Technology Consulting Company
Research, Development &
Global Standard

Virtual device for testing the Linux PCIe endpoint framework

Shunsuke Mie, IGEL CO.,Ltd.
KVM Forum - September 22, 2024

PCI Endpoint Controller

Manages the functionality of devices attached to the PCIe bus.

Some controllers can cooperate with software.

Software can define:

- any device type
- behavior the pcie function

Linux PCI Endpoint Framework

Linux provides a framework to support such the controllers.

The framework works as an abstraction layer to implement a software defined pcie endpoint function.

called PCIe Endpoint Function(EPF) Driver

Testing for Framework and EPF

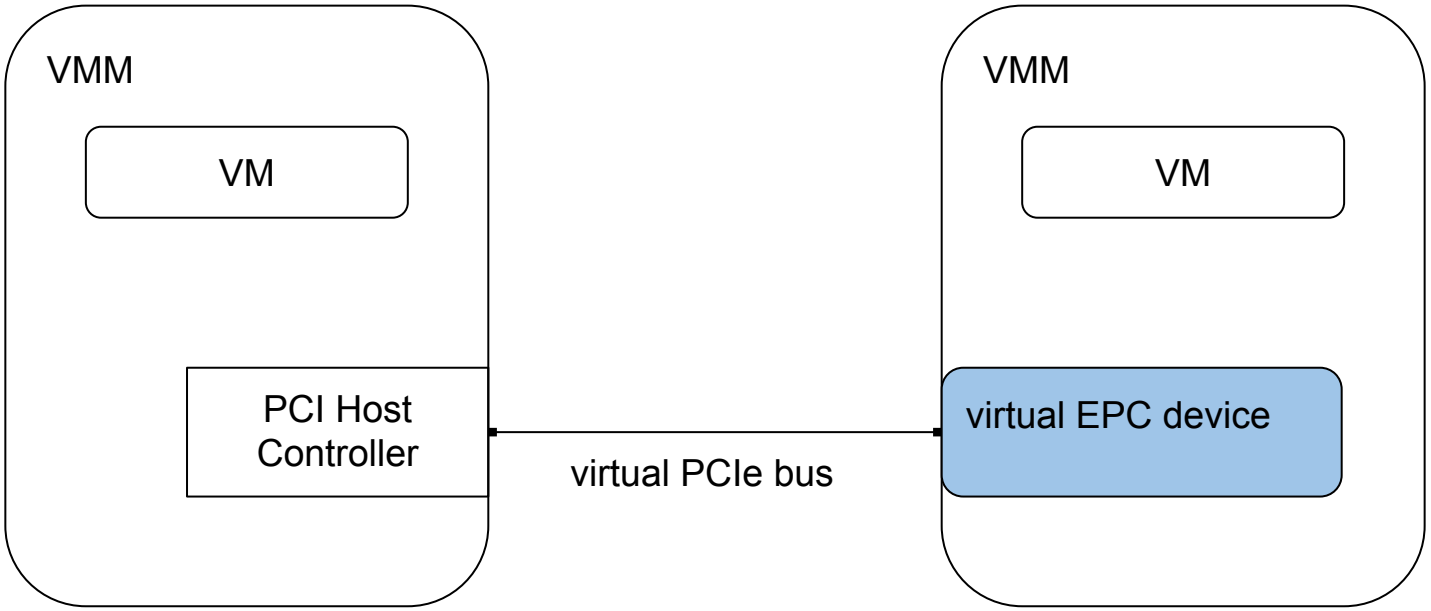
The framework and epf drivers are not fully mature yet.

Developer and tester need physical board and machine.

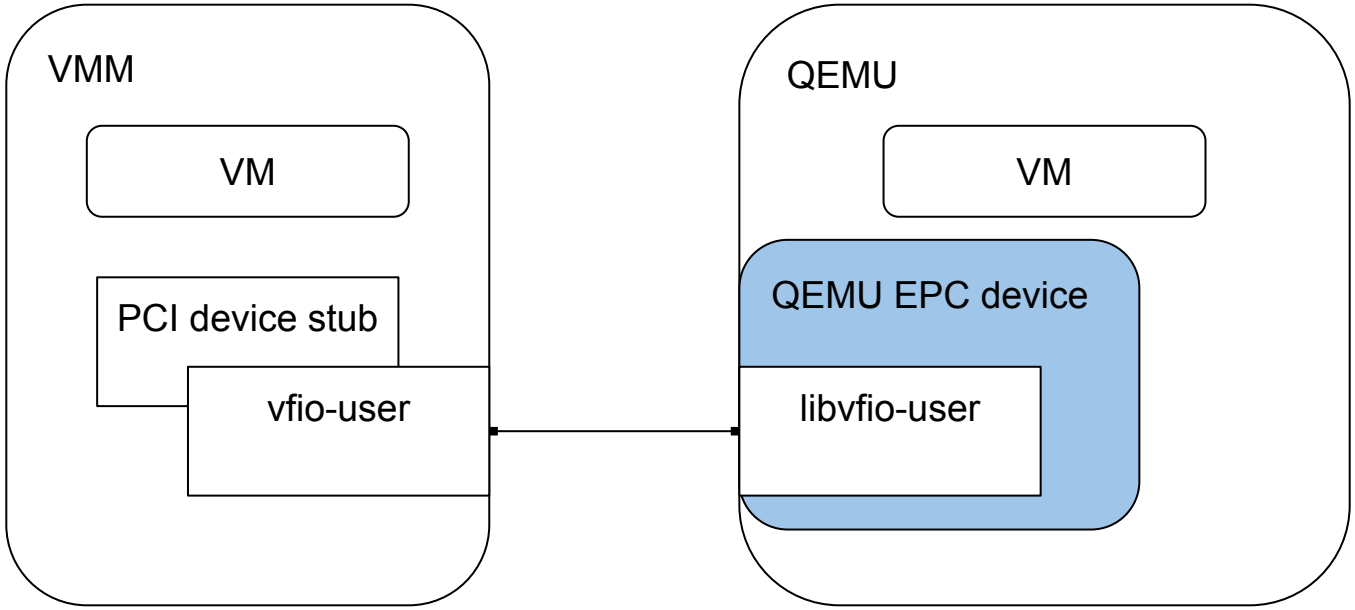
This constraint is hindering the improvement.

To solve this,
⇒ provide an environment that is easy to setup

Virtual EP Environment



QEMU EP Controller - Overview



vfio-user and libvfio-uesr

- Designed by nutanix
- Designed to implement device emulation at outside of vmm on same host
- Client and server model
  - client works as pcie host(root complex)
  - server works as pcie device (endpoint)
- Unix domain socket is used to control plane
- Shared memory is used to data plane
- VMM-agnostic

QEMU perspectives:

libvfio-user is already introduced to improve Multiprocess QEMU

QEMU EPC Design

igel

We designed a custom virtual EP controller.

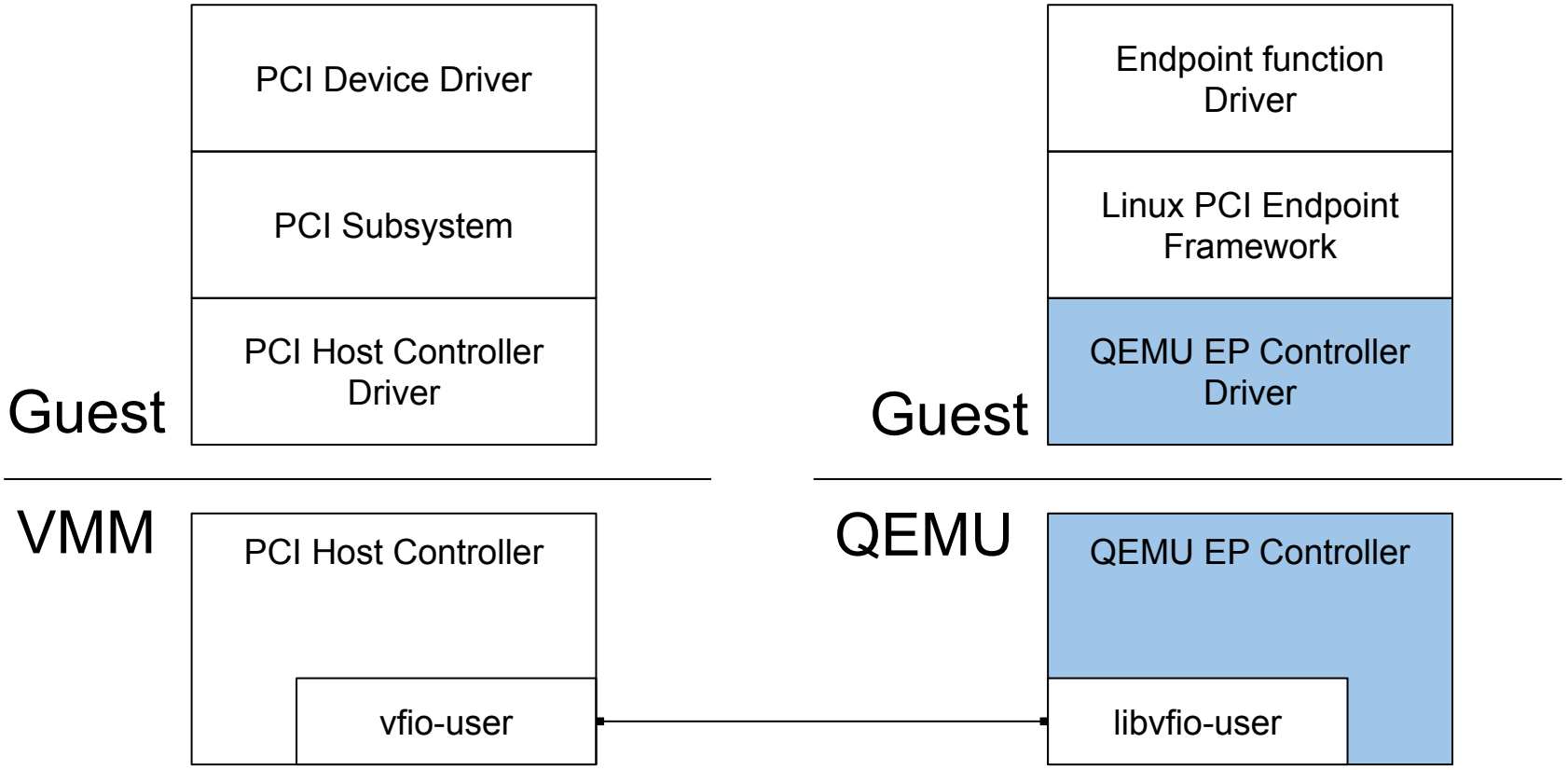
Emulation of existing EP controller has highly implementation cost, because the EPC

- has RC functionality, leading to large feature set
- has a lot of configuration for various embedded system

Disadvantage of the custom controller:

requires custom driver

QEMU EPC - Block diagram



QEMU EPC

igel

QEMU EPC provides functions to operate PCIe transaction from software.

QEMU EPC has 4 major functions:

1. Handling accesses to PCIe configuration space
   a. RC -> EP (config)
2. Handling accesses to device memory region
   a. RC -> EP (memory)
   b. The region indicated by BAR
3. Handling requests to accesses rc memory
   a. EP -> RC (memory)
   b. DMA
4. Raising interrupts
   a. EP -> RC

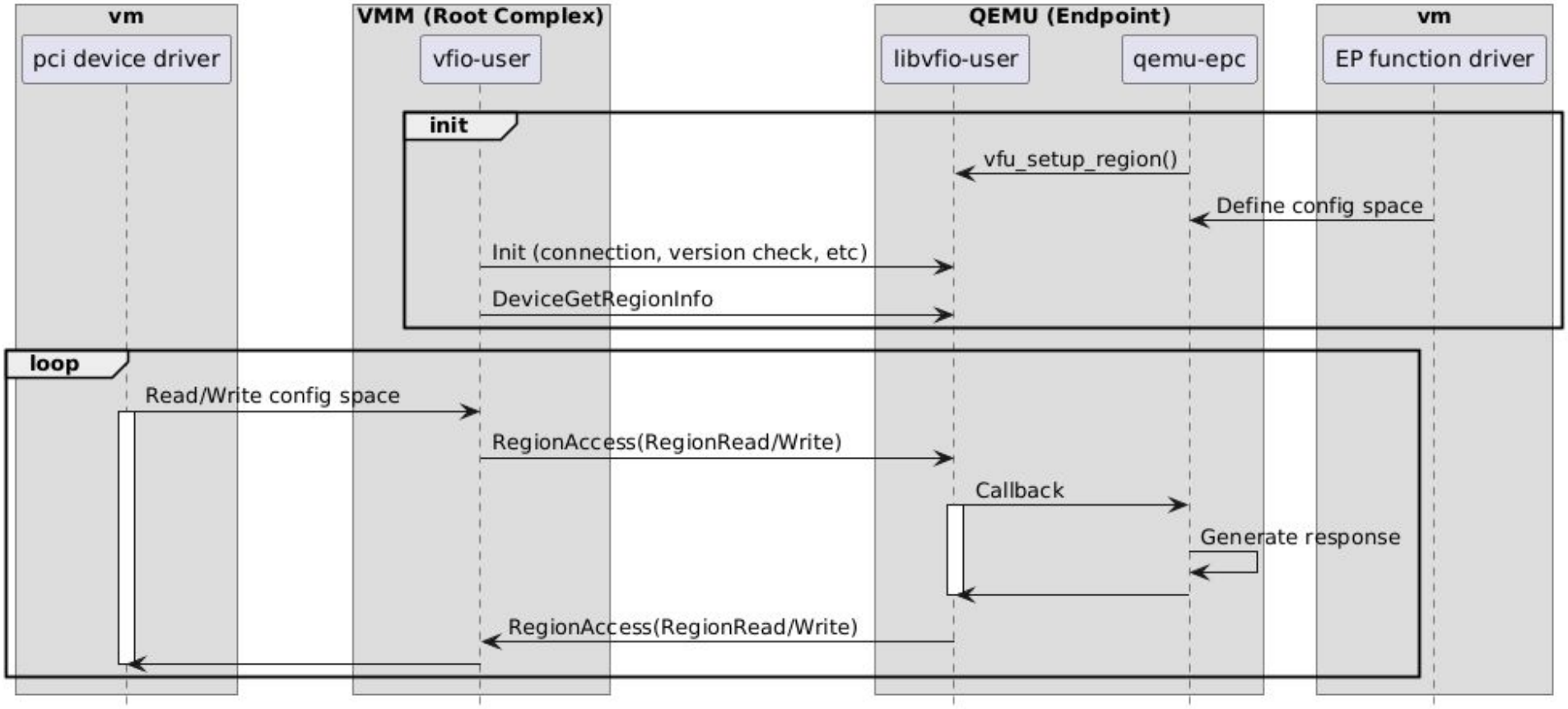
QEMU EPC

QEMU EPC provides to operate PCIe transaction from software.

QEMU EPC has 4 major functions to operate with EPF.

1. Handling accesses to PCIe configuration space
   a. RC -> EP (config)
2. Handling accesses to device memory region
   a. RC -> EP (memory)
   b. The region indicated by BAR
3. Handling requests to accesses rc memory
   a. EP -> RC (memory)
   b. DMA
4. Raising interrupts
   a. EP -> RC

1. Handling Configuration Access



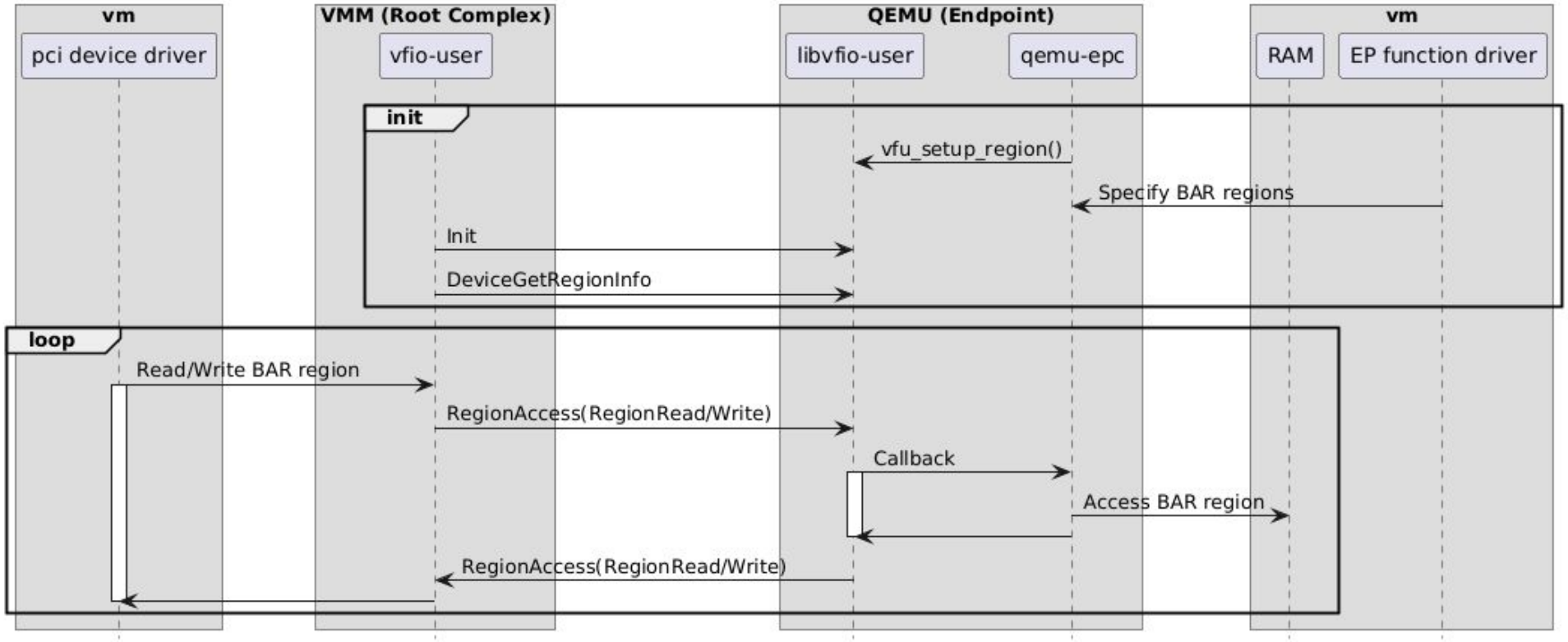
QEMU EPC

QEMU EPC provides to operate PCIe transaction from software.

QEMU EPC has 4 major functions to operate with EPF..

1. Handling accesses to PCIe configuration space
   a. RC -> EP (config)
2. Handling accesses to device memory region
   a. RC -> EP (memory)
   b. The region indicated by BAR
3. Handling requests to accesses rc memory
   a. EP -> RC (memory)
   b. DMA
4. Raising interrupts
   a. EP -> RC

2. Handling EP Memory Access



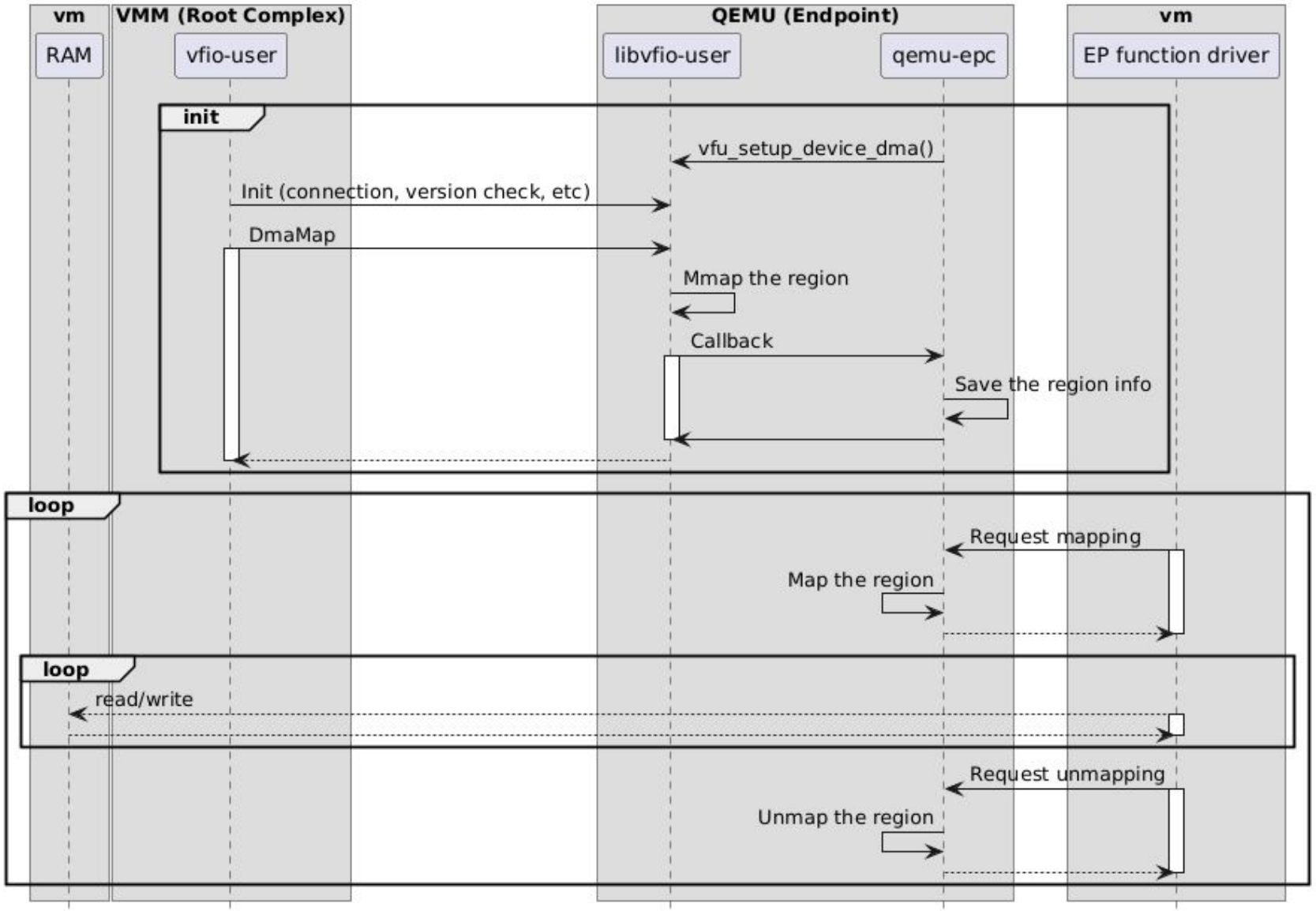
QEMU EPC

QEMU EPC provides to operate PCIe transaction from software.

QEMU EPC has 4 major functions to operate with EPF..

1. Handling accesses to PCIe configuration space
   a. RC -> EP (config)
2. Handling accesses to device memory region
   a. RC -> EP (memory)
   b. The region indicated by BAR
3. Handling requests to accesses rc memory
   a. EP -> RC (memory)
   b. DMA
4. Raising interrupts
   a. EP -> RC

3. Handling RC Memory Access



QEMU EPC

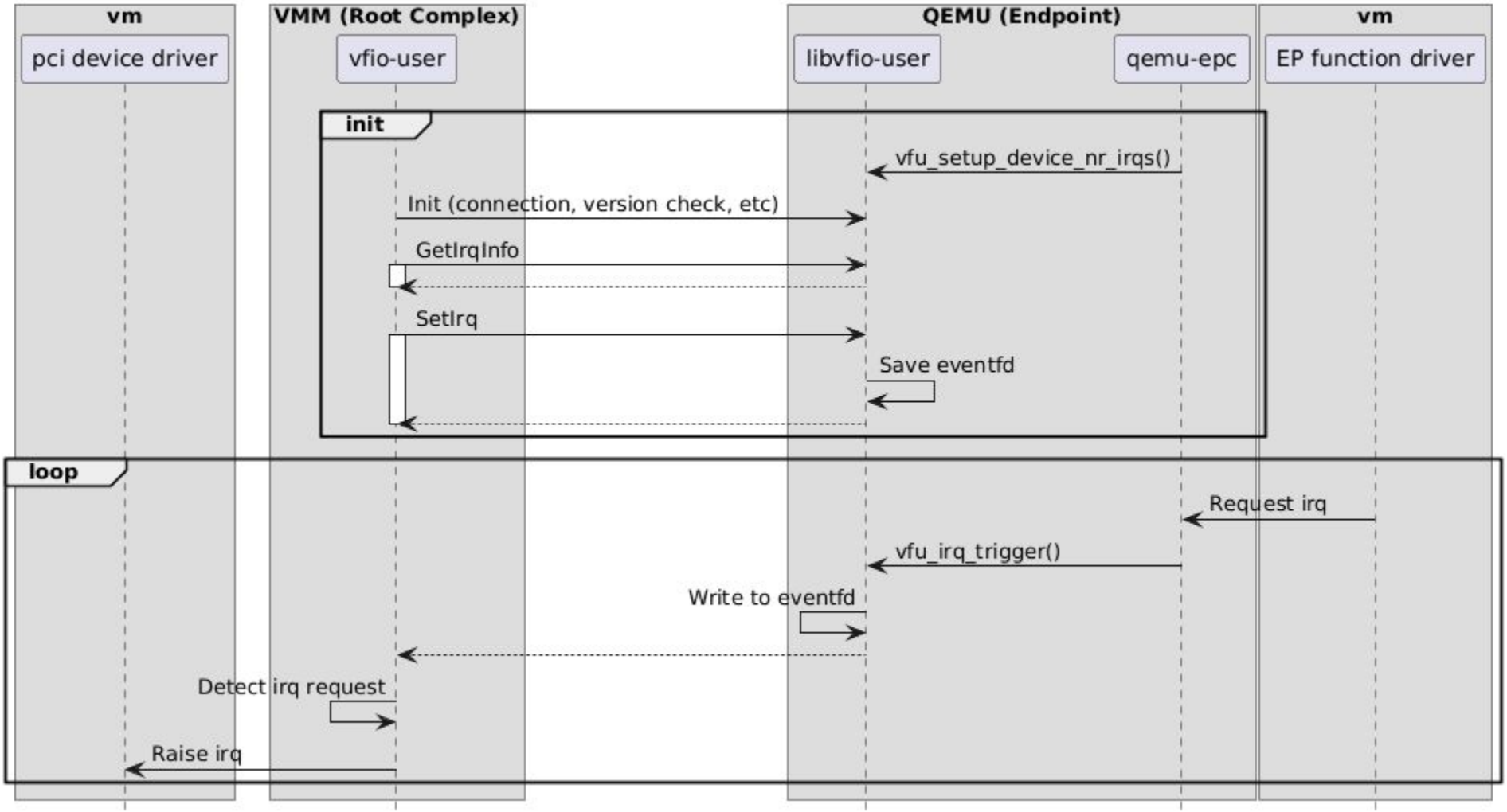
igel

QEMU EPC provides to operate PCIe transaction from software.

QEMU EPC has 4 major functions to operate with EPF..

1. Handling accesses to PCIe configuration space
   a. RC -> EP (config)
2. Handling accesses to device memory region
   a. RC -> EP (memory)
   b. The region indicated by BAR
3. Handling requests to accesses rc memory
   a. EP -> RC (memory)
   b. DMA
4. Raising interrupts
   a. EP -> RC

4. Raising Interrupts



Demo

igel

Following tools are used for demo.

VMM (Root Complex side)

- Cloud-Hypervisor

VMM (Endpoint side)

- QEMU with the virtual EPC device

OS

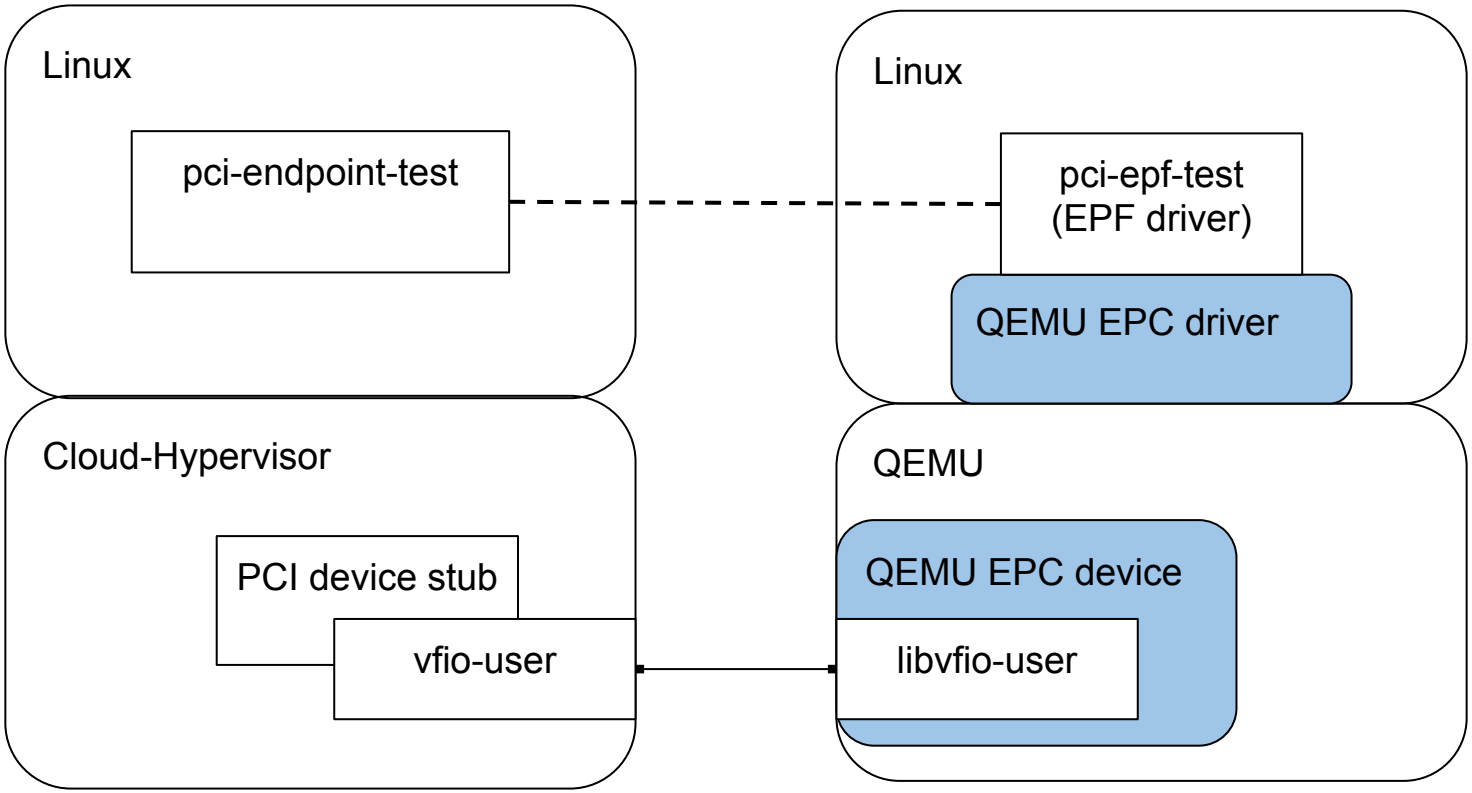
- Linux (both side)

drivers and tools:

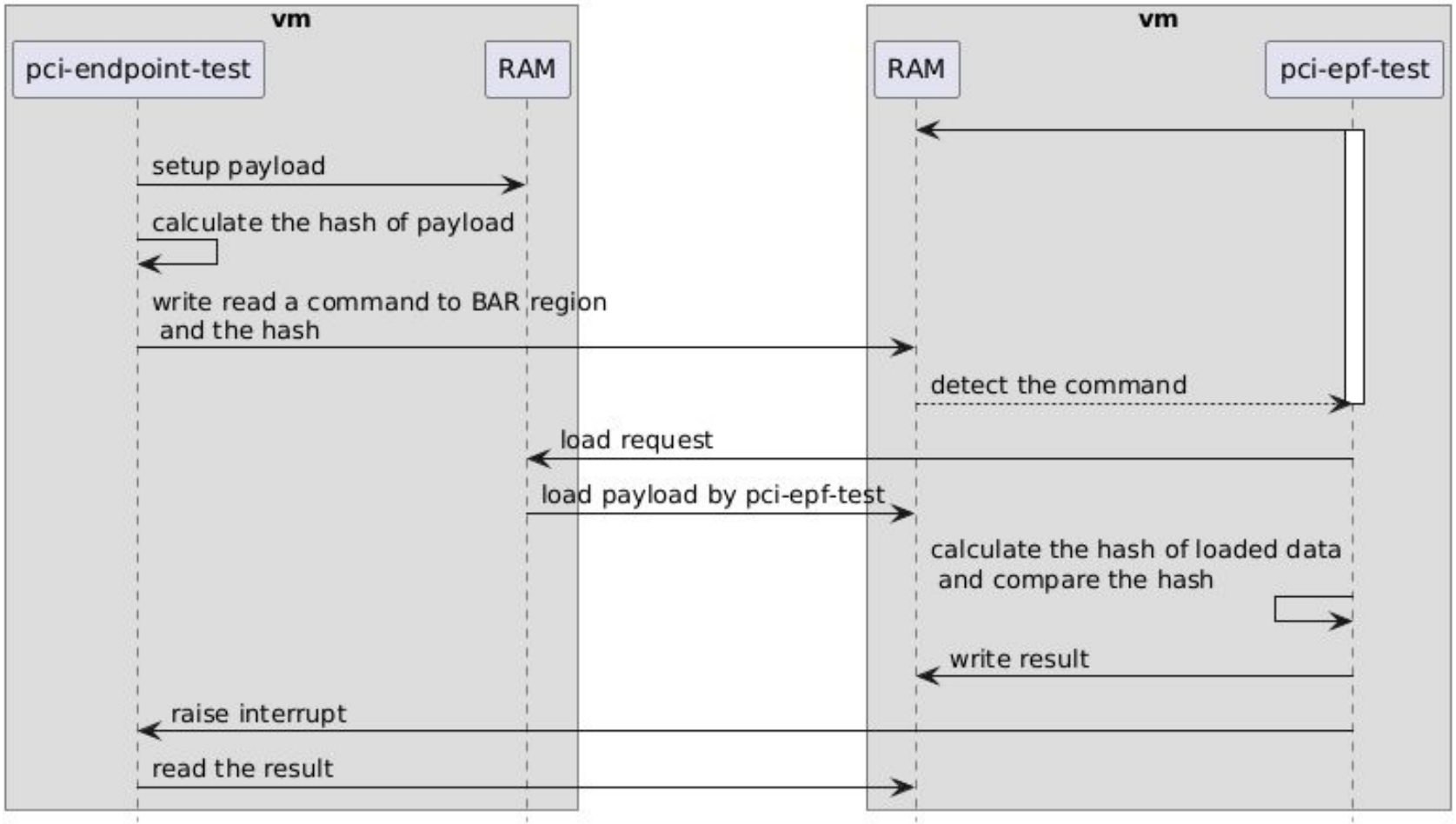
- pci-epf-test (EPF driver)
- pci-endpoint-test (Host driver for the EPF)
- pcitest (Userland app to run the host driver)

Developed to test functionality the EPF, EPC

Demo - Overview



pci-epf-test - Flow(write cmd)



Future Works

igel

Remains

- Fix buggy code
- Support PCI DMAC
- Live Migration

Physical PCIe EPC emulation

- Designware, Rockchip, Cadence, …?

Thank you for your listening!