Two years of BGP-EVPN to the host

Integrating Kubernetes into IP Fabrics

Christopher Dziomba | 2023-11-21

About Me

🏠 Located in Bonn, Germany

Software & Hardware tinkerer at night

Passionate Skier

Hi, my name is Chris!
DevOps Engineer @ DTAG

you can find me on
GitHub @chdxd1 | LinkedIn /in/cdziomba

01
Kubernetes – A Networkers View

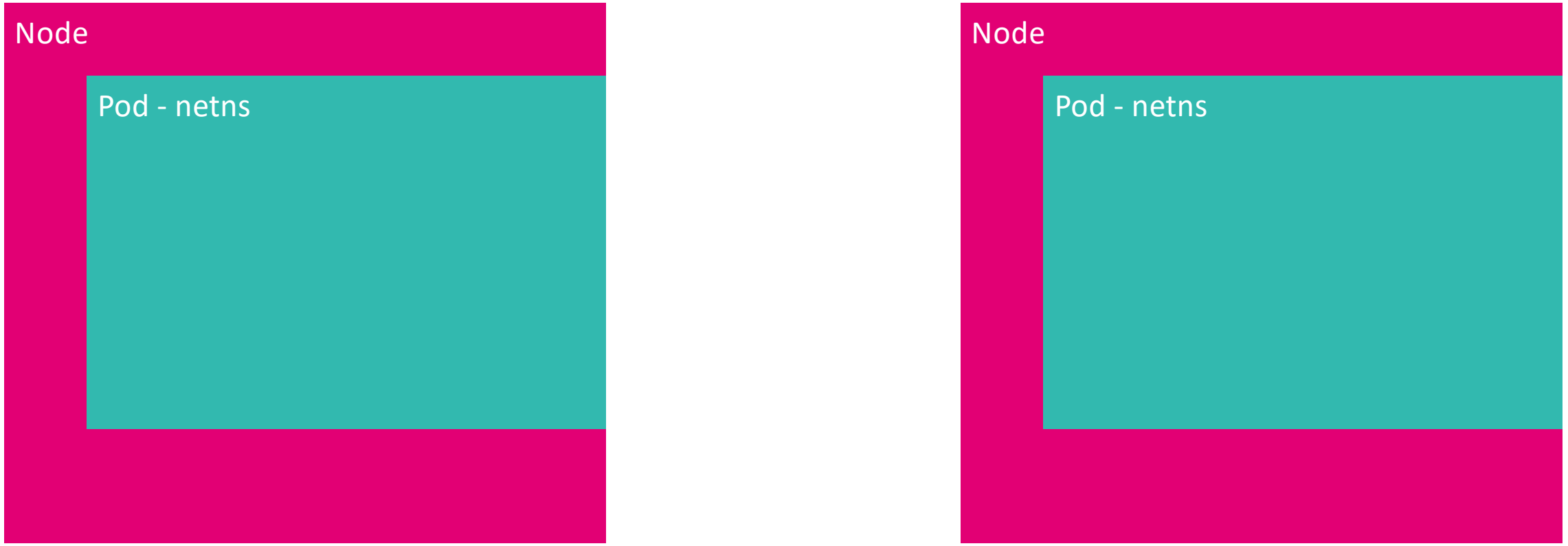
First: What is a Pod?

Linux Namespaces (e.g. mnt, pid, ipc, …)
allows separation / partitioning the Kernel

cgroups
limits / accounts resource usage (CPU, memory, disk I/O)

Network Namespaces
a Pod is a network namespaces with individual interfaces, netfilter rule set and so on

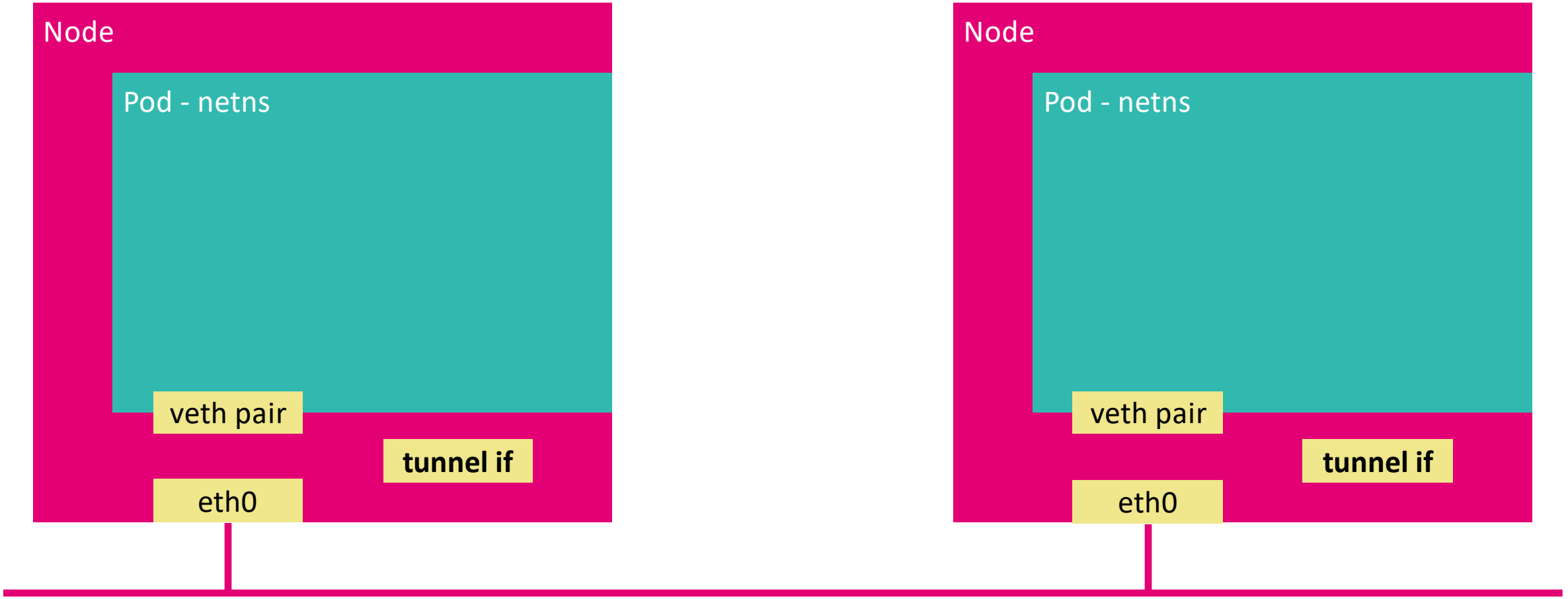
Empty Network Namespaces



Container Network Interface

- Specificiation for *pluggable* configuration of network interfaces for containers
- Implementations are also sometimes called CNIs in short (full: CNI plugins)
- Huge list to choose from depending on needs
- *Standard* / reference plugins
  - veth
  - macvlan/ipvlan
  - bridge
- Notable CNIs
  - Calico
  - Cilium
  - Coil
- Meta-plugins
  - Multus (*more on that later*)

CNI in Action



Second: What else?

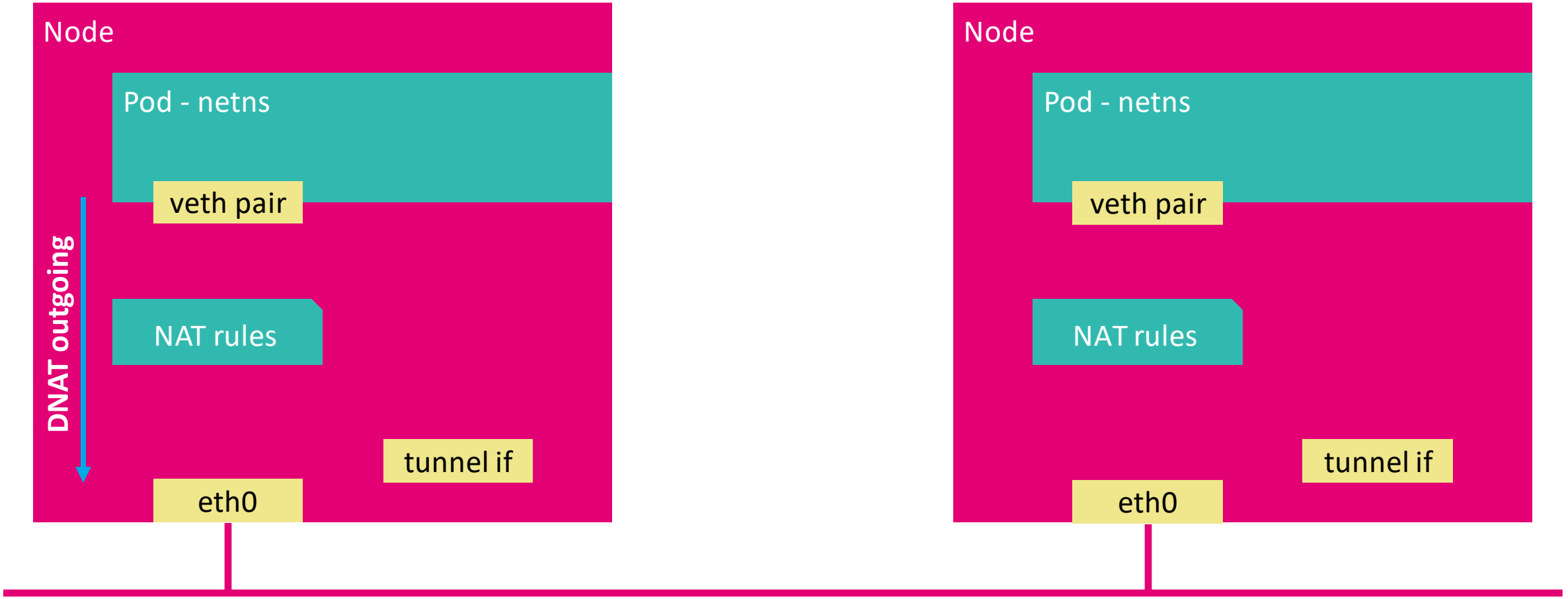
Services

Services have **endpoints**, *usually* Pods.

core-dns (K8s DNS service) can resolve them to e.g. **ClusterIPs** or be used for service discovery with **headless** services

kube-proxy programs (ip|nf)tables with NAT rules for load balancing from **ClusterIP** to service **endpoints**.

Services


Node
Pod - netns
veth pair
DNAT outgoing
NAT rules
tunnel if
eth0
Node
Pod - netns
veth pair
NAT rules
tunnel if
eth0

Second: What else?

Services

Services have **endpoints**, *usually* Pods.

core-dns (K8s DNS service) can resolve them to e.g. **ClusterIPs** or be used for service discovery with **headless** services

kube-proxy programs (ip|nf)tables with NAT rules for load balancing from **ClusterIP** to service **endpoints**.

LoadBalancer IP

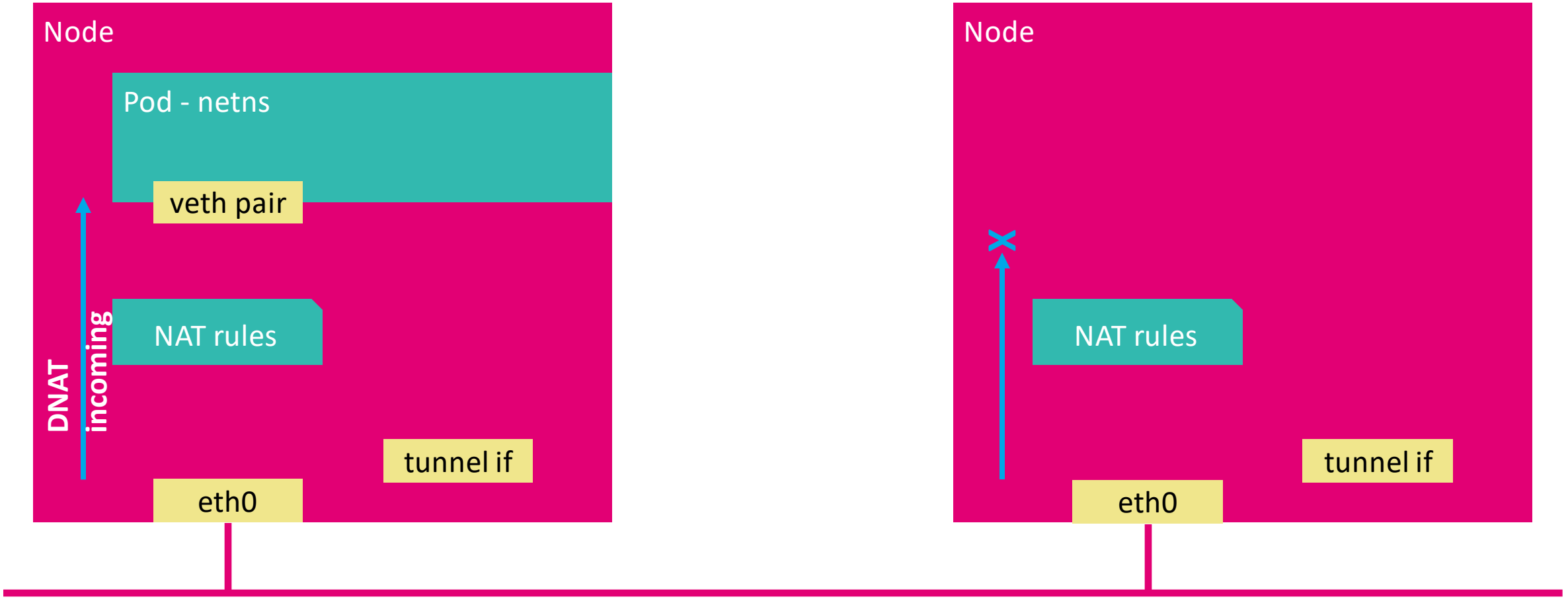
Services can have **LoadbalancerIPs**.

LoadBalancerIPs will also be written into NAT rules.

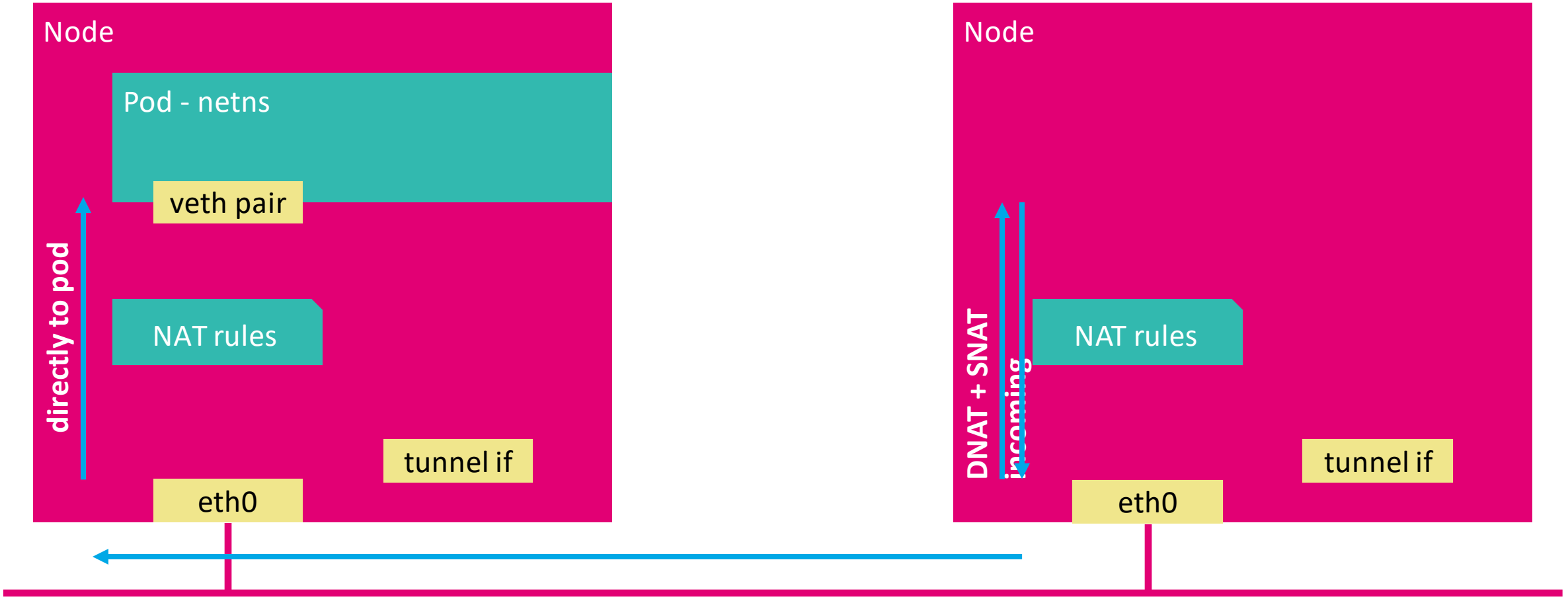
ExternalTrafficPolicy can be used to steer traffic:

- **Local** IP should only be advertised from nodes with **active** workload. Traffic only distributed on local node
- **Cluster** IP can be advertised from all nodes and traffic is forwarded to all nodes with **active** workload.

LoadBalancerIP (Policy: Local)



LoadBalancerIP (Policy: Cluster)



Second: What else?

Services

Services have **endpoints**, *usually* Pods.

core-dns (K8s DNS service) can resolve them to e.g. **ClusterIPs** or be used for service discovery with **headless** services

kube-proxy programs (ip|nf)tables with NAT rules for load balancing from **ClusterIP** to service **endpoints**.

LoadBalancer IP

Services can have **LoadbalancerIPs**.

LoadBalancerIPs will also be written into NAT rules.

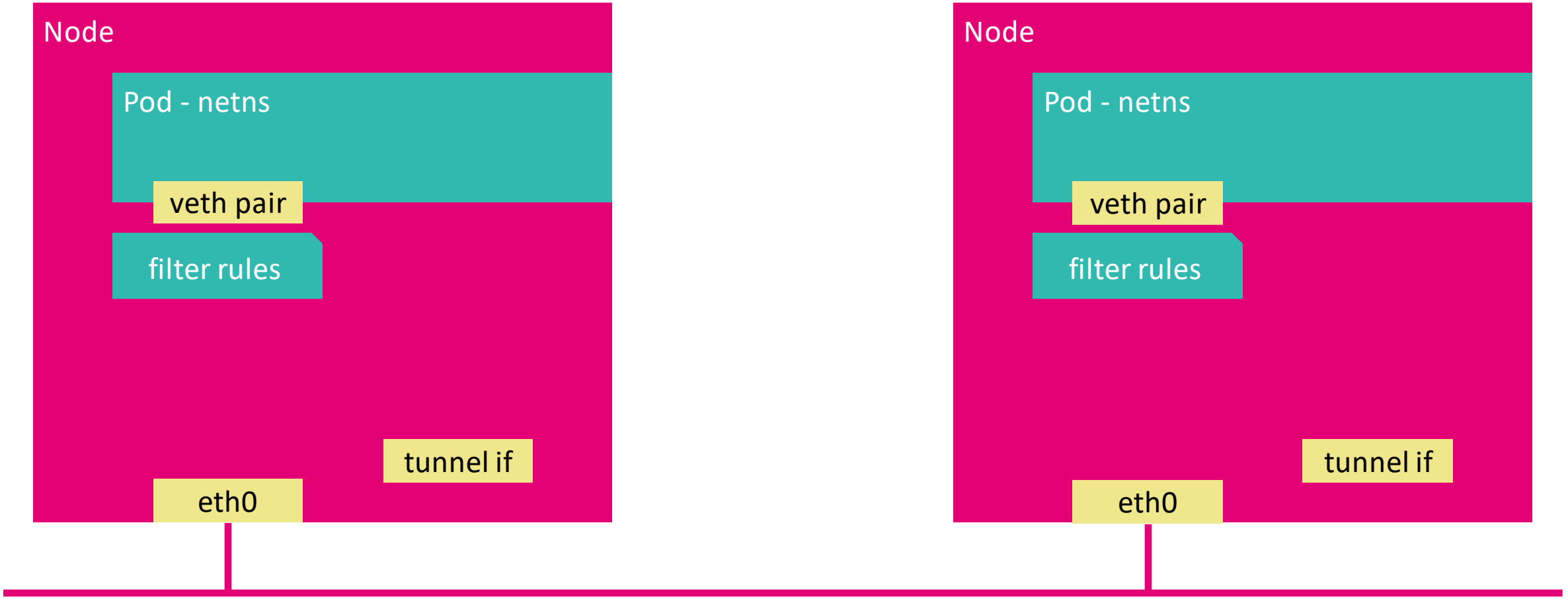
ExternalTrafficPolicy can be used to steer traffic:

- **Local** IP should only be advertised from nodes with **active** workload. Traffic only distributed on local node
- **Cluster** IP can be advertised from all nodes and traffic is forwarded to all nodes with **active** workload.

Network Policies

NetworkPolicies are used for firewalling workloads. The **CNI** will transform them into rules on the **veth** interface.

Network Policies



Network Requirements

Pod – Pod Reachability	Ingress from External	Egress to External
Intra-cluster traffic across nodes with service load balancing	Traffic needs to reach the cluster from external sources	Pods need to communicate with the outside world

A lot can be solved by using traditional technologies like **BGP** IPv4/IPv6 Unicast or even **BGP EVPN**.

Most CNIs can be used in **direct routing mode** instead of **tunneling**.

02

Das SCHIFF @ Deutsche Telekom

Why and How

“An internal, GitOps based Kubernetes Cluster as a Service Platform almost exclusively built using open source components.”

Managing Complexity

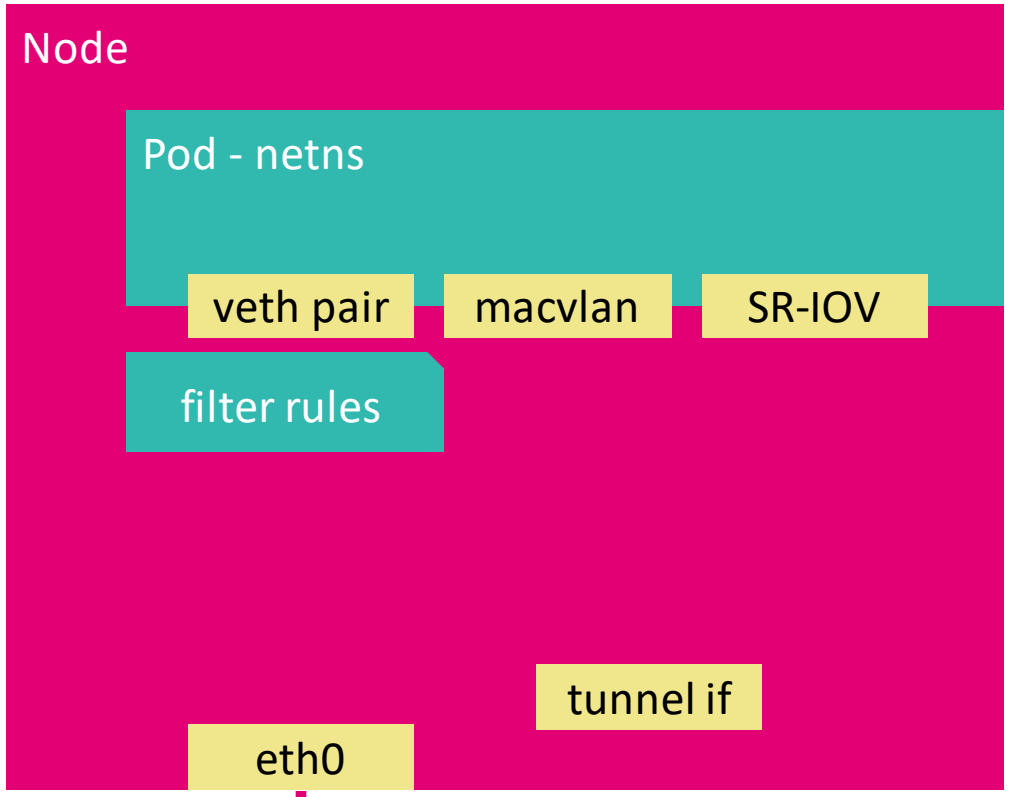
Our main customers are containerized network functions

Think about 5G Core, 5G Campus, IMS and various applications in that domain

- **Interconnection needed to multiple backbone VPNs/VRFs**
- **Complexity should not be centralized (in a multi-tenant fabric)**
- **Configurable by the consuming team**

Network Complexity

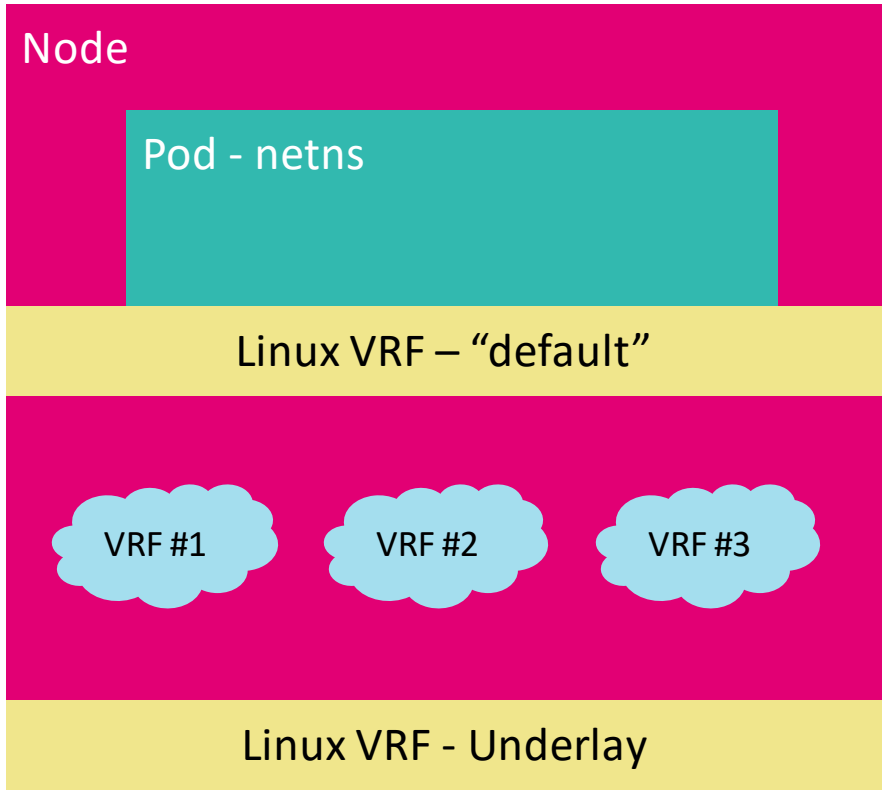
Additional Telco Requirements



So called **Containerized Network Functions (CNFs)** usually require **additional** networks using technologies like **macvlan** and **SR-IOV** (for fast-path).

- Requires usage of a “meta-CNI”
  - ”K8s Network Plumbing WG” Multus
  - Nokia’s DANM
  - orchestrate multiple sub CNIs for **multiple** interfaces

Our Setup



- Heavy users of Linux VRFs
  manage complexity on the individual node/tenant level
- Route-leaking from/to backbone VRFs
  using BGP peering instead of *import vrf*
- Anycast Gateway on Node
  providing Layer 2 /VLAN-like services for additional pod interfaces
- Configurable on the fly using Kubernetes ressources
  telekom/das-schiff-network-operator
- BGP-EVPN
  using BGP-unnumbered (RFC 5549) & MP-BGP **or** iBGP RR for EVPN

03
Summary

Caveats

SR-IOV

- Bypasses Linux Kernel
- Can't be de/encapsulated using the Kernel
  - VTEP on fabric
  - VTEP on NIC (using rte_flow)

Linux Kernel

- VRF route-leaking for local endpoints a pain
  - Without *<prot>_l3mdev_accept*
  - Use **veth** pairs between VRFs and **peer** using BGP
- **One** netns = **One** netfilter ruleset
  - kube-proxy / CNI rules already acting on traffic **in VRF**
  - Use eBPF for performance + bypass of NF
  - Alternative: Router in separate netns

FRR

- Various bugs in recent versions
  - Loosing EVPN remote-MACs on session flaps (#10298, #12391)
  - Loosing nexthop / neighbor group entries on fast interface flaps (#14481 etc)
  - MAC Mobility sequence numbering (#10468)
- Online re-configuration using hacky frr-reload.py
- Use **only** if you're willing to spend time on upstream

Summary

- We are running BGP-EVPN to the host for >2 years now
- Humble beginning for design (mainly Linux VRF related)
- FRR issues beginning of 2023 (especially during link flaps / fabric updates)

- Allows for a completely Layer2 free fabric design
- except SR-IOV (as of today)

- Future
- Move stack to DPU (“data processing units”) to offload and separate network from Kubernetes host

Questions?