

AMAZING POWERED BY INTEL



ENABLING NFV FEATURES IN KUBERNETES

IVAN COUGHLAN

IVAN.COUGHLAN@INTEL.COM

SOFTWARE ARCHITECT

KURALAMUDHAN RAMAKRISHNAN

KURALAMUDHAN.RAMAKRISHNAN@INTEL.COM

SENIOR SOFTWARE ENGINEER

DATA CENTER NETWORK SOLUTION GROUP

INTEL

LEGAL NOTICES AND DISCLAIMERS

- Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.
- No computer system can be absolutely secure.
- Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit <http://www.intel.com/performance>.
- Intel, the Intel logo, Xeon, and others are trademarks of Intel Corporation in the U.S. and/or other countries. *Other names and brands may be claimed as the property of others.
- © 2017 Intel Corporation.

WHAT WILL YOU LEARN TODAY?

1. Containers Deployment Models for NFV ecosystem
2. Addressing Data Plane Scalability in Containers:
 - Container Bare Metal Reference Architecture
 - Compute
 - Network

CONTAINERS NETWORKING DEPLOYMENTS CONSIDERATIONS



VNFs

vCMTS

vIMS

vEPC

vCPE

vSBC

NFVi- Network



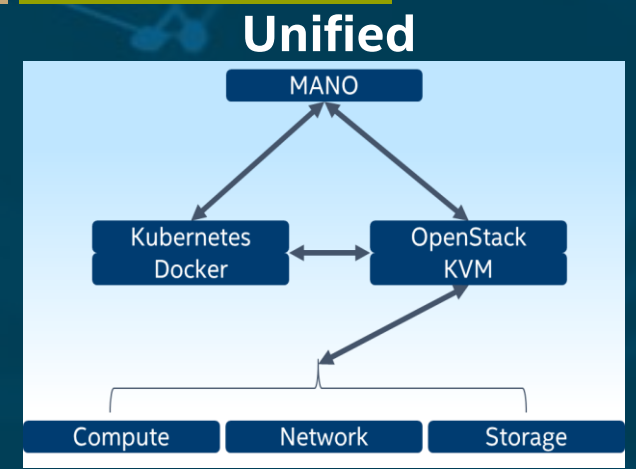
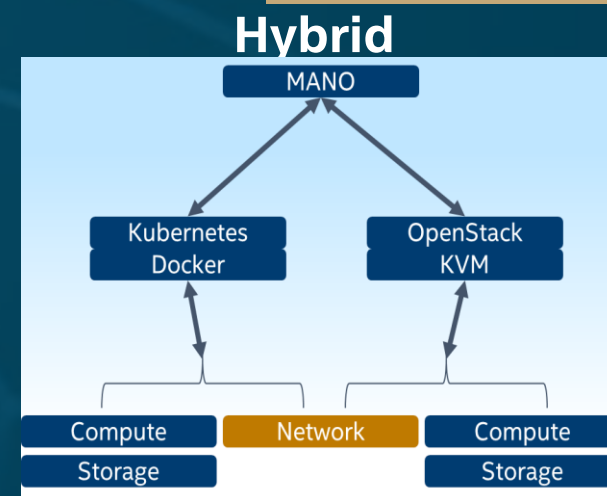
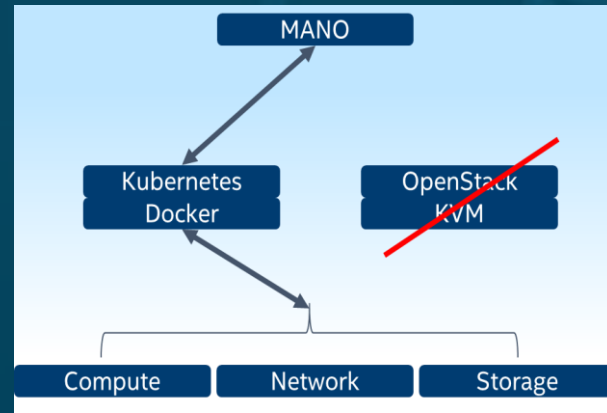
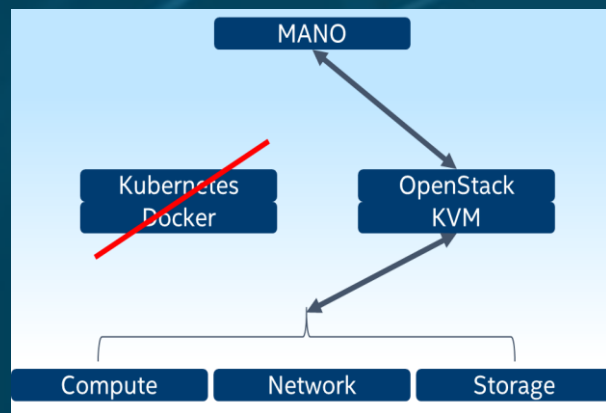
SR-IOV



NFV Orchestration



kubernetes



CONTAINERS NETWORKING DEPLOYMENTS CONSIDERATIONS

COLLABORATE WITH EARLY MOVERS, DRIVE OPEN SOURCE DEVELOPMENTS AND ENABLE THE INDUSTRY



VNFs

- VCMTS
- VIMS
- vEPC
- VCPE
- vSBC

NFVi- Network



SR-IOV



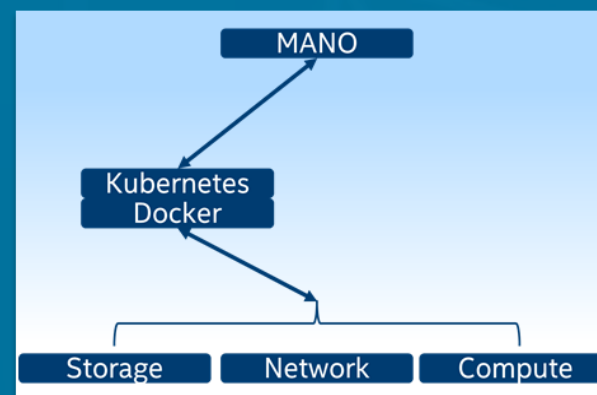
NFV Orchestration



kubernetes

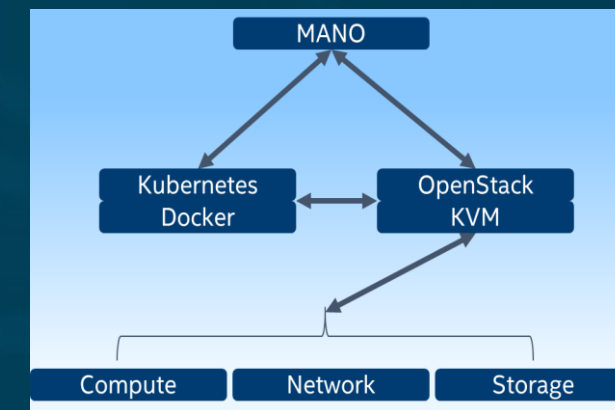
Containers Bare Metal

Containers



Containers & VM Unified Infrastructure

VM
Containers



DATA PLANE CHALLENGES IN CONTAINERS

Multiple network interfaces for VNFs



High performance Data Plane (N-S)



High performance Data Plane (E-W)



High performance Data Plane (UI)



Removing performance penalties for container in VM (UI)



Ability to request/allocate platform capabilities



CPU Core-Pinning for K8s pods



Dynamic Huge Page allocation



Platform telemetry information



CNI

- MULTUS
- CNI SR-IOV DPDK
- CNI VHOST USER DPDK
- CNI Master VM DPDK
- Kuryr-Kubernetes
- Node Feature Discovery
- CPU Manager for Kubernetes
- Native Huge page support for Kubernetes
- collectd

SCALING

Open Source: CNI plug-in - V2.0 June '17

Upstream K8s: TBD

Open Source: CNI plug-in - V2.0 April '17

Open Source: CNI plug-in - V1.0 Sep '17

In Development

Kuryr-k8s v0.1.0: Jun '17

Open Source: Nov. '16
Upstream K8: Incubation
Graduation TBD

Open Source: V1.2 April '17
Upstream K8: Phase 1 - V1.8
Sept '17

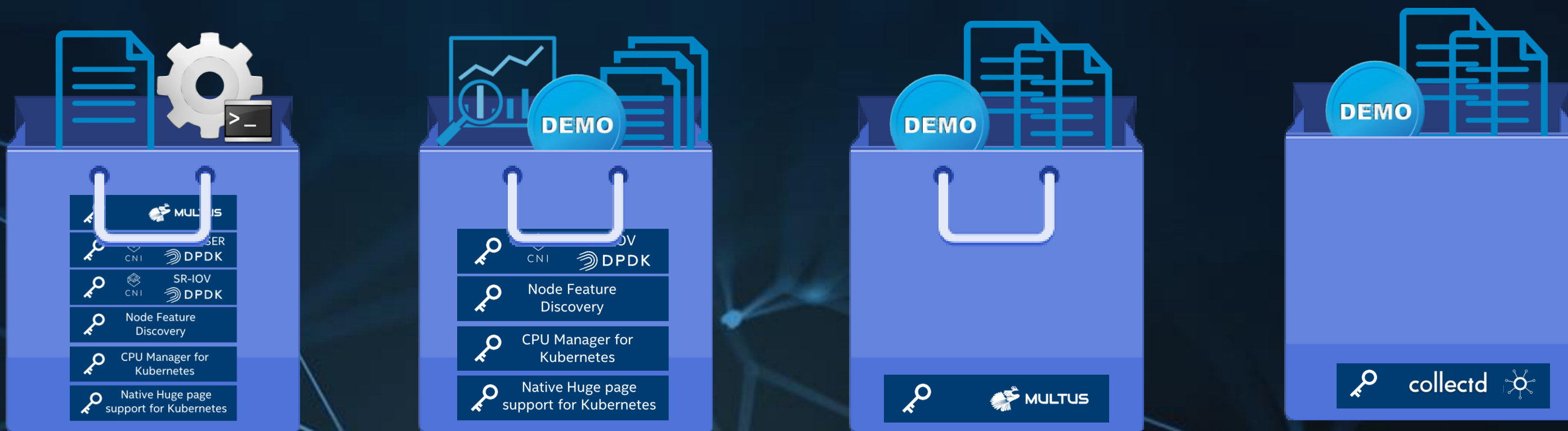
Upstream K8: V1.8 Sept '17

Upstream collectd: V5.7.2
June '17 ; 5.8.0 ((Q4 2017
date TBD)

Open Source: Available on Intel github <https://github.com/Intel-Corp> | NFD at <https://github.com/kubernetes-incubator/node-feature-discovery>

CONTAINER BARE METAL EXPERIENCE KITS

A LIBRARY OF BEST-PRACTICE DEVELOPMENT GUIDELINES FOR CONTAINER BARE METAL ORCHESTRATION



Reference Architecture

Reference Architectures
Installation Scripts
Reference Architecture
User Guide

Enhance Platform Awareness

Feature Brief
White Paper
Tech. Application Note
Benchmark Test Report
Demo

Kubernetes Networking

Feature Brief
Tech. Application Note
Demo

Platform Telemetry

Feature Brief
Tech. Application Note
Demo

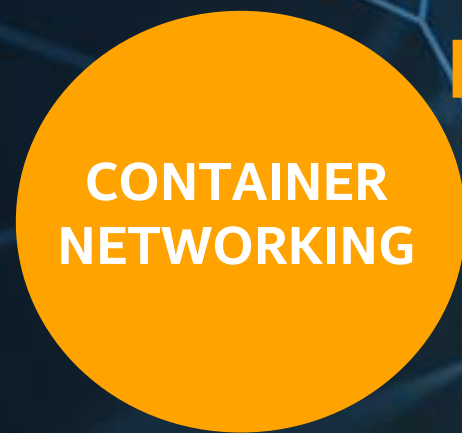
Released throughout December, 2017 on: <https://networkbuilders.intel.com/network-technologies/container-experience-kits>

CONTAINER NETWORKING GTM

Intel is addressing key challenges to using containers for NFV use cases

Many of these have been open sourced already

Material will be made available throughout November <https://networkbuilders.intel.com/network-technologies/intel-container-experience-kits>

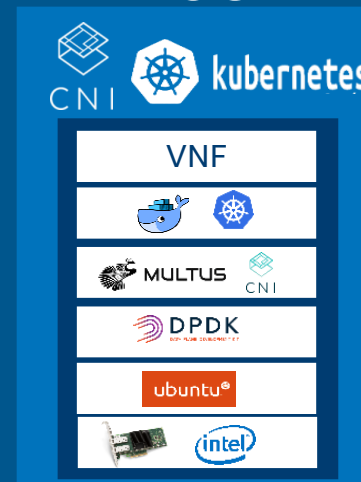


SOFTWARE CONTAINER CAPABILITIES

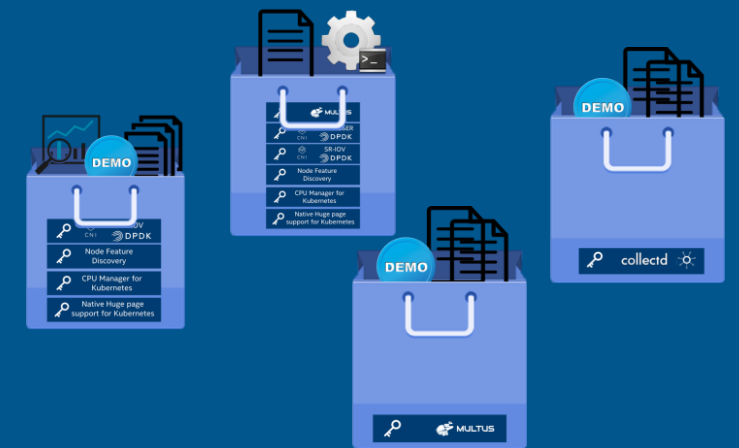
- MULTUS
- SR-IOV CNI PLUGIN
- HUGE PAGES
- NODE FEATURE DISCOVERY
- CPU MANAGER
- RESOURCE MANAGMENT
- HIGH AVAILABILITY



COMMUNITY OPEN SOURCE POC



SCALE WITH PARTNERS EXPERIENCE KITS Best Practice Guidelines



NETWORK CLOUDIFICATION

CONTAINERS BARE METAL

VNFs



NFVi- Network



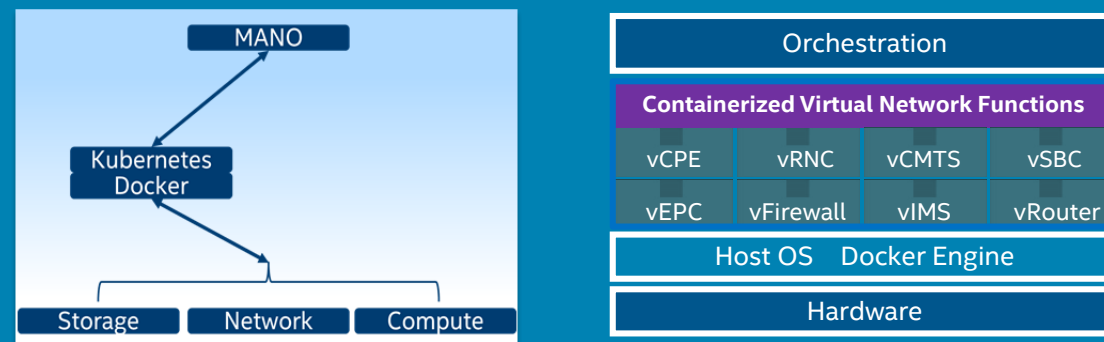
SR-IOV



NFV Orchestration



Containers Bare Metal



INDUSTRY CHALLENGES IN CONTAINERS BARE METAL



kubernetes



CNI

Multiple network interfaces for VNFs



MULTUS

Support for high performance Data Plane (N-S)



SR-IOV

DPDK

Support for high performance Data Plane(E-W)



VHOST USER

DPDK

Ability to request/allocate platform capabilities



Node Feature
Discovery

Support for CPU Core-Pinning for K8s pods



CPU Manager for
Kubernetes

Dynamic Huge Page allocation



Native Huge page
support for Kubernetes

MULTIPLE NETWORK INTERFACE FOR VNFS

PROBLEM

In NFV use cases, one key requirement is the functionality to provide multiple network interfaces to the virtualized operating environment of the Virtual Network Function (VNF). Kubernetes support only one Network interface – “eth0”

USE CASES

Functional separation of control and data network planes

Link aggregation for redundancy of the network

Support for implementation of different network protocol stacks and/or SLAs

Network segregation and Security

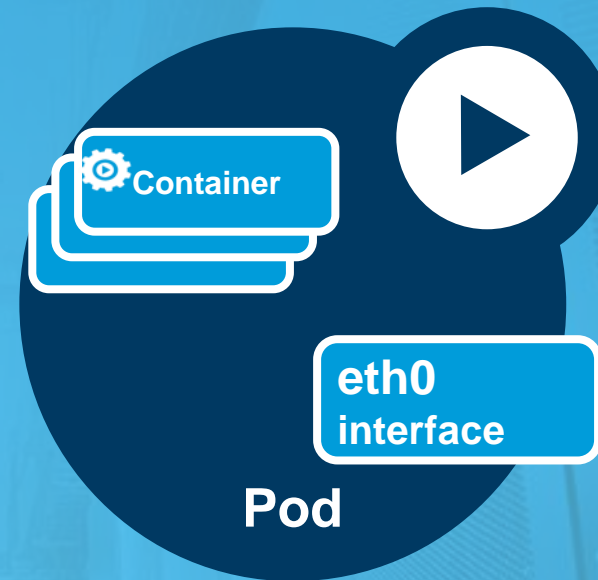
REFERENCE

Multus CNI – <https://github.com/Intel-Corp/multus-cni>

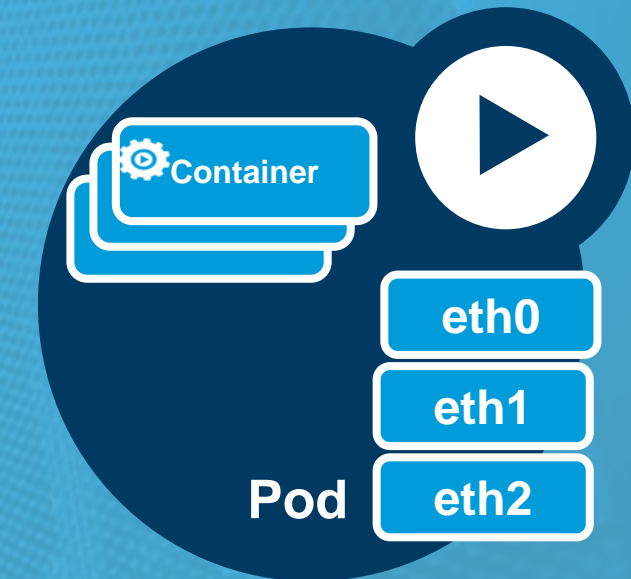
Native Kubernetes - Mailing list with details on discussions

<https://groups.google.com/forum/#!forum/kubernetes-sig-network>

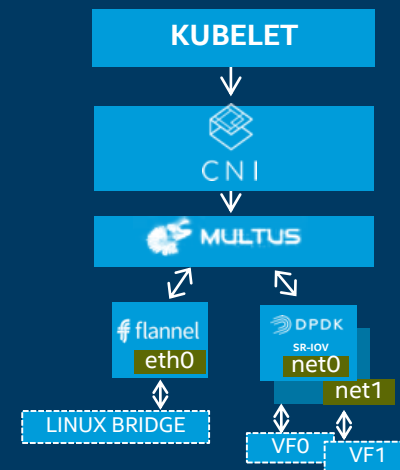
Current Kubernetes Networking



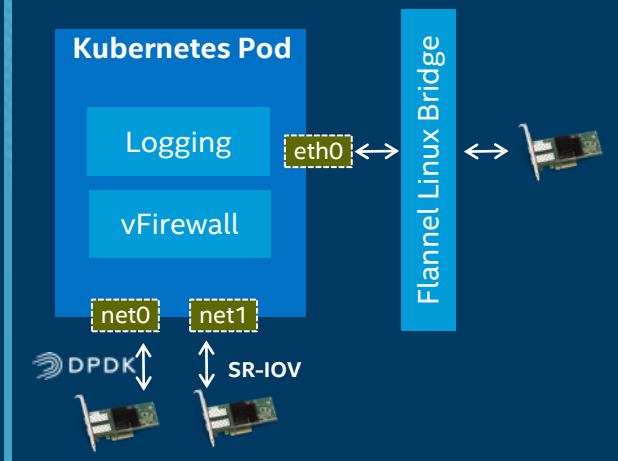
NFVI Requirement in Kubernetes Networking



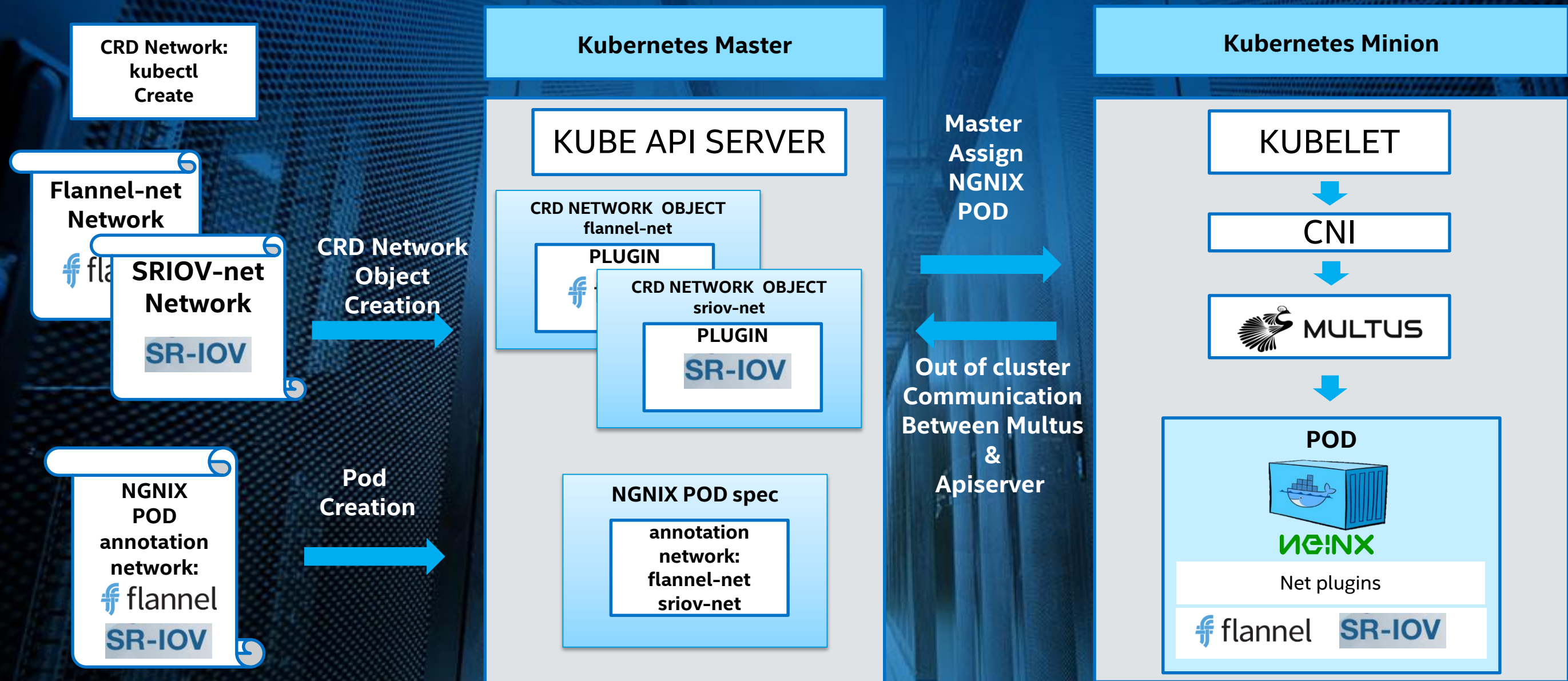
Network Control Flow with Multus



Pod Network Interfaces with Multus



MULTUS - MULTI NETWORKING IN KUBERNETES SETUP



DPDK - SRIOV CNI PLUGIN

PROBLEM

No support for physical platform resource isolation guaranteeing network I/O performance & determinism
No support for Data Plane Networking in userspace

SOLUTION

Enables NIC SR-IOV support in Kubernetes via a CNI plugin

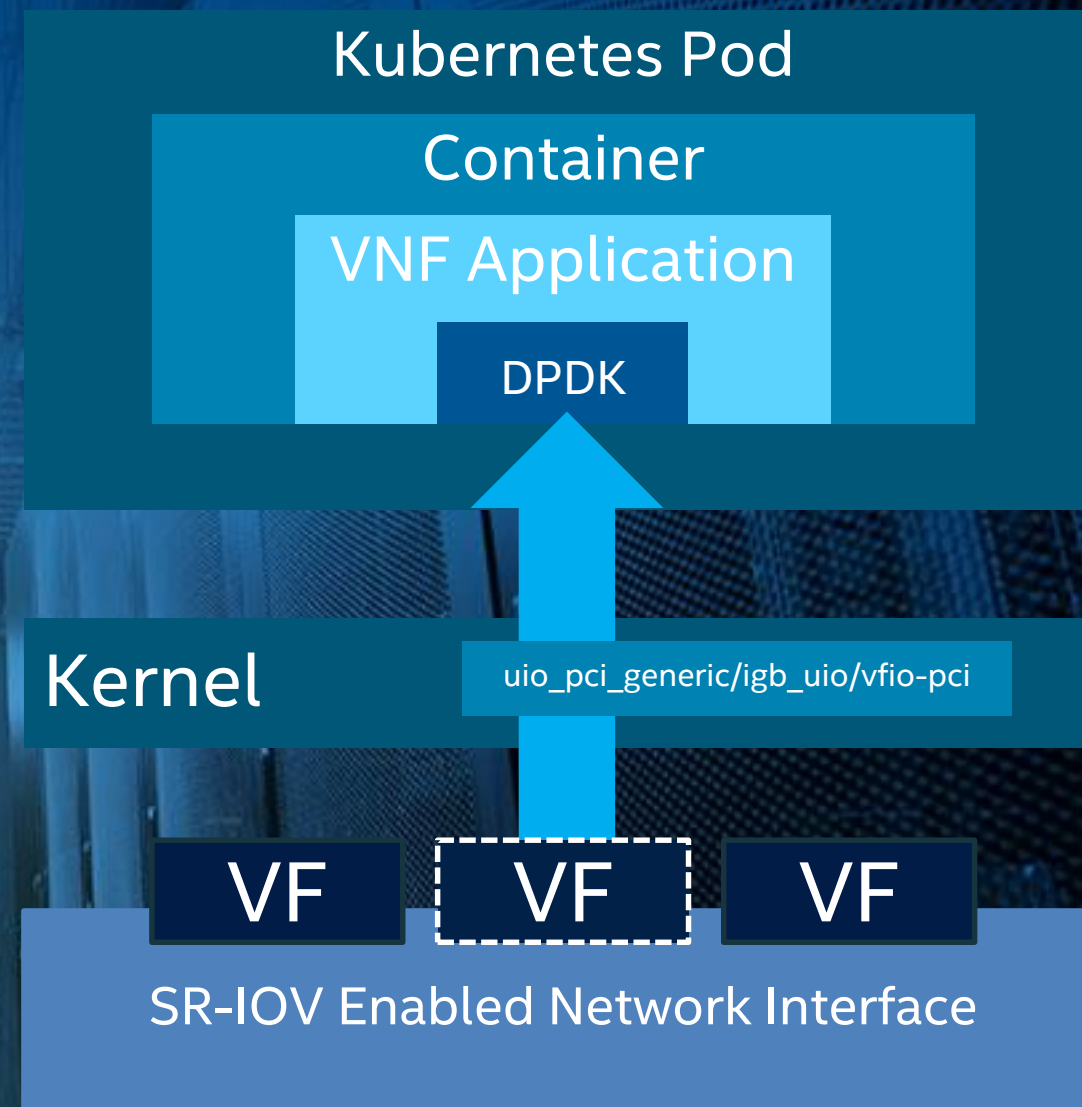
Supports two modes of operation:

SR-IOV : SR-IOV VFs are allocated to pod network namespace

DPDK : SR-IOV VFs are bounded to DPDK drivers in the userspace

REFERENCE

<https://github.com/Intel-Corp/sriov-cni>



VHOST USER CNI PLUGIN

PROBLEM

No networking solution with software acceleration for inter-pod connectivity on same host (e.g. SFC use case)

SOLUTION

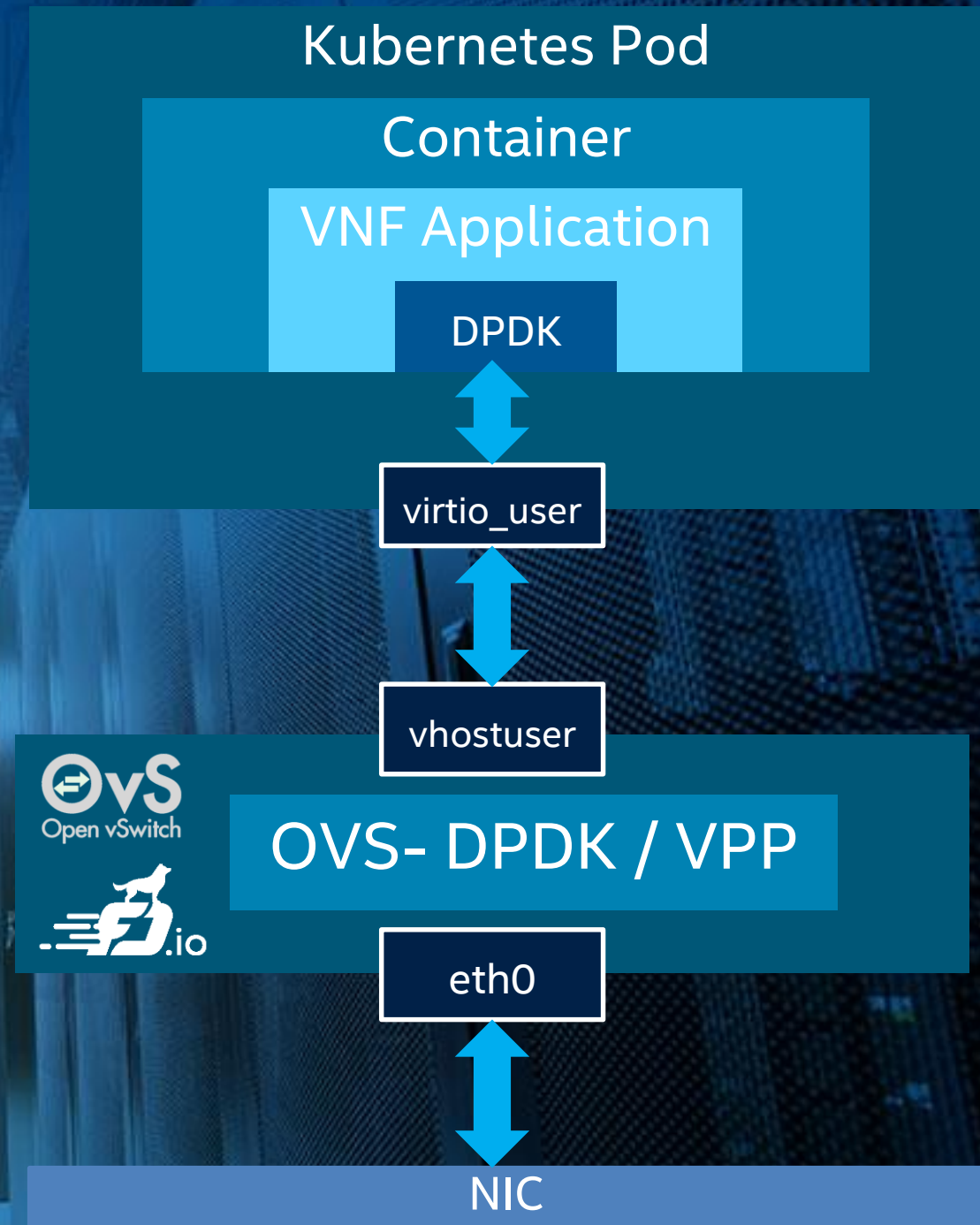
Virtio_user/ vhost_user gives boosted performance than VETH pairs

Support VPP as well as DPDK OVS

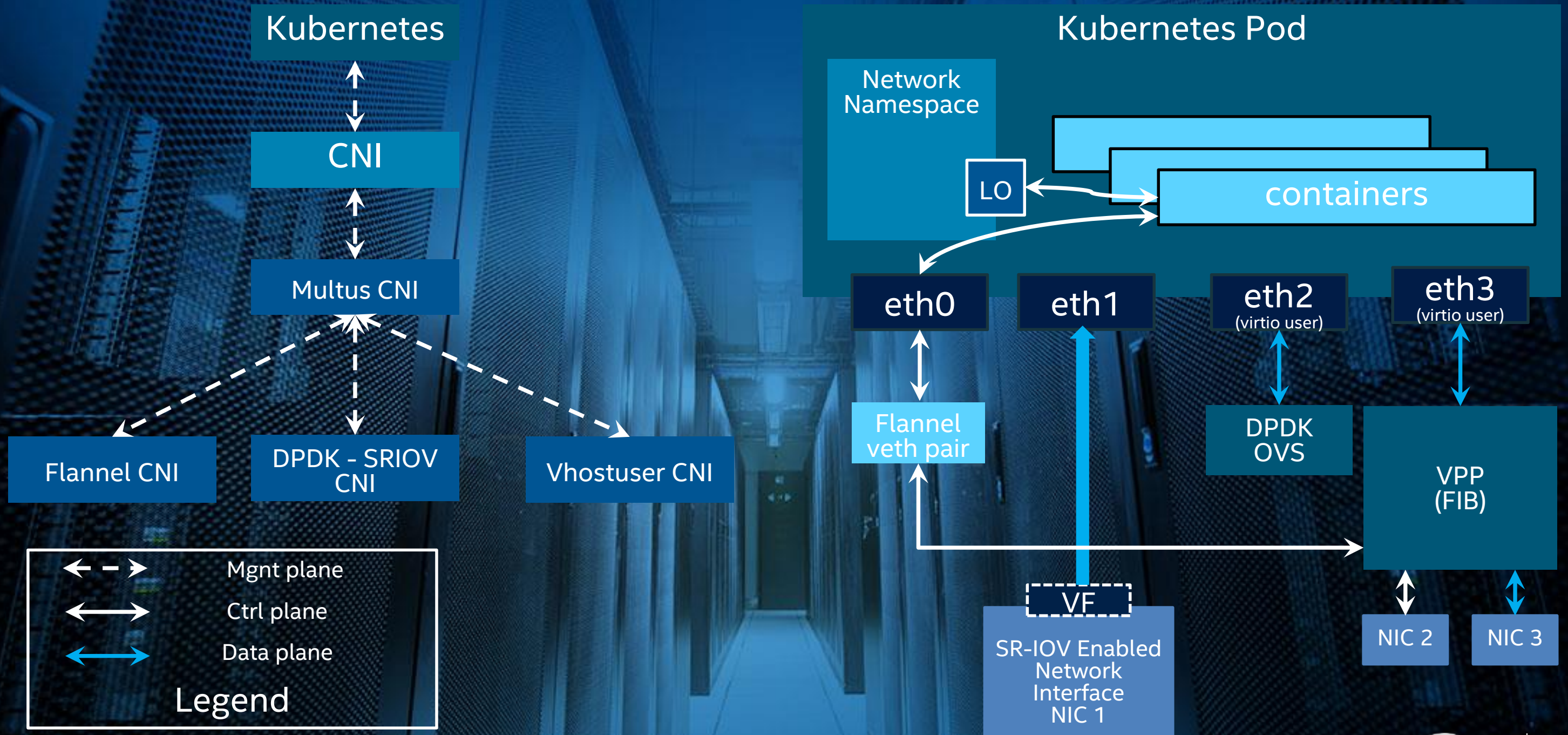
Vhost_user CNI plugin enables K8s to leverage data plane acceleration

REFERENCE

<https://github.com/intel/vhost-user-net-plugin> (V1.0 Sep '17)



VHOSTUSER CNI PLUGIN INTEGRATION WITH MULTUS



NODE FEATURE DISCOVERY(NFD)

PROBLEM

No way to identify hardware capabilities or configuration
Inability for workload to request certain hardware feature

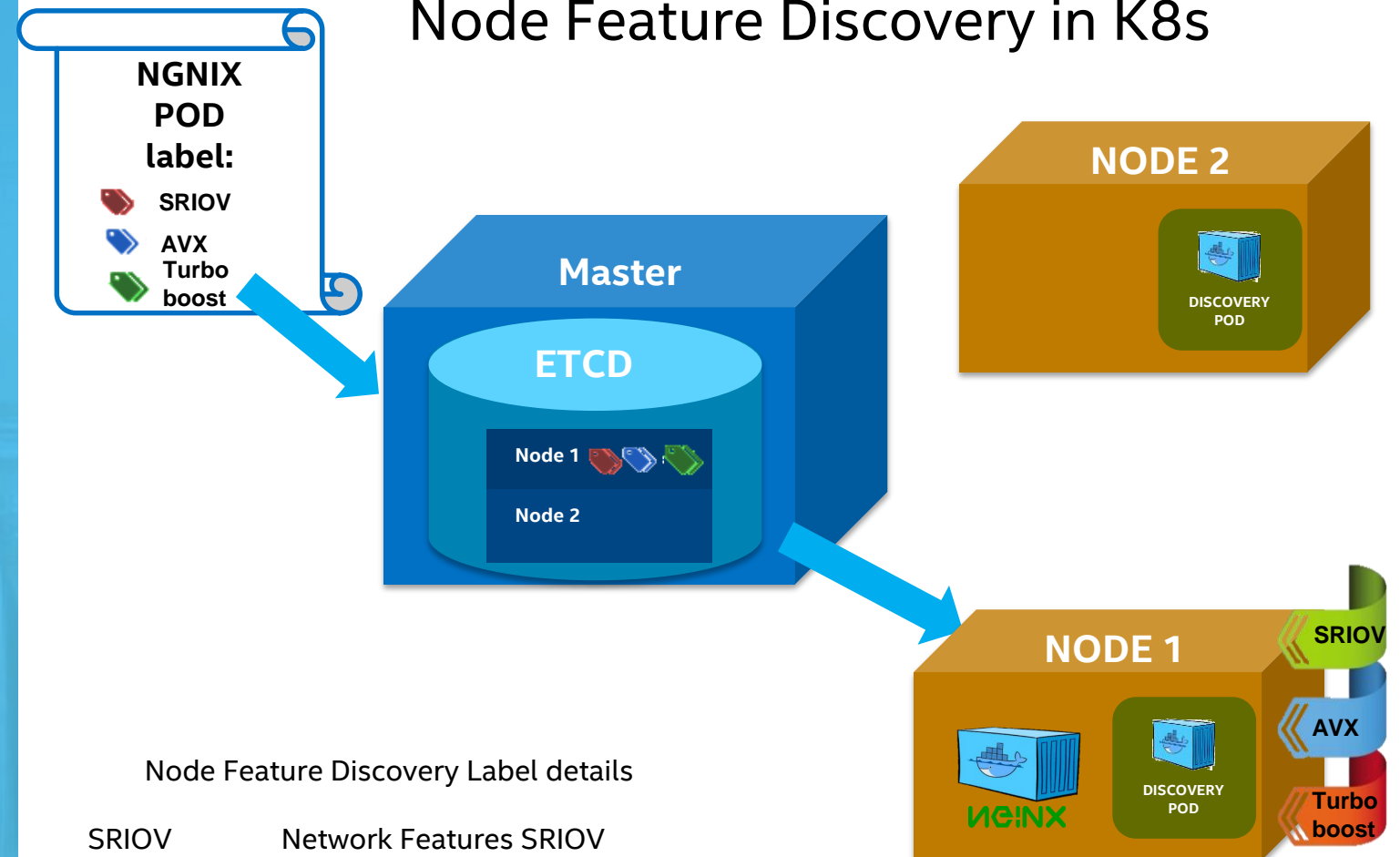
SOLUTION

Node feature Discovery brings Enhanced Platform Awareness (EPA) in K8s
NFD detects resources on each node in a Kubernetes cluster and advertises those features
Allows matching of workload to platform capabilities

REFERENCE

<https://github.com/kubernetes-incubator/node-feature-discovery>

Node Feature Discovery in K8s



Node Feature Discovery Label details

SRIOV	Network Features SRIOV
AVX	CPUID Features AVX
Turboboost	Intel TurboBoost enabled

CPU MANAGER FOR KUBERNETES – CPU PINNING AND ISOLATION

PROBLEM

Kubernetes has no mechanism to support core pinning and isolation

Results in high priority workloads not achieving SLAs

SOLUTION

CPU-Manager-For-Kubernetes introduces core pinning and isolation to K8s without requiring changes to the k8s code base

CMK guarantees high priority workloads are pinned to exclusive cores

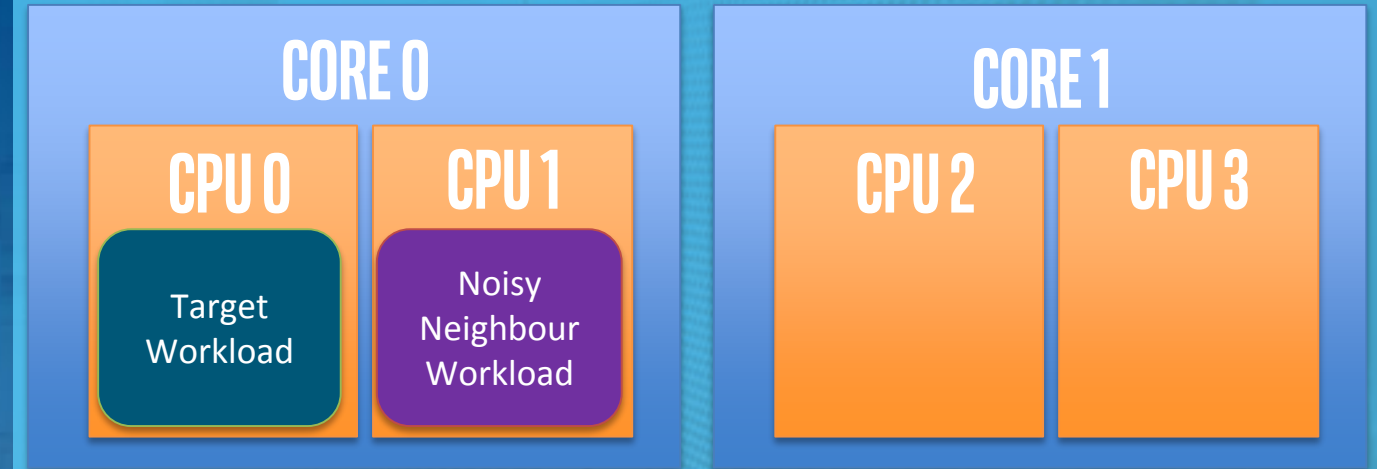
Gives a performance boost & determinism to high priority applications

Negates the noisy neighbour* scenario

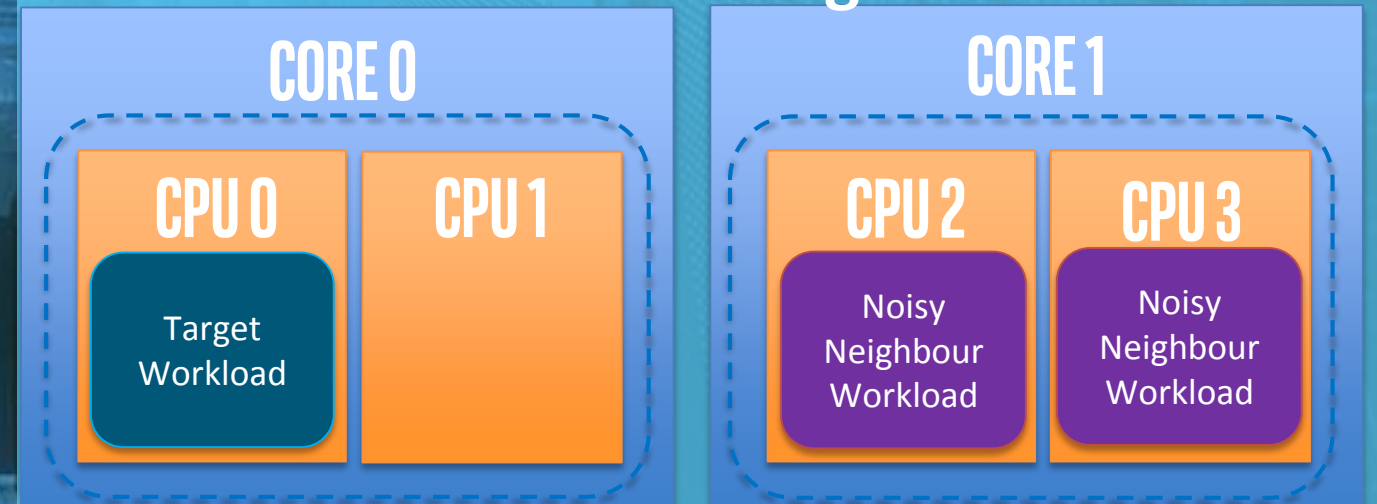
REFERENCE

<https://github.com/Intel-Corp/CPU-Manager-for-Kubernetes>

WITHOUT CMK: CPU Pinning and Isolation



WITH CMK: CPU Pinning and Isolation



* **Noisy Neighbor Workload:** An application that affects other applications sharing the infrastructure to suffer from nondeterministic performance e.g. context switching, cache affects

HUGE PAGE NATIVE SUPPORT IN KUBERNETES

PROBLEM

No resource management of Huge Pages in kubernetes
Responsibility of the cluster operator to handle it manually

SOLUTION

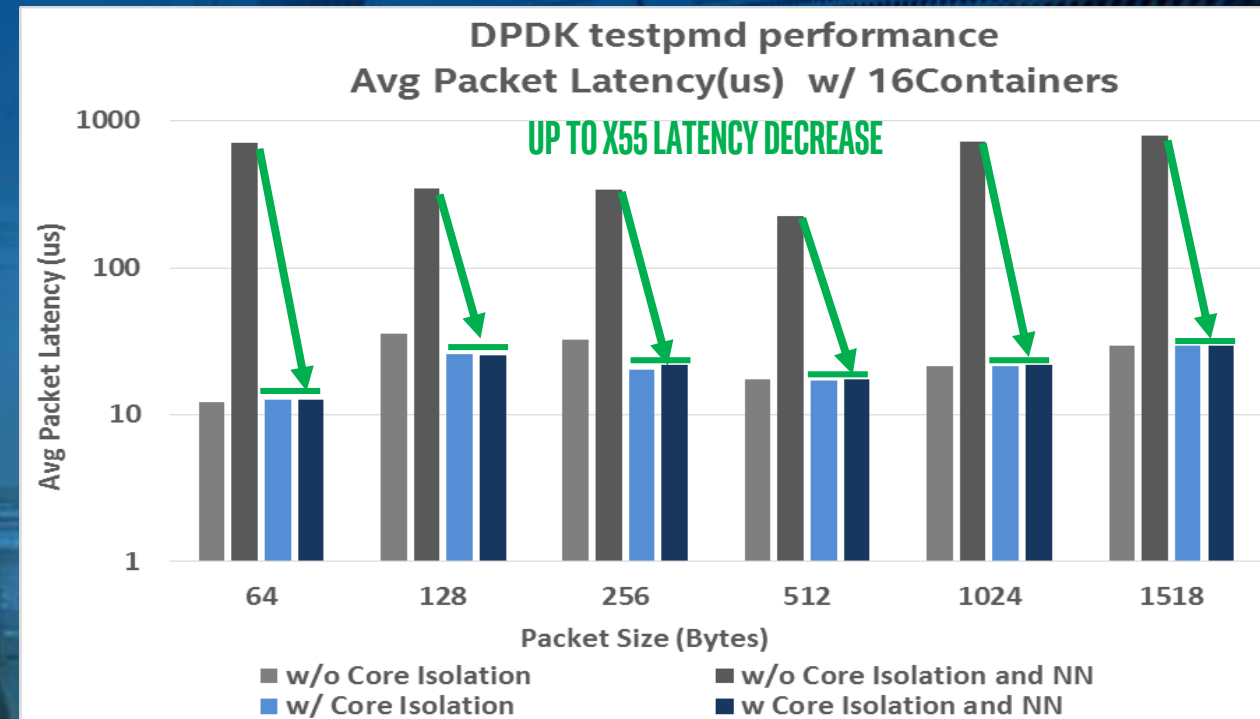
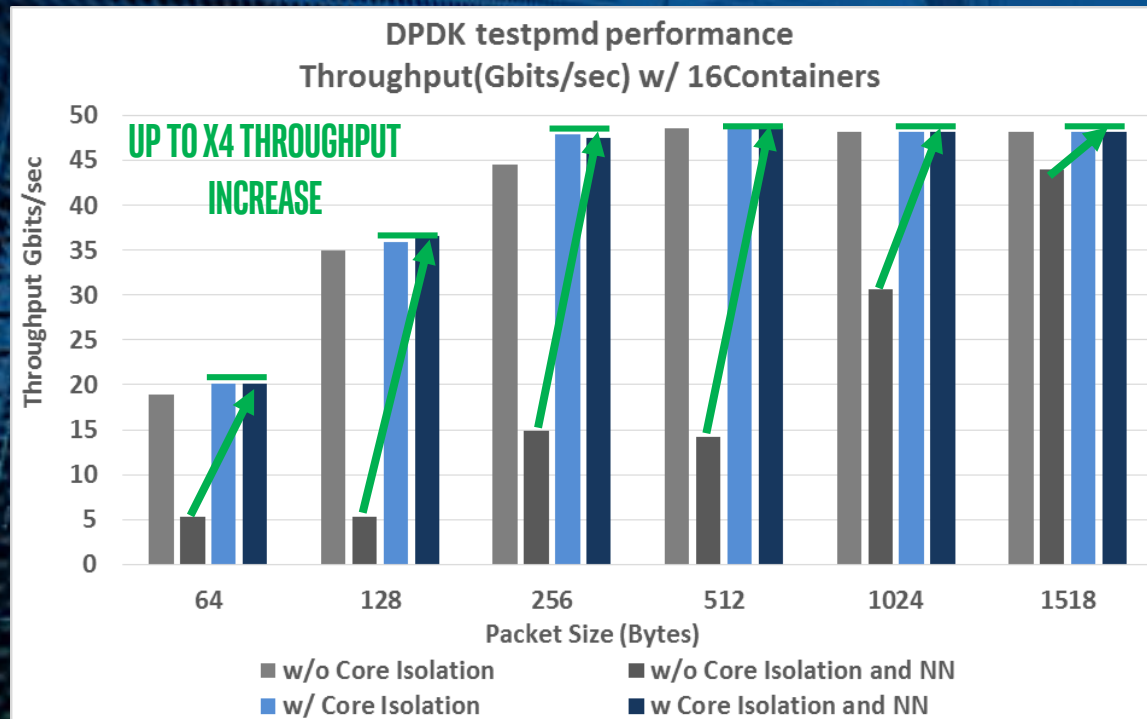
Huge Pages introduced as first class resource in kubernetes
Support for hugepages via hugetlbfs enabled through a memory backed volume plugin
Inherent accounting of Huge Pages
Automatic relinquishing of Huge Pages in case of unexpected process termination

REFERENCE

[Alpha support for pre-allocated hugepages](#)
[Hugetlbfs support based on empty dir volume plugin](#)

EXPERIENCE KIT EXAMPLE: CPU MANAGER FOR KUBERNETES BENCHMARK TEST

CORE ISOLATION LEADS TO PERFORMANCE CONSISTENCY SOLVING NOISY WORKLOADS PROBLEM



Core Isolation increase throughput of target-workload >200% for small packets in presence of Noisy Workload

Core Isolation decrease latency of target workload up >x13 in presence of Noisy Workload

Test are done with 16 Target Workloads" (=16 Containers) and with or without Noisy Workload present
1 Core with 2 threads are assigned to each container. Noisy Workload uses any available (non-isolated) cores in the system
Platform: Intel® Xeon® Gold Processor 20C@2.00 GHz (6138T); DPDK L2 Forwarding using XXV710 NICs

Disclaimer: For more complete information about performance and benchmark results, visit <http://www.intel.com/performance>. ; Test configuration: Master & Minion Nodes: {mother board: Intel Corporation; S2600WFQ; CPU: Intel® Xeon® Gold Processor 6138T; 2.0 Ghz; 2 socket; 20 cores; 27.5 MB; 125 W; Memory: Micron MTA36ASF2G72PZ; 1 DIMM/Channel, 6 Channel/Socket; BIOS: Intel Corporation SE5C620.86B.0X.01.0007.060920171037; NIC: Intel Corporation; Ethernet Controller XXV710 for 2x25GbE Firmware version 5.50; SW: Ubuntu 16.04.2 64bit; Kernel 4.4.0-62-generic x86_64; DPDK 17.05}; IXIA* - IxNetwork 8.10.1046.6 EA; Protocols: 8.10.1105.9, IxOS 8.10.1250.8 EA-Patch1

NETWORK CLOUDIFICATION

CONTAINER UNIFIED INFRASTRUCTURE DEPLOYMENT MODEL



VNFs

vEPC

vNAT

vIMS

vRouter

vGGSN

vFirewall

vCPE

vRNC

vHLR

vSGSN

vMME

vIDS

NFVi- Network



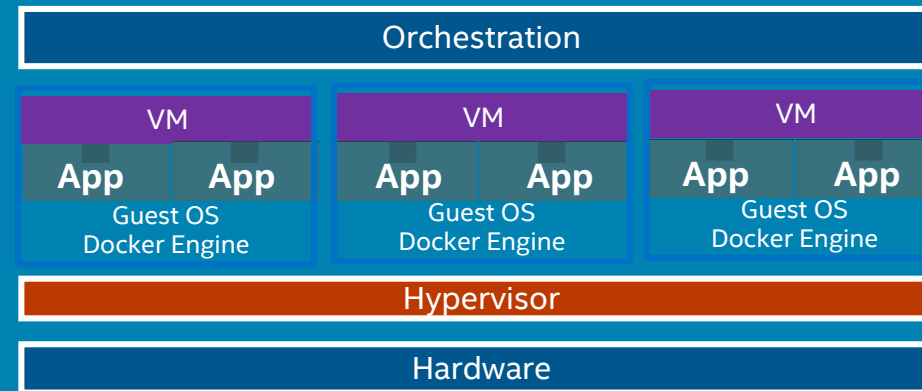
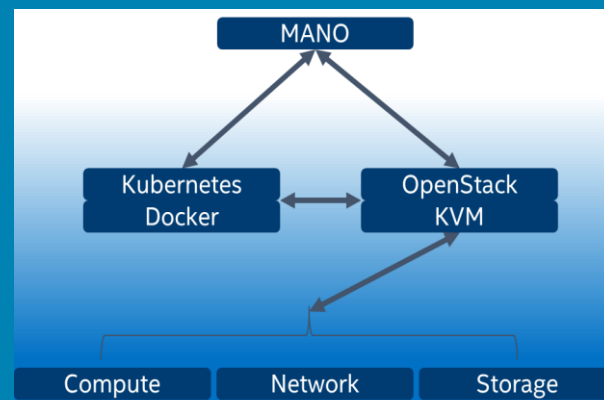
SR-IOV



NFV Orchestration



Containers Unified Infrastructure



INDUSTRY CHALLENGES IN CONTAINERS UNIFIED INFRASTRUCTURE

KURYR KUBERNETES

Removing Network performance penalties for container in VM 

Support for high performance Data Plane (E-W) 

Multiple network interfaces for VNFs 

Support for CPU Core pinning for Kuryr-K8s pods 



Kuryr-Kubernetes



MASTER VM
DPDK



MULTUS



CPU Manager for
Kubernetes

Same as in
Container
Bare Metal

MASTER VM FOR CONTAINERS

ENABLING DPDK IN NESTED CONTAINERS

OBJECTIVES

One Virtual Machine to many Containers
 Target: 1k Containers per VM
 Container Data Plane performance

USE CASE

Elasticity and scalability of containerized VNF application in VM

BENEFITS

VT-x ring de-privileging to move the VM and Container into userspace, making it accessible to the userspace vSwitch with just a single copy.

Standard Virtio interface & control plane supporting both interrupt and poll modes, VNF and Cloud based applications.

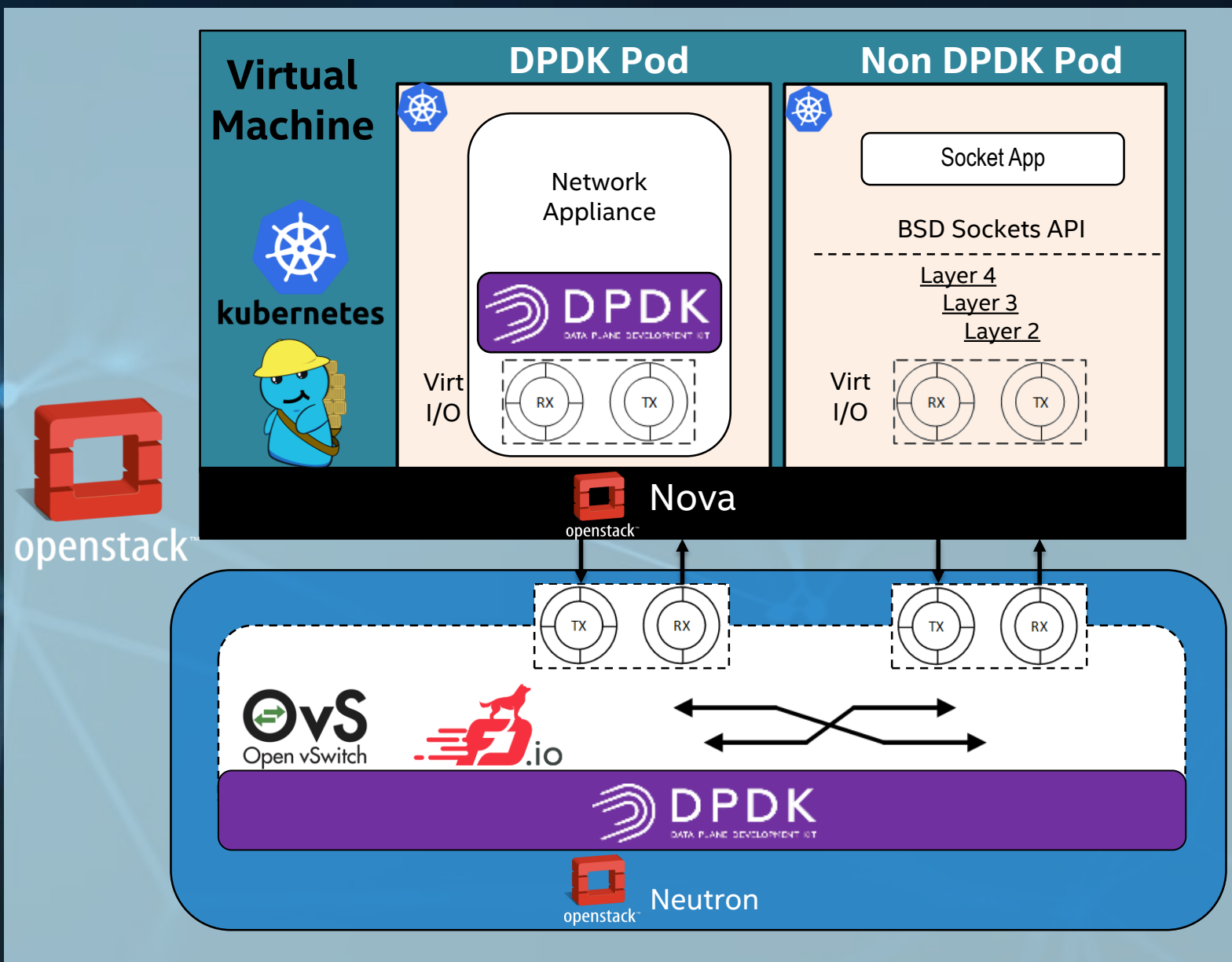
Standard Vhost shared memory interface between DPDK vSwitch and VNF.

SOLUTIONS

Enabling DPDK support in nested containers
 With Kuryr-DPDK plugin for Kubernetes

Master VM:

Co-existence of Containers and Virtual Machines
 Leverage OpenStack Infra DPDK based vSwitch to accelerate the Container Data Plane.



CALL TO ACTION

- We need feedback on the current ingredients e.g.
 - Multus
 - SR-IOV
 - Vhost user
- Be active in K8s SIGs
 - Network
 - Resource Management



MULTUS

SR-IOV



CNI



kubernetes
by Google

TALK TO US FOR YOUR CONTAINERS USE CASE?



ivan.coughlan@intel.com

kuralamudhan.ramakrishnan@intel.com

slack <https://intel-corp.herokuapp.com>

GitHub
thed00de
rkamudhan

AMAZING POWERED BY INTEL