Received 17 June 2025, accepted 6 July 2025, date of publication 11 July 2025, date of current version 23 July 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3588432

TOPICAL REVIEW

Rethinking Vehicle Architecture Through Softwarization and Servitization

ALAA KHAMIS[1], (Senior Member, IEEE), AND PARTHA GOSWAMI[2]
[1]AI for Smart Mobility Laboratory, IRC for Smart Mobility and Logistics, KFUPM, Dhahran 31261, Saudi Arabia
[2]PG Mobility Analysis, LLC, Farmington Hills, MI 48336, USA

Corresponding author: Alaa Khamis (alaa.rashwan@kfupm.edu.sa)

This work was supported by Interdisciplinary Research Center for Smart Mobility and Logistics for Smart Mobility and Logistics at the King Fahd University of Petroleum and Minerals (KFUPM): Contextual Observability of Software-defined Vehicles under Grant INML2501.

ABSTRACT The automotive industry is undergoing a foundational transformation driven by software-defined vehicles (SDVs), where software not only orchestrates vehicle functionality but also redefines the value proposition through servitization, making services a key differentiator rather than the physical product. This paper explores the industry's shift from a hardware-centric paradigm to one defined by software and presents a comprehensive review of SDVs, structured around a systematic analysis of scientific publications and patents. It lays out the evolution from a distributed to a centralized compute architecture and discusses the promises of SDVs and their challenges. In particular, the paper highlights critical obstacles including fragmented certification frameworks, lack of standardized toolchains, limited support for continuous validation, and rising complexity in software maintenance and repair. Additional concerns include vulnerability management, long-term software support, supply chain decentralization, regulatory compliance, and the absence of open datasets, simulators, and benchmarking tools. The software-centric nature of SDVs also imposes new demands on hardware scalability and future-proofing. Finally, we highlight future directions and research opportunities such as ecosystem collaboration, open innovation, open and standardized datasets, open education simulators and testbeds, quantifiable Quality of Experience (QoX), cognitive data sampling, and contextual observability. This work offers a structured and forward-looking perspective on how SDVs are fundamentally reshaping automotive technology and business models toward intelligent, service-centric mobility ecosystems. To fully realize this potential, several technical, operational, regulatory, and organizational challenges must be addressed across the SDV lifecycle.

INDEX TERMS Automotive Softwarization, feature-as-a-Service (FaaS), over-the-air (OTA) updates, servitization, software-defined vehicles (SDVs), vehicle E/E architecture.

I. INTRODUCTION

Software-defined Vehicles (SDVs) represent a transformative shift in the automotive industry, moving away from traditional hardware-centric design toward vehicles in which software plays the central role in defining functionality, performance, and user experience (UX). This transformation mirrors trends in consumer electronics devices, which have evolved to become software-driven, cloud-connected platforms to support a host of new digital applications and services that can be upgraded over time. SDVs also mark the convergence of the automotive and cloud computing industries, combining real-time control with scalable, connected, and continuously updatable software platforms. Leveraging advances in computing power, connectivity, service-oriented architectures (SoA), and microservices, SDVs enable vehicle features and functions to be updated, enhanced, and personalized over time, often without requiring any physical modifications.

The concept of software-defined vehicles has emerged partly in response to the static and limited user experience of traditional vehicles, particularly when contrasted with the evolution of smartphones, which are designed as updatable platforms that support continuous enhancement.

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Messina.

Many vehicle owners report dissatisfaction with the lack of adaptability and limited features in conventional vehicles [1]. In parallel, the rise of connected and automated vehicles (CAVs) further underscores the need for a modern, scalable cloud-connected architecture that decouples software from the vehicle hardware.

It is notable that the software complexity of modern luxury vehicles now exceeds that of an advanced aircraft. For instance, while the F-22 Raptor fighter jet contains approximately 1.7 million lines of code (LoC) and the Boeing 787 Dreamliner includes around 6.5 million LoC, today's high-end vehicles may integrate over 100 million LoC [2]. These multi-million LoC are typically embedded in nearly 100 electronic control units (ECUs) distributed across the vehicle, each running one or more specific vehicle functions, including a sophisticated suite of sensor and actuator networks. This distributed architecture of a growing number of ECUs with embedded software that cannot be modified after vehicle purchase is no longer a viable framework to manage the complexity of future vehicles. In the new SDV paradigm, all vehicle features and functions are envisioned as services driven by software, ideally running on a centralized compute platform rather than dedicated individual ECUs. This transformation, characterized by a modular architecture that decouples software from the underlying hardware, enables new opportunities for continuous innovation by over-the-air (OTA) modification or addition of domain-specific, cross-domain and ecosystem-based applications or services.

This paradigm shift toward software-centric, connected, and continuously updatable vehicles is not only redefining how vehicles are developed and maintained, but also how they are used and could be monetized. As SDVs evolve into programmable platforms, they open the door for new value creation models centered around services rather than just hardware. In this context, the concept of servitization becomes highly relevant. Servitization refers to the process by which original equipment manufacturers (OEMs) transition from selling products to offering integrated product–service systems (PSS), competing not solely through hardware but through value-added services [3]. It has emerged as a potential opportunity for OEMs to support future economic growth and product differentiation [4]. Servitization offers compelling value propositions by enabling market expansion, the establishment of long-term customer relationships, and multiple cost advantages [5]. It involves not only the addition of services to traditional offerings but, in some cases, the replacement of physical products with fully service-based solutions [6]. Table 1 illustrates how various servitization business models can be applied in the context of software-defined vehicles, highlighting the shift from product ownership to result-oriented service paradigms enabled by Feature-as-a-Service (FaaS).

The architectural evolution of SDVs significantly enhances OEMs' capacity to adapt to these emerging business models and evolving customer expectations. In the traditional model of services as a necessity, the customer primarily acquires the physical product with fixed feature sets, and there are very few additional transactions between the manufacturer and the customer, beyond occasional warranty support. As customer demands shift toward services as added value, SDVs empower OEMs to offer on-demand functionalities, personalized feature updates, potentially via OTA delivery. When advancing to the model of services as a business, the inherent flexibility of SDVs supports new business models such as subscription-based enhancements, usage-based pricing, and context-aware feature activation, allowing OEMs to dynamically monetize vehicle capabilities over time. Ultimately, in the most mature model, where services are envisioned as the core business, SDVs become enablers of fully outcome-oriented mobility platforms, where not only features but entire vehicles are abstracted from ownership. Services are orchestrated entirely by software to fulfill user needs on demand. Thus, SDV architecture not only supports but fundamentally enables this progression by decoupling software from hardware and allowing continuous, flexible,

TABLE 1. Servitization business model evolution in the automotive sector.

Business Model	Customer Mindsets	Service Value	Examples
Service only as a necessity	Prefers full ownership	Post sale reactive maintenance and repair services	- Purchases a vehicle with fixed features - Visits dealership for repairs and scheduled maintenance - No OTA updates; features are static
Value-added services	Open to additional features/ services post-purchase	Enhances ownership experience with digital services	- Purchases premium infotainment (e.g., Mercedes-Benz MBUX augmented reality navigation) or ADAS after initial sale (e.g., VW dynamic light assist, GM SuperCruise, Ford BlueCruise, or Tesla Autopilot/FSD) - Subscribes to telematics or remote diagnostics - Receives OTA software upgrades and patches
Service as a business	Values flexibility and convenience over ownership	Use-oriented	- Rents SDV with temporary access to premium features - Pay-per-use insurance or battery-as-a-service - Vehicle usage-based maintenance planning
Service as the core business	Doesn't specify the asset; specifies the need and pays for a result	Result-oriented	- Pays for outcome (e.g., trip from A to B), not the vehicle - Integrated with public transport, ride-hailing, and shared mobility platforms - Seamless digital access, personalization, and mobility credits

and user-centric service delivery throughout the vehicle lifecycle.

This paper provides a comprehensive review of SDVs, examining the current state of the industry and academic research, including technological advancements and roadblocks. It investigates the promises of SDVs, and the complexities of automotive softwarization. The paper also explores future research opportunities with a focus on the long-term implications of software in the automotive ecosystem. Through this review, the paper aims to contribute to the ongoing discourse on the future of SDVs and their impact on the automotive sector.

The remainder of this paper is organized as follows. Section II describe the three SDV generations followed by presenting a systematic review of SDVs in Section III, following the PRISMA methodology, with a focus on scientific publications and patents. Section IV explores the role of software in the automotive industry, discussing the promises and the challenges of automotive softwarization and how the industry is adjusting to the software-centric challenges. Potential directions and research opportunities are outlined in Section V. Finally, concluding remarks are summarized in Section VI.

II. EVOLUTION OF SDV ARCHITECTURES

Software-defined vehicles represent a fundamental transformation in both architecture and functionality. This transformation can be understood through four evolutionary stages: HDV/SDV 0.0, SDV 1.0, SDV 2.0, and SDV 3.0, as illustrated in Figure 1. Each stage reflects significant advances in the separation of hardware and software as well as in clustering and centralization of compute modules.

A. HDV/SDV 0.0: DISTRIBUTED ARCHITECTURE

The traditional automobile, often referred to as Hardware-defined Vehicles (HDVs), or SDV 0.0, comprises a distributed system of dedicated electronic control units (ECUs), with embedded software, connected to a vehicle bus. In the traditional automotive model, mechanical and electronic components operate independently, with a dedicated ECU for each function, with integrated or embedded software. A premium vehicle can potentially have over 100 such ECUs. The embedded software cannot easily be changed or updated. The vehicle relies on fixed-function hardware, leading to high development costs and long feature deployment cycles.

The motivation behind SDV architecture is to change this static system to a more dynamic and flexible one, by decoupling or separating the application software, operating system (OS), and other middleware from the underlying hardware and consolidating them into a more manageable number of ECUs. The resulting architecture can be better managed and modified, usually remotely, by OTA software updates. The first generation of Rivian electric pickups and SUVs, for example, consolidated the compute hardware from 80 to 17 ECUs. The second generation further clustered them from 17 to only 7 ECUs.

B. SDV 1.0: DOMAIN ARCHITECTURE

SDV 1.0 introduces a domain-based approach, where multiple functions within a subsystem (e.g., infotainment, connectivity, body, powertrain, vehicle control) are consolidated into Domain Control Units (DCUs), which manage a specific domain or set of vehicle functions, such as advanced driver assist system (ADAS) or chassis control functions. This structure offers some degree of software-hardware decoupling, enables OTA updates and results in a cleaner system layout. However, architectural silos persist, limiting the overall scalability and reusability of software components [7].

C. SDV 2.0: ZONAL ARCHITECTURE

SDV 2.0 shifts further toward modularity and abstraction. Zonal architectures distribute low-level sensor and actuator management to zone controllers, while a centralized compute platform coordinates high-level functionalities via a high-bandwidth communication network. This enables more efficient development and deployment, reduces wiring complexity, and enhances system scalability and updatability [8], [9]. It enhances hardware consolidation and further facilitates real-time functional upgrades, augmenting feature-as-a-service (FaaS) functionalities.

D. SDV 3.0: CONSOLIDATED COMPUTE ARCHITECTURE

SDV 3.0, represents further consolidation of the compute architecture with Centralized High-Performance Computers (HPCs) running virtualized software workloads across distributed zone gateways, enabled by high-speed Ethernet communication. This model maximizes the decoupling of software from hardware and supports real-time orchestration. [10]. While enabling tremendous flexibility and innovation, the fully centralized architecture also introduces new challenges in software lifecycle management, network complexity, safety validation, and cybersecurity.

Across all generations, the key characteristic and enabler of SDV evolution is the progressive decoupling of hardware and software. In SDV 2.0 and SDV 3.0, the architecture supports a full stack: functional hardware (sensors and actuators), E/E architecture for data exchange, compute platforms, OS kernels (e.g., Linux, QNX), middleware for unified messaging, and a service-oriented architecture (SoA) layer [7]. On top of this stack, application software–ranging from safety-critical functions to infotainment–operates independently from the underlying hardware. All three models, SDV 1.0/2.0/3.0, allow dynamic software development processes such as continuous integration and deployment (CI/CD), rapid feature iteration, and customer personalization through cloud-based services [8]. While SDV 2.0 enhances modularity by distributing computational tasks to zonal units and leveraging a central compute platform for complex, cross-zonal tasks, SDV 3.0 fully centralizes all compute tasks to optimize resource efficiency, though this introduces challenges such as the need for a uniform high-speed software-defined network

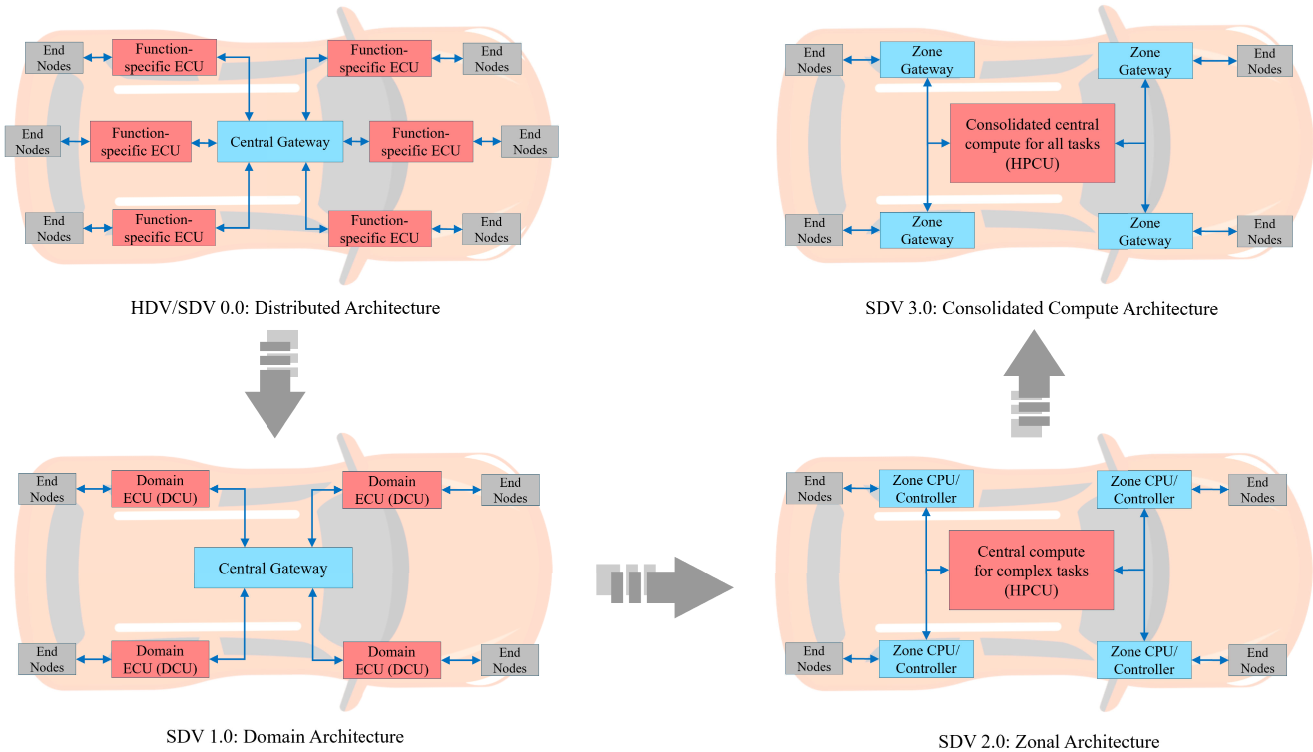

FIGURE 1. SDV Generations.

(SDN) and the risk of a single point of failure. Table 2 outlines the differences between HDV/SDV 0.0, SDV 1.0, SDV 2.0, and SDV 3.0 across key architectural and operational factors.

III. SYSTEMATIC REVIEW

Recent research on SDVs spans a broad range of technical and organizational challenges. Many works propose novel SDV architectures and frameworks, such as centralized computing platforms, zonal architectures, and virtualization-based systems [8], [11], [12], [13], [14], [15]. Key contributions also explore virtualization and containerization technologies for efficient software deployment, performance evaluation, and microservice orchestration [9], [16], [17], [18]. Middleware innovations and service-oriented designs continue to play a central role in SDV evolution [19], [20]. Several papers discuss resource allocation for safety-critical and non-safety-critical SDV applications, safety, cybersecurity, and functional integrity through secure software update pipelines, diagnostics frameworks, mixed-criticality execution models, and public key infrastructure [21], [22], [23], [24], [25]. In addition, several efforts are presented to investigate the challenges of SDVs. Liu et al. explore the impact of SDVs on the automotive ecosystem, identifying barriers in adapting traditional development models to meet iterative demands [26]. Bordoloi et al. address the limitations of centralized architectures, such as increased timing uncertainties and complex worst-case execution time (WCET) estimation, proposing solutions to enhance efficiency [27]. Panchal and Wang highlight the shift to software-dominated architectures, emphasizing the need to decouple software from hardware and integrate cloud-connected ecosystems for agile development [28]. Moreover, Sureddi addressed the integration of agile methodologies into the automotive industry [29]. The benefits, challenges, and opportunities of agile methods in the automotive industry are explored in [30] and the Software embedded system integration in Cross-organizational Agile TeamS (SCATS) framework for cross–organizational software integration in SDVs is described in [12]. These works collectively underscore the necessity for agile methodologies, and robust design frameworks to ensure the safety, reliability, and scalability of SDVs.

A systematic review on SDVs was conducted using the Preferred Reporting Items for Systematic reviews and Meta-Analyses (PRISMA) framework [31] to ensure a transparent and structured process for identifying, screening, evaluating, and including relevant literature. The review involved a comprehensive search across academic and industry databases (e.g., Web of Science, Google Scholar, SAE Digital Library, Google Patents, WIPO), using targeted keywords to identify both peer-reviewed publications and patents related to SDV technologies. After removing duplicates and excluding studies primarily focused on autonomous driving or vehicular networking–such as self-driving vehicles or software-defined networks–the remaining documents underwent full-text eligibility assessment. To avoid confusion with nonstandard or conflicting acronyms (e.g., using SDVs to refer to self-driving vehicles), only the literature explicitly addressing

TABLE 2. Comparison of SDV Architectural Generations: HDV/SDV 0.0 to SDV 3.0.

Aspect	SDV 0.0: Distributed Architecture	SDV 1.0: Domain Architecture	SDV 2.0: Zonal Architecture	SDV 3.0: Consolidated Compute Architecture
Architecture Type	Hardware-defined vehicle with distributed function-specific ECUs	A smaller number of domain controllers manage specific functional domains (e.g., infotainment, powertrain, body)	Software-defined vehicle with central HPC compute and zone CPUs/controllers	Central HPC platform with zonal gateways
HW-SW Relationship	Full coupling between hardware (HW) and embedded software (SW)	Domain-level separation of SW from HW for abstraction and reuse	Extensive HW/SW decoupling; more reliance on central compute	Near-total decoupling; HPC abstracts nearly all application logic
Development Cost	Very high (custom ECUs per function)	High (complexity across domains)	Moderate (zonal standardization)	Lower recurring cost after initial HPC investment
Function Update Frequency	Rare (manual updates)	Infrequent (limited OTA)	Frequent (OTA + zonal modular updates)	Continuous (CI/CD pipelines)
Scalability	Very low (HW locked per model)	Low to moderate	High (standardized zones)	Very high (uniform compute base)
Feature Time to Market	Slow (hardware-driven)	Moderate (SW-enhanced)	Fast (abstraction and parallelism)	Very fast (cloud-native agile deployment)
Feature as a Service (FaaS)	Not supported	Emerging	Enabled	Fully enabled and monetizable
Digital Value Stream	None	Basic (via gateway)	Cloud-integrated	AI-driven, DevOps-aligned
Physical Value Stream	Manual production logic dominates	Still prominent	Reduced (modular packaging)	Software-first lifecycle dominates
Complexity	High (many ECUs, complex wiring)	Moderate (domain silos)	High (HW simplified, SW complex)	Very high (virtualization, orchestration layers)
Reliability	High (mature HW-based)	Moderate (increasing SW role)	Moderate (OEMs adapting)	Moderate (SW bugs in safety-critical paths)
User Maintenance Cost (0–10 yrs)	High (manual + replacements)	Moderate (some OTA help)	Lower (remote diagnostics + OTA)	Lowest (predictive, software-centric)
User Maintenance Cost (10+ yrs)	Easy (widespread legacy support)	Challenging (HW-SW mismatch)	Risk of obsolescence in updates	High HPC-software mismatch over time
Dealer Maintenance Cost	Low (predictable)	Moderate (diagnostics needed)	High (tech and remote tools)	Very high (SW ops + cybersecurity)
Connectivity Dependency	Very low (standalone functions)	Moderate (some OTA)	High (cloud/edge features)	Very high (connectivity critical)
Cybersecurity Risk	Low (isolated ECUs)	Moderate (networked domains)	High (expanded attack surface)	Very high (full-stack threat model)

Software-defined Vehicles was considered. In total, 171 scientific and technical publications and 101 patents published between 2019 and 2024 were included in the final analysis. This systematic review provides a comprehensive examination of SDV research trends.

Figure 2 shows the number of scientific and technical publications focused on SDV. From 2019 to 2024, the data shows a general upward trajectory in SDV-related publications, with a slow start (2019–2021), and a rapid acceleration (2022–2024). The compound growth is remarkable: from 2 publications in 2019 to 90 in 2024, with an average annual growth rate exceeding 300% between 2021 and 2024. The trend suggests that SDV research has gained significant traction over this period, likely driven by technological advancements, industry investment, or academic focus.

The collected data highlights a global but uneven distribution, with a handful of nations driving the majority of SDV research output. Academia leads SDV publications at about 46%, followed closely by industry at about 35%, with joint academia-industry efforts contributing a notable but smaller 17% as shown in Figure 3. The data reveals a research landscape where academia and industry are the primary

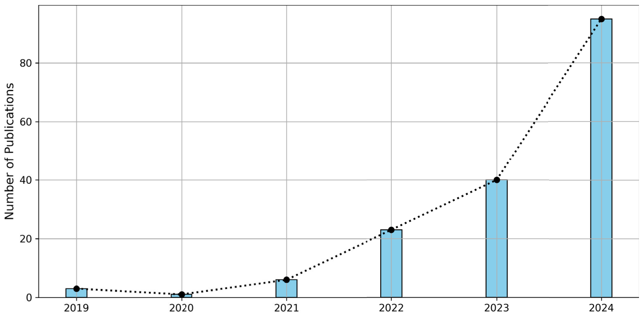

FIGURE 2. Trend of number of scientific/technical publications on SDVs.

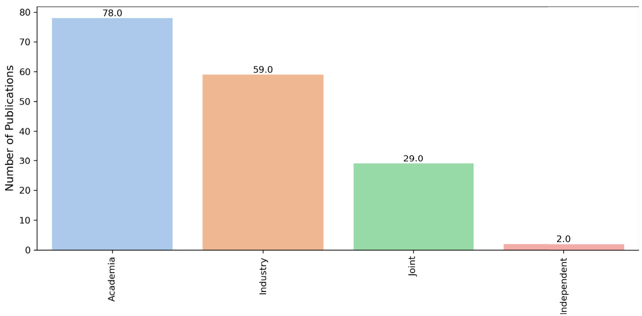

FIGURE 3. Academia, industry, and joint publications on SDVs.

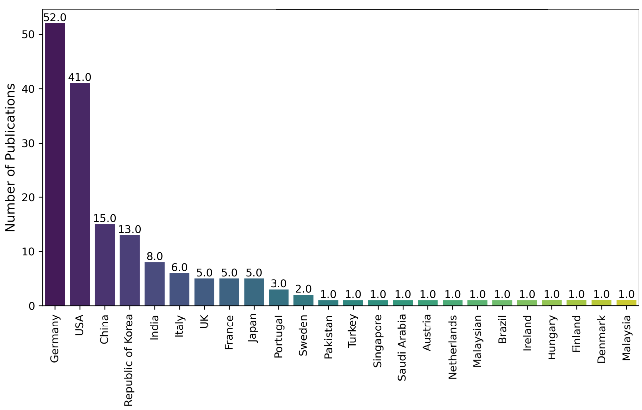

FIGURE 4. Countries with most scientific/technical publications on SDVs.

drivers, with academia slightly ahead, and collaboration playing a supporting but not dominant role.

Germany and the USA are the clear frontrunners in SDV scientific and technical publications, contributing about 17% and 13%, respectively, driven by their automotive and tech prowess as shown in Figure 4. China, South Korea, and India follow as emerging players, while the remaining countries, including traditional tech innovators like Japan and France, have smaller shares.

In terms of number of patents filed in the area of SDV, Figure 5 shows that number of SDV patents grew rapidly in 2020-2022 before declining in 2023-2024, illustrating a rise-and-fall trend. This trend reflects an initial boom in SDV patenting activity peaking in 2022 followed by a decline through 2024.

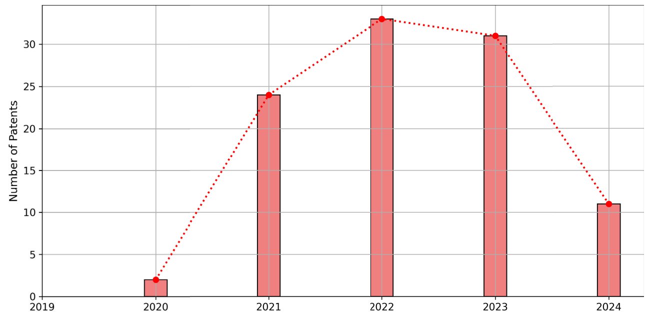

FIGURE 5. Trend of number of patents on SDVs.

The early increase (2020–2022) aligns with the rise of software-driven automotive solutions, where companies and inventors raced to secure intellectual property. The subsequent decrease (2023–2024) could suggest that foundational SDV technologies have been patented, leaving fewer novel ideas to claim, or that the industry has shifted toward the execution and refinement of existing innovations rather than generating new ones. Alternatively, external factors like economic conditions or regulatory changes might have influenced patent filings.

The SDV patent landscape is heavily industry-driven, with 92 patents originating from corporate entities, dwarfing the contributions of independent inventors (6 patents) and academia (3 patents) as shown in Figure 6.

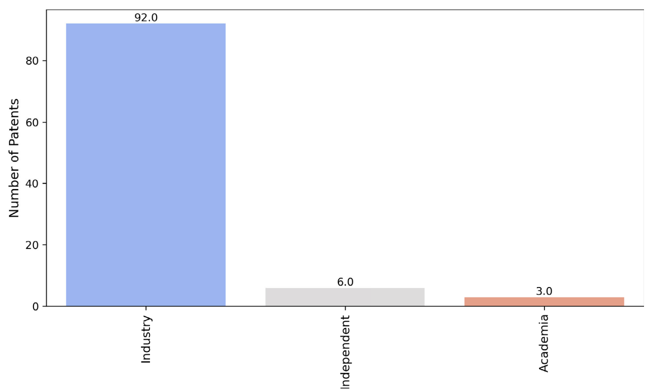

FIGURE 6. Academia, industry, and joint patent filings on SDVs.

This distribution underscores the commercial prioritization of SDV technologies, with industry leading the charge in intellectual property development. This also highlights the industry's focus on securing SDV intellectual property, while academia and independents play secondary roles in patenting, emphasizing a commercially driven patent ecosystem.

In terms of the market origin of patents, while a precise assessment of SDV-related patents globally is complicated and may have variation among the specific search platforms, the collected data indicates that China overwhelmingly leads with USA and EU as the second and third followers. Most of the Chinese patents appear to be in the category of "utility patents" as opposed to "design patents", indicating the emphasis on execution by the OEMs and rest of the industry eco-system. This highlights China's aggressive push to secure applied intellectual property in SDV technology. This is in contrast to the publication landscape, where US and Europe lead overwhelmingly. The findings indicate that SDV research is expanding rapidly, with industry prioritizing intellectual property protection and academia focusing on foundational studies and knowledge dissemination. The geographic distribution highlights strategic investments in SDV technology, particularly from China, Germany, and the USA.

IV. AUTOMOTIVE SOFTWARIZATION: PROMISES AND CHALLENGES

Following the lead of new electric vehicle (EV) manufacturers, such as Tesla, Rivian, Lucid, NIO, XPENG, etc., traditional automotive OEMs are increasingly adopting this new software-centric paradigm with the potential to redefine the ownership experience. Many positive outcomes are expected from this transformation–from product development to manufacturing and ultimately to end-to-end consumer experience. Some of the potential benefits include:

- **Simplified Architecture:** the modular design of SDVs enables the consolidation of vehicle components, such as standardized wiring harnesses, which reduces wiring, and overall mass.
- **Agile and Accelerated Development:** SDVs support a software-first, agile development approach that enables early-stage innovation through cloud-based simulation, virtual models, and digital twins. This shift-left strategy reduces time to market from years to months and promotes rapid iteration through MVP development and continuous feature delivery.
- **Continuous Feature Delivery:** over-the-air (OTA) updates allow manufacturers to fix bugs, improve performance, and roll out new features after the vehicle has been sold, ensuring continuous improvement.
- **Feature-as-a-Service (FaaS):** SDVs unlock on-demand capabilities such as premium driver assistance, adaptive drive modes, and in-cabin personalization. This supports flexible monetization and digital revenue streams.
- **Personalized User Experience:** software-defined cockpits can adapt to user preferences, offering personalized media, navigation, climate control, and in-cabin settings.
- **Increased Scalability and Code Reusability:** a unified software platform allows the same codebase to be deployed across multiple vehicle models, enhancing scalability and reducing duplication.
- **Sustainability and Operational Efficiency:** SDVs enable improved real-time, data-driven, and cloud-connected services, leading to smarter energy usage, optimized routing, and other features that contribute to more sustainable mobility.

While these benefits signal a promising shift toward SDVs, they are accompanied by a range of technical, operational, regulatory, and organizational challenges that must be addressed to realize the full potential of SDVs. The following subsections discuss these challenges.

A. MANAGING SOFTWARE IN THE VEHICLE DEVELOPMENT PROCESS

The development of software-defined vehicles introduces substantial technical challenges due to the deep integration of software into safety-critical, performance-sensitive, and highly distributed vehicular systems. As SDVs evolve into complex cyber-physical platforms, engineers must manage not only increased system complexity but also rigorous safety, security, and performance demands. A key challenge arises from the need to coordinate numerous heterogeneous software entities that interact in real-time under strict functional safety constraints. The vehicle, being a safety-critical device, requires that software operate reliably across subsystems with varying Automotive Safety Integrity Levels (ASIL) [32]. Furthermore, the dynamic and connected nature of SDVs exposes them to evolving cybersecurity threats, demanding robust and resilient software frameworks.

As automotive OEMs deeply integrate software into the vehicle development process, the industry faces several emerging challenges, including:

1) SOFTWARE AS AN ASSET

Traditionally, key vehicle functions were controlled by dedicated ECUs with embedded software, and the automotive industry did not manage software as an economic asset. Unlike the technology sector, which uses accounting practices to capture software value in financial books, automotive OEMs are now being challenged to redefine how they assess and manage software economically.

2) INTEROPERABILITY

There is no consistent industry-wide approach to delivering software from suppliers to OEMs, nor for integrating software from multiple sources. Incompatibility between software performing the same functions from different vendors is common, and the lack of standardized toolchains and validation processes further complicates interoperability.

3) CERTIFICATION

As software becomes more integral to vehicle functionality, ensuring reliability is critical–especially in safety-critical systems. This increases the need for third-party certification. While frameworks such as ASIL-D certification by TÜV [32] exist, there is no comprehensive system to certify automotive software. Current certification methods remain fragmented.

4) TESTING AND VALIDATION

Traditional automotive product development is a linear process, culminating at the endpoint, sometimes referred to as Start of Production (SOP). One of the major challenges in software-centric product development is transitioning from the traditional ''V-framework of design-development-testing'' [33] to a continuous development cycle that incorporates Continuous Integration and Continuous Delivery (CI/CD) practices. This shift necessitates the adoption of the principles of ''Shift Left'' (test early) and ''Shift Right'' (test and monitor in production), requiring tighter collaboration among designers, developers, and validators.

In the automotive context, ''Shift Left'' translates to embedding testing activities early in the design process, such as through model-in-the-loop (MiL) and software-in-the-loop (SiL) simulations, fault injection at design time, and automated unit and integration tests for safety-critical functions before hardware availability. This early-stage verification helps identify defects when they are cheapest to fix and ensures alignment with ISO 26262 safety goals from the outset. On the other hand, ''Shift Right'' emphasizes post-deployment monitoring and validation using in-field telemetry, OTA rollback mechanisms, and runtime anomaly detection, allowing developers to observe how systems behave under real-world conditions. This is particularly crucial for continuously evolving SDV features, where usage data and operational feedback can inform patches and improvements while the vehicle remains in operation.

Historically, automotive systems have been validated, revalidated, and rigorously safety certified according to ISO 26262 standards. Releasing software with equivalent rigor in a continuous delivery context is still an evolving challenge. The recent CrowdStrike incident in 2024 [34] demonstrated the potentially catastrophic consequences of a faulty, unvetted update pushed hastily to millions of connected devices. One can easily imagine the severity if those 8.5 million targets were safety-critical SDVs. Therefore, the industry must find ways to balance rigorous automotive-grade validation with the agility demanded by continuous software delivery. Although there are standards for functional safety (ISO 26262), cybersecurity (WP.29 in Europe) [35], and risk management (ISO/SAE 21434) [36], current testing protocols and certification schemes are insufficient for the demands of continuous delivery. OEMs and their supply chains must collaborate with certifying bodies such as TÜV, Exida, SAE, and ISO to develop updated processes and frameworks that can ensure both safety and agility.

5) ISSUES WITH OPEN-SOURCE SOFTWARE ADOPTION

While the benefits of open-source software in the SDV framework are understood, their adoption also opens the door for new vulnerabilities. Back in 2023, log4j, a widely used logging library under Java, had a serious security issue, and all Java applications had to be updated very soon to avoid execution of malicious code [37]. This is a good example of the adoption of an open-source library that can lead to significant safety and security issues. Similarly, researchers found a maliciously introduced backdoor to bypass normal authentication in the Linux utility XZ within the liblzma library [38]. The open-source codes being adopted need to be scanned regularly for any common vulnerability exposure (CVE), and the automotive industry needs to develop common counterstrategies against potential vulnerabilities.

6) SOFTWARE SECURITY

Since the 2015 Jeep Cherokee remote hack demonstration by Miller and Valasek [39], software security has remained a major concern. White-hat hackers have exposed vulnerabilities across numerous OEMs, including incidents involving Sirius-XM services and backend systems of brands such as BMW, Ferrari, and Mercedes-Benz [40]. Continuous scanning for vulnerabilities in both hardware and software libraries is essential. SDV security must be designed from the outset and include comprehensive protections for OTA updates. Although regulations such as UN R156 (SUMS) [41] offer guidance, they are not sufficient alone. Validating updates and preventing the propagation of faulty software are now critical components of the development lifecycle. Several studies have investigated the diverse safety and security challenges associated with software-defined vehicles. For instance, functional safety risk decomposition and optimization under ISO 26262 requirements have been proposed in [42]. Denial-of-service and jamming attacks remain prominent threats, as highlighted in [43]. Real-time intrusion response has been explored by applying adaptive trust models [44]. Challenges in integrating Linux into safety-critical systems are noted in [45]. Wireless vulnerabilities and attack surfaces in connected SDVs are discussed in [46]. Finally, mechanisms for license validation [47], secure service call authentication [48], and tamper-resistant vehicle data storage [49] have also been proposed.

B. REPAIR AND LONG-TERM MAINTENANCE

Repairs in SDVs are increasingly complex due to the vast scale of software involved–often ranging from 100 to 300 million lines of code across application layers, middleware, and hardware drivers. Troubleshooting or fixing corrupted software will require sophisticated diagnostic tools and skilled expertise. While many issues may be resolved remotely through OTA updates, others–particularly those impacting functional safety governed by standards like ISO 26262 [32]–may necessitate on-site service and direct collaboration with semiconductor or software suppliers. The involvement of third-party software components, especially if they introduce security vulnerabilities, could further complicate repair workflows and liability management.

In addition to technical challenges, the costs associated with software repair and maintenance are expected to be high. Unlike the traditional ecosystem of third-party service providers for physical components, the software in SDVs is typically proprietary and tightly integrated with the OEM's infrastructure. This exclusivity inherently drives up the cost of maintenance and restricts service options for vehicle owners.

Moreover, with vehicles expected to stay on the road longer, software maintenance over a product's lifespan becomes crucial. Challenges include maintaining traceability to older software versions and ensuring backward compatibility across generations of hardware. Even with ISO 26262 [32] and ASPICE traceability requirements [50], compatibility issues may arise between software and hardware of different eras. For further discussion, see Maro et al. [51]. Another growing concern is the risk of software discontinuity. Dependence on OEMs for updates and support increases the likelihood of systems becoming outdated if support is withdrawn for older technologies. These risks are further exacerbated if an OEM or its suppliers go out of business, are acquired, or declare bankruptcy–leaving users without access to critical software support or security patches.

C. PRODUCT DEVELOPMENT IN "SOFTWARE-FIRST" PARADIGM

With the decoupling of hardware and software in SDV architecture, frequent OTA updates promise to keep vehicles perpetually "fresh." However, this software-centric paradigm imposes new demands on hardware as well.

1) FUTURE PROOFING

While software can be updated over time, the underlying hardware will eventually require upgrades. This may lead to a vehicle ecosystem containing multiple hardware versions, complicating software compatibility. Ideas such as modular hardware design, chiplets, and plug-and-play components offer potential solutions, but there is no universally adopted strategy across OEMs and suppliers.

2) SCALABILITY

OEMs must scale hardware platforms across diverse vehicle models, price points, and trims. This scalability must allow both upward expansion to support premium features and downward adaptation for entry-level models. Notably, scaling down is often more challenging, especially for high-volume manufacturers aiming to deliver a broad product portfolio efficiently.

D. THE GAP BETWEEN TRADITIONAL AND NEW AUTOMAKERS

Tesla continues to sit at the top of the list of OEMs perfecting their SDV framework, and several new EV-only OEMs such as Lucid and Rivian in the US and Nio, Xpeng, Geely and BYD in China are closely following. These companies have embraced a "software-first" clean sheet approach, ensuring that software is central to every aspect of vehicle development and operation, while the traditional OEMs like VW, GM, Ford, Mercedes, Toyota, etc. are finding the transformation from a hardware-centric approach more challenging. This is the conclusion of the recent 2024 Gartner Digital Automaker Index [52] shown in Figure 7, which measures the digital maturity and technological capabilities of global automakers, focusing on their transition to software-centric, technology-driven enterprises.

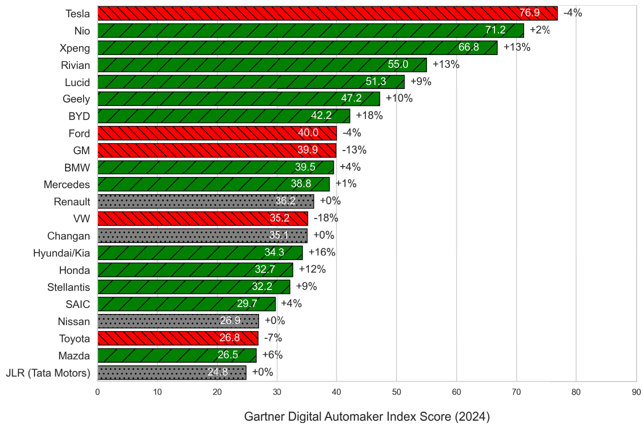

FIGURE 7. Gartner Digital Automaker Index 2024 and Improvement vs 2023. Green indicates a positive change, Gray indicates no change, and Red indicates a negative change compared to 2023.

This digital automaker index evaluates 22 automakers across multiple metrics such as leadership, talent, and technology integration, including software development, use of OTA updates, integration of AI, cloud adoption, and digital product strategies. The index highlights each automaker's progress in shifting from hardware-centric models to software-defined, service-oriented platforms, offering valuable insights into their readiness for the evolving mobility landscape. It reflects how automakers are adapting to major industry shifts, embracing the new technology, including connectivity and SDV paradigm, and identifies which companies are leading in integrating digital technologies into their vehicles and business operations. The [53], [54] provide an in-depth landscape of the challenges in launching SDV by the traditional OEMs.

E. CHANGING SUPPLY CHAIN STRUCTURE

The traditional supply chain of the industry is changing from a hierarchical structure where the Tier-1 suppliers such as Bosch, Continental, ZF etc. hold a pre-eminent position to a network-like structure – where OEMs may be working directly with anyone, even a startup. On one hand, this requires the OEMs to develop skill sets that traditionally reside in Tier-1s. On the other hand, Tier-1s, as well as large technology companies and small startups, have to figure out the optimum interaction with the OEMs as illustrated in Figure 8.

F. RESEARCH AND EDUCATIONAL BARRIERS

Despite the growing interest in SDVs, significant research and educational barriers continue to hinder progress in this domain. One major limitation is the limited availability of open-source simulators, testbeds, and experimentation platforms tailored to SDV development and validation. The proprietary and complex nature of automotive software

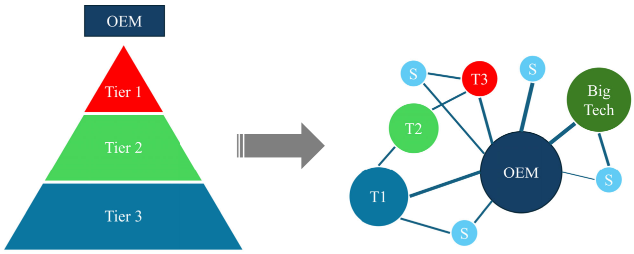

FIGURE 8. OEM: Orgingal Equipmeent Manufactuer, T1, 2,3: Tiered Suppliers, S: Startups, Big Tech: Technology giants such as Google, Nvidia, AWS, etc.

systems restricts access to realistic environments needed for hands-on learning and prototyping.

Promising initiatives are emerging to support SDV development and education. For instance, Eclipse Velocitas provides an open-source development toolchain for building containerized in-vehicle applications, while digital.auto's dreamKIT [55] and Playground [56] offer modular, cloud-based environments for rapid prototyping and simulation of vehicle functions. However, these platforms are still in the early stages of adoption and have not yet been widely integrated into academic curricula or research workflows. Moreover, there is a pressing need for accessible, high-quality platforms that support agile hands-on experimentation and lower the barrier for innovation in automotive softwarization. Compounding these issues is the lack of open datasets covering critical areas such as anomaly detection, causal inference, and runtime reliability, limiting the reproducibility, benchmarking, and advancement of data-driven research. Overcoming these challenges is essential to equip the next generation of engineers and researchers with the tools necessary to innovate in the SDV landscape.

G. SUSTAINABILITY AND WASTE CONCERNS

The increasing reliance on electronic and software components in SDVs introduces new sustainability challenges. One of the most pressing issues is obsolescence: the rapid evolution of hardware and software raises the risk of generating significant electronic waste. Unlike mechanical parts, which can often be repaired or reused for decades, digital components and platforms could become obsolete sooner.

To mitigate environmental impact, sustainability must be embedded in the design of SDVs. This includes developing hardware systems that support reusability and recyclability, reducing the ecological footprint throughout the product lifecycle. Designing with modularity in mind can also ease upgrades and prevent entire systems from becoming redundant due to a single outdated component. Furthermore, just as the electric vehicle industry has created a secondary market for used batteries, the SDV ecosystem may need to explore options for repurposing high-tech vehicle components. Collaboration among OEMs, semiconductor suppliers, and recyclers will be essential to ensure that components such as sensors, chips, and computing units can find a second-life utility beyond their original deployment.

H. REGULATORY AND STANDARDIZATION ISSUES

One of the key challenges facing the development and deployment of software-defined vehicles is effective data governance. As SDVs increasingly rely on the collection and processing of sensitive data, such as location information, driving behavior, and in-cabin preferences, it is critical to address privacy, equity, and data ownership concerns without compromising vehicle safety or performance. In addition, compliance with a complex and evolving landscape of global and regional regulations presents a persistent hurdle for automakers. Ensuring adherence to varying legal requirements across jurisdictions adds significant complexity to software deployment, data handling, and in-vehicle services.

Another major issue is the lack of standardized protocols and development frameworks specific to SDVs. Building software-centric vehicles requires manufacturers to adopt open standards and leverage open-source software where feasible. This approach allows automakers to allocate their limited in-house resources toward innovation and the creation of brand-differentiating features and applications. To support this vision, the industry has seen the emergence of several working groups and consortia over the past decade–including AUTomotive Open System ARchitecture (AUTOSAR), Eclipse SDV, Scalable Open Architecture for Embedded Edge (SOAFEE), and Connected Vehicle Systems Alliance (COVESA)–focused on standardization, interoperability, and common tooling across SDV ecosystems. Organizations such as Eclipse SDV, for instance, play a key role in promoting the adoption and governance of open-source solutions in automotive environments. However, as standardization efforts expand, coordination among these groups becomes increasingly important. Although each consortium operates under a unique charter and focuses on different aspects of the SDV landscape, there is a growing need for alignment and collaboration to avoid redundancy and ensure cohesive progress. Without such coordination, efforts may overlap, creating inefficiencies and slowing industry-wide adoption.

V. FUTURE DIRECTIONS AND RESEARCH OPPORTUNITIES

As software-defined vehicles continue to evolve, research and partnership opportunities are emerging. These opportunities span technical, operational, educational, and regulatory domains, reflecting the complex and multidisciplinary nature of SDV ecosystems. This section outlines key areas where innovation, collaboration, and further study are essential to address current limitations and unlock the full potential of SDVs.

Ecosystem Collaboration and Open Innovation for SDV Advancement:

Traditionally, the automotive industry has followed a relatively "closed innovation" model, with a strong emphasis on proprietary technology and tightly controlled development pipelines. However, with the advent of the software-first approach inherent in SDV architectures, where the underlying hardware is decoupled from application software and middleware, there is an increasing need for standardization, collaboration, and the adoption of open-source software, particularly for non-differentiating applications. This paradigm shift has catalyzed cross-industry collaboration, drawing expertise from cloud and IoT sectors and giving rise to new consortia such as SOAFEE, Eclipse SDV, and COVESA as previously mentioned. These organizations are focused respectively on bringing cloud-native technologies to automotive systems, fostering open-source development models, and establishing common data formats for interoperability.

Looking ahead, fostering deeper collaboration within the mobility ecosystem will be essential to unlock the full potential of SDVs. The open innovation approach offers a promising foundation for addressing the complexity of SDV development, where multiple stakeholders, including OEMs, suppliers, software vendors, and cloud providers, must coordinate across organizational and technological boundaries. This approach can encourage shared toolchains, co-development frameworks, and data interoperability across platforms. A forward-looking example is SDVerse, a novel automotive software marketplace developed by GM, Magna, and Wipro, designed to serve all OEMs, suppliers, and companies with relevant software offerings. SDVerse brings together buyers and sellers to streamline software procurement and accelerate integration workflows. Expanding the adoption of such open marketplaces and shared innovation frameworks will be instrumental in reducing development time, promoting reuse, and supporting scalable, modular SDV architectures. Continued research and industry alignment around open innovation models will be key to advancing the agility, interoperability, and sustainability of SDVs.

A. DEVELOPMENT OF OPEN DATASETS

Despite the growing complexity of SDV functionalities, there is a noticeable lack of publicly available datasets tailored to the unique characteristics of SDV systems. Proprietary restrictions and fragmented data collection practices across OEMs and suppliers hinder reproducibility and benchmarking efforts. To foster innovation, accelerate algorithm development, and enable robust evaluation frameworks, the SDV research community must prioritize the creation, curation, and standardization of open-source datasets that include multimodal data streams (e.g., vehicle state, sensor logs, software execution traces, and system health metrics). For instance, in the context of SDV observability, telemetry datasets that integrate metrics, traces, and logs are crucial for developing and validating real-time monitoring, anomaly detection, causality analysis, and automated incident response systems. The establishment of such datasets is essential for promoting transparent, reproducible research and fostering cross-disciplinary collaboration within the SDV ecosystem.

B. OPEN EDUCATION SIMULATORS AND TESTBEDS

The advancement of SDV research and education requires greater emphasis on accessible, modular, and open-source simulation and testing environments. While promising initiatives such as digital.auto's dreamKIT [55] and digital.auto Playground [56] have emerged, they remain in the early stages of adoption and have yet to see widespread integration into academic curricula or industrial research workflows and training centers. The lack of standardized, open-access SDV simulators and testbeds presents a significant barrier to hands-on learning, rapid prototyping, and cross-institutional collaboration. In parallel, there is also a notable shortage of online courses and structured training programs in the SDV domain, despite initial efforts such as the "SDV 101: Introduction to Software Defined Vehicles" course [57] offered by digital.auto. Future research and educational efforts should prioritize the development and dissemination of educational platforms that allow students, researchers, and practitioners to experiment with real-world SDV functionalities within reproducible and extensible environments. Such tools will be essential in scaling SDV knowledge, democratizing innovation, and bridging the gap between academic exploration and industry-grade implementation.

C. QUANTIFIABLE QUALITY OF EXPERIENCE (QOX)

The International Telecommunication Union (ITU) defines Quality of Experience (QoE or QoX) as the overall acceptability of an application or service, as perceived subjectively by the end user [58]. QoX reflects the degree of delight or annoyance experienced by users based on how well their expectations are met with respect to the utility and enjoyment of a given technology, in light of their personality and current state. In the context of software-defined vehicles, this translates to the user's satisfaction with features such as adaptive drive modes, personalized in-cabin settings, or infotainment services, all of which are increasingly defined and delivered through software. QoX is influenced by three major categories of factors: system influence factors, which relate to the performance and features of the SDV platform; human influence factors, encompassing the user's expectations, mood, and preferences; and context influence factors, such as driving conditions, traffic, or environmental context. Unlike traditional vehicles, where experience was tied to mechanical performance and static features, SDVs offer dynamic, software-driven experiences that can adapt in real-time. Measuring QoX in SDVs therefore requires the development of metrics that assess comfort, responsiveness, and safety under varying conditions, as well as intelligent algorithms that optimize vehicle behavior in response to real-time QoX feedback.

D. COGNITIVE DATA SAMPLING

SDVs generate vast amounts of feature utilization and telemetry data from various distributed and heterogeneous software entities, ranging from embedded systems controlling mission-critical vehicle functions to OTA updatable modules and user-facing applications. While these data are invaluable for improving vehicle performance, enabling predictive maintenance, creating new business opportunities and enhancing user experience, they also impose significant burdens on OEM infrastructures–raising costs for data collection, transmission, storage, and analysis, and straining bandwidth and computational resources. Additional resources need to be in place to adhere to sensible data privacy policies. To mitigate these challenges, cognitive data sampling methods are crucial. Rather than collecting all data indiscriminately, these methods use contextual cues to prioritize what is most relevant.

Contextual categories for data sampling in SDVs include several strategies tailored to optimize efficiency and relevance. Time-based sampling collects data according to temporal patterns, such as seasonal variations or peak usage periods. Location-based sampling focuses on data from specific geographic regions, particularly those with distinctive road infrastructures. Event-based sampling is triggered by specific operational, environmental, or regulatory events, such as over-the-air (OTA) updates, weather anomalies, or compliance checks. Finally, adaptive sampling dynamically adjusts data collection rates in response to real-time conditions like system health, resource limitations, or detected anomalies. For example, it may increase sampling during unexpected behavior or apply energy-aware strategies in electric vehicles.

To ensure sampling accuracy, statistical methods such as stratified sampling can be employed to maintain representativeness across vehicle types, usage patterns, and environments. In addition, achieving an optimal balance between cloud and edge computing is essential. Edge computing enables low-latency, real-time decisions (e.g., collision avoidance), while cloud platforms provide scalable storage and AI-driven insights from aggregated data. Strategic workload distribution across these layers ensures efficiency and performance.

Additionally, smart and targeted data sampling with appropriate consumer consent can also be important for consumer confidence. A spate of recent data leaks from vehicles has raised consumer unease on the subject. Instead of a broad-brush data collection, a more precise and limited approach, i.e., collect only what is necessary for a service, can put consumers at ease.

E. CONTEXTUAL OBSERVABILITY

Observability, defined as the ability to infer system internal states from its external outputs, is underdeveloped for SDVs and is critical for their safe and reliable operation. Efficient real-time data collection and analysis are required to adapt to dynamic environments. Key capabilities include continuous monitoring to detect outages, bottlenecks, or operational anomalies; causal inference to identify root causes and contributing factors behind detected issues; and automated incident response to enable real-time resolution, reducing downtime and maintaining system availability.

In the context of SDVs, emerging frameworks such as ASAM's SOVD (Service-Oriented Vehicle Diagnostics) [59] and cloud-integrated telemetry pipelines, often built on open-source tools like Prometheus and Grafana, are being explored to support scalable, real-time observability. Observability platforms such as Grafana, Elastic, Dynatrace, Cisco Systems (Splunk), Chronosphere, New Relic, Logz.IO, and SigNoz [60] provide foundational support but typically rely on correlation-based analyses. Integrating causal AI [61] techniques, including causal discovery, identification, estimation, and validation, will elevate observability precision. Moreover, these systems should incorporate critical contextual information such as Service Level Objectives (SLOs), OTA deployment logs, network conditions, peak load metrics, and environmental factors to ensure optimal insights and adaptive behavior. By enriching raw telemetry with this contextual information, the observability system can distinguish true anomalies from expected transients. For instance, it can recognize that a sudden CPU spike coinciding with a large OTA rollout is a normal, scheduled workload rather than a performance fault. This contextual awareness prevents false alarms, dynamically adjusts alert thresholds during known high-load periods, and ensures that alerts only trigger when deviations fall outside the bounds of planned updates and operational parameters.

VI. CONCLUSION

The shift toward Software-defined Vehicles (SDVs) is redefining the automotive industry by moving from hardware-centric systems to software-driven platforms. This evolution introduces modular architectures, continuous over-the-air (OTA) feature delivery, and new service-oriented business models such as Feature-as-a-Service (FaaS). While these advancements promise significant benefits, they also introduce substantial challenges in areas such as safety, scalability, reliability, and cost. These issues are especially complex for traditional OEMs and their supply networks. Addressing them requires the adoption of agile development methodologies and continuous integration and delivery (CI/CD) practices, while maintaining automotive-grade quality standards.

This paper presented a systematic review of the SDV landscape, highlighting both the intensity and imbalance of contributions from academia and industry. Academic research often focuses on foundational technologies and architectural concepts, whereas industry patents tend to emphasize system integration and commercialization. There is a need for stronger collaboration between academia and industry to bridge the current gap among practitioners, students, and researchers.

The review also identified critical challenges including fragmented certification protocols, the economic management of software as a strategic asset, insufficient tools for testing and validation, and increasing repair and maintenance costs. Additional hurdles include the need for resilient software operations across vehicle and cloud systems, scalable hardware platforms, evolving supply chain structures, and effective governance of vehicle data. Moreover, the lack of open datasets, testbeds, and simulation platforms continues to hinder reproducibility and slow progress in key areas such as anomaly detection and contextual observability.

This paper highlighted several future directions and research opportunities to address these gaps and unlock the full potential of SDVs. Key priorities include fostering ecosystem collaboration, adopting open innovation models, and advancing the development of open datasets, educational simulators, and testbeds. Further research should also focus on establishing quantifiable Quality of Experience (QoX) metrics, developing cognitive data sampling methods, and enhancing contextual observability frameworks.

SDVs represent more than a technological progression; they signify a systemic and unprecedented transformation of the automobile into a continuously evolving, software-defined digital platform. Realizing this vision will require coordinated efforts across innovation, regulation, infrastructure, and human capital development to enable a scalable and sustainable software-centric mobility future.

REFERENCES

[1] D. Slama, A. Nonnenmacher, and T. Irawan, *The Software-Defined Vehicle: A Digital-First Approach to Creating Next-Generation Experiences*. CA, USA: O'Reilly Media, 2023.

[2] A. Khamis, *Smart Mobility: Exploring Foundational Technologies and Wider Impacts*. New York, NY, USA: Apress, 2021.

[3] C. Yorozu and V. G. Shi, "Servitization and Japan's keiretsu organization," in *Proc. Spring Servitization Conf.*, 2014, pp. 1–12.

[4] T. Baines and V. G. Shi, "Servitization transformation: Drivers, benefit and barriers," in *Proc. Spring Servitization Conf.*, 2014, pp. 34–39.

[5] T. Baines, H. Lightfoot, J. Peppard, M. Johnson, A. Tiwari, E. Shehab, and M. Swink, "Towards an operations strategy for product-centric servitization," *Int. J. Oper. Prod. Manage.*, vol. 29, no. 5, pp. 494–519, Apr. 2009.

[6] C. Raddats, J. Burton, J. Zolkiewski, V. M. Story, T. Baines, and H. Lightfoot, "Servitization capabilities for advanced services: A multi-actor perspective," in *Proc. Spring Servitization Conf.*, May 2014, pp. 126–132.

[7] S. Jiang, "Vehicle E/E architecture and key technologies enabling software-defined vehicle," SAE, Tech. Rep. 2024-01-2035, 2024.

[8] F. Pan, M. Rickert, T. Betz, L. Wen, J. Lin, N. Petrovic, M. Lienkamp, and A. Knoll, "Towards software-defined vehicles: From model-based engineering to virtualization-based deployment," *IEEE Access*, vol. 12, pp. 192127–192145, 2024, doi: 10.1109/ACCESS.2024.3512002.

[9] A. Ghammam, R. Khalsi, M. Kessentini, and F. Hassan, "Efficient management of containers for software defined vehicles," *ACM Trans. Softw. Eng. Methodology*, vol. 33, no. 8, pp. 1–36, Nov. 2024.

[10] A. Ankit, K. Nimala, and M. Jadhav, "Creation of continuous integration continuous deployment pipeline using cloud," in *Proc. 5th Int. Conf. Intell. Commun. Technol. Virtual Mobile Netw. (ICICV)*, Mar. 2024, pp. 337–341.

[11] A. Szalavetz, "In-house software development for software-defined vehicles: Major changes ahead in automotive value chains," *Int. J. Automot. Technol. Manage.*, vol. 24, no. 5, pp. 22–39, 2024, doi: 10.1504/ijatm.2024.142127.

[12] J. Jahić, "SCATS framework for software integration in software-defined vehicle with cross-organizational agile teams," in *Proc. Eur. Conf. Softw. Archit.*, Jan. 2024, pp. 371–378.

[13] A. Mattausch, J. Schlosser, and M. Neukirchner, "E/E architectures and the automotive OS," in *Proc. Int. Stuttgart Symp.*, Jan. 2023, pp. 175–183.

[14] H. Askaripoor, T. Mueller, and A. Knoll, "E/E designer: A framework to design and synthesize vehicle E/E architecture," *IEEE Trans. Intell. Vehicles*, vol. 9, no. 7, pp. 5204–5221, Jul. 2024.

[15] L. Mauser and S. Wagner, "Centralization potential of automotive E/E architectures," *J. Syst. Softw.*, vol. 219, Jan. 2025, Art. no. 112220.

[16] A. Shamim, "Containerization for the software-defined vehicle," *ATZelectronics Worldwide*, vol. 18, no. 12, p. 58, Dec. 2023.

[17] L. Wen, M. Rickert, F. Pan, J. Lin, and A. Knoll, "Bare-metal vs. hypervisors and containers: Performance evaluation of virtualization technologies for software-defined vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2023, pp. 1–8.

[18] T. Betz, L. Wen, F. Pan, G. Kaljavesi, A. Zuepke, A. Bastoni, M. Caccamo, A. Knoll, and J. Betz, "A containerized microservice architecture for a ROS 2 autonomous driving software: An end-to-end latency evaluation," in *Proc. IEEE 30th Int. Conf. Embedded Real-Time Comput. Syst. Appl. (RTCSA)*, Aug. 2024, pp. 57–66.

[19] A. Kampmann, A. Mokhtarian, S. Kowalewski, and B. Alrifaee, "ASOA—A dynamic software architecture for software-defined vehicles," in *Proc. Aachen Colloq. Sustain. Mobility*, 2022, pp. 1–7.

[20] D. Philipp Klüner, M. Molz, A. Kampmann, S. Kowalewski, and B. Alrifaee, "Modern middlewares for automated vehicles: A tutorial," 2024, *arXiv:2412.07817*.

[21] F. Pan, J. Lin, M. Rickert, and A. Knoll, "Resource allocation in software-defined vehicles: ILP model formulation and solver evaluation," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2022, pp. 2577–2584.

[22] J. Becker, "A safety-certified automotive SDK to enable software-defined vehicles," in *Proc. Workshop Fahrerassistenz und automatisiertes Fahren*, 2022, pp. 1–6. [Online]. Available: https://www.uni-das.de/images/pdf/fas-workshop/2022/FAS2022-12-Becker.pdf

[23] N. N. Surjekar, Y. Patwardhan, and V. Konduju, "A case study on migrating towards functionally safe zonal architecture using MBSE," in *Proc. INCOSE Int. Symp.*, Jul. 2023, vol. 33, no. 1, pp. 1403–1417.

[24] G. Vitale and M. Hollander, "The software-defined vehicle: How to verify and validate software functions," in *Proc. Int. Stuttgart Symp.*, Jan. 2023, pp. 399–405.

[25] B. Boehlen, D. Fischer, and J. Wang, "Diagnostics of automotive service-oriented architectures with SOVD," SAE, Tech. Rep. 2024-01-7036, 2024.

[26] Z. Liu, W. Zhang, and F. Zhao, "Impact, challenges and prospect of software-defined vehicles," *Automot. Innov.*, vol. 5, no. 2, pp. 180–194, Apr. 2022.

[27] U. Bordoloi, S. Chakraborty, M. Jochim, P. Joshi, A. Raghuraman, and S. Ramesh, "Autonomy-driven emerging directions in software-defined vehicles," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Apr. 2023, pp. 1–6.

[28] J. H. Panchal and Z. Wang, "Design of next-generation automotive systems: Challenges and research opportunities," *J. Comput. Inf. Sci. Eng.*, vol. 23, no. 6, Dec. 2023, Art. no. 060818.

[29] S. Sureddi, "Adopting agile methodologies and frameworks in automotive industry," *J. Artif. Intell., Mach. Learn. Data Sci.*, vol. 1, no. 1, pp. 1727–1731, Nov. 2022.

[30] M. Askarpour, S. Kokaly, and S. Ramesh, "An exploration of agile methods in the automotive industry: Benefits, challenges and opportunities," 2024, *arXiv:2409.12676*.

[31] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, "Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement," *PLoS Med.*, vol. 6, no. 7, Jul. 2009, Art. no. e1000097.

[32] *ISO 26262: Road Vehicles—Functional Safety*, ISO Standard 26262, 2018. [Online]. Available: https://www.iso.org/standard/68383.html

[33] M. Staron and M. Staron, "Automotive software development," in *Automotive Software Architectures: An Introduction*. Cham, Switzerland: Springer, 2021, pp. 67–95.

[34] I. Naseer, "The crowdstrike incident: Analysis and unveiling the intricacies of modern cybersecurity breaches," *World J. Adv. Eng. Technol. Sci.*, vol. 13, no. 1, pp. 728–733, Oct. 2024.

[35] United Nations Econ. Commission for Eur. (UNECE). (2021). *UN Regulation, no. 155–Cyber Security and Cyber Security Management System*. Accessed: Apr. 20, 2025. [Online]. Available: https://unece.org/wp29-introduction

[36] *ISO/SAE 21434: Road Vehicles—Cybersecurity Engineering*, ISO/SAE Standard 21434, 2021.

[37] M. Kosinski. (Aug. 2023). *How to Detect and Patch a Log4J Vulnerability*. [Online]. Available: https://www.ibm.com/blog/how-to-detect-patch-log4j-vulnerability/
[38] D. Goodin. (Apr. 2024). *The XZ Backdoor: Everything You Need to Know*. [Online]. Available: https://www.wired.com/story/xz-backdoor-everything-you-need-to-know/
[39] C. Miller and C. Valasek. (Aug. 2015). *Remote Exploitation of an Unaltered Passenger Vehicle*. [Online]. Available: https://illmatics.com/Remote%20Car%20Hacking.pdf
[40] K. Dhingra. (Jan. 17, 2023). *White Hat Hackers Gain Access to Data at 7 More Automakers*. Automot. News. [Online]. Available: https://www.autonews.com/mobility-report/how-these-hackers-accessed-information-7-automakers
[41] United Nations Econ. Commission for Eur. (UNECE). (2021). *UN Regulation, no. 156–Uniform Provisions Concerning the Approval of Vehicles With Regards to Software Update and Software Update Management System (SUMS)*. [Online]. Available: https://unece.org/transport/documents/2021/03/standards/un-regulation-no-156-software-update-and-software-update
[42] G. Q. Xie, W. Wu, G. Zeng, R. F. Li, and S. Y. Hu, "Risk assessment and development cost optimization in software defined vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 7375–7386, Jul. 2023.
[43] C. Bodei, M. D. Vincenzi, and I. Matteucci, "From hardware-functional to software-defined vehicles and their security issues," in *Proc. IEEE 21st Int. Conf. Ind. Informat. (INDIN)*, Jul. 2023, pp. 1–8.
[44] R. Kaster and D. Ma, "Zero-trust enabling intrusion detection protection for software defined vehicles," in *Proc. IEEE 3rd Int. Conf. Comput. Mach. Intell. (ICMI)*, Apr. 2024, pp. 1–6.
[45] J. Schlosser and J. Petersohn, "Considering Linux for functional safety relevant system architecture: Pitfalls and potential," in *Proc. Int. ATZ Conf.*, Jan. 2024, pp. 89–102.
[46] N. R. Prasad, B. Andersen, and D. K. Rubin-Grøn, "Overview of security challenges in wireless IoT infrastructures for autonomous vehicles," in *Proc. Int. Conf. Saf. Secur. IoT*, Jan. 2024, pp. 34–41.
[47] Wuhu Lion Automotive Technology Co., Ltd. and Chery Automobile Co., Ltd., "Security license file analysis method, device, vehicle and storage medium," C.N. Patent 117 609 961-A, Feb. 27, 2024.
[48] Avita Technology (Chongqing) Co., Ltd., "Service call authentication method and system," C.N. Patent 118 740 451-A, Oct. 8, 2024.
[49] Chongqing Changan Automobile Co., Ltd., "Safe storage method, device, equipment and medium for automobile data," C.N. Patent 115 357 921-A, Nov. 18, 2022.
[50] Automot. SIG (Special Interest Group). (2023). *Automotive SPICE Process Assessment Model Version 4.0*. [Online]. Available: https://www.automotivespice.com
[51] S. Maro, J.-P. Steghöfer, and M. Staron, "Software traceability in the automotive domain: Challenges and solutions," *J. Syst. Softw.*, vol. 141, pp. 85–110, Jul. 2018.
[52] E. Surucinski. (Oct. 2024). *How Digital Are Car Manufacturers? Insights From Gartner's Digital Index 2024*. Accessed: Jul. 7, 2025. [Online]. Available: https://oneest.com/how-digital-are-car-manufacturers-in-2024-insights-from-gartner%E2%80%99s-digital-index-2024
[53] P. Goswami, "The software-defined vehicle: Its current trajectory and execution challenges," SAE Int., Warrendale, PA, USA, Tech. Rep. EPR2024027, 2024.
[54] P. Goswami, "The software-defined vehicle and its engineering evolution: Balancing issues and challenges in a new paradigm of product development," SAE Int., Warrendale, PA, USA, Tech. Rep. EPR2024007, 2024.
[55] Digital.auto. (2023). *DreamKIT—Developer Kit for Exploring the SDV Ecosystem*. Accessed: Apr. 20, 2025. [Online]. Available: https://www.digital.auto/dreamkit
[56] Digital.auto. (2023). *Digital.auto Playground—Explore Software-Defined Vehicle APIs*. Accessed: Apr. 20, 2025. [Online]. Available: https://www.digital.auto/playground
[57] Digital.auto. (2024). *SDV 101: Introduction to Software Defined Vehicles*. Accessed: Apr. 20, 2025. [Online]. Available: https://www.digital.auto/learn/sdv-101
[58] Int. Telecommun. Union. (2017). *ITU-T Recommendation P.10/G.100: Vocabulary for Performance, Quality of Service and Quality of Experience*. Accessed: Apr. 20, 2025. [Online]. Available: https://standards.globalspec.com/std/10264138/g-100
[59] ASAM. (2021). *ASAM SOVD: Service-Oriented Vehicle Diagnostics*. Accessed: Apr. 20, 2025. [Online]. Available: https://www.asam.net/standards/detail/sovd/
[60] Gartner, Inc. (2025). *Observability Platforms Reviews and Ratings*. Accessed: Jul. 9, 2025. [Online]. Available: https://www.gartner.com/reviews/market/observability-platforms
[61] J. S. Hurwitz and J. K. Thompson, *Causal Artificial Intelligence: The Next Step in Effective Business AI*. Hoboken, NJ, USA: Wiley, 2023.

ALAA KHAMIS (Senior Member, IEEE) is currently an Associate Professor and the Director of the AI for Smart Mobility Laboratory, Department of Industrial and Systems Engineering, and the Interdisciplinary Research Center for Smart Mobility and Logistics, KFUPM. Prior to joining KFUPM, he was the AI and Smart Mobility Technical Leader of General Motors Canada. He is also an Adjunct Faculty Member with the University of Toronto and an Adjunct Professor with Ontario Tech University. He has authored four books and over 190 scientific papers published in refereed journals and international conferences. He holds 72 U.S. patents, trade secrets, and defensive publications. He is the author of *Smart Mobility: Exploring Foundational Technologies and Wider Impacts* and *Optimization Algorithms: AI Techniques for Design, Planning, and Control Problems*. His research interests include the intersection of AI and mobility systems, services, and business models, addressing challenges, such as seamless integrated mobility, observability in software-defined vehicles (SDVs), combining reasoning and action, contextual causal inference, and optimization in people mobility, logistics, and transportation infrastructure. He was a recipient of the 2022–2024 GM Critical Talent Award, the First Place in the 2025 Sustainable Solutions for Pilgrims Challenge, the Best Paper Award in the Smart Mobility Governance Track at the 2023 IEEE International Conference on Smart Mobility, and the 2018 IEEE Member and Geographic Activities (MGA) Achievement Award.

PARTHA GOSWAMI received the M.B.A. degree from the University of Michigan, and the Ph.D. degree in applied mechanics from Iowa State University. He has a certification as a foresight professional from the University of Houston. He has more than 30 years of experience in the auto industry, 28 of which were spent at General Motors (GM) in diverse functions ranging from product development, research and development, planning, finance, and technology strategy. In his most recent role in GM as Professional Fellow of technology trends, he was responsible for identifying emerging tech trends and their impact on future competitive strategy. He currently provides consulting and advisory services as a Principal at PG Mobility Analysis, LLC, focusing on smart mobility and disruptions by connectivity, autonomous, services, electrification (CASE), and software-defined vehicles (SDVs). He regularly speaks at industry forums on technology trends and their impacts. He also serves as the director of 3IS, Inc. He is actively engaged as an advisor in several industry groups, as a judge for startups in GAMIC, Michigan, and is involved with COVESA and SAE in various capacities.