
KubeCon
CloudNativeCon
North America 2023


KubeCon | CloudNativeCon

North America 2023

Scaling Kubernetes Networking to 1k, 5k, ... 100k Nodes!?

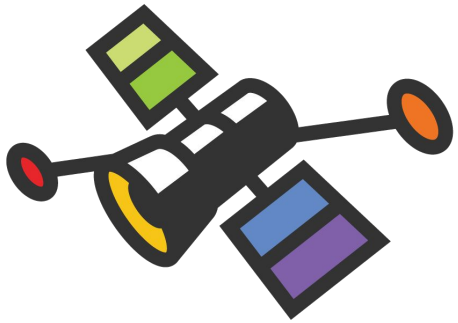
Marcel Zieba, Isovalent

Dorde Lapcevic, Google

What is Cilium?

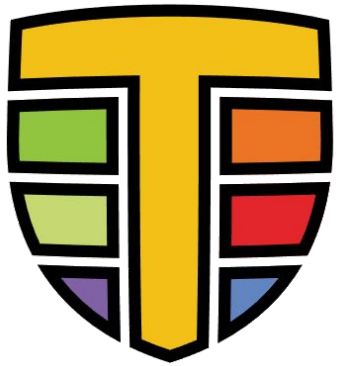
Cilium CNI

Scalable, Secure, High Performance CNI Plugin



Hubble

Network Observability

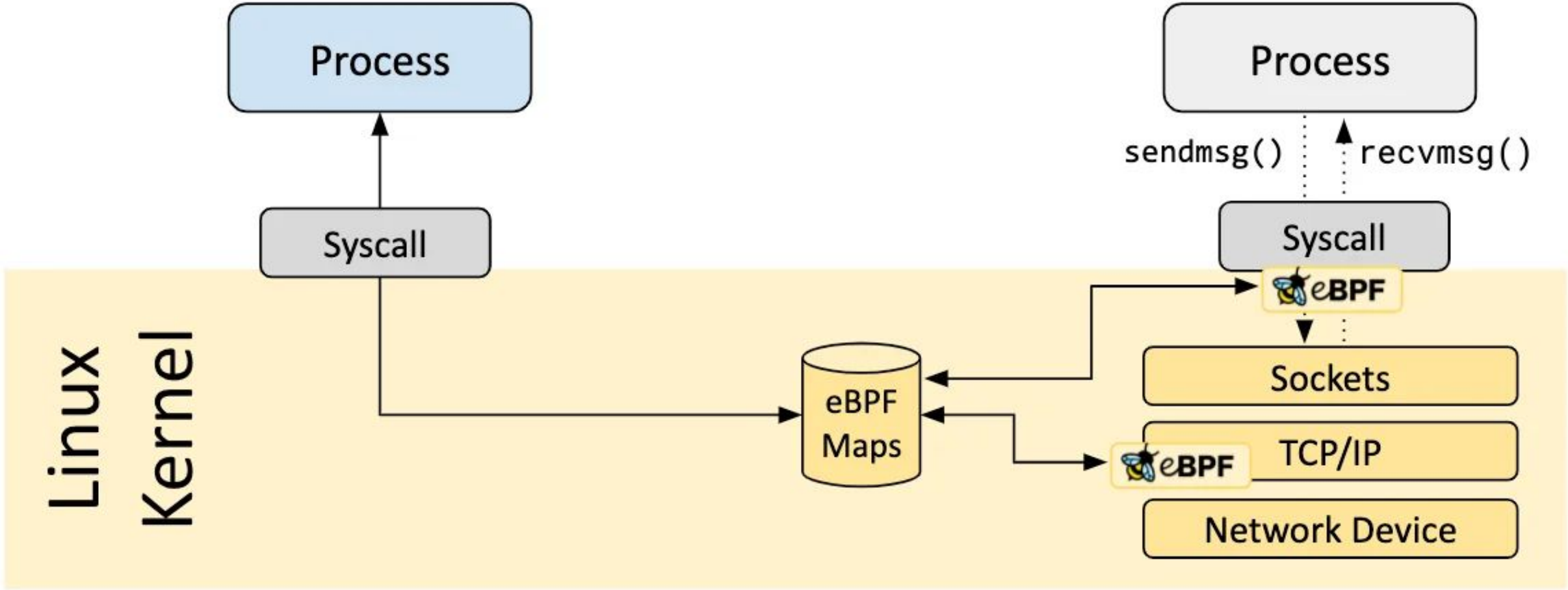


Tetragon

Security Observability & Runtime Enforcement

eBPF

What is eBPF?


Process
Syscall
Linux Kernel
eBPF Maps
Process
sendmsg()
recvmsg()
Syscall
eBPF
Sockets
eBPF
TCP/IP
Network Device

Cilium CNI overview

Efficient and scalable Kubernetes CNI
- IPv4, IPv6, NAT46, SRv6, …
- Overlays, BGP, …

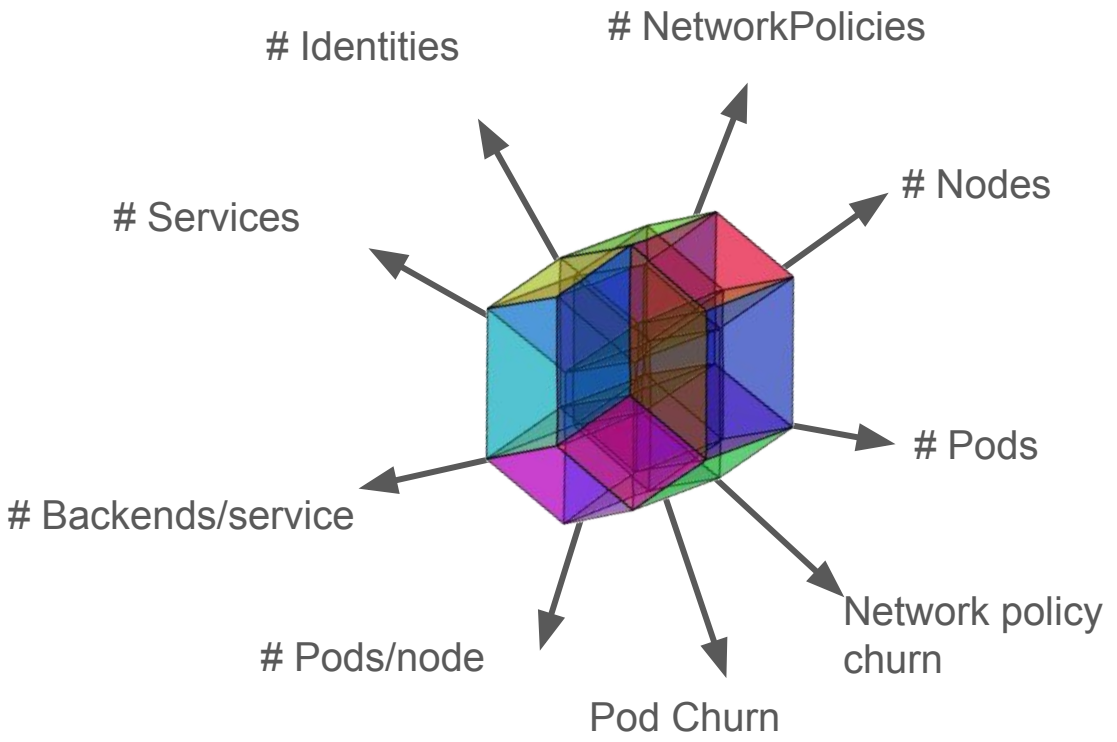
Security
- Kubernetes Network Policy
- Cilium Network Policy (FQDN, L7, …)
- Transparent Encryption

High-performance load balancing
- Kubernetes proxy replacement
- North-South load balancing

Multi-cluster & external workloads
- Global services, service discovery
- Integration of Metal and VMs

But… what does scalability even mean?

~~Scalability = # nodes~~



Networking Scalability SLIs/SLOs

- Pod Startup Latency
- Node Startup Latency
- Network Programming Latency
- Network Policy Enforcement Latency
- In-Cluster Network Latency


KubeCon | CloudNativeCon

North America 2023

Network security on a large scale

Network security on a large scale

How workload security with K8s network policies scales up to 5k nodes and 200k pods on a single cluster?

We are going to cover:

- Target scale
- Network policy implementation
- What challenges we overcame and how?
- Performance & metrics
- Improvements in progress

Network security on a large scale

Target scale

Scalability dimension	Limit
# Nodes	5k
# Pods	200k
Pod churn rate	100 per second
# Network policies	10k
Network policy churn rate	20 per second

Network policy implementation

- Custom resources:
  - Cilium Endpoint - An endpoint created for every pod.
  - Cilium Identity - A security identity created for every unique pod and namespace label set.
- Network policies select pod labels and namespace labels that should be allowed.
- Cilium-agent (daemonset) populates on every node:
  - Policy eBPF maps based on network policy rules and security identities
    - A policy map is created for every local pod
    - It contains a list of identities that are allowed to communicate with the pod
  - IPCache, IP to identity mapping for all pods in the cluster
- When establishing connection. verify if the matching identity for the peer IP exists in the policy map.

Network security on a large scale

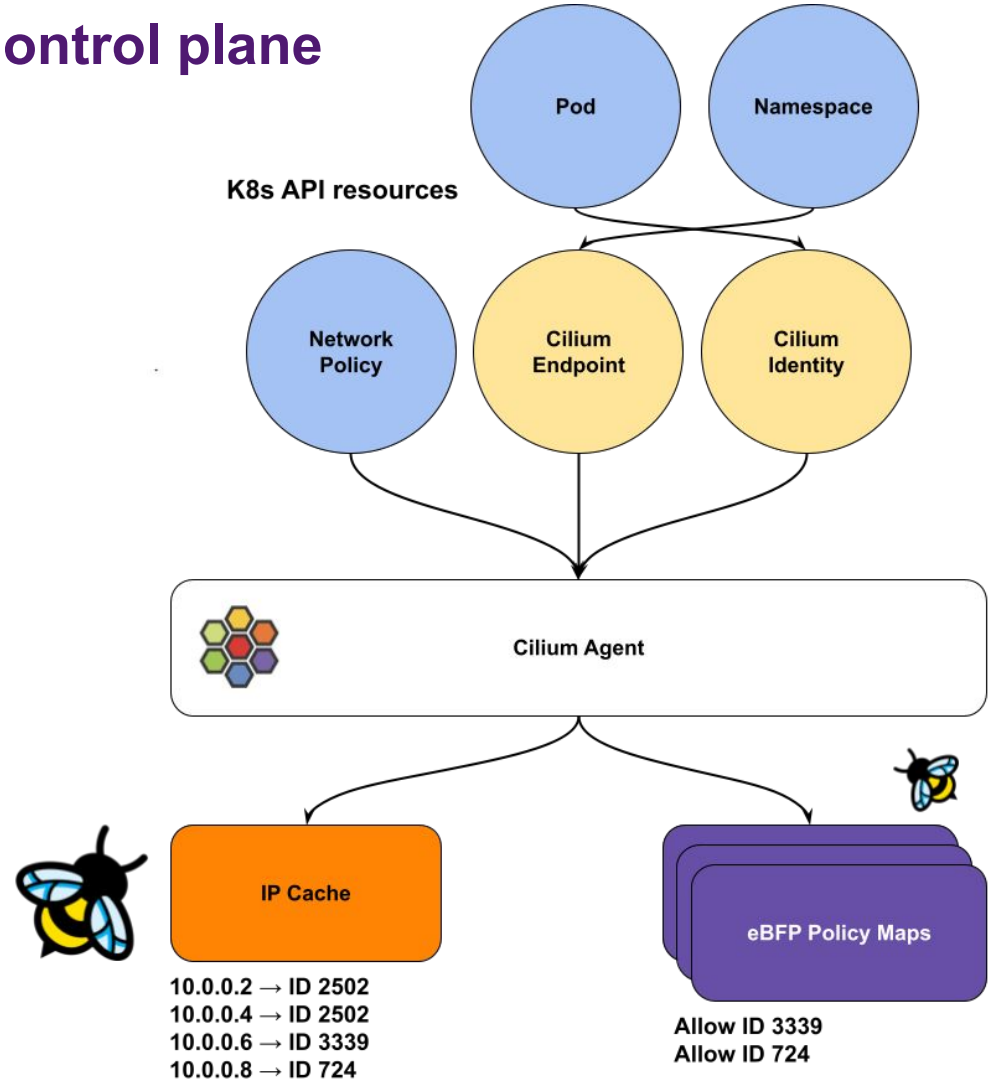
Security Identity 

- Security identity is generated from pod labels and namespace labels
- Network policies select pod labels and namespace labels
- Pod to pod communication is allowed only between pods that have selected identities

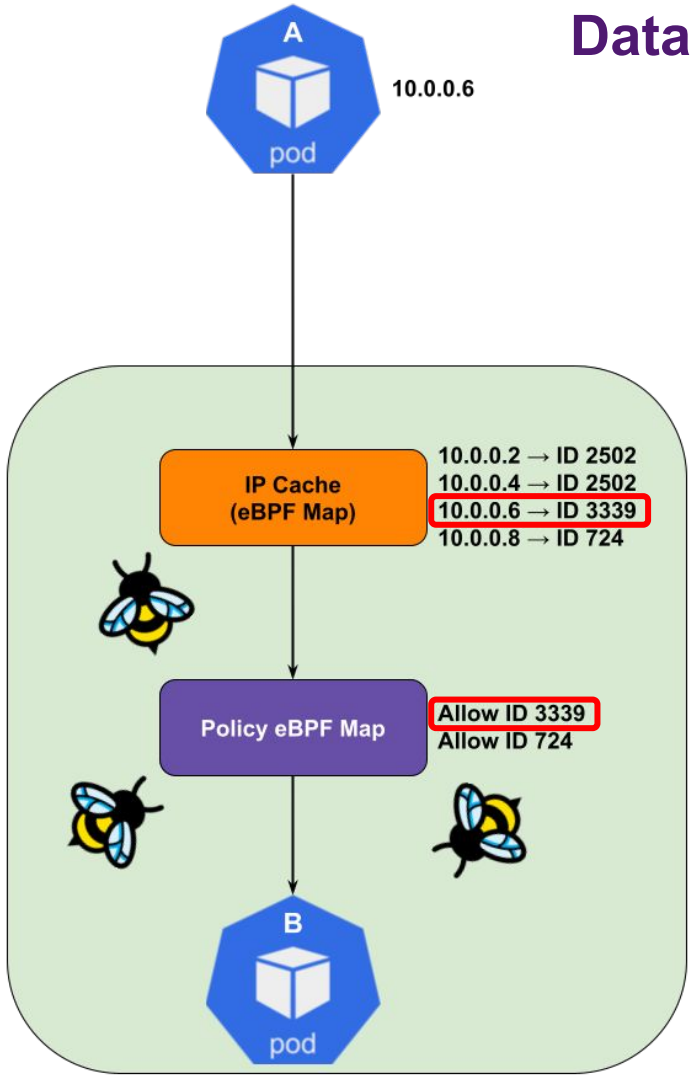
Network security on a large scale

Network policy implementation

Control plane



Data plane

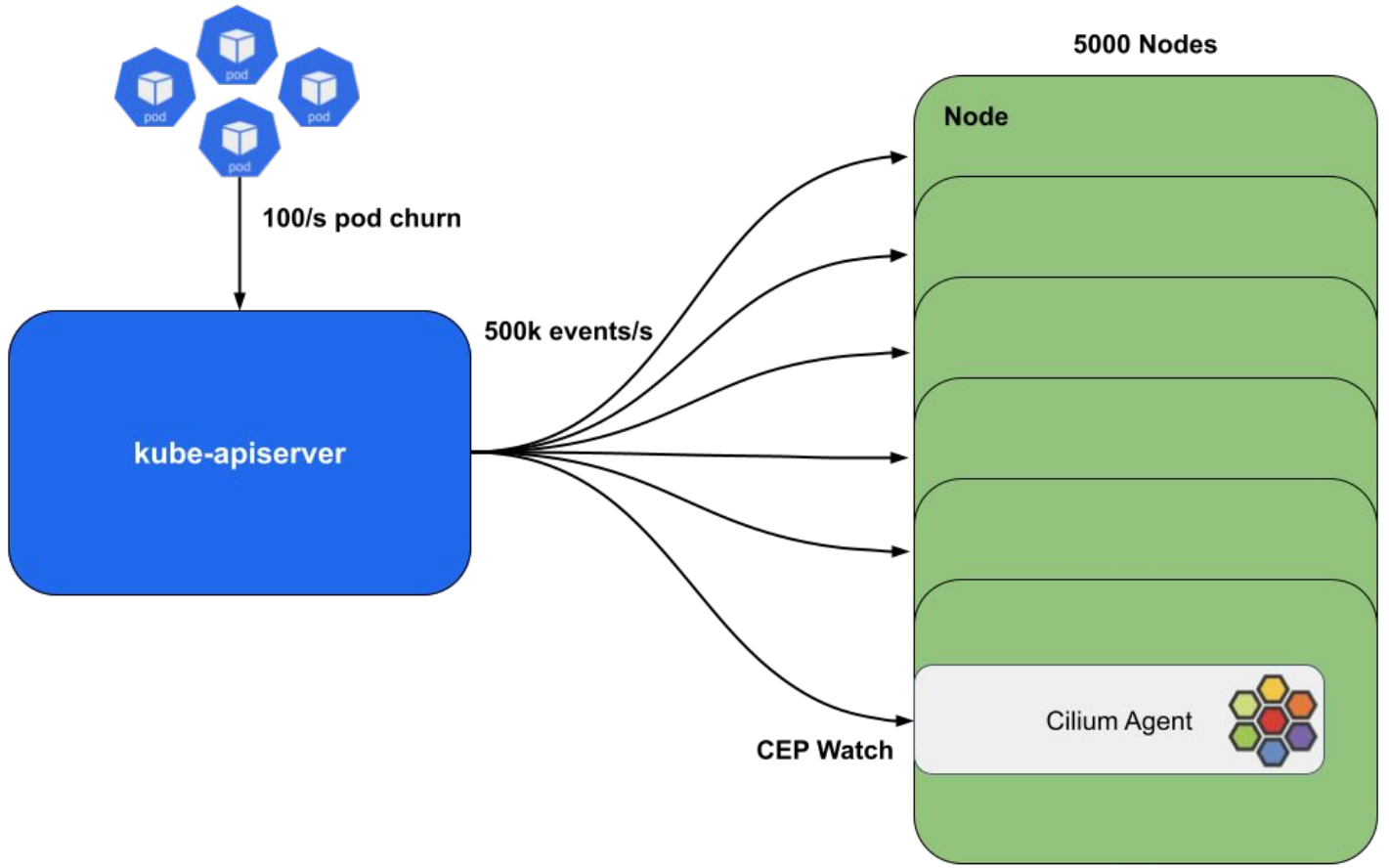


Network security on a large scale

What is the bottleneck?

k8s API events = Nodes * Pod changes

- All nodes need to know about pod IP to pod security ID mapping for every pod.
- 5000 nodes * 100 pod changes per second = 500k events per second
- Kube-apiserver on the most powerful VMs has trouble handling over 100k events/s. The safe limit is up to 1000 nodes for 100 pod changes per second.

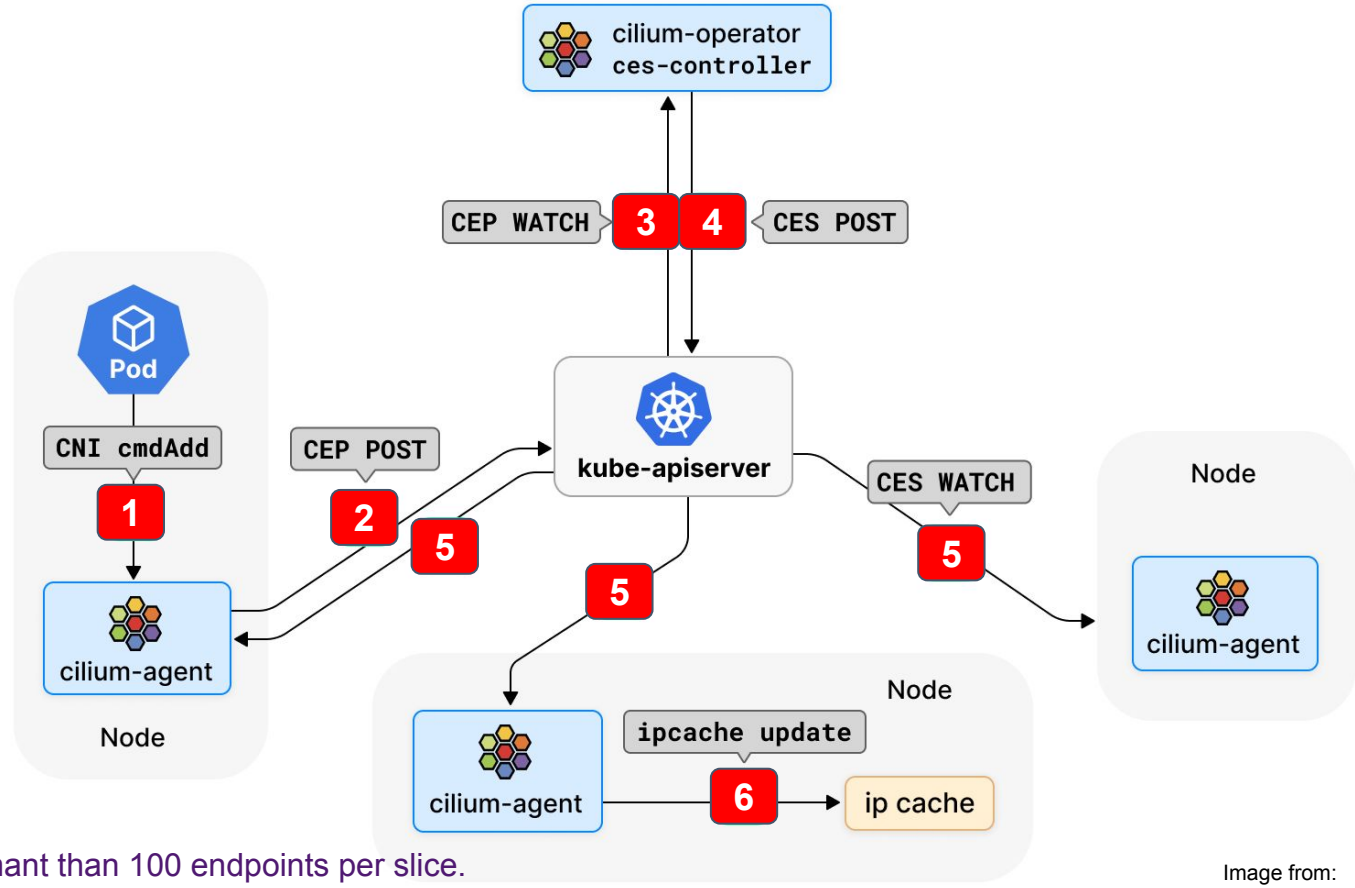


Network security on a large scale

What did we do to overcome it?

Batching Cilium Endpoints ✓

- Inspired by K8s Endpoint Slice
- High number of events of small objects
- Slice the entire pool of endpoints
- Batch them into groups of 50*

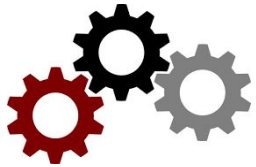


* Based on scalability tests size of 50 endpoints per slice proved to be more performant than 100 endpoints per slice.

Image from: https://isovalent.com/blog/post/2021-12-release-111/#cilium-endpoint-slices

Network security on a large scale

Why does it work?



- Overhead of handling a very high number of events in KCP heavily impacts the performance.
- Sending fewer larger events enables kube-apiserver to handle 100/s pod change rate.
- Cilium Endpoint Slice contains the minimum amount of data from Cilium Endpoints. Having security identity (int64) instead of a set of labels (list of strings) significantly reduces the size of each endpoint.
- One slice contains 50 endpoints. A full slice is on average 5 times smaller than 50 Cilium Endpoints.*

* Measured in json format. Cilium Endpoint size: ~2.2 kB; Full Cilium Endpoint Slice size: ~20.8 kB

Network security on a large scale

Performance

- Cilium Endpoint batching enables the scale to increase from 1000 nodes to 5000 nodes, with 100/s pod churn, and allow for more pods to be running in the cluster, up to 200k pods.

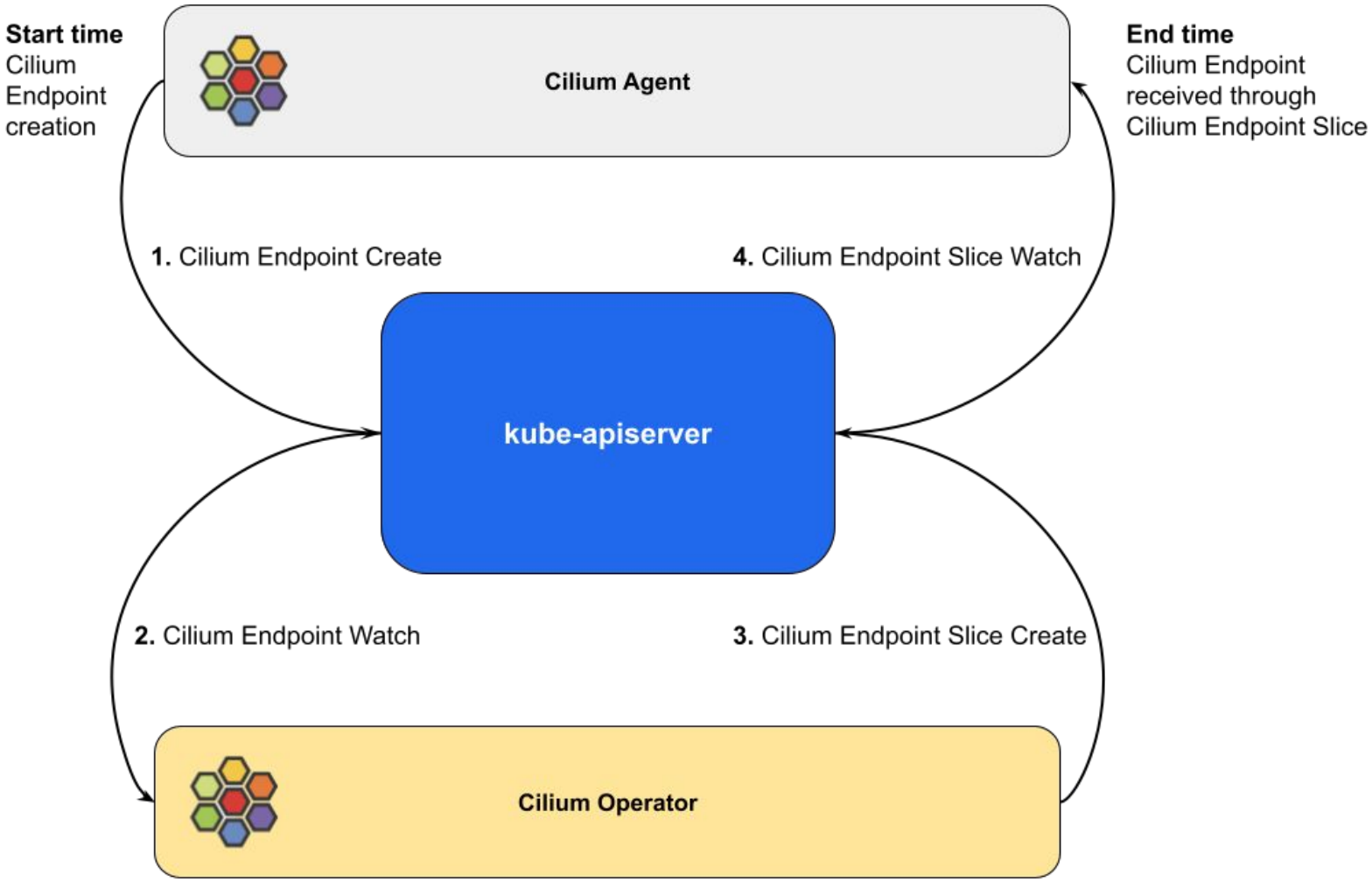
- Cilium Endpoint Slice updates rate limited to 10 per second on 5000 nodes.
  - Up to 500 pod updates per second.
  - Up to 50k Cilium Endpoint Slice events per second.

- Worst case scenario - Update all 200k pods at once
  - 6.67 minutes (400 seconds) = 200k pods / 500 pod updates per second

Network security on a large scale

Metrics / SLI

- Cilium Endpoint propagation delay metric represents Network policy enforcement latency
- Policy programming latency on each node takes low time (<5 sec) after propagation of endpoints, regardless of scale



Network security on a large scale

What are other challenges that we are facing?

Cilium Identity limits 

- Hard limits
  - 65k security identities per cluster
  - 16k security identities per policy eBPF map*
- Triggers
  - All pods have a unique label set
  - Identity duplication
- Remaining issue
  - Namespace label change

* Pods affected by allow-all network policies will be stuck in pod creation stage as long as there are over 16k security identities

Network security on a large scale

What improvements are we currently working on?

Centralized identity management ✓

- Move identity management to cilium-operator from a distributed management by cilium-agents
- Resolves Cilium Identity duplication
- Reduces pod startup latency - Cilium Identities are created on pod creation instead of Cilium Endpoint creation
- Security improvement - cilium-agent loses permission to write to Cilium Identity
- Enables further improvements and optimizations

Network security on a large scale

What are the improvements that we expect to have soon?

- Greatly reduce the number of security identities - “Security relevant labels filter”
- Improve performance and reliability depending on scale - “Dynamic Cilium Endpoint Slice update rate limiting”
- Faster policy enforcement for system critical pods - “Priority propagation of Cilium Endpoint Slices”
- Reduce policy enforcement latency on a large scale* - “Kube-apiserver optimization for processing events for many watchers (K8s 1.29)”

* https://github.com/kubernetes/kubernetes/pull/119801 + https://github.com/kubernetes/kubernetes/pull/120300

Network security on a large scale

Long term goal

- Continue stretching the limits of scalability dimensions with complete and performant network security support
- The improvements in progress make us very hopeful, but there is still a lot of work to be done, and we want to continue thinking more ahead
- Support a wide variety of cases to the full extent by utilizing different configurations (dynamically adjusted) and accepting different trade-offs
- Recognize that “one size fits all” is a much more difficult path when feature and scalability requirements are very diverse


KubeCon | CloudNativeCon

North America 2023

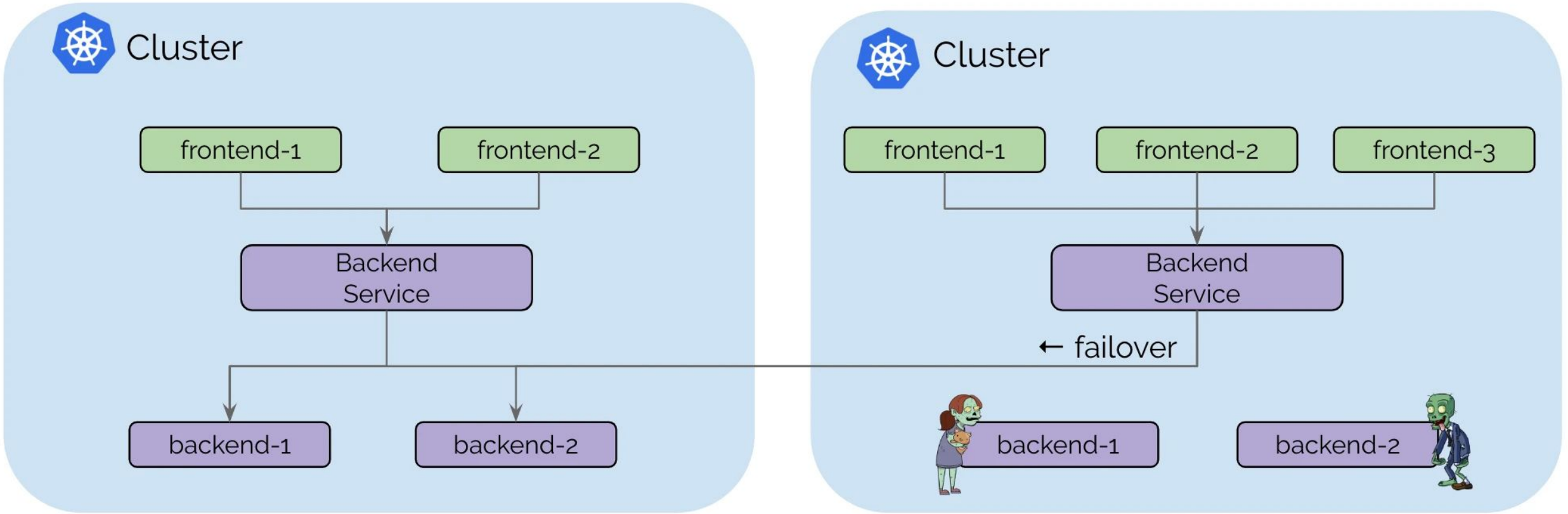
Cilium Clustermesh

Clustermesh in a nutshell

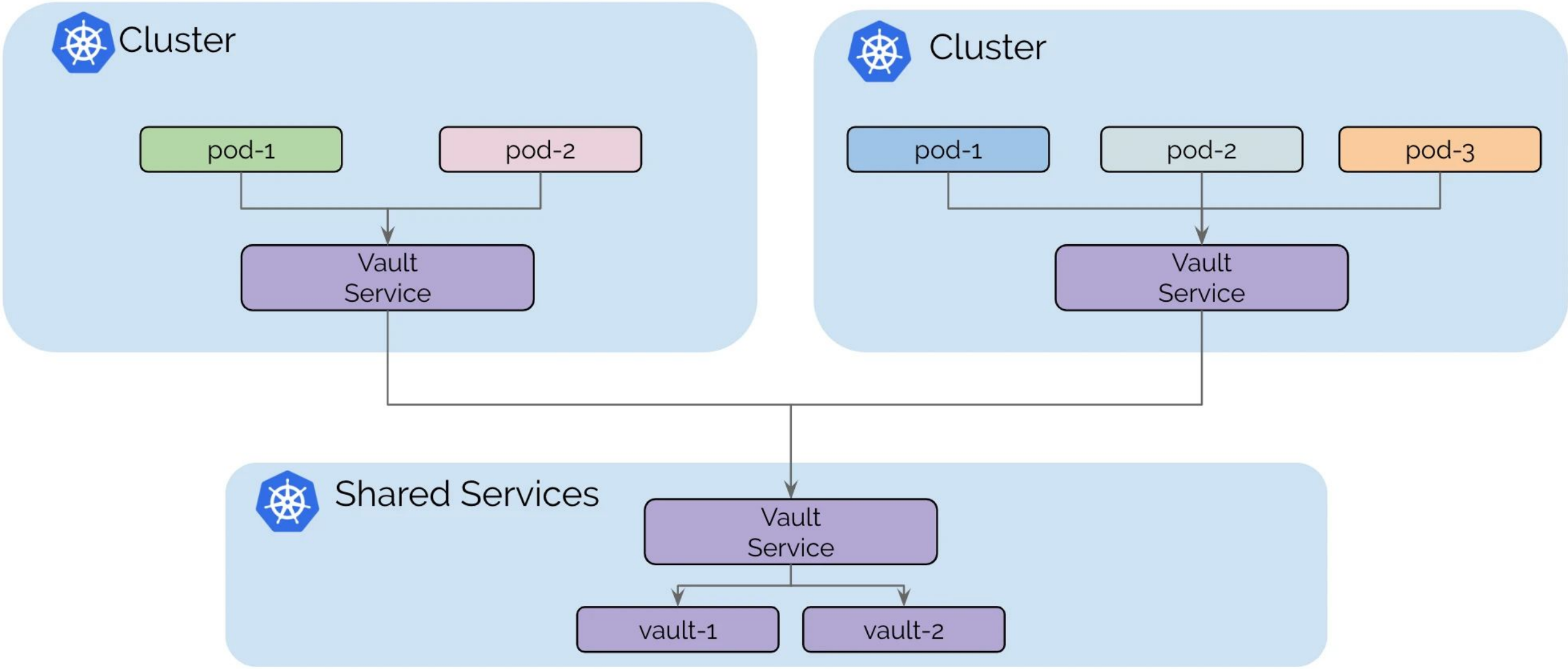
Clustermesh provides:

- Cross-cluster connectivity for your workloads
- Transparent service discovery with standard Kubernetes services and coredns/kube-dns
- Network policy enforcement spanning multiple clusters.
- Transparent encryption for all communication between nodes in the local cluster as well as across cluster boundaries.

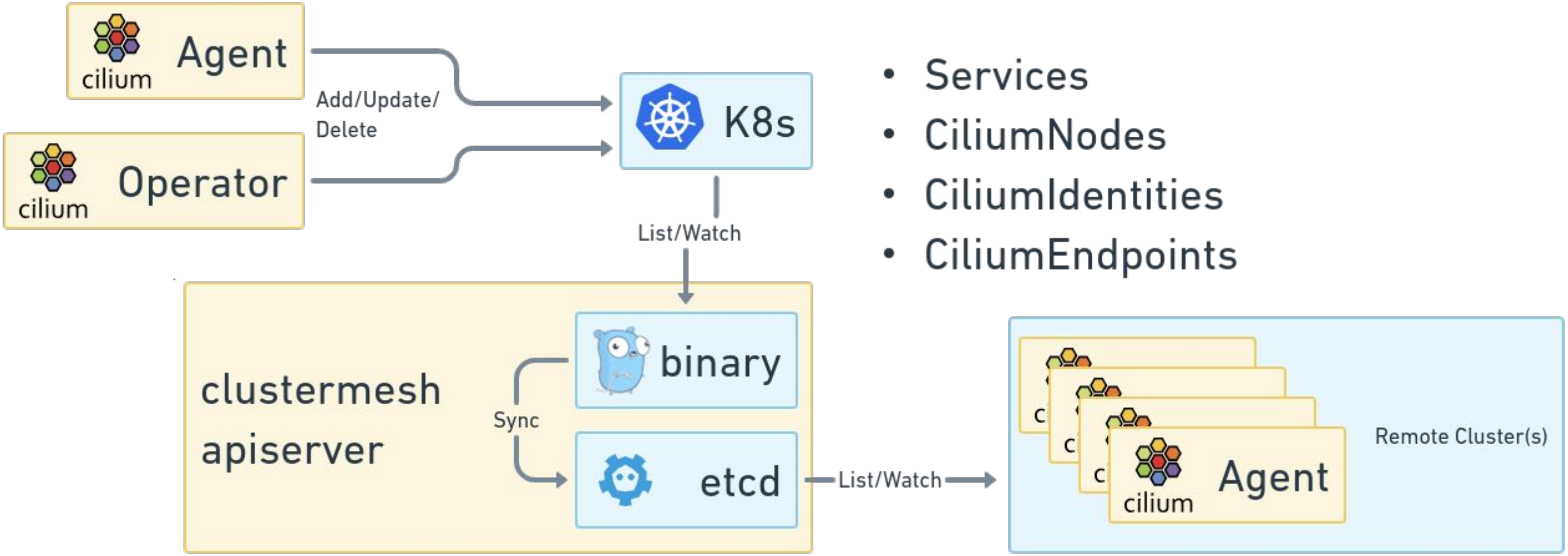
Use case: High availability



Use case: Shared services



Initial architecture

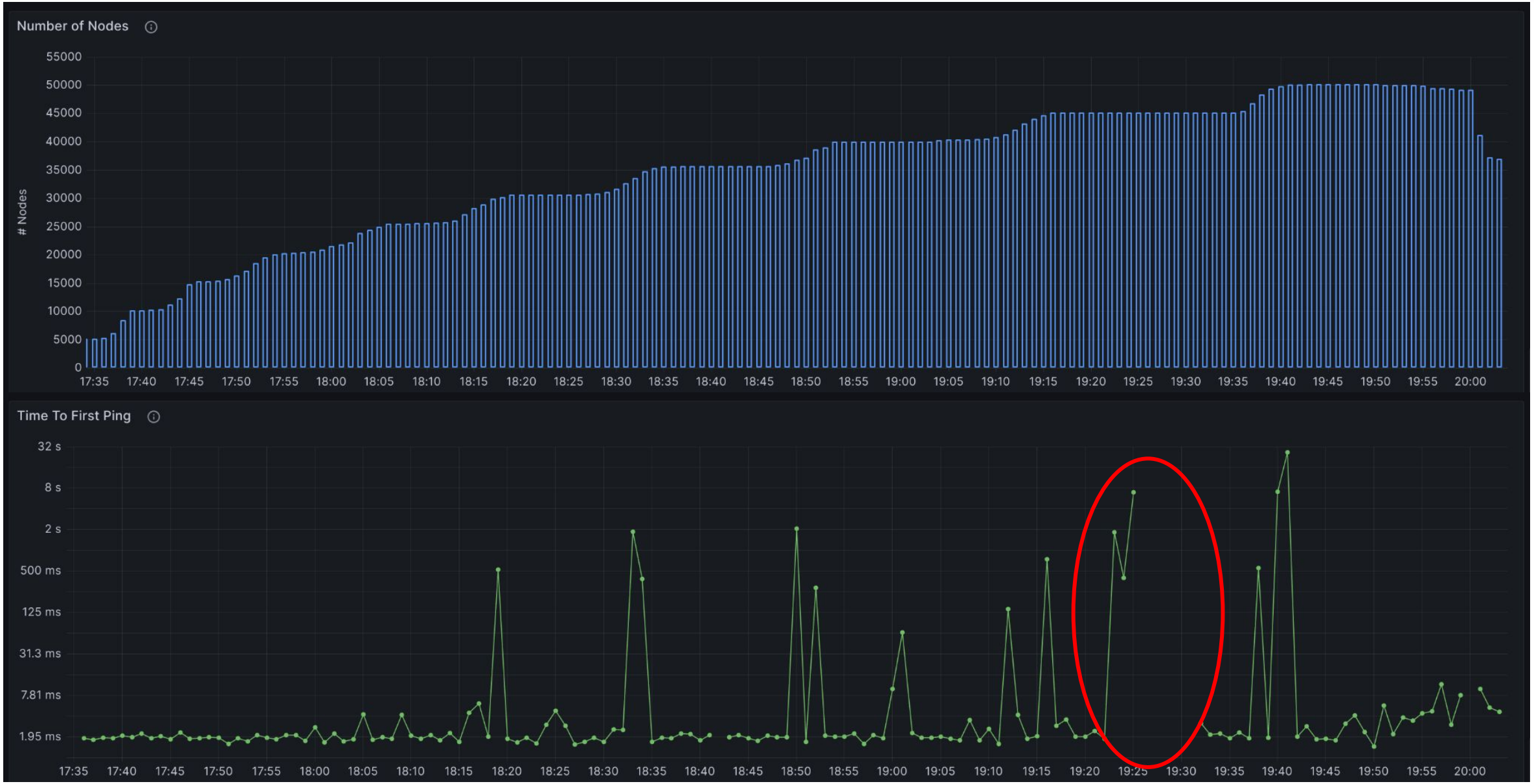


- Services
- CiliumNodes
- CiliumIdentities
- CiliumEndpoints

Clustermesh target scale

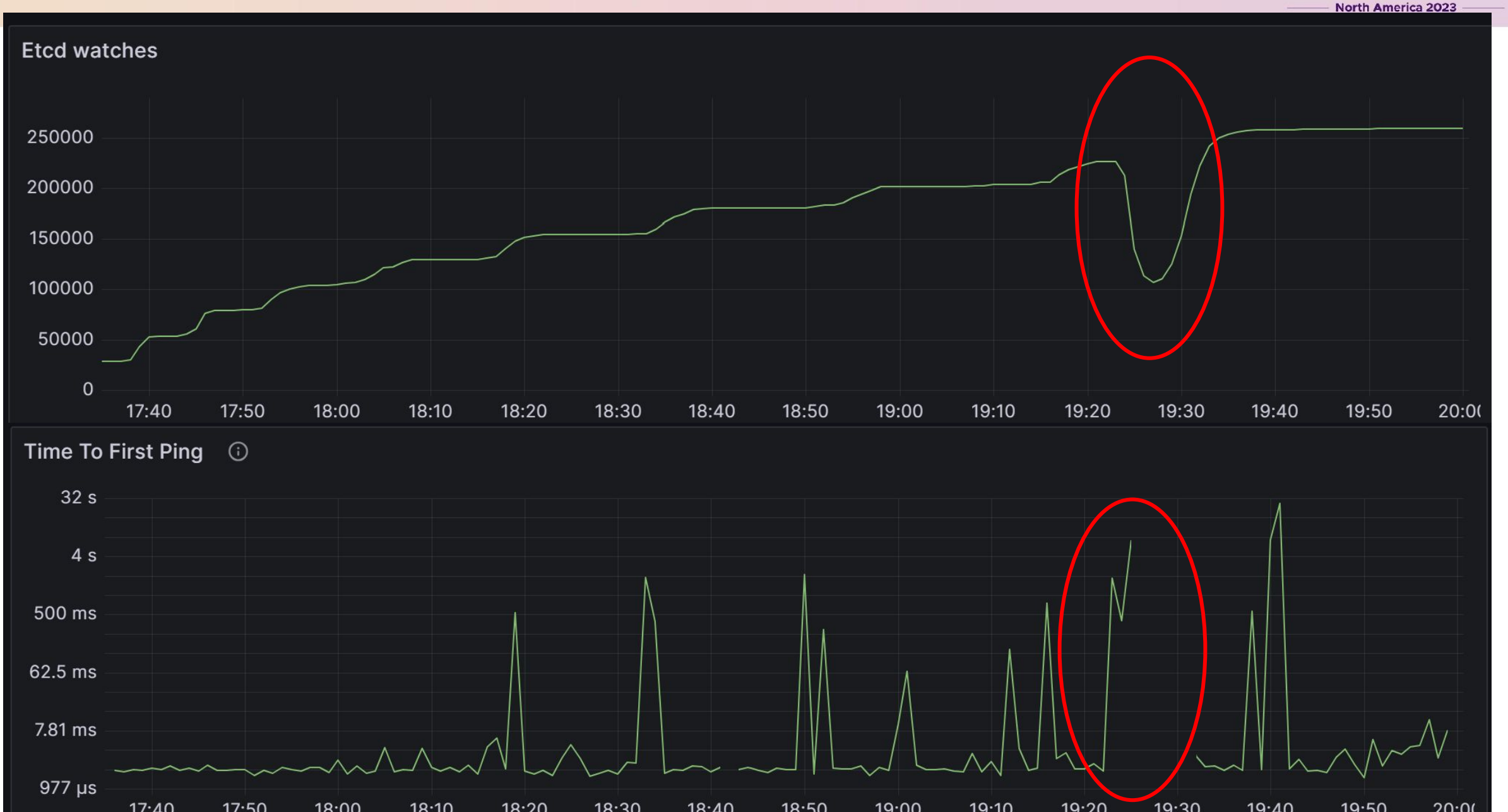
Scalability dimension	Limit
# Clusters	255
# Nodes	50k
Node churn	<50 nodes per second
Endpoint propagation rate	100 per second
# Pods in total	500k

Scale test results

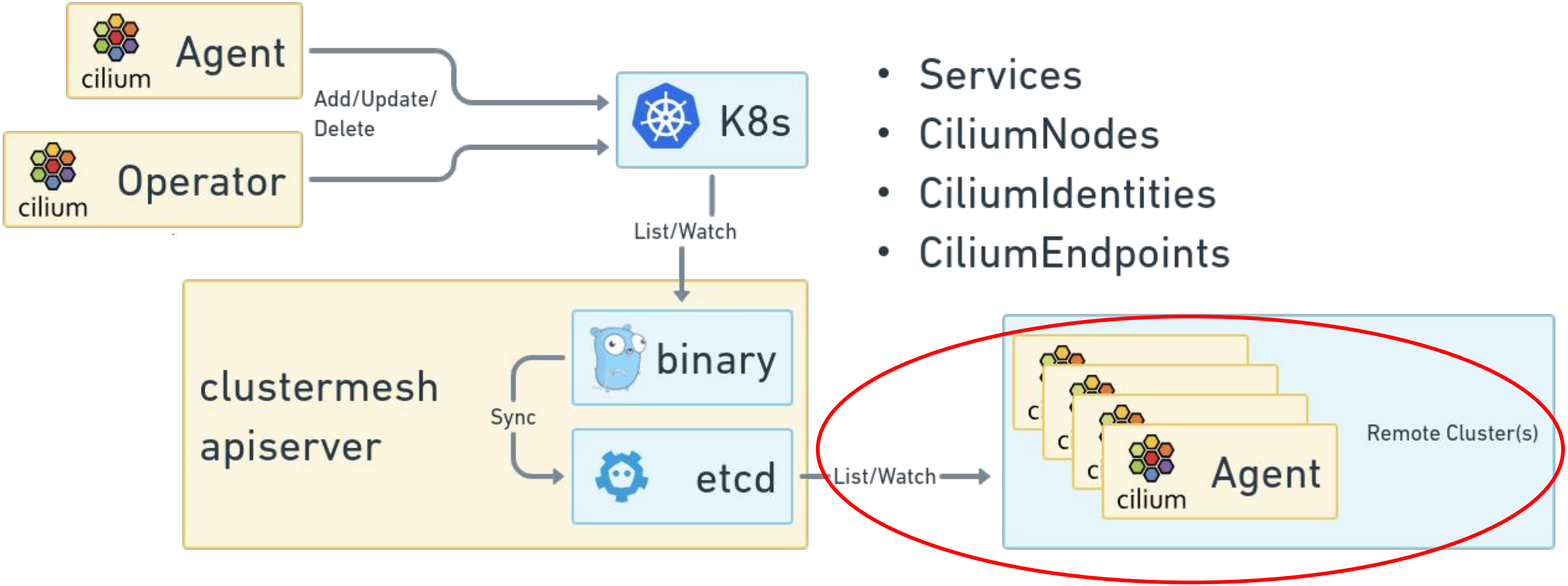


Scale test results

Scale test results

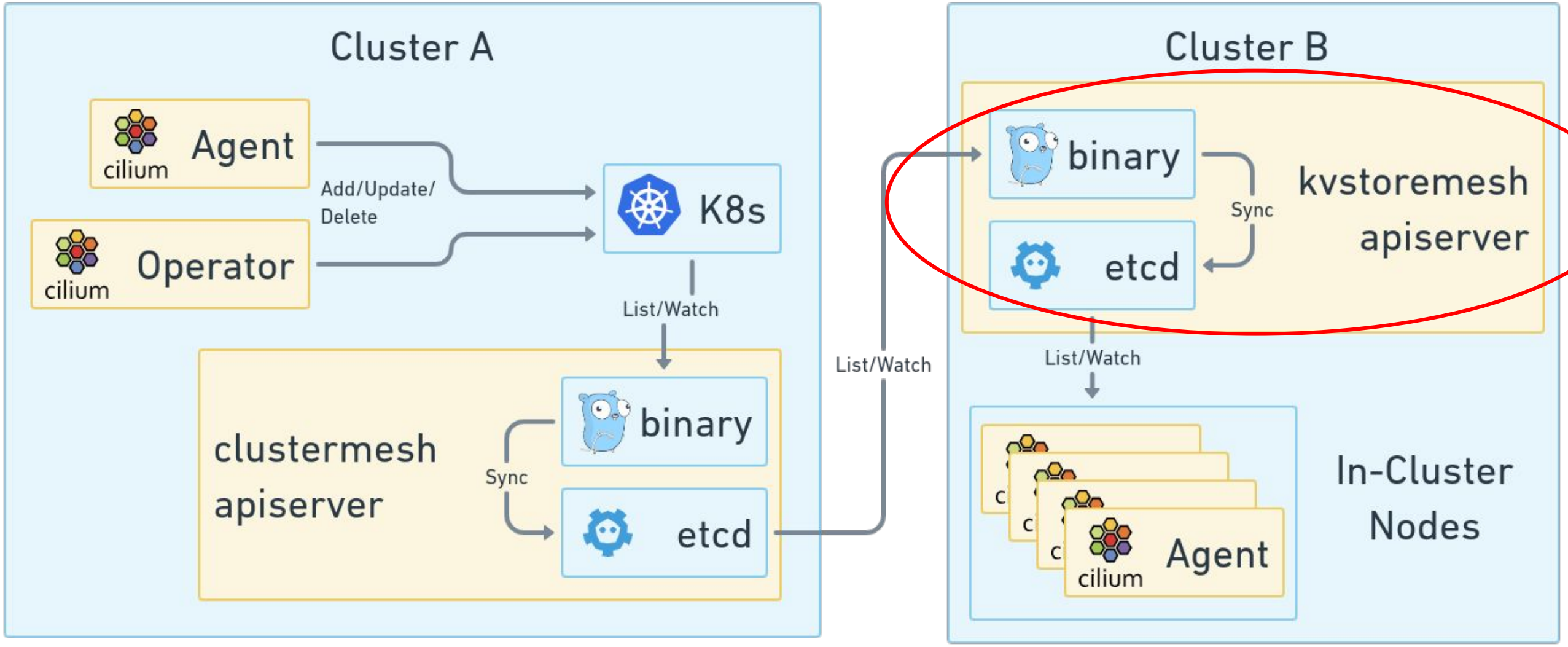


Bottleneck in initial architecture

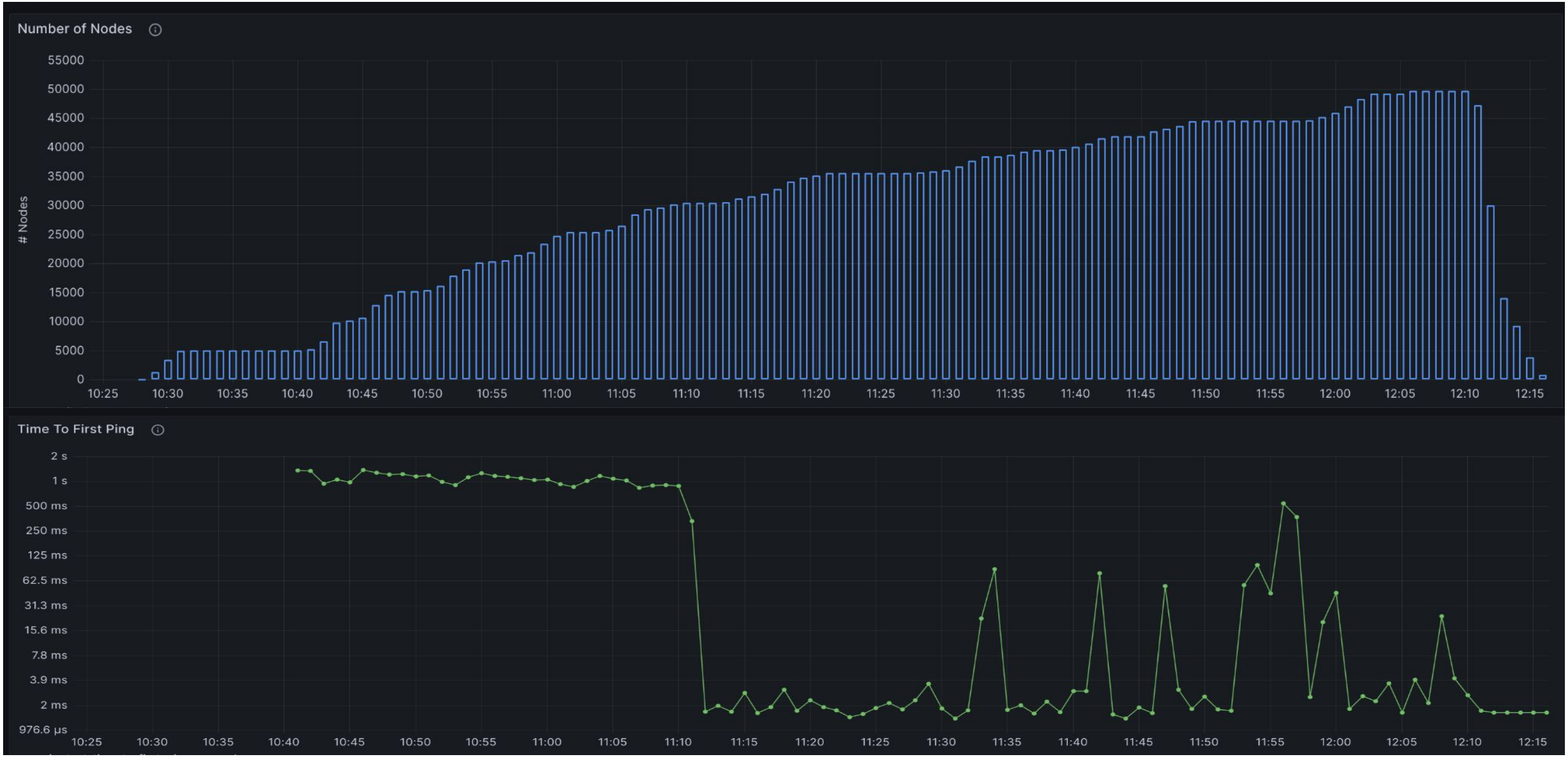


- Services
- CiliumNodes
- CiliumIdentities
- CiliumEndpoints

KVStoreMesh architecture



KVStoreMesh scale test results



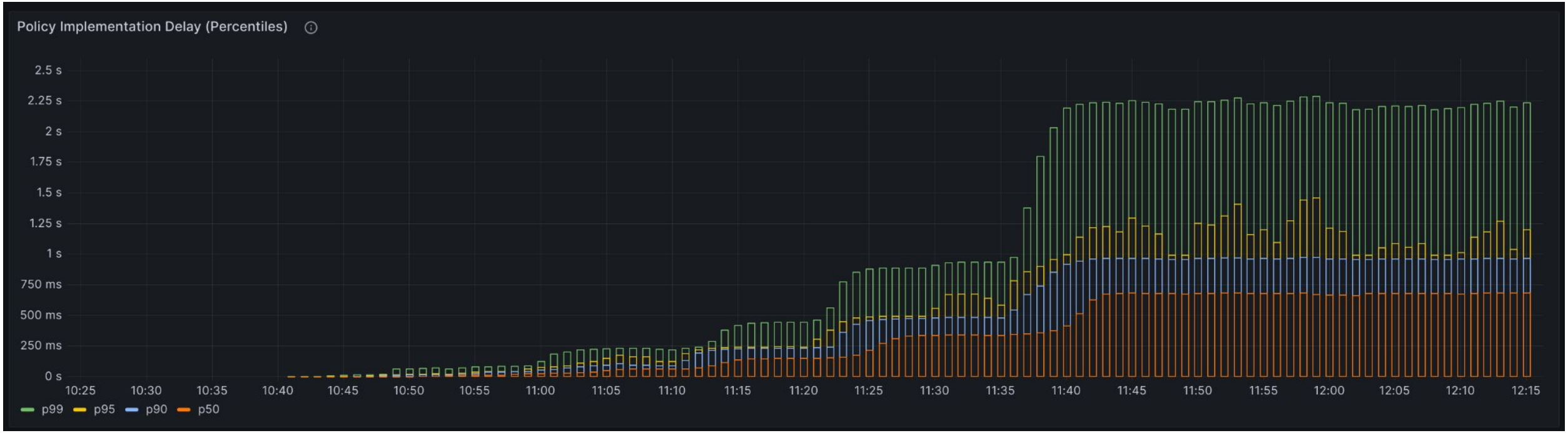
KVStoreMesh scale test results

Future KVStoreMesh improvements

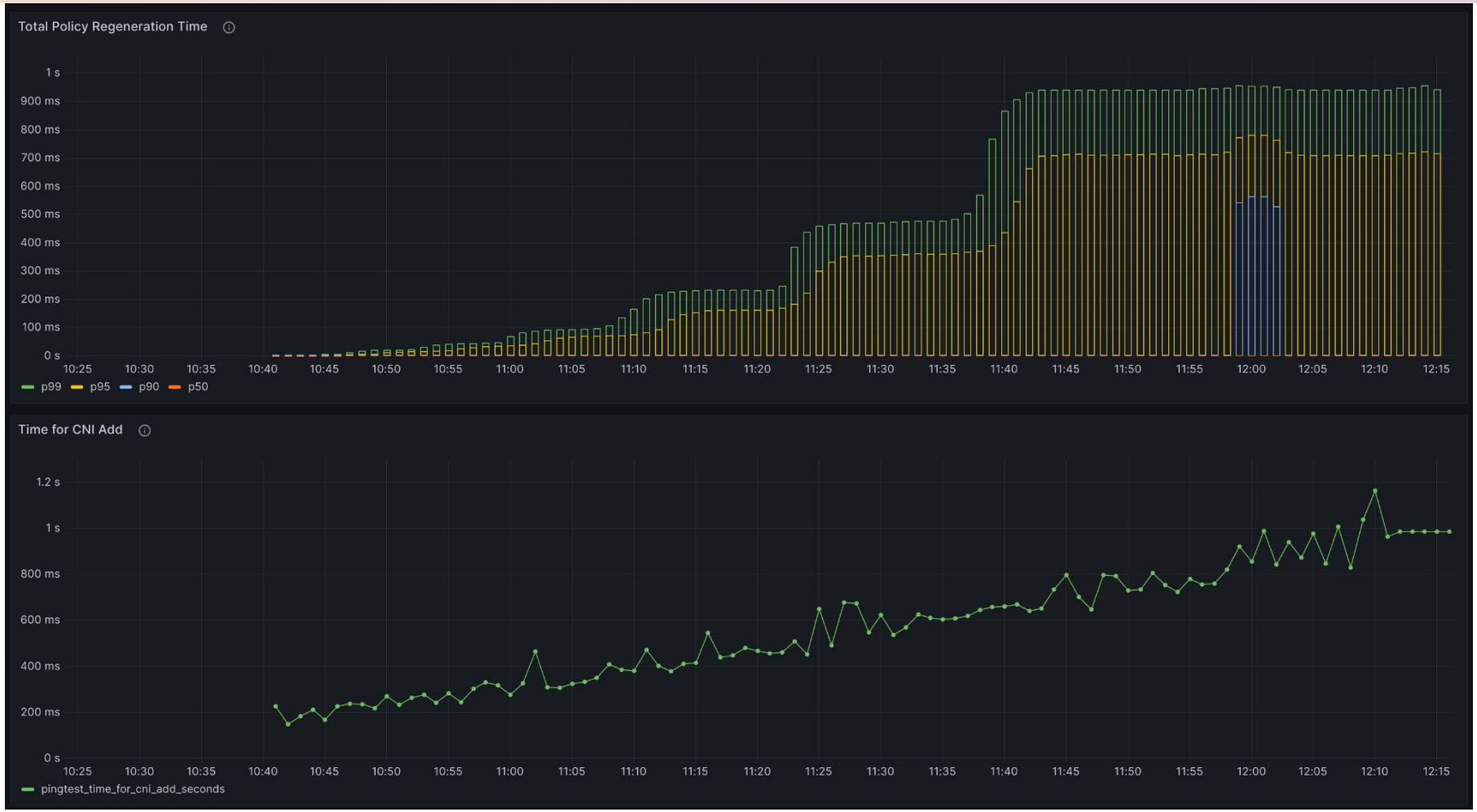
- Support for 511 clusters in clustermesh
- Increase propagation rate of endpoints to 500/s
- Reduce initialization time of Clustermesh control plane

Please scan the QR Code above
to leave feedback on this session

KVStoreMesh



KVStoreMesh



Latency measurement