archi-intelligence Research Series

Working Paper 2026-01

The Architectural Migration of the Century

From Engineering Ontology to Embodied Intelligence

A Cross-Industry Survey on Architectural Evolution, Benchmarks, and Maturity Frameworks

archi-intelligence Research Team

Released June 2026

archi-intelligence.org

English Edition Note: This is the dual-source English edition of the flagship report, developed in parallel with the Chinese edition rather than machine-translated. Arguments, data, structure, and figures are kept consistent across both editions; localized references and idiomatic expression may differ. See the Chinese edition for the canonical source.

0.0.1 Copyright & License

Digital Object Identifier (DOI)

DOI: 10.5281/zenodo.20357858 Permanent Archive: https://zenodo.org/records/20357858

Suggested Citation

archi-intelligence Research Team. (2026). *The Architectural Migration of the Century: From Engineering Ontology to Embodied Intelligence* (Working Paper 2026-01). archi-intelligence Research Series. https://doi.org/10.5281/zenodo.20357858

BibTeX

```
@techreport{archi_intelligence_2026_01,
  title       = {The Architectural Migration of the Century:
                 From Engineering Ontology to Embodied
                 Intelligence},
  author      = {{archi-intelligence Research Team}},
  institution = {archi-intelligence},
  year        = {2026},
  number      = {2026-01},
  type        = {Working Paper},
  series      = {archi-intelligence Research Series},
  doi         = {10.5281/zenodo.20357858},
  url         = {https://archi-intelligence.org/research/2026-01}
}
```

Contact

- Research Team: research@archi-intelligence.org
- Press Inquiries: press@archi-intelligence.org
- Corrections: corrections@archi-intelligence.org
- Website: https://archi-intelligence.org

0.0.2 Conflict-of-Interest Disclosure

Conflict of Interest and Editorial Independence Statement

To preserve the academic independence and credibility of this research series, the archi-intelligence Research Team hereby discloses the following.

1. About this research institution

archi-intelligence is an independent academic research body dedicated to establishing Architecture Intelligence (AI^2) as a research paradigm. Its mission is to advance the standardization and comparability of cross-industry architectural engineering practices through open methodology, transparent data attribution, and rigorous peer review.

2. On funding and sponsorship

At the v1.0 release stage, this research series received no direct or indirect financial support from any evaluated entity (including, without limitation, the OEMs, Tier-1 suppliers, platform companies, and chip vendors referenced in this report).

During its initial phase, archi-intelligence receives seed sponsorship from Arkimind, a commercial entity focused on the intelligentization of automotive electrical/electronic architecture. This relationship is disclosed explicitly here and is governed by the following isolation mechanisms safeguarding editorial independence:

- The editorial/research team does not report to Arkimind' s sales or business functions.
- Researcher compensation is not tied to any Arkimind commercial metric.
- Arkimind' s commercial team holds no review or modification rights prior to publication.
- The evaluation methodology and primary sources are 100% public.
- Any evaluated entity may submit correction feedback through a public process.

3. On editorial independence

This research' s methodological choices, case evaluations, and conclusions are made independently by the archi-intelligence Research Team. Research judgments are not influenced by any commercial interest, political stance, or geopolitical preference. We acknowledge that this editorial independence must ultimately be verified by readers through ongoing scrutiny of our research quality. We welcome critical feedback from academia, industry, and regulators, and commit to publicly documenting all material corrections in subsequent versions.

4. On sources and the time window

This research draws on public sources from January 2024 to January 2026, including but not limited to: public-company filings and SEC/HKEX disclosures; official OEM and supplier technical releases (AI Day, HDC, IAA, CES, etc.); published patent literature; peer-reviewed academic papers and conference proceedings; industry standards bodies (AUTOSAR, ISO, SAE, IEEE, etc.); and mainstream technical media. For unverifiable rumor or anonymous-source

information, we adopt a conservative posture—either not citing it, or explicitly labeling it as "directional inference based on public reporting."

5. On research limitations

As a survey work, this research openly acknowledges asymmetry in data availability. Public technical detail for Tesla, NVIDIA, and Google is relatively complete, whereas the vehicle-side low-level implementation detail for Huawei, Xiaomi, and some Chinese OEMs is limited in public disclosure. This report strives to flag this asymmetry in comparisons; readers should beware of misreading "differences in disclosure" as "differences in capability."

6. On trademarks and product names

All trademarks, product names, and product codenames mentioned in this report belong to their respective owners. Such names are referenced purely for academic discussion and industry analysis, and do not constitute any commercial affiliation, endorsement, or promotion.

7. Disclaimer

This report does not constitute investment advice, business consulting, or legal opinion of any kind. Any business decision based on the contents of this report is made at the deciding party' s own risk and responsibility.

- Research contact: research@archi-intelligence.org
- Corrections and feedback: corrections@archi-intelligence.org
- Full governance charter: https://archi-intelligence.org/governance

0.0.3 Table of Contents

Contents

0.0.4 List of Figures and Tables

Figures

Tables

0.1 Abstract

Taking "architecture" —an engineering concept spanning two millennia—as its starting point, this paper systematically traces the conceptual history from Vitruvius' s triad to the architecture-description ontology of ISO/IEC/IEEE 42010, and establishes a unified horizontal benchmark framework across five industry cross-sections: automotive electrical/electronic (E/E) architecture, smartphones/consumer electronics, internet/cloud, robotics, and embodied artificial intelligence. The study completes a cross-domain benchmark along five dimensions—**hardware, software, data/AI, ecosystem, and safety**—and identifies three architectural archetypes: the **vertically integrated closed-loop** type (Tesla), the **cross-device ecosystem** type (Huawei, Xiaomi), and the **platform-enablement** type (Google/Waymo, NVIDIA, Mobileye, Bosch, Aptiv). It analyzes in detail their underlying philosophical differences, reuse mechanisms, and constraints.

The central thesis of this study is that **architecture across contemporary industries is undergoing a two-layer evolution: convergence at the infrastructure layer, divergence at the layer of control semantics and burden of proof**. Atop a shared foundation (heterogeneous SoCs, virtualization, service-oriented interfaces, model-training pipelines, OTA pipelines, digital twins), automotive and robotics will long retain independent failure philosophies, safety enclosures, and liability-attribution layers; consumer electronics and cloud will continue to be dominated by fail-soft and rapid recovery. The paper further proposes the **AR0—AR5 architectural capability-threshold framework**, analogous to SAE J3016' s autonomous-driving levels, but explicitly emphasizing "capability thresholds rather than a timeline—lower tiers do not disappear but coexist with higher tiers over the long term." On this basis, the paper introduces **Architecture Intelligence (AI²)** as an overarching research paradigm, and proposes the **AI²-ML (Architecture Intelligence Maturity Levels)** framework—a 5-level × 5-dimension evaluation—as an independent measure of the maturity of architectural tools and methodology.

Keywords: architecture ontology; electrical/electronic architecture; software-defined vehicle; embodied artificial intelligence; architecture maturity; failure philosophy; Architecture Intelligence

---

0.2 Introduction: Why "Architecture"Is the Most Underrated Meta-Question of the Decade

Cars are becoming robots. Robots are becoming mobile data centers. Data centers are reconstructing the physical world in reverse through world models. Smartphones, wearables, and smart-home devices are unifying into a single "super-terminal" via distributed microkernels.

This cross-industry "great architectural migration" —from distributed ECUs to central compute, from monolithic OS to distributed microkernel, from rule-driven control to end-to-end neural networks, from industrial robotic arms to general-purpose humanoids—is not five independent industry stories, but **the projection of one and the same technological curve onto five cross-sections**. Yet industry discussion of this deep isomorphism remains at the level of conceptual slogans, lacking systematic ontological treatment, rigorous cross-domain benchmarks, and a clear characterization of the boundaries of convergence.

More troubling still: over the past decade, the concept of "architecture" has been frequently degraded in use. It is sometimes reduced to a "system block diagram," sometimes equated with a "bill of materials," sometimes confused with "technology selection." The cost of this conceptual dilution is enormous—the irreversibility of architectural decisions, their cross-lifecycle cost, and their long-term shaping of organizational capability are all obscured in the simplified narrative.

This paper attempts to reactivate the seriousness of "architecture" as a meta-question. We begin with two millennia of conceptual history, enter the rigorous definitions of contemporary engineering semantics, then anchor on automotive E/E architecture (because its engineering constraints are the most severe, its regulation the most complete, and its industrial evolution the clearest), benchmark horizontally against the four neighboring domains of phones, cloud, robotics, and embodied AI, and finally return to a unified maturity framework.

This paper is not an industry consulting report; it serves no particular company or geographic market. The research questions are strictly delimited as: **What isomorphism and heterogeneity do contemporary industry architectures exhibit in their evolution? What drives this structural difference? Is there a unified methodology to characterize and compare it?**

0.3 Chapter 1 The Ontology of Architecture: From Vitruvius to Architecture Intelligence

0.3.1 1.1 Origins: the Greek *archi-tekton* and Vitruvius' s triad

The word "architecture" derives from the Greek *ἀρχιτέκτων* (*archi-tekton*), a compound of *archi-* (chief, ruling) and *tekton* (builder), meaning "chief craftsman." Between 30 and 15 BCE, the Roman architect Vitruvius (Marcus Vitruvius Pollio) set out, in *De Architectura* (*The Ten Books on Architecture*), the foundational "Vitruvian triad" :

- **Firmitas** (firmness) —structural integrity and durability
- **Utilitas** (utility) —functionality and fitness for purpose
- **Venustas** (delight) —form and experience

This trinitarian trade-off framework was, two millennia later, rewritten by software architects as the "reliability—functionality—maintainability"triangle of non-functional requirements, and then extended by systems architects into a quality-attribute matrix. Its philosophical essence endures: **architecture is fundamentally an engineering art of trade-offs, not a science of perfection.**

0.3.2 1.2 From pattern language to software architecture: the conceptual leap of the late 20th century

In 1977, the architect Christopher Alexander published *A Pattern Language*, introducing the "pattern" as a reusable unit of design into design theory. In 1987, Kent Beck and Ward Cunningham transplanted Alexander' s pattern language into the software domain at the OOPSLA conference; in 1994, the "Gang of Four" (Gamma, Helm, Johnson, Vlissides) codified 23 classic object-oriented patterns in *Design Patterns*, and software architecture thereby acquired its methodological footing.

Software architecture as an independent discipline arose from the "software crisis" of the 1960s–70s. Key milestones include:

- The **1968 NATO Software Engineering Conference** (Garmisch) first formally established "software engineering" as a term;
- In the same year, Edsger Dijkstra published "The Structure of the THE Multiprogramming System," establishing the **layered-architecture** paradigm;
- In **1972, David Parnas** published "On the Criteria To Be Used in Decomposing Systems into Modules," introducing the **information-hiding** principle;
- In **1992, Perry & Wolf** ("Foundations for the Study of Software Architecture") elevated architecture research from "writing code" to "system-level organization, style, interfaces, and constraints" ;

- In **1996, Mary Shaw & David Garlan** published *Software Architecture: Perspectives on an Emerging Discipline*, formally establishing software architecture as an independent discipline;
- In **2000, the IEEE 1471 standard** was released, evolving into today's **ISO/IEC/IEEE 42010** architecture-description standard.

0.3.3 1.3 The contemporary rigorous definition: ISO 42010, TOGAF, and the Shaw-Garlan-Booch-Kruchten genealogy

In contemporary engineering semantics, "architecture" carries a highly rigorous definition. **ISO/IEC/IEEE 42010** defines architecture as "fundamental concepts or properties of a system in its environment, embodied in its elements, relationships, and the principles of its design and evolution." The standard's core contribution is shifting the focus from the "architecture diagram" to the "**architecture description**," and introducing the four-element conceptual model of **stakeholders**, **concerns**, **viewpoints**, and **views**.

TOGAF (The Open Group Architecture Framework), in an enterprise context, defines architecture as "the structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time." The depth of this definition lies in making explicit that **architecture is not the structure diagram itself, but the design logic and governance method behind it.**

The software-architecture community progressively refined its definitions along the Shaw-Garlan → Booch-Kruchten genealogy. Among these, the **"4+1 View Model"** proposed by **Philippe Kruchten** in 1995 is the most influential framework for architectural expression in engineering practice:

View	Concern	Primary stakeholders
Logical view	Functional requirements, object/service model	End users, requirements analysts
Process view	Concurrency, performance, scalability	System integrators
Physical view	Deployment topology, hardware distribution	Systems engineers, operations
Development view	Module organization, build dependencies	Developers, project managers
+1 Scenario view	Use-case-driven unification and verification	All stakeholders

Kruchten and Grady Booch further noted that architecture encompasses **the set of signif-**

icant decisions about the organization of a system: how to select structural elements and their interfaces, how to define the system's behavior in element collaborations, how to compose base components into more complex subsystems, and how to establish the architectural style guiding overall organization. This definition reveals architecture's **irreversibility**—it defines the boundaries and costs of evolution.

0.3.4 1.4 Two foundational laws of architecture

After half a century of methodological sedimentation, software and systems architecture has converged on two unshakable foundational laws:

First law: Everything is a Trade-off

There is no "optimal" architecture, only an architecture that is "relatively reasonable under specific constraints." Between functional requirements and non-functional requirements (availability, scalability, maintainability, safety, cost), structural conflict always exists. The architect's core work is not to eliminate conflict, but to make conflict explicit, quantify the cost, and provide a traceable chain of reasoning for decisions.

Second law: Why Dominates How

The strategic value of architecture lies not in the "how" of technology selection, but in the "why" of the reasoning behind the choice. The Why determines the system's boundaries of evolution, its fault-tolerance strategy, and the path-dependence of organizational capability; the How merely determines the cost and performance of the current generation's implementation. Architectural decisions that ignore the Why are especially costly in long-lifecycle systems (automobiles, robots, operating-system platforms).

0.3.5 1.5 The contemporary genealogy of architectural views: from structure to platform to ecosystem

After half a century of evolution, contemporary architectural practice has converged on five complementary "views of architecture":

1. **Structural/modular view**: emphasizes component decomposition and interface boundaries (inherited from Parnas's information hiding);
2. **Viewpoint/stakeholder view**: emphasizes that different roles need different architectural views (Kruchten 4+1);
3. **Enterprise-governance view**: emphasizes organization, process, standards, and portfolio (Zachman, TOGAF);
4. **Platform/ecosystem view**: emphasizes cross-product reuse, open interfaces, standardization, and ecosystem coordination (Android HAL/Treble/Mainline, AUTOSAR, Kubernetes);

5. **Quality-attribute engineering view**: emphasizes quantifiable non-functional metrics (ATAM, SAAM).

0.3.6 1.6 The meta-pattern of architectural evolution

Observed over the long durée of history, architectural evolution advances along a clear meta-pattern:

> **Static integration → modularization → service orientation → platformization → cloud-edge coordination → model-driven and AI-native**

What drives this curve is not technological fashion itself, but the fact that **as complexity rises, organizations must isolate change, reuse capability, and manage risk explicitly.** This driving force is especially strong in the era of automobiles, robots, and embodied AI—because these systems simultaneously face the fourfold pressure of high complexity, high safety requirements, long lifecycles, and cross-organizational collaboration.

0.3.7 1.7 Introducing Architecture Intelligence

When architecture evolves to the AI-native stage, a deeper proposition emerges: **if architecture itself is a complex decision system, can AI be used to assist, accelerate, or even automate architectural decision-making itself?**

We call this research paradigm **Architecture Intelligence** (abbreviated **AI²**, for the double "I" in "Architecture Intelligence")—the systematic study of applying intelligent methods (including knowledge graphs, formal verification, large language models, reinforcement learning, and reference-architecture benchmarking) to architectural design, constraint checking, and evolution prediction. Two adjacent but essentially different directions must be clearly distinguished:

- **Architecture Intelligence (AI²)**: using AI to do architecture—studying how to make the architectural-decision process itself more intelligent, more verifiable, and more reusable;
- **Architecture for AI**: designing architecture for AI—studying how to design optimal system architectures for AI workloads (training, inference, deployment).

This paper focuses on the former. Although the two are often discussed in an intertwined way in engineering practice, the core problems they address are entirely different—the former is a methodological problem, the latter an engineering-optimization problem.

Why is Architecture Intelligence needed? The architectural decisions of contemporary complex systems have exceeded the intuitive processing capacity of human architects. A single SDV involves hundreds of SoCs/MCUs, thousands of software components, tens of thousands of constraint rules, and dozens of compliance standards; a humanoid robot involves

coupled control of dozens of degrees of freedom, end-to-end vision-language-action inference, and cross-scenario generalization. At this level of complexity, **the quality of architectural decisions is increasingly limited by the maturity of tool support, not merely by the individual architect' s experience.**

Historically, every major leap in software engineering has been accompanied by a synchronous evolution of tooling—from assembly to high-level languages (the compiler revolution), from waterfall to agile (the CI/CD revolution), from monolith to microservices (the container-and-orchestration revolution). Today, complex-systems architecture itself is undergoing a similar toolification revolution: from hand-drawn diagrams to model-driven design, from discrete constraints to formal verification, from manual benchmarking to AI-assisted decision-making. In Chapter 5, this paper will formally propose the **AI²-ML (Architecture Intelligence Maturity Levels)** framework as a systematic measure of the maturity of architectural tools and methodology.

0.4 Chapter 2 Automotive E/E Architecture: From Distributed ECUs to Central Compute

0.4.1 2.1 Conceptual boundary: E/E architecture is more than harness-and-ECU topology

Automotive electrical/electronic (E/E) architecture refers to **the overall organization of all electronic hardware, sensors, actuators, compute units, communication networks, and underlying software platforms within a vehicle.** Crucially, E/E architecture is not merely the physical topology of harnesses and ECUs; it also includes:

- **Compute layer**: MCUs, domain control units (DCUs), zone control units (ZCUs), high-performance central compute units (HPCs);
- **Network layer**: the communication backbone evolving from CAN/LIN/FlexRay toward automotive Ethernet, TSN (time-sensitive networking), and SOME/IP;
- **Software layer**: RTOS/OS, AUTOSAR Classic/Adaptive, middleware, service bus, diagnostic protocols;
- **Safety mechanisms**: functional safety (FuSa), safety of the intended functionality (SOTIF), cybersecurity, AI safety;
- **Operations layer**: OTA update management, vehicle-cloud coordination, fleet-learning closed loop.

AUTOSAR defines itself as a "standardized software framework and open E/E system architecture." Industry leaders such as Bosch, Aptiv, Google, and NVIDIA define the core of the software-defined vehicle (SDV) as: **replacing a large number of isolated controllers with fewer, more powerful in-vehicle compute nodes, and continuously upgrading through software across the full lifecycle.**

0.4.2 2.2 The physical pressure of evolution: the ECU-count crisis

The immediate historical backdrop to E/E architecture evolution is the runaway growth in ECU count. Public Bosch data show that compact vehicles today typically carry 30–50 ECUs, high-end models exceed 100, and some luxury models approach 150. This numerical explosion brought multiple crises:

- **Harness complexity**: per-vehicle harness length can reach 4–5 km and 50–70 kg in weight, becoming the third-largest cost item in the vehicle BOM;
- **Supply-chain coupling**: each ECU involves an independent supplier, software stack, and verification process;
- **Verification-cost explosion**: cross-ECU integration test scenarios grow combinatorially;

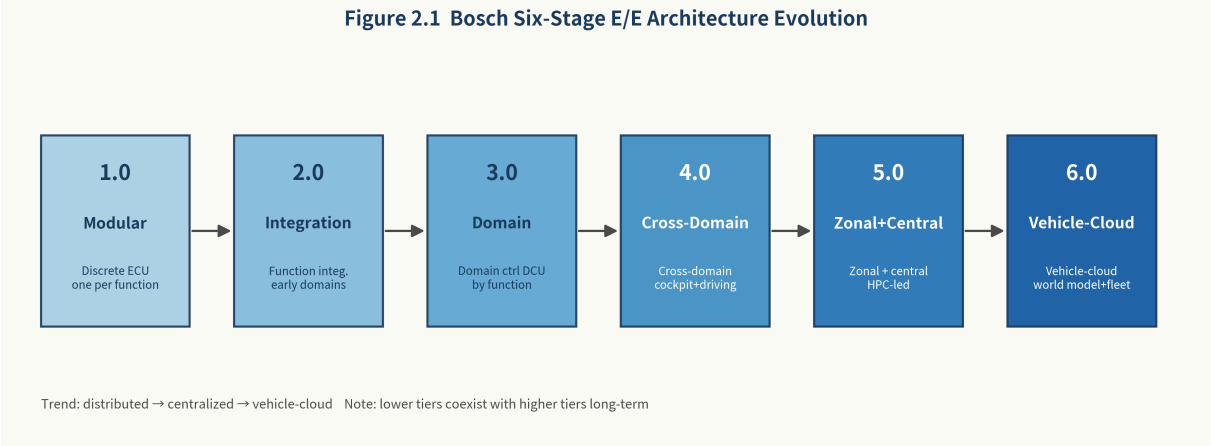

Figure 1: Figure 2.1 Bosch' s six-stage E/E architecture evolution. The trend is distributed → centralized → vehicle-cloud integration; lower tiers do not disappear but coexist with higher tiers over the long term.

- **OTA becomes impractical**: unified update management of heterogeneous ECUs is nearly impossible;
- **Iteration cycle rigidity**: hardware locks function, and software changes require hardware re-certification.

It is precisely this physical pressure that drove the platform-architecture evolution from "distributed functional ECUs" to "domain control" and then to "zonal + central compute."

0.4.3 2.3 Bosch' s six-stage evolution framework

Figure 2.1 Bosch' s six-stage E/E architecture evolution. The trend is distributed → centralized → vehicle-cloud integration; lower tiers coexist with higher tiers over the long term.

The six-stage E/E architecture evolution framework proposed by Bosch and widely adopted across the industry is the most influential current industry consensus:

Stage	Name	Core characteristics	Representative examples
1.0	Modular	One function = one ECU; deep hardware-software coupling; OTA nearly impossible	Most vehicles before the 2010s

Stage	Name	Core characteristics	Representative examples
2.0	Integration	CAN gateway + partial physical/logical ECU integration; preliminary function reuse	Mainstream mid-/low-end ICE vehicles
3.0	Functional Domain	Five major domains (powertrain, chassis, body, cockpit, ADAS); DCU + cross-domain gateway	Audi zFAS, VW MEB, Toyota TNGA, BYD e-Platform 3.0, most EVs 2018–2022
4.0	Cross-Domain	Cockpit-driving fusion on a single SoC; virtualization/Hypervisor partitioning	NVIDIA Thor, Qualcomm SA8775P, XPeng X-EEA 3.0, Li Auto LEEA 3.0
5.0	Central + Zonal	1–2 central HPCs + 3–6 ZCUs; gigabit/10-gigabit Ethernet backbone + TSN	Tesla HW4/AI5, Rivian R1, NIO Cedar, Mercedes MMA
6.0	Vehicle-Cloud	Compute dynamically allocated between vehicle and cloud; end-to-end model + world model; continuous fleet learning	Tesla Dojo→Cortex 2.0, Huawei ADS WEWA, Waymo Foundation Model

McKinsey's 2023 research quantified the supply-chain impact of this evolution: **the traditional ECU market is contracting at roughly 1% annually, while the DCU/ZCU/HPC market is expanding at 30–40% annual growth.** HPC unit price is USD 1,000–4,000; ZCU is USD 50–70. Rivian's electrical-architecture parsimony is emblematic—its latest platform, through a powerful central compute unit, cut the vehicle's traditional ECU count from 40–80 down to **just 7**, thereby eliminating about **1.6 miles (~2.5 km) of physical harness per**

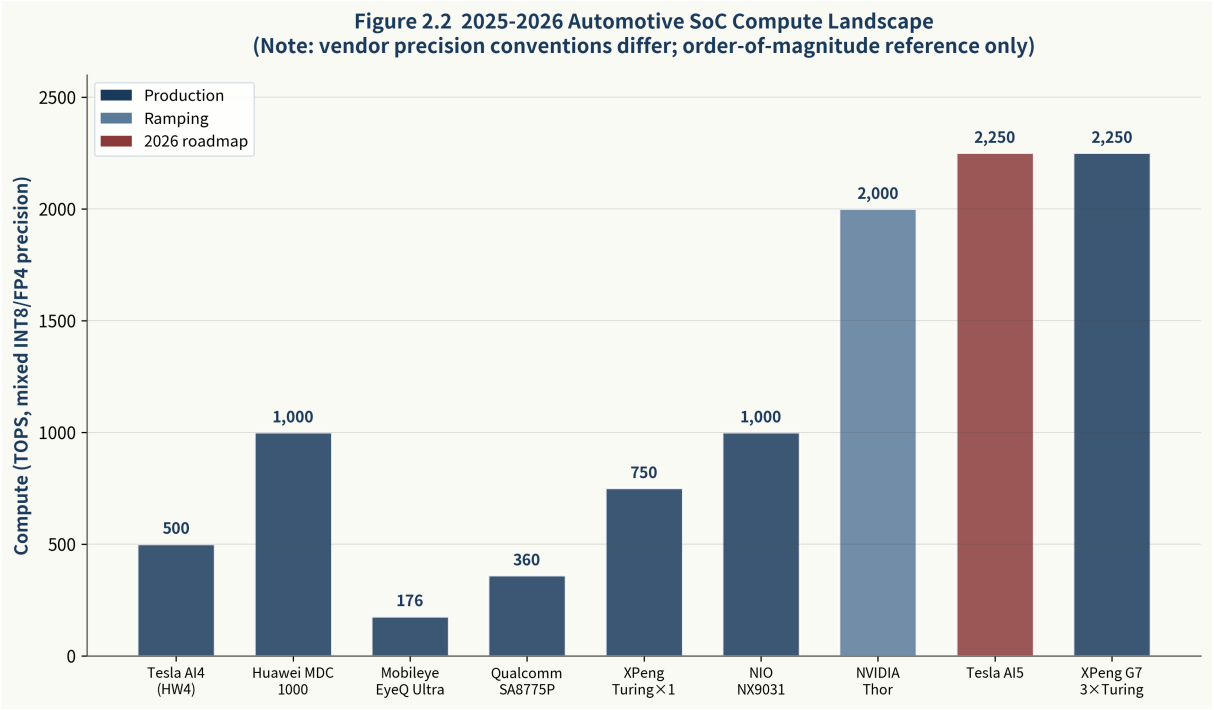

Figure 2: Figure 2.2 2025-2026 automotive SoC compute landscape. Vendor precision conventions differ; figures are for order-of-magnitude reference only.

vehicle.

0.4.4 2.4 The 2025–2026 compute landscape of major players

Figure 2.2 2025-2026 automotive SoC compute landscape. Vendor precision conventions differ; figures are for order-of-magnitude reference only.

Tesla FSD compute evolution: HW3 (2019, Samsung 14nm, 144 TOPS) → HW4/AI4 (2023, ~500 TOPS) → **AI5** (mass production late 2026 / early 2027, TSMC N3P, target 2,000–2,500 TOPS, debuting on Cybercab; the entire SoC evolves into an inference GPU with no standalone ISP/graphics unit) → **AI6** (Samsung Texas USD 16.5B contract, signed July 2025; a single SoC fusing inference + training, serving vehicles, Optimus, and Dojo 3) → **AI7** (2027–28, positioned as "space AI compute"). **A nine-month chip iteration cadence is unique in the industry.**

> Data-verification note: AI5 completed tape-out in April 2026, with up to 192 GB LPDDR5X memory; Musk's claim of "40× faster than AI4" is a composite figure, not a pure TOPS ratio. AI5 and AI6 both use a TSMC (Arizona) + Samsung (Taylor, Texas) dual-foundry strategy.

NVIDIA DRIVE Thor: production ramp 2024–2026, 2,000 TFLOPS FP8, 77 billion transistors, Arm Neoverse V3AE "Poseidon" CPU + Blackwell GPU + Transformer Engine; domain isolation of cockpit and ADAS on a single chip via MIG (multi-instance GPU); full

ASIL-D / ISO 26262 / ISO 21434 certification. Customers include BYD, XPeng, Li Auto, Zeekr, Mercedes, JLR, and Xiaomi (YU7 and the 2026 SU7 lineup all carry the Thor-U 700 TOPS single-chip version).

Qualcomm Snapdragon Ride Flex: SA8775P (5nm, cockpit-driving fusion, ASIL-D) + SA8650P (ADAS-dedicated) + SA8295P (cockpit platform, already widely deployed in Mercedes, NIO, XPeng, Zeekr, Xiaomi, Leapmotor).

Mobileye EyeQ6 High (mass production late 2024, deployment 2025) and **EyeQ Ultra** (12 dual-thread RISC-V cores + 256 GFLOPS Arm GPU + 64 proprietary accelerator cores, <100W, single-chip L4 target 2025–26). Mobileye' s differentiated strategy is REM crowd-sourced HD maps, the RSS formal safety model, and True Redundancy (independent subsystem redundancy of vision and radar).

Three breakthroughs in self-developed chips by Chinese new-energy makers:

- **NIO Shenji NX9031** (announced at NIO IN, July 2024): the industry' s first 5nm automotive-grade autonomous-driving chip, 50+ billion transistors, 32-core big.LITTLE CPU, LPDDR5x memory bandwidth of **546 GB/s**—equivalent to four NVIDIA Orin X—with an ASIL-D safety island and millisecond-level dual-chip failover.
- **XPeng Turing chip** (tape-out August 2024, mass production Q2 2025): a 40-core processor + 2 self-developed NPUs, **750 TOPS per chip**, supporting a 30B-parameter on-board LLM; the XPeng G7 Ultra carries 3 Turing chips = **2,250 TOPS.**
- **Huawei MDC 1000** (based on Ascend 910B): 1,000 TOPS, supporting 16 cameras + 3 LiDARs, with dual-MDC 1000 redundancy and <50ms failover, ASIL-D.

Generational leap of end-to-end neural networks:

- **Li Auto VLA model** (debuting on the i8, July 2025): the first mass-production vehicle-side Vision-Language-Action model, **trained on 1.2 billion km of real vehicle data**, with Mind GPT-3o multimodal end-to-end training on 3 trillion tokens.
- **Huawei ADS 4.0** (released April 22, 2025, **WEWA architecture**): a cloud-side World Engine generating 1,000× extreme-scenario density + a vehicle-side World Action Model native driving foundation model; end-to-end latency −50%, traffic efficiency +20%, hard braking −30%. As of January 2026, a cumulative 8.76 billion km of assisted-driving mileage.
- **BYD God's Eye / DiPilot** (February 2025, free across the lineup): DiPilot 100/300/600 three tiers; daily cloud-training data of 72 million km, drawn from a 440,000+ L2/L3 vehicle fleet.

0.4.5 2.5 The VW CARIAD dilemma and the Rivian alliance: a paradigm of transformation from vertically integrated setback to platform alliance

If Tesla represents the successful paradigm of vertical integration, then VW Group' s CARIAD project is the textbook case of a vertically integrated closed-loop strategy faltering and being forced to pivot toward a platform alliance. Understanding this case is critical to grasping the complexity of automotive-industry architectural evolution.

The generational dilemma of the E3 architecture: CARIAD aimed to build a unified Group Software Stack spanning all VW Group brands, integrating four core technology areas: the group driving stack (autonomous driving), the experience stack (cockpit experience), the cloud stack (connected-vehicle ecosystem), and the motion stack (vehicle dynamics control). Its architecture roadmap was:

- **E3 1.1**: MEB-platform support, partial HCP (high-performance computing platform) integration;
- **E3 1.2**: applied on the PPE platform (Audi Q6 e-tron, Porsche Macan EV), with 5 HCPs covering driving, assistance, infotainment, and safety;
- **E3 2.0**: originally planned exclusively for the SSP (Scalable Systems Platform), targeting a true central-compute + zonal architecture.

However, E3 2.0 development fell severely behind, forcing VW into three major architectural decisions:

Decision one: a geographic-split strategy—SDV West and SDV East. In Western markets, VW and Rivian formed a 50:50 joint venture, "RV Tech," in November 2024, with VW investing USD 5.8 billion and adopting Rivian' s central-zonal hardware architecture and software stack wholesale. The first vehicle, the ID.EVERY1, is planned for mass production in 2027. In the China market, VW chose deep cooperation with XPeng to jointly develop the CEA (China Electronic Architecture) to fit the local digital ecosystem and the extremely competitive iteration cadence.

Decision two: the disruptive parsimony of the Rivian architecture. The Rivian R1 / R2 electrical architecture, through strong central compute, cut the traditional 40–80 ECUs down to just 7, eliminating ~2.5 km of physical harness. Integrating ECUs via virtualization can reduce per-vehicle BOM cost by USD 800–1,500.

Decision three: a strategic alliance with Qualcomm. In January 2026, VW signed a letter of intent with Qualcomm to bring the Snapdragon platform into next-generation cockpit and ADAS systems.

The architectural lesson of this case is highly representative: **when the internal R&D capability of a vertically integrated closed loop is insufficient to support a generational architectural leap, a platform alliance becomes the only viable path.** But the

price of a platform alliance is the partial loss of control over architectural control points—Rivian holds the underlying electrical architecture and software stack, Qualcomm holds the cockpit and ADAS SoC, and VW retreats to the marginal role of "vehicle product definition + brand + manufacturing." This is precisely the micro-level reflection of value-chain rearrangement in the SDV era.

0.4.6 2.6 Reference frameworks and standards matrix

The dual-track software framework: **AUTOSAR Classic** (signal-driven, deeply embedded ECUs, OSEK RTOS) and **AUTOSAR Adaptive** (SOA, ARA::COM SOME/IP, POSIX, oriented toward SDV/AD/HPC) currently coexist in the R24-11 version. The former targets hard-real-time and functional-safety constraints; the latter introduces service orientation and high compute.

The cloud-native automotive stack: **SOAFEE** (Arm-led, founded 2021; joined by Bosch, Continental, ZF, AWS, Red Hat) brings the cloud-native paradigm (containers, Kubernetes orchestration) into the mixed-criticality automotive edge. **COVESA** (formerly GENIVI, renamed 2022) maintains VSS (Vehicle Signal Specification) and the vSomeIP open-source implementation.

The robotics stack enters the vehicle: **ROS 2 Jazzy Jalisco** (May 2024, current LTS) enters production vehicles through Apex.AI (ASIL-D-certified ROS 2) and Autoware. Between 2024 and 2026, academia and industry have reached consensus on the **complementarity of AUTOSAR AP and ROS 2**, with the mainstream approach bridging via a SOME/IP-DDS gateway.

The safety-and-compliance matrix:

Standard	Scope	Status
ISO 26262 (2018 v2)	Functional safety; ASIL A–D	Foundational standard
ISO 21448 SOTIF	Safety of the intended functionality	Required for ADAS/AV
ISO/SAE 21434 (2021)	Cybersecurity engineering full lifecycle	UN R155 statutory baseline
UN R155	Cybersecurity Management System (CSMS)	Mandatory for all new vehicles July 2024
UN R156	Software Update Management System (SUMS)	Same as above
ISO 24089 (2023)	Software update engineering	Engineering of R156
ISO/PAS 8800 (2024)	Automotive AI safety	Released December 2024

Standard	Scope	Status
EU AI Act (2024/1689)	High-risk AI systems	Vehicle AI safety components fully applicable 2027
EU Data Act (2023/2854)	Connected-device data access	Applicable September 2025
EU CRA (2024/2847)	Digital-product resilience	Fully applicable December 2027
GB 44495/44496/44497-2024 (China)	Vehicle cybersecurity/software update/data storage	Mandatory for new models January 2026

The deeper implication of this standards matrix is that **the evolution of automotive architecture is not a simple convergence toward the smartphone, but a convergence toward a regulated platform that is "upgradeable yet auditable, intelligent yet provable."** This is the fundamental reason automotive and consumer electronics maintain long-term divergence in architectural philosophy.

0.5 Chapter 3 Cross-Industry Benchmarks: Five-Dimensional Matrix, Failure Philosophy, and the Two-Layer Structure

0.5.1 3.1 The five-dimensional benchmark matrix

The table below condenses the architectural choices of the five major industries across five dimensions—hardware, software, data/AI, ecosystem, and safety:

Dimension	Automotive E/E	Smartphone / Consumer Electronics	Internet / Cloud	Robotics	Embodied AI
Hardware	Thor 2,000 TOPS, HW5/AI5, MDC 1000, cockpit-driving fusion SoC + zone controllers	A19/M5 Neural Engine 45 TOPS, Snapdragon 8 Elite Gen5, Kirin 9030	Blackwell GB200 NVL72 1.44 EFLOPS FP4, TPU v7 Ironwood 4.6 PFLOPS, Trainium2	Jetson Thor 2,070 TFLOPS FP4, electric/tendon-driven actuators, camera+LiDAR+tactile	As above + cloud training (Cortex/Dojo 3, CloudMatrix 384)
Software	AUTOSAR Classic/Adaptive, QNX, Linux, Tesla in-house, HarmonyOS-Auto	iOS/Darwin, Android/Linux, HarmonyOS NEXT microkernel, HyperOS hybrid kernel	Linux + Kubernetes (82% production adoption 2025) + service mesh + microservices + Serverless	ROS 2 DDS + Nav2 + MoveIt2, AUTOSAR-AP bridge	VLA + System1/System2, Diffusion Policy, Flow Matching

Dimension	Automotive E/E	Smartphone / Consumer Electronics	Internet / Cloud	Robotics	Embodied AI
Data/AI	Vehicle-side end-to-end NN (FSD V14, ADS 4.0 WEWA, Mind GPT VLA) + cloud-training fleet learning	On-device 3–8B model + private cloud compute (Apple PCC, AICore, Pangu) + LoRA	Hundred-billion to trillion-parameter training, FP8/FP4 precision	GR00T N1, $\pi 0/\pi 0.5$, Helix, Gemini Robotics, Skild Brain, OpenVLA	NVIDIA Cosmos world model + Isaac Sim + Omniverse + Wayve GAIA-3
Ecosystem	OEM full-stack (Tesla) / Tier-1 alliance (Huawei HIMA) / platform (NVIDIA, Qualcomm)	Walled garden (Apple, Huawei) / platform sharing (Android)	Open-source-led (CNCF 15.6M developers) + hyperscalers	Vertical integration vs. horizontal platform	NVIDIA three-computer paradigm (train-simulate-run)
Safety	ISO 26262 + 21434 + R155/156 + ISO/PAS 8800 mandatory	TEE/Secure Enclave + GDPR/PIPL + E2E encryption; no FuSa	SOC 2 / ISO 27001 / zero-trust / confidential computing	ISO 13482/10218 (2025 update)/13849	Standards gap; new proposals such as DeepMind ASIMOV Benchmark

0.5.2 3.2 Hardware convergence: the convergence of the global architectural substrate

Silicon platforms exhibit a pronounced cross-industry convergence trend. **Arm Neoverse V3AE appears simultaneously in NVIDIA Drive Thor, Grace, Vera, and AWS Graviton4**; **Blackwell GPU IP simultaneously powers the B200 data-center GPU**

and the Drive Thor automotive SoC. This is the first time in history that automotive and supercomputing share the same architectural substrate—meaning capability spillover between automotive OEMs and cloud hyperscalers will flow **bidirectionally.**

Heterogeneous computing (CPU + GPU + NPU + DSP + dedicated accelerators) has become the default paradigm across all industries. The difference lies not in "whether it is heterogeneous" but in **the compute mix, determinism guarantees, thermal design, and cost targets.**

0.5.3 3.3 Failure philosophy: the fundamental watershed of fail-soft vs. fail-operational

Hardware converges, but software diverges. **The real difference lies not in "whether there is OTA" or "whether Linux/Android is used," but in failure philosophy.**

Phones and the internet are closer to fail-soft + rapid recovery: a service crash can switch to a backup node in milliseconds; an app crash can restart immediately; a failed OTA can roll back. This soft-real-time architecture of "allowing failure and backstopping via redundancy" is impeccable in the virtual world. Android can continuously update system modules via Mainline; Kubernetes can do rolling upgrades and rollbacks. The cost of failure is degraded user experience, not physical harm.

Automotive and many robots demand fail-safe / fail-operational + explicable liability attribution: when a sensor detects a fatal obstacle, the time window for the compute platform to command the brake calipers is absolutely closed; when an actuator fails, the system must degrade to a safe state rather than crash; when an accident occurs, there must be an auditable decision chain for attributing responsibility. Vehicle updates must pass through safety analysis, version matching, and regulatory and supply-chain coordination, and **post-update behavior is not permitted to break verified safety boundaries.**

This difference in failure philosophy is the **true root** of the long-term architectural divergence between automotive and consumer electronics—deeper than real-time requirements, earlier than compliance. It directly determines:

- Why automobiles cannot simply run Linux but must embed AUTOSAR or a microkernel RTOS at the bottom layer;
- Why OTA is ten times more complex on the vehicle side than on the phone side;
- Why ASPICE, ISO 26262, and ISO 21434 can never be replaced by consumer-electronics processes;
- Why end-to-end neural networks on the vehicle side must be paired with an explicable intermediate layer and a runtime monitor.

0.5.4 3.4 The Hypervisor as the bridge across the digital-physical divide

The architectural systems of automobiles and high-degree-of-freedom robots face a profound contradiction: on the one hand, they **strongly desire an open-source, flourishing application ecosystem at the Android/Linux level** to host AI large models that require vast memory; on the other hand, at life-and-death moments they **desperately depend on the hard-real-time intervention mechanisms of AUTOSAR/RTOS** to ensure that, even when memory overflows or an upper-layer application crashes, the brake calipers and robotic arm can still execute absolutely safe physical commands.

In this "fish-versus-bear' s-paw" contest, **the Hypervisor virtualization layer is currently the only viable universal remedy.** The bottom layer runs AUTOSAR Classic in strict compliance with ISO 26262 ASIL-D, focused on hard-real-time braking and steering control; the upper layer runs Android Automotive, Linux, or a custom cockpit OS, specializing in cockpit entertainment and navigation; the two are strictly isolated by the Hypervisor on the same powerful SoC, forming a mixed-criticality architecture that is "interconnected yet mutually non-interfering."

QNX Hypervisor, COQOS, Hypervisor for ARM TrustZone, and the open-source Xen Project Automotive are the current mainstream choices. NVIDIA Drive Thor extends this virtualization paradigm from the CPU domain to the GPU domain via MIG (Multi-Instance GPU), giving ADAS and the cockpit hardware-level isolation on the same Blackwell GPU. This is a landmark technical breakthrough in the history of automotive architecture.

0.5.5 3.5 The edge-cloud division of labor in data/AI architecture: bidirectional convergence

Smartphones (on-device 3–8B parameters + private cloud compute) and automobiles (vehicle-side VLA + cloud-side world model) are **almost identical in architectural pattern**: constrained devices do real-time inference, the cloud does training + simulation + hard-case data generation.

The NVIDIA Cosmos world model has been adopted simultaneously by Toyota (DriveOS), 1X, Skild AI, Figure AI, Agility, Uber, Waabi, Foretellix, XPeng, Galbot, and Fourier—automotive and robotics are sharing the same world-model infrastructure. This is the most underrated fact of this architectural revolution.

The specific edge-cloud division of labor presents three typical paradigms:

1. **Small model local + large model cloud** (Apple Intelligence, Google Tensor + Cloud TPU): a 3–8B-parameter model resides on the device, triggering calls to cloud-side Gemini/GPT-class large models;

2. **Full model local inference + incremental learning in the cloud** (Tesla FSD, Huawei ADS): the device runs the complete end-to-end NN, the cloud only does training and version delivery;
3. **Edge-cloud dual-model coordination** (Waymo System 1/System 2, see Chapter 4): an on-device fast-thinking module + a cloud-side slow-thinking VLM, dynamically switching by scenario complexity.

0.5.6 3.6 Ecosystem: three philosophies of mobilization

A cross-industry observation reveals three ecosystem-building philosophies:

- **Vertically convergent** (the Tesla paradigm): a single enterprise develops the full stack from silicon, OS, and applications to cloud and robots;
- **Horizontal microkernel + large-model platform** (the Huawei paradigm): one microkernel OS + one foundation large model + one AI-accelerator substrate, covering phone—PC—vehicle—cloud—embodied robot;
- **Federated platform + best-partner** (the Google paradigm): TPU + Gemini provided to internal products and external partners.

Behind each philosophy is a different "architectural control-point" holding strategy. We will systematize this concept in Chapter 6.

0.5.7 3.7 The two-layer structure: infrastructure converges, control semantics and burden of proof diverge

Figure 3.2 Two-layer convergence-divergence model. The first layer (shared infrastructure) keeps converging; the second layer (industry-specific upper layer) keeps diverging.

Synthesizing the benchmark across five dimensions, the central judgment of this study is: **the future will not fully "merge into one architecture," but will form a two-layer structure.**

First layer: the shared foundation (convergence)

- Heterogeneous compute (CPU + GPU + NPU + dedicated accelerators)
- Ethernet backbone + TSN
- Virtualization and mixed-criticality OS
- Service-oriented interfaces (SOA, SOME/IP, DDS, gRPC)
- Digital twins and world-model simulation
- End-to-end model training and deployment pipelines
- OTA pipelines and continuous deployment

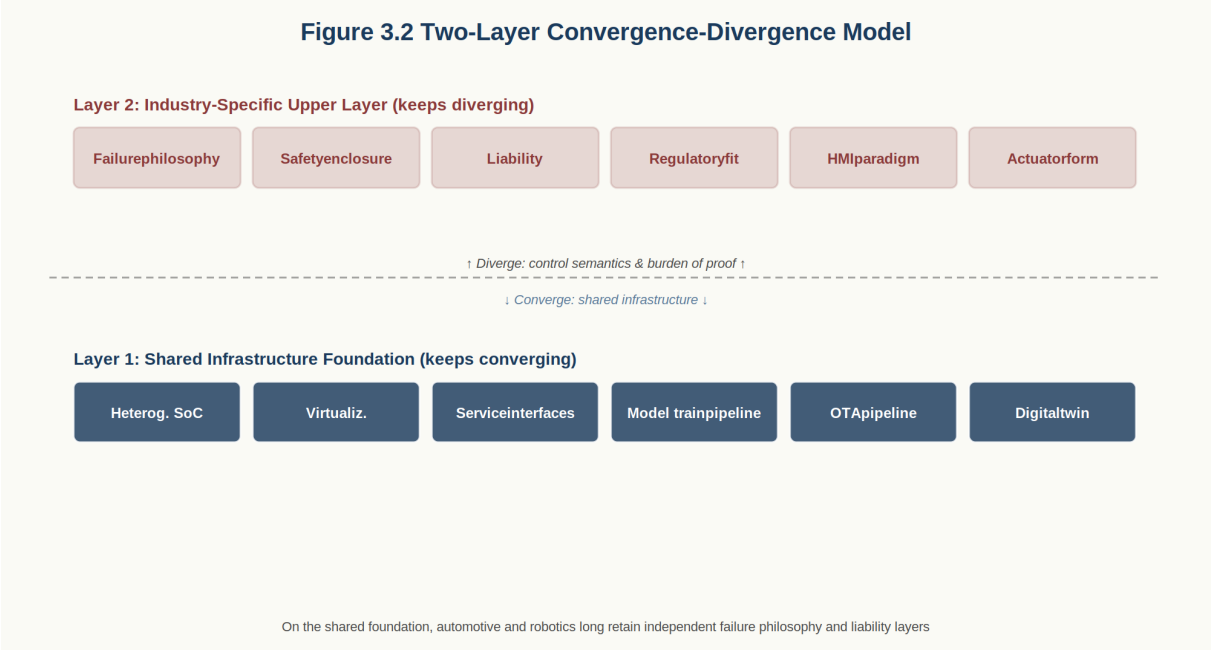

Figure 3: Figure 3.2 Two-layer convergence-divergence model. The first layer (shared infrastructure) keeps converging; the second layer (industry-specific upper layer) keeps diverging.

This part will converge markedly, and cross-industry boundaries will blur. SoC vendors, cloud platforms, and open-source ecosystems will dominate this layer.

Second layer: the industry-specific upper layer (divergence)

- Vehicle/robot physical-control layer
- Safety enclosure and failure-mode management
- HMI and user-interaction paradigms
- Operations strategy and lifecycle management
- Liability proof and auditability
- Regulatory adaptation and compliance frameworks

This part will continue to diverge, further solidified by intensifying regulation. Tier-1s, OEMs, robot-body manufacturers, and domain experts will dominate this layer.

In one sentence: infrastructure converges; control semantics and burden of proof diverge.

This two-layer judgment is more precise than the simple binary of "will converge / will diverge." It explains why we see both Tesla seamlessly migrating the FSD compute stack to Optimus (the shared foundation) and automotive and robotics maintaining independent evolutionary paths in functional safety, liability frameworks, and HMI (the industry-specific upper layer).

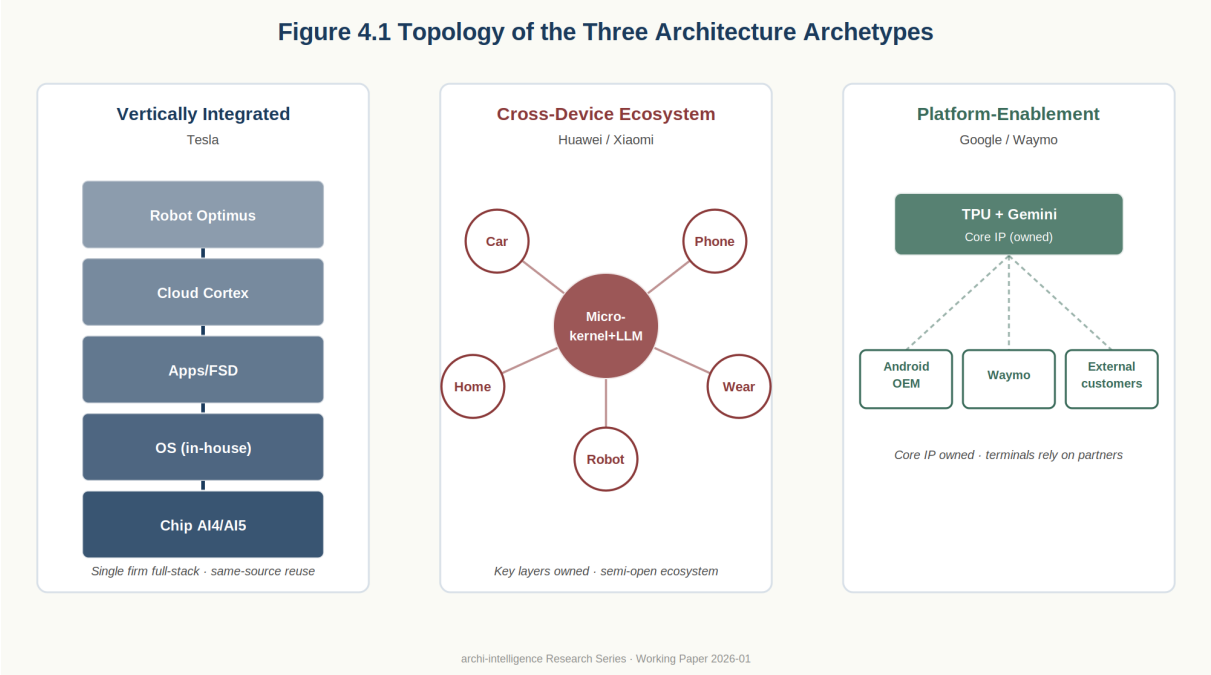

Figure 4: Figure 4.1 Topology comparison of the three archetypes: vertically integrated closed-loop (Tesla), cross-device ecosystem (Huawei/Xiaomi), and platform-enablement (Google/Waymo).

0.6 Chapter 4 In-Depth Benchmark of the Three Architectural Archetypes

Building on the ecosystem taxonomy of Chapter 3, this chapter conducts an in-depth case analysis of the three architectural archetypes. Each case unfolds along four layers: architectural philosophy—reuse mechanism—points of divergence—constraints.

0.6.1 4.1 Vertically integrated closed-loop: Tesla' s embodiment-agnostic perception stack

Figure 4.1 Topology comparison of the three archetypes.

Architectural philosophy: one AI chip + one end-to-end neural network + one embodiment-agnostic perception stack, spanning the car, the humanoid, the Robotaxi, and energy.

4.1.1 Hardware reuse: the same-source chip from FSD to Optimus Optimus' s compute brain (Bot Brain) directly reuses the mature automotive FSD hardware ecosystem. It currently relies mainly on the low-cost but highly efficient AI4 inference chip, and will migrate to AI5/AI6. This reuse is not a "make-do," but the result of Tesla' s meticulous optimization, when designing the FSD chip, of timing, power, and performance-per-watt—an extreme power-efficiency profile that happens to fit the battery-constrained Optimus robot (whose torso houses a 2.3 kWh battery pack, targeting 8-hour endurance).

Through low-level pipelined compute scheduling and efficient memory access, high-frequency sensor data can be shared across processes at extremely low latency. The AI5 chip was designed

with robotic inference loads explicitly in mind, allowing the same SoC to serve both vehicle and humanoid forms.

4.1.2 Direct migration of visual perception and the neural network Tesla insists on a **pure-vision route**—neither the car nor the robot uses LiDAR. This much-debated architectural decision is the **core technical premise** of FSD → Optimus reuse:

- **8-camera vision configuration**: Optimus reuses almost exactly the same camera combination;
- **Occupancy Networks (disclosed at 2022 AI Day)**: voxelized 3D dense occupancy + occupancy flow field, generating an embodiment-agnostic 3D scene representation from cameras, with **voxel size reducible to 10 cm near the robot**—this is the core technical reason the FSD stack can migrate to Optimus;
- **48 independent neural networks**: a full FSD build contains 48 independent NNs, consuming ~70,000 GPU-hours of training, each timestep outputting 1,000 discrete prediction tensors;
- **FSD V12 (January 2024), the first end-to-end NN**, replacing 300,000+ lines of C++ control code; **V13 (December 2024)** native HW4 resolution, 4.2× data volume; **V14 (2025–26)** parameters 4.5× V13.

These top-tier vision algorithms—used on the 2D road for semantic segmentation, object detection, monocular depth estimation, bird' s-eye view, and occupancy networks—were transferred to Optimus almost verbatim. The robot thus need not reinvent the wheel, obtaining world-class 3D environmental perception and object recognition directly.

4.1.3 The non-reusable kinematics dimension: the engineering granularity of divergence Despite sharing the same "brain" and "eyes," the car and the humanoid robot differ vastly in their "cerebellum" (low-level motion control) and actuator hardware.

The essential difference of the kinematics dimension: autonomous-driving trajectory planning takes place in a low-dimensional space (trajectory planning on the road plane), and the kinematics model is relatively simple; whereas Optimus Gen 3 has **22 hand degrees of freedom + 3 wrist DoF (25 DoF per arm, approaching the human hand' s 27 DoF) + 28 whole-body DoF**, and its walking gait involves dynamic center-of-gravity transfer, bipedal force-feedback balance, and anti-fall impedance control—an entirely new physical-control system independent of driving logic.

The radical design of the mechanical actuators: Optimus is equipped with **six fully self-developed Tesla custom actuators** (three rotary reducers + three linear actuators). By combining brushless motors with a planetary roller screw, a single actuator can burst out

enough thrust to lift a 500 kg grand piano within a mere 2-inch short stroke. **Gen 3 hands have 50 actuators, tendon-driven, with all actuators placed in the forearm, and tactile fingertip sensitivity 4× Gen 2.** In patent WO2024/073138A1, Tesla designed an ingenious under-actuated cable structure for the mechanical hand—using only 6 actuators (two dedicated to the thumb, one each for the other four fingers) in concert with front-mounted cables and torsion springs to achieve efficient, precise control of 11 hand joints, capable of the fine motion of picking up an egg.

4.1.4 Same-source reuse of training infrastructure The large-scale data-annotation pipeline, hardware-in-the-loop simulation system, and fleet-level evaluation engine that Tesla built for FSD are all reused for the robot' s closed-loop training. This infrastructure-level reuse greatly accelerates the path to practical deployment.

Training-compute evolution: Dojo D1 (2021, 7nm, 362 TFLOPS BF16/CFP8) → D2 (TSMC wafer-scale production April 2025) → **Dojo project disbanded August 2025** (Peter Bannon departs, ~20 engineers found DensityAI) → **Cortex cluster** (Austin, ~67,000 H100-equivalent compute) → **Cortex 2.0** (Giga Texas, 250 MW launched April 2026, 500 MW full production mid-2026) → **Dojo 3 restarted from January 2026 in the form of AI5/AI6/AI7 SoC boards.**

4.1.5 Organizational-level reuse: the merger of the FSD and Optimus teams In June 2025, leadership of the Optimus project was transferred from Milan Kovac to Ashok Elluswamy —who concurrently serves as VP of FSD/Autopilot. **This is the official signal of the de facto merger of the FSD and Optimus teams**, and the ultimate expression of the vertically integrated closed-loop architecture at the organizational level.

4.1.6 Constraints The core constraints of the Tesla paradigm are:

1. **Extremely high capital and talent thresholds**—simultaneously integrating silicon, OS, applications, cloud, and robots vertically requires tens of billions of dollars in investment;
2. **The safety controversy of the pure-vision route**—regulators remain reserved about L4 liability claims with a camera-only configuration;
3. **The fragility of the liability model**—once the end-to-end NN exhibits a hard-to-explain failure mode, vertical integration paradoxically makes attribution more difficult;
4. **The cost of a closed ecosystem**—the lack of a third-party developer ecosystem confines application innovation to Tesla' s single R&D cadence.

0.6.2 4.2 Cross-device ecosystem: the two Chinese paths of Huawei and Xiaomi

The defining characteristic of the cross-device ecosystem type is to enter from smartphones and smart homes with vast user bases, pool the battle-hardened underlying operating system and algorithms, and then "dimensionally empower" the automotive and embodied-robot domains at extremely low marginal cost. Huawei and Xiaomi represent two parallel paths of this archetype in China.

4.2.1 Huawei: a sanction-forced horizontal microkernel + large-model platform **Architectural philosophy**: one HarmonyOS microkernel + one Pangu large model + one Ascend compute substrate, covering phone—PC—vehicle—cloud—embodied robot.

Hardware: Kirin 9000s (SMIC N+2, 2023) → 9020 (2024) → **9030** (SMIC N+3, China's first 5nm-class production silicon, confirmed by TechInsights in 2025); Ascend 910B (376 TFLOPS BF16) → 910C (dual-die 780 TFLOPS); MDC 610/810/1000; **CloudMatrix 384 supernode**—384 Ascend 910C chips fully interconnected via 6,912 400G OSFP SiPh LPO optical modules, ~300 PFLOPS BF16 (1.67× NVIDIA GB200 NVL72), HBM 49.2 TB (3.6×), power 559 kW (3.9×). **"Trading optical interconnect + cluster scale for a single-chip generation gap" is Huawei's core tactic against the compute blockade.**

Software: HarmonyOS microkernel + Ark Compiler/Runtime + ArkTS + Cangjie language + BiSheng NDK. **HarmonyOS NEXT (5) commercialized October 22, 2024, fully stripped of AOSP and the Linux kernel; HarmonyOS 6 released November 25, 2025 with the Mate 80.** As of November 2025, HarmonyOS 5/6 installed on 27M+ devices, with **300,000+ native apps and meta-services** and 10M+ registered developers. Its core architectural weapons are the **distributed task-scheduling system** and the **Super Device abstraction layer**, allowing phones, watches, smart displays, and head units to share each other's compute, cameras, and sensor data in real time.

Data/AI: Pangu 5.0 → 5.5 (**718B-parameter deep-thinking model** + five vertical-industry sub-models, able to complete complex logical reasoning of more than 10 steps within 5 minutes); the **Pangu world model** generates 4D BEV video + LiDAR point clouds for driving and embodied-AI training.

The Kuafu humanoid robot: hardware-supported by Leju Robotics, first demonstrated at the 2024 Huawei Developer Conference (HDC). The architectural ingenuity lies in an extreme "cloud-edge-device" coupling:

- The bottom layer runs the OpenHarmony operating system;
- Its "soul" is fully connected to the **Pangu Embodied Intelligent Large Model** in Huawei's cloud;
- Pangu's powerful natural-language understanding, task planning, and generation are

reused to guide dual-arm collaborative tasks (sweeping, cooking);

- Pangu automatically generates operation tutorials and training videos for the robot to learn quickly, **greatly shortening the skill-generalization cycle**;
- The **low-latency characteristics of 6G networks** are planned for integration into the next-generation robot architecture, to break through the physical bottleneck of single-machine compute.

Full-stack ecosystem: three business models—pure components (Kirin/Ascend/MDC) + HI (Huawei Inside; Avatr, Audi Q6L e-tron) + **HIMA (Harmony Intelligent Mobility Alliance)** (AITO/Luxeed/Stelato/Maextro). **Qiankun ADAS surpassed one million installed vehicles in August 2025, 1.4 million by end-2025, targeting 3 million / 80+ models in 2026.**

4.2.2 Xiaomi: a consumer-brand-driven vertically integrated OEM **Architectural philosophy**: one HyperOS dual-kernel + one HyperConnect protocol + a set of world-best supply-chain silicon, building a "human × car × home" ecosystem.

Hardware: phones use the flagship Snapdragon 8 Elite Gen 5 + the 2025 self-developed **Xring O1** (its first 3nm application processor); the car uses the single-chip **NVIDIA Drive Thor-U 700 TOPS** (standard across the 2026 SU7 and YU7) + the Qualcomm 8295P cockpit SoC; LiDAR Hesai AT128 + 4D millimeter-wave + 11 cameras + 12 USS.

Software—the dual-kernel heterogeneous substrate: **HyperOS 1.0 (October 2023) → 2.0 (2024) → 3.0 (2025–26).** The bottom layer uses a **dual-kernel architecture**—Linux (phone/tablet/car) + **Vela RTOS** (Xiaomi' s own, derived from NuttX, empowering IoT since 2017, supporting 200+ processors / 20+ filesystems). The framework layer fuses AOSP + Vela + 8 new subsystems (including an AI subsystem). The **HyperConnect cross-device protocol** and **HyperOS Cabin** (Android 12 + Linux 5.4 kernel) support cross-device coordination.

The reuse dividend of CyberDog and CyberOne:

- **CyberDog (bionic quadruped, 2021; 2.0 in 2023)**: carries the NVIDIA Jetson Xavier NX edge supercomputer (with **384 CUDA cores + 48 Tensor cores**), processing data from 11 high-precision sensors;
- **Direct migration of vision algorithms**: through HyperOS' s unified interfaces, the deep imaging technology accumulated by Xiaomi' s phone division (AI interactive camera system, ultra-wide fisheye distortion correction, optical algorithms sedimented from the Leica partnership) directly empowers the robot, granting it top-tier environmental perception and depth modeling;
- **CyberOne (full-size bipedal humanoid, August 2022)**: 177 cm / 52 kg / 21 DoF, with **peak joint torque up to 300 Nm**. To support complex bipedal balance, Xiaomi de-

veloped atop HyperOS a dedicated **impedance-control-based mechanical-dynamics solver engine**—precisely the difference in control dimension between a phone "seeing" and a robot "seeing and generating precise physical reaction forces."

Data/AI: the **MiLM** large-model family + the **MiMo** 7B reasoning model (open-sourced 2025); the **XLA cognitive LLM** integrated into the 2026 SU7; Xiaomi HAD end-to-end NN, covering 100-city NOA in 2024, standard across the lineup in 2026.

Full-stack ecosystem: a vertically integrated OEM—9,100-ton HyperCasting integrated die-casting + self-developed titanium-alloy Titans Metal + self-developed V6/V6s/V6s Plus motors + co-engineered 800V battery + direct-sales stores. **SU7 cumulative 600,000+ units in 22 months on the market; 2026 EV target 550,000 units**; SU7 sold 258,000 units in China in 2025, surpassing the Model 3's 200,000.

4.2.3 Why such different architectural choices?

Dimension	Huawei	Xiaomi	Root cause
Kernel	Pure microkernel (HarmonyOS)	Hybrid (Linux + Vela + AOSP)	Sanction-forced vs. time-to-market priority
Business model	Tier-1 (HIMA)	Vehicle OEM	Ren Zhengfei's policy banning carmaking vs. capital abundance + consumer-brand channel
Silicon strategy	Fully self-developed (HiSilicon) + SMIC	Best-of-global procurement + selective self-development	Sanction blockade vs. time-cost optimum
Model path	Pangu in-house + own training compute	MiLM in-house + selective external procurement	AI-infrastructure strategic sovereignty vs. application priority

Architectural reuse points: Huawei is the industry's **cleanest horizontal cross-domain reuse** exemplar—the same HarmonyOS microkernel and the same Pangu LLM run under phone, PC, head unit, ADS, and the Kuafu robot. ADS WEWA = a driving-specialized version of the Pangu world model; Kuafu = an embodiment-specialized version of Pangu. Xiaomi

adopts **hybrid reuse**: HyperOS + HyperConnect are shared, but the underlying hardware is heterogeneous (phone Snapdragon, car NVIDIA, robot NVIDIA Jetson).

0.6.3 4.3 Platform-enablement: Google' s federated AI platform and Waymo' s provable-safety architecture

The defining characteristic of the platform-enablement type is to **empower others through open platforms, chips, toolchains, and models, without needing to own all terminal forms.** Google is the ultimate representative of this archetype.

4.3.1 Google's architectural philosophy: TPU + Gemini + vertical partners **Hardware**: Pixel 10 (Tensor G5, TSMC 3nm, August 2025, strategically abandoning Samsung foundry); **TPU v7 Ironwood** (GA November 2025, 4,614 TFLOPS FP8 per chip, slightly exceeding NVIDIA B200, 192 GB HBM3e, 9,216-chip superpod = 42.5 EFLOPS FP8, 30× perf/W vs. the first-gen TPU; **Gemini 3 trained entirely on TPU**—a key node in breaking free of NVIDIA dependence); Axion ARM CPU (based on Neoverse V2).

Ecosystem partnerships: Anthropic signed a 1-million-Ironwood order; Apptronik raised USD 520M in a February 2026 Series A extension led by Google + B Capital (USD 5B valuation)—the Apollo humanoid carries Gemini Robotics + Gemini 3, with customers including Mercedes, GXO, Jabil, John Deere.

4.3.2 The architectural refactoring of AAOS: from C++ sinkhole to TaskView layered architecture Android Automotive OS (AAOS) underwent a landmark internal architectural refactoring that profoundly reveals the strategic value of "screen sovereignty" in the SDV era.

Old architecture: the C++ sinkhole pattern. In the Native Era of Android 11/12, the casting system used an extremely rigid design: the phone sends a video stream, and the head unit, via a C++ native service, initializes a layer surface in the hardware abstraction layer (HAL), crudely "punching a hole" in the system UI and connecting the video-buffer pipeline directly to the underlying display hardware controller. The fatal flaw of this architecture is that **the Android WindowManager completely loses awareness of this casting region on the screen.** If the vehicle suddenly issues a battery alert or active-braking prompt, the system cannot safely overlay a notification at the top layer of the screen, causing UI conflicts and the risk of a black-screen crash.

New architecture: Layered Architecture + Concurrent Multi-User. Starting with AAOS 14, the old low-level C++ video-stream pipeline was completely abandoned, replaced by a layered architecture and a concurrent multi-user framework. The casting application is elevated to the Java/Kotlin application layer. When the phone connects, the **CarProjectionManager**

acts as the dispatcher, spinning up a virtual container called **TaskView** within the head unit, running the cast as a native standalone Android task.

The deeper strategic significance of the refactoring:

1. **The OEM regains full sovereignty over "every pixel"of the screen**—free to confine the casting window to a specific region and overlay native climate control or safety-alert UI above it;
2. **In concert with the DisplayManager mechanism**, passengers can freely move virtual-machine tasks among multiple in-vehicle screens, connecting multi-terminal ecosystem coordination;
3. **A standard framework layer between the phone and the vehicle hardware** provides the OEM with a robust checkpoint to strictly "gate" the permissions for the vehicle's private sensor data flowing back into the phone ecosystem.

This case reveals the subtle power game between platform companies (Google) and terminal brands (OEMs)—a subtle design choice at the architectural level determines who in the value chain holds "screen sovereignty."

4.3.3 Waymo's "provable-safety" AI architecture Waymo's ultimate belief about architectural design is: **in autonomous driving, safety must never be an "after-the-fact remedial layer" of functionality, but must be embedded into the genetic bones of the architecture from its embryonic stage.**

The Waymo Foundation Model ecosystem: the base is the grand Waymo Foundation Model, atop which three core functional components ride—

- **Driver**: actually executes driving decisions;
- **Simulator**: generates training and test scenarios;
- **Critic**: rigorously scores and challenges the Driver's decisions.

The three components grow in parallel on the same underlying AI architecture, forming a continuously self-reinforcing closed loop.

The System 1 / System 2 dual-track hybrid architecture:

- **Sensor-fusion encoder (System 1, fast thinking)**: fuses the massive real-time data from LiDAR, cameras, and radar at extremely low latency, directly outputting the motion attributes and semantic information of surrounding objects via neural embeddings, satisfying instinctive collision avoidance and rapid reaction in the vast majority of situations;
- **Driving Vision-Language Model (Driving VLM, System 2, slow thinking)**: deeply fused with the Gemini large language model, dedicated to slow thinking. **When**

the vehicle encounters an extremely rare long-tail scenario, even one beyond human common sense (a burning tanker truck ahead, a parade in bizarre costumes), System 2 is activated, using the broad physical common sense embedded in the language model for deep logical reasoning, thereby commanding the vehicle to execute advanced semantic hazard-avoidance behavior beyond pure geometric obstacle-avoidance rules (immediately turning around, planning a detour route dozens of kilometers away).

The Teacher-Student knowledge-distillation mechanism: to run this behemoth in real time on the resource-constrained vehicle compute platform—

- **Teacher Driver / Teacher Critic** conduct slow but deliberate gaming and simulation reasoning in the cloud compute center;
- High-quality response strategies and implicit world-cognition knowledge are **compressed and distilled** into an extremely lightweight **Student Driver** small model;
- The Student completes low-latency real-time driving inference on the physical vehicle' s chip;
- Before execution, a **dedicated independent in-vehicle verification layer** performs a final physical interception of the legality of the generated trajectory.

Dual-loop reinforcement learning:

- **Inner Loop**: the Simulator and Critic subject the Driver to brutal trials at million-fold speed in virtual space;
- **Outer Loop**: real road-test data captures the Driver' s suboptimal performance, finds improvement countermeasures in the cloud, and delivers updates again.

Structured world representation: unlike a pure black-box end-to-end network, the Waymo Foundation Model synchronously generates a structured world representation (traffic-participant objects, explicit semantic attributes, road-topology graph), which provides a critical visualization and explicability handle for the **safety-verification criteria** at inference time.

Waymo' s 6th-generation Driver (announced August 2024, fully autonomous on the road February 12, 2026): sensor count reduced 42% vs. the 5th gen, BOM <USD 20,000 (cost down >50%); snow validation completed (Boston, Pittsburgh, Denver); cross-platform deployment—the Ojai dedicated vehicle co-developed with Zeekr + Hyundai IONIQ 5 (**50,000-unit order, the largest single AV order in history**).

Operational metrics (early 2026): 15 million paid rides in 2025 (4× 2024); 20 million+ cumulative; ~400,000 rides/week; fleet of ~2,500 vehicles; 200 million miles of fully autonomous driving cumulative; operating in Phoenix, SF, LA, Austin, Miami; expanding to Washington DC, San Diego, Tokyo (testing), London (public 2026), New York (testing).

4.3.4 Waymo vs. Tesla: the fundamental opposition of two autonomous-driving architectural philosophies

Dimension	Waymo	Tesla
Sensors	LiDAR + camera + radar + audio (multi-redundant)	Pure vision
HD maps	Required at city scale	Not relied upon
ODD	Geofenced L4	Unfenced L2++ → Cybercab L4
Liability	Waymo (vehicle owner)	Driver (consumer car) / Tesla (Cybercab)
BOM	6th gen <USD 20,000	USD 1,000–2,000 (sensors only)
Software	Sensor fusion + world model + System 1/2	End-to-end NN
Expansion	City-by-city HD mapping	OTA global overnight deployment
Philosophy	"Safer than humans today" —hardware redundancy first	"Data + scale will converge" —software first

Will they converge? Unlikely. Waymo's 6th gen cutting sensors by 42% shows that ML maturity allows reduced redundancy; Tesla's validation fleet was spotted carrying LiDAR in Phoenix (for ground-truth annotation, not consumer deployment). But Waymo's geofence + L4 liability model fundamentally requires redundant-sensor certification of the "no-driver" claim; Tesla's economic model (consumer car + Optimus) cannot bear LiDAR + HD maps and must bridge the gap through end-to-end ML at scale. **The critical points at which the two paradigms each converge most likely fall in 2027–2029—at which time, whether end-to-end VLA can provide provable safety in a camera-only configuration will determine the future industrial landscape.**

0.6.4 4.4 Other sub-types of the platform-enablement type

The platform-enablement type is further subdivided into four sub-types, each corresponding to a different "key-bottleneck" control strategy:

Sub-type	Representative	Core enabling asset
Open OS platform	Google (AAOS)	Android/AAOS/Wear OS ecosystem, Treble HAL, Mainline modular updates
Autonomous-driving platform	Mobileye	EyeQ heterogeneous SoC, REM crowdsourced maps, RSS formal safety model, True Redundancy
Compute + simulation platform	NVIDIA	DRIVE + Hyperion + Halos; Isaac + Cosmos + GR00T; the train-simulate-deploy "three computers" paradigm
Production-engineering platform	Bosch, Aptiv	Central vehicle compute + zonal control (Aptiv SVA' s PDC + Open Server Platform), full-stack production engineering

4.4.1 Mobileye: an autonomous-driving platform centered on formal safety Mobileye represents a distinctive architectural philosophy: **not pursuing ownership of a terminal brand, but becoming the "de facto standard of safety methodology" in autonomous driving through deep technical specialization.** Its core enabling asset is a trinity—

- **The EyeQ heterogeneous SoC roadmap**: EyeQ4 (2.5 TOPS, 2018) → EyeQ5 (24 TOPS, 2021) → **EyeQ6 Lite/High** (production late 2024, single-chip support for full ADAS up to L2++) → **EyeQ Ultra** (12 dual-thread RISC-V cores + 256 GFLOPS Arm GPU + 64 proprietary accelerator cores, <100W TDP, targeting single-chip L4, production 2025–26). The EyeQ series' differentiation lies not in peak compute but in **extreme optimization of performance-per-watt**—a metric critical for production embedded systems.
- **REM (Road Experience Management) crowdsourced HD maps**: through real-time visual-data feedback from millions of production vehicles, builds and continuously updates centimeter-level HD maps. As of 2025, REM covers over 25 billion km of data, one of the largest crowdsourced driving datasets in the world.
- **RSS (Responsibility-Sensitive Safety) formal safety model**: in a seminal 2017 paper, Mobileye proposed RSS—a formal driving-decision framework based on mathematical axioms (safe longitudinal distance, safe lateral distance, reasonable assumptions, cautious

collision avoidance, responsibility allocation). RSS is not a black-box NN but a **provable safety envelope.** As end-to-end neural networks increasingly become the industry mainstream, RSS is more and more often combined with black-box models as a "runtime safety monitor" —echoing Waymo's "independent in-vehicle verification layer."

Mobileye also extends this platform capability to non-automotive domains. In 2025 it announced the extension of the EyeQ chip series and the perception/planning stack to humanoid robots—a move highly analogous to Tesla's "FSD stack empowering Optimus," but Mobileye takes the horizontal path of **empowering other robot-body manufacturers** rather than vertical integration.

The intrinsic tension of the architectural philosophy: Mobileye's strength lies in extreme specialization and provable safety in the autonomous-driving domain, but its weakness is rooted there too—**unlike Android/Tesla, it lacks a large consumer-side OS ecosystem.** This caps Mobileye's value at the role of "being integrated" —it cannot directly own end-user relationships and must reach consumers indirectly through OEM customers.

4.4.2 NVIDIA: the train-simulate-deploy "three computers" paradigm NVIDIA is the most ambitious player in the platform-enablement type. Its core architectural judgment is: **a unified compute and development stack from cloud to vehicle/robot is the largest enabling opportunity of this revolution.** This judgment translates into the "Three Computers" paradigm:

- **Training computer**: data-center GPUs (H100, B200, GB200 NVL72) + DGX/HGX clusters + CUDA + cuDNN + Megatron + NeMo training stack;
- **Simulation computer**: Omniverse + Isaac Sim + the **Cosmos world model** (adopted simultaneously by Toyota DriveOS, 1X, Skild AI, Figure AI, Agility, Uber, Waabi, Foretellix, XPeng, Galbot, Fourier);
- **Runtime computer**: DRIVE Thor (in-vehicle, 2,000 TFLOPS FP8) + Jetson Thor (robot, 2,070 TFLOPS FP4) + DRIVE OS + Isaac ROS.

The unified development stack spanning all three compute stages is NVIDIA's core moat —a developer can train and validate a VLA model in Omniverse, then seamlessly deploy it to Drive Thor or Jetson Thor with **minimal code and toolchain changes.** This "develop once, deploy across embodiments" capability is currently hard for other platforms to match.

NVIDIA further folds the methodological framework and foundation models into its enabling matrix via **Halos (autonomous-driving safety platform)** and **GR00T (robot foundation model).** Its business model sells neither complete vehicles nor complete machines—only platforms, chips, toolchains, and models—but amplifies value by **empowering the entire industry chain.**

The intrinsic tension of the architectural philosophy: the NVIDIA platform is extremely strong, but **automakers/robot companies must still complete product definition and the liability closed loop themselves.** NVIDIA is not directly responsible for the safety claims, regulatory compliance, or user experience of the terminal product—this is both its advantage (asset-light, high-margin) and its ceiling (always "being integrated" rather than "the integrator").

4.4.3 Bosch, Aptiv: production-engineering platform Tier-1s Bosch and Aptiv represent the most "traditional" yet most production-engineering-deep branch of the platform-enablement type. They started as Tier-1s, but in the SDV era actively reshape their positioning —**from "selling individual ECUs" to "selling central vehicle compute + zonal control + full-stack engineering services."**

Bosch's transformation path: evolving from a classic Tier-1 toward a "software-defined mobility company." Its differentiated asset is full-stack integration capability across body, chassis, sensors, powertrain, software, and AI. In the direction of central vehicle compute and zonal control, Bosch bets on the **cross-domain control** paradigm—one HPC + multiple ZCUs combined with Bosch's full-stack production-engineering capability, covering key safety functions from ADAS to chassis control. The theme of the AUTOSAR 2026 conference shifted toward "the combination of open-source, community-source, and commercial automotive software" — a signal behind which lies the strategic embrace of open ecosystems by leading Tier-1s such as Bosch.

Aptiv's Smart Vehicle Architecture (SVA): Aptiv's SVA is the most representative current Tier-1 zonal-architecture solution. Its core is a two-layer combination of the **PDC (Power Data Center)** zone controller + the **OSP (Open Server Platform).** The PDC handles local I/O, power distribution, and low-speed communication; the OSP serves as a high-compute central node with swappable hardware and abstracted software, allowing different vehicle models to flexibly configure compute gradients as needed. Aptiv's core strategic intent is to **extend vehicle lifecycle and smooth compute upgrades**—that is, through architectural-level decoupling, allowing a vehicle to swap its central compute unit over a 10–15-year lifecycle without redoing the whole-vehicle harness.

The intrinsic tension of the architectural philosophy: production-engineering platform companies like Bosch and Aptiv have extremely strong platform capability, but **the brand touchpoint and data closed loop are usually held by the OEM.** Final product definition still requires the OEM and upper-layer software partners to complete jointly. This position of "value in the middle, control at the two ends" is the deepest strategic anxiety of traditional Tier-1s in the SDV era.

4.4.4 Commonalities and differences of the four platform-enablement sub-types The commonality of these four platform-enablement companies is that **they do not pursue ownership of a terminal brand, but maximize value by empowering terminal brands.** The difference lies in the "key bottleneck" each commands:

- **Google (open OS platform)** commands the **consumer operating system and developer ecosystem**—the entry-control right on the consumer side;
- **Mobileye (autonomous-driving platform)** commands **autonomous-driving algorithms and provable-safety methodology**—the certification-control right on the liability side;
- **NVIDIA (compute + simulation platform)** commands the **unified compute substrate and cross-embodiment development stack**—the productivity-control right on the developer side;
- **Bosch, Aptiv (production-engineering platform)** command **large-scale production engineering and cross-domain system integration**—the engineering-control right on the supply-chain side.

This genealogy reveals a deep regularity: **in complex-systems industries, different "control points" along the value chain can be simultaneously held by different parties without needing to merge.** We will systematize this "architectural control point" concept in Chapter 6.

0.6.5 4.5 The horizontal comparison matrix of the three archetypes

Figure 4.6 Five-dimensional architectural-capability radar of representative OEMs (Snapshot 2026.1.31, each dimension out of 5).

The first three sections of Chapter 4 separately dissected the philosophical foundations, reuse mechanisms, and constraints of the three architectural archetypes—vertically integrated closed-loop (Tesla), cross-device ecosystem (Huawei and Xiaomi), and platform-enablement (Google and Waymo). This section places these three archetypes in a single matrix for a systematic horizontal comparison. The purpose is not to judge which is superior, but to reveal the resource-endowment assumptions, organizational-capability path-dependence, and regulatory-environment fit behind each archetype, and the feasibility boundaries derived therefrom.

4.5.1 Design principles of the comparison matrix Cross-archetype comparison must avoid two common traps. The first trap is **oversimplified induction**, reducing all archetypes to a single dimension (such as "open vs. closed" or "software-first vs. hardware-first"), which obscures the multidimensional coupling of actual architectural decisions. The second trap is **value preconception**, defaulting to treating a certain archetype as "advanced" or "backward,"

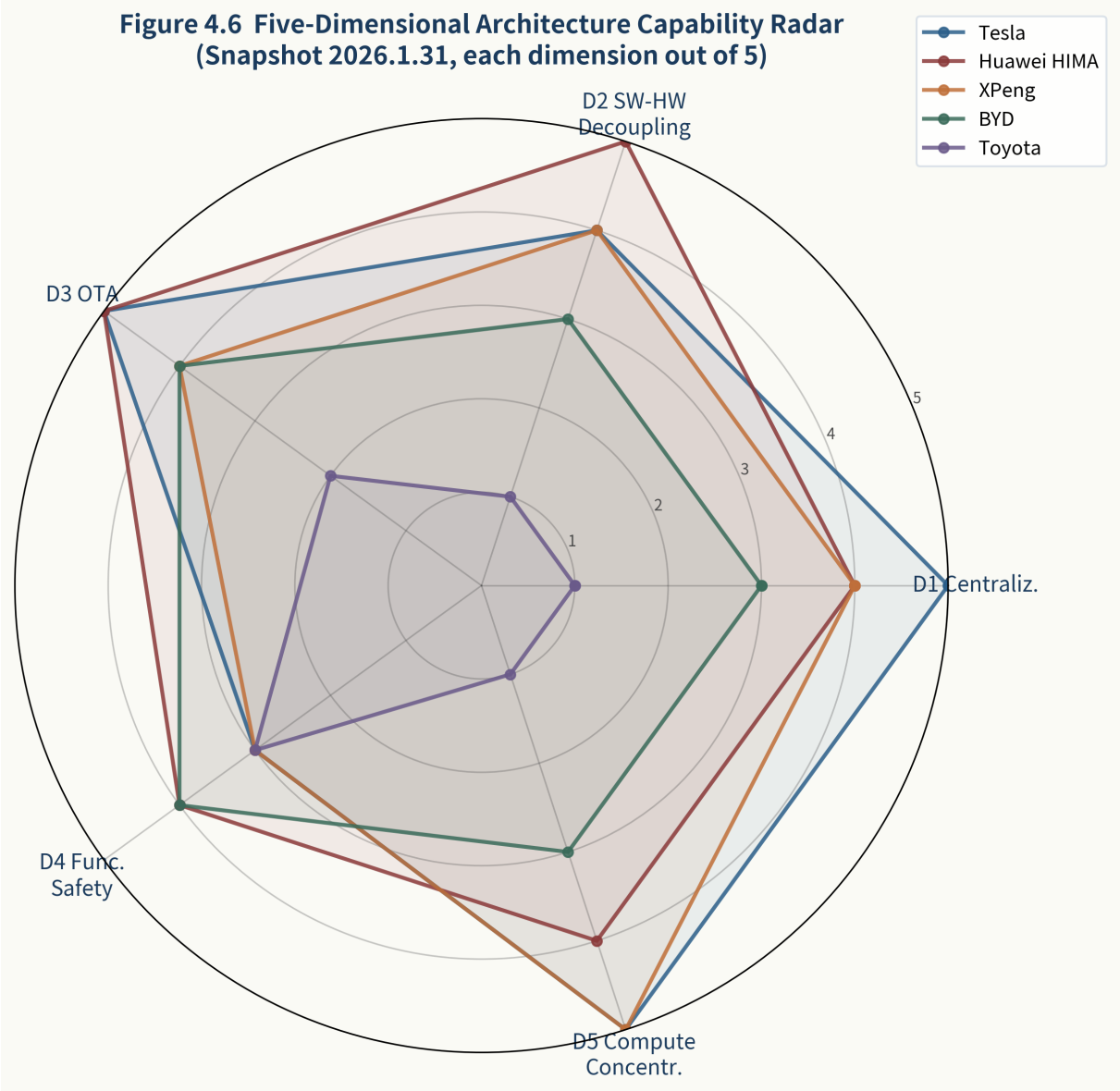

Figure 5: Figure 4.6 Five-dimensional architectural-capability radar of representative OEMs (Snapshot 2026.1.31, each dimension scored out of 5).

which ignores each archetype' s specific rationality under different markets, regulations, and resource conditions.

The comparison matrix in this section is based on five independent dimensions—depth of architectural control, reuse mechanism, ecosystem boundary, regulatory fit, and expansion bottleneck—giving an objective description of each archetype on each dimension, without high-low ranking.

4.5.2 The five-dimensional comparison matrix

Table 4.5: Horizontal comparison matrix of the three architectural archetypes

Dimension	Vertically integrated closed-loop (Tesla)	Cross-device ecosystem (Huawei / Xiaomi)	Platform-enablement (Google / Waymo)
Depth of architectural control	Full-stack ownership—silicon, OS, applications, cloud, and robots all self-developed	Key layers self-owned + peripheral cooperation—kernel OS, foundation model, AI chip self-developed; vehicle manufacturing optionally self-owned (Xiaomi) or partnered (Huawei HIMA)	Key IP self-owned + terminal cooperation—TPU, Gemini, Android, AAOS self-owned; terminal forms rely on partners
Reuse mechanism	Same-source chip + same-source perception stack + same-source training infrastructure + team merger	Same-source microkernel + same-source large model + same-source compute substrate; hardware may be heterogeneous (Xiaomi route)	Same-source chip substrate + same-source LLM infrastructure; terminal architectures highly heterogeneous

Dimension	Vertically integrated closed-loop (Tesla)	Cross-device ecosystem (Huawei / Xiaomi)	Platform-enablement (Google / Waymo)
Ecosystem boundary	Closed—no third-party app ecosystem; application innovation confined to a single R&D cadence	Semi-open—HarmonyOS developer ecosystem 10M+; HyperOS around Xiaomi's hardware ecosystem	Highly open—Android global ecosystem + TPU external customers (Anthropic, Apptronik, etc.)
Regulatory fit	High risk—the pure-vision route still faces regulatory doubt over L4 liability claims; explicability challenge of end-to-end NN	Moderate—excellent fit with China's regulatory environment; data-sovereignty challenges in overseas markets	Excellent—Waymo holds the most complete functional-safety engineering methodology; Google is battle-tested in GDPR/PIPL compliance
Expansion bottleneck	Capital and talent—simultaneously integrating all layers vertically requires tens of billions of dollars; any single-layer failure cascades through the full stack	Politics and geopolitics—Huawei forced by chip sanctions; Xiaomi constrained by global supply chain and market access	Missing terminal control point—Google does not directly own automotive and robot terminals, relying on partners' execution
Typical capital scale	Tens-of-billions-of-dollars level (only Tesla, with a multi-trillion market cap, supports this strategy)	Tens-to-hundreds-of-billions-of-RMB level (Huawei R&D investment and Xiaomi ecosystem investment)	Tens-of-billions-of-dollars level, but spread across multiple product lines

Dimension	Vertically integrated closed-loop (Tesla)	Cross-device ecosystem (Huawei / Xiaomi)	Platform-enablement (Google / Waymo)
Typical time window	Long-termism—the FSD project has been evolving for 12 years since 2014	Medium-long—HarmonyOS 7 years since 2019; Xiaomi's carmaking from 2021 announcement to SU7 production in just 3 years	Long—Waymo, 17 years since Google's self-driving project in 2009
Failure-mode loss	Full-stack failure—any layer's collapse may drag down the whole system; hard to stop loss locally	Modular failure—failure of one form is not fatal; can focus on other forms	Control-point failure—if TPU or Gemini loses competitiveness, the platform position collapses; terminal-layer failure is not directly fatal
Representative success metric	FSD production scale + proof of Optimus cross-embodiment reuse	HarmonyOS installed base + Huawei HIMA installed vehicles + Xiaomi SU7 sales	TPU external-customer count + Android global share + Waymo commercial Robotaxi mileage
Typical limiting case	Pure-vision safety controversy, "vanity project" drain (Cybertruck etc.), CEO personal risk	Huawei under chip-generation blockade; Xiaomi's first-model quality and brand challenges	Apple terminal absence (Google's repeated failures entering consumer hardware), partner-execution variance (Pixel vs. Samsung ecosystem friction)

4.5.3 Non-substitutability and selective borrowing A key insight of the comparison table above is: **these three archetypes are not simply substitutable for one another, yet specific practices within them can be selectively borrowed.**

The source of non-substitutability is the **fundamental difference in resource endowment and organizational DNA.** A traditional OEM cannot "become Tesla" in 18 months, because the capital scale, talent density, and organizational flatness that the vertical closed loop demands are the product of 20 years of accumulation; a US university spin-off cannot "become Huawei," because the national-level R&D coordination capability and long-term endurance that a horizontal microkernel + large-model platform demands are products of another civilizational tradition; a Chinese internet company cannot "become Google," because the global open-source governance experience, cross-jurisdictional compliance capability, and long-term trust relationships with hardware vendors that a federated AI platform demands cannot be built in the short term.

The source of selective borrowing is the **local replicability of specific practices and decision patterns.** Other players can borrow Tesla's "embodiment-agnostic perception stack" idea, reusing their own vision or control algorithms across embodiments; can borrow Huawei's "horizontal microkernel + large-model platform" idea, doing a small-scale implementation in a specific niche (such as car-phone coordination); can borrow Google's "open platform + top-tier IP" idea, opening their core technology to ecosystem partners to form a local "mini-ecosystem."

Specifically, this study observes the following three "intermediate-path" practices:

Path one: localized vertical integration. Some OEMs choose to vertically integrate in certain key layers (such as OS, battery management, ADAS) while continuing to procure or partner in others (such as cockpit, infotainment). Rivian's extreme simplification of central compute, BYD's vertical integration in battery management, and NIO's investment in self-developed chips (Shenji NX9031) all reflect this path.

Path two: narrow-domain cross-device ecosystem. Some non-super-ecosystem players choose to ecosystemize only on specific device-to-device coordination, rather than full-stack cross-device. Examples include the Polestar-Volvo cooperation on car-phone coordination, and Mercedes' attempt to unify cockpit and ADAS within MB.OS.

Path three: reverse platform-enablement. Some OEMs choose to "reverse-empower"—not playing platform host, but not platform consumer either, instead licensing their core capabilities (such as BMW's advanced ADAS, Toyota's hybrid control system) to other players, forming a local "mini-ecosystem."

4.5.4 A decision framework for choosing an archetype For industry readers, this section offers a simplified decision framework to help judge which archetype is most suitable to reference. The framework comprises four judgment conditions:

Condition one: capital and talent reachability. The vertical closed loop requires cumulative investment of USD 10 billion+ and a top-tier engineering team of 5,000+; the cross-device ecosystem requires at least one super-terminal entry (such as a phone or smart home)

and R&D investment at the USD 10 billion level; platform-enablement requires global-level open-source governance capability and a technology-IP reserve of USD 20 billion+.

Condition two: regulatory-fit controllability. The vertically integrated type carries higher risk in the highly mature regulatory markets of Europe and the US, and moderate risk in large single markets such as China and the US; the cross-device ecosystem type carries lower risk in the China market and faces data-sovereignty challenges in Europe and the US; the platform-enablement type adapts to almost all regulatory environments, at the cost of lower single-point margins.

Condition three: organizational-DNA compatibility. The vertical closed loop demands a flat, fast, fault-tolerant Silicon Valley startup culture; the cross-device ecosystem demands ultra-long-termism, strong central coordination, and the ability to endure technology blockades; platform-enablement demands mature open governance, cross-cultural coordination, and long-term trust with hardware partners.

Condition four: tolerance for failure. The vertical closed loop' s failure may be fatal because of full-stack coupling; the cross-device ecosystem' s failure can be locally contained; platform-enablement' s failure is confined to a single control point. This dimension is especially important for resource-limited players—choosing an archetype whose failure cost is bearable matters more than chasing the maximum valuation ceiling.

It must be emphasized that this decision framework cannot replace specific strategic judgment. Each company' s actual choice is shaped by the combined influence of its historical path, existing assets, leadership preferences, and market window. The purpose of this study is to provide analytical tools, not to give specific decisions.

4.5.5 Synergy and competition among the three archetypes Finally, it is worth noting that the three archetypes are not only in competition but also have significant room for synergy.

Tesla' s open nodes include accepting the NVIDIA Cosmos world model as a training supplement, cooperating with certain cloud vendors for training-data backup, and opening the FSD dataset (albeit at limited scale) to some research institutions. This means that even the most closed vertically integrated type cannot achieve complete zero external dependence.

Huawei' s open nodes include OpenHarmony open-sourcing (distinct from the fully self-developed HarmonyOS NEXT), the multi-OEM sharing of the HIMA brand within the alliance, and the external opening of the Pangu large model to government and enterprise customers. This means that even the most deeply vertically integrated cross-device ecosystem type retains considerable room for external enablement.

Google' s closed nodes include the internal-priority use of TPU (Gemini 3 trained entirely on TPU), the internal coordination of Pixel and AAOS, and the Waymo business model' s restrictions on external data cooperation. This means that even the most open platform-

enablement type has non-shareable core control points.

This "both closed and open" hybrid mode reflects the true form of contemporary complex-systems architecture—neither pure closure nor pure openness is sustainable; the winning players are all continuously optimizing the fine judgment of "what to open, what to close, to whom to open, and when to open." Chapter 6 of this study will further systematize this observation, introducing the "seven architectural control points" as an analytical tool.

0.7 Chapter 5 A Unified Maturity Framework: AR0—AR5 and AI²-ML

0.7.1 5.1 Inspiration and methodology

The success of SAE J3016' s autonomous-driving levels (L0—L5) lies in providing **a common language across enterprises, technology routes, and regulatory regions.** Any discussion of autonomous-driving capability can be quickly anchored to L2/L3/L4/L5. This "leveling language" is equally necessary for architectural evolution—but the industry currently lacks a comparable unified framework.

The robotics industry has attempted similar leveling (EarthSense' s Hands off / Eyes off / Mind off / Monitoring off / Development off), but it was confined to field/agricultural robots and never formed an industry-wide consensus. In the automotive E/E direction, Bosch' s six-stage framework is the most influential, but it is essentially an **evolution path within a single industry** and was never abstracted to a level usable across domains.

This study proposes the **AR0—AR5 architectural capability-threshold framework** (AR = Architecture Readiness), attempting to unify, at a higher level of abstraction, the architectural evolution of three major domains: automotive E/E, robotics, and intelligent terminals broadly. **An important methodological principle: AR levels characterize capability thresholds, not a timeline; lower tiers do not disappear as higher tiers emerge, but coexist with them over the long term.** This is consistent with the spirit of SAE J3016—L0/L1/L2/L3 vehicles will coexist on the road for decades.

0.7.2 5.2 The AR0—AR5 architectural capability-threshold framework

AR0 —Mechanical-Electrical Island **Core characteristics**: one controller per function, deep hardware-software coupling, software tied to hardware, almost no remote-update capability.

Key indicators: high ECU/controller count, weak cross-domain coordination, narrow or nonexistent OTA scope.

Typical state: traditional distributed ICE vehicles, monolithic industrial-robot controllers, the discrete baseband/AP/ISP/NPU era of early smartphones.

Architectural meaning: this stage belongs to the "digital stone age." The system lacks global perception of the external physical world; it remains essentially a collection of actuators, without the evolvable-platform properties of modern systems.

AR1 —Domain-Level Integration **Core characteristics**: integration of domains such as cockpit/body/intelligent-driving; AUTOSAR Classic/Adaptive coexist, or Adaptive is partially adopted within domains; OTA begins to scale but is still per-ECU.

Key indicators: 5±2 domain controllers, cross-domain communication still limited by physical bus bandwidth.

Typical state: most 2024–2026 production SDV transitional states, collaborative robotic arms, humanoid-robot demos requiring remote intervention.

Architectural meaning: an SOA prototype and basic middleware appear. But information flow between domains is still strictly cut off, and the vehicle or system cannot produce emergent joint intelligent decisions.

AR2 —Zonal Platform **Core characteristics**: zonal control + central compute, service-oriented diagnostics, virtualization, and a standard signal catalog.

Key indicators: controller count drops markedly (typically 7–15), the whole-vehicle software is platformized, cross-model/cross-product-line reuse strengthens.

Typical state: high-end EVs, next-generation SDV platforms, Boston Dynamics Spot commercial product, Agility Digit.

Architectural meaning: **the Hypervisor virtualization layer lets a consumer-grade OS and a hard-real-time microkernel safely coexist on the same heterogeneous SoC**—the most pivotal technical breakthrough in AR evolution.

AR3 —Cross-Device Collaborative Agent **Core characteristics**: car/phone/wearable/home/robot share identity, context, and task migration.

Key indicators: cross-device task continuity, unified account and policy, edge-cloud coordinated scheduling, cross-embodiment compute pooling.

Typical state: the Huawei/Xiaomi ecosystem direction; Google/Android is also extending toward in/out-of-vehicle linkage.

Architectural meaning: the architectural philosophy shifts from "optimizing a single device" to "optimizing user identity and task flow." The device itself becomes a replaceable combination of compute + sensors, and the software ecosystem leaps from device level to ecosystem level.

AR4 —Multi-Embodiment Physical AI Platform **Core characteristics**: training, simulation, and deployment unified in a closed loop; the world model and foundation model shared across car/robot.

Key indicators: simulation-and-real-data closed-loop efficiency, model reuse rate, policy-transfer efficiency, cross-embodiment generalization.

Typical state: Waymo Foundation Model, NVIDIA Cosmos + GR00T, Tesla AI5 + FSD V14 + Optimus same-source stack, Mobileye Robotics extension.

Architectural meaning: architecture thoroughly breaks the shackles of the underlying hardware. A distributed-microkernel OS + an end-to-end foundation model achieve dynamic pooling and sharing of compute across smartphones, wearables, head units, and robots. The sys-

tem broadly introduces a dual-track mechanism of System 1 (instantaneous intuitive perception) and System 2 (large-model logical reasoning).

AR5 —Trusted General Embodied Agent **Core characteristics**: a general policy layer can execute across multiple physical entities, continuously proving safety/compliance/liability.

Key indicators: not merely "able to do," but "provably able to do."

Typical state: no mature production paradigm yet; at the research and prototype stage.

Architectural meaning: the system crosses the threshold of the "silicon-based life form." **With only one underlying multi-core dispatching microkernel OS + one foundation model that comprehends the laws of world physics, it can simultaneously drive the user' s wearable holographic device, an unmanned physical shuttle in a blizzard-stricken mountain region, and a bipedal humanoid robot stir-frying in the kitchen.** The compute environment is thoroughly decentralized and ubiquitous. The physical-isolation barriers across devices and industries collapse entirely.

0.7.3 5.3 Visualizing the AR levels

AR0	AR1	AR2	AR3	AR4	AR5
Mechanical →	Domain- →	Zonal →	Cross- →	Multi- →	Trusted
-Electrical	Level	Platform	Device	Embodiment	General
Island	Integration		Collab.	Physical AI	Agent
			Agent	Platform	

It must be emphasized again: **these are capability thresholds, not a timeline.** In the 2030s, AR0–AR2 will still exist long-term as the mainstream form of traditional vehicle models and industrial control systems; AR3–AR4 will become the competitive frontier of high-end intelligent terminals; AR5 remains at the research and prototype stage.

0.7.4 5.4 Introducing Architecture Intelligence Maturity Levels (AI²-ML)

The AR framework characterizes the architectural maturity of the **designed system** (vehicles, robots, intelligent terminals). But architecture itself, as an engineering activity, also has a maturity of methodology and tool support. We call this dimension **Architecture Intelligence Maturity Levels (AI²-ML)**—measuring the maturity of the methodology, tools, knowledge base, and AI assistance an organization can call upon when making architectural decisions.

The five levels of AI²-ML

Level	Name	Core characteristics	Typical tools
L0	Drawing	Expressing architecture with general drawing tools, no formal semantics	Visio, Miro, PowerPoint
L1	Modeling	Expressing architecture with modeling tools, formal semantics but lacking constraint checking	PREEvision, Capella, Cameo, Enterprise Architect
L2	Constraint-Checked	The model supports constraint rules and consistency checking	PREEvision Pro, Capital Systems Architect
L3	Reference-Aware	The tool has a built-in reference-architecture library and can evaluate architecture against benchmarks	Reference-architecture databases, industry benchmarking platforms
L4	AI-Assisted	The tool uses AI to assist architectural decisions, auto-layout, and constraint reasoning	AI-enhanced architecture platforms
L5	Autonomous	The tool can automatically generate and evolve architecture under given goals and constraints	No mature product yet

The five scoring dimensions of AI²-ML When evaluating a tool or methodology, the AI²-ML framework uses the following five independent dimensions:

- **D1: Architectural Centralization**—supports full-spectrum expression from distributed to central compute;

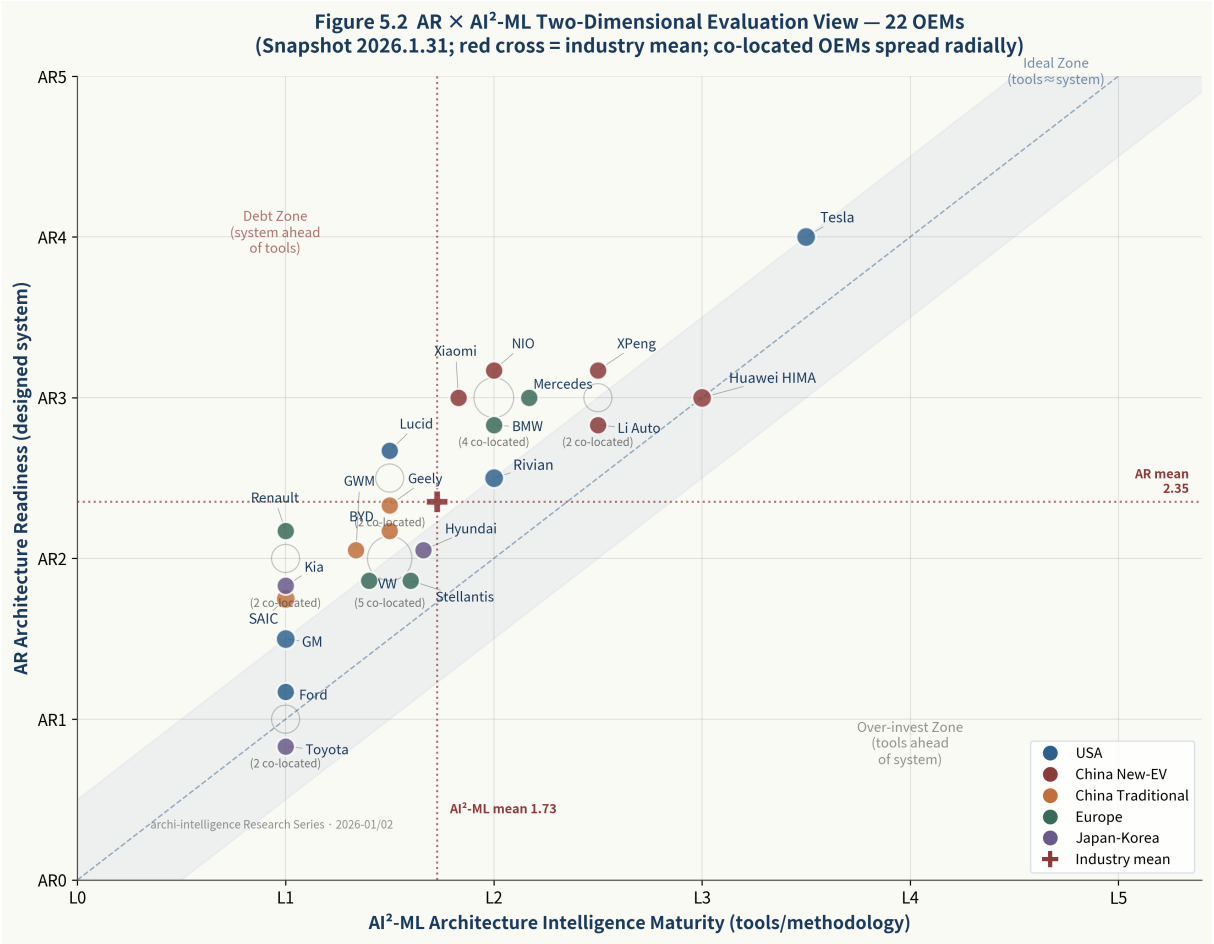

Figure 6: Figure 5.2 AR × AI²-ML two-dimensional evaluation view—22 OEMs positioned (Snapshot 2026.1.31). The diagonal band is the ideal zone (tools ≈ system).

- **D2: Software-Hardware Decoupling**—supports modeling of service orientation, virtualization, and abstraction layers;
- **D3: OTA Maturity**—supports modeling of full-lifecycle update management;
- **D4: Functional Safety Architecture**—supports formal expression of ISO 26262/SOTIF/ISO/PAS 8800;
- **D5: Compute Concentration**—supports topology and scheduling modeling of heterogeneous compute resources.

The AR × AI²-ML two-dimensional evaluation view *Figure 5.2 AR × AI²-ML two-dimensional evaluation view—22 OEMs positioned (Snapshot 2026.1.31). The diagonal band is the ideal zone; the red cross marks the industry mean.*

AR and AI²-ML are two **orthogonal, independent dimensions.** An organization may have an AR4-level system (Tesla FSD + Optimus) while holding L3–L4-level internal architecture tools; or it may have an AR2-level system yet, because it relies on L1-level drawing tools, find architectural evolution painfully slow.

This two-dimensional view reveals the **true constraint on the quality of architectural**

decisions: pursuing system-architecture maturity (AR) alone while neglecting the synchronous evolution of architecture-intelligence tools (AI²-ML) leads to the exponential accumulation of "architectural debt." If an AR4 system is maintained with L1 tools, its evolution cost will exceed what the team can bear. This is precisely one of the deep roots of the CARIAD dilemma—the target architecture was the AR3–AR4-level SSP/E3 2.0, but the tools and methodology remained at the L1–L2 level.

Cross-industry applicability of the AI²-ML framework The AI²-ML framework does not claim universality. Owing to differences in constraint characteristics, lifecycle, and regulatory intensity, different industries have starkly different demand intensity for system-level architecture tools. This study makes a preliminary assessment of applicability across six major industries:

Industry	Applicability score	Key basis for judgment
Automotive E/E architecture	5 / 5	Long lifecycle (10–15 years) + stringent regulation (ISO 26262, UN R155/R156, ISO/PAS 8800) + multi-supplier collaboration + cross-model reuse + software-hardware-decoupling evolution—all five dimensions are strongly constraint-driven
Robotics / Embodied AI	3 / 5	The industry is still early; standards and regulatory frameworks are incomplete, and the toolchain is fragmented. Demand is expected to materialize in 24–36 months as industry maturity rises
Aerospace systems	5 / 5	Extremely long lifecycle (15–30 years) + extremely strict regulation (DO-178C, ECSS, NASA NPR) + extremely low fault-tolerance budget. Small market but extremely high applicability

Industry	Applicability score	Key basis for judgment
Industrial automation	4 / 5	Medium-long lifecycle + IEC 61508/62443 regulation + modular design need. Slightly below automotive because monolithic-system complexity is usually lower
Smartphones / consumer electronics	2 / 5	Fast iteration (annual release cadence) + single-brand closed loop + standardized SoC platform dominance—architectural decisions rely more on the EDA toolchain than on system-level architecture tools
Medical devices	4 / 5	Stringent regulation (FDA, CE MDR, IEC 62304) + long validation cycles + high traceability requirements. Slightly below automotive because the industrial urgency of software-hardware-decoupling evolution is less than automotive' s

Figure 5.3 Cross-industry "tool-system" fit comparison. Automotive: the system keeps evolving under regulatory/safety drive, but dedicated architecture tools are scarce, so points skew to the debt zone and the over-investment zone is empty. Aerospace: extremely strong regulation forces tools and systems to stay synchronized, hugging the ideal zone. Internet: toolchains are advanced but individual systems are not necessarily complex, so the over-investment zone is visible.

The core insight of this assessment is: **the AI²-ML framework is most valuable in industries that simultaneously satisfy three conditions—long lifecycle, strong regulatory constraints, and software-hardware-decoupling evolution under multi-supplier collaboration.** Automotive E/E architecture is the "perfect sample" of this intersection, which

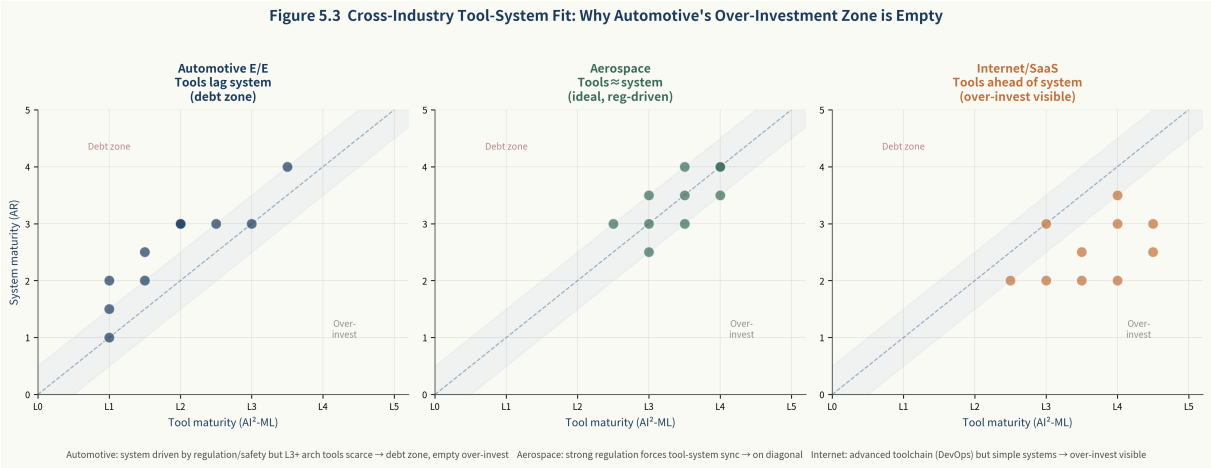

Figure 7: Figure 5.3 Cross-industry "tool-system" fit comparison: why automotive's "over-investment zone" is essentially empty.

is why this paper anchors its methodological discussion on automotive. In industries with lower applicability (such as smartphones), the marginal value of system-level architecture tools diminishes—this is not a failure of the framework, but an objective difference in industrial constraint characteristics.

The relationship between AI²-ML and existing methodologies The AI²-ML framework does not seek to replace existing architectural methodologies, but to provide, at a higher level of abstraction, an evaluation perspective on **"the maturity of architectural tools and methodology itself."** Its relationship to the following frameworks is complementary:

- **MBSE / SysML**: MBSE (Model-Based Systems Engineering) and SysML are the core methodological support at the L1–L2 level. AI²-ML does not replace them, but evaluates their implementation depth within a specific organization;
- **TOGAF / Zachman**: enterprise-architecture frameworks mainly serve business-IT alignment. AI²-ML focuses on engineering-system architecture and belongs to a different layer from enterprise-architecture frameworks;
- **ASPICE / CMMI**: process-capability maturity models focus on the maturity of the development process. AI²-ML focuses on the maturity of tools and methodology—the former asks "how standardized is the process," the latter asks "how intelligent is the tool support."

This positioning lets the AI²-ML framework neither conflict with existing standards nor fail to fill the long-vacant methodological niche of "architectural-tool maturity assessment."

0.7.5 5.5 The relationship between AR and SAE J3016

It must be made clear: the AR framework and SAE J3016 are **not mutually substitutable but two mutually independent dimensions.**

- **SAE J3016** describes **vehicle automation capability**—within a specific ODD, whether the vehicle can replace the human driver;
- **The AR framework** describes **the architectural maturity that carries this capability**—to what extent the system is software-defined, evolvable, and reusable across embodiments.

In theory, an L3 autonomous vehicle could be realized by an AR1 architecture (a highly customized, isolated ADAS system) or by an AR4 architecture (driven by an end-to-end foundation model). In industrial practice, **L4/L5 autonomous driving in fact requires an architectural foundation above AR3**—because continuous fleet learning, multi-scenario generalization, and cross-model reuse all depend on cross-device/cross-embodiment architectural capability.

0.8 Chapter 6 Conclusion, Research Boundaries, and Open Questions

0.8.1 6.1 Synthesis of the core theses

The core theses of this paper can be condensed into the following six:

Thesis one: Architecture is an engineering art of trade-offs, not a science of perfection. The two foundational laws—Trade-off, and Why > How—remain valid across two millennia. The irreversibility of contemporary complex-systems architectural decisions makes the quality of the Why reasoning directly determine the long-lifecycle cost of the system.

Thesis two: Architecture across contemporary industries is undergoing a two-layer evolution of "infrastructure convergence, divergence of control semantics and burden of proof." The shared foundation (heterogeneous SoC, virtualization, SOA, world model, OTA) is converging across industries; the industry-specific upper layer (failure philosophy, safety enclosure, liability attribution, compliance framework) will continue to diverge and further solidify under intensifying regulation.

Thesis three: Failure philosophy (fail-soft vs. fail-operational) is the fundamental watershed between automotive/robotics and consumer electronics/cloud—deeper than real-time requirements, earlier than compliance. This philosophical difference determines that an automotive OS cannot be reduced to Android, automotive OTA cannot be reduced to phone updates, and automotive AI cannot be reduced to an on-device LLM.

Thesis four: The three architectural archetypes—vertically integrated closed-loop, cross-device ecosystem, platform-enablement—correspond to three different "architectural control-point" holding strategies, with no absolute superiority, only differentiated fit to resource endowment, organizational capability, and regulatory environment. The respective successes of Tesla, Huawei, and Google are all built on clear cognition and extreme cultivation of their own control points.

Thesis five: The AR0—AR5 architectural capability-threshold framework can serve as a cross-industry common language, but must make explicit the methodological principle of "capability thresholds rather than a timeline; lower tiers do not disappear." This principle avoids an overly optimistic evolution narrative and acknowledges the multi-speed reality of industrial development.

Thesis six: Architecture Intelligence (AI²), as an emerging research paradigm, has a maturity of methodology and tools (AI²-ML) that is an independent constraint dimension on the quality of architectural decisions. Pursuing the architectural maturity of the designed system alone, while neglecting architecture-intelligence tools, leads to the exponential accumulation of architectural debt.

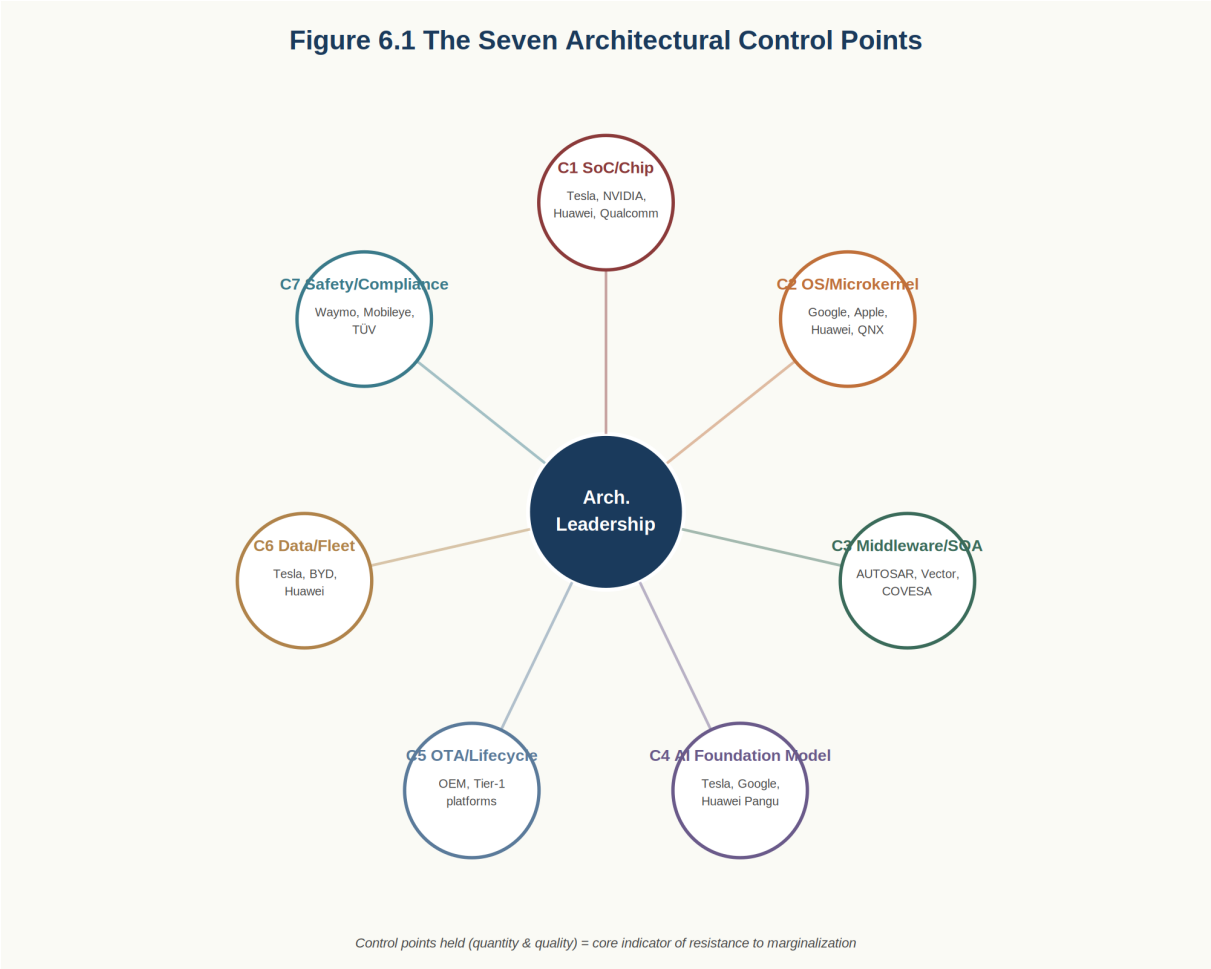

Figure 8: Figure 6.1 The seven architectural control points network. The quantity and quality of control points held are the core indicator of resistance to "marginalization" risk.

0.8.2 6.2 The seven architectural control points

Figure 6.1 The seven architectural control points network.

Cross-case analysis shows that all successful architecture leaders command at least two or three of the following seven control points:

Control point	Meaning	Typical holders
C1: Terminal entry	Devices and brands with a vast user base	Apple, Google (Android), Huawei, Xiaomi, Tesla
C2: SoC	Self-developed or deeply customized core compute chip	Apple, Tesla, Huawei, Google, NVIDIA, Qualcomm
C3: OS / microkernel	Owning the underlying operating system and kernel	Apple, Google, Huawei, QNX, Microsoft
C4: Middleware	Owning standardized middleware and service bus	AUTOSAR, ROS 2, Kubernetes, SOAFEE

Control point	Meaning	Typical holders
C5: Data closed loop	Owning a large-scale data collection → annotation → training loop	Tesla, Waymo, Huawei, Google, Mobileye
C6: Simulation and world model	Owning large-scale simulation and generative-data capability	NVIDIA, Waymo, Tesla, Wayve
C7: Safety case and compliance	Owning the engineering methodology and certification capability for functional/cyber/AI safety	Bosch, Mobileye, TÜV, Exida

The optimal strategy is not blind full-stack ownership, but identifying the "architectural control points" one can truly control—commanding at least two or three of them, while choosing platform cooperation in the rest. This is the core lesson of the CARIAD case: attempting to control all points simultaneously without sufficient internal capability instead loses all leadership.

0.8.3 6.3 The convergence-divergence endgame

Figure 6.3 Distribution of architectural maturity across 22 OEMs (Snapshot 2026.1.31), with the industry mean line.

What will keep converging:

- Silicon platforms (Arm Neoverse, Blackwell IP across domains)
- AI model architecture (Transformer + Diffusion + Flow Matching + VLA, generic across domains)
- The heterogeneous-computing paradigm and mixed-criticality OS
- Software-defined update mechanisms and continuous-deployment pipelines
- Edge-cloud AI division of labor and the world model

What will keep diverging:

- Failure philosophy and safety enclosure
- Real-time determinism guarantees
- Liability attribution and auditability
- Regulatory adaptation and compliance frameworks
- Physical actuators and sensor forms

Concretizing the endgame: at the AR5 stage, **with only one underlying multi-core dispatching microkernel OS + one foundation vision-and-logic large language model**

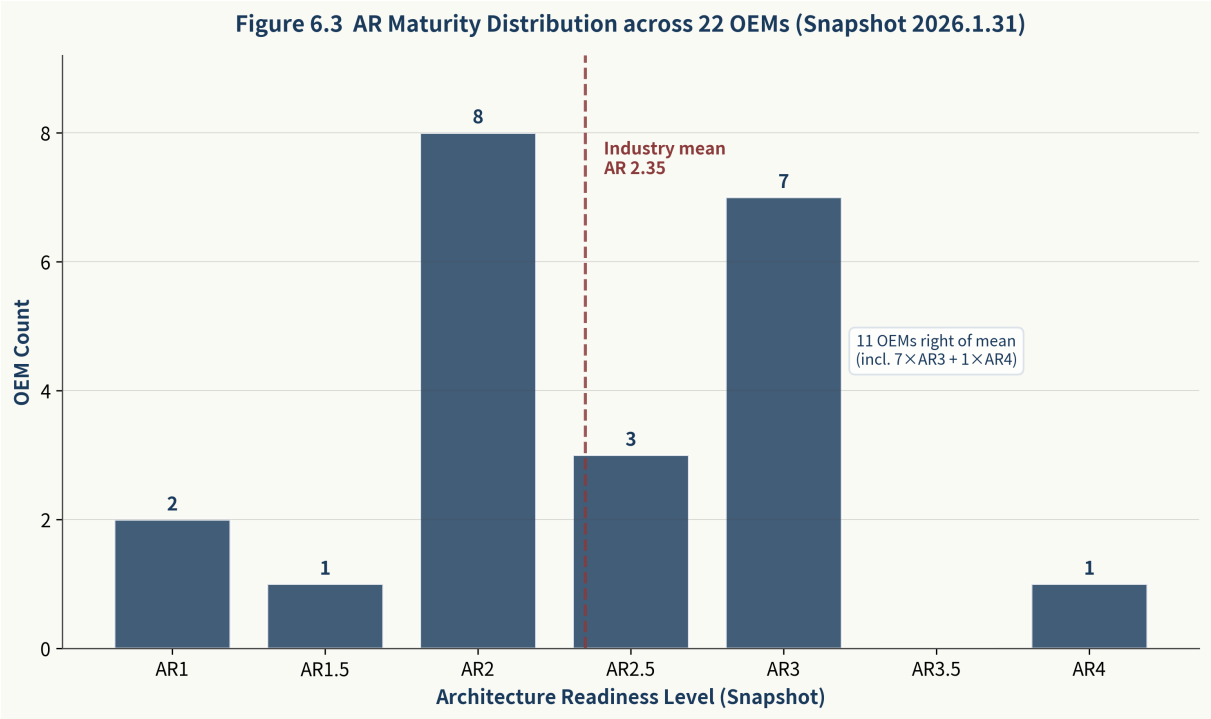

Figure 9: Figure 6.3 Distribution of architectural maturity across 22 OEMs (Snapshot 2026.1.31), with the industry mean line.

that comprehends the laws of world physics, the system can simultaneously and seamlessly drive a user's wearable holographic communication device, an unmanned physical shuttle racing through a blizzard-stricken mountain region, and a bipedal humanoid robot deftly stir-frying in the kitchen—all coordinating gracefully and operating autonomously in the complex physical 3D world. This is the ultimate vision toward which the architectural evolution of all contemporary industries jointly points—but the path, timeline, and concrete form of reaching this vision remain subject to enormous uncertainty.

0.8.4 6.4 Research boundaries and limitations

As a survey work, this study openly acknowledges the following limitations:

Limitation one: asymmetry of public information. Tesla AI Day, NVIDIA GTC, Google DeepMind releases, and AUTOSAR/AOSP open-source documents provide a wealth of verifiable technical detail; whereas the vehicle-side low-level implementation detail of Huawei and Xiaomi is relatively limited in public disclosure—many judgments can only be made as directional inference based on public white papers, annual reports, and product-release information. This paper strives to flag this asymmetry in comparisons, but readers should still beware of misreading "differences in disclosure" as "differences in capability."

Limitation two: uncertainty of time extrapolation. The actual deployment speed of Tesla AI6/AI7, Huawei Pangu 6, Google Gemini 4, Waymo's 7th-generation Driver, Mobileye EyeQ7, etc., after 2026, remains subject to the multiple influences of regulatory cadence, supply-

chain stability, partner execution, and geopolitics. The time windows mentioned in this paper are "reasonable estimates based on current roadmaps"and do not constitute predictive commitments.

Limitation three: the research nature of the AR0—AR5 framework. The AR framework is a methodological proposal of this study, not an industry-consensus standard. The robotics industry currently has no global unified architectural-maturity standard equivalent to SAE J3016. The value of this framework lies in providing a common language for cross-domain discussion, not in replacing any existing industry standard. The framework itself welcomes criticism, revision, and extension from the industrial and academic communities.

Limitation four: the boundaries of the AI²-ML framework. The AI²-ML framework fits well in long-lifecycle, strongly-constraint-driven domains such as automotive E/E architecture, robotics, aerospace, and industrial automation; it fits less well in fast-iterating, relatively loosely-constrained domains such as smartphones and pure internet applications (smartphone architecture relies more on the EDA toolchain than on system-level architecture tools). The framework does not claim universality.

Limitation five: this paper does not constitute investment advice or industrial-policy advice for any specific enterprise. All case selection and analysis serve methodological-research purposes only.

0.8.5 6.5 Open questions

In the course of this study, the following open questions were identified, and the academic and industrial communities are invited to explore them jointly:

1. **Where is the provable-safety boundary of end-to-end neural networks?** Under the EU AI Act and ISO/PAS 8800 frameworks, can a pure-black-box end-to-end system obtain an L4/L5 liability claim? Or does it inevitably require an explicable intermediate layer and a runtime monitor?
2. **Where is the algorithmic limit of cross-embodiment generalization?** Can Tesla' s "occupancy networks are embodiment-agnostic" hypothesis be validated across broader robot forms (quadruped, wheeled, flying)? Or will it encounter form-dependent physical limits?
3. **What is the minimal set of architectural control points?** Is there a minimal control-point subset that allows a company to still lead architectural evolution without commanding the SoC?
4. **What organizational capability does the leap from AR3 to AR4 require?** There are still few examples in the industry that have completed this leap. Is the CARIAD dilemma an isolated case or a general phenomenon?
5. **What is the L4—L5 implementation path of AI²-ML tools?** What formalization,

verifiability, and trust-building challenges does automating architectural decisions with AI face?

0.8.6 6.6 Acknowledgments and methodological statement

This study uses a survey research method, with material sources covering:

- **Academic literature**: journals such as IEEE Transactions, SAE Technical Papers, ACM Computing Surveys, Springer Automotive Innovation;
- **Standards documents**: standards and white papers issued by ISO, SAE, IEEE, UNECE, AUTOSAR, COVESA, and others;
- **Corporate disclosures**: Tesla AI Day, NVIDIA GTC, Google DeepMind blog, Huawei Developer Conference (HDC), Xiaomi launch events, Waymo technical blog, Bosch/Aptiv white papers;
- **Regulatory documents**: UNECE WP.29, the European Commission, China MIIT GB-series standards;
- **Industry research**: public reports from McKinsey, Roland Berger, Bain, BCG, Strategy Analytics, IHS / S&P Global Mobility;
- **Authoritative media**: professional outlets such as CnEVPost, CarNewsChina, TechCrunch, The Robot Report, Electrive, Electronic Design.

This study has done its utmost to cross-validate specific figures and time points when citing them, but owing to differences in the update speed of public information, some data may deviate from the latest facts. If readers find specific factual errors, feedback is welcome for updating in subsequent versions.

0.8.7 6.7 An open invitation to the research and industrial communities

As a survey work, this study openly acknowledges its limitations as the output of a single research institution. Architecture Intelligence (AI²), as an emerging research paradigm, requires broader academic and industrial participation for its methodological maturation, empirical testing, and cross-industry application. On the occasion of this research series' release, we extend specific invitations to several types of readers.

6.7.1 To the academic research community Invitation one: methodological criticism and extension. The AR0-AR5 framework and the AI²-ML framework proposed in this study are preliminary methodological proposals, far from settled. We invite researchers from

systems engineering, software architecture, automotive engineering, robotics, and AI safety to offer criticism, revision, and extension. Specific directions worth exploring include: whether the judgment indicators of each AR stage need further formalization (the current descriptions lean qualitative—could quantitative thresholds be introduced?); whether the AI²-ML five scoring dimensions need extension or merging (in particular, in robotics and embodied-AI scenarios, is a new "form-generalization capability" dimension needed?); and whether the relationship between AR and SAE J3016 needs more precise formalization.

Invitation two: cross-industry empirical research. This study's benchmark across five industries is based on public information. We invite researchers with frontline industry access to conduct deeper empirical research validating or challenging our core judgments—especially longitudinal AR-evolution case studies within a single OEM or Tier-1 (ideally with 5–10 years of data), cross-OEM horizontal comparisons based on consistent methodology, and deep dissections of failure cases.

Invitation three: exploring a standardization path. As research-oriented frameworks, the evolution of AR and AI²-ML toward industry standards requires the participation of standards bodies. We invite relevant working groups at SAE, IEEE, ISO, and INCOSE to assess the standardization potential of these two frameworks. The archi-intelligence Research Team is willing to participate as an initial contributor of the methodology but does not claim ownership of any final standard—any final standard should be formed by industry consensus, not led by a single research institution.

6.7.2 To OEMs and Tier-1 suppliers Invitation four: architectural self-assessment and feedback. In the 2026-02 report (*State of E/E Architecture 2026*), this study assesses the reference architectures of 22 OEMs on AR and AI²-ML. We acknowledge this assessment is entirely based on public sources and inevitably contains bias. We sincerely invite the internal architecture teams of the assessed OEMs to provide feedback, including corrections of facts omitted or outdated in public sources, revisions of architecture-evolution paths (especially undisclosed next-generation platform details), and disagreements in methodological application. The archi-intelligence Research Team commits that all feedback will be seriously considered, with major corrections explicitly recorded in subsequent versions. We also state clearly: accepting feedback does not mean abandoning independent judgment; when feedback conflicts with public evidence, we reserve the right to maintain the original assessment, while noting the existing dispute in the text. Feedback channel: corrections@archi-intelligence.org

Invitation five: open discussion of architectural evolution. Bosch's six-stage framework described in Chapter 2 is already industry consensus, but evolution toward stage 6.0 (vehicle-cloud coordination) and higher stages (AR4 multi-embodiment physical AI platform, AR5 trusted general embodied agent) still has many unresolved questions. We invite senior

architects at OEMs and Tier-1s to openly discuss these issues. These questions have no standard answers; we value the process of serious discussion over the production of hasty conclusions. archi-intelligence plans to organize a series of closed-door roundtables in the second half of 2026 (after ELIV 2026), inviting core architects from OEMs and Tier-1s. Those interested may contact: research@archi-intelligence.org

6.7.3 To regulatory and standardization bodies Invitation six: incorporating architectural maturity into regulatory consideration. Current automotive regulatory frameworks (ISO 26262, SOTIF, ISO 21434, UNECE R155/R156) focus mainly on the safety of specific functions and components. However, one of this study's core judgments is that **architectural-level irreversibility determines the safety and compliance boundaries of the system across its full lifecycle.** A traditional distributed architecture at the AR1 stage and a cross-device collaborative architecture at the AR3 stage pose fundamentally different challenges in safety management. We suggest regulators begin to consider treating "architectural maturity" as an independent dimension in the regulatory review of new vehicle models.

Invitation seven: Architecture Intelligence (AI²) as a new independent regulatory domain. The AI²-ML framework introduced in Chapter 5 describes **the maturity of architectural-decision tools**—a domain currently entirely unregulated. As practices such as LLM-assisted architectural decisions, AI-generated architecture proposals, and AI-automated constraint checking become widespread, the following regulatory questions will inevitably arise within 5–10 years: do AI-assisted architecture-decision records (ADRs) carry legal effect? When an AI tool's architectural recommendation is adopted and causes a problem, how is liability attributed? Does a certification mechanism for the architecture-intelligence tool itself (analogous to DO-178C for flight-control software) need to be established? We invite regulators and legal scholars to start the discussion early.

6.7.4 To investors and strategic decision-makers Invitation eight: the architectural dimension as an independent metric in due diligence and valuation. Current automotive-industry investment and strategic-decision frameworks (such as Wards Intelligence's SDV ScoreCard, PwC Strategy&'s SDV Maturity Model) focus mainly on market share, sales forecasts, financial health, and organizational capability. Architectural-level judgment is either absent or oversimplified into labels such as "whether a zonal architecture is used." One of this study's core propositions is that **architectural maturity (AR) and architecture-intelligence maturity (AI²-ML) should be treated as independent investment and strategic-decision dimensions, considered alongside the traditional market share and financial health.** Specifically: an OEM at the AR4 stage, even with low current market share, may have a far higher long-term competitiveness ceiling than an OEM at AR2 with higher

share; an OEM with L3–L4-level AI²-ML tools has a far lower "friction cost" of architectural evolution than one stuck at L1 tools (a difference that, over a 5–10-year horizon, may equate to a cumulative R&D-efficiency gap of billions of dollars); and a Tier-1's quantity and quality of holdings across the seven control points (C1–C7) is the core indicator of its resistance to "marginalization" risk. We invite investment banks, industry consultancies, PE funds, and strategy departments to incorporate these dimensions into their analytical frameworks. archi-intelligence plans to release a companion "AR & AI²-ML Investment-Analysis Methodology Guide" in the second half of 2026.

6.7.5 To the open-source community and developers Invitation nine: participating in the evolution of the archi-intelligence Research Series. This research series is released under the CC-BY 4.0 license, allowing anyone to download, adapt, and cite. But the deeper invitation is to make this series a **genuine open-collaboration research project**—raising issues and pull requests on the methodology and conclusions (once the GitHub repository is public), contributing new case studies (especially perspectives from different regions and industries), assisting with translation (v1.0 provides Chinese and English; we hope to extend to Japanese, German, and Korean), and assisting with visualization (better designers are welcome to contribute improved versions of the key infographics). We will explicitly list all contributors (by scale of contribution) in the v1.1 revision, and consider granting formal "contributing researcher" attribution depending on the scale of contribution.

6.7.6 A research series' commitment Finally, we wish to end this invitation section with a concrete commitment. The archi-intelligence Research Team commits to:

- **Continuous publication**—this series is planned to run for at least 5 years, with at least 2–3 formal working papers released annually;
- **Continuous revision**—after the v1.0 release, a revised edition every 6–12 months, publicly recording all major corrections;
- **Continuous openness**—all research methodology, primary data sources, and citation lists 100% public;
- **Continuous independence**—refusing funding from any assessed entity, maintaining editorial independence.

This research was compiled by the archi-intelligence Research Team.
Architecture Intelligence Research Series · Working Paper 2026-01
—End of main text —

0.9 Back Matter —Appendices

0.9.1 Appendix A: Glossary

Arranged alphabetically. Each entry includes the term, a concise definition, and the section of first appearance in this report.

A

Architecture —The fundamental concepts or properties of a system in its environment, embodied in its elements, relationships, and the principles of its design and evolution (ISO/IEC/IEEE 42010 definition). In this report, "architecture" refers specifically to the organization of complex systems in an engineering context, not in the architectural (building) sense. → First appears: §1.1

Architecture Description —The concrete expression of architecture, including but not limited to architecture diagrams, design documents, decision records, and constraint specifications. ISO 42010 emphasizes that the architecture description is more comprehensive than a single diagram. → §1.3

Architecture Intelligence (AI²) —The emerging research paradigm proposed by this research series: the systematic study of applying intelligent methods (knowledge graphs, formal verification, large language models, reinforcement learning) to architectural design, constraint checking, reference benchmarking, and evolution prediction. → §1.7

Architecture Intelligence Maturity Levels (AI²-ML) —A 5-level evaluation framework proposed by this series, measuring the maturity of the methodology, tools, knowledge base, and AI assistance an organization can call upon when making architectural decisions. → §5.4

Architecture Readiness (AR) Levels —A 0–5 framework proposed by this series, measuring the architectural-evolution stage of the designed system (vehicle, robot, intelligent terminal). Analogous to SAE J3016' s autonomous-driving levels, but describing the architectural maturity that carries capability, rather than the automation capability itself. → §5.1

ASIL (Automotive Safety Integrity Level) —The automotive functional-safety levels defined by ISO 26262, divided into ASIL A/B/C/D. ASIL-D is the highest level, corresponding to the strictest safety requirements. → §2.1

AUTOSAR (AUTomotive Open System ARchitecture) —A global automotive-industry standardized software framework and E/E system architecture, comprising two parallel versions: the Classic Platform (signal-driven, deeply embedded ECU) and the Adaptive Platform (service-oriented, high compute). → §2.1

C

CARIAD (Car.I.AM Digital) —VW Group' s software subsidiary, founded in 2020, responsible for the Group Software Stack. The delay of the E3 2.0 platform is its representative dilemma. → §2.5

Central Compute —The core component of E/E architecture stage 5.0: 1–2 high-performance

central compute units (HPCs) handle most computation, coordinating with zone control units (ZCUs). → §2.3

Cross-Embodiment Reuse —The reuse of the same compute stack, perception stack, and training infrastructure across different physical forms (such as car and humanoid robot). Tesla FSD to Optimus is the canonical case. → §4.1

E

E/E Architecture —The overall organization of all electronic hardware, sensors, actuators, compute units, communication networks, and underlying software platforms within a vehicle. → §2.1

Embodied Intelligence —An AI system able to perceive, reason, and act in the real physical world. In this study, it refers specifically to humanoid robots and cross-embodiment AI agents. → Introduction

F

Fail-Operational —The automotive and robotics safety philosophy: a critical function can continue to work or safely degrade after partial component failure. Fundamentally contrasts with the fail-soft of consumer electronics. → §3.7

Fail-Soft —The consumer-electronics and cloud failure philosophy: failure is allowed, backstopped by redundancy and rapid recovery; the cost of failure is degraded experience, not physical harm. → §3.3

Foundation Model —A large model pre-trained on massive data that can be adapted to many downstream tasks. In this study, it refers to driving/embodied foundation models such as the Waymo Foundation Model and Pangu. → §4.3

H

Hypervisor —A virtualization layer that runs multiple isolated operating systems on the same SoC. In automotive architecture, it allows a hard-real-time RTOS and a consumer-grade OS to safely coexist on the same chip—the key technology bridging the digital-physical divide. → §3.4

O

Occupancy Networks —Tesla' s perception method disclosed at 2022 AI Day: representing the world as a voxelized 3D dense occupancy field plus an occupancy flow field, producing an embodiment-agnostic 3D scene representation, and the core technical reason the FSD stack can migrate to Optimus. → §4.1

OTA (Over-The-Air) —Updating vehicle software remotely over a wireless network. Vehicle-side OTA, owing to the failure philosophy, is far more complex than phone-side OTA. → §2.1

S

SDV (Software-Defined Vehicle) —A vehicle whose core functions are defined and

continuously upgraded by software, replacing a large number of isolated controllers with fewer, more powerful compute nodes. → §2.1

SOTIF (Safety Of The Intended Functionality) —The safety of the intended functionality defined by ISO 21448, addressing safety risks arising from performance limitations or foreseeable misuse rather than component failure. → §2.6

V

VLA (Vision-Language-Action) Model —An end-to-end model that takes visual and language input and directly outputs action. Increasingly the mainstream paradigm for both autonomous driving and embodied robotics. → §2.4

Z

Zonal Architecture —An architecture that partitions vehicle electronics by physical zone (rather than functional domain), with zone control units (ZCUs) coordinating with central compute—the hallmark of E/E architecture stage 5.0. → §2.3

> The full bilingual glossary (with Chinese cross-references) is available in the Chinese edition.

0.9.2 Appendix B: AR & AI²-ML Quick-Reference Scoring Card

AR (Architecture Readiness) —capability thresholds

Level	Name	One-line discriminator
AR0	Mechanical-Electrical Island	One controller per function; software tied to hardware; no OTA
AR1	Domain-Level Integration	5±2 domain controllers; OTA per-ECU; cross-domain flow still cut off
AR2	Zonal Platform	Zonal + central compute; controllers down to 7–15; Hypervisor coexistence
AR3	Cross-Device Collaborative Agent	Car/phone/wearable/home share identity, context, task migration
AR4	Multi-Embodiment Physical AI Platform	Train-simulate-deploy unified loop; world model shared across car/robot
AR5	Trusted General Embodied Agent	General policy layer across entities; provably safe/compliant

AI²-ML (Architecture Intelligence Maturity Levels) —tool/methodology maturity

Level	Name	One-line discriminator
L0	Drawing	General drawing tools, no formal semantics
L1	Modeling	Modeling tools with formal semantics, no constraint checking
L2	Constraint-Checked	Model supports constraint rules and consistency checking

Level	Name	One-line discriminator
L3	Reference-Aware	Built-in reference-architecture library, benchmark-based evaluation
L4	AI-Assisted	AI assists decisions, auto-layout, constraint reasoning
L5	Autonomous	Auto-generates and evolves architecture under goals/constraints

AI²-ML five scoring dimensions: D1 Architectural Centralization · D2 Software-Hardware Decoupling · D3 OTA Maturity · D4 Functional Safety Architecture · D5 Compute Concentration.

Usage note: AR and AI²-ML are orthogonal. A high-AR system maintained with low-AI²-ML tools accumulates "architectural debt" exponentially (the deep root of the CARIAD dilemma). The diagonal band in Figure 5.2 is the ideal zone (tools ≈ system).

0.9.3 Appendix C: Cross-Industry Five-Dimensional Data Tables

This appendix provides the complete raw data for the cross-industry benchmark of Chapter 3. All data are drawn from publicly released official information, public-company filings, patent literature, industry-standards-body publications, and mainstream technical media.

The five dimensions (hardware, software, data/AI, ecosystem, safety) across the five industries (automotive E/E, smartphone/consumer electronics, internet/cloud, robotics, embodied AI) are summarized in the matrix of §3.1. The complete per-cell source attribution—including specific chip specifications, OS versions, model parameter counts, ecosystem metrics, and applicable standards—is maintained in the Chinese edition' s Appendix C and the companion working paper *State of E/E Architecture 2026* (Working Paper 2026-02), which provides 22-OEM detailed scoring with a five-tier source pyramid.

> For the full data tables with line-by-line source attribution, see the Chinese edition or the 2026-02 companion report.

0.9.4 Appendix D: About the archi-intelligence Research Series

Mission

archi-intelligence is an independent academic research body dedicated to establishing Architecture Intelligence (AI^2) as a research paradigm. Its mission is to advance the standardization and comparability of cross-industry architectural engineering practices through open methodology, transparent data attribution, and rigorous peer review.

Publication model

- All reports are permanently free and released under CC-BY 4.0.
- Each report is archived on Zenodo with a permanent DOI.
- All methodology, primary sources, and citation lists are 100% public.

The v1.0 release: the "three-deliverable" set

- **2026-01** (this report): *The Architectural Migration of the Century* —the flagship report (ontology, cross-industry benchmark, AR0—AR5 and AI^2-ML frameworks).
- **2026-02**: *State of E/E Architecture 2026* —AR & AI^2-ML assessment of 22 OEMs across a dual time dimension (Snapshot + Confirmed Roadmap).
- **2026-03**: *Tesla FSD-Optimus Unified Stack* —an in-depth dissection of the only complete AR4 vertically integrated closed-loop case.

Editorial independence

The research' s methodological choices, case evaluations, and conclusions are made independently by the archi-intelligence Research Team, uninfluenced by any commercial interest, political stance, or geopolitical preference. (See the full Conflict-of-Interest Disclosure in the Front Matter.)

Commitments

- Continuous publication (at least 5 years, 2–3 working papers annually);
- Continuous revision (a revised edition every 6–12 months, all major corrections publicly recorded);
- Continuous openness (all methodology and sources public);
- Continuous independence (refusing funding from any assessed entity).

Contact

- Research: research@archi-intelligence.org
- Corrections: corrections@archi-intelligence.org
- Website: https://archi-intelligence.org

This research was compiled by the archi-intelligence Research Team.

Architecture Intelligence Research Series · Working Paper 2026-01 (English Edition)

—End of document —
