



libkrun: microVMs for running AI, Multimedia and Automotive workloads

Sergio Lopez Pascual
Senior Principal Software Engineer
at Red Hat

What is libkrun?

In a single quote

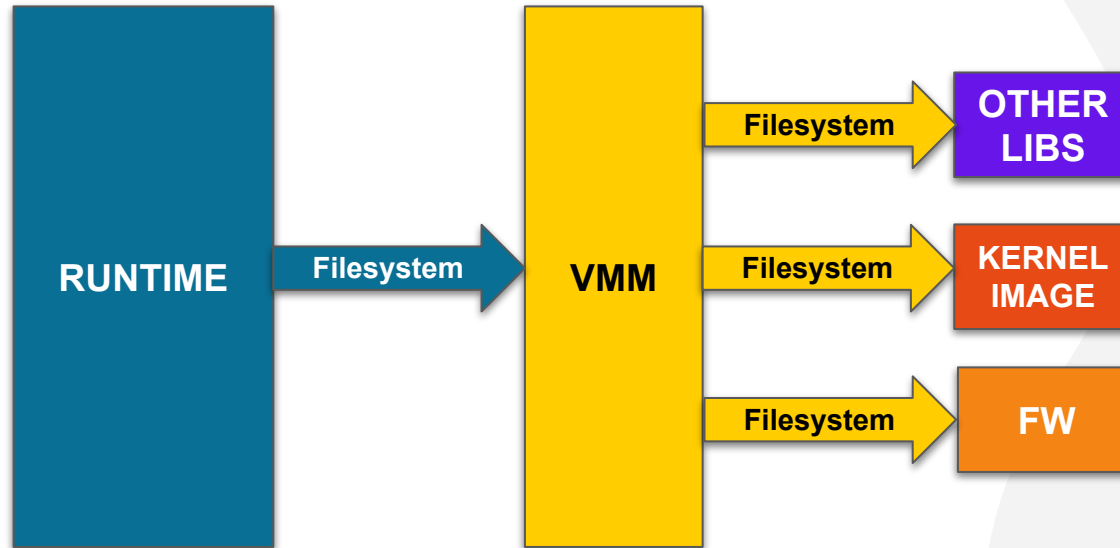
- “A dynamic library that enables other programs to easily gain Virtualization-based isolation capabilities, with the minimum possible footprint”
 - Originally, this quote said “KVM-based” instead of “Virtualization-based”, but libkrun now also supports Hypervisor.framework on macOS/ARM64.
- Originally intended to enable podman to run containers in a microVM, it has grown to address other use cases.

Key Aspects

- Written in Rust, but targeting C-bindings.
- Small!
- Integrates a virtio-fs server (no need to run an external process).
- Supports a minimal set of legacy devices, and the following virtio devices:
 - virtio-balloon (only free-page reporting)
 - Virtio-blk
 - Virtio-console
 - virtio-vsock (with TSI support)
 - Virtio-net
 - virtio-gpu with DRM native and venus contexts.
- It's very easy to import new devices from rust-vmm.
 - The goal is to eventually consume all devices from rust-vmm.
- Supports Confidential Computing!
 - SNP support merged, TDX and CCA incoming (PRs available)

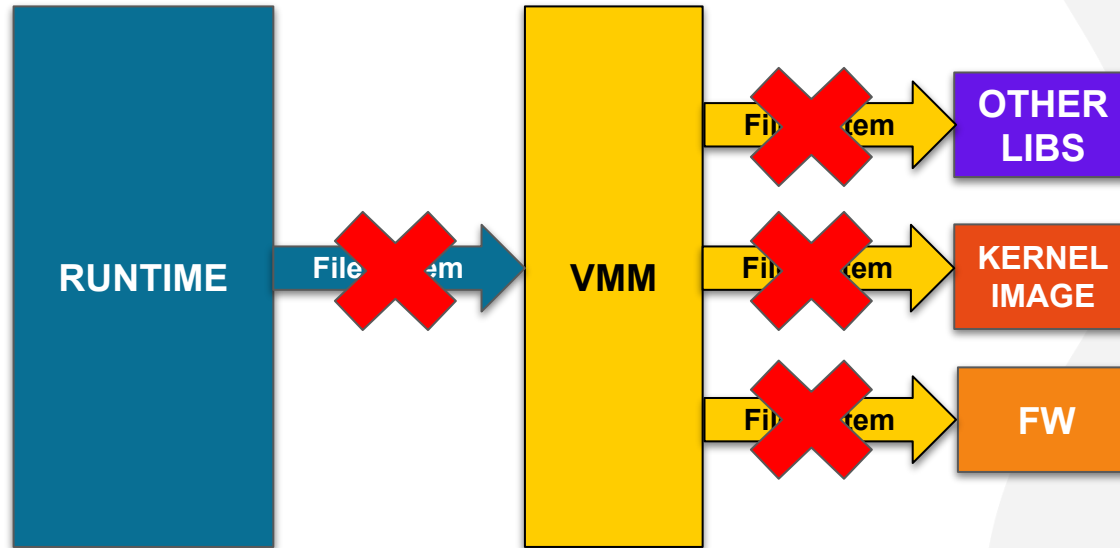
Why a dynamic library? (I)

Using an external VMM



Why a dynamic library? (II)

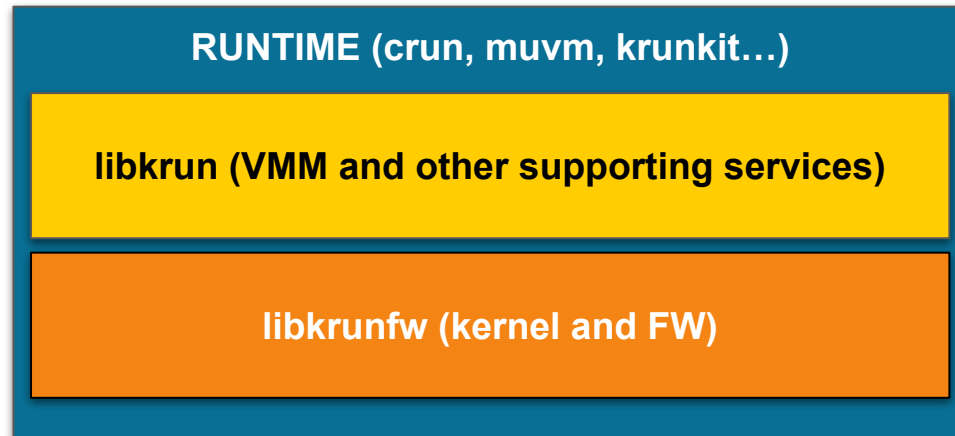
Using an external VMM after switching mountpoints



Why a dynamic library? (III)

Using libkrun

Process memory map



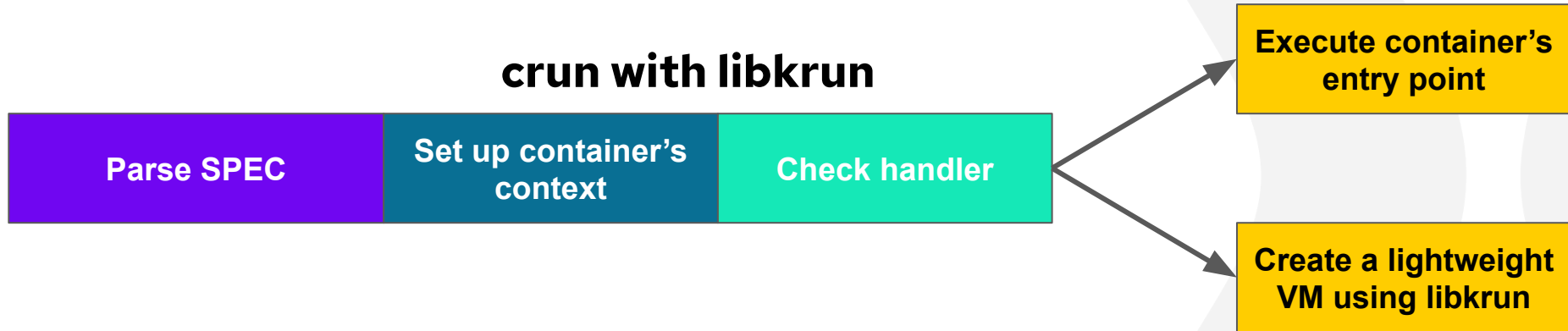
Integrating libkrun in crun

Executing a container's entry point (I)

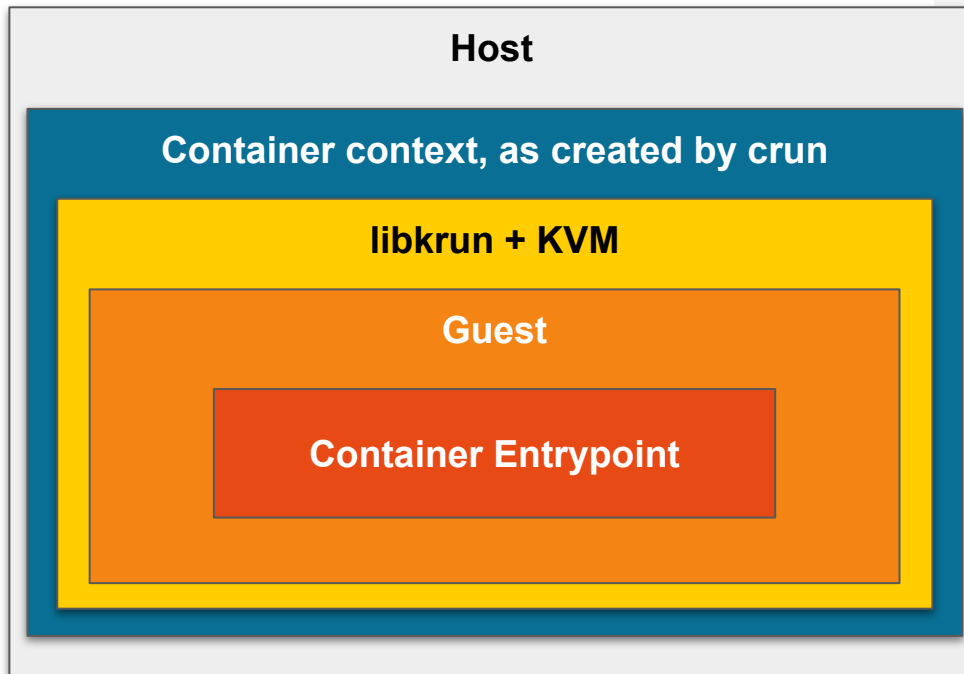
crun w/o libkrun



Executing a container's entry point (II)



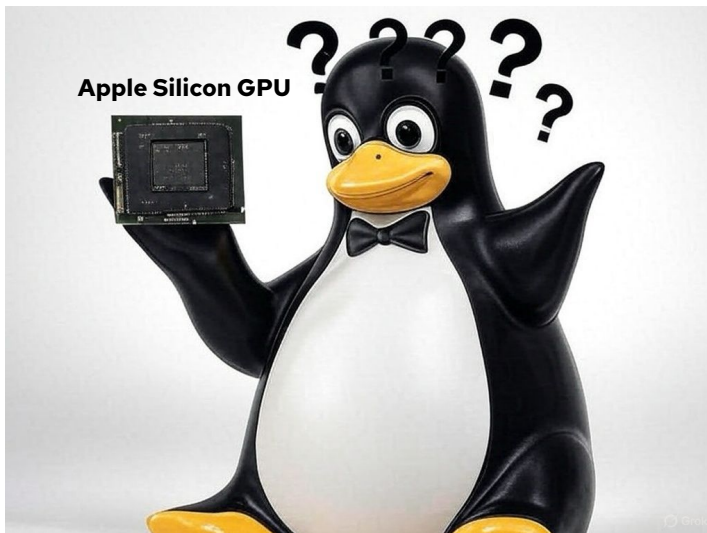
Combining containers and virtualization



Use cases: AI on macOS

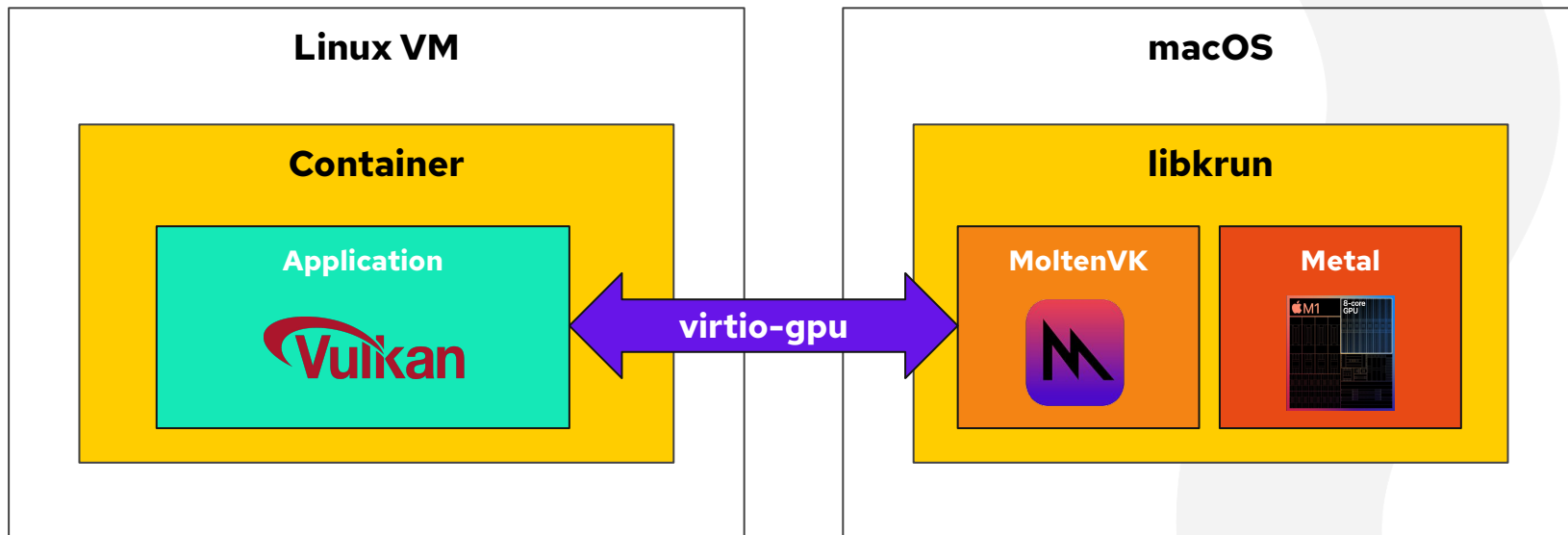
The Problem

- We want to run AI workloads in containers on macOS.
- Containers on macOS run in a Linux VM.
- Virtualization.framework provides a virtual GPU, but only to macOS guests.
- We want to use Vulkan!



The Solution

- libkrun can act as a VMM on macOS.
- virtio-gpu supports Vulkan through the Venus context.
- MoltenVK can translate Vulkan/GSL into Metal/MSL.
- Let's put them together!



Use cases: running games on Asahi Linux

The Problem

- Most games are x86_64, but Asahi Linux is aarch64.
 - We can use FEX-EMU or Box64.
- Asahi Linux runs with 16K pages, x86_64 games assume 4K granularity.
 - We can't realistically run a 4K kernel.

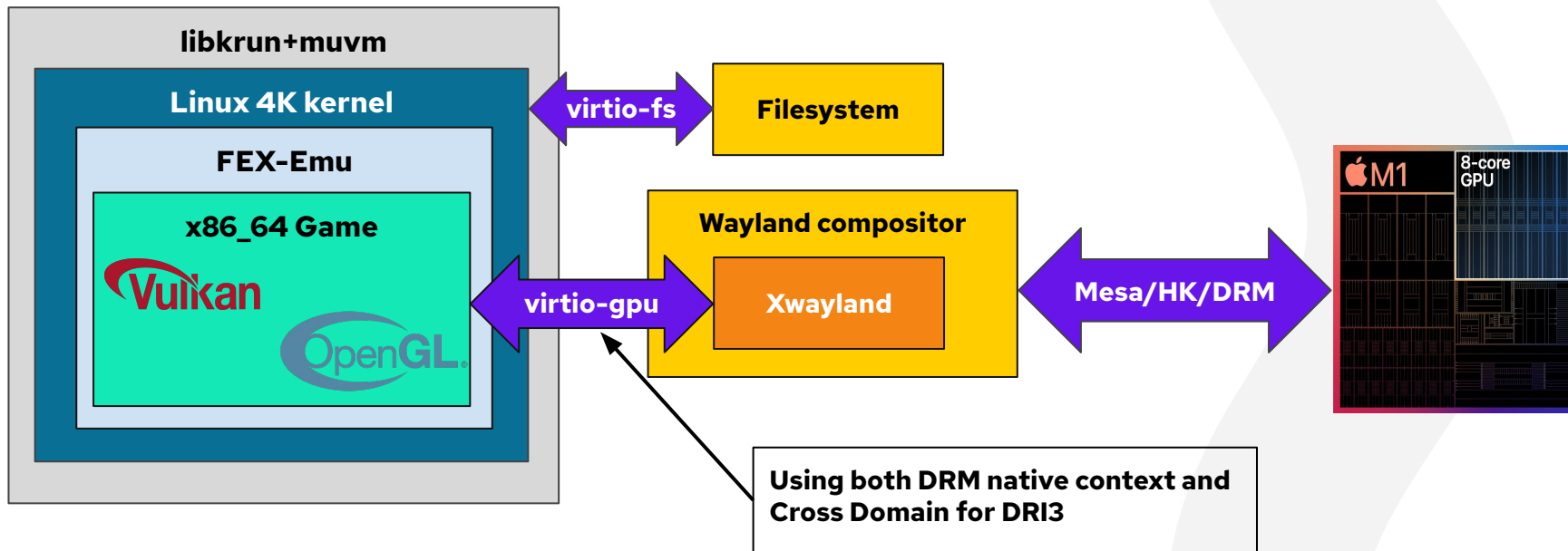


The Solution (I)

- libkrnl can run a 4K kernel in a 16K host.
 - It can provide a seamless experience, with the right config.
 - This is going to be muvm's job.
- But we still need a way to provide GPU acceleration to the guest (we want AAA games!)
 - Rob Clark's DRM native context in virtio-gpu has shown to provide good performance.
- We need something to coordinate with the host's compositor.
 - First option was Google's sommelier (a Wayland proxy).
 - Eventually got replaced with a simpler DRI3 proxy.

The Solution (II)

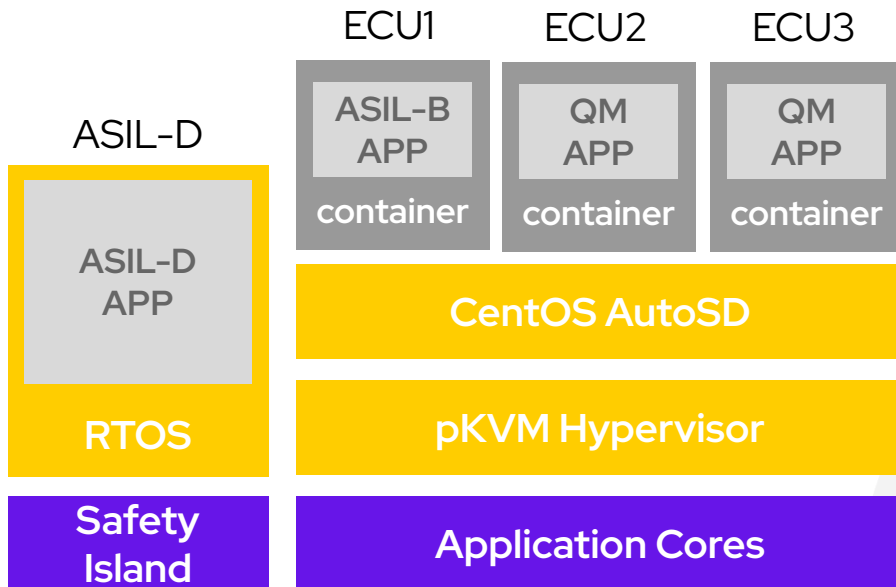
Asahi Linux (16K kernel)



Use cases: Virtualization-based isolation on AutoSD

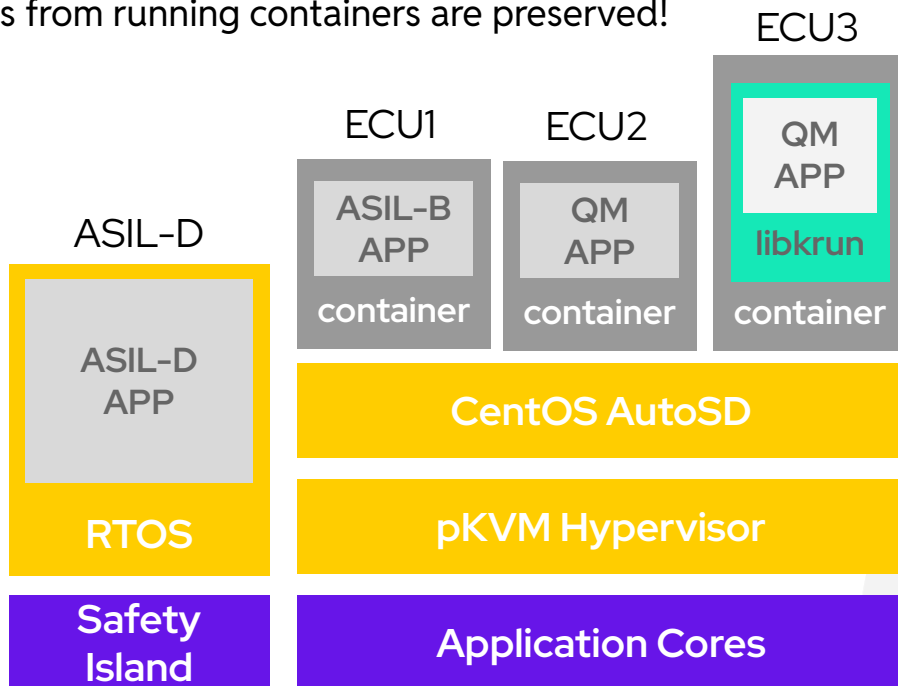
The Problem

- In AutoSD, we want to run every workload as a container.
 - But some environments require certain workloads to run as VMs.



The Solution

- crun, podman's OCI engine, is already able to use libkrun to run containers inside a VM, so let's just use it
 - Most benefits from running containers are preserved!





Thank You!