KubeCon CloudNativeCon India 2026

Zero Trust for Fintech: Building Secure Banking Infrastructure With Cilium

Herbert Sianturi, Krom Bank Indonesia
Prasta Maha, Krom Bank Indonesia

KubeCon CloudNativeCon India 2026

Herbert Sianturi

Senior DevOps Engineer
Krom Bank Indonesia

Herbert Sianturi

Prasta Maha

Lead DevOps Engineer
Krom Bank Indonesia

Prasta Maha

Topics

- Zero Trust Principal
- Banking Use Case
- Cilium Overview
- How Cilium Implement Zero Trust
- Key Takeaways

Zero Trust Principal

> ***"Never Trust, Always Verify."***
>
> *— NIST SP 800-207, Zero Trust Architecture (2020)*

No Implicit Trust

Network location does not grant access.

Always Authenticate

Every request, every session, every time.

Continuous Verification

Trust is re-evaluated, not granted once.

Banking Use Case

Three forces pushing every bank toward Zero Trust

Regulation

- PCI-DSS v4.0 — segmentation, encryption, logging
- Local rules — POJK, RBI, MAS, PSD2
- ISO 27001 and SOC 2 access-control expectations

Threat Model

- Lateral movement after a compromised pod
- Supply-chain risk in container images and CI
- Insider threat from privileged operators

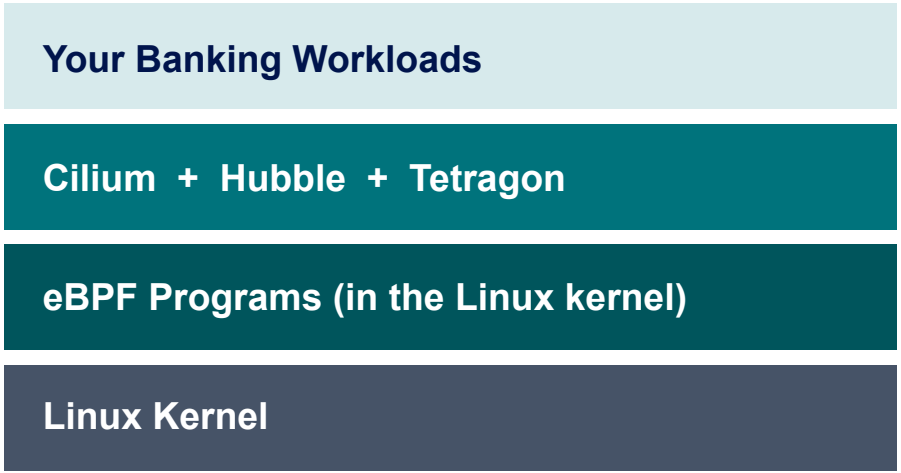
Cloud-Native Reality

- Pods rescheduled across nodes every minute
- Hundreds of microservices, multi-tenant clusters
- Static IP allow-lists do not scale or stay correct

Cilium Overview

KubeCon CloudNativeCon India 2026

Networking, security, and observability for Kubernetes built on eBPF



- **CNCF** — Graduated project (2023)
- **L3 → L7** — Identity, IP, port, HTTP, Kafka, gRPC, DNS
- **Kernel** — Enforcement, not sidecar proxies

How Cilium Implement Zero Trust

KubeCon CloudNativeCon India 2026

Identity
who am I?

K8s resources: Pod, Node, Cluster, Kube-apiserver

Network Policy
What am allowed to access?

Firewall on identity, FQDN, DNS, and HTTP / L7

Encryption
Who can read me?

WireGuard / IPsec + identity-based mTLS

Observability
What just happened?

Hubble — every flow, every drop, every L7 verb

Runtime Security
What's actually running inside the pod?

Tetragon — process, file and syscall events

Cilium Resource as Identity

KubeCon CloudNativeCon India 2026

Cilium translates Kubernetes objects into a numeric security identity — policy follows the label, not the IP.

Pod

Every pod derives identity from labels (app, tier, env). Reschedules do not change it.

```
id:1234 -> app=payment,env=production
```

Node

Node has its own identity. Used for host firewall policy on the kubelet.

```
reserved:host
reserved:remote-node
```

Cluster

In Cluster Mesh, each cluster carries an ID. Cross-cluster policy, no leaked topology.

```
cluster: jakarta  ↔  cluster: mumbai
```

kube-apiserver

A reserved identity for traffic to and from the Kubernetes API server itself.

```
reserved:kube-apiserver
```

Cilium Firewall / Network Policy

KubeCon CloudNativeCon India 2026

One policy language all in CiliumNetworkPolicy from L3, L4 and L7. By default, block all, and allow defined rules.

Identity (L3 / L4)

Control which identity a pod can reach use toEndpoints / fromEndpoints

```
toEndpoints:
- matchLabels:
    k8s:io.kubernetes.pod.namespace: payment
    app: payment
toPorts:
- ports: [{ port: "8080" }]
```

FQDN (egress allow-list)

Pods may only reach approved external domains

```
toFQDNs:
- matchName: "api.swift.com"
- matchPattern: "*.cncf.com"
toPorts:
- ports: [{ port: "443" }]
```

CIDR

Control which CIDR a pod can reach

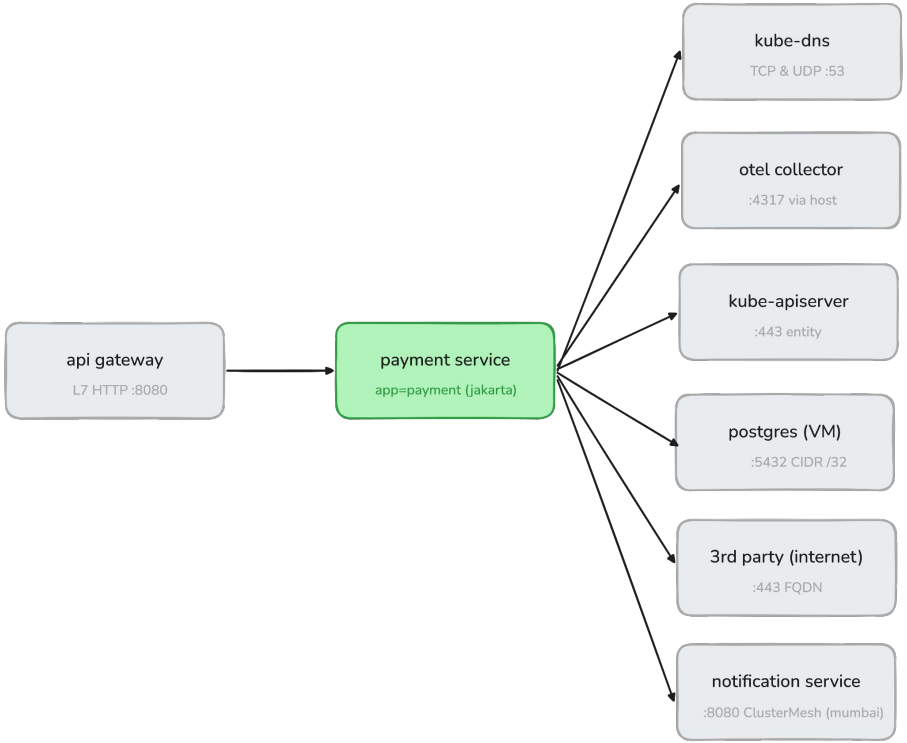
```
toCIDRSet:
- cidr:172.21.89.10/32
toPorts:
- ports: [{ port: "5432" }]
```

HTTP / L7 (verb · path · header)

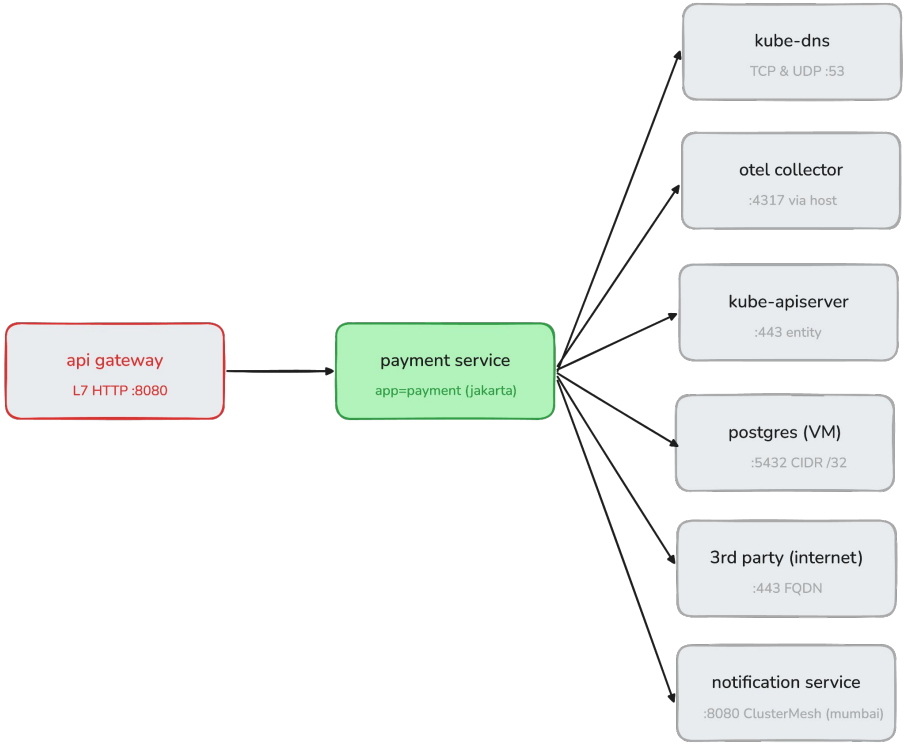
Only GET /balance and POST /transfer

```
toPorts:
- ports: [{ port: "8080" }]
  rules:
    http:
    - { method: "GET",  path: "/balance"  }
    - { method: "POST", path: "/transfer" }
```

Cilium Firewall / Network Policy

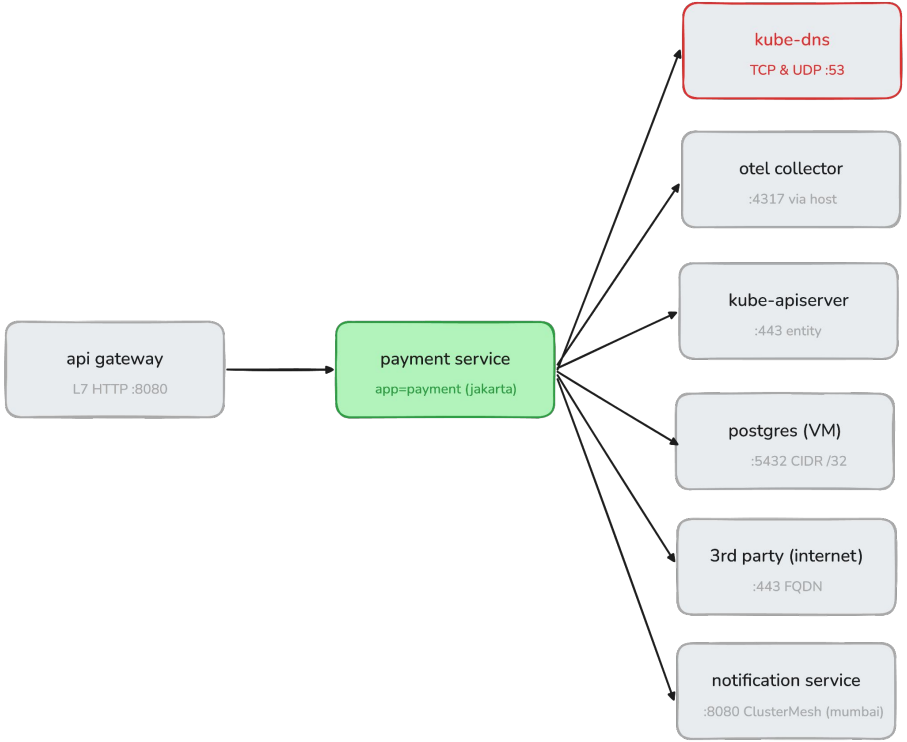


Cilium Firewall / Network Policy



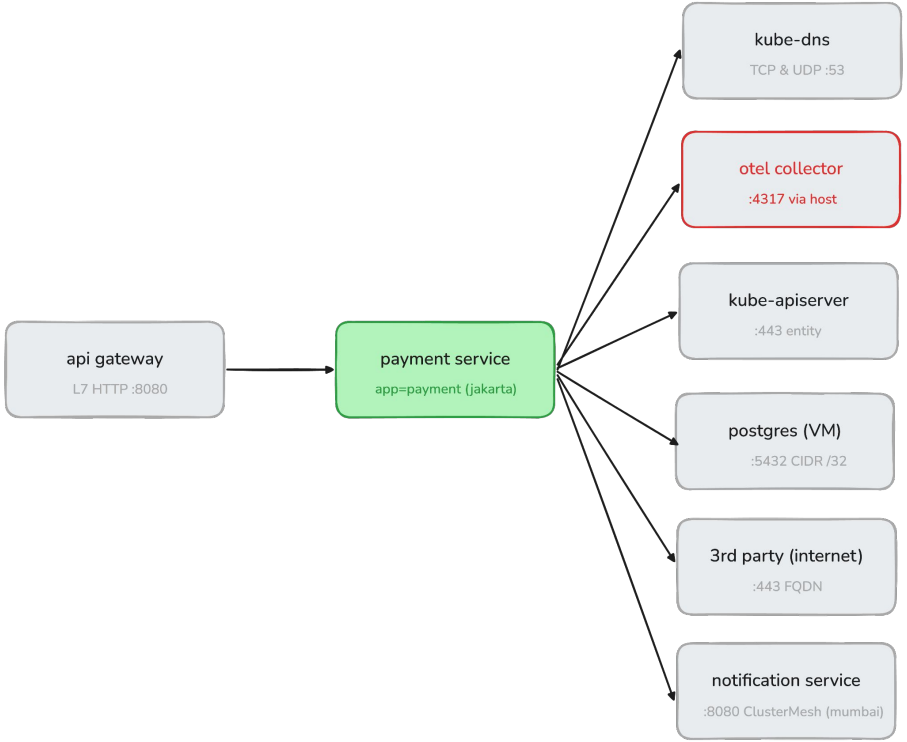

```
....
ingress:
  # API gateway -> payment, L7 HTTP only
  - fromEndpoints:
      - matchLabels:
          io.kubernetes.pod.namespace:
ingress-gateway
          app: api-gateway
    toPorts:
      - ports:
          - port: "8080"
            protocol: TCP
        rules:
          http:
            - method: "GET"
              path: "/balance"
            - method: "POST"
              path: "/transfer"
....
```

Cilium Firewall / Network Policy



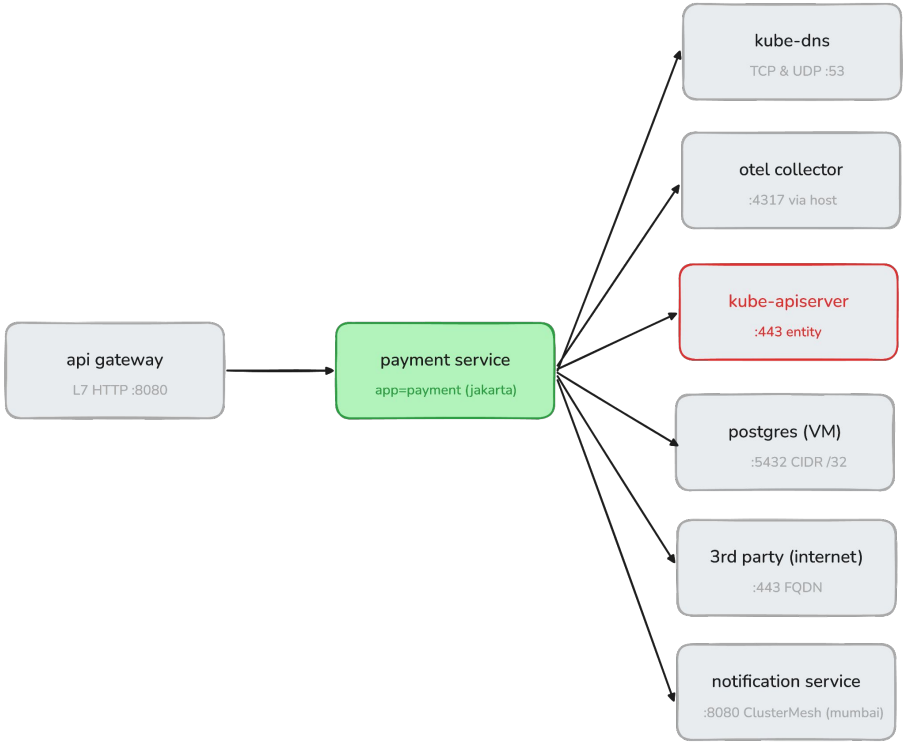

```
....
egress:
  # DNS - REQUIRED, otherwise toFQDNs below can't
resolve
  - toEndpoints:
      - matchLabels:
          io.kubernetes.pod.namespace: kube-system
          k8s-app: kube-dns
    toPorts:
      - ports:
          - port: "53"
            protocol: ANY
        rules:
          dns:
            - matchPattern: "*"
....
```

Cilium Firewall / Network Policy




```
....
egress:
....
  # Node-local OpenTelemetry
collector (DaemonSet)
  - toEntities:
      - host
    toPorts:
      - ports:
          - port: "4317"
            protocol: TCP
....
```

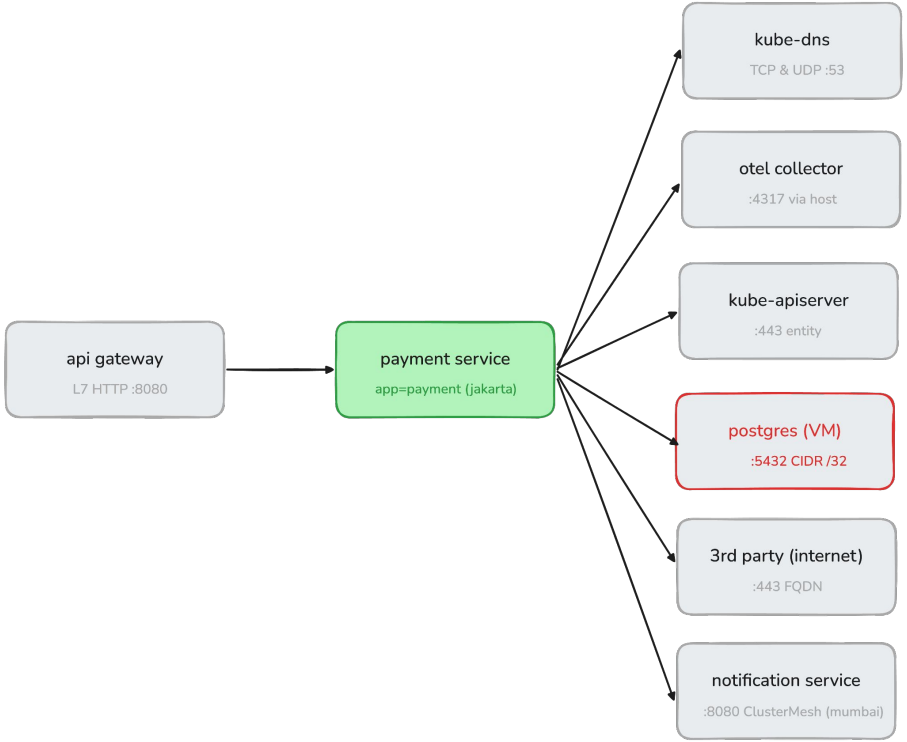
Cilium Firewall / Network Policy




```
....
  # Kubernetes API server
  - toEntities:
      - kube-apiserver
    toPorts:
      - ports:
          - port: "443"
            protocol: TCP
....
```

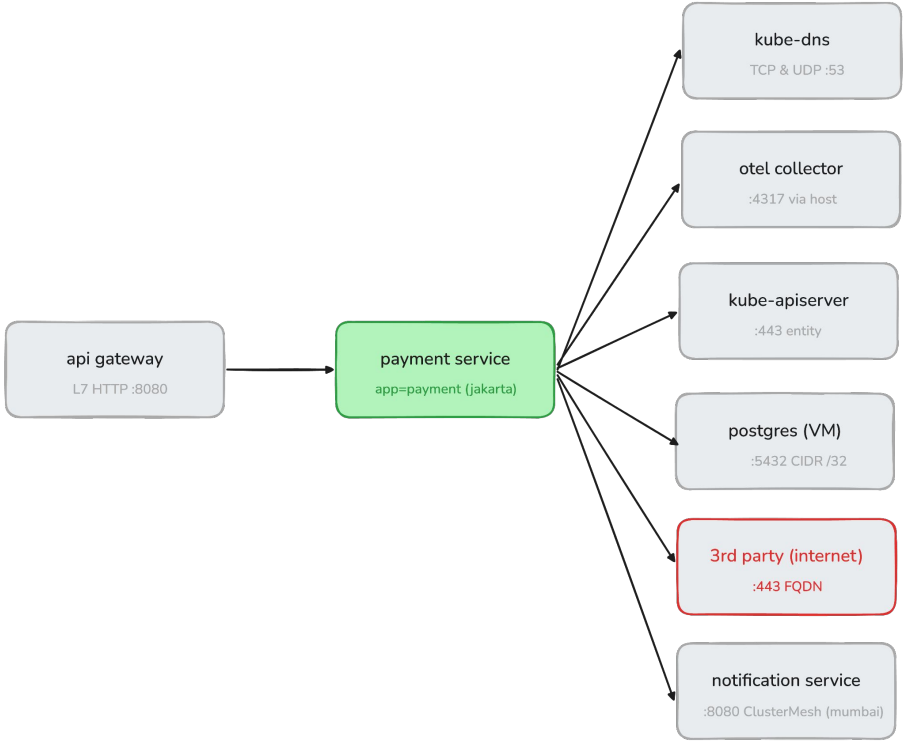
Cilium Firewall / Network Policy

KubeCon CloudNativeCon India 2026



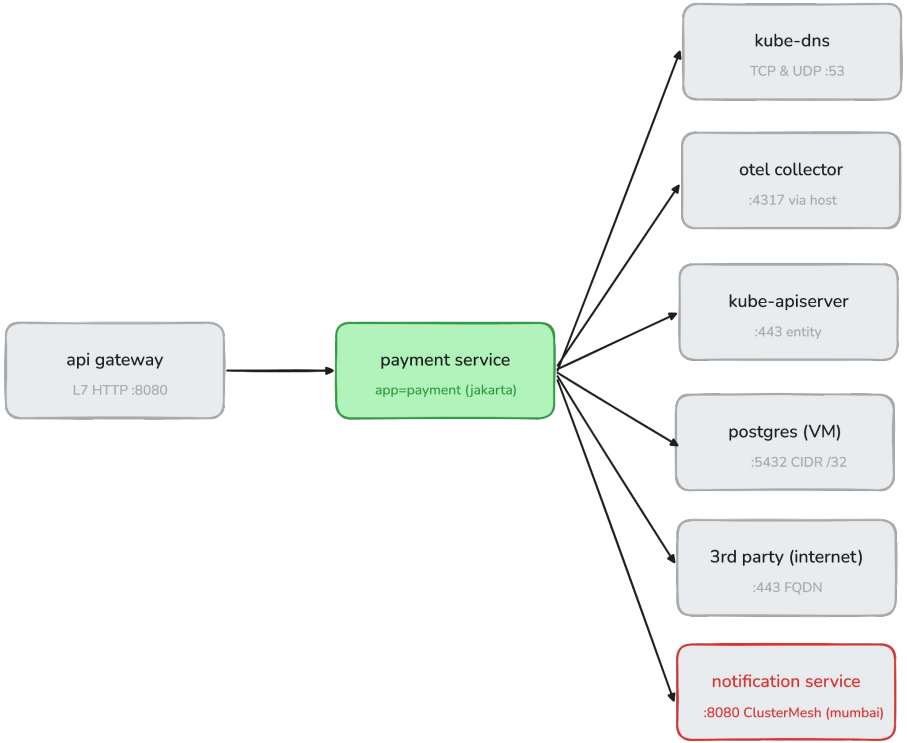

```
....
  # CIDR postgres (VM / managed service)
  - toCIDRSet:
      - cidr: 10.123.123.10/32
      - cidr: 10.123.123.20/32
    toPorts:
      - ports:
          - port: "5432"
            protocol: TCP
....
```

Cilium Firewall / Network Policy




```
....
  # FQDN 3rd party via internet
  - toFQDNs:
      - matchName: "api.abc123.com"
      - matchPattern: "*.cncf.io"
    toPorts:
      - ports:
          - port: "443"
            protocol: TCP
....
```

Cilium Firewall / Network Policy

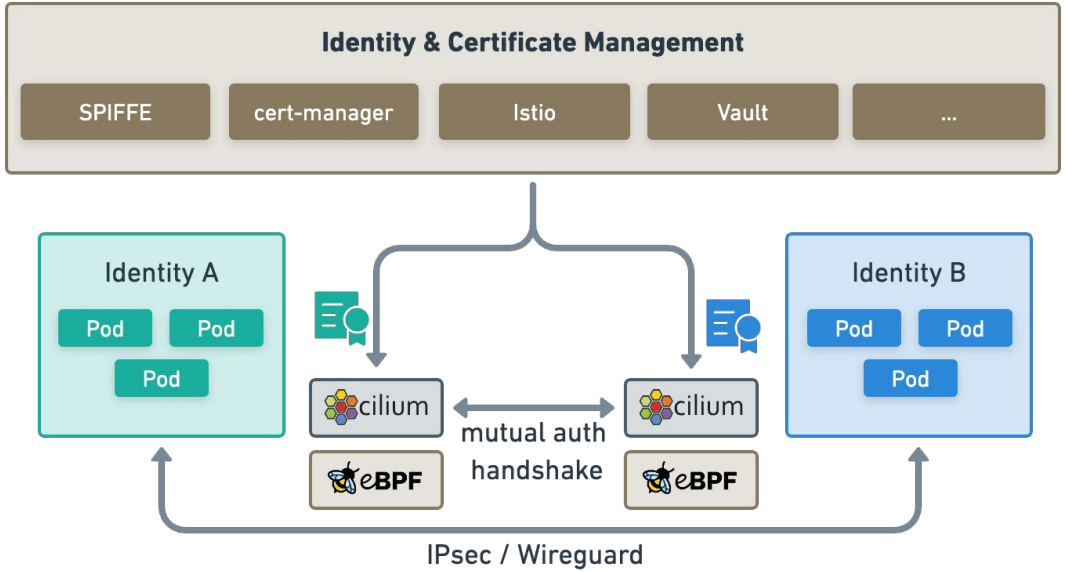



```
....
egress:
....
  # Cross-cluster via ClusterMesh
  - toEndpoints:
      - matchLabels:
          io.cilium.k8s.policy.cluster: mumbai
          io.kubernetes.pod.namespace: notif
          app: notif
    toPorts:
      - ports:
          - port: "8080"
            protocol: TCP
....
```

Cilium Encryption

KubeCon CloudNativeCon India 2026

Encrypt the wire AND the workload. Transparent transport + identity-based mTLS.




```yaml
ingress:
  - fromEndpoints:
      - matchLabels:
          App: service-a
    authentication:
      mode: "required"
    toPorts:
      - ports:
          - port: "8080"
            protocol: TCP
```

Use both: WireGuard / IPsec for the wire, mTLS for workload identity proof.

Cilium Audit & Observability

KubeCon CloudNativeCon India 2026

Hubble shows every flow, every drop, every L7 verb. Captured from eBPF, in the kernel.

```
$ hubble observe --namespace banking --verdict DROPPED

Jun 14 09:12:03  DROP  pay-svc → ledger-db:5432  TCP  policy-denied
Jun 14 09:12:04  DROP  web-fe → 198.51.100.7:80  TCP  policy-denied
Jun 14 09:12:06  DROP  fraud-svc → ledger-db:5432  TCP  no L7 match
Jun 14 09:12:08  DROP  dev-pod → ledger-db:5432  TCP  identity not
allowed
```

What the audit team gets

Forensic replay

Every dropped packet, source, destination, and reason — for any incident window.

Continuous evidence

Export to Prometheus, OpenTelemetry, or SIEM. Same data for SRE and audit.

Live service map

Hubble UI renders the live graph — invaluable for scope-of-PCI reviews.

Cilium Runtime Security

KubeCon CloudNativeCon India 2026

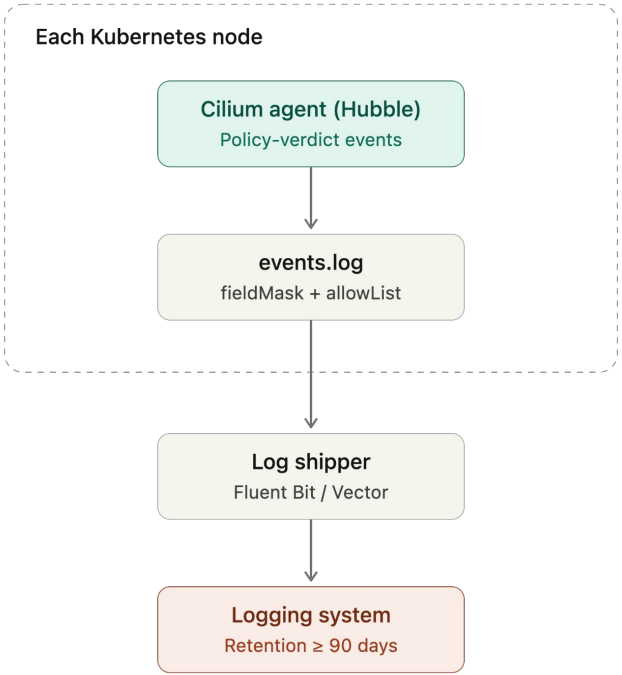
*Network policy stops bad packets. **Tetragon** stops bad behavior inside the pod itself.*

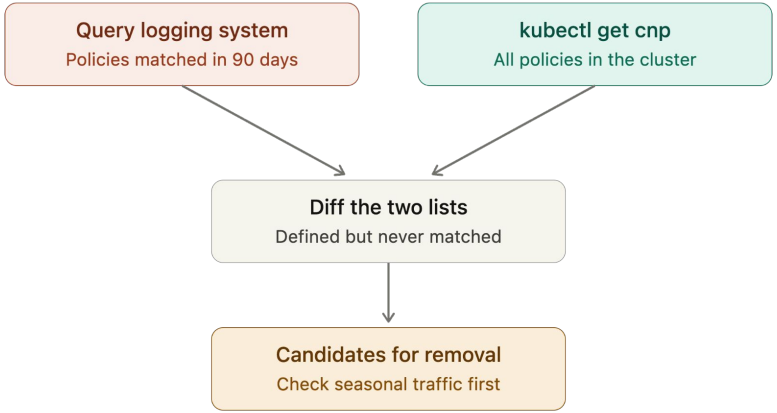
Observe and enforce, in the kernel

- **Process execution**
  Detect every exec — shell, curl, kubectl in prod pods.
- **File integrity**
  Alert on read or write of secrets paths and keystores.
- **Syscall tracing**
  Capture ptrace, raw sockets, capability use.
- **Process-aware network**
  Attribute every flow to a specific binary and PID.
- **Inline enforcement**
  SIGKILL or override syscalls when a policy matches.

```
$ tetra getevents -o compact

[exec]  payment-7d9  bash -c "..."
  parent: kubectl exec  user: root

[file]  ledger-pg  open() /etc/ssl/pg.key
  binary: cat   policy: secret-files   BLOCK

[net]   web-3kf2  connect 185.220.101.42:443
  binary: /tmp/.x   policy: tor-exit   SIGKILL

[cap]   dev-pod  ptrace(PEEKTEXT) pid=1421
  binary: gdb   policy: ptrace-deny
```

Evaluate Policy

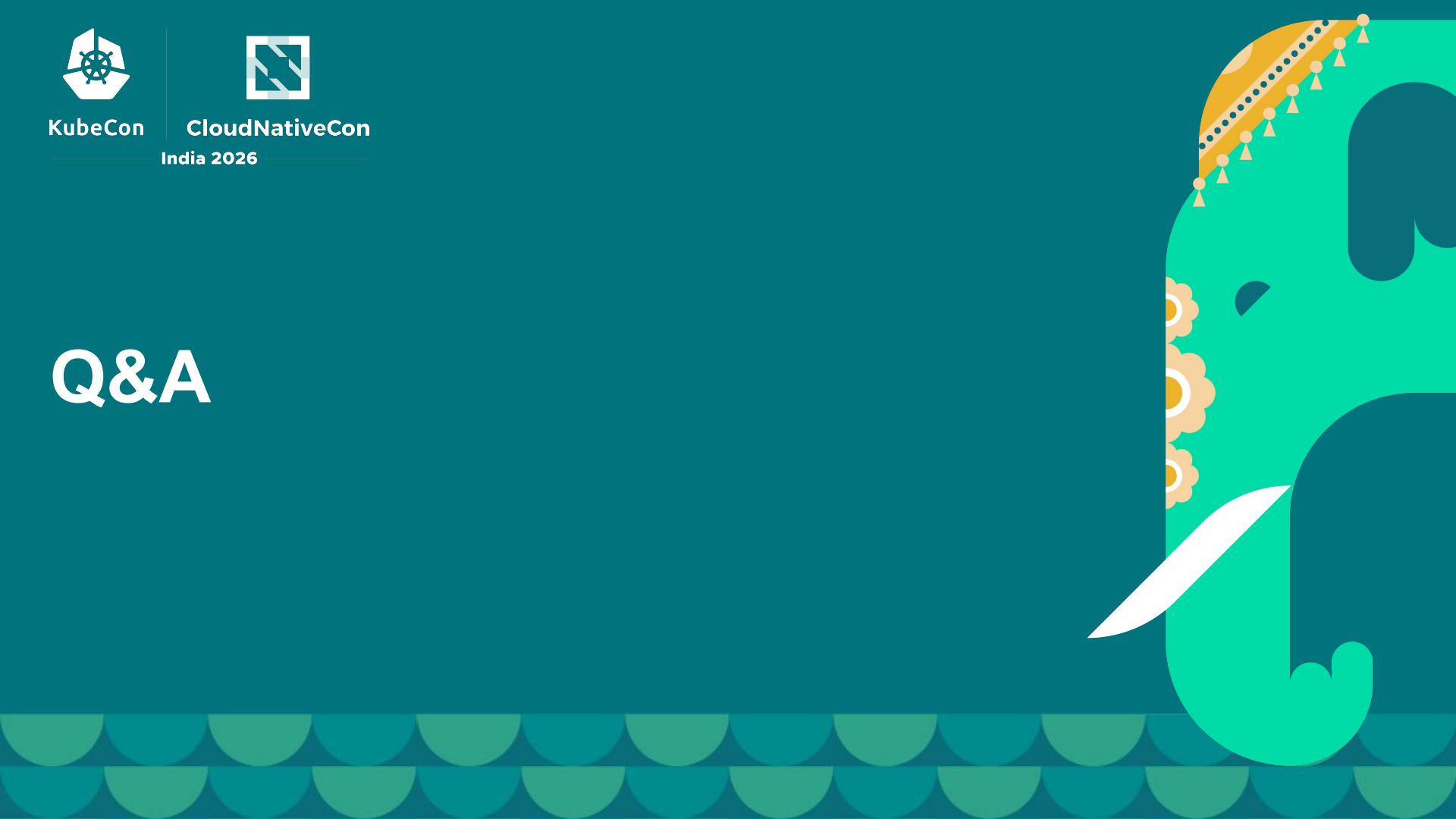




Key Takeaways

Zero Trust is a practice, not a product. Cilium gives you the pieces.

1. **Identity, not IP**
   K8s labels become identity. Policy travels with the workload.
2. **Policy at every layer**
   Identity, FQDN, DNS, and L7 one CRD, one control plane.
3. **Encrypt wire AND workload**
   WireGuard / IPsec for transport; mTLS for mutual identity proof.
4. **Observability is evidence**
   Hubble flow logs serve both the SRE and the auditor.
5. **Runtime closes the loop**
   Tetragon catches what bypasses the network in real time.


KubeCon
CloudNativeCon
India 2026
Q&A


KubeCon
CloudNativeCon
India 2026
Thank You
धन्यवाद