


KubeCon | CloudNativeCon

North America 2023

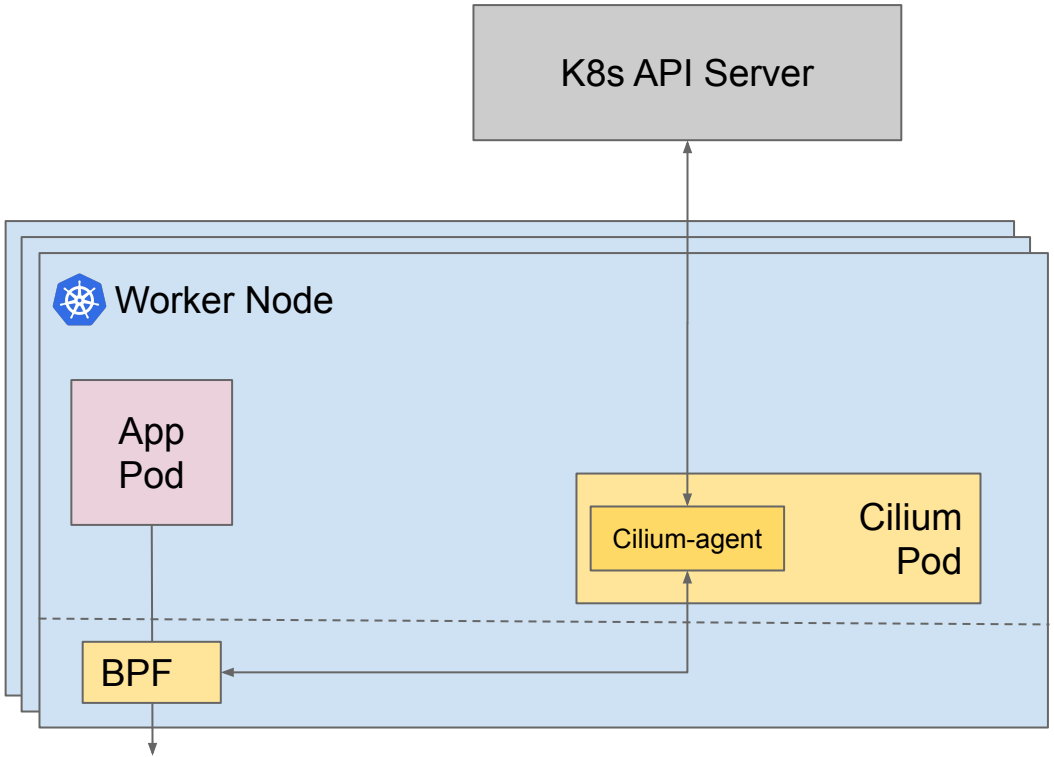
Cilium Overview for Developers

Bill Mulligan, Joe Stringer, Jef Spaleta
Isovalent

Agenda

- Architecture
- Core Concepts
- Structure / Internals
  - Programming the kernel (Datapath)
  - Observing the network (Hubble)
- Practical steps to get started

Architecture - High-level View



Declarative Intent:

cilium-agent watches K8s API objects:

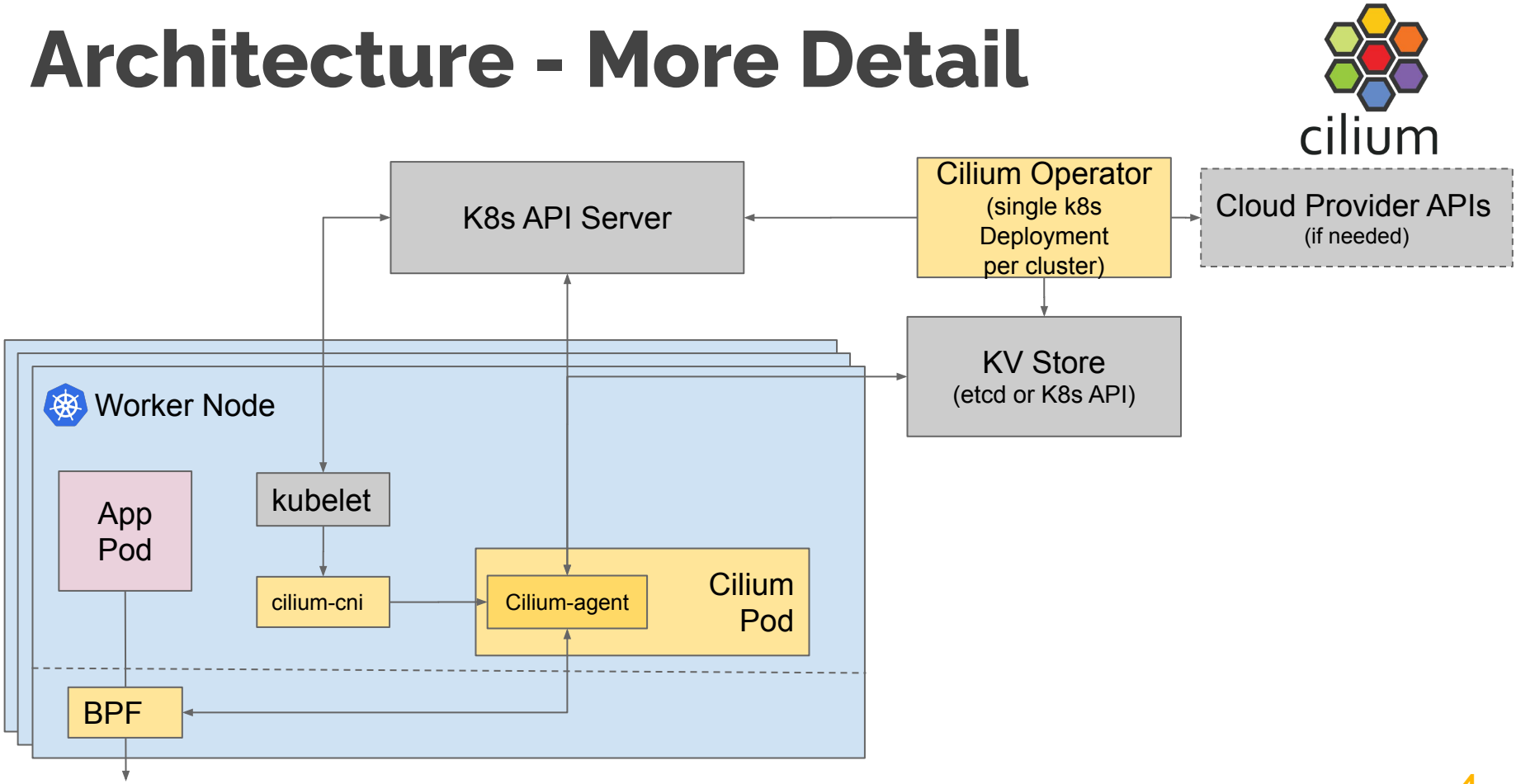
- K8s Nodes
- K8s Pods
- Network Policies
- Services + Endpoints

Runtime Network & Security State:

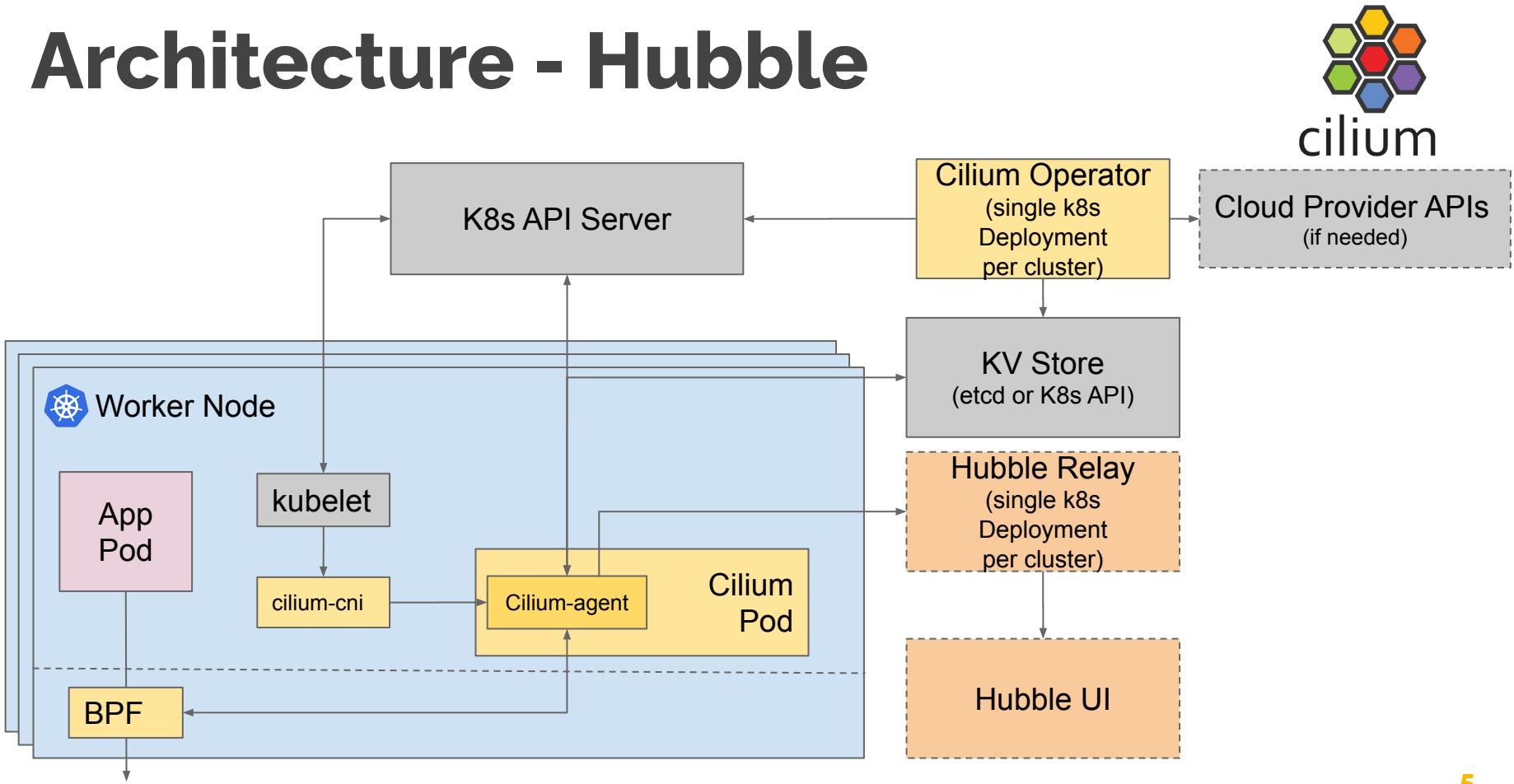
cilium-agent on each node generates eBPF programs based on the label identity of each pod that provide:

- Pod-to-pod connectivity (IPAM, overlay / direct routing)
- Pod to/from external connectivity (NodePort, Masq, …)
- Service-based Load Balancing
- Identity-aware Network policy filtering (Label, DNS).
- Identity-aware network flow visibility & metrics.
- Transparent Encryption (IPsec, WireGuard).
- Multi-cluster Routing & Security

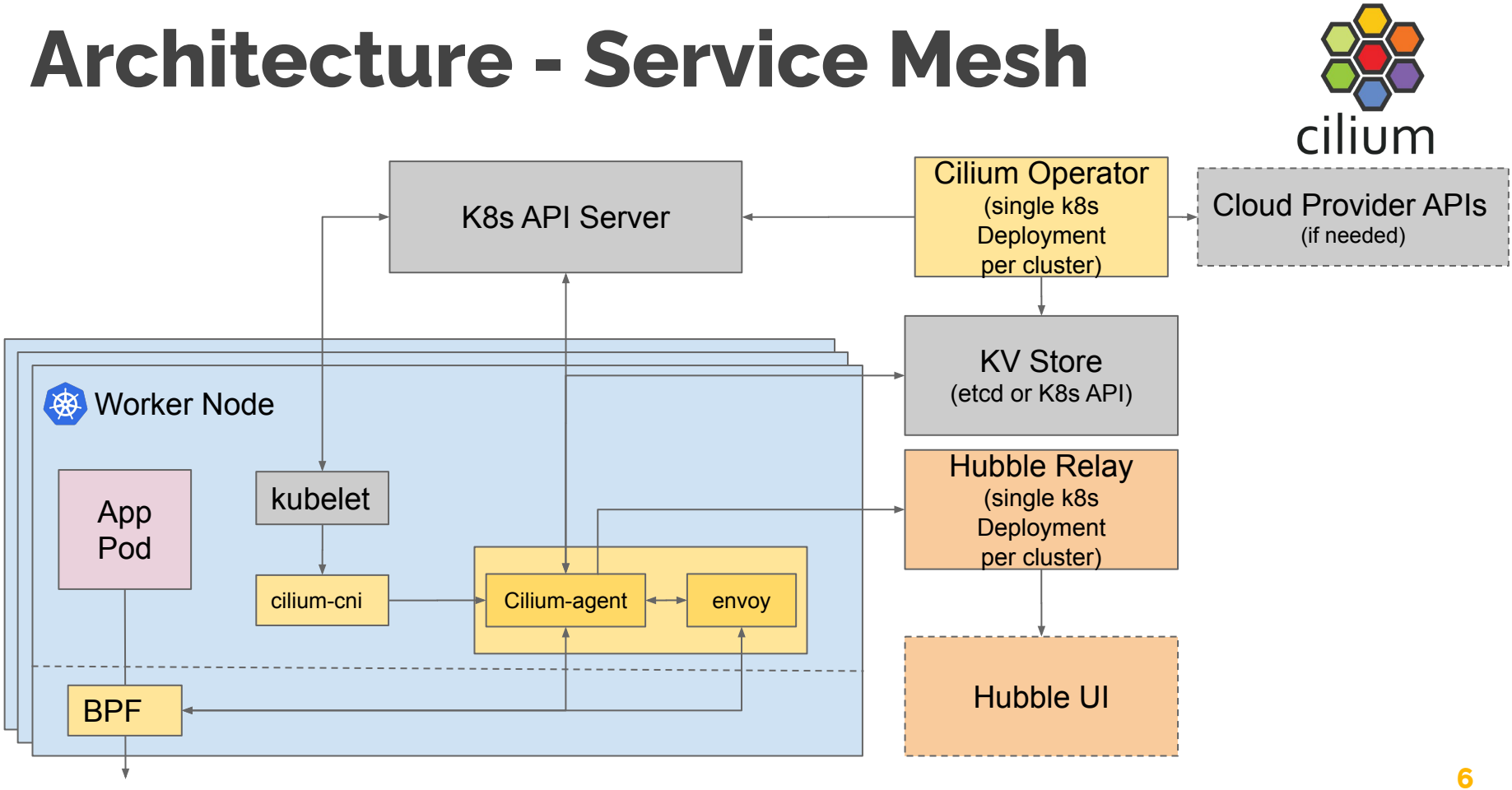
Architecture - More Detail



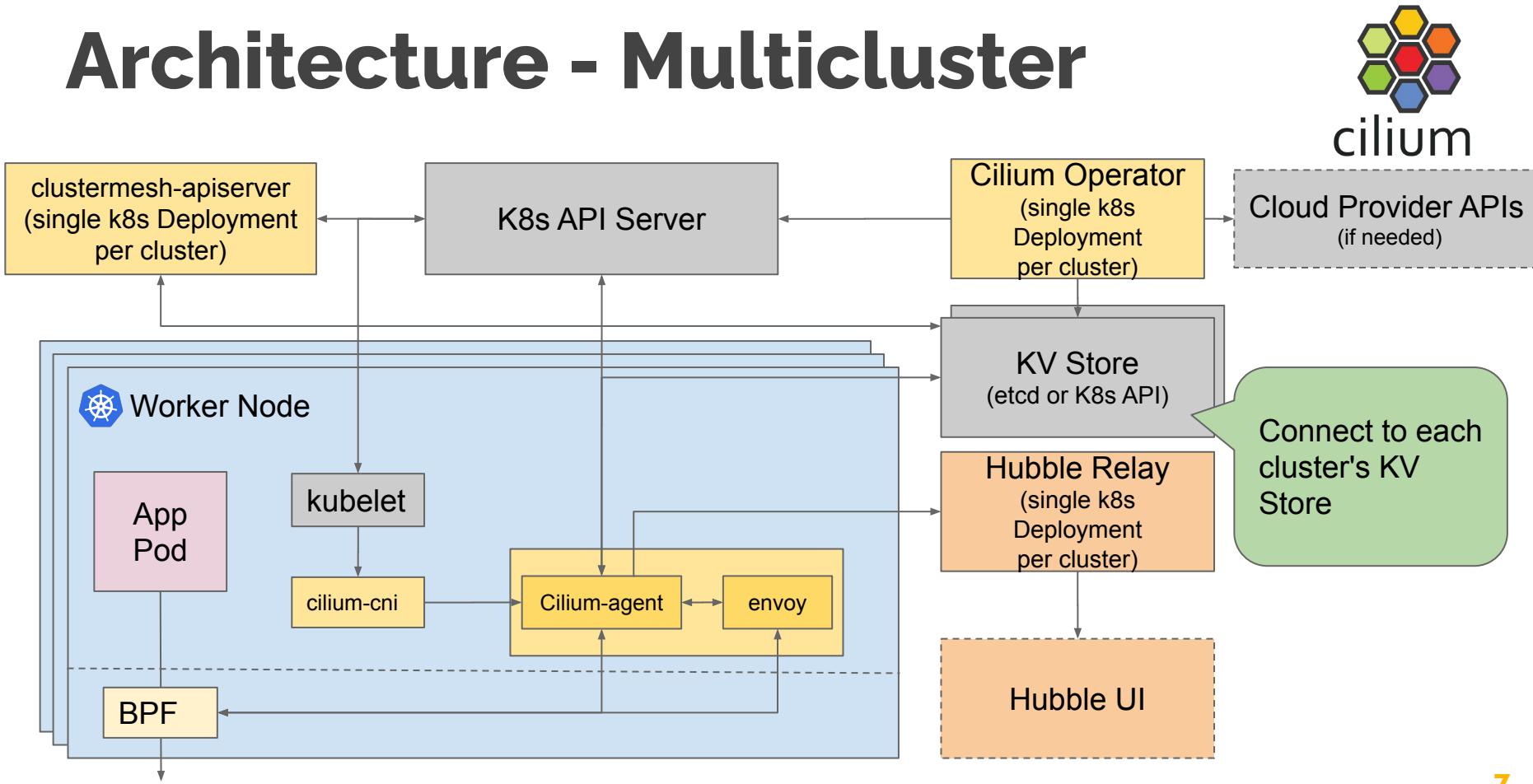
Architecture - Hubble



Architecture - Service Mesh



Architecture - Multicluster

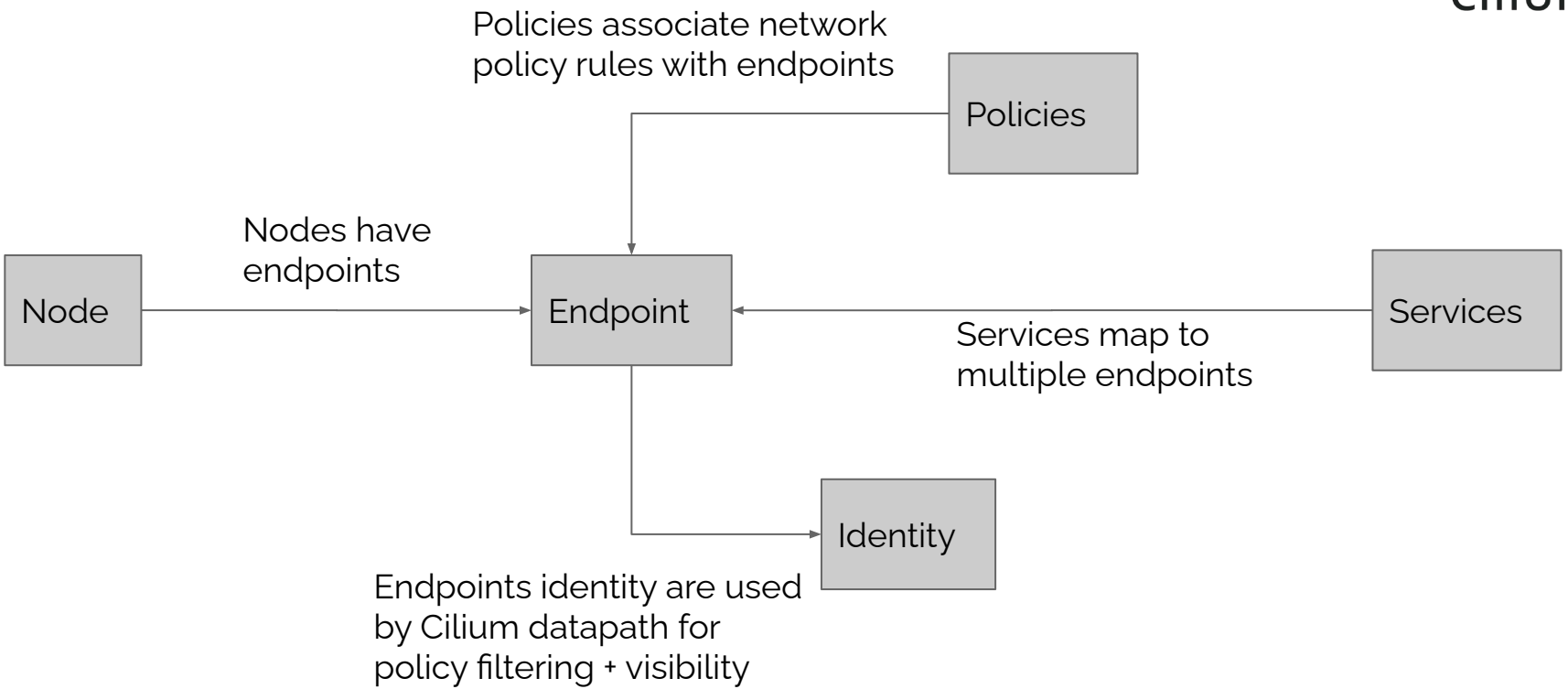


Core Resources

Core Cilium Abstractions

Cilium Concept	K8s Resource	Source of Truth	Comments
Node	Node / CiliumNode	Kubernetes	Represents a Kubernetes worker node
Endpoint	Pod / CiliumEndpoint (CEP) / CiliumEndpointSlice (CES) /	Node	Each Cilium-managed Pod ⇒ endpoint
Security Identity	CiliumIdentity	Identity Store (K8s / kvstore)	Each unique set of pod labels ⇒ identity.
Policy	NetworkPolicy / CiliumNetworkPolicy / CiliumClusterwideNetworkPolicy	Kubernetes	Describes ingress / egress connectivity of an endpoint
Service	Service	Kubernetes	L3/L4 load-balancing

Cilium Abstractions



Resources tooling

- Nodes
  - cilium-dbg node list
  - kubectl get ciliumnodes
- Cilium Endpoints
  - cilium-dbg endpoint list
  - kubectl get ciliumendpoints --all-namespaces
- Network Policies
  - cilium-dbg policy get
  - kubectl get [netpol|cnp|ccnp] --all-namespaces
- Security identities
  - cilium-dbg identity list
  - kubectl get ciliumidentities
- Services
  - cilium-dbg service list
  - kubectl get svc --all-namespaces

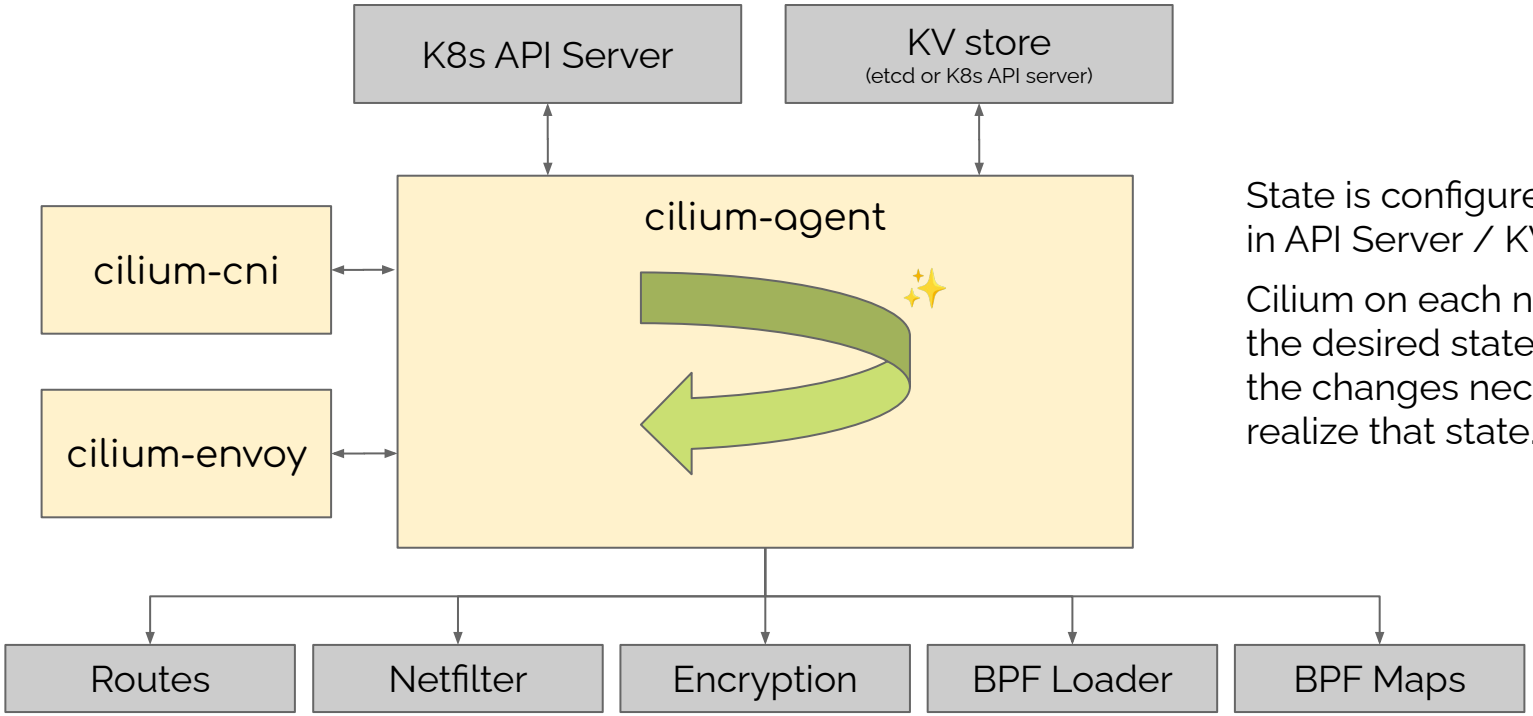
Before Cilium v1.15.0-pre.2, `cilium-dbg` was named `cilium` inside the Cilium container

Additional Custom Resources

Feature	What does it do?
CiliumExternalWorkload	Connect external nodes to the Cilium cluster
CiliumLocalRedirectPolicy	Help with local-node redirects. Example: DNS
CiliumEgressGatewayPolicy	Pods can connect outside the cluster with consistent IPs
CiliumEnvoyConfig / CiliumClusterwideEnvoyConfig	Apply more detailed Envoy configurations
CiliumBGPPeeringPolicy	Configure the way that Cilium connects to BGP peers
CiliumLoadBalancerIPPool	Assign IP addresses to LoadBalancer services
CiliumNodeConfig	Per-node Cilium configuration rather than cluster-wide
CiliumCIDRGroup	Associate a set of IPs with a name for use in policy
CiliumL2AnnouncementPolicy	Advertise service IPs onto the local area network
CiliumPodIPPool	Provide greater control over IP address management for Pods

Internal Architecture

cilium-agent **Operations**



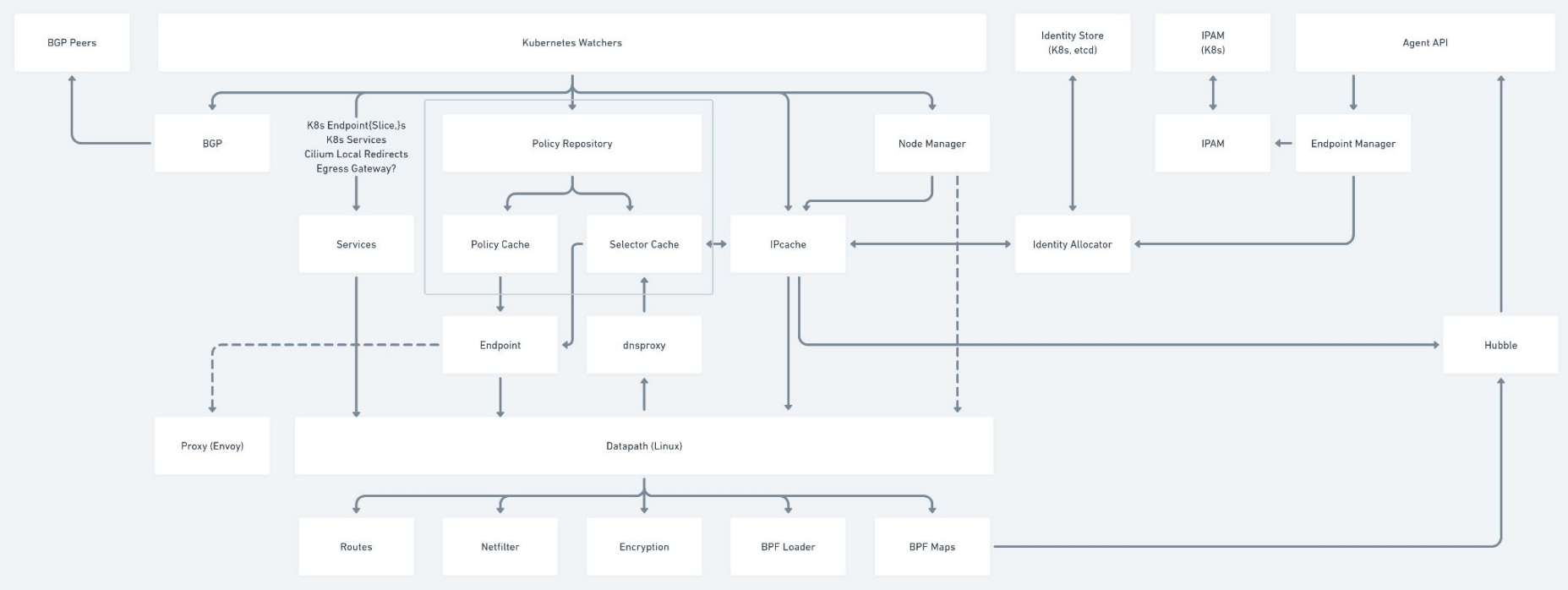
State is configured and shared in API Server / KV store.

Cilium on each node observes the desired state and computes the changes necessary to realize that state.

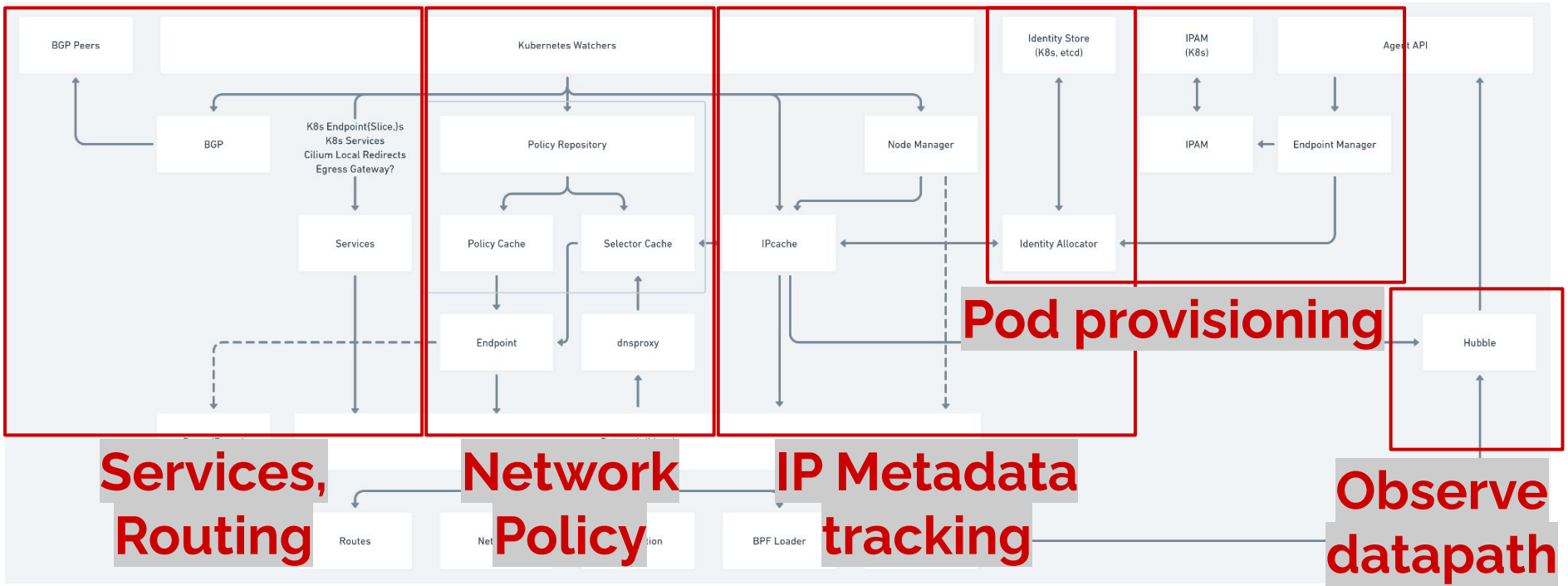
Hive / Cell

- (Bee) Hive is where feature dependencies are registered
- Cells are structures within the Hive that are responsible for a specific feature.
  - Home for the flags related to the feature
  - Metrics for the feature
  - Start / Stop hooks
- Guide to the Hive goes into more details

Cilium Internal Architecture

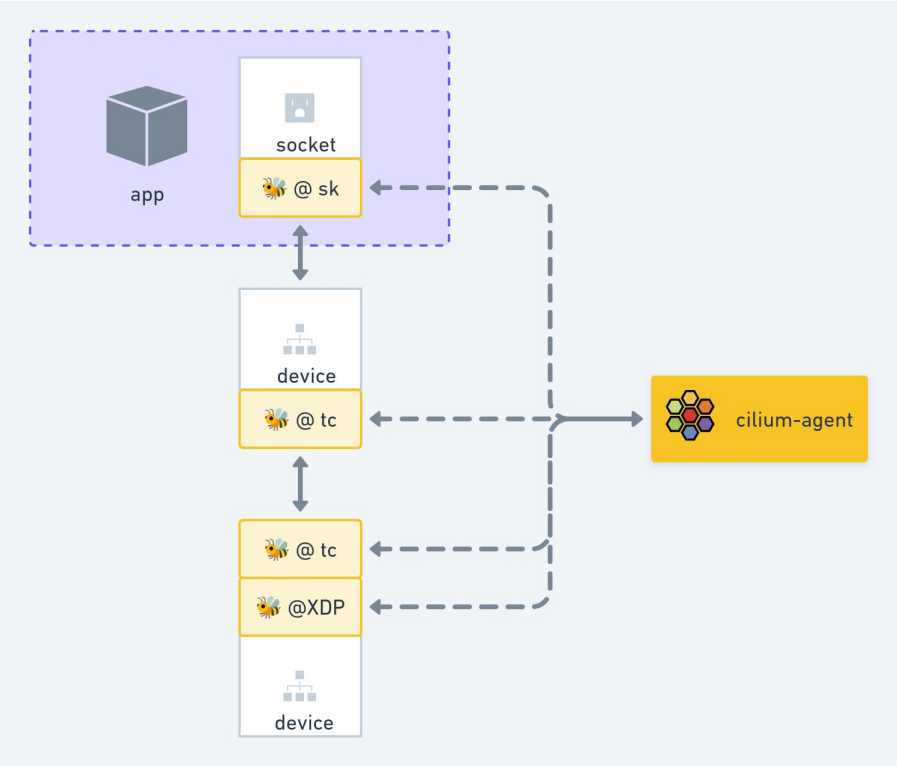


Cilium Internal Architecture



Datapath / eBPF

How Cilium Uses eBPF



Cilium generates and attaches eBPF programs + maps:

- At socket for east<->west load-balancing
- At traffic control layer for intra-node connectivity & packet enforcement
- At XDP for efficient north<->south load-balancing

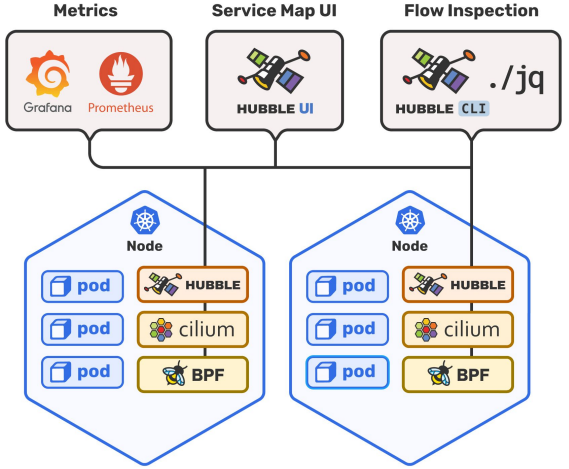
Datapath tooling

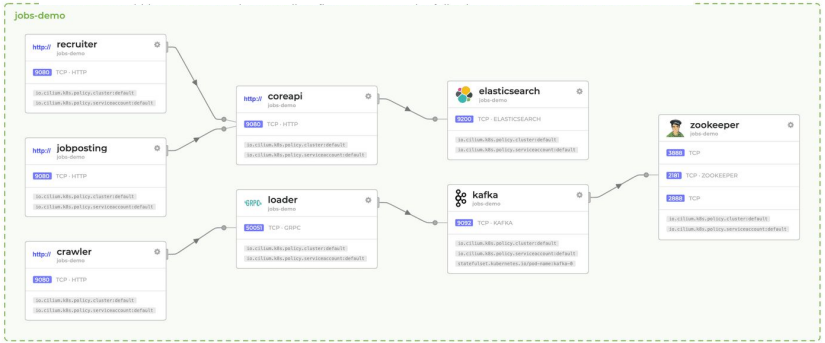
- **eBPF** and **XDP (eXpress Data Plane)**
  - Dynamic features controlled by map state - `cilium bpf *`, `bpftool map show`
  - Features are pre-compiled into programs - `bpftool {net,prog} show`
- Routing
  - policy-based routing - `ip rule`
  - routing tables - `ip route`
- Traffic Control
  - tc qdisc - `tc qdisc show`
  - tc filter - `tc filter show dev *`
- Encryption
  - WireGuard - `cilium bpf ipcache list`
  - ipsec policy - `ip xfrm policy`
- Netfilter
  - iptables - `iptables-save -c`

Hubble

Hubble Visibility

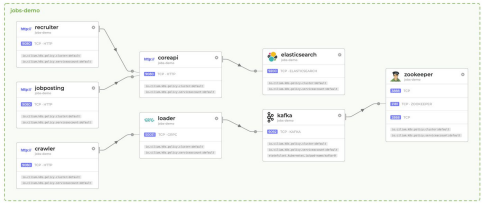
- Caches data on-node for near-term troubleshooting history
- Enables central querying of current + recent flow data across many nodes
  - Hubble API
  - Hubble CLI
  - Hubble UI
- Efficient aggregation/filtering/export of flow data to external systems:
  - Prometheus metrics on pod connectivity data.
  - Log collection / Security Information Event Management (SIEM)





cilium

Hubble Architecture



Hubble UI

Cilium-agent

Identity Metadata Cache

cilium monitor

Hubble server

Event Cache (ring buffer)

gRPC API

Observer

Metrics export

Monitor-reader

Hubble-relay

Hubble CLI

Prometheus

Cilium eBPF Datapath

Perf Ring Buffer

Hubble Basic Usage:

Basic query of Hubble API:

kubectl exec -n kube-system <cilium-pod> -- hubble observe --since 5m -n <namespace>

```
May  3 17:42:32.521   192.168.24.253:36546                          guestbook/frontend-59f5db8cfb-hglxc:80(http)       to-endpoint   FORWARDED   TCP Flags: SYN
May  3 17:42:32.521   guestbook/frontend-59f5db8cfb-hglxc:80(http)  192.168.24.253:36546                               to-stack      FORWARDED   TCP Flags: SYN, ACK
May  3 17:42:32.522   192.168.24.253:36546                          guestbook/frontend-59f5db8cfb-hglxc:80(http)       to-endpoint   FORWARDED   TCP Flags: ACK
May  3 17:42:32.522   guestbook/frontend-59f5db8cfb-hglxc:80(http)  192.168.24.253:36546                               to-stack      FORWARDED   TCP Flags: ACK, FIN
May  3 17:42:32.522   192.168.24.253:36546                          guestbook/frontend-59f5db8cfb-hglxc:80(http)       to-endpoint   FORWARDED   TCP Flags: ACK, FIN
May  3 17:42:32.523   192.168.24.253:36546                          guestbook/frontend-59f5db8cfb-hglxc:80(http)       to-endpoint   FORWARDED   TCP Flags: ACK
May  3 17:42:37.684   guestbook/redis-slave-96685cfdb-9vdqq:40722   guestbook/redis-master-596696dd4-lhnq5:6379(redis) to-stack      FORWARDED   TCP Flags: ACK, PSH
May  3 17:42:37.684   guestbook/redis-slave-96685cfdb-9vdqq:40722   guestbook/redis-master-596696dd4-lhnq5:6379(redis) to-endpoint   FORWARDED   TCP Flags: ACK, PSH
May  3 17:42:37.684   guestbook/redis-master-596696dd4-lhnq5:6379(redis)  guestbook/redis-slave-96685cfdb-9vdqq:40722  to-stack      FORWARDED   TCP Flags: ACK
May  3 17:42:37.684   guestbook/redis-master-596696dd4-lhnq5:6379(redis)  guestbook/redis-slave-96685cfdb-9vdqq:40722  to-endpoint   FORWARDED   TCP Flags: ACK
```

<timestamp> < SRC + DST namespace, pod-name, port > <tracepoint> <verdict> <TCP flags>

What can I do now?

Jef's Picks for Good, Good First Issues

Cleaning up documentation is a great first contribution:

https://github.com/cilium/cilium/issues/14177
https://github.com/cilium/cilium/issues/15394
https://github.com/cilium/cilium/issues/18005
https://github.com/cilium/cilium/issues/21986

But there a lot more to choose from:

Ref: https://github.com/cilium/cilium/labels/good-first-issue

Ref: https://github.com/orgs/cilium/projects/3/views/1

Prerelease Testing

Great way to contribute while you're learning about Cilium features

New users/contributors:

- Test for documentation regressions
- Test feature interactions

Existing users/contributors:

- Great way to test for regressions in "production-like" configurations

Current Pre-release:
https://github.com/cilium/cilium/releases/tag/v1.15.0-pre.2

Docs: https://docs.cilium.io/en/v1.15.0-pre.2/

Pre-release Testing Issue Template:


cilium

Appendix

Reference links

- Getting started developing Cilium
- Getting started developing Tetragon
- Good first issues
- Cilium Community Repo

Resources deep dive

Cilium Nodes

- A **node** represents the networking status of a node in the cluster
- IP Address Management (IPAM):
  - By default Cilium uses the IP address range assigned to the host for IPAM
  - IPAM is pluggable, many alternative options are available
- Cilium reserves an extra couple of IP addresses from this range:
  - Attach an IP address to **cilium_host** interface for local connectivity
  - Optionally reserve an IP address for traffic using Service Mesh Ingress
- Each node is associated with a number of endpoints which are locally scoped
- A list of all Cilium nodes within a cluster can be inspected with:
  - cilium-dbg node list
  - kubectl get ciliumnodes

Cilium Endpoints

- An **endpoint** represents the networking state of a single pod on a node
- **CNI ADD**: **cilium-cni** calls **cilium-agent** in order to create an endpoint
  - Cilium assigns IP addresses (IPv4/IPv6)
  - Cilium generates BPF programs to implement all connectivity/visibility for the endpoint
- Cilium "regenerates" an endpoint when config state change (e.g., policies)
- State Includes:
  - Configured state:
    - IP address(es), MAC, Linux device
    - Security-identity
    - Per-endpoint configured policy (including default-deny status)
  - Realized state:
    - Regeneration status and the log of recent regenerations.
    - Policy in terms of low-level identities
- Host-networking pods are **not** managed by Cilium ⇒ don't show up in CEP
  - Internals: special endpoint representing the host itself

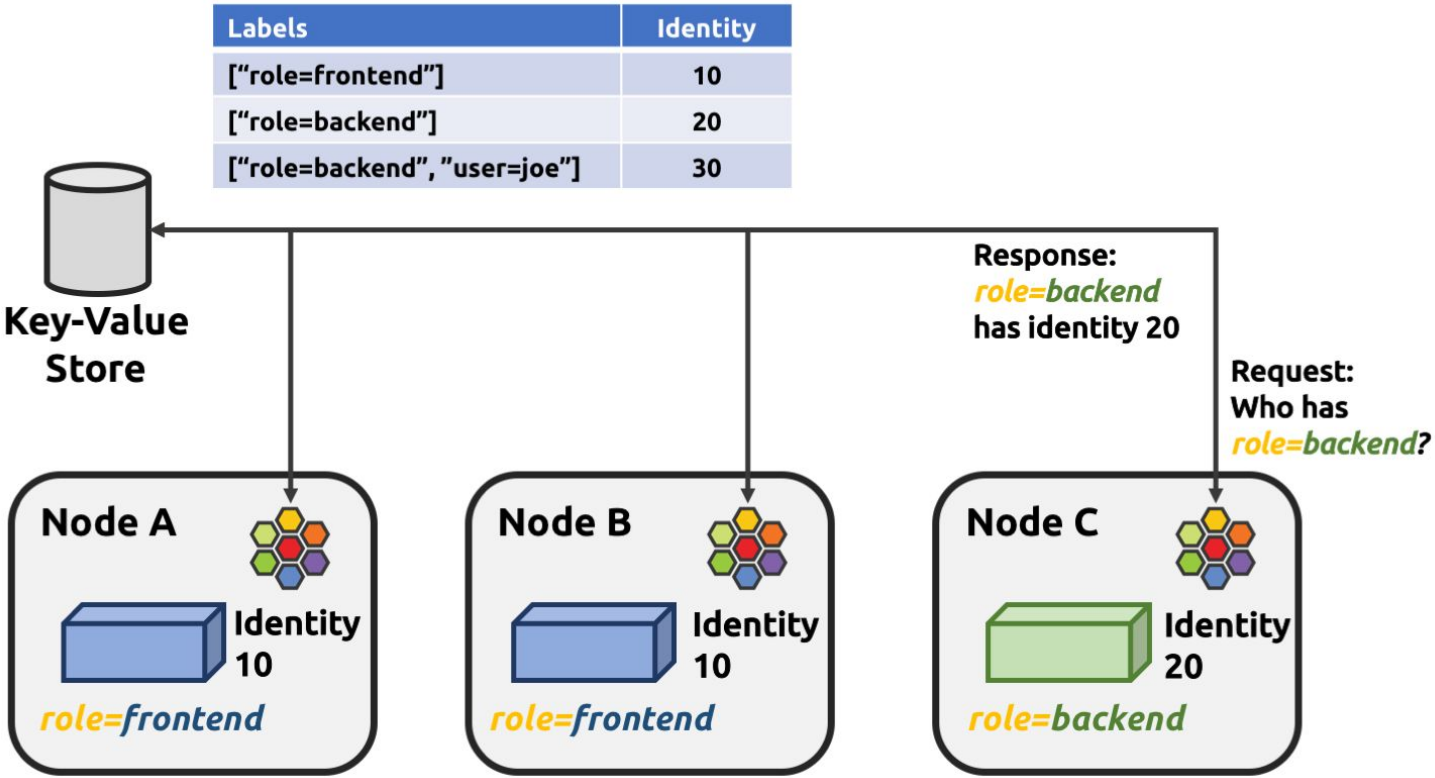
Inspecting Cilium Endpoints

- All cilium endpoints on a host can be inspected with
  - `cilium-dbg endpoint list`
  - `cilium-dbg endpoint get <endpoint-id>`
    - Contains deep detail of configured and realized state
- All Cilium endpoints in the cluster can be inspected with
  - `kubectl get ciliumendpoints --all-namespaces`
- The `CiliumEndpoint` CRD is used internally by Cilium to share information between cilium-agent pods
  - Endpoints are still node-local
  - Frequently changing endpoint state is not synced to CRD
    - Example: by default, no policy enforcement status updates

Cilium Security Identities

- Global **Identity** for each source/destination of a network connection for the purposes of more efficient datapath security enforcement/visibility
- Each endpoint has an identity.
- Pod “replicas” are different endpoints, but can share same identity.
  - app-xxxxx, app-yyyyy, app-zzzzz ⇒ 3 endpoints
  - identity=4323 ⇒ 1 identity
- The Cilium identity store ensures consistent identities across all worker nodes
- Useful commands:
  - cilium-dbg identity list
  - kubectl get ciliumidentities

Cilium Security Identities



Cilium Policies

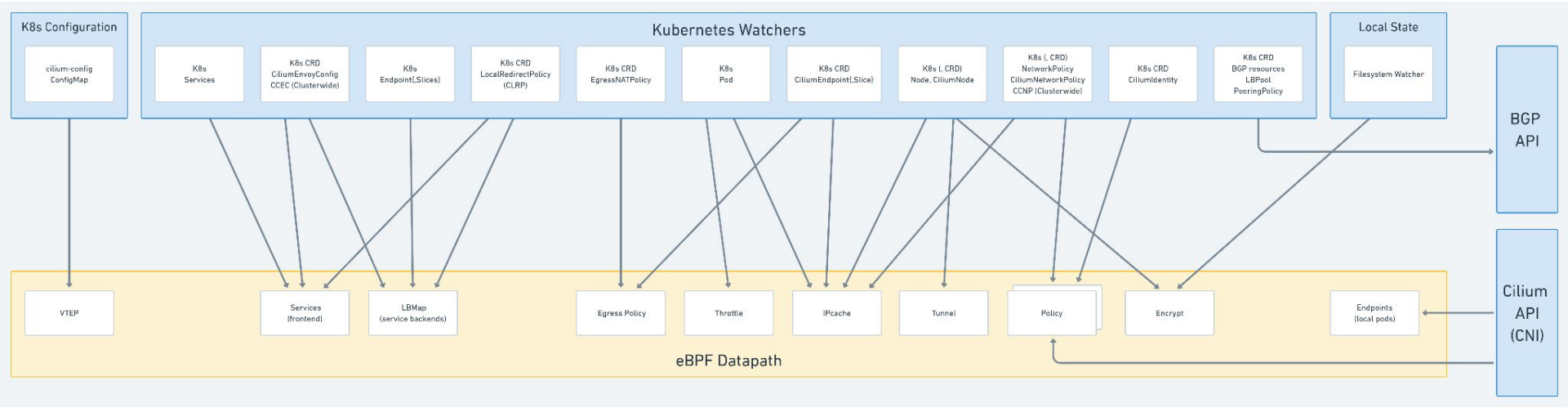
- Policies define limits to ingress and/or egress connectivity of endpoints
- Policy state is defined centrally via the K8s API as
  - Standard Kubernetes **NetworkPolicy** objects
  - More feature-rich **CiliumNetworkPolicy** and **CiliumClusterwideNetworkPolicy** objects
    - DNS-aware policy, L7-aware policy, host firewall, etc…
- **cilium-agent** syncs both types of policies and implements filtering via BPF
- Cilium follows core policy behaviors from K8s network policy spec:
  - **Example**: Default allow for ingress/egress if no policy is applied to endpoint.
- Useful commands:
  - cilium-dbg policy get
  - kubectl get networkpolicy --all-namespaces
  - kubectl get ciliumnetworkpolicy --all-namespaces
  - kubectl get ciliumclusterwidenetworkpolicy --all-namespaces

Cilium Services

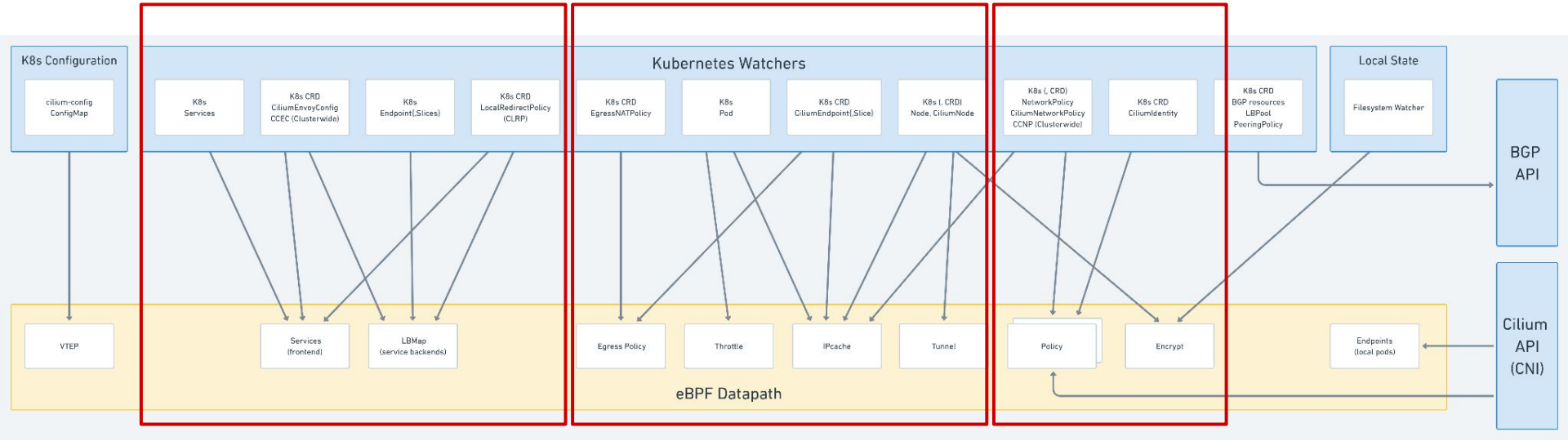
- A service describes L4 load-balancing for pods
- Maps directly to Kubernetes Services + endpoints
    - Cilium watches this data from the K8s API
    - Creates BPF state in datapath to perform service to endpoint LB
- Behaviour:
    - Cilium performs pod-to-pod LB by default on all kernels
    - Remaining LB behaviour (NodePort, etc…) is controlled by kube-proxy-replacement flag
- Useful commands:
    - cilium-dbg service list
    - kubectl get svc --all-namespaces

Resource data flow

Cilium Data Flow



Cilium Data Flow



Services,
Intra-node Routing

Connectivity,
Inter-node Routing

Security,
Network Policy