
white
duck

Next-level Kubernetes networking with Cilium

ContainerDays Hamburg 2023

Who we are

Philip Welz
(Senior DevOps & Kubernetes Engineer, Azure MVP)

+49 8031 230159-0

philip.welz@whiteduck.de

@philip_welz

www.linkedin.com/in/philip-welz

Nico Meisenzahl
(Head of DevOps Consulting & Operations, Cloud Solution Architect)

+49 8031 230159-0

nico.meisenzahl@whiteduck.de

@nmeisenzahl

www.linkedin.com/in/nicomeisenzahl

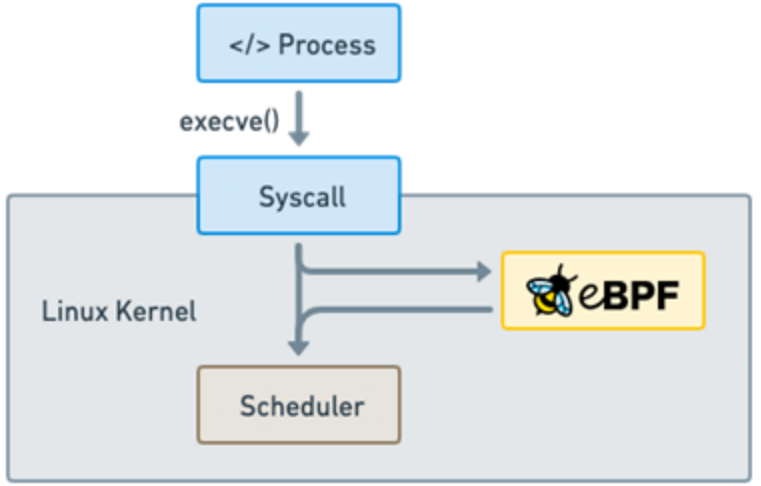
Agenda

- Cilium & eBPF introduction
- Zero trust networking and observability with Cilium CNI & Hubble
- Seamless multi-cluster connectivity with Cilium Cluster Mesh
- Application-centric networking with Cilium Service Mesh
- Cilium Mesh – one mesh to connect them all

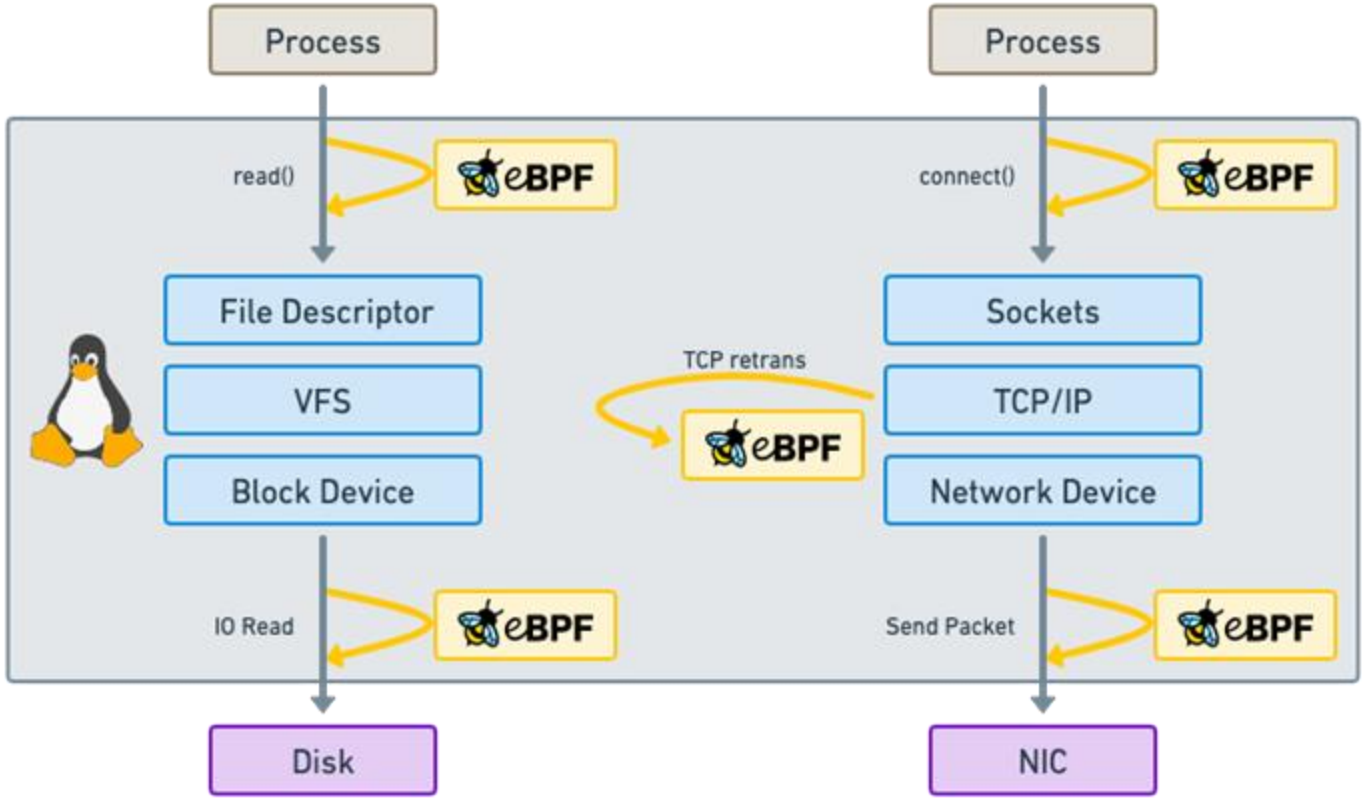
Cilium & eBPF introduction

What is eBPF?

- “What JavaScript is to the browser, eBPF is to the Linux Kernel”
- Makes the Linux kernel programmable in a secure and efficient way



eBPF programs act on events

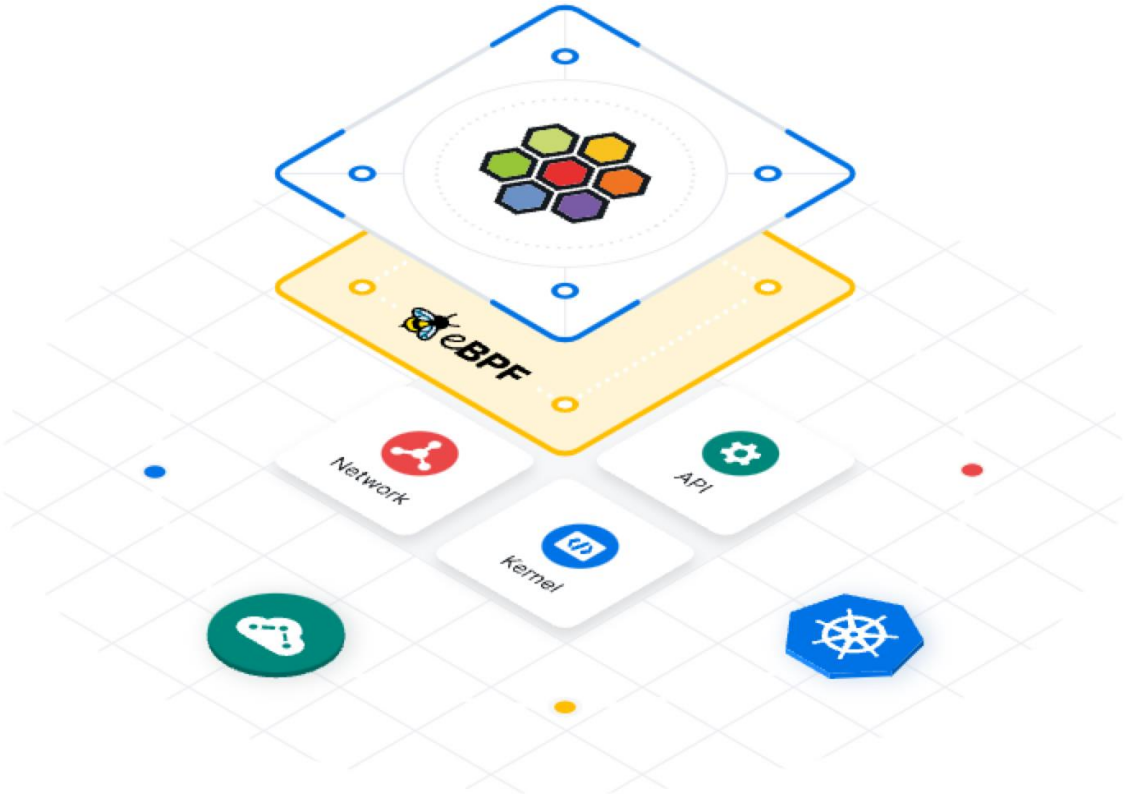


Attachment points:

- Kernel functions (kprobes)
- Userspace functions (uprobe)
- System calls
- Tracepoints
- Sockets
- Network devices
- …

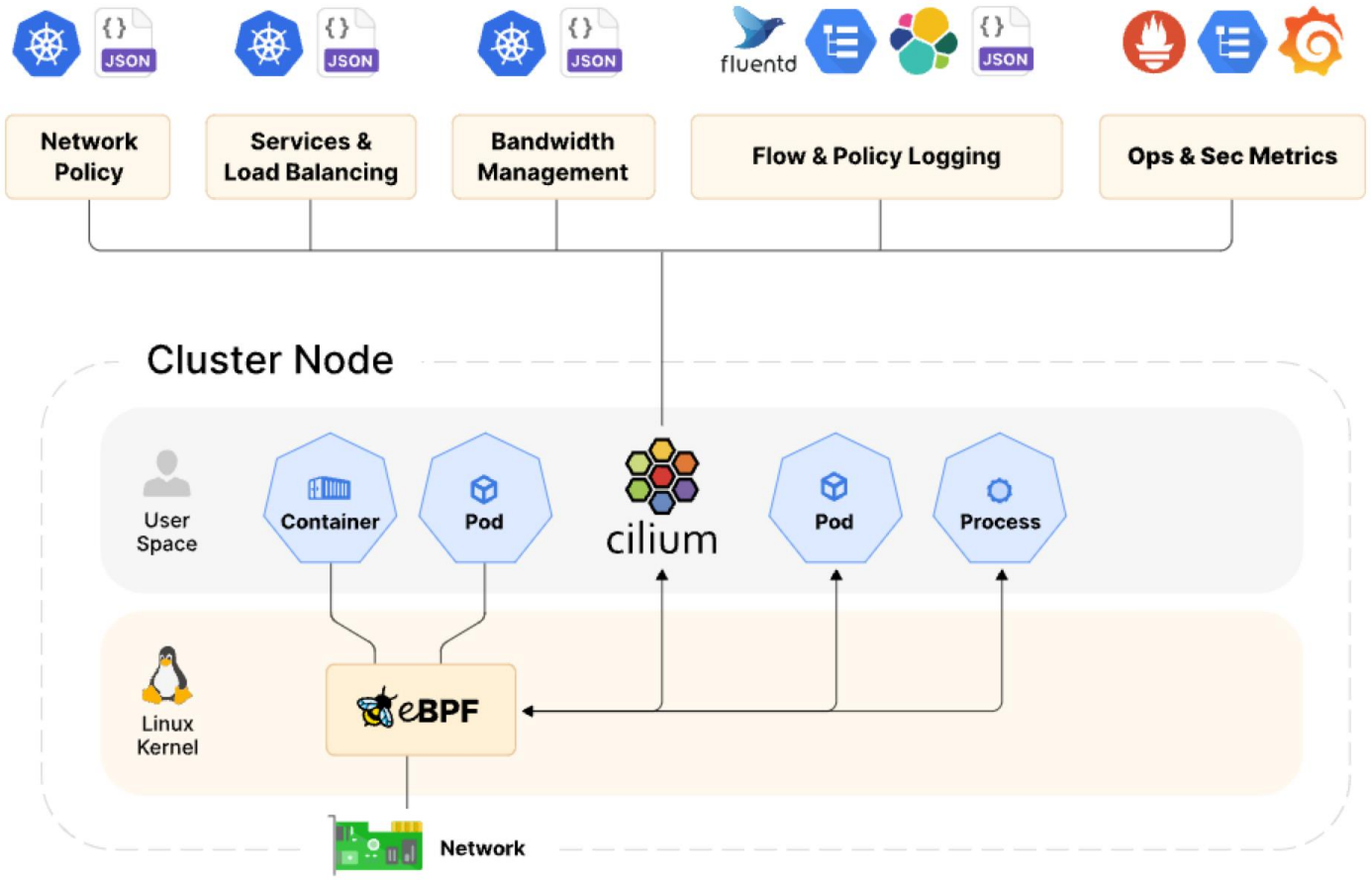
What is Cilium?

- „eBPF-based Networking, Observability, Security“
- A suite containing of
  - Cilium CNI
  - Hubble
  - Cilium Mesh
  - Cluster Mesh
  - Service Mesh
  - Tetragon (not covered today)
  - Isovalent Cilium Enterprise (not covered today)



Zero trust networking and Observability with Cilium CNI & Hubble

Cilium CNI (Container Network Interface)



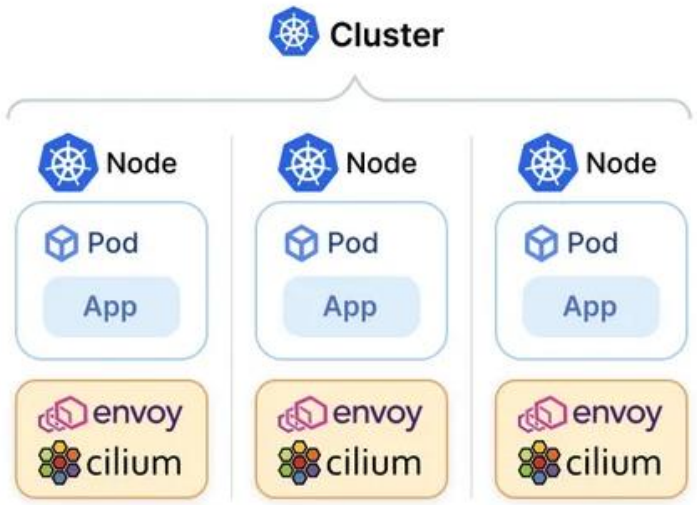
Helps with:

- Enhanced networking speed
  - Abstracts kube-proxy
- Advanced Network Policies
- Traffic encryption
- Load-Balancing

Cilium agent & Envoy proxy

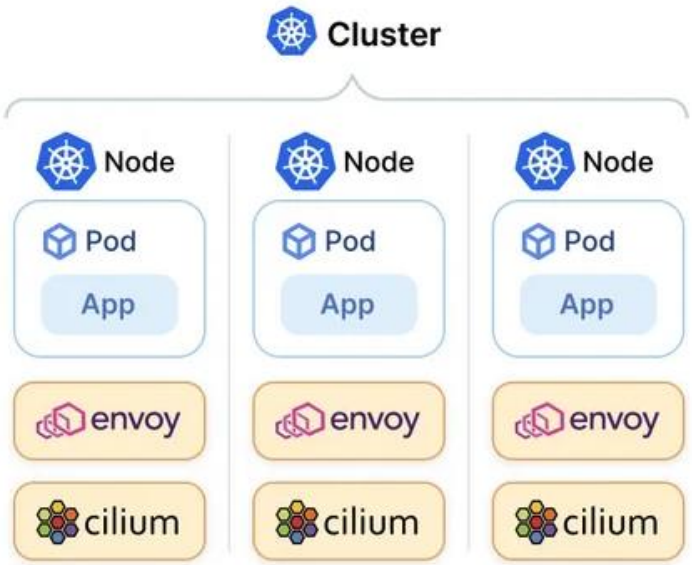
Cilium agent:

- Deployed on every node
- Injects eBPF program to the node
- Load-Balancing

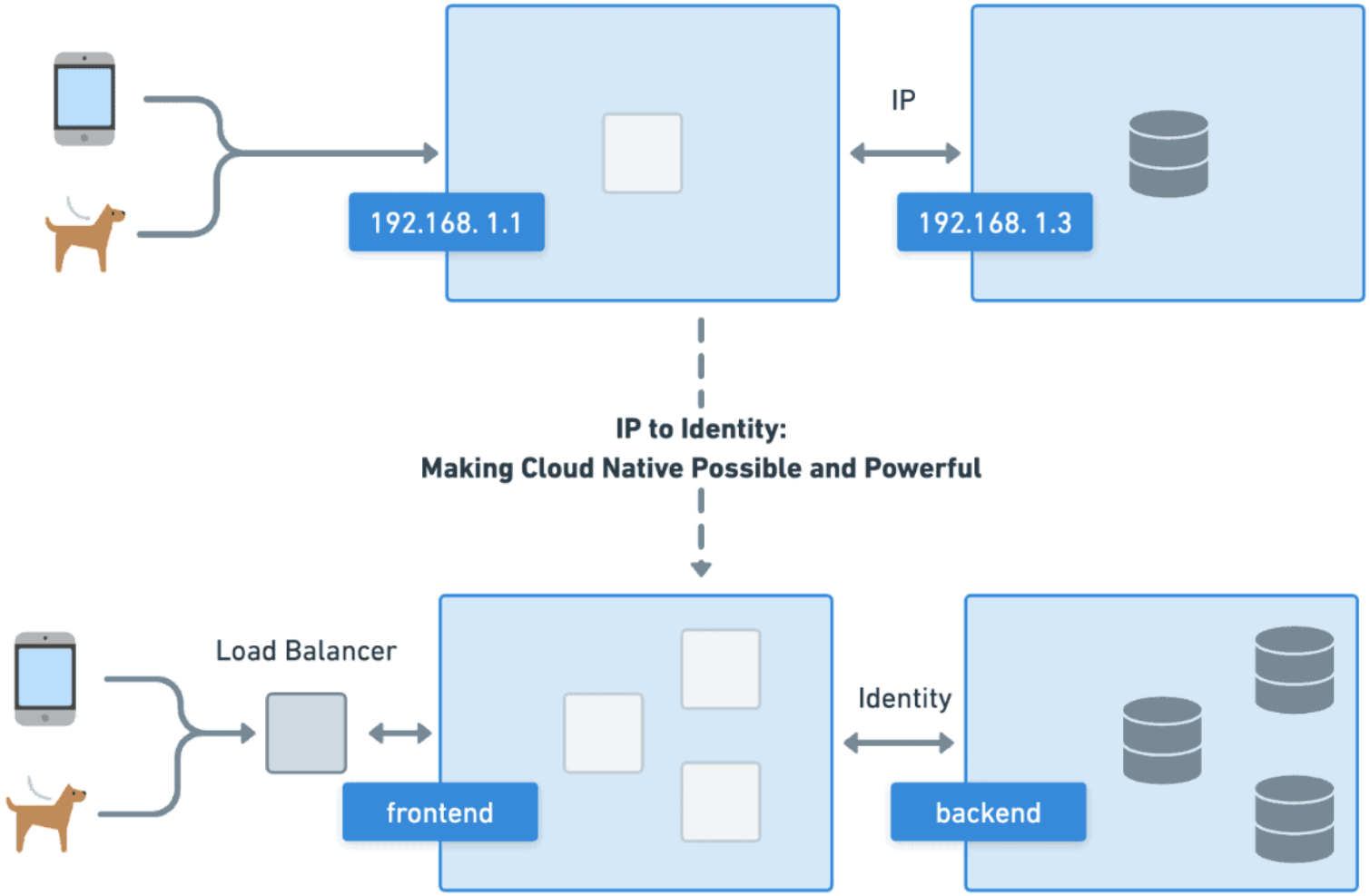


Envoy Proxy:

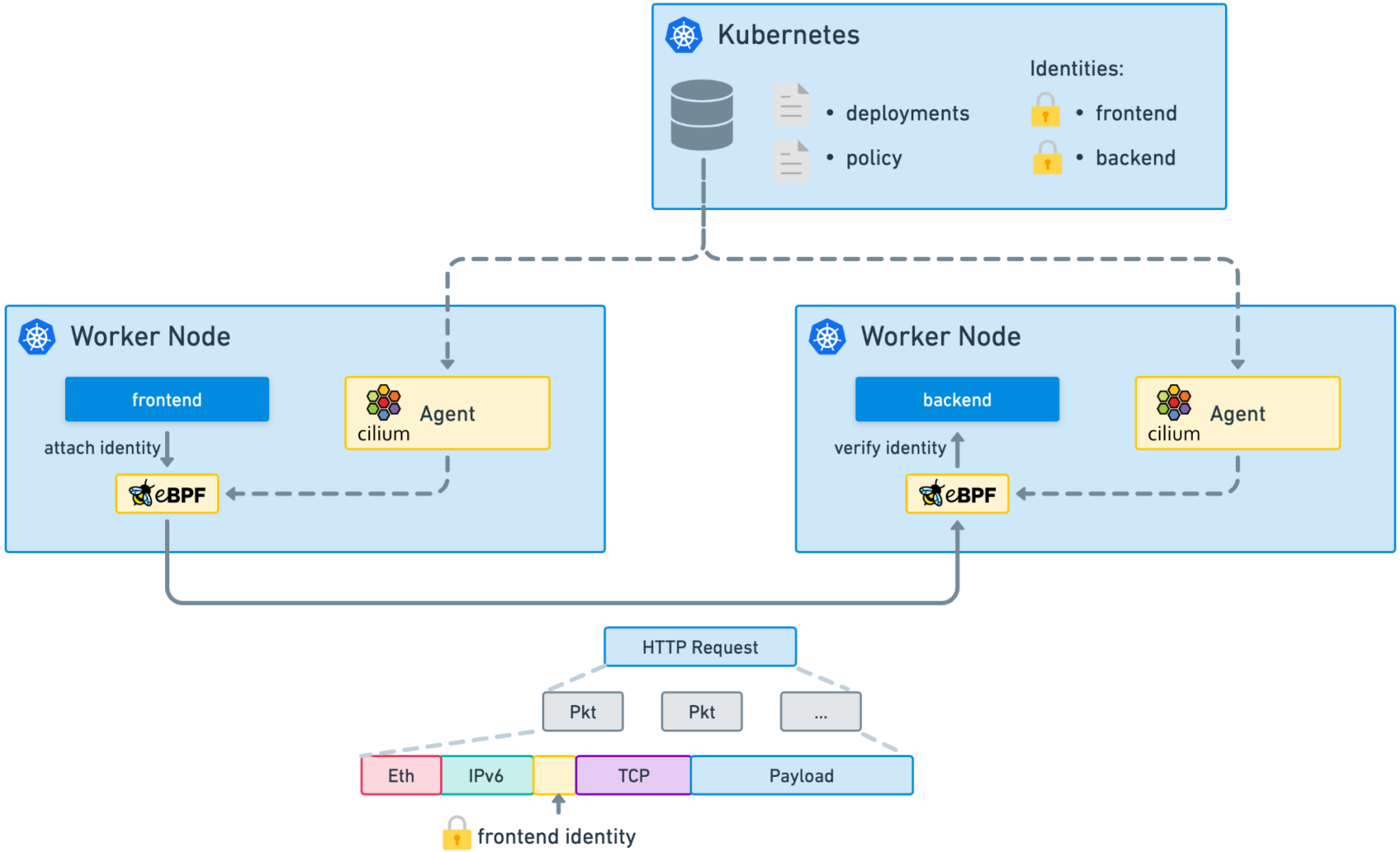
- Seperate process within the Cilium agent pod
- L7
- Ingress / Gateway API
- Expected default mode soon



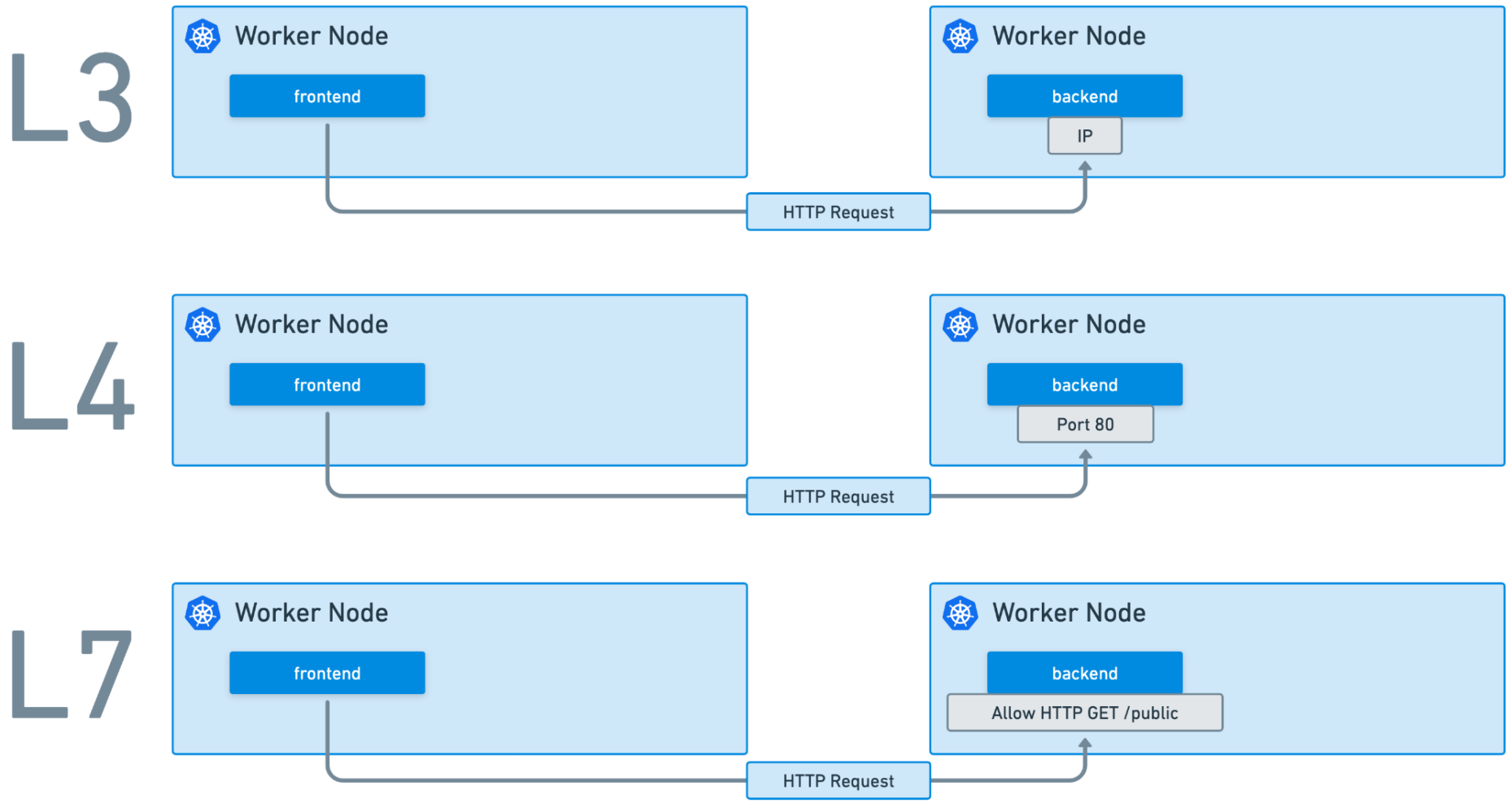
Identity-based Network Security



Identity-based Network Security



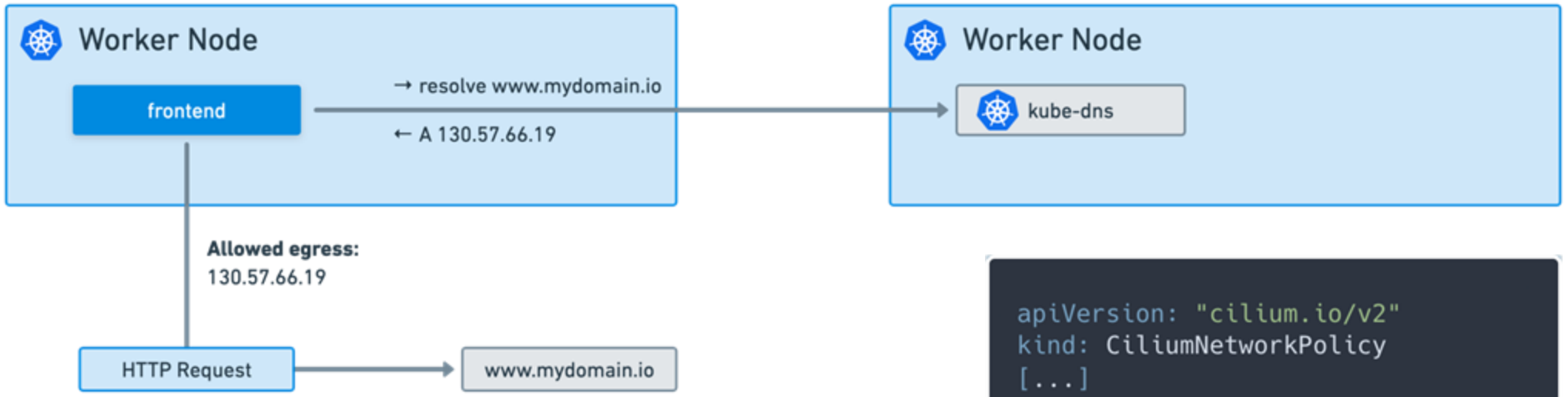
API-aware Authorization



HTTP-Aware Cilium Network Policy

```yaml
apiVersion: "cilium.io/v2"
kind: CiliumNetworkPolicy
metadata:
  name: "http-aware-rule"
spec:
  description: "L7 policy to restrict access to specific HTTP call"
  endpointSelector:
    matchLabels:
      role: frontend
  ingress:
  - fromEndpoints:
    - matchLabels:
        role: frontend
    toPorts:
    - ports:
      - port: "80"
        protocol: TCP
      rules:
        http:
        - method: "GET"
          path: "/public"
```

DNS-aware Cilium Network Policy



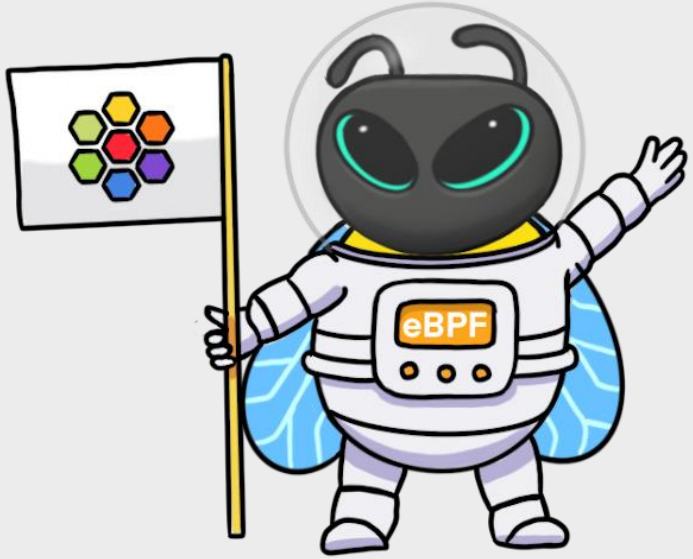

```
apiVersion: "cilium.io/v2"
kind: CiliumNetworkPolicy
[...]
specs:
- endpointSelector:
  matchLabels:
    app: frontend
  egress:
  - toFQDNs:
    - matchName: "*.mydomain.io"
    toPorts:
    - ports:
      - port: "443"
        protocol: TCP
```

Network Observability with Hubble

Demo: Cilium Identity & Hubble

- observe network traffic with Hubble CLI & UI
- inspect Cilium Identities
- create L3-L4 Cilium network policy
- make L7 traffic visible and add L7 Cilium network policy

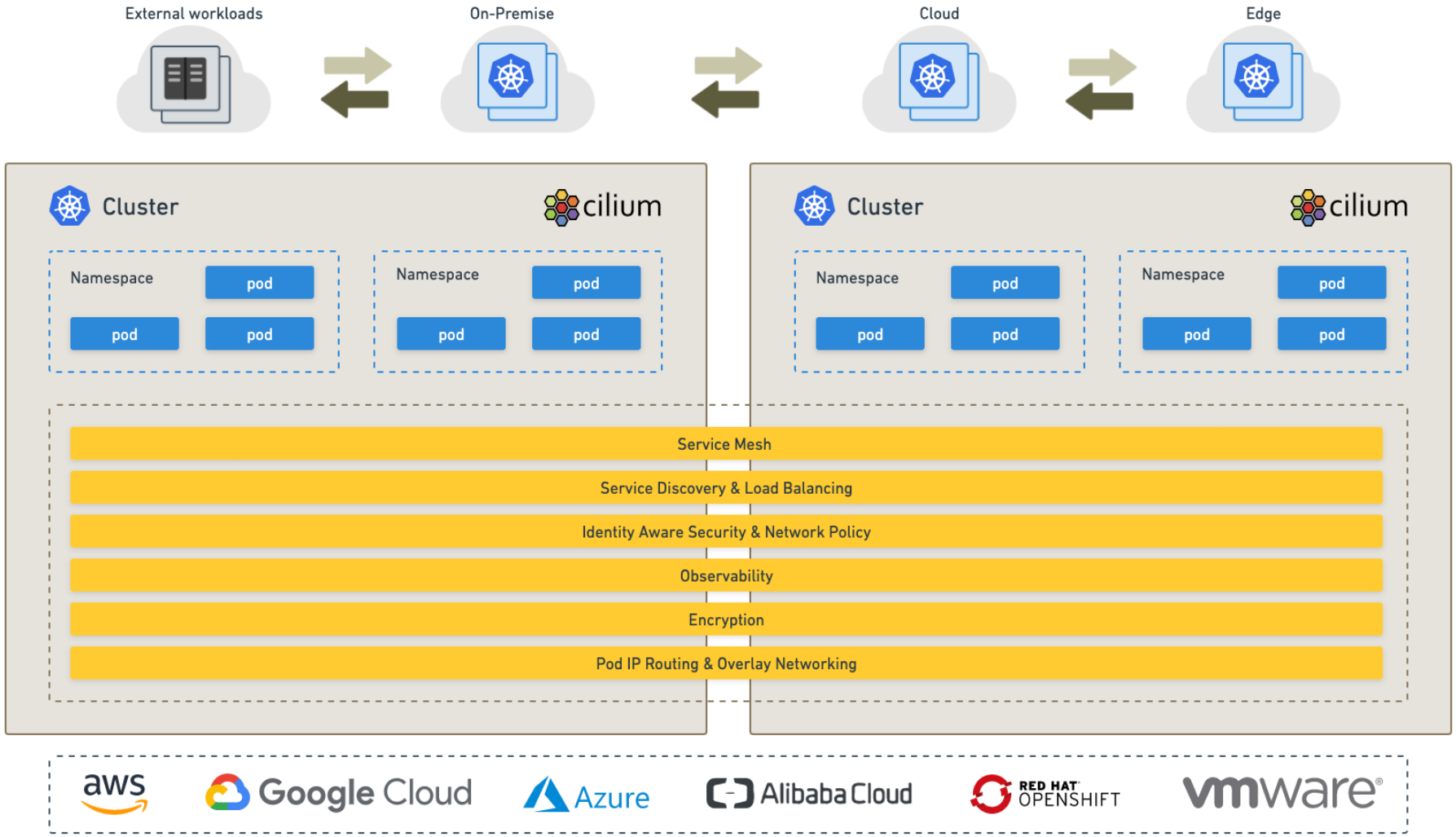
Seamless multi-cluster connectivity with Cluster Mesh



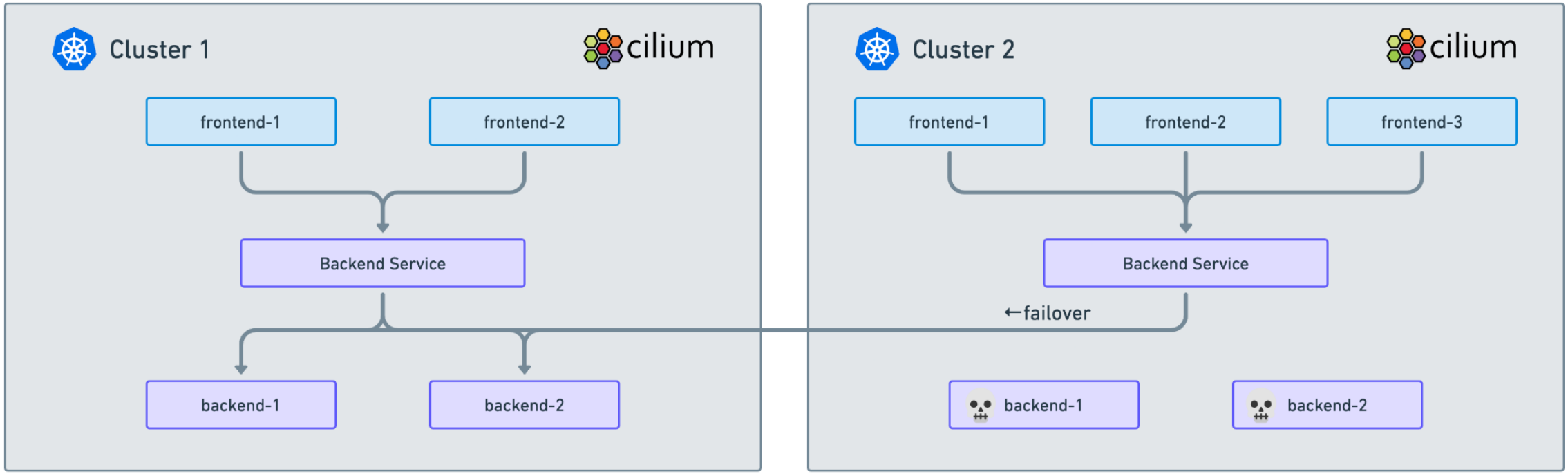
Cluster Mesh

- „Seamless Connectivity For Multiple Kubernetes Clusters“
- Helps with multi-cluster
  - High availability and fault tolerance
  - Transparent service discovery
  - Shared services across clusters
  - Effortless Pod IP routing (via direct-routing or tunneling)

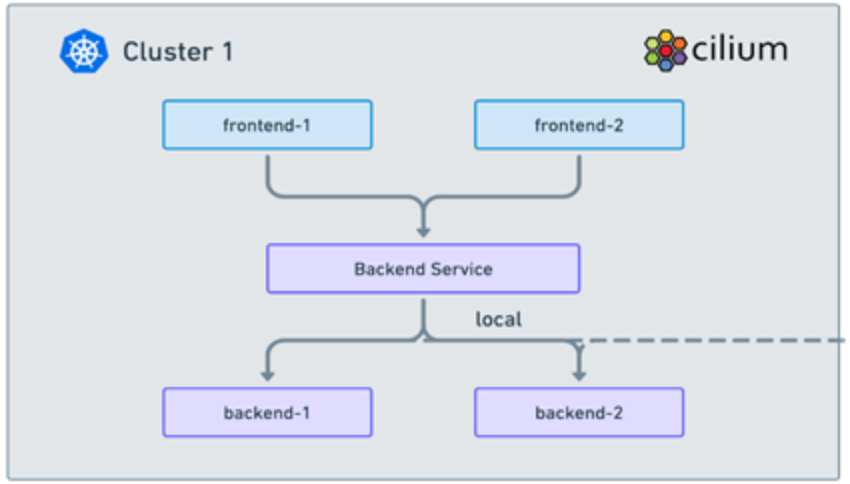
Big Picture

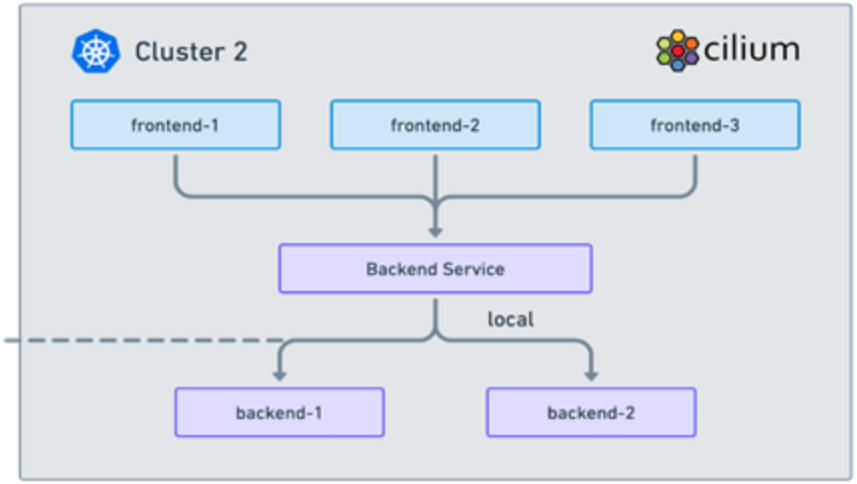

External workloads
On-Premise
Cloud
Edge
Cluster
cilium
Namespace
pod
Cluster
cilium
Namespace
pod
Service Mesh
Service Discovery & Load Balancing
Identity Aware Security & Network Policy
Observability
Encryption
Pod IP Routing & Overlay Networking
aws
Google Cloud
Azure
Alibaba Cloud
RED HAT OPENSHIFT
vmware

High Availability

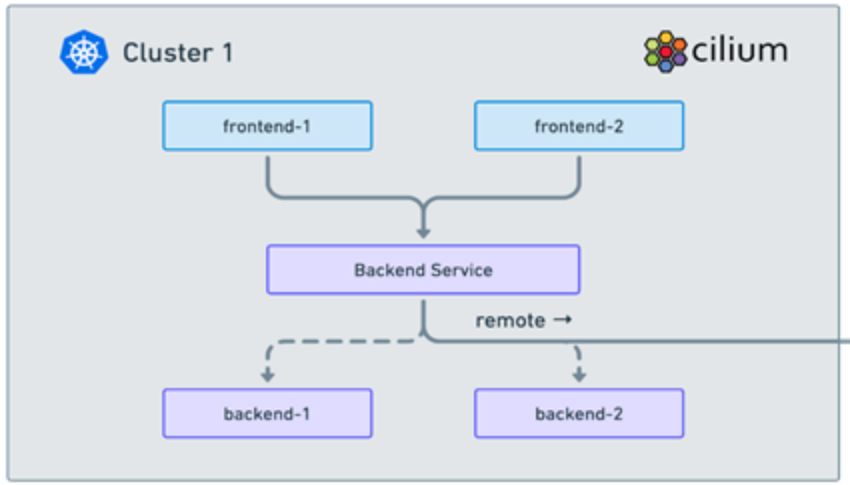


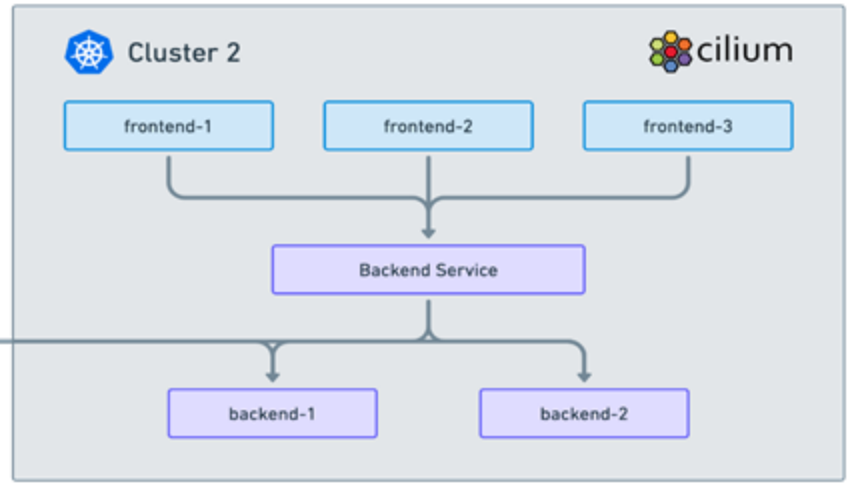
Cluster Network Affinity





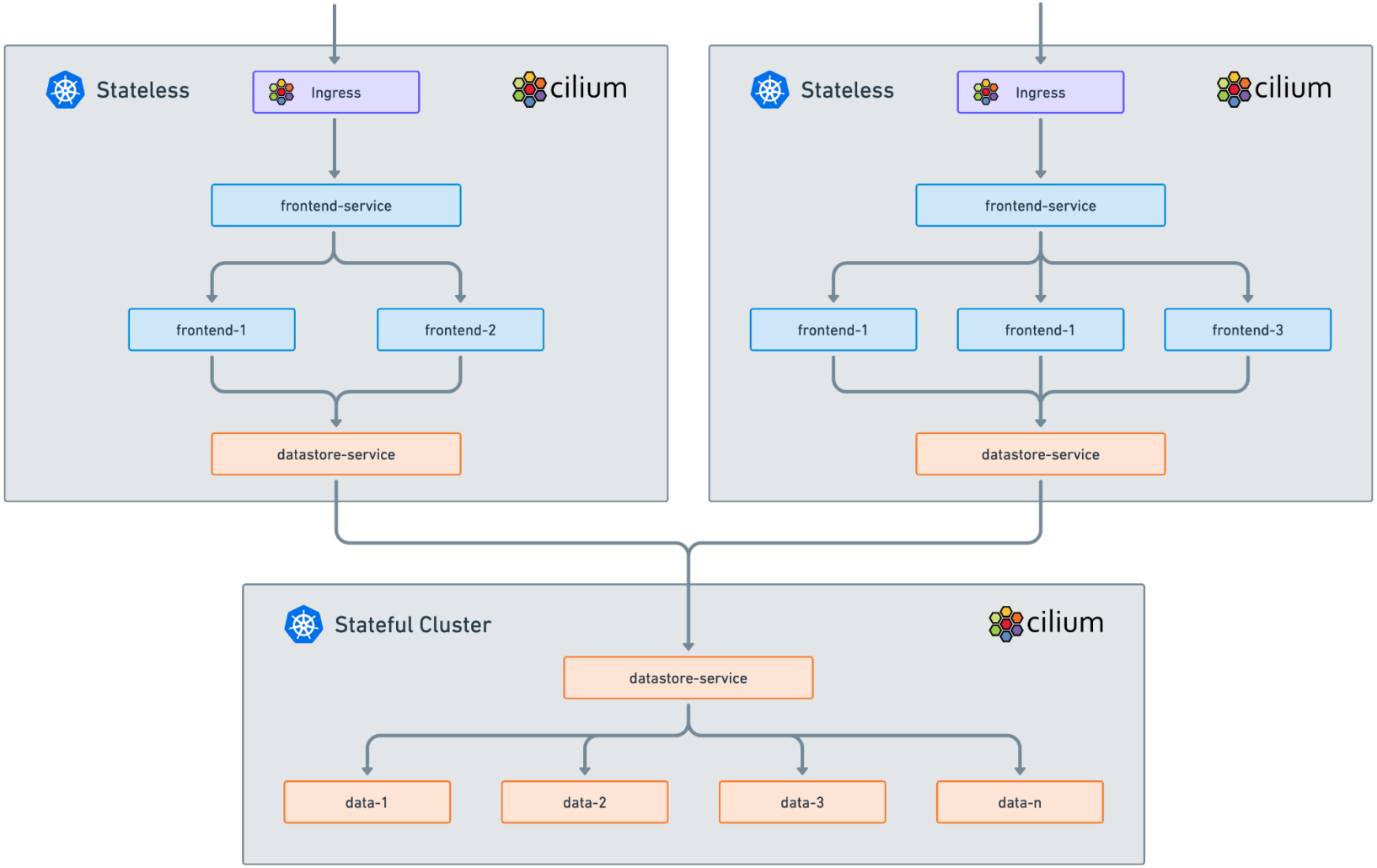

```yaml
apiVersion: v1
kind: Service
metadata:
  name: backend-service
  annotations:
    io.cilium/global-service: "true"
    io.cilium/service-affinity: local
spec:
  type: ClusterIP
  ports:
  - port: 80
  selector:
    name: backend
```






```yaml
apiVersion: v1
kind: Service
metadata:
  name: backend-service
  annotations:
    io.cilium/global-service: "true"
    io.cilium/service-affinity: remote
spec:
  type: ClusterIP
  ports:
  - port: 80
  selector:
    name: backend
```

Splitting Services across Clusters



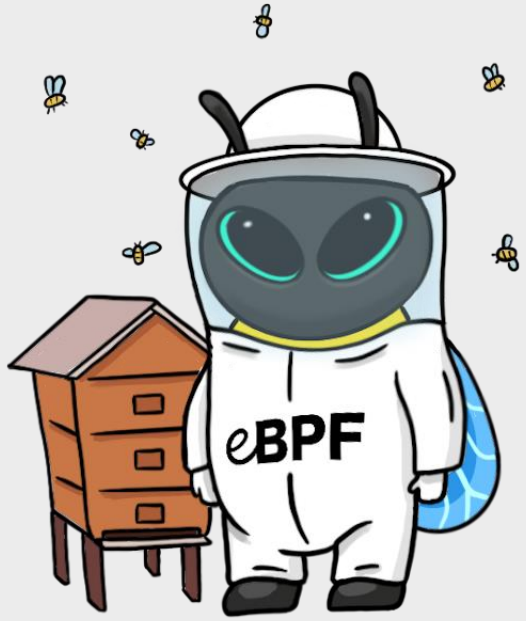
Cilium Network Policies across Clusters

```yaml
apiVersion: "cilium.io/v2"
kind: CiliumNetworkPolicy
metadata:
  name: "ingress-to-rebel-base"
spec:
  description: "Allow x-wing in cluster-1 to contact rebel-base in cluster2"
  endpointSelector:
    matchLabels:
      name: rebel-base
      io.cilium.k8s.policy.cluster: cluster-2
  ingress:
  - fromEndpoints:
    - matchLabels:
        name: x-wing
        io.cilium.k8s.policy.cluster: cluster-1
    toPorts:
    - ports:
      - port: "80"
        protocol: TCP
```

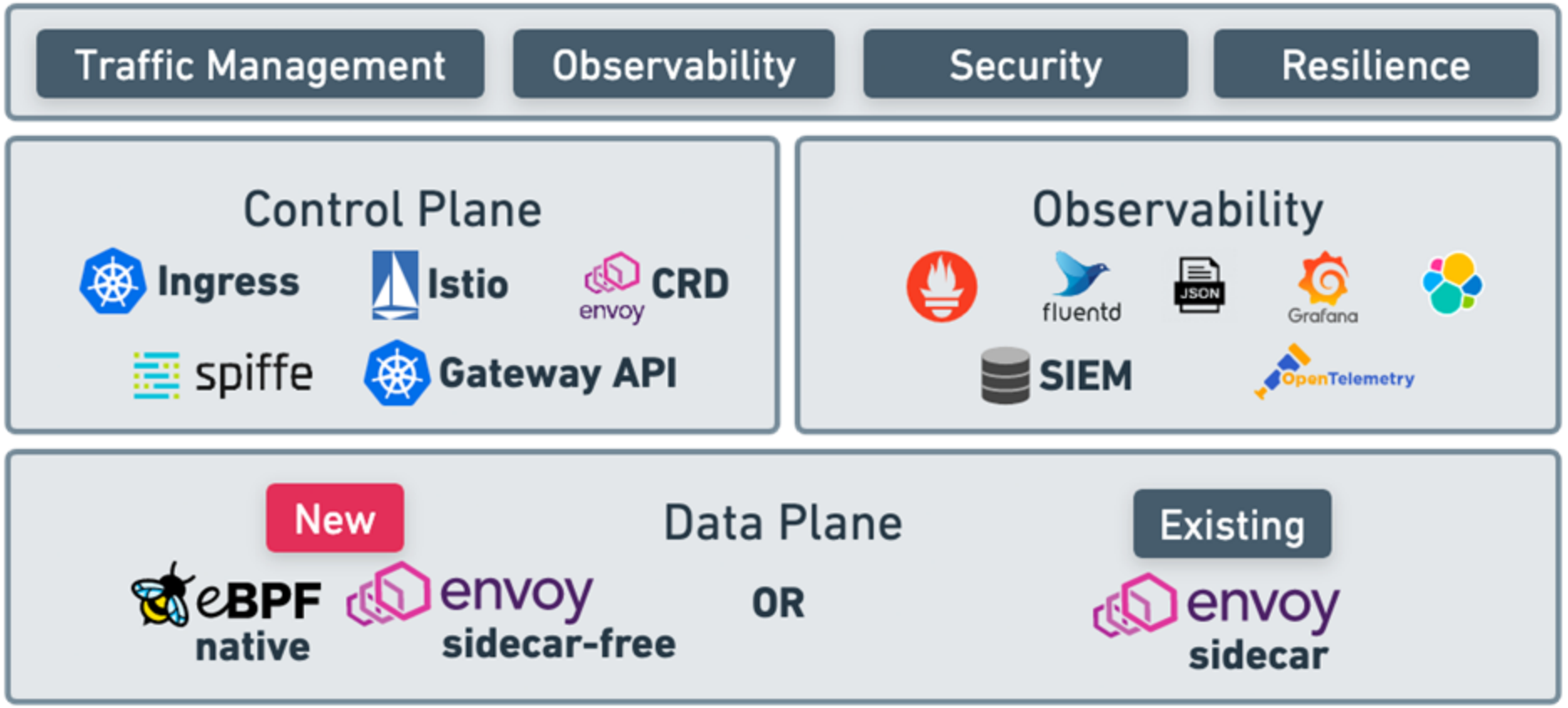
Demo: Global Service with Cluster Mesh

- deploy application into two cluster
  - Service must exist on both Clusters with the same name and in the same namespace
- verify high availability by scaling down the app to zero on cluster01
- create cross cluster network policy

Application-centric networking with Cilium Service Mesh



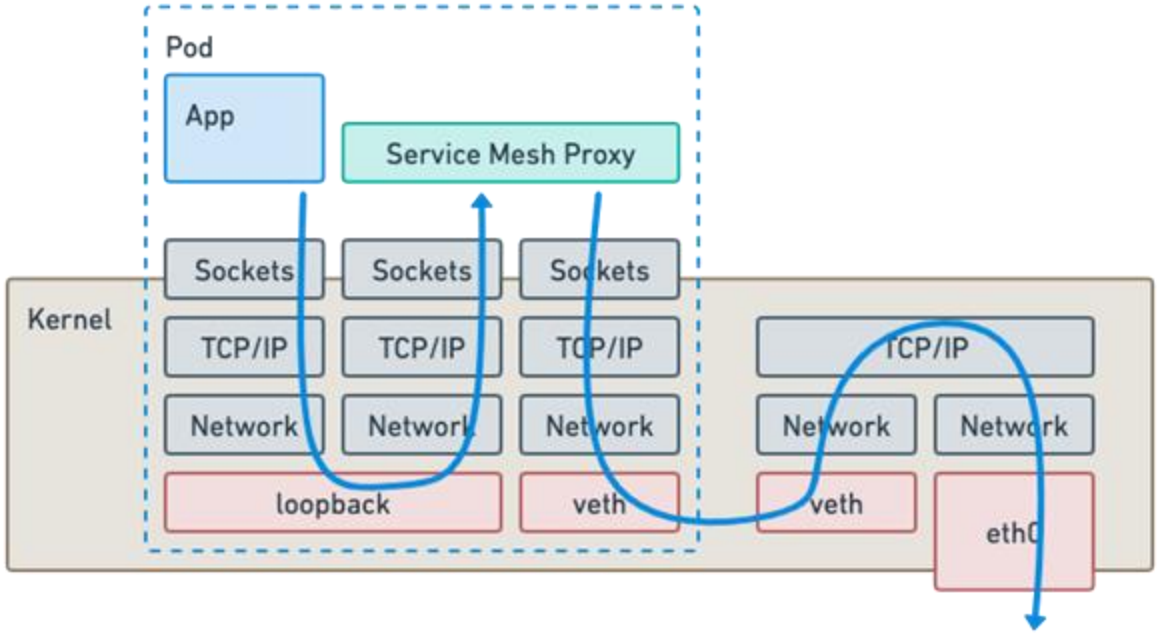
Cilium Service Mesh

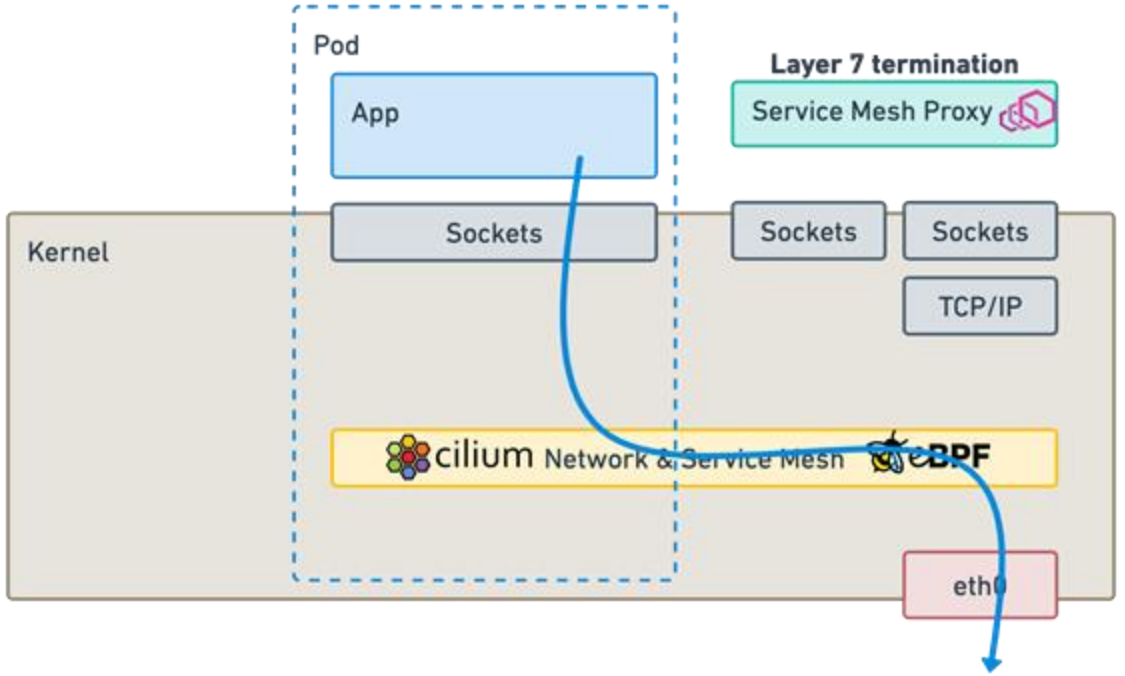


Service Mesh

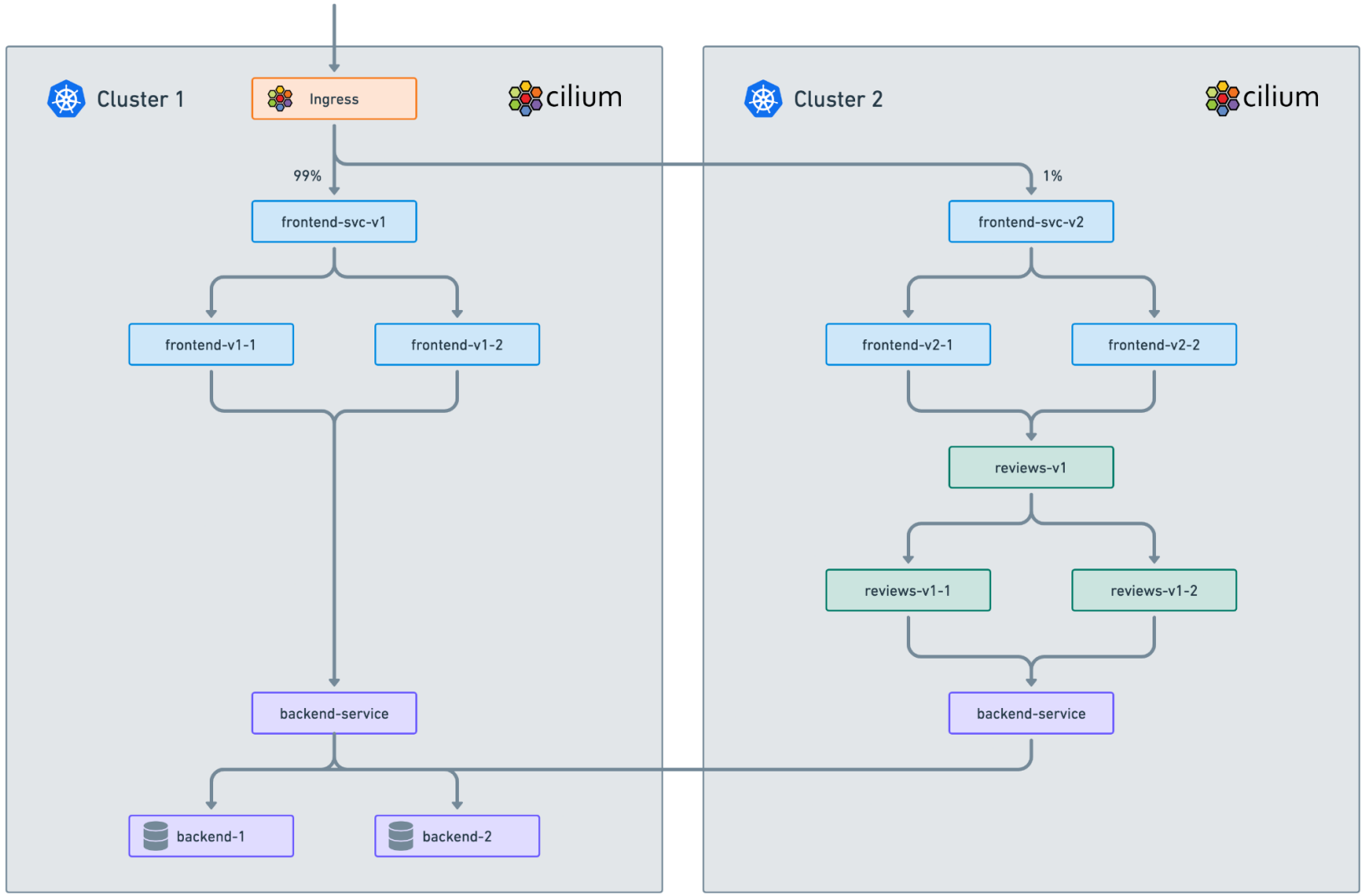
- Reduced operational complexity
- Reduced resource usage and better performance
  - sidecar-free routing (based on the Control plane)
- Flexible and supports everything you need
  - IP, TCP, UDP, HTTP, Kafka, gRPC and DNS
- Decide on your Control plane
  - Ingress, Gateway API, EnvoyConfig, Istio, Spiffe
- Identity-based Security

Cost of sidecar injection





Canary Rollout with Cluster & Service Mesh

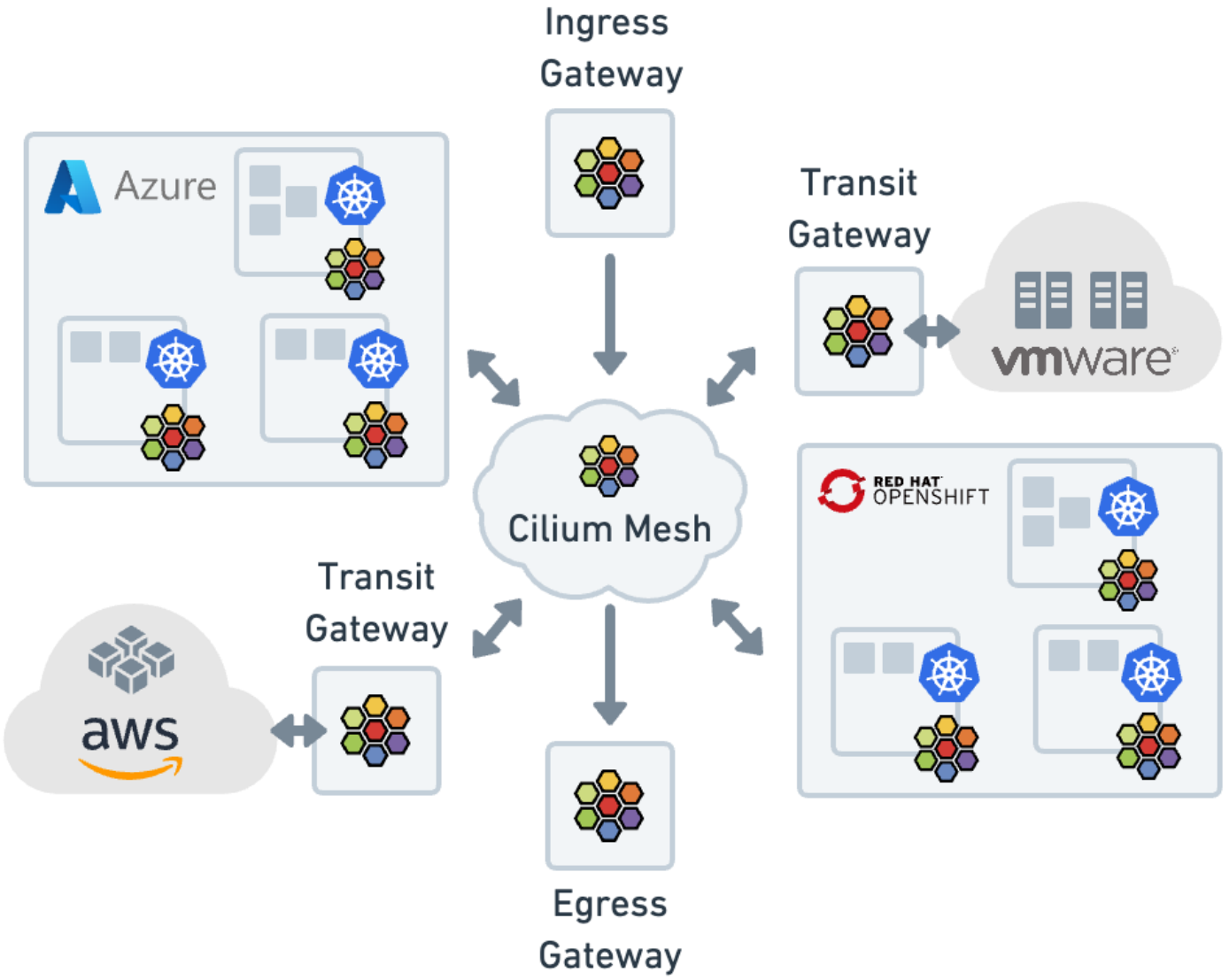


Demo: Canary Rollout with Cilium

- Application v1 running on Cluster 01
- Application v2 running on Cluster 02
  - Frontend and API
- Application Ingress via Cluster 01
  - GatewayAPI and Route
- leverage Cilium Cluster Mesh and Cilium Service Mesh to control traffic distribution of the application

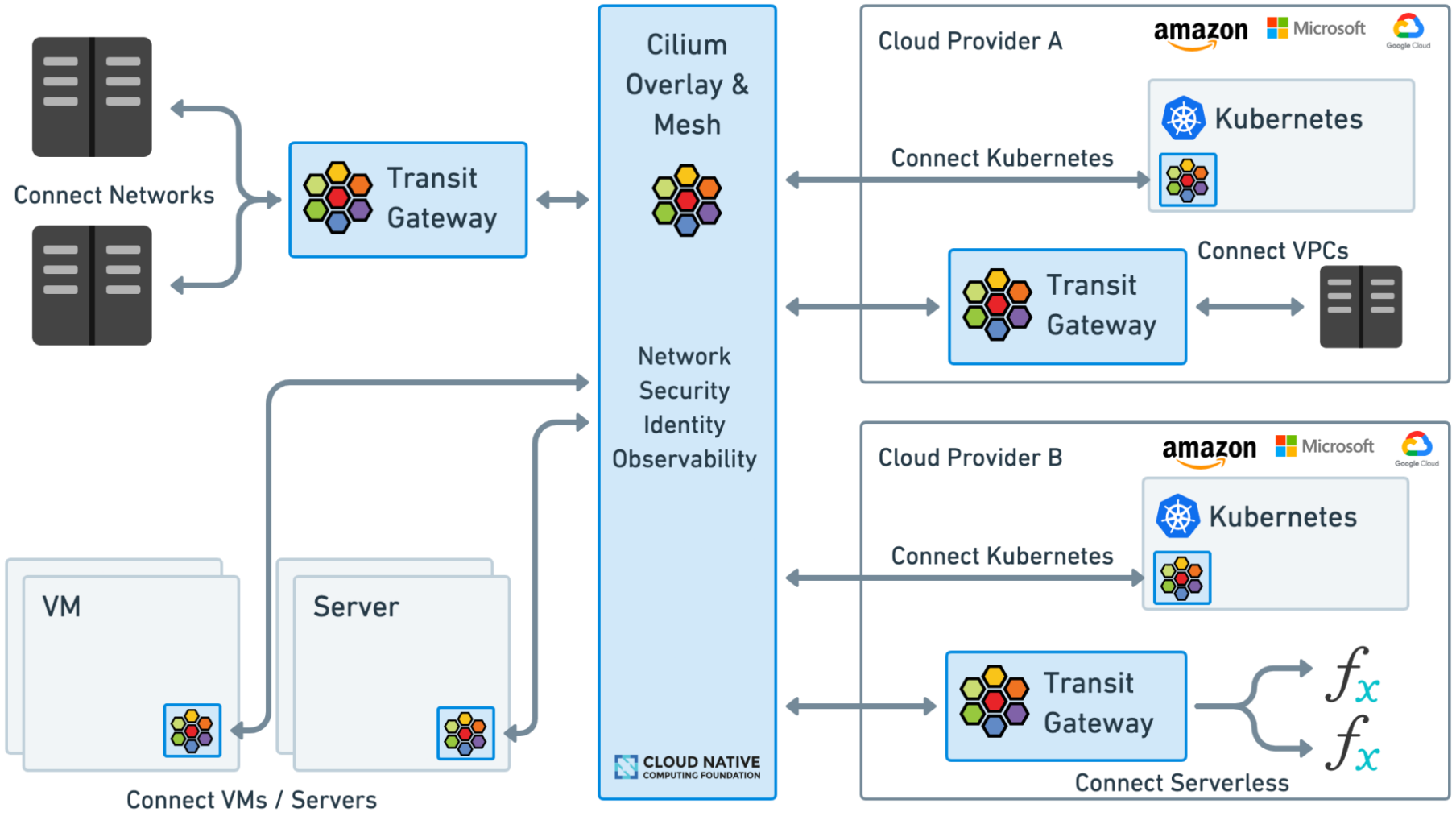
Cilium Mesh – one mesh to connect them all

Cilium Mesh – Big Picture



Transit Gateway requires Isovalent Cilium Enterprise

Connect them all



Links & Getting started

- https://github.com/whiteducksoftware/cilium-next-level-k8s-networking
- https://cilium.io
- https://docs.cilium.io
- https://networkpolicy.io
- https://github.com/cilium/cilium

Thanks to Isovalent for the graphics and bees!

Other sessions

Raymond de Jong
Isovalent

Session Title:
"Cilium & Grafana LGTM!"

Nico Meisenzahl
white duck

Philip Welz
white duck

Session Title:
"How to Prevent Your Kubernetes Cluster From Being Hacked"

Session Title:
"Next-level Kubernetes Networking with Cilium"

New Speakers

www.containerdays.io

Questions?

Philip Welz
(Senior DevOps & Kubernetes Engineer, Azure MVP)

+49 8031 230159-0

philip.welz@whiteduck.de

@philip_welz

www.linkedin.com/in/philip-welz

Nico Meisenzahl
(Head of DevOps Consulting & Operations, Cloud Solution Architect)

+49 8031 230159-0

nico.meisenzahl@whiteduck.de

@nmeisenzahl

www.linkedin.com/in/nicomeisenzahl


white
duck
Thank you!