Sharing OVN
among Kubernetes Clusters

Hareesh Puthalath & Alin Serdean

Agenda

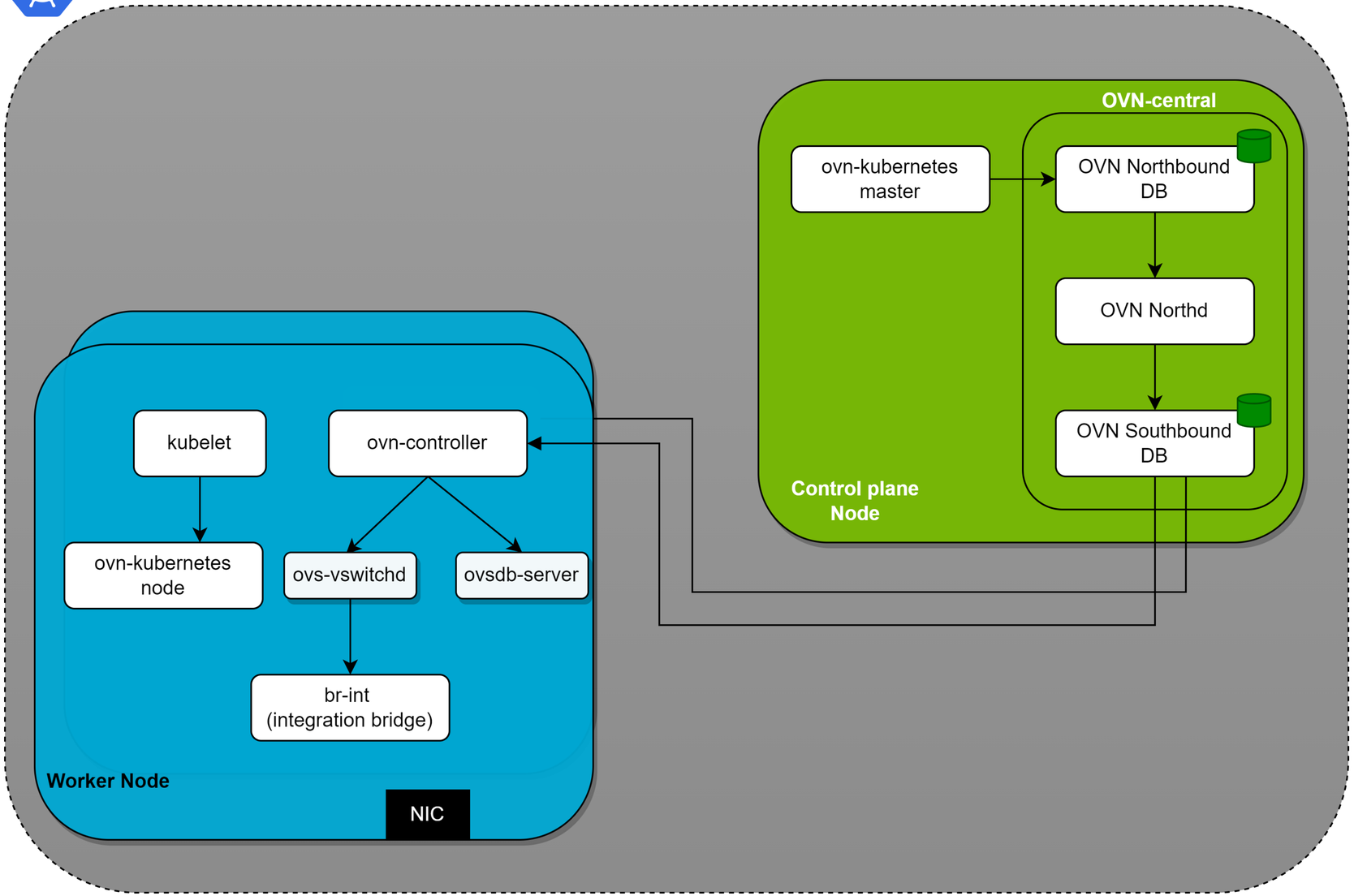
- ovn-kubernetes overview
- Sharing OVN among kubernetes clusters
- Supporting workloads in DPU
- Shared OVN with DPU
- Demo

ovn-kubernetes

- Container network interface (CNI)
- Kube-proxy replacement

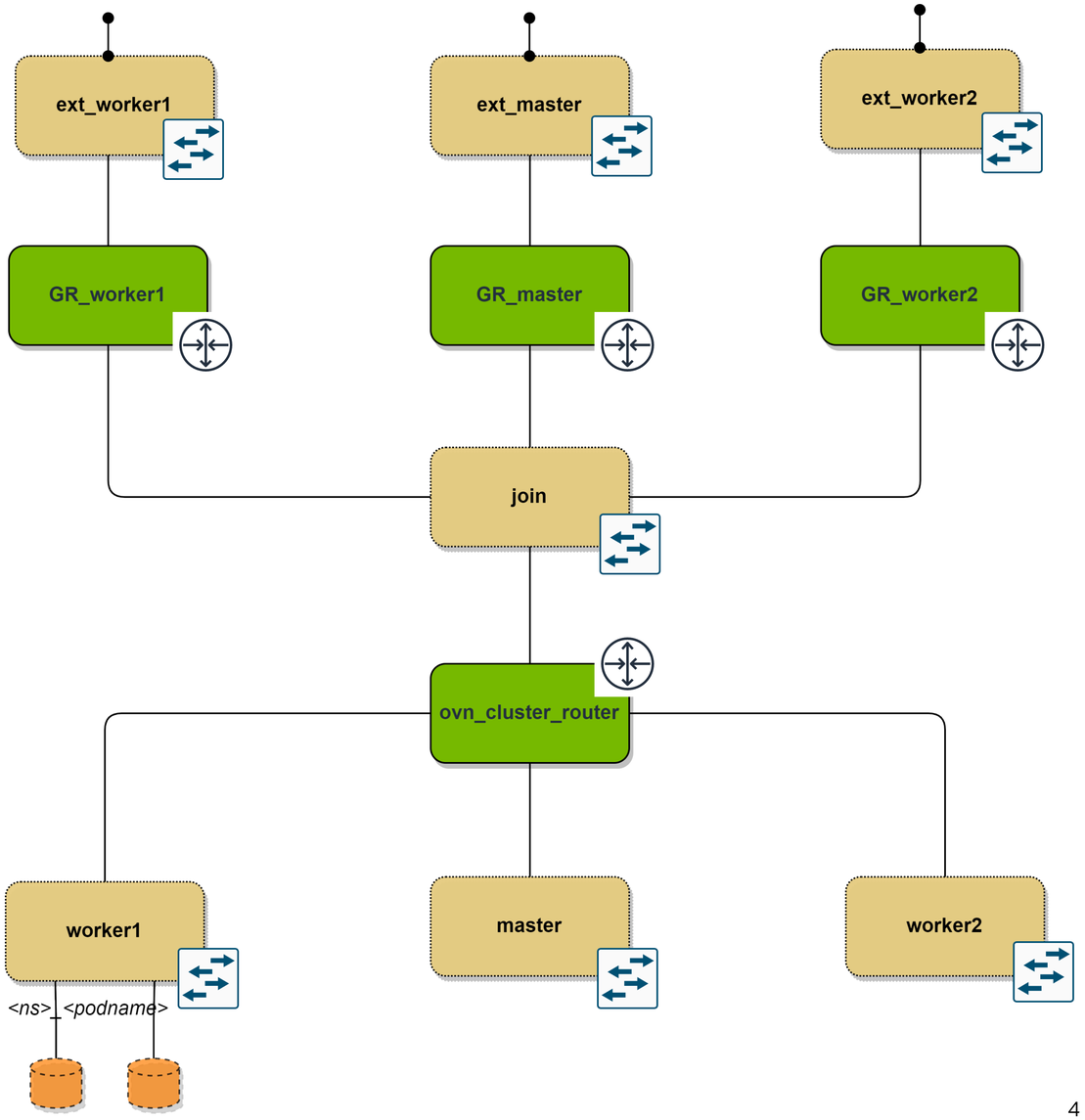
- SRIOV Device plugin
- Multus

- Control-plane Node
  - ovn-kubernetes master
  - OVN DBs
    - OVN Northbound DB
    - OVN Southbound DB
    - OVN-northd
- Worker node
  - ovn-kubernetes node
  - ovn-controller
  - ovs components



Logical Topology

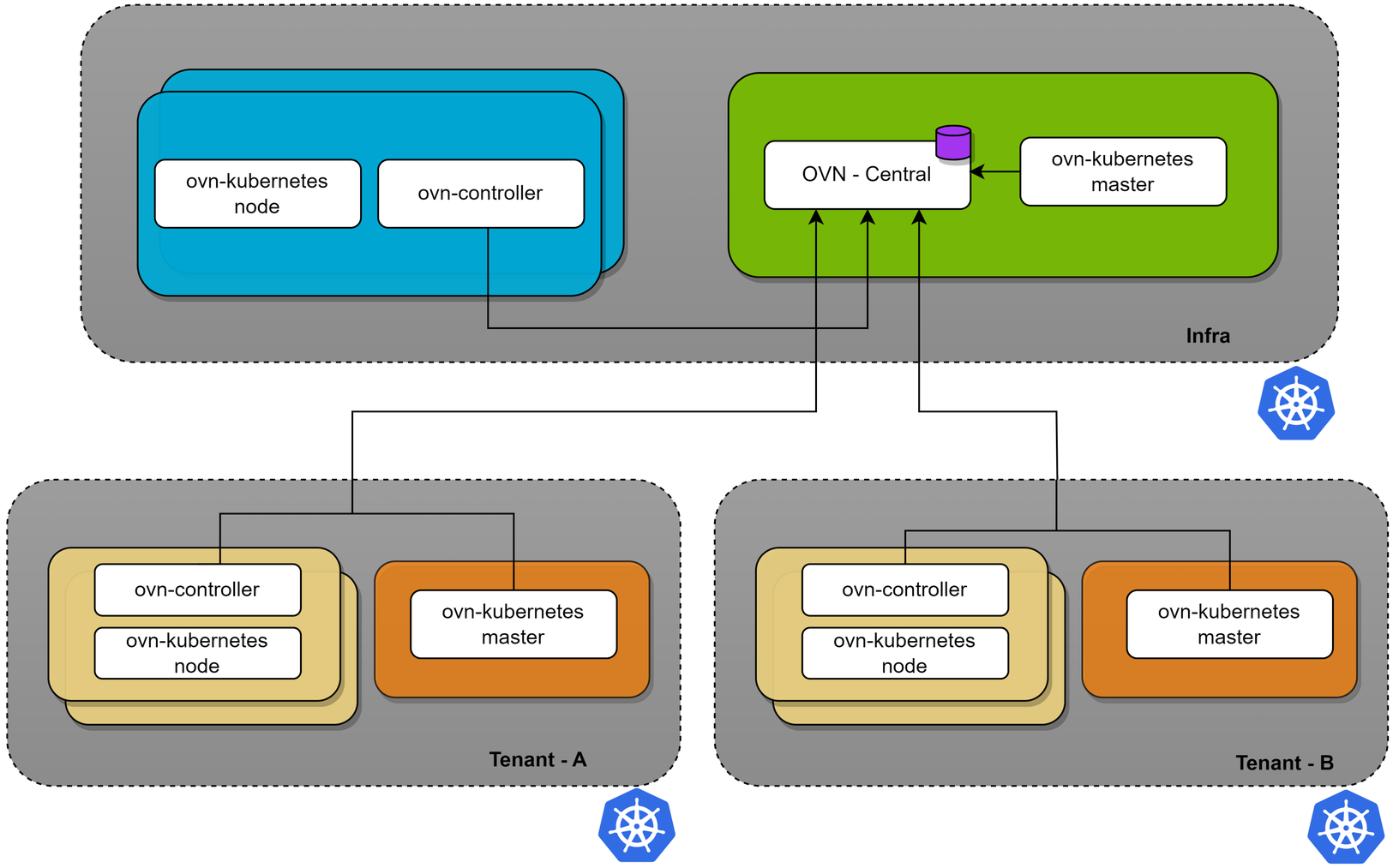
- ovn-kubernetes master builds a logical topology in OVN for the Kubernetes cluster
- Consists of
  - Cluster router
  - Per Node Logical switches
  - Per Node GW routers (for external connectivity)
  - Join switch
  - Load Balancers
    - For services
  - ACLs
    - For network policies



Sharing OVN among Kubernetes clusters

- Share ovn-central components across multiple Kubernetes clusters.

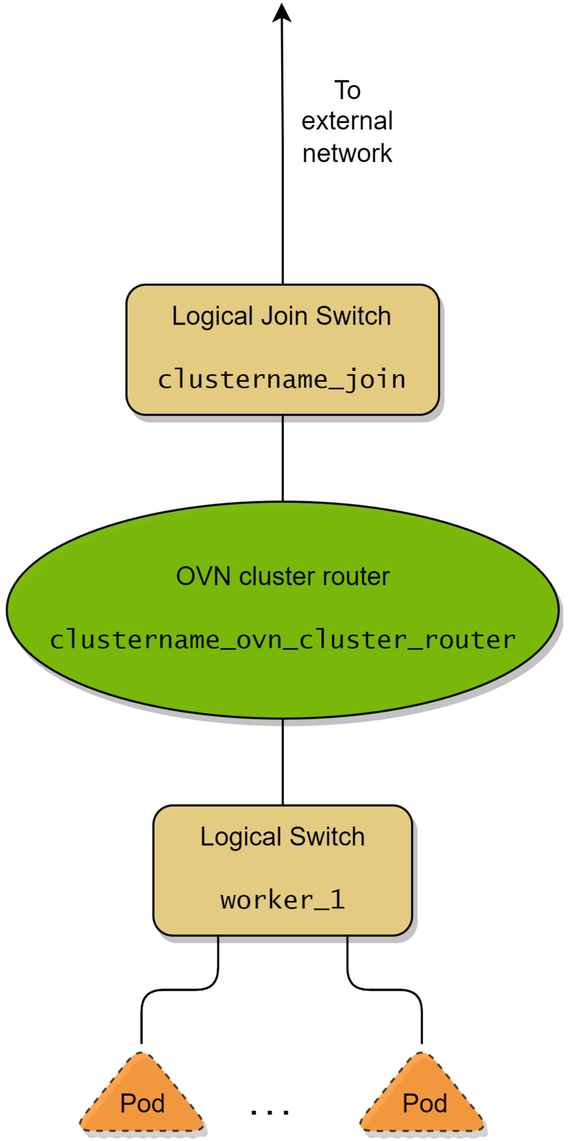
- Use cases
  - Managed k8s clusters for tenants
  - Connectivity



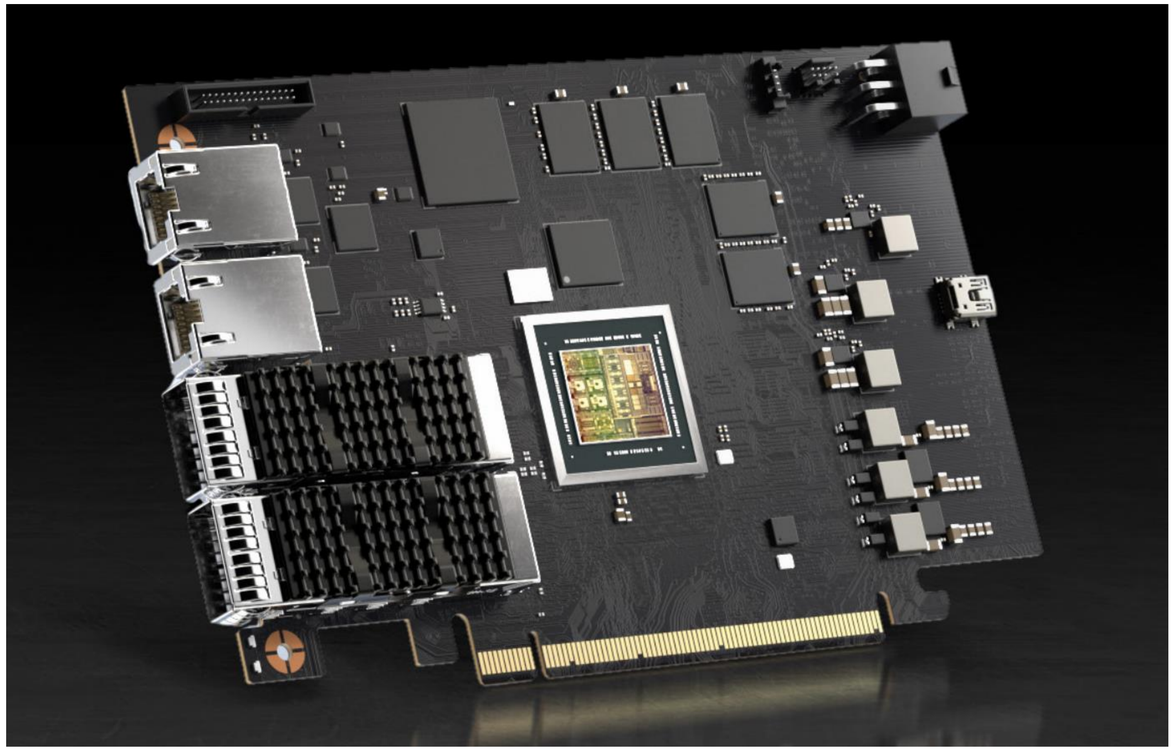
Changes

- Common logical objects:
  - ovn_cluster_router router
  - join switch
- Notion of *cluster_name* (KEP 1645)
- Generate logical element names with *cluster_name* prefix.
- Isolation
  - Shared DB => all clusters can see each others' elements.
  - Logical object lifecycle management should be cluster specific.
- Marking
  - Via external_ids for all Logical elements (switches and routers, etc.)
  - Logical elements have the *cluster_name* as an additional external_id

    `external_ids : { cluster_name=<<value>>, ……}`
- Filtering
  - DB queries use the cluster_name=<value> search predicate when *cluster_name* is present
- Stale object management

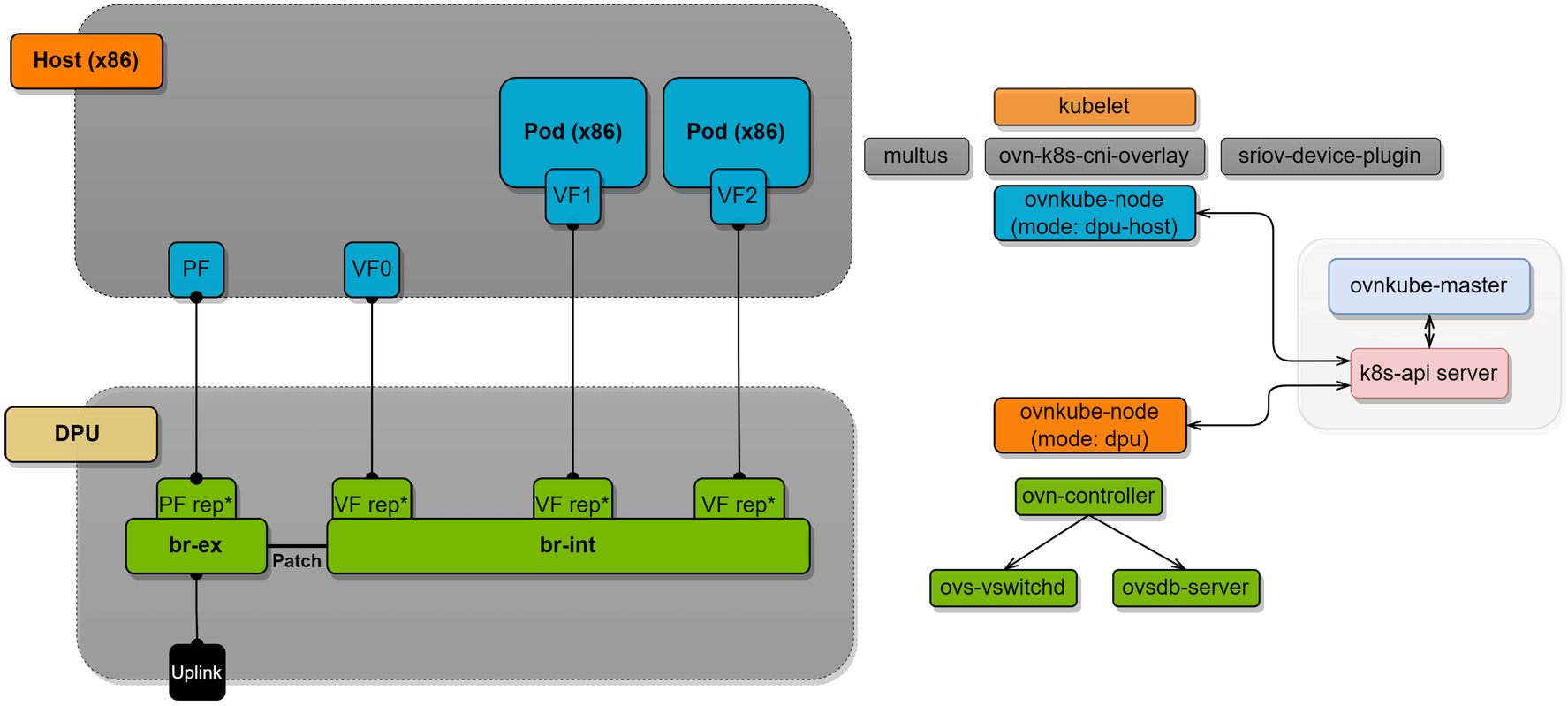


Shared OVN
with DPU offloading



DPU Offload Model

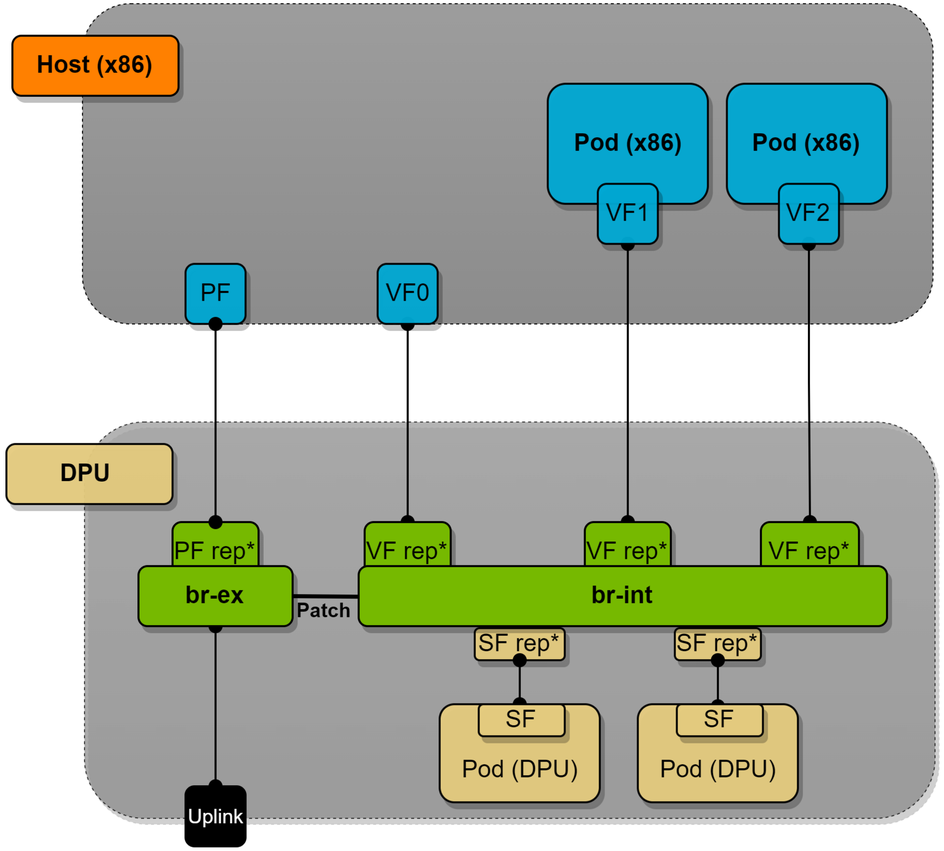
- SRIOV VF
- Kernel switchdev model
- OVS and OVN-controller runs in the DPU.
- ovn-kubernetes node modes
  - dpu-host mode
  - dpu mode



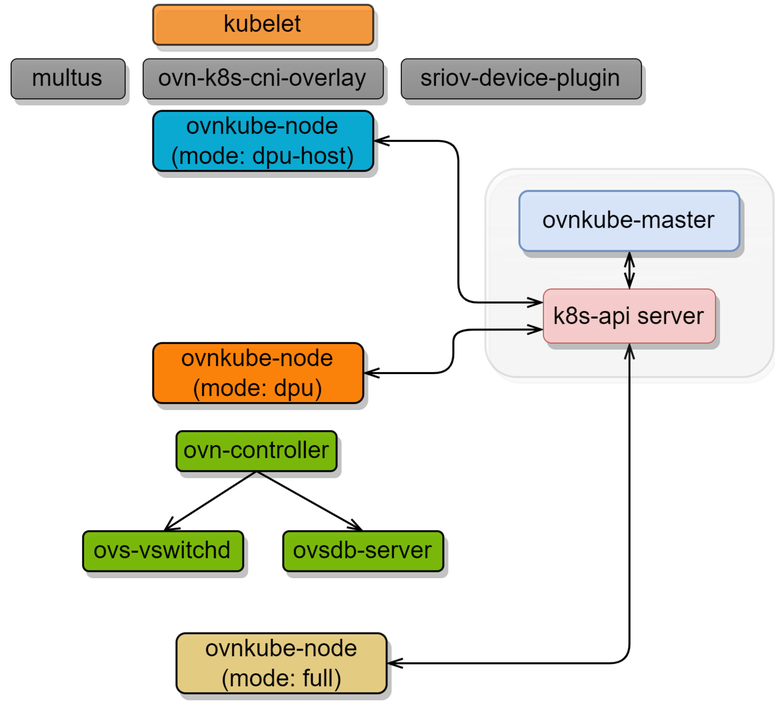
*rep = Representor

DPU Workload Support

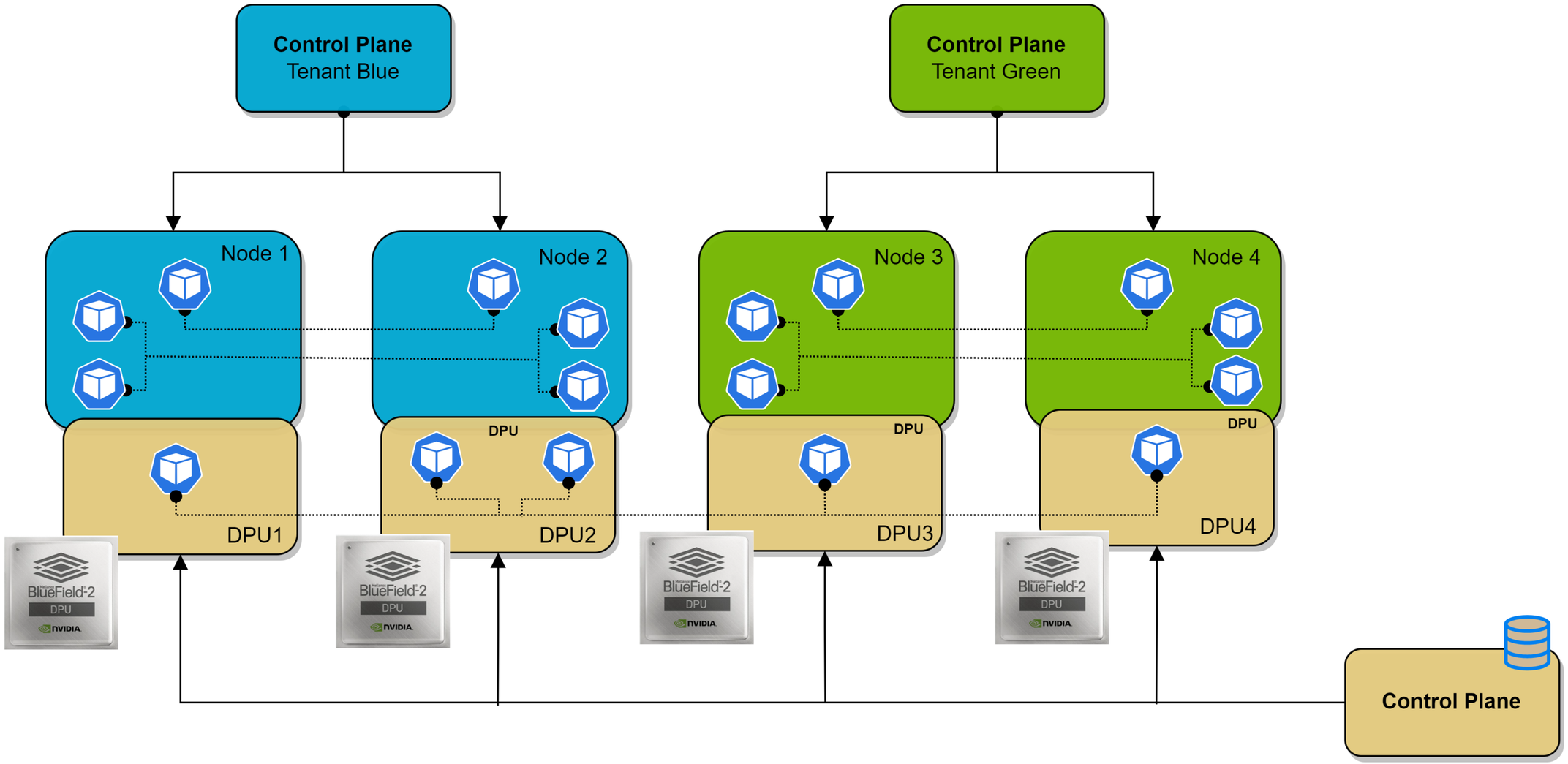
- DPU as a full Kubernetes node.
- Scheduling and running applications and services in the DPU



*rep = Representor



Shared OVN with DPU



Changes

- Running multiple instances of ovnkube-node on the same DPU host.
  - Mode: **full** for dpu pods and services
  - Mode: **dpu-host** for tenant pods in the BM host
- Different external bridge for each cluster
- Chassis association and stale object management
- Parameterized properties
  - Management port name
  - Conntrack zones
  - Metric ports.

DEMO

References and additional helpful links

- **OVN architecture**
  - https://www.ovn.org/support/dist-docs/ovn-architecture.7.html
- **KEP 1645 - Multi cluster services API**
  - https://github.com/kubernetes/enhancements/tree/master/keps/sig-multicluster/1645-multi-cluster-services-api
- **OVS hardware offload**
  - https://github.com/openshift/ovn-kubernetes/blob/master/docs/ovs_offload.md
- **ovn-kubernetes DPU support**
  - https://github.com/openshift/ovn-kubernetes/blob/master/docs/design/dpu_support.md
- **Scalable Functions (SF)**
  - https://github.com/Mellanox/scalablefunctions/wiki
- **Linux Subfunctions**
  - https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/Documentation/networking/devlink/devlink-port.rst?h=v5.13#n125


NVIDIA