MHP
A PORSCHE COMPANY

BOSCH

WHITEPAPER

When In-Vehicle AI Scales, Who Controls the Cost?

Strategic Choices
for the Next Generation
of Smart Cabin Platforms

Contents

Foreword

Artificial intelligence is becoming a defining element of the next generation of vehicles. Following its impact on ADAS, the Smart Cabin is emerging as the next frontier of the AI-defined vehicle. As vehicles move from software-defined to AI-defined, the cockpit is where that shift becomes visible to the customer: a system that understands context, interacts with occupants, coordinates services and executes complex tasks. We are moving from generative AI to agentic use cases and agentic loops.

This shift requires OEMs to make consequential decisions across four success factors: compute, AI, data and portfolio scale. These factors multiply rather than add, and they behave differently. These decisions determine which use cases run at the edge or in the cloud, how interaction data is used and whether In-Vehicle AI can scale economically across platforms and markets.

OEMs must act despite significant uncertainty. Declining token prices may be offset by rising usage and interaction complexity, while willingness to pay and future revenue pools remain difficult to forecast. At the same time, compute platforms must be scoped years before the use cases and revenue pools they serve are understood. Too little capacity constrains differentiation and increases cloud dependency, with unexpectedly high consumption costs. Too much capacity raises vehicle costs without guaranteed monetization. The objective is therefore not to predict the perfect architecture, but to retain flexibility and control.

Bosch and MHP have joined forces because we see the challenges linked to In-Vehicle AI in our daily work. AI decisions are often based on a single demand forecast, without transparency on today's cost per active user. At the same time, E/E architecture choices constrain tomorrow's AI potential. Neither challenge is solved by lower token prices.

This whitepaper gives leadership teams a practical decision guide that connects strategic ambition, technical feasibility and economic sustainability. Its core question is not whether In-Vehicle AI will scale, but who will control the economics, the experience and the customer interface across different edge vs. cloud compute configurations.

Our central message is straightforward: waiting for uncertainty to disappear is not a viable strategy. OEMs must start building the organizations, architecture, capabilities, data foundations and decision mechanisms required to move forward without locking themselves into assumptions that may not hold. Market experimentation and sequenced architecture decisions make it possible to commit capital late while learning and data accumulate early.

The winners in In-Vehicle AI will be those that retain control over where intelligence is executed, how costs scale, how data compounds and how the experience evolves across the vehicle lifecycle.

Alex Artamonow
Partner,
Software-Defined Vehicle
MHP Management- und IT-Beratung GmbH

Christian Köpp
Senior Vice President
Business Unit Compute Performance,
Cross-Domain Computing Solutions,
Robert Bosch GmbH

Executive Summary

1 **The benchmark has moved from features to orchestration.**

Competitive advantage no longer comes from offering an AI assistant, but from integrating intelligence across the vehicle, customer context and service ecosystem. Execution depth, iteration speed and control of the AI orchestration now define the benchmark for the AI-defined vehicle.

2 **Agentic AI changes the cost physics.**

Agentic tasks combine reasoning, tool use, memory, context retrieval and multimodal inputs, consuming substantially more inference than simple generative requests. Agentic loops multiply this effect and can turn seemingly simple tasks into workloads requiring twenty to fifty times more inference.

3 **Cloud-only architectures make success the cost risk.**

Better experiences increase adoption, interaction frequency and complexity. In a cloud-only architecture, this directly increases variable cost, making successful adoption one of the most important and least adequately modeled risks in current In-Vehicle AI business cases.

4 **Cost control is an architecture capability, not a procurement outcome.**

Total cost depends on model routing, model mix, edge-cloud distribution, token budgets, use-case prioritization and partner governance. Lower token prices, reserved capacity and volume discounts can improve costs, but they cannot replace architectural control.

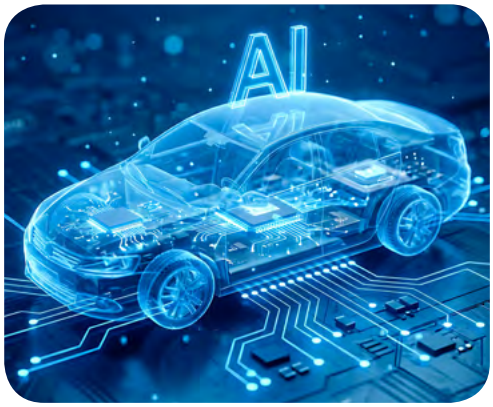
5 **Costs can be measured; revenues remain uncertain.**

Token consumption, use-case diversity and interaction complexity are observable, while willingness to pay, take rates and adoption speed remain difficult to forecast. OEMs should therefore begin a controlled hybrid path preserving flexibility before the next platform generation freezes critical architecture decisions.

6 **Success is multiplicative, and every factor matters.**

Success = Data × AI/SW Verticalization × Compute × Portfolio Scale. Weakness in any factor can undermine the entire business case. Compute and AI models can be purchased or licensed, but interaction data must be accumulated and portfolio scale requires consolidated platforms that reuse common AI, software and compute layers.

The Benchmark Has Moved



For roughly a decade, in-cabin intelligence was assessed as a feature list. Does the vehicle offer voice control? Does it understand natural language? Can it answer open-domain questions? Programs were scoped, sourced and benchmarked against that list.

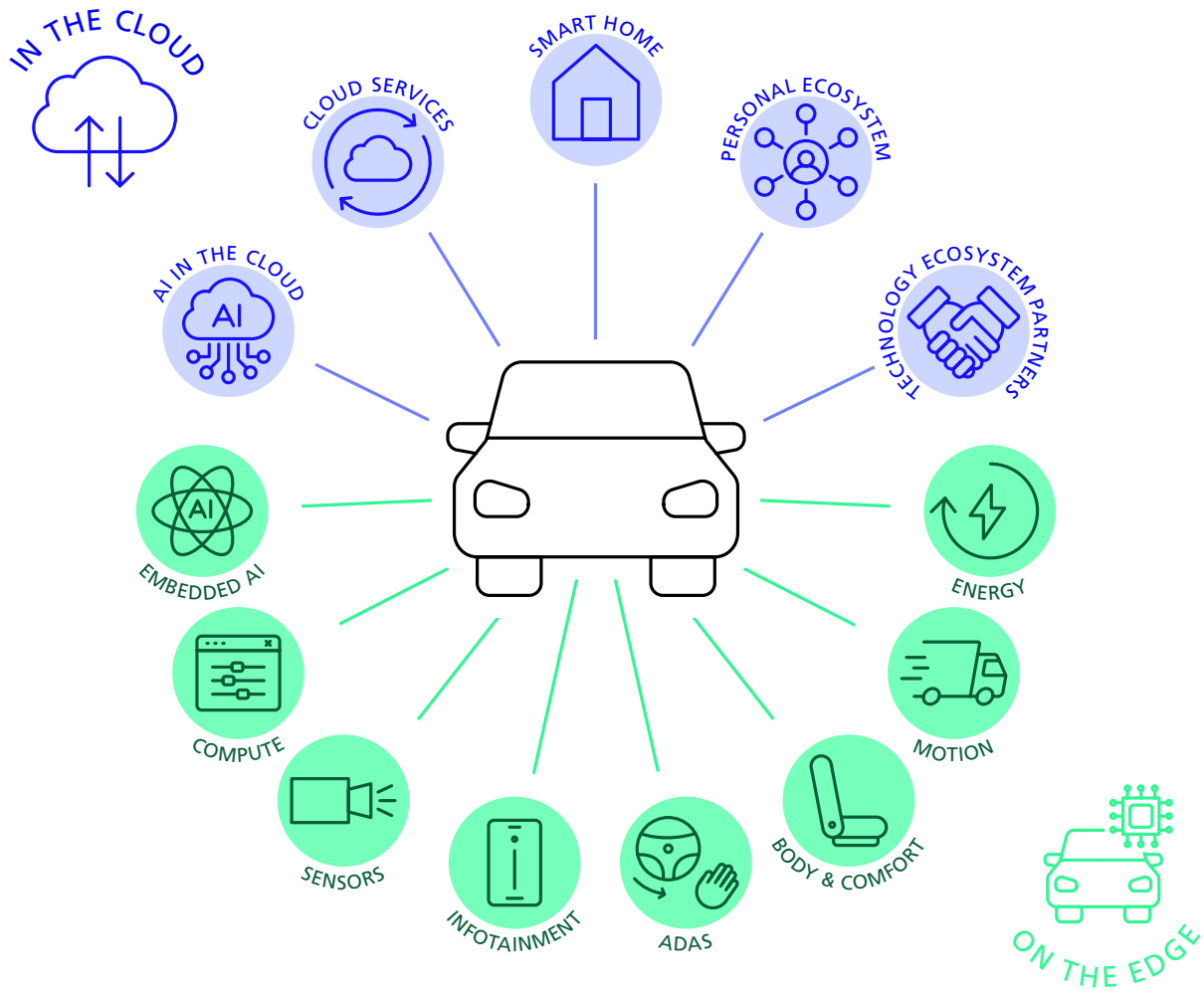
That framing is no longer sufficient. The competitive question is not whether a vehicle has an AI assistant, but how deeply intelligence is integrated into vehicle context, the service ecosystem of the customer and execution. An agent that understands vehicle state, remembers the user, perceives the cabin multimodally, calls external services and routes each request to the appropriate model is not simply a better voice assistant. It is a different product category, with a different architecture and a different cost structure.

This view is not confined to vehicle manufacturers and their suppliers. Asked what will move the industry from software-defined to AI-defined vehicles, Matt Crowley, Group Product Management at Google, locates the shift in the triggering mechanism rather than in the capability itself:

"Instead of hardcoded software routines triggering predefined actions, the vehicle may perceive the full in-cabin and external context and autonomously orchestrate vehicle functions, predictive maintenance, and personalized experiences with minimal explicit user prompting."

Matt Crowley,
Group Product Manager
Google Android Automotive

SMART CABINS ARE ENABLING DIFFERENT EXPERIENCE WITH DIFFERENT TRIM LEVELS



NATURAL & HUMAN LIKE

EMOTION INTELLIGENCE

CONTEXT INTELLIGENCE

DOMAIN CONNECT ALL ACCESS

ONE INTELLIGENT, EVOLVING ECOSYSTEM THAT GOES WAY BEYOND THE VEHICLE

Exhibit 1: The ecosystem of In-Vehicle AI solutions

Three observations define the new benchmark.

1

Integration depth beats feature count.
The differentiating capability is the combination of vehicle context, persistent user memory, multimodal perception, proactive assessment, service execution and intelligent model routing. Individual elements can be procured. Combining them into one coherent experience is an architecture decision.

This distinction matters because integration depth creates both value and exposure. The deeper the system reaches into vehicle and service domains, the more useful it can become. The more context it processes, the more actions it executes and the more important the decision becomes about where, and by which model each request is handled. Routing may be invisible to the customer, but it determines whether the experience can scale economically. The orchestrator module of the AI architecture is becoming the differentiating control point that should be owned by any OEM with an ambition to offer a compelling AI-defined experience.

2

Iteration cadence beats launch scope.
AI-defined competitors increasingly improve cockpit capabilities through frequent software releases rather than waiting for the next vehicle generation. Market monitoring in China illustrates the structural difference particularly clearly: manufacturers operating under the same regulatory and connectivity environment still show materially different software release cadences.

For established manufacturers, the constraint is therefore rarely model availability alone. It is the coordination required across vehicle domains, supplier systems, validation processes and release governance. Where every relevant change requires broad reintegration or renewed validation, an AI product cannot evolve at AI speed. System partitioning and organizational ownership consequently become competitive capabilities, and both are decided long before launch.

3

The market is moving from AI ambition to provable value.
The 2026 Gartner Hype Cycle for Agentic AI and the Omdia/Sonatus SDV Reality Check point in the same direction: governance, cost control and measurable value are moving from afterthoughts to design criteria. For In-Vehicle AI, capability alone is no longer enough. The experience must also be operable, measurable and economically scalable.

Underneath these three observations sits a simple structural relationship:

Success =
Data × Compute
× AI/SW Verticalization
× Portfolio Scale

The relationship is deliberately multiplicative. A program with strong models but little privileged vehicle and interaction data produces a capable generic assistant without relevant differentiation. A program with valuable data but insufficient compute cannot use it at the required latency and cost. A program with both, but without the orchestration capability to decide how intelligence is executed, has assembled the components without controlling the product. A program without enough scale across the portfolio is at risk to reach a positive ROI over lifetime.

The four factors also behave differently. Models can be licensed from a rapidly moving market. Compute can be procured, although vehicle-platform lead times constrain when and where they become available. Data is accumulated through deployed experiences and cannot be purchased retrospectively. Waiting therefore does not merely postpone expenditure; it can postpone the learning and data accumulation on which future differentiation depends.

"Vehicles will have «multiple intelligences», that will need to leverage compute where available, allocating it depending on mixed-critical considerations, quality of service, user expected user experience."

Stefano Marzani,
Emerging Tech Leader,
Automotive and Manufacturing at AWS

The management question consequently changes. It is no longer "which AI function should we integrate?" It is "which architecture, cost logic and control mechanisms allow an AI experience to improve continuously and scale economically over the vehicle lifecycle?"

The ladder on the next page is not a target hierarchy. Different manufacturers may rationally stop at different levels. But each step changes the economics. Once intelligence moves from answering questions to executing tasks and continuously interpreting context, the traditional feature-cost logic breaks down. Chapter 2 examines that change in cost physics.

5 AGENTIC LOOPS

- Automated and proactive loops running in the background
- Connecting domains
- 3rd party services

COST CHARACTERISTIC
- Low predictability as loops could run automated and in dynamic frequencies

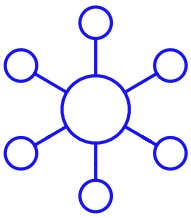
ARCHITECTURAL IMPLICATION
- Token consumption transparency
- Continuous optimization for models with best cost/performance curve

4 VEHICLE-WIDE ORCHESTRATION

- Cross-domain coordination
- Multimodal context
- Proactive execution

COST CHARACTERISTIC
- Usage-driven cost increasingly decoupled from explicit user request

ARCHITECTURAL IMPLICATION
- Edge/cloud distribution and orchestration become strategic architecture decisions

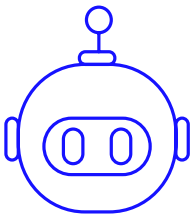


3 AGENTIC COCKPIT

- Multi-step tasks
- Tool calls
- Service execution and memory

COST CHARACTERISTIC
- Variable cost per task
- Complexity increases consumption disproportionately

ARCHITECTURAL IMPLICATION
- Routing
- Observability and service integration become essential



2 Gen AI ASSISTANT

- Open-domain dialogue and knowledge access

COST CHARACTERISTIC
- Variable cost per interaction

ARCHITECTURAL IMPLICATION
- Cloud access and model integration become relevant

1 VOICE ASSISTANT

- Command-and-control
- Predefined domains

COST CHARACTERISTIC
- Variable cost per interaction

ARCHITECTURAL IMPLICATION
- Cloud access and model integration become relevant

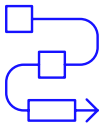
Exhibit 2: The In-Vehicle AI Integration Depth Ladder

The Token Economy: The New Cost Physics

The unit of analysis is wrong in most business cases

A vehicle lives ten to fifteen years. A cockpit ECU is costed once, at a unit price, and booked into the vehicle component cost. An AI experience is costed continuously, as inference, for as long as the vehicle is used. The relevant unit is therefore not the token price and not the Bill of Material (BOM) delta. It is cost per monthly active user, per month, sustained across the vehicle lifecycle and, at portfolio level, the cumulative operating expense of the installed base.

That second figure is the one that surprises steering committees. Per-vehicle cost can look flat or even declining while total exposure grows sharply, simply because the AI-active share of the fleet is growing at the same time. A program that models per-vehicle cost and ignores fleet penetration can present a reassuring chart and an incorrect conclusion.



One agentic task shows why

Consider a driver approaching the next appointment who asks the cockpit agent to find and reserve a restaurant, reroute around traffic and adjust the cabin temperature. Climate control may execute locally, while search and reservation require several reasoning steps, tool calls and cloud interactions. The workload is driven less by the spoken request than by the surrounding context: system instructions, tool schemas, vehicle state, location, memory and intermediate results processed across the task.

The hidden consumption is context, not output. Persistent memory, multimodal input and tool responses enlarge the context across successive calls; proactive assistance can create inference even without an explicit request. An agentic task can therefore consume twenty to fifty times the inference of a simple generative request without producing an answer that looks proportionally longer. The path towards agentic loops multiply the token consumption again by a significant factor.

Orchestration efficiency therefore matters before model prices do. Prompt and context caching, context compression, routing to the smallest sufficient model and local execution of repetitive tasks reduce the inference required to resolve the same intent. Together, these mechanisms determine how directly usage translates into cost. At global OEMs, it’s often an organizational challenge. Organizations and system architectures onboard and offboard must be re-wired in a way that constant model updates and replacements are possible.

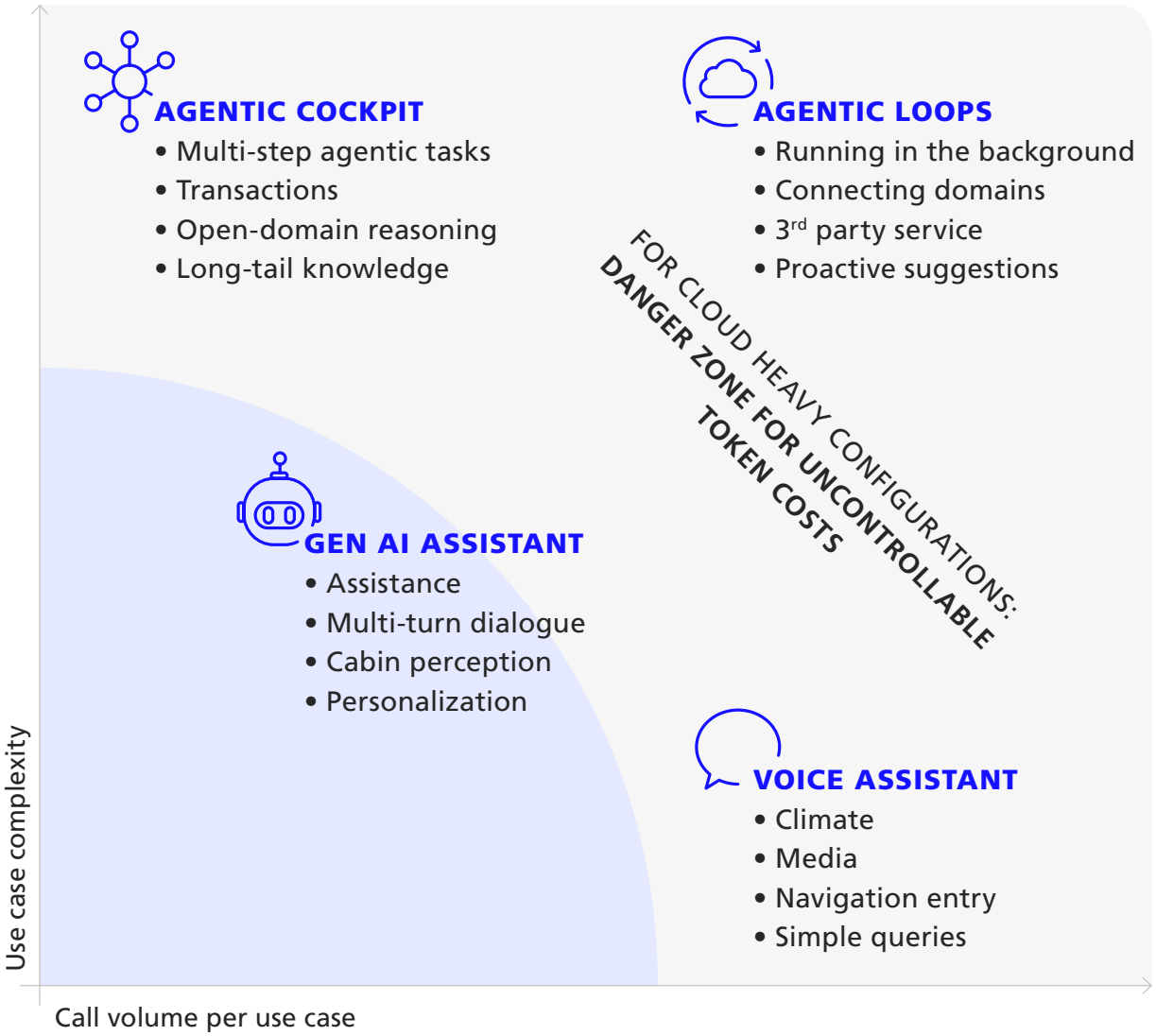


Exhibit 3: Use cases are in a business danger zone if cloud token consumption compounds

The deflation trap

The most common objection in steering committees is that token prices are collapsing, so the problem solves itself. The arithmetic does not support that.

Token price, usage frequency and interaction complexity are independent variables that move at different rates. Price can fall quickly while usage and complexity rise as a direct consequence of the experience becoming better. Success can therefore absorb the saving before it reaches the P&L.

A similar dynamic is reflected in NVIDIA's Tokenomics framework, which identifies reasoning overhead and agentic loops as key multipliers of inference demand.

Exhibit 4 shows the trap in numbers, indexed to a baseline of 100 at program launch. Its an illustrative model. The point is the structure of the arithmetic, not the precision of any single figure.

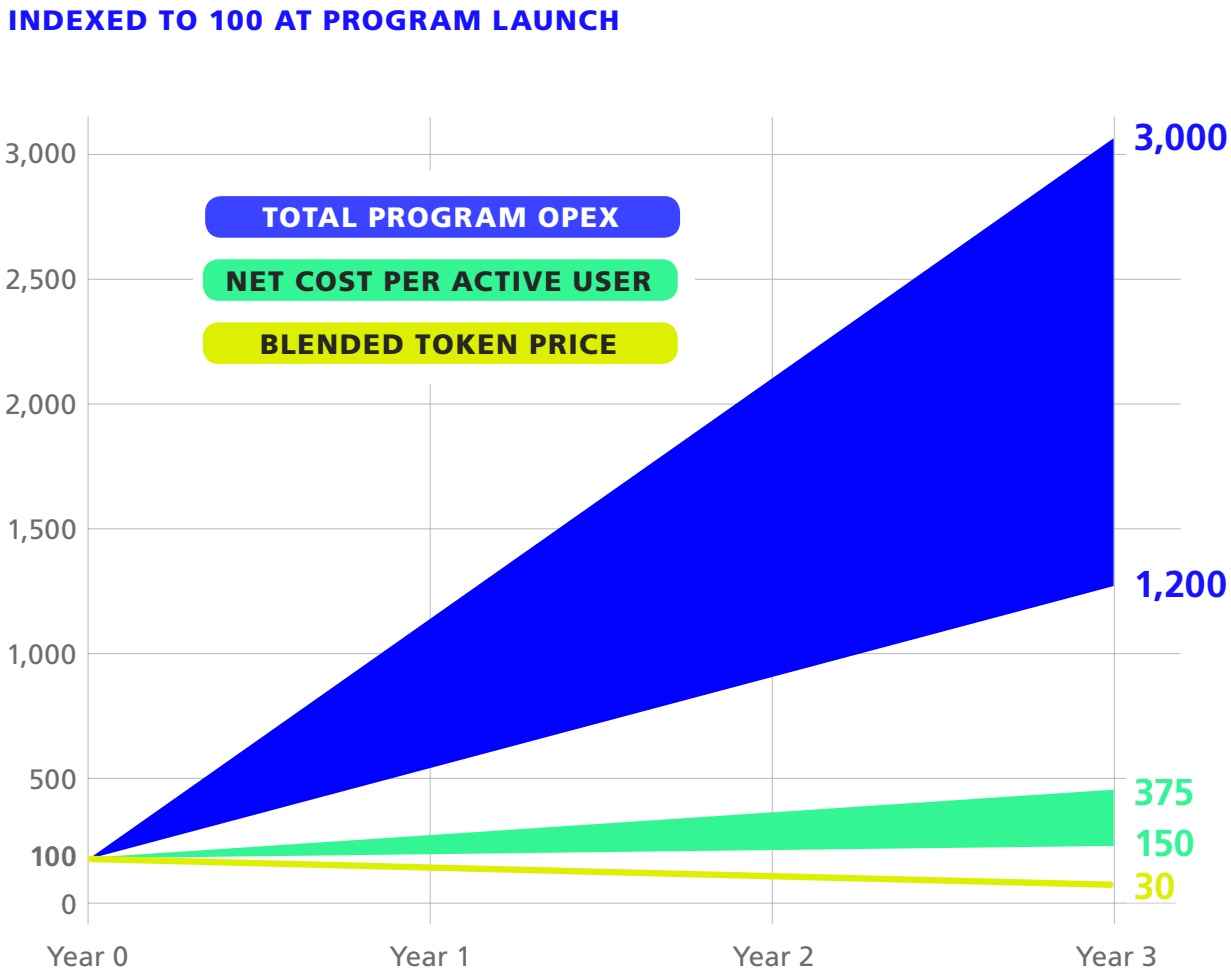


Exhibit 4: Net Cost per active User accelerates

Two conclusions follow. First, a 70 per cent price decline can be absorbed by moderate growth in usage and complexity. It's a scenario that is not pessimistic but simply describes a successful product. Second, the number that reaches the CFO is the fleet aggregate, and it is driven by rollout at the same time as the program is trying to maximize adoption.

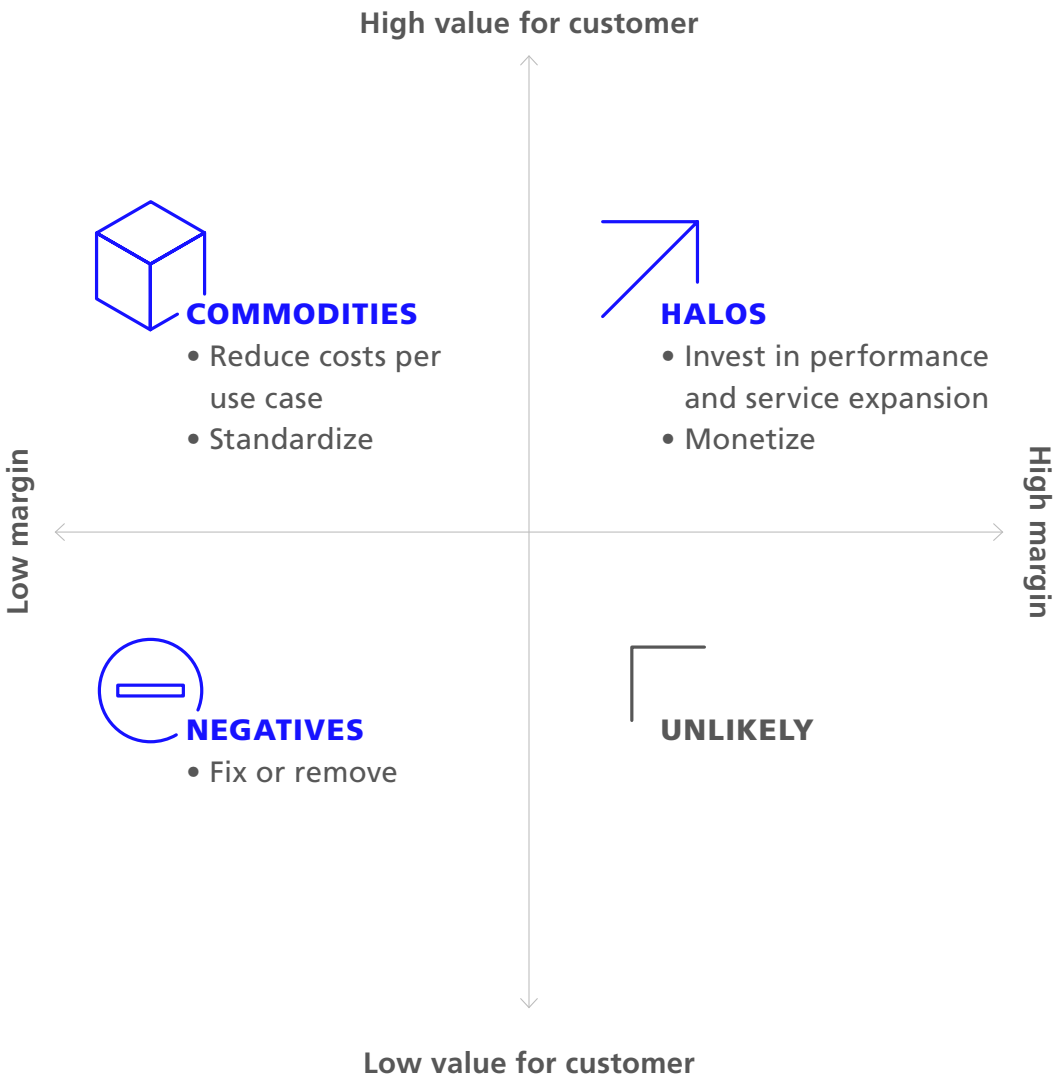


Exhibit 5: Cluster the use cases constantly to identify the halos

Run a use-case triage. Score every existing and planned use case on value, frequency, complexity and willingness to pay. The output is a routing policy: which use cases justify a large model, which belong on a small on-device model, and which should not be built at all.

Architecture changes the exposure

Cloud-heavy architectures concentrate exposure to this dynamic: every interaction incurs remote inference and data movement, and high-value use cases such as multimodal perception and proactive service execution create the largest recurring workload. The default is attractive for deployment speed and model flexibility, but variable cost scales directly with success unless routing and budgets constrain it.

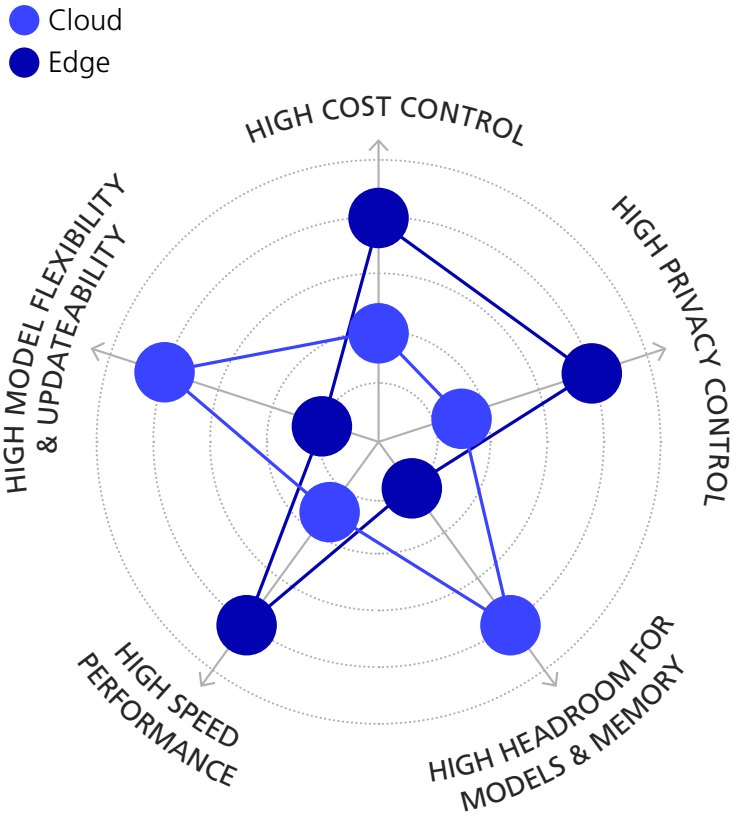
Edge execution changes the cost structure rather than eliminating cost. Local inference can reduce cloud billing, latency and data movement, but requires hardware headroom, memory bandwidth, model optimization and lifecycle support. The decision is therefore how much recurring exposure to retain and how much capability to commit in advance.

	CLOUD-HEAVY	CONTROLLED HYBRID	EDGE-HEAVY
Simple interactions (climate, media, nav)	CLOUD LLM	EDGE SLM	EDGE SLM or LLM
Multimodal cabin perception	STREAMED TO CLOUD	ON-DEVICE, EVENTS ONLY TO CLOUD	ON-DEVICE
Share of requests never leaving the vehicle	○○○○	●●●○	●●●●
Relative monthly inference cost per vehicle	●●●●	●●○○	●○○○
Recurring connectivity cost (carrier data attributable to AI)	●●●●	●●○○	●○○○
One-off engineering effort, release & lifecycle effort per model update	●○○○ MAINLY CLOUD-SIDE	●●○○ DUAL-PATH RELEASES	●●●● IN-VEHICLE MODEL RELEASES WITH VALIDATION RELEASES
Additional edge compute in BOM	○○○○	●●○○	●●●○
More flexibility	●●●●	●●○○	●●●○
Offline capability	○○○○	●●○○	●●●○
Future readiness	○○○○	●●○○	●●●○
TCO risk over lifetime	●●●●	●●○○	●○○○

Exhibit 6: The assessment of different implementation options

The balancing between cloud and edge deployment leads to different profiles as documented in Exhibit 7. OEMs and suppliers need to evaluate for which of the different vectors they want to optimize their Smart Cabin solutions. The final configuration could be influenced by desired brand positioning, financial bandwidth, local regulation or technical circumstances.

BALANCING MODEL EXECUTION AND ORCHESTRATION BETWEEN CLOUD AND EDGE



- **Edge vs. cloud balancing** result in different profiles related to costs, performance and update flexibility.
- **Agentic AI use-cases could lead to higher token consumptions,** leading to less cost control in cloud heavy setups.
- **System design flexibility is critical** for adopting the configuration based on ecosystem developments.

Exhibit 7: Balancing model execution and orchestration between cloud and edge

What it costs to build, before it costs anything to run

Most vehicle architectures, onboard and offboard are not prepared to cover the scope of Smart Cabin AI.

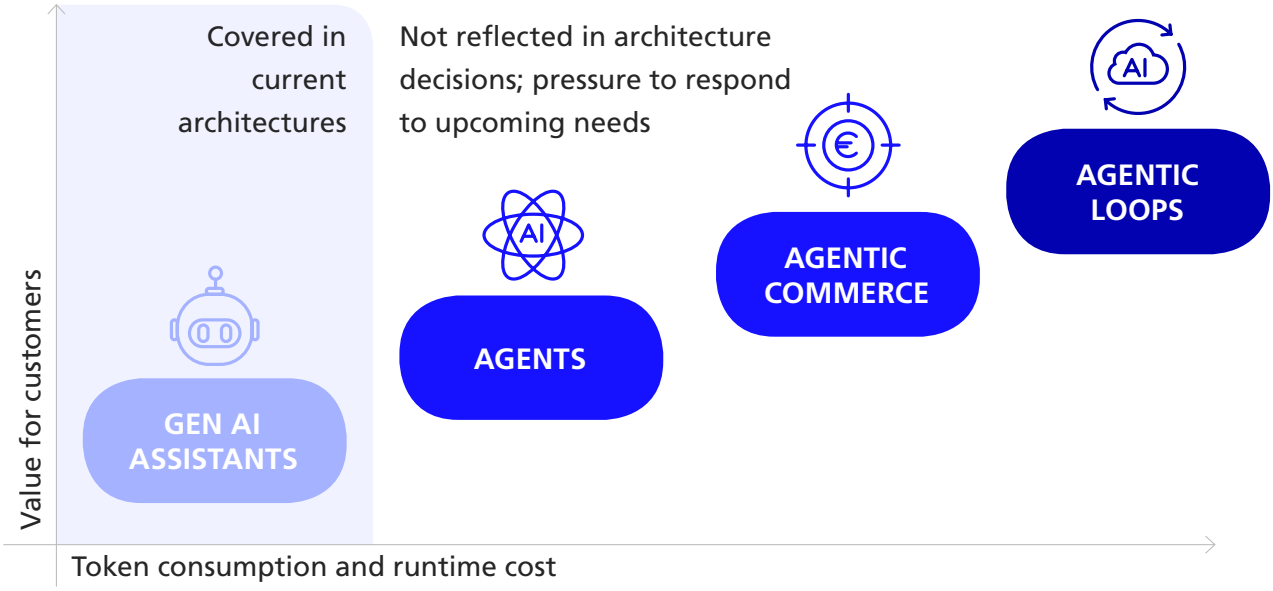


Exhibit 8: Most OEM architectures are not ready for Smart Cabin AI

Significant investments are required, independent if OEMs aim for cloud- or edge-heavy solutions. We modelled the upfront development costs and running costs that could occur during several years of usage, based on a scenario where the OEMs take the lead to building the Smart Cabin architecture. The figure refers to a single OEM taking the lead on building the Smart Cabin architecture, accumulated over a ten-year period, and excludes the domain controller bill of material. It is a model output, not a benchmark: the TCO simulation must be run per OEM and per vehicle program.

One management implication could be to fund the control layer once, then use it to steer model placement, cloud consumption and service access over the lifecycle.

An AI architecture that is controlled by OEMs determines which costs remain variable and which become budgetable.

"The cost of in-vehicle AI extends well beyond the vehicle chipset. Development costs include data, model customization, application integration, multilingual support, safety, cybersecurity, privacy and validation. [...] The largest savings come from platform reuse, efficient local inference, selective cloud use and consolidation of previously separate computing functions."

Sri Subramanian,
Global Head of GenAI for Automotive
at NVIDIA

10-YEAR COST PROFILE

TOTAL LIFECYCLE COST ~€255M

OF OFFERING AN IN-VEHICLE AI SOLUTION

42% upfront 58% recurring

WHERE THE COST SITS

ORCHESTRATION & CONTROL ~€100M

↗ largest recurring exposure

SERVICE INTEGRATION ~€75M

↗ largest CAPEX block

CLOUD MODEL SERVICES ~€60M

↗ usage-driven OPEX

EDGE INFORMATION STACK ~€20M

↗ primarily upfront

DECISION LENS

Balance upfront control investment against open-ended cloud OPEX and dependency.

Exhibit 9: A proprietary In-Vehicle AI foundation can require ~€255m in development costs over ten years

Managing and operating the orchestration layer and the cloud side of the AI-stack are the recurring cost that are often understated. Indicative ranges from MHP's cost model for automotive-grade Smart Cabin AI could provide first guidance for OEMs and suppliers. The TCO simulation must be done per OEM and vehicle program to get to a higher level of details and TCO certainty level.

"The main cost drivers will mostly be R&D to develop the functionality (big loop -> data collection -> new software features -> update), and then the cost to run AI on cloud-native systems. If proper AI-powered workflows are widespread, I see a radical decrease of cost/time to code and test updated or new functionalities."

Stefano Marzani,
Emerging Tech Leader,
Automotive and Manufacturing at AWS

The following cost levers dominate the variance, and both are decided early:

- **ARCHITECTURE CHOICE.** Full edge inference requires additional model compression, optimization, memory and integration effort compared with a hybrid deployment. That engineering cost belongs in the comparison just as much as recurring cloud inference does.

- **TEAM COMPOSITION.** The internal-versus-external mix shifts the break-even point and determines whether orchestration, data and lifecycle competence remain in the organization afterwards.

The view above excludes the domain controller bill of material. To properly determine the most suitable architecture and business case for Smart Cabin AI, we likewise need to look at the vehicle's in-car computing capacity. Cockpit silicon is scaling faster than expected. Large models with an increasing parameter performance could be deployed on the latest compute platforms from Qualcomm Technologies, Inc., SemiDrive and others. And this is state of the art in 2026 mass production, not the 2 to 3-year horizon industry roadmaps once suggested.

Yet memory, not compute, is the binding constraint. Consumer LPDDR5X module prices rose by roughly 80 to 90 per cent quarter-over-quarter in 2026 depending on the source, part of a structural DRAM shortage driven by wafer capacity reallocating toward AI-grade High Bandwidth Memory. Automotive-qualified memory follows a slower, contract-based curve, but the constraint reaches vehicle programs through lead times rather than spot prices: advanced memory is currently quoted at close to a year, against platform decisions that have to be frozen well before start of production.

To complete the picture, local AI is constrained by more than nominal TOPS. Memory capacity and bandwidth, sensor connectivity, thermal headroom and model optimization determine which workloads can move on device. Chapter 4 treats these as control and platform-architecture questions rather than as a silicon benchmark.

The architecture can determine which cost drivers remain controllable. It cannot determine how quickly customers adopt the experience or what they will ultimately pay for it. That asymmetry is the decision problem taken up in Chapter 3.

For OEMs, the strategic implication is stark. The token economy of Smart Cabin AI cannot be treated as a downstream engineering detail. It must be factored into architecture decisions from the outset, determining which tasks are resolved at the edge (where token costs simply don't apply) versus escalated to the cloud (where every additional reasoning step carries a direct, compounding cost). The hybrid edge-cloud model is therefore not only a latency and privacy safeguard, but a first-order lever for controlling the token economics that will ultimately determine whether agentic use cases are commercial. Edge-cloud hybrid architectures are no longer just a latency strategy. They are a hedge against volatile, AI-driven memory economics that now outpace NPU performance gains.

"As vehicles become increasingly AI-defined, success will depend on delivering intelligent, personalized experiences while balancing performance, cost, privacy, and energy efficiency. The future lies in a hybrid architecture where edge and cloud work together seamlessly, enabling agentic AI to understand context, continuously learn, and adapt in real time. Scalable, power-efficient compute platforms will be essential to bringing these experiences to production across vehicle segments and global markets."

Andreas Hennig,
Vice President,
Sales and Business Development,
Automotive Europe,
Qualcomm Technologies GmbH

Deciding Under Uncertainty:

Four Scenarios, One Conclusion

A business case for In-Vehicle AI cannot be settled by a single forecast. Token prices, usage, interaction complexity and architecture evolve at the same time, and they do not move in the same direction. More importantly, the uncertainty is not evenly distributed.

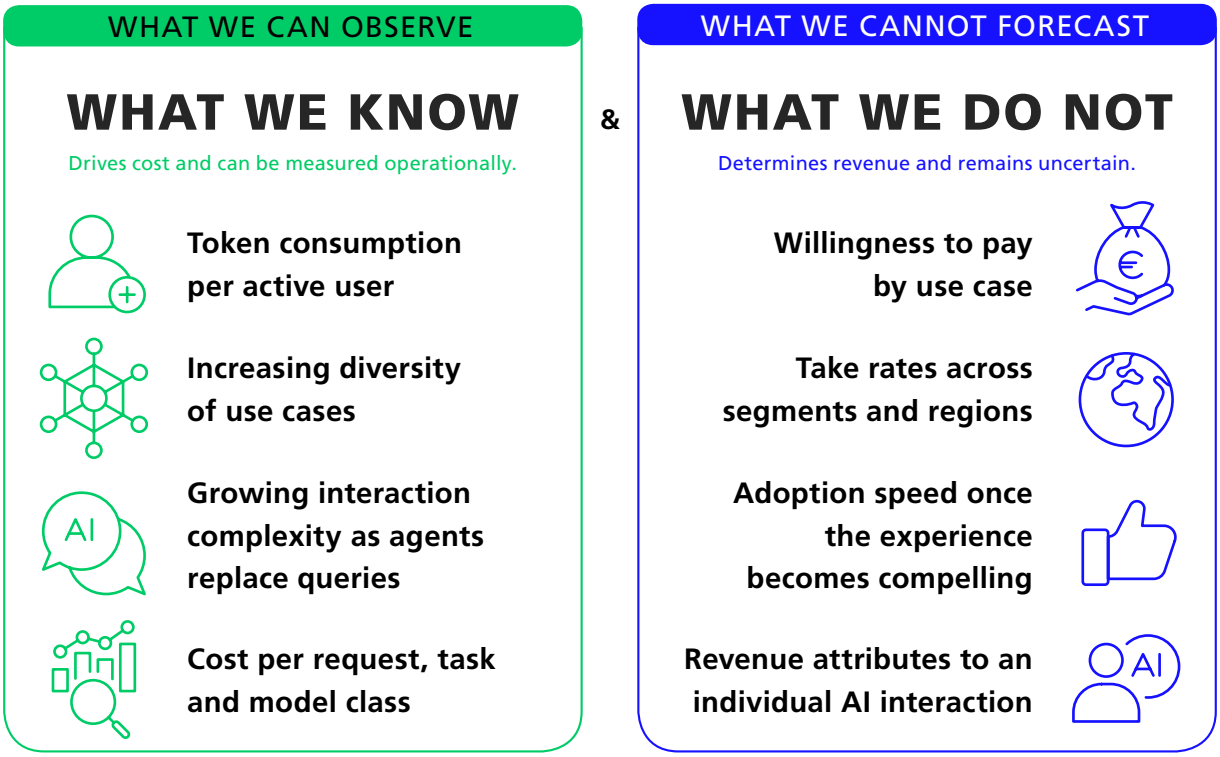


Exhibit 10: What We Know and What We Do Not

Everything on the left drives cost and can be measured operationally. Everything on the right determines revenue and remains uncertain.

That asymmetry is the actual decision problem. A business case built around a single demand forecast creates false precision. A more defensible approach is to design an architecture whose cost behavior remains steerable across different demand outcomes. The objective is not to predict the next ten years correctly. It is to preserve the ability to act when the prediction is wrong. Four scenarios are worth stress-testing.

SCENARIO 01 HAPPENS TO YOU

Deflation Wins

PRICE	FALLS SHARPLY
USAGE	MODERATE GROWTH
COMPLEXITY	BROADLY STABLE

Inference prices continue to decline, usage grows moderately and interaction complexity remains broadly stable. This is the benign scenario — but also the one the manufacturer does not control. It depends primarily on the supplier market.

COST EFFECT


Cloud-heavy default


Cost per active user falls

SCENARIO 02 HAPPENS TO YOU

Usage Absorbs Deflation

PRICE	FALLS
USAGE	RISES SHARPLY
COMPLEXITY	MORE REQUESTS OF THE SAME KIND

Prices decline, but experience becomes good enough to be used habitually. The number of requests rises faster than the savings per request. Adoption — the upside in the revenue case — absorbs much of the expected cost reduction.

COST EFFECT


Unit savings are consumed by adoption

SCENARIO 03 HAPPENS TO YOU

Agentic Escalation

PRICE	FALLS
USAGE	RISES SHARPLY
COMPLEXITY	TASKS BECOME MATERIALLY MORE COMPLEX

Multi-step reasoning, tool invocation, persistent memory, multimodal context and proactive assessment create more model calls, larger context windows and more data movement. The product can become technically better and commercially more successful while its operating-cost exposure increases.

COST EFFECT


Total exposure rises despite falling unit prices

SCENARIO 04 DECIDED

CONTROLLED HYBRID

PRICE	FALLS
USAGE	RISES, BUT ROUTED AND BUDGETED
COMPLEXITY	RISES, BUT IS ORCHESTRATED

Usage and complexity still rise, but execution is actively governed. The architecture decides which requests remain on the vehicle, which require cloud models, which model class is sufficient, and what happens when budget, connectivity or provider constraints are reached.

COST EFFECT


Edge/Cloud split, model routing, budgets and fallbacks keep the curve steerable

Controlled hybrid does not guarantee the lowest cost at every point in time. It creates a different cost structure. More capability and governance are committed deliberately; less lifecycle cost is left to be determined afterwards by adoption. In economic terms, the objective is to decouple the cost curve from the adoption curve.

Scenarios 1 to 3 can emerge without an architectural decision. They follow from supplier economics, customer behaviour and the product roadmap. Only the first is benign, and even that outcome belongs largely to the market rather than to the manufacturer. Scenario 4 is different: it exists only if model placement, routing, budgets, fallback paths and governance have been designed as decision variables.

That distinction also explains why procurement alone cannot solve the problem. Reserved capacity, volume discounts and committed-spend agreements can improve the commercial position within a scenario. They do not determine how many model calls an agentic task requires, create an alternative execution path, make a model substitutable, or prevent successful adoption from increasing variable cost.

The management decision is therefore not simply cloud versus edge. It is whether the organization retains enough control to respond when its original assumptions stop being true.

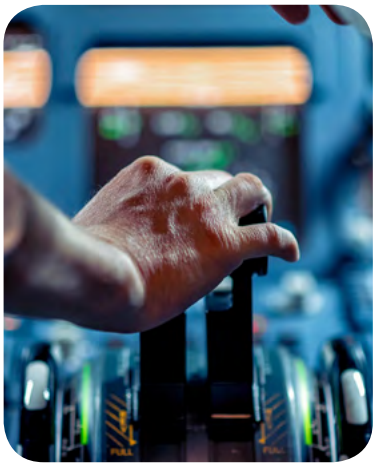
A scenario-based business case should consequently define trigger points rather than a single point forecast. Four questions should be answerable in every steering committee:

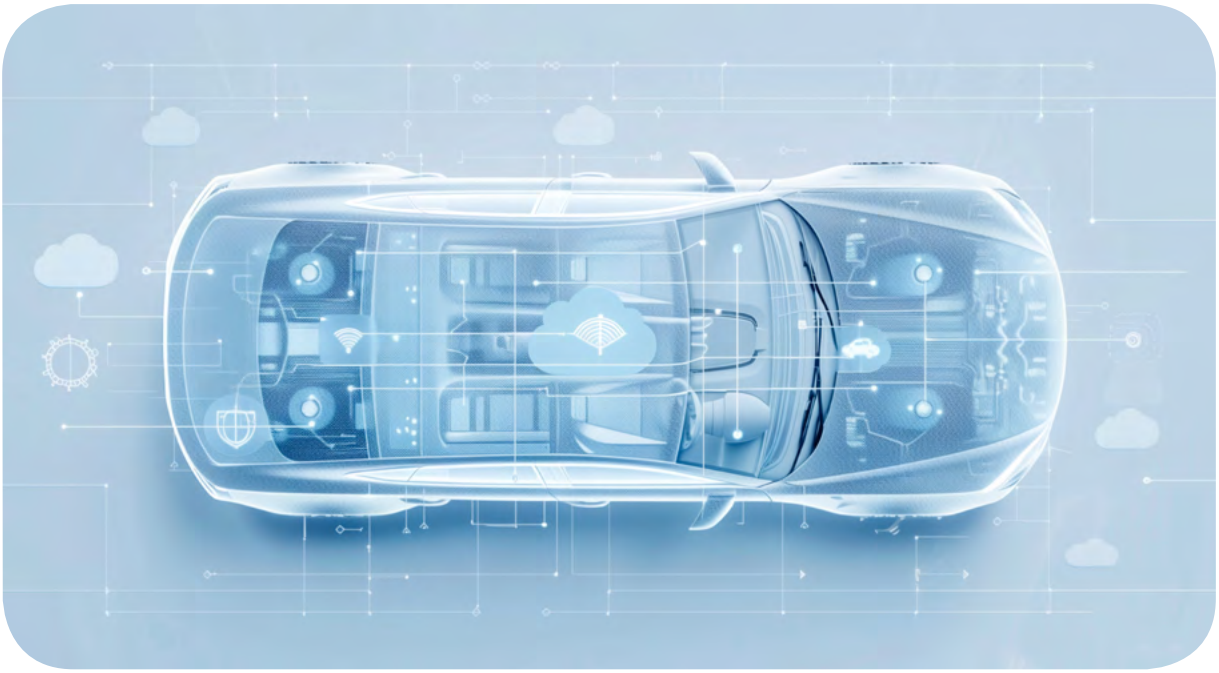
- At what usage and complexity level does our current architecture stop being economically attractive?
- Which architecture decisions are still reversible, and when will the platform freeze them?
- Which operational indicators would show that we are moving from simple usage growth into agentic escalation?
- Which revenue unknowns like willingness to pay, take rate and adoption will the next twelve months actually resolve?

The conclusion is not that cloud is wrong or that edge is always cheaper. It is that control has economic value under uncertainty. Decision makers need to consider the full lifecycle when defining the architecture and business cases of Smart Cabins. Ignoring the expected increase of token demand over lifetime, either through more usage or through more use cases, could lead to unsustainable costs.

The Control Architecture

If the previous chapter establishes that only the controlled scenario is chosen rather than inherited, this chapter specifies what "controlled" means concretely. First as a component list, then as six control points grouped in three levels.





What has to exist

Before the control discussion, the build list. An agentic in-cabin capability requires different building blocks as seen in Exhibit 11. None is optional; the delivery model for each is a decision.

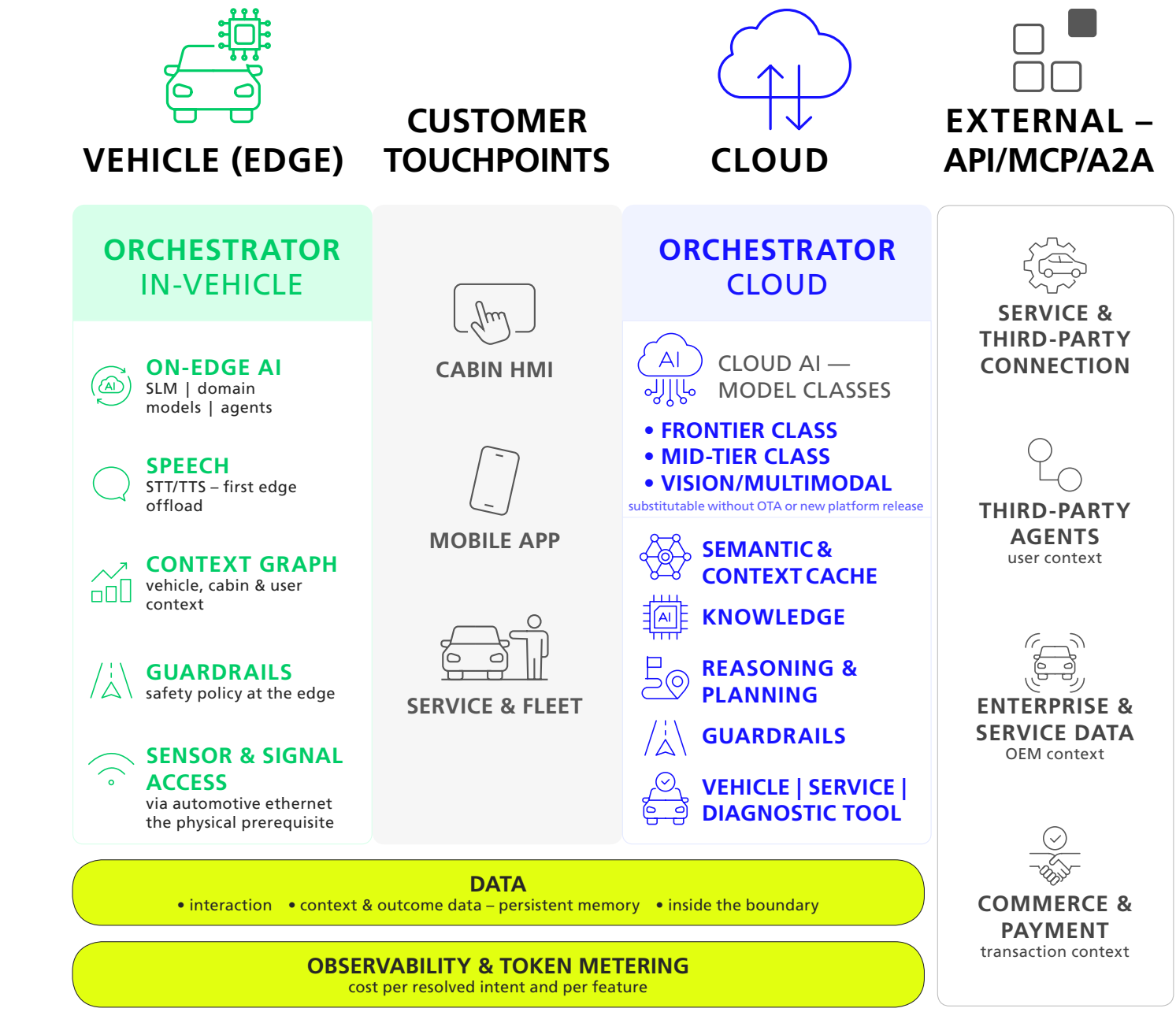


Exhibit 11: The building Blocks of a Smart Cabin AI solution

The Building Blocks

BUILDING BLOCK	FUNCTION	TYPICAL DELIVERY MODEL	CONTROL SIGNIFICANCE
Orchestrator (edge and cloud side)	Routes every request: local or cloud, small or large, own or partner	Build to keep control	The control point, as described below. An edge-heavy orchestrator is reducing the token / cloud costs close to zero.
On-edge AI	Small and large models running in-vehicle; low latency, offline capable	Partner (silicon + model toolchain) and source	Determines the offload ratio and therefore the cost floor
Cloud AI	Large models for complex reasoning and long-tail knowledge	Buy, with substitutability designed in	Substitutability is the anti-lock-in mechanism
Speech (STT / TTS)	The interaction surface; high frequency, latency-sensitive	Partner or buy	High call volume; the first candidate for edge offload
Service and third-party connection	Tool and API layer: maps, charging, calendar, payments, commerce	Partner, under OEM governance	Where the commercial interface is won or ceded
Sensor connection (in-cabin and external)	Camera, microphone and vehicle sensor data reaching the compute node, over automotive Ethernet	Build; a platform architecture decision	The hard physical constraint. Discussed below.
Data	Interaction, context and outcome data as an accumulating asset	Build; cannot be procured	The only building block with no purchase option

Two of these deserve emphasis because they are routinely underestimated in program planning.

The sensor connection is a physical prerequisite, not a software feature. An agent that cannot see cabin state and vehicle state is a chatbot in a dashboard. Getting camera and microphone streams plus vehicle signals to the compute node at usable latency requires automotive Ethernet bandwidth and a topology that supports it. This is decided in the E/E architecture, years before the AI program existed, and it is the constraint that most often determines whether an add-on compute module can deliver its theoretical benefit. Any evaluation of a retrofit path should begin here rather than with TOPS figures.

Data is the only block that cannot be bought. Compute can be procured with a known lead time. Models can be licensed and swapped. Interaction data accumulates only from vehicles in the field with a shipped experience, and the accumulation cannot be accelerated retroactively. This is the concrete content of the Success = Data × AI/SW verticalization × Compute × Portfolio scalefrom Chapter 1, and the reason the wait-and-see path in Chapter 5 carries a cost that does not appear in any budget line.

The 6 control points across 3 levels

Seven building blocks describe what to assemble. The following six control points describe what to retain authority over once assembled.

Level 1: Economic Control

Cost control. Token budgets and hard caps per use case and per user tier. Aggressive caching of repeated context. Real-time cost observability at the level of individual features, not monthly aggregate invoices. Explicit management of data transfer volumes, which in multimodal use cases can exceed inference cost.

The prerequisite is measurement. Most organizations today cannot answer what their existing voice assistant costs per active user per month, because the cost sits in a cloud commitment that is not attributed to a product. That measurement gap is the first thing to close, and it is cheap to close.

Use-case control. Not every use case justifies a premium model and a cloud round trip. Prioritize models on dimensions like value, latency, frequency, complexity and willingness to pay and route accordingly. High-frequency, low-value interactions belong on a small on-device model. Low-frequency, high-value tasks can justify the largest available model, balancing need to happen between cloud and edge deployment. Treating all requests identically is the single most expensive default in the stack.

Level 2: Technical Control

Model control. The model router, also called orchestrator, is a strategic asset, not an implementation detail. It decides per request whether to serve a small local model or a large remote one; it enables model substitution without re-integration; it makes regional model strategies executable; and it provides the fallback path when a provider degrades or a region becomes unavailable. The orchestrator is an asset to manage the business case, as cloud vs. edge token volumes could be controlled. An OEM that cannot change its foundation model without a platform release has already lost this control point.

Architecture control. The cloud-edge distribution, latency-critical routing, and the decision with the longest shadow linked to how much hardware headroom to buy today for models that do not yet exist. This is genuinely difficult. Headroom is paid for through cost, multiplied across millions of vehicles, for a capability whose requirements are not yet defined. Chapter 5 treats it as an option-pricing problem rather than a specification problem, which is the only framing that survives contact with a CFO.

Level 3: Ecosystem Control

Partner control. Governance of the API, MCP and A2A surfaces through which third parties reach the vehicle. Certification requirements for partner agents. Revenue-sharing models defined before the first partner integration, not after. The question to answer early: when a third-party agent books a service from inside your vehicle, whose customer relationship is that?

Lifecycle control. Model updates, regression testing, red teaming and release gates across a ten-year lifecycle, with cost dashboards as a standing input to release decisions. A model update is not a software update: it changes behavior probabilistically, which means the validation model that the industry built for deterministic software does not transfer without modification.

The orchestration layer is the strategic control point. It decides, per request, between local and cloud, small and large, own and partner, probabilistic and deterministic. Whoever owns it controls cost, latency, data and customer experience. Whoever cedes it becomes an integration service provider inside their own vehicle.

This is not an abstract claim. In a layered view of the In-Vehicle AI stack with multimodal interactions, context acquisition, context graph, orchestration, models, tools and APIs, vehicle gateway, safety and policy, execution, observability, the orchestration layer is the only one that touches every other layer and the only one where a decision is made rather than a capability provided. It is also the layer where the supplier landscape is least consolidated, which means it is still available.

Choosing Your Path

Four implementation paths exist. Each is right for a specific profile and wrong for the others. The purpose of this chapter is not to recommend one but to make the choice explicit and then to argue that the paths are sequence able, which changes the decision from a bet into a migration.

What it costs to build, before it costs anything to run

The compute decision is downstream of the use-case portfolio, and the portfolio sorts cleanly into three complexity tiers. Programs that begin with a TOPS target rather than with this table tend to buy either too much silicon or the wrong kind.

TIER	EXAMPLES	LATENCY REQUIREMENT	WHERE IT MUST RUN	COST CHARACTERISTIC
Low complexity	Climate, media, navigation entry, simple queries, wake word	Sub-second, high frequency	Always on Edge	High call volume, low value per call. Cloud routing here is the single largest avoidable cost in most stacks.
Medium complexity – cross domain relevance increases	Contextual assistance, multi-turn dialogue, cabin perception, personalization	Sub-second where perception-driven	Edge dominates, cloud acceptable	Cloud routing of video / DMS data is not sustainable, edge headroom to be planned
High complexity - cross domain necessary	Multi-step agentic tasks, transactions, open-domain reasoning, long-tail knowledge	Seconds tolerable for certain use cases	Edge dominates long-term, Cloud for agentic tasks and long-tail knowledge required	Low frequency, high value per task. Cloud cost here is justified.

Read the right-hand column as the routing policy: **low-complexity traffic on the edge is where the money is saved; high-complexity traffic in the cloud is where the money is well spent in "halo" use cases**. An architecture that cannot separate the two pays cloud prices for wake words.

CLOUD-HEAVY

↗ KEEP CURRENT ARCHITECTURE ↗ CLOUD-FIRST

PATH 1

IVI DOMAIN CONTROLLER » CLOUD

EDGE	CLOUD
Edge NLU	LLM (all requests)
Automatic Speech Recognition (lightweight)	World Knowledge
	VLM (streamed video)

AI-BOX

↗ ADDITIONAL DOMAIN CONTROLLER ↗ RETROFIT

PATH 2

IVI DOMAIN CONTROLLER » AI-BOX

EDGE	CLOUD
ASR + TTS (on-device)	LLM (complex tasks)
RAG + memory	World Knowledge
Guardrails	Agents
SLM + LLM (small model)	Fleet learning

UPGRADED MAIN DOMAIN CONTROLLER

↗ MAIN DOMAIN CONTROLLER ↗ EDGE BY DESIGN

PATH 3

SINGLE HIGH COMPUTE IVI DOMAIN CONTROLLER

EDGE	CLOUD
ASR + TTS (on-device)	World Knowledge & long tail
SLM + LLM (on-device)	Fleet learning
VLM (cabin perception)	
Agents + memory	

WAIT AND SEE

↗ DEFERRED DECISION ↗ WITH A PRICE TAG

PATH 4

NO ARCHITECTURE DECISION

THE PRICE

- Lost learning curve, the second program is faster only because of the first.
- Lost data access differentiating interaction data is generated only by shipping.
- Rising catch-up cost — others set the standards you must then meet.

Exhibit 12: Choosing your path: four routes to In-Vehicle AI

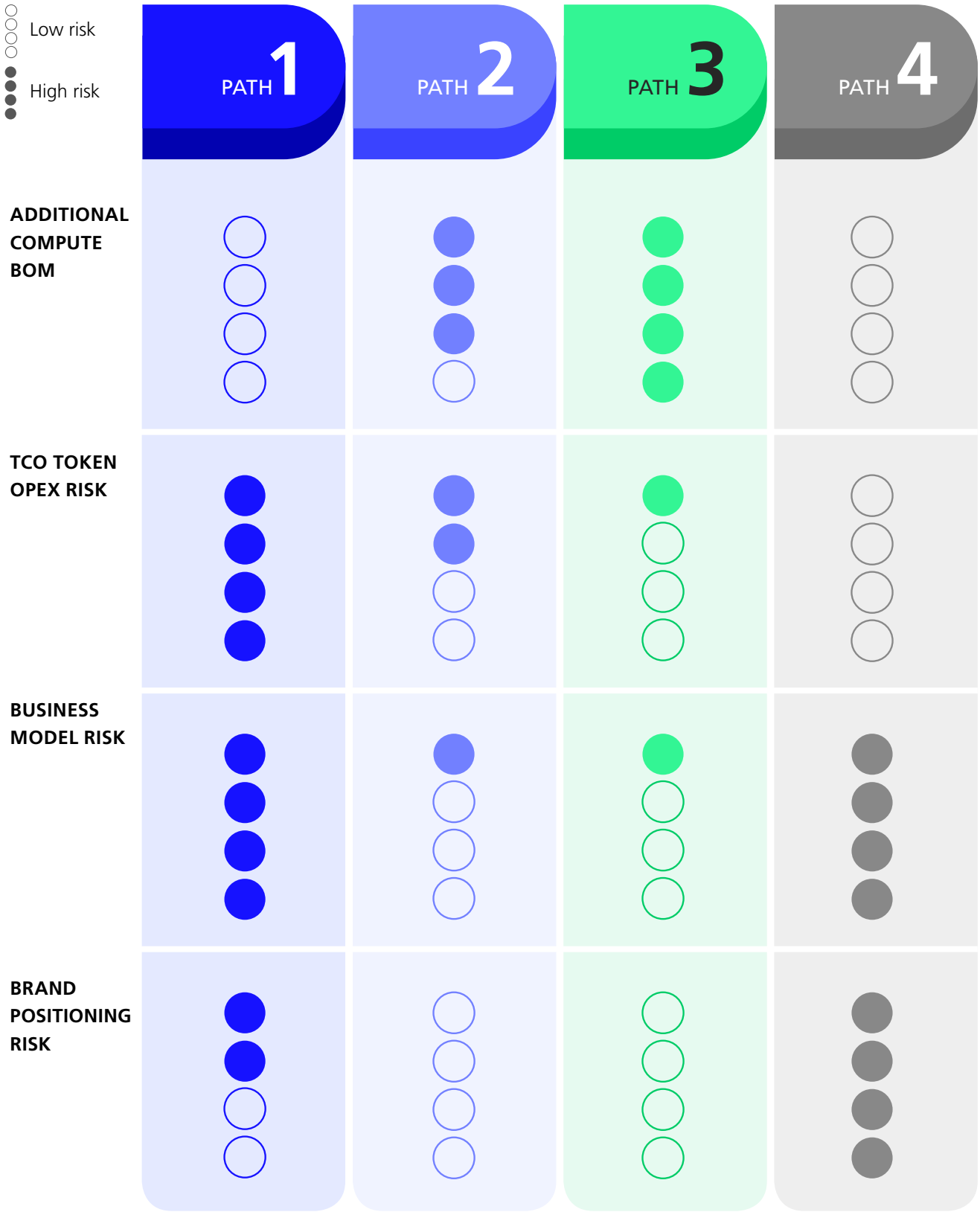


Exhibit 13: Risks related to different paths

Path 1: Keep Current Architecture and Rely on a Cloud-Heavy Setup

Inference runs in the cloud; the vehicle contributes context and interface.

Strengths. Fastest time to market. Maximum model flexibility as new models can be adopted as they release. No hardware commitment, therefore, no unit-cost exposure and no platform dependency.

Limits. Variable cost exposure is uncapped and scales with success. Latency and connectivity dependency limit the use-case set: anything requiring sub-second response or offline operation is out of scope. Use cases that require access to vehicle data and the different domains are out of scope. Data governance burden is highest, because every multimodal input leaves the vehicle. Costs scale with the increase of usage and with the amount of use cases. This path shifts the cost risk to the usage phase and that must be modelled in a TCO evaluation over the vehicle lifetime.

Right for. Piloting and learning. Long-tail use cases with low frequency. Programs with low expected usage intensity. Organizations that need to establish a capability baseline before committing capital.

Path 2 via an AI-BOX: Extension via an Additional Domain Controller

A retrofittable compute module, in architectural terms, an additional domain controller, extends an existing platform with edge inference without changing the main compute architecture. Bosch's AI-extension Platform (AIxP), launched at CES 2026, is one instance of this category, delivering roughly 150–200 TOPS as a retrofit.

The distinction from Path 3 matters and is often blurred in vendor material: Path 2 adds a node to the architecture; Path 3 replaces one. Adding a node avoids re-opening the platform but inherits every constraint of the platform it attaches to the most relevant sensor connection discussed in Chapter 4.

Strengths. A bridge architecture: it reduces cloud load and latency for defined workloads on platforms that were never designed for on-device inference. It is the only path that improves the economics of vehicles already in production or already frozen.

Limits. Variant complexity: an add-on creates a hardware variant with its own homologation, supply and lifecycle burden. Integration depth is bounded: the module's value depends entirely on how well it reaches vehicle data and how tightly it integrates with the orchestration layer. A module that runs a model but cannot see vehicle state or can't access the DMS bandwidth delivers a fraction of the benefit. Lifecycle management of a retrofit population is materially harder than of a factory-fit one. The steep increase of memory prices are limiting the sustainability of the AI-BOX business case, frontloading costs are significantly higher than at the beginning of 2026.

Right for. Running platforms with three to five years of residual life. Rapid retrofit of differentiating functions where the alternative is doing nothing until the next generation.

Path 3: Upgraded Main Domain Controller

Edge inference is designed into the main domain controller of the next platform generation or iteration as a first-class capability, with deliberate hardware headroom rather than added alongside it.

Strengths. This is the structural answer. Control over latency, data residency, unit cost and inference economics is designed in rather than bolted on. It is the only path that makes always-on multimodal use cases economically viable at volume.

Limits. Could be investment-intensive, depending on the architectural preconditions.

This path requires an architecture decision several years before start of production, when the model requirements it must serve are not yet known. Requires an explicit model lifecycle and update concept as a domain controller with ten years of field life needs a plan for the models it will run in year eight. And it requires a headroom decision that is, unavoidably, a bet.

Right for. The next platform generation or iteration. High expected usage intensity. Premium differentiation where the AI experience is part of the brand promise rather than a feature.

Path 4: Wait and See

We include this as a legitimate option with a price tag, not as a straw man.

Model capabilities and costs are moving fast enough that a decision deferred by eighteen months is made with materially better information. Hardware bought today for models that do not exist may be the wrong hardware. Some organizations are genuinely not ready to operate an AI product and would build an expensive failure.

The price. Lost learning curve, which cannot be compressed later. Keep in mind, the second program is faster than the first because of what the first taught, and that sequence cannot be skipped. Lost data access: the interaction data that trains a differentiated experience is generated only by shipping one. Growing platform dependency, because the default in the absence of a decision is someone else's stack. And a rising catch-up cost, because architectural decisions taken in the interim by others become the standards you must then meet.

The additional price tag. The first wave of AI-native cabin experiences will also define a new UI/UX benchmark. The new experiences can create brand loyalty. OEMs that delay participation risk missing a differentiation window that narrows as modern vehicle interiors, particularly in China, normalize AI-led interaction patterns.

Four decision dimensions could be used to determine the path and to build archetypes:

PLATFORM RESIDUAL LIFE:

New generation in definition, or running platform with 3–5 years remaining?

EXPECTED USAGE INTENSITY:

Driven by the use-case portfolio, not by ambition

SEGMENT AND WILLINGNESS TO PAY:

Volume or premium, what are the "halo" use cases?

INTERNAL SOFTWARE & AI MATURITY:

Can the organization develop and operate what it wants to reach?

Exhibit 14: Dimensions to select the best paths and archetype

	ARCHETYPE	PROFILE	RECOMMENDED PATH
A	**Volume, running platform**	3 to 5 years residual, cost-sensitive segment, moderate usage	Cloud, with strict use-case triage; an AI-BOX only for the two or three highest-frequency low-complexity functions
B	**Premium, running platform**	3 to 5 years of residual, high expected usage, brand-relevant experience	AI-BOX now; subject to a sensor-connection check; upgraded main domain controller in the next generation
C	**Next generation in definition**	SOP 2028 to 2030, architecture still open	Upgraded main domain controller with explicit, budgeted headroom and Ethernet topology to match
D	**Low AI internal or market maturity**	Capability gap in ML operations and orchestration	Cloud with a partner-operated orchestration layer, but own the context graph and the interaction data

Exhibit 15: Project, business and market archetypes as guidance for In-Vehicle AI implementation paths

Regional footprint does not change the path. It changes what the orchestration layer has to support, regional model sourcing, data residency and fallback when a provider or region becomes unavailable.

Sequencing is also the correct response to the asymmetry in Exhibit 10. Each step commits capital only after the preceding step has resolved one of the unknowns — the cloud phase reveals actual usage and take rates; the AI-BOX phase reveals which use cases justify local compute; only then does the domain controller decision have to be made with real numbers rather than a demand forecast.

This matters for a practical reason beyond neutrality. A sequenced path lets an organization build the orchestration layer once, on cloud infrastructure where iteration is cheap, and then re-target it as edge capacity becomes available. The orchestration layer is the asset. The compute location is a deployment decision that the orchestration layer should be able to change without re-integration.

"We expect automotive in-vehicle AI to follow a hybrid, edge-first architecture. Latency-sensitive, privacy-sensitive and vehicle-aware workloads — such as speech, cabin perception, personalization and vehicle interaction — will execute locally on platforms such as NVIDIA DRIVE AGX. This provides immediate response, offline availability and greater control of sensitive data."

Sri Subramanian,
Global Head of GenAI for Automotive at NVIDIA

Conclusion

The decisive question for OEMs is no longer whether In-Vehicle AI will scale, but whether they will control the economics, experience and customer interface when it does.

This whitepaper shows that Smart Cabin AI is not a feature upgrade, but a new operating model for the value chain. Agentic use cases change the cost physics, because usage, interaction complexity and fleet penetration can compound faster than token prices decline. Under these conditions, a cloud-only strategy may be fast to launch, but it leaves the long-term cost curve exposed to successful adoption.

The sustainable path is therefore not a binary choice between cloud and edge. It is a controlled hybrid architecture in which OEMs retain authority over the orchestration layer, model routing, token budgets, edge-cloud distribution, partner access and lifecycle governance. This control layer turns uncertainty from a passive risk into an active management capability: use cases can be routed differently, models can be substituted, cloud exposure can be capped, and local execution can be expanded as platform headroom becomes available.

Leadership teams should move from generic AI ambition to a sequenced decision roadmap: measure real usage and cost per active user, triage the use-case portfolio, define routing and budget policies, secure the data and context foundations, and decide which platform generations require additional edge headroom. The objective is not to predict the perfect architecture, but to preserve the right to adapt before architecture freezes to make the decision irreversible.

OEMs that build this capability will be able to scale Smart Cabin AI economically while keeping the customer relationship inside their own vehicle. OEMs that defer the decision may still launch AI features, but risk discovering later that cost control, data accumulation and commercial ownership have shifted to someone else's stack. In the AI-defined vehicle, the architecture decision is the business model decision.

References

Gartner. (2026). Hype Cycle for Agentic AI, 2026. Published 2 April 2026. Gartner Research

NVIDIA. (2026). AI Tokenomics: A Framework for Deploying and Monetizing Inference at Scale. NVIDIA Tokenomics Guide

Omdia and Sonatus. (2026). The 2026 SDV Reality Check: The Great Recalibration. Sonatus Resource Library

SemiDrive. (2025). Chinese chipmaker SemiDrive unveils new processors for AI cockpit. Published 24 April 2025. China Daily

SigmaIntel, cited by TweakTown. (2026). DRAM prices surged by up to 89% in Q2 2026. Published 25 June 2026. TweakTown

TrendForce, cited by Dataconomy. (2026). Mobile DRAM prices to increase by up to 83% in Q2 2026. Published 29 May 2026. Dataconomy

About MHP

MHP Management- und IT-Beratung GmbH

MHP is a management and IT consultancy headquartered in Ludwigsburg, Germany. For more than 30 years, MHP has combined deep industry expertise with technological know-how, supporting leading companies in complex strategy and IT transformations across the entire value chain. Its industry focus spans automotive and manufacturing, aerospace, defense, and energy, as well as the public sector. MHP provides strategic and operational consulting in key future-focused areas such as business transformation, artificial intelligence, SAP and cloud transformation, manufacturing and supply chain digitalization, software-defined products and operations, data-driven business models, and platform & ecosystem development. At the core is the ambition to translate technological excellence into industrial-scale solutions – fast, sovereign, and resilient. In doing so, MHP strengthens the global competitiveness of its clients. Around 4,500 employees worldwide work together to deliver on this ambition.

About Bosch

Bosch Mobility

Mobility is the largest Bosch Group business sector. It generated sales of 55,8 billion euros in 2025, and thus contributed around 61 percent of total sales. This makes the Bosch Group one of the leading mobility suppliers of technology and services. Bosch Mobility pursues a vision of mobility that is safe, sustainable, and exciting. For its customers, the outcome is integrated mobility solutions. The business sector's main areas of activity are electrification, software and services, semiconductors and sensors, vehicle computers, advanced driver assistance systems, systems for vehicle dynamics control, repair-shop concepts, as well as technology and services for the automotive aftermarket and fleets. Bosch is synonymous with important automotive innovations, such as electronic engine management, the ESP anti-skid system, and common-rail diesel technology.

Team

Alex Artamonow
Executive Sponsor
Partner, Software-Defined Vehicle
MHP Management- und IT-Beratung GmbH
alex.artamonow@mhp.com

Augustin Friedel
Whitepaper Lead
Associated Partner, Software-Defined Vehicle
MHP Management- und IT-Beratung GmbH
augustin.friedel@mhp.com

Yunus-Emre Wintergerst
Author
Senior Manager, Infotainment & Connected Solutions
MHP Management- und IT-Beratung GmbH
yunus-emre.wintergerst@mhp.com

Christian Köpp
Executive Sponsor
Senior Vice President Business Unit Compute Performance
Cross-Domain Computing Solutions, Robert Bosch GmbH
christian.koepp@bosch.com

Dr. Markus Benjamin Götzl
Whitepaper Lead
Vice President Product Management & Acquisitions Compute Performance
Cross-Domain Computing Solutions, Robert Bosch GmbH
markusbenjamin.goetzl@de.bosch.com

Auston Payyappilly
Author
Director Product Management & Acquisitions Compute Performance
Cross-Domain Computing Solutions, Robert Bosch LLC
auston.payyappilly@us.bosch.com

Michael Kram
Author
Chief Technology Officer Compute Performance
Cross-Domain Computing Solutions, Robert Bosch GmbH
michael.kram@de.bosch.com

Lars Radmacher
Author
Director Product Management Compute Performance
Cross-Domain Computing Solutions, Robert Bosch GmbH
lars.radmacher@de.bosch.com

mhp.com

bosch-mobility.com