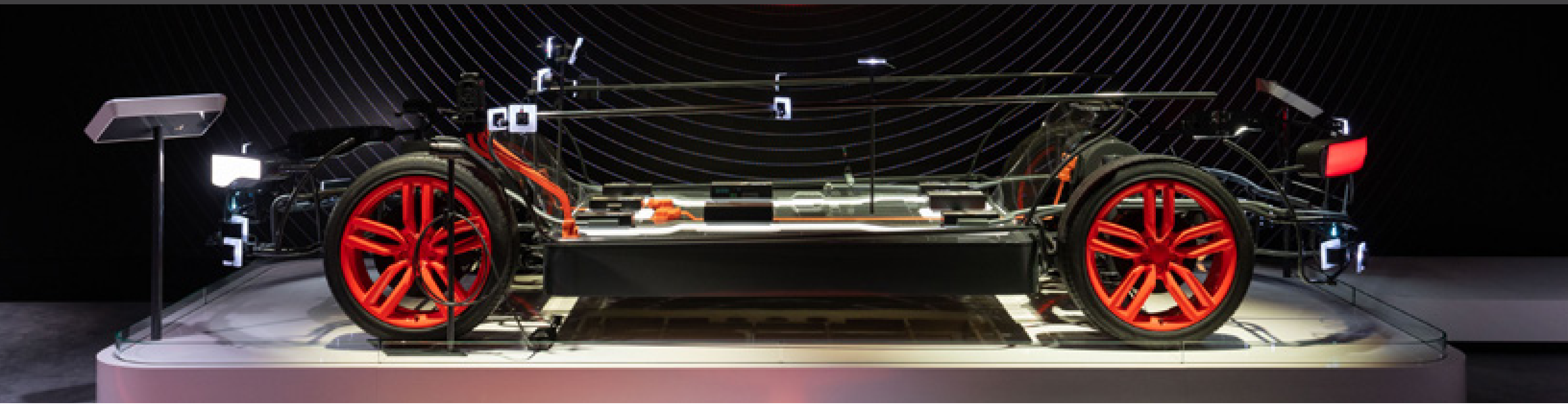
APTIV

WHITE PAPER



Smart Vehicle Architecture™

A SUSTAINABLE APPROACH TO BUILDING THE NEXT GENERATION OF VEHICLES

The automotive industry has introduced an unprecedented array of electrical and electronic innovations in recent decades — from passive safety features such as airbags to immersive user-experience and infotainment, as well as active safety features such as automatic emergency braking.

Each new innovation requires its own electronic control unit (ECU) with its own power, its own processing, its own data and its own connectivity. Each feature's hardware brings its own wiring, introduces complexity, takes up space and adds weight to the vehicle.

This approach barely meets the needs of today's feature-rich vehicle and certainly won't scale as the industry moves toward fully autonomous driving, the most complex challenge it has ever contemplated. What's needed is a new vehicle architecture that simplifies the design, centralizes computing power and optimizes electrical/electronic content, components and functionality.

Smart Vehicle Architecture (SVA™)

This is an exciting and dynamic time for the automotive industry. Advancements in software, compute and sensors are enabling a wide array of innovations in advanced safety systems on the path to fully autonomous driving. Consumers are demanding new features for safety, comfort and convenience with increasing frequency. Consumer preferences, tightening regulations and improving battery costs are moving the industry toward electric vehicles. And 5G and other wireless technologies are creating opportunities to deliver vehicles that are even more connected than they are today.

The challenge, of course, is that all these trends are happening at once. Space within a vehicle's chassis is finite, as is the customer's wallet. It's not sustainable to continue the traditional approach of adding a new ECU for each new feature — as each requires its own power, its own processing, and its own data and connectivity. It won't scale, and it's too complex.

OEMs realize this. They see that the incremental, monolithic approach to delivering customer features and functions is driving unmanageable complexity in all phases of the vehicle life cycle. In the development phase, where speed-to-market is critical to competitiveness, complexity increases development times. And a monolithic approach to development, where software and hardware are inextricably linked, limits reuse and makes any engineering changes difficult. In the manufacturing and assembly phase, complexity leads to components that are difficult to assemble manually and don't lend themselves to automation. And in the postproduction phase, complexity reduces the ability to update features throughout the vehicle's life.

What's needed is a simpler approach: a new vehicle architecture for electrical and electronic systems designed from the ground up for today's feature-rich vehicles as well as tomorrow's highly automated vehicles.

THE PHILOSOPHY

To address these challenges and prepare for the future, Aptiv developed Smart Vehicle Architecture™. SVA™ embodies a vehicle-level design philosophy with three primary goals. The architecture must:

- **Reduce complexity.** By simplifying the hardware and software topology within the vehicle, SVA reduces interdependencies between the many different ECUs currently required to enable various functions.
- **Unite diverse applications.** SVA brings together software from many different domains across the vehicle to unlock new functionality and improve life cycle management.
- **Empower OEMs**. SVA gives OEMs the ability to fully control the software that defines the user experience of their vehicles and to enhance that functionality over time.

SVA achieves these goals through three fundamental principles that differentiate the approach from today's architectures.

First, SVA abstracts the software from the hardware. While such separation is already common on most of today's IT platforms, this concept is now gaining momentum in the automotive industry. Separating software from hardware allows continuous release cycles for the software. Just as apps on today's smartphones regularly receive incremental updates and improvements, the software in a vehicle should be able to update more frequently than the hardware it runs on. This separation also allows developers to reuse software more easily as they move it to different platforms, rather than port it.

Second, SVA separates input/output (I/O) from compute. That is, the architecture takes all of the physical connections to peripheral sensors and devices and places this functionality into zone controllers that are separate from the computers in the domain controllers. An analogy is the docking station for a laptop computer. All of the peripherals — keyboard, mouse, printer and so forth — plug into the docking station, allowing the laptop to be swapped in and out easily. In a vehicle with SVA, the zone controller delivers power and data connections to the sensors and other devices, with just a backbone connection to the domain controllers. This approach improves scalability and reduces physical complexity.

Third, SVA “serverizes” compute. Once the I/O is separated from the compute, the approach can allocate the computing resources in a vehicle among various software applications dynamically, as needed, much like a cloud computing model. A vehicle with SVA can allocate the necessary compute power, RAM, graphics processing, and so on, to applications based on priority and need. Serverization can even allow sharing of resources among physically separate domain controllers, so they can operate logically as one. Additionally, this approach supports mixed criticality; that is, a critical safety feature that requires more processing power, for example, has priority over less critical functions such as infotainment.

THE PHYSICAL COMPONENTS

Servers provide value at three service layers — infrastructure, platform and software (see Figure 1) — and these layers affect how SVA physically manifests itself in a vehicle. The physical layout of the architecture brings with it additional benefits, such as design for automated assembly, support for redundant power and data, and electrification.

Foundational components include:

- **High-voltage busbars.** These sit directly on the battery and deliver power throughout an electric vehicle. Their flat profile and semi-rigid nature make it easier to package them into a vehicle.
- **Dock & Lock™ system.** This dock attaches to the floorboard of a vehicle and provides a base to which a robot can attach all of the other central elements of the architecture.
- **Unified power and high-speed backbone.** This backbone carries power to every component of the architecture. It also aggregates all the data communications within the vehicle. It can even support redundancy easily and efficiently when needed through a dual-ring topology.

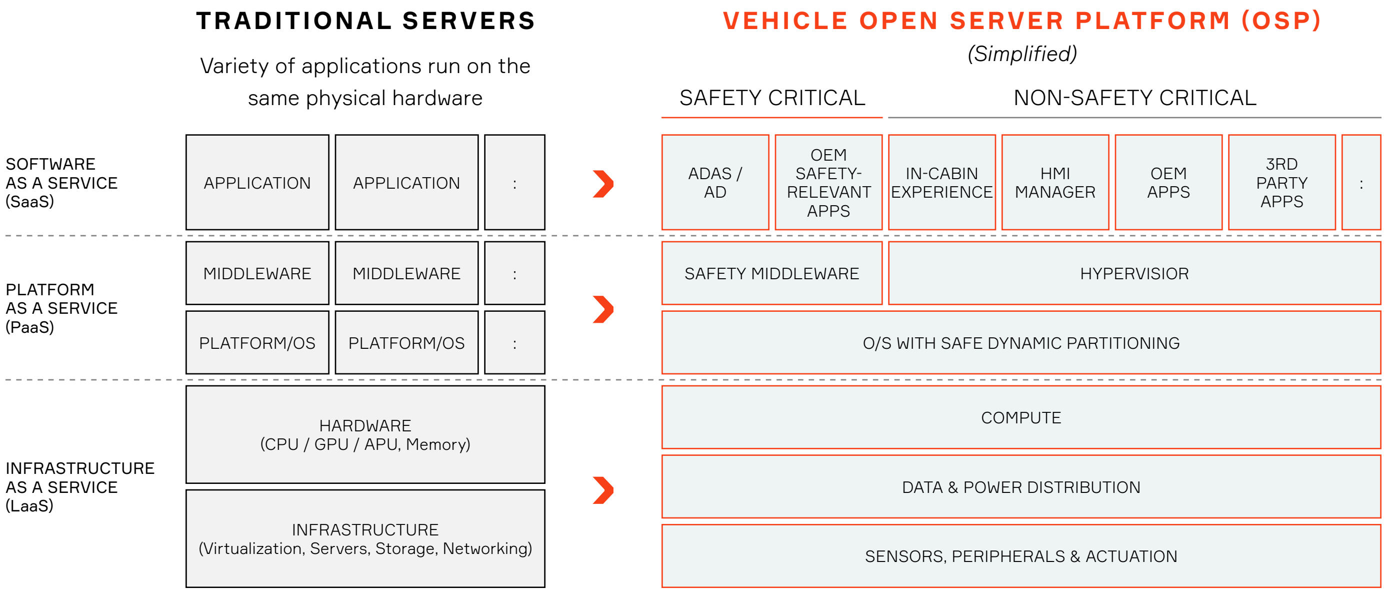


Figure 1. How a server model translates to SVA’s Open Server Platform.

Connected to the backbone are the elements resident in the central compute cluster, which include:

- **Secure Connected Gateway (SCG).** This is the SVA master controller and body master. It controls critical functions related to waking up the system and the flow of data into and out of the vehicle through wireless connections.
- **Open Server Platform.** These domain controllers run the software required for enabling advanced safety features and the in-cabin user experience. They also have the ability to dynamically share compute resources, providing improved performance and cost-efficient redundancy.
- **Power Data Centers.** These are the zone controllers, the “docking stations” that connect all of the sensors and peripherals. Depending on the configuration of the vehicle, there could be two to six PDCs, with different variants helping to scale performance appropriately.
- **Propulsion and Chassis Controller.** This controller provides the mission-critical engine management (for an internal combustion engine) or battery management system (for a battery electric vehicle), as well as all chassis functionality such as steering and braking.

HOW SVA REDUCES COSTS

This approach to the software architecture and physical structure of the vehicle represents the logical future state as OEMs continue to scale up the features and intelligence of their vehicles. But consumers also have to be able to pay for it. The good news is that by reducing the complexity, SVA effectively lowers the total cost of ownership throughout all phases of the vehicle life cycle: development, manufacturing and postproduction.

Development

Current development approaches are very linear in the automotive industry. After the concept phase, developers have to wait for the target hardware to understand how software will function on that system. Then, after the software has been fully coded, it has to be tested and validated, which can take a long time.

SVA allows developers to create software completely independently of the underlying hardware. It defines hardware performance classes, enabling integrators to combine diverse software applications and then certify their performance for a chosen hardware class. Developers don’t have to know which exact device the software will run on — they just have to define what hardware performance levels are required for the software to run optimally. As long as the hardware meets the specifications of the hardware class, the software will be able to use it.

PARALLEL DEVELOPMENTS ACCELERATE TIME TO MARKET

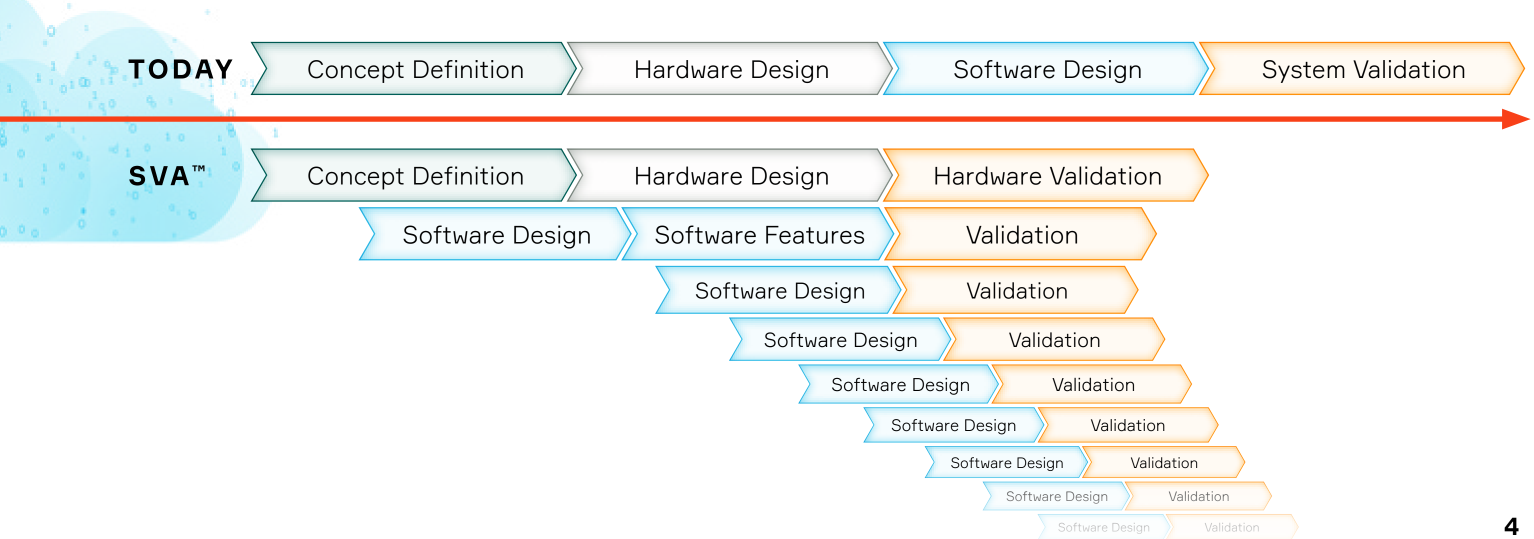


Figure 2. SVA allows developers to create software in parallel.

Taking cues from the iterative development techniques used in agile methods, SVA allows developers to upload updates dynamically. Testing and validation become easier to manage, and developers can go to market faster with richer features.

Taken together, Aptiv estimates these techniques cut system integration and testing costs by 75 percent and lower warranty costs by 75 percent as well.

Manufacturing

SVA also cuts costs in the manufacturing phase in two important ways.

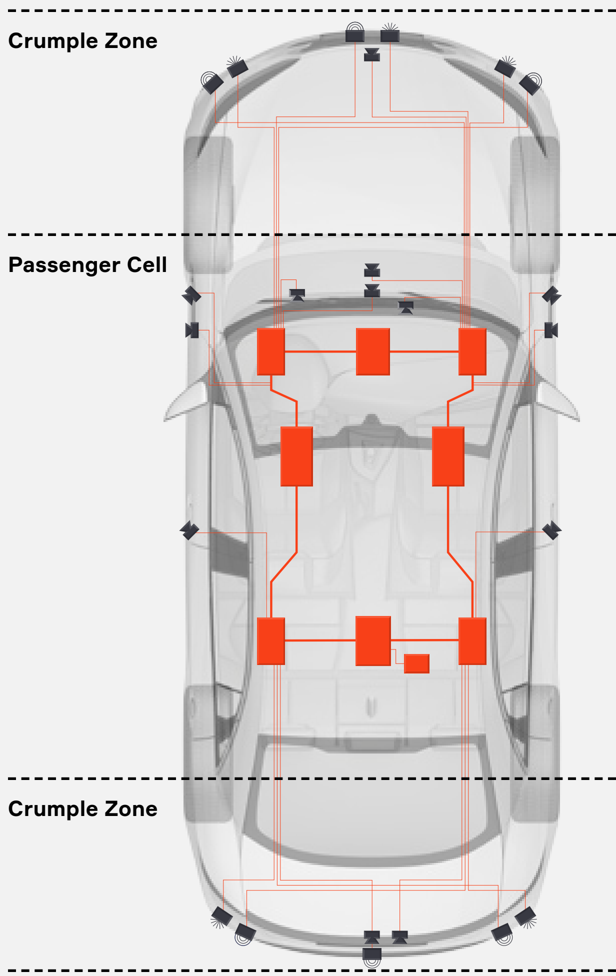
The first is through up-integration. Today, functions are distributed across multiple ECUs located throughout a vehicle. When those are consolidated into a smaller set of domain controllers, the vehicle is able to shed multiple microcontrollers, multiple power supplies, multiple housings, and copper wiring — all while maintaining or even increasing compute capabilities. This results in a 20 percent reduction in weight for the wiring harnesses and a 20 percent reduction in weight and packaging space for compute.

The second way it cuts costs is through direct labor reduction. For example, because the PDCs simplify the physical complexity and connect directly to the sensors, wiring harnesses can be limited to 2.5 meters or less, which means our customers need just one or two people to install them. Compare that to the 10 or more people it takes to install today's most complex architectures, and OEMs could save 50 percent in labor costs.

Further, because SVA takes advantage of a rigid backbone and zone harnesses, as well as the Dock & Lock connection system, SVA can achieve the highest levels of automation, further reducing labor costs and meeting the increasingly strict quality thresholds these advanced features and functions demand.

ASSET PROTECTION

In SVA, the sensor intelligence and processing power are centrally located in the passenger compartment of the vehicle, rather than distributed in the sensors themselves, as is common today. This helps reduce the total system cost, as well as the cost of sensor components, which in turn reduces costs associated with minor accidents involving those sensors — and therefore the cost of insurance.



Post-production

SVA continues to reduce costs even after a vehicle has left the factory. With software abstracted from hardware, vehicle manufacturers could build a library of certified software — an “app store” for the car, in effect. These apps could potentially include software developed by the OEM, by Aptiv, or even by a third party. Over time, that library could expand into new functions or allow for updates to existing applications.

Vehicle manufactures and software developers can upgrade software in vehicles throughout its life cycle using over-the-air (OTA) updates. The vehicles receive those updates via the Aptiv SCG, which in turn updates the other systems in the vehicle after the new code has been validated, and then at an appropriate time when the vehicle can be safely updated. This lowers warranty costs by fixing issues without having to go to a dealership, and results in greater brand loyalty and customer satisfaction.

For OEMs, this capability creates the potential for software reuse, enabling an infinite number of vehicle-specific software builds and virtually eliminating software maintenance costs tied to model year updates.

FIRST STEPS

SVA is a holistic vehicle-level architecture approach, but it allows OEMs to take incremental steps to get there. The first step or “building block” is typically to implement domain controllers to up-integrate and expand some of the compute that is currently distributed throughout vehicles, particularly for domains such as advanced safety or infotainment. The next major step is to use zone controllers to break apart the physical complexity into more manageable zones, while further driving up-integration of distributed ECUs. From there, an OEM can then move toward a server architecture with abstraction and dynamic allocation of compute.

With those pieces in place, an OEM has the building blocks of SVA, and will be able to take advantage of its ability to enable advanced features and high degrees of automation through a software-defined architecture that is sustainable well into the future.

DOMAIN AND ZONE CONTROL REPRESENT BUILDING BLOCKS TOWARDS SVA™

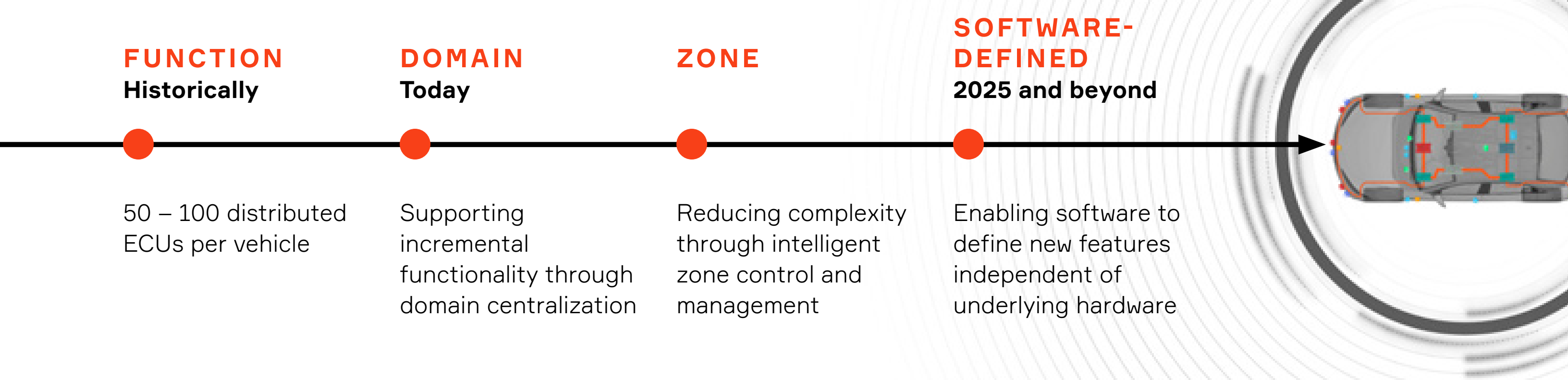


Figure 3. Steps toward the full realization of SVA.