Survey on Integrated Vehicular Platforms for Next Generation Mobility

Chaeyeong Lee, Moonbeom Kim, Hyewon Seo, and Jeongyeup Paek
Department of Computer Science & Engineering
Chung-Ang University
Seoul, Republic of Korea
{cxaexeong, mbkim, hyewon981019, jpaek}@cau.ac.kr

***Abstract*—The vehicle is becoming a computing platform rather than mechanical machinery. Vehicles are now being equipped with advanced electronic functions such as autonomous driving, in-vehicle infortainment (IVI), and driver assistant systems together with communication capabilities. To this end, studies on open platforms for vehicles to resolve dependencies between vehicle software and hardware have been actively conducted. In this paper, we first investigate the research on representative vehicle operating systems and software platforms, *Automotive Open System Architecture* (AUTOSAR) and *Connected Vehicle Systems Alliance* (COVESA). We review academic research papers published by major academic publishers and categorize them according to their purposes. We also investigate recent research trends on *open vehicle platforms* by major automobile companies and research institutes to provide insight into future directions for next-generation mobility.**

***Index Terms*—OSEK/VDX, AUTOSAR, COVESA, SOME/IP, Autonomous Vehicle Platform**

I. Introduction

Vehicle, conventionally a simple means of transportation, is becoming our personal space today. With this paradigm shift, In-Vehicle Infotainment (IVI) system is expanding to serve the new features in terms of time and space beyond the existing roles of vehicles. In addition, technologies fused with sensors such as human machine interface (HMI) or advanced driver assistance systems (ADAS) are being actively developed to enhance driver safety and convenience. In these systems, various functions are controlled by multiple *electronic control units* (ECUs) with different operating systems. Therefore, as function more diversity, it is inevitable to install additional ECUs in the vehicle to support them. However, adding ECUs is not only practically difficult due to the space constraint but also increases operational complexity and management costs. For this reason, the need to integrate the technical standards for vehicle systems have emerged.

To improve the portability and reusability of application software, OSEK/VDX[1]standardized the operating system (OSEK OS), communication stack (OSEK COM), and network protocol (OSEK NM) targeting in-vehicle embedded systems [1]. However, OSEK/VDX is inadequate to become the foundation of a standard platform due to compatibility issues that require software to be developed according to the hardware design of each ECU manufacturer. With the goal of providing standardized software architecture, the *automotive open system architecture* (AUTOSAR) development partnership was concluded in 2003 [2]. AUTOSAR is actively working to improve the electric/electronic architecture and uses middleware to resolve hardware dependencies through layer separation and virtualization. *Connected vehicle systems alliance* (COVESA) recognized the necessity of the connected car specialized platform and was established in 2016 actively share data between different manufacturers' vehicles [3]. They provide the open source vehicle platform to eliminate vendor dependencies, aiming to integrate computing domains of IVI systems on vehicles.

Vehicle platforms have become a popular topic in both industry and academia. To understand the recent research directions and trends, in this work, we review the research papers and categorize these studies according to the topic and purpose. We also show the current state of vehicle platform development in industrial fields. To the best of our knowledge, there is no comprehensive study about vehicle platforms yet. We believe that our work provides introductory information about vehicle platforms.

II. Overview

In this section, we describe the representative vehicle platforms, AUTOSAR and COVESA, and provide an overall understanding of them.

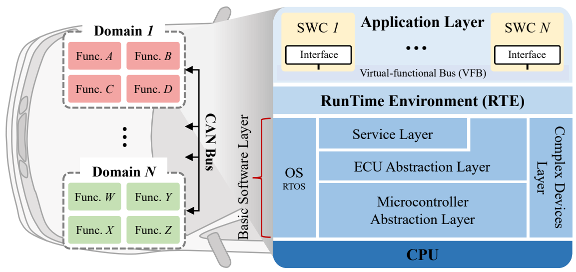
A. AUTOSAR

AUTOSAR proposes automotive standard software architecture for eliminating hardware dependency and increasing software scalability. We introduce architecture and key specifications for *classic* and *adaptive* platforms provided by AUTOSAR, summarizing the differences between the two platforms in Table. I.

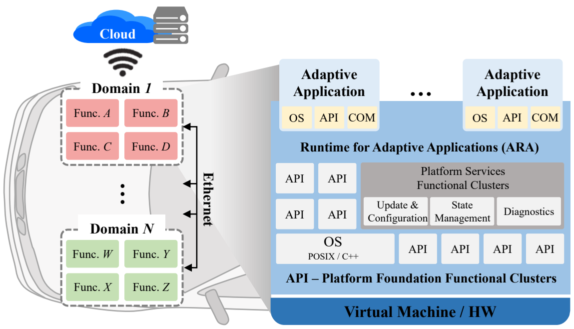
Classic platform (CP) separates between a *basic software layer* (BSW) and an *application layer* through a *runtime*

This research was supported by the MIST(Ministry of Science and ICT), Korea, under the National Program for Excellence in SW supervised by the IITP(Institute for Information & communications Technology Promotion) (20170001000061001), and also by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(No. 2022R1A4A5034130).

[1]In 2005, OSEK/VDX is combined OSEK and VDX developed in Germany and France respectively.



(a) Classic platform



(b) Adaptive platform

Fig. 1. AUTOSAR Classic and Adaptive platform architectures.

TABLE I
SUMMARY OF THE DIFFERENCE BETWEEN CLASSIC AND ADAPTIVE

Platform / Specification	Classic	Adaptive
ECU	ECU allocation per function domain	Centralized processing
OS	RTOS	POSIX
Task	Developer-defined static foundation	Dynamic foundation by operating system
Resource	Static allocation	Dynamic allocation
Communication method	Signal-oriented	Service-oriented
S/W supporting	No software added or updated	Support for adding and updating software

environment (RTE, a middleware), which allows platforms to independently implement different applications. Fig. 1(a) shows the overall architecture of AUTOSAR CP. *Software components* (SWCs) in the application layer intercommunicate using the virtual-function bus (VFB). CP uses signal-oriented communication based on static resource allocation over a bus network (*i.e.,* CAN, LIN, and FlexRay) for interoperating between each ECU.

AUTOSAR CP reflects the standards by OSEK/VDX, and mechanisms have been extended to accommodate more functions. For instance, OSEK OS is adopted to support functions on the OS, such as resource and event management, alarms, counters, and error handling. After release 4.0, AUTOSAR added a specification for a multi-core embedded real-time operating system (RTOS) that can simultaneously perform different functions as advanced vehicles evolve. At the heart of this specification is the *spinlock* mechanism for synchronizing tasks between cores, whether or not each CPU core is identical. For network reliability and stability, CP provides *network management* (NM) mechanisms, which include OSEK NM (direct/indirect NM) and AUTOSAR NM (CAN NM). *Direct NM* allows nodes to check each other status by sending management messages in a token-based logical ring method. On the other hand, *indirect NM* periodically sends application messages to check the status of nodes attached to a network. In the case of *CAN NM*, it is based on periodic message transmission but has several modes of operation, such as (Prepare) Bus-sleep mode and network mode (repeat message state, normal operation state, and ready sleep state).

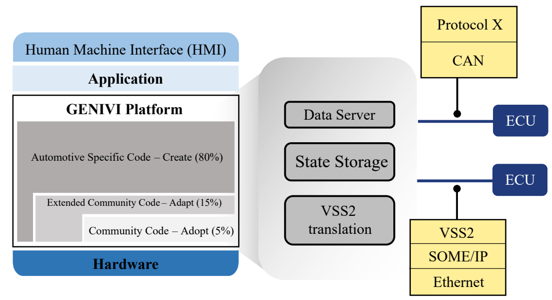


Fig. 2. COVESA (GENIVI) Platform architecture.

AUTOSAR also has provided a specification for a time synchronization mechanism based on IEEE 802.1AS [4], achieving the precise time synchronization of all ECUs connected over Ethernet. The objective of the specification is to synchronize time bases and their corresponding Ethernet messages. Above this, AUTOSAR has standardized cryptographic stacks to model keys in software and use cryptographic services for cybersecurity. It is applied to the *Secure Onboard Communication* (SecOS) module to manage the key and is used to verify in-vehicle communication between ECUs [5].

Adaptive Platform (AP) developed to overcome the limitations of CP, because signal-based communication directed by the CP platform may cause a problem of network overload. It provides high-performance computing power and flexible communication for the infotainment of future mobility systems. As shown in Fig. 1(b), the intermediate layer of the AP consists of three basic services (gray box) and eight functional clusters (written as API). Three basic services provided by middleware called an *ara::com* and eight functional clusters communicate with other applications through each API.

With the introduction of AP, the biggest difference between the two platforms is the communication method. The AP adopts a service-oriented communication method that supports the Scalable service-Oriented MiddlewarE over IP (SOME/IP) protocol [6]. SOME/IP is an automotive/embedded communication protocol that supports event notifications, the underlying serialization/wire format, and remote procedure calls. SOME/IP also is used for client/server serialization between ECUs. It can be applied to other operating systems (*i.e.,* AUTOSAR, COVESA, and OSEK), and even to embedded devices that do not have an operating system.

The AP uses the portable operating system interface (POSIX) [7]. Compared to RTOS, it has much higher computational processing power, and multi-core process support makes automotive software development much more flexible.

B. COVESA

In the past, several systems such as ADAS and IVI distribute to separate computing domains. In 2016, COVESA appeared to drive each of these computing domains into the integrated platform, toward an open standard community. The COVESA platform is based on open source provided by the Linux Foundation for 80 % of the current underlying platform, with about 15 % modifications and complements to other existing open source modules to IVI. Only 5 % of modules were developed specifically for vehicles by COVESA. Fig. 2 shows overveiw of COVESA.

III. Latest Research Trend

In this section, we introduce the studies that have contributed to the advancement of vehicle platforms. We categorize these studies into five subtopics according to the purpose (*i.e.,* what were they trying to optimize or improve) and provide a summary of research trends by category in *'academia'* field in the Table. II.

A. Task and Network Management

As the complexity of automotive ECU networks increases, there are considerations. First, multi-core systems are an alternative to running various functions in parallel to minimize power consumption and complete faster than single-core CPUs. AUTOSAR constitutes a mechanism that allows tasks to be mapped to other cores while the memory exchanged between tasks is usually protected by a spinlock. Unfortunately, the spinlock mechanism is essentially used only for short critical sections. Thus, R. Hottger *et al.* [8] proposes Time-Release-Delta-based Runnable Reordering (TDRR). TDRR provides a viable sequence for tasks to reduce busy latency depending on specific system conditions. At the same time, TDRR also considers feasible dependencies, so functional behavior is maintained and re-verification is not required.

Preemption Threshold Scheduling (PTS) is an effective technique to reduce stack memory usage by selectively disabling proactive between task pairs. Q. Zhao *et al.* [9] specifies the following two algorithms to minimize stack use of mixed systems using PTS in the AUTOSAR model;

- PA-DMMPT assigns the current priority to the task with the largest blocking time limit among the remaining tasks. This assigns the one with the smallest reduction in interference from higher priority tasks if the blocking time limit for all tasks are negative.
- HeuPADMMPT based on Heuristics defines priority allocation on the basis of PA-DMMPT and then explores task merge opportunities.

Second, real-time network management is one of the most important mechanisms for ensuring communication safety between ECUs. As the number of ECUs mounted on vehicles increases significantly, the complexity of the network increases. Consequently, the number of nodes to be connected increases, and the probability of network failure occurrence is higher as variable functions are added. Therefore, wei *et al.* [10] proposes a dynamic network management framework based on the OSEK NM protocol so that it can automatically adapt to unstable network loads. When the network load has occurred, the dynamic framework is complete by using it as a core parameter of the network management mode. They also add a method for calculating thresholds for appropriate loads. There are two main methods of obtaining load values (Counter based and Packet capturing). Due to fundamental real-time constraints in mission-critical components in an automotive network, the latter requires more resources than the former. Accordingly, the authors of this paper adopt the counter-based method and the most famous network load measurement is periodic measurement. Consequently, in the paper, the network load is defined by the total number of messages transmitted from the network at a specific time zone. In summary, it is possible to dynamically switch the network management mode by measuring the current network load. Based on this, the proposed framework meets the necessary network information gathering requirements without sacrificing the performance of the vehicle ECU network. A conventional OSEK NM network system is based on a static logical ring and cannot adaptively adjust work performance according to changes in network load. Therefore, wang *et al.* [11] present the Logical Ring Dynamic Adjustment Algorithm (LADR); *that Dynamically adjusts the radius of different logical rings by varying the length of the timer* after analyzing the operating mechanism of OSEK NM. In this mechanism, they have used two network load flavors (average load and momentary load) and Monitoring periodic network load systems. Then, set the mapping relationship between the instantaneous load factor of the BUS network and the radius of the logical ring to ensure optimum performance.

B. Co-operability

Research is underway to evaluate the interoperability of communication standards and present solutions in different operating systems adopted by each company. M. Bellanger *et al.* [12] suggest guidelines for ensuring compatibility between AUTOSAR and COVESA based on SOME/IP (respective middleware ara:com and COMMONAPI), a communication standard used in the automotive industry. Since both organizations use SOME/IP communication, the Linux API is guaranteed, but another critical operating system to consider here is Android. Both the APIs of Android and Linux implement the same communication protocol at the bottom layer, but the difference in the serialization of payload data at the runtime layer is still likely to cause problems. Thereby, L. Bilac *et al.* [13] specify Android's API implements the SWC (Communication Manager) on the ADAS side to see if this communication is possible.

C. Security

The Internet of Vehicles (IoV) plays a very critical role in the field of intelligent transportation. The in-vehicle network is

TABLE II
SUMMARY OF LATEST RESEARCH TRENDS AND ENTERPRISE

Sortation	Category	Summary	Reference
Academia	Task and Network Management	Presents methods and algorithms for multi-core OS and network load management.	[8], [9], [10], [11]
	Co-Operability	Interoperability between different operating systems.	[12], [13]
	Security	CAN protocol security study for complex network improvement.	[14], [15], [16]
	Energy Consumption	Plan to meet overall power requirements with limited resources.	[17], [18]
	Synchronization	Develop a way to correlate different timestamps.	[19]
Industry	Motional	E-GMP generation and commercialization test of its own electric vehicle platform.	[20]
	Tesla	Create Tesla's own OS and platform for ADAS. It has the most data among autonomous vehicle companies.	[21]
	Toyota	e-Palette, a multi-purpose modular electric vehicle. Partnerships with various companies to promote commercialization projects.	[22]
	Baidu	Introducing the real-time motion planning system, learning-based vehicle dynamic modeling procedure.	[23], [24]

a key in intelligent transportation consisting of CAN. However, a malicious attacker can exploit a security loophole in the CAN protocol to cause a deliberate malfunction. Concerned about the validity of the issue, H. Zhang *et al.* [14] highlighted the vulnerability of CAN protocols that threaten the safety of drivers and occupants due to several vulnerabilities without encryption, authentication, and integrity checks. To compensate for this problem, P. Biondi *et al.* [15] presents TOUCAN, a security protocol, in compliance with AUTOSAR "Secure Onboard Communication", *profile 1*, standard. TOUCAN leverages the Chaskey MAC (Message Authentication Code) to provide authentication and integrity to CAN frames [16].

D. Energy Consumption

It is important to meet the power requirements to develop automated embedded systems with limited resources. N. Mahmud, G *et al.* [17] aim to minimize the total power consumption of applications distributed across multiple computing devices. For this work, they stipulate Integer Linear Programming for the allocation of fault-tolerant developed using the AUTOSAR standard. Kehr *et al.* [18] also present a practical parallelism approach technique called Parcus to demonstrate the energy-saving potential of multi-core processors.

E. Synchronization

Most synchronizations are used in time relationships and also help identify in-vehicle networks. Raju *et al.* [19] utilize multiple data logging devices and time synchronization distributed over the vehicle network for its on-board fault diagnosis system. If time synchronization between Time Master and Time Slave is successful, the synchronized moments are used to filter fault data logging. The time synchronization mechanism used in this paper enables offline analysis by establishing a meaningful temporal relationship between timestamps in the defected record.

IV. INDEPENDENT PLATFORM DEVELOPMENT TRENDS

There are also many efforts to develop a platform suitable for autonomous vehicles. S.-C. Lin *et al.* [25] presented the design constraints for building autonomous driving systems in terms of performance, power, and so on. To find trade-offs between the designing and building of autonomous driving systems, they implement the representative end-to-end system by using state-of-the-art algorithms (award-winning) and identify the actual design constraints of systems [26]. They also reduced the tail latency of the system by using acceleration systems such as GPU, FPCA, and ASIC.

Furthermore, many vehicle manufacturers are working on the development of their platforms to prepare for fully autonomous driving in the future. In this section, we examine the recent vehicle platform trends for autonomous driving with a focus on the industrial field; The *'industry'* field in the Table. II summarizes the current state of the manufacturers.

- *Motional*: The electric global modular platform (E-GMP) is a platform dedicated to electric vehicles that leads Hyundai Motor Group's development of the next-generation electric vehicle. The free battery module configuration to match the car class, car model, and customer's lifestyle provides a foundation for running longer distances. In particular, Ioniq 5, which has self-driving level 4, has recently started self-driving delivery for Uber Eats customers in Santa Monica, California [20].
- *Tesla*: Tesla launched Autopilot, a semi-autonomous driver support technology, in 2014. Since then, it has supported software updates for ADAS. In particular, Tesla has an overwhelming amount of self-driving data compared to other companies. Through this, it has already established its operating system. It also provides its autonomous driving platform, and car owners can remotely upgrade their software. Above all, once the StarLink project is complete, it will be possible to download the latest software or use various functions anywhere [21].
- *Toyota*: The vision of the 'e-Palette Concept' that can show Toyota's future mobility strategy was announced at

CES 2018. e-Palette is a multi-purpose modular electric vehicle that can provide necessary mobility services in various industries by utilizing connected and autonomous driving technologies [22]. In addition, to strengthen the platform, it has formed partnerships with companies such as Amazon, Pizza Hut, and Uber to push for projects from the service planning stage to the commercialization stage.
- *Baidu*: H. Fan, F *et al.* [23] introduced a real-time motion planning system based on China's Baidu Apollo autonomous driving platform. The paper deals with autonomous driving for multiple lanes and single lanes in a hierarchical manner. The described system has been distributed to dozens of Baidu Apollo autonomous vehicles since Apollo 1.5 was announced in September 2017. We also introduce a highly automated learning-based vehicle dynamic modeling procedure for large-scale vehicle dynamics modeling on the Baidu Apollo platform [24].

V. Conclusion

Tremendous effort on development of integrated autonomous vehicle platform are underway in both industry and academia. Along with its functional strengths of interoperability and software integration, the fact that it is based on Ethernet, which is drawing attention as a next-generation vehicle communication, has endless possibilities. With these advantages, AUTOSAR and COVESA are expected to contribute to providing platform standards that will serve as the foundation for future leading technologies for intelligent transportation systems. Throughout this paper, we conducted a comprehensive survey, including materials in addition to reference publications, which is the first of its kind to our knowledge. We aim this paper to serve as a pointer and directional guide to many fellow researchers who wish to study autonomous vehicle platform in the future.

References

[1] "OSEK/VDX," [last accessed on August 2022]. [Online]. Available: https://www.osek-vdx.org/

[2] "AUTomotive Open System ARchitecture," [last accessed on August 2022]. [Online]. Available: https://www.autosar.org/

[3] "Connected Vehicle Systems Alliance," [last accessed on August 2022]. [Online]. Available: https://www.covesa.global/

[4] "Time Synchronization Protocol Specification AUTOSAR FO R20-11," 2020, [last accessed on August 2022]. [Online]. Available: https://www.autosar.org/fileadmin/user_upload/standards/foundation/20-11/AUTOSAR_PRS_TimeSyncProtocol.pdf

[5] "Specification of Secure Onboard Communication AUTOSAR CP Release 4.3.1," 2017, [last accessed on August 2022]. [Online]. Available: https://www.autosar.org/fileadmin/user_upload/standards/classic/4-3/AUTOSAR_SWS_SecureOnboardCommunication.pdf

[6] "SOME/IP Protocol Specification AUTOSAR FO R19-11," 2019, [last accessed on August 2022]. [Online]. Available: https://www.autosar.org/fileadmin/user_upload/standards/foundation/19-11/AUTOSAR_PRS_SOMEIPProtocol.pdf

[7] "Specification of Operating System Interface AUTOSAR AP Release 17-10," 2017, [last accessed on August 2022]. [Online]. Available: https://www.autosar.org/fileadmin/user_upload/standards/adaptive/17-10/AUTOSAR_SWS_OperatingSystemInterface.pdf

[8] R. Höttger, B. Igel, and O. Spinczyk, "On reducing busy waiting in AUTOSAR via task-release-delta-based runnable reordering," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2017, pp. 1510–1515.

[9] Q. Zhao, Z. Gu, and H. Zeng, "Design optimization for AUTOSAR models with preemption thresholds and mixed-criticality scheduling," *Journal of Systems Architecture*, vol. 72, pp. 61–68, 2017.

[10] C. Wei, M. Yao, P. Lu, Q. Hu, and N. Zheng, "OSEK/VDX-based dynamic network management on automotive network," in *2009 International Conference on Embedded Software and Systems*. IEEE, 2009, pp. 131–137.

[11] Y. Wang, B. Pan, Y. Li, F. Huang, and Z. Guo, "Real-time dynamic logic loop mechanism of vehicular network management based on instantaneous load sensing," in *2019 12th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 2. IEEE, 2019, pp. 27–30.

[12] M. Bellanger and E. Marmounier, "Service Oriented Architecture: impacts and challenges of an architecture paradigm change," in *10th European Congress on Embedded Real Time Software and Systems (ERTS)*, 2020.

[13] L. Bilac, D. Stanisic, D. Kenjic, and M. Antic, "One Solution of an Android In-Vehicle Infotainment Service for Communication with Advanced Driver Assistance System," in *45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*. IEEE, 2022, pp. 1420–1425.

[14] H. Zhang, X. Meng, X. Zhang, and Z. Liu, "CANsec: a practical in-vehicle controller area network security evaluation tool," *Sensors*, vol. 20, no. 17, p. 4900, 2020.

[15] P. Biondi, G. Bella, G. Costantino, and I. Matteucci, "Implementing CAN bus security by TOUCAN," in *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2019, pp. 399–400.

[16] N. Mouha, B. Mennink, A. V. Herrewege, D. Watanabe, B. Preneel, and I. Verbauwhede, "Chaskey: an efficient MAC algorithm for 32-bit microcontrollers," in *International conference on selected areas in cryptography*. Springer, 2014, pp. 306–323.

[17] N. Mahmud, G. Rodriguez-Navas, H. Faragardi, S. Mubeen, and C. Seceleanu, "Power-aware allocation of fault-tolerant multirate autosar applications," in *25th Asia-Pacific Software Engineering Conference (APSEC)*. IEEE, 2018, pp. 199–208.

[18] S. Kehr, E. Quiñones, D. Langen, B. Böddeker, and G. Schäfer, "Parcus: Energy-aware and robust parallelization of AUTOSAR legacy applications," in *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2017, pp. 343–352.

[19] S. Raju, G. Jeyakumar, A. Mukherji, and J. K. Thanki, "Time synchronized diagnostic event data recording based on AUTOSAR," in *IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2017, pp. 1–6.

[20] "Motional © 2021," [last accessed on July 2022]. [Online]. Available: https://motional.com/news/motional-and-uber-eats-launch-autonomous-deliveries-in-santa-monica

[21] "Tesla © 2022," [last accessed on July 2022]. [Online]. Available: https://www.tesla.com/ko_kr

[22] "CORPORATION, T.M.(n.d.).Toyota launches new mobility ecosystem and concept vehicle at 2018 ces® — corporate — global newsroom." [last accessed on July 2022]. [Online]. Available: https://global.toyota/en/newsroom/corporate/20546438.html

[23] H. Fan, F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, and Q. Kong, "Baidu apollo em motion planner," *arXiv preprint arXiv:1807.08048*, 2018.

[24] J. Xu, Q. Luo, K. Xu, X. Xiao, S. Yu, J. Hu, J. Miao, and J. Wang, "An automated learning-based procedure for large-scale vehicle dynamics modeling on baidu apollo platform," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 5049–5056.

[25] S.-C. Lin, Y. Zhang, C.-H. Hsu, M. Skach, M. E. Haque, L. Tang, and J. Mars, "The architectural implications of autonomous driving: Constraints and acceleration," in *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*, 2018, pp. 751–766.

[26] S. Hadfield, K. Lebeda, and R. Bowden, "The visual object tracking VOT2014 challenge results," in *European Conference on Computer Vision (ECCV) Visual Object Tracking Challenge Workshop*, 2014.