IOMMUFD – Choice of Adapting IOMMU Advancements to Userspace Drivers

Yi Liu

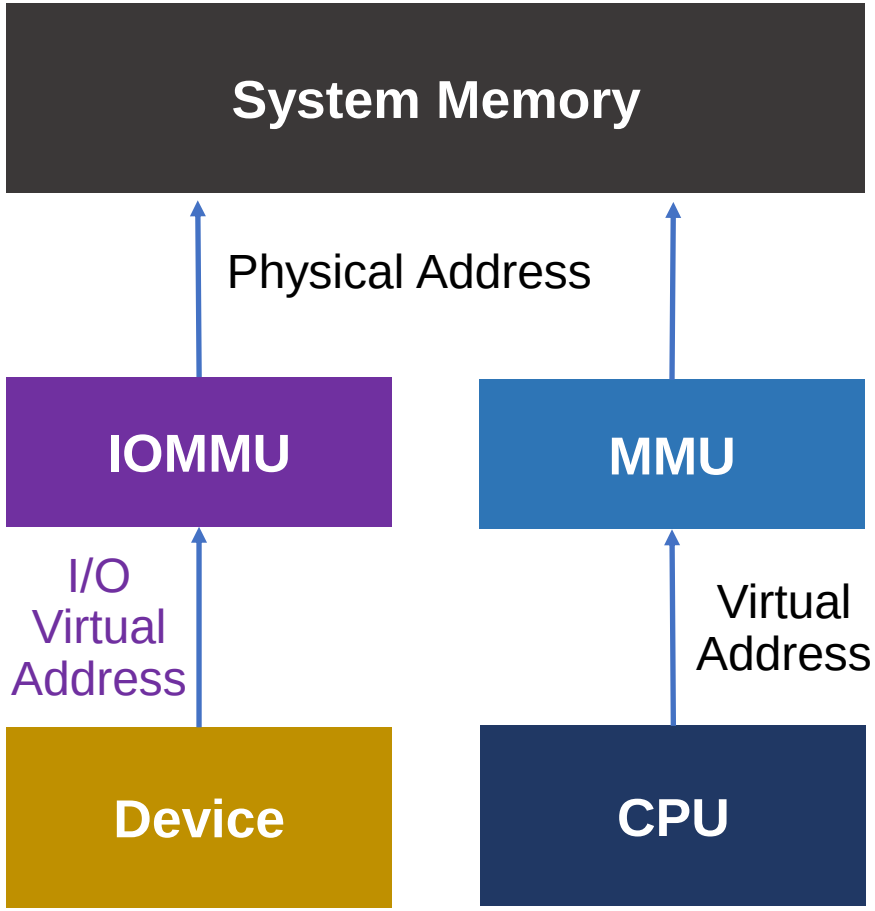
Dec. 26th, 2021

Agenda

- Backgrounds
  - IOMMU Recap
  - Userspace Driver Recap
  - Challenges for software
- IOMMUFD Proposal
  - Key concepts of IOMMUFD proposal
  - Basic flow of IOMMUFD usage
  - Adapting existing frameworks to IOMMUFD
- Status & Future work
- Q&A

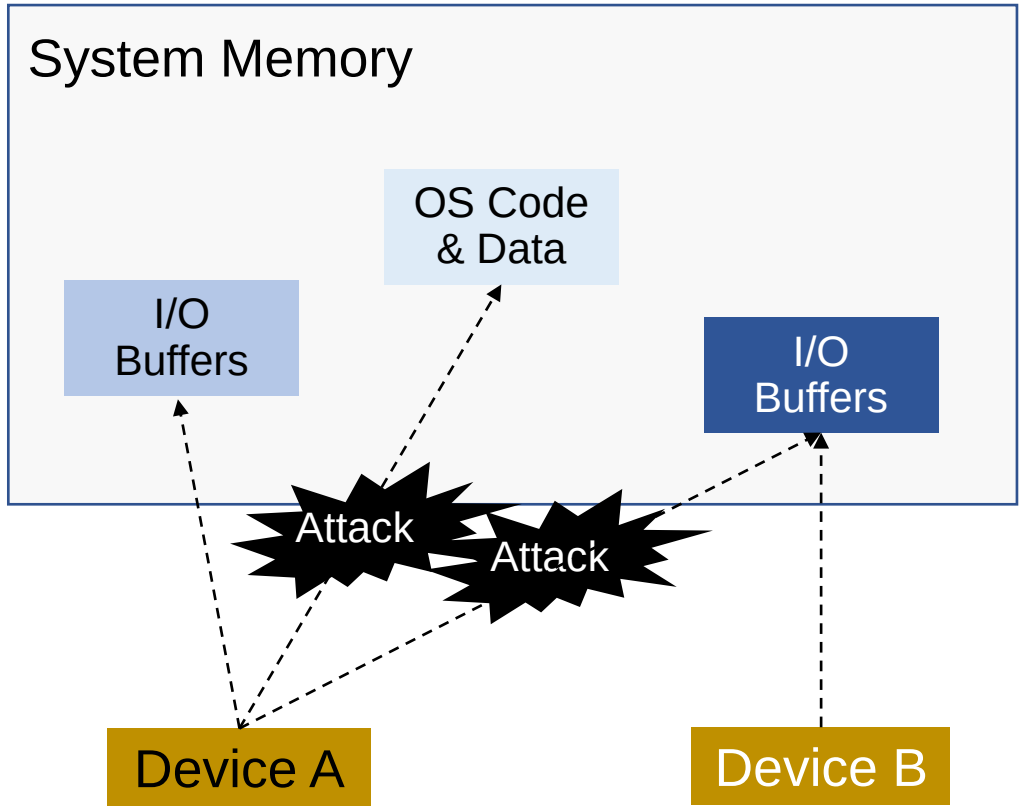
IOMMU Recap

- IOMMU (**I**/**O** **M**emory **M**anagement **U**nit)
- It connects a Direct Memory Access (DMA) capable I/O bus to the system memory[1]
- I/O Virtual Address (IOVA) is used in device memory access
- I/O Page table is used for the IOVA to PA translation

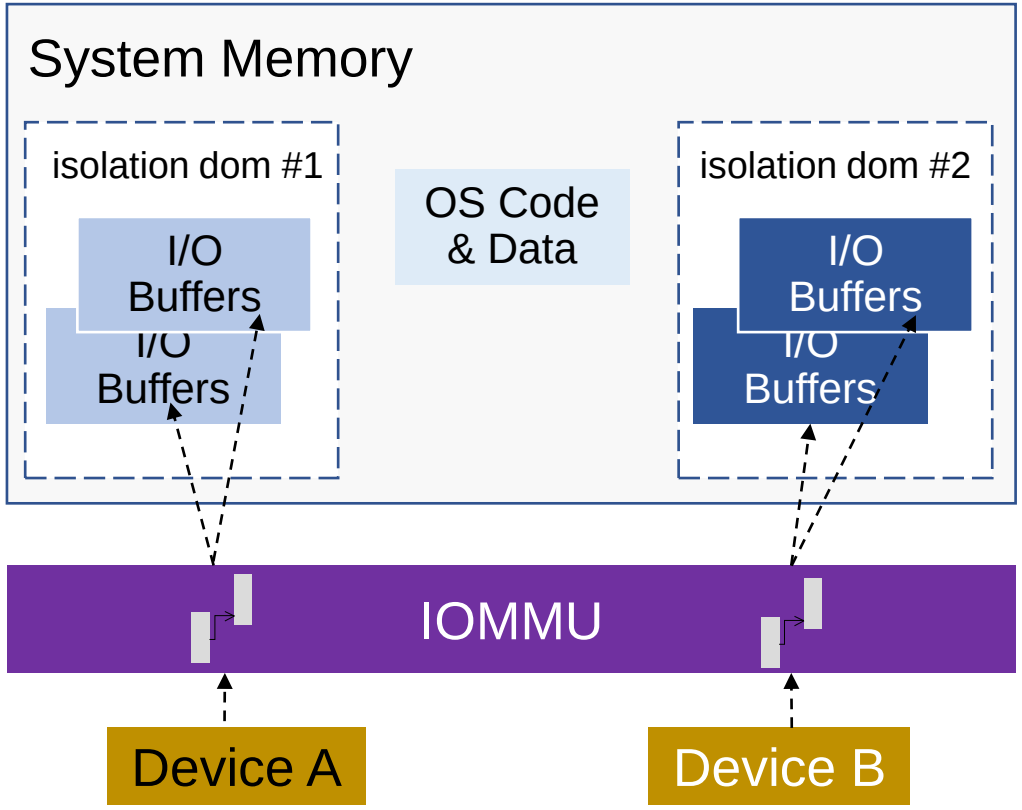


[1] https://en.wikipedia.org/wiki/Input%E2%80%93output_memory_management_unit

Direct Memory Access (DMA) Isolation



DMA w/o IOMMU



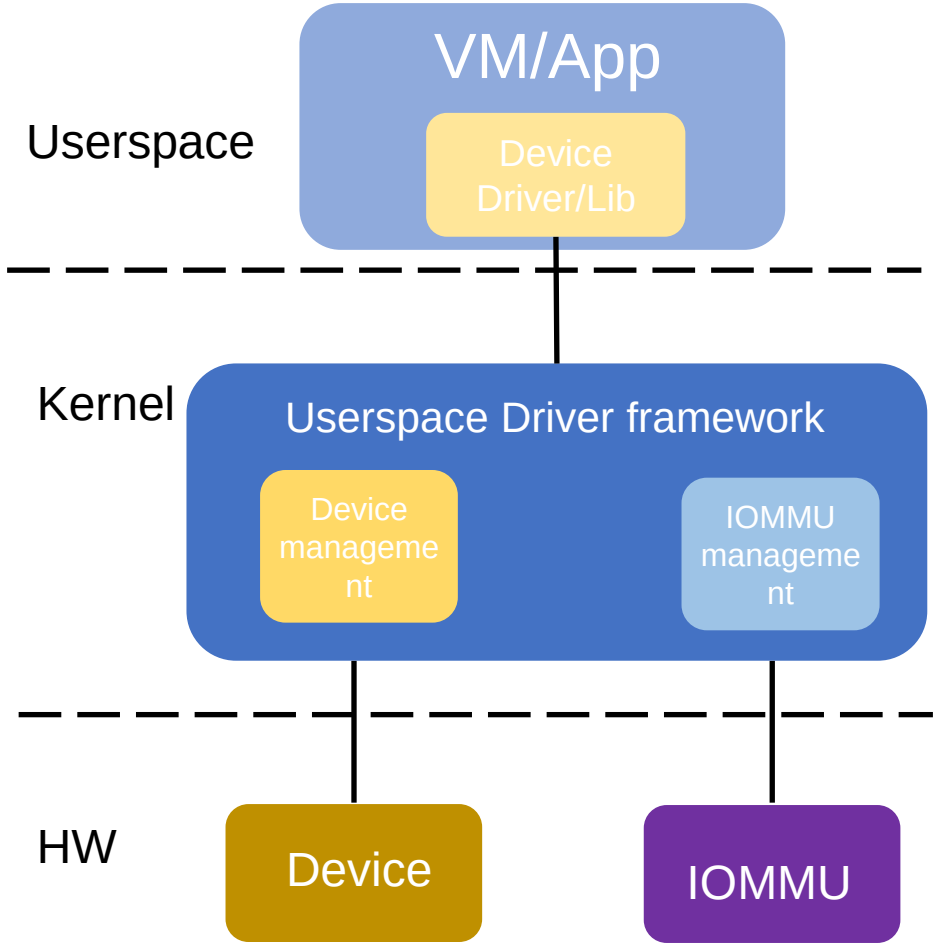
DMA w/ IOMMU

IOMMU Advancements

Feature	Usage	Requirements to software
PASID Granular Translation	SIOV, SVM	PASID management and page table management
Nested Translation	vIOMMU	Support user-managed page table and IOTLB invalidation from page table owner (e.g., QEMU/VM)
Dirty bit support in I/O Page Table	Live migration	Enable/disable dirty tracking, large page splitting during the dirty page logging and merge afterward
I/O Page Fault	Eliminate page pinning on DMA buffers	Forward I/O page fault to page table owner (can be either kernel or user, e.g., user-managed page table owned by user)

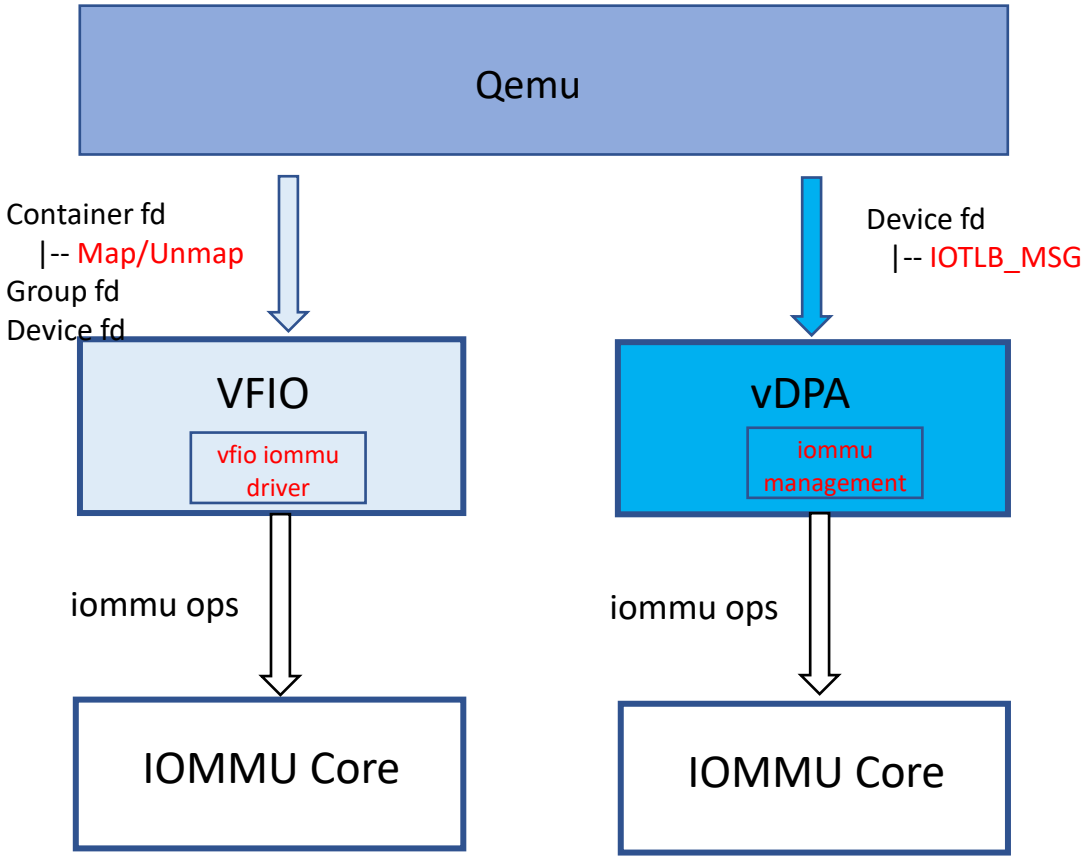
Userspace Driver Recap

- Userspace driver framework
  - Device management
    - Handle the device access like PCI configuration space r/w, BAR mmap, interrupts, etc.;
    - Example: vfio-pci, vfio-platform;
  - IOMMU management
    - Map/unmap the userspace memory
    - Example: vfio iommu type1 driver;
  - Current userspace driver frameworks
    - VFIO (Virtual Function I/O), vDPA (Virtual data path acceleration);



Challenges at Today and Future

- Existing problem
  - Unable to share I/O page table between devices managed by different userspace driver frameworks (e.g., vfio device and vdpa device)
    - Reduced IOTLB efficiency
    - Extra memory footprint
- Not scaled if supporting a new IOMMU feature requires duplicated logic in every userspace driver framework



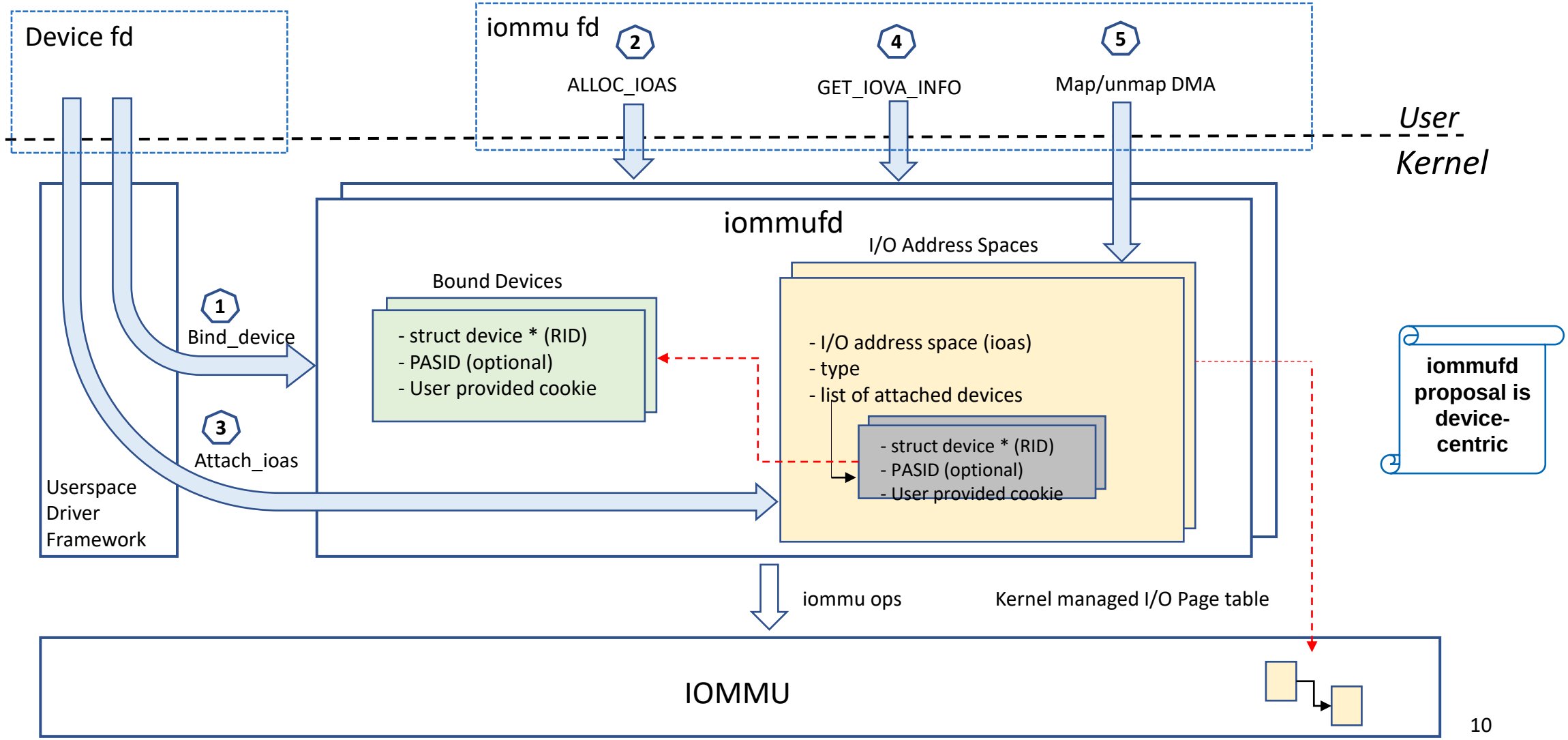
IOMMUFD

- iommufd is a new framework dedicated for managing I/O page tables for devices managed by userspace drivers
  - Consolidates all userspace iommu operations and interactions between userspace driver and kernel
    - map userspace memory in I/O page table, forward I/O page fault and response etc.
- The single portal of supporting new IOMMU advancements for all userspace driver frameworks
  - Improved IOTLB efficiency and less memory footprint compared with today
  - Simplified maintenance model

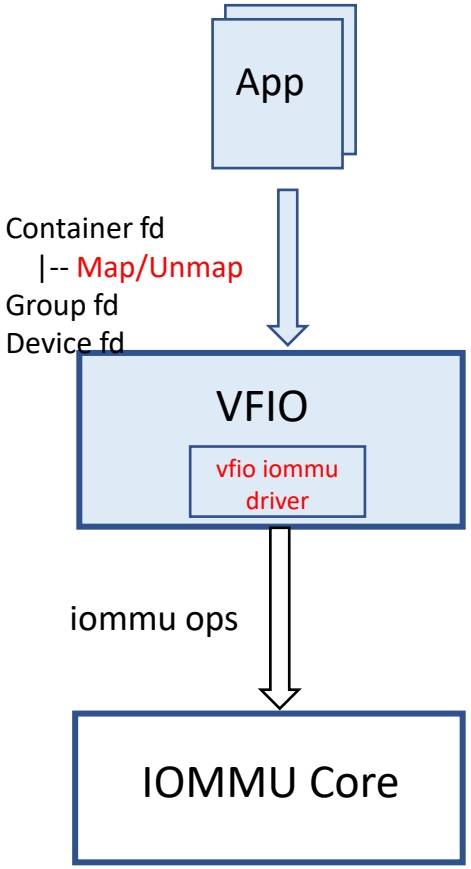
Key Concepts of IOMMUFD

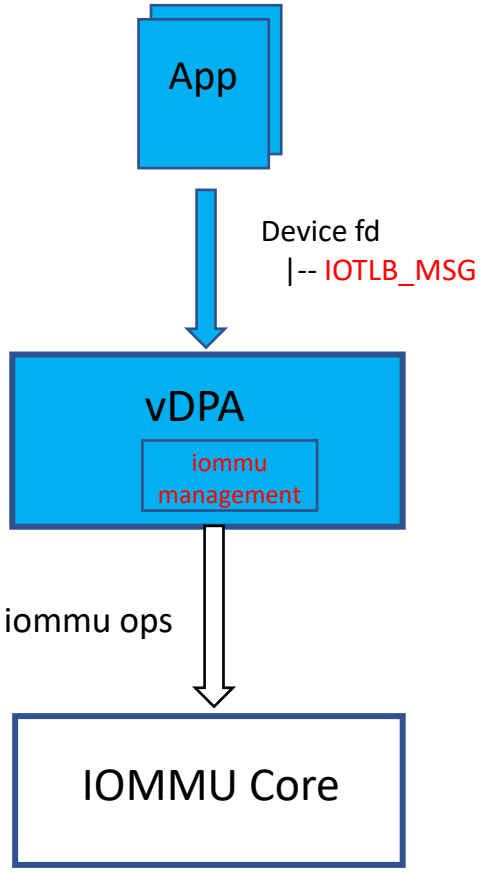
- IOMMU FD
  - Created per /dev/iommu opening
  - Hold multiple I/O address spaces for the current process
  - Support the iommu operations (e.g., DMA map/unmap) from userspace
- IOAS
  - An iommufd-local software handle representing an I/O address space
  - Allocated via iommufd
- Helper for interacting with userspace driver frameworks
  - Bind/unbind device to iommufd
  - Attach/detach selected IOAS to device

Basic Flow of IOMMUFD

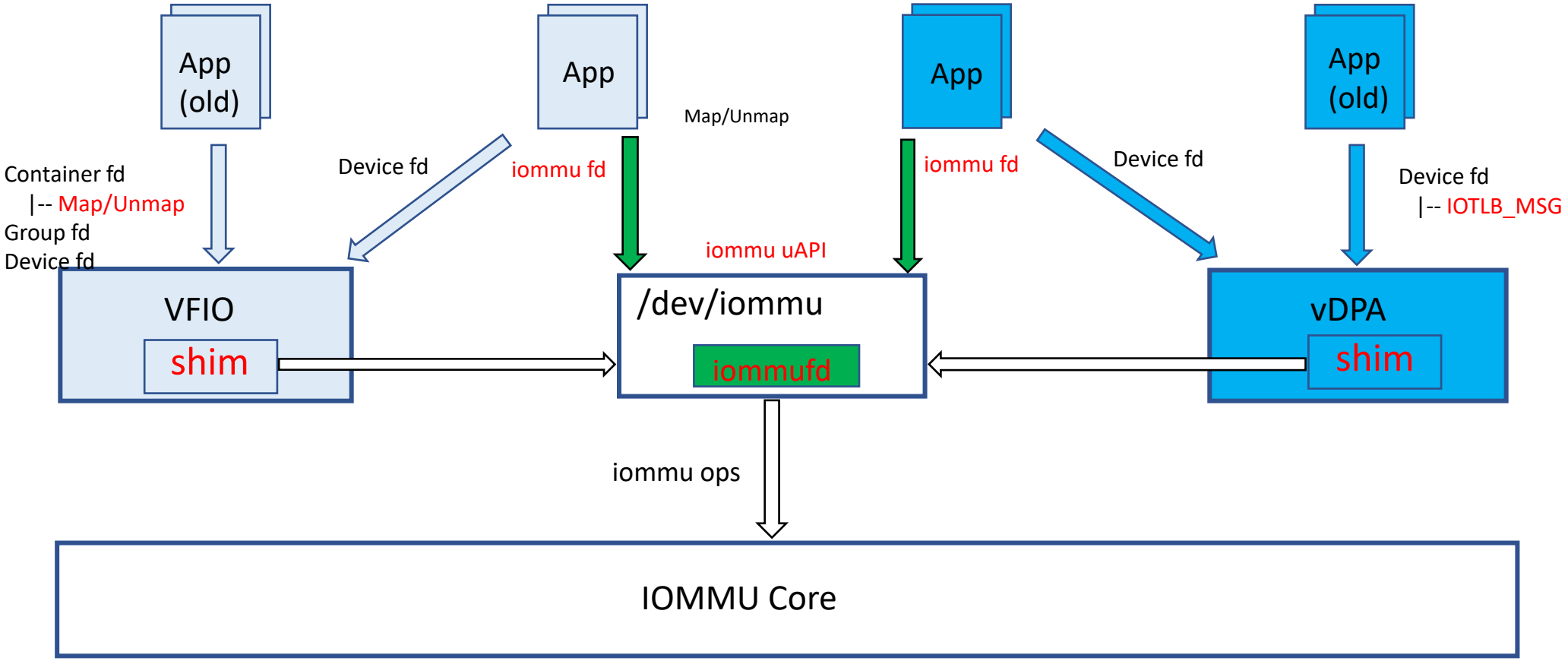


Adapting VFIO/vDPA to IOMMUFD

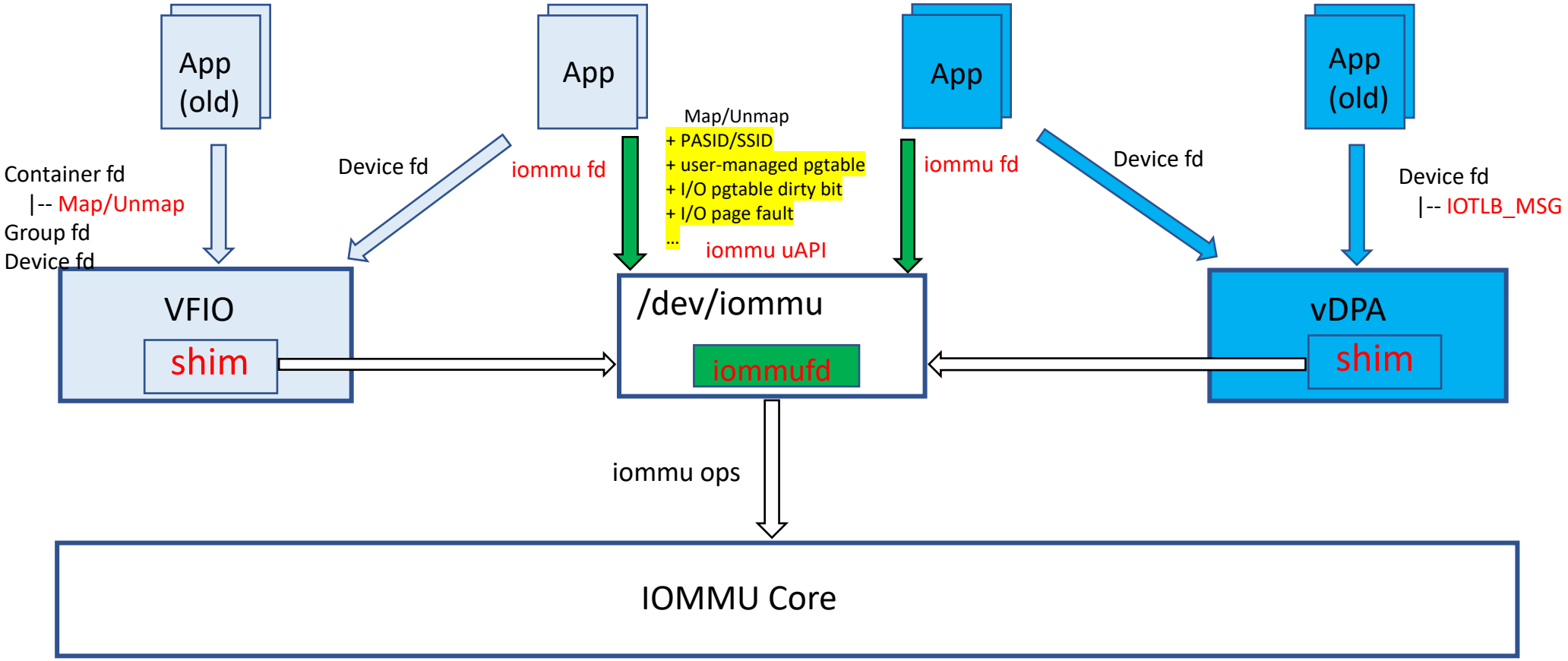




Adapting VFIO/vDPA to IOMMUFD (Cont.)



Adapting VFIO/vDPA to IOMMUFD (Cont.)



Status & Future Work

- Status
  - RFCv1was sent out
    - https://lwn.net/Articles/869818/
    - Only PCI devices and only basic DMA map/unmap feature supported by iommufd in RFCv1.
  - QEMU part change was not sent out yet
  - A work aiming at enabling iommufd usage in DPDK is in progress
    - https://github.com/luxis1999/iommufd

- Future Works
  - Enable the iommufd basic flow in QEMU usage
  - Consolidate the iommu kernel APIs regards to the requirement of SIOV and SVM features
  - Enable new iommu features in iommufd

Summary

- Emerging IOMMU advancements brings new requirements and complexity to software
- Existing userspace driver framework implementations doesn’t scale as new IOMMU features step in
- IOMMUFD is a new framework dedicated for managing I/O page tables for all kinds of devices managed by userspace driver
- IOMMUFD accelerates supporting new IOMMU features (Scalable IOV, SVM, dirty bit tracking, I/O page fault etc.) in Linux world

Q&A