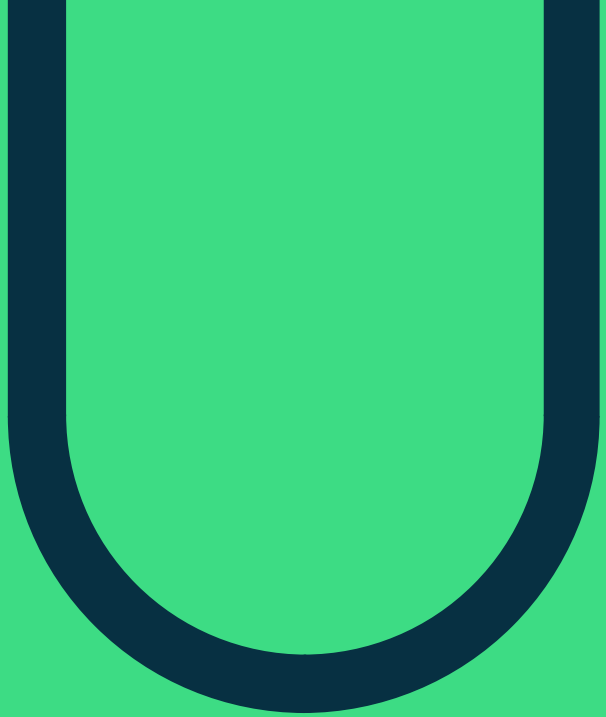
Android Automotive OS Virtualization



android

Enrico Granata

egranata@google.com
Software Engineer & Tech Lead
Android Automotive OS

Alistair Delva

adelva@google.com
Software Engineer & Tech Lead
Cuttlefish Virtual Platform

Cuttlefish/Cloud Android

- A reference virtual platform for Android (x86_64/arm64)
- Runs mainline Linux kernel/Android Common Kernel
- Upstream Virtio based: `block`, `console`, `gpu`, `input`, `net`, `pmem`
- Development focused on phone and tablet feature development
- Progress since last LPC: Vulkan, Secure HALs, Bootloader, New Radio, Audio WIP

Goal: As much as possible, mimic the behavior of real devices using the capabilities of hypervisors and cloud-hosting technologies

Trout/Android Automotive OS

- Downstream of Cuttlefish
- A reference *virtual* platform for Android Automotive
- Development focused on automotive features
- Deploy on real vehicle hypervisors (not just QEMU)

Goal: A virtual reference platform that supports all Android Automotive OS features in a compatible and performant manner

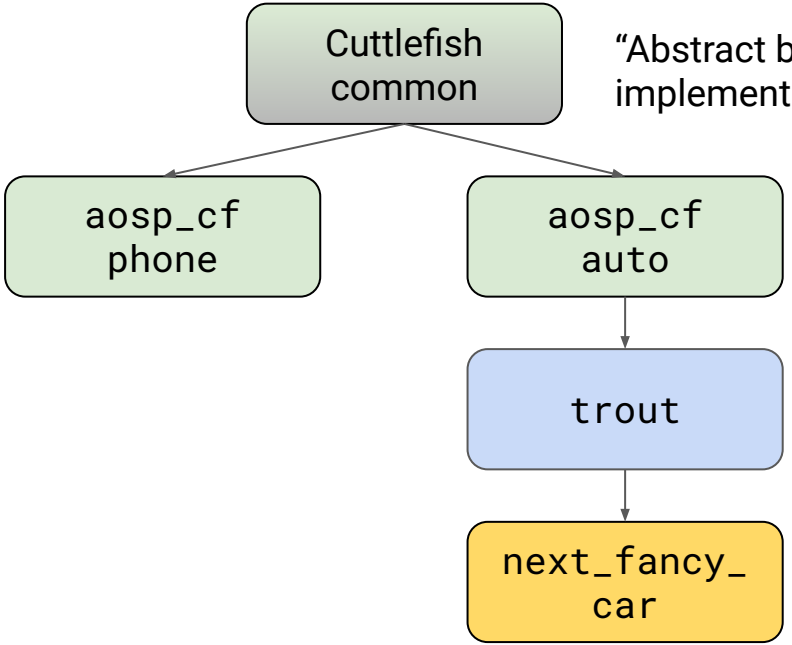
Cuttlefish Meet Trout

Existing AOSP support code

Existing AOSP *lunch* target

New AOSP *lunch* target

OEM-specific *lunch* target



"Abstract base", virtio-based; common HAL implementations, configuration files, and so on

Adds automotive features (EVS, Vehicle HAL, and more)

Adds virtualized HAL implementations

Downstream support for OEM-specific UI and features

android

In-car Virtualization

- Cockpit consolidation is a real trend: lower BOM cost, lower vehicle weight
- Ever more powerful SoCs enable more use cases
- Some use cases are critical (for example, interrupt response guarantees, memory isolation)
- Shared hardware access imposes unique challenges (for example, which network does one Wi-Fi adapter connect to?)

In-car Virtualization

Virtualization allows:

- Availability guarantees (for example, guaranteed core/memory/device allocation)
- Disparate environments (for example, Linux + RTOS) and well-defined boundaries

Good fit for automotive use cases/concerns:

- Certification relies heavily on “what properties can this system guarantee”?
- Safety is a major focus area in the industry
- Security increasingly important, as cars become digital hubs

New VirtIO Devices

Open standards allow easier integration, deployment and maintenance. We can support Android-as-a-VM once, and deploy across multiple OEMs.This is critical for Android Automotive OS.

We collaborated on new virtio devices:

- `virtio-snd` (approved, landing in 1.2)
- `virtio-video` (work in progress, v4 latest drop, v3 used in AAOS)
- `virtio-scmi` (work in progress, used alongside new sensor features in SCMI protocol)

New drivers being implemented in Android Common Kernel

Committed to upstream.

vsock

Some AAOS features are backed by vsock/gRPC:

- Vehicle HAL (Android interface to CAN bus/car status)
- Dumpstate HAL (Android interface to collect helper system logs, or host VM in our use case)
- Audio Control HAL (management of audio hand-off between Android and car)

Some of these are open opportunities for new virtio features:

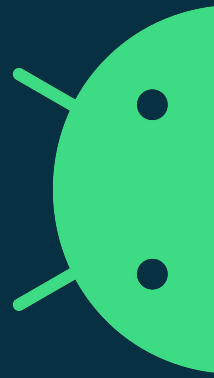
- `virtio-car`
- `virtio-snd` + audio bus ownership

Hacking on AAOS

- Trout will be released with Android 11 (soonish)
- Download the code and build for `aosp_trout_arm64-userdebug`
- You don’t need a vehicle :)
- It can work in crosvm (with some limitations)
- Reference implementation on hardware is WIP
    - Separate announcement to follow

Discussion

Android Automotive OS Virtualization



BACK UP SLIDES

Cloud Android - Architecture