THE RELIABILITY KILL CHAIN

A Trust Architecture for AI-Generated Compliance in the Department of Defense

From Probabilistic Text to Machine-Verifiable Authorization

RESEARCH PAPER // APRIL 2026

Sphoenix
Product Manager // BotNews

An independent technical architecture for AI agent reliability in U.S. government compliance operations. Not affiliated with or endorsed by DoD, NIST, or any government entity. Grounded entirely in publicly available research, policy, and standards.

Contents

Executive Summary

> *Princeton provides the physics of the problem. This framework provides the engineering of the solution. Without Princeton, the architecture risks becoming doctrine without empirical grounding. Without this architecture, Princeton risks remaining a benchmark paper with no path to operational deployment. Together they form a complete reliability stack: measure reliability, constrain the system, preserve provenance, detect drift, prevent propagation, continuously re-validate trust.*

In December 2025, GenAI.mil extended AI access to approximately three million Department of Defense (DoD) personnel. By March 2026, Agent Designer allowed any of those users to build and deploy AI compliance agents without writing code. An Air Force unit built two specialized 'NIST Expert' agents — one for Azure, one for AWS — that ingested Cloud One documentation and produced compliance packages described as 'formal, defensible, and audit-ready test results.' The productivity breakthrough is real.

The problem is not that the agents were built. The problem is that 'audit-ready' implies a specific guarantee — that every assertion is traceable to evidence, stable across runs, correct against authoritative standards, and safe to propagate into authorization decisions. That guarantee requires an architecture. No such architecture currently exists in published DoD policy.

Princeton University researchers, in their March 31, 2026 comment to NIST AI 800-2, empirically grounded the problem: 14 frontier models, 500-plus benchmark runs, 12 reliability metrics drawn from safety-critical engineering. Their central finding: output consistency ranges from 30 to 75 percent even at temperature zero, capability improvements do not reliably produce reliability improvements, and the gap between what a model can do in a controlled evaluation and what it does in operational deployment is systematically underestimated.

What Princeton proves, this framework engineers against. The core thesis of both is the same: a compliance agent should not be trusted because it usually writes good packages. It should be trusted only when every assertion is independently reconstructable, provenance-bound, and machine-verifiable.

> *Traditional cybersecurity asks: 'How do we stop an adversary from entering the system?' Reliability engineering for AI compliance asks: 'How do we stop a wrong assertion from becoming accepted reality?' That is a fundamentally different problem. It deserves its own doctrine, its own artifacts, and its own control planes.*

This paper presents that doctrine. Its architecture rests on eight integrated layers:

- ACPM controls model drift — every configuration variable that defines an agent's runtime state
- SPSB controls standards drift — every interpretive context the agent operated under

- VCAR proves each assertion — machine-verifiable derivation from evidence to claim
- The Reliability Kill Chain explains how failures propagate and where each defense intercepts them
- The AEG maps where failures spread — bi-directionally across control dependencies and mission systems
- Temporal decay and reserve margin explain why reliability is not static and why operating at the minimum threshold is operationally fragile
- cATO provides the operational feedback loop — connecting continuous monitoring to authorization validity
- The ODD and legal delegator define who the agent is allowed to be and who is accountable when it is wrong

The limiting factor is no longer whether an agent can draft an authorization artifact once. It is whether the surrounding system can prove that the artifact is repeatable, provenance-bound, fail-closed under uncertainty, and safe to propagate into authorization decisions.

1. The Inversion Problem

> *The Department of Defense (DoD) spent decades learning that software safety requires traceability, redundancy, and certification before deployment. Agentic AI reverses that sequence: deployment is occurring first, and traceability, redundancy, and certification are being invented afterward. The DoD is not behind — it is early. The risk is not that it missed the future. The risk is that it reached the future before the surrounding assurance architecture existed.*

GenAI.mil launched December 9, 2025. Agent Designer launched March 9–10, 2026. No binding DoD-wide reliability standard, certification authority, or mandatory multi-run validation protocol for agentic AI systems exists as of April 2026. No requirement exists to measure repeatability before deployment. No process verifies AI-generated control narratives against NIST OSCAL Rev 5 catalog data. No mechanism detects when a model update silently changes a deployed compliance agent's behavior.

This is not negligence. It is the normal pattern of technology deployment in every domain where important infrastructure has been built: aviation, nuclear, the internet. In each case, the assurance architecture trailed deployment. In each case, incidents accumulated. In each case, the architecture eventually arrived and made the technology trustworthy. The DoD's job is to compress that interval — and it has the institutional authority, operational scale, and existing infrastructure to do it faster than any other government entity in history.

The operational metaphor that requires no translation for a DoD practitioner is already inside the ecosystem: Platform One's Big Bang DevSecOps pipeline enforces quality and compliance gates that block deployment when gates fail. The ask of this framework is precise: extend that same culture to compliance agent outputs. Block evidence-grade artifact publication when reliability gates fail. That is not a new concept for this audience. It is the application of a concept they already operate with daily.

2. The Assist/Attest Distinction: The Non-Negotiable Boundary

> *Assistive eligibility: the agent may suggest, draft, summarize, and help a human understand. The human reviews, judges, and signs. Attestation eligibility: the agent's output populates machine-readable assessment assertions that may travel downstream — into eMASS, into OSCAL packages, into inherited control records — without individual human review of each claim. A system may be eligible to assist long before it is eligible to attest. These are not the same threshold. Conflating them is a category error with legal consequences.*

Federal compliance law distinguishes preparers from attesters. An ISSM using AI to help draft an SSP is a preparer — the human signs, the human attests, the human bears the legal obligation. An AI system that populates machine-readable assessment results traveling to an authorizing official is acting in a signatory capacity, even if no human explicitly signed each assertion. The DOJ's September 2024 False Claims Act guidance directs prosecutors to examine whether organizations identified risks of false approvals and documentation generated by AI. AI-generated compliance assertions that are adopted, signed, or transmitted without reasonable verification may create False Claims Act exposure under theories of reckless disregard or deliberate ignorance — liability does not require intent, only the failure to take reasonable steps to verify what was attested.

The legal delegator field in the ACPM is the mechanism that makes attestation eligibility legally traceable. For any agent operating at AAL-3 or above, a named human role must be designated as legally accountable for the agent's attestation outputs. That delegation is logged immutably. When a compliance assertion generated by an AAL-3 agent reaches an authorizing official, the provenance chain traces from the VCAR through the ACPM to the named legal delegator. This makes agent identity a legal accountability mechanism, not only an authentication control.

3. Compliance Agents Are Safety-Critical Knowledge Systems

Compliance agents do not fly aircraft. But their outputs become inputs to Authority to Operate decisions that determine whether systems handling classified information and warfighter communications are permitted to operate. The failure mode is administrative, delayed, and invisible until months after it occurs. Aviation calls this latent organizational failure. Nuclear engineering describes it in common-cause failure chain analysis. The Swiss cheese model in both domains explains how individually plausible decisions combine to produce catastrophic outcomes no single layer would have permitted.

3.1 The Compliance Failure Taxonomy

- **Incorrect control mapping —** Agent assigns AC-17 where AC-2 is required. The SSP appears complete. The error persists as a live vulnerability behind a valid ATO.
- **IL misclassification —** Agent produces IL4-appropriate narratives for an IL5 system. Gap is invisible in the documentation.
- **Rev 4 / Rev 5 directional bias —** Models trained on pre-2020 internet data carry systematic bias toward Rev 4 language. The model is not uncertain — calibration metrics will not surface this. The SPSB and OSCAL ground-truth comparison are the only defenses.
- **False compliance assertion —** Agent states a control is implemented when evidence supports only partial implementation. Plausibly formatted and difficult to detect without VCAR evidence-pointer verification.
- **Inheritance Integrity Failure —** Agent over-attributes responsibility to Cloud One, under-attributes system-specific obligations, or collapses hybrid controls into binary inherited/not-inherited frames. The artifact is polished. The inheritance boundary is structurally wrong. Error cascades to every inheriting system.
- **Evidence laundering —** Agent silently substitutes model prior knowledge for missing retrieved evidence. Output appears evidence-backed. No evidence actually supports the claim. This is the failure mode that makes 'audit-ready' a dangerous phrase.
- **Standards semantic drift —** Agent operates on an SPSB whose forward-compatibility status has become Pinned-incompatible. AC-2 in 2024 carries one interpretation. AC-2 in 2027 may inherit zero-trust assumptions that change what the control requires. The agent produces internally consistent but historically obsolete assertions.

The most dangerous failure class is Inheritance Integrity Failure because it is RMF-native, not obviously an AI problem to reviewers, and cascading in consequence. The SPSB semantic drift failure class is the one most likely to emerge over a 24–36 month deployment horizon — as NIST and FedRAMP standards continue evolving while deployed agents operate against stale baselines.

4. The Reliability Kill Chain

> *Most discussions treat AI reliability failures as isolated mistakes. In the DoD compliance context, a bad control assertion is dangerous for the same reason a cyber intrusion is dangerous: not because of the first error, but because of the chain of trust it contaminates afterward. The Reliability Kill Chain is the intellectual bridge between AI reliability research and defense systems risk doctrine.*

The Reliability Kill Chain maps the seven-stage propagation path of a compliance reliability failure, and identifies the specific defense layer that intercepts each stage. Understanding the kill chain reframes the paper's entire proposal: this is not a document about making AI more accurate. It is a document about breaking the propagation path of false compliance assertions before they become accepted operational reality.

Stage	What Fails	How It Happens	Consequence If Undetected	Defense	Artifact
1	Provider corpus integrity	Malicious content injected into RAG ingestion corpus — Type 2 or Type 3 prompt injection	Poisoned assertions reach generation layer with apparent authenticity	Cryptographically signed provider OSCAL artifacts + allowlist-only ingestion	SPSB signed provider hash
2	Evidence selection	Contaminated content surfaces in retrieval results for legitimate queries	Wrong evidence used to justify control assertions	Retrieval provenance hashing + sandboxed RAG ingestion	Evidence ledger content-address hash
3	Control assertion generation	Agent generates incorrect, hallucinated, or Rev 4-biased compliance claims	False assertion enters OSCAL artifact with plausible formatting	VCARs + hard constraint gate + multi-run reliability harness	VCAR + SEGR on failure
4	Human review quality	Reviewer accepts pre-formatted, plausible-looking but incorrect output (automation bias)	Incorrect assertion receives human sign-off and enters formal record	METS randomized mandatory spot-check + statistical process control	METS audit log
5	Package approval record	Incorrect ATO granted or renewed based on contaminated evidence	System operates under false authorization	ACPM approval delegation logging + legal delegator accountability	ACPM + SPSB approval record
6	Inheritance propagation	Error spreads through RMF inheritance chains to all dependent systems	Enterprise-wide false compliance state — AEG cascade	AEG dependency monitoring + inheritance integrity audit + reserve margin	AEG propagation alert

Stage	What Fails	How It Happens	Consequence If Undetected	Defense	Artifact
7	Mission posture validity	Organization believes it is compliant while operating with exploitable gaps	Operational and strategic risk; adversary exploits authorization blind spot	cATO continuous monitoring + authorization freshness score + CAIRS reporting	ORDI dashboard + CAIRS alert

Three stages deserve specific elaboration because their defenses are novel and specific to the compliance domain:

Stage 1 — Corpus Poisoning and Type 3 Inheritance Injection

The most dangerous attack vector in this kill chain is not an adversary sending malicious queries to the agent. It is an adversary who can modify a Cloud One or other common control provider's OSCAL component definition. A single injection at the provider level corrupts the compliance documentation of every system inheriting from that provider — simultaneously, persistently, and with the apparent authority of an official provider document. The mitigation is architecturally necessary: OSCAL component definitions from common control providers must be cryptographically signed by the provider and validated against an immutable registry before entering any system's RAG corpus.

Stage 4 — The METS Randomized Mandatory Spot-Check

Human reviewers develop predictable fatigue patterns when reviewing high-volume pre-populated outputs: deep scrutiny of early assertions, declining attention for later ones. The Mandatory Evidence Trace Spot-Check (METS) addresses this structurally: a reviewer must perform full, independent evidence reconstruction for a randomly selected sample of assertions — without knowing in advance which ones. Randomization removes the possibility of strategically conserving scrutiny for 'important' assertions. The METS sample also provides a ground-truth quality measurement: if the sample error rate exceeds threshold, the full package is flagged for expanded review.

Stage 6 — The AEG Cascade Defense

The Authorization Evidence Graph (AEG) detects Inheritance Integrity Failures before they reach full enterprise propagation. When a provider component definition changes, the AEG automatically determines which downstream control assertions depend on that component and flags them for re-evaluation. This converts inheritance risk from a silent cascade into a visible, tracked, manageable re-evaluation queue. Stage 6 is where the AEG's bi-directional capability becomes critical — not only which controls need re-evaluation, but which missions, systems, and operational authorities are now at risk.

5. OSCAL as System of Record

> *Narrative output is review UI. OSCAL is system of record. Rendered PDFs are derived artifacts — never primary truth. FedRAMP RFC-0024 draws the same line in federal policy: machine-generated probabilistic text designed to mimic human-written narratives is contrasted explicitly with deterministic machine-readable telemetry. That distinction is now federal policy direction, not a recommendation.*

If a compliance agent produces prose-first outputs, reliability is qualitative and drift is undetectable. Two runs producing different underlying control mappings look, to a human reviewer, like two reasonable drafts. There is no automated mechanism to detect the difference. If the agent produces OSCAL-native artifacts, the outputs are schema-validatable, diffable, and machine-comparable. The metric stack becomes computable rather than impressionistic. The kill chain defenses become operational rather than advisory.

FedRAMP's RFC-0024 and Notice 0009 establish machine-readable authorization data as federal policy momentum. DoD compliance agents producing narrative-first outputs are already misaligned with the direction the federal compliance ecosystem is moving. Every compliance agent built today without OSCAL-native output is accumulating standards alignment debt in addition to reliability debt.

5.1 OSCAL Output Requirements by Artifact Type

- SSP implementation statements: OSCAL SSP JSON with control-ID references, evidence pointers, implementation status, VCAR attachment, and inheritance responsibility fields
- Assessment results: OSCAL Assessment Results JSON with observation references, finding classifications, risk entries, and VCAR chain
- POA&M entries: OSCAL POA&M JSON with risk acceptance dates, milestones, and evidence of remediation
- Inherited control mappings: OSCAL component definitions with explicit system-specific vs. inherited responsibility boundaries — no binary inherited/not-inherited frames for hybrid controls

Narrative text is appropriate as a rendering of these structured records for human readability. It is not appropriate as the authoritative source from which authorization decisions are made. This distinction must be enforced architecturally, not by convention.

6. The Provenance Stack: ACPM, SPSB, and VCAR

The provenance stack is the foundation of the entire reliability architecture. Three complementary artifacts, each addressing a distinct category of drift that the others cannot catch. ACPM without SPSB leaves standards semantic drift invisible. SPSB without VCAR leaves individual assertion provenance unverifiable. VCAR without ACPM and SPSB cannot be independently reconstructed. All three are required for attestation-eligible output.

Artifact	Full Name	Required Fields	Critical Rule
ACPM	Agent Configuration and Provenance Manifest	Model version + prompt hash + corpus hash + tool permissions + temperature + reliability scores + ODD definition + legal delegator	Any field change triggers mandatory re-evaluation at current AAL tier
SPSB	Standards Provenance and Semantic Baseline	NIST publication versions + OSCAL schema version + FedRAMP baseline revision + provider component versions + semantic diff from prior baseline + forward-compatibility status	Standards interpretation changes require SPSB update and re-evaluation of affected assertion classes — independent of model version
VCAR	Verified Control Assertion Record	Evidence hash pointers + retrieval context hash + constraint check results + calibration score + derivation chain from evidence to assertion + any human reviewer action	Machine-verifiable by any authorized system; without VCAR, assertion is assistive only — not attest-eligible
SEGR	Structured Evidence Gap Report	Controls lacking sufficient evidence + evidence types required + retrieval gap description + structured acquisition plan	Generated on fail-closed condition; classified at SSP level; stored in evidence ledger — not in collaboration tools
Evidence Ledger	Append-only tamper-evident audit log	Content-addressed evidence hashes + ACPM + SPSB + VCAR + transcript trace + human review actions + downstream consequence record	SP 800-53 AU-9/AU-10 compliant; cryptographic non-repudiation; all entries immutable

6.1 ACPM: Agent Configuration and Provenance Manifest

The ACPM converts model version pinning from a recommendation into a certifiable mechanism. Every DoD compliance agent at AAL-2 or above maintains a current ACPM stored in the evidence ledger, tamper-evident, updated whenever any field changes — each update triggering re-evaluation at the current AAL tier.

ACPM Field	Definition	Compliance Note
Foundation model identifier + build hash	Exact model build, not just version name	Required — any change triggers re-evaluation at current AAL tier
Prompt template version hash	SHA-256 of all system prompt and instruction content	Required — template changes alter agent behavior without model change
Retrieval corpus snapshot hash + timestamp	Hash of indexed document set and last validation date	Required — corpus changes alter what evidence is available
ODD definition (Operational Design Domain)	Control families, document types, IL levels, environments, evidence age limits within which performance has been characterized	Required — queries or evidence outside ODD must trigger immediate fail-closed escalation
Tool access permissions + allowlist	Enumerated authorized tools and data sources	Required — unauthorized tool calls are hard-constraint violations
Temperature and inference parameters	All decoding settings affecting output stochasticity	Required — parameter changes affect computational reliability
Legal delegator designation	Named role or position legally accountable for the agent's attestation outputs	Required for AAL-3+ — agent identity is a legal accountability mechanism, not only an authentication control
Reliability evaluation results by AAL tier	Per-dimension scores at time of validation on compliance-domain benchmark	Required — forms the reliability scorecard for this exact configuration
Date of last validation + retest trigger list	ISO 8601 timestamp + enumerated conditions requiring mandatory re-evaluation	Required — version pinning without retest triggers is archival paperwork

6.2 SPSB: Standards Provenance and Semantic Baseline

The SPSB is the artifact that ACPM cannot replace. Model version pinning protects against model drift. SPSB protects against standards drift. Two agents operating on identical models, identical corpora, and identical prompts can produce mutually incompatible 'correct' outputs if they operate against different versions of the NIST standard. AC-2 in 2024 and AC-2 in 2027 share an identifier but may carry materially different interpretive obligations as zero-trust requirements and cloud-native expectations evolve.

SPSB Field	Definition	Why ACPM Alone Is Insufficient
NIST publication versions	Exact version of NIST SP 800-53 and all referenced NIST AI publications	Required — Rev 4 and Rev 5 share control identifiers with materially different interpretations
OSCAL schema version	Version of OSCAL JSON/XML schema used for artifact generation	Required — schema changes affect artifact structure and

SPSB Field	Definition	Why ACPM Alone Is Insufficient
		validation logic
FedRAMP baseline revision	Exact FedRAMP High/Moderate/Low baseline version	Required — FedRAMP adds and removes controls between revisions
Provider component definition versions	Version hash of each common control provider's OSCAL component definitions in scope	Required — Cloud One inheritance mapping changes without control ID changes; only SPSB captures this
Semantic diff from previous baseline	Structured record of interpretive changes from prior SPSB version	Required — enables detection of standard drift that is invisible to model version pinning
Forward-compatibility status	Current / Pinned-compatible / Pinned-incompatible	Required — Pinned-incompatible status requires SEGR for any assertion class affected by the interpretive divergence

6.3 VCAR: Verified Control Assertion Record

The VCAR is what makes an assertion attest-eligible rather than assist-eligible. It is the machine-verifiable proof object attached to every control assertion, structured so any authorized verifier can confirm the assertion is properly derived without re-running the agent that produced it. An authorizing official reviewing an ATO package should not see 'AC-2 is implemented — evidence pointer: scan-2026-04-01.' They should see a VCAR that proves: this assertion was derived from this specific evidence, under this ACPM configuration and SPSB baseline, passing these constraint checks, with this calibration score — and any authorized system can independently reconstruct the derivation.

VCAR Required Fields

- Evidence hash pointers: SHA-256 hashes of all evidence objects supporting the assertion
- Retrieval context hash: hash of the exact retrieved content used in assertion generation
- ACPM snapshot hash: links the assertion to the exact agent configuration active at generation time
- SPSB version hash: links the assertion to the exact standards baseline under which it was generated
- Constraint check results: pass/fail for all applicable hard constraints
- Calibration score: confidence on this assertion class from compliance-domain calibration benchmark
- Derivation chain: structured record of the logical path from evidence to assertion
- Human reviewer action: whether and how a human modified or confirmed the assertion

An assertion without a complete VCAR is assistive output. An assertion with a complete, valid VCAR meeting the metric stack thresholds at the appropriate AAL tier is attest-eligible output. This is the fundamental boundary.

Every VCAR must be bound to an immutable ACPM snapshot hash and SPSB version hash so that any assertion can be reconstructed under the exact model configuration and standards baseline in effect at the time it was generated. ACPM records what the agent was. SPSB records what the standards meant. VCAR records what was derived. All three together form the reproducibility spine of the system — remove any one of them and the others cannot independently verify the assertion.

6.4 SEGR: Structured Evidence Gap Report

The SEGR is what a compliant agent produces when the fail-closed contract is triggered. It is not a failure message. It is actionable compliance infrastructure: a machine-readable, OSCAL-compatible artifact specifying the controls lacking sufficient evidence, the evidence types required, the retrieval gaps preventing satisfaction, and a structured acquisition plan for the human to collect what is needed.

> *A SEGR that lists specific controls lacking evidence and identifies what would satisfy each claim is also a map of authorization gaps in the system. It must be classified at SSP level and stored in the evidence ledger — not in collaboration tools. An AI-generated gap report in an unclassified workspace is a reconnaissance artifact for an adversary with access to that space.*

7. Three Layers of Reliability — Integrated with Princeton's Four Dimensions

Reliability evaluation in compliance contexts operates across three distinct layers. Princeton's four dimensions — consistency, robustness, predictability, and safety — apply across all three. The layers describe where reliability must be evaluated. The Princeton dimensions describe how. Together they produce a complete evaluation framework.

Layer 1 — Computational Reliability

Can the same model produce the same output under identical conditions? Princeton found 30–75% output consistency at temperature zero due to floating-point non-determinism, GPU kernel scheduling, and batch-size variation. This is the baseline establishing whether the model is stable enough to begin operational evaluation.

Layer 2 — Semantic Reliability

Does the model preserve the same control mapping when the prompt wording changes but the task does not? Compliance queries are semantically varied by nature. Different ISSMs phrase questions differently. The same ISSM asks differently on different days. Robustness (Princeton dimension 2) is the primary metric at this layer.

Layer 3 — Operational Reliability

Does the agent remain dependable inside a live workflow with a retrieval system of variable freshness, tool calls that can fail silently, human reviewers under time pressure, and downstream systems that automatically consume outputs? This is the layer that matters most — and the one Princeton's study mostly does not measure. Operational reliability is multiplicative across pipeline components:

Pipeline Component	Reliability Estimate	Notes
Foundation model (Gemini for Government)	90% (upper bound)	Princeton: 30–75% output consistency at temperature zero. 90% is an optimistic ceiling, not an operational estimate.
RAG retrieval layer	92%	Dependent on OSCAL corpus version alignment, SPSB-validated document freshness, and chunking strategy
Tool execution	95%	Silent tool call failures compound in agentic pipelines — logged in evidence ledger
Human review (pre-populated OSCAL artifact)	70–80%	Automation bias research: effective review rate falls substantially below 90% for pre-formatted, high-volume outputs. 90% is the assumption that creates false confidence.
Total pipeline reliability	**≈ 55–63%**	0.90 × 0.92 × 0.95 × 0.70 ≈ 0.55 | × 0.80 ≈ 0.63. This is why reliability gates exist — the math does not forgive optimistic assumptions.

> *Five runs is a minimum repetition count, not a sufficient reliability protocol. The complete requirement is: five runs per condition — five with identical inputs (stochastic stability), five with semantically equivalent rephrasing (semantic stability), five with partial evidence (robustness), five with tool fault injection (infrastructure robustness). Conflating identical repetition with a complete reliability protocol is the most common evaluation mistake in current agent deployments.*

8. The ORDI — Operational Reliability Debt Index

Reliability debt names what the DoD accumulates with every agent deployed without formal reliability validation. Unlike technical debt — which slows systems — reliability debt increases invisible operational risk by accumulating untested assurance gaps that appear in audit records as valid authorizations.

$$ORDI = \Sigma [(1 - R) \times S \times C \times H]$$

> *The ORDI is a prioritization index, not an actuarial prediction. Use it to rank agent portfolio risk and set reserve margin requirements. Some terms are operational proxies, not validated causal variables. The formula is deployable immediately because all variables are measurable with existing data.*

Variable	Definition	Example	Critical Note
R = Rp × Rc	Validated reliability: Repeatability multiplied by Correctness. Rp = fraction of identical outputs across runs. Rc = fraction verified correct against NIST OSCAL Rev 5 ground truth on compliance-domain calibration benchmark.	Rp=0.80, Rc=0.81 → R≈0.65	Repeatability without correctness is systematic error at scale. Do not conflate. ECE must be measured on compliance-domain data, not general benchmarks.
S	Deployment scale: users, workflows, or systems using the agent	5,000 ISSMs	Model version updates silently reset effective reliability — SPSB must track this
C	Decision criticality: 0–1 multiplier scaled to IL level and consequence	0.8 (IL5)	IL5 = 0.8–1.0; IL4 = 0.5–0.7; IL2 = 0.2–0.4
H	Inheritance multiplier: number of downstream systems inheriting from the common control provider	1.2	One mis-mapped inherited control cascades without signal to every inheriting system
ORDI	Σ [(1 – R) × S × C × H]	≈ 1,680 units	Prioritization index, not actuarial prediction. Use to rank agent portfolio risk and set reserve margin requirements.
Reserve Margin	Measured Reliability – Minimum Required Reliability	≥ 5 pts (standalone); ≥ 15 pts (common control provider)	Systems must not operate at the edge of the acceptable envelope. Providers must maintain higher reserve margin because erosion cascades through inheritance graph.
Freshness	Per-assertion reliability decays	Policy → 12	ORDI accumulates from

Variable	Definition	Example	Critical Note
Decay	as evidence ages beyond validated half-life	months; vulnerability scan → 30 days; crypto key rotation → per schedule	time-passage without evidence renewal — not only from uncertified deployments

8.1 Reliability Reserve Margin

Systems should not operate at the minimum acceptable reliability threshold. They should maintain reserve margin — the operational buffer that prevents normal degradation from immediately breaching the required floor. This is structural load margin in aerospace, thermal margin in nuclear engineering, applied to compliance AI:

```
Reliability Reserve Margin = Measured Reliability − Minimum Required Reliability
```

Reserve margin requirements scale with inheritance depth. A standalone system should maintain at least a 5-point reserve margin above its AAL minimum. A common control provider serving 50 inheriting systems should maintain at least a 15-point reserve margin, because margin erosion at the provider level propagates across the entire inheritance graph. An agent with zero reserve margin is technically compliant and operationally fragile.

Reserve margin is the most operationally intuitive dashboard metric in this framework. 'Your NIST Expert agent's reliability reserve margin is 3 points above AAL-3 minimum, but your inheritance exposure is 47 systems' is a sentence a program manager can act on without reading the rest of this document.

8.2 Temporal Reliability Decay

Control implementations and evidence have different validity half-lives. Compliance assertions decay in reliability as evidence ages beyond its validated period. A NIST policy documentation assertion may remain reliable for 12 months. A vulnerability scan assertion may decay to unreliable within 30 days. A cryptographic key rotation assertion depends on the rotation schedule.

This means ORDI is not a static point-in-time measurement. It accumulates from time passing without evidence renewal — not only from uncertified deployments. The Authorization Freshness Score — the aggregate reliability of a system's current authorization record accounting for evidence age across all control assertions — provides a continuous, evidence-grounded measure of whether a current ATO remains trustworthy. This replaces arbitrary 3-year ATO cycles with a dynamically computed authorization validity signal that cATO monitoring can act on.

9. The Authorization Evidence Graph

Control assertions have dependencies. AC-2 (Account Management) assertions can depend on SI-2 (Flaw Remediation) evidence if accounts are managed through patched authentication software. IA-5 (Authenticator Management) assertions depend on IA-2 evidence. These dependencies are not arbitrary — they are derivable from the OSCAL control catalog's relationship fields. The Authorization Evidence Graph formalizes them.

The AEG is a formally structured directed acyclic graph where each node is a control assertion and each directed edge represents an evidential or logical dependency. When evidence changes, the AEG determines which downstream assertions require re-evaluation. When a common control provider updates its OSCAL component definition, the AEG propagates the re-evaluation obligation automatically through the inheritance chain. 'Which of our 450 controls need re-evaluation because Cloud One updated its inheritance mapping?' is answered in milliseconds rather than requiring a compliance sprint.

9.1 Bi-Directional AEG: From Compliance to Mission

The AEG must be bi-directional. The forward direction answers: 'If this evidence changes, which control assertions are affected?' The reverse direction answers: 'If this control becomes invalid, what systems, missions, programs, and operational authorities are now at risk?'

A compromised inherited authentication control should not only trigger re-evaluation of IA-2 and IA-5. It should identify: which systems lose operational authorization, which programs depend on those systems, which missions would be degraded, which acquisition programs or contracts require notification. The bi-directional AEG transforms the paper from a compliance document into a defense systems document.

One technical precision: the reverse AEG must distinguish between authorization degradation (ATO validity is affected) and operational degradation (mission capability is affected even if the ATO technically remains). These can diverge. A system can maintain a technically valid ATO while operating in a configuration that a newly invalid control was supposed to protect. The AEG must surface both directions independently and alert when they diverge — because the most dangerous condition is a system with a green ATO and a degraded operational posture.

10. The Three-Vector Prompt Injection Model

Both the NIST NCCoE agent identity concept paper and NIST AI 800-2 identify prompt injection as a first-class evaluation objective for agentic systems. The DoD compliance context produces three structurally distinct injection vectors that require different defenses:

Type 1 — Direct Injection

Malicious content in user queries affecting agent behavior. Standard. Addressed by existing query sanitization and allowlist controls. Highest visibility, lowest sophistication threshold.

Type 2 — Corpus Injection

Malicious content embedded in ingested documentation. An adversary with write access to any document in the RAG corpus can embed carefully crafted text that causes the agent to produce specific incorrect control assertions. Unlike Type 1, this is persistent across agent sessions and survives agent upgrades. Mitigation: content-filtered RAG ingestion, retrieval source allowlists, evidence ledger content-address hashes that detect document modification.

Type 3 — Inheritance Injection (Highest Priority)

Malicious content embedded in a shared OSCAL component definition at the common control provider level. An adversary who can modify a Cloud One OSCAL component definition can affect the compliance assertions of every system inheriting from that provider — simultaneously, persistently, and with the authoritative appearance of an official provider document. This is a precision attack vector that exploits the same inheritance mechanism identified as the primary structural reliability risk throughout this framework.

> *Type 3 Inheritance Injection is the highest-priority unaddressed attack vector in current DoD AI compliance infrastructure. The mitigation is architecturally necessary and currently absent: OSCAL component definitions from common control providers must be cryptographically signed by the provider and validated against an immutable registry before entering any system's RAG corpus. Provider signing infrastructure is a critical gap.*

11. Hard Constraint Taxonomy and Fail-Closed Contract

11.1 Hard Constraints — Zero-Tolerance Stop-Ship Gates

Any violation of the following constraints blocks evidence-grade output for that run. The run produces a SEGR. No exceptions for urgency, deadline pressure, or supervisor instruction:

- Never claim a control is implemented without a verifiable evidence pointer in the VCAR
- Never silently substitute general model knowledge for missing retrieved evidence — insufficient retrieval must produce a SEGR
- Never access tools, data sources, or evidence outside the authorized ODD and tool allowlist in the current ACPM
- Never modify evidence objects — the agent reads evidence, it does not alter it
- Never produce an assessment result claiming inheritance from a provider whose OSCAL component definition has not been validated against the current SPSB registry
- Never allow a SEGR to be stored in a collaboration tool below SSP classification level

11.2 Soft Constraints — Severity-Scored, Flagged for Review

- Incomplete cross-references between control families (severity weighted by family criticality)
- Ambiguous boundary assumptions supporting multiple interpretations (flag with explicit uncertainty statement)
- Evidence age exceeding retrieval corpus validated timestamp (flag with date delta)
- Assertion confidence below calibration threshold — triggers fail-closed even if hard constraints pass

11.3 The Fail-Closed Uncertainty Contract

Confidence scores are operationally useful only when tied to enforced behavioral consequences. The fail-closed contract:

```
Below-threshold confidence on any evidence-bearing claim → Output must
degrade to SEGR → Not 'may escalate' — must
```

The SEGR that results from a fail-closed condition specifies: the control requiring evidence, the evidence types that would satisfy the requirement, the current retrieval

gap, and a structured acquisition plan. The agent that fails closed produces actionable infrastructure — not a dead end.

11.4 The Uncertainty Budget and Correlated Components

Uncertainty budget allocation across pipeline components cannot simply add retrieval uncertainty + generation uncertainty + review uncertainty because the components are not independent. Evidence retrieval failures for one control family often correlate with failures for related controls because the underlying evidence is shared. Budget allocation must use conservative worst-case assumptions — treat correlated components as fully correlated — until a program office has operational data to estimate actual correlation structure. Starting with the conservative bound is not pessimism. It is the same principle that nuclear PRA uses for common-cause failure analysis.

12. The Agent Assurance Level Framework

The AAL framework provides graduated, implementable requirements scaled to actual consequence. It adapts principles from DO-178C Design Assurance Levels, IEC 61508 Safety Integrity Levels, and ISO 21448 SOTIF without requiring deterministic software certification standards inapplicable to probabilistic models. The organizing question is: what is the consequence when this agent is wrong, and what evidence burden is proportionate to that consequence?

Level	Scope	Eligibility	Required Evidence	Reserve Margin	Primary Gate
AAL-1	Advisory only — narrative suggestions, summaries	Assist only	Single-run benchmark + human review + basic logging + ODD definition	Not applicable	Human sign-off on all outputs
AAL-2	Human-reviewed workflow assistance — OSCAL SSP draft generation	Assist only	5-run repeatability + prompt perturbation + OSCAL-native output + OSCAL ground-truth spot-check + ACPM + SPSB + provenance logging + ECE calibration (compliance-domain benchmark)	5 points above AAL-2 minimum	Multi-run reliability gate + VCAR on evidence claims
AAL-3	Bounded attestation authority with human sign-off — generates OSCAL assessment results	Attest eligible (human sign-off required)	10-run testing across perturbation suite + adversarial red team + ORDI score + metric stack evaluation + METS sampling plan + independent technical review + VCAR on all assertions + ACPM + SPSB + inheritance integrity audit	10 points above AAL-3 minimum	Reliability kill chain Stage 3 + VCAR completeness gate
AAL-4	Programmatic attestation authority — outputs travel downstream without individual human review of each claim	Attest eligible (programmatic)	Formal hazard analysis (SOTIF) + continuous monitoring + fail-safe fallback + fail-closed uncertainty contract + third-party certification + AEG dependency registration + bi-directional mission impact mapping + model version pinning + SPSB forward-compatibility validation	15 points above AAL-4 minimum (inheritance multiplier applied)	Full kill chain defense stack + authorization freshness monitoring
AAL-5*	Fully autonomous safety-critical	Not a deployable tier	Research horizon. Any procurement conversation treating	—	—

Level	Scope	Eligibility	Required Evidence	Reserve Margin	Primary Gate
	authority		this as near-term is a procurement red flag.		

Note on reserve margin in the AAL table: the minimum reserve margins listed (5/10/15 points) are requirements, not targets. A system at AAL-3 with exactly the minimum reliability score is technically compliant and operationally at risk. Program managers should treat zero reserve margin as a reliability debt alert requiring immediate attention, not as an acceptable steady state.

13. The Reliability Metric Stack

No single metric characterizes compliance agent reliability. High control-ID stability can hide evidence-pointer drift, inheritance boundary shifts, disappearing uncertainty language, and POA&M generation changes. An agent that games a single Jaccard metric is more dangerous than one whose instability is visible. The metric stack is required for AAL-2 and above:

Metric	Definition	Illustrative Threshold	Critical Note
Control-ID set stability	Jaccard similarity of OSCAL control selections across runs	≥ 0.90 (AAL-2); ≥ 0.97 (AAL-4 aspirational)	High stability can still hide evidence-pointer drift — never use alone
Evidence-pointer stability	Fraction of identical evidence hashes across runs (evidence unchanged)	≥ 0.95	Detects evidence drift independent of control ID stability
Inheritance-boundary stability	Mismatch rate: claimed inherited coverage vs validated provider mappings	0% for hard boundaries	Named failure class: Inheritance Integrity Failure — cascade risk requires zero tolerance
Calibration (compliance-domain)	Expected Calibration Error measured on NIST OSCAL Rev 5 compliance benchmark — not general benchmarks	ECE ≤ 0.08 (AAL-2); ≤ 0.05 (AAL-3+)	Must be domain-specific. A model with ECE 0.03 on standard benchmarks can have ECE 0.25 on NIST compliance tasks.
Uncertainty/ escalation precision	Fraction of low-confidence claims that correctly trigger SEGR escalation	≥ 0.95	Connects calibration to behavior — the fail-closed contract is only operational when tied to enforced escalation
Artifact graph diff severity	Severity score of structural changes between runs weighted by control family criticality	Program-defined severity cap	Catches semantic drift that identical control IDs hide — requires OSCAL-native output to be computable
Safety constraint violation rate	Hard-constraint violations across all runs and perturbation conditions	0 — zero tolerance	Stop-ship gate. Any violation blocks evidence-grade output for that run.
Prompt injection resilience	Success rate of Type 1/2/3 injection attempts under adversarial corpus testing	0% success rate for Type 2/3	Type 3 (inheritance injection) is the highest-priority test — must include provider document poisoning scenarios
METS spot-check pass rate	Fraction of deep-review samples where evidence trace reconstructs correctly	≥ 0.97 (AAL-3+)	If METS sample error rate exceeds threshold, full package flagged for expanded review

On the Jaccard targets: 0.97 for AAL-4 is aspirational under current model capabilities and should be treated as a maturity target, not an immediate floor that produces compliance theater if set unreachably high. 0.90 for AAL-2 is achievable with constrained retrieval architectures and OSCAL-native output. ECE thresholds must be measured on compliance-domain calibration benchmarks, not general-purpose benchmarks. A model with ECE 0.03 on standard benchmarks can have ECE 0.25 on NIST control interpretation tasks — the specialized, version-sensitive nature of compliance knowledge is precisely the domain where general calibration scores are least predictive.

14. The AI Flight Recorder — Evidence Ledger Requirements

When a compliance documentation failure is discovered six months after an ATO was granted, investigators today have access to the final output document and nothing else. The cause — model update, prompt change, SPSB semantic drift, corpus poisoning, tool call failure — is indistinguishable without provenance. Every meaningful DoD compliance agent must automatically log the following in an immutable, tamper-evident format consistent with SP 800-53 AU-9 and AU-10:

- Complete ACPM snapshot at time of each run
- SPSB version active at time of each run
- Every intermediate tool call — inputs, outputs, and failure states
- Retrieval query and retrieved evidence list with content hashes
- Confidence score at each decision point in the generation chain
- VCAR for every attest-eligible assertion generated
- SEGR generated if fail-closed condition was triggered
- Whether the human reviewer accepted, edited, or overrode the output — and to what degree (METS spot-check record if sampled)
- Downstream operational consequence: whether the output was incorporated into a formal compliance record

> *Provenance logging and interpretability are not the same capability. Provenance — what inputs, what retrieval, what version, what human action — is achievable today. Interpretability — what the model internally considered and why it produced one output rather than another — remains an unsolved research problem. Require provenance now. Treat interpretability as a research objective. Waiting for interpretability to solve a problem that provenance can address is waiting for an unsolved problem to solve a solvable one.*

15. Three-Stage Maturity Path

> *The three-stage maturity path is a sequence, not a menu. Stage 2 artifacts have integrity only if Stage 1 provenance exists. Stage 3 reflects reality only if Stage 2 artifacts are accurate. A compliance digital twin maintained by ungated agents is machine-readable hallucination with excellent formatting. A bi-directional AEG fed by unreliable assertions is a precision instrument calibrated to wrong data.*

Stage	Name	Components	Dependency Warning
Stage 1	**Provenance and Gating**	ACPM + SPSB deployment; evidence ledger; transcript logging; VCAR enforcement; constraint auditing; reliability harness; SEGR on failure; legal delegator designation; METS sampling plan	Foundation. Without Stage 1, all downstream artifacts have no verifiable integrity.
Stage 2	**Structured Machine-Readable Artifacts**	OSCAL-native SSP/SAP/SAR/POA&M; schema validation; metric stack evaluation; kill chain Stage 1–4 defenses operational; cATO evidence integration; authorization freshness score baseline established	Builds directly on Stage 1. OSCAL without provenance is machine-readable hallucination with excellent formatting.
Stage 3	**Mission-Aware Authorization Intelligence**	Bi-directional AEG with mission dependency mapping; compliance digital twin; temporal decay monitoring; ORDI dashboard; CAIRS operational; reserve margin tracking; global interoperability standards alignment	The reward for Stage 1+2 working correctly. A digital twin maintained by ungated agents is a strategic liability.

The compliance digital twin — continuously updated, queryable, with diffable snapshots and bi-directional mission dependency mapping — is the right long-term architecture target. It is also the reward for getting Stages 1 and 2 right. Organizations that attempt to build Stage 3 before Stage 1 and 2 are in place will produce an authoritative-looking compliance record whose integrity is exactly as reliable as the uncertified agents that maintain it.

16. A Confidential AI Incident Reporting System for DoD

NASA's Aviation Safety Reporting System has processed over two million reports and issued more than 6,200 safety alerts since 1976. Its success rests on three structural principles: confidential (reports cannot be used for enforcement), non-punitive (good-faith reporters receive FAA immunity), and neutral third party (NASA operates it, not the FAA). The DoD has no analogous system for compliance agent failures.

A Confidential AI Incident Reporting System (CAIRS) for DoD compliance agents would capture: near-misses (mis-mapped controls caught during review, incorrect inheritance assumptions, ODD boundary violations), drift incidents (agent behavior changes after model or corpus update), adversarial incidents (Type 1/2/3 injection attempts, evidence spoofing attempts), and Inheritance Integrity Failures discovered during manual audit.

The CAIRS requires specific legal architecture beyond the ASRS policy model: unlike aviation near-misses, a compliance failure report describing a specific incorrect control assertion may also constitute a disclosure of a security vulnerability in a specific system. The Defense Health Agency's patient safety reporting system — which adapted ASRS principles within a classified environment — is a closer operational model than ASRS itself, because it was designed with classification considerations analogous to the DoD compliance context.

17. Recommendations: The Five Non-Negotiables

Immediate Actions — 0 to 90 Days

1. Require model version pinning for all AAL-2+ compliance agents on GenAI.mil. Establish ACPM as a mandatory deployment artifact. Include model version, prompt hash, corpus hash, ODD definition, and legal delegator designation as required fields.
2. Designate NIST OSCAL Rev 5 as the authoritative ground-truth reference for AI-generated NIST compliance assertions. Require automated comparison against OSCAL data in any agent validation package.
3. Require SPSB alongside ACPM for any compliance agent generating OSCAL assessment results. Establish a signed provider registry for Cloud One OSCAL component definitions — the foundational defense against Type 3 inheritance injection.
4. Issue CDAO policy clarifying that compliance agents producing SSP content, control narratives, or ATO-relevant documentation are classified AAL-2 or higher with corresponding validation requirements.
5. Audit the inheritance implications of all currently deployed compliance agents. For any agent whose outputs have been incorporated into common control provider documentation, conduct VCAR-equivalent ground-truth review of cited controls.

Near-Term Actions — 90 Days to 12 Months

6. Establish mandatory provenance logging (evidence ledger) for all DoD compliance agents, with tamper-evident storage consistent with SP 800-53 AU-9/AU-10. Minimum log fields as specified in Section 14.
7. Integrate multi-run reliability testing (minimum five runs per condition, minimum ten for AAL-3) into GenAI.mil's Agent Designer workflow as a required validation step before agent sharing.
8. Publish the ORDI formula, reserve margin requirements, and temporal decay framework as reference tools for program managers. Develop a standard reporting template for agent reliability portfolio assessment.
9. Launch a pilot CAIRS within a single command, using the Defense Health Agency patient safety model as the legal architecture template. Designate a neutral analysis function outside CDAO.
10. Incorporate METS randomized spot-check protocols into ISSM training and SSP review workflows. Define sampling rates by AAL tier.

Strategic Actions — 12 to 36 Months

11. Develop and publish a crosswalk between NIST AI RMF (AI 100-1) and ISO 21448 SOTIF specifically addressing agentic AI in compliance workflows. Pursue as a joint NIST CAISI / CDAO effort.

12. Deploy the bi-directional Authorization Evidence Graph as Stage 3 infrastructure, after OSCAL-native output and provenance logging are operational. Define reverse-direction alerting thresholds for mission impact notification.
13. Incorporate model version and SPSB change events into DoD RMF change management workflows under SP 800-53 CM family controls. Foundation model updates should trigger the same configuration management review as any other change to security-relevant system components.
14. Publish the full reliability doctrine stack — ACPM, SPSB, VCAR, AAL framework, ORDI, AEG, METS — as open standards available to Five Eyes partners and NATO allies. This is the interoperability action that converts a DoD-internal framework into a global standard.

Conclusion: Building the First Airworthiness Framework for Machine-Governed Trust

In 1958, the Federal Aviation Act gave the FAA authority to define what 'airworthiness' meant for the jet age. The agency's definitions — precursors to DO-178, to FAR Part 25, to airworthiness certification — became the global standard not because they were perfect but because they were first, rigorous, and backed by institutional authority. Every allied aviation authority eventually adopted or mirrored them. Commercial aviation's safety record — which went from catastrophic loss rates in the 1950s to the safest mode of mass transportation in history — is the downstream consequence of that standard-setting work.

The Department of Defense (DoD) is in an analogous position today. It has the scale: three million users, the largest enterprise AI deployment in any government. It has the urgency: the Agent Designer was released in March 2026 and teams are already building compliance agents. It has the institutional authority to mandate rather than suggest. And it has 18 months, at most, before agentic compliance infrastructure becomes embedded in every allied government's workflow — at which point the standards will be set by whatever the market produced rather than whatever the DoD defined.

What is being defined here is not a workflow. It is an epistemic standard for what it means to know that a system is compliant. The shift this paper proposes is from compliance as process to compliance as proof. Not 'a human reviewed this' but 'this assertion is machine-verifiably derived from this evidence, under these conditions, with this confidence, and any authorized system can reconstruct the derivation independently.'

> *The reliability stack proposed here is not specific to DOD compliance. It applies to nuclear command and control, healthcare authorization, critical infrastructure cybersecurity, autonomous logistics, civilian government auditing, and safety-critical industrial AI. The organizations that define these reliability artifacts first will define the international standard for machine-governed trust. The DoD is in that position now.*

Princeton proves that reliability is multidimensional and fragile. This framework accepts that conclusion and builds the machinery needed to survive it. The Reliability Kill Chain is the intellectual bridge between empirical reliability research and operational defense doctrine. The ACPM, SPSB, VCAR, AEG, and reserve margin are the engineering implementations. Together they form the complete stack: measure reliability, constrain the system, preserve provenance, detect drift, prevent propagation, continuously re-validate trust.

The Air Force team that built the NIST Expert agents was right to build them. They demonstrated what practitioners can accomplish when given the tools. This framework does not constrain that ingenuity. It provides the architecture that allows it to be trusted — not because the outputs usually look good, but because every assertion is independently verifiable, every failure is caught before it propagates, and every authorization decision rests on machine-verifiable proof rather than machine-generated prose.

> *That is the real successor to 'audit-ready.' Not a document that looks defensible. A proof object that is.*

Appendix: Standards, Research, and Policy References

NIST Standards

- NIST AI 100-1: Artificial Intelligence Risk Management Framework (January 2023)
- NIST AI 800-2 ipd: Practices for Automated Benchmark Evaluations of Language Models (January 2026, comment period closed March 31, 2026)
- NIST AI 800-3: Statistical Models for Benchmark Evaluation (February 2026)
- NIST AI 800-4: Challenges to the Monitoring of Deployed AI Systems (March 2026)
- NIST SP 800-53 Rev 5: Security and Privacy Controls (OSCAL machine-readable catalog at NIST CSRC)
- NIST OSCAL: Open Security Controls Assessment Language — SSP, SAP, Assessment Results, POA&M, Component Definition models (pages.nist.gov/OSCAL)
- NIST COSAiS: Control Overlays for Securing AI Systems (agent overlays in development)
- NIST AI Agent Standards Initiative (February 2026)
- NIST NCCoE: Accelerating Adoption of Software and AI Agent Identity and Authorization (concept paper, February 2026)
- NIST TN 1900: Evaluating and Expressing the Uncertainty of NIST Measurement Results
- NIST SP 800-92: Guide to Computer Security Log Management

DoD Policy Documents

- CDAO AI Test and Evaluation Strategy Frameworks (April 2024)
- DOD Manual 5000.101: Operational Test and Evaluation and Live Fire Test and Evaluation
- DoD CIO Continuous Authorization to Operate (cATO) Implementation Guide
- DoD CIO cATO Evaluation Criteria
- CDAO Responsible AI Toolkit v1.0 (November 2023)
- DoD AI Acceleration Strategy (January 2026)
- Defense AI Guide on Risk (DAGR)
- Platform One Big Bang Security Documentation — compliance gate architecture

Federal Compliance Infrastructure

- FedRAMP RFC-0024: Machine-Readable Authorization Data Proposal — specifically the contrast between 'machine-generated probabilistic text

designed to mimic human-written narratives' and deterministic machine-readable telemetry
- FedRAMP Notice 0009: Machine-Readable Authorization Data Scope Clarification
- Air Force Materiel Command: Cloud One — Inheritable RMF Controls for ATO (afmc.af.mil)
- DoD CDAO: Agent Designer and NIST Expert Agent social post (LinkedIn, March 2026)

Safety-Critical Engineering Standards

- RTCA DO-178C: Software Considerations in Airborne Systems and Equipment Certification
- SAE ARP4754B: Guidelines for Development of Civil Aircraft and Systems (December 2023)
- IEC 61508: Functional Safety of E/E/PE Safety-Related Systems — SIL framework
- ISO 21448:2022: Road Vehicles — Safety of the Intended Functionality (SOTIF) — ODD and unknown unsafe scenario methodology
- IEEE Std 603: Standard Criteria for Safety Systems for Nuclear Power Generating Stations
- ASME ANSI/ANS-58.21: Standard for Level 1 LERF Probabilistic Risk Assessment
- ISO/IEC TS 5723:2022: Trustworthiness — Vocabulary (reliability definition grounding)

Primary Research

- Rabanser, Kapoor, Kirgis, Liu, Utpala, Narayanan. 'Towards a Science of AI Agent Reliability.' Princeton University, arXiv:2602.16666 (February 2026)
- Princeton CITP/AI Lab. Comment on NIST AI 800-2 Initial Public Draft (March 31, 2026) — sage.cs.princeton.edu/documents/NIST_benchmark_reliability.pdf
- NASA ASRS: Aviation Safety Reporting System — program briefing, caveats, and confidentiality model (asrs.arc.nasa.gov)