Qualcomm

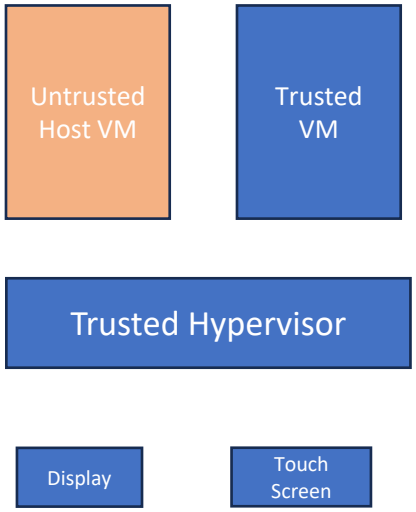
# Device Assignment


Srivatsa Vaddagiri

Principal Engineer, Qualcomm India Private Limited


@qualcomm

# Device Assignment Scenario …




- **Requirements**
  - Runtime ***temporary*** assignment of platform devices between VMs.
    - VM owns a device for part of its lifetime
  - Latency of ownership transfer should be “low”
- **Downstream approach**
  - Host
    - Transfer ownership of IO regions and IRQs (hypervisor-specific API)
    - Reject subsequent userspace requests to use device
    - [device power is retained across ownership transfer]
  - Guest
    - (At bootup) Driver probe skips touching device registers
    - Accept ownership of IO regions/IRQS and “validate”
    - Start using device
- **Upstream approach (VFIO)**
  - Host
    - Unbind from native driver
    - Bind to VFIO (platform) driver
    - Assign device to target VM(M)
  - Guest
    - Bind to native driver
- **Potential Issues with upstream approach**
  - Latency
  - User experience – panel switches off and on during ownership transfer
  - Stacked drivers, with which applications have active connections
  - Validation of device credentials

# Device Assignment ..

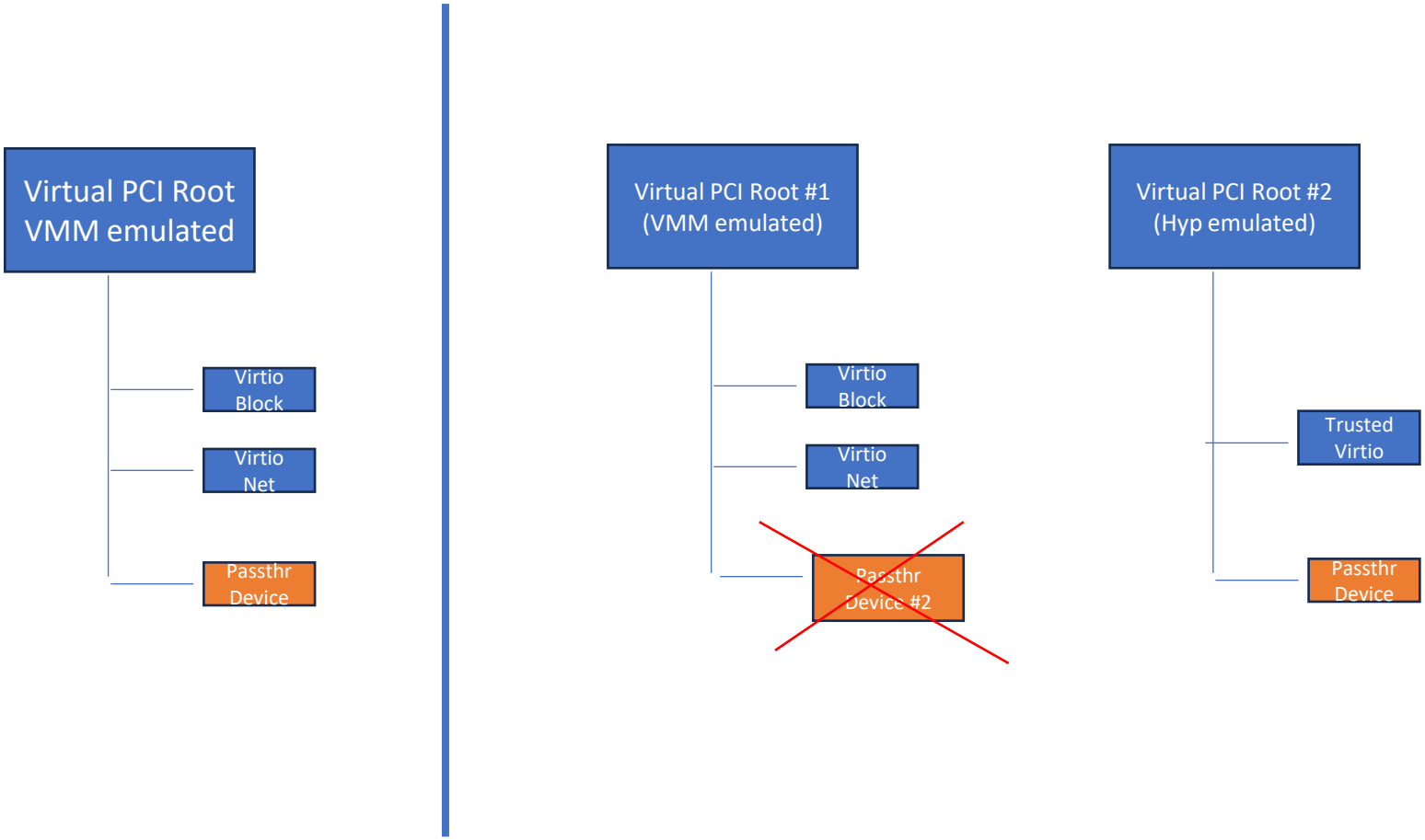
- **Device Attestation**
  - How can VM be assured it got the right device (esp platform devices)?
  - Downstream mechanism depends on guest knowing platform IO space (via signed device tree), identity mapping of IPA->PA and verifying it got the right IPA
    - Extend TDISP for platform devices? David Hartley of Qualcomm is presenting on this proposal later this week
- **PCI device assignment (without hardware TDISP support)**
  - Downstream implementation may rely on hypervisor emulated PCI root complex
- **Device sanitization upon VM (abrupt/normal) termination**
  - pKVM relies on vendor drivers at EL2 to handle sanitization, which may bloat EL2
  - Delegate sanitization to a “service” VM?
- **Power management**
  - Device and its allied controls (GPIO, Clock, Regulator) as one unit of transfer
  - Runtime power management under guest control
- **Handling IOMMU topology changes at runtime**
  - For PCI devices, BDF->Stream ID mapping can be dynamic
  - Dynamic IOMMU groups?
- **Unit of assignment**
  - Ensuring device and its “siblings” (sharing the same IOMMU for ex) are assigned as one unit

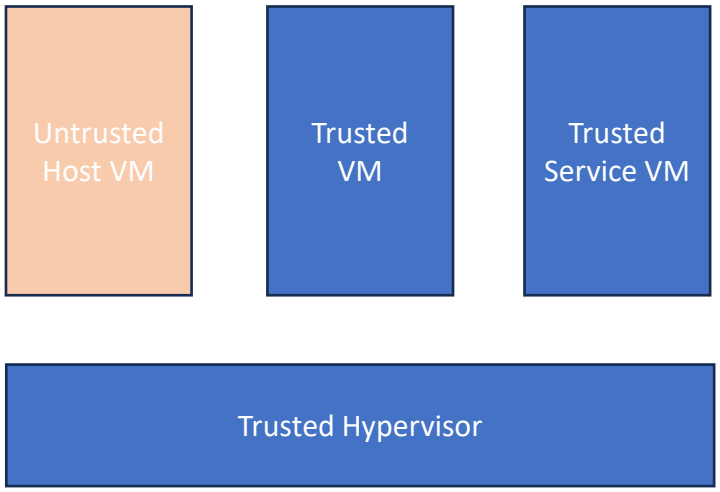
# Thank you



Follow us on:

For more information, visit us at qualcomm.com & qualcomm.com/blog


Virtual PCI Root
VMM emulated
Virtio
Block
Virtio
Net
Passthr
Device
Virtual PCI Root #1
(VMM emulated)
Virtio
Block
Virtio
Net
Passthr
Device #2
Virtual PCI Root #2
(Hyp emulated)
Trusted
Virtio
Passthr
Device




- **Requirements**
  - Sanitize state of any device assigned to Trusted VM when it crashes
- **Considerations**
  - VM memory could be small – OR perhaps VMM is no longer “alive” - makes it difficult to use kexec
  - Device could be in arbitrary power-saving state. Sanitizing may require that we determine/change the current power state (is the clock turned on?) before we attempt to sanitize
  - Reset power state (remove clock votes?) before relinquishing ownership
  - Device could be another DSP, which would need SoC specific mechanism to communicate with DSP
  - Trusted Service VM will need to know, with help of hypervisor, what devices were assigned to a given trusted VM