

# Generic Buffer Sharing Mechanism for Mediated Devices

Tina Zhang  
tina.zhang@intel.com

# Agenda

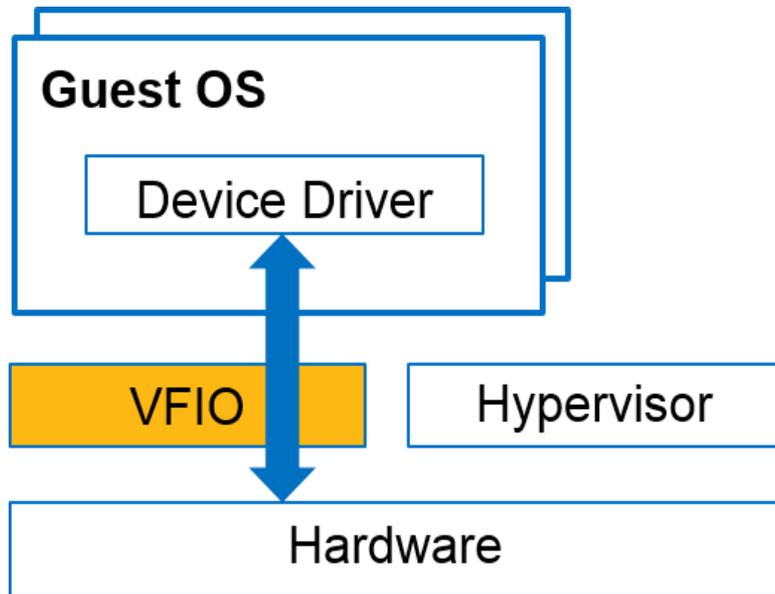
Background

Generic Buffer Sharing in MDEV Framework

Status

Summary

# Virtual Function I/O

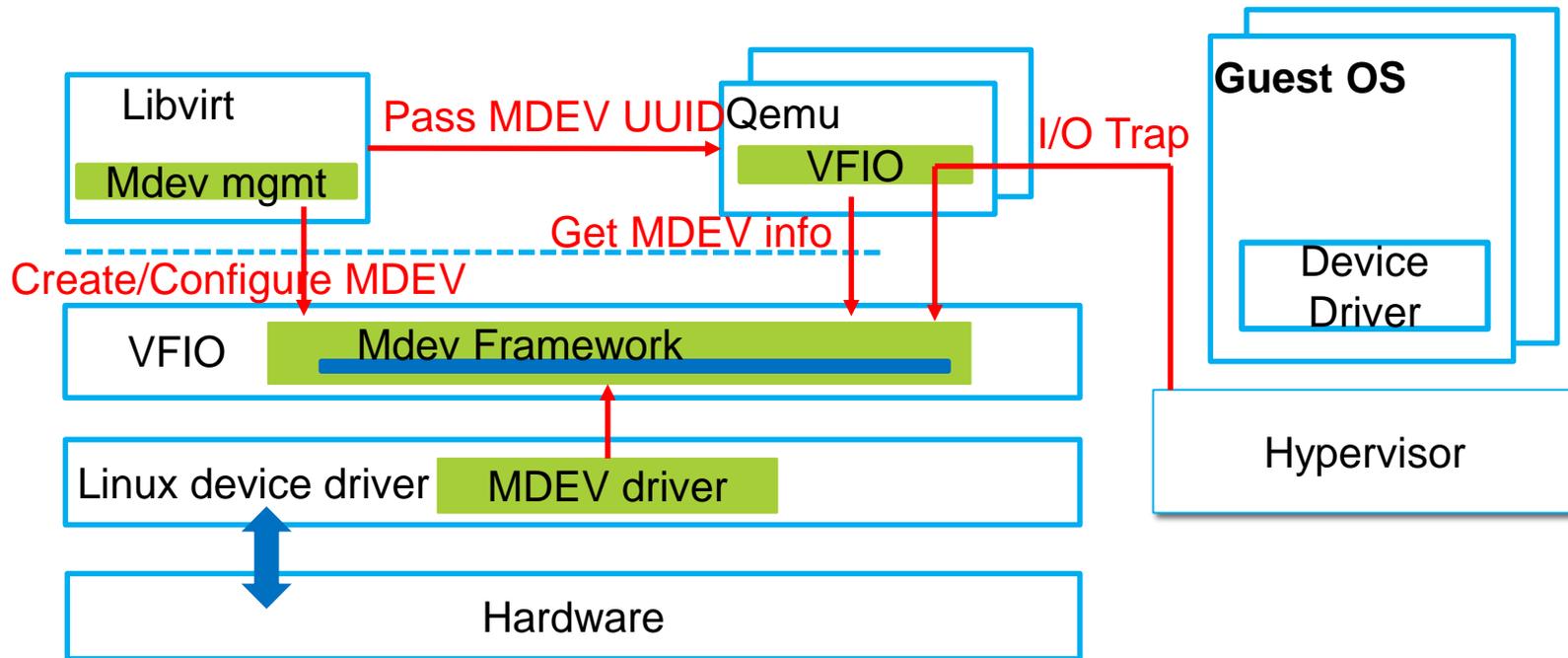


## Virtual Function I/O

- Secure, userspace driver framework
- Assigns physical I/O device to VMs

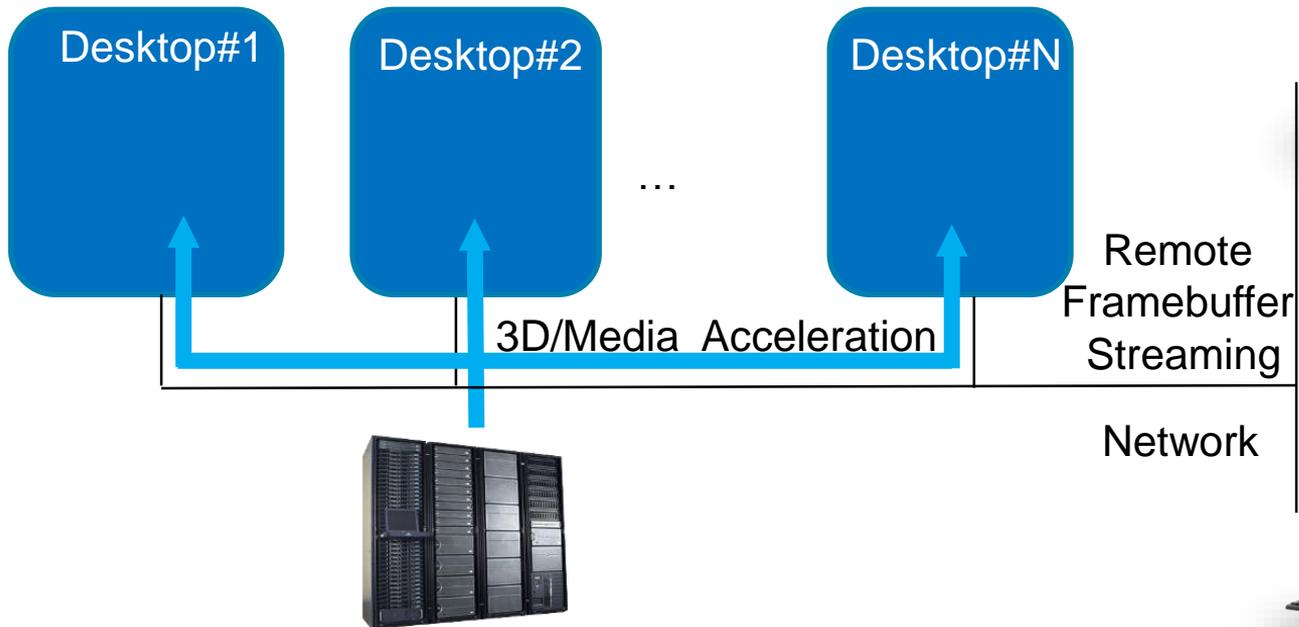
# Mediated Device in VFIO

## Host Linux

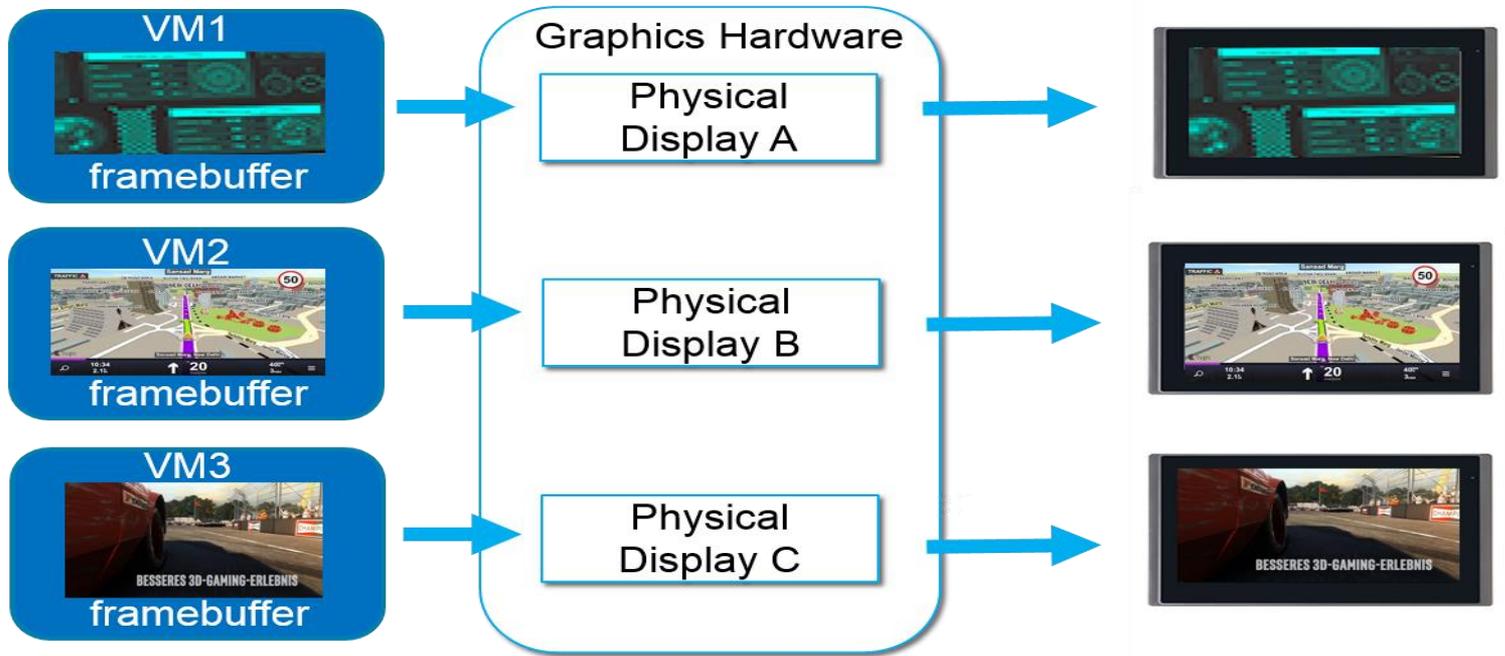


# A KEY MDEV Usage Requirement in GPU Virtualization Is To Render Guest Framebuffer In The Host Side

# Scenario-1: Remote Virtual Desktop

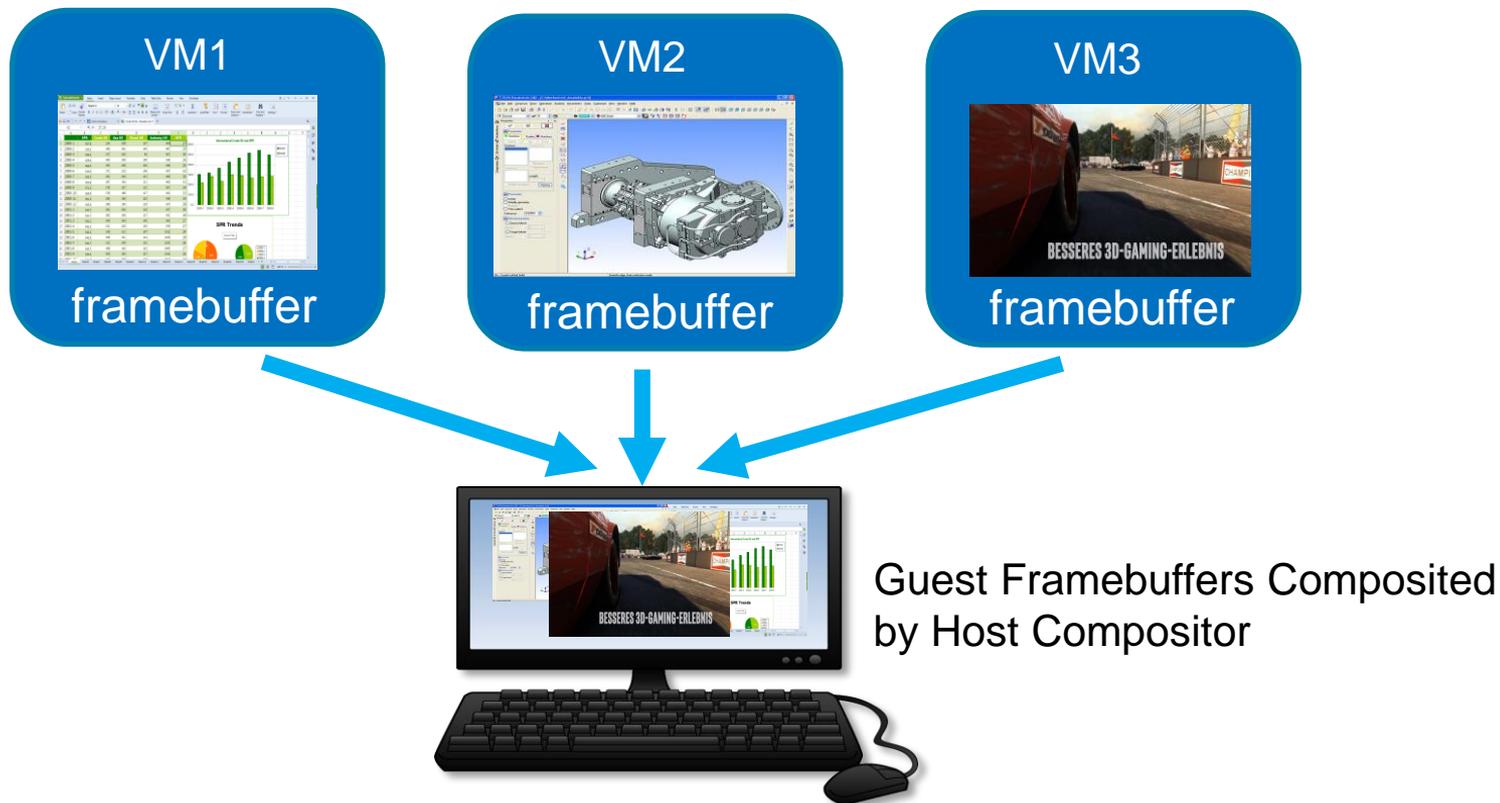


# Scenario-2: Automotive Virtualization



GPU Mediator Directly Configure VM Framebuffer to Display Panel

# Scenario-3: Rich Virtual Client

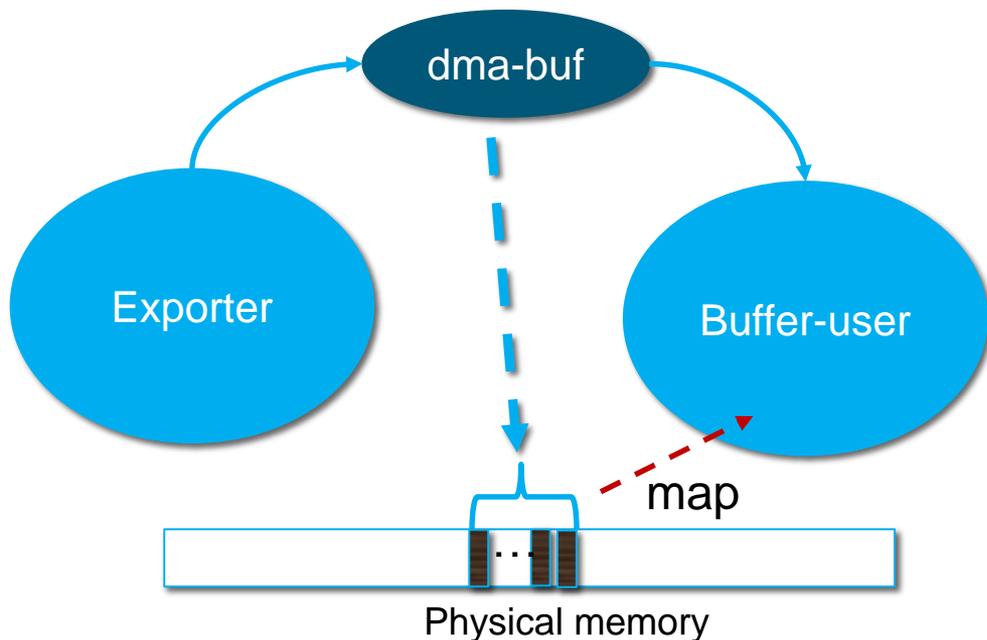


# Generic Buffer Sharing in MDEV Framework

# Design Philosophy - Generic

- Based on dma-buf subsystem which
  - Generic way provided by kernel to share DMA buffers
  - Already been supported by many device drivers
  - Supported by Linux graphics stacks
  - Used by remote protocol (e.g. SPICE)
- ABIs proposed by different MDEVs in the same category

# A Short Note On Dma-buf



## The dma-buf

- Represents a `sg_table` (SGT)

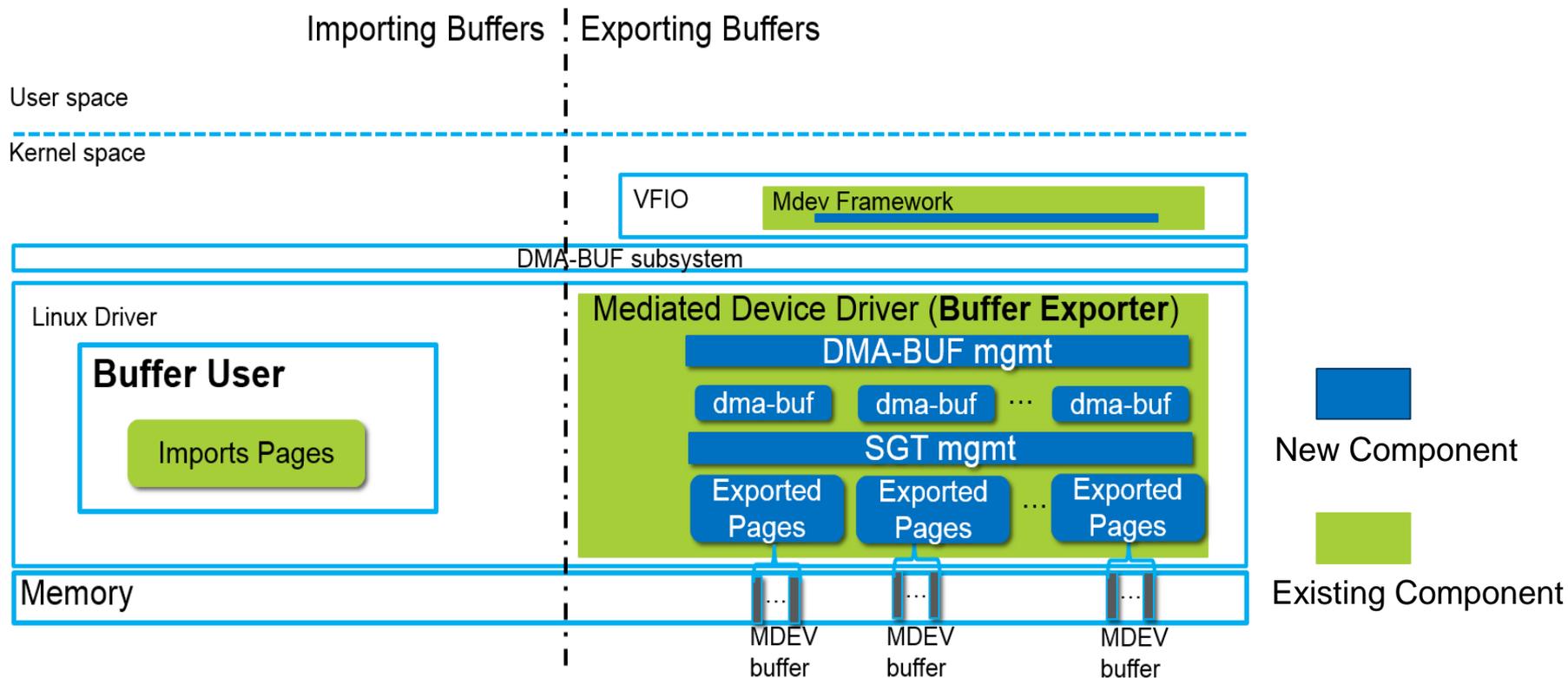
## The exporter

- Producer of the dma-buf
- Implements `dma_buf_ops`
- Exports dma-buf as a file descriptor

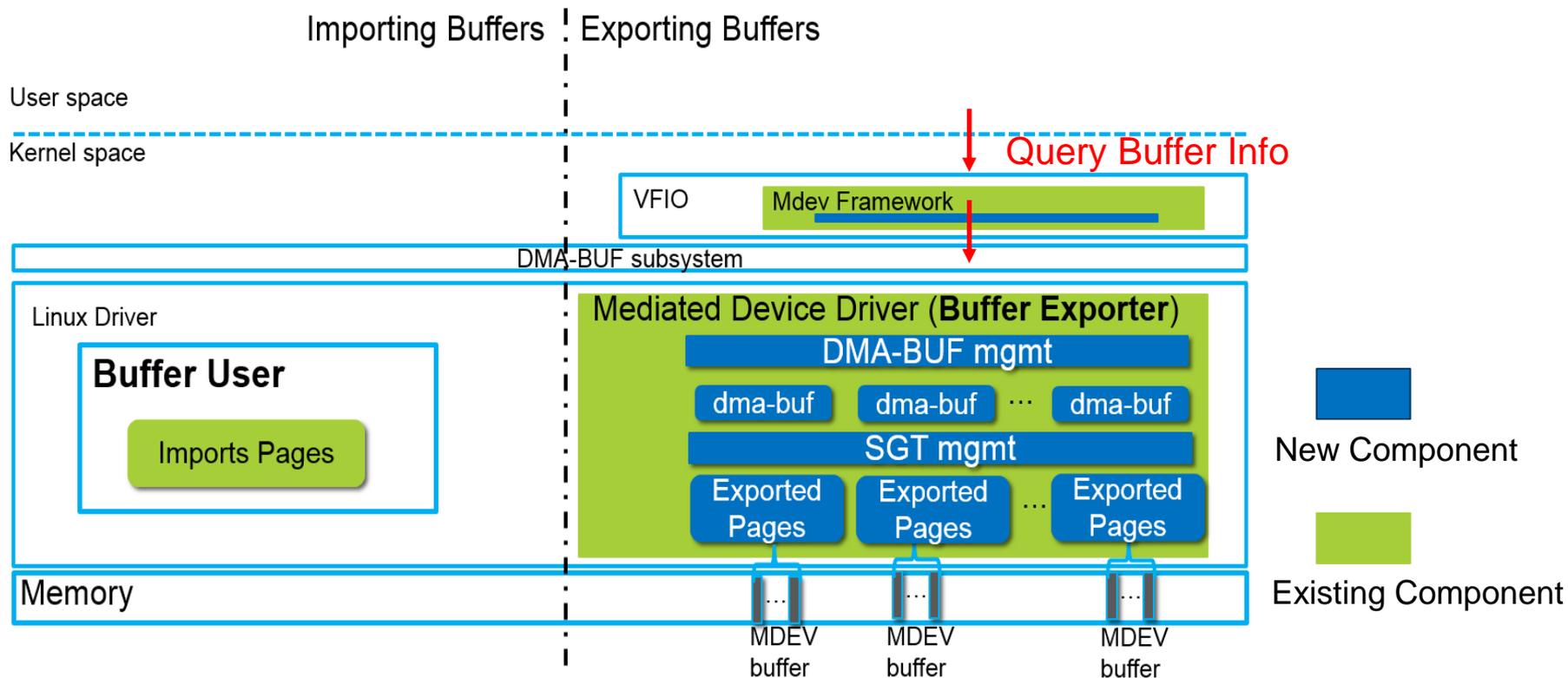
## The buffer-user

- Imports the dma-buf
- Gets the scatterlist in SGT
- Maps buffer address

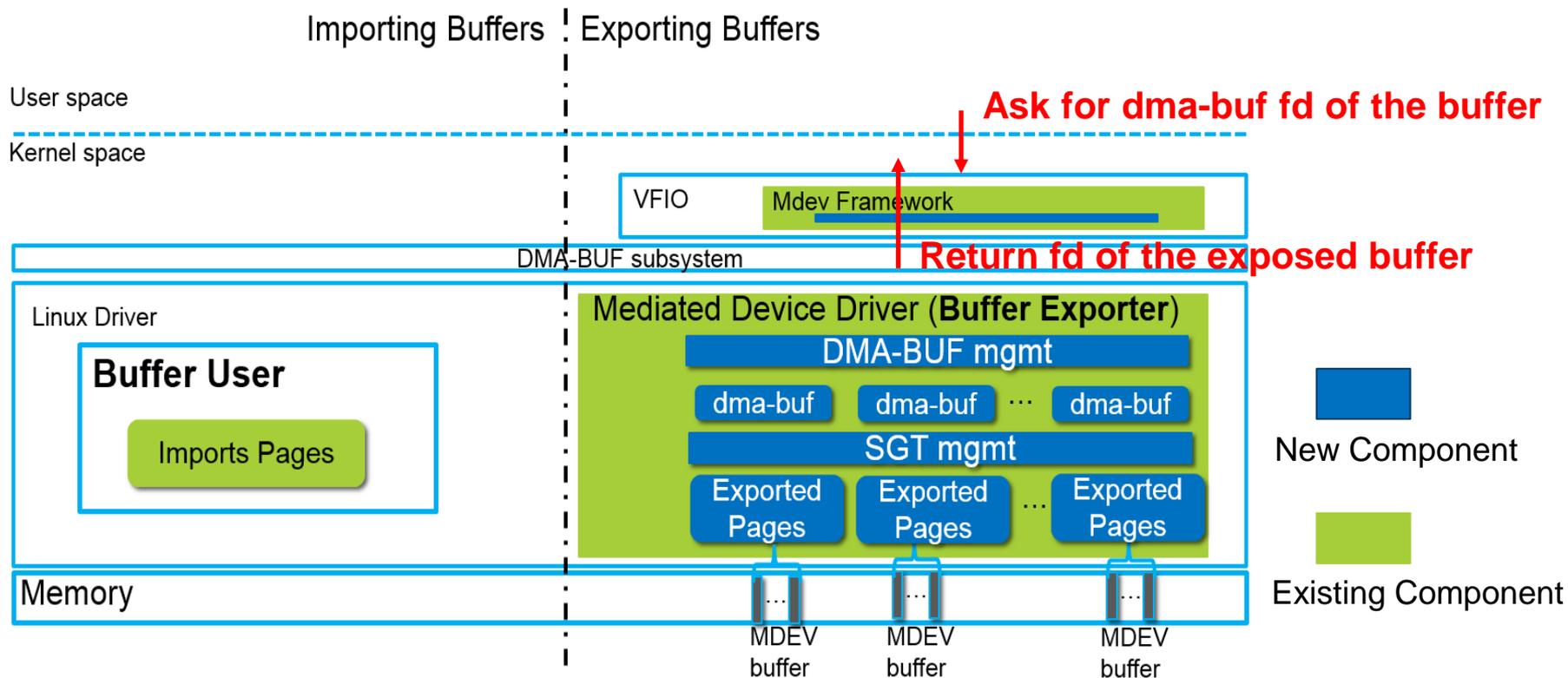
# Buffer Sharing in MDEV Framework



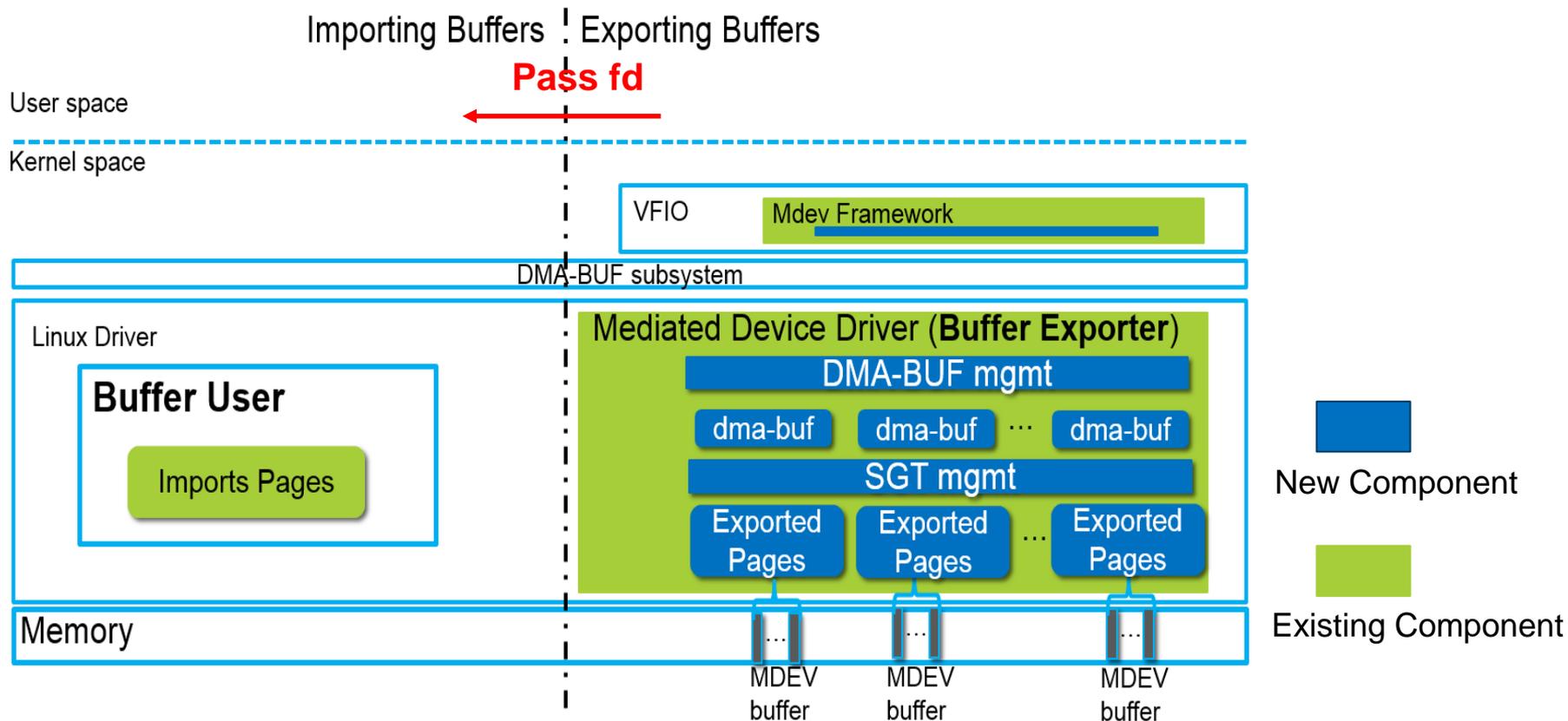
# Buffer Sharing in MDEV Framework



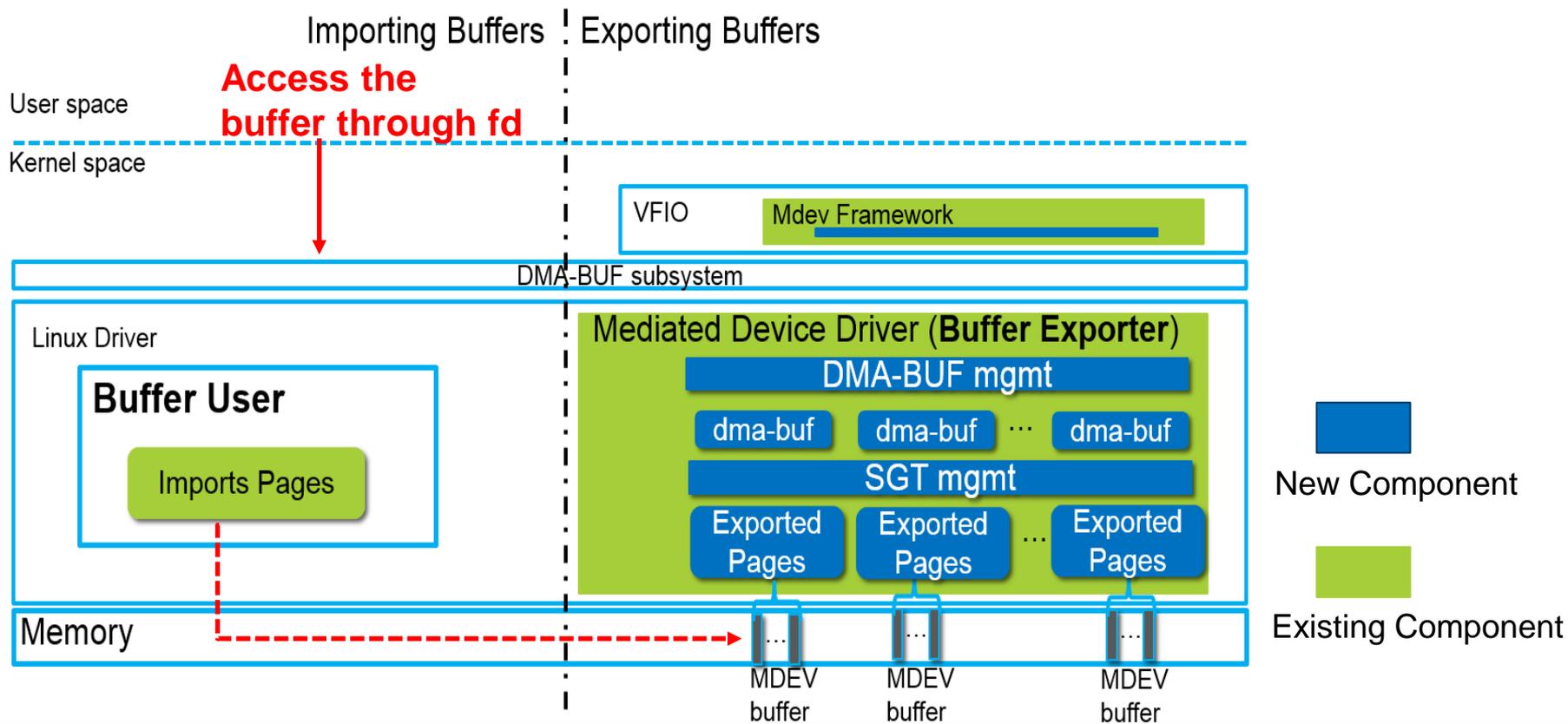
# Buffer Sharing in MDEV Framework



# Buffer Sharing in MDEV Framework



# Buffer Sharing in MDEV Framework



# Shared Guest Framebuffer Based on DMA-BUF(1/2)

## VFIO\_DEVICE\_QUERY\_GFX\_PLANE

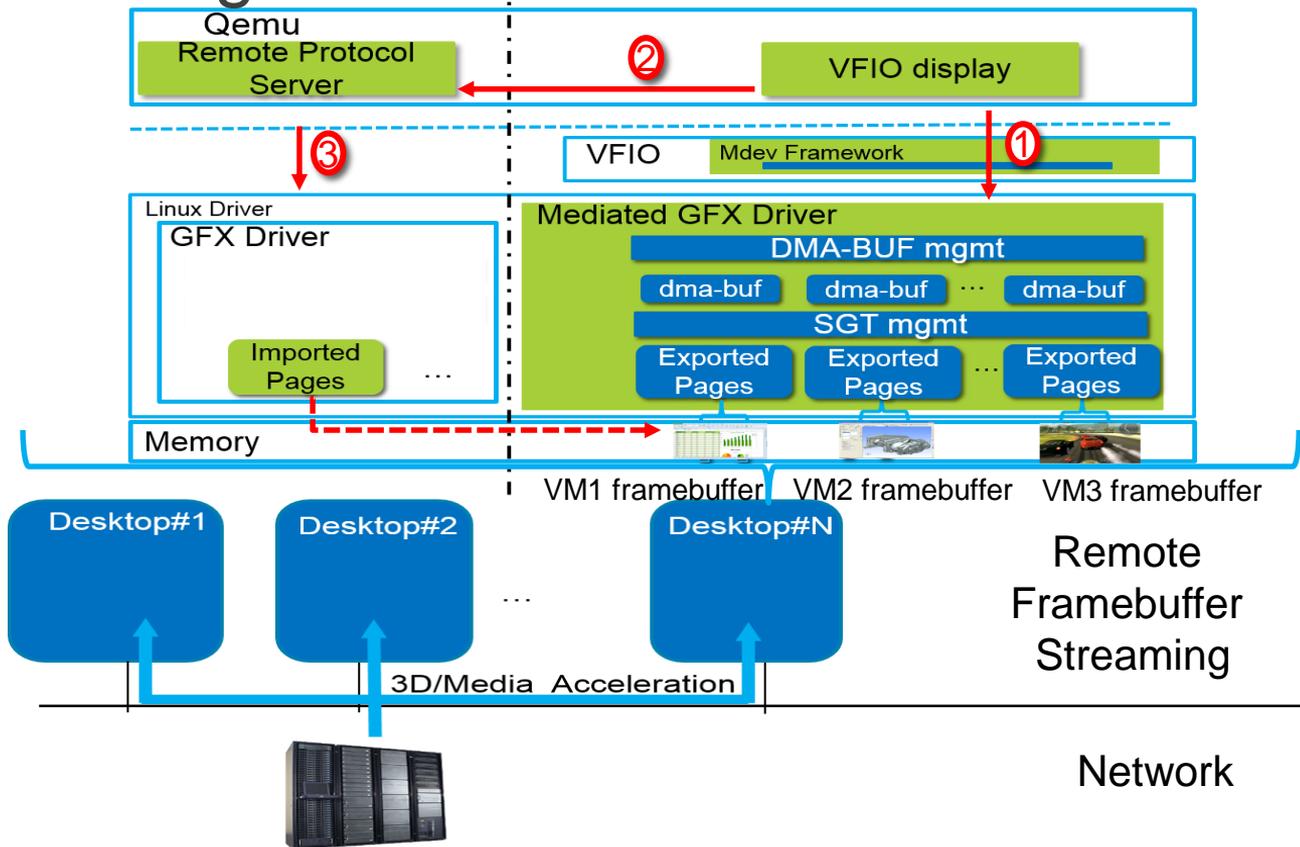
```
struct vfio_device_gfx_plane_info {  
    /* in */  
    __u32 drm_plane_type; /* type of plane: DRM_PLANE_TYPE_* */  
    /* out */  
    __u32 width; /* width of plane */  
    __u32 height; /* height of plane */  
    __u32 stride; /* stride of plane */  
    __u32 size; /* size of plane in bytes, align on page*/  
    ...  
    __s32 dmabuf_id; /* dma-buf id */  
};
```

# Shared Guest Framebuffer Based on DMA-BUF(2/2)

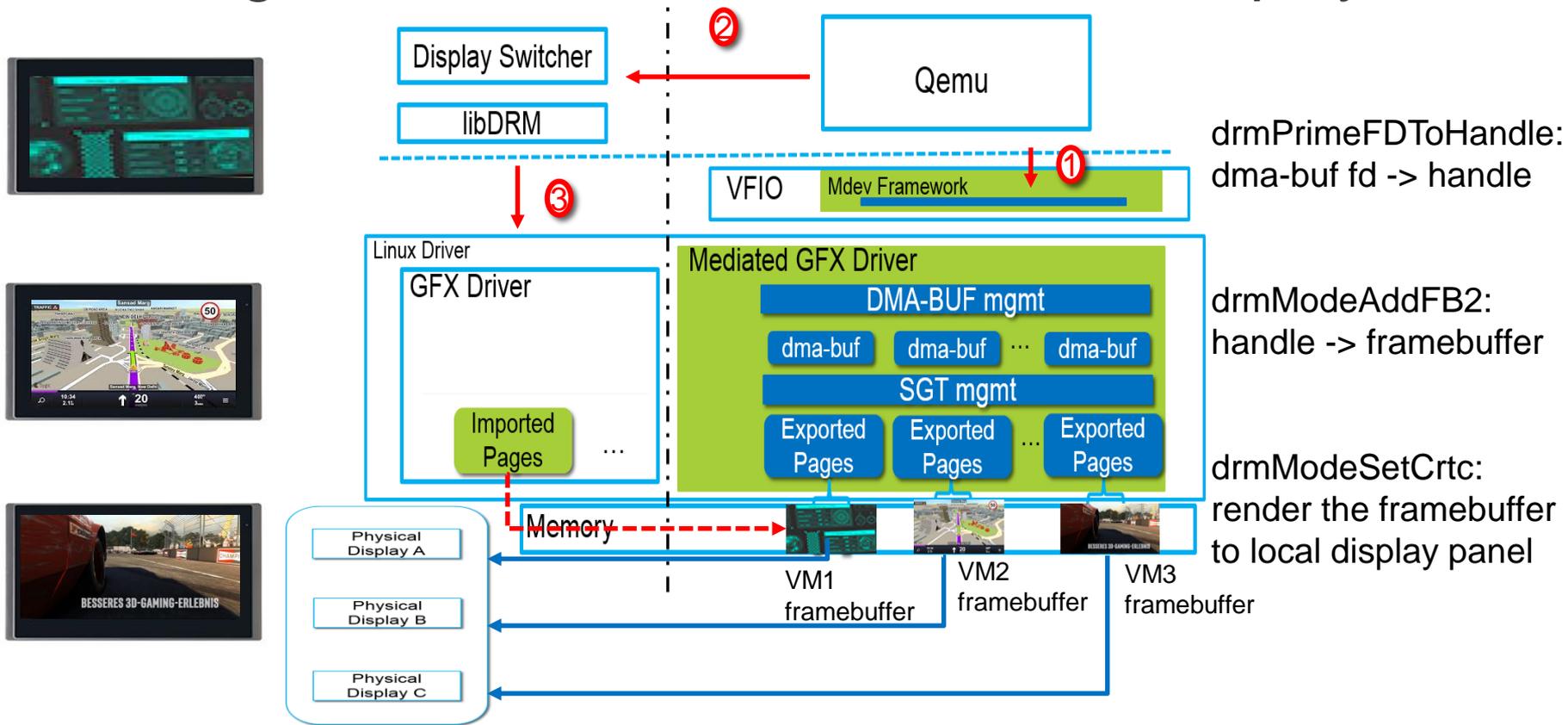
## VFIO\_DEVICE\_GET\_GFX\_DMABUF

```
struct vfio_device_gfx_dmabuf_fd {  
    __u32 argsz;  
    __u32 flags;  
    /* in */  
    __u32 dmabuf_id;  
    /* out */  
    __s32 dmabuf_fd;  
};
```

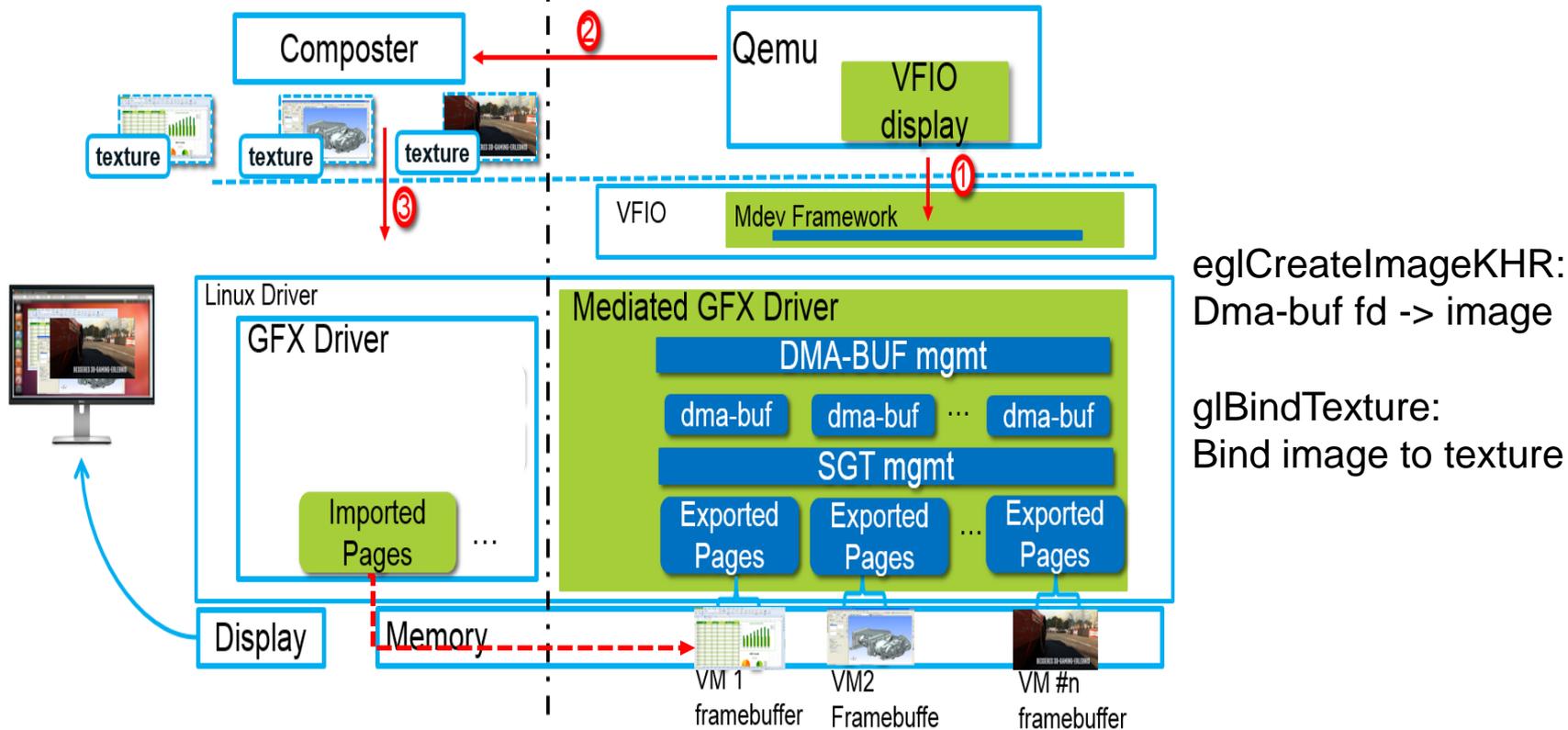
# Sharing vGPU Framebuffer For Remote Desktop



# Sharing vGPU Framebuffer For Direct Display



# Sharing vGPU Framebuffer For Composited Display



# Implementation Details

## DMA-BUF Management

- Implements dma-buf operations
- Creates dma-buf object with an installed fd
- Caches the exposed DMA-BUF object

## SGT Management

- Allocates/Releases the SGT
- Gets address of the mediated device buffer

## Guest buffer Information Collection

- Gets the info by decoding values in the mediated device registers

# Status

## **PATCH v15:**

- Tested with Linux kernel 4.13
- Working with community for upstreaming

## **Userspace:**

- Qemu UI patch-set of “start adding dma-buf support” designed by Gerd Hoffman
- Qemu UI RFC patch-set of “add display support” designed by Gerd Hoffman

# Summary

- The design is generic enough for I/O mediators.
  - Based on DMA-BUF subsystem
- The implementation is generic for mediated vGPUs.
- Welcome innovations based on this generic buffer sharing mechanism



<https://github.com/01org/gvt-linux>