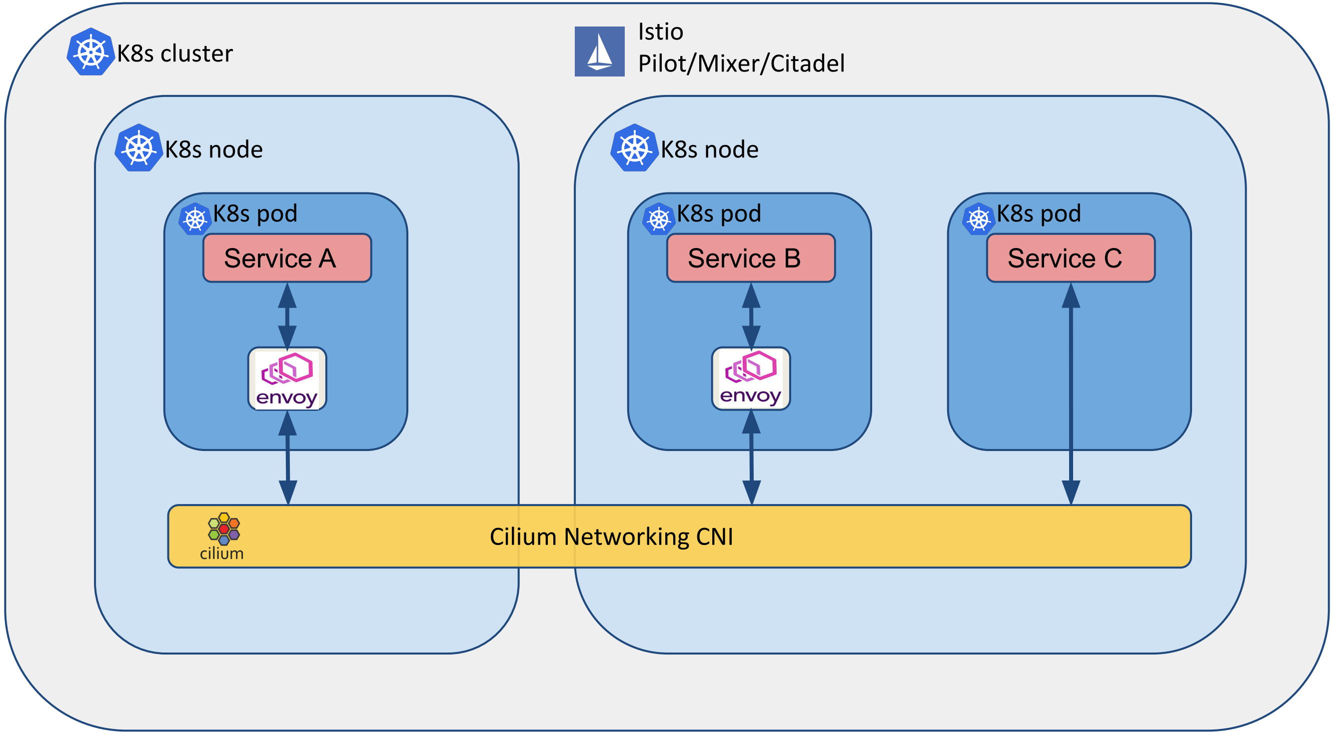
Istio (Transparent Sidecar injection) without Cilium



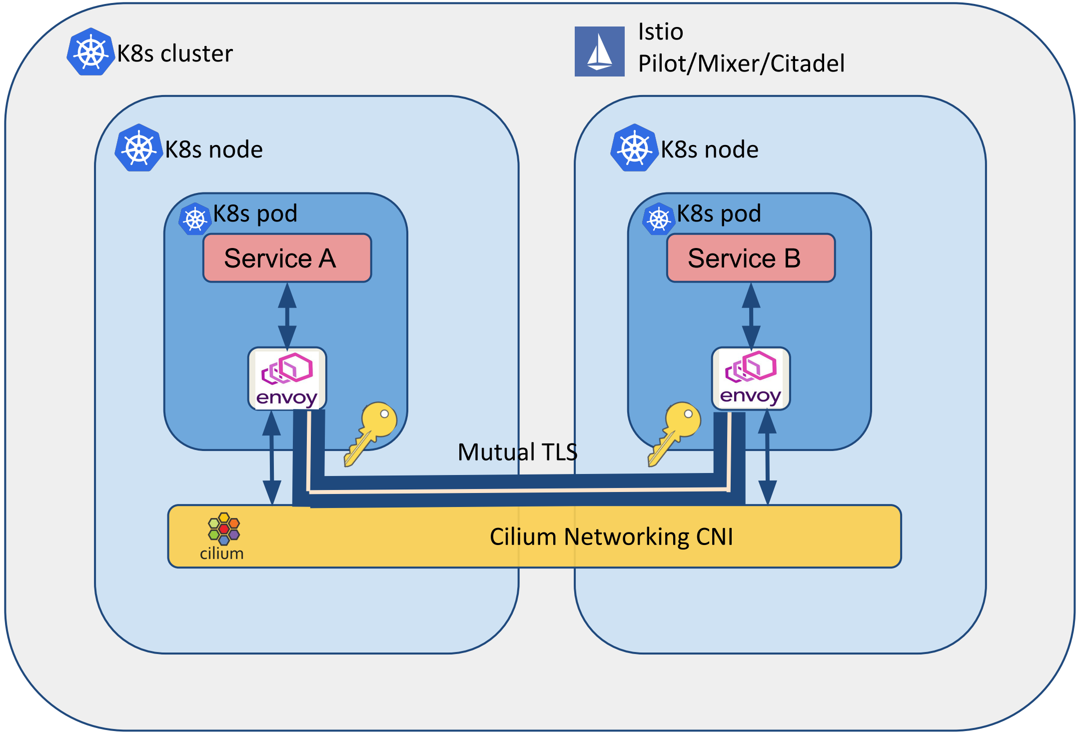
Istio with cilium and sockmap



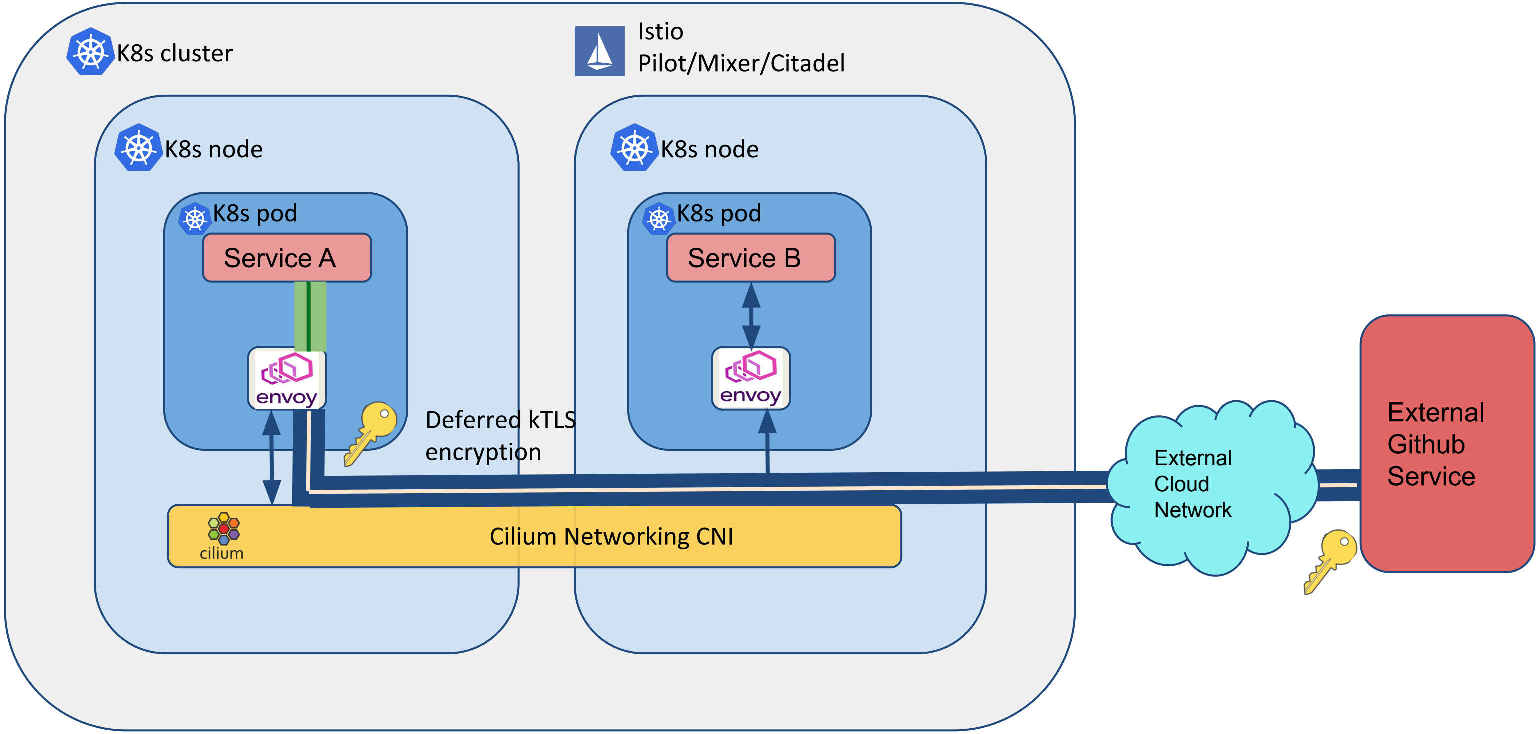
Istio


K8s cluster
Istio
Pilot/Mixer/Citadel
K8s node
K8s node
K8s pod
Service A
envoy
K8s pod
Service B
envoy
K8s pod
Service C
cilium
Cilium Networking CNI

Istio - Mutual TLS



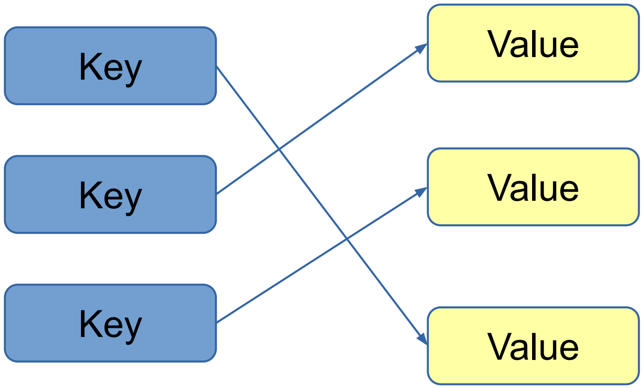
Istio - Deferred kTLS



Kubernetes Services

BPF, Cilium

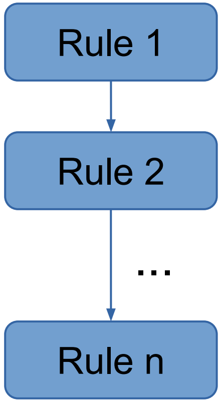
- Hash table.



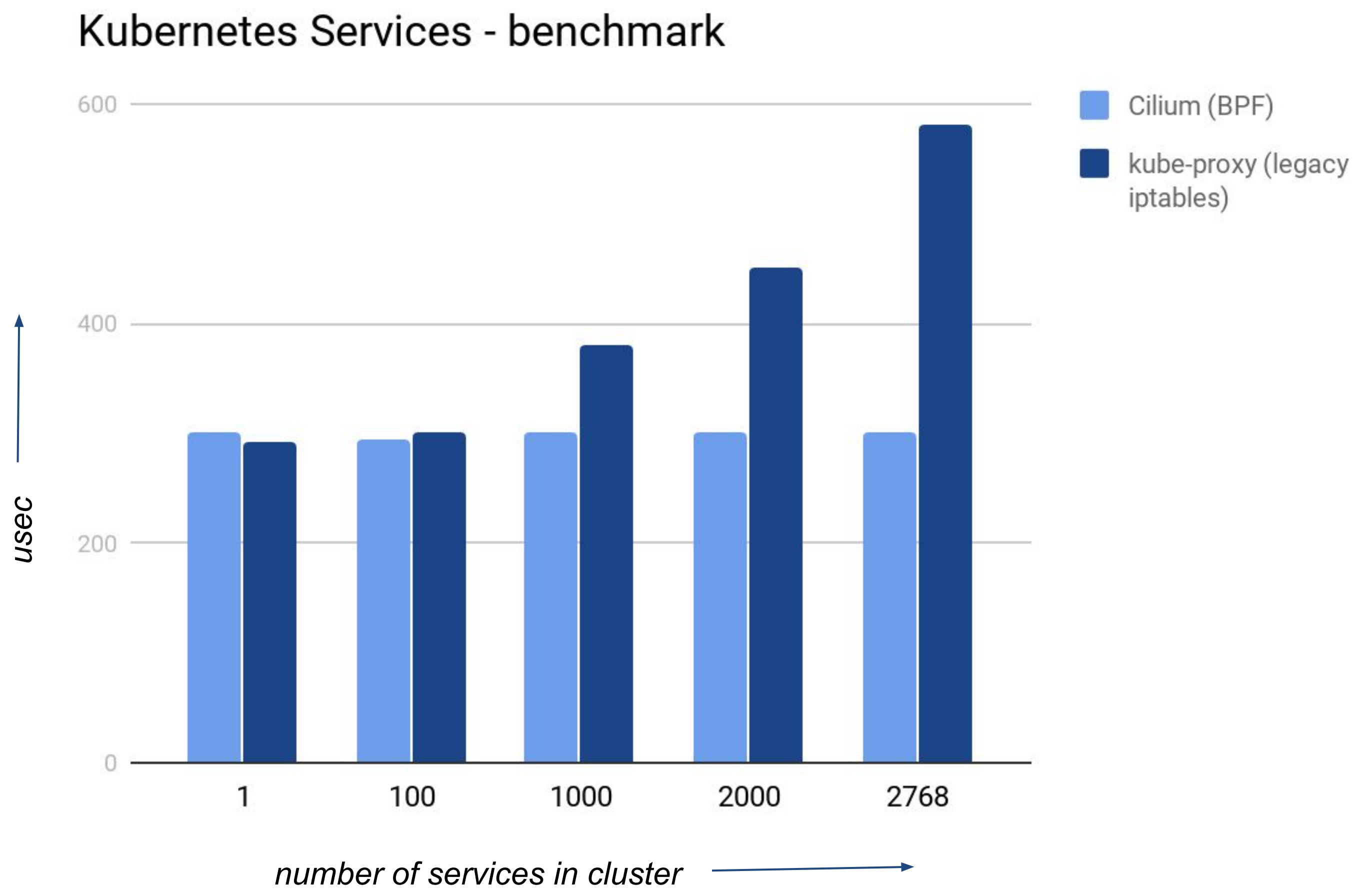
Search O(1)
Insert O(1)
Delete O(1)

Iptables, kube-proxy

- Linear list.
- All rules in the chain have to be replaced as a whole.



Search O(n)
Insert O(1)
Delete O(n)



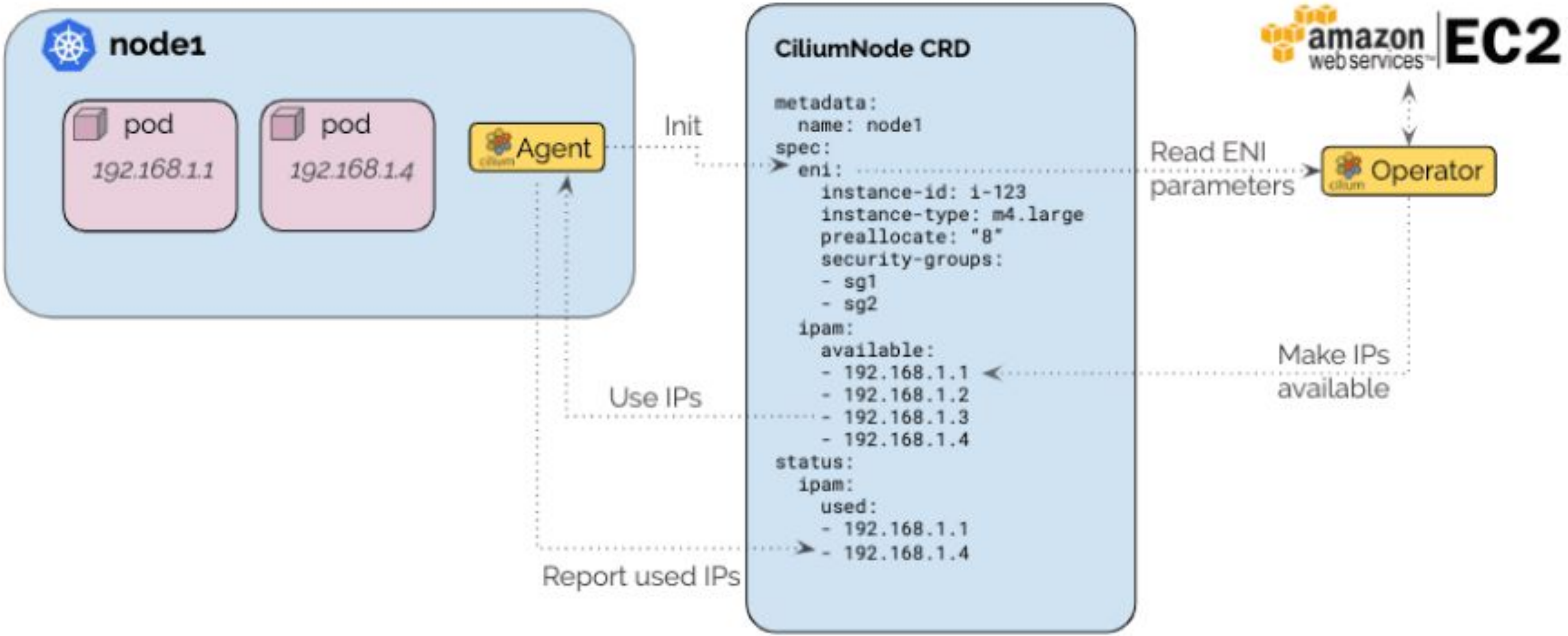
Kubernetes Services - benchmark
600
400
200
0
1
100
1000
2000
2768
Cilium (BPF)
kube-proxy (legacy iptables)
usec
number of services in cluster

CNI chaining

Policy enforcement, load balancing, multi-cluster connectivity

IP allocation, configuring network interface, encapsulation/routing inside the cluster

Native support for AWS ENI



HUBBLE

Hubble is a fully distributed networking and security observability platform for cloud native workloads.
It is built on top of Cilium and eBPF to enable deep visibility in a transparent manner.

Hubble provides

- Service dependencies and communication map
- Operational monitoring and alerting
- Application monitoring
- Secure observability

Known limitations of Hubble:

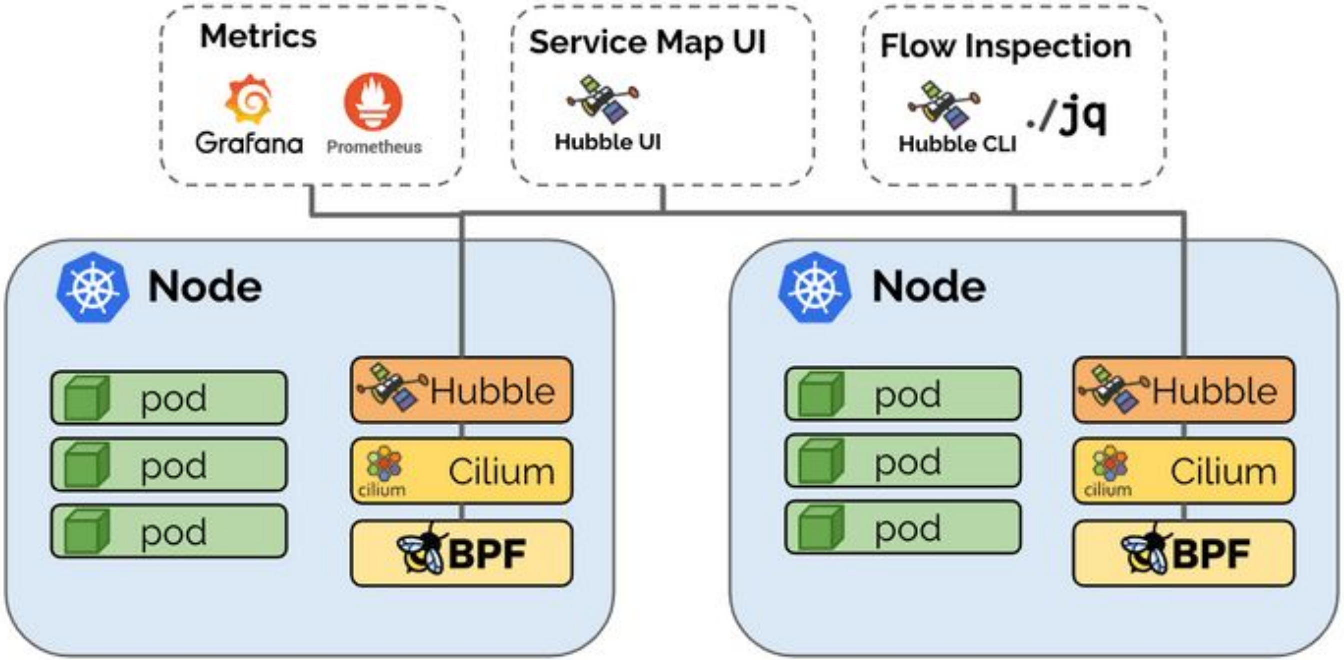
- Hubble is in beta
- Not all components of Hubble are covered by automated testing.
- Architecture is scalable but not all code paths have been optimized for efficiency and scalability yet

HUBBLE Components

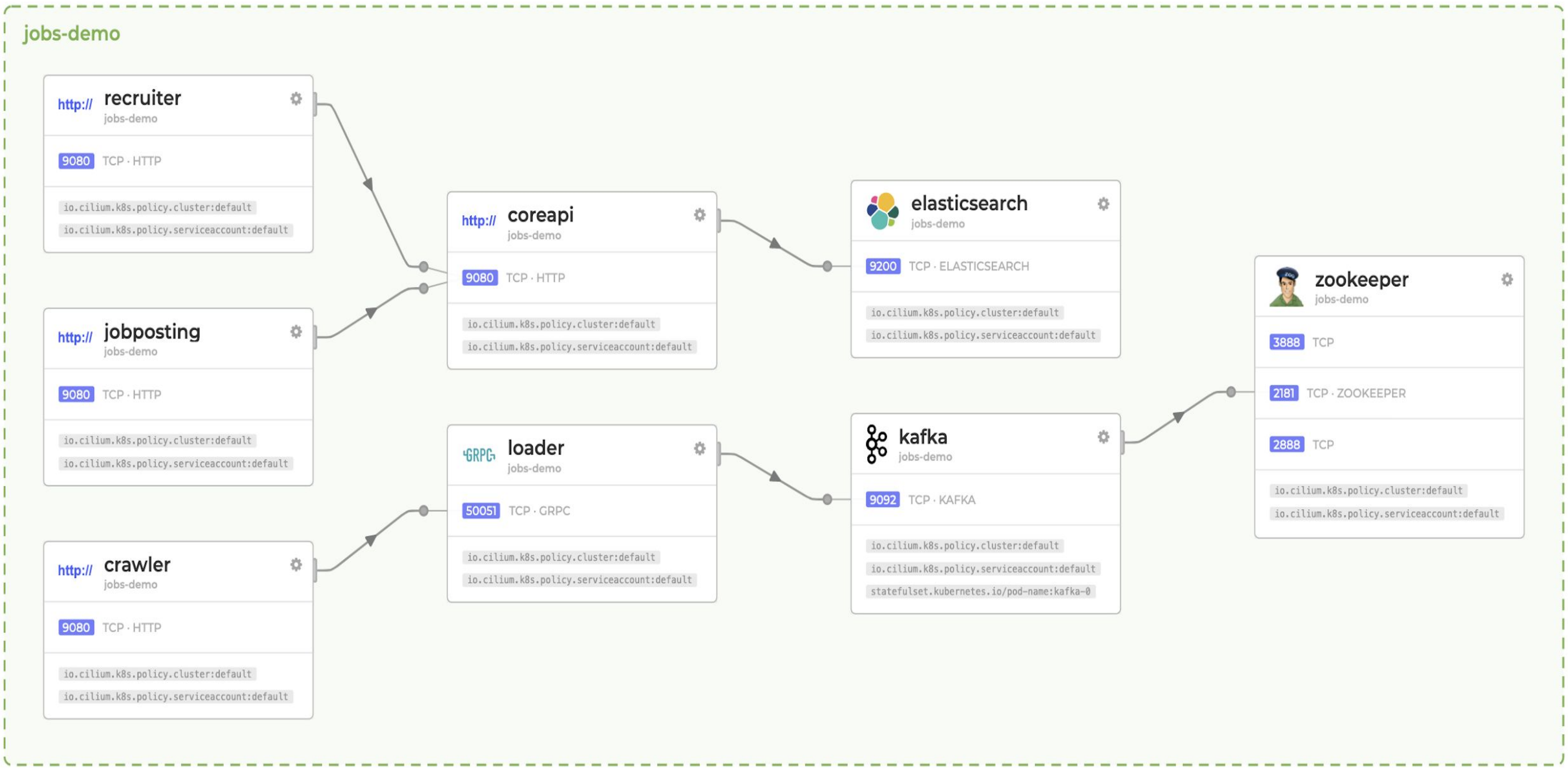
The following components make up Hubble:

- Hubble Agent
  - The Hubble Agent is what runs on each worker node. It interacts with the Cilium agent running on the same node and serves the flow query API as well as the metrics.
- Hubble Storage
  - Hubble storage layer consists of an in-memory storage able to store a fixed number of flows per node.
- Hubble CLI
  - The CLI connects to the flow query API of a Hubble agent running on a node and allows to query the flows stored in the in-memory storage using server-side filtering.
- Hubble UI
  - The Hubble UI uses the flow query API to provide a graphical service communication map based on the observed flows.

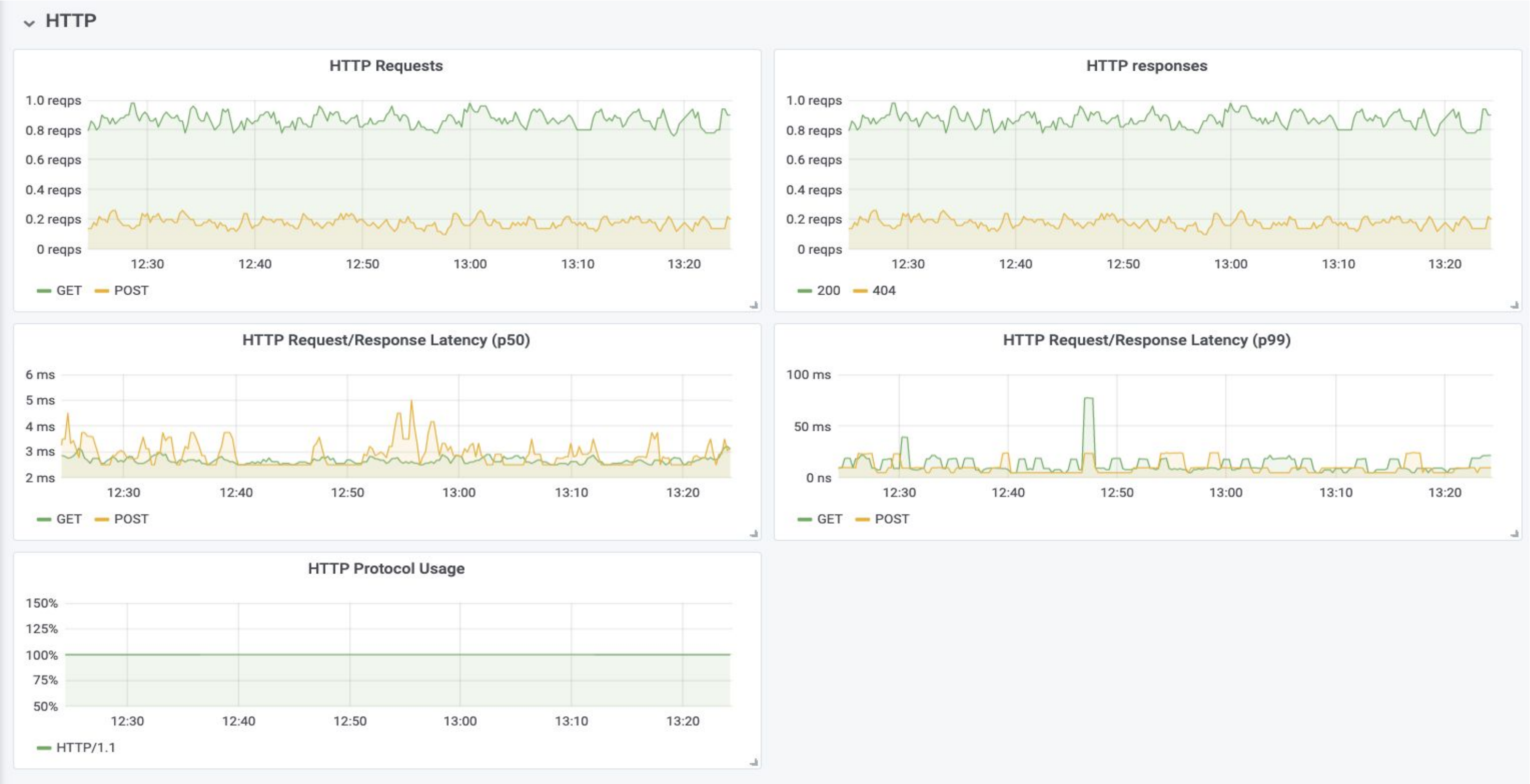
Hubble running on top of Cilium and eBPF



Hubble Service Maps



Hubble HTTP metrics



To sum it up

Why Cilium is awesome?

- It makes disadvantages of **iptables** disappear. And always gets the best from the **Linux kernel**.
- **Cluster Mesh** / multi-cluster.
- Makes **Istio** faster.
- Offers **L7 API Aware filtering** as a Kubernetes resource.
- Integrates with the other popular **CNI plugins** – **Calico**, **Flannel**, **Weave**, **Lyft**, **AWS CNI**.

We adapt. You succeed.