

Software-Defined Vehicle Solutions

Pierre Cotte

Christian Michel Sendis

Simon Oudin

May 2024

| **Public** | NXP and the NXP logo are trademarks of NXP B.V. All other product or service names are the property of their respective owners. © 2024 NXP B.V.



Agenda

- What is a Software-Defined Vehicle?
- How to Build a Software-Defined Vehicle
  1. Select optimal E/E architecture
  2. Re-architect software
  3. Adopt new Ethernet technology
  4. Enable vehicle-to-cloud software
  5. Use virtual modelling
  6. Select the right partner(s)
- Conclusion



Agenda

- What is a Software-Defined Vehicle?
- How to Build a Software-Defined Vehicle
  1. Select optimal E/E architecture
  2. Re-architect software
  3. Adopt new Ethernet technology
  4. Enable vehicle-to-cloud software
  5. Use virtual modelling
  6. Select the right partner(s)
- Conclusion

What is a software-defined vehicle?

Software …

defines the customer experience

becomes the differentiator

is the focus point for innovation

brings OEMs new revenue streams

Hardware …

is a stable, scalable future-proof platform

New features include …

car & cloud integration

over-the-air updates

flexibility to continuously evolve

service-oriented architectures

virtualization and containerization

New processes are …

DevOps: continuously integrate, test and deployment of software

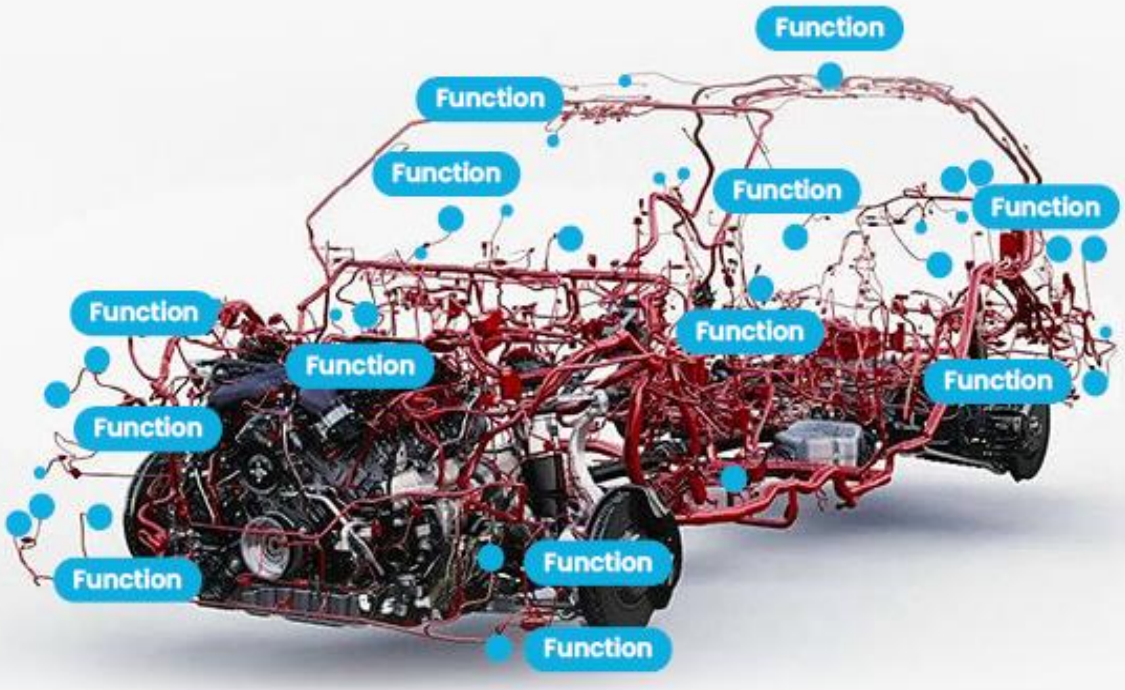
Agile and lean methods

digital twin virtual modelling



Vehicle transformation underway

From **hardware-oriented** vehicle

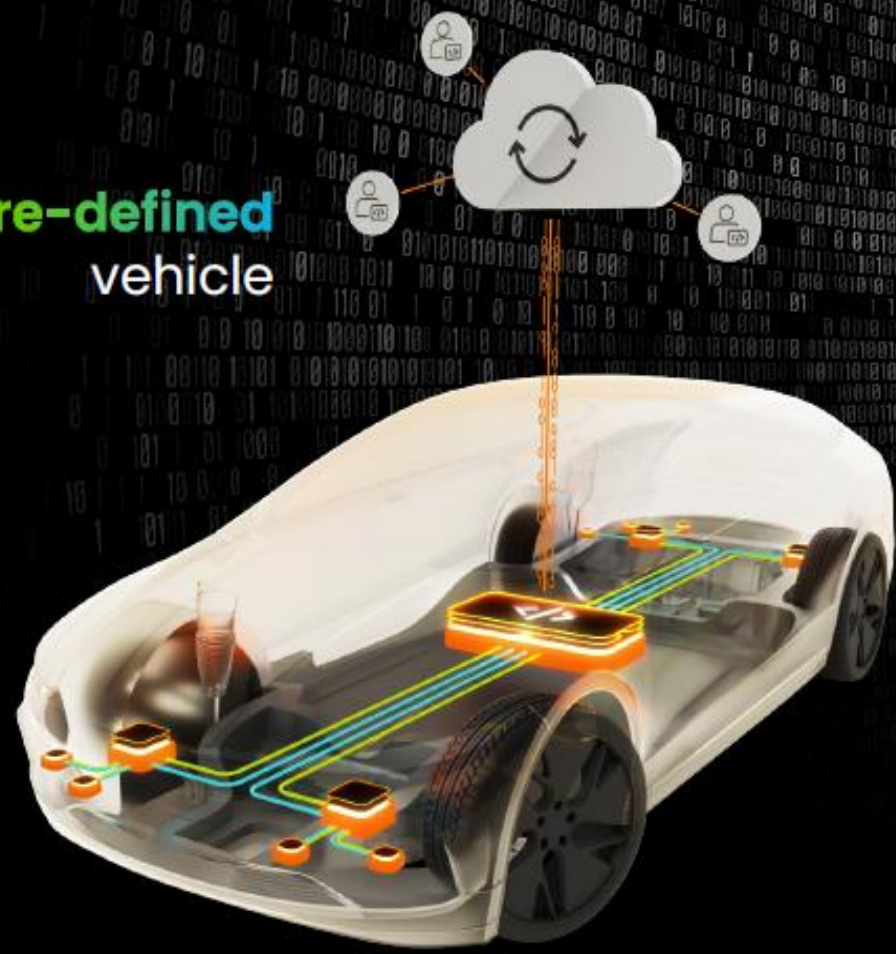


Functions **tied to ECUs**

Static architecture

Exponential increase in **complexity**

To **software-defined** vehicle



Functions can be **anywhere in the vehicle**

Flexible architecture

Simplification of implementation

Software creates new business paradigms for automotive

Engineering efficiencies

E/E and system BoM

Software R&D

Overall system performance

Speed time to market

Freedom to innovate



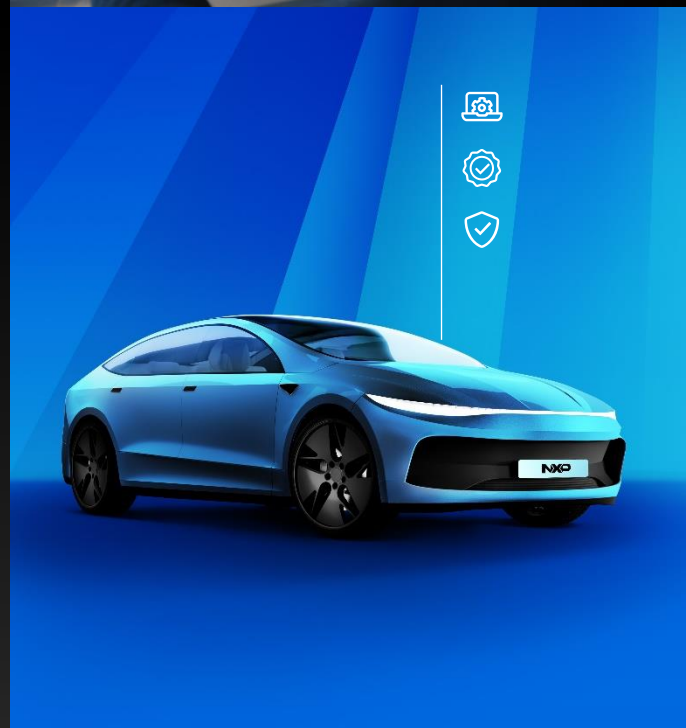
New value creation

Data-driven revenue streams

New features after vehicle is sold

Predictive maintenance

Customization and personalization



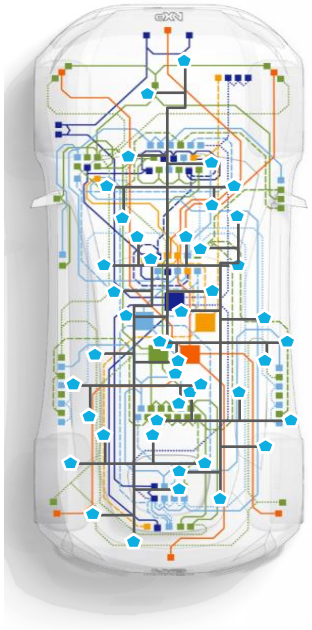


Agenda

- What is a Software-Defined Vehicle?
- How to Build a Software-Defined Vehicle
  1. Select optimal E/E architecture
  2. Re-architect software
  3. Adopt new Ethernet technology
  4. Enable vehicle-to-cloud software
  5. Use virtual modelling
  6. Select the right partner(s)
- Conclusion

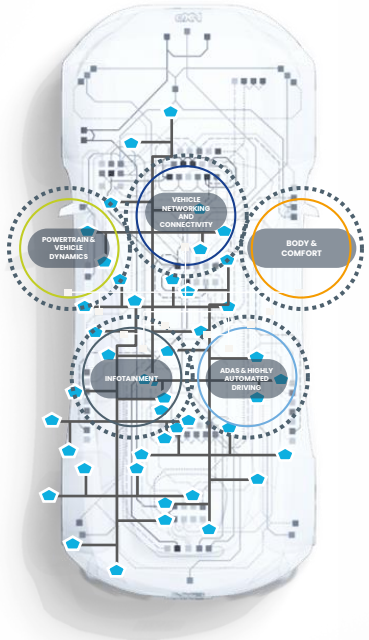
E/E Architecture Evolution: MULTIPLE steps to Software-Defined Vehicles

Flat



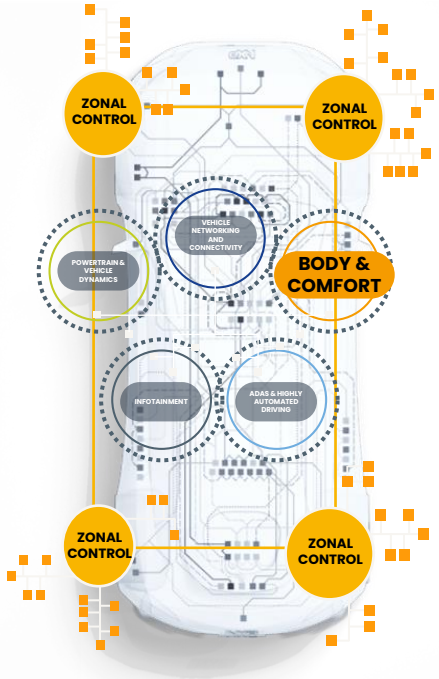
LEGACY ARCHITECTURE ORGANICALLY GROWN

Domain



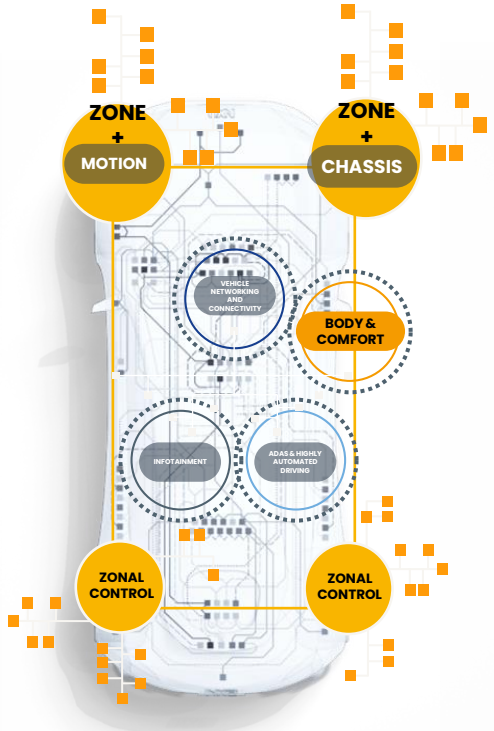
SCALABLE AND CENTRALIZED SOFTWARE DEVELOPMENT

Domain + Body Zonal



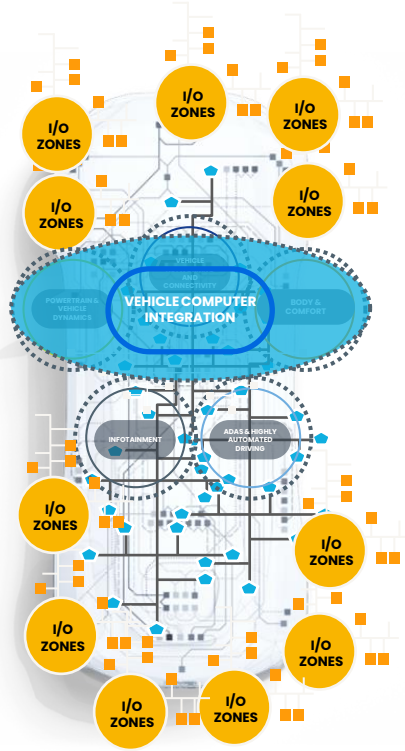
DOMAIN SW BENEFIT + SIMPLIFIED WIRING AND VEHICLE NETWORK

Cross-Domain Zonal



LOWER HW COST VS MORE COMPLEX SW

Consolidated Compute



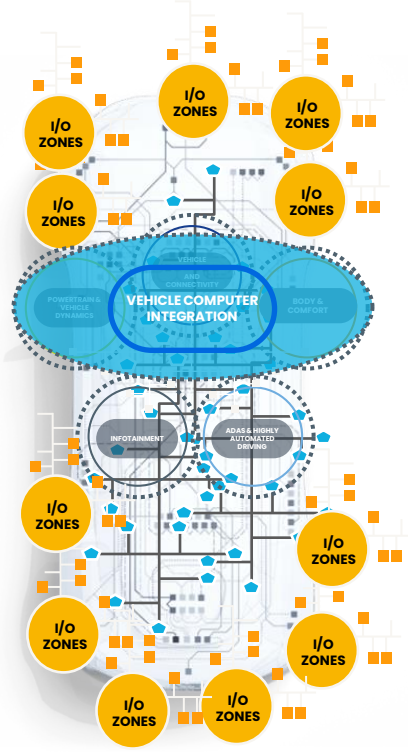
SIGNIFICANTLY REDUCES SW DEVELOPMENT COST, VS HIGHER HW COST FOR FUTURE-PROOFING

Logical and physical transitions

Flat



Consolidated Compute



TWO PARALLEL ARCHITECTURAL CHANGES

1 **<u>CENTRAL DOMAIN FOCUS</u>: SCALABLE AND CENTRALIZED SOFTWARE DEVELOPMENT**

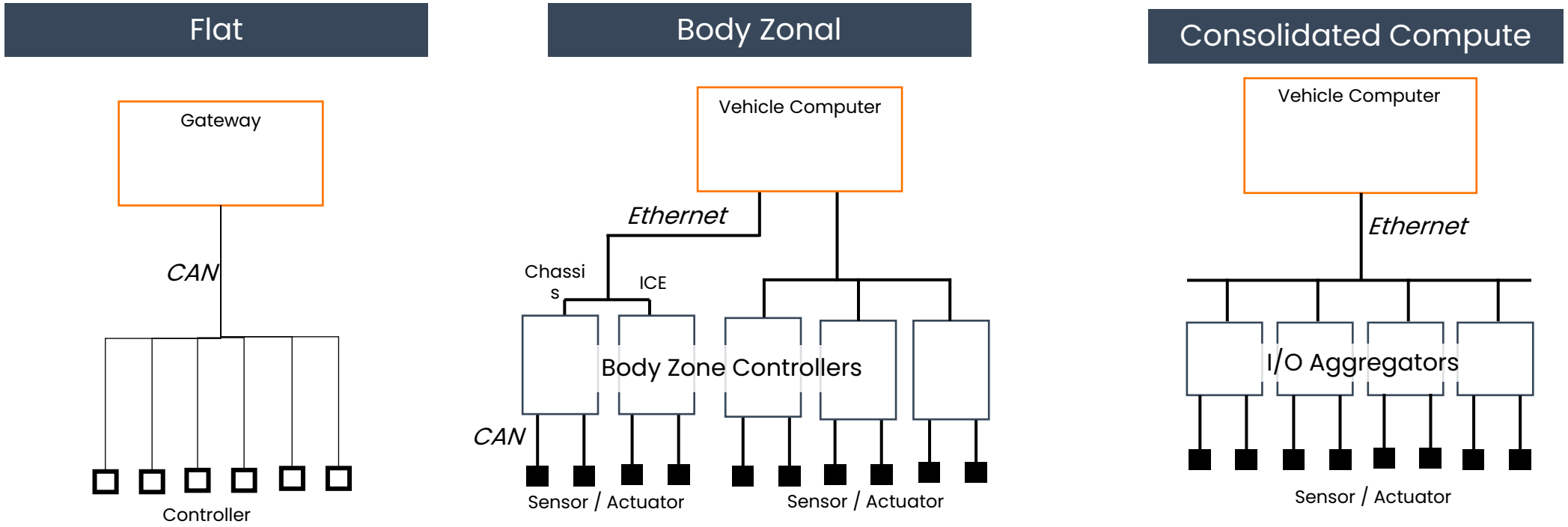
1. Flexible & scalable software environment
2. Efficiently supports the user defined vehicle
3. Centralized OTA, easily upgradable Software

+

2 **<u>ZONAL FOCUS</u>: SIMPLIFIED WIRING AND CONNECTIVITY**

1. Dramatically reduced wire routing and cable costs
2. Flexible data monetization
3. Easily upgradable Hardware

Implications of E/E architecture changes



- Communications – transition to high bandwidth Ethernet
- Function distribution – dedicated controller at the edge, to hierarchical setup
- ECU consolidation – one vehicle function = one ECU is history

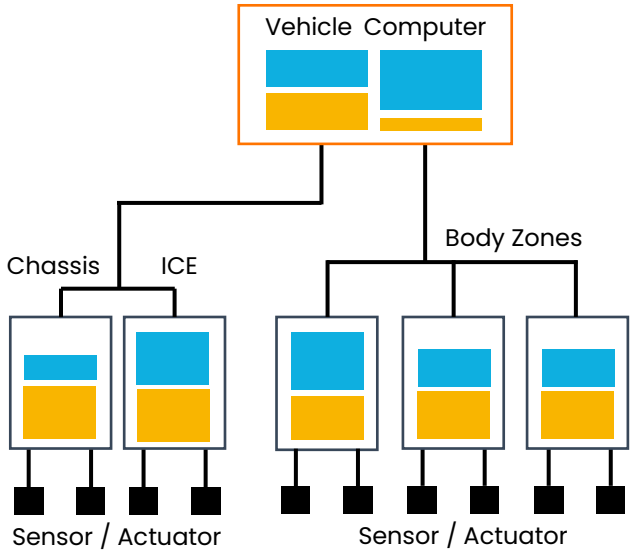
Zone control vs Vehicle computer

Real Time Compute | I/O control

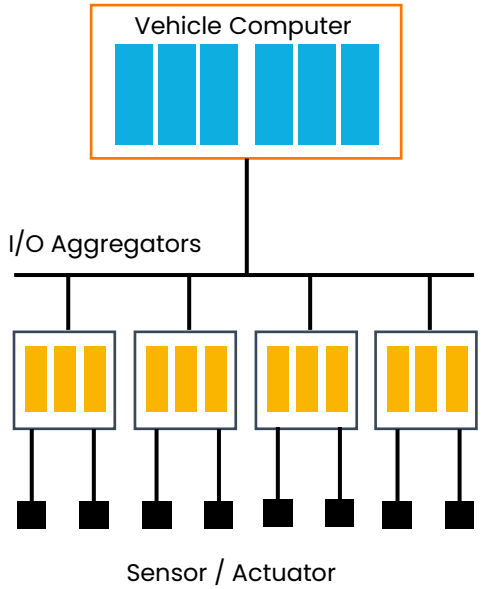
Domain Controller / Vehicle Computer
- Higher level of control
- New features in software

Body Zonalization
- Wiring harness reduction, modularity introduced
- Primary / secondary body controllers

Body Zonal



Consolidated Compute



Vehicle Computer
- Hosts "All decision making"
- Fully flexible platform for SDV

I/O Aggregators
- Generic sensor / actuator control
- Available to multi-applications

Development path	Closer to legacy	Newer approach, more development
Cost	Each zone controller different Low re-use	Each zone control the same (I/O Aggregator), economies of scale
Realtime control	Easier to manage propulsion from zone controller (or domain controller)	Propulsion management from vehicle computer possible
Ability to perform feature updates via software OTA	Possible, but complex – code can be placed in vehicle computer & zone controllers	High – compute code all in vehicle computer
Advantage	Quickest path to cost reduction (from wiring)	Single location for new software features Easier for SW deployment/maintenance & network mgmt

Where will future updates be implemented?

NXP Vision

- What functionality is inside the zone controller? (e.g. I/O aggregation, body functions, motion functions)
- What software do you want to update on regular cadence?
- What headroom (processing/memory/bandwidth) do you add for future new features?
- Where do you want to add new functions? (e.g. ADAS/IVI only, or also Vehicle Computer)

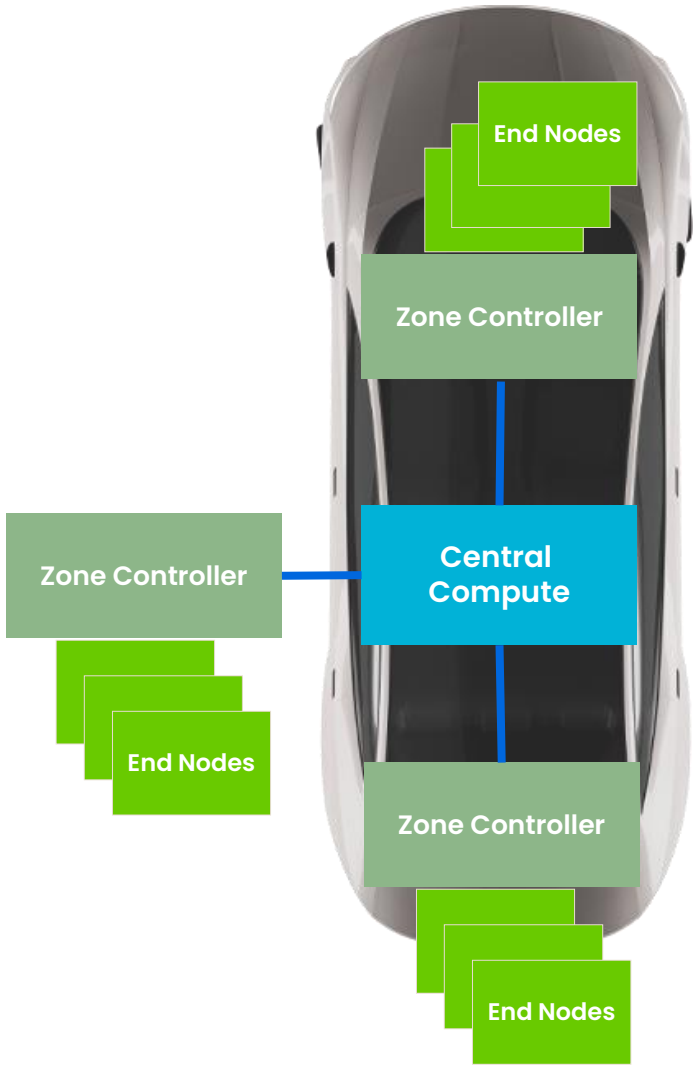




Agenda

- What is a Software-Defined Vehicle?
- How to Build a Software-Defined Vehicle
  1. Select optimal E/E architecture
  2. Re-architect software
  3. Adopt new Ethernet technology
  4. Enable vehicle-to-cloud software
  5. Use virtual modelling
  6. Select the right partner(s)
- Conclusion

SDV COMPUTE : A HIERARCHY OF AUTOMOTIVE PROCESSING



- **Central Compute**
  - High-performance, multi-core processors (real-time + applications)
  - Consolidation of vehicle functions, vehicle management and services
  - High-speed Ethernet backbone connections to Zone Controllers

- **Zone Controller**
  - High-performance microcontrollers and real-time processors
  - Consolidation of vehicle functions located in that area of the vehicle
  - High-speed Ethernet backbone connection with Central Compute
  - Local input/output (I/O) control with sensors/actuators (CAN, LIN…)

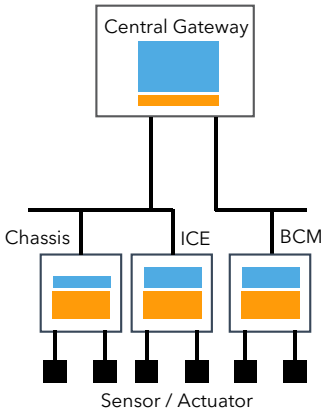
- **End Nodes**
  - Low-power, microcontrollers for fixed vehicle functions
  - Interface directly with sensors and actuators (CAN, LIN…)
  - Examples: wiper, lighting, switches

NOTE: Vehicle functions can migrate over time for further consolidation:
End Nodes → Zone Controllers and Zone Controllers → Central Compute

SW Partitioning – Architecture Choices

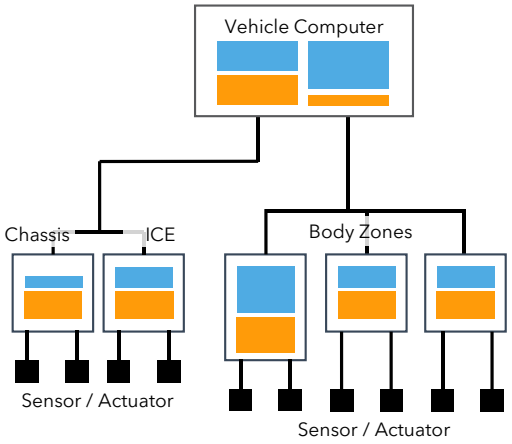
Real Time Compute | I/O Control

Flat



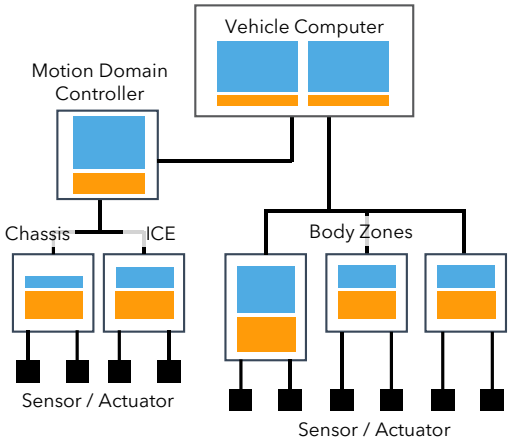
Legacy ECUs
Features enabled with new hardware

Body Zonal



Body Zonalization
- Wiring harness reduction, modularity introduced
- Primary / secondary body controllers

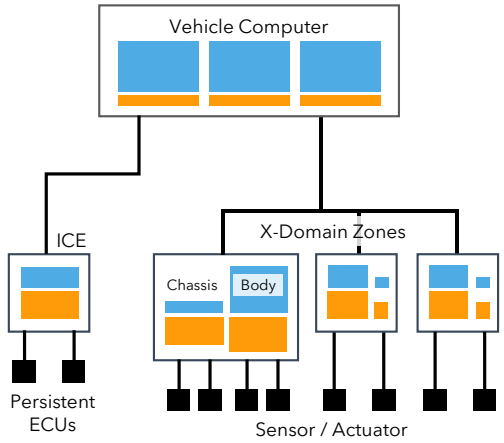
Domain + Body Zonal



Domain Controller / Vehicle Computer
- Higher level of control
- New features in software

Body Zonalization
- Wiring harness reduction, modularity introduced
- Primary / secondary body controllers

Cross-Domain Zonal

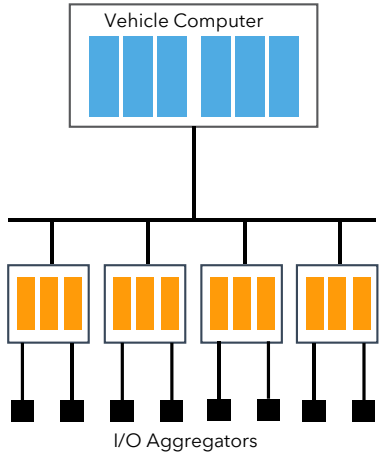


Vehicle Computer
- Consolidation, ECU reduction
- New features in software

X-Domain Zonalization
- Multi-tenancy, more ECU consolidation
- Compute local to zone

Persistent ECUs
- Limited value to migrate (e.g. too complex, lifespan)

Consolidated Compute



Vehicle Computer
- Hosts "All decision making"
- Fully flexible platform for SDV

I/O Aggregators
- Generic sensor / actuator control
- Available to multi-applications

Architectures focussed on vehicle infrastructure – exclude ADAS & Infotainment

Re-Architecting Software

RT COMPUTE
Function

First Step:
Identify how to split existing application software into RT compute vs I/O control

I/O CONTROL
Function

GOAL

Not bound by strict functional or geographical constraints

Maximum flexibility to co-locate these functions, less HW dependent, quicker to update and deploy

GOAL

I/O interface is application agnostic

Multi-consumers of sensors via an API call or service subscription

(e.g. Video stream, temp sensor, …)

I/O Interfaces

Networked (CAN, Eth)

Network

Direct

Raw I/O (ADC, Timer, Pixel)

Control Loops

Loosely Coupled

Higher

Lower

Tightly Coupled

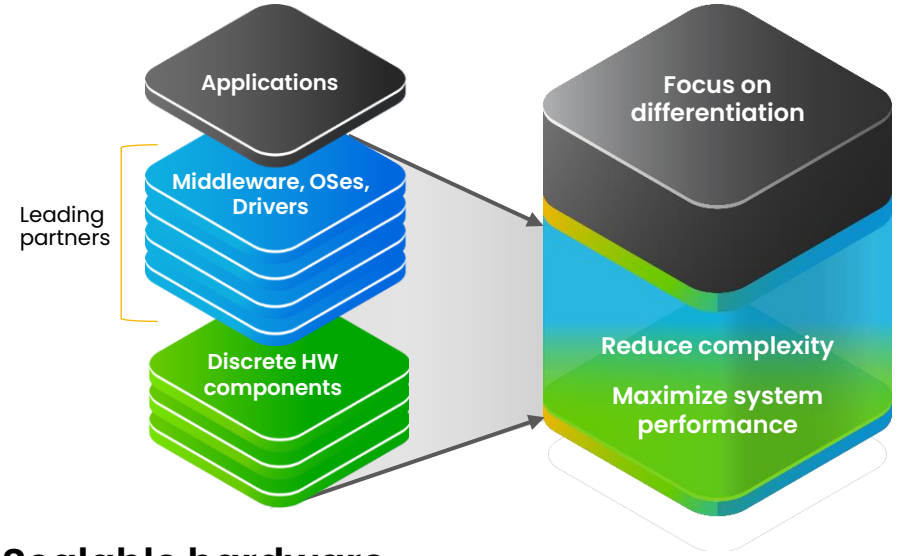
Introducing

S32 CoreRide platform

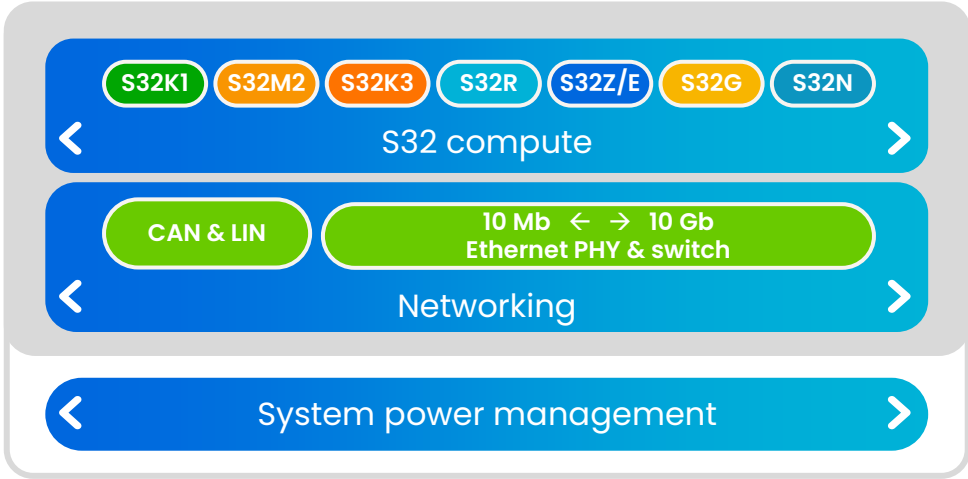
Open with unprecedented combination of scale and integration, paving the way to a new era of SDV development



Tightly integrated software and hardware

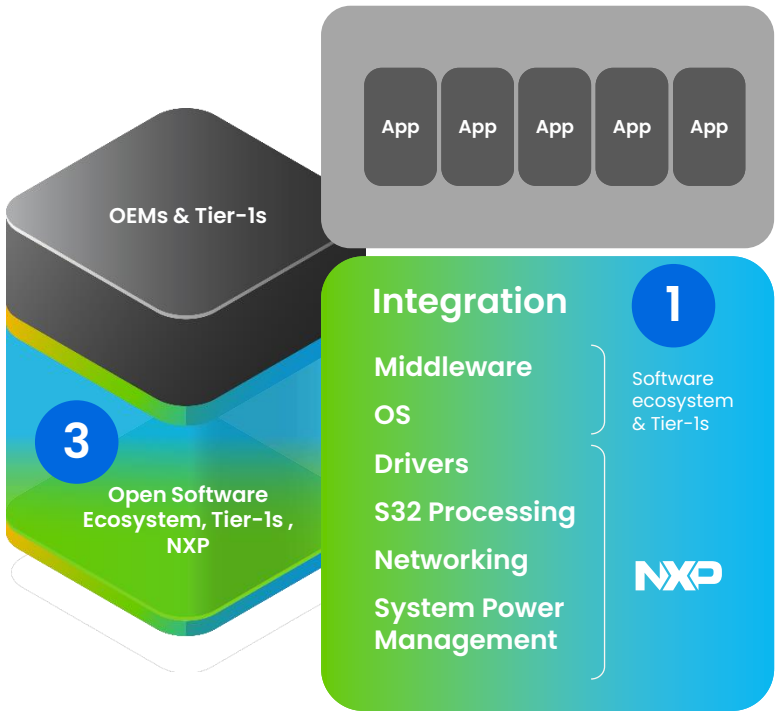


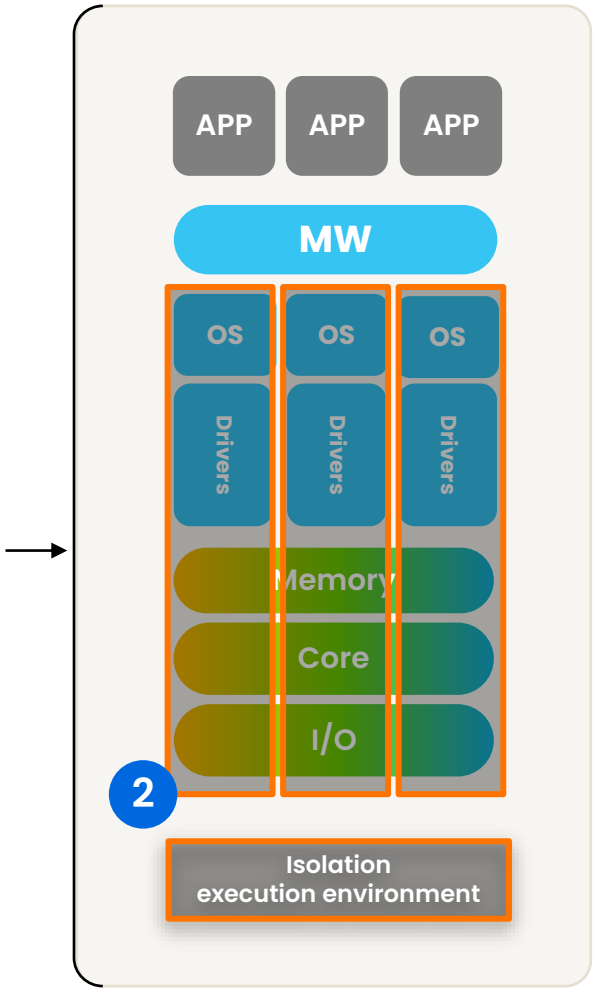
Scalable hardware



S32 CoreRide platform

Reducing integration efforts and enabling ECU consolidation through HW isolation





1 Software integration

- Integrated driver interfaces and OS solutions optimized for NXP SoCs
- Simplifies development and optimizes integration with production-grade software platforms

2 Isolation execution environments

- Software-defined, hardware-enforced, safe, secure, and efficient isolated runtime operation of ECU SW
- Hardware isolation of key SoC resources to ensure freedom of interference
- Allows for easy integration and redeployment of ECU functions

3 Open platform for partners and Tier-1s

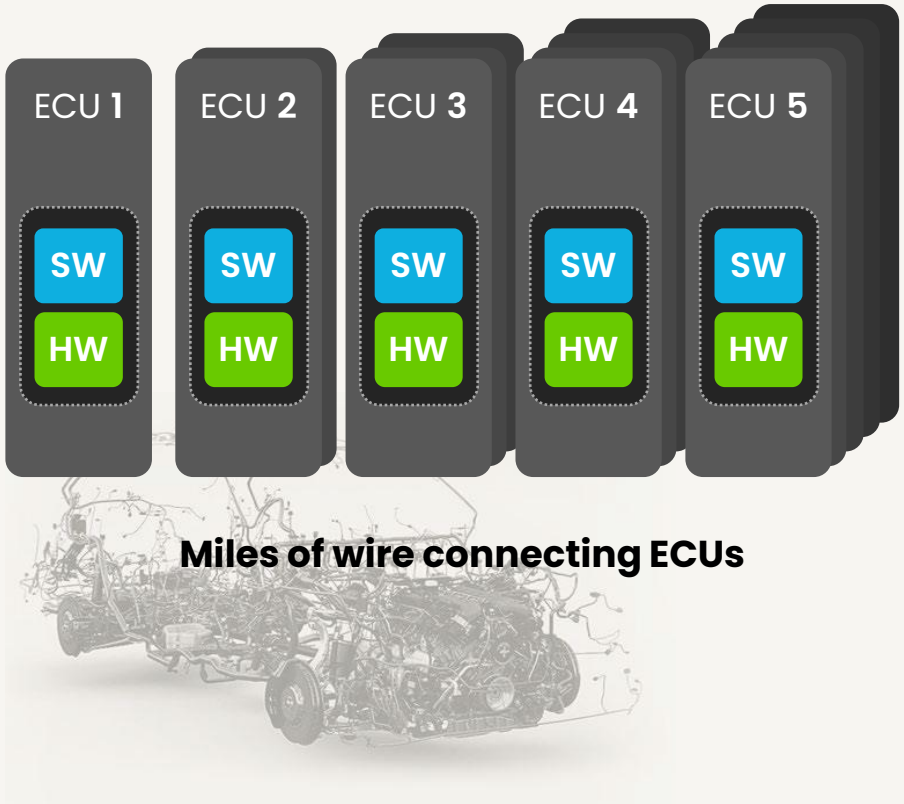
- Provides integration support for the software continuum
- Integration-ready software optimizes path to production
- Middleware provided by industry-proven partners, performance-optimized for NXP SoCs

NXP S32 CoreRide™ Platform

Enables safe and secure ECU consolidation

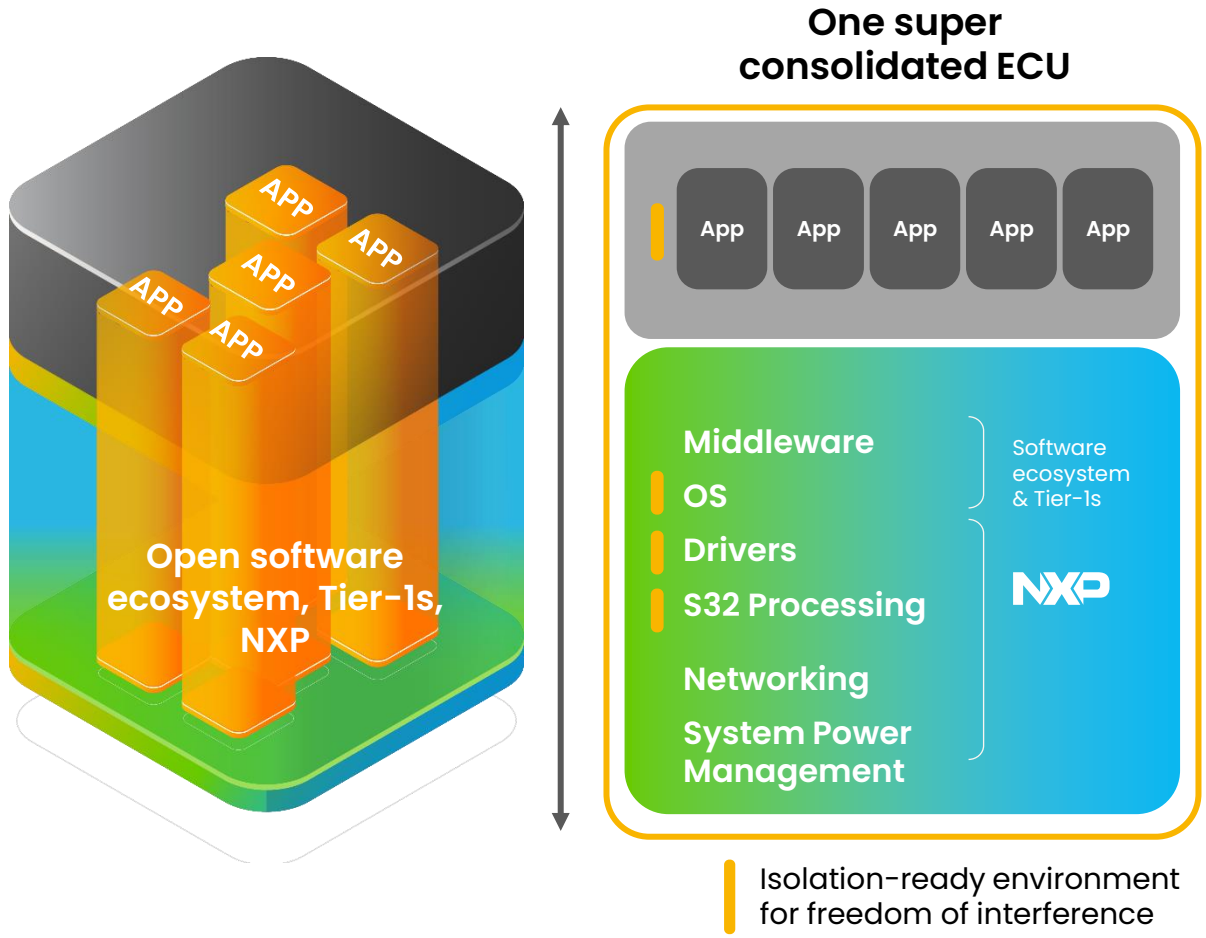
Traditional architecture

Multiple ECUs and integration efforts

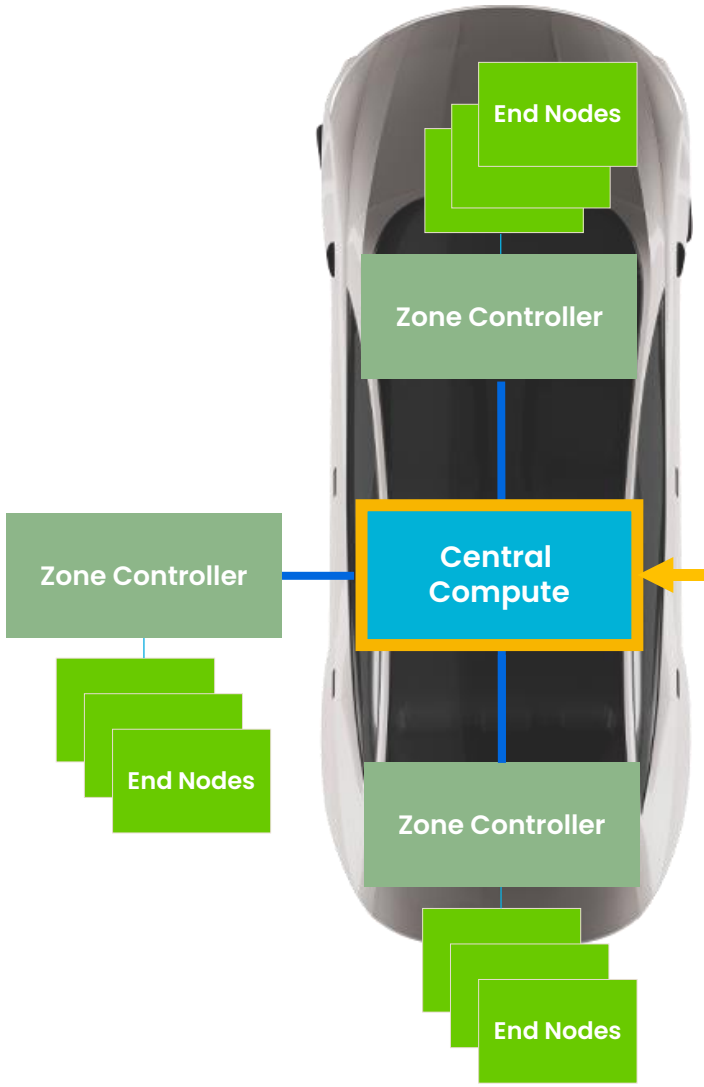


With S32 CoreRide™ Platform

Consolidate ECUs and reduce integration efforts



S32N: The heart of NXP's S32 CoreRide central compute platform



Safe, super-integration of vehicle functions

Hardware isolation and virtualization

Freedom from interference



Scalable real-time and applications compute

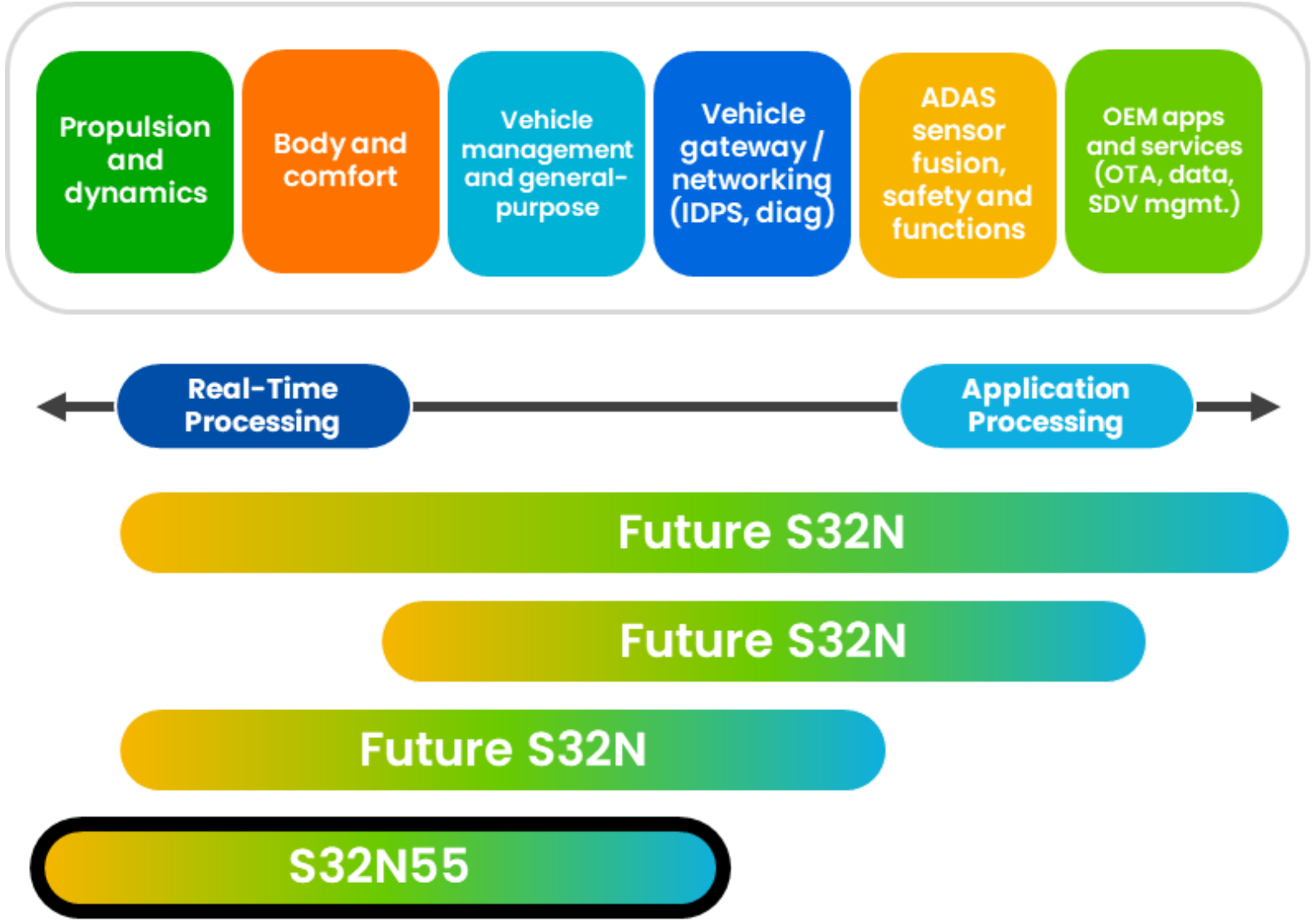
Pre-integrated software

Networking and system power management

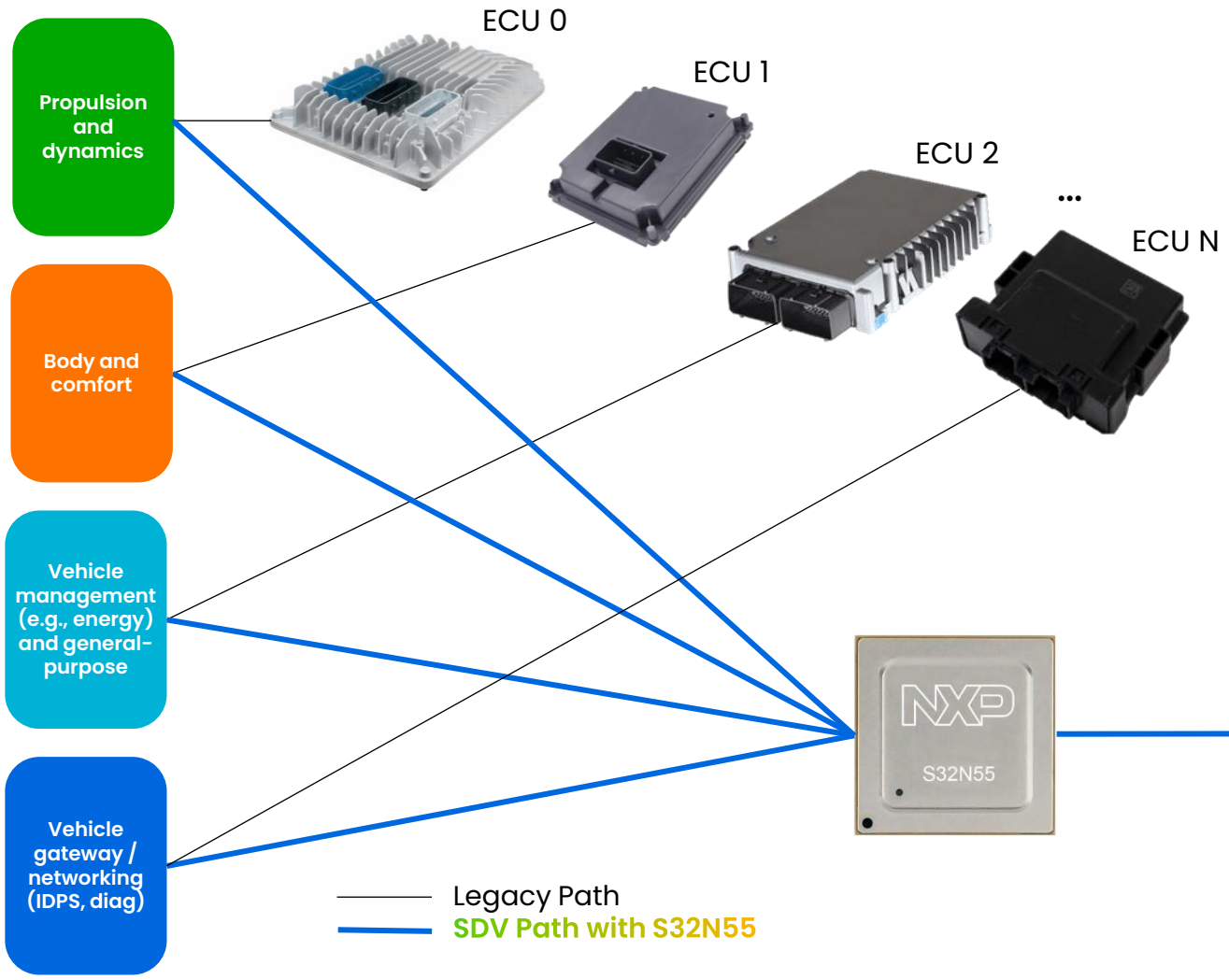
Core Vehicle Functions

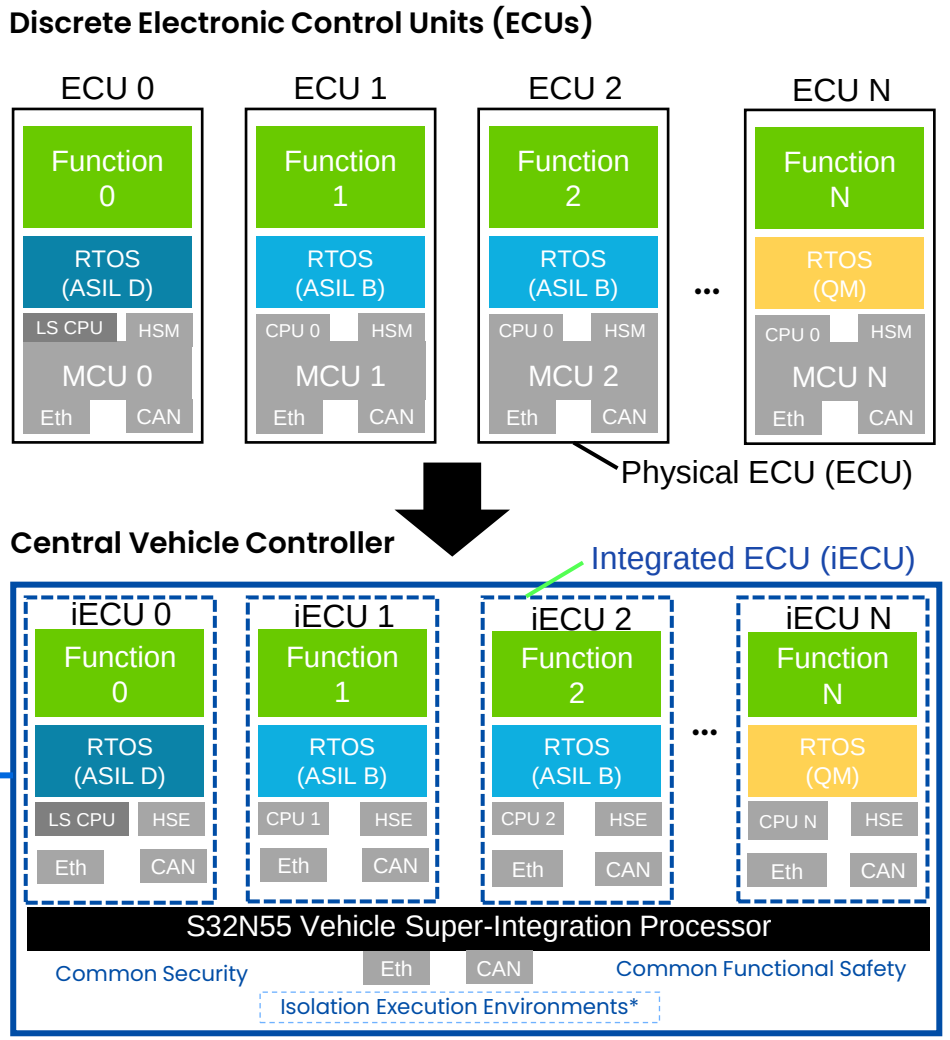
- Propulsion and dynamics
- Body and comfort
- Vehicle management and general-purpose
- Vehicle gateway / networking (IDPS, diag)
- ADAS sensor fusion, safety and functions
- OEM apps and services (OTA, data, SDV mgmt.)

S32N Family: Super-integration of cross-vehicle functions



NXP S32N55 for real-time super-integration of vehicle functions





* Environments can be extended using multi-core and multi-OS for more complex iECUs.

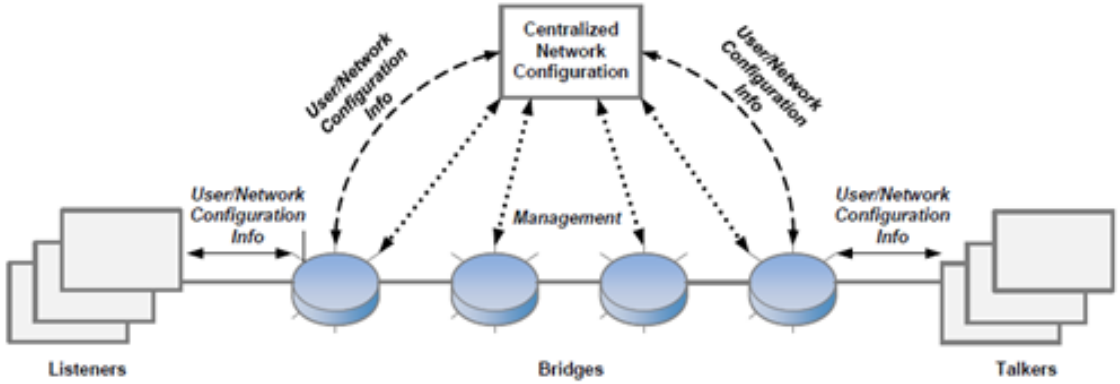


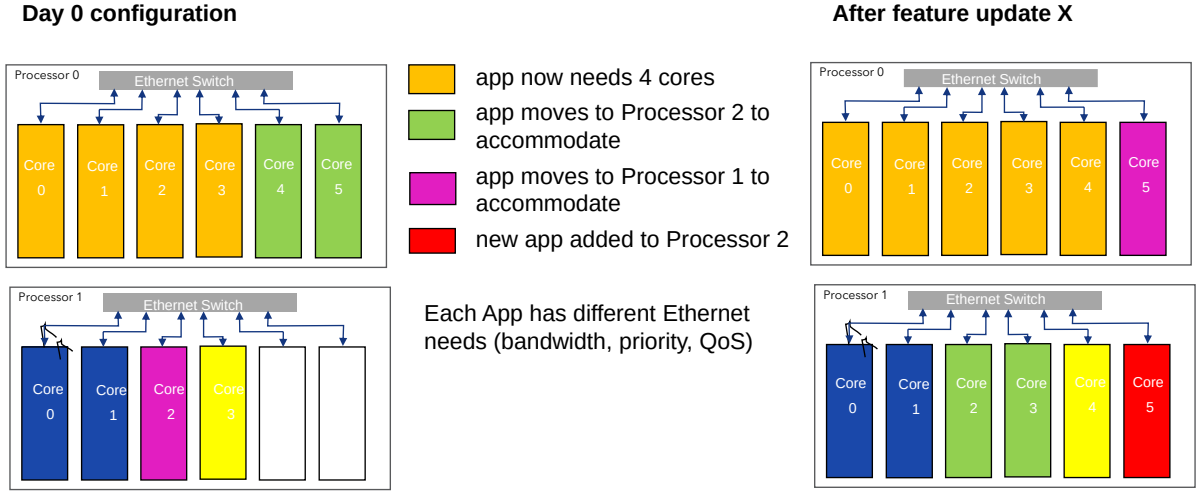
Agenda

- What is a Software-Defined Vehicle?
- How to Build a Software-Defined Vehicle
  1. Select optimal E/E architecture
  2. Re-architect software
  3. Adopt new Ethernet technology
  4. Enable vehicle-to-cloud software
  5. Use virtual modelling
  6. Select the right partner(s)
- Conclusion

Software-defined networking

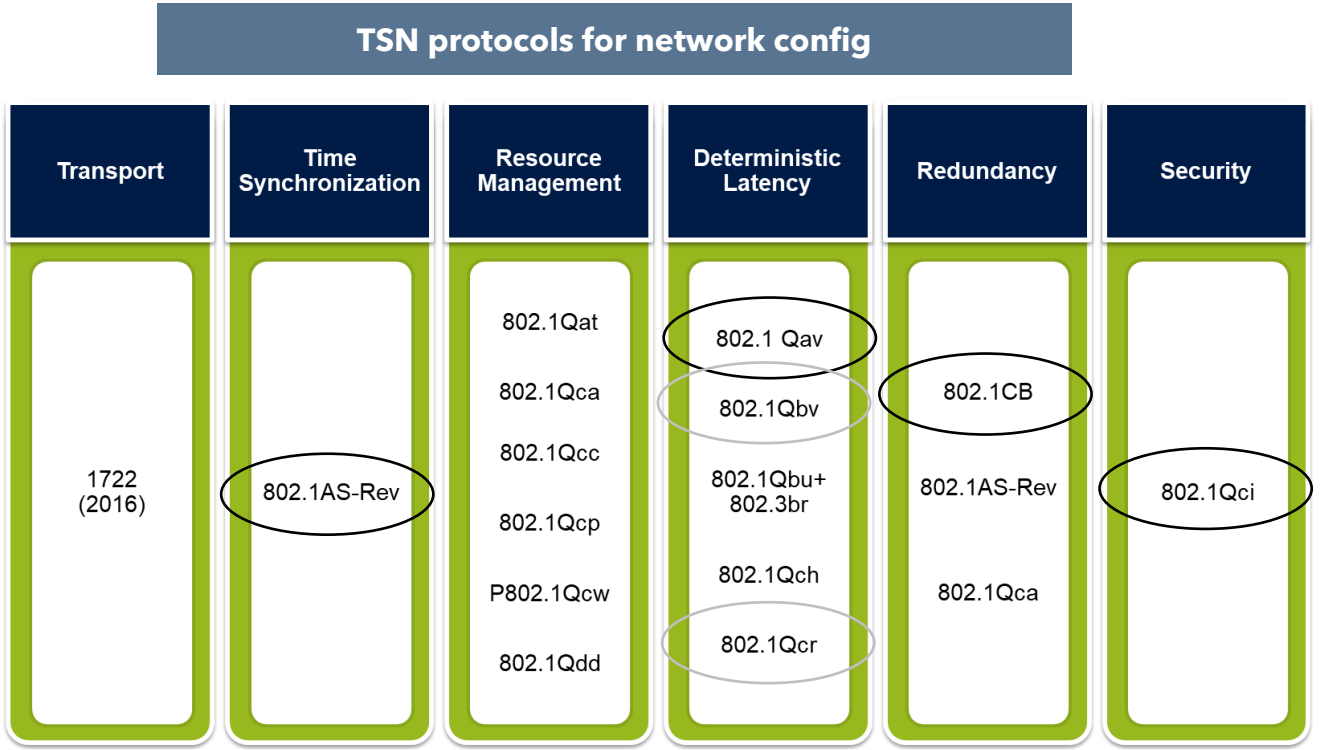
- Trends of modern vehicle networks
  - OTA service updates and dynamic communication patterns
  - Every networking component must be programmed with the correct routing matrix, policies and TSN features
- These dynamics are increasing in complexity
  - Networks have nodes with different mix of criticality, scaled and heterogenous hardware
  - Very high configuration complexity
- Software-Defined Networking (SDN) shows promise as future technology by
  - Providing a global view on abstracted network elements topology and state
  - Enabling automation, elasticity & scalability
  - YANG data model approach that is proven in IT and Data Centres





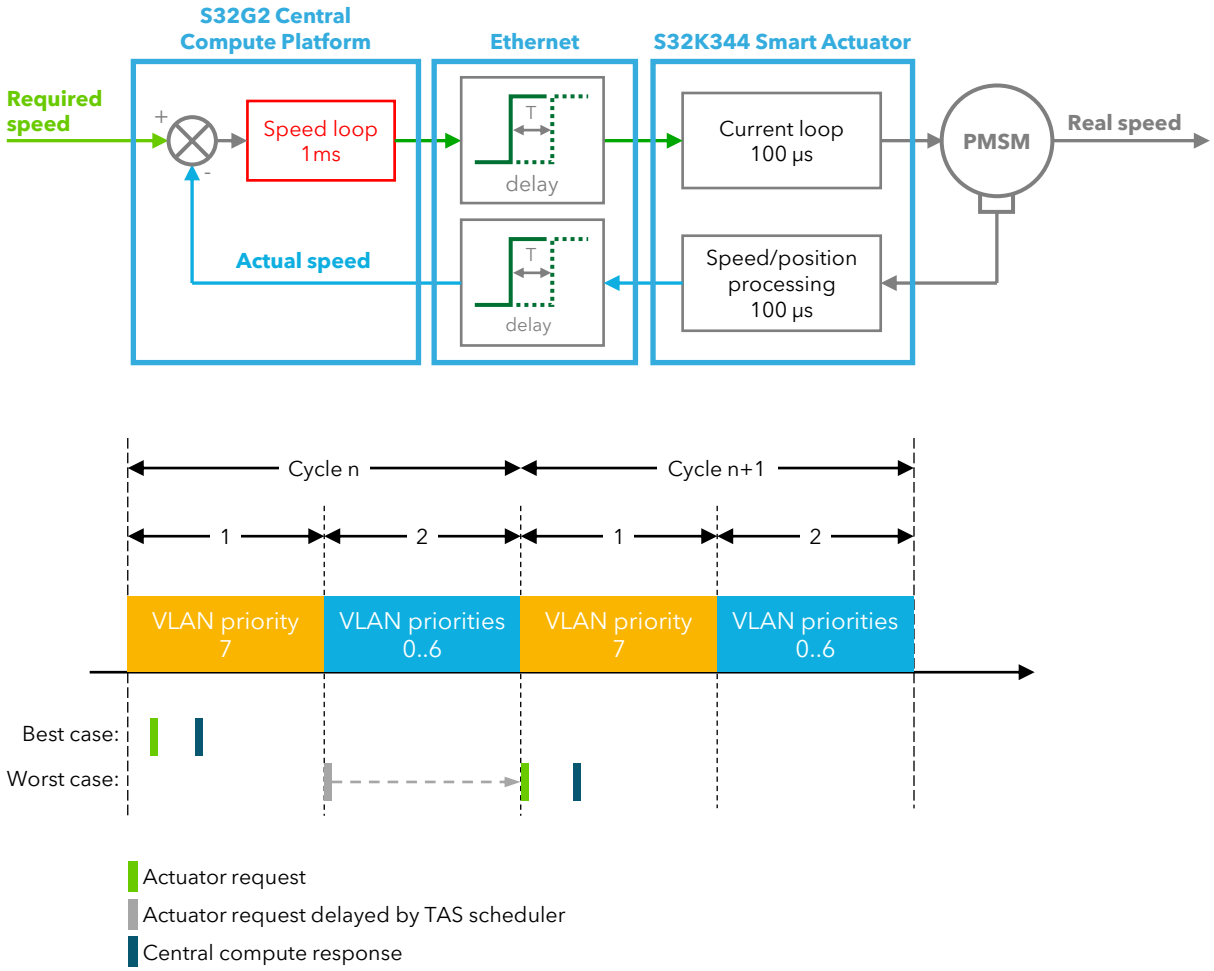
Ethernet TSN

- Functions are distributed within the software-defined vehicle
- Ethernet has been universally adopted for the vehicle backbone
- Network becomes critical for success vehicle operation, and must be configured to provide system-required policies
  - time synchronization (e.g. audio, blinkers)
  - bounded latency (e.g. steering)
  - guaranteed message delivery (e.g. airbag)
  - QoS (priorities, credits, slots, pre-emption)
  - redundancy (L3+ related apps)
  - security (L3+ related apps)
- Configuring multi-bridge network is highly complex, risky and time-consuming
- Require in-life network upgrading / re-configuration as features are added



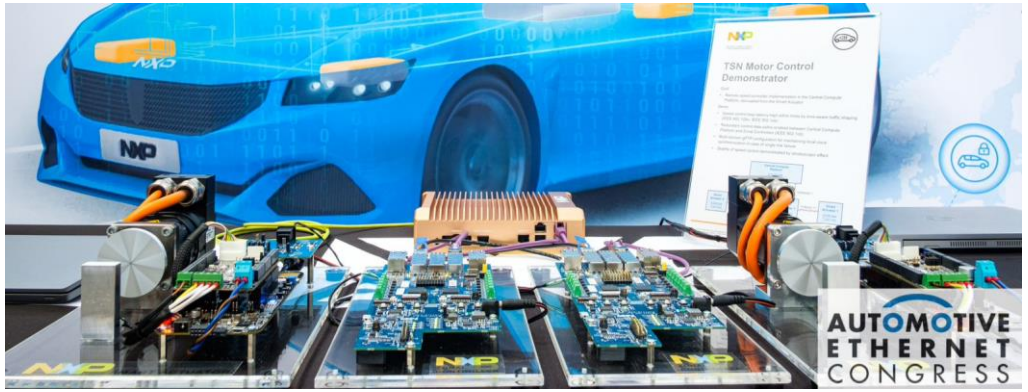
- **802.1AS** (gPTP) – time synchronization
- **802.1CB** – enabling redundant control data paths
- **802.1Qci** – per stream filtering and policing (AUTOSAR included this in the R22-11 standard)
- **802.1Qav** – Credit Based Shaper – already used as part of AVB standard
- **802.1Qbv** & **802.1Qcr** have interest but are complex to implement, under investigation within AUTOSAR

Realtime / latency challenge
e.g. 3-phase permanent-magnet synchronous motor



Motor control over a network demonstrated by:

- Redundant control data paths enabled between central compute platform and zonal controllers (IEEE 802.1CB)
- Multi-domain gPTP (generalized precision time protocol – IEEE 802.1AS) configuration for maintaining the local clock synchronization in case of a single link failure between central compute platform and one of the zonal controllers
- TAS (time-aware shaper – IEEE 802.1Qbv) scheduler configuration example with cycle time of 500µs split evenly into time intervals:
  - Time interval 1: 250 µs, VLAN priority 7 (speed control)
  - Time interval 2: 250 µs, VLAN priorities 0..6 (other traffic)





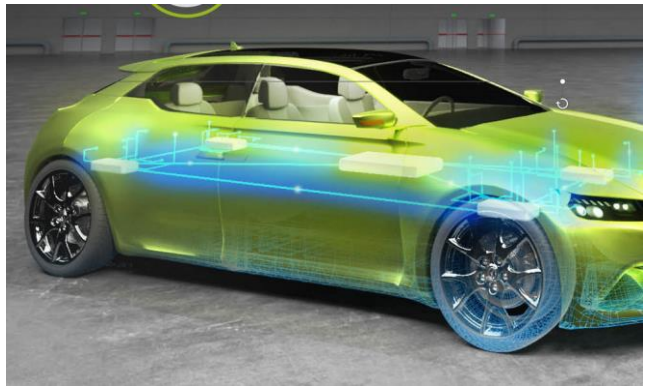
Agenda

- What is a Software-Defined Vehicle?
- How to Build a Software-Defined Vehicle
  1. Select optimal E/E architecture
  2. Re-architect software
  3. Adopt new Ethernet technology
  4. Enable vehicle-to-cloud software
  5. Use virtual modelling
  6. Select the right partner(s)
- Conclusion

Software for the Software-Defined Vehicle

- Software platform abstracting complex vehicle network
- Managing updates across vehicle

- Decoupling hardware and software development process and lifecycle
- Enabling ecosystem



Applications

Platform services

Middleware

Core (low-level) software

Hardware



Software factory

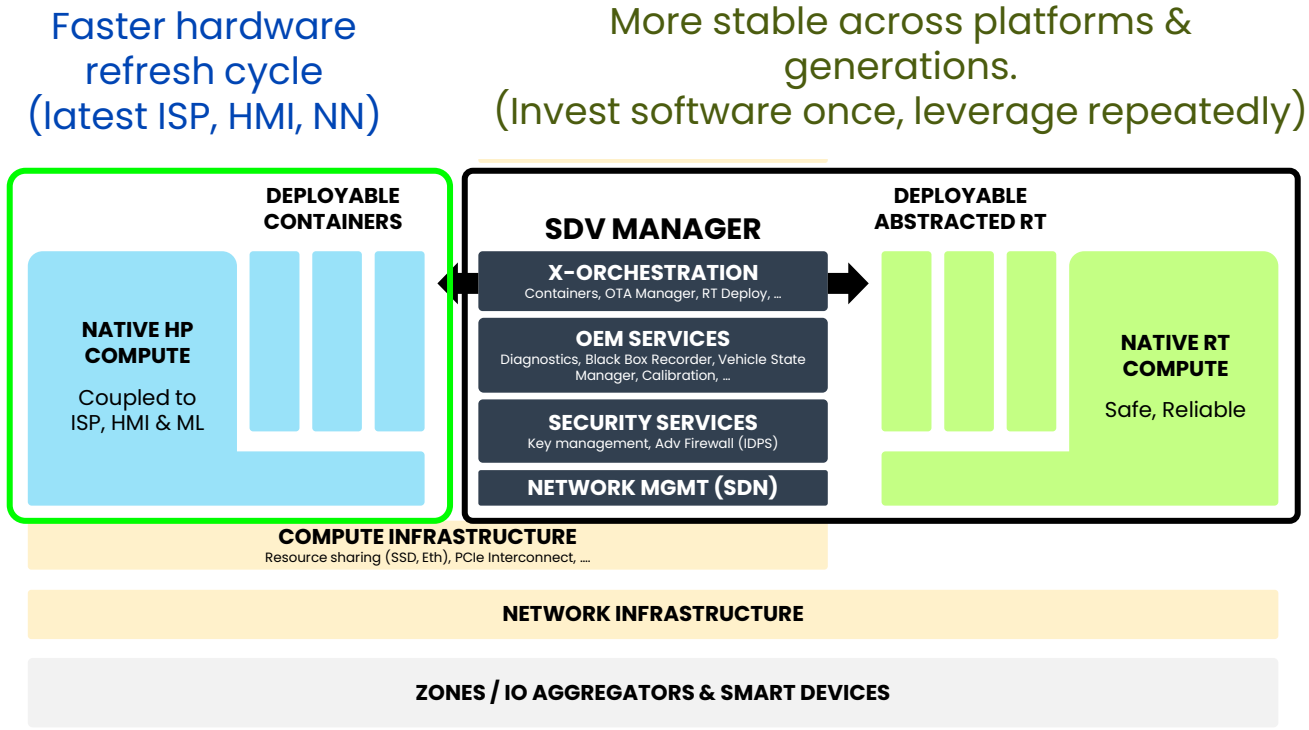
- Automates development, testing, integration & testing
- Core software & applications

Cloud services

- Configuration
- Lifecycle management
- Digital twin

The Software-Defined Vehicle manager: Separate infrastructure from features

- **Flexibility & stability**
  - Diverging lifecycle & variability of infrastructure vs compute features
  - Optimize software investment in stable infrastructure
- **Improved security**
  - Strong firewall between consumer-oriented features & OEM services (e.g. User Apps vs OTA manager)
  - Optimized networking & security hardware
  - Networking & hardware offload engines – dedicated focus toward vehicle infrastructure
- **Safety**
  - Safety 'System Manager' for Central ECU
  - ASIL-D features & strong hardware isolation
- **Power / Wake-up partitioning**
  - Addressing <50ms wakeup requirements for first network packets
  - Access to "key-off" features, without waking full HP Compute

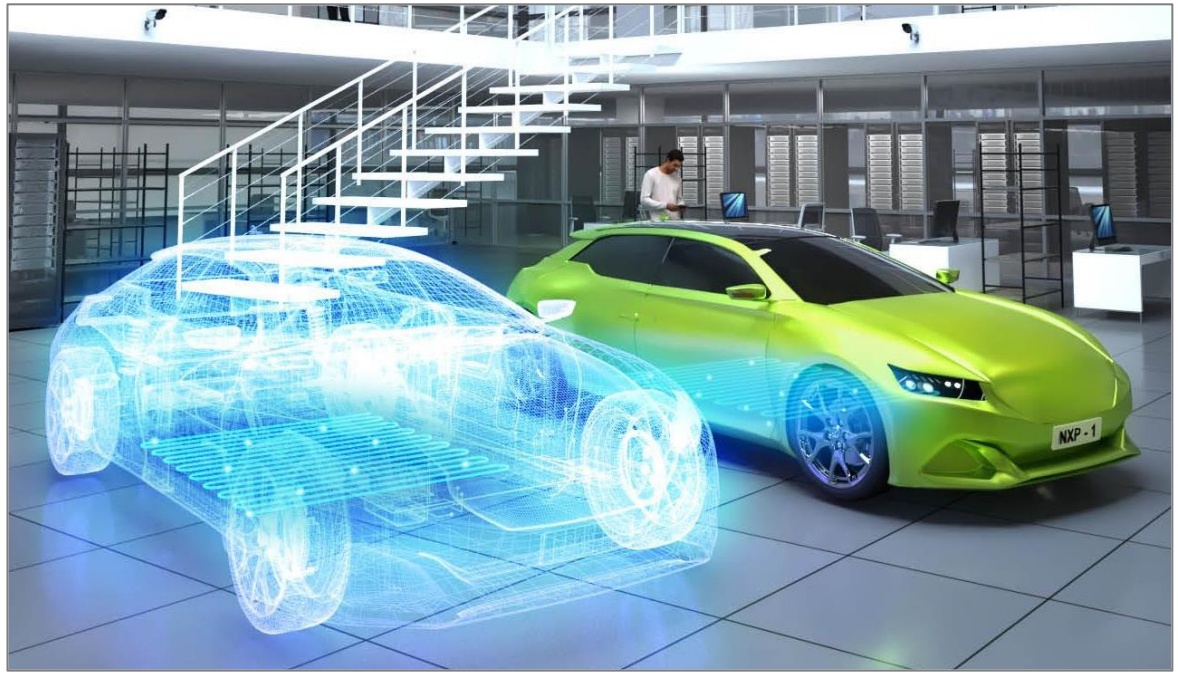




Agenda

- What is a Software-Defined Vehicle?
- How to Build a Software-Defined Vehicle
  1. Select optimal E/E architecture
  2. Re-architect software
  3. Adopt new Ethernet technology
  4. Enable vehicle-to-cloud software
  5. Use virtual modelling
  6. Select the right partner(s)
- Conclusion

Use of virtual models



System complexity is increasing,
performance is increasing
software complexity is increasing

Shift left

Tier-one suppliers and OEMs need to start software development as early as possible

Industry moving to the SDV

means a focus on virtual tooling – used throughout the lifetime of the device, for development, testing and deployment

SHIFT LEFT
9 – 12 months

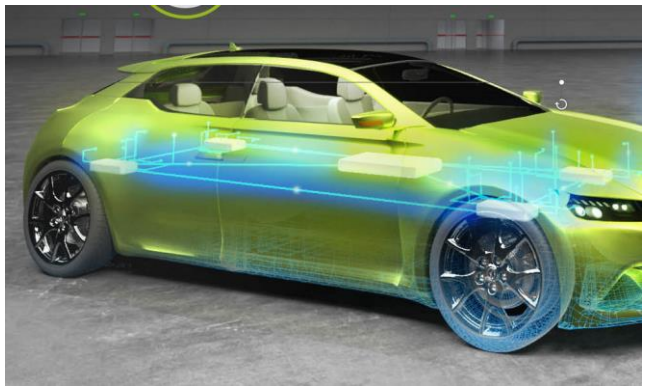
SILICON AVAILABILITY

STRETCH RIGHT
1 – 10 years

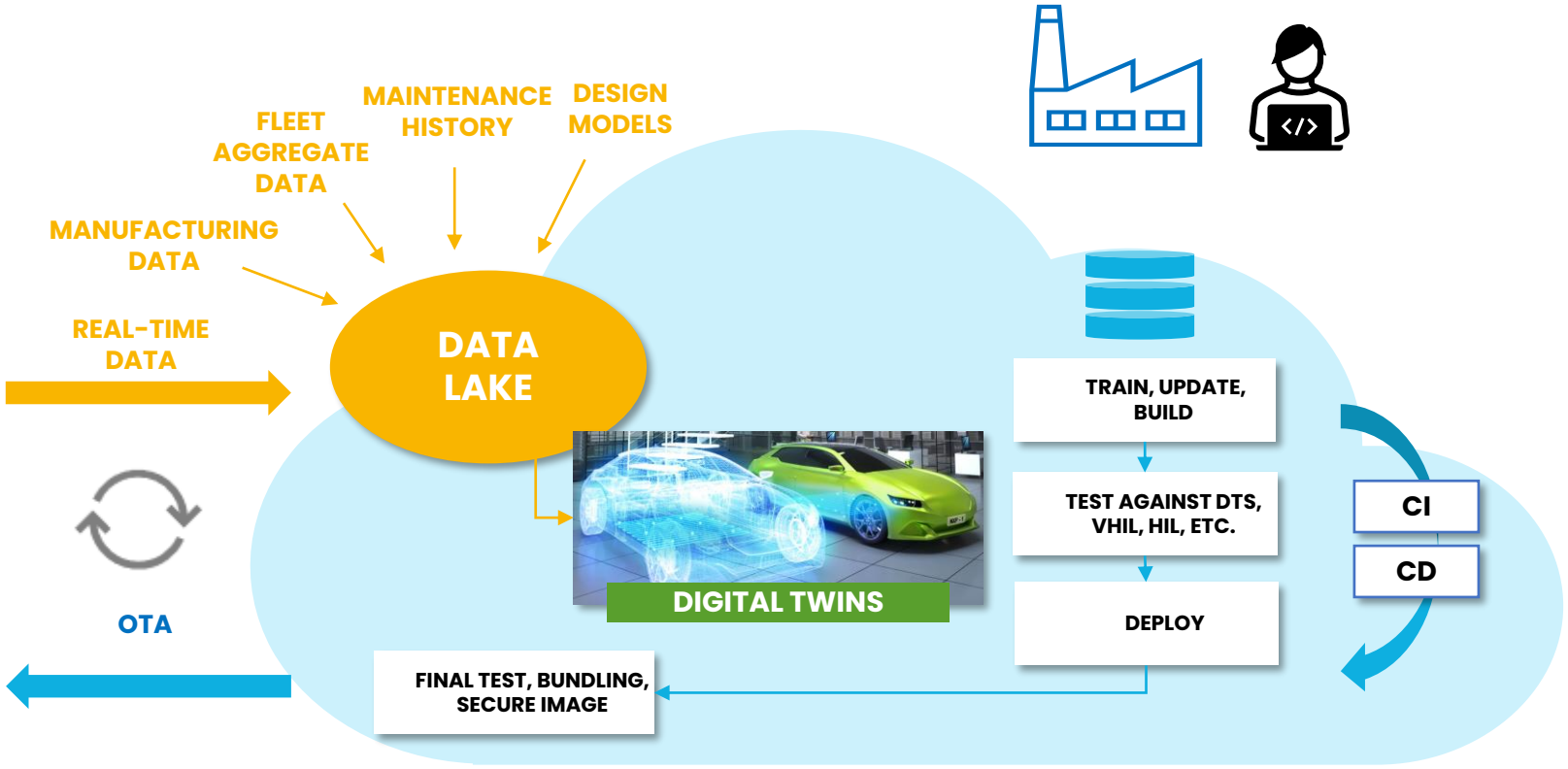
Use Case: Early Software development

Use Case: Regressions running on virtual prototypes as digital twins

Digital Twins in Automotive



Physical Assets
(vehicle sensor data)



DevOps Pipeline

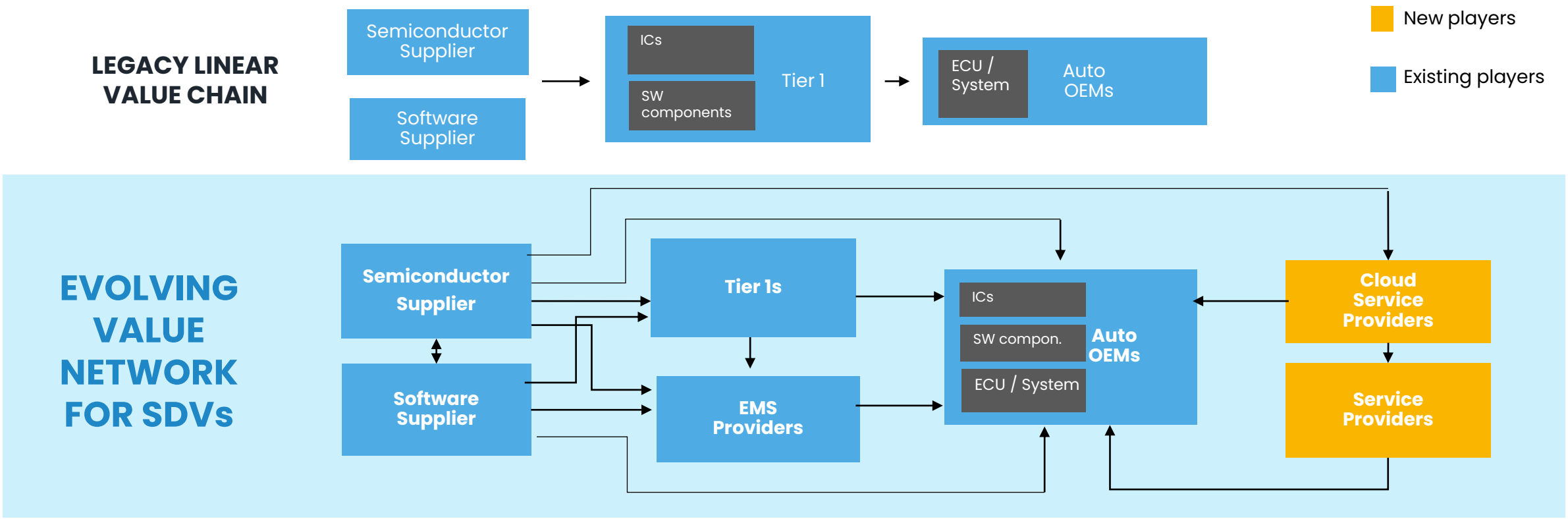
Data lifecycle and code lifecycle combine to create continual vehicle upgrades



Agenda

- What is a Software-Defined Vehicle?
- How to Build a Software-Defined Vehicle
  1. Select optimal E/E architecture
  2. Re-architect software
  3. Adopt new Ethernet technology
  4. Enable vehicle-to-cloud software
  5. Use virtual modelling
  6. Select the right partner(s)
- Conclusion

Value chain evolving from linear to value network



To drive change, OEM is more involved in electronics – influencing silicon and software

Supply chain must evolve to meet OEM requirements – tier-one suppliers & semiconductor supplier roles are changing

EMS providers becoming more important

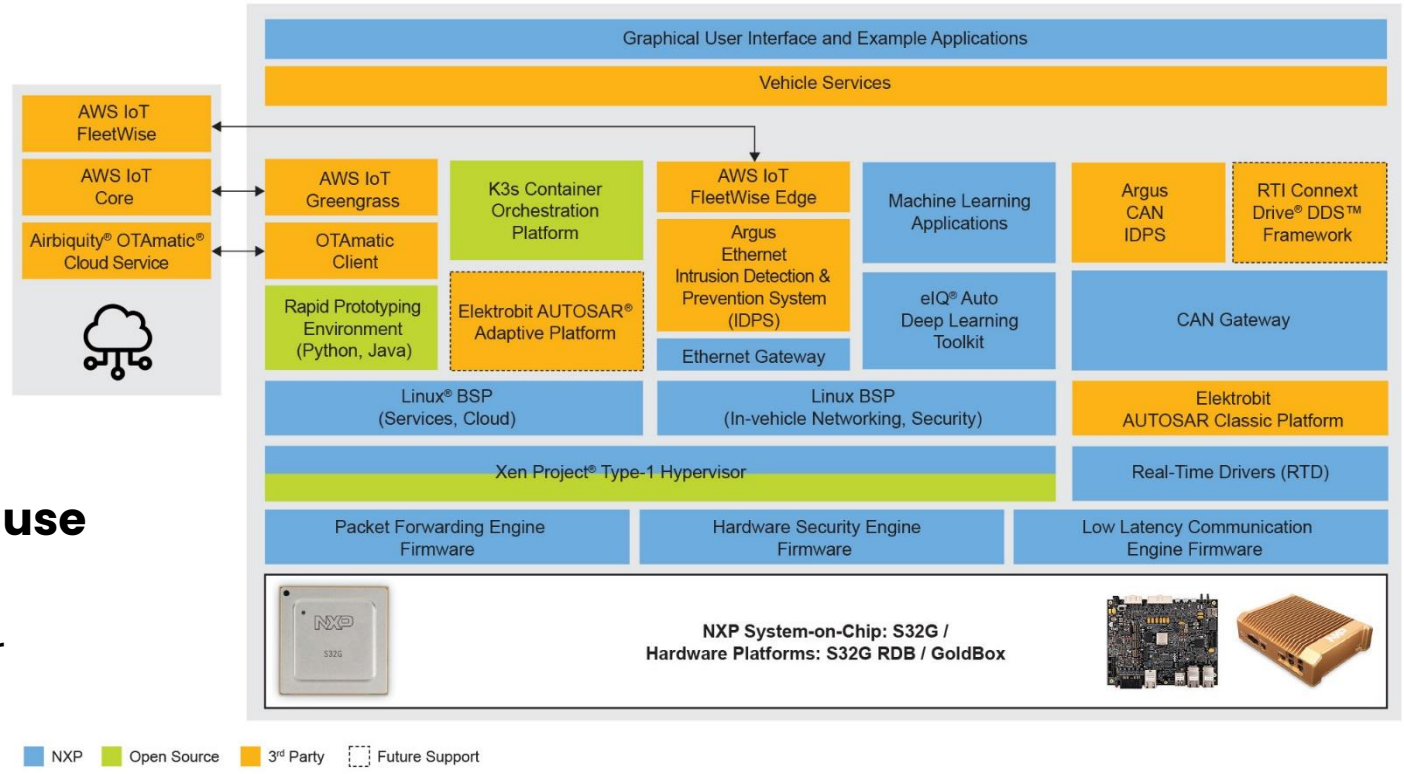
VIP: Vehicle integration platform

Reference platform for vehicle network processing applications, including service-oriented gateways

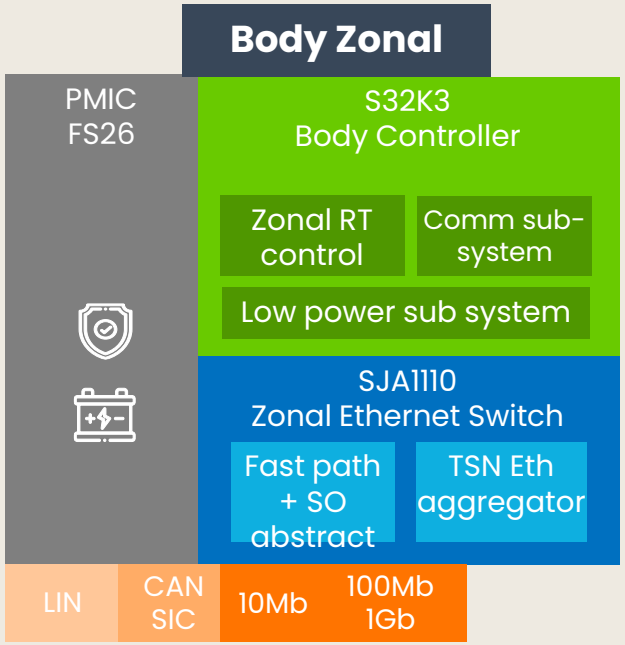
- **Addresses key use cases & trends**:
  - Vehicle Networking
  - Service-Oriented Architecture (SoA)
  - Data Analytics
  - Over-the-Air (OTA) Services
  - Virtualization & Isolation
  - Network Security and Security Services

- **Accelerates development through ease of use**
  - Demonstrate KVPs of the S32G2 in 10 mins
  - Customers to create a prototype gateway in 1hr
  - Abstraction of hardware complexity
  - Minimize support through software reference

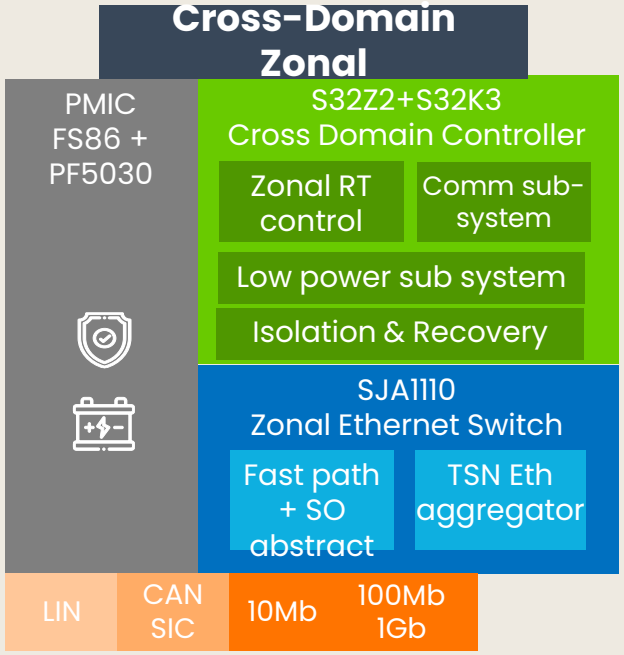
GoldVIP for S32G



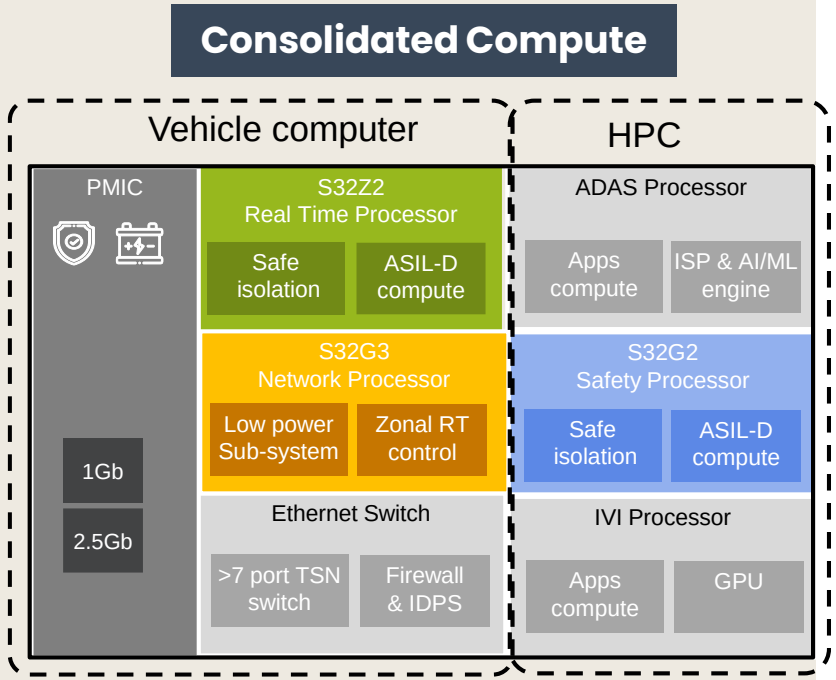
NXP SDV solutions landscape



- Three-chip solution (plus PHYs)
- High port and I/O processing scalability
- Smart power distribution, fast wake up
- Strong safety + security



- Three-chip solution (plus PHYs)
- Strong on-chip isolation & recovery
- High port and CAN scalability
- Powerful network, smart power distribution
- Strong safety + security



- Multi-chip full-function HPC solution
- Can be built as single box (connected by PCIe and/or Ethernet) or multiple boxes
- Strong safety + security

COMMON CONFIGURATION & SOFTWARE

Broad partner network with strong automotive expertise

- The Automotive Processing Partner Program comprises a global network of independent companies that offer the vital tools, software, technology, engineering services and training to accelerate customer designs based on NXP components.

Engineering ecosystem:
software and hardware engineering services, consultancy and training on products, safety and security and more.

Application ecosystem:
application software and networking stacks.

Enablement ecosystem:
OS & hypervisors, development tools, virtualization and modelling tools.

Engineering Partners

诚迈科技 ARCHERMIND, eInfochips, 映驰科技 Enjoy Move, CETITEC, DASAN, HF Automation Systems, HORIBA Automotive, i&m Ideas&Motion, LINEO Solutions, Lorit Consultancy, Luxoft, Megatronix, MicroSys, Mobile Knowledge, MOVE.B, PhoenixMicro Inc., Quest global, SONATUS, technica engineering, TESSOLVE, VOLANSYS Trusted Technology Partner

Application Partners

aicas, Airbiquity, ALGOLION, ARGUS CYBER SECURITY, aws, etas embedded communication, EmSA, excelfore, DYNAMO EDGE, GUARD KNOX, MOTER TECHNOLOGIES, INC., PROVENRUN, QUESTAR Driving Knowledge, rti, SIBROS, Simma Software, TERAKI, TRUSTONIC

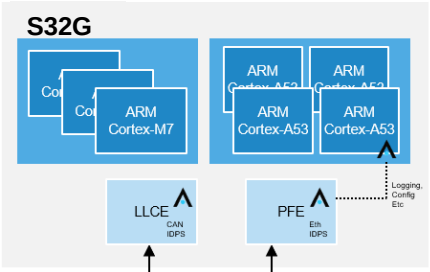
Enablement Partners

CETC 普华基础软件, Elektrobit, ETAS, Green Hills SOFTWARE, HAILO Empowering Intelligence, HIGHTEC, IAR SYSTEMS, iSYSTEM, KEIL Tools by ARM, LAUTERBACH DEVELOPMENT TOOLS, LDRA, MathWorks, Neusoft, OPENSYNERGY, PE micro, pls Development Tools, SEGGER, SYSGO EMBEDDING INNOVATIONS, SYNOPSYS, TASKING, VECTOR, WITTENSTEIN, WNDRVR

Expanding ecosystem partnerships

Intrusion Detection and Prevention (IDPS)

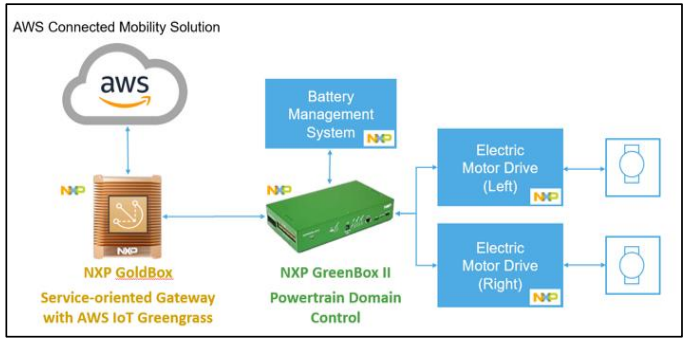
- Argus Cyber Security offers advanced IDPS capabilities that leverage S32G PFE and LLCE hardware accelerators and cloud
- Provide extended intrusion detection and minimize processing on Arm Cortex-A53 and Cortex-M7 cores



ARGUS CYBER SECURITY

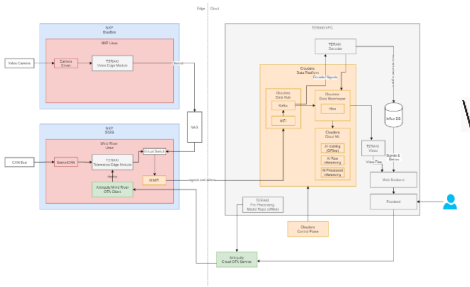
Connected EV Management System

- Joint NXP-AWS solution that showcases real-time connected EV management capabilities
- Unlocks the value of real-time vehicle data benefits
- Establish a foundation to extend use cases for vehicle-wide support



The Fusion Project Data Lifecycle with AI/ML (NXP, Teraki, Wind River, Cloudera, Airbiquity)

- State-of-the-art hardware, software, and cloud platform for data collection, analysis, and OTA updates.
- Starting with autonomous driving use cases, but supports others use cases that require an end-to-end AI/ML solution.



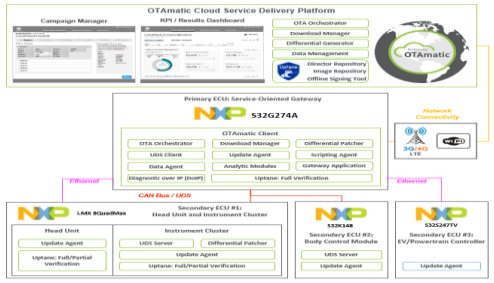
TERAKI

WNDRVR

Airbiquity

Vehicle-wide Over-the-Air (OTA) Services

- S32G updates vehicle-wide ECUs using Airbiquity OTAMatic comprehensive cloud-to-vehicle solution
- Demo includes all NXP hardware (S32G,S32S,S32K,i.MX 8)



Airbiquity

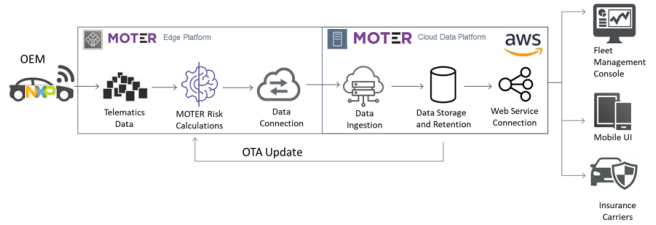
AI - Vehicle Health Management (VHM)

- Enables customers to design, test and deploy AI/deep learning-based models and software that manage vehicle health



InsurTech for Usage Based Insurance

- Integration of real-time UBI algorithms and processing in the vehicle using S32G vehicle edge hardware
- Enables data-driven insurance, smart crash handling, new and improved auto insurance, enhanced risk management and safety improvement



MOTER



Agenda

- What is a Software-Defined Vehicle?
- How to Build a Software-Defined Vehicle
  1. Select optimal E/E architecture
  2. Re-architect software
  3. Adopt new Ethernet technology
  4. Enable vehicle-to-cloud software
  5. Use virtual modelling
  6. Select the right partner(s)
- Conclusion

The Software-Defined Vehicle

Software….

defines the customer experience

becomes the differentiator

is the focus point for innovation

brings OEMs new revenue streams

Hardware….

is a stable scalable future-proof hardware platform

New features include…

car & cloud integration

over-the-air updates

flexibility to continuously evolve

service-oriented architectures

virtualization and containerization

New processes are…

DevOps: Continuous integrate, test and deploy of software

Agile and lean methods

Digital twin virtual modelling

HOW TO BUILD A SOFTWARE-DEFINED VEHICLE?

1. Select optimal E/E architecture
2. Re-architect software
3. Adopt new Ethernet technology
4. Vehicle-to-cloud software
5. Use of virtual modelling
6. Select the right partner(s)
7. Celebrate success!



Technical Session Survey

Thank you for your feedback.

Tour our immersive all-digital technology showroom from anywhere in the world, in just one click.

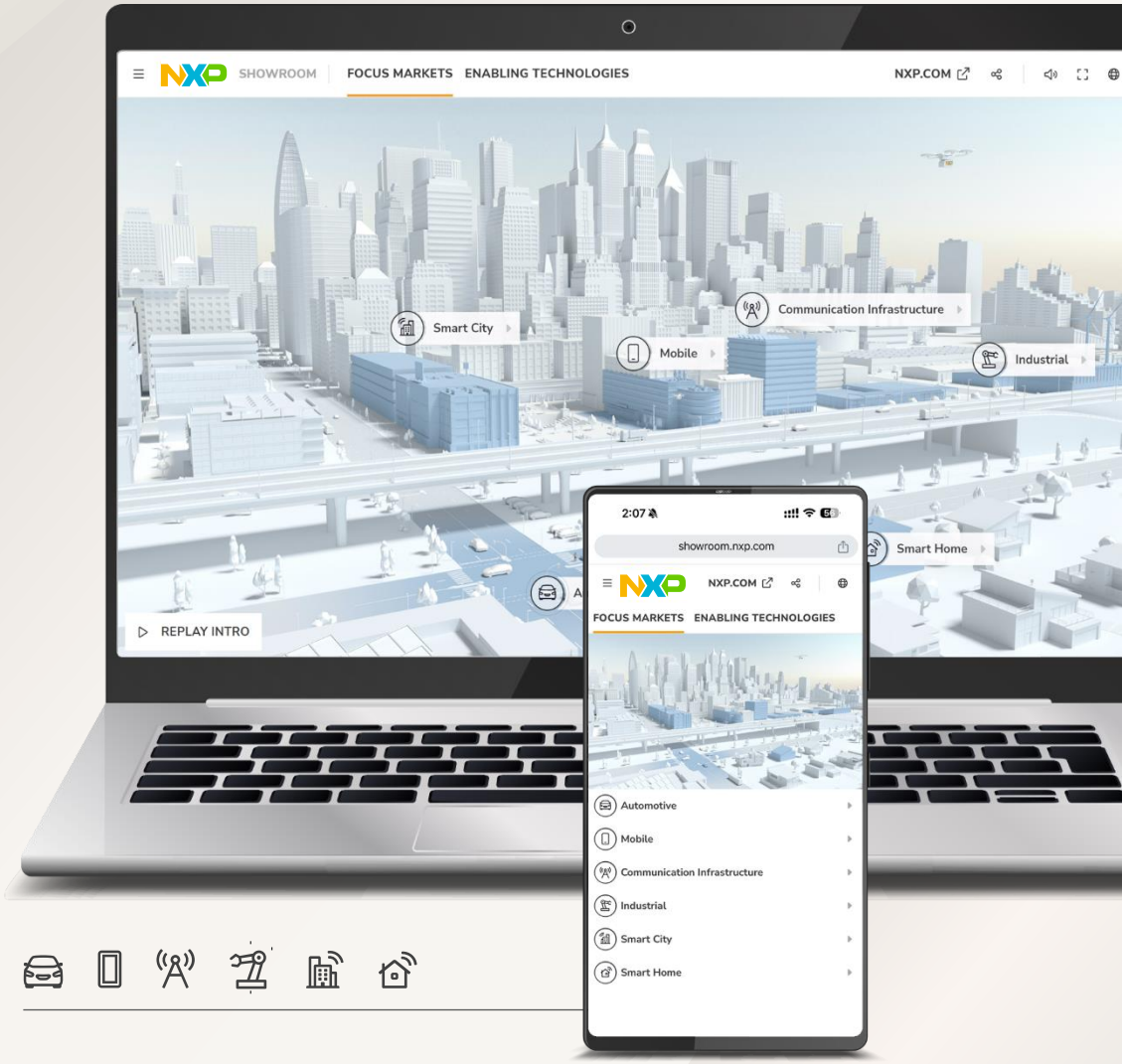
Journeys | focus
Automotive
Industrial & IoT
Mobile
Communication Infrastructure
Smart city
Smart home

Journeys | enabling technologies
Edge & AI/ML
Safety & security
Connectivity
Advanced analog
Sustainability
Low power innovations

Journeys | engagement
Self-guided tour
Live-streaming at set times
Guided tours

40+ virtual demos
Focus on system solutions
Set up along NXP verticals

showroom.nxp.com





nxp.com

| Public | NXP and the NXP logo are trademarks of NXP B.V. All other product or service names are the property of their respective owners. © 2024 NXP B.V.