Xen Based GPU Virtualization - VirtIO/Passthrough

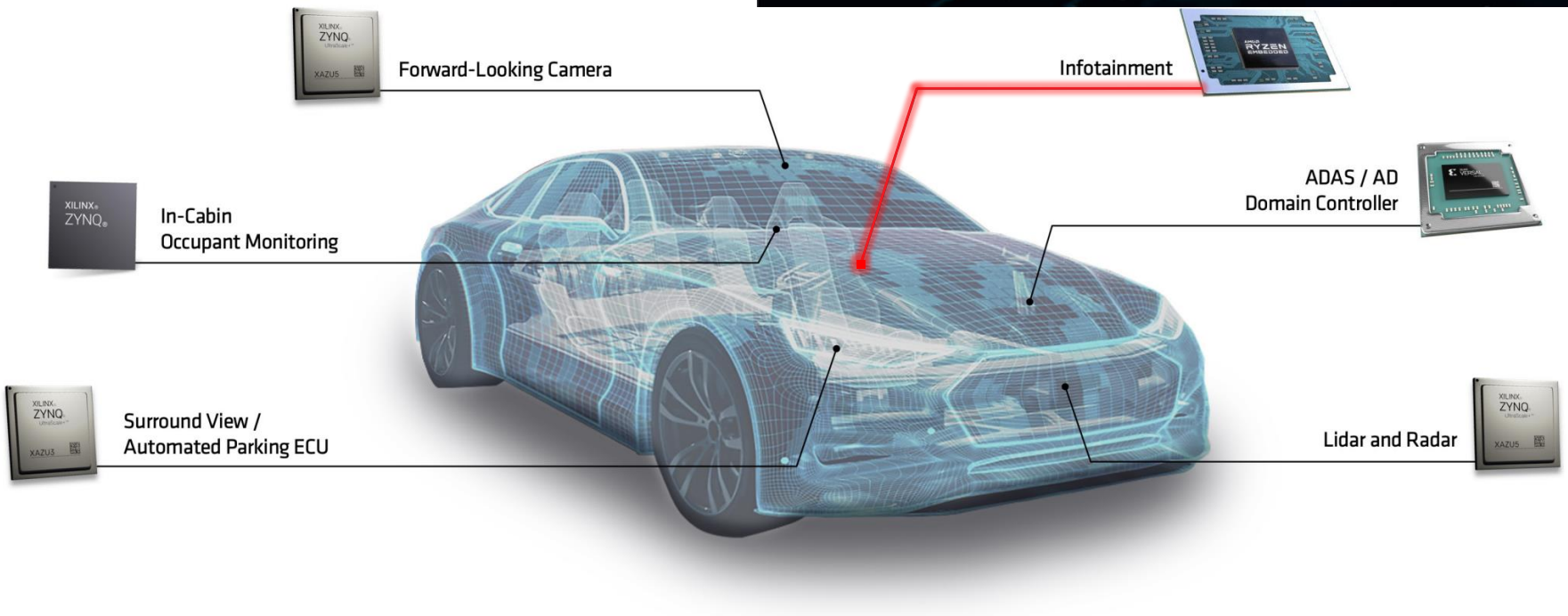
Huang Rui <ray.huang@amd.com>

Who are we?

- Ray Huang - China 
  - Kernel (GPU, Xen)/QEMU/Xen
- Julia Zhang - China 
  - Mesa 3D (OpenGL, Vulkan)/Virglrenderer/QEMU
- Honglei Huang - China 
  - Mesa Multimedia/Virglrenderer/QEMU/ROCm
- Jiqian Chen - China 
  - Kernel (GPU, Xen)/QEMU/Xen
- Pierre-Eric Pelloux-Prayer - France 
  - Mesa 3D (OpenGL, Vulkan)/Virglrenderer/QEMU
- Xenia Ragiadakou - Greece 
  - Xen/QEMU/Kernel (Xen)
- Leo Liu - Canada 
  - Mesa Multimedia/Virglrenderer
- Boyuan Zhang - Canada 
  - Mesa Multimedia/Virglrenderer
- Robert Beckett - United Kingdom 
  - Mesa 3D (OpenGL, Vulkan)/Virglrenderer/QEMU

Intention

- Automotive Infotainment on AMD
  - Multiple Systems (Guest VM) in one Car
  - Xen based GPU Virtualization
  - 3D Graphic and Multimedia hardware acceleration



Technical Background

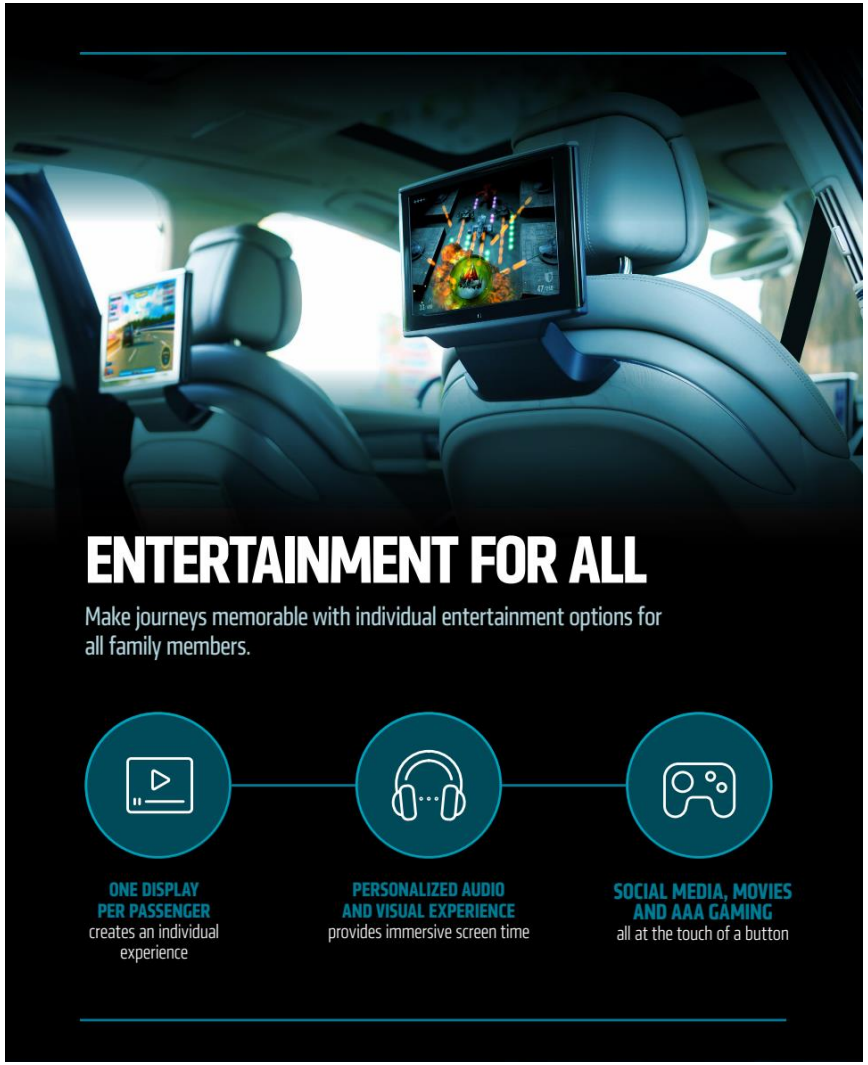
- VirtIO GPU - Only Virgl
  - Virgl is a 3D OpenGL implementation for VirtIO GPU
  - Not fully verified on Xen
  - No Vulkan support

- DOM0 PV on x86 CPU
  - PV is using the swiotlb for DMA operations on all PCI devices
  - However, in AMD GPU use cases, there are always many large sized buffer mappings as large as Mbytes or Gbytes on the device. Swiotlb is not usable for GPU

- PCI Passthrough on PV dom0
  - GPU passthrough is a very popular use case for virtualization
  - But it only can be mapped into one guest DOMU
  - Passthrough is only supported under PV dom0

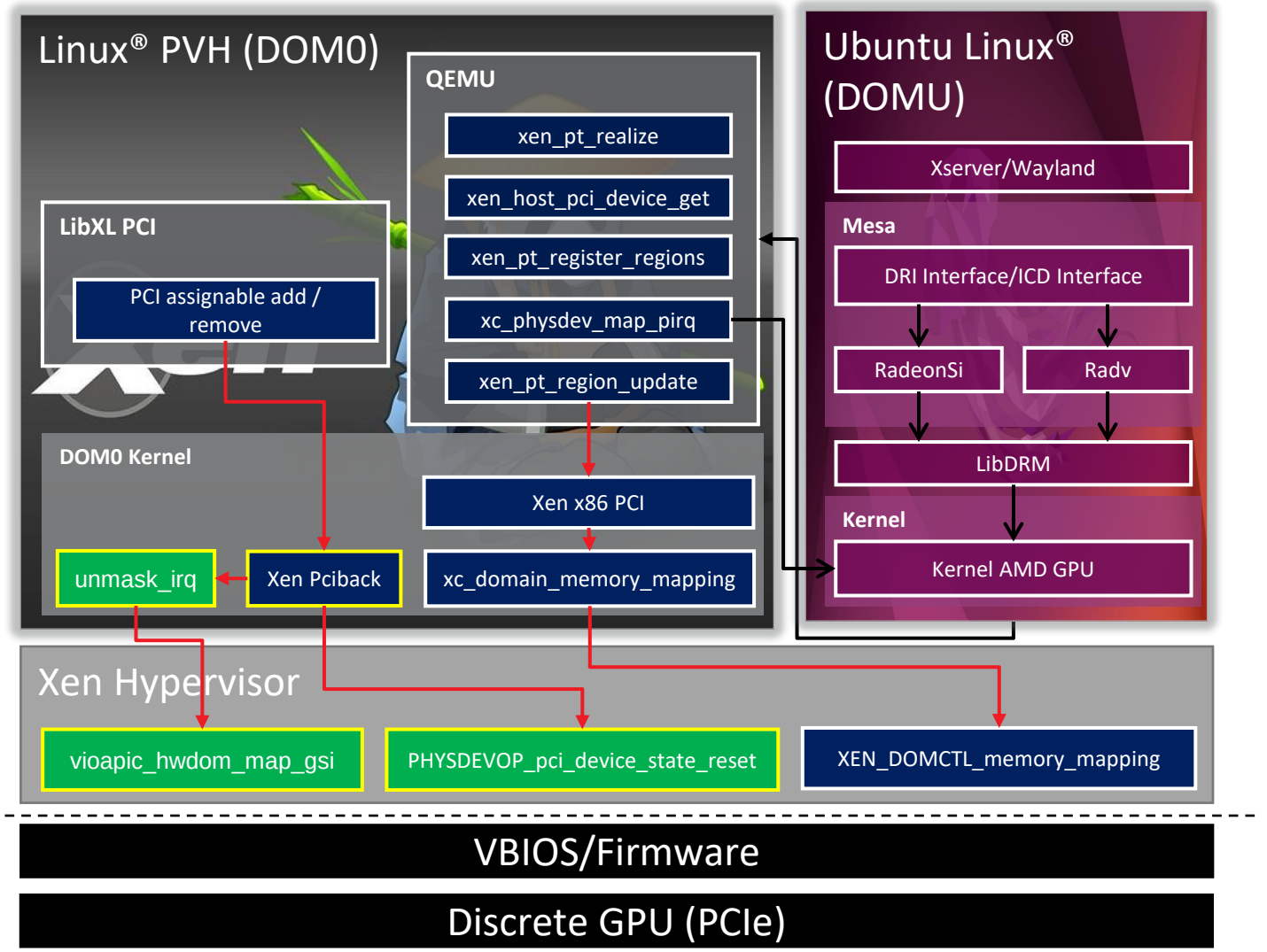
New Proposal to improve GPU virtualization on Xen

- Graphic stack including OpenGL and Vulkan
  - Improve OpenGL support in guest domU
  - Add Vulkan support in guest domU
  - Steam games and graphic benchmark coverage
- Xen PVH dom0 for AMDGPU on x86 CPU
  - Support native AMDGPU kernel on dom0
  - Support PCIe Passthrough for AMDGPU on PVH dom0
- Multiple types of GPUs support in one guest DOMU
  - VirtIO GPU and Passthrough GPU transaction together
  - Multiple VirtIO GPUs together



GPU Passthrough on Xen PVH DOM0

- **Kernel**
  - Unmask gsi **(New for PVH dom0)**
    - io_apic_write - trap into Xen hypervisor
  - Reset Device's state **(New for PVH dom0)**
    - PHYSDEVOP_pci_device_state_reset
- **QEMU**
  - Config initialization
    - Get real device base info
    - Initialize emulated registers
  - Map gsi to pirq
    - Translate irq to gsi(**New for PVH dom0**)
    - Use gsi to do PHYSDEVOP_map_pirq
  - IO Port/Memory Mapping
    - XEN_DOMCTL_ioport_mapping and XEN_DOMCTL_memory_mapping



Blob Memory Introduction

- Blob Memory Usage
  - VirtIO GPU Driver: Virgl / Venus
  - AMD Native Driver: RadeonSi / Radv
- QEMU on Xen mem-path (No udma-buf on Xen)
  - Remove udma-buf to support Xen
- Virtual GPU local memory - Blob Memory
  - Add local memory in virtual MMIO bar
  - Provide the direct memory access like VRAM for virtual GPU
  - QEMU/Virglrenderer responds the virtio command to create and map the blob memory from guest virtual MMIO bar

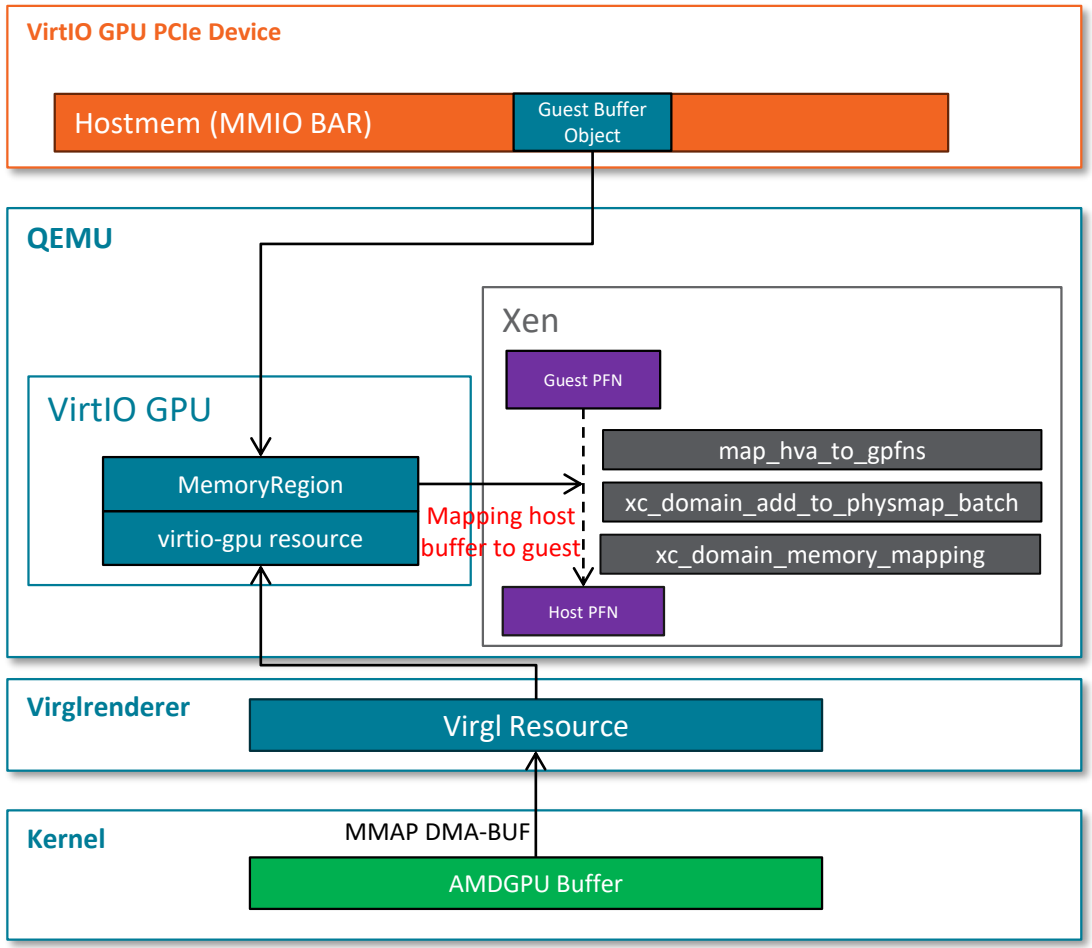
```
device_model_args_hvm = ["-display", "sdl,gl=on", "-device", "virtio-vga gl,context_init=true,blob=true,hostmem=4G"]
```

```
[   19.211594] [drm] Host memory window: 0x200000000 +0x100000000
[   19.211706] [drm] features: +virgl -edid +resource_blob +host_visible
[   19.211707] [drm] features: +context_init
[   19.228967] [drm] number of scanouts: 1
[   19.229092] [drm] number of cap sets: 3
```

```
00:04.0 VGA compatible controller: Red Hat, Inc. Virtio GPU (rev 01) (prog-if 00 [VGA controller])
        Subsystem: Red Hat, Inc. Virtio GPU
        Physical Slot: 4
        Flags: bus master, fast devsel, latency 0, IRQ 32
        Memory at f1000000 (32-bit, prefetchable) [size=8M]
        Memory at f1b74000 (32-bit, non-prefetchable) [size=4K]
        Memory at f1b70000 (64-bit, prefetchable) [size=16K]
        Memory at 200000000 (64-bit, prefetchable) [size=4G]
        Expansion ROM at 000c0000 [disabled] [size=128K]
        Capabilities: <access denied>
        Kernel driver in use: virtio-pci
```

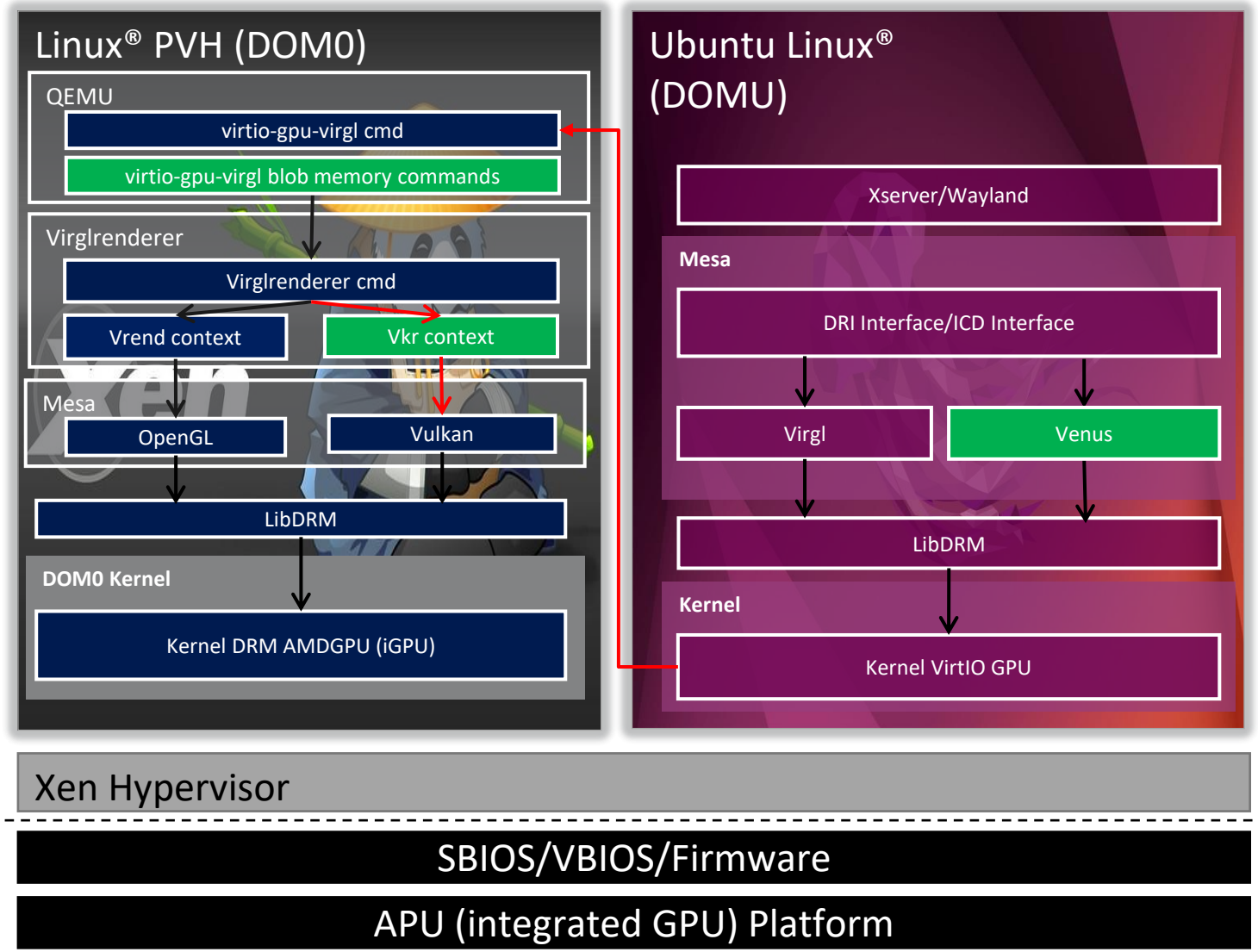
Blob Memory to Local GPU buffer on Xen

- Guest Mesa (Virgl/Venus or Native Context)
  - Be able to use blob memory for direct access in the guest instead of command transfer
- Guest Kernel
  - Expose a MMIO bar in PCI config space for blob memory. (hostmem)
- Blob Memory Commands in QEMU
  - Provide blob memory commands implementation
- Virglrenderer
  - Get the host virtual address of local GPU buffer which exposed by DMA-BUF
- Xen
  - Convert the host virtual address to host physical address to connect it with guest physical address with hypervisor calls
  - Only can be used for pining buffer (no eviction) - challenge



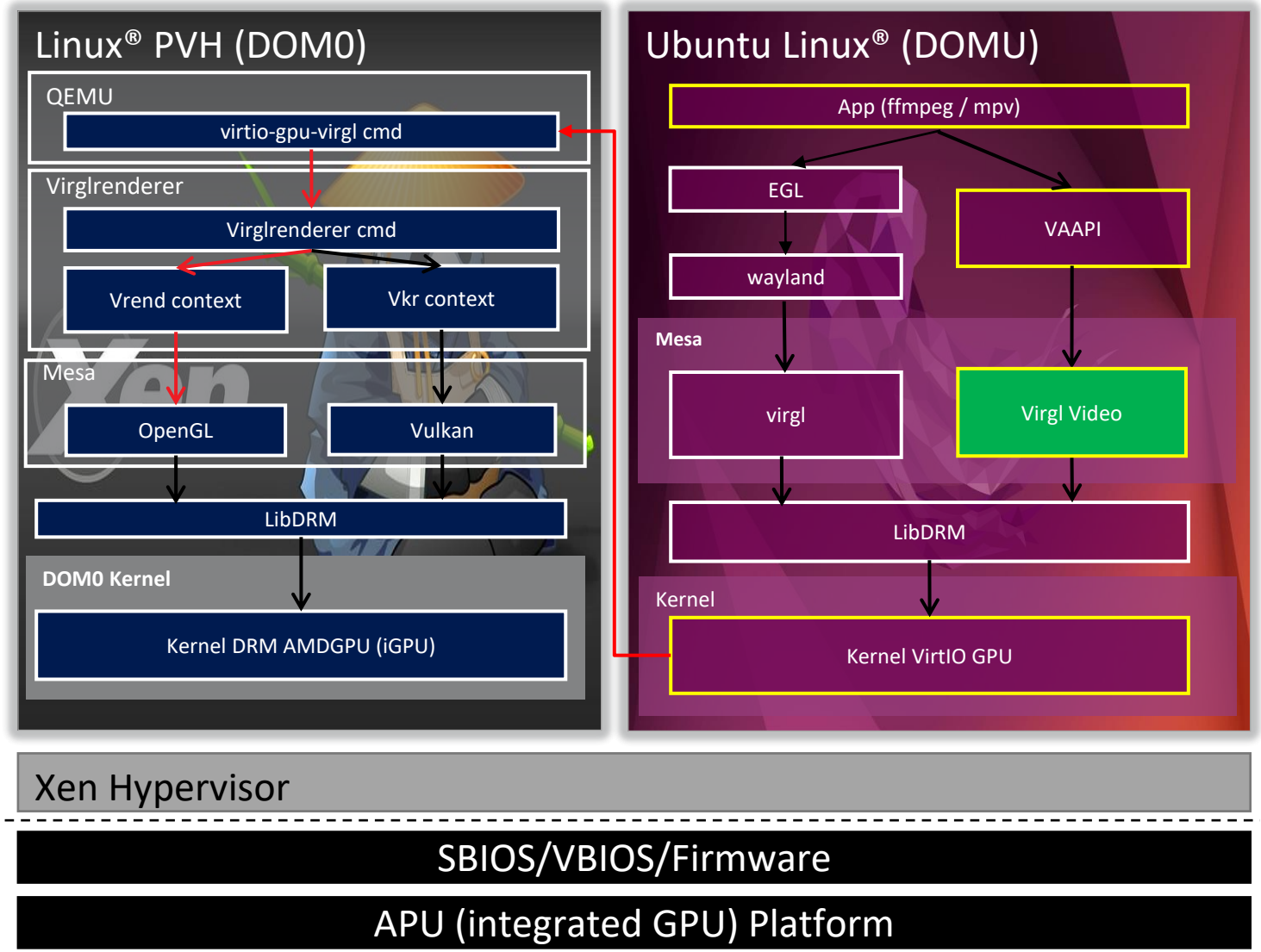
Virgl and Venus

- Bring up Venus Support on Xen
  - No Vulcan is functional on Xen guest based on VirtIO GPU
  - Implement blob memory support with virtio-gpu-virgl in QEMU
  - Implement the use case to import a blob resource of external vulkan with OpenGL while egl is not initialized
- Virglrenderer
  - https://gitlab.freedesktop.org/virgl/virglrenderer/-/merge_requests/1068
- Mesa
  - https://gitlab.freedesktop.org/mesa/mesa/-/merge_requests/23680
- QEMU - Blob Memory
  - https://lore.kernel.org/qemu-devel/20230915111130.24064-1-ray.huang@amd.com



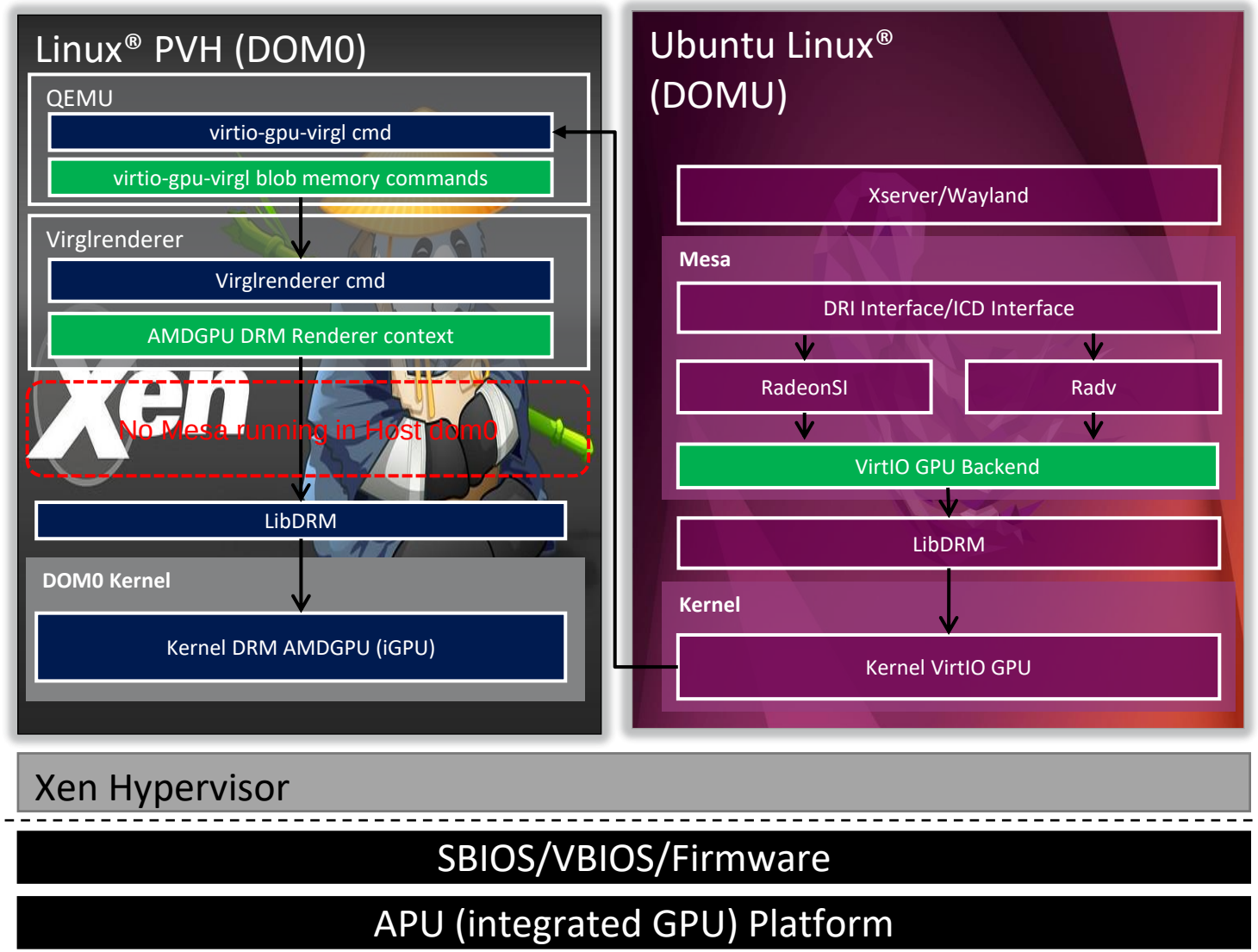
Virgl Video

- Introduce Virgl Video on Xen
  - Support multiple video codec
    - Mpeg12/VC1/JEPG/HEVC10bit/VP9
- Virgl video corresponding behavior of VM VAAPI in host
  - Initial VAAPI when the VM starts.
  - Create Vasurface
  - Create a true config and context for this decoding in host
  - Start the actual decoding process in host while VM calls renderPictures()
- Mesa/Virglrenderer
  - https://gitlab.freedesktop.org/mesa/mesa/-/merge_requests/22108
  - https://gitlab.freedesktop.org/virgl/virglrenderer/-/merge_requests/1068

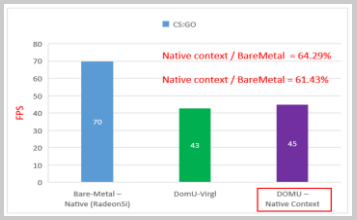
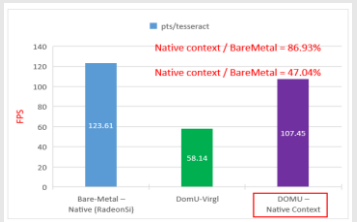
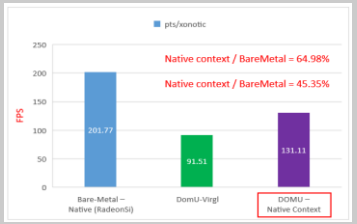
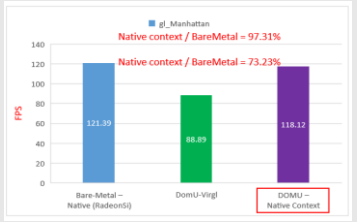


VirtIO Native Context

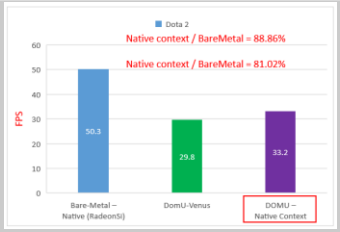
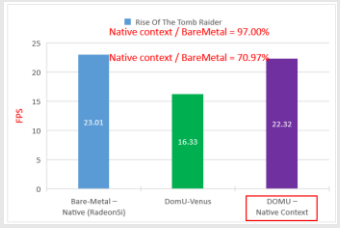
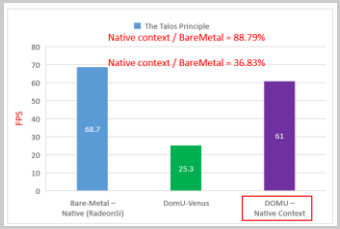
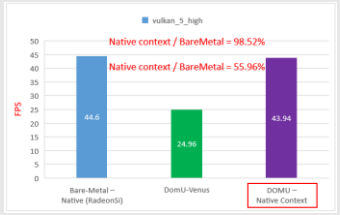
- VirtIO Native Context with AMDGPU on Xen - Coming
  - One more straightforward way than Virgl/Venus
  - API Forward with Libdrm instead of OpenGL/Vulkan interfaces
  - Leverage blob memory with native OpenGL/Vulkan in the guest
  - No Mesa running in the host (faster)
- Initial the design bases on the work of Bob Clark and refer the corresponding presentation on XDC2022
  - https://gitlab.freedesktop.org/mesa/mesa/-/merge_requests/21658



OpenGL Comparison between Virgl and Native Context

Test Cases (Unit: FPS)	Histogram	Bare-Metal – Native (RadeonSi)	DOMU – Virgl	DOMU – Native Context (RadeonSi)
CS:GO		70	43 (61.43%)	45 (64.29)
pts/tesseract		123.61	58.14 (47.04%)	107.45 (86.93%)
pts/xonotic		201.77	91.51 (45.35%)	131.11 (64.98)
GFXBench (gl_Manhattan)		121.39	88.89 (73.23%)	118.12 (97.31%)

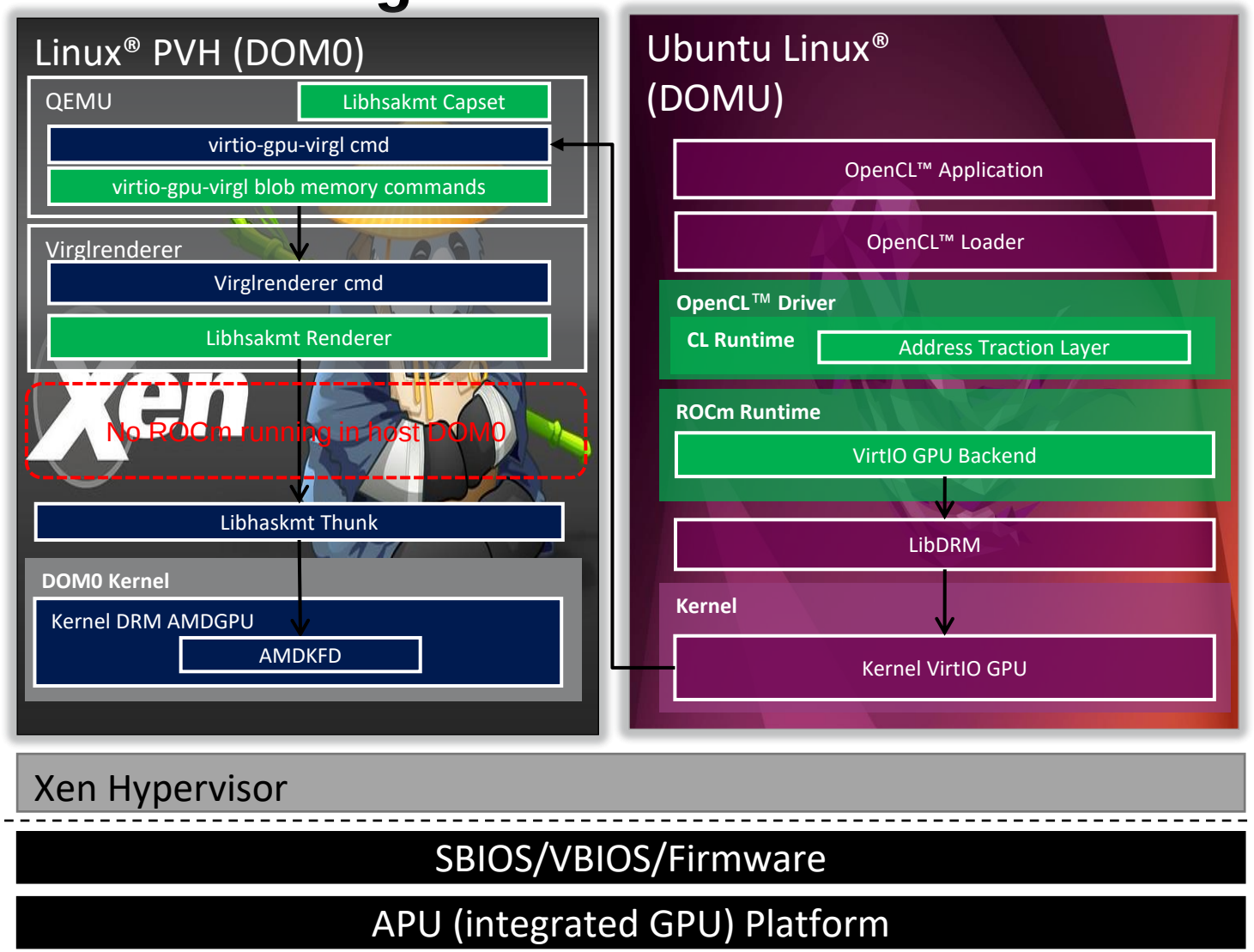
Vulkan Comparison between Venus and Native Context

Test Cases(Unit: FPS)	Histogram	Bare-Metal – Native (Radv)	DOMU – Venus	DOMU – Native Context (Radv)
Dota 2		50.3	29.8 (81.02%)	33.2 (88.86%)
Rise Of The Tomb Raider		23.01	16.33 (70.97%)	22.32 (97.00%)
The Talos Principle		68.7	25.3 (36.83%)	61.0 (88.79%)
GFXBench (vulkan_5_high)		44.60	24.96 (55.96%)	43.94 (98.52%)

ROCm with VirtIO Native Context - Design Preview

- AMD ROCm on guest DOMU - Coming
  - Support OpenCL™ over ROCm for virtualization
  - Also inspired by virtio native context on graphic design

- API Forward for Libhsakmt (Thunk)
  - Introduce VirtIO GPU backend in ROCm runtime and OpenCL™ runtime
  - Add libhsakmt capacity in QEMU
  - Leverage blob memory
  - Add libhsakmt renderer in virglrenderer



The Best is Yet to Come

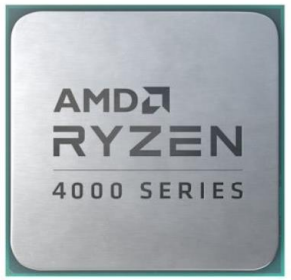
- Continue upstream for the whole solution - On going
- Full graphic benchmark verification
- Resolve the direct mapping of virtual MMIO bar for QEMU on Xen
  - Plan to use an MMU notifier
- **Implement the approach to enable AMD ROCm compute stack at guest VM with VirtIO**
  - Support leading AI Frameworks over ROCm for virtualization

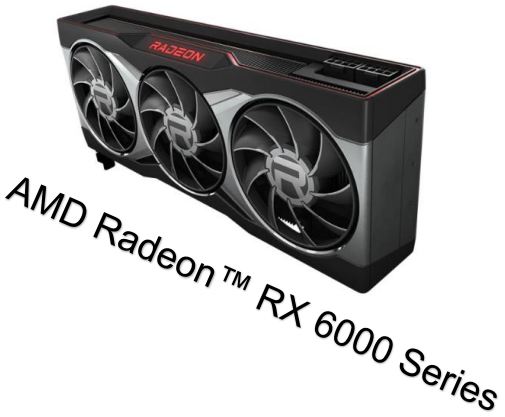
AMD ROCm

TensorFlow

PyTorch

References





- Hardware
  - AMD Ryzen 4000G Series APUs
  - AMD Radeon™ RX 6000 Series GPUs
- Xen Project Developer and Design Summit 2023
  - https://xen2023.sched.com/event/1LKln
- Repositories
  - Kernel - https://git.kernel.org/pub/scm/linux/kernel/git/rui/linux.git/log/?h=upstream-for-xen-v2
  - Xen - https://gitlab.freedesktop.org/rui/xen/-/tree/upstream-for-xen-v2
  - QEMU - https://gitlab.freedesktop.org/rui/qemu-xen/-/tree/upstream-for-xen-v2
  - Virglrenderer - https://gitlab.freedesktop.org/rui/virglrenderer/-/tree/upstream-for-xen-v2
  - Mesa - https://gitlab.freedesktop.org/rui/mesa/-/tree/upstream-for-xen-v2
- Upstream is in progress
  - Kernel - https://lore.kernel.org/lkml/20230312120157.452859-1-ray.huang@amd.com/
  - Xen - https://lore.kernel.org/xen-devel/20230312075455.450187-1-ray.huang@amd.com/
  - QEMU - https://lore.kernel.org/qemu-devel/20230915111130.24064-1-ray.huang@amd.com/
  - Virglrenderer - https://gitlab.freedesktop.org/virgl/virglrenderer/-/merge_requests/1068
  - Mesa (MM) - https://gitlab.freedesktop.org/mesa/mesa/-/merge_requests/22108
  - Mesa (Native Context) - https://gitlab.freedesktop.org/mesa/mesa/-/merge_requests/21658
  - Mesa (Venus) - https://gitlab.freedesktop.org/mesa/mesa/-/merge_requests/23680

Demo, Q&A, and Thank You

Disclaimer:

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS." AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.


AMD