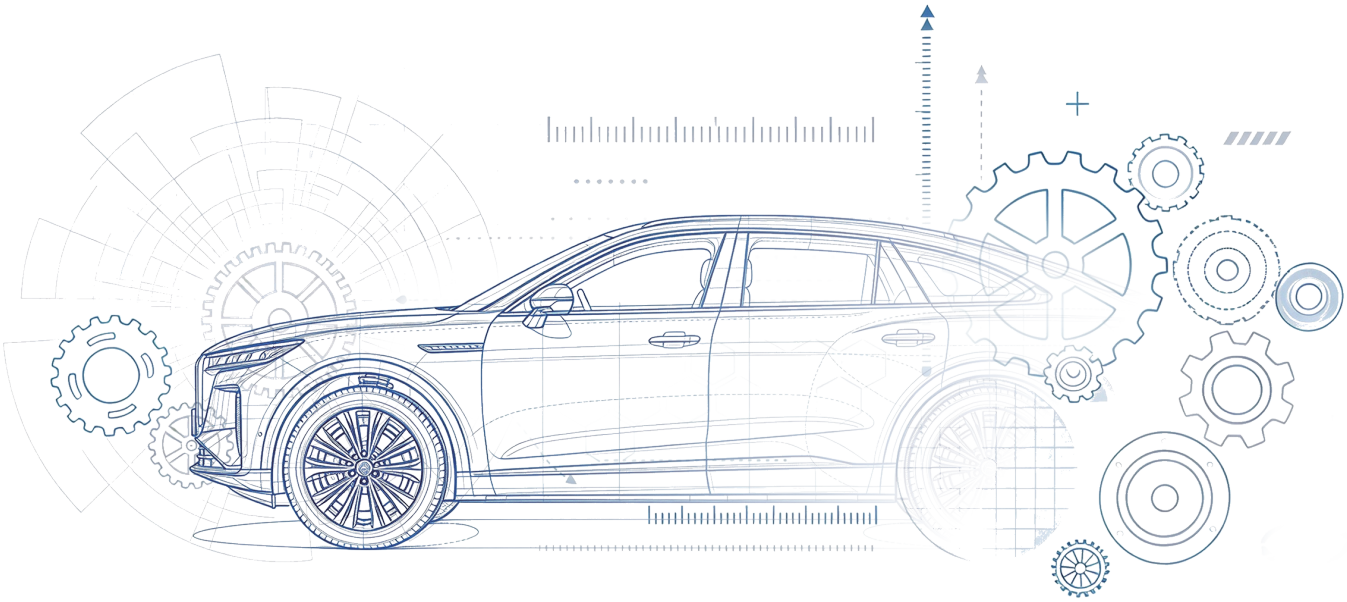
REENGINEERING AUTOMOTIVE SOFTWARE INFRASTRUCTURE

AUTOSAR, CAPI, AI and the Industrial Transformation of the Next Decade

《重构汽车软件基础设施》

AUTOSAR、CAPI、AI与未来十年的产业变革



October 2026

Published by

China FAW Group Co., Ltd.

iSOFT Infrastructure Software Co., Ltd.

Contributing Organization

AUTOSAR

Supporting Organizations

Black Sesame Technologies Co., Ltd.

Axera Semiconductor Co., Ltd.

Hangzhou C-SKY Microsystems Co., Ltd. (Xuantie, Alibaba DAMO Academy)

FPT Software Company Limited

Wuhan Kotei Informatics Co., Ltd.

Special thanks to the experts who contributed to this white paper:

AUTOSAR | Joachim Langenwalter, Steffen Schneider

AUTOSAR China Hub | Jing Zhe, Liu Xiang

FAW Group | Gao Hongwei, Li Yan

iSOFT | Zhang Xiaoxian, Liang Hao

Contents

Executive Summary 6

Chapter 1 Introduction: AI-Defined Vehicles and the Reengineering of Software Infrastructure 7

1.1 Background 7

1.2 Objectives of This White Paper 7

1.3 AUTOSAR's Changing Role in the SDV Era 8

Chapter 2 An Industry at a Crossroads: The Challenges behind the Opportunity 10

2.1 Duplicated Basic Software Development Constrains Differentiation 10

2.2 HPC Software Stacks: From Common Standards to Fragmentation 10

2.3 Adaptive Platform: Overcoming the Barriers to Deployment 11

2.4 Three Engineering Pain Points: Configuration, Integration and Debugging ... 12

2.4.1 Inconsistent Requirements Descriptions 12

2.4.2 The Complexity of ARXML Configuration 12

2.4.3 Cascading Configuration Errors 13

2.5 Open-Source Competition and CAPI's Strategic Response 13

Chapter 3 AUTOSAR's Evolution: Specifications, Shared Implementations and Verification 14

3.1 AUTOSAR's Foundational Value 14

3.2 Continued Evolution of Platform Specifications: 15

3.3 CP and AP: Complementary Platforms 15

3.4 CAPI: Core Value and Operating Model 16

3.4.1 Seven Core Value Pillars 16

3.4.2 Five Strategic Advantages 17

3.4.3 CAPI's Operating Model: A Reinforcing Cycle between Standards and Code 18

3.5 CAPI 1.0 Release 20

Chapter 4 AI-Defined Vehicles: Reshaping Development and Business 23

4.1 AI and AUTOSAR: Integrating Two Software Paradigms 23

4.2 A Practical Approach: AI Assistance under Human Control 24

4.3 How AI Changes the AUTOSAR Engineer's Role 24

Chapter 5 Industry Practice: Perspectives from the Field 25

5.1 China SAE: Vehicle Intelligence Is Advancing Faster than Expected 25

5.2 FAW Basic Software: AI-Assisted AUTOSAR AP Engineering 25

5.3 Xiaomi EV: Practical Cross-Domain Integration 26

5.4 Huawei: The Digital Foundation for Intelligent Vehicles 26

5.5 Alibaba DAMO Academy: RISC-V Moves into Core Automotive Computing 27

5.6 iSOFT: A Decoupling-First Fourth-Quadrant Approach 27

Chapter 6 Outlook and Action Roadmap 27

6.1 AUTOSAR's Transformation 28

6.2 Role-Specific Roadmaps and Recommendations 28

6.2.1 OEMs: From Integrators to Platform Architects and Ecosystem Leaders . 28

6.2.2 Tier 1 Suppliers: From Black-Box Delivery to Transparent Collaboration and System Integration 29

6.2.3 Tool Vendors: From Point Solutions to AI-Enabled Integrated Platforms and Ecosystem Support 29

Appendix A: Glossary 30

Appendix B: Recommended References 32

Executive Summary

The automotive industry is moving from software-defined vehicles (SDVs) toward AI-defined vehicles (AIDVs). As software complexity continues to grow, the need to accelerate development and improve engineering efficiency is becoming increasingly urgent. The adoption of AI at scale further underscores the value of precise, machine-readable specifications. Future development will increasingly follow closed-loop workflows—specification definition → AI-assisted implementation → automated verification → continuous iteration and optimization—while automating the generation of evidence for functional safety and cybersecurity.

Against this backdrop, the AUTOSAR framework is evolving to establish AUTOSAR as a global automotive software foundation for the AIDV era. By bringing together standardized specifications, shared implementations and AI-enabled engineering capabilities, it aims to reduce development costs and accelerate industry innovation while ensuring interoperability.

Through the Common Adaptive Platform Implementation (CAPI), specifications and executable implementations can evolve together. This reduces duplicated development of non-differentiating middleware and frees engineering resources, enabling automakers to focus on AI, automated driving and innovative services for end users.

This white paper systematically examines AUTOSAR’s technical evolution and AI’s impact on development paradigms, drawing on firsthand industry experience from the China Society of Automotive Engineers, China FAW Group, Xiaomi, Huawei, Alibaba DAMO Academy and iSOFT. It provides industry decision-makers, architects and engineers with a fact-based, clearly structured strategic reference for exploring how next-generation automotive software infrastructure will evolve.

Chapter 1 Introduction: AI-Defined Vehicles and the Reengineering of Software Infrastructure

1.1 Background

Automotive electrical/electronic (E/E) architectures are undergoing fundamental change, driven by the vision of software-defined vehicles and accelerated by AI integration. This transition is reshaping how vehicle software is developed, deployed and evolved, with far-reaching implications for AUTOSAR as a core automotive software standard. Discussions and technical demonstrations at AOC 2026 highlighted the evolution of AUTOSAR and its ecosystem, the challenges they face and the new approaches emerging in response.

Rapid advances in large language models (LLMs), generative AI (GenAI), on-device AI agents, multimodal perception and embodied intelligence are taking the automotive industry beyond SDVs toward AI-defined vehicles (AIDVs).

Against this background, the 17th AUTOSAR Open Conference (AOC 2026) adopted the theme of bringing together open-source, community-source and commercial automotive software, in line with AUTOSAR's strategic direction. Participants explored CAPI, AI, safety and security, and platform strategy. The conference presented AUTOSAR's latest strategic thinking and technical direction for the SDV and AIDV eras, marking a transition from an established standardization framework toward a more open, collaborative ecosystem.

1.2 Objectives of This White Paper

As software-defined vehicles (SDVs) advance toward higher levels of intelligence, the automotive software industry is entering a new phase. Centralized computing architectures, high-performance computing (HPC) platforms, service-oriented architecture (SOA), AI, open-source collaboration and continuous software operations are reshaping its technologies, engineering practices and division of responsibilities.

As one of the leading global standards organizations for automotive software, AUTOSAR is undergoing its most profound transformation in more than two decades. From Classic Platform (CP) to Adaptive Platform (AP), from a specification-first approach to the joint evolution of specifications and implementations, and now to CAPI, AUTOSAR's role is expanding from standards development to software ecosystem development.

DDS, Rust, VDP, CAPI, changes to safety and security architectures, and AI-assisted software development are also taking automotive software infrastructure into a new phase.

This white paper examines AUTOSAR's latest developments across technology, architecture, engineering, ecosystems and industry structure, helping readers understand how automotive software architectures may evolve over the next five to ten years.

It focuses on four areas:

1. AUTOSAR's core technical evolution: examining key changes in Classic Platform and Adaptive Platform communication, safety and security, data management, development methods and engineering practice. Particular attention is given to DDS, VDP, Rust, stronger security architectures and expanded platform capabilities, clarifying AUTOSAR's principal technical directions.
2. AUTOSAR's changing role in the SDV era: exploring its evolution alongside centralized computing, HPC platforms, service-oriented communication and open-source ecosystems such as Linux, ROS 2, SOAFEE and Eclipse SDV; and assessing the respective roles and collaboration of Classic Platform, Adaptive Platform and CAPI in future architectures.
3. The impact of new technologies and ecosystems on the value chain: assessing the opportunities and challenges that CAPI, DDS, Rust, AI and data-driven architectures create for OEMs, Tier 1 suppliers, semiconductor companies, toolchain providers and software teams, and examining changes in industry structure and competition.
4. Software infrastructure for AI-defined vehicles: drawing on the practices and technical work presented at AOC 2026 to examine AI's impact on development workflows, software architecture, functional safety, cybersecurity and engineering organizations, and to explore potential technology paths and ecosystems in the transition from SDVs to AIDVs.

This white paper is neither a guide to AUTOSAR specifications nor a substitute for official technical documentation. Its purpose is to connect standards evolution, engineering practice and ecosystem change from an industry perspective, providing automotive software professionals with both technical depth and strategic insight.

1.3 AUTOSAR's Changing Role in the SDV Era

AUTOSAR (AUTomotive Open System ARchitecture) was established in 2003 as automotive electronic systems grew rapidly in complexity. Its purpose was clear: use a common

software architecture, standardized interfaces and a consistent development methodology to address duplicated development, difficult cross-platform porting and inefficient supply-chain collaboration.

Over more than two decades, AUTOSAR Classic Platform has established a global collaborative development framework spanning OEMs, Tier 1 suppliers, semiconductor companies and toolchain providers. Its greatest value lies not in any individual software module, but in enabling a transition from project-based to platform-based development, supporting software reuse at scale and collaboration across the industry.

AUTOSAR originally addressed software complexity primarily at the ECU level. Rapid SDV development now brings new challenges: expanding software, increasingly centralized computing, changing functional deployment and longer software lifecycles. Complexity is extending from individual controllers to vehicle-wide systems.

The industry is also entering a more open and diverse software environment. Linux, Android Automotive, ROS 2, SOAFEE, Eclipse SDV and COVESA are expanding into automated driving, intelligent cockpits, vehicle–cloud integration and development toolchains. As complexity and development costs rise, a traditional specification-driven model alone is increasingly insufficient.

AUTOSAR is therefore undergoing one of the most significant transitions since its founding. Specifications have historically been its primary means of enabling industry collaboration; in the SDV era, specifications alone can no longer meet the full range of needs.

High AP deployment costs, differences among supplier implementations, fragmented HPC software stacks and rapidly growing open-source ecosystems are driving AUTOSAR to evolve from a specification developer into an ecosystem in which standards and implementations advance together:

- From Specification First to Specification + Implementation;
- From closed collaboration to more open industry collaboration;
- From standards development to ecosystem development;
- From delivering specifications to also providing reference implementations.

Alongside continued enhancement of CP and AP, AUTOSAR has made CAPI a strategic priority and is introducing DDS, Rust, Vehicle Data Protocol (VDP) and Remote Persistence to improve its suitability for HPC platforms and open ecosystems.

Over the longer term, AUTOSAR seeks to establish unified software infrastructure spanning MCUs, SoCs, HPC platforms and cloud environments, providing next-generation SDVs with a foundation that combines safety and security, determinism, scalability and continuous evolution.

Chapter 2 An Industry at a Crossroads: The Challenges behind the Opportunity

2.1 Duplicated Basic Software Development Constrains Differentiation

A costly but long-overlooked reality is that OEMs and Tier 1 suppliers have committed substantial development budgets and highly skilled engineers to repeatedly building basic software, including middleware, communication protocol stacks, storage services, security frameworks and diagnostics. Each company bears the full cost of developing this foundational software independently. The result is a vast many-to-many matrix of hardware and software adaptations, difficult integration and duplicated verification.

The consequence is direct: investment and engineering capacity are absorbed while application-level innovation is starved of resources. Teams remain occupied with developing and maintaining foundational platforms and basic software that provide little competitive differentiation. This diverts resources from AI algorithms, automated driving experiences and intelligent cockpit interactions that could distinguish products. As vehicle codebases and complexity continue to grow, isolated development limits systemic innovation across automotive software, costing the entire industry a critical window of opportunity. Distributed exploration is necessary when new technical approaches and development models first emerge. Once the industry moves into volume production, however, efforts must converge rather than remain dispersed across countless implementations of the same foundations.

2.2 HPC Software Stacks: From Common Standards to Fragmentation

In deeply embedded ECUs, AUTOSAR CP was effectively the dominant standard. A common framework substantially reduced the costs of communication, collaboration and maintenance. In centralized architectures built around high-performance processors, a single SoC may run AUTOSAR AP, QNX, Linux, Android Automotive, ROS 2 and AI inference

frameworks, separated by a hypervisor. Simply ensuring that these systems coexist without interference is already a substantial challenge.

Communication adds further complexity. SOME/IP, DDS, gRPC and shared memory coexist, requiring deep integration across frameworks and supplier solutions and causing integration difficulty to increase exponentially. Functional safety and cybersecurity certification boundaries also become less clear, extending verification schedules.

The implication is clear: HPC lacks a universally followed standard comparable to AUTOSAR CP, so collaborative development is once again an unresolved challenge. This diverse ecosystem urgently needs new models for coexistence and cooperation.

2.3 Adaptive Platform: Overcoming the Barriers to Deployment

Designed for microprocessors, Adaptive Platform represents a major advance toward HPC and SOA. This relatively young platform has been progressively developed and refined. As with many new technologies, early production projects faced complex architecture, a steep learning curve, heavy toolchain dependencies and multi-supplier integration challenges. These barriers discouraged many engineering teams. Communication overhead across suppliers and difficult fault diagnosis further slowed delivery.

These engineering barriers raise a practical question: can the industry bypass AP? The answer is clearly no. As SDVs drive E/E architectures from distributed to centralized designs, computing resources are converging on domain controllers and central HPC units. SOA, OTA updates, cross-domain coordination and intelligent driving all require a high-performance, service-oriented runtime. AUTOSAR AP was designed as the industry's standard solution for these requirements and has effectively become essential to next-generation vehicle software architectures. The question is therefore no longer whether to adopt AP, but how to deploy it at lower cost and higher quality. Rebuilding middleware, toolchains and verification capabilities independently increases individual project costs and duplicates resources across the industry, creating a clear need for shared, reusable implementations.

"AP's engineering complexity does not arise solely from AUTOSAR specifications. It also comes from system-level coupling among standards, operating systems, middleware, communication bindings, Execution Management, deployment configuration, toolchains and OEM/Tier 1 supply chains. Reducing deployment costs therefore requires more than simpler specifications: shared implementations, common toolchains, virtual integration and automated verification must work together."

AUTOSAR AP provides a highly flexible service-oriented architecture with strong support for the functional safety and cybersecurity required in automotive systems, giving teams starting HPC software development an established architectural foundation. Despite the barriers to adopting new technology, production use has continued to grow. Users include SDV innovators such as FAW, XPeng, Xiaomi and Geely, while established global manufacturers such as Mercedes-Benz and BMW have progressed to at least their second generation of vehicle implementations.

In practice, developers new to AUTOSAR must first master Execution Management, Software Cluster, State Management, manifests, Machine Design, Service Interface and Deployment Configuration, while handling extensive ARXML configuration and deployment work. Making AUTOSAR AP easier to use in software development remains a significant challenge.

2.4 Three Engineering Pain Points: Configuration, Integration and Debugging

2.4.1 Inconsistent Requirements Descriptions

Real projects rarely express requirements across domains and abstraction levels in a common language. Inputs may come from Excel, DBC, JSON, proprietary OEM formats or custom Tier 1 templates. Even a vehicle-speed signal may be named Vehicle_Speed, Spd_Veh or VehicleVelocity. Engineers spend considerable time reconciling documents, communication matrices, interface definitions and specifications, repeatedly mapping signals, converting names and matching interfaces. Little time remains for implementing logic. These non-standardized elements create substantial additional work.

2.4.2 The Complexity of ARXML Configuration

AUTOSAR comprises more than 30,000 pages of specifications, over 80 basic software (BSW) modules and thousands of configuration parameters. ARXML is the central format carrying this information, yet a considerable conceptual gap separates business requirements from a usable configuration.

The workflow typically runs from requirements definition to interface design, SWC design, port configuration, deployment configuration and ARXML generation. The resulting files can contain tens or hundreds of thousands of XML lines. A small inconsistency at any stage can propagate through the workflow and accumulate into substantial configuration debt.

An industry saying captures the problem: “20% of the work is coding; 80% is fixing ARXML.” For teams new to AUTOSAR, ARXML is far more than a configuration file: it is a major barrier to entry and a source of persistent complexity.

2.4.3 Cascading Configuration Errors

AUTOSAR engineers will recognize the sequence: import a configuration, select Generate, and receive “Error × 100” or even “× 1000.” Dependencies between configuration items are often difficult to see, and a single incorrect parameter can fill the console with cascading errors. Messages can be obscure, reported locations far removed from the root cause, and dependency chains difficult to navigate. Resolution often depends on experienced colleagues who have encountered the problem before, leaving newcomers struggling to make sense of the errors.

Projects enter a recurring cycle: an afternoon changing configurations, followed by two or three days resolving errors. Engineering capacity is consumed by non-differentiating configuration debugging, repeatedly displacing higher-value development.

2.5 Open-Source Competition and CAPI’s Strategic Response

SOAFEE, Eclipse SDV and COVESA have rapidly become significant forces in automotive software. Their approach prioritizes open source, cloud-native compatibility, DevOps, CI/CD and developer experience. Within days, developers can access runnable code, sample projects, automated toolchains and community support, enabling rapid onboarding and iteration. A model long established in IT is increasingly entering automotive development.

This highlights weaknesses in AUTOSAR’s traditional business model, in which AUTOSAR defines standards and commercial software companies implement them: expensive tool licenses, long production development cycles and slow specification updates. The industry continues to assess whether commercial software’s maturity, quality, customization options and product accountability justify its cost. AUTOSAR has therefore confronted a fundamental choice: retain its traditional role as a specification developer, or actively participate in and lead an open software ecosystem.

The underlying issue is consistent. AUTOSAR has traditionally delivered specifications, methodology, templates and interface definitions, while commercial suppliers have implemented the platforms. Each supplier must independently build AP’s foundational

capabilities. Divergent interpretations create interoperability concerns, keep verification and integration costs high, and slow production deployment.

In the AIDV and centralized computing era, software development speed is critical. AUTOSAR has made a fundamental change to better support the evolution of industry development practices.

CAPI is therefore a strategic shift as well as a technical development. It marks AUTOSAR's transition from Specification First toward Code First: extending beyond specification documents to a buildable, verifiable software baseline for production use. This directly addresses AP deployment barriers and strengthens AUTOSAR's ability to support new software development models across the industry in the AIDV era.

Chapter 3 AUTOSAR's Evolution: Specifications, Shared Implementations and Verification

3.1 AUTOSAR's Foundational Value

Despite these challenges, AUTOSAR retains strengths that other ecosystems cannot readily replicate: more than two decades of functional safety experience, a comprehensive OEM–Tier 1 collaboration framework, global production validation, mature software methodology, an extensive toolchain ecosystem and cross-supplier interoperability.

Its core strength is not an individual runtime or middleware product, but an industry collaboration framework combining standards, methodology, toolchains, supply chains and production experience. It defines broadly consistent interfaces and artifacts across companies, allowing application software, basic software and integration processes to operate under common agreements.

CP and AP address different computing and runtime constraints. CP serves embedded control on numerous microcontrollers; AP targets high-performance ECUs and service-oriented software. Both continue to evolve. The wider ecosystem also includes Linux, QNX, Android Automotive, ROS 2, and open-source and proprietary middleware. AUTOSAR provides standardized architecture and engineering constraints where appropriate; it does not require all software to run on a single technology stack.

3.2 Continued Evolution of Platform Specifications:

AUTOSAR's public concept roadmap lists the following directions for R26-11:

- CANsec: an accessible secure communication solution for CAN XL, supporting data rates of up to approximately 10 Mbit/s.
- Dynamic CAN PNCs (Partial Network Clusters): supporting post-SOP E/E architecture changes to improve flexibility with minimal implementation effort.
- SecOC encryption extension: optionally enabling authenticated encryption for selected PDUs, using authenticated encryption with associated data (AEAD) to provide PDU encryption and integrity authentication together.
- Post-build configuration class simplification: simplifying configuration class definitions in Classic Platform AUTOSAR SWS specifications.
- MACsec revisions: retaining existing functionality while adding driver-level interaction without intrusive changes to upper-layer cryptographic modules such as CSM.
- DMA driver standardization: improving the reusability of software components that use direct memory access (DMA).

These items indicate the direction of development in communication security, dynamic network management, cryptographic capabilities, configuration engineering and software reuse. Applicable platforms, interface constraints and final delivery status must be determined from the official R26-11 release documentation and accompanying Release Notes.

Over the longer term, standards must better support reusable software models, automatically checkable engineering workflows and more consistent descriptions of security events. Releases should be assessed not simply by the number of features added, but by whether tools, implementations and tests can interpret them accurately and deploy them reliably.

3.3 CP and AP: Complementary Platforms

CP and AP address software needs under different resource and runtime conditions. CP targets microcontrollers where predictability, real-time behavior and resource constraints are critical. AP targets OS-supported ECUs with greater computing resources that require service orientation and dynamic management. They may coexist across domains and controllers in the same vehicle, working together through common interfaces and communication mechanisms.

AP should therefore not be regarded as a wholesale replacement for CP, nor should Linux, ROS 2 or other stacks be treated as inherently opposed to AUTOSAR. Architecture must determine where functions are deployed, how they communicate, how faults are isolated, who owns safety and security responsibilities, and how lifecycles are managed. Standardized interfaces reduce collaboration costs, but actual interoperability also depends on consistent models, configurations, version management and end-to-end testing.

The concept roadmap for R26-11 addresses communication and security across multiple platforms, illustrating that standards do not evolve solely within individual platforms. Project costs often depend on whether cross-platform interfaces, data models and integration workflows can be described and verified accurately.

3.4 CAPI: Core Value and Operating Model

CAPI is a shared implementation of the AUTOSAR Adaptive Platform specifications, intended to provide a common, continuously evolving code foundation for the industry. Its first release, CAPI 1.0, was officially published on August 20, 2026.

CAPI's licensing model differs from fully open projects such as Linux and ROS 2. It uses Community Source + Commercial Exploitation, a Controlled Community Source model tailored to automotive requirements. In practical terms, its source code can be freely downloaded online, while commercial use is licensed through AUTOSAR partnership agreements, requiring membership in AUTOSAR.

3.4.1 Seven Core Value Pillars

1. **A Shared Code Foundation for Production**
   CAPI's most immediate value is high-quality source code that can be used immediately. It provides a clear starting point for a production implementation, allowing companies to build proprietary product stacks, extensions and customer-specific solutions on a common foundation and significantly reduce time to production.
2. **Standardized Middleware Designed for Certification Readiness**
   To address the high costs of functional safety (ISO 26262) and cybersecurity (ISO/SAE 21434) certification, CAPI aims to provide a certification-ready middleware foundation. Its development processes, documentation traceability and code quality are intended to provide a robust starting point and

supporting evidence for subsequent certification, substantially reducing duplicated compliance work across suppliers.

3. **Industry Collaboration on Shared Foundational Capabilities**
   This is CAPI's central economic rationale. A shared implementation replaces repeated independent development of core middleware with collaborative development of a common foundation. Experts pool their efforts to solve common technical problems and share the results through CAPI, directly reducing non-differentiating investment.
4. **Alignment of Frameworks, Toolchains and Industry Engineering Needs**
   CAPI is more than a code repository: it is an engineering framework that includes reference integration and verification mechanisms. Frameworks and toolchains are aligned with practical industry requirements, providing a ready-to-use development environment that promotes consistency and efficient collaboration among users.
5. **Integration with Open-Source Ecosystems and Modern Infrastructure**
   CAPI adopts modern infrastructure and is designed for straightforward project integration. It marks AUTOSAR's transition from traditional specification development toward code-first, open collaboration, incorporating proven open-source practices to improve development efficiency and ecosystem participation.
6. **Bidirectional Alignment of Code and Specifications**
   CAPI addresses the gap created when implementation trails specification development by keeping code and specifications dynamically aligned. Code-first development feeds improvements back into specifications, making them more precise and less ambiguous. Implementation validation and specification refinement form a reinforcing cycle that makes standards actionable.
7. **Institutionalized IP Management to Reduce Legal Overhead**
   To address intellectual property barriers in cross-company collaboration, CAPI uses AUTOSAR's Exploitation License framework to provide structured IP management. Partners can legally use AUTOSAR's IP pool, minimizing costly, time-consuming legal reviews and removing legal risks to industry collaboration.

3.4.2 Five Strategic Advantages

These value pillars provide five measurable strategic advantages:

1. **Faster production deployment**: a pre-existing, automotive-grade foundation removes the need to build complex middleware from scratch, enabling teams to focus on differentiated functions. It can reduce duplicated basic software work and shorten development and integration cycles for parts of the platform software.
2. **Leaner collaboration and lower complexity**: shared implementations of non-differentiating capabilities eliminate repeated development of equivalent foundational components, significantly reducing development expenditure, project risk and supply-chain integration difficulty.
3. **Designed for automotive safety and cybersecurity engineering**: drawing on AUTOSAR's twenty years of standards development, the foundation is designed around ISO 26262 and ISO/SAE 21434. Its code provides a comprehensive basis for certification and can directly support the safety and security compliance of core vehicle functions.
4. **Standards compliance, interoperability and supplier neutrality**: full alignment with AUTOSAR AP specifications enables seamless interaction among components from different suppliers. The architecture evolves with industry standards, supporting long-term use and simpler certification while reducing dependence on a single basic software supplier and improving supply-chain substitutability.
5. **Greater capacity for ecosystem innovation**: a common industry foundation takes responsibility for generic middleware, freeing OEM and supplier budgets and engineers from duplicated development. Teams can focus on brand experience, advanced automated driving and new in-vehicle business models, enabling differentiated application-level innovation.

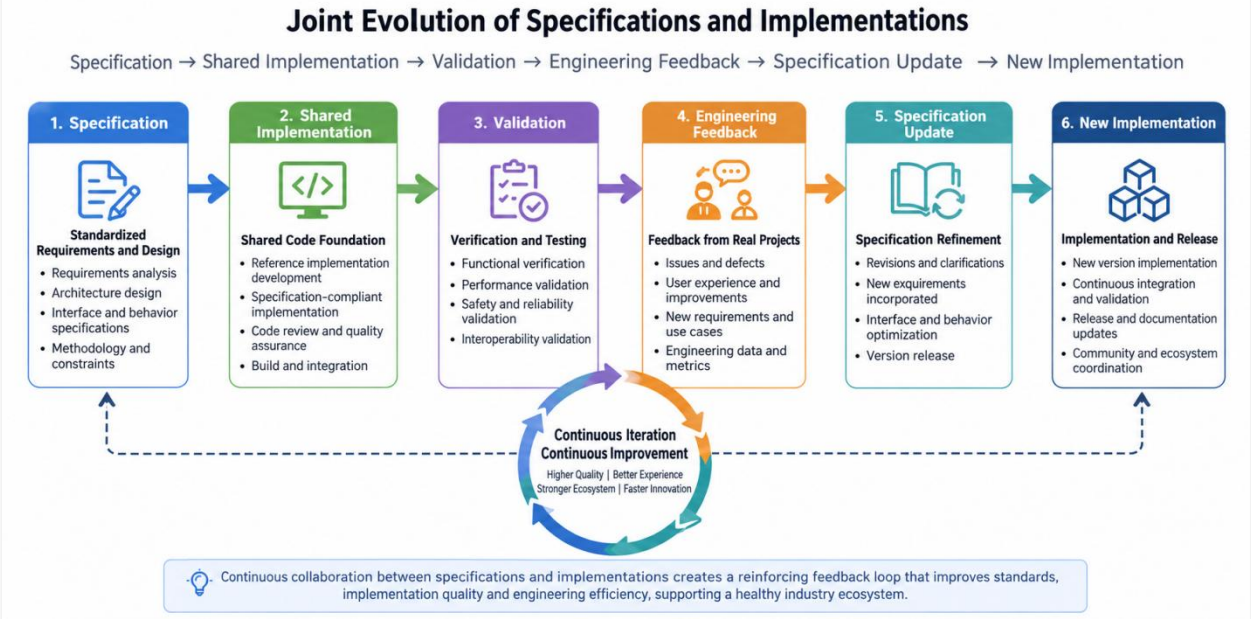
3.4.3 CAPI's Operating Model: A Reinforcing Cycle between Standards and Code

The traditional AUTOSAR model looks as follows: AUTOSAR specifications → independent supplier implementations → OEM product procurement → project deployment

The challenge:

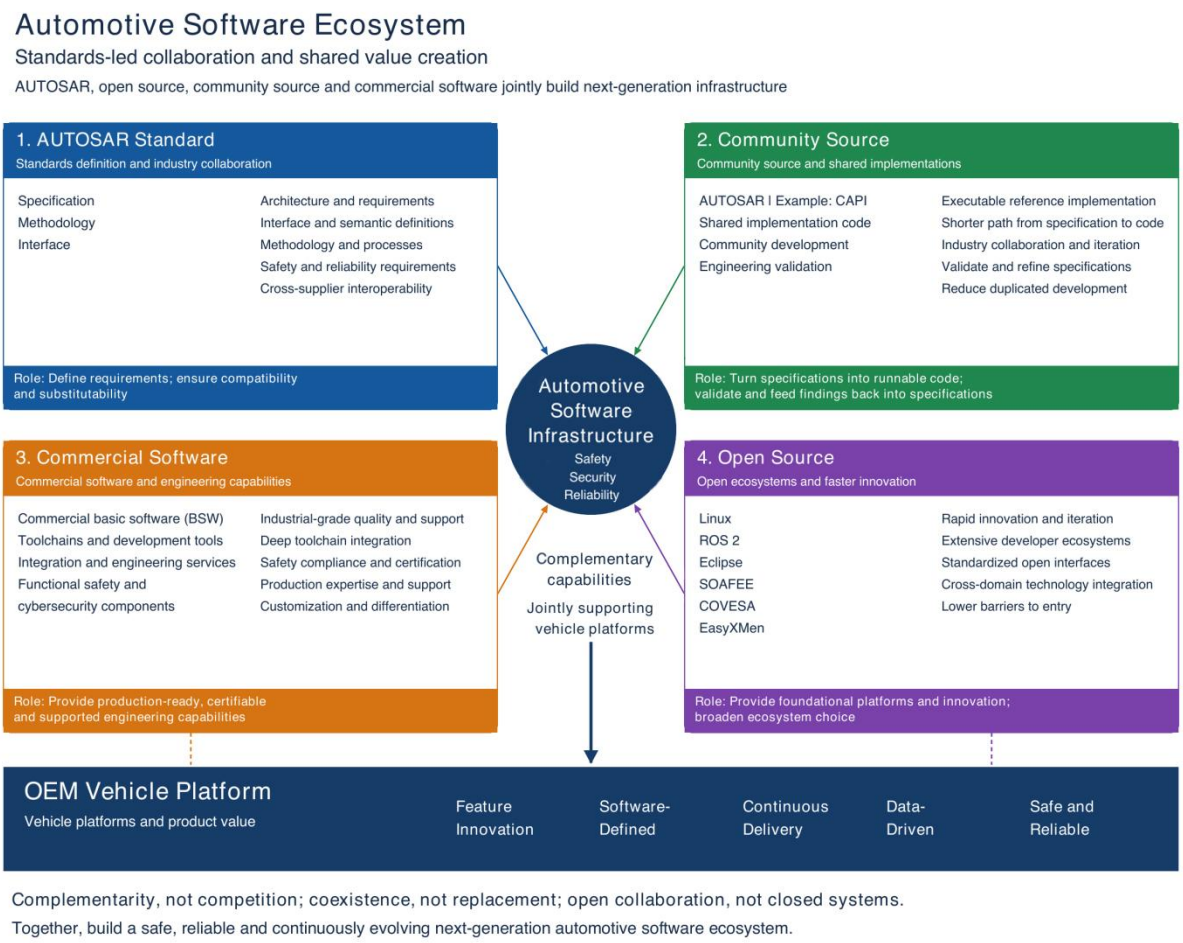
Specification authors cannot fully anticipate the complexity of real projects. Many problems emerge only during implementation or even production, contributing to long specification update cycles.

CAPI enables code and specifications to be developed together, shortening the development and validation cycle.



This means:

1. Implementation experience from real projects feeds back into official AUTOSAR specifications;
2. Development can proceed in parallel across the value chain;
3. The foundation evolves steadily while application innovation accelerates.



3.5 CAPI 1.0 Release

AUTOSAR formally released CAPI 1.0 on 20 August 2026. As the principal contributor of the initial baseline, iSOFT Infrastructure Software Co., Ltd. provided the AUTOSAR community with a complete, production-oriented foundational code base.

Release download: https://github.com/AUTOSAR/capi/releases/tag/v1.0.0

Official CAPI project page: https://www.autosar.org/capi

As the core baseline contributor to this release, iSOFT delivered a complete, production-capable foundational code baseline to the AUTOSAR community. The release is based on AUTOSAR Adaptive Platform R20-11 and provides forward compatibility with selected R23-11 interface definitions. It includes 17 runtime service and foundational component modules, together with an open-source code generator. Coverage comprises Adaptive Core, Communication Management, Cryptography, Diagnostics, Execution Management, Firewall, Intrusion Detection System Management, Log and Trace,

Network Management, Operating System Interface, Persistency, Platform Health Management, Raw Data Stream, State Management, Time Synchronization, and Vehicle Update/Update and Configuration Management.

Core CAPI 1.0 capabilities:

- Communication stack: an in-house SOME/IP stack and zero-copy IPC, with end-to-end (E2E) protection and compatibility with mainstream automotive operating systems including Linux and QNX;

- Execution and configuration management: secure startup, fine-grained hardware resource control, live software updates and automatic rollback on failure;

- Cybersecurity: comprehensive cryptographic services, firewall, intrusion detection, and identity and access management, providing defense in depth;

- Diagnostics: cross-ECU deployment and cost-effective migration of CP diagnostic configurations to AP;

- Log and trace: an asynchronous, non-blocking architecture that reduces CPU consumption under high thread concurrency;

- Platform health management: multidimensional fault supervision, coordinated with a hardware watchdog for recovery and reset.

CAPI 1.0 Establishes a New Automotive Software Development Model for the SDV Era

CAPI 1.0 represents a new development model for SDVs. AUTOSAR is moving from its traditional specification-first approach to industry collaboration driven jointly by specifications and executable code. This ends the need for each participant to reimplement AP's foundational capabilities and provides an automotive-grade software foundation for SDVs and AIDVs.

CAPI 1.0 is a major milestone in AUTOSAR's history: for the first time, the industry has an official, production-oriented shared AP reference implementation. Alongside standards documents, AUTOSAR now delivers buildable, verifiable automotive-grade source code. Independent implementations of the same standard give way to a community-developed baseline, reducing duplicated investment in non-differentiating basic software, improving cross-supplier interoperability and accelerating AP deployment on centralized HPC architectures to support the transition from SDVs to AIDVs.

Statements from AUTOSAR leadership:

1. AUTOSAR Steering Committee: “CAPI 1.0 is an important milestone in AUTOSAR’s development. It marks the point at which AUTOSAR not only defines standards but also enables executable software. It demonstrates that industry collaboration can accelerate innovation while upholding AUTOSAR’s defining principles of safety, quality and interoperability.”

2. AUTOSAR Spokesperson: “CAPI is far more than a software release. It introduces a new approach to automotive software development based on executable implementations, community collaboration and AI-native engineering. It is a significant step toward AUTOSAR’s vision of a global software platform for software-defined and AI-defined vehicles.”

CAPI will continue to expand its capabilities and align with successive AUTOSAR AP specifications through its established release cadence.

- **CAPI Upcoming release**: work toward the next release is planned for Spring 2027, following the established cadence. It will enhance module capabilities, optimize runtime performance and add test cases, strengthening the software foundation for production use.
- **Medium and long-term evolution**: maintain alignment with new AUTOSAR AP specifications; strengthen supporting evidence for functional safety and cybersecurity; expand interoperable demonstrations; and continue refining community governance.

Community collaboration: continuously gather user requirements and evolve the CAPI baseline through joint development

CAPI uses a Controlled Community Source model combining community source with commercial licensing, with source code available to all AUTOSAR partners. OEMs, Tier 1 suppliers, semiconductor companies, toolchain providers and research institutions worldwide are invited to contribute requirements, pull requests and defect reports, jointly evolve the baseline and build next-generation automotive software infrastructure.

Chapter 4 AI-Defined Vehicles: Reshaping Development and Business

4.1 AI and AUTOSAR: Integrating Two Software Paradigms

AUTOSAR's methodology centers on standardization, structure, safety and security, real-time behavior, determinism and traceability. AI, by contrast, is rooted in data, training and probability. AUTOSAR is gradually evolving from a standards framework centered on specifications into an ecosystem that combines specifications with executable implementations for AI-defined vehicles. In the AI era, specifications are becoming even more important. These two approaches converge in the automotive industry, where stringent safety requirements create both challenges and opportunities.

As the industry moves toward AIDVs, vehicle functions are evolving beyond updatability to incorporate intelligence and the capacity for self-improvement. AI capabilities are becoming a core source of competitive differentiation for automakers and an important consideration in consumers' purchasing decisions. Industry stakeholders, toolchain providers and software suppliers urgently need to automate the generation of supporting evidence for functional safety and cybersecurity, reducing certification costs and improving efficiency. The next leap in productivity will extend beyond AI-generated code to the use of AI in building a complete body of certification evidence that can be maintained over the long term.

This broad AIDV trend has three clear dimensions:

1. AI-Native Features

- Definition: vehicle functions whose core logic depends entirely on AI.
- Examples: end-to-end automated driving, generative AI cockpits and world models. These functions would not exist without AI.

2. AI-Enhanced Features

- Definition: AI modules integrated into conventional vehicle or software functions to improve capability or efficiency, or reduce costs.

- Examples: adding AI algorithms to improve engine efficiency or chassis ride comfort.

3. AI-Powered Development

- Definition: using AI to improve development and testing across vehicle subsystems. Even when AI is absent from the final product, it remains a differentiating technology that improves competitiveness in ways end users can experience.
- Examples: AI-assisted code generation, AI-driven automated testing and verification, intelligent variant maintenance, and even AI-assisted mechanical design.

For AUTOSAR, future standards should better support the development and production deployment of AI-native and AI-enhanced functions. The industry has already undertaken numerous experiments in AI-powered development, with initial results.

4.2 A Practical Approach: AI Assistance under Human Control

The pragmatic industry consensus is to use AI to assist engineers while retaining human control. AI serves as an enabling component within established AUTOSAR development, verification and management workflows, with applications in three main areas:

1. AI-assisted configuration: CP codebases and generators are already mature. Using AI to generate tool configurations, then generating code through established tools, is easier to implement than direct AI code generation and better supports code quality and collaboration across teams and companies.
2. AI-driven test automation and verification: automatically generating test specifications and executable scripts, resolving compilation and runtime errors through self-correction loops, running tests and producing reports.
3. AI Loop: assigning individual AUTOSAR development steps to AI agents and linking them into a self-optimizing loop.

4.3 How AI Changes the AUTOSAR Engineer's Role

The convergence of AI and AUTOSAR is redefining engineering roles. As AI takes on repetitive work, engineering value increasingly shifts in two directions:

- Upstream, toward architecture: defining formal specifications and designing architectures.

- Downstream, toward assurance: providing independent verification and final release approval.

Chapter 5 Industry Practice: Perspectives from the Field

5.1 China SAE: Vehicle Intelligence Is Advancing Faster than Expected

By 2030, new energy vehicles will account for 70% of the market, and L2-and-above driving assistance penetration will reach 95%. Future automated driving must combine onboard intelligence with vehicle–road–cloud coordination rather than rely on individual vehicles alone. AI's transformation of the vehicle can be divided into three stages:

- Stage 1: AI in Car (around 2025). AI enhances individual functions such as voice assistance, navigation and driver assistance, while humans remain in control.
- Stage 2: AI Car (around 2030). The vehicle becomes an AI-driven embodied agent that proactively perceives, makes decisions and provides services.
- Stage 3: Car in AI (around 2035). The vehicle becomes a terminal node within a broader AI ecosystem rather than an independent intelligent agent, integrating mobility, work and everyday life.

5.2 FAW Basic Software: AI-Assisted AUTOSAR AP Engineering

As the industry moves toward AI-defined vehicles, FAW continues to develop automotive OS middleware and build production-oriented AUTOSAR AP engineering expertise.

- **AI-enabled tools address configuration challenges**: to tackle the large number of AUTOSAR AP configuration parameters, inefficient manual configuration and the risk of human error, FAW has developed an intelligent automotive OS middleware configuration toolchain that integrates AI into configuration generation, validation and automated issue resolution.
- **Intelligent automation across the workflow accelerates deployment**: the toolchain enables end-to-end intelligent automation from configuration definition to code generation, backed by robust engineering capabilities that support vehicle product development and deployment.

- **Industry collaboration advances standards**: as an AUTOSAR member, FAW continues to use production experience to advance standards and support the shared CAPI baseline. It shares AP engineering expertise with industry partners to establish a sustainable feedback loop across industry engineering activities.

5.3 Xiaomi EV: Practical Cross-Domain Integration

Xiaomi proposes a coordinated approach spanning architecture, collaboration and operations:

- Architecture: logical decoupling, isolation and static binding. Composition SWCs provide logical decoupling; memory protection units (MPUs) isolate OS-Applications; and specific tasks are statically bound to designated CPU cores for multicore scheduling.
- Collaboration: a regular cadence and a common pipeline. An Agile Release Train approach uses two-week sprints, a unified code repository, and a CI/CD pipeline with daily automated builds and tests.
- Operations: mandatory observability. Standardized logs and MCU-level core dumps support diagnosis. Vehicle–cloud coordination combines real-time onboard log collection and local detection with cloud dashboards for fleet-wide monitoring, enabling proactive protection.

Results show support for parallel development by more than ten teams, a reduction in build and integration time from weeks to hours, and an 85% improvement in issue diagnosis and resolution efficiency. AUTOSAR AP is now part of Xiaomi EV's middleware solution.

5.4 Huawei: The Digital Foundation for Intelligent Vehicles

Yinwang's intelligent Digital Vehicle Platform (iDVP) consolidates distributed computing into a central computing system, with regional hubs distributing commands.

Its service-oriented software architecture exposes low-level operations as standardized services, enabling software reuse of up to 90%. Yinwang also provides HAS Studio, an online collaborative toolchain supporting parallel OEM and supplier development to shorten development cycles. Huawei, which became an AUTOSAR Core Partner in 2026, continues to use both CP and AP standards in its intellig•ent automotive systems.

5.5 Alibaba DAMO Academy: RISC-V Moves into Core Automotive Computing

Xuantie at Alibaba DAMO Academy highlights RISC-V's support for supply-chain autonomy and control, freedom from architecture licensing fees, and custom optimization as drivers of its growing role in central vehicle computing. The flagship Xuantie C950 integrates a matrix computation unit directly into the processor core and supports local execution of large language models.

For AUTOSAR integration, RISC-V provides native virtualization support, foundational functional safety and robust isolation, with hardware acceleration tailored to AUTOSAR workloads. The objective is a foundation that is open in both hardware and software.

5.6 iSOFT: A Decoupling-First Fourth-Quadrant Approach

Advanced automated driving faces a mismatch between monthly AI iteration and ASIL D certification cycles that can take one or two years. In a tightly integrated system, changing a single application-level line of code can require recertification of the underlying software.

iSOFT's response is a neutral, shared operating system foundation in the "fourth quadrant," built on a decoupling-first approach:

- Separate safety from AI: certify the foundational safety modules once and keep them stable, allowing application-level AI to iterate independently and resolving the mismatch in development pace;
- Decouple the OS from the silicon;
- Decouple algorithms from the OS.

iSOFT's open-source strategy comprises EasyAda, known as "Da" (the underlying OS), and EasyXMen, known as "EasyXMen" (middleware). Its AUTOSAR AP stack has been contributed to AUTOSAR and adopted as the CAPI project's code baseline.

Chapter 6 Outlook and Action Roadmap

Industry experience presented at AOC 2026 shows AUTOSAR fundamentally renewing its role, from specification developer to a central enabler of the next-generation automotive software ecosystem. As a scalable, safe and reliable automotive-grade

foundation supporting modern development, it helps users accelerate time to market and substantially reduce lifecycle costs.

6.1 AUTOSAR's Transformation

AUTOSAR is moving beyond collaboration centered on specification development toward the coordinated evolution of specifications, shared implementations and engineering verification.

The focus is on:

1. Real platforms and real software
2. Real-world adoption
3. Actual development and delivery speed
4. Tangible improvements in user experience

CAPI 1.0 was officially released on August 20, 2026. As AUTOSAR's executable shared code baseline for Adaptive Platform, its public source code is available on GitHub. CAPI extends beyond specification documents to provide an automotive-grade implementation foundation for execution on ECUs. AUTOSAR Adaptive Platform has already been deployed in vehicle projects at FAW, XPeng and other manufacturers.

6.2 Role-Specific Roadmaps and Recommendations

6.2.1 OEMs: From Integrators to Platform Architects and Ecosystem Leaders

- Short term (within one year): participate actively in AUTOSAR, assign engineers to working groups such as AP standards evolution, and feed production experience back into standards. Define a clear platform strategy, CP/AP responsibility boundaries and mixed-deployment rules.
- Medium term (one to three years): drive supplier adoption of standard workflows and require Tier 1 suppliers to provide AUTOSAR-compliant system design descriptions as contractual appendices.
- Long term (three to five years): establish enterprise-wide data contracts and a unified vehicle data catalog. Lead or jointly develop open-source reference implementations, and open-source suitable non-differentiating internal modules or contribute them to CAPI.

6.2.2 Tier 1 Suppliers: From Black-Box Delivery to Transparent Collaboration and System Integration

- Short term (within one year): adopt standard reference implementations and proactively integrate open-source AP stacks such as CAPI, reducing internal basic software investment and focusing on value-added algorithms. Treat models as deliverables, providing OEMs with design models aligned with AUTOSAR workflows.
- Medium term (one to three years): develop virtual ECU integration capabilities and establish capabilities for multi-instance integration and optimization.
- Long term (three to five years): work with tool vendors to establish AI-assisted development workflows that produce safety-compliant evidence, enabling certifiable AI-generated components. Become software supply-chain managers, handling third-party integration, compatibility and OTA updates for OEMs.

6.2.3 Tool Vendors: From Point Solutions to AI-Enabled Integrated Platforms and Ecosystem Support

- Short term (within one year): broadly integrate AI assistance, embedding AI-assisted generation and verification in configuration tools. Support emerging standard formats and ensure toolchains can import and validate them.
- Medium term (one to three years): create unified design environments with integrated AI-assisted architecture, component design and safety analysis. Extend test automation to interface directly with AI-generated code.
- Long term (three to five years): build automotive software cloud platforms offering end-to-end SaaS services from design and simulation to virtual integration. Enable software marketplaces by integrating automotive component markets and automated integration into tool platforms.

Appendix A: Glossary

Abbreviation	Full Name
AUTOSAR CP	AUTOSAR Classic Platform
AUTOSAR AP	AUTOSAR Adaptive Platform
CAPI	Common Adaptive Platform Implementation
SWC	Software Component
RTE	Run-Time Environment
ARXML	AUTOSAR XML
BSW	Basic Software
DDS	Data Distribution Service
VDP	Vehicle Data Protocol
SOME/IP	Scalable service-Oriented Middleware over IP
ara::com	Adaptive Communication API
EM	Execution Management
UCM	Update & Configuration Management
HPC	High Performance Computing

Abbreviation	Full Name
SoC	System on Chip
Hypervisor	Hypervisor
ASIL	Automotive Safety Integrity Level
A-SPICE	Automotive SPICE
CI/CD	Continuous Integration / Continuous Delivery

Appendix B: Recommended References

- AUTOSAR Classic Platform (CP) Specifications
- AUTOSAR Adaptive Platform (AP) Specifications
- Official Common Adaptive Platform Implementation (CAPI) Technical Blueprint
- Public AOC 2026 Presentations and White Paper Release Materials
- ISO 26262 Road Vehicles — Functional Safety
- ISO/SAE 21434 Road Vehicles — Cybersecurity Engineering
- DDS OMG Specification
- SOAFEE, Eclipse SDV and COVESA Open-Source Architecture and Specification Documents

The complete white paper is available to download online at https://www.i-soft.com.cn/service/report.html.

We welcome your comments and suggestions on this white paper. Please write to CAPI@i-soft.com.cn. We look forward to your feedback and corrections.