RF-100 Alignment with OWASP Agentic AI Top 10 and NIST AI Frameworks

Derek Hone — July 29, 2026

Document version: 1.0
Assessment basis: RF-100 v1.0 Public Review Draft, July 2026; OWASP Top 10 for Agentic Applications 2026; NIST AI RMF 1.0; NIST AI 600-1; and NIST SP 800-53 Rev. 5.

Executive Summary

RF-100 is an open, implementation-neutral standard for governing consequential actions proposed by human, automated, or AI actors. The July 2026 RF-100 v1.0 Public Review Draft is not a finalized standard, certification program, or evidence that any implementation conforms. It defines a verification-first architecture in which a Governed Action cannot take effect until an administratively and operationally independent Verification Gate evaluates the actor's authority, required evidence, a pinned policy snapshot, and declared system-state and risk signals. RF-100 then requires the resulting ALLOW, HOLD, or DENY decision and its basis to be durably recorded in a signed Verification Decision Record (VDR). This creates a technical bridge from risk-management principles to a decision made before real-world effect, rather than relying exclusively on model behavior, cooperative guardrails, or retrospective logging.[1]

The OWASP Top 10 for Agentic Applications and NIST frameworks address complementary layers of this problem. OWASP identifies concrete agentic attack and failure modes, including goal hijacking, tool misuse, privilege abuse, poisoned components, unexpected code execution, memory poisoning, insecure inter-agent communication, cascading failures, human trust exploitation, and rogue-agent behavior.[2] [3] NIST AI RMF 1.0 organizes organizational risk activity through GOVERN, MAP, MEASURE, and MANAGE; NIST AI 600-1 identifies generative-AI risks including confabulation, human-AI configuration, information integrity and security, data privacy, and value-chain integration; and NIST SP 800-53 supplies established access-control, authentication, configuration, and input-validation controls.[4] [5] [6] [7] RF-100 does not replace those frameworks. Its distinctive contribution is a narrowly focused, testable pre-execution control layer for consequential actions, including independent authorization, execution binding, bounded reuse, failure posture, and reconstructable decision records.

RF-100 is the open standard, while ExecutionProof is the commercial implementation from Remnant Fieldworks Inc. ExecutionProof is pilot-ready, not fully production-ready, and does not claim full RF-100 conformance; the RF-100 materials expressly state that asymmetric signing conforming to RF-100.8.4 and other required controls are not yet operational in the platform.[8] [9] [1] Accordingly, this report treats ExecutionProof mechanisms as proposed deployment patterns implemented through the ExecutionProof enforcement interface, not as guarantees of completed functionality or certification. In this role, ExecutionProof can serve as an implementation bridge for Verification Before Execution™: intercept a proposed consequential action, evaluate it independently, return ALLOW/HOLD/DENY, bind an authorization to the verified action, and preserve a ProofRecord as the branded implementation concept for a VDR. This framing makes no unsupported claim about performance, coverage, NIST or OWASP endorsement, or full conformity.

Methodology, Interpretive Status, and Limitations

This crosswalk compares control objectives and enforcement mechanisms rather than treating different frameworks as interchangeable. The analysis uses:

1. The actual normative requirement text of the July 2026 RF-100 v1.0 Public Review Draft, with requirement identifiers and RFC-style MUST, MUST NOT, SHOULD, and MAY language.[1]
2. The canonical OWASP Top 10 for Agentic Applications taxonomy and its official risk scope.[2] [10] [3]
3. NIST AI RMF 1.0, NIST AI 600-1, and the specified NIST SP 800-53 Rev. 5 controls.[4] [5] [6] [7] [11] [12]
4. ExecutionProof patterns as prospective implementation guidance through the ExecutionProof enforcement interface, clearly separated from RF-100's normative text.

A mapping indicates a meaningful relationship, not equivalence. NIST itself notes that crosswalk relationships are not always one-to-one and can be subjective.[7] RF-100 is narrower than an enterprise AI governance program, broader than an agent-only security filter in its coverage of human and automated actors, and more prescriptive than the referenced risk frameworks about the point at which consequential execution must be stopped. Conversely, RF-100 does not supply the full organizational, social, privacy, model-evaluation, software-development, or supply-chain program required to address every OWASP and NIST concern.

RF-100 is a July 2026 Public Review Draft. It is not final, frozen, externally approved, or certified. RF-101 assessment procedures and RF-102 certification are under development, and no RF-100 certification currently exists.[8] [9] This report is an interpretive technical mapping, not legal advice, an audit opinion, an attestation, or a conformance determination. Neither OWASP nor NIST is represented as endorsing RF-100, Remnant Fieldworks Inc., ExecutionProof, the ExecutionProof enforcement interface, or this crosswalk.

Section 1: OWASP Agentic AI Top 10 RF-100 Crosswalk

Version and Scope Note

As of this report date, the canonical OWASP Top 10 for Agentic Applications released in December 2025 is labeled ASI01–ASI10 and designated for 2026. The supplied evidence gives December 9 and December 10, 2025 release references; the first-party OWASP evidence supports December 9, 2025, while associated repository evidence cites December 10. The material conclusion is that it is the December 2025 release designated "2026," not an OAA- or AA-labeled list.[2] [10] [13] [14] Any OAA/AA examples used in earlier materials can describe valid conceptual risks, but they are not the current canonical identifiers.

The canonical items and concise official-scope descriptions are:

1. **ASI01 — Agent Goal Hijack:** manipulation of an agent's objectives or decision pathways through deceptive inputs, prompt injection, tool output, or poisoned data.
2. **ASI02 — Tool Misuse & Exploitation:** use of legitimate tools in unintended, destructive, unsafe, or unauthorized ways.
3. **ASI03 — Identity & Privilege Abuse:** exploitation of excessive permissions, inherited or leaked credentials, or trust relationships so an agent operates beyond its intended scope.
4. **ASI04 — Agentic Supply Chain Vulnerabilities:** compromise or poisoning of agentic runtime components, tools, plugins, descriptors, MCP servers, packages, or third-party integrations.
5. **ASI05 — Unexpected Code Execution:** natural-language or agent-controlled execution paths that lead to unauthorized code, shell-command, or remote-code execution.
6. **ASI06 — Memory & Context Poisoning:** corruption of persistent memory, context, or retrieval stores to alter later agent behavior.
7. **ASI07 — Insecure Inter-Agent Communication:** spoofed, malicious, or insufficiently protected messages that misdirect agents or multi-agent workflows.
8. **ASI08 — Cascading Failures:** propagation and amplification of errors or false signals through interconnected agents and automated pipelines.
9. **ASI09 — Human-Agent Trust Exploitation:** confident or misleading agent output that induces human operators to approve harmful actions.
10. **ASI10 — Rogue Agents:** compromised, misaligned, concealed, or self-directed behavior that departs from intended purpose or authority.[2] [10] [3]

Crosswalk Table

OWASP item and risk	RF-100 requirements/sections	Rationale and coverage boundary	ExecutionProof enforcement point
ASI01 — Agent Goal Hijack	3.1–3.4; 4.1–4.4; 5.1–5.4; 7.1–7.7; AI.2–AI.4a; AI.7; AI.10	External policy, authenticated evidence, scoped capability, isolated injection signals, and deterministic checks can prevent a manipulated objective from authorizing an out-of-scope effect. RF-100 does not claim to eliminate goal hijacking inside the model; it constrains the resulting Governed Action.	the ExecutionProof enforcement interface intercepts the proposed tool action, validates scope and deterministic parameters, evaluates out-of-band injection signals, and returns HOLD or DENY where requirements are unmet.
ASI02 — Tool Misuse & Exploitation	3.1; 5.2; 6.1; 7.1, 7.10; 8.1; 9.1–9.3; 11.1–11.2; AI.2, AI.3, AI.6, AI.10	Tool invocation is governed by explicit authority and action scope. At E2+, the verified parameter digest must match execution. This controls consequential tool calls but does not secure every internal vulnerability of the tool or Effector.	API gateway, tool-call proxy, MCP gateway, or service-mesh interposition validates a proposed call and releases it only after ALLOW.
ASI03 — Identity & Privilege Abuse	6.1–6.3; 8.1; 8.8; 12.1; AI.1–AI.3; AI.8–AI.9; AI.12	Explicit identity-bound authority replaces implicit or ambient authority. Agent identity includes model/version, lineage, and deployment context. RF-100 governs action authorization; it does not replace enterprise account lifecycle or credential-management systems.	the ExecutionProof enforcement interface binds workload and agent identity to an EBP capability set and checks current authority or delegation before decision rendering.

OWASP item and risk	RF-100 requirements/sections	Rationale and coverage boundary	ExecutionProof enforcement point
ASI04 — Agentic Supply Chain Vulnerabilities	3.2–3.5; 4.4; 8.1, 8.4–8.8; 12.1; AI.1, AI.12; C.1	Content-addressed policy, source-authenticated evidence, gate version/configuration records, key governance, and agent re-versioning support integrity and traceability. RF-100 does not provide a complete supplier-assurance, SBOM, dependency-scanning, or secure-development program.	Deployment metadata and component attestations can be required as evidence; changed agent versions or configurations trigger re-evaluation before further governed proposals.
ASI05 — Unexpected Code Execution	3.1; 4.1–4.4; 6.1; 7.1–7.4, 7.10; 9.1–9.3; 11.1–11.2; AI.2, AI.7, AI.10	Shell, code, and privileged execution can be classified as Governed Actions and blocked unless independently authorized. RF-100 does not itself sandbox interpreters, remediate code vulnerabilities, or inspect all code semantics.	A privileged-execution broker holds credentials or network access and issues an action-bound execution capability only after ALLOW.
ASI06 — Memory & Context Poisoning	4.1–4.4; 5.2, 5.4; AI.4–AI.4a; AI.7; AI.12	Actor-provided context is not Evidence. Evidence must be authenticated, fresh, and integrity-protected. This limits poisoned context from elevating a decision, but RF-100 does not clean or restore memory stores.	the ExecutionProof enforcement interface separates untrusted context and Intent Records from authenticated evidence and allows consistency analysis only to increase risk toward HOLD or DENY.
ASI07 — Insecure Inter-Agent Communication	4.4; 6.1–6.2; 7.10; 8.1, 8.4; AI.1, AI.8, AI.8a, AI.9	Session attribution, authenticated authority, complete delegation chains, signed records, and independent per-hop verification mitigate spoofing and unauthorized delegation. They do not substitute for transport security or secure protocol implementation.	Every consequential agent-to-agent hop is treated as a new enforcement interface proposal in ExecutionProof with independent identity, authority, evidence, policy, and parameter checks.

OWASP item and risk	RF-100 requirements/sections	Rationale and coverage boundary	ExecutionProof enforcement point
ASI08 — Cascading Failures	3.1; 5.2; 7.2–7.9; 9.3; 10.1–10.3; AI.5–AI.6a; AI.8a; D.2	Consequence tiers, cumulative and velocity limits, strongly consistent quota state, bounded reuse, fail-closed behavior, and per-hop governance limit propagation. RF-100 cannot prevent all failures outside declared boundaries.	Shared gate state enforces cumulative and velocity limits; circuit-breaking policy returns HOLD or DENY and prevents one ALLOW from authorizing later hops.
ASI09 — Human-Agent Trust Exploitation	5.4; 6.2; 7.3, 7.7; 8.1; AI.4a; AI.11	Agent explanations and Intent Records cannot be Evidence or elevate an action toward ALLOW. HOLD approval triggers re-verification rather than direct execution. RF-100 does not eliminate automation bias or guarantee approver judgment.	Escalation UI separates agent assertions from verified evidence, displays authority and policy basis, and requires independently authenticated approval followed by re-evaluation.
ASI10 — Rogue Agents	5.1–5.4; 6.1; 7.1–7.2; 9.1–9.3; 11.1–11.2; AI.1–AI.3; AI.6–AI.12; Annex C	The actor cannot alter the gate, evidence in transit, policy, or decision. Scoped capability, external authority, E2+ interposition, velocity controls, and identity re-versioning constrain effects. RF-100 contains rather than proves the absence of rogue behavior.	ExecutionProof deployment places the ExecutionProof enforcement interface outside the agent's administrative boundary and withholds credentials, tokens, or network access until ALLOW.

ASI01 — Agent Goal Hijack

OWASP's concern is that adversarial instructions or poisoned data redirect the agent's objective.[2] [3] RF-100 addresses the execution consequence rather than claiming to make the model immune. RF-100.5.1 requires verification by a gate "administratively and operationally independent of the proposing Actor," and prohibits the actor from modifying gate policy, configuration, evidence in transit, or the decision. RF-100.5.2 requires verified authority, required evidence, a pinned policy snapshot, and evaluated state and risk signals for every decision.[1]

The AI profile adds two important boundaries. RF-100:AI.2 limits proposals to the EBP-declared capability set. RF-100:AI.4a states that Intent Records are actor-asserted context, "MUST NOT be treated as Evidence," and cannot elevate a decision toward ALLOW. RF-100:AI.7 requires prompt- or input-injection indicators to be treated as risk signals for above-threshold actions; detection must operate outside the actor's observable or influenceable context, and a flagged session cannot receive ALLOW for such actions without elevated review.[1]

ExecutionProof's implementation pattern is therefore an external ExecutionProof enforcement interface decision point. Goal interpretation can inform risk, but ALLOW should depend on deterministic scope, schema, authority, threshold, and authenticated-evidence checks. This is partial mitigation: malicious internal reasoning may persist, but its consequential output is denied, held, or constrained at the boundary.

ASI02 — Tool Misuse & Exploitation

Tool misuse occurs when a legitimate capability is used in an unintended or unauthorized manner.[2] [3] RF-100.6.1 requires explicit identity binding for authority to propose or approve each action class and rejects implicit or ambient authority as support for ALLOW. RF-100.8.1 binds the decision to the canonical action identifier and a digest of the complete canonicalized parameters. RF-100.9.2a then requires E2+ Enforcement Points to compare the execution-time parameter digest with the VDR and block mismatches.[1]

Sections 7 and 11 establish the practical control boundary. A valid decision must precede effect, and at E2 or higher no invocation may reach the Effector without a rendered decision. The Enforcement Point must remain outside the proposing actor's administrative control.[1] ExecutionProof can implement this through the ExecutionProof enforcement interface at an API gateway, MCP gateway, service mesh, or tool-call proxy. This controls whether and how a tool is invoked; separate controls remain necessary for vulnerabilities inside the tool, excessive functions not declared as Governed Actions, and downstream behavior outside the EBP.

ASI03 — Identity & Privilege Abuse

OWASP ASI03 concerns excessive permissions, credential inheritance, and operation outside intended scope.[2] [3] RF-100 complements conventional account management by making authority an explicit input to each consequential decision. RF-100.6.1 disallows implicit or ambient authority, while RF-100:AI.9 states that agents must not hold, issue, or approve their own authority. RF-100:AI.1 requires model/version, base-model lineage, and deployment context in each VDR, and AI.12 treats changes to model, lineage, weights, fine-tuning, or system prompt as a new identity version requiring scope and limit re-evaluation.[1]

The ExecutionProof enforcement interface implementation should bind the enterprise workload identity, agent-version identity, delegating principal, session, approved capabilities, and action class. ExecutionProof should query an authoritative identity and entitlement source rather than trust identity claims supplied in the prompt. RF-100 does not define the full account lifecycle, credential issuance system, or authentication protocol; those remain enterprise dependencies.

ASI04 — Agentic Supply Chain Vulnerabilities

ASI04 covers compromised or poisoned tools, MCP servers, plugins, packages, descriptors, and runtime components.[2] [3] RF-100 provides integrity and change-control hooks rather than a complete supply-chain program. RF-100.3.2 requires versioned, content-addressable policy. RF-100.3.4 pins each decision to one immutable policy snapshot. RF-100.4.4 requires evidence source authentication and cryptographic digests. RF-100.8.1 records gate software version and configuration hash, while RF-100.12.1 makes gate deployment, configuration, signing-key, EBP, and approver-set changes Governed Actions.[1]

ExecutionProof can require component identity, version, signature, approved-registry status, or assessment result as evidence before allowing a sensitive tool invocation. Agent changes should trigger the AI.12 re-evaluation workflow. This mapping does not amount to dependency scanning, secure build assurance, vulnerability remediation, or supplier governance; those controls must feed trusted evidence into the gate.

ASI05 — Unexpected Code Execution

OWASP ASI05 addresses unauthorized shell, code, or remote-code execution through agent-controlled paths.[2] [3] RF-100 can designate code execution, command execution, deployment, or privileged process creation as Governed Actions. RF-100.7.1 requires a decision before effect, while 7.10 requires the executing component or Enforcement Point to verify the gate signature, parameter digest, single-use status, and validity window before producing an effect.[1]

For the highest-consequence paths, E3 is the strongest RF-100 architecture: execution credentials, capability tokens, or network reachability are issued only after ALLOW. The ExecutionProof enforcement interface broker should canonicalize the target, command or artifact, arguments, environment, identity, and destination into the action digest. RF-100 does not itself sandbox code or establish that generated code is safe; it controls whether a specific verified execution may proceed.

ASI06 — Memory & Context Poisoning

Memory poisoning seeks to reshape later behavior by corrupting long-lived context or retrieval stores.[2] [3] RF-100's principal safeguard is evidentiary separation. RF-100.5.4 permits only Section 4 artifacts to constitute Evidence. RF-100.4.2 rejects stale evidence for ALLOW, and 4.4 requires source authentication and integrity protection from source to gate.[1]

ExecutionProof should treat model memory, retrieved text, conversation history, and Intent Records as untrusted context unless an artifact independently satisfies RF-100 evidence requirements. A model-based consistency evaluator may contribute only toward HOLD or DENY and must be isolated from the proposing actor's input context under AI.4a.[1] This prevents poisoned context from becoming a positive authorization fact, but does not detect, remove, or recover all poisoned memory.

ASI07 — Insecure Inter-Agent Communication

ASI07 addresses spoofed or malicious messages among agents.[2] [3] RF-100:AI.8 makes every proposal attributable to a session identity, while AI.8a requires each consequential hop in a chained or multi-agent workflow to undergo full independent verification. The complete delegation chain must be preserved, and an ALLOW for one hop cannot authorize a subsequent hop.[1]

ExecutionProof should therefore avoid a single workflow-wide authorization. the ExecutionProof enforcement interface should authenticate each proposing agent, verify its delegated authority and evidence, and create a distinct ProofRecord for every consequential hop. Secure transport and message integrity remain necessary independently of RF-100; a signed VDR does not repair an insecure inter-agent protocol.

ASI08 — Cascading Failures

OWASP ASI08 concerns errors or false signals propagating and amplifying through automated pipelines.[2] [3] RF-100:AI.6 requires cumulative and velocity limits for agent-proposed action classes. AI.6a requires strongly consistent shared state across gate instances or statically partitioned quotas whose sum does not exceed the global limit. AI.8a prevents one decision from authorizing an entire chain.[1]

RF-100 also requires fail-closed behavior for gate outage, network partition, evidence-service failure, authority-service failure, and clock fault. Absence of a valid decision means no execution.[1] ExecutionProof should implement shared counters, bounded decision reuse, consequence-based thresholds, and per-hop checks. The boundary remains the EBP: undeclared actions or systems that bypass the Enforcement Point can still propagate failure.

ASI09 — Human-Agent Trust Exploitation

ASI09 concerns polished or confident agent statements that mislead human operators.[2] [3] RF-100 directly limits the authorization value of such statements. Intent Records are untrusted context and cannot elevate a decision. Under RF-100.7.7, approving a held action authorizes re-evaluation, not execution; authority, evidence freshness, system state, and risk must be checked again before ALLOW.[1]

An ExecutionProof escalation interface should clearly distinguish verified evidence from agent assertions, identify missing controls, display the applicable policy snapshot, and authenticate the human approver. Where M-of-N approval is required, approvals must be independently authenticated and recorded. RF-100 cannot guarantee that a human will exercise sound judgment, so training, interface design, workload management, and anti-automation-bias measures remain necessary.

ASI10 — Rogue Agents

OWASP ASI10 includes compromised, misaligned, concealed, or self-directed behavior.[2] [3] RF-100's answer is containment at the effect boundary. The proposer is expressly treated as the component that cannot be trusted to self-govern; verification and enforcement therefore reside outside its control.[1] Scoped capability, external authority, injection-risk handling, velocity limits, E2/E3 enforcement, independent approvals, and identity re-versioning constrain the agent's capacity to create effects.

This is not proof that an agent is aligned or free of rogue behavior. It is an architectural reduction in the authority of the agent's output. ExecutionProof should be deployed so the agent cannot directly access production credentials, effectors, gate administration, or decision records capable of authorizing execution.

Section 2: NIST AI RMF / SP 800-53 / AI 600-1 RF-100 Crosswalk

Upfront Crosswalk

NIST framework item	RF-100 requirements/sections	Mapping boundary	ExecutionProof enforcement point
AI RMF GOVERN	3.1–3.5; 8.2, 8.7; 10.1–10.3; 11.1; 12.1; C.1; D.1–D.3	EBP ownership, policy versioning, records, enforcement classification, administration governance, adversary scope, and assessment evidence operationalize selected governance decisions. RF-100 is not a complete organizational governance system.	Policy owners approve versioned EBP and policy artifacts; the ExecutionProof enforcement interface enforces the active immutable snapshot and records administrative changes.
AI RMF MAP	1; 2; 3.1; 4.1–4.2; 7.6, 7.8; 11.1; AI.2, AI.5; C.1	Action inventory, consequence tiers, evidence, authority, validity, availability, capability, and adversary scope establish execution context. Broader social and lifecycle context remains outside RF-100.	Deployment discovery classifies actors, tools, effectors, action classes, consequences, and bypass paths.
AI RMF MEASURE	3.3, 3.5; 5.2; 8.1–8.6; 9.2; 10.1–10.3; 13.1; D.2	Simulation, risk signals, VDRs, reconciliation, reconstruction, trusted time, and fault injection produce control evidence. RF-100 does not prescribe full model-quality or fairness metrics.	Shadow evaluation, decision telemetry, fault tests, replay tests, digest comparisons, and ProofRecord verification.

NIST framework item	RF-100 requirements/sections	Mapping boundary	ExecutionProof enforcement point
AI RMF MANAGE	5.3; 7.1–7.9; 9.1–9.3; 11.1–11.2; 12.1; AI.6–AI.11	ALLOW/HOLD/DENY, failure posture, limits, escalation, interposition, and governance changes are direct risk-treatment mechanisms.	the ExecutionProof enforcement interface blocks, escalates, or releases actions according to policy and consequence tier.
AI 600-1: Confabulation	4.1–4.4; 5.2, 5.4; AI.4–AI.4a	Agent assertions cannot become evidence merely because they are confidently stated. Does not eliminate false model outputs.	Require authenticated external evidence for consequential factual premises.
AI 600-1: Human-AI Configuration	6.2; 7.3, 7.7; AI.4a, AI.11	Separates agent statements from evidence and structures human escalation. Does not address all over-reliance or anthropomorphism risks.	Evidence-focused review with authenticated approvals and mandatory re-verification.
AI 600-1: Information Integrity	4.4; 8.1–8.6; 9.2a; 10.3	Source authentication, content digests, signatures, and execution binding support integrity.	Verify evidence and action digests; preserve independently verifiable ProofRecords.
AI 600-1: Information Security	5.1; 7.2, 7.4, 7.10; 8.4, 8.8; 9.3; 11–13; AI.7	External gate, fail-closed behavior, anti-replay, key isolation, interposition, and injection signals address selected security risks.	Interposed gateway or credential broker outside the agent boundary.
AI 600-1: Data Privacy	8.2, 8.7; 10.2	Access control, confidentiality, retention, field minimization, and possible crypto-shredding apply to VDR data. RF-100 is not a complete privacy framework.	Minimized ProofRecord schema, protected fields, role-based audit access, retention and deletion workflows.
AI 600-1: Value Chain and Component Integration	3.2–3.5; 4.4; 8.1; 12.1; AI.1, AI.12	Versioning, provenance, authenticated evidence, and change re-evaluation provide integration controls, not comprehensive supplier assurance.	Require component attestations and block unapproved version or configuration changes.

NIST framework item	RF-100 requirements/sections	Mapping boundary	ExecutionProof enforcement point
SP 800-53 AC-2 — Account Management	6.1–6.3; 8.1; AI.1, AI.8, AI.9, AI.12	RF-100 consumes identity and authority state for action decisions but does not replace account provisioning and lifecycle management.	Identity adapter checks active status, revocation, agent version, and delegation.
AC-3 — Access Enforcement	5.2; 6.1; 7.1, 7.10; 11.1–11.2	RF-100 provides action-level access enforcement at the execution boundary.	E2 gateway or E3 credential broker blocks unauthorized invocations.
AC-4 — Information Flow Enforcement	3.1; 4.4; 8.7; 9.1–9.2a; 11.1–11.2	Governs declared data operations and flow-producing actions; does not replace network-wide flow controls.	Validate source, destination, data class, permitted fields, and action digest before transfer.
AC-6 — Least Privilege	6.1; AI.2, AI.5, AI.9–AI.10	Explicit authority, capability scope, thresholds, and external enforcement support least privilege and "least agency."	Action-specific, time-bound, single-use authorization or credentials.
IA-2 — Identification and Authentication (Organizational Users)	6.1–6.2; 8.1; 7.10	RF-100 requires authenticated identities and approvals but does not prescribe the organization's complete user authentication solution.	Integrate enterprise identity and strong approval authentication.
IA-10 — Adaptive Identification and Authentication	5.2; 7.7; AI.5–AI.7; AI.11	Risk signals, consequence tiers, injection flags, and elevated review can drive adaptive requirements.	Raise authentication or approval requirements based on action risk and state.
CM-7 — Least Functionality	3.1; 11.1; AI.2, AI.10	EBP capability sets and enforcement classes constrain exposed functions. RF-100 does not configure operating-system services or ports.	Publish only approved tool functions through the ExecutionProof enforcement interface and deny undeclared actions.
SI-10 — Information Input Validation	4.2–4.4; 5.2, 5.4; 7.4; AI.4a, AI.7	Freshness, integrity, authentication, schema checks, malformed-decision rejection, and injection signals support validation.	Canonicalize and validate parameters before policy evaluation and again at execution.

NIST AI RMF 1.0

NIST AI RMF 1.0 is voluntary, non-sector-specific guidance organized around GOVERN, MAP, MEASURE, and MANAGE. GOVERN is cross-cutting; MAP establishes context; MEASURE assesses and monitors risks using qualitative, quantitative, or mixed methods; and MANAGE prioritizes and treats identified risks.[4] [15] [5] RF-100 should be used as one implementation layer within that lifecycle, not represented as a substitute.

GOVERN RF-100 turns selected governance choices into versioned, inspectable configuration. RF-100.3.1 requires the EBP to enumerate Governed Actions, consequence tiers and threshold, evidence and freshness requirements, authority model, enforcement and availability classes, validity windows, and velocity consistency model where applicable. RF-100.3.2–3.4 require content-addressed policy and immutable decision snapshots. Section 12 brings changes to the governance mechanism itself inside the governed boundary.[1]

ExecutionProof should assign named owners for the EBP, policy, evidence sources, identity integration, escalation, retention, cryptographic keys, and exception handling. the ExecutionProof enforcement interface should enforce approved versions rather than selecting policy dynamically from labels. The broader GOVERN function—including organizational culture, workforce responsibilities, and third-party governance—requires controls beyond RF-100.[5]

MAP MAP establishes the intended use, context, benefits, risks, and affected environment needed for informed decisions.[5] RF-100's execution-focused analogue is boundary declaration. The organization identifies actors, effectors, Governed Actions, consequences, evidence, authority, validity windows, availability needs, capability sets, and adversary classes.

ExecutionProof deployment should begin with transaction and tool-flow mapping, including direct and indirect paths to effectors. A control cannot be an effective pre-execution control if an agent retains an alternate credential, network path, administrative interface, or batch channel that bypasses the ExecutionProof enforcement interface.

MEASURE MEASURE applies assessment and monitoring methods based on the mapped context.[5] RF-100.3.3 requires policy simulation without execution effects. RF-100.3.5 recommends shadow evaluation before enforcement. Sections 8 and 10 establish decision records and reconstruction, while D.2 requires fault injection across gate outages, network partitions, evidence and authority failures, and clock faults.[1]

ExecutionProof's assurance pattern should collect shadow-decision differences, policy-rule coverage, evidence freshness failures, HOLD and DENY reasons, expired or replayed decisions, execution-digest mismatches, and bypass-test results. These are control-effectiveness measures, not measures of overall model accuracy, fairness, toxicity, or social impact.

MANAGE MANAGE applies treatment strategies based on mapped and measured risks.[5] RF-100 supplies a direct treatment mechanism through ALLOW/HOLD/DENY, fail-closed behavior, escalation, expiry, bounded reuse, interposed enforcement, velocity limits, and per-hop verification.

ExecutionProof should connect the decision to an actual effect-control mechanism. A dashboard warning without interposition is not equivalent to E2 or E3 enforcement. Low-consequence E1 use remains cooperative and requires post-hoc reconciliation; above-threshold AI actions require E2 or higher under RF-100:AI.10.[1]

NIST AI 600-1 Generative AI Risk Dimensions

NIST AI 600-1, published in final form in July 2024, is a cross-sectoral companion to AI RMF 1.0. It identifies 12 generative-AI risks, including the dimensions most material to agent authorization and pre-execution governance assessed below.[6] [16]

Confabulation NIST describes confabulation as false or erroneous content presented with high confidence.[6] RF-100 does not try to prove that every model statement is factual. Instead, it prevents actor

assertions from serving as authorization evidence. Required evidence must be declared by action class, fresh, source-authenticated, and integrity-protected; missing evidence produces HOLD or DENY, never ALLOW.[1]

ExecutionProof should obtain facts such as balances, transaction status, target state, approved change records, or compliance prerequisites from authenticated systems of record. Model-generated summaries may be shown as context but should not satisfy evidence rules.

Human-AI Configuration NIST's human-AI configuration risk includes over-reliance, automation bias, and inappropriate anthropomorphism.[6] RF-100:AI.4a limits reliance on agent explanations, and 7.7 makes human approval of a HOLD an authorization to re-evaluate rather than execute. AI.11 introduces independence conditions for above-threshold AI approvers.[1]

ExecutionProof should present the missing or failed control, evidence provenance, consequence tier, and exact requested action separately from the agent narrative. Human reviewers must retain a genuine denial path and should not receive an interface that treats approval as the default.

Information Integrity NIST identifies increased production and dissemination of unvetted content as an information-integrity risk.[6] RF-100 addresses integrity of decision inputs and execution records through authenticated evidence, content digests, immutable policy snapshots, asymmetric signatures, action binding, and offline-verifiable VDRs.[1]

This supports integrity for governed decisions, not the truth of all generated content. If publication or communication is consequential, the publication itself can be declared a Governed Action with evidence and approval requirements.

Information Security NIST AI 600-1 identifies expanded cyberattack capability and attack surface as information-security risks.[6] RF-100 applies an independent gate, fail-closed behavior, validity and anti-replay checks, key isolation, external Enforcement Points, injection risk signals, and governed administration.

ExecutionProof should be positioned at credential, network, API, or infrastructure choke points. The product pattern complements, rather than replaces, secure software development, vulnerability management, endpoint controls, segmentation, and incident response.

Data Privacy NIST identifies unauthorized disclosure and de-anonymization of sensitive data as data-privacy risks.[6] RF-100.8.7 requires VDR access controls and confidentiality protections commensurate with sensitivity, field-level minimization, and—where legally required—support for cryptographic erasure without invalidating remaining chain integrity. Audit access cannot permit VDR modification under 10.2.[1]

A ProofRecord should therefore contain what is necessary to reconstruct the decision, not complete prompts, datasets, or unnecessary personal data. Privacy impact analysis, lawful-basis determination, individual-rights handling, and broader data-governance obligations remain outside this crosswalk.

Value Chain and Component Integration NIST highlights limited transparency and insufficient vetting of third-party components and data.[6] RF-100 contributes immutable version references, authenticated evidence sources, recorded gate versions and configuration hashes, and mandatory re-evaluation when agent identity-relevant components change.

ExecutionProof can make an approved component version, provenance attestation, or assessment result a prerequisite for ALLOW. Supplier due diligence and component security assessment must be performed by other functions and supplied to the gate as authenticated evidence.

NIST SP 800-53 Rev. 5

NIST SP 800-53 Rev. 5 is a flexible catalog of security and privacy controls addressing both functionality and assurance. NIST issued a minor Release 5.2.0 in August 2025, but the supplied update list does not identify changes to the eight controls mapped here.[7] The following mappings are implementation interpretations, not NIST mappings or endorsements.

AC-2 — Account Management AC-2 covers creation, modification, disabling, and oversight of system accounts.[11] RF-100 relies on identity and authority services but adds transaction-time checks: authority must be explicit, revocation must affect unrendered decisions, and actor identity and delegation must be recorded.

ExecutionProof should query authoritative lifecycle status at decision time and reject disabled, expired, unrecognized, or improperly versioned identities. the ExecutionProof enforcement interface is not an account-management system and should not become the authoritative account registry unless separately designed and assessed for that purpose.

AC-3 — Access Enforcement AC-3 concerns enforcement of approved access-control policy.[11] RF-100 maps strongly at the action boundary: verified authority is mandatory, a valid decision must precede effect, and E2/E3 prevents invocation from reaching the Effector without a decision.

The ExecutionProof pattern is an interposed policy enforcement point. At E3, the strongest implementation is to withhold credentials, capability tokens, or network reachability until ALLOW rather than trusting the agent to honor the result.

AC-4 — Information Flow Enforcement AC-4 controls information flows within and between systems to prevent unauthorized transfers.[11] [17] RF-100 can govern data export, disclosure, replication, transmission, publication, or cross-boundary API operations as actions. Evidence and action digests provide integrity, while the EBP defines permitted classes and destinations.

ExecutionProof should canonicalize data classification, source, destination, operation, permitted fields, and volume. It does not replace network-level flow enforcement for traffic outside the declared execution path.

AC-6 — Least Privilege AC-6 requires minimum privileges necessary for assigned functions.[11] [17] RF-100.6.1 rejects ambient authority; AI.2 scopes agent capability; AI.9 externalizes authority; and AI.10 requires interposed enforcement for consequential agent actions.[1]

ExecutionProof should favor action-bound, short-validity, single-use permissions over standing credentials. This makes authority specific to the verified operation while preserving enterprise role and entitlement systems as the source of allowable delegation.

IA-2 — Identification and Authentication IA-2 addresses identification and authentication of organizational users, including relevant authentication mechanisms and account types.[11] [12] RF-100 requires authenticated actor authority and independently authenticated approvals, but does not prescribe a particular enterprise authentication protocol.

ExecutionProof should integrate with the organization's approved identity provider and preserve the authenticated identity, approval, delegation, and agent metadata in the ProofRecord. Authentication at login alone is insufficient if the action is later modified or delegated.

IA-10 — Adaptive Identification and Authentication IA-10 concerns adjustment of authentication based on risk, state, or behavior.[12] RF-100 supports adaptation through consequence tiers, system-state and risk signals, injection flags, M-of-N approval, elevated review, and post-HOLD re-verification.

the ExecutionProof enforcement interface policy can require stronger authentication, additional approvers, or a fresh human confirmation when risk crosses a declared threshold. The factors and authentication methods themselves must come from the organization's approved authentication architecture.

CM-7 — Least Functionality CM-7 requires systems to expose only essential functions, ports, protocols, and services.[12] RF-100's closest analogue is the EBP-declared action set and AI capability set. An agent may propose only actions in its declared set, and high-consequence actions must use E2 or E3 enforcement.

ExecutionProof should publish only approved tools and operations through the ExecutionProof enforcement interface. Undeclared functions should be unreachable, not merely absent from the prompt. Operating-system and network least-functionality controls remain separate.

SI-10 — Information Input Validation SI-10 requires validation of information inputs to prevent injection, malformed-data processing, and related vulnerabilities.[7] RF-100 requires evidence freshness, authentication, integrity, decision validation, deterministic parameter-schema checks, and external treatment of prompt/input-injection indicators.

ExecutionProof should canonicalize inputs before hashing, reject malformed or ambiguous parameters, validate required fields and thresholds, and repeat digest validation at execution. Semantic model-based checks can increase risk, but deterministic controls should gate ALLOW under AI.4a.[1]

Recent NIST Work on AI-Agent Identity and Authorization

NIST launched an AI Agent Standards Initiative in February 2026 focused on industry-led standards, community-led protocols, and research into agent authentication, identity infrastructure, and security evaluation.[18] [19] The NCCoE also published a February 2026 concept paper, *Accelerating the Adoption of Software and AI Agent Identity and Authorization*, addressing identification, authentication, authorization, delegation, least privilege, auditing, non-repudiation, and prompt-injection mitigation in agentic architectures.[20] [21]

These are recent initiatives and concept-stage project materials, not final NIST requirements establishing a universal pre-execution authorization architecture. NIST is considering or evaluating technologies such as MCP, OAuth 2.0/2.1, OpenID Connect, SPIFFE/SPIRE, and NGAC; that status must not be recast as a mandate for any one protocol.[20] NIST is also developing SP 800-53 control overlays for AI systems, including single- and multi-agent use cases.[22] [23] [24] This work is directionally aligned with identity, delegation, and authorization concerns addressed by RF-100, but it does not endorse RF-100 or ExecutionProof.

Section 3: Synthesis — Where RF-100 Leads

Fair Scope Comparison

OWASP leads as a threat taxonomy: it supplies a recognizable set of agentic risks and common language for attack and failure analysis. NIST AI RMF leads as an organization-wide, lifecycle risk framework. NIST AI 600-1 identifies broader generative-AI risk dimensions, and SP 800-53 provides a mature catalog of security and privacy controls. RF-100 has a different and narrower question: what must be verified, recorded, and enforced immediately before a consequential action takes effect?

RF-100 therefore "leads" only in the sense of prescriptive depth within this particular execution-governance layer. It should not be portrayed as superior across model evaluation, fairness, content safety, privacy governance, secure development, supply-chain assurance, organizational accountability, or societal risk.

Distinctive Pre-Execution Gate

RF-100.7.1 states: "A Decision MUST be rendered before any Governed Action takes effect." RF-100.7.2 requires fail-closed behavior under specified gate, network, evidence-service, authority-service, and clock failures: without a valid decision, execution does not occur.[1] This is the core pre-execution control.

The requirement is stronger than merely logging a decision, warning an operator, or asking an agent to self-check. At E2+, no Governed Action invocation may reach the Effector without a decision, and the Enforcement Point must be outside the actor's administrative control. E3 goes further by gating the capacity to execute through credentials, capability tokens, or network reachability.[1]

Independent Verifier and Proposer Separation

RF-100's central premise is that the proposing actor cannot be trusted to self-govern. RF-100.5.1 consequently requires an administratively and operationally independent Verification Gate and prevents the actor

from changing gate policy, configuration, evidence in transit, or the decision.[1]

This separation is significant for ASI01, ASI02, ASI03, and ASI10. A compromised or misaligned actor cannot grant itself authority merely by producing plausible reasoning. The residual risk is gate or operator compromise; Annex C expressly treats gate-operator compromise as only partially addressed and requires residual risk to be stated in the EBP.[1]

Evidence, Authority, Policy, and System-State Verification

RF-100.5.2 identifies four mandatory decision inputs:

- verified actor authority;
- required evidence;
- the pinned policy snapshot; and
- evaluated system state and risk signals declared in the EBP.[1]

This structure separates authorization facts from model confidence. Evidence must be action-class-specific, fresh, source-authenticated, integrity-protected, and digest-referenced. Missing evidence yields HOLD or DENY, never ALLOW. Policy is content-addressed and immutable for the duration of a decision.[1]

Bounded Decisions and Execution Integrity

An ALLOW is not a general permission. RF-100 binds it to the canonical action and complete parameter digest, gives it a validity window, and makes it single-use unless the EBP explicitly declares a bounded multi-use window. Before effect, the executor or Enforcement Point must verify signature, digest, use status, and validity.[1]

At E2+, parameter divergence blocks execution and generates an alert. At E1, RF-100 requires a declared post-hoc reconciliation procedure. The distinction matters: E1 is cooperative and limited to low-consequence use; E2 and E3 provide stronger preventive enforcement.

VDR and ProofRecord-Oriented Durable Records

Every ALLOW, HOLD, and DENY must produce a VDR. RF-100.8.1 requires the record to contain, at minimum, action and parameter identity, actor identity, decision and basis, policy hash, evidence references, authority and approval records, complete delegation chain, gate identity and configuration, timestamps, and validity window.[1]

VDRs must be durable, retained according to the EBP, and independently verifiable offline. RF-100 requires asymmetric signatures and prohibits symmetric MACs as the sole integrity mechanism for a conformance claim. Append-only external anchoring is a SHOULD rather than a MUST.[1] ExecutionProof uses the term ProofRecord for its branded implementation pattern; the generic normative RF-100 term remains VDR (Verification Decision Record).

Trusted Time

Evidence freshness, decision validity, revocation, replay resistance, and velocity limits depend on trustworthy time. RF-100.13.1 requires authenticated time synchronization and a documented maximum drift tied to the tightest relevant freshness limit or validity window. Independently verifiable timestamps are recommended.[1]

An implementation must therefore test clock fault, drift, rollback, and unavailable-time behavior. Trusted time is not merely a recordkeeping feature; it is part of whether ALLOW remains valid.

Enforcement Architecture

RF-100 defines three enforcement classes:

- **E1 — Cooperative:** the actor calls the gate and voluntarily honors the decision; permitted only for low-consequence actions.

- **E2 — Interposed:** a gate-controlled Enforcement Point sits in the execution path, so no invocation reaches the Effector without a decision.
- **E3 — Infrastructure-enforced:** credentials, capability tokens, or network reachability are released only after ALLOW.[1]

AI-profile actions at or above the consequence threshold require E2 or higher. This prevents an implementation from characterizing optional SDK checks as sufficient for high-consequence agent execution.

AI-Agent Scoped Authority and Intent Record Status

RF-100:AI.2 requires EBP-scoped agent capability, and AI.9 prevents an agent from holding, issuing, or approving its own authority. These controls distinguish the capacity to propose from the authority to execute.

Intent Records are optional supporting context under AI.4, not trusted evidence. AI.4a prevents them from elevating a decision toward ALLOW. A consistency evaluator may contribute only toward HOLD or DENY, must be isolated from the proposing actor's context, and does not replace deterministic checks.[1] This precision avoids treating an agent's account of its own purpose as proof of authorization.

Consequence and Velocity Limits

The EBP must define a consequence threshold that activates elevated AI controls. RF-100:AI.6 requires cumulative and velocity limits, while AI.6a requires strongly consistent shared state or statically partitioned quotas whose sum remains within the global limit.[1]

These requirements address not only a single harmful action but also repeated individually permissible actions that become harmful in aggregate. Implementation depends on robust distributed-state management; local counters that can each authorize the global maximum do not satisfy the stated consistency rule.

Input-Injection Defenses

RF-100:AI.7 requires prompt- or input-injection indicators to influence above-threshold decisions. Detection must be outside the actor's observable or influenceable context, flagged-session state must persist for the session, and above-threshold actions cannot receive ALLOW without elevated review.[1]

RF-100 does not claim perfect injection detection. Its stronger point is the consequence attached to a signal and the separation of the detector from the potentially manipulated context.

Per-Hop Governance

RF-100:AI.8a requires each consequential hop in a chained or multi-agent workflow to be treated as a separate Governed Action. Each hop receives full verification; each VDR preserves the complete delegation chain; and an ALLOW for one hop does not authorize the next.[1]

This prevents authority laundering through an intermediary agent and limits cascading failures. It also increases record volume, policy complexity, and latency sensitivity, which deployments must account for without assuming unsupported performance.

Precise Meaning of ALLOW/HOLD/DENY

RF-100.5.3 recognizes only ALLOW, HOLD, or DENY. No other result authorizes execution.[1]

- **ALLOW** means that verification succeeded for the specified action and parameters under the identified policy snapshot, evidence, authority, state, validity window, and reuse bound. It does not authorize modified parameters, expired execution, reuse outside the declared bound, or subsequent workflow hops.
- **HOLD** means the proposal must follow the EBP-declared escalation path. It does not authorize execution. Resolution triggers complete re-verification; human or M-of-N approval authorizes re-evaluation only.

- **DENY** means execution is not authorized. DENY must be recorded with the same completeness as ALLOW.
- A malformed, unauthenticated, unverifiable, expired, consumed, or otherwise invalid decision is treated as absent. Under fail-closed rules, absence means no execution.[1]

Specification Versus Future or Implementation Capability

RF-100 normatively specifies requirements, not a particular product architecture. ExecutionProof, ProofRecord, the ExecutionProof enforcement interface, hosted APIs, specific user interfaces, integrations, and deployment tooling are implementation choices. External transparency anchoring is recommended rather than universally required. Break-glass is optional and tightly bounded. RF-101 assessment procedures and RF-102 certification remain future work.[8] [9] [1]

ExecutionProof does not currently claim full RF-100 conformance. The RF-100 draft specifically records that an asymmetric-signing capability conforming to 8.4 and other controls must become operational before such a claim can be made.[9] [1] "Pilot-ready" means suitable for controlled evaluation and integration work; it does not establish production readiness, resilience, scalability, security, or conformity. Remnant Fieldworks Inc. states, as organizational framing, that it has 56 patent applications pending; this is not a representation that any patent has been granted or that patent status establishes control effectiveness.

Section 4: Implementation Guidance — How ExecutionProof Enforces Each Mapped Requirement in Production Agentic Systems

Deployment Blueprint Status

The following is a proposed deployment blueprint for pilot-ready ExecutionProof using the ExecutionProof enforcement interface. It describes how a production-oriented implementation could satisfy mapped control objectives. It is not a guarantee that every capability is complete, deployed, effective, or conformant.

1. Inventory and Boundaries

Create an inventory of:

- human, service, automated, and AI actors;
- agent model/version, base-model lineage, deployment context, and system prompt version;
- sessions and delegation relationships;
- tools, APIs, MCP servers, databases, infrastructure controllers, financial systems, and physical or communication effectors;
- direct and indirect execution paths;
- credentials, capability tokens, network routes, queues, batch processes, and administrative interfaces;
- action classes and canonical parameter schemas;
- consequence tiers and thresholds;
- required evidence, source, freshness, and integrity method;
- authority and M-of-N approval rules;
- validity, reuse, velocity, and cumulative limits;
- availability, break-glass, retention, privacy, and adversary requirements.

The EBP must identify which actions are governed. A pilot should deliberately search for bypasses: direct API credentials, inherited cloud roles, local shell access, alternate queues, emergency interfaces, and agent-accessible gate administration.

2. Identity and Authority Binding

Bind every proposal to:

- authenticated enterprise or workload identity;
- AI identity metadata required by RF-100:AI.1;

- session identity;
- originating human or organizational authority;
- complete delegation chain;
- EBP capability set;
- current authority status and revocation state;
- required approver set.

the ExecutionProof enforcement interface should obtain these facts from authoritative identity services or authenticated credentials. Agent-supplied identity text is not sufficient. Agent version changes under AI.12 should suspend governed execution until scope, consequence, capability, and limit assignments are re-evaluated.

3. Proposed Action and Intent Capture

Define a canonical proposal envelope containing:

- action identifier;
- complete canonicalized parameters;
- actor and session identity;
- delegation chain;
- requested execution time;
- declared target Effector;
- evidence references;
- optional Intent Record;
- correlation and workflow identifiers.

Canonicalization must be deterministic so the same action produces the same digest at verification and execution. Ambiguous, omitted, or malformed fields should not be silently inferred for consequential actions.

Intent Records should be stored and displayed as actor assertions. They may support audit and risk detection, but must not count as Evidence or increase the likelihood of ALLOW.

4. Independent Verification

Deploy the Verification Gate outside the proposing agent's administrative and operational control. the ExecutionProof enforcement interface should evaluate:

1. Is the actor identity authenticated and current?
2. Is the action in the actor's capability set?
3. Is explicit authority present and properly delegated?
4. Are all approvals independently authenticated?
5. Is each required evidence artifact present, fresh, authenticated, and integrity-protected?
6. Does the action comply with the exact immutable policy snapshot?
7. Do system-state conditions permit execution?
8. Are cumulative and velocity limits available and within bounds?
9. Are injection, anomaly, or other declared risk signals present?
10. Does the action exceed a consequence threshold requiring elevated review?
11. Are time, availability, and consistency conditions valid?
12. Is the gate itself using an approved version, configuration, and signing key?

A model-based evaluator should be supplemental and risk-increasing only where RF-100:AI.4a applies. Deterministic checks must control ALLOW.

5. ALLOW/HOLD/DENY Handling

ALLOW Create a decision bound to the canonical action digest, policy snapshot, evidence, authority, validity window, and reuse status. At E2 or E3, deliver it to an Enforcement Point outside the actor boundary. The actor should not receive standing authority merely because one proposal was allowed.

HOLD Route the proposal to the declared escalation path. Present verified evidence, missing requirements, risk signals, action parameters, authority chain, and policy basis separately from the agent narrative. Authenticate the reviewer and preserve the approval. Then re-run verification; do not translate approval directly into execution.

DENY Block the action and create a complete ProofRecord. Do not reduce logging merely because execution did not occur. Denial reasons should be structured for policy review without exposing sensitive detection logic to the proposing actor where that would weaken controls.

Invalid or Absent Decision Reject malformed, expired, consumed, unverifiable, unauthenticated, or digest-mismatched decisions. Gate, identity, evidence, network, and time failures should fail closed except for a correctly designed, pre-authorized RF-100 break-glass instrument.

6. ProofRecord and VDR Persistence

A ProofRecord intended to implement the VDR concept should contain all RF-100.8.1 minimum fields. The persistence design should provide:

- durable retention;
- asymmetric digital signatures conforming to RF-100.8.4 before any conformance claim;
- signing keys protected in an HSM or equivalent isolation;
- independent offline verification;
- role-based and confidentiality-aware access;
- field-level data minimization;
- immutable audit access;
- optional externally anchored append-only transparency;
- cryptographic agility and migration planning;
- legally appropriate erasure handling without invalidating remaining chain integrity.

Until these capabilities are implemented and assessed, a ProofRecord must not be represented as necessarily satisfying every VDR requirement.

7. Execution Binding

Immediately before effect, the Enforcement Point should verify:

- gate signature;
- action and parameter digest;
- validity window;
- single-use or bounded-use status;
- intended Effector and environment;
- current decision state;
- any action-bound credential or capability.

At E2, the Enforcement Point blocks invocation unless these checks pass. At E3, the actor lacks execution capacity until the infrastructure releases an action-specific capability. Consumption state must be protected against race conditions and replay.

8. Audit and Reconstruction

The audit path should answer, from the ProofRecord and referenced artifacts:

- Who proposed the action?
- Which agent version and session were involved?
- Under whose authority and delegation?
- What exact parameters were proposed and executed?
- Which evidence was used, from what source, and at what freshness?

- Which immutable policy snapshot applied?
- What system state and risk signals were evaluated?
- Why was the result ALLOW, HOLD, or DENY?
- Which gate version and configuration rendered it?
- How and where was the decision enforced?
- Was execution successful, denied, held, expired, or conducted under break-glass?

Audit roles should be read-only with respect to ProofRecords. E1 deployments require documented reconciliation between Effector records and decision digests.

9. Exception and Change Management

Treat changes to gate software, policy, configuration, keys, EBP, approver sets, and agent identity as governed events. Use shadow evaluation before activating new policy. Prefer drain-and-swap rollout so in-flight decisions complete against the prior immutable snapshot.

Emergency access should use pre-authorized, single-use, action-bound, time-bound break-glass instruments. Their issuance should itself be governed. Use must generate a VDR when the gate returns and trigger mandatory review.[1]

Architecture Pattern: API and Tool Calls

For API tools, deploy the ExecutionProof enforcement interface as an E2 API gateway, tool-call proxy, MCP gateway, or service-mesh control point.

1. Agent proposes a structured call.
2. the ExecutionProof enforcement interface canonicalizes method, endpoint, target, parameters, data classification, and environment.
3. Verification evaluates identity, authority, evidence, policy, state, and risk.
4. ALLOW produces an action-bound decision.
5. Gateway rechecks signature, digest, use, and validity.
6. Only then does the request reach the API.
7. Effector outcome is correlated with the ProofRecord.

The agent must not retain alternate credentials or routes around the gateway.

Architecture Pattern: Financial Operations

Treat payment creation, transfer, trade, disbursement, refund, account change, and approval as separate action classes as appropriate. Relevant policy fields can include amount, currency, beneficiary, account, business purpose, cumulative exposure, velocity, segregation of duties, and evidence freshness.

Above-threshold actions can require human or lineage-independent M-of-N approval. ALLOW should bind the exact beneficiary, amount, account, and transaction type. Changing any field requires a new decision. Shared quota state is necessary to prevent distributed gates from independently exceeding global limits.

Architecture Pattern: Privileged Actions

For infrastructure deployment, identity changes, secret access, production database modification, shell execution, or security-policy changes, prefer E3.

- Keep privileged credentials outside the agent environment.
- Require approved change evidence and current system-state checks.
- Issue a short-lived, action-specific capability only after ALLOW.
- Bind capability use to target, parameters, identity, and validity.
- Record execution results and reconcile them with the ProofRecord.
- Govern changes to the gate and its keys through a separate gate or documented M-of-N control.

Architecture Pattern: Multi-Agent Delegation

For each consequential hop:

1. Authenticate the proposing agent and session.
2. preserve the originating authority and complete delegation chain;
3. verify that delegation permits the specific action;
4. apply the receiving agent's own EBP capability and consequence rules;
5. create a new action digest and decision;
6. create a new ProofRecord;
7. prevent the prior hop's ALLOW from being reused.

Transport authentication remains necessary. Per-hop RF-100 governance does not replace secure inter-agent messaging.

Test and Assurance Evidence

A pilot should produce, at minimum:

- EBP and policy approval records;
- actor, agent, tool, and Effector inventory;
- architecture and trust-boundary diagrams;
- bypass-path assessment;
- canonicalization and digest test vectors;
- evidence-source authentication tests;
- freshness and expiry tests;
- authority revocation tests;
- HOLD escalation and mandatory re-verification tests;
- DENY completeness tests;
- replay and single-use tests;
- parameter-substitution tests;
- gate-signature verification tests when asymmetric signing is available;
- gate, network, evidence-service, authority-service, and clock fault injection;
- velocity consistency and concurrent-consumption tests;
- multi-agent delegation and per-hop tests;
- E1 reconciliation tests where E1 is used;
- audit reconstruction demonstrations;
- signing-key protection and rotation evidence;
- retention, access-control, minimization, and deletion tests;
- policy shadow-mode and rollback results;
- residual-risk register.

RF-100.D.2 specifically requires fault testing to demonstrate no execution without a valid, authenticated, in-window decision.[1]

Policy Ownership and Human Escalation

Assign accountable owners for:

- EBP scope and consequence classification;
- action policy;
- authority and delegation rules;
- evidence sources and freshness;
- agent identity and capability;
- human approval and M-of-N design;
- injection-risk handling;
- privacy and retention;

- cryptography and trusted time;
- gate availability and break-glass;
- assurance testing and residual risk.

Escalation service levels should reflect operational needs, but availability pressure must not silently convert HOLD or an absent decision into ALLOW. Break-glass must follow the narrow mechanism permitted by RF-100.7.9.

Operational Limitations

Implementation teams should explicitly document:

- actions not covered by the EBP;
- effectors or interfaces that cannot be interposed;
- E1 cooperative paths;
- identity or evidence sources without strong authentication;
- latency and availability dependencies;
- distributed-state limitations affecting velocity controls;
- inability to verify some model, memory, or supply-chain properties;
- risks from compromised gate operators or effectors;
- records that do not yet support offline verification;
- cryptographic controls not yet conforming to RF-100.8.4;
- privacy constraints on record content;
- false positives and false negatives in injection detection;
- manual escalation capacity and reviewer workload;
- conditions under which the pilot must stop or revert.

Concise Control Implementation Checklist

☐ Versioned EBP defines every Governed Action and consequence tier.
☐ All execution and bypass paths are inventoried.
☐ Policy is content-addressed and immutable per decision.
☐ Agent identity includes model/version, lineage, and deployment context.
☐ Authority is explicit, current, external to the agent, and delegation-aware.
☐ Required evidence is authenticated, integrity-protected, and freshness-checked.
☐ Intent Records and model context are not treated as Evidence.
☐ Independent ExecutionProof enforcement interface verification occurs before execution.
☐ Only ALLOW/HOLD/DENY outcomes are accepted.
☐ HOLD routes to authenticated escalation and full re-verification.
☐ Invalid, absent, expired, or consumed decisions fail closed.
☐ E2/E3 Enforcement Points remain outside the actor's control.
☐ Execution parameters match the authorized digest.
☐ ALLOW is time-bound and single-use unless expressly bounded otherwise.
☐ Velocity and cumulative limits use compliant shared or partitioned state.
☐ Injection signals are out-of-band and persist for the session.
☐ Every consequential multi-agent hop is independently governed.
☐ Every decision creates a complete, durable ProofRecord.
☐ VDR-oriented records support asymmetric signing and offline verification before conformity is claimed.
☐ Gate keys are isolated and gate administration is governed.
☐ Trusted-time drift is defined and tested.
☐ Fault injection demonstrates no execution without a valid decision.
☐ Audit reconstruction is demonstrable from records and referenced artifacts.
☐ Exceptions, break-glass, policy changes, and agent changes are governed.
☐ Residual risks and implementation gaps are documented.

Appendix: Condensed Enterprise Compliance Summary Matrix

Abbreviations: AI RMF functions: **GOV** = GOVERN, **MAP** = MAP, **MEA** = MEASURE, **MAN** = MANAGE. NIST AI 600-1 dimensions: **CONF** = Confabulation, **HAI** = Human-AI Configuration, **II** = Information Integrity, **IS** = Information Security, **DP** = Data Privacy, **VCI** = Value Chain and Component Integration. RF-100 enforcement: **E1** = Cooperative, **E2** = Interposed, **E3** = Infrastructure-enforced. **VDR** = Verification Decision Record; ExecutionProof's branded implementation pattern is the **ProofRecord**.

OWASP item	NIST control/function	RF-100 section(s)	ExecutionProof mechanism
ASI01 Goal Hijack	GOV, MAP, MAN; SI-10; IS	3; 4; 5; 7; AI.2, AI.4a, AI.7, AI.10	ExecutionProof enforcement interface: scope, evidence, injection-risk, and deterministic parameter gate
ASI02 Tool Misuse	MAN; AC-3, AC-6, CM-7	6.1; 7.1, 7.10; 8.1; 9; 11; AI.2	E2 tool/API/MCP proxy; E3 action-bound capability
ASI03 Identity & Privilege Abuse	GOV; AC-2, AC-3, AC-6, IA-2, IA-10	6; 8.1; AI.1–AI.3, AI.8–AI.9, AI.12	Identity, capability, revocation, delegation, and adaptive approval binding
ASI04 Agentic Supply Chain	MAP, MEA; VCI; CM-7	3.2–3.5; 4.4; 8.1, 8.4–8.8; 12; AI.12	Component attestations as evidence; version and configuration governance
ASI05 Unexpected Code Execution	MAN; AC-3, AC-6, CM-7, SI-10; IS	3.1; 6.1; 7; 9; 11; AI.2, AI.10	Privileged execution broker; credentials released only after ALLOW
ASI06 Memory & Context Poisoning	MEA, MAN; CONF, II; SI-10	4; 5.2, 5.4; AI.4–AI.4a, AI.7	Separate context from authenticated evidence; risk-only semantic evaluator
ASI07 Insecure Inter-Agent Communication	GOV, MAN; AC-3, IA-2; II, IS	4.4; 6; 7.10; 8.1, 8.4; AI.8–AI.9	Authenticated per-hop proposals, delegation chains, distinct ProofRecords
ASI08 Cascading Failures	MAP, MEA, MAN; IA-10	3.1; 7.2–7.9; 9.3; AI.5–AI.6a, AI.8a; D.2	Shared velocity state, circuit-breaking HOLD/DENY, per-hop gating
ASI09 Human-Agent Trust Exploitation	GOV, MAN; HAI; IA-2, IA-10	5.4; 6.2; 7.3, 7.7; AI.4a, AI.11	Evidence-separated review, authenticated approval, full re-verification
ASI10 Rogue Agents	GOV, MAN; AC-3, AC-6, CM-7; IS	5; 6; 7; 9; 11; AI.1–AI.12; Annex C	Independent ExecutionProof enforcement interface boundary; E2/E3 containment; external authority

OWASP item	NIST control/function	RF-100 section(s)	ExecutionProof mechanism
Cross-cutting action inventory	MAP; CM-7	1–3; 11.1; AI.2, AI.5; C.1	Actor/tool/Effector inventory and EBP boundary declaration
Cross-cutting decision evidence	MEA; II; SI-10	4; 5.2; 8.1; 10.3	Authenticated evidence adapters and reconstructable ProofRecord
Cross-cutting policy governance	GOV; AC-3	3.2–3.5; 12.1	Content-addressed policy, shadow evaluation, governed rollout
Cross-cutting account lifecycle	AC-2; IA-2	6.1–6.3; AI.1, AI.8, AI.12	Authoritative identity integration and transaction-time revocation check
Cross-cutting least privilege	AC-6; CM-7	6.1; 11; AI.2, AI.9–AI.10	Scoped tools and action-specific, short-lived authority
Cross-cutting information flow	AC-4; DP, II	3.1; 4.4; 8.7; 9; 11	Source/destination/data-field policy and digest-bound transfer
Cross-cutting adaptive controls	IA-10; HAI, IS	5.2; 7.7; AI.5–AI.7, AI.11	Consequence-based authentication, approval, and escalation
Cross-cutting input validation	SI-10; CONF, IS	4.2–4.4; 5.4; 7.4; AI.4a, AI.7	Canonical schema validation, freshness, provenance, injection signals
Cross-cutting decision handling	MAN	5.3; 7.1–7.10	Precise ALLOW/HOLD/DENY routing and fail-closed behavior
Cross-cutting execution binding	AC-3, AC-6; MAN	7.10; 8.1; 9.1–9.3; 11	Signature/digest/TTL/use validation immediately before effect
Cross-cutting auditability	GOV, MEA; II, DP	8; 10; 13	Minimized, protected, durable, independently verifiable ProofRecord
Cross-cutting fault assurance	MEA, MAN; IS	7.2, 7.4; 13.1; D.2	Outage, partition, dependency, replay, expiry, and clock-fault testing
Cross-cutting gate governance	GOV; AC-3, AC-6	8.8; 12.1	Separate gate or M-of-N control for gate, policy, key, and approver changes
Multi-agent governance	GOV, MAP, MAN; AC-3, IA-2	8.1; AI.8, AI.8a, AI.9	Independent decision and complete delegation record at every hop

OWASP item	NIST control/function	RF-100 section(s)	ExecutionProof mechanism
Privacy of decision records	DP; AC-2, AC-3	8.2, 8.7; 10.2	Field minimization, protected audit access, retention and erasure controls

References

1. https://raw.githubusercontent.com/derekhone/rf-100/main/RF-100-v1.0-draft.md
2. https://genai.owasp.org/2025/12/09/owasp-top-10-for-agentic-applications-the-benchmark-for-agentic-security-in-the-age-of-autonomous-ai/
3. https://genai.owasp.org/download/52117
4. https://nvlpubs.nist.gov/nistpubs/ai/nist.ai.100-1.pdf
5. https://airc.nist.gov/airmf-resources/airmf/5-sec-core/
6. https://airc.nist.gov/docs/NIST.AI.600-1.GenAI-Profile.ipd.pdf
7. https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final
8. https://github.com/derekhone/rf-100
9. https://raw.githubusercontent.com/derekhone/rf-100/main/README.md
10. https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/
11. https://csrc.nist.gov/CSRC/media/Projects/risk-management/800-53%20Downloads/800-53r5/SP_800-53_v5_1-derived-OSCAL.pdf
12. https://csrc.nist.gov/files/pubs/sp/800/53/r5/ipd/docs/sp800-53r5-draft-baselines-markup.pdf
13. https://github.com/requie/LLMSecurityGuide
14. https://github.com/requie/AI-Red-Teaming-Guide
15. https://www.nist.gov/publications/artificial-intelligence-risk-management-framework-ai-rmf-10
16. https://www.nist.gov/publications/artificial-intelligence-risk-management-framework-generative-artificial-intelligence
17. https://csrc.nist.gov/CSRC/media/Projects/risk-management/800-53%20Downloads/800-53r5/SP_800-53B_derived-OSCAL.pdf
18. https://www.nist.gov/artificial-intelligence/ai-agent-standards-initiative
19. https://www.nist.gov/news-events/news/2026/02/announcing-ai-agent-standards-initiative-interoperable-and-secure
20. https://www.nccoe.nist.gov/sites/default/files/2026-02/accelerating-the-adoption-of-software-and-ai-agent-identity-and-authorization-concept-paper.pdf
21. https://csrc.nist.gov/pubs/other/2026/02/05/accelerating-the-adoption-of-software-and-ai-agent/ipd
22. https://csrc.nist.gov/News/2025/control-overlays-for-securing-ai-systems
23. https://csrc.nist.gov/csrc/media/Projects/cosais/documents/NIST-Overlays-SecuringAI-concept-paper.pdf
24. https://csrc.nist.gov/Projects/cosais/use-cases