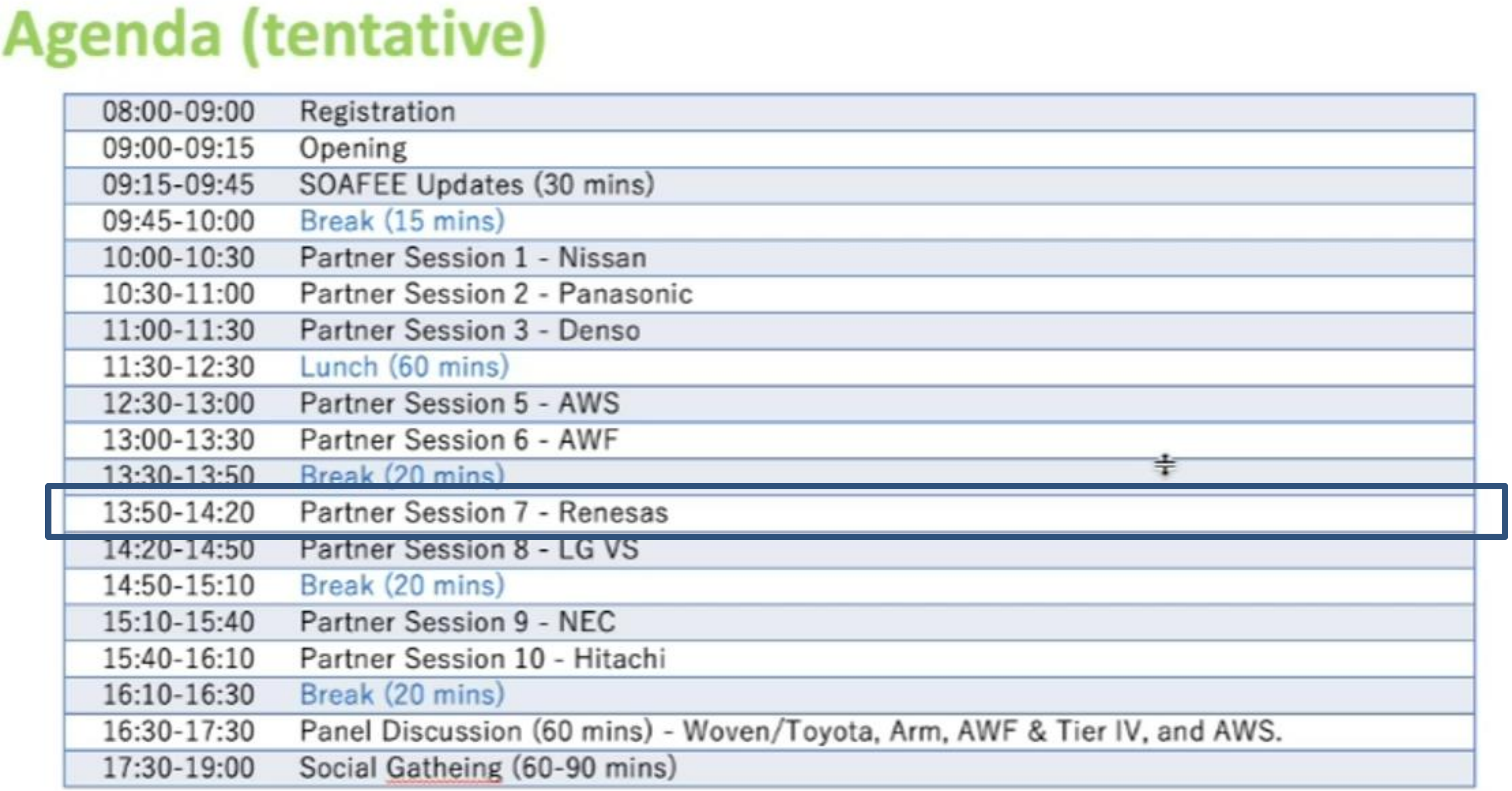
SOAFEE PROJECT

PROPOSAL OF BSP DRIVERS' STANDARDIZATION WITH ARM SYSTEMREADY

SEP.21, 2023
TUNG LUU
HPC SOFTWARE SOLUTION DIVISION
HIGH PERFORMANCE COMPUTING, ANALOG AND
POWER SOLUTIONS GROUP
RENESAS ELECTRONICS CORPORATION

SCHEDULE OF PRESENTATION



Agenda (tentative)

08:00-09:00	Registration
09:00-09:15	Opening
09:15-09:45	SOAFEE Updates (30 mins)
09:45-10:00	Break (15 mins)
10:00-10:30	Partner Session 1 - Nissan
10:30-11:00	Partner Session 2 - Panasonic
11:00-11:30	Partner Session 3 - Denso
11:30-12:30	Lunch (60 mins)
12:30-13:00	Partner Session 5 - AWS
13:00-13:30	Partner Session 6 - AWF
13:30-13:50	Break (20 mins)
13:50-14:20	Partner Session 7 - Renesas
14:20-14:50	Partner Session 8 - LG VS
14:50-15:10	Break (20 mins)
15:10-15:40	Partner Session 9 - NEC
15:40-16:10	Partner Session 10 - Hitachi
16:10-16:30	Break (20 mins)
16:30-17:30	Panel Discussion (60 mins) - Woven/Toyota, Arm, AWF & Tier IV, and AWS.
17:30-19:00	Social Gatheing (60-90 mins)

INTRODUCTION OF PRESENTER

Who am I ? Staff Engineer of Renesas Design.

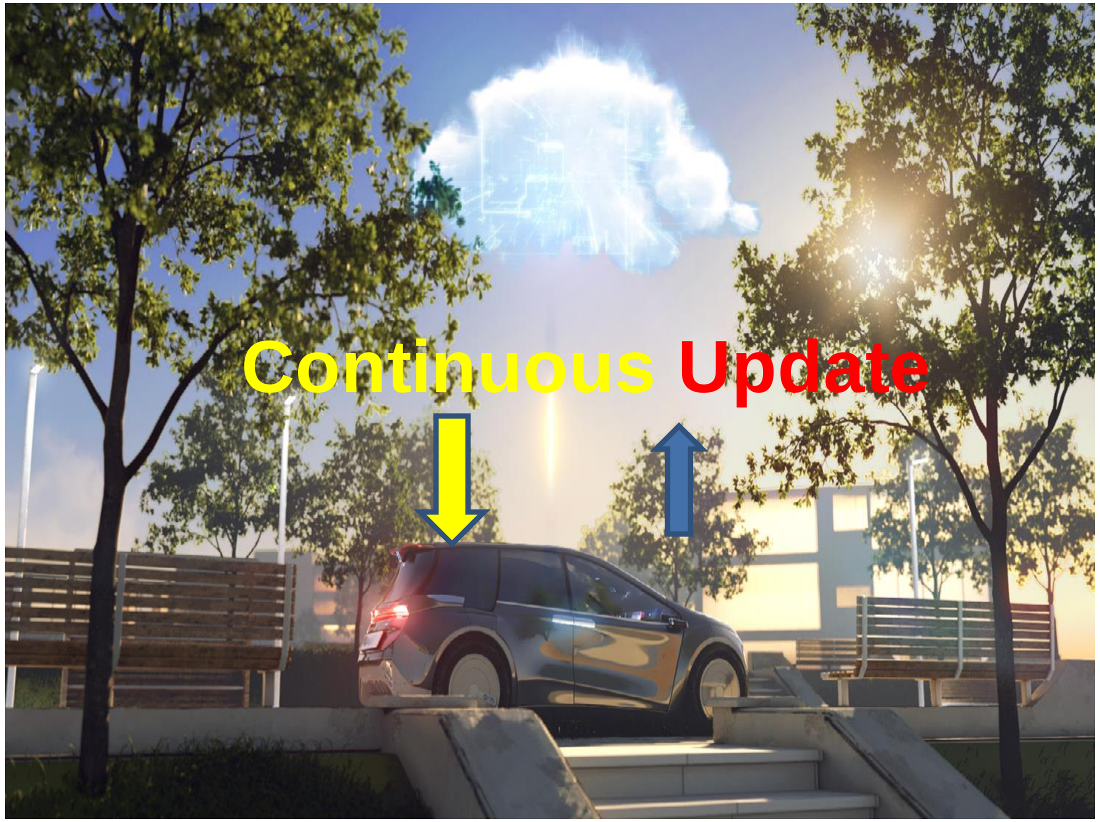
- Automotive E/E/ **BSP developer/ Integrator** of R-Car platform with Linux OS, INTEGRITY OS and Window Embedded Compact OS.
- Embedded Software Engineer/ Drivers porting in over 12-year experiences.
- Join Cloud Native with EWAOL enabling for SOAFEE and Edge orchestrator project feasibility checking for R-Car platforms – recently 1 year.
- Love footballs and traveling – I coach a team and got some achievements in local tournament.

AGENDA

▪ Background	**Page 05**
▪ Problem Statement	**Page 07**
▪ Past Approach	**Page 09**
▪ Future Approach with SOAFEE/Arm/Renesas	**Page 12**
▪ Roadmap	**Page 16**

BSP DRIVERS' STANDARDIZATION WITH ARM SYSTEMREADY

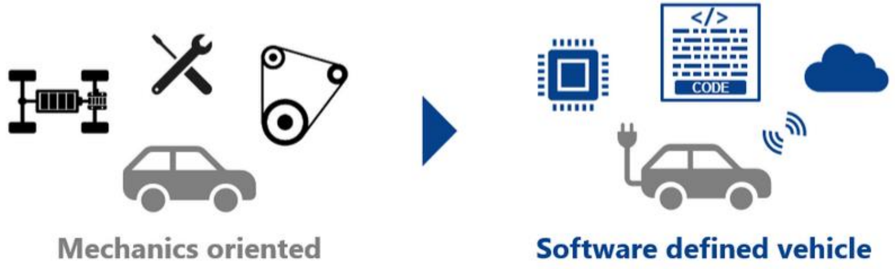
BACKGROUND - BUILDING THE ESSENTIAL VEHICLE INFRASTRUCTURE



Overview

Background – moved SDV era.

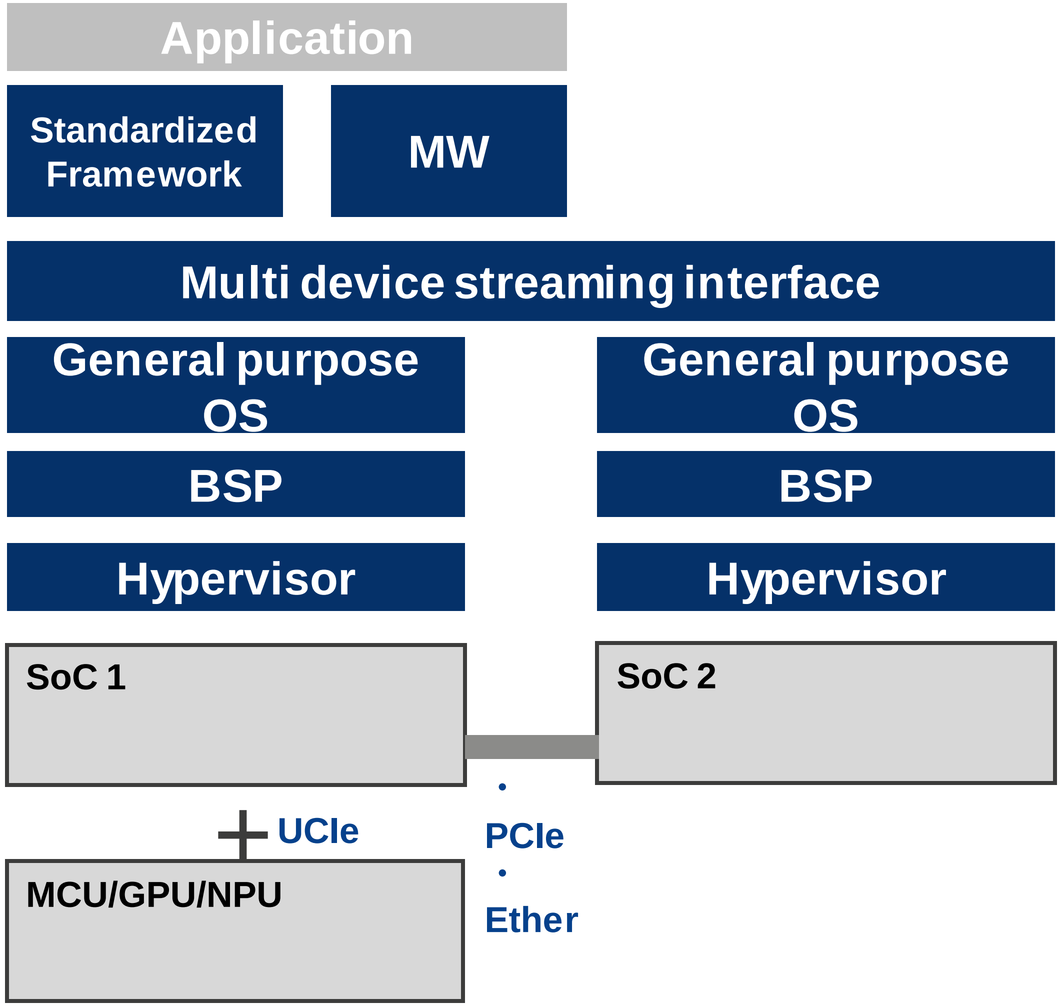
- Automotive E/E arch evolves into **HPC/Zonal** arch.
- Improvement of development efficiency is important to cope with complex SW development.
- HPC requires devices that can support multiple segments of ADAS/IVI/GW.
- Also, wide variety of supported OSs Linux, QNX, AOSP , QNX HV, Xen HV, COQOS HV, etc.
- Rapid updates with Cloud Native technology → SOAFEE elements.



Reference : CloudNative_to_SOAFEE_white_paper

BSP DRIVERS' STANDARDIZATION WITH ARM SYSTEMREADY

PROBLEM STATEMENT



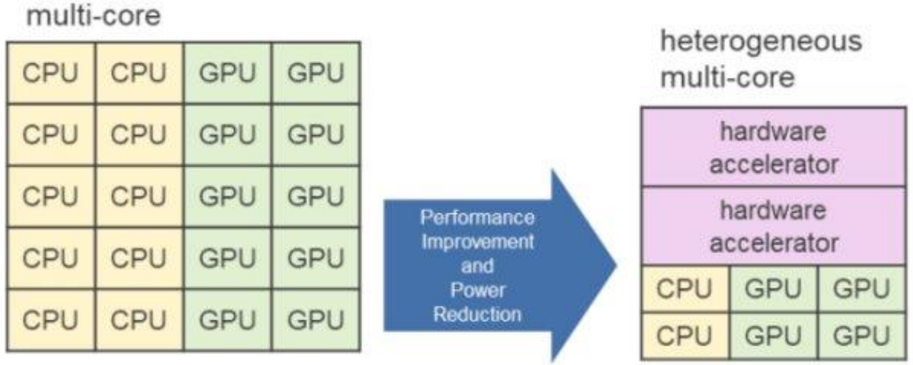
Restricted scalability due to complexity

Software Complexity

- Transition towards "Software Defined"
  - More capabilities express in software
  - Volume of code in production vehicles going through exponential growth
- Software tied to "BSP" through non-standard interfaces and characteristics
- Software is not portable
  - Requirement huge amount of re-integration
  - Some OEM's prefer to start again
- Unstable to amortize software cost over multiple production runs

BSP DRIVERS' STANDARDIZATION WITH ARM SYSTEMREADY

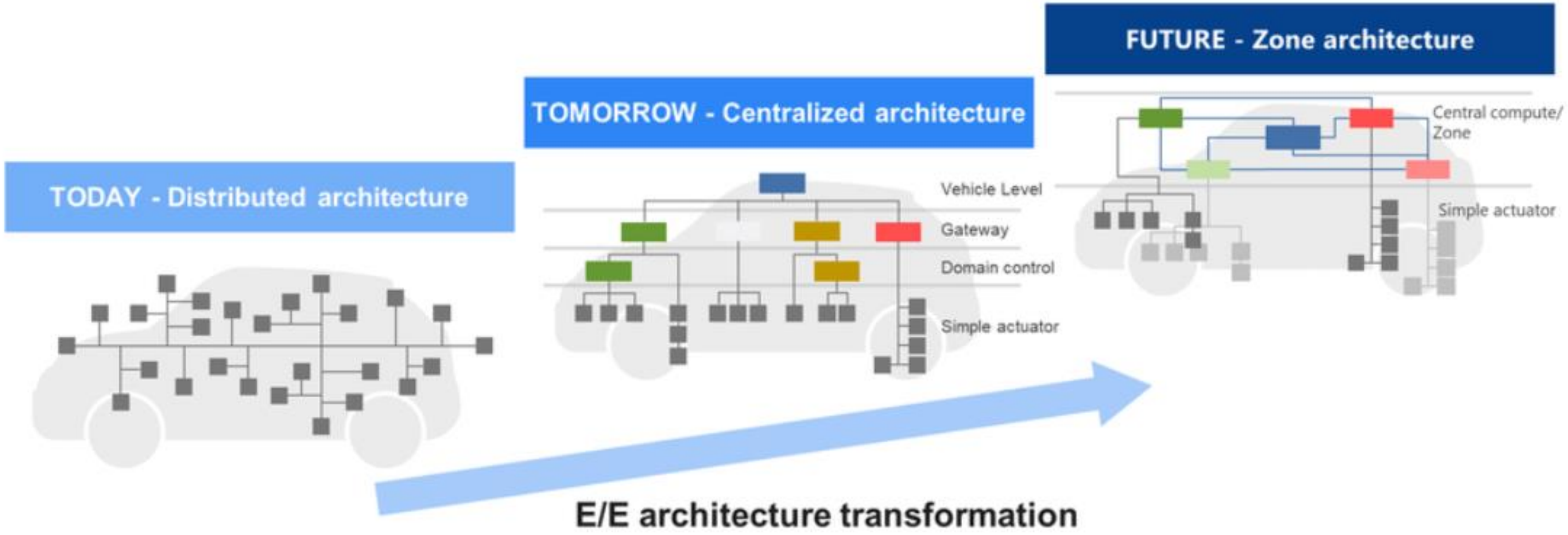
PROBLEM STATEMENT (2/2)



Restricted scalability due to complexity

Hardware Complexity

- Heterogeneous SoC's becoming increasingly complex
  - CPU/MCU/GPU/NPU/…
- Multiple compute islands within the system that have different execution characteristics.
- Difficult to the software and developers to make efficient use of all hardware available.
- Software becomes tied to a specific hardware architecture and system topography.


FUTURE - Zone architecture
TOMORROW - Centralized architecture
TODAY - Distributed architecture
Central compute/ Zone
Simple actuator
Vehicle Level
Gateway
Domain control
Simple actuator
E/E architecture transformation

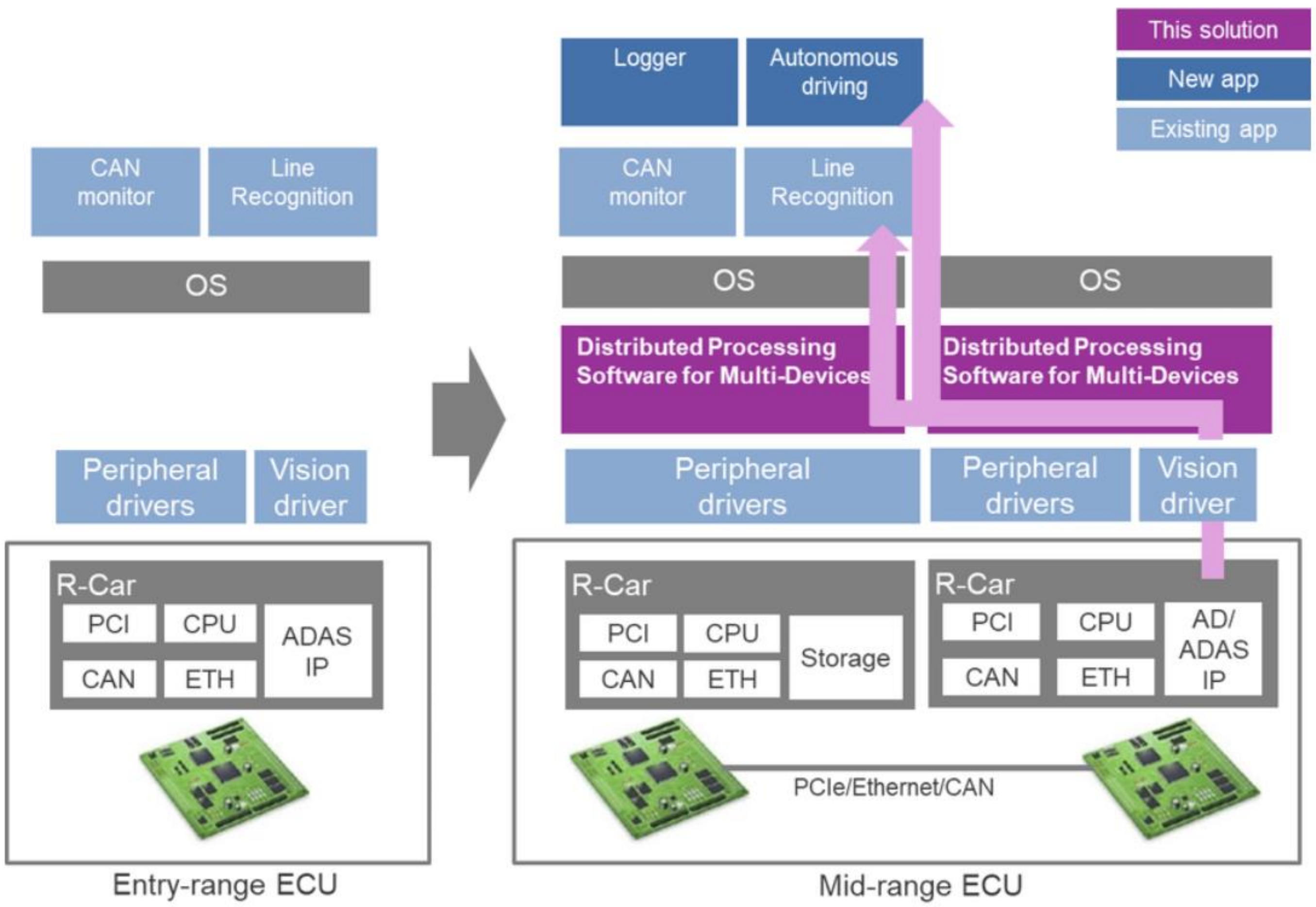
Conclusion :

We've got multiple **compute domains** within SoC. **Multiple SoCs and MCUs** will be integrated into a **single ECU**.

So that the complexity of SW is now most definitely set by HW definitions / HW updates / HW modifications because there's difficult way that my SW can run with new HW in the term of "**scalability**" in short time to market.

BSP DRIVERS' STANDARDIZATION WITH ARM SYSTEMREADY

PROBLEM STATEMENT (1/2)



Restricted scalability due to complexity

- ○ Different in devices specifications are absorbed by drivers, driver programming models differ from OS to OS
- ○ In order to absorb differences in device specifications and to accommodate differences in driver programming models

Or different operating systems, semiconductor vendor driver developers need to continue porting and developing drivers repeatedly

- ○ This work is not inherently productive, but it is necessary and wastes labor (cost)

When the upgrade happens in both SW and HW, the shorten time to market, the saving for the cost are development targets.

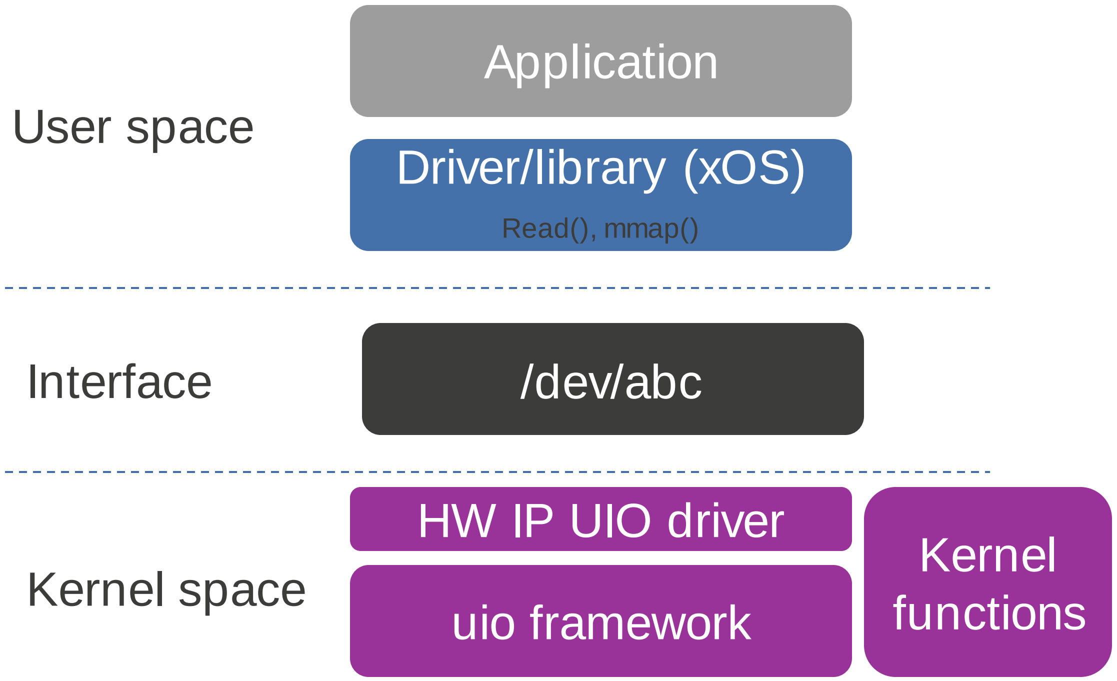
BSP DRIVERS' STANDARDIZATION WITH ARM SYSTEMREADY

PAST APPROACHES (1/3)

Establish some of methodologies or infrastructures called “Common Platform”

○ UIO : User space I/O which separated system into User and Kernel spaces.

Generalized UIO approach for all related HW



User space

Application

Driver/library (xOS)

Read(), mmap()

Interface

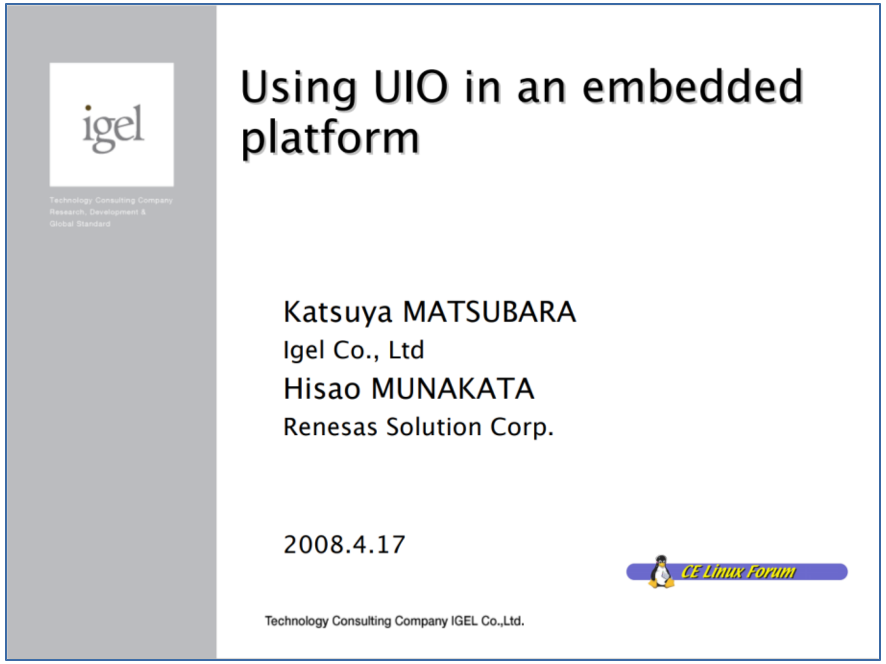
/dev/abc

Kernel space

HW IP UIO driver

uio framework

Kernel functions



igel

Technology Consulting Company
Research, Development &
Global Standard

Using UIO in an embedded platform

Katsuya MATSUBARA
Igel Co., Ltd
Hisao MUNAKATA
Renesas Solution Corp.

2008.4.17

Technology Consulting Company IGEL Co.,Ltd.

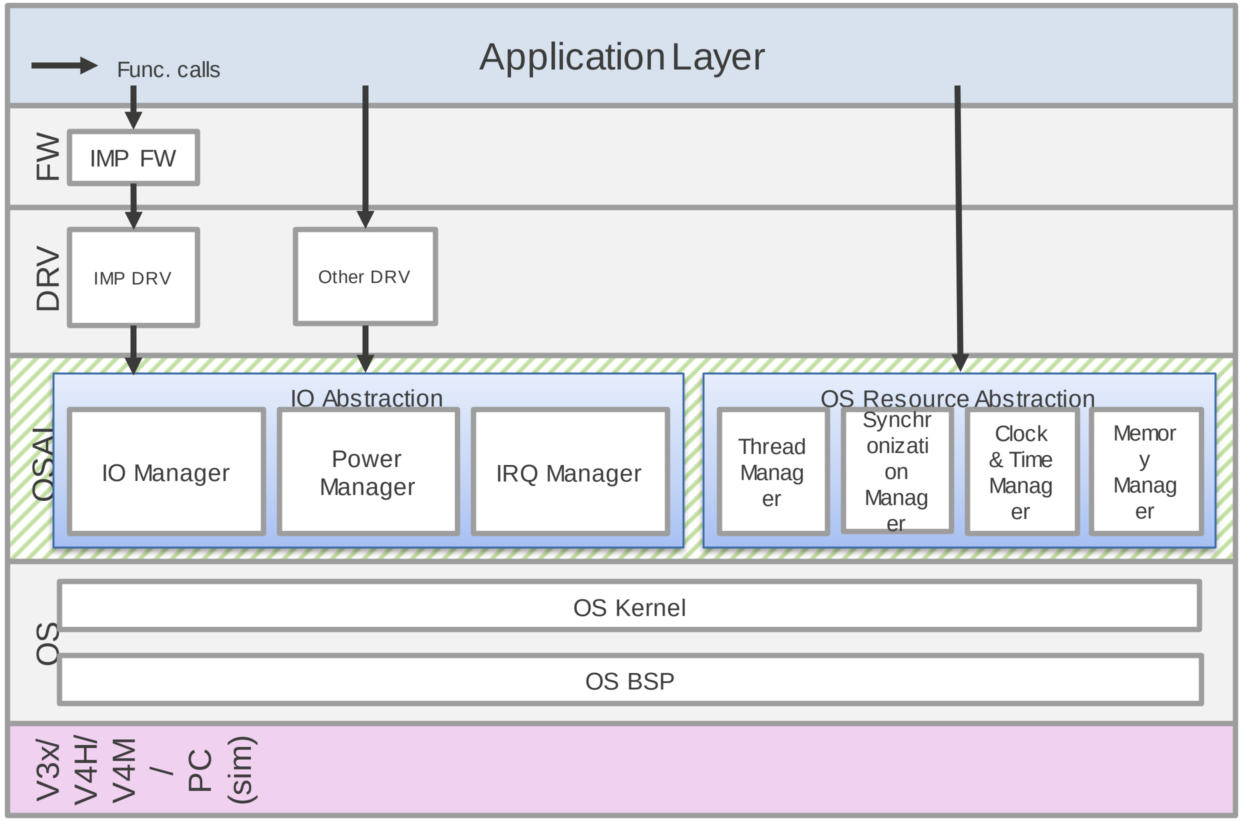
https://elinux.org/images/b/b0/Uio080417celfelc08.pdf

BSP DRIVERS' STANDARDIZATION WITH ARM SYSTEMREADY

PAST APPROACH (2/3)

Establish some of methodologies or infrastructures called “Common Platform”

○ OSAL : OS Abstraction Layer developed the concept of user-mode drivers to improve the portability among Operating System



OSAL consists of 2 function category.

1. OS Resource Abstraction
   Provide OS resource control functions. It’s similar to POSIX APIs.
2. IO Abstraction
   Provide I/O device control functions. This block is abstract running the Target HW/ISS environments.

If application is used OSAL, application layer can be portable for several OS and server target devices.

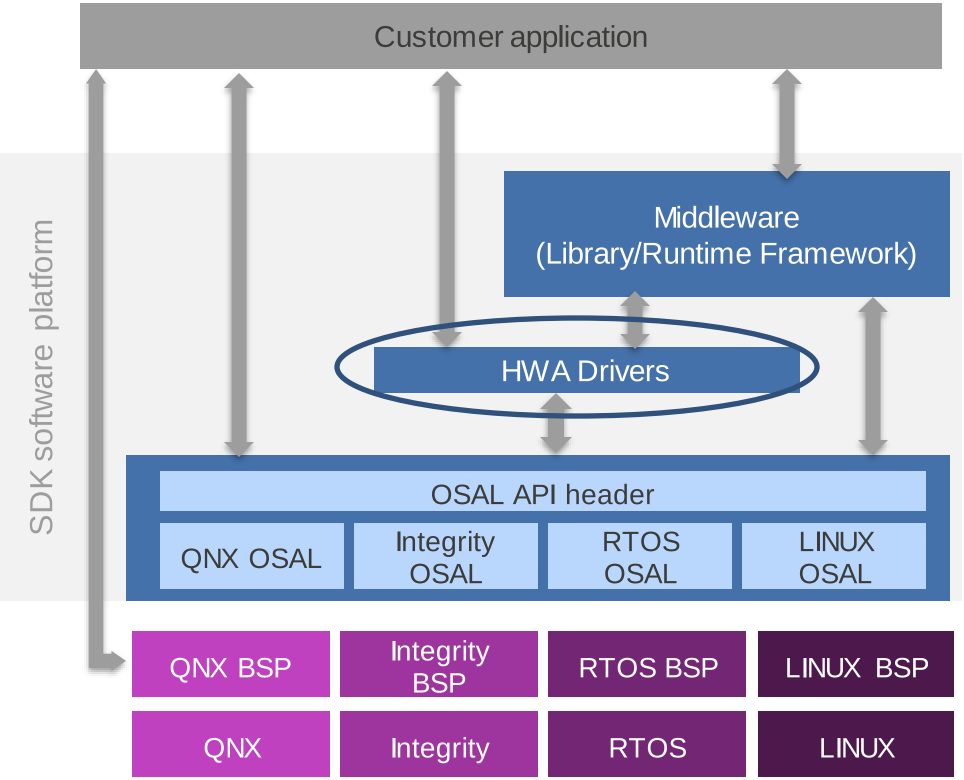
But, application shall handle HWA sharable memory using OSAL Memory Manager, and Power policy as HP (High Performance), CG (Clock Gating), PG (Power Gating)

BSP DRIVERS' STANDARDIZATION WITH ARM SYSTEMREADY

PAST APPROACH (3/3)

Establish some of methodologies or infrastructures called "Common Platform"

- HAL Hardware Abstraction Layer : concept of commonization drivers by having layers in BSP drivers.
  HWA : Hardware Acceleration



Renesas introduces the R-Car SDK platform for the ADAS R-Car SoCs from V3x onwards.

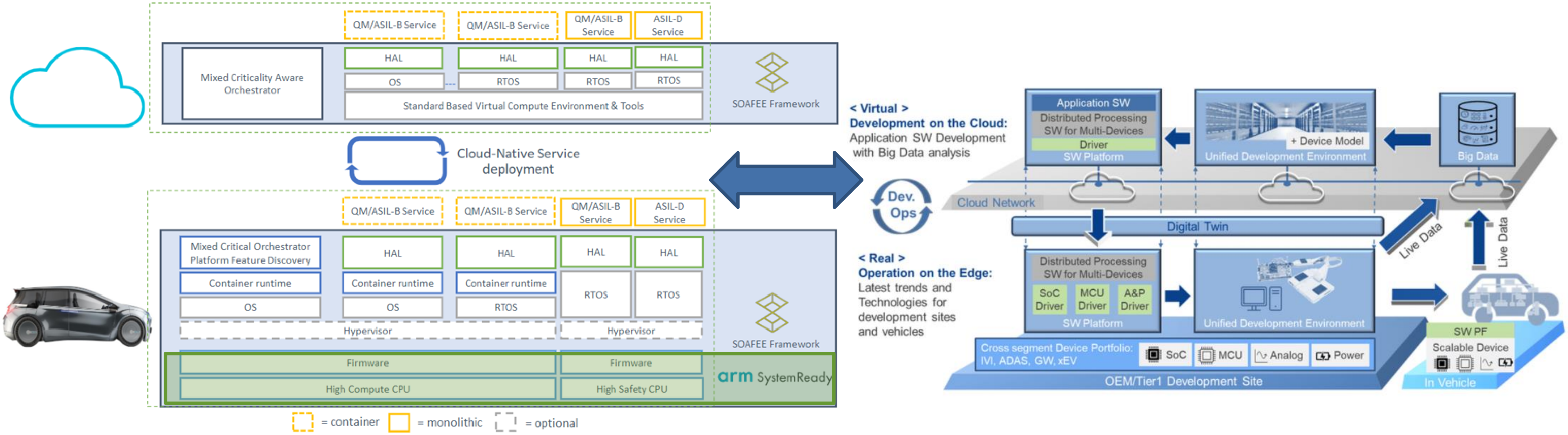
- An OS abstraction layer (OSAL)
  - Shall support RTOS, QNX, Linux
- OS and device independent HWA drivers and middleware:
  - ISP, IMR, VIP, IMP-X (incl CNN)…
- Middleware services :
  - IMP Framework, HWA Memory manager, etc…
- Tools related to the HWA drivers:
  - E.g.: compilers, simulator, debug tools SILS/HILS

Starting from this solid SDK1 SW platform basis, Renesas focusing resources on feature increase (M/W extension) starting from SDK2 in 2021.

Shorten time to market switching to new device/OS & Focusing on value added components

BSP DRIVERS' STANDARDIZATION WITH ARM SYSTEMREADY

FUTURE APPROACH WITH SOAFEE/ARM/RENESAS (1/4)

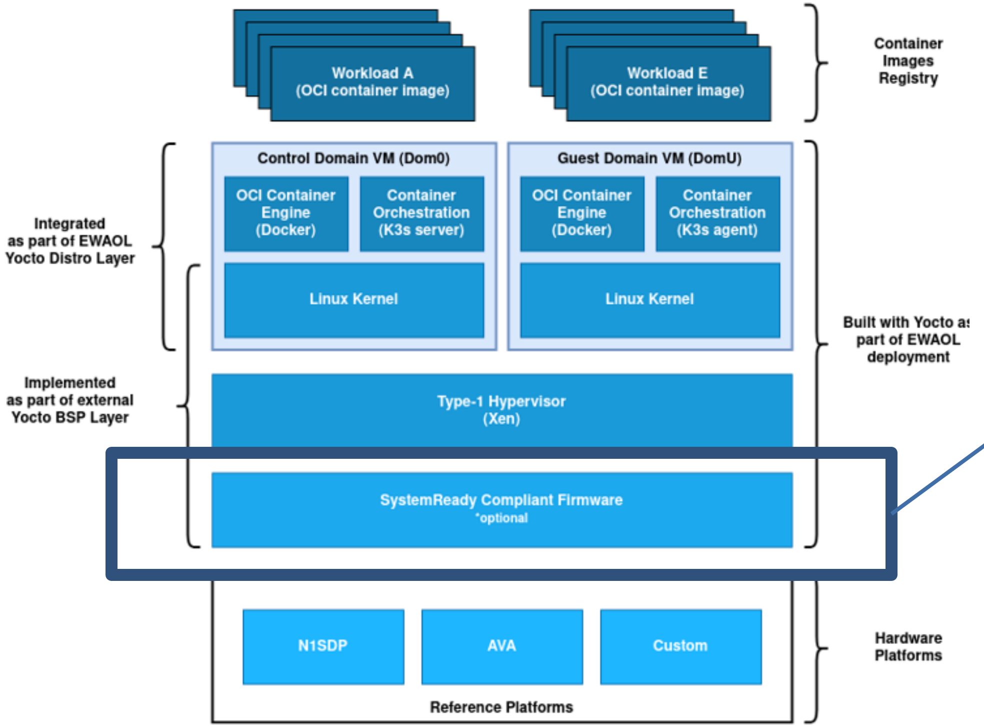


Definitions

- Formula of SCALABILITY is Cloud Native (DevOps/Continuous Delivery/Microservices/Containers) + Functional Safety/ Real-time/Heterogeneous Compute/Accelerator Aware/IO Aware … **BECOME SOAFEE**

BSP DRIVERS' STANDARDIZATION WITH ARM SYSTEMREADY

FUTURE APPROACH WITH SOAFEE/ARM SYSTEMREADY/RENESAS(2/4)



Approaches

There are approaches when Arm SystemReady **WOULD BE** applied to Renesas OSAL/HWA components as solutions to standardize parts of the current architectures.

Arm SystemReady Certification should be satisfied to development system

Standard enabled by Arm SystemReady, PSA, …

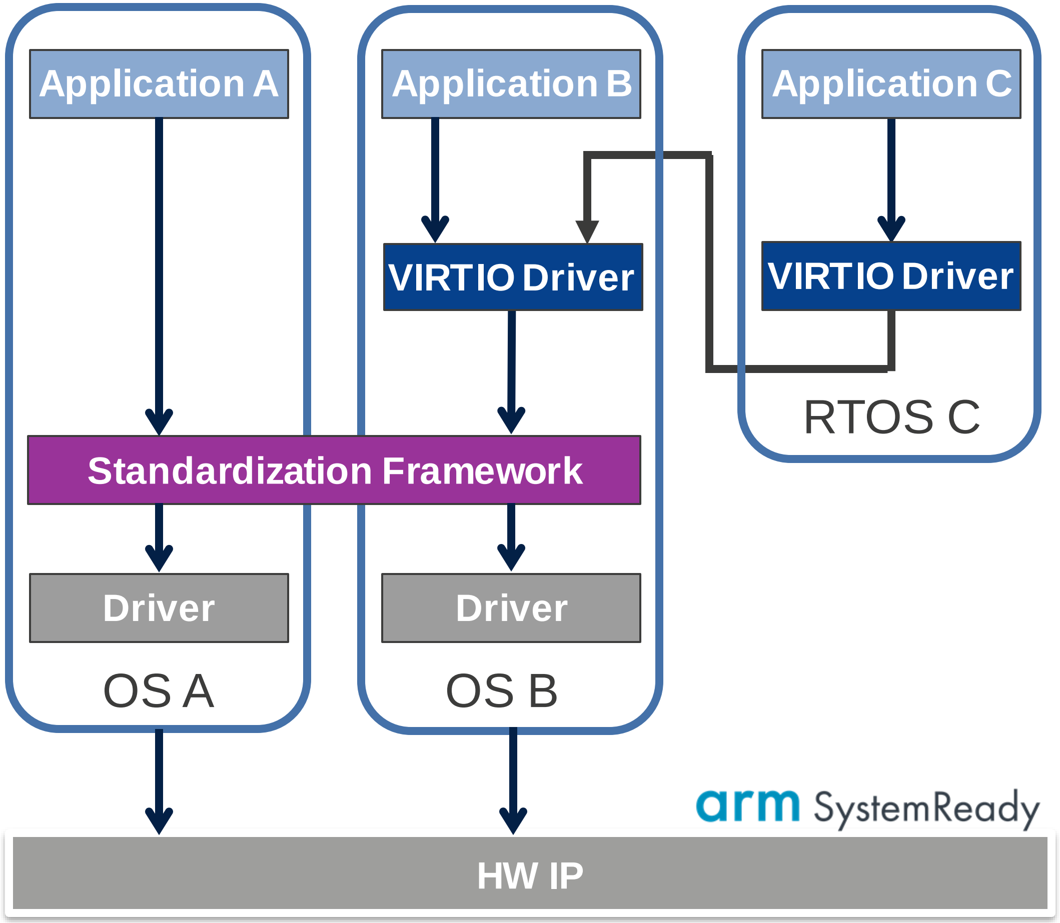
For low-level software, we incorporate standards built on commonly-accepted industry approaches, such as **SystemReady and PSA Certified.** as we started confirmation with **Renesas RZ/G2M Awarded Arm SystemReady IR 1.1 Certification . Now continue with R-Car + SOAFEE also another types of development** Reference : four pillars software-defined-vehicles

BSP DRIVERS' STANDARDIZATION WITH ARM SYSTEMREADY

FUTURE APPROACH WITH SOAFEE/ARM SYSTEMREADY/RENESAS(3/4)

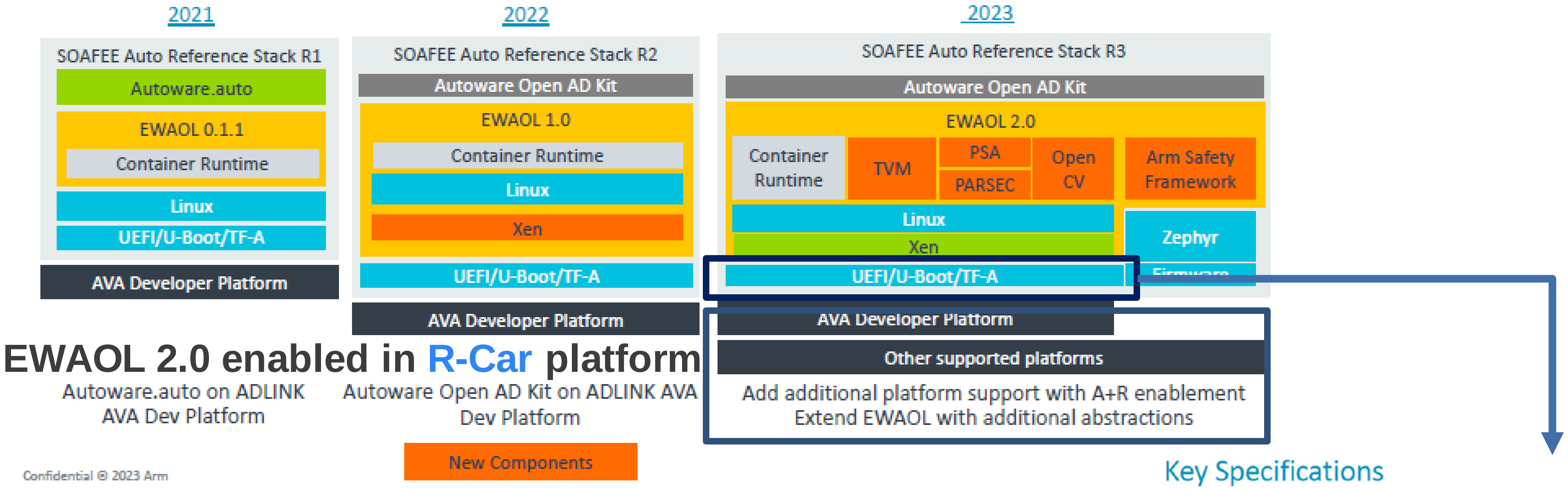
Approaches

Standardize OS-side drivers by having the OS call the Arm SystemReady driver I/F in vs. OS (like Virt I/O)



BSP DRIVERS' STANDARDIZATION WITH ARM SYSTEMREADY

FUTURE APPROACH WITH SOAFEE/ARM SYSTEMREADY/RENESAS(4/4)



2021

SOAFEE Auto Reference Stack R1

Autoware.auto

EWAOL 0.1.1

Container Runtime

Linux

UEFI/U-Boot/TF-A

AVA Developer Platform

2022

SOAFEE Auto Reference Stack R2

Autoware Open AD Kit

EWAOL 1.0

Container Runtime

Linux

Xen

UEFI/U-Boot/TF-A

AVA Developer Platform

2023

SOAFEE Auto Reference Stack R3

Autoware Open AD Kit

EWAOL 2.0

Container Runtime

TVM

PSA

PARSEC

Open CV

Arm Safety Framework

Linux

Xen

Zephyr

UEFI/U-Boot/TF-A

Firmware

AVA Developer Platform

Other supported platforms

Add additional platform support with A+R enablement
Extend EWAOL with additional abstractions

EWAOL 2.0 enabled in R-Car platform

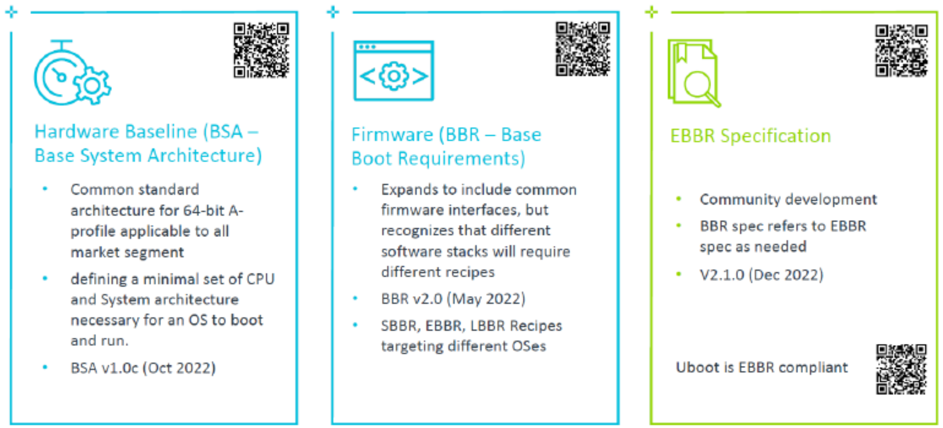
Autoware.auto on ADLINK AVA Dev Platform

Autoware Open AD Kit on ADLINK AVA Dev Platform

New Components

Confidential © 2023 Arm

Key Specifications



Hardware Baseline (BSA – Base System Architecture)

- Common standard architecture for 64-bit A-profile applicable to all market segment
- defining a minimal set of CPU and System architecture necessary for an OS to boot and run.
- BSA v1.0c (Oct 2022)

Firmware (BBR – Base Boot Requirements)

- Expands to include common firmware interfaces, but recognizes that different software stacks will require different recipes
- BBR v2.0 (May 2022)
- SBBR, EBBR, LBBR Recipes targeting different OSes

EBBR Specification

- Community development
- BBR spec refers to EBBR spec as needed
- V2.1.0 (Dec 2022)

Uboot is EBBR compliant

Arm SystemReady includes components :

- Base system architecture BSA which describes how the SoC the device is being build.
- Then Base boot requirement called BBR specification that describes how the system SW the firmware is being integrated with the HW to offer a complete solution so that developers can develop for the application.
- ACS Suite the Architectural Compliance Suite that test the whole system to make sure it’s compliant and meets the specification.

Save costs : Removing the customizations, the fine tuning, the methods, the verifications, and managing multiple binaries for multiple platform – these are huge cost savings for ODMs and OEMs.

BSP DRIVERS' STANDARDIZATION WITH ARM SYSTEMREADY

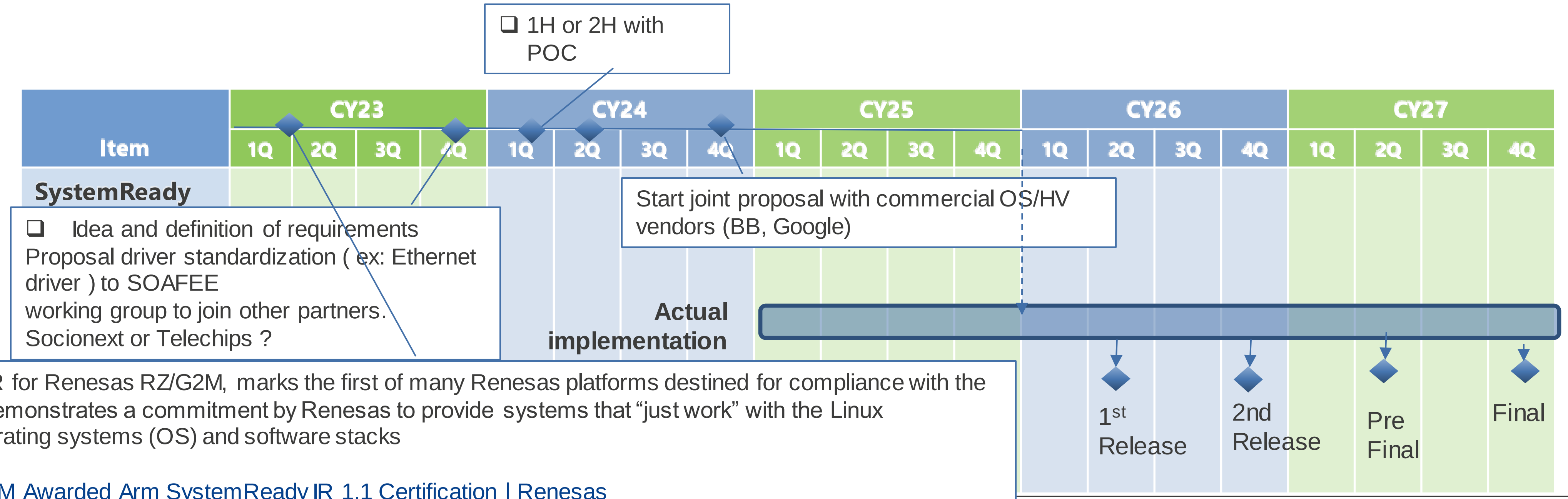
ROADMAP

Procedures

○ Renesas will be working with Arm and others on this kind of activity.

That roadmap needs to be discussed with Arm + Renesas Executives.

- ❑ During 2023: Idea and definition of requirements
- ❑ During 2024: POC (Proof of concepts or demonstrations)
- ❑ During 2025: Development implementation of production
- ❑ During 2026-2027: Actual release to vehicle production.



❑ 1H or 2H with POC

❑ Idea and definition of requirements
Proposal driver standardization (ex: Ethernet driver) to SOAFEE
working group to join other partners.
Socionext or Telechips ?

Start joint proposal with commercial OS/HV vendors (BB, Google)

SystemReady IR for Renesas RZ/G2M, marks the first of many Renesas platforms destined for compliance with the Arm program, demonstrates a commitment by Renesas to provide systems that “just work” with the Linux ecosystem, operating systems (OS) and software stacks

Renesas RZ/G2M Awarded Arm SystemReady IR 1.1 Certification | Renesas

BSP DRIVERS' STANDARDIZATION WITH ARM SYSTEMREADY

DISCUSSION

Thank you for your attention !!!

Q&A session

APPENDIX

KEYWORDS AND REFERENCE

- SOAFEE(Blog, Join and etc)
- SOAFEE Architecture Spec 1.0
- EWAOL Documentation
- SOAFEE Virtual
- SOAFEE Architecture
- Arm DEVSUMMIT
- Arm SystemReady Ceritification program
- **R-Car Renesas System On Chips (SOCs)**
- **Renesas Automotive Business Strategy**
- **Renesas Strategy for Automotive Software**
- SystemReady IR IoT Integration, Test, and Certification Guide
- SystemReady Security Interface Extension User Guide
- SystemReady IR website
- SystemReady Requirements specification (SRS)
- Base System Architecture (BSA)
- Base Boot Requirements (BBR)
- Embedded Base Boot Requirements (EBBR)

Renesas.com