What’s new in Virtio 1.1?

… and surrounding areas

Jens Freimann
Red Hat

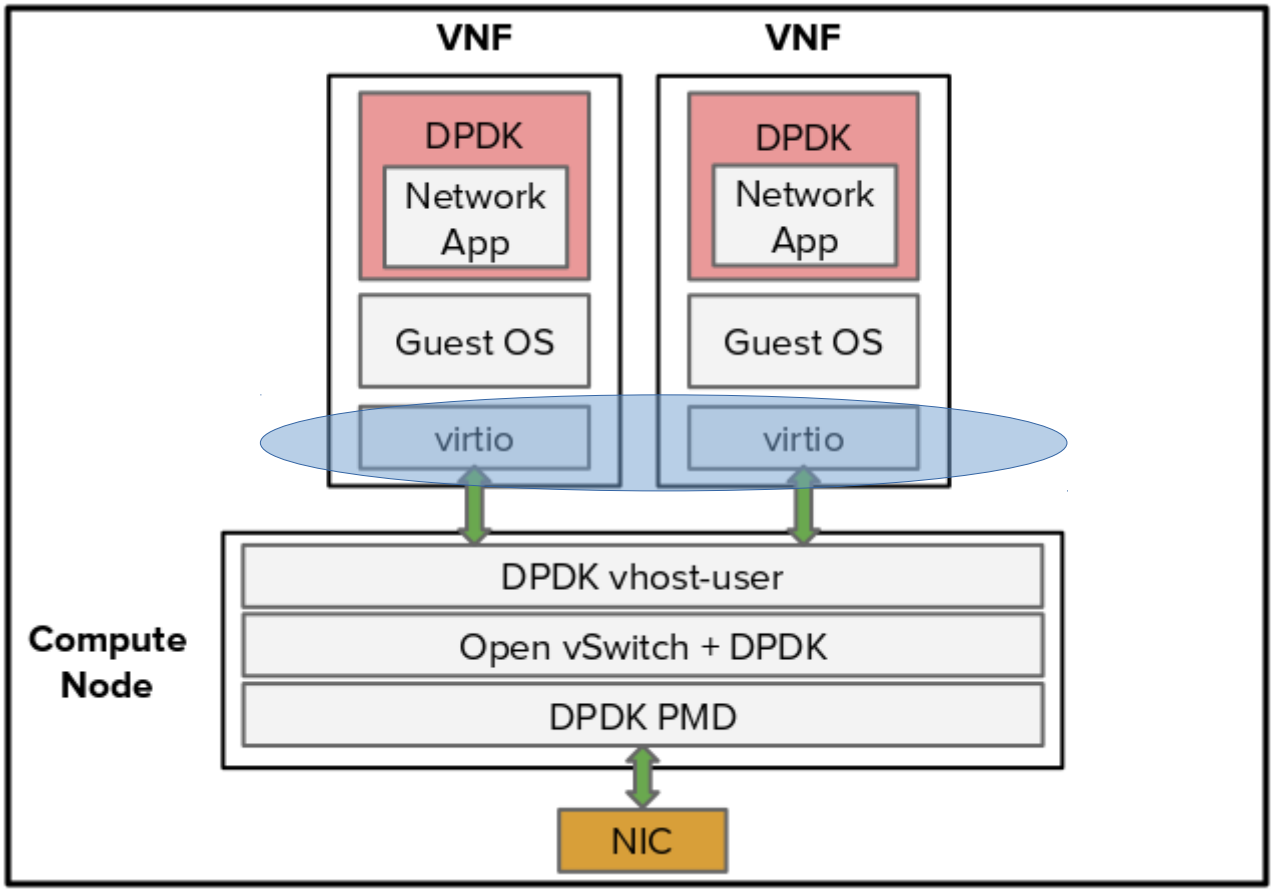
FOSDEM 2018

What is Virtio?

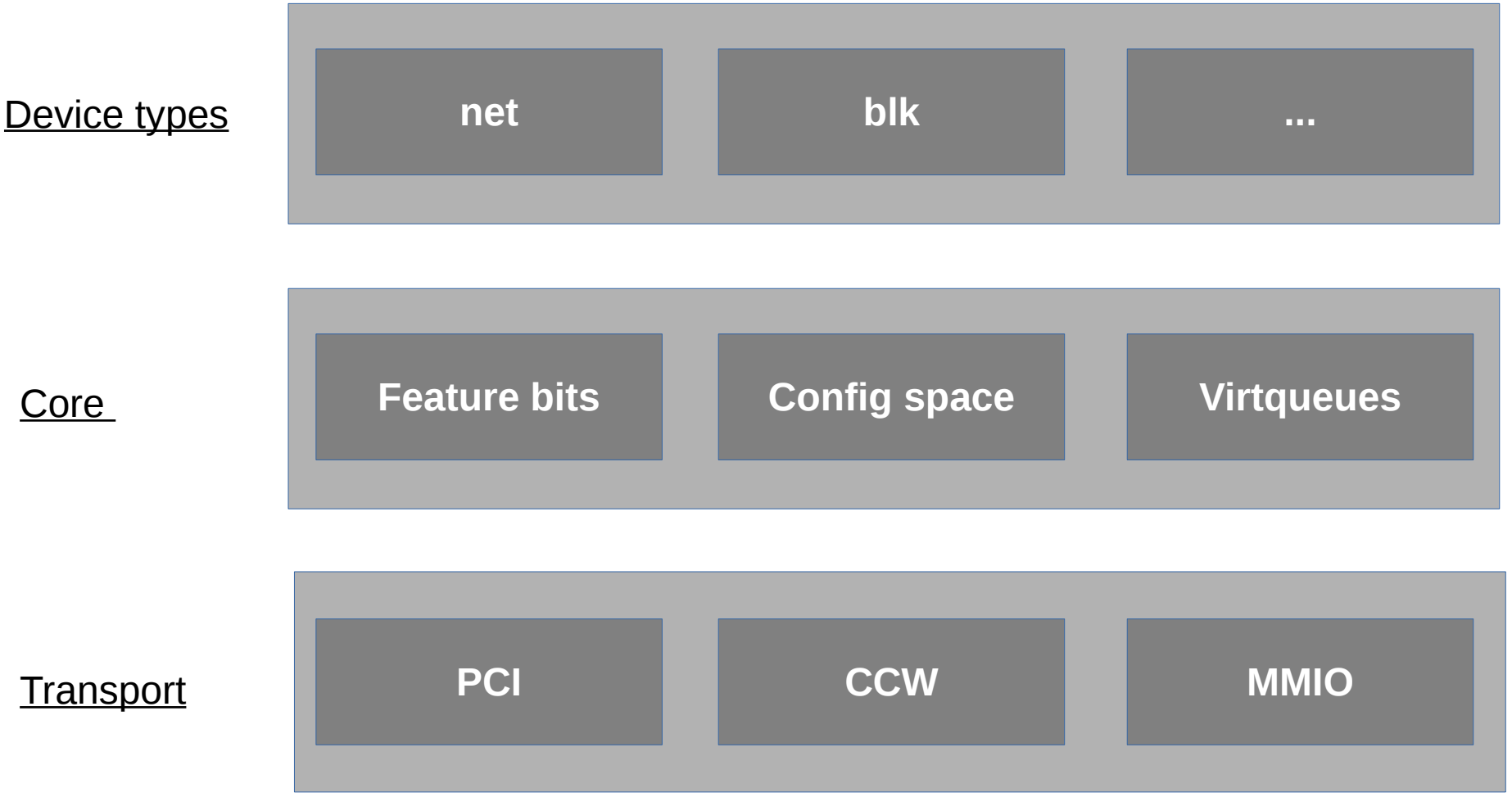
- Easy mechanism to provide virtual devices to guests
  - Disk, net, gpu, ...
- Standard driver means compatibility across hypervisors and operating systems

“The purpose of VIRTIO is to ensure that virtual environments and guests have a straightforward, efficient, standard, and extensible mechanism for virtual devices, rather than boutique per-environment or per-OS mechanisms.”

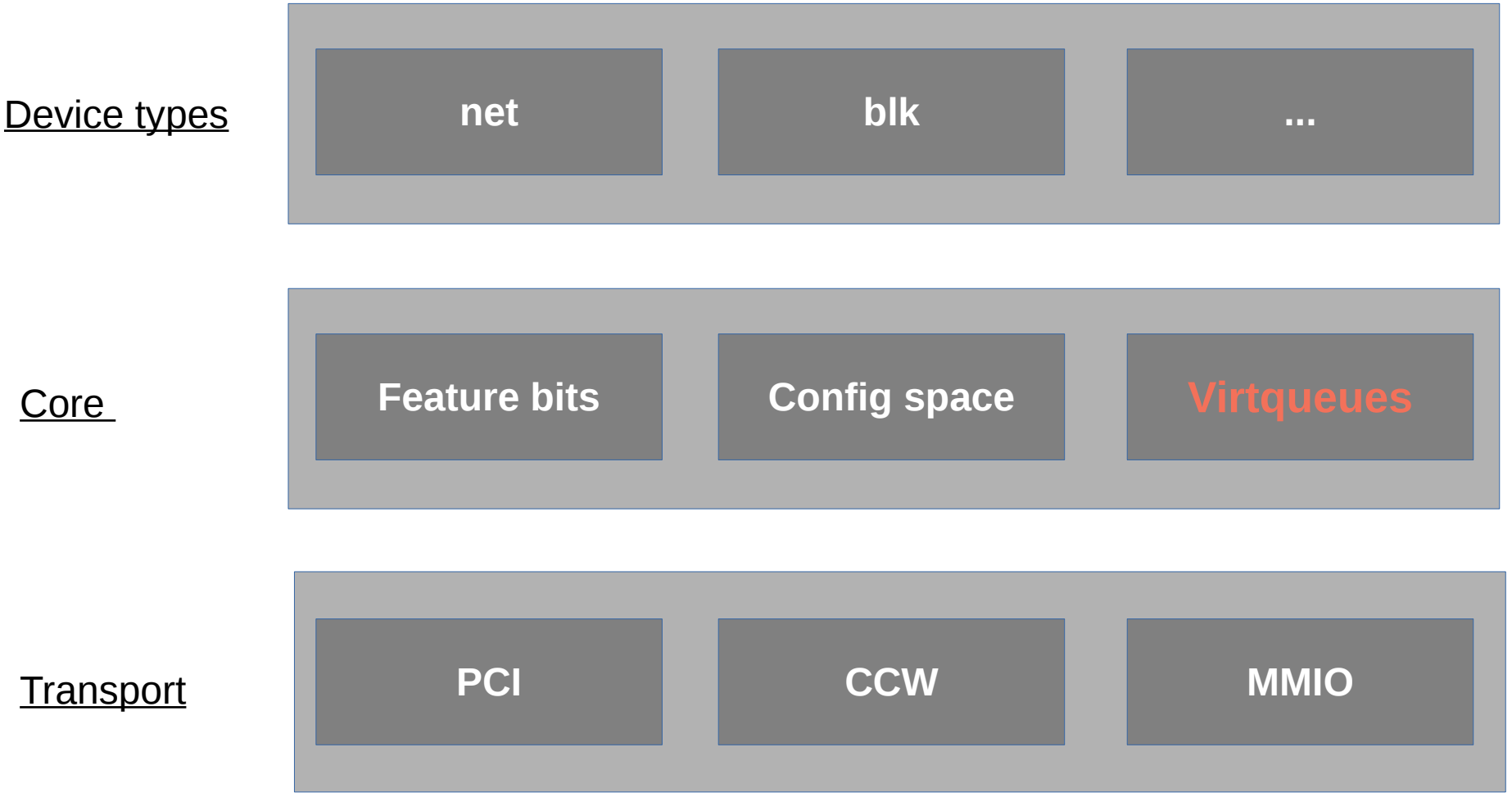
Virtio and NFV



Virtio components



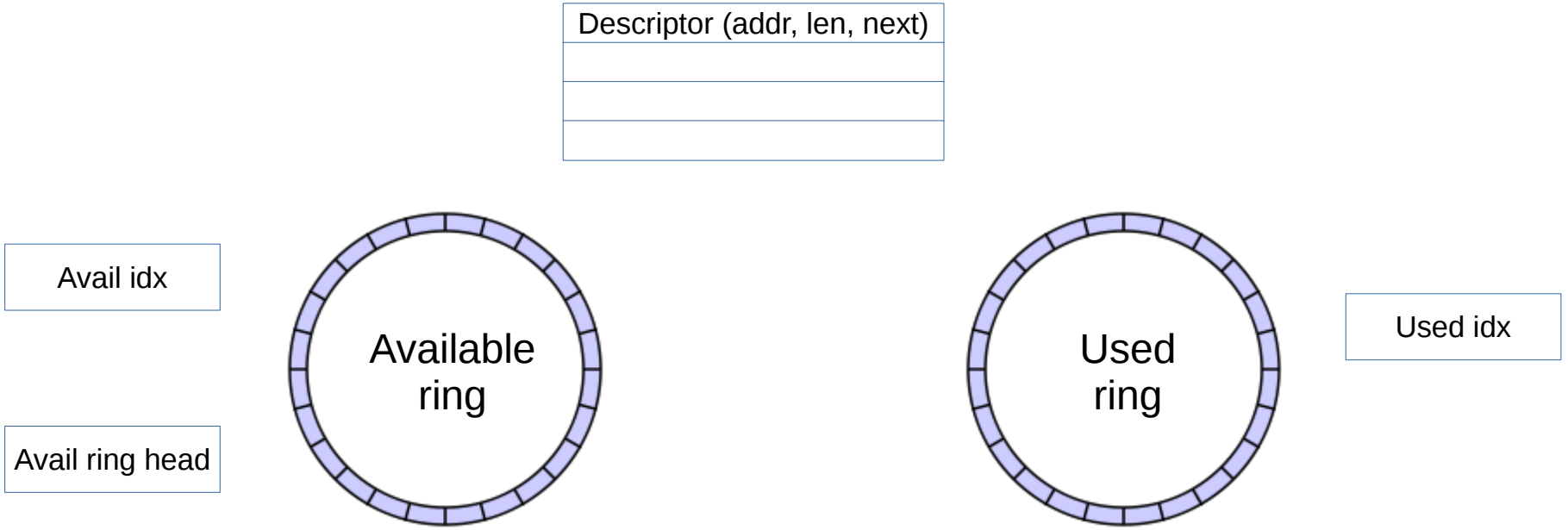
Virtio components



Packed Virtqueues

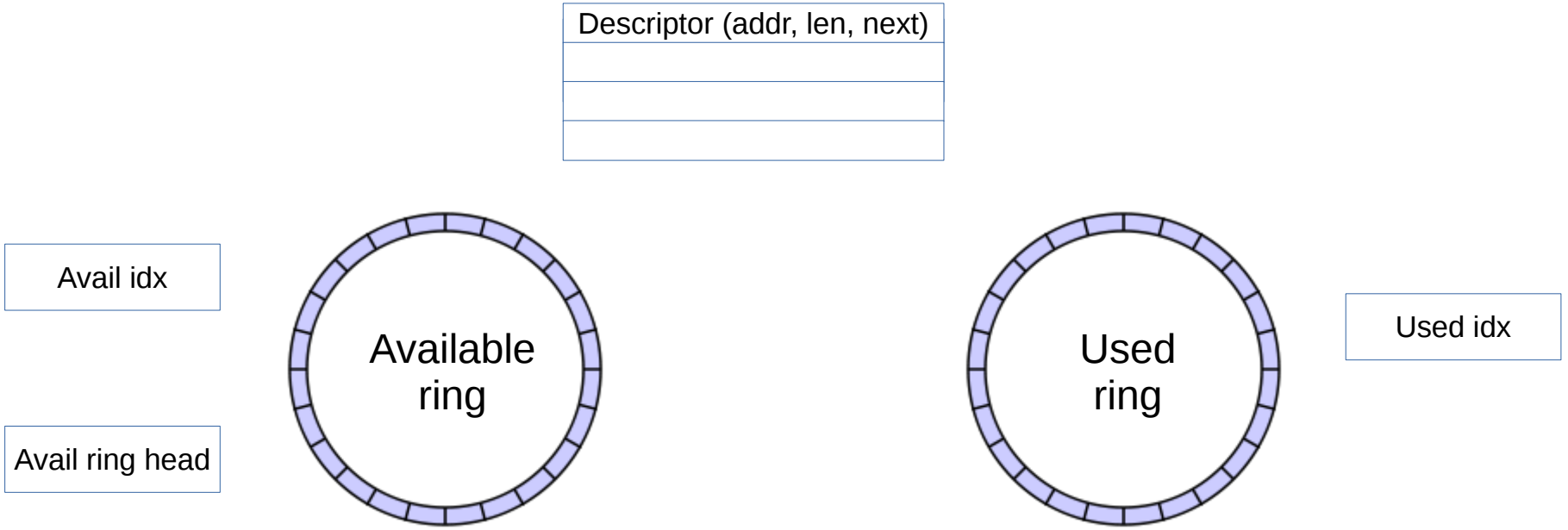
- Motivation
  - Make it easier to implement virtio in hardware
  - Performance gains
    - Optimize ring layout

Split virtqueue



- Located in shared memory
- Host and guest using shared memory to pass messages
- Possibly on different CPUs
- Causing cache synchronization

Split virtqueue



- Located in shared memory
- Host and guest using shared memory to pass messages
- Possibly on different CPUs
- Causing cache synchronization

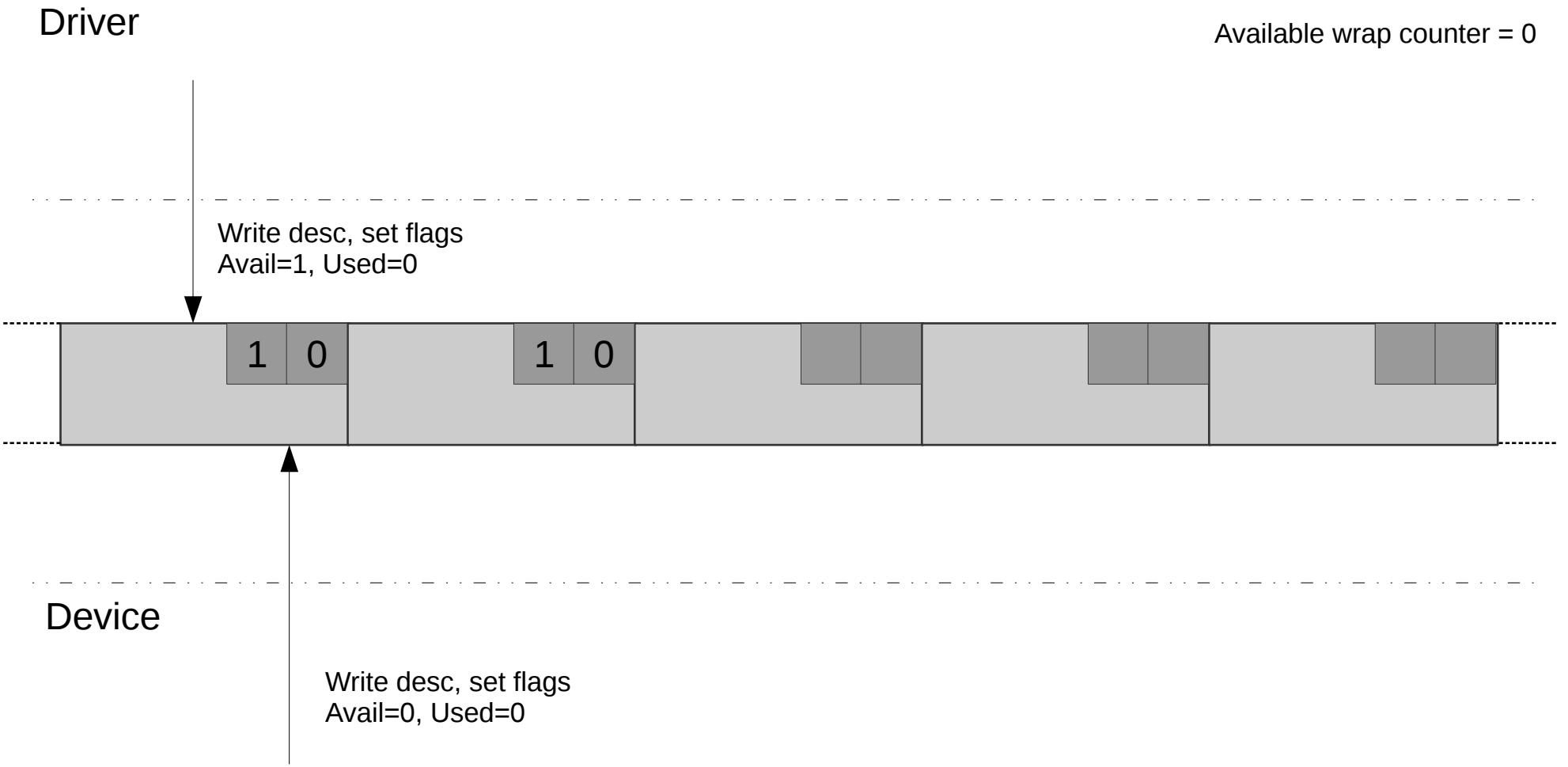
Reducing the overhead

- Information is spread across too many data structures
- Tighter packing will save cache misses.
- How about packing everything in a single data structure?

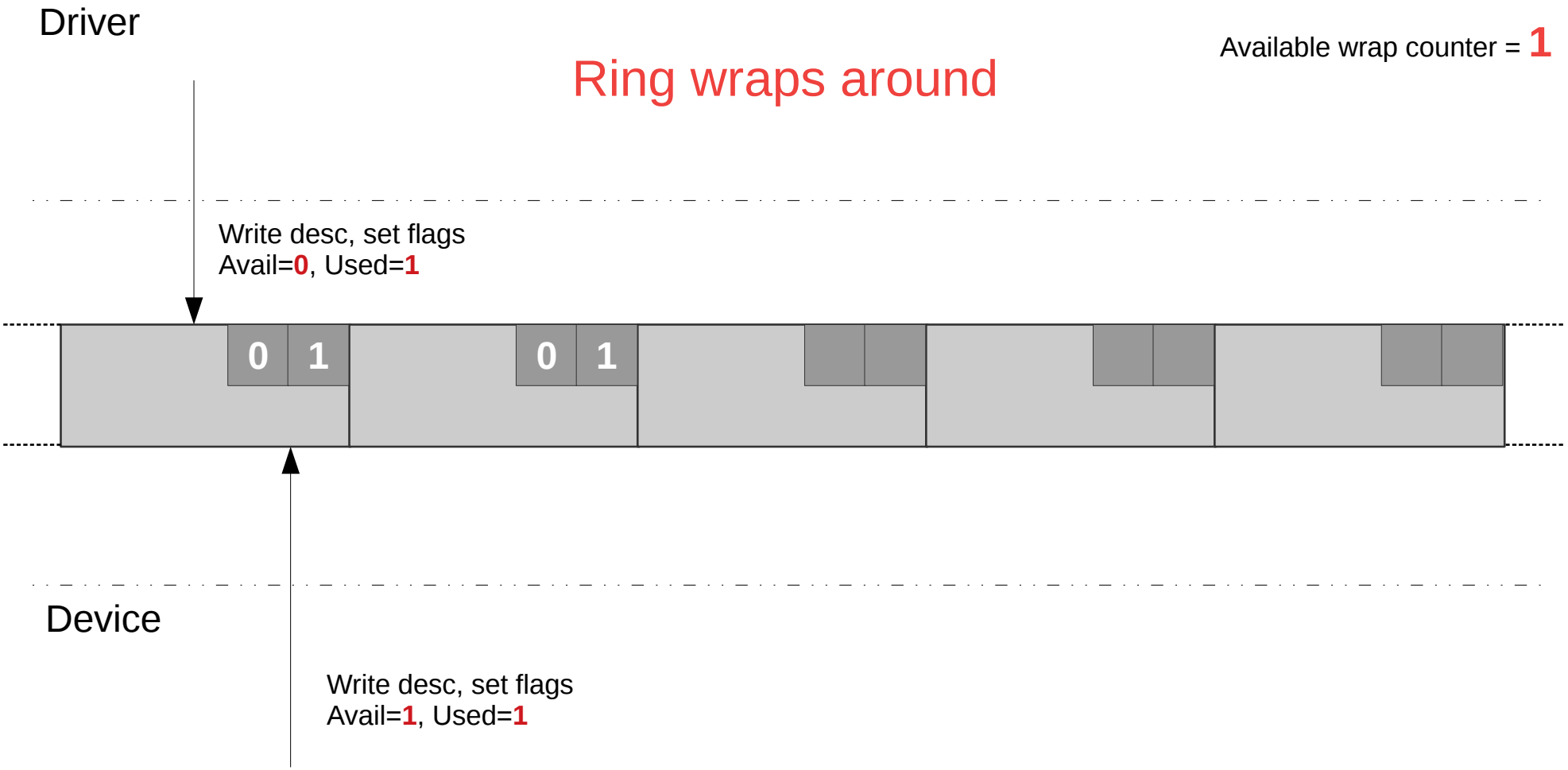
Descriptor ring

- Driver writes out available descriptors in a ring
- Device writes out used descriptors in the same ring
- Descriptor: addr, len, avail, used
- To mark a descriptor available, flip the avail bit
- To mark a descriptor as used, flip the used bit

Descriptor states



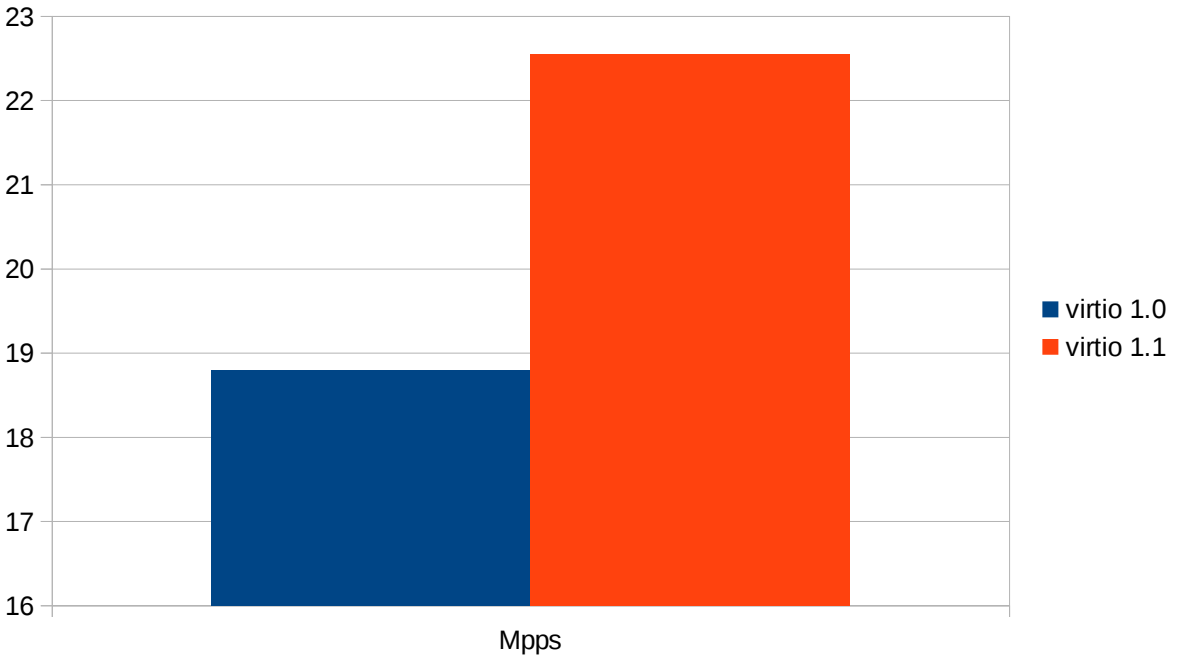
Descriptor states



Performance

- PVP test setup as described in http://www.dpdk.org/doc/guides/howto/pvp_reference_benchmark.html
- CPU: Intel(R) Xeon(R) CPU E5-2667 v4 @ 3.20GHz
- Host and Guest: RHEL 7.4
- Packet generator: T-REX
- Acceptable loss: 0.05%
- DPDK v17.11
- Patches at https://github.com/jensfr/dpdk/tree/virtio-1.1-v1

Performance – 64 byte packet throughput



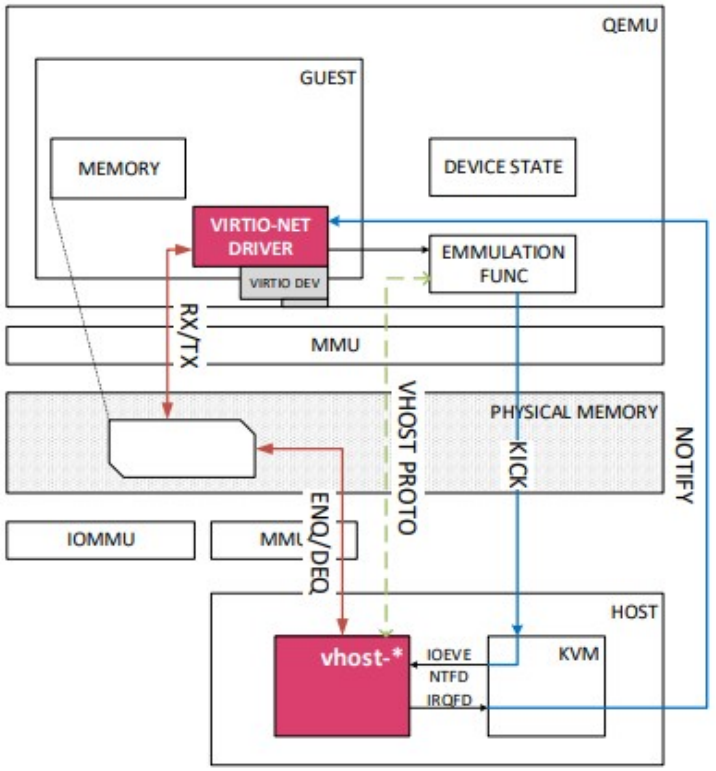
Participate in Virtio

- git clone https://github.com/oasis-tcs/virtio-spec Edit :)
- sh makeall.sh (needs xelatex, e.g. from texlive)
- virtio-comment-subscribe@lists.oasis-open.org
- Patch: virtio-comment@lists.oasis-open.org
- If no comments – email, ask for a vote ballot
- Total time: up to 2 weeks

Participate in Virtio

- git clone https://github.com/oasis-tcs/virtio-spec
  Edit :)
- sh makeall.sh (needs xelatex, e.g. from texlive)
- virtio-comment-subscribe@lists.oasis-open.org
- Patch: virtio-comment@lists.oasis-open.org
- If no comments – email, ask for a vote ballot
- Total time: up to 2 weeks

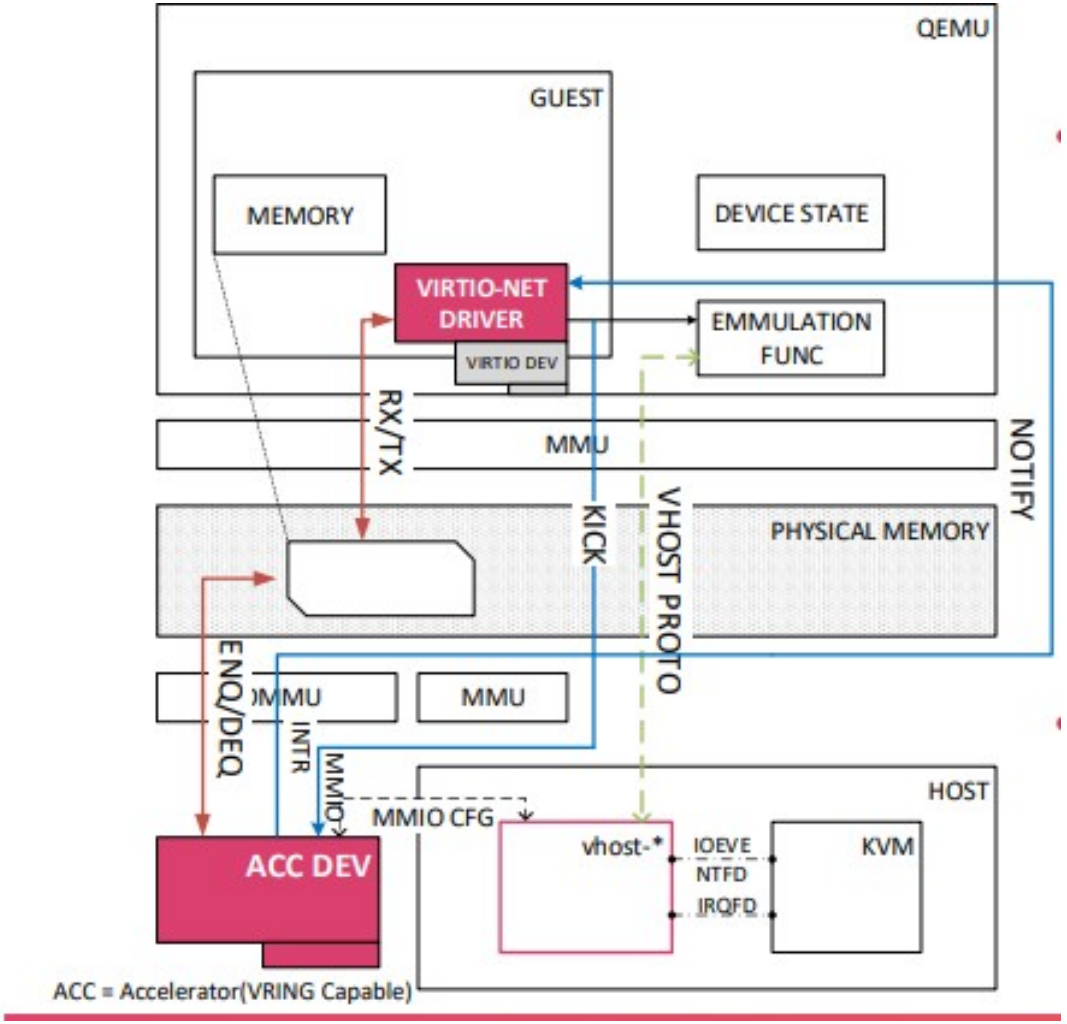
vDPA – virtual Datapath Accelerator



- PCI CSR Trapped
- Device-specific register trapped (PIO/MMIO)
- Emulation backed by backend adapter via VHOST PROTO
- Packet I/O via Shared memory
- Interrupt via IRQFD
- Doorbell via IOEVENTFD
- Diverse VHOST backend adaption

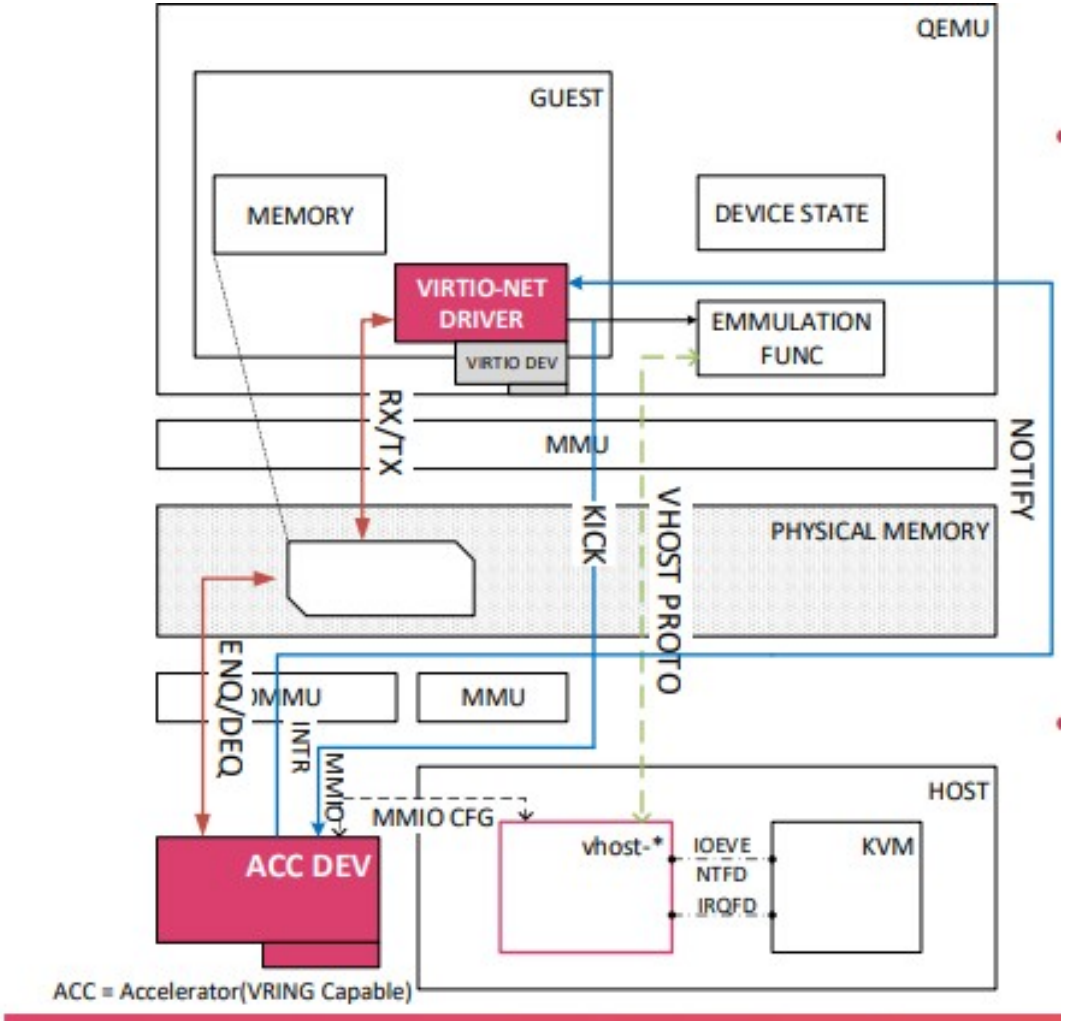
https://events.static.linuxfound.org/sites/events/files/slides/KVM17%27-vDPA.pdf

vDPA – virtual Datapath Accelerator



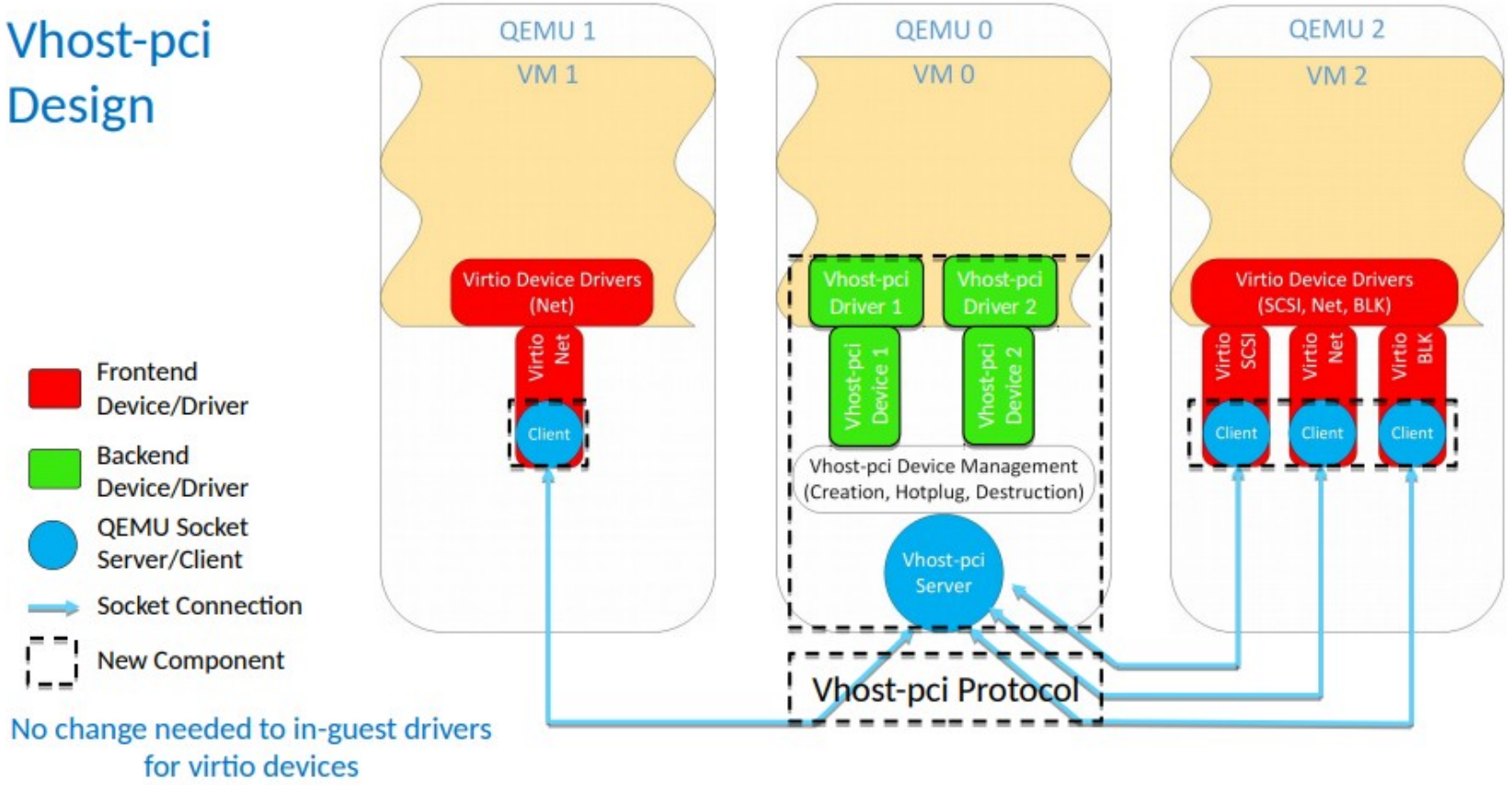
https://events.static.linuxfound.org/sites/events/files/slides/KVM17%27-vDPA.pdf

vDPA – virtual Datapath Accelerator



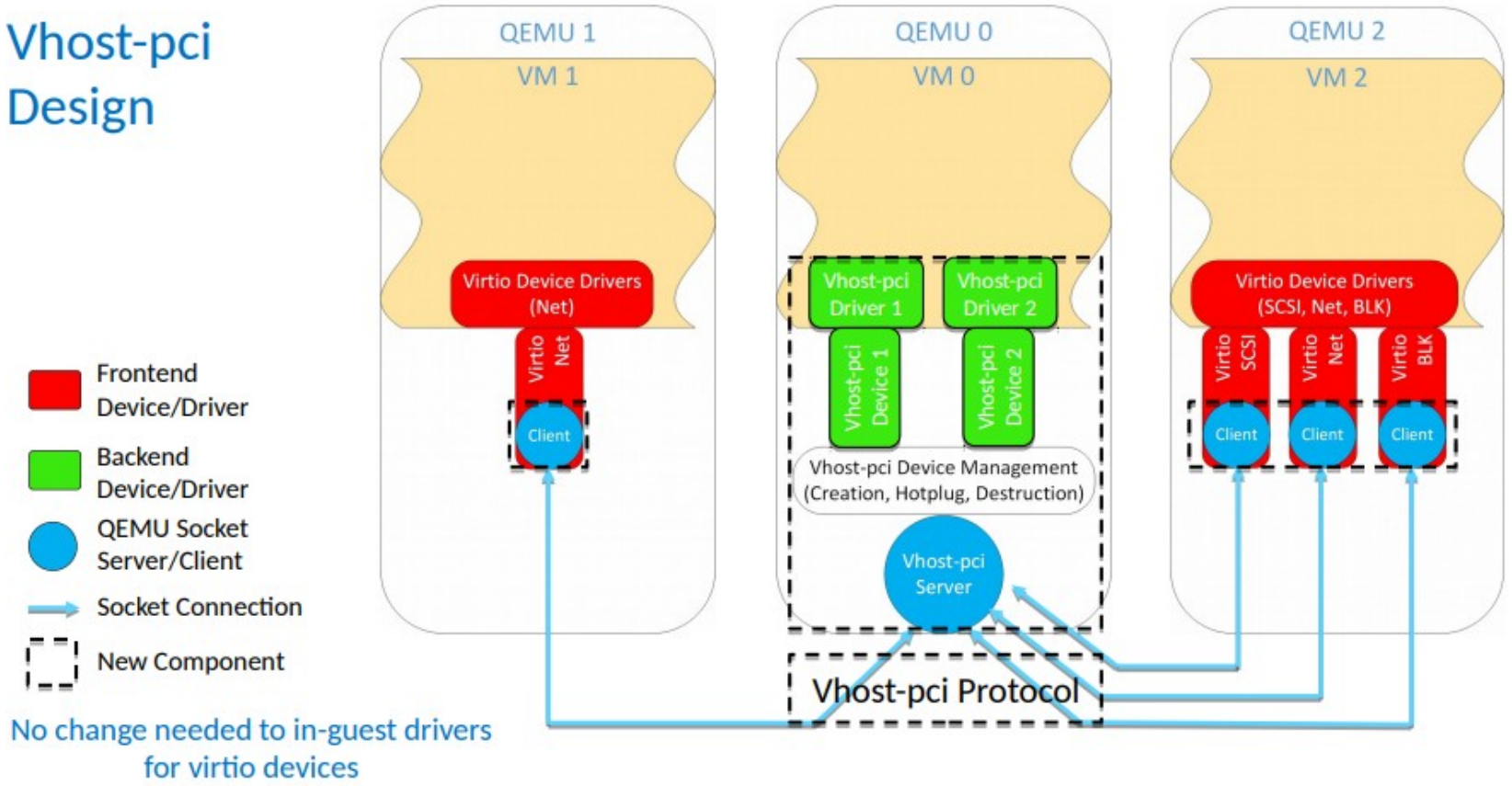
https://events.static.linuxfound.org/sites/events/files/slides/KVM17%27-vDPA.pdf

Vhost-pci – Fast VM-to-VM communication



http://www.linux-kvm.org/images/5/55/02x07A-Wei_Wang-Design_of-Vhost-pci.pdf

Vhost-pci – Fast VM-to-VM communication



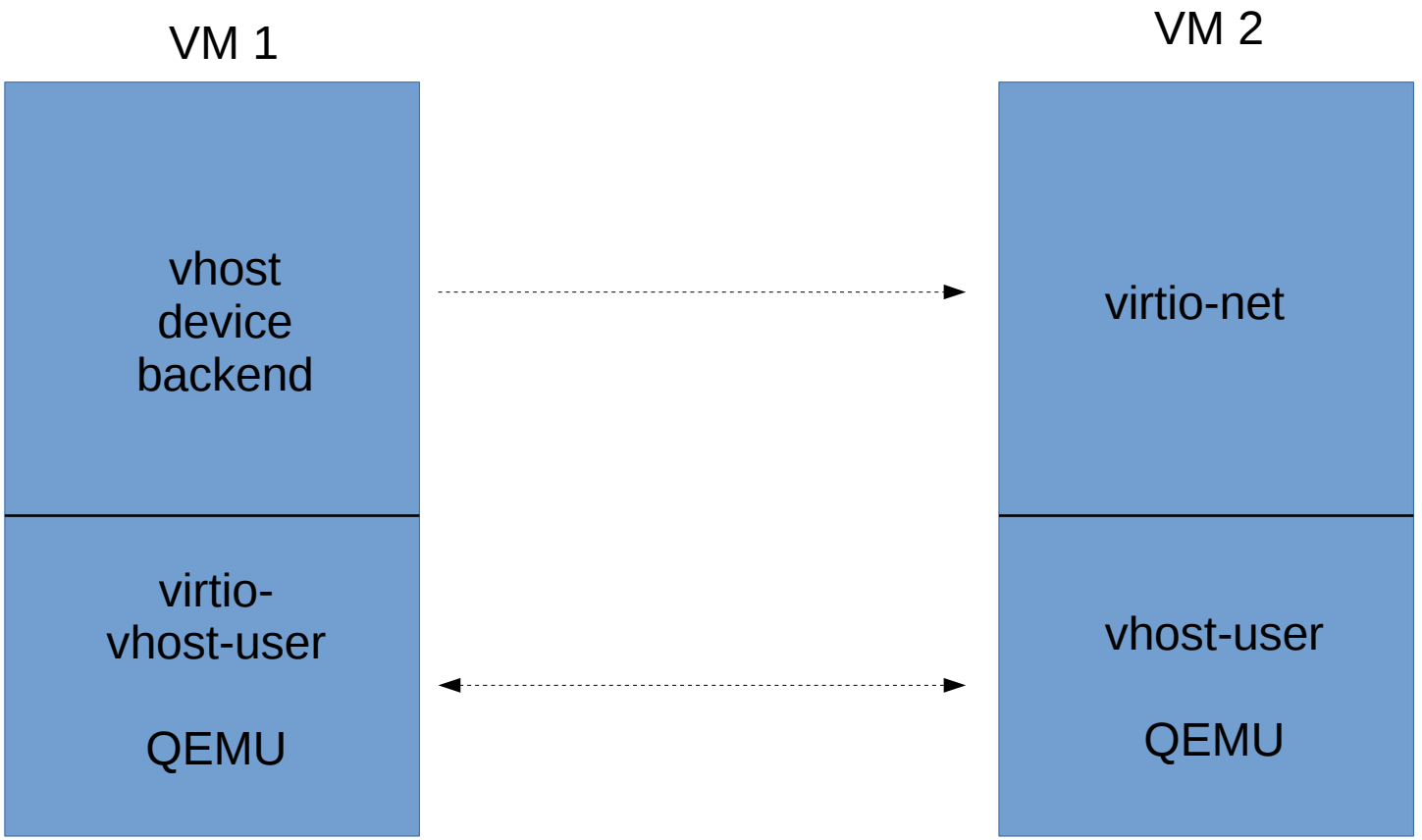
http://www.linux-kvm.org/images/5/55/02x07A-Wei_Wang-Design_of-Vhost-pci.pdf

virtio-vhost-user

- Slightly different approach to vhost-pci but same goal
- Lets guests act as vhost device backends
  - Virtual network appliances can provide virtio devices to other guests
  - Provide high-performance vhost-user appliances to other guests in the same cloud environment
- Exitless fast VM-to-VM communication
  - With poll mode drivers, even with interrupts fast because of ioeventfds

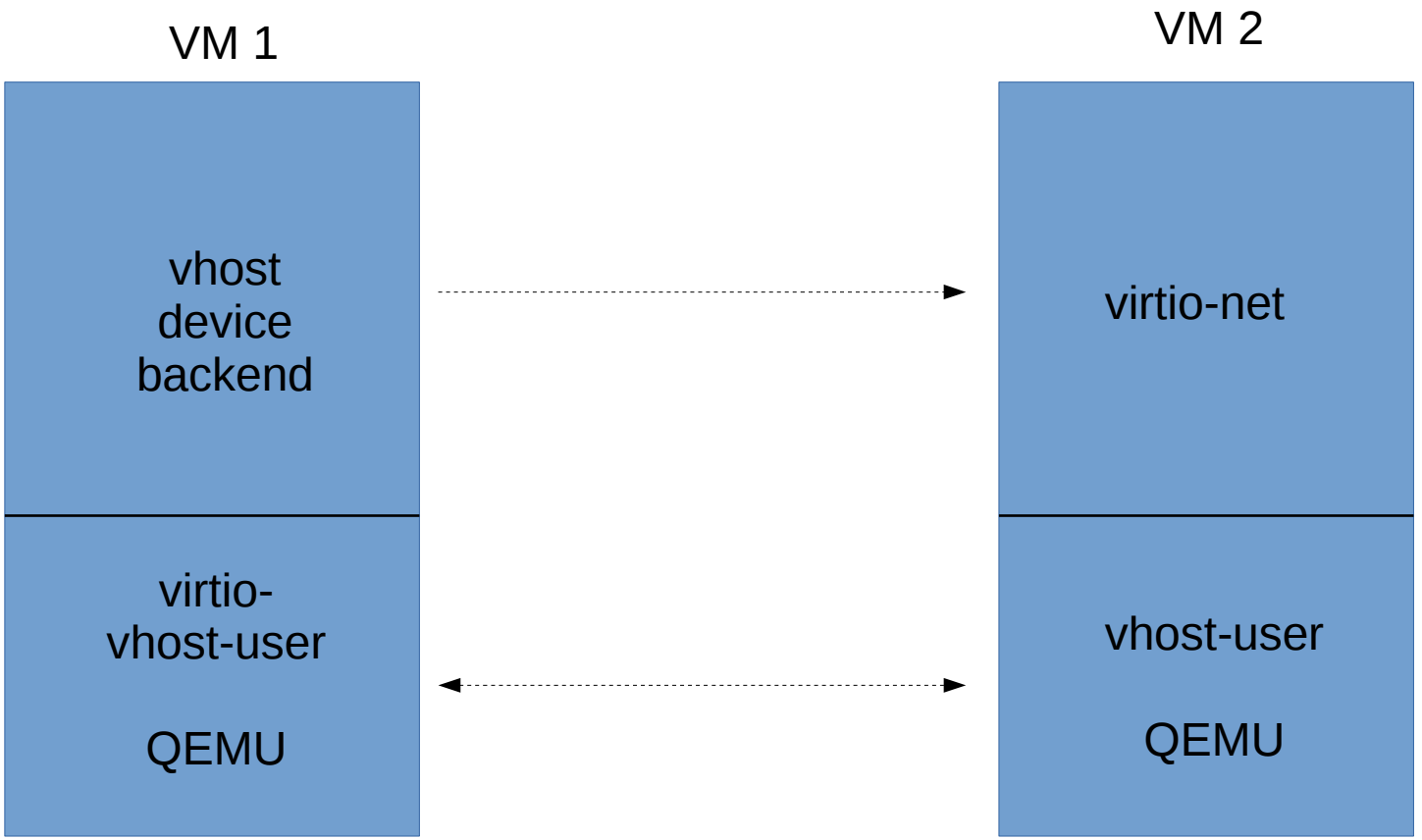
https://wiki.qemu.org/Features/VirtioVhostUser

virtio-vhost-user



https://wiki.qemu.org/Features/VirtioVhostUser

virtio-vhost-user



https://wiki.qemu.org/Features/VirtioVhostUser

Transparent bonding for SR-IOV devices

- Bond a virtio device and a SR-IOV device
  - When virtio and SR-IOV drivers load, look for other NIC that matches MAC address
  - Enslave NIC and bring up the slave
- Switch over to virtio device before migration
  - Link of SR-IOV device goes down
- On target system switch back to SR-IOV device (if available, virtio as fallback)

Conclusion

- Virtio 1.1 will be a big release, focus on
  - Performance
  - Hardware implementation

  - DPDK implemenation of packed virtqueues:

    <insert link>
  - There is a monthly DPDK Virtio meeting where we discuss progress, let me know if you'd like to join.

    jfreimann@redhat.com

Questions?

jfreimann@redhat.com