Authority at Runtime

What AI is allowed to say, do, and rely on — under what conditions

IDENTITY AND ACCESS CONTROL ARE NOT ENOUGH FOR HIGH-STAKES AI.

Audience: executive sponsors, product leaders, platform/security teams, legal/compliance stakeholders.

LSAS Stack by Inference Stack LLC
Enterprise AI Execution Authority™

ADDRESS
Inference Stack LLC
5365 W. 73rd Pl.
Arvada, CO 80003

CONTACT
+1 720.334.3838
www.lsas-stack.com
www.lsas-stack.com/contact

Central idea

Traditional governance focuses on who is allowed to use a system. High-stakes AI governance must go further: it must define what the system is allowed to say, what it is allowed to do, and what it is allowed to rely on before trust is granted.

Authority becomes meaningful only when expressed in runtime conditions, not just policy prose.

Authority at runtime

SAY
Narrative output, advice, summaries, client-facing statements

DO
Actions, tool calls, API side effects, routing and release

RELY ON
Sources, citations, retrieval results, structured records

Three authority planes that matter at runtime

Why role-based access is necessary but incomplete

A role assignment can determine who may access an assistant, an agent, or a workflow surface. It does not determine whether the output is grounded, whether the action is safe, or whether the sources are acceptable for a client-facing or regulated use case. That is why execution authority requires more than RBAC — it requires runtime predicates.

Authority at Runtime

Authority dimension	Question	Representative controls
SAY	What claims, recommendations, summaries, or explanations may the system release?	Evidence thresholds, citation validation, output constraints, audience-aware transformation, abstention rules.
DO	What actions, tool calls, API requests, or state changes may the system trigger?	Action gating, workflow-scoped permissions, step-up review, explicit override confirmation, allow/deny policies.
RELY ON	Which sources, records, retrieval results, or structured systems may the system treat as acceptable evidence?	Approved source registries, retrieval policy, source ranking, freshness rules, validator requirements.

Runtime predicates: the conditions that determine authority

- Workflow criticality: a marketing draft does not warrant the same conditions as a clinical discharge summary or legal advisory note.
- Risk tier: the runtime must know whether the interaction is low, standard, regulated, or locked down.
- Data class: PHI, PCI-like patterns, secrets, and internal sensitive data alter what the system may release or retain.
- Evidence sufficiency: the system needs to know whether approved evidence exists, whether validators passed, and whether a human attestation is required.
- Actor and boundary context: the same model behavior may be acceptable inside a draft workspace and unacceptable at a client-facing, patient-facing, or operational boundary.

A practical policy grammar

The most useful governance language is concrete. Instead of saying ‘use AI responsibly,’ mature programs encode explicit policy statements such as:

Representative policy statements

The system may summarize protected content for internal clinical use, but it may not produce a non-clinical outbound summary unless minimum-necessary transformation controls pass.

The system may draft a legal memo, but client-facing release requires supported citations from approved sources or explicit reviewer approval.

The system may classify transaction anomalies, but it may not echo card-like values, secrets, or raw sensitive fields into outbound output.

Authority at Runtime

What proof should exist after execution

Proof artifact	What it answers
Decision outcome	What the runtime allowed, constrained, blocked, abstained from, or escalated.
Finding set	Which rules or validators fired and what risk indicators were present.
Evidence status	Whether approved evidence was present, missing, stale, or insufficient.
Remediation action	What the system did next: redact, narrow, retry, escalate, or stop.
Authority chain	Whether a human exception was requested, approved, confirmed, or expired.
Version linkage	Which policy pack or configuration version produced the decision.

Applied examples

Clinical

A clinician may be allowed to use AI for internal note summarization. That does not automatically authorize the system to produce a patient-safe public summary from the same note. At runtime, the system should detect the audience shift, inspect PHI exposure risk, apply minimum-necessary controls, and either transform or withhold the output.

Legal

A legal team may allow AI drafting for internal research. That does not authorize unsupported citations to survive into client work product. Authority at runtime means that the system can draft, but release depends on evidence sufficiency and explicit review conditions.

Fintech

A fraud or payments workflow may authorize classification, routing, or anomaly explanation. It should not authorize sensitive values, secrets, or PCI-like data to appear in prompts, logs, or outbound narratives. Authority therefore depends on redaction and action gating, not merely user identity.

Authority at Runtime

What LSAS brings to this model

LSAS aligns well to an authority-at-runtime framing because it already exposes the primitives required for it: risk classification, claim inspection, evidence checks, policy tiers, deterministic outcomes, governed overrides, and decision telemetry. That makes it possible to express authority as executable behavior rather than a slide deck.

Without runtime authority

- Policy expectations stay abstract.
- Review burden grows because every team must infer what happened from fragmented evidence.
- Exceptions happen informally and are hard to reconstruct.

With runtime authority

- Policy is encoded into enforceable execution paths.
- Reviewers see why a decision happened and what proof existed.
- Exceptions become explicit, bounded, and auditable.

Selected references

1. LSAS Stack website — home. Application-layer runtime positioning, pipeline, risk domains, telemetry, and deployment options. https://www.lsas-stack.com/

2. LSAS Stack — Sandbox / Guided demo. Scenario list, decision envelope behavior, governed overrides, and audit metadata framing. https://www.lsas-stack.com/playground

3. LSAS Stack — Compliance posture. Controls mapping language, deployment/residency posture, data minimization, auditability, and operational hooks. https://www.lsas-stack.com/compliance-posture

4. McKinsey — The AI reckoning: How boards can evolve (2025). Board posture, ownership, and governance policy guidance for AI oversight. https://www.mckinsey.com/capabilities/mckinsey-technology/our-insights/the-ai-reckoning-how-boards-can-evolve

5. American Medical Association — AI usage among doctors doubles as confidence grows (2026). Physician AI usage, current use cases, and concerns around privacy and patient relationships. https://www.ama-assn.org/press-center/ama-press-releases/ama-ai-usage-among-doctors-doubles-confidence-technology-grows

6. FTI Consulting / Relativity — The General Counsel Report, AI adoption in legal departments (2026). Corporate legal department AI adoption, roadmap maturity, and common task areas. https://www.fticonsulting.com/about/newsroom/press-releases/ai-adoption-in-corporate-legal-departments-doubles-according-to-the-general-counsel-report

7. World Economic Forum — Artificial Intelligence in Financial Services (2025). Responsible-AI governance patterns and adoption of AI governance frameworks in financial services. https://reports.weforum.org/docs/WEF_Artificial_Intelligence_in_Financial_Services_2025.pdf

8. Inference Stack — Contact. Inference Stack contact page; based in Denver, Colorado; serving clients worldwide. https://www.inference-stack.com/contact