THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference

How AGL SoDeV Accelerates the Future of Mobility Through Open-Source Collaboration

Yuichi Kusakabe
Chief Architect
Honda Motor R&D Co., Ltd.

About myself(Yuichi Kusakabe)

HONDA
The Power of Dreams

- 2005/4 Automotive Tier1 in Japan
  - Renesas H8S AVN power control(Non-OS)
  - Panasonic UniPhier USB-Audio(Fastest in the industry with HDD transfer function)
- 2009/12 Working together with in Japan OEM
  - Engaged in the R&D of SoC for next-generation IVI and multimedia OS
- 2011/1 Paradigm shift to use OSS
  - Renesas RMA1 Linux-PF Project Leader, **AGL Board Member**
  - Renesas R-Car M2/H3/M3 Fast Boot, Java, HTML5 browser
- 2020/6 Join Honda Motor in IVI software development team
  - Qualcomm SA8155P AAOS based IVI Lead Architect
- 2024/4 Launched Honda **OSPO(Open Source Program Office)**
  - SDV(Software Defined Vehicle) SW Chief Architect and **OSPO Tech Lead**
  - **Xen Board Member, ELISA and Zephyr Member**

SPDX LITE

OPENCHAIN 2.1 ISO/IEC 5230:2020 Open Source License Compliance Program

OPENCHAIN 1.1 ISO/IEC DIS 18974 Open Source Security Assurance Program

Xen Project

ELISA Enabling Linux in Safety Applications

Zephyr®

AUTOMOTIVE GRADE LINUX

Agenda

- **Background to software changes in the automotive industry**
- **Why open source is important**
- **SDV Reference PF AGL -SoDeV-**
- **Conclusion**

Background to software changes in the automotive industry

Background to software changes in the automotive industry

- Shift 1: Hardware → Software-Defined Vehicle
  - Vehicle value shifts from hardware to software
- Shift 2: Close → Open Standards & Consortiums
  - Siloed, proprietary systems limit scalability
- Shift 3: Linear Supply Chain → Ecosystem
  - The traditional OEM and Tier 1 hierarchy is changing

Shift 1: Hardware → Software-Defined Vehicle

AUTOMOTIVE GRADE LINUX
OSSNA2026

- Vehicle value shifts from hardware to software
  - Functions are defined and updated by software
  - Over-the-air (OTA) updates enable continuous evolution
  - The vehicle becomes a software-driven platform

Shift 1: Hardware → Software-Defined Vehicle

AUTOMOTIVE GRADE LINUX
OSSNA2026

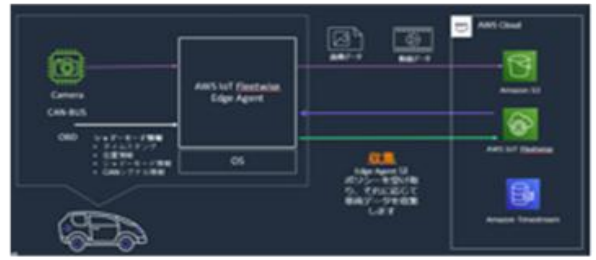
Virtual, digital twin development

Out-Car

Virtual development environment

Data utilization base

Data driven development



Utilize standard architecture/ technology/tool

Hardware-less development/ verification environment

End To End environment including other ECUs

In-Car Core ECU

OTA Manager

Cloud Connector

Safety App

API

Vehicle Data Lake

Service

Container

Container

CP

CP

OS

Adaptive-Core

Classic-Core

UX app, high-speed evolution

Real time, safety

Soft area: provision area of most of the services

Hard area: with knowledge and actual results

Enhance development efficiency by automation

Data driven development/ analysis environment

Cross-generation SW development (shift to one-branch)

https://sched.co/1eKvI

Shift 2: Close → Open Standards & Consortiums

AUTOMOTIVE GRADE LINUX
OSSNA2026

- Siloed, proprietary systems limit scalability
  - Common data models and APIs enable interoperability
  - Industry consortiums (e.g., JasPar) drive standardization
  - OEMs focus on differentiating software and services

Shift 2: Close → Open Standards & Consortiums

AUTOMOTIVE GRADE LINUX
OSSNA2026

What is JASPAR?

- Japan Automotive Software Platform and Architecture
- Established in 2004
- Focuses on the standardization of automotive software and E/E architecture

Mission

Through standardization activities in Japanese automobile industry,
we will realize the following.

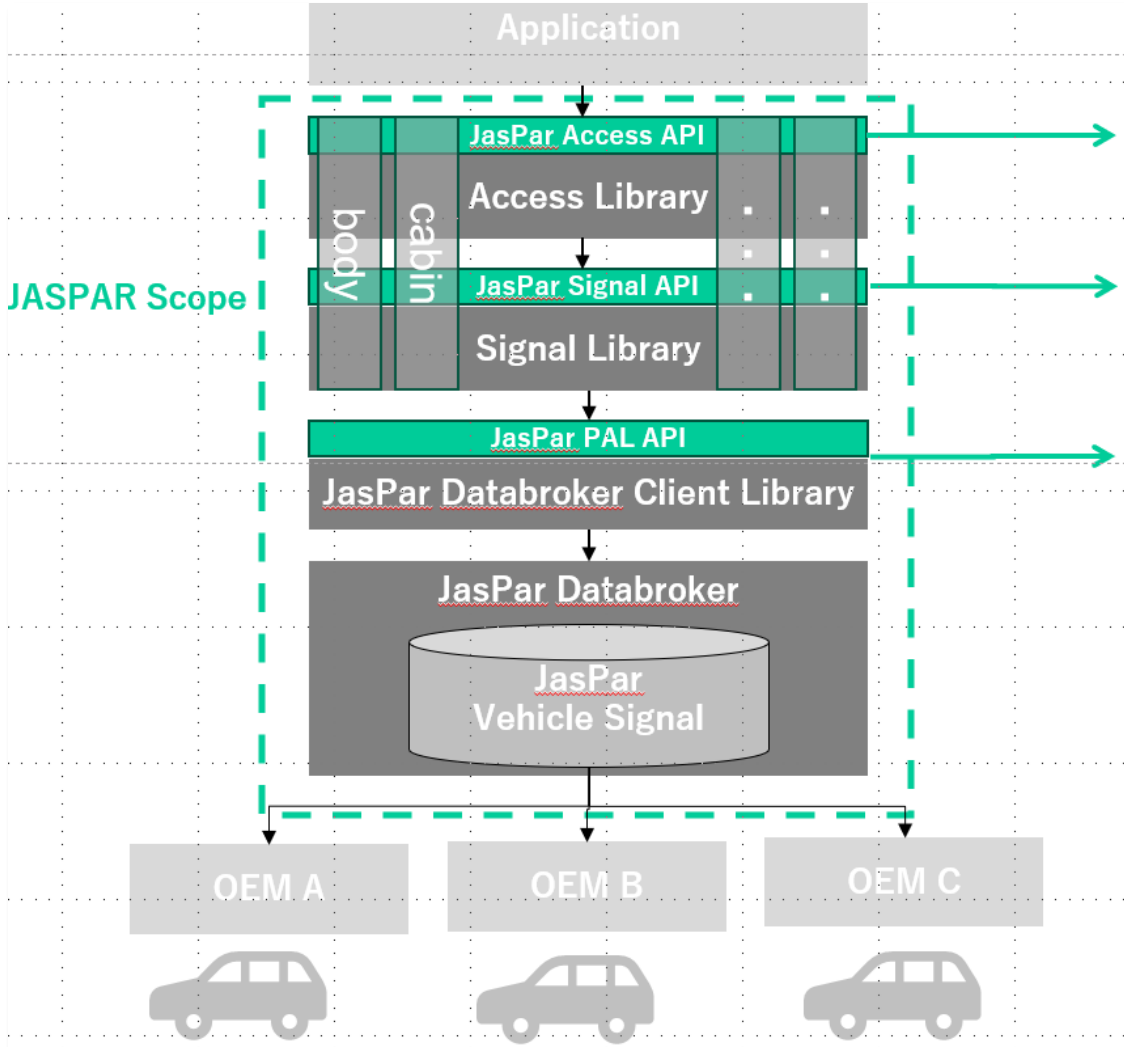
- Development of common foundation for automotive industry
- Increased development productivity
- Promoting technological development

JASPAR promotes the standardization of vehicle systems in Japanese automotive industry

JasPar © JASPAR All Rights Reserved.

https://eclipsesdv.org/wp-content/uploads/2025/12/Yasushi-Ando-API-Standardization-in-JASPAR-for-SDV.pdf

Shift 2: Close → Open Standards & Consortiums



Access API (Under discussion)

- Abstracted APIs (Abstract data structure (vehicle signals) and provide more user-friendly APIs)

Signal API

- Access Vehicle Signals
- Vehicle Signals are created based on COVESA VSS (standardize signals for vehicle sensor/actuator)

PAL API

- PAL: Platform Abstraction Layer
- Abstract OEMs Middleware/OS

JasPar © 2025 JASPAR All Rights Reserved.

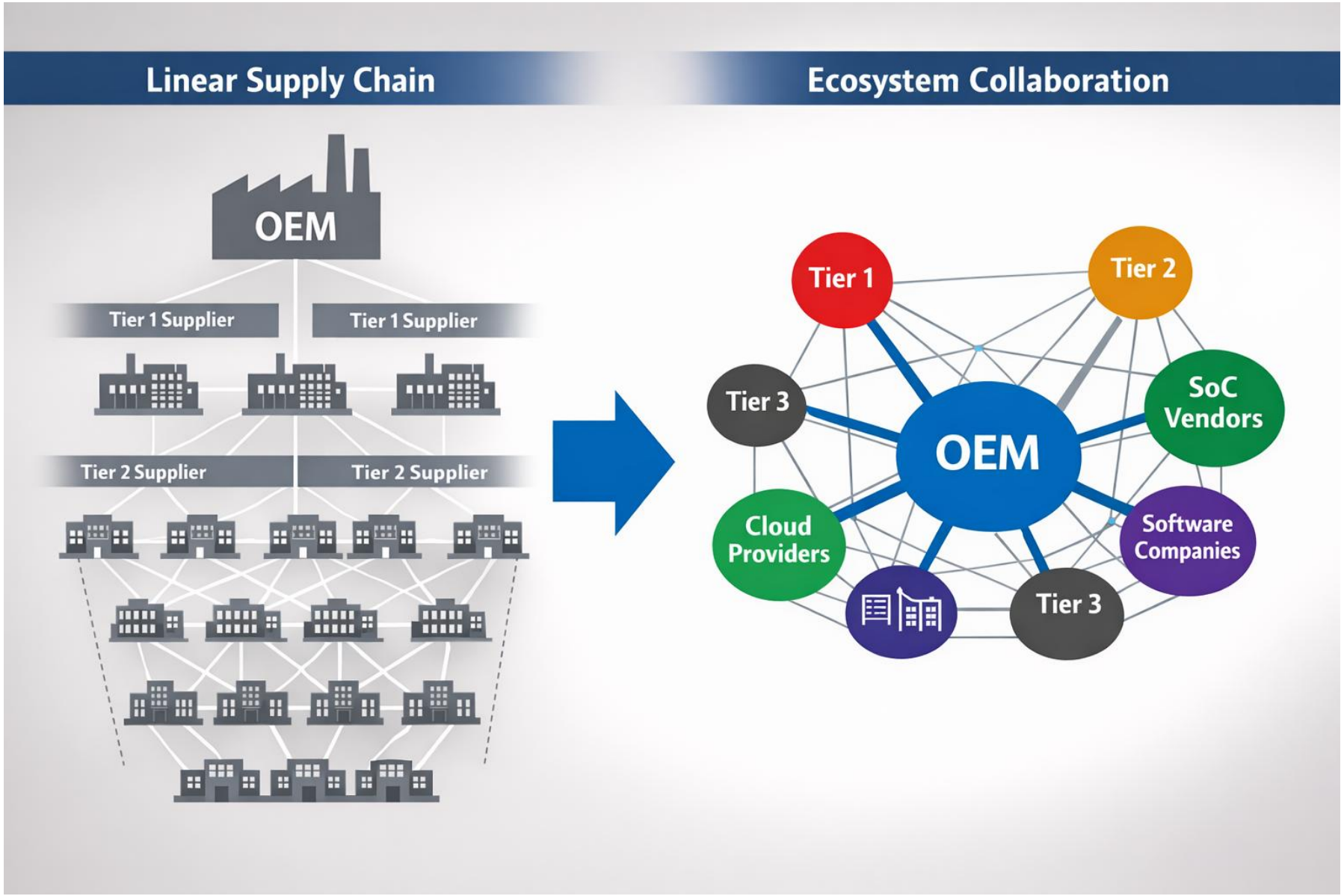
https://eclipsesdv.org/wp-content/uploads/2025/12/Yasushi-Ando-API-Standardization-in-JASPAR-for-SDV.pdf

Shift 3: Linear Supply Chain → Ecosystem

AUTOMOTIVE GRADE LINUX
OSSNA2026

- The traditional OEM and Tier 1 hierarchy is changing
  - Semiconductor, cloud, and software players gain importance
  - Value is created through cross-industry collaboration
  - OEMs increasingly act as ecosystem orchestrators

Shift 3: Linear Supply Chain → Ecosystem



Why open source is important

Lots of OSS communities involved in SDV

Automotive Grade Linux (AGL)

https://www.linuxfoundation.org/projects

Why open source is important

AUTOMOTIVE GRADE LINUX
OSSNA2026

- Shift 1: Hardware → Software-Defined Vehicle
  - Vehicle value shifts from hardware to software
- Shift 2: Close → Open Standards & Consortiums
  - Siloed, proprietary systems limit scalability
- Shift 3: Linear Supply Chain → Ecosystem
  - The traditional OEM and Tier 1 hierarchy is changing

Shift 1: Hardware → Software-Defined Vehicle

AUTOMOTIVE GRADE LINUX
OSSNA2026

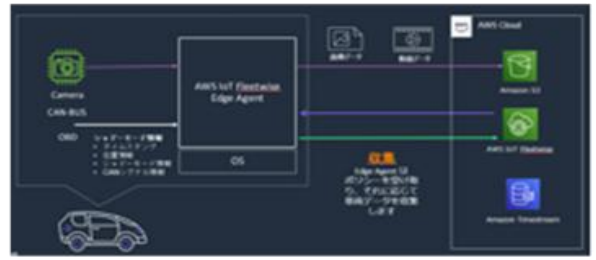
Virtual, digital twin development

Out-Car

Virtual development environment

Data utilization base

Data driven development



Utilize standard architecture/ technology/tool

Hardware-less development/ verification environment

End To End environment including other ECUs

In-Car Core ECU

OTA Manager

Cloud Connector

Safety App

API

Vehicle Data Lake

Service

Container

Container

CP

CP

OS

Adaptive-Core

Classic-Core

UX app, high-speed evolution

Real time, safety

Soft area: provision area of most of the services

Hard area: with knowledge and actual results

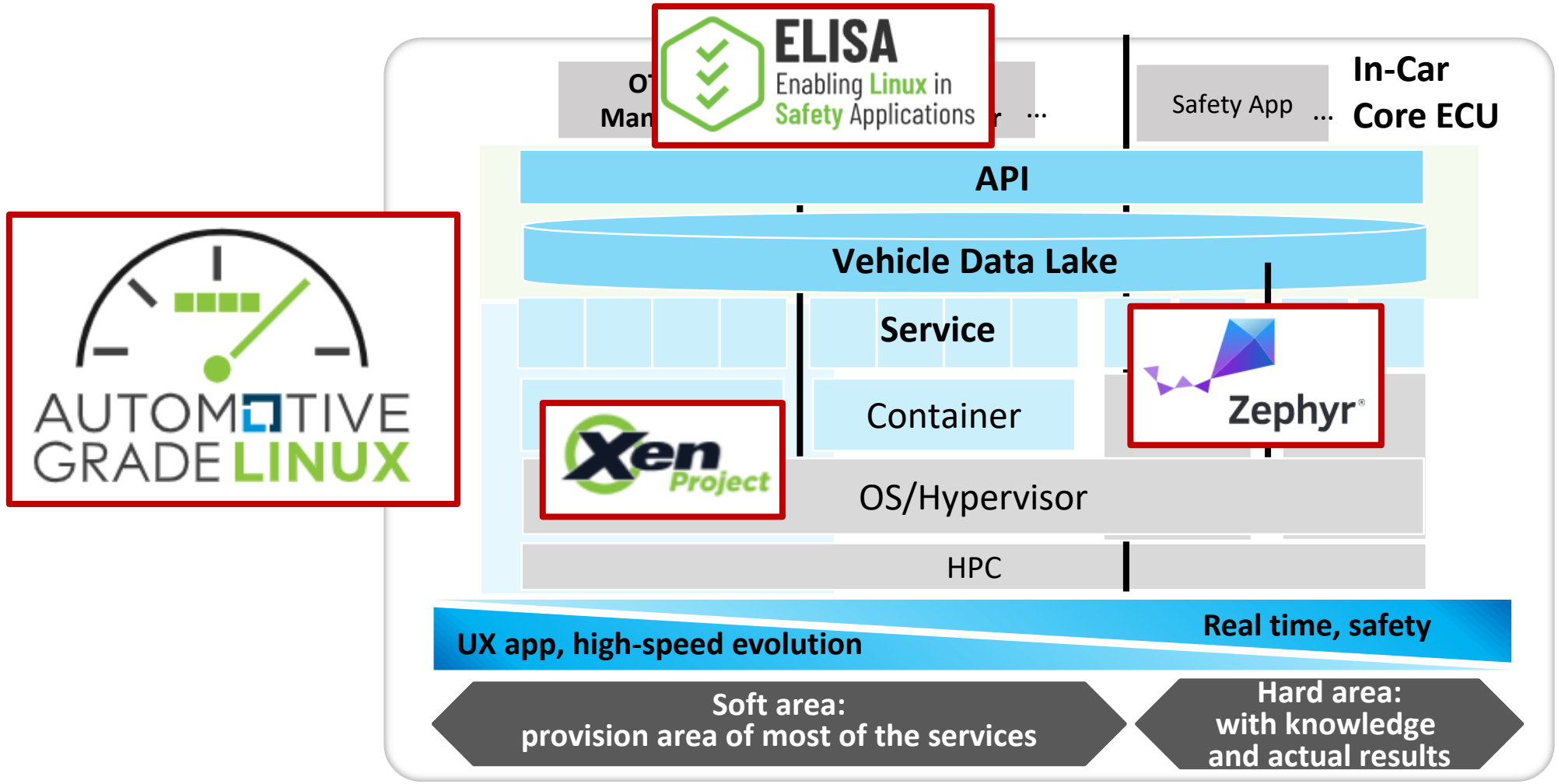
Enhance development efficiency by automation

Data driven development/ analysis environment

Cross-generation SW development (shift to one-branch)

https://sched.co/1eKvI

Shift 1: Hardware → Software-Defined Vehicle



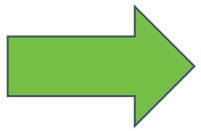
What is Automotive Grade Linux?

Automotive Grade Linux is a collaborative, open source project that brings together automakers, suppliers, and technology companies for the purpose of building Linux-based, open source software platforms for automotive applications that can serve as de facto industry standards.

AGL address all software in the vehicle: infotainment, instrument cluster, heads-up-display (HUD), telematics, connected car, advanced driver assistance systems (ADAS), functional safety, and autonomous driving.

Adopting a shared platform across the industry reduces fragmentation and allows automakers and suppliers to reuse the same code base, which leads to rapid innovation and faster time-to-market for new products.

AGL is a Linux Foundation project and its goals are as follows:



- Build a single platform for the entire industry
- Develop 70 to 80% of the starting point for a production project
- Reduce fragmentation by combining the best of open source
- Develop an ecosystem of developers, suppliers, and expertise that all use a single platform

You can find additional overview information on the "About Automotive Grade Linux" page. You can find information on the AGL Unified Code Base on the "Unified Code Base" page.

https://docs.automotivelinux.org/en/salmon/

AGL SDV Expert Group

- VirtIO available in UCB for hypervisor use cases
- Non-hypervisor (loopback) use cases complete for Quillback
- Unified HMI – Virtual display used by different ECUs based on VirtIO (Panasonic)
- ***SDV reference PF under development***

Charter

The AGL Software Defined Vehicle (SDV) Expert Group (SDV-EG) (formerly known as Virtualization and Containers Expert Group (EG-VIRT)) is responsible to design and implement virtualization solutions for AGL. Containers, Hypervisors (both based on Virtualization Extensions and TrustZone) and any other virtualization solution for x86/ARM are considered to be of interest for this Expert Group. The AGL Unified Code Base (UCB) supports the KVM hypervisor on the Renesas RCar M3 platform.

The SDV-VIRT expert group has been focused on defining the Virtualization platform architecture of AGL since 2018. One result of the work is this white paper

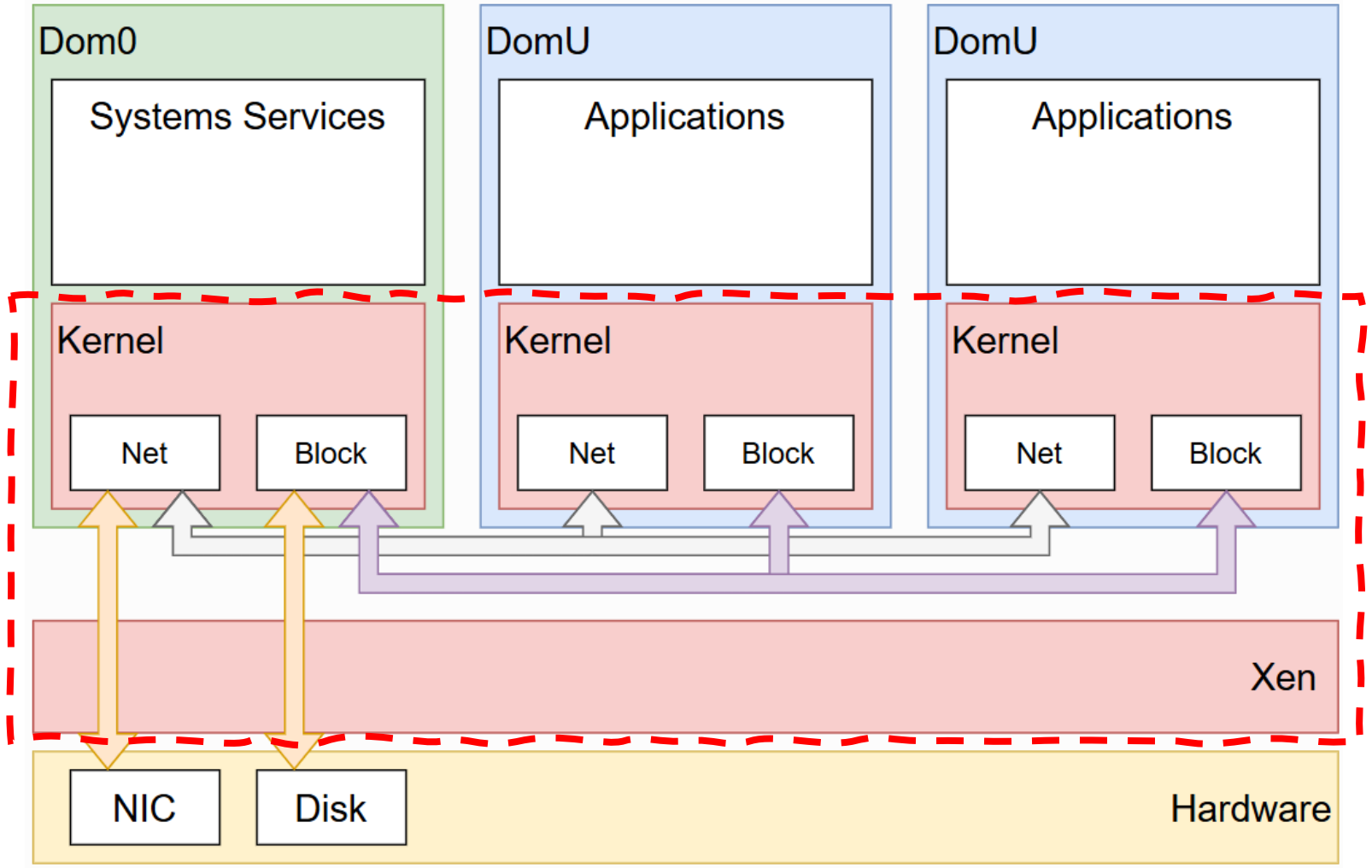
https://lf-automotivelinux.atlassian.net/wiki/spaces/VE/overview

Type1 Hypervisor: Xen

Xen is an open source, bare metal hypervisor. It runs as the most privileged piece of software, and shares the resources of the hardware between virtual machines.

In Xen terminology, there are domains, commonly abbreviated to dom, which are identified by their numeric domid.

When Xen boots, dom0 is automatically started as well. Dom0 is a virtual machine which, by default, is granted full permissions



https://xenbits.xen.org/docs/latest/admin-guide/introduction.html

Type1 Hypervisor: Xen

Functional safety

With sponsorships from the likes of AMD, we're on a mission to align Xen with top safety standards, including ISO 26262 ASIL D and IEC 61508 SIL 3.

Initiatives like the MISRA C course for community members and striving for MISRA C compliance in the Xen upstream codebase underscore our dedication to ensuring Xen's role in safety-critical environments.

Xen is not just a technology; it's a vision coming to life, shaping the future of embedded systems, and redefining what's possible.

https://xenproject.org/projects/embedded-and-automotive/

The Final Phase of Xen Safety: Solving Coverage and Residual Gaps - Stefano Stabellini, AMD

Click here to remove from My Schedule.

Wednesday May 20, 2026 11:55am - 12:35pm CDT

200C (Level Two)

AMD, in collaboration with the Xen community, continues to advance efforts to make the Xen hypervisor safety-certifiable to ISO 26262 ASIL D and IEC 61508 SIL 3. The project has progressed from Safety Concept Approval toward the final certification phase.

This presentation will share practical lessons learned, including how we structure requirements and architecture specification documents to make them easier to review for Open Source experts. It will describe the tools and processes we use to maintain end-to-end traceability and explain how we leverage GitLab to automate requirements-based testing and verification pipelines.

We will also address the remaining challenges on the path to completion, including code coverage and FMEA. In particular, we will explain why achieving comprehensive code coverage is uniquely challenging for a widely used Open Source project such as Xen and outline the strategies we are applying to meet 100% code coverage targets.

Finally, we will describe our approach to FMEA (Failure Mode and Effects Analysis) and how it evolved to better align with existing upstream Xen failure-handling practices.

Speakers

Stefano Stabellini

Fellow, AMD

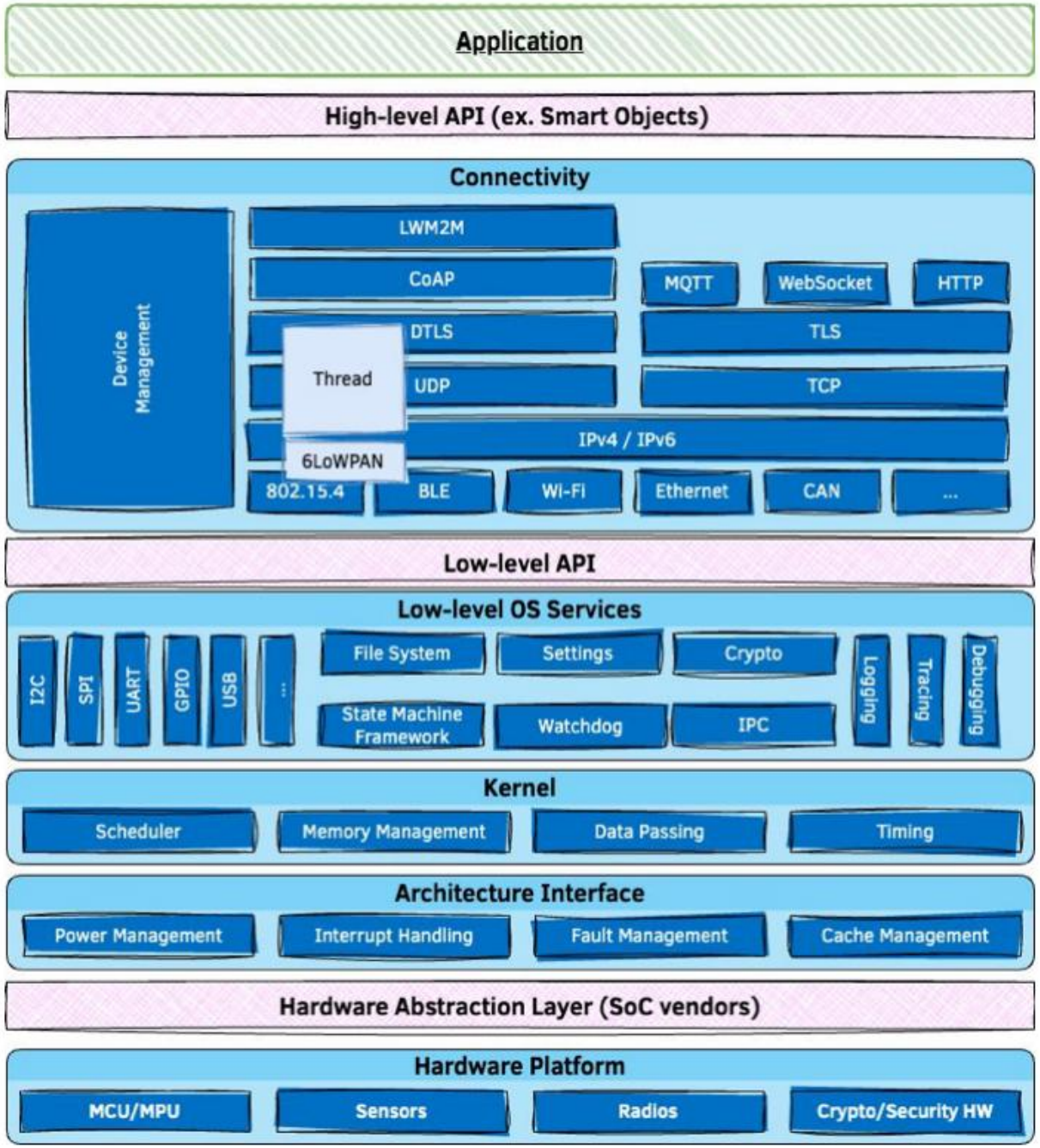
Stefano Stabellini is a Fellow at AMD, where he leads system software architecture and the virtualization team. Previously, he developed a virtualization-based security solution for containers and authored several security articles. Stefano has been involved in Xen development since... Read More →

Safety-critical Software

https://sched.co/2JQxS

Architecture

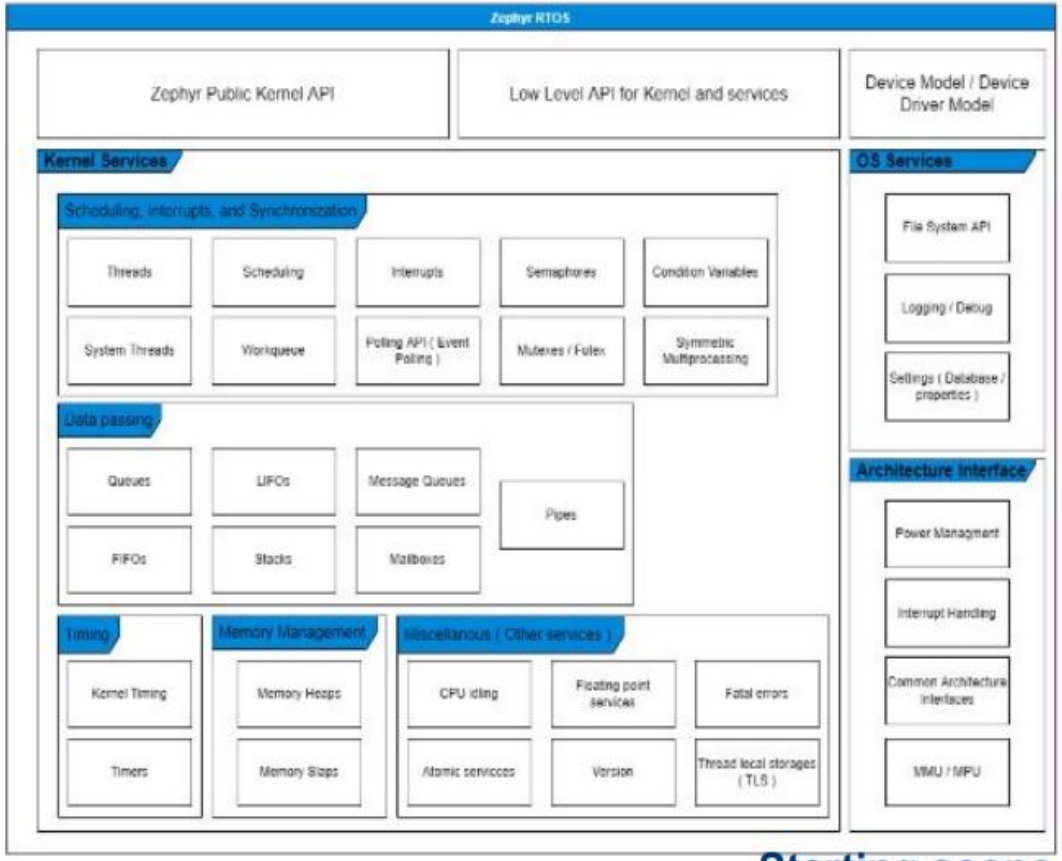
- **Lightweight kernel & supporting drivers and services**
- **Portable, secure, power-efficient**
- **Highly connected**
  - Bluetooth 5.0 & BLE
  - Wi-Fi, Ethernet, CANbus, ...
  - IoT protocols: CoAP, LwM2M, MQTT, OpenThread, ...
  - USB & USB-C



https://zephyrproject.org/

Zephyr Initial Certification Focus

- Start with a limited scope of kernel and interfaces
- Initial target is IEC 61508 SIL 3 / SC 3 (IEC 61508-3, 7.4.2.12, Route 3s)
- Option for 26262 certification has been included in contract with certification authority should there be sufficient member interest



Starting scope

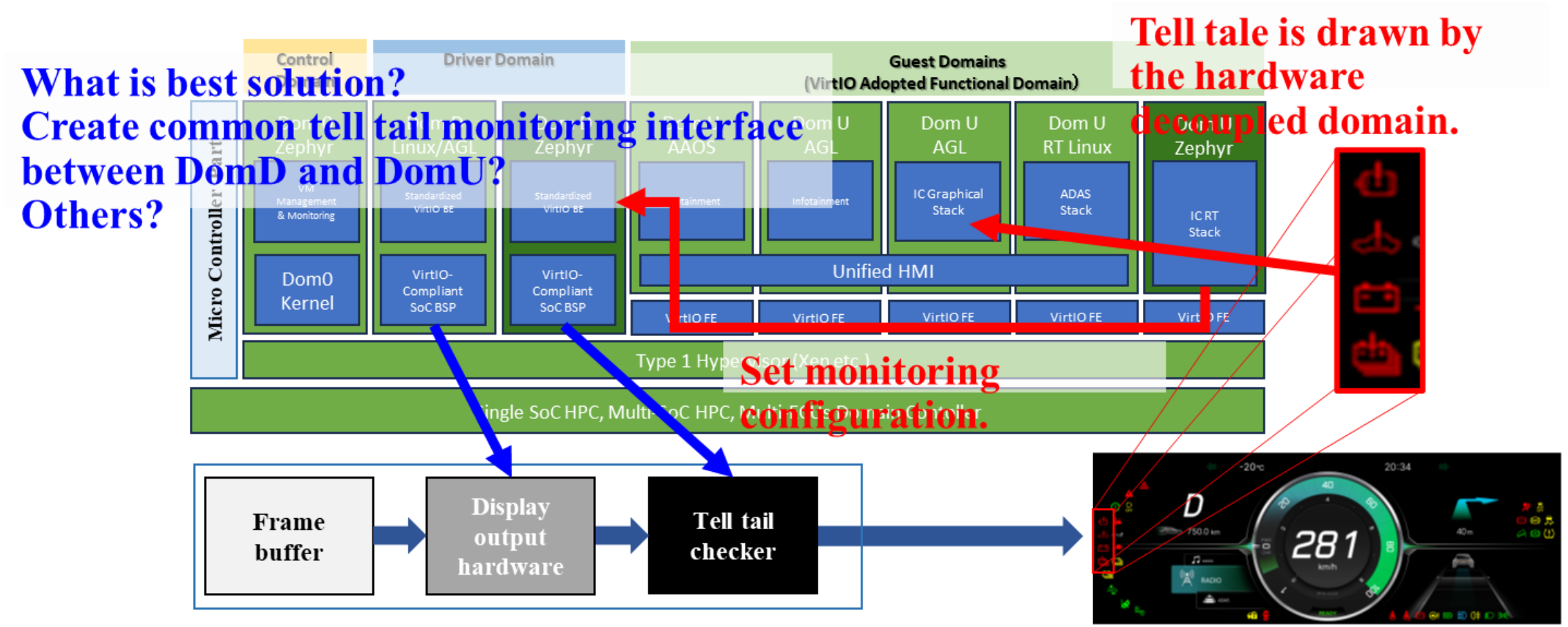
Scope can be **extended** to include **additional components** with associated **requirements** and **traceability** as determined by the safety committee

https://sched.co/1urXZ

The Enabling Linux In Safety Applications (ELISA)

ELISA
Enabling Linux in Safety Applications

- How to achieve safety?
  - Example use case: Tell tale for instrument cluster.



https://lf-automotivelinux.atlassian.net/wiki/spaces/VE/pages/602996743/AGL+SDV+Reference+Platfrom+-+SoDeV

Why open source is important

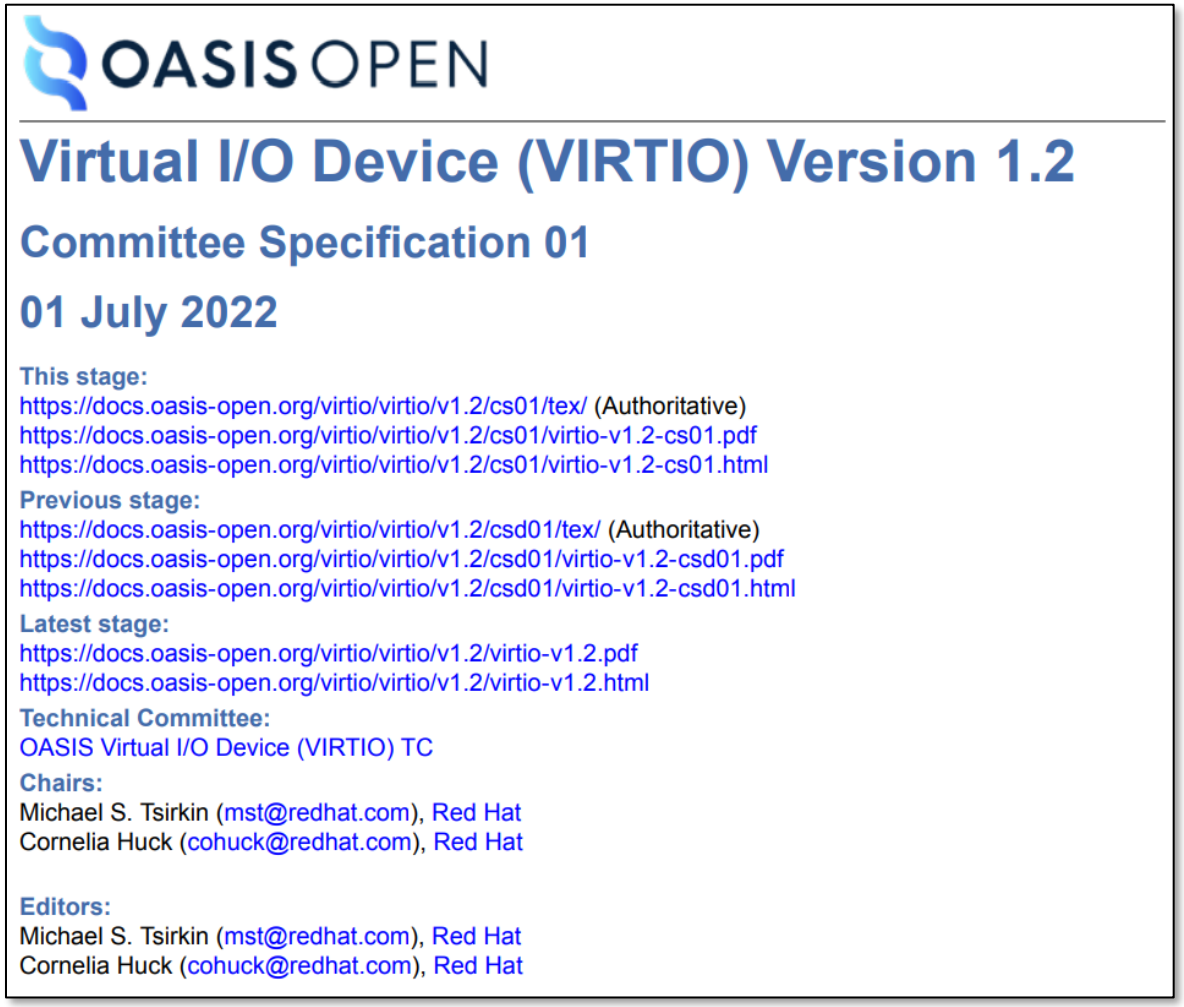
AUTOMOTIVE GRADE LINUX
OSSNA2026

- Shift 1: Hardware → Software-Defined Vehicle
  - Vehicle value shifts from hardware to software
- Shift 2: Close → Open Standards & Consortiums
  - Siloed, proprietary systems limit scalability
- Shift 3: Linear Supply Chain → Ecosystem
  - The traditional OEM and Tier 1 hierarchy is changing

Open Standards & Consortiums(VirtIO)

-What is VirtIO (a standardized device virtualization technology)
VirtIO is an open standard for device I/O in virtual environments. It allows the same drivers and device models to work across different operating systems, hypervisors, and hardware. The specification is developed and maintained as an international standard by OASIS.

-OASIS activities: defining and maintaining the VirtIO specification
OASIS operates the Virtual I/O Device (VIRTIO) Technical Committee.This committee publishes and updates the VirtIO specifications (such as VirtIO v1.1 and v1.2) as official Committee Specifications.This ensures an open, neutral standard without vendor lock-in.



OASIS OPEN

Virtual I/O Device (VIRTIO) Version 1.2

Committee Specification 01

01 July 2022

This stage:
https://docs.oasis-open.org/virtio/virtio/v1.2/cs01/tex/ (Authoritative)
https://docs.oasis-open.org/virtio/virtio/v1.2/cs01/virtio-v1.2-cs01.pdf
https://docs.oasis-open.org/virtio/virtio/v1.2/cs01/virtio-v1.2-cs01.html

Previous stage:
https://docs.oasis-open.org/virtio/virtio/v1.2/csd01/tex/ (Authoritative)
https://docs.oasis-open.org/virtio/virtio/v1.2/csd01/virtio-v1.2-csd01.pdf
https://docs.oasis-open.org/virtio/virtio/v1.2/csd01/virtio-v1.2-csd01.html

Latest stage:
https://docs.oasis-open.org/virtio/virtio/v1.2/virtio-v1.2.pdf
https://docs.oasis-open.org/virtio/virtio/v1.2/virtio-v1.2.html

Technical Committee:
OASIS Virtual I/O Device (VIRTIO) TC

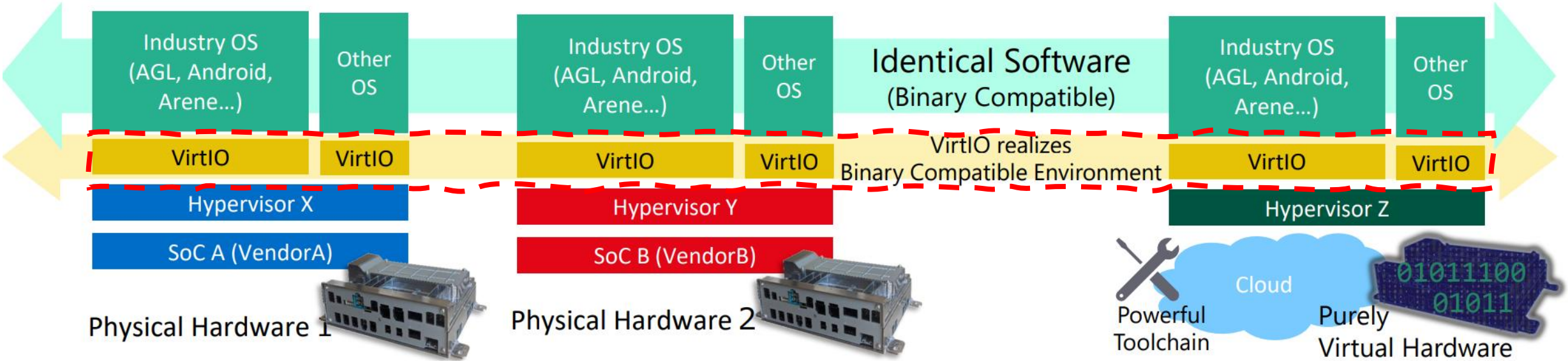
Chairs:
Michael S. Tsirkin (mst@redhat.com), Red Hat
Cornelia Huck (cohuck@redhat.com), Red Hat

Editors:
Michael S. Tsirkin (mst@redhat.com), Red Hat
Cornelia Huck (cohuck@redhat.com), Red Hat

https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=virtio#overview

VirtIO support for Xen

ARM: Add “tech preview” implementation for VirtIO. Xen now includes full support for VirtIO on embedded systems, on ARM, for the virtio-mmio transport, allowing a wide range of VirtIO devices to be supported. This includes front-end support in Linux, toolstack (libxl/xl) and dom0less support, and a userspace backend. Currently, the following stand-alone backends are available and have been tested: virtio-disk, virtio-net, i2c, and gpio.



https://sched.co/1uFjM

Panasonic Automotive Systems secures industry-wide endorsement for its initiative to standardize VirtIO device virtualization technology

～Development of VirtIO-compliant in-vehicle software platform completed～

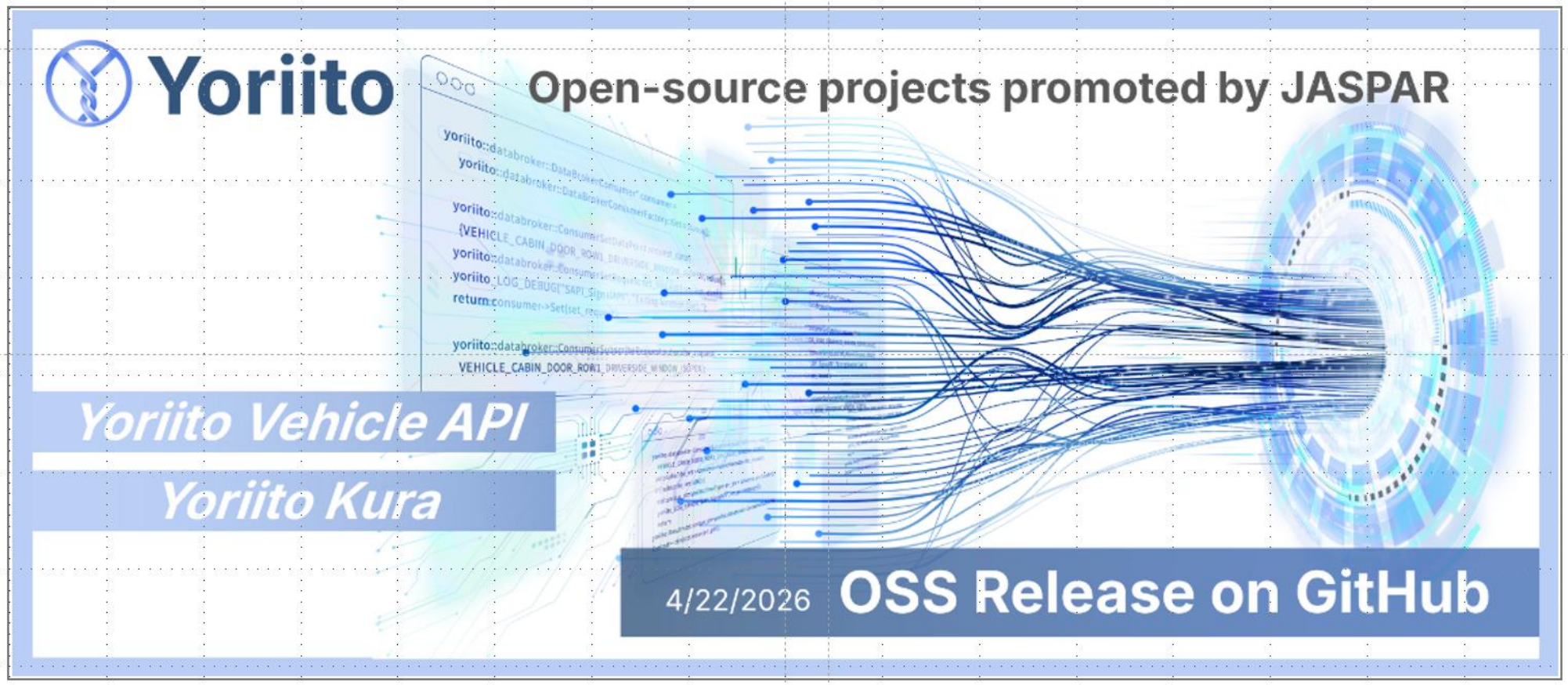
Yokohama, Japan, February 24, 2026 – Panasonic Automotive Systems Co., Ltd. (Headquarters: Yokohama, Kanagawa, Japan; President: Masashi Nagayasu) has been actively promoting the global adoption and industry-wide standardization of VirtIO—an open-source device virtualization technology advanced by industry groups such as AGL*1, Android™, OASIS*2, and SOAFEE*3—for automotive applications. Today, our efforts have gained industry-wide endorsement from Honda Motor Co., Ltd., Mazda Motor Corporation, Mitsubishi Motors Corporation, Nissan Motor Co., Ltd., Toyota Motor Corporation, as well as related companies in the IT and other sectors. In addition, we have completed the development of a VirtIO-compliant in-vehicle software platform for CDC (Cockpit Domain Controller). Moving forward, we will continue to develop and promote VirtIO-compliant automotive software to contribute to innovation in vehicle development.

https://news.panasonic.com/global/press/en260224-2

Open Standards & Consortiums(JASPAR)

AUTOMOTIVE GRADE LINUX
OSSNA2026

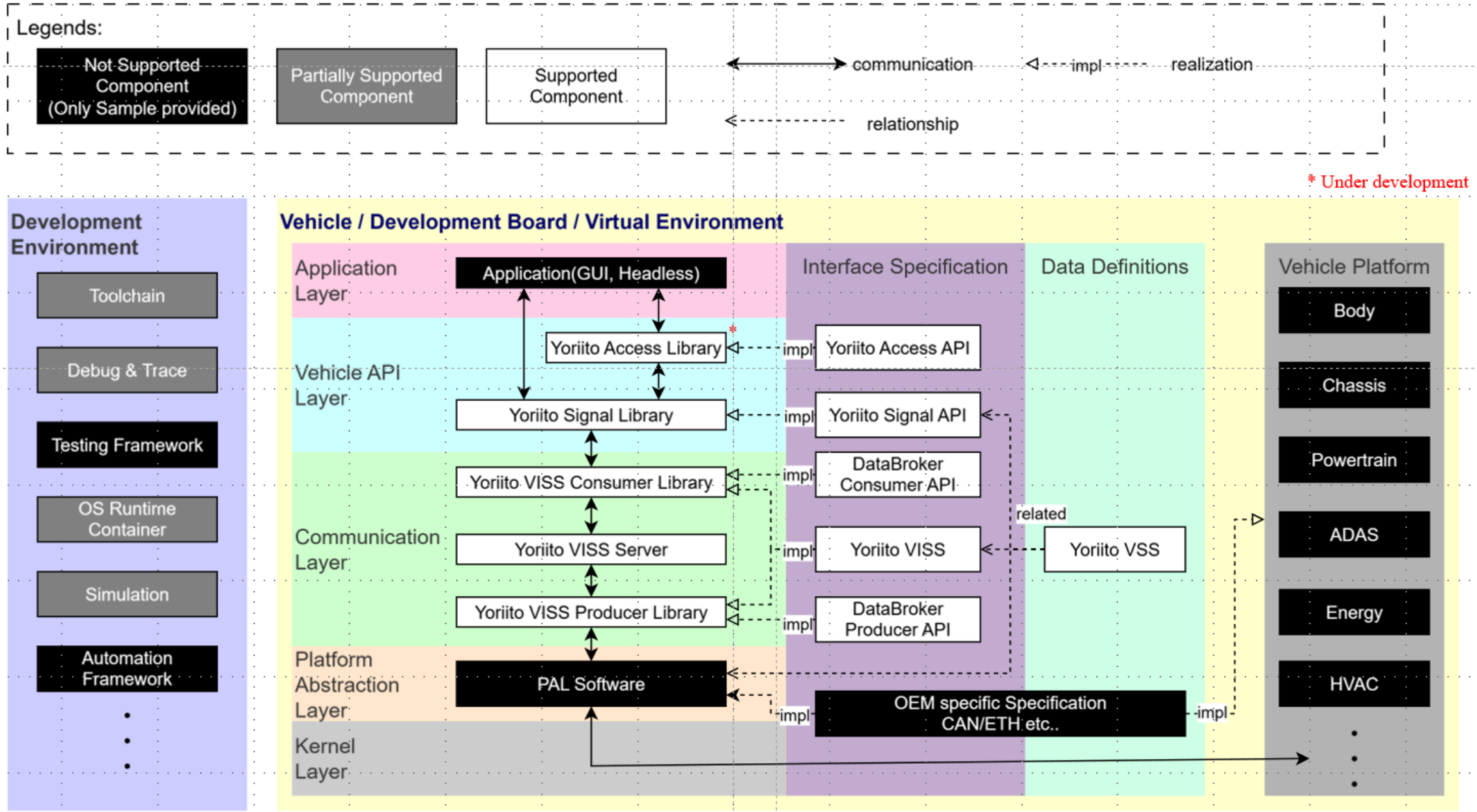
1st OSS Release !!



More open. More free. Automotive software—built by everyone

JasPar © JASPAR All Rights Reserved.

Architecture and scope



© JASPAR All Rights Reserved.

README Apache-2.0 license Security

Welcome to the Yoriito Project

This GitHub organization hosts artifacts from the activities of the JASPAR API standardization WG.

Introduction to the Yoriito Project

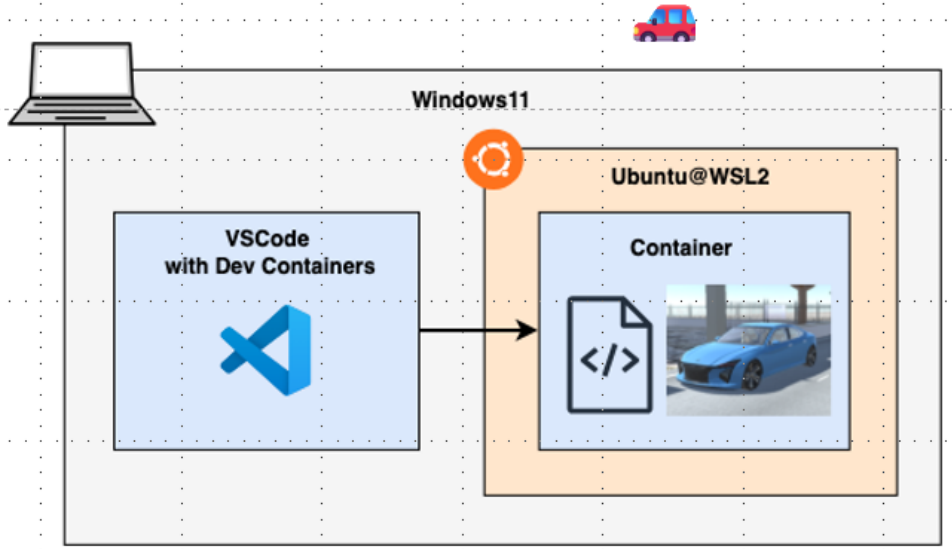
For more information, please see the Yoriito homepage.

Organization Repositories

Repository	Description	Status
yoriito	Main repository for the Yoriito project	✅ active

https://github.com/yoriito/yoriito

Build develop environment



Recommended/Tested Environment

- OS: Windows 11
- Virtualization: WSL2
- Linux distribution: Ubuntu 24.04 LTS (on WSL2)

Advanced preparation

- Install on Windows 11
  - VS Code
- Install on Ubuntu
  - Git
  - Docker
- Install in VS Code
  - Dev Containers

Clone the Repository

On Ubuntu@WSL2

```
$ git clone https://github.com/yoriito/yoriito-tutorial.git
```

```
$ cd yoriito-tutorial
$ git submodule init
$ git submodule update
```

Start the Container

In the target repository's root directory

```
$ code .
```

After VS Code starts,

- open the Command Palette with `Ctrl + Shift + P`.
- Type `Dev Containers: Rebuild and select "Dev Containers: Rebuild and Reopen in Container"`.

The container will be built and libraries will be installed.

Start the Simulator

On VSCode

```
$ cd /work/ref_pal
```

```
$ npm install
$ ./init_pal.sh
$ npx electron . --no-sandbox
```

© JASPAR All Rights Reserved.

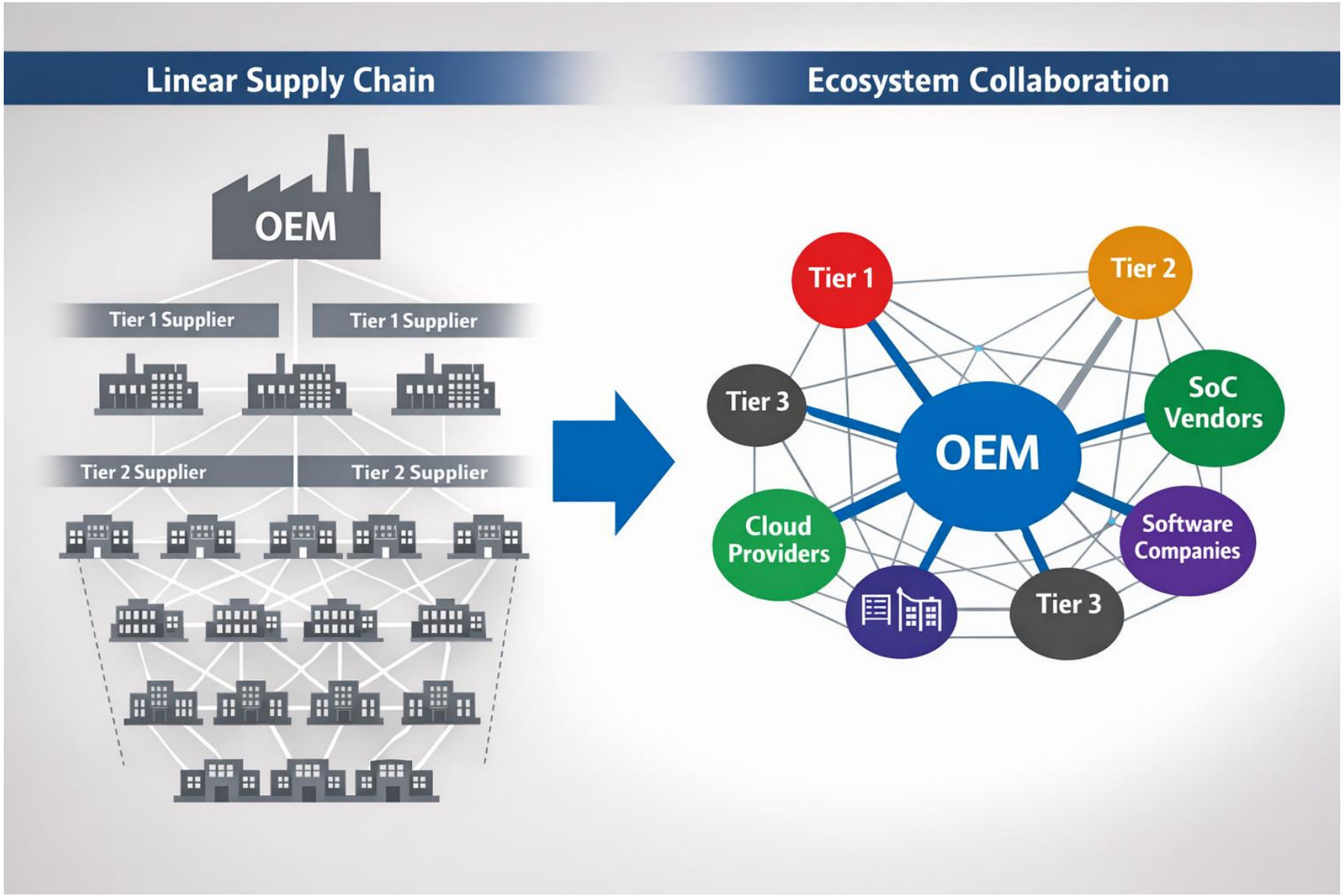
https://github.com/yoriito/yoriito

Why open source is important

AUTOMOTIVE GRADE LINUX
OSSNA2026

- Shift 1: Hardware → Software-Defined Vehicle
  - Vehicle value shifts from hardware to software
- Shift 2: Close → Open Standards & Consortiums
  - Siloed, proprietary systems limit scalability
- Shift 3: Linear Supply Chain → Ecosystem
  - The traditional OEM and Tier 1 hierarchy is changing

Shift 3: Linear Supply Chain → Ecosystem



android / platform / manifest / refs/heads/android14-qpr3-release / . / **default.xml**

blob: 13b083f073883156df7cdb23190827b56d491e7d [file] [log] [blame] [edit]

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest>

  <remote  name="aosp"
           fetch=".."
           review="https://android-review.googlesource.com/" />
  <default revision="android14-qpr3-release"
           remote="aosp"
           sync-j="4" />

  <superproject name="platform/superproject" remote="aosp" revision="android14-qpr3-release"/>
  <contactinfo bugurl="go/repo-bug" />
  <project path="build/make" name="platform/build" groups="pdk,sysui-studio" >
    <linkfile src="CleanSpec.mk" dest="build/CleanSpec.mk" />
    <linkfile src="buildspec.mk.default" dest="build/buildspec.mk.default" />
    <linkfile src="core" dest="build/core" />
    <linkfile src="envsetup.sh" dest="build/envsetup.sh" />
    <linkfile src="target" dest="build/target" />
    <linkfile src="tools" dest="build/tools" />
  </project>
  <project path="build/orchestrator" name="platform/build/orchestrator" groups="pdk" />
  <project path="build/bazel" name="platform/build/bazel" groups="pdk" >
    <linkfile src="bazel.WORKSPACE" dest="WORKSPACE" />
    <linkfile src="bazel.BUILD" dest="BUILD" />
  </project>
  <project path="build/bazel_common_rules" name="platform/build/bazel_common_rules" groups="pdk
  <project path="build/blueprint" name="platform/build/blueprint" groups="pdk,tradefed" />
```

https://android.googlesource.com/platform/manifest/+/refs/heads/android14-qpr3-release/default.xml

OSS Repository
58.9%(799/1,357)

platform/external/aac

platform/external/abi-compliance-checker

platform/external/abi-dumper

platform/external/abseil-cpp Bug: 121037047

platform/external/accessibility-test-framework Bug: 328779485

platform/external/accompanist Bug: 324277532

platform/external/actionbarsherlock

platform/external/adeb Bug: 111852163

platform/external/adhd Bug: 111264136

platform/external/adt-infra

platform/external/aehd Bug: 306906844

platform/external/aes

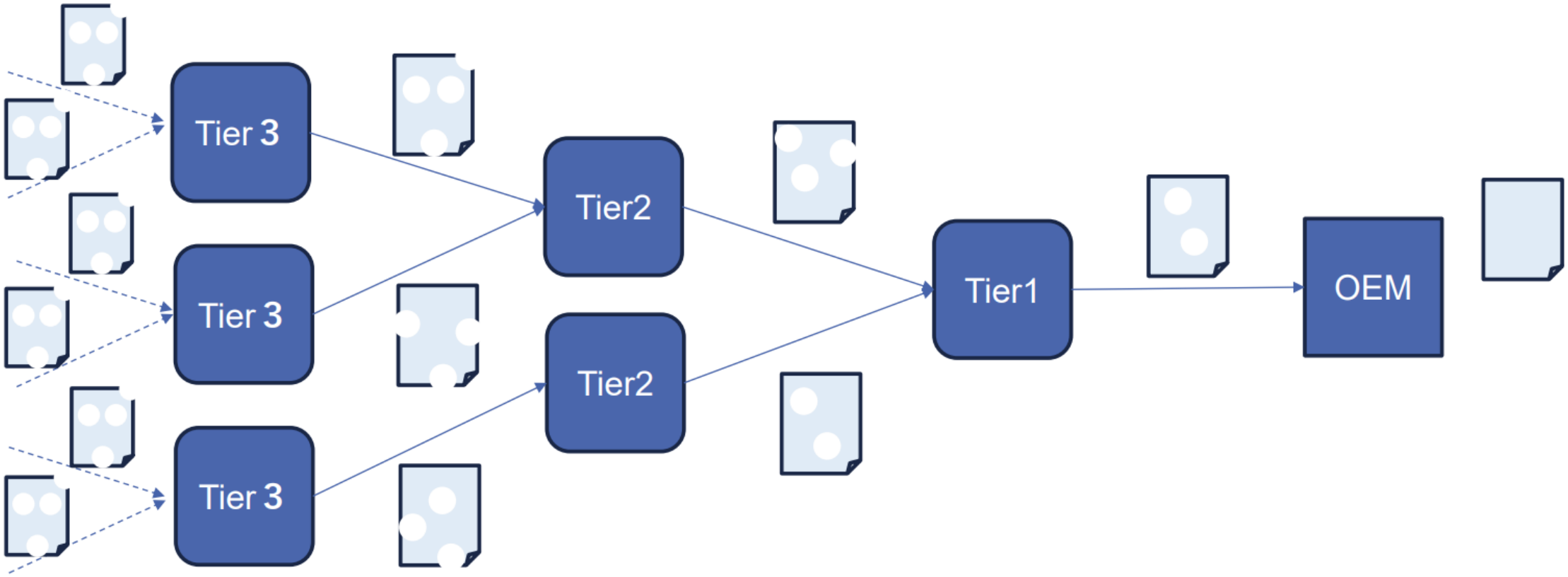
platform/external/AFLplusplus

platform/external/alac

platform/external/alsa-lib

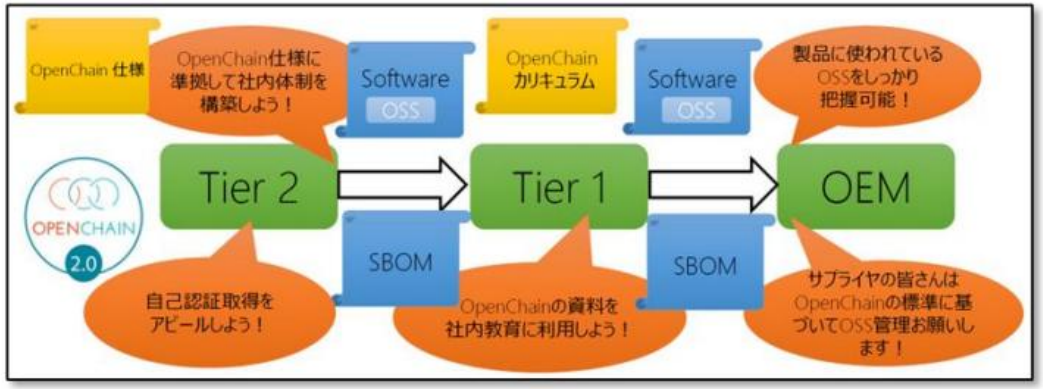
https://lfms25.sched.com/event/1urXW/understanding-and-managing-sboms-in-modern-automotive-vehicles-a-journey-yuichi-kusakabe-takashi-ninjouji-honda-motor-co-ltd

Addressing resource constraints and workflow integration



https://lfms25.sched.com/event/1urXW/understanding-and-managing-sboms-in-modern-automotive-vehicles-a-journey-yuichi-kusakabe-takashi-ninjouji-honda-motor-co-ltd

■ We adopted OpenChain's SPDX Lite (Excel) to smoothly share OSS information between companies.



項	3.1 PackageName	3.2 Package SPDX Identifier	3.3 Package Version	3.4 PackageFileName	3.7 PackageDownloadLocation	3.8 Files Analyzed	3.11 PackageHomePage	3.13 Concluded License	3.15 Declared License
	パッケージ名	パッケージSPDX識別子	パッケージバージョン	パッケージファイル名	パッケージダウンロード位置（入手先）	解析したファイル（手作業の場合false）	ホームページ（OSS開発コミュニティサイト）	結論されたライセンス	宣言されたライセンス
	LibXML2		2.99	libxml2-2.9.9.tar.gz	ftp://xmlsoft.org/libxml2/	FALSE	http://xmlsoft.org/	MIT	MIT
	PCRE		8.4343	pcre-8.43.tar.gz	ftp://ftp.pcre.org/pub/pcre/	FALSE	https://www.pcre.org/	BSD-3-Clause	BSD-3-Clause
	SQLite		3300100.tartar	sqlite-autoconf-3300100.tar.gz	https://www.sqlite.org/download.html	FALSE	https://www.sqlite.org/index.html	その他（ライセンス名を記載）	その他（ライセンス名を記載）
	Zlib (1.2.3)		1.2.11	zlib-1.2.11.tar.gz	https://www.zlib.net/	FALSE	https://www.zlib.net/	Zlib	Zlib
	cURL (7.41.0)		7.66.0	curl-7.66.0.tar.bz2	https://github.com/curl/curl/releases	FALSE	https://curl.haxx.se/	MIT	MIT

G.2 Format of SPDX Lite

The SPDX Lite profile is a subset of the SPDX specification. SPDX Lite consists of mandatory fields of the Document Creation and Package Information sections and other basic information. Cardinality of each item is not changed.

The mandatory part of the SPDX document creation information section (which consists of SPDX Version, Data License, SPDX Identifier, Document Name, SPDX Document Namespace, Creator and Created) is used for keeping compatibility with SPDX tools.

The main part of the Package Information (those are Package Name, Package Version, Package File Name, Package Supplier, Package Download Location, Package Home Page, Concluded License, Declared License, Comments on License and Copyright Text) is used for exchanging license information.

https://www.meti.go.jp/shingikai/mono_info_service/sangyo_cyber/wg_seido/wg_bunyaodan/software/pdf/003_04_00.pdf
https://github.com/OpenChain-Project/Japan-WG-General/tree/master/License-Info-Exchange
https://spdx.github.io/spdx-spec/v2.3/SPDX-Lite/

Format

Annex of ISO/IEC 5942 (SPDX 2.2.1), SPDX 2.3

Spreads heet

Elements

Annex of ISO/IEC 5942 (SPDX 2.2.1), SPDX 2.3

Component
- Package Name
- Package Version
- Concluded License
- Copyright Text
- Other elements

https://lfms25.sched.com/event/1urXW/understanding-and-managing-sboms-in-modern-automotive-vehicles-a-journey-yuichi-kusakabe-takashi-ninjouji-honda-motor-co-ltd

Self-certification to “OpenChain” Process Management Standards

- Clearly documented policy & training encourage positive attitude
- OSPO, structured organizational process, and toolchain
- Well communication , with confident, and continuously Improve

Methods

Materials reviewed and advised by a self-certified Japanese company

ISO/IEC 5230

ISO/IEC 18974

- https://openchainproject.org/news/2023/12/06/honda-announces-iso5230-program
- https://openchainproject.org/news/2024/10/16/honda-announces-iso18974-program

https://lfms25.sched.com/event/1urXW/understanding-and-managing-sboms-in-modern-automotive-vehicles-a-journey-yuichi-kusakabe-takashi-ninjouji-honda-motor-co-ltd

AUTOMOTIVE GRADE LINUX

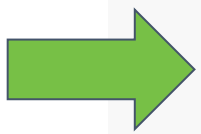
What is Automotive Grade Linux?

Automotive Grade Linux is a collaborative, open source project that brings together automakers, suppliers, and technology companies for the purpose of building Linux-based, open source software platforms for automotive applications that can serve as de facto industry standards.

AGL address all software in the vehicle: infotainment, instrument cluster, heads-up-display (HUD), telematics, connected car, advanced driver assistance systems (ADAS), functional safety, and autonomous driving.

Adopting a shared platform across the industry reduces fragmentation and allows automakers and suppliers to reuse the same code base, which leads to rapid innovation and faster time-to-market for new products.

AGL is a Linux Foundation project and its goals are as follows:



- Build a single platform for the entire industry
- Develop 70 to 80% of the starting point for a production project
- Reduce fragmentation by combining the best of open source
- Develop an ecosystem of developers, suppliers, and expertise that all use a single platform

You can find additional overview information on the "About Automotive Grade Linux" page. You can find information on the AGL Unified Code Base on the "Unified Code Base" page.

https://docs.automotivelinux.org/en/salmon/

SDV Reference PF
AGL -SoDeV-

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA
Embedded Linux Conference

AGL SDV Expert Group

- VirtIO available in UCB for hypervisor use cases
- Non-hypervisor (loopback) use cases complete for Quillback
- Unified HMI – Virtual display used by different ECUs based on VirtIO (Panasonic)
- ***SDV reference PF under development***

Charter

The AGL Software Defined Vehicle (SDV) Expert Group (SDV-EG) (formerly known as Virtualization and Containers Expert Group (EG-VIRT)) is responsible to design and implement virtualization solutions for AGL. Containers, Hypervisors (both based on Virtualization Extensions and TrustZone) and any other virtualization solution for x86/ARM are considered to be of interest for this Expert Group. The AGL Unified Code Base (UCB) supports the KVM hypervisor on the Renesas RCar M3 platform.

The SDV-VIRT expert group has been focused on defining the Virtualization platform architecture of AGL since 2018. One result of the work is this white paper

https://lf-automotivelinux.atlassian.net/wiki/spaces/VE/overview

Automotive Grade Linux Launches Open Source SoDeV Reference Platform to Accelerate Software Defined Vehicles

By AGL | December 5, 2025 | Announcements

New AGL SoDeV reference platform supports ECU consolidation, virtualization, hardware abstraction and cloud integration for next-generation vehicles.

Highlights:

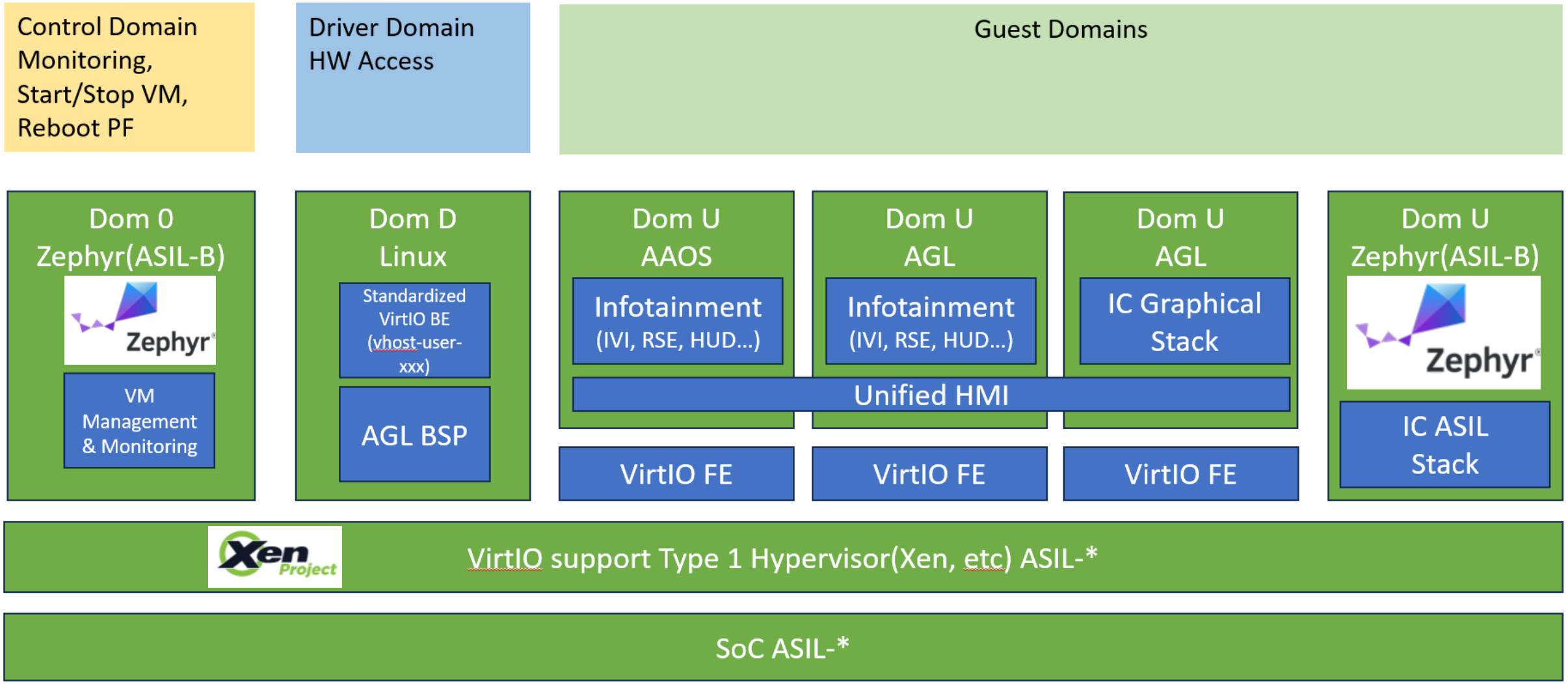
- **AGL SoDeV** is an open source reference platform for software defined vehicles enabling software-first development, independent of hardware constraints.
- The initiative is led by **Panasonic Automotive Systems, Honda, and the AGL SDV Expert Group**, with contributions from Toyota, Mazda, AISIN and Renesas.
- SoDeV integrates the **AGL Unified Code Base** with key open source projects including **Linux Containers, VirtIO, Xen, Yocto Project, Zephyr and ELISA**.
- Availability is planned for **early 2026** for virtual environments, cloud processors and automotive SoCs.

SAN FRANCISCO, December 5, 2025 – Automotive Grade Linux (AGL), a collaborative cross-industry effort developing an open source platform for connected car technologies, today announced SoDeV, a new open source SDV reference implementation that enables software-first development, decoupled from hardware constraints.

Led by Panasonic Automotive Systems, Honda and the AGL Software Defined Vehicle (SDV) Expert Group, SoDeV combines the AGL Unified Code Base (UCB) with multiple open source projects hosted by the Linux Foundation to support consolidation of ECUs, hardware abstraction using virtualization, and cloud integration for next-generation vehicles. Additional contributors to SoDeV include Toyota, Mazda, AISIN and Renesas.

https://www.automotivelinux.org/announcements/sodev/

SoDeV Architecture Overview(SoC ASIL-A/B)

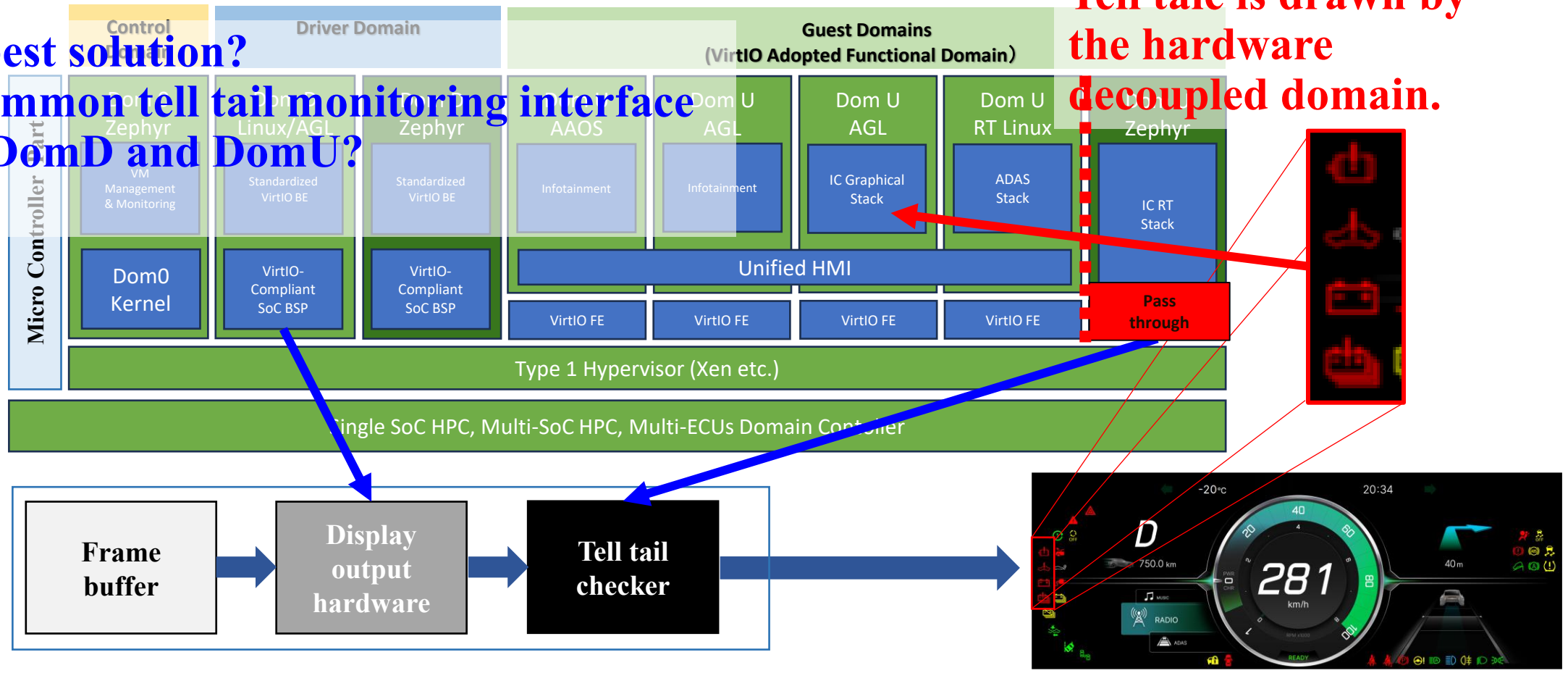


SoDeV safety use case(Zephyr pass-through)

- How to achieve safety?
  - Example use case: Tell tale for instrument cluster.

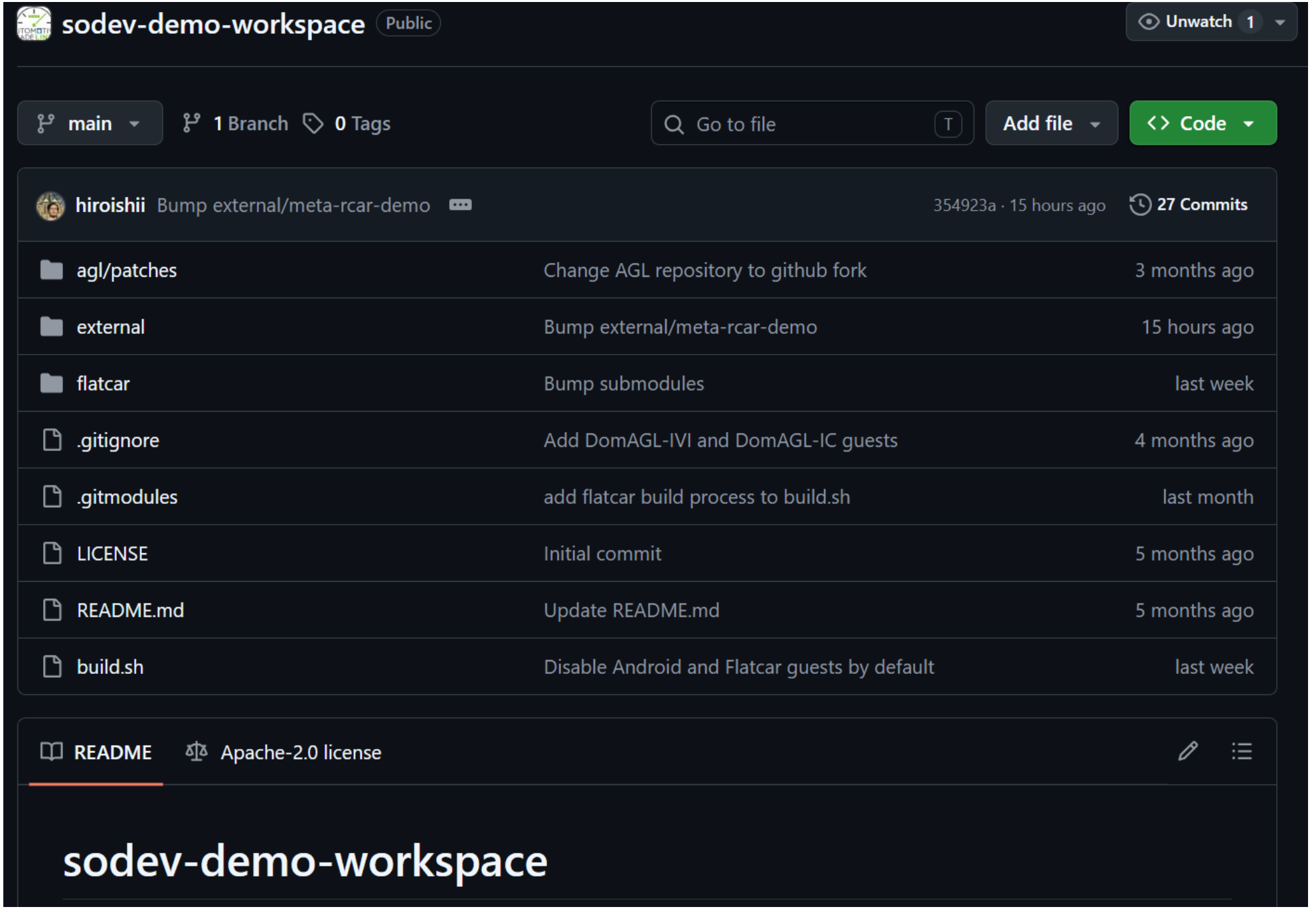
What is best solution?
Create common tell tail monitoring interface between DomD and DomU?
Others?

Tell tale is drawn by the hardware decoupled domain.



AGL releases initial version of the open source SoDeV reference platform for software-defined vehicles and welcomes EMQ, Lineo Solutions, MediaTek, VA Linux Systems Japan, and Very Good Ventures as new members

https://www.automotivelinux.org/announcements/automotive-grade-linux-releases-open-source-sodev-reference-platform-for-software-defined-vehicles-and-welcomes-five-new-members/



https://github.com/automotive-grade-linux/sodev-demo-workspace

AGL SDV Reference PF(V4H) - SoDeV -

AUTOMOTIVE GRADE LINUX
OSSNA2026

Dom D

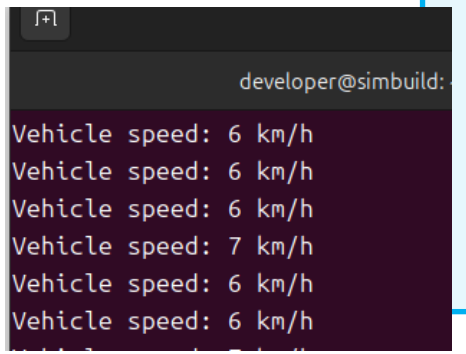
VIRTIO Backend

Guest AOSP
Honda IVI

Android System
App

Honda IVI

ASIL Apps



Guest Zephyr
ASIL Domain

Zephyr

ASIL Apps
App

Dom 0

(Control Domain)

BSP drivers
Linux kernel (BSP part)

VirtIO Frontend
Linux kernel (GKI part)

Virtual Machine
CPU | Timer
memory | GIC

Virtual Machine
CPU | Timer
memory | GIC

Virtual Machine
CPU | Timer
memory | GIC

Virtual Machine
CPU | Timer
memory | GIC

Full Virtualized
Schedular | Timer
MMU | interrupt

Xen hypervisor

Cortex®-A76 (x4)

Renesas R-Car V4H

Text Channels

japanese-language

RPi5 SoDeV立ち上げスレッド

general

all-member-meetings

off-topic

weekly-dev-call

google-summer-of-code

industry-news

autoworld-japan

embedded-world

Voice Channels

Lounge

Meeting Room

Expert Groups

system-architecture-team

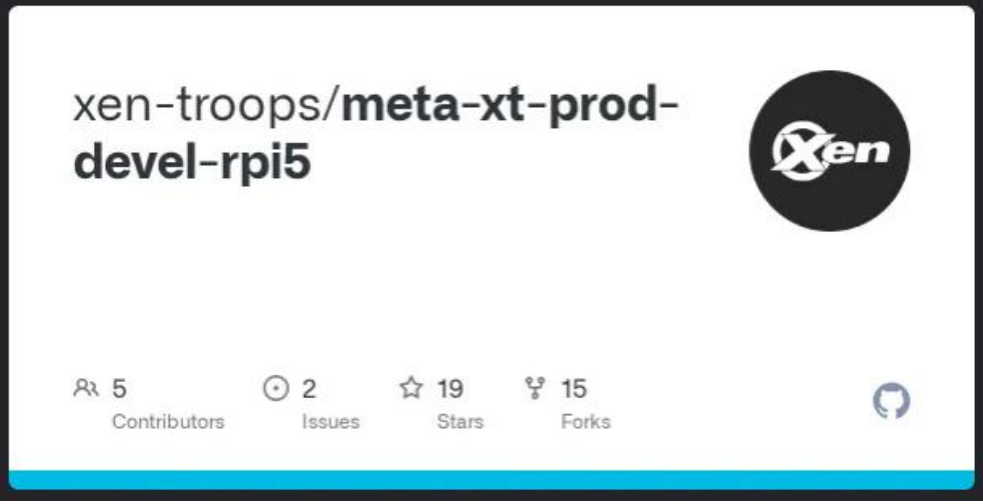
https://github.com/xen-troops/meta-xt-prod-devel-rpi5/tree/main
を使用して、下記の構成案を実現出来れば、SoDeVが実現出来るかもしれない

RPi5 (2x HDMI)
└─ Xen (Xen-Troops)
  ├─ Dom0: Zephyr
  ├─ DomD: Linux/Yocto (GPU/HDMI)
  │ ├─ Weston (kiosk-shell, DRM backend)
  │ │ ├─ HDMI-A-1: DomD UI (AGL IC)
  │ │ └─ HDMI-A-2: QEMU (SDL) = DomU GUI
  │ └─ QEMU device-model (SDL, virtio-vga-gl)
  └─ DomU: Linux/Yocto (VirtIO-GPU + AGL IVI)

GitHub

GitHub - xen-troops/meta-xt-prod-devel-rpi5

Contribute to xen-troops/meta-xt-prod-devel-rpi5 development by creating an account on GitHub.



https://discord.com/channels/1278057614349176882/1478164066945204225/1478164702436528129

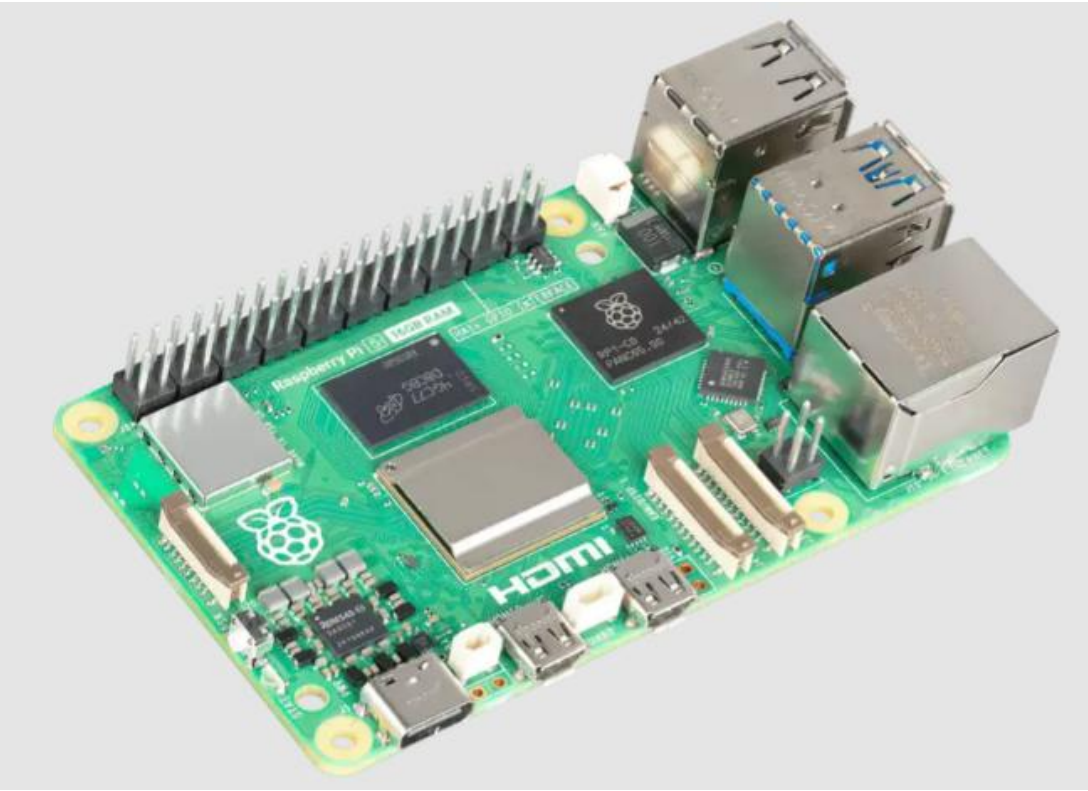
Raspberry Pi5(RPi5) HW

AUTOMOTIVE GRADE LINUX
OSSNA2026

Raspberry Pi 5 16GB RAM

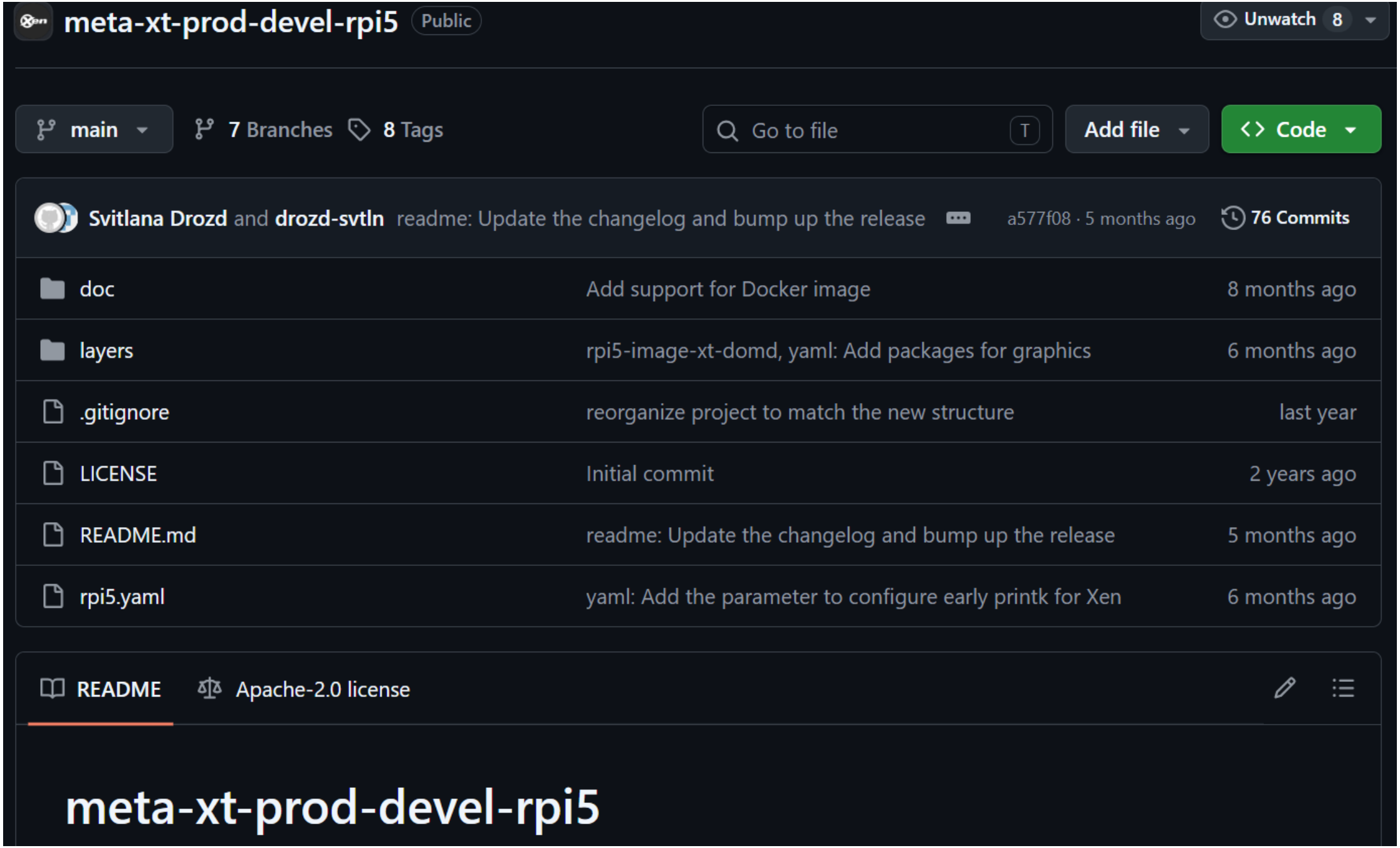
For power users and professional applications.

Raspberry Pi 5 16GB is available now for **$305**.



https://www.raspberrypi.com/products/raspberry-pi-5/

- Broadcom BCM2712 2.4GHz quad-core 64-bit Arm Cortex-A76 CPU, with cryptography extensions, 512KB per-core L2 caches and a 2MB shared L3 cache
- VideoCore VII GPU, supporting OpenGL ES 3.1, Vulkan 1.3
- Dual 4Kp60 HDMI® display output with HDR support
- 4Kp60 HEVC decoder
- LPDDR4X-4267 SDRAM (1GB, 2GB, 4GB, 8GB, and 16GB)
- Dual-band 802.11ac Wi-Fi®
- Bluetooth 5.0 / Bluetooth Low Energy (BLE)
- microSD card slot, with support for high-speed SDR104 mode
- 2 × USB 3.0 ports, supporting simultaneous 5Gbps operation
- 2 × USB 2.0 ports
- Gigabit Ethernet, with PoE+ support (requires separate PoE+ HAT)
- 2 × 4-lane MIPI camera/display transceivers
- PCIe 2.0 x1 interface for fast peripherals (requires separate M.2 HAT or other adapter)
- 5V/5A DC power via USB-C, with Power Delivery support
- Raspberry Pi standard 40-pin header
- Real-time clock (RTC), powered from external battery
- Power button



https://github.com/xen-troops/meta-xt-prod-devel-rpi5

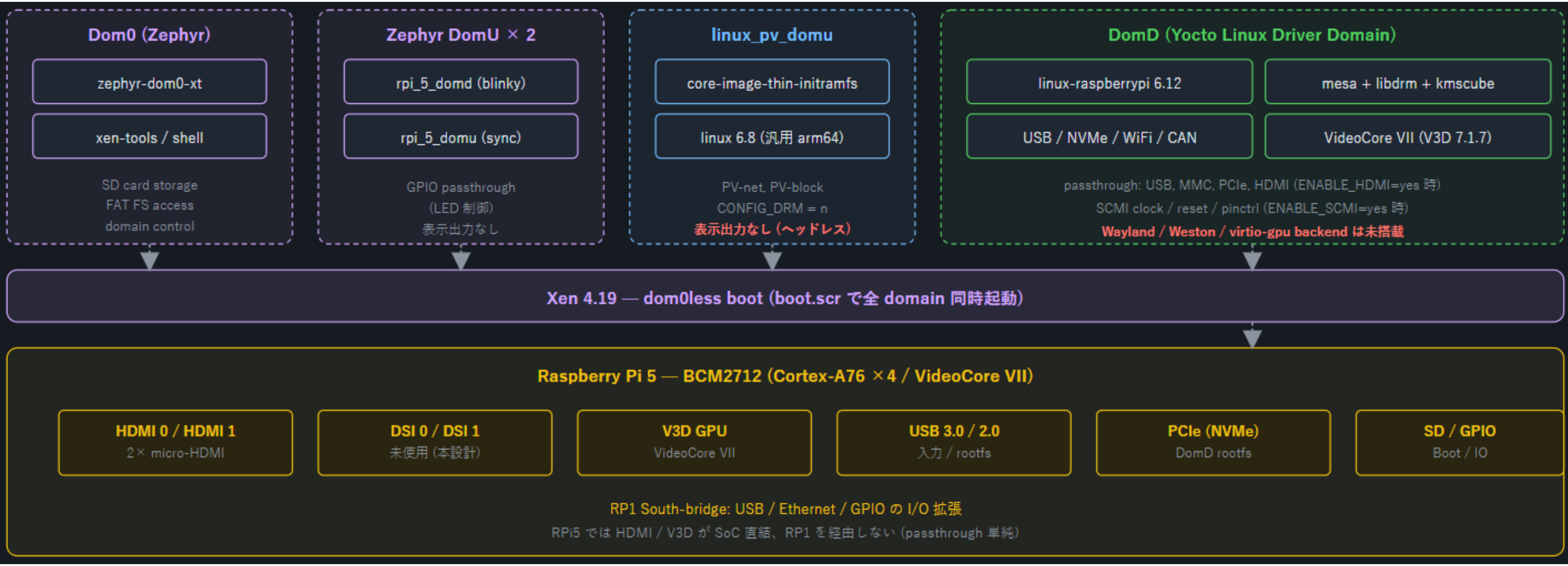
meta-xt-prod-devel-rpi5

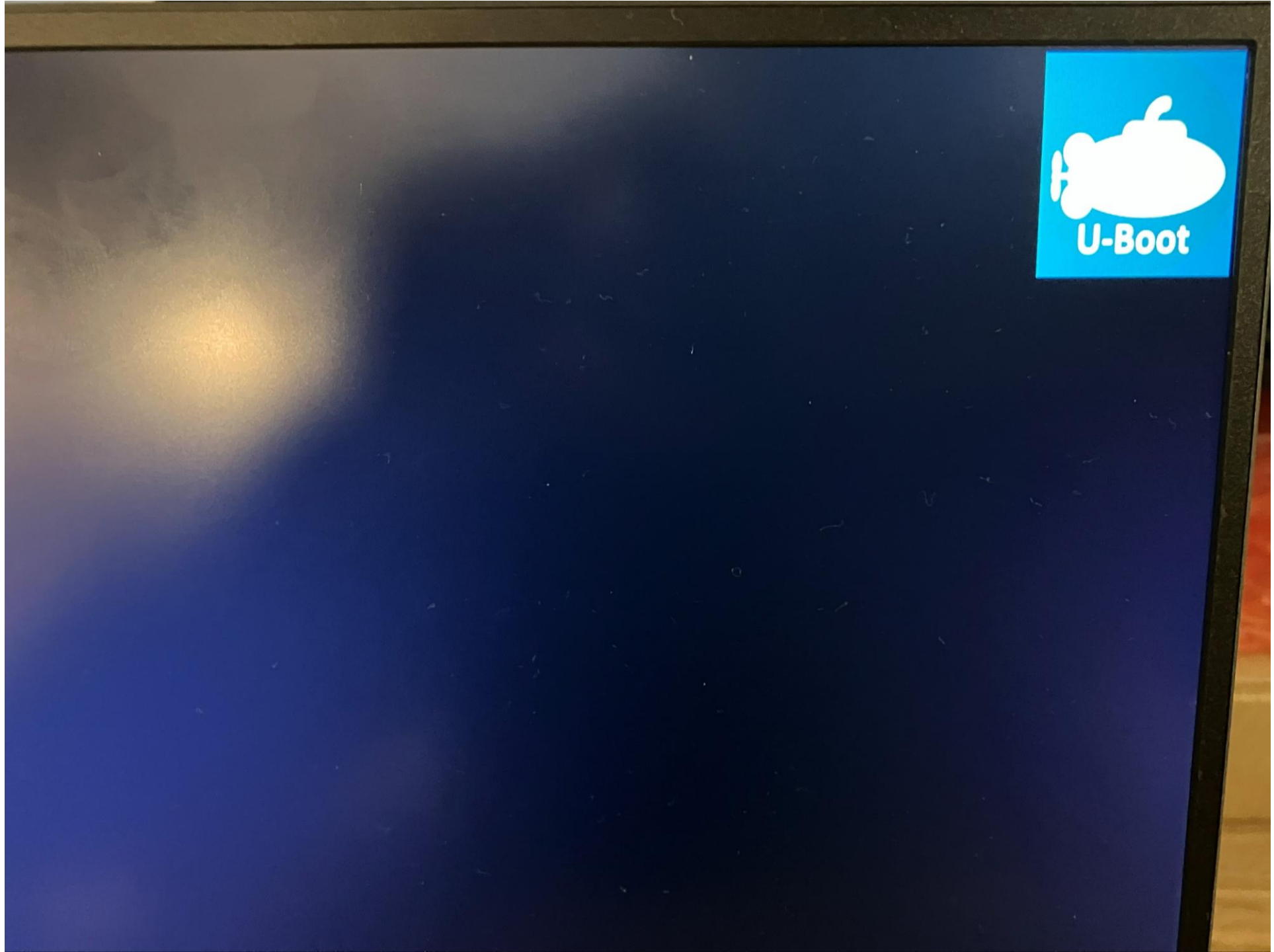
AUTOMOTIVE GRADE LINUX
OSSNA2026

```
$ moulin rpi5.yaml --help-config
usage: moulin rpi5.yaml [--MACHINE {rpi5}] [--XEN_EARLY {yes,no}] [--DOMD_ROOT {usb,nvme}] [--ENABLE_SCMI {yes,no}] [--ENABLE_WIFI
{yes,no}]
                        [--ENABLE_CAN {mcp2515-can,seeed-can-fd-hat-v2,no}] [--ENABLE_HDMI {yes,no}]

Config file description: Raspberry 5 with xen dom0less

options:
  --MACHINE {rpi5}      Raspberry Pi machines (default: rpi5)
  --XEN_EARLY {yes,no}  Enable early printk for Xen (default: yes)
  --DOMD_ROOT {usb,nvme}
                        Domd root device (default: usb)
  --ENABLE_SCMI {yes,no}
                        Enable ARM SCMI support (default: no)
  --ENABLE_WIFI {yes,no}
                        Allow wifi in domd (default: no)
  --ENABLE_CAN {mcp2515-can,seeed-can-fd-hat-v2,no}
                        Allow CAN in DomD and choose it type. The naming of options align with the overlay naming used in RPI OS (default: no)
  --ENABLE_HDMI {yes,no}
                        Allow HDMI in DomD (default: no)
```

Xen(ENABLE_HDMI) + Dom0(Zephyr) + DomD/U(Linux)

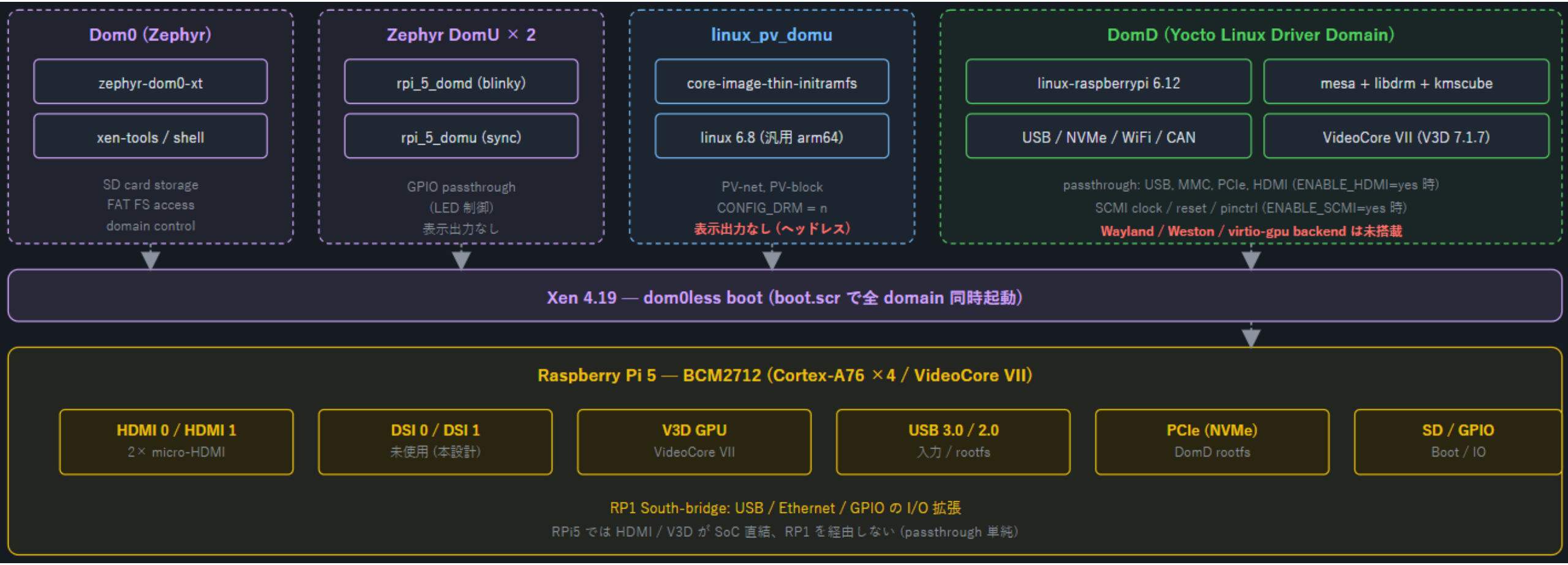
AUTOMOTIVE GRADE LINUX
OSSNA2026

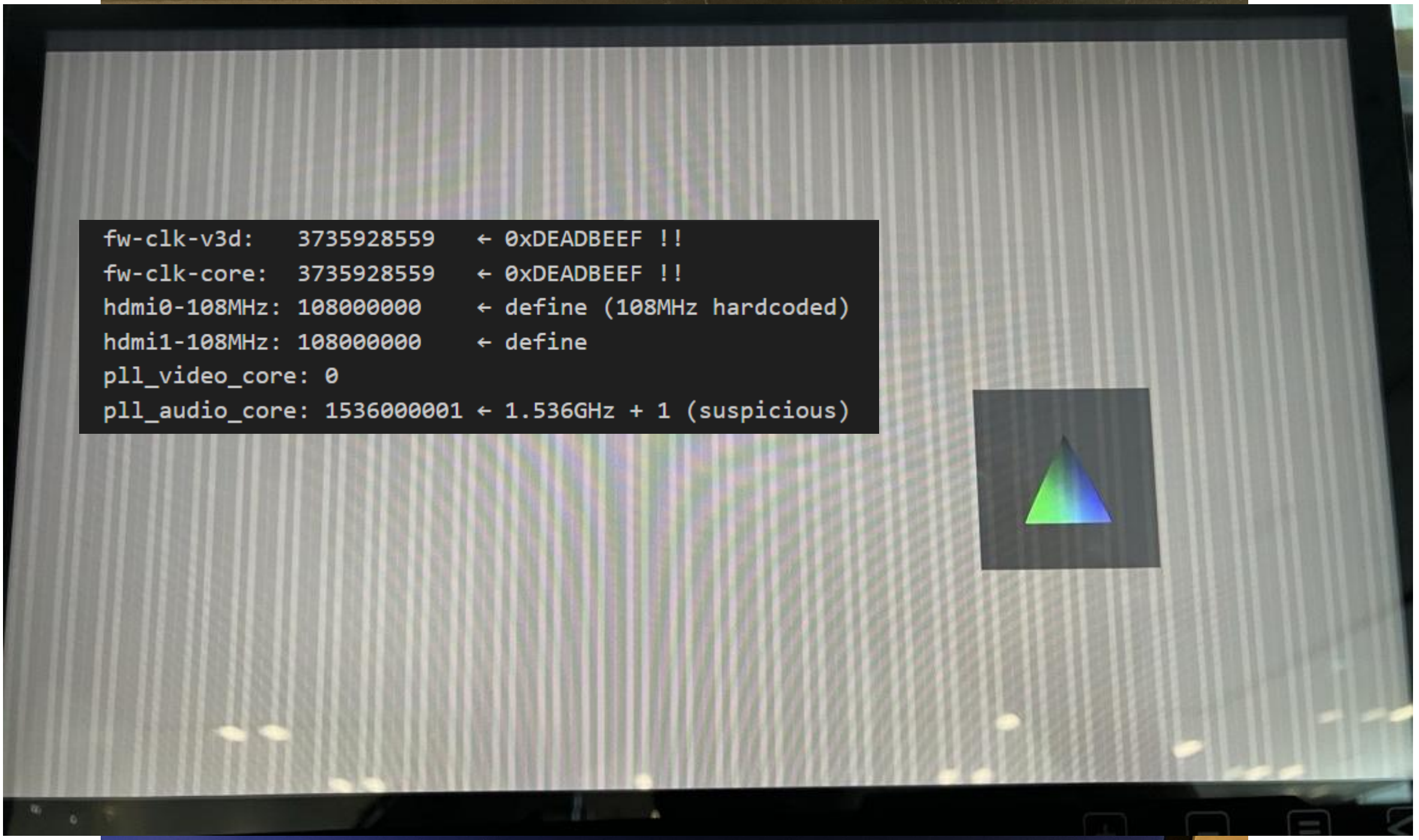



U-Boot

Xen + Dom0(Zephyr) + DomD/U(Waland/Weston/Virtio-gpu)

AUTOMOTIVE GRADE LINUX
OSSNA2026




fw-clk-v3d: 3735928559 ← 0xDEADBEEF !!
fw-clk-core: 3735928559 ← 0xDEADBEEF !!
hdmi0-108MHz: 108000000 ← define (108MHz hardcoded)
hdmi1-108MHz: 108000000 ← define
pll_video_core: 0
pll_audio_core: 1536000001 ← 1.536GHz + 1 (suspicious)

V4H Linux Dom0 vs RPi5 Zephyr Dom0

AUTOMOTIVE GRADE LINUX
OSSNA2026

xl subcommand	V4H Linux Dom0	RPi5 Linux DomD	RPi5 Zephyr Dom0 (`xu`)
`xl create <cfg>`	✅	❌	🟦 `xu create`
`xl create -p` (paused)	✅	❌	🟦 `xu create -p`
`xl create -c` (console attach)	✅	❌ (no xenconsoled)	❌
`xl create -F` (foreground)	✅	❌	❌
`xl destroy <dom>`	✅	❌	🟦 `xu destroy`
`xl shutdown <dom>`	✅	❌	❌
`xl reboot <dom>`	✅	❌	❌
`xl pause <dom>`	✅	❌	🟦 `xu pause`
`xl unpause <dom>`	✅	❌	🟦 `xu unpause`

Coverage by category (combined: RPi5 DomD + Zephyr Dom0)

Category	Commands	Available on RPi5	Coverage
A. Domain lifecycle	14	6 (via xu)	43 %
B. Domain info	10	4	40 %
C. VCPU	3	0	0 %
D. Memory	4	0	0 %
E. Block	5	2 (including xenstore-write substitutes)	40 %
F. Network	6	2	33 %
G. USB / vTPM	4	0	0 %
H. PCI	4	0	0 %
I. xen-troops PV	3	3 (via xenstore-write)	100 %
J. CPU pool / scheduler	11	0	0 %
K. Debug / trace	4	0	0 %
L. Boot / Console	3	1 (xu console)	33 %
M. Driver domain	1	**1 (xl devd ✅)**	**100 %**
Total	**72**	**19**	**26 %**

Xen + Dom0(Zephyr) + DomD/U(without GPU)

AUTOMOTIVE GRADE LINUX
OSSNA2026

Dom0 (Zephyr)

id=0, 128 MB, 1 vCPU

- zephyr-dom0-xt
- xen-tools / xu shell

domain control
+ DomU 起動 (xu create)

★ domain config:
・linux_pv_domu
・linux_pv_domu2

DomU.1 — linux_pv_domu (AGL IC)

id=2, 256 MB, 2 vCPU / Yocto Poky 5.0.10

- agl-ic-demo (Cairo + SHM 自作)
- weston (DomU 内、mesa CPU)
- xen-front DRM (display front)
- xen-blkfront (rootfs /dev/xvda)

bootargs: `console=hvc0` (icdemo 無し)

app-id "DomU.1" → HDMI-A-2
1920×720, SPEED 0..120 km/h sweep
TACHO, FPS 表示

DomU.2 — linux_pv_domu2 (AGL ロゴ)

id=3, 256 MB, 2 vCPU / Yocto Poky 5.0.10

- agl-logo-rotate (simple-egl 派生)
- weston (DomU 内、mesa CPU)
- xen-front DRM (display front)
- xen-blkfront (loop99 on DomD)

bootargs: `icdemo=simple-egl`

app-id "DomU.2" → HDMI-A-1
1920×1080 FHD, 4 秒周期回転
FPS 表示

DomD (Driver Domain)

id=1 (Dom0less-1), 512 MB, 1 vCPU / Yocto core-image-weston

- weston (kiosk-shell.so, pixman renderer)
  [output] HDMI-A-1 ← DomU.2, simple-egl | [output] HDMI-A-2 ← DomU.1
- displ_be v0.2.1-52-gb218-dirty
  C-ii patch (gntalloc 経路) / SHM via /dev/xen/gntalloc → wayland client
- xendriverdomain (xenstored proxy + backend)
- vc4-kms-v3d (DRM master / KMS)
- setup-domu2.service (USB SSD → loop99)

Linux raspberrypi5-domd 6.12.25-v8-16k aarch64
passthrough: HDMI 0/1, V3D, USB, SD/SSD
★ 集中点: **DomD 内 weston が KMS master、全 DomU frame の最終合流地**

↓ ↓ ↓ ↓

Xen 4.19-unstable — dom0less + Xen PV transport

xen-blkfront / xen-front (display) / displ_be SHM via gntalloc / event channel
`xsm=flask flask=permissive | gnttab_max_frames=128 | dom0_mem=128M, dom0_max_vcpus=1`

↓ HW passthrough (DomD 専有) ↓

Raspberry Pi 5 — BCM2712 + RP1

HDMI-A-1 (HDMI0)
→ AGL ロゴ 表示
(DomU.2 / 1920×1080)

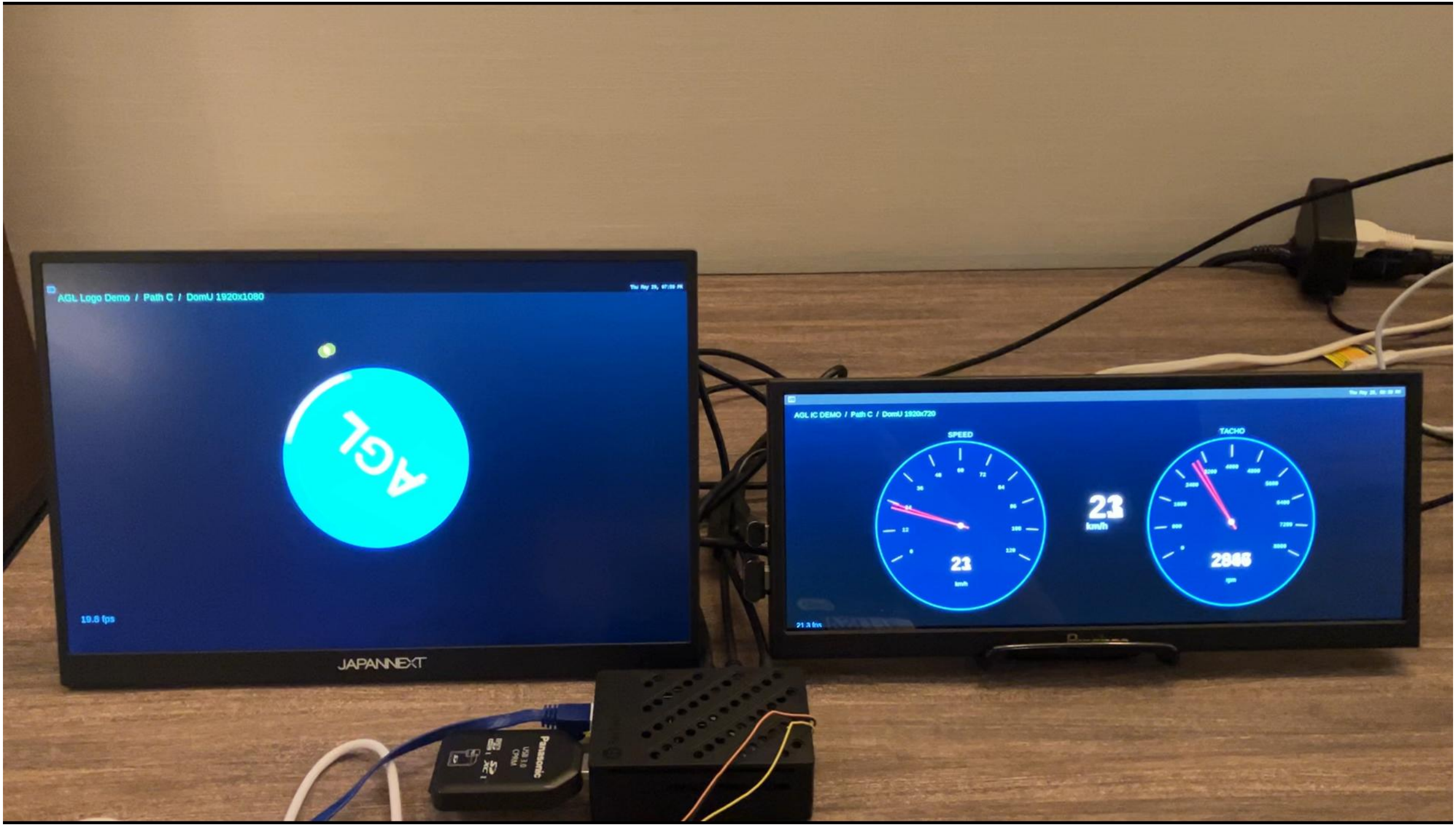
HDMI-A-2 (HDMI1)
→ AGL IC 表示
(DomU.1 / 1920×720)

V3D / VPU GPU
DomD 専有
(KMS / vc4-kms-v3d)

USB 3.0 SSD
DomD: /dev/sda1
DomU: sda2 → loop99

SD カード
SD boot: Xen + Zephyr
+ DomD/DomU kernel

app-id 振り分け: DomU.1 → HDMI-A-2 (AGL IC, cluster 1920×720) / DomU.2 → HDMI-A-1 (AGL ロゴ, FHD 1920×1080)
描画フロー: DomU 内 weston (mesa CPU) → xen-front DRM → displ_be (gntalloc SHM) → DomD weston (kiosk-shell, pixman) → vc4-kms-v3d → HDMI


AGL Logo Demo / Path C / DomU 1920x1080
AGL
19.6 fps
JAPANNEXT
AGL IC DEMO / Path C / DomU 1920x720
SPEED
TACHO
23
km/h
2846
rpm

Xen + Dom0(Zephyr→Linux) + DomD

AUTOMOTIVE GRADE LINUX
OSSNA2026

Dom0 (Zephyr)

id=0, 128 MB, 1 vCPU

- zephyr-dom0-xt
- xen-tools / xu shell

domain control
+ DomU 起動 (xu create)

★ domain config:
・linux_pv_domu
・linux_pv_domu2

DomU.1 — linux_pv_domu (AGL IC)

id=2, 256 MB, 2 vCPU / Yocto Poky 5.0.10

- agl-ic-demo (Cairo + SHM 自作)
- weston (DomU 内、mesa CPU)
- xen-front DRM (display front)
- xen-blkfront (rootfs /dev/xvda)

bootargs: `console=hvc0` (icdemo 無し)

app-id "DomU.1" → HDMI-A-2
1920×720, SPEED 0..120 km/h sweep
TACHO, FPS 表示

DomU.2 — linux_pv_domu2 (AGL ロゴ)

id=3, 256 MB, 2 vCPU / Yocto Poky 5.0.10

- agl-logo-rotate (simple-egl 派生)
- weston (DomU 内、mesa CPU)
- xen-front DRM (display front)
- xen-blkfront (loop99 on DomD)

bootargs: `icdemo=simple-egl`

app-id "DomU.2" → HDMI-A-1
1920×1080 FHD, 4 秒周期回転
FPS 表示

DomD (Driver Domain)

id=1 (Dom0less-1), 512 MB, 1 vCPU / Yocto core-image-weston

- weston (kiosk-shell.so, pixman renderer)
  [output] HDMI-A-1 ← DomU.2, simple-egl | [output] HDMI-A-2 ← DomU.1
- displ_be v0.2.1-52-gb218-dirty
  C-ii patch (gntalloc 経路) / SHM via /dev/xen/gntalloc → wayland client
- xendriverdomain (xenstored proxy + backend)
- vc4-kms-v3d (DRM master / KMS)
- setup-domu2.service (USB SSD → loop99)

Linux raspberrypi5-domd 6.12.25-v8-16k aarch64
passthrough: HDMI 0/1, V3D, USB, SD/SSD
★ 集中点: **DomD 内 weston が KMS master、全 DomU frame の最終合流地**

↓ ↓ ↓ ↓

Xen 4.19-unstable — dom0less + Xen PV transport

xen-blkfront / xen-front (display) / displ_be SHM via gntalloc / event channel
`xsm=flask flask=permissive | gnttab_max_frames=128 | dom0_mem=128M, dom0_max_vcpus=1`

↓ HW passthrough (DomD 専有) ↓

Raspberry Pi 5 — BCM2712 + RP1

HDMI-A-1 (HDMI0)
→ AGL ロゴ 表示
(DomU.2 / 1920×1080)

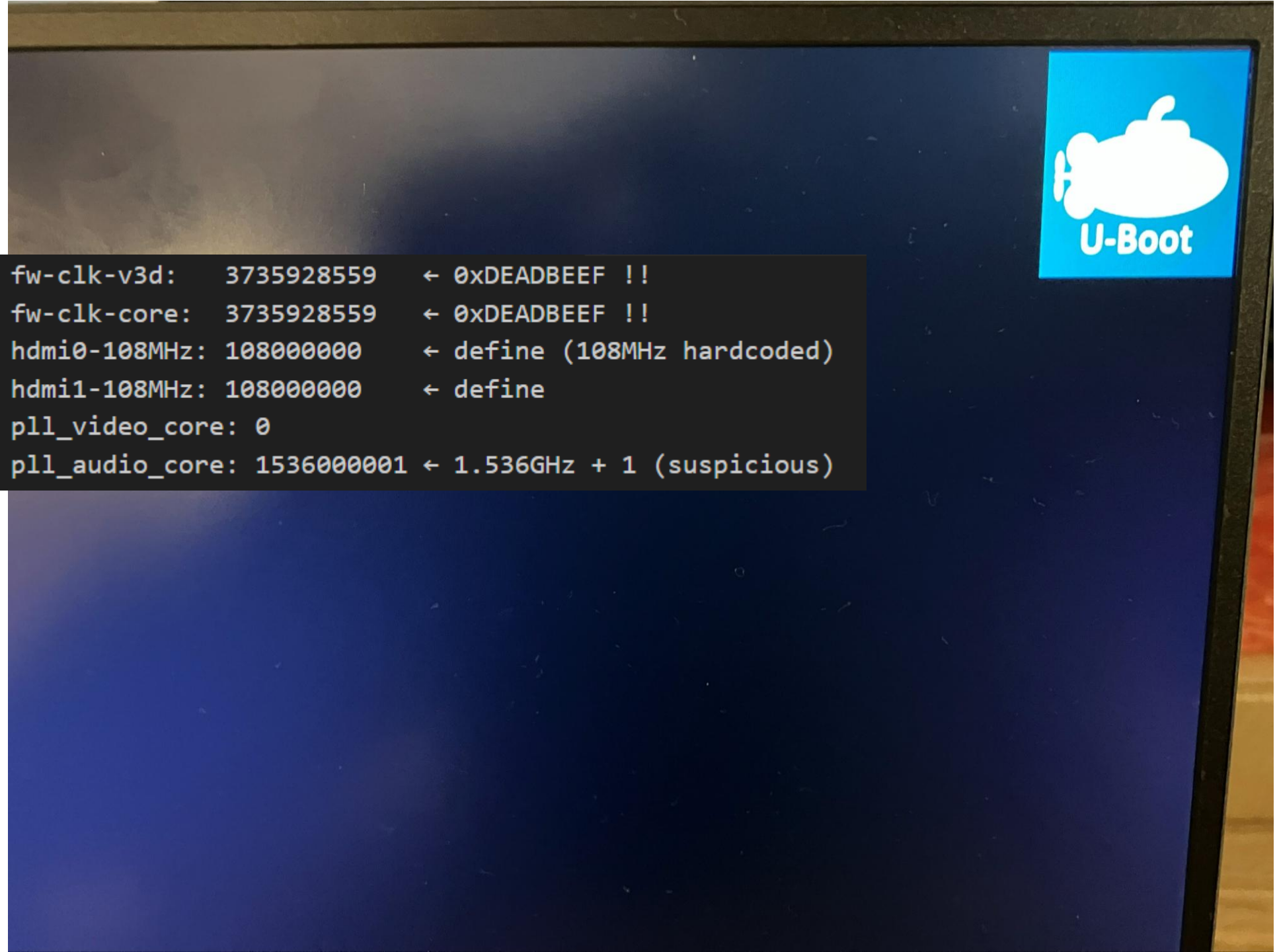
HDMI-A-2 (HDMI1)
→ AGL IC 表示
(DomU.1 / 1920×720)

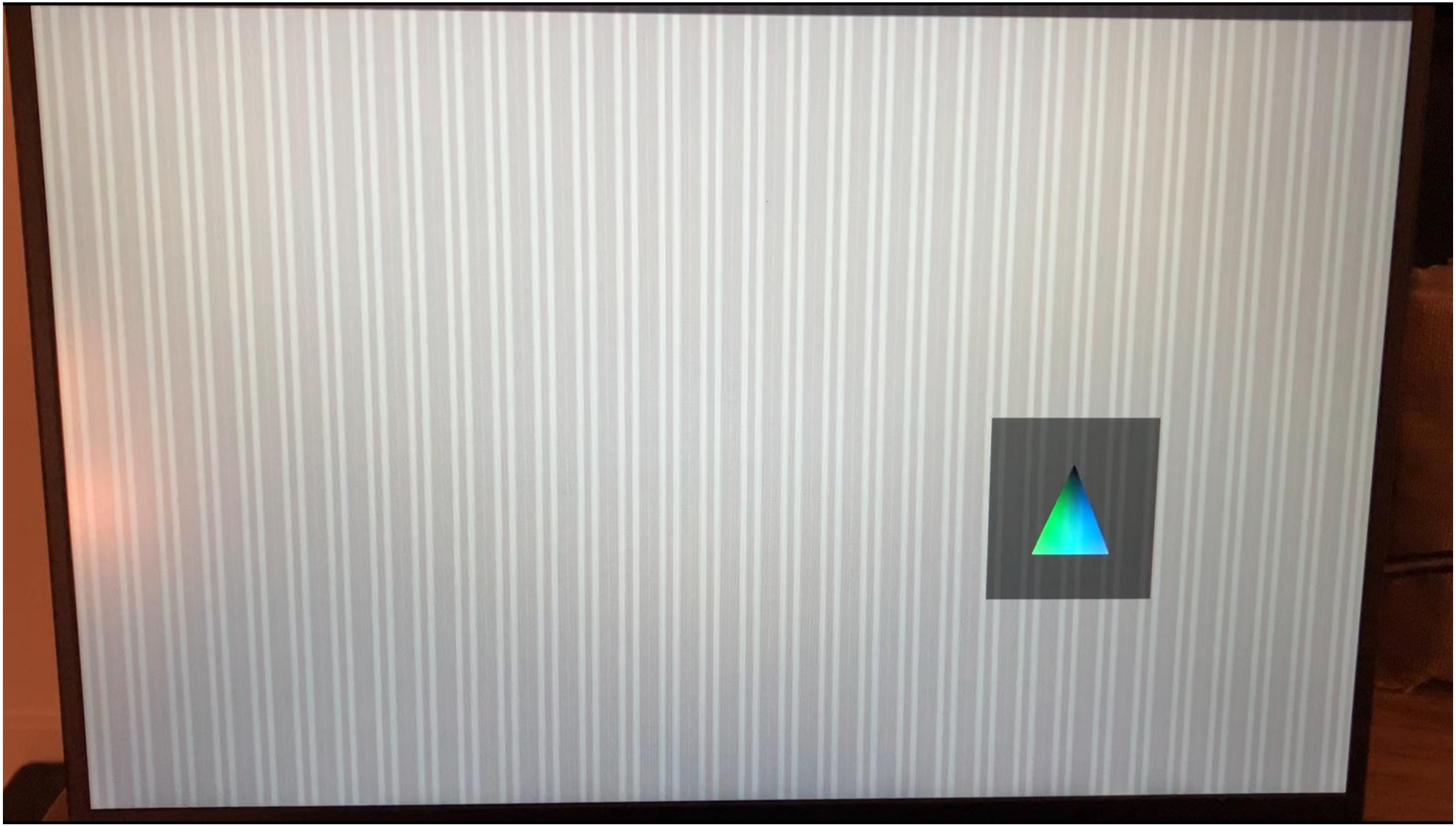
V3D / VPU GPU
DomD 専有
(KMS / vc4-kms-v3d)

USB 3.0 SSD
DomD: /dev/sda1
DomU: sda2 → loop99

SD カード
SD boot: Xen + Zephyr
+ DomD/DomU kernel

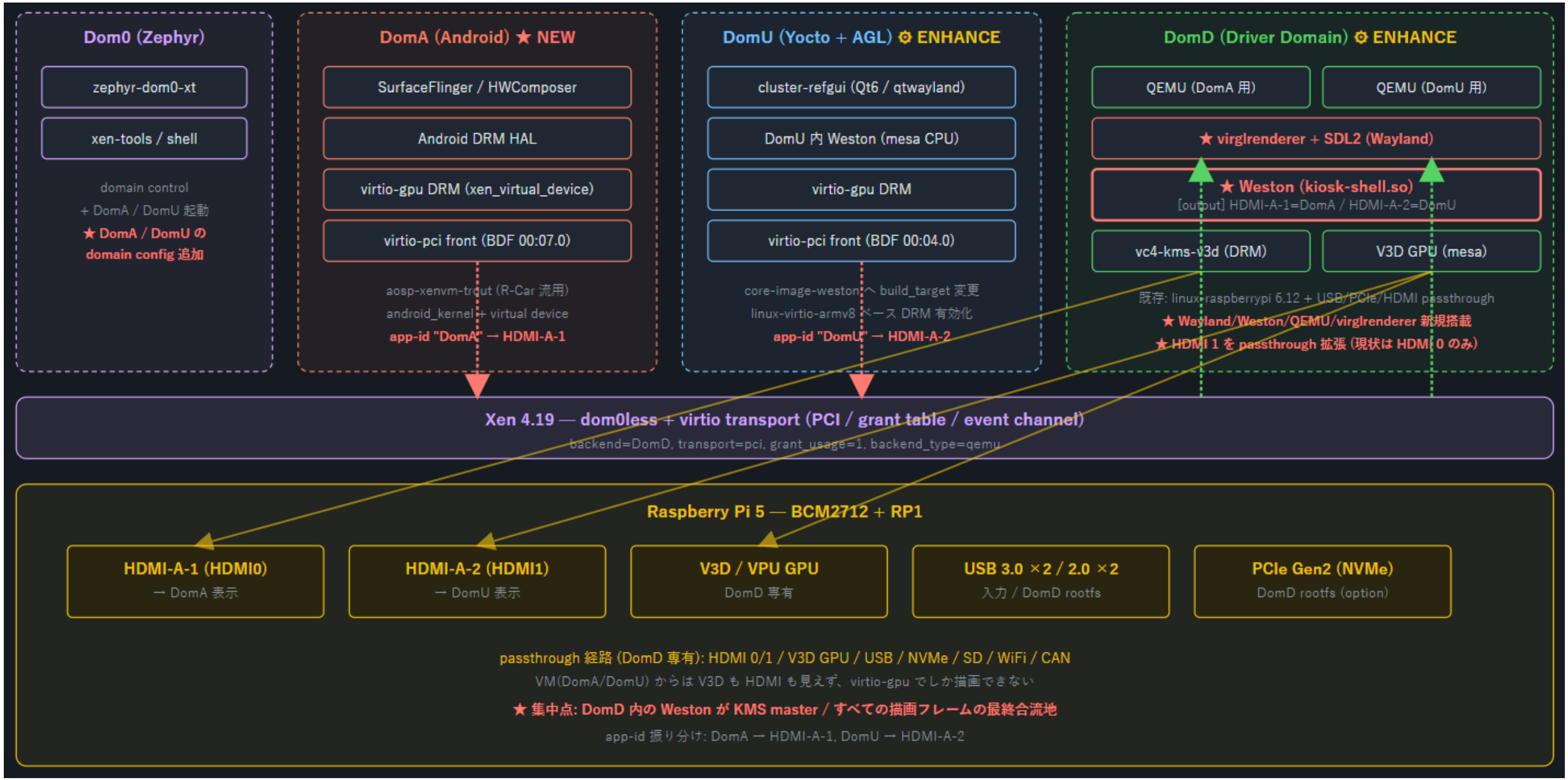
app-id 振り分け: DomU.1 → HDMI-A-2 (AGL IC, cluster 1920×720) / DomU.2 → HDMI-A-1 (AGL ロゴ, FHD 1920×1080)
描画フロー: DomU 内 weston (mesa CPU) → xen-front DRM → displ_be (gntalloc SHM) → DomD weston (kiosk-shell, pixman) → vc4-kms-v3d → HDMI


U-Boot
fw-clk-v3d: 3735928559 ← 0xDEADBEEF !!
fw-clk-core: 3735928559 ← 0xDEADBEEF !!
hdmi0-108MHz: 108000000 ← define (108MHz hardcoded)
hdmi1-108MHz: 108000000 ← define
pll_video_core: 0
pll_audio_core: 1536000001 ← 1.536GHz + 1 (suspicious)



Xen + Dom0(Zephyr) + DomD(Linux) + DomU(AGL IC) + DomU(AAOS)

AUTOMOTIVE GRADE LINUX
OSSNA2026



Conclusion

Conclusion

- The automotive industry(SDV) is undergoing a paradigm shift
- It is impossible to achieve it alone
- However, the OSS community already has many solutions
- We need to move forward together with the OSS community

Open Source Software and Community one of the keys

Text Channels

japanese-language

RPi5 SoDeV立ち上げスレッド

general

all-member-meetings

off-topic

weekly-dev-call

google-summer-of-code

industry-news

autoworld-japan

embedded-world

Voice Channels

Lounge

Meeting Room

Expert Groups

system-architecture-team

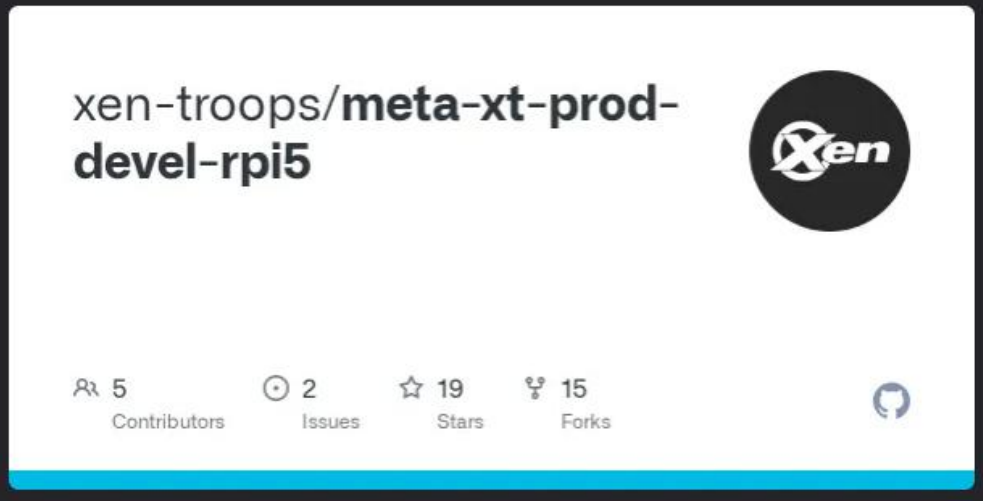
https://github.com/xen-troops/meta-xt-prod-devel-rpi5/tree/main
を使用して、下記の構成案を実現出来れば、SoDeVが実現出来るかもしれない

RPi5 (2x HDMI)
└─ Xen (Xen-Troops)
  ├─ Dom0: Zephyr
  ├─ DomD: Linux/Yocto (GPU/HDMI)
  │ ├─ Weston (kiosk-shell, DRM backend)
  │ │ ├─ HDMI-A-1: DomD UI (AGL IC)
  │ │ └─ HDMI-A-2: QEMU (SDL) = DomU GUI
  │ └─ QEMU device-model (SDL, virtio-vga-gl)
  └─ DomU: Linux/Yocto (VirtIO-GPU + AGL IVI)

GitHub

GitHub - xen-troops/meta-xt-prod-devel-rpi5

Contribute to xen-troops/meta-xt-prod-devel-rpi5 development by creating an account on GitHub.



https://discord.com/channels/1278057614349176882/1478164066945204225/1478164702436528129

Thank you very much

Let‘s build the future together and Open Way

THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

Embedded Linux Conference