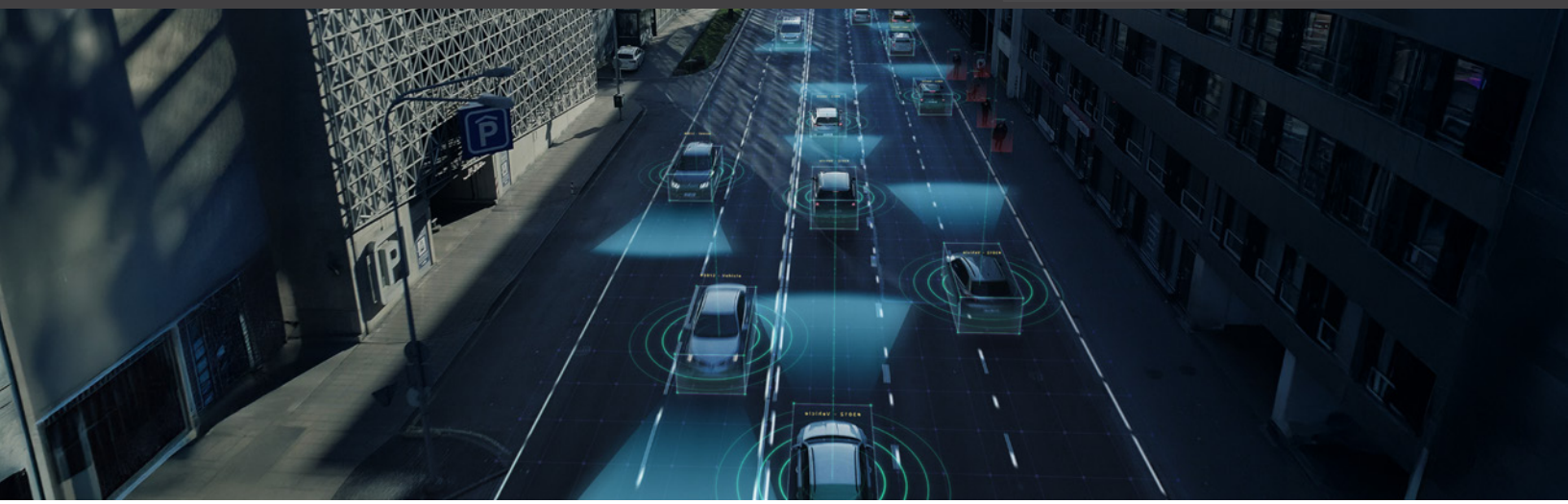

A Cross-Domain Software Infrastructure Platform is Necessary for Cloud-Native SDVs

As vehicles evolve into software-defined systems on wheels, consumers are demanding more personalized experiences from those vehicles. That means the software needs to do more: It has to act as an intelligent edge device, uniting functions and data not just from various domains within the vehicle but also from the cloud.

Whether the functions reside in a different application or operating system or in a cloud-based data center thousands of miles away, they all have to be easily and seamlessly linked together and remain responsive without experiencing life-threatening latency.

The solution is a comprehensive software infrastructure platform that is modular, hardware-agnostic and cross-domain, allowing it to be flexible, future-resilient, safe and secure. It must provide an open software framework with a developer-friendly environment that embeds edge analytics, containers and over-the-air (OTA) update capabilities for continuous and sustainable lifecycle management.

This approach is both more cost-effective and more adaptable than current alternatives and will lead the automotive industry and other mission-critical industries into the software-defined future.

A SOFTWARE PLATFORM FOR A NEW WORLD

Vehicle software applications used to be confined to individual domains, such as infotainment, ADAS, powertrain and body control. Each acted independently in individual electronic control units (ECUs) that were often dedicated to a single function and could not be updated easily.

Today, the world of vehicle software is much broader. Cloud connectivity can enable OTA updates, offload compute-intensive tasks and aggregate data from multiple sources, including other vehicles, to deliver advanced services and analytics. These capabilities improve performance and reduce hardware costs, but they also introduce significant challenges in coordinating real-time operations without latency risks.

What is needed is a unified, cross-domain and cloud-native software platform to enable consistent communication, dynamic updates, remote diagnostics and efficient lifecycle management — a platform that unites fragmented systems to achieve better scalability and decrease development complexity and cost. This platform must balance workloads across systems, enable seamless communication between services, optimize resource use, reduce complexity and accelerate innovation across the vehicle.

The traditional approach is to connect operating systems and applications through middleware, a layer of software that standardizes the interfaces between them and acts as a translator to ensure that applications can communicate effectively with various hardware systems without needing to be tailored to each specific component.

With middleware, a developer can write a software application once and know that it can run on various hardware and operating systems without needing to be rewritten to fit each one. Middleware takes care of the details, letting OEMs focus on developing the business value in software-defined vehicle (SDV) applications, from personalized infotainment offerings to critical updates.

"What is needed is a unified, cross-domain and cloud-native software platform to enable consistent communication, dynamic updates, remote diagnostics and efficient lifecycle management — a platform that unites fragmented systems to achieve better scalability and decrease development complexity and cost."

LIMITATIONS OF MOST MIDDLEWARE

But middleware is just one piece of a much larger puzzle. And despite the many advantages middleware offers — such as abstraction, interoperability and scalability — today's market dynamics have exposed significant limitations. Originally designed with high expectations and adopted widely across industries, many traditional middleware solutions have become increasingly complex and rigid, frustrating developers and slowing innovation.

A common pain point is API fragmentation. Application programming interfaces are the point of contact between applications and the functionality that middleware provides. As the functionality evolves, it can be difficult to maintain compatibility between APIs and applications. As a result, many platforms fail to address the real-world challenges of modern software architectures and lifecycle management, such as compatibility among disparate offerings, hampering optimal performance and restricting the ability to perform timely upgrades. Done right, middleware should not merely comply with specifications — it should actively remove barriers to efficient development, deployment and evolution.

While interoperability is the core promise of middleware, most current offerings fall short by locking developers into proprietary ecosystems. From the middleware itself to associated tools

and training, organizations often find themselves dependent on a single vendor. This vendor lock-in leads to inflated costs, limited flexibility and constrained innovation. Customizations can incur exorbitant fees, so development teams are forced to adapt their workflows to rigid vendor-prescribed models — often requiring steep learning curves and productivity trade-offs.

These challenges are not confined to a single domain within automotive. Whether it is ADAS, infotainment, powertrain control or body electronics, the need for open, modular and developer-friendly middleware is increasingly critical. Each of these domains faces unique integration and lifecycle hurdles, yet they all benefit from platforms that enable seamless interoperability, reduce vendor lock-in and accelerate development. There is a clear opportunity for middleware solutions that go beyond compliance — empowering OEMs and suppliers to innovate across domains without being constrained by proprietary ecosystems or rigid workflows.

KEY ELEMENTS OF SOFTWARE INFRASTRUCTURE PLATFORMS

Even with middleware that meets these requirements, more is needed to fully support SDVs. A software infrastructure platform must have several key elements. The well-orchestrated interplay among them is essential to achieving a safe, flexible and cost-effective development environment for SDVs. Among the top requirements are:

Over-the-air updates. OTA is a key enabler of the software-defined vehicle, allowing for

CLOUD-NATIVE SDV ARCHITECTURE

Speeds development, streamlines deployment and optimizes lifecycle

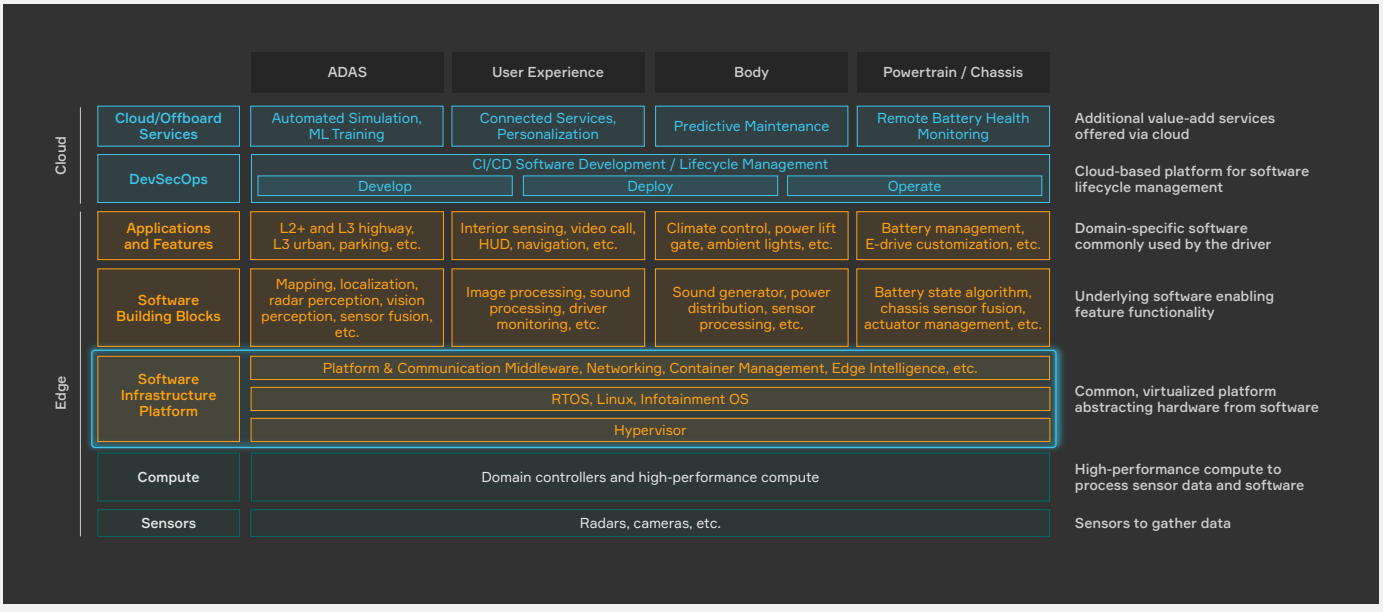

Fig. 1. A cloud-native software infrastructure platform for SDVs enables faster development, better security and lower costs.

real-time updates and design flexibility. Secure and reliable OTA updates allow OEMs to deploy new features, bug fixes and security patches to vehicles already on the road, keeping fleets current and competitive. OTA updates enable consumers to avoid dealership visits, increasing customer satisfaction. For OEMs, OTA updates save money by not requiring manual labor from a service technician. OTA warranty fixes can achieve near-complete coverage in a matter of days. When paired with vehicle diagnostic tools and processes, automotive OTA updates can reduce warranty costs by up to 50 percent for addressable software-related items.

DevOps. A software infrastructure platform must support industry standards and the full software lifecycle, from development and testing to deployment and operations, allowing developers more flexibility and customization so they can innovate faster, reduce costs and maintain software quality across vehicle generations. Most importantly, the platform must respect developers' own workflows and not impose external frameworks that can cause delays and unnecessarily complicate internal processes.

Containerization. Cloud environments depend on containers to isolate applications and manage their deployment. With the help of an in-vehicle container orchestrator, applications can be deployed, updated and tested independently of one another, making the software development lifecycle far more agile. For example, developers can create test environments for new features, simulate those features' behavior in real time and deploy updates over the air without disrupting other vehicle subsystems. This container-based approach ensures faster integration and testing because it corrals potential issues and reduces the complexity of testing different systems together.

Microservices and security. Secure containerized environments foster trust among OEMs, Tier 1 suppliers, and software partners, enabling safe codevelopment and integration across the SDV ecosystem. Containerized microservices ensure secure and resilient software deployments — protecting vehicle systems and user data while complying with automotive cybersecurity regulations, such as UNECE WP.29 and ISO/SAE 21434. In a properly configured software infrastructure platform, each containerized microservice is isolated, reducing the effects of attempted attacks. Security policies can be applied at the service level, enabling granular control and minimizing systemic vulnerabilities. This ensures that only verified, trusted containers are deployed in production environments.

Network management. SDVs need to manage a growing array of interconnected domains. A software infrastructure platform should allow for the seamless management of network configurations, enabling the proper handling of security protocols as well as real-time communication between multiple modules, subnetworks and applications, thus reducing latency to enhance critical safety and functionality in the vehicle.

Edge intelligence. A well-developed software infrastructure platform should allow for seamless and customizable data collection, enabling OEMs to easily collect and manage various types of vehicle data — including controller area network (CAN), Ethernet, media, diagnostic, network and ECU statistics — without requiring separate, dedicated hardware or customized agents running on each ECU. Real-time data collection provides OEMs with insights into system performance and driver behavior, supporting continuous improvement in vehicle quality, safety, customer experience, cost optimization and fleet management. By understanding how systems and drivers interact, OEMs can refine features and personalize experiences — making the driving experience safer, smoother and more intuitive.

Determinism. It is essential that a software infrastructure platform perform in a deterministic manner — even across individual components — because ISO 26262 compliance requires deterministic execution of safety-critical tasks. In an SDV, operations such as braking and lane-keeping must be performed within strict timing constraints and without unpredictable delays. Deterministic behavior ensures that sensor inputs, actuator commands and control loops are processed in real time, which is critical

for maintaining vehicle responsiveness and operational safety. Deterministic systems are easier to test, validate and debug. Engineers can reliably reproduce issues, which is essential for regression testing, certification and long-term maintainability.

The abstraction of middleware. When part of a larger software infrastructure platform, middleware provides more value than when used in traditional approaches. It holds the other elements together, allowing them to be drawn upon as necessary. Middleware abstracts the underlying hardware and communication protocols, allowing software components to be developed, tested and updated independently of specific hardware configurations. This abstraction enables seamless portability across different systems-on-a-chip (SoCs) and device classes and facilitates cross-domain data exchange, ensuring interoperability across heterogenous systems and ECUs. There are three different levels of abstraction that middleware must provide:

- **Communications.** Middleware abstracts communication protocols by elegantly hiding the complexity of underlying transport layers (such as CAN, Ethernet and SOME/IP) while allowing developers to focus on application logic instead of low-level communication details. This facilitation of cross-domain data exchange ensures interoperability across heterogenous systems and ECUs.
- **Operating systems.** Modern vehicles can contain more than 100 ECUs, each running its own OS. Middleware allows developers to smoothly work within this complexity by standardizing communication protocols between ECUs, managing cross-domain functionality and providing data pipelines for edge AI applications, ensuring consistent access to sensor and actuator data regardless of the OS.
- **Hardware.** The right middleware allows for greater flexibility in hardware design, permitting an SDV to be hardware-agnostic. Developers can select whatever SoCs are appropriate, and middleware will help applications run on them. In addition, middleware allows projects to be ported without excessive updating. Applications developed for one type of vehicle can be reused in other types, without developers having to rewrite the code.

A software infrastructure platform helps designers move from building software for isolated ECUs to integrating them as a unified system. It helps with vehicle lifecycle management, such as overseeing how data is being obtained from vehicles. In turn, that data informs decisions about which components and functions need to be updated and when, in a digital feedback loop.

THE FUTURE

OEMs that are serious about building and maintaining a competitive advantage must go beyond exploration to natively build software-defined vehicles. In this market reality, OEMs should seek a vendor that can provide a comprehensive software infrastructure platform that incorporates advanced middleware techniques to enable efficient lifecycle management for generations of software-first vehicles. This approach will reduce costs and production time while increasing opportunities for customization and domain-specific architectures both at the edge and in the cloud.

A software infrastructure platform can also extend beyond automotive to power intelligent robotics systems, from autonomous drones to mobile robots. As AI-based autonomy scales across industries, OEMs will require a modular, secure and real-time infrastructure to build, deploy and evolve intelligent machines at the edge.

In an increasingly AI-driven industry, where vehicles are intelligent edge devices that interact with the cloud, and OEMs must reckon with the complexities of physical AI across multiple domains, OEMs will need a trusted, experienced partner that can provide an end-to-end software solution with support services to ensure a smooth workflow, solid architecture and reliable performance for their customers.

ABOUT THE AUTHORS

Craig Turner
Vice President and Managing Director, Digital Cockpit and Middleware

Craig Turner leads Aptiv's infotainment, interior sensing, and software infrastructure platform product lines, creating next-generation in-vehicle experiences. Since joining Aptiv in 2020, he has driven customer engineering for user experience products, delivered engineering services, and established Aptiv's DevSecOps practice. With more than 23 years of experience in software and embedded systems across automotive, mobile, robotics and gaming, Craig is a passionate advocate for software-driven transformation.

Niheer Patel
Head of Product Management, Middleware & DevOps

Niheer Patel oversees Aptiv's software infrastructure platform, driving software-defined transformation through scalable middleware and DevOps. With 20 years of experience in embedded software, safety and security, he has led the development and certification of mission-critical software infrastructure for the automotive, aviation and industrial markets, shaping the future of intelligent mobility through innovative products.

LEARN MORE AT APTIV.COM/MIDDLEWARE →