

# USBManager for USB Device Pass-Through

## Motivation

Users need access to USB devices from virtual machines; currently for that purpose pass-through of the whole USB controller (via PCI pass-through) is done. This comes with two big disadvantages:

1. PCI pass-through often has problems with IOMMU groups
2. Individual devices connected to USB ports (on the same controller) cannot be passed through to different virtual machines

The current implementation is using pass-through of USB devices via:

1. bus number (which mostly corresponds with the PCI address of the controller)
2. the device number - a simple counter to enumerate plugging in USB devices; example:
  - insert a mouse -> device number 3
  - insert a keyboard -> device number 4
  - remove the mouse
  - insert the same mouse again -> device number 5

## History

Currently there is an implementation in EVE writing bus number and device number into the Qemu config file.

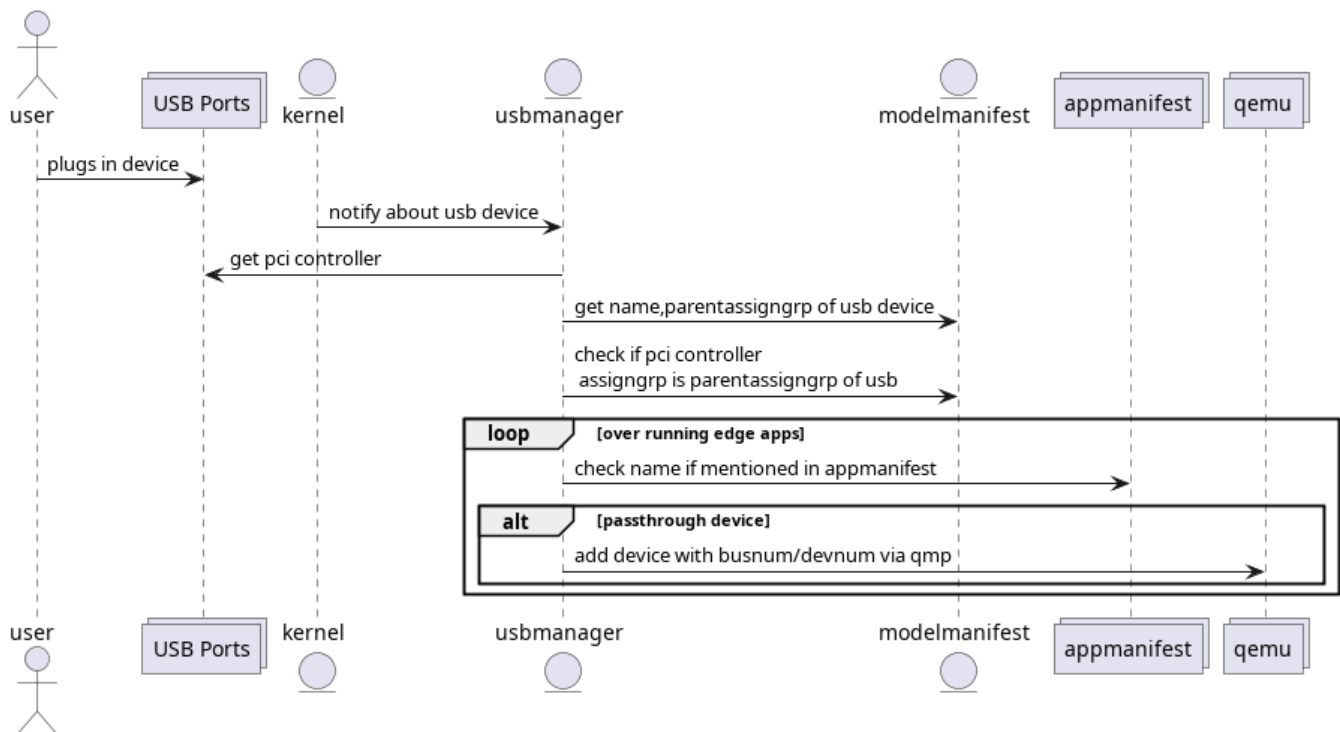
Disadvantages of this approach are:

1. Qemu does not inform the virtual machine of removal events
2. Removal and re-insertion of the same device does not add the device to the virtual machine again
3. The virtual machine does not start if the USB device is not plugged in

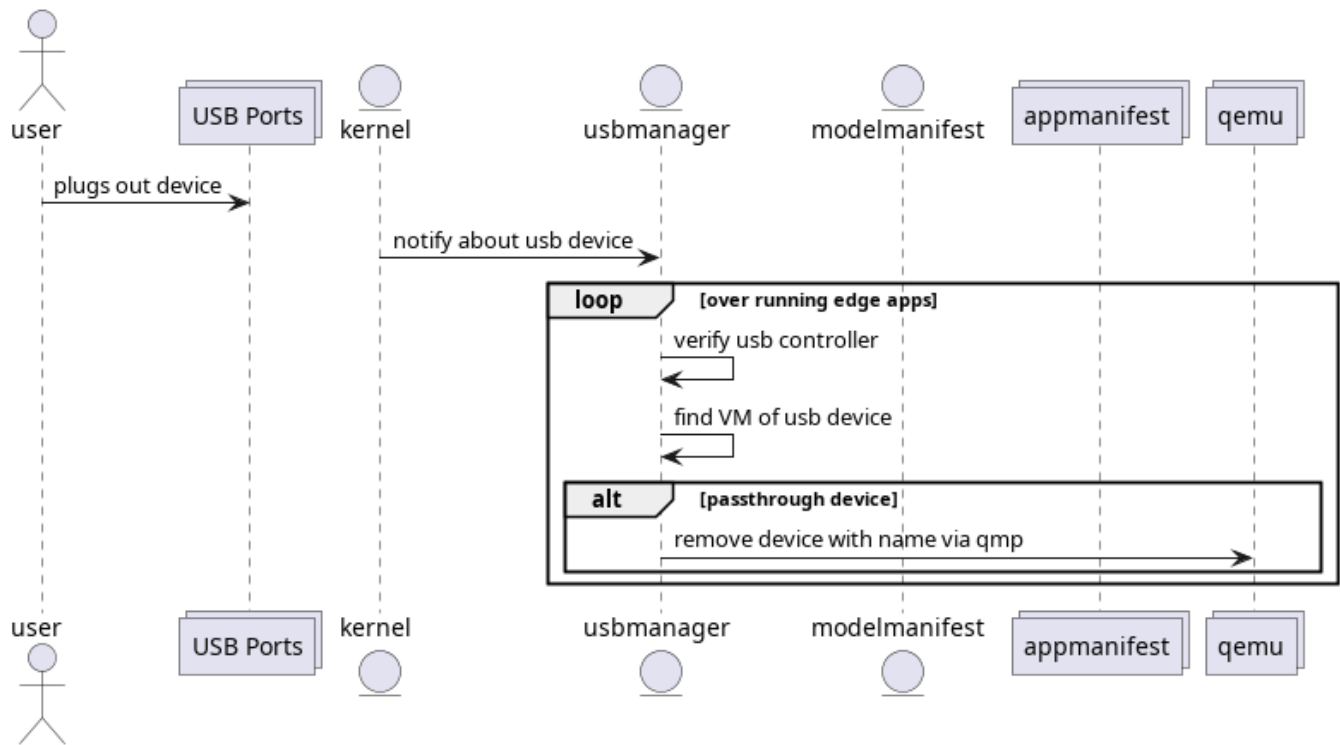
## Proposed EVE architecture

In order to be more flexible a new component is introduced that is called usbmanager; is listening itself to USB events and decides whether at all and to which virtual machine a USB device will be connected to. Once the decision is made, usbmanager uses QMP to tell Qemu to pass-through the USB device (via bus number and device number).

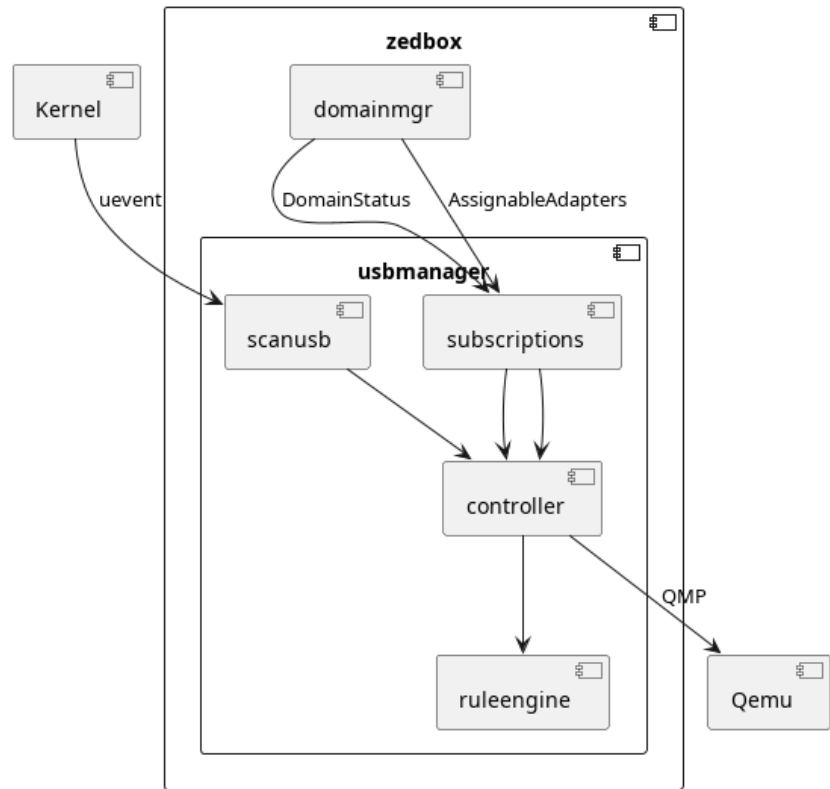
When the user plugs in a USB device:



When the user plugs out a USB device:



In particular the usbmanager is interacting with several components:



# Usage

## Pass-Through by USB Port

Use `lsusb -vt` to see which device is connected to which port:

```
/: Bus 03.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/12p, 480M
ID 1d6b:0002 Linux Foundation 2.0 root hub
|__ Port 9: Dev 22, If 0, Class=Human Interface Device, Driver=usbhid, 12M
   ID 045e:0823 Microsoft Corp. Classic IntelliMouse
```

This means that the device is connected via bus number 3 and port 9.

If there is a USB hub in between, then it might look like this:

```
/: Bus 03.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/12p, 480M
ID 1d6b:0002 Linux Foundation 2.0 root hub
|__ Port 1: Dev 28, If 0, Class=Hub, Driver=hub/4p, 480M
   ID 2109:2824 VIA Labs, Inc.
   |__ Port 2: Dev 33, If 0, Class=Human Interface Device, Driver=usbhid, 12M
      ID 045e:0823 Microsoft Corp. Classic IntelliMouse
```

This means the bus number is 3 and the port is 1.2.

Now the following can be added to the model manifest:

```
"ioMemberList": [
  {
    "ztype": "IO_TYPE_USB_CONTROLLER",
    "phylabel": "USB Controller Phy",
    "assigngrp": "USB-controller-1",
    "phyaddrs": {
      "PciLong": "0000:14:00.0",
    },
    "logicallabel": "USB Controller",
    "usagePolicy": { }
  },
  {
    "ztype": "IO_TYPE_USB_DEVICE",
    "phylabel": "USB device 9",
    "assigngrp": "USB",
    "phyaddrs": {
      "usbaddr": "003:9",
    },
    "logicallabel": "USB device 9",
    "parentassigngrp": "USB-controller-1"
  }
]
```

Please note `parentassigngrp` which makes the dependency between the USB device and the USB controller (connected via PCI).

## Pass-Through by USB Vendor ID / Product ID

Use `lsusb` to see the connected USB devices:

```
Bus 003 Device 033: ID 045e:0823 Microsoft Corp. Classic IntelliMouse
```

The vendor id is 045e (i.e. Microsoft <http://www.linux-usb.org/usb.ids> ) and the product id is 0823.

The following can be added to the model manifest to represent this device:

```
"ioMemberList": [
  {
    "ztype": "IO_TYPE_USB_CONTROLLER",
    "phylabel": "USB Controller Phy",
    "assigngrp": "USB-controller-1",
    "phyaddrs": {
      "PciLong": "0000:14:00.0",
    },
    "logicallabel": "USB Controller",
    "usagePolicy": { }
  },
  {
    "ztype": "IO_TYPE_USB_DEVICE",
    "phylabel": "IntelliMouse",
    "assigngrp": "USB",
    "phyaddrs": {
      "usbproduct": "045e:0823",
    },
    "logicallabel": "IntelliMouse",
    "parentassigngrp": "USB-controller-1"
  }
]
```

## Restrictions

- Devices connected to a USB hub can be forwarded, but USB hubs themselves unfortunately cannot be forwarded
- The physical port on the device might have a different bus number and port number depending on the usb version of the connected device!
- It (currently) only works with Qemu.
- Pass-through of devices that do PCI over Thunderbolt/USB4 is not supported (might work with PCI pass-through, though)