

Virtio & Vhost-User Ecosystem BoF

Aligning VMM Governance and Production Automotive stack.

Goals:

- Establish better coordination across the virtio/vhost-user ecosystem
- Align technical roadmaps for cloud and automotive deployments
- Address specification governance challenges

Expected Outcome:

- vhost-user spec governance decision
- Automotive coordination process
- rutabaga_gfx roadmap alignment
- Shared Rust crate identification

Key discussion topics:

1. Vhost-user Specification Governance: Discuss official location for the vhost-user specification.

2. Automotive Virtualization Requirements & Reference Platforms: Coordinate deployment needs across Automotive platforms

3. Real-time performance needs (camera, display, CAN)

4. Status Update: Ongoing work on vhost-user device (virtio-media, virtio-can, virtio-rtc, etc)

5. Graphics Stack Alignment: on roadmaps for shared dependencies like rutabaga_gfx across libkrun, crosvm, vhost-device-gpu

6. Shared Virtio Message-Parsing Crates: Evaluate creating unified Rust crates for Virtio device specs to share virtqueue parsing logic across backends and VMs.

7. Open questions about:

- vhost-user Specification Refinements: Identify protocol gaps, missing features, and necessary standard improvements.
- New Virtio/vhost-user Device Proposals.

Vhost-user Specification Governance

Options

OASIS VirtIO appendix

Separate repo under qemu-project

New independent project

Status quo

Discussion

- Other stacks want a neutral home; `qemu-devel` is a high-volume barrier
- **OASIS**: visibility vs bureaucracy; this is VMM ↔ backend IPC, not VirtIO proper
- **Not automatically faster for QEMU**
 - Today: spec + impl in one release
 - Versioned spec repo: spec → spec release → QEMU → QEMU release
 - Faster only as a living `main` (or submodule), no extra release gate
- Freeze need not block `vhost-user.rst`; splitting may lose QEMU reviewers

Reference: [Where should the vhost-user specification live?](#)

Shared Virtio Message-Parsing Crates

Problem & Proposal

- **Problem:** Each Rust backend reimplements device protocols (`vhost-device-gpu`). Frontends duplicate/invert this.
- **Proposal:** Shared spec crates (`virtio-{gpu,snd,...}-proto`).
- Same codec on backend & frontend (guest writes cmd / host writes resp; swapped Read/Write).

Pattern today (`virtio-media`)

- Crate owns structs + `handle_command` (read cmd → dispatch → write resp)
- Backend only implements traits: desc-chain Read/Write, eventq, guest memory
- `vhost-device-media` is adapters + backend, not a second parser

Layers

1. **Queue/Memory:** `virtio-queue` / `vm-memory` (rings + GPA)
2. **Spec Crate:** Protocol messages & layout definition
3. **Transport / VMM:** `vhost-user` sockets, features, mapping
4. **Device Backend:** V4L2, virgl, and specific hardware logic

Talking Points

- Same encode/decode for both ends
- Spec crate owns wire layout; backends/frontends own policy and hardware
- Where it lives: `rust-vmm` vs per-device repo

Automotive Virtualization: Reference Platform & Device Roadmap

Reference Platform Status

- Android Automotive Trout reference platform discontinued August 2026
- Migration to Cuttlefish recommended by Google
- Cuttlefish's virtio device support is not clearly documented
- Last Trout update used 7 virtio device types: `virtio-video`, `virtio-gpu`, `virtio-snd`, `virtio-console`, `virtio-scmi`, `virtio-input`, `vsock`

The Gap

- VirtIO 1.4 spec defines 23 device types (including automotive-critical devices)
- **Missing from Trout:** `virtio-can`, `virtio-rtc`
- **Spec governance issue:** Trout used “**virtio-video**” (IDs 30/31) for EVS camera decoding/encoding - not yet in **VirtIO 1.4** but has **virtio-media** (ID 48)
- Production systems using draft specs = certification nightmare

Discussion Points

- **Is anyone here working on automotive/IVI deployments?**
- **Device priorities for automotive & specification clarity:**
 - `virtio-can` for vehicle networking (in VirtIO 1.4, not in any reference platform)
 - `virtio-rtc` for timing/synchronization (in VirtIO 1.4, not in any reference platform)
 - **virtio-video vs virtio-media** Which should automotive use for camera/EVS?
- **What does the community need from a reference platform?**
 - **Clear documentation** of supported virtio devices
 - Safety certification path (ISO 26262, ASIL-B/D)

Rutabaga_gfx Coordination & Roadmap Alignment



