

AI-native automotive: Redesign around outcomes, not functions

AI-native automotive begins when the unit of design shifts from tasks, predefined processes and system functions to outcomes, value flows and the decisions needed to achieve them.

Contents

03 Do not automate the org chart with agents

04 The opportunity: Improve end-to-end performance

05 The implementation: What needs to change

09 The result: What this looks like in practice

Do not automate the org chart with agents

Most automotive processes and systems were designed for human roles and conventional software. Layering AI onto that model does not make an enterprise AI-native. AI-native means redesigning how outcomes are delivered around the combined capabilities of people, software and AI.

The agent trap

Reproducing the current organization with AI actors

Recreating the functional structure of your enterprise in software is a common error.

If you build a requirements agent, a validation agent and a release agent because those teams exist today, the underlying model remains unchanged. Handoffs persist, context is duplicated and accountability stays fragmented. Automation masks the coordination burden rather than removing it.

The alternative

Start with the outcome and design the value flow required to achieve it

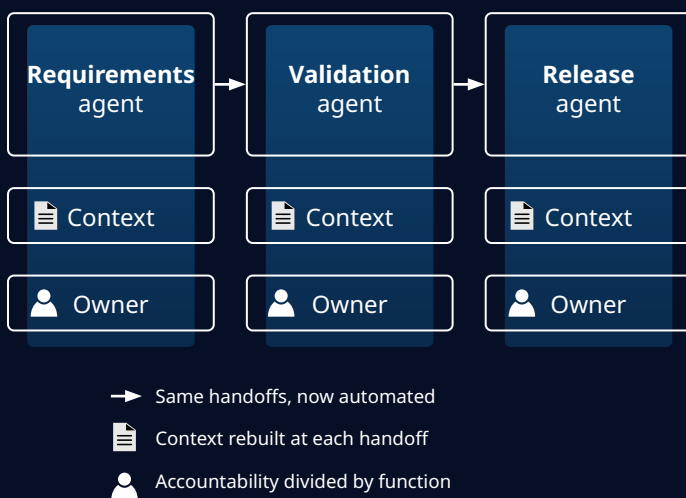
Define the outcome, redesign the end-to-end flow of decisions and actions, identify constraints and capabilities, then determine where autonomy should reside and where human accountability must remain.

In an AI-native organization, the unit of design shifts from tasks, predefined processes and system functions to outcomes, value flows and decisions. The current process is one possible route to the outcome. It describes how work happens now and is an input to redesign, not the blueprint for the target state.

The agent trap

Agents mirror the org chart

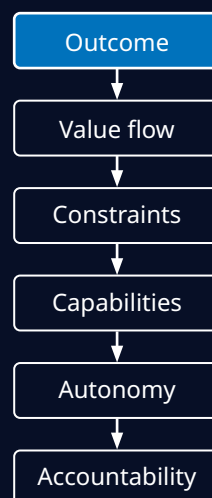
Functional agents



AI-native design

The outcome drives the value flow

Design sequence



Supported by

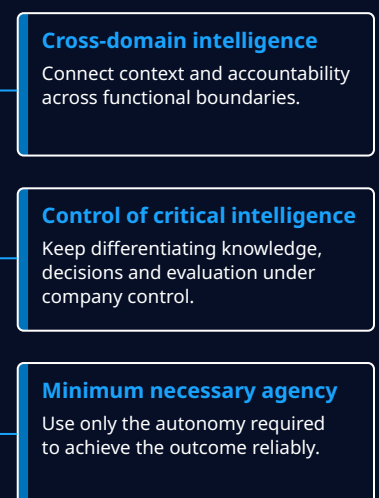


Figure 1: Agent trap vs AI-native design.

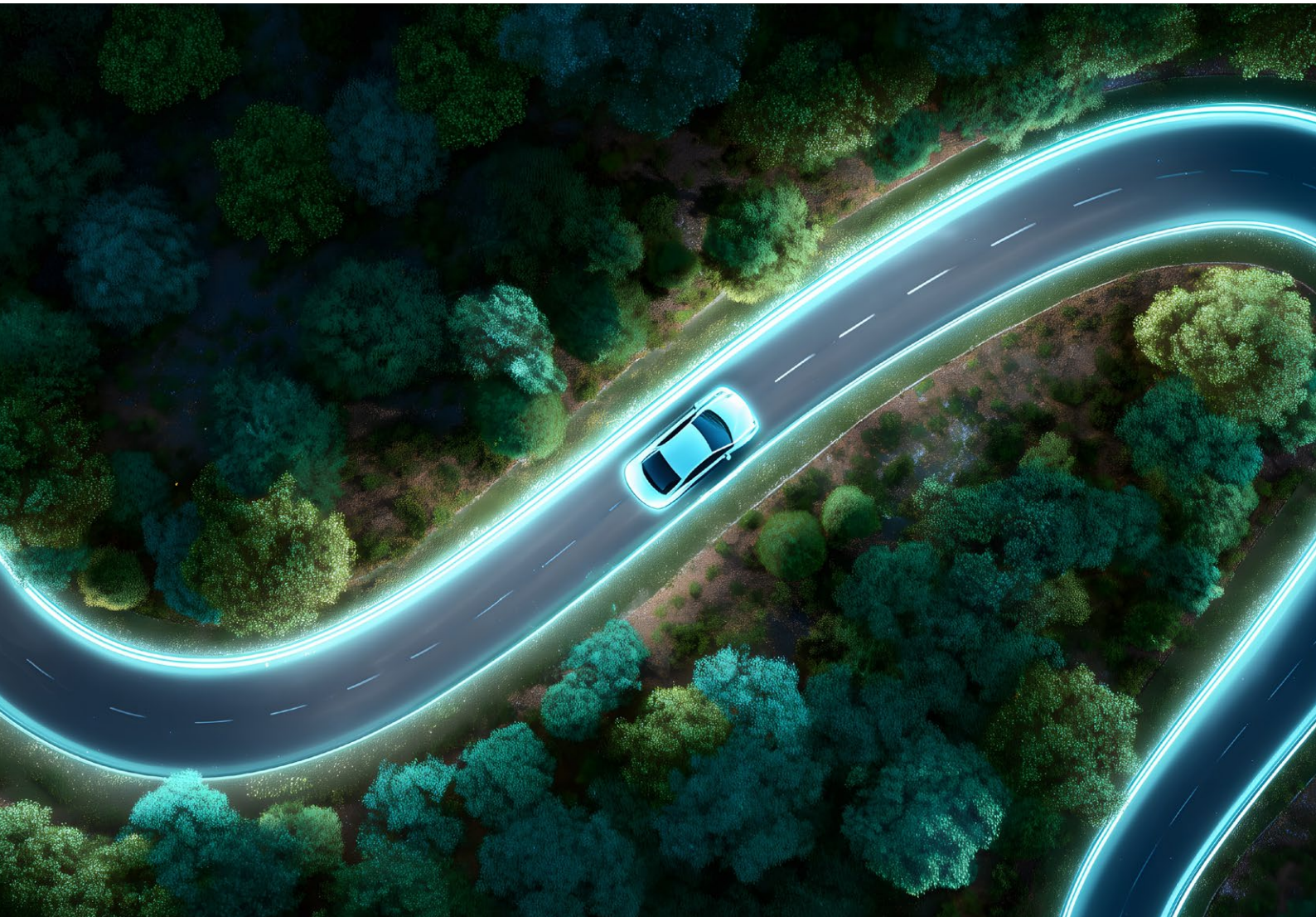
The opportunity: Improve end-to-end performance

AI can make an individual task faster without improving the related business outcome. If release readiness, fault resolution or a customer journey still stalls at the next handoff, productivity gains in individual tasks do not become an enterprise advantage.

The strategic opportunity is to improve end-to-end performance. Reduce waiting and rework, shorten the path from information to decision, and respond faster and with higher quality, all as products, customer needs and operating conditions change.

The more you rely on AI to support or make decisions, the more your competitive advantage will depend on your ability to authorize those decisions safely and at scale. You therefore need to retain control of the knowledge, evaluation criteria and decision rights that influence how your products, services and brands behave.

Becoming an AI-native automotive company requires decisions about your technology and operating model, including where intelligence sits, how decisions are made and what remains under the organization's control.



The implementation: What needs to change

1. Start with the outcome and value flow, not the predefined process

Most software asks people to translate what they want into application steps. Business processes do the same: complete a task, hand over the output and wait. When exceptions or new information change the next step, people become the integration layer, connecting information to determine what should happen next.

Outcome-led design works from the result backward:

- The outcome defines what good looks like
- The value flow sets the decisions, information and actions required end to end
- Constraints define what must not happen
- Required capabilities show what the flow needs to execute
- Autonomy and accountability are part of the target design

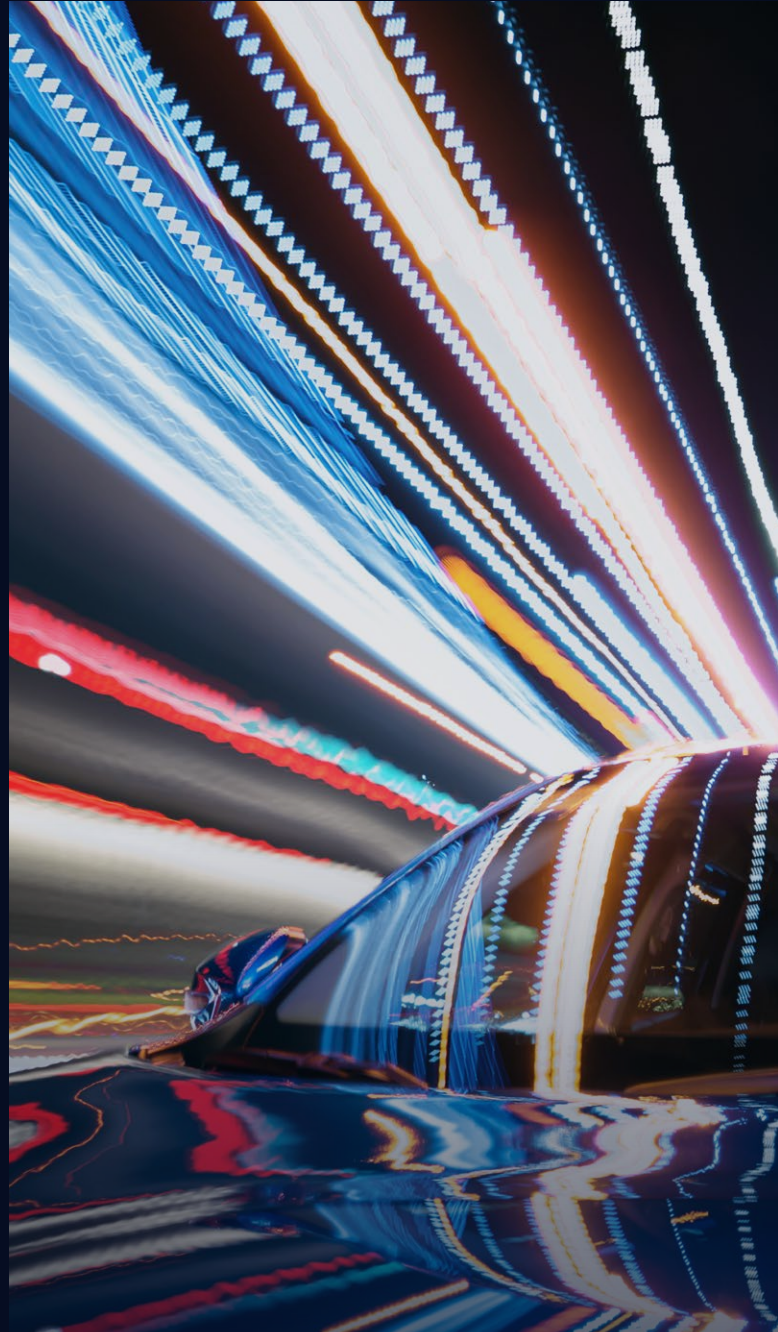
Only then should decisions be made about what belongs in deterministic software (governed by fixed logic), maintained knowledge (verified, up-to-date information), models, human expertise or agents.

Outcome-led does not mean making every step dynamic

Where logic can be specified reliably in advance, keep it deterministic. Use AI where interpretation, investigation or judgment are needed.

“Carmakers will compete on the outcomes they deliver. The edge goes to those who combine human expertise and AI across domains, with clear accountability.”

Ralf Malter, Managing Director Global Automotive, Business Consulting and AI, NTT DATA



2. Create cross-domain intelligence

Outcomes and their value flows rarely fit within one system or function. Cross-domain intelligence brings together information and expertise from different business functions so decisions account for how a change affects the whole business, not just one team.

For example, accepting a component substitution may depend on supplier availability, manufacturing constraints, compatibility, test results and cost. Intelligence in one function cannot determine whether the substitution is acceptable; it needs information from all the functions affected.

Cross-domain intelligence must therefore retrieve information from connected systems while preserving meaning and context.

This means automotive companies need a shared understanding of how requirements, components, configurations, tests, defects and other relevant information relate to one another, including their source and current state. A component identifier must refer to the right revision; a test to the relevant configuration; a suspected cause must not become fact through repetition.

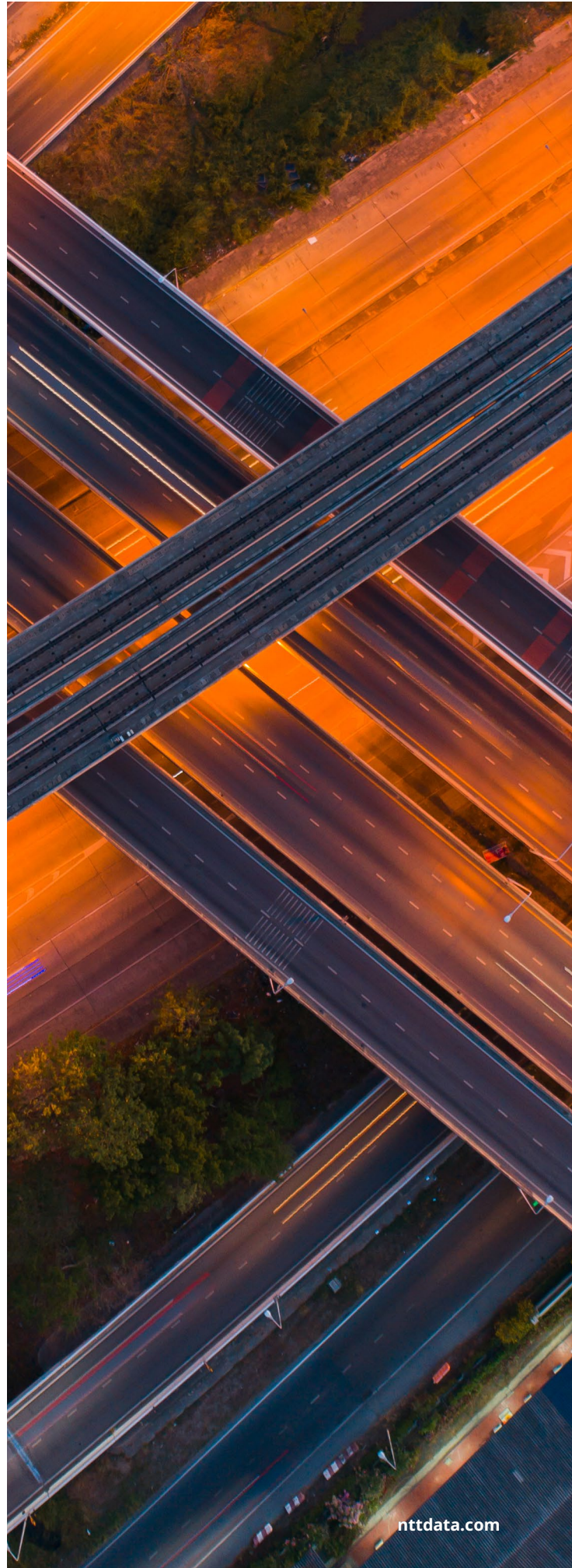
The Connected Vehicle Systems Alliance (COVESA) describes requirements, modeling, testing and validation tools as silos and is pursuing lifecycle interoperability.¹ The harder challenge is creating the semantic and decision context that allows intelligence to reason across those boundaries without losing important distinctions in information.

Governance must span the same boundaries as the intelligence

That does not mean centralizing every decision. The outcome needs clear accountability, while authority can remain distributed across the functions qualified or required to exercise it. Decision rights, escalation paths and responsibility for cross-functional trade-offs the need to be explicit.

The National Institute of Standards and Technology (NIST) AI Risk Management Framework emphasizes clear rules for overseeing AI, including governance, roles and accountability.²

Cross-domain intelligence without a corresponding model for accountability and decision rights produces faster reports, not better outcomes.



¹ COVESA. [Introducing the COVESA Generative AI Project](#). December 15, 2025.

² National Institute of Standards and Technology. [Artificial Intelligence Risk Management Framework \(AI RMF 1.0\)](#). NIST AI 100-1, January 2023.



3. Retain control of differentiating intelligence

You automotive company does not need to build every model, platform or service itself. But you do need control over the intelligence that defines product and service behavior: authoritative knowledge and data rights, decision logic, evaluation criteria and, where relevant, brand behavior.

How should the organization respond when information is incomplete, an option is unavailable or a fault is unresolved? These choices affect the product, service and customer experience.

Permissions must be explicit

Reading a vehicle record, recommending an action, committing money and changing software are different authorities. The Open Worldwide Application Security Project (OWASP) recommends limiting functionality, permissions and autonomy, with authorization enforced outside the model's judgment.³ A recommendation cannot grant itself permission to proceed.

Some intelligence may sit in models, some in maintained knowledge and some in enforceable rules. The practical test is whether you can inspect and change consequential decisions, correct authoritative knowledge, evaluate behavior, revoke authority or stop the service, and change providers without losing what differentiates your business.

³ OWASP. [LLM06:2025 Excessive Agency](#). OWASP Top 10 for Large Language Model Applications, 2025.

4. Use the minimum agency necessary

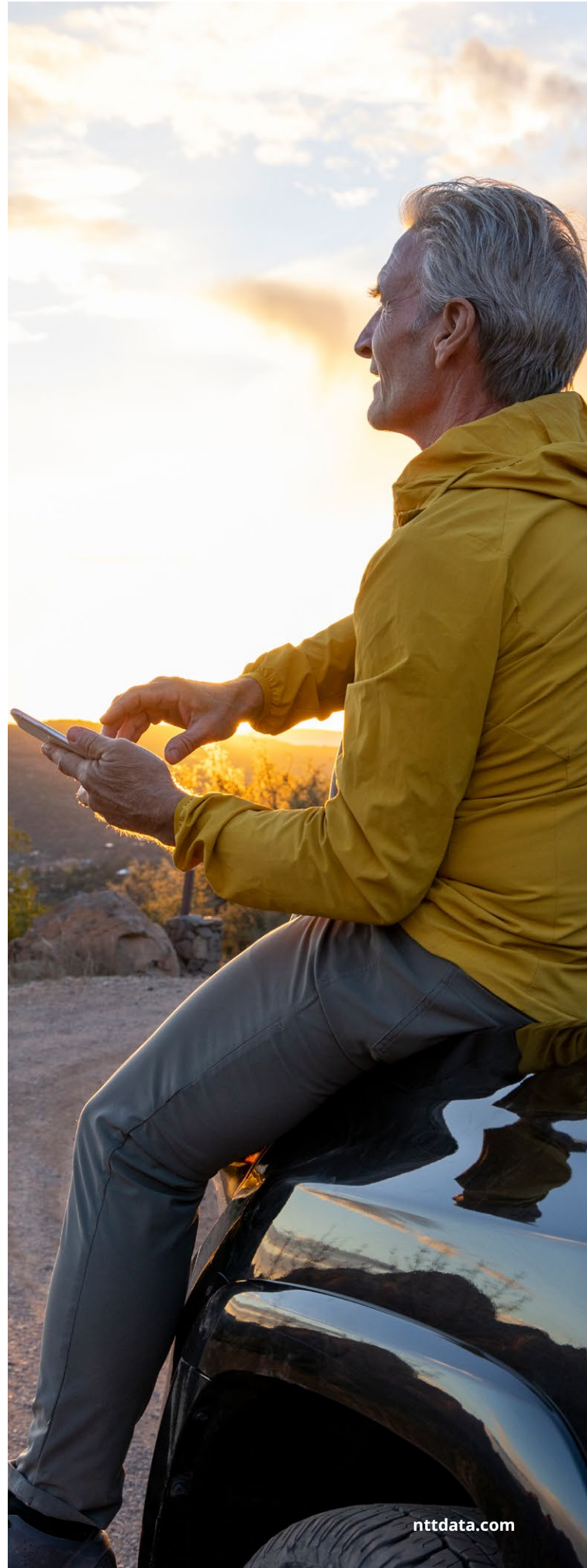
AI-native does not mean putting the most capable model or an agent behind every step. Different steps of the same value flow may require different forms of intelligence:

- Deterministic services for known logic
- Bounded-decision models for defined judgments
- Large language models for interpretation and more open-ended reasoning
- Agents for dynamic planning and tool use
- Human judgment where it is required

Use only as much autonomy as the outcome requires

Agency is a design choice. Compose the right intelligence for each part of the flow rather than forcing everything through one model or agent.

Multiple agents can help when specialist work can genuinely run independently, requires distinct permissions or benefits from parallelism. But they also introduce coordination, monitoring and failure modes. NeurIPS research identifies problems involving system design, interagent alignment and task verification.⁴ The objective is not to minimize agent count for its own sake, but to use only the autonomous complexity needed to achieve the outcome reliably. Each additional agent should earn its place by improving the outcome enough to justify the cost and complexity it introduces.



⁴ Cemri, M., et al. [Why Do Multi-Agent LLM Systems Fail? Advances in Neural Information Processing Systems 38, Datasets and Benchmarks Track, 2025](#). December 2025.

The result: What this looks like in practice

These four changes apply to all automotive companies. Engineering and product development show how the logic crosses original equipment manufacturer (OEM) and supplier boundaries; customer experience shows how it changes the relationship between an OEM and its customers.

Engineering and product development

Supplier-driven component change

The impact of a supplier-driven component change can extend across requirements, architecture, configurations, tests, defects and the release baseline.

An AI-native approach will:

- Connect that context throughout the lifecycle
- Identify what is affected and where information is missing
- Trigger permitted checks
- Distinguish what is confirmed from what is inferred or uncertain

Release readiness can then be assessed continuously, while important decisions remain behind clear approval gates.

Case study: NTT DATA and Continental

NTT DATA's work with Continental provides an example of this. An AI-based requirements-engineering tool was reported to reduce specification-analysis effort by up to 80%. While the case study focuses on requirements engineering, larger opportunity is to connect such gains across the product-development value flow rather than move the bottleneck downstream.

Customer experience

Automotive OEM and partner services

You're moving between meetings on a long, demanding day. One meeting runs over and another is waiting. Through services you have chosen to connect to your vehicle and OEM account, the system can see that your meeting has ended later than planned. It handles routine tasks you have already authorized, such as paying for parking and cooling the vehicle before your return. As you get back into the car, the assistant asks whether the delay affected your plans. When you mention that you skipped lunch, it combines that information with the hot weather, your schedule and permitted vehicle signals, such as location and fuel level, to suggest you stop for a snack and a cold drink at a place you already like, with the option to let a colleague know you'll be a few minutes late.

This is different from putting a conversational assistant on top of existing journeys. The experience can be composed around context and intent across vehicle, OEM and partner capabilities. What must be designed in advance is not every interaction, but the knowledge, constraints, decision logic, permissions and brand behavior that govern what happens.

The best AI experience may be the one the customer barely needs to invoke: it knows when to act, when to suggest and when to ask.

“Engineering and customer experience are becoming one value flow. What customers need from their car will decide what engineers release next.”

Sai Sekar, Senior Vice President, Global Automotive and Global Business Enablement, NTT DATA

Migrate incrementally; design end to end

1. Start with one high value outcome

Choose a strategically important outcome that crosses functions or systems.

Define what success looks like, then work backward to the decisions, information and actions required to achieve it. Use current processes to identify what to keep, change or remove — not every step or handoff has to be part of the new design.

2. Design the operating model around the outcome

Define how people, deterministic software and AI should work together to deliver it.

Make constraints, capabilities, decision rights, accountability and human involvement explicit. Connect the required context across domains while retaining control of critical knowledge and using only the autonomy the outcome actually requires.

3. Prove it end to end, then reuse what works

Validate the operating model with an implementation that includes the decisions, controls, integrations and evidence needed to deliver the outcome.

In production, measure whether the end-to-end business outcome improves. Use what you learn to refine the model, then reuse the capabilities, controls and patterns that can accelerate the next outcome.



Visit nttdata.com to learn more.

NTT DATA is a \$30+ billion business and technology services leader in AI and digital infrastructure. We accelerate client success and positively impact society through responsible innovation. As a Global Top Employer, we have experts in more than 70 countries. NTT DATA is part of NTT Group.



