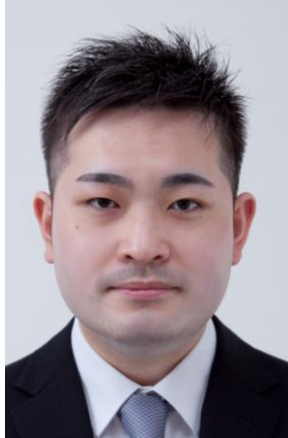


Update on VirtIO CTS and Integration of Virtio-rtc and Virtio-media Into AGL UCB

Kazuki Kuzu
Yuki Teramura

Panasonic Automotive Systems Co., Ltd.

ABOUT ME



Kazuki Kuzu

Panasonic Automotive Systems Co., Ltd.

Member of Automotive Grade Linux - SDV Expert Group

Working for: Embedded and Hypervisor for 6 years

Like

- PC Games
- Skiing
- Meat
- whisky
- Green Tea



Agenda

- ✓ Introduction
- ✓ VirtIO CTS Concepts and Roadmap
- ✓ Integration of virtio-devices

Agenda

- ✓ Introduction
- ✓ VirtIO CTS Concepts and Roadmap
- ✓ Integration of virtio-devices

Importance of Virtualization and VirtIO in SDV

Virtualization in Automotive Domain

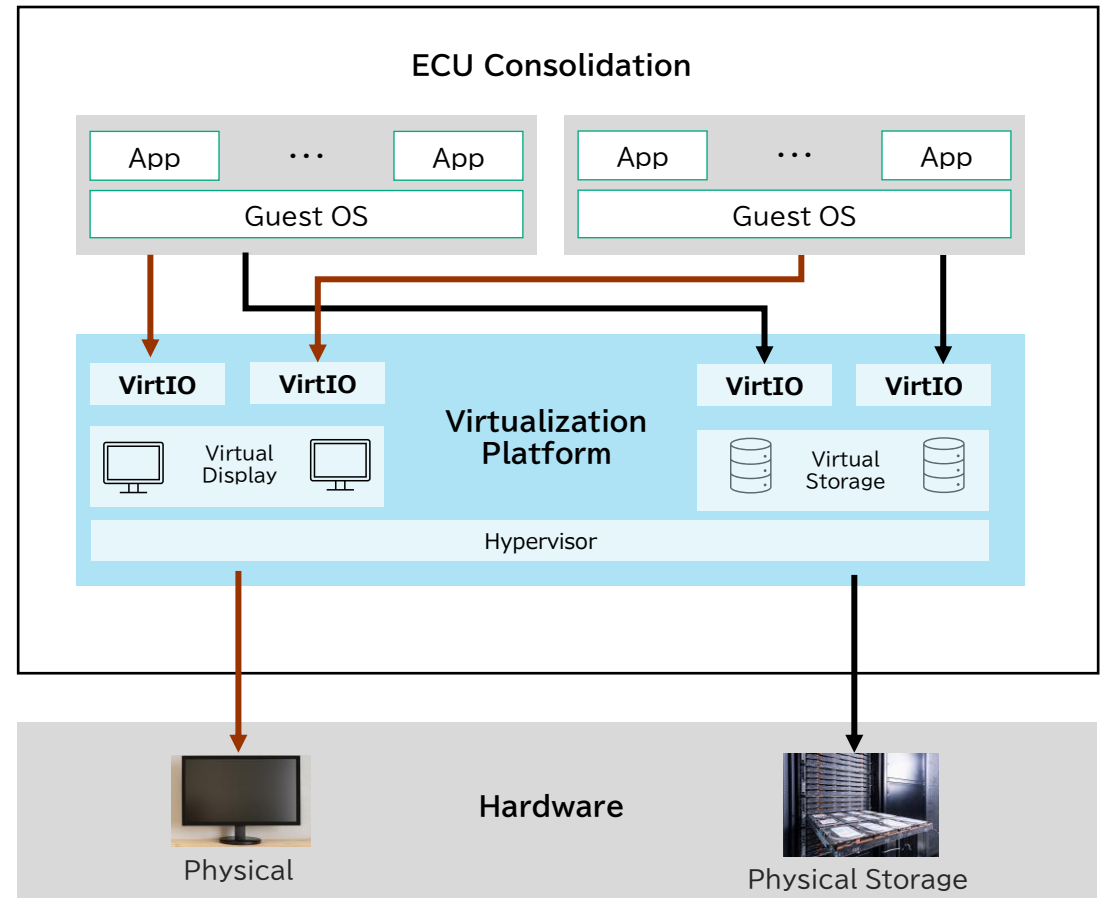
By consolidating multiple systems on high-performance computers, virtualization enables centralized software updates.

In addition, hardware abstraction allows cloud-native development and easier porting to next-generation hardware.

Additional benefits provided by VirtIO

VirtIO provides a standardized device paravirtualization interface.

By decoupling software from hardware, it improves portability across different hardware platforms.

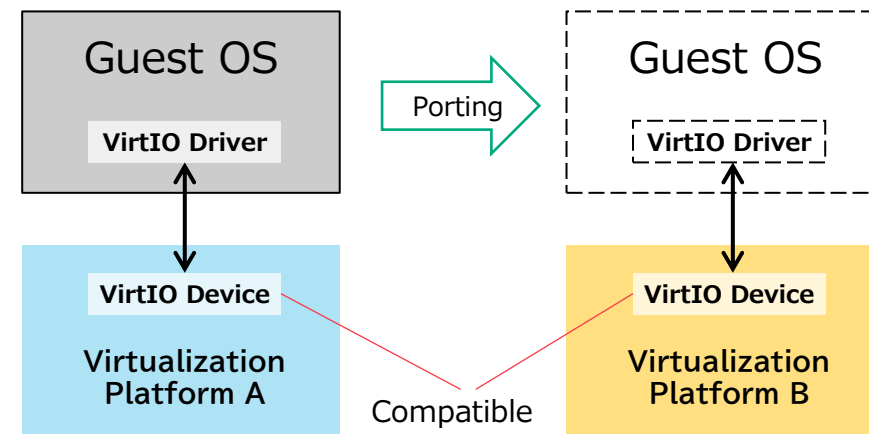


Our Efforts to Enhance VirtIO support for SDV

Improving SW portability by checking the compatibility of VirtIO Devices

Due to long product lifecycles and strict requirements, automotive systems often introduce proprietary VirtIO implementations.

VirtIO Conformance Test Suites (VirtIO CTS) addresses this Challenge by verifying VirtIO device compatibility to maintain software portability.



New VirtIO Devices in AGL

In OASIS VirtIO Version 1.4, several new VirtIO device types—such as CAN, SPI, RTC, and Media—were standardized.

Among them, we integrate VirtIO RTC and VirtIO Media into AGL to support automotive use cases.

Virtual I/O Device (VIRTIO) Version 1.4

Newly standardized device types

- CAN
- SPI Controller
- **Media**
- **RTC**

Agenda

- ✓ Introduction
- ✓ VirtIO CTS Concepts and Roadmap
- ✓ Integration of virtio-devices

VirtIO CTS (Conformance Test Suites)

VirtIO CTS Validation

VirtIO CTS offers a unified, cross-hypervisor validation approach to ensure conformance with OASIS specifications.

> Guest OS Perspective Testing

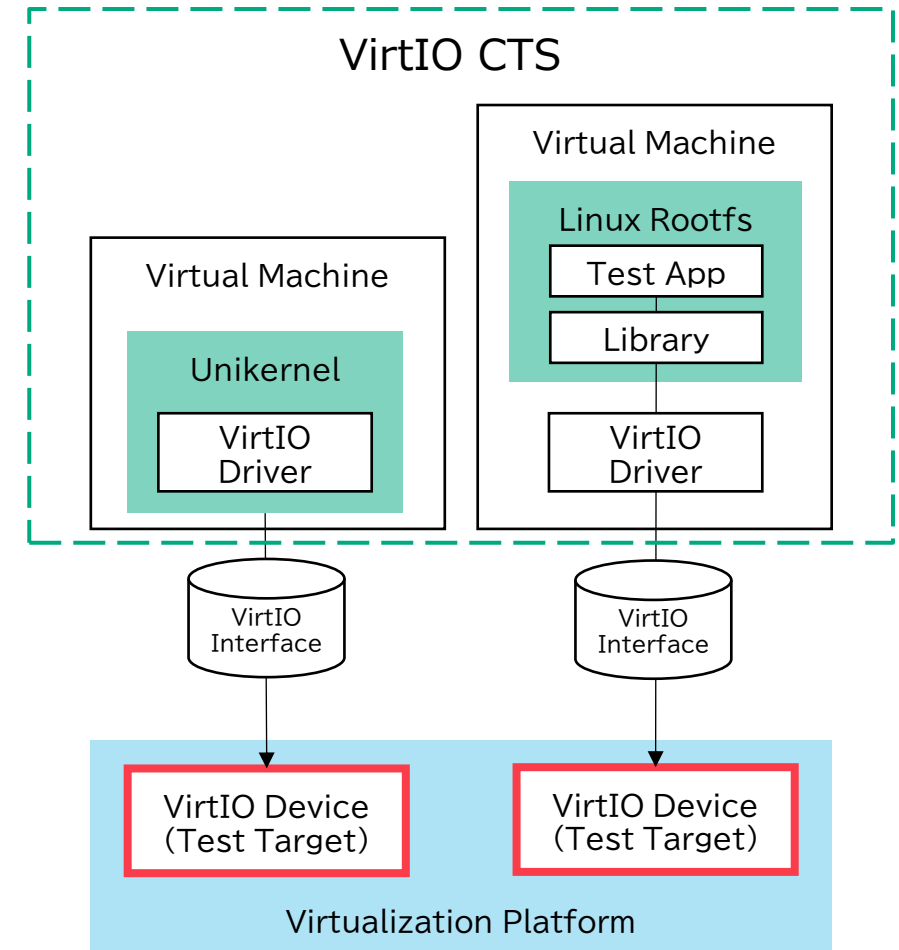
All tests run from the guest OS viewpoint to verify that VirtIO devices behave as expected by a generic guest OS.

> Platform Independent Architecture

Tests avoid hypervisor dependencies, enabling consistent execution across multiple virtualization platforms.

> Dual Test Approaches

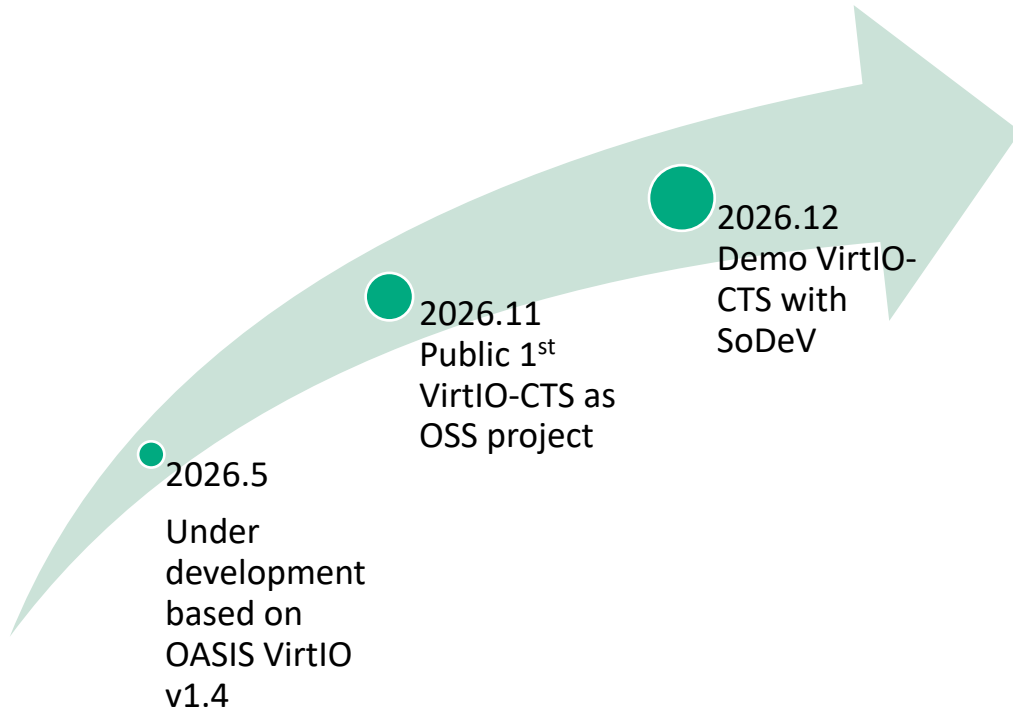
Unikernel-based tests offer fast basic compliance checks, while Linux-based tests handle complex device validation.



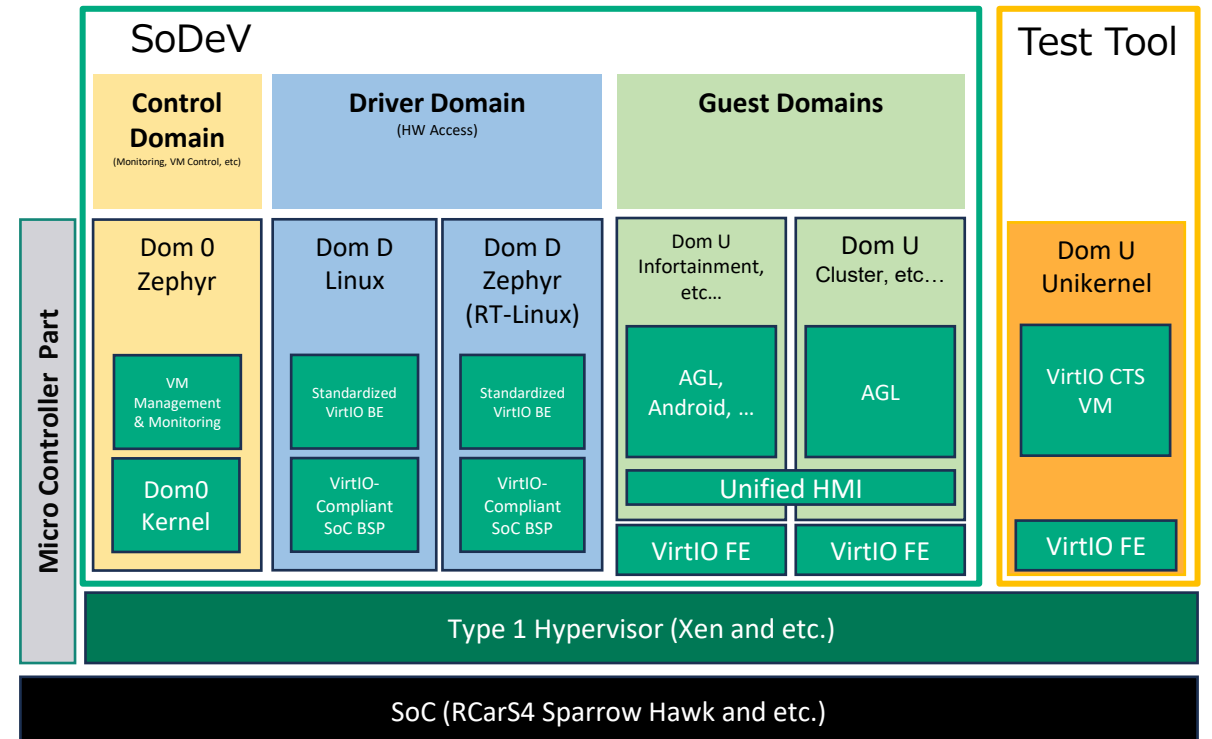
Future Plan

- > We are planning to release VirtIO CTS on 2026.11
- > We are also planning to demonstrate VirtIO CTS with SoDeV at the coming AMM.

Roadmap



Demonstration



Agenda

- ✓ Introduction
- ✓ VirtIO CTS Concepts and Roadmap
- ✓ Integration of virtio-devices



Yuki Teramura
Panasonic Automotive Systems Co., Ltd.



Like

- ◆ cooking (especially making lunch boxes)
- ◆ visiting aquarium
- ◆ dancing



Devices scheduled to contribute to AGL

- > Contribution the newly added **virtio-rtc** and **virtio-media** module
from OASIS VirtIO 1.4 (Virtual I/O Device (VIRTIO) Version 1.4 : 2026/04/08 published)

> VirtIO RTC

Provided Features

- > Time synchronization between the host VM and guest VMs
(the guest VM obtains time information from the host VM)
- > Alarm setting support

Background for OASIS Submission

- > Insufficient accuracy for real-time applications
- > Recognition of UTC, TAI, and Monotonic clock types by guest VMs

Automotive Use Cases

- > Handling of time information during guest VM suspend / resume
- > Debugging : collection and analysis of event logs across multiple VMs

> VirtIO Media

Provided Features

- > Virtualization of the V4I2 UAPI for camera and codec devices supported by V4I2, and providing them to guest VMs

Background for OASIS Submission

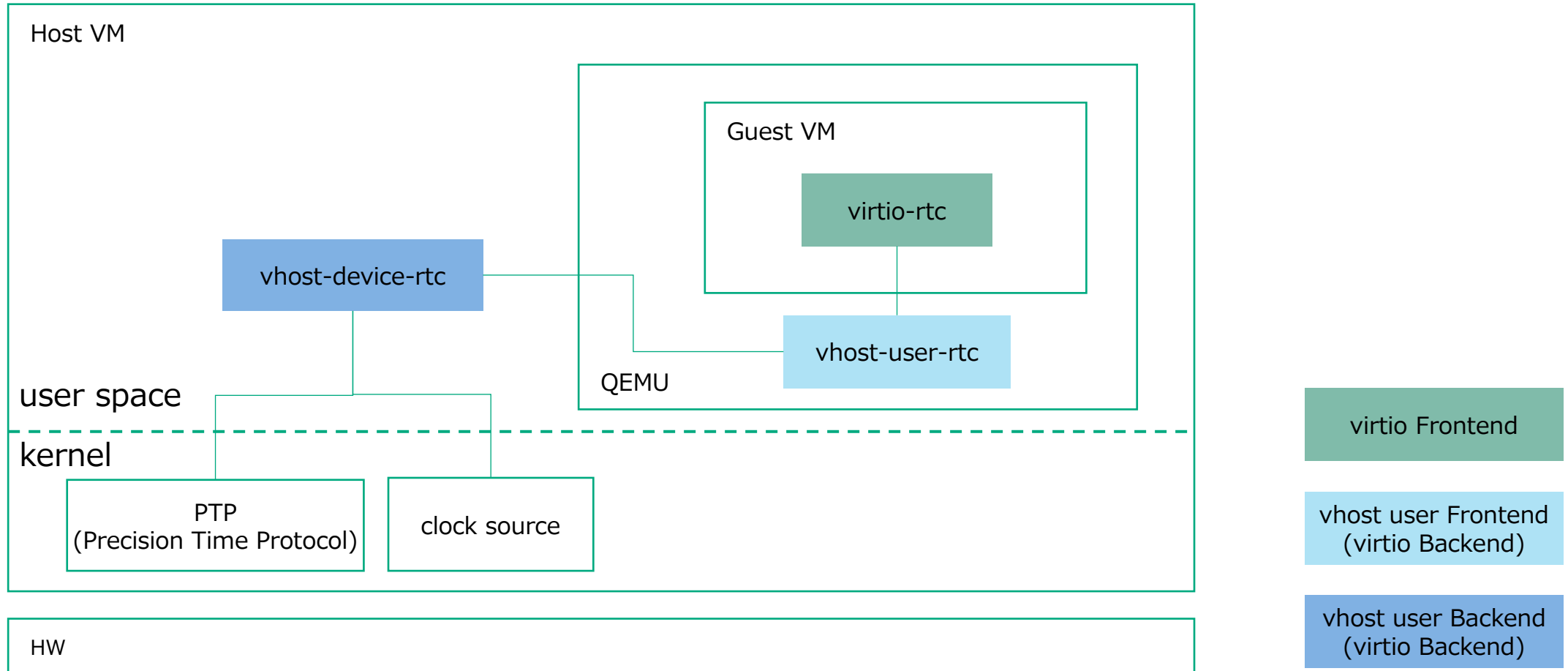
- > Strong demand for full V4I2 virtualization for use cases involving camera, ISP , M2M and capture -> encode -> decode pipeline

Automotive Use Cases

- > Provision of camera hardware video stream to guest VMs
(virtio-video is mode codec-oriented and is not defined in OASIS)

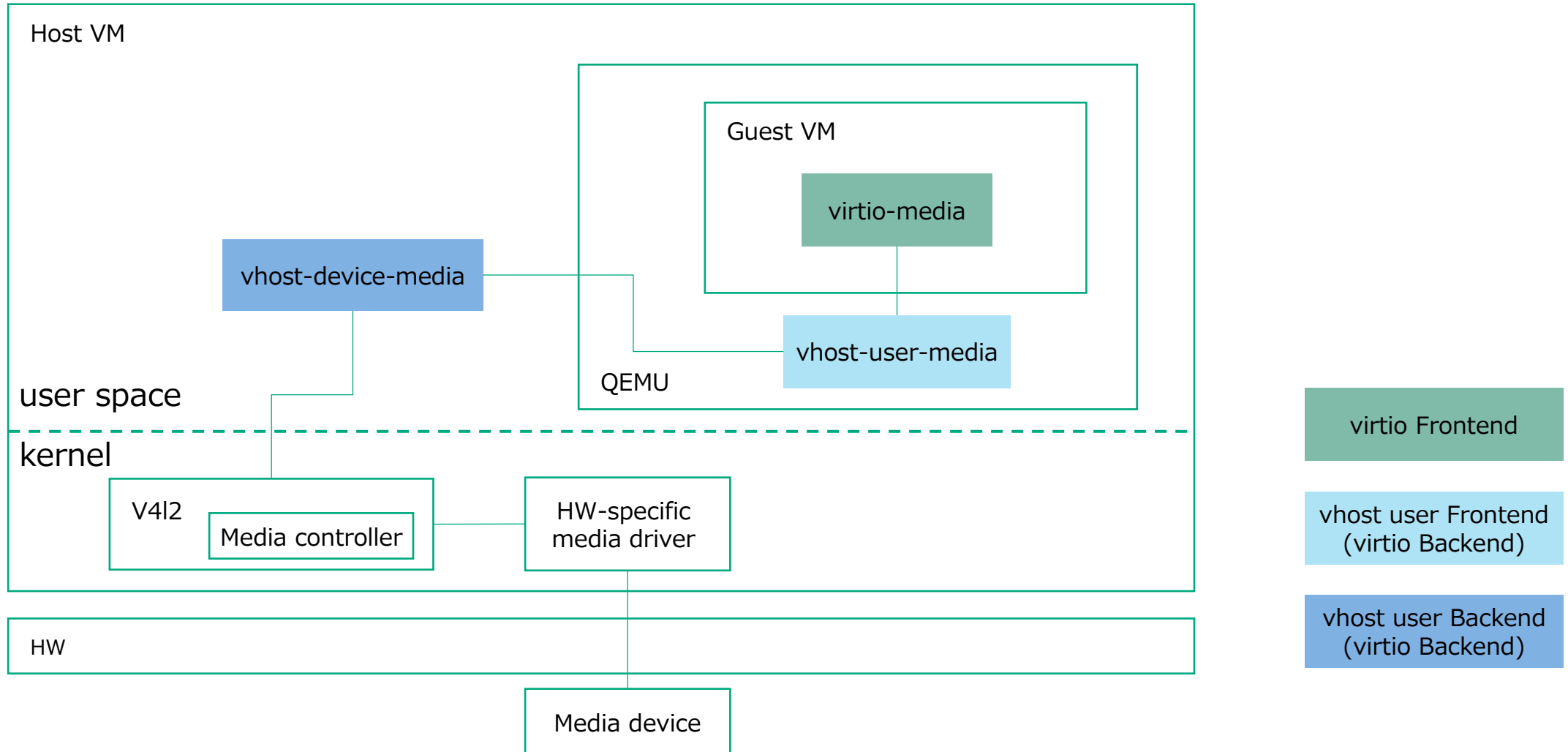
Software component diagram focusing VirtIO RTC

> running virtio-rtc in a two-VM configuration

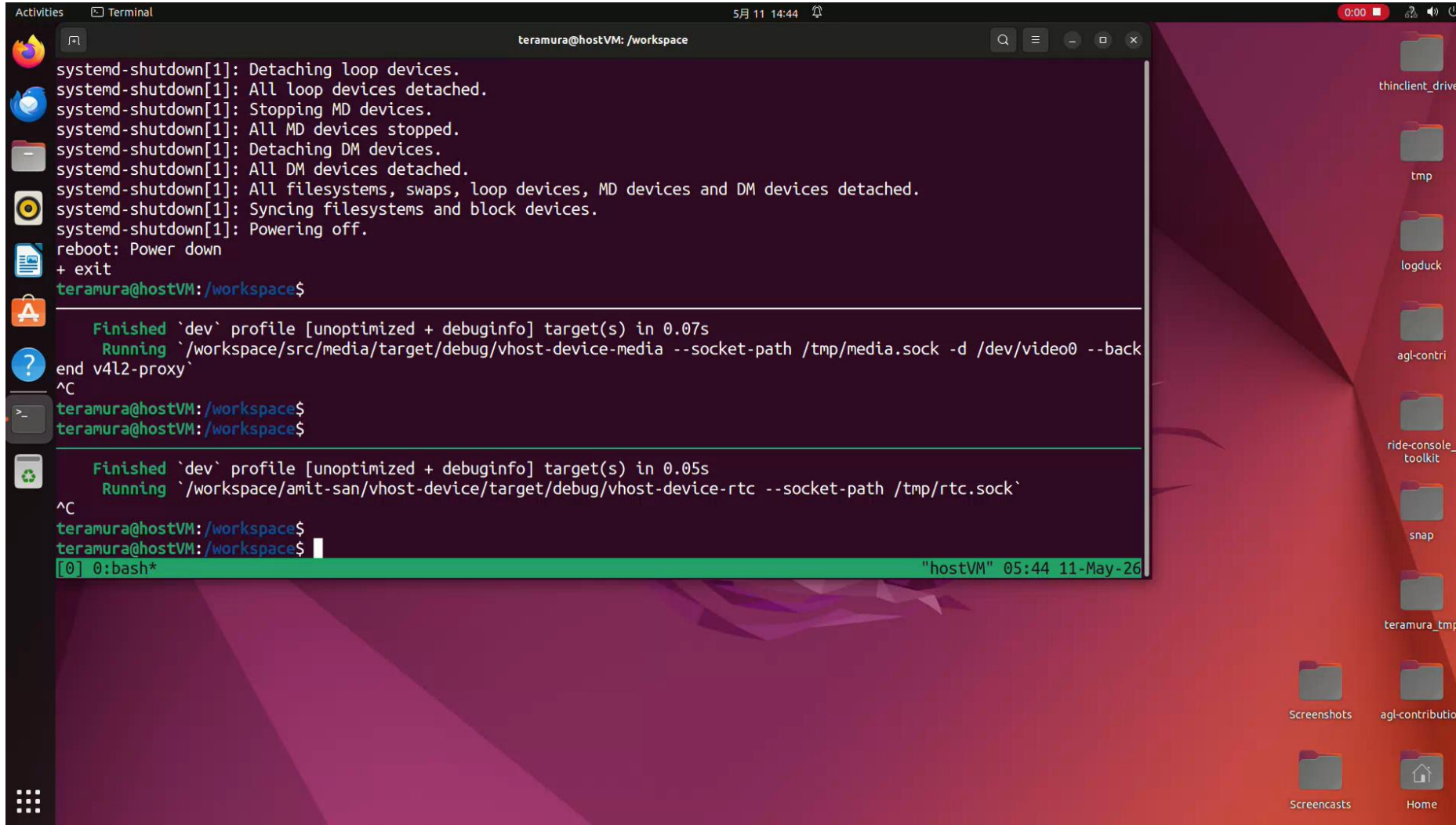


Software component diagram focusing VirtIO Media

> running virtio-media in a two-VM configuration



VirtIO RTC and Media Demo Video



HW	Desktop PC
Host VM	Ubuntu 24.04
Guest VM	AGL unagi21.0.0

virtio-gpu Issue Found During virtio-rtc Integration

- > Discovered during suspend resume testing for alarm feature
- > Observed issues
 - > GUI did not stop during suspend
 - > Screen became black after resume
- > Root cause
 - > Suspend / resume support for the virtio-gpu driver has only recently been defined (by OASIS VirtIO 1.4)
 - > Implementation is not yet available
 - > Rendering not stopped on suspend
 - > GPU not reinitialized on resume
- > Impact
 - > Display continues to render even when the system is suspended
 - > Display failure after resume

Future work

- > Scheduled for release on Vimba

Thank you

Appendix

> RTC

> How to build

> [vhost-device/vhost-device-rtc at main · rust-vmm/vhost-device](#)

> How to run socket

> `$ cargo run -- --socket-path <socket path>`

> Media

> How to build

> [vhost-device/vhost-device-media at virtio-media · aesteve-rh/vhost-device](#)

> How to run socket

> `$ cargo run -- --socket-path <socket path> -d <video device> --backend v4l2-proxy`

> VirtIO driver in Linux kernel

> RTC (Since it has already been merged into master, I will backport it to AGL)

> Apply below patches

[Add virtio_rtc module and related changes \[LWN.net\]](#)

[\[PATCH v7 1/4\] virtio_rtc: Add module and driver core — Linux Kernel](#)

[\[PATCH v7 2/4\] virtio_rtc: Add PTP clocks](#)

[\[PATCH v7 3/4\] virtio_rtc: Add Arm Generic Timer cross-timestamping](#)

[Linux-Kernel Archive: \[PATCH v5 4/4\] virtio_rtc: Add RTC class driver](#)

> Media (It has not yet been merged into the master branch and is currently in the form of a patch)

> apply below a patch

[LKML: Alexandre Courbot: \[PATCH v3\] media: add virtio-media driver](#)

> QEMU

> Base version : 10.2.0 (The criteria for the Scarth Gap in Vimba, the next UCB)

> Apply below patches

[\[PATCH\] virtio: Add vhost-user-rtc and vhost-user-rtc-pci](#)