Hardware-Enforced Admissibility Boundaries for Automotive AI Execution: A Structural Governance Architecture

Alexanja Senke

Independent Researcher
Interlink Bridge, Rendsburg, Germany
AlexanjaGT5S@proton.me
https://interlink-bridge.com

Abstract. The integration of AI inference systems into safety-critical automotive functions introduces a governance problem that ISO 26262 and SOTIF do not fully address: the pre-execution admissibility of AI-generated action proposals. Current architectures validate inference outputs after computation through monitoring, plausibility checks, and override logic. At ASIL-C and ASIL-D, this post-execution approach cannot satisfy the independence requirements that ISO 26262 demands for safety monitors, because the monitoring function and the monitored function share the same execution environment.
This presentation introduces *execution-bound governance*: a pre-execution admissibility layer that determines whether a state transition is structurally allowed before an inference result reaches the actuator command layer. The central invariant is: if a transition is inadmissible, the execution path does not exist. The architecture is grounded in a formal spectral stability criterion for AI reasoning systems and maps directly onto ISO 26262 constructs including independent safety monitor, safe state, hardware watchdog, and E2E protection. We further address the robustness dimension—signal integrity, tamper detection, and fail-closed defaults—that distinguishes logically correct enforcement from physically trustworthy enforcement. Implementation-specific enforcement logic is intentionally withheld.

Keywords: Automotive AI Safety · ISO 26262 · SOTIF · Hardware Enforcement · Admissibility · Formal Methods · ASIL · Execution-Bound Governance

1 The Governance Gap

The dominant paradigm in automotive AI safety is reactive. An AI inference engine produces an output, and a supervisory layer evaluates that output against safety constraints after the fact. While this supports traceability, it does not prevent the formation of invalid system states.

At ASIL-C and ASIL-D, ISO 26262 requires that safety monitors be independent of the monitored function (Clause 5.4.7) [1]. A software monitor executing in

the same environment as the AI inference engine cannot satisfy this requirement. Beyond the independence problem, two further structural limitations apply:

1. Probabilistic AI outputs cannot be certified safe at ASIL-D through post-execution validation alone—a validated incorrect inference remains incorrect.
2. In multi-step agentic systems, a single inadmissible inference propagates through the reasoning chain before any validator catches it.

SOTIF (ISO 21448) [2] extends this problem. AI triggering conditions are by definition within a model's output distribution but contextually inadmissible. No pre-execution structural mechanism currently prevents their execution. The gap is not one of missing test coverage—it is architectural.

2 The Architecture: Execution-Bound Governance

We propose a three-layer architecture that enforces admissibility before an inference result reaches the actuator command layer.

Layer 1 — Admissibility Engine. A software-layer pre-execution classifier operating as middleware before every model or tool call. It evaluates structural admissibility against formally defined criteria—not inference content, correctness, or optimality. Output is one of five governance states: ADMISSIBLE, PROBABILISTIC, REQUIRES-BOUNDARY, NON-GOVERNED, REJECT. The classifier draws on an assumption graph, external dependency detection, chain-taint propagation, and pattern-based structural analysis.

Layer 2 — Signal Encoding. Translates the software governance status into a hardware-readable bit vector. Authorization signals (`authority_bit`, `presence_bit`, `commit_bit`) are sourced externally from the inference engine—from authenticated system context, user identity, or supervisory infrastructure. This separation is critical: the model cannot grant its own execution authority.

Layer 3 — Hardware Enforcement Kernel (CoChip). A deterministic logic block receiving the governance bit vector. No intelligence, no learning, no interpreta-tion. A fixed priority function maps inputs to exactly one of four outputs: AL-LOW, DELAY, THROTTLE, HALT. The HALT path is physically non-bypassable at the silicon level. The hardware execution path to ALLOW is closed when HALT is asserted—this is a physical property, not a software policy. The enforcement layer can be realized as a hardware-adjacent integrity kernel operating independently from the AI inference execution environment, with deterministic fail-closed behavior and bounded latency guarantees. This separation ensures that execution authority is not derived from the inference system itself, but from an externally enforced integrity domain.

The end-to-end flow is:

$$\texttt{[AI Request]} \rightarrow \texttt{[Admissibility Engine]} \rightarrow \texttt{[Signal Encoding]} \rightarrow \texttt{[CoChip]} \rightarrow \texttt{[Actuator Command Bus]}$$

Execution reaches the actuator bus only if ALLOW is asserted.

3 Formal Grounding

The admissibility criterion rests on a spectral stability model for AI reasoning systems. Let the system state evolve as:

$$\frac{dx}{dt} = R_t(x_t, u_t) - \Gamma_{\text{adj}}(\Delta\rho) \tag{1}$$

where R_t is the reasoning operator at time t, u_t is external input, and Γ_{adj} is an active damping term that engages when spectral deviation $\Delta\rho$ is detected.

Definition 1 (Structural Admissibility). *A reasoning trajectory is structurally admissible if and only if the spectral radius of R_t satisfies:*

$$\rho(R_t) = \sup_i |\lambda_i(R_t)| \leq 1 \tag{2}$$

where λ_i denotes the i-th eigenvalue of R_t.

When this bound is exceeded, the system has entered a non-admissible state—independent of whether the inference output appears locally correct. This is the standard discrete- and continuous-time stability criterion (Lyapunov sense) [6]; it is intentionally model-agnostic and does not require access to model weights, activations, or internal representations. It operates on the observable dynamic properties of the reasoning trajectory.

Structural load accumulates monotonically:

$$L(t) = L_0 + \int_0^t \varphi(x_\tau, R_\tau)\, d\tau - S(t), \quad \varphi \geq 0 \tag{3}$$

where $S(t)$ is selective shedding of unstable side-paths. No decay term is permitted—temporal irreversibility eliminates state-reset exploits. The combined halt condition is therefore:

$$\text{HALT} \iff \rho(R_t) > 1 \quad \text{OR} \quad L(t) \geq L_{\max} \tag{4}$$

This criterion is model-agnostic: applicable to transformer-based models, reinforcement learning agents, and hybrid systems without access to model weights or activations.

4 ISO 26262 and SOTIF Alignment

The architecture maps directly onto established automotive safety constructs, requiring no novel safety argumentation:

For SOTIF: AI triggering conditions without an admissible execution path are structurally unreachable—not merely unlikely. This is a prevention mechanism, not an additional detection layer, and complements rather than replaces existing SOTIF test campaigns.

For UN R157 [3]: ODD boundary violations can be classified as non-admissible before the inference result is acted upon, providing structural ODD enforcement.

Table 1. Architecture–standard mapping

Architecture Element	ISO 26262 Construct	ASIL
HW Enforcement Kernel	Indep. Safety Monitor (Cl. 5.4.7)	C/D
HALT output	Safe State (Cl. 7.5)	All
Upstream watchdog	Hardware Watchdog (Cl. 6.4.10)	B+
Signal integrity/freshness	E2E Protection (AUTOSAR P5/6)	B+
Enforcement separation	Freedom from Interference (Cl. 5.4.9)	Mixed
Default HALT on POR	Fail-Safe Default State	B+

5 Robustness: Beyond Logical Correctness

A logically correct enforcement architecture is not automatically physically trustworthy. We identify four robustness requirements and their mechanisms:

Signal integrity (`signal_valid`). Parity and freshness timestamp on every bit vector. A malformed or stale vector forces HALT. Closes replay and bit-manipulation attacks.

Tamper detection (`tamper_detect`). Glitch detection, replay detection, and output register override detection. The resulting HALT is sticky—it clears only on hardware power cycle, preventing transient injection attacks.

Watchdog monitoring (`watchdog_ok`). Heartbeat monitoring of the upstream signal source. Sustained silence beyond a configurable grace window forces HALT. Closes the frozen-at-ADMISSIBLE attack vector.

Fail-closed default. Hardware pull-down on `allow_o`. Power-on state is unconditionally HALT. Execution requires a completed initialization handshake.

These four mechanisms distinguish between a system that *should not* execute under software governance and one that *cannot* execute under physical enforcement.

6 Hardware Integration Path

Three integration levels provide escalating assurance:

Level A (Firmware Governor). Runtime governor embedded in the inference stack. Synchronous enforcement. HALT = hard execution stop with state snapshot. ASIL-B capable; suitable for ASIL decomposition with a second independent channel.

Level B (Integrity Coprocessor). Dedicated sideband controller on an independent clock domain. Non-maskable halt interrupt. The compute plane cannot override the enforcement plane. ASIL-C capable.

Level C (Dedicated Silicon). Minimal FPGA or ASIC block. Non-programmable beyond configuration thresholds. The HALT path is physically non-bypassable. ASIL-D capable. The architecture is compatible with established automotive-grade platforms including Renesas R-Car, NXP S32, and Infineon AURIX at the respective integration levels described above.

7 Scope and Value to the escar Community

This presentation addresses the security–safety intersection directly relevant to escar: the enforcement boundary between AI inference and vehicle actuation is simultaneously a safety boundary and an attack surface. The tamper detection, replay protection, and fail-closed mechanisms are security properties that enable the safety claims.

The contribution to the escar audience is structural: a defined, hardware-coupled boundary between AI inference and actuation, grounded in a formal stability criterion, and aligned with ISO 26262 / SOTIF / UN R157 constructs. This is an architectural position paper, not a product presentation. Implementation-specific enforcement logic, signal structures, and hardware control semantics are intentionally withheld.

The presentation will cover: the governance gap and its structural cause; the three-layer architecture; the formal grounding; the ISO 26262 construct mapping; the robustness layer; hardware integration levels; and anticipated challenges including the primary failure point (upstream classification accuracy) and the open problem of authorization signal sourcing.

Disclosure of Interests. The author declares no competing interests. This work is the author's independent research. No funding was received for this work.

References

1. ISO 26262:2018—Road vehicles: Functional safety. International Organization for Standardization (2018)
2. ISO 21448:2022—Road vehicles: Safety of the Intended Functionality. International Organization for Standardization (2022)
3. UN Regulation No. 157: Automated Lane Keeping Systems (ALKS). United Nations Economic Commission for Europe (2021)
4. Regulation (EU) 2024/1689—Artificial Intelligence Act. European Parliament and Council of the European Union (2024)
5. AUTOSAR: Specification of E2E Communication Protection, Release 22-11 (2022)
6. Khalil, H.K.: Nonlinear Systems, 3rd edn. Prentice Hall, Upper Saddle River, NJ (2002)