Insight report

Software Defined Vehicle Value Chains

ADVANCED PROPULSION CENTRE UK

Contents

Department for Business & Trade

The Advanced Propulsion Centre UK (APC) operational costs are funded by the UK Government's Department for Business and Trade (DBT) and industry contributions. DBT is the ministerial department for economic growth, supporting businesses to invest, grow, and export, creating jobs and opportunities across the country. Our combined mission is to accelerate the development of advanced propulsion technologies to reduce greenhouse gas emissions, minimise embedded carbon, and improve air quality.

Introduction:
What is a Software-Defined Vehicle?

The automotive sector is moving from vehicles defined primarily by hardware to vehicles increasingly shaped by software, compute, data, and connectivity. This changes how vehicles are designed, validated, updated, and supported, and is reshaping where value is created across the automotive industry.

The definition of a Software-Defined Vehicle (SDV) is still evolving across the global automotive industry. In practice, the term is used to describe a vehicle where software plays a central role in controlling functions, enabling updates, personalising user experiences and supporting continuous improvement after launch. Key characteristics of an SDV include:

- Functions delivered, controlled, and updated through software
- Features updated beyond Start of Production (SOP)
- Software issues resolved remotely after vehicle release
- Hardware and software becoming increasingly decoupled

The SDV transition is not simply a technology preference, but instead a response to a structural change in what modern vehicles must do and how quickly they must evolve. Vehicles are becoming more connected and compute-intensive, with advanced functionality (for example in driver assistance and energy optimisation) relying on increasingly complex software. At the same time, customers have been accustomed to products that improve through updates, making continuous evolution an expectation rather than a novelty.

A key implication is that SDV is not limited to infotainment, but extends to vehicle-critical domains, where software influences safety and performance outcomes, and where updates must be validated and governed accordingly. In the SDV model, the vehicle increasingly behaves like a "living product," in which there is always a subsequent version or iterations in development. The operating model requires persistent software teams and supporting toolchains, leading to new business models (including subscriptions), continuous improvement loops from user feedback and telemetry, and a pathway to improve safety and performance over time.

This report sets out how the SDV transition is reshaping the automotive value chain, where value is likely to move, and which capabilities become strategically important. It focuses on the hardware, software, cloud, data, and integration layers that underpin SDV development, before considering where the UK may have an opportunity to build competitive advantage.

Why the automotive industry is changing

The automotive industry is experiencing a shift in how value is created, captured, and sustained, with differentiation moving beyond purely mechanical systems towards compute, software, data, and cloud-enabled services.

This reflects both customer pull, with a demand for seamless digital experiences, personalisation and continuous improvement, and industry push, as rising system complexity challenges traditional architectures and development models.

In practice, this changes how vehicles are engineered, launched, and supported. More connected functions increase system interdependency, network traffic, and integration burden. This makes architecture-first design, earlier verification and validation, software reuse, and automated delivery practices more important. For OEMs and suppliers, SDV therefore changes not only the vehicle architecture, but also the skills, tools, partnerships, and operating models needed to compete.

SDV value chain overview

The SDV value chain spans hardware, software (on-board and off-board), integration, and the lifecycle operations required to support continuous improvement beyond SOP. SDV shifts the product model from a one-off release, with refresh cycles, toward a platform which is continuously updated, monitored and refined.

Hardware

Vehicle hardware and electrical architectures do not dictate whether a vehicle is an SDV, however it is a critical enabler. Electrical/electronic (E/E) architecture choices influence how effectively software can be deployed and updated at scale. Central compute, zonal controllers, and telematics connectivity provide foundations for scalable software platforms and over-the-air (OTA) operation.

Within the automotive industry, there is a general shift from the incumbent distributed architectures towards domain-based and then zonal architectures moving forwards. This transition is not simply a hardware change, but a response to the increasing software complexity, network traffic and integration burden created by connected and software-rich vehicles.

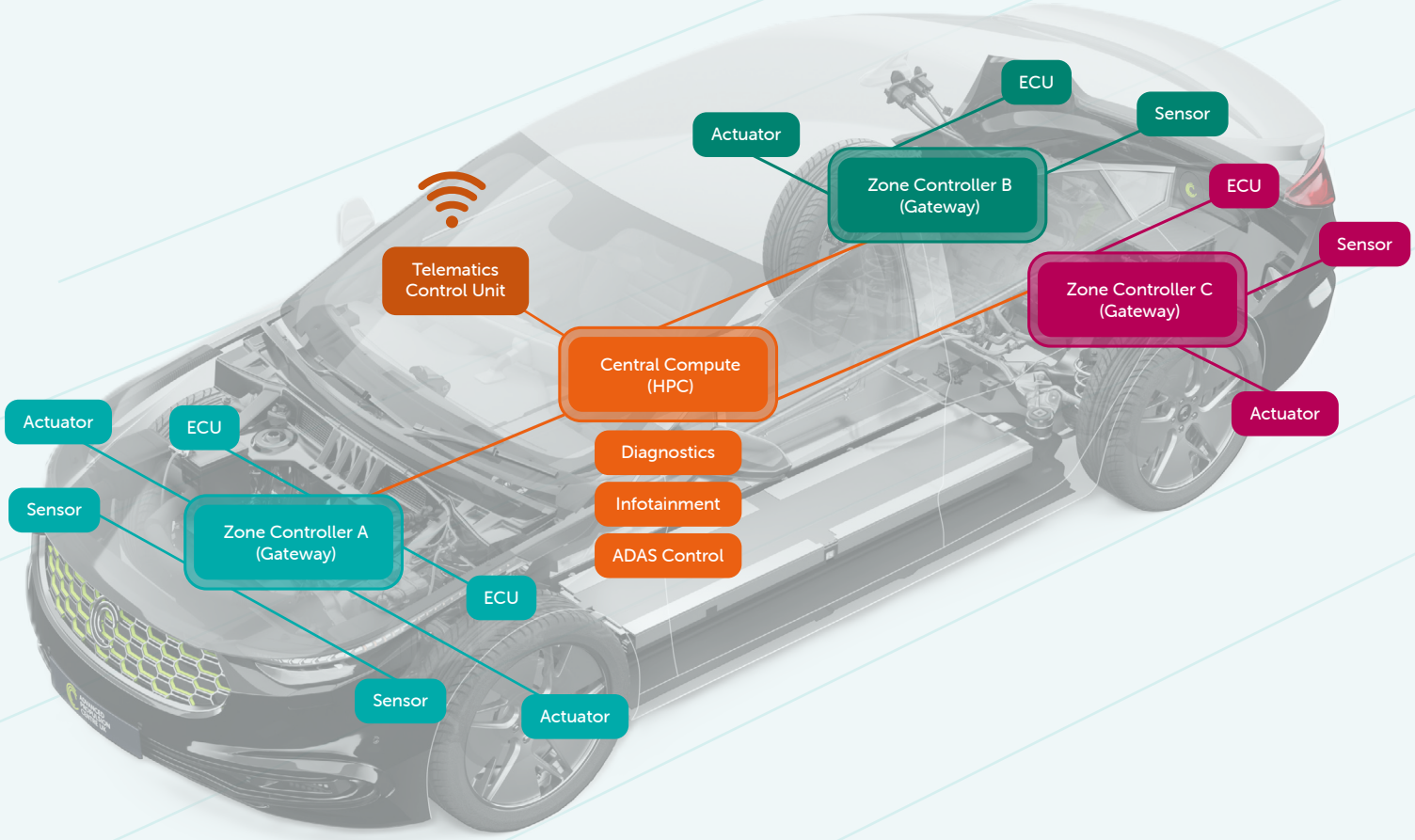
Zonal

A zonal architecture is organised by physical zones of the vehicle rather than functional domains, with the majority of processing power in the central vehicle computer. Benefits include reduced wiring complexity, lower weight, fewer but more capable compute nodes, and a more scalable basis for software deployment across vehicle models.

Incumbent OEMs face a fundamentally different challenge compared to new entrants, as established manufacturers continue to work with legacy E/E architectures that have proven reliability, established supplier interfaces and known validation pathways. They are therefore layering zonal concepts on top of existing systems rather than outright replacement. This creates a pragmatic, lower-risk pathway that still meets customer requirements, but also results in an interim hybrid approach, where old and new architectures must co-exist.

Key components in a zonal architecture include central compute, zonal controllers, and a telematics control unit.

E/E Architectures – Zonal Data flow



Central compute

The central compute platform provides centralised high-bandwidth computing, acting as the vehicle's primary processing and orchestration layer. Whereas zonal controllers are distributed across different physical areas of the vehicle and act as local hubs, the central compute platform is responsible for coordinating higher-level functions and allocating compute resources across the vehicle.

Central compute typically consists of a small number of high-performance computers located centrally in the vehicle. These platforms can host resource-intensive applications, run operating system layers, support cybersecurity and safety orchestration, manage system-level coordination, and provide the foundation for software reuse across vehicle platforms. As vehicles transition towards software-defined architectures, the strategic importance of these compute resources is increasing significantly. In response, OEMs are taking greater ownership of their design, in some cases extending to the procurement of custom semiconductors.

Zonal controllers

Zonal controllers are smaller, lower-power compute units placed in different physical zones or regions of the vehicle. These controllers gather signals from sensors and actuators in their zone, provide power and communication gateways, and enforce safety and redundancy at the edge.

By aggregating local device inputs and outputs, zonal controllers simplify wiring complexity and reduce the need for long harness runs between individual components and central processing units. In an SDV context, their value is not only in physical simplification, but also in creating a more scalable interface between local vehicle hardware and centralised software control.

Telematics control unit (TCU)

The TCU is the core hardware module which bridges the on-board vehicle environment with external networks. It collects data from vehicle systems, sends and receives data from the cloud, enables OTA updates, and supports vehicle-to-everything communications.

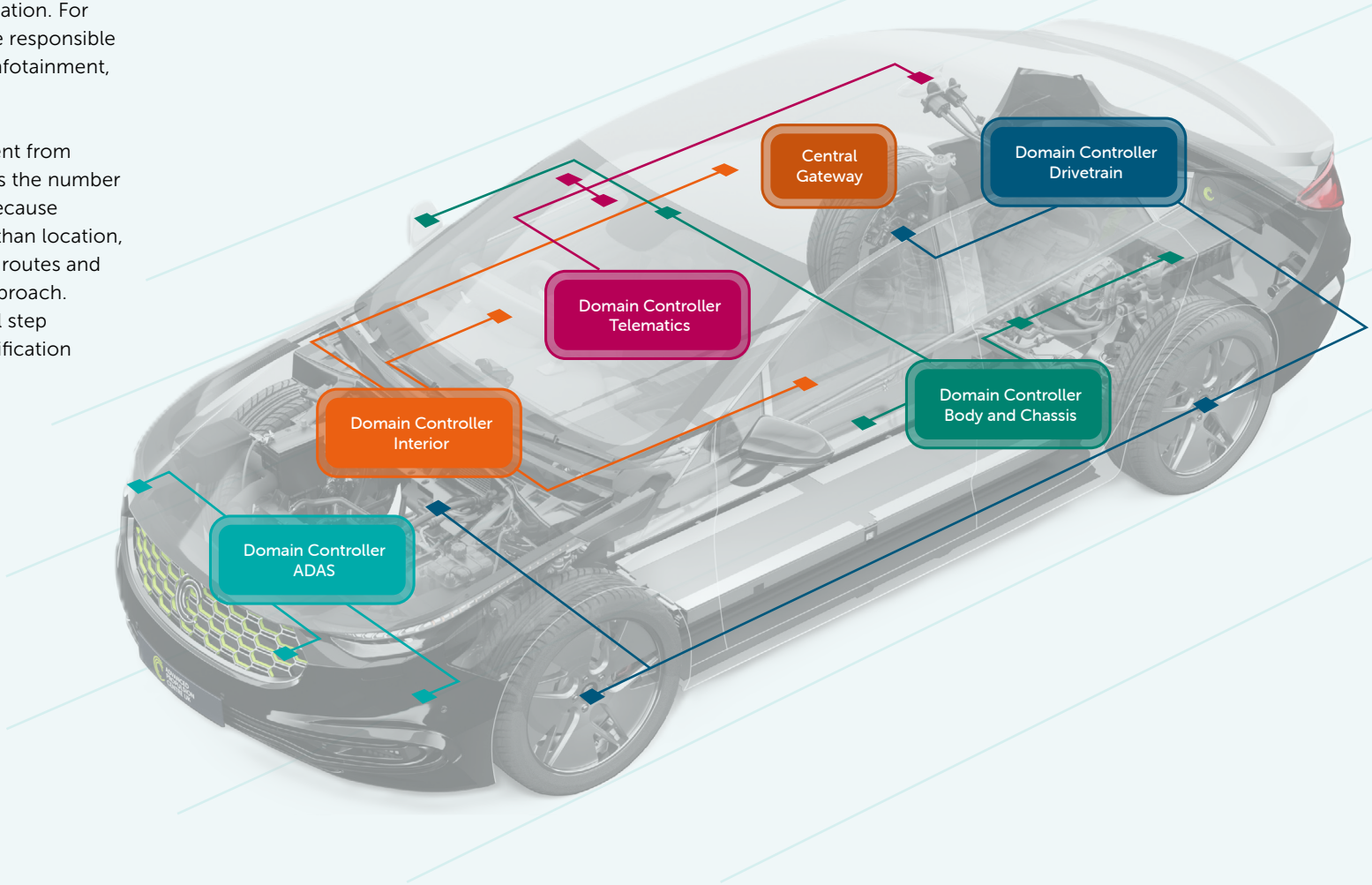
In the SDV value chain, the TCU is therefore a key enabler of this vehicle's connection to cloud services, fleet monitoring, data analytics, and remote software delivery. Without reliable connectivity, the continuous improvement model of SDV is constrained, as the feedback loop between vehicle, cloud and engineering teams becomes slower and less scalable.

Domain

In comparison to zonal control, domain control architecture consolidates functions by system type or functional domain rather than by physical location. For example, separate domain controllers may be responsible for areas such as body, chassis, powertrain, infotainment, or ADAS.

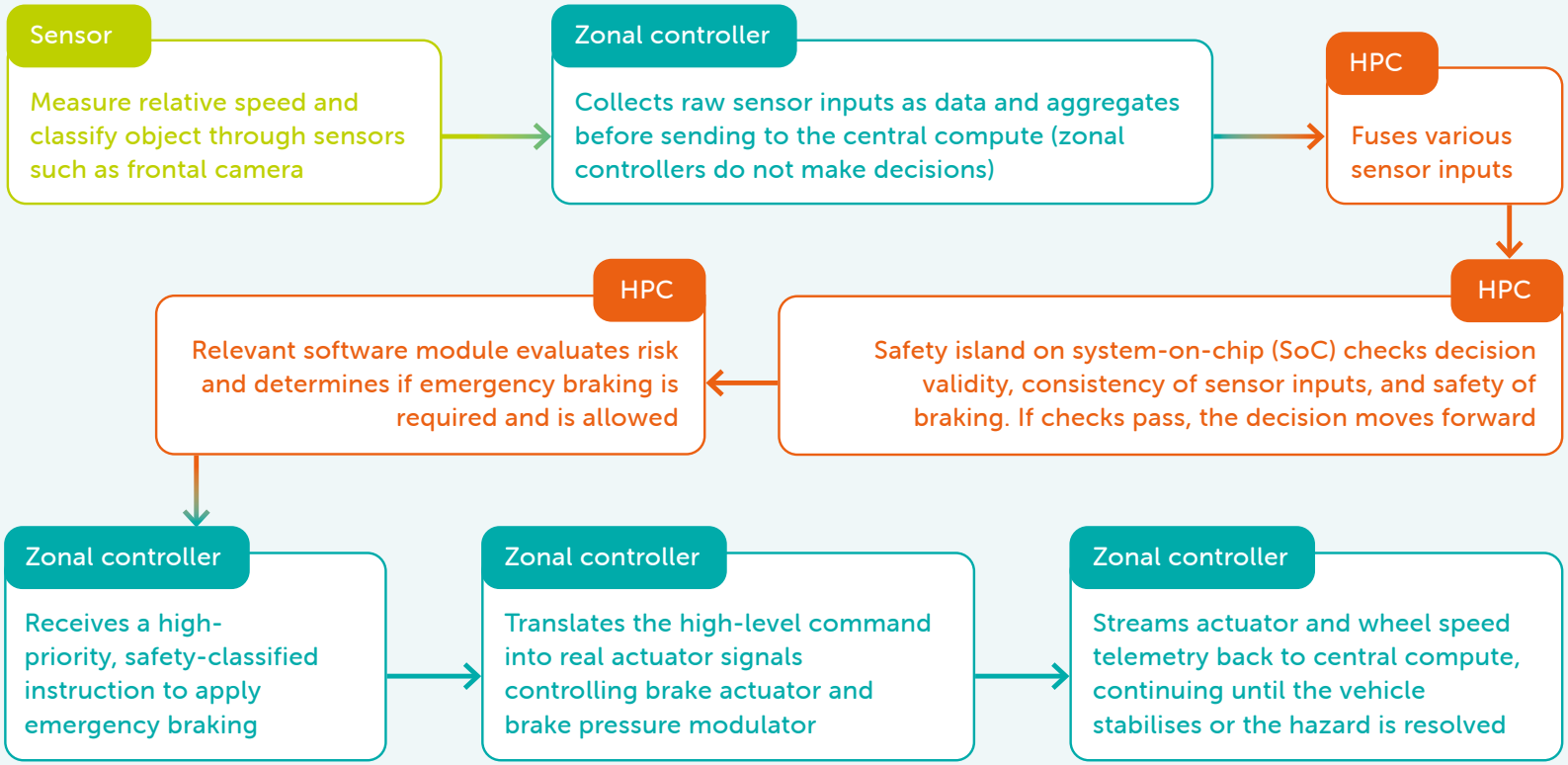
Domain control represents a step improvement from highly distributed architectures, particularly as the number of on-vehicle systems increases. However, because components are grouped by function rather than location, the architecture can still create longer wiring routes and a higher harness burden than a fully zonal approach. It therefore provides an important transitional step but does not capture all of the physical simplification benefits associated with zonal architectures.

E/E Architectures – Domain



Scenario: Adaptive cruise control with automated emergency braking

Vehicle ahead decelerates sharply on a motorway



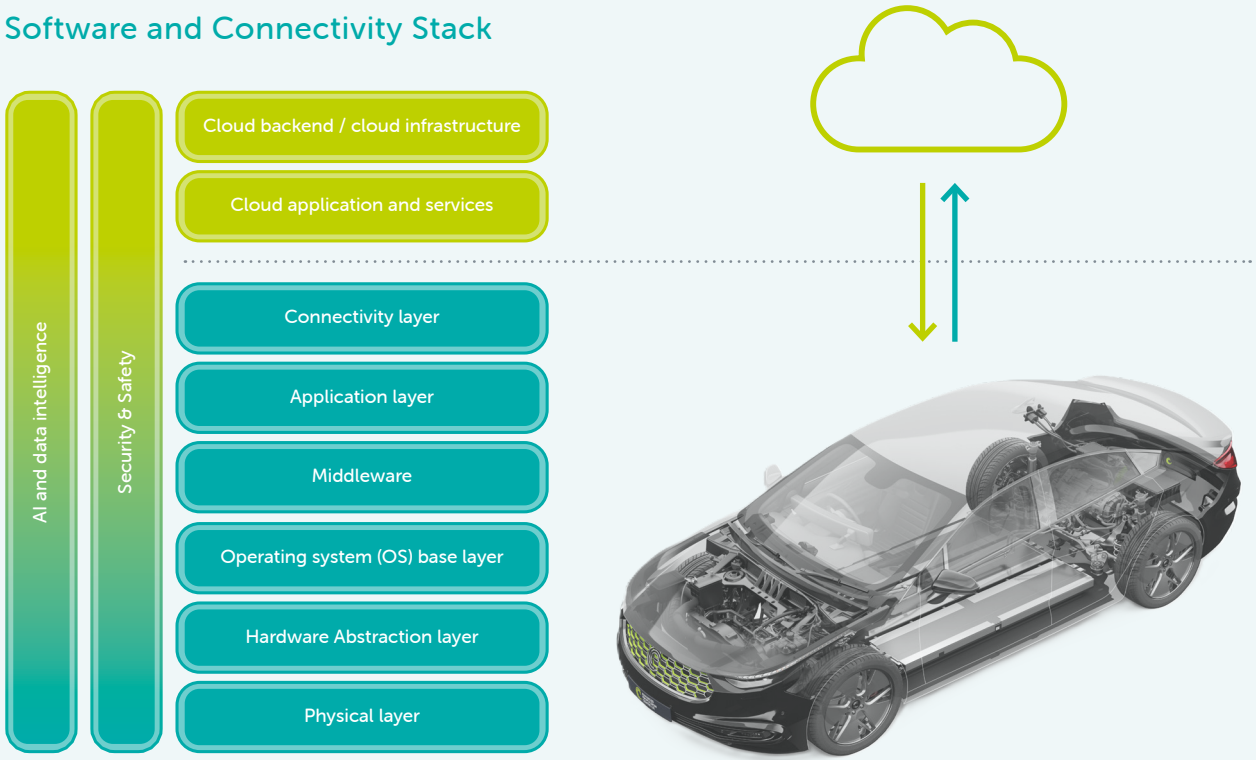
This scenario demonstrates that the central compute is tasked with decision making, however the zonal controllers serve as data aggregators and communicators.

Software and connectivity stack

Software value creation increasingly moves up the stack, as on-board software layers combine with off-board cloud and data platforms. As software persists post-SOP, integration and validation become decisive capabilities. OEMs and suppliers must orchestrate multiple software layers, supplier interfaces, test environments, release processes, and governance models across a live vehicle fleet.

Due to the complexity of SDVs, it is unlikely that a single automotive player will provide a holistic solution across the entire stack. Instead, different players are likely to own or specialise in specific elements, introducing opportunities for non-traditional automotive companies to enter the value chain. This is particularly visible in areas such as middleware, cloud infrastructure, AI, cybersecurity, software tooling, and data services.

The SDV software stack spans both on-board and off-board environments, with security and safety embedded, and AI and data intelligence across every layer. On-board, the stack runs from the physical layer through hardware abstraction, operating systems, middleware, and applications. Off-board, cloud infrastructure and services support data storage, analytics, digital twins, OTA management, and fleet-level intelligence.



On-board

Physical layer
The physical layer consists of the vehicle hardware, including sensors, actuators, high-performance computers, ECUs, communication buses, and network interfaces. Although SDV is not defined by hardware alone, this layer determines the physical constraints within which software must operate, including compute capacity, latency, power consumption, redundancy, and network bandwidth.

Hardware abstraction layer
The hardware abstraction layer (HAL) provides standardised interfaces between the physical hardware and the operating system of software layers above it. This is the key mechanism that enables hardware and software decoupling.

By reducing dependency on specific supplier implementations, the HAL enables software reuse across platforms and can reduce the complexity of updating or replacing underlying hardware. This becomes increasingly important as OEMs seek to scale software functions across multiple vehicle lines without rewriting software for each hardware variant.

Operating system (OS) base layer
The operating system base layer includes real-time operating systems and general-purpose operating systems. Real-time operating systems provide deterministic and safety-relevant execution for functions where timing and reliability are critical, while general-purpose operating systems support more flexible computing environments for less time-critical applications.

This layer provides secure and abstracted access to hardware resources and forms part of the foundation for managing mixed workloads across central compute platforms. As more functions share computing resources, the operating system layer becomes increasingly important for safety, security, scheduling, and resource allocation.

Middleware and enabling functions
Middleware acts as a bridge between applications, operating systems, tools, and databases. It enables communication between software components and supports service-oriented architecture through standardised interfaces and reusable services.

By enabling modularity and reuse, middleware allows functions to be developed, updated and deployed more independently. However, while service-oriented architecture can reduce complexity through standardisation, it also increases the importance of integration discipline, real-time coordination, and system-level validation, particularly where mixed-criticality functions share centralised compute resources.

Middleware can also influence future business models because the standardised interfaces and reusable services make it easier to deploy, update and monetise software-enabled features across multiple vehicle lines. This supports models such as feature-on-demand, subscriptions, fleet services and paid performance or convenience upgrades.

Application layer
Applications consume the services and data exposed by the middleware layer. These include functions such as ADAS, infotainment, diagnostics, energy management, and vehicle personalisation.

In an SDV, the application layer is one of the most visible sources of customer value, because it enables new features, improved user experience and functionality that can evolve after the vehicle has entered production. However, the ability to update applications depends on the maturity of the layers beneath them, particularly middleware, operating systems, abstraction layers, and integration tooling.

Connectivity layer
The connectivity layer manages communication between the vehicle and external systems. This includes data exchange with the cloud, OTA update management, remote diagnostics, and vehicle-to-everything communication.

This layer is critical to the SDV lifecycle because it enables the vehicle to participate in a continuous feedback loop. Data can be collected from the fleet, analysed off-board, and converted into validated updates, services, or engineering improvements that are then deployed back to the vehicle.

Off-board

Cloud backend / infrastructure
The cloud backend receives, stores and processes vehicle data. It includes hyperscaler infrastructure, OEM cloud systems, data lakes, analytics environments, and digital twin infrastructure.

Cloud infrastructure is central to the SDV value chain because it enables fleet-level visibility and continuous improvement. It allows data from vehicles in use to be aggregated and analysed, supporting predictive maintenance, performance optimisation, issue detection, software validation, and future feature development.

Cloud applications and services
Cloud applications and services run off-board and interact with the vehicle through continuous data exchange. Examples include fleet management platforms, predictive maintenance tools, data analytics services, usage-based services, and OTA management systems.

These services shift value from the vehicle as a one-off product towards the vehicle as part of a connected platform. The commercial opportunity increasingly sits in actionable intelligence rather than raw data, with value created through the ability to convert vehicle data into services, updates, and operational improvements.

Cross-cutting pillars

Security and safety
Security and safety are cross-cutting requirements that must be embedded across the entire vehicle and cloud architecture from day one. Unlike traditional vehicles, SDVs expose a significantly larger cybersecurity attack surface due to their reliance on software, connectivity, and continuous updates. Security and safety, therefore, have a role throughout the hardware layer, operating system, middleware, application layer, and cloud environment. Although implementation methods may differ between layers, the underlying principles remain consistent: secure access, controlled change, traceability, validation, resilience, and the ability to manage issues across a live fleet. These principles are increasingly reflected in regulatory requirements, particularly UNECE UN R155 on Cybersecurity Management Systems (CSMS) and UN R156 on Software Update Management Systems (SUMS), which require manufacturers to demonstrate effective cybersecurity risk management and secure, traceable software update processes throughout the vehicle lifecycle.

AI and data intelligence
AI and data intelligence can sit both on-board and off-board. On-board, edge AI enables computation within the vehicle for real-time decision-making, sensor fusion, perception, and driver monitoring. These use cases are particularly important where latency, availability, and safety requirements prevent reliance on external connectivity.

Many OEMs are likely to use a hybrid approach in which AI operates on-board for real-time or safety-critical functions, and off-board for large-scale analytics, model improvement, fleet learning, and service optimisation. This hybrid model reinforces the importance of consistent data pipelines, robust validation, and clear governance of software updates.

Scenario: OTA updates to improve energy management in cold weather climates

Fleet data analysis

Fleet telemetry data, energy usage and charging behaviour is analysed within OEM backend

Fleet data analysis

Analysis shows software logic is overprotecting battery longevity and there is headroom to precondition battery less aggressively

Software stack preparation

A new software package becomes available within the OEM cloud, targeting vehicles OTA that operate in cold climates and with specific battery variant

Software stack preparation

On-vehicle the update is downloaded in the background, whilst update manager checks vehicle state and compatibility

Software stack preparation

System prepares an inactive partition, as to not interfere with active system that is running

Installation

The new software partition is activated overnight, with rollback protections in place for any failures

Post-deployment

OEM monitors relevant parameters such as energy consumption delta and battery temperature

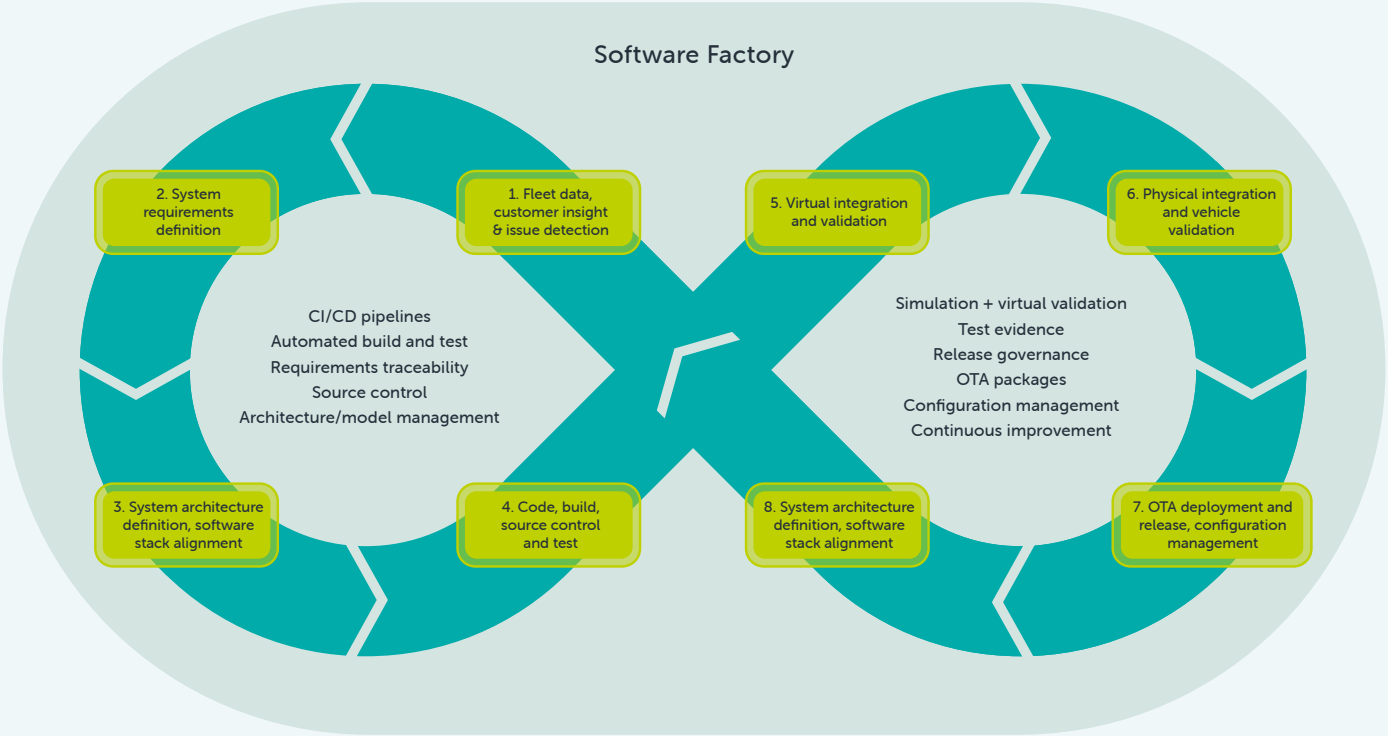
This scenario demonstrates that software-over-the-air (SOTA) is not just about patching or bug fixes, but about controlled interventions into the software stack.
A challenge is that different versions of software can exist across different vehicle variants and architectures.

SDV Integration

In a Software-Defined Vehicle, integration shifts from a late-stage programme activity to a continuous lifecycle capability. Traditional automotive development still requires the rigour of systems engineering, traceability, and physical validation, particularly for safety-critical functions. However, SDVs also require faster, more iterative software delivery before and after start of production. The integration value chain is therefore best described as a continuous loop, where fleet data and customer insights (a data lake) inform requirements, which are then aligned to the software and connectivity stack, software is built and tested, validated virtually and physically, released over-the-air, and then monitored in use to trigger the next cycle.

The central enabler of this loop is the software factory, which is the industrialised capability that enables software to move through development, integration, validation, and release more quickly and repeatedly. It brings together continuous integration/continuous delivery (CI/CD) pipelines, automated build and test processes, simulation, virtual validation, release governance, and OTA packaging. Rather than being a single step in the process, the software factory acts as the digital backbone that exchanges artefacts with every stage of the integration loop, including requirements, code, models, test evidence, OTA packages, and fleet feedback.

SDV integration value chain: continuous software delivery loop



The purpose of this model is to reduce time-to-market while maintaining safety, cybersecurity, and quality assurance. Virtual validation methods such as model-in-the-loop (MiL), software-in-the-loop (SiL), virtual electronic control units (vECUs), and digital twins help move testing earlier in the process, reducing dependence on scarce physical assets and identifying issues before late-stage vehicle integration. Physical testing through hardware-in-the-loop (HiL), system benches and vehicle validation remains essential, but becomes part of a broader continuous assurance model rather than a single end-of-programme gate.

Fleet data, customer insight, and issue detection

1. Fleet data, customer insight, and issue detection

Uses telemetry, diagnostics, customer feedback, and field issues to identify improvement opportunities and trigger the next development cycle.

2. System requirements definition

Defines the functional, safety, cybersecurity, performance, and updateability requirements that the software release must satisfy.

3. System architecture definition, software stack alignment

Aligns the software release to the target E/E architecture, compute environment, middleware, interfaces, and cloud connectivity model.

4. Code, build, source control, and test

Develops, versions, builds and tests the software baseline through controlled repositories, automated pipelines, and repeatable toolchains.

5. Virtual integration and validation

Validates software behaviour in simulated, model-based, and virtual environments before committing scarce physical assets.

6. Physical integration and vehicle validation

Confirms software performance, timing behaviour, safety interactions, and customer-facing functionality on HiL rigs, benches, and vehicles.

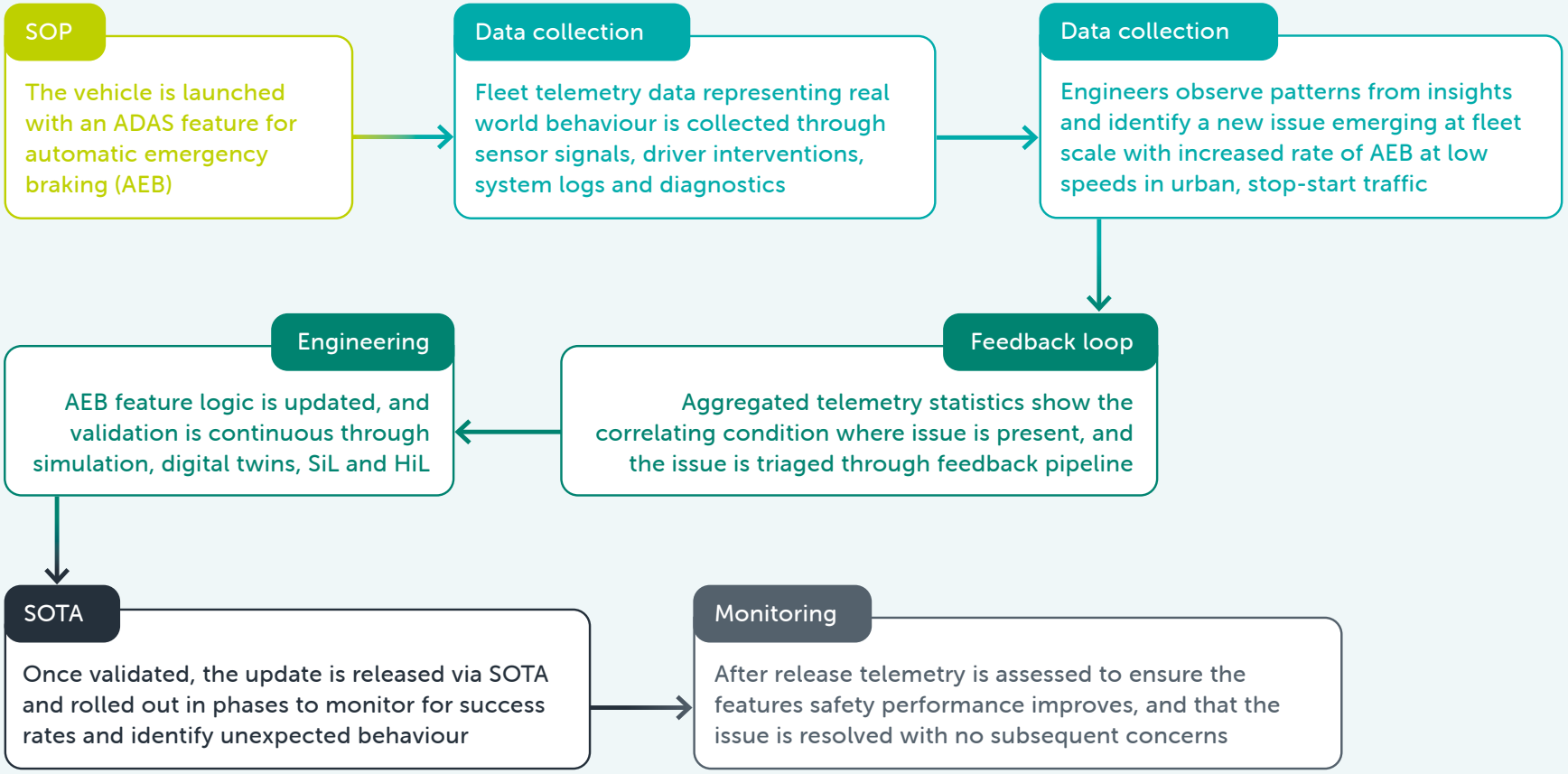
7. OTA deployment and release, configuration management

Packages, approves, and deploys software updates over the air while managing compatibility, staged rollout, rollback, and configuration control.

8. In-use monitoring, feedback, and continuous improvement

Monitors deployed software across the live fleet to detect issues, confirm performance and feed evidence back into future releases.

Scenario: Post-release improvement of an ADAS safety feature



Modern vehicle architectures

Modern SDV architectures are moving from many distributed electronic control units (ECUs) towards consolidation around central compute and more physically organised control structures. A key direction is the shift from domain-based E/E architectures towards zonal architectures supported by central vehicle computers.

The move towards central compute and zonal control supports reduced wiring complexity, cost and weight, software reuse, more efficient integration, improved scalability, and a more consistent basis for lifecycle updates across vehicle platforms.

Distributed architectures were effective when vehicle functions were largely delivered through dedicated ECUs with clearer functional boundaries. However, as vehicles become more connected, software-intensive, and data-driven, the number of interactions between systems increases significantly. This places greater pressure on network capacity, integration processes, validation methods, and the ability to manage software dependencies across the vehicle.

Domain architectures provide an intermediate step by consolidating control within major functional areas. This reduces some of the complexity associated with highly distributed systems and creates clearer domain-level responsibility. However, because the architecture is still organised around function rather than vehicle location, it can retain wiring and integration inefficiencies as the number of connected devices increases.

Zonal architectures organise electronics by physical areas of the vehicle, such as front, rear, or cabin zones. Zonal controllers aggregate local sensors and actuators, while major processing is consolidated centrally. This can reduce wiring weight and complexity, while enabling more scalable software deployment across models. Central compute provides high-bandwidth processing and orchestration, whilst zonal controllers focus on sensing and acting at the edge, translating high-level commands into actuator signals, and returning telemetry.

Incumbent OEMs face a transitional “messy middle.” Rather than replacing proven legacy architectures outright, many are layering new architectural concepts on top of existing platforms. This pragmatic pathway reduces delivery and validation risk but increases interim integration burden. The result is a period where legacy, domain and zonal concepts may co-exist within the same vehicle programme or across the same model portfolio.

This reinforces the importance of clear requirements, early validation, and virtual deployment. As architectures become more consolidated and software-dependent, integration risk is increasingly driven by interfaces, resource constraints, timing behaviour, and cross-domain dependencies. Modern vehicle architecture becomes a key enabler of SDV, but also a source of transition complexity for established OEMs.

Cloud and data as new value drivers

Cloud and data capabilities are central to SDV because they enable vehicles to operate as connected, continuously improving platforms. Cloud infrastructure supports fleet telemetry ingestion and storage, analytics and AI training environments, OTA update management, and digital twin infrastructure.

Together, these create a closed loop where vehicles generate data, cloud platforms convert into insights, and validated updates are deployed back to vehicles at scale.

Value increasingly sits in actionable intelligence rather than raw data. Fleet-level insights can enable services such as predictive maintenance, fleet and telematics software-as-a-service (SaaS) offerings, diagnostics, energy optimisation, and risk analytics that support insurance services. These services depend on reliable data pipelines and consistent interfaces between vehicle and cloud.

Data ownership and commercial exploitation introduce governance challenges. Privacy boundaries, access control, consent management, data quality, cybersecurity, versioning across fleets and OTA release governance become critical. As different vehicles operate with different software versions, OEMs must be able to understand which data came from which vehicle configuration and how insights should be interpreted.

Without robust architecture and governance, scaling data-driven services increases operational risk and constrains value capture. The commercial opportunity is significant, but it depends on the ability to manage data as an operational asset, not simply as a by-product of connectivity.

Continuous updates and lifecycle evolution

A defining SDV shift is that development does not end at SOP. Rather than a single final release, the vehicle becomes a living software platform with a continuous pipeline of updates, with security patches, bug fixes, performance improvements, and new functionality delivered OTA.

This changes the lifecycle model for both the product and the organisation. Product and software teams need to persist beyond launch to monitor the live fleet, manage software versions, respond to issues, and deliver controlled improvements over time. Real-world telemetry creates a feedback loop back into engineering, supporting refinements and safety improvements that previously may have waited for the next model cycle or service campaign.

However, continuous updates also increase operational responsibility. OEMs must manage compatibility, cybersecurity risks, validation evidence, release governance, and customer communication across vehicles that may not all be running the same software configuration.

The enabling infrastructure, including secured platforms, data pipelines, analytics, update management tools and digital twins, becomes essential to operate SDVs safely and efficiently at scale. Continuous lifecycle evolution is a business and organisational shift in how vehicles are supported post-launch.

Future direction and UK opportunity

SDVs are shifting value from a hardware-centric model towards software platforms that require continuous integration, deployment, validation, cybersecurity, and lifecycle management, while increasing reliance on high-performance hardware platforms that remain critical enablers of software capability and performance.

As vehicles become more connected, software-intensive, and updatable, competitive advantage will increasingly depend on the ability to control key software layers, manage release cadence, validate changes safely, and convert vehicle data into operational and commercial value.

Industry direction

The direction of travel is towards greater OEM control over the SDV stack. This does not mean OEMs will own every layer outright, but they are likely to seek stronger control over the areas that determine speed, quality, safety assurance, and monetisation.

Key areas of increasing OEM control include:

- Software platforms and middleware, where standardised interfaces enable software reuse and updatability.
- Integration environments, including software factories, CI/CD pipelines, automated testing, and release governance.
- Compute and silicon decisions, as central compute platforms become more strategically important.
- Data and cloud interfaces, where control over vehicle data supports services, diagnostics, OTA updates, and customer experience.
- Lifecycle operations, including version management, fleet monitoring, cybersecurity response, and update deployment.

This reflects a broader shift from traditional supplier-led delivery towards more platform-led operating models, where OEMs seek to own the critical interfaces and governance points even where delivery remains ecosystem-based.

UK opportunity

The UK's strongest opportunity is in the high-value enabling capabilities that help OEMs and suppliers develop, integrate, validate and operate SDVs safely at scale. These capabilities are increasingly central to competitiveness because SDV value is not only created at vehicle launch, but through the ability to improve, assure and update vehicles throughout their lifecycle. The UK also has a strong foundation in semiconductor engineering, creating opportunities to support the development of advanced compute platforms underpinning SDV capability.

Priority areas could include:

- Systems integration across software, hardware, data, and suppliers
- Physical AI, data science and machine learning operations capabilities to support development, deployment and continuous improvement of SDVs
- Virtual validation and software factory capabilities for repeatable build, test, release, and SOTA deployment
- Cybersecurity, safety assurance, and OTA governance
- Cloud, data and fleet analytics for diagnostics, update monitoring, lifecycle improvement, and AI-driven insight generation
- Specialist engineering services around calibration, diagnostics, platform conformance, and validation evidence.

The UK opportunity is not simply about participating in the SDV transition, but about becoming a trusted centre for the capabilities and talent that underpin SDV safety, scalability, and commercial viability. The areas of greatest opportunity are integration, validation, cybersecurity, software factory maturity, data intelligence, and lifecycle software delivery.

As SDV architectures mature and software-defined capability extends further into safety-critical and autonomous functions, the value chain will continue to evolve. Ongoing research will be needed to track where value is moving, where capability gaps are emerging, and how the UK can position itself in the next phase of software-defined mobility.

Integration as a key SDV differentiator

In SDV, integration shifts from a late-stage programme activity to a continuous lifecycle capability. Traditional automotive development still requires the rigour of systems engineering, traceability, and physical validation, particularly for safety-critical functions. However, SDVs also require faster and more iterative software delivery before and after SOP.

Testing, verification, and validation must therefore be conducted earlier and become more automated. A modern integration chain focuses on system requirements and architecture through MiL, SiL, vECUs, HiL, system-level testing and vehicle validation, followed on by post-SOP monitoring and phased OTA releases. Virtual validation reduces dependence on scarce physical assets and allows earlier detection of architecture and resource constraints, especially where many functions share centralised compute.

Software factories industrialise this capability: standardised toolchains and CI/CD pipelines automate build, test, and packaging steps, supporting repeatable releases with predictable quality. In SDV, delivery capability, building, integrating and validating safety at scale becomes as critical as the software features themselves.

This also creates a need to assess software readiness differently. In an SDV context, readiness is not only a measure of whether the software product performs its intended function, but also whether the software factory is mature enough to build, validate, release, and monitor it repeatedly. Software product maturity covers requirements, design, integration, and safety, while software factory maturity covers CI/CD pipelines, automation, virtualisation, V&V capability, OTA packaging, and fleet monitoring. This distinction is important as SDV readiness depends on both the software and the repeatable delivery system around it.

Authors

Angad Jessel
Automotive Trends Analyst, Advanced Propulsion Centre UK

Dr Hadi Moztarzadeh
Head of Technology Trends, Advanced Propulsion Centre UK

The APC would like to thank our industrial and academic stakeholders, and in particular, the Automotive Council members that contributed into this report and value chains.

Further information

If you have any questions or would like more detail on any of the data email **info@apcuk.co.uk**

Sharing this document
This report is produced by APC UK, when sharing the content of this document, please reference APC to help others connect with our research and support.

The Advanced Propulsion Centre UK
University Road
Coventry CV4 7AL

apcuk.co.uk

@theapcuk
Advanced Propulsion Centre UK

Department for
Business & Trade

ADVANCED
PROPULSION
CENTRE UK

Accelerating
Progress