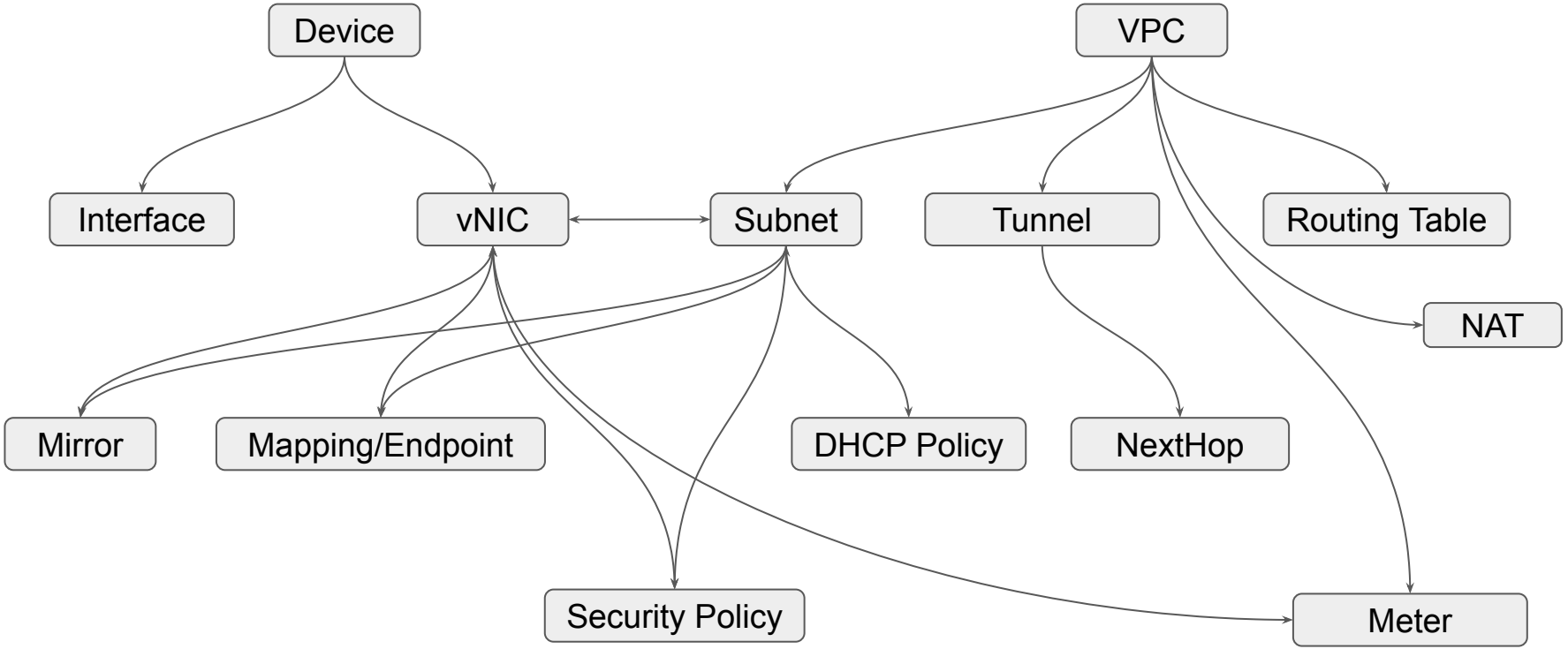
OPI Networking Protos

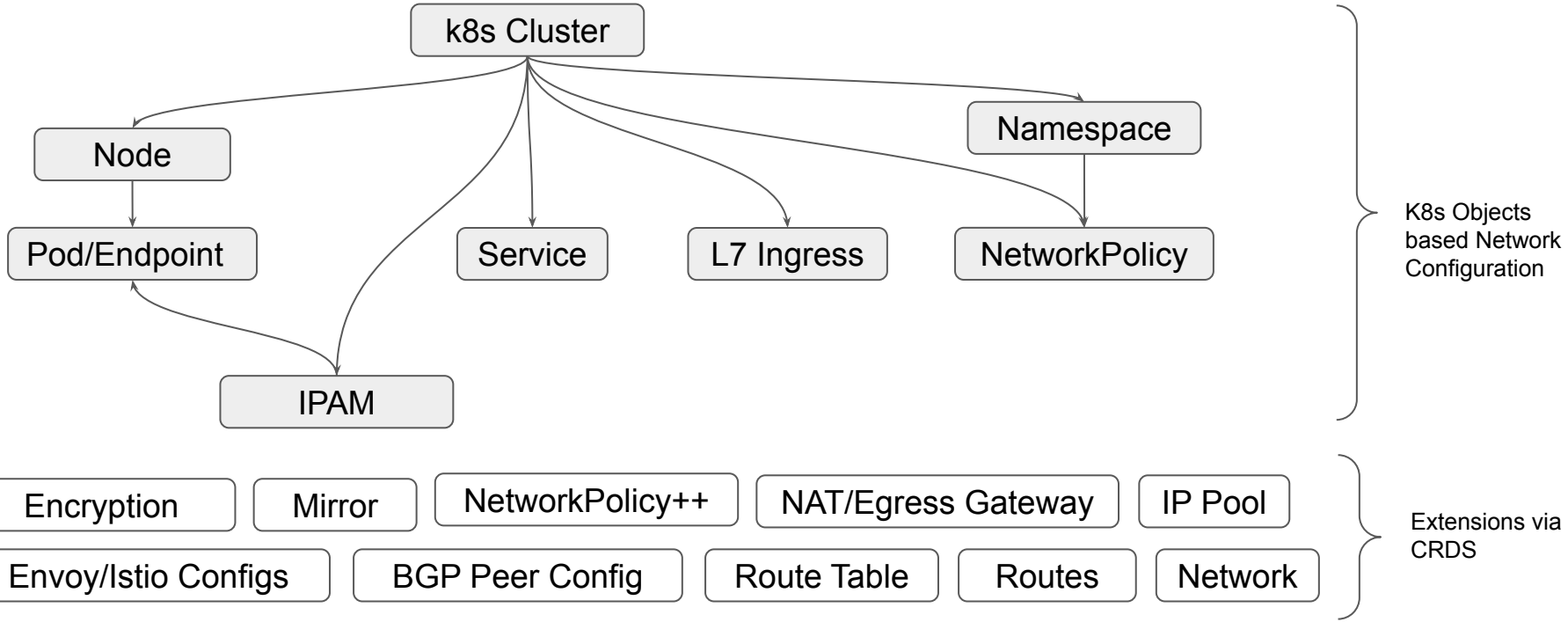
OPI Community Discussion
Vipin Jain, AMD Pensando
Oct 12th, 2022

Cloud Networking Model - for BMaaS/IaaS



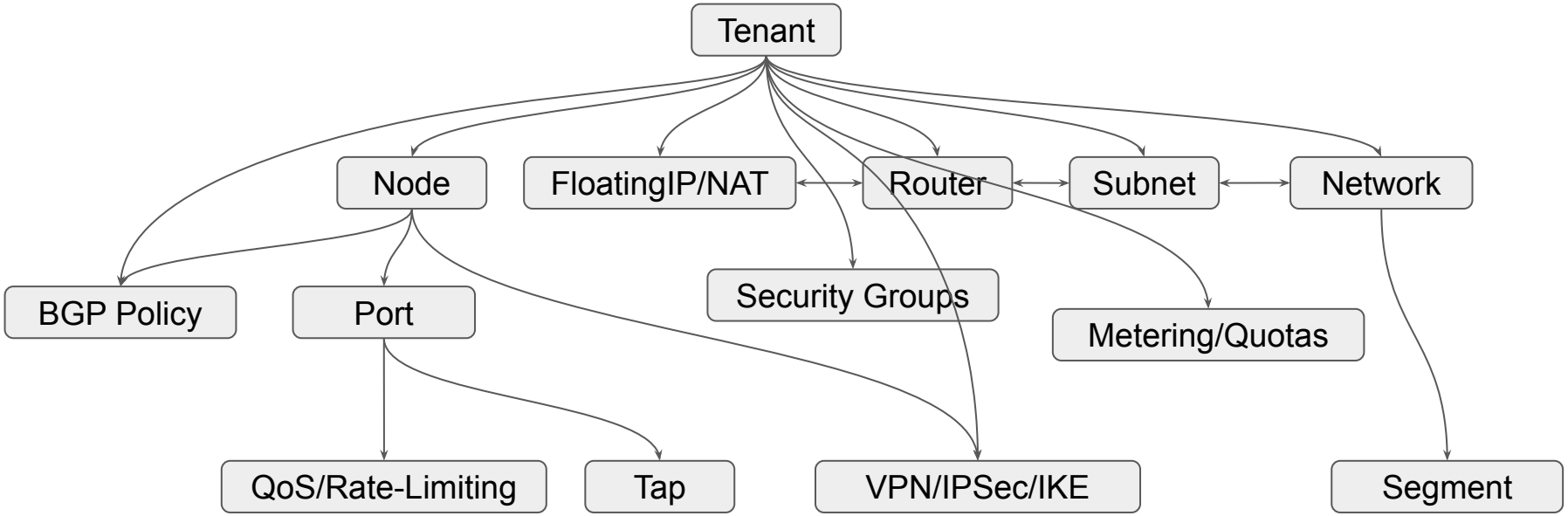
- https://github.com/Azure/DASH/tree/main/documentation/gnmi
- https://docs.oracle.com/en-us/iaas/Content/Network/Concepts/overview.htm
- https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html

K8s Networking Model - for CaaS



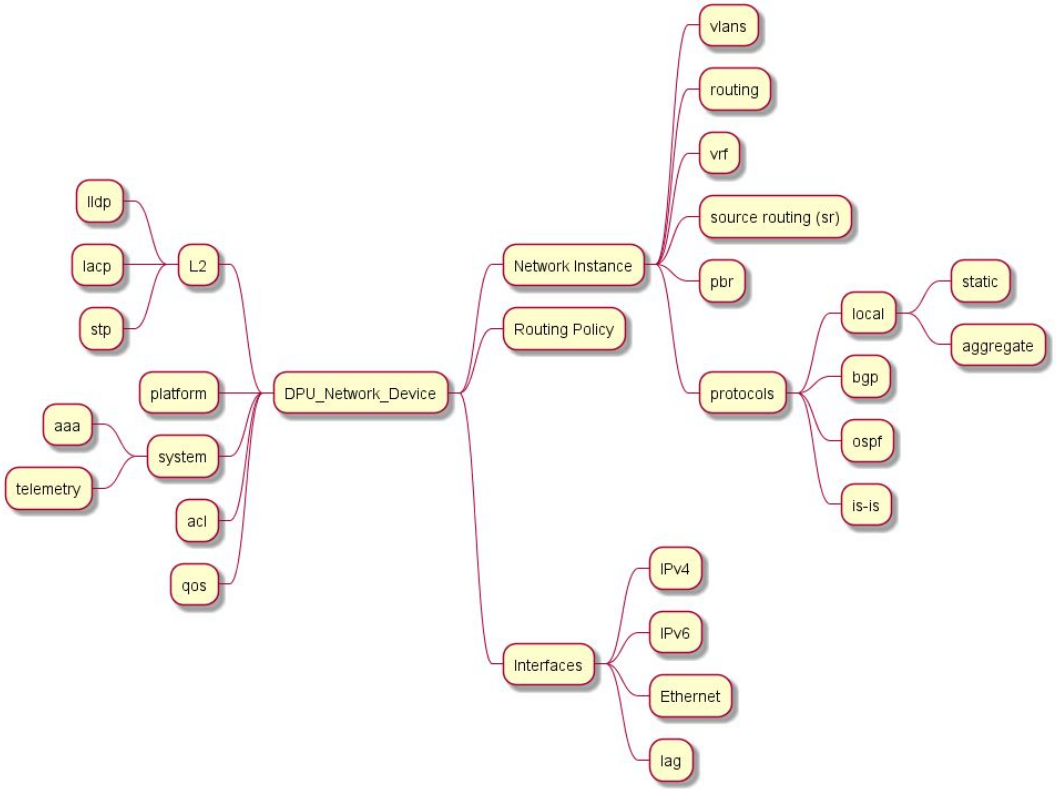
- https://github.com/cilium/cilium/blob/master/api/v1/openapi.yaml
- https://github.com/projectcalico/api/blob/master/pkg/openapi/openapi_generated.go
- https://github.com/ovn-org/ovn-kubernetes/tree/master/docs
- https://www.ovn.org/support/dist-docs/ovn-sb.5.html, https://www.ovn.org/support/dist-docs/ovn-nbctl.8.html

Openstack Neutron Networking - private cloud?



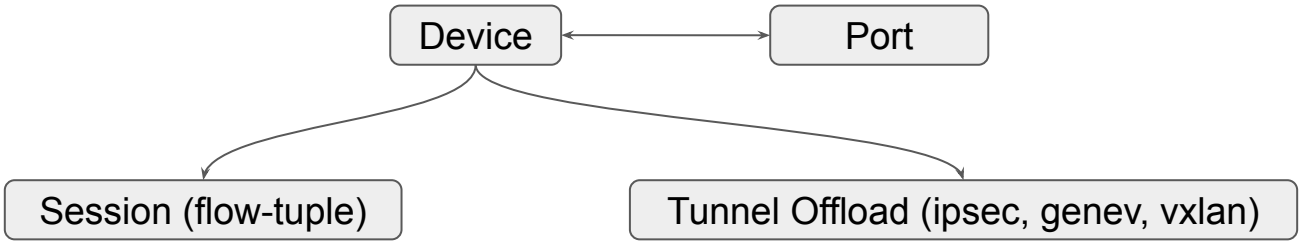
- https://docs.openstack.org/api-ref/network/v2/index.html
- Above is northbound APIs, drivers can use OVS, Linux Kernel and/or SR-IOV (DPU like device) based datapath

Switch/router configuration/telemetry - using OpenConfig



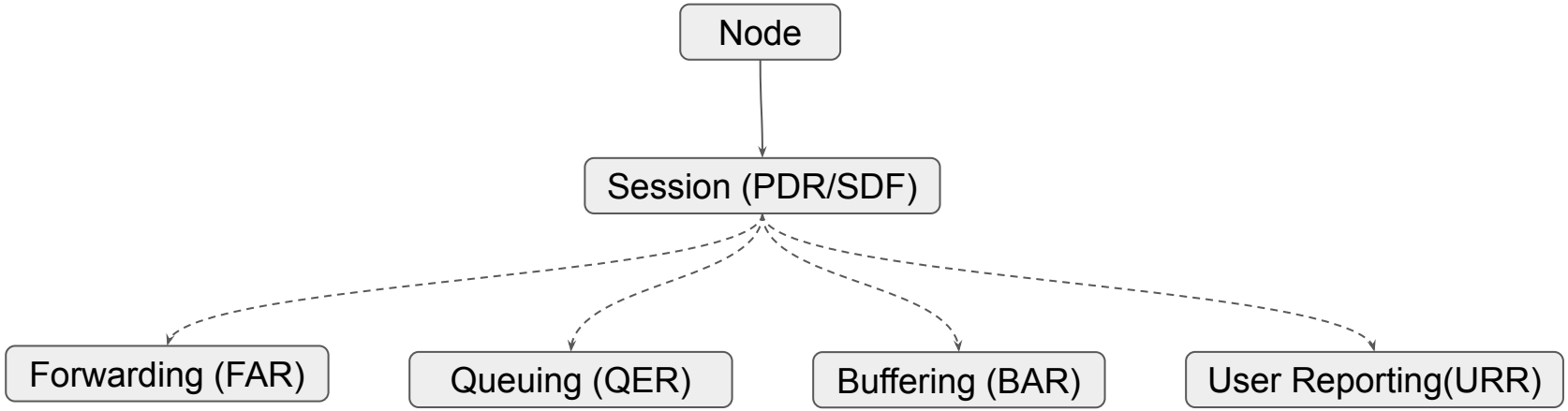
- https://github.com/opiproject/opi-api/blob/main/network/OpenConfig-Model.png
- Primarily for switch configuration (e.g. OSPF, IS-IS, etc.), do we want to model DPU as a switch/router?

Appliance Acceleration (e.g. SessionOffload)



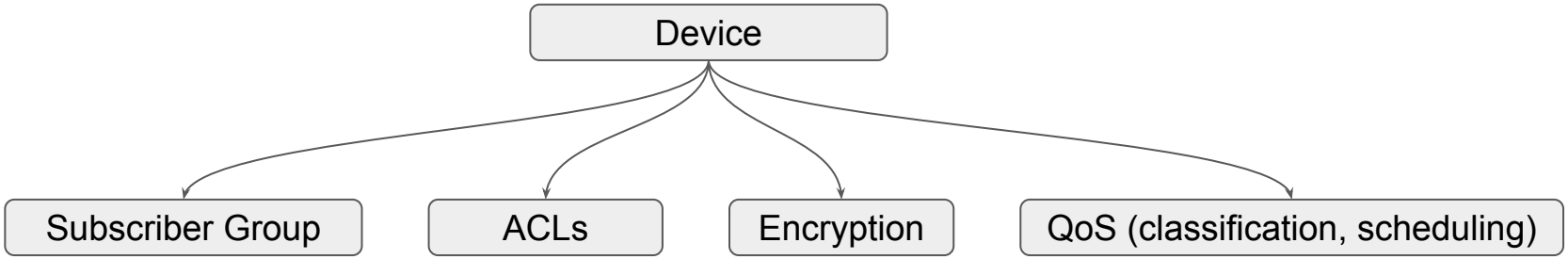
- https://github.com/att/sessionOffload/tree/master/protos
- Service provider use case (driven by AT&T) to offload long lived flows and/or tunnel termination/encryption/etc.

5G-UPF Networking Model (N4)



- Go PFCP Implementaton: https://github.com/wmnsk/go-pfcp
- UPF implementation with PFCP APIs using P4: https://www.cs.princeton.edu/~jrex/papers/up4-sosr21.pdf

CMTS (vCMTS) Networking Model



- Intel's vCMTS ref Implementation (C lang): https://www.intel.com/content/www/us/en/download/715489/intel-vcmts-reference-dataplane.html?cache=1643847231#:~:text=intel%2Dvcmtsd%2D-,v21.-%2D10%2D0.tar

Datapath for host networking

- Openflow/OVS
  - Based on open-flow programmable pipeline standard (by ONF), can run on
  - Well defined APIs for flow offload to any hardware (may or maynot be xPU)
- P4
  - Native language for parsing and defining stateful datapath, compiled using high level program
  - Standardized in P4 PNA (portable NIC architecture) at P4.org
- eBPF
  - Linux native C language programmable engine, portable across clouds and on-prem
  - Offloading requires natively running the compiled eBPF byte-code natively on xPU datapath
- DPDK/VPP
  - Defines RTE_FLOW offload APIs in DPDK implementation
  - Popular in VNF/CNF implementations; can leverage DPDK-OVS
- Linux Kernel
  - Implements routing, bridging, NAT, tunneling, load balancing, stateful ACLs, ipsec, etc.
  - Some parts can be offloaded (via tc), if not all of it

Portability - Object Level vs. Network Datapath

- Object Level (aka proto definition)
  - Benefit: Allows vertical integration, innovation for providing acceleration
  - Can use open source mechanisms underneath
  - Disadvantage: prescriptive (but for a given domain), and can evolve v1, v2 APIs, etc.
  - Benefit: quicker time to market, clear separation of boundaries between provider and user
  - Example: Cloud provider's APIs, K8s object models
- Network Datapath
  - Complimentary to object definitions, cuts at lower layer of integration for some users
  - Benefit: Allows flexibility, user can own the business logic, allows users to differentiate without using open source. In contrast object definition can also allow proprietary APIs/extensions
  - Disadvantage: Longer development time, dealing with vendor specific details (devil is always in details); can be expensive (cost of development/solution)
  - Example: Openflow (used by OVS), DPDK/RTE_FLOW, P4, eBPF
- Vendor's Dilemma
  - One datapath API mechanism doesn't work for all the domains e.g.
    - eBPF is gaining a lot of momentum with k8s
    - RTE_FLOW is popular with VNFs/CNFs vendors
    - P4 is suitable for high perf cloud services
  - One object level API doesn't work for all domains e.g.
    - Tenancy separation in cloud is very different form a trusted VNF/CNF running on the host
    - K8s API doesn't define multi-tenant VPCs work for cloud provider for BM

What is OPI's portability/multi-vendor goal?

	DPDK	P4	eBPF	Linux (tc, iptables,etc.)	Openflow (OVS)	Etc…
Cloud Services						
5G-UPF						
Appliance Accel						
CMTS						
CaaS/K8s						
Openstack?						
Future…						

- The table is intentionally left blank
- PR #128 attempts to address cloud services, using whatever datapath vendor chooses to use
- What datapath is better for which domain? eBPF, OVS (kernel or DPDK), or Linux kernel