Journal of Information and Intelligence 1 (2023) 23–35

Contents lists available at ScienceDirect

Journal of Information and Intelligence

journal homepage: www.journals.elsevier.com/journal-of-information-and-intelligence

Vehicle Computing: Vision and challenges

Sidi Lu[a,*], Weisong Shi[b]

[a] *Department of Computer Science, Wayne State University, MI 48202, USA*
[b] *Department of Computer and Information Sciences, University of Delaware, DE 19716, USA*

A R T I C L E I N F O

Keywords:
Vehicle Computing
Connected vehicle
Edge computing
Autonomous driving

A B S T R A C T

Vehicles have been majorly used for transportation in the last century. With the proliferation of onboard computing and communication capabilities, we envision that future connected vehicles (CVs) will be serving as a mobile computing platform in addition to their conventional transportation role for the next century. In this article, we present the vision of Vehicle Computing, *i.e.,* CVs are the perfect computation platforms, and connected devices/things with limited computation capacities can rely on surrounding CVs to perform complex computational tasks. We also discuss Vehicle Computing from several aspects, including several case studies, key enabling technologies, a potential business model, a general computing framework, and open challenges.

1. Connected vehicles: From present to future

With the vast improvements in computing technologies and the wide deployment of communication mechanisms, connected vehicles (CVs) are swiftly transforming automotive industry, with a tremendous amount of next-generation services (*e.g.,* mobility-as-a-service, fuel efficiency optimization, high-definition (HD) map generation, and adaptive cruise control). These software-based services fuel the CV market — the global CV market is valued at $65 billion in 2021 ($56 billion in 2020 due to Covid-19), and is projected to recover from the impact of the global coronavirus pandemic and reach $225 billion by 2027 with a compound annual growth rate (CAGR) of 17% [1]. Automotive edge computing consortium (AECC) predicts that every new vehicle (with a total number up to 400 million) will be connected by 2025, which will result in 50% of national vehicles on the road with connected features [2].

This article envisions that future CVs will be serving as a mobile computing platform in addition to their conventional transportation role for the next century. Specifically, we start with the concept of Vehicle Computing, which emphasizes that future CVs are expected to provide efficient onboard computation for connected devices/things. We then illustrate the significance of Vehicle Computing and explain why it is emerging. Five promising case studies are introduced, followed by a discussion of the key technology for Vehicle Computing, *i.e.,* software-defined vehicles (SDV). Next, we present a transformative Vehicle Computing business model that opens opportunities for new revenue streams, thus offsetting the high cost of building and maintaining vehicle fleets. Finally, we conclude the article by introducing a general computing framework called EdgeArC and discussing the remaining challenge, including a quantitative analysis of data transmission cost, which in turn calls for Edge Computing [3] in connected mobility. We hope this article will gain attention from the automotive communities and inspire more research in Vehicle Computing.

* Corresponding author.
E-mail addresses: lu.sidi@wayne.edu (S. Lu), weisong@udel.edu (W. Shi).

https://doi.org/10.1016/j.jiixd.2022.10.001
Received 10 August 2022; Received in revised form 18 October 2022; Accepted 18 October 2022
Available online 30 October 2022

2. Vehicle Computing

In this section, we first present our definition and vision of Vehicle Computing, then we list important reasons why Vehicle Computing is important for the years to come.

2.1. *Definition of Vehicle Computing*

Vehicle Computing refers to the enabling technologies that allow computing to be performed on CVs, where CV serves as a computation platform for diverse edge-enabled services. Different from Vehicular Networking [4], which serves as a communication enabler for enormous applications associated with transportation, Vehicle Computing focuses on the computation function of CV and emphasizes that CV is a promising computing platform that can assist in analyzing data flow from on-board sensors and surrounding connected devices/things, even if the vehicle is parked or in the charging status.

More specifically, the concept of Vehicle Computing is inspired by the fact that future CVs will have compelling computing and communication capabilities; hence, computation-constrained connected devices/things are able to rely on surrounding CVs to complete computation-intensive tasks and send related results to the end-user. For instance, suppose a police officer equipped with a body-worn camera is on duty on the side of a highway. The body-worn camera keeps capturing and sending video to the surrounding police vehicle in real-time for delay-sensitive applications (*e.g.,* license plate detection). When an event of interest occurs, the police vehicle will issue a warning immediately. In this example, the vehicle acts as an efficient computing platform to analyze real-time data received from the body-worn camera, and the computation resources can be utilized reasonably and efficiently, and therefore, time-sensitive applications can be completed on time.

Drawing from the definition of Vehicle Computing, we further introduce the future Vehicle Computing paradigm in Fig. 1, which is driven by the communication of Vehicle-to-Infrastructure (V2I), Vehicle-to-Vehicle (V2V), and potentially Vehicle-to-Everything (V2X). V2X not only enables CVs to communicate with diverse transportation systems such as cellular towers, traffic cameras, road-side units (RSUs), scooters, cyclists, and pedestrians, it also allows CVs to communicate with elements of the surrounding environment such as industry IoT devices, health sensors, smart home sensors, and edge servers.

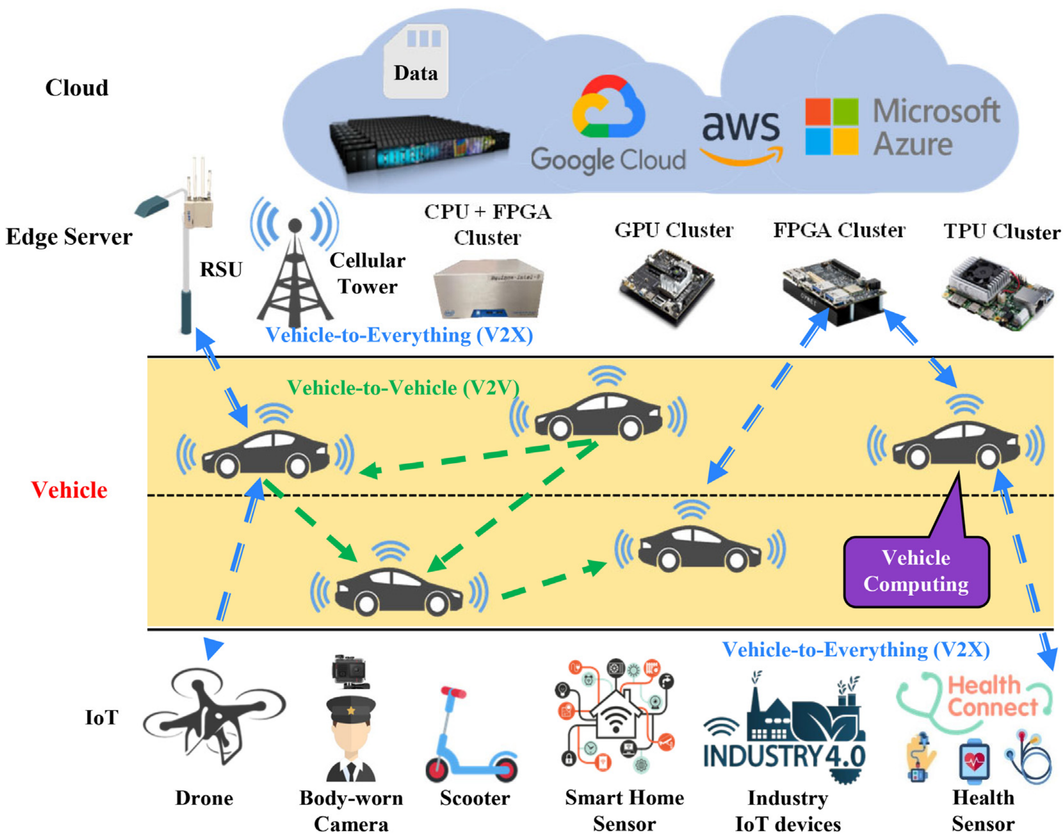


Fig. 1. The paradigm of Vehicle Computing.

2.2. Significance of Vehicle Computing

2.2.1. Push to clouds and edge servers

Today, a single traditional vehicle may be equipped with more than 100 sensors, generates approximately 25 GB of data per second, and the daily data volume can exceed 11 TB [5]. Besides, a single connected and autonomous vehicle (CAV) is able to generate 19 TB data per hour [6]. At the same time, the amount of data generated by CV is still growing. Estimates predict that global CVs will transmit 1 to 10 exabytes of data traffic per month by 2025, at least 1000 times larger than the present volume [2]. In this context, the speed of data transmission and limited bandwidth both have become bottlenecks for pushing data to the cloud or edge server for data analysis, which poses a grand challenge to provide latency-sensitive services. In addition, even if previous work compressed CV data before sending it out, the raw data may still be exposed bringing potential risks of privacy leakage. Hence, the latency bottlenecks, bandwidth limitations, and privacy concerns, in turn, call for Vehicle Computing, a new computing paradigm that places the computing near the data source. Previous studies also proved the potential benefits (*e.g.,* the significant reduction in response time and energy consumption) by moving computing from the cloud to the proximity of data.

2.2.2. Pull from IoT devices

Cisco predicts that the quantity of global IoT devices will grow up to 500 billion by 2030. In the meanwhile, most of the DNNs focus on boosting accuracy at the expense of substantially increased model complexity — the depth of the current state-of-the-art networks, such as Inception-v4 and ResNet-50, can reach dozens or even hundreds of layers to outperform previous networks for related tasks with accuracy. A single layer may require millions of matrix multiplications. Such heavy calculation brings challenges to deploy these DNN models on resource-constrained IoT devices. In this context, we infer that IoT devices will leverage the surrounding CVs with powerful computation capabilities to execute computation-intensive models, which will have a great impact on IoT and automotive communities.

3. Case studies

Next, we list and discuss five promising use cases to further illustrate out vision of Vehicle Computing.

Infrastructure Health Management: To date, approximately 46 K national bridges have structural defects due to inadequate maintenance, and around 178 million trips are taken across these bridges every day [7]. The sudden collapse of aging and deficient bridges (*e.g.*, Pittsburgh bridge and the 35 W bridge over the Mississippi River) strongly warned us the urgent need for infrastructure health management.

Previously, researchers installed numerous types of sensors throughout the full length of the bridge to monitor its health status. In the Vehicle Computing era, since vehicle is equipped with a variety of sensors (*e.g.,* pressure sensor, vibration sensor, GPS, camera, radar, and LiDAR), vehicle itself can be considered as a mobile sensor. Specifically, as shown in Fig. 2, when a vehicle runs on a bridge, it can continuously measure and collect the structural response of the bridge (*e.g.,* vibration of the bridge), which can be considered as indicators of bridge health status. Besides, in addition to sensor data, vehicles can also generate real-time driving data (*e.g.,* speed, stop, and acceleration) and capture context data (*e.g.,* weather, vehicle model, and road information). Then, it can directly analyze these data

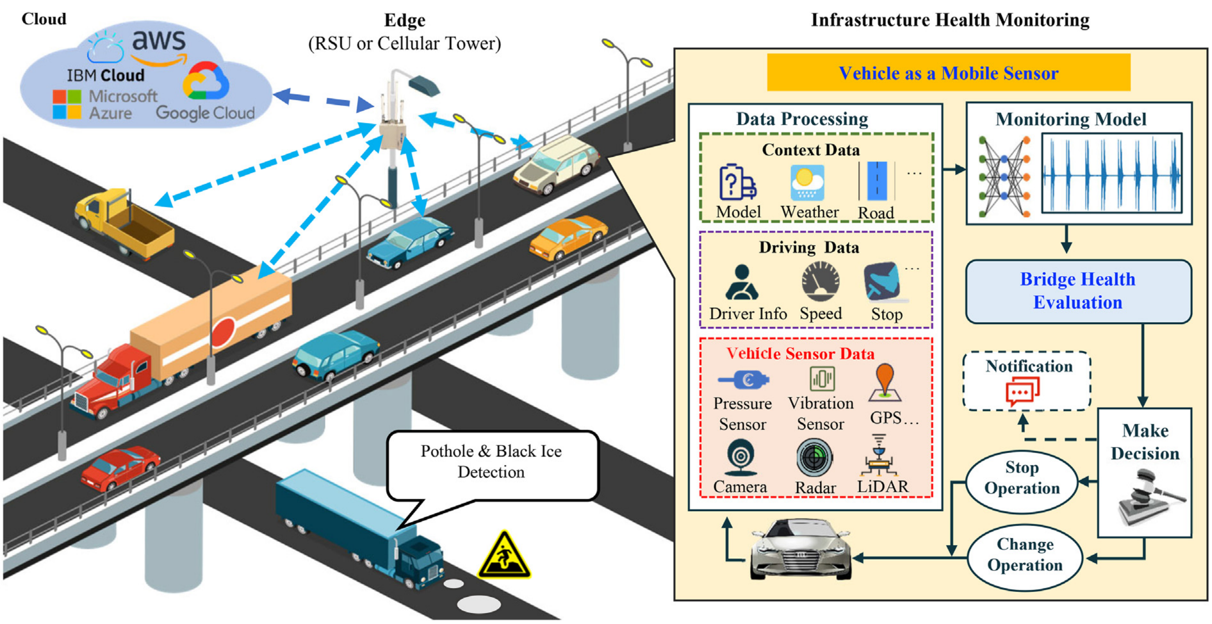


Fig. 2. An example of infrastructure health management.

through onboard computing for the bridge health evaluation and send their initial decision to surrounding edge such as RSU or cellular tower. Edge receives related information from multiple vehicles and performs further computation for confirmation. If needed, edge will send warnings and traffic control signals to surrounding and forwarding vehicles and push related results to cloud for storage and potential further analysis.

In-Vehicle Meeting: Besides, take a fully connected and autonomous vehicle (CAV) as an example, it can realize safe navigation by itself without the driver's concentration on driving. Therefore, we opine that future CVs are expected to provide next-generation services such as online in-vehicle meetings. To be concrete, with the dramatic developments of ultra-reliable and low-latency communications, future CVs are capable of enabling smooth in-vehicle meetings, which allows people to share information in real-time, instead of attending virtual meetings in person at home or in the office. In addition, people can seamlessly participate in the same meeting at home, in the vehicle, and in the office, without being bothered by the repeated logout and login process, which could significantly improve work efficiency.

In-Vehicle Delivery: In addition, we project that vehicles will become key components of smart homes to improve people's daily life. For instance, in-vehicle delivery is becoming an alternative option to door-to-door delivery since it can greatly reduce the inconvenience caused by missed deliveries or trips of picking up packages at the local post office. Nowadays, Amazon is taking the obvious next step by cooperating with established automotive companies to launch early in-vehicle delivery services. The customer will be notified when the delivery driver sends a request to remotely unlock the vehicle for delivery. Depending on the parcel size, the driver will then place packages in the cargo area or cabin and send a remote command to lock the vehicle again, and the customer will receive a final notification. In this way, even if customers are not at home, they can still safely receive their packages.

In-Vehicle Augmented Reality: Furthermore, we opine that augmented reality (AR) technologies will be able to turn vehicles' windshields into movie screens, which can provide passengers with full-color graphics of the driving environment with a wide viewing angle, making the journey more interesting and safe. Nowadays, Civil Maps, a 3D map software provider, shows passengers how CVs equipped with AR displays can navigate in a complex driving environment. In addition, Alibaba also invested $18 million in the head-up display (HUD) company, WayRay, the company released Navion, which is the first holographic AR car navigation system that can display travel details and hazard warnings in real time without wearing AR helmets or glasses. We opine that AR-enabled HUDs will be replaced by AR-enabled windshields, which can respond to diverse hand gestures and voice commands.

In-Vehicle Entertainment: Similarly, vehicles have the potential to transform the way people travel by providing new audio and video entertainment to enhance people's ride experience. MarketsandMarkets projects that by the end of 2022, the in-vehicle entertainment market will reach $30.47 billion, with a CAGR of 11.79%. Besides, automakers have integrated smartphone operating system interfaces into the vehicles' dashboards. For instance, Lincoln and Ford vehicles will be equipped with Google's Android operating system in 2023 to provide Google applications. This kind of evolution reveals that in-vehicle entertainment is on the rise. We envision that in the near future, passengers are able to choose a variety of extended reality (XR) games that can provide real-time physical vehicle feedback, such as the driver's acceleration, stopping, and steering; therefore, each game experience is unique. In addition, due to the rapid development of V2V communication, passengers of different CVs can also play in-vehicle games together on the road, which will further enrich the types and diversity of in-vehicle entertainment.

4. Software-defined vehicles

In this section, we introduce software-defined vehicle (SDV), which is an emerging key technology for Vehicle Computing and a practical approach to support vehicle application development. We first provide an application landscape for CVs and describe what is SDV and why SDV is a forthcoming evolution in automotive industry. We also introduce technical approaches to realize SDV.

4.1. Application landscape

Nowadays, vehicles are becoming more software dependent than ever: CVs today can already have up to 150 million lines of software code, distributed among more than 100 electronic control units (ECUs) and a growing number of sensors [8]. Inspired by previous studies and our vision of Vehicle Computing, we divide CV applications into four categories according to their themes: Safety, mobility, information, and computation:

i) **Safety application** mitigates the risk of hazards by issuing warnings to vehicle operators or to directly control-ling a single vehicle's actuators. These applications (*e.g.,* automatic braking, blind spot warning, and hazard reporting) usually address the critical need and/or call for hard real-time data processing and vehicle response.
ii) **Motion application** includes individual motion services and group motion services. Individual motion services intelligently provide soft real-time motion advisories for a single vehicle based on vehicle's real-time locations, destinations, and dynamic driving environment (*e.g.,* work-zone queue lengths and vehicle types), including path planning, assisted lane switching, and dynamic vehicle routing. Group motion application uses vehicle sensors and external data to influence or control the behavior of vehicles and drivers in aggregate. The purpose is to maximize a series of objective functions, *e.g.,* to save maximal fleet fuel or to reduce overall transit time.
iii) **Information application** aims to enhance users' comfort and ability to perform other tasks while driving or allow viewing vehicle parameters remotely. These applications generally tolerate transmission delays (soft real-time or non-time-critical) and errors, such as dynamic travel time prediction, fuel efficiency optimization, and in-vehicle air quality management.

iv) **Computation application** reflects the efficient on-board computation capability of CVs, which can help the computation-constrained connected devices/things to finished computation-intensive tasks even when CV is in charging or parking mode. Specific examples are listed in Section 1.

4.2. A forthcoming evolution in automotive industry

In the last decades, successful information technology companies (*e.g.,* Apple) have opened a new path to win over and lock in customers by continuously providing the applications and experience they desire. It makes customers feel that they are always at the forefront of technical innovation. The products they sell (*e.g.,* iPhone) are not one-time purchases that bring only one-time revenue, instead, selling a product is just an initial transaction that will further trigger a continuum of applications and connections, and finally results in a fiercely loyal customer base.

The Evolution of Automotive Software: Automobile manufacturers must do the same to stand out from the fierce competition. With the rapid development of automotive electronics and mechanical units, automotive hardware systems will gradually become standardized and unified, and it will be difficult for automotive companies to differentiate in hardware. At this time, merging intelligent software and algorithms will become the core elements of competition among automotive manufacturers. Unlike mobile phones, the life cycle of vehicles usually spans 10 to 15 years. Such a long-life cycle calls for the continuous addition and update of new applications, including oil life prediction, brake pad prognostics, trajectory planning, dynamic travel time prediction, adaptive cruise control, battery failure prediction, lane detection, black ice detection, pothole detection, real-time object detection, and in-vehicle air quality management. Moreover, to meet the realities of consumer expectations and the emerging performance requirements, the complexity and quantity of software in the vehicle is still increasing exponentially to provide different types of applications.

The Evolution of Automotive Computing System: Moreover, the forthcoming evolution in CV is not limited to software only, and it will require a major redesign of the underlying supporting automotive computing system architecture. Currently, most production vehicles are equipped with various small and fixed-function ECUs, which are usually produced by different suppliers. All ECUs are connected via Controller Area Network (CAN) bus and can communicate with each other without a host computer. The CAN bus sequentially transmit data frames to all devices. In the case of a conflict with more than one transmission device, the one with the highest priority device can continue while the others will back off. As shown in Fig. 4 (a), various automotive functions are distributed across multiple ECUs throughout a vehicle. Given the limited resources in an ECU's micro-controller, it is almost impossible to deploy diverse computation-intensive applications onto such a device without incurring significant efforts in redesigning the vehicle's computing system architecture.

To help formalize the analysis, some leading automotive companies such as Tesla and APTIV [9] have adopt a type of sensible computing system architecture as shown in Fig. 4 (b). The idea behind this in-vehicle high performance computer architecture is to keep most existing ECU designs as is to not only better meet the software-based functional safety (FuSA) and real-time constraints, but also ease the transition from traditional vehicle architectures to the future of software-defined architecture. Under this architecture model, ECUs are partitioned into multiple functionally related zones controlled by their zone controllers (ZCs). ZCs are further interconnected with each other and partitioned into a set of domains controlled by domain controllers (DCs). Both ZCs and DCs are modern microprocessors capable of running embedded Linux operating systems. This system architecture not only greatly simplifies vehicles' system interconnection, but also makes the deployment of software functionalities to both ZCs and DCs possible. With SDV's application services become more and more intelligent, it will demand for more domain specific accelerators or general-purpose accelerators (such as GPUs and FPGAs), which can be easily integrated with ZCs and DCs through the standard microprocessor interfaces, such as PCIe (Peripheral Component Interconnect Express) and CXL (Compute Express Link). Through wireless connections, this architecture can be easily complemented by edge servers and clouds for collaboration.

4.3. What is software-defined vehicle

In recognition of the growing complexity and importance of software, as well as the forthcoming evolution in automotive computing system architecture (Figs. 3 and 4), thought-leaders from automotive industry, automotive OEMs (Original Equipment Manufacturers), and mobile network operators (MNOs) should work together in pursuit of the SDV.

The idea behind SDV is to emphasize a software-centric view of customers' experience with the vehicle, which can be personalized and updated with newer versions of software and firmware throughout the life-cycle of the vehicle, and it can integrate components from many different suppliers (just like a smartphone). Moreover, software updates no long require customers visiting a dealership, but can be done through an over-the-air (OTA) update technology whenever as needed without affecting driving, which has already proven to improve consumer satisfaction [10].

More specifically, in SDV, the software will deeply participate in the process of the CV development, verification, sale, and service, and all these processes will be continuously changed and optimized through software updates to continuously improve the performance, safety and comfort of CVs. The emerging of SDV is opening opportunities for new revenue streams, efficiencies, and closer lifetime relationships with consumers. Estimates predict that, by 2025, many automotive companies are likely to sell CVs at prices close to cost and provide value to users mainly through software.

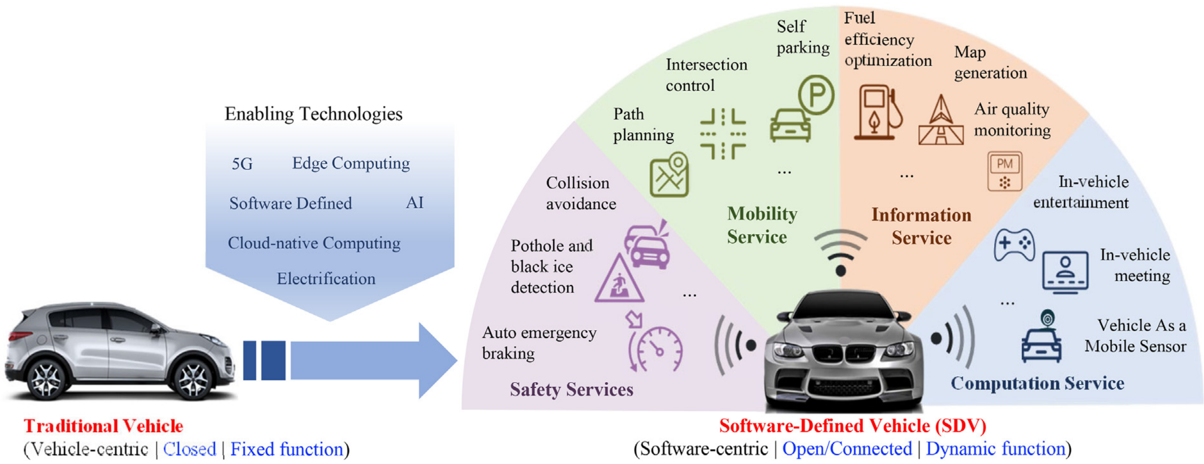


Fig. 3. Disruptive transformation of automotive mobility.

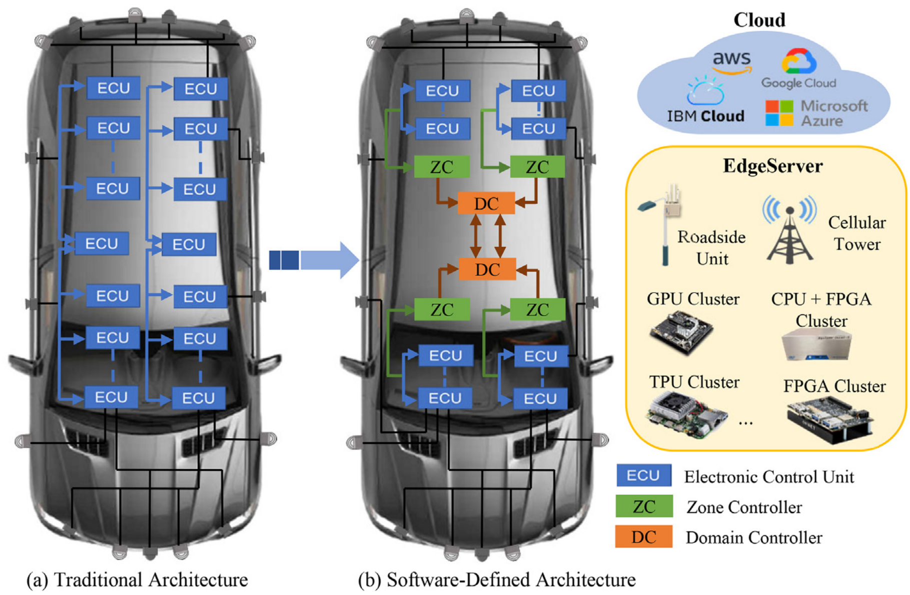


Fig. 4. The paradigm shifts in automotive computing system architecture.

4.4. Why is SDV important?

First, SDV is the next technological disruption to address software related issue and therefore guarantee the safety of CVs and reduce automotive recalls and costs. The number of automotive recalls and costs linked to software failures has risen exponentially in the U.S. — in 2020, a total of 83.2 million vehicles have undergone recalls, of which 75 million (90%) vehicles have been affected by software related problems. This reveals that more than a quarter of vehicles on the roads nationally have been recalled due to the software problem at least once a year (with an estimated 286.9 million registered vehicles in the U.S [11]). Since the average cost of an auto recall over the last 10 years was about $500 per vehicle [12], it can be estimated that the average software-related recall cost nationally is around **$38 billion** in 2020. With the increasing number of vehicles, the software-related safety threat and cost is greater than ever.

All these recalls and costs could have been avoided if there were appropriate over-the-air (OTA) software updates enabled by SDV. It can provide continuous assistance for patching against security holes and fix bugs that can cause malfunction in vehicles. OTA update also provide enormous advantages in keeping in-vehicle software systems up to date and maintaining consumer satisfaction.

Besides, other benefits of SDV include but not limited to: *i*) *Simplified integration and automatic deployment:* Each state-of-the-art service can be deployed collaboratively and independently with standard continuous integration and continuous delivery (CI/CD) infrastructure; *ii*) *scalability and reusability across applications:* Enable the resue of code and the mass deployment of software between vehicle models and generations; *iii*) *better fault isolation*: SDV can isolate software from its environment and ensure that each module can work uniformly and independently; hence, it can improve the safety of CV by realizing fault isolation.

4.5. How to realize SDV?

Cloud-native in Automotive: Achieving software consolidation and developing effective multi-layered architecture are the keys to realizing SDV. As a modern software-development technology, cloud-native computing [13] is considered as a promising design paradigm providing a scalable solution to the reality of SDV. It embodies a number of proven workflows and design patterns for large-scale software development and complex service management.

The key concept of the cloud-native design principle is Service Oriented Architecture (SOA), in which an application is decomposed as a set of self-contained functional services (also known as microservices) that can be deployed and orchestrated onto computing devices at different locations. These microservices are typically deployed as either containers or virtual machines (VMs), depending on the nature of the application requirements. The open-source project Kubernetes (or K8s) has become the de facto industry standard orchestration platform for container life-cycle management, deployment, and scaling across computing systems. It relies on technologies such as microservices, containers, service meshes, immutable infrastructure, CI/CD, and declarative APIs. There are many K8s variants for embedded devices or Internet-of-Things (IoTs), such as K3s, Microk8s, MicroShift.

A properly designed cloud-native vehicle architecture can consolidate a modern vehicle's various resources (such as system hardware, software, cloud services) into an integrated and centrally managed platform. With a cloud-native SDV orchestration platform, SDV services can be run as different microservices inside either a container or virtual machine, enabling scalable management and smooth migration of microservices across different computing devices, in the vehicle, on the edge, or across data-center clouds.

However, only transferring cloud-native technologies to automotive is far from realizing SDV. Having standardized reference open-source code libraries is a grand challenge. The power of the open standardized approach is that it encourages people to innovate in the container ecosystem and create a solution that meets the domain-specific requirements of a particular deployment. Without open standards, the path to deliver SDV will be longer and the related cost will be higher. In this context, Arm propose the Scalable Open Architecture for Embedded Edge (SOAFEE) project that offers an open-source reference implementation enabling cloud-native technologies (*e.g.,* container orchestration) for SDV, which helps automotive developers accelerate time to market.

5. Vehicle Computing business model

Although Vehicle Computing bring a myriad of potential benefits, it comes with additional costs, including the hardware cost for computing and communication, and the software cost for service and maintenance. We opine that it is necessary to have a transformative business model, which is different from previous business model between automotive OEM, vehicle dealership, and end user. In this section, we propose a potential business model for Vehicle Computing, which opens opportunities for new revenue streams and offsets the high cost of building and maintaining intelligent CVs.

Specifically, Fig. 5 depicts this Vehicle Computing business model, which includes five key components:

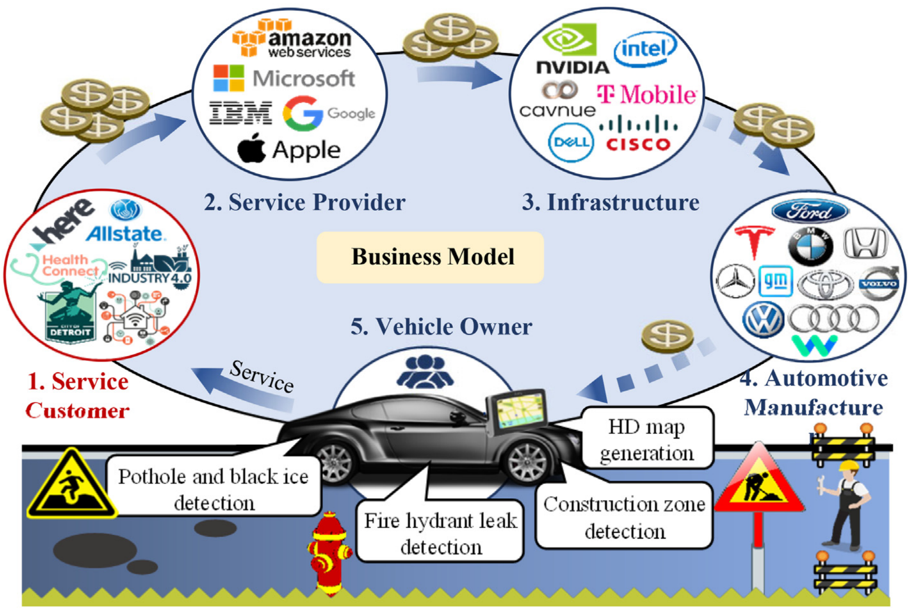


Fig. 5. The potential business model for Vehicle Computing.

i) **Service customer** includes but are not limited to insurance companies, government departments, technology companies, smart homes, *etc.* They receive and pay for the services provided by CVs through Vehicle Computing, such as pothole and black ice detection, fire hydrant leak detection, construction zone detection, and HD map generation.
ii) **Service provider** refers to the enterprise that powers a variety of emerging mobility services through CV innovation. For example, AWS has developed a complete set of automotive-specific services and solutions to empower digital transformation across the automotive industry.
iii) **Infrastructure** usually provides building blocks needed to develop and validate CV technologies or provide communication support to deliver everything needed for CV at scale. For instance, Cavnue is in the process of developing the physical, digital, coordination, and operational infrastructure for CVs to make roads safer and less congested.
iv) **Automotive manufacturer** collects and provides generated vehicular data through open data analytics platforms or software/system architectures, such as Arm's SOAFEE, AUTOSAR (AUTomotive Open System ARchitecture), and OpenVDAP (Open Vehicle Data Analysis Platform).
v) **Vehicle owner** utilizes their vehicles' computation resources and captured/generated data to provide services to customer and therefore obtain extra revenue.

To be concrete, take an electric vehicle (EV) as an example, suppose it drives over a few roads in downtown before slow charging. Since the majority of slow charging points are rated at 3 kW and will recharge an EV in 8–12 h. This makes EVs very suitable for charging overnight or when the owner is in the office, which provides a perfect opportunity to perform Vehicle Computing. During slow charging, EV is able to perform energy-intensive computation based on data generated by its various sensors, such as identifying if roadwork affects lane or road boundaries and then creating reliable HD maps in slow charging status and provide this updated HD map to service customer (*e.g.,* HERE Technologies). In this way, Vehicle Computing in slow charging status provides an economic way to keep HD maps up to date. Besides, if EV detects a leaking or broken hydrant leaking based on its captured video, it can also report this information to the associated government department. In this example, service customers (*i.e.,* HERE Technologies and local government department) will pay for the desired services. Since the service provider, infrastructure, automotive manufacturer, and the vehicle owner both contribute to these services, they will all receive remuneration (as shown in Fig. 5). In this business model, the relationship of these five key components is mutually beneficial as they supported each other by opening revenue streams or providing desired services efficiently.

Collaboration is Happening: Currently, automotive industry and tech companies realized that they have to work together to create an exciting future as opposed to working by themselves, and the automotive leaders are accelerating their transformation into software-driven mobility provider. For example, Toyota has been collaborating with Amazon Web Services Inc. (AWS) on applying AWS's applications to Toyota's Mobility Services Platform (MSPF) [14], and Toyota has been developing a new automotive operating system called Arene to support OTA update, which is planed to be integrated into its vehicles by 2025 [15]. Ford and Google have announced a strategic partnership to provide unique vehicle applications and accelerate Ford's transformation plan [16]. In addition, General Motors and Cruise have partnered with Microsoft to bring new mobility services for CAVs [17]. Besides, SAIC motor has teamed up with Chinese e-commerce giant Alibaba to launch an all-new intelligent electric vehicle brand [18].

6. EdgeArC: Edge-based architecture for connected vehicles

The upgrading of automotive end-to-end architecture is mainly reflected in three aspects: hardware architecture, software architecture, and computing architecture. Also, as depicted in Fig. 3, software architecture is developing from high coupling of software and hardware to hierarchical decoupling (containerized). Also, as described in Fig. 4, hardware architecture is changing from distributed to domain control/centralized development.

In this context, we propose and illustrate an edge-based computing architecture for CVs, called EdgeArC, which enables vehicles to collaborate with surrounding vehicles, offloading workloads to edge servers and connected de-vices/things (denoted as XEdge) and remote clouds.

Specifically, Fig. 6 presents the three-tier paradigm of EdgeArC. In EdgeArC, vehicles are software-defined where the consolidation of functional blocks within the vehicle can be enabled, and the performance, safety, and comfort of CVs can be continuously improved through OTA updates. In vehicle, a hypervisor is deployed on top of the vehicle computing unit, which can provide well isolated virtualized environments for diverse software. Each ECU codebase can run almost unmodified in its own virtual machine and the resulting server platform may run a mix of real-time operating system (RTOS) for safety-critical and hard real-time applications (*e.g.,* collision avoidance and real-time diagnostics) and rich operating system (OS) for the soft real-time services (*e.g.,* map generation) and non-time-critical services (*e.g.,* infotainment).

Besides, thanks to the rapid development of the V2V technologies, vehicle services could be migrated to collaborative vehicles that have idle computation resources. Here, the decentralized approach is an appropriate approach to support the data transmission between vehicle fleets. More importantly, vehicle may also need to push its data to XEdge (*e.g.,* RSUs, cellular towers, gas stations, charging piles, and the computation devices that are installed in a connected home) and pull corresponding analysis results from XEdge, where the publish/subscribe and broadcast can be suitable approaches. In the XEdge layer of Fig. 6, an arc that decreases in size from left to right indicates a decreasing number of these XEdge devices that may be encountered by the host vehicle. In the meanwhile, XEdge devices may also upload data or computation-intensive services to clouds for further analysis and then receive related results, and the centralized mechanism is suggested for XEdge devices to send and receive data through cellular or satellite communication, *etc.*

Fig. 6. An illustration of EdgeArC.

7. Open challenges

To realize the vision of Vehicle Computing, we argue that the systems, algorithms, and network community need to work together. In this section, we will further summarize the remaining challenges in detail.

7.1. Real-time constraints of automotive services

The first challenge is to meet the real-time constraints of automotive services, *i.e.,* identifying the most critical tasks from a multitude of dynamically changing tasks. Here, real-time constraints can be divided into three categories:

i) Hard real-time: Missing a hard real-time deadline may cause catastrophic consequences, *ii*) soft real-time: Missing a soft real-time deadline may make the computation results useless, and *iii*) non-time-critical: Missing a non-time-critical deadline may reduce the effectiveness and degrade the utility of the results. For instance, in a normal driving scenario, the processing of video streams may be soft real-time or even non-time-critical. However, in the case of vehicle collision avoidance, the analysis of camera video streams can suddenly become extremely critical and hard real-time. In this case, the system must quickly identify critical services and provide absolutely guaranteed system resources to process those critical messages to ensure the real-time constraints on vehicle collision avoidance tasks.

7.2. Heterogeneity of vehicle system components

The second challenge is the heterogeneity of vehicle system components from different vendors, including a mixture of micro-controllers, real-time processors, microprocessors, and an array of general purpose accelerators such as Graphic Processor Units (GPUs) and Field Programmable Gate Arrays (FPGAs). Moreover, some of the micro-controllers cannot be even re-programmed or configured as a general-purpose computing device. For example, most of ECUs' functionalities cannot be changed without explicit permission, and some even come with a protection lock in a form of digital rights management. This is understandable given that the ECU's software and hardware are typically developed by tier-one suppliers with FuSA in mind. They have to undergo rigorous testing and validation processes (such as the failure mode and effect analysis) to catch various failures that may lead to catastrophic injuries to customers.

Moreover, because of the heterogeneity of vendor-specific devices, there is no standard profiling methodology to correlate metrics collected from those heterogeneous devices, which makes architectural-level evaluation and bench-marking almost impossible.

7.3. Open APIs

Machine learning-based applications have been gaining ever-increasing impetus in CV community due to their state-of-the-art performance. Unfortunately, the number of open-source vehicular computing platforms supporting data analysis is very limited. Except for Arm'SOAFEE and Baidu's Apollo, a majority of automakers (*e.g.,* General Motors and Ford) are focusing on their proprietary platforms. Furthermore, although Apollo is public, it based on the in-vehicle only computing approach that place all data processing on the vehicle, which is neither suitable nor scalable for mass-market vehicles with an abundance of third-party services.

Different from the proprietary platform, open-source platforms can provide the automotive community with real-field vehicle data and free APIs, and these open APIs can enable researchers and developers to develop and evaluate CV applications in real environments. Recently, AWS and BlackBerry are jointly developing a scalable cloud-connected software platform called BlackBerry IVY, which will allow automakers to leverage the new BlackBerry QNX and AWS technologies to improve CV operations. In addition, researchers have proposed the OpenVDAP [19], a full-stack hardware/software platform that offers a library of public edge-enabled applications. CVs needs more open APIs to promote the development of third-party services.

7.4. Vehicular communication

Another open problem is the unreliable networking connectivity for a constantly on-the-wheel moving vehicle. This poses significant challenges for vehicle service continuity in the case of service migration across different wireless networks and cloud providers.

The mainstream communication mechanisms, such as Dedicated Short Range Communication (DSRC), Long Term Evolution (LTE), Cellular-Vehicle-to-Everything (C–V2X), and WiFi, have enabled CVs to communicate with nearby vehicles, connected devices/things, and cloud. Although DSRC is a mature mechanism with decades of development history, it has non-negligible shortcomings such as low throughput and small coverage. By contrast, LTE and WiFi can provide more bandwidth but achieve poor performance in the mobile scenarios. As a recently developed communication mechanism, C–V2X performs well in environments with high vehicle density mobility. Therefore, several established automotive manufacturers (*e.g.,* BMW and Ford) and well-known chip manufacturers (*e.g.,* Ericsson and Qualcomm) have recently shifted their focus to C–V2X. However, C–V2X is a relatively new technology. Compared with DSRC, its cost is not affordable, and the scope of deployment is not wide. Hence, ultra-reliable, low-latency, and energy-efficient communication for distributed and moving CVs with massive connectivity density may be one of the most challenging tasks.

7.5. Computation offloading

To ensure safe and reliable driving, vehicles, connected devices/things, and cloud often work together to process sensor data, transmit essential information, expand sensing capabilities, and coordinate their decisions. It was estimated that 81% of collisions can be potentially addressed by V2V and V2I [20].

However, due to latency and reliability limitations, computation offloading is not always feasible. Considering the heterogeneity of computing power and the interdependence of computation services, previous researchers have proposed novel programming frameworks and task offloading algorithms to support and optimize the problem of computation offloading [5]. Nonetheless, almost all of the work is based on simulations, and the evaluation in real-world driving scenarios needs more exploration.

Furthermore, although computation offloading can also enable computation-intensive applications on vehicles with limited computation capabilities and power, the data transferring for computation offloading poses potential privacy threats, which may further lead to location and identity manipulation, vehicle tracking, and virtual vehicle hijacking. Therefore, avoiding the privacy leakage of computational offloading still remains a challenge.

7.6. Energy consumption

Due to the tremendous amounts of sensors and the deployment of complex algorithms on CVs, huge energy consumption has become a major problem. Take NVIDIA Drive PX Pegasus as an example, when the AI computing power is 320 INT8 TOPS, the power consumption can up to 500 W. In addition, if a replication system is installed to ensure the reliability of vehicle applications, the total power consumption can reach 2000 W.

Moreover, taking the electric vehicle (EV) as an example, suppose that the total mileage of each EV is composed of 55% of city mileage and 45% of highway mileage, driving at a speed of 31 mph and 56 mph, respectively. Then, we can infer that the annual energy consumption of EVs nationwide for computation is approximately 180 TW-hours [21]. According to reports [22], Google data centers now consume around 12 TW-hours of electricity per year. Therefore, we further infer that each year, the national energy consumption of EVs is approximately equal to the total energy consumption of 15 representative technology companies' data centers.

Hence, how to solve the problem of the huge amount of energy consumption is imperative. In addition, since most of the energy is consumed by the vehicle motor, it is important to jointly design the energy management system, computing system, and battery to achieve energy-saving driving.

7.7. Computation hardware

Today, the majority of representative CV computing hardware is designed based on GPU, FPGA, Digital Signal Processor (DSP), and Application-Specific Integrated Circuit (ASIC). They have higher processing speed and energy efficiency, such as Texas Instruments'

TDA, NVIDIA DRIVE AGX and Jetson AGX Orin.

Nevertheless, the computing hardware of CVs still has several open problems to be solved. First, it is important to determine the maximum computing speed that can be achieved by hardware with limited processing power. Secondly, CV's computing platform may be composed of multiple heterogeneous computing hardware. Therefore, how to efficiently manage computing resources between heterogeneous hardware and dynamically schedule applications are deserved the researchers' attention. Moreover, it is also essential to evaluate the suitability of the hardware system in a specific application scenario. In addition, the average cost of building a traditional non-luxury vehicle is around $30 K, while for a CAV, the total cost is almost $250 K [23], of which the cost of sensors and computing platforms is approximately two-thirds of the total price. Hence, designing a reasonably priced hardware system for CVs is imperative.

7.8. Security and privacy

With the reliance on diverse technologies, the security problem of CVs has evolved from the hardware damage of conventional vehicles to comprehensive security requiring multi-domain knowledge [24]. Here, we list and discuss several security issues closely related to CVs, current attacking methods, and mainstream response approaches.

7.8.1. Sensing security

The security of vehicle sensors is of paramount importance for CVs. In general, spoofing attacks and jamming attacks are two main attack types for a wide variety of sensors [25,26]. For example, a spoofing attack can generate interference signals, which may make the vehicle detect false obstacles [27]. To protect sensing security, redundant sensors and randomized signals (*e.g.*, LiDAR and radar) are widely used [28,29]. Besides, the GPS can check signal characteristics [30] and authenticate data sources [31] to prevent attacks. In addition, sensing data fusion is an effective mechanism to protect sensing security.

7.8.2. Data security

Data security refers to preventing data leakage from the perspective of data storage and data transmission [32]. Cryptography is a mainstream method to protect data storage (such as encrypted databases [33] and file systems [34]). In addition, access control technology is also an effective method [35]. For example, an access control framework with different access control models has been proposed to protect real-time and historical data for CVs [36].

7.8.3. Communication security

Communication security includes the security of external communication and internal communication. Currently, internal communication, such as CAN, LIN, and FlexRay, are facing serious security threats [37–39], and external communication has been studied in VANETs with V2X communications [40]. Cryptography is a mainstream solution to guarantee the transmitted data is integrated, confidential, and authenticated [41]. However, the usage of cryptography is limited due to the high computational cost, which is difficult to be implemented in resource-constrained electronic control units (ECUs). In this context, another solution is to leverage the gateway to prevent access without permission [42].

7.8.4. Control security

With the electronicization of vehicles, drivers can control their vehicles (such as opening doors) through apps or voice. Nevertheless, this has also led to a new attack surface, with various attack methods [25], such as replay attacks and jamming attacks, etc. For example, for those vehicles that support voice control, an attacker can successfully control the vehicle by using a voice that humans cannot hear [43]. Typically, these attacks can be classified as communication security, data security, and sensing security, which can be addressed by the appropriate protection mechanisms.

7.8.5. Privacy

CVs can generate enormous amounts of data per day, which usually contains private information. For example, an attacker can obtain the location information directly from the captured GPS data [44]. The straightforward solution is to prevent data leaking (*e.g.*, access control [35,36] and data encryption [25]). Besides, data desensitization is also widely used to protect privacy, such as anonymization and differential privacy [45].

7.9. Transmission cost

As the number of vehicles increase, the transmission cost associated with uplink (data update) and downlink (soft-ware/firmware update) is arguably one of the most important factors for automotive manufacturers to determine who pay for the resulting cost and therefore make thoughtful plans. According to a latest Guidehouse Insights report [46], 10 million vehicles can transmit more than 20 PB of data and lead to over $1 billion cost annually [47].

Specifically, as to uplink, previous work projects that, by 2023, a single CV will generate 40 TB data, and 8 GB data volume will be transmitted from the vehicle per day on average [48]. As to downlink, take the in-vehicle infotainment system (IVI) update as an example, the update file size can be 500 MB per vehicle, per update, on average [47].

In this context, MNOs, such as AT&T and Verizon, usually offer two different types of cost plans specifically for vehicles: (*i*) **Cost per usage**: Based on a report by Spendemont [47], the average price of 1 GB of mobile data worldwide is $8.53, and U.S. has a relatively high cost where 1 GB can cost up to $12.37 (Canada: $12.02 per GB; China: $9.89 per GB; Japan: $8.34 per GB). (*ii*) **Unlimited prepaid**

data plan: Wi-Fi/LTE plans for vehicles start usually as $20 per month and may include unlimited data. For example, Chevy is the first mass-market automaker to offer a prepaid unlimited data plan costing $20 per month.

Based on these statistics and considering an average of one software/firmware update every quarter [47], even using the most affordable way to pay for the transmission costs, a single CV every year will lead to a cost of $240 and $25 for uplink and downlink, respectively. For millions of vehicles in a fleet, the data transmission costs can quickly become astronomical, which in turn calls for technologies of Edge Computing in connected mobility. Edge Computing allows vehicles to have their data processes in much closer proximity and reduce unnecessary data transmission. According to Analysys Mason [49], enterprises across industries can expect a 10–30% reduction in costs from using Edge Computing and an average operational cost savings of 10–20%. However, how to break down the barriers to automotive edge adoption is still an open problem.

8. Conclusion

In this article, we envision that future CVs will be serving as a mobile computing platform in addition to their conventional transportation role for the next century. We present the concept of Vehicle Computing. Then we depict several reasons why Vehicle Computing is important and emerging, followed by several case studies to further illustrate our vision. Next, we introduce SDV, a key technology to Vehicle Computing. Then, we present a transformative Vehicle Computing business model, which has the potential to open new revenue streams. Finally, related challenges are also discussed, and a quantitative analysis of data transmission cost is presented.

Acknowledgments

We thank our collaborators and industry partners for their feedback and discussion during the process of writing this paper, and this work is supported in part by National Science Foundation (NSF) grant 2113817.

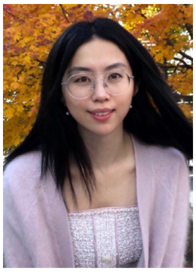
Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A.M. Research, Connected Car Market Size, Share, Growth & Trends Analysis Report by Technology, Connectivity Solution, Service, End-Use, and Segment Forecasts, 2020-2027, April 2020. https://reports.valuates.com/market-reports/ALLI-Manu-3Z1/connected-car.
[2] Automotive Edge Computing Consortium (AECC), Distributed computing in an AECC system (White Paper), August 2021, Version 1.0.0. https://aecc.org/resources/publications/.
[3] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, IEEE Internet of Things Journal 3 (5) (2016) 637–646.
[4] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, T. Weil, Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions, IEEE Communications Surveys & Tutorials 13 (4) (2011) 584–616.
[5] S. Lu, Y. Yao, W. Shi, CLONE: collaborative learning on the edges, IEEE Internet of Things Journal 8 (13) (2021) 10222–10236.
[6] F. Götz, The data deluge: What do we do with the data generated by AVs?, January 2021, https://blogs.sw.siemens.com/polarion/the-data-deluge-what-do-we-do-with-the-data- generated-by-avs/.
[7] American Society of Civil Engineers (ASCE), Report card for America's infrastructure (online), 2021. https://www.asce.org/topics/report-card-for-americas-infrastructure.
[8] M. Insider, What is a software-defined vehicle? (online), March 2020. https://www.aptiv.com/en/insights/article/what-is-a-software-defined-vehicle.
[9] L. Bauer, White Paper: Smart vehicle architecture: A sustainable approach to building the next generation of vehicles, in: APTIV White Paper (online), March 2020, https://www.aptiv.com/en/insights/article/white-paper-smart-vehicle-architecture-overview.
[10] S. Halder, A. Ghosal, M. Conti, Secure over-the-air software updates in connected vehicles: A survey, Computer Networks 178 (2020), 107343.
[11] H. Laguna, Recap of 2020 recalls reveals impact of pandemic on compliance and the continuing threat (online), May 2020. https://www.recallmasters.com/recap-of-2020-recalls-reveals-impact-of-pandemic-on-compliance-and-the-continuing-threat/.
[12] C. Isidore, P. Valdes-dapena, Hyundai's recals 82,000 electric cars is one of the most expensive in history (online), February 2021. https://www.kktv.com/2021/02/26/hyundais-recals-82000-electric-cars-is-one-of-the-most-expensive-in-history.
[13] G. Dennis, R. Barga, N. Sundaresan, Cloud-native applications, IEEE Cloud Computing 4 (5) (2017) 16–21.
[14] Toyota Newsroom, Toyota and Amazon Web Services Collaborate on Toyota's Mobility Services Platform (online), August 2020. https://pressroom.toyota.com/toyota-andamazon-web-services-collaborate-on-toyotas-mobility-services-platform/.
[15] FutureCar, Toyota to launch its 'Arene' vehicle operating system in models by 2025, February 2022. https://www.futurecar.com/5107/Toyota-to-Launch-its-Arene-Vehicle-Operating-System-in-Models-by-2025.
[16] G. Wilson, Ford & Google collaborate to accelerate auto innovation, February 2021. https://manufacturingdigital.com/smart-manufacturing/ford-and-google-collaborate-accelerate-auto-innovation.
[17] Microsoft News Center, Cruise and GM team up with Microsoft to commercialize self-driving vehicles (online), January 2021. https://news.microsoft.com/2021/01/19/cruise-and-gm-team-up-with-microsoft-to-commercialize-self-driving-vehicles/.
[18] O. Wehring, Supplier confirms SAIC/Alibaba premium EV JV plans, January 2021. https://www.just-auto.com/news/supplier-confirms-saic-alibaba-premium-ev-jv-plans.
[19] Q. Zhang, Y. Wang, X. Zhang, L. Liu, X. Wu, W. Shi, H. Zhong, OpenVDAP: An open vehicular data analytics platform for CAVs, in: Proceedings of the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), IEEE, Los Alamitos, 2018, pp. 1310–1320.
[20] J. Harding, G. Powell, R. Yoon, J. Fikentscher, C. Doyle, D. Sade, M. Lukuc, J. Simons, J. Wang, et al., Vehicle-to-vehicle communications: readiness of V2V technology for application, August 2014. https://rosap.ntl.bts.gov/view/dot/27999.
[21] Teraki, Autonomous cars' big problem: The energy consumption of edge processing reduces a car's mileage with up to 30%, May 2019. https://medium.com/@teraki/energy-consumption-required-by-edge-computing-reduces-a-autonomous-cars-mileage-with-up-to-30-46b6764ea1b7.
[22] R. Bryce, How google powers its 'monopoly' with enough electricity for entire countries, October 2020. https://www.forbes.com/sites/robertbryce/2020/10/21/googles-dominance-is-fueled-by-zambia-size-amounts-of-electricity/?sh=1beeb7f968c9.

[23] S. LeVine, What it really costs to turn a car into a self-driving vehicle, March 2017. https://qz.com/924212/what-it-really-costs-to-turn-a-car-into-a-self-driving-vehicle.
[24] L. Liu, S. Lu, R. Zhong, B. Wu, Y. Yao, Q. Zhang, W. Shi, Computing systems for autonomous driving: State of the art and challenges, IEEE Internet of Things Journal 8 (8) (2021) 6469–6486.
[25] K. Ren, Q. Wang, C. Wang, Z. Qin, X. Lin, The security of autonomous driving: Threats, defenses, and future directions, Proceedings of the IEEE 108 (2) (2020) 357–372.
[26] E. Yağdereli, C. Gemci, A.Z. Aktaş, A study on cyber-security of autonomous and unmanned vehicles, The Journal of Defense Modeling and Simulation 12 (4) (2015) 369–381.
[27] C. Yan, W. Xu, J. Liu, Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle, DEF CON 24 (8) (2016) 109.
[28] H. Shin, D. Kim, Y. Kwon, Y. Kim, Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications, in: Proceedings of the Cryptographic Hardware and Embedded Systems–CHES 2017, Springer International Publishing, Cham, 2017, pp. 445–467.
[29] J. Petit, B. Stottelaar, M. Feiri, F. Kargl, Black Hat Europe, November 2015. https://www.blackhat.com/eu-15/briefings.html.
[30] A. Konovaltsev, M. Cuntz, C. Häettich, M. Meurer, Autonomous spoofing detection and mitigation in a GNSS receiver with an adaptive antenna array, in: Proceedings of the 26th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2013), ION, Manassas, 2013, pp. 2937–2948.
[31] B.W. O'Hanlon, M. L Psiaki, J.A. Bhatt, D. P Shepard, T.E. Humphreys, Real-time GPS spoofing detection via correlation of encrypted signals, Navigation 60 (4) (2013) 267–278.
[32] H. Zhong, L. Pan, Q. Zhang, J. Cui, A new message authentication scheme for multiple devices in intelligent connected vehicles based on edge computing, IEEE Access 7 (2019) 108211–108222.
[33] R.A. Popa, C.M.S. Redfield, N. Zeldovich, H. Balakrishnan, CryptDB: Protecting confidentiality with encrypted query processing, in: Proceedings of the twenty-third ACM symposium on operating systems principles, Association for Computing Machinery, New York, 2011, pp. 85–100.
[34] M. Blaze, A cryptographic file system for UNIX, in: Proceedings of the 1st ACM conference on Computer and communications security (CCS '93), Association for Computing Machinery, New York, 1993, pp. 9–16.
[35] R.S. Sandhu, P. Samarati, Access control: Principle and practice, IEEE Communications Magazine 32 (9) (1994) 40–48.
[36] Q. Zhang, H. Zhong, J. Cui, L. Ren, W. Shi, AC4AV: A flexible and dynamic access control framework for connected and autonomous vehicles, IEEE Internet of Things Journal 8 (3) (2020) 1946–1958.
[37] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, Experimental security analysis of a modern automobile, in: Proceedings of the 2010 IEEE Symposium on Security and Privacy, IEEE, Piscataway, 2010, pp. 447–462.
[38] J.M. Ernst, A.J. Michaels, LIN bus decurity analysis, in: Proceedings of the IECON 2018- 44th Annual Conference of the IEEE Industrial Electronics Society (IECON 2018), IEEE, Piscataway, 2018, pp. 2085–2090.
[39] D.K. Nilsson, U.E. Larson, F. Picasso, E. Jonsson, A first simulation of attacks in the automotive network communications protocol FlexRay, in: Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems (CISIS'08), Springer, New York City, 2008, pp. 84–91.
[40] I. Ali, A. Hassan, F. Li, Authentication and privacy schemes for vehicular ad hoc networks (VANETs): A survey, Vehicular Communications 16 (2019) 45–61.
[41] D.R. Stinson, M. Paterson, Cryptography: Theory and Practice, CRC Press, Boca Raton, 2018.
[42] J.H. Kim, S. Seo, N. Hai, B.M. Cheon, Y.S. Lee, J.W. Jeon, Gateway framework for in-vehicle networks based on CAN, FlexRay, and ethernet, IEEE Transactions on Vehicular Technology 62 (10) (2015) 4472–4486.
[43] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, W. Xu, Dolphin attack: inaudible voice commands, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17), Association for Computing Machinery, New York, 2017, pp. 103–117.
[44] Z. Xiong, W. Li, Q. Han, Z. Cai, Privacy-Preserving Auto-Driving: A GAN-Based Approach to Protect Vehicular Camera Data, in: Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM), IEEE, Los Alamitos, 2019, pp. 668–677.
[45] F. Martinelli, F. Mercaldo, A. Orlando, V. Nardone, A. Santone, A.K. Sangaiah, Human behavior characterization for driving style recognition in vehicle system, Computers & Electrical Engineering 83 (2020), 102504.
[46] A. Sam, Automotive over-the-air updates: A cost consideration guide (online), April 2021, https://www.auroralabs.com/wp-content/uploads/2021/05/OTA_Update_Cost_ Consideration_Guide_Apr2021.pdf.
[47] T. Lida, Is data transmission the new fuel?, (online), June 2021. https://www.auroralabs.com/is-data-transmission-the-new-fuel/.
[48] O. Joel, Connected car-all that data-cost and impact on the network (online), February 2019. https://blogs.cisco.com/sp/connected-car-all.
[49] Y. Gorkem, Edge computing: Operator strategies, use cases and implementation (online), July 2020. https://www.cisco.com/c/dam/en/us/solutions/service-provider/edge-computing/pdf/white-paper-sp-analysys-mason-research.pdf.



Sidi Lu received a B.E. degree from Xidian University, Xi'an, China, in 2016. She is currently working toward a Ph.D. degree in the Department of Computer Science at Wayne State University, Detroit, USA. Her academic advisor is Prof. Weisong Shi, and her general research interests span edge computing and applied artificial intelligence, with the objective of making computer systems (such as connected vehicles and IoT devices) more reliable, scalable, secure, and efficient. She has received all prestigious awards given to Ph.D. students, including the Ralph H. Kummler Award for Distinguished Graduate Research Achievement (offered by the College of Engineering) and Michael E. Conrad Graduate Research Award (offered by the Department of Computer Science).

Weisong Shi is a Professor and Chair of the Department of Computer and Information Sciences at the University of Delaware (UD), where he leads the Connected and Autonomous Research (CAR) Laboratory. Dr. Shi is an internationally renowned expert in edge computing, autonomous driving and connected health. His pioneer paper entitled "Edge Computing: Vision and Challenges" has been cited more than 5000 times. Before he joins UD, he was a professor at Wayne State University (2002–2022) Dr. Shi has published more than 270 articles in peer-reviewed journals and conferences and served in editorial roles for more than 10 academic journals and publications, including EIC of Smart Health, AEIC of IEEE Internet Computing Magazine. He is a fellow of IEEE, and a distinguished member of ACM. More information can be found at http://weisongshi.org.