Linaro Connect
MADRID 2024 | MAY 14-17 2024

Orko: where are we now?

Alex Bennée, Virtualisation and Emulation Tech Lead

Outline

- Project Orko review
- Multimedia Devices and VirtIO
- Hardware Challenges
- Demo overview

Project Orko

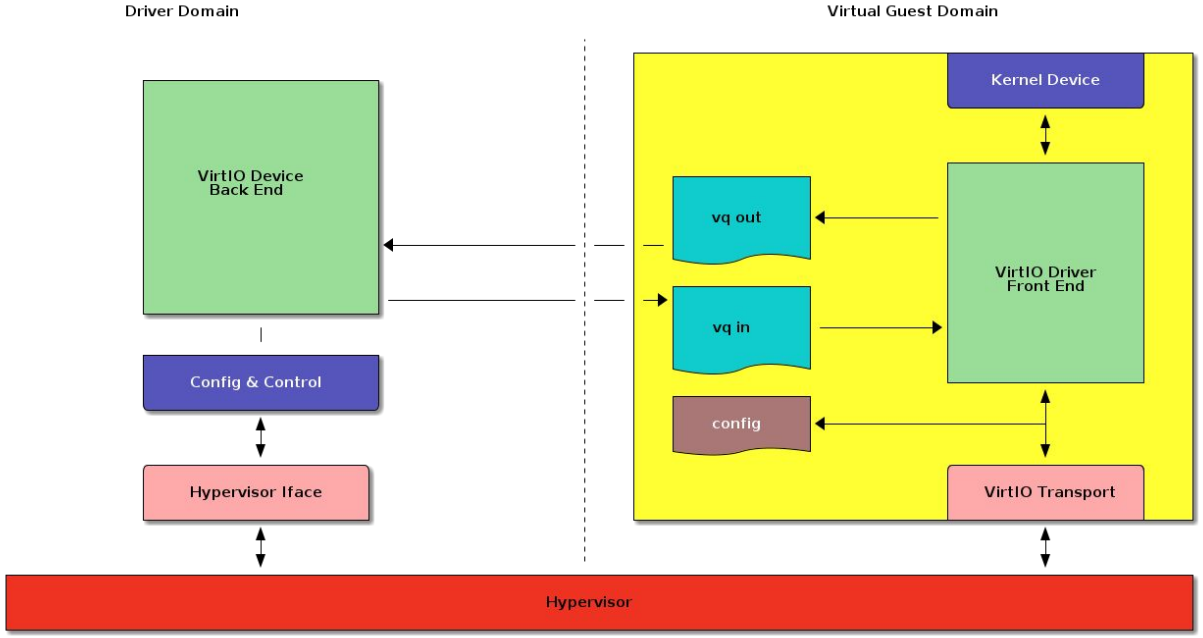
Project Orko Aims

"To provide safe and efficient VirtIO devices for hypervisor agnostic cloud native workloads"

- Me, just now

VirtIO (@ OASIS)

- Para-virtualised hypervisor aware devices
- Minimise expensive guest exits

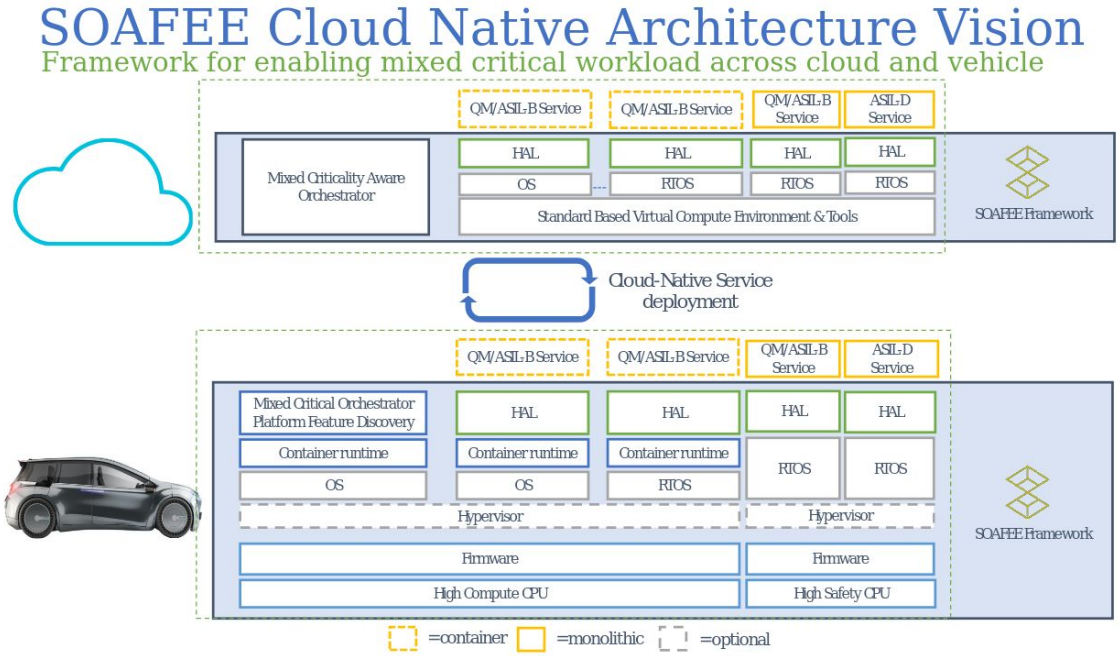


Hypervisor Agnostic

- Originally a Linux/KVM creation
- Backends often tied to host/hypervisor
  - e.g. vhost-kernel tied to guts of Linux
- Orko leverages vhost-user
  - Backends in host userspace
  - Usable by multiple VMMs (QEMU, crosvm, standalone)
  - Hypervisor differences abstracted by libraries
  - Demonstrated on Xen

Cloud Native

- Develop and test in the Cloud, deploy on the Edge
- Strong Abstractions Needed



Source: soafee.io

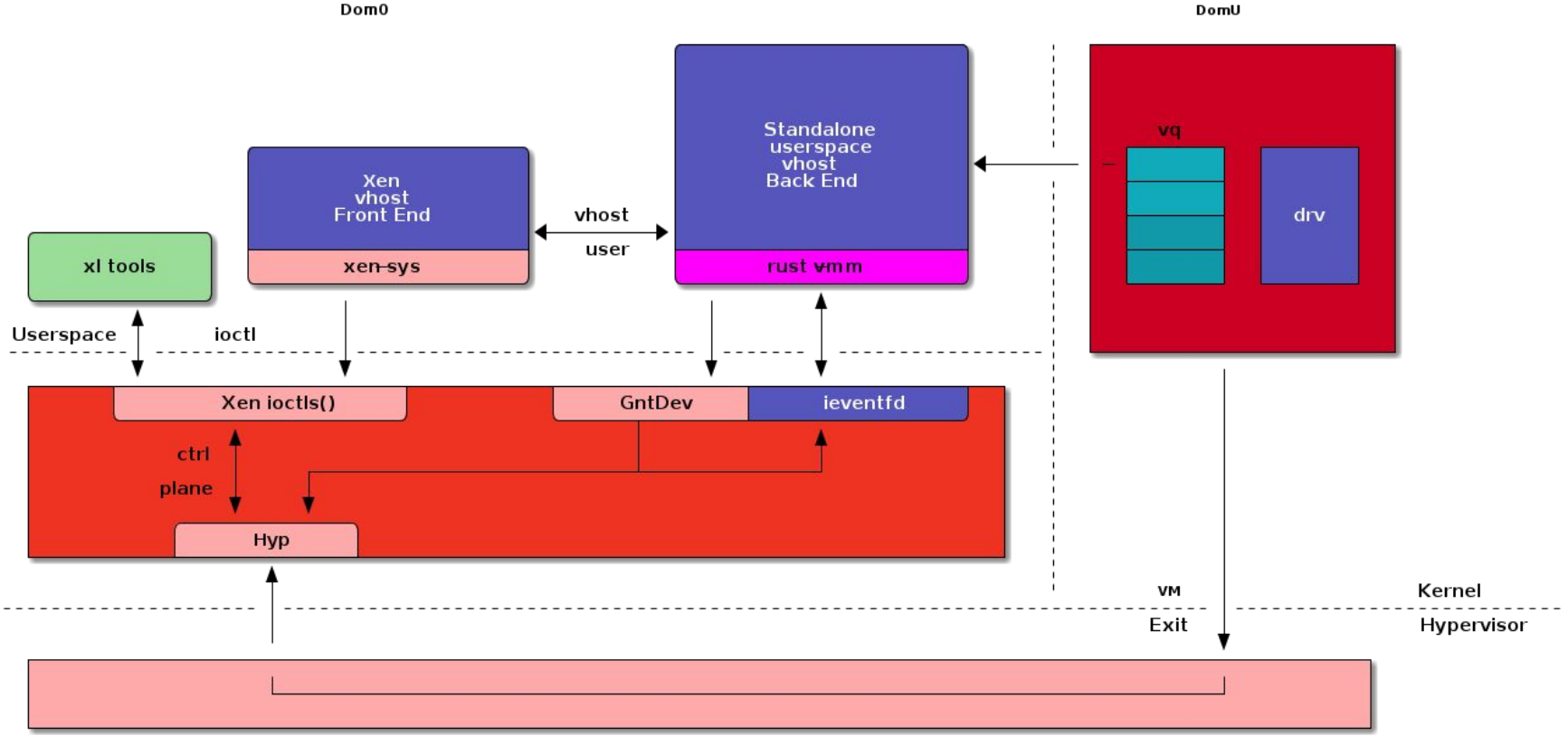
Safe and Efficient

- Original VirtIO design
  - Untrusted guests, potential for memory safety errors
  - Trusted device backend (“sees all”)
- Orko backends
  - Implemented in rust under rust-vmm project
  - Stricter memory model (“host sees what it needs”)

rust-vmm and vhost-device

- Created in December 2018
  - Leverages work from CrosVM and Firecracker
    - Amazon, Google, Intel, Red Hat and others
  - Components for building VMMs
    - vmm-reference
    - Cloud Hypervisor
- vhost-device, vhost-user backends
  - Production: gpio, i2c, input, rng, scmi, scsi, sound, vsock
  - Staging: video (awaiting standardisation)
  - PRs: console, can, spi
  - Maintainers from Linaro and Red Hat

Orko Architecture

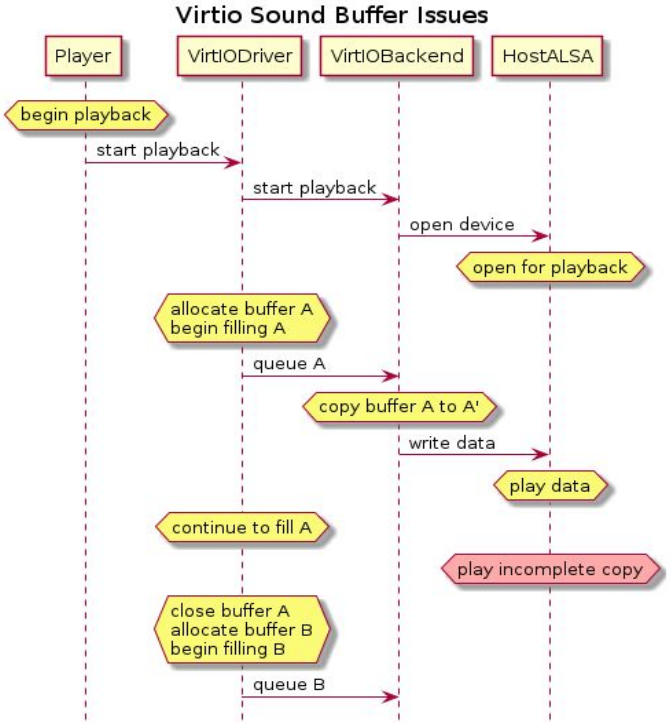


Media Devices in VirtIO

Media Challenges

- Higher throughput
  - Mp3: 128-320 kbps
  - Raw Wav: 1,411 kpbs
  - Compressed Video: 1.5 to 68 Mbps
  - GPU: up to 40Gbps
- Tighter Latency Requirements
  - IRQ latency gets in way
  - Zero-copy sought after

virtio-sound getting ahead of itself

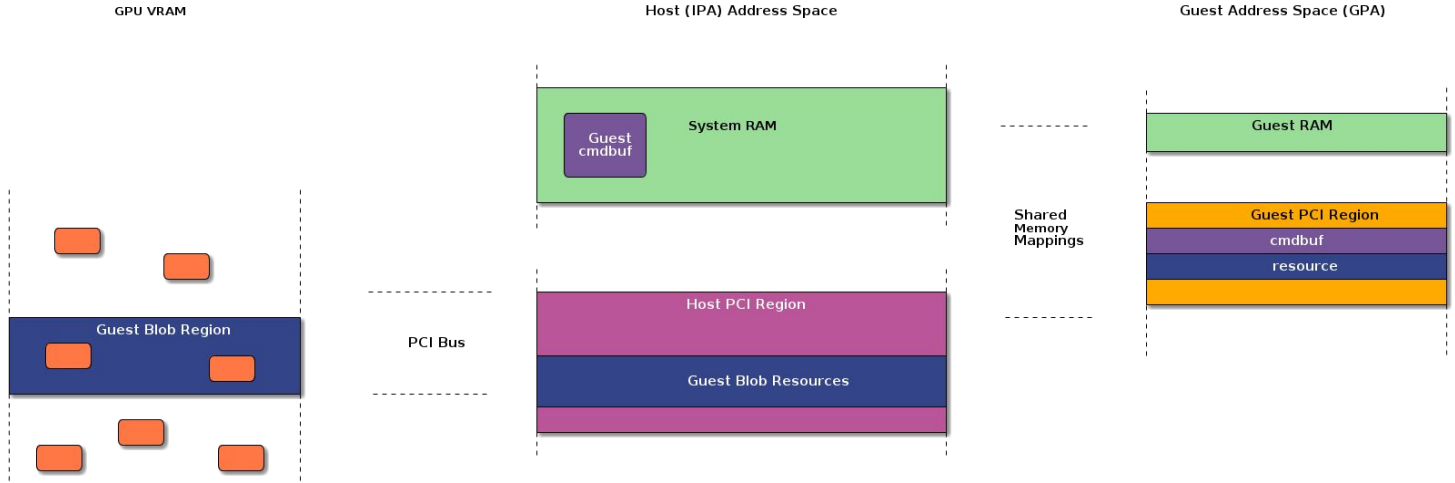


Adding [clarity to the spec]():

“The device MUST NOT access or modify buffers on a virtqueue after it has notified the driver about their availability.”

Shared Memory

- pre-allocated and shared
  - From host or guest domain
  - May have backend requirements
  - Still need to coordinate between domains



Stage 2 Tables

- Type-1 hypervisors manage stage 2 directly
  - Complex rules for merging and propagating attributes

Source: [Learn the Architecture - AArch64 memory model, Arm]

Tales of ~~woe~~

I mean “engineering opportunities”

PCI Implementations

- Errata on multiple-writes to PCI address space
  - Card memory treated as device memory
    - Different from x86 memory semantics
  - Workarounds in kernel
    - Same needed in the guest kernel
  - Implement quirk workarounds in Guest kernel

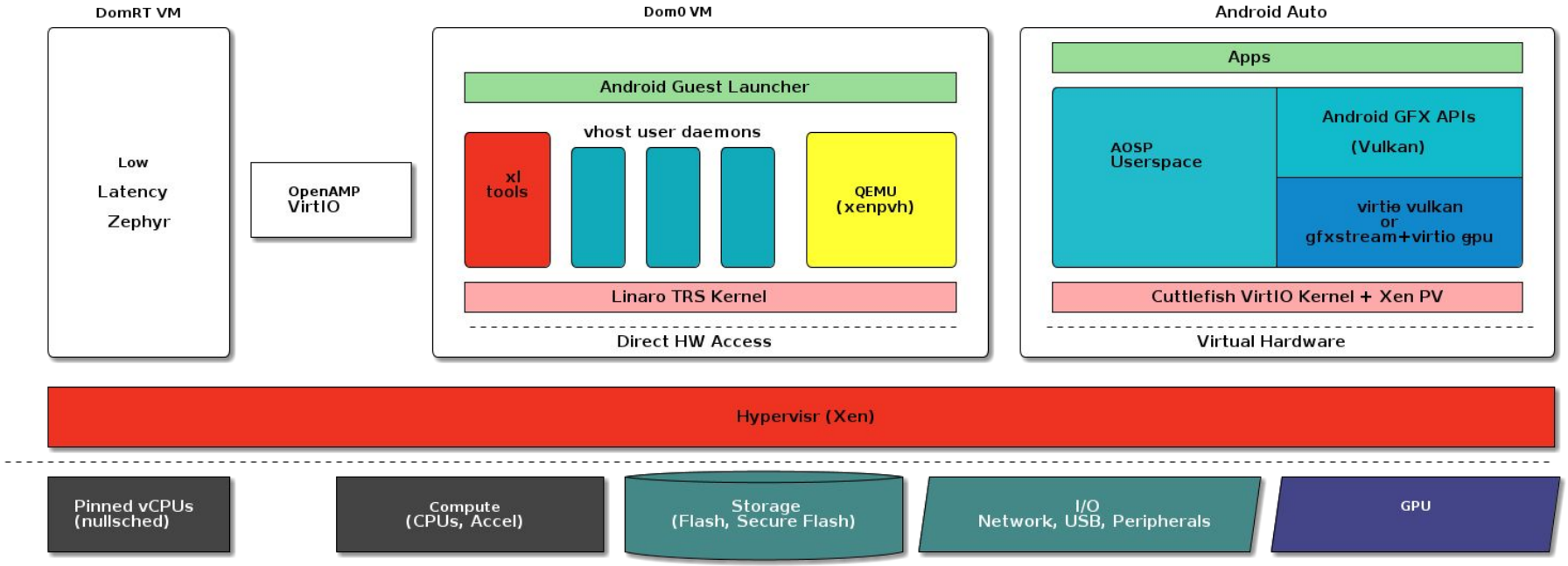
Bleeding Edge Software Stack

- A lot of tip-of-tree
  - Mesa (v24 for virtio-gpu Venus)
  - QEMU (8.2+)
  - Kernel (5.16+)
  - CrosVM gfxstream
- Multiple virtio-gpu approaches
  - Virglrenderer
  - Rutabaga with Wayland passthrough and SMO support
  - Venus/Vulkan with SMO support
  - Native Context

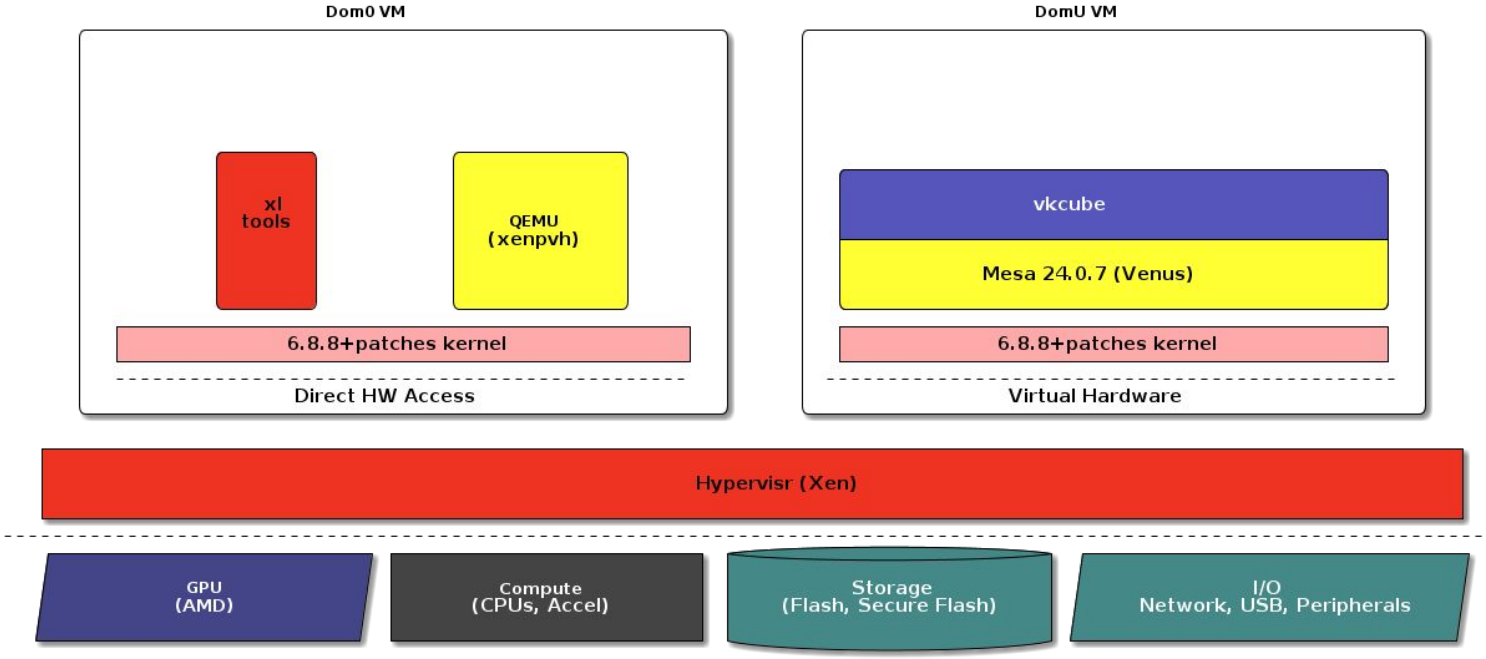
The Orko Demo

What’s on the Demo Friday Desk

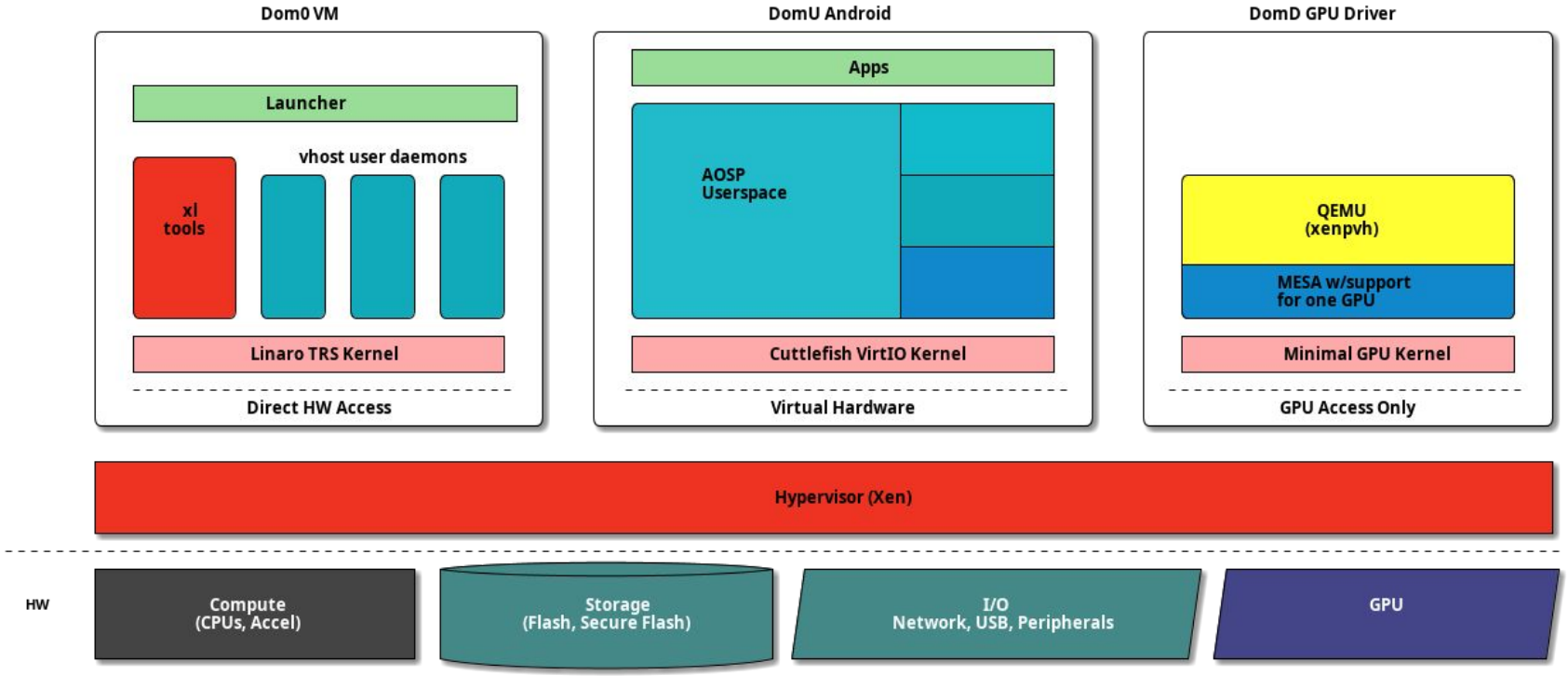
Original plan



Current State



Split Driver Domain



Questions?

Linaro Connect
MADRID 2024 | MAY 12-17 2024

Thank you