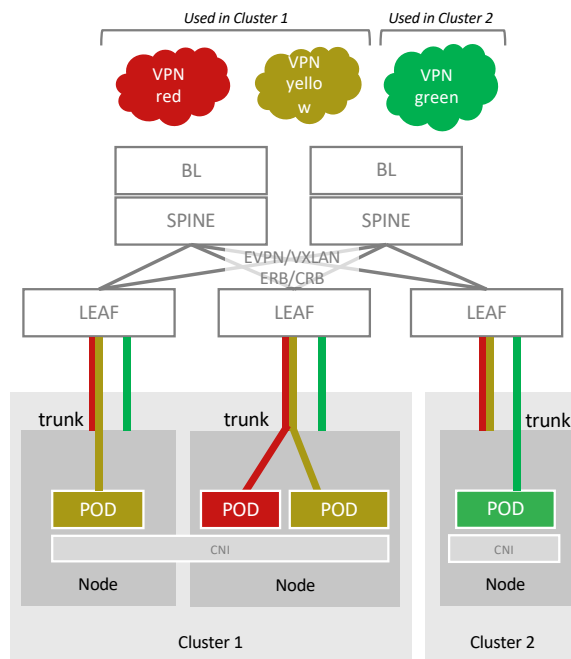


AUTOMATING NETWORK CONNECTIVITY WITH DT HOST BASED ROUTER

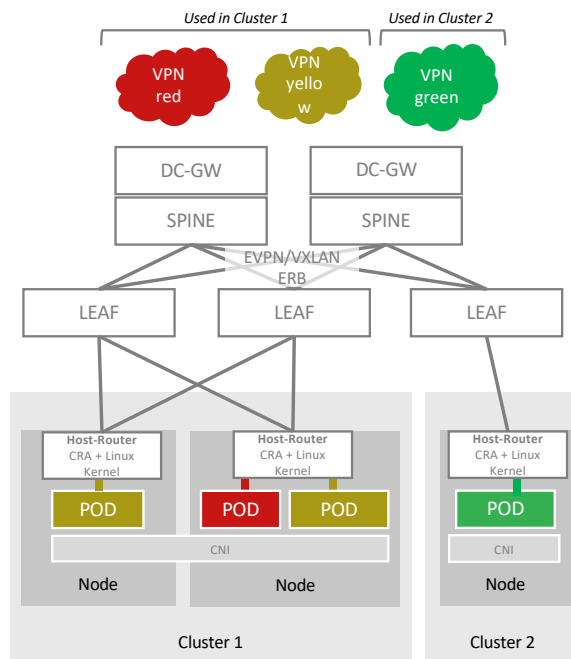
Robert Petračić DTCS

TRADITIONAL EVPN FABRIC DESIGN FOR CNF



- Leaf to host connections are L2-based (VLAN trunk ports)
- No native K8s mgmt tools that support network to K8s cluster coordination
- Most of connectivity complexity pushed to the DC fabric
- Frequent network device reconfigurations - for every tenant network change like adding new VLAN, VRF etc.

HOST BASED ROUTING DESIGN



- DT has developed HBR solution to mitigate problems of L2 design for “Das Schiff” DT CaaS solution
- Leaf to host connections become L3-based and host becomes part of EVPN control and data plane
- Containerized Routing Agent: swappable component from vendor or opensource (FRR)
- Network-operator is responsible for the configuration and monitoring of the host router according to the specified declarative state.
- No need for LEAF config change for tenant network changes

INSIDE THE HOST

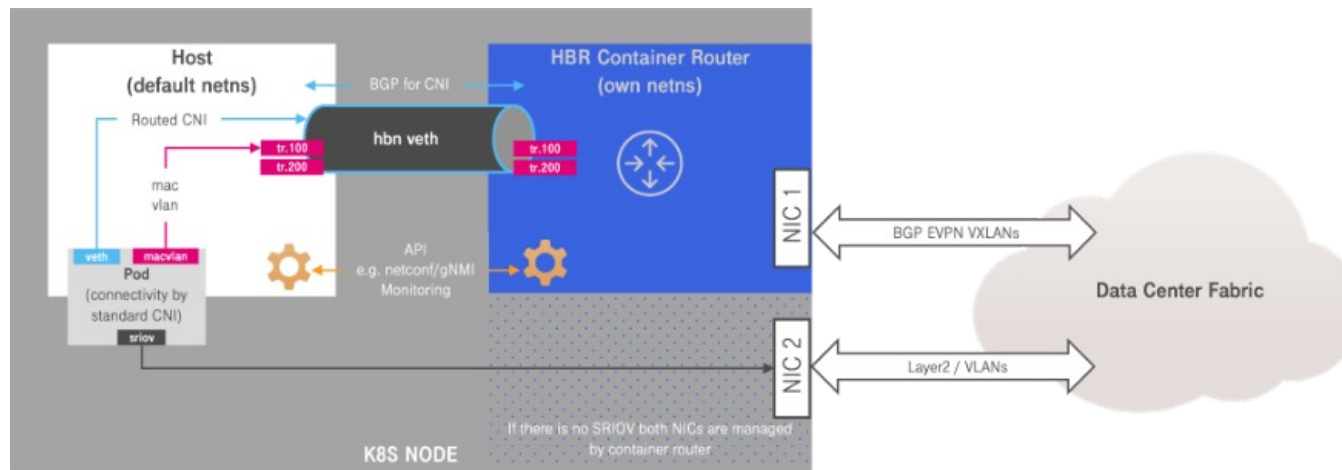
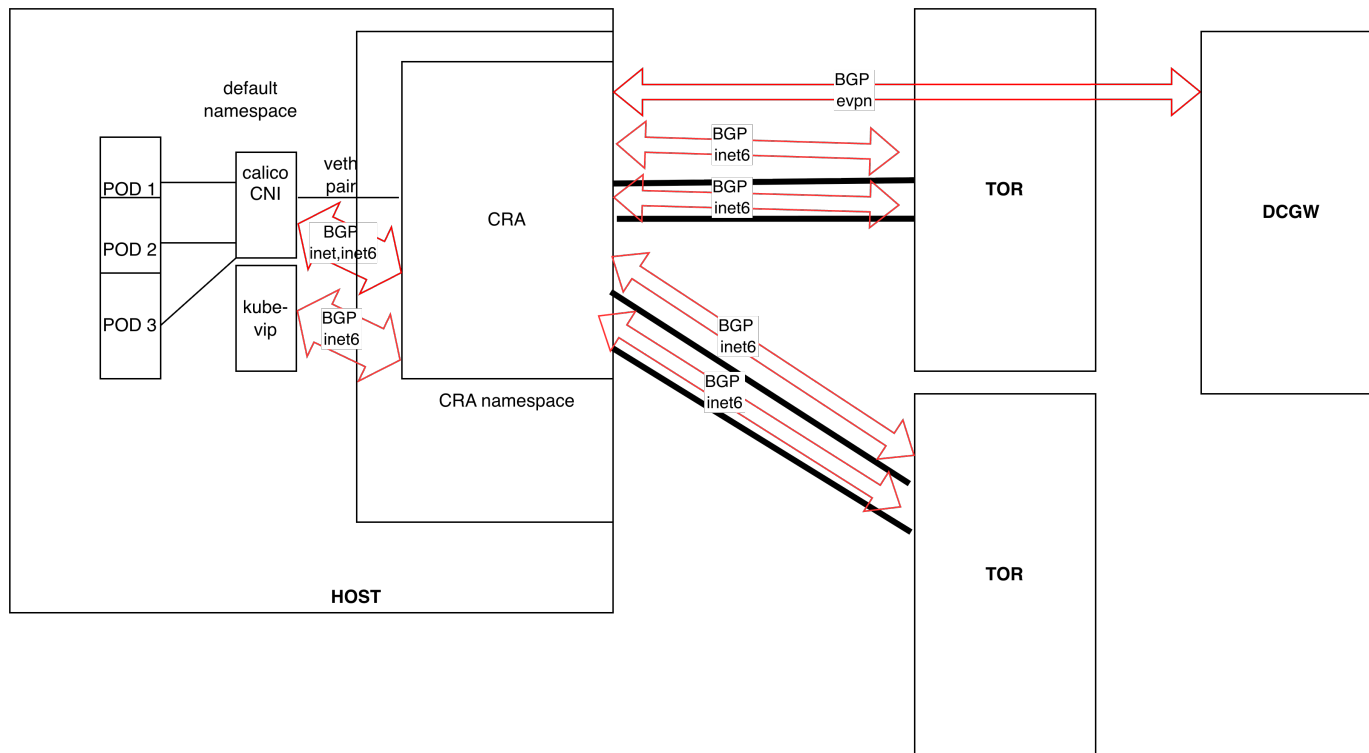


Image from DT HBR solution design

- Host router is managed natively by network operator
- NWOP gets desired network config from K8s custom resources
- In DT chosen HBR is a 6WIND VSR (first version had FRR as host service)
- Data plane is Linux kernel (netlink) layer with BGP control plane

BGP CONTROL PLANE INTEGRATION



NETWORK OPERATOR LEGACY CRDS

network.t-caas.telekom.com/v1alpha1 API group is the original, low-level interface to the network operator.

Users must understand the full low-level config model (VNI, route targets, prefix lists, etc.) to configure even simple use cases.

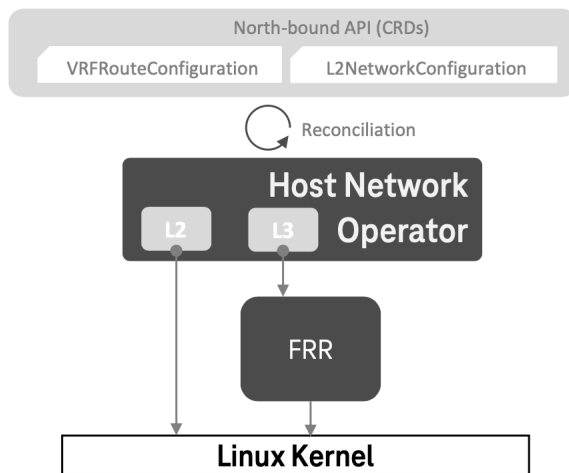


Image from Sylva HBR POC document

Example:

```
apiVersion: network.t-caas.telekom.com/v1alpha1
kind: VRRRouteConfiguration
metadata:
  name: example-vrf
spec:
  vrf: example
  vni: 100100
  routeTarget: "64512:100100"
  seq: 10
  import:
    - cidr: 10.0.0.0/8
      action: permit
  export:
    - cidr: 192.168.10.0/24
      action: permit
      le: 32
  communities:
    - "64512:100"
  nodeSelector:
    matchLabels:
      node-role.kubernetes.io/worker: ""
```

NEW INTENT BASED OPERATOR

Intent based concept uses higher abstraction level which means never configuring FRR, netlink, VXLAN or MetalLB directly as it hides low level details.

Config intent is declared via Kubernetes custom resources in the API group network-connector.sylva-project.org/v1alpha1.

The operator does the following:

- **Resolves** the references between used resources (like networkRef, vrfRef, label selectors).
- **Derives** the concrete per-node configuration.
- **Rolls it out** node by node as NodeNetworkConfig and NodeNetplanConfig resources, which the node agents apply.

INTENT BASED CUSTOM RESOURCES

The intent model raises the abstraction level in a following way:

- A small **foundation** (VRF, Network) is the contract boundary with infrastructure provisioning.
- A **routing layer** (Destination) names reachable prefixes.
- A **usage layer** (Inbound, Outbound, Layer2Attachment, PodNetwork, BGP peering, traffic mirroring) expresses workload needs by *reference*, not by duplicating fabric detail.

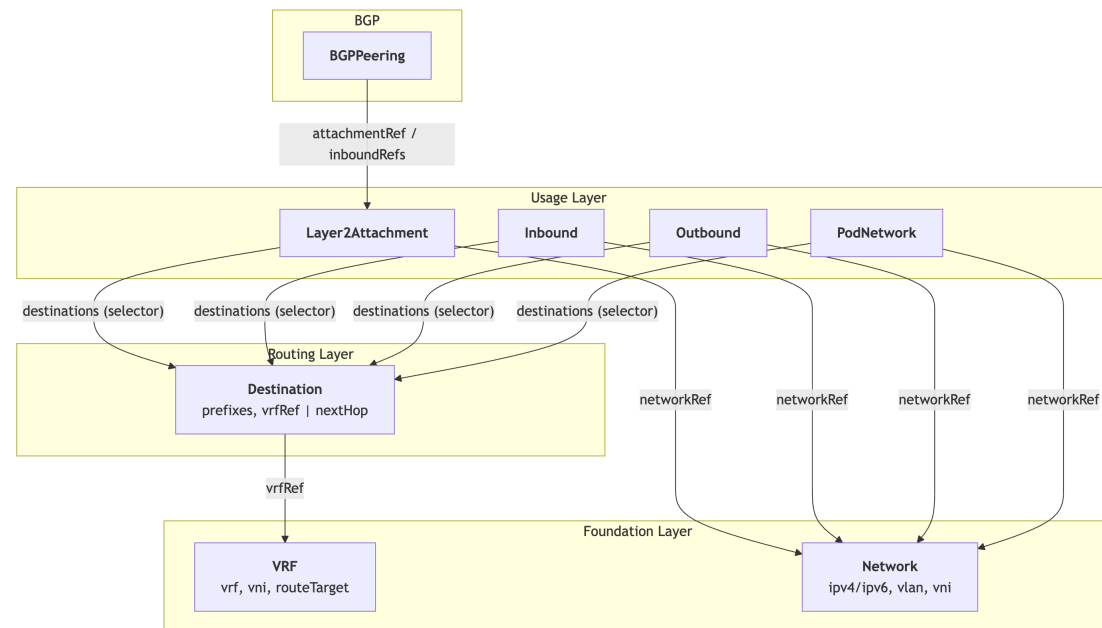


Image from <https://github.com/telekom/das-schiff-network-operator/blob/main/docs/getting-started/concepts.md>

SYLVA CONTRIBUTION

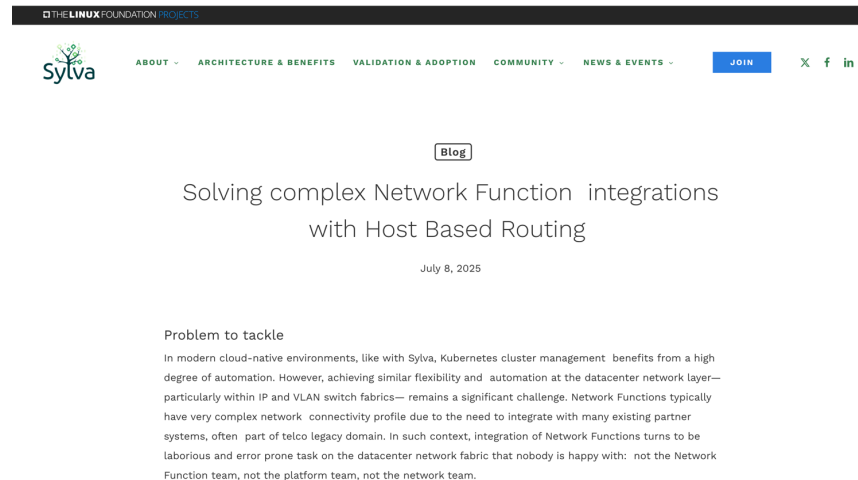


Image of <https://sylvaproject.org/solving-complex-network-function-integrations-with-host-based-routing/>

- Sylva is project to design standard Telco cloud, part of Linux foundation
- Deutsche Telekom as the founding member of Sylva intends to move the HBR development activities to Sylva and put the code on disposal under **Apache 2.0 license**, fostering open collaboration and adoption.
- Code already at: <https://github.com/telekom/das-schiff-network-operator>

HOST BASED ROUTING

Q&A

PACKET FLOW

