



Tech Club webinář

Kubernetes, Openshift, Cilium a nový Cisco SmartSwitch:
jak zapadají do nové koncepce bezpečnosti DC a aplikací

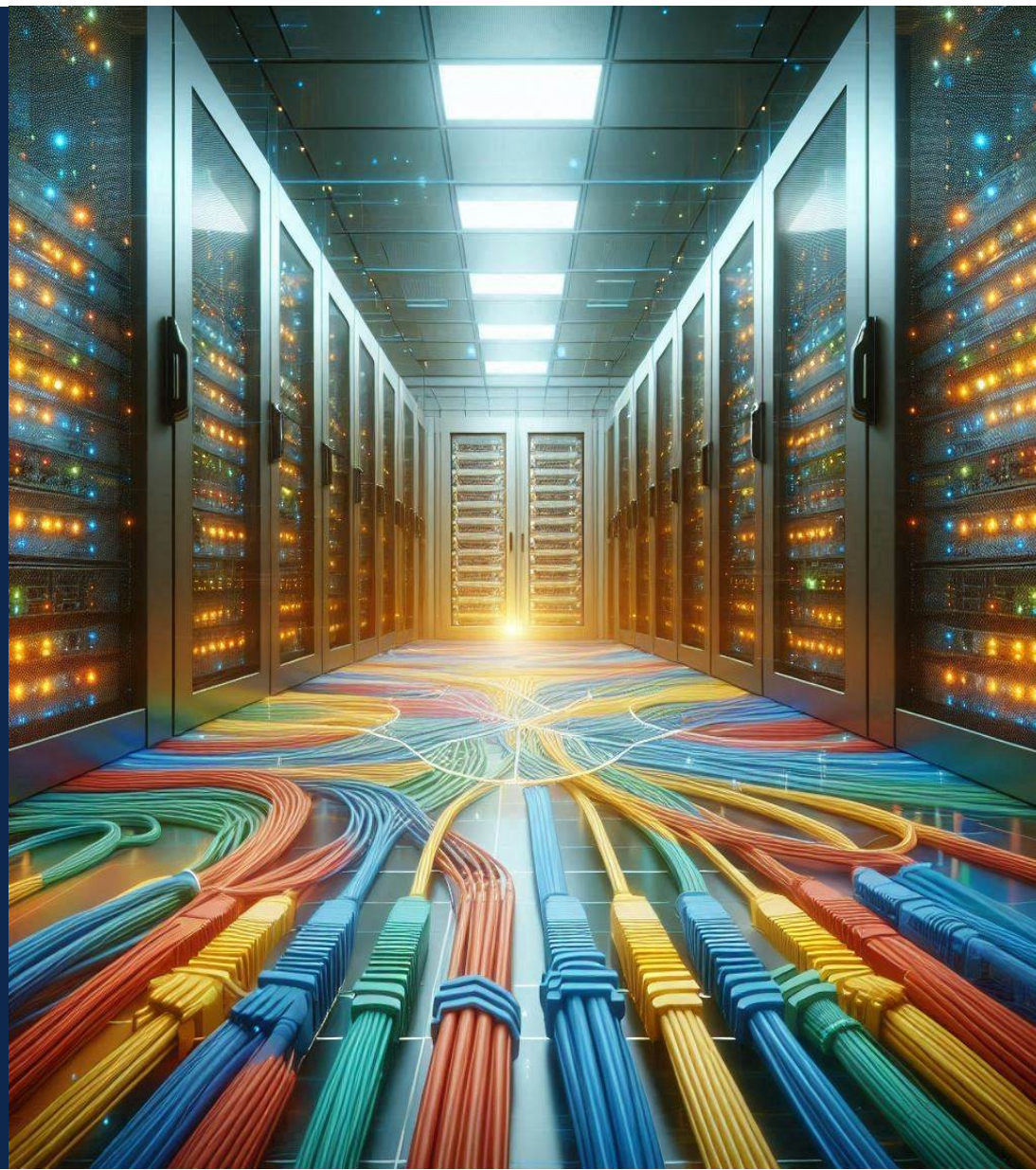
Martin Diviš, TSE, mdivis@cisco.com

1. duben 2025

Segmentace – tradiční řešení bezpečnosti v DC

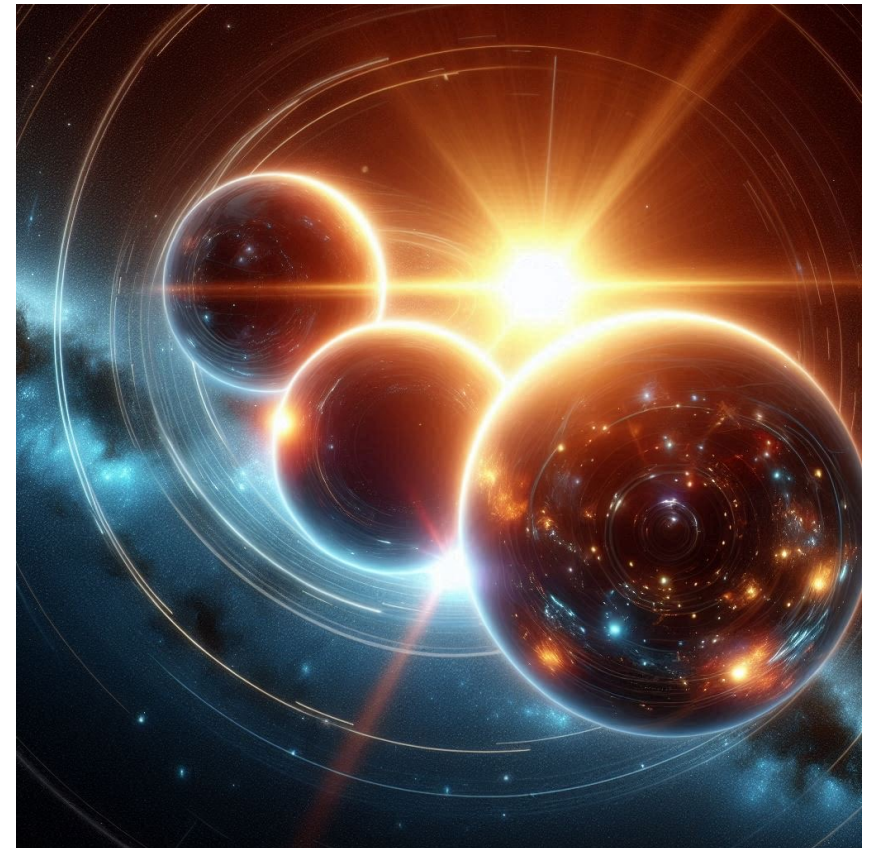


© 2025 Cisco and/or its affiliates. All rights reserved.



Segmentation – Policy vs. Enforcement

- The "three-body problem"
 - **Security rules** – who can do what, talk to who, etc.
 - **Identity** – how to identify the subject of the rule?
 - **Observe and Enforce** – how to make sure we see the behavior of a subject and how to enforce the rules?



Segmentation in DC

- **Identity** – IP address, IP subnet
- **Security rules** – ACL-like, requires a lot of knowledge and work, hard to maintain over time
- **Observe and Enforce** – make sure data flow is seen: network ACLs, Firewalls (often routing between subnets), ACI policies, host based firewalls

Segmentace v prostředí kontejnerových platformem

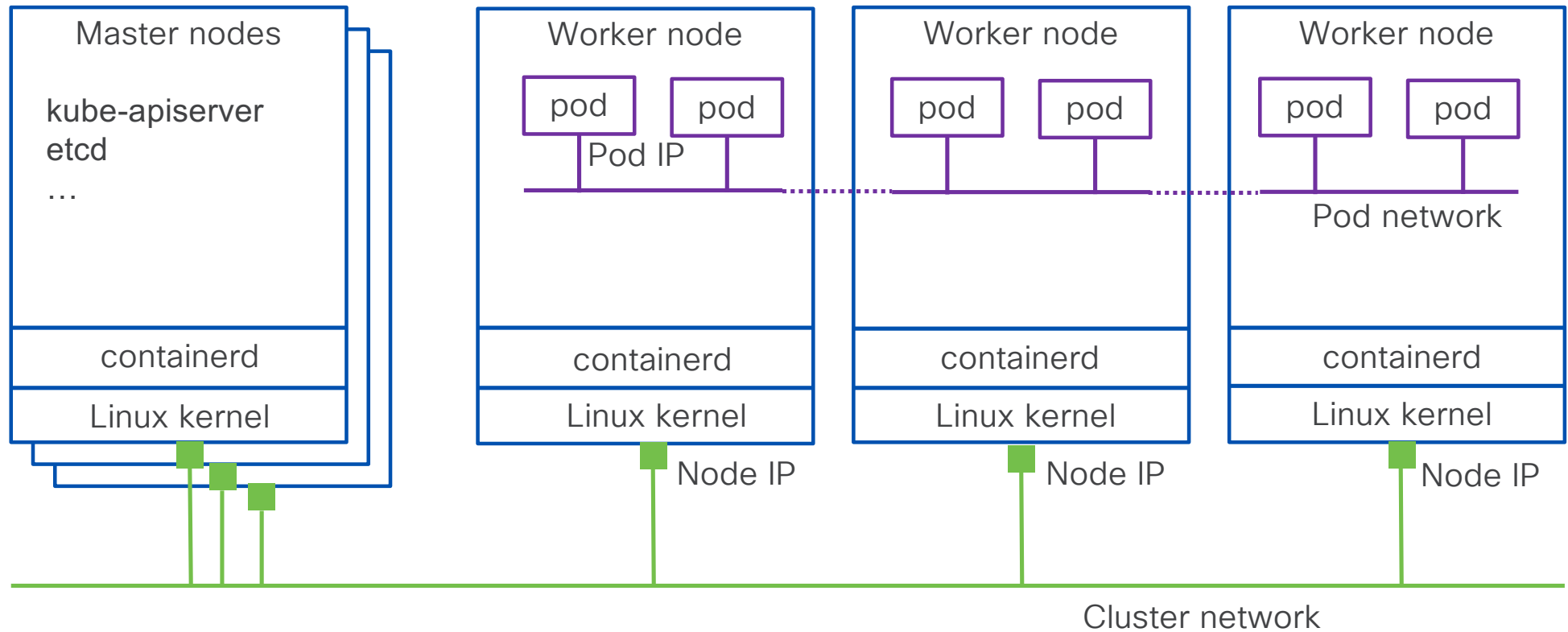


Kubernetes – the cloud native application platform

- Kubernetes (K8s) is an open-source system for automating deployment, scaling, and management of containerized applications.
- It provides a platform for managing resilient, scalable, and distributed systems.
- Key benefit: Abstracts away the complexities of managing individual containers.
- It allows for declarative configuration.

Kubernetes cluster

Pod network – flat address space, random IP addresses of pods



How are application really exposed

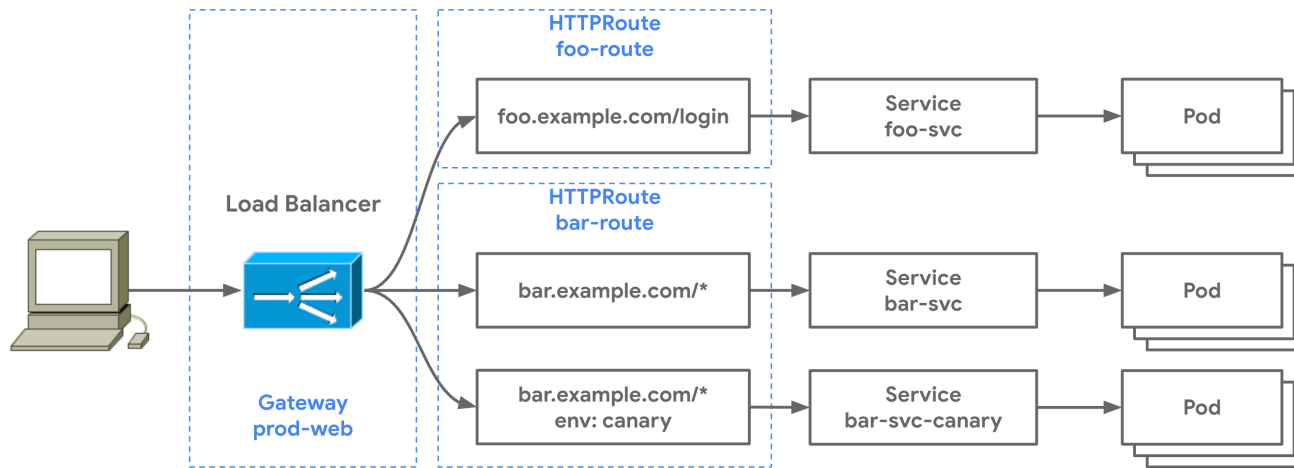
- Pods have "random" IPs – direct communication is rather rare
- Pods have replicas – we need loadbalancing among the replicas
- Services
 - Expose applications in pods to other pods and to external world
 - Loadbalance traffic to pod replicas
 - Check availability of pods via probes (readiness, health)

Services types

```
apiVersion: v1
kind: Service
metadata:
  name: my-loadbalancer-service
spec:
  type: LoadBalancer
  selector:
    app: my-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

- **Cluster IP** – not exposed outside of the cluster, for internal Pod-Service communication only
- **NodePort** – the same port assigned to an application on all nodes on node IPs. Externally, connect to any node on that port to reach the service
- **LoadBalancer** – single external endpoint to the service, often relies on NodePort mappings. Not native part of Kubernetes, it can be external to the cluster (physical F5,...) or SW running on the cluster (MetalLB,...). External IP address provided by the LoadBalancer management and sent back to the K8S management.

Gateway API – Application Routing into K8S



Implementations:

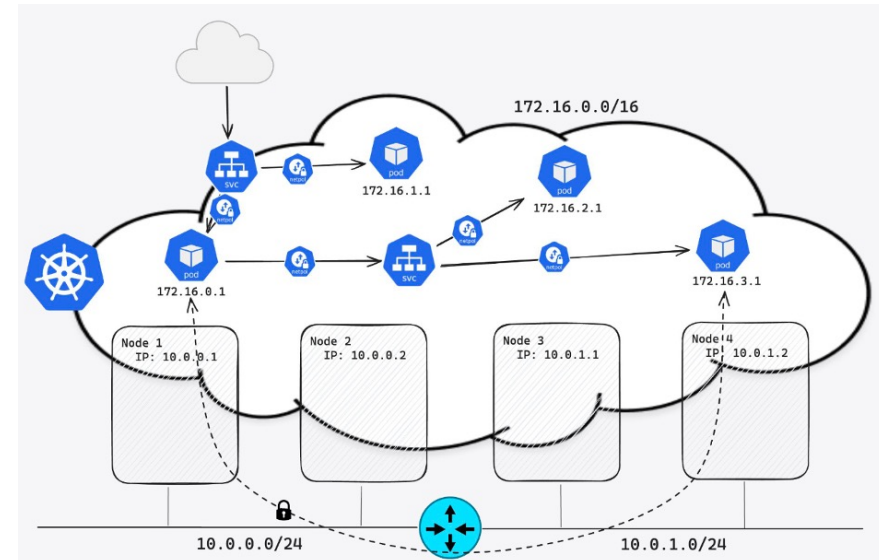
<https://gateway-api.sigs.k8s.io/implementations/>

```
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: example-gateway
spec:
  gatewayClassName: example-gateway-class
  listeners:
  - name: http
    protocol: HTTP
    port: 80
```

```
---
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: example-route
spec:
  parentRefs:
  - name: example-gateway
  hostnames:
  - "example.com"
  rules:
  - backendRefs:
    - name: example-svc
      port: 80
```

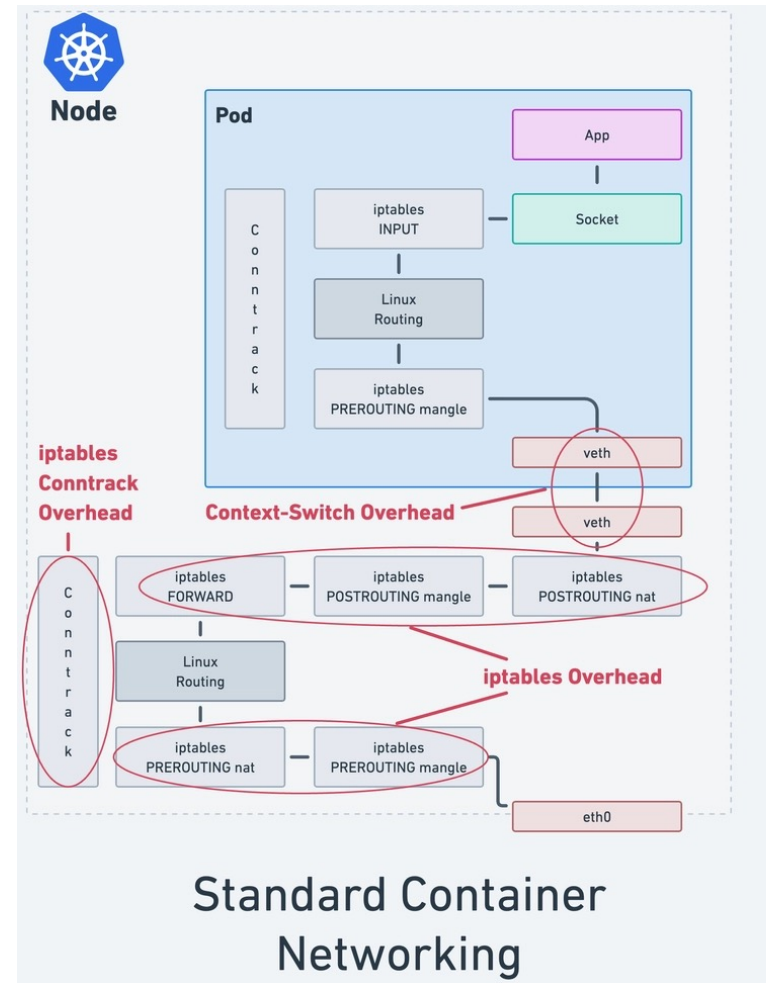
Kubernetes Networking - Summary

- IPAM - Dynamic (CNI)
- Routing - Dynamic (CNI)
- East-West connectivity
 - Service type: ClusterIP (kube-proxy)
- North-South connectivity
 - Service type: LoadBalancer (kube-proxy)
- Service discovery: Dynamic (CoreDNS)
- Security:
 - NetworkPolicy (CNI)
 - Transparent Encryption (CNI / Service Mesh)



Challenges of iptables

- ACLs as sequential list of rules
- Updates all rules in single transaction
- Matches based on IP proto/addr/port only
- New IP/Port require rules added and chain changed
- Larger overhead
- Reduced performance and increased latency at scale



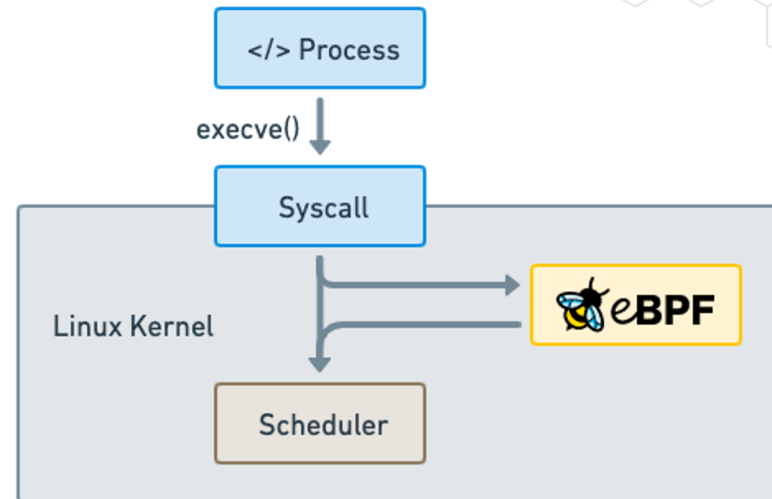
A Solution Arises





Makes the Linux kernel programmable in a secure and efficient way.

“What JavaScript is to the browser, eBPF is to the Linux Kernel”



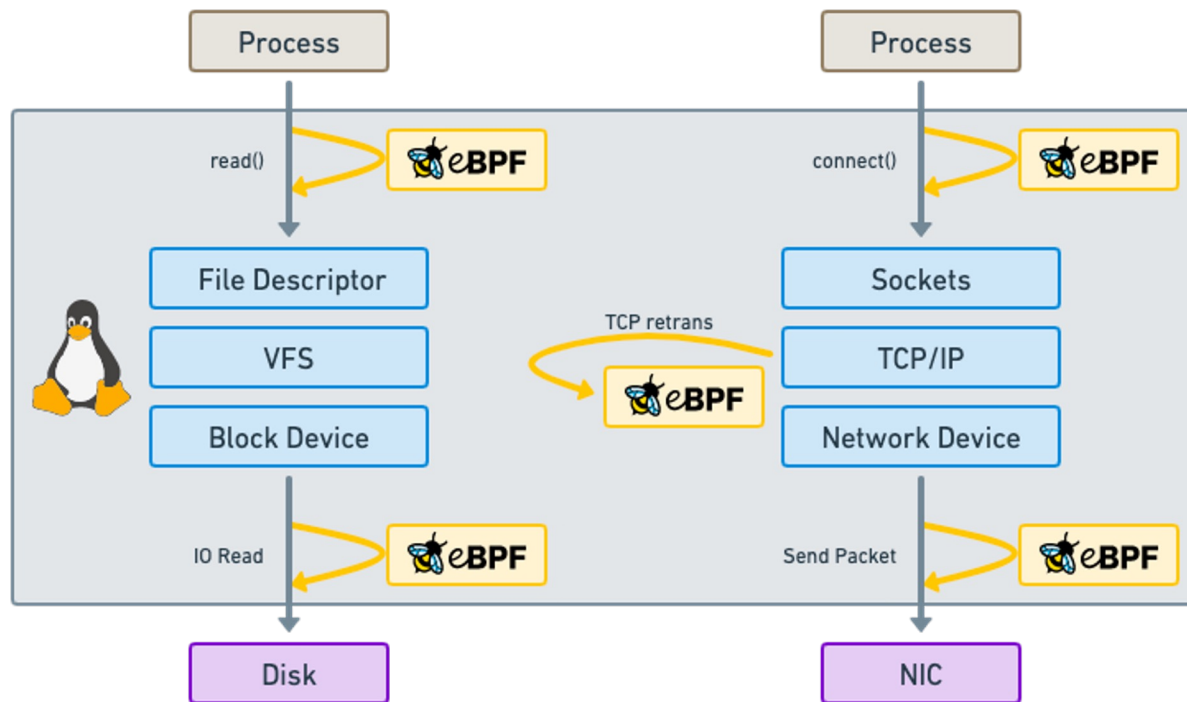
```
int syscall__ret_execve(struct pt_regs *ctx)
{
    struct comm_event event = {
        .pid = bpf_get_current_pid_tgid() >> 32,
        .type = TYPE_RETURN,
    };

    bpf_get_current_comm(&event.comm, sizeof(event.comm));
    comm_events.perf_submit(ctx, &event, sizeof(event));

    return 0;
}
```



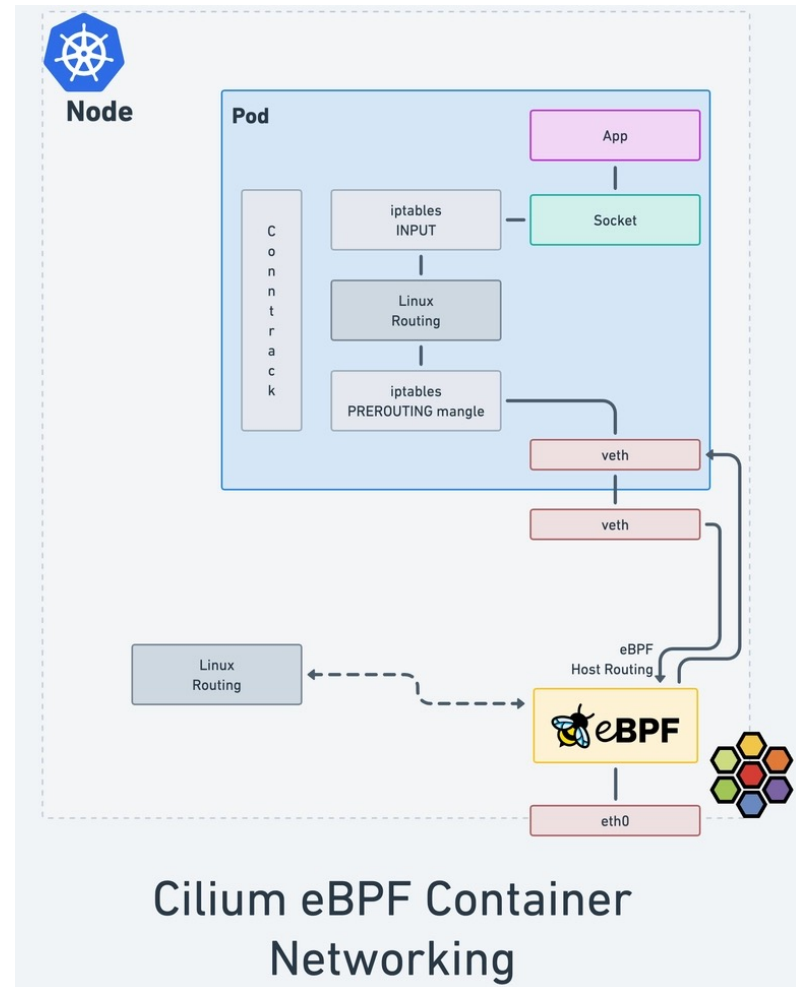
Run eBPF programs on events



Attachment points

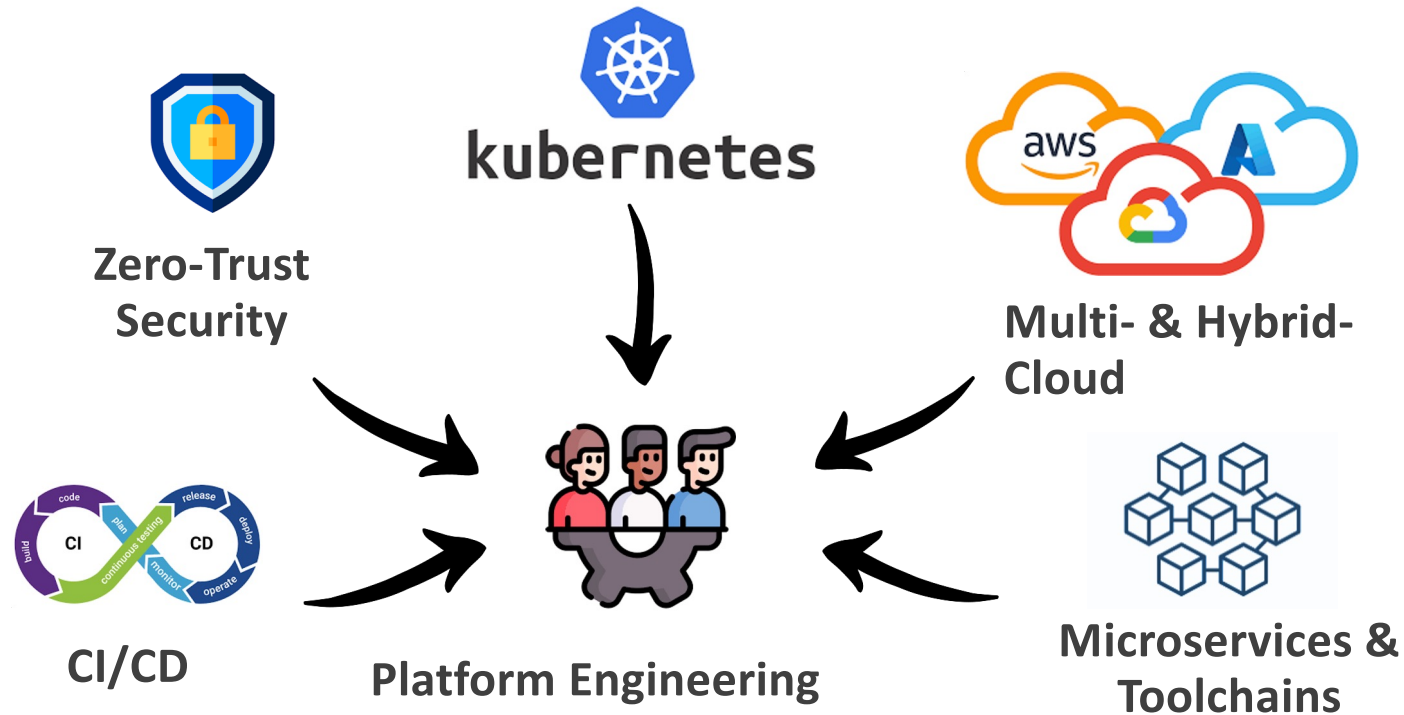
- Kernel functions (kprobes)
- Userspace functions (uprobe)
- System calls
- Tracepoints
- Sockets (data level)
- Network devices (packet level)
- Network device (DMA level) [XDP]
- ...

Removing iptables



Platform Engineering

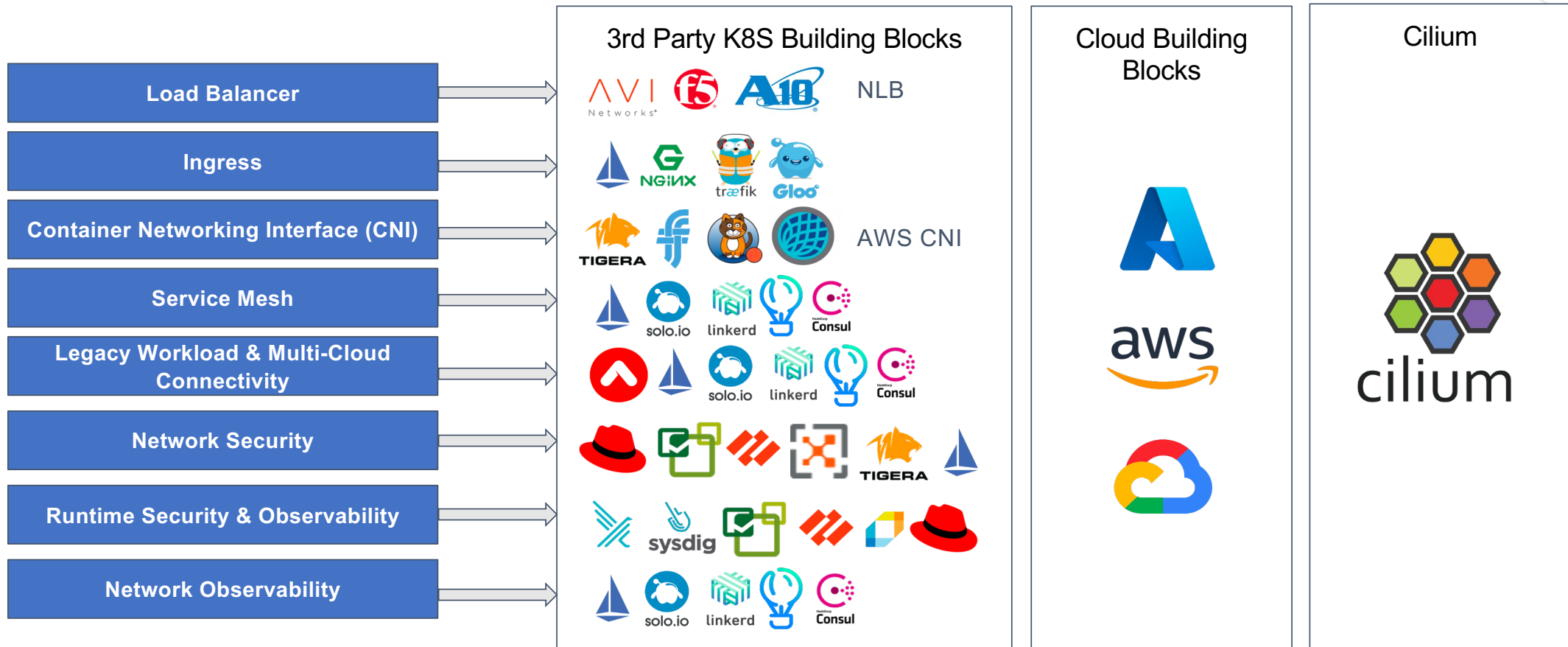
Key Infrastructure Trend



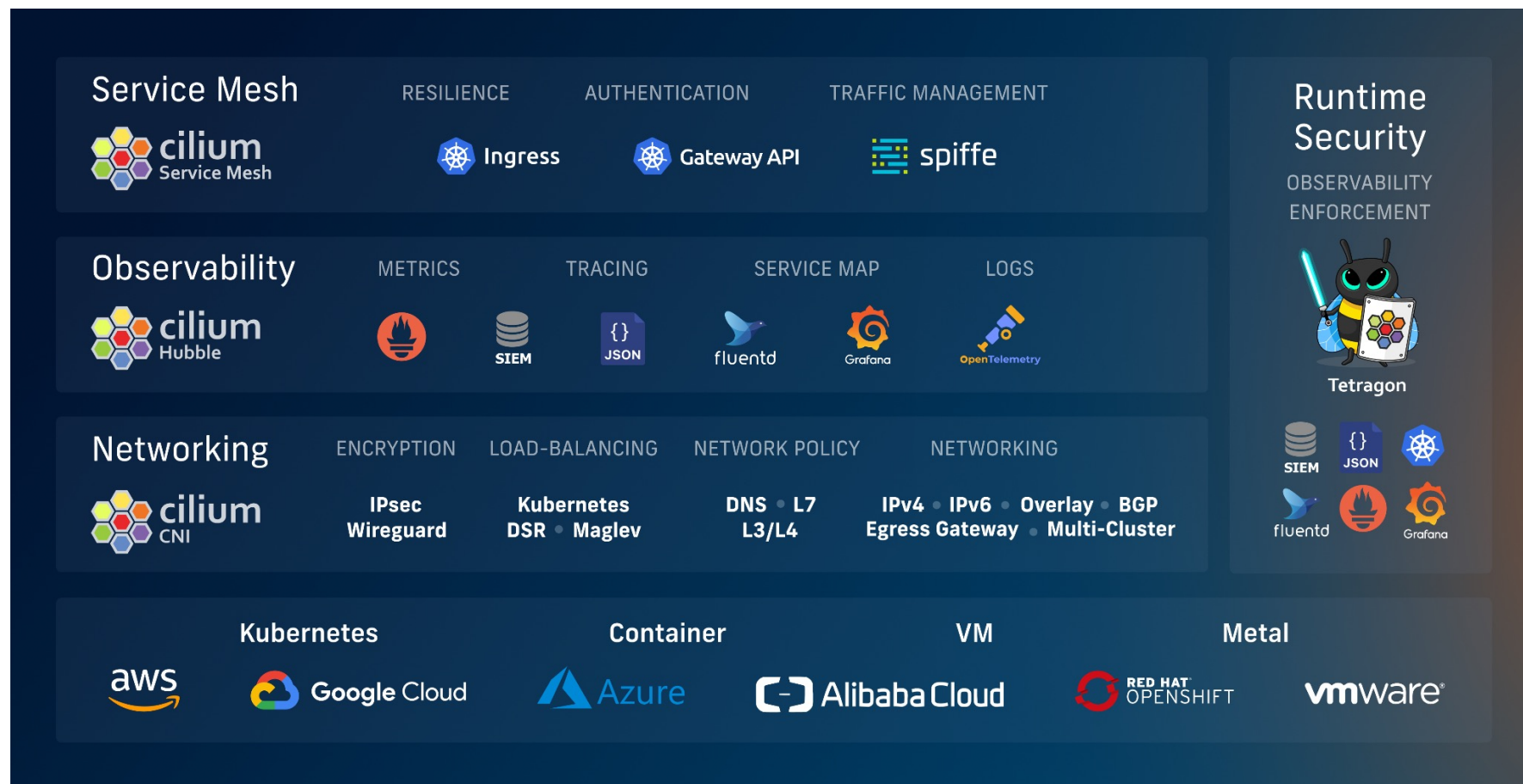
Platform engineering is the newest sociotechnical discipline to arise in response to the cloud native world. As the process of designing, building, and maintaining workflows and tools for software engineering organizations, platform engineering helps drive consistency and speed up common tasks.



Kubernetes Networking Tool Sprawl



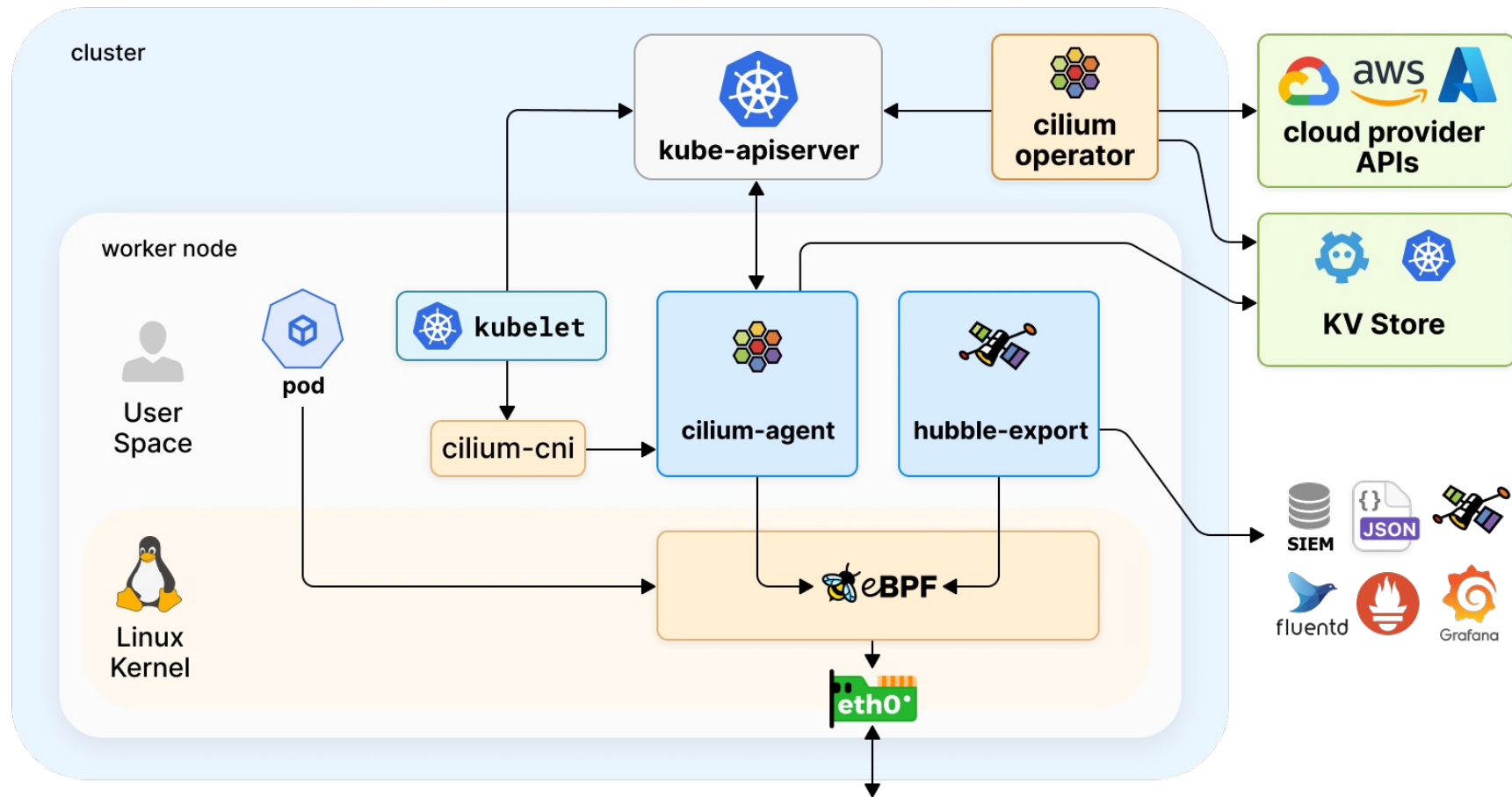
The Isovalent Overview



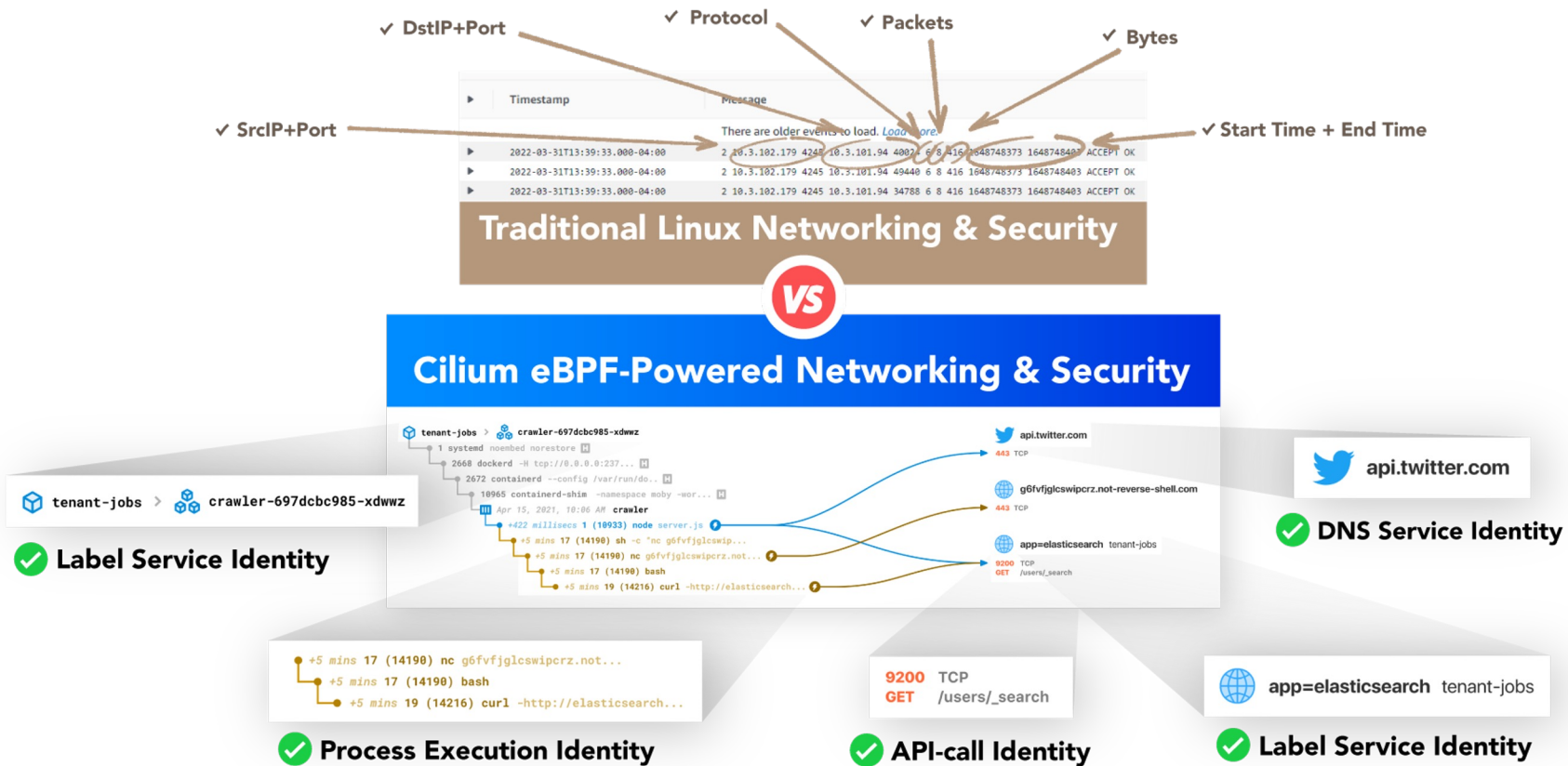
Cilium + Hubble



Cilium Architecture – More Detail

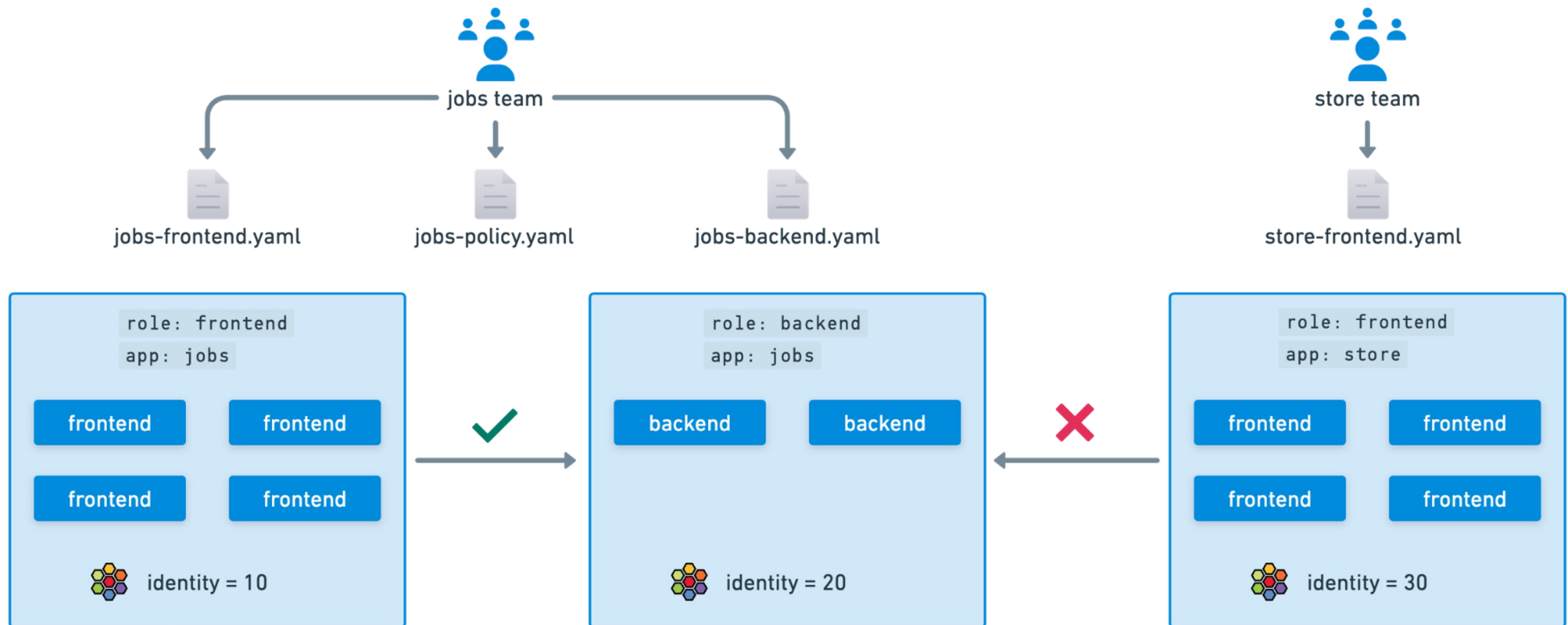


Identity-based Security



Micro-Segmentation

Label based East-West Application or Multi-tenant Security Enforcement





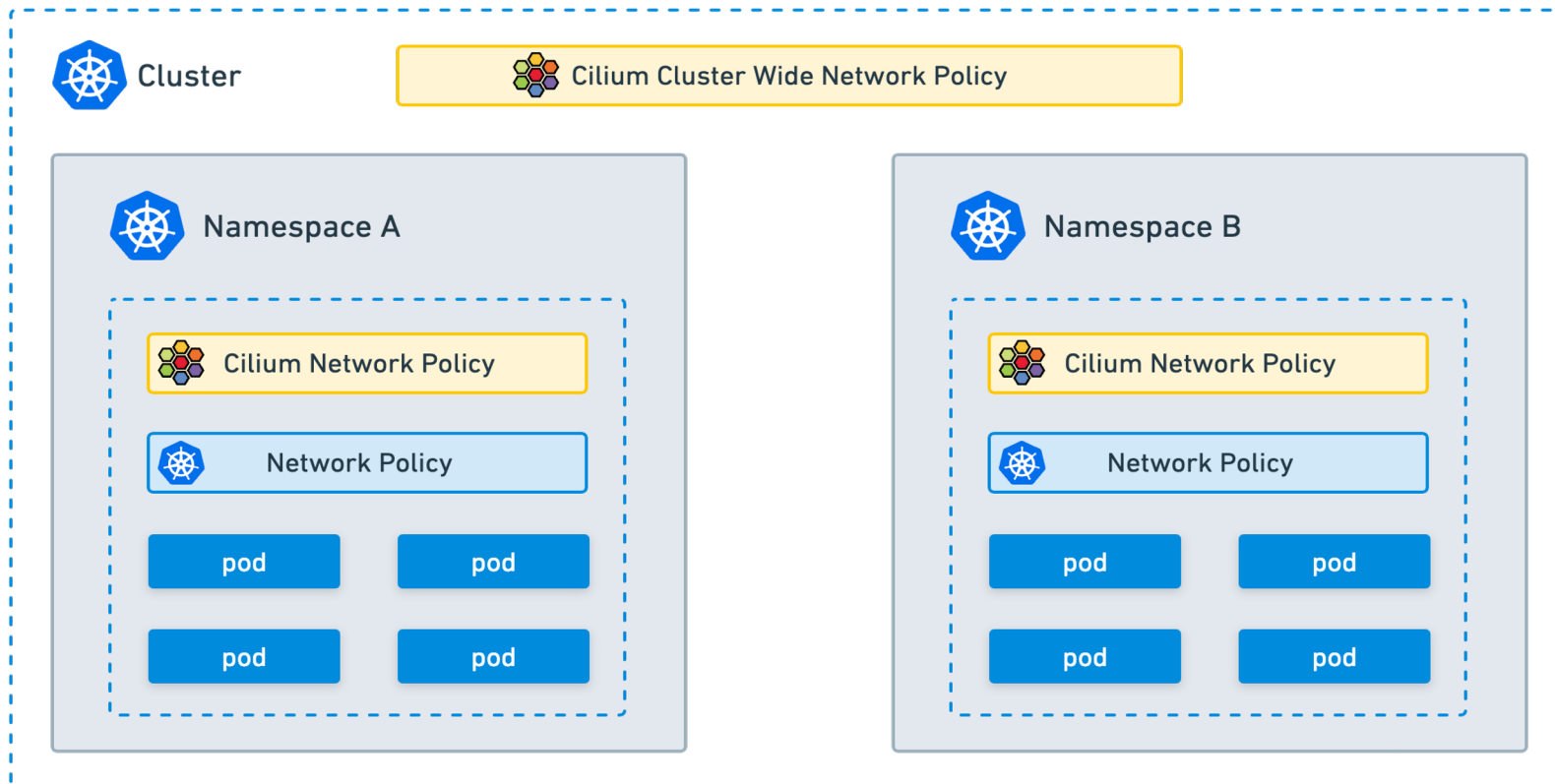
Example Layer 7 Rule

- CiliumNetworkPolicy
- Allows matching pods on labels
- Defines ingress from source labels
- Defined on destination range
- Applies to POST operations to specific URI

```
apiVersion: "cilium.io/v2"
kind: CiliumNetworkPolicy
metadata:
  name: "rule1"
spec:
  description: "L7 policy to restrict access to specific HTTP call"
  endpointSelector:
    matchLabels:
      org: empire
      class: deathstar
  ingress:
    - fromEndpoints:
        - matchLabels:
            org: empire
      toPorts:
        - ports:
            - port: "80"
              protocol: TCP
          rules:
            http:
              - method: "POST"
                path: "/v1/request-landing"
```

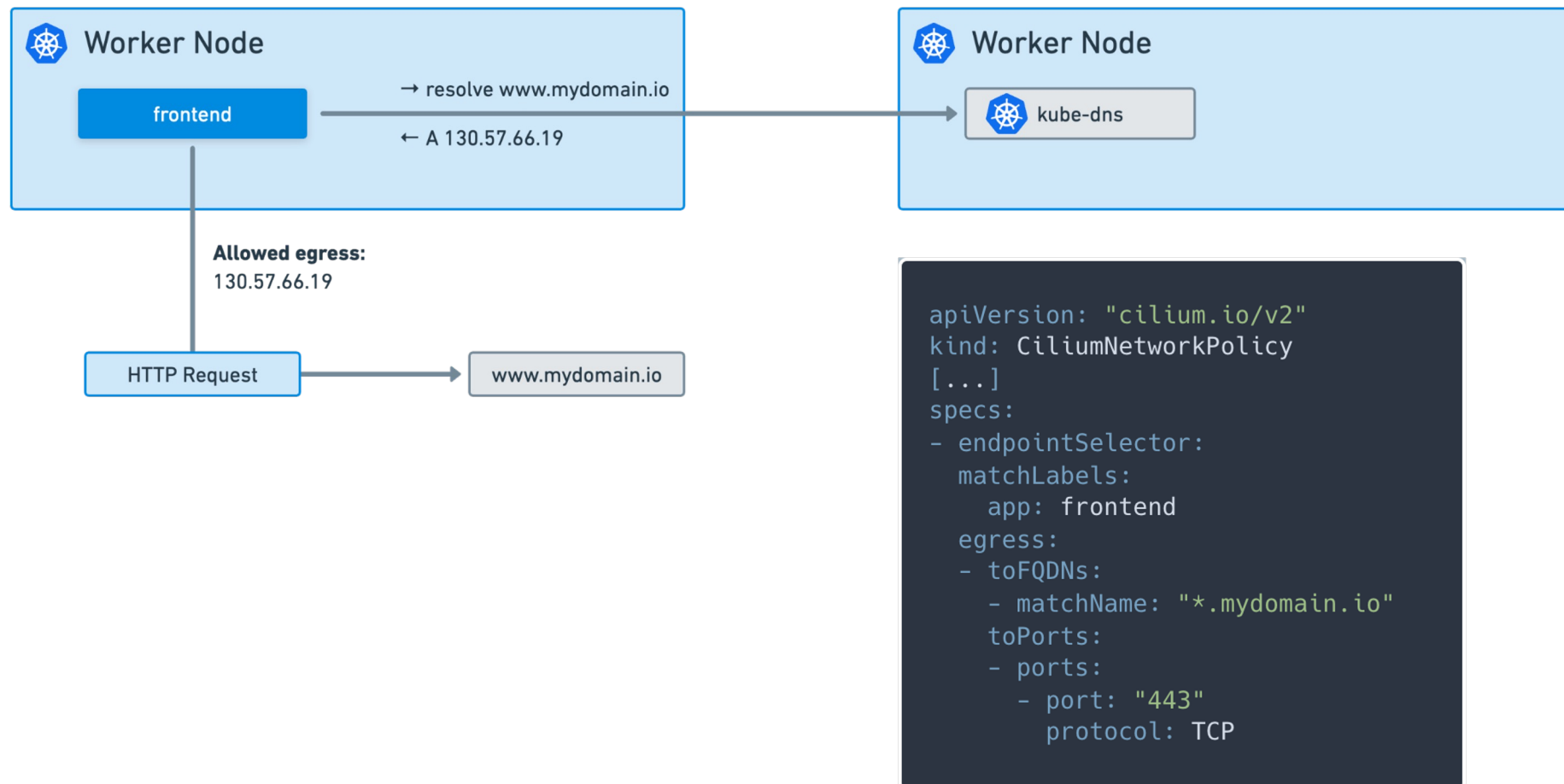
Enforce Consistent Policies across Clusters

Simplify Network Management and set Guardrails for your Platform





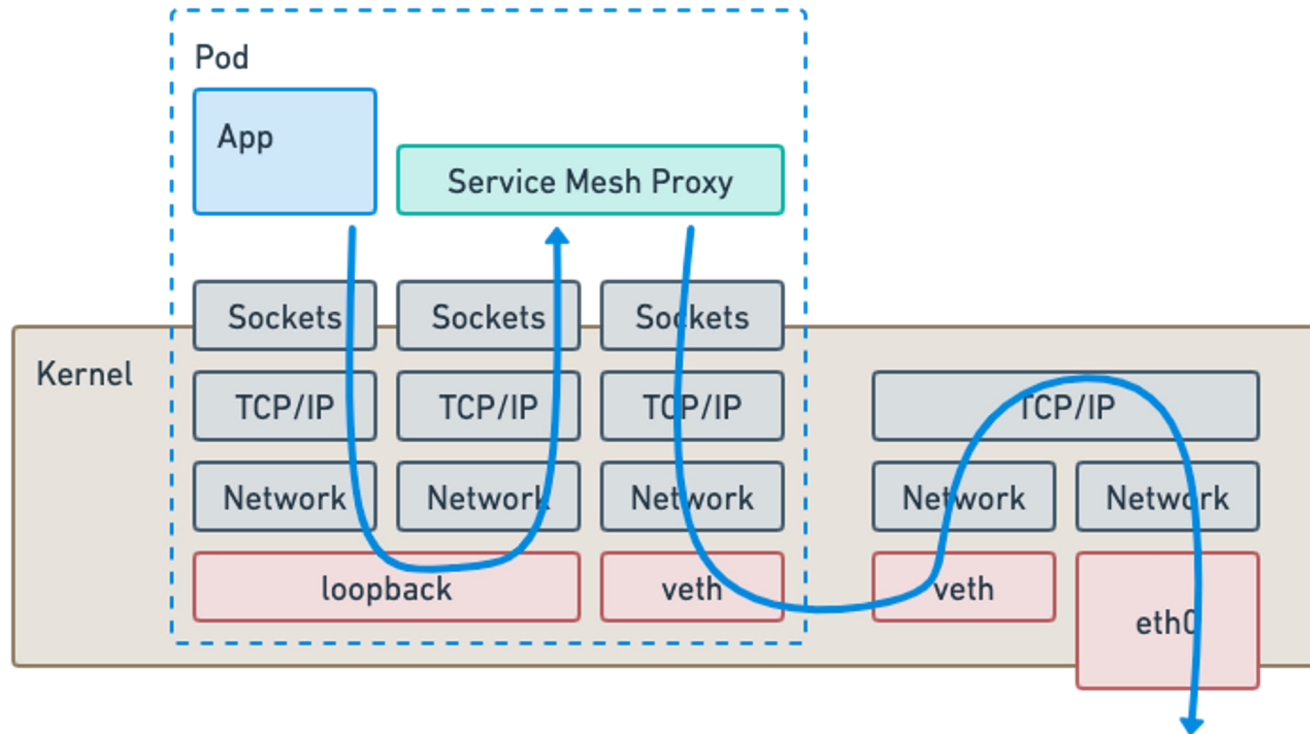
DNS-aware Cilium Network Policy





Service Mesh

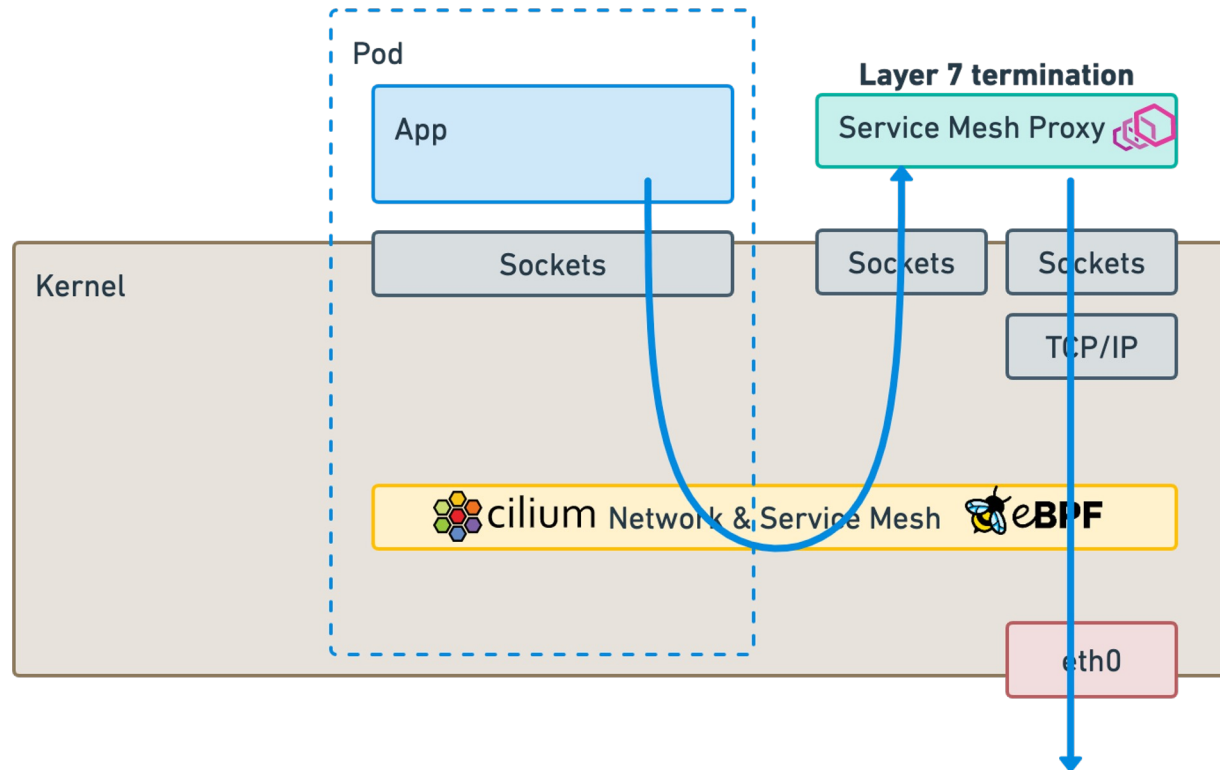
Cost of sidecar injection



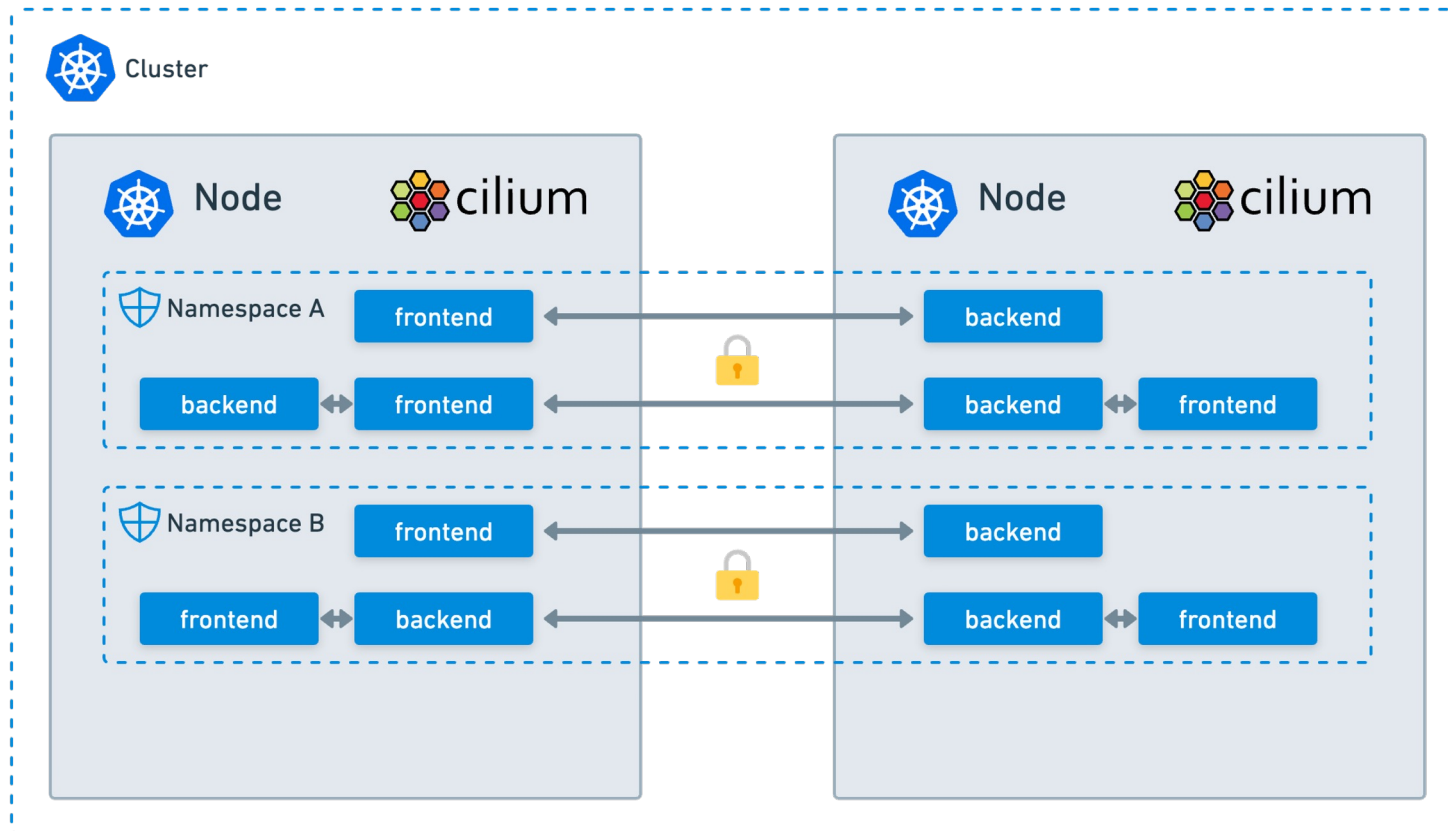


Service Mesh

Envoy for Layer 7 termination when needed

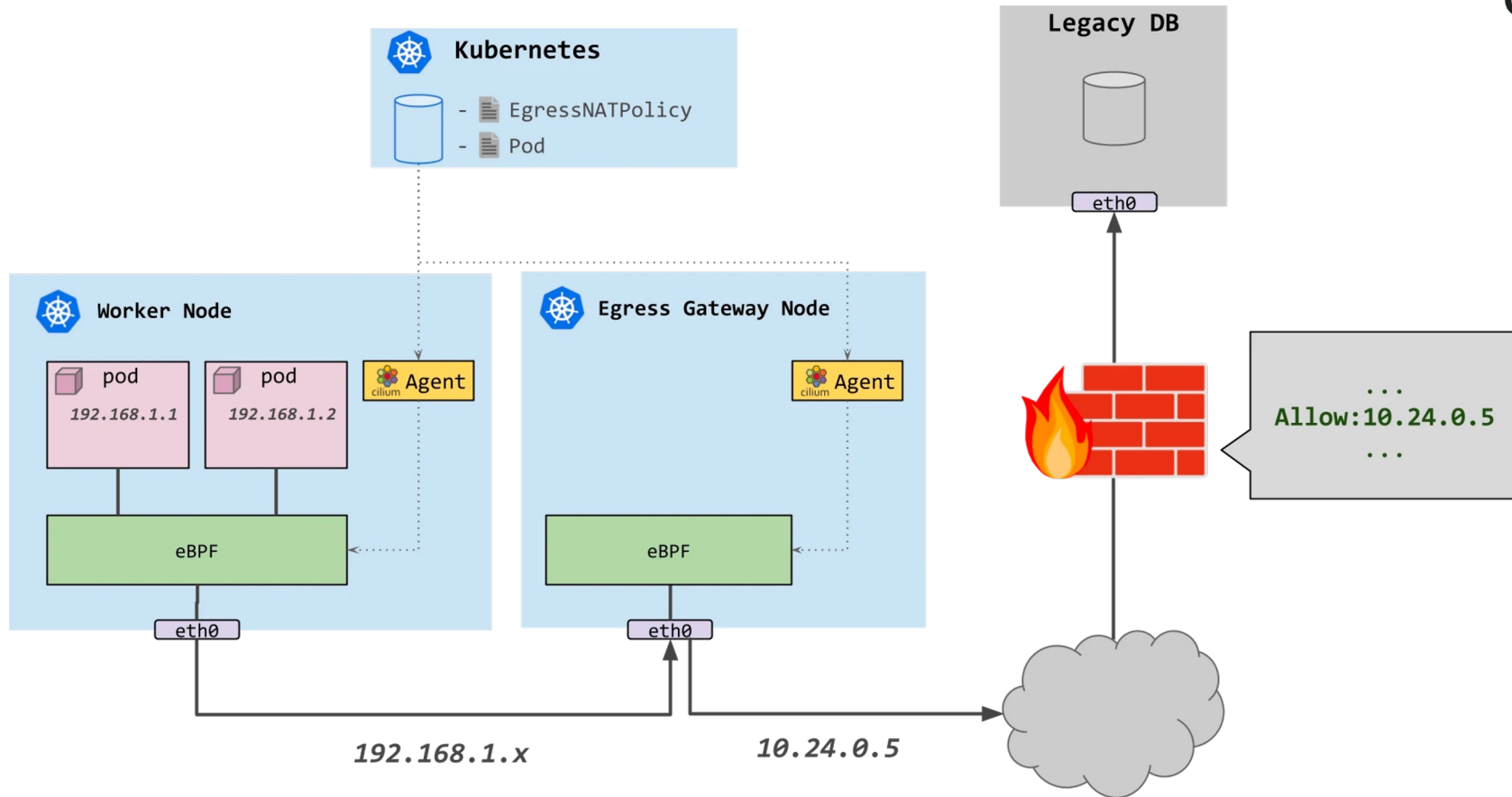


Transparent Data Encryption



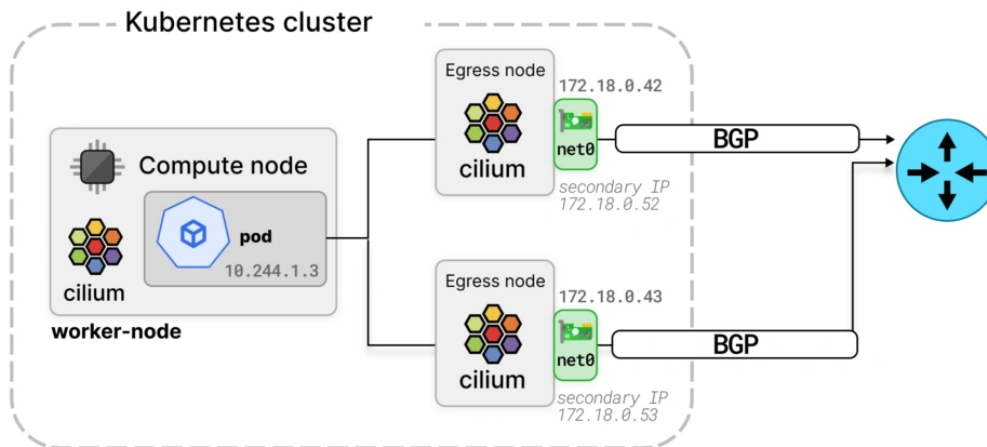


Egress Gateway



Consistent Peering

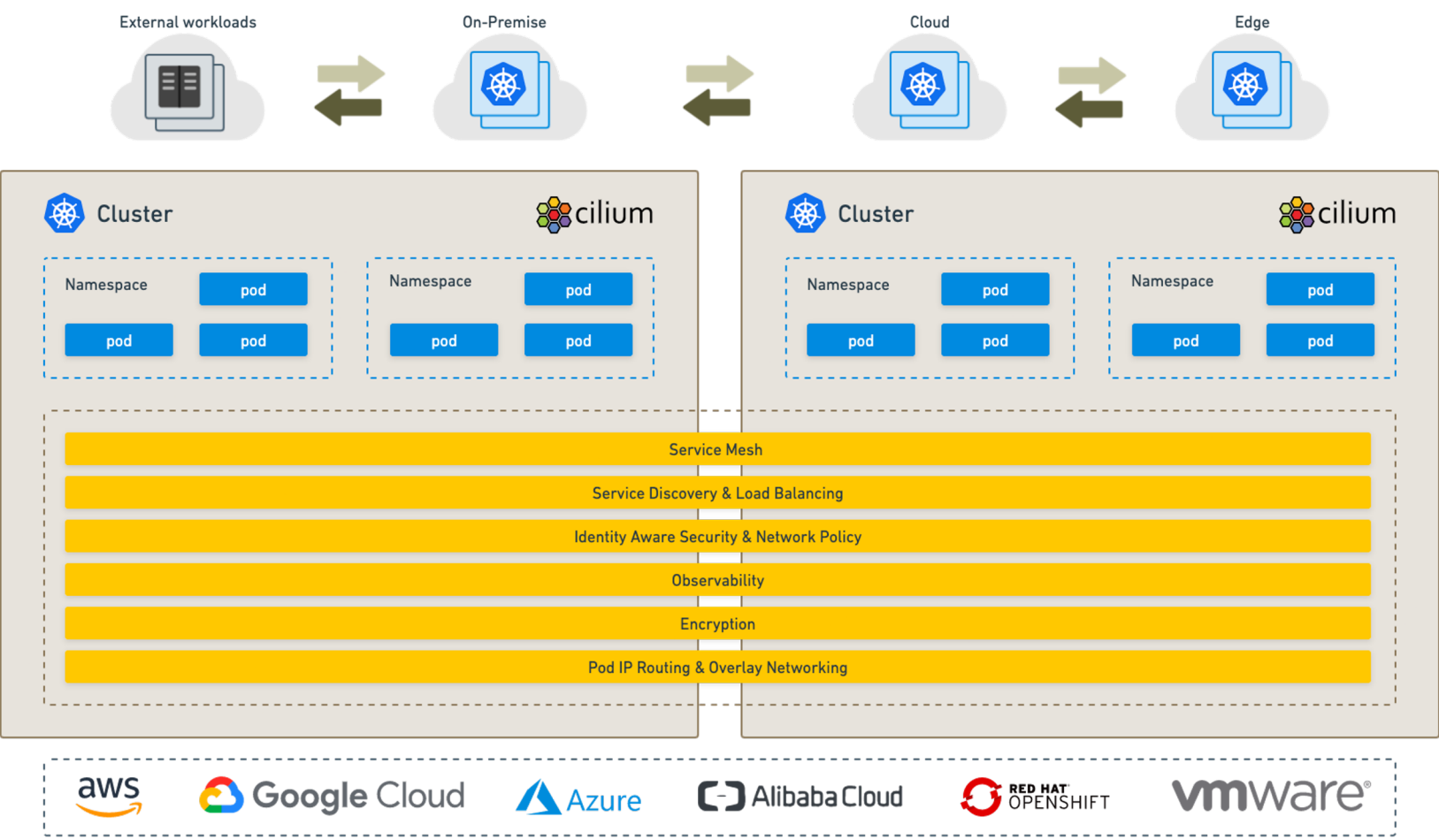
Peer into your existing spine and leaf network



- BGP Support
 - Service Advertisement
 - BFD functionality
 - Supported by CiliumBGP CRD
 - LocalPref
 - Communities
 - IPAM
 - ClusterIP and ExternalIP advertisements

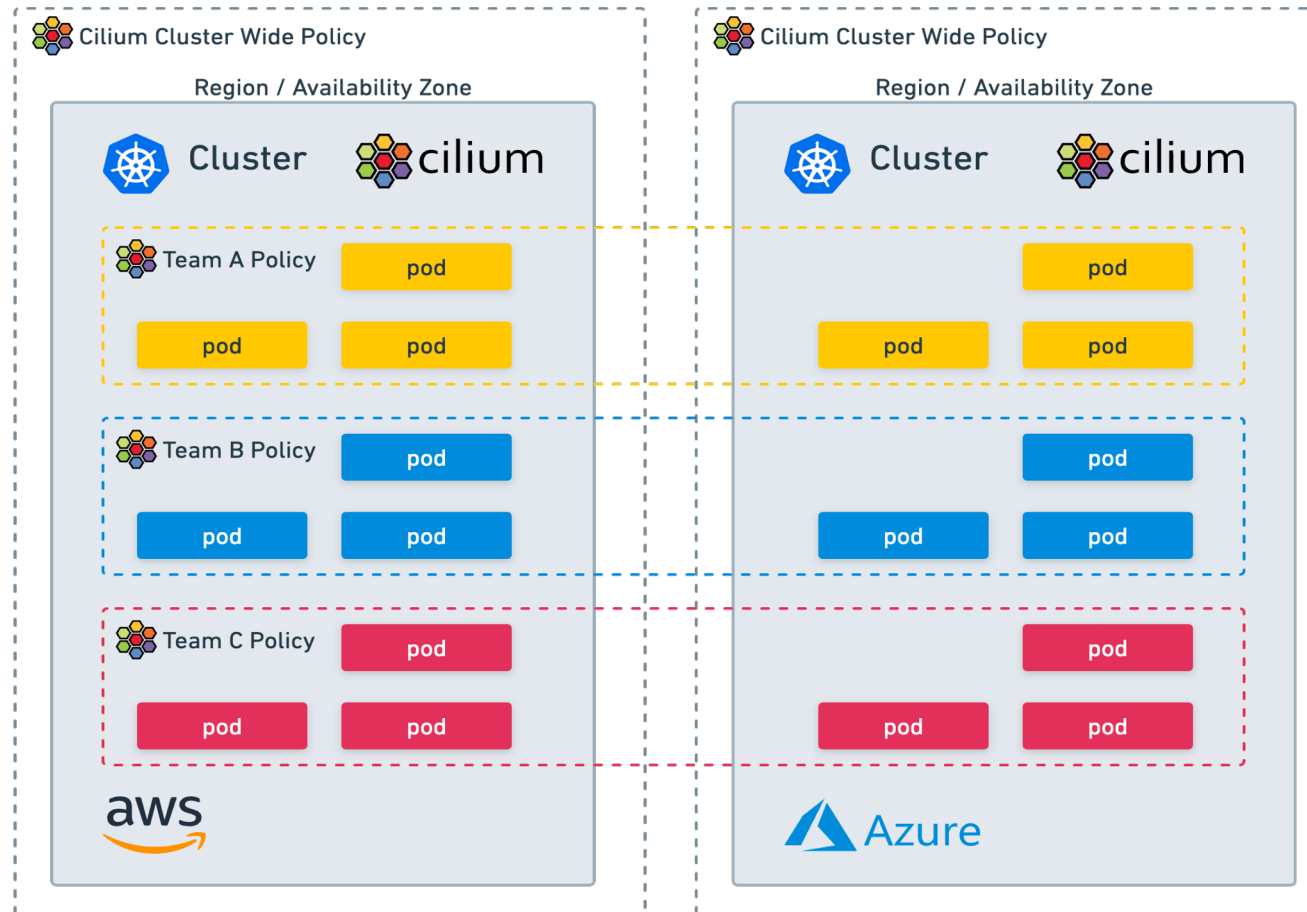


Cluster Mesh - Introduction



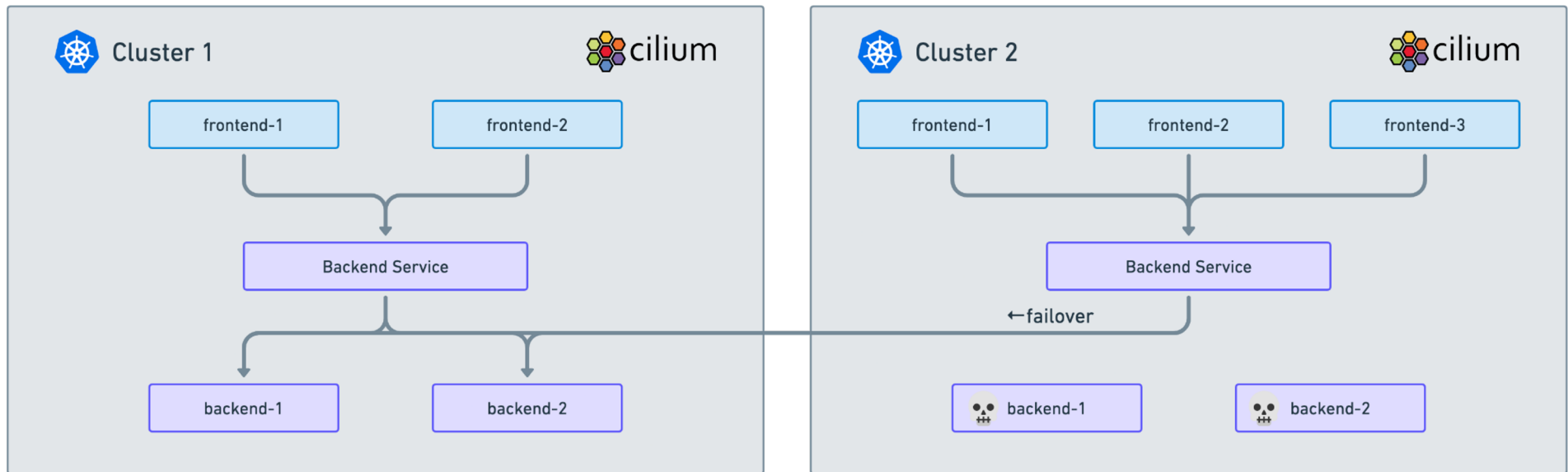
Multi-Cluster Security

Policy Enforcement across Multiple Clusters



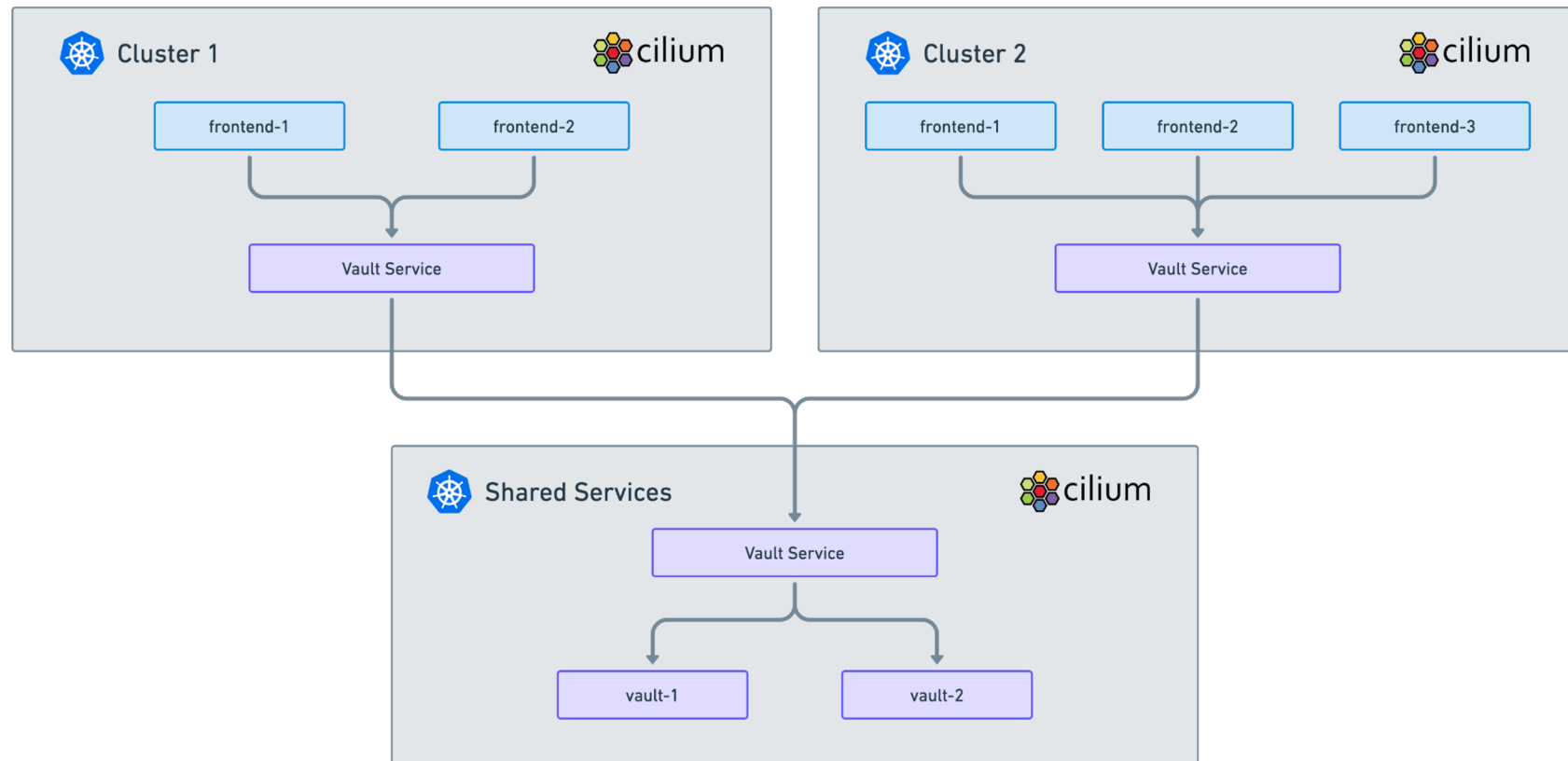


Cluster Mesh - High Availability



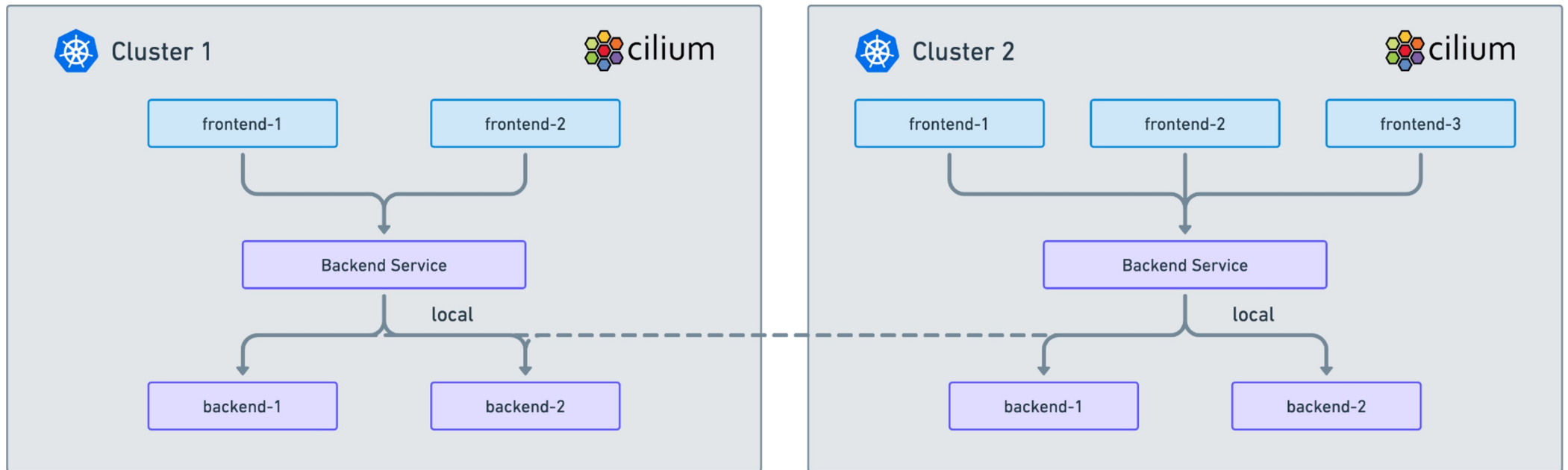


Cluster Mesh - Shared Services



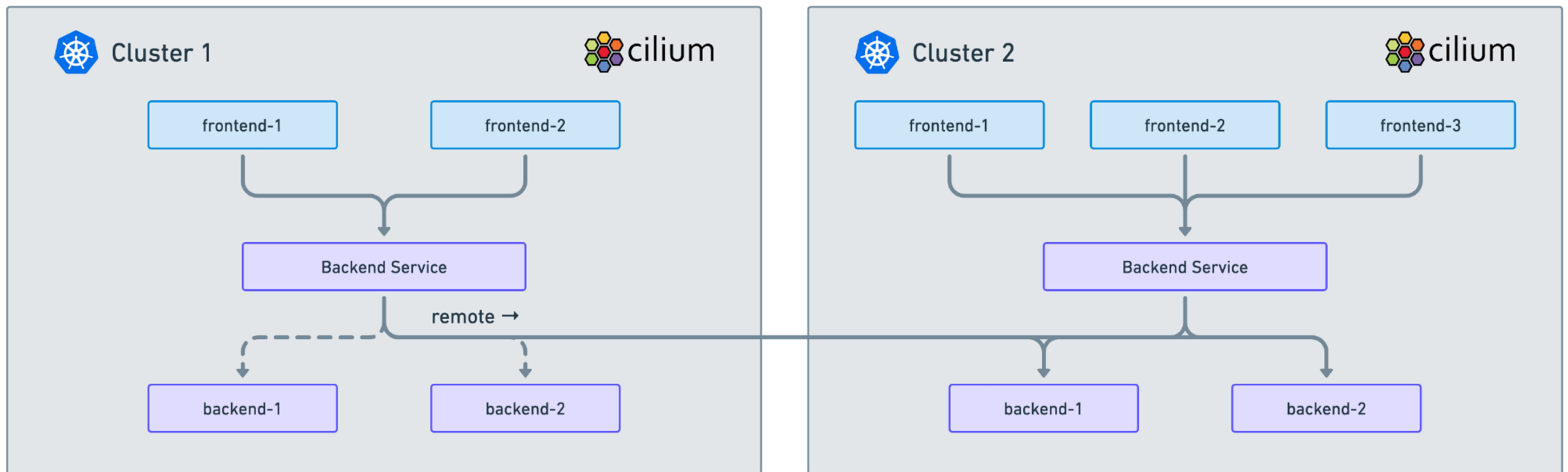


Cluster Mesh - Local Service Affinity





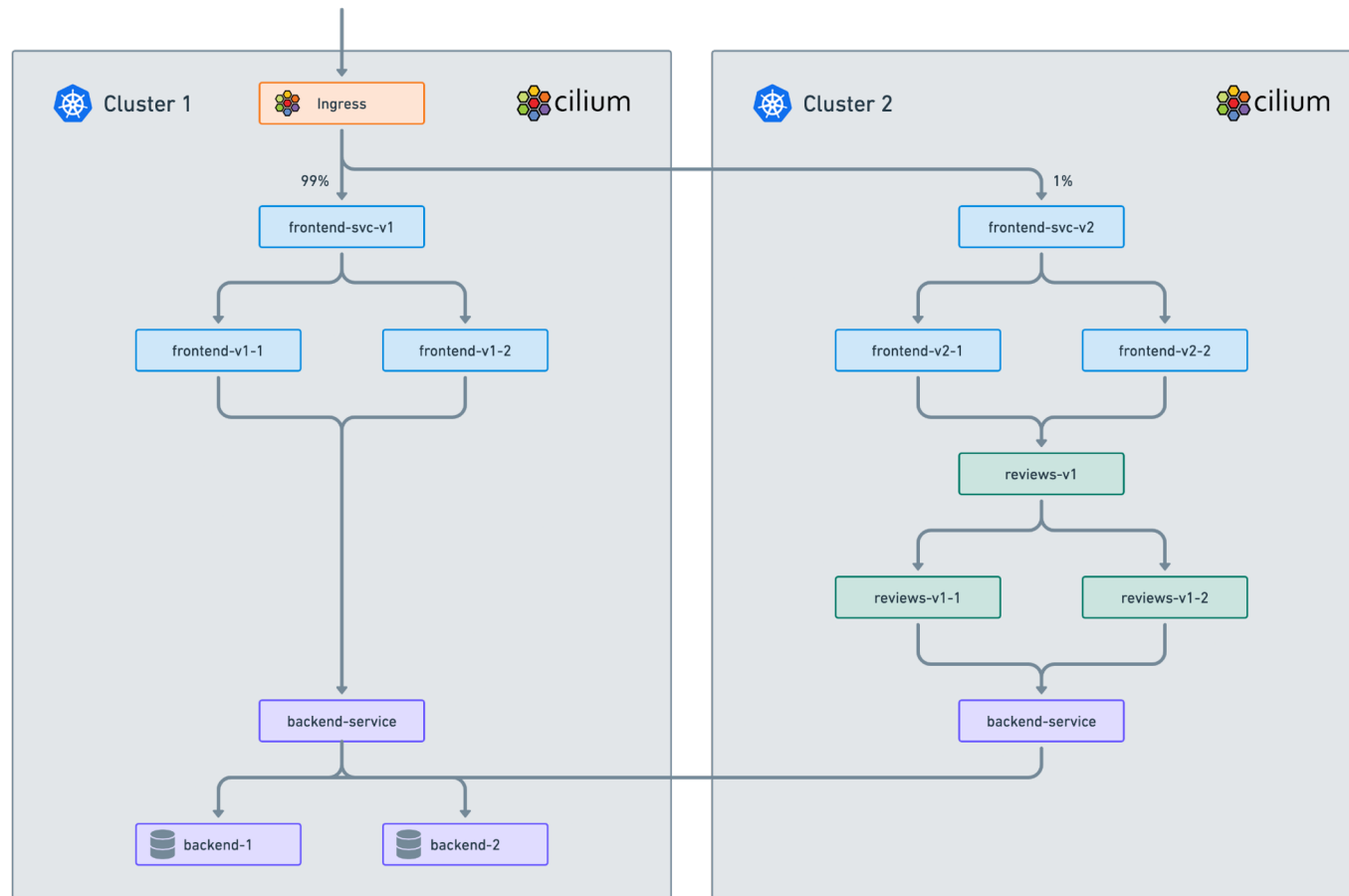
Cluster Mesh - Remote Service Affinity



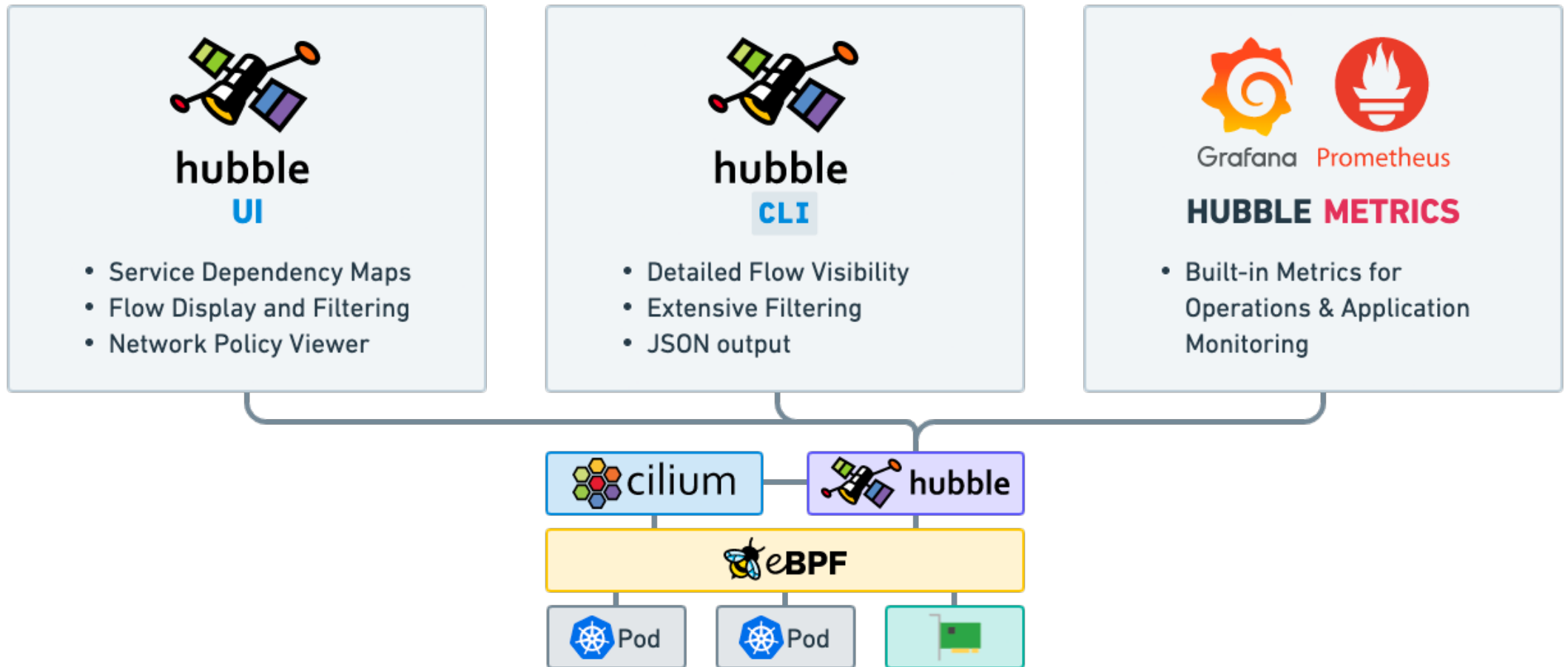


Cluster Mesh with Service Mesh

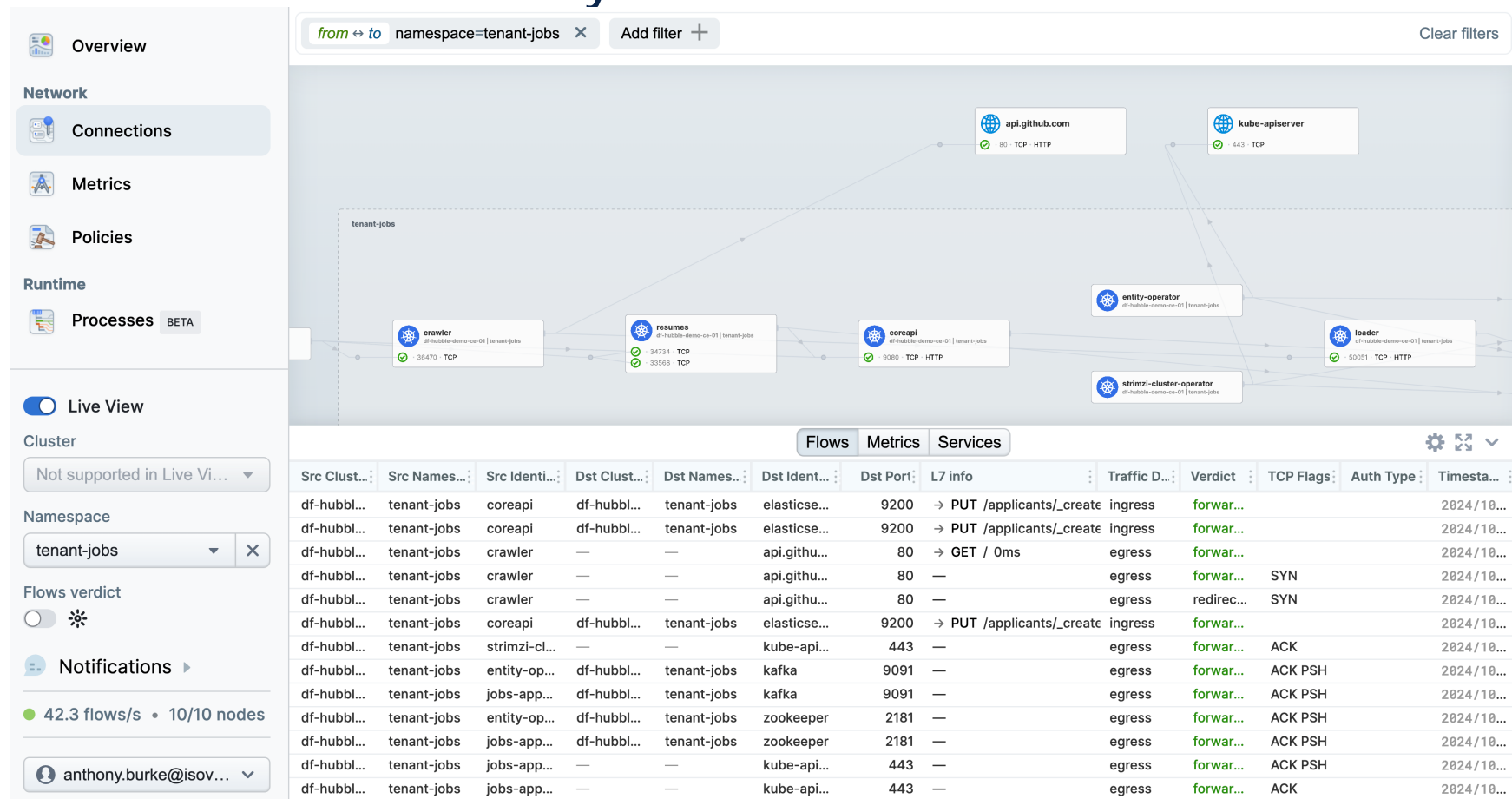
Canary Rollout to other Cluster



Hubble Overview



Hubble Observability



Hubble Network Policy Editor

Overview

Network

Connections

Metrics

Policies

Runtime

Processes BETA

Live View

Cluster

Not supported in Live Vi...

Namespace

tenant-jobs

Flows verdict

Any verdict

Aggregate flows

Network policies

Filters applied

Visualize all

All policies visualized on map

allow-all-within-namespace

dns-visibility

Notifications

42.3 flows/s • 10/10 nodes

anthony.burke@isov...

from to namespace=tenant-jobs

Add filter

Clear filters

Outside Cluster

Any endpoint

In Namespace

Any pod

strimzi.io/cluster=jobs... → :2888|TCP

strimzi.io/cluster=jobs... → :3888|TCP

strimzi.io/cluster=jobs... → :9090|TCP

strimzi.io/kind=cluster... → :2181|TCP

strimzi.io/kind=cluster... → :8443|TCP

strimzi.io/name=jobs-a... → :9091|TCP

Matched selectors

Ingress Default Deny

Egress Default Deny

Outside Cluster

Any endpoint

Any endpoint :80|TCP

In Namespace

Any pod

In Cluster

Everything in the cluster

Kubernetes DNS

DNS proxy on

+ New

K8S

Cilium

```

1 apiVersion: cilium.io/v2
2 kind: CiliumNetworkPolicy
3 metadata:
4   name: jobs-app-network-policy-zookeeper
5   namespace: tenant-jobs
6 spec:
7   endpointSelector:
8     matchLabels:
9       strimzi.io/cluster: jobs-app
10      strimzi.io/kind: Kafka
11      strimzi.io/name: jobs-app-zookeeper
12   ingress:
13     - fromEndpoints:
14       - matchLabels:
15         strimzi.io/cluster: jobs-app
16         strimzi.io/kind: Kafka
17         strimzi.io/name: jobs-app-zookeeper

```

Preferences

Src Identity	Dst Identity	Traffic Direction	Verdict
coreapi	elasticsearch-master	ingress	forwarded
coreapi	elasticsearch-master	ingress	forwarded
coreapi	elasticsearch-master	ingress	forwarded
coreapi	elasticsearch-master	ingress	forwarded
crawler	api.github.com	egress	forwarded
crawler	api.github.com	egress	forwarded
crawler	api.github.com	egress	redirected
coreapi	elasticsearch-master	ingress	forwarded
jobs-app-entity-operator	kube-apiserver	egress	forwarded
coreapi	elasticsearch-master	ingress	forwarded
coreapi	elasticsearch-master	ingress	forwarded
coreapi	elasticsearch-master	ingress	forwarded
coreapi	elasticsearch-master	ingress	forwarded
resumes	coreapi	ingress	redirected
jobs-app-entity-operator	kube-apiserver	egress	forwarded
coreapi	elasticsearch-master	ingress	forwarded
coreapi	elasticsearch-master	ingress	forwarded
coreapi	elasticsearch-master	ingress	forwarded
coreapi	elasticsearch-master	ingress	forwarded

Tetragon



Open Source

- Apache 2.0 (userspace) & GNU GPL (eBPF)
- Part of CNCF as a subproject of Cilium



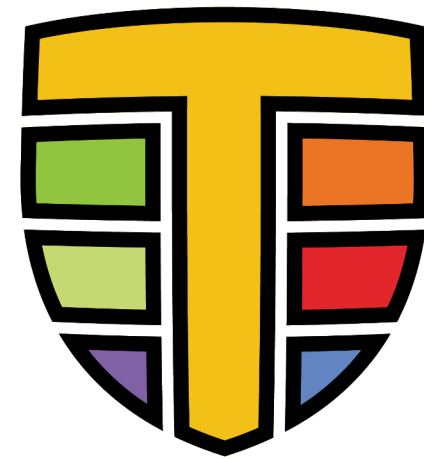
eBPF-based

- Generic low level process events
- In-kernel filtering and enforcement



Kubernetes-native

- Kubernetes metadata in events
- Configuration via custom resources



tetragon

Tetragon Overview

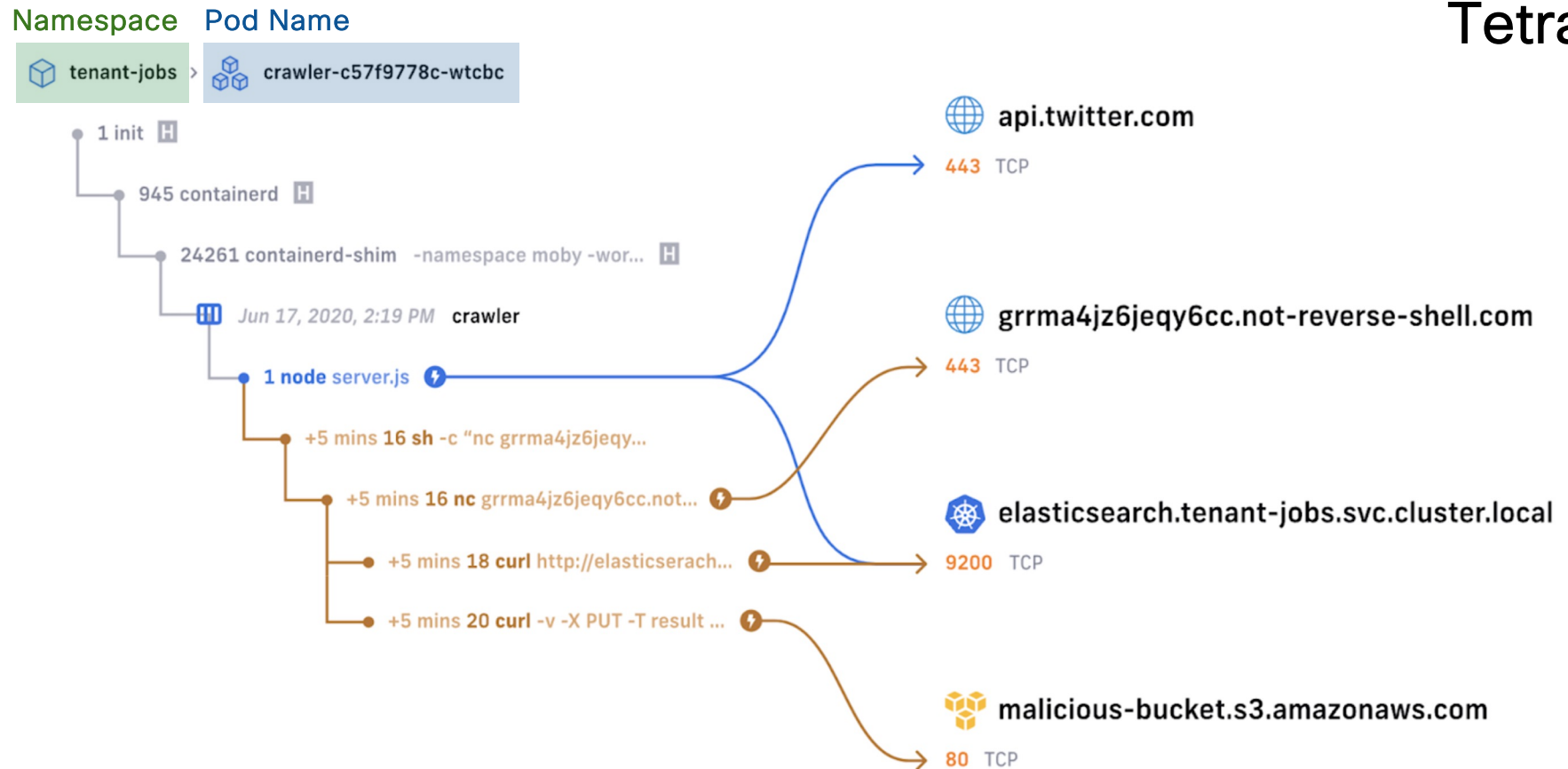


What activities do we care about?

- Network traffic
- File & I/O traffic
- Running of executables
- System Call activity
- Changing privileges & namespace boundaries

Every malicious actor will do one or more of these things

Let's Deep Dive into a Kubernetes Pod



Learn more!

ISOVALENT

For the Enterprise

Hardened, enterprise-grade eBPF-powered networking, observability, and security.

isovalent.com/product

isovalent.com/labs



OSS Community

eBPF-based Networking, Observability, Security

cilium.io

cilium.slack.com

[Regular news](#)



Base technology

The revolution in the Linux kernel, safely and efficiently extending the capabilities of the kernel.

ebpf.io

[What is eBPF? - ebook](#)

ISOVALENT

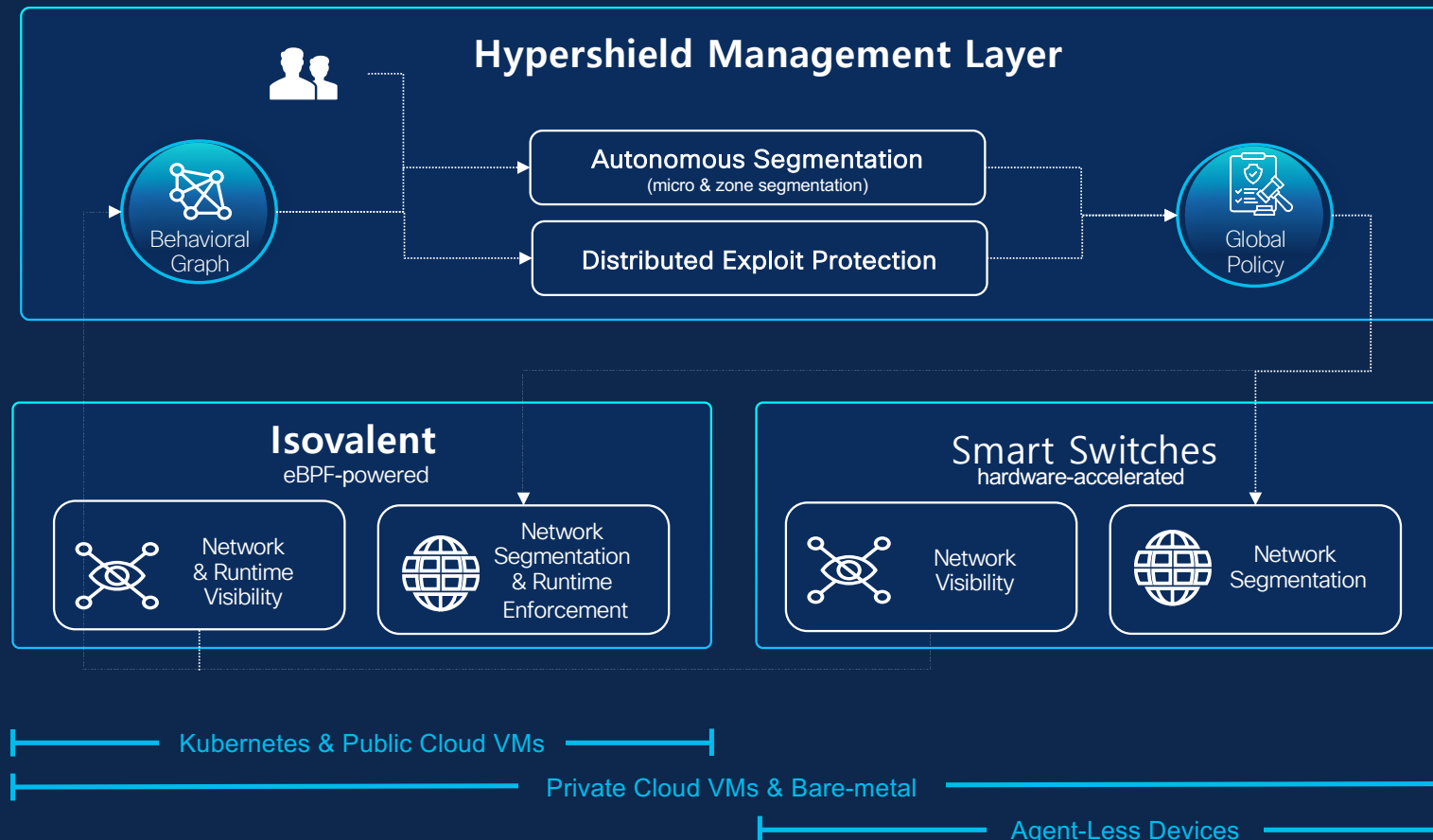
From Cilium to Hypershield



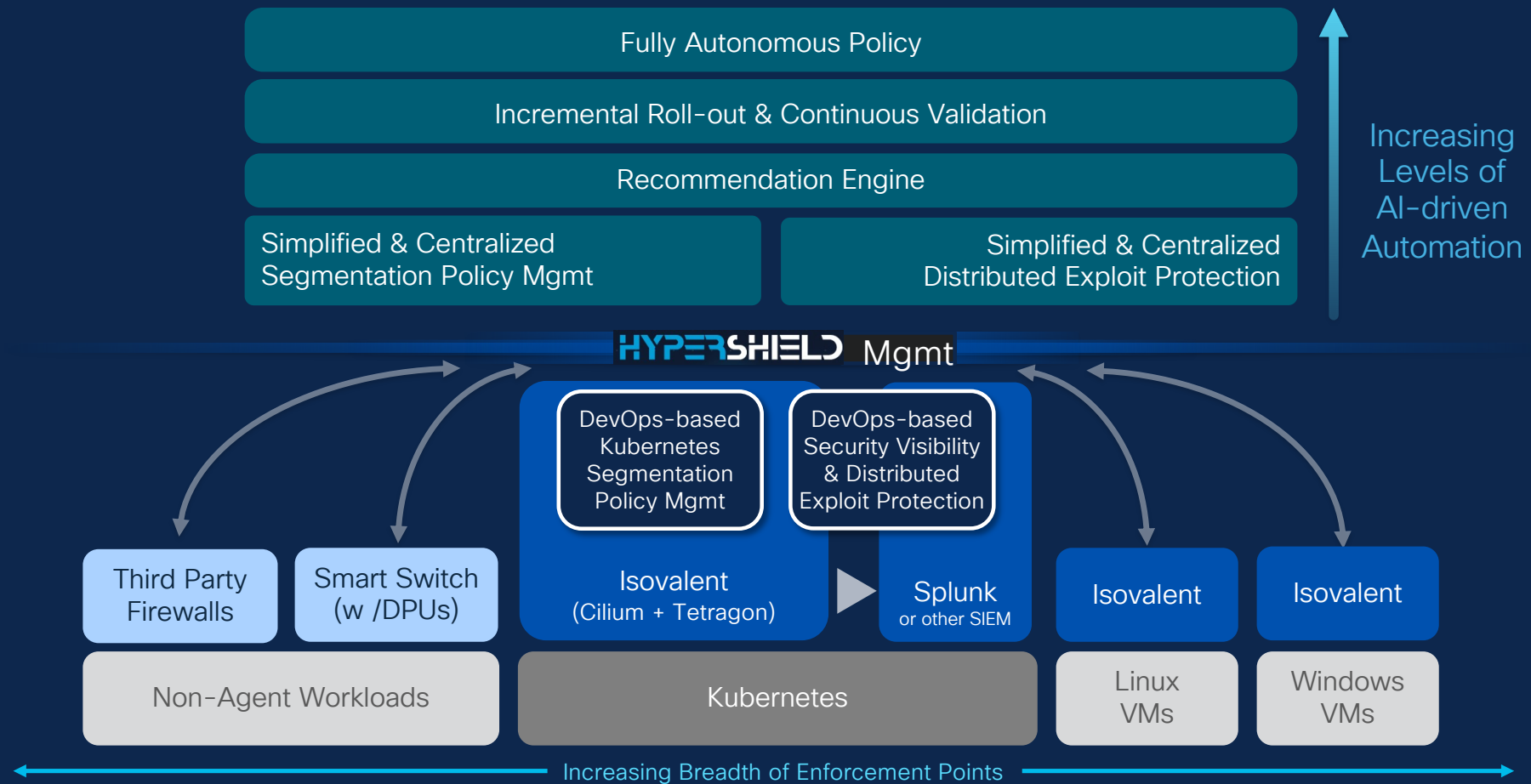
© 2025 Cisco and/or its affiliates. All rights reserved.



Hypershield Architecture



Isovalent as a starting point for Hypershield adoption



Cisco N9300 Series Smart Switches

Best of breed platforms for data center services

Target General availability April 2025

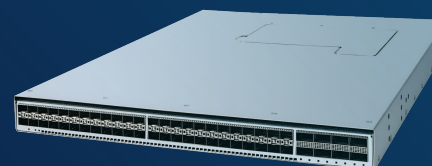


N9324C-SE1U

24-port 100G

- Cloud Edge, Zone-Based segmentation, DCI, Top-of-Rack
- 800G Services Throughput
- Silicon One E100 ASIC + AMD DPUs

Target General availability July 2025



N9348Y2C6D-SE1U

48-port 25G, 6-port 400G, 2-port 100G

- DC Top-of-Rack
- 800G Services Throughput
- Silicon One E100 + AMD DPUs

Cisco and AMD – Better Together

Unmatched flexibility, performance, and efficiency



- Rich NX-OS Features and Services
- High-speed connectivity and scalable performance
- Optimized for latency and power efficiency



Routing
Switching



EVPN/MPLS/
VXLAN/SR



Rich
Telemetry



Line-rate
Encryption



Power
Efficiency



- Software-defined Stateful Services
- Programmable at all layers: add new services without HW change
- Scale-out services with wire-rate performance
- Power down DPU complex when not used



Large-Scale
NAT



IPSEC
Encryption



Distributed
Firewall



Event-Based
Telemetry

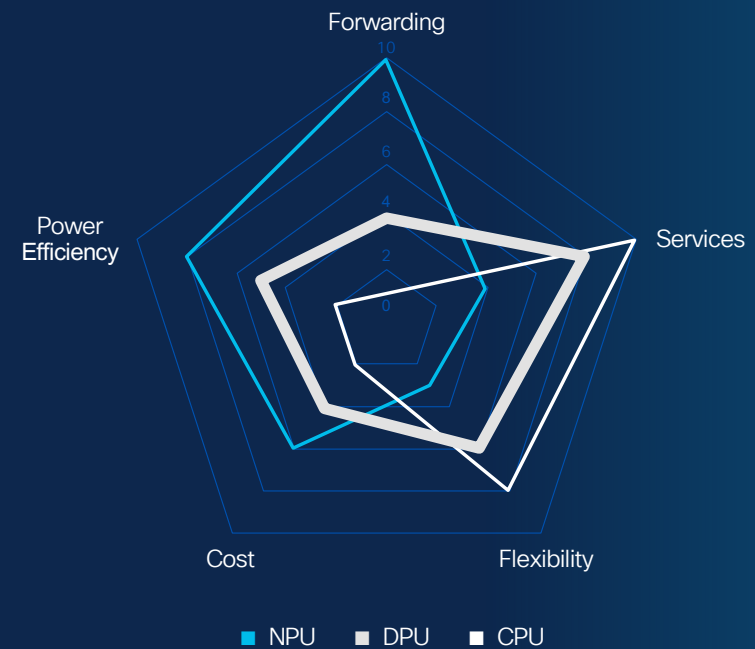
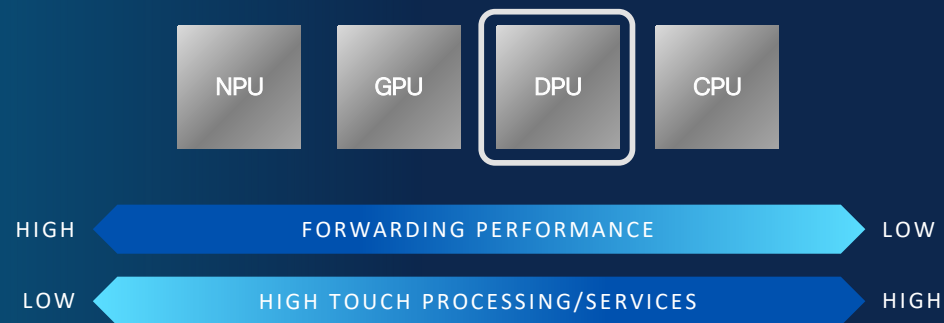


DoS
Protection



Data Processing Units

A game changer to deliver network services at scale



A Platform to Enable Stateful Services

Cisco N9300 Series Smart Switches

Telemetry and Analytics

- Packet inspection
- Timestamping
- DDoS offload

Performance and Optimization

- Virtual network bridging
- High scale traffic filtering

Security and Policy

- Stateful segmentation
- Service chaining/steering

Network services offload

- Encryption
- Large scale NAT

Hypershield
recommends
segmentation
policies based
on observations

Security Cloud Control

Search

Nik Business Corp, Inc

Dashboard

AI Ops Insights

Inventory

Policies

Objects

Connectivity

Troubleshoot and Logs

Settings

Application Dependency Map

Search or describe what you're looking for

Environments

Asset risk

Environment criticalities

Applications

Filters

Custom hierarchy

Environments > Production

Web app

Web app

database

frontend

analytics

apache server 02

Kubernetes Service 05

Legend

Web app

Application

Data

Recommendations 1

Identifiers

Enforcement available

It is recommended you enforce the security of this application with Autonomous Segmentation policies. These policies were generated by Hypershield from observed behaviors and tested in your environment with live traffic.

View policies in map

Allow Web app frontend can access database

Allow Web app frontend can access analytics

Allow Web app analytics can access database

Default allow and observe Web app policy...

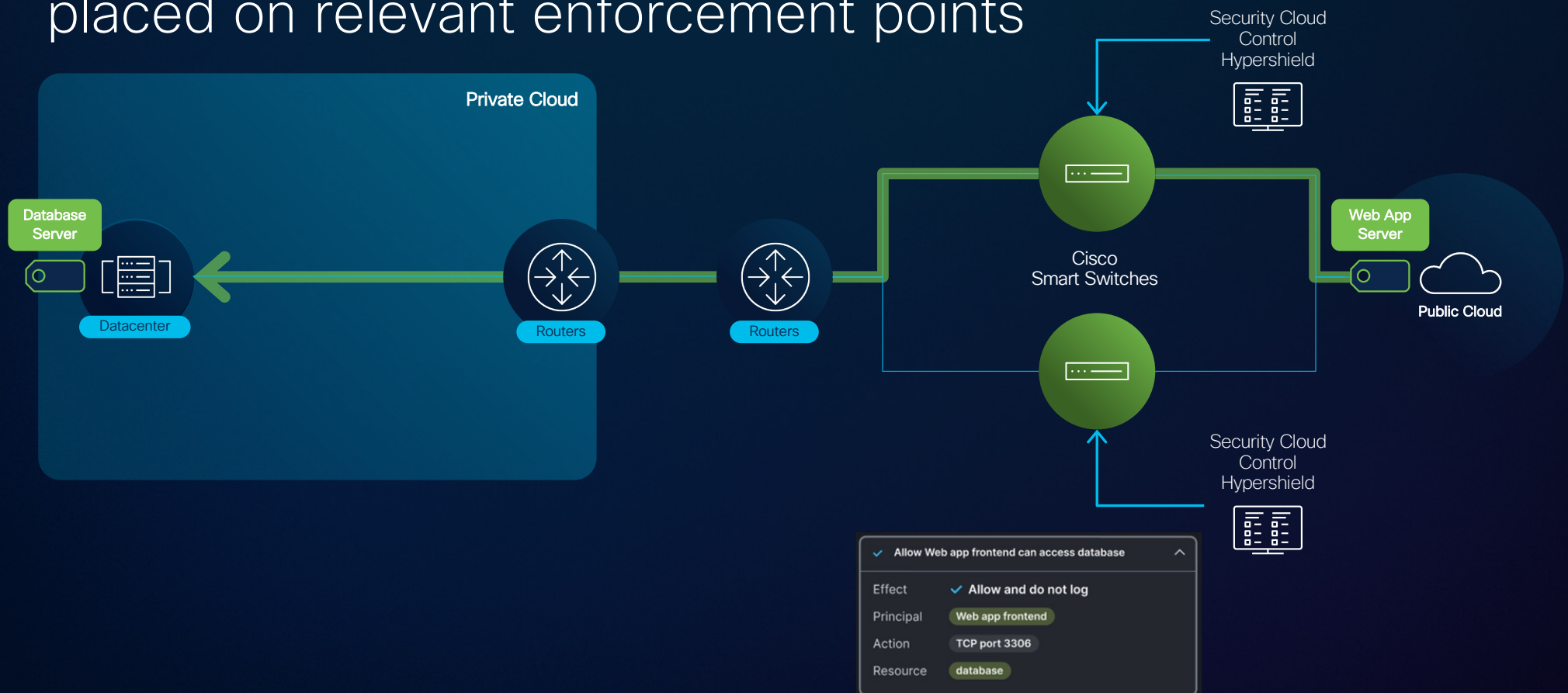
Why are these policies recommended?

Deploy

View test results

Archive

Hypershield policy is automatically placed on relevant enforcement points





Děkujeme za Vaši pozornost

Následující Tech Club webinář:

15.4. Co je nové v SP a ve světě rozlehlých sítí?

Přednášející: Peter Morvay

Registrovat se můžete na oficiálním webu **Cisco Tech Club webináře**



