STORAGE DEVELOPER CONFERENCE

SDC 23

BY Developers FOR Developers

Disaggregated Storage with Marvell® OCTEON® DPUs and OPI

Presented by

Satananda Burla, Senior Principal Engineer, Marvell

Forward-looking statements

Except for statements of historical fact, this presentation contains forward-looking statements (within the meaning of the federal securities laws) including statements related to future revenue, future earnings, and the success of our product releases that involve risks and uncertainties. Words such as “anticipates,” “expects,” “intends,” “plans,” “projects,” “believes,” “seeks,” “estimates,” “can,” “may,” “will,” “would” and similar expressions identify such forward-looking statements. These statements are not guarantees of results and should not be considered as an indication of future activity or future performance. Actual events or results may differ materially from those described in this presentation due to a number of risks and uncertainties.

For other factors that could cause Marvell’s results to vary from expectations, please see the risk factors identified in Marvell’s Quarterly Report on Form 10-Q for the fiscal quarter ended July 29, 2023 as filed with the SEC on August 25, 2023 and Marvell’s Annual Report on Form 10-K for the fiscal year ended January 28, 2023 as filed with the SEC on March 9, 2023, and other factors detailed from time to time in Marvell’s filings with the SEC. Marvell undertakes no obligation to revise or update publicly any forward-looking statements.

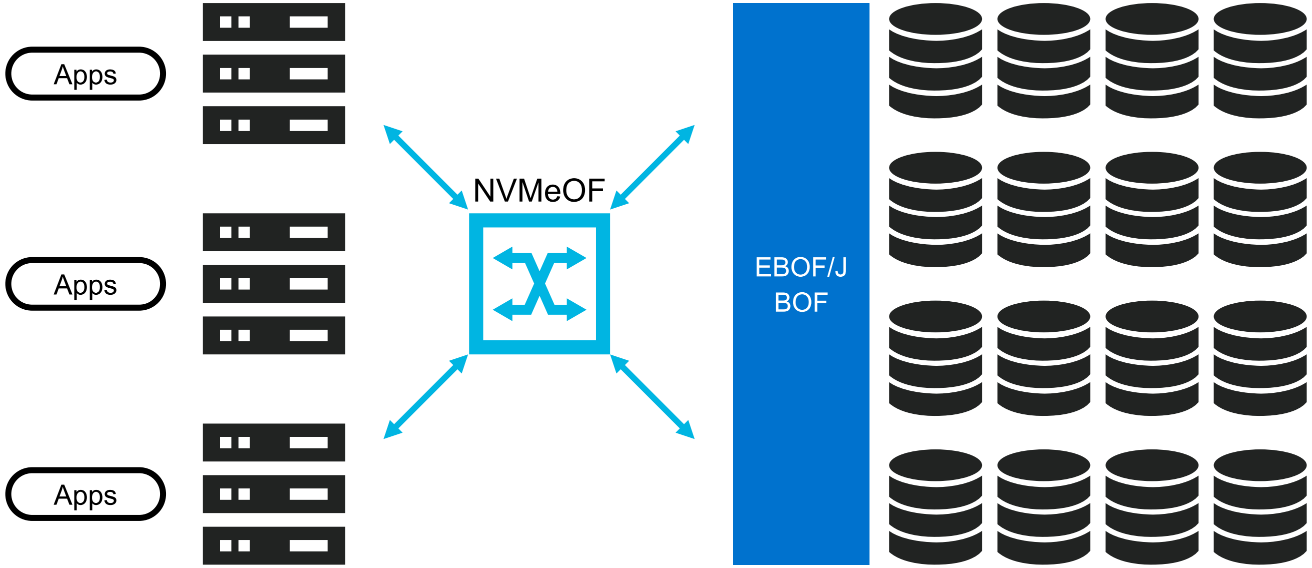
Agenda

- **Disaggregated Storage**
  - NVMeOF over smartNIC
  - Local NVMe/Virtio-Blk over NVMeOF using DPUs
- **OCTEON DPUs**
  - OCTEON DPUs of past, present and future
  - Marvell Velox SDK
  - Octeon DPU NVMe Offload Architecture
- **OPI Integration**
  - Introduction to OPI
  - Storage APIs for front end and middle end
  - Integration into orchestrators
- **Next Steps, Future Work**
  - OCTEON DPU OPI improvements
  - TCP offload options
  - More DPU services for Storage

Disaggregated Storage

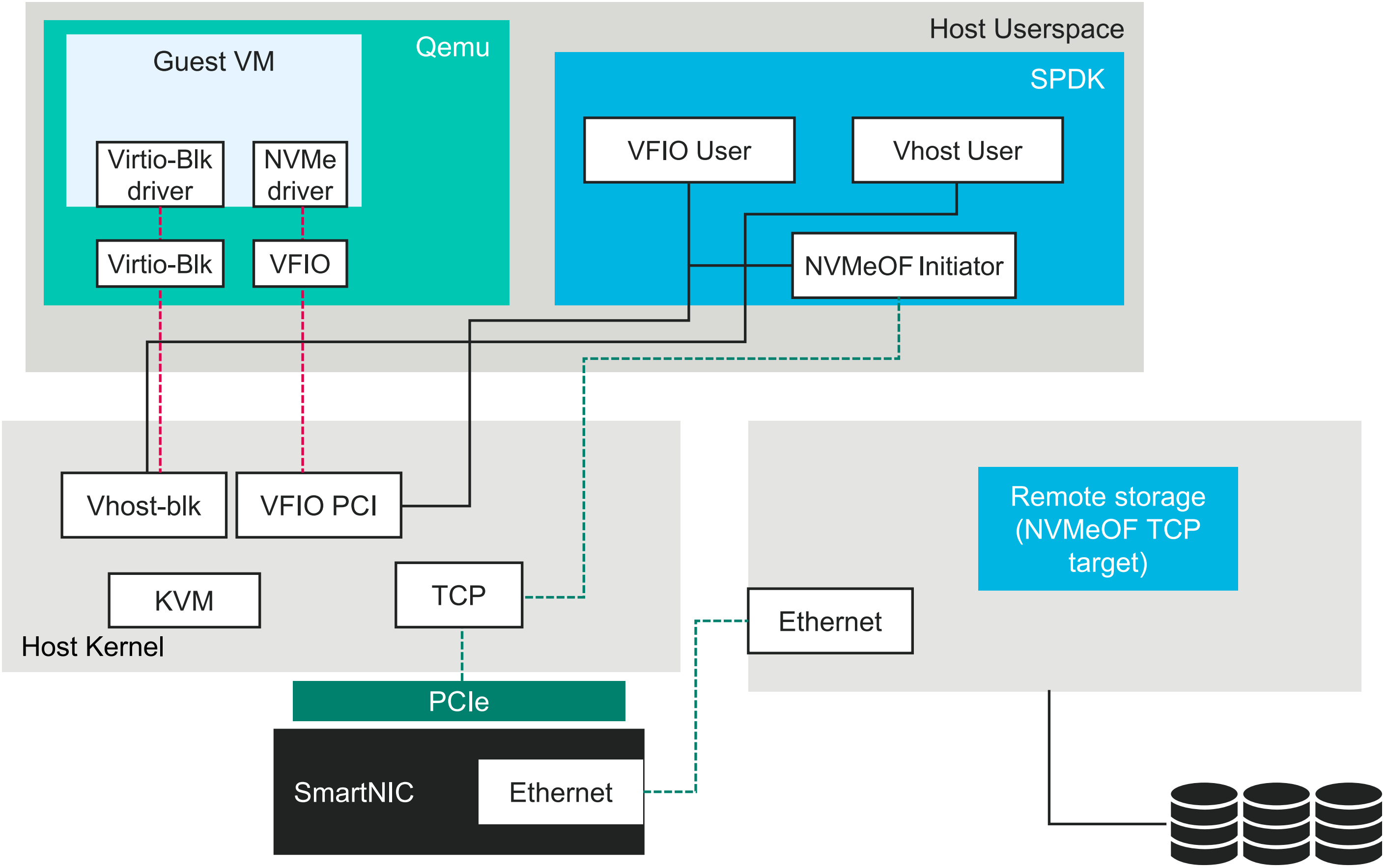
Disaggregated storage

- Compute and Storage resources are separated
- Storage be collocated in the DC or across the world
- Multiple advantages
  - Allows for scale out and greater efficiency
  - Allows for flexibility during upgrade, replace cycles
  - Does not need dedicated hardware like SANs, works using Ethernet
- NVMe provides simplistic control/data interface
- NVMeOF TCP enables storage over the most ubiquitous network protocols (TCP/IP + Ethernet)
- SmartNICs and DPUs have supported NVMeOF TCP for cloud/VM use cases
- DPUs provide additional control plane capability offloading NVMeOF initiation and termination from hypervisor



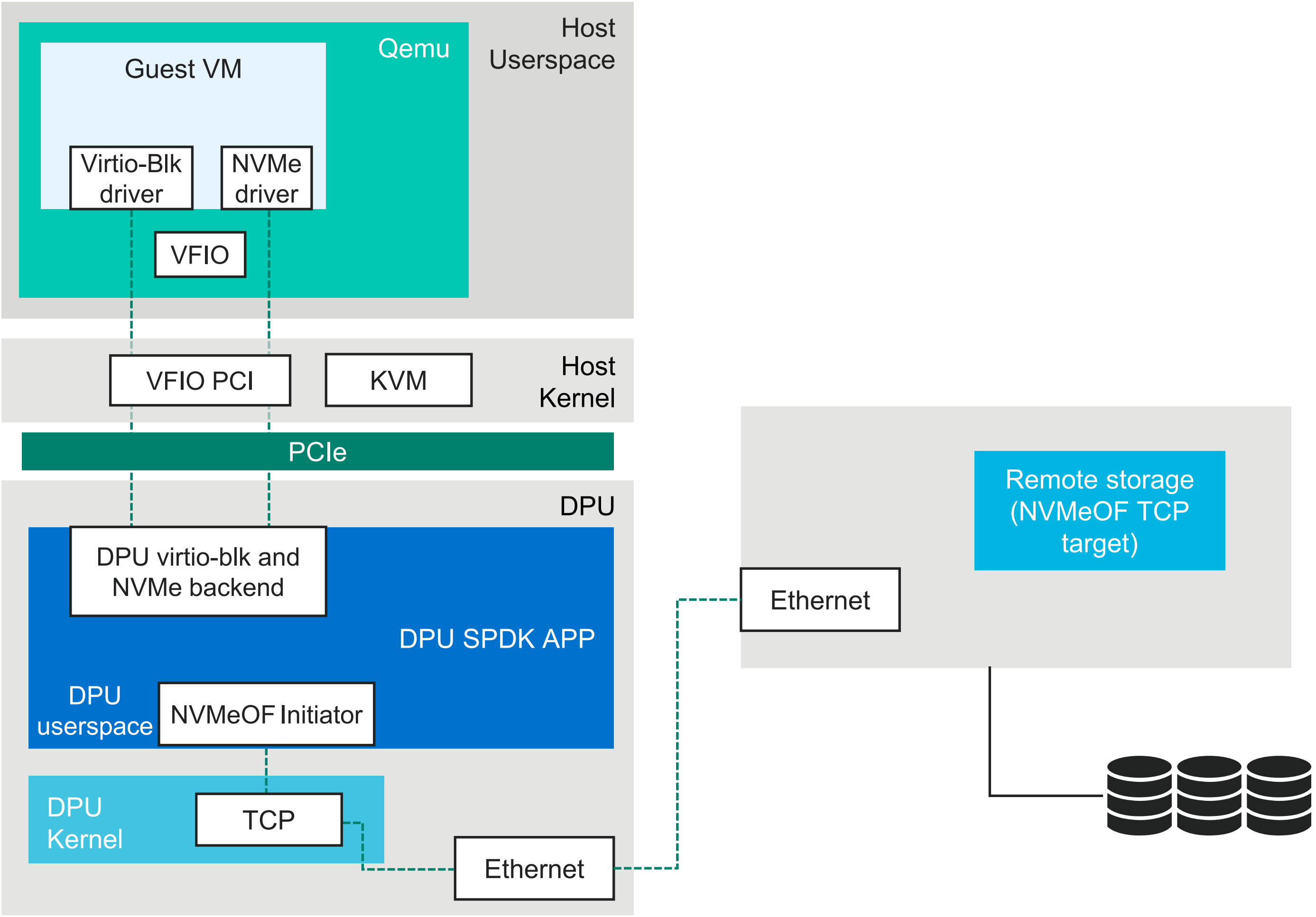
Disaggregated storage for guests with SmartNIC

- SmartNIC provides Ethernet connectivity with protocol offloads(csum/TSO/LRO)
- NVMeOF TCP protocol runs on the host either kernel/SPDK
- Virtio-blk/NVMe backend emulation has to be run on host
- CPU usage on Host which cannot be used for Tenant workloads
- QoS is limited to DCBX separation



Disaggregated storage for guests with DPU

- DPU provides Local NVMe/Virtio-Blk-PCI PF/VF devices to host
- DPU handles the Virtio-blk/NVMe backend and runs NVMeOF TCP initiator
- Guest VMs use NVMe/Virtio-Blk devices directly with passthrough
- NVMeOF initiator configuration needs to be done on DPU
- Saves Host NVMe/Virtio emulation and NVMeOF TCP cycles
- Provides better QoS separation for VMs by throttling local NVMe VFs



OCTEON DPUs

OCTEON DPUs

OCTEON® multicore

2005

Industry's 1st DPU

LiquidIO®

2010

OCTEON TX ®

2015

First Arm based DPU

OCTEON TX2 ®

2019

6th Generation Marvell DPU

OCTEON® 10

2021

7th Generation Marvell DPU

Production

- LIO4
- 36C ARMv8 TX2, CN9880
- 200G
- FHFL
- NVMe

- CN98 PCIE
- 36C ARMv8 TX2, CN9880
- 200G
- FHFL

- CN106 PCIE
- 24C ARMv9 N2, CN10624 (5nm)
- 100G
- FHHL

2023

2H

- CN103 PCIE
- 8C ARMv9 N2, CN103 (5nm)
- 100G
- HHHL

FUTURE

- Next Gen Octeon

OCTEON 10 DPU

Cores

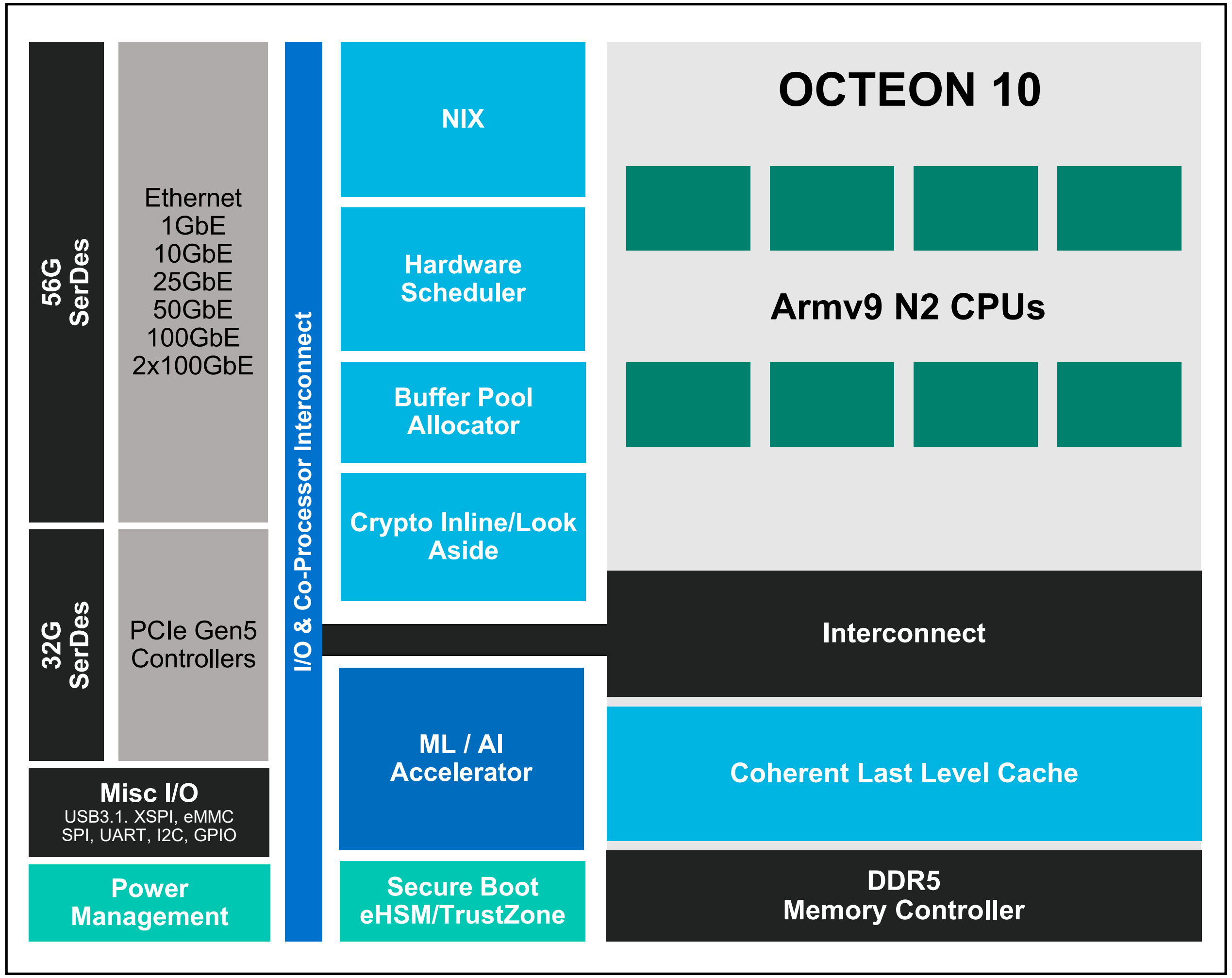
- ARMv9.0 64-bit Neoverse N2 cores
- Out-of-Order execution; Fully Virtualized, Scalable Vector Extension

Memory Subsystem and Connectivity

- Shared Last Level Cache
- DDR5 w/ sideband-ECC and Memory Encryption
- 100G/50G/25G, 10G/QSGMII
- PCIe Gen5, controllers

HW Acceleration

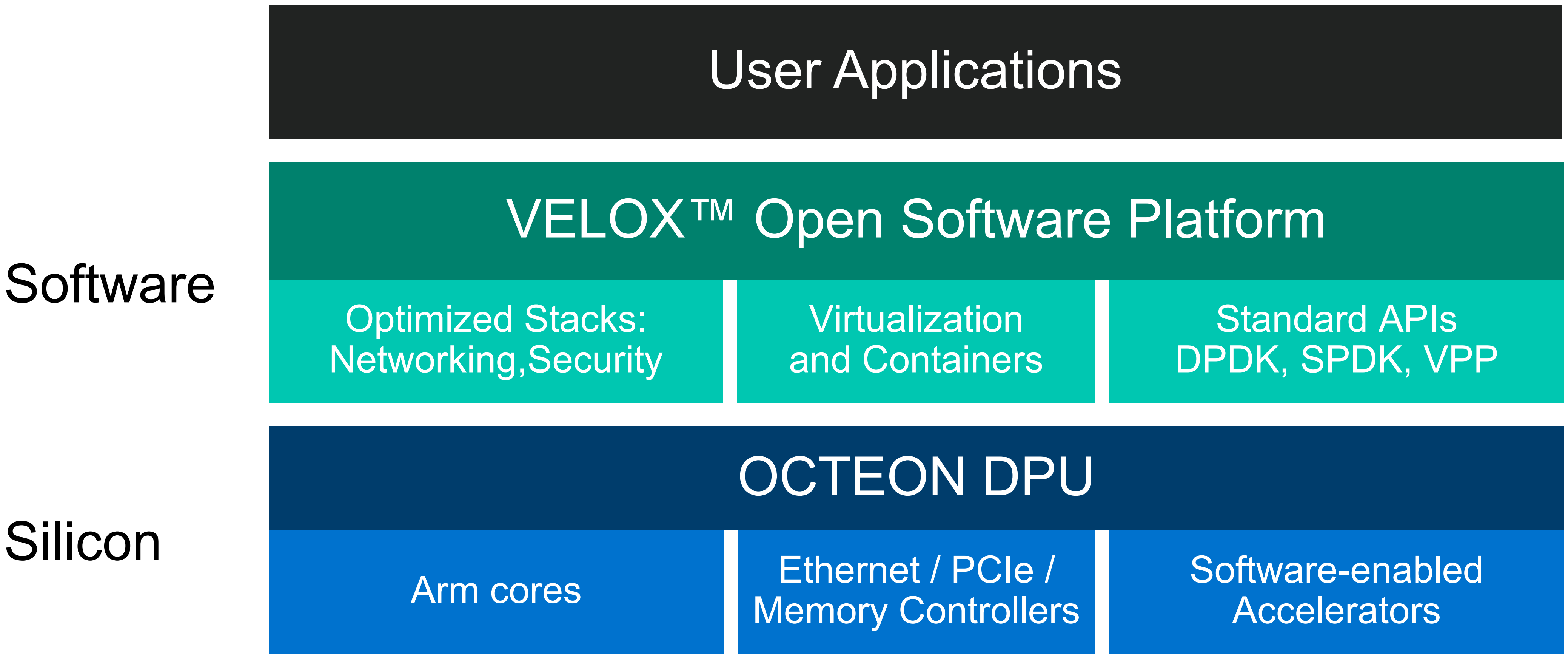
- Highly-virtualized, software-friendly NIC
- Packet Processing, QoS, Hierarchical queues with shaper & WDRR scheduler
- Multi-level header packet parsing
- Schedule, Synch., & Ordering
- Packet Processing Through PCI-e End Point
- Inline & Co-processor Security (SSL/IPSec)
- Inline ML inferencing engine
- Secure Boot + embedded Hardware security module



Next gen OCTEON DPU Storage enhancements

- Integrated NVMe and Virtio-Blk front end eliminating the need for separate NVMe accelerator SoC
- Up to **2K** NVMe and/or Virtio-blk SRIOV VFs **8K+** Queue Pairs
- Hotplug support for NVMe and Virtio
- PASID and SIOV support
- PCIe Gen6

OCTEON DPU platform



Software

Silicon

User Applications

VELOX™ Open Software Platform

Optimized Stacks: Networking,Security

Virtualization and Containers

Standard APIs DPDK, SPDK, VPP

OCTEON DPU

Arm cores

Ethernet / PCIe / Memory Controllers

Software-enabled Accelerators

VELOX™ SDK highlights

DPU software stacks

- Networking (DPDK, IPDK/PNA, native VPP, P4, PFC, QoS)
- Security (OPTEE, TLS, IPSEC, OpenSSL, Symmetric and Asymmetric Crypto)
- Storage (NVMeOF TCP, SPDK, compression, crypto)

Workload acceleration

- OVS, VPP, CNI and CSI offload
- IPSEC and TLS Offload
- Common transport interface, open offload

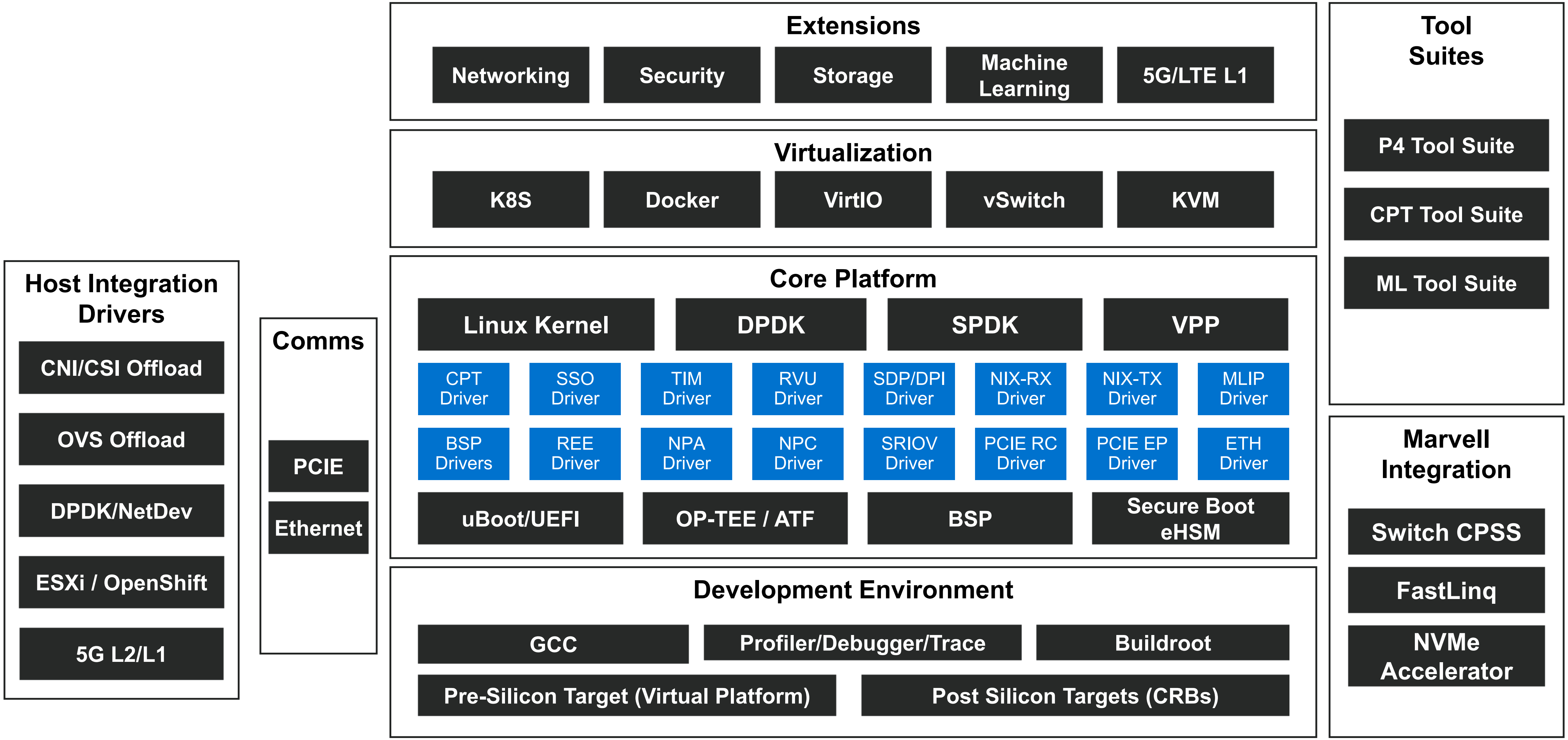
Machine learning

- TVM compiler, expanded operators and efficiency, network/security use cases

Ecosystem and Opensource integration

- Distro integration (RHEL, SUSE, Wind River)
- Management and control (BMC, MCTP, Redfish)
- Cloud native platform integration (RedHat, OpenShift, Wind River OCP, VMWare ESXi)

VELOX™ SDK for OCTEON DPU



Host Integration Drivers
- CNI/CSI Offload
- OVS Offload
- DPDK/NetDev
- ESXi / OpenShift
- 5G L2/L1

Comms
- PCIE
- Ethernet

Extensions
- Networking
- Security
- Storage
- Machine Learning
- 5G/LTE L1

Virtualization
- K8S
- Docker
- VirtIO
- vSwitch
- KVM

Core Platform
- Linux Kernel
- DPDK
- SPDK
- VPP
- CPT Driver
- SSO Driver
- TIM Driver
- RVU Driver
- SDP/DPI Driver
- NIX-RX Driver
- NIX-TX Driver
- MLIP Driver
- BSP Drivers
- REE Driver
- NPA Driver
- NPC Driver
- SRIOV Driver
- PCIE RC Driver
- PCIE EP Driver
- ETH Driver
- uBoot/UEFI
- OP-TEE / ATF
- BSP
- Secure Boot eHSM

Development Environment
- GCC
- Profiler/Debugger/Trace
- Buildroot
- Pre-Silicon Target (Virtual Platform)
- Post Silicon Targets (CRBs)

Tool Suites
- P4 Tool Suite
- CPT Tool Suite
- ML Tool Suite

Marvell Integration
- Switch CPSS
- FastLinq
- NVMe Accelerator

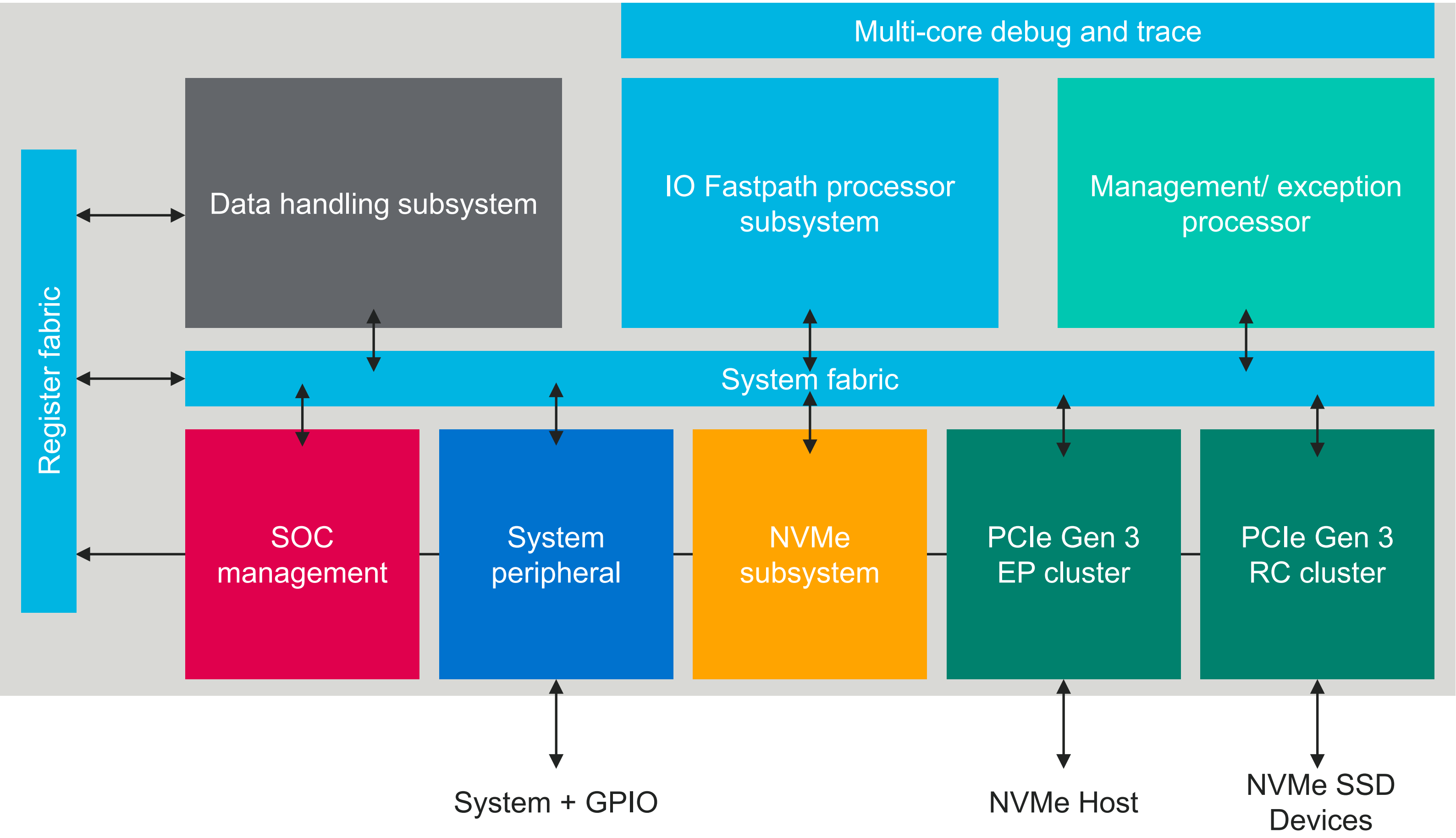
NVMe Offload Architecture

CN98XX DPU

Features	Capability
I/O	2 x 200G PAM4 PCIe Gen3 x16, bifurcated as two x8
Memory	4 CH x 8GB DDR4@ 3200MTs with ECC, 32GB total
ARM cores	36 ARM V8.2, 2.2GHz, 360 SPECINTRate
Performance	220 MPPS, 200Gbps
Local NVMe	64 NVMe VFs, 1.4M iops, 256 QPs (per SQ QoS)
NVMe over TCP	SPDK with 200Gbps TCP with DCBX support
IPSEC, RSA 2K, 1KB OpenSSL, TLS1.3 support	200Gbps IPSEC, 90Kops RSA 2K, 200Gbps 1KB OpenSSL

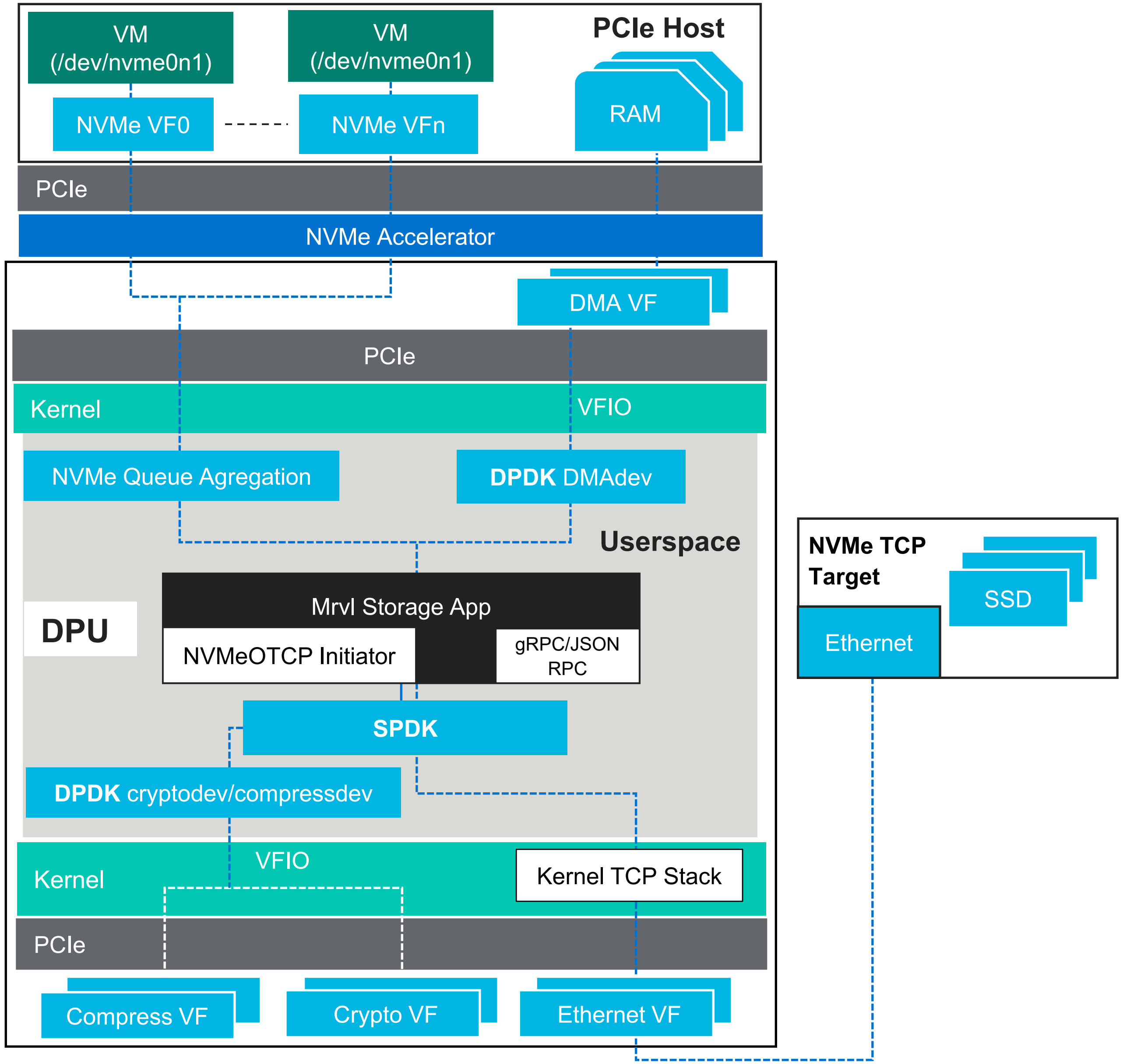
Marvell 88NR2241 NVMe storage accelerator

Interface FE	PCIe Gen3x8 (2x4, 1x8)
Interface BE	PCIe Gen3x8 (4x2 or 2x4)
4k RR IOPS	1.2M (IOV performance)
Multi-VF/PF*	32/16
Typical Power	5W
SW Feature	QoS / IO Metering / Management
DRAMless	Yes
Silicon Status	Production



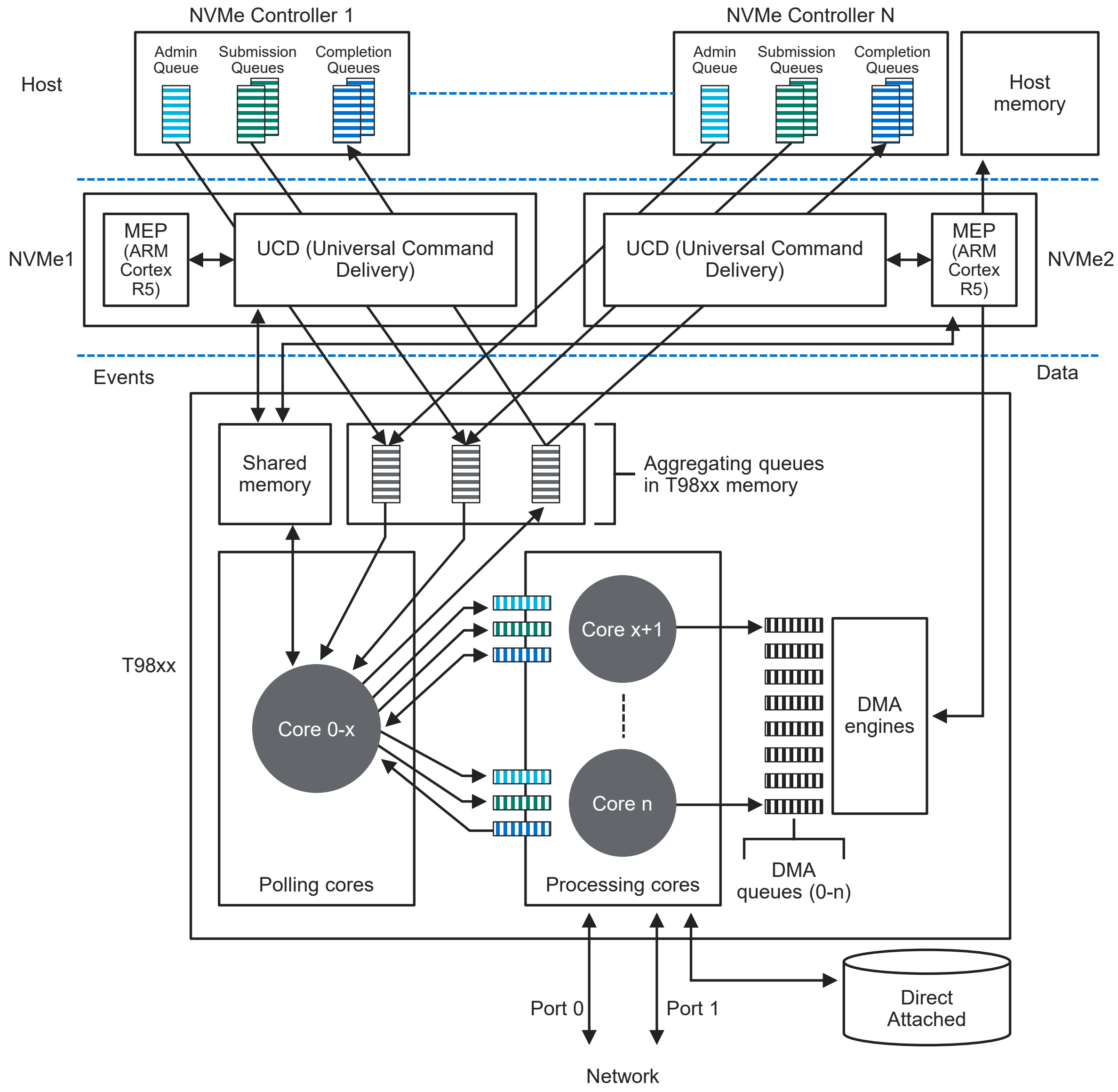
CN9XX DPU NVMe

- Provides Local NVMe controllers that can be attached to VMs with backend provided by SPDK based bdevs
- 100 Gbps/1.4M iops
- Upto 64 NVMe SRIOV VFs and 256 queue pairs
- NVMe over TCP TLS or AES_XTS offloaded using Crypto block
- Compression can be offloaded using Compress block
- QoS per SQ, per NVMe VF and VM
- Cloud integration with Mrvl Json RPC/OPI gRPC (more on this later)



Software architecture

- Each NVMe accelerator presents 1 PF and 32 VFs
- MEP (Management/Exception Path Processor) is an Arm® Cortex R5 processor for managing control plane
- UCD (Universal Command Delivery): The UCD engine (Universal Command Delivery) is responsible for fetching the NVMe commands from host and posting the completions back to host
- Storage Application: Is an SPDK application which polls for admin and submission queue commands, completes the commands using NVMeOF TCP, DMAs data and posts completions through completion queues



Software arch continued... Read request walkthrough

- Host posts an NVMe read request through one of the IO submissions queues and hits the doorbell.
- UCD observes that a new request is available, fetches the request and posts it in the submission aggregating queue in T98xx memory.
- CN98XX core notices a new entry in the aggregating submission queue, fetches the entry from the queue and pushes it to the next available processing core.
- CN98XX processing core processes the command and posts this request to SPDK using bdev APIs. SPDK stack processes this command and takes required actions (for example, reads the data from a direct attached storage or posts request to Networked storage).
- Once a response is received, CN98XX DMA engines are used to push this data to the host memory.
- Once the DMA operation is complete, core posts a completion entry in aggregating completion queue in CN98XX memory.
- UCD notices a new entry in completion queue and posts this to the corresponding host completion queue. UCD can also interrupt the host on posting completion queue entry.

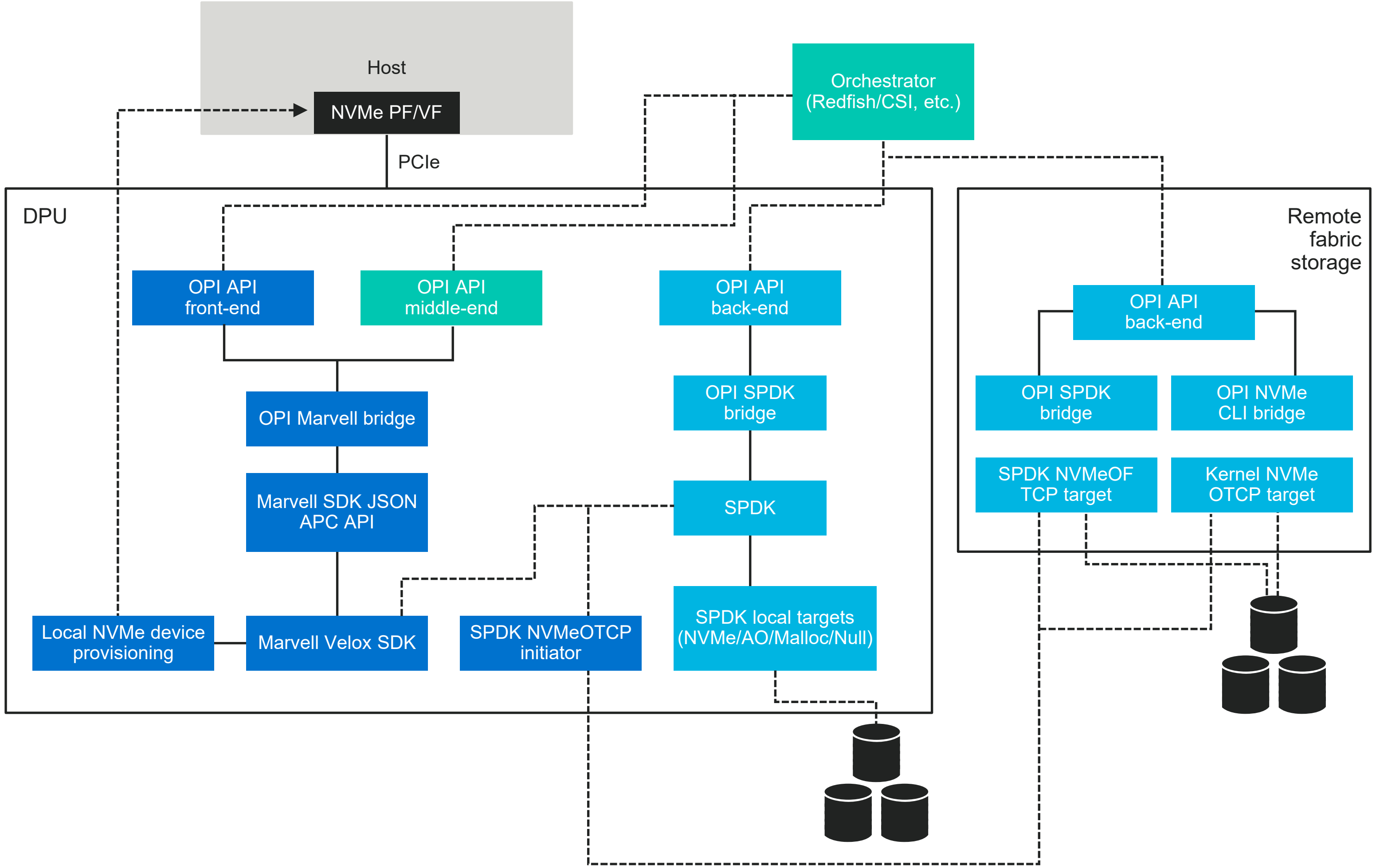
OPI integration

Marvell DPU NVMe provisioning

- Marvell Velox SDK started off with providing low level C API for NVMe PF/VF provisioning for host
- This was contributed to OPI after Marvell joined OPI
- Provides interface on DPU for enabling NVMe local PFs/Vfs to host with namespaces being added by running SPDK based initiator on DPU
- SPDK json-rpc used for creating subsystems, namespaces and associating local namespace with remote namespace
- Marvell has since moved to using gRPC with OPI-MARVELL bridge and OPI-SPDK bridge

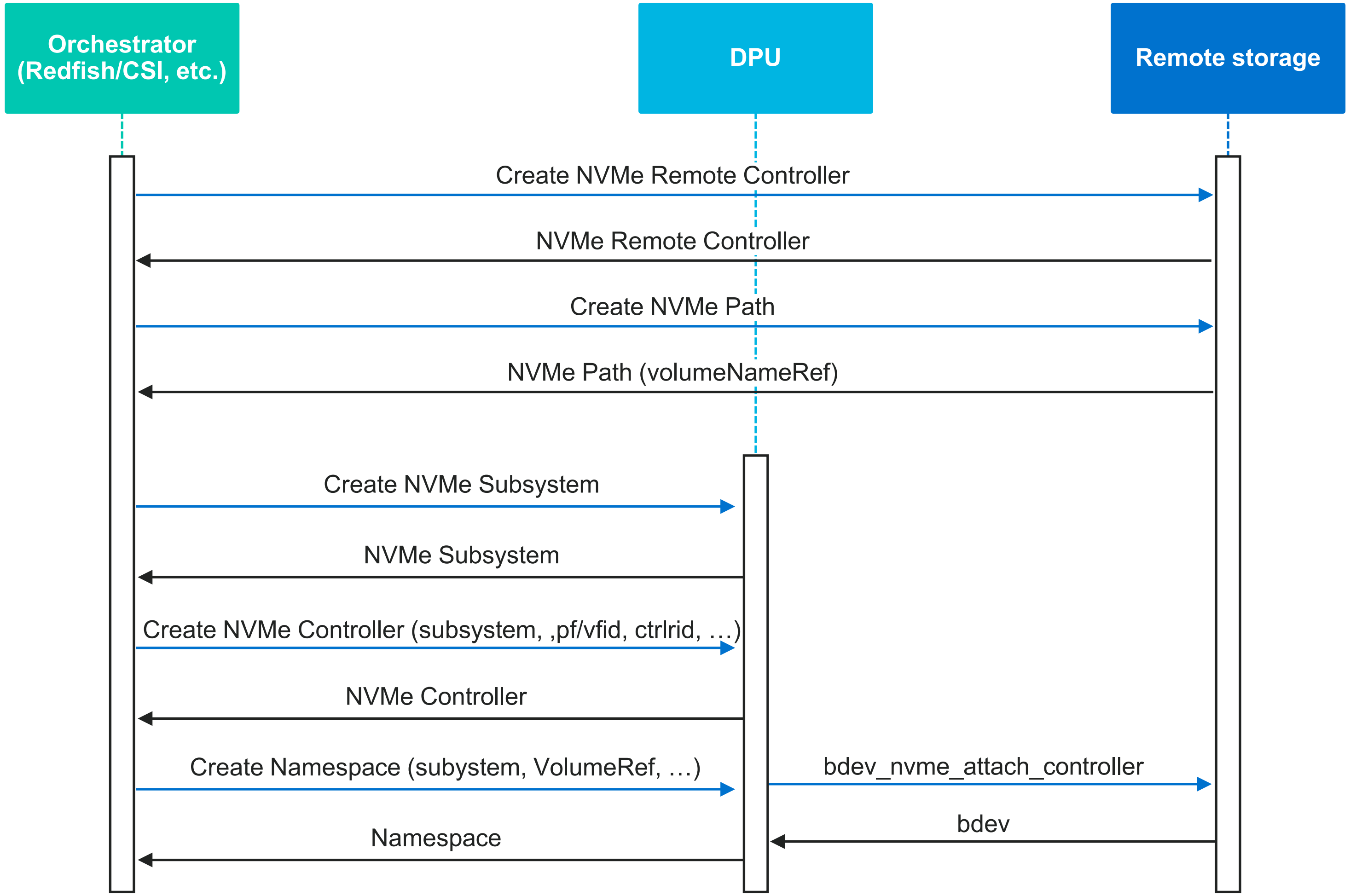
OPI-Marvell bridge and OPI-SPDK bridge

- **OPI-Marvell Bridge**
  - Converts OPI API calls to Marvell Velox SDK specific API for NVMe local PCIe device provisioning
  - Reuses SPDK API for NVMeOF Initiator
- **OPI-SPDK Bridge**
  - Converts OPI gRPC to SPDK Json RPC calls for NVMeOF provisioning



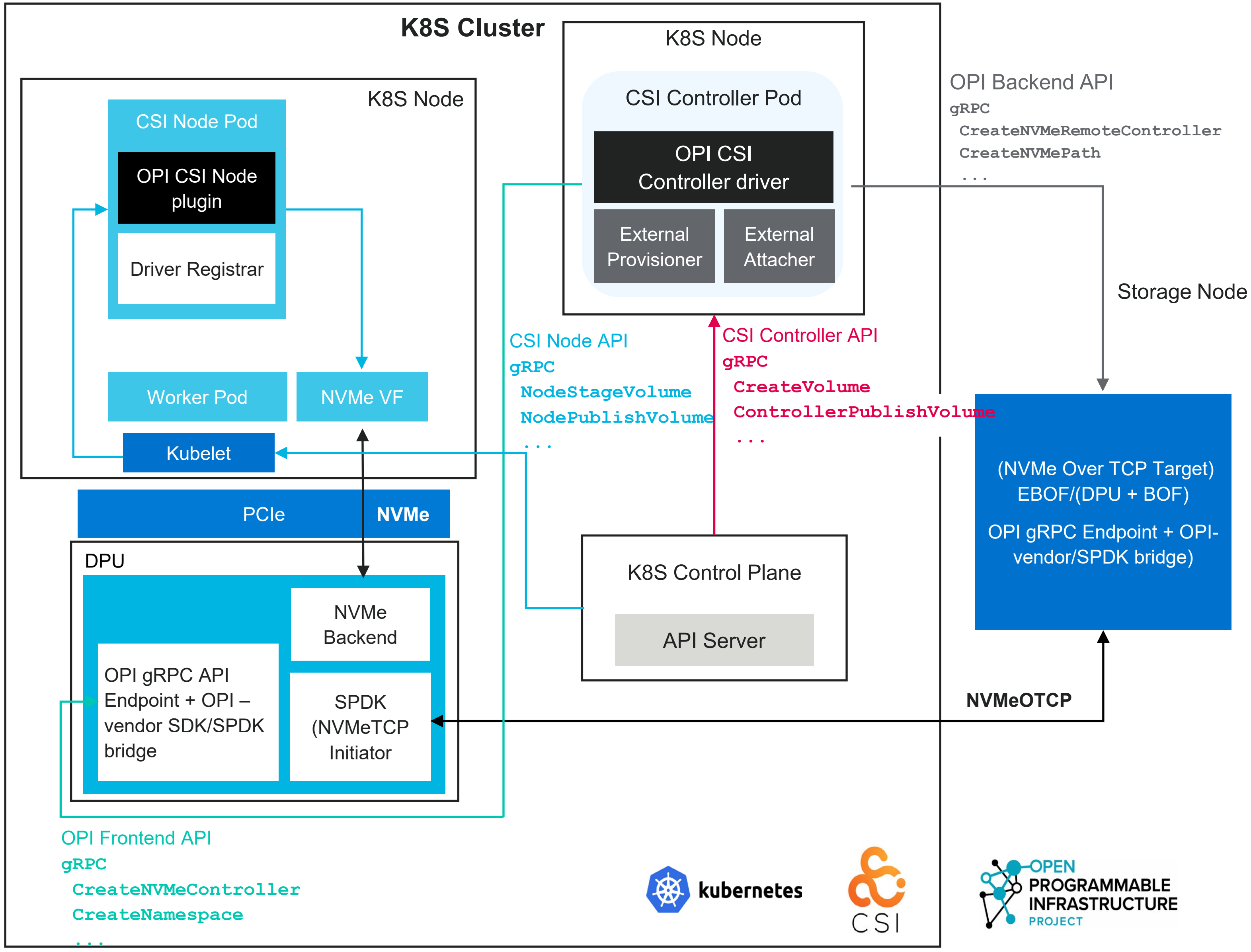
Provisioning workflow

- OPI storage
  - Frontend
    - Provisioning the DPU NVMe interfaces for the host
    - Associating local namespaces to remote volumes
  - Middle end
    - Provisioning QoS/Compression/Encryption
  - Backend
    - Provisioning remote volumes (creating and publishing volumes)


Orchestrator (Redfish/CSI, etc.)
DPU
Remote storage
Create NVMe Remote Controller
NVMe Remote Controller
Create NVMe Path
NVMe Path (volumeNameRef)
Create NVMe Subsystem
NVMe Subsystem
Create NVMe Controller (subsystem, ,pf/vfid, ctrlrid, …)
NVMe Controller
Create Namespace (subystem, VolumeRef, …)
bdev_nvme_attach_controller
bdev
Namespace

K8S OPI Integration

- OPI group working on a CSI driver
- OPI provides CSI node component and CSI controller component
- DPU vendors provide OPI API to Vendor SDK translation



K8S Cluster

K8S Node

CSI Node Pod

OPI CSI Node plugin

Driver Registrar

Worker Pod

NVMe VF

Kubelet

PCIe

NVMe

DPU

NVMe Backend

OPI gRPC API Endpoint + OPI – vendor SDK/SPDK bridge

SPDK (NVMeTCP Initiator

OPI Frontend API

```
gRPC
 CreateNVMeController
 CreateNamespace
 ...
```

K8S Node

CSI Controller Pod

OPI CSI Controller driver

External Provisioner

External Attacher

CSI Node API

```
gRPC
 NodeStageVolume
 NodePublishVolume
 ...
```

CSI Controller API

```
gRPC
 CreateVolume
 ControllerPublishVolume
 ...
```

K8S Control Plane

API Server

OPI Backend API

```
gRPC
 CreateNVMeRemoteController
 CreateNVMePath
 ...
```

Storage Node

(NVMe Over TCP Target) EBOF/(DPU + BOF)

OPI gRPC Endpoint + OPI-vendor/SPDK bridge)

NVMeOTCP

kubernetes

CSI

OPEN PROGRAMMABLE INFRASTRUCTURE PROJECT

Next steps/Future work

OCTEON DPU OPI improvements

- Working on direct OPI gRPC to Marvell low level API conversion to avoid intermediate JSON-RPC
- Support for Middle end services on Marvell DPU
- Support for PCIe hotplug for composability
- OPI CSI integration
- Performance improvements

TCP offload options

- How to avoid DPU DDR bouncing
  - Want to stick with Linux Kernel TCP stack - We all know the issues of custom TCP stacks
  - Have to run NVMeOF and TCP header processing in kernel and keep payload away
  - Split header data approach
  - Kernel can consume TCP header
  - Userspace/kernel deals with PDU header and enables DMA
  - Ability to DMA between DPU NIC SRAM buffers and Host buffers
  - Userspace needs to setup DMA from DPU NIC buffers to Host and vice versa.
  - Multiple attempts happening in kernel mailing lists for enabling p2p network bufffers
    - https://lists.openwall.net/netdev/2023/07/11/1
    - https://lore.kernel.org/netdev/6376CA34-BC6F-45DE-9FFD-7E32664C7569@fb.com/t/

More storage services by DPUs

- **Redundancy**
  - Erasure coding
  - RAID 5/6
  - Can these be added as OPI backend services for storage directly attached to DPU?
- **Computational Storage**
  - DPUs can provide computational storage capabilities with DPU + BOF
  - NVMe coming up with computational storage command set.
  - What will be OPI's role in enabling Computational storage specifically for providing this service where the DPU is acting as an intermediate hop ?
- **AI/ML with DPUs that can run inferencing on pretrained models**
  - AI/ML has ubiquitous use cases in every domain (storage is no exception)
  - To predict read/write patterns for caching (CDN, Key value stores, object storage etc)
  - To predict lifetimes, to reduce write amplification in SSDs, to predict and warn about wear leveling, remaining PE cycles, bad blocks etc.
  - How OPI API can help in enabling usage of DPU provided inferencing capabilities?

Thank You.

Questions?

MARVELL™

Essential technology, done right™

Please take a moment to rate this session.

Your feedback is important to us.