亚马逊云科技

利用基于模型的方法和云原生开发实现软件定义汽车

叶江荣, 亚马逊云科技

2024 MathWorks
中国汽车年会

Key Challenge

HYPERSCALE

Thousands of cores of compute for development and validation. TBs of data to collect, ingest and store every day translates into PB scale data processing, storage and management problem.

AGILITY & SPEED

Optimized software engineering to reduce development and validation costs and enable faster Time to Market. Future proof R&D cycles. Integrated and Agile to rapidly innovate.

COST

PB scale data storage and large scale compute costs, managing fleet operations, significant capex of on-prem compute, lack of AV expertise requires significant human investment.

SAFETY

Safety of passengers and surroundings are top of mind of Amazon Web Service, customers and vehicle end users. Decisions are moving from human to vehicle.

ECOSYSTEM PLAY

Interoperability and seamless Integration of multiple first party and third party workload specific tools.

E/E COMPLEXITY

Modern vehicles have several tens of Electronic Control Units, making the system hard to test and update.

GLOBAL, SECURITY, DATA PRIVACY

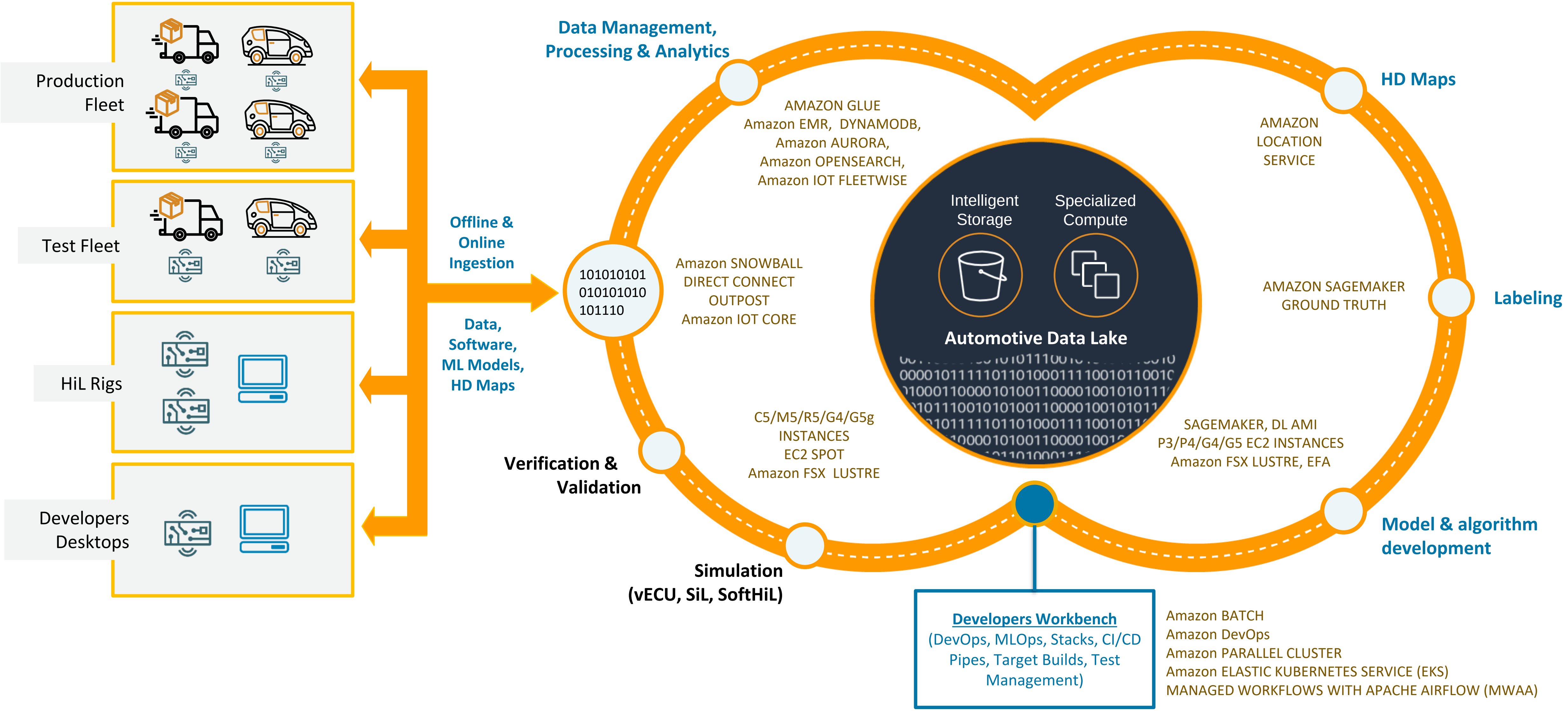
Global fleet requires managed service for complex operations, attain data and security compliance across the globe.

SDV is a *vehicle* whose **functions** can be **updated**, **secured**, and **personalized** throughout its lifetime.

-

The insights generated from SDV improve **current** and **future** generation of **vehicles**.

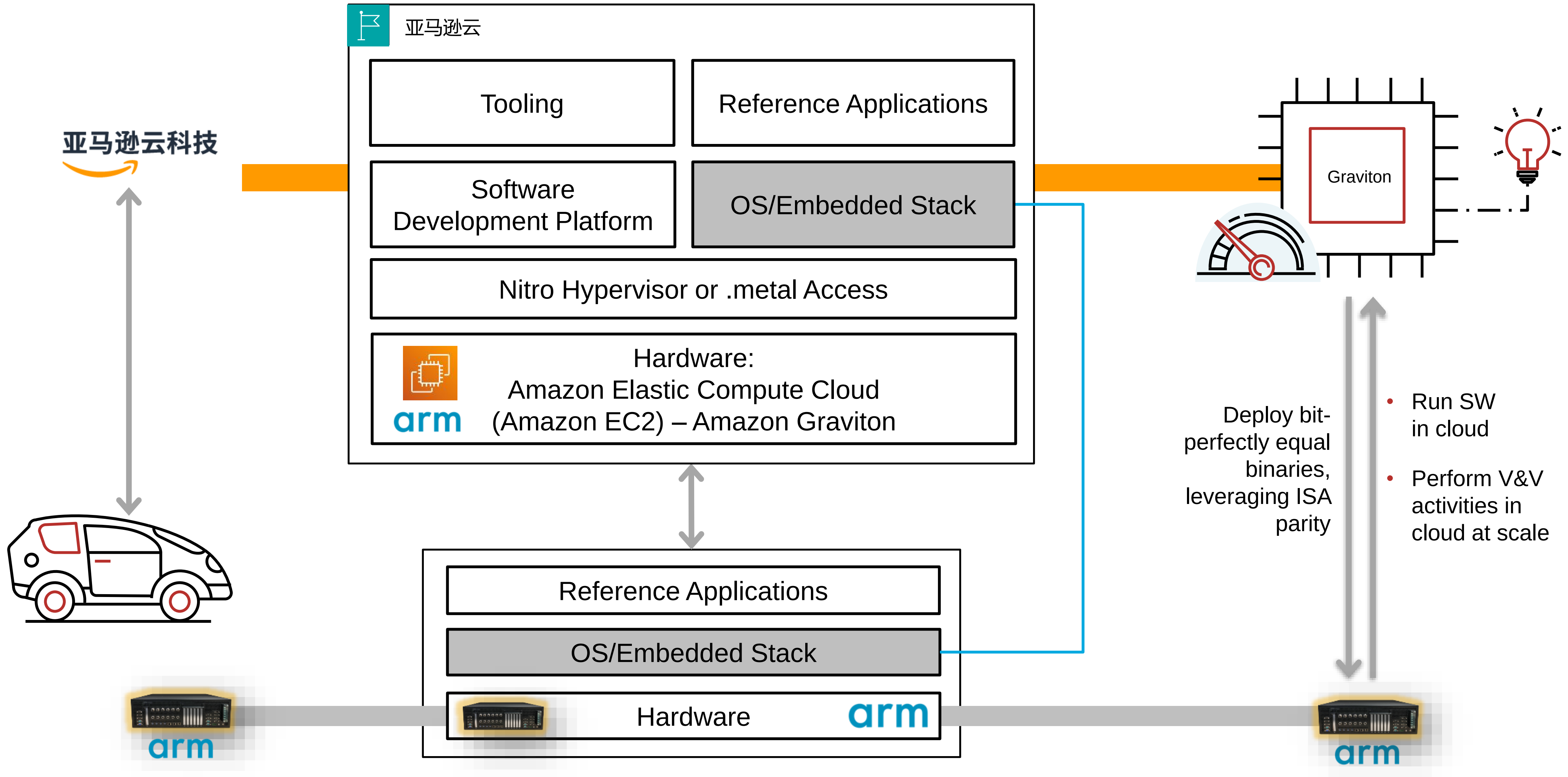
Automotive Development “Infinity” Workflow



Environmental Parity: Level and Personas

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

Environmental Parity with Amazon Web Service and ARM

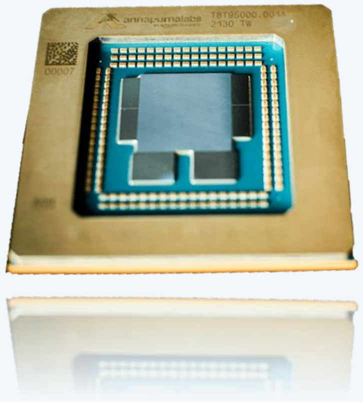


With ARM instances in cloud we are bringing parity between development and production silicon

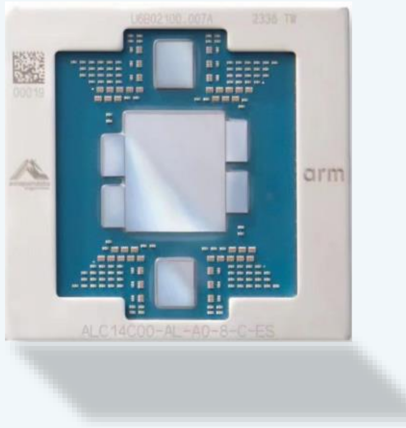
Graviton
2018

Graviton2
2019

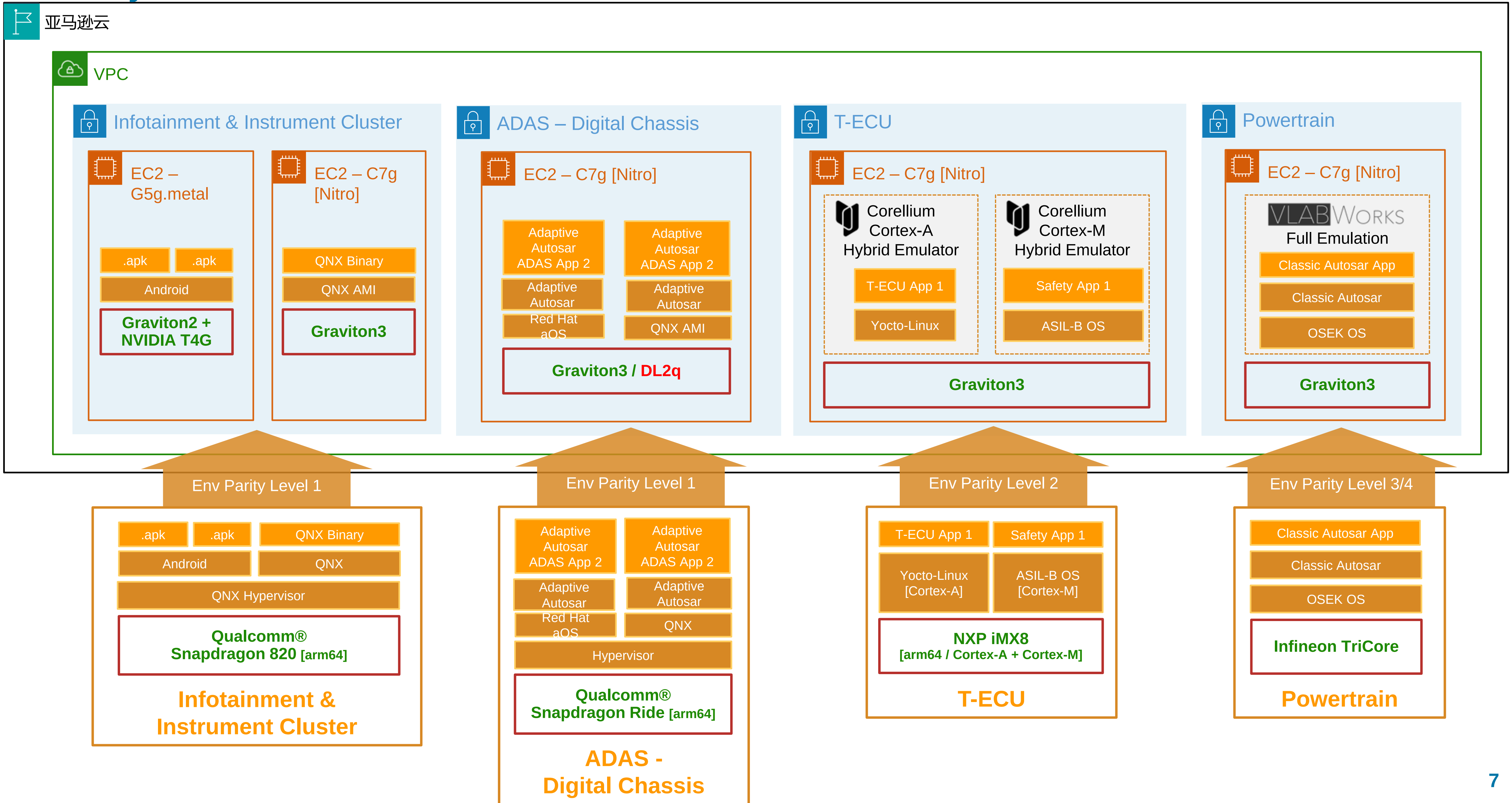
Graviton3
2021



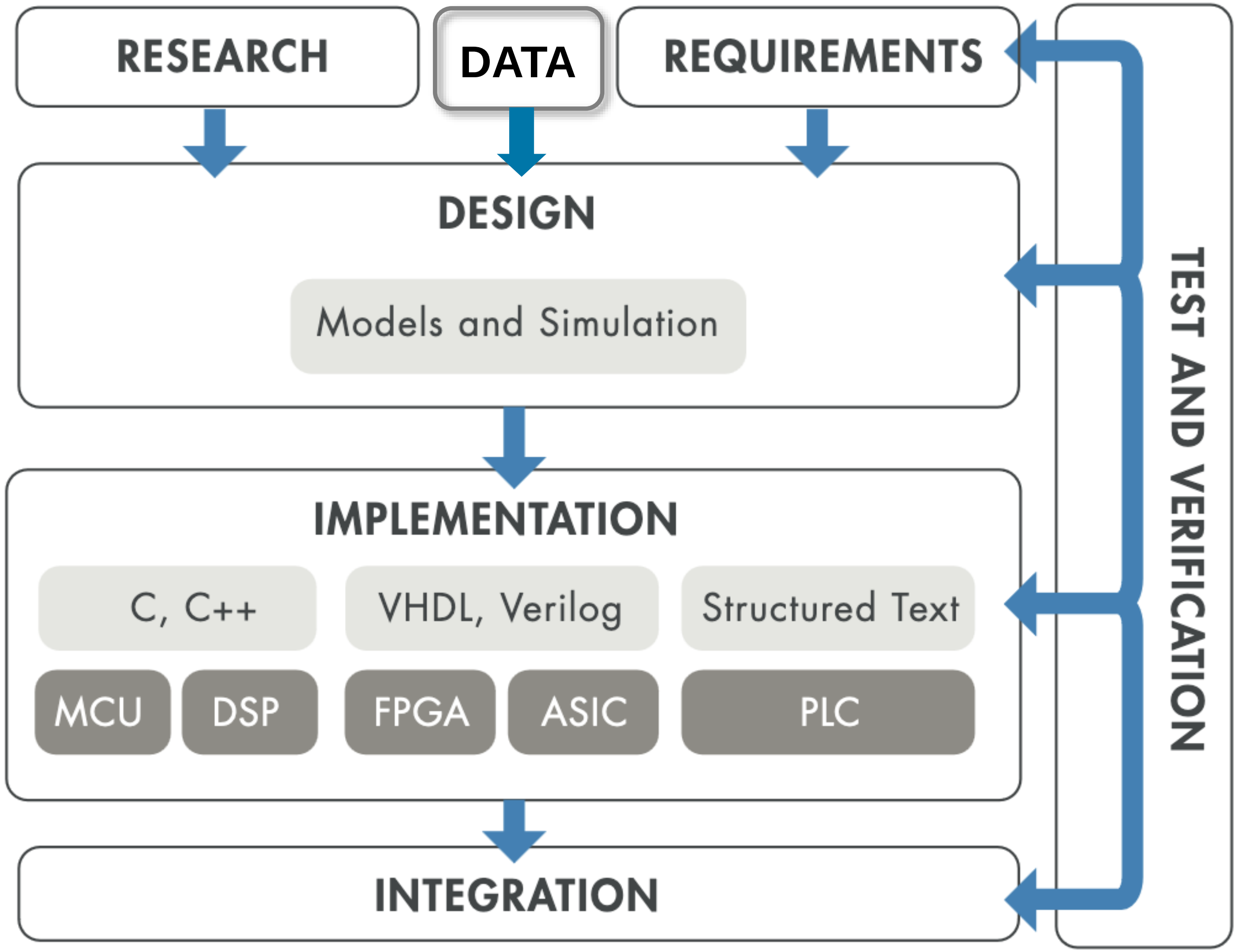
Graviton4
2023



Parity Levels Use Cases



Model-Based Design Workflow



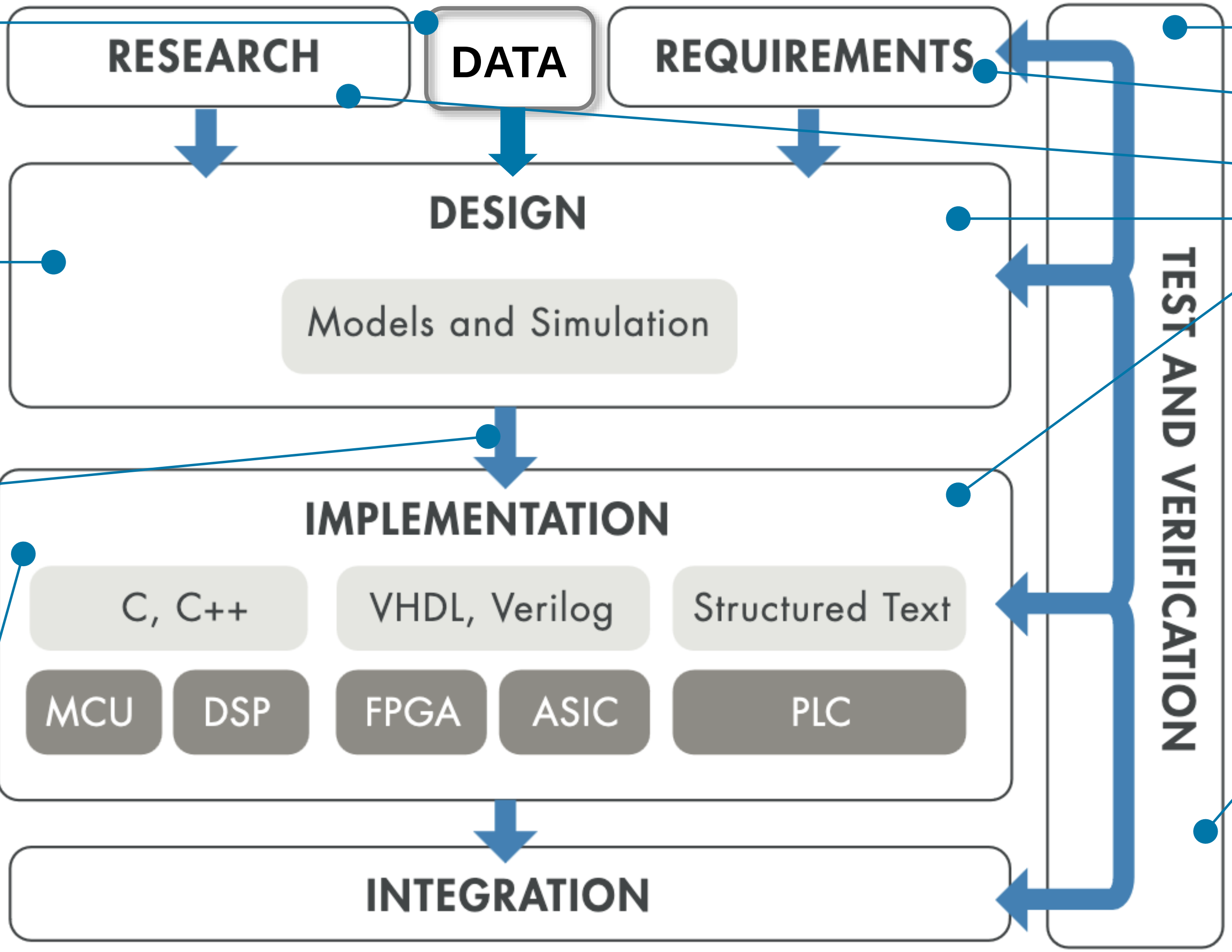
Model-based & Cloud-Native Design Workflow

Integrate **fleet data** with simulation

Broken down in a **microservices** perspective, where possible

Can generate the **same software** that will run across ECUs (virtual and physical)

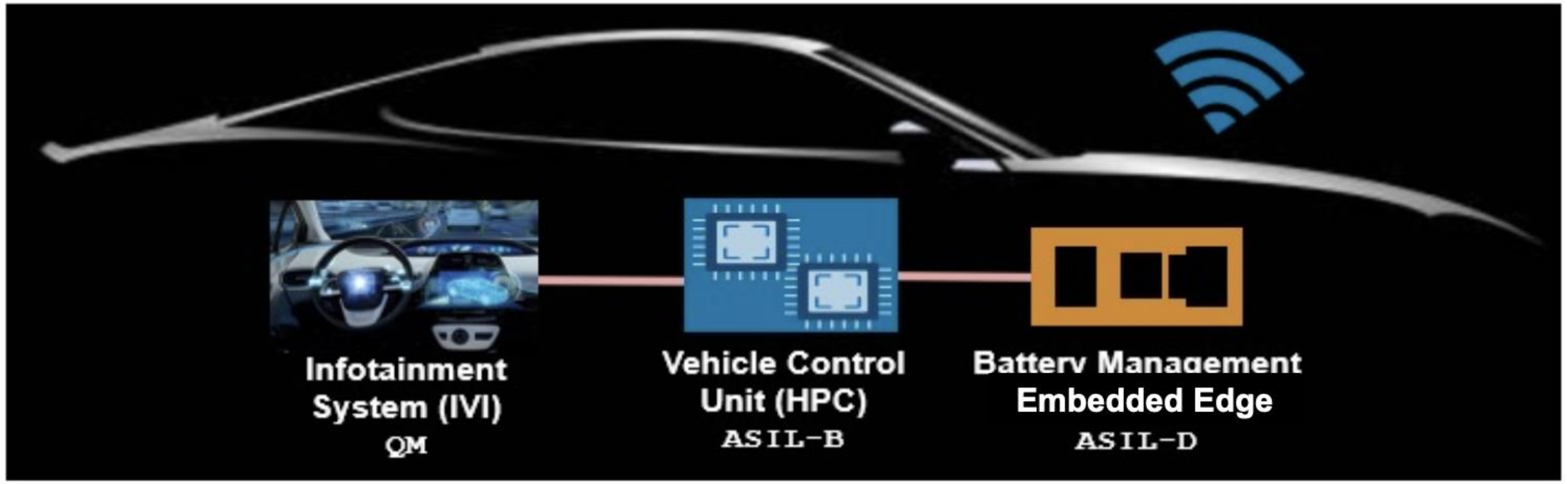
Targeting different ECUs or **vECUs** or **Automotive-Native AMIs** depending on the stage

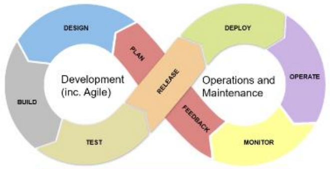


Separation of ”**Targets**” and “**Tools**”: the implemented software and the tools used to specify, generate, and test it need to be de- coupled and managed through standard interfaces - APIs

For most part in Cloud, leveraging cloud **scalability** and **elasticity** properties to ensure robustness of software function through **virtual vehicle simulation, real, and synthetic data**

SDV Development in Action – Demo Vision





Adopt

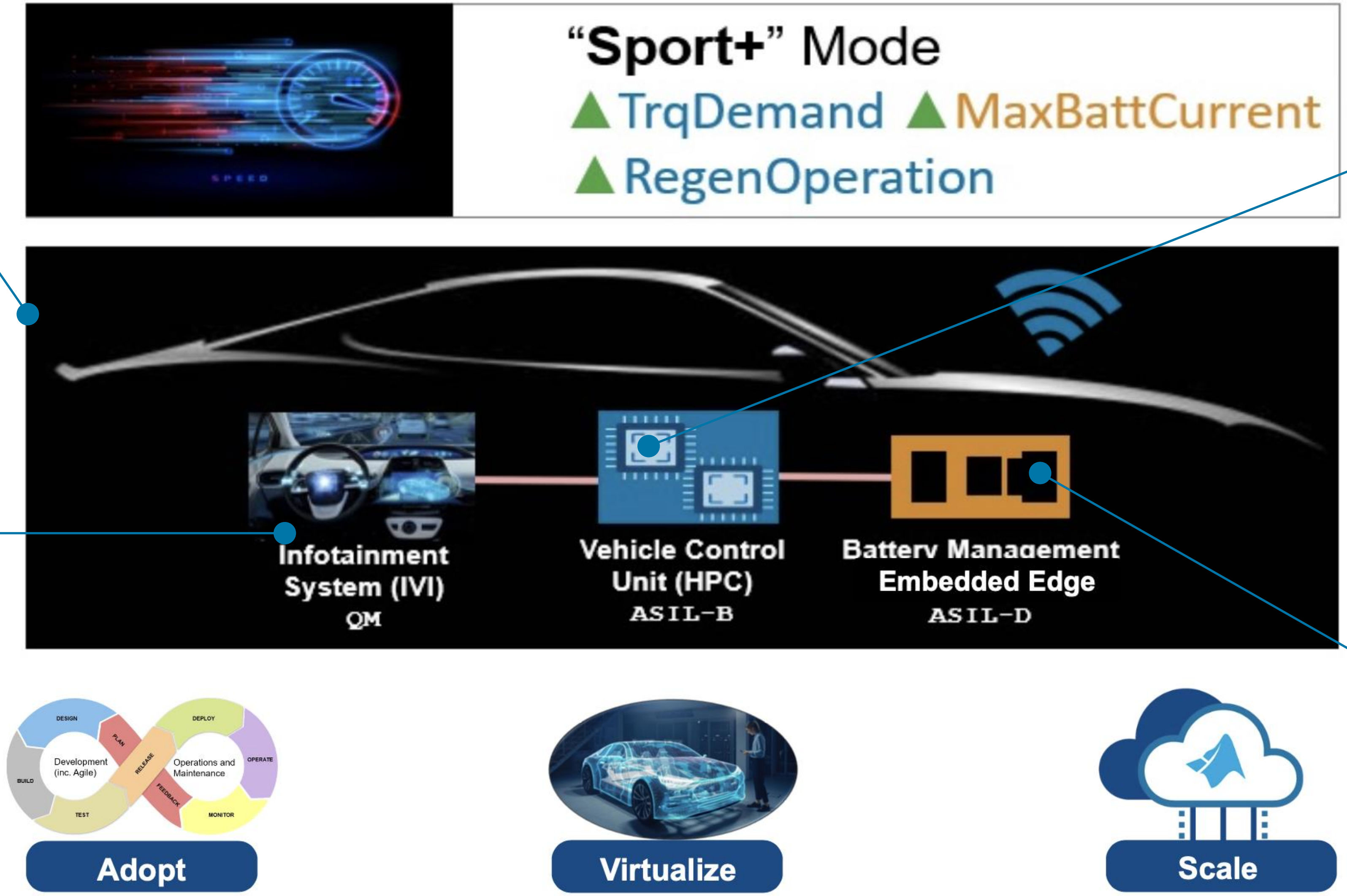
Virtualize

Scale

SDV Development in Action – Demo Vision

Model-Based Design and integration in **Simulink**

Automotive-Native **Amazon Machine-Image** (AMI)

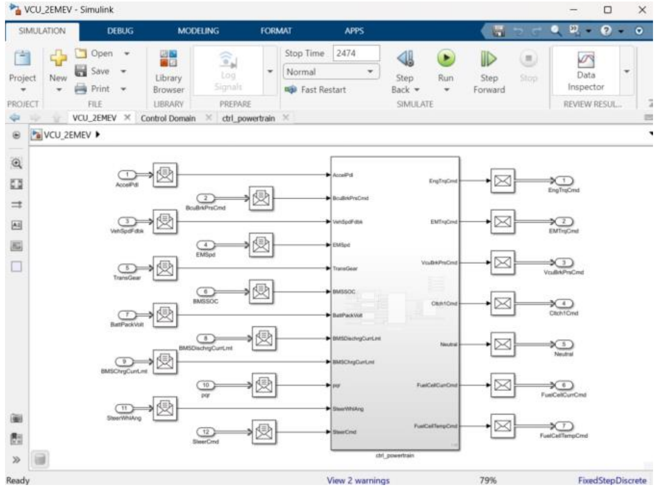


Elektrobit Adaptive Autosar Virtual ECU running on a POSIX containerized environment on Graviton EC2

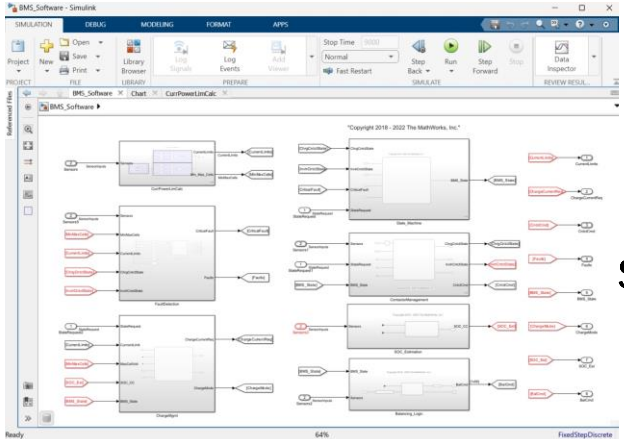
Elektrobit Classic Autosar ECU Level 3 emulation and simulation with **Synopsys Silver**

SDV Development in Action – Simulink Workflow

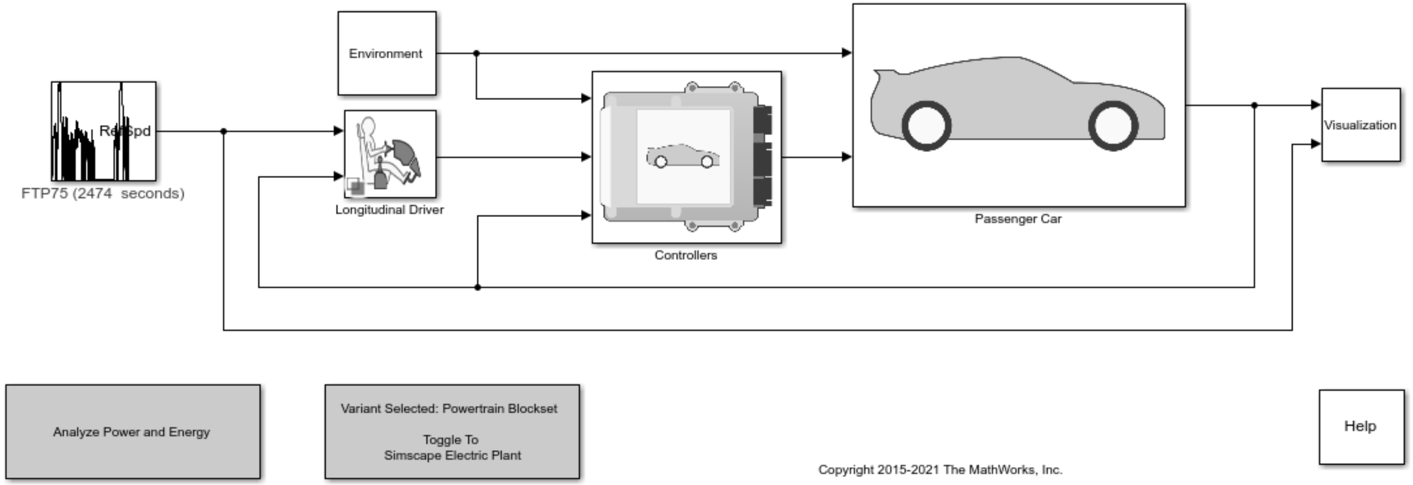
Vehicle Control Unit (VCU) Model

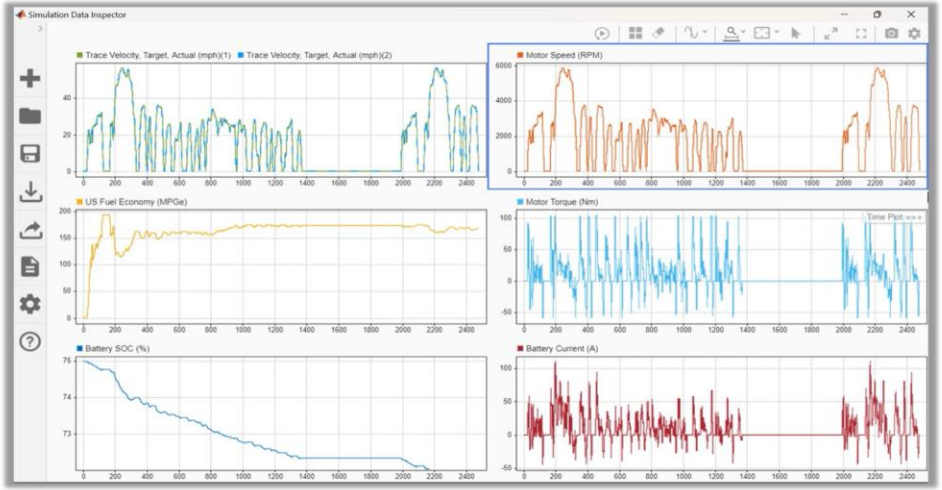


Update Control Strategies



Battery Management System (BMS) Model

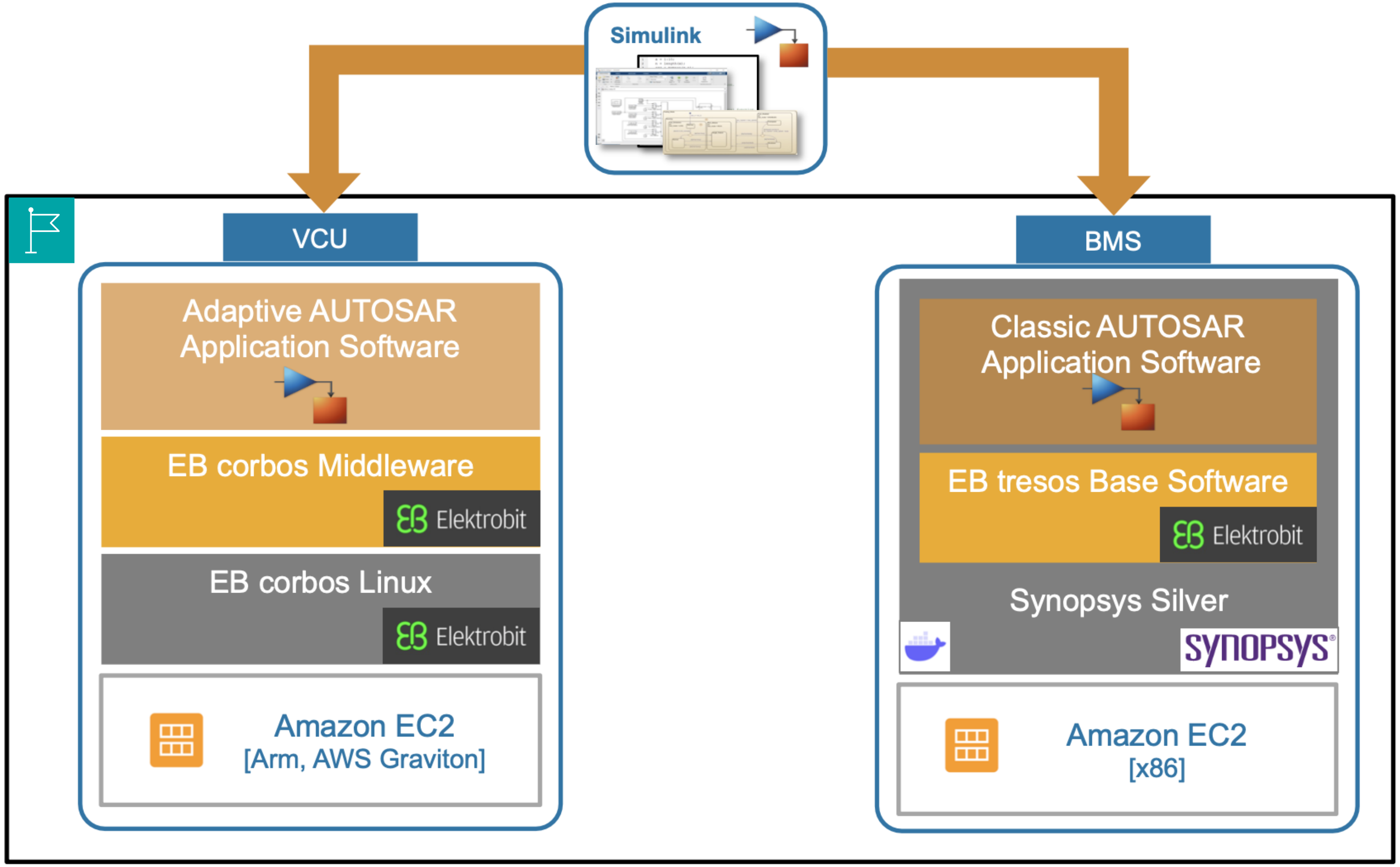




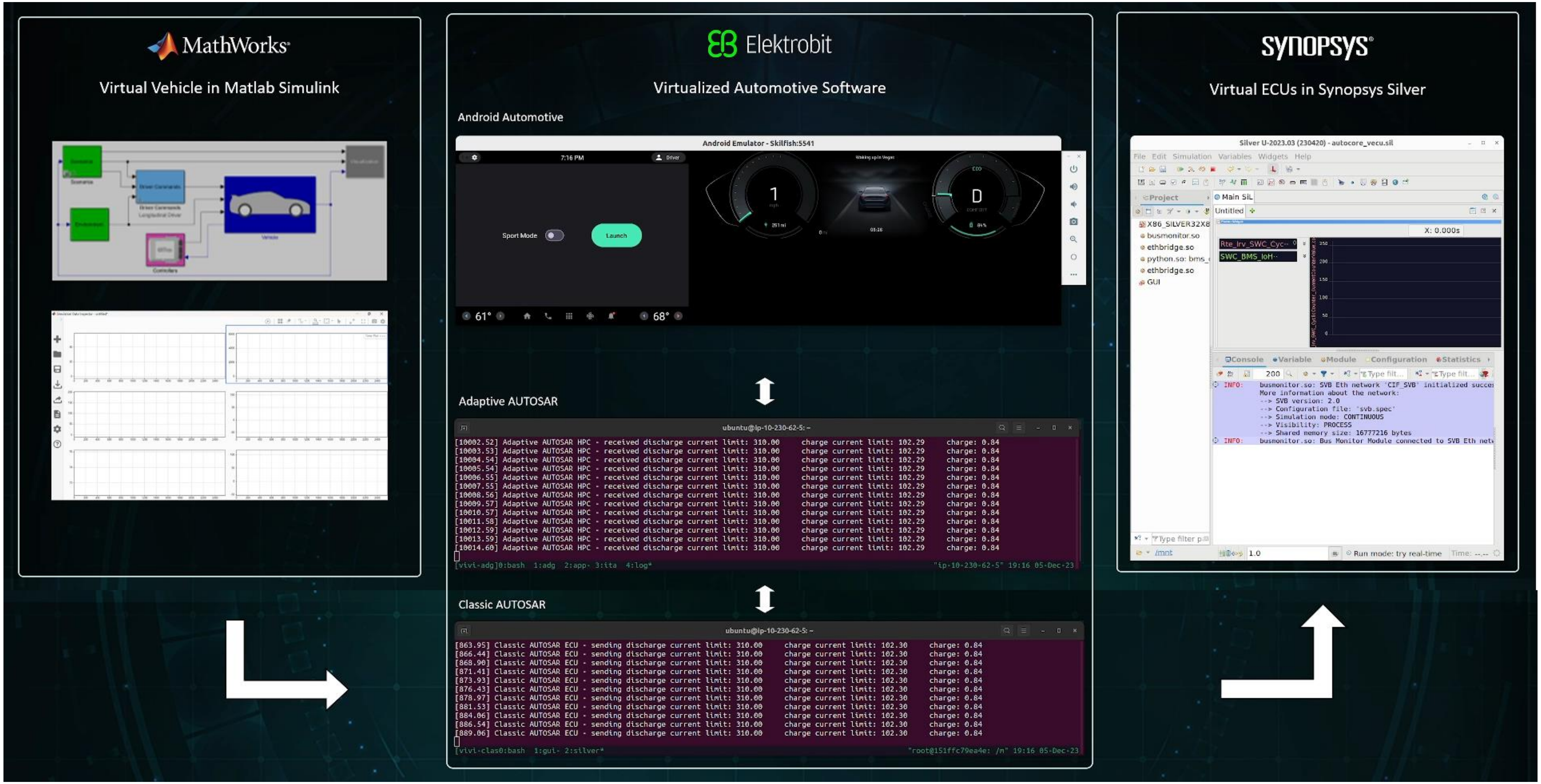
Integrate VCU and BMS into Virtual Vehicle Model

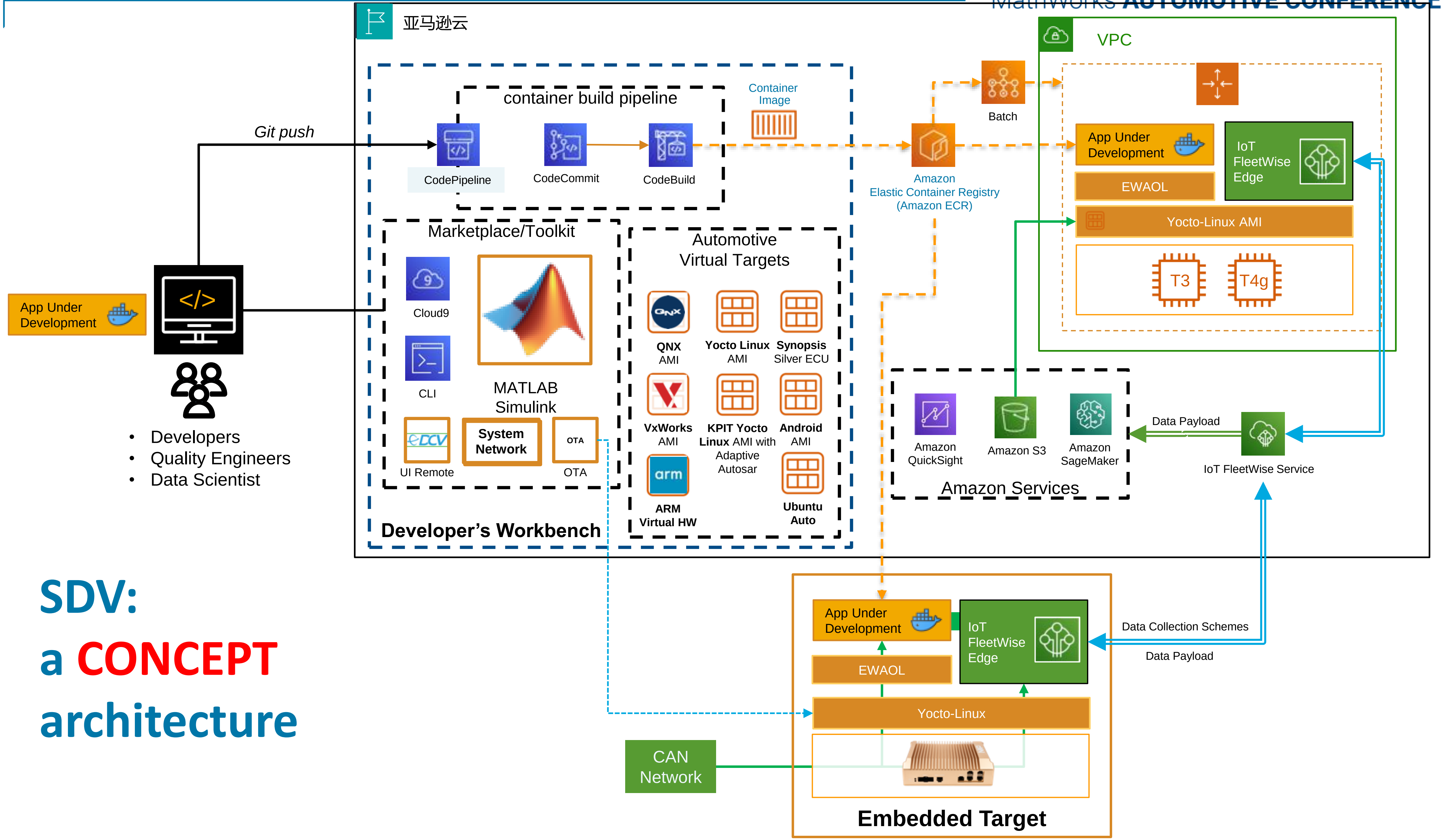
Analysis of Cloud based Simulation Data

SDV Development in Action – Cloud Deployment



SDV Development in Action – end to end demo





SDV:
a CONCEPT
architecture

Engineering Workbench

References

Continental
TOYOTA
STELLANTIS

Partners

KPIT
GlobalLogic
A Hitachi Group Company
Continental

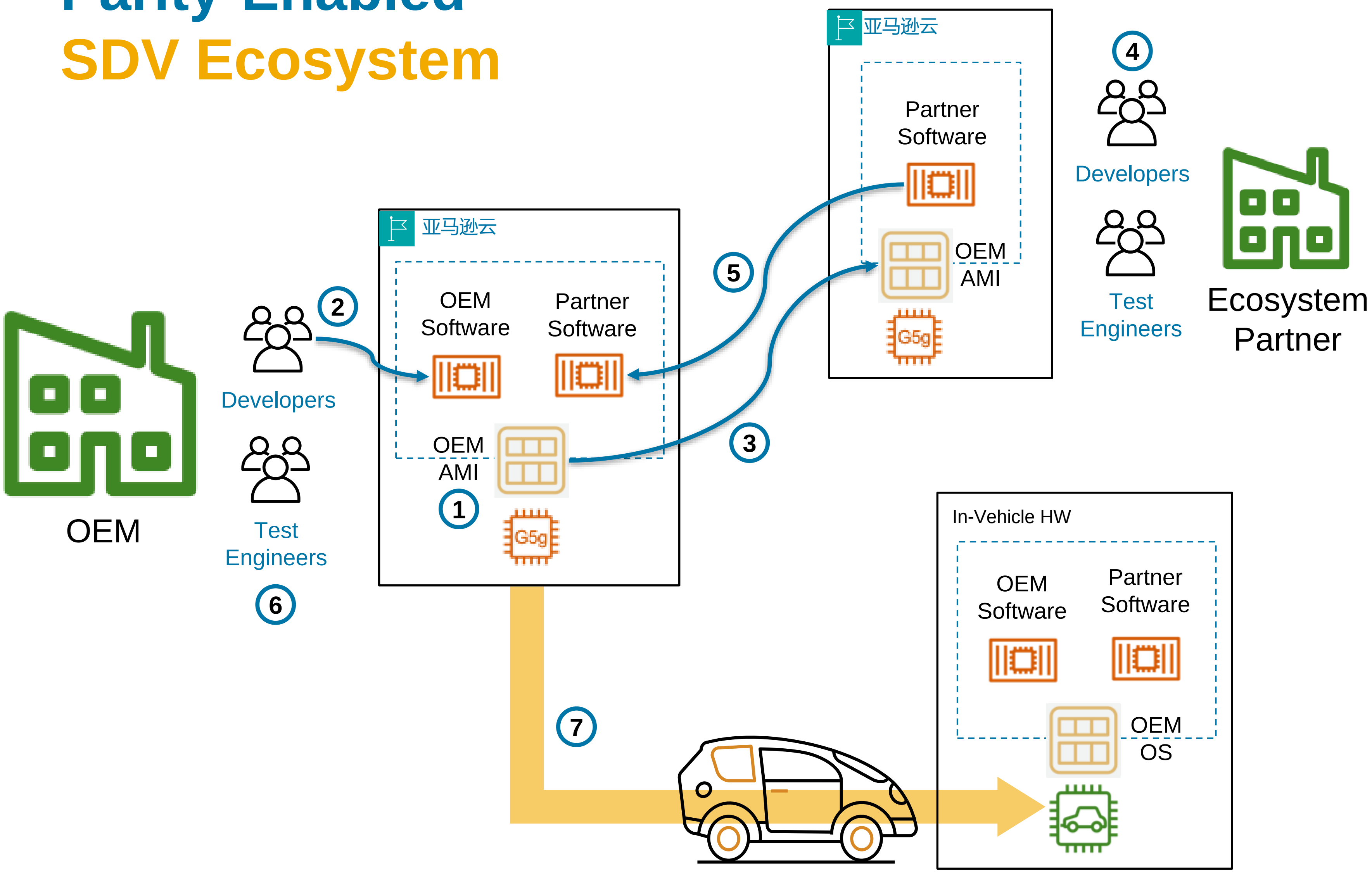
Cloud-Native Tool Collaborations

MathWorks
Qt Group
SYNOPSYS
dSPACE
CORELLIUM
VECTOR
RIGHTWARE
Genymotion
ETAS

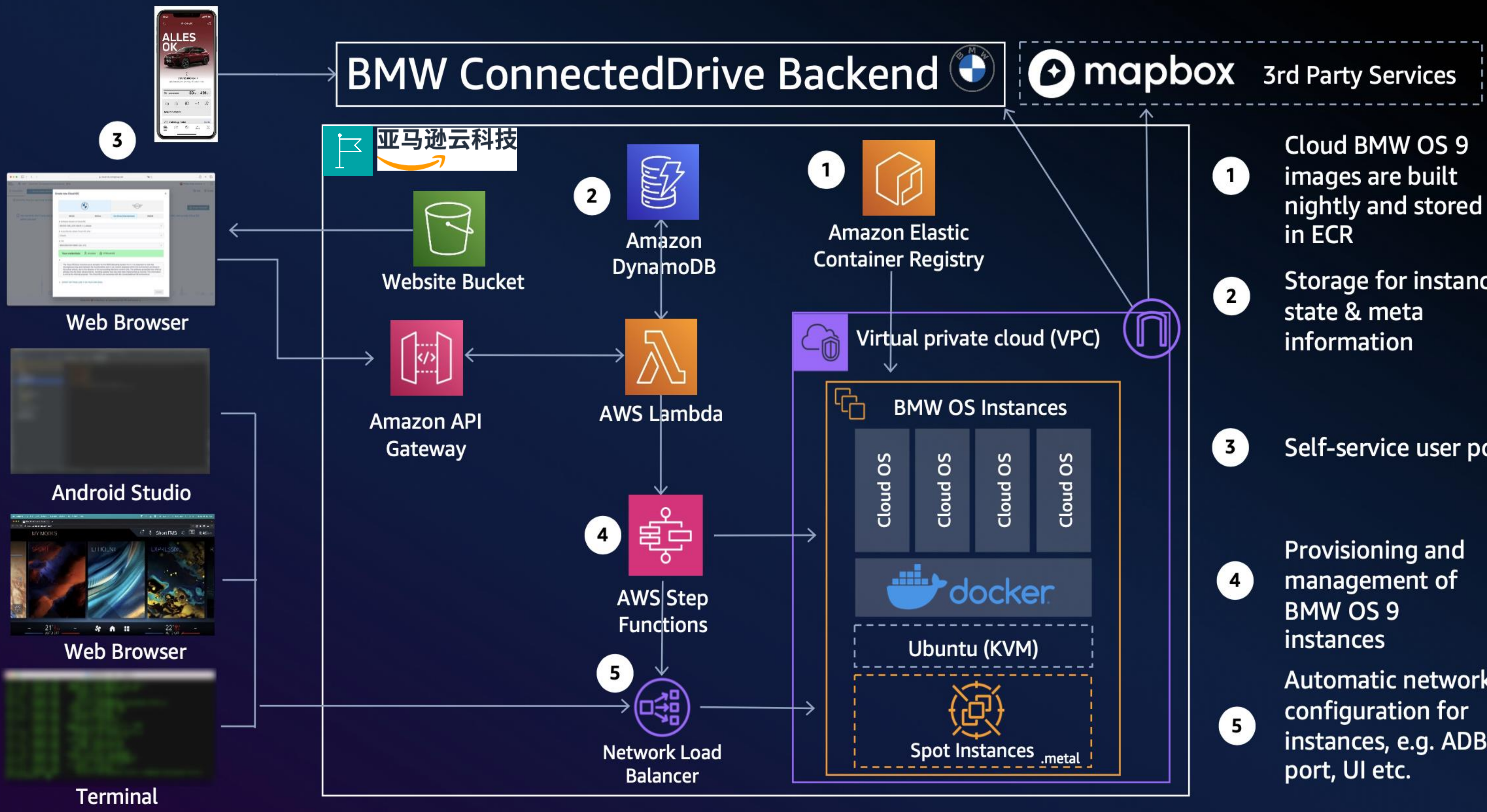
Virtualized Targets

yocto PROJECT
AUTOMOTIVE GRADE LINUX
Continental
Red Hat
Panasonic
QUALCOMM
Ubuntu
VxWorks
MARELLI
BlackBerry QNX
arm
Amazon Graviton
C7g

Parity-Enabled SDV Ecosystem



1. OEM creates and maintains “native” **OEM AMIs**
2. **OEM developers** can start to develop native applications on Amazon Web Service
3. OEM distributes the OEM AMI to its ecosystem of **partners**
4. Using the native OEM AMI, partners can *develop* and *test* **containerized applications and services**
5. The partner **submits** the container back to OEM
6. OEM **validates** the containerized app/service
7. The containers are **deployed** in production vehicles



1. Cloud BMW OS 9 images are built nightly and stored in ECR
2. Storage for instance state & meta information
3. Self-service user portal
4. Provisioning and management of BMW OS 9 instances
5. Automatic network configuration for instances, e.g. ADB port, UI etc.

Summary

- SDV is happening, and requires a deep transformation of the tools and workflows in automotive
- Model-Based is a first layer of abstraction perfect to design SDV systems!

  Thanks to the collaboration with MathWorks, we can provide our customer with even more powerful and “SDV”-aligned Model-Based tools, properly architected, and leveraging cloud-native practices.
- Virtual targets and workbench would accelerate the SDV journey.

2024 MathWorks
中国汽车年会

Thank you

MathWorks