Software Defined Vehicles and the need for Standardization

Andrea Gallo
VP of Business Development

AUTOMOTIVE SPIN ITALIA

Linaro

Agenda

- Software Defined Everything
- Implications from the Software Defined Vehicle (SDV) Revolution
- Linaro Open Source Projects to enable SDV

1870s and 1940s (electro)mechanical calculators

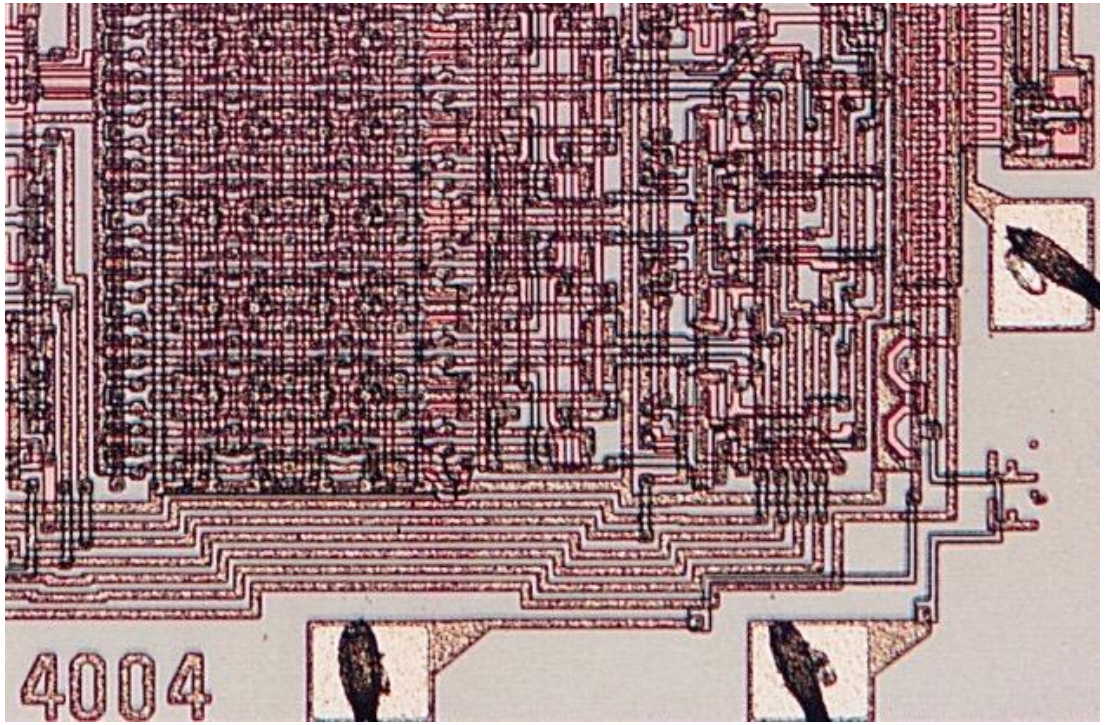
https://veroantico.altervista.org/prodotto/antica-calcolatrice-meccanica-thales/

https://it.wikipedia.org/wiki/Olivetti_Divisumma_14

1964 the Olivetti P101 – Pier Giorgio Perotto

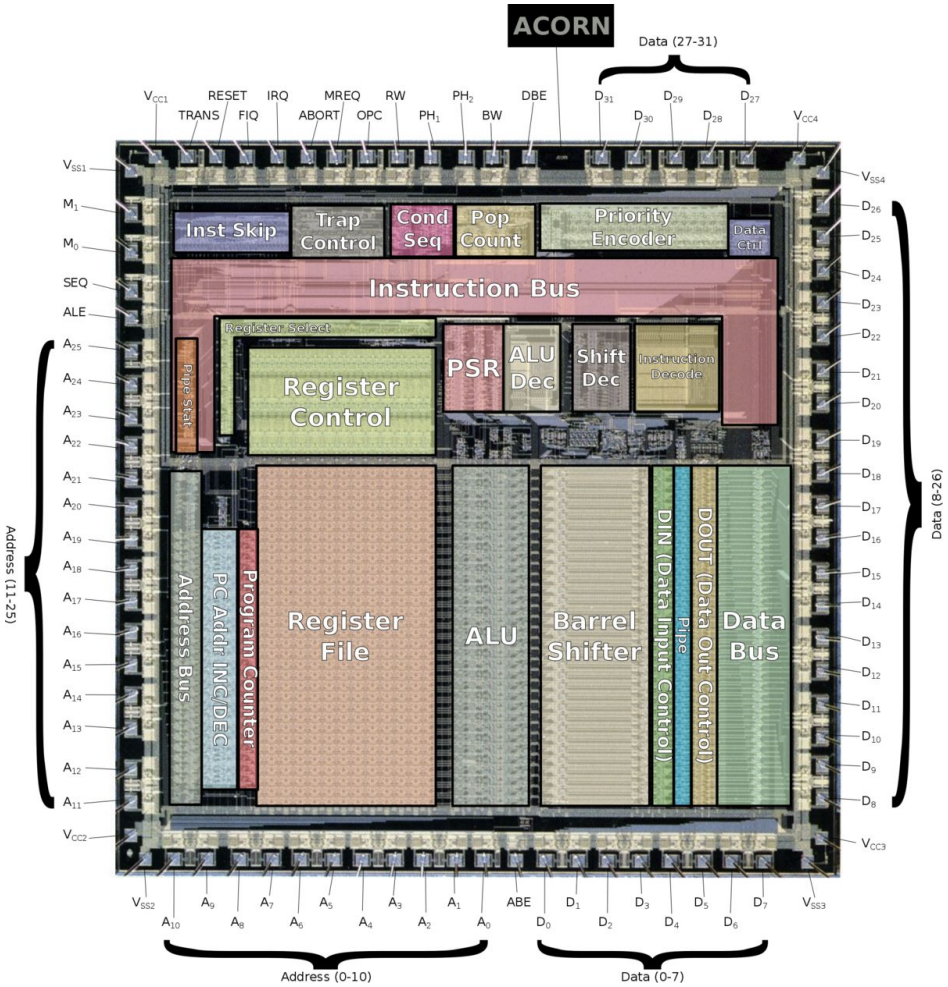
https://upload.wikimedia.org/wikipedia/commons/thumb/5/5e/Olivetti_Programma_101_-_Museo_scienza_e_tecnologia_Milano.jpg/390px-Olivetti_Programma_101_-_Museo_scienza_e_tecnologia_Milano.jpg

1971 Intel 4004 – Federico Faggin



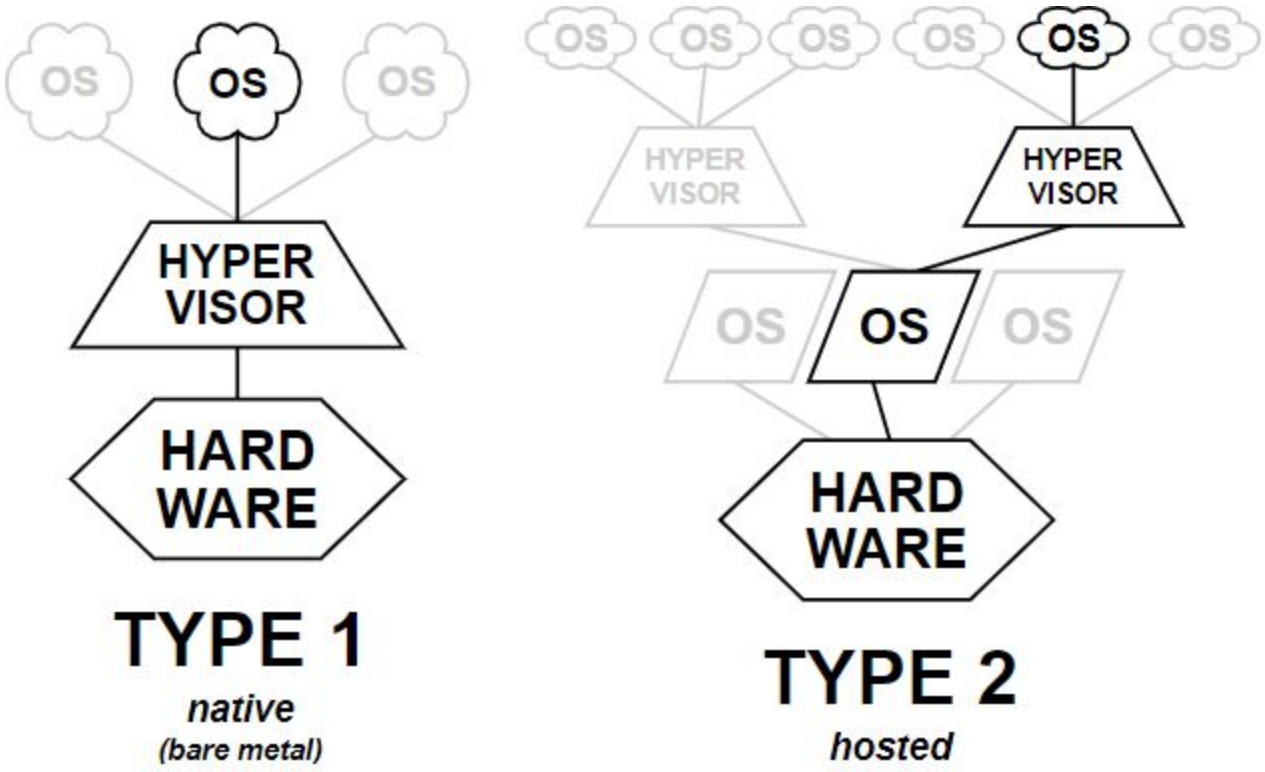
http://www.intel4004.com/

1985 Acorn ARM1 processor – Steve Furber & Sophie Wilson



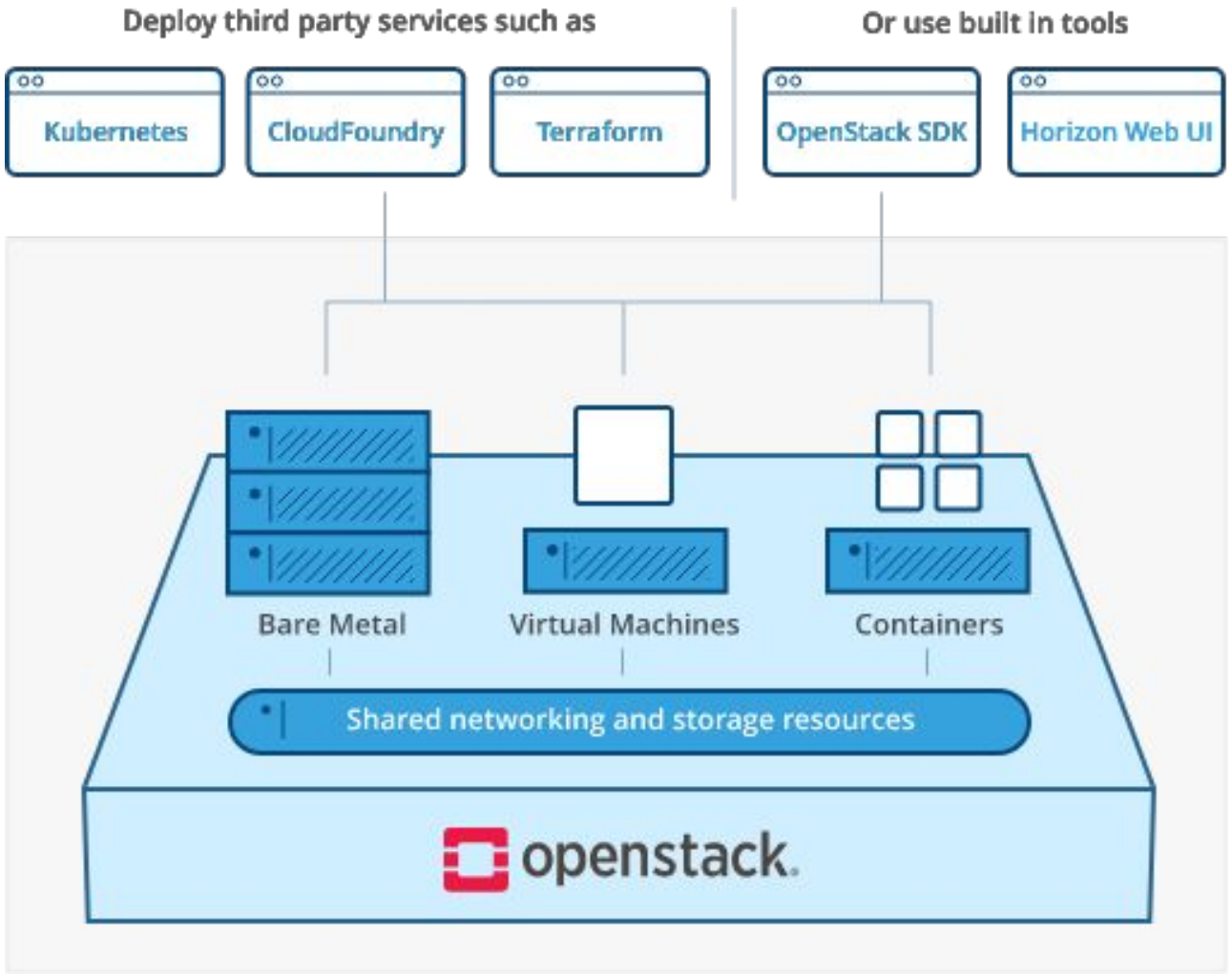
https://en.wikichip.org/w/images/thumb/5/5f/arm1_die_shot_%28annotated%29.png/1200px-arm1_die_shot_%28annotated%29.png

1974 Virtualization – Gerald J. Popek and Robert P. Goldberg



https://en.wikipedia.org/wiki/Hypervisor#/media/File:Hyperviseur.svg

2010 OpenStack initial release



https://www.openstack.org/

Software Defined Everything

NFV

Networking

Cloud

Edge

Virtualization

Hypervisor

IaaS

Infrastructure

SaaS

PaaS

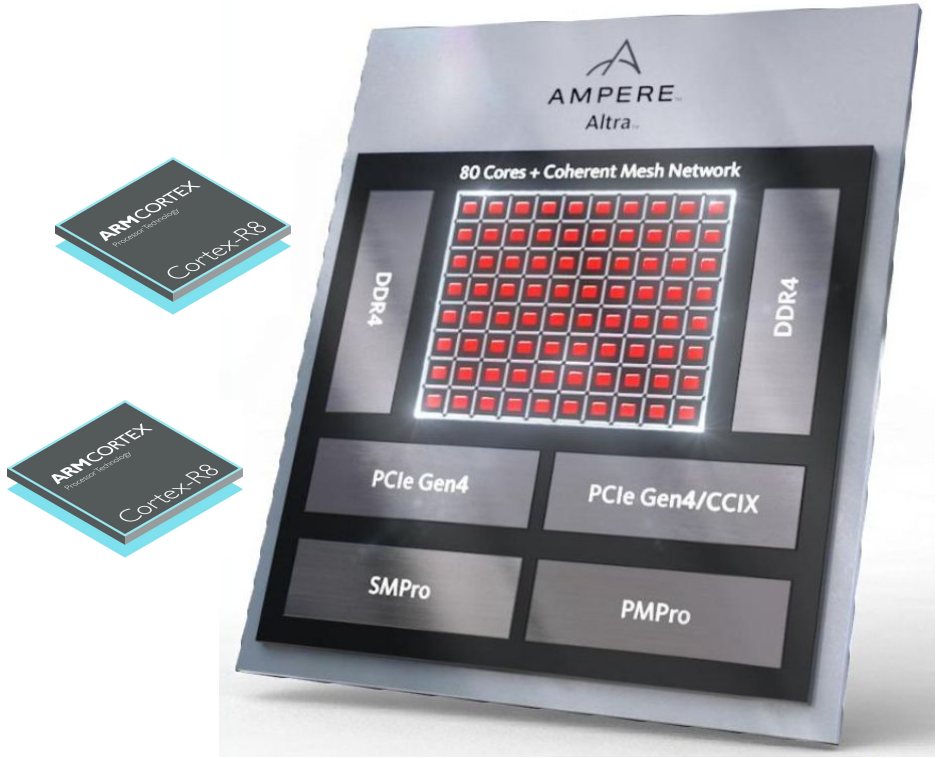
Platform

SDS

Service

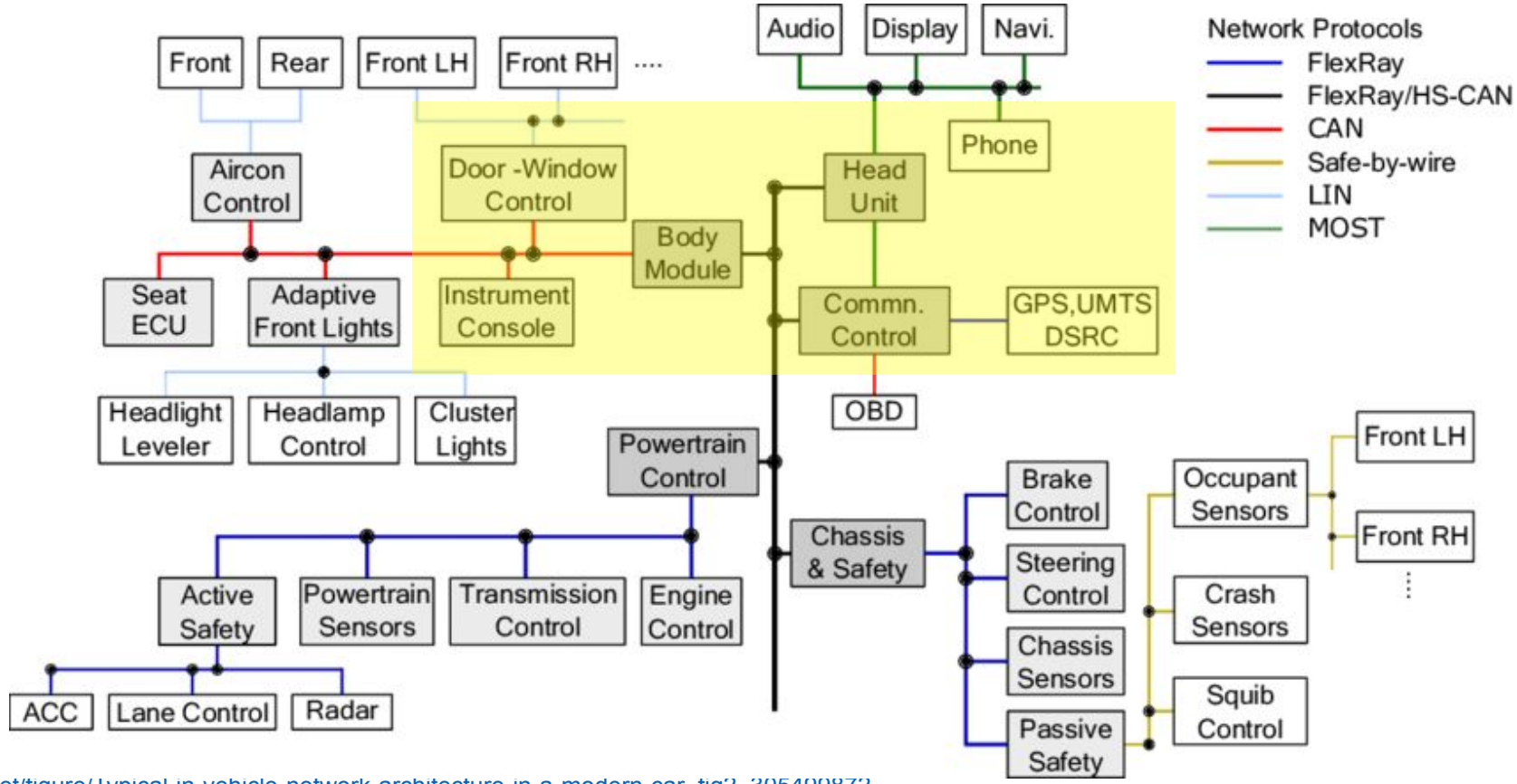
Software Defined Vehicles

From hundreds of ECUs to one Automotive Server



Software Defined Vehicles

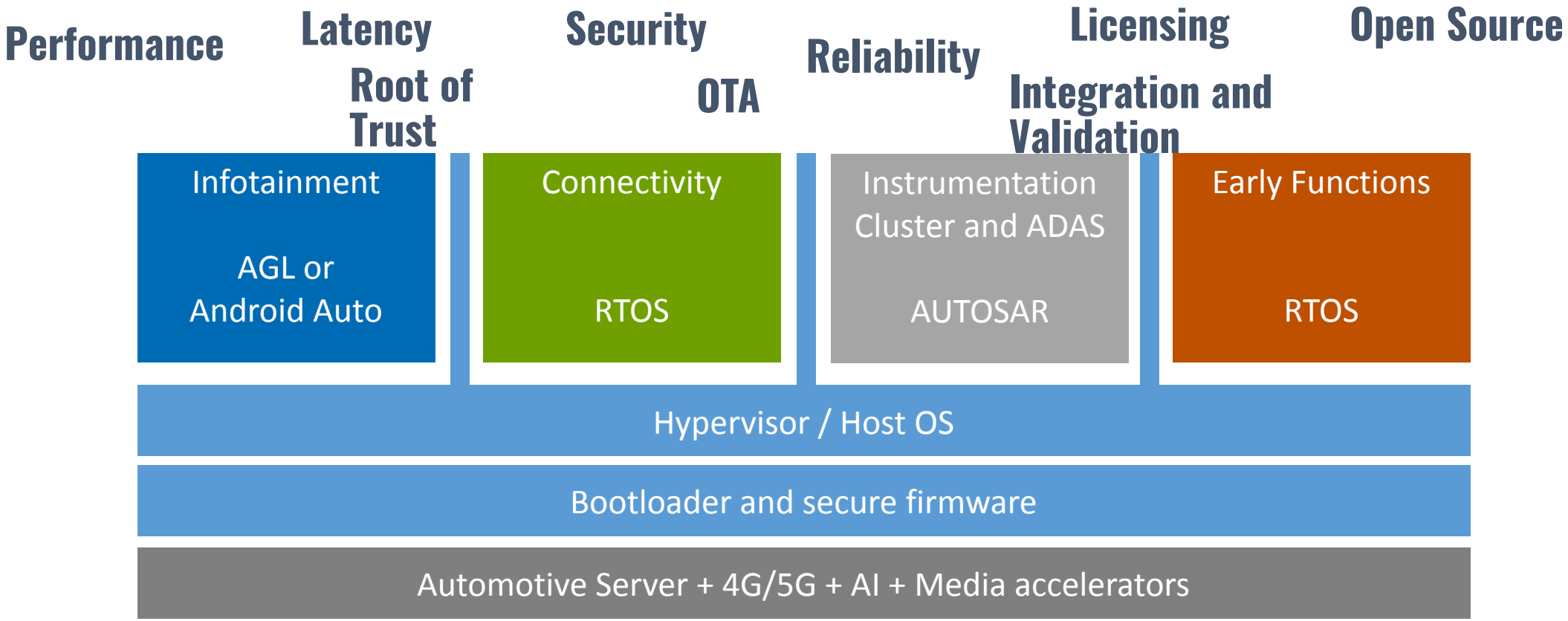
- From multiple embedded single-function ECUs to one central automotive server and a few zonal gateways



https://www.researchgate.net/figure/Typical-in-vehicle-network-architecture-in-a-modern-car_fig2_305499872

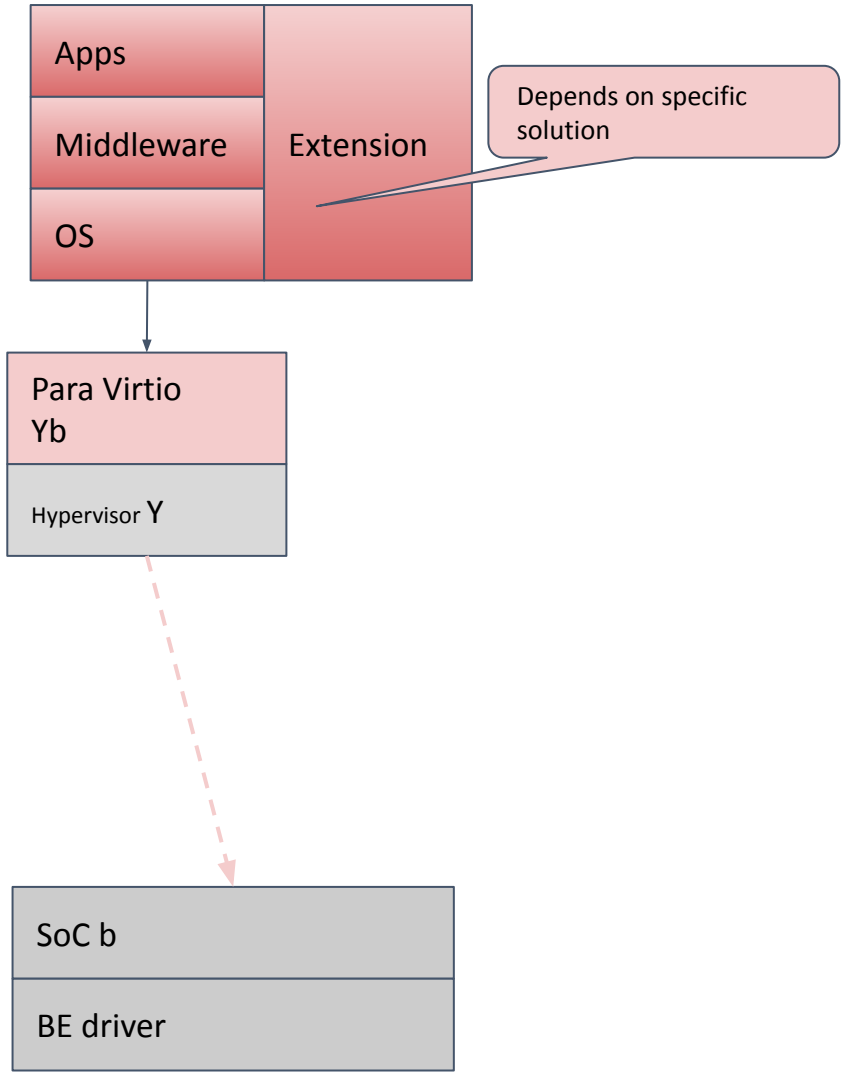
Software Defined Vehicles

- From multiple embedded firmware to one hypervisor running multiple images

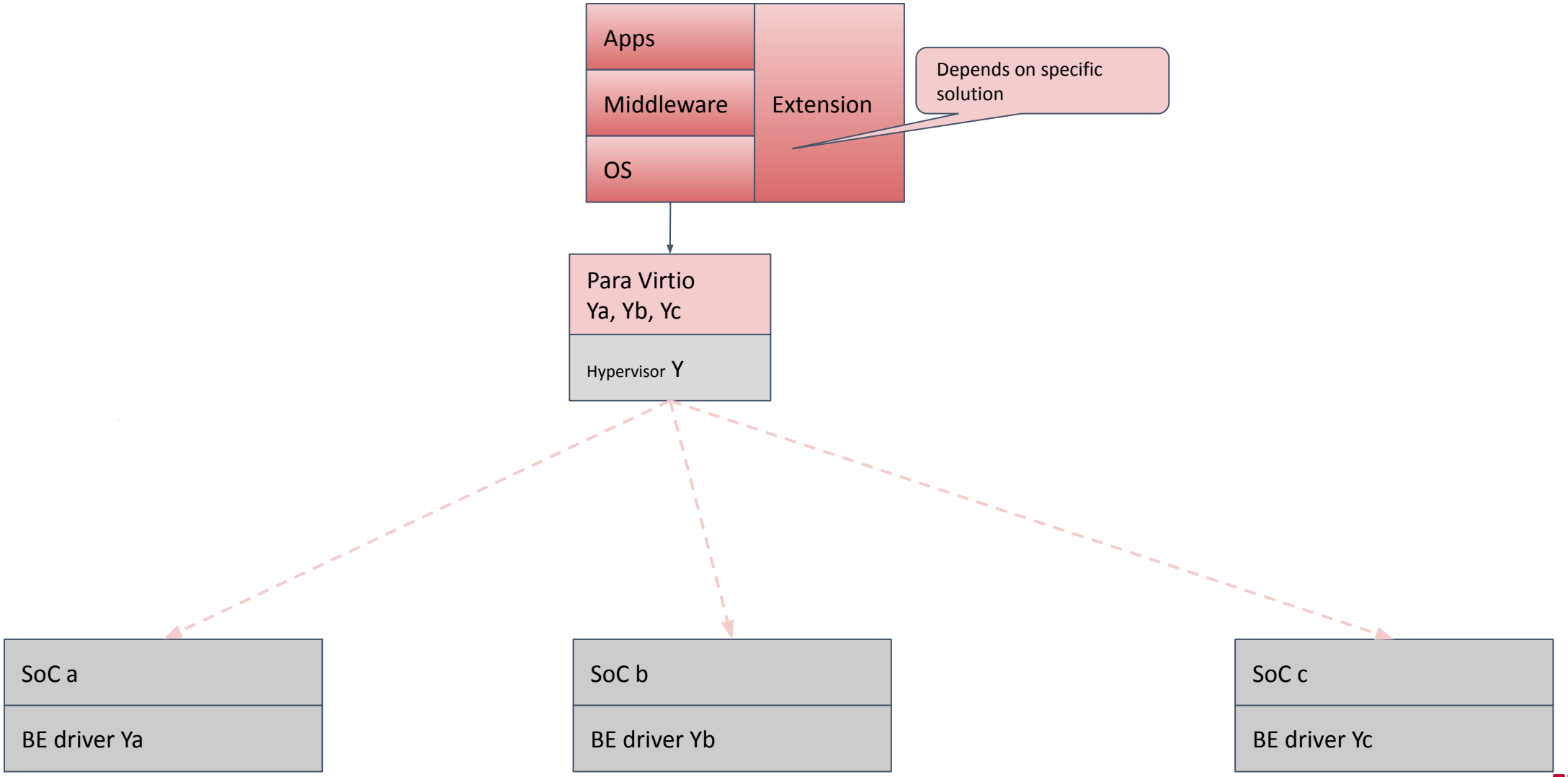


Hypervisors

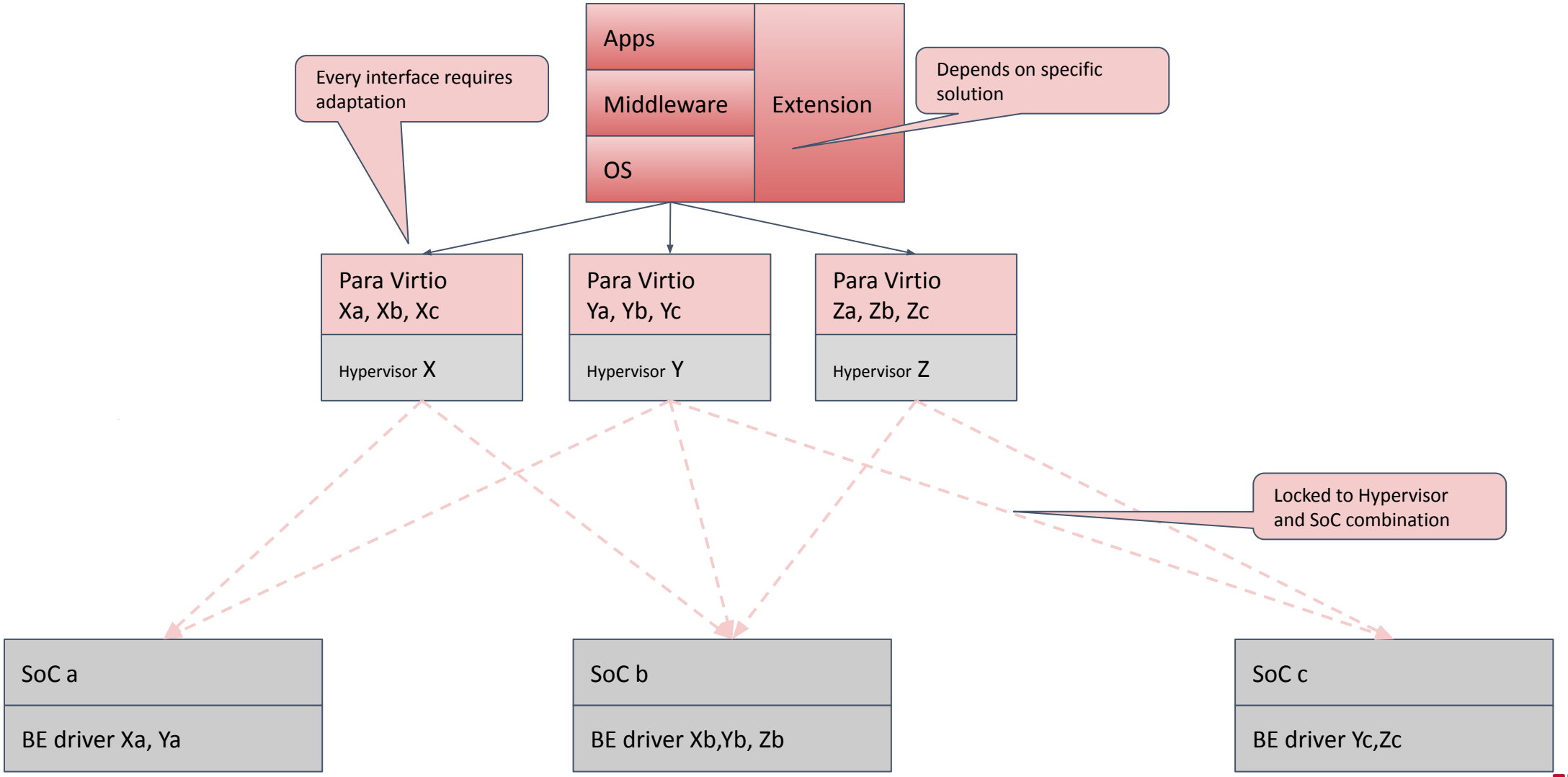
Virtualization-specific work



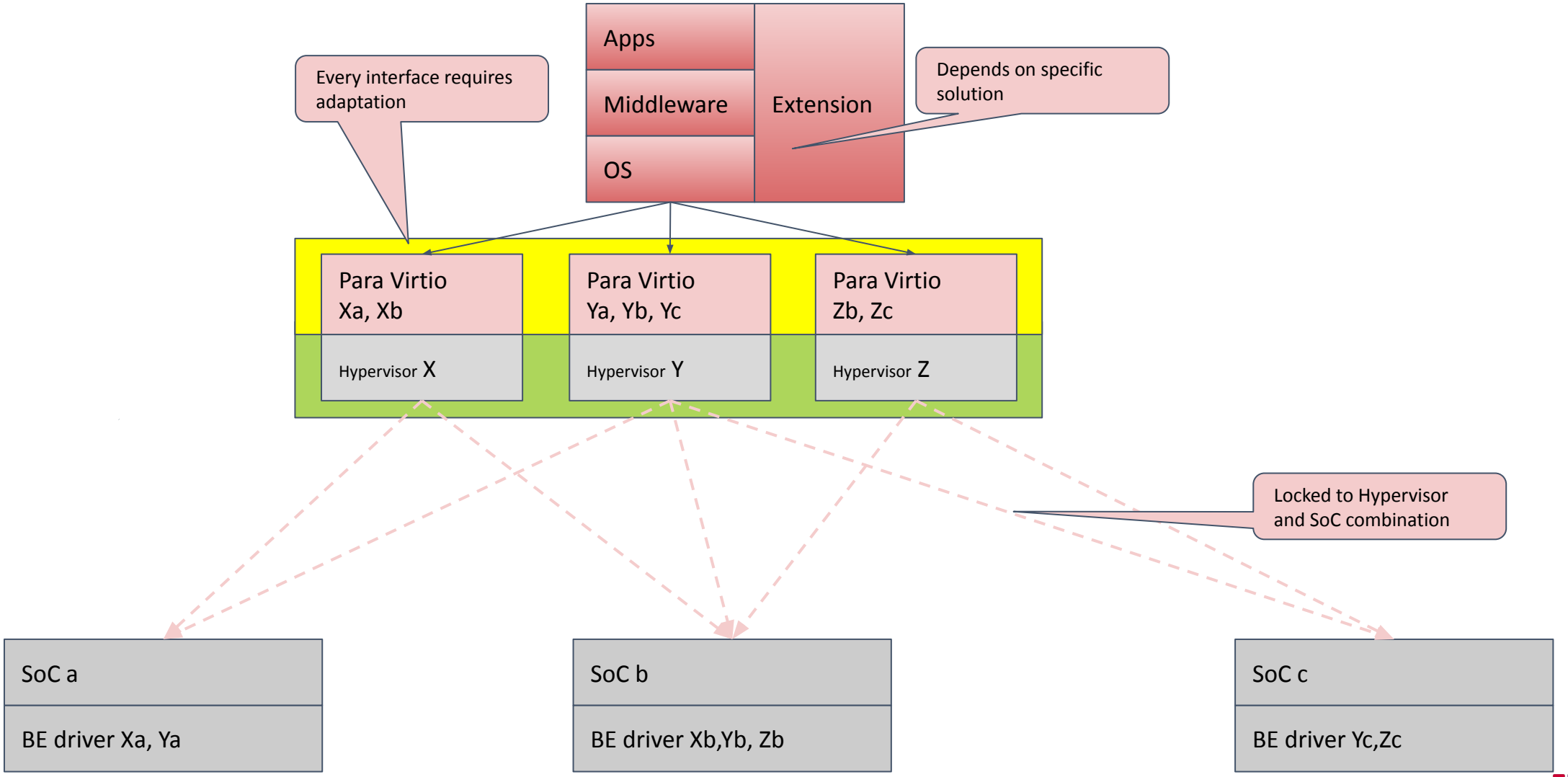
SoC-specific work for a given hypervisor



SoC-specific work for each hypervisor

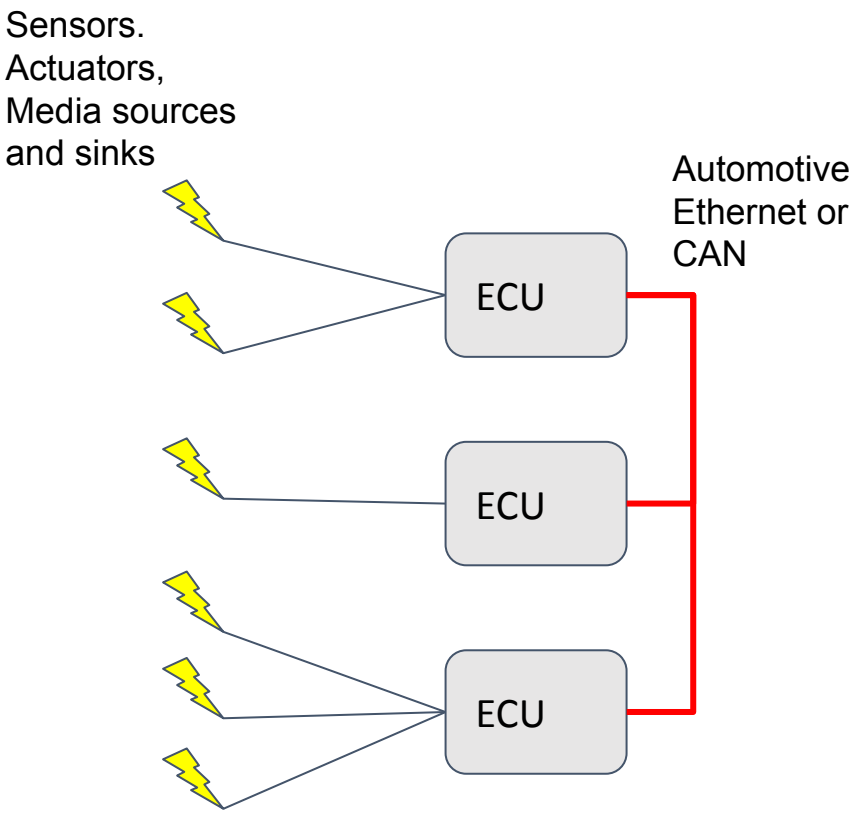


Standardize virt devices and hypervisor APIs

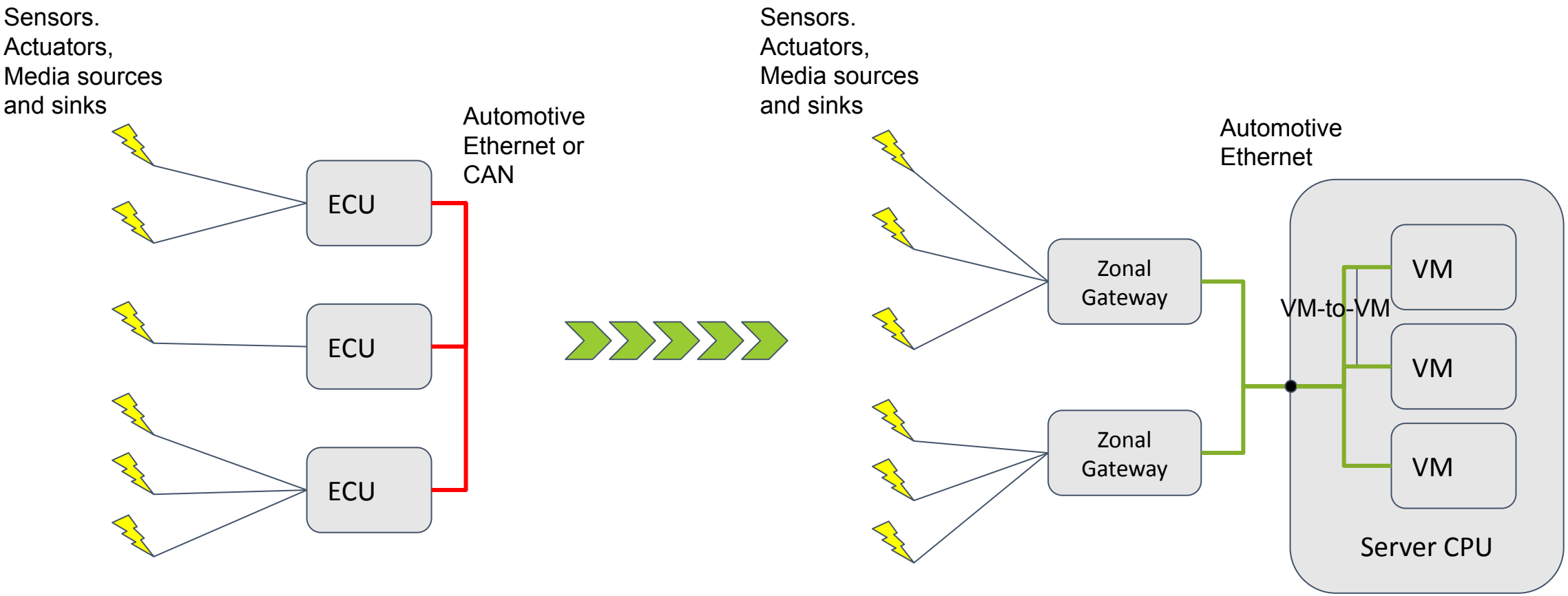


Automotive Ethernet

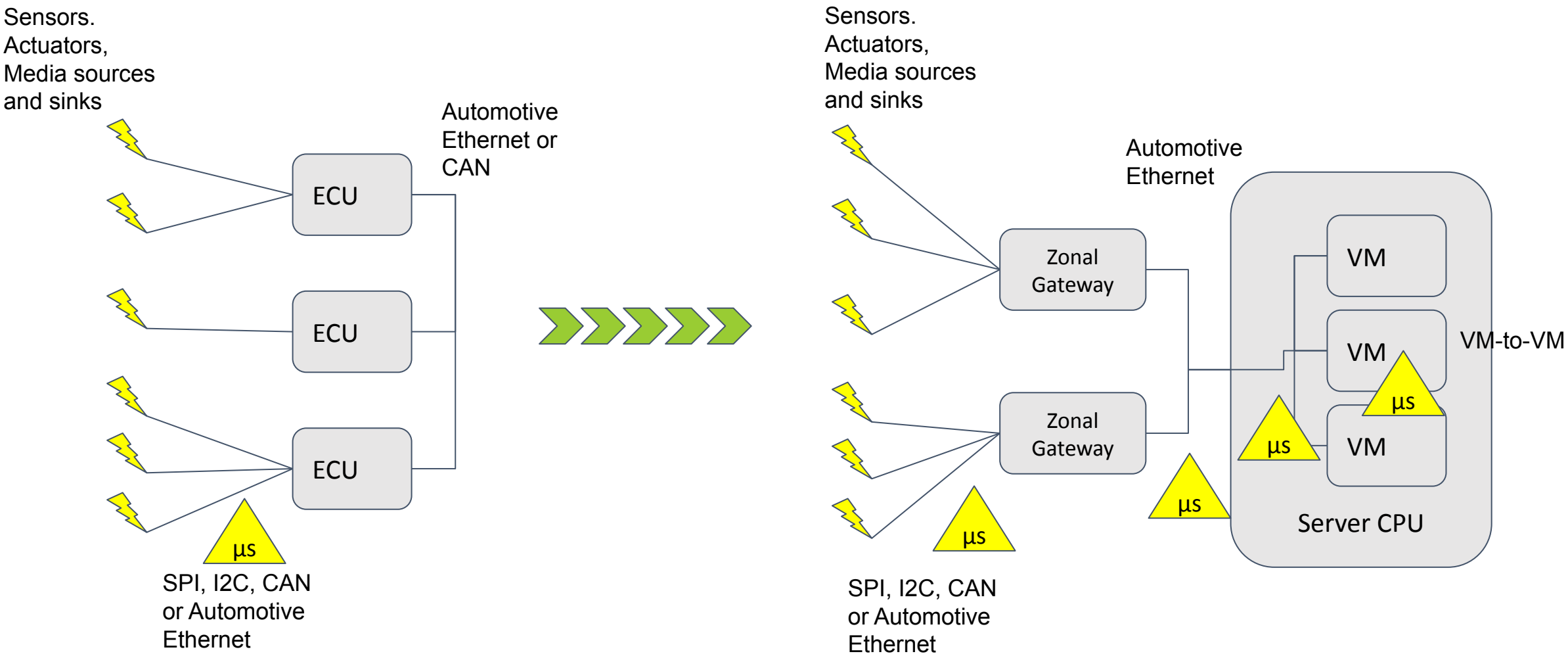
From Ethernet to VM-to-VM communication



From Ethernet to VM-to-VM communication



Latency from physical bus up to the VM user code



Over The Air software upgrades

Secure reliable Over-The-Air Updates

https://www.arm.com/blogs/blueprint/software-defined-vehicle

Enabling SDV with open source at Linaro

We enable New Markets through Collaborative Engineering

Android Ecosystem	IoT & Embedded	Artificial Intelligence	Kernel & Toolchain	Security	OS Build & Test	Trusted Substrate	Datacenter & Cloud	96Boards
Android Common Kernel Upstreaming AOSP Reference Boards	Smart Sensors & IoT Devices Open Source RTOS Security & Trusted Firmware	HPC/AI Microcontrollers	Kernel & toolchain technologies are essential to maintain software health. We facilitate the access to maintainers, support regression testing and improve compiler technology.	Security is no longer an option, it's a vital ingredient to be able to protect intellectual property, communications, bank accounts, personal digital belongings etc.	Open Source Tools Kernel & Toolchain CI Middleware CI	Over-the-air Updates Edge Computing Dependable Boot Trusted Services	Cloud Infrastructure Big Data Server Architecture HPC/Server	Open Source Hardware Built on the latest silicon Open Platform specifications
Linaro Consumer Group	Linaro IoT & Embedded Group					Linaro Edge & Fog Computing Group	Linaro Datacenter & Cloud Group	

Linaro accelerates product deployment in the Arm ecosystem

66
OSS Maintainers

Contributors to the Linux kernel In the top
10

Patches submitted per month
866

250
OSS Engineers worldwide

Tests on Linux kernel per week
1m

4m
Tests on Android per week

70
Major contributor to Open Source projects

Software Defined Vehicle

- From multiple embedded single-function ECUs to one central automotive server and a few zonal gateways
  - ECU software → VMs and hypervisor agnostic
  - CANbus / Automotive Ethernet
  - TSN → Time sensitive applications
  - Diversity of SoC → Common Standards
  - Security, OTA and software updates
  - FuSa

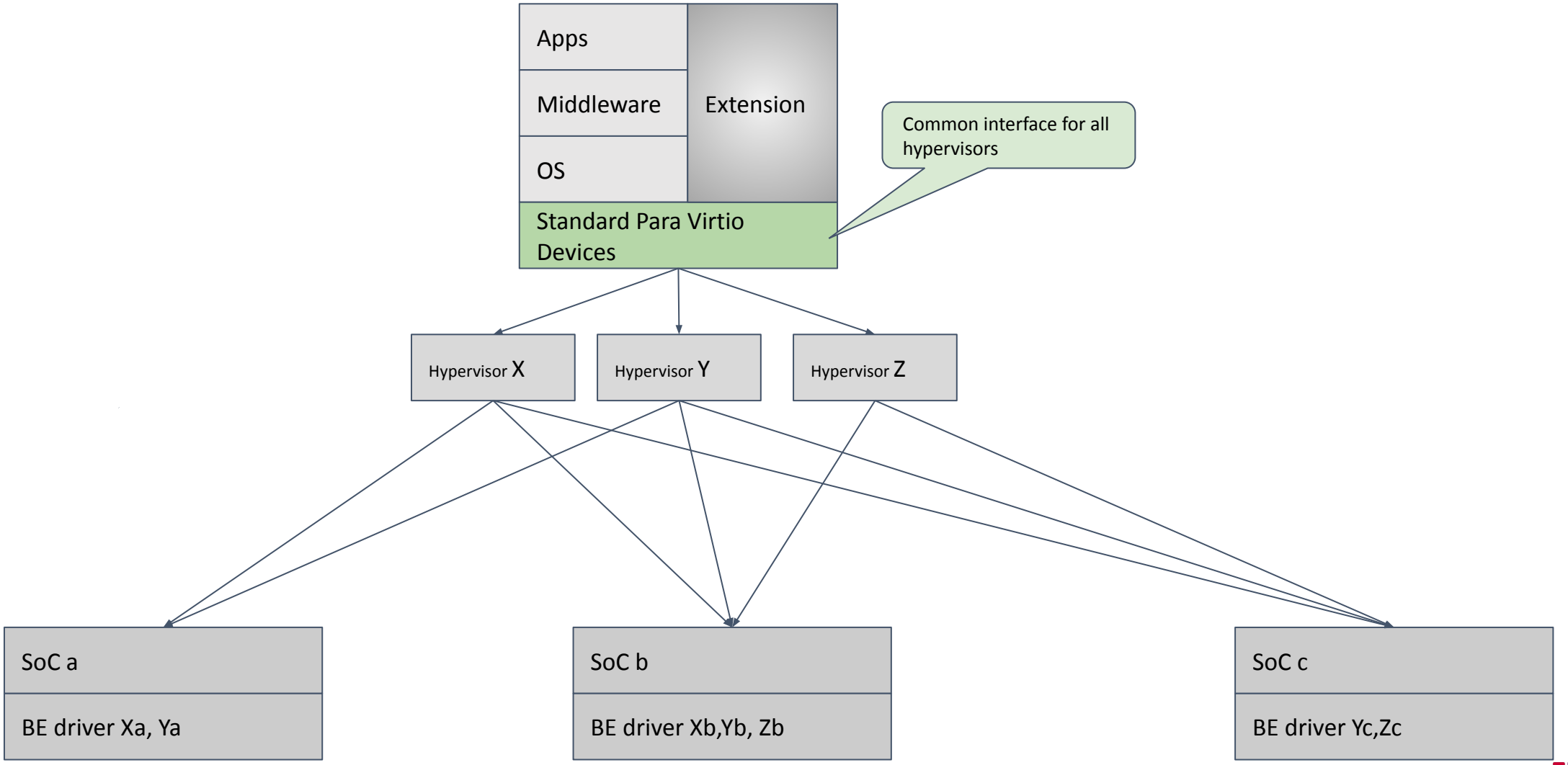
Virtio and Project Stratos

https://linaro.atlassian.net/wiki/spaces/STR/overview

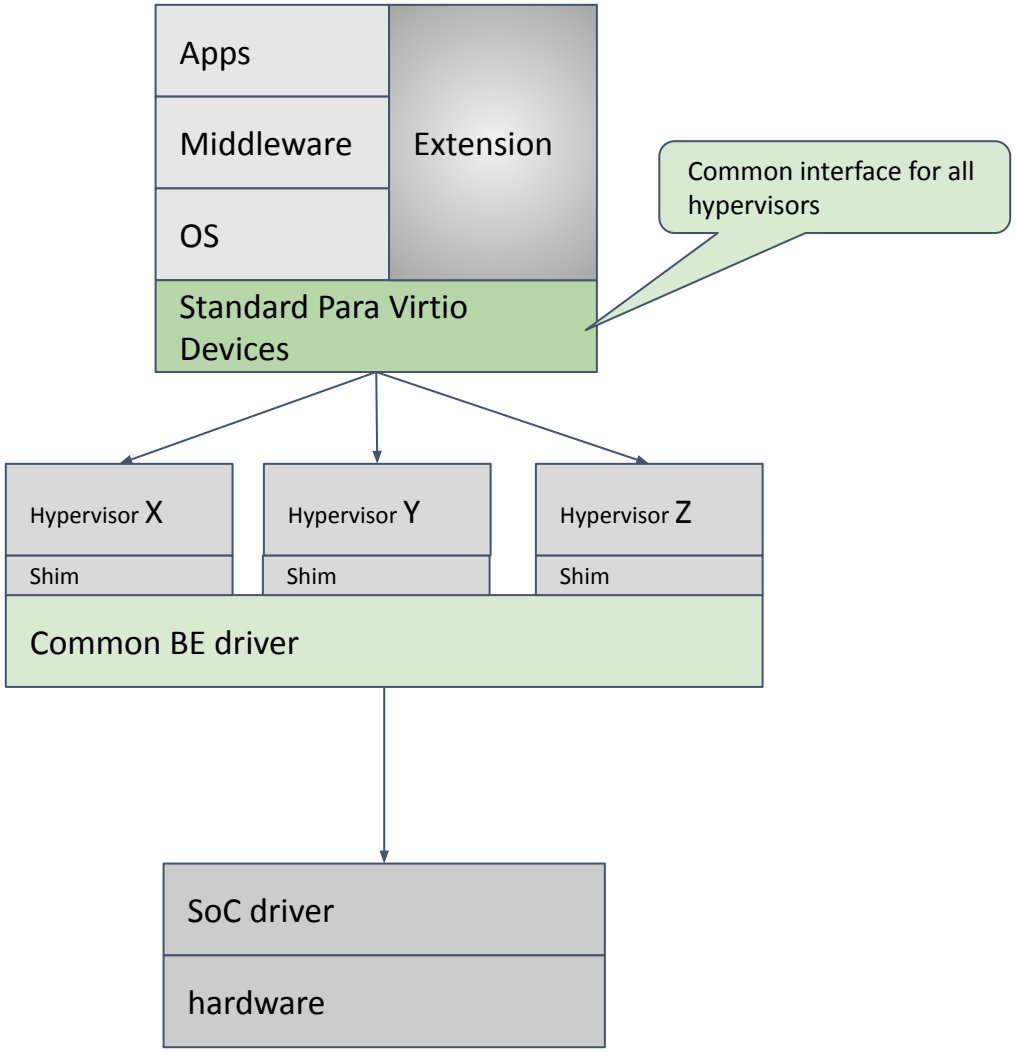
- Establish virtio as the standard interface between hypervisors, freeing a mobile, industrial or automotive platform to migrate between hypervisors and reuse the backend implementation.

- The four key areas of interest are
  - High-performance Virtio interfaces
  - Virtual Machine Monitors with a safety island
  - Boot Orchestration
  - Written Standards for the hypercalls

Virtio as a common framework



Common backend for a device class



Example Virtio Device Specifications

I2C

Virtio Specifications already merged

https://github.com/oasis-tcs/virtio-spec/blob/master/virtio-i2c.tex

Linux Front-end being merged

https://lore.kernel.org/linux-i2c/bcf2fb9bbe965862213f27e05f87ffc91283c0c5.1627018061.git.jie.deng@intel.com/

Backend under review

https://github.com/rust-vmm/vhost-device/pull/1

GPIO

Virtio Specification under review

https://lists.oasis-open.org/archives/virtio-dev/202107/msg00232.html

Initial Linux driver implementation

https://lore.kernel.org/linux-gpio/cover.1627989586.git.viresh.kumar@linaro.org/

Backend is WIP

Software Defined Vehicle

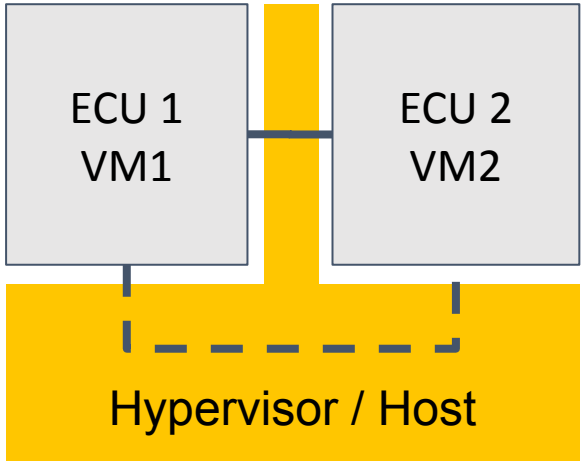
- From multiple embedded single-function ECUs to one central automotive server and a few zonal gateways
  - ECU software → VMs and hypervisor agnostic
  - CANbus / Automotive Ethernet
  - TSN → Time sensitive applications
  - Diversity of SoC → Common Standards
  - Security, OTA and software updates
  - Security and FuSa

Secure and reliable VM-to-VM communication

Physical Automotive Ethernet shall map to a virtio Ethernet device

Each hypervisor can implement in different ways

ECUs do not know which hypervisor implements it how -- is this safe and reliable?

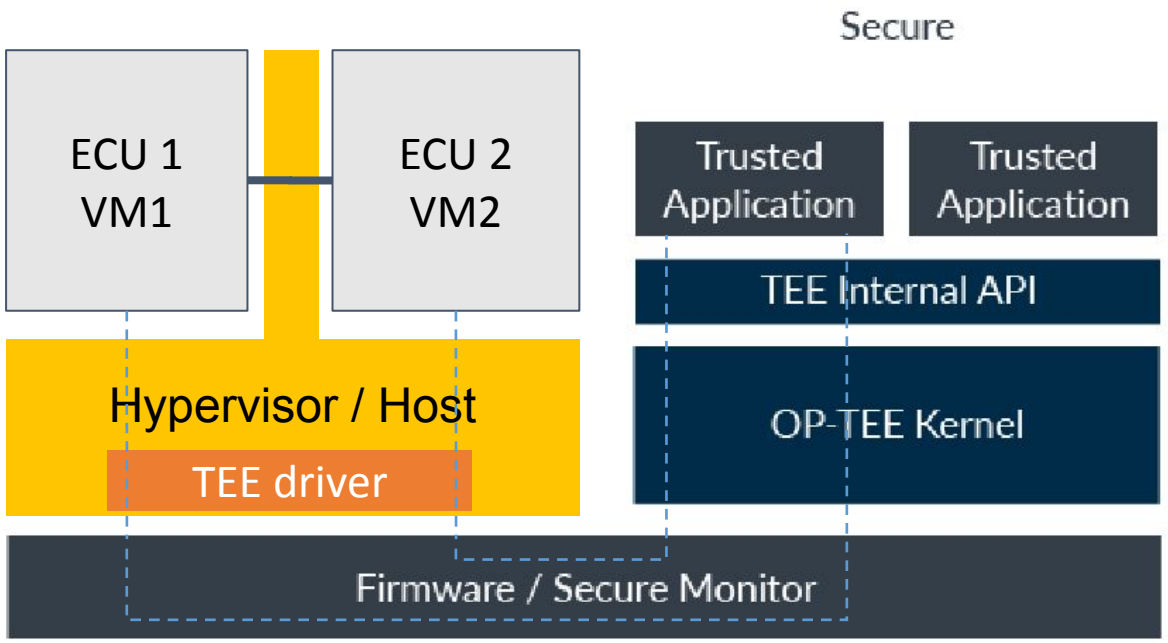


Secure and reliable VM-to-VM communication

Physical Automotive Ethernet shall map to a virtio Ethernet device

Each hypervisor can implement in different ways

ECUs do not know which hypervisor implements it how -- is this safe and reliable?

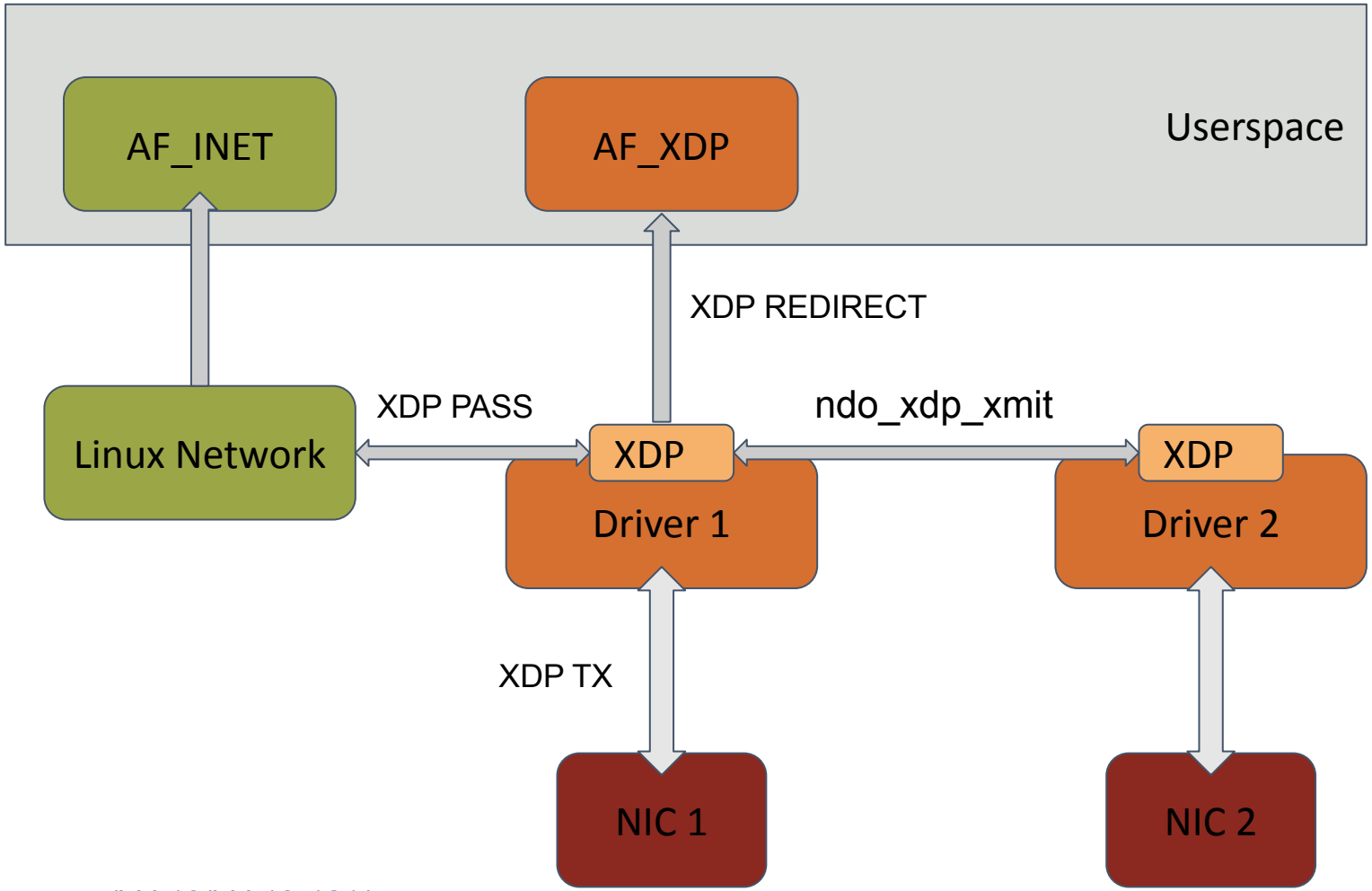


Would a TEE / secure mode handshake or secure infrastructure implementation make sense?

Software Defined Vehicle

- From multiple embedded single-function ECUs to one central automotive server and a few zonal gateways
  - ECU software → VMs and hypervisor agnostic
  - CANbus / Automotive Ethernet
  - TSN → Time sensitive applications
  - Diversity of SoC → Common Standards
  - Security, OTA and software updates
  - FuSa

What is AF_XDP



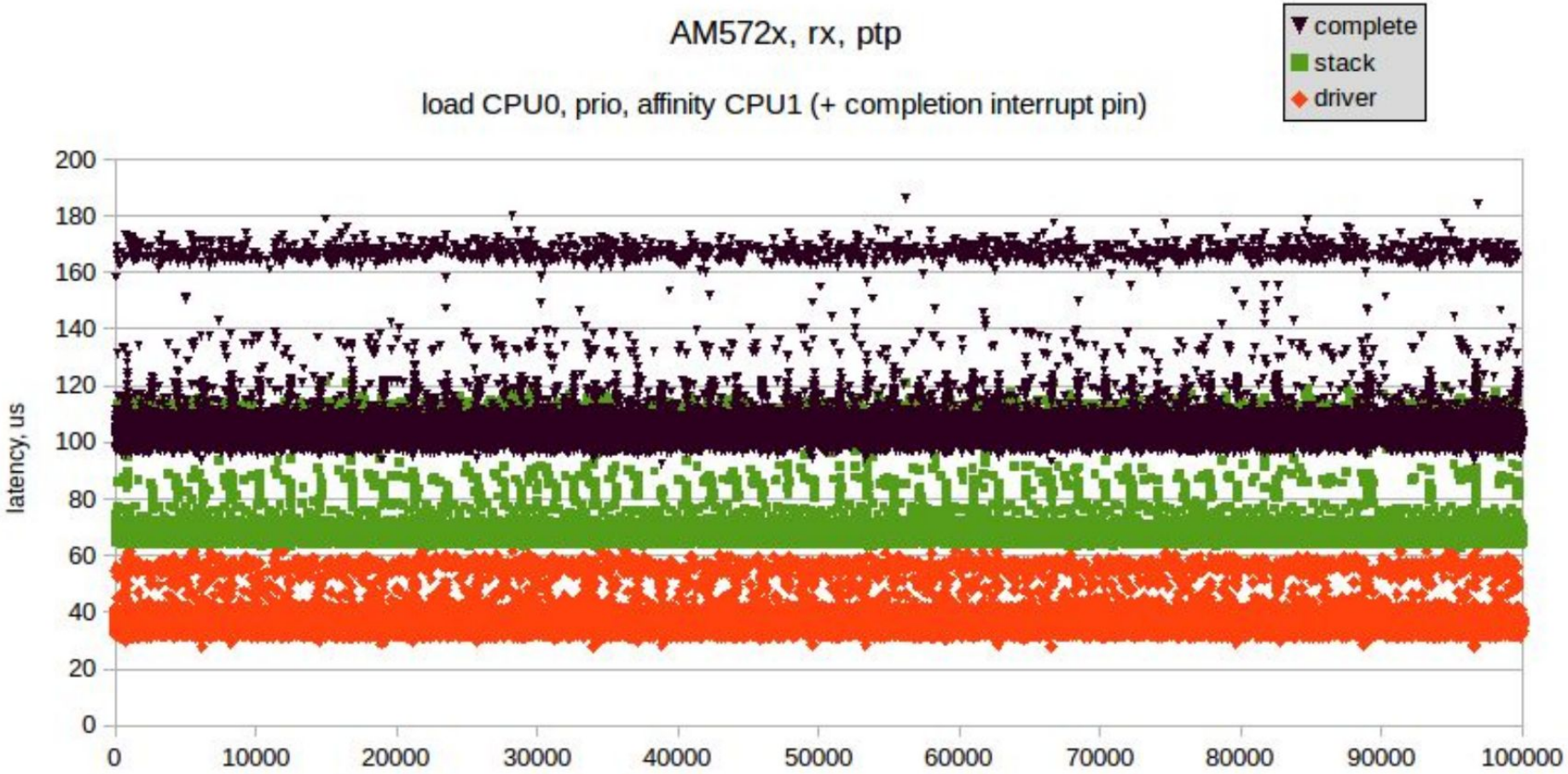
https://connect.linaro.org/resources/bkk19/bkk19-121/

Why XDP for TSN

- TSN requires bounded latency and low bounded jitter
- XDP can't guarantee jitter. It can offer significantly lower latency compared to the default network stack
- Due to its design can work well with 'mixed' packet scenarios. TSN packets can be offloaded to user-space while non-critical traffic is handled by the kernel
- AF_XDP is designed to operate as a socket
- AF_XDP L2 packets can work really well with existing user-space L2 solutions (VPP, LWIP etc)

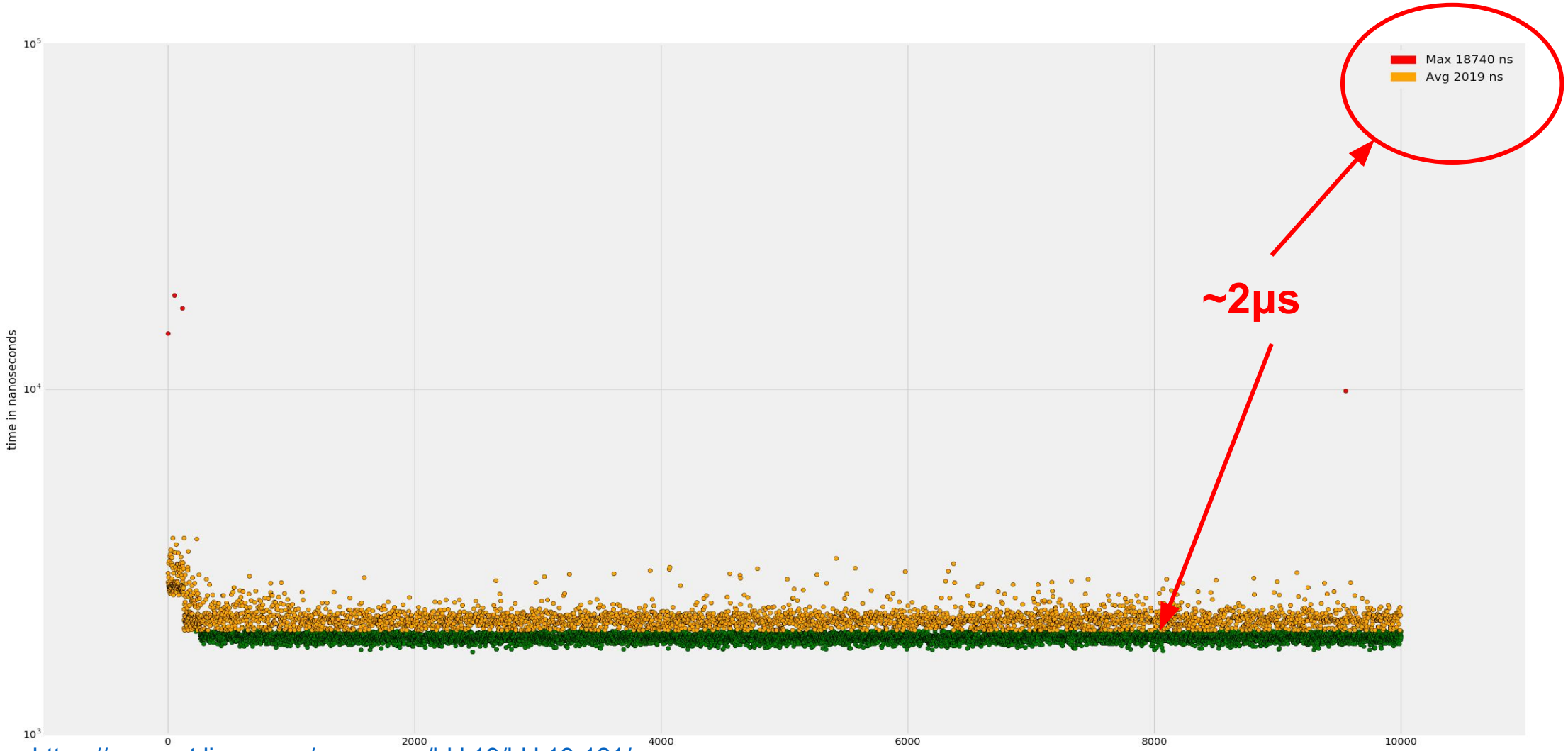
TSN without AF_XDP

latency, us	driver	stack	complete
maximum	70.05	121.67	186.16
minimum	27.97	63.12	92.30
mean +- RMS	37.04 ±2.90	67.41 ±5.73	104.45 ±8.15



https://connect.linaro.org/resources/sfo17/sfo17-209/

TSN with AF_XDP 10k packets 1kpps mmio access disabled

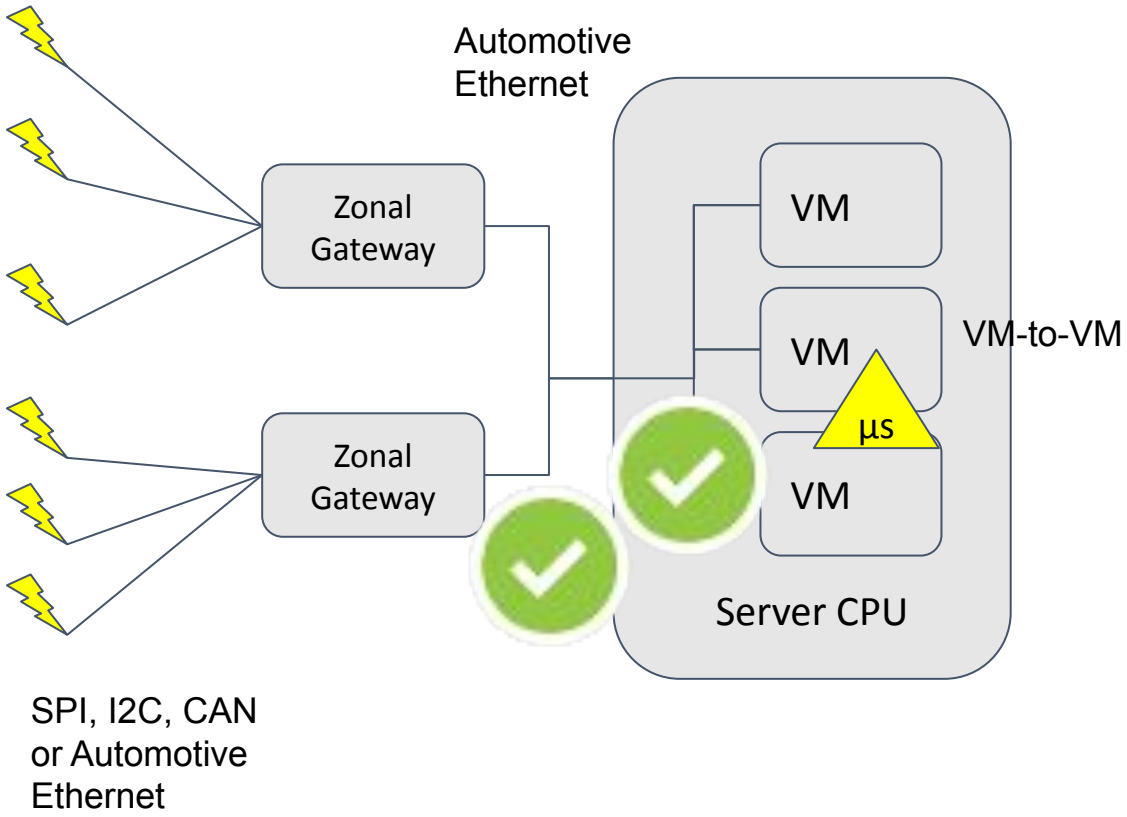


https://connect.linaro.org/resources/bkk19/bkk19-121/

To be completed: AF_XDP in VMs

AF_XDP provides the required latency up to user space code running in the host

Combine virtio and AF_XDP to deliver the required latency up to user space code running in a VM



Software Defined Vehicle

- From multiple embedded single-function ECUs to one central automotive server and a few zonal gateways
  - ECU software → VMs and hypervisor agnostic
  - CANbus / Automotive Ethernet
  - TSN → Time sensitive applications
  - Diversity of SoC → Common Standards
  - Security, OTA and software updates
  - FuSa

Arm SOAFEE

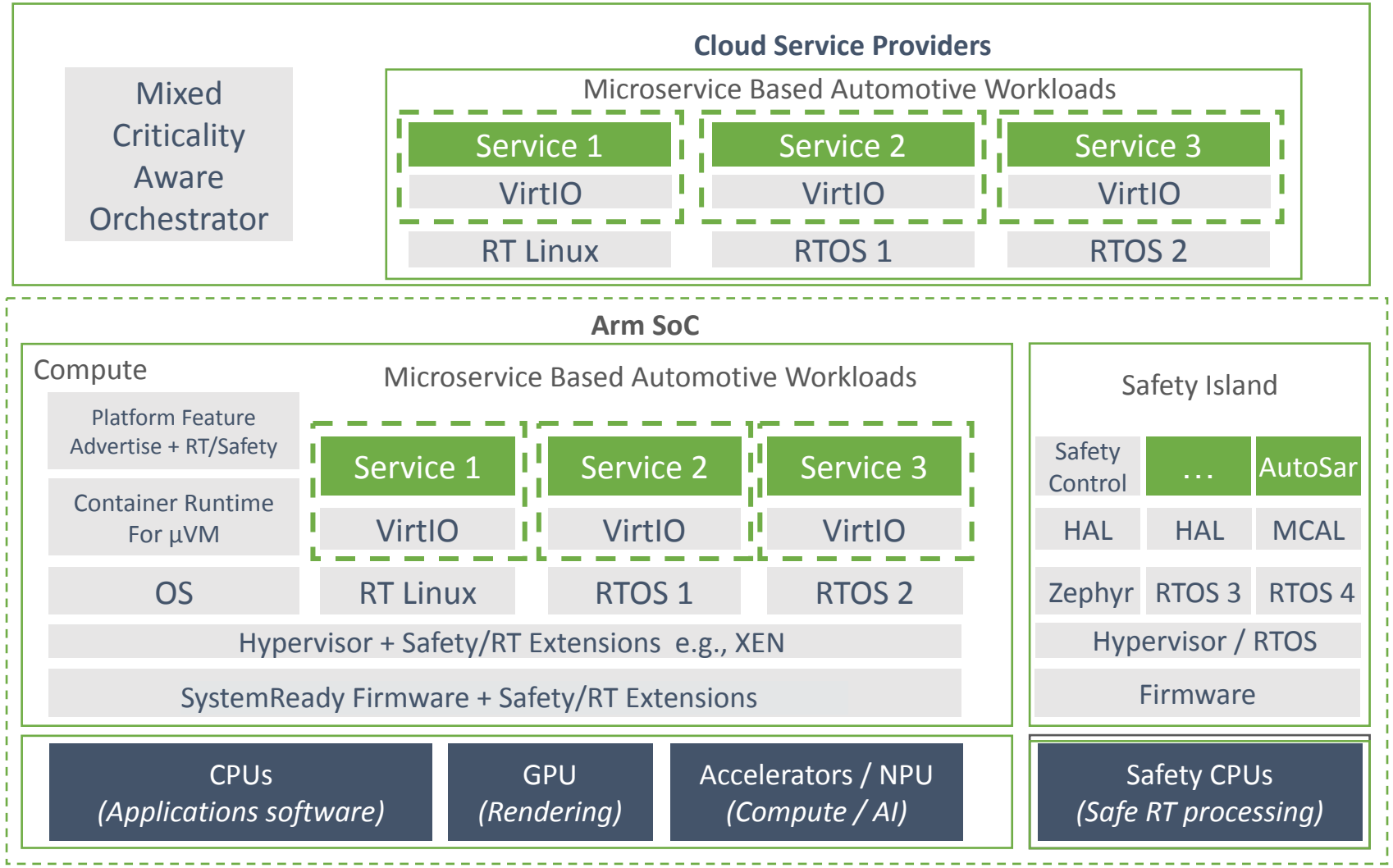
Initiative to enable cloud native software experience across Arm's embedded edge ecosystem
Based on Project Cassini and System Ready and extended to cloud native mixed critical workload development

Open standards for cloud native embedded edge
Open forum and Special Interest Group to adopt and extend standards for cloud native mixed critical SW development

Open-source reference software stack
Reference software stack for development and ecosystem seeding to enable path from development to commercial deployment

SOAFEE

Arm SOAFEE Cloud Native Architecture



Open-Source or Commercial applications benefiting from SOAFEE Architecture

Open-source or commercial technologies influenced by SOAFEE Architecture

Container

µService

Device HAL

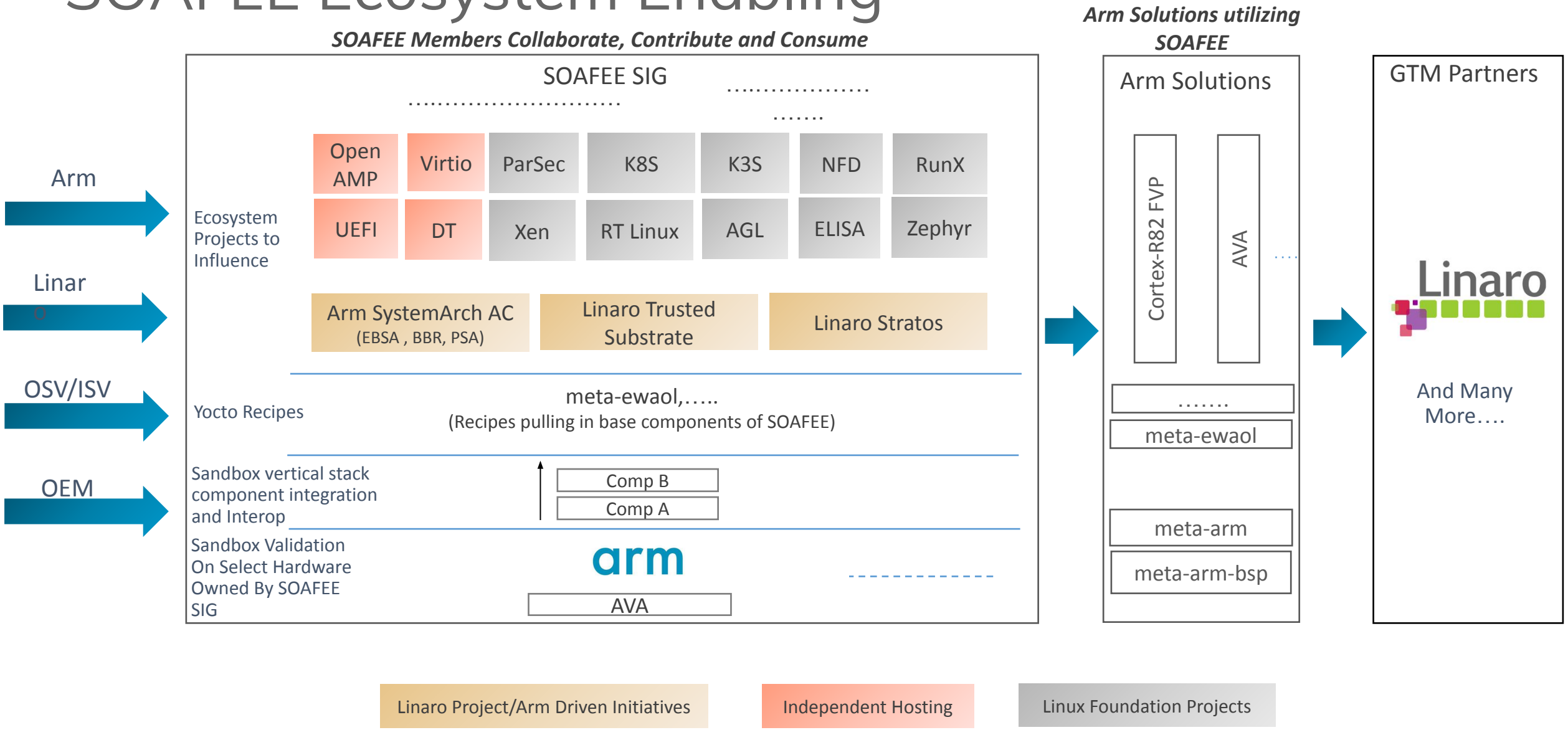
OS / RTOS

PARSEC

Arm PSA

arm SystemReady

SOAFEE Ecosystem Enabling



Software Defined Vehicle

- From multiple embedded single-function ECUs to one central automotive server and a few zonal gateways
  - ECU software → VMs and hypervisor agnostic
  - CANbus / Automotive Ethernet
  - TSN → Time sensitive applications
  - Diversity of SoC → Common Standards
  - Security, OTA and software updates
  - FuSa

Trusted Substrate

Dependable Boot

When exposed outside data centers, computers of all sizes are vulnerable to a whole new set of risks...

LEARN MORE

Over-the-air Updates

Over-the-air updates have been around for a while but this process needs to reach a degree of scalability and trust never seen before...

LEARN MORE

Trusted Services

Trusted Substrate is being developed to facilitate portable Trust Services across processor architectures and platforms...

LEARN MORE

https://www.linaro.org/trusted-substrate/

LKFT

- Improve the quality of the Linux kernel by performing functional testing on Arm hardware - LTS, stable, and upstream development branches

LKFT

About | LKFT 101 | Boards | Branches | Tests | Logs

Test Results

linux-stable-rc-linux-4.4.y	testing...	5 hours ago	478 passed	0 failed	0 skipped	View Builds
linux-stable-rc-linux-4.9.y		5 hours ago	24 passed	3 failed	0 skipped	View Builds
linux-stable-rc-linux-4.14.y	testing...	5 hours ago	1854 passed	3 failed	299 skipped	View Builds
linux-stable-rc-linux-4.19.y	testing...	5 hours ago	651 passed	0 failed	1 skipped	View Builds
linux-stable-rc-linux-5.4.y	testing...	5 hours ago	1559 passed	0 failed	124 skipped	View Builds
linux-stable-rc-linux-5.10.y	testing...	5 hours ago	8619 passed	76 failed	1125 skipped	View Builds
linux-stable-rc-linux-5.13.y	testing...	5 hours ago	73604 passed	1056 failed	12309 skipped	View Builds
linux-stable-rc-linux-5.14.y	testing...	4 hours ago	46685 passed	259 failed	7374 skipped	View Builds
linux-mainline-master	testing...	18 hours ago	14138 passed	507 failed	2362 skipped	View Builds
linux-next-master	testing...	14 hours ago	58042 passed	1079 failed	9893 skipped	View Builds

https://lkft.linaro.org/

https://tuxsuite.com/

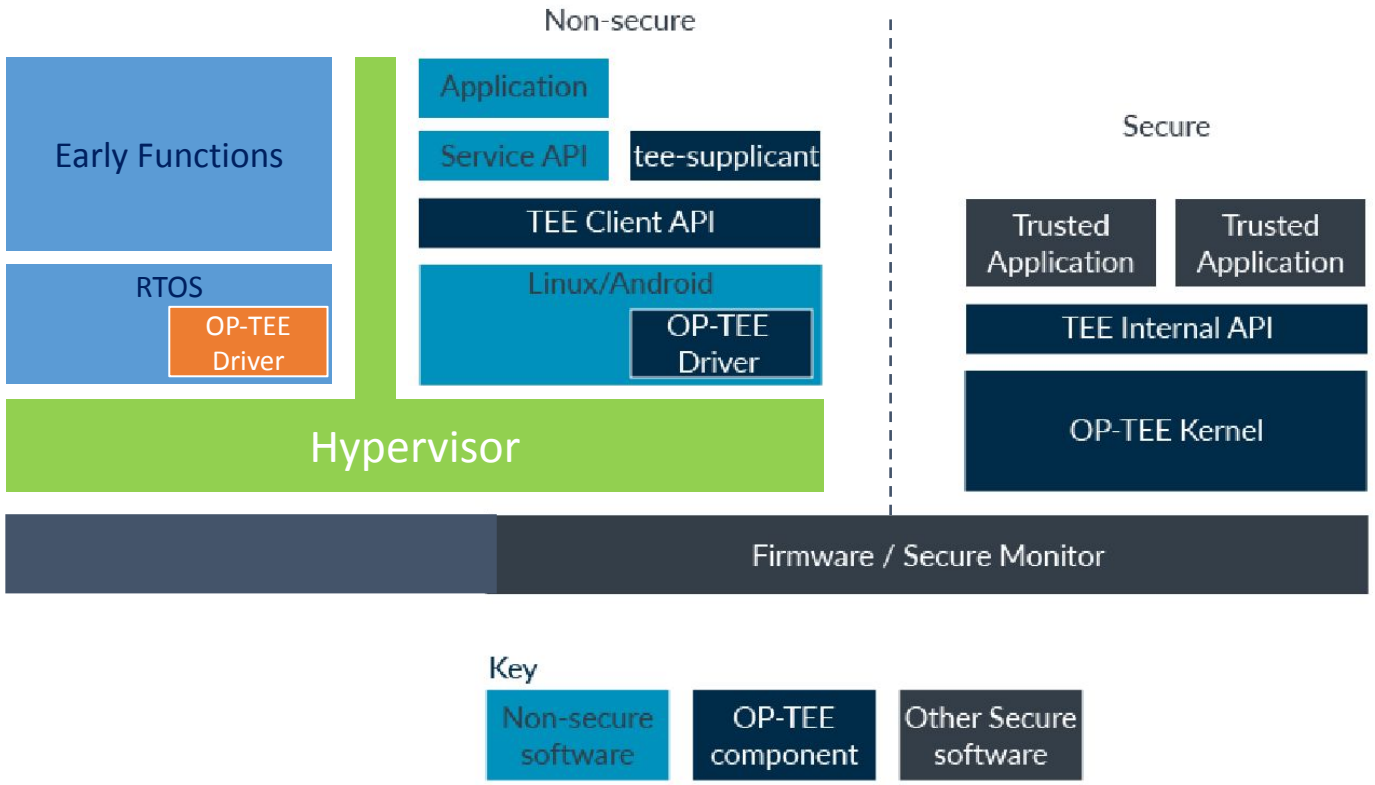
Software Defined Vehicle

- From multiple embedded single-function ECUs to one central automotive server and a few zonal gateways
  - ECU software → VMs and hypervisor agnostic
  - CANbus / Automotive Ethernet
  - TSN → Time sensitive applications
  - Diversity of SoC → Common Standards
  - Security, OTA and software updates
  - FuSa

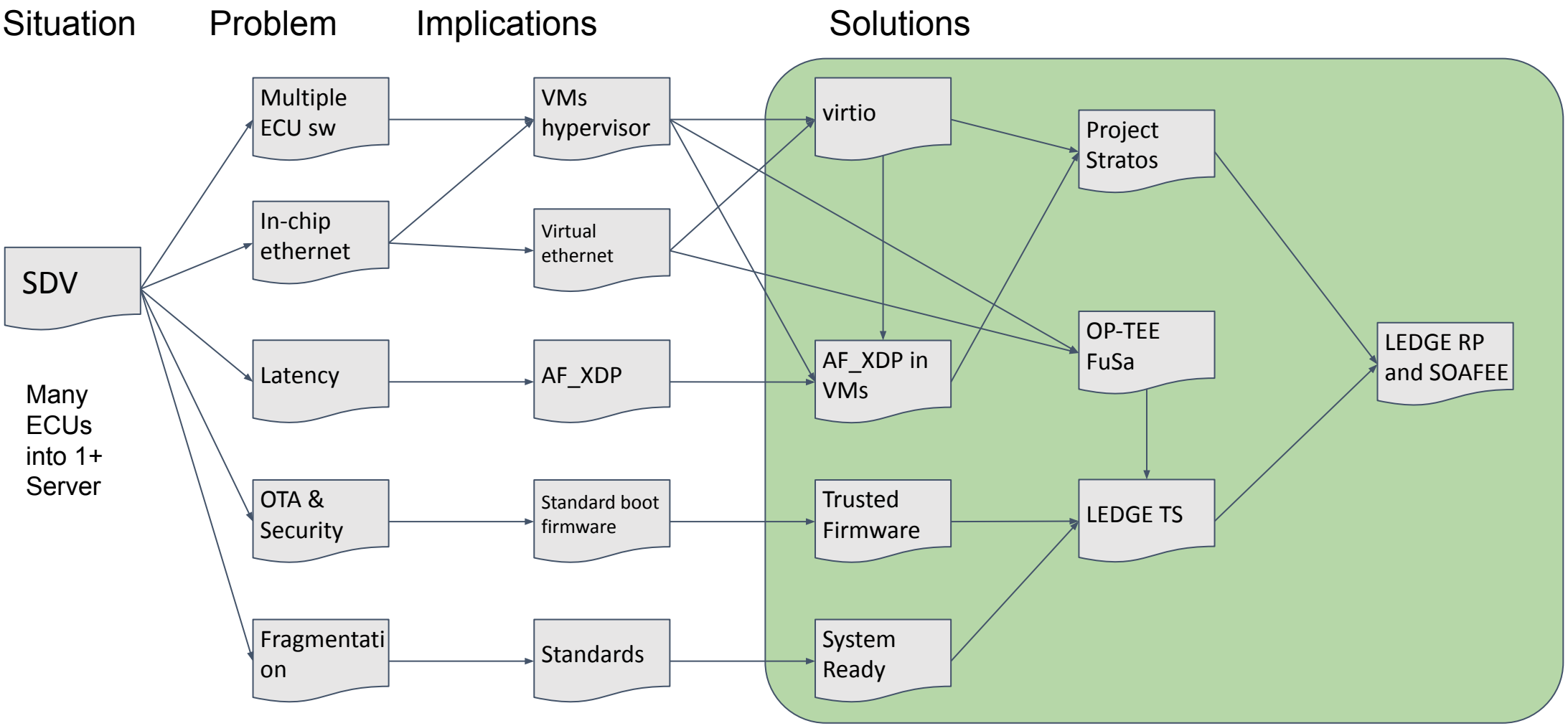
OP-TEE FuSa workshop

- Each provider of the software running in each VM is responsible for Functional Safety compliance where required

- Ongoing efforts to investigate the requirements for OP-TEE and for the Secure Monitor firmware



Implication graph



Summary

- Linaro is working closely with Arm to enable the Software Defined Vehicle disruption
- Arm SOAFEE is the reference software architecture for Cloud Native Edge
- Linaro is leading the collaborative development of open source projects in SOAFEE

Get involved with us!

Linaro Connect keynotes
Android Automotive OS keynote by Google
5G and AI keynote by Qualcomm
From Mobile to Automotive: Delivering Intelligent, Next-Gen Digital Cockpit Solutions, keynote by Qualcomm

White Papers
Software Defined Vehicles and the need for standardization
Trusted Substrate

Thank you

andrea.gallo@linaro.org