

Global Private-Sector AI Assurance Landscape 2026

Under what conditions may AI decisions and actions be adopted and executed?

A comparative review of execution controls, verifiable evidence and confirmation of outcomes

Compare adoption decisions, execution controls and outcome checks as distinct assurances. Identify what is established and which connections remain unverified.

Research cut-off: 15 October 2026

Edition: Final source-checked edition (4th edition; complete English translation)

Coverage: 24 principal cases, with related specifications, evaluation and insurance initiatives

Analysis of public primary sources, research and development records.

Not third-party product certification, a market census or an assessment of patent validity.

Core questions in this report

24 cases compared on eight common axes / Availability versus assurance scope / Developer experiments versus external research / Technical history versus patent priority / Reproducible evaluation design

Bracketed reference numbers identify the public primary sources at the end. N1–N3 are non-public drafts reviewed only as supplementary material, separately from the public comparison.

EXECUTIVE SUMMARY

Executive summary: findings and limits

1 Runtime controls are no longer a single-company concept

AWS general availability, Microsoft public preview, NVIDIA execution infrastructure and Invariant-derived guardrails demonstrate multiple ways to inspect and constrain actions outside the model. Enforcement, however, extends only to the paths each implementation actually covers. [1, 2, 6, 7, 10]

2 A proof is not assurance of the whole business process

Rule checking, code or environment attestation, tamper detection and external outcome reconciliation provide different assurances. Primary documentation distinguishes PreFlight's asynchronous proofs, OpenShell's proof exclusions and Weaveframe's unimplemented features; product names alone do not. [15, 8, 23]

3 The depth of third-party scrutiny still varies substantially

The report separates public code and CI, vendor experiments, partner announcements and external comparative research. AgentDyn shows that some static defenses can lose utility on dynamic tasks. No common dataset was obtained that would support a safety ranking of all products. [39, 38]

4 Japan's technical history is better documented, not a priority ruling

The reported October–November 2025 origins of Responsibility OS were compared with relevant passages in retained drafts. Their identity with the documents actually filed remains unverified. An early filing date alone does not establish precedence for every later configuration. [40, 41] / N1–N3

5 International cooperation should connect policy goals and checkable evidence

Private-sector initiatives such as AARM, ACS and AP2 make it useful to discuss how policy goals connect to scoped evidence and execution conditions. The task is to link pre-deployment evaluation with operational controls, not to replace existing evaluations. [43, 5, 46]

Finding: externally enforced controls are becoming concrete.

The public evidence reviewed here does not establish widespread, empirically demonstrated end-to-end assurance from input authenticity to external outcomes.

Reading guide: methods and axes → 24-case matrices → technical profiles → evidence and claim audit → technical history and priority → shared specifications and evaluation design → conclusions → primary sources.

SCOPE & METHOD

1 Scope, methods and treatment of evidence

The subject is technology governing the conditions, authority, evidence, suspension, change and outcomes associated with adopting AI or automated outputs and executing actions in business processes. This report uses “adoption and execution assurance” as an analytical label. “Responsibility OS” denotes one company’s technology family, not a globally established product category.

Selection and exclusions

The previous 16 cases were reassessed and expanded to 24 through functional searches, checks of major platforms and established governance providers, and citation tracing in papers and specifications. Inclusion required a concrete public description of the control target and mechanism. The final review rechecked claims, dates, URLs and matrix classifications. Evaluation-focused providers were included where their evidence supports deployment decisions; general ethics policies, generic chat products and unsupported “safe” marketing were excluded from the main comparison.

Evidence class	What this report can establish	What it does not establish alone
Provider description	What is publicly described as a product or concept	Implementation completeness; production performance
Detailed specification / public artifact	Intervention points, inputs, outputs, exclusions and traceable artifacts	Production deployment; hidden paths
Developer experiment / CI	Recorded test outcomes for a particular setting and version	Independent replication; specification adequacy
User / partner announcement	Deployment or collaboration acknowledged by the other party	Audit of all functionality; sustained performance
External comparative study	Successes and failures within that experiment	Generalization to all products; certification of current versions

Limitations and interests

AI Assurance, Inc. commissioned this review and is also a comparison subject. Supplementary checks against its non-public drafts are separated from the public product comparison. The report does not present hands-on tests of every product or independently rebuilt formal proofs.

Searches primarily used English and Japanese, with supplementary Chinese-language AgentScope documentation. The design does not establish complete regional coverage or estimate market size. Search bias, undisclosed implementations and later updates remain limitations. “Not established” does not mean “does not exist.”

SELECTION REGISTER

2 Selection process and excluded areas

Functional search routes

Search group	Example terms / sources	Cases added or examined further
Execution boundary	agent runtime policy enforcement / pre-execution / SDK gateway	AWS, Microsoft, NVIDIA, Galileo, Cisco, IBM
Evidence / outcomes	proof-carrying agent actions / verifiable runtime / outcome assurance	ICME, EQTY, OSuite, Weaveframe, ProofGrid
Decision support	AI assurance deployment evidence / regulatory evaluation / formal verification	Resaro, LatticeFlow, Citadel, Imandra, Alinia
Counterevidence / comparison	CaMeL / dynamic agent benchmark / prompt injection defenses	AgentDyn, AgentSpec, limitations sections
Shared specifications / business	AARM / agent control specification / AP2 / AI insurance	CSA, Vanta, Google, MLCommons, insurance parties

This register records functional terms and sources needed to retrace the search; it is not an archive of every search-engine result. Rankings change, so inclusion decisions should be traced through the cited URLs, versions and passages. Search-result counts and market shares are not estimated.

Items outside the 24 principal cases

Item	Treatment and reason
Hanzo AI	Retained for follow-up, not the main comparison. Correspondence between the blog’s agent-proof claims and the current README’s AI Chain/settlement inventory was not established. [53, 54]
AARM / AP2	Analyzed as shared specifications and interoperability initiatives, not scored alongside company products. [43, 46]
AVERI and collaborators / AIUC / Armilla	Connections to model evaluation and risk transfer, distinct from infrastructure enforcing each action. [48, 49, 50]
AgentScope	Supplementary execution, sandbox and observation infrastructure. End-to-end assurance from adoption evidence to external outcomes was not established here. [51]
AgentSpec	Included as earlier academic research in the technical history, not counted as a private-sector product. [52]

Invariant Labs and Snyk are treated as one case of technology succession. CaMeL is a research prototype involving corporate research, not a generally available product. Classification reflects roles and evidence, not superiority. [11, 9]

COMPARISON FRAMEWORK

3 Eight axes: what assurance, over what scope?

These eight axes were defined for this comparative review. They are not a conformity test against a particular standard. The report does not total the axes into a score or rank companies by feature counts.

Axis	Question	Typical conflation
A Conditions / scope	Which rules, targets, purposes, versions and times are fixed?	Reusing a strong model evaluation for another purpose/version
B Enforcement	Where can execution be stopped? What bypasses remain?	Treating an optional tool or warning as a mandatory gate
C Action–authority binding	Whose approval or mandate binds which operation and arguments?	Reusing old approval for a modified action
D Facts / authenticity	Who observes inputs and outcomes, and checks provenance?	Treating signed input as true content
E Portable / recheckable evidence	What can another party recompute or check, using what?	Equating log access or a verification API with independent recomputation
F External outcomes	Is business-system state checked beyond the tool response?	Treating HTTP success or a model's completion claim as completion
G Change / stop / resume	How are revocation, expiry and changed states rechecked?	Assuming a policy update also revokes old credentials
H Formal assurance / implementation	Does a proved proposition correspond to source/deployed code?	Equating a formal-model proof with whole-system safety

How to read the matrices

DOC: a concrete specification or procedure is documented in a primary source. PROTO: described as a prototype or research implementation. DESIGN: a concept or reference design. OUT: explicitly outside the stated assurance scope. UNK: not established by the sources reviewed. A code applies only to the subfunction described in its cell, not to completion of the entire axis or operational certification.

An assurance chain can break at an unverified connection.

Rules → facts → decisions → execution → outcomes → re-verification must be examined separately.

Even a valid cryptographic proof cannot justify a business conclusion when the facts supplied to the proof are wrong. Conversely, a suitably isolated gate may prevent a specific action without a formal proof. These approaches are not necessarily substitutes.

24 CASES | AXES A-D

4 Cross-case matrix: conditions and boundaries (1)

DOC = documented; PROTO = prototype; DESIGN = concept; OUT = explicitly excluded; UNK = not established. This compares public descriptions, not hands-on test results.

Case / source	A Conditions / scope	B Enforcement	C Action / authority	D Facts / provenance
01 AWS [2, 3]	DOC: Cedar / Dogwood history conditions	DOC: within Gateway	DOC: principal, target, arguments	UNK: independent acquisition of business facts
02 Microsoft [5, 6]	DOC: ACS / policies	DOC: registered hooks / wrappers	DOC: principal and approval context; host-supplied	DOC: evidence providers; authenticity is separate
03 NVIDIA [7, 8]	DOC: explicit boundary policies	DOC: within sandbox	DOC: permitted resources / operations	UNK: truth of business facts
04 CaMeL [9]	PROTO: capabilities / data flow	PROTO: constrained interpreter	PROTO: provenance and permissions	PROTO: trusted / untrusted separation
05 Invariant [10]	DOC: history / condition rules	DOC: integrated agent paths	UNK: approval/mandate binding; history is A	DOC: collected traces; authenticity is separate
06 Galileo [12, 13]	DOC: central policies	DOC: calls wrapped by @control	UNK: approval/mandate binding to each action	DOC: observed traces; authenticity is separate
07 Cisco [14]	DOC: inspection / Gateway settings	DOC: supported SDK / MCP paths	UNK: mandate binding; I/O inspection documented	DOC: I/O checks, not factual truth
08 ICME [15]	DOC: supplied rules and action	DOC: mandatory hook; optional calls bypassable	DOC: checked action; execution identity depends on integration	OUT: truth of asserted facts
09 EQTY [18]	DOC: runtime policies	DOC: isolated Runtime controls; deployment UNK	DOC: credentials / delegation binding	DOC: environment attestation, not business truth
10 Traefik [20]	DOC: external authorization	DOC: traversed Gateway	DOC: delegation / external authorization	DOC: identity / path observations; record truth OUT
11 OSuite [22]	PROTO: action / decision / evidence contract	PROTO: SDK / Gateway; some ingestion-only paths	PROTO: approval-action binding	PROTO: observers / receipts; independence path-dependent
12 Weaveframe [23]	PROTO: mandates / conditions / evidence	OUT: preventive gate	PROTO: mandate-violation demo; approver authentication unbuilt	PROTO: provenance separation; independent acquisition unbuilt

Strength on some of A-D does not imply strong evidence portability or outcome confirmation. Read with E-H below. A deployment-evaluation workflow is not automatically a gate on individual actions.

24 CASES | AXES A-D

4 Cross-case matrix: conditions and boundaries (2)

DOC = documented; PROTO = prototype; DESIGN = concept; OUT = explicitly excluded; UNK = not established. This compares public descriptions, not hands-on test results.

Case / source	A Conditions / scope	B Enforcement	C Action / authority	D Facts / provenance
13 FlowX [24, 31]	DESIGN: regulation → rule; PROTO: one rule	DESIGN: observation / review controls	DESIGN: identity / role / action	DESIGN: telemetry; PROTO: synthetic logs
14 ProofGrid [25]	DESIGN: authority / conditions / actions	DESIGN: pre-execution control	DESIGN: origin / delegation / action	DESIGN: evidence / provenance
15 Resaro [26]	DOC: evaluation plan / operating conditions	UNK: per-action gate	UNK: per-action authority binding; deployment evidence is A	DOC: test data / observations
16 LatticeFlow [28]	DOC: frameworks → technical evaluation	UNK: per-action gate	UNK: per-action authority binding; evaluation mapping is A	DOC: technical evaluation results
17 Citadel [29]	DOC: agent context	DOC: controls in EAP	UNK: specific approval/mandate binding	DOC: behavior / risk detection
18 Imandra [33]	DOC: formalized specifications	UNK: general-purpose action gate	UNK: per-action approval binding	UNK: truth of business facts
19 Alinia [34]	DOC: semantic policy evaluation	UNK: mandatory post-verdict execution path	UNK: per-action approval binding	DOC: text evaluation; not authenticity
20 AI Assurance [36, 37]	PROTO: requirements / versions / adoption	PROTO: PoC draft output	UNK: real-party authority/approval binding; conditions are A	OUT: real-world validity of dummy facts
21 Holistic AI [55]	DOC: versioned endpoint policies	DOC: layered endpoint controls	DOC: endpoint / account / operation	DOC: endpoint observations; full facts UNK
22 Fiddler [56]	DOC: thresholds / sensitive-data rules	DOC: model Gateway	UNK: mandate binding; I/O checks documented	DOC: input / output inspection
23 IBM [57]	DOC: central policies / evaluation	DOC: connected control plane; bypass tests UNK	DOC: enterprise identity / access	DOC: connected-agent observations
24 Arthur [58]	DOC: app / model / environment	DOC: application enforces verdict	UNK: specific approval/mandate binding	DOC: multiple detectors / observation sources

Strength on some of A–D does not imply strong evidence portability or outcome confirmation. Read with E–H below. A deployment-evaluation workflow is not automatically a gate on individual actions.

4 Cross-case matrix: evidence and outcomes (1)

Re-verification is separated into record-tampering checks, decision recomputation, formal-proof checking and external outcome reconciliation. None is assumed to subsume the others.

Case / source	E Portable / recheckable	F External outcomes	G Change / stop	H Formal scope
01 AWS [2, 3]	DOC: decision records; independent certificate UNK	UNK: external state reconciliation	DOC: temporal-policy update invalidates sessions	UNK: full deployment proof
02 Microsoft [5, 6]	DOC: structured evidence / verdict; recomputation UNK	UNK: external state; response checks documented	DOC: interventions / interruption points	UNK: all-path / deployment proof
03 NVIDIA [8]	DOC: inclusion check; general evidence interoperability UNK	UNK: external state; task-safety proof OUT	DOC: boundary rechecking	DOC: policy inclusion; enforcement proof OUT
04 CaMeL [9]	PROTO: public research artifacts; business evidence interoperability UNK	UNK: persistent external outcomes	PROTO: capabilities; limited dynamic adaptation	PROTO: design-level safety argument; full deployment UNK
05 Invariant [10]	DOC: detection / reasons; independent certificate UNK	UNK: state beyond tools	DOC: rule changes / blocking	UNK: complete proof including probabilistic detection
06 Galileo [13]	DOC: audit traces; independent recomputation UNK	UNK: persistent external outcomes	DOC: configuration changes / blocking	UNK: formal assurance of decision model
07 Cisco [14]	DOC: inspection results; general re-verification UNK	UNK: external state; tool-response checks documented	DOC: inspection- based blocking	UNK: proof covering unsupported paths
08 ICME [15]	DOC: asynchronous proofs; pinned independent reproduction UNK	UNK: actual external effects	DOC: allow/deny; full credential revocation UNK	DOC: computation- proof claim; inconsistent LLM-role descriptions
09 EQTY [18]	DOC: provenance / attestation; end-to-end reproduction UNK	UNK: business outcomes	UNK: concrete revocation / resumption procedure	UNK: proposition- implementation link; attestation is E
10 Traefik [20]	DOC: Merkle / witnesses; completeness separate	UNK: external-system state	DOC: credential / authorization integration	UNK: full deployment proof
11 OSuite [22]	PROTO: replay / history; protected evaluation data	PROTO: outcome records / updates; limited authenticity	PROTO: hold / reapproval / update	UNK: theorem-prover correspondence
12 Weaveframe [23]	PROTO: hashes / trails; stronger external preservation is DESIGN	PROTO: demo ERP; production independence unbuilt	DESIGN: uncertainty / exception; not emitted by prototype	UNK: full deployment proof

UNK under H does not assert that no formal methods are used. It means the reviewed sources do not establish the correspondence between the relevant proposition and the deployed implementation.

4 Cross-case matrix: evidence and outcomes (2)

Re-verification is separated into record-tampering checks, decision recomputation, formal-proof checking and external outcome reconciliation. None is assumed to subsume the others.

Case / source	E Portable / recheckable	F External outcomes	G Change / stop	H Formal scope
13 FlowX [24, 31]	DESIGN: audit evidence; PROTO: synthetic-log checks	UNK: complete external outcome reconciliation	DESIGN: hold / review	UNK: correspondence to formal proof
14 ProofGrid [25]	DESIGN: verification receipts	DESIGN: outcome classes; retains UNKNOWN	DESIGN: expiry / revocation / stop	UNK: deployment / formal proof
15 Resaro [26]	DOC: evaluation / deployment evidence; execution-proof interoperability UNK	UNK: individual external effects; performance testing separate	DOC: re-evaluation / change management	UNK: whole-operation formal proof
16 LatticeFlow [28]	DOC: evaluation evidence; action-proof interoperability UNK	UNK: each action's external effects	DOC: continuous evaluation	UNK: proof of complete legal conformity
17 Citadel [29]	DOC: evaluation / monitoring; independent execution proof UNK	UNK: persistent external outcomes	DOC: detection / restrictions	UNK: whole-operation formal proof
18 Imandra [33]	DOC: formal analysis / counterexamples; translation separate	UNK: actual external outcomes	DOC: re-analysis after change	DOC: formal-model analysis; deployed identity separate
19 Alinia [34, 35]	DOC: some public evaluation data; certificate recheck UNK	UNK: external outcomes	UNK: full credential revocation	UNK: formal assurance; statistical evaluator
20 AI Assurance [36, 37, 38]	PROTO: Python / JS recomputation	OUT: actual business-system enforcement / outcomes	PROTO: hold / return; stopping in separate model	DOC: limited Lean-model CI; real system OUT
21 Holistic AI [55]	DOC: versioned / timed records; independent computation proof UNK	UNK: business-system outcomes	DOC: re-evaluation / response status	UNK: all-path formal proof
22 Fiddler [56]	DOC: verdict records; independent certificate UNK	UNK: persistent external tool effects	DOC: reject / redact / fail closed	UNK: formal proof of detector
23 IBM [57]	DOC: audit / tracking; independent execution proof UNK	UNK: external state; completion-rate evaluation separate	DOC: pre-release evaluation / ongoing management	UNK: full deployment proof
24 Arthur [58]	DOC: evaluation / verdict records; independent recomputation UNK	UNK: business outcome reconciliation	DOC: blocking / change monitoring	UNK: full deployment proof

UNK under H does not assert that no formal methods are used. It means the reviewed sources do not establish the correspondence between the relevant proposition and the deployed implementation.

5 Different enforcement points: three platforms

01 AWS — AgentCore Policy

Stage: general availability confirmed; temporal policies additionally reviewed in current documentation.

The Gateway evaluates Cedar policies. Current Dogwood documentation covers approval history and time windows; adding or updating a temporal policy invalidates existing sessions. The next request receives a 409 response, and a new session starts with empty history. [1, 2, 3]

Boundary: distinguish ENFORCE from LOG_ONLY. History is session-scoped and depends on identity/token propagation and call-path integration; it does not confirm external completion. Temporal features are not backdated to the March general-availability launch.

02 Microsoft — Agent Governance Toolkit / ACS

Stage: public preview; public repository and primary specification reviewed.

ACS passes evidence and context to rules at eight intervention points, separating the verdict from host-side enforcement. The pinned README places the decision engine upstream in ACS, with AGT providing host APIs and adapters. [4, 5, 6]

Boundary: history and approval state depend on a host-supplied snapshot. Enforcing configured denials is distinct from intercepting every path; the original announcement and current architecture also differ. Post-tool inspection is not independent confirmation of persistent external state. Code was not executed in this review.

03 NVIDIA — OpenShell

Stage: public execution infrastructure and policy-verification documentation reviewed.

OpenShell isolates agents and bounds resource access. Its policy prover checks whether a permission policy fits a supplied boundary; unsupported or inconclusive cases are not treated as success. [7, 8]

Boundary: the proof concerns policy inclusion. Documentation explicitly excludes the adequacy of least privilege, actual sandbox enforcement and safety of the overall task.

Comparison: gateways, in-application hooks and isolated environments are different integration choices.

Each requires disclosure of bypass paths and downstream processes it cannot stop.

5 Context-sensitive checks: research and SDKs

04 CaMeL — A data-flow-based research design

Stage: public research and prototype, including corporate research.

The design separates processing of untrusted content from privileged planning and execution, using data provenance and capability constraints. [9]

Boundary: the research's static, constrained execution setting is not equivalent to a dynamic, long-running business workflow. External counterevidence is discussed in Section 6.

05 Invariant / Snyk — History-aware guardrails

Stage: published guardrail technology; post-acquisition succession is counted as one case.

Policies can express relationships between tool sequences and values, not only inspect an isolated output. Contextual checks can be combined with statistical detection. [10, 11]

Boundary: detector accuracy, rule adequacy and paths outside the integration remain separate issues. Exact condition checking must be distinguished from probabilistic detection.

06 Galileo — Agent Control

Stage: concrete SDK configuration procedures reviewed.

Centrally managed policies are applied to LLM and tool inputs and outputs. The documented procedure includes wrapping the relevant calls with @control. [12, 13]

Boundary: easy configuration changes do not mean implementation changes are unnecessary. Comprehensive enforcement over uninstrumented calls was not established.

07 Cisco — AI Defense

Stage: technical descriptions of SDK/MCP and gateway approaches reviewed.

The integration creates inspection paths for supported SDK or MCP operations, connecting protected calls to an inspection service or gateway. [14]

Boundary: supported and unsupported calls must be separated. Screening a tool response for risk does not verify business success or rollback.

5 Unpacking proof: cryptography and environments

08 ICME Labs — PreFlight

Stage: product and technical documentation reviewed; no independent reproduction of the complete product.

PreFlight combines action-rule checks with cryptographic evidence. It separates immediate verdicts from asynchronous proof generation and discloses that the JOLT foundation is alpha-stage and unaudited. The underlying paper alone does not establish product-wide safety. [15, 16, 17]

Boundary: the truth of input facts is excluded; enforcement depends on mandatory-hook integration or equivalent controls. The same FAQ describes both compile-time-only LLM use and dual checks including a runtime custom LLM check. Implementation evidence is needed to resolve the LLM's role and resistance to extraction errors.

09 EQTY Lab — Verifiable Runtime

Stage: Runtime announcement and partner evidence for the underlying technology reviewed; full deployment not independently confirmed.

The approach links isolated infrastructure, credentials and delegation, policies and execution-origin attestations. Intel's announcement also confirms collaboration on the compute foundation. [18, 19]

Boundary: a 2024 foundational partnership is not an independent audit of the entire 2026 Runtime. Establishing that an environment had a particular configuration is different from establishing business-appropriate action.

10 Traefik Labs — Sovereign Trust Plane

Stage: early-access and planned-GA announcements reviewed; completed GA was not established in this review.

The design connects delegation, external authorization and evidence verification. Merkle-based records and external witnesses provide a concrete mechanism for detecting subsequent rewriting. [20]

Boundary: actions never logged, and the truth of logged content, require separate checks. The passage of a planned late-September GA date is not evidence of completed availability.

Measure verdict latency separately from proof-completion latency.

A design that executes before a proof is complete also needs a response to later proof failure.

5 After execution: designs versus implementations

11 OSuite / Ond — PCAA

Stage: paper and trial program; public aggregates are separated from non-public evaluation data.

PCAA treats the action, approval, execution path and outcome record as a common contract-like structure. Enforcement differs between SDKs, gateways, observation and record-ingestion paths. [21, 22]

Boundary: some evaluation scenarios and materials are non-public, so aggregate results alone cannot be reproduced. The term “proof-carrying” does not establish theorem-prover-checked correctness.

12 Weaveframe — Independent outcome assurance

Stage: partial prototype with explicit exclusions and implementation gaps.

The distinguishing problem is to reconcile mandate, execution and outcome separately, rather than rely solely on the executor’s declaration of completion. [23]

Boundary: it is not a preventive gate. Source and approver authentication, independent read paths and outputs for proposed uncertainty/exception states include unimplemented elements. Demo-ERP reconciliation is not production independence.

13 FlowX.AI — GAVEL

Stage: product architecture and developer experiment on one rule; full deployment not independently confirmed.

GAVEL links regulatory requirements, rules and audit evidence. Technical paper v1.1 tests one rule against 165 synthetic traces and reports that LLM checkers detected cases missed by a regular-expression checker. [24, 31]

Boundary: the experiment covers one rule and self-generated logs, not the full GAVEL system, legal compliance or production RBAC and evidence collection. The illustrative rule is not the law itself.

14 The ProofGrid

Stage: conceptual and illustrative technical design.

The design connects authority origins, delegation, operations and outcomes, distinguishing TIMEOUT from an unknown outcome. [25]

Boundary: retain the documentation’s own conceptual/illustrative status. This review does not treat production operation or enforcement over all paths as verified.

5 From evaluation results to deployment decisions

15 Resaro — Approved Intelligence Platform

Stage: evaluation/assurance product; activity corroborated by the organization hosting its contributions.

The platform relates operating conditions to evaluation and verification, organizing evidence for deployment decisions. AI Verify Foundation also documents development contributions and evaluation activity. [26, 27]

Boundary: an evaluated system is not the same as each individual execution. A mandatory gate on all tool calls or independently proved external outcomes was not established.

16 LatticeFlow AI — Connecting frameworks and technical evaluation

Stage: platform announcement in July 2026.

The platform maps framework requirements to technical evaluation. Its relevance lies in connecting governance frameworks to testing and evaluation work, rather than treating them solely as checklists. [28]

Boundary: requirement mapping, executable evaluation and legal conformity are distinct. The number of supported frameworks is not a measure of legal sufficiency or formal assurance strength.

17 Citadel AI — From evaluation and governance to agents

Stage: established evaluation business distinguished from the Agents EAP announced on 7 October 2026.

Citadel announced an EAP addressing agent context and behavioral risks across platforms. It separately announced plans to provide a solution to Sumitomo Mitsui Banking Corporation. [29, 30]

Boundary: the 2 October banking announcement describes planned provision and is separate from the 7 October new-product EAP. It is not evidence that the latter completed production deployment or has an operational track record.

What must accompany evaluation evidence

The review's analysis is that deployment evidence used at runtime needs to specify what was examined, for which configuration, purpose and period. A pass label alone cannot establish whether evidence remains applicable after an operational change.

5 Formalization, semantic evaluation and adoption

18 Imandra — CodeLogician

Stage: formal-analysis product and original research.

The approach converts specifications and code into analyzable forms for formal checks and counterexample generation. The assessment concerns what has actually been formalized. [32, 33]

Boundary: correctness of the translation from natural-language requirements, and correspondence to deployed code, remain separate questions. A system analyzed as a paper’s example is not thereby a customer deployment.

19 Alinia — Seny

Stage: a model evaluating conformity with policies and related requirements, with some public evaluation data.

Seny semantically evaluates text against legal, policy and related requirements. It is relevant as a component for judging content that is difficult to capture in explicit rules alone. [34, 35]

Boundary: this is probabilistic evaluation, not formal proof or authority enforcement. The dataset description says expert-authored English examples were machine-translated into eight languages and 5% were spot-checked by native speakers. This is not expert review of every item or independent replication.

20 AI Assurance, Inc. — Public Responsibility OS-related work

Stage: public PoC and limited formal model; not enforcement over an entire production operation.

The civic PoC derives adoption status from conditions and evidence and recomputes records. For the EU-related Lean model, the review confirmed recorded success of two CI verification steps for a specified commit. [36, 37, 38]

Boundary: the PoC controls its own draft-answer output, not an entire administrative system. CI success checks proofs in a limited model; it does not certify factual authenticity, physical-system safety or compliance with an entire law.

Formal proofs, semantic evaluation models and execution gates answer different questions. Names or the shared word “safety” do not establish interchangeability.

5 Established governance providers are also expanding

21 Holistic AI — EndLayer

Stage: endpoint product architecture and availability information reviewed.

EndLayer connects versioned policies with endpoint observations, describing Sensor, Decoder and Guardian layers and endpoint inspection/blocking mechanisms. [55]

Boundary: periodic asset discovery is not action-time blocking. Configuration differences, unobserved paths and completeness of downstream business outcomes need further examination.

22 Fiddler AI — Guardrails

Stage: gateway-based implementation procedures reviewed.

A concrete LiteLLM example inserts checks before and after model calls, applying redaction or rejection. Fail-closed configuration is also shown. [56]

Boundary: controlling model traffic does not imply control of all tool executions on other paths. Detector-threshold performance and any semantic changes caused by redaction require separate measurement.

23 IBM — watsonx Orchestrate Agent control plane

Stage: product documentation for centralized controls, evaluation, identity and audit reviewed.

The platform covers connected agents from pre-deployment evaluation to continuous monitoring and policy management. Integrated platforms from major providers also belong in this comparison. [57]

Boundary: a broad product description does not establish bypass prevention on every path or implementation of portable proofs that a verifier can take away.

24 Arthur AI — Agent Security & Governance

Stage: product documentation explicitly distinguishes detection, observation, verdicts and integrator responsibilities.

Arthur evaluates policies for the application, model and environment. It explicitly assigns verdict enforcement to the calling application or gateway. [58]

Boundary: returning a verdict and enforcing it without bypass are assessed separately. Audit records are not equivalent to reconciliation against external business outcomes.

EVIDENCE & COUNTEREVIDENCE

6 Corroboration and unfavorable evidence

Research results depend on their experimental setting

Source / relationship	Reported result / status	Scope of inference
CaMeL v2 Developer research	The abstract reports 77% task completion on AgentDojo versus 84% without defense. [9]	Evaluation of the designed capability/data-flow constraints. Do not mix versions or settings.
AgentDyn v1 External university researchers	Multiple defenses tested on 60 dynamic tasks and 560 attack cases. Large utility losses are reported for approaches strongly tied to an initial plan. [39]	External counterevidence, but a limited benchmark—not all 24 current products under common conditions.
PCAA v1 Developer evaluation	Actions, paths and evidence are evaluated, but scenarios and other materials include non-public portions. [22]	Research aggregates, not confirmed third-party replication.
GAVEL v1.1 Developer experiment	On 165 synthetic logs, LLM checkers detected cases missed by a regular-expression checker for one rule. [31]	One-rule experiment, not validation of full GAVEL or legal compliance.

Interpretation: low false-authorization rates are insufficient without utility for legitimate work. AgentDyn also reports cases where model-layer defenses achieve a comparatively favorable balance. This report does not conclude that formal or externalized controls always outperform alternatives, nor subtract results across CaMeL and AgentDyn to calculate a decline in performance.

What user and partner sources corroborate

Other-party source	Relationship corroborated	Not established
Intel → EQTY [19]	Compute-verification foundation collaboration	Independent certification of the full 2026 Runtime.
AI Verify Foundation → Resaro [27]	Development contributions / evaluation activities	Performance assurance of every product.
MLCommons → joint evaluation [48]	Pilot protecting non-public models and evaluation materials	Comprehensive safety of the evaluated model.
ElevenLabs → AIUC [49]	Use of certification / insurance	A guarantee preventing every incident.
Chaucer → Armilla [50]	Insurance collaboration / product structure	Elimination of coverage conditions or exclusions.

“Independent” has multiple meanings here. Confirmation by another legal entity, public code, replication by external researchers and regulatory certification are distinct. A prominent publisher is not a substitute for third-party audit.

CLAIM AUDIT

7 Claim audit: revised interpretations

Potential claim / conflation	Source-check result	Treatment in this report
Is Microsoft GA, with all-path enforcement?	Public Preview. ACS decides; the host enforces. [5, 6]	Explicit dependence on registered hooks and supplied context.
Does NVIDIA's proof establish whole-operation safety?	Inclusion checks do not prove enforcement or task safety. [8]	Proof target and exclusions stated.
Are PreFlight's proof timing and LLM role clear?	Asynchronous proof. Compile-time-only and runtime LLM-check descriptions coexist. [15]	No claim of complete LLM independence.
Does AWS carry old approvals forward unconditionally?	Temporal-policy changes invalidate sessions; logging-only mode is distinct. [3]	Change-time control added to matrix.
Is Traefik GA because its planned date passed?	Planned GA is documented; completed GA not established here. [20]	Plans separated from observed availability.
Can an outsider fully reproduce OSuite's results?	Some evaluation data is protected or non-public. [22]	Developer experiment; independent reproducibility unconfirmed.
Does Weaveframe implement uncertainty outputs?	Independent reads, authentication and uncertainty/exception outputs include unimplemented items. [23]	Prototype and proposal separated in matrix.
Does Citadel's bank announcement validate the Agents product?	Planned bank provision and the new-product EAP are separate. [30, 29]	Different functions and versions kept separate.
Does Hanzo's blog match its current proof inventory?	README focuses on AI Chain, provenance and settlement. [53, 54]	Correspondence unresolved; retained for follow-up.
Does Responsibility OS CI certify the real system?	Success recorded for a specified commit of a limited Lean model. [37, 38]	Developer CI records, not independent re-execution.

Inconsistencies within a source are retained as unresolved questions. These distinctions do not dismiss the work; they prevent evidence from being transferred to a different version, function or assurance claim.

8 Japan’s technical lineage: draft cross-checks

The reported October–November 2025 origins were checked against both the developer’s public article and retained drafts. The table asks whether a feature occurs in the draft examined—not whether that exact text was filed with the patent office on that date. [40] / N1–N3

History in public article	Draft and passages examined	Cross-check / remaining limit
2025-10-30 JP application 2025-183162 Finite-closure energy control	N1: non-public draft [0028]–[0031], [0052], [0057]–[0058], [0070]–[0071]	Descriptions of regulatory/contract changes, allowed commands, post-command checking, reconfiguration and audit were found. Identity with the filed original is unverified.
2025-10-31 JP application 2025-185135 Communication-integrated zero trust	N2: non-public draft [0021]–[0028], among others	State management incorporating time, policy and purpose; safe-side suspension and rejoining are described. Identity with the filed original is unverified.
2025-11 JP application 2025-201777 ADIC recomputation / proof	N3: ADIC-related draft [0007]–[0013], among others	Exact arithmetic records and independent re-verification are described. Correspondence to this application number and identity with the filed version are not established.

What the cross-check strengthens

The drafts contain functional descriptions related to the public article’s execution-control and independent-reverification lineages. This adds consistency between documents, not independent proof of the original filing content or international precedence.

What this cross-check does not establish

The exact originally filed disclosure, authenticity of priority documents, disclosure of later claimed combinations, novelty and inventive step over global prior art, and production maturity remain separate issues. Non-public drafts were not used to score public functionality against other companies.

To avoid additional disclosure of non-public technology, the technical discussion stays within the existing public article. Full drafts, full claims and personal information are not reproduced.

9 Publication dates versus filing dates

Period	Event and source	Meaning of the date
2024-12	Intel announces compute-verification collaboration with EQTY. [19]	Publication of foundational collaboration, not completion of the later Runtime.
2025-03-04	CaMeL, AgentSpec, CodeLogician and Invariant research/technical publications. [9, 52, 32, 10]	Earlier overseas public work on constraints and verification outside AI models.
2025-09	AP2 published. [46]	Common treatment of intent, approval and payment instructions.
2025-10-11	Developer article reports two finite-closure-control filings and ADIC filing timing. [40]	Japanese filing dates, not publication dates of the article.
2026-01-06	Developer article describes extensions to responsibility state, evidence preservation, AI decisions, cyber operations and logistics. [40]	Development/filing history, not production completion of all functions.
2026-03-06	AWS GA, Microsoft publication, EQTY Runtime, ACS, PCAA and related announcements. [1, 4, 18, 5, 22]	Mixture of publication, availability and research stages.
2026-07-17 2026-08-30	Responsibility OS history article published and updated to explain two lineages. [40]	Retrospective publication of the reported 2025 filing history.
2026-09-10	Traefik, Weaveframe and Citadel’s Agents EAP. [20, 23, 29]	Early access, partial prototypes and EAP are not equated with GA.

What the chronology supports

Related public research and products existed outside Japan by spring 2025. The Japanese filings reported for October and November 2025 also precede several 2026 announcements. Both facts belong in the record.

What the chronology cannot establish

A later publication date does not imply a later start to research. An earlier filing does not establish anticipation of every feature of another company’s later product. Product announcements, research publications and patent filings are not a single precedence ranking.

PATENT BOUNDARIES

10 Patent priority and functional comparisons

Two lineages do not automatically give their combination an early priority date

Section 6.05 of WIPO’s PCT International Search and Preliminary Examination Guidelines explains that separate earlier applications containing feature C and feature D do not automatically establish an earlier priority date for their combination. There are limits to assembling a combination by simply mosaicking multiple priority documents. [41]

The necessary disclosure need not appear only in the earlier application’s claims. The question is whether the specific combination is supported by the description, claims and drawings as a whole. General related terminology must be distinguished from concrete disclosure supporting the later invention. [41, 42]

Comparisons possible now, and evidence still needed

Comparison unit	Possible with current material	Evidence for a rigorous determination
Allowed scope / execution gate	Distinguish N1/N2 descriptions from gateway, sandbox and SDK roles elsewhere	Originally filed text and date evidence; pinned product specifications
Independent re-verification	Distinguish N3/public PoCs from logs, signatures and proofs	Draft-application correspondence; verifier, inputs and expected outputs
Post-execution state difference	Compare public history with outcome receipts and external reconciliation	Relevant later claims; earlier disclosure of the complete configuration
Assurance carried to later stages	Apply common axes to assumptions, versions, purposes and revocation	Claim-by-claim feature chart and full prior public disclosures

The technical history is better supported, but this report does not conclude that October 2025 is the established priority date of the entire Responsibility OS, that another party infringes, or that the technology was the world’s first.

A formal assessment requires claim-specific disclosure passages, feature combinations, publication and filing dates, and the applicable jurisdiction. The abstract functional matrix is not a novelty or inventive-step chart.

INTEROPERABILITY & HAIP

11 Shared specifications and international dialogue

Shared initiative	Reviewed subject	Remaining question
AARM / CSA / Vanta [43, 44, 45]	Nine requirements: Core R1–R6 are MUST; Extended R7–R9 are SHOULD. They address pre-action intervention, context, verdicts, tamper-evident records and identity binding.	The specification requires working-group review of an evidence package. A listing or self-claim alone is not treated as conformity or legal certification.
Microsoft ACS [5]	Evidence providers, intervention points, verdicts and the enforcement host’s role.	A shared format does not establish identical decisions or enforcement coverage; interoperability tests are needed.
AP2 / FIDO [46, 47]	Binding user intent, mandates and payment instructions through shared specifications.	Payment authority is distinct from delivery, service fulfillment and actual loss confirmation.

A two-way connection between policy and private technology

The proposed HAIP question is not which single company’s system to adopt, but which conditions, evidence and unknowns allow different parties to make a shared assessment. Policymakers articulate goals; private actors present checkable mechanisms and limitations; practical experience feeds back into goals and evaluation methods. This is an agenda proposal derived from the review.

Discussion question	Proposed concrete output
Can assurance scope travel?	A common evidence description carrying target, purpose, version, period, assumptions and unknowns
Can the reason for stopping be rechecked?	Conformance tests comparing allow, deny and hold decisions and their grounds across implementations
Can incident sharing prevent recurrence?	A format separating invalidated conditions, bypass paths, missing evidence and actual impact

Evaluations involving AVERI and the insurance connections of AIUC and Armilla are separate layers, not substitutes for runtime controls. Providing evaluation evidence for underwriting is not the same assurance as preventing an incident during execution. [48, 49, 50]

PROPOSED TEST PROTOCOL | NOT EXECUTED

12 Reproducible evaluation: twelve proposed cases

This is a proposed replication protocol developed for this report, not completed test results. Different implementations receive the same business requirements, with inapplicable or unsupported cases recorded explicitly. Use simulated business systems and synthetic data, not real payments, production equipment or actual personal information.

Test	Perturbation	Behavior / evidence to check
T01 Normal case	Conditions, authority and evidence all match	Authorize and retain records corresponding to the required external state.
T02 Missing evidence	Remove required observation or approval	Do not authorize on missing evidence; distinguish hold and denial reasons.
T03 Version mismatch	Apply old evidence to new rules	Do not unconditionally reuse another version's pass evidence.
T04 Post-approval change	Change an approved recipient, quantity or other parameter	Detect approval-execution mismatch immediately before execution.
T05 Post-check change	Change authority or target state after checking	Recheck at execution or use an appropriate atomic commitment.
T06 Revocation / expiry	Revoke approval or let it expire	Do not continue executing with stale tokens or cached decisions.
T07 Bypass path	Try direct API, another SDK or another connection	Block within declared scope, or explicitly identify the path as out of scope.
T08 Missing response	Drop only the response after acceptance	Distinguish failure from unknown outcome; proceed to reconciliation.
T09 Retry / duplicate	Resend the same operation	Check required idempotency and detect duplicate effects.
T10 False completion	Executor reports success; external state is unchanged	Compare the claim with independent state and retain the mismatch.
T11 Missing / altered trail	Delete, reorder or replace an entry	Distinguish detectable tampering from omissions that cannot be detected.
T12 Semantic / long dependency	A legitimate new condition emerges mid-task	Measure reapproval/replanning without excessive blocking.

T05 tests the gap between checking and execution; T10 tests the gap between a tool response and the actual outcome. T11 measures tampering with recorded entries separately from capture of every required entry.

12 Metrics, costs and reproducibility

Do not compare on one success rate

Metric	Measurement / caution
False authorization / false denial	Report prohibited actions allowed and legitimate actions blocked separately. Do not arbitrarily count holds as successes or failures.
Task utility	Separate normal and attacked completion rates; retain blocked, timed-out and unsupported tasks in the denominators.
Assurance coverage	Identify controlled paths, independently observed paths and recheckable evidence items.
Effect consistency	Reconcile the verdict, the operation actually sent and external state. Count duplicate effects and unresolved outcomes.
Time / cost	Measure verdict, proof-completion and external-confirmation latency separately; report tails, not only medians.
Re-verifiability	Check verification in another environment, dependence on external APIs or secret keys, and which evidence is exportable.

What a reproducible test must fix

Record versions of the product, SDK, policy, model, prompt and data; test identities; timestamps; before/after states; randomness; timeouts; retries; and external dependencies. Within permitted access, pin commits, releases and test configurations rather than relying only on mutable webpages.

Define expected results from business requirements before running the product. Do not use only the provider’s own verdict as ground truth. For statistical components, report repetitions and uncertainty. Separate publicly reproducible work from portions requiring non-public access.

Principal trade-offs

Stricter blocking may reduce incidents but also stop legitimate work. Signatures, proofs and independent observations add latency, storage, key management, integration maintenance and operating procedures. Sending sensitive data to an evaluation service also creates privacy requirements for the evaluation itself.

Multiple guardrails are not necessarily independent. Shared faulty inputs or evaluation models can produce correlated errors despite more evidence. The comparison axes also help expose shared assumptions and allocation of responsibilities.

These are evaluation design and engineering analysis by this review, not unperformed product benchmarks presented as results.

CONCLUSIONS

13 Conclusions and unresolved boundaries

Several directions are becoming concrete in parallel

Pre-deployment model/agent evaluation, runtime controls, portable records and external outcome checks are developing under different actors and at different maturity levels. AWS execution boundaries, Microsoft intervention specifications, evaluation providers' deployment evidence and shared specifications illustrate distinct layers. [2, 5, 26, 28, 43]

The central question: how far can the assurance chain be closed?

A strong decision engine cannot stop an action on a bypass path. Tamper-evident records may still be incomplete. A correct proof may be inapplicable to a different target, purpose or input. An executor's success response does not necessarily establish the required external effect. These distinctions follow from the cross-case analysis.

Does the conclusion survive removal of individual companies?

Removing the commissioning company still leaves concrete controls outside the model and shared specifications. Removing concepts and prototypes still leaves generally available controls and SDK procedures. However, this sample cannot estimate market-wide adoption, and it does not prove that broad end-to-end assurance does not exist. [1, 13, 14, 43]

Revision conditions and the role of technical history

The finding that widespread, empirically demonstrated adoption is not established would be revised if multiple public studies of pinned production deployments tested bypasses, input authentication, external outcomes and independent re-verification under common conditions. Japan's technical history is presented alongside earlier spring-2025 overseas publications, without date-only superiority claims. [40, 9, 52, 41] / N1-N3

The international agenda supported by this report is:

With government policy formation and private verification/control technology working together, what should become independently checkable?

Remaining limits: no common-condition hands-on tests of every product, independent rebuilds of public proofs, long-term deployment measurements, authentication of original patent filings or claim-level priority assessment. No market-wide ranking or legal patent superiority is inferred without that evidence.

Failure to find evidence is not proof that a technology does not exist. The final review applies consistent criteria to positive functionality, unfavorable results and remaining unknowns.

PRIMARY SOURCES

Primary sources 1–7

Last accessed: 2026-10-15. Unknown dates remain unknown. Titles are original, shortened or descriptively translated. Provider announcements support their stated claims/procedures, not necessarily independent certification.

[1] Amazon Web Services | Policy in Amazon Bedrock AgentCore is now generally available

Date: 2026-03-03 | Type: developer/provider primary source

General-availability announcement, not the first-publication date of later additions.

<https://aws.amazon.com/about-aws/whats-new/2026/03/policy-amazon-bedrock-agentcore-generally-available/>

[2] Amazon Web Services | Amazon Bedrock AgentCore Developer Guide: Policy

Date: no update date stated; version at review | Type: developer/provider primary source

Pre-execution Gateway controls and policy evaluation.

<https://docs.aws.amazon.com/bedrock-agentcore/latest/devguide/policy.html>

[3] Amazon Web Services | Temporal policies in AgentCore

Date: no update date stated; version at review | Type: developer/provider primary source

Dogwood, session history, approvals, ENFORCE/LOG_ONLY and invalidation after temporal-policy changes.

<https://docs.aws.amazon.com/bedrock-agentcore/latest/devguide/policy-temporal.html>

[4] Microsoft | Introducing the Agent Governance Toolkit: open-source runtime security for AI agents

Date: 2026-04-02 | Type: developer/provider primary source

Positioning at the time of initial publication.

<https://opensource.microsoft.com/blog/2026/04/02/introducing-the-agent-governance-toolkit-open-source-runtime-security-for-ai-agents/>

[5] Microsoft | Introducing Agent Control Specification: Portable runtime governance for AI Agents

Date: 2026-06-02 | Type: developer/provider primary source

Eight intervention points; evidence, verdicts and host-side enforcement.

<https://commandline.microsoft.com/agent-control-specification-runtime-governance/>

[6] Microsoft | Agent Governance Toolkit — README

Version: pinned commit 0b83ab91508e9ceca159090240ddaaf9d4b2b88e | Type: public repository

Public Preview, govern, GovernanceDenied and upstream ACS dependencies. Final check uses the README at this commit.

<https://github.com/microsoft/agent-governance-toolkit/blob/0b83ab91508e9ceca159090240ddaaf9d4b2b88e/README.md>

[7] NVIDIA | OpenShell documentation: Overview / Architecture

Date: no update date stated; version at review | Type: developer/provider primary source

Agent execution environment and policy boundaries.

<https://docs.nvidia.com/openshell/about/overview>

PRIMARY SOURCES

Primary sources 8–14

Last accessed: 2026-10-15. Unknown dates remain unknown. Titles are original, shortened or descriptively translated. Provider announcements support their stated claims/procedures, not necessarily independent certification.

[8] NVIDIA | OpenShell documentation: Policy prover

Date: no update date stated; version at review | Type: developer/provider primary source

Standalone boundary-inclusion check. Unsupported/inconclusive results; distinctions from real kernel, DNS and task safety.

<https://docs.nvidia.com/openshell/how-it-works/policies/prover>

[9] CaMeL paper authors | Defeating Prompt Injections by Design (CaMeL), arXiv:2503.18813v2

Date: v1 2025-03-24 / reviewed v2 2025-06-24 | Type: developer-authored original research

Architecture, AgentDojo evaluation and limitations.

<https://arxiv.org/html/2503.18813v2>

[10] Invariant Labs | Introducing Invariant Guardrails

Date: 2025-04-17 | Type: developer/provider primary source

Policies and guardrails incorporating execution history.

<https://invariantlabs.ai/blog/guardrails>

[11] Snyk | Snyk Labs / Invariant Labs

Date: acquisition announced 2025-06-24 / page updated 2026-09-28 | Type: developer/provider primary source

Organizational and technical succession; not counted twice as separate companies.

<https://labs.snyk.io/resources/snyk-labs-invariant-labs/>

[12] Galileo | Agent Control: Overview

Date: no update date stated; version at review | Type: developer/provider primary source

LLM/tool input-output controls, policies and audit trails.

<https://docs.galileo.ai/concepts/agent-control/overview>

[13] Galileo | Initialize and configure Agent Control

Date: no update date stated; version at review | Type: developer/provider primary source

SDK initialization, @control instrumentation and configuration.

<https://docs.galileo.ai/how-to-guides/agent-control/initialize-and-configure-agent-control>

[14] Cisco | Securing AI Agents with Cisco AI Defense

Date: 2026-06-29 | Type: developer/provider primary source

agentsec.protect, SDK/MCP, API inspection and gateway approaches.

<https://blogs.cisco.com/ai/securing-ai-agents-with-cisco-ai-defense>

PRIMARY SOURCES

Primary sources 15–21

Last accessed: 2026-10-15. Unknown dates remain unknown. Titles are original, shortened or descriptively translated. Provider announcements support their stated claims/procedures, not necessarily independent certification.

[15] ICME Labs | PreFlight — Product

Date: no update date stated; version at review | Type: developer/provider primary source

Asynchronous verdict/proof timing, truth limits, mandatory hooks and unaudited foundation. Inconsistent FAQ descriptions of LLM roles also noted.

<https://www.icme.io/product>

[16] ICME Labs | PreFlight documentation

Date: no update date stated; version at review | Type: developer/provider primary source

Action inputs, rule checks, proofs and verification.

<https://docs.icme.io/documentation>

[17] Wyatt Benno et al. | Jolt Atlas: Verifiable Inference via Lookup Arguments in Zero Knowledge

Date: 2026-02-19 | Type: author-published original research

Foundation for verifiable inference, not a product-wide PreFlight performance test.

<https://arxiv.org/abs/2602.17452v1>

[18] EQTY Lab | Introducing Verifiable Runtime

Date: 2026-03-18 | Type: developer/provider primary source

Isolated execution infrastructure, authentication, policies and evidence architecture.

<https://www.eqtylab.io/blog/introducing-verifiable-runtime>

[19] Intel | Verifiable Compute: Enhancing the Accountability of Confidential AI

Date: 2024-12-18 | Type: primary announcement by technology partner

Foundational collaboration with EQTY, not an independent audit of the entire 2026 Runtime.

<https://community.intel.com/t5/Blogs/Products-and-Solutions/Security/Verifiable-Compute-Enhancing-the-Accountability-of-Confidential/post/1650286>

[20] Traefik Labs | Agentic Telemetry Is Not an Audit Trail: Why We Built the Sovereign Trust Plane

Date: 2026-09-14 | Type: developer/provider primary source

Delegation, external authorization and records; planned GA distinguished from completed availability.

<https://traefik.io/blog/why-we-built-the-sovereign-trust-plane>

[21] Ond / OSuite | Proof-Carrying Agent Actions — Research

Date: 2026-06-24 announcement | Type: developer/provider primary source

Research and enterprise trial program.

<https://ond.cc/research/pcaa>

PRIMARY SOURCES

Primary sources 22–28

Last accessed: 2026-10-15. Unknown dates remain unknown. Titles are original, shortened or descriptively translated. Provider announcements support their stated claims/procedures, not necessarily independent certification.

[22] PCAA paper authors | Proof-Carrying Agent Actions, arXiv:2606.04104v1

Date: 2026-06-02 / v1 | Type: developer-authored paper/experiment

Execution paths, review, receipts, evaluation and limitations, especially Sections 9.8, 10 and 11.

<https://arxiv.org/html/2606.04104v1>

[23] Weaveframe | Independent Agentic Assurance — White Paper

Date: 2026-09-27 | Type: developer/provider primary source

Partial prototype, independent outcome reconciliation and sections on unimplemented/excluded capabilities.

<https://www.weaveframe.ai/independent-agentic-assurance-white-paper>

[24] FlowX.AI | GAVEL — Research

Date: no publication date stated; version at review | Type: developer/provider primary source

Research architecture linking requirements, runtime observations and audit evidence. The linked paper's limited experiment is treated separately in [31].

<https://www.flowx.ai/research/gavel>

[25] The ProofGrid | Platform / Verification

Date: no publication date stated; version at review | Type: developer/provider primary source

Authority-action-outcome reconciliation. Retain the conceptual/illustrative qualification.

<https://www.theproofgrid.com/platform/verification/>

[26] Resaro | Approved Intelligence Platform

Date: no update date stated; version at review | Type: developer/provider primary source

Evaluation plans, operating conditions and evidence-based deployment decisions.

<https://resaro.ai/platform>

[27] AI Verify Foundation | Resaro: Leading the Charge in Contributing to Open-Source Development

Date: no publication date stated; version at review | Type: primary source from contribution-hosting organization

Confirmation of development contributions and evaluation activity, not an audit of every product function.

<https://aiverifyfoundation.sg/tools/user-stories/resaro-leading-the-charge-in-contributing-to-open-source-development/>

[28] LatticeFlow AI | LatticeFlow AI Launches a Single Platform to Control AI Risk in the Agentic World

Date: 2026-07-15 | Type: developer/provider primary source

AI Atlas and the link between governance frameworks and technical evaluation.

<https://latticeflow.ai/news/latticeflow-ai-launches-platform-to-control-ai-risk>

PRIMARY SOURCES

Primary sources 29–35

Last accessed: 2026-10-15. Unknown dates remain unknown. Titles are original, shortened or descriptively translated. Provider announcements support their stated claims/procedures, not necessarily independent certification.

[29] Citadel AI | Citadel Radar for Agents

Date: 2026-10-07 | Type: developer/provider primary source

Early Access Program.

<https://citadel-ai.com/news/2026/10/07/citadel-radar-for-agents/>

[30] Citadel AI | Announcement of planned AI governance solution provision to Sumitomo Mitsui Banking Corporation

Date: 2026-10-02 | Type: provider announcement of customer engagement

A planned-provision announcement for a different product and time.

<https://citadel-ai.com/news/2026/10/02/smbc-citadel-ai/>

[31] Bogdan Răduță / FlowX.AI | GAVEL: Governance That Runs Where the Agent Runs

Date: 2026-08 / v1.1 | Type: developer-authored technical paper/experiment

Cover; Section 6 (Table 1, PDF page 10); Section 8 (PDF page 13). One-rule, 165-trace synthetic evaluation and implementation/legal-interpretation limitations.

<https://www.flowx.ai/research/gavel.pdf>

[32] Imandra | Imandra Releases CodeLogician

Date: 2025-03-26 | Type: developer/provider primary source

Formal analysis of specifications and code; availability audience at announcement checked.

<https://www.imandra.ai/articles/imandra-releases-codelogician>

[33] Hongyu Lin et al. | Imandra CodeLogician: Neuro-Symbolic Reasoning for Precise Analysis of Software Logic

Date: v1 2026-01-17 / reviewed v2 2026-02-06 | Type: developer-authored original research

Formalization–implementation correspondence and evaluation setup.

<https://arxiv.org/html/2601.11840v2>

[34] Alinia | Seny model — official announcement

Date: 2026-10-01 | Type: developer/provider primary source

Model evaluating conformity with policies, legal requirements and related criteria.

<https://www.alinia.ai/resources/seny-model>

[35] Alinia | Benchmark Data for Legal and Tax Advice

Date: 2026-10-01 | Type: provider benchmark description

Expert-authored English examples machine-translated into eight languages; 5% native-speaker spot review. Not full-item review or independent evaluation.

<https://www.alinia.ai/resources/benchmark-data-for-legal-and-tax-advice>

PRIMARY SOURCES

Primary sources 36–42

Last accessed: 2026-10-15. Unknown dates remain unknown. Titles are original, shortened or descriptively translated. Provider announcements support their stated claims/procedures, not necessarily independent certification.

[36] AI Assurance, Inc. | Civic AI Answer Adoption Assurance — README

Date: accessed 2026-10-15; main branch | Type: public PoC; README reviewed
PASS/HOLD/REJECT, file-output control, dummy data and provisional collaboration language.
<https://github.com/GhostDriftTheory/civic-ai-answer-adoption-assurance/blob/main/README.md>

[37] AI Assurance, Inc. | Responsibility OS × EU AI Act — README

Date: referenced commit dated 2026-10-03 | Type: public formal model; README reviewed
Selected articles, formal propositions and exclusions; pinned commit.
<https://github.com/GhostDriftTheory/responsibility-os-eu-ai-act-mapping/blob/bef1531b839b8fa55b6ab86786d538305b8858f2/README.md>

[38] GitHub Actions / GhostDriftTheory | Lean verification — run 37097244762

Date: 2026-10-03 | Type: developer-managed CI execution record; records checked in this review
API records confirm both Build mapping and warnings-as-errors verification succeeded for the same commit.
<https://github.com/GhostDriftTheory/responsibility-os-eu-ai-act-mapping/actions/runs/37097244762>

[39] AgentDyn paper authors | AgentDyn, arXiv:2602.03117v1

Date: 2026-02-03 / v1 | Type: original experiment external to the compared developers
External evaluation of dynamic tasks, attacks and multiple defense approaches.
<https://arxiv.org/html/2602.03117v1>

[40] AI Assurance, Inc. | How Japan’s “Responsibility OS” emerged: the convergence of October 2025 finite-closure execution control and November ADIC re-verification

Date: published 2026-07-17; updated 2026-08-30 | Type: developer’s public technical history
Developer-published filing dates, numbers and technical history, not official filing receipts.
<https://zenn.dev/ghostdrift/articles/049534ae1a288e>

[41] WIPO | PCT International Search and Preliminary Examination Guidelines, PCT/GL/ISPE/14

Version: PCT/GL/ISPE/14, issued 2025-12-12; effective 2026-01-01 | Type: official examination guidelines
Sections 6.05–6.10 (printed pages 55–56): combination disclosure and priority assessment from the description, claims and drawings as a whole.
<https://www.wipo.int/documents/d/pct-system/docs-en-texts-ispe.pdf>

[42] European Patent Office | Guidelines for Examination, F-VI 2.2: The same invention

Date: 2026 edition | Type: patent-office examination guidelines
Subject matter directly and unambiguously derivable from the priority document as a whole.
https://www.epo.org/en/legal/guidelines-epc/2026/f_vi_2_2.html

PRIMARY SOURCES

Primary sources 43–49

Last accessed: 2026-10-15. Unknown dates remain unknown. Titles are original, shortened or descriptively translated. Provider announcements support their stated claims/procedures, not necessarily independent certification.

[43] AARM / Cloud Security Alliance | Autonomous Action Runtime Management — System Category Specification v1.0

Date: dated 2026-02; version at review | Type: primary shared private-sector specification

Core R1–R6, Extended R7–R9 and treatment of conformance.

<https://aarm.dev/spec>

[44] Cloud Security Alliance | Autonomous Action Runtime Management (AARM) Working Group

Date: no update date stated; version at review | Type: primary source from receiving organization

Working-group activities and institutional status.

<https://cloudsecurityalliance.org/research/working-groups/autonomous-action-runtime-management-aarm>

[45] Vanta | Vanta donates AARM to the Cloud Security Alliance

Date: 2026-04-30 | Type: primary announcement by transferring organization

Announcement transferring a company-origin specification into shared stewardship.

<https://www.vanta.com/resources/vanta-donates-aarm-to-csa>

[46] Google Cloud | Powering AI commerce with the new Agent Payments Protocol (AP2)

Date: 2025-09-16 | Type: developer/provider primary source

Binding mandates, user intent and payment instructions.

<https://cloud.google.com/blog/products/ai-machine-learning/announcing-agents-to-payments-ap2-protocol>

[47] Google | Agent Payments Protocol joins FIDO Alliance

Date: 2026-04-28 | Type: primary announcement by transferring organization

Transfer to shared stewardship, not assurance of post-payment fulfillment.

<https://blog.google/products-and-platforms/platforms/google-pay/agent-payments-protocol-fido-alliance/>

[48] MLCommons | The key to trustworthy AI evaluation is secrecy by design

Date: 2026-08-27 | Type: primary source from pilot participant

Evaluation pilot with AVERI, OpenMined and other participants.

<https://mlcommons.org/2026/08/double-blind-reliability-evaluation/>

[49] ElevenLabs | AIUC announcement

Date: 2026-02-12 | Type: primary announcement by user company

Certification/insurance announcement by the user company.

<https://elevenlabs.io/blog/aiuc-announcement>

PRIMARY SOURCES

Primary sources 50–56

Last accessed: 2026-10-15. Unknown dates remain unknown. Titles are original, shortened or descriptively translated. Provider announcements support their stated claims/procedures, not necessarily independent certification.

[50] Chaucer | Chaucer and Armilla AI launch Vanguard AI coordinated insurance structure

Date: 2026-02-10 | Type: primary announcement by insurance-side party

Insurance-side confirmation of collaboration; coverage terms require case-specific review.

<https://www.chaucergroup.com/news/press-release-chaucer-and-armilla-ai-launch-vanguard-ai-coordinated-insurance-structure>

[51] AgentScope | AgentScope Runtime — Introduction (Chinese documentation)

Date: no update date stated; version at review | Type: public developer documentation; Chinese

Sandbox, execution and observation infrastructure.

<https://runtime.agentscope.io/zh/intro.html>

[52] Haoyu Wang, Christopher M. Poskitt, Jun Sun | AgentSpec: Customizable Runtime Enforcement for Safe and Reliable LLM Agents

Date: 2025-03-24 / v1 | Type: author-published original research

Runtime constraints through triggers, predicates and interventions.

<https://arxiv.org/html/2503.18666v1>

[53] Hanzo AI | Formal Proofs Ship

Date: 2026-03-26 | Type: developer/provider primary source

Claims concerning formal proofs for agent infrastructure.

<https://hanzo.blog/blog/2026-03-26-formal-proofs-ship/>

[54] Hanzo AI | Hanzo Formal Proofs — README

Date: accessed 2026-10-15; main branch | Type: public repository; README reviewed

Current AI Chain, model provenance, DID, TEE and settlement inventory. Correspondence with every blog claim remains unconfirmed.

<https://github.com/hanzoai/proofs/blob/main/README.md>

[55] Holistic AI | EndLayer — Guardian Agents for AI

Date: no publication date stated; version at review | Type: developer/provider primary source

Sensor, Decoder and Guardian; endpoint observation/control and versioned records.

<https://www.holisticai.com/endlayer>

[56] Fiddler AI | Prevent AI Agents from Leaking Sensitive Data with Fiddler Guardrails and LiteLLM

Date: 2026-09-14 | Type: developer/provider primary source

Gateway integration example, pre_call/post_call and fail-closed configuration.

<https://www.fiddler.ai/blog/prevent-ai-agent-data-leakage-guardrails>

PRIMARY SOURCES

Primary sources 57–59

Last accessed: 2026-10-15. Unknown dates remain unknown. Titles are original, shortened or descriptively translated. Provider announcements support their stated claims/procedures, not necessarily independent certification.

[57] IBM | watsonx Orchestrate — Agent control plane

Date: no publication date stated; version at review | Type: developer/provider primary source

Integration of pre-deployment evaluation, policies, identity, audit and observation.

<https://www.ibm.com/products/watsonx-orchestrate/agent-control-plane>

[58] Arthur AI | Agent Security & Governance

Date: no publication date stated; version at review | Type: developer/provider primary source

Detection, monitoring, verdict API and application-side enforcement responsibilities.

<https://www.arthur.ai/discover-and-govern-agents>

[59] M. Kaptein, V.-J. Khan, A. Podstavnychy | Runtime Governance for AI Agents: Policies on Paths, arXiv:2603.16586v1

Date: 2026-03-17 | Type: original research; abstract-only supplementary review

Abstract reviewed: a framework involving partial execution paths, actions and organizational state. No full-text implementation verification.

<https://arxiv.org/abs/2603.16586v1>

SUPPLEMENTARY SOURCES & REVIEW LOG

Supplementary sources and review record

Non-public retained drafts: supplementary checks only

These drafts were used solely for limited checks of technical history. They have no public URL and cannot be fully re-examined by a reader from this report alone. Their submission dates, correspondence to application numbers and identity with the filed originals have not been established.

ID	Document examined	Passages / use
N1	Non-public draft N1	[0028]–[0031], [0052], [0057]–[0058], [0070]–[0071]. Cross-check of the published execution-control and audit account.
N2	Non-public draft N2	[0021]–[0028], among others. Cross-check of published communication, suspension and rejoining descriptions.
N3	Non-public draft N3	[0007]–[0013], among others. ADIC recomputation/re-verification cross-check. Correspondence with the cited application remains unverified.

Public-artifact records actually checked

Microsoft: the README at commit 0b83ab91508e9ceca159090240ddaaf9d4b2b88e was rechecked for Public Preview status, govern procedures and upstream ACS dependencies. Full-source execution and code audit were not performed. [6]

Responsibility OS / EU mapping: the README at commit bef1531b839b8fa55b6ab86786d538305b8858f2 was reviewed. GitHub API records for Lean verification run 37097244762 confirmed success of both Build mapping and Verify source with warnings as errors. This is distinct from successful Pages deployment. [37, 38]

Hanzo: the current README inventory was compared with the blog, but proof files, theorems and executable versions corresponding to every blog claim were not identified. This does not mean a source build was observed to fail. [53, 54]

Edition history

This fourth edition source-checks the third edition’s primary documents, dates, URLs and principal claims. It corrects AWS history invalidation, Microsoft’s upstream specification, inconsistent ICME descriptions and unimplemented Weaveframe outputs. The old BSI source, whose substantive text could not be retrieved again, was removed as evidence; [31] now identifies the limited GAVEL v1.1 experiment. Mutable documents reflect the retrieval date; a long-term archive of every page was not created.

Supplementary research: the abstract of Policies on Paths describes policy evaluation over partial execution paths, prospective actions and organizational state. It supplements the history/state analysis, not claims of commercial deployment or performance. [59]