The Journey from hardware-centric to function-centric car development

nagarro

Table of contents

Introduction

Have you ever wondered why connected car services are still a niche product in our highly connected life? What makes them so unique? Because building these services is a highly complex task, dependent on multiple factors. To understand the sheer complexity involved in designing and delivering connected car services, you should first understand how the car design and manufacturing process have changed over time.

This e-book walks you through the architecture, challenges, and best practices of adopting connected services. You can also know more about the differences in the approaches of new OEMs as compared to traditional OEMs.

Onboard vs. offboard-centric organizations

For decades, a car has been treated as a collection/integration of hardware products. Each new generation of cars has come with newer and better hardware-driven features like a better lighting system or better driving capabilities/experience. All these years, the hardware has been the driver of innovation, and hence:

- **OEMs always looked at selling cars as a one-time task**
- **Pricing was seen as a bottom-up, cost-driven task**
- **They adopted a sequential and static development process**
- **Hardware quality has reflected the car quality**
- **They saw software as an enabler for selling hardware**

The new-age tech companies are leveraging their capabilities to reimagine cars as a collection of hardware and software features. The new approach to cars looks at hardware as actuators and sensors; everything else is dependent on the software.

Such a paradigm shift in how cars are imagined and designed has opened the doors for easy application of data-driven mindset and business, ready-to-use security measures, agile and flexible ways of working, and efficient software ecosystems. The focus has now shifted to software and delivering functions, intelligence, and experience to the customers.

Benefits of this approach:

- **Selling cars is now seen as the starting point of a long, connected journey**
- **Pricing is now a top-down value-based task**
- **Manufacturers adopt an agile development approach**
- **Software quality significantly impacts the car quality**
- **Hardware is adopted to encourage selling software and services**

Connected cars between offboard and onboard

Adopting connected services might seem simple to an outsider. For instance, if an OEM plans to launch automated parking or keyless door unlock systems, can't they connect these applications with ready-to-use backend services? An OEM just needs to simply connect its existing Onboard solutions with a connected Offboard platform hosted by the OEM or purchased from an external service provider. Right?

Wrong. In reality, things are quite different. Currently, the in-cabin connected experience lacks the intuitiveness or user-centric approach of connected solutions used at homes or offices.

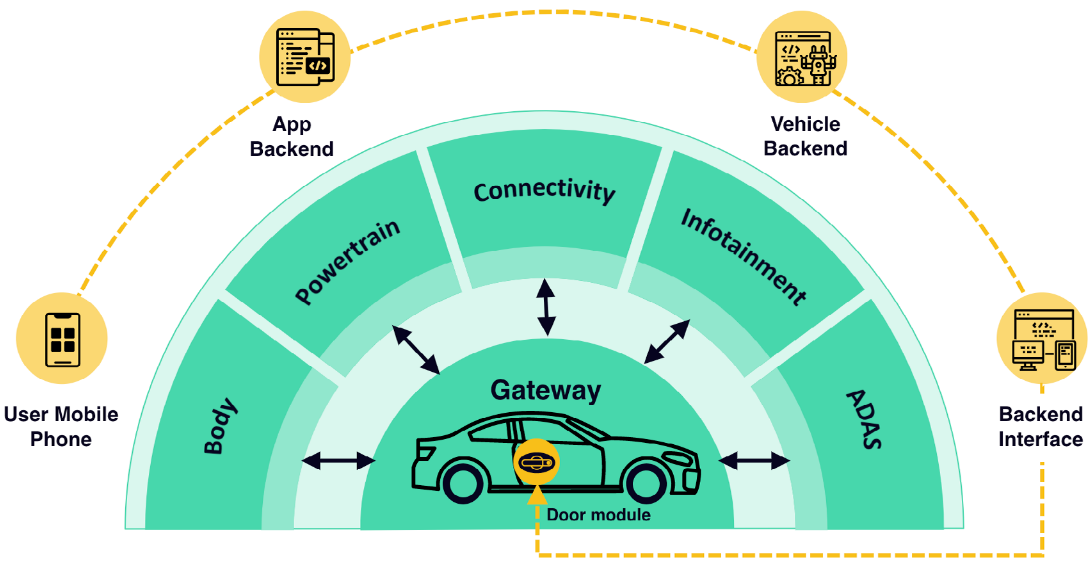


Figure1: Connected vehicle between Offboard and Onboard

How does product architecture enable the shift towards connected services

Product architecture reflects the organizational structure and influences and drives other product-related decisions. In Automotive, the Electric/Electronic (E/E) architecture of the vehicle influences the data collection strategy, testing concepts, ways of working, ways of feature implementation, teams' cooperation models, and pricing models.

Let's deep dive into the E/E architecture to understand the challenges faced while implementing connected services.

Traditional onboard-centric company

Till the '70s, vehicles used to be entirely controlled by hydraulic and mechanical systems. In the ensuing decades, electronic chips took over the car systems, especially the most complex ones, like the engine. For years, car systems were built around electronics where the focus was on replacing local functions through monolithic software and hardware systems.

When required, the architecture can be extended to new local domains, like offboard servers for connected services or the ADAS domain for driving assistance systems. The resultant E/E architecture is more distributed and domain-centric. Each domain focuses on its software development in a domain controller and communicates with other domains through gateways and backbone technologies to seek information and execute tasks. For instance, an ADAS domain will often use infotainment systems to display driver warnings and offboard systems to download online data like maps and powertrain to drive the car autonomously.

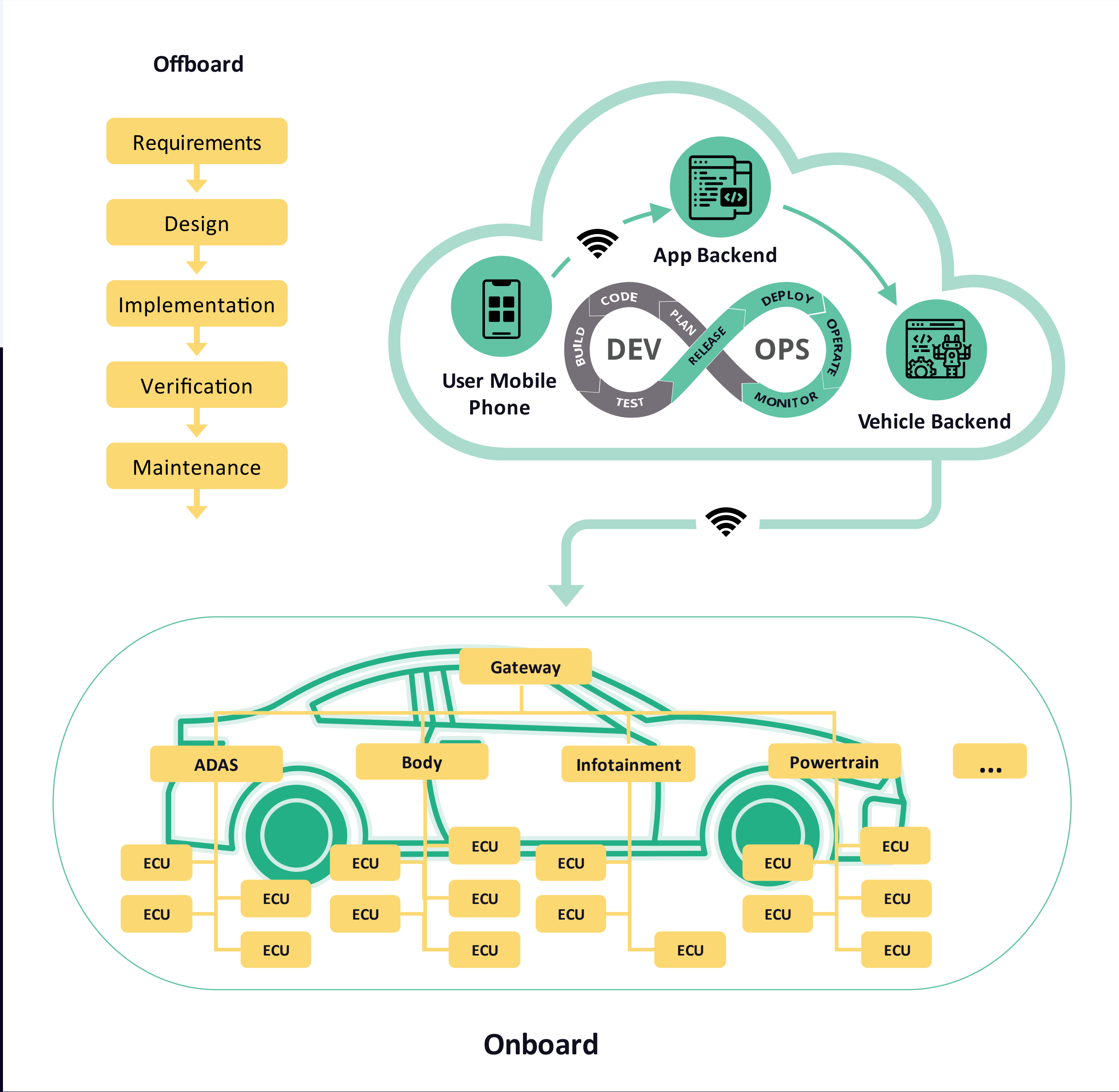


Figure2: Domain-centric E/E architecture

A seamless and intuitive customer function to enhance the in-cabin experience requires the contribution of different domains like remote services, drive functions, display & infotainment features, and even body actuators like ambient lights. Developing such features requires the cooperation of different distributed domains with a lot of complex data communication, thus making testing, architecture, security, safety, and data collection much more difficult.

The rapid growth of vertical software and function development has left companies with huge business units acting as silos, functioning completely independently from each other. They are often unaware of the functions, benchmarks, products, goals, competitors, and challenges of the other departments of their company. Connected services are implemented through adding a new silo "Offboard" to bring the backend capabilities to the existing Onboard capabilities.

Moreover, Automotive has now become a mature industry, with more than 13 decades spent in incremental changes of a static product, with minimal willingness to change or evolve. This has led to a complex portfolio and a fragmented business.

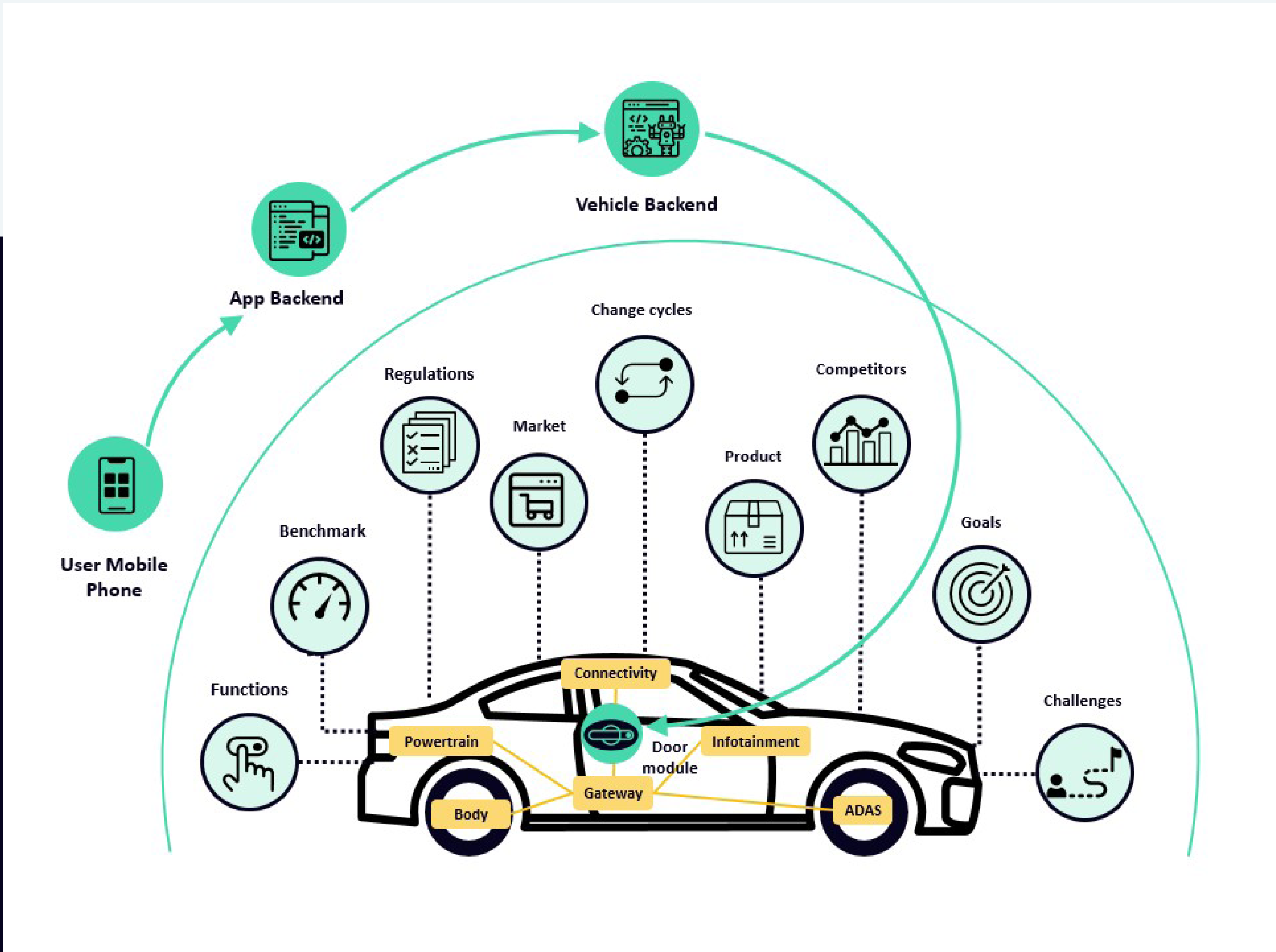


Figure3: Onboard-centric organization structure

New digital offboard-centric company

Over the last two decades, this shift from pure hardware to more software and service-based products has opened the automotive market for new-age, technology-led companies. All these new entrants come from the offboard/IT part of the car. With their backgrounds, these companies are reimagining the car as an IT product. They put customer needs at the center of product design, and hardware is seen as a utility to sell services.
While developing and selling a car is certainly an OEM's forte, developing and selling new software and services is the playing field of technology experts. It is important to have a good mix of both tech and hardware to deliver a seamless customer experience.

We must think about how a software developer would develop a vehicle function and how such a thought process would facilitate the transition to becoming a connected services company.

An IT expert who develops vehicular functions will go through the following tasks:

- **Download APIs for the basic functions of the sensors, actuators, and offboard data**
- **Develop a functioning architecture by leveraging the API library**
- **Choose a computer with the needed computing and memory resources for the real time part of the function architecture**
- **Use the cloud as a resource extension to compute non-real-time tasks**
- **Request API owners to develop their basic functions**
- **Develop, test, and roll out the core functionality to the central computer in an agile manner**

During this process, the vehicle will be reimagined as a central computer(s) where all the car intelligence and functionalities are hosted, with an optional extension to the cloud for cost-efficiency and flexibility. Only the basic functions which are hardware-dependent (such as the braking system) will be implemented in a monolithic and waterfall model, while their functions will be exposed in the form of APIs. (Refer to Figure: 4)

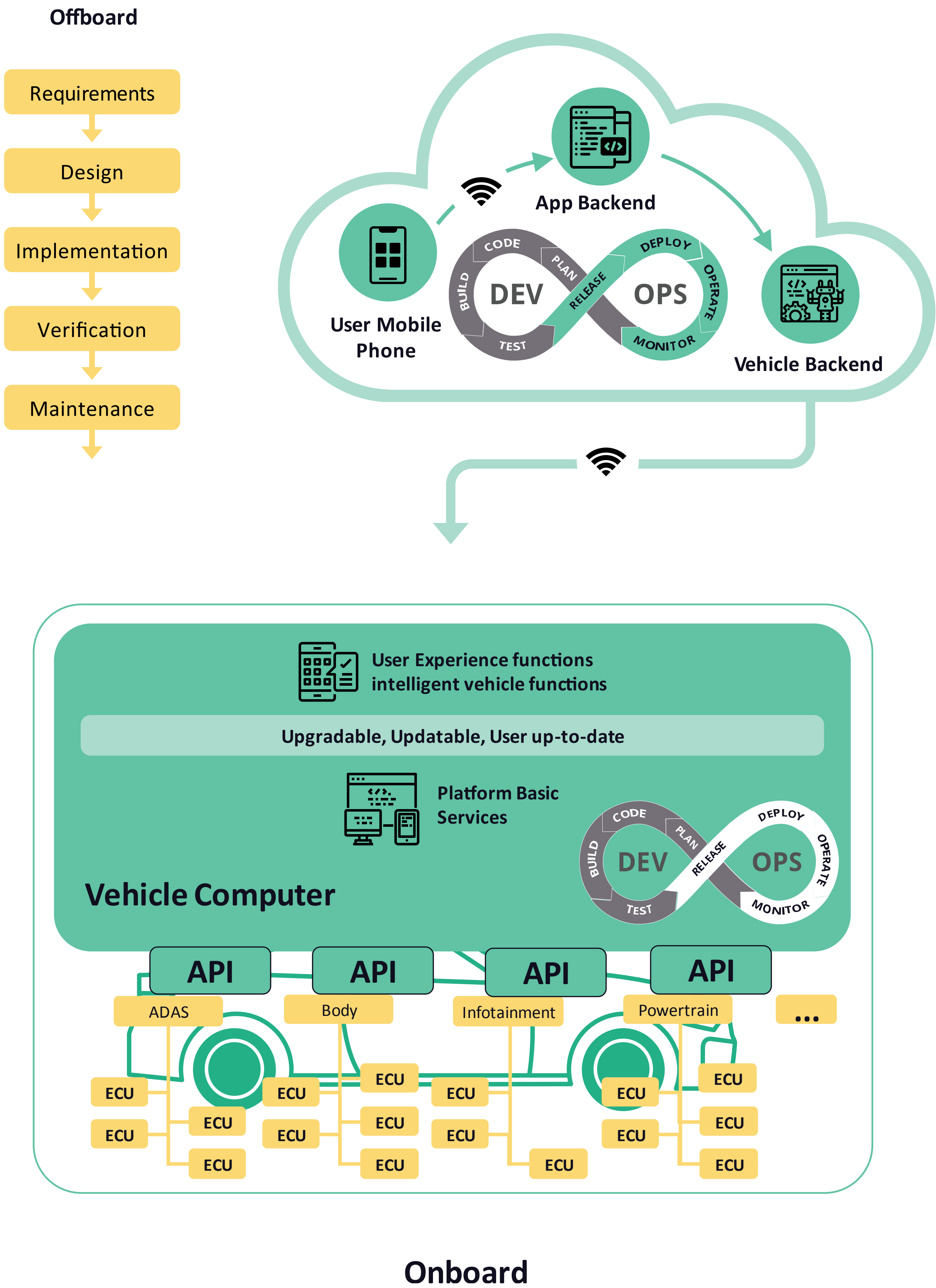


Figure4: Software development lifecycle ecosystem

This process will facilitate data collection as data now flows inside the same vehicle's computer and is stored in the same internal flash. It is easier to access data, providing key insights into each vehicle's functionality. This streamlines the architecture and implementation of complex functions, automates testing, and reduces the dependency on hardware.

As the organization structure evolves with the product's architecture, new-age mobility companies will comprise:

- **Business departments responsible for planning, developing applications and testing the integrated products**
- **Functions and another internal service provider responsible for giving the needed onboard/offboard hardware and ecosystem.**
- **Deploying and testing the build and integration of the whole platform.**

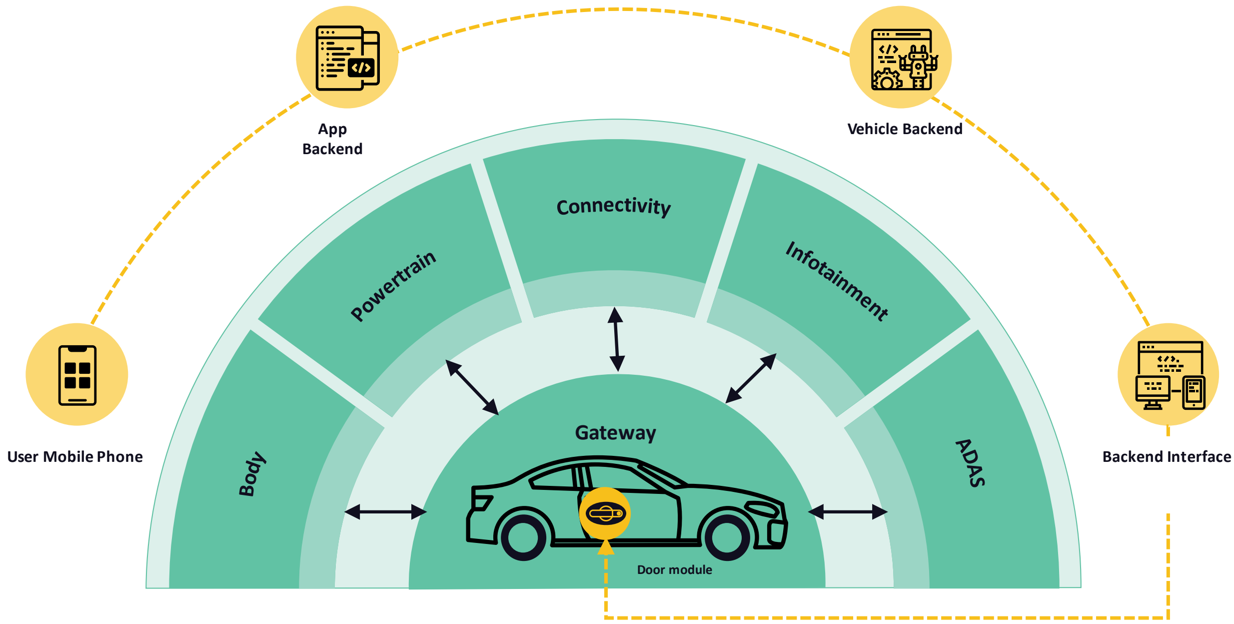
The End-to-End Feature Dilemma

Connected services are overarching features that go beyond one, two, or even three domains. It is an end-to-end service that starts from the user's mobile phone and goes through different backend systems inside the connected platform. It then uses an edge connectivity component to enter the vehicle's onboard part through one or different sensors, gateways, or actuators. To develop a simple remote door-unlock service, you need the cooperation of at least four domains, making it difficult for the OEM to develop and test new end-to-end features.

Developing a connected service demands a lot more than just connecting the involved domains. It comes with challenges in each step of the feature development process - from ideation and architecture through implementation until test and roll-out. Let's deep dive into them and take a 'remote door unlock' feature as an example.

Tackling these challenges requires a company to be focused more on customer experience than on business units. OEMs need to leverage new technologies such as AI, DevOps, IoT, or Cloud to reach an aligned vision.

In the following section, we list the challenges faced by traditional OEMs during the feature development phases (V-model), from the idea to the realization and roll-out. We then look at ways to address these challenges.



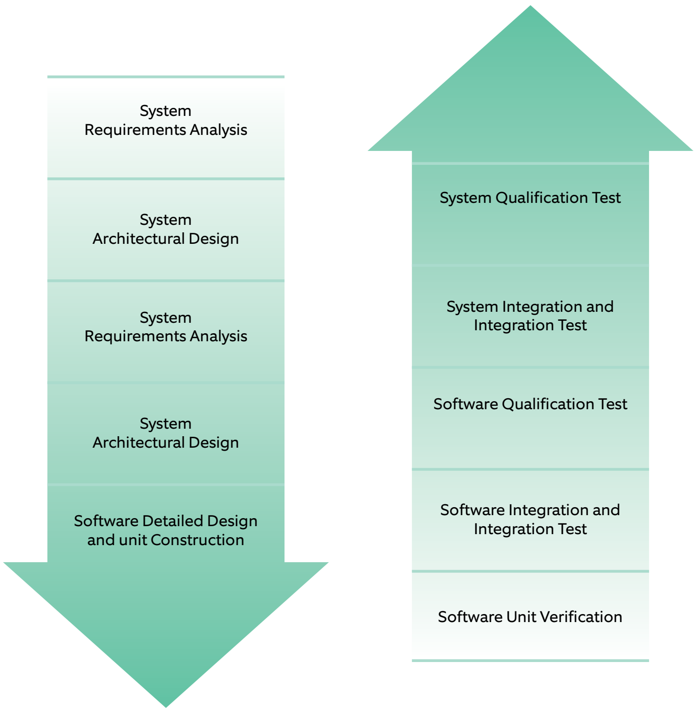


Figure5: Challenges in all feature development life cycle gates

Ideation and business plan

Challenges: Each business domain within any organization tries to grow its business by building on the existing features. Any domain can initiate the ideation process for a new feature. It could be the offboard business unit that may reach out to onboard domains to make their features remotely accessible or the other way around. In both cases, the initiator will need help from other relevant business units to evaluate the feature feasibility and create a business plan. However, the business plan for the new feature may be extremely specific, meeting the business perspective of only the concerned business unit.

Solution: OEMs must understand customer requirements by using big data analytics. This will help create a customer-specific product portfolio built by utilizing data-driven insights. The organization must also be restructured.

The enterprise must restructure around business lines with two dedicated teams, one for ideation and strategy and the other for infrastructure and technology.

This will create a data-driven company that knows how to attract customers by delivering some basic features and enhanced customer experiences. Through connected services, they will also know how to create new ways of a continuous and direct relationship with its customers.

As service providers, OEMs must learn how to price and sell software by using a top-down approach instead of a bottom-up expenses estimation. They can no longer use the 8% margin formula as it would not cover the upfront investment to build software capabilities and infrastructure. According to BCG research, OEMs can improve their margin by up to $2600-$7500 per car in the premium segment through new price models and lower material costs.

Architecture

Challenges: Currently, OEMs do not have integrated teams that oversee the development of end-to-end features. The initiator business unit builds the architecture based on their understanding and support from other business units. The efficiency of the architecture depends completely on their cooperation. The initiating business unit will come up with oversized architecture solutions based on assumptions about the parts that fall under the other business units. The result will be an underwhelming customer experience because the ideas came from different business unit-centric use cases with different goals in mind. Today, the in-vehicle architectures are domain-centric, hardware-driven, and decentralized, thus increasing the number of involved components for a single connected service.

Solution: The OEM must enforce a DevOps and agile mindset where both departments come together. They should be led by E2E enterprise architects to ideate and discuss cross-functional topics. Hiring agile consultants can help build this culture. Moreover, the in-vehicle architecture will follow the enterprise structure to be more centralized and developed around customer features, where the cloud is seen as a seamless extension of the vehicle's resources. The vehicle API securely exposes the car functions through robust API management to onboard third-party developers. They help in bringing creative and innovative ideas for personalization and customer experiences. As a result, the organization can build a service ecosystem and even a vehicle app store without spending resources on ideating or developing applications themselves. The next stage would include the OEMs standardizing towards a single-vehicle API that will encourage using off-the-shelf applications. This will reduce the development and deployment cost drastically. At the same time, more external developers will be motivated to start this journey, as was the case with the Android ecosystem

Implementation

Challenges: As the offboard and onboard departments take different approaches ranging from agile to waterfall, there is not enough collaboration or coordination on the incremental product features. Each department focuses on implementing its product without taking any input from other teams. The result? A plethora of monolithic products where hardware and software are mixed and tailored for specific use cases.

Resource allocation for hardware and software is often cost-driven. In most cases, the features are built to meet the needs of the use case at hand. This limits the creativity of developers and makes it difficult to deliver feature upgrades.

Each department/domain uses its own toolchain, programming language, and software solutions, preventing a smooth and seamless customer experience.

Solution: OEMs must adapt architectures with completely decoupled hardware and software implementations. They should introduce general-purpose computing hardware, with a good abstraction layer of general-purpose operating systems like Linux and well-standardized middleware. This will ensure that the teams developing the software will deliver on the same platform by using the same tools. The agile processes will also ensure higher cooperation and collaboration, with a focus on customer experience.

Introducing such powerful computing processors with the option to move functions seamlessly to the cloud (which is continuously connected to the vehicle) will increase the creativity of developers. Features will be continuously integrated, deployed and updated, opening new revenue streams for OEMs.

Testing

Challenges: Testing is mostly done on the component level inside each domain. It is hard to coordinate and execute integration and system tests between domains because business units have different test plans, strategies, and tools which lack synchronization. Test cases and plans are not compatible across teams and are often locally accessible. The most connected services are OEM-specific, with less standardization where we cannot rely on API testing. In this case, the feature must be tested end-to-end (E2E) where onboard devices are needed in their real form, like in car tests, as a simulation like hardware in the loop or software in the loop. The main problem will always remain that OEMs today try to rely a lot on vehicles to test the E2E behavior of a feature. This can be problematic as the offboard part is not included in vehicle core development and must be integrated into the test case.

Solution: Centralized architecture makes system testing easier as the end-to-end connected services are no longer distributed over multiple components. They are distributed over a maximum of four components – an in-vehicle platform including a connectivity solution, a cloud solution, an optional front-end app, and an edge sensor or actuator.

The OEM must formulate teams or communities to oversee the system testing tasks and control the output of unit tests. As the OEMs start moving more towards CI/CD habits that enable feature delivery in weeks and hence need E2E testing, the importance of test automation will begin to increase. Utilizing test automation frameworks and expertise to integrate existing onboard simulations into the offboard test suites is highly effective.

Data

Challenges: As they make the transition towards connected vehicles, OEMs have realized the importance of data, especially in understanding customer preferences and behaviors which further enables them to customize products. This requires efficient data collection. With multiple teams working in silos, data is also often stored in silos, making it ineffective and difficult to use. Organizations need multiple scripts to collect ECU-internal data or over communication busses.

Solution: Since the core functionality is implemented in the vehicle computer, most contributed parts are hosted on the same component. Consequently, data collection will just comprise storing internal computer data in a flash or similar non-volatile memory to be retrieved when needed by the cloud.
One of the main goals of collecting onboard data should be to apply AI-powered solutions like autonomous driving capabilities and autonomous security features. But AI needs qualitative data in real-time. The proposed data architecture is robust and highly performant and can provide the data needed to apply AI at the edge. AI-based systems can be designed and implemented with different complexity levels, from simple real-time predictive maintenance up to a network of interoperable Ais, paving the way towards fully autonomous driving.

The proposed data infrastructure will lay the foundation for AI-driven systems that can lead. After developing (or even while developing) the AI strategy, we can start our end-to-end AI implementation process, as shown in the following figure 6

It starts with a general understanding of AI and its possibilities for the OEM business. Discovery workshops will help identify and prioritize AI use cases. It's a well-structured and proven approach to concretize one or two of the most promising AI use cases. As the next step, we will implement a proof-of-concept (POC) to get the first AI-system results. These results will then be analyzed and tested by the OEM business and IT. The PoC results will then be transferred into production. Once all this is done, the continuous learning/improvement pipeline shall finally be implemented.

Security

Challenges: The security architecture is closely dependent on the E/E architecture and fragmented into different domains. For instance, in the absence of end-to-end identity verification between the door module and the mobile app, security can be compromised. Security identity verification require multiple teams functioning in silos to work together.

Solution: The Security approach must be top-down, starting from system design. The first involves identifying different parts and components where the security features must be incorporated. Only such a consideration will then comply with the newly released security standard ISO/SAE 21434 and as well as the security extensions of the ASPICE standard, which puts the security task in a V-model format. A simplified and more straightforward system architecture like the one explained above can help in getting a strong security architecture.

Communities and cross-functional platforms will discuss more efficient end-to-end security measures. Using powerful computing hardware will help adopt strong security measures complemented by strong hardware security modules and a trusted execution environment to host secure applications. As the adopted hardware and software platforms are more enterprise and general-purpose and no longer automotive-grade, security practices that have been well established for decades in the IT world can be reused and adapted to the vehicle onboard, namely the IdAM, firewall, sandboxing systems, Linux hardening.

AI UNDERSTANDING

Workshop

- General understanding
- Presentation of AI Basics
- Example use cases
- Identification of areas of interest

AI USE CASE IDENTIFICATION

Workshop(s)

- Identification of business applications
- Structured evaluation based on predefined criteria
- Data driven Decision support
- AI-PoC implementation plan

AI APPLICATION-MANAGEMENT

Operations

- End-to-end smart application
- Continuous Learning
- Support for AI applications
- Onshore, nearshore and offshore models available

AI PROOF OF CONCEPT

Lab environment

- Business specification
- Decision for technology and AI components
- Agile implementation in the lab environment
- Train/test AI model
- Presentation results

IMPLEMENTATION

Project

- Business specification
- Decision for technology and AI components
- Agile implementation in the lab environment
- Train/test AI model
- Presentation results

Figure 6: Central vehicle computer as a system on the chip

Culture

Challenges: As traditional OEMs grew to be a coexistence of different silos in terms of challenges, goals, used tools and ways of working, developing a new horizontal feature will need cooperation between teams across silos. These silos would have worked earlier in significantly different ways. For example, an Offboard team will typically deliver often in a pure agile way using sprints lasting only a few weeks, as they have less dependency on hardware.
Meanwhile, an onboard team will have far longer development cycles where features must be planned years upfront because of the monolithic architectures and the right coupling of the software with the given hardware.
Other teams, especially the ones belonging to onboard and having continuously interface to offboard, will try to adopt a hybrid model between agile and waterfall. Such teams will be significantly different from each other in terms of culture.
All this means that cooperation and coordination could be extremely difficult in all stages of product development.

Solution: It is important to conceptualize and implement a culture with a common goal and mission through agile masters. This will require restructuring the organization to be more business-driven and customer-centric. This will create a digital and agile company aligned toward one authentic mission. It differs from other brands through an enhanced, available, and performant customer experience and continuous cutting-edge innovation that makes driving such a culture much safer and much more fun.

The way forward

Traditional OEMs are dreaming of a personalized and upgradeable product created around customer experiences. They aim to wow their customers and to keep their loyalty. They think about how to conquer new markets through a connected service ecosystem open to onboard multiple new stakeholders like telecom, insurance, banking, and healthcare. It seems attractive, but it requires an enormous reinvention, restructuring, and adoption of different technologies. This is because digital transformation isn't restricted to the installation of a digital platform, app, or tool. It involves using data and analytics-for-making customer-centric decisions and adopting a 'ONE Team' mindset. Above all, the E/E architecture must adapt to the requirements of the implemented functions and strategies on top.

It is highly recommended that OEMs start building partnerships, especially with tech companies, to learn how to structure their business around the customer and explore new business models with innovative prices and margins.

About the author

Houssemeddine Ghanmi

Houssem is an accomplished architect with over ten years of successfully designing, developing, and testing in-vehicle architectures, especially its security and network aspects. He has extensive experience in developing and designing safe, secure, scalable, and resilient in-vehicle onboard high-performance systems and software architectures while taking care of driver needs and regulatory demands. He has conceptualized various solutions in the ADAS domain for premium OEMs and suppliers and is seasoned in AUTOSAR, Security, A-SPICE, MISRA rules, etc.

We have helped leading automakers around the world start their 'connected cars' journey. We come with not just tech expertise but also an understanding of the rapidly evolving mobility needs of the customers. If you are keen to build futuristic mobility solutions, let us help you lay down the route. Interested? Talk to us at **paramita.nath@nagarro.com**

About Nagarro

Nagarro is a global digital engineering leader with a full-service offering, including digital product engineering, digital commerce, customer experience, AI and ML-based solutions, cloud, immersive technologies, IoT solutions, and consulting on next-generation ERP. We help our clients become innovative, digital-first companies through our entrepreneurial and agile mindset, and we deliver on our promise of thinking breakthroughs.

We have a broad and long-standing international customer base, primarily in Europe and North America. This includes many global blue-chip companies, leading independent software vendors (ISVs), other market and industry leaders, and public sector clients.

Today, we are over 16,000 experts across 30 countries, forming a Nation of Nagarrians, ready to help our customers succeed.