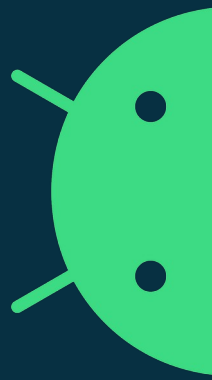
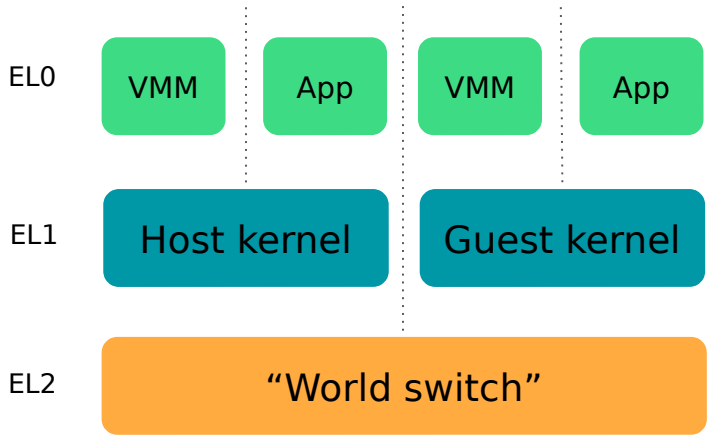
pKVM in-hypervisor tracing for ftrace^W tracefs

Tracing Summit 2022

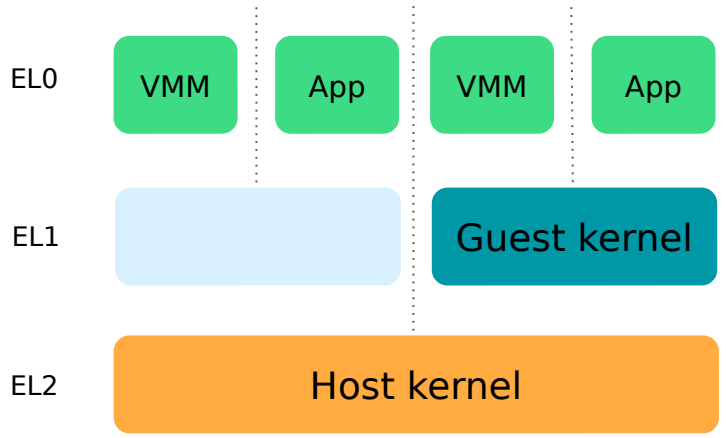


android

Arm64 KVM modes

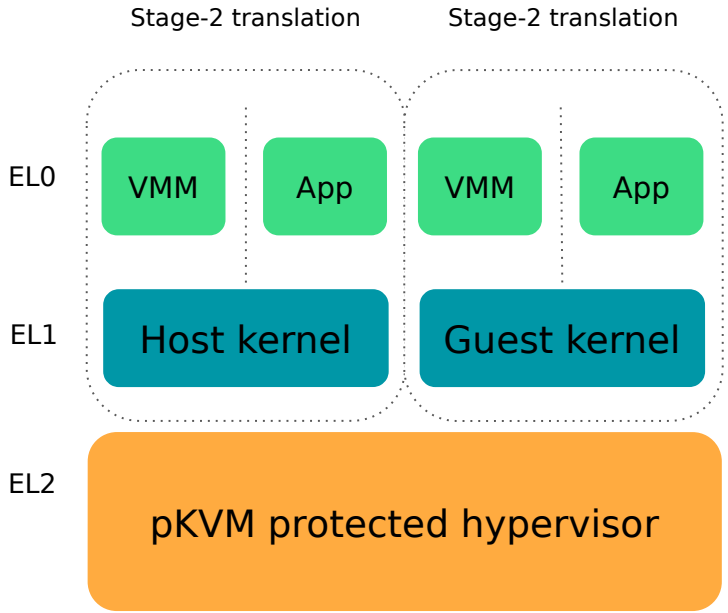


nVHE (v8.0)



VHE (v8.1)

Arm64 KVM modes



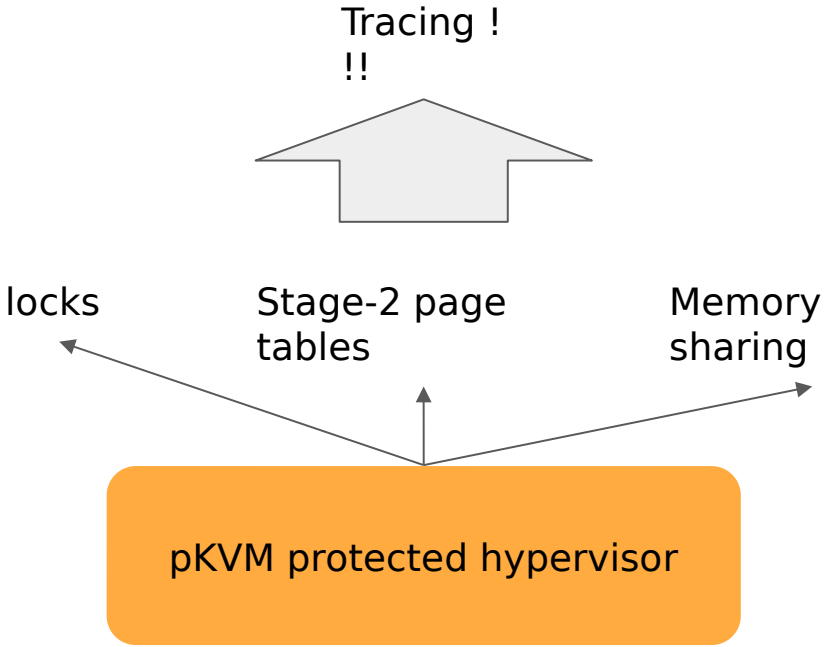
host VA != hyp VA

pKVM

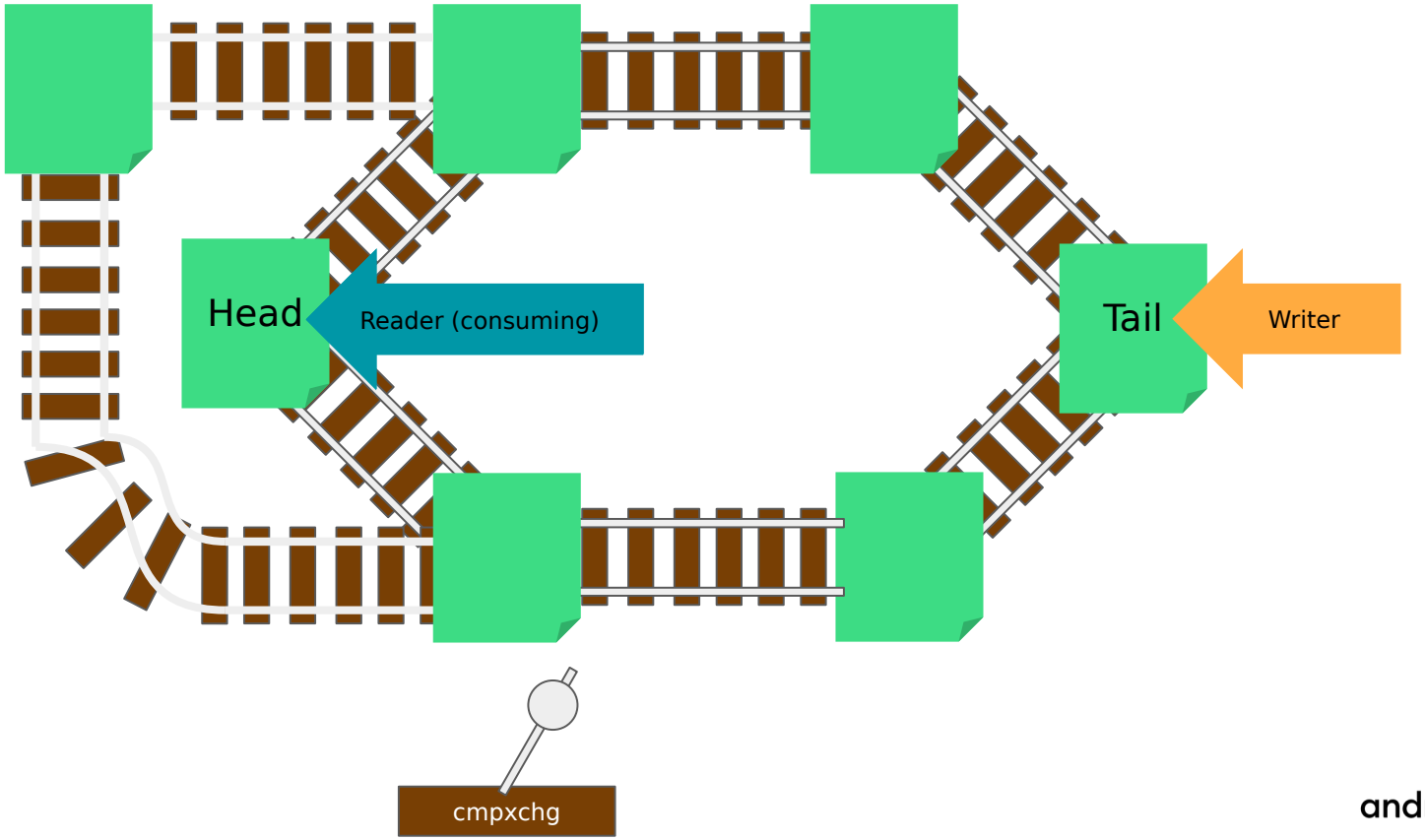
More? See *pKVM deep dive* @ KVM forum 2022
https://sched.co/15jLg

android

Motivations



Ftrace ring-buffer

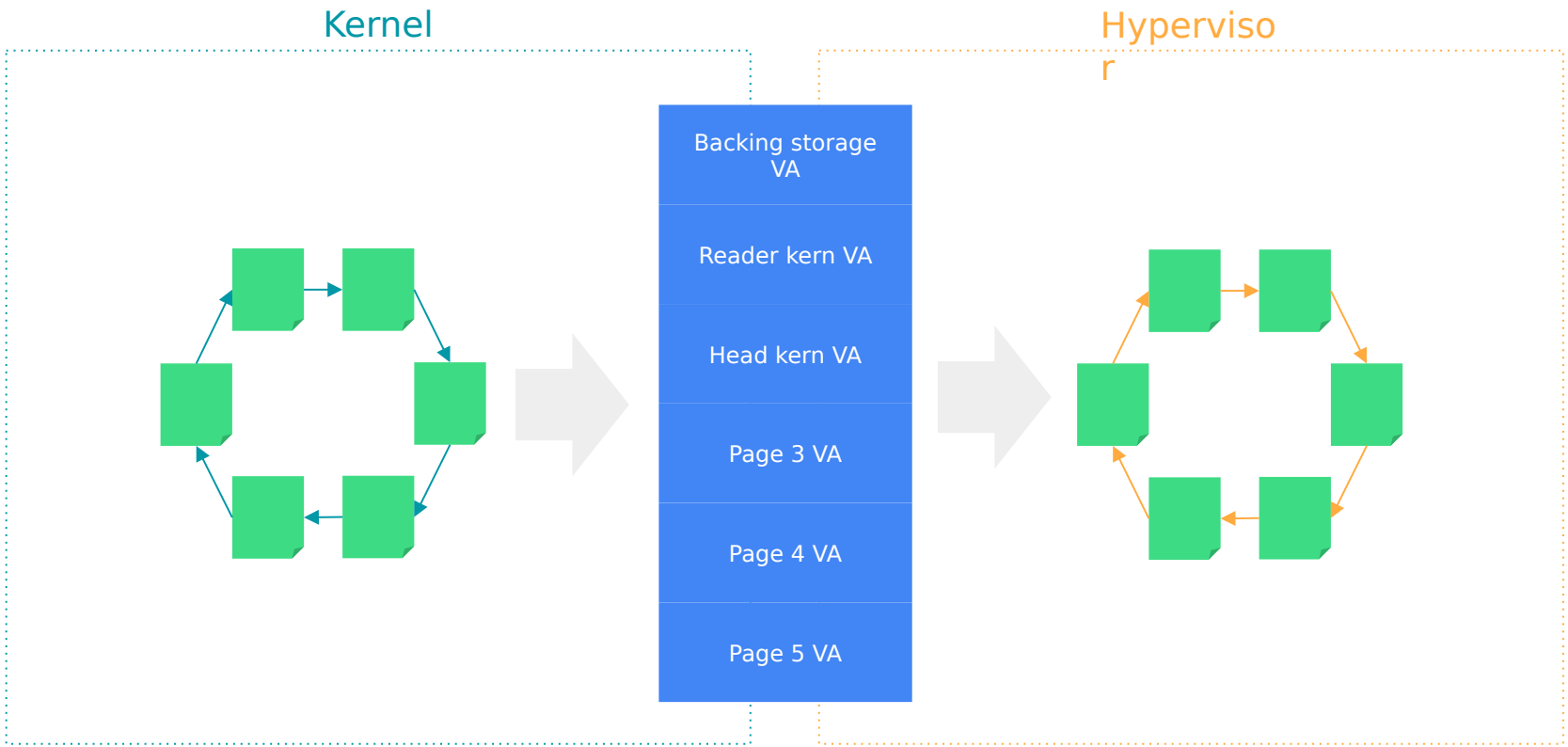


Problems?

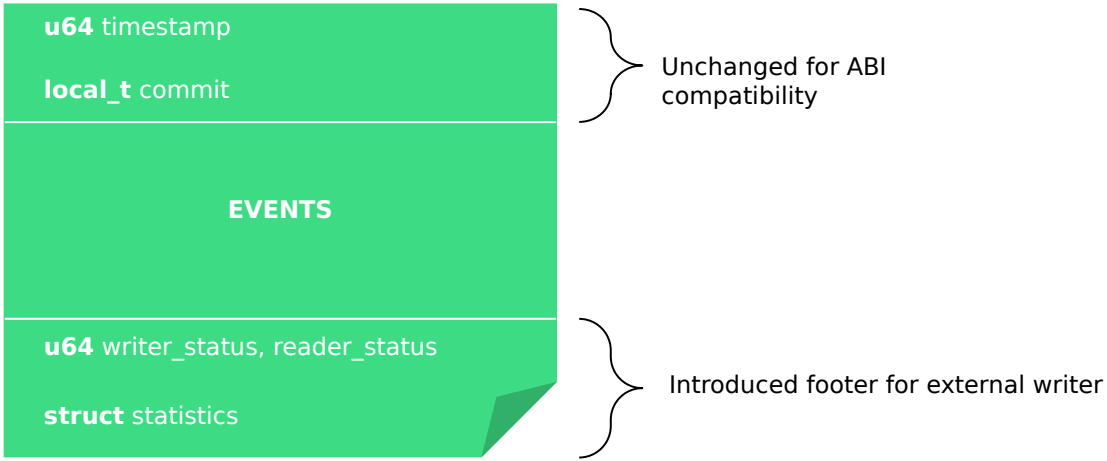
- Rails are just pointers… in kernel VA
- Rail switch embedded in those pointers

- Conversion Kernel VA to Hyp VA
- Can’t trust the kernel

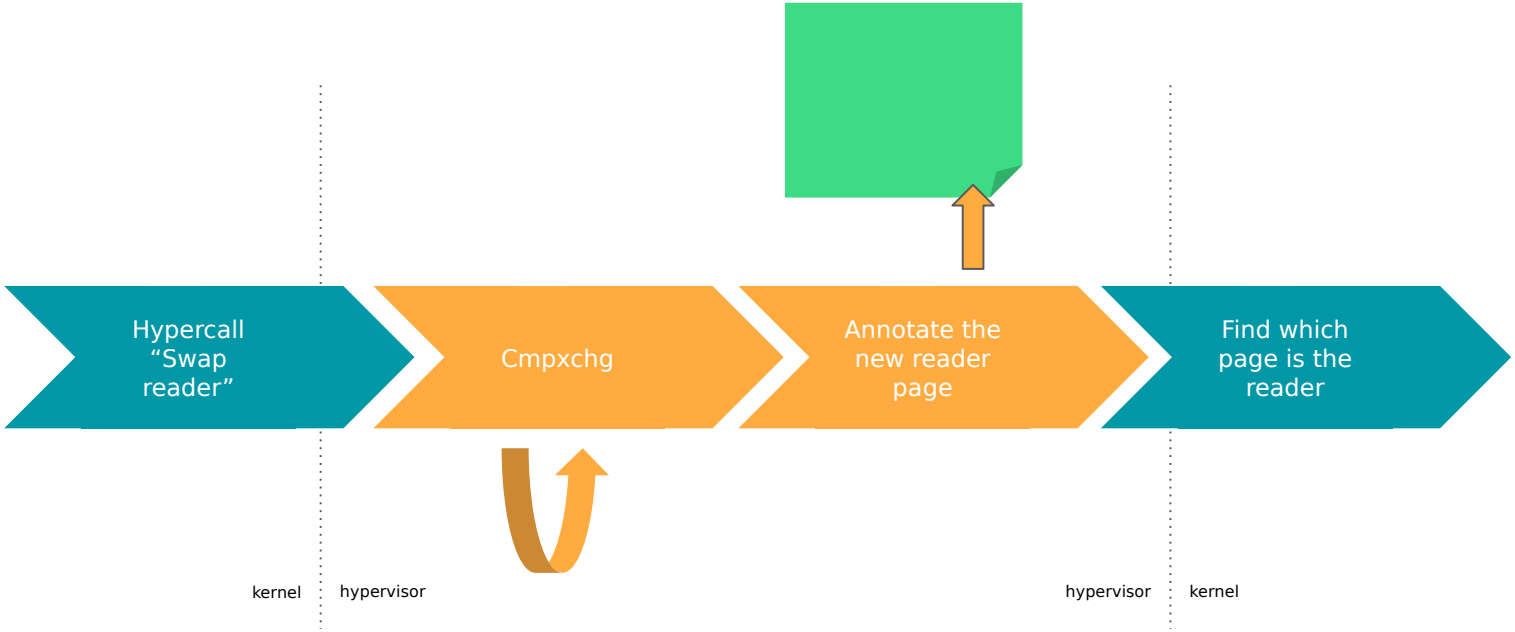
The links can’t be shared


Kernel
Hyperviso
r
Backing storage
VA
Reader kern VA
Head kern VA
Page 3 VA
Page 4 VA
Page 5 VA

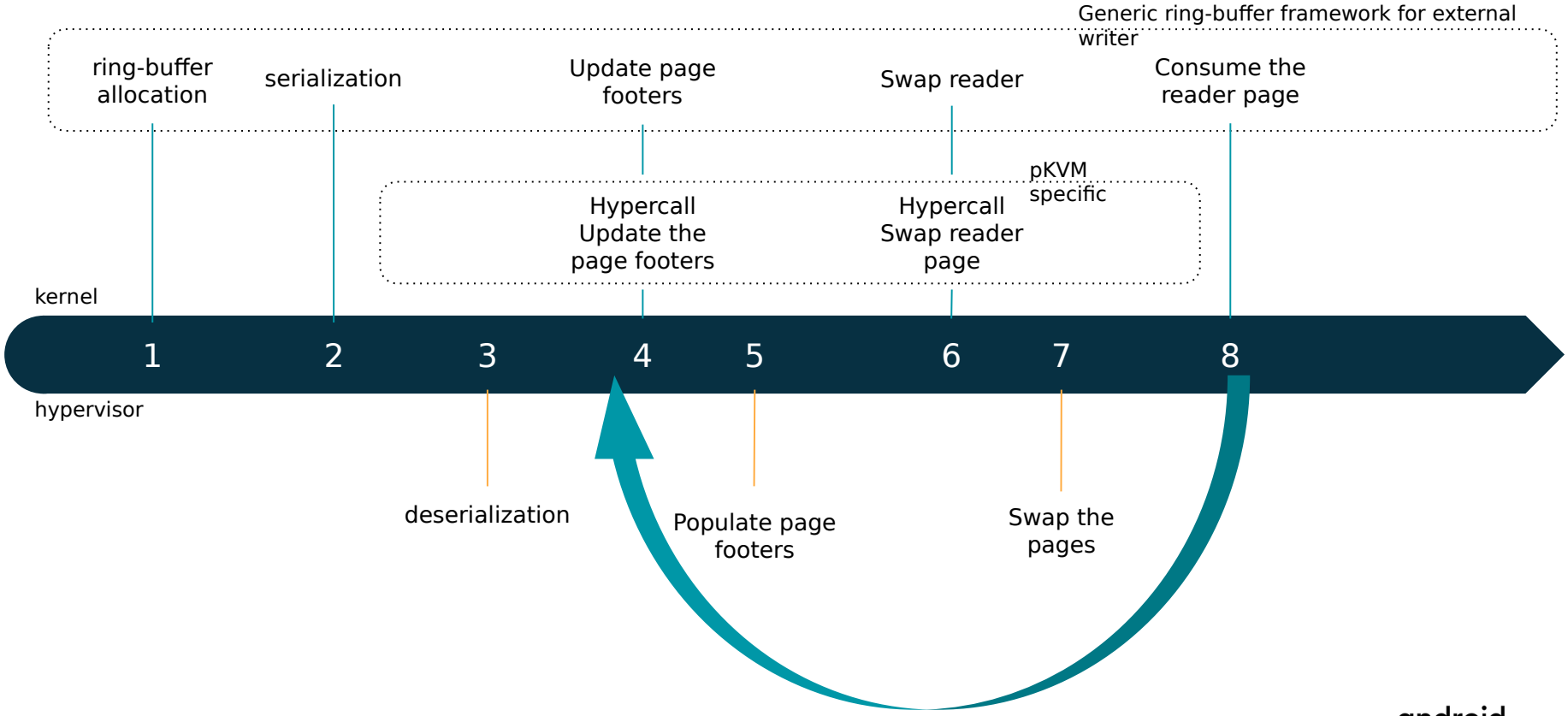
Sharing ring-buffer meta-data



What about the rail switch?



“External” writer support?



Userspace interface

Simply duplicate the root interface into a specific subfolder

- Ring buffers are specific anyways
- Different capabilities and features
- Similar to instances
- Easy for userspace tool to reuse

```
/sys/kernel/tracing/
                    hyp/
                        tracing_on
                        trace
                        trace_pipe
                        per_cpu/
                              cpuX/
                        events/
                              eventA/
                                    enabled
                              eventB/
                                    enabled
```

Current status

- No Nested writes
  - No preemption @ EL2
  - No IRQs taken @ EL2
  - Only Serrors … which could be masked
- Proof of concept used on a Pixel7 for debug and profiling
- Targeting production for Android 14

Thanks