FTF 2016
TECHNOLOGY FORUM

KVM VIRTUALIZATION: LEVERAGING I/O VIRTUALIZATION ON QorIQ PLATFORMS FOR VNFS

BHARAT BHUSHAN
PRINCIPAL STAFF ENGINEER
DIANA CRĂCIUN
SOFTWARE ENGINEER
FTF-NET-N1844
MAY 2016

Software Products and Services

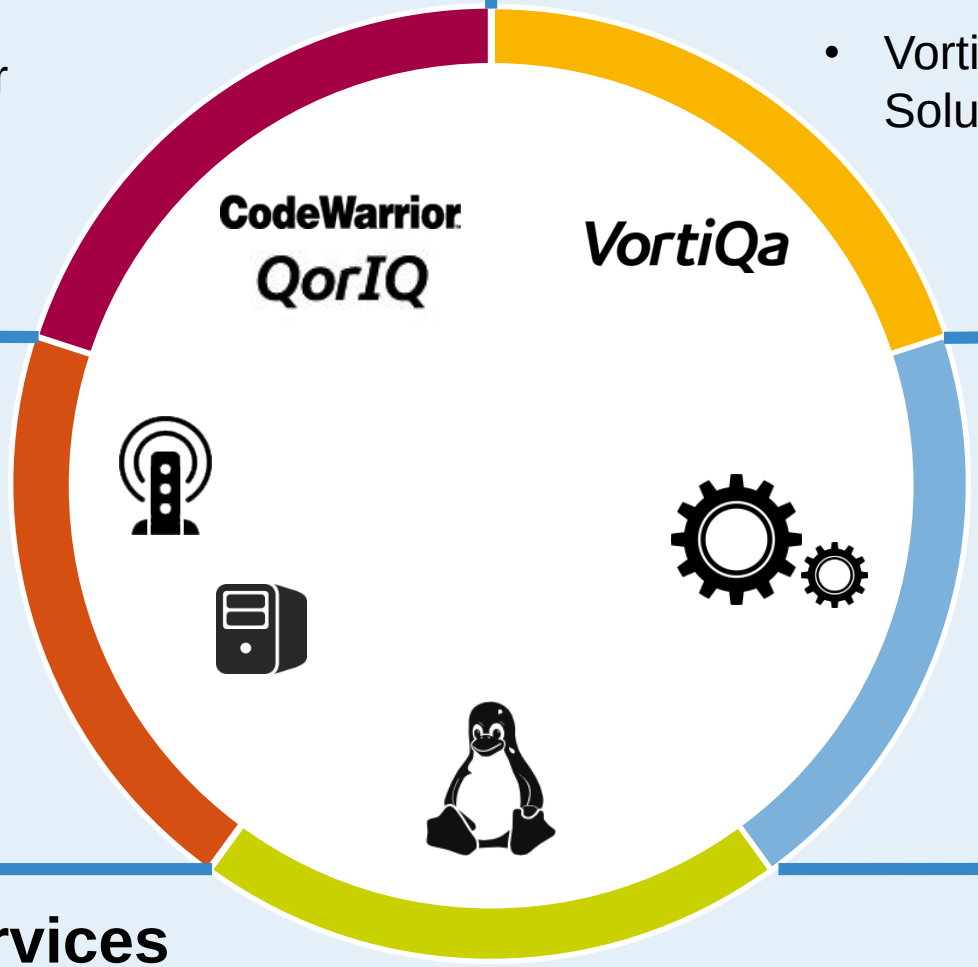
Visit us in the Tech Lab – #247

Development Tools

- CodeWarrior

Runtime Products

- VortiQa Software Solutions



Solutions Reference

- IOT Gateway
- OpenWRT+

Integration Services

- Security Consulting
- Hardened Linux

Linux® Services

- Commercial Support
- Performance Tuning

Accelerate **Customer Time-to-Market**

Deliver **Commercial Software, Support, Services and Solutions**

Simplify **Software Engagement with NXP**

Create **Success!**

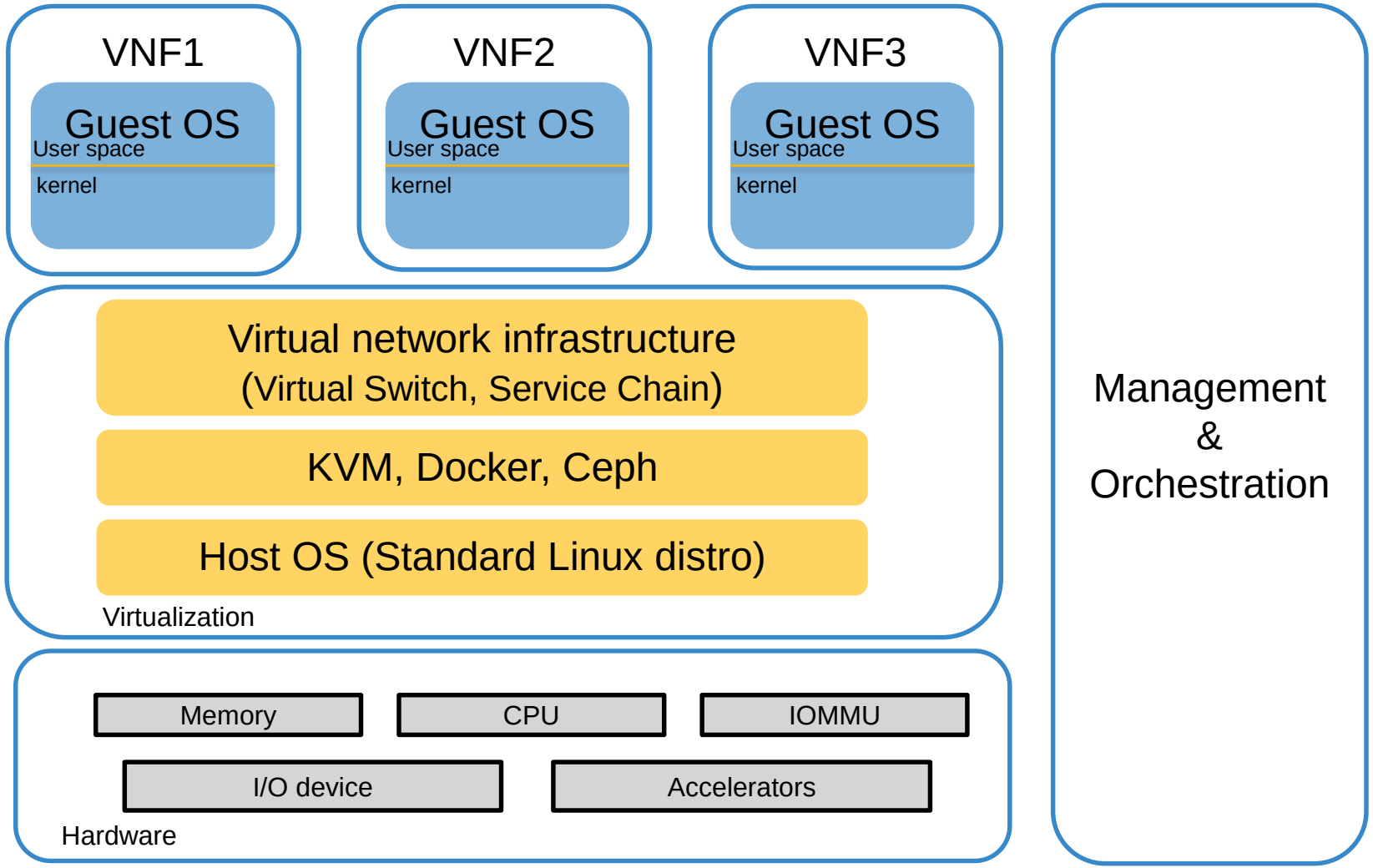
Find us online at www.nxp.com/networking-services

AGENDA

- Virtualization Overview
- I/O Virtualization
- Direct Assignment
- VirtIO
- Conclusions

VIRTUALIZATION OVERVIEW

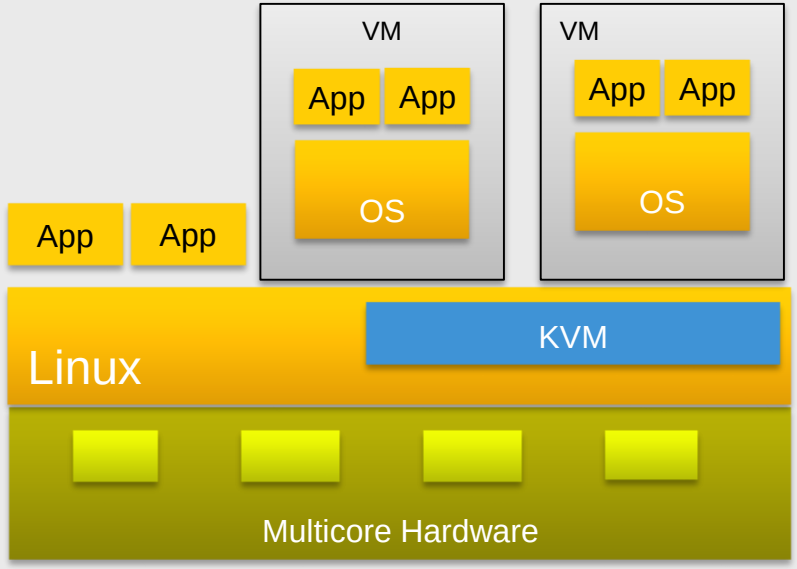
NFV and VNFs


VNF1
Guest OS
User space
kernel
VNF2
Guest OS
User space
kernel
VNF3
Guest OS
User space
kernel
Virtual network infrastructure
(Virtual Switch, Service Chain)
KVM, Docker, Ceph
Host OS (Standard Linux distro)
Virtualization
Memory
CPU
IOMMU
I/O device
Accelerators
Hardware
Management
&
Orchestration

NXP virtualization solutions

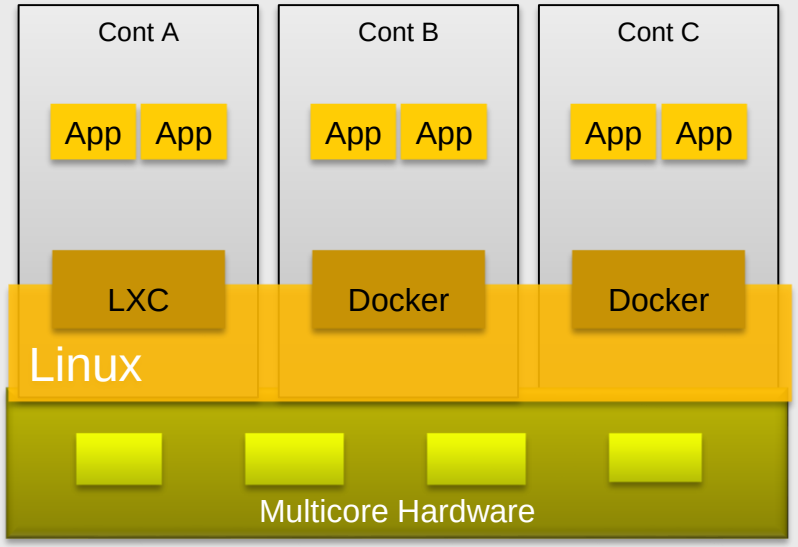
KVM

- Linux ® Hypervisor
- Resource Virtualization/oversubscription
- Open source
- Qemu user space emulation used

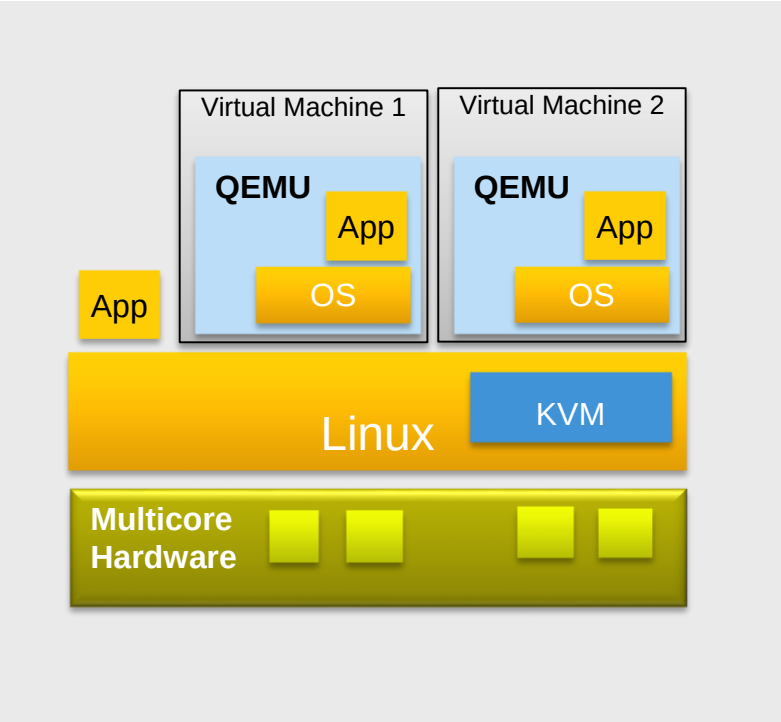


OS Virtualization

- Lightweight Overhead
- Isolation and Resource Control in Linux ®
- Decreased Isolation (Kernel sharing)

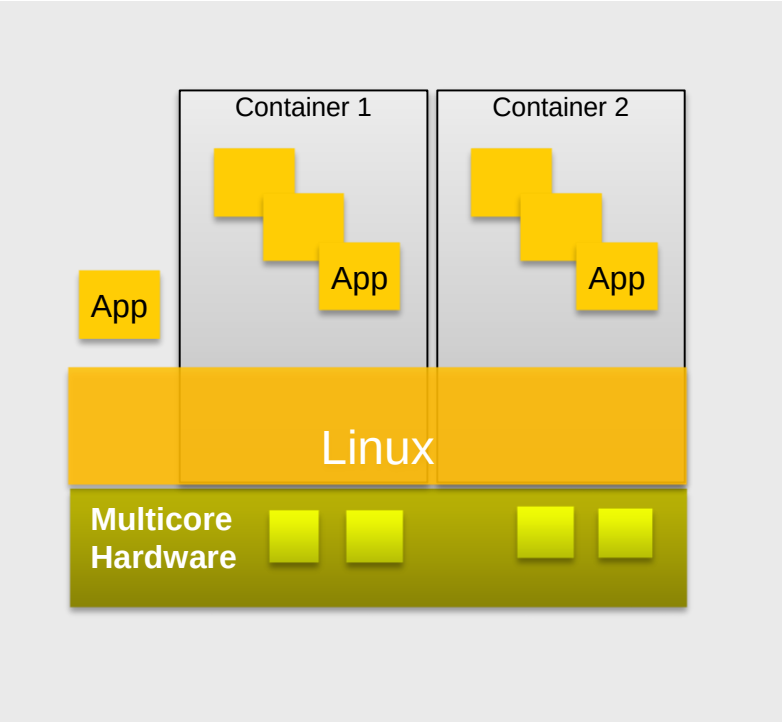


KVM/QEMU



- KVM/QEMU– open source virtualization technology based on the Linux® kernel
- KVM is a Linux kernel module
- QEMU is a user space emulator that uses KVM for acceleration
- Run virtual machines alongside Linux applications
- No or minimal OS changes required
- Virtual I/O capabilities
- Direct/pass thru I/O – assign I/O devices to VMs

Linux Containers



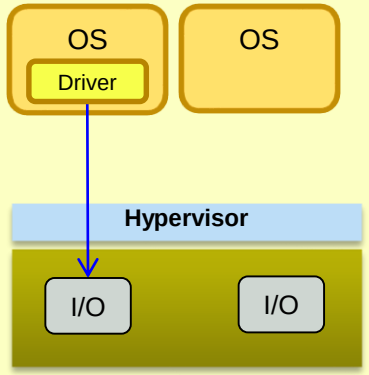
- OS level virtualization / process level virtualization
- Single kernel for host and guests, virtualized userspace instances – OS appears isolated
- Low overhead, lightweight, secure partitioning of Linux applications into different domains
- Per domain resource utilization control – CPU, memory, I/O bandwidth
- Multiple resource instances – namespaces
  - Process – process trees
  - Network – network stack (netdevs, socket families, FDBs)
- Based on a collection of technologies including kernel components (cgroups, namespaces), and user space tools (LXC, libvirt, Docker)

I/O VIRTUALIZATION

Device Usage in Virtual Environments

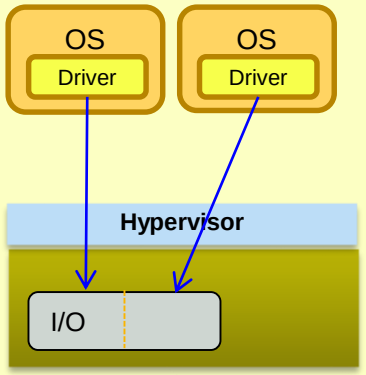
Direct Access

- Fastest native performance
- Direct access to hardware



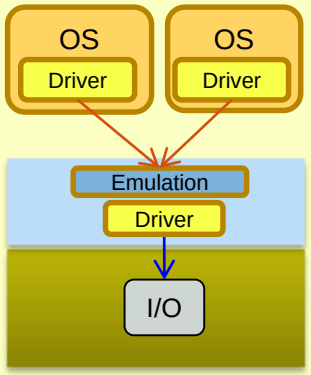
Hardware support

- Hardware partitioned
- One hardware block



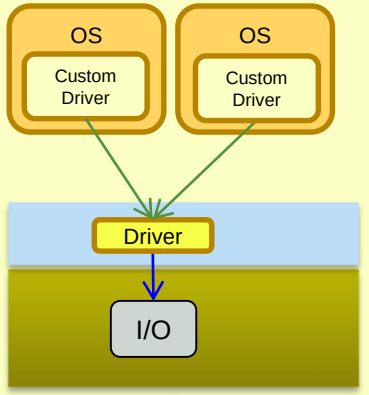
Full emulation

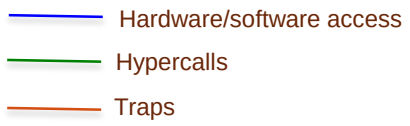
- Driver in Hypervisor
- Emulation in Hypervisor
- Unmodified Drivers in Guest



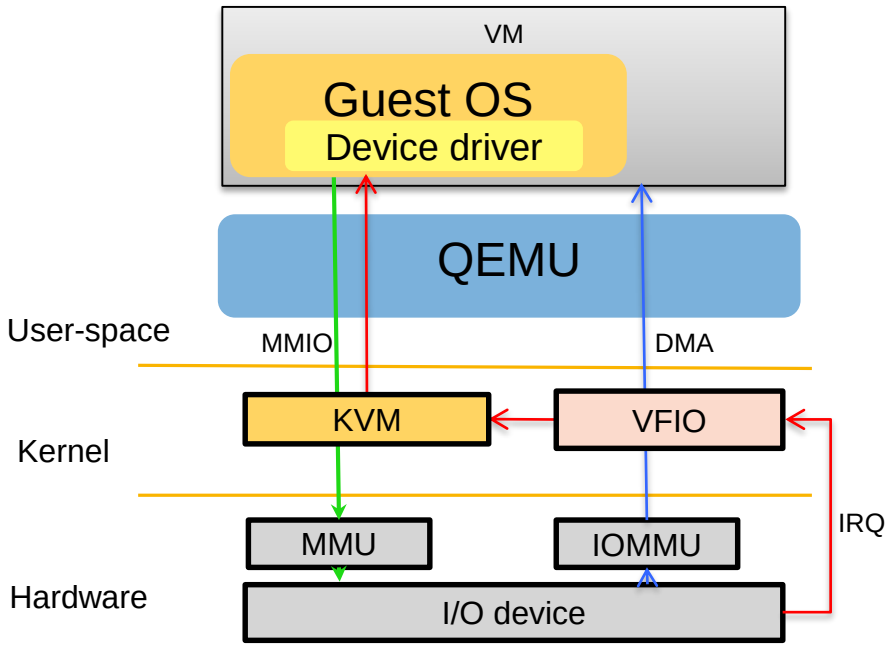
Para-Virtualized

- Driver in Hypervisor
- Modified Drivers in Guest

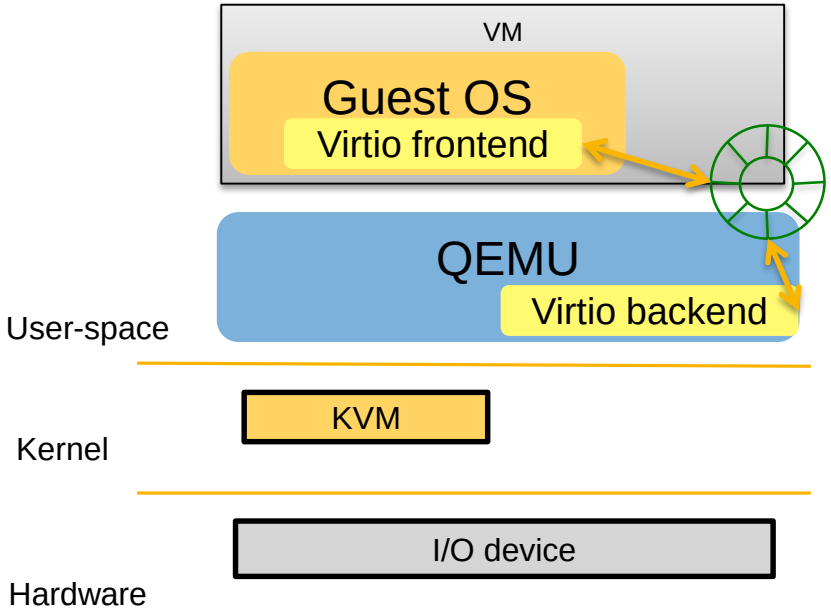




Device Usage in KVM/Linux

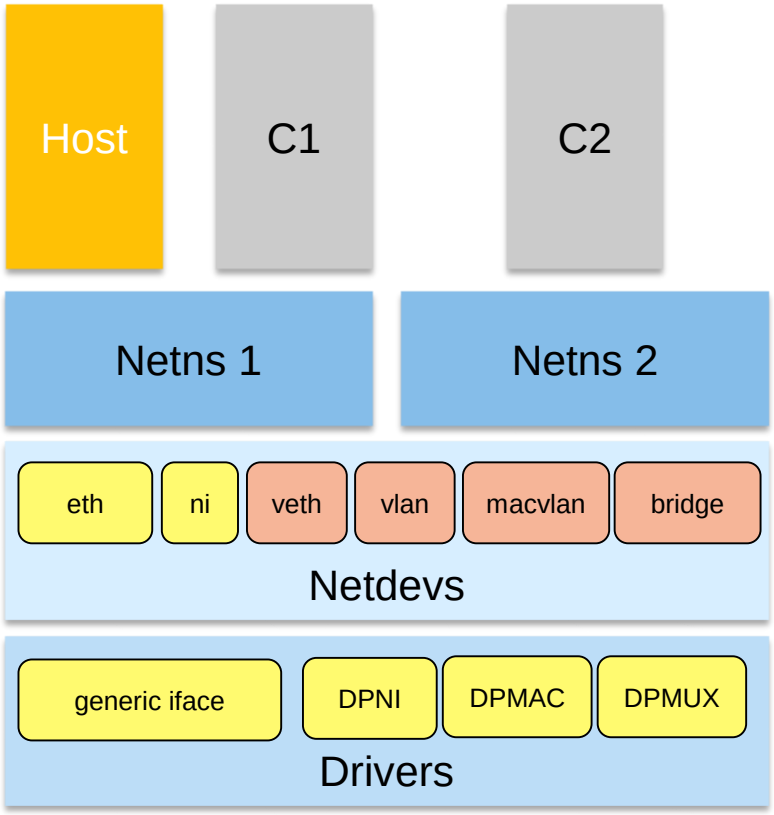


VFIO (simplified view)



Virtio (simplified view)

Device Usage in Containers

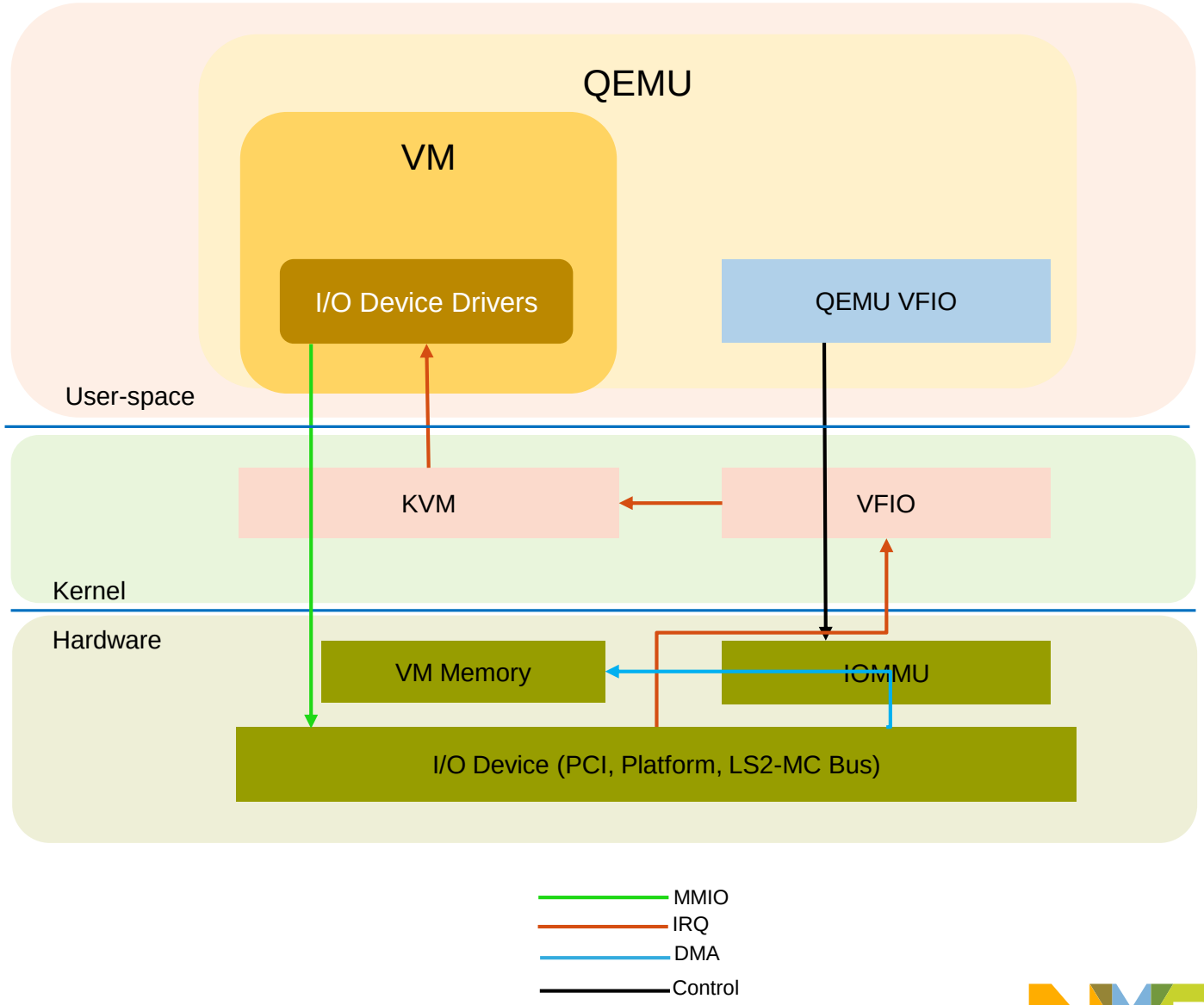


- Each container (userspace instance) has a net namespace
  - Multiple containers can share the same netns
- Each netdev belongs to a net namespace
- The netdev can be:
  - Physical: has an associated HW device or abstraction
  - Virtual: entirely SW (veth, vlan, bridge, etc.)
- Virtual netdev overhead is low – differences arise from technology specifics
  - Bridge: kernel switching
  - MACVLAN: MAC level VLAN
  - VETH: IP level SW pairs
- Mix and match

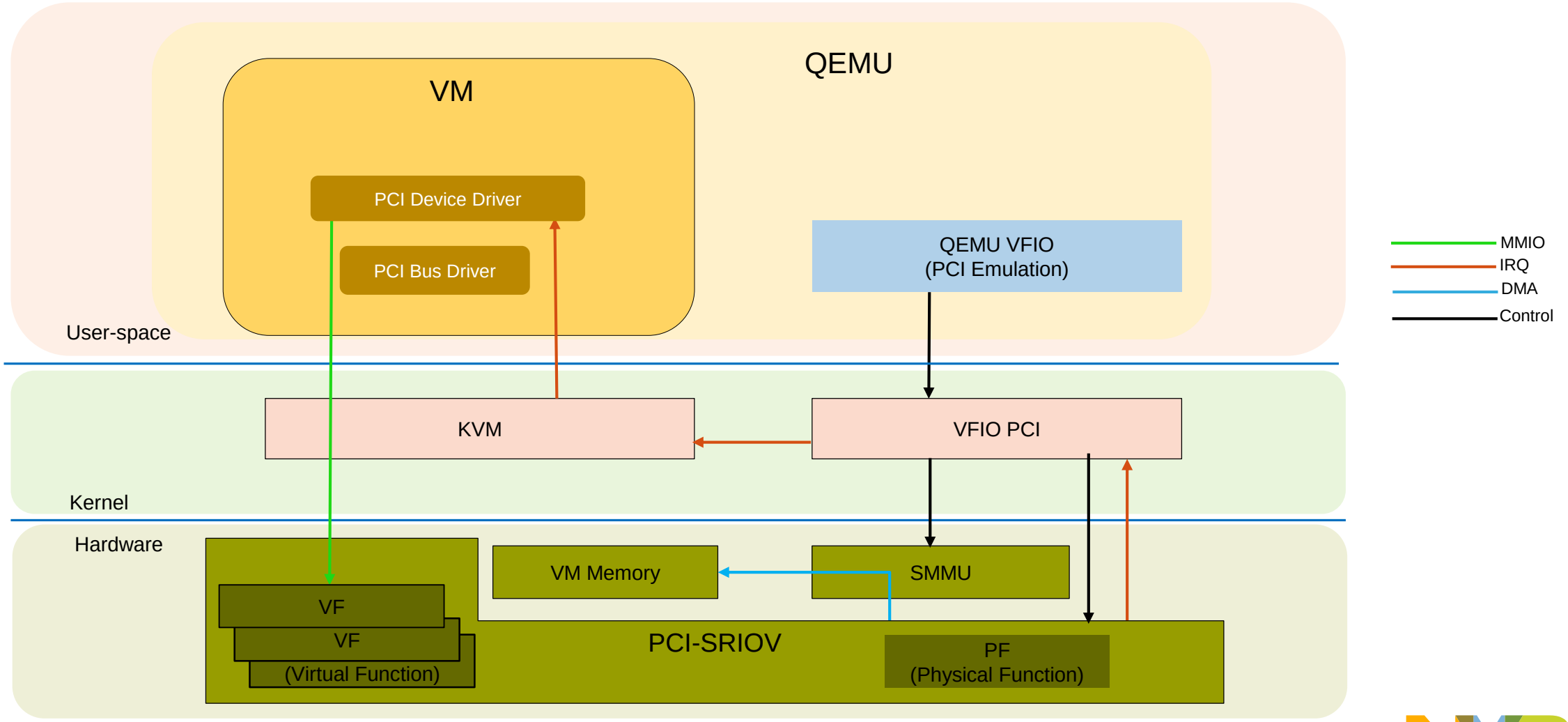
DIRECT ASSIGNMENT

Introduction to VFIO

- VFIO (Virtual Function IO)
  - Linux user space driver infrastructure
  - Enforces IOMMU protection
- VFIO Provides
  - Device access (mmap() device MMIO regions)
  - IOMMU programming interface
  - High performance interrupt support
- Bus support
  - PCI, platform devices, LS2 MC bus

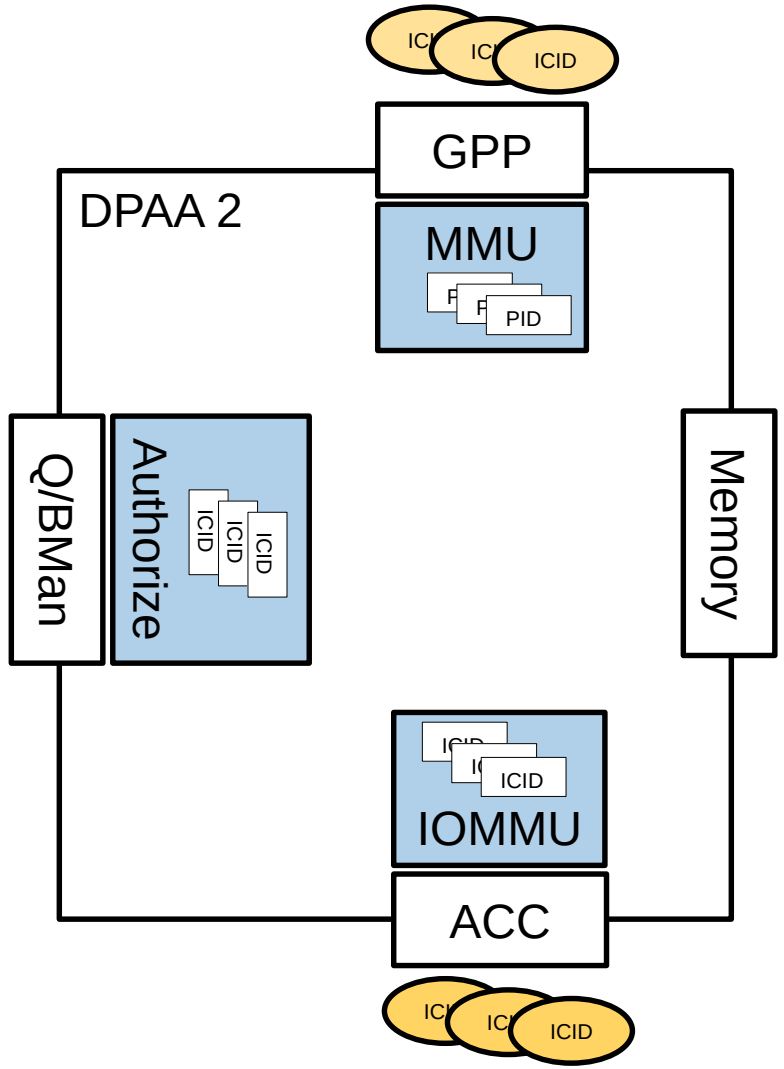


PCI Device Direct Assignment to VM

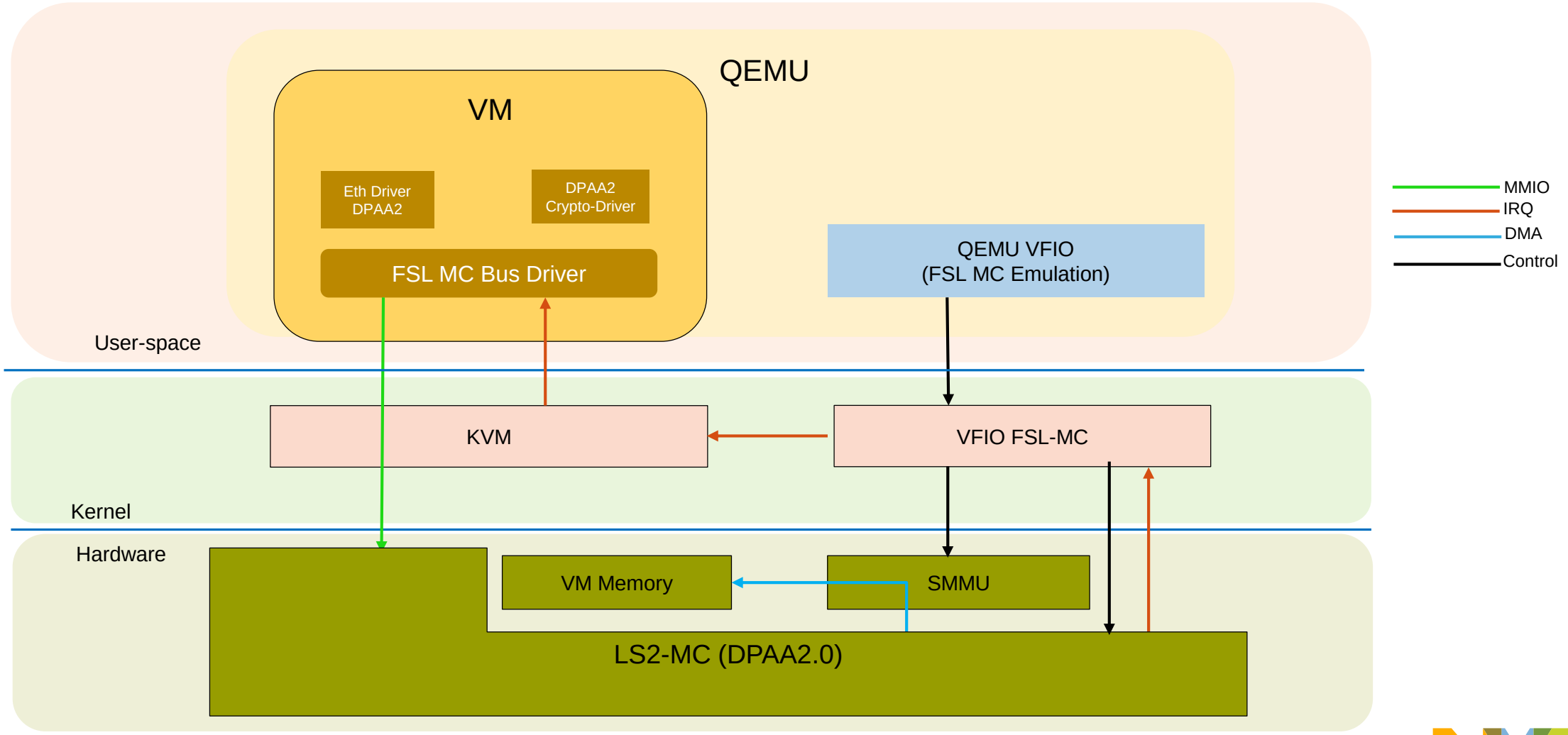


DPAA2 Enables Secure Direct Assignment

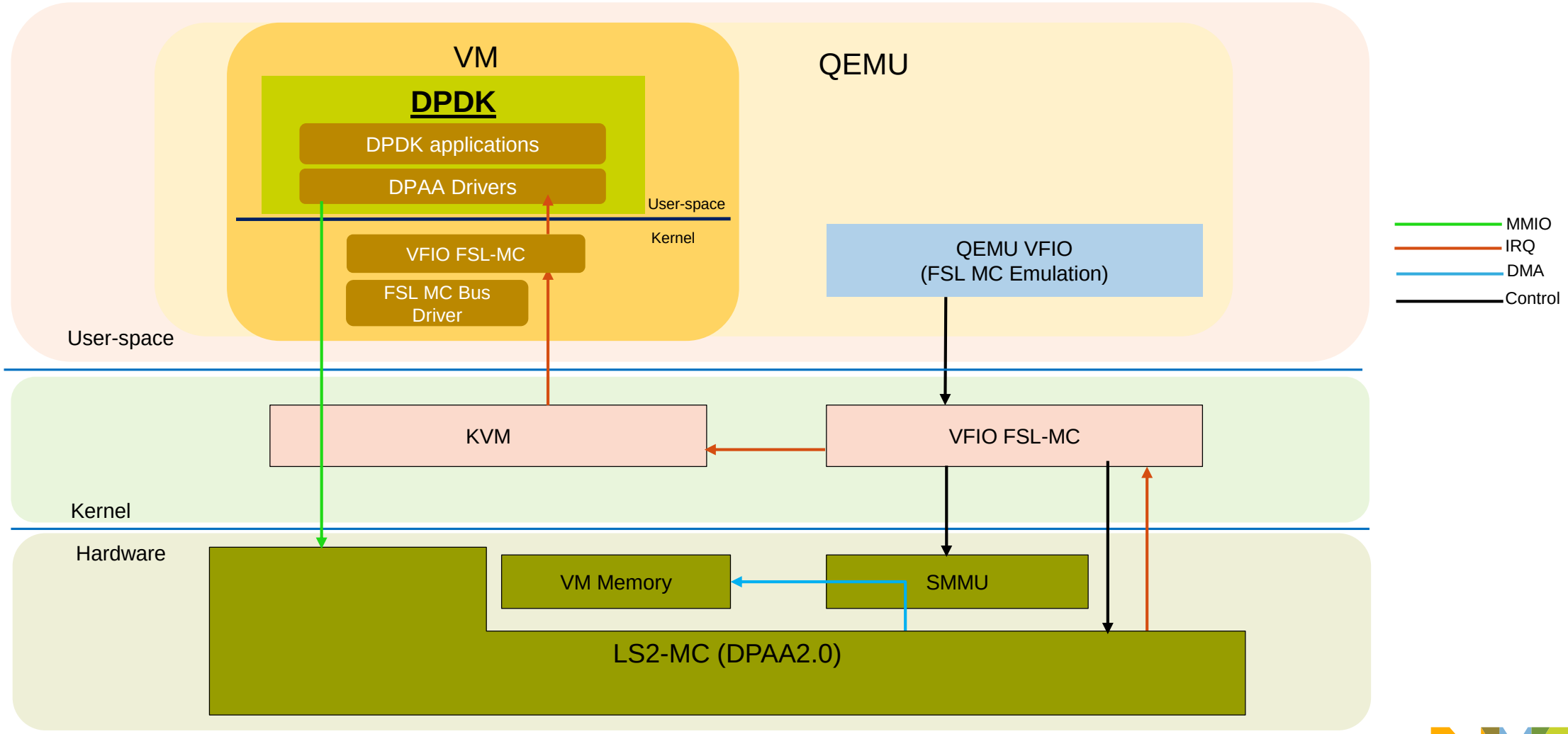
- Management Complex (MC) is optimized for resource assignment to various software contexts through Management Complex
  - Linux MC bus
  - Resource management tool
- IOMMU translation and protection for user-space (DPDK and QEMU)
  - ICID (StreamID)
  - MC bus integration with VFIO
  - Device reset
- DPAA secured with Authorization Tables



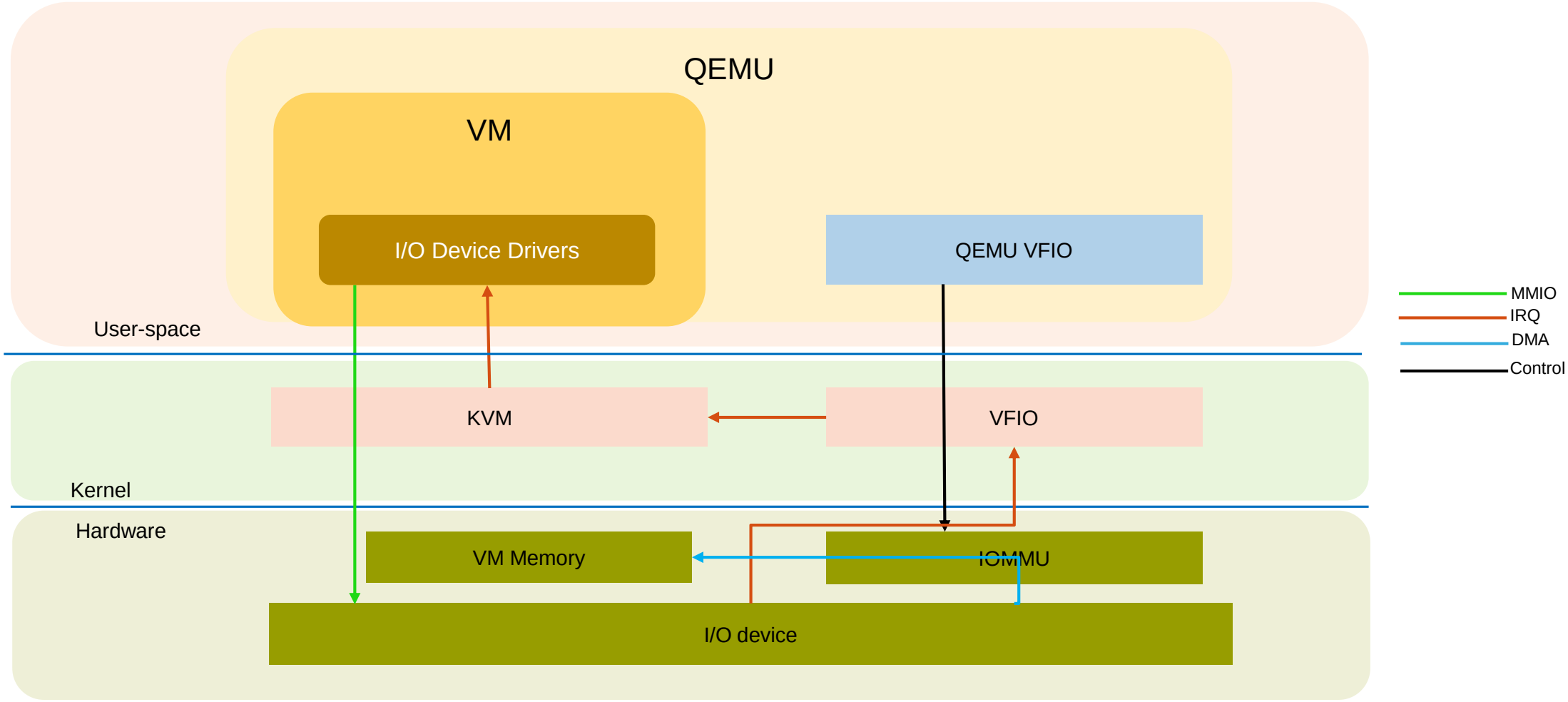
DPAA2 Device Direct Assignment to VM



DPAA2 Device Pass-through to DPDK in VM



Platform devices direct assignment

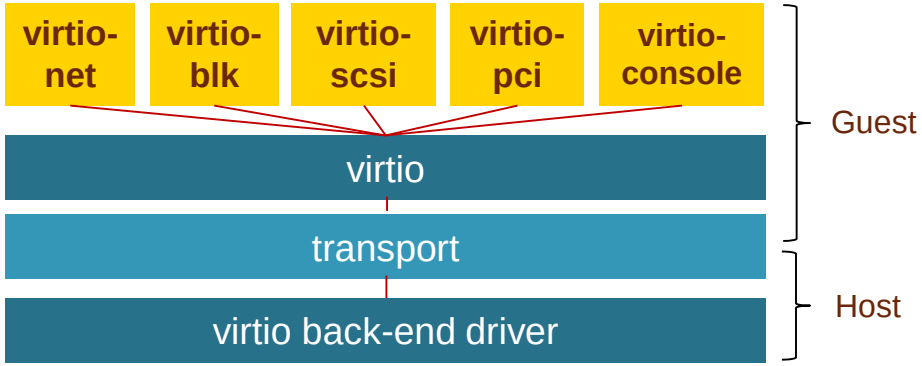


VIRTIO DETAILS

Virtual I/O Device

Virtio family of devices

- Found in virtual environments
- By design they look like physical devices
- Use guest standard drivers and discovery mechanisms
- Specification defined by OASIS technical committee



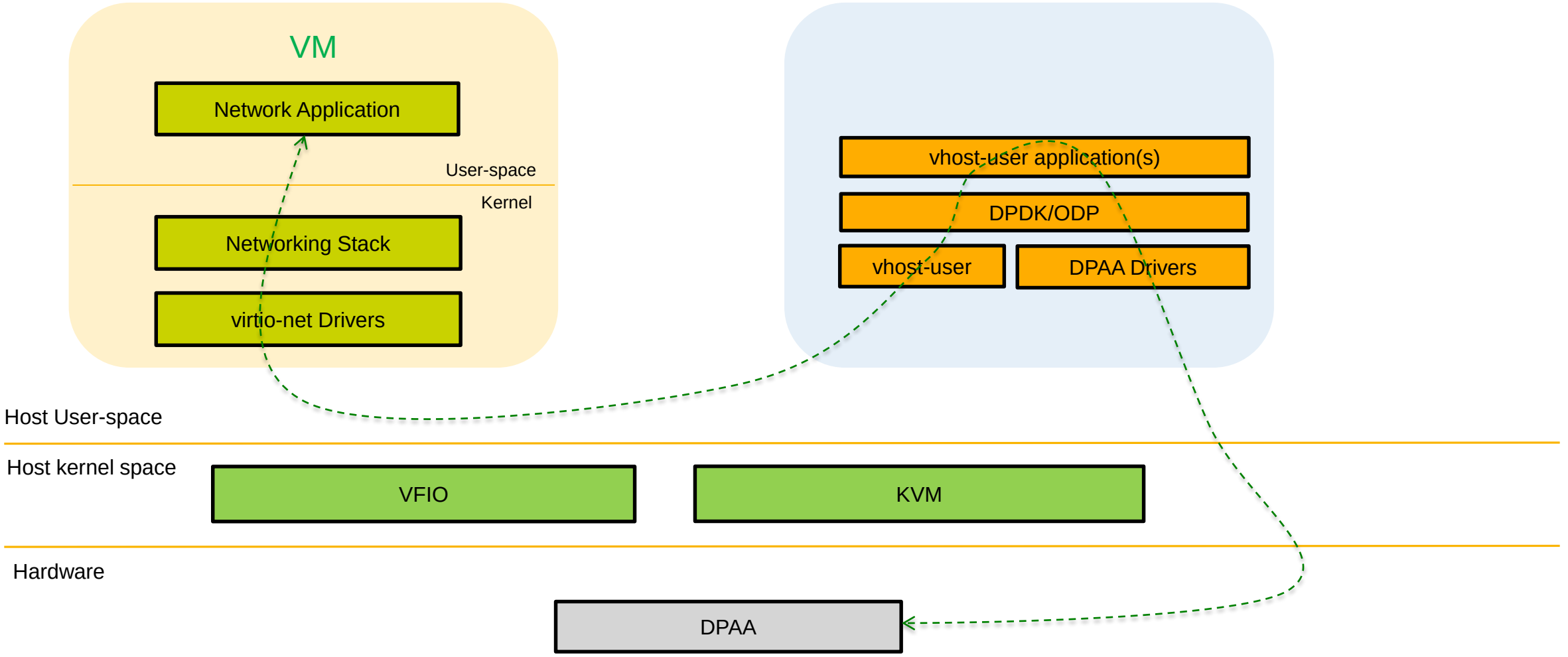
Virtio specification purpose

- Straightforward - use normal bus mechanisms of interrupts and DMA
- Efficient - rings of descriptors for both input and output, laid out to avoid cache effects
- Standard - makes no assumptions about guest environment beyond supporting MMIO, Channel I/O or PCI bus transports.
- Extensible - devices contain feature bits acknowledged by the guest OS

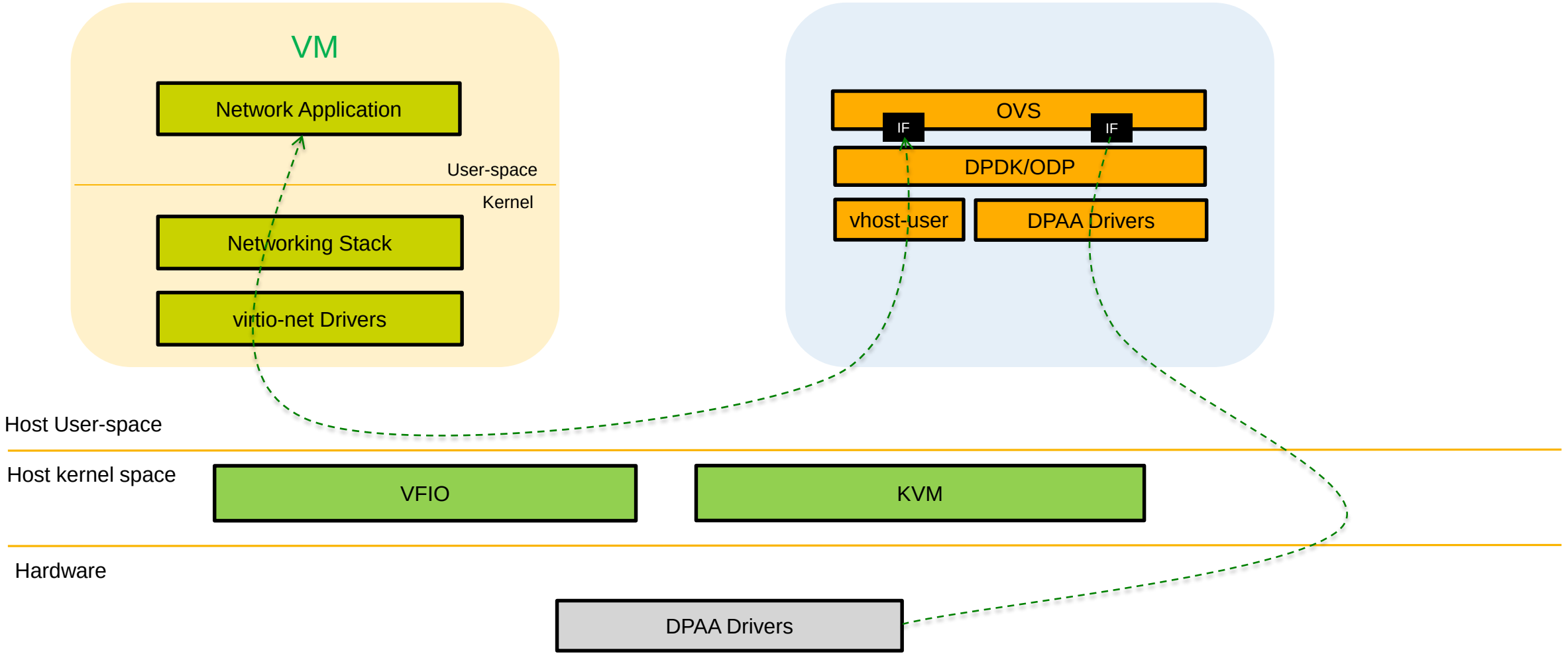
Virtio device facilities

- Device status field
- Feature bits
- Device Configuration space
- One or more virtqueues

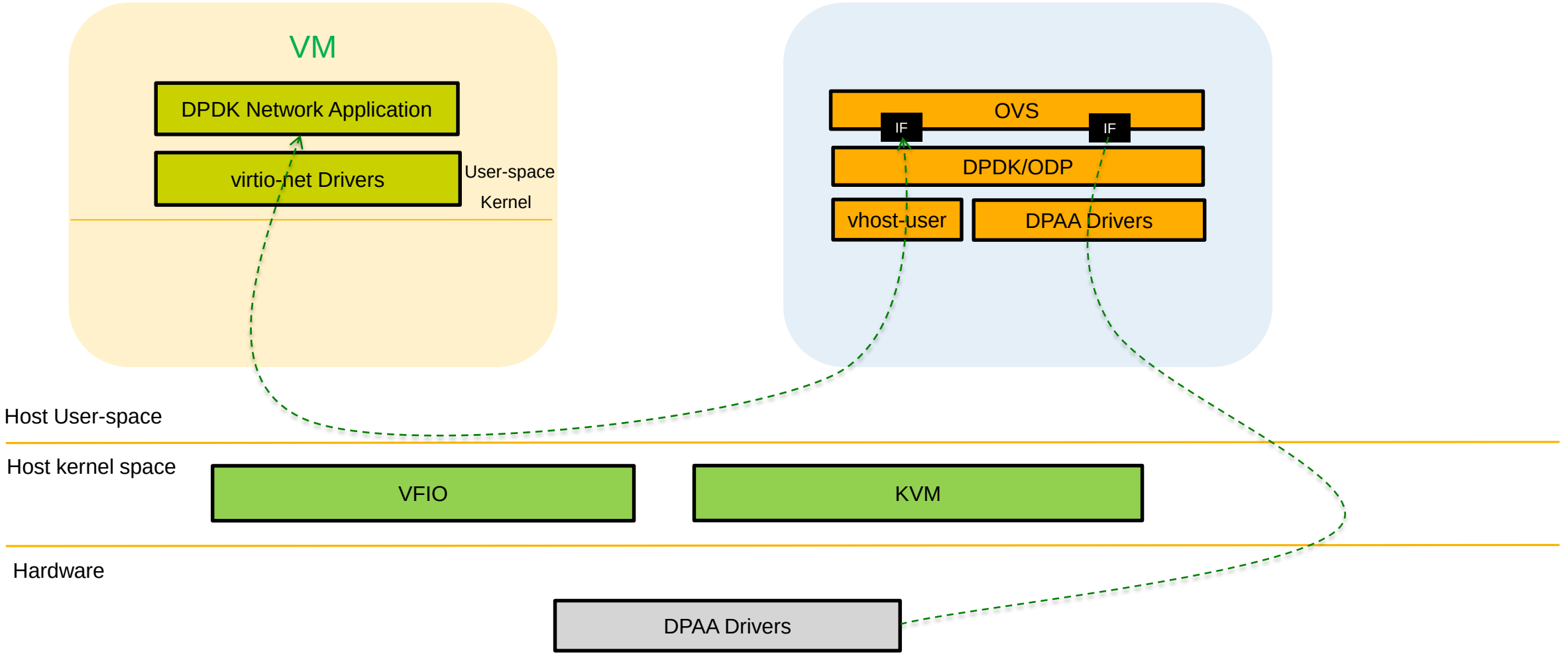
Virtio-net: Vhost back-end in user space



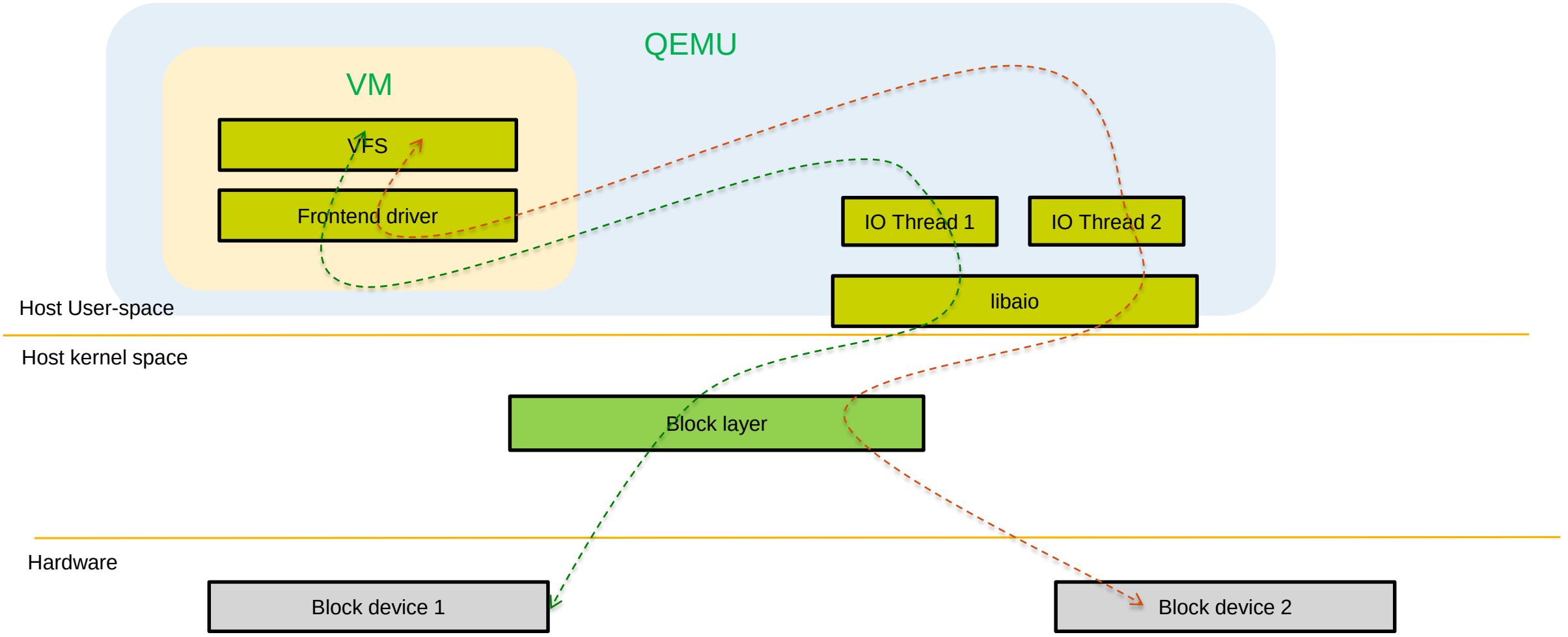
Virtio-net: DPDK-OVS backend



Virtio-net: DPDK in guest using virtio-net



Virtio-block dataplane



CONCLUSIONS

Conclusions

- Efficiency, performance and flexibility in I/O virtualization solutions are important ingredients for network function virtualization
- KVM provides VirtIO and direct assignment offering NFV system designers the possibility to choose the best suited solution for their applications.

SECURE CONNECTIONS
FOR A SMARTER WORLD

ATTRIBUTION STATEMENT