© NOV 2020 | IRE Journals | Volume 4 Issue 5 | ISSN: 2456-8880

Enhancing Software-Defined Vehicles with Service-Oriented Architecture: A Framework for Scalable and Modular Mobility Solutions

ANKUSH KESKAR

Abstract- The new paradigm in the automotive industry is Software Vehicles (SDV), the concept where Software is used to realize functionalities rather than Hardware with the functionalities. The service-oriented architecture (SOA) framework evolved from existing SOA practices and enables modular, scalable, and adaptive vehicle systems. In this article, the basic principles of the SOA and its application in the SDVs are being reviewed, revealing how they can increase flexibility, allow for fast iterations, and decrease productivity. We can see what SOA in mobility can do in real-world implementations, such as Tesla's software ecosystem and autonomous driving systems. However, SOA adoption has issues with legacy system integration, cybersecurity, and data complexity. With new standardization efforts, improved middleware, and new security solutions addressing these barriers, the industry can unlock the potential of SOA-based SDVs. Last but not least, the integration of SOA within SDVs will serve as the seeds of tomorrow's automated, connected, and green intelligent mobility systems for the worldwide revolution of transportation.

Indexed Terms- Software-defined vehicles (SDVs), Service-Oriented Architecture (SOA), Modular mobility solutions, Over-the-air (OTA) updates, Autonomous driving systems, Connected vehicles

I. INTRODUCTION

We are at the edge of an automotive transformation where the role of the vehicle is increasingly driven by software. This trend directly results from the new concept of software-defined vehicles (SDV), which focus not on the traditional mechanical pieces but heavily rely on their sophisticated software systems. Innovation is no longer subject to the physical constraints of hardware with SDVs. Rather, it is now driven by the flexibility and adaptability of software.

For SDVs to reach their full potential, however, adopting SOA has become necessary. An architectural style, SOA is the art of structuring software as modular, reusable, and coarse-grained services. This enables automotive systems to scale and react fast to changing requests in an integrated and robust way with minimum overhead. The interplays between SDVs and SOA deliver the basis of scalable, modular, and future-proof mobility solutions.

1.1. Evolution of Software Defined Vehicles (SDVs)

Software-controlled systems are not a new concept, but as they apply to vehicles, how they are realized has changed greatly. Since the early 2000s, many cars have been equipped with basic software systems for handling information and engine control. Yet these systems were typically isolated, acting as earmarked silos.

Fast forward to today, and SDVs are a revolution where every aspect, from the driving dynamics to connectivity and user interfaces, is based on software. The industry is evolving because of advancements in computing power, cloud technologies, and increasing demand for connected, autonomous, and electrified vehicles. Unlike traditional cars, which leverage heavy static hardware, SDVs can be updated over the air (OTA), similar to a smartphone, so continuous innovation occurs without needing physical change.

1.2. The Role of Service-Oriented Architecture (SOA) in SDVs

SOA has emerged as a cornerstone for enabling SDVs to achieve their full potential. The vehicle software has traditionally been monolithic, requiring major overhauls, more time, and cost. SOA shatters the monolithic structures into smaller, independent services that communicate perfectly.

For example, in an SOA-enabled SDV, the system controlling navigation can operate independently while sharing data with other services, such as the system managing battery performance. Based on this modularity, auto manufacturers rapidly only need to introduce new features, enhance features, or respond to customer needs with changes to certain submodules in the system, not affecting the entire vehicle ecosystem. Furthermore, versioning, such as SOAP, enables interoperability among various software vendors.

1.3. Why We Need Scalability and Modularity in Modern Mobility

With an automotive landscape that is changing rapidly, scalability and modularity are no longer nice to have; they're a must have. Contemporary mobility solutions must cover diverse user needs, technological progress, and regulatory requirements. Scalability allows for the extension of systems to absorb greater volumes or includes additional functionality with few disruptions. For example, as autonomous driving technology matures, scalable architectures will allow increased computation that may support more sophisticated algorithms without re-engineering the entire vehicle.

Similarly, modularity allows automakers to build vehicles like "software LEGO sets," where individual components can be upgraded or replaced independently. This approach is extremely important to bring down the costs of development and time to market as it gives customers highly customizable solutions. Scaling and modularity help automakers remain competitive and offer superior user experiences while building for an era of software-driven mobility.

II. SOFTWARE-DEFINED VEHICLES?

Software-defined–Vehicles (SUVs) are revolutionizing the automotive industry through software, not hardware. Unlike traditional vehicles, which rely on mechanical and static electronic systems, SDVs rely on software to control, improve, and update their functions. Like what happened in the tech industry, this is essentially a transformation that we have seen in the past; a device like a modern smartphone gets better and better through software updates, allowing for new features without having to change its hardware.

Fig 1. Software Defined Vehicle

2.1. Overview of SDVs and Their Components

At their core, SDVs consist of two main pillars: a hardware platform and a software-driven system. The hardware is the foundation, including processors, sensors, and basic mechanical elements that facilitate essential vehicle operations. But the real innovation is in the software layer. This software deals with vehicle dynamics, infotainment, connectivity, and even highly advanced driver assistance systems, such as ADAS. Centralized computing also elegantly replaces the numerous fragmented electronic control units used in today's cars, enabling SDVs and their software to work smoothly and empowering its software to support over-the-air (OTA) updates for the continuous improvement cycle.

Another key component of SDVs is their connectivity framework. Integration of cloud computing and real-time data exchange within SDV upholds higher unity of communication between vehicles, infrastructure, and external networks. This connectivity enables vehicles to update, download new features, and interact with other smart devices or systems on the road, all of which pave the way for technologies such as Autonomous driving and Smart Traffic management.

2.2. Advantages of SDVs Over Traditional Vehicles

SDVs offer several advantages over conventional vehicles, primarily from their reliance on software. The biggest benefit is that you can update the vehicle's capabilities remotely. OTA updates allow you to download new features and software fix bugs, without visiting a dealership. Additionally, it is convenient for production and users, and lower your expenditures.

Another advantage lies in their adaptability. One thing that SDVs can integrate with new technologies (for example, artificial intelligence or machine learning algorithms) is to enhance their performance, safety, and user experience. As an example, an SDV with ADAS features undergoing continual evolution can become progressively more effective in dealing with sophisticated driving situations through improved software packages.

Moreover, SDVs support enhanced customization. Automakers can tailor features to suit individual customer preferences, such as optimizing navigation systems or adjusting vehicle settings for specific driving habits. This level of personalization is difficult to achieve in traditional vehicles with fixed, hardware-dependent systems.

2.3. Examples of SDVs in Today's Market

Several automakers have already introduced SDVs to the market to demonstrate what software-driven mobility is capable of. A prime example is Tesla, which has vehicles with much control through software, such as autonomous driving, OTA updates, configurable driving settings, etc. For example, the SDVs concept has been realized as General Motors' Ultifi platform, which reflects the idea and is built to allow for modular software applications to enhance connectivity and functionality.

Volvo and Polestar also use Google's Android Automotive OS to deliver over-the-air software-driven infotainment and connectivity services. Additionally, newcomers like Rivian and Lucid Motors manufacture electric vehicles with superior software digital ecosystems to compete with incumbent automakers.

SDV is gaining traction quickly, and we see tomorrow's mobility being offered through scalable, software-centric frameworks. This shift means that vehicles are not machines anymore—they are becoming intelligent, adaptive mobility, and far more platforms.

Table 1. Key Differences between Traditional Vehicles and SDVs

Feature	Traditional Vehicles	Software-Defined Vehicles (SDVs)
Functionality Control	Primarily hardware-driven	Software-driven
Update Mechanism	Physical intervention required	Over-the-air (OTA) updates
System Integration	Isolated systems (ECUs)	Unified, modular architecture
Scalability	Limited	Highly scalable
Customization	Minimal	Extensive

III. UNDERSTANDING SERVICE-ORIENTED ARCHITECTURE (SOA)

Service oriented architecture (SOA) architecture style is a software design strategy in which the functionalities are organized into modular services in order to fulfill an individual task. This architectural framework of reusability, flexibility, and interoperability is an excellent counterpart to systems continually requiring updates and scalability. The principles championed by SOA have been widely adopted across all industries, and their incorporation into the automotive space is helping transform the way vehicles work and make changes.

3.1. Key Principles of SOA in Software Systems

Several fundamental principles define its effectiveness in software systems built based on SOA. Modularity is at its core, where functionalities are broken down into elegant, small, independent services that can be developed, tested, and deployed independently. The services are loosely coupled, i.e., they interact through well-defined interfaces and remain autonomous – changes to any one service should not affect the other services.

The second important principle of servicing is interoperability: the ability for services to talk to each other freely and without fail between different platforms or technologies. This is typically done by using standardized protocols such as REST or SOAP. Moreover, SOA promotes reusability because developers can reuse existing services in one enterprise application on other enterprise applications, hence cutting costs and time.

Furthermore, to maintain scalability, services of this type are structured to meet the growing demands by adding or changing other services. Finally, centralized management guarantees that all services meet the organization's needs while providing consistency and reliability.

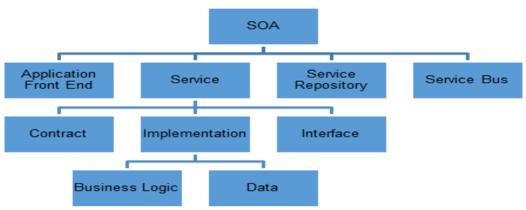

Fig 2. Service Oriented Architecture

3.2. Applying SOA to the Automobile Industry
SOA has emerged as an answer to the increasing complexity of software-defined vehicles in the automotive industry. Functions such as braking, steering, and infotainment for traditional cars are controlled in silos through traditional control units. SOA fills in the gap by driving the approach in a unified way with a service approach instead of treating each function as independent.

For instance, in an SOA-themed vehicle, the navigation system, battery management, and other driver assist services are moduled services running in a common environment. These services can communicate with each other, share data, and even update individually without breaking the whole system.

SOA also allows automakers to easily integrate with third-party applications, enabling them to work with technology providers to add vehicle functionality. With connected and autonomous vehicles, automatic innovation of automotive systems depends on SOA's ability to manage distributed systems efficiently.

3.3. Benefits of SOA for SDVs
SOA brings numerous benefits to software-defined vehicles, starting with enhanced flexibility. Automakers can quickly adapt to market trends or customer needs by deploying new services or updating existing ones without significant delays or costs.

Another advantage is cost-effectiveness. Since SOA emphasizes reusability, manufacturers can repurpose software components across different vehicle models, reducing redundancy and development expenses. SOA also ensures faster updates, as individual services can be modified or improved without disrupting the entire vehicle ecosystem.

SDVs use SOA's interoperability to communicate naturally between in-vehicle systems and various external networks, such as cloud platforms or smart infrastructure. This capability enables advanced features such as predictive maintenance, real-time traffic management, and tailored user experiences.

IV. THE INTERSECTION OF SDVS AND SOA

Software-defined vehicle and service-oriented architecture convergence is the new change that is rapidly sweeping through the automotive industry. SDVs are naturally difficult because of their open-ended nature, and SOA can address this challenge by drawing attention to modularity, scalability, and continuing evolution.

4.1. How SOA Enables Modularity in SDVs
One of the key capabilities of SDVs is modularity, and SOA is an enabler of this. In an SOA-based SDV, each service is contained within a single function, such as lane-keeping assist, climate control, or voice recognition. This modularity allows automakers to add, remove, or upgrade specific services without re-engineering the entire system.

For instance, an automaker can introduce a new parking assist feature as a standalone service. This doesn't affect existing services, such as steering or braking ones. Its flexibility speeds up the development cycles and allows designers and manufacturers to customize the vehicles for other markets or users' preferences.

4.2. Scalability through Service-Based Systems in Mobility Solutions

Scalability is essential for SDVs to handle the increasing complexity of modern mobility solutions. SOA's service-based architecture allows automakers to scale their systems by adding or expanding existing services.

For instance, as vehicle autonomy increases, SOA will enable the addition of more sophisticated algorithms or additional sensor streams as needed, all without having to rebuild the entire vehicle software ecosystem. Cloud-based services, like remote diagnostics or real-time traffic updates, can be easily scaled to satisfy the growing number of users or vehicles. This scalability guarantee ensures that SDVs are future-proof and easily adapt to changing demands.

4.3. Enhancing Functionality with Microservices Architecture

A microservices architecture is an extension of SOA that focuses on breaking down services into even smaller, highly specialized components. This approach enhances functionality by enabling SDVs to deliver precise, efficient, scalable solutions for complex tasks.

For instance, microservices can manage specific tasks such as obstacle detection, path planning, or speed control in an autonomous vehicle. The microservices run separately but come together as part of interfacing with other microservices. It's a separation that helps keep failures in one service from taking down the whole system, making it more reliable and maintainable.

SDVs paired with SOA are a good framework for developing intelligent, connected, and even adaptive mobility solutions when microservices are used. This has created a synergy that has completely reshaped the automotive industry, propelling innovation and setting the stage for a new software-driven transportation future.

V. CORE COMPONENTS OF A SERVICE-ORIENTED SDV FRAMEWORK

The project builds a robust structure for a Service Oriented Architecture (SOA) framework of the Software Defined Vehicles (SDVs) system, allowing for service scalability, modularity, and communication between components. The features of this architecture permit the SDVs to behave as intelligent adaptive systems capable of adding enhanced features and developing through time. The key elements of this framework include service layers, middleware, and a system for data communication and interoperability.

5.1. Service Layers and Their Functionality

Service layers form the foundation of an SOA-based SDV framework, organizing functionalities into distinct, reusable modules. Each layer addresses a specific aspect of the vehicle's operation, enabling efficient management and integration of complex systems.

The presentation layer handles user interaction and managing interfaces like infotainment displays, voice controls, or mobile app integrations. This layer ensures seamless communication between the user and the vehicle.

Core vehicle functionalities like navigation, driver assist systems, and climate control live on the application layer. These services are decoupled to work independently, so the updates or changes on any service shouldn't affect the whole system.

The infrastructure layer at the base manages the hardware resources (sensors, processors, and communication modules) situated at the base. It provides the computational power and connectivity needed to support higher-level services.

By segmenting these layers, the framework ensures modularity, enabling automakers to develop, test, and deploy new services with minimal disruption to existing systems.

Table 2. Service Layer Functions in SOA-Based SDVs

Service Layer	Functions

Presentation Layer	- Manages user interfaces (e.g., infotainment displays, mobile apps) - Facilitates user interaction through voice controls and touch interfaces - Ensures seamless communication between users and vehicle systems
Application Layer	- Hosts core vehicle functionalities (e.g., navigation, climate control, driver assistance) - Allows independent updates and modifications of services - Integrates third-party applications for enhanced features
Infrastructure Layer	- Manages hardware resources (e.g., sensors, processors, communication modules) - Provides computational power and connectivity for higher-level services - Ensures reliable data transfer between various system components

5.2. Middleware in SOA-Based SDVs
The SOA based SDV framework is also supported by middleware, the connective tissue that allows you to communicate by integrating hardware, software, and external systems. It presents a unified interface to services regardless of the technologies they run on; the trick is that the services can talk to each other.

In SDVs, middleware is essential for managing resource allocation, data routing, and security protocols. For example, it allows the vehicle's navigation system to access real-time traffic data from external servers while simultaneously communicating with the ADAS for route optimization.

Middleware also supports abstraction, hiding the complexity of underlying hardware or communication protocols from application developers. This enables faster development cycles and ensures compatibility across different platforms, making it a cornerstone for innovation in SDVs.

5.3. Data Communication and Interoperability
Efficient data communication is crucial for the operation of SOA-based SDVs. That necessitates real-time information exchange between services within the vehicle (e.g., sensor data, user inputs, and cloud-based updates). This is possible only with the help of a framework based on standardized communication protocols like RESTful API or MQTT, which guarantee reliable data transfer between the services.

The other significant aspect is interoperability, allowing SDVs to talk to other systems, such as smart infrastructure, cloud platforms, or other vehicles. For example, an SOA-enabled SDV can communicate with the city's traffic management system to choose a route that optimizes traffic congestion data.

This interoperability also supports over-the-air (OTA) updates, enabling automakers to roll out new features, debug bugs, or improve performance without touching the vehicle. The framework lays the road map for smooth data communication and interoperability, helping enable advanced connected and autonomous mobility solutions.

VI. ADVANTAGES OF SOA IN THE SOFTWARE-DEFINED VEHICLES.

Adopting Service-Oriented Architecture (SOA) in SDVs has many benefits, including improved flexibility and cost savings. SOA is organized as modular services that provide the capabilities of vehicular adaptability, evolution, and efficiency in a fast-changing technological landscape.

6.1. Better adaptability and flexibility
One of the biggest advantages of SOA is that the approach is flexible. The flexibility of SOA allows automakers to change or replace services individually without influencing the entire vehicle system. Flexibility is also greatly valued in the highly dynamic automotive market, in which new traits and innovations are frequently rolled out.

It is easy to add new language into the voice recognition system of an SDV or enrich ADAS functionality of an SDV with minor change in the core SDV. Furthermore, both personalization and region sensibility are supported by this flexibility, allowing automakers to tailor their offerings according to user preferences or region-specific needs.

Furthermore, SOA makes it easy to work together with third-party app developers in a collaborative process that integrates the service developed seamlessly into the vehicle. The open ecosystem helps speed up the pace of innovation and broaden the available features to the consumer.

6.2. Faster Updates and Upgrades

Implementing new features or fixing software issues with traditional vehicles often requires physical intervention, such as visiting a dealership. SOA eliminates this limitation by supporting over-the-air (OTA) updates, enabling automakers to deploy changes remotely.

This capability allows manufacturers to roll out new functionalities, address security vulnerabilities, or refine system performance quickly and efficiently. For instance, an SOA-enabled SDV can receive an upgrade for improved energy management, enhancing battery life for electric vehicles without requiring hardware modifications.

Additionally, by continuously updating holders of the SDVs, they will maintain dimensionality with changing technology and customer needs, increasing lifespans and user satisfaction.

6.3. Cost-Effective Development and Maintenance

SOA significantly reduces the cost of developing and maintaining SDVs. Automakers can leverage existing services across multiple vehicle models or platforms by emphasizing reusability. An example is a navigation service for a vehicle that can be easily adjusted for other vehicles with minimal changes, avoiding the time and resources needed to develop new services for each new car.

Again, this makes maintenance easier, as problems can be located with specific services instead of needing a global reboot. For instance, if a buggy is discovered on the infotainment system, the problem can be fixed without affecting navigation and ADAS.

Furthermore, the modular nature of SOA permits manufacturers to develop according to a phased approach, focusing first on essential features and then slowly supplementing capabilities with one service at a time. It also makes development cost-effective for automakers and consumers as the upfront investment is minimal, and development costs depend on market demand.

SOA represents a transformational way for building and maintaining SDVs and empowers SDVs to be based on something that enhances flexibility, speed, and efficiency required to compete in the lightning-speed pace of technological advancements. In addition to improving vehicle performance, it contributes tangible benefits for manufacturers and users.

Table 3. Advantages of SOA in SDVs

Benefit	Description
Flexibility	Modular services allow quick updates and feature additions.
Faster Updates	Over-the-air updates improve the speed of software deployment.
Cost-Effectiveness	Reusable services reduce development and maintenance costs.
Enhanced Interoperability	Standardized protocols enable seamless interaction between systems.
Scalability	Easily accommodate advanced functionalities or new technologies.

VII. REAL-WORLD APPLICATIONS OF SOA IN SDVS

Service-Oriented Architecture (SOA) has been integrated into Software-Defined Vehicle (SDV) to revolutionize the automotive industry. Automakers have added innovative features with SOA that improve vehicle functionality, connectivity, and user experiences. In the fast-changing world of mobility, real-world examples show how SOA makes things modular and scalable and allows continuous upgrades.

7.1. Case Study: Tesla’s Software Ecosystem

A good example is Tesla, an automaker whose use of SOA principles led to the delivery of features in cutting-edge SDVs. It is built around modular, service-based architectures that seamlessly integrate various functionalities into the company's software ecosystem.

Tesla's cars routinely receive updates over the air, enabling improvement or adding to existing features. Take Tesla's Autopilot and Full Self-Driving (FSD) systems, for example, which are improved with software upgrades that might enhance driver–assist algorithms or better ability to identify lanes. Tesla's software's service-oriented design allows for these updates, as individual modules can be updated without impacting any other system.

Moreover, Tesla's infotainment services (in-car gaming or streaming services) are still independent and integrated into a centralized back end. This modularity allows Tesla to create different levels for the user and maintain a cohesive vehicle ecosystem. Tesla's success in the software-driven approach reinforces the possibility that SOA can revolutionize mobility.

7.2. Autonomous Driving Systems Built on SOA Principles

The complex and distributed nature of autonomous driving functionalities makes SOA completely relied on for autonomous driving services. For ADAS and full autonomy, you need real-time data exchange between different subsystems, such as cameras, sensors, mapping software, and AI-driven decision-making algorithms.

For instance, in an SOA-based autonomous vehicle, the object detection service takes LiDAR and radar sensor data to identify obstacles. In contrast, the path-planning service determines the safest path in real-time when encountered obstacles occur. They can run independently, although they communicate using standardized protocols.

SOA also supports scalability in autonomous systems. Being technology-driven, newer capabilities such as better object recognition or dynamic traffic prediction can be added without revisiting the whole system. The modularity of SOA helps propel the development and implementation of self-driving technologies, and thus, SOA — what we think you will agree — creates a foundational framework for autonomy.

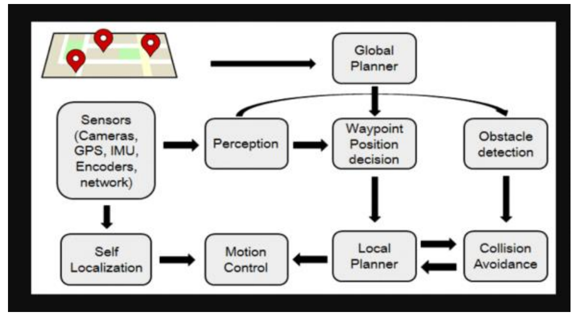

Fig 3. Architecture of Autonomous Driving Vehicle

7.3. Examples from Connected Vehicle Services

Connected vehicle services have also seen significant changes with SOA, enabling vehicle-to-vehicle, infrastructure, and cloud platform communication. Use-based service architecture enables features like real-time traffic updates, predictive maintenance, and remote diagnostics to function efficiently.

For example, BMW's Connected Drive platform is a consistent platform of services based on SOA principles (e.g., navigation, voice control, driving conditions monitoring, etc.). The system combines these services into a unified system and lets them interact dynamically in sync to present a coherent user experience. The General Motors Ultifi platform works similarly, supporting modular applications at the backend and allowing users to download and personalize applications — from advanced parking assistance to personalized infotainment options — on the move.

SOA demonstrates versatility in bringing together third-party applications to ecosystems for connected vehicles. Tech companies can come together with automakers to create innovative solutions to add value to their vehicle offerings.

VIII. SOA IMPLEMENTATION FOR SDVS: CHALLENGES

While the benefits of SOA in SDVs are substantial, implementing this architecture presents several challenges. Automakers must address legacy system integration, security, and data complexity issues to realize the full potential of service-oriented vehicles.

8.1. Integration of Legacy Systems

Many automakers face the challenge of integrating SOA into vehicles with existing legacy systems. Many traditional vehicles follow a monolithic architecture with tightly bound software and hardware. The transition to an SOA-based framework demands significant re-engineering of these systems.

For example, legacy-specific functionally designed electronic control units (ECUs), such as engine control or brakes, may not be compatible with modular service layers. Automakers must spend money on middleware solutions to bridge the gap and make legacy systems talk to new SOA components.

Integration processes can be costly and time-consuming, particularly for manufacturers with large vehicle portfolios based on legacy hardware. However, these challenges can be overcome by gradual migration strategies, like hybrid architectures that mix legacy and service-oriented systems.

8.2. Security Concerns in Service-Oriented SDVs

SOA-based SDVs are inherently linked and rely on external communication networks; security is a major concern. Attackers may use a tactic or attack to compromise sensitive data, disrupt operations, or even cause harm to passengers by launching cyberattacks against vehicles or cloud-based services.

For example, if living in an SOA-based navigation service, an attacker can take advantage of its vulnerabilities to change the routing of vehicles or deactivate some of its functionalities. Strong security requires all services to implement end-to-end encryption, authentication protocols, intrusion detection systems, etc.

The fact that there will be an increasing number of connected vehicles installing third-party applications only magnifies the challenge. Automakers must establish stringent security standards to vet these applications and prevent unauthorized access to the vehicle ecosystem.

8.3. Managing Data Complexity

The service-oriented design of SDVs generates vast amounts of data from sensors, user interactions, and cloud-based services. A major challenge is how efficiently this data can be managed in real-time.

For instance, an automated driving system requires sending huge amounts of exchangeable data between one service, such as object recognition and determination of pathways, to another like a decision support system. The trouble is that such downtimes or delays can potentially injure a system and make it less effective.

This puts pressure on automakers to get on top of their data management approach, real-time such as edge computing soon the vehicle will be able to process data within the car without sending the data to the cloud and insights via cloud-based analytics for longer horizons. Data formats and communication protocols are also standardized to help ensure interoperability and lower processing overhead.

Table 4. Challenges of SOA Implementation in SDVs

Challenge	Description
Integration of Legacy Systems	From a technical perspective, a transition from traditional monolithic architectures to SOA requires substantial re-engineering of existing legacy systems, which is costly and time consuming.
Security Concerns	Hence safety measures for SOA-based SDVs are important because these are particularly vulnerable to cyberattacks since they depend on connectivity and external communication.
Managing Data Complexity	The service-oriented design generates large volumes of data, requiring efficient data management and real-time processing to ensure system performance and safety.

Standardization Issues	Lack of industry-wide standards for protocols and interfaces can hinder interoperability between different manufacturers' systems.
Middleware Development	Creating robust middleware solutions that facilitate seamless communication and resource allocation among diverse services is a complex task.

IX. ADOPTING SOA IN SDVS: OVERCOMING BARRIERS.

Integrating Service-Oriented Architecture (SOA) in Software-Defined Vehicles (SDVs) can provide a more modular, scalable, and efficient way to combine the functionality of a vehicle, but its usage comes with hurdles. This transforms automakers from a technical, organizational, and security challenge that can only be solved collaboratively and innovatively. These barriers must be addressed to harvest SOA's full potential in the automotive industry.

9.1. Efforts in Standardization in the Automotive Industry

One of the main impediments to the introduction of SOA in SDVs is the need for industry-wide standards. Interoperability between manufacturers' systems is still challenging without standardized protocols, interfaces, and architectures. Reducing this fragmentation can slow invention and raise development costs.

The establishment of common standards is already underway. Organizations like the Automotive Open System Architecture (AUTOSAR) consortium are formulating frameworks to introduce SOA into automotive systems. For example, the Adaptive Platform specs of AUTOSAR propose guidelines for developing service-based architectures that run over various software and hardware platforms.

Standardization allows car makers to cooperate better, utilizing shared resources. It makes microservices integration with third-party software seamless and encourages greater innovation without requiring modification of standards across the industry. As these efforts become more widely adopted, SOA will become easier and more streamlined to use in SDVs.

9.2. Developing Robust Middleware Solutions

Middleware is critical in bridging the gap between legacy systems and an SOA-based framework. Nevertheless, creating middleware solutions that guarantee smooth communication, allocation of resources, and security among all the services takes a lot of work.

Because hardware platforms, operating systems, and communication protocols among systems can be diverse, our middleware has to be robust. On the other hand, it should also accommodate real-time data processing and fault-tolerable mechanisms for uninterrupted vehicle operation. A group of middleware solutions from RTI Connext and QNX is already taking strides to fulfill these requirements. Still, more development is necessary to satisfy fully autonomous and connected vehicles' needs.

Middleware development is an investment whose benefit will ease SOA adoption, improve system performance, cut expenses, and quicken SDVs' innovation.

9.3. Cybersecurity Innovations for SDVs

SDVs depend on connectivity and communication with the outside world, so cybersecurity is a growing concern. With multiple interconnected services, SOA-based architectures are vulnerable to hackers.

Automakers are mitigating risks of these sorts through advanced cybersecurity solutions. This includes end-to-end encrypted services, secure boot processes, and blockchain-based authentication protocols for the reliable validation of data and services. Additionally, threat detection and countermeasures are being accomplished in real-time through artificial intelligence (AI) and machine learning (ML) and include the installation of proactive defense mechanisms to protect against potential threats.

For automakers to gain trust in SOA-based SDVs, collaboration with cybersecurity experts and adherence to, e.g., the regulatory standards ISO/SAE

21434 for automotive cyber security will be necessary. Resolving security questions effectively will act as a catalyst for the widespread adoption of serviceoriented architectures in vehicles.

X. FUTURE OF MOBILITY: THE ROLE OF SOA AND SDVS

Service oriented architecture (SOA) and software defined vehicles (SDVs) are the future of mobility and will reframe what mobility means. These technologies can be leveraged to promote smart, safer, and more efficient transport; and fit into a growing need for greater sustainability and connexion in today's transport systems. In software-defined mobility, future vehicles will become less static machines and more adaptable, continuously updated platforms. Manufacturers will also rely heavily on air (OTA) technology, enabling them to elevate vehicle performance, include new features, and remedy problems without physically intervening. This process will increase vehicles' overall lifecycle and help ensure that they remain industrially compatible with emerging technologies.

SOA will also further introduce autonomous driving systems as modular architectures enable the integration of advanced sensors, artificial intelligence, and real-time decision-making algorithms. These sophisticated systems will allow vehicles to navigate complex environments better, reducing accidents and helping traffic run more smoothly. The role of cloud and edge computing will also grow, ensuring rapid data processing and storage for connected vehicles.

SOA adoption will act as a catalyst in bringing together automakers, technology companies, and third parties. Unified operating systems, modular middleware, and scalable services will be co-created on shared platforms and standards. These collaborations will lower the development cost, speed up the rollout of new features, and bring about the shipment of resources within the industry.

Long-term mobility vision consists of fully integrated smart systems with vehicles that communicate freely with each other, the infrastructure, smart devices, and macro, meso, and microsystems. For instance, these systems will enable real-time traffic optimization, automated fleet management, and personal travel experience. Electric and autonomous SDVs will moreover be able to integrate with sustainable energy grids and urge life-like transport solutions. Finally, there will be a focus on sustainable practices through electric and autonomous SDVs that can mesh with solar and other renewable energy grids and encourage life-like transport solutions.

Transformation of the automotive industry and the journey towards intelligently connected, sustainable future mobility is taking place with the convergence of SOA and SDVs.

CONCLUSION

In terms of software defined vehicles (SDV) and service oriented architecture (SOA), the automotive industry is reinventing its ways of implementing contemporary mobility. Due to a transition to a software – based platform it is possible, for example, to implement over – the – air updates, add integration with the latest technologies, and focus on user preferences. Taking SOA to the real world shows the promise of SOA in reshaping vehicle functionality: Tesla's software ecosystem and connected vehicle services.

Although this carries many benefits, the adoption of SOA in SDV faces challenges such as the integration with legacy systems, the cybersecurity risk required, and the data complexity. But, overcoming these obstacles is being traversed with collaborative efforts, robust middleware solutions, and standardization initiatives.

SOA has an important part to play in contributing to the intelligent, safe and environmentally sustainable future mobility systems. The future of transportation will be connected and autonomous vehicles, with infrastructure and the environment as an important part of that future. SOA gives the automotive industry an opportunity to build a more efficient, more innovative and greener mobility ecosystem which will be beneficial to both consumers and the society as a whole.

REFERENCES

[1] SOA là gì? Tổng quan về SOA Testing (Service Oriented Architecture). (n.d.). https://bizflycloud.vn/tin-tuc/tong-quan-ve-soa-testing-service-oriented-architecture-20180528104722989.htm

[2] Huh, Jun-Ho & Seo, Yeong-Seok. (2019). Understanding Edge Computing: Engineering Evolution with Artificial Intelligence. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2945338.

[3] Software-Defined Vehicles: The Ultimate Guide. (n.d.). https://blackberry.qnx.com/en/ultimate-guides/software-defined-vehicle

[4] F. Bock, D. Homm, S. Siegl, and R. German, "A Taxonomy for Tools, Processes and Languages in Automotive Software Engineering" Computer Science & Information Technology (CS & IT), pp. 241–256, 2016

[5] T. Woopen et al., "UNICARagil - Disruptive Modular Architectures for Agile, Automated Vehicle Concepts" in 27th Aachen Colloquium, Aachen, Germany, 2018

[6] Kampmann et al., "A Dynamic Service-Oriented Software Architecture for Highly Automated Vehicles" in 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019, pp. 2101–2108

[7] W. Zeng, M. A. S. Khalid, and S. Chowdhury, "In-Vehicle Networks Outlook: Achievements and Challenges," IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 1552–1571, 2016

[8] P. Dai, K. Liu, Q. Zhuge, E. H.-M. Sha, V. C. S. Lee, and S. H. Son, "Quality-of-Experience-Oriented Autonomous Intersection Control in Vehicular Networks," IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 7, pp. 1956–1967, Jul. 2016, conference Name: IEEE Transactions on Intelligent Transportation Systems.

[9] J. Chen, H. Zhou, N. Zhang, P. Yang, L. Gui and X. Shen, "Software defined Internet of vehicles: Architecture challenges and solutions", J. Commun. Inf. Netw., vol. 1, no. 1, pp. 14-26, Jun. 2016.

[10] T. Hackel, P. Meyer, F. Korf, and T. C. Schmidt, "Software-Defined Networks Supporting Time-Sensitive In-Vehicular Communication," in IEEE 89th Vehicular Technology Conference (VTC2019-Spring), 2019.

[11] Chaudhary, Arslan Asad. "EXPLORING THE IMPACT OF MULTICULTURAL LITERATURE ON EMPATHY AND CULTURAL COMPETENCE IN ELEMENTARY EDUCATION." Remittances Review 3.2 (2018): 183-205.

[12] Chaudhary, A. A. (2022). Asset-Based Vs Deficit-Based Esl Instruction: Effects On Elementary Students Academic Achievement And Classroom Engagement. Migration Letters, 19(S8), 1763-1774.

[13] Al Bashar, M., & Khan, I. H. (2017). Artificial Intelligence in Industrial Engineering: A Review. International Journal of Scientific Research and Engineering Development, 2(3).

[14] Bashar, M. A., Taher, M. A., Johura, F. T., & Ashrafi, D. (2017). Decarbonizing the supply chain: A green approach.

[15] J. C. Nobre, A. M. de Souza, D. Rosario, C. Both, L. A. Villas, ´

[16] E. Cerqueira, T. Braun, and M. Gerla, "Vehicular software-defined networking and fog computing: Integration and design principles," Ad Hoc Networks, vol. 82, pp. 172–181, Jan. 2019.

[17] Zhao, F., Liu, Z., Hao, H., Shi, T.: Characteristics, trends and opportunities in changing automotive industry. J. Automot. Saf. Energy 9(3), 233–249 (2018)

[18] Chaudhary, A. A. (2018). Enhancing Academic Achievement and Language Proficiency Through Bilingual Education: A Comprehensive Study of Elementary School Students. Educational Administration: Theory and Practice, 24(4), 803-812.

[19] Vdovic, H., Babic, J., Podobnik, V.: Automotive software in connected and autonomous electric vehicles: a review. IEEE Access 7, 166365–166379 (2019)

[20] Bjelica, M., Lukac, Z.: Central vehicle computer design: software taking over. IEEE Consum. Electron. Mag. 8(6), 84–90 (2019)

[21] Kuang, X., Zhao, F., Hao, H., et al.: Intelligent connected vehicles: the industrial practices and impacts on automotive value-chains in China. Asia Pac. Bus. Rev. 24(1), 1–21 (2018)

[22] S. Arole, "From Monolith to Service-Oriented Architecture: A Model-Based Design Approach towards Software-Defined Vehicles," 2023 5th International Conference on Electrical, Control and Instrumentation Engineering (ICECIE), Kuala Lumpur, Malaysia, 2023, pp. 1-9, doi: 10.1109/ICECIE58751.2024.10457489.

[23] Margaret PP. What is software-defined networking (sdn)? - definition from whatis.com. http://searchsdn.techtarget.com/definition/software-defined-networking-SDN

[24] Soua A, Tohme S. Multi-level SDN with vehicles as fog computing infrastructures: a new integrated architecture for 5G-VANEts. In: 2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN). Paris, France: IEEE; 2018: 1-8.