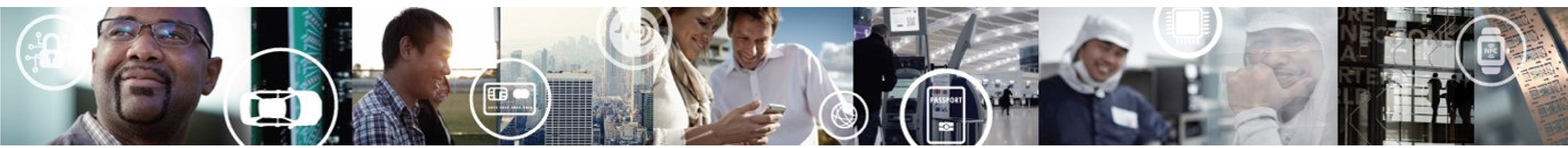
EUF-NET-T1745

ARM® V8 VIRTUALIZATION FOR LAYERSCAPE MULTICORE COMMUNICATIONS PROCESSORS

NXP TECH DAY טל אביב - 22 MARCH 2016

PETER.VANACKEREN@NXP.COM - SR. FAE EMEA



EXTERNAL USE

SECURE CONNECTIONS
FOR A SMARTER WORLD

Agenda

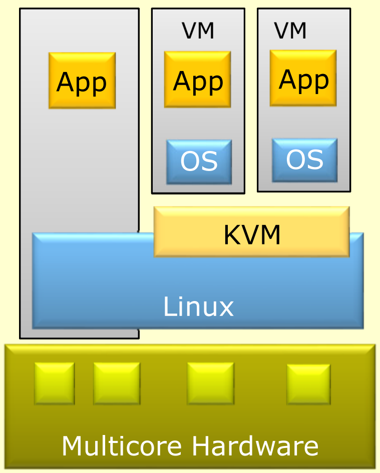
- Virtualization Introduction
- Layerscape ARM®v8 Virtualization Status & Roadmap
- I/O in KVM Environments
  - Device Virtualization - virtio
  - Device Direct-Assignment - VFIO
- Q&A

VIRTUALIZATION INTRODUCTION

Virtualization Technologies for QorIQ Layerscape architecture

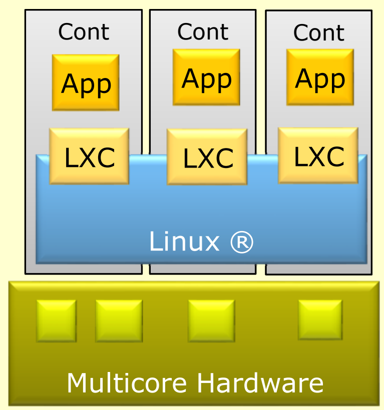
KVM

- Linux ® Hypervisor
- Resource Virtualization
- Resource Oversubscription
- 3rd Party OSs



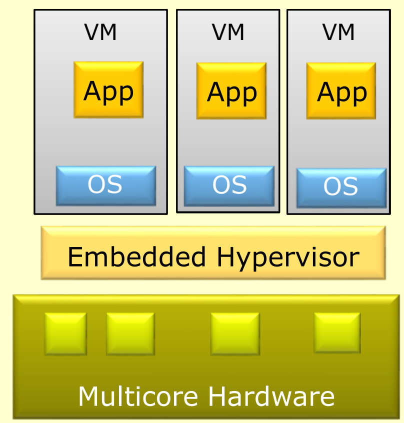
Linux Containers

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

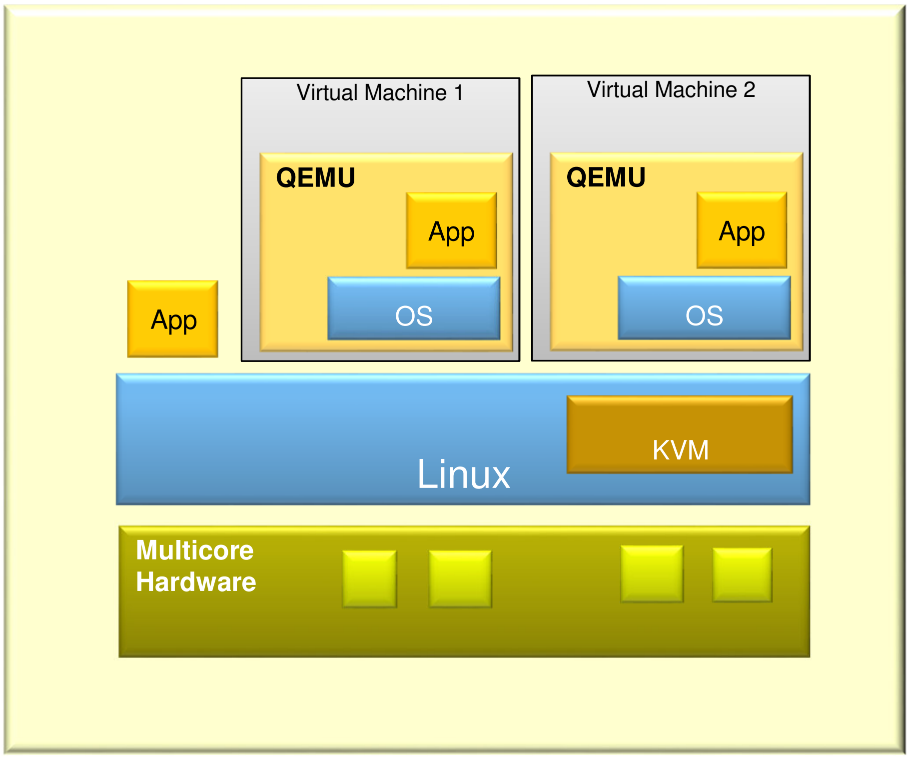


Embedded Hypervisor

- Lightweight Hypervisor
- Resource Partitioning
- Para-Virtualization
- Failover support
- 3rd Party OSs



KVM/QEMU – Overview

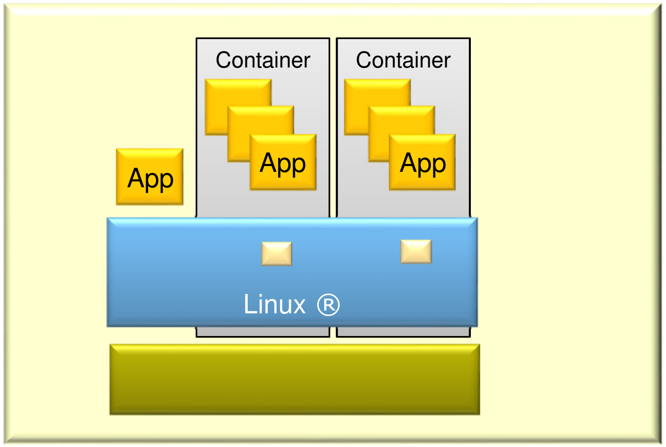


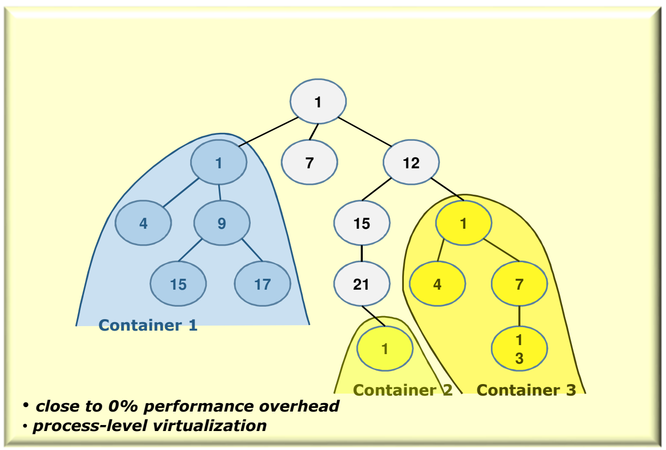
- KVM/QEMU– open source virtualization technology based on the Linux kernel
- KVM is a Linux kernel module
- QEMU is a user space emulator that uses KVM for acceleration
- Run virtual machines alongside Linux applications
- No or minimal OS changes required
- Virtual I/O capabilities
- Direct/pass thru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

KVM/QEMU

- QEMU is a user space emulator that uses KVM for acceleration
  - Uses dedicated threads for vcpus and I/O
  - KVM leverages hardware virtualization to run guest with higher privileges
  - Virtual chip emulation in kernel
  - I/O
    - Provides dedicated virtio I/O devices and standard drivers in Linux kernel
    - Uses VFIO Linux framework to direct assign physical PCI devices
    - Direct notifications between I/O threads and KVM using eventfds
    - vhost provides virtio emulation and I/O thread and in kernel
    - Multi-queue virtio devices connected to multi-queue tap devices
  - Provides services for console, debug, reset, watchdog, etc

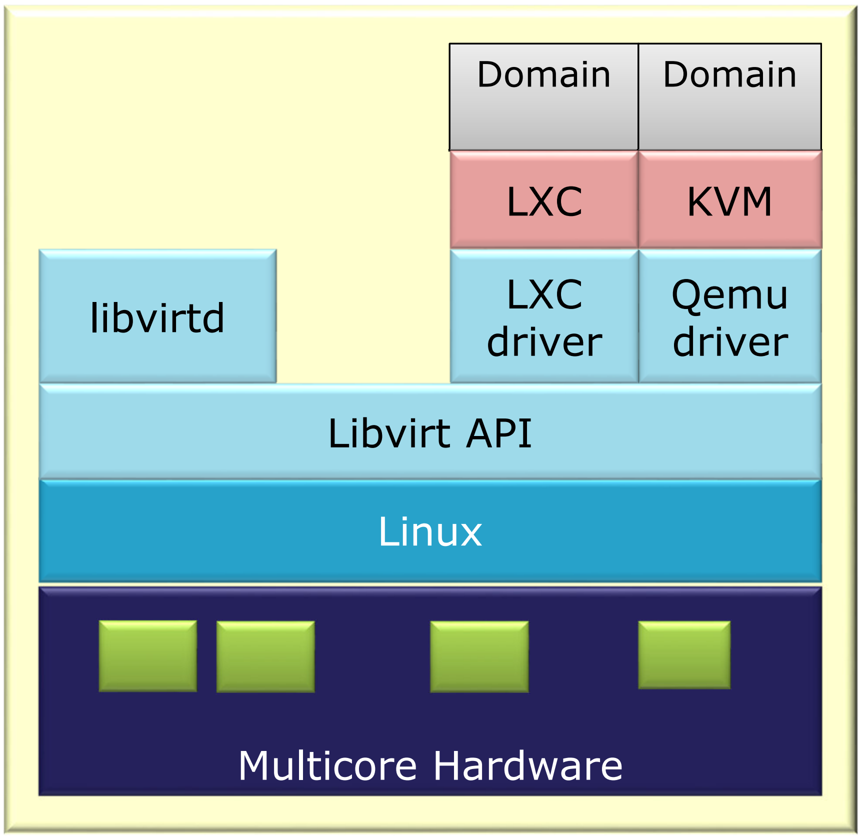
Linux Containers





- **L**inu**X** **C**ontainers is based on a collection of technologies including kernel components (cgroups, namespaces) and user-space tools (LXC).
- OS level virtualization
- Guest kernel is the same as the host kernel, but OS appears isolated
- Low overhead, lightweight, secure partitioning of Linux applications into different domains
- Can control resource utilization of domains– CPU, Memory, I/O bandwidth

Libvirt

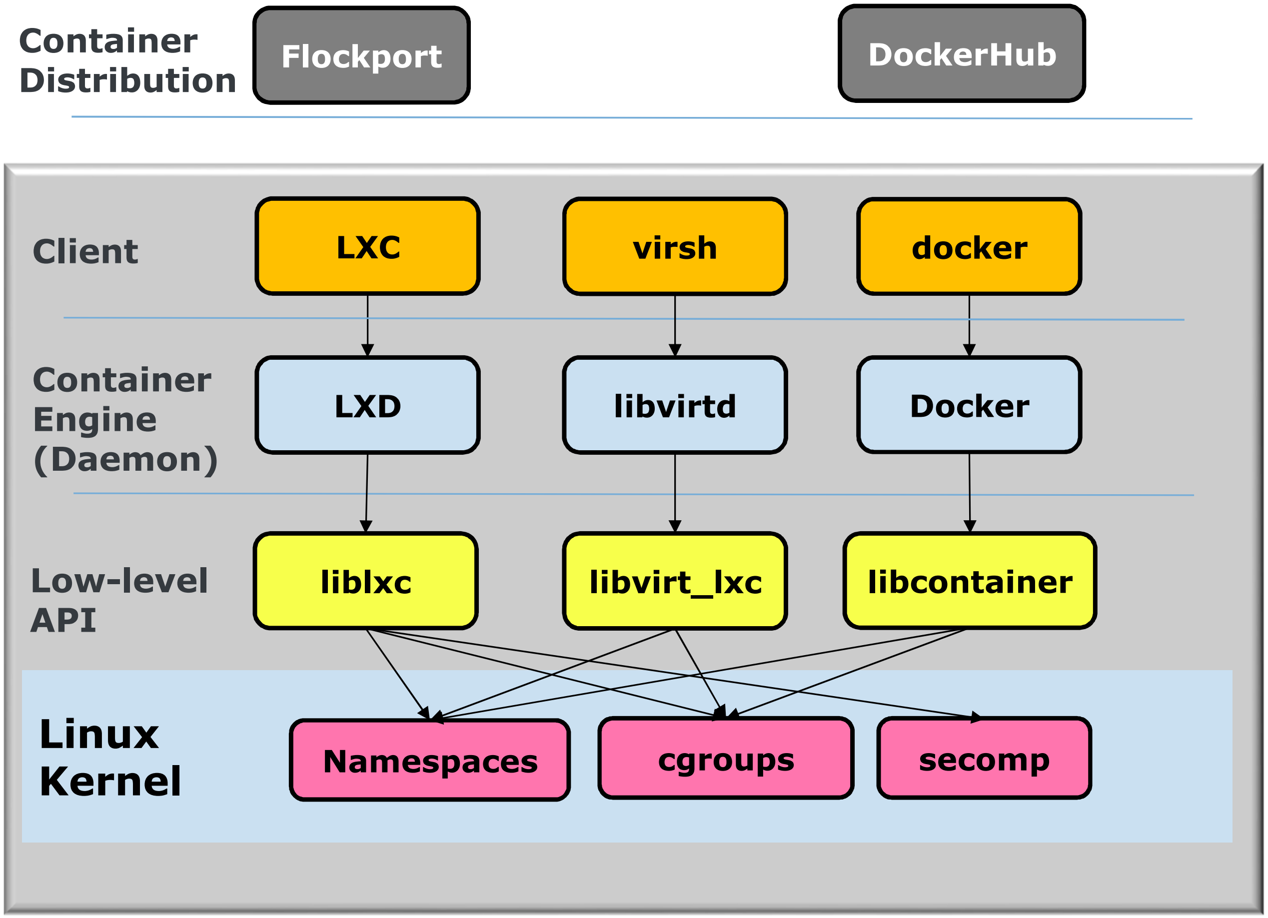


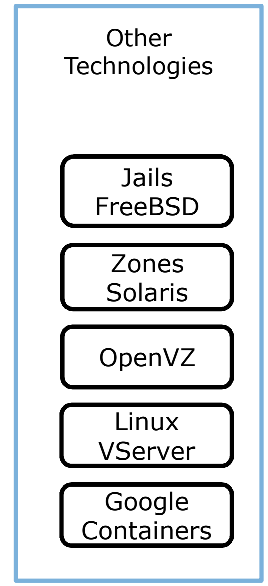
- A toolkit to interact with the virtualization capabilities of Linux (and other OSes / hypervisors)
- Goal: to provide a common and stable layer sufficient to securely manage domains on a node, possibly remote
- Has drivers for KVM/QEMU and Linux containers
- Many management applications supported
- http://libvirt.org/

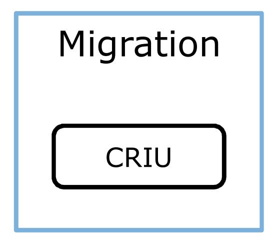
Linux Containers

- Platforms supported: all … not platform dependent
- Features
  - Technologies: LXC, Docker, Libvirt
  - Setups: Busybox system containers, application containers
  - Networking
    - Shared with host
    - Host interface assignment
    - Virtual Ethernet device pair
    - VLAN / MACVLAN
    - USDPAA
  - Security: capabilities, seccomp, user namespace
- Upstream status: upstream

Container Technologies







Container Comparison

LXC/LXD	Docker	Libvirt
• **Full system and application** containers • Focus on **performance and stability** • Lightweight Linux containers • Containers are like VMs with a fully functional OS • Comprehensive set of tools for container lifecycle management • **Data can be saved in a container** or outside • LXD allows you to use LXC to create containers on other machines • LXD aiming to used hardware that "**guaranteed isolation of containers**" on the chip level • Container distribution platform – Flockport.com • Developed by Ubuntu/Canonical **"Blindingly fast virtualization"**	• Single application virtualization engine based on containers • Focus on **ease of use**. Easy delivery of apps in a Docker container • Each application has its own container. "**Container as an app**" • Docker restricts the container to a **single process** only • Instances are ephemeral. **Persistent data is stored in host** • Trade off in complexity and constraints. Suitable for read only app that is 'frozen in state' • Container Distribution Platform – Docker Hub • Developed by PaaS providers (dotCloud) **"Great application delivery mechanism"**	• **Virtualization high-level API** with support for containers • Focus on **unification** with different virtualization technologies • Own version of container API – libvirt_lxc – tradeoff in order to fit the overall architecture of libvirt • Developed by Red Hat

Containers vs Hypervisors

	Linux Containers	Embedded Hypervisor	KVM
HW Support Needed?	No	Yes	Yes
Overhead	Low	Yes	Yes
Isolation	Good	High	High
Partitioning	Yes	Yes	Yes
Virtualization	Yes	Yes	Yes
Multi OS	No (Linux Only)	Yes Linux RTOS Bare-board 3rd Party OSs	Yes Linux Bare-board 3rd Party OSs
Features		Failover	Oversubscription
Licensing	Mainstream Open Source	Private Open Source	Mainstream Open Source Mature

QORIQ
LAYERSCAPE SDK

VIRTUALIZATION
STATUS & ROADMAP

SDK Virtualization Status

Technology	e500v2	e500mc	e5500	e6500	ARMv7	ARMv8
KVM-PPC	Up	Up	Up	Up		
KVM-ARM					Leveraged	Leveraging
FSL Hypervisor (bare metal)		Public Sources				
LXC	Leveraged & Fixes					Leveraging
Libvirt	Leveraged & Fixes				Leveraged	Leveraging

- 3 complementary solutions supported on multiple platforms
- Focus on enabling core virtualization support, upstreaming and good OOB experience in NXP SDK
- I/O performance optimization in progress

Layerscape ARM®v8 Virtualization Roadmap

LS1043 BSP 0.5
[KVM] Basic KVM
[KVM] virtio
[LXC] LXC Support

LS2080 SDK 1.0
[KVM] Kernel 4.1 & QEMU 2.4
- MSI support
- vhost-net, vhost-blk dataplane

[LXC] LXC
[Libvirt] Libvirt

LS2 POC
[KVM]
- *DPAA 2.x Direct Assignment with interrupts*
- *KVM-RT on LS2*

[Docker] Docker engine on LS2

LS2 EAR 0.6
[LXC] LXC with seccomp
[Docker] Docker engine

Benchmarking
***[KVM]** Benchmarking*

POC
***[KVM]** PCI device assignment to KVM VMs*
***[LXC]** CRIU Demo*

SDK 2.0
[KVM]
- *DPAA 2.x Direct Assignment v1 (VFIO)*
- *PCI Device Assignment (VFIO-PCI)*

Upstreaming
***[KVM]** DPAA 2.x Direct Assignment v2 (VFIO)*

POC
[KVM]
- *LS2 User Space (e.g ODP) direct assignment in guest*
- *Performance improvements*
- *LE Guest on BE Host*
- *BE Guest on LE Host*
- *Direct assignment of SEC*
- *DPAA 2.x vhost-user optimizations*
- *Live Migration Demo*

Upstreaming
***[KVM]** DPAA 2.x Direct Assignment v2 (VFIO)*

POC
[KVM]
- *Performance improvements*
- *Direct assignment of platform devices*
- *virtio-crypto demo*
- *USB pass-through*
- *Shared-memory (ivshmem)*

2015 | **2016** 1Q | 2Q | 3Q

Italic features depend on upstream support

Color Legend
- Released
- Roadmap Date
- Current Release
- Major Release

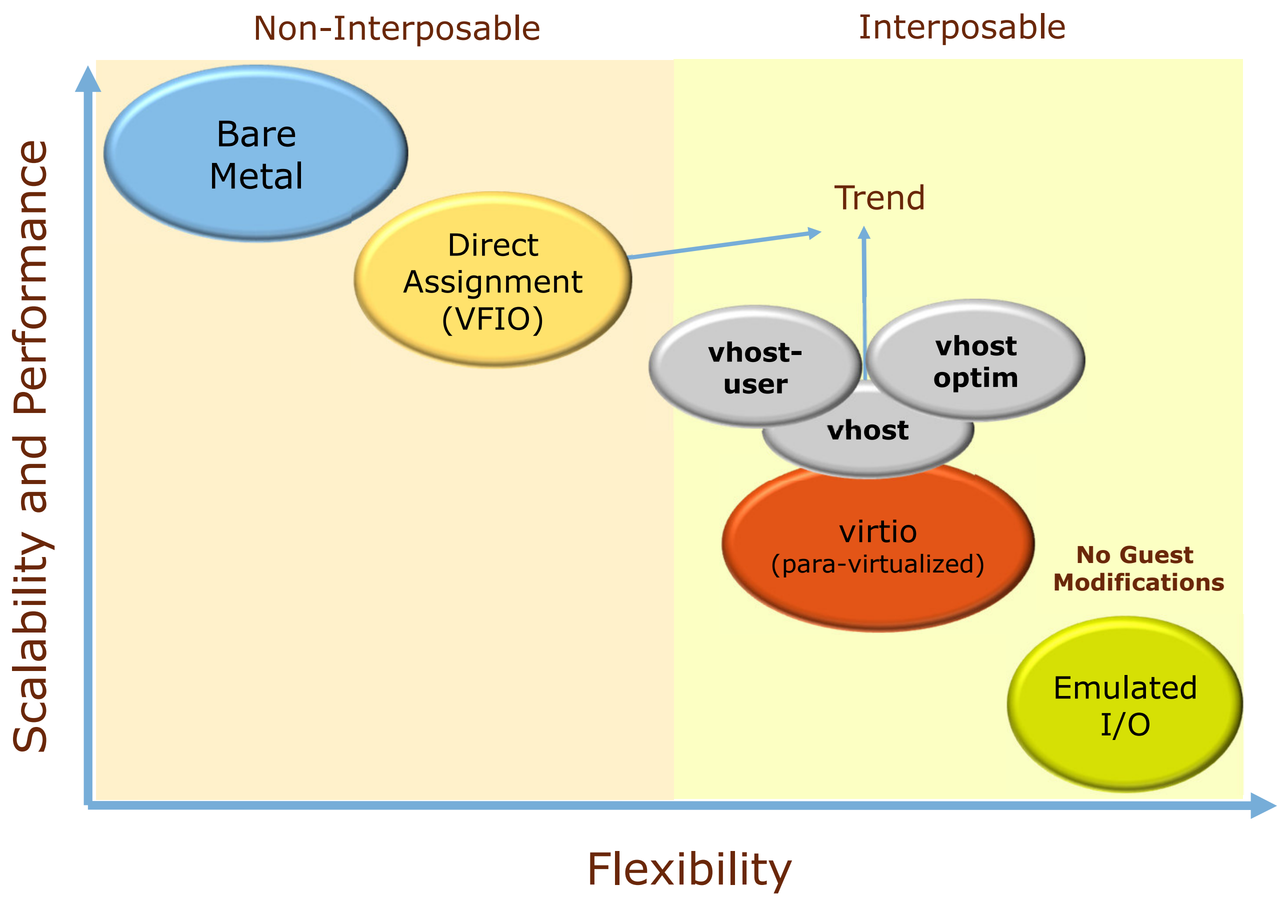
KVM - Out-Of-Box RFS Enablement

- Required components :
  - KVM support enabled in kernel
  - Guest image
  - Guest root file system
  - QEMU

	Kernel config	Guest rootfs	Guest image	QEMU
`fsl-image-core`	NO	NO	NO	NO
`fsl-image-full`	NO	NO	NO	YES
`fsl-image-virt`	NO	YES	YES	YES

I/O IN KVM ENVIRONMENTS

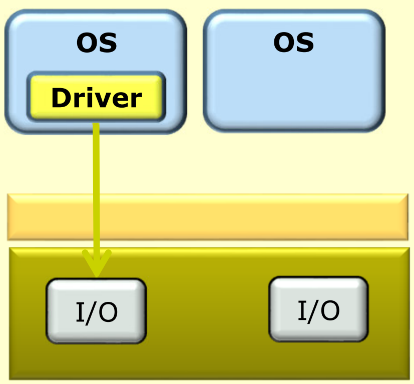
I/O Virtualization - Performance vs Flexibility



Device Usage in Virtual Environments

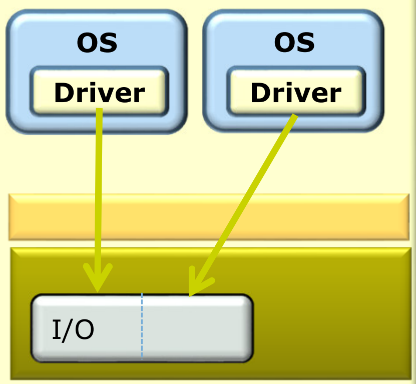
Direct Access

- Fast native performance
- Direct access to hardware



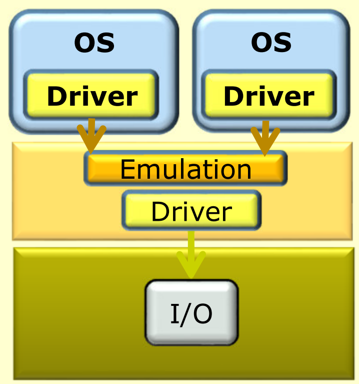
Partitionable HW

- Hardware partitioned
- One hardware block



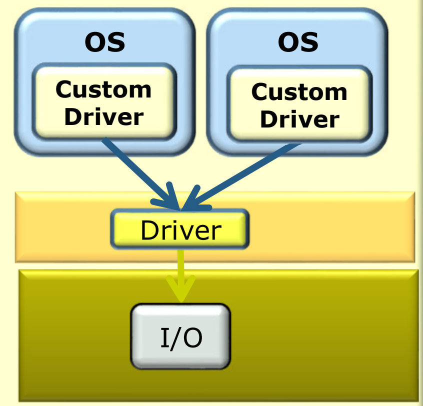
Emulated

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



Para-Virtualized

- Driver in Hypervisor
- Modified Drivers in Guest



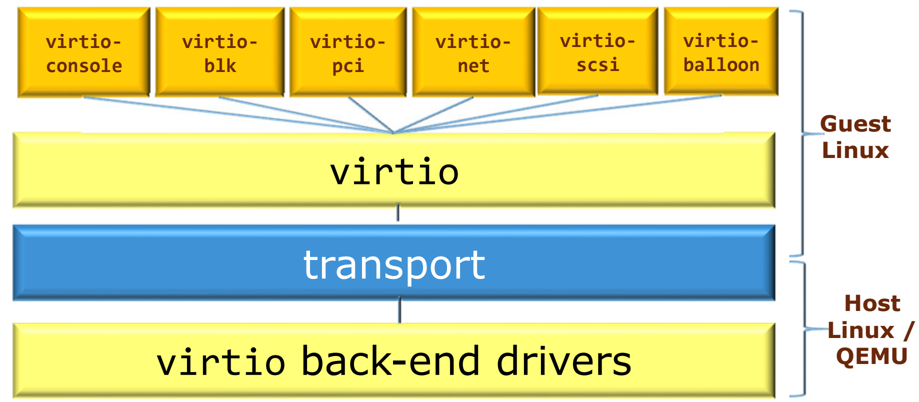
Hardware/software access
Hypercalls
Traps

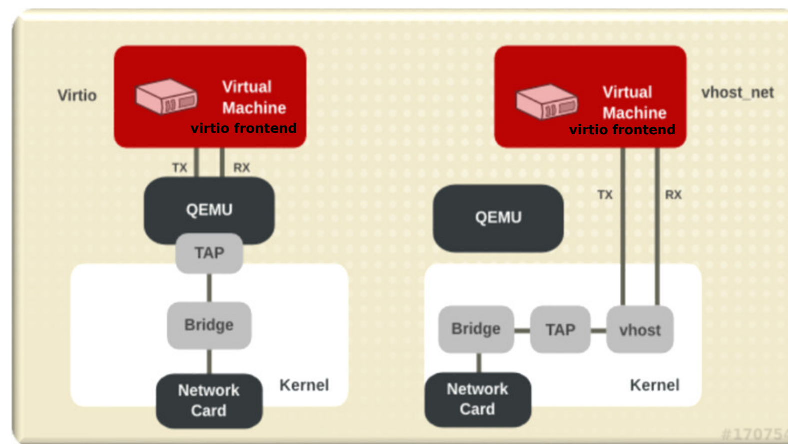
DEVICE VIRTUALIZATION

VIRTIO

virtio

- Device abstraction layer of para-virtualized hypervisor
  - Standard for VMs/VNFs
  - Appearance as physical devices
  - Uses standard virtual drivers and discovery mechanisms
    - `virtio-net` : Ethernet virtual driver
    - `vhost-net` : optimizes Ethernet virtual driver by eliminating QEMU context switch
    - `virtio-pci`
- Backend drivers are vendor specific in host Linux; transparent to VM/VNFs





Sources: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Virtualization_Tuning_and_Optimization_Guide

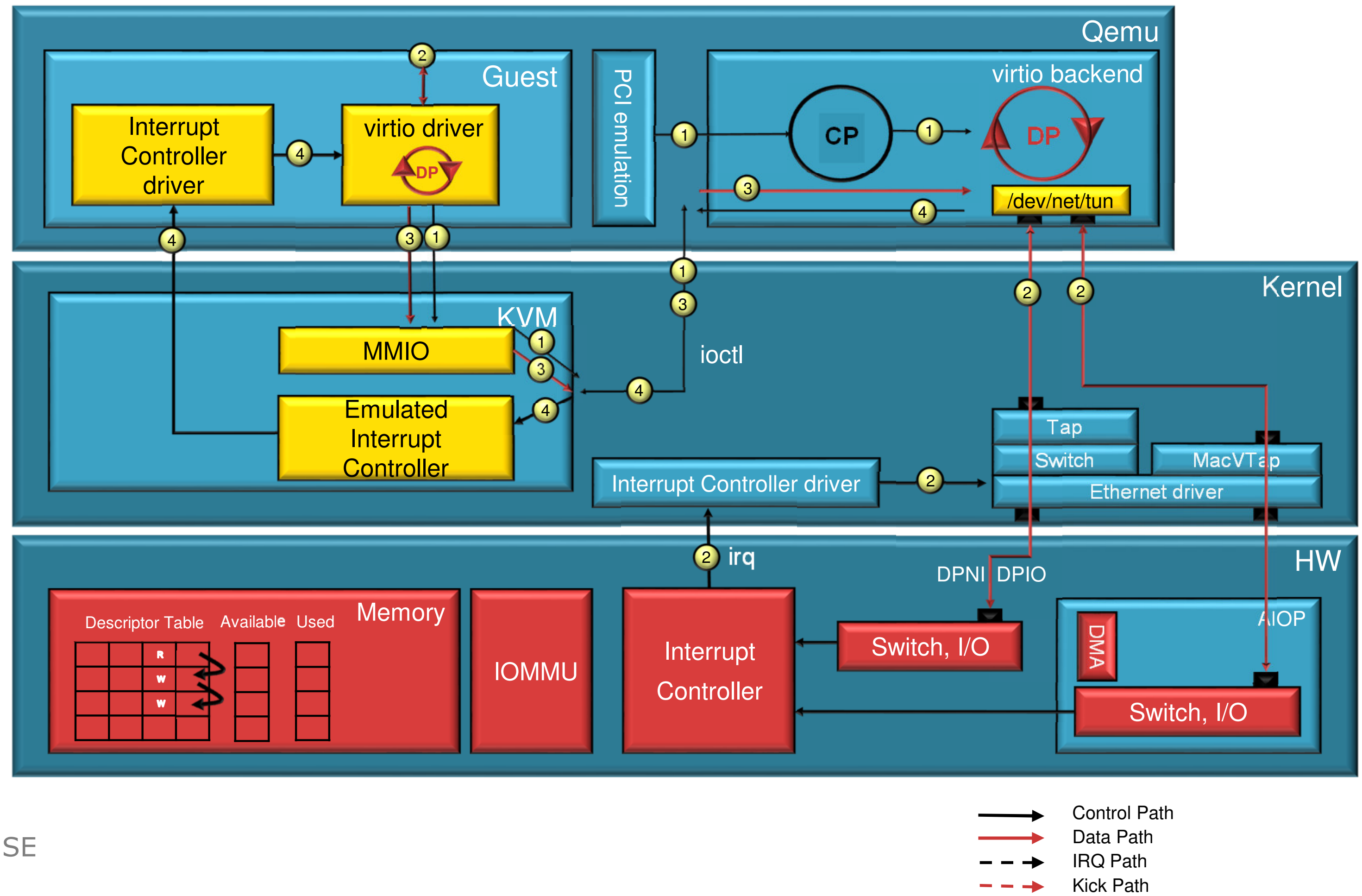
virtio

- Device facilities
  - Device status field / Feature bits / Device Configuration space / virtqueues
- Transport protocols: PCI, MMIO
- virtio specification, defined by OASIS technical committee
  - Straightforward - use normal bus mechanisms for interrupts and DMA
  - Efficient - rings of input/output descriptors
  - Standard - 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

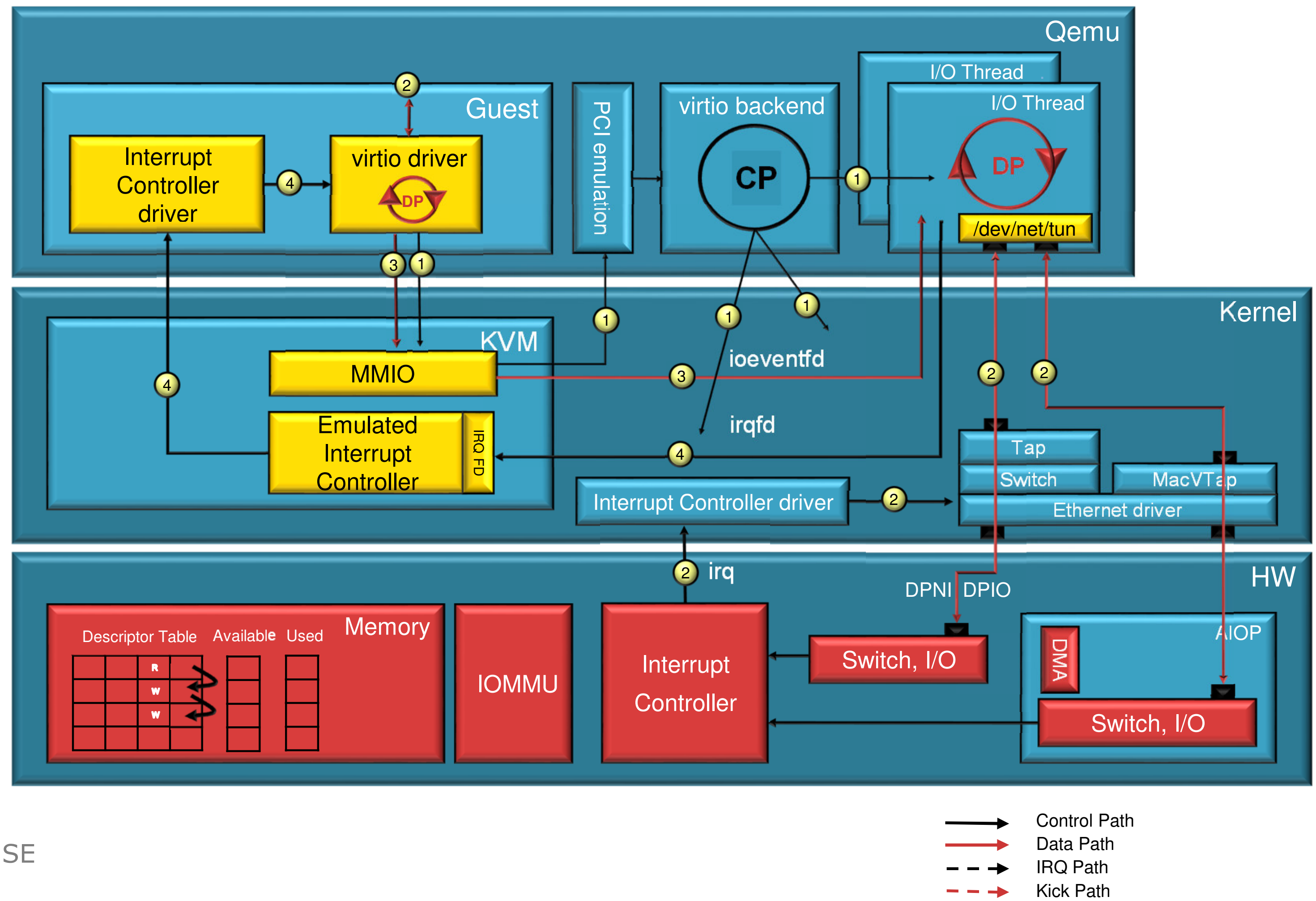
KVM/Qemu `virtio` Back-end Drivers

virtio front-end	Qemu/KVM back-ends	
`virtio-net`	`virtio-net (legacy)`	Qemu
	`virtio-net (data-plane)`	Qemu, I/O thread
	`vhost`	Kernel
	`vhost-user`	User space
`virtio-blk`	`virtio-blk`	Qemu
	`virtio-blk data-plane`	Qemu, I/O thread
`virtio-scsi`	`virtio-scsi`	Qemu
	`vhost-tcm`	Kernel

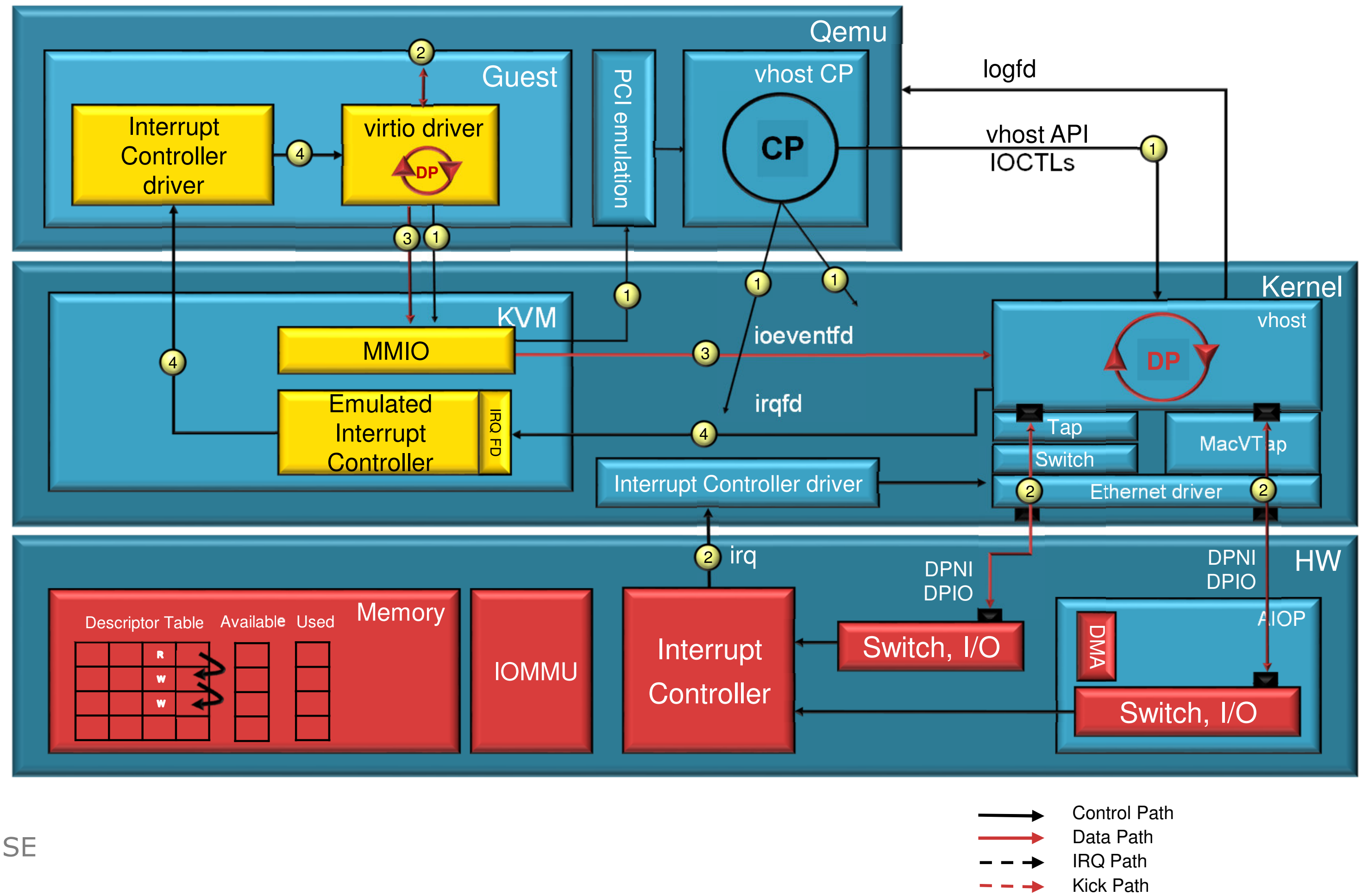
Back-end - virtio-net (legacy)



Back-end - virtio-net (data-plane)

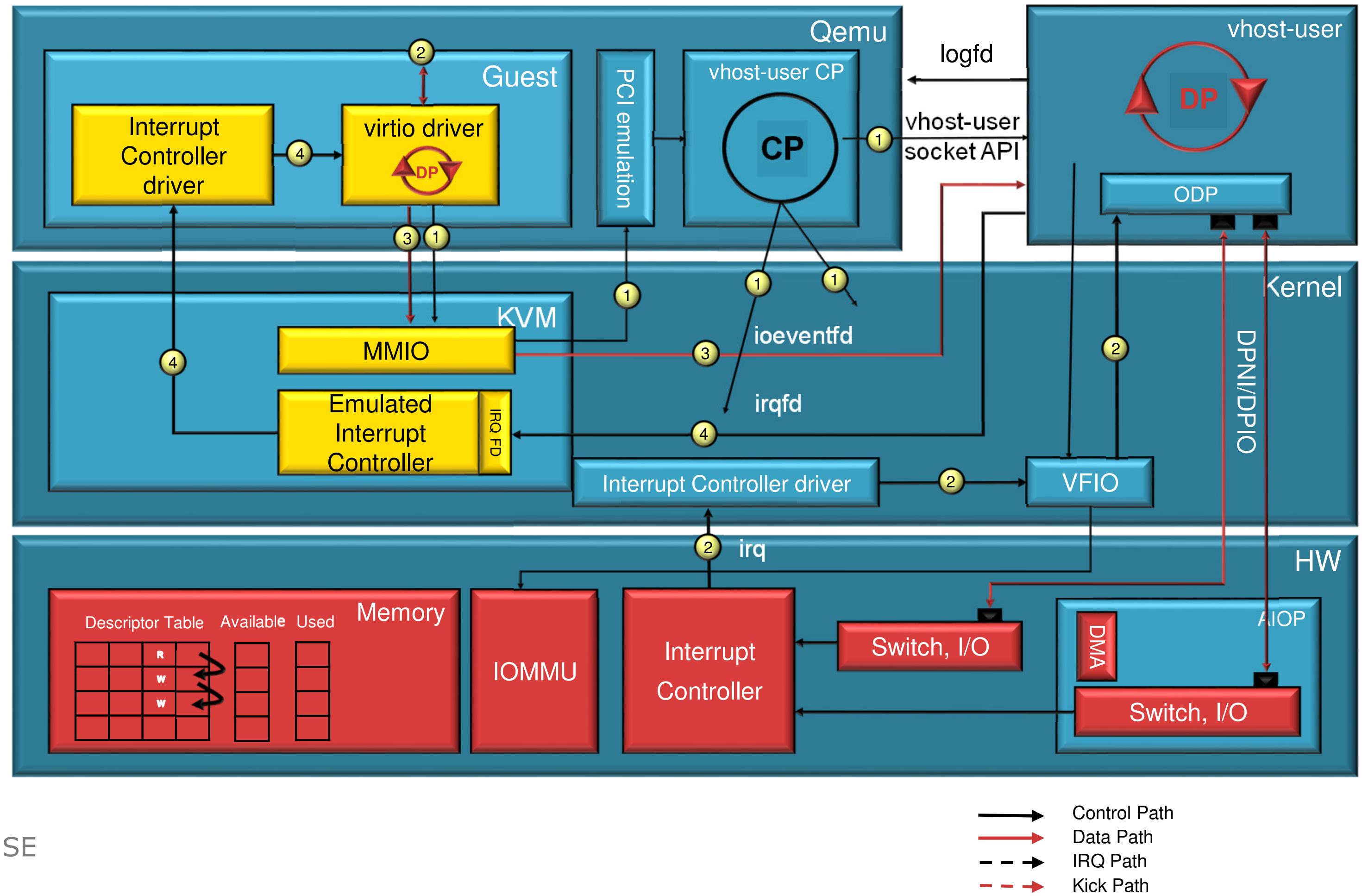


Back-end - vhost

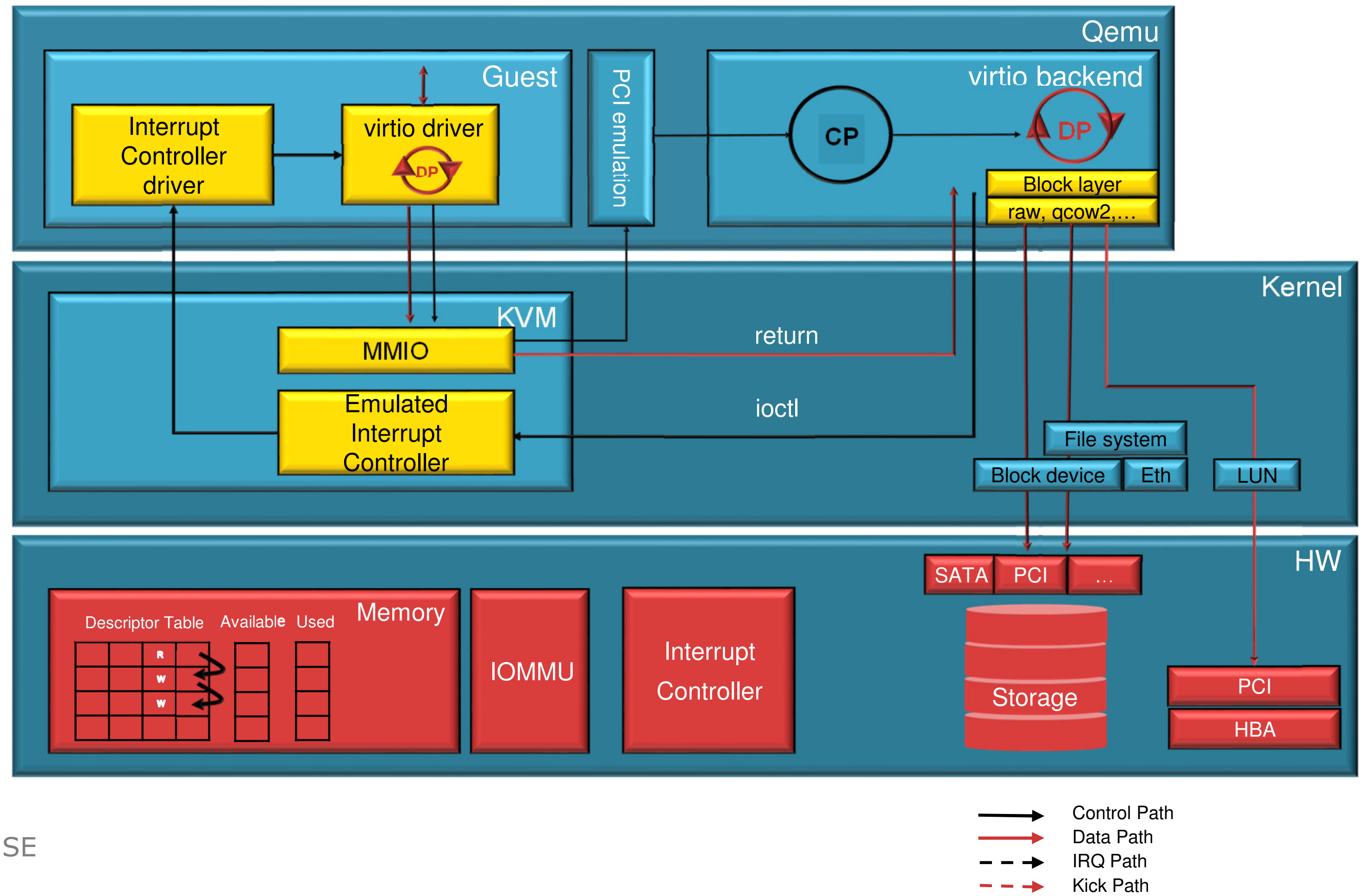


 EXTERNAL USE

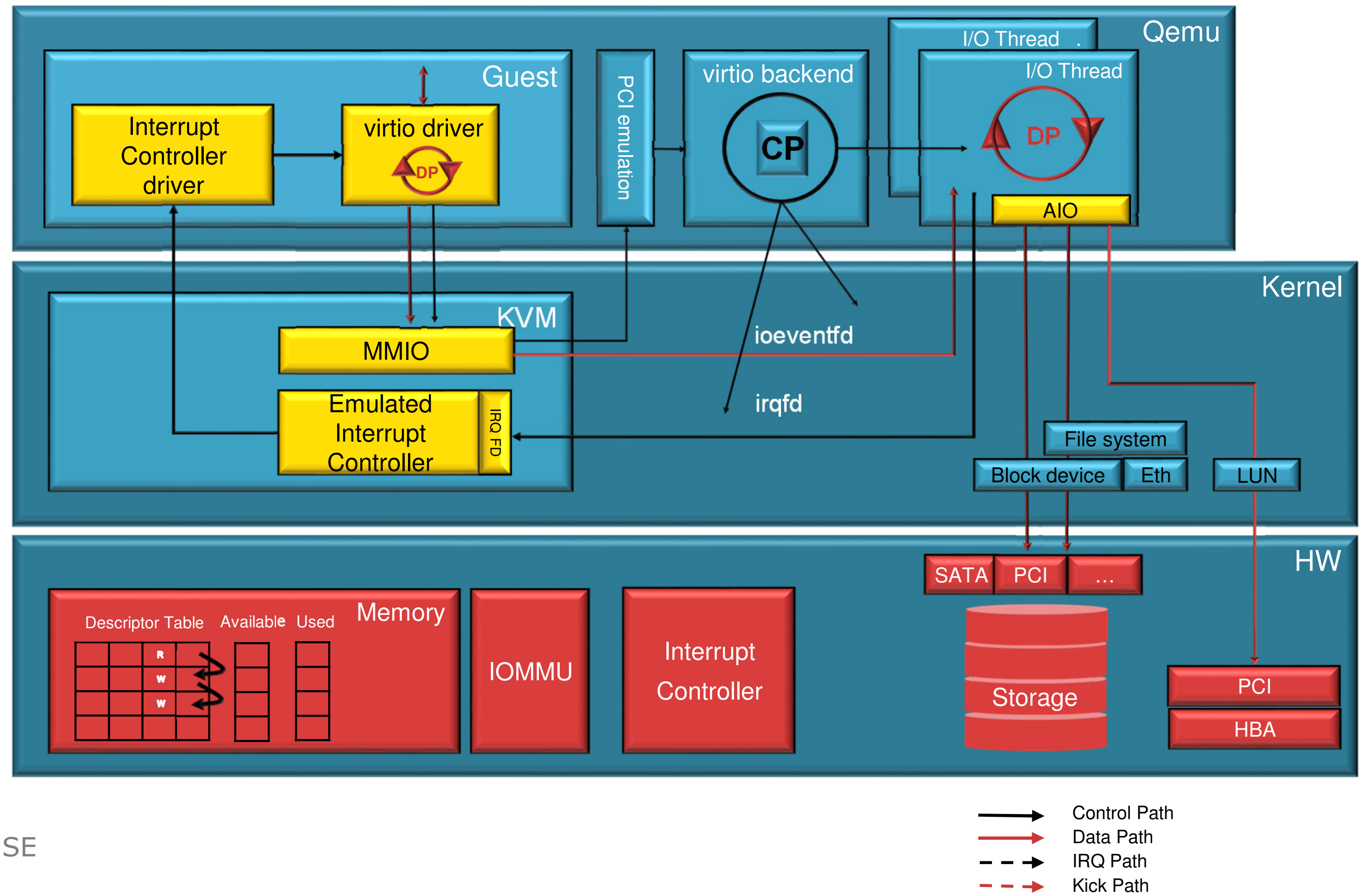
Back-end - vhost-user



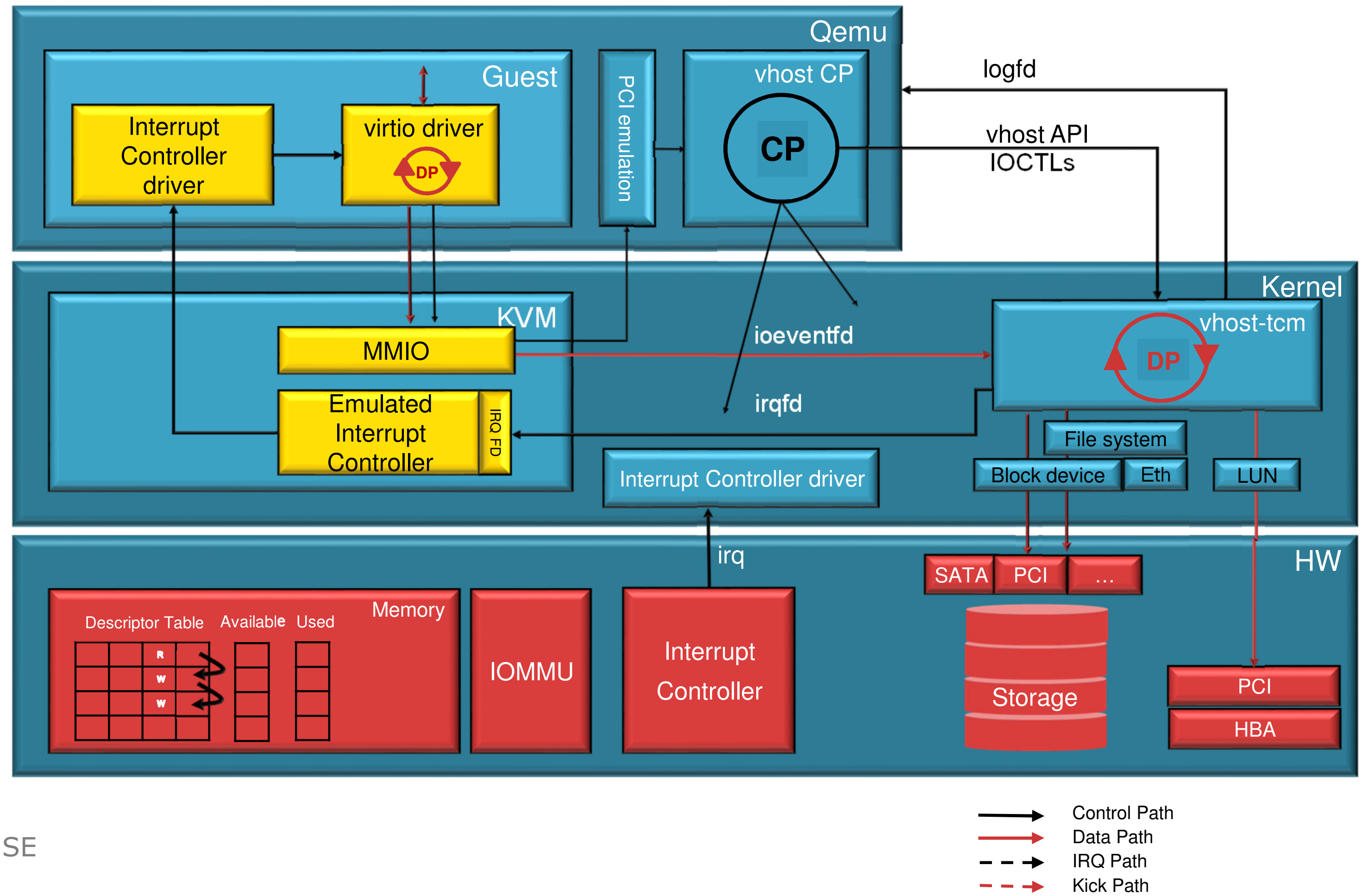
Back-end - virtio-blk



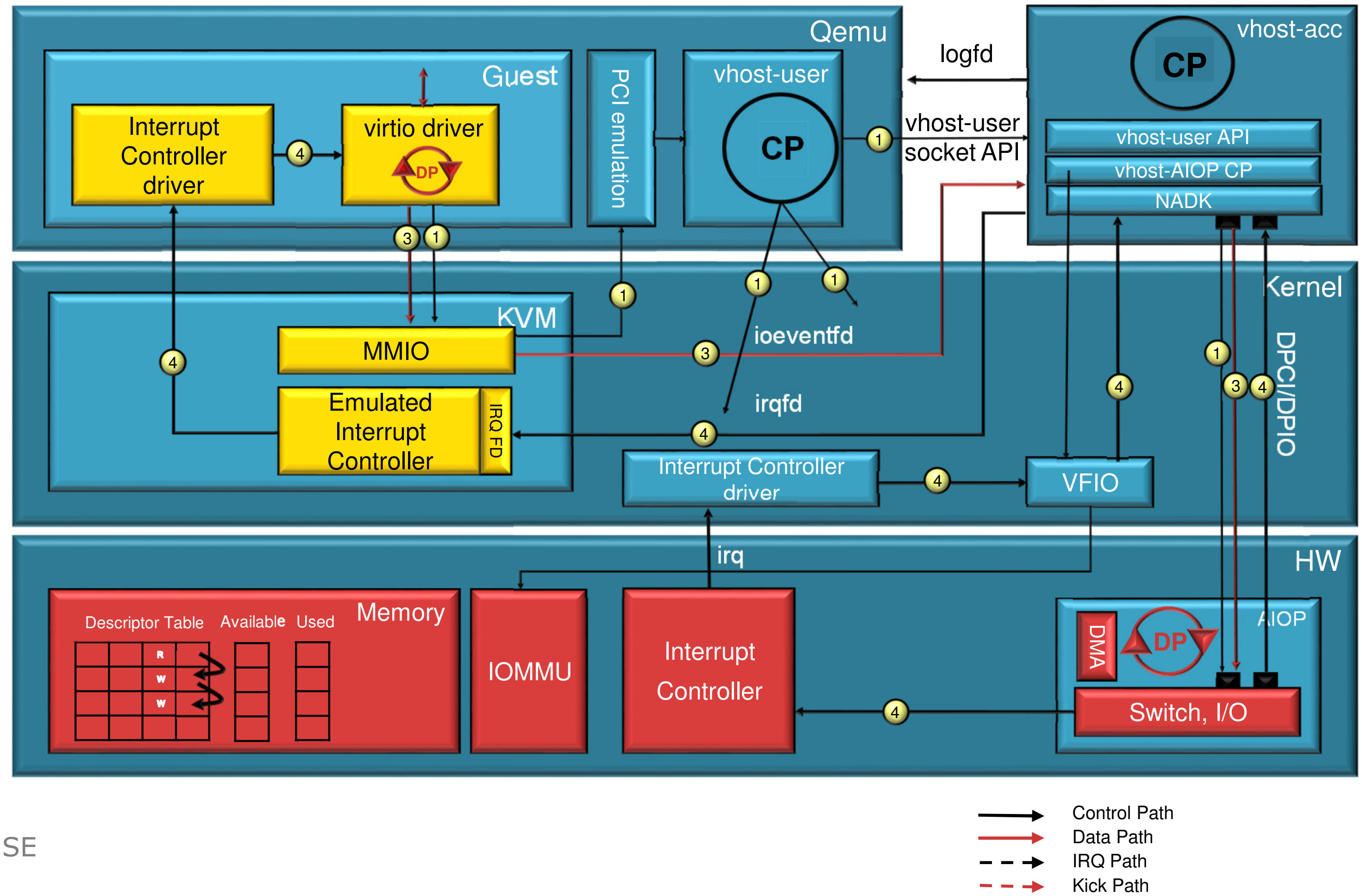
Back-end - `virtio-blk` data-plane



Back-end virtio-scsi - vhost-tcm



Back-end Acceleration - vhost-acc (preliminary)



 EXTERNAL USE

DEVICE DIRECT-ASSIGNMENT

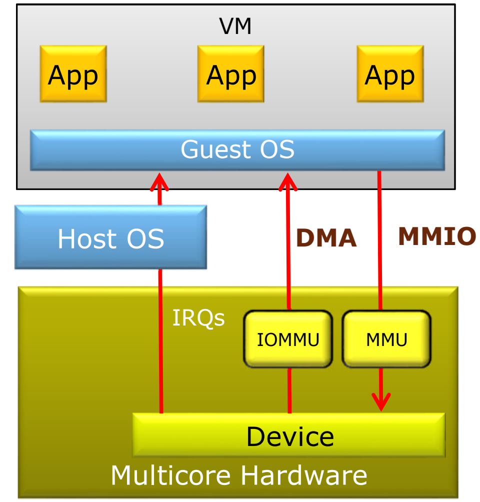
VFIO

Device Direct-Assignment

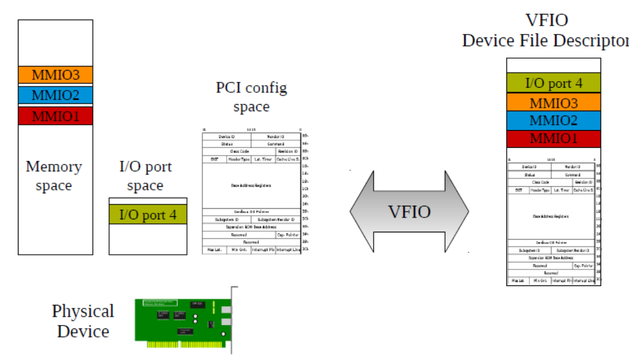
- Device drivers access from user-space
  - Device pass-through (`libusb`, `libscsi`)
  - Map /mem (not recommended)
  - UIO (User-space I/O)
    - Device access (`mmap` device MMIO regions)
    - Interrupt support
    - No isolation or translation
  - VFIO (Virtual Function IO)
    - Linux user space driver infrastructure for DMA devices
    - Device access (`mmap` device MMIO regions)
    - Enforces IOMMU translation and isolation (`iova` to real address)
    - High performance interrupt support (INTx, MSIs & MSI-X)

VFIO

- VFIO (Virtual Function IO) : Linux user space driver infrastructure
  - Enforces IOMMU protection
  - Device access : `mmap()` device MMIO regions
  - IOMMU programming interface
  - High performance interrupt support
  - Bus support : PCI, platform devices, LS2 MC bus
- VFIO PCI - abstracts devices as :
  - Regions :
    - PCI configuration space
    - MMIO and I/O port BAR spaces
    - MMIO PCI ROM access
  - IRQs include
    - INTx (legacy interrupts)
    - Message Signaled Interrupts (MSI & MSI-X)

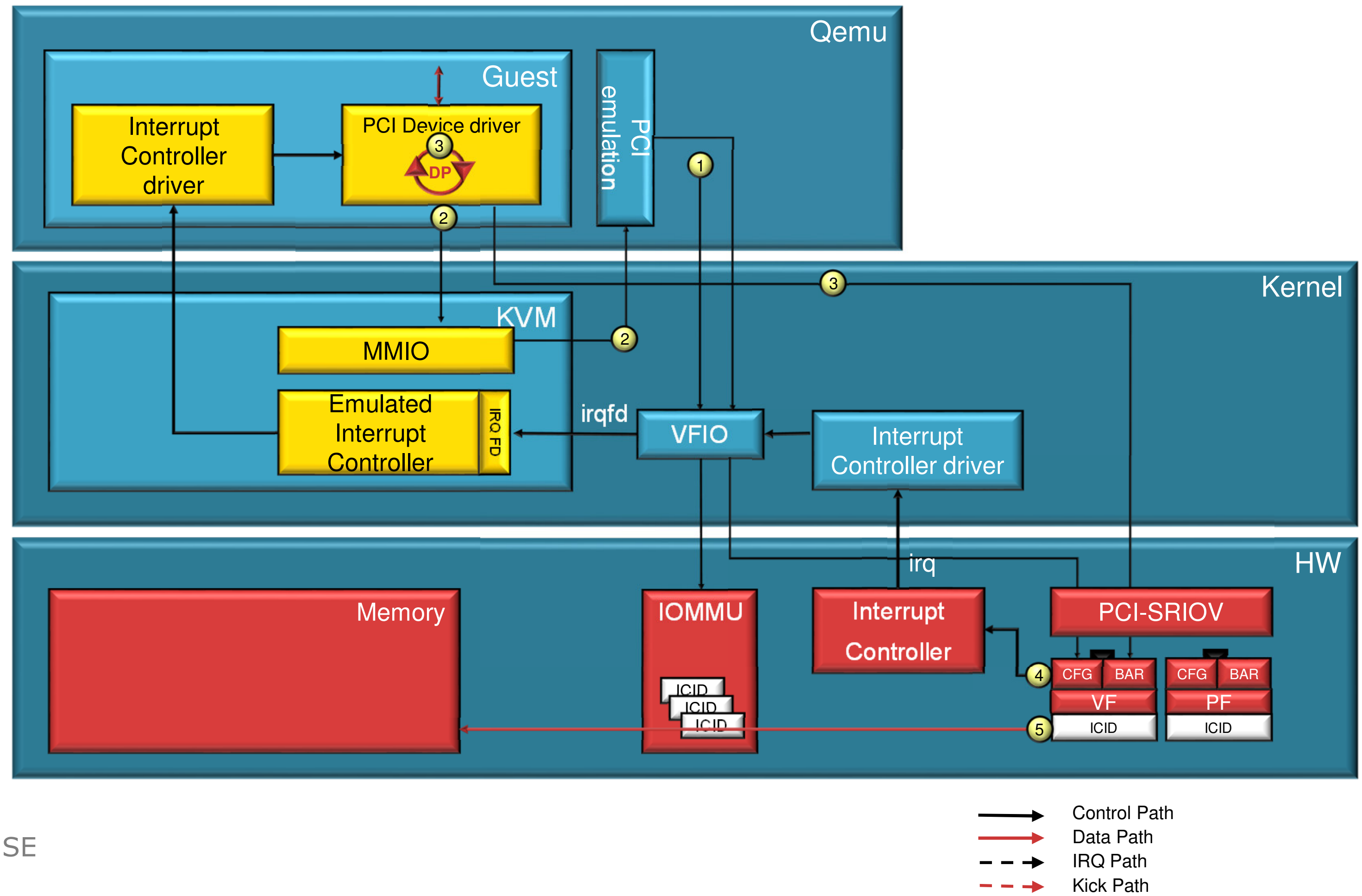


VFIO Device Decomposition



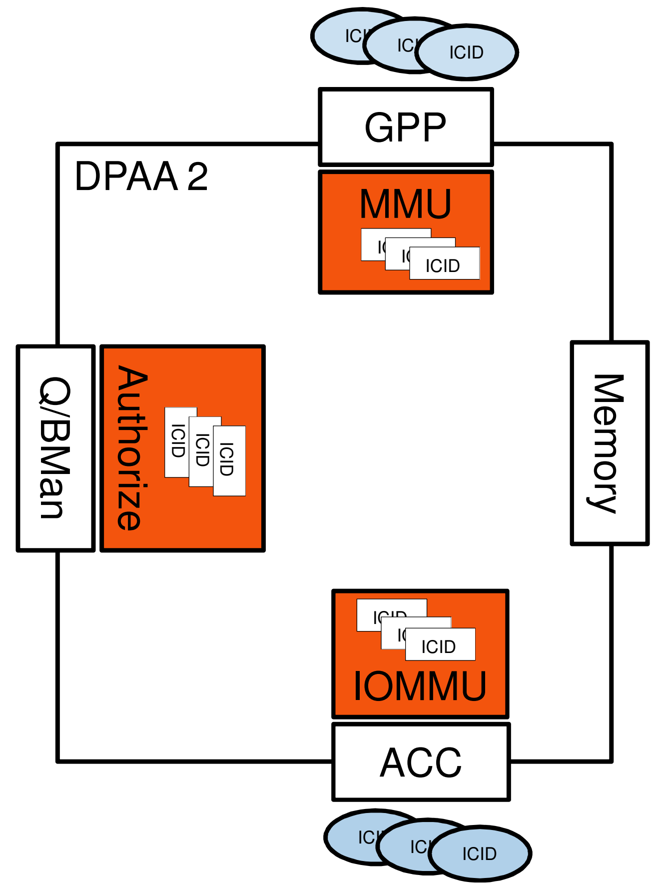
Source: www.linux-kvm.org/wiki/images/e/ed/Kvm-forum-2013-VFIO-VGA.pdf

VFIO for PCI Bus

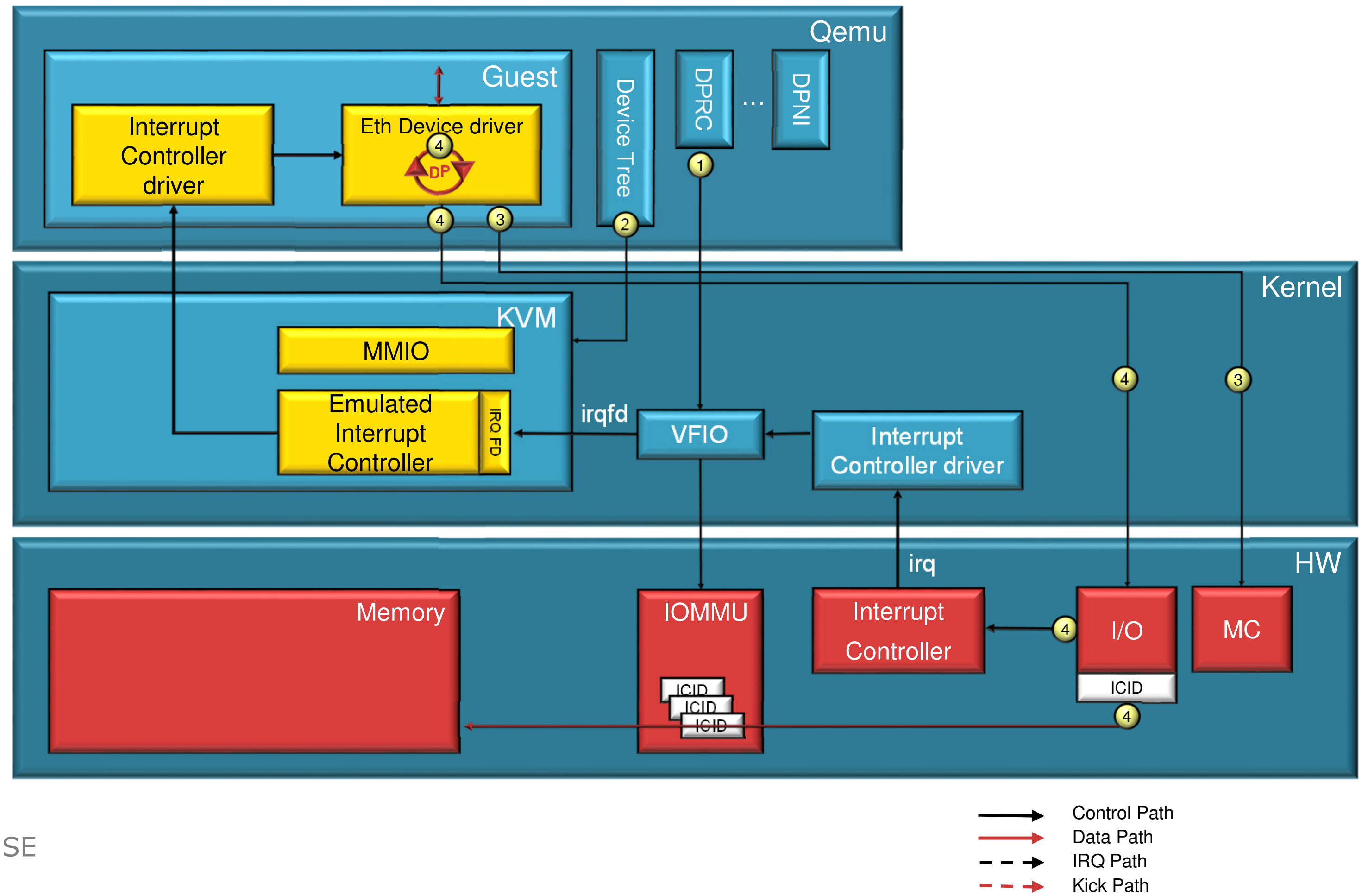


DPAA 2 Secure Direct Assignment

- DPAA 2 architecture
  - Optimized for resource assignment to various software contexts through Management Complex
    - Linux MC bus
    - Resource management tool
  - IOMMU translation and protection for user-space (ODP and QEMU)
    - ICID (StreamID)
    - MC bus integration with VFIO
    - Device reset
  - DPAA secured with Authorization Tables



VFIO for MC Bus



virtio vs Direct Assignment (VFIO)

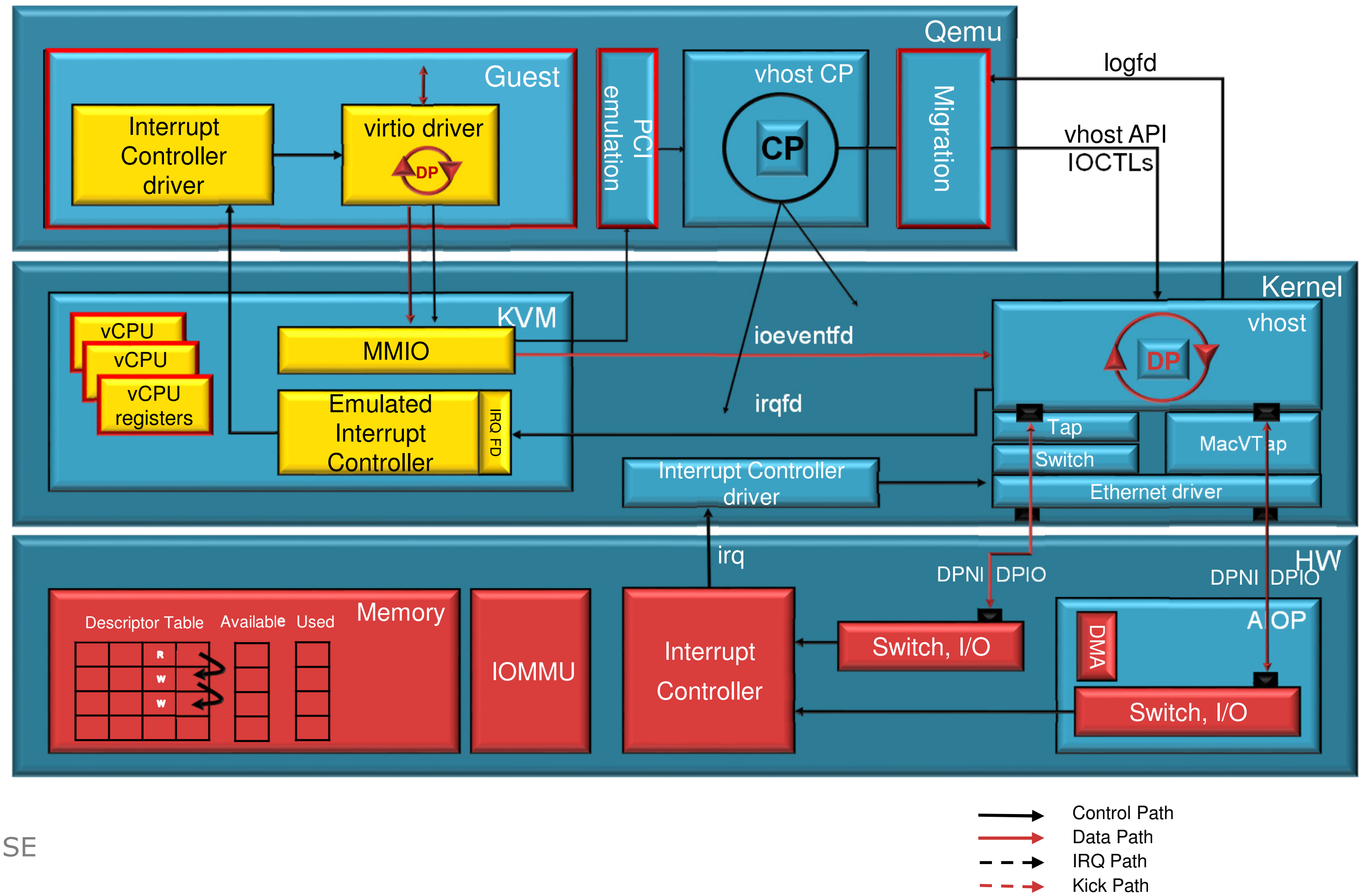
	virtio	Direct Assignment
Flexibility	High	Med
Guest Driver	Generic	HW dependent
Device Sharing	Yes	No
Live Migration	Yes	PoC prototyping
Performance	Medium	High
Processing	Backend is SW emulated in Host or in Firmware	Reduced processing in Host
HW support for isolation	No	Required (SMMU)
Licensing	Open Source*	Open Source
Upstreamable ?	Firmware accelerations - NO	YES
History	Started as software implementation in Linux and now API is standardized (OASIS) Standard add-ons may not be accepted in Linux upstream.	Framework implemented in Linux for PCI devices that is extended for Platform devices.

Q & A

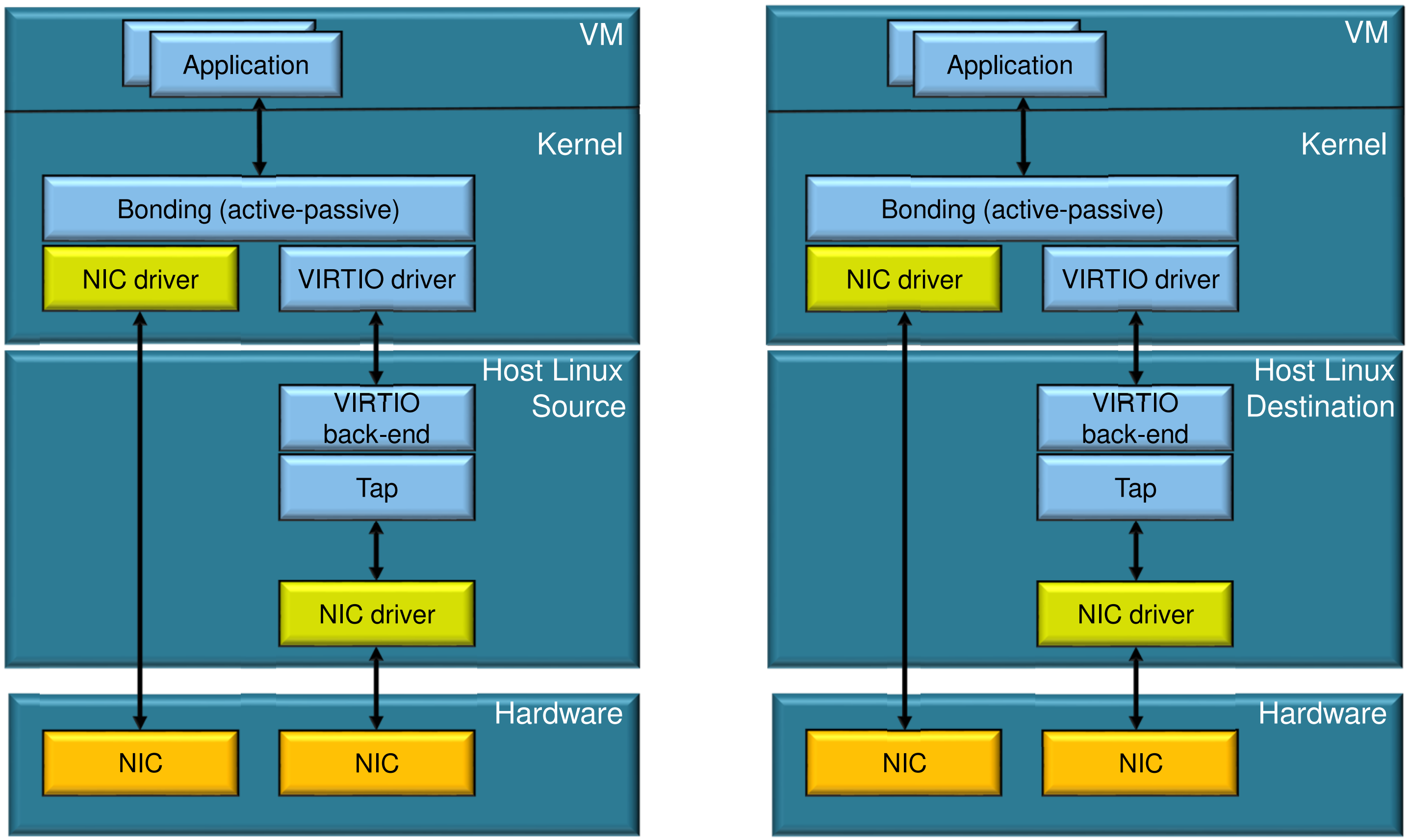
SECURE CONNECTIONS
FOR A SMARTER WORLD

VM LIVE MIGRATION

VM Live Migration with virtio devices

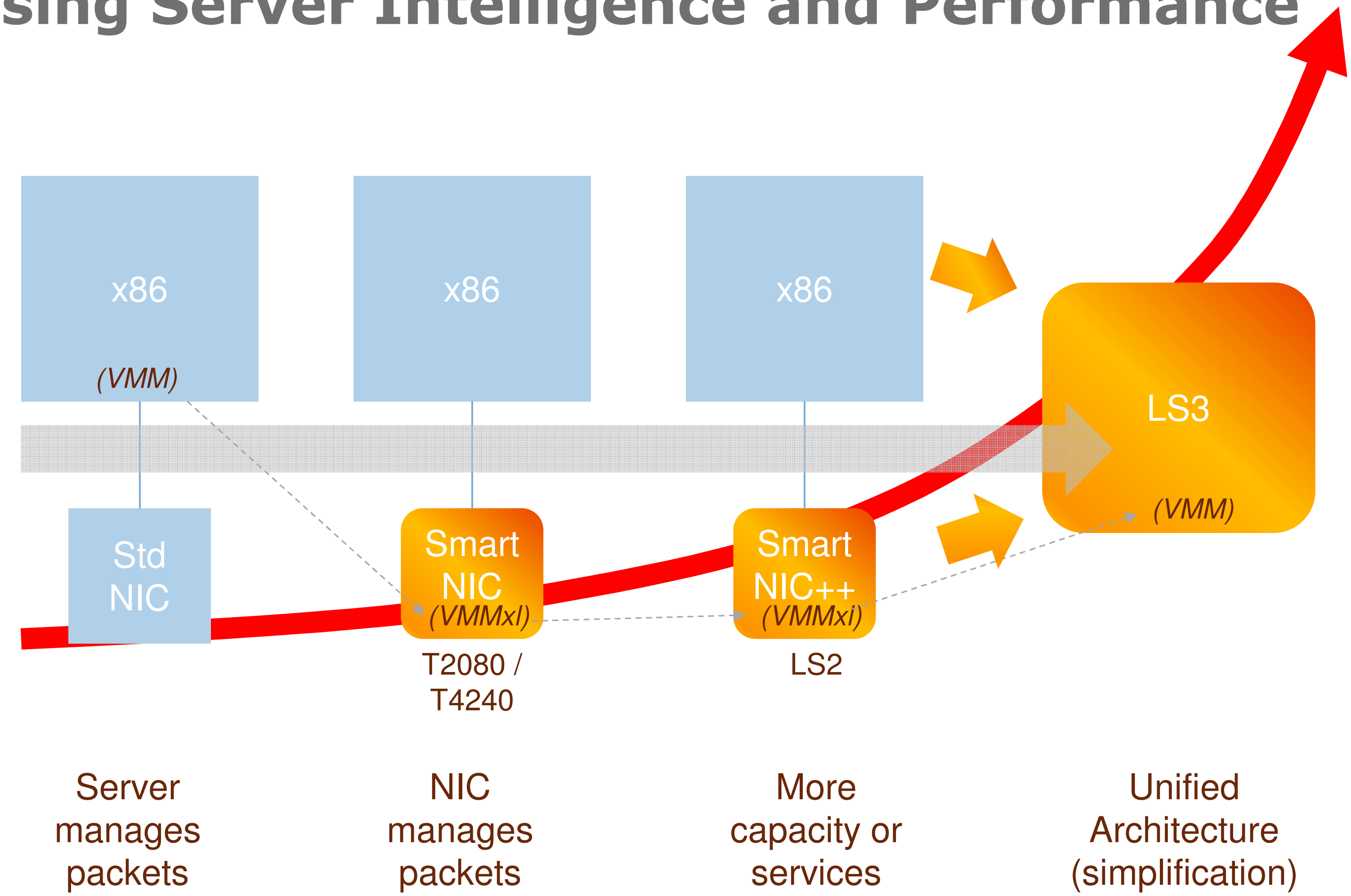


VM Live Migration with Direct Assignment

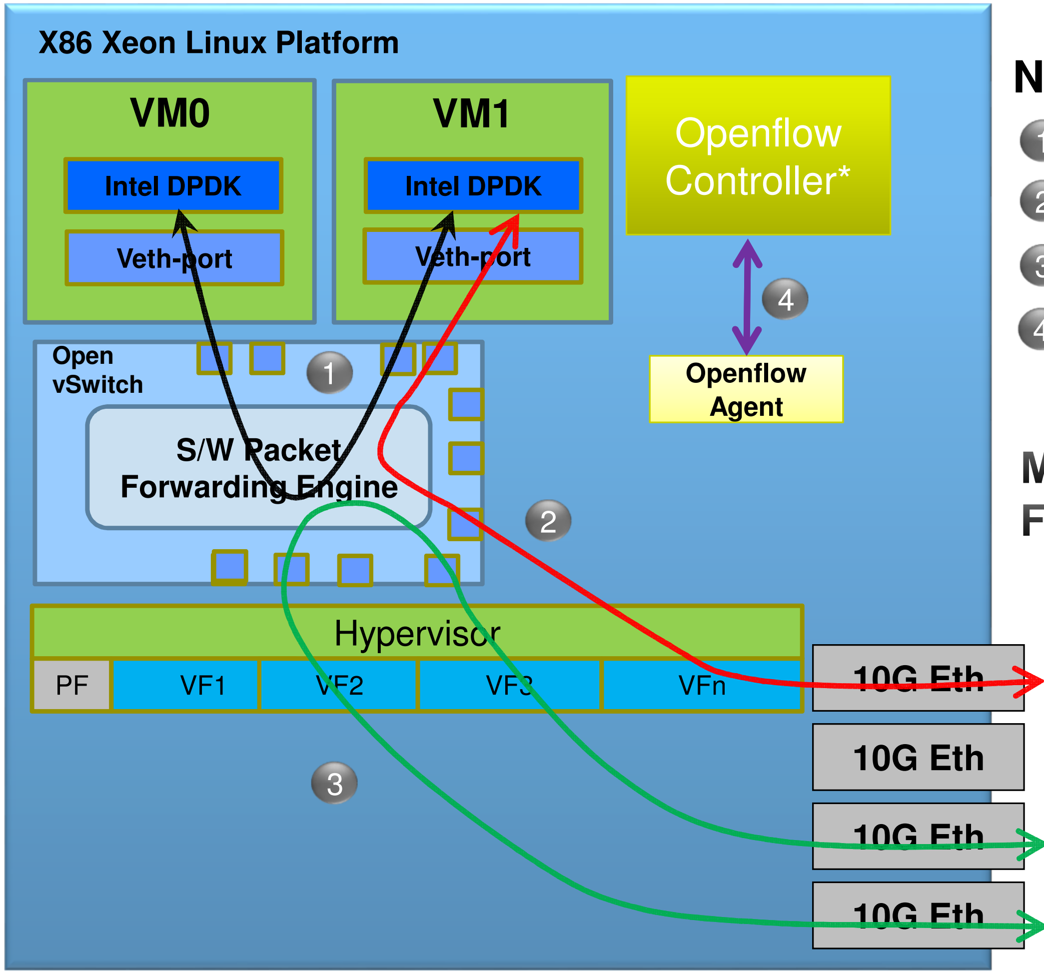


DATACENTER – SERVERS INIC SOLUTIONS

Increasing Server Intelligence and Performance



Server non-iNIC case - iNIC demo Traffic Flow



* Can be external

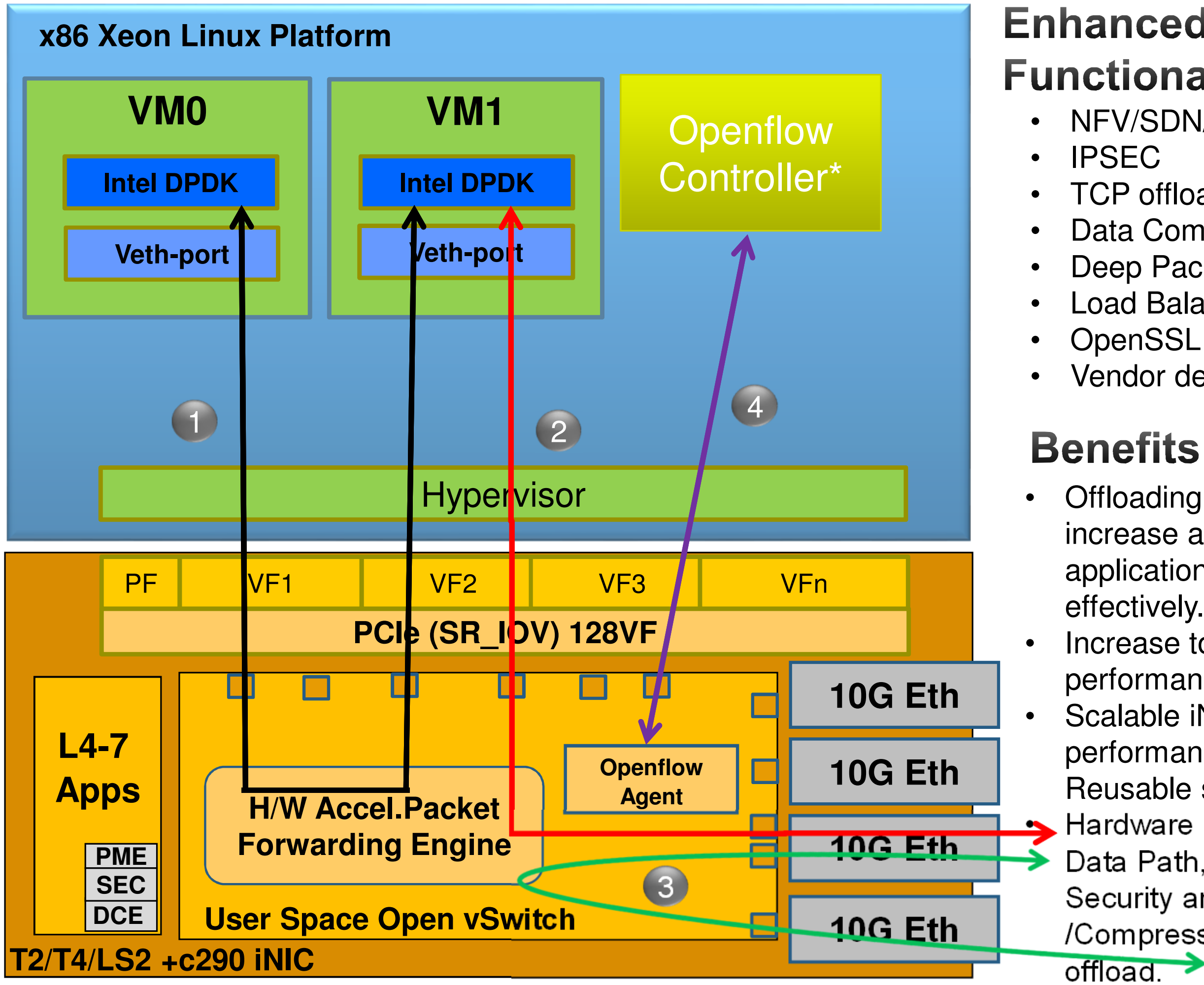
NIC Traffic Flow

1. VM-to-VM
2. VM-to-ethernet
3. ethernet-to-ethernet
4. Openflow Control Plane path

Main Components and Functions

- DPDK NIC driver
- Support for up to 128 VF
- Kernel Space vSwitch
- Openflow Agent for traffic management

Server non-iNIC case - iNIC demo Traffic Flow



* Can be external

Enhanced L4-7 Functionality

- NFV/SDN/Firewall/ACL
- IPSEC
- TCP offload
- Data Compression
- Deep Packet Inspection
- Load Balancing
- OpenSSL + record offload
- Vendor defined applications

Benefits

- Offloading of x86 CPU to increase aggregate with application performance cost effectively.
- Increase top end server performance
- Scalable iNIC platform performance T2080 to T4240. Reusable software.
- Hardware acceleration for Data Path, Pattern Matching, Security and Decompression /Compression, PKC/Record offload.

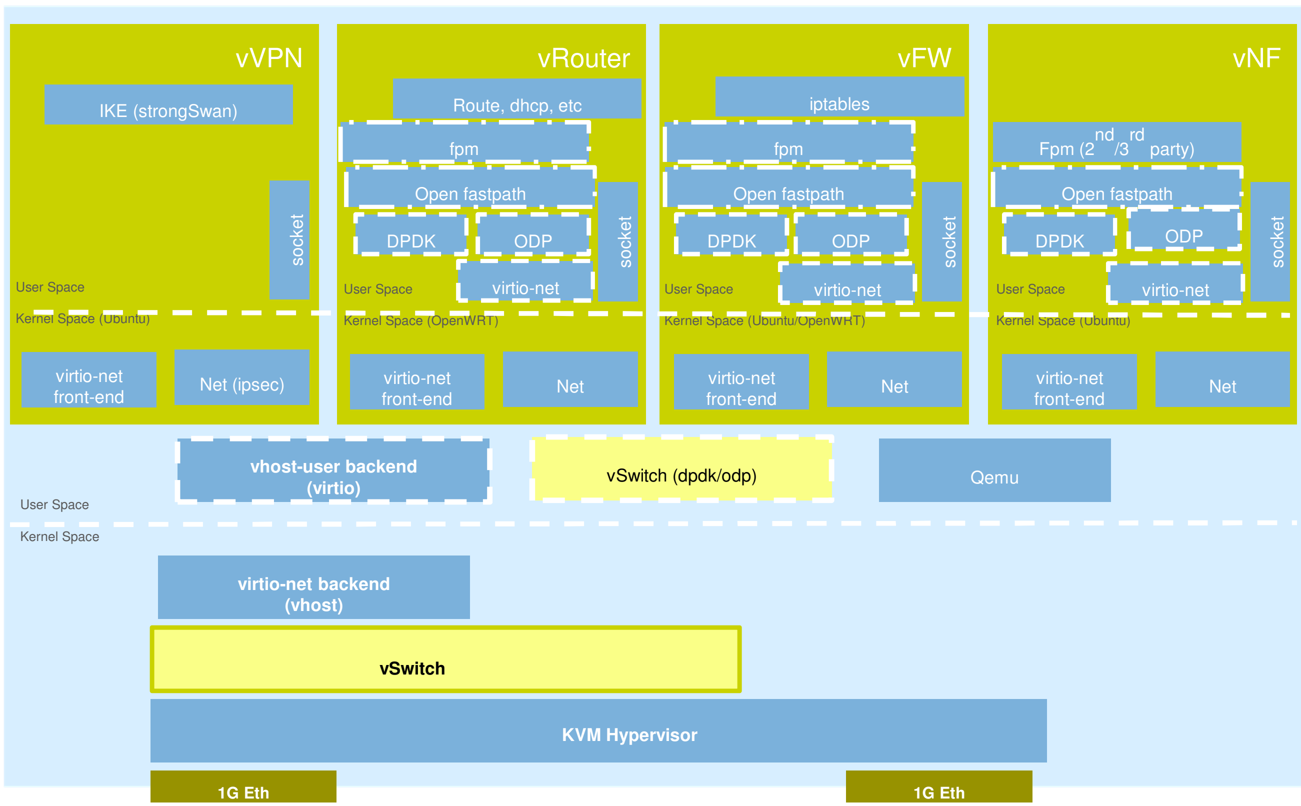
VCPE / EDGE NETWORK

VNFS

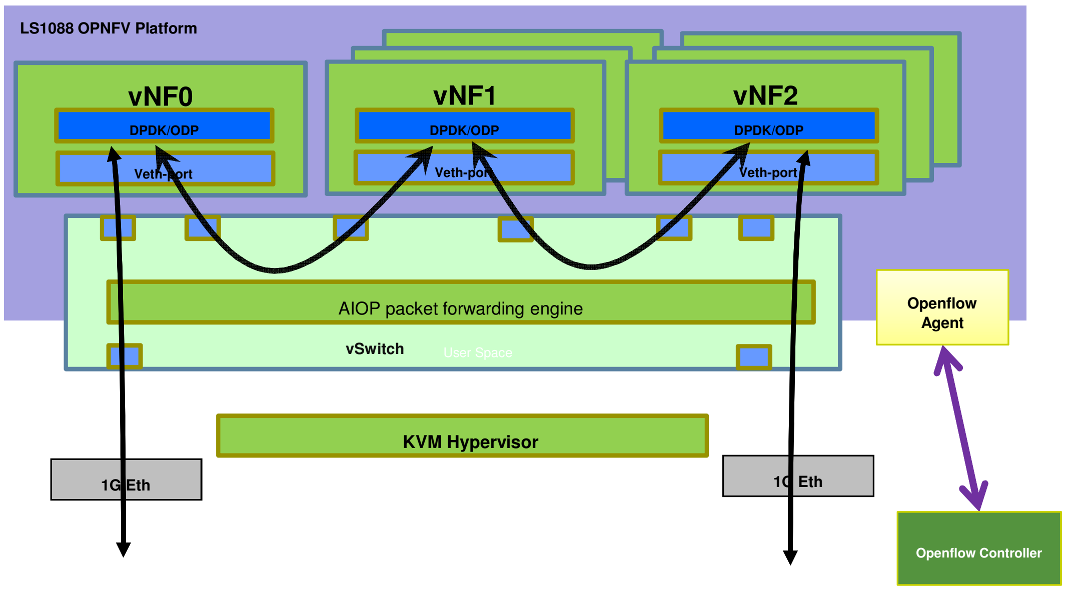
-

VIRTUALIZED NETWORKS FUNCTIONS

vCPE Solution (OPNFV Platform + vCPE functions)



vCPE Solution
(LS1088/LS2088 ARM v8 ISA + Advanced Packet Processing)



- ✓ Up to 8 GPP cores available for vNFs
- ✓ Full OPNFV platform compliance.
- ✓ vNFs 100% source compatible with x86
- ✓ 1 to 8x performance scalability on a single software platform
- ✓ AIOP packet forwarding engine frees up GPP cores AND significantly improves network throughput

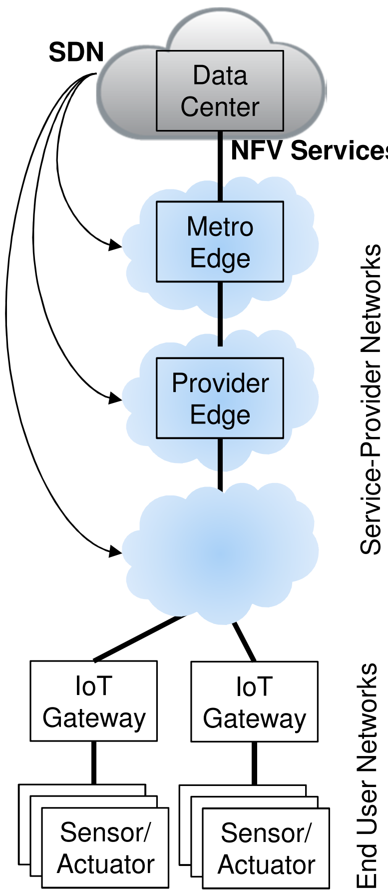
CONCLUSIONS

Conclusions

- I/O in virtualized environments is driven by new trends NFV, vCPE, vAccess
- Goal is to provide standard I/O support efficient and flexible
- KVM provides virtio, direct-assigned and pass-through devices
- Optimizing virtio, direct-assignment, pass-through is instrumental
- Supporting efficiently the Virtual Machine migration is a priority

NXP SOLUTIONS FOR VIRTUALIZATION

NFV/SDN Service Delivery - NXP Reference Designs



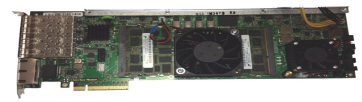
DC Solutions
- ❖ NFV Compute / Storage
- ❖ iNIC / SSL Accel
- ❖ C-RAN L1 Accel
- ❖ ADC / WoC
- ❖ ToR Router

Metro Edge Solutions
- ❖ Metro Routers
- ❖ L4-7 Appliances
- ❖ Content Delivery
- ❖ WAN Optimization

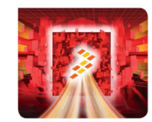
Metro Access Solutions
- ❖ Aggregation Routers
- ❖ Broadband Gateway
- ❖ Mobile BTS / C-RAN

CPE Access Solutions
- ❖ Campus Router
- ❖ Broadband Access
- ❖ Wireless / Mobile AP

IoT Gateway
- ❖ Building / Factory
- ❖ Smart Energy
- ❖ Transportation
- ❖ Digital Signage
- ❖ Medical / Fitness
- ❖ Remote Monitoring



NFV iNIC / Compute (T4/T2+C29x)
iNIC OVS offload with DPDK support
OpenStack / Open Daylight Framework
VortiQa v1.3 OF-Controller+L4-7



QorIQ Processors

SDN L2-7 Router + NFV (T2/LS2+C29x)
VortiQa OF-Agent+L4-7 Processing
Virtualized data plane (QorIQ & Switch)
NFV edge-based services

Mobile AP
(e.g. BSC913x)

IoT Gateway + NFV (LS1/T1)
WLAN AP
IoT AP
OpenWRT
Cloud mgmt
VortiQa OF-Switch+L4-7
NFV edge-based Services