ELISA
Enabling Linux in Safety Applications

WORKSHOP

Automotive Grade Linux SoDeV Project

ELISA Workshop Munich, Germany
November 18-20, 2025
Co-hosted with Red Hat

What is Software Defined

Mechanical (in 2025/summer)

Digital

Software oriented

The same Software Evolution is occurring in Automotive : Software Defined Vehicle

ELISA Enabling Linux in Safety Applications WORKSHOP

Speaker Biography

- Name: Harunobu Kurokawa
  - Working in Renesas Electronics Corporation. for 15+ years.
  - Embedded Software Engineer since 2007,
  - Linux and OSS for automotive development (2013 ~)
  - OSS team manager

- My experience of Open-Source Community
  - Join AGL development (2016/Dec ~)
    - AGL BSP developer, reviewer and SAT member
    - Joined Instrument Cluster Expert Gr. (2019 ~), AGL Steering Committee member (2020 ~)
  - Join Xen/Zephyr Working Gr and boards (2025 ~)

Speaker Biography

- Name: Naoto Yamaguchi
  - Doctor of Informatics
  - Working in AISIN CORPORATION. for 15+ years.
  - Embedded Software Engineer since 2007,
  - Linux and OSS for automotive development (2011 ~)

- My experience of Open-Source Community
  - Join to AGL (2014~)
    - Instrument Cluster Expert Gr. (2019 ~)
    - AGL Steering Committee/Board member (2020 ~)
    - SoDeV Development (2025 ~)
  - Join to ELISA (2019 ~)

AUTOMOTIVE GRADE LINUX

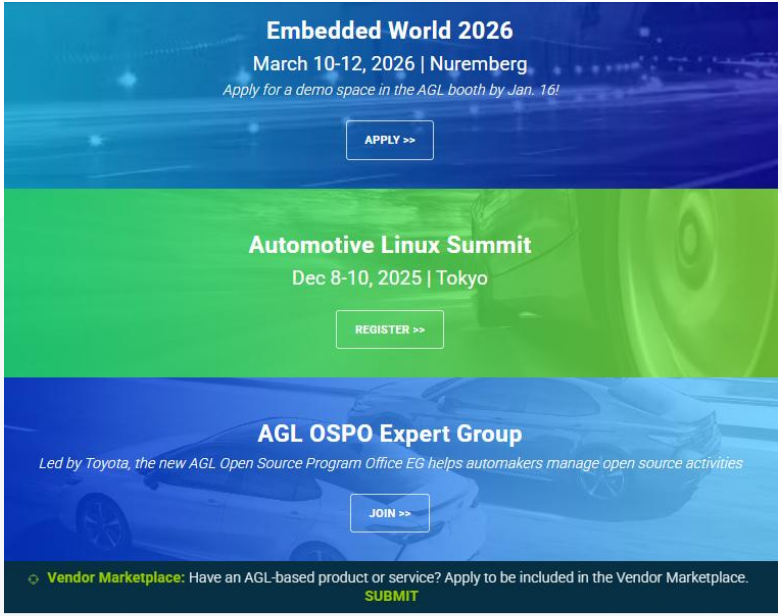
What is Automotive Grade Linux

ELISA
Enabling Linux in Safety Applications

WORKSHOP

Overview

- The opensource project for automotive.
- Target
  - Initially, AGL launches to create common IVI(In-Vehicle Infotainment) opensource platform about 10 years ago.
  - Recently, AGL covers to cockpit system(IVI + Instrument Cluster), telematics not only IVI.
  - Most recently target is an SDV(Software Defined Vehicle) and OSPO(Open-Source Program Office).
- Two key procedure.
  - **Incubate and develop** opensource software components to covered automotive specific use-case.
  - **Integrate** Linux based automotive software platform to share good practice.



What is Automotive Grade Linux?

Automotive Grade Linux is a collaborative open source project that is bringing together automakers, suppliers and technology companies to accelerate the development and adoption of a fully open software stack for the connected car. With Linux at its core, AGL is developing an open platform from the ground up that can serve as the de facto industry standard to enable rapid development of new features and technologies.

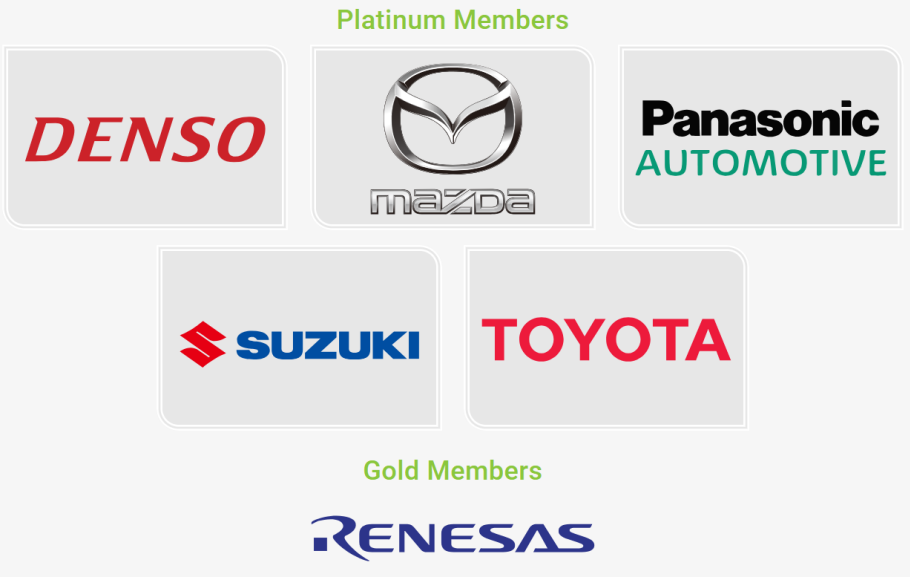
ELISA Enabling Linux in Safety Applications

WORKSHOP

Membership

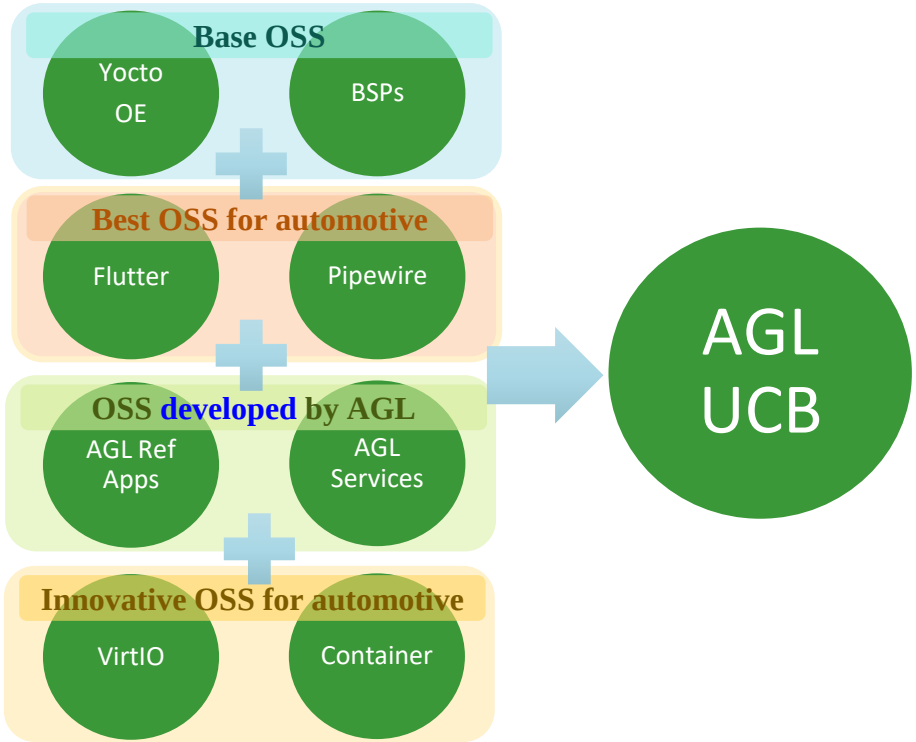
- Many companies join to AGL.

Platinum : 5 Gold : 1 Silver : 5 Bronze : 35



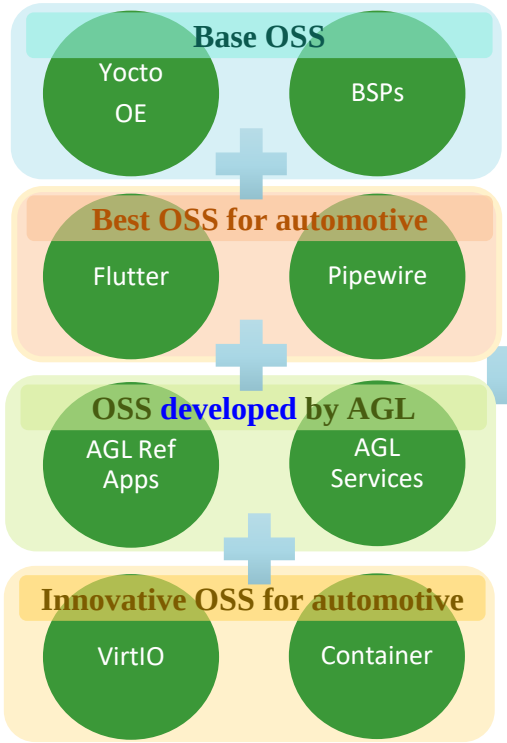
From: https://www.automotivelinux.org/about/members/

Unified code base



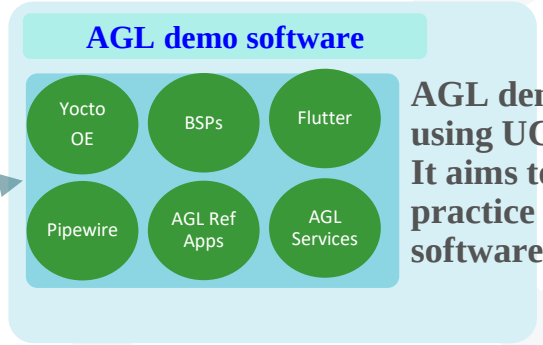
- UCB is a AGL code base.
  - It's constructed from various open-source software.
    - Base open-source software for Linux.
    - Best open-source software for automotive use.
    - AGL **developed** open-source software for automotive.
    - Innovative open-source software for automotive.
  - UCB is a repository.
    - Not a fork. Reference to existing repository.
    - Self hosting to AGL developed OSS.

Unified code base



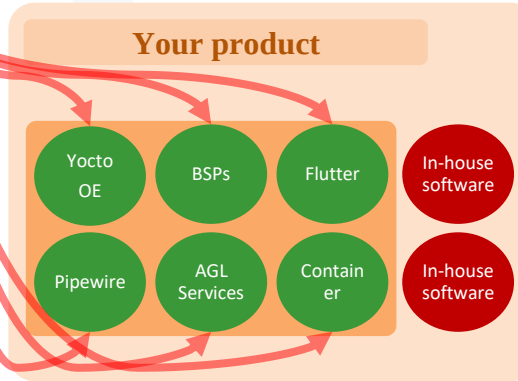
UCB is incubator

Incubates AGL demo software to evaluate AGL software.



AGL demo software **integrates** using UCB.
It aims to show one of the best practice of AGL based full software stack.

Incubates AGL based product.



You can pick and use any software from UCB. AGL does not have restriction such as compliance program, certification, and etc.

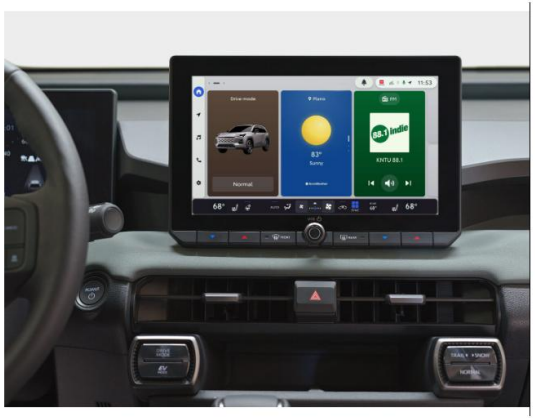
ELISA Enabling Linux in Safety Applications WORKSHOP

Incubated products from UCB

Toyota and Suzuki Announce official launch of Products Powered by Automotive Grade Linux – 2025 Release

TOYOTA
NEWSROOM

While the system platform architecture remains on Automotive Grade Linux, featuring open-source software, it also benefits from Woven by Toyota's Arene software development kit. The introduction of Arene is Toyota's first step toward fully software-defined vehicles. The platform will be the foundation for Toyota's most advanced safety, security and connectivity technologies and will create new opportunities to enhance features for its customers on a global scale.



https://pressroom.toyota.com/the-latest-evolution-of-toyotas-multimedia-coming-to-a-screen-near-you/

Qt Group

Suzuki is building its cockpit solution on the open-source Automotive Grade Linux (AGL), allowing developers to build and deploy software on a common, industry-standard platform, rather than starting from scratch. Qt Group has been a key partner of AGL since 2015, actively collaborating to develop advanced, open-source, Linux-based solutions for connected car infotainment and HMI systems. With nearly a decade of joint development, Qt is fully optimized to run on AGL, helping manufacturers like Suzuki deliver high-quality, scalable, and reliable HMI experiences.

https://www.qt.io/ja-jp/press/qt-group-suzuki

https://www.qt.io/press/suzuki-chooses-qt-to-power-its-digital-cockpit-for-mainstream-evs?hsLang=en

ELISA
Enabling Linux in Safety Applications

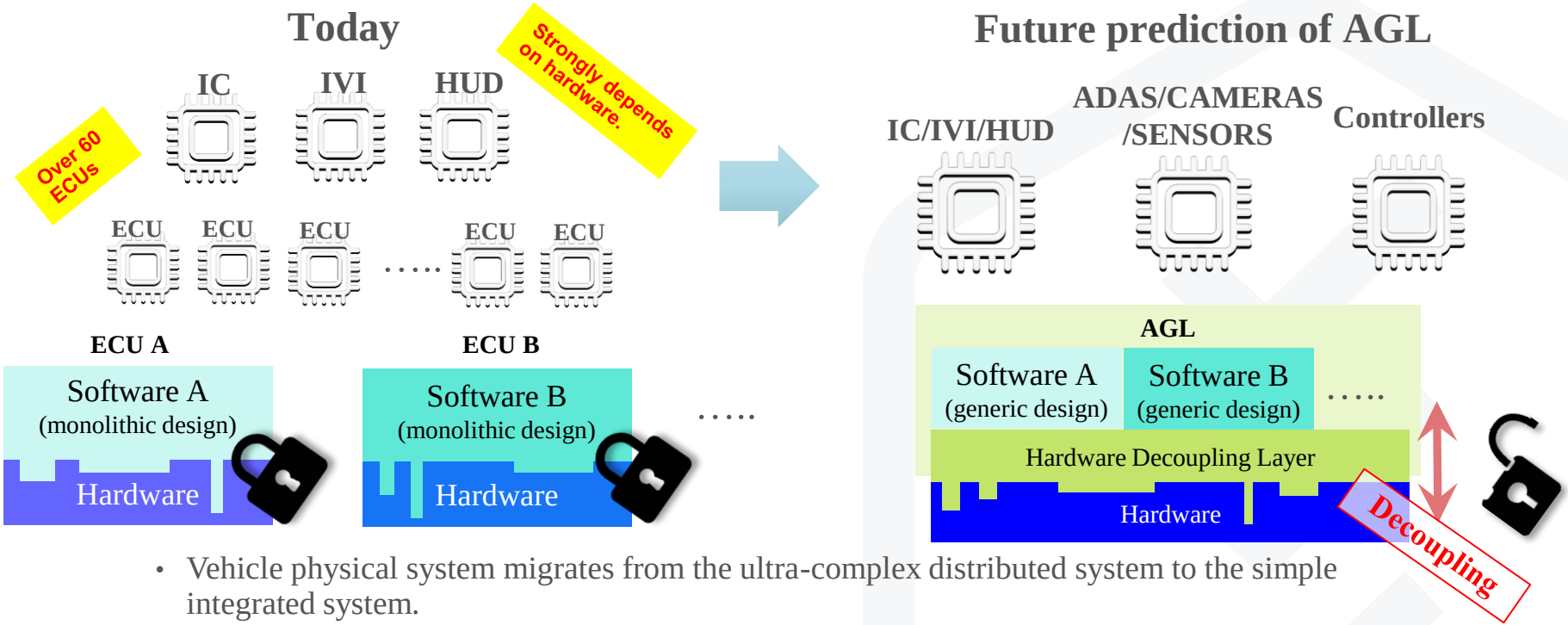
WORKSHOP

Conclusion for What is AGL

- AGL is a technology incubator for the Automotive Industry.
  - All technology is constructed with open-source software.
  - AGL focuses on system integration using open-source software whose scope is not limited to Linux distributions.
- Collaboration method of AGL.
  - AGL shares the generic part of the system by code.
  - AGL shares the specific part of the system by document, such as the presentation of the All-Member Meeting.
- AGL users (not limited to member companies) develop products using AGL technology seeds.
  - Some product is incubated by AGL in this year.

What is SoDeV

Evolution of Vehicle Software

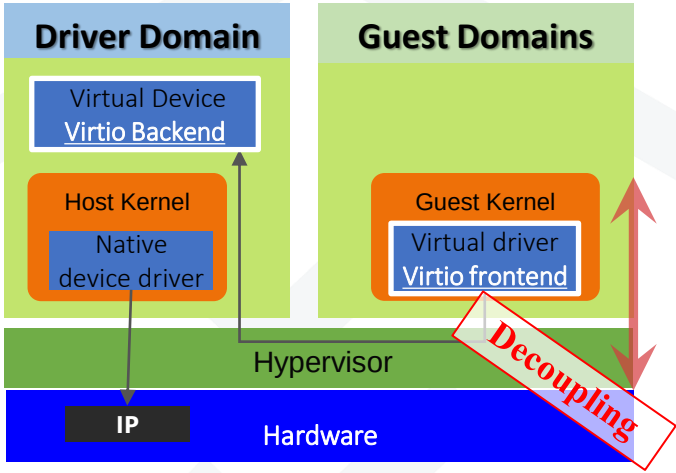


- Vehicle physical system migrates from the ultra-complex distributed system to the simple integrated system.
- Hardware decoupling is the most important thing. Reduce complexity, use the same software, easy to upgrade and more.

ELISA Enabling Linux in Safety Applications WORKSHOP

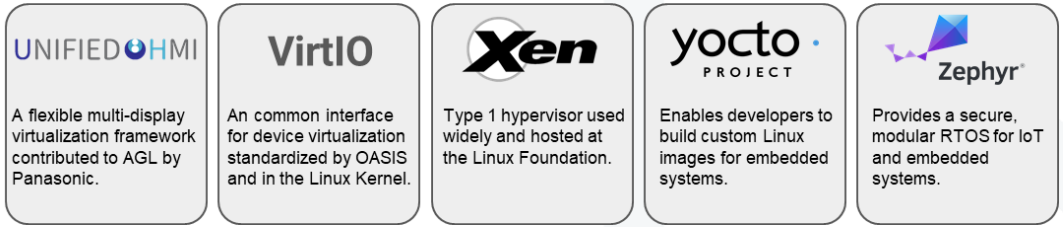
Hardware decupling Layer : VirtIO

- AGL focuses on achieving hardware decoupling.
- Our proposal is
  - 'VirtIO is a common and generic hardware interface in automotive systems'.
- Keypoints:
  - VirtIO is a standardized interface for efficient I/O (input/output) in virtualized environments.
    - It makes fast communication between the Guest VM and host.
    - The guest sends requests to the host through virtio instead of using real devices.
  - It's an open standard. Some open-source software supported it.
    - Linux Kernel, Xen, KVM, etc...
- It achieves software reusability between automotive systems.



AGL SDV

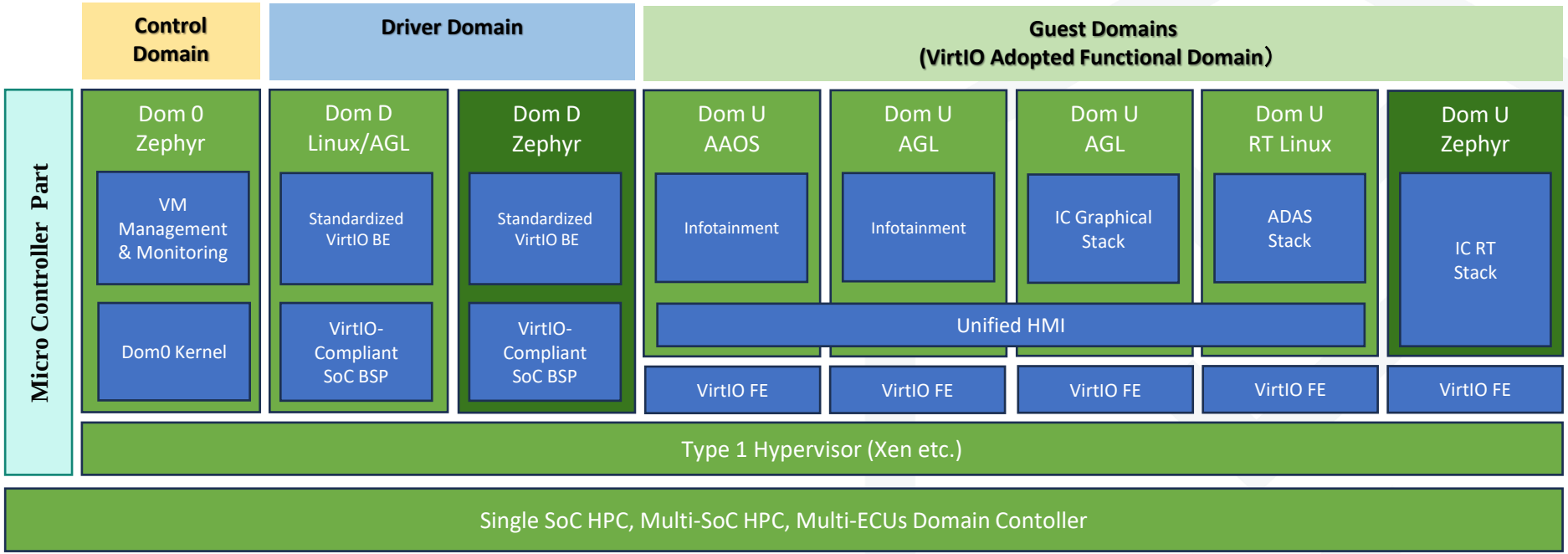
- AGL SDV must be built using open-source software.
  - AGL adopts open-source technologies.
  - AGL applies to automotive requirements into OSS.
  - Focuses on implementation rather than specifications as Code-First.
  - Integrates the implemented stack and makes it available to everyone
    - The proprietary software is implemented on the common software stack during the product development phase.



- To achieve this, the reference SDV platform is provided by AGL — **this is SoDeV.**

ELISA Enabling Linux in Safety Applications WORKSHOP

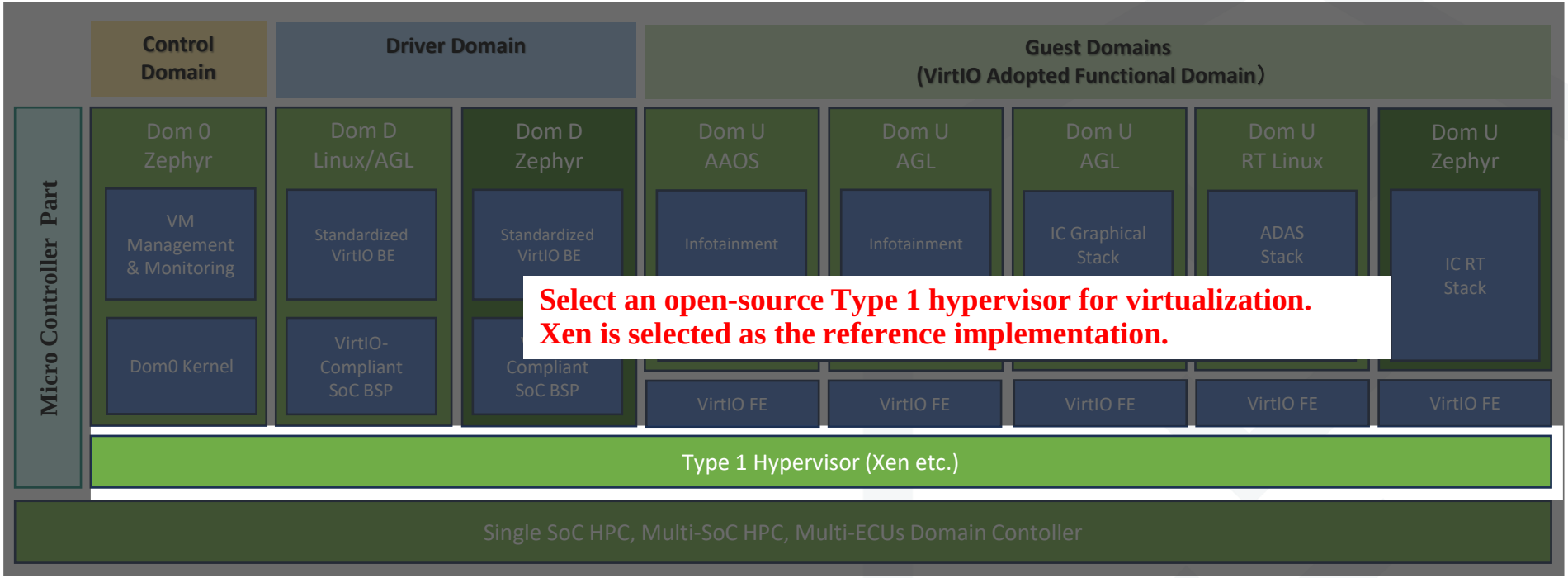
SoDeV Architecture



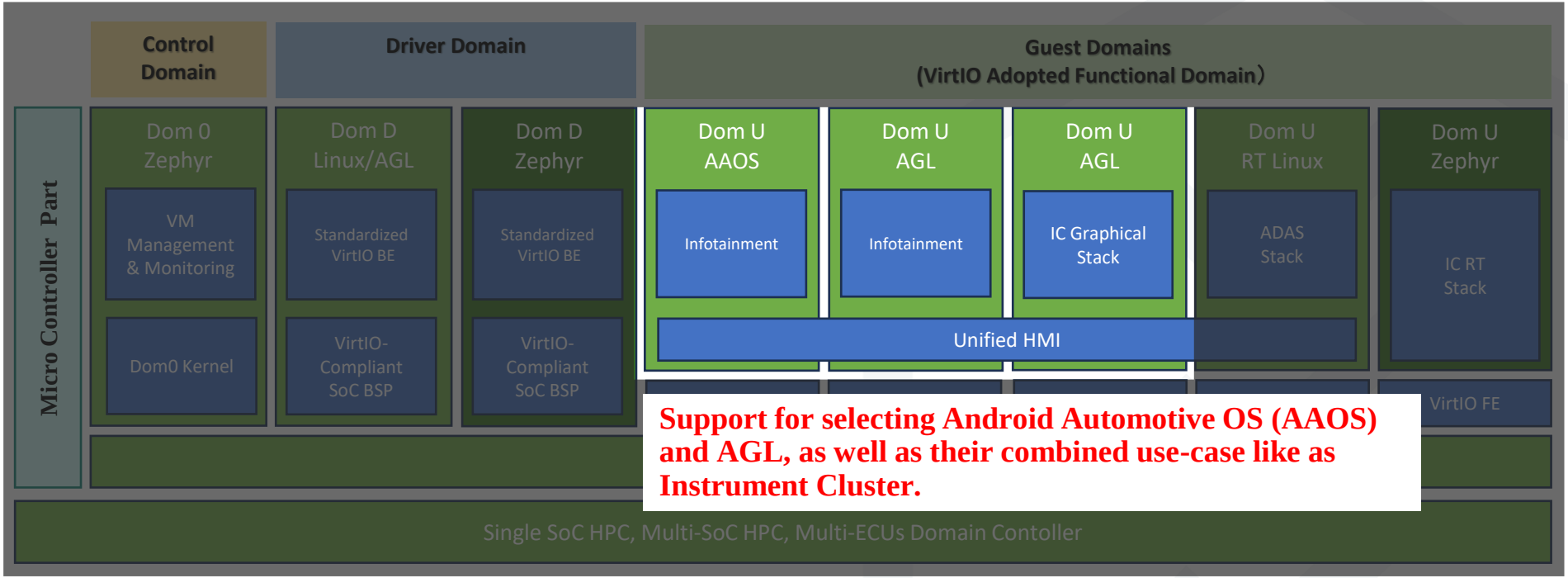
ELISA Enabling Linux in Safety Applications

WORKSHOP

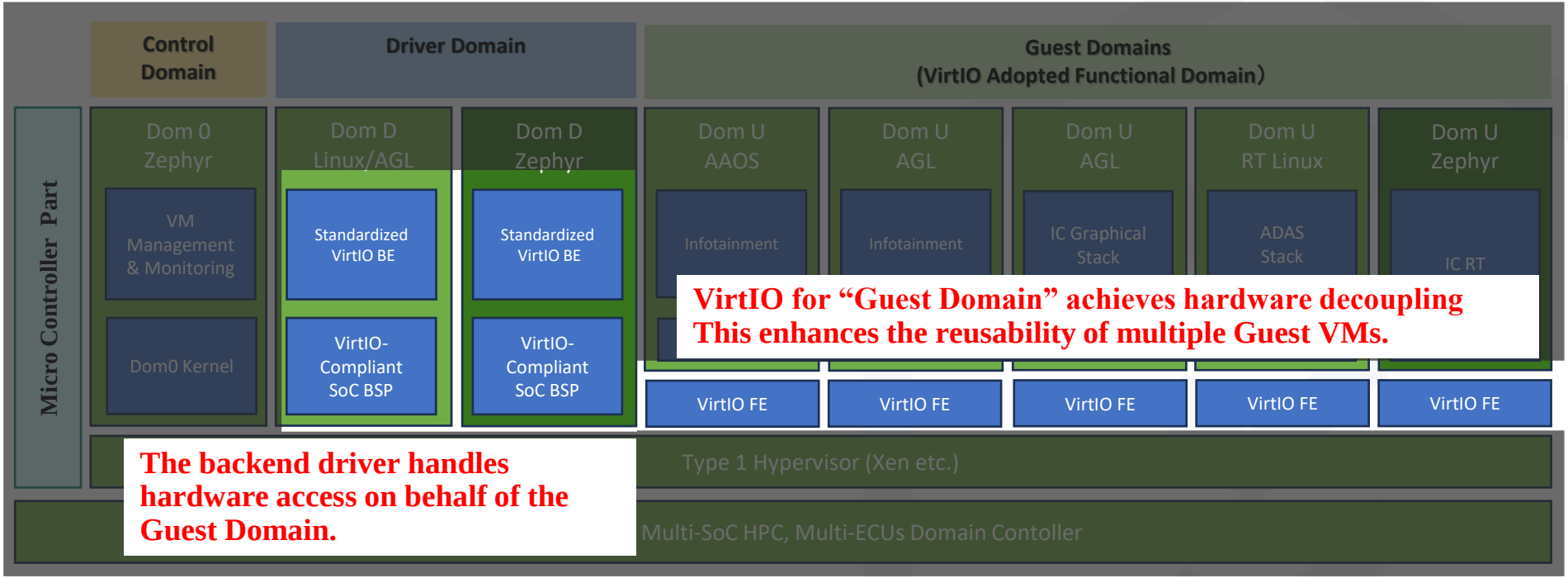
SoDeV Architecture



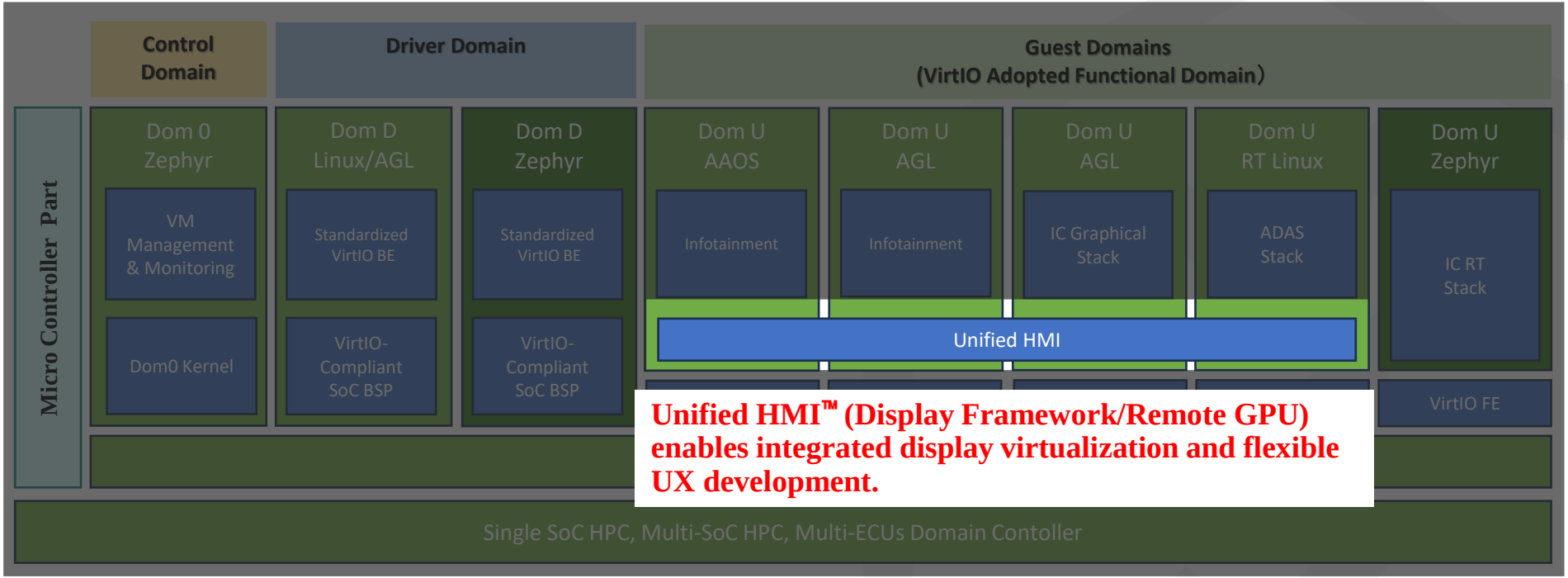
SoDeV Architecture



SoDeV Architecture



SoDeV Architecture

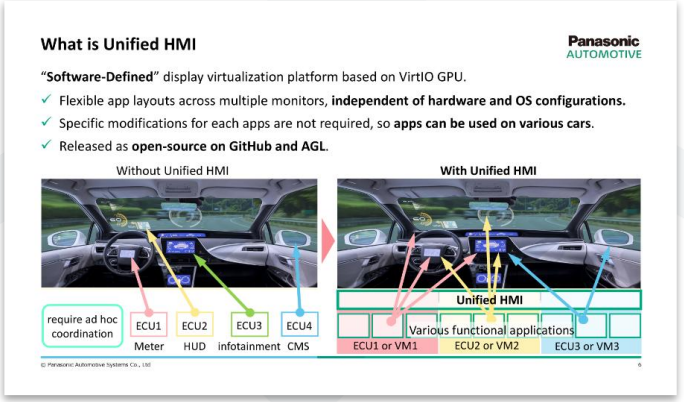


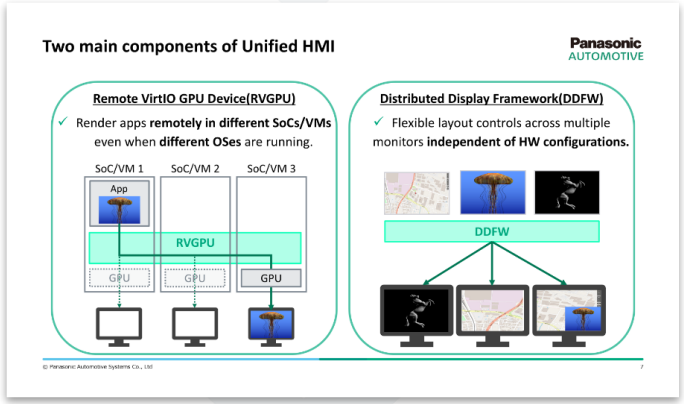
ELISA Enabling Linux in Safety Applications

WORKSHOP

Unified HMI™

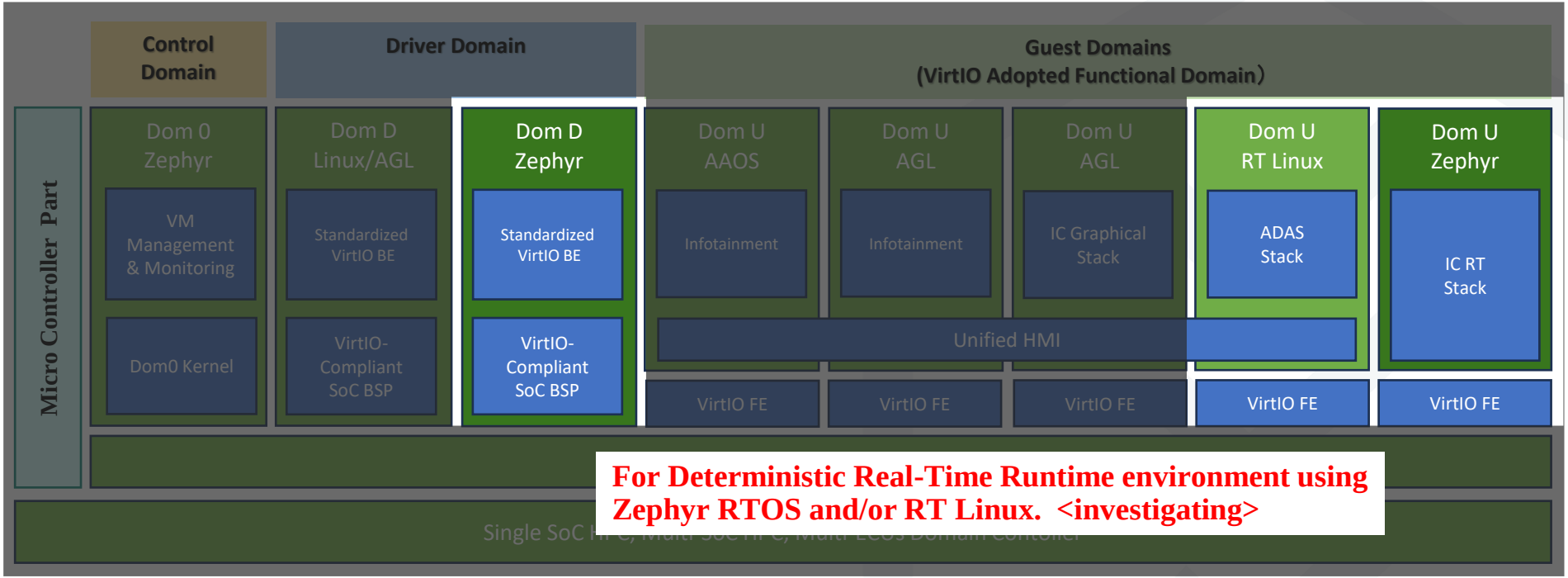
- It is Multi-display virtualization platform based on VirtIO GPU in AGL SDV-EG (led by Panasonic)
- Unified HMI allows for flexible development of the entire cockpit and cabin UI/UX
  - across multiple displays, independent of hardware and OS configuration.
    - Remote VirtIO GPU Device(RVGPU)：Render apps remotely in different SoCs.
    - Distributed Display Framework(DDFW)：Flexible layout control of apps across multiple displays.



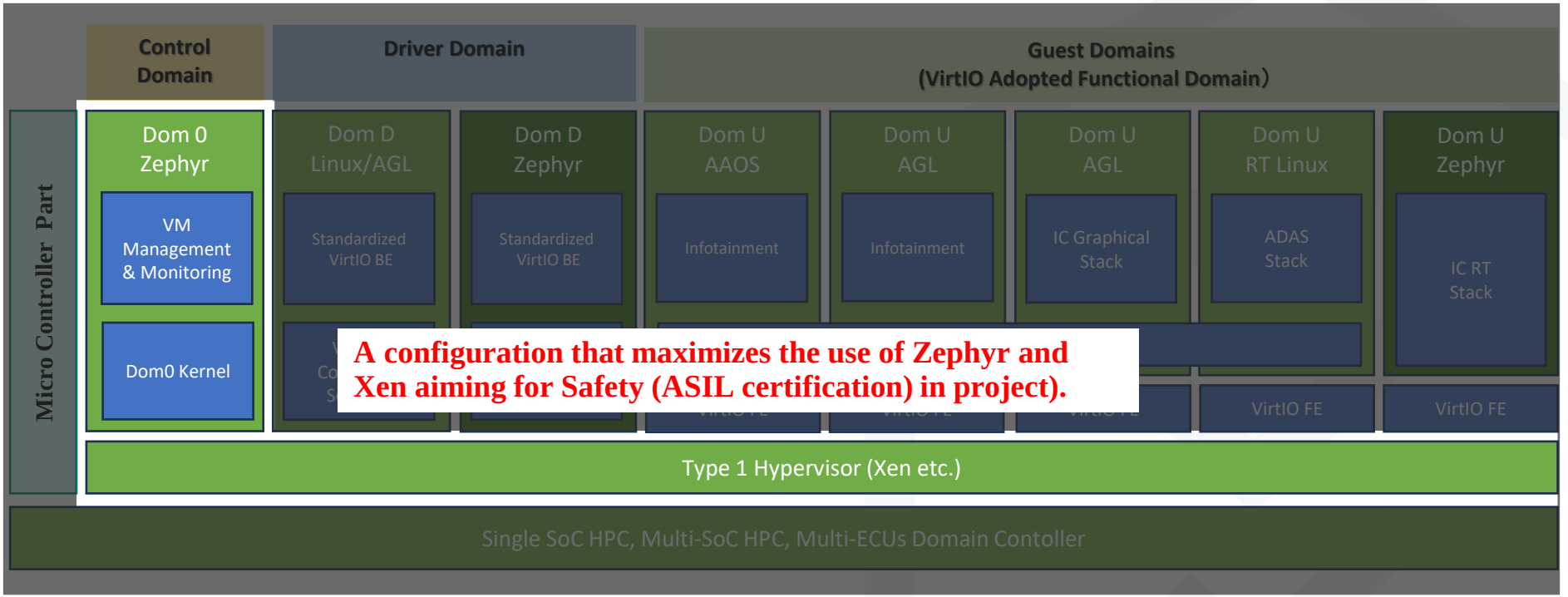


https://static.sched.com/hosted_files/aglammsummer2025/18/Upgraded%20Unified%20HMI%20Simplifying%20Window%20System%20Dependencies.pdf : presented Murakami-san from Panasonic

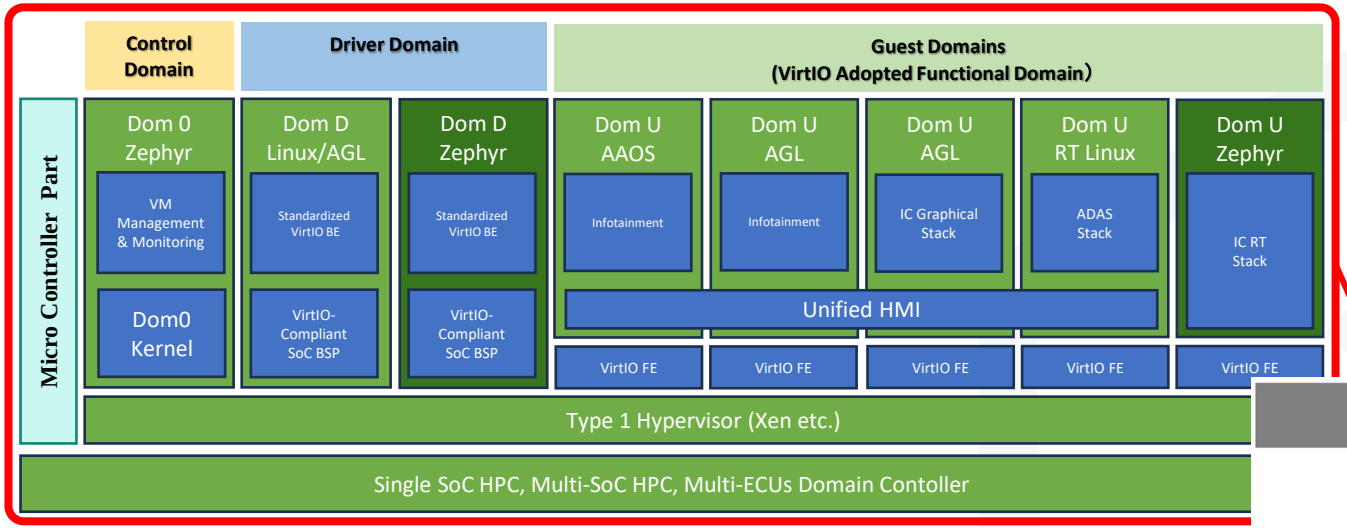
SoDeV Architecture



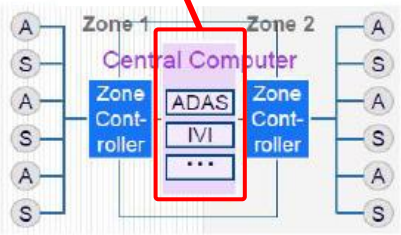
SoDeV Architecture



SoDeV vs Vehicle Physical System Architecture



Zone & Central

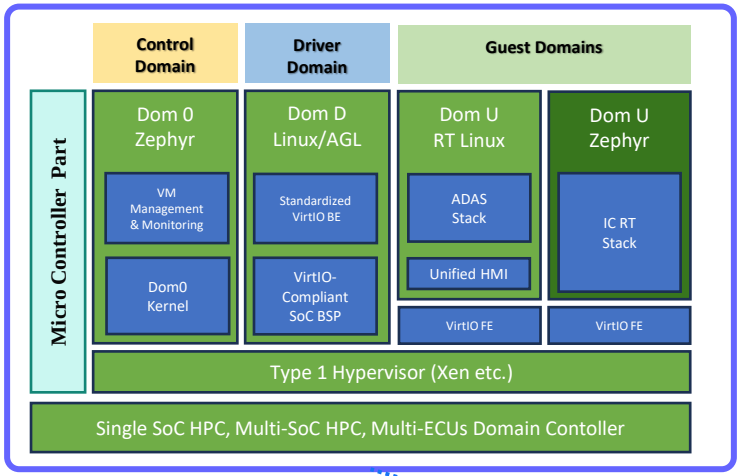


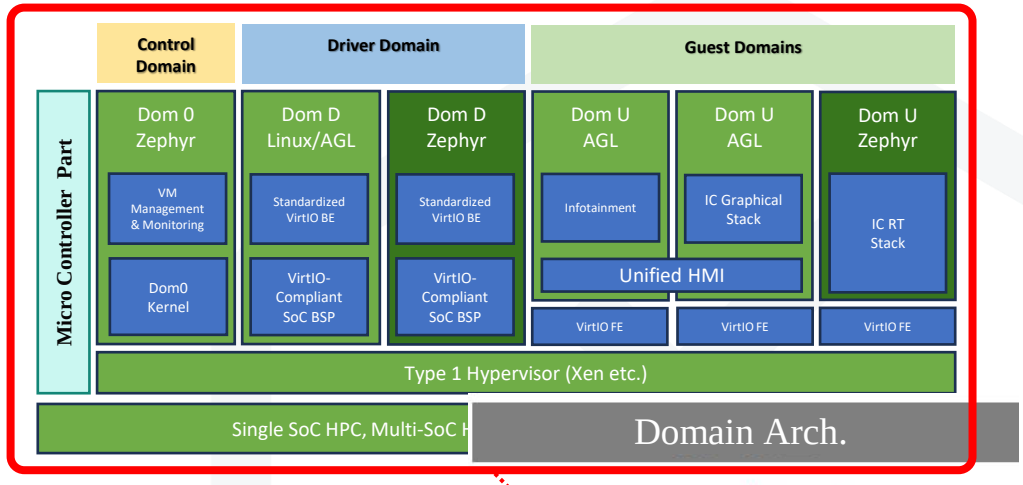
- SoDeV scalability
  - For the Zone & Central vehicle physical system architecture with High Performance Computing, it enables many functional domain on VirtIO runtime environment.

ELISA Enabling Linux in Safety Applications

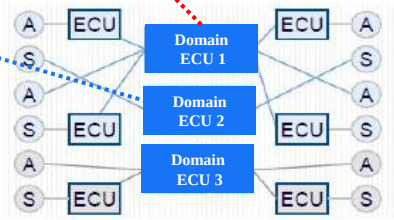
WORKSHOP

SoDeV vs Vehicle Physical System Architecture





- SoDeV scalability
  - For the Domain vehicle physical system architecture, it enables scaling to fit each domain controller with VirtIO runtime environment.



Conclusion for SoDeV

- SoDeV is a reference SDV platform that is accelerated by open-source technology.
- SoDeV focuses on
  - Hardware decoupling to realize software reusability.
    - It's accelerated by VirtIO infrastructure.
  - VM-based architecture to realize system scalability.
    - It's accelerated by hypervisor infrastructure.
  - Integrate deterministic and non-deterministic software into one system using VMs.
    - Remote rendering capability for deterministic domain accelerated by Unified HMI.
    - Accelerate some SDV trends such as cloud development and others.
- These are the research points.

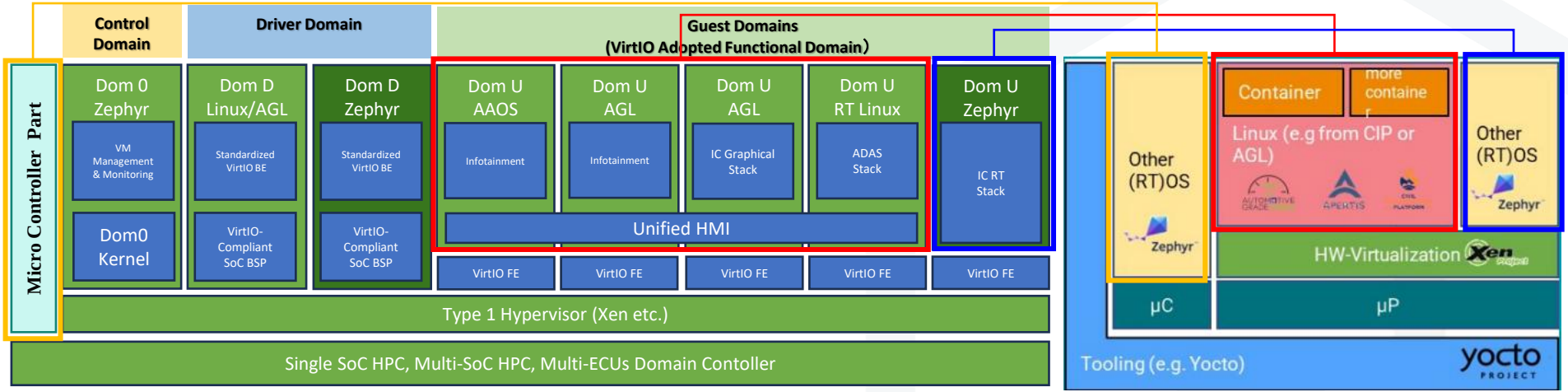
SoDeV and Safety Collaboration point

ELISA
Enabling Linux in Safety Applications

WORKSHOP

Antinomy between SoDeV and Safety

- What is the difference between SoDeV and ELISA architecture?



- SoDeV is one more breakdown from ELISA architecture.
  - Defined hardware decoupling, not only a software domain isolation.
  - Divided device-specific domain - DomD.
  - DomU software shall be reusable with other systems.

Antinomy between SoDeV and Safety

Linux in Safety Critical Systems

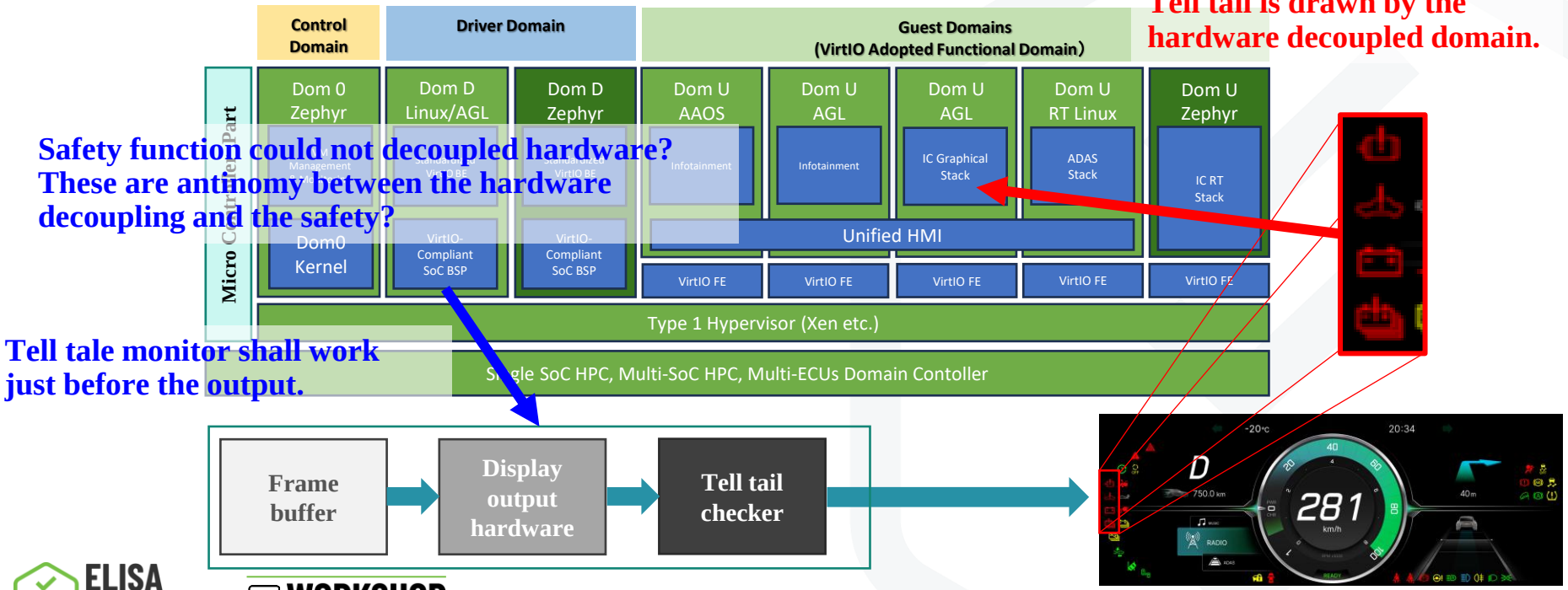
“Assessing whether a system is safe, requires understanding the system sufficiently.”

- Understand Linux within that system context and how Linux is used in that system.
- Select Linux components and features that can be evaluated for safety.
- Identify gaps that exist where more work is needed to evaluate safety sufficiently.

From: The Way for Safe Systems Constructed by Open Source Software, Philipp Ahmann

Antinomy between SoDeV and Safety

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

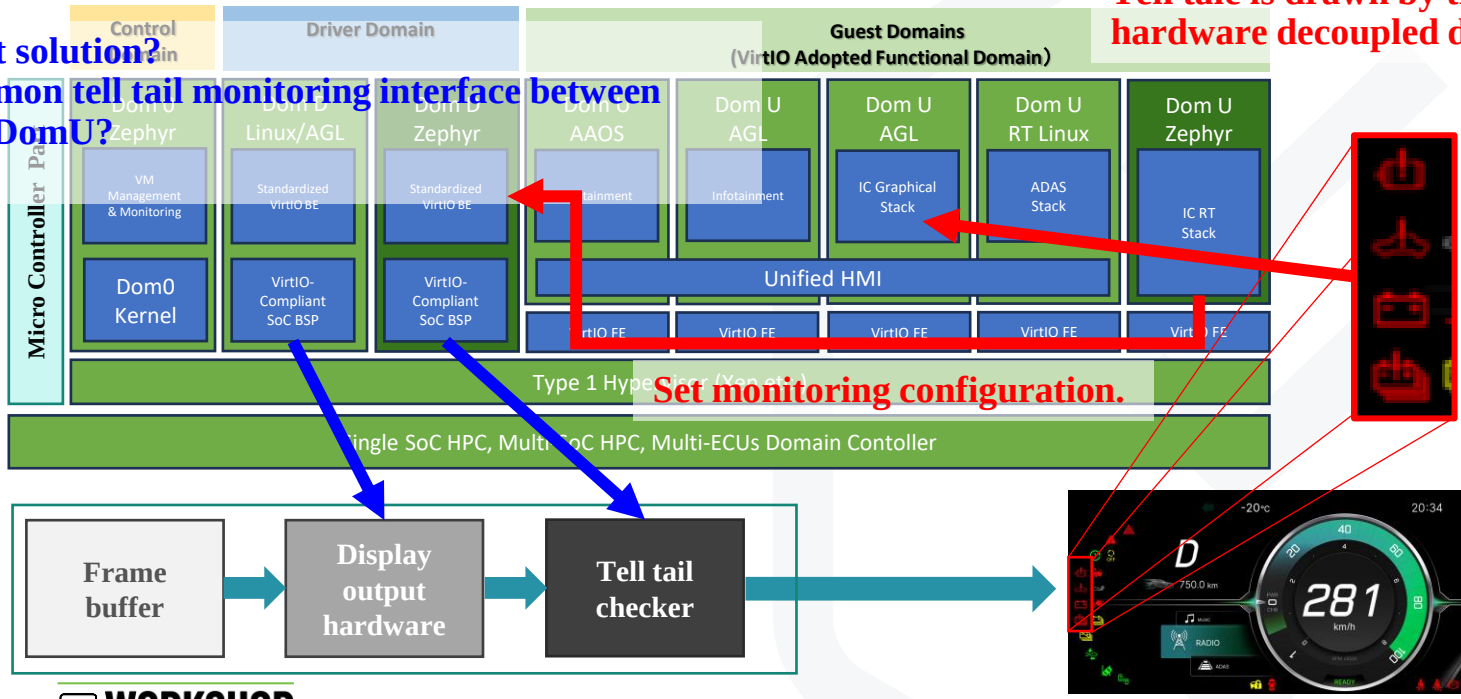


Antinomy between SoDeV and Safety

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

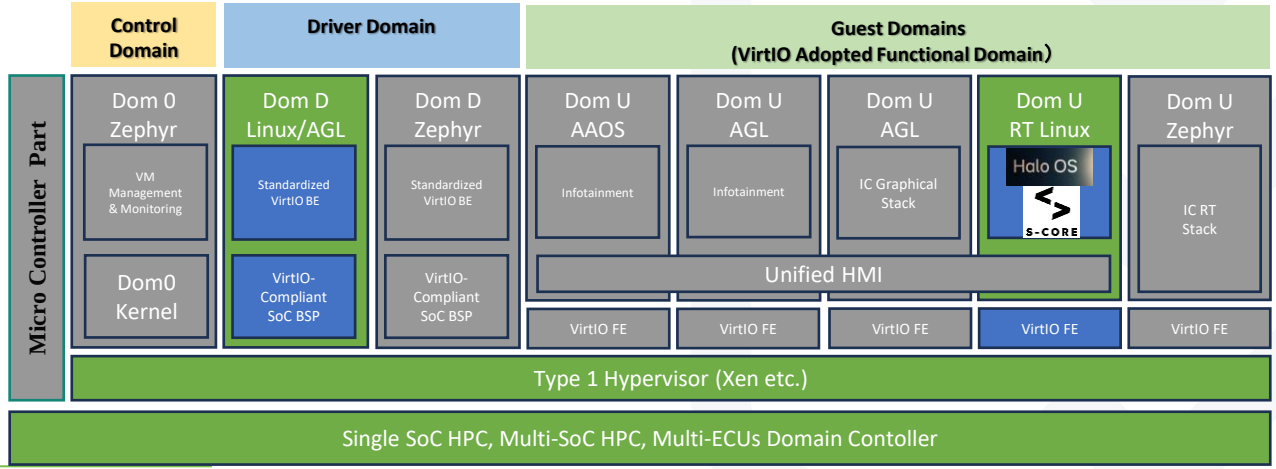
What is best solution?
Create common tell tail monitoring interface between DomD and DomU?
Others?

Tell tale is drawn by the hardware decoupled domain.



Antinomy between SoDeV and Safety

- AGL doesn't have a deterministic runtime environment.
  - That point works by some communities such as S-CORE project, Halo.OS project. AGL will follow and integrate with SoDeV.
  - We need to investigate whether these activities are possible to reuse on VirtIO runtime environment.



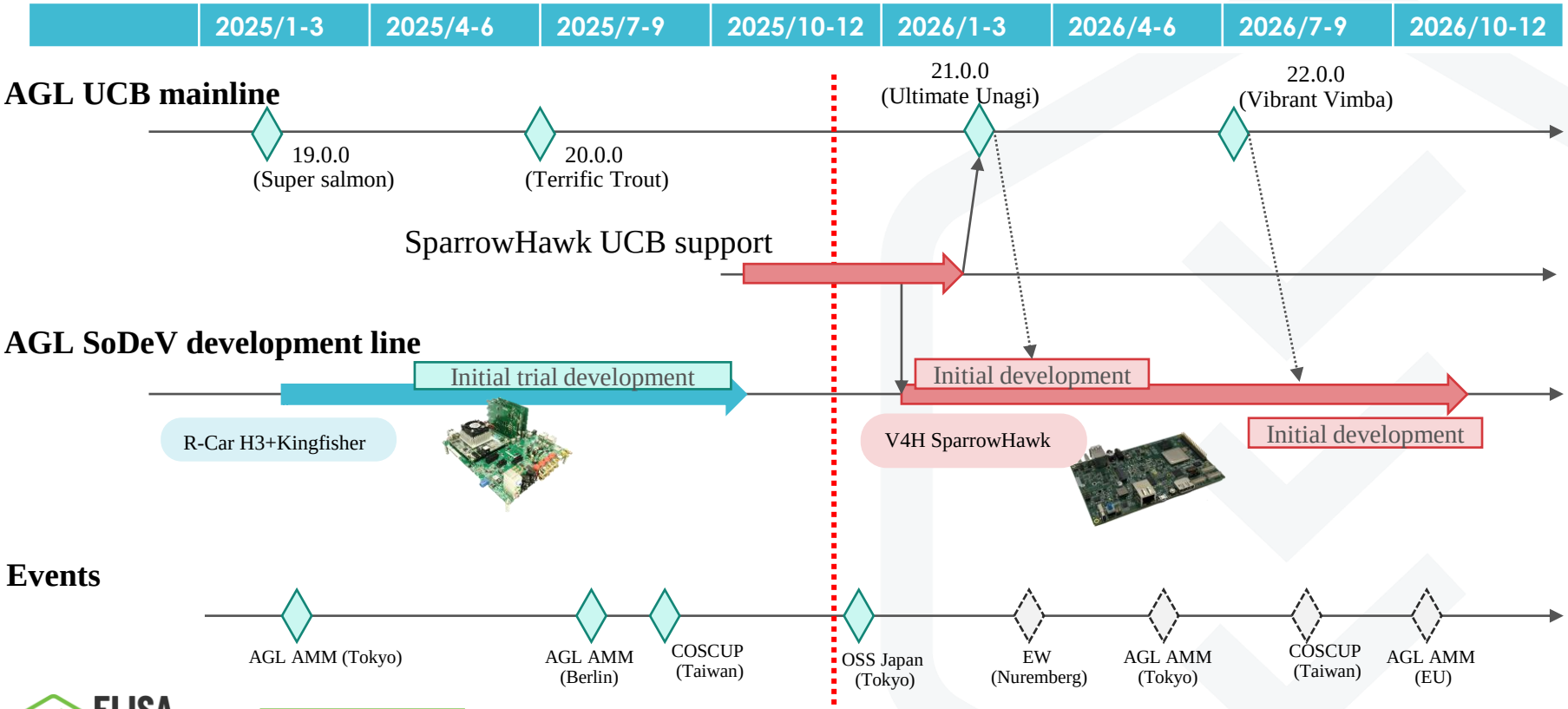
AUTOMOTIVE GRADE LINUX

Our development activity

ELISA
Enabling Linux in Safety Applications

WORKSHOP

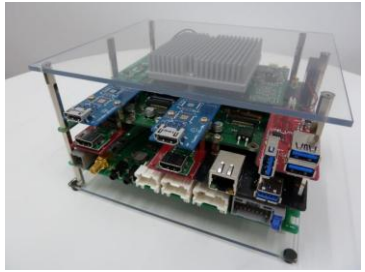
AGL UCB and SoDeV schedule



ELISA Enabling Linux in Safety Applications

WORKSHOP

SoDeV Reference HW : Sparrow Hawk



R-Car H3 AGL RefHW	Function	V4H Sparrow Hawk
4x Arm® Cortex®-A57 4x Arm® Cortex®-A53 (Armv8-A / GIC v2)	**CPU**	**4x Arm® Cortex®-A76 3x Arm® Cortex®-R52 (Armv8.2-A / GIC v3)**
N/A	**CNN-IP**	**30TOPS(Dense)**
PowerVR GX6650	**GPU**	**PowerVR AXM-8-256**
8GB LPDDR4	**DRAM**	**8GB/16GB LPDDR5**
16MB QSPI	**Flash Memory**	**64MB QSPI**
N/A	**Camera I/F**	**2x Raspberry Pi Camera**
3x HDMI output	**Display**	1x DP, 1x Pi Display
USB3.0 x 2 / USB2.0 x 2	**USB**	**4x USB 3.0 Port**
1x MicroSD	**Removal Media**	1x MicroSD
154mm x 178mm x 101mm	**Size**	**146 x 90 mm**

https://wiki.automotivelinux.org/_media/eg-rhsa/rh_manual_ver.1.1.pdf

https://www.retronix.com.tw/en/product_sbc.html

ELISA Enabling Linux in Safety Applications

WORKSHOP