Cloud-Native Automotive Edge: AGL for Software-Defined Vehicle

Dec 2022

AGL Virtualization Expert Group Leader

Jerry Zhao, Panasonic Automotive Systems Co., Ltd.

AGL Containers & Mesh Expert Group Leader

Haydn Peterswald, AWS Automotive

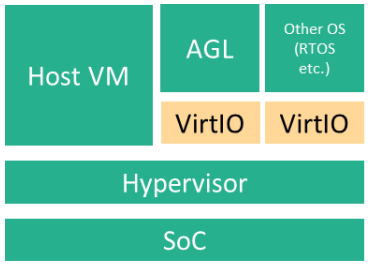
Continuous Evolvement to Software-Defined Vehicle in AGL

Trendy: Software Defined Vehicle

- Trendy: “Software Defined Vehicle”
  - A lot of **hype** surrounding this term
- AGL:
  - Welcome to the party!
  - AGL has been doing this for **8 years**!
  - AGL is addressing all major software functions
  - We have always believed that future vehicles will be defined by software
  - Lines are blurring between embedded software and IT/Cloud
  - Working closely with Yocto, ELISA, CNCF – all under one roof at Linux Foundation

AGL Slide from circa 2015

Virtualization EG



Powerful SOCs and Cockpit Consolidation with hardware-agnostic “VirtIO”

Container and Service Mesh EG

Growing Software Complexity and Scale with “microservice orchestration”

Virtualization EG

Decoupling SW from HW - Device Virtualization with VirtIO to achieve Software-Defined AGL

Jerry Zhao
Supervisor, Software Department, Integrated Control System Development Center, R&D Division
Panasonic Automotive
chou.kensou@jp.panasonic.com

About AGL Virtualization Expert Group

- Start of the Expert Group:
  The expert group was created from a BoF held in the Munich AGL AMM 2016 meeting and started activities from 2017.

- Responsibility:
  The AGL Virtualization Expert Group (Virt-EG) is responsible to design and implement virtualization solutions for AGL.

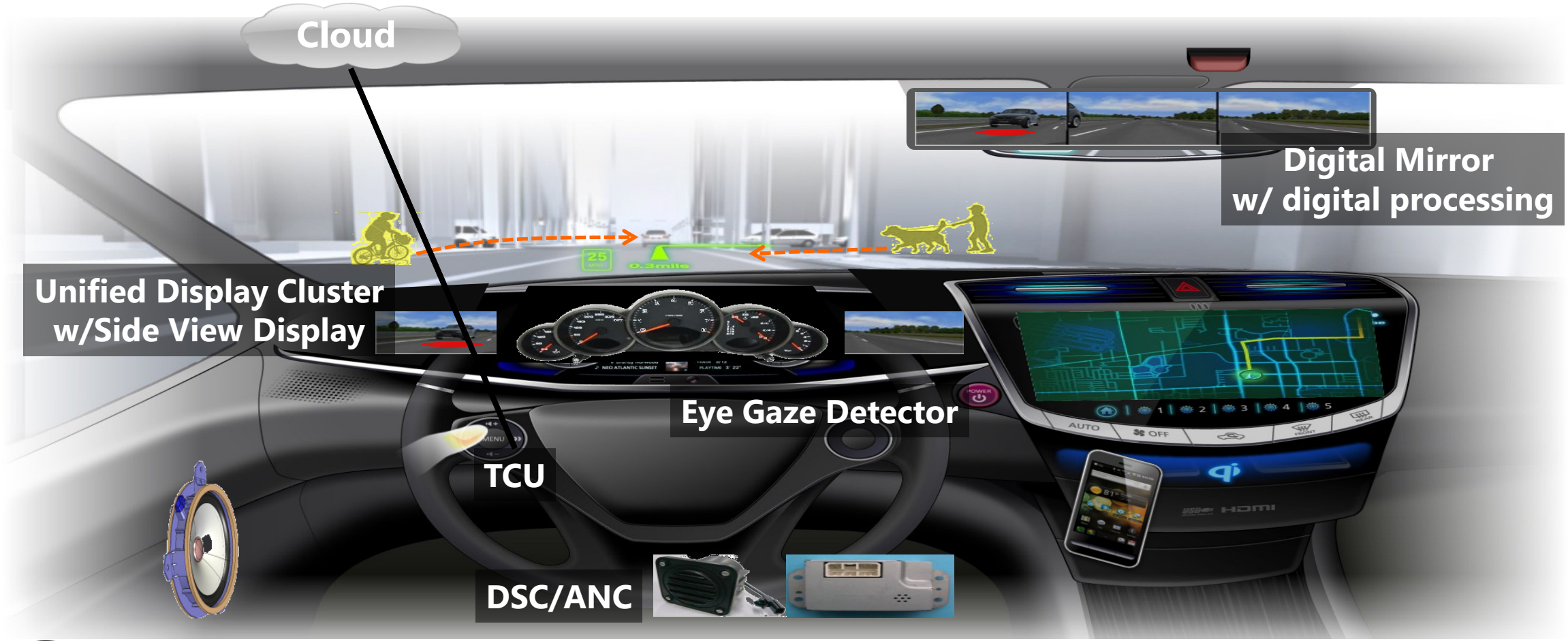
- EG Members:
  On average 10~20 members from different fields are joining the bi-weekly call.

- Interest Points:
  Hypervisors, Containers and any other virtualization solution for x86/ARM. **Nowadays, Virt-EG is focusing on applying and extending VirtIO for diverse AGL use cases.**

Device Virtualization for Software Defined Vehicle

Cockpit Digital Transformation

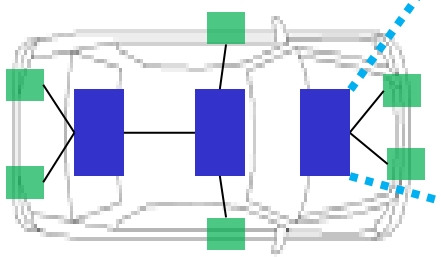
Cockpit is going to be filled with digital instruments.

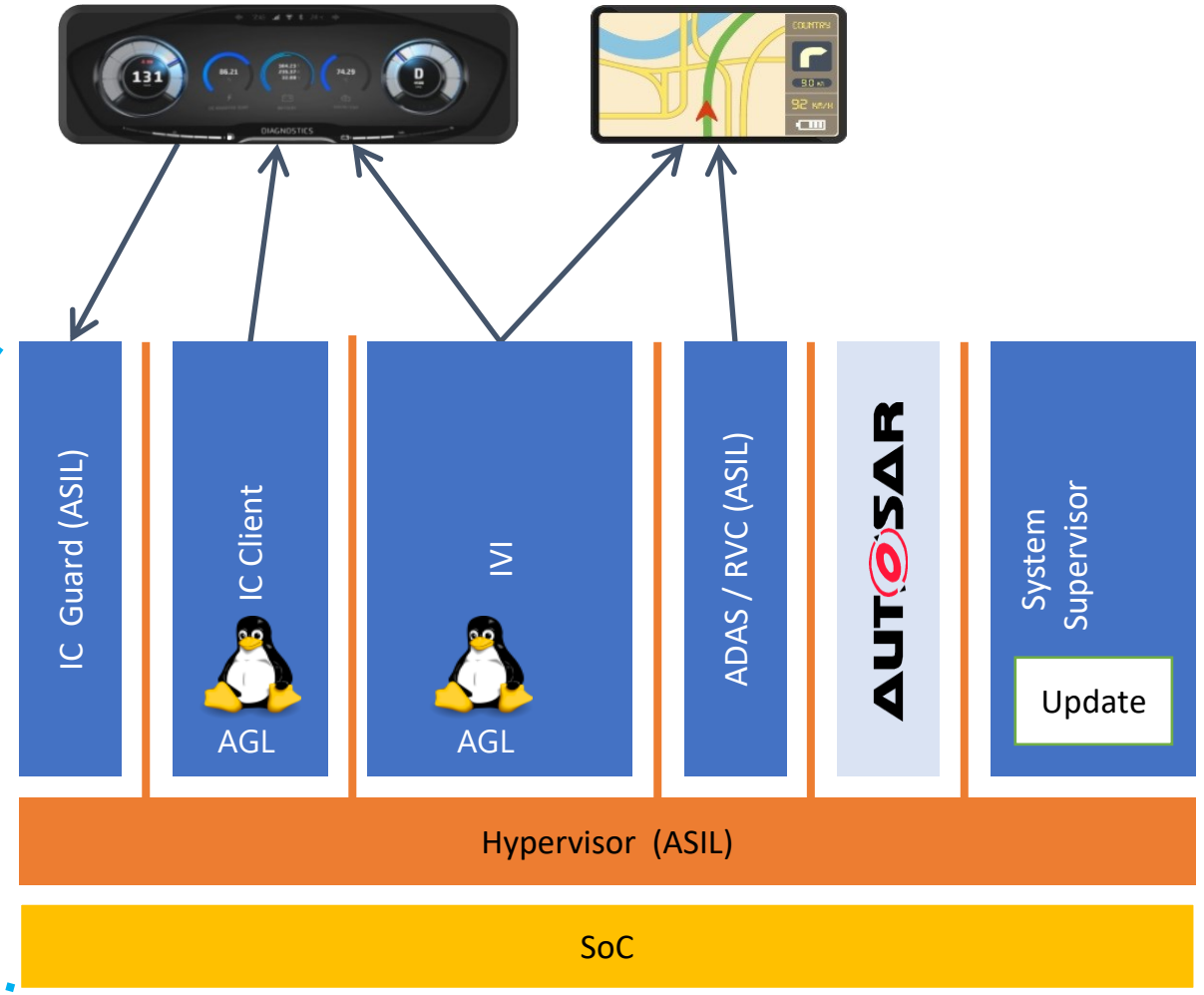


Excerpt from Panasonic's Keynote Presentation at the AGL AMM July 2020

ASIL/Security Partitioning with Virtualization

- Computing in a vehicle is going to be performed by a smaller number of ECUs.
- Hypervisor separates functions with different operating systems, different requirements on real-time behavior and functional safety
  - Non-safety-critical functions may be run as VM or containers.
  - Safety-critical functions (e.g. displaying tell-tales and local ADAS functions) underling ASIL requirements are run under a separated VM.





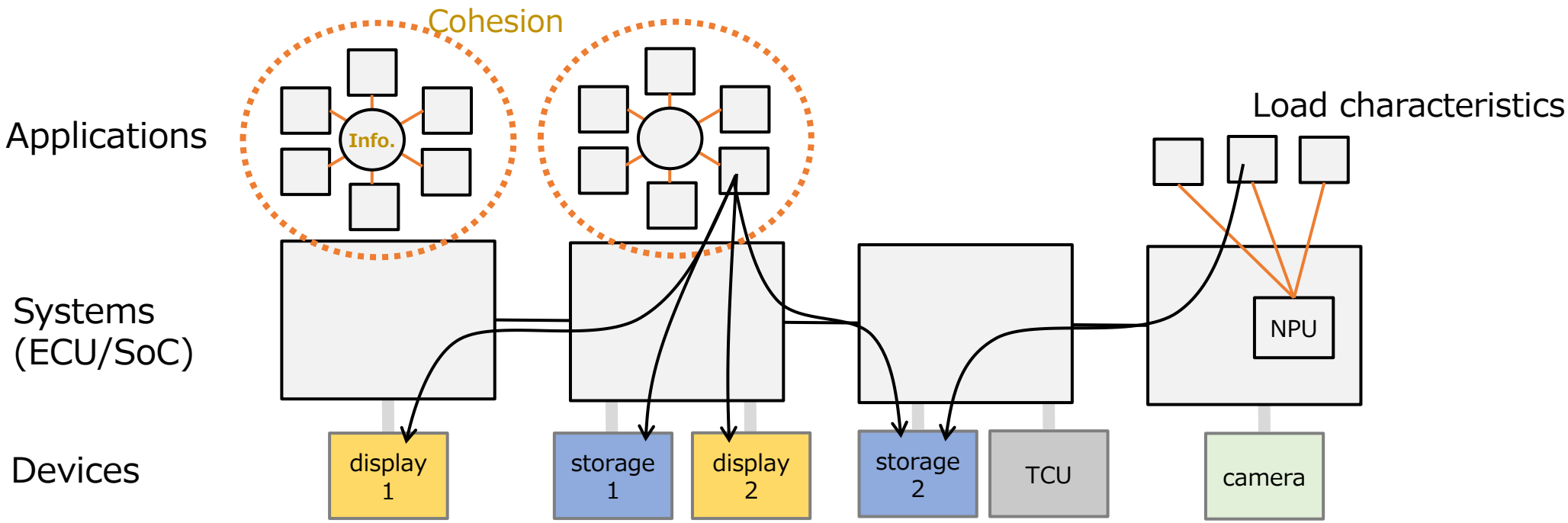
Example Integrated ECU Architecture with Virtualization

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

Device Virtualization: Specific Necessity in Automotive

***Common abstraction* of diverged devices among car models and *location transparency* of devices** are especially critical for application software asset.

- Diversity of Devices due to Various Car Models
- Highly Distributed Architecture and conflicts between optimized allocation policies of applications and devices

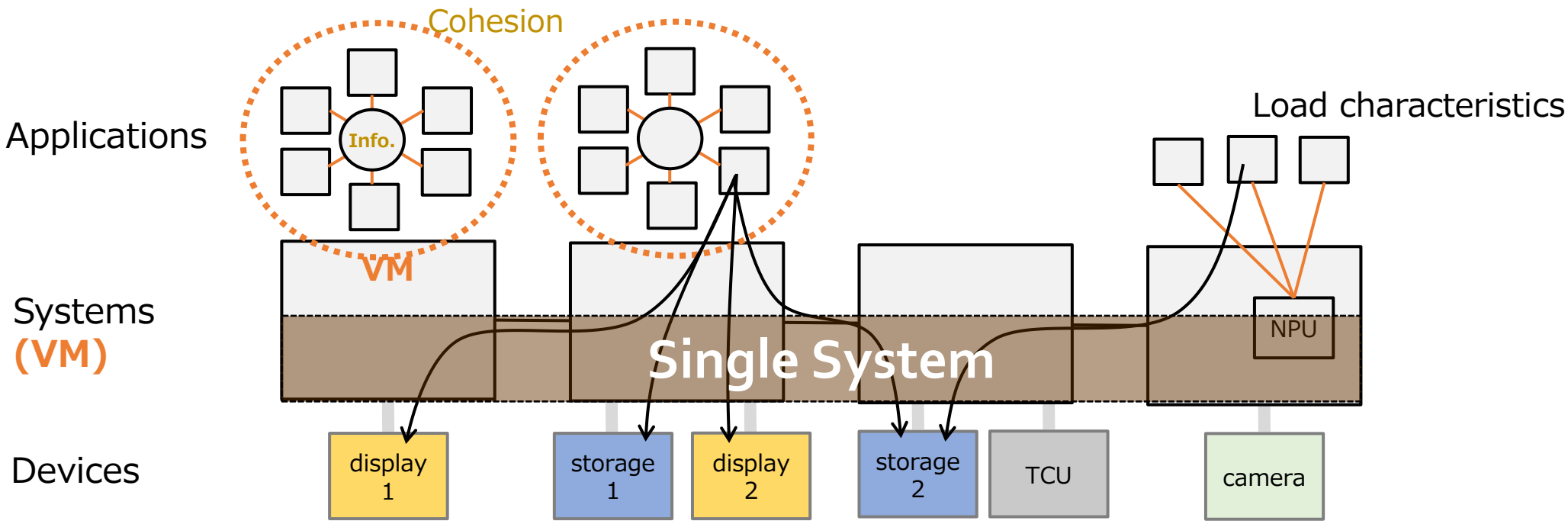


Excerpt from Panasonic's Keynote Presentation at the AGL AMM July 2020

Device Virtualization: Specific Necessity in Automotive

***Common abstraction* of diverged devices among car models and *location transparency* of devices** are especially critical for application software asset.

- **Even for centralized ECU, the same argument applies because the system inside is divided to multiple VMs**

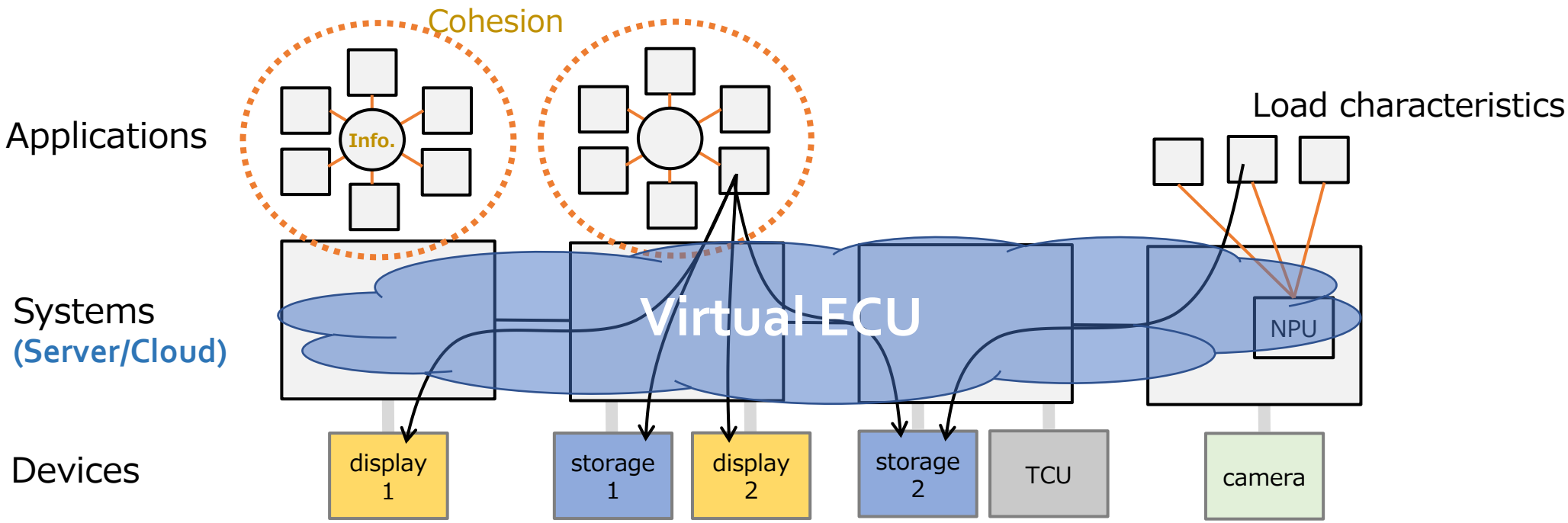


Excerpt from Panasonic's Keynote Presentation at the AGL AMM July 2020

Device Virtualization: Specific Necessity in Automotive

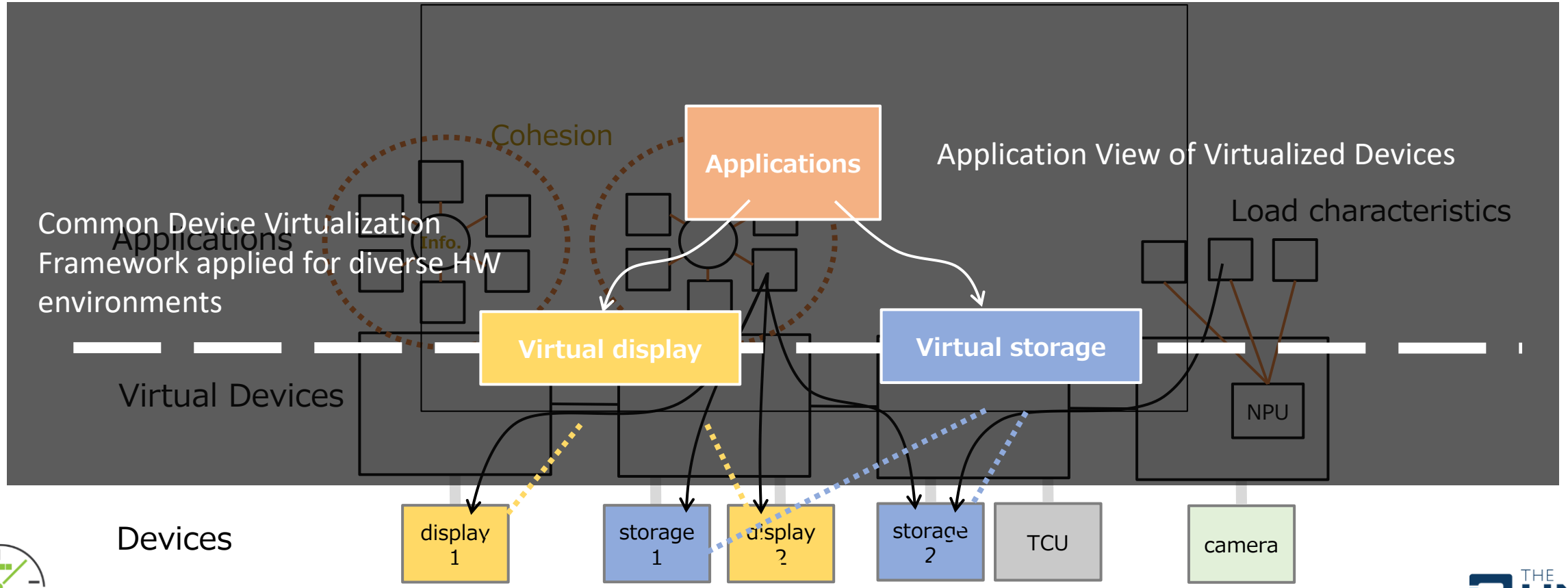
***Common abstraction* of diverged devices among car models and *location transparency* of devices** are especially critical for application software asset.

- **In addition, same issue will apply to Virtual ECU constructed in Server/Cloud where has completely different device natures from the one on the automotive edge.**



Device Virtualization: Key to Software Defined Vehicles

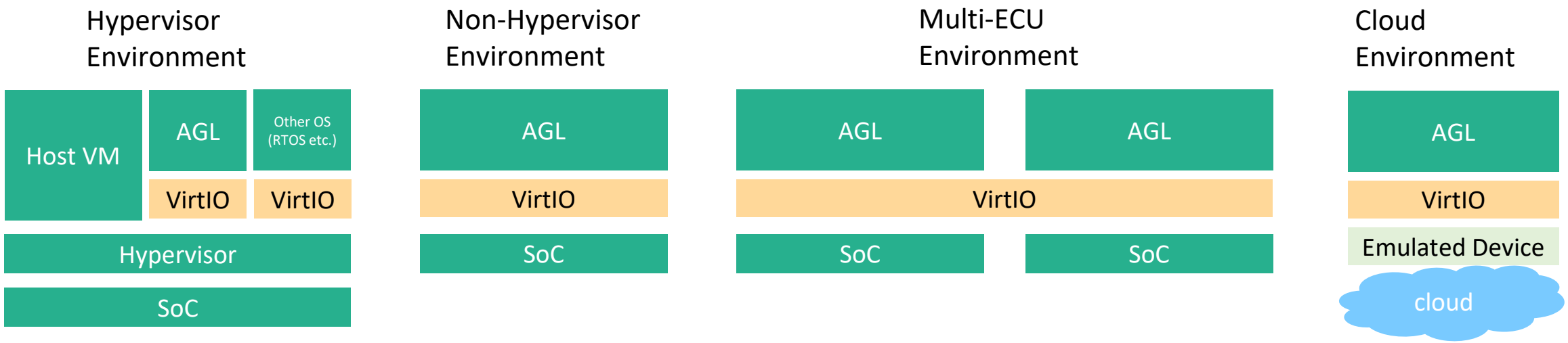
Software Defined Vehicle needs a common device virtualization framework to decouple software implementation from diverse hardware targets across vehicle variants/generations, architectures (single/multiple-ECU) and development environments (real/virtual ECU)



VirtIO as A Common Device Virtualization Framework

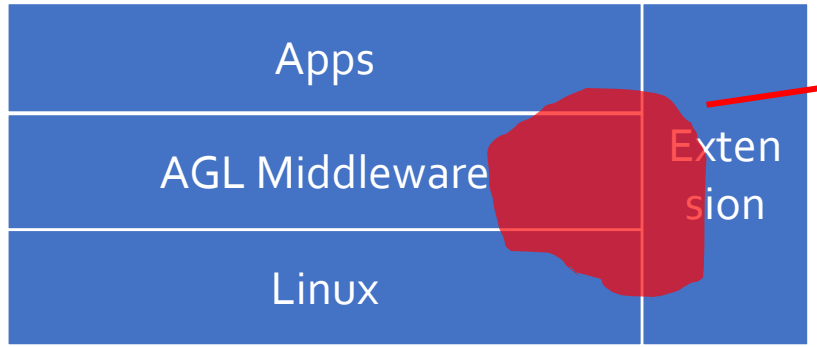
Overview of Device Virtualization in AGL - Concept

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



Pains around Virtualized AGL in the Past

Fragmentation
Serious Barrier
For "Virtualization
Ready BSP"



Dependency
on specific virtualization solution

Must Adapt to Every Single Incompatible Interface!

Proprietary Para Virtual Devices XA

Proprietary Para Virtual Devices YA,YB...

Proprietary Para Virtual Devices ZC,ZD

Hypervisor X

Hypervisor Y

Hypervisor Z

Limited Freedom to Choose Hypervisor and SoC Combination!

Proprietary BE Device Drivers XA, YA

SoC A

Proprietary BE Device Drivers YB

SoC B

Proprietary BE Device Drivers YC, ZC

SoC C

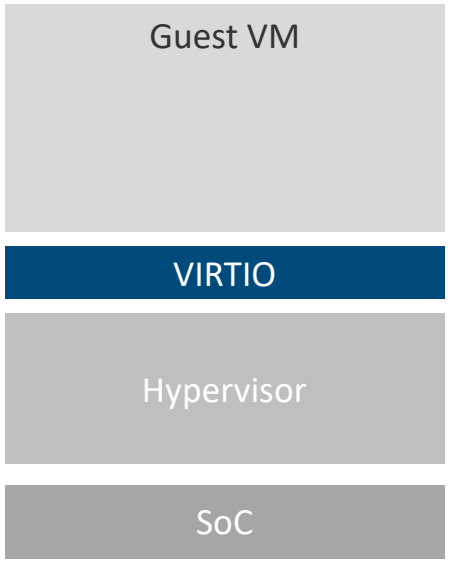
Proprietary BE Device Drivers ZD

SoC D

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

Enter Standard Virtualization Framework - VirtIO

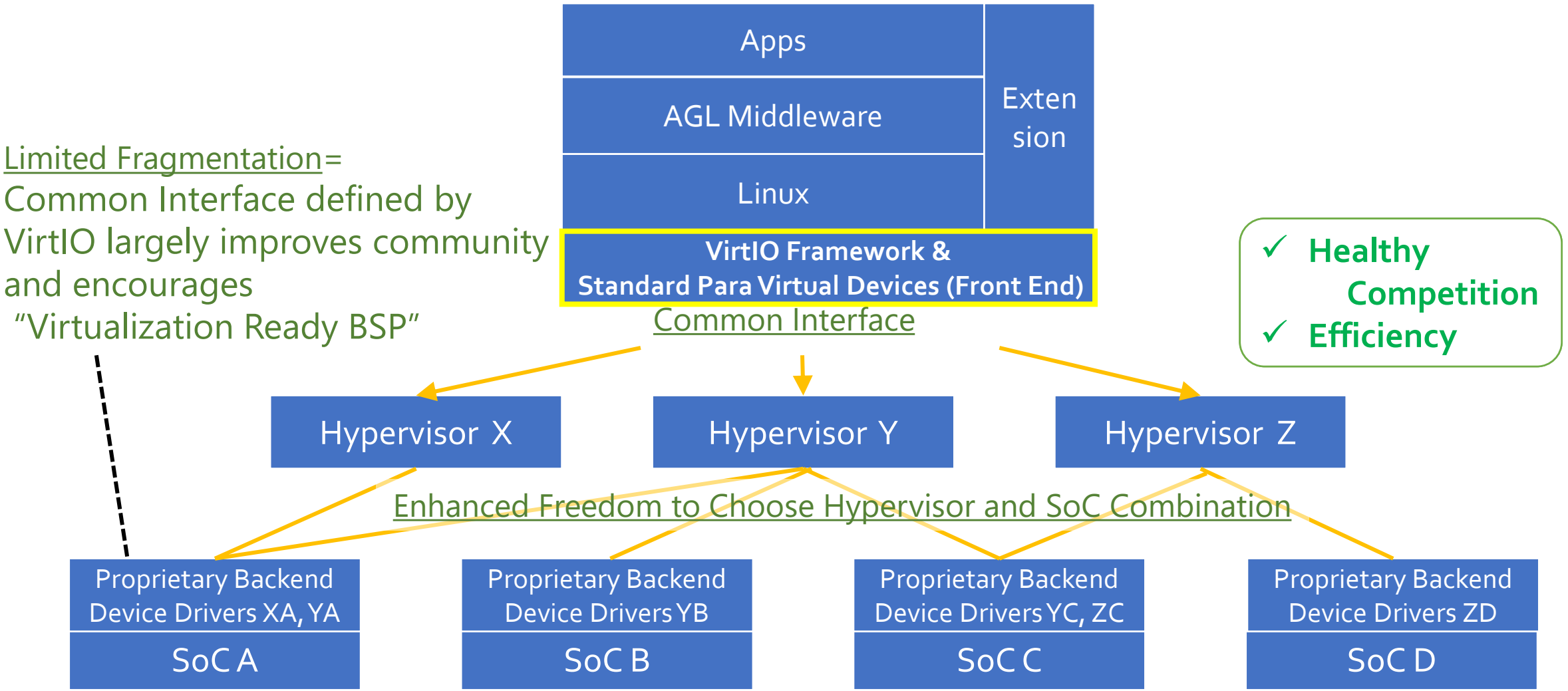
- Developed in 2008 as a hypervisor neutral way of accessing devices
- Provide virtual machines access to Input/Output
- A standardized interface for I/O between virtual machines and hypervisors
- Abstract device functionality instead of hardware
- Drivers are widely available in all major operating 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

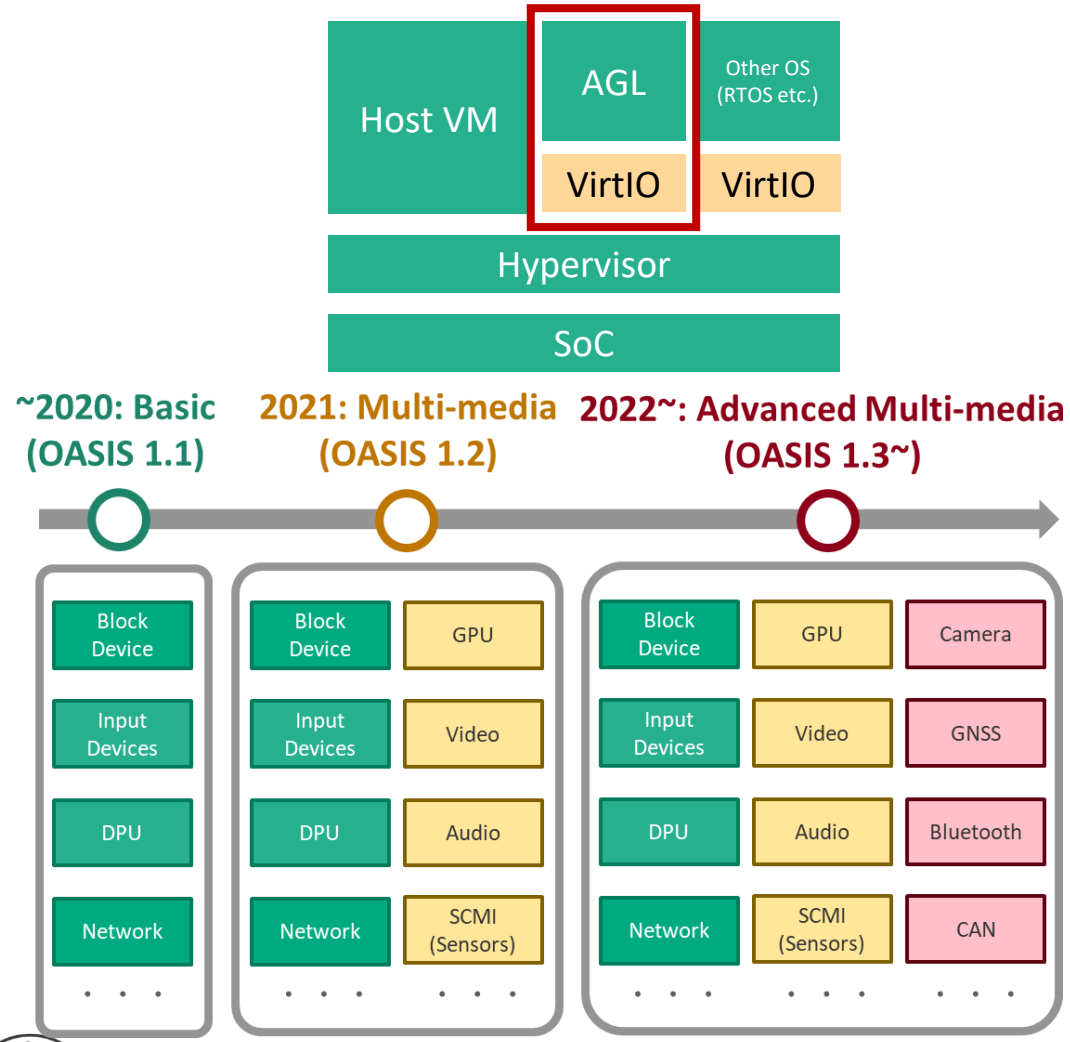
Excerpt from Virt-EG Member Company OpenSynergy's Presentation at AGL F2F Oct 2022

VirtIO as a Common Framework for Virtualized AGL



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

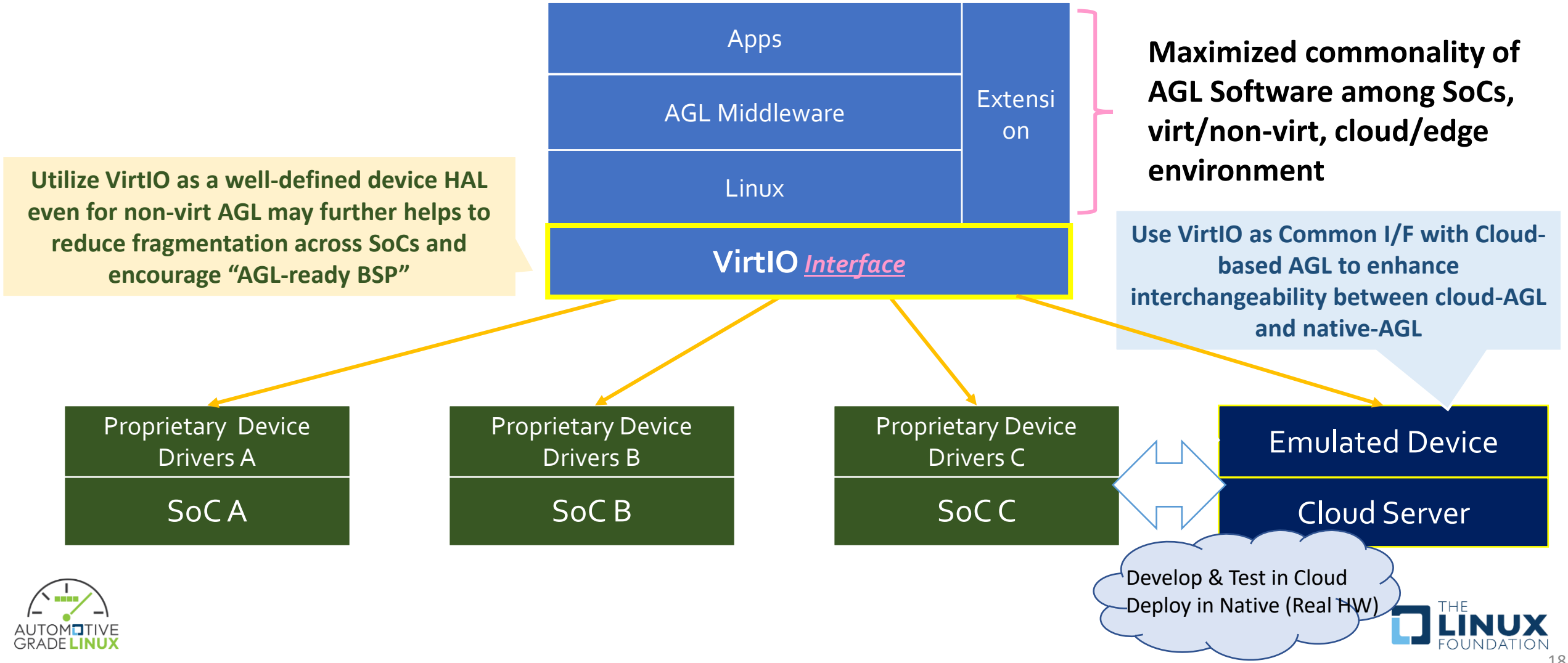
VirtIO Work for Hypervisor Environment



Virt-EG VirtIO Device Development Focus

- Kooky Koi (2021.2): VirtIO has been officially supported by AGL as common device framework for Virtualized with basic device support.
  - Block device: virtio-blk
  - Console: virtio-console
  - Display and GPU: virtio-gpu
  - Touch Panel: virtio-input
  - Network: virtio-net
  - Random Number Generator: virtio-rng
  - Virtual Socket: virtio-socket
- Lucky Lamprey (2021.7):
  - New feature in existing virtio devices
    - Multi-touch support in virtio-input
  - New virtio devices
    - Sound Card: virtio-sound
- Magic Marlin (2022.2):
  - New virtio devices
    - Video Decoder/Encoder: virtio-video
    - Camera: Camera streaming features based on virtio-video
- Nifty Needlefish (2022.7):
  - New virtio devices
    - SCMI: virtio-scmi (acceleromoter & gyroscope sensors)
    - Bluetooth: virtio-bt

VirtIO Beyond Virtualized AGL



VirtIO Work for Non-Hypervisor Environment

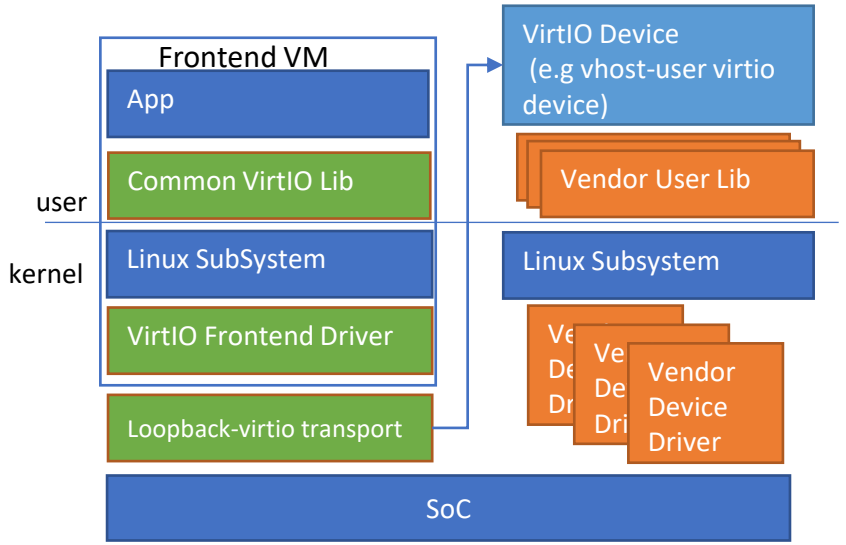
Non-Hypervisor Environment

- Finished Design & Implementation of a common virtio-based HAL layer “virtio-loopback” portable to execute on both native and virtual environments
  - Implemented use cases with block, RNG and input device
  - Added “touch sensitivity control” features from IVI-PR to existing virtio-input
- Ready to start next-step about future support activities for IVI-EG/IC-EG

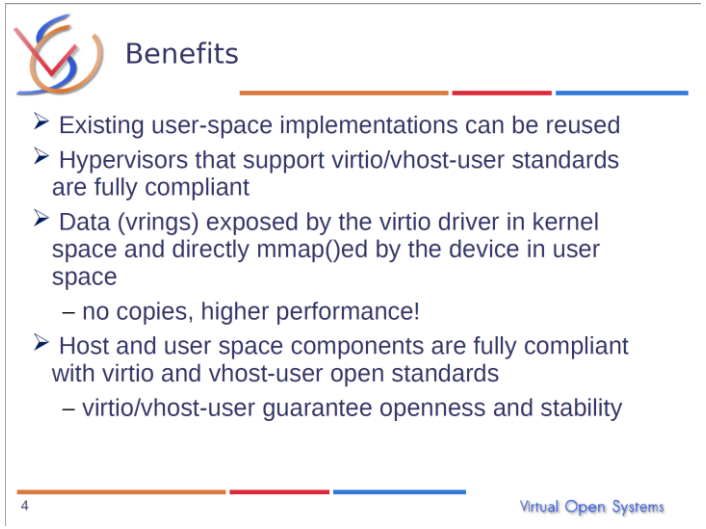
Priority of Device Virtualization Voted by each AGL EG (2021)

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

High Level Architecture Design



Features & Benefits Discussion in EG



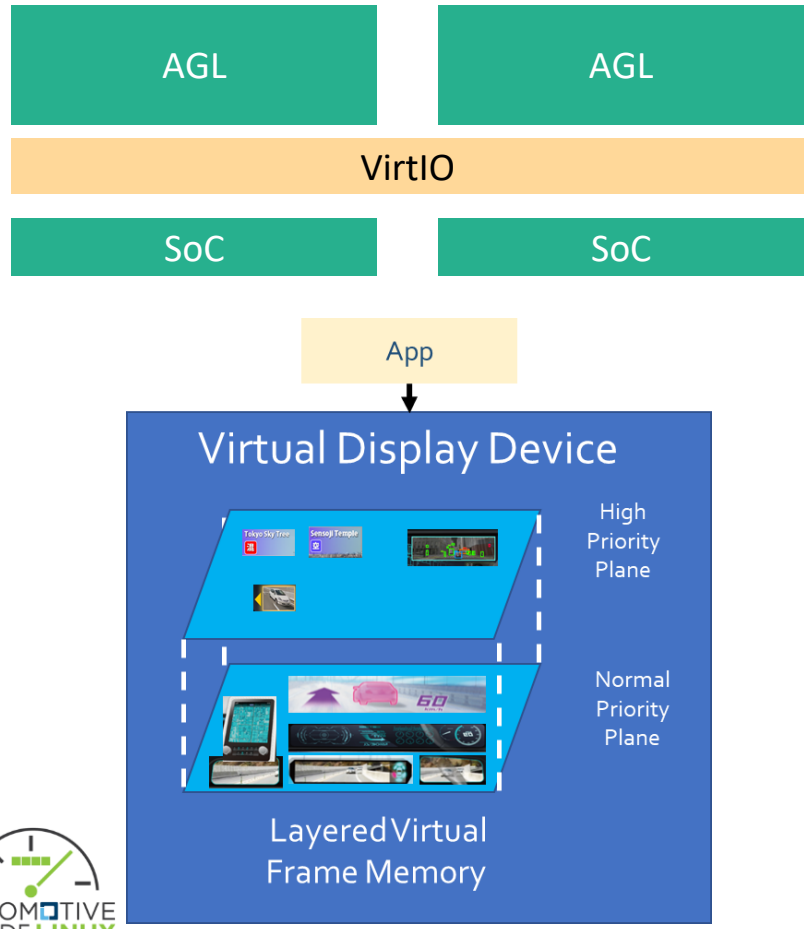
AUTOMOTIVE GRADE LINUX

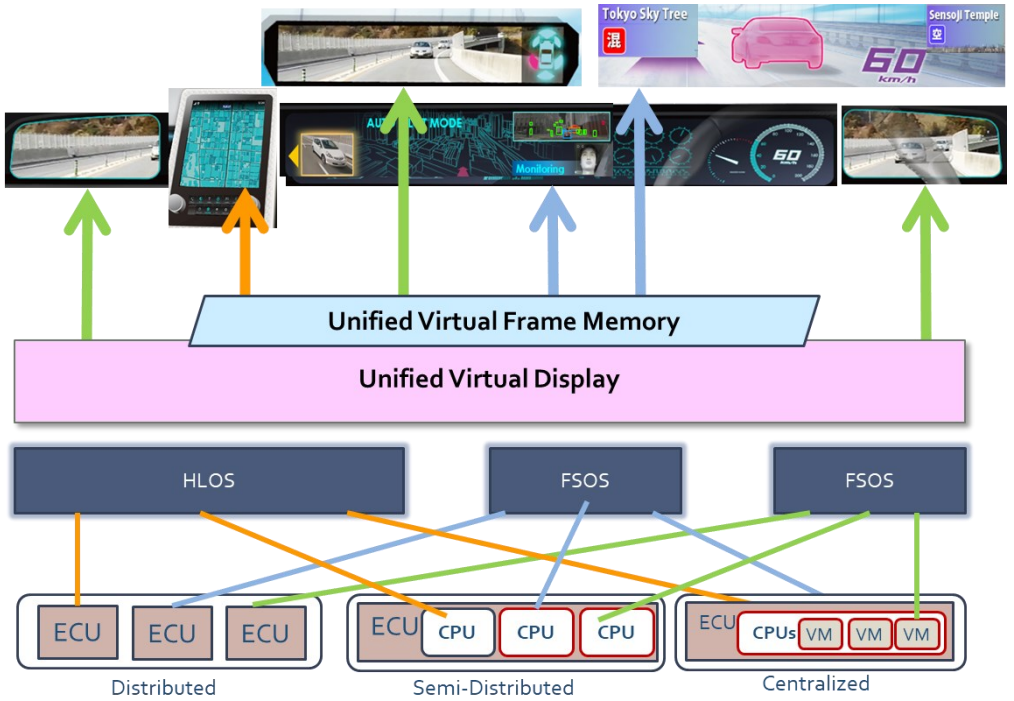
THE LINUX FOUNDATION

VirtIO Work for Multi-ECU

A Unified Virtual Display based on VirtIO-GPU (“Unified HMI” technology) can be established to have Integrated control of multiple display on distributed SoC systems

Multi-ECU Environment



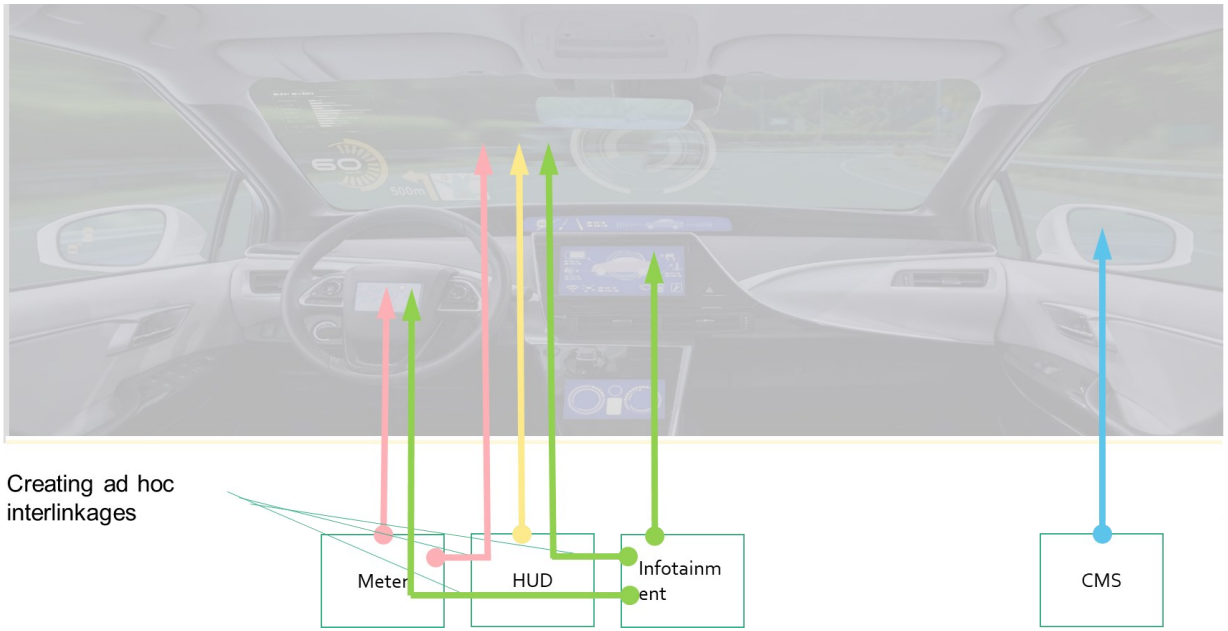


- **Mappig multiple physical displays** of cockpit & cabin into a **single large virtual display**
- **Rendering each application** to its **arbitrary region**

VirtIO Work for Multi-ECU

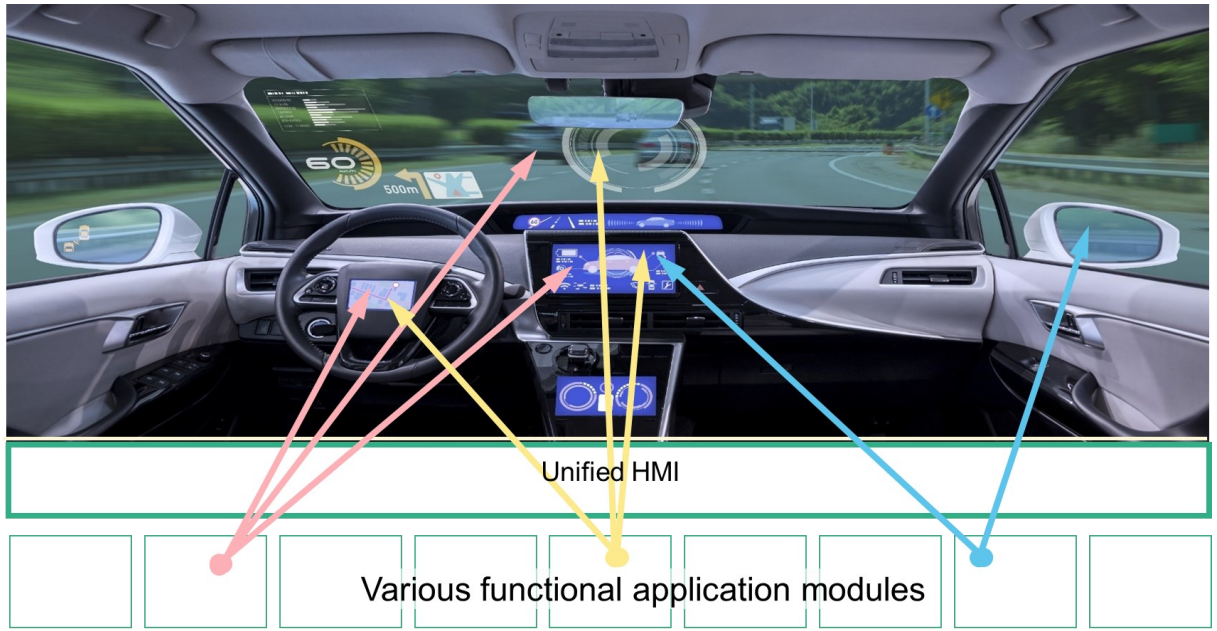
Legacy HMI System

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

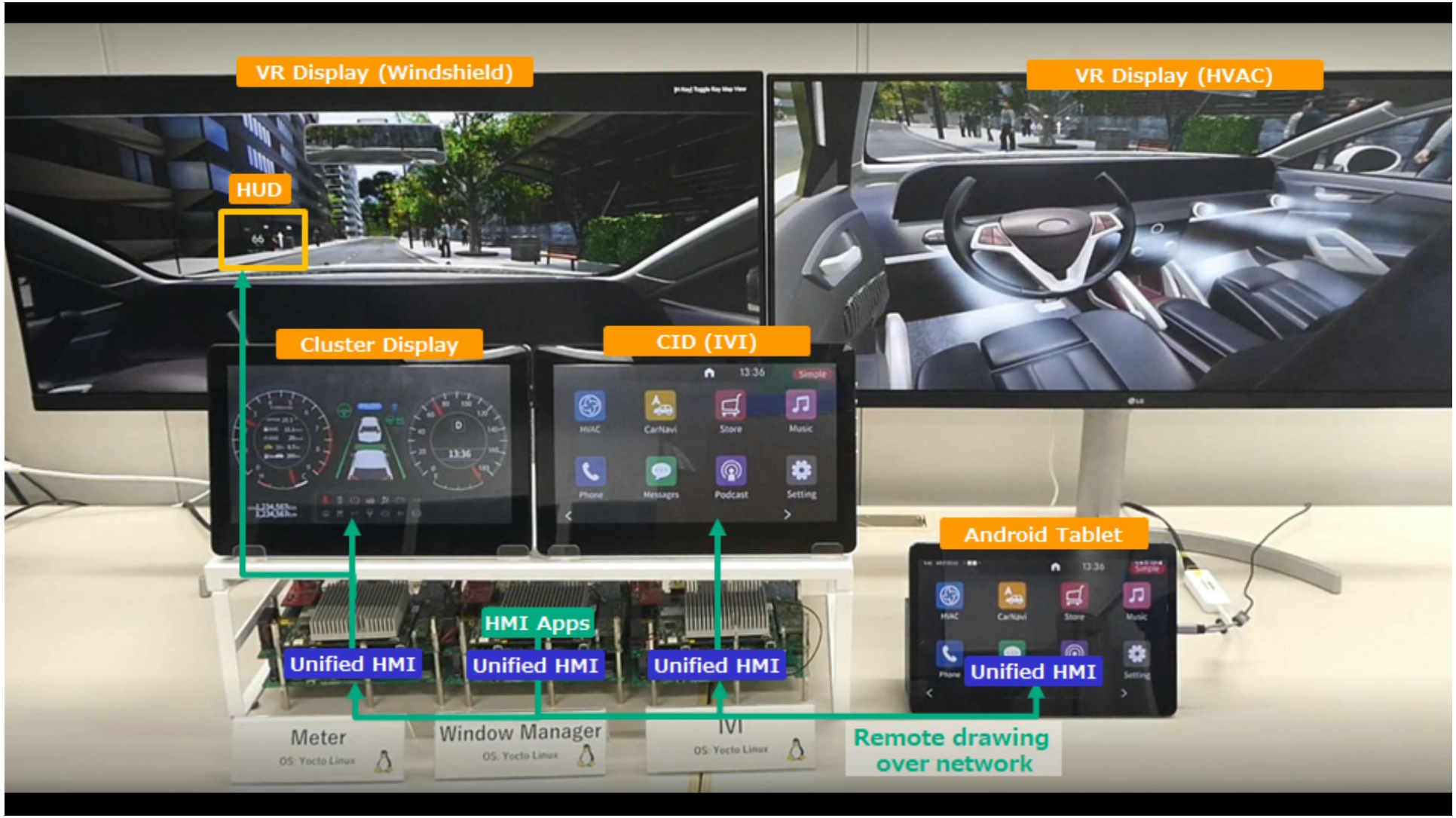


Unified HMI System

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

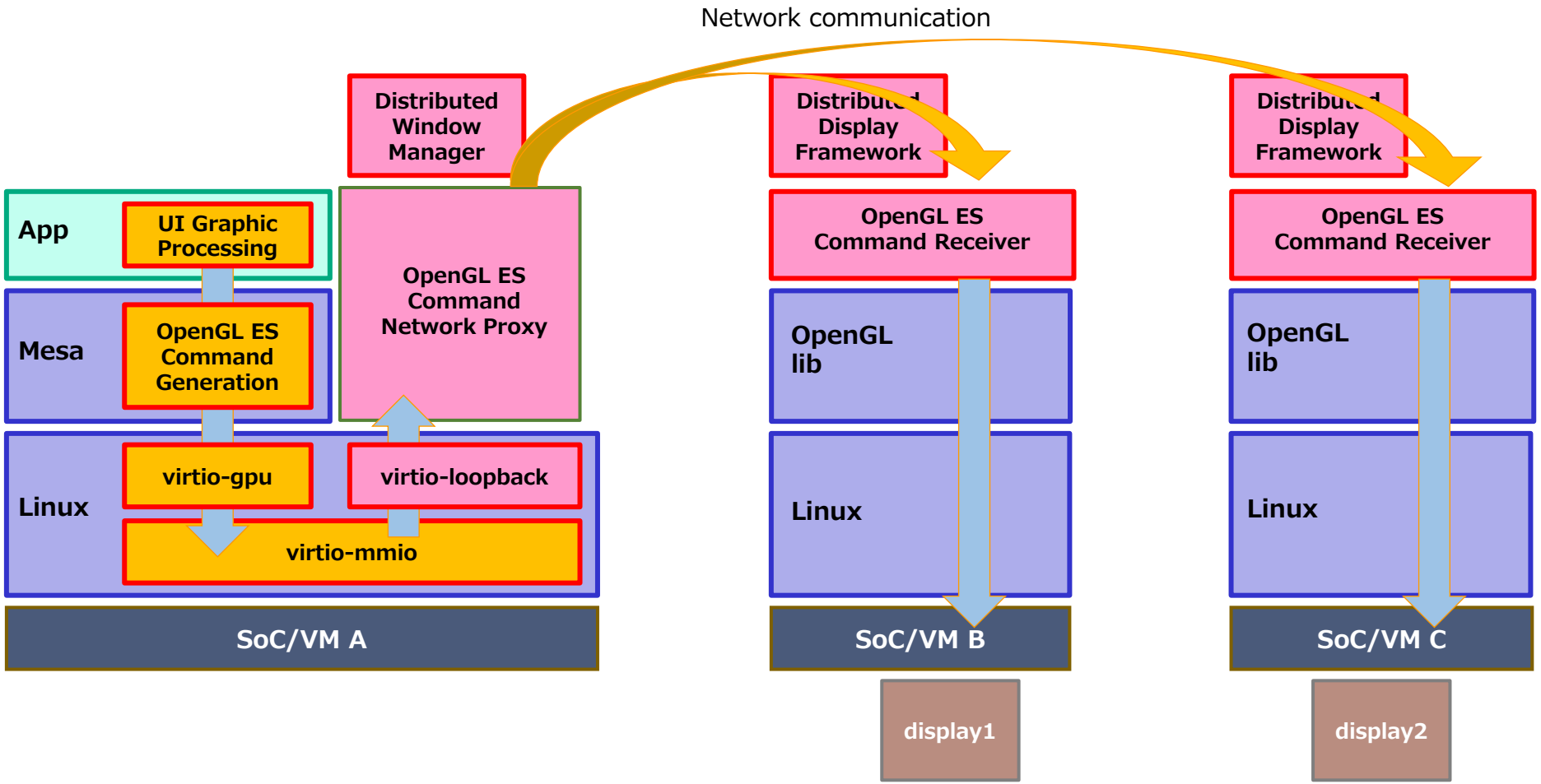


EG Member – Panasonic's Unified HMI Demo



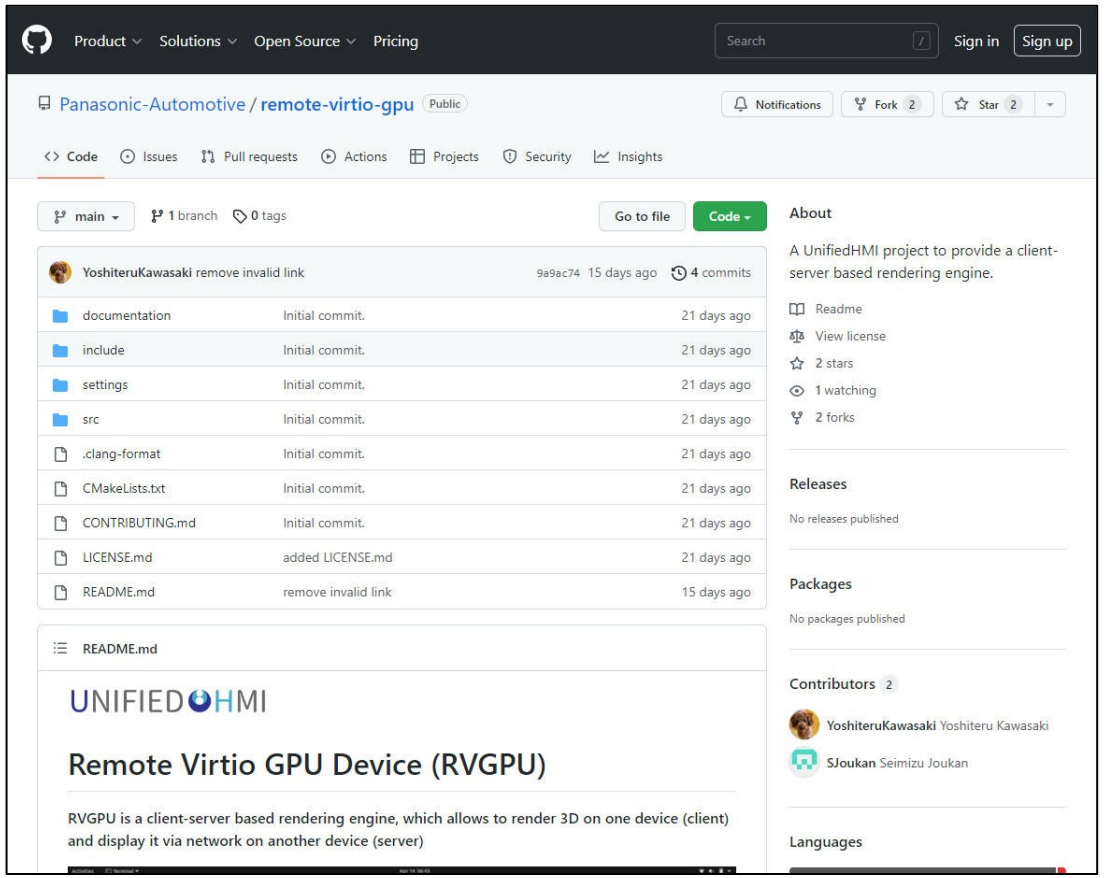
Unified HMI Schematic Architecture

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



Unified HMI OSS Plan

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



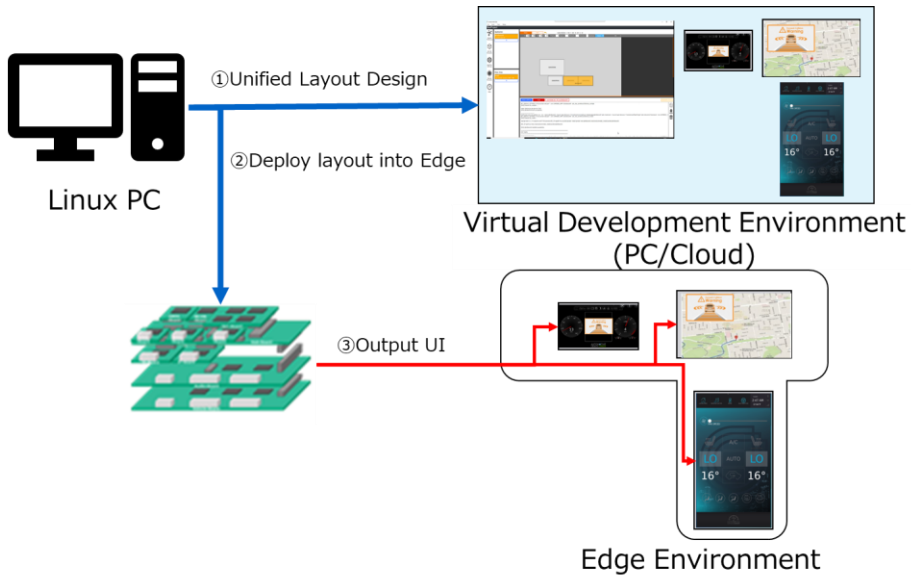
OSS Roadmap for Unified HMI

Publish Unified HMI on GitHub* (2022.9)

Apply Unified HMI to AGL UCB OO/PP Release (2023)

Unified HMI Demo in CES (2023.1)

We will show an AGL-integrated Unified HMI Demo with cloud-native features in the CES AGL Booth.

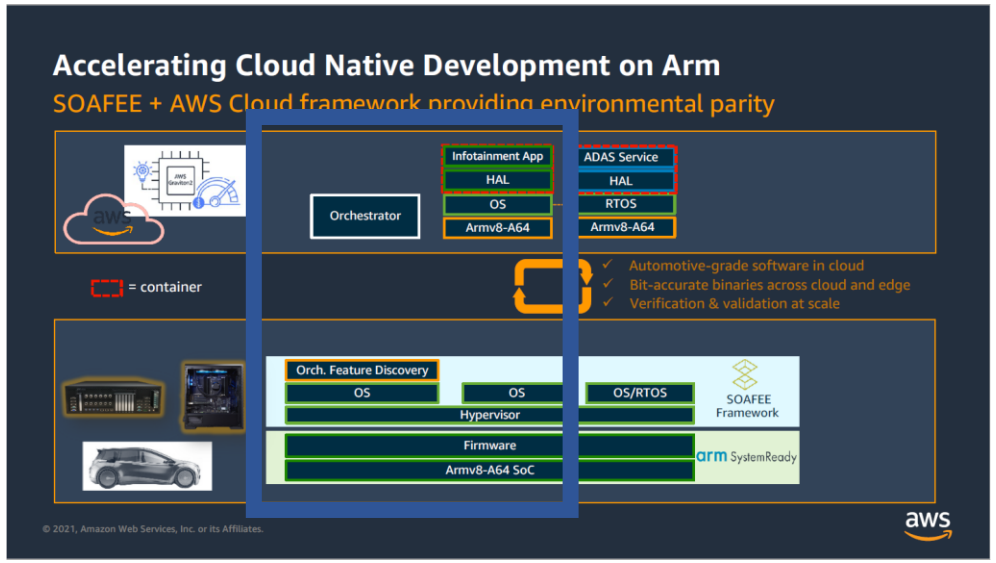


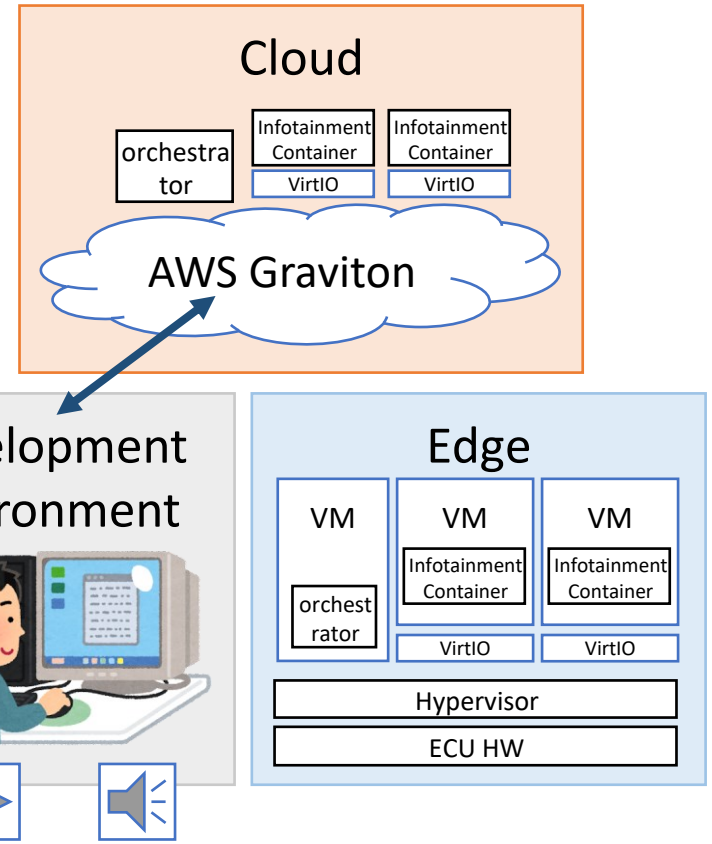
Unified layout of multi-displays able to developed on PC/cloud and seamlessly deployed to automotive edge

VirtIO Work for Cloud-Native Environment - AGL on AWS Graviton

- **Establish a reference environment of cloud-native AGL**
- **Make VirtIO & Orchestration work on both cloud and edge AGL instances, and enable developer to develop HMI services on cloud environment which graphic & audio can be verified on local clients**

Instrument Cluster & IVI which are most related to UI/UX and need rapid development & OTA are one of the automotive devices that enjoy benefits from cloud-native most.





Graphic & Sound are necessary for development with Cloud

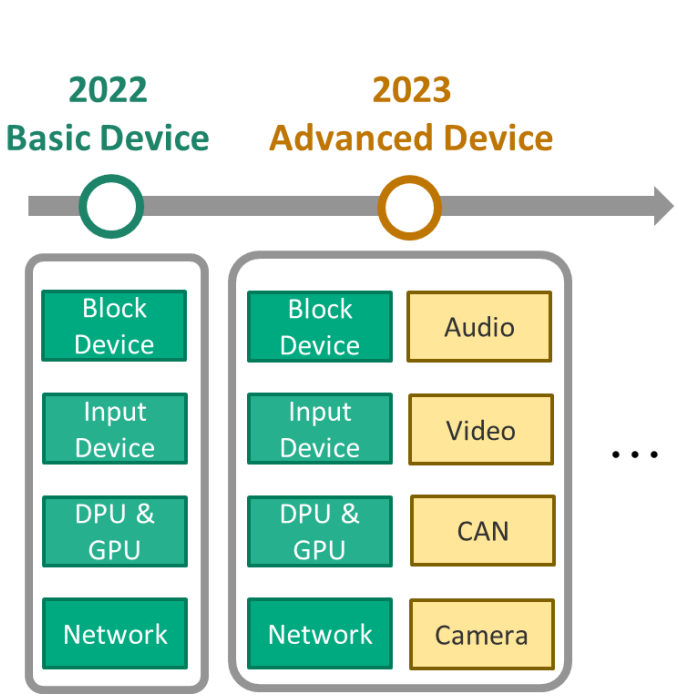
VirtIO Work for Cloud-Native Environment - AGL on Mac

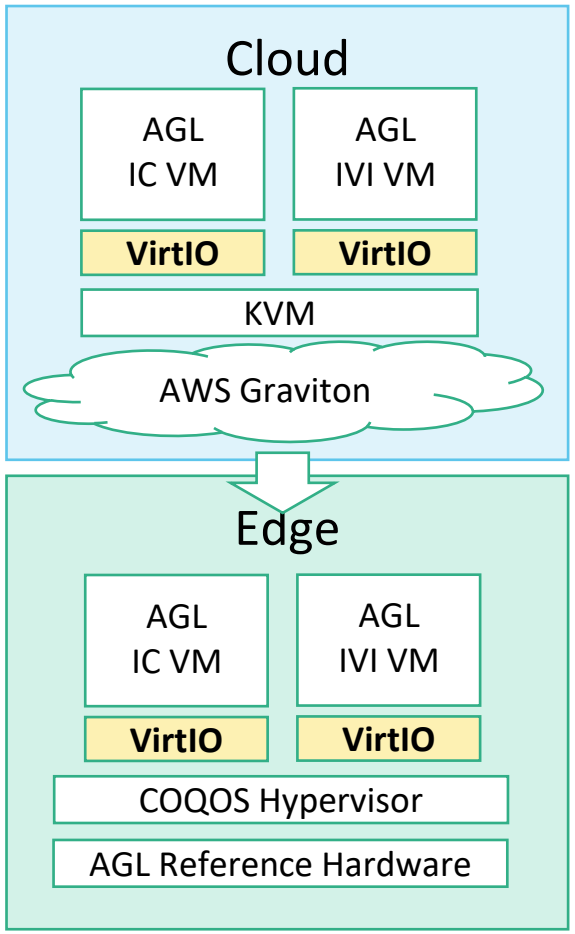
- **EG Member Francois from the company Shokubai has created a demo to run the AGL with VirtIO on the top of MacBook with Apple MacOS 13 virtualization framework.**
- **This can be done without any changes to AGL thanks to VirtIO.**

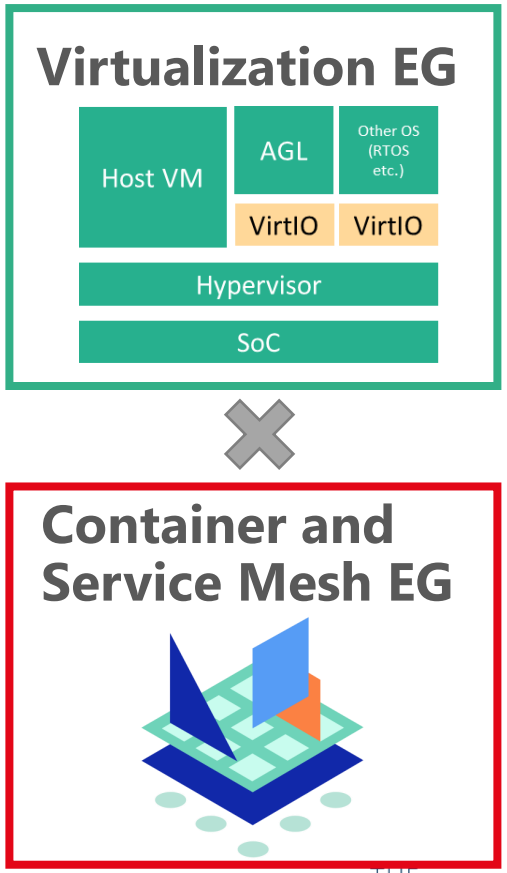
https://www.youtube.com/watch?v=5DT-l2sWeVY

VirtIO Work for Cloud-Native Environment – Demo@CES2023 AGL Booth

- Collaborative efforts between Virt-EG and Container-EG to enable cloud-native AGL
- Identical AGL IVI binary is now able to run on both cloud and edge with graphics visible
- A Joint demo will be shown in the CES2023 AGL Booth







Containers & Mesh EG

Cloud-Native Computing on Automotive Grade Linux

Haydn Peterswald
WW Tech Leader – Digital Customer Engagement
AWS Automotive
haydnpet@amazon.com

Agenda:

- Challenge facing Automotive Industry
- Scope of the Containers & Mesh Expert Group
- AGL on AWS
- Expert group roadmap for the next months

Trends and Challenges driving the need for an SDV strategy

- Connected
- Autonomous
- Shared
- Electrification

- Safety features
- Quality and security
- Wiring harness costs
- Integration testing, V&V
- Regional regulatory differences
- Supply chain fragility
- Innovation

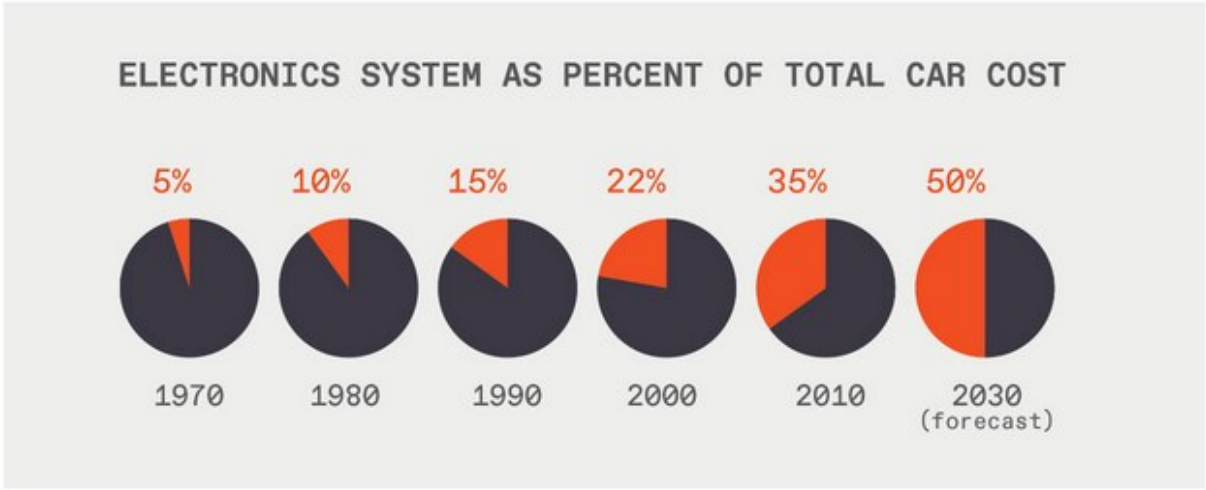
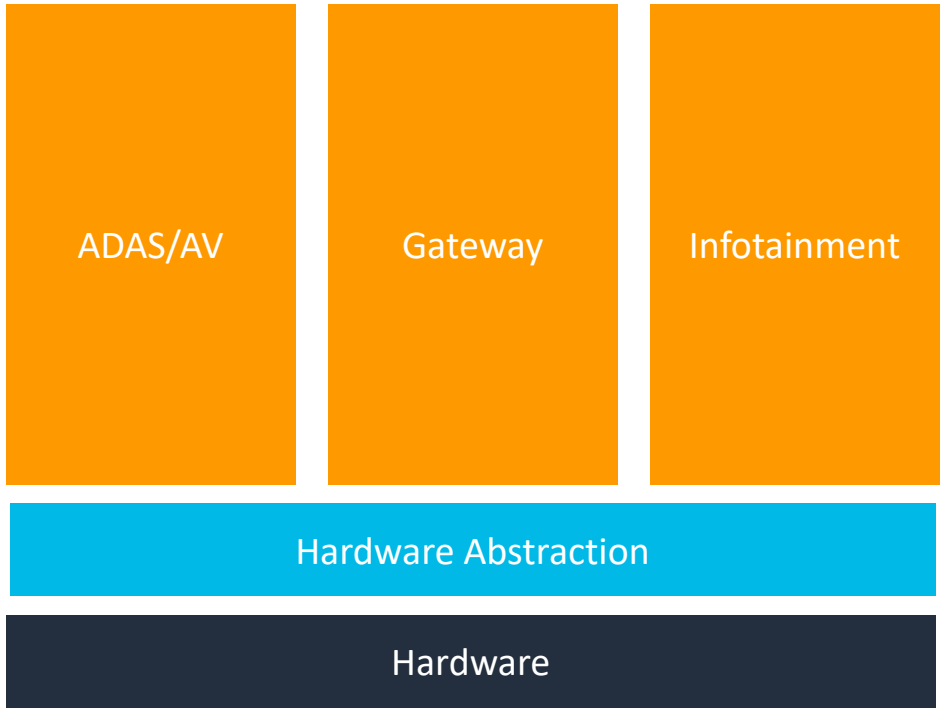


Chart: Mark Montgomery; Source: Deloitte Touche Tohmatsu Limited

https://spectrum.ieee.org/cars-that-think/transportation/advanced-cars/software-eating-car

What is “Software Defined”? How to make it possible?



Develop in Cloud and deploy into in-vehicle hardware

- Functions enabled by software are **abstracted** from hardware
- Functions are enabled using a cloud-native **microservices** development model:
  - Delivered as self-contained units of software (**virtual ECU** packaged in containers)
  - Mechanism for publishing available services and their interfaces and characteristics
  - Authorization of the use of these services by other functions
- Functions are **continuously updatable** over the air

Architectural considerations for Software Defined Vehicles

HW Consolidation & Virtualization

By virtualizing the sensors, networking, and hardware interfaces, customers can achieve parity between cloud and vehicle

In-Vehicle Microservices

Provide a normalized, consistent, secure data access layer that allows developers to create new insights & microservices

Application Encapsulation

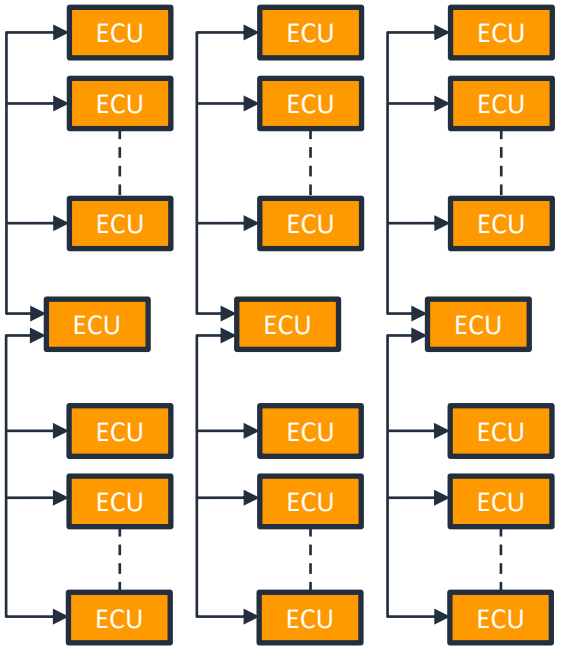
Containers provide self contained, isolated, easily distributable packages for development of virtual ECUs deployed to vehicles

Automotive DevOps

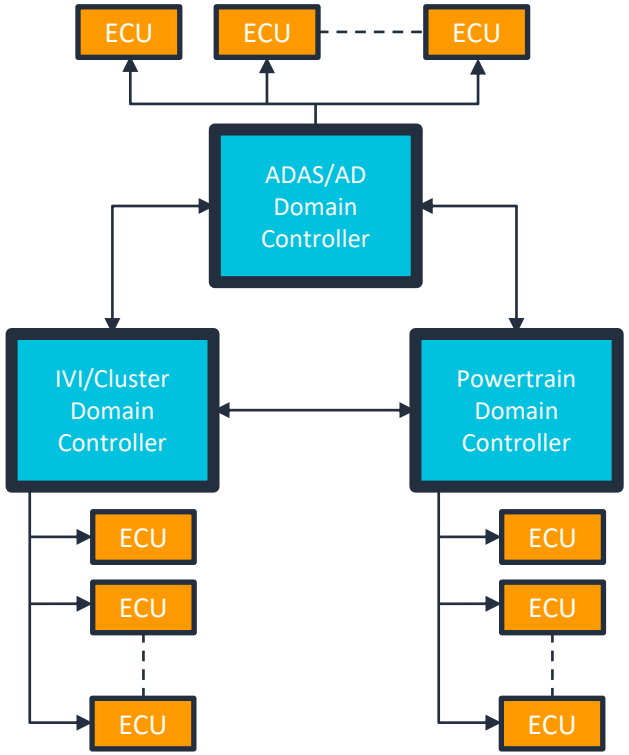
- Agility with managed **CI/CD** to more rapidly and reliably build and deliver services inside the vehicle
- **Mixed-critical management** of workloads from edge-to-cloud
- **Environmental parity** and optimized execution environment
- **Application-level networking** to discover services and enable communication across multiple types of compute infrastructures

In-vehicle hardware consolidation enables cloud native

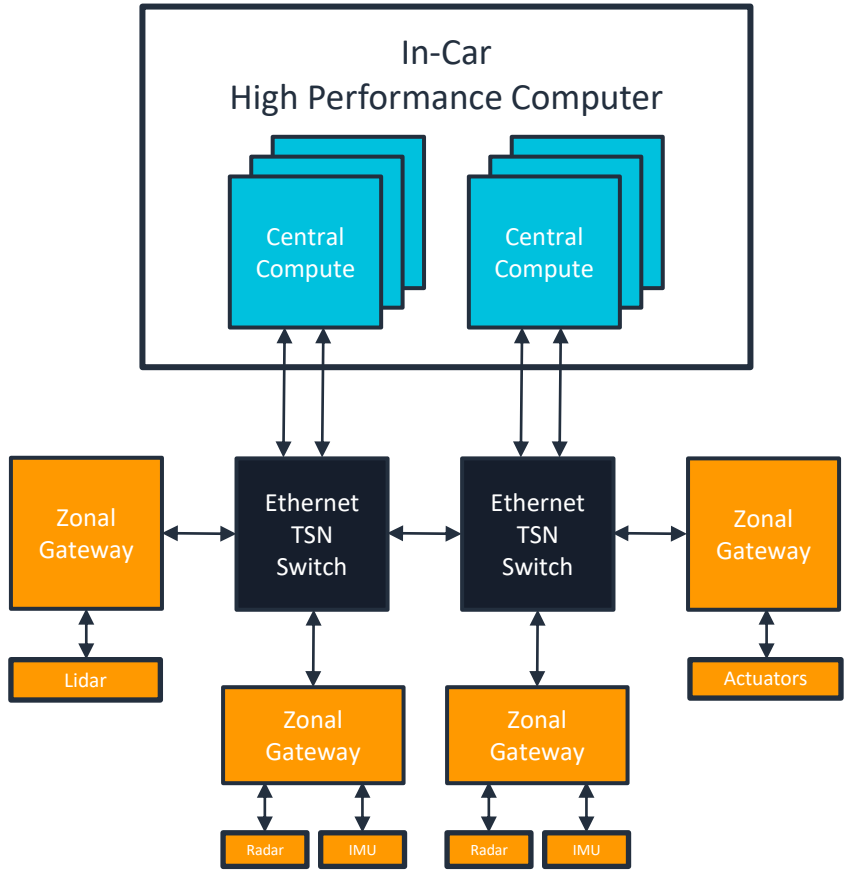
Traditional Architecture



Domain Architecture



Zonal Architecture



The shift towards centralized architectures is leading to “Datacenter on Wheels” design paradigm paving the way to cloud native software in automotive

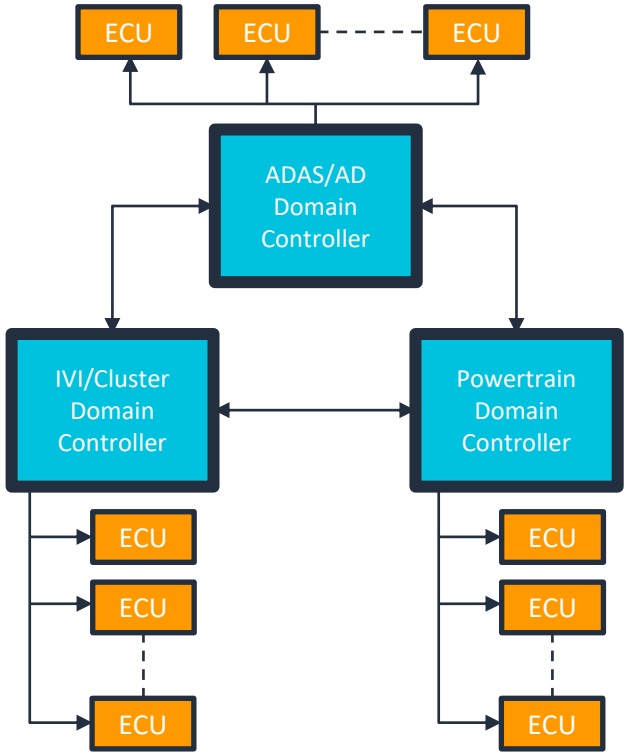
LINUX FOUNDATION

AUTOMOTIVE GRADE LINUX

In-vehicle hardware consolidation enables cloud native

Traditional Architecture

Domain Architecture



Zonal Architecture

The shift towards centralized architectures is leading to “Datacenter on Wheels” design paradigm paving the way to cloud native software in automotive

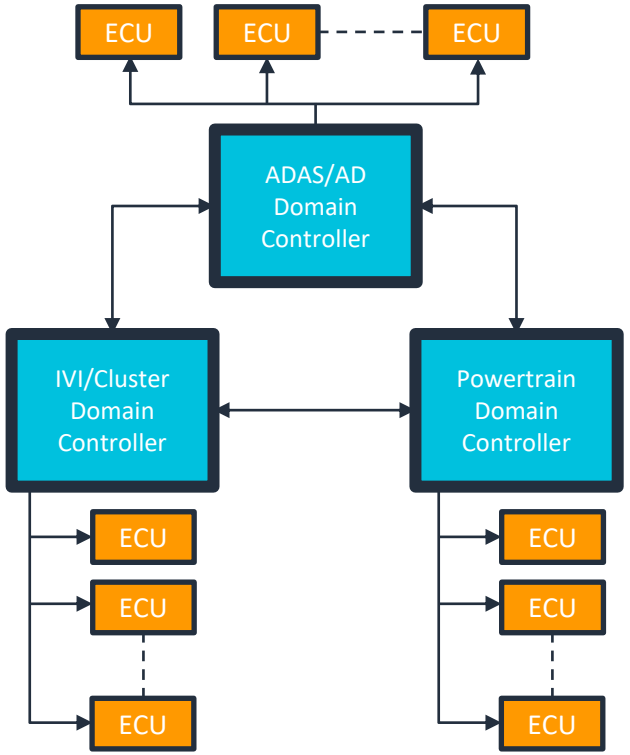
LINUX FOUNDATION

AUTOMOTIVE GRADE LINUX

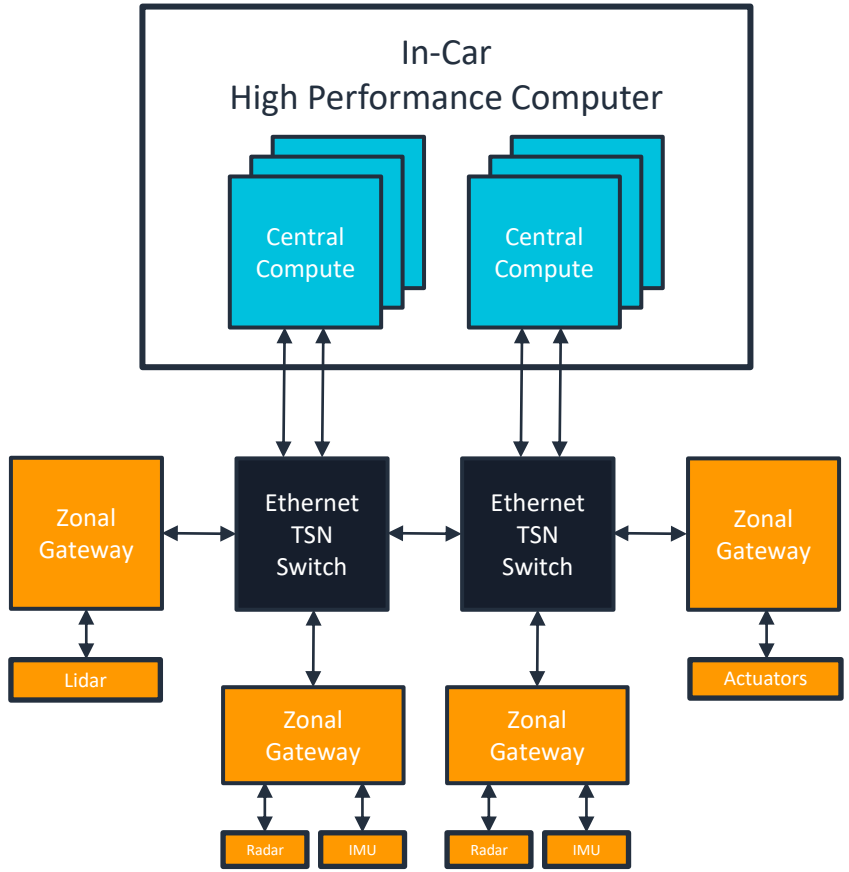
In-vehicle hardware consolidation enables cloud native

Traditional Architecture

Domain Architecture



Zonal Architecture



The shift towards centralized architectures is leading to “Datacenter on Wheels” design paradigm paving the way to cloud native software in automotive

LINUX FOUNDATION

AUTOMOTIVE GRADE LINUX

Cloud-Native Computing?

Cloud Native Computing Foundation (CNCF)

Linux Foundation project founded in 2015 to advance container technology and align the tech industry around its evolution

Launched with Kubernetes 1.0

Now covers key container and service mesh technologies

- **Scope of the Containers & Mesh Expert Group**

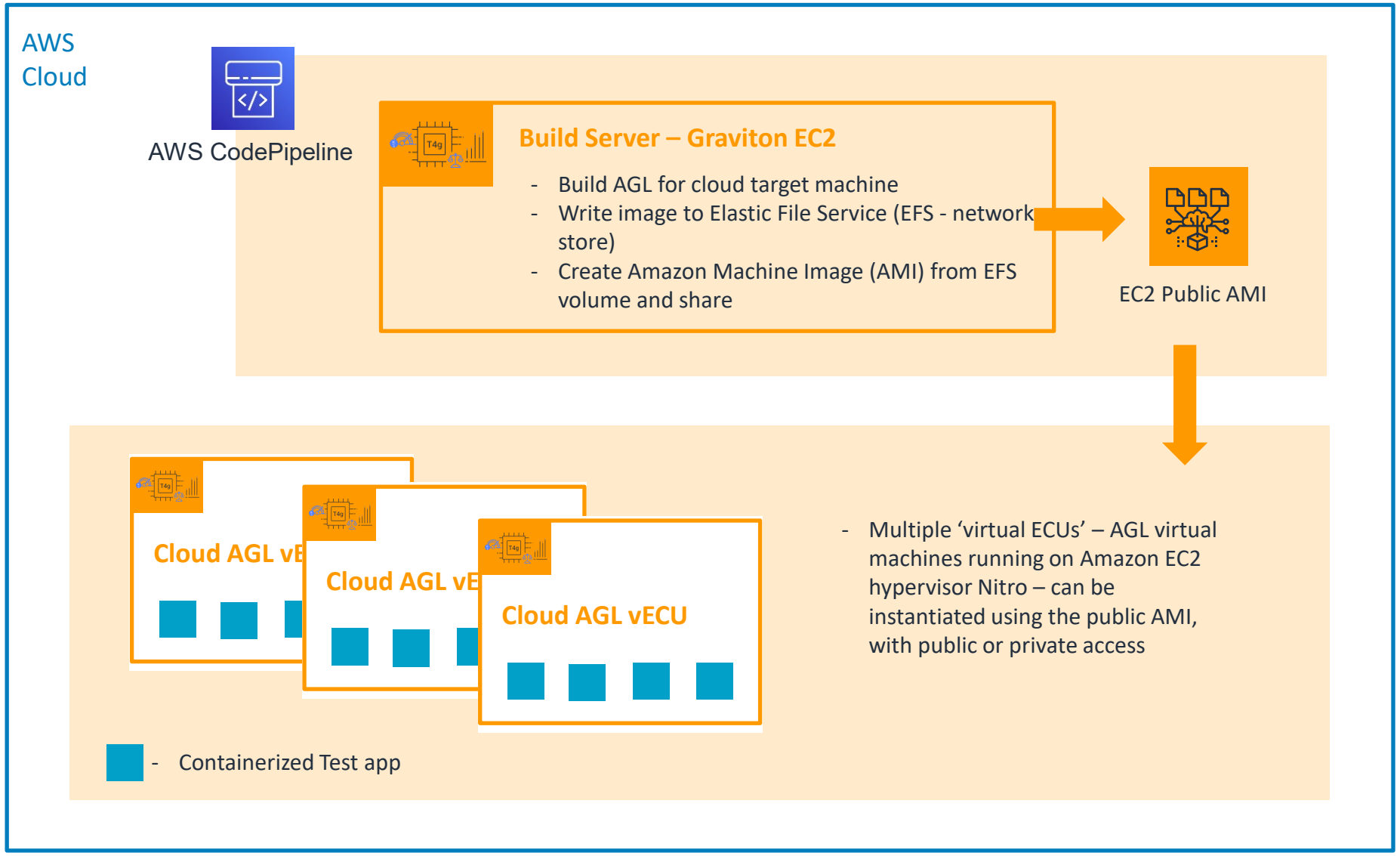
Container orchestration across single or multiple nodes inside and outside the vehicle

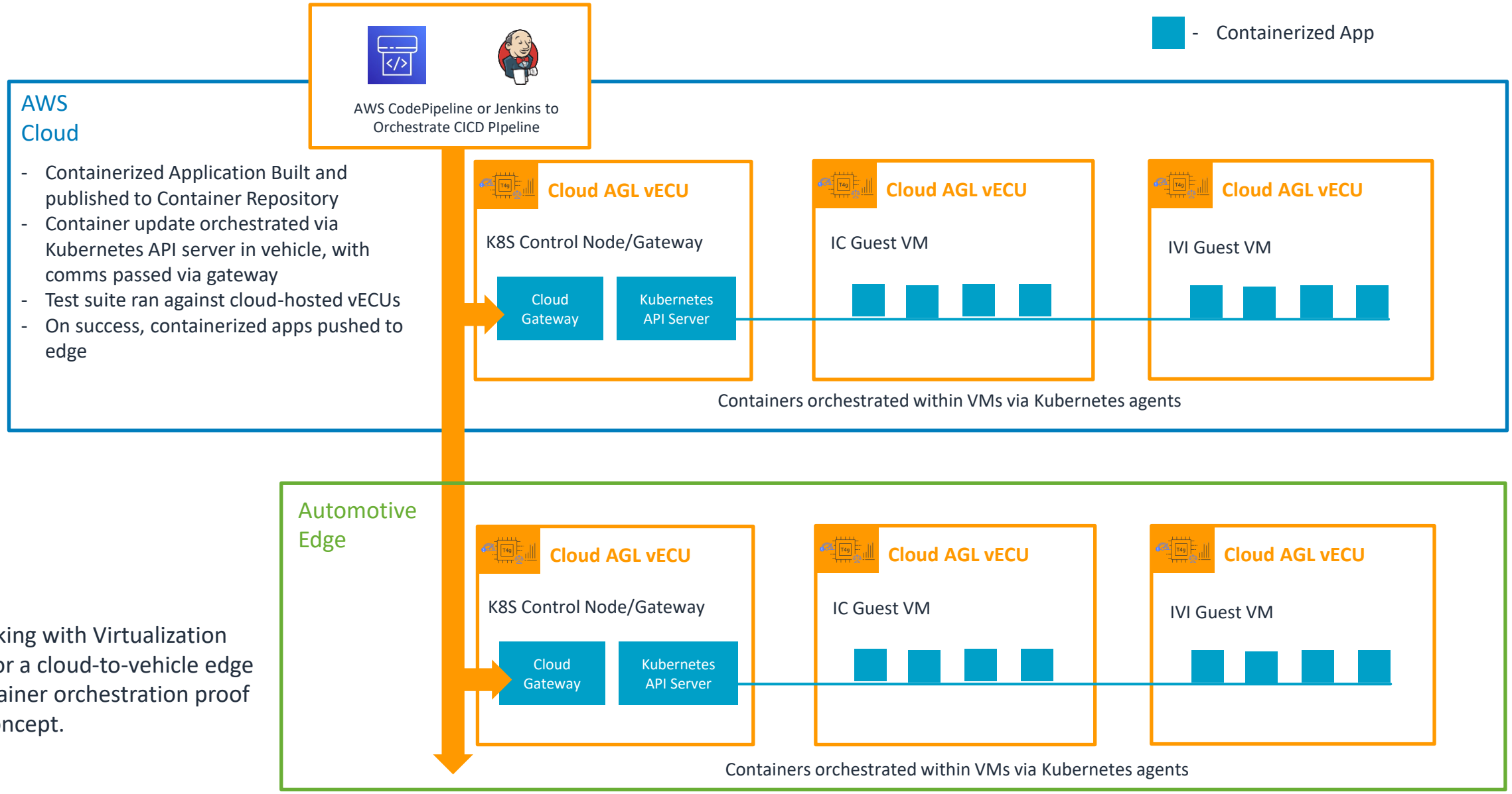
Application management: service deployment and update over-the-air

Application layer networking (eg service mesh) for container-based microservices

Collaborate closely with Virtualization EG, including bringing AGL to AWS EC2

Initial efforts ISO26262 QM only, for mixed-critically need to run containers on hypervisor


AWS
Cloud
AWS CodePipeline
Build Server – Graviton EC2
- Build AGL for cloud target machine
- Write image to Elastic File Service (EFS - network store)
- Create Amazon Machine Image (AMI) from EFS volume and share
EC2 Public AMI
Cloud AGL vE
Cloud AGL vE
Cloud AGL vECU
- Multiple ‘virtual ECUs’ – AGL virtual machines running on Amazon EC2 hypervisor Nitro – can be instantiated using the public AMI, with public or private access
- Containerized Test app



Working with Virtualization EG for a cloud-to-vehicle edge container orchestration proof of concept.

Thank you