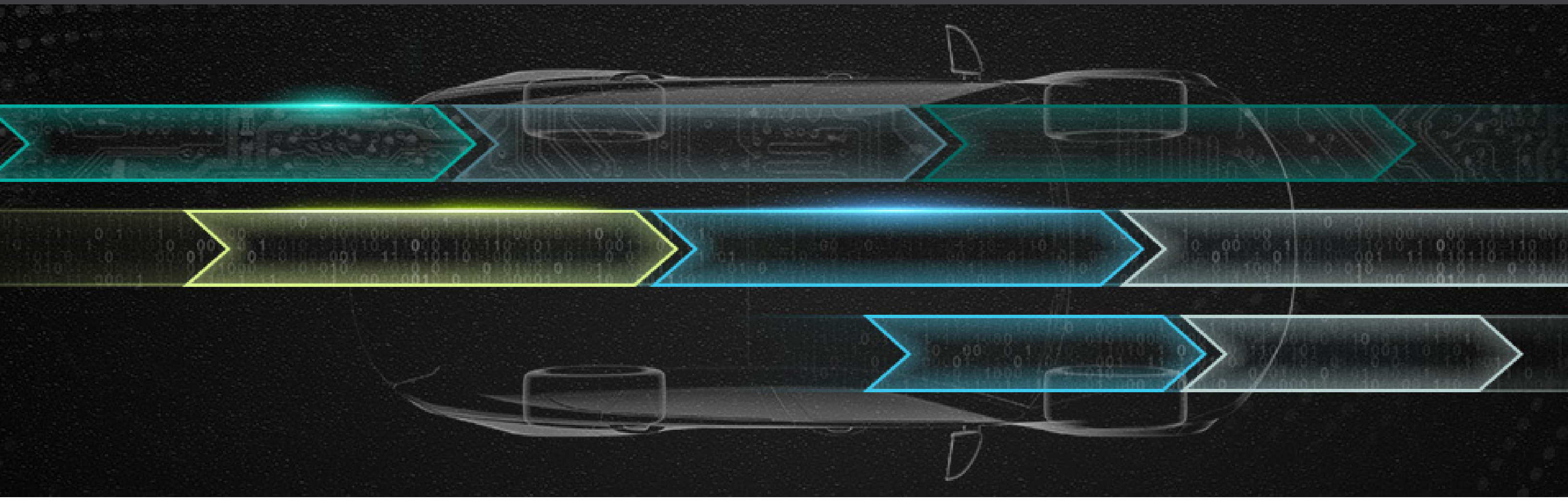


Decoupling Hardware and Software Accelerates Development

Software defines the most advanced features of current and upcoming vehicles, from infotainment and connectivity to advanced driver-assistance systems and autonomous driving. As more consumers expect those features to be up to date — much like apps on their smartphones — OEMs and their partners need the freedom to meet those expectations through over-the-air (OTA) updates that extend throughout a vehicle's life.

Decoupling software from hardware provides that freedom.

Importantly, it also enables OEMs to meet global, regional and national requirements to use aftersales software updates to keep vehicles compliant with the latest regulations governing ADAS, automated driving and other functions.

THE TRADITIONAL APPROACH

Historically, automotive companies developed software to run on specific, specialized electronic control units (ECUs). A premium vehicle contains more than a hundred such ECUs, connected by a huge spiderweb of cables for the power and automotive bus systems. The cables are bundled into thick and heavy cable harnesses, which usually have to be assembled by hand. During production, a harness then has to be placed manually into a vehicle body and manually connected to all of those ECUs through the vehicle. It is nearly impossible to change anything after the vehicle is sold.

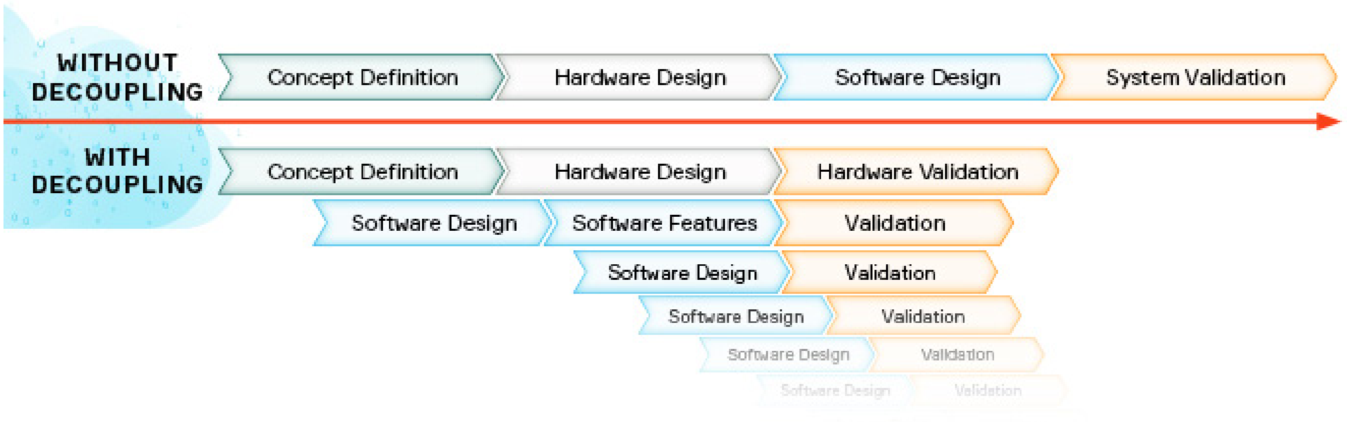
Software architecture has not been treated much differently. Each ECU contains features for which the software interfaces are defined at a network protocol level for CAN, LIN, FlexRay and Ethernet. This definition also includes time constraints, such as cycle times. When a vehicle and all of is components are validated, they are considered final – and they are not expected to be touched again except for regular service.

The development process worked sequentially. First came the product definition and requirements phase, which started at least five years before the start of production — followed by hardware design, software design, a request for information, a request for quotation, a sampling phase and a bidding phase. Then the integration and test phase would start, concluding with vehicle validation and ramp-up of production. Each phase of development had to be completed before the next could begin.

Even though OEMs created toolbox systems for car lines – which helped developers reuse mechanical components and ECUs for new car lines – it still took a lot of work to integrate those ECUs with new ones. If the hardware changed, developers had to build all-new software for each new hardware platform because they were unable to efficiently reuse code between those platforms. Once set, requirements were rigid, and incremental development cycles presented few opportunities to revise and test code. Integration of a high number of different ECUs together in a vehicle contributed to complex testing and validation processes and made it very difficult to address bug fixes when the vehicle was in the field.

PARALLEL DEVELOPMENTS ACCELERATE TIME TO MARKET

Decoupling hardware from software is an important element in modern automotive development that allows engineers and developers to create more features, faster.



THE DECOUPLING ADVANTAGE

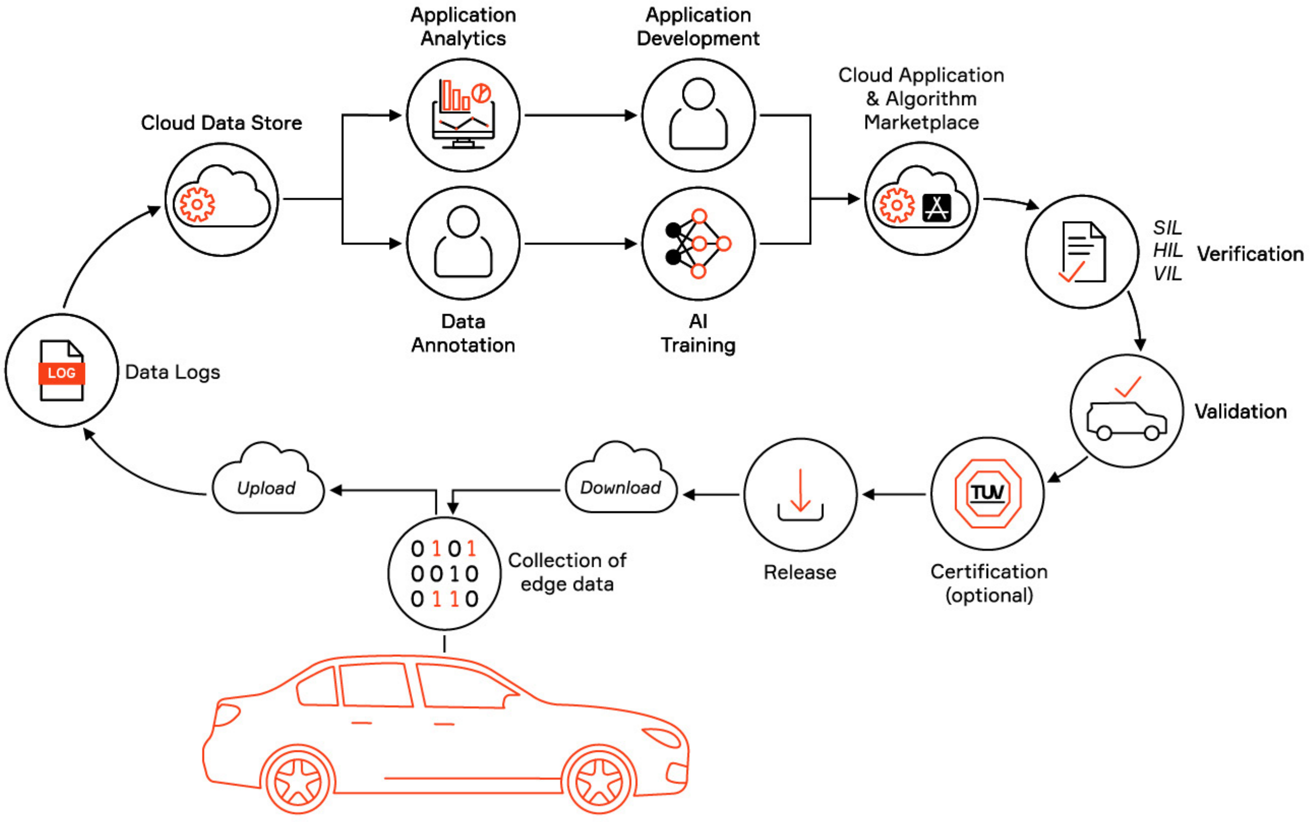
In a decoupled architecture, hardware and software are free to evolve on their own independent development timelines and update cycles. In addition, different software modules are decoupled from each other using various technologies, creating what we call "freedom from interference" between software modules. The separation minimizes the impact of software changes and, where possible, obviates the need for revalidation of an entire vehicle when changes are made. With decoupling in place, software modules can receive updates more easily over the life of the vehicle — and can run on different hardware platforms.

This approach also means that developers can more easily consolidate — or up-integrate — software that today might run on independent ECUs distributed throughout a vehicle onto a smaller number of "serverized" hardware platforms.

Serverization centralizes the processing, but it also brings benefits from the traditional IT server and cloud worlds into the vehicle. On those platforms, software applications share resources, efficiently supporting the same or greater processing capability and achieving freedom from interference in both software and hardware, while allowing manufacturers to remove microcontrollers, power supplies, housings and wiring to cut costs, weight and complexity.

Cycle Through the Cloud

Decoupling software from hardware allows OEMs and partners to continuously improve software through a feedback cycle from development, testing and release through data collection and updating.



With software untethered from hardware, OEMs are better empowered to control the software that defines their vehicles, and they can work with partners to define features closer to their release date. Instead of specifying requirements three to five years early — which equates to several generations in the consumer electronics world — the software features that OEMs and their partners specify today could roll out months later to vehicles in production or in the field that are using different generations of hardware. This vast improvement in speed to market, which is closer to the pace set by today's smartphone apps, results in features that more closely match consumer expectations. Certainly, there will be areas where the OEM will keep the update frequency low, such as with safety features. There will also be areas that require a very rapid update, such as a security patch or the fix of a corner case in automated driving.

A BETTER SOFTWARE DEVELOPMENT PROCESS

Just as important, decoupling allows OEMs to embrace modern DevOps methods and tools for software development. Continuous integration and continuous deployment (CI/CD), along with continuous testing, are key to accelerating development while improving software quality and allowing more teams to contribute to a transparent process. CI/CD enables small, autonomous teams to develop software through frequent iterations, integrating and testing code early and often, thus creating a feedback loop for quickly adding features and improving code.

Further, by standardizing a toolchain, OEMs can make development practices consistent across all internal teams as well as those of partners. Separate teams can create, own and update applications — up to and beyond the start of production — and standard APIs maintain the independence of those applications while allowing them to be automatically integrated with software from other teams.

Some developers already use CI/CD methods, but the next step is to establish one consolidated, cloud-based toolchain in which developers working together around the world can automatically integrate their latest code and deploy it into a hardware-in-the-loop testbed or a test car environment at any site. OEMs, suppliers and other third parties can collaborate with a single management view into all toolchains for governance and security.

ENABLING NEXT-GENERATION ARCHITECTURES

Having a platform that uses standard APIs – or a "software landing zone" concept – allows companies to manage vehicle features within a single central source for everything in the system. The platform contains models for the interfaces on all of the modules in a vehicle, as well as the mapping to different vehicle hardware architectures. This means that developers can create software features that can be compiled for different hardware configurations without altering the feature itself.

This is an important concept to enable the Smart Vehicle Architecture™ approach. SVA™ simplifies vehicle architecture and consolidates functions from multiple ECUs onto a flexible combination of zone controllers, a central vehicle controller and an open server platform. Moving to that architecture – and making it scalable and updateable over time – requires abstracting software from hardware in that way.

THE WAY FORWARD

Decoupling also takes advantage of OTA capabilities, allowing developers to push the most robust versions of their software to all vehicles in the field so that consumers always have the best features that their vehicle's hardware can support.

Developers can even use vehicles in the field to help them refine software on an ongoing basis. Through "shadow testing," test code can run in parallel to production code on every vehicle, and the system can report on how the test code performed, uploading data to the cloud. Analysts can combine the data with data from other vehicles, use the results to inform changes to the test code and pre-validate the changes in the cloud. If the software requires more extensive validation – perhaps because it is safety-critical – the developers would then run the code in vehicles in a test fleet. Either way, when validation is complete, the updates would be pushed back to the vehicles as production code.

Decoupling is a fundamental change, but it can begin with smaller initiatives — for example, the formation of small, autonomous teams to develop pieces of software that can run independently on the same domain controller. By adopting automated tools, developers can gradually integrate these applications via standard APIs.

This change calls for a shift in mindset toward new organizational interfaces among currently separate hardware, software and networking teams. But it also provides an opportunity to implement real software-as-a-product and software-as-a-service strategies in automotive development and bridge the gap between the vehicle and the cloud.

Over the past five years, Aptiv has not only implemented CI/CD with a high level of automation hosted completely in the cloud, we have also created innovations that achieve faster development in spite of increased software complexity. Our extensive development experience helps OEMs lead in defining the next-generation software-defined vehicle.

ABOUT THE AUTHOR

Ralph-Heiner Haar
Senior Manager, Engineering

Ralph-Heiner Haar is passionate about software-defined vehicles and related innovations, and believes that complex software-defined vehicles will not be possible without a quantum leap in engineering capabilities. Engineering will become highly automated, and augmentation will help engineers do their work more efficiently and flawlessly. Ralph has more than 26 years of experience in development, architecture and management at automotive OEMs and Tier 1 suppliers in Europe and Asia. Every day, he works to build high-performing teams, deliver at the speed of business and innovate beyond expectations.

LEARN MORE AT APTIV.COM/SVA →