



# THE FUTURE OF AUTOMOTIVE INNOVATION: EXPLORING THE IN-VEHICLE SOFTWARE ECOSYSTEM AND DIGITAL VEHICLE PLATFORMS

**Sukruthi Reddy Sangannagari**

Senior Quality Assurance Specialist and Full Stack Developer, FM Global, USA.

## ABSTRACT

*The automotive industry is undergoing dramatic transformation as a result of connectivity, software-defined vehicle technologies, and electrification. Software-defined vehicles (SDVs) are transforming the in-vehicle software environment, facilitating continuous innovation through software-driven updates and new feature enhancements without the need for hardware modifications. Cloud connectivity is revolutionizing vehicles into dynamic, software-centric platforms that facilitate over-the-air updates, real-time data exchange, and communication with cloud-based applications. Cooperative ecosystems, such as OEMs, software providers, and tech providers, support the migration from hardware-centric to software-centric cars, enabling effective development and open innovation. Digital Vehicle Platform (DVP) integrates cloud services, zone-based electrical topologies, and centralised computing into a scalable, adaptable software ecosystem. DVP enables advanced driver assistance systems, continuous over-the-air software upgrades, and personalised digital experiences. The research explores architecture, implementation process, and key*

*technologies enabling DVP, including cybersecurity, interoperability, integration complexity of software, and regulatory compliance. Indicators of progress reveal improved quality, safety, and delivery performance of software, enabled by AI-powered verification and advanced testing frameworks. The DVP paves the way for software-defined cars with enhanced autonomy, edge-cloud integration, and green electrification.*

**Keywords:** Software-defined vehicles (SDVs), Collaborative ecosystems, Digital Vehicle Platform (DVP), Cybersecurity, Edge-Cloud Integration

**Cite this Article:** Sukruthi Reddy Sangannagari. (2022). The Future of Automotive Innovation: Exploring the in-Vehicle Software Ecosystem and Digital Vehicle Platforms. *Journal of Advanced Research Engineering and Technology (JARET)*, 1(2), 39-51. DOI: [https://doi.org/10.34218/JARET\\_01\\_02\\_004](https://doi.org/10.34218/JARET_01_02_004)

[https://iaeme.com/MasterAdmin/Journal\\_uploads/JARET/VOLUME\\_1\\_ISSUE\\_2/JARET\\_01\\_02\\_004.pdf](https://iaeme.com/MasterAdmin/Journal_uploads/JARET/VOLUME_1_ISSUE_2/JARET_01_02_004.pdf)

## 1. Introduction

The in-vehicle software ecosystem of hardware, software, and services is changing at a fast pace because of the proliferation of software-defined vehicles (SDVs) that facilitate software-based updates and improvements. Software-defined vehicles that use software for innovation and ongoing development are known as in-vehicle software (SDVs) [1]. They provide benefits such as dynamic functionality improvement, over-the-air (OTA) updates, and new features without needing hardware changes. The Vehicle Operating Systems (Vehicle OS) regulate vehicle functions and run different applications. The Application Layer is comprised of apps and services that interact with the features of the vehicle, including infotainment, networking, and navigation. Middleware enables data exchange and communication among the components. SDVs are getting more cloud-connected, using over-the-air upgrades, real-time information, and other cloud resources. Embedded Systems control some automotive functions, including brakes, engine control, and safety features. The functioning of the car relies on the underlying hardware platform, such as CPUs, sensors, and communication modules [2].

The car industry is transforming from hardware-defined to software-defined cars, where software is specifying features and functionality. Cloud-based services are advancing connectivity and data gathering, with open innovation platforms coming out for external developers to develop services and apps for automobiles. In-vehicle software development needs collaboration and ecosystems, with OEMs, software providers, and technology providers cooperating to design efficient and user-centered solutions [3].

The vehicle software ecosystem provides various advantages such as ongoing innovation, enhanced user experience, increased safety and efficiency, and new business models. Software updates enable new features and capabilities to be added, and software-defined features and apps enable a greater variety of entertainment choices and personalization options. Software can also improve vehicle performance, fuel efficiency, and safety features [4].

Increased dependence on software in vehicles creates security issues that need to be addressed. Interoperability is essential for smooth integration of hardware and software systems. Scalability is difficult because of the intricacies involved in software development over numerous car models. Talent is essential for the auto industry to keep pace with swift software evolution. Automotive industry's challenges in-vehicle software environment, including hardware, software, and communications technology, is fundamental to next-generation vehicles, entailing research in software development, architecture, quality assurance, and AI. The vehicle software ecosystem in the automotive sector is a sophisticated area with many publications that explore its multiple facets. The research is concentrated on software architecture, development process, quality assurance, and the effect of sophisticated technology such as artificial intelligence and connected car services, Agile development practices, and master-slave software product line expansion.

It encompasses software product line development, AUTOSAR (Automotive Open System Architecture), embedded systems, software quality management, connectivity and networked vehicles, data analytics, and software-defined vehicles.

Agile methodologies are employed to oversee software development throughout vehicle generations, while AUTOSAR standardizes the software development process and maintains interoperability. Embedded systems contribute to making cars more autonomous, connected, and sustainable [6]. Software quality management delivers quality automotive software by way of a rigorous integration, testing, and verification process. Wireless communications and connected car services also impact the automotive industry. A Digital Vehicle Platform (DVP) is a digital platform that supports numerous aspects of automotive operations, including digital twin development, design verification and validation, and vehicle e-commerce.

A DVP is a critical document that outlines the tests, order, and acceptance standards for a great product [5]. It also encompasses creating a digital twin of a vehicle, an internet replica which replicates its performance, behaviors, and physical attributes. It is used for simulation, testing, and optimization. For the automotive market, a DVP can be an electronic vehicle

portfolio or online store that helps dealerships manage inventory, simplify sales processes, and provide an improved online experience. Alternatively, DVP could refer to any electronic platform utilized to facilitate vehicle activities, such as data analysis, over-the-air software updates, and connected vehicle capabilities. A Digital Vehicle Platform (DVP) is a critical design verification component in the automotive marketplace.

It defines the specific tests to verify the design, environment, tools, and procedures of the tests, test acceptance criteria for the tests to be passed, and a report on the test outcomes, identifying any problems or areas for optimization. "Digital vehicle platform" can be utilized to define comprehensive design verification plans or common digital solutions to vehicle processes [5].

## 2. Methodology

The dashboard software environment of the car is adopting a software-defined approach, particularly Digital Vehicle Platforms (DVPs), that facilitates regular updates and improvement of vehicles. DVPs provide a platform for coordinating and managing software components such that over-the-air upgrades, personalization, and networking are feasible. The Digital Vehicle Platform (DVP) serves as the basis for software development, deployment, and management in a vehicle. The Vehicle Operating System (VOS) supports multiple vehicle applications and capabilities, including AUTOSAR Classic, AUTOSAR Adaptive, and Linux. Software modules or applications do specific tasks, for example, entertainment, safety functions, or connection services. Connectivity enables the vehicle to talk with other vehicles, infrastructure, and the cloud. Over-the-Air (OTA) updates enhance performance and repair issues over the air. Data handling entails collecting, processing, and applying car data to analyze, personalize, and diagnose. Cloud capabilities make more aspects and services possible. Software-Defined Vehicles (SDVs) is a more holistic way of designing cars, in which software is critical to define and make vehicle capabilities possible [6].

A DVP-Powered in-Vehicle Software Ecosystem provides a number of benefits such as ongoing improvement, enhanced connectivity, customized experiences, new business models, accelerated innovation, lower costs, greater safety, and greater automation. Vehicles are able to talk to the world around them, allowing more services and capabilities. Software-defined vehicles also enable quicker creation and deployment of new capabilities, improving flexibility and responsiveness to market demands. Remote software updates reduce maintenance expenses, and advanced safety functions enhance passenger and driver safety. Further,

software-defined vehicles can provide support for various levels of automation, ranging from driver-aid programs to full autonomous driving [7].

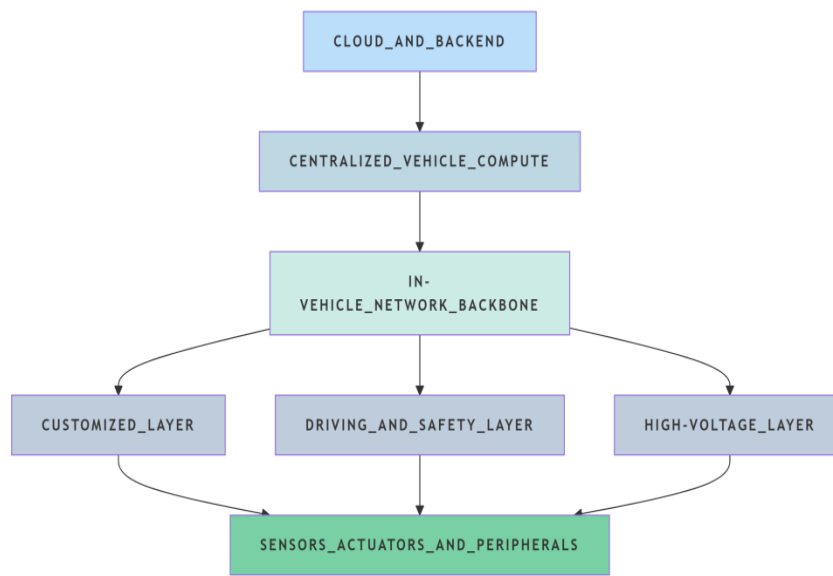
Examples of implementations of DVP are QNX's Connected Vehicle Platform that enables real-time sharing of data and smart traffic lights, Tata Elxsi's TETHER AUTO which is a cloud platform for navigation, position tracking, and auto care, Bosch's Vehicle Integration Platform which connects various domains, and Sonatus Vehicle Platform which provides a platform for AI-based services and features in vehicles. These platforms enable the vehicles to communicate with the world outside [7].

Design Verification Plan (DVP) is a vital element of vehicle engineering development. It entails methodically testing and verifying vehicle systems to guarantee they are safe, meet performance requirements, and comply with legal standards. DVP ensures that problems are detected early, meaning time and cost are saved. Mazda uses DVP to guarantee that new models comply with safety standards before production starts, meaning new models comply with safety standards [8]. Automotive Retail's Digital Vehicle Portfolio (DVP) is an online vehicle history report that includes inspection reports, warranty details, and accident history. The information is useful to customers when making a purchase decision, making the dealerships more transparent and trustworthy to potential customers.

Software-defined cars are being developed on varying digital platforms. The In-Vehicle Red Hat Operating System is an open-source operating system that provides a stable and flexible operating system for rapid development. It is compatible with top companies such as NXP Semiconductors and LG Electronics. Qualcomm Snapdragon Digital Chassis features automotive technologies such as infotainment, connectivity, and driver assistance. Top automakers such as General Motors, Stellantis, and Mercedes-Benz utilize this platform for enhancing the digital capabilities of their vehicles. The new-generation RAV4 will also come with Toyota's in-house-developed Arene operating system by March 2026 with the objective to provide a foundation for future autonomous driving functions. These platforms are a part of the industry trend towards integrating advanced software ecosystems into vehicles to make vehicles more usable, safe, and useful [8].

The Digital Vehicle Platform (DVP) is an architecture for developing, testing, and deploying software-defined vehicle (SDV) functions. Its scalable and modular architecture enables autonomous development and introduction of new features by means of software updates, frequently distributed over-the-air. Centralized and cross-domain E/E architecture consolidates the vehicle functions on common high-performance processing units. Open

standards and interoperability are applied in order to enable collaboration among software developers, OEMs, and Tier-1 suppliers to minimize vendor lock-in. Cloud-native technologies and DevOps practices are enabled to support continuous software testing, validation, integration, and deployment, which decrease the time-to-market and accelerate innovation cycles [3]. DVP implementation process is a vital process comprising various major factors as shown in the below Figure 1:



**Figure 1:** Architecture of DVP Process

### 1. Cloud & Backend Layer:

The highest layer is responsible for software development and deployment, including OTA updates, CI/CD, and AI-powered data analytics. It maximizes the performance of the vehicle and user experience through real-time monitoring, data-driven insights, remote management, and software updates. The cloud backend facilitates scalability and speed of innovation cycles in software-defined vehicle features.

### 2. Centralized Vehicle Compute

The vehicle security feature involves Secure Connected Gateways (SCG) and domain controllers that are secure and resilient, and manage operating systems, middleware, and vehicle applications. They are centralized computer units that can dynamically share resources between domains or applications, deploy modular software, and secure data integrity and vehicle communication. Computing resource centralization simplifies design, increases processing power, and supports software-defined features like real-time data processing and OTA updates.

### 3. In-Vehicle Network Backbone:

The network topology features fast data communication buses like CAN and Automotive Ethernet, linking control units, sensors, and central calculation units. Power & LAN Boxes (PLB) provide scalable and modular wire harnesses, handling network connectivity and power distribution within vehicle zones. Zone controllers (Power Data Centres) handle power and data transfer within allocated vehicle zones, enhancing data speed and system security while reducing wire complexity, weight, and cost.

### 4. Three-Layer Vehicle Architecture:

The three-layer vehicle structure includes Customised Layer, Driving and Safety Layer, and High-Voltage Layer. The Customised Layer is used for adaptable communication and power distribution to minimize the impact on the wiring and design of the vehicle. The Driving and Safety Layer offers the operation of safety-critical systems like ADAS and autonomous driving units with redundant power and communication routes to guarantee safety. The High-Voltage Layer manages the battery systems, charging networks, and high-voltage power supply with a focus on electric vehicle power management.

### 5. Sensors, Actuators & Peripherals:

This bottom level consists of physical interfaces such as cameras, LiDAR, radar, infotainment platforms, and ADAS/autonomous driving modules, which provide perception, decision-making, and actuation functions for vehicle operation and user experience by offering critical data inputs and executing control commands from centralized compute platforms.

The firm employs a Digital Vehicle Platform approach to ensure a uniform, modular, and updatable software foundation across its range of vehicles that allows for over-the-air updates and adding new capabilities continuously without impacting hardware controls to support multiple facets such as entertainment, ADAS, and EV drive. The most important Digital Vehicle Platform (DVP) used as [9]

- Unification and Modularity of Software Platform: GM is creating a software platform that can be compatible across different car models, allowing for feature integration and scalable upgrades.
- Distributed Oecus with Centralised Compute: The platform architecture keeps latency-critical controls on localised oecus but leverages centralised compute resources for high-level computations.

- **Quality Assurance and Advanced Testing:** GM has set up a software quality lab with closed-loop hardware-in-the-loop (HIL) testing benches for automated testing along with remote access for developers.
- **OTA Updates:** The DVP's ability to provide OTA updates facilitates software delivery in all vehicle oecus reliably.
- **High-Bandwidth Electrical Design:** The platform's advanced electrical design supports up to 4.5 terabytes of data per hour.
- **Integration of AI and Simulation Tools:** GM and NVIDIA work together to utilize digital twins, AI, and simulation tools to test automobile software and manufacturing processes virtually.
- **Software as a Service (SaaS) Model:** GM's Vehicle Intelligence Platform provides subscription-based software offerings such as Super Cruise for continuous feature enhancements and other post-purchase services.

### 3. Challenges and Solutions

Integrating software from diverse sources into contemporary vehicles poses a number of challenges. These consist of integrating control systems complexity, components and protocol interoperability, safety and regulatory compliance, co-design and testing complexity between software and hardware, data security and privacy concerns, cost and complexity management.

In order to simplify integration, implement modular software designs such as microservices and containerization, offer uniform interfaces and layers of communication with the aid of middleware frameworks, automate deployment and testing with the use of trusted CI/CD pipelines, and manage streaming data successfully through advanced data handling technologies. Foster interdisciplinary collaboration among cybersecurity, software, hardware, and quality assurance teams in coordinating integration activities [9].

Another problem is interoperability between components and protocols since cars contain parts from various manufacturers, some with proprietary standards and protocols. To solve these, promote the use of industry standards for software and safety certification, modularize communication and power distribution through zone-based architectures, deal with protocol incompatibilities openly through middleware and secure gateways, and facilitate

software interoperability and mask hardware differences through standardized APIs and interfaces.

Regulatory compliance and safety are also important concerns with varying car regulations across the world leading to a fragmented compliance framework. To facilitate fail-safe and fail-operational functionality in autonomous modes and software updates, engage regulatory bodies early, employ strict validation plans, incorporate security by design principles, and employ OTA update mechanisms with robust authentication and rollback [10].

Data privacy and security are also important, as connected vehicles are susceptible to hacks and preserve data privacy while complying with regulations such as GDPR. Adopt secure communication practices, deploy hardware security modules, keep software up to date, and conduct periodic vulnerability analysis and security monitoring.

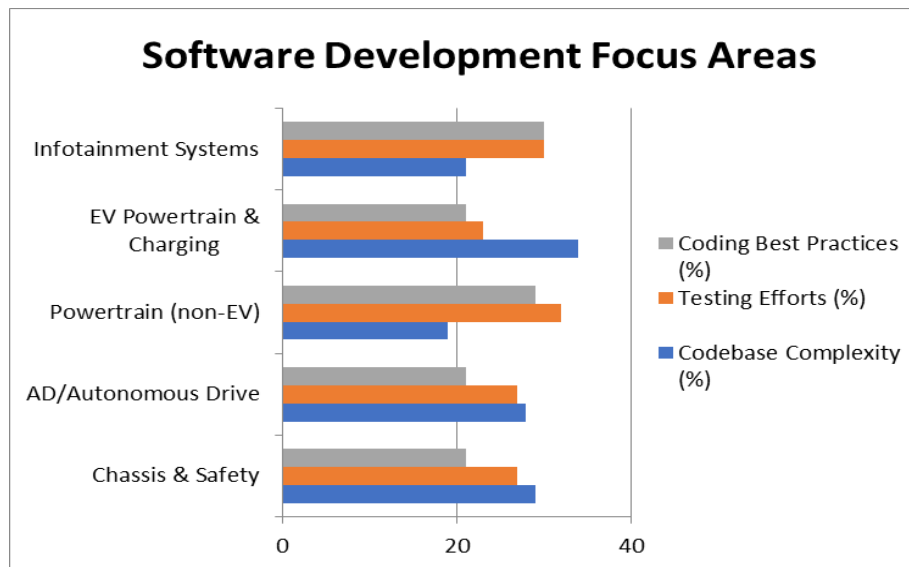
The complexity of the codebase of software in various domains depends on variables like algorithmic complexity, dependencies, modularity, and line count. The EV Powertrain & Charging domain has the highest complexity (34%), followed by AD/Autonomous Drive and Chassis & Safety because of advanced sensing and decision-making algorithms and safety-critical real-time control. Infotainment and powertrain (non-EV) have relatively lower complexity, indicating better established or less safety-critical software stacks [11].

Testing activity is most intense in infotainment and powertrains (non-EV) (32% and 30%), conceivably because of expectations from user experience, regulatory demands, and back-end requirements. Chassis & Safety and AD/Autonomous Drive need most testing (27%). EV Powertrain & Charging has a slightly lower testing activity (23%).

Best coding practices are evaluated, where the greatest compliance rates (30% and 29%, respectively) exist in infotainment systems and non-EV powertrains. The areas of AD/Autonomous Drive and Chassis & Safety have poor compliance (21%), perhaps because of real-time limitations and autonomous driving software complexity. EV Powertrain & Charging indicates 21%, indicating ongoing attempts to increase the coding bar in fast-evolving EV software stacks [11].

Safety-critical areas, like AD and chassis, have high testing effort and complexity because of strict regulation. Infotainment and non-EV powertrain have user experience, security, and system reliability as top priority, while legacy systems have low complexity but high coding and testing standards. Variations in coding best practices, testing effort, and

complexity across automotive domains are indicated by the data from "Software Development Focus Areas by Automotive Domain" as shown in below Figure 2:



**Figure 2:** Software Development Focus Areas by Automotive Domain

The 2025 Automotive Software Development report lists a number of key performance indicators for software development. The key issue is quality, mainly because of the complexity of the software, inadequate testing, and application of best practices. Safety is an emerging issue as a result of AI and the necessity of robust testing of autonomous driving software. Security is paramount with emphasis on OTA update protection and secure code. Productivity of teams is problematic, with problems in managing distributed teams, resources, and hardware and software coordination. Testing is a problematic area, with problems in long QA cycles and rigorous testing impacting release dates [10].

Four key delivery performance metrics are identified as follows: Automobile Adaptation Metric, Cycle Time, Development Velocity, Testing Coverage and Automation Rate, and Complexity and Maintainability of Code Metrics. High-performing teams will normally perform tasks within 1.8 to 3.4 days, whereas development velocity supports predictability of delivery dates. Testing coverage and automation rate minimize QA cycle times, and complexity and maintainability of code metrics determine codebase complexity.

It is very focused on safety and quality, with significant investment in software quality organizations and robust validation, such as digital twin simulations and HIL. Reliability and frequency of OTA updates are significant KPIs, and wants to reduce change failure rates while

growing deployment frequency. Productivity metrics for teams are paramount, and automation of testing and shrinking cycle times are ongoing priorities in order to speed up feature delivery and maintain safety requirements.

**Table 1: Metrics for Assessing and Enhancing Automotive Software Development**

| Metric                                     | Automotive Use                                                | Target / Benchmark                                                     |
|--------------------------------------------|---------------------------------------------------------------|------------------------------------------------------------------------|
| Concern about Quality                      | Complexity of the code and thoroughness of testing            | Constant progress, less than 5% of serious flaws after the release     |
| Safety Issue                               | Observance of ISO 26262, confirmation of AI safety            | There were no serious safety incidents.                                |
| Modify Lead Time                           | It's time to confirm that the test environments have changed. | Weeks to days, getting better with automation                          |
| Regularity of Deployment                   | OTA update frequency                                          | Updates every month or every three months, growing over time           |
| Modification Failure Rate                  | Failed deployments or updates                                 | <5% of upgrades or deployments fail, with the opportunity to turn back |
| Transitional Period for                    | Recuperation After failure, it's time to repair and redeploy. | Days to hours, based on the severity                                   |
| Time of Cycle                              | Speed of task completion                                      | 1.8–3.4 days per assignment                                            |
| Development Velocity                       | Each sprint's worth of work                                   | varies by team; trend analysis is monitored.                           |
| Coverage of Testing For important systems, | automated test coverage                                       | at least 70–80% is desirable.                                          |

In table 1 above, the range of metrics offers a complete set of guidelines for evaluating and improving automotive software development to ensure platforms such as GM's DVP deliver reliable, continually changing car software [11].

#### 4. Conclusion

Digital Vehicle Platform (DVP) is transforming automotive engineering by substituting traditional hardware-based vehicles with software-defined, networked, and end-to-end-upgradeable platforms. It employs cloud integration, modular zone-based electrical and network structures, and centralised computing to offer scalable, secure, and flexible vehicle software environments. It encompasses enhanced software quality and safety, seamless over-the-air (OTA) upgrades, modular and scalable architecture, robust cybersecurity, data-driven innovation, and software quality, safety, and delivery performance improvements.

DVP and affiliated automotive software ecosystems hold great promise with potential for significant breakthroughs. Centralized computing growth and domain consolidation will make fully software-defined vehicles (SDVs), sophisticated autonomous driving integration,

synergy between edge and cloud, enhanced security and privacy models, sustainability and electronification, open platforms and ecosystem collaboration, personalized user experience and services, possible.

The dedication to leading the revolution of software-defined vehicles is supported by progress metrics, which reflect continued improvements in software quality, safety, and delivery performance. Future development of the platform entails the realization of fully software-defined vehicles, advanced autonomous driving integration, synergies between edge and cloud, enhanced security and privacy frameworks, sustainability and electronification, open platforms and ecosystem collaboration, and personalized user experiences and services.

## References

- [1] “The Self-Driving Car Timeline – Predictions from the Top 11 Global Automakers”, Dan Faggella, May 29, 2017.
- [2] “What are the best ecosystems for automotive software development?”, Joseph Sibony, November 8, 2022.
- [3] “An approach for design Verification and Validation planning and optimization for new product reliability improvement”, Mohammadsadegh Mobin, Zhaojun Li Steven, S. Hossein Cheraghi, Gongyu Wu b, Elsevier, Reliability Engineering & System Safety, Volume 190, October 2019, 106518.
- [4] “Future In-Vehicle Electronic Platform”, Hiromichi YASUNORI, Hideaki UEMURA, Akio ISHIHARA, Shinya ITO, and Yutaka MATSUMURA, Future In-Vehicle Electronic Platform, SEI TECHNICAL REVIEW, APRIL 2019.
- [5] Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018), “Digital Twin in manufacturing: A categorical literature review and classification”, IFAC-PapersOnLine, 51(11), 1016-1022.
- [6] Amorim, T., Martin, H., Ma, Z., Schmittner, C., Schneider, D., Macher, G., ... & Kreiner, C. (2017). “Systematic pattern approach for safety and security co-engineering in the automotive domain”, 329-342. [https://doi.org/10.1007/978-3-319-66266-4\\_22](https://doi.org/10.1007/978-3-319-66266-4_22).
- [7] Assessment of Competitive Edge of Major Global Semiconductor Vendors for Self-Driving Solutions (Level 3 and Above)- Evaluation of Qualcomm, Intel, and Nvidia Jae-Kyung Kim,

Jon-Mo Yoon, Bong-Soo Lee<sup>3</sup>, Asia-Pacific Journal of Convergent Research Interchange, Vol.6, No.10, October 31 (2020), pp.165-180.

- [8] “AVL Integrated and Open Development Platform” , AVL <https://www.avl.com/en-in/integrated-and-open-development-platform>.
- [9] “GM’s Vehicle Intelligence Platform and the Prospects of Software as a Service”, Glenn Sanders, New digital vehicle platform enables OTA updates and subscriptions, 07 May 2021, Article.
- [10] “Challenges and Solutions for Applications and Technologies in the Internet of Things”, Saad Albishi, Ben Soh, Azmat Ullah, Fahad Algarni, Elsevier, Procedia Computer Science, Volume 124, 2017.
- [11] “Computer Architecture - Performance Metrics”, Dr A. P. Shanthi, <https://www.cs.umd.edu/~meesh/411/CA-online/chapter/performance-metrics/index.html>.