Panasonic
AUTOMOTIVE

Enabling a Software-Defined Automotive Edge with VirtIO based Device Virtualization

Sep 21, 2023
Jerry, Jiancong Zhao
Supervisor

Panasonic Automotive Systems Co., Ltd.

Heartmotive

Who am I ?

Panasonic
AUTOMOTIVE

- ✓ Born in Tianjin, China
- ✓ University in Hong Kong
  - Bachelor of Engineer in Industrial Engineering in HKU
- ✓ Work in Japan
  - Panasonic Automotive Systems Co., Ltd. since 2017
- ✓ Focusing on SDV related advanced engineering including virtualization and cloud-native. Representatives of company in multiple OSS initiatives
  - Expert Group Leader of Automotive Grade Linux (AGL) Software-Defined Vehicle Expert Group (SDV-EG)
  - SOAFEE activities (e.g. TSC)
- ✓ Hobby: Travel, Tea, Game

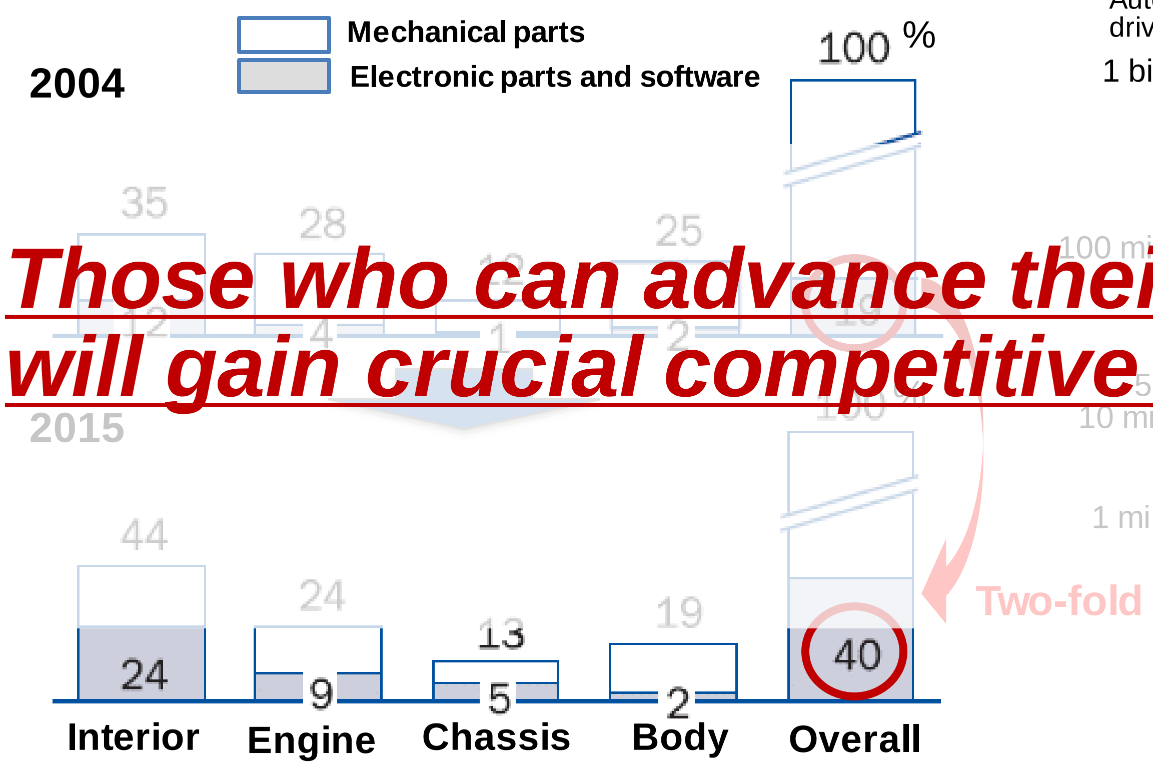
Background: Why SDV is Needed and How Device Virtualization Enables it

Tide of shifting to Software-Defined Vehicles (SDV)

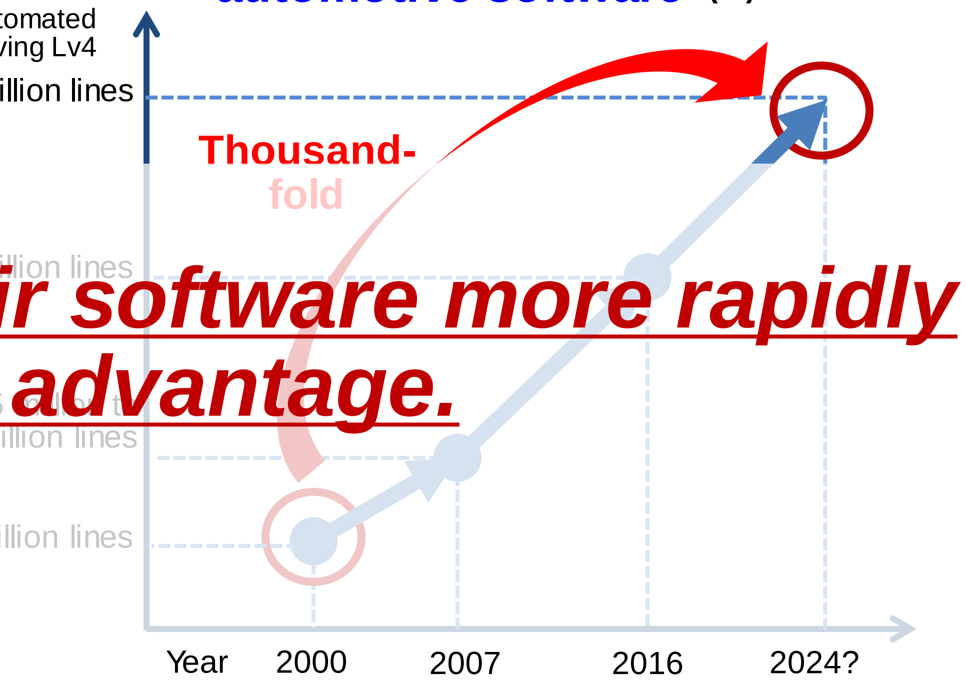
Panasonic AUTOMOTIVE

Most explosive evolutions are happening by DIGITAL and SOFTWARE.

Electronic parts and software ratio has doubled in ten years. (*1)



More complicated automotive software (*2)



<u>Those who can advance their software more rapidly will gain crucial competitive advantage.</u>

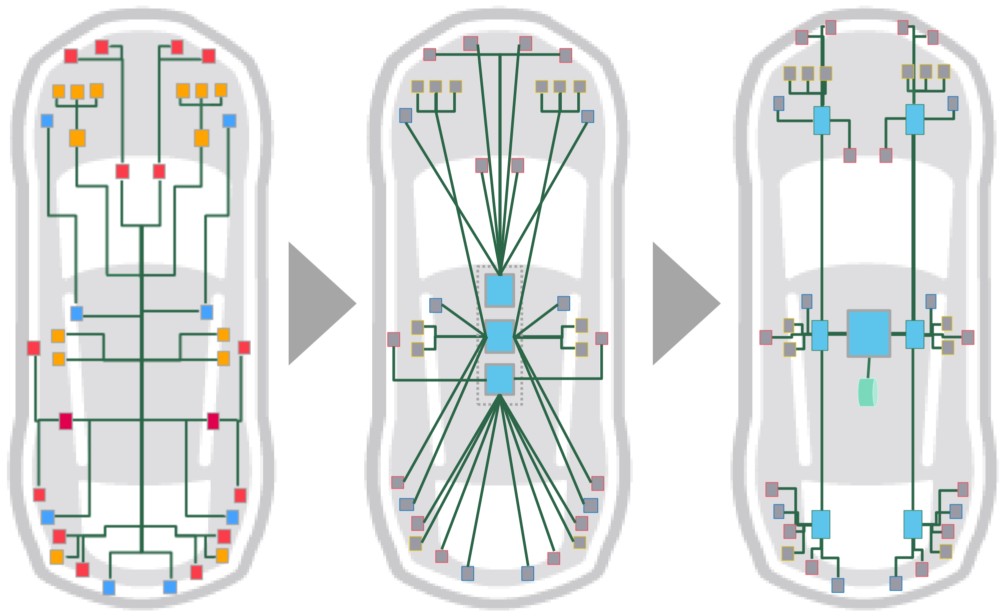
[Reference: The number of source code lines of other products]
- Android OS:12 million lines
- F-35 combat aircraft: 24 million lines
- Microsoft Office 2013: 44 million lines

*1: Source: McKinsey "Managing innovations on the road"

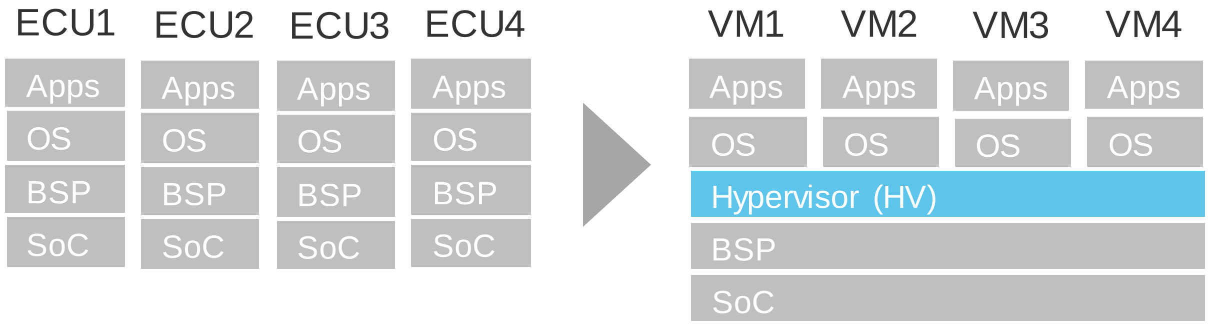
*2: Source: Ministry of Economy, Trade and Industry "Toward acceleration of productivity improvement by IT" Mitsubishi UFJ Morgan Stanley Securities' materials, etc.

Drastic Changes in Automotive Architecture

ECU Consolidation



Virtualization Platform



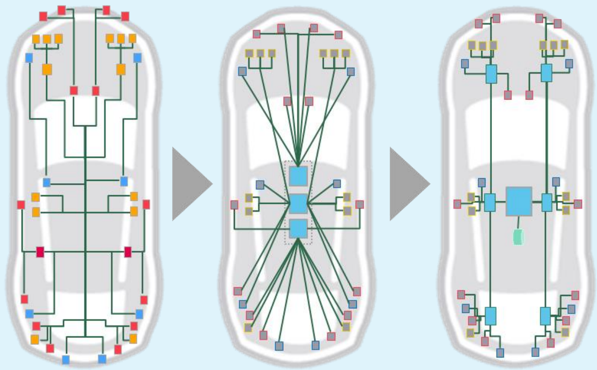
ECU1	ECU2	ECU3	ECU4
Apps	Apps	Apps	Apps
OS	OS	OS	OS
BSP	BSP	BSP	BSP
SoC	SoC	SoC	SoC

VM1	VM2	VM3	VM4
Apps	Apps	Apps	Apps
OS	OS	OS	OS
Hypervisor (HV)			
BSP			
SoC			

Shifting to SDVs - Changing the Mind of Vehicle Values

Panasonic
AUTOMOTIVE

Drastic Changes in Architecture



Open source, de facto standard

AUTOMOTIVE GRADE LINUX

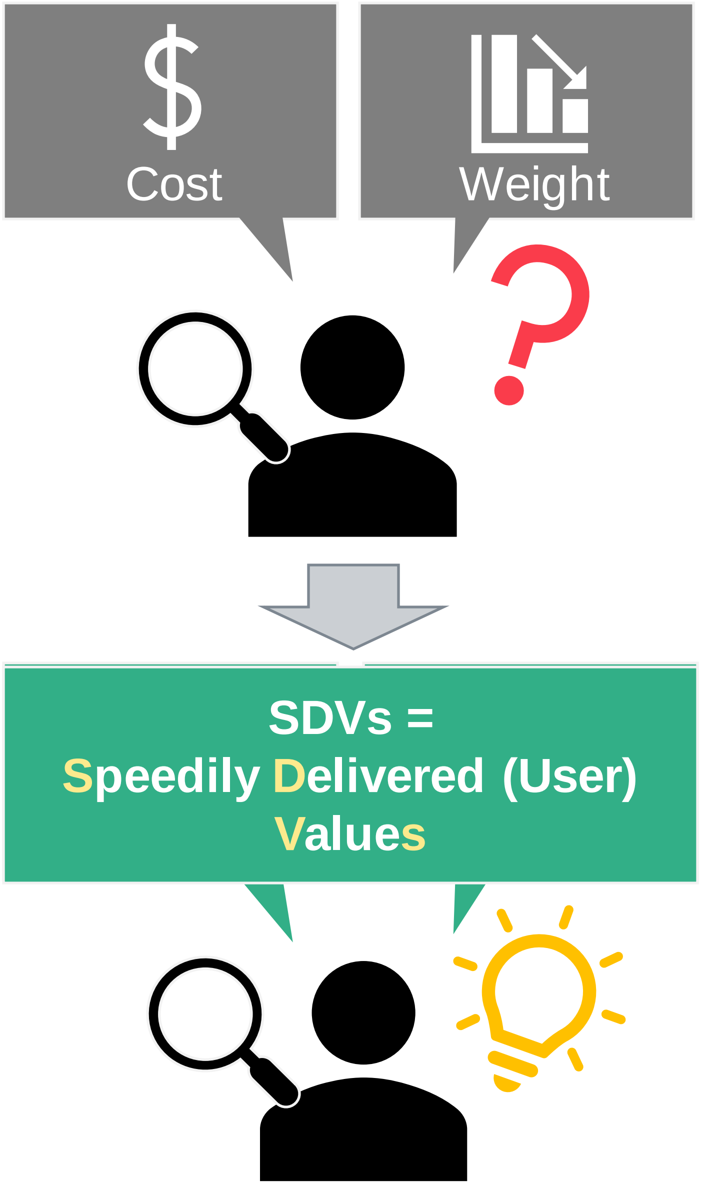
SOAFEE

The Best Stocker - stock.adobe.com

Connected and AI-powered

Big IT players

gguy - stock.adobe.com



SDVs =
Speedily Delivered (User)
Values

Game Change in the Automotive Industry

Those who can evolve software more rapidly will secure competitive advantages.

Key Strategies

- Maximizing LOC per man-month
- Possessing larger software team

✗

Key Strategies

- Sophistication of architecture
- Complementing with ecosystems
- Rapid product discovery

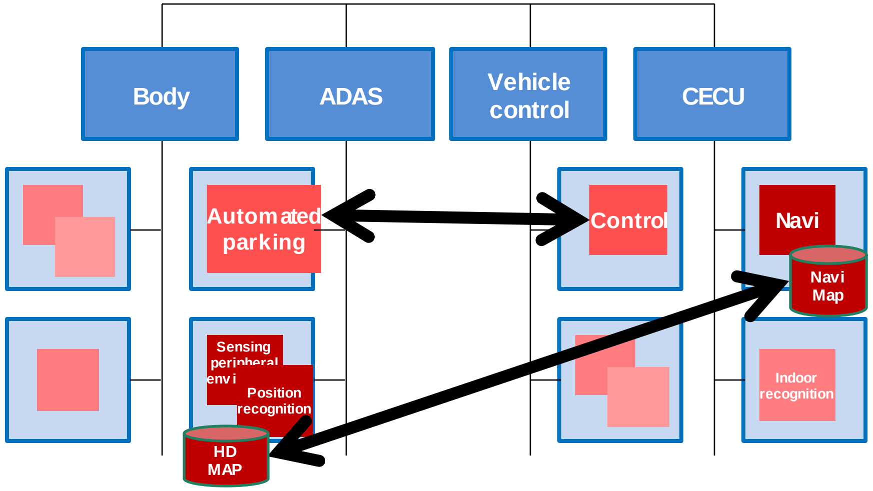
✓

Desirable Direction of Automotive System Architecture

Panasonic
AUTOMOTIVE

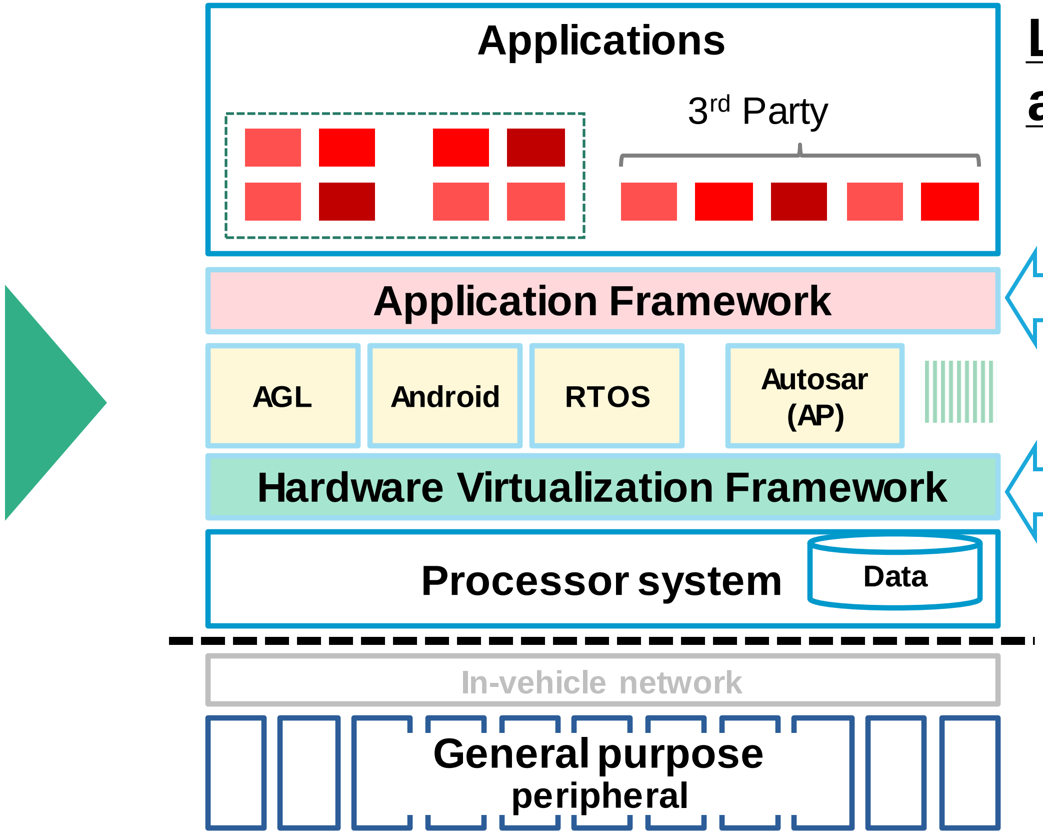
ECU consolidation is not a purpose but means --- The true purpose is to establish the optimal architecture for evolution of software.

"Those who can advance their software more rapidly will gain crucial competitive advantage."



Advancement of technology and updates are difficult.
Overlap of computing resources is an issue also.

- 🟦 Domain Controllers
- ▢ ECUs
- 🟥 Applications and Data



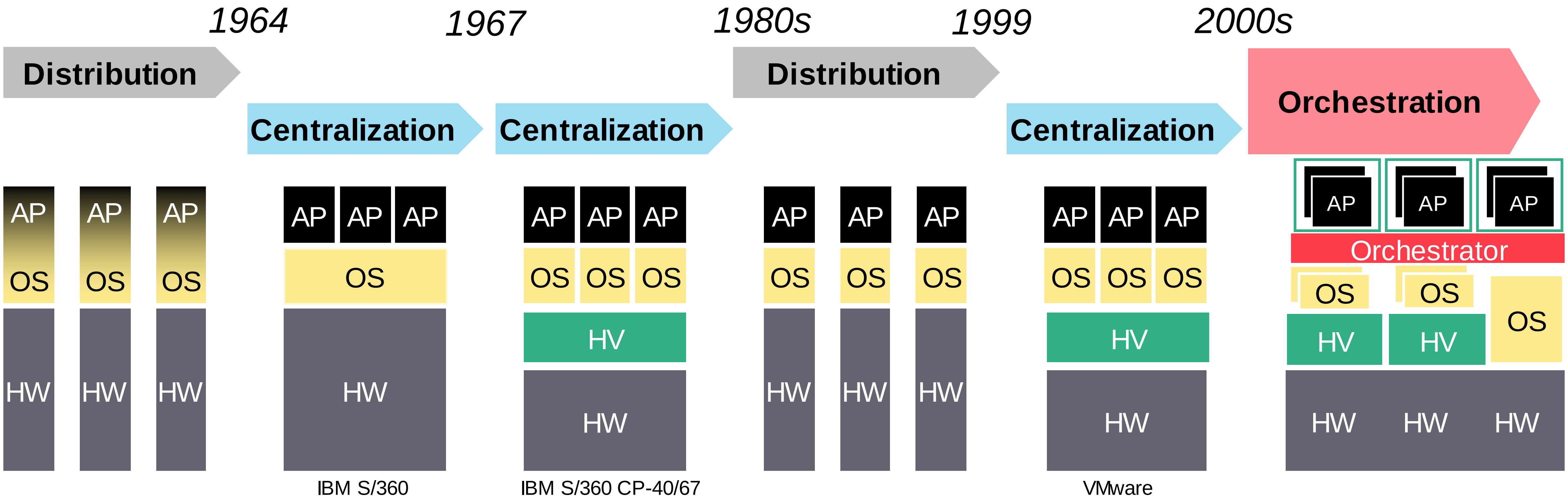
Logical architecture

Makes Apps code simple by hiding the detail of the underlying platform

Decouples software from underlying hardware

Historical Trend of General Computing Architecture (Distribution and Centralization)

The history of general computing architecture is **repeating the cycle between centralization and distribution**, and the automotive industry is following a similar path.

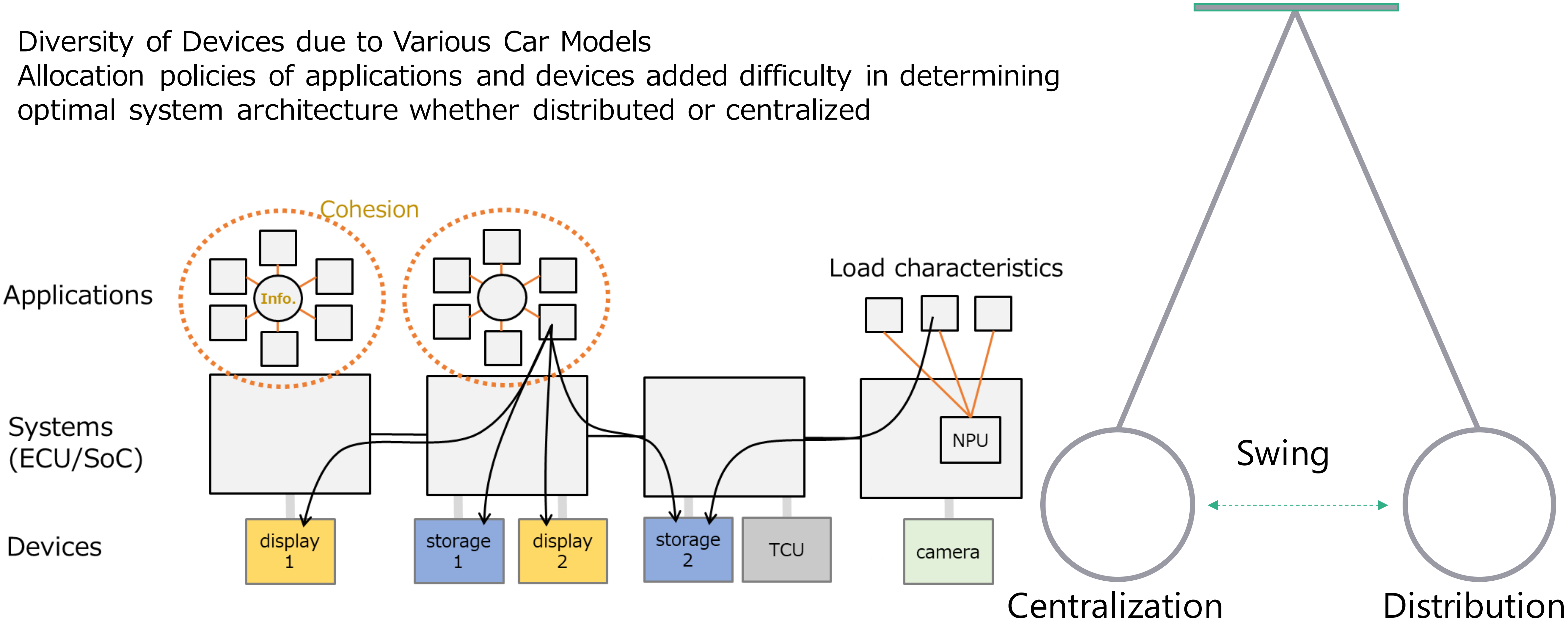


Created by Panasonic Automotive Systems referring to ITmedia IT solution cram school [Graphic explanation] History of virtualization on a single sheet
https://blogs.itmedia.co.jp/itsolutionjuku/2015/06/post_90.html

Greater Complexity in Automotive to Determine Optimal Architecture

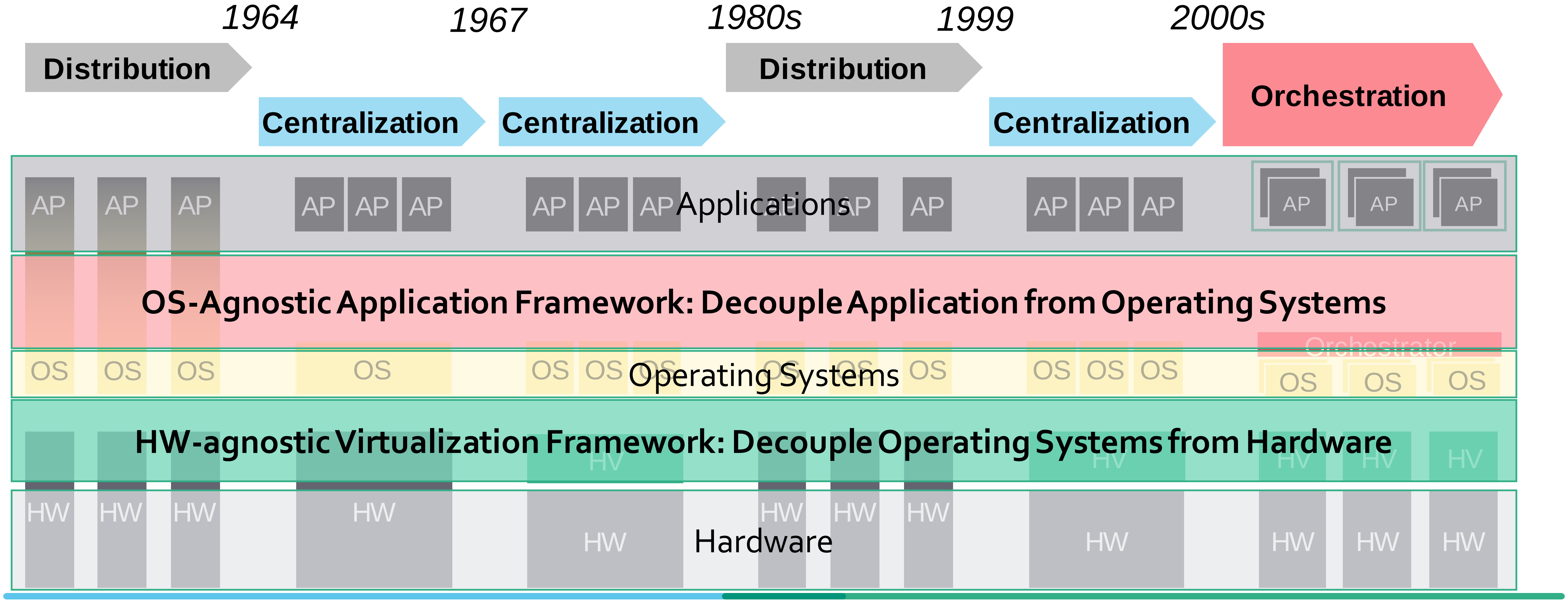
Complicated natures of both devices and applications make a greater complexity for automotive

- Diversity of Devices due to Various Car Models
- Allocation policies of applications and devices added difficulty in determining optimal system architecture whether distributed or centralized



Historical Trend of General Computing Architecture (Distribution and Centralization)

No matter how the underlaying computing architecture has changed, a consistent objective is to decouple apps (directly contributed to user values) from underlying computing architecture
→ An Operating-System-Agnostic Application Framework and a Hardware-Agnostic Abstraction Framework are continuously to be the key to drive industry shift from hardware-centric to software-defined



Applications

OS-Agnostic Application Framework: Decouple Application from Operating Systems

Operating Systems

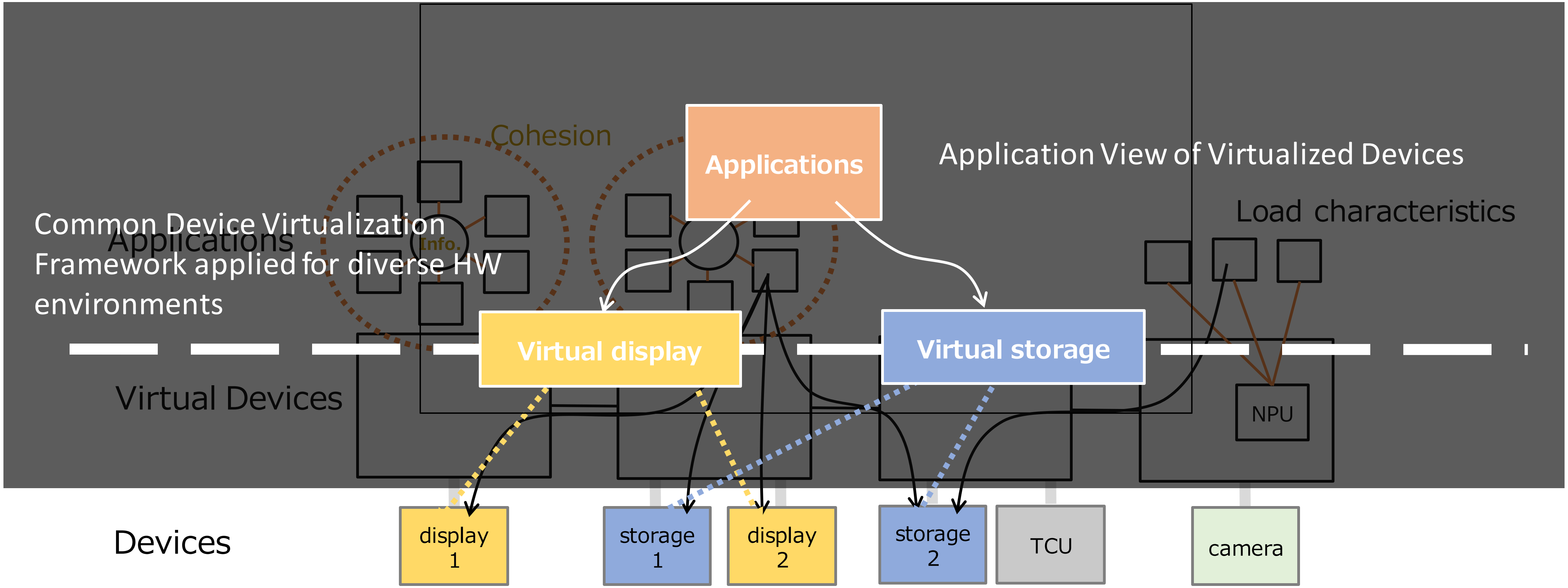
HW-agnostic Virtualization Framework: Decouple Operating Systems from Hardware

Hardware

Today Focus: Device Virtualization – Key to SDV

Panasonic
AUTOMOTIVE

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)

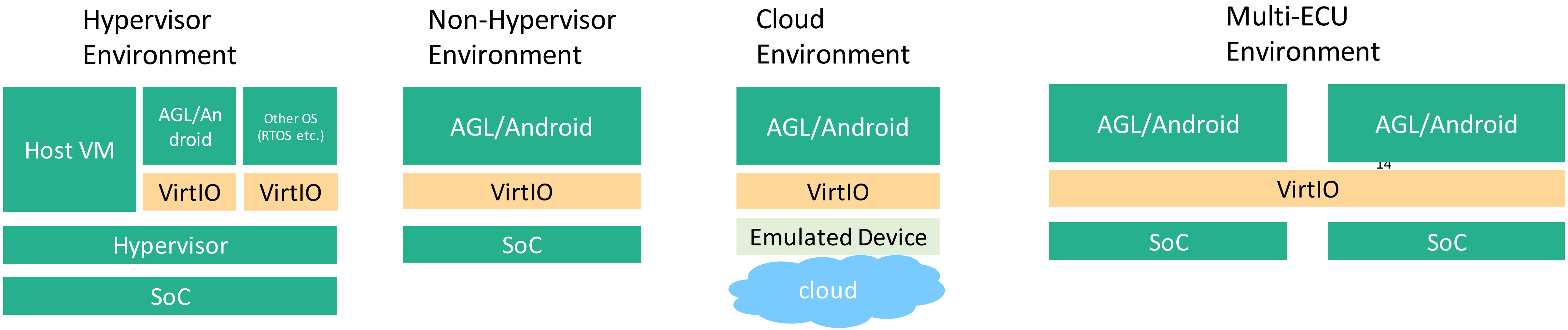


Standard Device Virtualization Framework - VirtIO

Overview of Device Virtualization - Concept

Panasonic
AUTOMOTIVE

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



Hypervisor Environment

Host VM

AGL/Android

Other OS (RTOS etc.)

VirtIO

VirtIO

Hypervisor

SoC

Non-Hypervisor Environment

AGL/Android

VirtIO

SoC

Cloud Environment

AGL/Android

VirtIO

Emulated Device

cloud

Multi-ECU Environment

AGL/Android

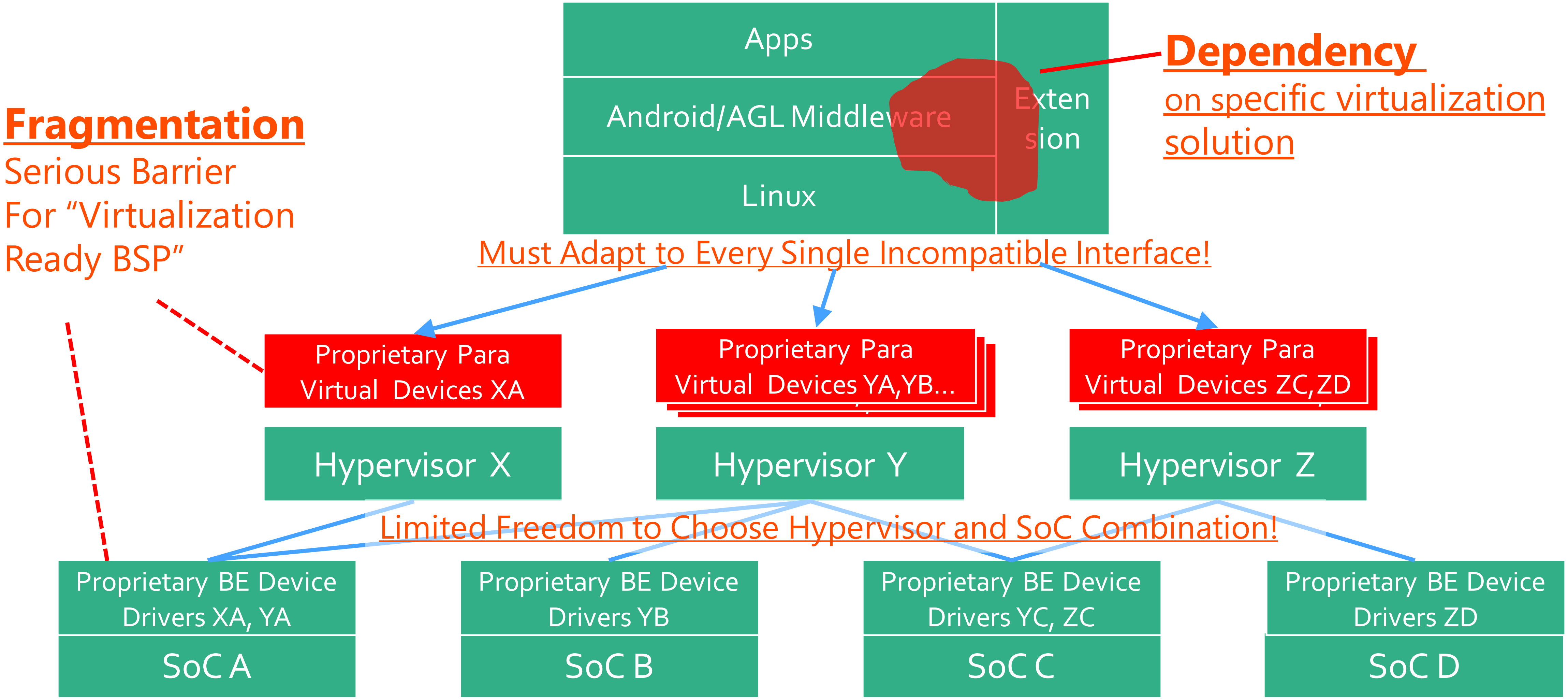
AGL/Android

VirtIO

SoC

SoC

Pains around Virtualization in the Past



Fragmentation
Serious Barrier
For “Virtualization
Ready BSP”

Dependency
on specific virtualization solution

Must Adapt to Every Single Incompatible Interface!

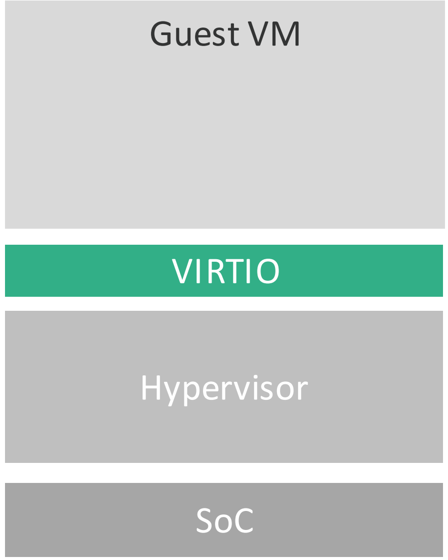
Limited Freedom to Choose Hypervisor and SoC Combination!

Apps
Android/AGL Middleware
Linux
Exten sion
Proprietary Para Virtual Devices XA
Proprietary Para Virtual Devices YA,YB…
Proprietary Para Virtual Devices ZC,ZD
Hypervisor X
Hypervisor Y
Hypervisor Z
Proprietary BE Device Drivers XA, YA
SoC A
Proprietary BE Device Drivers YB
SoC B
Proprietary BE Device Drivers YC, ZC
SoC C
Proprietary BE Device Drivers ZD
SoC D

Excerpt from Panasonic’s Keynote Presentation at the AGL AMM July 2020

Enter Standard Virtualization Framework - VirtIO

- Developed in 2008 as a hypervisor neutral way of accessing devices
- Provide virtual machines access to Input/Output
- A standardized interface for I/O between virtual machines and hypervisors
- Abstract device functionality instead of hardware
- Drivers are widely available in all major operating systems (Linux, Android, BSD, Windows, etc)
- Supported by all clouds and enterprise hypervisors



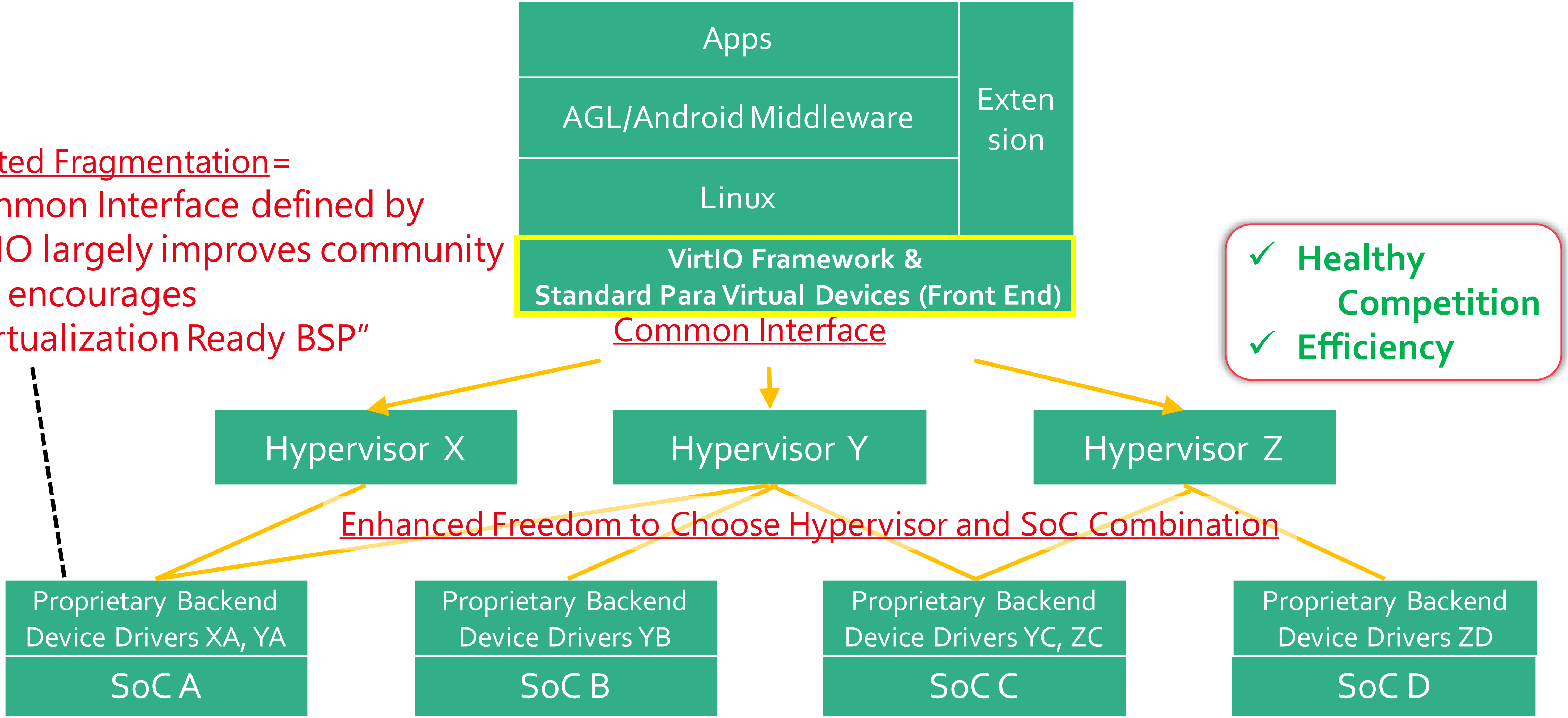
- Reliable and proven technology
- Versatile abstraction model
- Scalable and high performance
- Multiple interoperable implementations
- Broad ecosystem across multiple industries

Excerpt from AGL F2F OpenSynergy's Presentation on Oct 2022

VirtIO as a Common Device Virtualization Framework for Automotive

Panasonic
AUTOMOTIVE

Limited Fragmentation = Common Interface defined by VirtIO largely improves community and encourages "Virtualization Ready BSP"



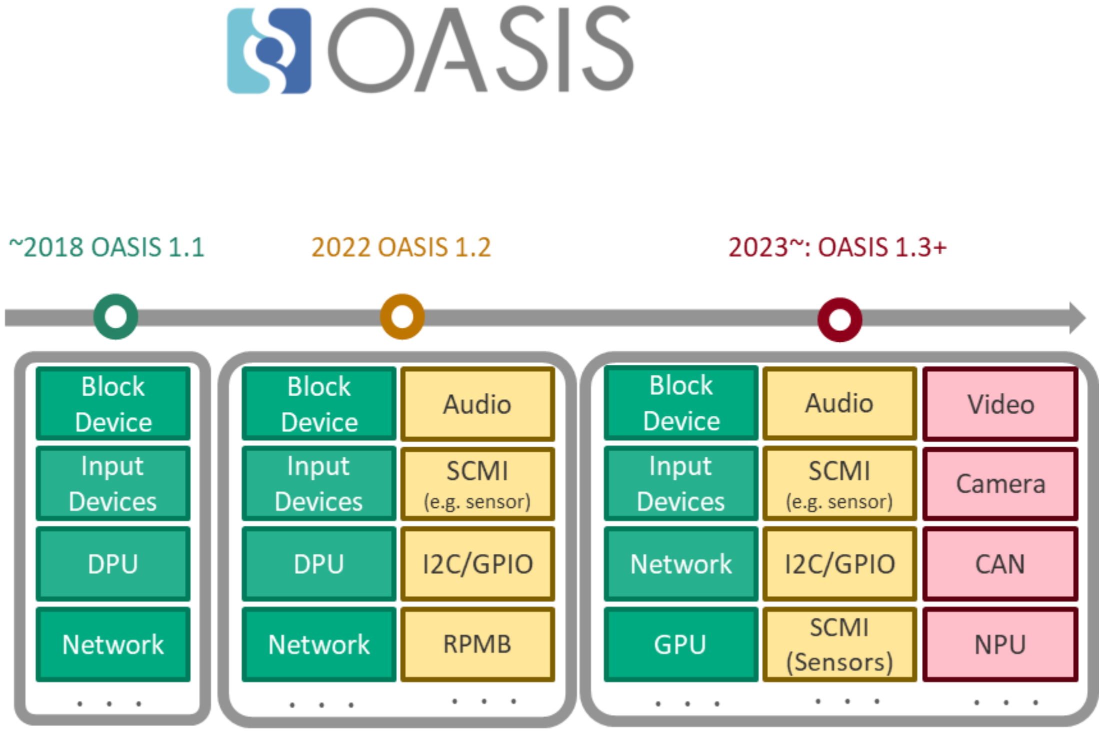
Apps
AGL/Android Middleware
Linux
Extension
VirtIO Framework & Standard Para Virtual Devices (Front End)
Common Interface
Hypervisor X
Hypervisor Y
Hypervisor Z
Enhanced Freedom to Choose Hypervisor and SoC Combination
Proprietary Backend Device Drivers XA, YA
SoC A
Proprietary Backend Device Drivers YB
SoC B
Proprietary Backend Device Drivers YC, ZC
SoC C
Proprietary Backend Device Drivers ZD
SoC D

✓ Healthy Competition
✓ Efficiency

Excerpt from Panasonic's Keynote Presentation at the AGL AMM July 2020

VirtIO as a Common Framework for Virtualized Automotive OS

In parallel to steady progress of specification standardization, rapid implementation evolvement has been occurred in various OSS communities to fulfill most of fundamental automotive, indicating VirtIO as the de facto standard of automotive industry.



OASIS

~2018 OASIS 1.1

Block Device
Input Devices
DPU
Network
. . .

2022 OASIS 1.2

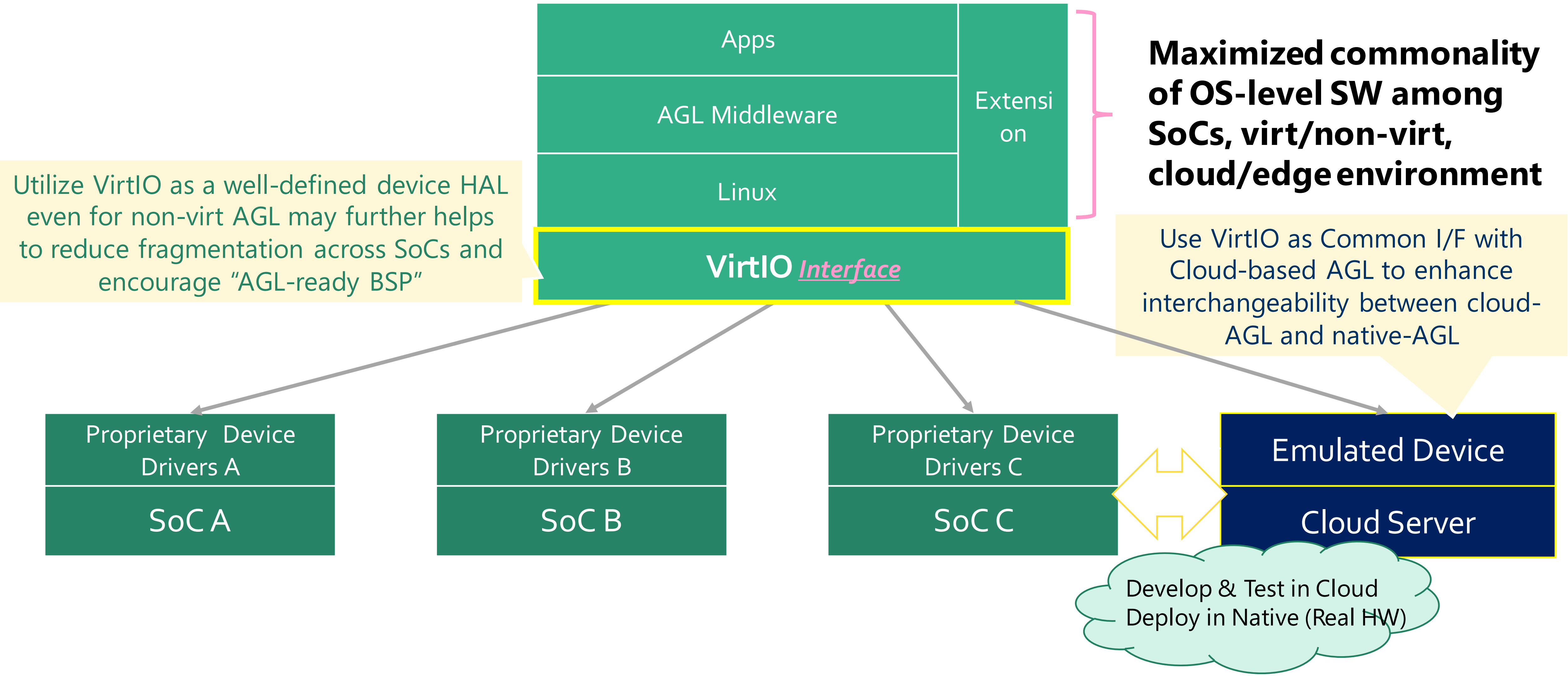
Block Device | Audio
Input Devices | SCMI (e.g. sensor)
DPU | I2C/GPIO
Network | RPMB
. . . | . . .

2023~: OASIS 1.3+

Block Device | Audio | Video
Input Devices | SCMI (e.g. sensor) | Camera
Network | I2C/GPIO | CAN
GPU | SCMI (Sensors) | NPU
. . . | . . . | . . .

SOAFEE
VirtIO
AUTOMOTIVE GRADE LINUX

VirtIO Beyond Traditional Hypervisor Virtualization



Apps

AGL Middleware

Linux

Extensi
on

VirtIO *Interface*

Maximized commonality of OS-level SW among SoCs, virt/non-virt, cloud/edge environment

Utilize VirtIO as a well-defined device HAL even for non-virt AGL may further helps to reduce fragmentation across SoCs and encourage “AGL-ready BSP”

Use VirtIO as Common I/F with Cloud-based AGL to enhance interchangeability between cloud-AGL and native-AGL

Proprietary Device Drivers A

SoC A

Proprietary Device Drivers B

SoC B

Proprietary Device Drivers C

SoC C

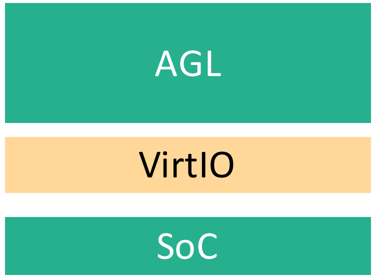
Emulated Device

Cloud Server

Develop & Test in Cloud
Deploy in Native (Real HW)

VirtIO Work for Non-Hypervisor Environment

Non-Hypervisor Environment



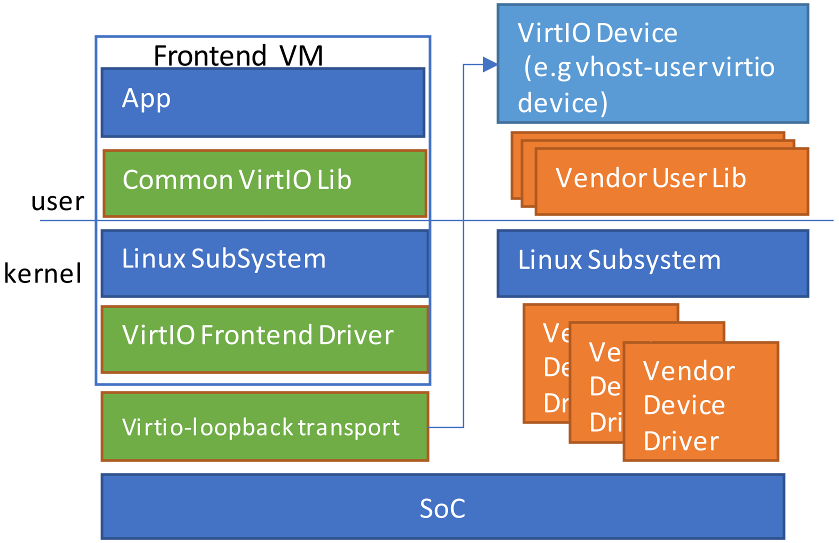
Priority of Device Virtualization
Voted by AGL Members (2021)

Device	Total Score	Priority
Input Device	29	1
Display	27	2
GPU	26	3
CAN bus	20	4
Block Device	19	5
Audio	18	6
Ethernet	11	7
Bluetooth	9	8
SPI	8	9
Serial console	8	9
SCMI	8	9

Working together with AGL community members, Non-hypervisor VirtIO based virtualization has been in a steady progress.

- Finished Design & Implementation of a common virtio-based HAL layer "virtio-loopback" portable to execute on both native and virtual environments with basic devices (blk, rng, input) support
- Continue next-step work to support more devices this year to enable a complete AGL UCB running on the top of virtio-loopback devices
- Plan to extend the use case from single-ECU to multi-ECU and cloud-native

High Level Architecture Design



Status and Future Plan

2022
- virtio-blk
- virtio-rng
- virtio-input

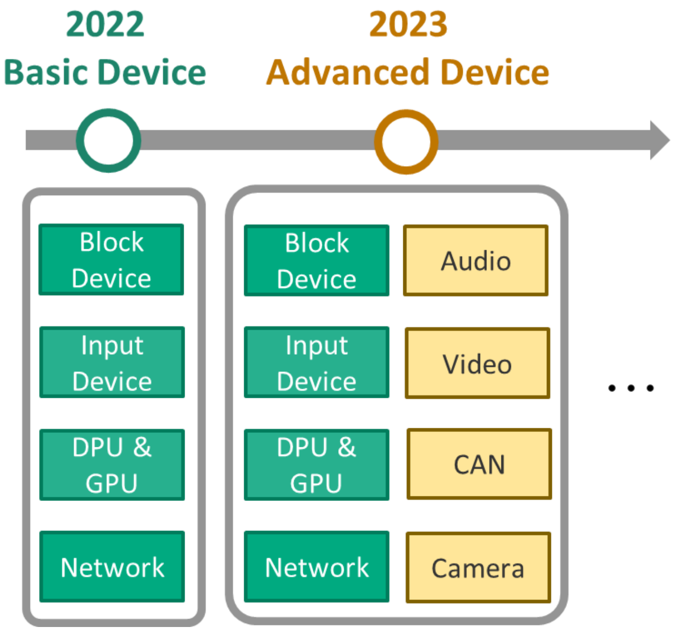
2023
- virtio-gpu
- virtio-snd
- virtio-can
- virtio-gpio, i2c

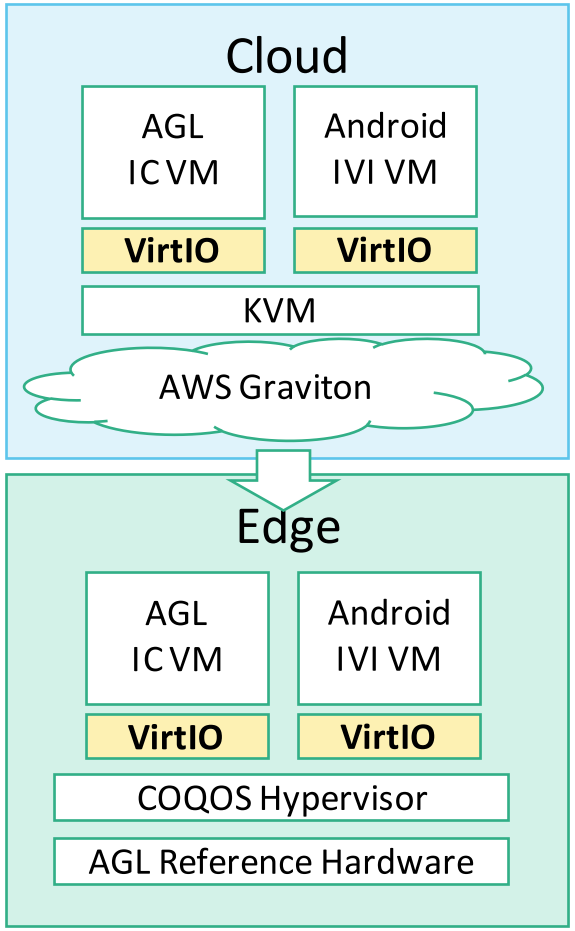
Future
- Support on multi-ECU architecture
- Support on cloud-native

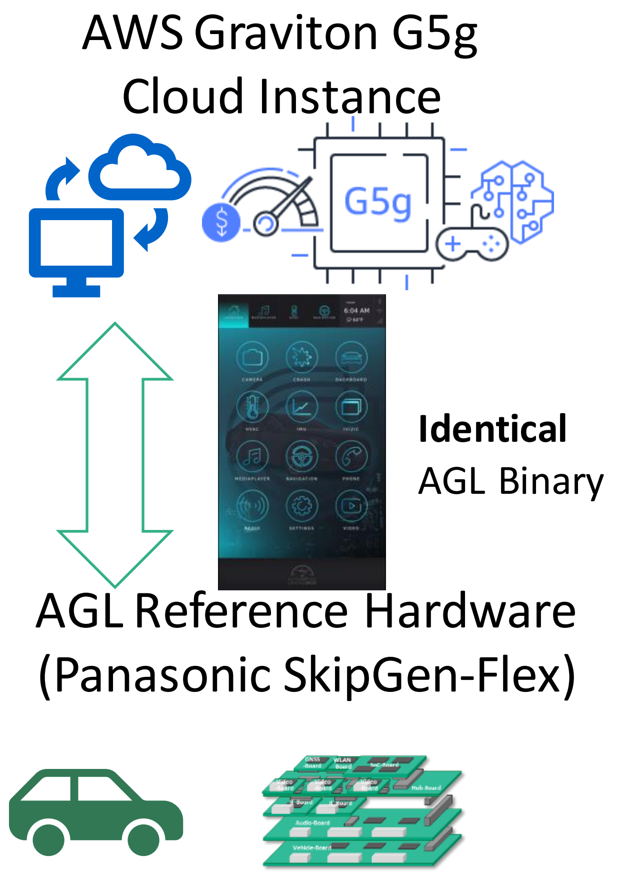
* Check more details about Non-hypervisor VirtIO at https://www.youtube.com/watch?v=Lfj3dYCAiik&list=PL6EdENMl-83iGkh4kpeWFclW5ULIJDqxs&index=23

VirtIO Work for Cloud-Native Environment

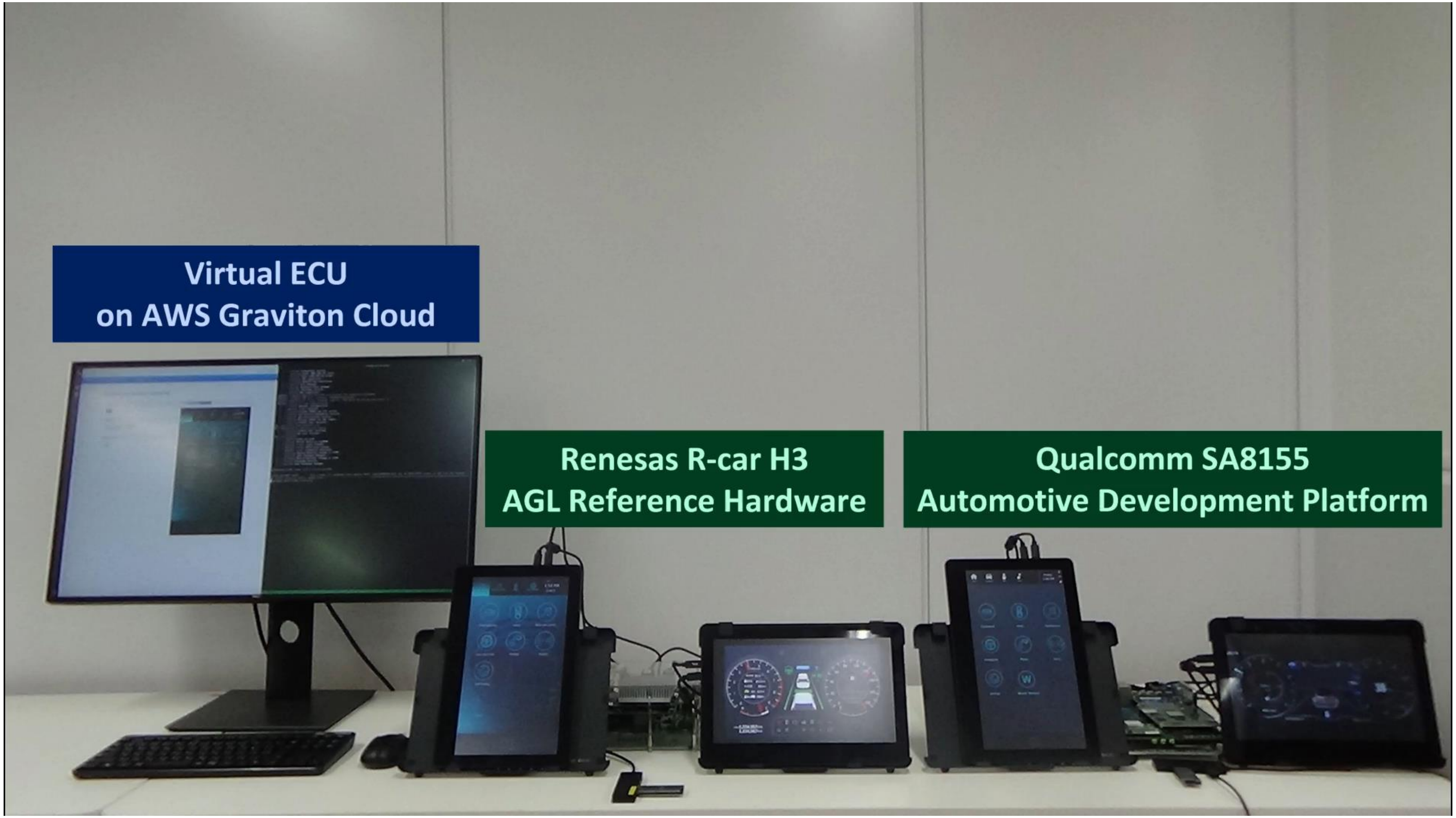
- Identical IVI binary can be run on both cloud and edge -> OS-level binary parity
- Same OS binary can be deployed to different automotive hardware -> SoC Agnostic







VirtIO Work for Cloud-Native Environment



Panasonic
AUTOMOTIVE

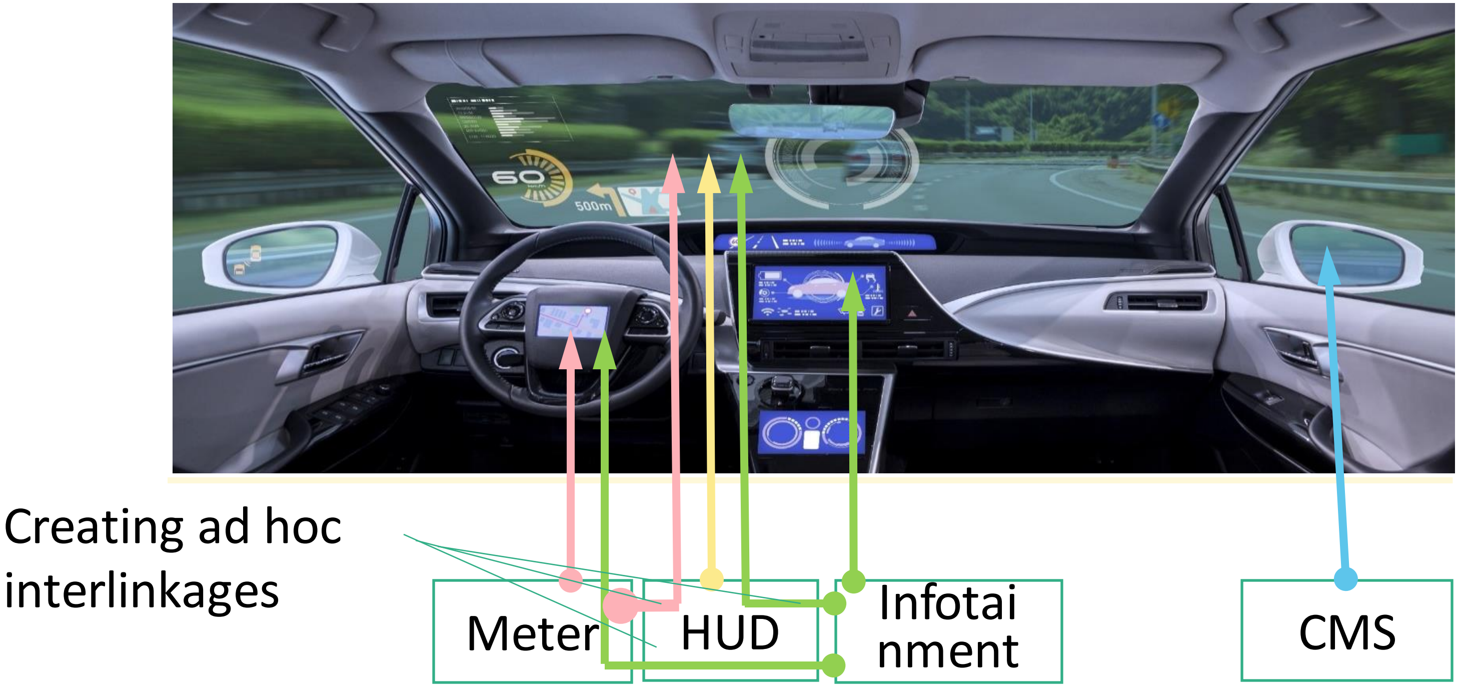
Advanced Use Case of VirtIO: Display Virtualization Enabling Software-Defined Architecture

Display Trends in the Automotive Industry

Panasonic
AUTOMOTIVE

- The increasing number of in-vehicle displays has created a demand 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 that separates software from hardware.



Creating ad hoc interlinkages

Meter

HUD

Infotai
nment

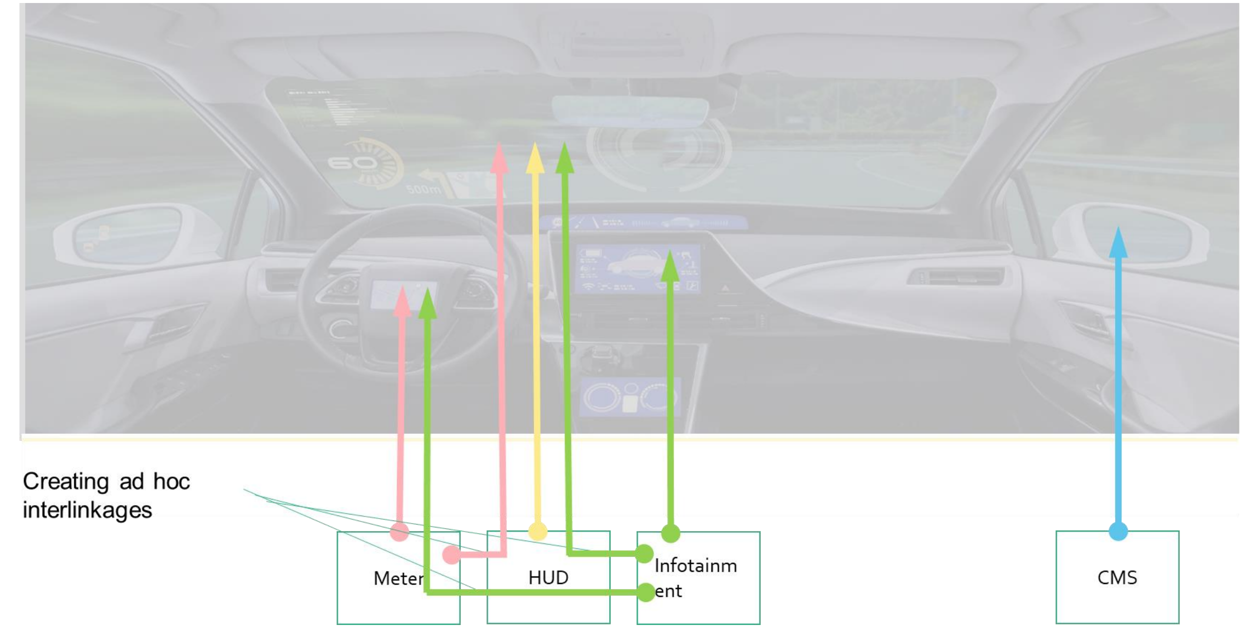
CMS

Software-Defined Display Virtualization Technology - Unified HMI

Panasonic
AUTOMOTIVE

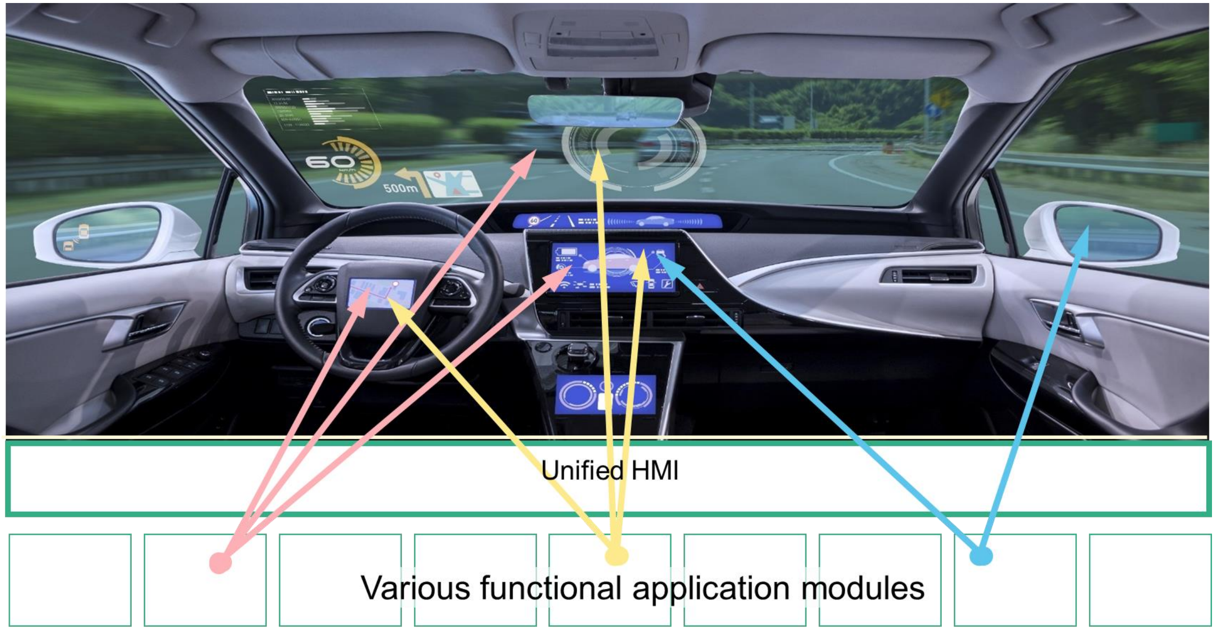
Legacy HMI System

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



Unified HMI System

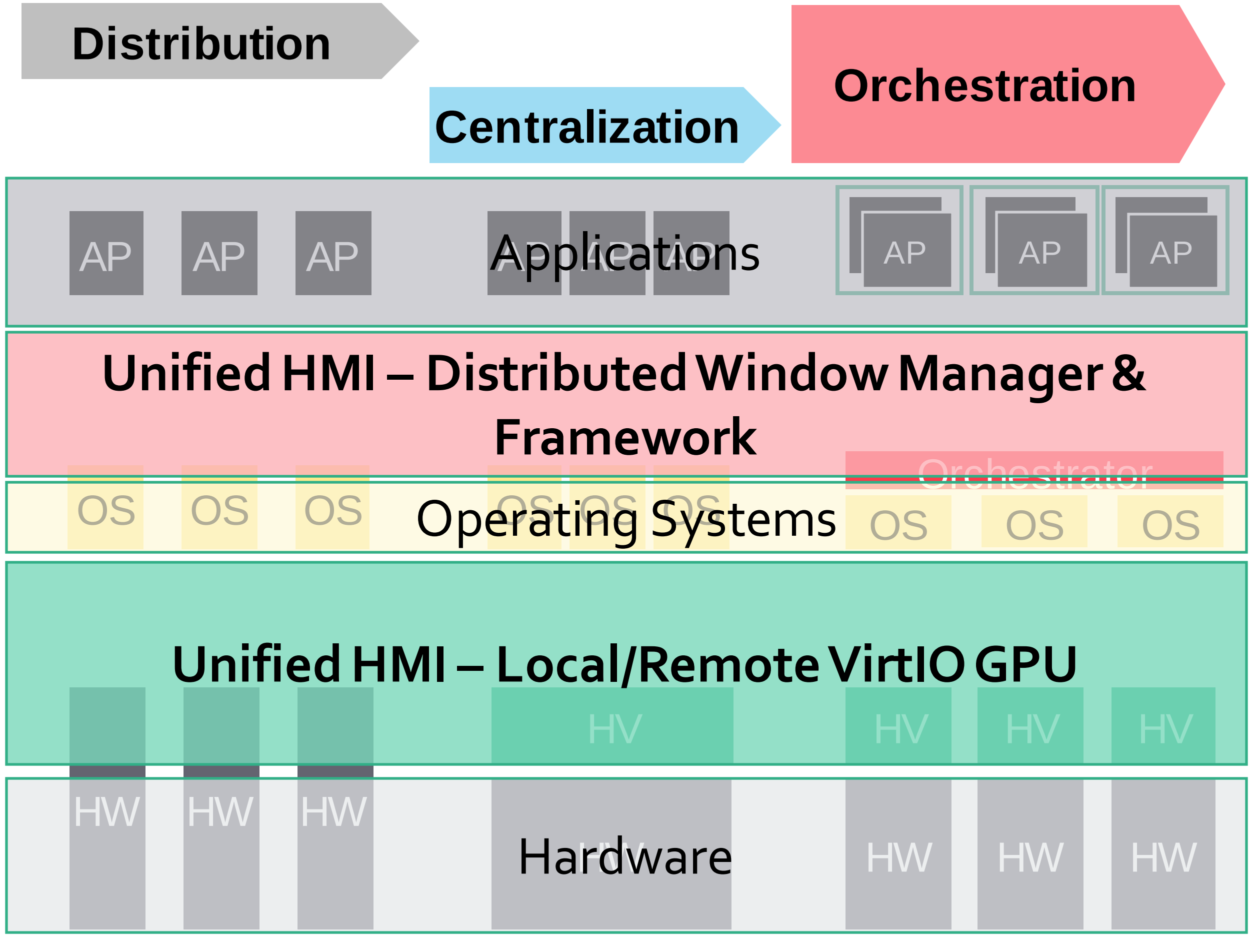
Full Flexibility on ECU & Function-Display Relationship for Cockpit UX Innovation



Software-Defined Display Virtualization Technology - Unified HMI

Panasonic
AUTOMOTIVE

Unified HMI, as a virtual display technology, has been a good example of decoupling application implementation from underlying computing architecture, which enables a flexible, dynamic and fast-evolved Cockpit/Cabin UI/UX



Distribution

Centralization

Orchestration

Applications

AP AP AP AP AP AP

Unified HMI – Distributed Window Manager & Framework

Orchestrator

Operating Systems

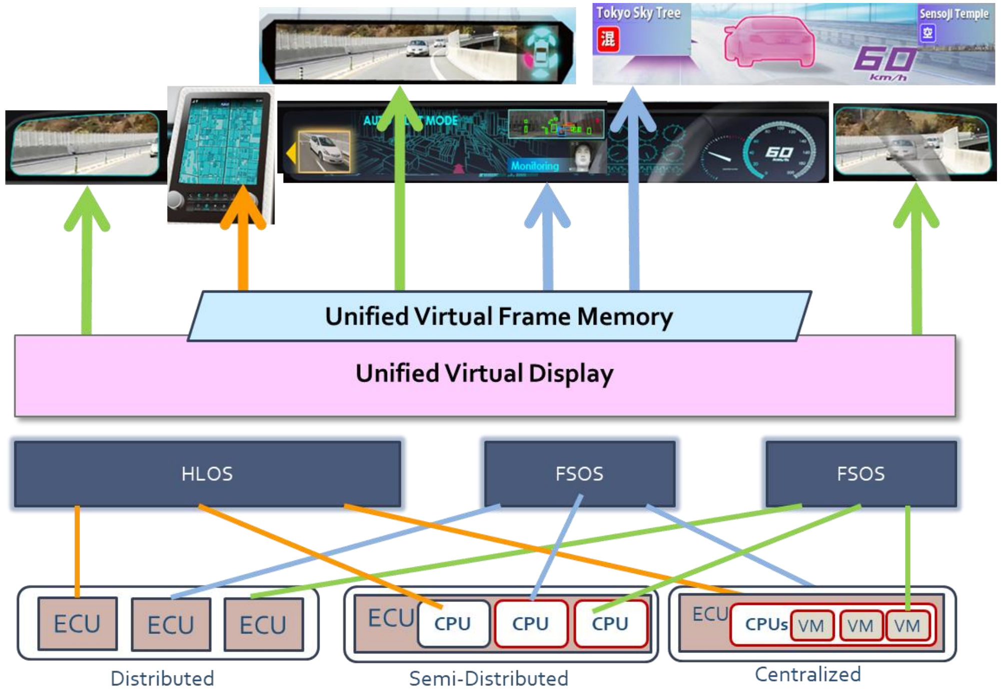
OS OS OS OS OS OS

Unified HMI – Local/Remote VirtIO GPU

HV HV HV HV

Hardware

HW HW HW HW HW HW



Unified Virtual Frame Memory

Unified Virtual Display

HLOS

FSOS

FSOS

ECU ECU ECU

Distributed

ECU CPU CPU CPU

Semi-Distributed

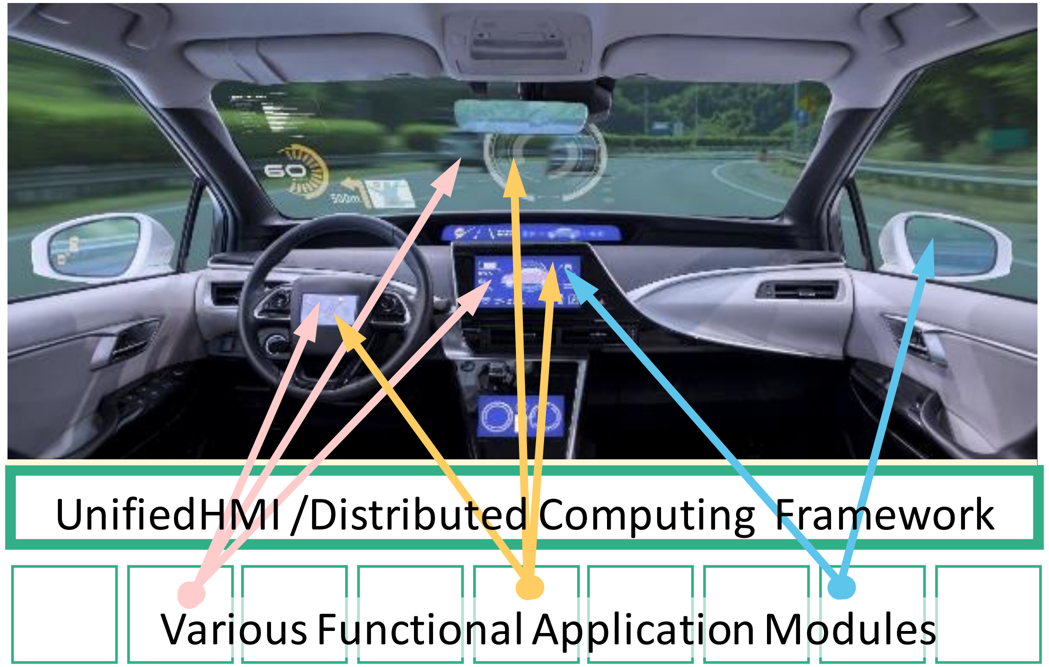
ECU CPUs VM VM VM

Centralized

Unified control of all physical displays by **mappig multiple physical displays** into a **single large virtual display**

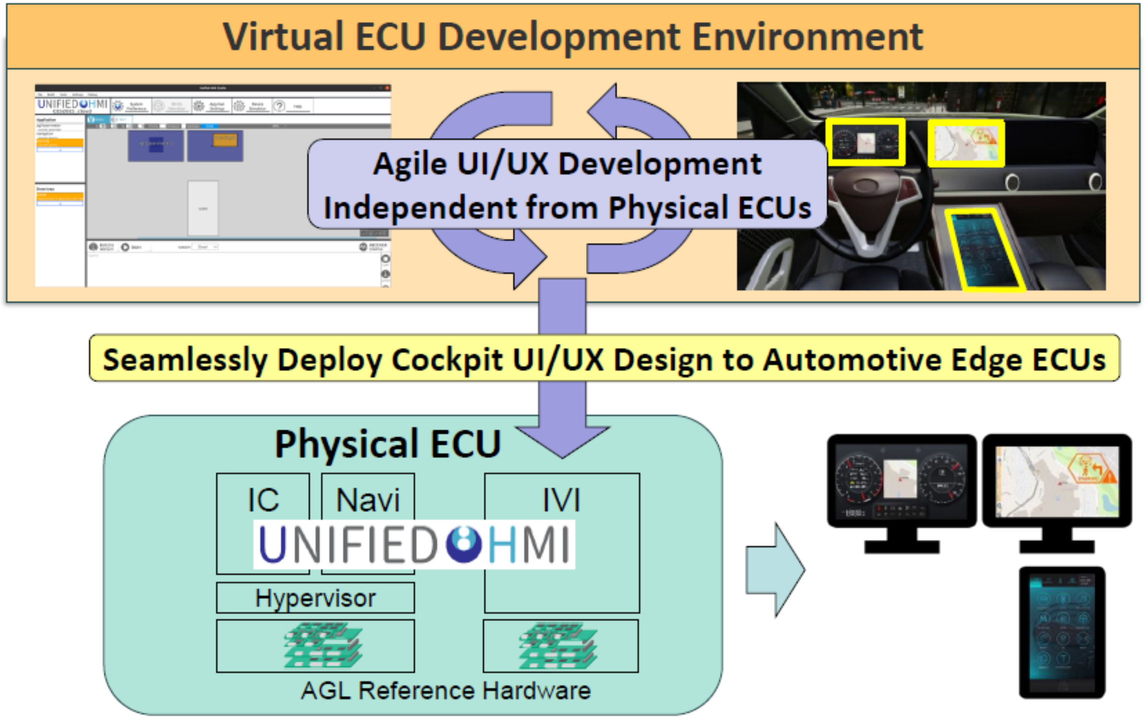
Concept of Unified HMI

- Unified HMI is a “Software-Defined” display virtualization platform that allows for flexible development of the entire cockpit UI/UX across multiple displays **independent of hardware and OS configuration**.
- The entire cockpit UI/UX is developed using virtual ECU in the cloud and it can be seamlessly deployed to the physical ECU, enabling rapid development and deployment through OTA updates.



UnifiedHMI /Distributed Computing Framework

Various Functional Application Modules



Virtual ECU Development Environment

Agile UI/UX Development Independent from Physical ECUs

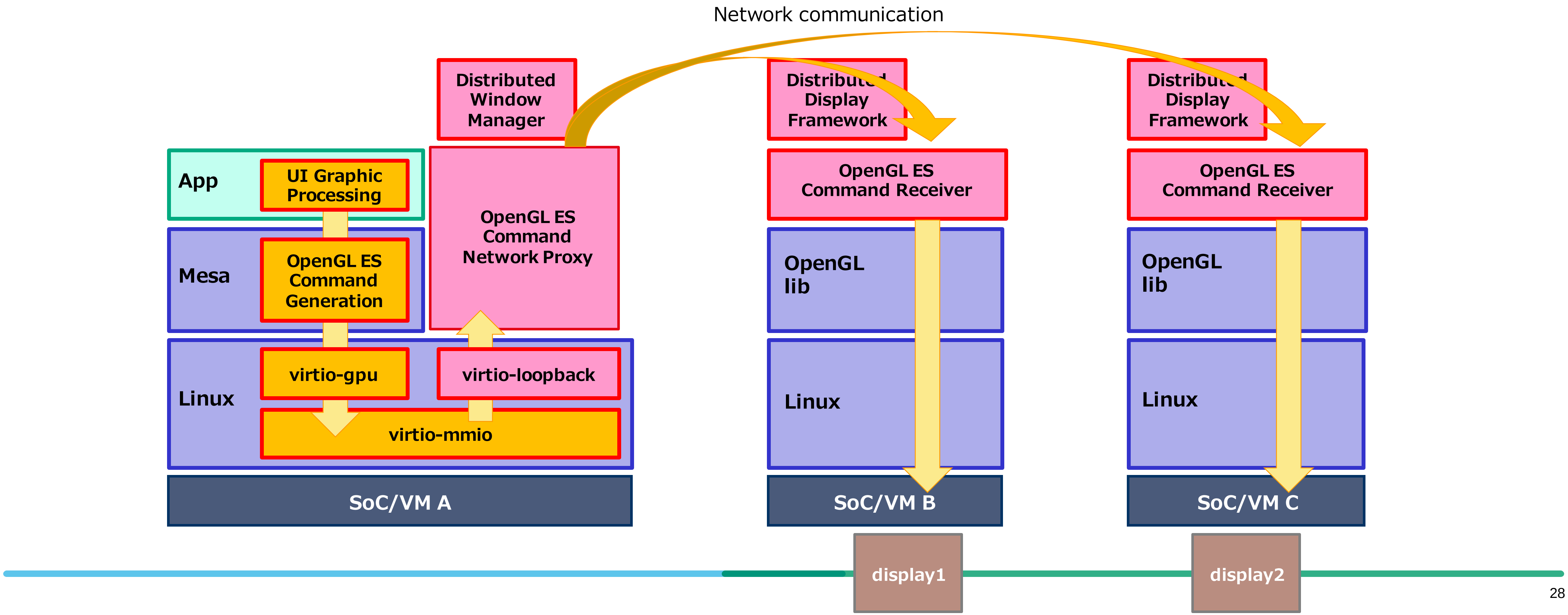
Seamlessly Deploy Cockpit UI/UX Design to Automotive Edge ECUs

Physical ECU

Unified HMI Schematic Architecture

Panasonic AUTOMOTIVE

① Apps are rendered with virtual GPU (VirtIO-GPU)
② Graphics are drawn by remote system through proxy requests
③ Layouts are managed by the distributed window manager



Network communication
Distributed Window Manager
App
UI Graphic Processing
Mesa
OpenGL ES Command Generation
OpenGL ES Command Network Proxy
Linux
virtio-gpu
virtio-loopback
virtio-mmio
SoC/VM A
Distributed Display Framework
OpenGL ES Command Receiver
OpenGL lib
Linux
SoC/VM B
Distributed Display Framework
OpenGL ES Command Receiver
OpenGL lib
Linux
SoC/VM C
display1
display2

Unified HMI Demo

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

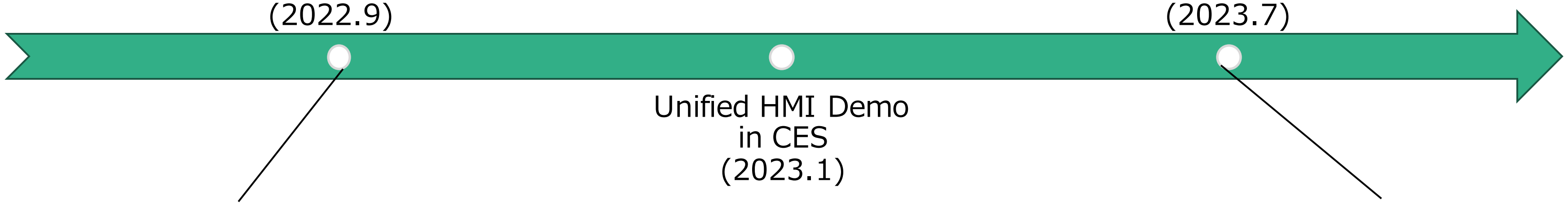
Fast-evolving & Personalized Cockpit UI/UX Experience

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

Unified HMI OSS Activities

Panasonic
AUTOMOTIVE

Unified HMI OSS Roadmap

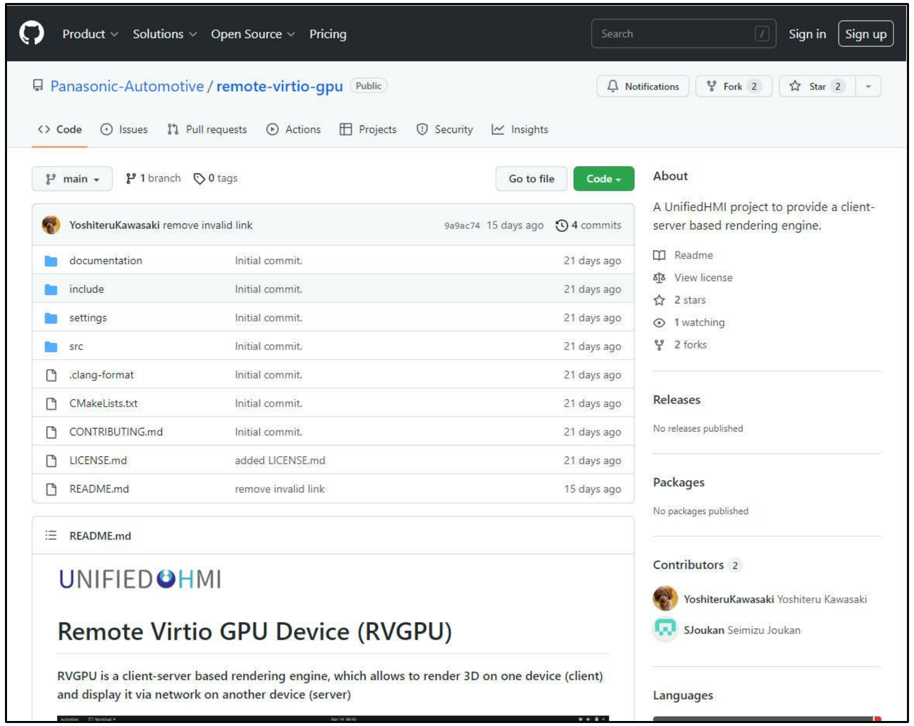


Publish Unified HMI on GitHub* (2022.9)

Unified HMI Demo in CES (2023.1)

Apply Unified HMI to Latest AGL UCB Version PP (2023.7)

*) https://github.com/Panasonic-Automotive/remote-virtio-gpu



Panasonic-Automotive / remote-virtio-gpu Public

YoshiteruKawasaki remove invalid link 9a9ac74 15 days ago 4 commits

documentation	Initial commit.	21 days ago
include	Initial commit.	21 days ago
settings	Initial commit.	21 days ago
src	Initial commit.	21 days ago
.clang-format	Initial commit.	21 days ago
CMakeLists.txt	Initial commit.	21 days ago
CONTRIBUTING.md	Initial commit.	21 days ago
LICENSE.md	added LICENSE.md	21 days ago
README.md	remove invalid link	15 days ago

README.md

UNIFIED HMI

Remote Virtio GPU Device (RVGPU)

RVGPU is a client-server based rendering engine, which allows to render 3D on one device (client) and display it via network on another device (server)

About

A UnifiedHMI project to provide a client-server based rendering engine.

Readme
View license
2 stars
1 watching
2 forks

Releases

No releases published

Packages

No packages published

Contributors 2

YoshiteruKawasaki Yoshiteru Kawasaki
SJoukan Seimizu Joukan

Languages

*)https://www.youtube.com/watch?v=k_T2zbEjlA0&list=PL6EdENMl-83iGkh4kpeWFclW5ULIJDqxs&index=12

Achieving a Software-Defined Multi-Display System with Unified HMI - Kenta Murakami, Panasonic @AGL All Member Meeting 2023 Summer

Achieving a Software-Defined Multi-Display System with Unified HMI

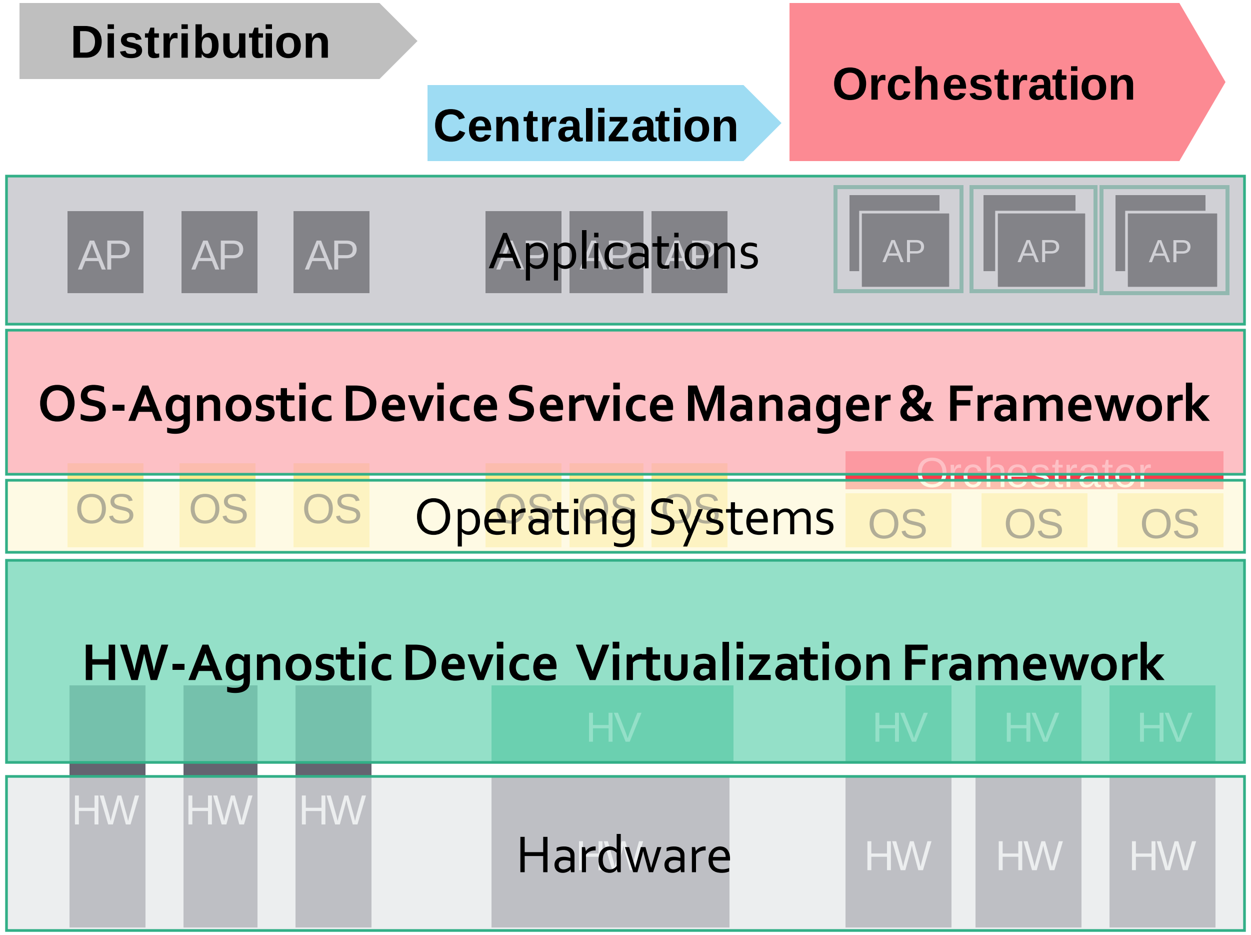
Kenta Murakami, Panasonic Automotive Systems Co., Ltd.

Panasonic
AUTOMOTIVE

Moving Forwards: Constructing a Healthy Ecosystem around Device Virtualization

Ideal Device Virtualization Framework for Software Defined Vehicle

Panasonic AUTOMOTIVE

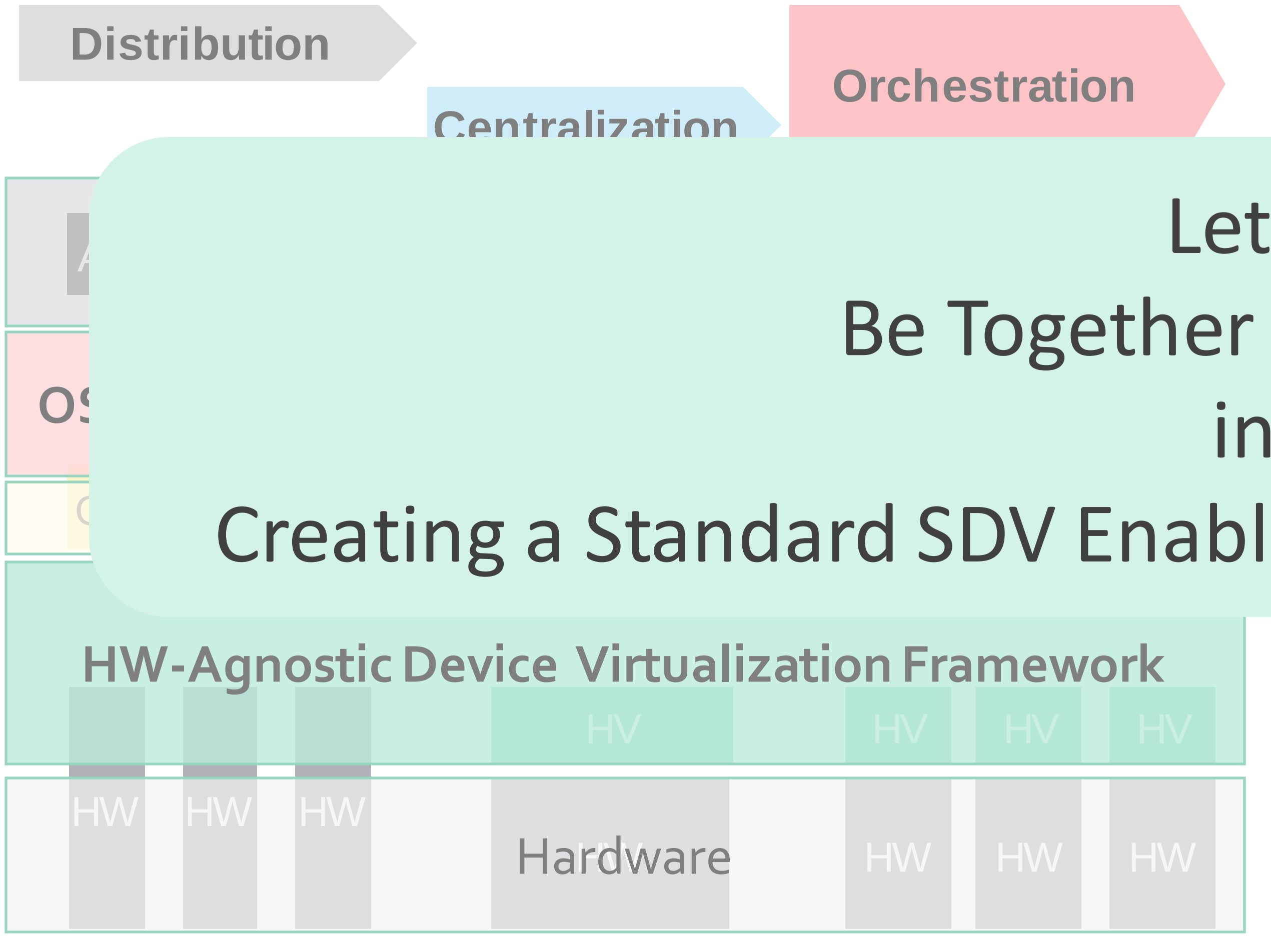


Scalable **O**pen **A**rchitecture **f**or **E**mbedded **E**dge

SOAFEE

Scalable, **O**pen, **A**utomotive, **F**lexible, **E**fficient, **E**ndurable

Ideal Device Virtualization Framework for Software Defined Vehicle



ge

Scalable, Open, Automotive, Flexible, Efficient, Endurable

Let’s

Be Together & Be Open

in

Creating a Standard SDV Enabling Virtualization Framework