



Before You Choose a Model, Engineer the Context.

A reference for systems engineering teams
preparing for AI-assisted work in regulated,
safety-critical product development.

Contents

I. The Decisive AI Question Is Not Which Model	3
II. The Early Failures Are About Context, Not the Model	6
III. From Chunked Retrieval to Graph-Aware Retrieval	7
IV. The Schema That Makes Engineering AI Deterministic at the Boundary	10
V. The Frameworks Customers, Auditors and Regulators Will Apply	12
VI. Where the Data Goes Is Not an Afterthought	17
VII. Do Not Trust AI Because It Sounds Right. Test It.	18
VIII. Where AI Helps and Where It Doesn't	21
IX. An Honest Ladder, With Observable Checks	23
X. A Pilot That Cannot Fail Is Not a Pilot	25
XI. Four Audiences, Four Positions	27
XII. Notes and References	29

EXECUTIVE SUMMARY

I. The Decisive AI Question Is Not Which Model

In systems engineering, the decisive question is not which AI model to use. It is what the model will be allowed to read, retrieve and reason over. AI value flows through the layer that supplies structured, versioned, permission-aware engineering context. Whichever layer owns that becomes the substrate for trustworthy AI — and the place where engineering trust either holds or breaks.

A NOTE ON SCOPE

This paper is about generative AI — large language models — used to assist engineers working on engineering data: drafting, summarizing, proposing trace links, assembling evidence packs. It is not about machine learning embedded as a function in the product, which is a different problem governed by a different set of standards (Section V). The two are routinely conflated, and that conflation is the source of most unhelpful AI-readiness advice.

Two terms are often used interchangeably. They should not be.

Data readiness

Engineering artifacts exist as typed objects with stable identifiers, are connected by explicit relations, are scoped to versions and baselines, and are governed by permissions and review state.

AI readiness

The workflow around the AI is engineered: retrieval is bounded by an evidence contract, outputs cite the objects they used, evaluation is continuous, deployment respects sovereignty and IP constraints, and a human disposition is recorded for every accepted change.

A team can have neither, one, or both. AI readiness without data readiness produces confident-sounding answers that fail review. Data readiness without AI readiness leaves leverage on the table. This paper is about how to reach both.

The architecture, in four stages

Everything that follows sits on one structure, and it is worth stating before the argument for it. An authoritative engineering graph holds typed objects and human-authored relations under configuration management. Governed evidence selection decides, through a declared contract and an authorization check applied before retrieval, what may enter the model's context. Bounded AI reasoning then works only on what was admitted, and produces a proposal with citations. Deterministic validation and commit closes the loop: the proposal is checked against schema and policy, approved explicitly by a person, and executed by a system-side service under that person's own permissions, with the disposition recorded.

While it is a challenge to make AI deterministic, we can aim to make AI's engineering context and authority deterministic.

That distinction carries the rest of the paper. The reasoning in stage three is not deterministic and this architecture does not pretend otherwise — a language model asked to detect ambiguity or propose a candidate trace link is doing something genuinely inferential.

What is deterministic is everything around it: which objects were admitted, under whose authority, with what provenance, and what it takes for a proposal to become a change of record. Trustworthiness comes from bounding the inputs and the authority, not from claiming the inference is exact.

Together they pushed the conversation from *is AI safe to use* to *how do you contain it*. Third, assessment practice moved. Automotive SPICE has moved to PAM 4.1, while customer assessment baselines may transition on different schedules, and the shift from test to verification terminology introduced with 4.0 raises the bar on traceability and verification evidence at exactly the moment AI tools are entering those workflows.

Organizations that use the window will set the patterns their auditors, customers and regulators come to expect. Those that skip them will rebuild under pressure.

Why this matters now

Three things have changed the calculus, and none of them is a model release. First, the regulatory picture firmed up rather than tightened. The Digital Omnibus on AI, Regulation (EU) 2026/1744, in force since 27 July 2026, moved the EU AI Act's high-risk deadlines out to December 2027 and August 2028. That converts a compliance scramble into a design window — and makes correct classification, rather than blanket caution, the thing to get right.

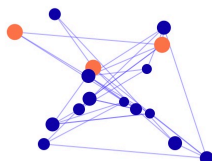
Second, security guidance caught up with retrieval, and then with agency. The 2025 revision of the OWASP Top 10 for LLM Applications added vector and embedding weaknesses to the catalogue; the 2026 edition, published in August, moved Excessive Agency up to third and broadened System Prompt Leakage into Hidden Context Exposure, covering retrieved material and tool schemas.



FRAGMENTED ARTIFACTS
Dispersed data, siloed tools,
lost context.



CONNECTED ENGINEERING DATA
AI links, classifies and structures
across sources.



STRUCTURED ENGINEERING CONTEXT
Unified, traceable and ready
for decisions.

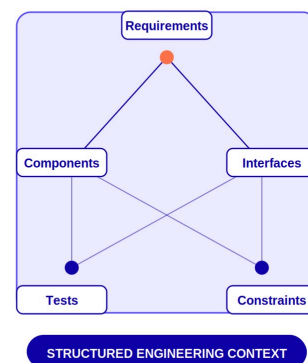


Figure 1: Fragmented artifacts become connected engineering data, then structured engineering context.

THE THESIS, AT A GLANCE

Engineering data starts as fragmented artifacts in siloed tools. AI value flows only when those artifacts are linked, classified and scoped into structured engineering context: typed objects, traceable relations, permission-aware boundaries.

WHAT ACTUALLY BREAKS FIRST

II. The Early Failures Are About Context, Not the Model

A pilot that asks an LLM to summarize a requirements document will produce something readable and plausible, and an engineer will say it sounds about right. The same pilot, asked which tests cover that requirement in the current customer baseline, returns an answer that mixes draft and released versions, references requirement numbers that no longer exist, and silently omits results from a variant the user does not have permission to see. The model did its job. The context did not.

The recurring failure modes are familiar to anyone who has worked on a complex program. Requirements that combine multiple obligations in one paragraph, so neither a reviewer nor a model can decompose them cleanly. Trace links that point to documents rather than to the underlying engineering objects, so a change to a requirement does not propagate. Test results stored alongside their tests but disconnected from the requirement intent and the variant they verify. Baselines that exist as labels rather than as scoped configurations. Permission models that depend on file-system locations rather than on the engineering meaning of the data.

None of these are AI problems. They are engineering data problems made visible by AI. The discipline AI requires is the discipline good systems engineering already wants — which is why teams with strong engineering practice tend to extract value from AI faster, regardless of which model they choose.

The discipline AI requires is the discipline good systems engineering already wants.

THE TECHNICAL SHIFT

III. From Chunked Retrieval to Graph-Aware Retrieval

Most early enterprise AI deployments use chunked-document retrieval-augmented generation. A document is broken into passages, each passage is converted to a vector, and the model is given the passages most semantically similar to the user's question. This works well for unstructured knowledge bases. On engineering data it runs into a limit that is worth stating precisely, because the imprecise version of this argument is easy to refute.

Consider a question a development engineer might actually ask: *which tests cover REQ-1042 in baseline 4.3 for the EU defense variant, and what is the current verification status?* Three of the four things that question needs are available to a well-built chunk store. Exact identifier matching is what hybrid retrieval is for, and REQ-1042 is a high-signal token that sparse retrieval handles well. Baseline and variant are metadata fields, and metadata filtering is a first-class primitive in every mainstream vector store. Permission scoping at query time — security trimming — predates language models by fifteen years and ships as a documented feature of the major search platforms. Anyone who tells you otherwise is describing a tutorial pipeline, not a competent integration.

The fourth thing is different. **Relation type** is not a filter. Metadata selects flat attributes; it does not distinguish *verifies* from *is allocated to* from *depends on* as traversable typed edges, and it does not walk from a requirement to its verification criteria to their test results to the evidence files. That is a traversal, and a chunk store has no traversal.

The deeper point is not about retrieval technology at all, and this is the version of the argument that holds. Filters can only select on attributes that were captured as data. Baselines that exist as document names rather than as scoped configurations are not filterable. Trace links that live in a table inside an exported Word file are not edges. Variant applicability recorded in a column heading is not a variant scope. Every one of those controls — the filters, the identifier matching, the permission sync — is also a component the integrator has to build, populate and keep synchronized with the engineering model; none is a default, and index-side permission copies are stale between synchronizations by construction. So the honest claim is not that chunked retrieval cannot do these things. It is that chunked retrieval over exported documents cannot recover structure that was never captured, and that the controls it does offer are separately maintained rather than inherent.

A graph-aware retrieval layer answers the deterministic part of the question deterministically — by traversing typed objects and relations within the user's permission scope and baseline filter — and then hands a focused, cited evidence pack to the model for synthesis. The model's job becomes describing what was found, not finding it.

When this architecture is not necessary

It is worth being direct about the limits of this argument, because the opposite claim is not credible. Ordinary document search and vector retrieval are perfectly adequate for low-assurance knowledge retrieval — finding the relevant passage in a process manual, summarizing a design rationale, answering a question whose correctness does not turn on engineering semantics. If the answer does not depend on typed relations, configuration or baseline state, variant applicability, permission boundaries or verification provenance, the additional structure described here buys governance you do not need and cost you cannot justify.

The graph-grounded architecture earns its keep precisely when those semantics materially affect whether the answer is right: coverage and impact analysis across a baseline, evidence assembly for a review, anything a reviewer or an auditor will act on. Most organizations will want both, applied to different questions. Deciding which question is which is the first piece of engineering judgment this architecture asks for.

One clarification, because the vocabulary is crowded. This is not what the literature calls GraphRAG, where a language model extracts a knowledge graph from unstructured text and retrieval then runs over that inferred graph. Published evaluations of that approach report substantial entity loss during construction and heavy build costs, and on precise single-hop lookup it does not reliably beat plain vector retrieval. What is described here starts from a graph that already exists: typed objects and relations authored by engineers and held under configuration management. The retrieval is deterministic because the structure is given, not inferred — which is a narrower claim, and a much easier one to defend.

This is the second stage of the four. Retrieval here is **deterministically bounded** rather than deterministic in any absolute sense: given a fixed snapshot of the graph, correct relations and a fixed authorization state, the same query admits the same set of objects. It does not follow that the graph is complete, that the relations are right, or that the answer is therefore correct — only that what the model was allowed to see is knowable and repeatable.

Bound the context and the authority deterministically. Let the reasoning be reasoning.

Bounded AI reasoning inside deterministic engineering authority

The context and the authority are deterministic. The reasoning is bounded, not deterministic.

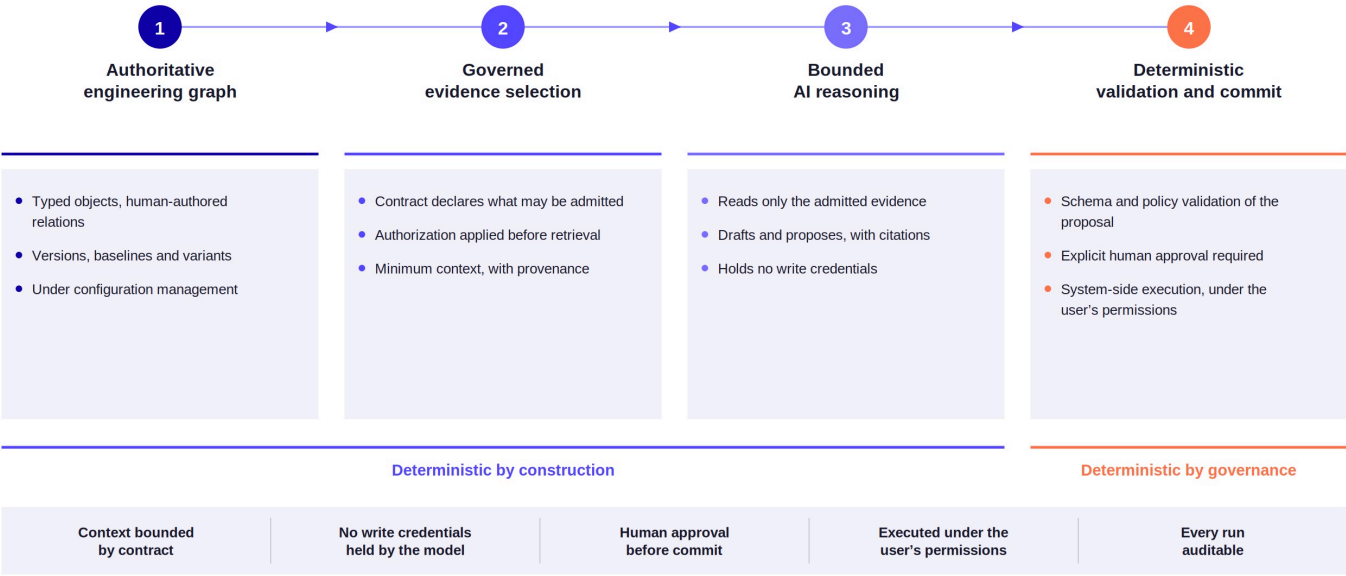


Figure 2: The four-stage architecture. The context and the authority are deterministic; the reasoning is bounded, not deterministic.

THE EVIDENCE CONTRACT

IV. The Schema That Makes Engineering AI Deterministic at the Boundary

Stage two is where most of the governance lives, and the construct that carries it is the **evidence contract**: a declared specification of what may enter the AI's context window for a given workflow.

An evidence contract for a “test coverage gap” workflow specifies:

Allowed object types	System requirement, software requirement, verification criterion, test case, test result, evidence file.
Allowed relations	<code>verifies</code> · <code>satisfies</code> · <code>derives_from</code> · <code>allocated_to</code> · <code>is_evidence_for</code>
Baseline scope	Current customer baseline only. Drafts and released siblings are excluded.
Variant scope	Active variant tree for the user's project.
External sources	Test execution results from the connected test management system, restricted to the user's project.
Permission filter	Intersection of the user's read permissions across all admitted object types.
Citation requirement	Every claim in the answer must reference at least one object identifier within the admitted set.
Abstention rule	The workflow returns “insufficient evidence” rather than synthesizing when retrieval was incomplete — a required source was unreachable, the permission filter removed objects the answer depends on, or the traversal hit a declared depth limit. It must not abstain merely because a count came back low: in a coverage-gap workflow, “zero test cases verify this requirement” is the finding, not a failure to find one. Separating “I found nothing” from “I could not look properly” is the hard part, and the contract has to say which signals mean which.

A contract is not a prompt. A prompt is what the model sees. A contract is what the system decides the model is allowed to see. A prompt can be argued with; a contract is applied by the retrieval layer, not requested of the model. Be precise about which clauses that covers: object types, relations, baseline and variant scope, external sources and the permission filter are admission control, applied before the model is invoked. The citation requirement and the abstention rule are checks on what comes back, and can only be applied after it.

Admission control is also not an injection defense, and it is worth saying so rather than implying otherwise. Controlling which objects enter the context window says nothing about what those objects say. A requirement imported from a supplier, a comment field on a test result, the body of a change request — all admitted types in the contract above — can carry text addressed to the model, and the contract admits them precisely because they are in-baseline, in-variant and permission-cleared. There is no reliable filter for this, and a vendor who claims one should be asked to demonstrate it.

What does work is removing the model's ability to act. The boundary is worth stating as a sequence, because "human in the loop" is too vague to assess: the model produces a structured proposal; that proposal is validated against a schema and against policy; a person approves it explicitly; a system-side service then executes it under the requesting user's own permissions; and the result together with the human disposition is written to the run log. The model holds no write credentials and no direct write authority at any point in that sequence.

Note the precise form of the containment claim. The model reaches an approved inference endpoint — Section VI permits a sovereign or, under conditions, an external one — so it is not true that there is no network path. What there is no path for is autonomous outbound action: the workflow gives the model no tool with which to send mail, upload a file, call an external service, transact, or write to engineering data. An instruction injected into a requirement description can still be read, and an injected instruction therefore has no autonomous actuator. Its remaining path to harm is through the proposal and its human review, which is why meaningful review and the automation-bias controls in Section VII still matter. That is a containment argument rather than a detection argument, and containment is the one that survives contact with an adversary — but containment bounds the damage, it does not remove the need to read what comes back.

This is where most generic enterprise AI deployments fail in regulated engineering. Without a contract, the model's behavior is whatever it was trained to do, modulated by whatever instructions a user can phrase. With a contract, the model operates inside a deterministic boundary owned by the engineering tool, not by the model vendor.



STANDARDS AND RISK MAPPING

V. The Frameworks Customers, Auditors and Regulators Will Apply

The relevant question is not whether to comply with all of them, but to be explicit about which apply to a given workflow and how the architecture supports them.

One scoping note on the security guidance. The seven entries below are drawn from the OWASP Top 10 for LLM Applications. The architecture described here deliberately minimizes agentic authority: the model proposes and a governed service commits. Where an implementation goes further and gives the model tools, persistent memory, or the ability to initiate consequential actions, the OWASP Top 10 for Agentic Applications should be treated as complementary guidance rather than as out of scope.

ISO/IEC 42001 — AI management system

The standard now appearing in supplier contracts. It defines an AI management system in the sense ISO 9001 defines a quality management system: documented policies, defined roles, lifecycle controls, risk treatment, continual improvement. Engineering tools that introduce AI features will be expected to provide the audit hooks — run logs, version pinning, decision records — that customers' 42001 programs depend on.

NIST AI RMF and the GenAI Profile (NIST AI 600-1)

The framework's four functions — Govern, Map, Measure, Manage — apply cleanly to engineering AI, with Govern cutting across the other three rather than preceding them. The 2024 GenAI profile adds specific actions for content provenance, pre-deployment testing and incident response that directly inform the design of an evidence contract, an evaluation harness and an audit log.

OWASP Top 10 for LLM Applications (2026)

The 2026 edition supersedes the 2025 list in the project's own presentation, and three of the identifiers relevant here moved: Excessive Agency from LLM06 to LLM03, Misinformation from LLM09 to LLM07, and Vector and Embedding Weaknesses from LLM08 to LLM09. Two categories now bear directly on this architecture and are included below.

RISK	CONTROL IN A GRAPH-GROUNDED ENGINEERING AI
LLM01 <i>Prompt Injection</i>	Treat every retrieved artifact as untrusted input, including content authored inside the trust boundary — a supplier requirement or a compromised account arrives through the blessed channel. Isolate instructions from evidence in the prompt structure. Then contain rather than filter: read-only retrieval, no write credentials held by the model, no autonomous outbound action channel. Normalization and content checks are defense-in-depth; they are not the boundary.
LLM02 <i>Sensitive Information Disclosure</i>	The contract declares the authorization constraint; the authorization and data-access layer enforces it, before retrieval. Filtering again at response time is defense-in-depth, not the boundary. An access-control property is argued from the design, not from a pass rate.
LLM03 <i>Excessive Agency</i>	The model proposes; it does not execute. Proposals pass schema and policy validation, then explicit human approval, and a system-side service performs the write under the requesting user's own permissions. Apply least privilege to those services — the minimum operations the workflow needs — and treat rubber-stamped approval as the failure mode it is.
LLM05 <i>Data and Model Poisoning</i>	Controls belong on ingestion, not on the prompt. Admit engineering content only with source identity, provenance and configuration or review state recorded; keep change history and content hashes for every admitted object; restrict write access to the graph and to any retrieval index to system services, never to the model; quarantine or validate imported sources before they become admissible; and test explicitly for poisoning, including content planted through a legitimate import path.
LLM07 <i>Misinformation</i>	Require citations to specific object identifiers, paths, baselines and evidence objects. Treat uncited claims as failures, not stylistic variation.
LLM08 <i>Hidden Context Exposure</i>	The contract bounds which engineering objects may enter context; it does not make anything in that context secret. Admit only the minimum required; enforce authorization before retrieval; never place credentials or secrets in prompts or retrieved content; minimize exposure of internal instructions and tool schemas; treat retrieved engineering content as potentially hostile; and test explicitly for context leakage. Response filtering is defense-in-depth.
LLM09 <i>Vector and Embedding Weaknesses</i>	Prefer typed graph traversal and identifier matches over similarity search where the engineering meaning is well-defined. Use embeddings only for genuinely fuzzy retrieval, and scope them to the same admitted set.

EU AI Act

Start by classifying correctly, because the common assumption is wrong. High-risk status under Regulation (EU) 2024/1689 is a closed-list determination, not a judgment about how consequential the work feels. An assistant that helps engineers author requirements, propose trace links or assemble an evidence pack is a development-lifecycle tool: it is not a safety component of the product being built, and authoring engineering documentation is not one of the eight Annex III use cases. Most internal engineering AI therefore sits outside the high-risk regime altogether. The *Digital Omnibus on AI* moved further in that direction, narrowing when AI inside a regulated product counts as a safety component at all: AI performing user assistance, efficiency, automation, convenience or non-safety quality-control functions does not become a safety component merely by being integrated into such a product.

Several horizontal provisions can apply irrespective of high-risk classification, depending on the system and the use case, and these are the ones to plan against. Article 5 *prohibits* a closed list of practices — a restriction on what may be done, not a duty every system performs. Article 50 sets transparency obligations for certain AI systems, with distinct triggers — direct interaction with a person, generation of synthetic content, emotion recognition, deepfakes — so it applies where a trigger is met rather than universally. Article 4 asks providers and deployers to take measures to support AI literacy among the staff who operate these systems, and applies broadly.

Two paths do lead into the high-risk regime, and both deserve an explicit check rather than an assumption. A tool used to evaluate engineer performance or allocate work on the basis of behavior falls under Annex III(4)(b). AI that becomes a safety component of the product itself falls under Annex I. For those, the Omnibus deadlines are **2 December 2027** and **2 August 2028** respectively, deferred from 2026 and 2027.

Where the regime does apply, the core assurance architecture remains: a documented risk management process, traceability of training and operation, technical documentation sufficient for regulator review, human oversight that is meaningful rather than ceremonial, and record-keeping. “Unchanged” would be the wrong word, though. The Omnibus left the requirements on risk management, record-keeping, transparency, human oversight and accuracy alone, amended the data-governance and technical-documentation requirements, and narrowed the classification perimeter itself. The assurance architecture remains; its scope and its edges moved.

The practical consequence is not less rigor. It is honest scoping. A supplier who claims the high-risk label where it does not apply invites questions it cannot answer; a supplier who cannot say which tier applies invites the same questions. Document the classification, and document the reasoning behind it.

Automotive SPICE

Automotive SPICE PAM 4.1 was published in 2026, together with the third edition of the VDA QMC Automotive SPICE Guidelines for PAM 4.1. Customer program, assessor qualification and contractual assessment baselines may transition on different schedules, so confirm which version governs a particular assessment.

The process areas that AI workflows most often touch are the same either way. The work products affected — SYS.2 System Requirements Analysis, SYS.3 System Architectural Design, SWE.1 Software Requirements Analysis, SWE.6 Software Verification — all carry traceability and verification expectations that AI must reinforce, not erode. The vocabulary change relevant to this paper originated with 4.0, which replaced *test* with *verification* across SWE.6, SYS.4, SYS.5, SWE.4 and SWE.5. The process identifiers discussed here remain applicable in 4.1, so the AI-workflow mapping is unaffected. A team still mapping its AI workflows against 3.1 process names is mapping to a vocabulary the assessment no longer uses. Map every AI workflow to the work product it actually touches.

Domain standards — and the AI standards now sitting beside them

The four domain standards most often in scope — ISO 26262:2018, ISO/SAE 21434:2021, DO-178C and ARP4754B — contain no AI-specific clauses, and none has an AI amendment in force. That much is still true. It is no longer the whole picture, and a paper that stops there invites a correction from the first safety engineer who reads it.

AI-specific companions now exist. **ISO/PAS 8800:2024**, Road vehicles — Safety and artificial intelligence, is applied in combination with ISO 26262 and ISO 21448 wherever a vehicle function contains an AI model. **ISO/IEC TR 5469:2024** addresses functional safety and AI systems generally, and **ISO/TS 5083:2025** covers safety for automated driving systems.

In aerospace, EASA's AI concept papers are the operative guidance — Issue 2 (2024) is the last finalized version, with a proposed Issue 03 released for comment in June 2026 and not finalized at the time of writing — while **SAE ARP6983 / EUROCAE ED-324** is in development and not yet published: EUROCAE lists ED-324 as a draft with a target publication date of 31 December 2026, and Issue 1 is expected to address non-adaptive, supervised machine learning in the product rather than generative AI. As of September 2026 there is no published AI-specific supplement to DO-178C in the DO-330/331/332/333 series.

Read the scope statements, though, and a distinction emerges that matters more than the list. These standards govern AI *embedded as a function in the product*. ISO/PAS 8800 says so in as many words: it “does not provide specific guidelines for software tools that use AI methods,” and “the development of AI elements that are not part of the vehicle is not within the scope of this document.” Published drafts of ARP6983 are scoped to non-adaptive, supervised machine learning in the product and exclude generative AI, though the scope of Issue 1 should be confirmed against the published text. An LLM that drafts a requirement or proposes a trace link is not an AI element of the vehicle or the aircraft, so none of these documents govern it as a product function.

What governs the engineering workflow is much thinner. ISO/IEC TR 5469 names “use of AI systems to design and develop safety related functions” as one of its three usage classes — the only place in the published corpus where AI-as-engineering-tool is squarely in scope — but it is an informative Technical Report, industry-agnostic, and written around classical machine learning rather than generative models.



EASA's proposed Issue 03 scopes in "AI-based development and verification tools used to develop approved products," treats general-purpose LLMs as a distinct category, caps them at the lowest assurance levels, and then leaves the objectives to future rulemaking. The FAA's AI safety assurance roadmap mentions LLMs for requirements traceability and answers with an accountability principle rather than a method: the applicant is responsible for its compliance evidence *regardless of how those artifacts were created*.

There is one route into the domain standards that does not need an AI clause at all, and it is the one an assessor will raise first. **ISO 26262-8:2018 Clause 11**, Confidence in the use of software tools, and DO-330, Software Tool Qualification Considerations, are written about tools — and an LLM that drafts requirements or assembles verification evidence is a tool used inside a safety lifecycle. The two frameworks ask for the same thing in different vocabularies, and mixing those vocabularies is a reliable way to lose an assessor's confidence.

Under ISO 26262-8, the route runs from Tool Impact and Tool Error Detection to a **tool confidence level** (TCL), and confidence at that level is argued through increased confidence from use, evaluation of the development process, validation of the tool, or development in accordance with a safety standard. Under DO-178C and DO-330 the terms are different: tool criteria 1, 2 or 3 combine with the software level to give a **tool qualification level** (TQL-1 to TQL-5), and qualification is against DO-330 objectives.

Both ask the same question in the end: does the tool eliminate, reduce or automate a process, and if so what is the argument that its output can be trusted. Where that argument is redundant verification of the output, both care whether the reviewer is genuinely independent. The answers are program-specific and belong in the assessment rather than in a whitepaper — but a team that has not prepared them will be asked for them in the first audit.

That is also where the red line in Section VIII earns its keep. "AI-generated evidence does not replace independent verification" is not a policy preference; it is the shape of the tool-confidence argument most programs will end up making. The domain standards impose independence requirements for verification where the applicable safety or assurance level requires it, alongside configuration management, change impact analysis and traceability of decisions — and they impose them regardless of what produced the artifact. Nothing normative yet tells a team how to demonstrate that once a generative model drafted the first version. Until something does, the evidence contract, the run log and the recorded human disposition are how a team shows its own answer — which is why they are worth building deliberately rather than retrofitting.

DEPLOYMENT ARCHITECTURE

VI. Where the Data Goes Is Not an Afterthought

Defense, automotive and aerospace customers ask the same first question: where does the data go? The answer must be specific, not aspirational. A defensible deployment for engineering AI has four properties.

01

Inference is sovereign by default.

The reference deployment runs inference on a customer-controlled host, in a customer-chosen jurisdiction. Cloud LLM APIs are an option only under zero-retention contracts and only for data classified as eligible.

02

No customer engineering data is used for model training.

This is a contractual guarantee, not a setting. It applies to base models, retrieval indexes, fine-tuning data, and any logs that leave the customer perimeter.

03

Every run is auditable, and replayable under declared conditions.

A run log records user identity, contract, prompt template version, model digest and sampling configuration, retrieval index version, baseline, the retrieved object set with content hashes, output, and the human disposition. Note the wording: LLM inference is not bit-reproducible in general, because reduction order in GPU kernels varies with server batching. Bit-identical replay requires batch-invariant kernels on a fixed parallelism configuration, at a material latency cost. What the log guarantees is that any run can be reconstructed and accounted for — which is what an audit actually asks of you.

04

The model proposes; the system commits.

The model emits a structured proposal and holds no write credentials. The proposal is validated against schema and policy, approved explicitly by a person, and then executed by a system-side service under the requesting user's own permissions, with the result and the disposition logged. This is a property of the architecture, not a configurable policy.

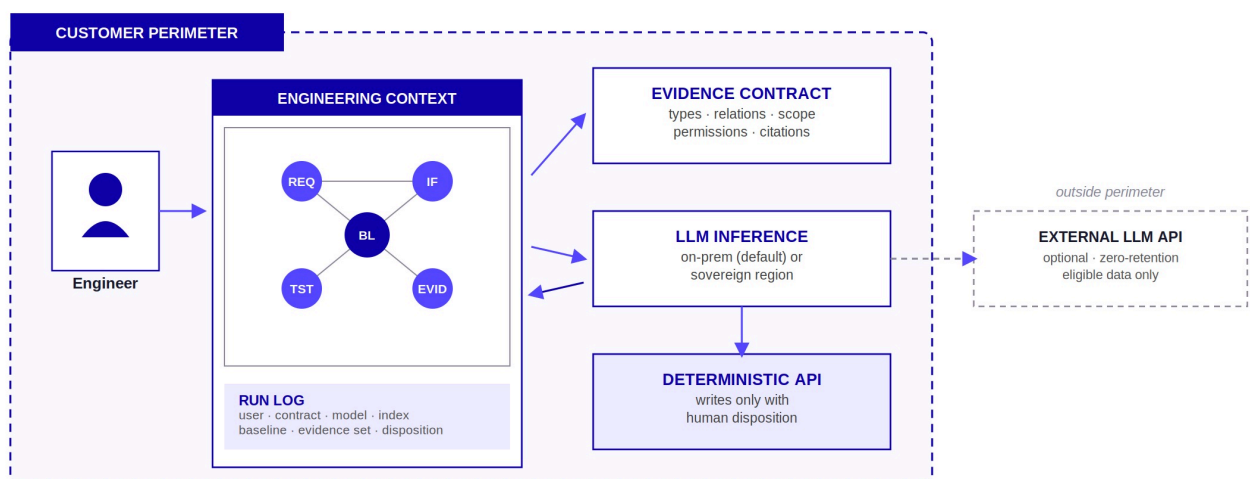


Figure 3: The trust boundary is the customer perimeter. The AI proposes; deterministic APIs commit.

EVALUATION DISCIPLINE

VII. Do Not Trust AI Because It Sounds Right. Test It.

An engineering AI workflow without a measured evaluation harness is a demo, not a product. The metrics below distinguish workflows that survive audit from those that do not.

The targets in the right-hand column are acceptance criteria this framework proposes. They are *not* published industry norms: no standards body has set threshold values for any of these metrics, and reported figures in the research literature are not comparable across different corpora and grading methods.

Treat them as program-specific criteria — declared before a pilot starts, then calibrated against your own measured baseline.

Citation precision	Definition: Per citation: the share of cited identifiers whose content actually entails the claim attached to them. Distinct from whether the citation merely resolves to an admitted object.	<i>Target: ≥ 0.85 for assistive production use with mandatory human verification; 0.75–0.85 remediate before scaling; < 0.75 unacceptable.</i>
Citation resolvability	Definition: Per citation: the share of cited identifiers that resolve to an object inside the admitted set. A necessary condition for precision, and much easier to hit.	<i>Target: 1.0; a citation to an object outside the contract is a contract failure, not a quality miss.</i>
Citation recall	Definition: Per claim: of the objects that should have been cited for it, the share that were. The metric that catches missed evidence.	<i>Target: ≥ 0.85 for assistive production use; declare the pilot target explicitly. Do not leave this one unthresholded — missed evidence is the failure mode that matters most in safety work.</i>
Faithfulness	Definition: Per claim: the share of claims entailed by their cited evidence rather than inferred or imagined. The same entailment judgment as citation precision, counted per claim instead of per citation.	<i>Target: ≥ 0.90 for assistive production use with mandatory human verification. This is hallucination, named precisely and measured.</i>
Abstention accuracy	Definition: Of the golden questions that should be abstained on, the share where the workflow correctly returns “insufficient evidence.” Note the denominator: it is the unanswerable subset, not the whole set, so the target is high rather than mid-range.	<i>Target: ≥ 0.90 on the unanswerable subset, with false abstentions on answerable questions tracked separately.</i>

Scope adherence	Definition: Share of answers that respect the declared baseline and variant boundary.	<i>Target: Approaches 1.0; any leak is a defect.</i>
Permission compliance	Definition: Share of answers that do not surface objects the user lacks permission to read.	<i>Target: 1.0; any failure is a security incident.</i>
Stability under drift	Definition: Movement on the golden set when model, prompt, index or schema changes.	<i>Target: Any regression beyond the measurement interval of the golden set blocks release. State that interval — see below.</i>

Read those thresholds with the tier in mind. None of them describes an autonomous workflow. They are acceptance criteria for assistive use in which a competent engineer verifies the output before it becomes a change of record — which is why a figure below 1.0 is tolerable at all. Quoted without that condition, a citation precision of 0.85 sounds like a claim that fifteen percent evidence errors are acceptable in production, and it is not.

A practical evaluation set is a curated bank of 30–100 golden questions per workflow, each with expected cited identifiers, expected scope, and an acceptable answer range, including a deliberate subset that should be abstained on. The set is owned by engineering, not by the AI team, because the questions encode engineering judgment.

Ownership should be a name, not a department. Every workflow that reaches production has a named engineering or assurance owner for its golden set. The set is version-controlled alongside the workflow it evaluates, and changes to it are reviewed like any other assurance asset — because a golden set that can be quietly edited to make a threshold pass is not evidence of anything. Regression runs gate every change to model, prompt or index.

Automation bias

One risk deserves naming because the whole architecture leans on it. Human review is not automatically meaningful merely because a person clicked approve. People asked to check plausible machine output over time tend to check it less carefully, and a workflow that produces confident, well-cited drafts is exactly the kind that invites this. The paper's own oversight requirement fails quietly if that happens, and nothing in a run log distinguishes a considered approval from a reflexive one.

Three practices help, and none of them is a metric target. Do not set acceptance-rate goals — they put institutional pressure on reviewers to agree, which manufactures the problem. Preserve reject and edit reasons rather than only outcomes, so that the shape of reviewer disagreement stays visible. And sample accepted outputs for periodic independent review by someone who did not approve them, on the assumption that some proportion of approvals will not survive that check. What you learn from that sample is a better signal than any acceptance rate.

How these are measured, and why the numbers move

Citation precision and faithfulness require a judgment about whether a cited object entails a claim. That judgment is usually made either by an engineer reviewing a sample or by a model acting as an automated grader. The two do not agree perfectly, and the disagreement is large enough to matter: automated graders align with careful human review well enough to track a trend, but not well enough to certify a threshold. Any program that gates a release on these metrics should state, alongside the number, who or what graded it, on how many items, and with what inter-rater agreement — and should keep a human-reviewed sample as the reference. A precision figure quoted without its grading method is not a measurement.

Sample size then decides what the numbers can actually distinguish. Thirty-two correct out of forty golden questions gives a ninety-five percent confidence interval of roughly 0.65 to 0.90 — about **twenty-four percentage points wide, or ± 12 points**. A two-tier gate at 0.75 and 0.85 is therefore not resolvable on that evidence base, and a run that reads 0.83 has not demonstrably missed a 0.85 target. Either size the set to the decision, or state the interval alongside the number and treat a near-miss as a measurement result rather than a verdict. The point of these metrics is to catch a regression you would otherwise ship, not to manufacture a decimal place the evidence cannot support.

Two of these are security and scope properties rather than quality metrics, and they should not be argued from a sample at all. Permission compliance and citation resolvability are architectural: a permission bypass is a defect with a root cause, not a rate to be averaged.

Demonstrating 1.0 across forty questions bounds the true failure rate at roughly seven percent, which no security assessor will accept as evidence of an access-control property. Argue those from the design plus adversarial testing, and use the golden set to detect regressions in them, not to certify them.

The problem this architecture does not solve

Permission scoping produces a second-order failure that deserves naming, because it is the same failure Section II opened with. An answer assembled inside a permission boundary is complete *with respect to that boundary* and silently incomplete with respect to the engineering reality. For a coverage-gap workflow this is not a hypothetical: “no gaps found” can mean “no gaps among the objects you are cleared to see,” and in a safety context those are different statements with the same wording.

There is no clean answer. Disclosing that N objects were excluded leaks the existence and shape of data the user is not cleared for, which is the reason the boundary exists. Not disclosing it produces a confidently incomplete answer of exactly the kind this paper argues against. What a workflow can do is declare which of the two it has chosen, per workflow, and record that choice in the contract — so that a reviewer reading an answer knows whether completeness was scoped or absolute. For safety-relevant coverage and impact analysis, the defensible default is to run the query under a role that spans the full variant tree and to route the answer to someone cleared to receive it, rather than to give a partially-sighted answer to whoever asked. We would rather state this openly than let a metric called permission compliance imply it has been handled.

WHAT AI SHOULD AND SHOULD NOT DO

VIII. Where AI Helps and Where It Doesn't

Stage three of the architecture is where this question gets settled. Bounded reasoning is not a euphemism for summarizing: the model may genuinely infer, propose and discover inside the admitted evidence. What the boundary fixes is the input and the authority, not the ambition of the reasoning. So the useful split is not between reasoning and lookup — it is between work where inference adds something and work where a query already answers the question.

The fuzzy work, where AI earns its place

Detecting ambiguity in requirement text. Suggesting candidate trace links for human approval. Summarizing a change request and its dependencies. Drafting a verification evidence pack for an engineer to refine. Surfacing potentially related requirements across projects. Classifying requirements by quality criteria. Drafting first-pass documentation that an engineer edits.

The deterministic work, where AI does not earn credit

Counting requirements. Listing requirements without verification links. Finding requirements last modified before a date. Listing tests assigned to a tester. These are queries against a structured model, and the graph answers them without a language model in the path. Presenting them as AI accomplishments dilutes the case for AI where AI actually helps.

The example in Section III — which tests cover this requirement in this baseline for this variant — belongs in this second category, and that is the point of using it. It demonstrates what the retrieval layer can establish on its own, which is precisely what makes the fuzzy work above safe to attempt: the model is reasoning over evidence that was selected deterministically rather than searching for it. Cross-project discovery is a different workflow with a different contract, and it should be scoped and evaluated as one rather than smuggled into a baseline-scoped query.

RED LINES, BY DESIGN

- The AI does not autonomously modify baselined engineering objects.
- The AI does not author safety-critical or cybersecurity-critical requirements; it may draft language for explicit engineer authorship and review.
- External content does not become admissible evidence without controlled ingestion: source identity, provenance and review state recorded, and validation or quarantine first. Normalization and content checks are defense-in-depth; they are never treated as a prompt-injection boundary.
- The model holds no write credentials and no direct write authority. Every state change is a validated proposal, explicitly approved, executed by a system-side service under the requesting user's permissions.
- AI-generated evidence does not replace independent verification.

Trajectory

The current product direction expands AI's role in candidate suggestion (additional link types, additional artifact types) and in evidence-pack assembly. It does not include autonomous write on baselined objects. Customers asking *will the AI eventually do this for me* should know which side of that line their use case sits on.

Statements about product direction describe current intent at the date of this paper and are not commitments.



MATURITY AS SELF-ASSESSMENT

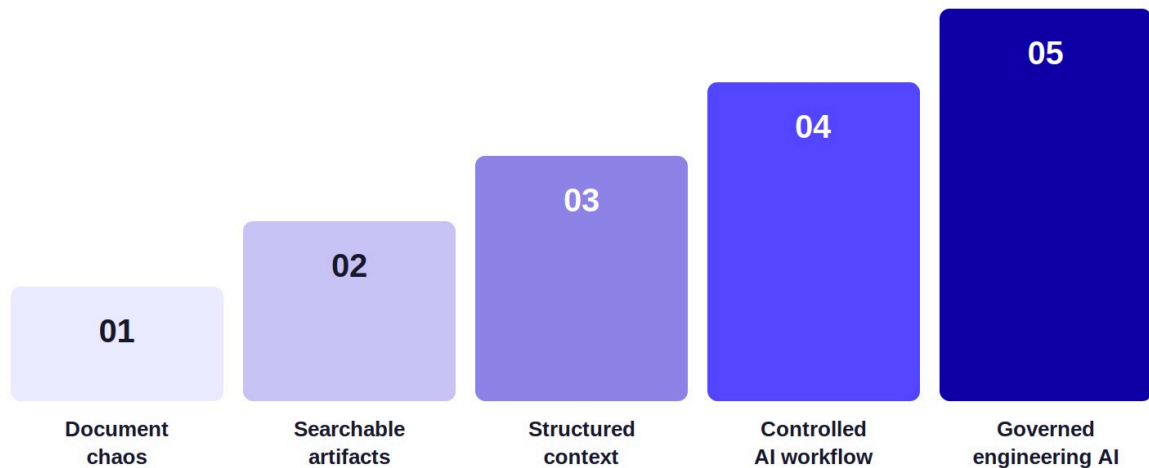
IX. An Honest Ladder, With Observable Checks

The five levels below are useful only if they are testable. Each is defined by two checks. The levels are cumulative: an organization is at level N when the checks for level N and every level below it pass. That matters, because the checks are independent of one another — a team can have a regression-tested evaluation harness without capturing run logs — and without the cumulative rule the ladder would happily certify a level 5 with no audit trail.

Two honest caveats. Levels 1 and 2 are written as descriptions of a deficient state, so read them as “this still describes us” rather than as achievements to pass. And assess a single workflow at a time: the level 4 check asks for one governed workflow while the level 5 check asks about every change, so a whole organization score is not meaningful. Most real programs sit at different levels for different workflows, and the mixed estate — typed objects for new work, document-to-document links for legacy — is the normal case rather than the exception. Where a check turns on a word like **primarily** or **manual effort**, write down the threshold your program means before you use it to score yourself, or the ladder will flatter you.

From document chaos to governed engineering AI

Five levels of engineering-AI maturity. Each is defined by two observable checks, listed in the table below.



SIGNALS TO TRACK BETWEEN LEVELS

- ✓ Citation precision
- ✓ Citation recall
- ✓ Abstention accuracy
- ✓ Reviewer disposition
- ✓ Auditable run logs

Figure 4: Five levels of maturity, with the signals worth tracking between them.

LEVEL	FIRST CHECK	SECOND CHECK
01 · Document chaos	Engineering knowledge primarily lives in documents and exports.	Reconstructing a baseline requires manual effort.
02 · Searchable artifacts	Key content is findable by full-text search.	Trace links exist, but primarily between documents rather than between engineering objects.
03 · Structured context	Core engineering objects are typed and linked.	Versions and baselines are queryable, not just labeled.
04 · Controlled AI workflow	At least one AI workflow operates under a declared evidence contract with permission-aware retrieval and citations.	Run logs are captured and reviewable.
05 · Governed engineering AI	A regression-tested evaluation harness gates every change to model, prompt or index.	AI workflows are explicitly mapped to work products in the relevant standards.

A 90-DAY PILOT WITH EXPLICIT THRESHOLDS

X. A Pilot That Cannot Fail Is Not a Pilot

Weeks 1–2	Choose the workflow. One workflow, buyer-relevant, narrowly scoped. Strong candidates: test coverage gap analysis for a specific baseline; change impact analysis on a defined feature; safety-evidence pack assembly for an upcoming review; or requirements quality screening on a defined input set.
Weeks 3–4	Define the evidence contract. Object types, relation types, baseline and variant scope, external sources, permission filter, citation requirement, abstention rule. Documented and signed off by the engineering owner of the workflow.
Weeks 5–6	Build the golden set. 30–50 questions with expected cited identifiers and expected scope, including a deliberate subset that should be abstained on. Construction is engineering-led; the AI team supports tooling. Note what this set can and cannot do at this size: it will catch a regression, and it will not resolve a ten-point difference between two thresholds.
Weeks 7–9	Run the workflow. Engineers use the AI workflow on real work, with citations required and abstention enabled. Every run is logged. Every disposition (accept, edit, reject) is recorded with a reason code.
Weeks 10–12	Decide on scale. The hard gates below must clear; any failure stops the decision outright. The quality measures are not gates on this first cycle — the pilot is the first time they have been measured, so it establishes the baseline and sets the thresholds the next cycle will be held to. Record the numbers, record the grading method, and write down the targets before the next cycle starts.

SCALE-DECISION HARD GATES

Any failure stops the decision. Note that the first four are checked against the evidence contract, so they are only as strong as the contract is narrow — which is why the fifth is on the list.

- Permission compliance = **1.0** — argued from the design and adversarial testing, not from the sample.
- Scope adherence = **1.0** — like permission compliance, argued from the design and adversarial testing, with the golden set used to detect regressions rather than to certify the property.
- Citation resolvability = **1.0** — a citation outside the contract is a contract failure.
- Documented mapping to at least one applicable standard work product.
- The contract itself reviewed and signed off by someone other than its author, on the test that the declared scope is the narrowest that serves the workflow.
- Adversarial testing performed to a declared scope and attacker model, by testers independent of the build, with every finding triaged and none of severity high or above left unresolved.

MEASURED, NOT GATED, ON THE FIRST CYCLE

These establish the baseline and become the thresholds for the next cycle. Report each with its grading method and sample size.

- Citation precision and citation recall, per Section VII.
- Faithfulness, and abstention accuracy on the unanswerable subset.
- Reviewer disposition — the mix of accept, edit and reject, with reason codes. Track it; do not gate on it. A target acceptance rate puts pressure on reviewers to accept, which manufactures exactly the ceremonial oversight the AI Act asks you to avoid, and a rigorous reviewer population that rejects a lot of first drafts would fail such a gate for the right reasons.

A pilot that fails a hard gate has not been scaled, and that is the point of having gates. It will also have produced a precise diagnosis of which part of the engineering context is not yet ready — worth having, but not a substitute for the decision it just lost.

POSITIONS BY ROLE

XI. Four Audiences, Four Positions

EXECUTIVES

AI value in engineering is gated by the layer that supplies structured, versioned, permission-aware context. Owning that layer — or buying a tool that does — is a more durable bet than betting on a specific model.

ENGINEERING LEADERS

Pick one painful workflow. Define the evidence contract. Build the golden set. Measure evidence quality, not time saved. Scale only when the thresholds clear.

SAFETY, SECURITY AND QUALITY

Treat engineering AI as a tool-supported workflow with permissions, review gates, audit trails, regression tests and a defined map to the standards in scope. The same disciplines that make systems engineering work make AI safe to use within it.

PROCUREMENT AND LEGAL

Contractual surfaces that matter: data residency, training-data carve-outs, model-vendor retention, run-log access for audit, sub-processor disclosure, EU AI Act conformity support, ISO/IEC 42001 audit hooks.

IN CLOSING

The decisive question is not which model

It is whether the engineering context the model receives is structured enough to be trusted, scoped enough to be safe, and governed enough to be auditable. An organization that gets the context layer right will use whichever model is best at any given time. An organization that gets it wrong will be locked into whichever model first papered over its data debt — and will discover the cost when an audit, an incident or a customer asks for evidence.

Want a half-day readiness assessment of your requirements, traceability and verification structure?

info@systemweaver.com

systemweaver.com

XII. Notes and References

1. OWASP Top 10 for Agentic Applications for 2026, published 9 December 2025 by the OWASP Gen AI Security Project. Its identifiers use the ASI prefix and address goal hijack, tool misuse, agent identity and privilege abuse, memory and context poisoning, and related agentic failure modes.
2. OWASP Top 10 for LLM Applications 2026, published August 2026 by the OWASP Gen AI Security Project, superseding the 2025 edition. LLM08:2026 Hidden Context Exposure broadens the former LLM07:2025 System Prompt Leakage to cover retrieved material, tool schemas, credentials and operational context.
3. Regulation (EU) 2024/1689, Article 6, as amended by Regulation (EU) 2026/1744, which inserted new paragraphs 6(1a) to 6(1c) narrowing the circumstances in which an AI system integrated into a product covered by the legislation listed in Annex I is to be treated as a safety component.
4. Regulation (EU) 2024/1689, Article 5, applying since 2 February 2025. Regulation (EU) 2026/1744 added two further prohibited practices, applying from 2 December 2026.
5. Regulation (EU) 2024/1689, Article 50, Transparency obligations for providers and deployers of certain AI systems, applying from 2 August 2026. The machine-readable marking duty in Article 50(2) has a transition to 2 December 2026 for systems already placed on the market.
6. Regulation (EU) 2024/1689, Article 4, as amended by Regulation (EU) 2026/1744, which softened the duty from ensuring a defined level of AI literacy to taking measures to support its development.
7. Regulation (EU) 2024/1689, Article 113, as amended by Regulation (EU) 2026/1744: Chapter III Section 2 applies from 2 December 2027 for systems classified under Article 6(2) and Annex III, and from 2 August 2028 for systems classified under Article 6(1) and Annex I.
8. Articles 9 and 12 to 15 of Regulation (EU) 2024/1689 were not amended by Regulation (EU) 2026/1744. Article 10 was: paragraphs 10(1) and 10(6) were re-scoped, and paragraph 10(5), which permitted processing of special categories of personal data for bias detection, was deleted and relocated into a new Article 4a with wider scope. Article 11(1) was amended to extend the simplified technical documentation route to small mid-caps alongside SMEs and start-ups.
9. Automotive SPICE Process Reference Model / Process Assessment Model, Version 4.1, VDA QMC, 2026. The Automotive SPICE Guidelines are versioned separately from the PAM; the third edition, VDA QMC 2026, is issued for PAM 4.1 and replaces the previous edition. Assessor qualification schemes and customer contractual baselines transition on their own schedules, so the version that governs a given assessment should be confirmed with the customer or the certified assessor.
10. EUROCAE WG-114 deliverables list, consulted September 2026: ED-324 recorded as Draft, target publication 31/12/2026. Draft 7 completed EUROCAE Open Consultation in September 2025. The SAE designation is ARP6983; the AS6983 label used in some older material refers to the same project under an earlier classification. No published issue existed at the time of writing.
11. ISO 26262-8:2018, Clause 11, Confidence in the use of software tools. TI and TD classification determine the TCL; the clause sets out the permitted means of qualification at each level.
12. DO-178C §12.2 determines whether tool qualification is required and which tool criteria apply; RTCA DO-330 / EUROCAE ED-215 sets the objectives for each TQL. There is no TCL in the DO-178C framework and no TQL in ISO 26262 — the terms are not interchangeable.
13. Wilson score interval at 95 percent for 32 correct of 40: approximately 0.652 to 0.895. The normal-approximation interval is of similar width at this sample size but less reliable near the bounds.

References

Regulation (EU) 2024/1689 (Artificial Intelligence Act), as amended by Regulation (EU) 2026/1744 of 8 July 2026 (Digital Omnibus on AI), in force 27 July 2026 — high-risk obligations deferred to 2 December 2027 for Annex III systems and 2 August 2028 for Annex I product-embedded systems. NIST AI 100-1, Artificial Intelligence Risk Management Framework (AI RMF 1.0), 2023, and NIST AI 600-1, Generative Artificial Intelligence Profile, 2024. OWASP Top 10 for LLM Applications 2026, OWASP Gen AI Security Project, August 2026, superseding the 2025 edition. OWASP Top 10 for Agentic Applications for 2026, OWASP Gen AI Security Project, 9 December 2025. ISO/IEC 42001:2023, Information technology — Artificial intelligence — Management system. ISO 26262:2018, Road vehicles — Functional safety (12 parts). ISO/SAE 21434:2021, Road vehicles — Cybersecurity engineering. ISO 21448:2022, Road vehicles — Safety of the intended functionality. ISO/PAS 8800:2024, Road vehicles — Safety and artificial intelligence. ISO/IEC TR 5469:2024, Artificial intelligence — Functional safety and AI systems. ISO/TS 5083:2025, Road vehicles — Safety for automated driving systems. RTCA DO-178C / EUROCAE ED-12C, 2011, Software Considerations in Airborne Systems and Equipment Certification. SAE ARP4754B, 2023, Guidelines for Development of Civil Aircraft and Systems. SAE ARP6983 / EUROCAE ED-324 (in development; EUROCAE lists ED-324 as a draft with target publication 31 December 2026). EASA Artificial Intelligence Concept Paper, Issue 2, 2024, and Proposed Issue 03, released for comment June 2026 and not finalized at the time of writing. RTCA DO-330 / EUROCAE ED-215, 2011, Software Tool Qualification Considerations. FAA Roadmap for Artificial Intelligence Safety Assurance, Version 1, 2024. Automotive SPICE Process Reference Model / Process Assessment Model, Version 4.1, VDA QMC, 2026 (Version 4.0, October 2023). Automotive SPICE Guidelines, 3rd edition, VDA QMC, 2026, issued for PAM 4.1. SystemWeaver platform documentation.

About. SystemWeaver is a model-based systems engineering platform used in automotive, defense and aerospace programs. This paper sets out the architecture and governance patterns SystemWeaver recommends for AI-assisted work on engineering data.

Regulatory and standards positions are summarized as at September 2026 and are provided for information only. Nothing here is legal advice; confirm the classification and obligations that apply to your own systems with your counsel and your assessors. Standards and regulations cited are subject to revision.

© 2026 Systemite AB. Version 1.3, September 2026. SystemWeaver is a trademark of Systemite AB. Other marks are the property of their respective owners.