



Qorix Developer

AI-Assisted Engineering for Deterministic SDV Platforms

Less manual configuration, fewer integration errors, faster iteration. Qorix Developer removes the hidden cost of fragmented SDV toolchains, without forcing a migration away from what OEMs already have.

The Challenge

As SDV architectures grow in complexity, integration challenges increasingly originate before runtime, during configuration and validation.

Fragmented toolchains, inconsistent stack definitions, and late discovery of integration conflicts slow down platform programs and increase program risk. Traditional configuration tooling often amplifies this complexity, creating higher support effort and longer time-to-market. Modern SDV programs require faster configuration cycles, CI enablement, and scalable software factory workflows.

The Solution

Qorix Developer is built around three principles: one unified model, full configuration traceability, and zero disruption to existing workflows. Everything else follows from that. Qorix Developer brings AI-assisted engineering to configuration and validation, without compromising determinism or control. One tool, one model, across the full SDV middleware stack, catching errors before they reach integration, not after.

AI assists the workflow. Final artifacts remain transparent, traceable, and engineer-controlled.

Use Case 1: Configuration Quality and AI

The Challenge

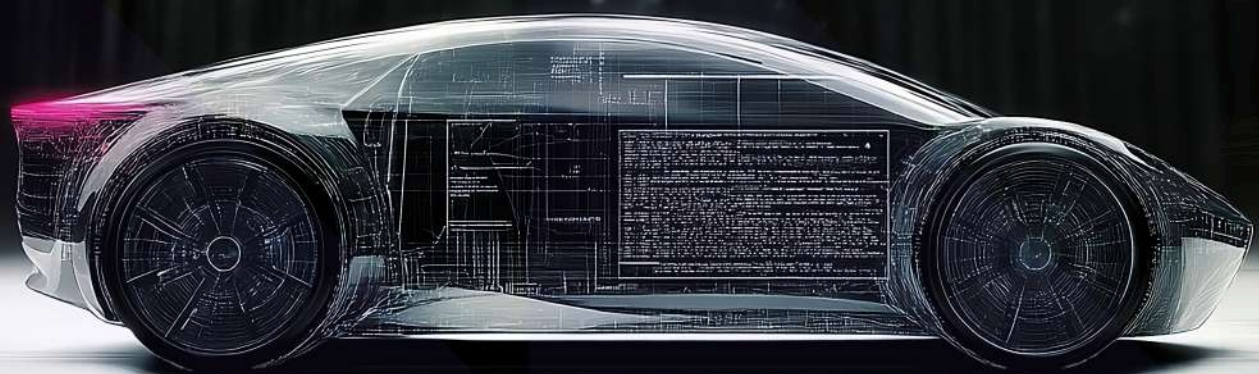
Configuring middleware across a growing SDV stack today means multiple tools, multiple formats, and errors that only surface once integration has already started, expensive to trace, slow to fix.

The Solution

One tool, one model, for the whole stack. AI Assist runs on a curated, Qorix-built know-how database, not a generic model guessing at values. That's what catches a broken reference or a missing dependency in seconds, not weeks later during integration.

Why It Matters

Fewer tools and licenses, less retraining every time a new engineer joins. Every change stays traceable, who made it, why, and when, automatically, which matters for ISO 26262-relevant work. And it's already proven in the field: the same approach configured Bootloader for a real Tier-1 development program, twice, right the first time.





Use Case 2: Integration, Teamwork and Flexibility

The Challenge

Switching middleware stacks usually means vendor lock-in, because existing configuration work doesn't carry over. Multiple engineers configuring the same middleware stack in parallel usually means silent conflicts that surface late, and tools that don't fit a customer's existing CI/CD pipeline just add integration work on top.

The Solution

Existing configuration work imports directly, nothing rebuilt from scratch. Every configuration change is tracked like code. If two engineers touch the same interface, Qorix Developer flags the conflict the moment it happens, with full visibility into what changed and by whom. And because it accepts multiple input formats, it plugs directly into the CI/CD pipeline a customer already runs, instead of forcing a new one.

Why It Matters

Conflicts get caught before they cost a sprint, not after. And teams keep the CI/CD setup they've already invested in, no rip-and-replace required.

At a Glance

Qorix Developer

- ▶ Transforms engineering intent into validated configuration artifacts
- ▶ Detects configuration conflicts early
- ▶ Ensures cross-stack consistency across Classic, Adaptive, Bootloader, Performance, and Classic Lite
- ▶ Supports deterministic and reproducible outputs
- ▶ Provides explainable AI recommendations for engineering decisions
- ▶ Enables CI-driven configuration validation and automated integration workflows
- ▶ Supports faster onboarding and migration from legacy tooling environments

What this means for your program

- ▶ Reduced configuration complexity across large-scale SDV programs
- ▶ Fewer late-stage integration issues
- ▶ Improved traceability across SDV stacks
- ▶ Scalable deployment across Tier-1 development programs
- ▶ Shortened time-to-market through CI-enabled configuration workflows
- ▶ Lower tooling support effort and reduced integration cost
- ▶ Simplified migration from existing platforms and configuration tools