SOAFEE

Blueprints - Decoupling Cockpit Software and Hardware

Introduction of

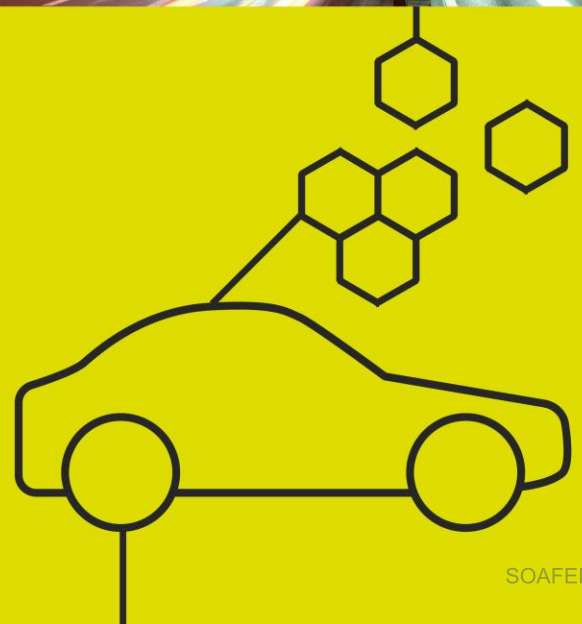
Unified HMI and vSkipGen

Jerry Zhao,
Chief SDV Architect
Panasonic Automotive Systems Co., Ltd.

2025/11/12

SOAFEE

- Background of SDV and Software-Hardware Decoupling
- Unified HMI for seamless Multi-ECU/OS Graphics
- VirtIO-based vSkipGen for Cloud-Native Development
- Future Vision with SOAFEE



SOAFEE

Background of SDV and Software-Hardware Decoupling

Atsuya Nasu, Principal Software Engineer at
Panasonic Automotive Systems Co., Ltd.

2025/5/15

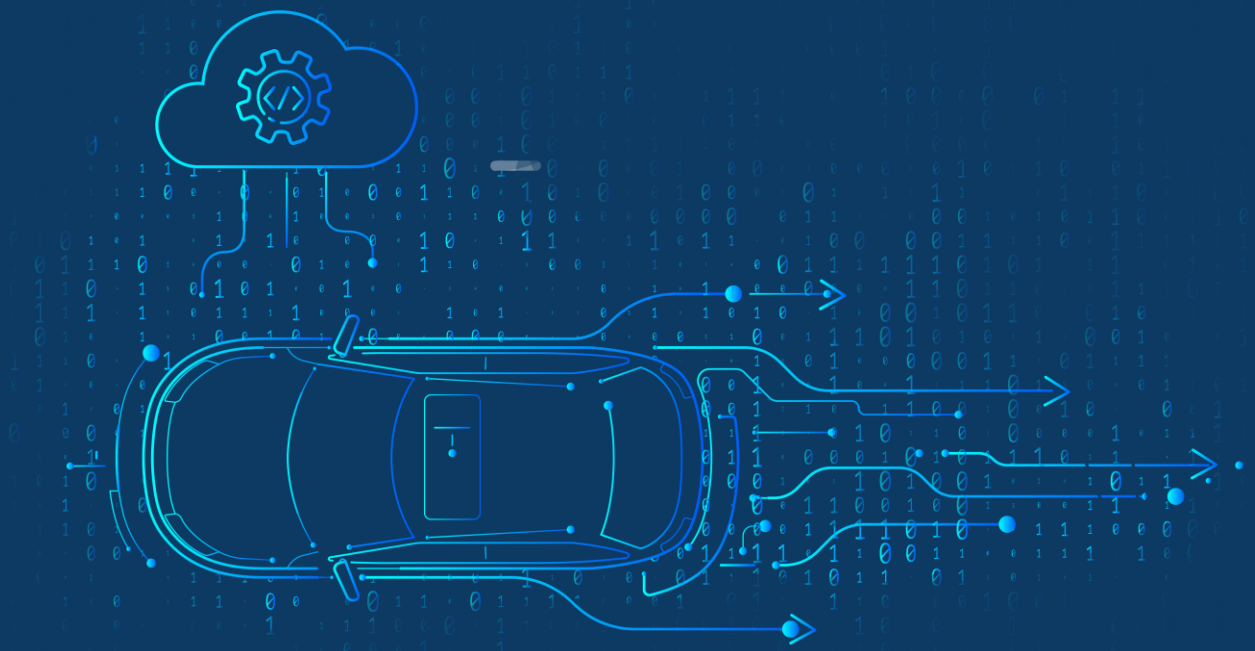
Industry Trend with SDV

Software-Defined Vehicles

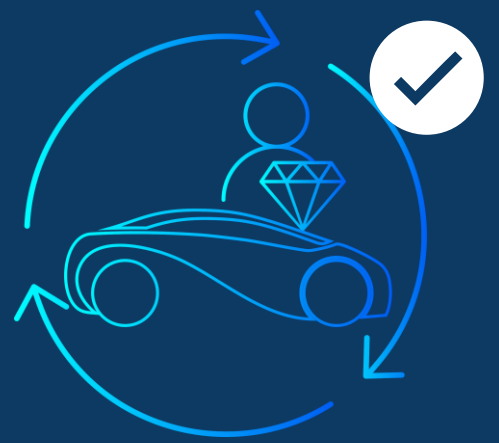
Drastic Changes
in EE Architecture

Open Source,
De Facto Standard

Connected
& AI-powered

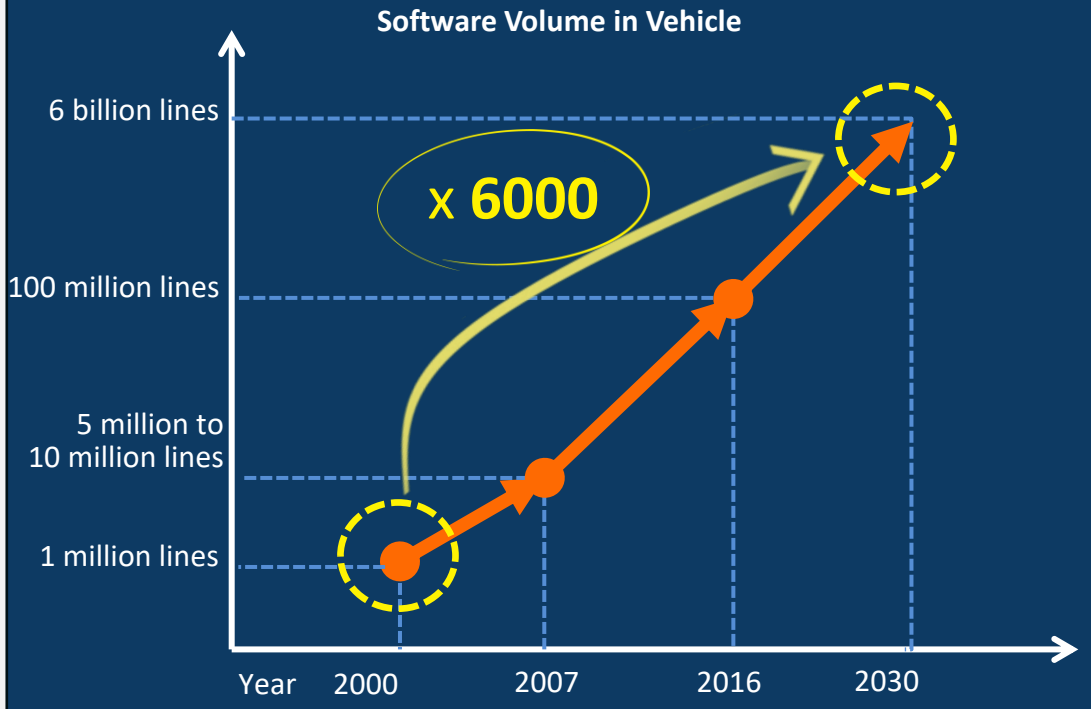


Speedily Delivered Values



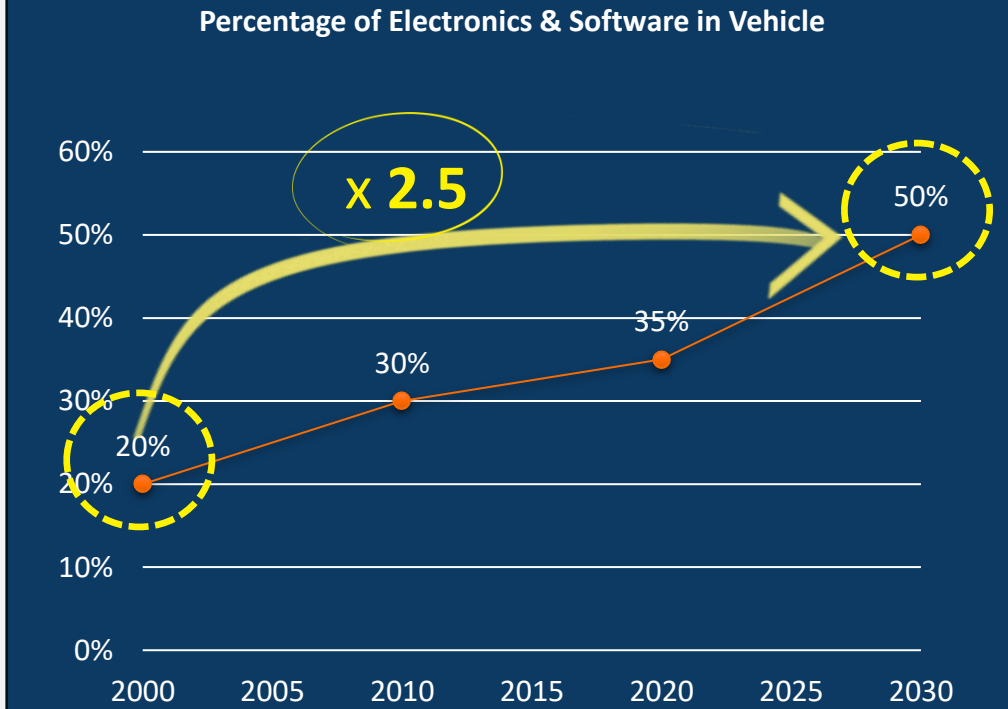
Shift to SDV Industry Trend

Increased Complexity of Vehicle Software (*1)



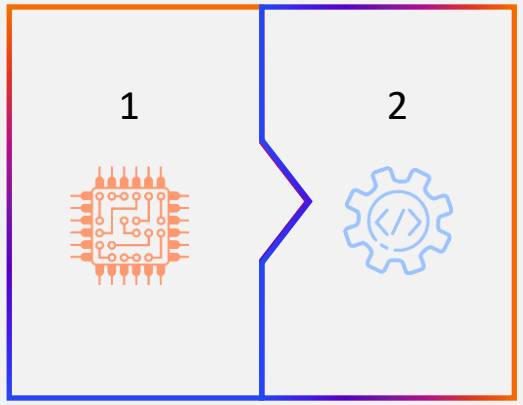
*1: Source: Ministry of Economy, NXP Semiconductors, Quora, Ignition in Action, NYC AVITAION, Trade and Industry "Toward acceleration of productivity improvement by IT" Mitsubishi UFJ Morgan Stanley Securities' materials, etc.

Increased Cost Contribution of Vehicle Software (*2)



*2: Source: Lux Research

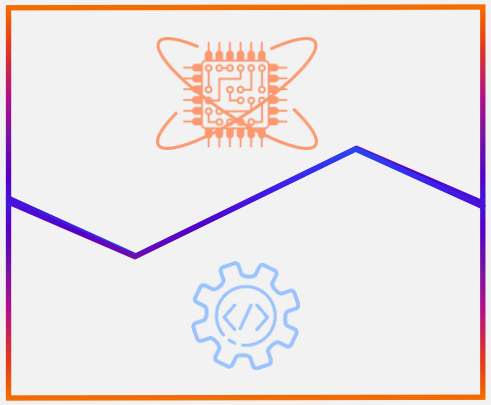
From Hardware First To Software First



Traditional

Manufacture HW prototype and develop SW

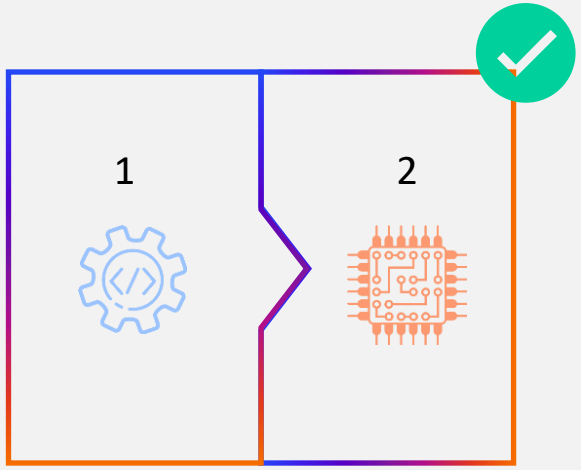
- Long wait time for limited HW
- High sample cost



HW Emulation

Emulate HW and develop SW simultaneously

- Limited to low-level SW & HW
- Costly & time-consuming



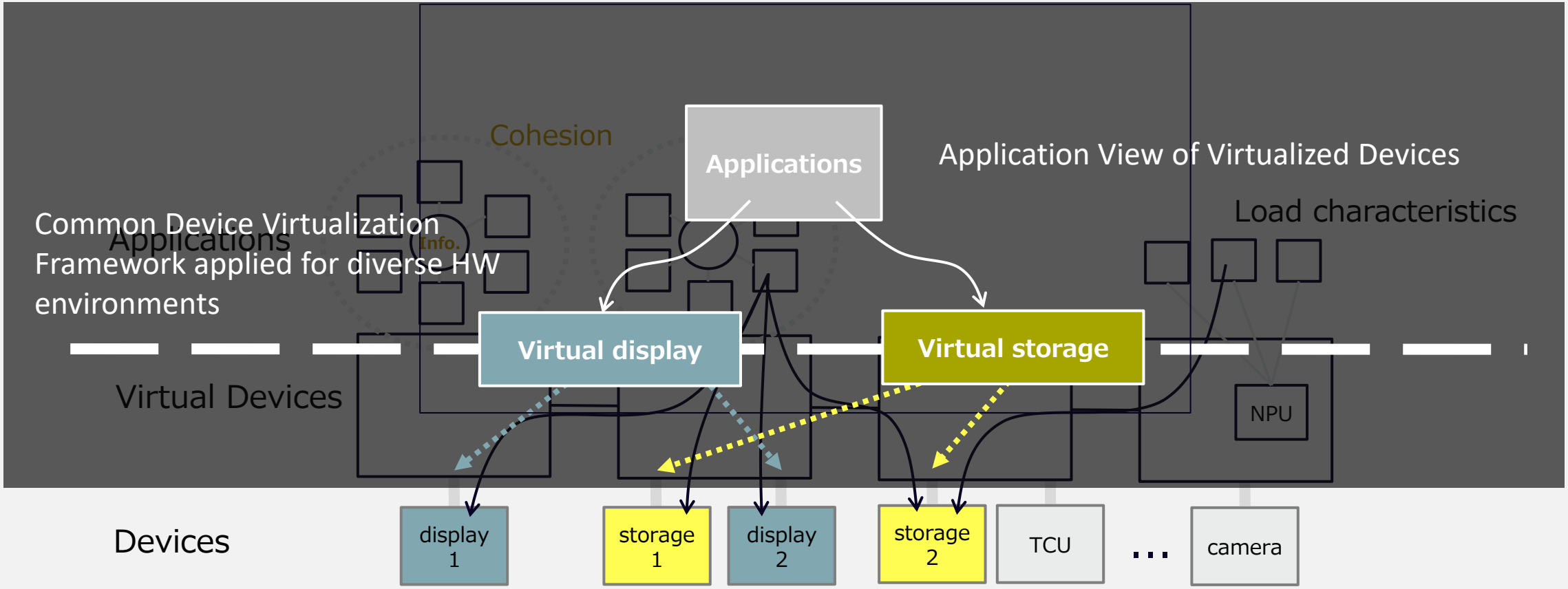
Cloud-Native

Develop SW on Cloud and select optimal HW

- Rapid function update
- Scalable for large-scale development

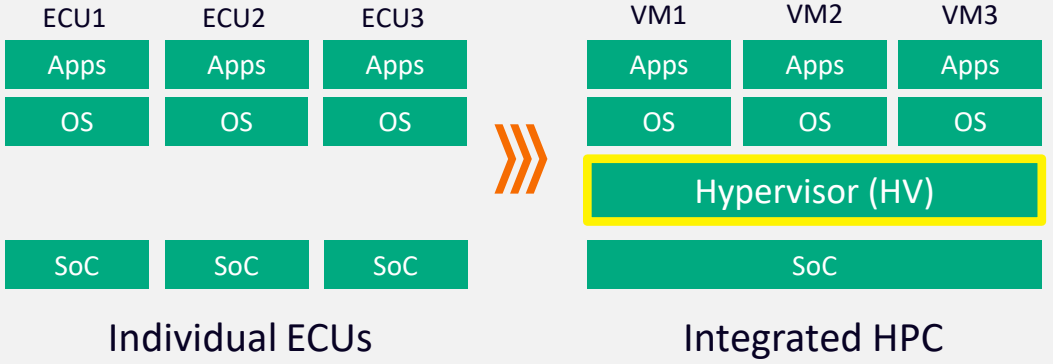
Device Virtualization: Key to SDV

Software Defined Vehicle needs a common device virtualization framework to decouple software implementation from diverse hardware targets across vehicle variants/generations, architectures (single/multiple-ECU) and development environments (real/virtual ECU)

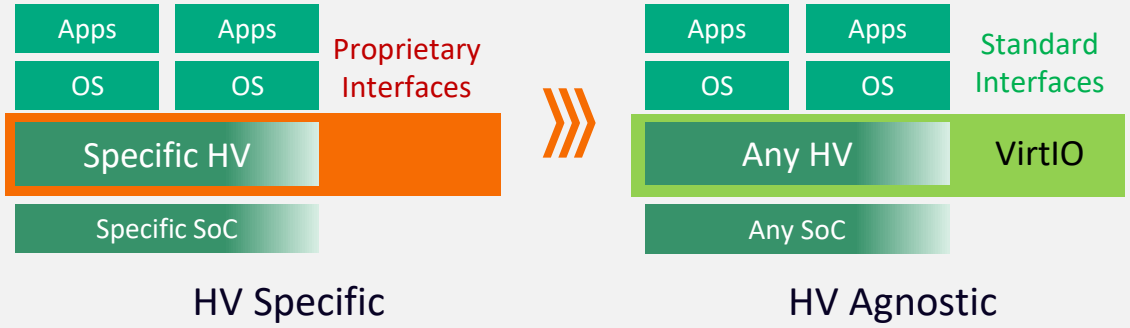


Decouple Software and Hardware

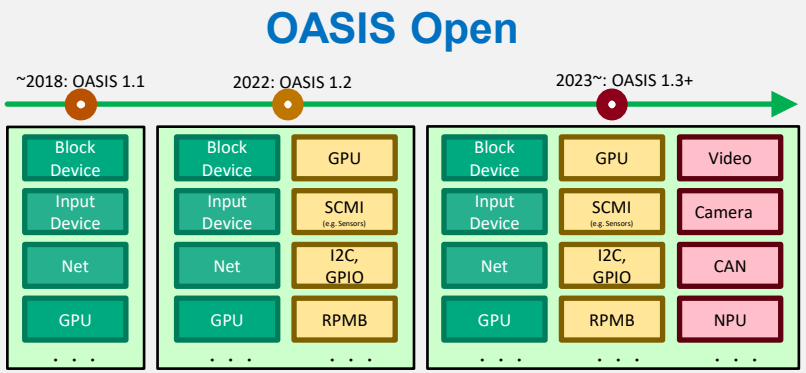
Consolidation requires Hypervisor



Hypervisor needs Standard Interfaces



Standard Specification



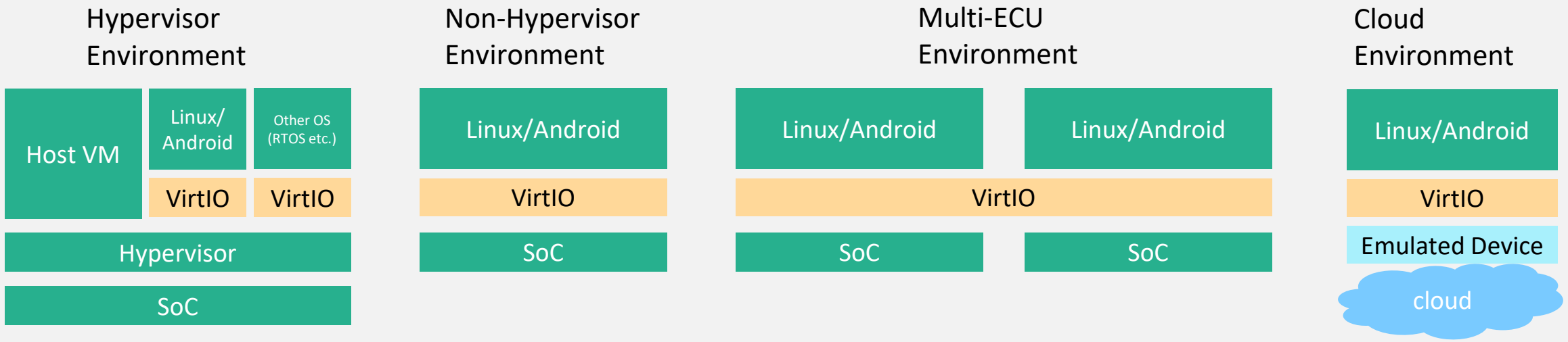
Standard Implementation

AUTOMOTIVE GRADE LINUX

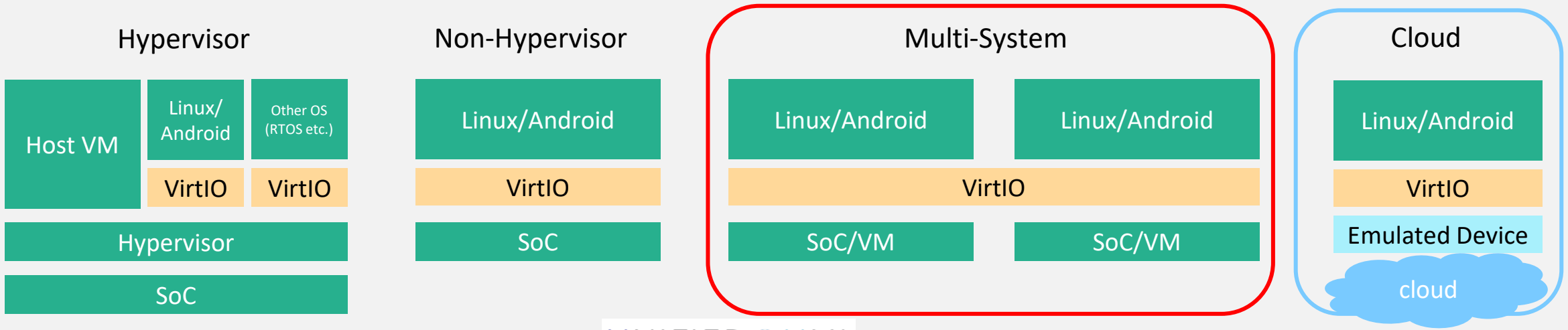
*Android Robot is copied or modified from a work created and provided by Google and used in accordance with the terms set forth in Creative Commons 3.0 License.

Overview of Device Virtualization - Concept

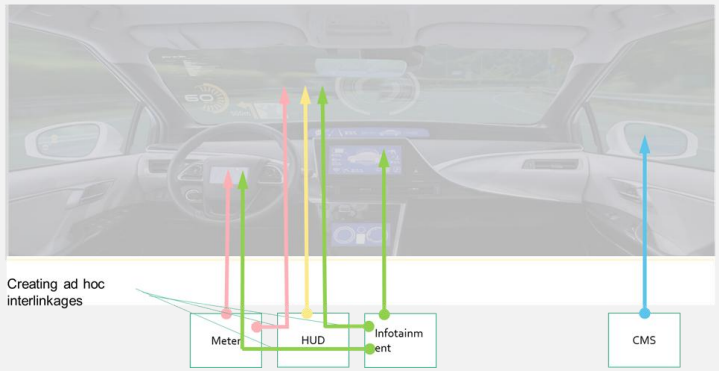
Device Virtualization with VirtIO benefits in establishing a complete and healthy ecosystem to enhance interchangeability and interoperability in various scenarios.



Concept of SDV Solutions with Unified HMI and vSkipgen

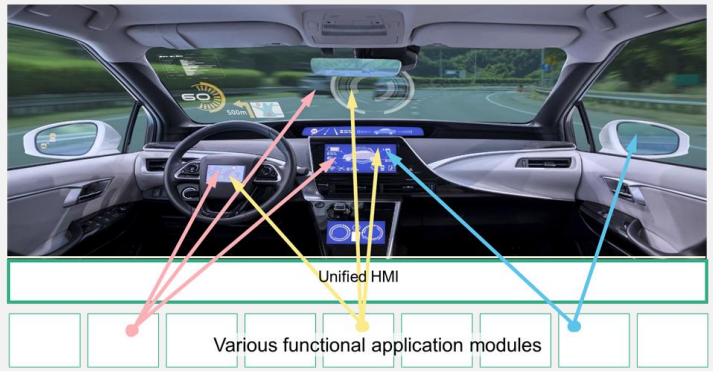


Legacy HMI System



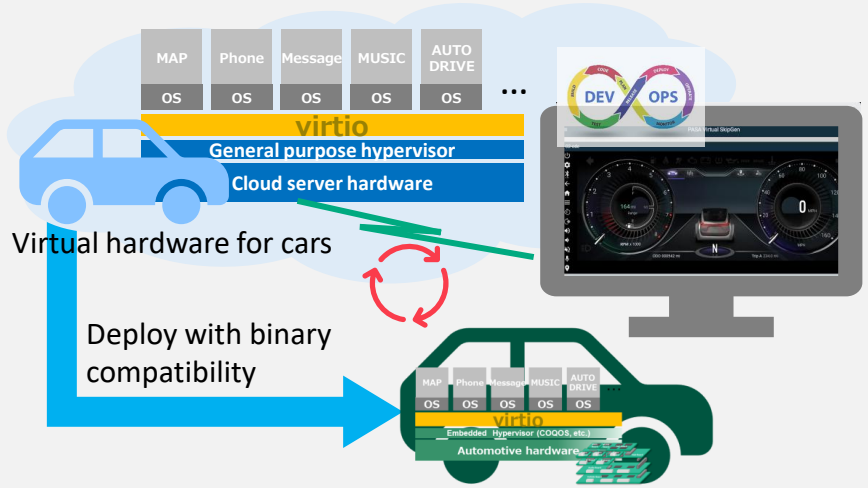
Strict Restriction on ECU & Function-Display Relationship causing harmful impediment for Cockpit UX.

UNIFIED HMI



Full Flexibility on ECU & Function-Display Relationship for Cockpit UX Innovation.

vSkipGen

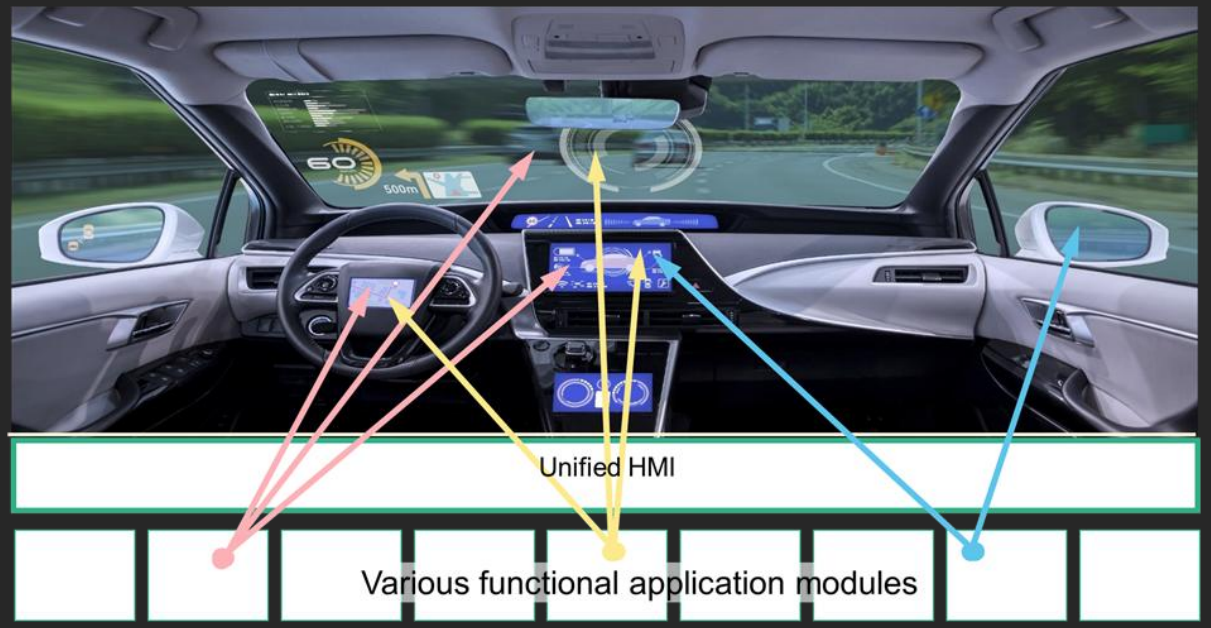


High-efficiency development with Cloud-Native

SOAFEE

Unified HMI for seamless Multi-ECU/OS Graphics

UNIFIED HMI



Trends in the Automotive Industry

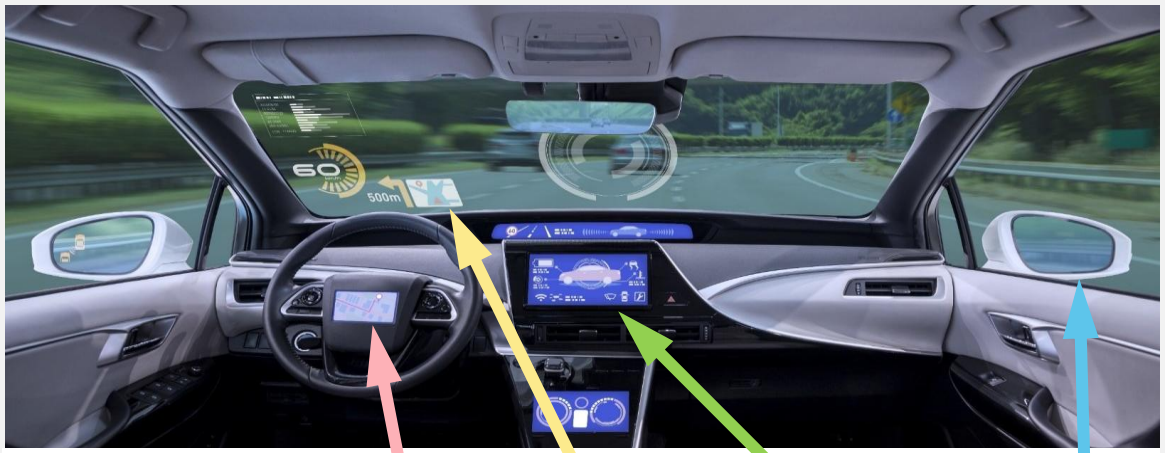
- The increasing number of in-vehicle displays has created demands for flexible application display across multiple displays, introducing new UI/UX possibilities.
- However, developing this flexibility using existing graphic frameworks is costly.

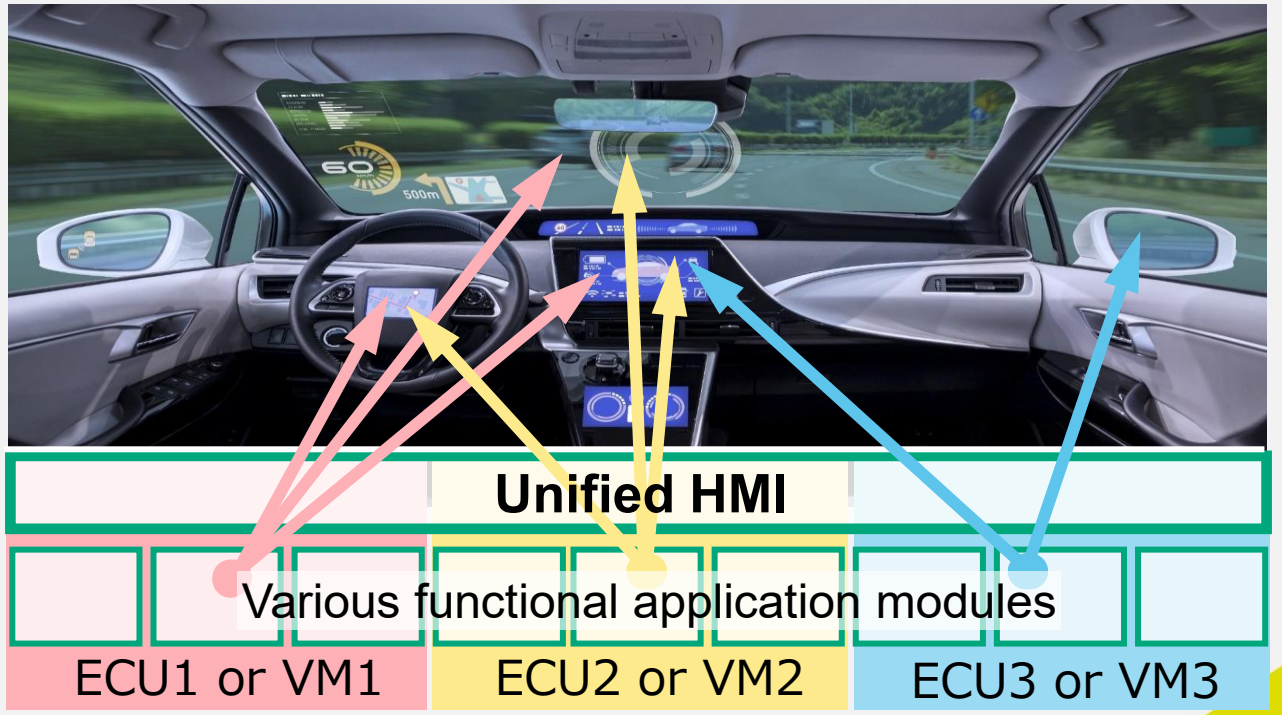
  => Needs a "Software-Defined" display framework for decoupling software and hardware.

What is Unified HMI

"**Software-Defined**" display virtualization platform based on VirtIO GPU.

- ✓ Enable flexible development of the entire cockpit UI/UX across multiple displays **independent of hardware and OS configurations**.
- ✓ UI/UX can be developed in physical and virtual environments, seamlessly deploying to ECUs.





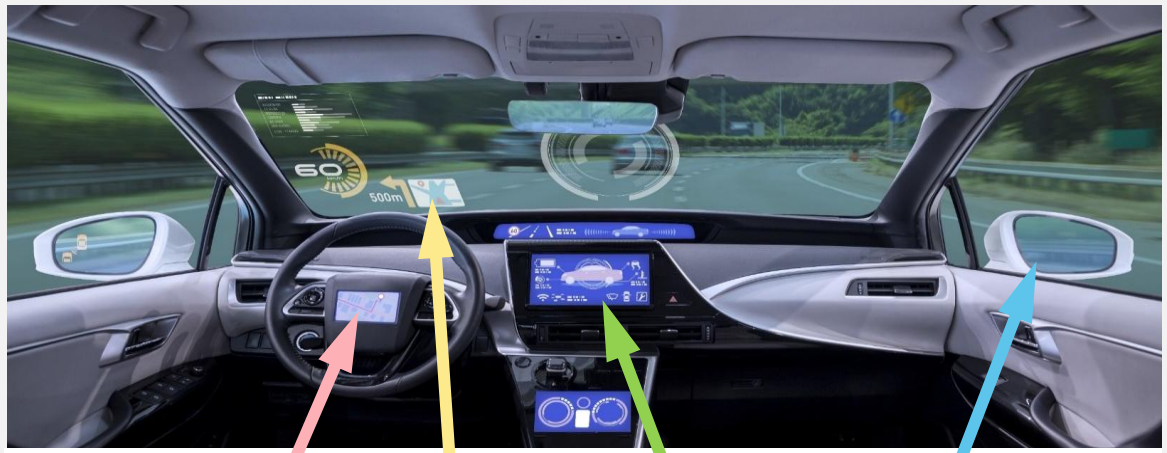
UNIFIED HMI Solution

require ad hoc coordination

require ad hoc coordination

require ad hoc coordination

require ad hoc coordination

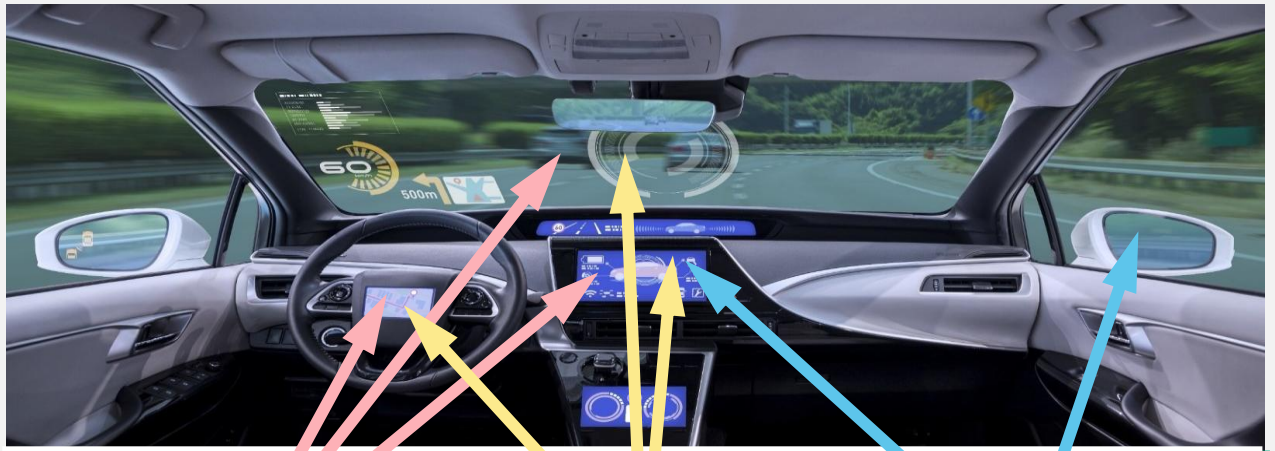


ECU1 Meter

ECU2 HUD

ECU3 infotainment

ECU4 CMS



Value Unified HMI provides

For Automotive Developers

Agile & Software-Defined Cockpit UI/UX Development

- Efficient and integrated cockpit UI/UX development & evaluation on virtual environment
- Scalable to deploy seamlessly to various car grades/models

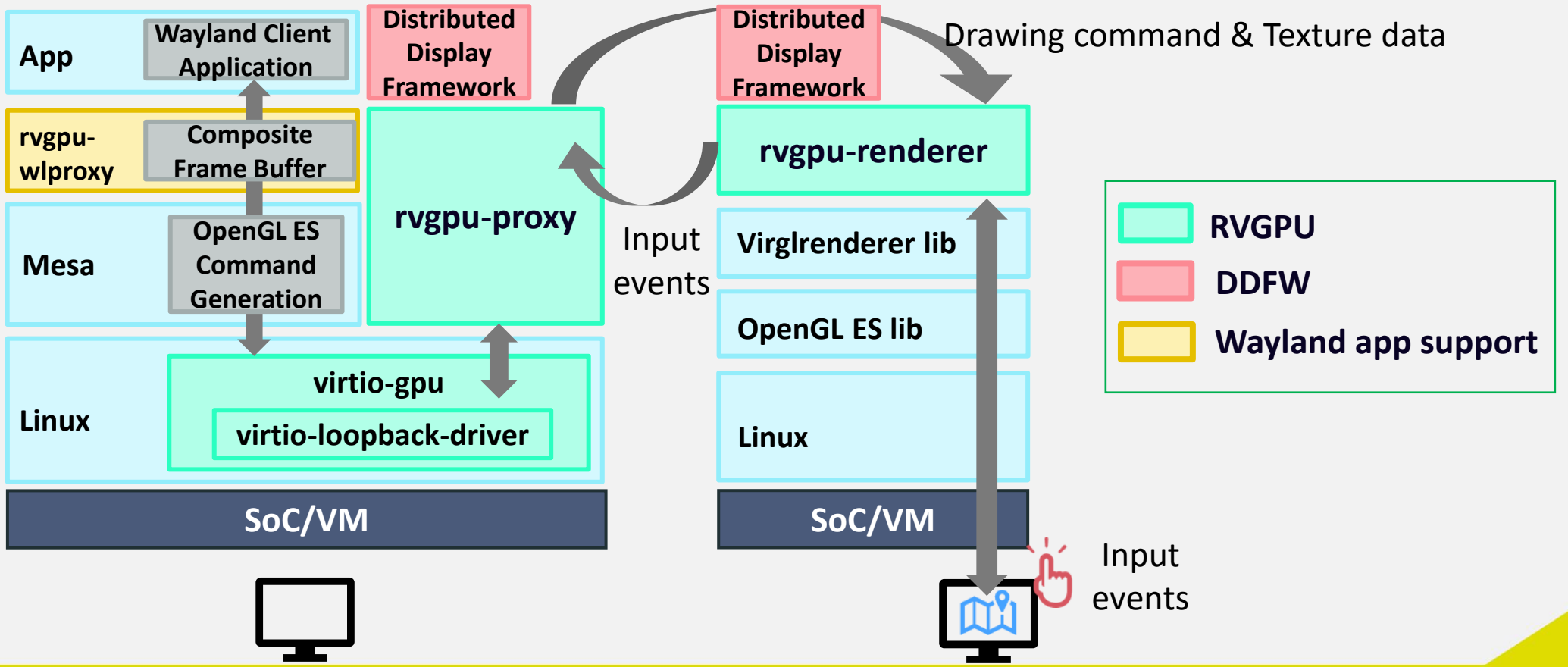
For Automotive Users

Fast-evolving & Personalized Cockpit UI/UX

- Upgraded customer experience from frequent OTAs on UI/UX improvements
- Cockpit UI/UX can be customized flexibly according to user preference no matter of car grades/models

Unified HMI architecture

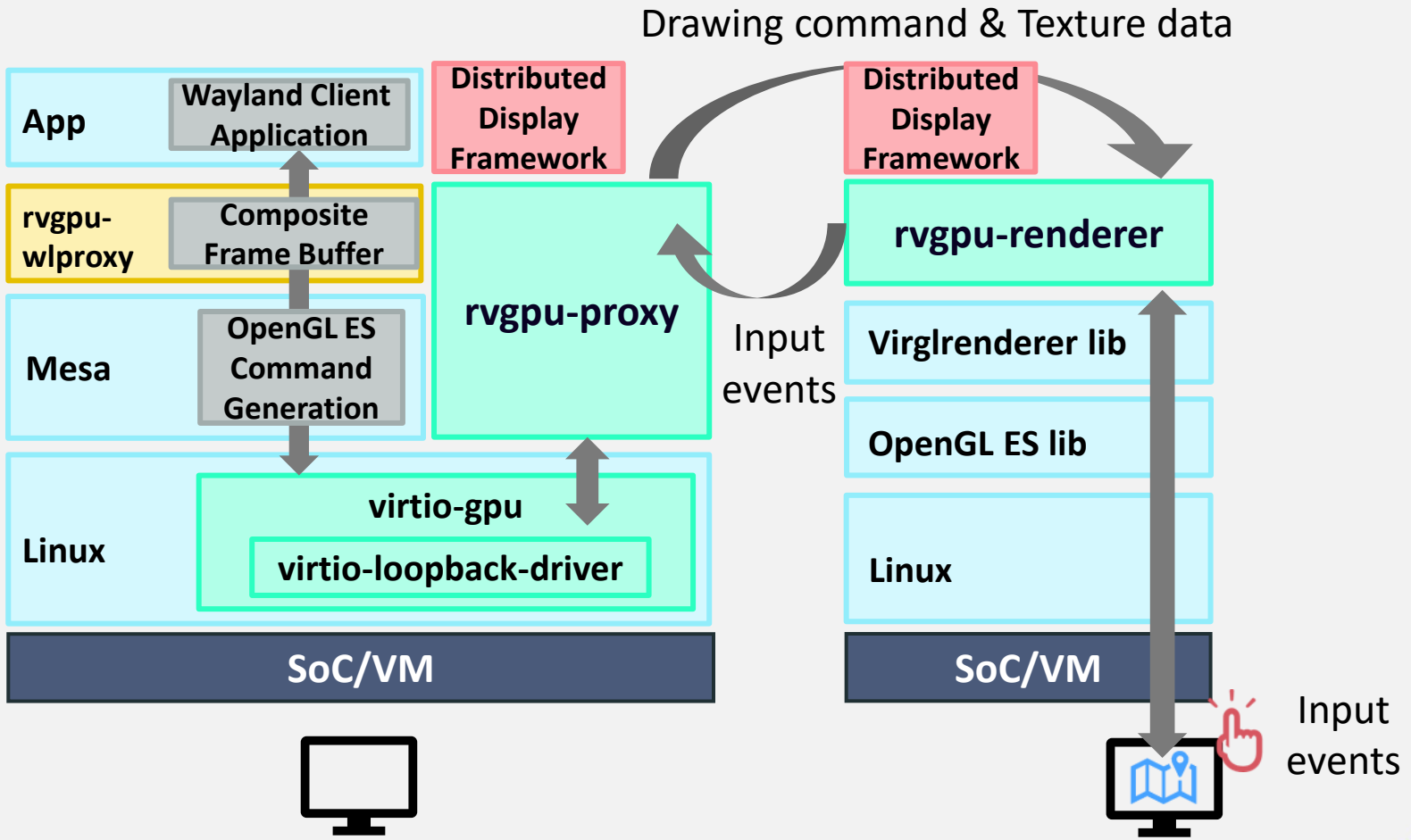
1. Remote VirtIO GPU Device(RVGPU)：Render apps remotely in different SoCs/VMs.
2. Distributed Display Framework(DDFW)：Flexible layout control of apps across multi-display



Remote VirtIO GPU Device (RVGPU)

Network extension of **VirtIO GPU** commonly used in GPU virtualization for VM

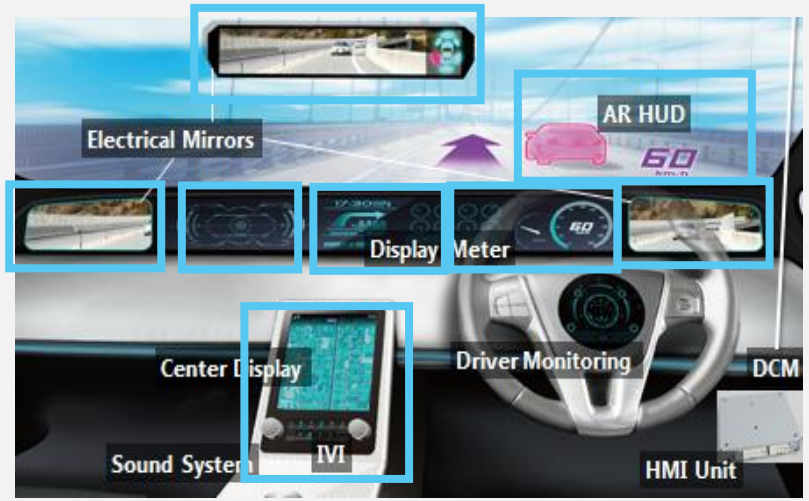
- ✓ **virtio-loopback-driver**: Transmit GPU commands generated by OpenGL ES to rvgpu-proxy
- ✓ **rvgpu-proxy** : Send GPU commands to rvgpu-renderer on any SoCs/VMs
- ✓ **rvgpu-renderer** : Receive GPU commands and draw graphics



Distributed Display Framework (DDFW)

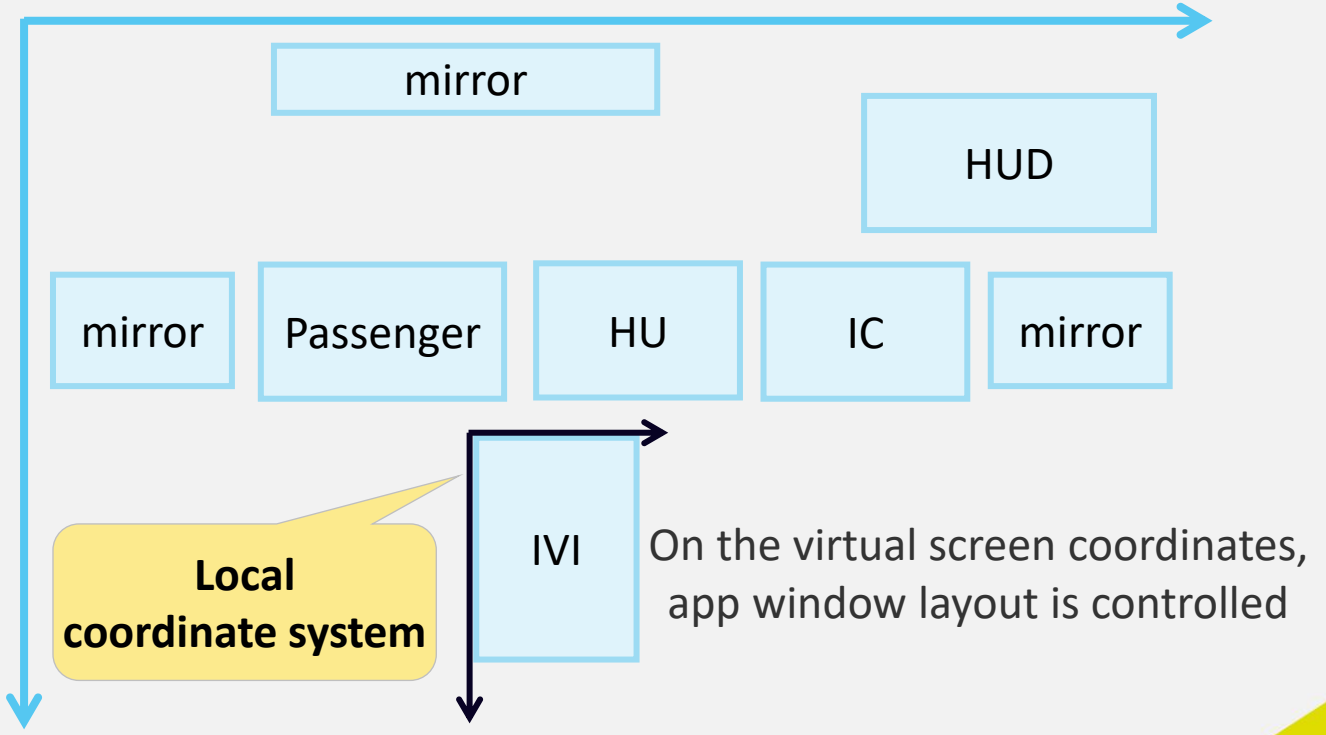
- ✓ Mapping multiple cockpit physical displays into a single large virtual screen
- ✓ Control layout such as location, size, and display order of multiple apps

Multiple **physical displays** inside the viecle



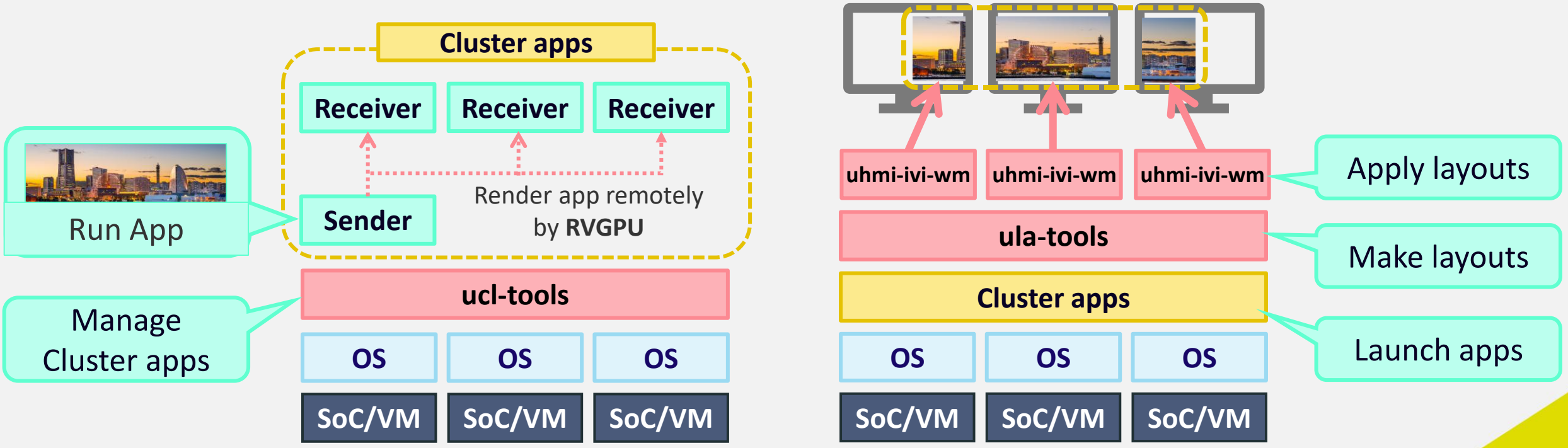
mapping

Virtual Screen



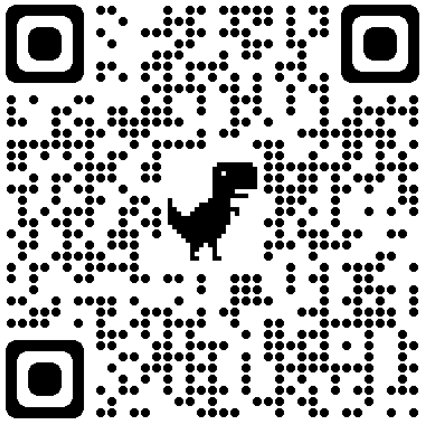
Components of DDFW

- ✓ Unified Clustering Tools(ucl-tools): Ensures group operation of sender/receiver apps (Clustering).
- ✓ Unified Layout Tools(ula-tools): Provides unified control of the display layout for cluster apps.
- ✓ Unified HMI IVI Window Manager(uhmi-ivi-wm): Apply ivi-layers and ivi-surfaces layouts made by ula-tools to the screen.



How to access Unified HMI

GitHub



- ✓ Access the source code for Unified HMI
- ✓ Compatible with Ubuntu for easy development
- ✓ Contribute to the project and report issues

AUTOMOTIVE GRADE LINUX

- ✓ Unified HMI is a standard component of AGL
- ✓ Includes meta-uhmi Yocto recipe for embedded integration
- ✓ Supported Platforms, QEMU, Raspberry Pi 4, and AGL Reference Hardware

How to use Unified HMI

Sender side: Ubuntu

Receivers side: AGL

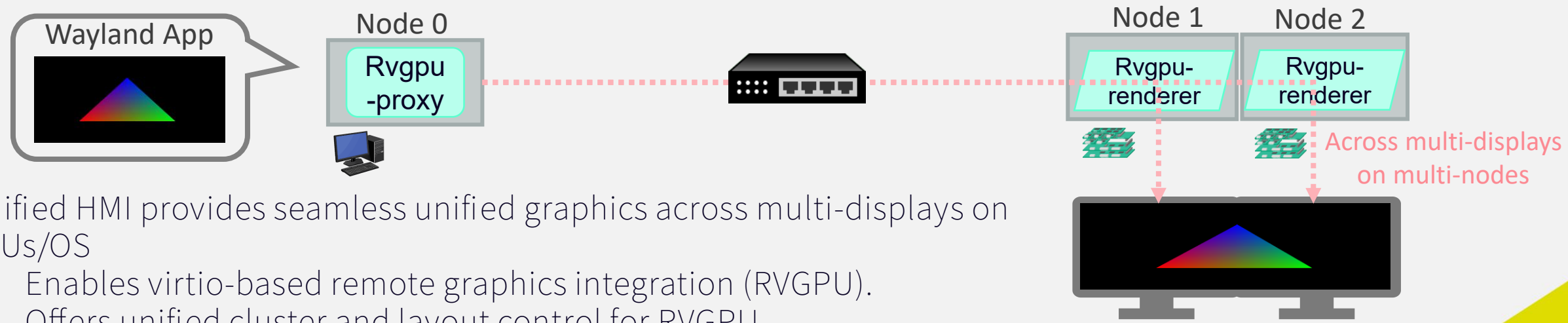
cmd

```
$ ucl-virtio-gpu-wl-send                ¥
        -s 1920x1080@0,0                ¥
        -n <IP Address>:<Port>          ¥
        -n <IP Address>:<Port>          ¥
        <Wayland Application>
```

Multi-Target

```
$ ucl-virtio-gpu-wl-recv                ¥
        -s 960x1080@0,0                 ¥
        -P <Port>
```

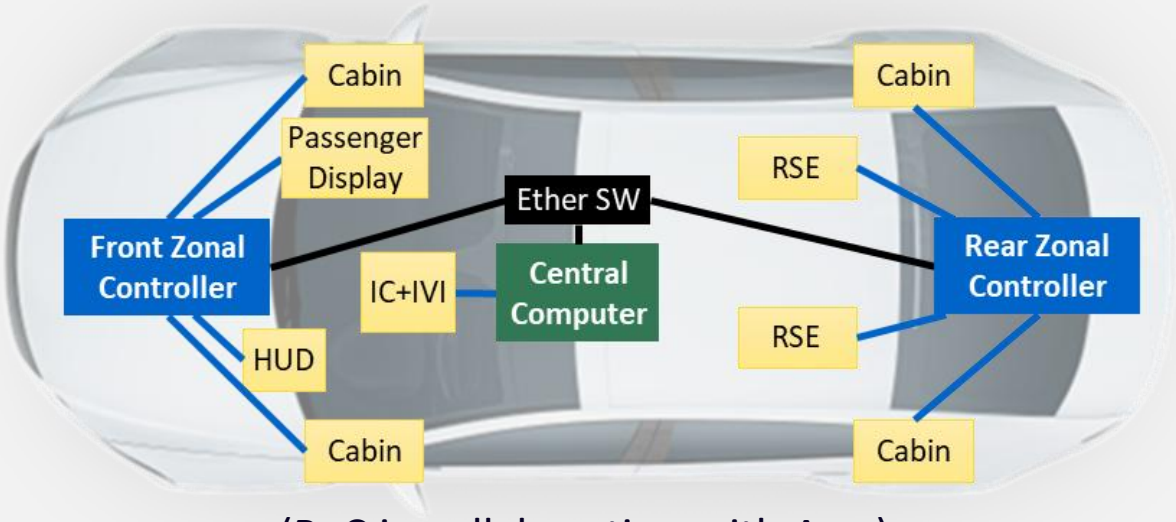
```
$ ucl-virtio-gpu-wl-recv                ¥
        -s 960x1080@960,0               ¥
        -P <Port>
```



- Unified HMI provides seamless unified graphics across multi-displays on ECUs/OS
  - Enables virtio-based remote graphics integration (RVGPU).
  - Offers unified cluster and layout control for RVGPU.

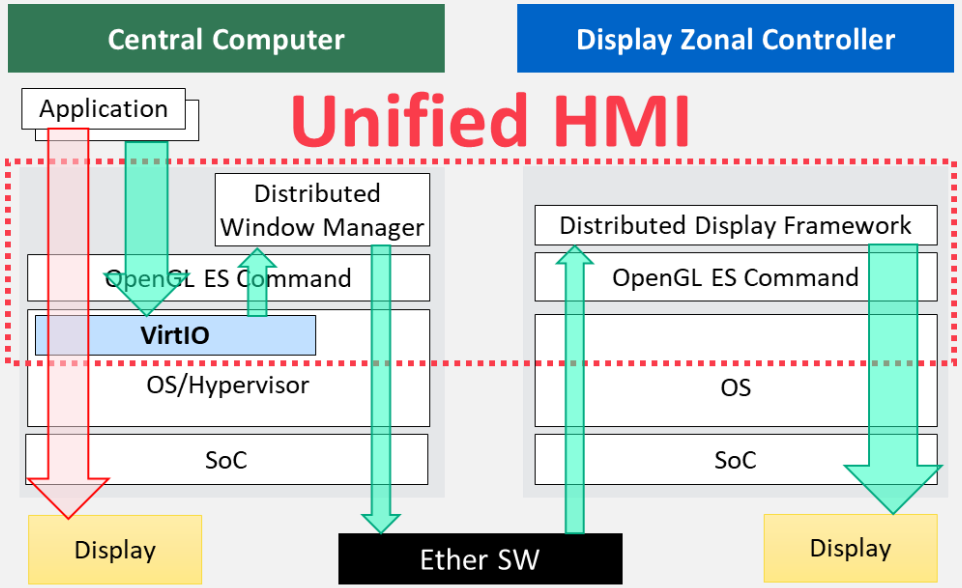
Use case of Unified HMI for Display Zonal Architecture

Concept of Display Zonal Architecture



(PoC in collaboration with Arm)

Implementation /w UHMI

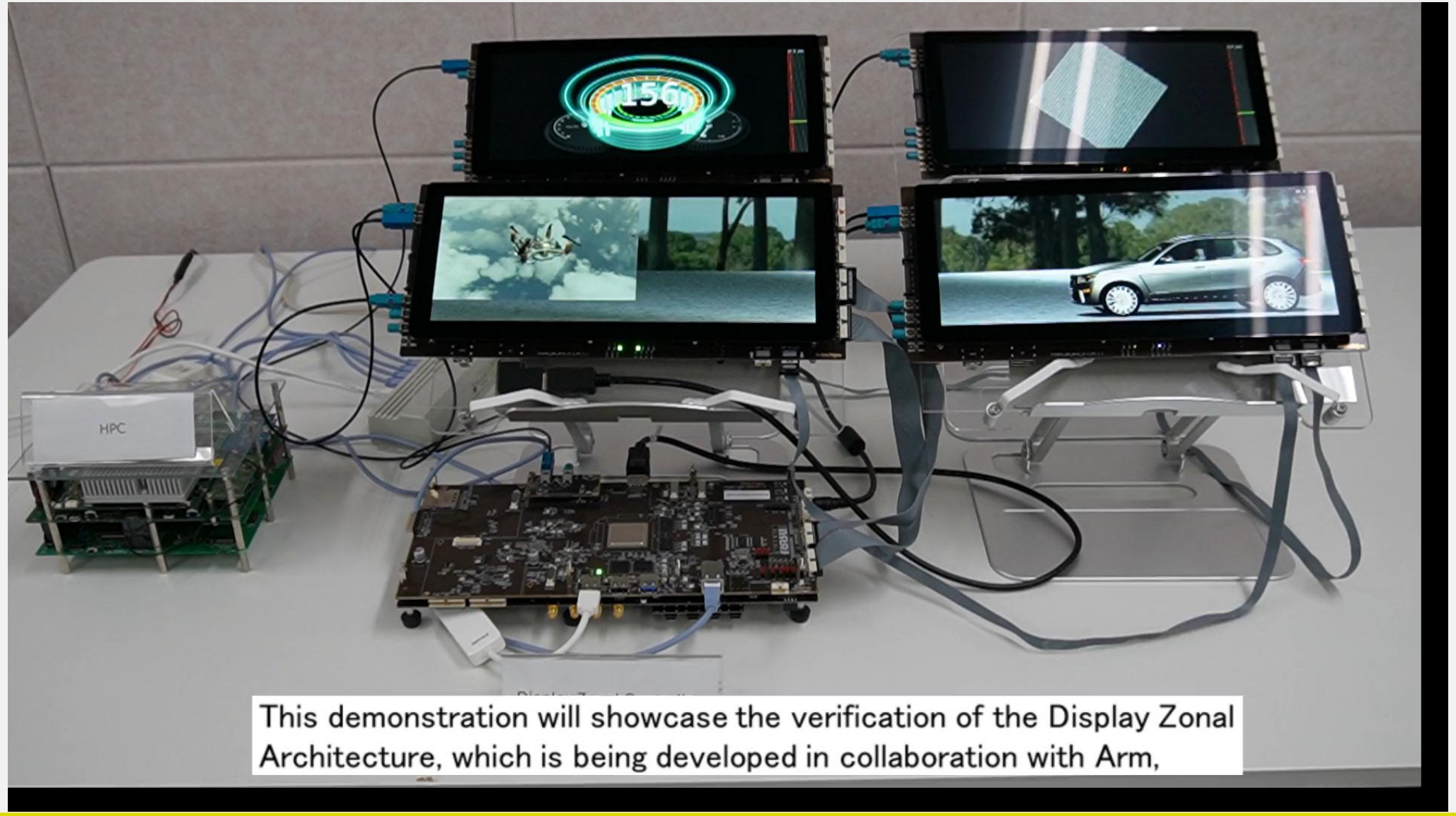


A Flexible Distribution FW:

From Centralization / Distribution (limited choice) to Centralization & Distribution (flexible choices)

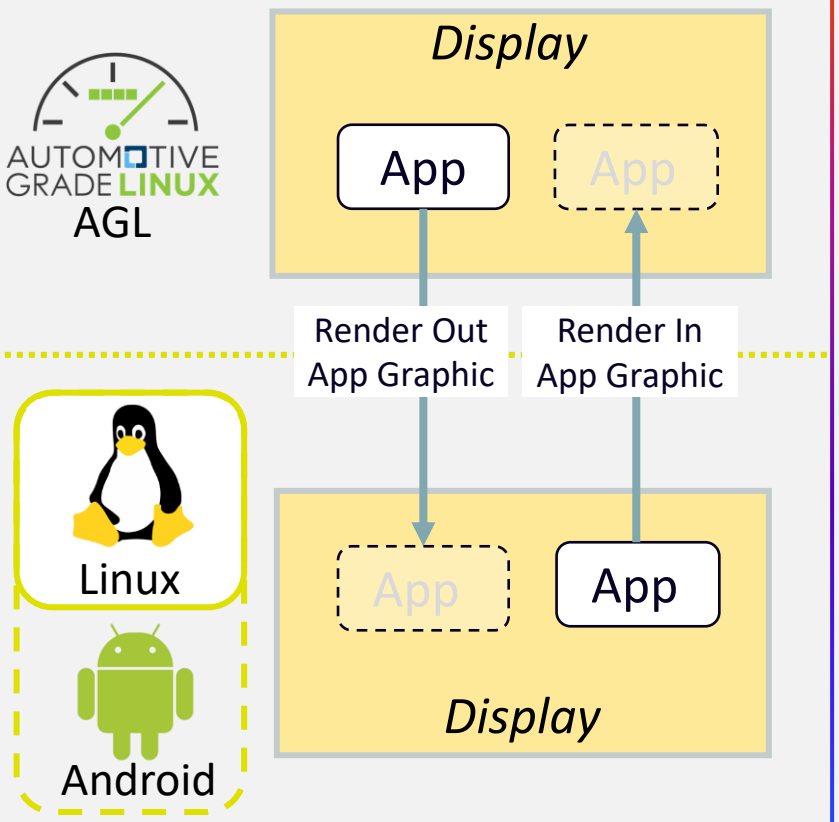
- ✓ Centralize all the applications (cpu computing) and distribute graphic rending (gpu computing) only
- ✓ Centralize at first and distribute flexibly when necessary for later gens.

Unified HMI x Arm PoC for Zonal Display Controller

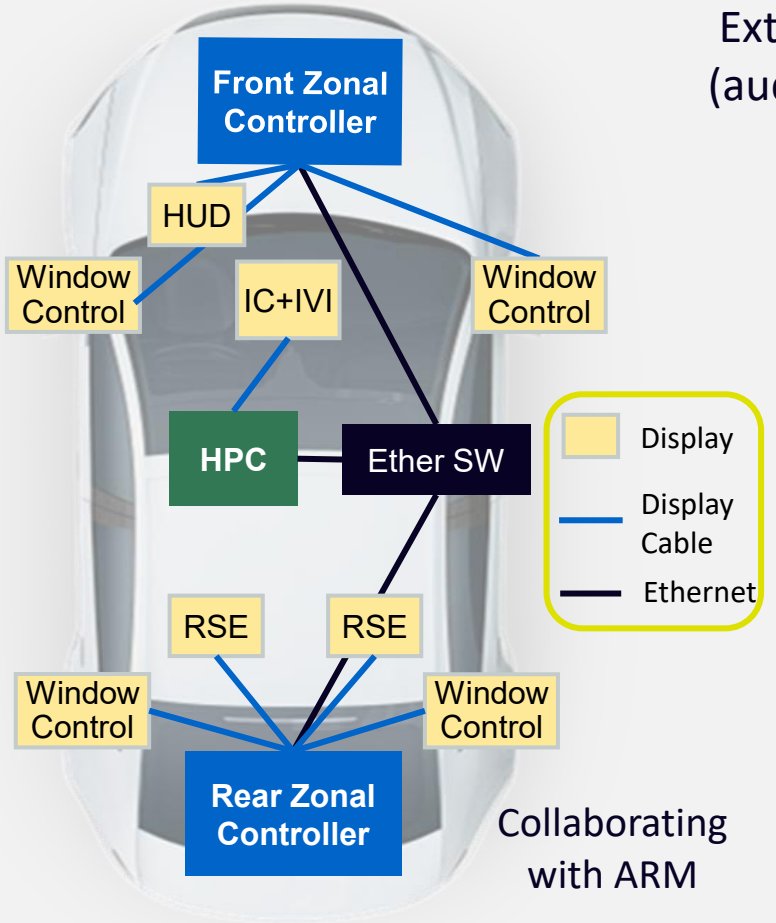


Future vision of Unified HMI

① Enable Application Graphics to Render In/Out between AGL, Android, and others

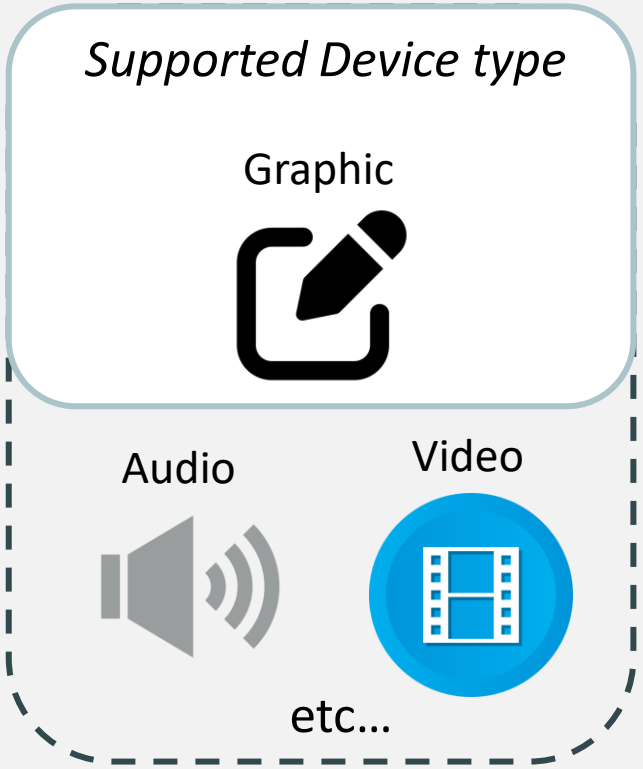


② Future Zonal Architecture



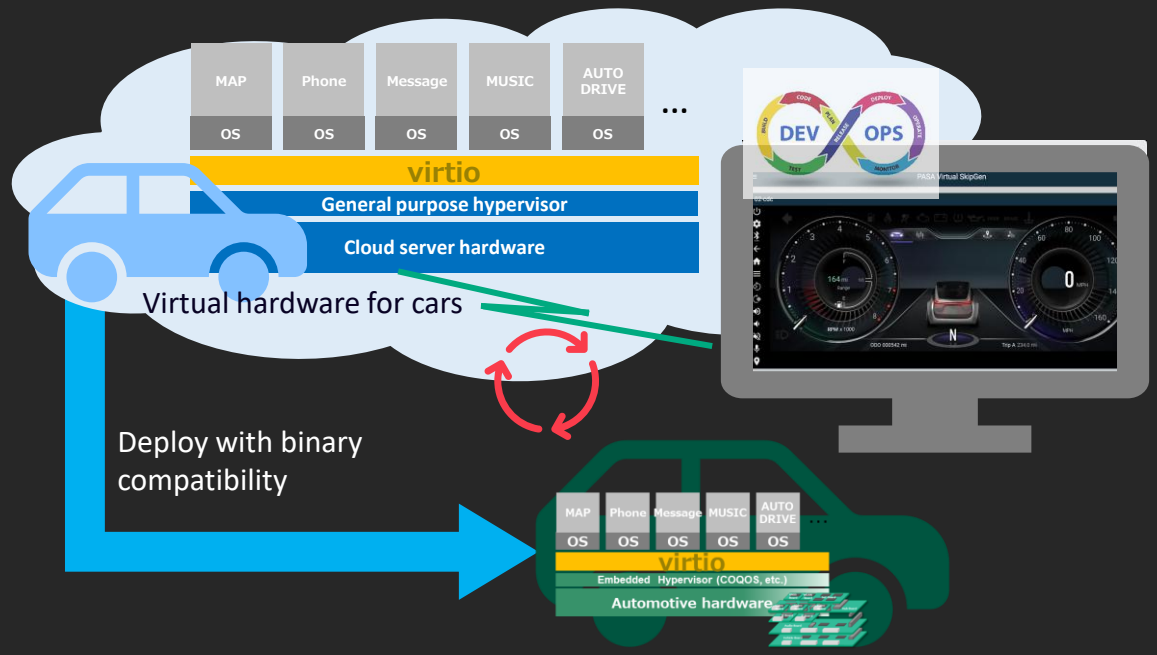
Collaborating with ARM

Extend Unified HMI to more media (audio, video and etc.)

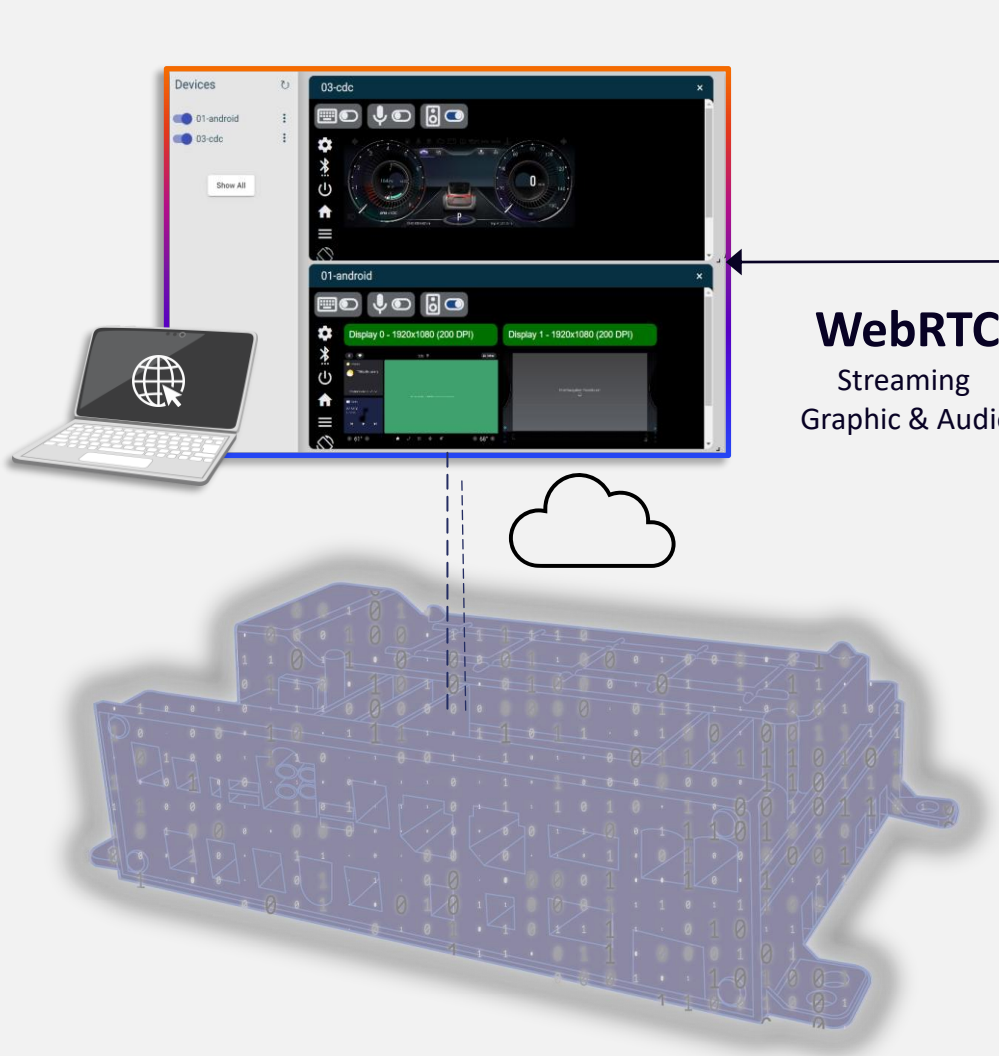


SOAFEE

VirtIO-based vSkipGen for Cloud-Native Development



What is vSkipGen



WebRTC
Streaming
Graphic & Audio

VM1
Cluster
AGL
Kernel
VirtIO: Front End

VM2
IVI
Option A
Android
Kernel
VirtIO: Front End

VM3
IVI
Option B
AGL
Kernel
VirtIO: Front End

VMM
VMM
VMM

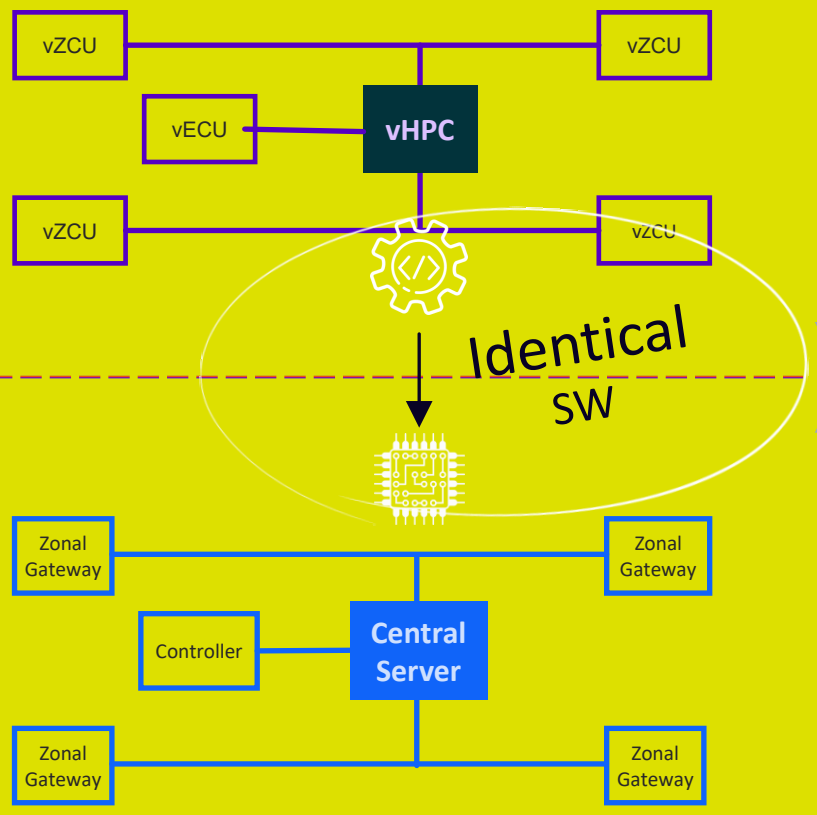
Linux User Space
VirtIO: Back End
Linux Kernel

G5g
NVIDIA GPU

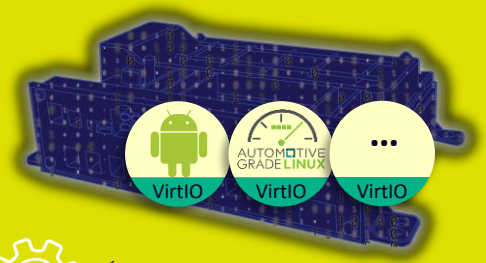
* Android Robot is copied or modified from a work created and provided by Google and used in accordance with the terms set forth in the Creative Commons Attribution 3.0 License.

How it maps the concept of Virtual Hardware

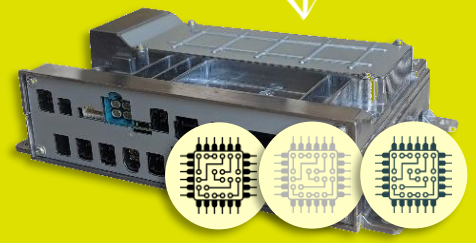
Concept



Solution

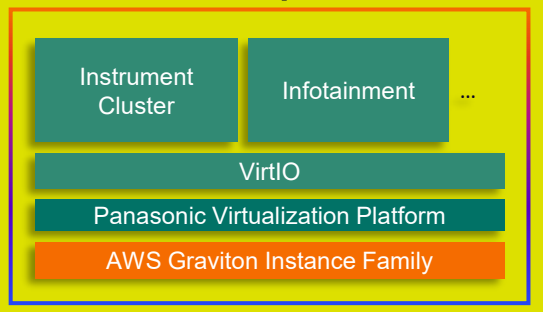


✓Virtually develop & deploy **any OS software** supporting VirtIO

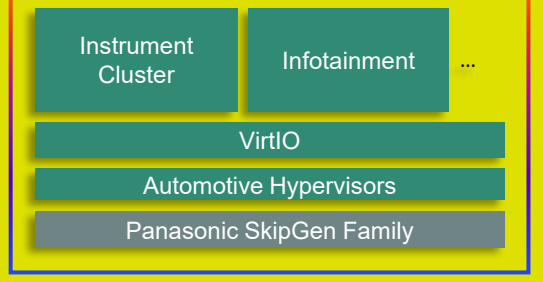


✓Migrate **same binary among various** automotive hardware

Virtual SkipGen

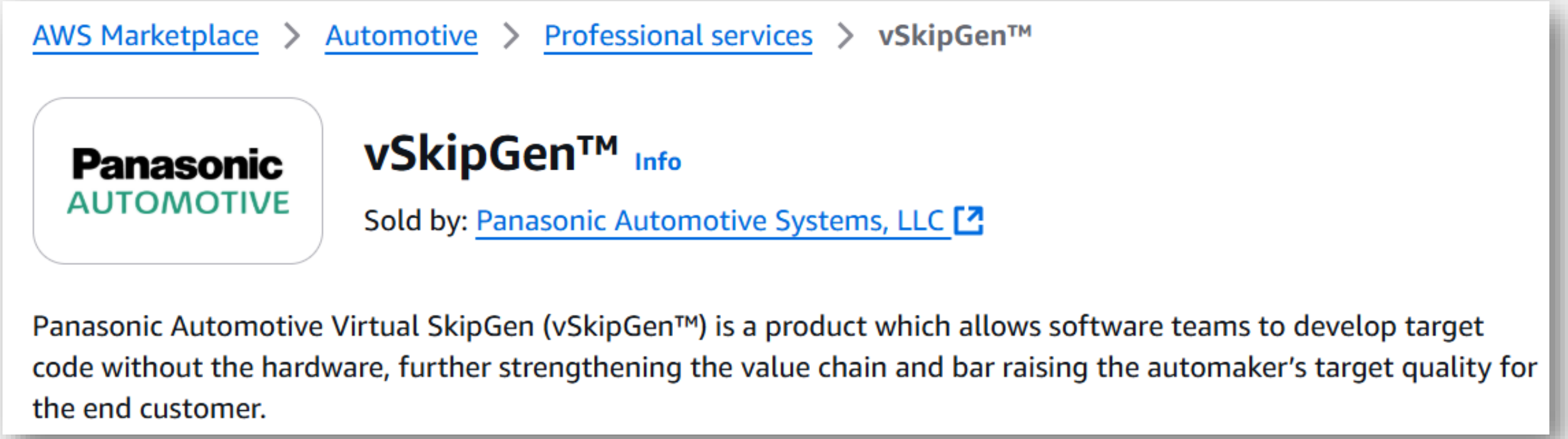


Physical SkipGen



* Android Robot is copied or modified from a work created and provided by Google and used in accordance with the terms set forth in the Creative Commons Attribution 3.0 License.

How to access vSkipGen



https://aws.amazon.com/marketplace/pp/prodview-4fafk4r33mqag

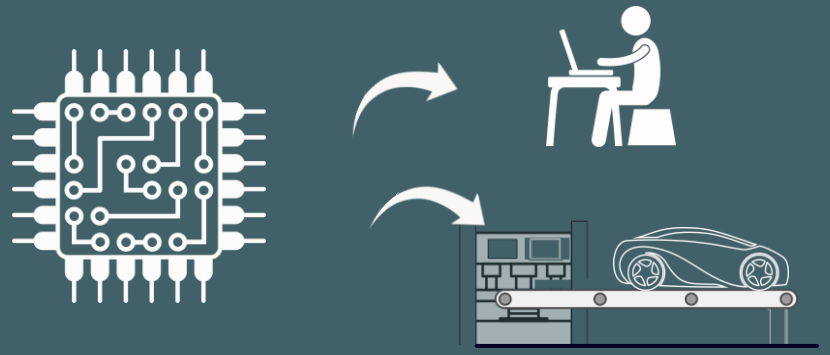
For more details, please, contact
vSkipGenSupport@panasonicautomotive.com

How to use vSkipGen

Software 1st

Identical SW

Hardware 2nd



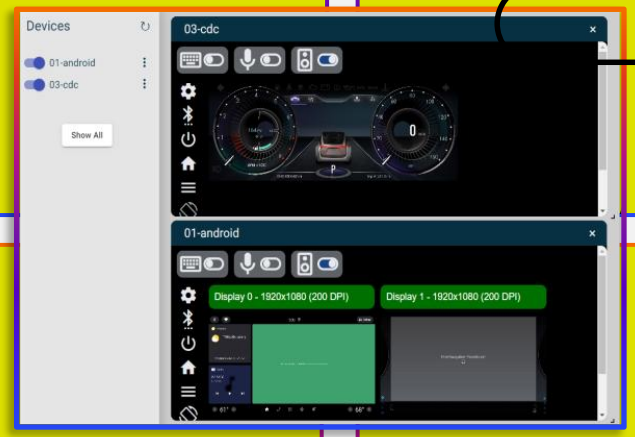
- ✓ Allows for early access to SW development and testing before the HW is available
- ✓ Reduces the physical HW for development
- ✓ Allows for rapid and global scalability
- ✓ Enables CI/CD

- ✓ Validate SW functionality on target HW (e.g., on your desk, remote, in-vehicle)

Value of vSkipGen

✓ **Develop & test full cockpit software with automotive peripheral simulations**

✓ **Enable sustainable scaling on-demand and contribute to a greener world**



✓ **Integrate with CI/CD workflows and simplify global collaboration**

✓ **Optimize the use of vSkipGen for cost versus performance**

SOAFEE

Future Vision with SOAFEE

Atsuya Nasu, Principal Software Engineer at
Panasonic Automotive Systems Co., Ltd.

2025/5/15

Unified HMI & vSkipGen as SOAFEE Blueprints

SOAFEE: “Working together to build open-source architecture for software-defined vehicles. Together we have one single goal - to create a shared platform for vehicles using **cloud-native architecture** that accommodates multiple hardware configurations. By making SOAFEE **hardware agnostic**, we plan to **simplify vehicle software solutions radically**.”

https://www.soafee.io/

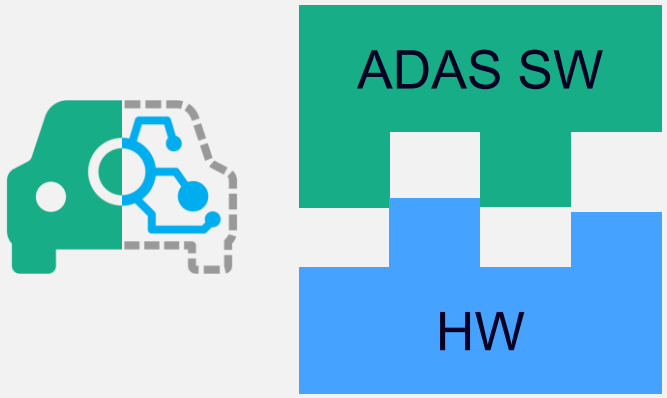
- ✓ **Abstraction**: To abstract and decouple software from hardware -> Hardware agnostic
  - ➢ **Unified HMI:**
    - Enables hardware-agnostic display virtualization.
    - Provides consistent UI/UX across different hardware configurations.
    - Offers developers flexibility in software development and deployment.

- ✓ **Containerization**: To establish simple architecture for microservices and enable migration between cloud and automotive edge

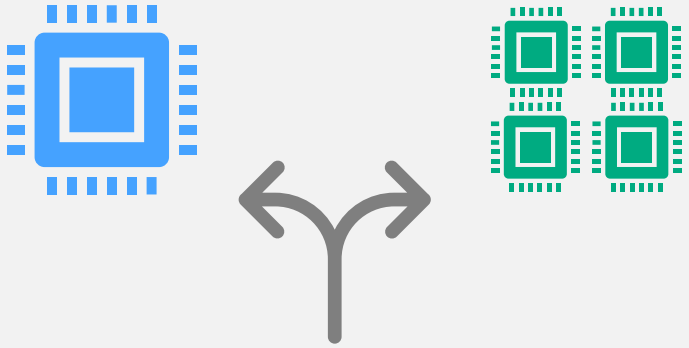
- ✓ **Cloudification**: To create cloud-native architecture to enable automotive software capable to be developed or executed in cloud environment
  - ➢ **vSkipGen:**
    - Provides a cloud-native development environment for automotive software.
    - Enables rapid development and testing in the cloud.
    - Facilitates seamless deployment to actual vehicles.

Further Challenges with Virtualization Technologies

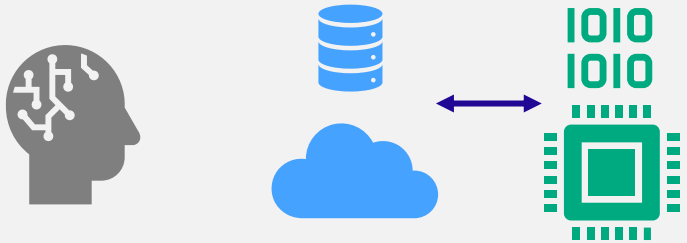
*D*ecoupling of ADAS/HW.



*D*ecoupling from Sys. Arch.



*D*ecoupling of AI/HW



Coupling of Cloud-native/Models

AGL AMM2025 Keynote

https://static.sched.com/hosted_files/aglammspring2025/a0/202502AMM_Keynote_Mizuyama.pdf

SOAFEE

Thank You