DPDK SUMMIT APAC•2021

An Ideal I/O Device Virtualization Framework in DPDK

XIUCHUN LU, CHENBO XIA

Agenda

DPDK SUMMIT APAC 2021

- Way to ideal I/O device virtualization solutions
- I/O device virtualization framework in DPDK
- Build your emulated device
- Current status and next plan

Ideal I/O Device Virtualization Solution

DPDK SUMMIT APAC•2021

Ideal I/O device virtualization solutions?

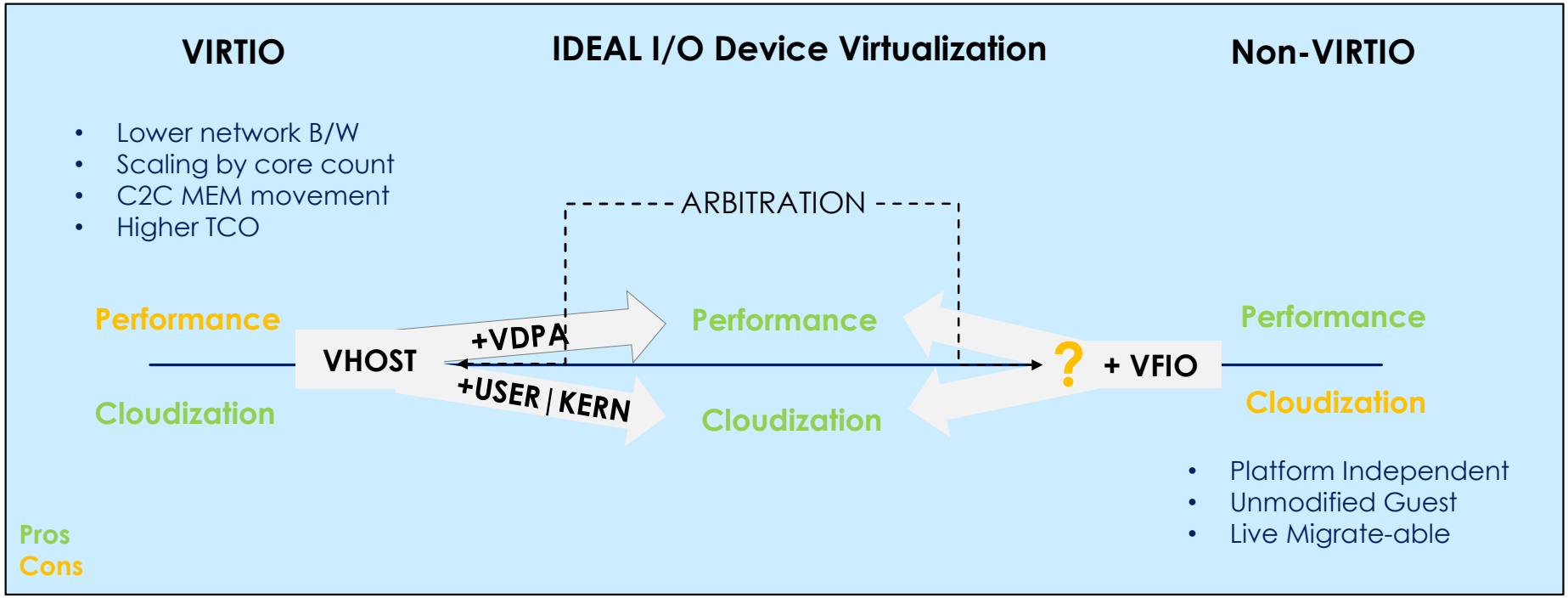
- VIRTIO?
  - Yes, then
    - Can VIRTIO cover all the user scenarios?
    - Is VIRTIO the only answer to this question?
    - …

VIRTIO as the Answer

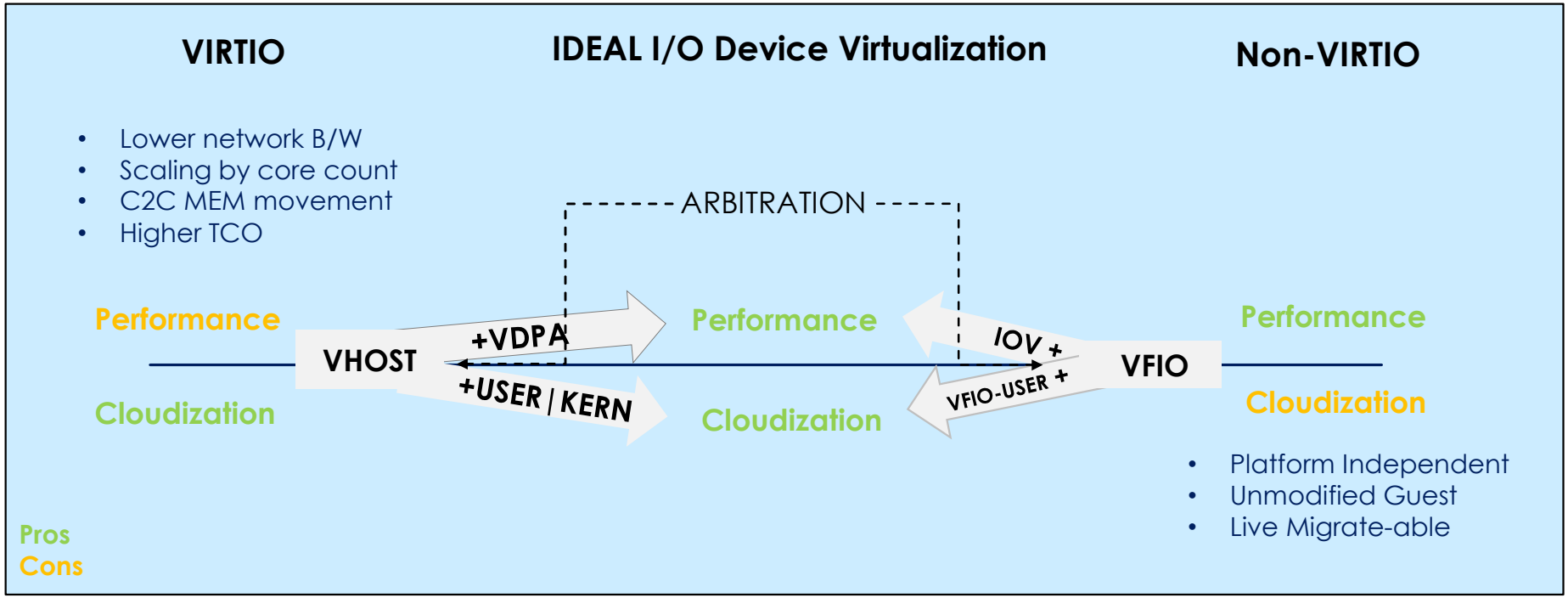
DPDK SUMMIT APAC•2021

- Most likely **YES**
  - Platform independent
  - Unmodified guest
  - A standardized open interface
  - A number of open source communities adopted
  - The performance issue can be addressed by vDPA
- What if VIRTIO incompatible devices
  - High performance
  - Proprietary accelerators
  - …

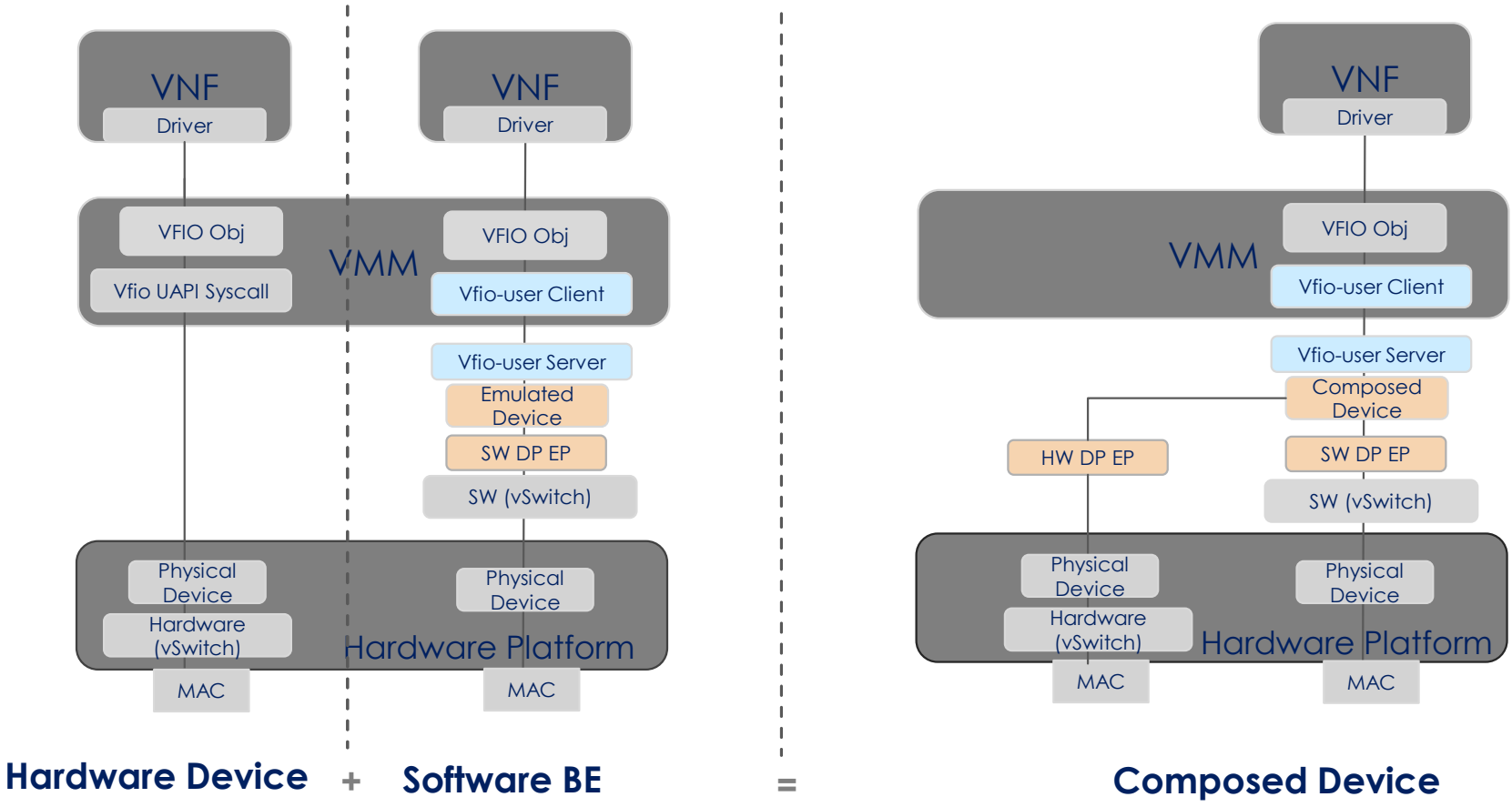
Path to the Ideal I/O Device Virtualization



Path to the Ideal I/O Device Virtualization



Platform Independent Composed Device



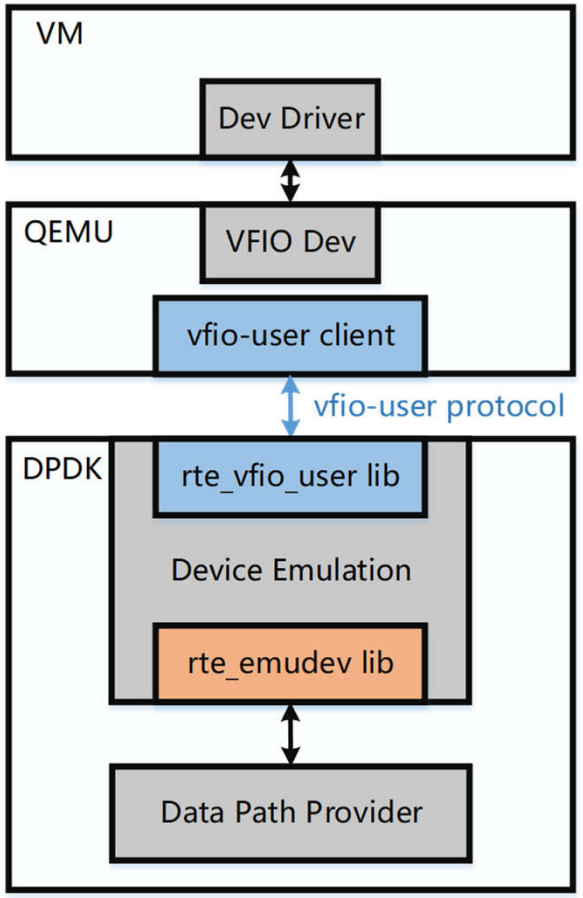
Agenda

DPDK SUMMIT APAC•2021

- Way to ideal I/O device virtualization solutions
- I/O device virtualization framework in DPDK
- Build your emulated device
- Current status and next plan

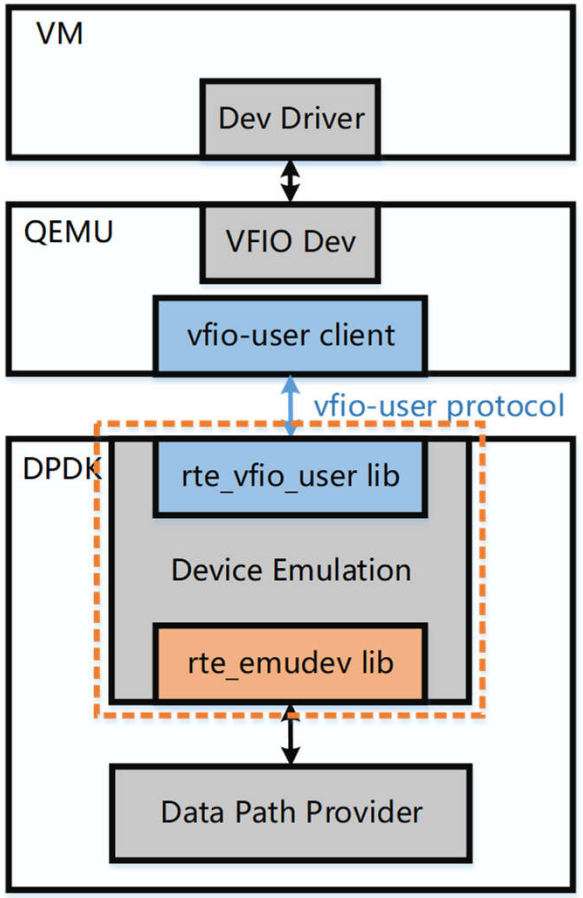
I/O Device Emulation Framework Overview

- vfio-user protocol: Unix domain socket messages based on VFIO device definition (i.e., Region/IRQ/DMA/...)
- rte_vfio_user lib and rte_emudev lib enable DPDK to be an alternative I/O device emulation library
- VFIO Dev : Device Emulation : DP Provider = 1 : 1 : 1

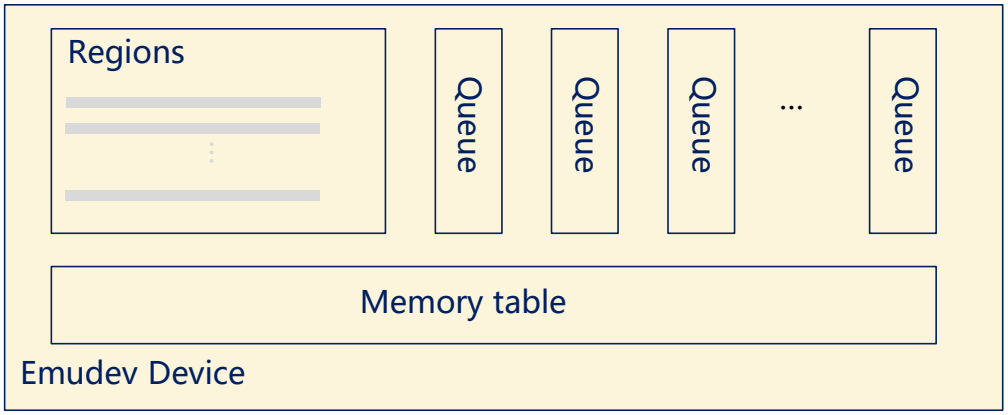


How to Build Your Emulated Device

- device emulation uses rte_vfio_user lib and rte_emudev lib
- rte_vfio_user lib
  - act as vfio_user server in this case
  - provide client implementation for container usage
- rte_emudev lib
  - abstracted as emudev device

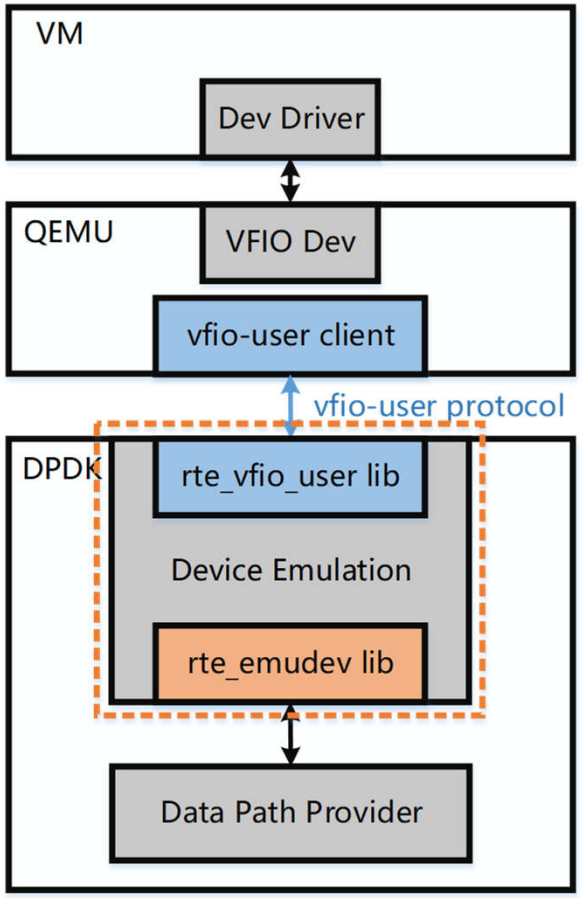


Emudev Device: Objects



Emudev device objects

- Regions: device layout
- Queues: address/size + doorbell + interrupt
- Memory table: DMA mapping table



Emudev Device: Ops

For application

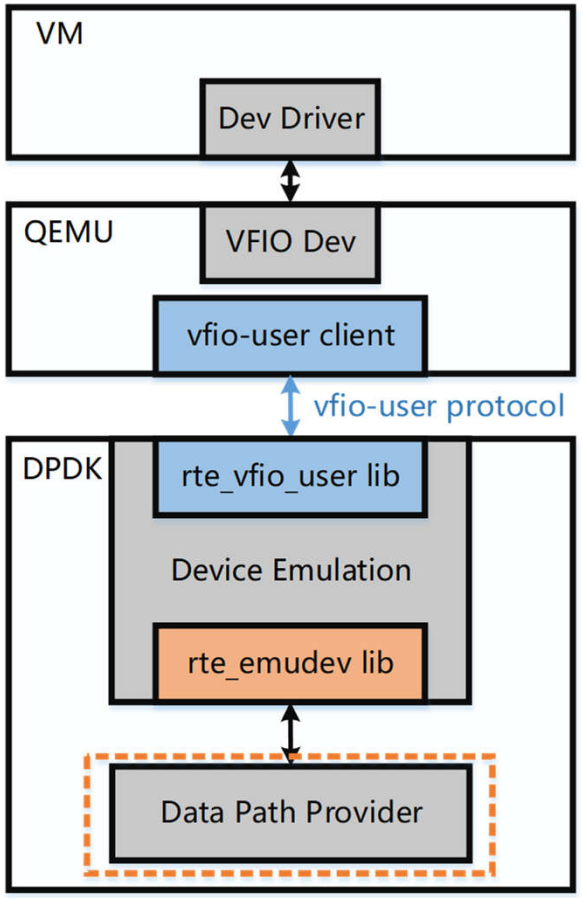
- Lifecycle management
- Region info config and set
- Register notify callback

For data path provider

- Register notify callback
- Region channel setup and read/write
- Queue and queue notify scheme setup
- DMA table setup

How to Build Your Emulated Device

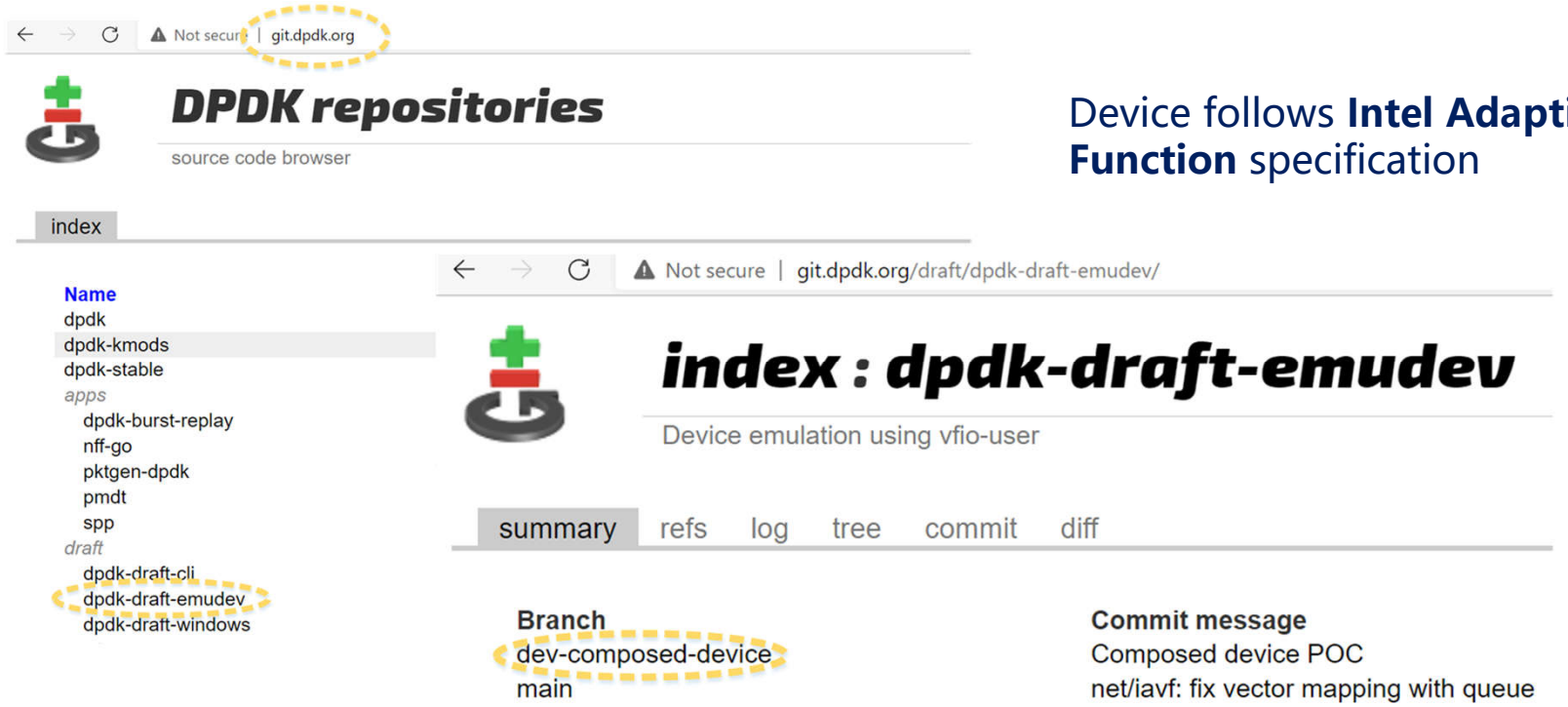
- Data path provider could be a virtual device (vdev) or PCI device driver
- Could be Ethdev
- Emudev APIs + Ops provided by device emulation



Build up for SW/HW Data Path Provider

- Query device info
  - Similar for SW/HW
- Set up regions/attributes/queues
  - SW: set up all regions/attributes
  - HW: only set up **Control Plane (CP)** regions/attributes/queues
- Set up DMA
  - SW: maintain the whole DMA table
  - HW: maintain part of DMA table in software (for CP), rest used to set up HW IOMMU
- Register event callback

Hands on Composed Device



Device follows **Intel Adaptive Virtual Function** specification

Hands on Composed Device

DPDK SUMMIT APAC 2021

Server:

```
./avfbe_for_summit/build/app/dpdk-testpmd -l 32-33 \
        -n 4 --socket-mem 1024 \
        --vdev 'emu_iavf0,sock=/tmp/sock1,queues=1' \
        -b 0000:af:01.0 \
        --file-prefix=hoqst -- -i \
```

Client:

```
./avfbe_for_summit/build/app/dpdk-testpmd -l 30-31 \
        -n 4 --socket-mem 1024 \
        --vdev 'iavf_client,path=/tmp/sock1' \
        --no-pci --file-prefix=host \
        -- -i \
```

Server attaches **HW** Data Plane:
(Provided by **Intel E810 NIC**)

```
testpmd> port attach 0000:af:01.0,dpa=1,emu=emu_iavf0
Attaching a new port...
EAL:   using IOMMU type 1 (Type 1)
EAL: Probe PCI driver: net_iavf (8086:1889) device: 0
EAL: Releasing pci mapped resource for 0000:af:01.0
EAL: Calling pci_unmap_resource for 0000:af:01.0 at 0
EAL: Calling pci_unmap_resource for 0000:af:01.0 at 0
EAL: Probe PCI driver: net_iavf_dpa (8086:1889) device
EAL:   using IOMMU type 1 (Type 1)
```

Server attaches **SW** Data Plane:

```
testpmd> port attach net_iavfbe,emu=emu_iavf0
Attaching a new port...
rte_pmd_iavfbe_probe(): Initializing pmd_iavfbe

eth_dev_iavfbe_create(): Creating iavfbe ethdev
```

Agenda

DPDK SUMMIT APAC 2021

- Way to ideal I/O device virtualization solutions
- I/O device virtualization framework in DPDK
- Build your emulated device
- Current status and next plan

Status & Plan

DPDK SUMMIT APAC 2021

- vfio-user Protocol patch (v7)
  - https://patchew.org/QEMU/20201130161229.23164-1-thanos.makatos@nutanix.com/
- DPDK patch
  - rte_vfio_user lib: http://patchwork.dpdk.org/project/dpdk/list/?series=14711
  - rte_emudev lib + iavf emudev driver: http://patchwork.dpdk.org/project/dpdk/list/?series=14712
  - iavf back-end driver: http://patchwork.dpdk.org/project/dpdk/list/?series=14570&state=*
  - iavf client driver: http://patchwork.dpdk.org/project/dpdk/list/?series=14576&state=*
- DPDK sub-tree
  - http://git.dpdk.org/draft/dpdk-draft-emudev/
- Contact
  - XIUCHUN LU (xiuchun.lu AT intel.com) / CHENBO XIA (chenbo.xia AT intel.com)


DPDK
SUMMIT
APAC•2021