

Article

Research on Advancement Constraint Screening and Cost Evaluation of Centralized Architecture Platforms for Intelligent Vehicles Under Different R&D Solutions

Wang Zhang ^{1,2} , Fuquan Zhao ^{1,2} and Zongwei Liu ^{1,2,*}

¹ State Key Laboratory of Intelligent Green Vehicle and Mobility, Tsinghua University, Beijing 100084, China; zhangwan21@mails.tsinghua.edu.cn (W.Z.); zhaofuquan@tsinghua.edu.cn (F.Z.)

² Tsinghua Automotive Strategy Research Institute, Tsinghua University, Beijing 100084, China

* Correspondence: liuzongwei@tsinghua.edu.cn

Abstract

The electronic and electrical architecture of vehicles has rapidly evolved to centralized. At present, there is no unified consensus on the R&D strategy of the platform in the industry, and there is also a lack of a quantitative decision-making framework that can be implemented. This study takes the centralized architecture platform as the research object, constructs a two-stage analysis framework of “advanced constraint screening-cost quantitative evaluation”, uses a fuzzy-set qualitative comparative analysis method to screen feasible R&D strategy combinations that meet the requirements of the architectural advancement, builds a total cost of ownership evaluation system around the software and hardware elements related to the architecture platform, and systematically analyzes the optimal cost R&D strategy combinations of car enterprises with different mass production scales under the two scenarios of Multi-Box and One-Board. The research results show that adaptive platform middleware and framework middleware are the core necessary elements to realize the advanced architecture; the amortization cost of architecture is negatively correlated with the scale of mass production, and the cost of in-house R&D is highly dependent on large-scale amortization; and there are differentiated optimal solutions in the framework selection and R&D strategy combination of automakers with different mass production scales. This study can provide quantitative reference and practical guidance for R&D decision making of centralized architecture platform for automakers.

Keywords: centralized architecture platform; R&D solution decision making; fuzzy-set qualitative comparative analysis; total cost of ownership



Academic Editors: Felipe Jiménez,
Jose Eugenio Naranjo and
Alfredo Valle

Received: 17 March 2026

Revised: 7 April 2026

Accepted: 9 April 2026

Published: 12 April 2026

Copyright: © 2026 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the [Creative Commons
Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The industrial wave of software-defined vehicles (SDVs) has driven the in-depth evolution of automotive electrical and electronic architecture (EEA) towards centralization [1]. Meanwhile, the rapid iteration and implementation of large language model technologies have further made AI-defined vehicles (AIDVs) a new trend in the industry [2]. AIDVs impose higher requirements on the vehicle’s computing power scheduling, software–hardware collaboration, and cross-domain function integration, which urgently requires original equipment manufacturers (OEMs) to develop centralized architecture platforms [3]. By constructing a human-like architectural form of “brain–nerve–organ”, these platforms can provide strong computing and communication support for the continuous upgrading of automotive intelligence [4].

The centralized architecture platform is a highly complex cyber–physical system for electrical and electronics, integrating multiple advanced software and hardware technologies such as heterogeneous chip design, operating system and middleware development, and framework integration [5]. Its R&D threshold is much higher than that of traditional distributed architectures. However, most OEMs are highly dependent on the supply chain system in traditional automotive R&D and have not mastered these core advanced technologies. Faced with the new technological challenges of centralized architecture platforms, there is no consensus on a unified R&D solution in the industry [6]: some new car-making forces adopt an aggressive full-stack in-house R&D strategy, attempting to build more advanced architectural products by independently controlling core technologies, while some traditional automotive giants continue the industrial cooperation model, tending to carry out joint development with long-term cooperative suppliers to share R&D risks and reduce technical trial-and-error costs.

Existing research on the centralized architecture platform of intelligent vehicles mostly focuses on qualitative analysis to discuss R&D strategies, lacking a systematic quantitative analysis framework that balances technical advancement and economic feasibility. It is difficult to provide actionable and differentiated R&D decision-making guidelines for OEMs with different production volumes and technical development needs. This study attempts to fill this research gap and provide scientific reference for the R&D practice of OEMs. The subsequent content of this paper is organized as follows: Section 2 combs the relevant research results on architecture platform evolution, core technical elements, and evaluation systems through a literature review; Section 3 clarifies the research scope and proposes a two-stage research framework; Sections 4 and 5 elaborate on the research model construction and quantitative evaluation results in detail by stages; Section 6 conducts multi-dimensional analysis, interpretation and discussion on the research results, and puts forward differentiated R&D scheme suggestions; and Section 7 summarizes the core conclusions of the research.

2. Literature Review

2.1. Evolution of Automotive EEA

Traditional distributed EEA takes independent functional electronic control units (ECUs) as the core units. With the increasing complexity of automotive functions, the peak number of ECUs per vehicle can reach more than 100, leading to problems such as computing power fragmentation, high hardware redundancy, slow software iteration, and bottlenecks in in-vehicle network bandwidth [7]. These issues make it unable to meet the core requirements of scenarios such as high-level autonomous driving and intelligent cockpits. Meanwhile, the industrial transformation of SDVs requires EEA to achieve software–hardware decoupling and full-lifecycle function upgrading, centralization has become an inevitable path for EEA evolution [8].

The centralized evolution of EEA proceeds synchronously along two dimensions: physical and logical [9]. The core of the physical dimension is the intensive integration of control units. By reducing the number of independent ECUs, the scattered computing power and control authority are concentrated in domain controllers and central computing units while optimizing the overall vehicle wiring harness layout and power consumption level. The core of the logical dimension is the hierarchical decoupling and service-oriented reconstruction of the architecture. Through standardized hierarchical software design, the decoupling of hardware drivers, basic software platforms, and application software is realized, supporting the implementation of service-oriented architecture (SOA) and enabling flexible configuration and rapid iteration of vehicle functions.

As shown in Figure 1, existing research generally divides the centralized evolution of EEA into four stages [10]: distributed modular architecture, domain-centralized architecture, cross-domain integrated architecture, and “central computing + zonal control” architecture. Among them, the fourth stage can be further divided into Multi-Box, One-Box, One-Board, and One-SoC according to the form of the central computing platform. From the perspective of industrial implementation, mainstream OEMs in the first echelon of the global automotive industry have gradually realized the mass production of Multi-Box architecture, where multiple computing printed circuit boards (PCBs) are independently packaged and powered at the central position of the vehicle and interconnected through high-speed Ethernet. A very small number of enterprises have completed the technical verification and small-scale mass production of the One-Box architecture, in which PCBs originally belonging to different boxes are integrated into a single box, optimizing the wiring harness network, reducing costs, and improving system stability through a more centralized design. The One-Board architecture will be the core research direction of the industry in the next 5 years, which integrates computing system-on-chips (SoCs) responsible for different functional domains into a single PCB and is in the frontier exploration stage. Due to hardware technology limitations, the One-SoC architecture cannot meet the functional safety requirements of automotive grade and has no mass production feasibility in the short term.

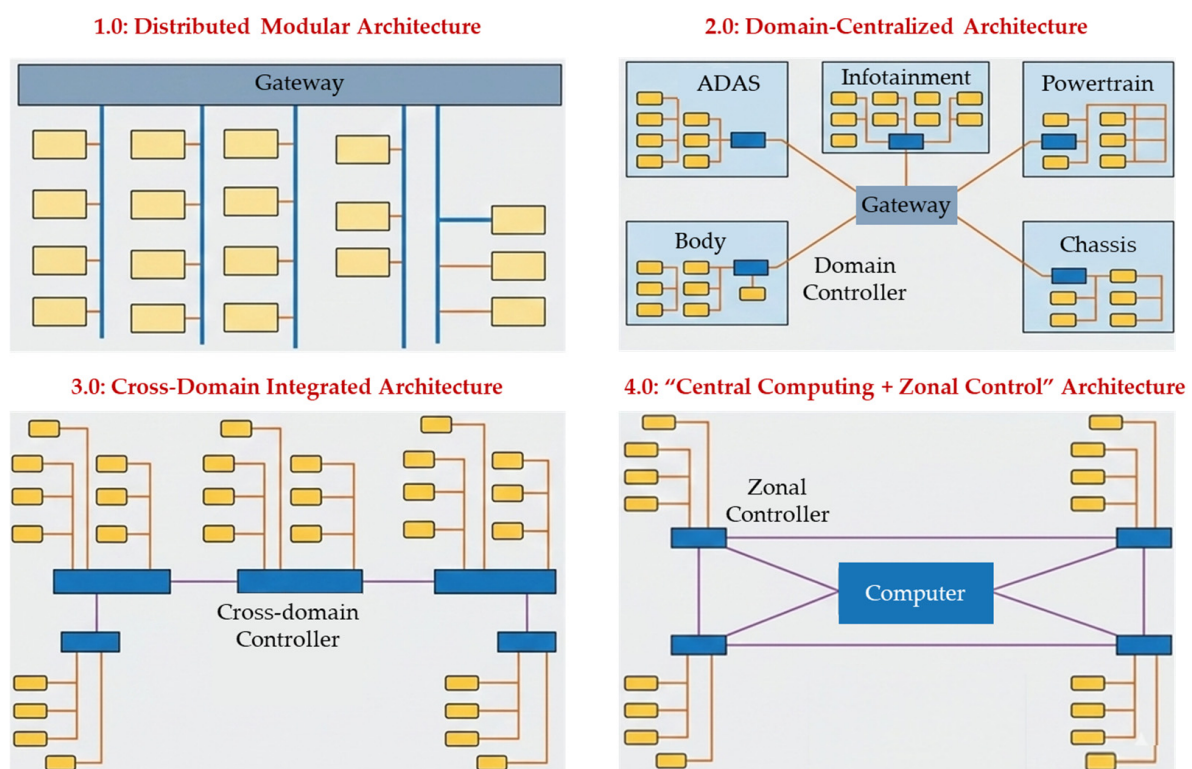


Figure 1. Four stages of EEA evolution.

2.2. Core Technical Elements Related to Centralized Architecture Platforms

The mass production and implementation of centralized architecture platforms are not merely the iteration and upgrading of a single hardware component but rather the full-stack technical collaboration between hardware architecture and basic software. Zhao et al. pointed out that the core competitiveness of centralized architecture platforms lies not only in the integration of hardware but also in the integration and interconnection of software, and software–hardware decoupling is even the core prerequisite for architectural centralization [11]. Specifically, it involves the following core software and hardware

technical elements, which collectively determine the engineering feasibility, functional safety level, and full-lifecycle iteration capability of the architecture:

- **The central computing platform** is the physical foundation and core hardware carrier of the centralized architecture platform. The core connotation of this element is to realize the computational capacity integration of functions such as autonomous driving, intelligent cockpit, body control, and power system through the in-depth integration of high-computing-power heterogeneous SoCs and cooperate with zonal controllers to complete the distributed access of peripheral interfaces and actuators, forming a minimalist hardware topology of “central computing + zonal control” [12]. Relevant studies have shown that the central computing platform can reduce the number of ECUs in the entire vehicle from more than 100 to less than 20 and shorten the length of the vehicle wiring harness by more than 30% [13].
- **The automotive operating system (OS) kernel** is the underlying software core of the centralized architecture platform and serves as the core carrier for hardware resource scheduling, task management, and security isolation [14]. Its technical connotation includes real-time kernels for high-safety scenarios (such as QNX), ecological kernels for Internet of Things scenarios (such as Android), and AI kernels for high-performance AI computing (such as Linux) [15]. In the centralized architecture platform, functions with different safety levels and real-time requirements are centrally deployed, and the real-time performance, isolation, and security of the kernel directly determine the safety boundary of vehicle functions.
- **Platform middleware** represented by Automotive Open System Architecture (AUTOSAR) is the core bridge connecting the underlying operating system and upper-layer applications. Taking AUTOSAR as an example, its technical connotation is divided into two major systems: the classic platform and the adaptive platform [16]. The classic platform is oriented to high-safety-level vehicle control functions, providing basic software modules that comply with functional safety standards to support the deployment of body, powertrain, and other domain functions in zonal controllers. The adaptive platform is oriented to high-computational-capacity intelligent scenarios, supporting POSIX standard interfaces and dynamic function deployment and providing a standardized platform for the rapid development of upper-layer intelligent applications. Through standardized hardware abstraction, communication protocol stacks, and basic software modules, deep decoupling of underlying software and hardware is achieved. Platform middleware can shorten the vehicle software development cycle by more than 40% and increase the software module reuse rate to more than 60%, significantly reducing the engineering development cost of centralized architecture platforms [17].
- **Framework middleware** oriented to SOA/agent is the core basic software for realizing cross-domain integration of centralized architecture platforms. The SOA framework encapsulates vehicle functions into standardized service interfaces and realizes low-latency cross-domain calls of vehicle functions through service registration, discovery, invocation, and dynamic orchestration. Based on the SOA framework, OEMs can quickly develop and iterate scenario-based functional applications without modifying the underlying hardware and software [18]. In the era of AIDVs, the agent framework further realizes intelligent service orchestration on the basis of the SOA framework, supporting the realization of functions in complex scenarios such as autonomous driving and vehicle–road–cloud integration through multi-agent collaboration [19].

2.3. Evaluation Indicators of Centralized Architecture Platforms

Currently, academic and industrial evaluations of centralized architecture platforms mainly focus on three core dimensions:

- Architectural advancement is the key to measuring the core technological competitiveness of centralized architectures, with core indicators including computing power concentration, software–hardware decoupling degree, hardware redundancy, functional scalability, and network topology optimization rate [20]. These indicators are the core criteria for distinguishing the generational differences of centralized architectures, determining the technological iteration potential and functional upper limit of the architecture.
- Quality and reliability indicators are the core compliance and safety bottom line for the mass production of centralized architectures, with core indicators including functional safety level, information security protection capability, hard real-time performance, and electromagnetic compatibility [21]. Centralized architectures feature highly concentrated computing power and functions, where the impact scope of single-point failure is much larger than that of distributed architectures. Networked scenarios also bring higher information security risks, making these indicators directly determine the compliance and driving safety of the architecture [22].
- Economic indicators determine the industrial promotion value of centralized architecture platforms, with core indicators including development and operation and maintenance costs, bill of materials (BOM) costs, and platform-based hardware reuse rate [23]. Although centralized architectures reduce hardware redundancy through ECU integration, the R&D investment in high-computing-power platforms and complex basic software has increased significantly. These indicators directly affect the mass production profit space of OEMs.

2.4. Impact of R&D Solutions on Centralized Architecture Platforms

The R&D strategy of OEMs directly determines the integration effect, iteration efficiency, and engineering implementation capability of architecture platforms. A centralized EEA is a full-stack collaborative system covering hardware, underlying systems, middleware, and upper-layer applications. Different combinations of R&D strategies determine OEMs' control over core technologies, the depth of software–hardware collaboration, and the efficiency of supply chain collaboration, which, in turn, directly affect the performance upper limit and commercial feasibility of architecture platforms [24].

In theory, the full-stack in-house R&D solution can realize the fully independent definition of software–hardware interfaces, communication protocols, and the overall architecture, completely eliminating the adaptation barriers among multiple suppliers and maximizing software–hardware collaborative optimization and in-depth architectural integration [25]. However, its R&D investment grows exponentially with the coverage of the technology stack, and it is highly dependent on large-scale mass production to achieve cost amortization. Restricted by technical reserves, capital strength, and sales scale, only a few global leading OEMs have the capability of large-scale implementation.

Therefore, the full-stack controllable solution has become the core R&D strategic direction for more and more OEMs. By establishing a hierarchical division of labor collaboration system with core suppliers, OEMs can, on the premise of firmly controlling key links such as architecture definition, a few core technologies, and safety systems, jointly develop and implement non-core modules with suppliers. This can not only ensure the effect of architectural integration but also effectively share R&D investment and technical risks, adapting to the R&D needs of OEMs with different scales and resource endowments [26].

Currently, academic research on OEM R&D decisions mainly focuses on non-technical influencing factors. Amiri et al. analyzed the compatibility contradiction between the decentralized organizational structure of traditional OEMs and the R&D requirements of centralized architectures, identifying key influences on decisions such as organizational structure, systems engineering capabilities, and management cognition [27]. Moreover, existing achievements are dominated by qualitative research, mostly adopting methods such as multiple case studies, in-depth industry interviews, and theoretical deduction. Only a small number of studies have attempted to conduct quantitative analysis on a single technical element, such as R&D decisions for automotive chips [28]. The core reasons for the difficulty in advancing quantitative research can be summarized into two points: First, decision variables have strong coupling and high uncertainty. R&D decisions need to consider multi-dimensional variables such as technology, organization, supply chain, market, and regulations, and many non-technical factors are difficult to standardize and quantify. Second, the availability of core data is extremely poor. Data such as input–output, project cycle, and internal management of OEMs’ centralized architecture platform R&D are mostly commercial secrets, and there is a lack of large-sample public data, which cannot meet the sample requirements for quantitative analysis. The next section will introduce in detail how the quantitative research framework designed in this study fills the above gaps.

3. Research Framework

3.1. Definition of Research Objects

Based on the literature review, this study mainly focuses on 7 core software and hardware technical elements closely related to centralized architecture platforms, namely: central computing platform (*Computer*), real-time and secure OS kernel (*RT-OS*), ecological OS kernel (*Eco-OS*), high-performance AI OS kernel (*AI-OS*), classic platform middleware (*CP-MW*), adaptive platform middleware (*AP-MW*), and framework middleware (*Framework*). Among them, the central computing platform constitutes the hardware foundation of the architecture; the three types of OS kernels are the core basis for hardware resource scheduling and the operation of upper-layer applications; and the three types of middleware are the core hubs for realizing software–hardware decoupling, service-oriented deployment, and cross-domain collaboration. These seven elements fully cover the core technical links of centralized architecture R&D without omitting any key capability modules.

From the perspective of technical hierarchy, each of the seven elements corresponds to an independent technical module, R&D division of labor system, and business cooperation model in centralized architecture R&D. Each element has an independent decision-making space for R&D strategies, which can clearly distinguish the R&D characteristics of different modules. From the perspective of decision-making practice, further decomposing the technical granularity downward (such as decomposing middleware into a single functional module) will lead to a dimensional explosion of R&D strategy combinations, exceeding the actual management boundary of OEM R&D decisions; merging the technical granularity upward will obscure the R&D differences and decision-making logic of different technical modules, making it impossible to achieve accurate optimization analysis of R&D strategy combinations. In addition, this granularity is completely consistent with the actual decision-making units of centralized architecture R&D of global OEMs, which can ensure the applicability of the research conclusions. Therefore, this research granularity can fully adapt to the analysis objectives and application scenarios of this study. This study divides four R&D strategies for each type of core technical element.

For the *Computer*, its core components include three core modules: core neural network processing unit (NPU) IP, SoC, and PCB [29]. Based on the OEMs’ R&D responsibilities,

depth of technical control, and ownership of IP for the three modules, the four R&D strategies are defined as follows:

- Full outsourcing strategy: OEMs do not participate in the R&D design, testing, and verification of any modules of the central computing platform and directly purchase standardized hardware finished products that have passed complete automotive-grade certification from suppliers.
- In-house R&D board strategy: OEMs independently complete the entire process of PCB board-level design and prototype development and adopt third-party standardized procurement solutions for SoCs.
- In-house R&D SoC strategy: OEMs independently complete the entire process of automotive-grade SoC design and automotive-grade certification, adopt third-party standardized commercial authorization solutions for core NPU IP, and simultaneously independently complete the entire process of PCB board-level R&D.
- In-house R&D core IP strategy: OEMs independently complete the entire R&D process of core NPU IP cores and complete the entire SoC design process and PCB board-level design based on the in-house-developed IP cores.

For the six basic software elements, including *RT-OS*, *Eco-OS*, *AI-OS*, *CP-MW*, *AP-MW*, and *Framework*, the four R&D strategies are uniformly defined based on the OEMs' development responsibilities for software code, degree of customization, degree of utilization of open-source ecosystems, ownership of IP, and commercial payment models, combined with the general business models in the automotive software industry:

- Outsourcing strategy: OEMs purchase complete standardized commercial software products from suppliers at the industry-unified average price and pay usage fees through a per-vehicle license model.
- Customization strategy: Based on the standardized benchmark software products provided by suppliers, OEMs put forward customized development requirements for some functional modules and adaptation needs, and the suppliers complete the customized code development, testing, verification, and on-vehicle adaptation; OEMs pay commissioned development fees and per-vehicle license fees for the basic products according to the customization ratio.
- Cooperation strategy: OEMs and suppliers jointly carry out software R&D work and make full use of open-source software components to build the basic framework, and the remaining core functional modules are developed, tested, and verified by OEMs and suppliers in accordance with the agreed proportion of division of labor; OEMs only pay the corresponding proportion of per-vehicle license fees to suppliers according to their cooperative contributions.
- In-house R&D strategy: OEMs fully dominate the entire process of software R&D and iterative optimization, build the basic framework based on open-source software components, and independently complete the R&D of all remaining functional modules; OEMs independently bear all R&D investment and full-lifecycle maintenance costs and do not need to pay any license fees to suppliers.

As shown in Figure 2, the core research object of this study is the combination of R&D strategies for the above seven technical elements, totaling 4 to the 7th power (16,384) combinations of solutions.

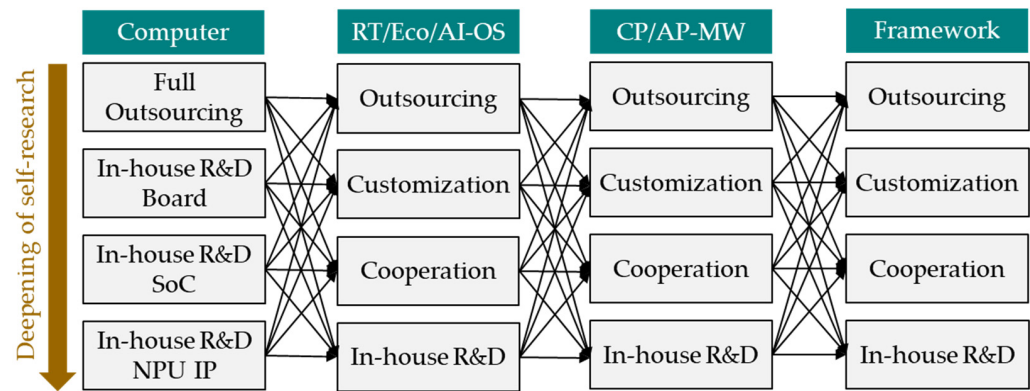


Figure 2. Combination of R&D strategies for multi-technical elements.

3.2. Selection of Evaluation Indicators

Combining industrial practice and the academic literature, this study selects architectural advancement and total cost of ownership (TCO) as core decision-making indicators and takes quality as a constraint condition, which are specifically explained as follows:

- **Architectural advancement:** This study does not focus on specific performance indicators but uses the degree of architectural centralization described in Section 2.1 to characterize architectural advancement, which is divided into seven levels (Distributed, Domain, Cross-Domain, Multi-Box, One-Box, One-Board, and One-SoC). This grading is determined by three core dimensions: (1) hardware integration and computing power centralization; (2) in-vehicle network topology optimization; and (3) software–hardware decoupling and SOA enablement. Each level has clear technical boundaries, corresponding to distinct mass-production stages and advancement thresholds. This system fully covers the entire evolution path from traditional distribution to ultimate centralization, provides a quantifiable benchmark for architectural advancement evaluation in this study, and avoids excessive granularity that would cause decision-making dimensionality explosion.
- **TCO:** It covers the one-time R&D investment, the variable cost per mass-produced vehicle, and the lifecycle maintenance cost. The main purpose of TCO in this study is to reflect the cost that OEMs need to pay to implement the target architecture platform. Therefore, the TCO here does not include the potential economic benefits brought by the centralized architecture platform, such as the reduction of vehicle material costs and software re-development costs. In fact, these potential economic benefits are strongly associated with architectural advancement.
- **Quality:** As a constraint indicator, it covers requirements such as functional safety, information security, automotive-grade certification, and environmental reliability. This study incorporates quality into the parameter calibration of the TCO model, where higher quality requirements are reflected in higher costs.

3.3. Scenario Setting

Among the seven levels of architectural advancement described in Section 3.2, architectural forms such as Distributed, Domain, and Cross-Domain have formed relatively mature industrial systems and R&D models and are not the core R&D direction for current OEMs' architecture upgrading.

The Multi-Box form is the mainstream mass production solution for OEMs in the first echelon of the industry. Compared with the Multi-Box form, the One-Box form is primarily an upgrade in physical hardware integration technology, which integrates multiple independent PCBs into a single enclosure while retaining the independent functional domain computing chips and software architecture design logic of the Multi-Box scheme. From

the perspective of core R&D decision making, the core technical elements, R&D strategy division, and the causal mechanism between R&D strategy combinations and architectural advancement of the One-Box scenario are highly consistent with the Multi-Box scenario. Separate analysis of the One-Box scenario will not produce new core conclusions on the necessary technical elements and scale-dependent optimal R&D strategies but will lead to research redundancy and repeated verification of existing causal relationships. Meanwhile, the One-Box scenario does not raise new full-stack collaborative requirements for underlying hardware and basic software beyond the Multi-Box architecture, which means it cannot provide forward-looking decision-making reference for the next-generation high-level centralized architecture evolution, one of the core research objectives of this study.

For the One-SoC architecture, it integrates all functional domain computing capabilities into a single system-on-chip, which is the ultimate form of physical centralization of automotive EEA in theory. However, from the perspective of industrial practice and technical feasibility, there is currently no automotive-grade SoC that can meet the full-scenario functional safety requirements and pass complete automotive-grade mass production certification worldwide. There are no effective mass production cases and complete project cycle data in the industry to support the calibration of the fsQCA model and TCO evaluation system, and the inclusion of this scenario will lead to a significant reduction in the reliability and practical guidance of the research conclusions. In addition, the One-SoC architecture has no clear mass production feasibility in 5 years, which is inconsistent with the core positioning of this study to provide actionable R&D decision-making reference for OEMs' current and medium-term architecture platform development.

Therefore, Multi-Box and One-Board are the only typical scenarios that simultaneously align with the research objectives, industrial reality, and practical guidance value of this study. The Multi-Box scenario represents the current mainstream mass production solution of the industry, while the One-Board scenario covers the forward-looking technical direction of the industry in the next 5 years, which can fully reveal the iteration law of R&D strategies in the process of architectural centralization.

3.4. Two-Stage Research Framework

As shown in Figure 3, this study constructs a two-stage research framework of “feasible solution screening under architectural advancement constraints–TCO quantitative evaluation” to realize dual verification of the technical feasibility and commercial rationality of R&D strategy combinations.

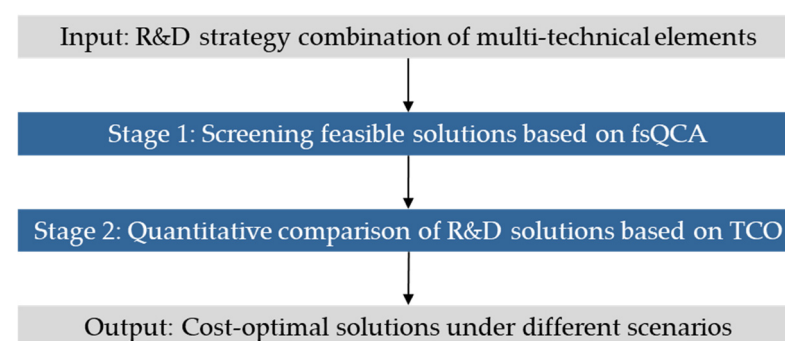


Figure 3. Two-stage research framework.

The first stage is the screening stage of feasible R&D strategy combinations, which adopts the fuzzy-set qualitative comparative analysis (fsQCA) method for analysis. Based on set theory and Boolean algebra, fsQCA integrates the idea of fuzzy mathematics, which can accurately identify the concurrent causal relationship between combinations of multiple conditional variables and outcome variables, breaking through the analysis limitation of

traditional regression methods on the independent effects of variables [30]. The core of this study is the combined synergistic effect of R&D strategies for the 7 technical elements, rather than the independent impact of a single element, so fsQCA is highly compatible with this research problem. In this stage, architectural advancement is taken as the outcome variable, and the R&D strategies of the 7 core technical elements are taken as conditional variables. Configuration analysis is carried out for the Multi-Box and One-Board scenarios, respectively, and the strategy combinations that do not meet the requirements of architectural advancement are eliminated to complete the screening of feasible solutions.

The second stage is the cost quantitative evaluation stage of feasible strategy combinations. Based on the TCO evaluation model, the cost of each element under different R&D strategies is quantitatively calculated. The model finally outputs the total average cost per vehicle. This study sets 4 production scale gradients, corresponding to local small-scale OEMs with an annual output of 200,000 vehicles, local medium-scale OEMs with an annual output of 500,000 vehicles, local large-scale OEMs with an annual output of 1,000,000 vehicles, and international ultra-large-scale OEMs with an annual output of 2,000,000 vehicles, fully covering the operating scenarios of OEMs of different sizes in the Chinese passenger car market.

Through this two-stage framework, the cost-optimal R&D strategy combinations for OEMs under different architectural scenarios and different production scales can be systematically output, providing quantitative references for OEM R&D decisions. This design ensures that all strategies involved in the cost comparison can achieve the corresponding potential benefits of the target architecture level, eliminating the interference of benefit differences on cost comparative analysis and ensuring that the screening of cost-optimal strategies is conducted under a unified benefit benchmark.

4. Screening of Feasible R&D Solutions Based on fsQCA

This section adopts fsQCA to screen feasible R&D strategy combinations that meet the requirements of architectural advancement. The standardized implementation process of fsQCA in this study strictly follows the classic analytical framework and mainstream application norms in the engineering management field, which is divided into four core steps:

1. Calibration of variables: converting the original data of conditional variables and outcome variables into fuzzy-set membership scores between 0 and 1, which is the core basis of fsQCA analysis;
2. Necessary condition test: verifying whether a single conditional variable constitutes a necessary condition for the outcome of high architectural advancement, with a consistency threshold of 0.9 as the judgment standard;
3. Truth table construction: constructing a truth table based on calibrated fuzzy-set data, setting a case frequency threshold and a raw consistency threshold of 0.8 to filter valid configurations;
4. Configuration analysis and Boolean minimization: performing standard Boolean minimization operations on the filtered truth table through the Quine–McCluskey algorithm and identifying the core necessary configurations that can achieve high architectural advancement.

The detailed calibration rules of fuzzy sets and implementation details of each step are elaborated in the subsequent content.

4.1. Sample Database Construction and Data Calibration

This study collected 143 valid cases of vehicle architecture platform solutions from the Chinese passenger car market between 2018 and 2025, with data obtained from four

complementary channels, and completed multi-source cross-validation for all cases. The detailed composition of data sources is as follows:

- Publicly available industry information (accounting for 62% of the core parameter dimensions of all cases): including official technical disclosure documents of OEMs and tier 1 suppliers, technical papers, authoritative media reports, and third-party technology evaluation reports.
- Internal enterprise research, expert consultation, and project practice accumulation (accounting for 38% of the core parameter dimensions of all cases): including the R&D strategy division of core technical elements, non-public technical parameters of software and hardware modules, and engineering implementation details of architecture platforms, which are obtained through in-depth interviews with senior technical experts from mainstream OEMs and core suppliers in the automotive industry, as well as the first-hand project data accumulated by the research team for a long time. For the sensitive data involving enterprise trade secrets in this channel, we have completed desensitization processing, only retaining the classification information of R&D strategies and technical level required for fsQCA calibration, without involving specific enterprise confidential data such as detailed R&D investment and internal technical parameters.

To ensure the validity and representativeness of the sample database, we formulated strict and clear sample exclusion criteria before data collection: (1) concept car or prototype vehicle architecture solutions that have not entered the mass production verification stage, to avoid the deviation of theoretical design from actual engineering implementation; (2) cases where the core technical parameters of more than 30% of the 7 core elements are missing and cannot be supplemented and verified through multi-source channels; and (3) architecture solutions that only upgrade a single functional domain without iterative optimization of the whole vehicle EEA, which cannot reflect the overall advancement of the centralized architecture platform.

This case database includes mainstream OEMs in the Chinese market, including traditional OEMs, new vehicle startups, and Sino-foreign joint venture brands. The samples have sufficient industry representativeness and can fully reflect the complete evolution process of centralized automotive architecture platforms from 2018 to 2025. Two researchers with more than 5 years of research experience in the field of automotive EEA and software-defined vehicles independently completed the labeling and initial coding of all cases in a double-blind manner. For all cases, we ensured that the coding results of each variable were supported by at least two independent data sources to eliminate subjective bias in the labeling process.

For architectural advancement, the outcome variable of this study, its fuzzy-set membership score is comprehensively determined by two core dimensions: physical centralization and logical centralization of the architecture, which is consistent with the dual-dimensional evolution law of automotive EEA proposed in Section 2.1. Among them, physical centralization is characterized by the integration level of hardware control units (i.e., the reduction of independent ECUs and the intensive integration of computing units), and logical centralization is characterized by the hierarchical decoupling degree of software and the service-oriented reconstruction level of the architecture. The two dimensions are weighted equally in the scoring system, and the final comprehensive score is normalized to the range of 0 to 1. Its quantitative coding rules are formulated based on the mainstream EEA evolution stage classification in existing research and industry practice [31], and the corresponding relationship between the comprehensive score interval and the architecture level is as follows: Distributed (0–0.1), Domain (0.1–0.3), Cross-Domain (0.3–0.6), Multi-Box

(0.6–0.7), One-Box (0.7–0.8), One-Board (0.8–0.9), and One-SoC (0.9–1.0). A higher value indicates a deeper degree of architectural centralization and higher architectural advancement.

For the calibration of the outcome variable's fuzzy-set membership, we set three critical anchors in accordance with the standard fsQCA calibration norms: full membership anchor, crossover point, and full non-membership anchor. For the Multi-Box scenario, the research objective is to identify configurations that can achieve the Multi-Box and higher-level architecture advancement, so the full membership anchor is set to 0.6 (the minimum score for the Multi-Box architecture level); the crossover point (maximum fuzzy point) is set to 0.3, which is the critical threshold between the domain-centralized architecture and cross-domain integrated architecture, representing the minimum standard for entering the centralized architecture evolution stage; and the full non-membership anchor is set to 0, corresponding to the distributed architecture with no centralized characteristics. For the One-Board scenario, which focuses on high-level centralized architecture, the full membership anchor is set to 0.8 (the minimum score for the One-Board architecture level); the crossover point is set to 0.4, which is the critical threshold between the cross-domain integrated architecture and Multi-Box architecture, representing the minimum standard for entering the high-level centralized architecture stage; and the full non-membership anchor is set to 0.

For the conditional variables, i.e., the R&D strategies of 7 core technical elements, we adopt a standard four-value fuzzy set for coding and calibration, which is a widely used calibration scheme in fsQCA research for categorical variables with clear hierarchical characteristics. This scheme is highly compatible with the four-level R&D strategy division defined in Section 3.1, which can clearly distinguish the degree of OEMs' in-house R&D and technical control for each technical element and avoid the information loss caused by binary calibration. The corresponding relationship is as follows: (1) full non-membership (0): corresponds to the outsourcing/full outsourcing strategy, where OEMs have no R&D participation and technical control over the element and fully rely on third-party suppliers; (2) sub-crossover point membership (0.33): corresponds to the customization/in-house R&D board strategy, where OEMs only participate in the functional customization or board-level design of the element and the core technology is still controlled by suppliers, with a low degree of independent R&D; (3) supra-crossover point membership (0.67): corresponds to the cooperation/in-house R&D SoC strategy, where OEMs and suppliers jointly carry out R&D, master part of the core technology and IP, and have a medium-to-high degree of independent R&D, crossing the fuzzy crossover point of "whether to master the core R&D capability"; and (4) full membership: corresponds to the in-house R&D/in-house R&D core IP strategy, where OEMs fully dominate the whole R&D process, own all core IP, and have complete independent R&D capability for the element. A higher membership score indicates a deeper degree of independent research and development of OEMs for the corresponding technical element.

4.2. fsQCA for Multi-Box Scenario

First, a necessary condition test was conducted on the 7 conditional variables to determine whether a single condition is a necessary condition for achieving high architectural advancement. It is generally considered that a conditional variable with a consistency greater than 0.9 is a necessary condition for the outcome variable, and the test results are shown in Table 1.

As shown in Table 1, AP-MW (consistency = 0.911) and Framework (consistency = 0.913) are necessary conditions for achieving high architectural advancement in the Multi-Box scenario, while the remaining conditional variables do not meet the requirements of necessary conditions.

Table 1. Necessary condition test for Multi-Box scenario.

Conditional Variable	Consistency	Coverage
Computer	0.750	0.529
RT-OS	0.727	0.471
Eco-OS	0.724	0.494
AI-OS	0.774	0.565
CP-MW	0.700	0.412
AP-MW	0.911	0.965
Framework	0.913	0.988

On the basis of necessary condition test, this study constructed a truth table containing all possible configuration combinations of 7 conditional variables and set strict filtering thresholds in accordance with mainstream fsQCA application norms: the case frequency threshold was set to 1 (to retain all valid case configurations in the sample database), and the raw consistency threshold was set to 0.8 (to filter configurations with sufficient explanatory power for the outcome variable). The filtered truth table is shown in Table 1 in the Appendix A. Subsequently, this study performed standard Boolean minimization operations on the filtered truth table through the Quine–McCluskey algorithm and adopted the intermediate solution as the core result of configuration analysis (the intermediate solution can effectively balance the complexity of the solution and the interpretability of the results, which is the most widely used solution type in fsQCA research). The core necessary configurations for achieving high architectural advancement in the Multi-Box scenario are shown in Table 2.

Table 2. Configuration analysis for Multi-Box scenario.

Conditional Variable	Core Necessary Configuration
Computer	⊗
RT-OS	⊗
Eco-OS	⊗
AI-OS	⊗
CP-MW	⊗
AP-MW	⊗
Framework	⊗
Configuration Consistency	0.915
Configuration Coverage	0.962

It can be seen from the configuration analysis results that in the Multi-Box scenario, high architectural advancement can be achieved only by satisfying the dual-core independent research and development of AP-MW and Framework, without blindly pursuing full-stack independent research and development of other elements. The core reason lies in: the core pain points of the Multi-Box architecture are cross-domain collaboration, software–hardware decoupling, and interface standardization under the hardware isolation of multiple computing platforms. AP-MW and Framework are the core carriers to solve the above pain points and are the rigid demand foundation of the multi-domain centralized architecture. Their impact on architectural advancement is far greater than that of underlying hardware and OS kernel layer elements.

4.3. fsQCA for One-Board Scenario

The results of the necessary condition test and configuration analysis for the One-Board scenario are shown in Tables 3 and 4, respectively.

Table 3. Necessary condition test for One-Board scenario.

Conditional Variable	Consistency	Coverage
Computer	0.500	0.714
RT-OS	0.455	0.595
Eco-OS	0.483	0.667
AI-OS	0.500	0.738
CP-MW	0.400	0.476
AP-MW	0.911	0.976
Framework	0.902	0.976

Table 4. Configuration analysis for One-Board scenario.

Conditional Variable	Core Necessary Configuration
Computer	☑
RT-OS	☒
Eco-OS	☒
AI-OS	☑
CP-MW	☒
AP-MW	☑
Framework	☑
Configuration Consistency	0.928
Configuration Coverage	0.881

As shown in Table 3, AP-MW and Framework remain the necessary conditions for achieving high architectural advancement in the One-Board scenario, while the remaining conditional variables do not meet the requirements of necessary conditions. Different from the Multi-Box scenario, the configuration analysis results show that in addition to the core necessary conditions of AP-MW and Framework, it is also necessary to simultaneously satisfy the in-depth independent research and development of Computer and AI-OS to achieve high architectural advancement. The core reason is that the One-Board architecture has exponentially increased requirements for centralized computing power scheduling, software–hardware integration, and ecological compatibility. AP-MW and Framework are still the core hubs connecting the underlying hardware and upper-layer applications, but middleware alone can no longer meet the advancement requirements of high-level architectures. It is necessary to realize the full-chain collaborative drive of “hardware foundation + AI OS kernel + key middleware”, resulting in a significant increase in the R&D threshold.

5. TCO Evaluation of Core Technical Elements Under Different R&D Strategies

5.1. TCO of Central Computing Platform Under Different R&D Strategies

The TCO of the central computing platform can be decomposed into one-time R&D costs $C_{\text{computer_R\&D}}$, outsourced engineering fees $C_{\text{computer_out}}$, and variable costs per mass-produced vehicle $C_{\text{computer_unit}}$. As shown in Figure 4, $C_{\text{computer_R\&D}}$ can be further decomposed into three sub-items: C_{NPU} , C_{SoC} , and C_{PCB} . Different R&D strategies correspond to different cost sub-items covered by the one-time R&D costs. This study selects a 5 nm automotive-grade computing platform as a case, and the calculation details of the cost sub-items of its one-time R&D costs are shown in Table 5.

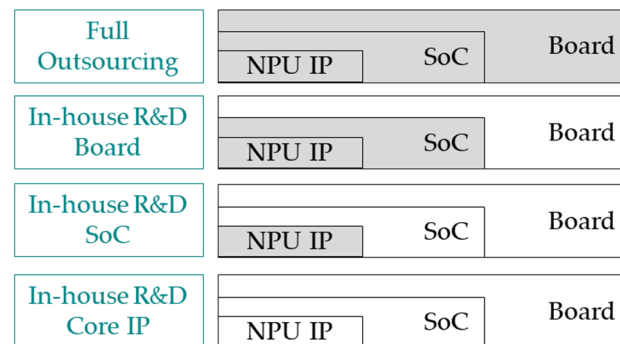


Figure 4. Different R&D strategies of central computing platform.

Table 5. R&D cost structure of central computing platform.

Cost Sub-Item	Subdivision Subject	Calculation Details	Amount
C_{NPU}	Labor Cost	- Annual per capita salary: ¥800,000 - Team size: 120 people - R&D cycle: 3 years	¥288,000,000
	EDA Licensing Fee	- Annual license fee: ¥1,200,000/seat - Number of concurrent seats: 60 - R&D cycle: 3 years	¥216,000,000
	Test Verification Cost	- Prototype verification platform: ¥14,000,000 - Simulation computation capacity: ¥12,000,000 - Functional safety verification: ¥8,000,000	¥34,000,000
	Other Supporting Cost	Including patent applications, pre-verification, etc., accrued at 8% of the total amount of the first three subjects	¥43,000,000
C_{SoC}	IP Licensing Fee	- Basic IP package: ¥120,000,000 - Automotive-grade NPU IP: ¥80,000,000	¥120,000,000 ¥200,000,000
	Labor Cost	- Annual per capita salary: ¥800,000 - Team size: 280 people - R&D cycle: 2 years	¥448,000,000
	EDA Licensing Fee	- Annual license fee: ¥1,500,000/seat - Number of concurrent seats: 120 - R&D cycle: 2 years	¥360,000,000
	Test Verification Cost	- Prototype verification platform: ¥32,000,000 - Simulation computation capacity: ¥50,000,000 - Full pre-silicon verification: ¥48,000,000	¥130,000,000
	Tape-out Cost	5 nm automotive-grade process: ¥340,000,000 for a single tape-out 2 regular tape-outs	¥680,000,000
	Automotive-grade Certification Cost	- AEC-Q100: ¥12,000,000 - ASIL-D certification: ¥28,000,000 - Environmental reliability test: ¥10,000,000	¥50,000,000
C_{PCB}	SoC Licensing Fee	Automotive-grade SoC: ¥18,000,000	¥18,000,000
	Labor Cost	- Annual per capita salary: ¥500,000 - Team size: 30 people - R&D cycle: 2 years	¥30,000,000

Table 5. Cont.

Cost Sub-Item	Subdivision Subject	Calculation Details	Amount
C_{PCB}	Design Tool Licensing Fee	- Annual license fee: ¥3,500,000 - R&D cycle: 2 years	¥7,000,000
	Test Verification Cost	- Signal integrity test: ¥8,000,000 - Environmental reliability test: ¥5,000,000 - Electromagnetic compatibility test: ¥7,000,000	¥20,000,000
	Cost of Prototyping and Pilot Production	Including prototype fabrication, low-volume pilot production, material loss, etc., with a total cost of ¥10,000,000	¥10,000,000

Meanwhile, OEMs need to pay a small amount of special outsourced engineering fees outside the scope of independent R&D. The in-house R&D core IP strategy requires services such as special process simulation and third-party testing, which are accrued at 2% of the total $C_{computer_R\&D}$. The in-house R&D SoC strategy is accrued at 3% of the total $C_{computer_R\&D}$. The in-house R&D board strategy requires services such as underlying drivers and board support package adaptation for third-party mature SoCs, with a fixed industry quotation benchmark of 120 million RMB. The full outsourcing strategy requires services such as customized engineering debugging and function cropping for mature mass-produced central computing platforms, with a fixed industry quotation benchmark of 35 million RMB.

In addition, different R&D strategies also affect the variable cost per mass-produced vehicle of the central computing platform. The full outsourcing strategy requires full assumption of R&D amortization, IP royalties, and profits from upstream manufacturers, resulting in the highest variable cost (2500 RMB per vehicle). The in-house R&D board strategy can save part of the PCB integration premium, with a variable cost of 2400 RMB per vehicle. The in-house R&D SoC strategy can further save the huge premium of chip manufacturers, with a variable cost of 2000 RMB per vehicle. The in-house R&D core IP strategy does not require NPU IP royalty expenditure, with a variable cost of 1920 RMB per vehicle.

In summary, Table 6 shows the complete TCO composition of the central computing platform under different R&D strategies.

Table 6. TCO composition of the central computing platform under different R&D strategies.

R&D Strategy	C_{NPU}	C_{SoC}	C_{PCB}	$C_{computer_out}$	$C_{computer_unit}$
In-house R&D Core IP	¥581,000,000	¥1,788,000,000	¥67,000,000	¥48,720,000	¥2500
In-house R&D SoC	¥0	¥1,868,000,000	¥67,000,000	¥58,050,000	¥2400
In-house R&D Board	¥0	¥0	¥85,000,000	¥120,000,000	¥2000
Full Outsourcing	¥0	¥0	¥0	¥35,000,000	¥1920

5.2. TCO of Automotive Basic Software Under Different R&D Strategies

The TCO of automotive basic software such as OS kernels and middleware can be decomposed into one-time R&D costs $C_{SW_R\&D}$, per-vehicle third-party software license fees C_{SW_unit} , and full-lifecycle maintenance costs $C_{SW_maintain}$. Based on the definition of the four R&D strategies in Section 3.1, this study constructs an automotive basic software TCO evaluation model, as shown in Figure 5.

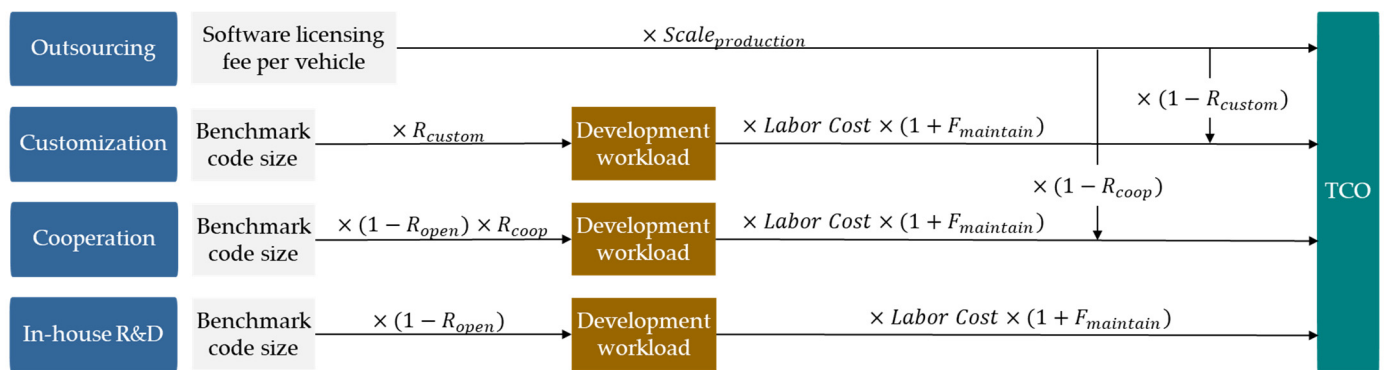


Figure 5. Automotive basic software TCO evaluation model.

R_{custom} is the ratio of customized software development; R_{open} is the ratio of open-source software components; and R_{coop} is the ratio of workload undertaken by OEMs in cooperative development. This study selects the products with the highest market share/most representativeness of various types of software as benchmark products and conducts extensive research on industry practice to obtain the data shown in Table 7.

Table 7. Benchmark parameters of automotive basic software R&D.

Software Element	Benchmark Product	C_{SW_unit}	R_{custom}	R_{open}	R_{coop}
RT-OS	QNX	¥200	10%	60%	30%
Eco-OS	Android	¥2000	20%	70%	30%
AI-OS	Linux	¥1000	30%	80%	30%
CP-MW	AUTOSAR CP	¥500	10%	30%	30%
AP-MW	AUTOSAR AP	¥1000	30%	50%	50%
Framework	NeuSAR/ROS 2	¥2000	60%	60%	60%

The model follows the following basic assumptions:

- OEMs pay fees to software suppliers based on per-vehicle license fees and production scale: This assumption is consistent with the mainstream business model in the current Chinese automotive industry.
- Third-party software license prices are stable: Due to the low marginal cost of software, supplier pricing is prone to falling into a vicious price-cutting cycle under extreme market conditions [32]. This study considers a benign market and assumes that software suppliers maintain similar and stable pricing for the same type of software.
- For the same type of software, the code size remains consistent under different R&D strategies: Automotive basic software is all large-scale software, and its code size can effectively reflect its functional completeness. This assumption is equivalent to controlling the variable of software functional completeness, thereby better analyzing the cost differences under different R&D strategies.

The core calculation process of this model lies in how to convert software code size into actual development costs. The marginal development cost of automotive basic software increases exponentially, and there are significant differences in the R&D characteristics of different software types and the development efficiency of different teams [33]. The constructive cost model (COCOMO) is an empirical software cost estimation model derived from a large number of software engineering project data [34]. Its basic function form highly matches the exponential growth law of automotive basic software development, and it provides dozens of adjustable parameter factors to adapt to the forward development mode of automotive basic software. Based on COCOMO II, this study constructs a software

development cost evaluation model adapted to automotive scenarios, specifically as shown in Equations (1) and (2):

$$DW = A \times Size^E \times \prod_{i=1}^{17} EM_i \quad (1)$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j \quad (2)$$

DW is the software development effort in person-months; $Size$ is the software code size in kilo lines of code; E is the scale exponent; EM_i is the effort multiplier, covering 17 parameters in four categories: product factors, personnel factors, platform factors, and project factors; and SF_j is the exponent scaling factor, covering 5 items: precedent (PREC), development flexibility (FLEX), risk resolution (RESL), team cohesion (TEAM), and process maturity (PMAT). A and B are core calibration constants of the COCOMO II model, with detailed definitions and calibration rules as follows:

- A is the productivity coefficient constant, which determines the linear baseline of software development workload and reflects the overall development efficiency of the engineering team. The value is calibrated according to software type: 3.6 for embedded software and 3.0 for semi-organic and semi-embedded software.
- B is the scale exponent baseline constant, which affects the nonlinear growth characteristics of development workload with code size and reflects the economies/diseconomies of scale in software development. The value is also calibrated based on software type: 1.2 for embedded software and 1.12 for semi-organic and semi-embedded software, to match the nonlinear growth law of R&D difficulty of different software types.

Different types of software have different R&D characteristic factors (for example, RT-OS has higher safety requirements). Different R&D strategies also lead to differences in R&D characteristic factors (for example, suppliers have more rich development experience than OEMs). Therefore, for different types of software and different R&D strategies, this study comprehensively calibrates all parameter factors combined with technical characteristics and industry practical experience (see Table 3 for details). All parameter calibrations strictly follow the classic framework of the COCOMO II model and the industry specifications of automotive-grade software engineering. All parameter values are within the general benchmark range of published research in the field of automotive software engineering, which avoids subjective random setting of parameters and ensures the universality and robustness of the benchmark model in industry general scenarios. Through the multi-dimensional parameter calibration, we have strictly controlled the consistency of the functional completeness, safety level, and reliability of software products under different R&D strategies, ensuring that the cost comparison is carried out under a unified quality and performance benchmark.

The above analysis solves the cost allocation problem between OEMs and suppliers. Automotive basic software also requires continuous maintenance during its lifecycle. In current business practice, per-vehicle license fees usually include later maintenance fees. However, for software in-house or customized by OEMs, OEMs need to bear additional maintenance costs, which have a certain proportional relationship with R&D costs. $F_{maintain}$ in Figure 5 is the maintenance proportional factor. The deeper the degree of in-house R&D, the easier it is for OEMs to add, delete, modify, and query functional modules [35], and the corresponding $F_{maintain}$ is usually lower (this study sets the in-house R&D strategy to 8%, cooperation strategy to 15%, customization strategy to 25%, and outsourcing strategy to 0 per year). This set of $F_{maintain}$ is the industry average benchmark value determined based on in-depth interviews and project data research with 3 mainstream OEMs and 2 leading

automotive basic software suppliers. In summary, the software TCO results shown in Table 8 can be obtained.

Table 8. R&D cost structure of automotive basic software.

Software Element	R&D Strategy	$C_{SW_R\&D}$	C_{SW_unit}	$C_{SW_maintain}$
RT-OS	In-house R&D	¥2,897,000,000	¥0	¥231,760,000
	Cooperation	¥436,000,000	¥140	¥65,400,000
	Customization	¥100,000,000	¥180	¥25,000,000
	Outsourcing	¥0	¥200	¥0
Eco-OS	In-house R&D	¥24,424,000,000	¥0	¥1,953,920,000
	Cooperation	¥3,233,000,000	¥1400	¥484,950,000
	Customization	¥1,726,000,000	¥1600	¥431,500,000
	Outsourcing	¥0	¥2000	¥0
AI-OS	In-house R&D	¥13,436,000,000	¥0	¥1,074,880,000
	Cooperation	¥1,597,000,000	¥700	¥239,550,000
	Customization	¥2,534,000,000	¥700	¥633,500,000
	Outsourcing	¥0	¥1000	¥0
CP-MW	In-house R&D	¥1,671,000,000	¥0	¥133,680,000
	Cooperation	¥296,000,000	¥350	¥44,400,000
	Customization	¥36,000,000	¥450	¥9,000,000
	Outsourcing	¥0	¥500	¥0
AP-MW	In-house R&D	¥1,917,000,000	¥0	¥153,360,000
	Cooperation	¥560,000,000	¥500	¥84,000,000
	Customization	¥156,000,000	¥700	¥39,000,000
	Outsourcing	¥0	¥1000	¥0
Framework	In-house R&D	¥4,010,000,000	¥0	¥320,800,000
	Cooperation	¥1,261,000,000	¥800	¥189,150,000
	Customization	¥938,000,000	¥800	¥234,500,000
	Outsourcing	¥0	¥2000	¥0

5.3. Vehicle Amortization Cost Under Different Mass Production Scales

Since the TCO includes both one-time costs and costs that increase with the expansion of vehicle production scale, to facilitate comparison on the same scale, this study converts all costs into the average cost per vehicle.

The formulas for calculating the vehicle amortization cost of the central computing platform and automotive basic software are shown in Equations (3) and (4), respectively:

$$C_{computer_vehicle} = \frac{C_{computer_R\&D} + C_{computer_out}}{T \times Scale_{production}} + C_{computer_unit} \quad (3)$$

$$C_{SW_vehicle} = \frac{C_{SW_R\&D} + C_{SW_maintain} \times T}{T \times Scale_{production}} + C_{SW_unit} \quad (4)$$

$Scale_{production}$ is the annual vehicle production scale of the architecture platform, and four production scales have been introduced in Section 3.4. T is the lifecycle of the architecture platform, which is set to 6 years. On the one hand, it is because the breakthrough of hardware technology has slowed down, and the iteration of the architecture platform can no longer achieve evolution every 2–3 years as in the past few years; on the other hand, it is because the warranty period provided by mainstream OEMs for products is generally no more than 6 years.

The evaluation results of the vehicle amortization cost under different production scales are shown in Figure 6, where (a–g) correspond to Computer, RT-OS, Eco-OS, AI-

OS, CP-MW, AP-MW, and Framework, respectively, with the ordinate unit being RMB per vehicle.

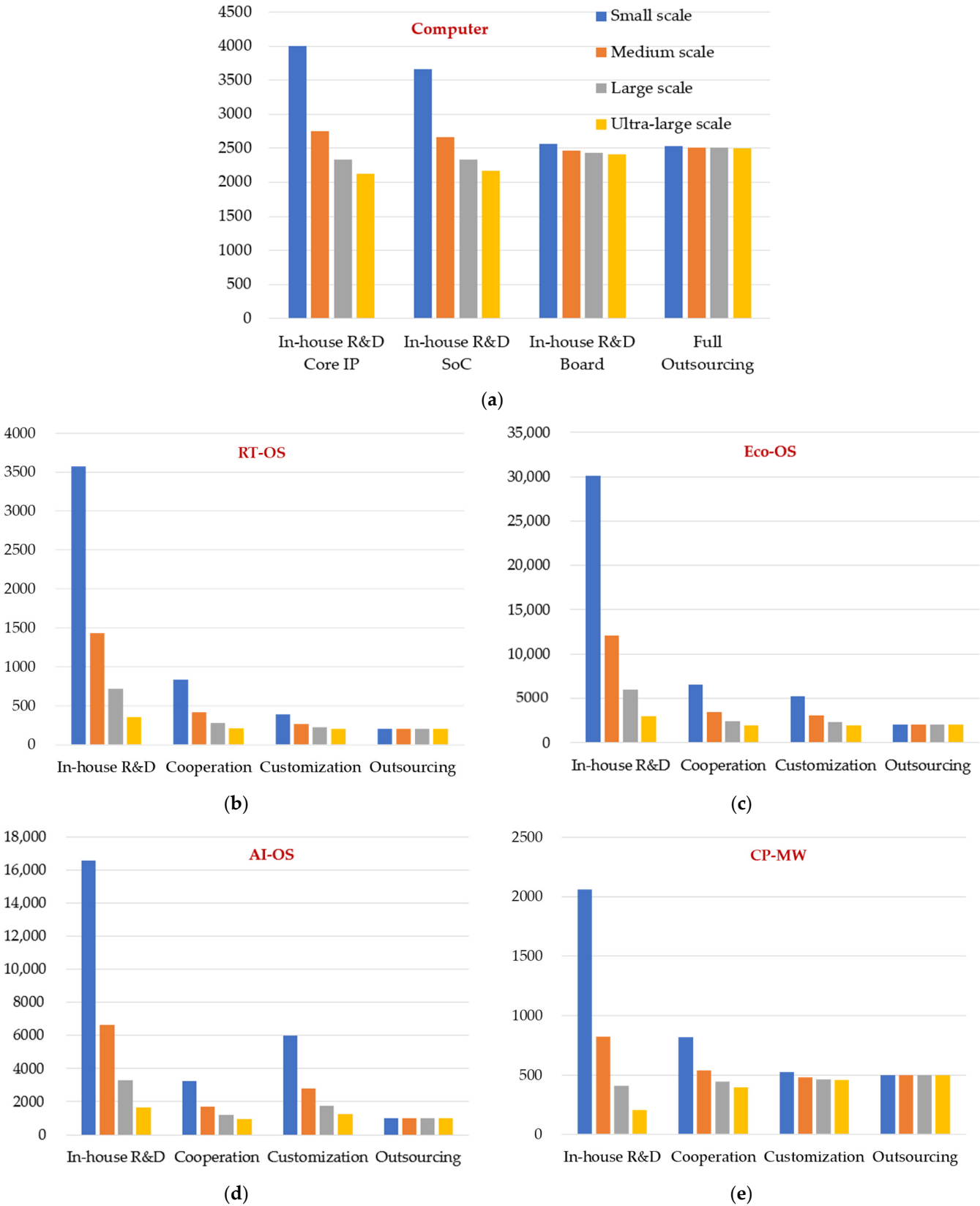


Figure 6. Cont.

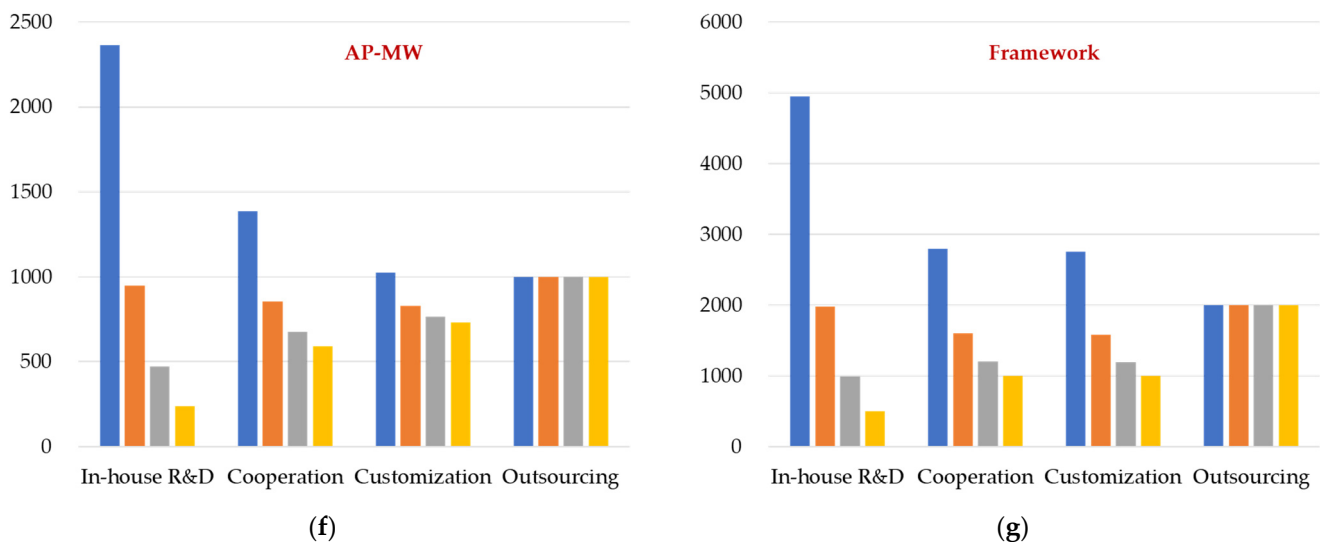


Figure 6. (a) Vehicle amortization cost of Computer; (b) vehicle amortization cost of RTOS; (c) vehicle amortization cost of Eco-OS; (d) vehicle amortization cost of AI-OS; (e) vehicle amortization cost of CP-MW; (f) vehicle amortization cost of AP-MW; (g) vehicle amortization cost of Framework.

6. Results and Discussion

6.1. Scenario-Based TCO Comparison of R&D Solutions

Based on the feasible solution constraints screened in Section 4, this study calculates the total vehicle amortization cost for the feasible R&D strategy combinations under the Multi-Box and One-Board scenarios, respectively.

6.1.1. Multi-Box Scenario

The constraint condition for the Multi-Box scenario is that AP-MW and Framework must adopt the in-house R&D strategy, and the evaluation results are shown in Figure 7. In the figure, the ordinate is the total vehicle amortization cost of the R&D strategy combination, with the unit being RMB per vehicle; the abscissa is the degree of in-house R&D, where the four R&D strategies are counted as 0/1/2/3, the scenario where all 7 types of strategies are outsourcing is counted as 0, and the scenario where all are in-house R&D is counted as 21.

In the Multi-Box scenario, the larger the production scale, the lower the optimal value of the total average cost per vehicle. From small-scale to ultra-large-scale production, the optimal costs are ¥13,539, ¥9072, ¥7406, and ¥6131, respectively, representing a reduction of 54.7%.

The cost-optimal R&D strategy combination for small-scale production is as follows: in addition to independently developing AP-MW and Framework, all other elements are outsourced. Compared with the highest-cost R&D strategy combination, the cost is reduced by 78.7%.

The cost-optimal R&D strategy combination for medium-scale production is as follows: independently developing Computer Board, AP-MW, and Framework, customizing CP-MW, and outsourcing other elements. Compared with the highest-cost R&D strategy combination, the cost is reduced by 65.9%.

The cost-optimal R&D strategy combination for large-scale production is as follows: independently developing Computer SoC, CP-MW, AP-MW, and Framework and outsourcing other elements. Compared with the highest-cost R&D strategy combination, the cost is reduced by 49.0%.

The cost-optimal R&D strategy combination for ultra-large-scale production is as follows: independently developing Computer NPU IP, CP-MW, AP-MW, and Framework, cooperatively developing AI-OS and Eco-OS, and outsourcing RT-OS. Compared with the highest-cost R&D strategy combination, the cost is reduced by 30.0%.

It can be seen that with the increase in production scale, strengthening the degree of independent R&D by OEMs helps to reduce the total vehicle amortization cost.

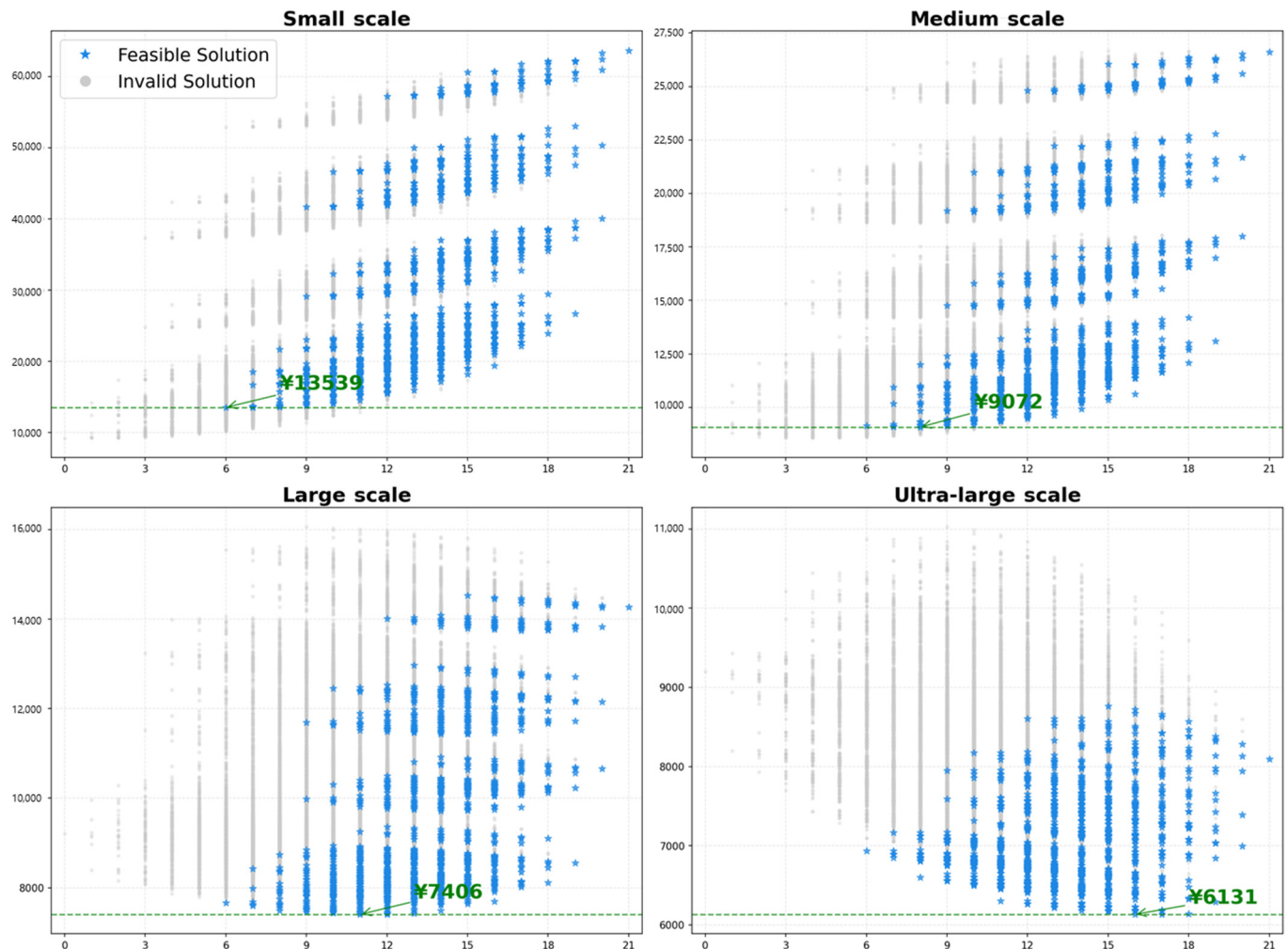


Figure 7. Total vehicle amortization cost of the R&D strategy combination in the Multi-Box scenario.

6.1.2. One-Board Scenario

The constraint conditions for the One-Board scenario are as follows: (1) AP-MW and Framework must adopt the in-house R&D strategy. (2) Computer must adopt the in-house R&D SoC or in-house R&D Core IP strategy. (3) AI-OS must adopt the in-house R&D or cooperation strategy. The evaluation results are shown in Figure 8.

In the One-Board scenario, from small-scale to ultra-large-scale production, the optimal costs are ¥16,899, ¥9979, ¥7612, and ¥6131, respectively. Expanding the production scale can reduce the optimal cost by 63.7%.

The cost-optimal R&D strategy combination for small-scale production is as follows: independently developing Computer SoC, AP-MW, and Framework, cooperatively developing AI-OS, and outsourcing other elements. Compared with the highest-cost R&D strategy combination, the cost is reduced by 73.4%.

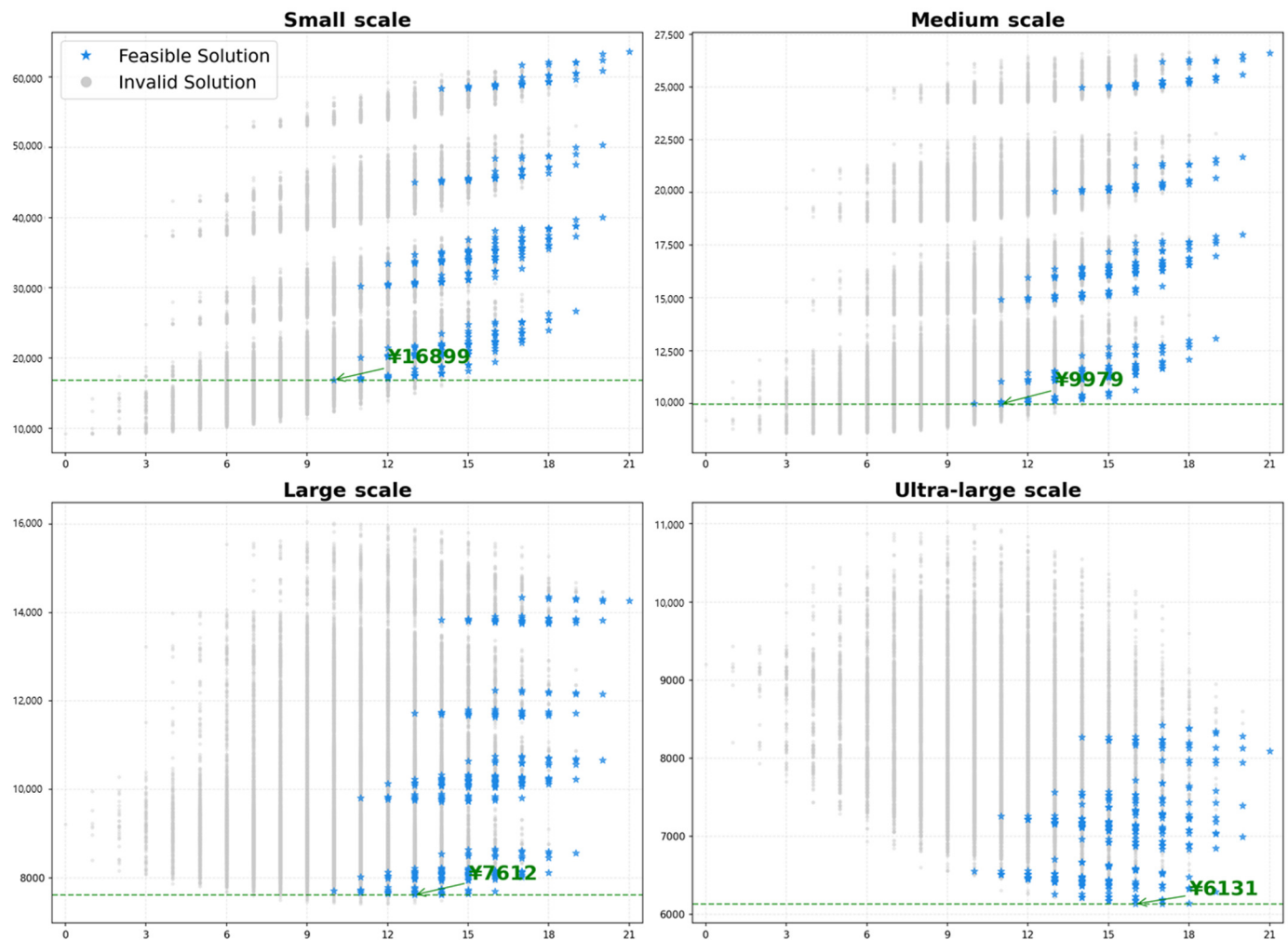


Figure 8. Total vehicle amortization cost of the R&D strategy combination in the One-Board scenario.

The cost-optimal R&D strategy combination for medium-scale production is as follows: independently developing Computer SoC, AP-MW, and Framework, cooperatively developing AI-OS, customizing CP-MW, and outsourcing Eco-OS and RT-OS. Compared with the highest-cost R&D strategy combination, the cost is reduced by 62.5%.

The cost-optimal R&D strategy combination for large-scale production is as follows: independently developing Computer SoC, CP-MW, AP-MW, and Framework, cooperatively developing AI-OS, and outsourcing RT-OS and Eco-OS. Compared with the highest-cost R&D strategy combination, the cost is reduced by 47.0%.

The cost-optimal R&D strategy combination for ultra-large-scale production is as follows: independently developing Computer NPU IP, CP-MW, AP-MW, and Framework, cooperatively developing AI-OS and Eco-OS, and outsourcing RT-OS. Compared with the highest-cost R&D strategy combination, the cost is reduced by 27.2%.

6.2. Decision Making of R&D Solutions by Mass Production Scale

For OEMs with small-scale production, the Multi-Box scenario has a cost advantage of nearly 20% in terms of optimal cost compared with the One-Board scenario (¥13,539 vs. ¥16,899). In absolute terms, the cumulative cost gap over 6 years is as high as 4 billion RMB. The gap mainly comes from the requirement of the One-Board scenario that OEMs must independently develop the SoC of the Computer and cooperatively develop the AI-OS. Therefore, if the profitability of an enterprise's models/brands is insufficient to cover the 3.4 billion RMB cost gap, it is recommended that OEMs prioritize the Multi-

Box architecture temporarily to avoid the cost out of control caused by blind pursuit of high-level architecture and full-stack in-house R&D.

For OEMs with medium-scale production, the Multi-Box scenario still has a 9.1% advantage in optimal cost compared with the One-Board scenario (¥9072 vs. ¥9979). In absolute terms, the cumulative cost gap over 6 years reaches 2.7 billion RMB (mainly due to the requirement of the One-Board scenario that OEMs must independently develop the SoC of the Computer and cooperatively develop the AI-OS). Therefore, OEMs need to carefully balance technical needs and business operations. If the brand premium and profitability can cover the cost gap, the pre-research and technical reserve of One-Board architecture can be carried out; if cost control is the core goal, it is recommended to give priority to Multi-Box architecture.

For OEMs with large/ultra-large-scale production (annual output > 1 million vehicles), the cost difference between the Multi-Box scenario and the One-Board scenario is basically negligible (<3%). On the one hand, this is because the cost-optimal R&D strategy combinations under the two scenarios are nearly convergent; on the other hand, it is because the production scale is sufficient to amortize the R&D investment. Therefore, it is recommended that OEMs prioritize technical advancement and forward-looking layout and accelerate the R&D and implementation of the One-Board architecture platform.

It should be noted that the above decision-making recommendations are mainly based on the supply chain system and cost structure of the Chinese passenger car market. For OEMs in other regional markets, the cost threshold of in-house R&D and the optimal production scale boundary may change due to differences in labor costs, supplier pricing systems, and industry chain maturity, and the specific R&D strategy selection needs to be adjusted in combination with local market conditions.

6.3. R&D Strategy Selection by Technical Elements

Based on the TCO calculation results of all elements, this study puts forward cost-optimal R&D strategy suggestions for OEMs for each of the seven core technical elements, respectively:

- For Computer: Under the Multi-Box scenario, small- and medium-scale OEMs should prioritize full outsourcing or in-house R&D board, while large/ultra-large-scale OEMs can choose in-house R&D SoC/core IP; under the One-Board scenario, small- and medium-scale OEMs should prioritize in-house R&D SoC, and ultra-large-scale OEMs can choose in-house R&D core IP.
- For RT-OS: Third-party mature solutions have complete technology and low pricing, so OEMs of all scales are recommended to directly adopt the outsourcing strategy.
- For Eco-OS: In-house R&D is as difficult as recreating an Android, with extremely high costs, and the industry has formed mature open-source and commercial solutions. Small- and medium-scale OEMs are recommended to directly outsource; only ultra-large-scale OEMs can consider cooperation/customization to achieve differentiation and cost optimization.
- For AI-OS: It has high technical difficulty and large R&D investment. Only ultra-large-scale OEMs can realize that the cost of cooperative R&D is lower than that of outsourcing through large-scale amortization; for OEMs of other scales, if there are no rigid constraints, it is recommended to adopt the outsourcing strategy and entrust professional suppliers for development.
- For CP-MW: It has mature technology and controllable R&D costs, and independent R&D can be gradually realized with the improvement of production scale; small- and medium-scale OEMs can adopt the customization/outsourcing strategy, while large/ultra-large-scale OEMs are recommended to conduct in-house R&D.

- For AP-MW and Framework: These two technologies are necessary conditions for achieving high architectural advancement in both scenarios, and there are no mature and low-cost commercial solutions in the industry. OEMs of all scales must adopt the in-house R&D strategy to grasp core competitiveness.

7. Conclusions

This study takes the central integrated architecture platform of intelligent vehicles as the research object and constructs a two-stage analysis framework of “advancement constraint screening–TCO quantitative evaluation”. Based on the fsQCA method and TCO model, it systematically analyzes the optimal solutions of R&D strategy combinations for OEMs with different production volumes under two scenarios: Multi-Box and One-Board. The main research conclusions are as follows:

First, the implementation of the central integrated architecture platform hinges on mastering core technical elements. The AP-MW and Framework are two core essential components for achieving high architectural advancement, requiring in-house R&D strategies in both the Multi-Box and One-Board scenarios. The One-Board scenario imposes higher requirements on full-chain technical synergy; thus, it also needs to be combined with the in-depth R&D of the central computing platform and AI-OS to meet the advancement requirements of high-level architectures.

Second, there is a significant negative correlation between the TCO of the central integrated architecture and production scale, and the amortization of in-house R&D costs is highly dependent on large-scale mass production. For OEMs with small- and medium-scale production, excessively increasing the degree of in-house R&D will significantly raise the average cost per vehicle. In contrast, for large/ultra-large-scale production, appropriately deepening the degree of in-house R&D can continuously optimize the cost per vehicle. Under ultra-large-scale production, the cost differences between the Multi-Box and One-Board scenarios can be completely negligible.

Third, there is a clear cost-optimal orientation for the R&D strategies of different technical elements. Due to the maturity and low pricing of third-party solutions, RT-OS is recommended for full-industry outsourcing. Given the extremely high R&D costs of Eco-OS and AI-OS, only ultra-large-scale OEMs are suitable for carrying out cooperative/customized R&D. CP-MW can gradually achieve in-house R&D with the improvement of production scale. The in-house R&D depth of the central computing platform needs to match the architectural scenario and production scale. AP-MW and Framework are core elements that all OEMs must consider during in-house R&D.

Fourth, there are differentiated optimal solutions for architectural selection and R&D strategy combinations among OEMs with different production scales. Small- and medium-scale OEMs are prioritized to choose the Multi-Box architecture and adopt the strategy of “core middleware in-house R&D + outsourcing/customization for other elements”. Large/ultra-large-scale OEMs can accelerate the R&D of the One-Board architecture and adopt the combined strategy of “core hardware + AI-OS + middleware collaborative in-house R&D + cooperation/outsourcing for non-core elements”.

This study still has certain limitations. First, the core analysis framework of this study is cost centered under the constraints of architectural advancement and quality compliance and does not directly incorporate the full-lifecycle potential economic benefits of the architecture platform into the quantitative decision-making model. Although we have verified the strong positive correlation between these potential benefits and architectural advancement and completed the feasibility screening based on advancement constraints, the current framework cannot fully quantify the comprehensive return on investment of different R&D strategies in real industrial scenarios. Second, this study does not fully

consider the impact of new business models such as software payment and ecological monetization on the full-lifecycle cost and benefit accounting of architecture platforms [36]. Third, this study does not take into account the impact of external non-technical and non-economic factors such as supply chain security, geopolitics, and OEMs' long-term technological strategic planning on R&D strategy selection [37]. Therefore, future research can be carried out in the following directions: first, construct a full-cycle cost-benefit integrated evaluation model and incorporate the potential economic benefits of centralized architecture platforms into the quantitative decision-making framework; second, explore in depth the impact of new business models such as software-defined vehicle commercialization and ecological monetization on the R&D strategy formulation of architecture platforms; and third, incorporate multi-dimensional non-economic factors such as functional safety, cybersecurity, supply chain resilience, and long-term technological strategy to construct a multi-objective comprehensive decision-making model for R&D strategies of centralized architecture platforms.

Author Contributions: Conceptualization, W.Z. and F.Z.; methodology, W.Z. and Z.L.; validation, W.Z., Z.L. and F.Z.; formal analysis, W.Z. and Z.L.; investigation, W.Z. and Z.L.; resources, W.Z. and Z.L.; data curation, W.Z.; writing—original draft preparation, W.Z.; writing—review and editing, W.Z. and Z.L.; visualization, W.Z. and Z.L.; supervision, F.Z.; project administration, F.Z.; funding acquisition, F.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Tsinghua-Toyota Joint Research Fund under Grant 20253930035 and the Natural Science Foundation of Beijing Municipality under Grant 9232011.

Data Availability Statement: The data generated or used during the current study are available from the corresponding author upon request.

Acknowledgments: The authors hereby express their gratitude to all individuals and enterprises that provided valuable empirical data during the course of this research.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SDV	Software-defined vehicle
EEA	Electrical and electronic architecture
AIDV	AI-defined vehicle
OEM	Original equipment manufacturer
R&D	Research and development
ECU	Electronic control unit
SOA	Service-oriented architecture
PCB	Printed circuit board
SoC	System-on-chip
OS	Operating system
AUTOSAR	Automotive open system architecture
NPU	Network processing unit
IP	Intellectual property
TCO	Total cost of ownership
fsQCA	Fuzzy-set qualitative comparative analysis

Appendix A

Table 1. Truth table for Multi-Box scenario.

Configuration Serial Number	Computer	RT-OS	Eco-OS	AI-OS	CP-MW	AP-MW	Framework	Case Frequency ¹	Raw Consistency ²	PRI Consistency ³
1	1	1	0	1	0	1	1	10	0.962	0.941
2	0	1	1	0	1	1	1	9	0.958	0.935
3	1	0	1	1	0	1	1	8	0.951	0.928
4	0	0	1	0	1	1	1	7	0.947	0.922
5	1	1	0	0	1	1	1	7	0.942	0.915
6	0	1	0	1	0	1	1	6	0.938	0.907
7	1	0	0	1	1	1	1	6	0.930	0.898
8	0	0	0	0	0	1	1	5	0.925	0.890
9	1	0	1	0	1	1	1	4	0.918	0.882
10	0	1	1	1	0	1	1	4	0.912	0.875
11	1	1	1	0	0	1	1	3	0.905	0.868
12	0	0	1	1	1	1	1	3	0.897	0.859
13	1	0	0	0	0	1	1	2	0.889	0.848
14	0	1	0	0	1	1	1	2	0.882	0.839
15	1	1	0	1	1	1	1	2	0.875	0.827
16	0	0	0	1	0	1	1	2	0.868	0.815
17	1	0	1	1	1	1	1	2	0.860	0.802
18	0	1	1	0	0	1	1	2	0.852	0.790
19	1	1	1	1	1	1	1	1	0.845	0.782
20	0	0	1	0	0	1	1	1	0.838	0.775
21	1	0	0	0	1	1	1	1	0.829	0.768
22	0	1	0	1	1	1	1	1	0.821	0.760
23	1	1	1	0	1	1	1	1	0.812	0.753
24	0	0	0	1	1	1	1	1	0.805	0.745
25 *	1	1	0	1	0	1	0	4	0.782	0.721
26	0	1	1	0	1	0	1	4	0.775	0.715
27	1	0	1	1	0	0	1	3	0.768	0.708
28	0	0	1	0	1	1	0	3	0.759	0.699
29	1	1	0	0	1	0	1	3	0.752	0.692
30	0	1	0	1	0	0	1	2	0.745	0.685

Table 1. Cont.

Configuration Serial Number	Computer	RT-OS	Eco-OS	AI-OS	CP-MW	AP-MW	Framework	Case Frequency ¹	Raw Consistency ²	PRI Consistency ³
31	1	0	0	1	1	1	0	2	0.738	0.678
32	0	0	0	0	0	0	1	2	0.729	0.669
33	1	0	1	0	1	1	0	1	0.721	0.662
34	0	1	1	1	0	0	1	1	0.715	0.655
35	1	1	1	0	0	1	0	1	0.708	0.648
36	0	0	1	1	1	0	1	1	0.699	0.639
37	1	0	0	0	0	0	0	1	0.205	0.182
38	0	1	0	0	1	0	0	1	0.198	0.175

¹ Case Frequency: The number of actual brand/model architecture cases corresponding to the conditional variable combination in the research sample database, reflecting the industry adoption of the R&D strategy combination. ² Raw Consistency: The consistency index of the conditional variable combination to the outcome variable (high architectural advancement), which measures the degree to which the combination can lead to high architectural advancement. A higher value indicates a stronger causal relationship, and the threshold for effective configuration in this study is set to 0.8. ³ PRI Consistency: Proportional Reduction in Inconsistency, an auxiliary consistency index to avoid simultaneous subset relationships of the same configuration in both outcome and non-outcome variables, which further verifies the robustness of the causal relationship of the configuration. * The gray case in the table is invalid and will not participate in the subsequent standard Boolean minimization operation.

Table 2. Truth table for One-Board scenario.

Configuration Serial Number	Computer	RT-OS	Eco-OS	AI-OS	CP-MW	AP-MW	Framework	Case Frequency	Raw Consistency	PRI Consistency
1	1	1	0	1	0	1	1	11	0.955	0.932
2	1	0	1	1	0	1	1	9	0.948	0.925
3	1	1	0	1	1	1	1	7	0.941	0.918
4	1	0	0	1	1	1	1	6	0.933	0.909
5	1	0	1	1	1	1	1	4	0.925	0.901
6	1	1	1	1	0	1	1	3	0.918	0.892
7	1	1	1	1	1	1	1	2	0.909	0.883
8	0	1	1	0	1	1	1	10	0.788	0.725
9	0	0	1	0	1	1	1	8	0.779	0.718
10	1	1	0	0	1	1	1	6	0.772	0.710
11	0	1	0	1	0	1	1	5	0.765	0.703
12	0	0	0	0	0	1	1	5	0.758	0.696
13	1	0	1	0	1	1	1	4	0.750	0.688
14	0	1	1	1	0	1	1	4	0.742	0.681
15	1	1	1	0	0	1	1	3	0.735	0.674

Table 2. Cont.

Configuration Serial Number	Computer	RT-OS	Eco-OS	AI-OS	CP-MW	AP-MW	Framework	Case Frequency	Raw Consistency	PRI Consistency
16	0	0	1	1	1	1	1	3	0.728	0.666
17	1	0	0	0	0	1	1	2	0.720	0.659
18	0	1	0	0	1	1	1	2	0.712	0.652
19	0	0	0	1	0	1	1	2	0.705	0.645
20	0	0	1	0	0	1	1	1	0.698	0.637
21	1	0	0	0	1	1	1	1	0.690	0.630
22	0	1	0	1	1	1	1	1	0.683	0.623
23	1	1	1	0	1	1	1	1	0.675	0.615
24	0	0	0	1	1	1	1	1	0.668	0.608
25	1	1	0	1	0	1	0	3	0.482	0.421
26	0	1	1	0	1	0	1	3	0.475	0.415
27	1	0	1	1	0	0	1	2	0.468	0.408
28	0	0	1	0	1	1	0	2	0.459	0.399
29	1	1	0	0	1	0	1	2	0.452	0.392
30	0	1	0	1	0	0	1	1	0.445	0.385
31	1	0	0	1	1	1	0	1	0.438	0.378
32	0	0	0	0	0	0	1	1	0.429	0.369
33	1	0	1	0	1	1	0	1	0.421	0.362
34	0	1	1	1	0	0	1	1	0.415	0.355
35	1	1	1	0	0	1	0	1	0.408	0.348
36	0	0	1	1	1	0	1	1	0.399	0.339
37	1	0	0	0	0	0	0	1	0.105	0.082
38	0	1	0	0	1	0	0	1	0.098	0.075

This table is the truth table for the One-Board architecture scenario. The definition and calibration rules of all columns are completely consistent with Table 1, and the outcome variable is set to high architectural advancement under the One-Board scenario in accordance with the calibration rules in Section 4.1. The gray case in the table is invalid and will not participate in the subsequent standard Boolean minimization operation.

Table 3. Calibration parameters of the COCOMO II model for automotive basic software.

Software Element	R&D Strategy	Exponent Scaling Factors ¹					Effort Multiplier ²				A, B ³
		PREC	FLEX	RESL	TEAM	PMAT	Product Factor	Personnel Factor	Platform Factor	Project Factor	
RT-OS	In-house R&D	3.72	3.04	4.24	3.29	4.68	3.03	0.78	2.21	1.09	A = 3.6 B = 1.2
	Cooperation	2.48	2.03	2.83	4.38	6.24	3.03	0.58	2.21	1.20	
	Customization	1.24	1.01	1.41	2.19	1.56	3.03	0.29	2.21	1.09	
Eco-OS	In-house R&D	4.96	2.03	4.24	3.29	4.68	3.43	1.27	1.53	1.09	
	Cooperation	3.72	2.03	2.83	4.38	6.24	3.43	0.79	1.53	1.20	
	Customization	1.24	1.01	1.41	2.19	3.12	3.43	0.35	1.53	1.09	
AI-OS	In-house R&D	6.2	4.05	5.65	3.29	6.24	4.50	1.39	2.21	1.09	
	Cooperation	4.96	3.04	4.24	4.38	7.8	4.50	0.90	2.21	1.20	
	Customization	2.48	2.03	2.83	2.19	3.12	4.50	0.38	2.21	1.09	
CP-MW	In-house R&D	3.72	2.03	4.24	3.29	3.12	3.43	0.85	2.21	0.89	A = 3.0 B = 1.12
	Cooperation	2.48	2.03	2.83	4.38	4.68	3.43	0.63	2.21	0.98	
	Customization	1.24	2.03	1.41	2.19	1.56	3.43	0.29	2.21	0.89	
AP-MW	In-house R&D	4.96	4.05	5.65	3.29	4.68	4.50	1.55	3.19	0.89	
	Cooperation	3.72	3.04	4.24	4.38	6.24	4.50	1.08	3.19	0.98	
	Customization	1.24	3.04	2.83	2.19	3.12	4.50	0.44	3.19	0.89	
Framework	In-house R&D	6.2	5.07	7.07	3.29	4.68	4.50	1.94	3.19	1.00	
	Cooperation	4.96	4.05	5.65	4.38	6.24	4.50	1.17	3.19	1.10	
	Customization	2.48	3.04	4.24	2.19	3.12	4.50	0.54	3.19	1.00	

¹ Exponent scaling factors: Five types of parameters, including PREC (precedent), FLEX (development flexibility), RESL (risk resolution), TEAM (team cohesion), and PMAT (process maturity), are used to calculate the scaling index E (Equation (2)). A higher value represents higher R&D difficulty and complexity of the project, which will lead to a nonlinear increase in development workload. ² Effort multiplier: Seventeen detailed parameters including the product factor, personnel factor, platform factor, and project factor. The comprehensive value of each category is the product of the corresponding detailed parameters, which adjusts the linear baseline of development workload. A value greater than 1 means the factor will increase the development workload, and a value less than 1 means the factor will reduce the workload. ³ A, B: Core calibration constants of the COCOMO II model. The detailed definition, economic connotation, and calibration basis of the two constants can be found in Section 5.2.

References

- Wang, W.; Guo, K.; Cao, W.; Zhu, H.; Nan, J.; Yu, L. Review of electrical and electronic architectures for autonomous vehicles: Topologies, networking and simulators. *Automot. Innov.* **2024**, *7*, 82–101. [\[CrossRef\]](#)
- De Vincenzi, M.; Bodei, C.; Matteucci, I. When AI Takes the Wheel: AI-defined Vehicles Principles and Pitfalls. *Front. Robot. AI* **2026**, *13*, 1770121. [\[CrossRef\]](#) [\[PubMed\]](#)
- Mausier, L.; Wagner, S. Centralization potential of automotive E/E architectures. *J. Syst. Softw.* **2025**, *219*, 112220. [\[CrossRef\]](#)
- Xie, J.; Qin, Y.; Zhang, Y.; Chen, T.; Wang, B.; Zhang, Q.; Xia, Y. Towards human-like automated vehicles: Review and perspectives on behavioural decision making and intelligent motion planning. *Transp. Saf. Environ.* **2025**, *7*, tdae005. [\[CrossRef\]](#)
- Zhang, W.; Shi, M.; Liu, X.; Ren, L. Review and Prospect of Integration Compatibility in Digital Vehicles: Multi-Dimensional Challenges and Industry Practice. *Machines* **2025**, *13*, 786. [\[CrossRef\]](#)
- Liu, Z.; Zhang, W.; Zhao, F. Impact, challenges and prospect of software-defined vehicles. *Automot. Innov.* **2022**, *5*, 180–194. [\[CrossRef\]](#)
- Zhu, H.; Zhou, W.; Li, Z.; Li, L.; Huang, T. Requirements-driven automotive electrical/electronic architecture: A survey and prospective trends. *IEEE Access* **2021**, *9*, 100096–100112. [\[CrossRef\]](#)
- Askaripoor, H.; Mueller, T.; Knoll, A. E/E designer: A framework to design and synthesize vehicle E/E architecture. *IEEE Trans. Intell. Veh.* **2023**, *9*, 5204–5221. [\[CrossRef\]](#)
- Krog, J.; Suden, A.T.; Schneider, D.; Vietor, T. Definition and use of logical and physical architecture within vehicle concept development. In *Proceedings of the 2022 IEEE International Symposium on Systems Engineering (ISSE), Vienna, Austria, 24–26 October 2022*; IEEE: New York, NY, USA, 2022; pp. 1–8.
- Shetiya, S.S.; Vyas, V.; Renukuntla, S. Zonal Architecture Development with evolution of Artificial Intelligence. *arXiv* **2024**, arXiv:2412.01840. [\[CrossRef\]](#)
- Zhao, F.; Song, H.; Liu, Z. Identification and analysis of key technical elements and prospects for software-defined vehicles. In *Proceedings of the SAE 2021 Intelligent and Connected Vehicles Symposium Part II, Chongqing, China, 4 November 2021*; SAE Technical Paper; SAE International: Warrendale, PA, USA, 2022.
- Zhang, W.; Zhao, F.; Liu, Z. Development Strategies of Intelligent Automotive Industry Under the Background of Increasing Demand for Computational Capacity. In *Proceedings of the Society of Automotive Engineers (SAE)-China Congress, Shanghai, China, 22–24 November 2022*; Springer Nature: Singapore, 2022; pp. 113–128.
- Lu, S.; Shi, W. Vehicle computing: Vision and challenges. *J. Inf. Intell.* **2023**, *1*, 23–35. [\[CrossRef\]](#)
- Khan, M.U.; Gupta, V. *Achieving Reliability, Safety & Security in SDV OS Architecture*; SAE Technical Paper; SAE International: Warrendale, PA, USA, 2026.
- Cui, G.; Zhou, J.; Han, W.; Zhuang, T. High-Performance Peripheral Virtualization with Real-Time Resource Orchestration for Automotive Virtualization OS. In *Proceedings of the 2025 IEEE Smart World Congress (SWC), Calgary, AB, Canada, 18–22 August 2025*; IEEE: New York, NY, USA, 2025; pp. 803–810.
- Henle, J.; Stoffel, M.; Schindewolf, M.; Nagele, A.-T.; Sax, E. Architecture platforms for future vehicles: A comparison of ROS2 and Adaptive AUTOSAR. In *Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), Macau, China, 8–12 October 2022*; IEEE: New York, NY, USA, 2022; pp. 3095–3102.
- Bodei, C.; De Vincenzi, M.; Matteucci, I. Formal analysis of an AUTOSAR-based basic software module. *Int. J. Softw. Tools Technol. Transf.* **2024**, *26*, 495–508. [\[CrossRef\]](#)
- Hackenbeck, C.; Hirmer, P.; Walz, A.; Wagner, S.; Weyrich, M. Towards Standardized Interfaces for Software-Defined Vehicles in Automotive. In *Proceedings of the 2025 IEEE International Automated Vehicle Validation Conference (IAVVC), Baden-Baden, Germany, 30 September–2 October 2025*; IEEE: New York, NY, USA, 2025; pp. 1–7.
- Dinneweth, J.; Boubezoul, A.; Mandiau, R.; Espié, S. Multi-agent reinforcement learning for autonomous vehicles: A survey. *Auton. Intell. Syst.* **2022**, *2*, 27. [\[CrossRef\]](#)
- Mausier, L.; Wagner, S.; Ziegler, P. Methodical approach for centralization evaluation of modern automotive E/E architectures. In *Proceedings of the European Conference on Software Architecture, Prague, Czech Republic, 19–23 September 2022*; Springer International Publishing: Cham, Switzerland, 2022; pp. 165–179.
- Ma, J.; Wang, G.; Lu, J.; Zheng, X.; Yan, Y. Application of multi architecture modelling method in intelligent electric-vehicle design. *Int. J. Prod. Res.* **2025**, *63*, 5493–5511. [\[CrossRef\]](#)
- Askaripoor, H.; Hashemi Farzaneh, M.; Knoll, A. E/E architecture synthesis: Challenges and technologies. *Electronics* **2022**, *11*, 518. [\[CrossRef\]](#)
- Collin, A.; Siddiqi, A.; Imanishi, Y.; Rebentisch, E.; Tanimichi, T.; de Weck, O.L. Autonomous driving systems hardware and software architecture exploration: Optimizing latency and cost under safety constraints. *Syst. Eng.* **2020**, *23*, 327–337. [\[CrossRef\]](#)
- Candelo, E.; Troise, C.; Matricano, D.; Lepore, A.; Sorrentino, M. The evolution of the pathways of innovation strategies in the automotive industry. The case of Fiat Chrysler Automobiles. *Eur. J. Innov. Manag.* **2022**, *25*, 1368–1387. [\[CrossRef\]](#)

25. Szalavetz, A. In-house software development for software-defined vehicles: Major changes ahead in automotive value chains? *Int. J. Automot. Technol. Manag.* **2024**, *24*, 22–39. [[CrossRef](#)]
26. Felser, K. Evolving IT Sourcing Strategies in the German Automotive Industry: A Blueprint for Managing Transition. Ph.D. Thesis, University of Gloucestershire, Cheltenham, UK, 2022.
27. Amiri, R.S.; Saur-Amaral, I.; Coelho, M.C. Legacy Versus Innovation: How Do Established Car Manufacturers Compete With New Entrants? *Int. J. Innov. Manag.* **2025**, *29*, 2540025. [[CrossRef](#)]
28. Chin, K.; Lee, H. Study on Collaborative Activities in Automotive Semiconductor Technology: Focusing on Joint Patent Analysis. *J. Conver. Cult. Technol.* **2024**, *10*, 829–847.
29. Kumar, M.N. Planning a Successful SOC: Activities Beyond Actual Design. In *SOC-Based Solutions in Emerging Application Domains*; Springer Nature: Cham, Switzerland, 2025; pp. 163–171.
30. Kumar, S.; Sahoo, S.; Lim, W.M.; Kraus, S.; Bamel, U. Fuzzy-set qualitative comparative analysis (fsQCA) in business and management research: A contemporary overview. *Technol. Forecast. Soc. Change* **2022**, *178*, 121599. [[CrossRef](#)]
31. Phadnis, T.P.; Feyerabend, N.; Axmann, J. Visualizing and analysing data-driven shift from decentralized to centralized automotive E/E architectures. *Proc. Des. Soc.* **2024**, *4*, 453–462. [[CrossRef](#)]
32. Rohitratana, J.; Altmann, J. Impact of pricing schemes on a market for software-as-a-service and perpetual software. *Future Gener. Comput. Syst.* **2012**, *28*, 1328–1339. [[CrossRef](#)]
33. Zhai, Y.; Vetter, A.; Sax, E. Analysis of current challenges of automotive software in the view of manufacturing. In *Proceedings of the 23rd Stuttgart International Symposium, Stuttgart, Germany, 4–5 July 2023*; SAE Technical Paper; SAE International: Warrendale, PA, USA, 2023.
34. Jeklin, U.; Saad, M.I. Evaluation Of COCOMO Model Accuracy In Software Effort Estimation. *Bull. Inf. Technol. (BIT)* **2025**, *6*, 126–135.
35. Qu, W.G.; Pinsonneault, A. Economic impacts of in-house and packaged software investments: The influence of software investment opportunities. *Ind. Manag. Data Syst.* **2023**, *123*, 746–761. [[CrossRef](#)]
36. Sangannagari, S.R. The Future of Automotive Innovation: Exploring the In-Vehicle Software Ecosystem and Digital Vehicle Platforms. *Int. J. Res. Appl. Innov.* **2022**, *5*, 7355–7367. [[CrossRef](#)]
37. Chistyakov, I.A. The impact of geopolitical challenges on the development of the automotive market in the context of Industry 4.0. *Natl. Secur./Nota Bene* **2025**, *1*, 57–76. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.