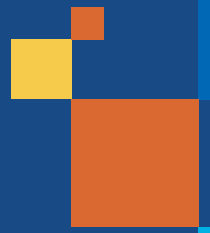
Storage Performance Development Kit (SPDK)
Persistent Memory Development Kit (PMDK)
Intel® VTune™ Profiler

Virtual Forum

Time to change, SPDK VHOST evolutional solution

Changpeng Liu, DPG/Intel



intel®

Introduction to vhost and vfio-user

Virtio devices

- Lots of device types
- Guest device driver
- QEMU emulation
- Filesystem devices
- Block devices

  BLK: virtio/vhost-user in QEMU

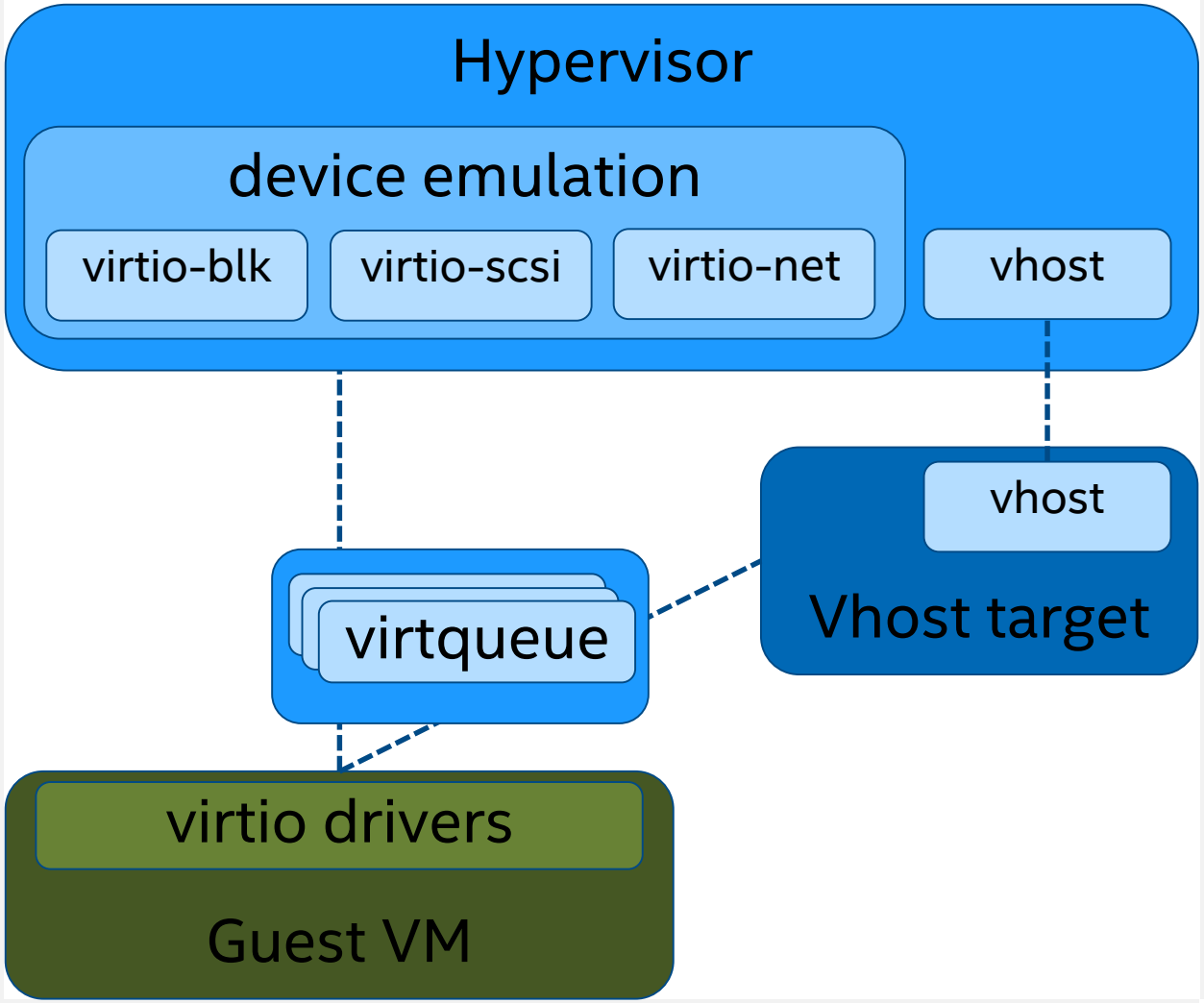
  SCSI: virtio/vhost/vhost-user in QEMU

Device ID	Virtio Device
0	reserved (invalid)
1	network card
2	block device
3	console
4	entropy source
5	memory ballooning (traditional)
6	ioMemory
7	rpmsg
8	SCSI host
9	9P transport
10	mac80211 wlan
11	rproc serial
12	virtio CAIF
13	memory balloon
16	GPU device
17	Timer/Clock device
18	Input device
19	Socket device
20	Crypto device
21	Signal Distribution Module
22	pstore device
23	IOMMU device
24	Memory device

Device types defined in virtio specification

VIRTIO & VHOST

- Para virtualized driver specification
- Common mechanisms and layouts for device discovery, I/O queues, etc.
- vhost protocol for communicating guest VM:
  - memory
  - number of virtqueues
  - virtqueue locations



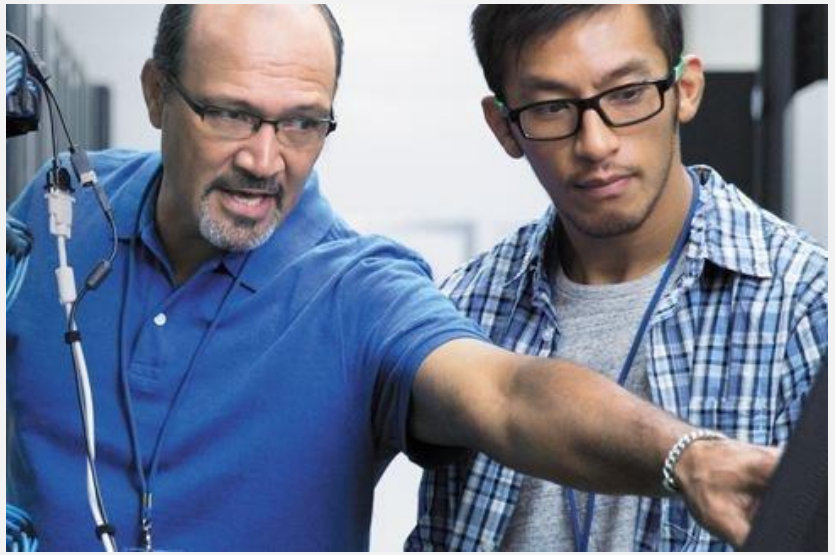
Vhost summary

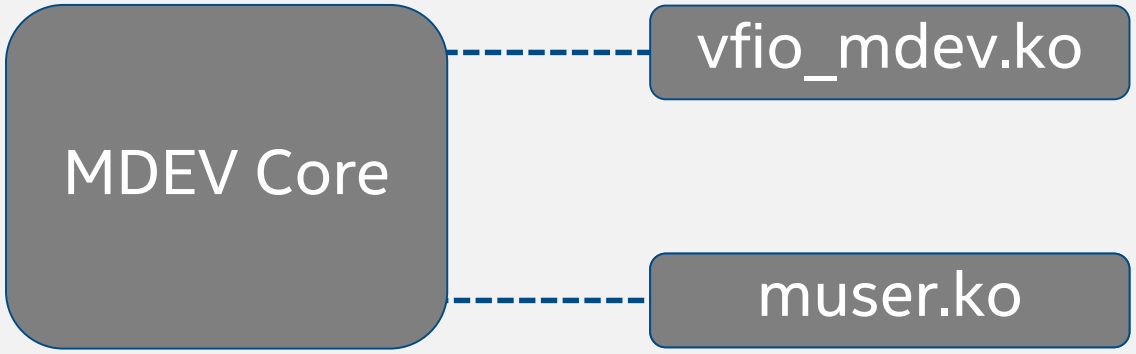
- Device emulation in vhost target
- Device driver in QEMU communicate with vhost target via socket or ioctl
- Virtio devices only
- Each device type needs a driver in QEMU

A better way?

VFIO AND VFIO MEDIATED DEVICES

- Mediated core driver provides an interface for device management that can be used by drivers of different devices
- VFIO provides unified APIs for direct device access
- Mediated device does not have to be backed by a physical device
- Muser is only a communication channel, can we have another implementation?





Vfio APIs

Start	End	Name	Type
00h	3Fh	PCI Header	
PMCAP	PMCAP+7h	PCI Power Management Capability	PCI Capability
MSICAP	MSICAP+9h	Message Signaled Interrupt Capability	PCI Capability
MSIXCAP	MSIXCAP+Bh	MSI-X Capability	PCI Capability
PXCAP	PXCAP+29h	PCI Express Capability	PCI Capability
AERCAP	AERCAP+47h	Advanced Error Reporting Capability	PCI Express Extended Capability

- PCI Header
  - Vendor ID/Device ID/Class Code
- MSI-X Capability
  - Enable/Disable
  - MSI-X Table
- BAR
  - MMAP can be supported based on offset
  - Device specific definition

Vfio-user

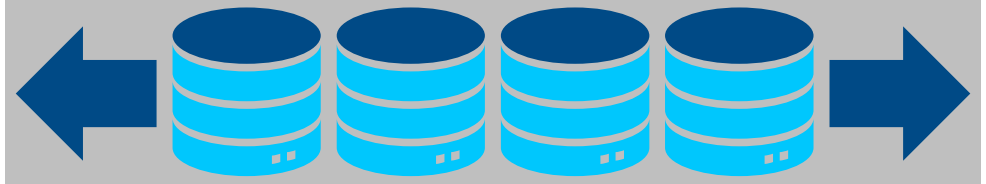
- Unified interface based on PCI semantics
- Thin library in hypervisor, eliminate QEMU zoo issue
- Multiple clients, QEMU/Rust VMM/SPDK
- Similar with vhost-user solution can provide zero copied fast data path

SPDK MUSER target based on vfio-user

Today's standard

VIRTIO-BLK

Simple read/write/flush/discard /write zeroes commands



VIRTIO-SCSI

Full SCSI command set
Multiple disks per virtio-scsi device

NVM Express

Benefit from NVMe interface

- Industry specification
- Software ecosystem

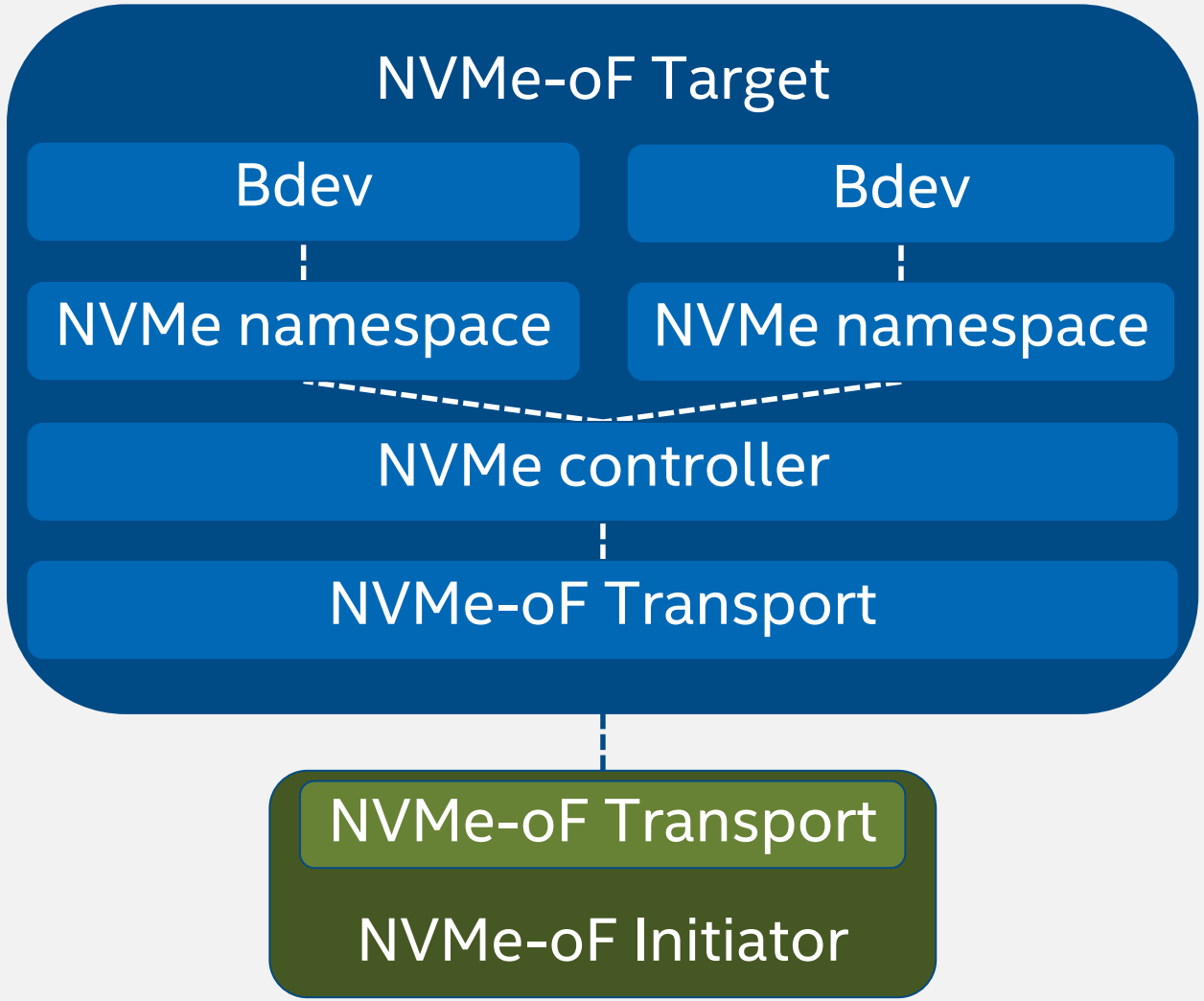
Current specifications

Advantest America
Alibaba (China) Co., Ltd.
Apeiron Data Systems
Apple Inc.
Attala Systems
Avery Design Systems
Beijing Memblaze Technology Co. Ltd.
Broadcom
Cadence Design Systems
Chelsio Communications, Inc.
CNEX Labs, Inc.
DERA Co., Ltd.
Eideticom
FADU Inc.
G2M Communications Inc.
Google. Inc.
Hagiwara Solutions Co., Ltd
HP, Inc.
Huawei Technologies Co., Ltd.
IBM
Inspur Electronic Information Industry Co., Ltd.
Kalray, Inc.
Lite-On Technology Corporation
MediaTek Inc.
Microchip Technology Inc
NVIDIA
Oracle America Inc.
Pensando Systems Inc.
Phison Electronics Corp.
Pliops Ltd.
Pure Storage, Inc.
Realtek Semiconductor Corp.
ScaleFlux, Inc.
Silicon Motion
SK hynix
Solarflare Communications, Inc.
StarWind
Teledyne LeCroy
ULINK Technology, Inc.
UNIC Memory Technology Co., Ltd.
VIA Technologies, Inc
VMware, Inc.
Xilinx, Inc.
Yangtze Memory Technologies Co., Ltd.
Zettastone Technology Co., Ltd.

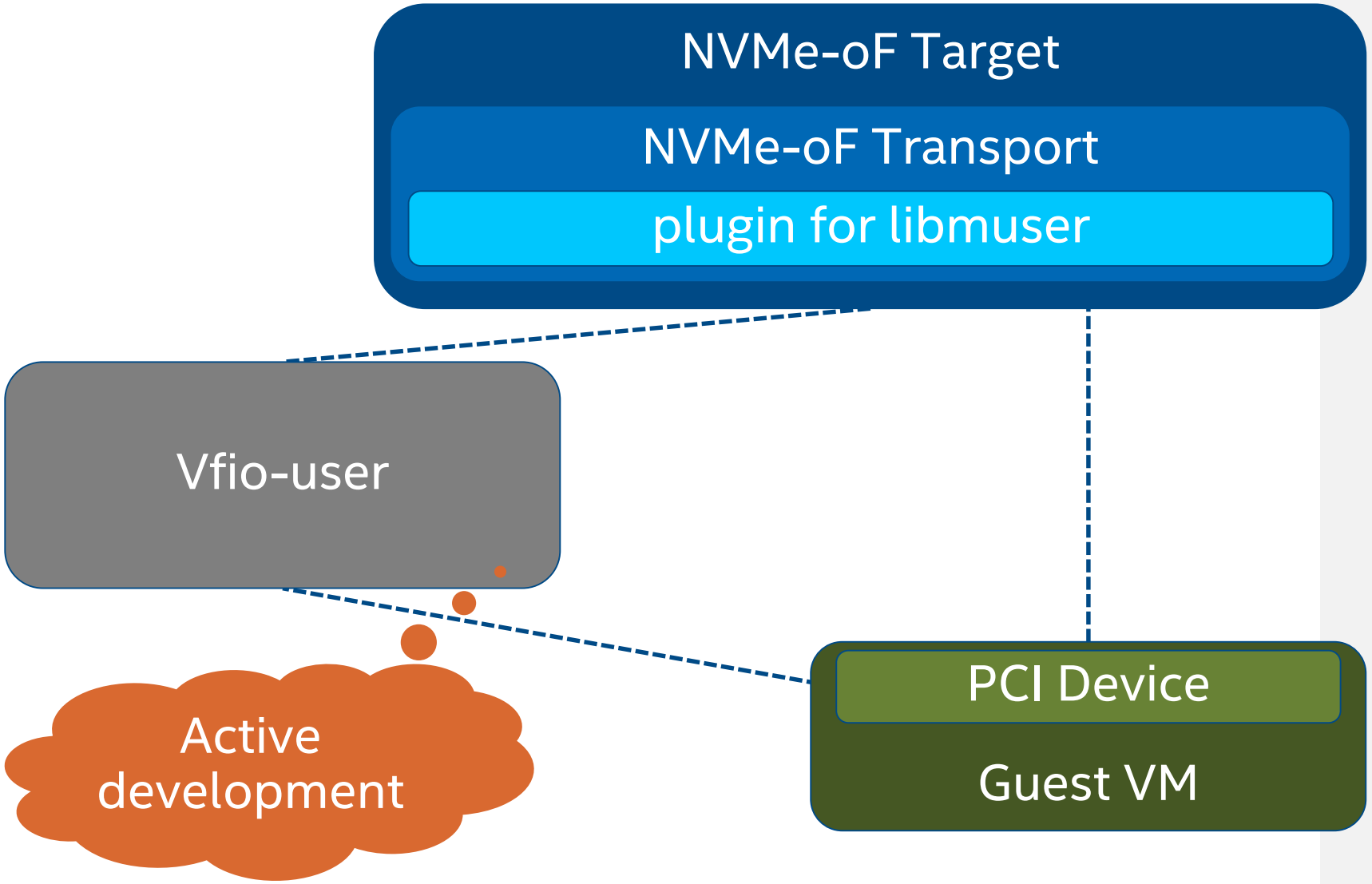
Membership list

NVME DEVICE EMULATION

- Attaching any bdev device to NVMe namespace
- Emulation of NVMe controller
- Exposing controller via Transport

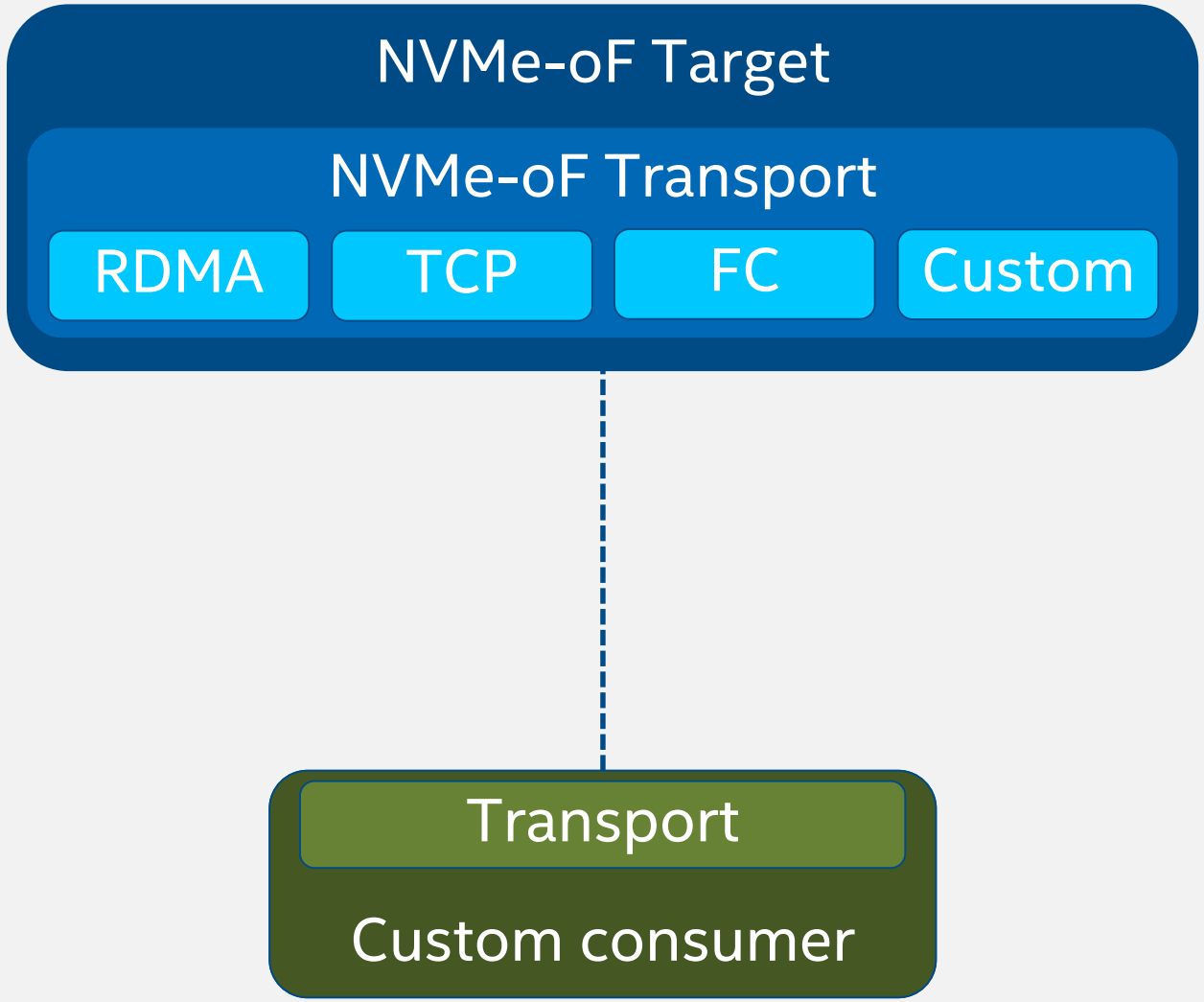


Vfio-user


NVMe-oF Target
NVMe-oF Transport
plugin for libmuser
Vfio-user
Active development
PCI Device
Guest VM

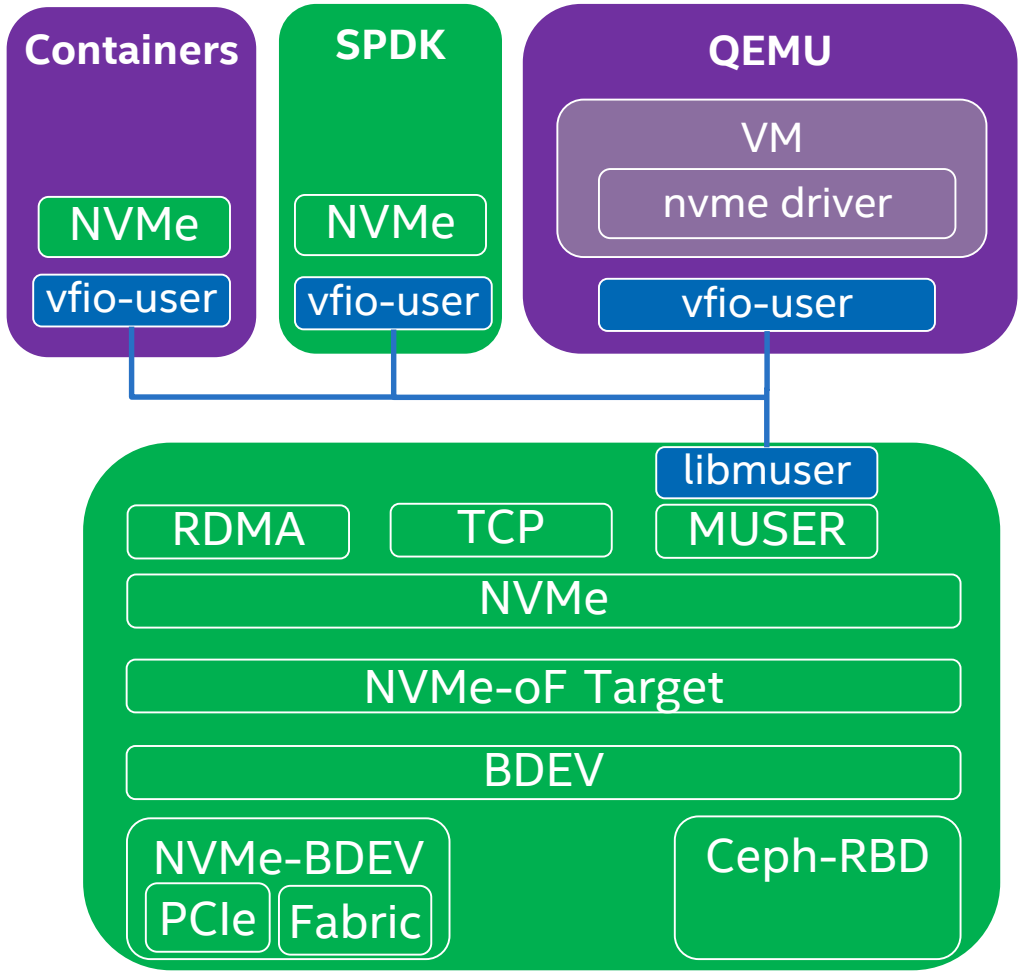
NVME-OF TRANSPORTS

- Compatiblity between Target and Initiator with common Transport
- *Fabric* Transports
- *Non-fabric* Transports



NVMe device model usage scenarios

- VM
- SPDK NVMe driver
- Containers



Usage

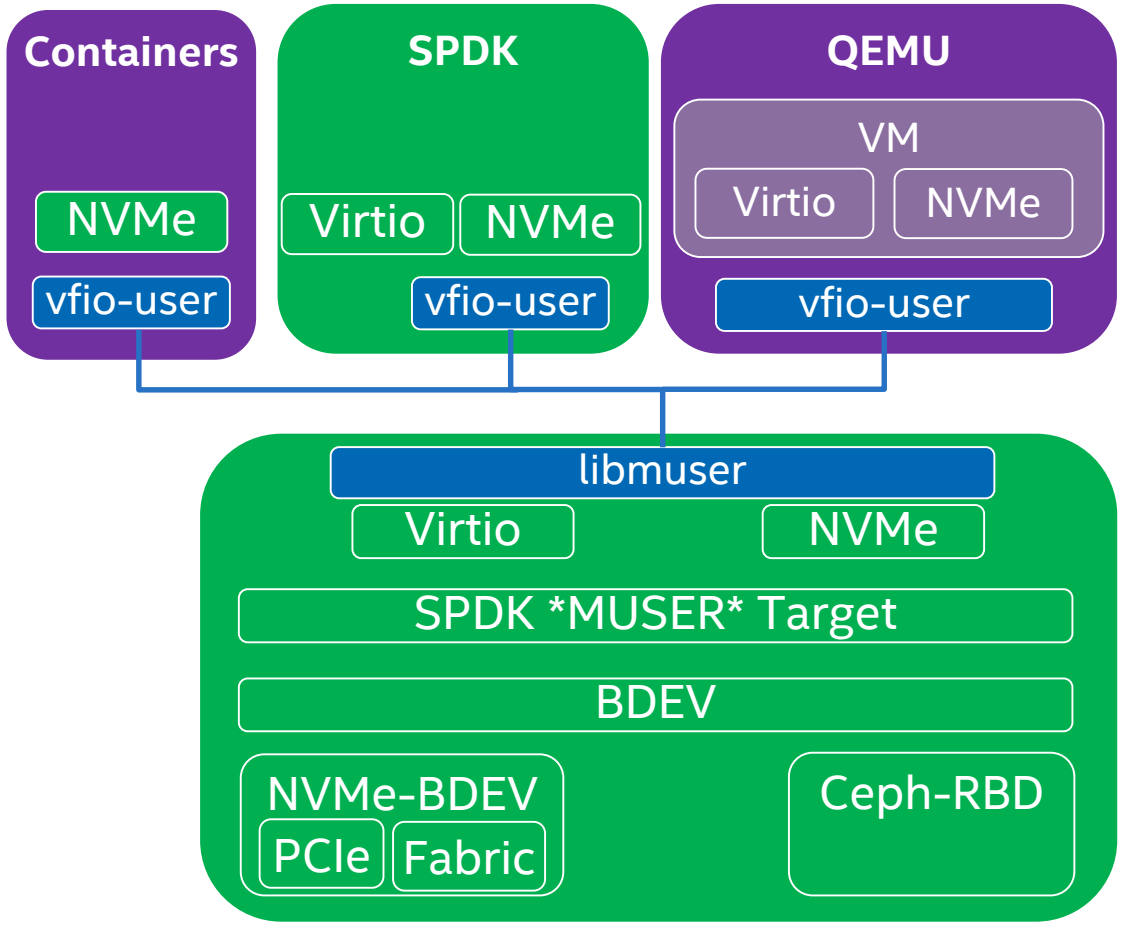
- QEMU VFIO: “-device vfio-pci,sysfsdev=/sys/bus/pci/devices/0000:86:00.0”
- QEMU VFIO-MDEV: “-device vfio-pci,sysfsdev=/sys/bus/mdev/devices/83b8f4f2-509f-382f-3c1e-e6bfe0fa1001”

- QEMU VFIO-USER: “-device vfio-user-pci,socket=/var/run/muser/domain/muser0/8”
- SPDK: “trtype:CUSTOM traddr:/var/run/muser/domain/muser0/8”

Summary

SPDK VHOST evolution in future

- Unified SPDK Target
- VM with different device types
- SPDK NVMe/Virtio driver
- Containers



Summary

- Vfio-user solution
- SPDK Muser target to provide NVMe emulation to VM and SPDK NVMe initiator driver

Patches for evaluation

- https://github.com/nutanix/muser, libmuser
- https://review.spdk.io/gerrit/c/spdk/spdk/+/3838, SPDK
- https://review.spdk.io/gerrit/c/spdk/spdk/+/3839, SPDK
- https://github.com/oracle/qemu/tree/vfio-user-v0.1, QEMU
- Slack: spdk-team.slack.com#vnvme and libmuser.slack.com


intel