

Tech paper

Virtual development of in-vehicle infotainment (IVI) systems



Abstract

As software-defined vehicles (SDVs) become the new industry standard [1], the complexity of in-vehicle infotainment (IVI) systems continues to grow. Traditional hardware-dependent development approaches inherit long development cycles, increased costs, and late-stage integration challenges. Virtual development and virtualization technologies offer a faster, more flexible, and cost-efficient alternative to addressing these issues.

This paper explores how virtualization solutions enable end-to-end IVI system development in a fully virtualized environment. By leveraging virtualization, digital twins, and cloud-based development, OEMs and Tier 1 suppliers can accelerate prototyping, enable multi-OS integration (such as Linux, Safe POSIX, Android, Classic and Adaptive AUTOSAR stacks, etc.), and optimize software validation without physical hardware.

Key advantages of virtualizing IVI systems development include:

- **Accelerate:** Deliver production ready applications in less time with parallel workflows.
- **Quality:** Test earlier and at scale for better quality and increased test coverage.
- **Automate:** Continuous software delivery with CI/CD/CT and faster than real-time test execution.
- **Save:** Avoid or reduce expensive bench set ups, prototypes, HiL systems, and test vehicles.
- **Less risk:** De-risk development with hardware agnostic setups, less delays, and higher software maturity.

Using real-world use cases, this paper highlights how virtualization solutions empower automakers to create next-generation IVI systems that efficiently and securely accelerate innovation while ensuring seamless user experiences, compliance, and reliability.



Summary of tech paper contents

Abstract	2
The evolution of IVI in software-defined vehicles (SDVs)	4
Challenges in traditional IVI development	5
SDV development starts with virtual IVI	6
How Android-based virtual development is leveraging the next generation of IVI	7
Benefits	9
Accelerated development cycles, faster time to market	9
Improved collaboration	10
Hardware independence	11
Shift-left development and testing	12
Total cost reduction	13
Elektrobit is your partner for virtual SDV and IVI development	15
Case studies	16
Conclusion: Accelerating SDV transformation with virtual IVI development	17
About the authors	18
References	19

The evolution of IVI in software-defined vehicles (SDVs)

In-vehicle infotainment (IVI) systems have evolved rapidly from basic media and navigation units into complex, software-defined digital cockpits. Today's IVI systems serve as a user-facing hub for connectivity, personalization, safety features, and real-time system coordination. With the increasing adoption of centralized and software-driven architectures in vehicles, IVI systems are no longer standalone entities—they are integral to the entire software-defined vehicle (SDV) platform.

To place this evolution in context, we use the Eclipse SDV Level Framework [2], which defines the maturity levels for SDV platforms. When applied to IVI systems, these levels reflect the degree of software decoupling, updatability, and multi-domain integration:

- **SDV Levels 0–1:** Static IVI architectures with minimal software update paths, tightly coupled to hardware.
- **SDV Levels 2–3:** Support for over-the-air (OTA) updates, basic modularity in user space, and limited integration across IVI, ADAS, and body domains.
- **SDV Levels 4–5:** Full decoupling of hardware and software, centralized cockpit compute, integrated CI/CD delivery pipelines, and real-time orchestration across safety- and user-critical domains via virtualization, digital twins, and service-based architectures.

Achieving SDV Level 5 for IVI systems demands a complete paradigm shift: virtual-first development, containerized deployment, hypervisor-enabled domain separation, and continuous integration across RTC- and HPC-based components. Traditional IVI development models, which are deeply dependent on fixed-function hardware and siloed toolchains, cannot scale to meet these requirements. Their rigid architecture causes delays in feature iteration with complicated cross-domain integrations and inherent bottlenecks in testing and validation.

To keep pace, OEMs and Tier 1 suppliers need to adopt a fully virtualized, software-defined development lifecycle—one that leverages simulation, cloud-native collaboration, and standards-driven modularity to deliver scalable, future-proof IVI systems.

Challenges in traditional IVI development

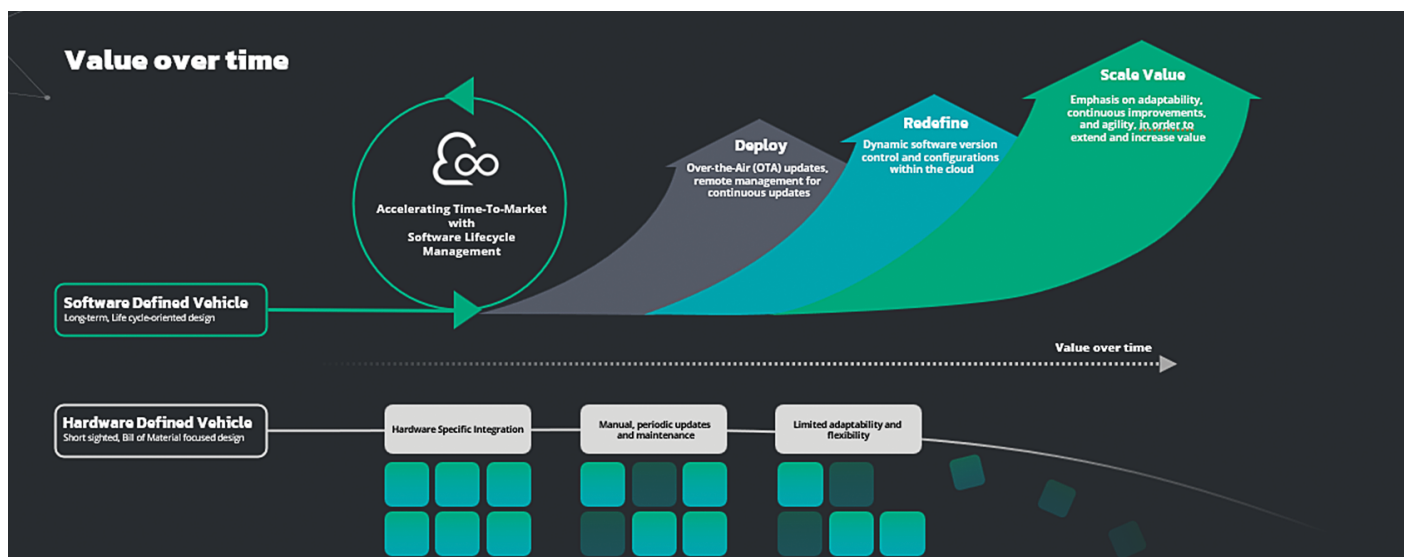
Despite the rapid advancements in IVI systems, development processes still face critical challenges that hinder innovation, escalate costs, and complicate scalability.

One of the persistent obstacles is **hardware dependency**. Traditional IVI development has relied heavily on physical hardware, which limits prototypes and feature development until production-ready electronic control units (ECUs) are available. Delays in hardware procurement create bottlenecks in development cycles, slowing down innovation and extending time to market. As vehicle architecture grows more complex, hardware integration and validation require extensive testing, which further prolongs development timelines.

Long development cycles and integration complexity also pose major obstacles. The journey from concept to production can span several years, meaning that a vehicle's deployed IVI system may already be outdated by the time it reaches the market. Integrating diverse operating system environments—such as Android Automotive OS, Linux, and QNX—requires substantial engineering effort, often resulting in compatibility issues and communication challenges. Additionally, the lack of standardized development and validation platforms makes it increasingly difficult to collaborate across teams and suppliers.

The **cost implications** of development and testing are another significant challenge. Running software validation on physical ECUs requires expensive hardware prototypes that are often limited in terms of their availability. Any hardware malfunction or last-stage specification changes can lead to expensive rework and project delays. Traditional testing methods that rely on physical vehicle prototypes further increase the financial burden and make validation slower thanks to resource-intensive processes.

These challenges highlight the need for a fundamental shift in IVI development methodologies—one that leverages **virtualization and cloud-native approaches** to break free from hardware limitations, compress development timelines, and significantly optimize cost and resource efficiency.



SDV development unlocks the power of hardware independence and connectivity, with value evolving over time.

SDV development starts with virtual IVI

While the broader transformation toward software-defined vehicles (SDVs) can appear overwhelming—requiring complex orchestration across vehicle domains, centralized compute, and real-time capabilities—there is a more practical path: **start with infotainment**.

IVI is the most accessible and cost-effective entry point to SDV. It already runs on high-performance application processors, supports modern operating systems like Android Automotive OS (AAOS) [5], and offers user-centric use cases that align with SDV principles such as over-the-air (OTA) updates, feature modularity, and cloud integration. **Over 90% [3] of consumer-visible features in a software-defined vehicle can be implemented through the IVI system [4]**, provided that the necessary vehicle data is abstracted and made securely accessible. As cockpit platforms evolve with centralized compute, containerized runtimes, and advanced middleware (e.g., VHAL, binder IPC, hypervisor abstraction), the IVI system becomes the natural delivery surface for most digital SDV functionality—from personalization to OTA services, ADAS visualization, and real-time driver feedback.

Rather than rearchitecting the entire electrical/electronic (E/E) architecture or rewriting legacy software stacks, automakers can generate quick SDV wins by:

- **Confining SDV scope to IVI-deliverable features:** Focus upgrades on cockpit experiences, personalization, AI-based voice agents, usage-based services, and contextual UI behaviors.
- **Reusing existing software components and SDKs:** Android Automotive provides a stable foundation with a mature developer ecosystem and extensive tooling (e.g., AIDL, Binder IPC, VHAL abstraction, and Jetpack libraries).
- **Leveraging virtualization to eliminate hardware bottlenecks:** Teams can use virtual ECUs (vECUs) and cloud-based CI/CD to develop, integrate, and validate new features in parallel without waiting for physical hardware or production-grade ECUs.

This virtual-first approach allows rapid prototyping, decoupled domain development, and faster validation—ideal for Android-based IVI systems. Developers can emulate hardware abstraction layers (HAL), simulate VHAL interfaces, and integrate OEM extensions (e.g., diagnostic agents, telemetry collectors, or AI inference engines) using containerized user-space applications. Virtual environments also make it easier to scale tests across multiple configurations (e.g., different trims, regions, or brands) without hardware respins.

Ultimately, IVI-driven SDV development allows OEMs to prove out SDV concepts in production vehicles **without destabilizing mission-critical domains** such as ADAS or powertrain. It enables real-world data capture, iterative software delivery, and fast feedback cycles using a domain that is already isolated from functional safety constraints.

In other words: **you do not need a Level 5 E/E architecture to offer a Level 5 SDV experience—if you design your IVI domain with the right tools and mindset.**

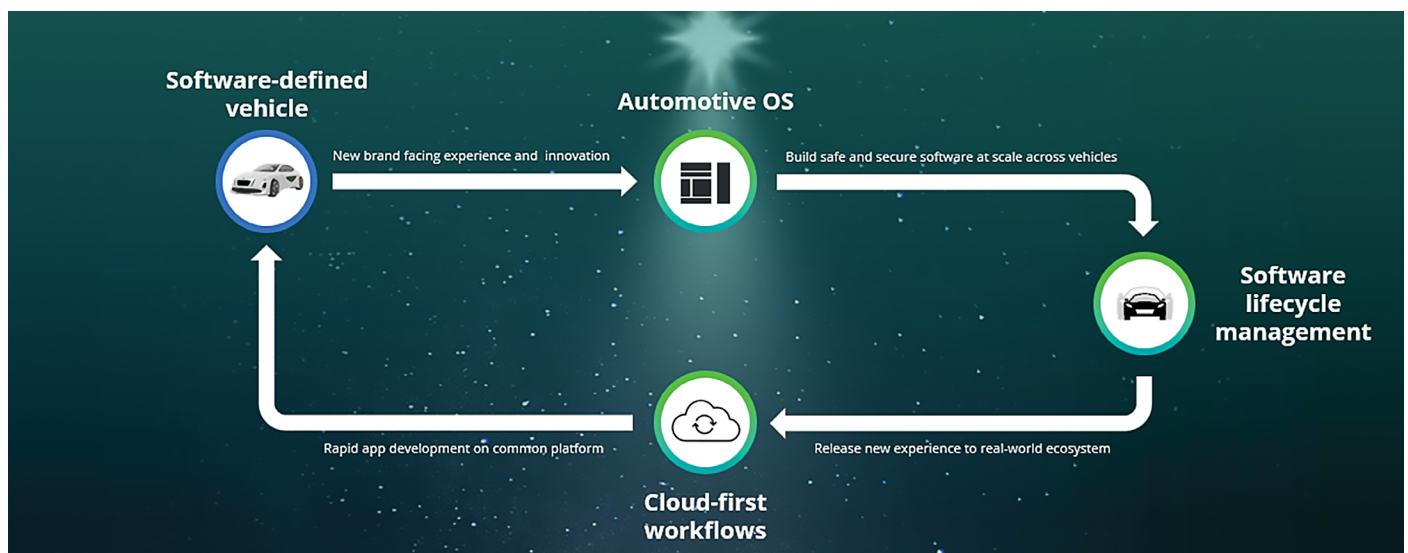
How Android-based virtual development is leveraging the next generation of IVI

Virtualization is a foundational enabler for the evolution of in-vehicle infotainment (IVI) systems toward true software-defined vehicle (SDV) maturity. By decoupling software development from physical hardware constraints, virtualization transforms traditional IVI workflows, delivering unprecedented flexibility, scalability, and operational efficiency.

At the core, vECUs and digital twin technologies allow engineering teams to prototype, test, and validate IVI software without waiting for production-grade hardware or physical test benches. This approach eliminates critical bottlenecks inherent in hardware-dependent development, enabling parallel workflows across distributed teams and accelerating overall deployment timelines.

The fastest and most cost-effective path for OEMs aiming to reach SDV Level 3 or higher without rearchitecting the entire electrical/electronic (E/E) architecture begins with Android. This approach offers immediate practical advantages:

- **Realistic Android HAL emulation** supporting the **VHAL generator** and Android Binder IPC communication mechanisms in virtual environments enables realistic interaction between Android apps and simulated vehicle hardware layers without physical devices.
- **Secure multi-domain orchestration.** Thanks to hypervisor-based partitioning and containerization, this approach allows safe coexistence of Android-based infotainment, real-time Linux systems, and Safe POSIX workloads—crucial for integrating mixed-criticality functions such as infotainment, ADAS, and powertrain control.
- Enabling **multi-domain orchestration** where real-time systems and user-space applications seamlessly coexist and dynamically coordinate data and control flows is key to advancing from siloed IVI systems to integrated SDV cockpits.



Key success factors of software-defined vehicles. Accelerate code to road with an Automotive OS and cloud-first workflows

Cloud-based, virtualized IVI development platforms extend these capabilities by providing globally accessible, synchronized workspaces. They enable real-time software deployment, debugging, automated regression testing, and continuous integration/continuous deployment (CI/CD) pipelines that accelerate feedback loops while improving software quality and maintainability. Beyond pure development agility, virtualization significantly reduces costs by minimizing reliance on expensive hardware prototypes and physical testing infrastructure. Early-stage validation in virtualized environments shortens time to market and keeps pace with consumer demands for frequent over-the-air (OTA) updates and rapidly evolving feature sets.

Virtualization supports multi-OS validation—including **Android Automotive OS (AAOS) [5], Android Open-Source Project (AOSP), Linux, and Safe POSIX-compliant, real-time environments**—ensuring compatibility and performance across diverse runtime platforms prior to deployment. This multi-platform flexibility futureproofs infotainment development amid evolving vehicle architectures and industry standards such as AUTOSAR and SOAFEE [6].



AAOS Development Environment



No-Code Customization



Integration with CI/CD/CT



Virtual & Cloud Targets



Simulation Tools

Virtualization is not just about optimizing workflows for infotainment, it enables a complete shift in how vehicle software is imagined, built, and delivered. By starting with Elektrobit's Android-based Virtual IVI solution, OEMs and Tier 1s benefit from:

- True hardware/software decoupling
- Real-time, cloud-enabled collaboration
- Secure, scalable cross-domain orchestration
- Reduced development costs and cycle times

In summary, if you want to build an SDV, start with virtual IVI. If you want to build it fast, use Android.

Benefits

Accelerated development cycles, faster time to market

Virtualization significantly accelerates development cycles and reduces time to market—making it a cornerstone of modern SDV development strategies, particularly in the IVI domain. By enabling **hardware-independent development**, virtual platforms allow teams to rapidly prototype, test, and iterate software across domains such as infotainment, connectivity, and ADAS—all in parallel.

Virtualization offers huge advantages, especially when all components are cleverly orchestrated. An ideal model of an IVI system based on Android Automotive enables:

- **Use of standard Android tools:** Developers can continue using familiar Android SDKs and toolchains without requiring cockpit-specific proprietary APIs.
- **Access to vehicle data via Vehicle Signal Specification (VSS) from COVESA [10]:** Data is abstracted and exposed via the vehicle hardware abstraction layer (VHAL) and uniform resource identifier (URI)-flattened VSS structures, allowing standardized, scalable access to vehicle signals in development and testing.
- **VHAL generator and test tooling:** Tools can automatically generate virtual HAL layers, feed them with test data using SOME/IP or synthetic inputs, and validate app behavior without need for a physical vehicle.
- **Screen and user interface (UI) abstraction:** Designers can configure Android screen topologies, theme apps, and simulate cockpit UI variations—system-wide. Integrated design, tooling, and theming engines with graphical user interfaces (GUI) emancipate design from coding.
- **Self-service and cloud instantiation:** Virtual cockpit environments can be instantiated on-demand as Amazon Machine Images (AMIs) or deployed in customer-managed virtual workbenches. This allows development teams to scale across global cloud or on-premises infrastructure—without vendor lock-in.



These capabilities reduce bottlenecks associated with hardware availability, regional development silos, and manual integration steps. Virtualization **enables continuous CI/CD** for IVI software, driving shorter validation cycles and more rapid feature delivery.

As a result, automakers and Tier 1s benefit from:

- **Shorter release cycles** and predictable delivery timelines
- **Lower development and testing costs**
- **Higher reuse of software components across vehicle programs**
- Ability to deliver frequent **over-the-air (OTA)** updates aligned with consumer expectations.

In an industry driven by software velocity, virtualization is not just a development convenience—it is a competitive necessity enabling the transition from monolithic, hardware-bound delivery models to agile, scalable, and cloud-native SDV development.

Improved collaboration

Real-time and worldwide

Virtualized development environments allow engineering teams to collaborate in real time—regardless of geographic location or availability of hardware. Android-based IVI systems can be instantiated in the cloud, allowing developers, UX designers, and systems engineers to:

- Test applications with simulated vehicle data via **VSS and VHAL**
- Validate screen behavior and cockpit UI through configurable layouts
- Apply OTA updates and run continuous integration pipelines entirely in the cloud

This model reduces the need for physical test benches and shortens feedback loops across distributed development cycles.

Build your custom Android Image with VSS vendor partition

data with COVEVA VSS Signals and Android VHAL properties to link your Android app to simulate vehicle signals. simultaneously to feed data into your emulators.

Custom Android Image Generator

ID	Name	Type	Access	Change mode	Min sample rate	Max sample rate	Initial value	Read permission	Write Permission
0u0005	VehicleAccelerationLongitudinal	float	read	continuous				1	Select
0u0006	VehicleAccelerationLateral	float	read	continuous				1	Select
0u0007	VehicleAngularVelocityYaw	float	read	continuous				2	Select
0u0008	VehicleAngularVelocityPitch	float	read	continuous				2	Select

Launch vanilla or custom images

ID	Status	Started By	Image	Last Updated
c17d2b20-2b0b-4b1a-bf19-d3b03160957	SETTING UP	admin	Android 14 - EB Skiffish Fleet Mgmt and Navigation	2025-05-07 09:35:52

Available Android Images

Android 12 - EB Skiffish

SKIFFISH-ANDROID_31_0-2024-12-16-23-34-skiffish_car_of_userdebug_flashimage.zip

Created at Wed 05 February 2025 at 16:47
Last updated at Fri 07 February 2025 at 13:34

Details

Android 14 - EB Skiffish

SKIFFISH-ANDROID_34_0-2025-01-22-23-35-skiffish_car_of_userdebug_flashimage.zip

Created at Wed 05 February 2025 at 16:55
Last updated at Fri 07 February 2025 at 13:33

Details

Android 15 - Vanilla

release_15.0.0_rustfish.zip

Created at Wed 05 February 2025 at 16:56
Last updated at Wed 05 February 2025 at 16:56

Details

Android 14 - EB Skiffish Fleet Mgmt and Navigation

SKIFFISH-ANDROID_34_0_CES-VHAL-Gem-Edition-2025-01-08-skiffish_car_of_user_flashimage.zip

Created at Wed 05 February 2025 at 16:56
Last updated at Wed 05 February 2025 at 16:56

Details

Ecosystem-aligned tooling

By exposing vehicle data through standardized interfaces (e.g., VSS over VHAL) and relying solely on Android APIs, the platform empowers developers to work with familiar toolchains and avoid cockpit-specific variations. The approach supports Linux, AOSP, and **AUTOSAR Classic or Adaptive** backends, providing flexibility across different EE architectures.

Develop & debug and stimulate app via the VHAL

Generate SOME/IP connection for the VHAL for real test data

Android Vehicle Signal Mapping

ID	Name	Type	Access	Change mode	Min sample rate	Max sample rate	ES conbox Link attribute
0u0000	PERF_VELOCITY_SPEED_M	float	read_write	continuous	1.0	10.0	Select
0u0001	ABS_ACTIVE_W	boolean	read_write	on_change			Select
0u0002	ENGINE_RPM_W	float	read_write	continuous	1.0	10.0	Select
0u0003	ENGINE_COOLANT_TEMP_W	float	read_write	continuous	1.0	10.0	Select
0u0004	EV_BATTERY_LEVEL_W	float	read_write	continuous	0.0	100.0	Select
0u0005	SEAT_BELT_BUCKLED_W	boolean	read_write	on_change			Select

Interoperability between **AUTOSAR components**, **hypervisor-based mixed-criticality partitioning**, and **Linux-based user-space environments** become essential within this broader SDV software ecosystem. When cohesively orchestrated, these elements allow IVI systems to operate securely alongside safety-relevant domains without requiring a full E/E architecture overhaul. Although IVI itself typically runs in isolated, user-space domains, adjacent enablers (such as **cloud-based software distribution**, **system orchestration platforms**, and **cross-domain middleware**) play a vital role in achieving scalable, maintainable, and service-oriented in-vehicle experiences.

Concurrent development streams

Virtualization also unlocks true parallel development. IVI teams can work independently on:

- UI/UX theming and composition without touching application code,
- Middleware integration and Android configuration via a **configurable runtime**,
- Feature testing with synthetic signal stimulation and real-time emulation.

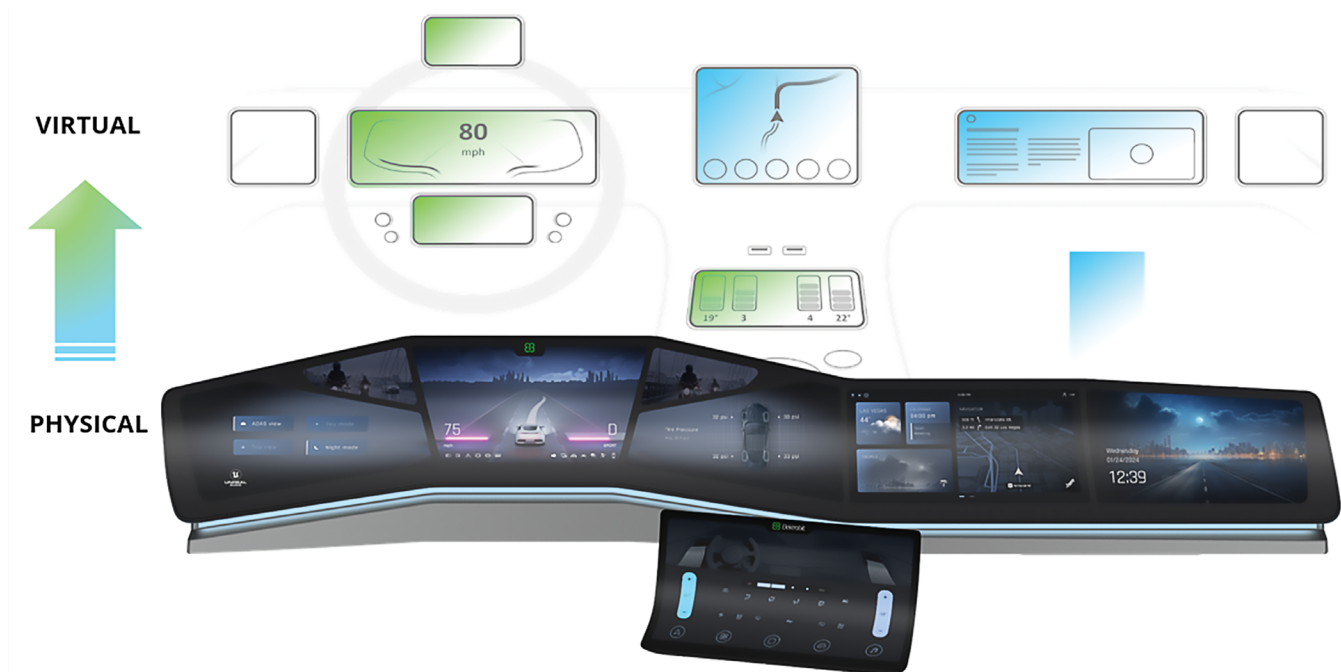
Meanwhile, safety-critical domains (e.g., ADAS or powertrain) developed in AUTOSAR environments can be validated alongside infotainment through a shared orchestration layer—enabled by an integrated software platform.

By aligning infotainment development with the larger SDV journey, this approach encourages a modular, cooperative engineering model that is scalable from concept to SOP.

Virtualized IVI development does not exist in isolation, it thrives within an orchestrated ecosystem that bridges Android innovation, real-time system integrity, and cloud-scale DevOps. This ecosystem—supported by robust middleware, containerization strategies, and standards-driven development environments—offers a blueprint for how automakers can move fast, stay modular, and scale efficiently toward SDV Level 5.

Hardware independence

Virtualization decouples IVI software development from physical hardware availability, enabling a more agile, resilient, and scalable development lifecycle. This independence not only mitigates supply chain and prototyping delays, but also unlocks key opportunities for accelerated innovation.



Early IVI prototyping and validation

With vECUs and containerized IVI stacks, developers can begin designing, integrating, and validating infotainment features well before production hardware is available. This shift significantly reduces project risk and shortens time-to-feature deployment.

Elimination of hardware bottlenecks

Virtual platforms remove dependency on scarce development ECUs or physical test benches. Teams can press on regardless of hardware readiness, using virtual environments that simulate real-world conditions with high fidelity. This is especially valuable in fast-moving IVI domains where frequent updates and agile iteration are expected.

Flexible testing across configurations

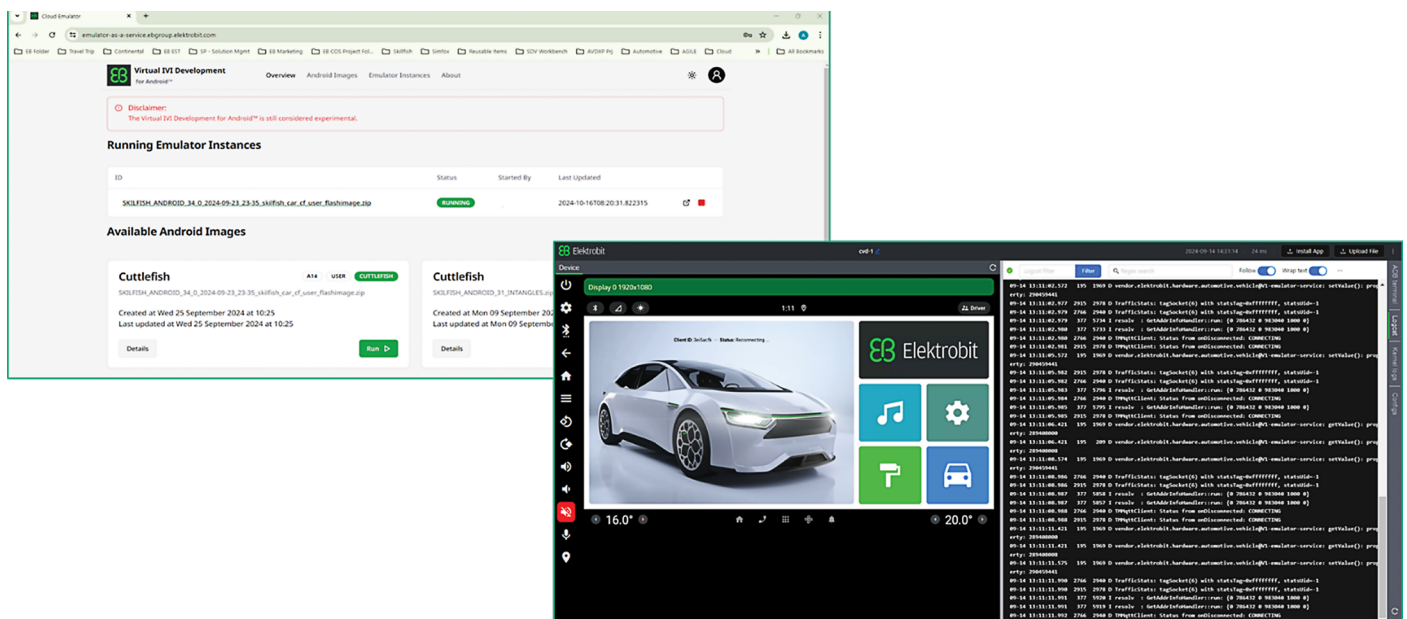
Developers can use cloud-based virtual environments for rapid testing across multiple OS variants—such as Android Automotive OS, AOSP, or embedded Linux—and target configurations, without having to reimage hardware setups. This supports concurrent testing of HMI layers, middleware integration, and API behavior across variants and domains.

Scalable integration with SDV architectures

When combined with solutions for hypervisor-based separation, service orchestration, and AUTOSAR-compliant modules, virtual IVI development offers seamless scaling within the broader SDV architecture. It also enables early integration with backend systems such as OTA update infrastructure, diagnostics, and telemetry, fostering a fully software-driven cockpit experience.

Shift-left development and testing

In the context of software-defined vehicles (SDVs), adopting a (virtualization-enabled) shift-left approach to IVI development empowers teams to identify issues earlier, improve software quality, and accelerate delivery cycles. Combined with integrated toolchains, vECUs allow infotainment features to be developed and validated in parallel with other vehicle domains.



Early detection of integration and runtime issues

Virtualized IVI environments allow engineers to test application logic, middleware services, and system behavior far ahead of physical integration phases. Bugs, regressions, and misconfigurations can be caught and resolved at an earlier point—when they are cheaper and easier to fix.

Continuous validation through CI/CD

Integrating vECUs into cloud-based CI/CD pipelines means that every code commit can trigger automated testing, from UI responsiveness to middleware compliance and inter-process communication. This ensures that infotainment software remains robust as it evolves and scales. Tools such as EB corbos and EB tresos support these loops through configuration validation and runtime integration readiness across AUTOSAR and Linux environments.

Proactive system simulation and tuning

Simulating full-stack behavior—across UI frameworks, Android services, and backend connectivity—gives developers an early understanding of performance characteristics. System loads, startup times, and data flow patterns can be profiled and optimized before they reach real hardware. This is particularly valuable in IVI, where user experience and responsiveness are tightly coupled with perceived quality.

Enabling smarter IVI integration in SDV platforms

Shift-left testing also supports broader SDV goals by facilitating earlier integration with vehicle-wide orchestration layers, diagnostics, and OTA frameworks. Infotainment systems can be tuned for both standalone excellence and ecosystem alignment thanks to cross-domain testing and API validation, enabled by modular toolchains and standardized interfaces.

Total cost reduction



Adopting virtualization for IVI development not only accelerates timelines but also delivers substantial cost benefits across the vehicle software lifecycle. By decoupling software from hardware, OEMs and Tier 1s can streamline development processes and optimize resource allocation without compromising on quality or innovation.

Reduced dependency on costly prototypes

vECUs and digital twins eliminate the need for extensive physical test benches and prototype ECUs during early development stages. For IVI systems, this means that UI behavior, Android services, and middleware logic can be developed and validated in parallel—long before cockpit hardware is available—leading to significant capital expenditure savings.

Lower integration risk and rework costs

By enabling early-stage fault detection, dependency mapping, and real-time software validation, virtualization minimizes the likelihood of late-phase surprises. Integration efforts are smoother and less reactive, dramatically reducing the hidden costs of rework, schedule slips, and feature rollback during SOP-critical phases.

Higher engineering efficiency

Cloud-based environments empower distributed teams to reuse test scenarios, SDKs, and software modules across multiple programs and platforms. Standardized toolchains, such as EB tresos for embedded configuration and EB corbos for runtime integration, enable modular reuse and reduce duplication, ensuring greater return on engineering investment.

Scalable cost structure

Virtualized IVI development allows for more efficient scaling across vehicle platforms and variants. Instead of duplicating hardware for each configuration, manufacturers can simulate multiple SKUs in parallel. When combined with orchestration tools and connectivity frameworks, such as EB corbos Link, the result is a sustainable, cost-effective SDV development model that is adaptable over time.

Elektrobit is your partner for virtual SDV and IVI development

Elektrobit combines deep software expertise with proven platform competence to drive next-generation infotainment and software-defined vehicle (SDV) innovation. From early-stage design to final deployment, we offer a comprehensive, virtualized development environment tailored to today's dynamic cockpit and vehicle architectures.

With decades of experience in automotive software and a trusted track record with leading OEMs and Tier 1 suppliers, Elektrobit delivers tailored solutions that enable agility, compliance, and competitive advantage across the full IVI lifecycle.

Accelerate cockpit development with our virtualized SDV platform [7]

Our cloud-native, hardware-independent development stack enables fast iterations, parallel engineering, and early-stage validation, helping customers bring IVI features to market faster without waiting on physical ECUs or test benches.

Tailored pipelines, seamless integration

We align with your program needs by customizing simulation environments, automated validation flows, and CI/CD pipelines, ensuring consistency with vehicle targets, SOP constraints, and in-vehicle experience design. Whether adapting UX flows, Android frameworks, or middleware configurations, we deliver without compromise.

Enabling continuous innovation

Our platform is designed to support real-time OTA deployment, digital twin synchronization, and continuous software delivery. This allows developers to maintain rapid, iterative development cycles that keep pace with evolving user expectations and technological shifts.

Interoperability across ecosystems

Our tooling supports multi-OS development—including Android Automotive OS, AOSP, Linux, and **Safe POSIX** platforms—allowing flexible integration across mixed-criticality domains. When paired with Elektrobit's broader ecosystem, such as **EB tresos** for embedded stack configuration, **EB corbos** for scalable runtime environments, and **EB corbos Link** for connected service integration, you benefit from a cohesive, modular SDV development flow built for long-term success.

Case studies

Sonatus

Sonatus leverages Elektrobit's virtualized IVI platform to showcase a cloud-native, over-the-air (OTA) update pipeline. In a combined demonstration, the Sonatus Updater delivers dynamic in-vehicle configuration updates via a live cloud deployment, fully validated within Elektrobit's virtual development environment. This highlights the power of virtualization in streamlining OTA delivery, which eliminates hardware bottlenecks and accelerates deployment of SDV-native data and automation solutions.

MOTER Technologies (MOTER)

MOTER accelerates the development of data-driven automotive insurance products by integrating virtualized IVI systems into its development workflow. Using Elektrobit's cockpit emulation environment, MOTER rapidly prototypes connected insurance experiences directly within the digital cockpit. This enables fast iteration, seamless data integration, and accelerated time-to market while ensuring alignment with evolving vehicle data standards and privacy regulations.

Intangles

Intangles develops AI-powered predictive maintenance and diagnostic solutions integrated with Elektrobit's hardware-independent IVI platform. By enabling smart prognostics, continuous health monitoring, and actionable insights, Intangles empowers OEMs and end-users with a resilient, real-time solution. This digital cockpit supports on-premise inferencing, delivering a seamless customer experience through proactive vehicle health management.

Here Technologies

Here Technologies collaborates with Elektrobit [8] to deliver immersive, location-aware navigation solutions into infotainment systems through a fully cloud-native development pipeline. Elektrobit's virtual IVI platform serves as a high-fidelity, hardware-agnostic cockpit emulator, enabling real-time visualization, rapid UI customization, and seamless integration of Here's advanced location intelligence.

Amazon Web Services (AWS)

Elektrobit launched a general availability of **Virtual IVI Development for Android™** [9] as an Amazon Machine AMI hosted on AWS's virtual SDV workbench. This offering democratizes Android IVI development by providing a fully virtualized, cloud-native cockpit environment that spans early prototyping through to deployment—without the need for physical hardware.

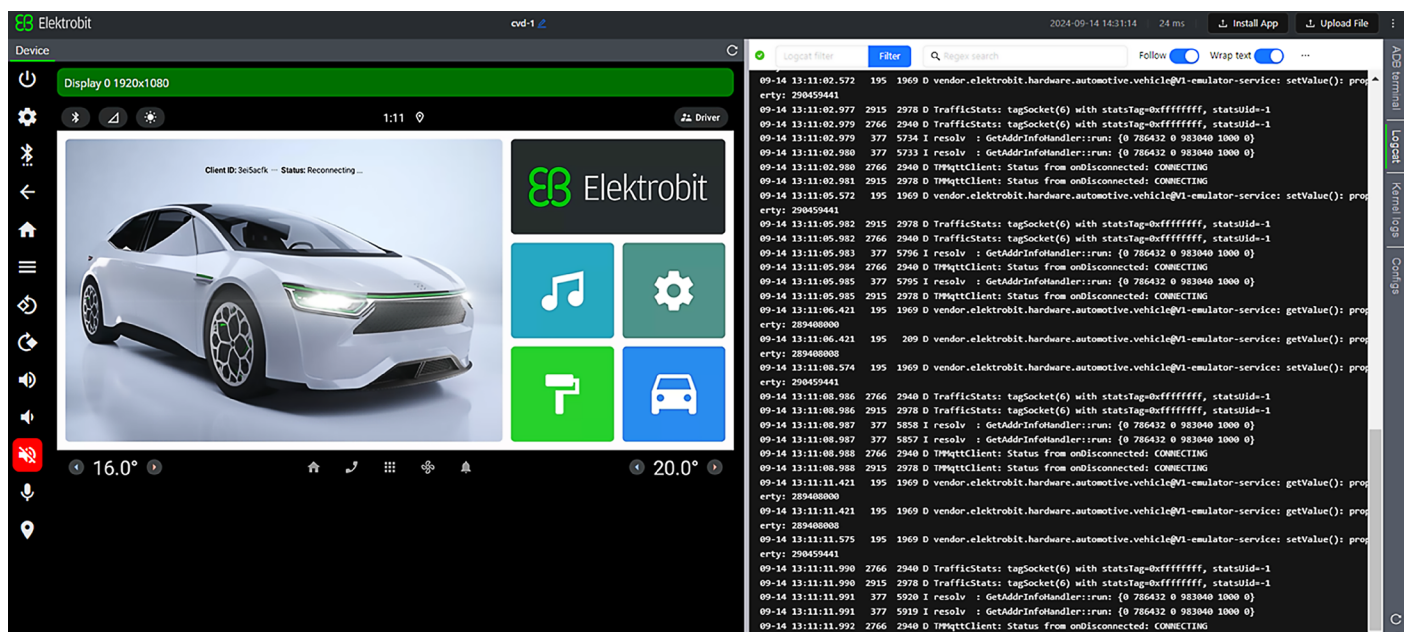
Developers can instantiate virtual cockpit targets in their private AWS environments to design, test, and debug Android-based infotainment systems from anywhere, at any time. This hardware-independent solution functions as a digital twin of the IVI system, empowering automotive teams to scale experimentation, streamline testing, and reduce time to market, all while leveraging AWS's elastic infrastructure for global scalability.

More case studies are available upon request.

Conclusion: Accelerating SDV transformation with virtual IVI development

The path to software-defined vehicles does not have to start with full E/E re-architecture. As demonstrated, **infotainment offers the most accessible and cost-efficient entry point** to SDV transformation. By leveraging Android, vECUs, and a mature ecosystem of developer tools and platforms, automakers can deliver high-impact user experiences—rapidly and at scale.

Virtualization is not just a technical enabler, it is a strategic imperative. From earlier fault detection and reduced prototype costs to faster OTA delivery and scalable collaboration, virtual development unlocks new speed, flexibility, and resilience in IVI innovation.



Elektrobit's encompassing expertise in embedded software, combined with a powerful SDV toolchain (including solutions like [EB tresos](#), [EB corbos](#), and [EB corbos Link](#)) enables seamless integration across AUTOSAR, Linux, and Android-based architectures. Our virtual IVI platform is production-ready, cloud-native, and proven across real-world customer deployments.

The automotive cockpit is evolving from static displays to dynamic digital ecosystems. By embracing virtualization, automakers can reimagine infotainment as a cornerstone of the SDV era: future-ready, developer-friendly, and user-focused.

About the authors



Ayhan Ak

Ayhan is Product Solution Manager at Elektrobit Automotive GmbH.

Having previously held leadership roles at global companies in the automotive software and consumer products sectors, and with over 17 years of professional experience in connected products and services across a range of industries, he joined Elektrobit in 2022 as Technical Product Manager in Service Engineering.

He is passionate about shaping products that connect technology with customer value through product innovation in a connected world. As a strategic product leader, he drives portfolio management and cross-functional initiatives that transform business goals into leading solutions.



Raúl Latorre Fortes

Raúl is Director of Business Development for EMEA at Elektrobit.

He has over 20 years of experience in the automotive and energy sectors. Since joining Elektrobit in 2020, he has worked in both Program Management and Business Development roles.

Prior to Elektrobit, Raúl held leadership positions at Framatome, Areva, and various automotive companies, focusing on connecting technology development with market growth.

Passionate about strategy and innovation, he enables business growth and efficient collaboration.

References

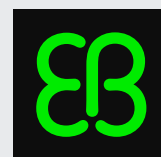
- [1] Gassmann, O., Böhm, J., & Palmié, M. (2021). *Software-defined vehicles: An emerging paradigm in automotive development*. In *Future Automotive Design* (pp. 221–235). Springer.
- [2] Eclipse Foundation, “Eclipse SDV-LVL (Eclipse Software-Defined Vehicle Levels of Value),” 2024. [Online]. Available: <https://projects.eclipse.org/proposals/eclipse-sdv-lvl>
- [3] McKinsey & Company. (2023). *The software-defined vehicle: Unlocking the full potential of automotive software*. McKinsey & Company.
Retrieved from <https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/the-software-defined-vehicle>
- [4] [Automotive | Android Open Source Project](#); Android Automotive OS whitepapers describe the IVI as the primary surface for user engagement and updates
- [5] Google. (2023). *Android Automotive OS Overview*.
Retrieved from <https://source.android.com/devices/automotive>
- [6] SOAFEE Special Interest Group. (2022). *Scalable Open Architecture for Embedded Edge (SOAFEE) Whitepaper v1.1*.
Retrieved from https://www.soafEE.io/wp-content/uploads/2022/07/SOAFEE_Architecture_Whitepaper.pdf
- [7] Elektrobit. (2024). *Virtual IVI Development for Android™ on AWS: Technical Overview*. Elektrobit.
Retrieved from <https://www.elektrobit.com/products/automotive-software/virtual-ivi-android-aws/>
- [8] HERE Technologies & Elektrobit. (2024). *High-Fidelity Navigation and UX Prototyping Using Virtual IVI Cockpits*.
Retrieved from <https://www.here.com/partners/elektrobit>
- [9] Amazon Web Services. (2024). *Developing Automotive IVI Systems in the Cloud Using Virtual ECUs and Android Automotive*.
Retrieved from <https://aws.amazon.com/marketplace/pp/prodview-virtual-ivi-android>
- [10] Vehicle Signal Specification (VSS). Connected Vehicle Systems Alliance (COVESA). Retrieved May 26, 2025, from <https://covesa.global/vehicle-signal-specification>



About Elektrobit

Elektrobit is an award-winning and visionary global vendor of embedded and connected software products and services for the automotive industry. A leader in automotive software with over 35 years of serving the industry, Elektrobit's software powers over five billion devices in more than 600 million vehicles and offers flexible, innovative solutions for car infrastructure software, connectivity & security, automated driving and related tools, and user experience. Elektrobit is a wholly-owned, independently-operated subsidiary of Continental.

For more information, visit us at elektrobit.com



Elektrobit Automotive GmbH
Am Wolfsmantel 46
91058 Erlangen, Germany

Phone: +49 9131 7701 0
Fax: +49 9131 7701 6333

sales@elektrobit.com

www.elektrobit.com