# PCI Endpoint Subsystem Open Items Discussion

## Linux Plumbers Conference

Vienna, Austria, 2024

Manivannan Sadhasivam
Linaro

Arm Solutions at Lightspeed

# About me

- Senior Linux Kernel Engineer - Qualcomm Landing Team of Linaro
- Open source contributor since 2016
- Maintainer of the PCI Endpoint subsystem, MHI bus and some ARM SoCs
- Reviewer of PCI Controller Drivers
- Working from Erode, Tamilnadu, India

# linaro™ is the software engine of the arm Ecosystem

**Linaro empowers rapid product deployment within the dynamic arm Ecosystem.**

- Our cutting-edge solutions, services and collaborative platforms facilitate the swift **development, testing, and delivery of arm-based innovations,** enabling businesses to stay ahead in today’s competitive technology landscape.

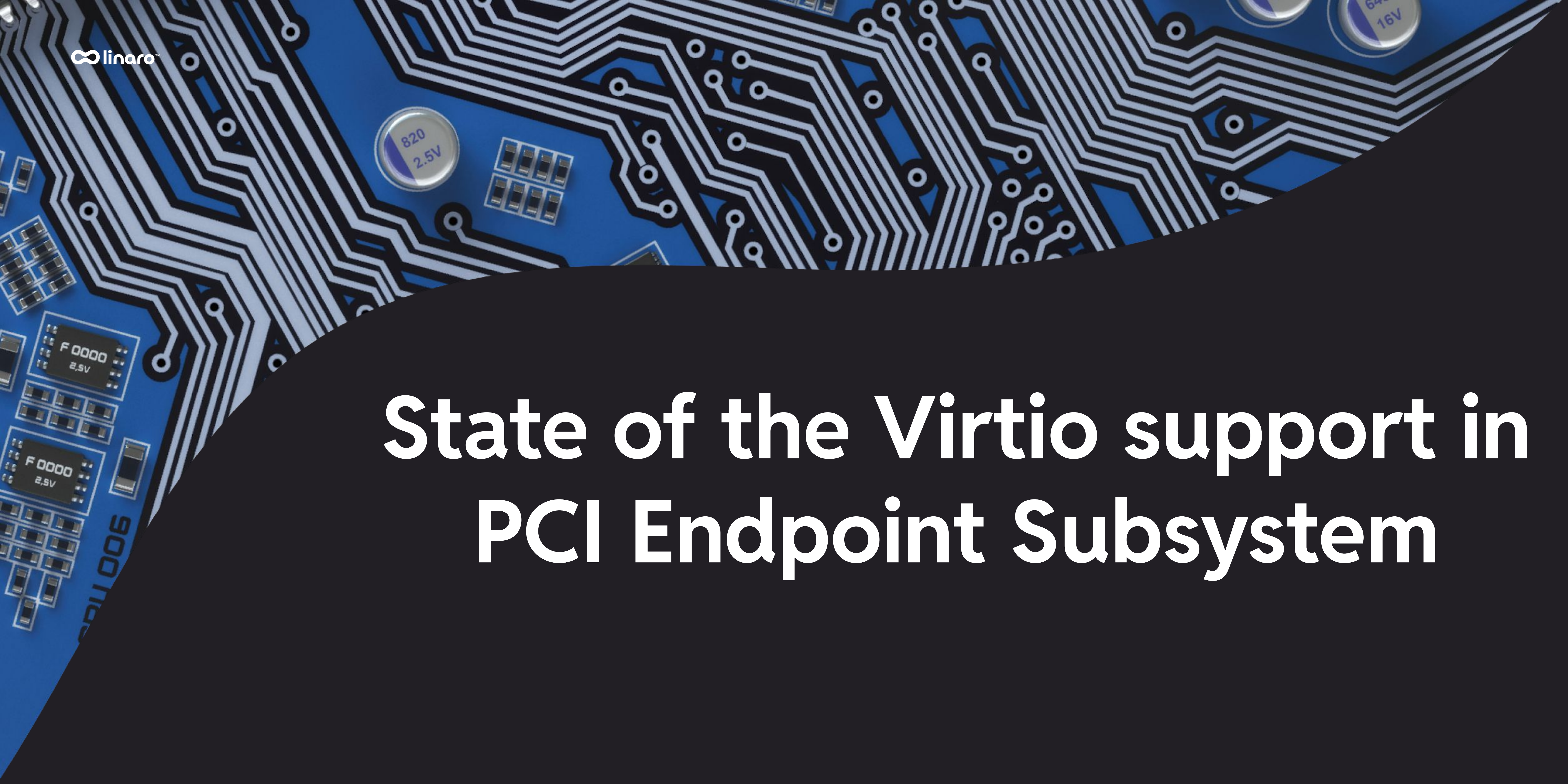
- **Linaro** fosters an environment of collaboration, standardization and optimization among businesses and **open source ecosystems to accelerate the deployment of arm-based products and technologies** along with representing a pivotal role in open source discovery and adoption.

- **Automotive, Testing, Linux Kernel, Security, Cloud & Edge Computing, IoT & Embedded, AI, CI/CD, Toolchain, Virtualization**

## Linaro has enabled trust, quality and collaboration since 2010

# Agenda

- State of the Virtio support in PCI Endpoint Subsystem
- Using QEMU for testing PCI Endpoint Subsystem
- Repurposing Interrupt Controllers for Doorbells in Endpoint Devices



# State of the Virtio support in PCI Endpoint Subsystem

# Virtio: Recap

# Virtio: Recap

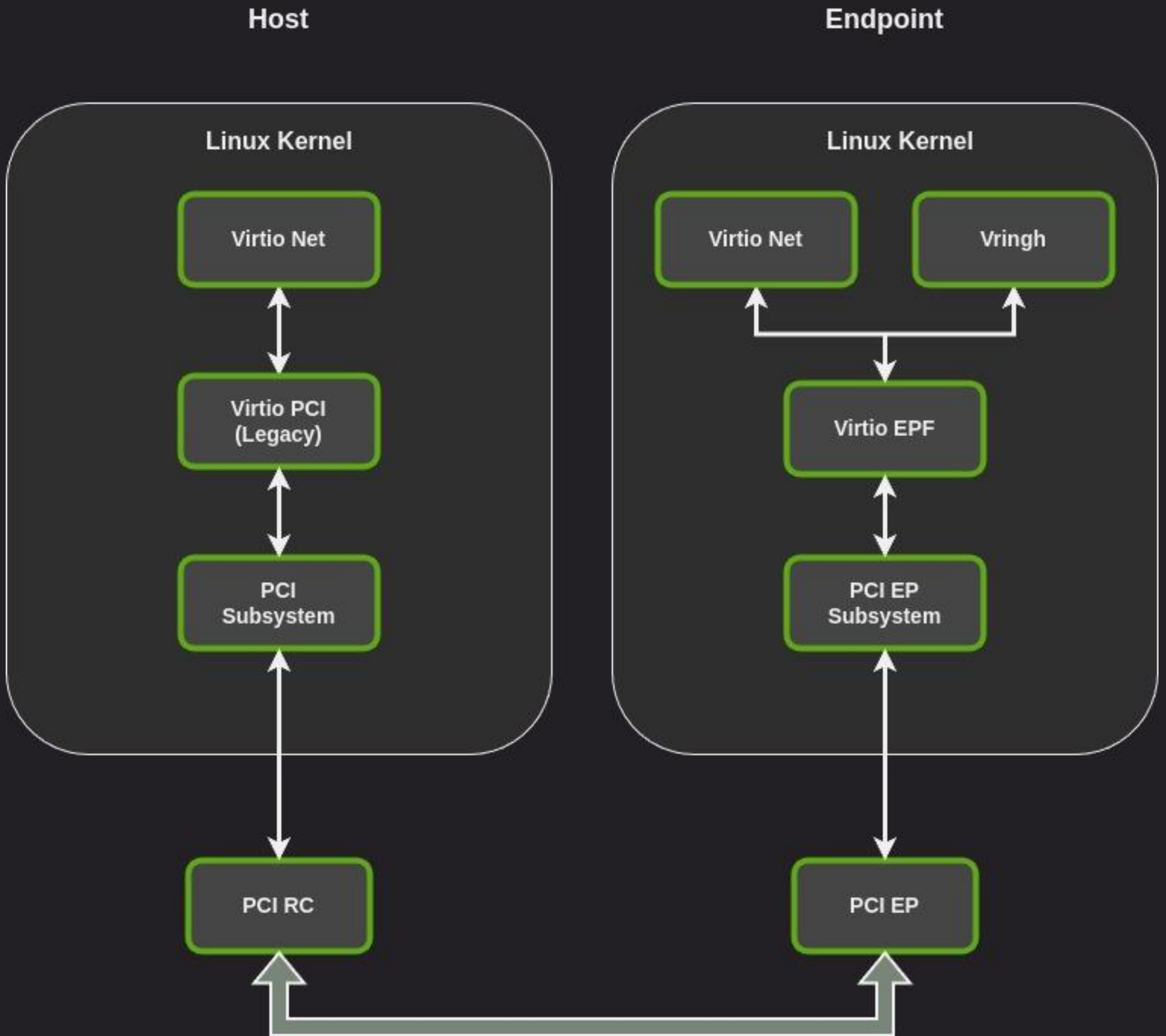
- Presented 3 proposals last year

linaro™

# Virtio: Recap

- Presented 3 proposals last year
- Got consensus for proposal from Shunsuke Mie

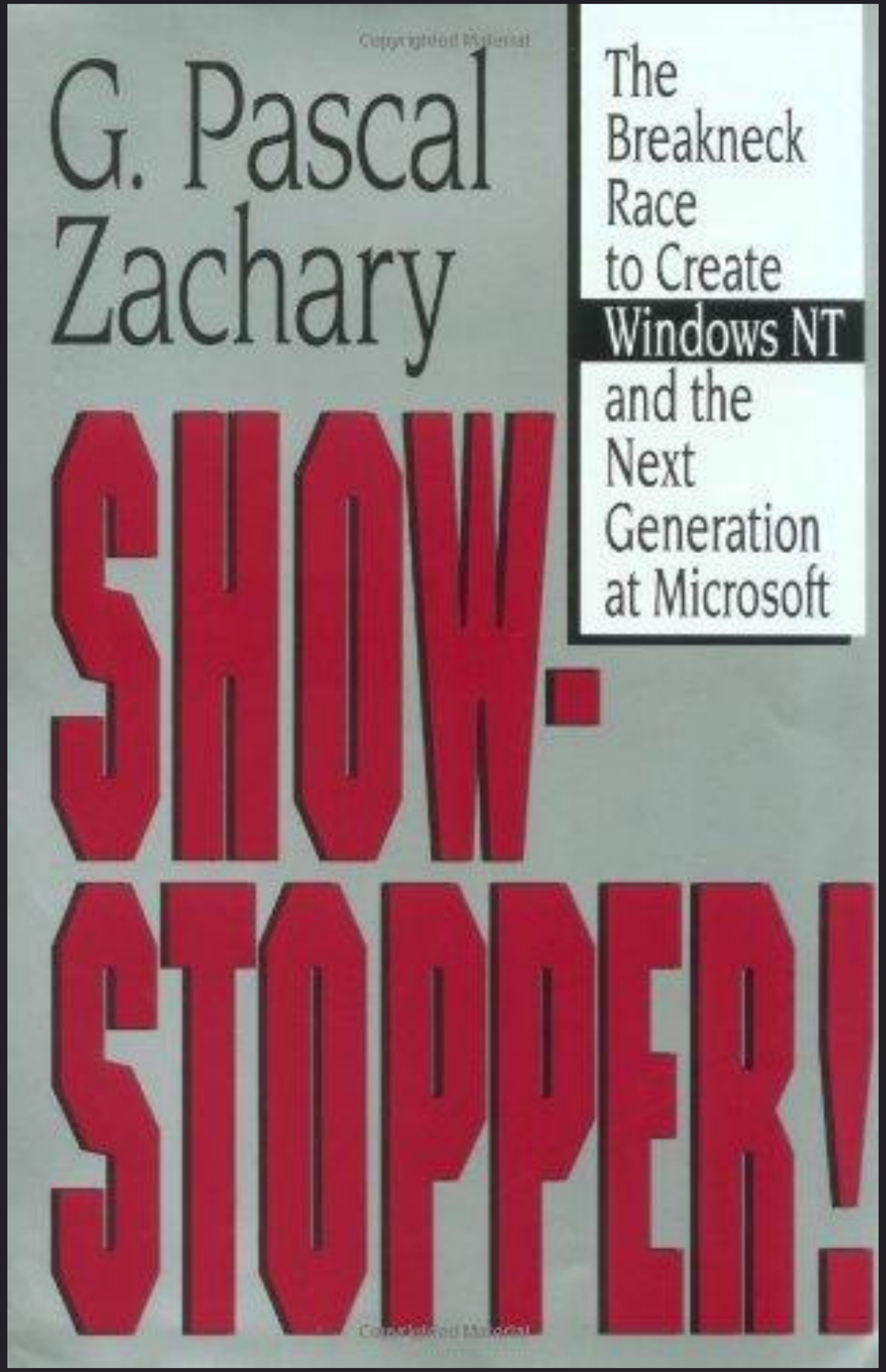
# Virtio: Recap



# Showstopper!!!

# Showstopper!!!

Not this one



G. Pascal Zachary

SHOW-STOPPER!

The Breakneck Race to Create Windows NT and the Next Generation at Microsoft

# Showstopper!!!

- Implementation issues in the agreed proposal

# Showstopper!!!

- Implementation issues in the agreed proposal
  - No MSI/MSI-X

# Showstopper!!!

- Implementation issues in the agreed proposal
  - No MSI/MSI-X
  - Exposed legacy Virtio device only (No modern Virtio)

# Showstopper!!!

- Implementation issues in the agreed proposal
  - No MSI/MSI-X
  - Exposed legacy Virtio device only (No modern Virtio)
  - No IOMMU support

# Showstopper!!!

- Implementation issues in the agreed proposal
  - No MSI/MSI-X
  - Exposed legacy Virtio device only (No modern Virtio)
  - No IOMMU support
  - Race between Virtio device and driver

# MSI/MSI-X

# MSI/MSI-X

- Virtio spec supports both INTx and MSI-X

# MSI/MSI-X

- Virtio spec supports both INTx and MSI-X
- But no MSI

# MSI/MSI-X

- Virtio spec supports both INTx and MSI-X
- But no MSI
- Submitted a proposal for adding MSI to Virtio spec
  - https://lore.kernel.org/virtio-comment/20240712140144.12066-1-manivannan.sadhasivam@linaro.org

# MSI/MSI-X

- Virtio spec supports both INTx and MSI-X
- But no MSI
- Submitted a proposal for adding MSI to Virtio spec
  - https://lore.kernel.org/virtio-comment/20240712140144.12066-1-manivannan.sadhasivam@linaro.org
- And a patch to Linux kernel
  - https://lore.kernel.org/virtualization/20240712142914.16979-1-manivannan.sadhasivam@linaro.org/

# Modern Virtio device

# Modern Virtio device

- Requires configurable PCIe Vendor Capability

# Modern Virtio device

- Requires configurable PCIe Vendor Capability
    - Used to discover the location of Virtio structures

# Modern Virtio device

- Requires configurable PCIe Vendor Capability
  - Used to discover the location of Virtio structures
- PCIe Vendor Capability not configurable on generic endpoint devices

# Modern Virtio device

- Requires configurable PCIe Vendor Capability
  - Used to discover the location of Virtio structures
- PCIe Vendor Capability not configurable on generic endpoint devices
- Working on a proposal to discover Virtio structures without Vendor Capability

# Modern Virtio device

- Requires configurable PCIe Vendor Capability
  - Used to discover the location of Virtio structures
- PCIe Vendor Capability not configurable on generic endpoint devices
- Working on a proposal to discover Virtio structures without Vendor Capability
  - Fixed offset/BAR location for Virtio structures?

# IOMMU

# IOMMU

- Most of the modern host platforms have IOMMU for PCI

# IOMMU

- Most of the modern host platforms have IOMMU for PCI
- IOMMU (translated address) is only supported on modern Virtio spec

# IOMMU

- Most of the modern host platforms have IOMMU for PCI
- IOMMU (translated address) is only supported on modern Virtio spec
- Should be addressed after migrating to modern Virtio spec

# Race between Virtio device and driver

# Race between Virtio device and driver

- Absence of sync point between Virtio device and driver causes race condition

# Race between Virtio device and driver

- Absence of sync point between Virtio device and driver causes race condition
- Issue applicable only to physical endpoint devices

# Race between Virtio device and driver

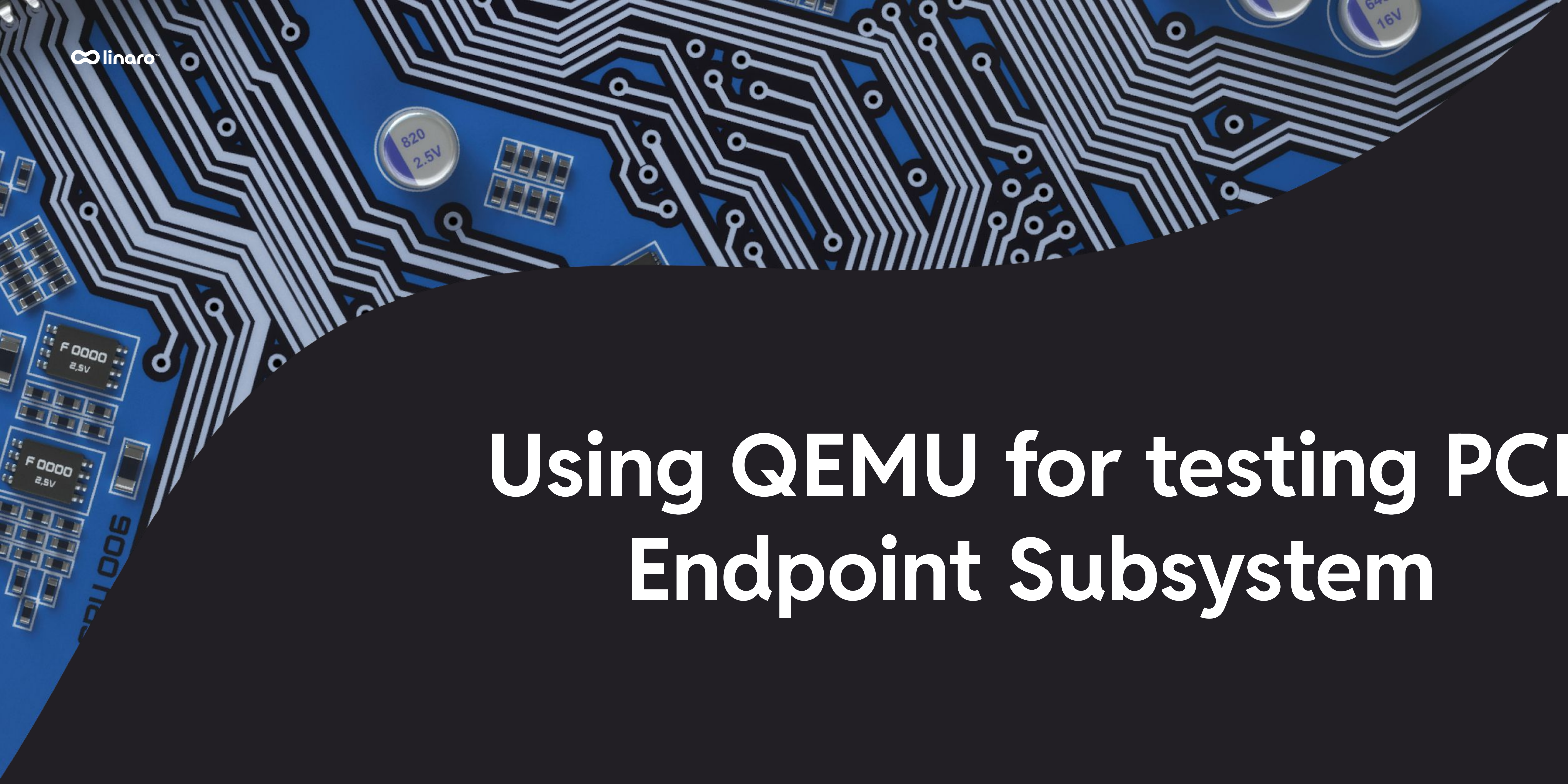
- Absence of sync point between Virtio device and driver causes race condition
- Issue applicable only to physical endpoint devices
  - Not on QEMU as the endpoint accesses are trapped by the hypervisor

# Race between Virtio device and driver

- Absence of sync point between Virtio device and driver causes race condition
- Issue applicable only to physical endpoint devices
  - Not on QEMU as the endpoint accesses are trapped
- Trapping endpoint access is not possible in a physical endpoint device

# Race between Virtio device and driver

- Absence of sync point between Virtio device and driver causes race condition
- Issue applicable only to physical endpoint devices
  - Not on QEMU as the endpoint accesses are trapped
- Trapping endpoint access is not possible in a physical endpoint device
- Requires a spec change adding sync point between device and driver



# Using QEMU for testing PCI Endpoint Subsystem

# Problem Statement

# Problem Statement

- Testing PCI Endpoint Subsystem requires host and a configurable endpoint

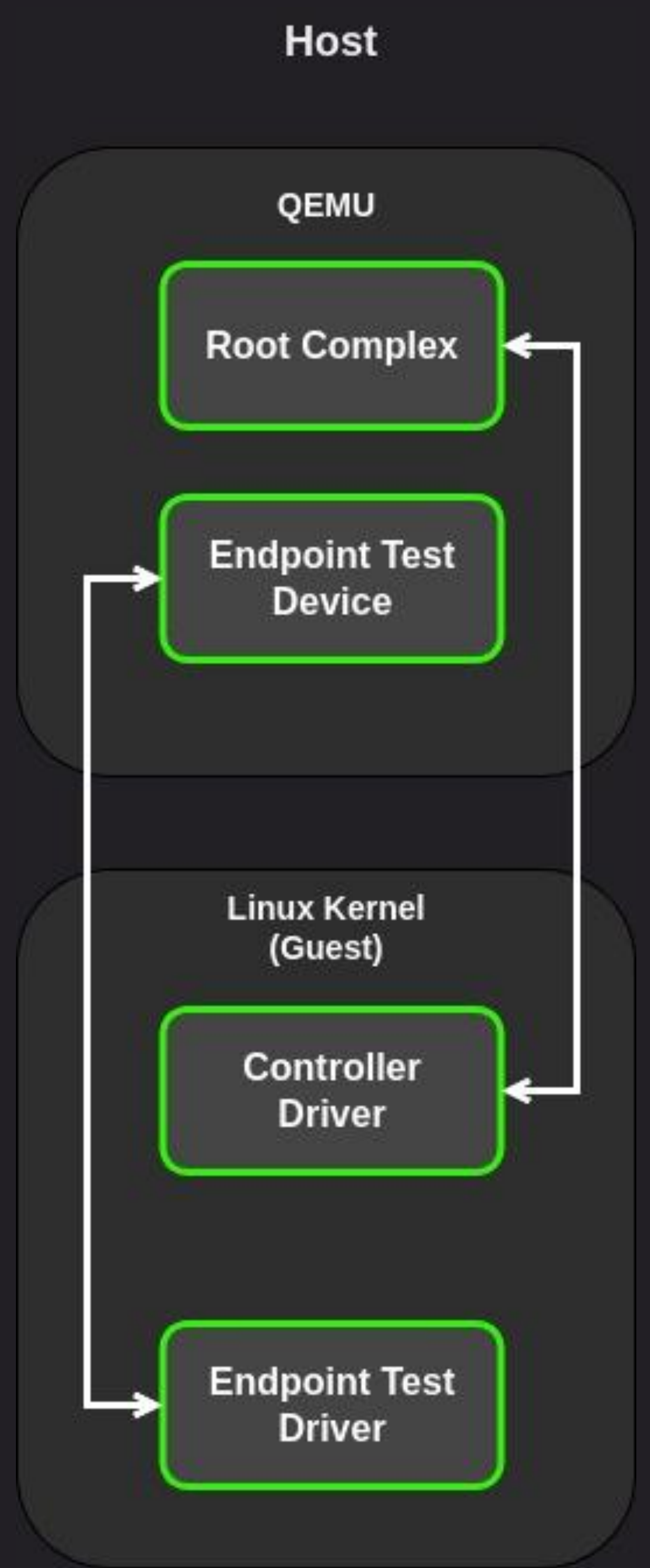
# Problem Statement

- Testing PCI Endpoint Subsystem requires host and a configurable endpoint
- Need a way to test without hardware!

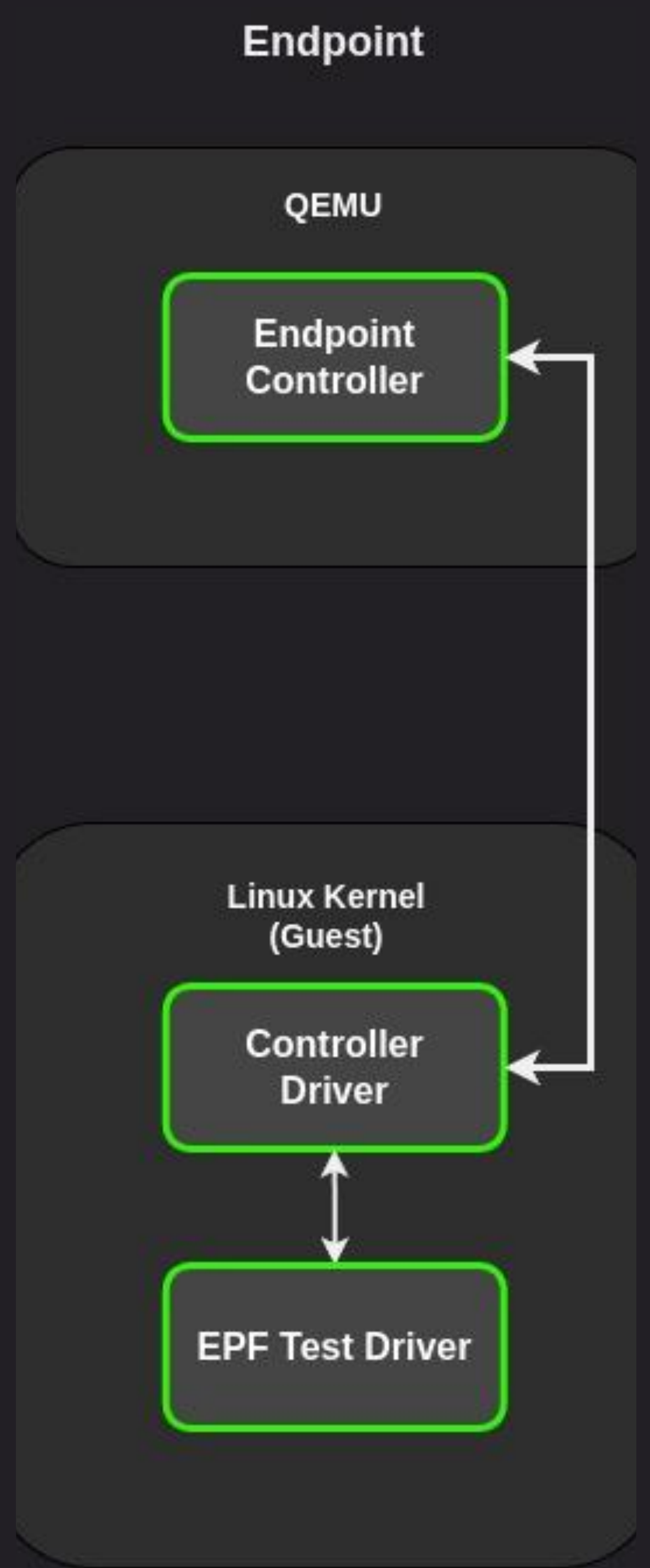
# Problem Statement

- Testing PCI Endpoint Subsystem requires host and a configurable endpoint
- Need a way to test without hardware!
  - QEMU seems to be the natural choice

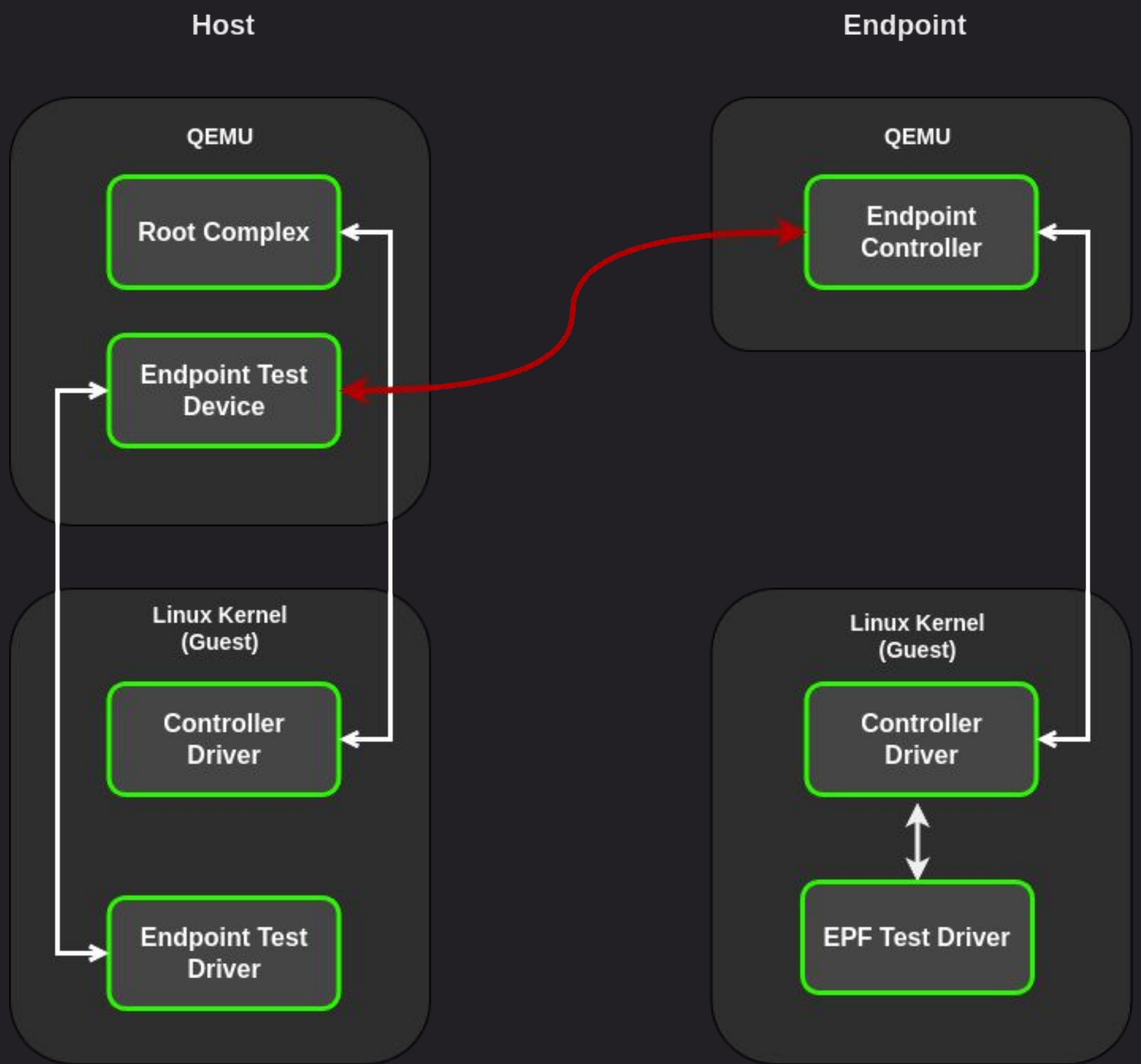
# QEMU for PCI Host



# QEMU for PCI Endpoint



# QEMU End to End



# Proposal

# Proposal

- Proposal from Shunsuke Mie
  - https://lore.kernel.org/qemu-devel/CANXvt5oKt=AKdqv24LT079e+6URnfqJcfTJh0ajGA17paJUEKw@mail.gmail.com/

# Proposal

- Proposal from Shunsuke Mie
  - https://lore.kernel.org/qemu-devel/CANXvt5oKt=AKdqv24LT079e+6URnfqJcfTJh0ajGA17paJUEKw@mail.gmail.com/
- Requires 2 Guests on the same host

# Proposal

- Proposal from Shunsuke Mie
  - https://lore.kernel.org/qemu-devel/CANXvt5oKt=AKdqv24LT079e+6URnfqJcfTJh0ajGA17paJUEKw@mail.gmail.com/
- Requires 2 Guests on the same host
  - Communication through UNIX domain socket

# Proposal

- Proposal from Shunsuke Mie
  - https://lore.kernel.org/qemu-devel/CANXvt5oKt=AKdqv24LT079e+6URnfqJcfTJh0ajGA17paJUEKw@mail.gmail.com/
- Requires 2 Guests on the same host
  - Communication through UNIX domain socket
- Endpoint
  - PCI Endpoint Controller implemented as a QEMU PCI device

# Proposal

- Proposal from Shunsuke Mie
  - https://lore.kernel.org/qemu-devel/CANXvt5oKt=AKdqv24LT079e+6URnfqJcfTJh0ajGA17paJUEKw@mail.gmail.com/
- Requires 2 Guests on the same host
  - Communication through UNIX domain socket
- Endpoint
  - PCI Endpoint Controller implemented as a QEMU PCI device
  - Requires a new controller driver in Linux Kernel (Guest)

# Proposal

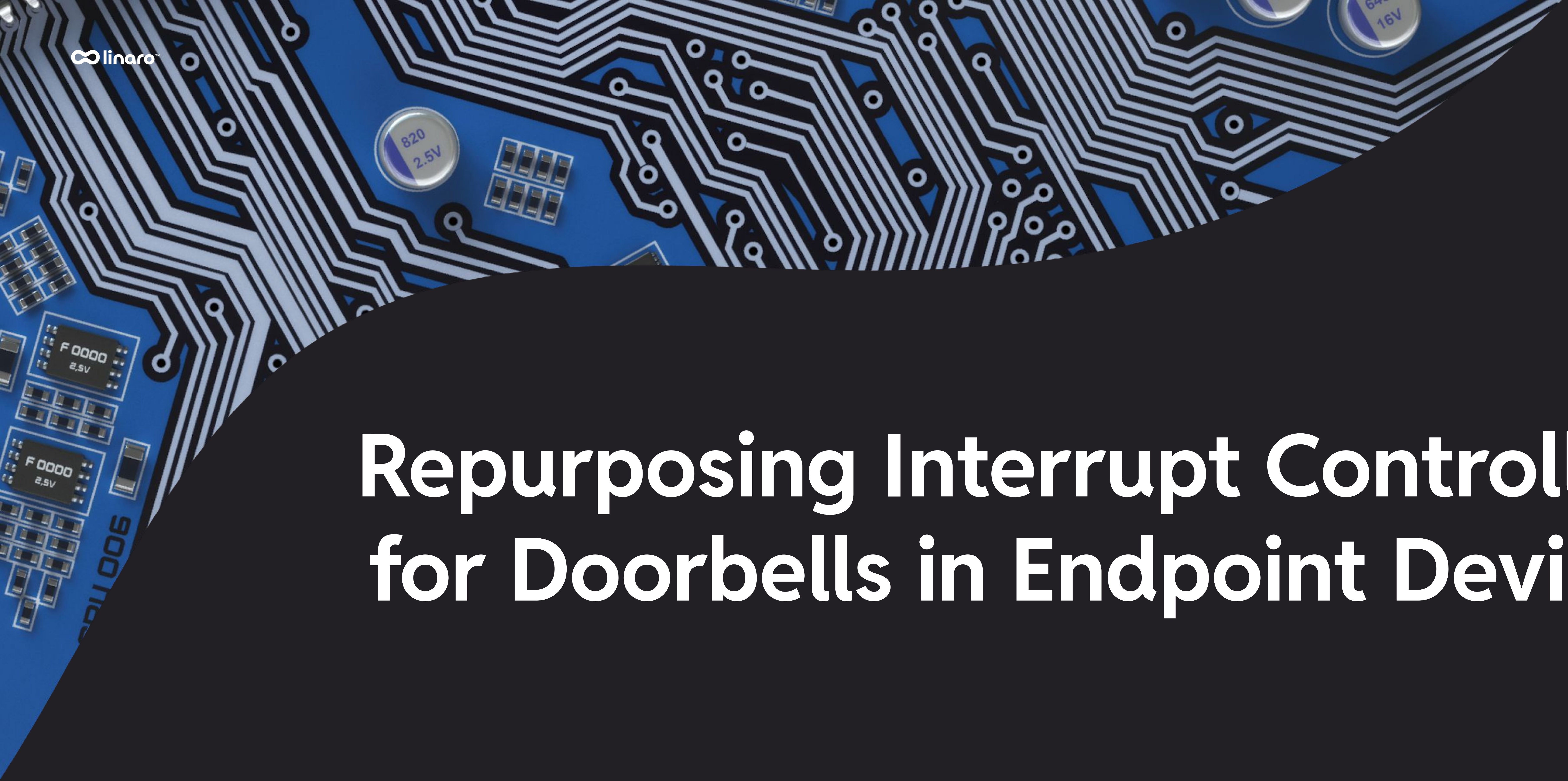
- Proposal from Shunsuke Mie
  - https://lore.kernel.org/qemu-devel/CANXvt5oKt=AKdqv24LT079e+6URnfqJcfTJh0ajGA17paJUEKw@mail.gmail.com/
- Requires 2 Guests on the same host
  - Communication through UNIX domain socket
- Endpoint
  - PCI Endpoint Controller implemented as a QEMU PCI device
  - Requires a new controller driver in Linux Kernel (Guest)
- Host
  - Separate QEMU EPF Bridge device

# Proposal

- Proposal from Shunsuke Mie
  - https://lore.kernel.org/qemu-devel/CANXvt5oKt=AKdqv24LT079e+6URnfqJcfTJh0ajGA17paJUEKw@mail.gmail.com/
- Requires 2 Guests on the same host
  - Communication through UNIX domain socket
- Endpoint
  - PCI Endpoint Controller implemented as a QEMU PCI device
  - Requires a new controller driver in Linux Kernel (Guest)
- Host
  - Separate QEMU EPF Bridge device
  - Exposes the endpoint test device to host

# Proposal

- Proposal from Shunsuke Mie
  - https://lore.kernel.org/qemu-devel/CANXvt5oKt=AKdqv24LT079e+6URnfqJcfTJh0ajGA17paJUEKw@mail.gmail.com/
- Requires 2 Guests on the same host
  - Communication through UNIX domain socket
- Endpoint
  - PCI Endpoint Controller implemented as a QEMU PCI device
  - Requires a new controller driver in Linux Kernel (Guest)
- Host
  - Separate QEMU EPF Bridge device
  - Exposes the endpoint test device to host
  - Talks to the QEMU PCI endpoint controller device on endpoint



# Repurposing Interrupt Controllers for Doorbells in Endpoint Devices

# Problem Statement

# Problem Statement

- PCIe spec doesn’t define interrupts (doorbells) from PCIe RC to EP

# Problem Statement

- PCIe spec doesn’t define interrupts (doorbells) from PCIe RC to EP
  - INTx/MSI/MSI-X are only from PCIe EP to RC

# Problem Statement

- PCIe spec doesn’t define interrupts (doorbells) from PCIe RC to EP
  - INTx/MSI/MSI-X are only from PCIe EP to RC
- Vendors use their own way to send doorbell to EP

# Problem Statement

- PCIe spec doesn’t define interrupts (doorbells) from PCIe RC to EP
  - INTx/MSI/MSI-X are only from PCIe EP to RC
- Vendors use their own way to send doorbell to EP
  - Like triggering interrupt in EP using a register in BAR

# Proposal

# Proposal

- Repurposing interrupt controller in EP to receive doorbell from RC
  - Frank Li - https://lore.kernel.org/linux-pci/20230911220920.1817033-1-Frank.Li@nxp.com

# Proposal

- Repurposing interrupt controller in EP to receive doorbell from RC
  - Frank Li - https://lore.kernel.org/linux-pci/20230911220920.1817033-1-Frank.Li@nxp.com
- EP to expose interrupt vector address and value to write through BAR

# Proposal

- Repurposing interrupt controller in EP to receive doorbell from RC
  - Frank Li - https://lore.kernel.org/linux-pci/20230911220920.1817033-1-Frank.Li@nxp.com
- EP to expose interrupt vector address and value to write through BAR
- Host to write the value to the address for triggering doorbell in EP

# Proposal

- Repurposing interrupt controller in EP to receive doorbell from RC
  - Frank Li - https://lore.kernel.org/linux-pci/20230911220920.1817033-1-Frank.Li@nxp.com
- EP to expose interrupt vector address and value to write through BAR
- Host to write the value to the address for triggering doorbell in EP
- Feedback
  - Thomas Gleixner suggested using IMS

# Thank You!

Visit www.linaro.org

## Contact me at:

manivannan.sadhasivam@linaro.org
linkedin.com/in/manisadhasivam
OFTC/mani_s