Connecting Your Laptop to Present

CiliumCon NORTH AMERICA

If you plan to use speaker notes, please follow the steps below to ensure your laptop is set up correctly. This will allow you to view your notes while the audience sees only your slides.

Before Connecting

- Close all unnecessary apps & tabs
- Turn off notifications
- Disable Night Mode / TrueTone
- Don’t enter full-screen presentation mode yet

At the podium

- Plug in the HDMI cable
- If needed, use your USB-C to HDMI adapter
- On Mac, click “Allow” if prompted

USE “EXTENDED DISPLAY” MODE (NOT MIRRORED)

Using MAC

- Go to System Settings > Displays
- Click the “Arrangement” tab
- Make sure “Mirror Displays” is unchecked

Using PC

- Press ⊞Win + P, then select “Extend”
- Or: Right-click on the Desktop > Display Settings
- Under Multiple Displays, select “Extend these displays”

Connecting Your Laptop to Present

CiliumCon NORTH AMERICA

When Using PowerPoint

- Start slideshow: ⌘Cmd + Enter (Mac) or F5 (PC)
- Or: Click Slide Show > Play from Start
- If notes show on the wrong screen, click “Swap Displays”
- **https://www.youtube.com/watch?v=gQ3D4m-5pww**

Google Slides

- Use Google Chrome as your browser
- Click the dropdown next to ‘Slideshow’ > select Presentation Display Options
- Check “Presenter View” and “Full Screen”
- If prompted, select "Allow"
- Choose the external display (may have a strange name like PanasonicXYZ)
- Click Start Slideshow
- **https://www.youtube.com/watch?v=-GT7WCvPcys**



CiliumCon

NORTH AMERICA



CiliumCon
NORTH AMERICA

Don’t Make a Mesh of It

Multi-Cluster Networking at Cloud Scale



Anubhab Majumdar
Senior Software Engineer
Microsoft

Vamsi Kalapala
Software Engineer
Palantir

Vanessa Chammas
Software Engineer II
Microsoft

Agenda

CiliumCon NORTH AMERICA

01

The problem

Why multi-cluster at cloud scale is hard

02

Primer

Cluster Mesh & MCS-API in 60 seconds

03

Layer 1 · Sharing

Opt-in, scoped export

04

Layer 2 · Data path

Route by topology

05

Wrap-up

Tradeoffs, demo & takeaways

Two layers, one question each: **what crosses** and **how it crosses**

WHY NOW

Multi-cluster: from demos to production

CiliumCon NORTH AMERICA

Why teams run many clusters

- **High availability & failover**
- **Fault isolation**
- **Tenant separation**
- **Geographic latency**

MCS-API is shifting to real workloads

ServiceExport / ServiceImport and clusterset.local DNS are now part of Cilium Cluster Mesh

255

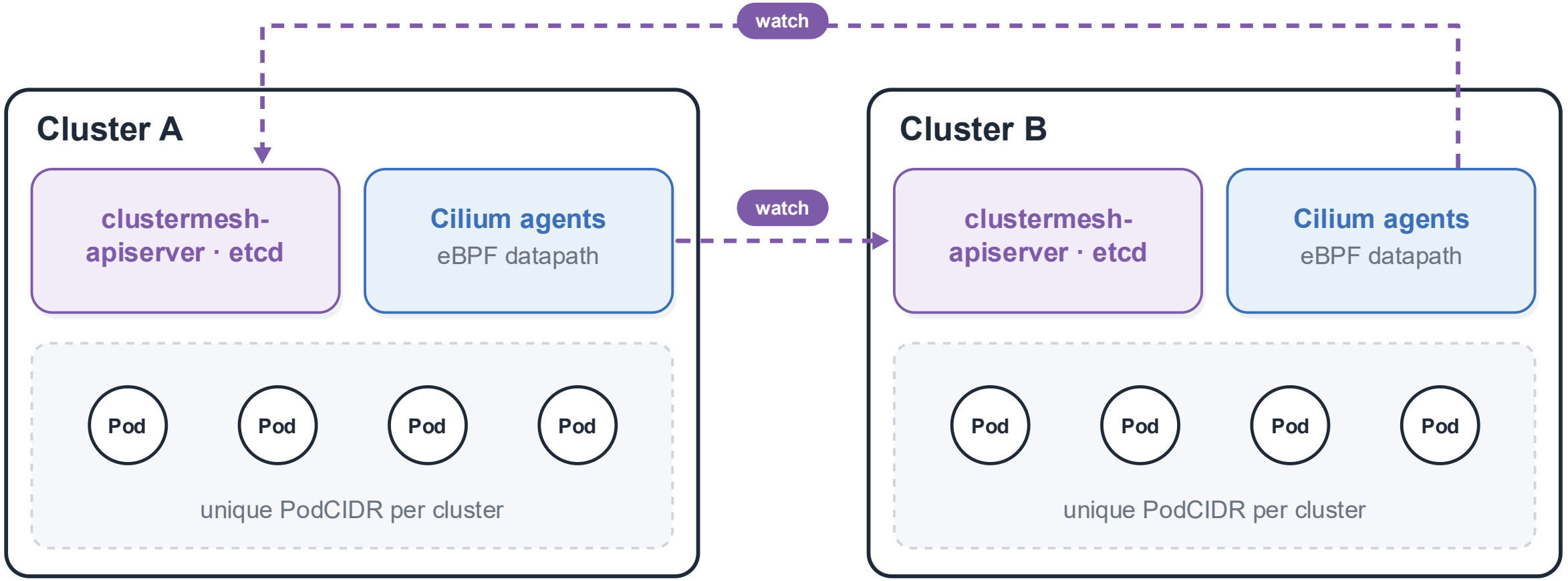
clusters per mesh · 8-bit cluster ID

200 × 300

clusters × nodes profiled for Cluster Mesh scaling (CFP-39876)

Cloud scale = many large clusters, mixed topologies, and operators who must control what crosses a boundary.

Cluster Mesh: the control plane is shared

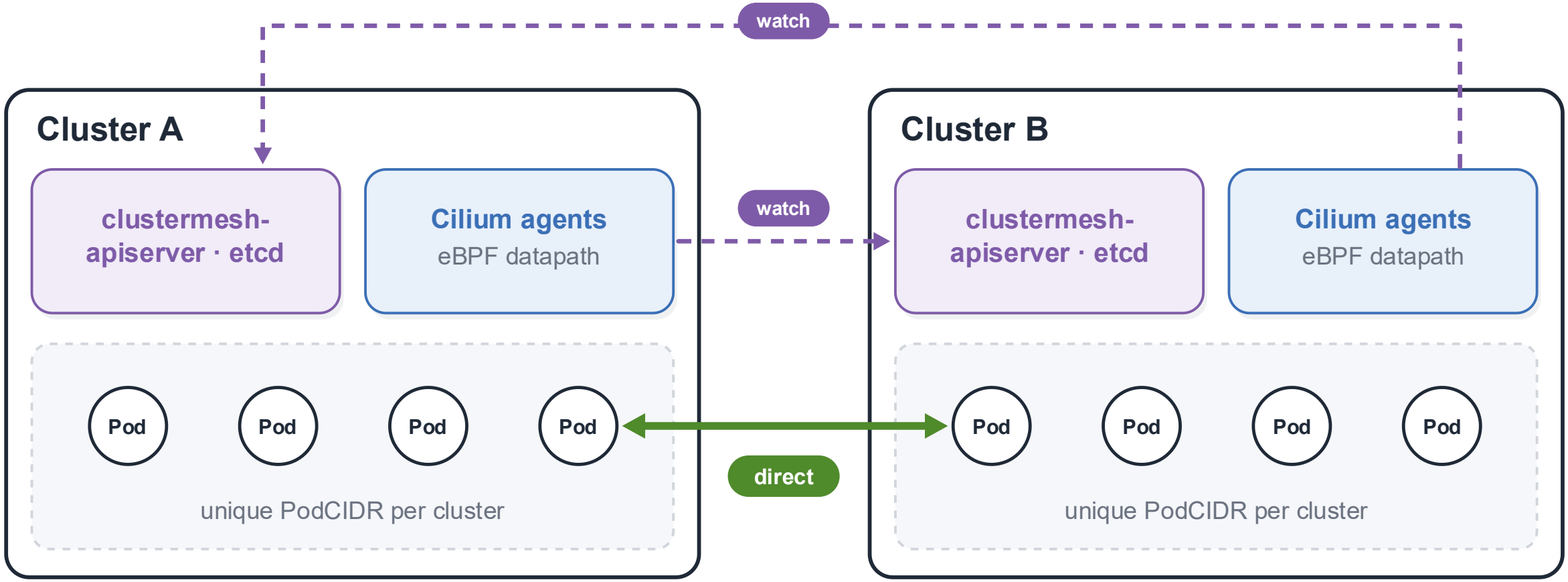


Every agent watches its peers: global Services, Identities, Endpoints / ipcache. Policy is identity-based and enforced across clusters.

PRIMER · 60 SECONDS

…and the data path stays direct

CiliumCon NORTH AMERICA



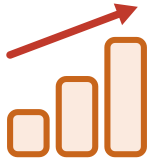
Requires

- Unique PodCIDRs
- Same datapath mode
- Shared CA
- Node IP reachability

THE PROBLEM

Three hard problems at cloud scale

CiliumCon NORTH AMERICA



Mesh scale

Every agent receives every endpoint and identity. Memory and churn grow with mesh size, not usage.

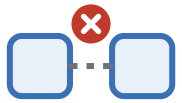
→ Layer 1 · Sharing



Boundary control

Platform teams need precise control over what crosses a cluster boundary. Shared-by-default is the wrong posture.

→ Layer 1 · Sharing



Mixed topology

Peered VNETs, overlay pod IPs, disjoint subnets. One unroutable path forces tunneling for all traffic.

→ Layer 2 · Data path

Two layers make it practical: **Sharing** — what crosses **Data path** — how it crosses



03 LAYER 1

Sharing

What crosses the boundary

The default shares everything

15 GiB

of memory per Cilium agent

200 clusters

300 nodes / cluster

15k endpoints / cluster

Cluster Mesh historically propagates all CiliumIdentities and CiliumEndpoints to every cluster.

Source: CFP-39876 performance profiling (also 500 identities / cluster, 0 services / cluster)

Every agent pays for the whole mesh



- Cost follows mesh size, not what workloads use
- Most workloads never talk cross-cluster
- Exposure is implicit: visible everywhere unless hidden

The goal

Make export opt-in and scoped.

Share exactly what platform teams choose — nothing crosses by accident.

TODAY

Shared by default

Anything in the mesh is visible everywhere unless explicitly hidden

GOAL

Shared by intent

Exposure becomes an explicit, reviewable platform decision

LAYER 1 · SHARING

Two knobs: per namespace, per service

CiliumCon NORTH AMERICA

Per namespace · global namespaces

cilium-config

```
# cilium-config
clustermesh-default-global-namespace: "false"
```

namespace.yaml

```
apiVersion: v1
kind: Namespace
metadata:
  name: payments
  annotations:
    clustermesh.cilium.io/global: "true"
```

Only endpoints and identities in global namespaces cross the mesh

Per service · MCS-API ServiceExport

serviceexport.yaml

```
apiVersion: multicluster.x-k8s.io/v1alpha1
kind: ServiceExport
metadata:
  name: checkout
  namespace: payments
```

service annotation

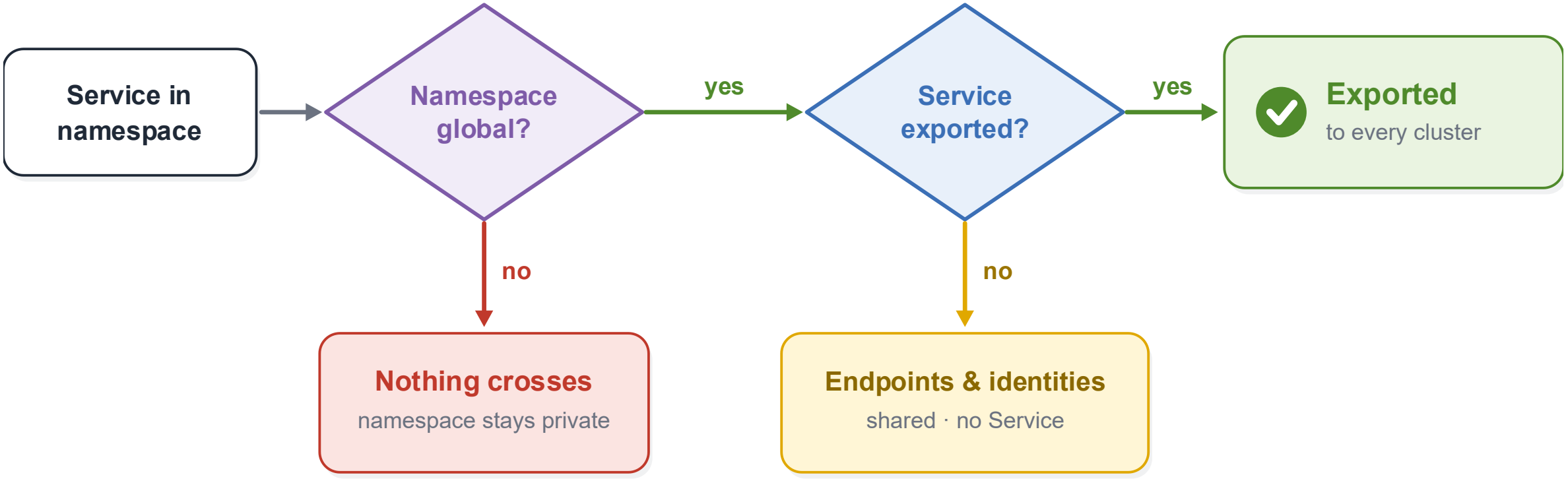
```
# or annotate the Service
service.cilium.io/global: "true"
```

Exports one Service to the clusterset; the global-service annotation works the same way

Default is true — existing meshes keep today's behavior

Sources: CFP-39876 (Cilium 1.19); cilium/cilium #42905, #43385

Two gates — both must open



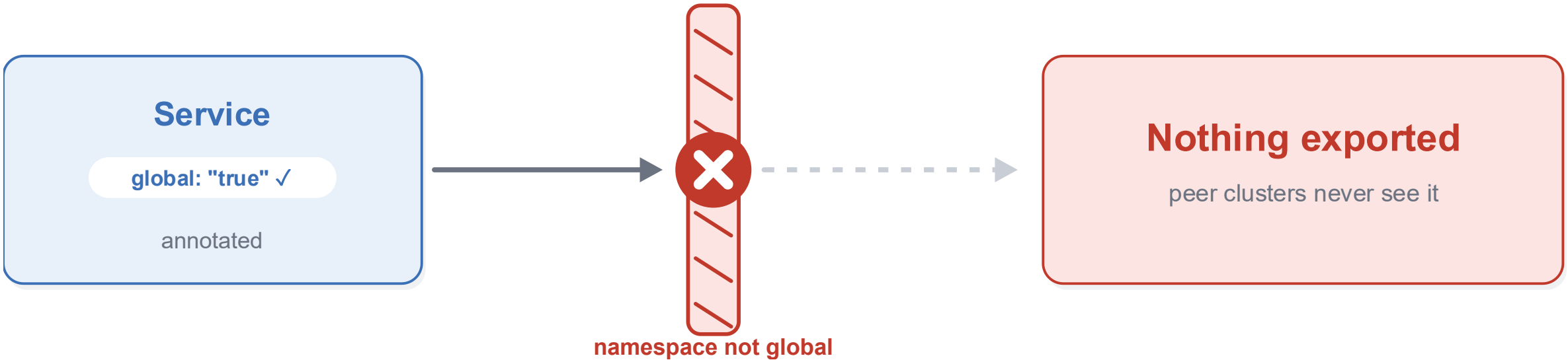
`default-global-namespace: true` or annotation `global: "true"`

`ServiceExport` or `service.cilium.io/global`

Once the default is off, namespace scope and service opt-in are both required.

Source: cilium/cilium #46090 (docs for global-namespace knobs)

The common gotcha

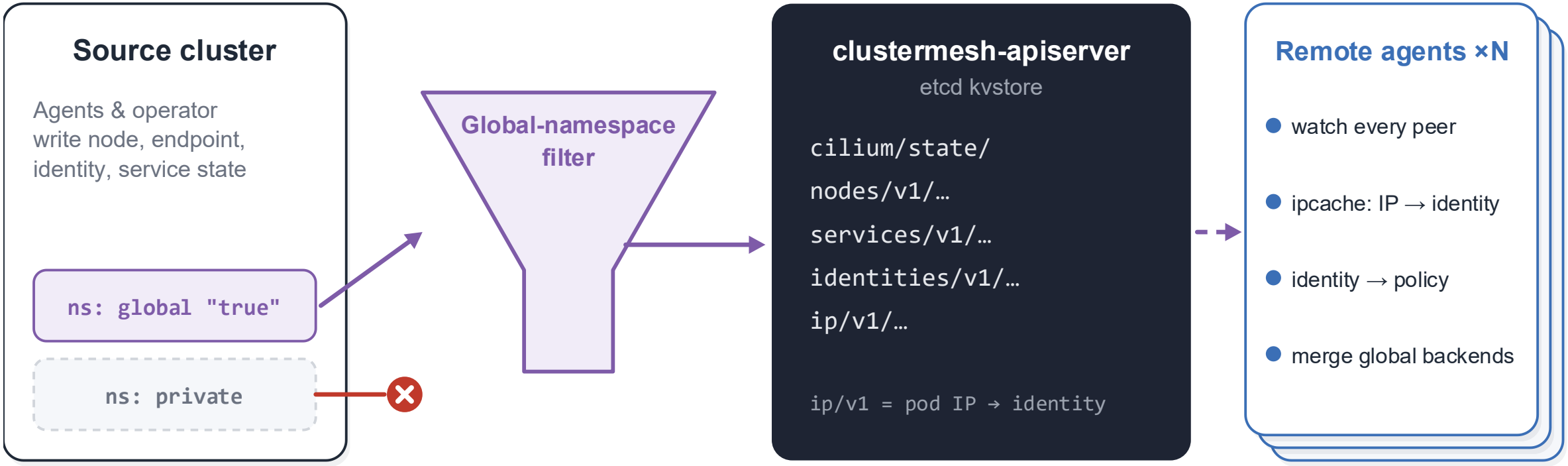


Default `false` + annotate only the Service = nothing exported

The same namespace gate applies to global services, MCS-API ServiceExports and network policy.

Source: cilium/cilium #46090

The filter sits at the source



Fan-in by design
N clusters → each agent holds state from N−1 peers

Cost driver
Memory and watch traffic scale with exported endpoints + identities

Filter at the source
Private namespaces never leave their cluster

Sources: CFP-39876; cilium/cilium #42905 (endpoints, identities), #43385 (services, MCS ServiceExports); Cilium docs (Cluster Mesh architecture, kvstore layout)

LAYER 1 · PAYOFF

Scale follows sharing

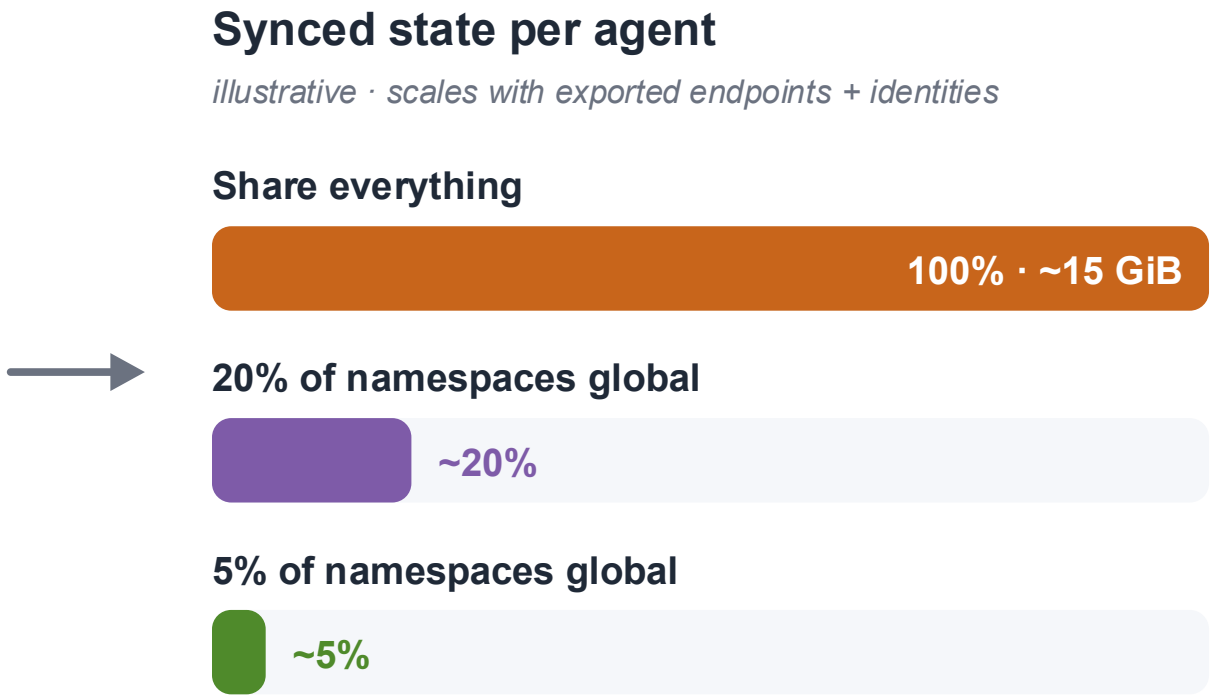
CiliumCon NORTH AMERICA

200
clusters

60,000
nodes · 300 / cluster

3M
endpoints · 15k / cluster

100k
identities · 500 / cluster



Each agent watches all 199 peers — adding a cluster grows state on every other node. Scope the export and state tracks what you share, not how many clusters you join.

Source: CFP-39876 profiling scenario (200 clusters × 300 nodes, 500 identities and 15k endpoints per cluster; ~15 GiB per agent). Bars assume linear scaling with exported endpoints/identities.

LAYER 1 · OPERATE

Roll out namespace by namespace

CiliumCon NORTH AMERICA

1

Inventory

cross-cluster consumers per namespace

2

Annotate

those namespaces global first

3

Flip the default

default-global-namespace: false

4

Watch it shrink

remote state, cluster by cluster

Verify what each cluster publishes

verify

```
$ cilium clustermesh status
$ cilium-dbg identity list
$ cilium-dbg service list
# remote IDs and backends only
# from global namespaces
```

Gotchas

- Default off: a global service needs a global namespace and the service annotation
- MCS-API needs `clustermesh.mcsapi.enabled` and CoreDNS ≥ 1.12.2

Sources: CFP-39876; cilium/cilium #42905, #43385, #46090; Cilium docs (MCS-API). Namespace and service names are examples.

LAYER 1 · TRADEOFFS

What scoped export buys — and what it costs

CiliumCon NORTH AMERICA

Gains

✓ Control-plane state scales with what is shared, not with mesh size

✓ Cross-cluster exposure is an explicit, reviewable decision

✓ One model across global services, MCS-API and policy

✓ Opt-in default preserves existing meshes

Costs & cautions

! Cross-cluster endpoint access and policy only apply to pods in global namespaces

! Plan namespace layout before turning the default off

! Flipping the default is a behavior change — roll out namespace by namespace

! Document the two knobs for app teams



04 LAYER 2

Data path

How it crosses

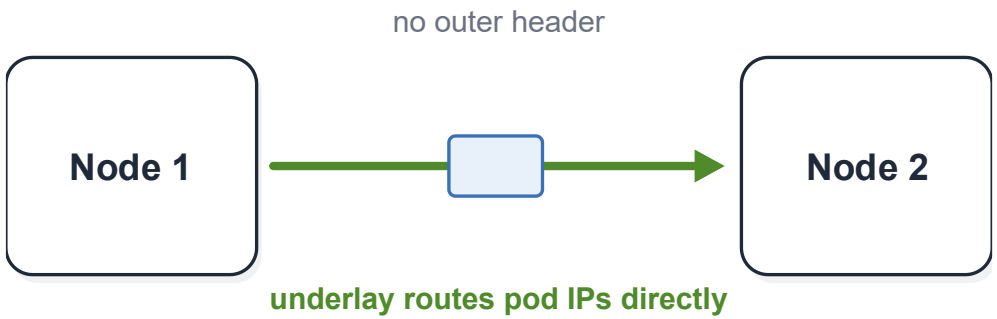
Two routing modes today — one per mesh

Encapsulation (tunnel)



NEEDS	Only node-to-node IP reachability
COSTS	Extra header, smaller MTU, encap/decap CPU
FITS	Unknown or disjoint topologies, cross-VPC/VNET

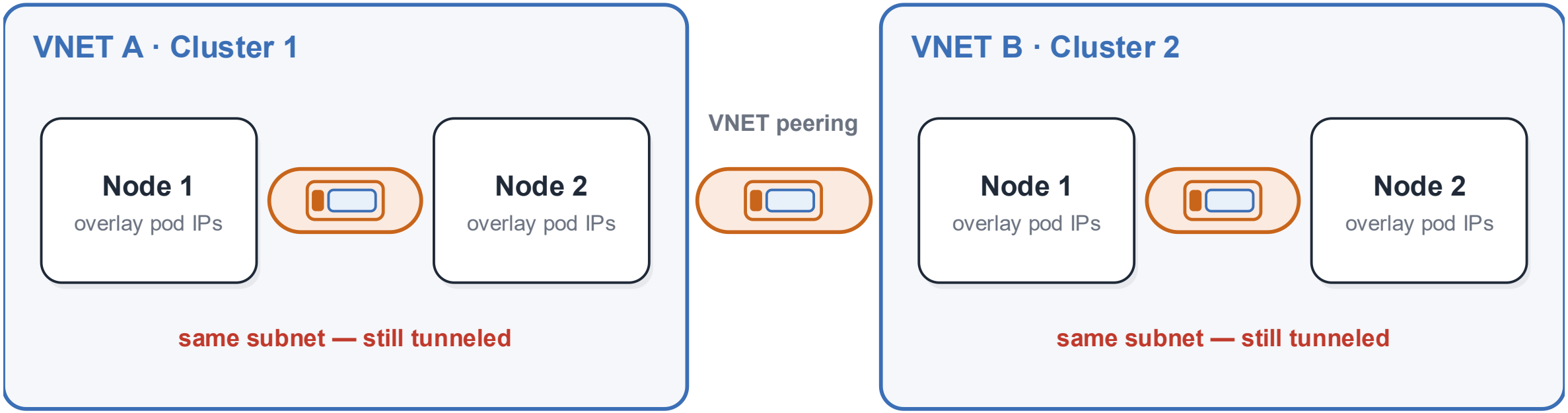
Native routing



NEEDS	Underlay knows PodCIDRs; native-routing CIDR covers every cluster
COSTS	No encapsulation overhead
FITS	Flat cloud networks with routable pod IPs

All clusters in a mesh must use the same datapath mode. Source: Cilium docs — Routing; Setting up Cluster Mesh

One unroutable path tunnels everything



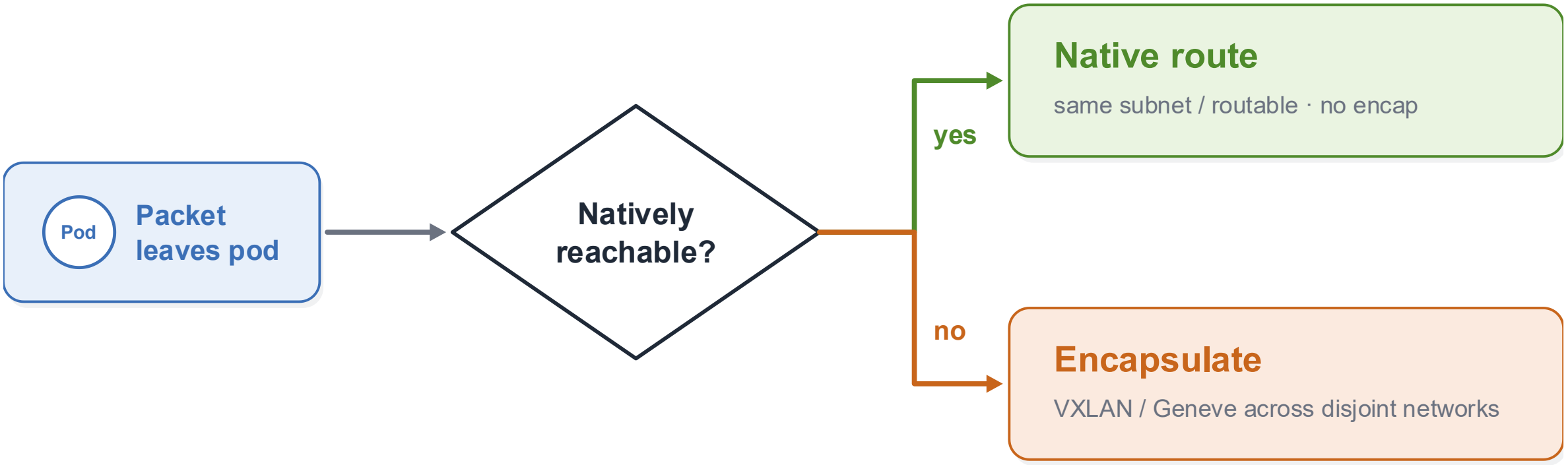
Scenario A Two clusters in peered VNETs, pods addressed from an overlay network

Scenario B One large cluster drawing pod addresses from two subnets with no direct connection

Result today: every packet is encapsulated — even when native routing would work

Hybrid routing: decide per packet, at runtime

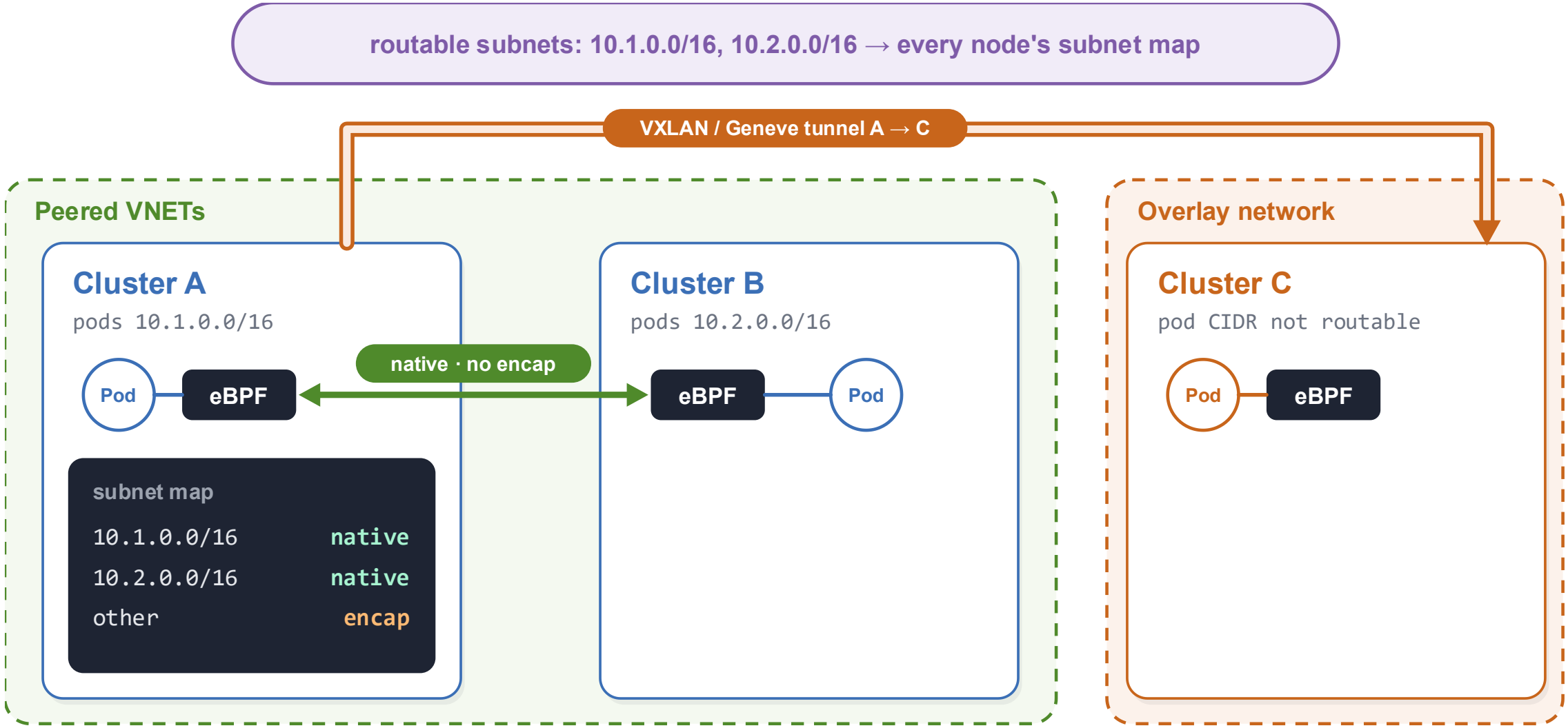
A third routing mode: native where the topology allows, encapsulate only where it must



Pay overhead only when needed: large meshes stop paying VXLAN/Geneve cost on flat paths

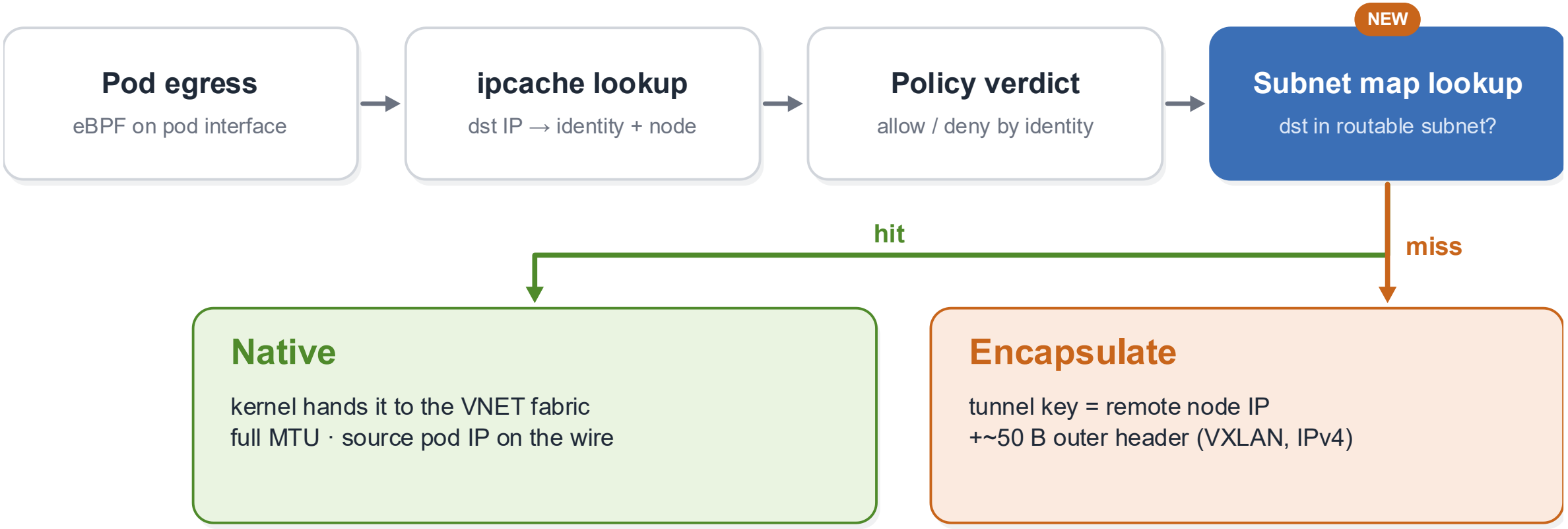
Design: CFP-32810 — Anubhab Majumdar (Microsoft), Vamsi Kalapala (Palantir), Krunal Jain (Microsoft) · status Implementable

One mesh, two paths



Same config, same map: traffic back to A also goes native. C isn't in A's map, so A's node wraps the packet; C's node unwraps it. CFP-32810 · datapath #41868 · control plane #43438 · core #45001 · CIDRs illustrative

One extra lookup picks the path



Map shape
Subnet → routing decision, in a BPF map the agent owns

MTU
Pods share one MTU — size it for the tunnel path

Encryption
WireGuard supported in hybrid mode

Sources: CFP-32810; cilium/cilium #41868 (datapath), pkg/maps/subnet; hybrid + WireGuard commit. Stage order simplified; VXLAN overhead is the standard IPv4 figure.

LAYER 2 · CONTROL PLANE

Config to BPF map — live, no restarts

CiliumCon NORTH AMERICA



Add a subnet

→ Agent upserts the entry

→ Next packet to it goes native

Remove a subnet

→ Entry deleted

→ Traffic falls back to encap

on a node · illustrative

```
$ cilium-dbg bpf subnet list
PREFIX          MODE
10.1.0.0/16     native
10.2.0.0/16     native
# not listed → encapsulate
```

Sources: CFP-32810; cilium/cilium #43438 (control plane), #45001 (core, merged), #45002 (subnet list), #45579 (routes), #47367 (delegated IPAM). Output illustrative.

LAYER 2 · OPERATE

Turn it on, then prove the path

CiliumCon NORTH AMERICA

1 · Enable

enable · example values

```
routingMode: hybrid
ipv4NativeRoutingCIDR: 10.0.0.0/8
tunnelProtocol: vxlan
# Helm support in progress;
# agent flags work today
```

2 · Pre-flight

- ☐ VNET peering routes the pod subnets you list
- ☐ UDP 8472 (VXLAN) or 6081 (Geneve) open to overlay clusters
- ☐ UDP 51871 if WireGuard is on
- ☐ MTU sized for the tunnel path

3 · Prove the path

prove the path

```
$ cilium-dbg status
$ cilium-dbg bpf subnet list
$ hubble observe \
    --to-ip 10.2.4.7
# native: no tunnel hop
```

What it saves

- ✓ No encap on flat paths: larger effective MTU, less CPU per packet
- ✓ Pod IPs visible to the fabric for routing and flow logs
- ✓ One mesh spans flat and overlay clusters without a gateway

Known edges

- ! Delegated-plugin IPAM: identical local router IP on every node breaks NodePort (tracked upstream)
- ! Route installation, delegated IPAM and Helm still landing on main

Sources: CFP-32810; cilium/cilium #45001, #45579, #47367, #47368, #49630. Values and IPs illustrative; check flag names againstthe release you run.

LAYER 2 · STATUS

Hybrid routing: from proposal to production

CiliumCon NORTH AMERICA

Tracked in cilium/cilium #41867 · core support landed in main

Jun 2025
CFP-32810
design discussion

Sep 2025
Tracking issue
#41867 opened

Nov 2025 – Jan 2026
Datapath + BPF
tests merged

Landed
Core hybrid routing
merged to main · e8bac79

REMAINING

Part 6
Route installation
#45579

Part 7
Delegated IPAM
#47367

Part 8
Helm support
#47368

WRAP-UP · TRADEOFFS

Pick per mesh — let topology, not habit, decide

CiliumCon NORTH AMERICA

	BEST WHEN	WATCH OUT FOR
DATA PATH		
Encapsulate everything	Underlay is unknown, disjoint or cross-cloud	Header overhead, reduced MTU, lower throughput
Native everywhere	Flat network; underlay routes every PodCIDR	Non-overlapping CIDRs and routes across all clusters
Hybrid routing	Flat within a network, disjoint across	Newer mode; remaining parts still landing
SHARING		
Scoped export	Large meshes, few cross-cluster services	Namespace planning; policy only in global namespaces
Share everything	Small meshes, broad east-west traffic	Agent memory and churn grow with every cluster

WRAP-UP · ≈5 MIN

Demo

Scoped export and hybrid routing on a two-cluster mesh

CiliumCon NORTH AMERICA

1

Mesh two clusters

Namespace not global → Service is not visible in the peer cluster

2

Opt in

Annotate the namespace and add a ServiceExport → Service imported and reachable

3

Watch the data path

Same-subnet traffic routes natively; cross-network traffic is encapsulated

WRAP-UP

Key takeaways

CiliumCon NORTH AMERICA



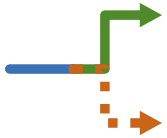
Share by intent

Make cross-cluster export opt-in: per service with ServiceExport, per namespace with global namespaces.



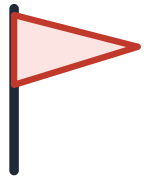
Scale follows sharing

Control-plane cost tracks what crosses the mesh — scope it and large meshes stay affordable.



Route by topology

Native routing on flat networks; encapsulate only where the network is disjoint.



Decide early

PodCIDRs, cluster IDs, namespace layout and datapath mode are hard to change later.

CiliumCon NORTH AMERICA

Thank you

Questions?

REFERENCES

CFP-32810 — Hybrid routing mode

CFP-39876 — ClusterMesh filtered (namespace-based) export

cilium/cilium #41867 — Hybrid routing tracking issue

cilium/cilium #43918 — Related implementation PR

Cilium docs — Cluster Mesh, MCS-API, Routing

Anubhab Majumdar
Microsoft
github.com/anubhabMajumdar

Vamsi Kalapala
Palantir
github.com/vakalapa

Q&A · 5 min