



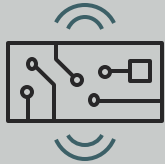
Cloud-First Automotive Development: a **Big Loop** with **Parity**

Kazuo Kajimoto, Stefano Marzani
Sep 21st, 2023



Vehicle Software complexity is growing

Today's vehicles

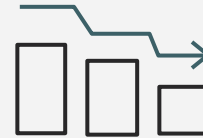


Number of ECUs
>100

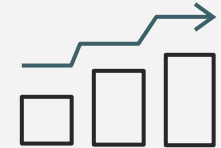


Lines of code
100–150M

Fully software-defined vehicles



Number of ECUs
Consolidating



Lines of code
Increasing

Software-defined vehicles journey

Breaking down SDV



Hardware abstraction

Enabling a seamless experience between a developer's local environment, the cloud, and the vehicle



SW & Devices lifecycle management

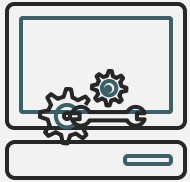
Auto DevOps Platform to increase SW quality, developer productivity, and provide better security and governance



Vehicle data management

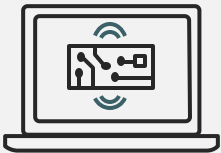
Enabling automotive customers to unlock enterprise-wide value from connected car data

How cloud can help



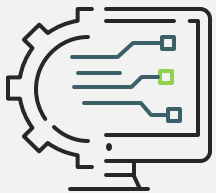
Virtual workbenches

Enabling developer efficiency through platform engineering concepts focused on providing self-service, global use and security, reduction in Hardware dependency, consistent reproducible SW artifacts and toolchains



Virtual Electronic Control Units (vECUs) and Targets

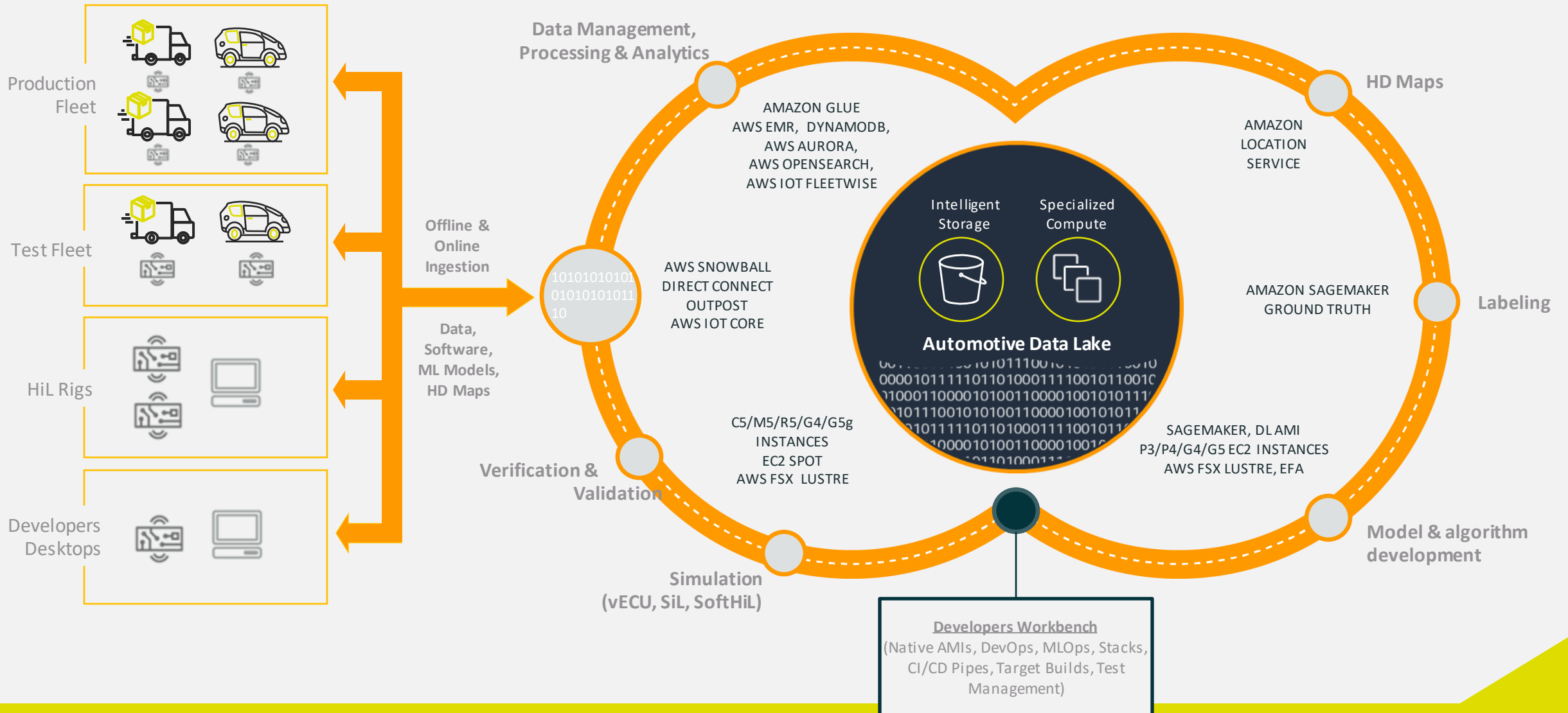
Enabling a seamless experience between a developer's local environment, the cloud, and the vehicle



Cloud-Native Automotive Software


Rearchitecting automotive software stacks and tools to facilitate software modularity, portability, and integration

Automotive Development Workflow: a Big Loop




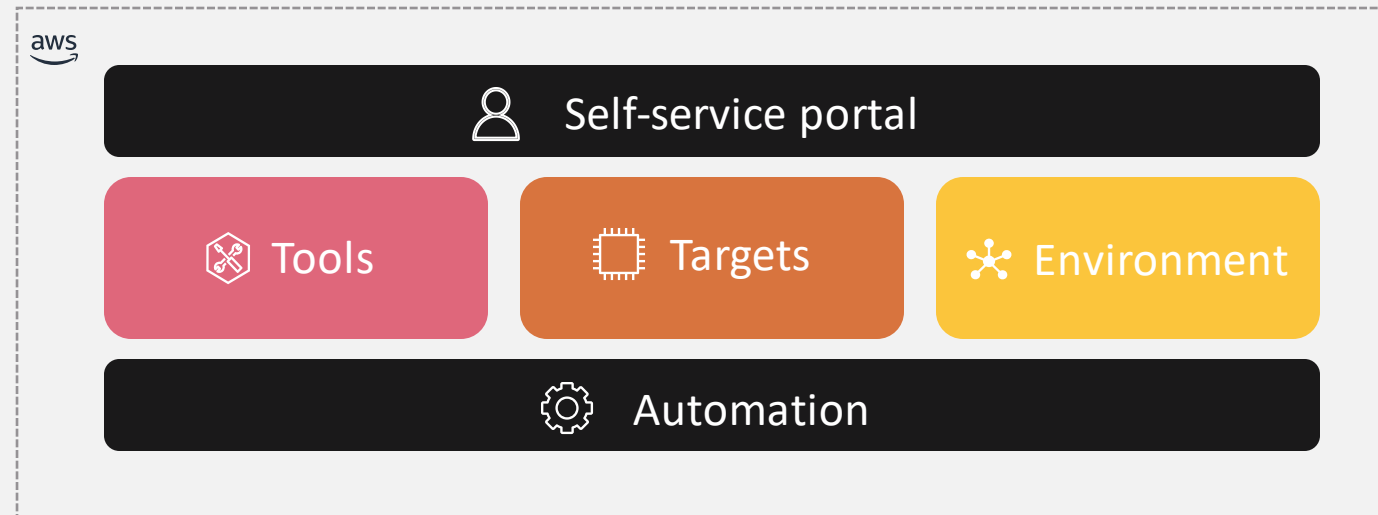
Virtual Engineering Workbench (VEW)

The Virtual Engineering Workbench consists of three main pillars: tools, targets and environment. We manage that on the platform and the interface to the user is a self-service portal empowering to do things in a fully automated manner.

 Predefined use case specific environments with all of the **tools**, IDE's, licensing – everything that enables a user to get started on a use case.

 Various levels of abstractions of the **target** platform that the user builds for.

 The **environment** is the input, in different levels of abstraction or fidelity, that we give to the workload on the targets to verify their function,



Engineering Workbench / Customers

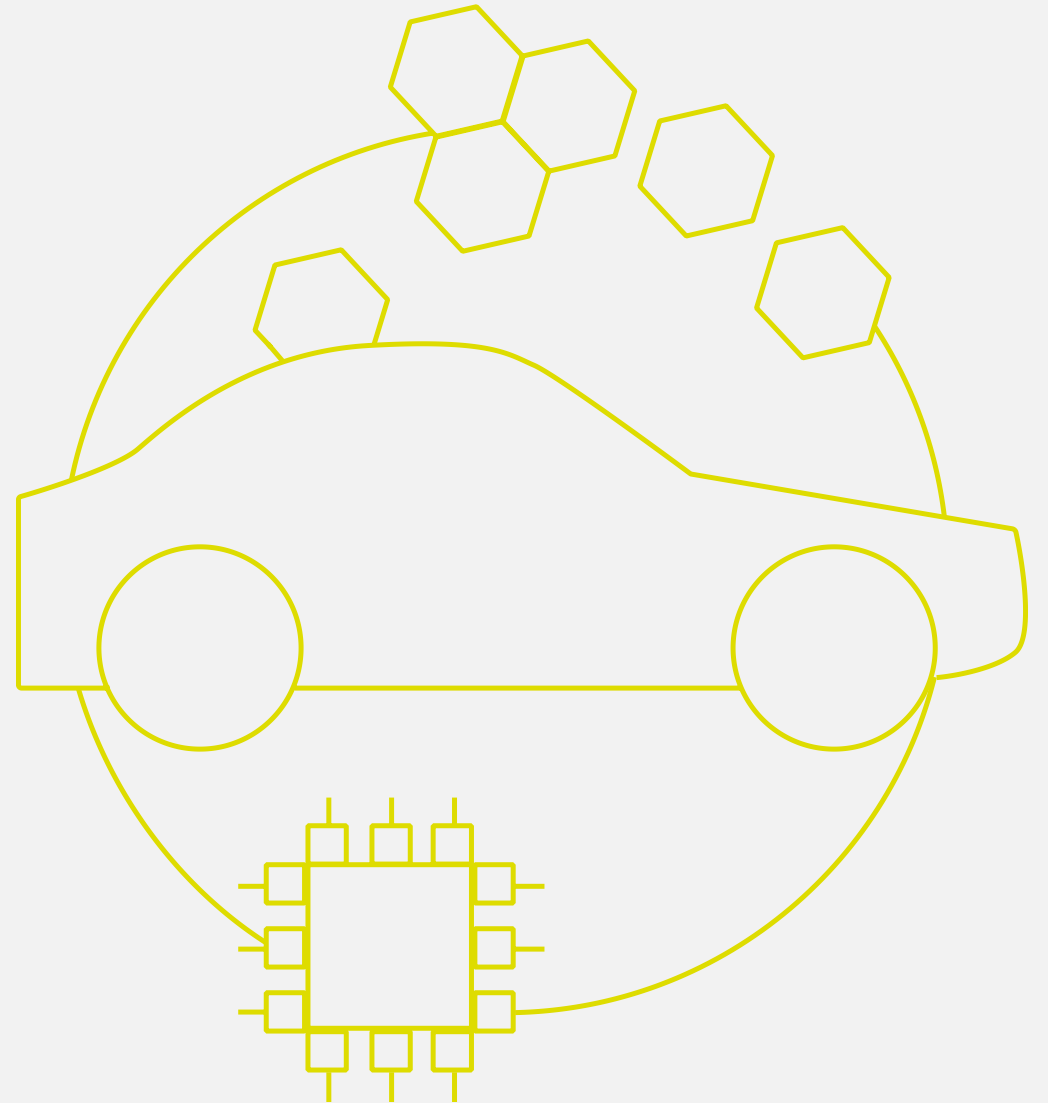
“Software is at the forefront of innovation in cars. With our CAEdge platform, we are establishing a new approach to developing vehicle architectures and software in the automotive industry. We have found a strong collaborator in AWS to help us accomplish this task and will continue to expand our leading role in the field of automotive software”



“Both companies are creating a cloud-based product development environment called the “Virtual Engineering Workbench,” which provides automated workflows to manage software development and testing, high-performance simulations, machine learning model training, and data collection and analysis. ”

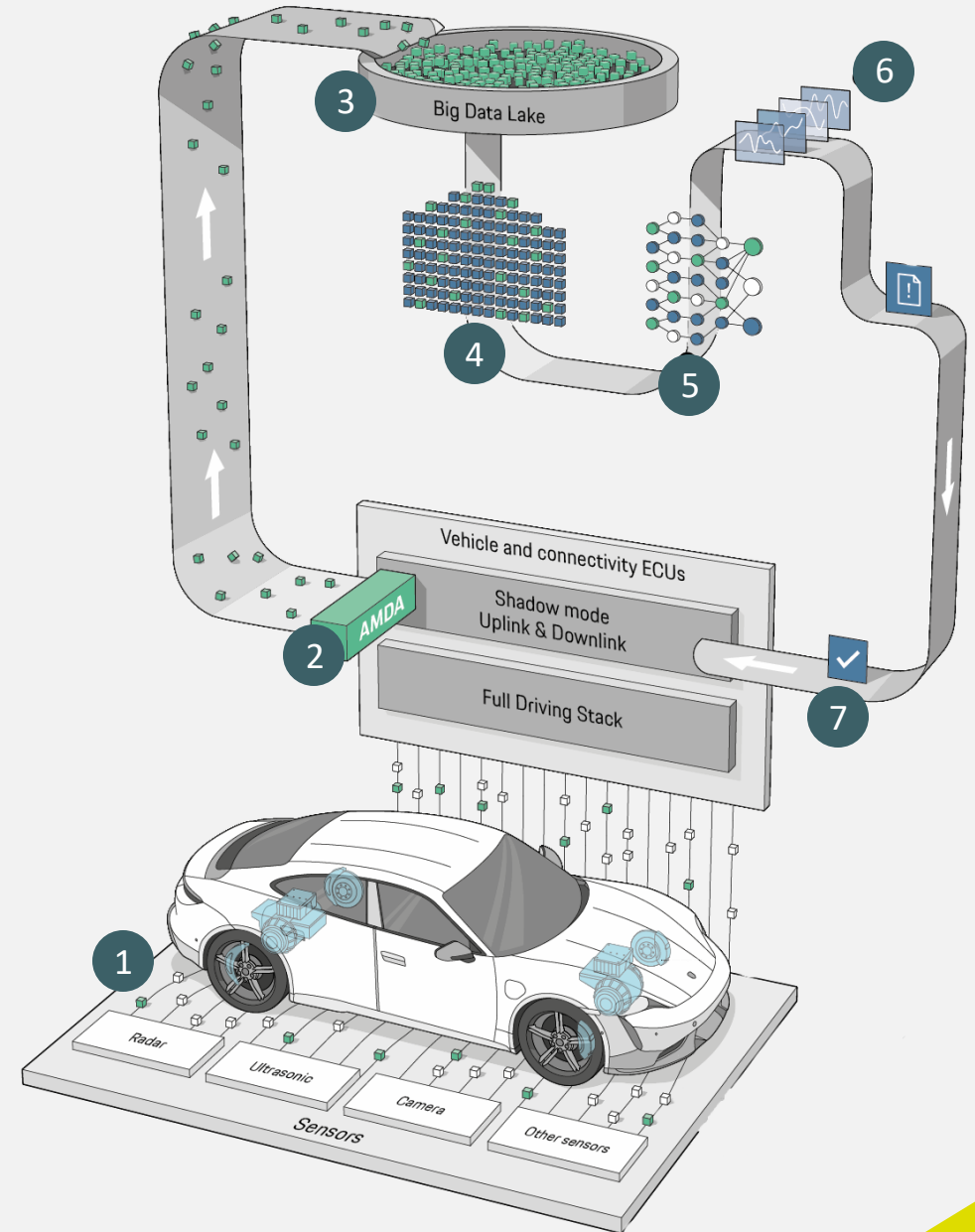


Automotive is a
“Big Loop” ...

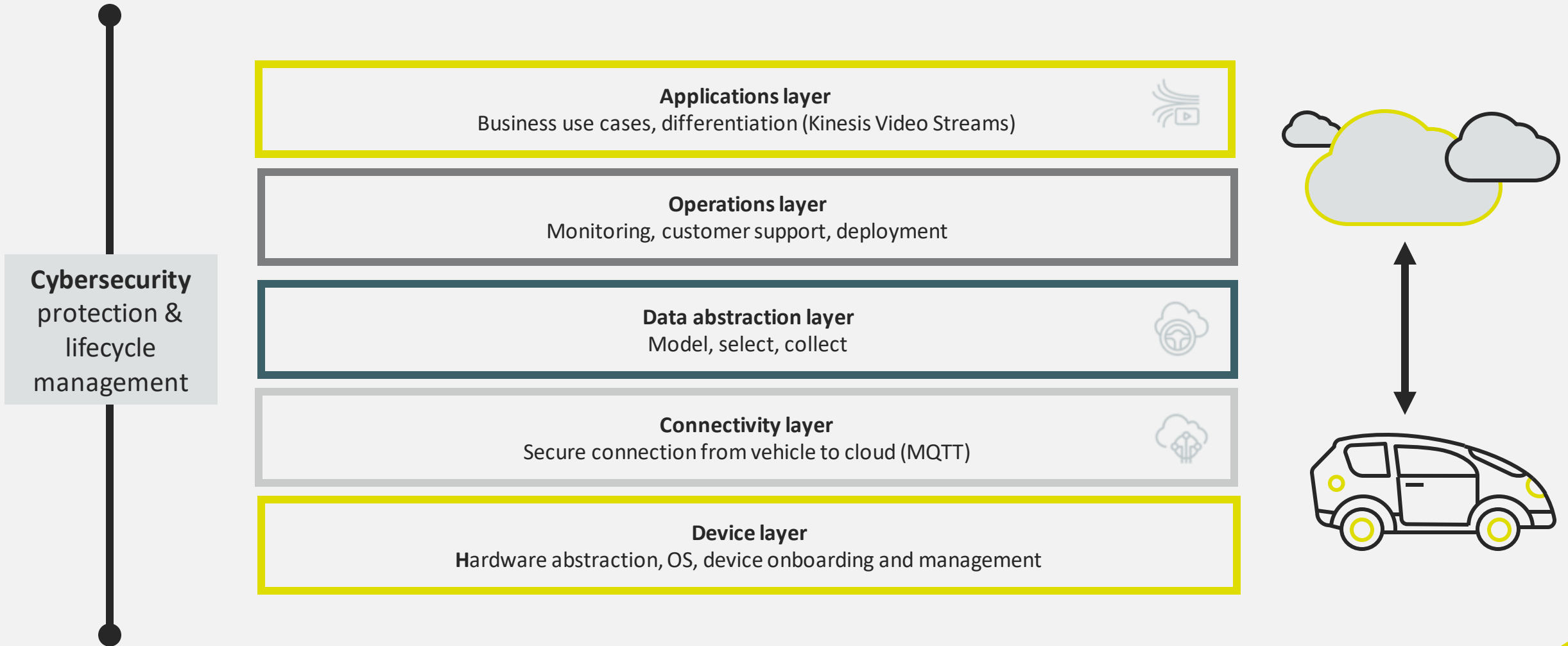


Enter Big Loop

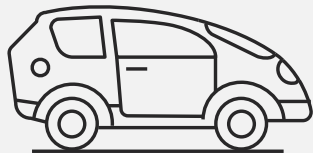
- 1 Vehicle Sensors provide data
- 2 An algorithm selects relevant data to be transferred to the Cloud
- 3 Data is collected in the Cloud
- 4 Variants of the situation are generated in Cloud
- 5 Neural Network is retrained
- 6 New software version is created
- 7 The new SW release is uploaded over the air to the vehicle and activated



Connected Vehicle platform architectural layers



AWS IoT Core is the optimal MQTT message broker for Connected Vehicle platforms



AWS IoT Core is a **fully managed** service that securely connects **millions of vehicles** and routes **trillions of messages** to AWS

Millions

Devices connected to the cloud

Trillions

Messages routed to and from the cloud every month

99.9%+

AWS IoT Core uptime Service Level Agreement

AWS IoT FleetWise



Standardized vehicle data

Analyze fleet-wide vehicle data by creating a common data format using virtual vehicle modeling



Intelligent data collection

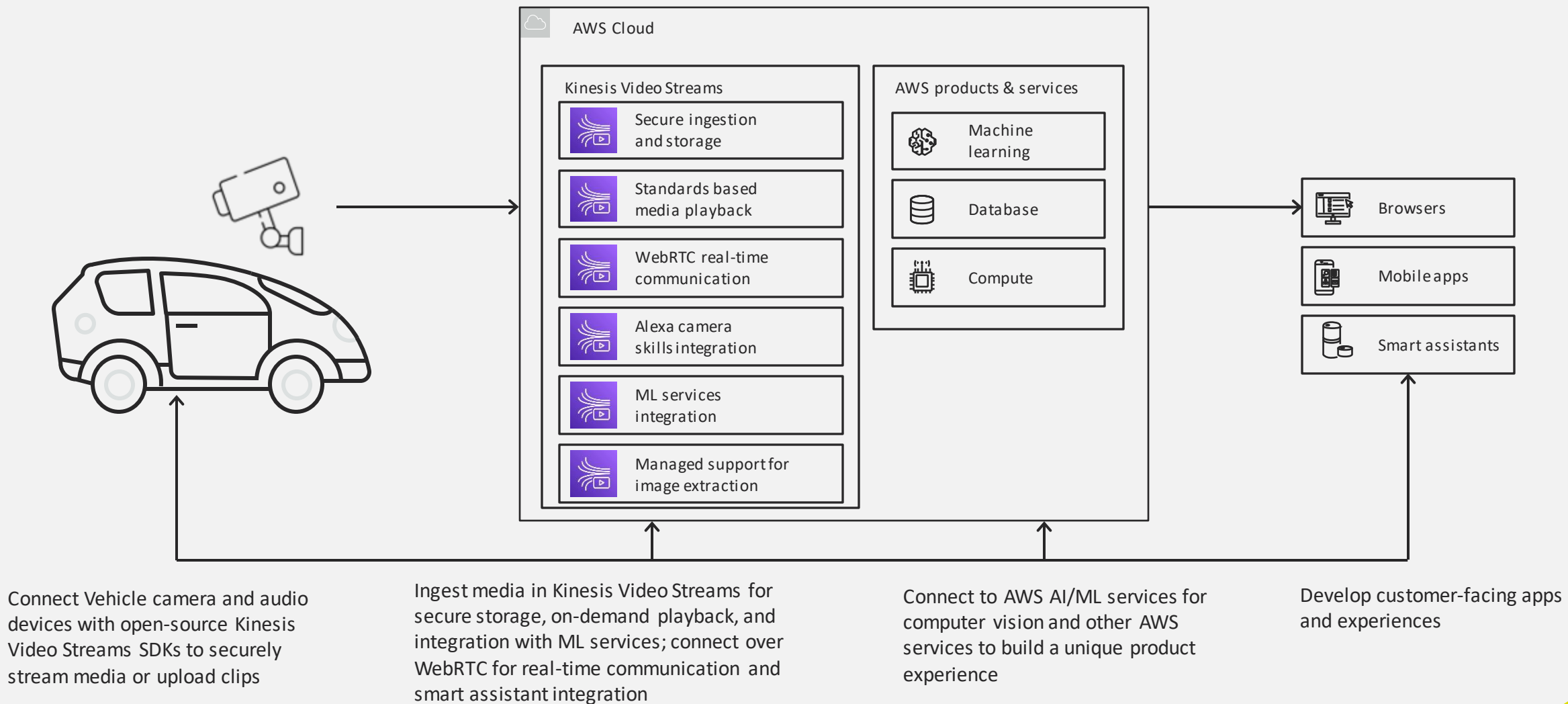
Improve data relevance with time- and event-based data collection campaigns that give access to more useful data in the cloud



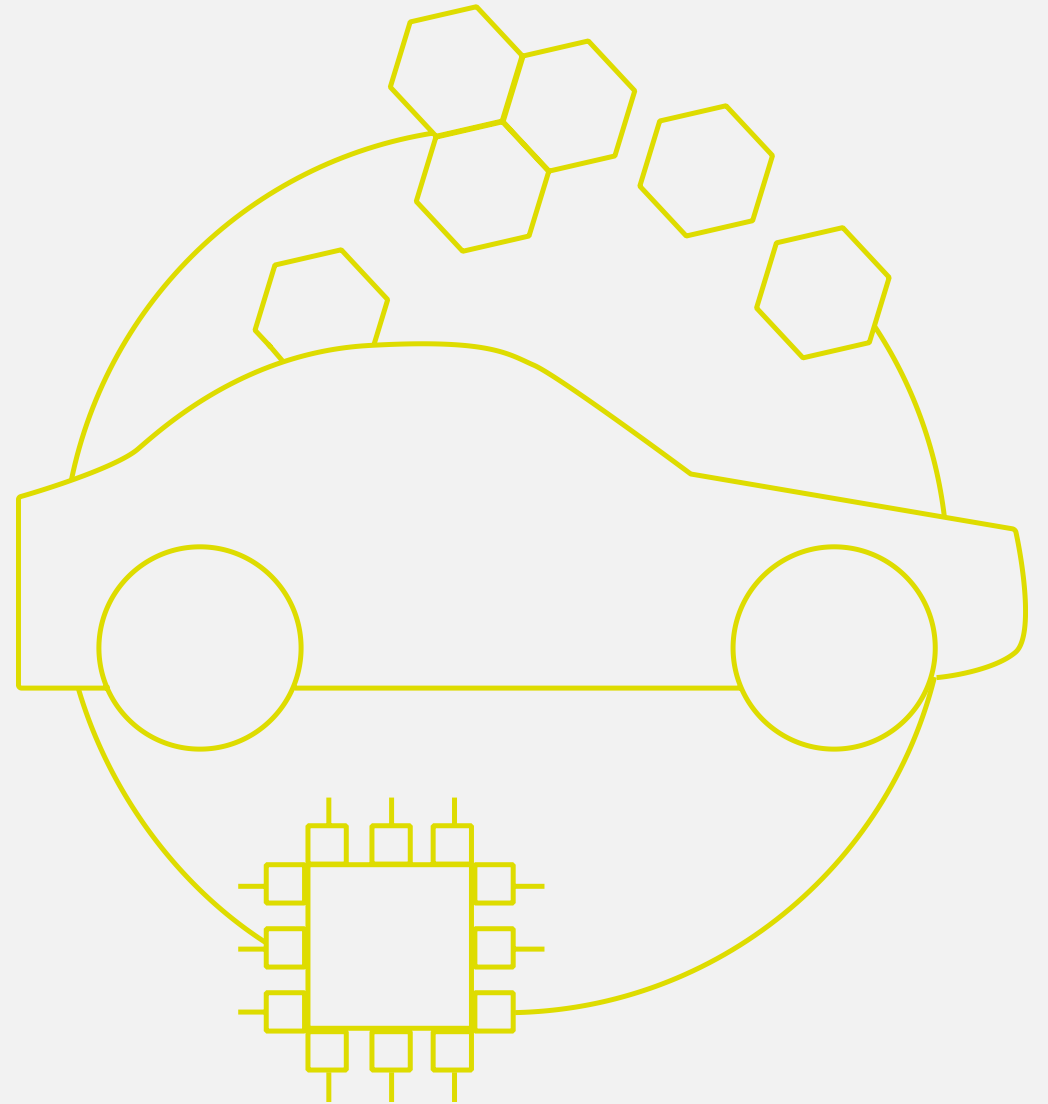
Near real-time data

Take quick, corrective action by notifying teams when problems occur

Kinesis Video Streams

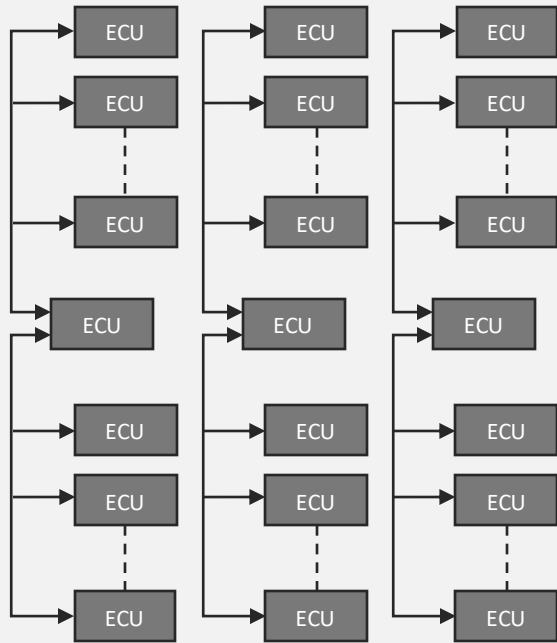


Automotive is a
“Big Loop”
that requires
“Parity”



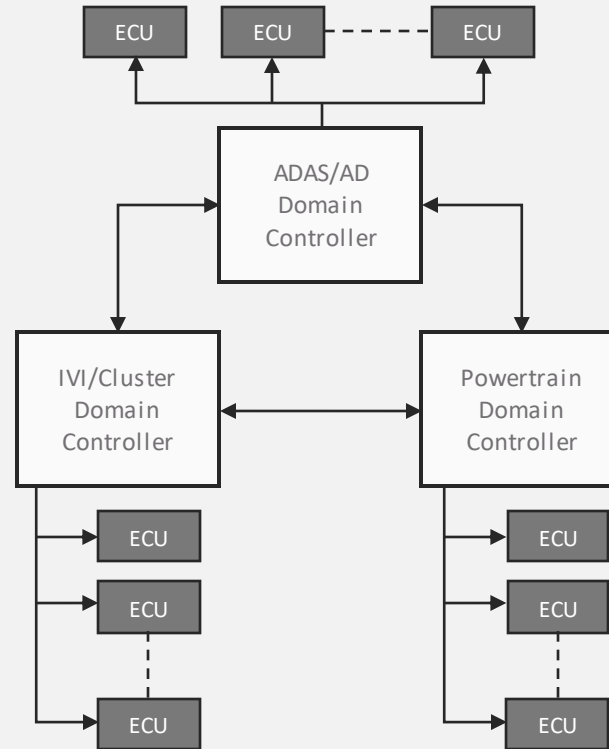
E/E Architectural Evolution

Traditional Architecture



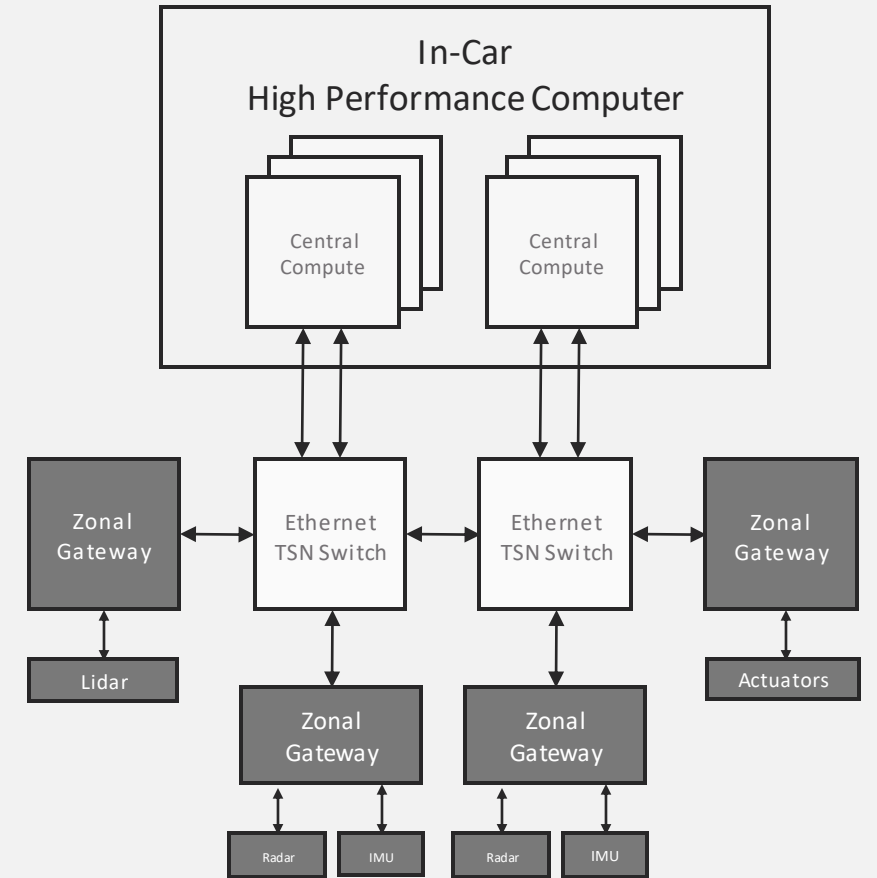
Legacy

Domain Architecture



Today

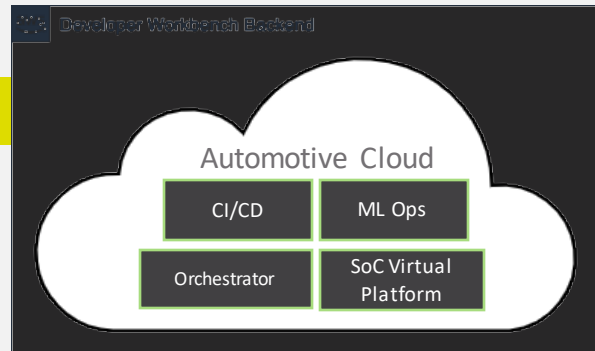
Zonal Architecture



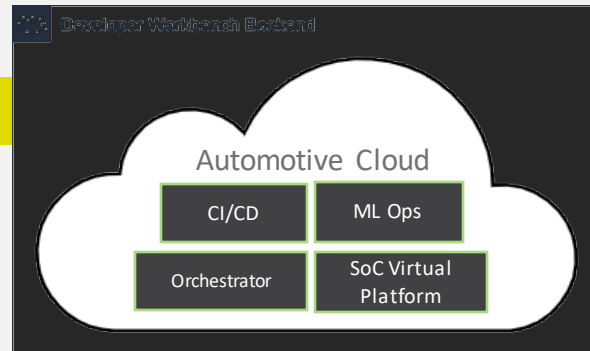
Next Generation

Cloud-Native In Automotive Is A Journey

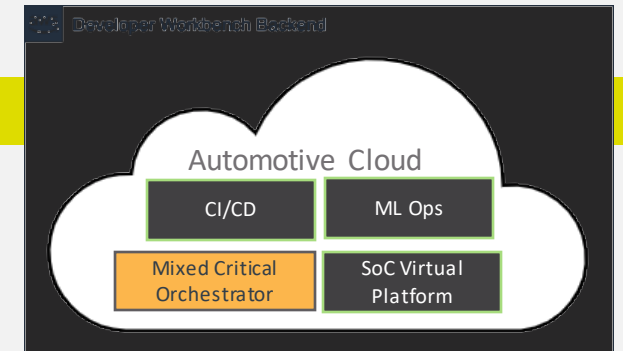
Monolithic Automotive Stack



Microservices Based Automotive Stack

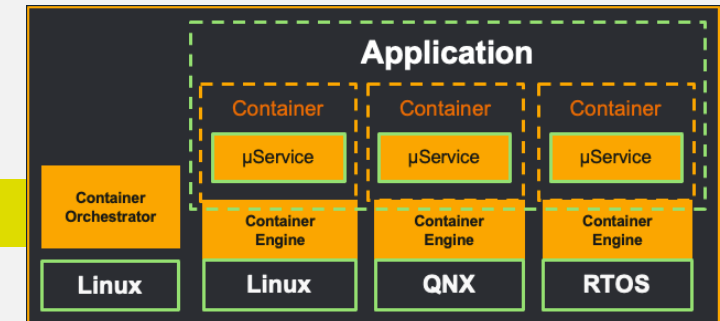
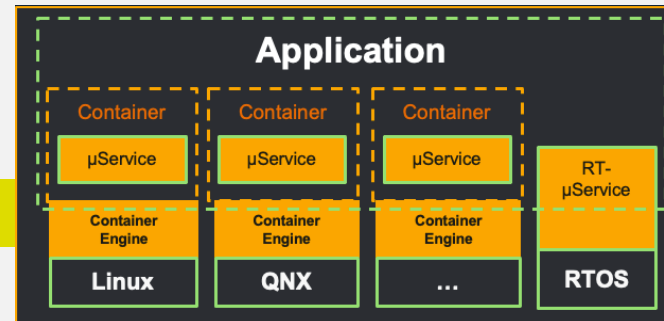
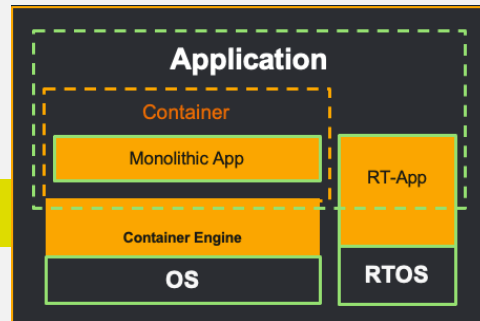


Microservices Based Automotive Stack With Mixed Criticality Orchestrator



Cloud-Native Development

- DevOps / MLOps
- Continuous Delivery
- Microservices based approach
- Containers for SW parity
- Environmental parity at multiple levels



AWS Graviton

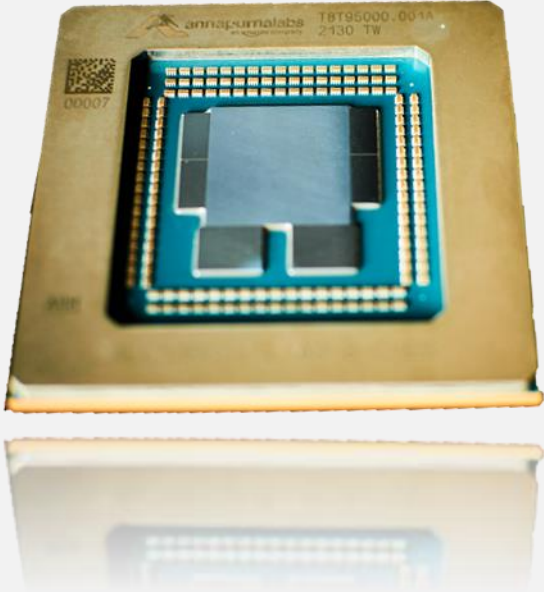
Graviton
2018



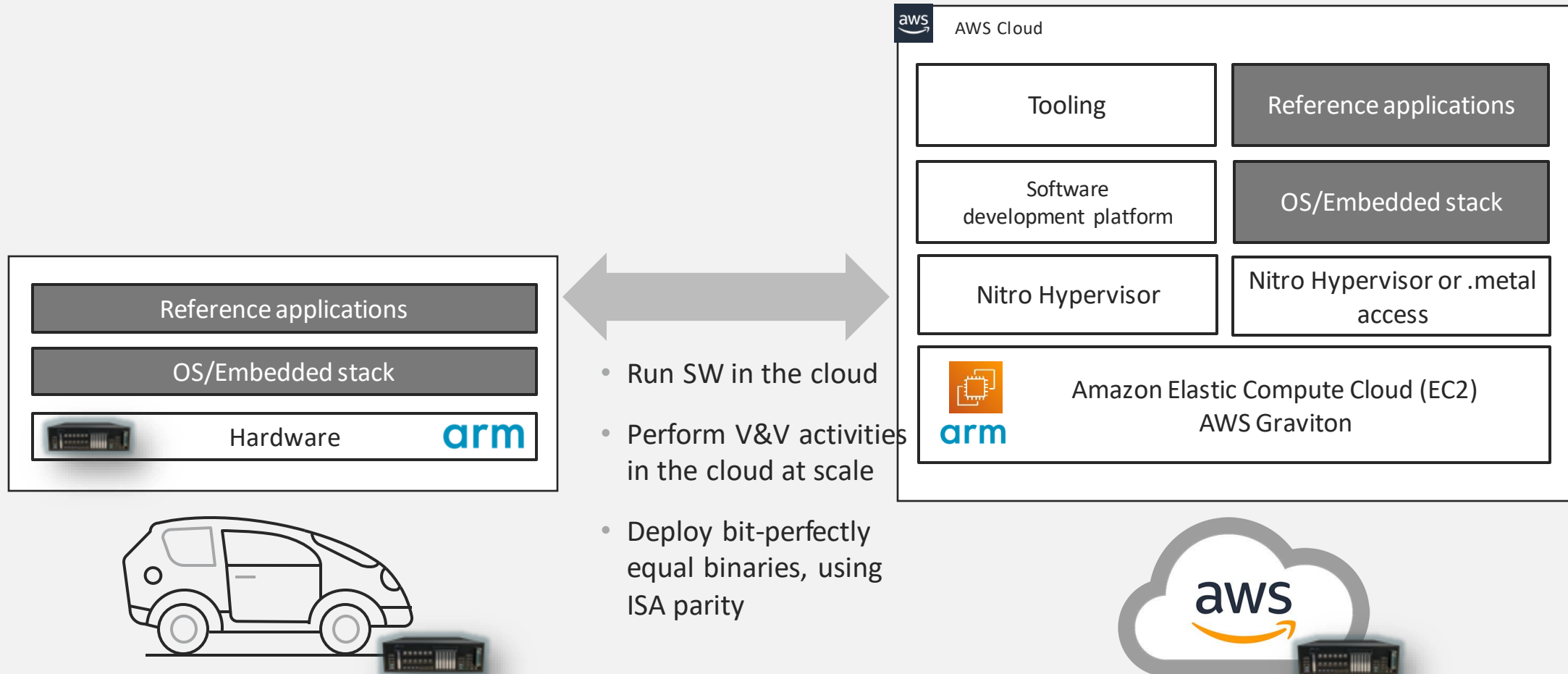
Graviton2
2019



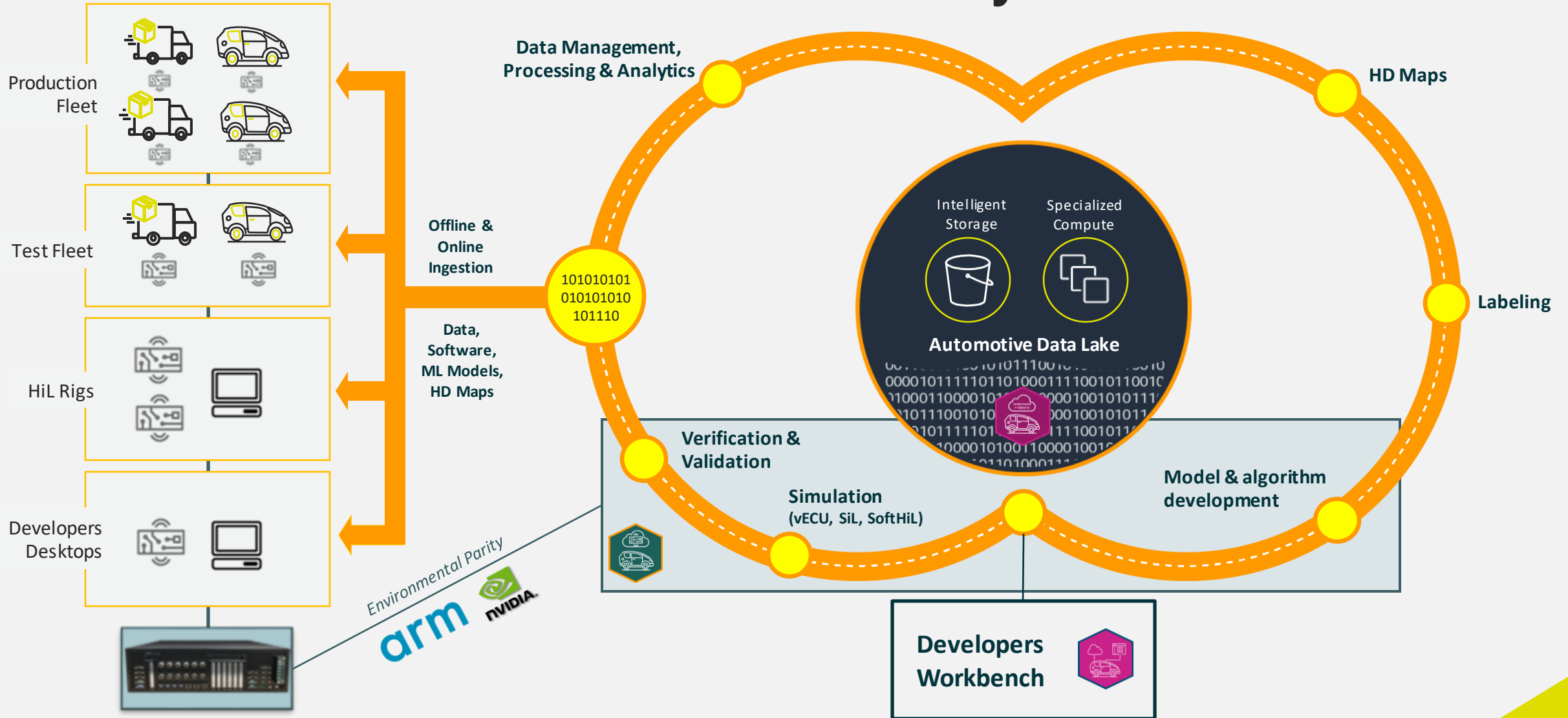
Graviton3
2021



Environmental parity with AWS Cloud and Arm

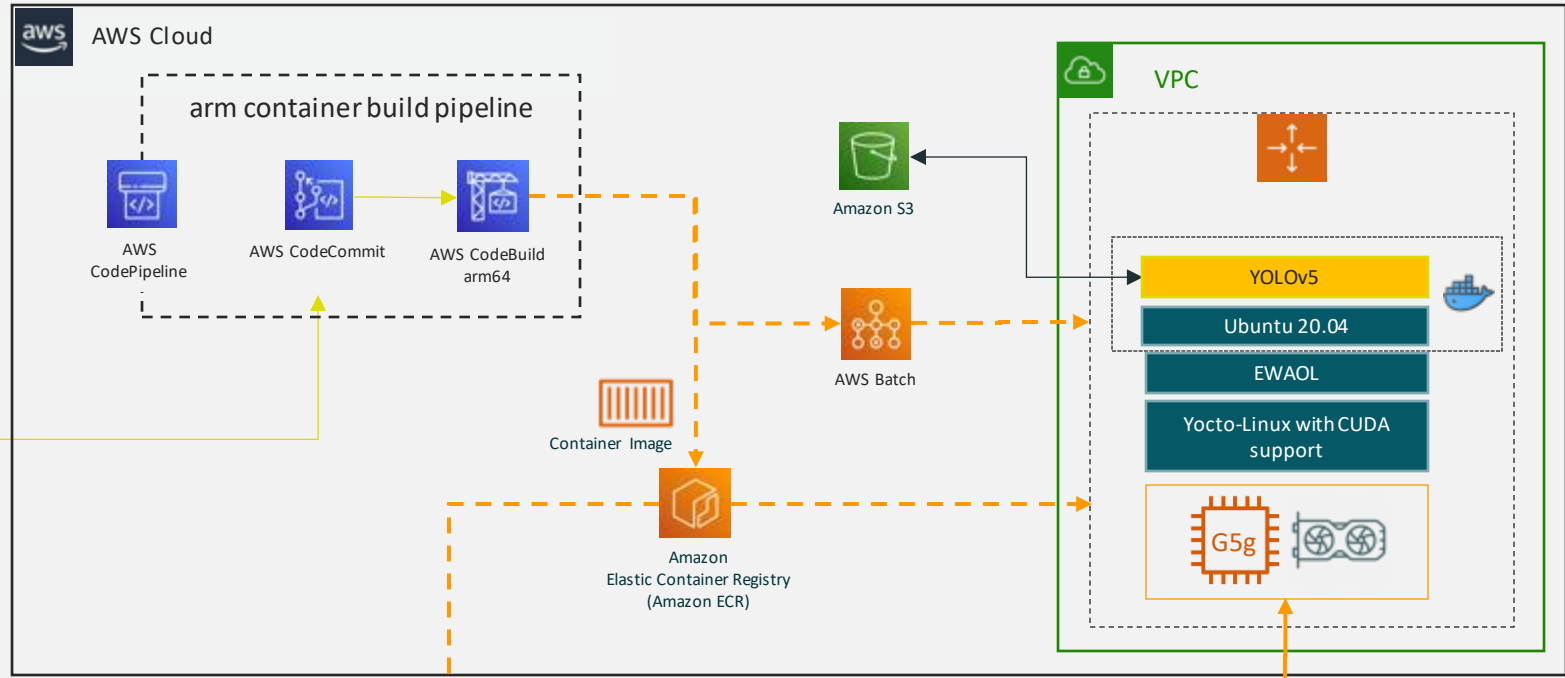


Native Environment Execution Parity

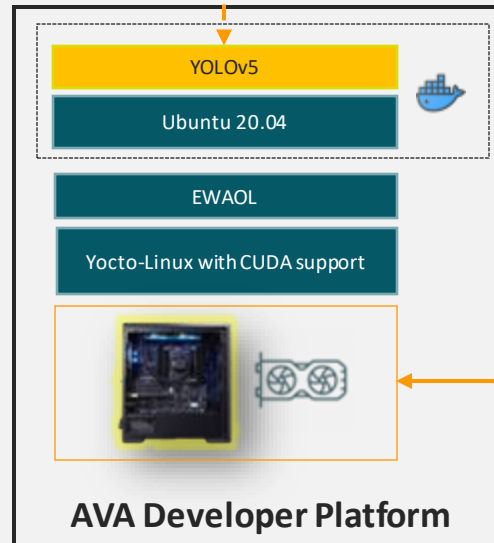




Git push



Docker Pull



- Graviton2 (Neoverse N1 Cores)
- NVIDIA T4G GPU

- Ampere Altra (Neoverse N1 Cores)
- NVIDIA RTX4000 GPU

CES 2022

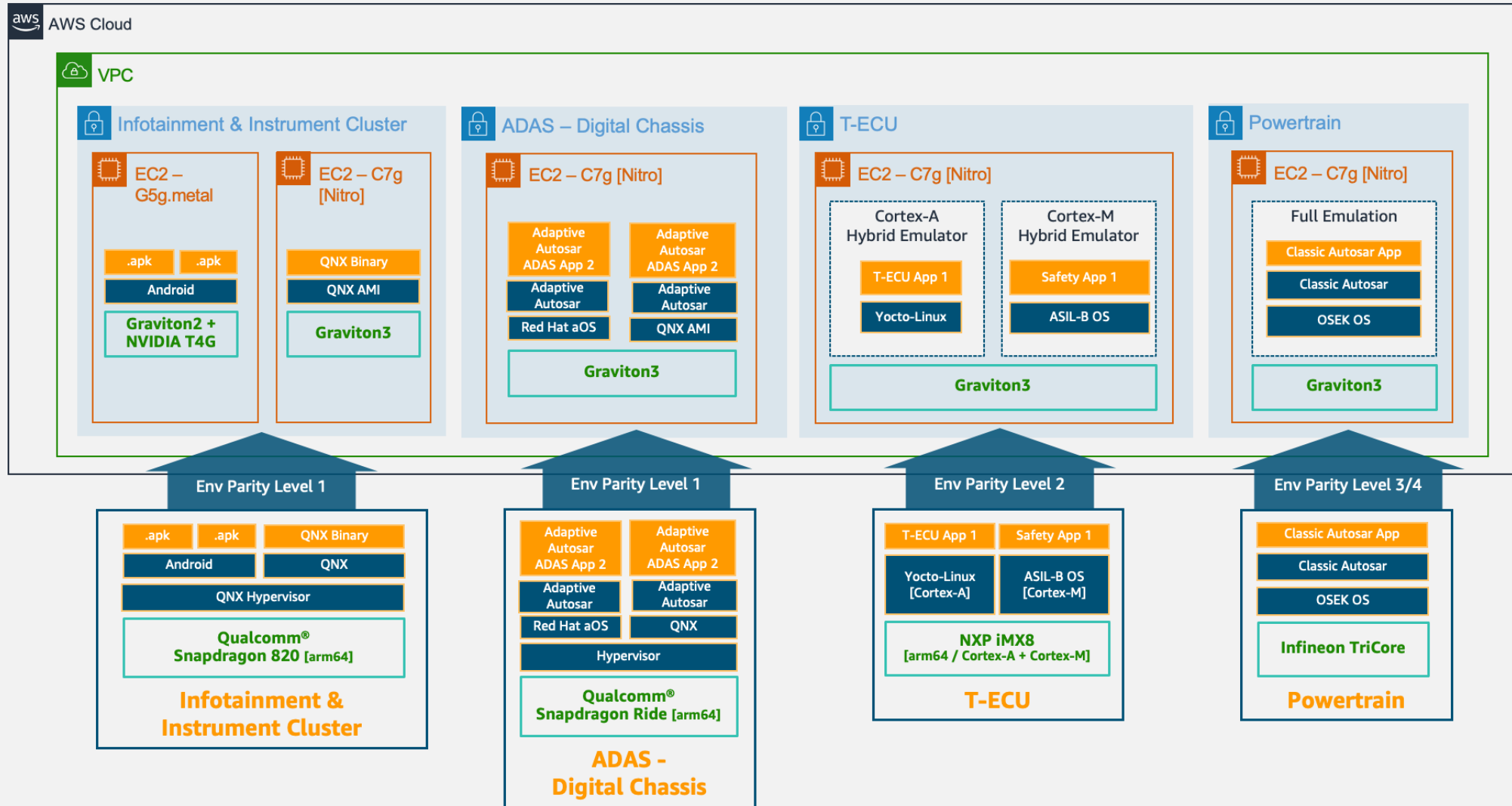
Demo Architecture

Environmental Parity

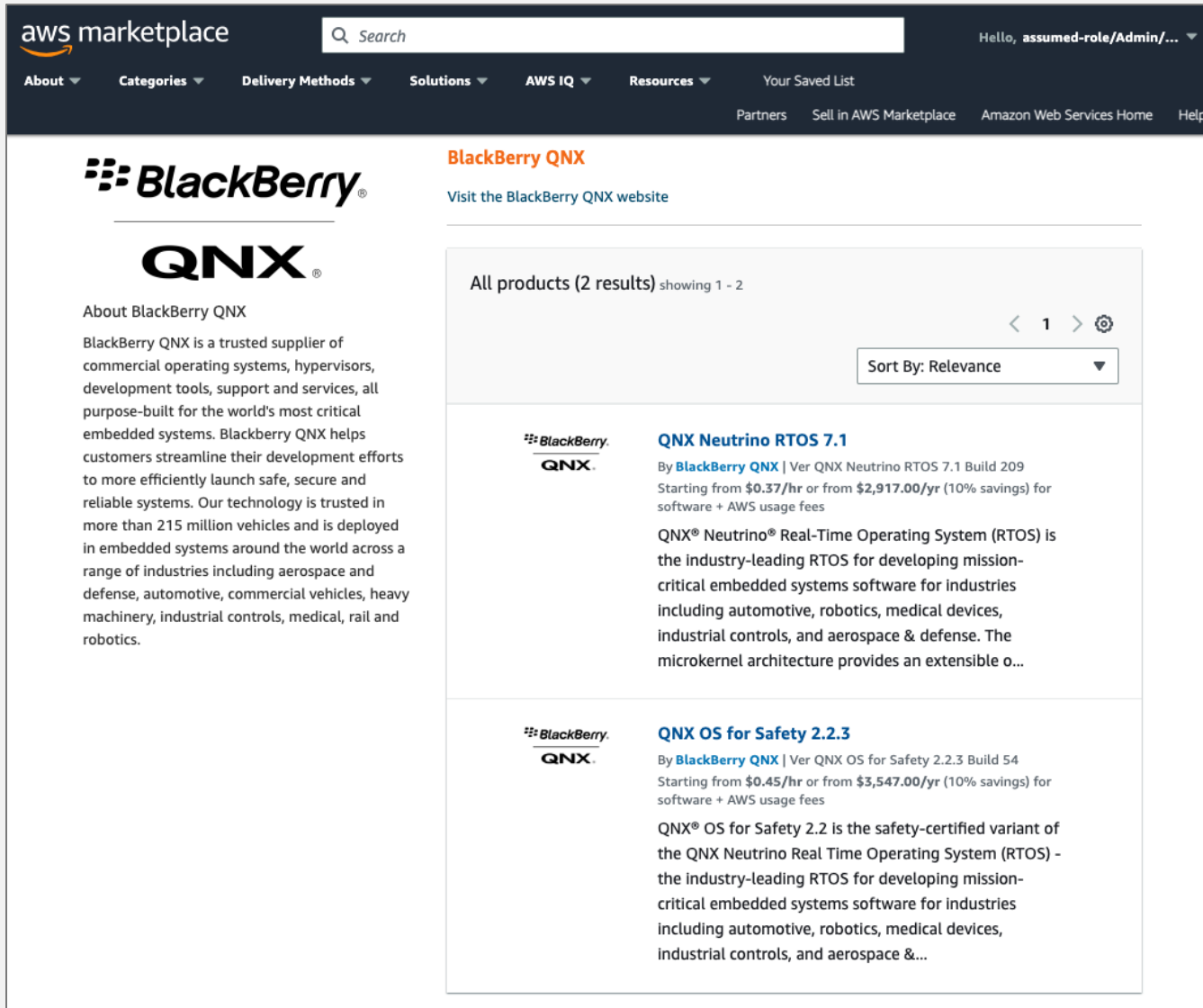
Developer Persona		Environmental Parity	Example Usage	Enabled By
Application Developers	Current SOCs	ISA Parity	Improved DX enabling fast dev feedback/SIL running same binary in cloud/edge	Arm-based cloud instance & Arm-based Automotive Edge
	Future SOCs	CPU Architecture Parity	Software dev/test using extended Arm architecture features (e.g. Algo dev)	Arm CPU Models
Platform Developers		SoC Parity	Software dev/test needing SoC base features and/IO access (e.g. BSP, etc.)	Arm SystemReady Compliant SoC Models and Devices
System Developers		System Parity	System dev, integration, verification and validation incl. FuSA, Realtime	Digital Twins

<https://armkeil.blob.core.windows.net/developer/Files/pdf/white-paper/arm-aws-edge-environmental-parity-wp.pdf>

Environmental Parity Examples



Example of an Automotive-Native AMI: QNX



aws marketplace

Search

Hello, assumed-role/Admin/...

About Categories Delivery Methods Solutions AWS IQ Resources Your Saved List

Partners Sell in AWS Marketplace Amazon Web Services Home Help

BlackBerry QNX

Visit the BlackBerry QNX website

QNX

About BlackBerry QNX

BlackBerry QNX is a trusted supplier of commercial operating systems, hypervisors, development tools, support and services, all purpose-built for the world's most critical embedded systems. Blackberry QNX helps customers streamline their development efforts to more efficiently launch safe, secure and reliable systems. Our technology is trusted in more than 215 million vehicles and is deployed in embedded systems around the world across a range of industries including aerospace and defense, automotive, commercial vehicles, heavy machinery, industrial controls, medical, rail and robotics.

All products (2 results) showing 1 - 2

Sort By: Relevance

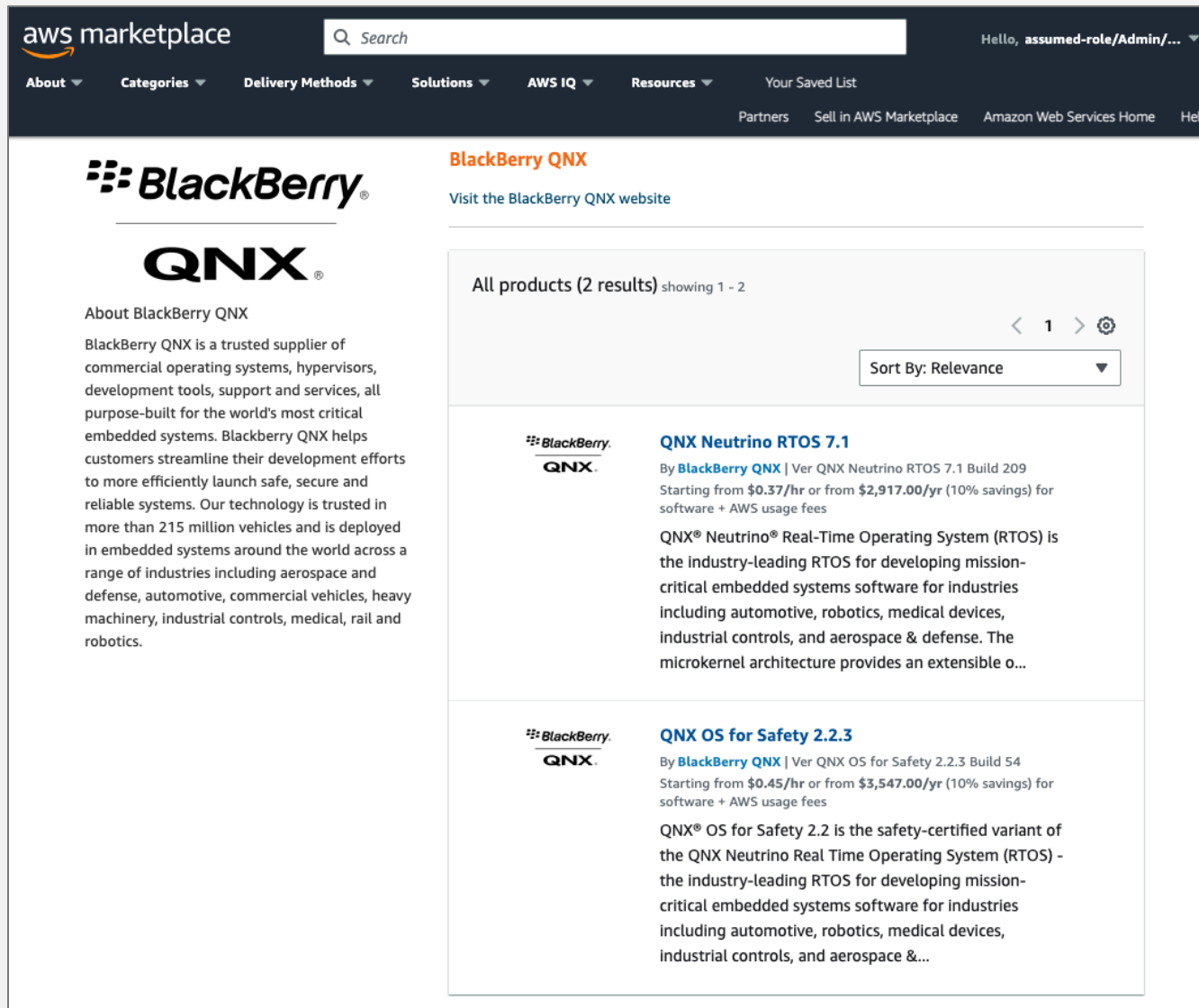
- QNX Neutrino RTOS 7.1**
By BlackBerry QNX | Ver QNX Neutrino RTOS 7.1 Build 209
Starting from \$0.37/hr or from \$2,917.00/yr (10% savings) for software + AWS usage fees
QNX® Neutrino® Real-Time Operating System (RTOS) is the industry-leading RTOS for developing mission-critical embedded systems software for industries including automotive, robotics, medical devices, industrial controls, and aerospace & defense. The microkernel architecture provides an extensible o...
- QNX OS for Safety 2.2.3**
By BlackBerry QNX | Ver QNX OS for Safety 2.2.3 Build 54
Starting from \$0.45/hr or from \$3,547.00/yr (10% savings) for software + AWS usage fees
QNX® OS for Safety 2.2 is the safety-certified variant of the QNX Neutrino Real Time Operating System (RTOS) - the industry-leading RTOS for developing mission-critical embedded systems software for industries including automotive, robotics, medical devices, industrial controls, and aerospace &...



```
ssh
marzanis@08f8bc643200 Keys % ssh -i "MrzCloudDef.pem" root@ec2-44-234-66-223.us-west-2.compute.amazonaws.com
# uname -a
QNX localhost 7.1.0 2022/07/12-18:14:46EDT AWS_Graviton2_t4g.xlarge aarch64le
#
```

[Workshop Link](#)

Example of an Automotive-Native AMI: QNX

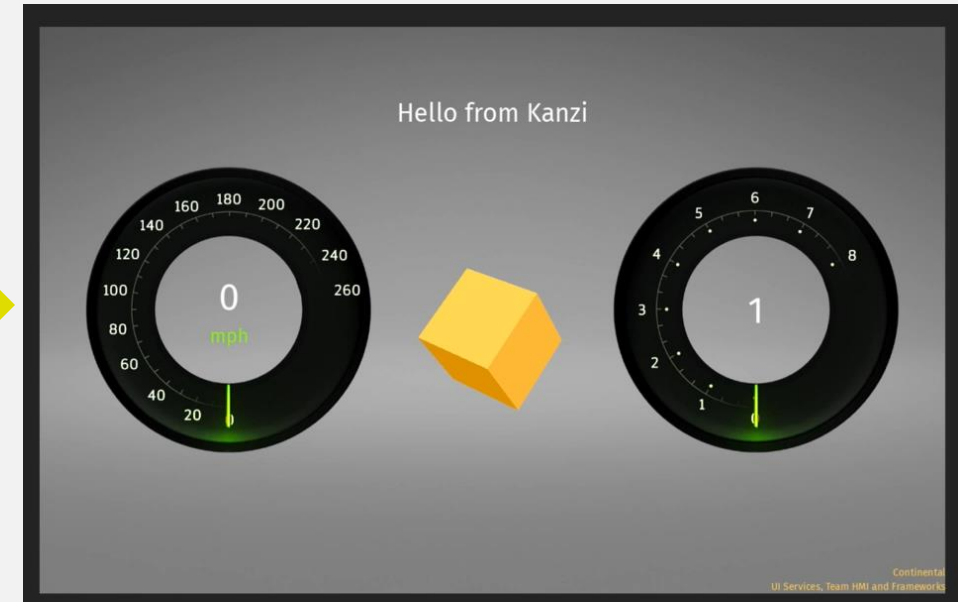


The screenshot shows the AWS Marketplace interface. At the top, there's a search bar and navigation links. The main content area features the BlackBerry QNX logo and a list of products. Two products are visible:

- QNX Neutrino RTOS 7.1**: By BlackBerry QNX | Ver QNX Neutrino RTOS 7.1 Build 209. Starting from \$0.37/hr or from \$2,917.00/yr (10% savings) for software + AWS usage fees. Description: QNX® Neutrino® Real-Time Operating System (RTOS) is the industry-leading RTOS for developing mission-critical embedded systems software for industries including automotive, robotics, medical devices, industrial controls, and aerospace & defense. The microkernel architecture provides an extensible o...
- QNX OS for Safety 2.2.3**: By BlackBerry QNX | Ver QNX OS for Safety 2.2.3 Build 54. Starting from \$0.45/hr or from \$3,547.00/yr (10% savings) for software + AWS usage fees. Description: QNX® OS for Safety 2.2 is the safety-certified variant of the QNX Neutrino Real Time Operating System (RTOS) - the industry-leading RTOS for developing mission-critical embedded systems software for industries including automotive, robotics, medical devices, industrial controls, and aerospace &...



```
ssh
marzani@008f8bc643200: Keys % ssh -i "MrzCloudDef.pem" root@ec2-44-234-66-223.us-west-2.compute.amazonaws.com
# uname -a
QNX localhost 7.1.0 2022/07/12-18:14:46EDT AWS_Graviton2_t4g.xlarge aarch64le
#
```



Example of an Automotive-Native AMI: Ubuntu RT*

aws marketplace Hello, assumed-role/Admin/...

About Categories Delivery Methods Solutions AWS IQ Resources Your Saved List

Become a Channel Partner Sell in AWS Marketplace Amazon Web Services Home Help



Ubuntu Pro 22.04 LTS with Real-time Kernel

By: [Canonical Group Limited](#)

Latest Version: Ubuntu Pro Realtime 22.04 ARM 20230823

The official real-time Ubuntu Linux, with ten years of maintenance for the time-bound workloads of modern enterprises. Perfect for development in a digital-twin environment of Ubuntu RTK on real hardware

[Show more](#)

Typical Total Price
\$0.072/hr

Total pricing per instance for services hosted on c6g.large in US East (N. Virginia). [View Details](#)

Linux/Unix

Continue to Subscribe

Save to List



```
ubuntu@ip-172-31-51-159:~$ uname -a
Linux ip-172-31-51-159 5.15.0-1043-realtime #48-Ubuntu SMP PREEMPT_RT
Thu Jul 13 20:54:21 UTC 2023 aarch64 aarch64 aarch64 GNU/Linux
ubuntu@ip-172-31-51-159:~$
```

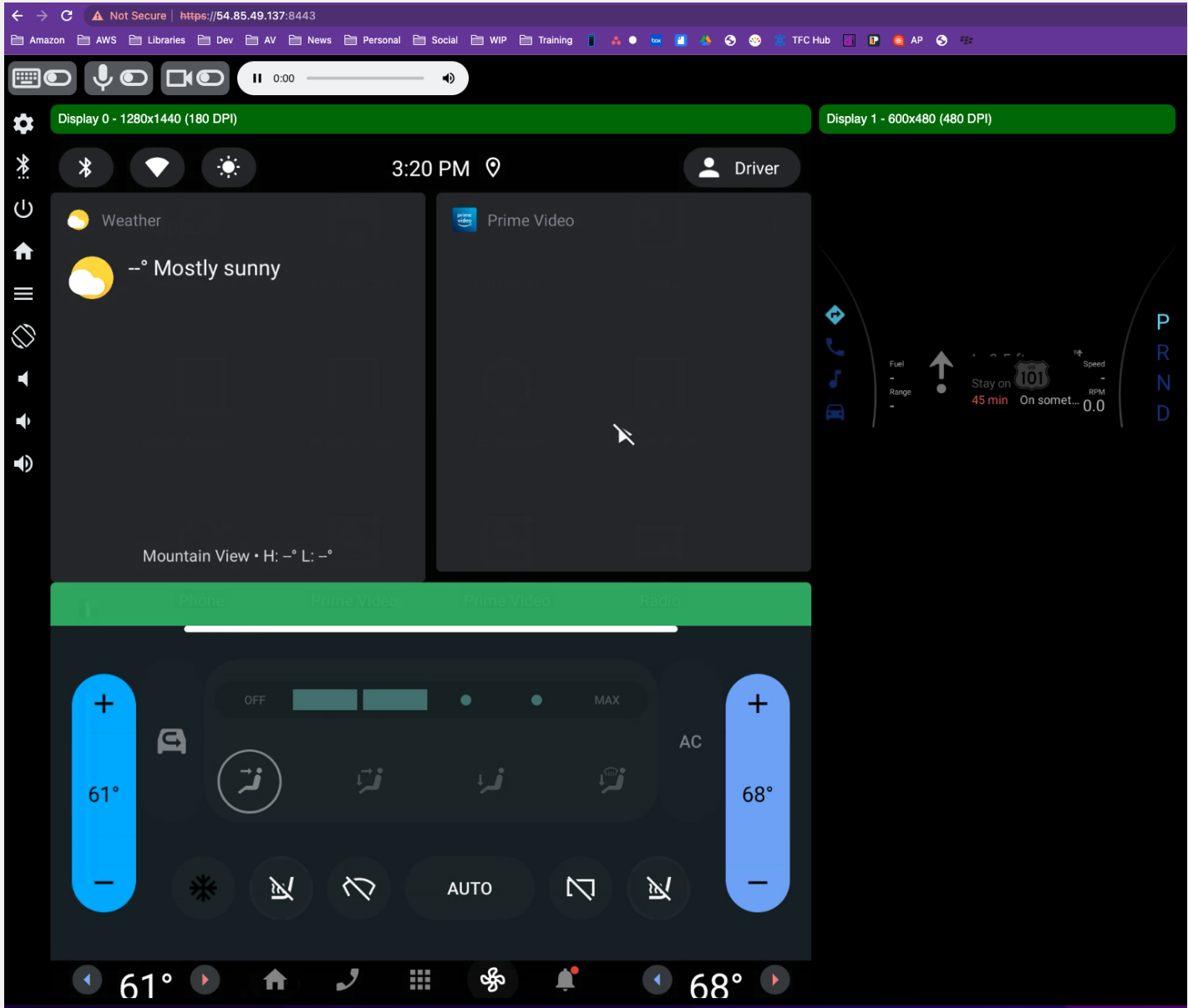
i-086b473d312a4014b (Ubuntu RT) ✕

PublicIPs: 18.246.8.107 PrivateIPs: 172.31.51.159

Product Overview

"Real-time Ubuntu Pro is an Ubuntu 22.04 LTS Jammy Jellyfish server AMI that enables the real-time kernel option on boot, running with the 5.15-rt patch set implemented.

Android on Graviton



Customer implementation @ IAA Munich 2023



Android OS in vehicle



Same Android OS image running on Graviton, bi-directional UI streamed in browser through WebRTC

...happen, Today
...the Vehicle Together
...the Road to the Cloud.



Continental
Video- & Tonaufnahme
Das Interaktion über die Video- & Tonaufnahme an unserem
Stand ist nur für den Zweck der Dokumentation und nicht für andere
Zwecke vorgesehen. Die Aufnahme wird nicht veröffentlicht.

Cabin digital twin on AWS and QNX AMI



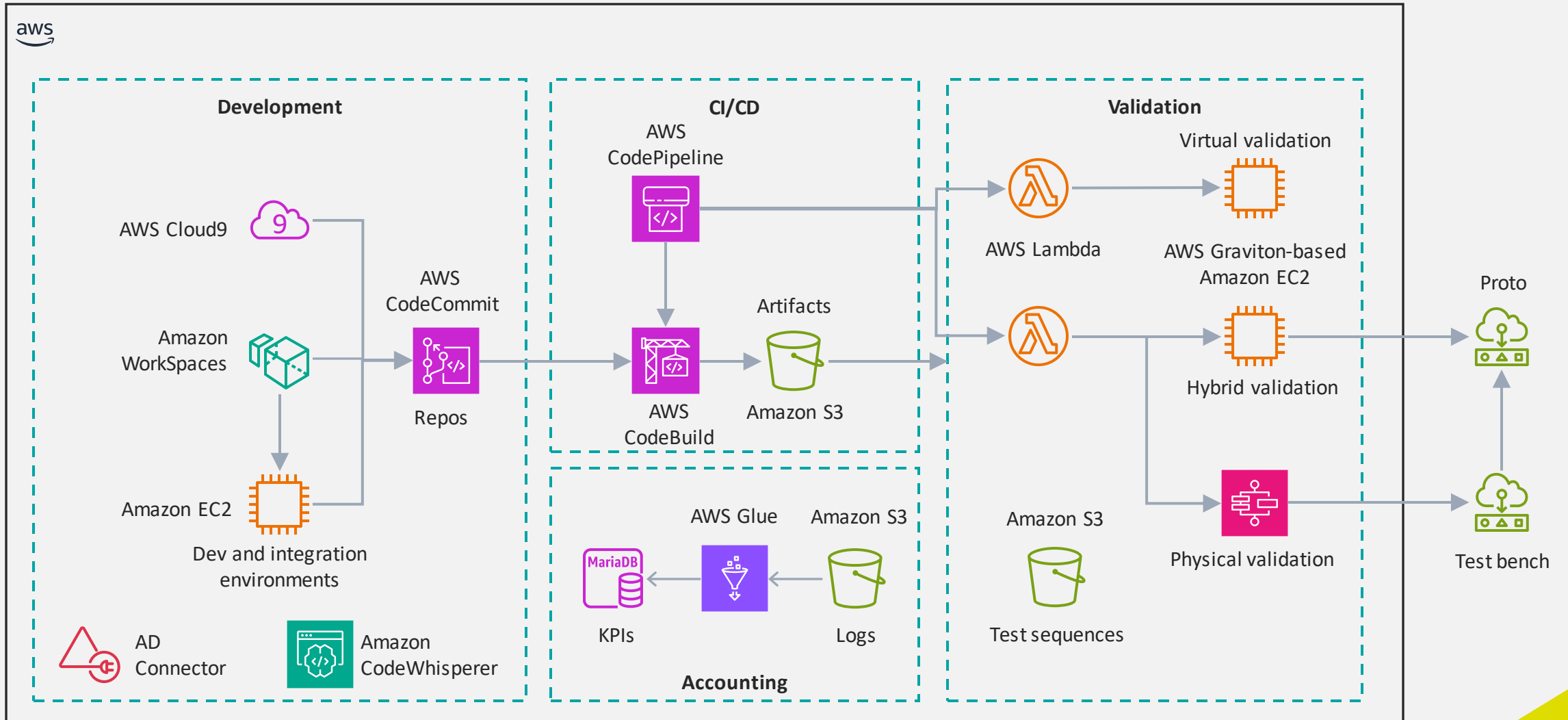
Marelli press release: bit.ly/3qVCTNB

“By combining the strengths of Marelli and AWS services, we can deliver the driving experience consumers want while ensuring security and reliability. Marelli’s digital twin empowers software engineers to reduce development time by up to 70% and realize cost savings on prototypes up to 30%, delivering software evolutions more efficiently and cost-effectively.”

Yannick Hoyau

VP and Head of Engineering & Innovation,
Marelli Electronic Systems

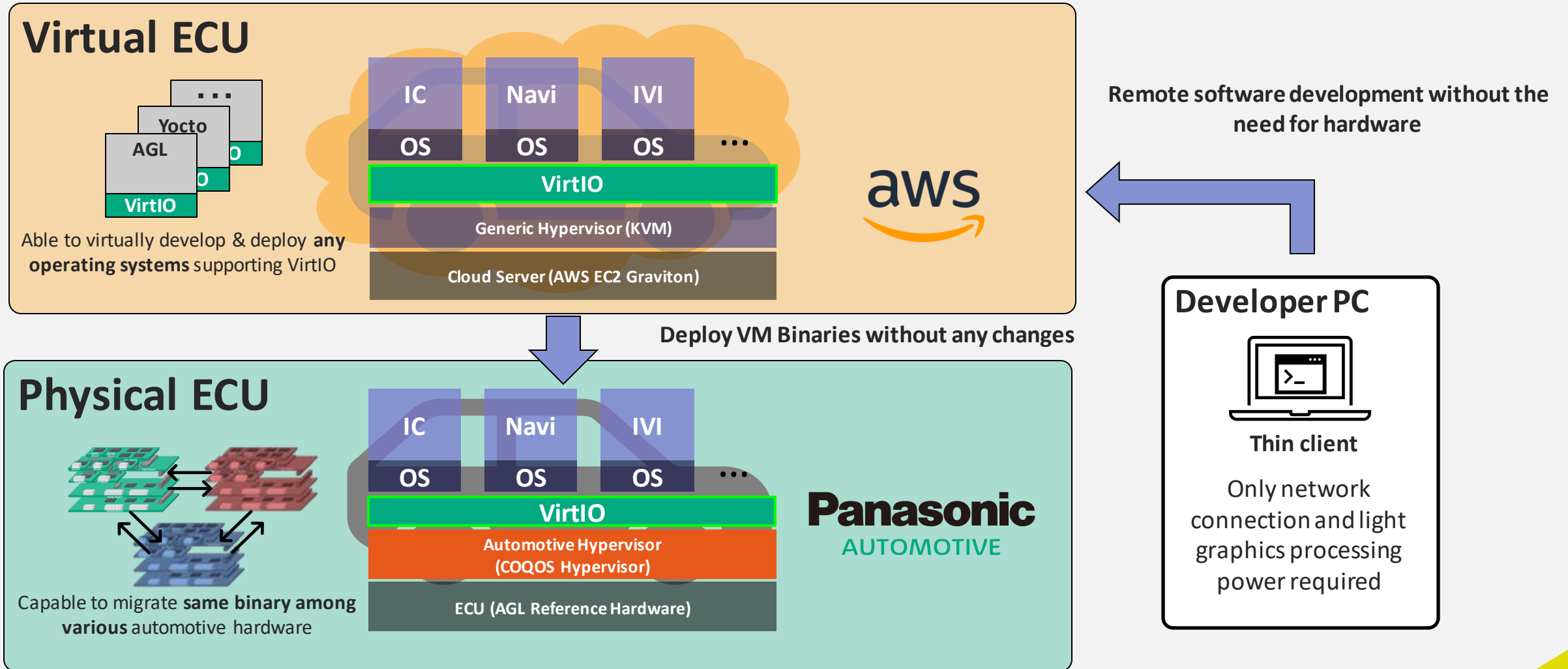
Marelli DigiMate project



The Next Level: AGL CES2023 Demo from Panasonic and AWS



VirtIO Abstraction for OS portability



Example of an “Hybrid-Emulation”: Arm Virtual HW

The screenshot shows the AWS IoT console interface for a device named 'feline-brochure' (iMX8m Arm Cortex Complex | 2.2.0 | 5.10.72-2.2.0). The console displays system logs and a terminal window. The terminal window shows the output of the 'lscpu' command, which identifies the processor as Cortex-A53. The device is named 'feline-brochure' and is an iMX8m Arm Cortex Complex.

```
sh-5.1# uname -a
Linux imx8mpevk 5.10.72-lts-5.10.y+ga68e31b63f86 #1 SMP PREEMPT Tue Nov 23 06:02:20 UTC 2021 aarch64 aarch64 aarch64 GNU/Linux
sh-5.1# lscpu
Architecture: aarch64
CPU op-mode(s): 32-bit, 64-bit
Byte Order: Little Endian
CPU(s): 4
On-line CPU(s) list: 0-3
Thread(s) per core: 1
Core(s) per socket: 4
Socket(s): 1
NUMA node(s): 1
Vendor ID: ARM
Model:
Model name: Cortex-A53
Stepping: r0p2
CPU max MHz: 1800.0000
CPU min MHz: 1200.0000
BogoMIPS: 243.75
NUMA node0 CPU(s): 0-3
Vulnerability Itlb multihit: Not affected
Vulnerability L1tf: Not affected
Vulnerability Mds: Not affected
Vulnerability Meltdown: Not affected
Vulnerability Spec store bypass: Not affected
Vulnerability Spectre v1: Mitigation: user pointer sanitization
Vulnerability Spectre v2: Not affected
Vulnerability Srbds: Not affected
Vulnerability Tsx async abort: Not affected
Flags: fp asimd evtstrm aes pmull sha1 sha2 crc32 atomics fphp asimdhp cpuid asimdrdm lrcpc dcpop asimddp ssbs
sh-5.1#
```

Virtual iMX8 console access

Virtual iMX8 display in Browser

Cortex-A53 “hybrid” emulation on AWS Graviton’s Neoverse

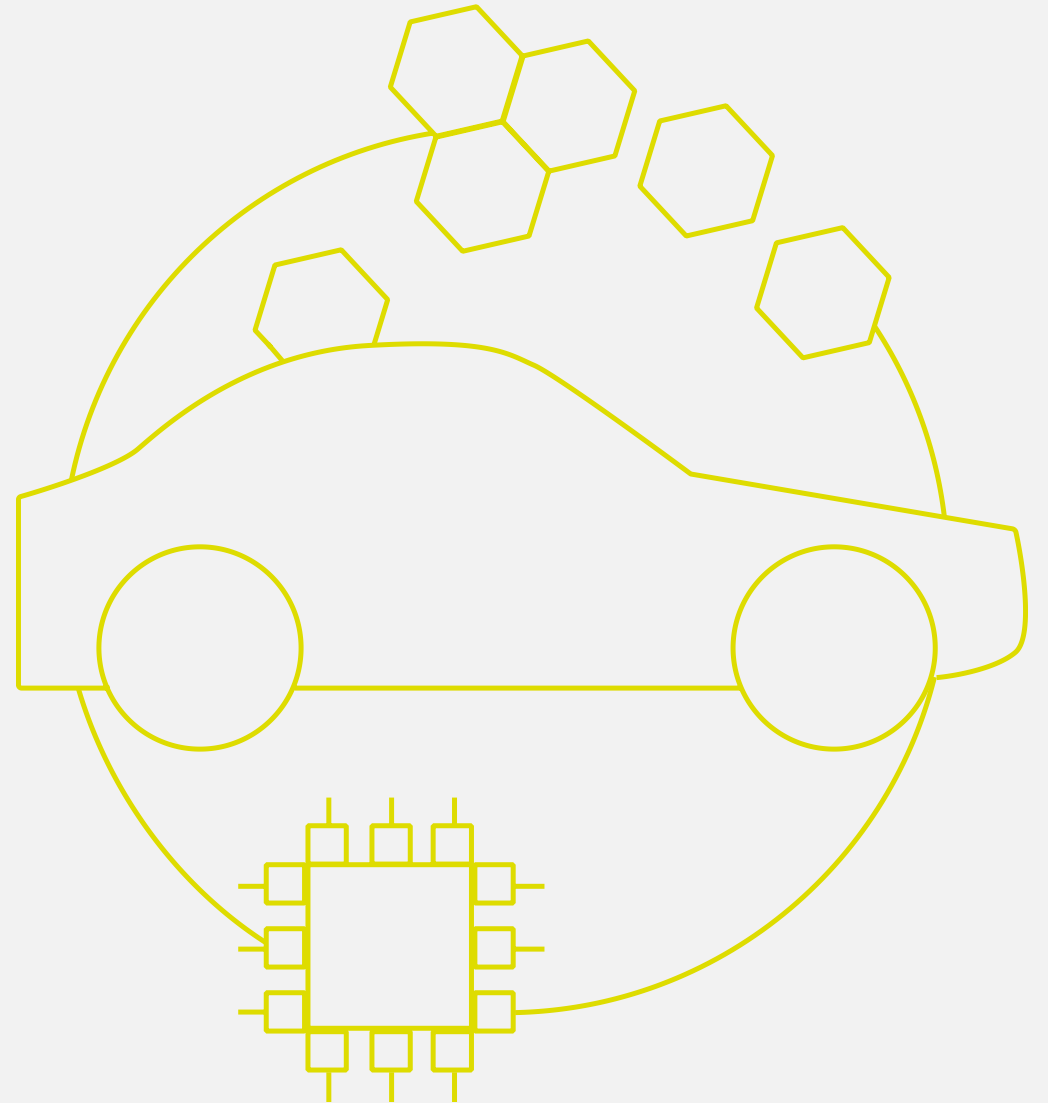
Hybrid Emulation on Arm Virtual HW / Corellium

The screenshot displays the ARM DevStudio interface for an iMX8M Virtual Target. The top navigation bar includes the ARM logo, a search bar, and user information (David Walters). The main content area is divided into a sidebar and a main panel. The sidebar contains navigation options: Connect, CoreTrace, Settings, Console, System Console, Sensors (highlighted), and Snapshots. The main panel is titled 'Sensors' and 'Camera & Microphone'. It features an 'ENABLED' toggle switch, a 'CAMERA' dropdown menu set to 'HD Webcam USB (0c45:6366)', and a 'MICROPHONE' dropdown menu set to 'Default - eMeet Luna (Bluetooth)'. Below these is an 'UPLOAD' section with a file input field containing 'video.hevc' and a URL input field containing 'https://example.com/video.mp4'. At the bottom of the upload section are 'PLAY VIDEO' and 'STOP VIDEO' buttons. The left sidebar also shows a 'LEDs' indicator at the bottom.

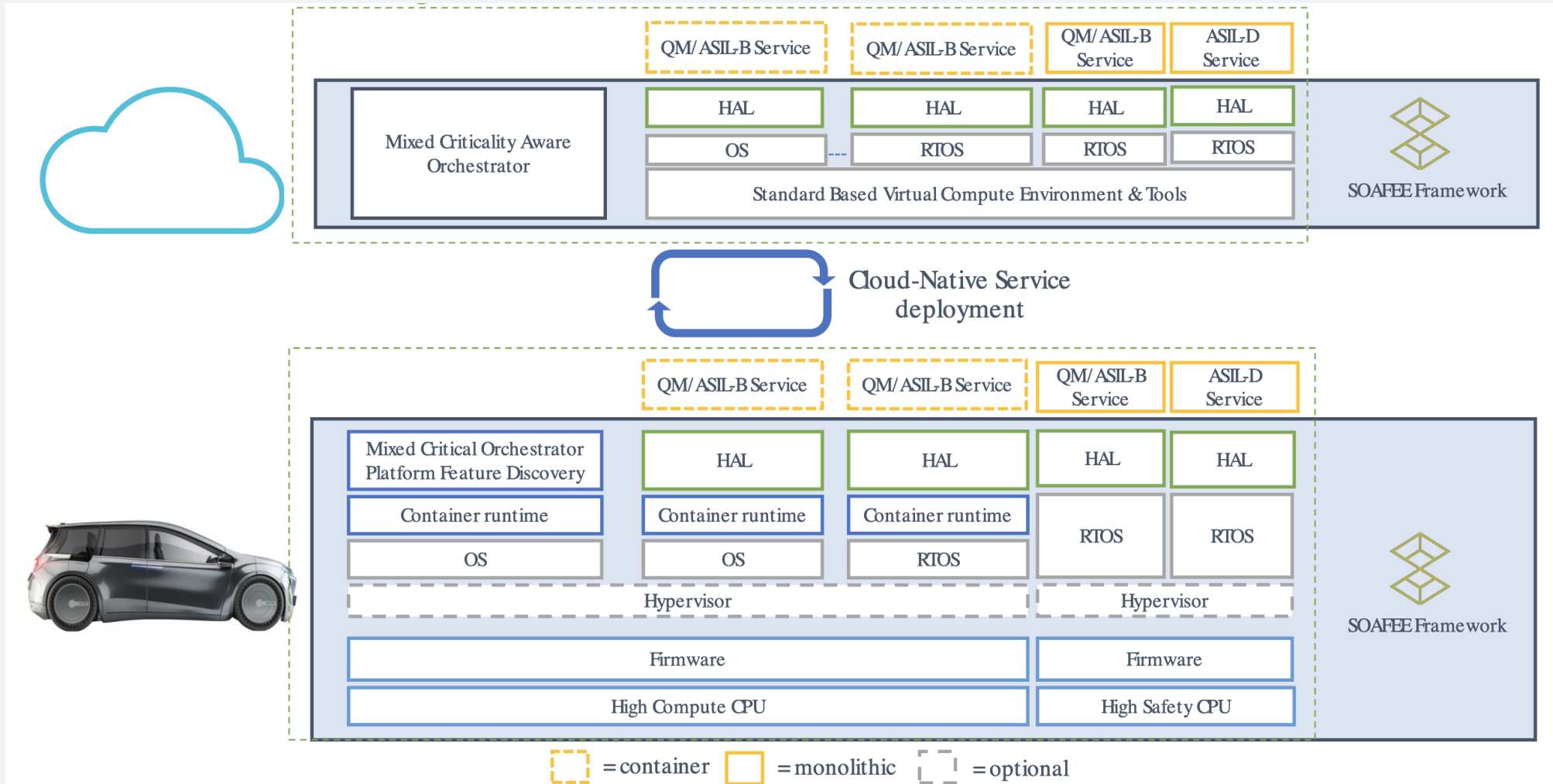
LINK



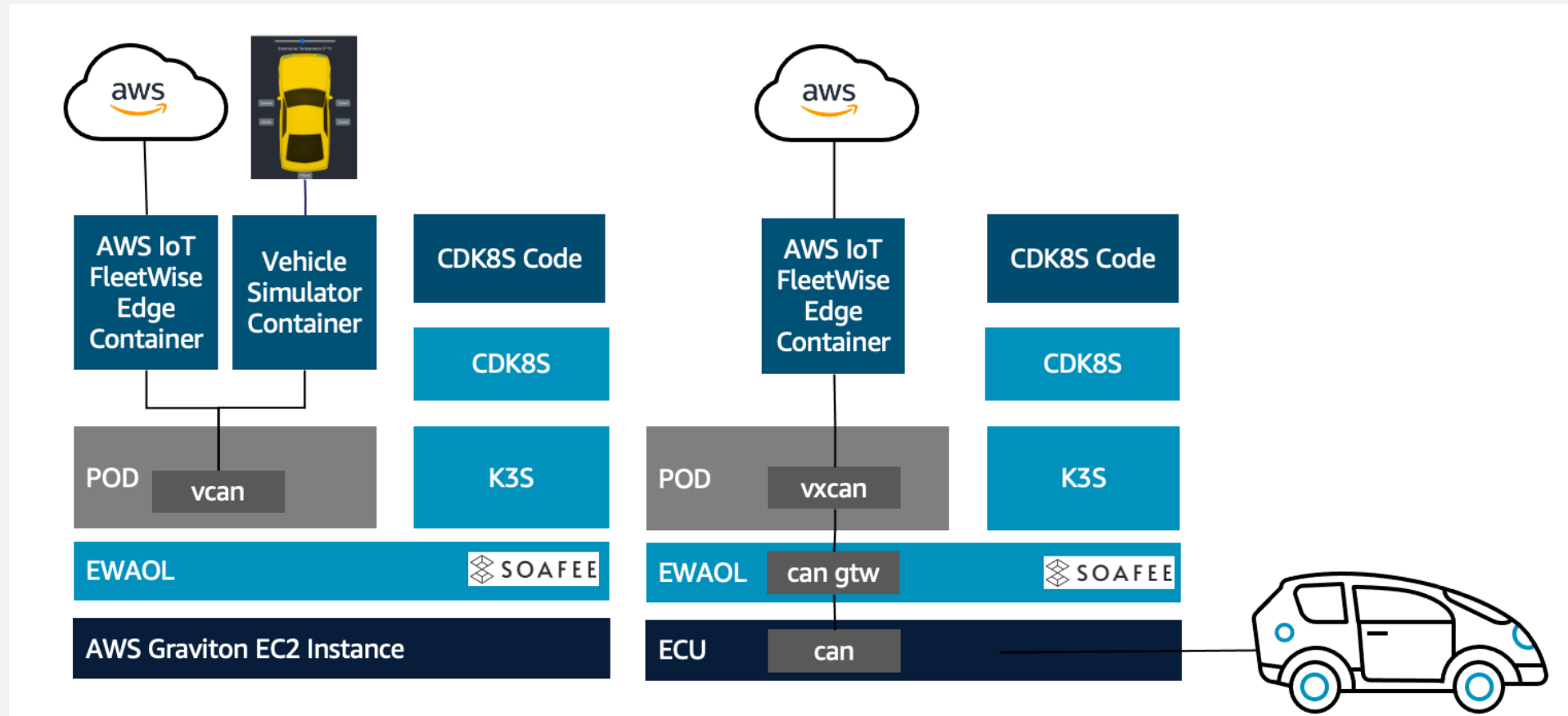
SOAFEE & Upper layers: **orchestration**



SOAFEE High-Level Reference Architecture



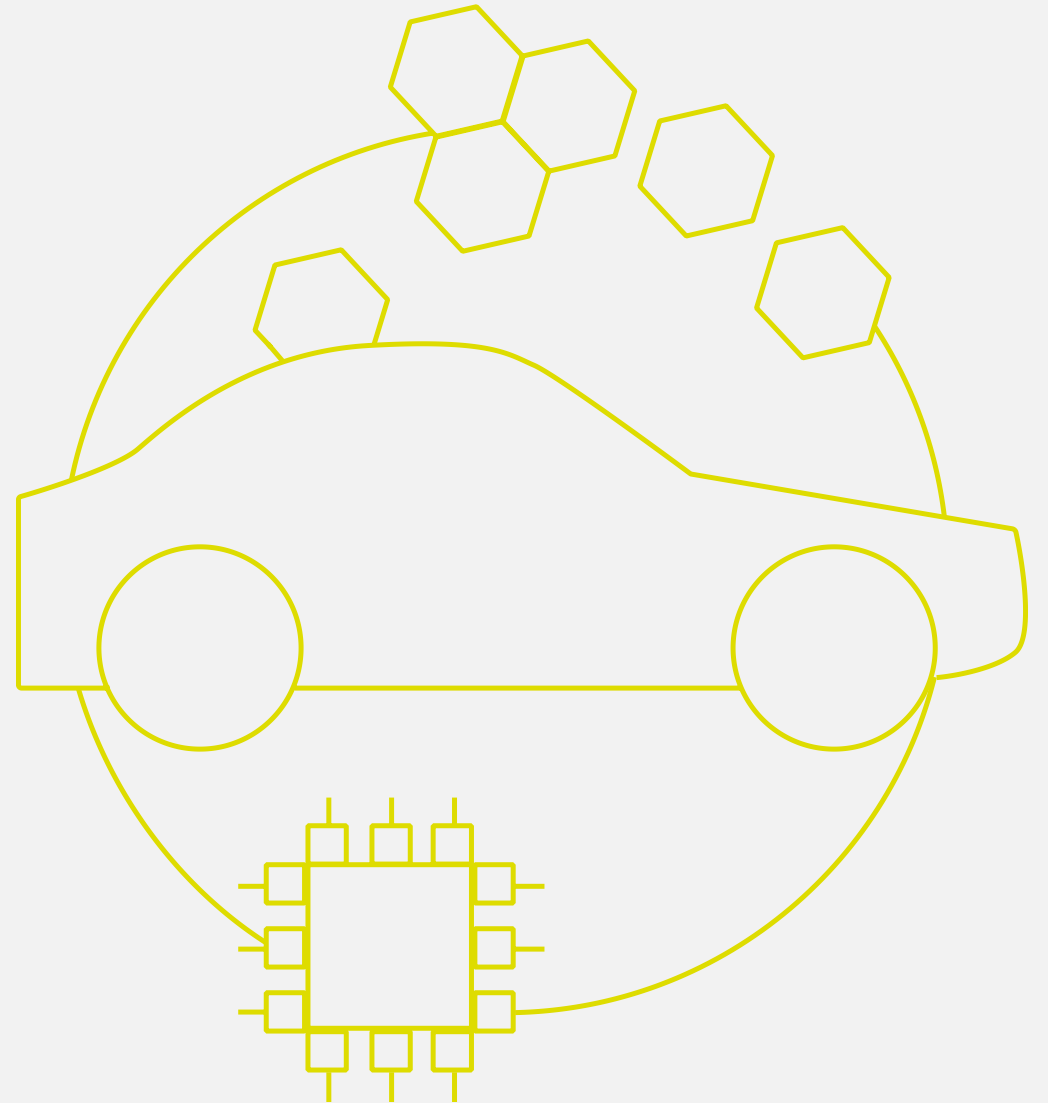
SOAFEE Container Orchestration with CDK8S

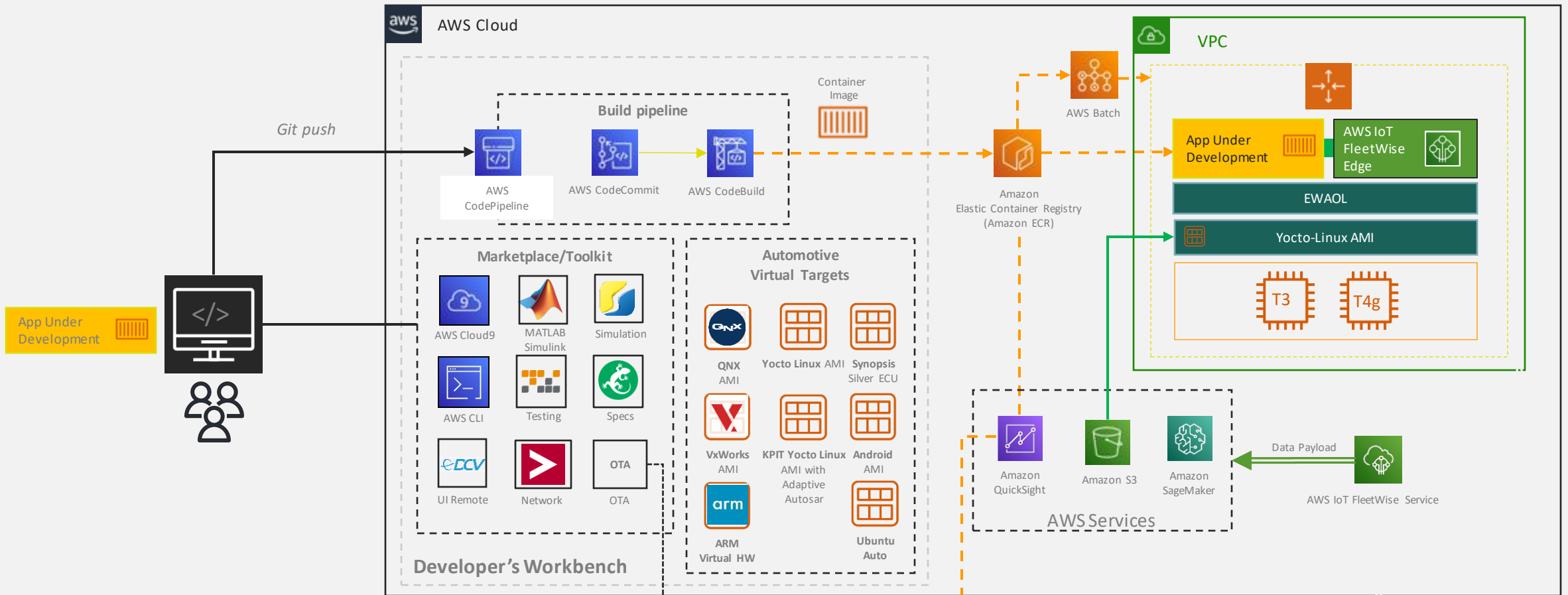


<https://github.com/aws-samples/demo-soafee-aws-iotfleetwise>

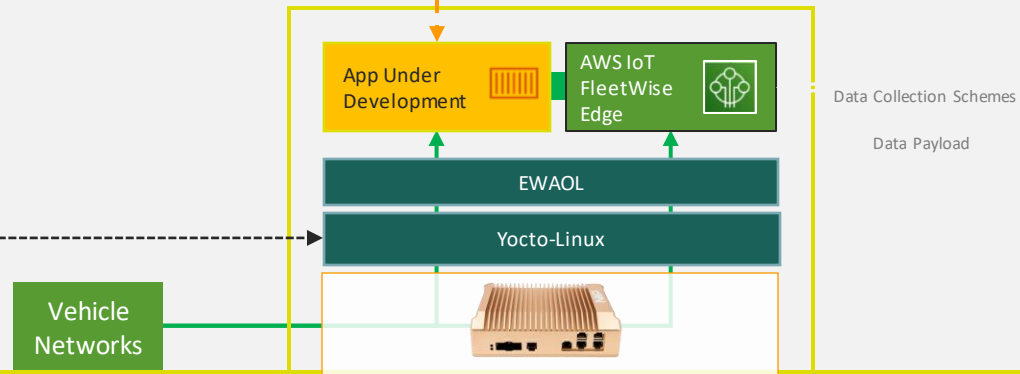


Towards an SDV Architecture: **Cloud-First Development**





SDV:
a **CONCEPT**
architecture



AWS Partners & Ecosystem Engagement in SDV

Virtual Engineering Workbench



Cloud-Native Tool Collaborations



Virtualized Targets



Final remarks

- **SDV is happening**, and requires a deep transformation of the technologies, tools and workflows in automotive.
- Part of that transformation is to find the **right level of abstraction** (HALs and Container orchestration), develop and diffuse new technologies if necessary.
- It's a **deep ecosystem play**, and each of us has opportunities to engage, for example with SOAFEE, Eclipse SDV, AGL, Yocto, etc.



Thank You

Danke

Gracias

Grazie

谢谢

ありがとう

Asante

Merci

감사합니다

धन्यवाद

Kiitos

شكرًا

ধন্যবাদ

תודה