ISSN: 2754-6659

Journal of Artificial Intelligence & Cloud Computing

SCIENTIFIC Research and Community

Mini Review

Open Access

AI Proof Layer: An Outcome-Assurance Architecture for Reliable, Safe, and Auditable AI Models

Patrick Casimir

PhD, AI Proof Layer Founder, USA

ABSTRACT

Large Language Models (LLMs) deliver exceptional generative capability, but they share a structural limitation: they cannot prove that a given output is correct, grounded in authoritative evidence, or compliant with policy. As a result, hallucinations, unverifiable claims, and policy violations persist, especially in high-risk settings such as finance, healthcare, legal reasoning, and enterprise operations.

This paper introduces **AI Proof Layer**, an external *outcome-assurance* layer that operates independently of the model. AI Proof Layer evaluates model outputs against explicit, measurable guarantees ("claims"), enforces **ALLOW/BLOCK** decisions, and generates immutable **Evidence Packs** suitable for audits, incident reviews, and regulatory reporting, without modifying or constraining the underlying model architecture.

By separating *generation* (the model) from *permission* (AI Proof Layer), the system converts generative AI models from a probabilistic text generator into a **certifiable decision system**. We present the conceptual framework, reference architecture, decision contract, example workflows, and compliance mappings that enable organizations to reduce hallucinations and establish traceable accountability across the AI lifecycle.

***Corresponding author**

Patrick Casimir, PhD, AI Proof Layer Founder, USA.

Received: March 21, 2026; **Accepted:** March 23, 2026; **Published:** April 04, 2026

Keywords: AI Governance, Auditability, Hallucination Mitigation, Outcome Assurance, Evidence Artifacts, Compliance-By-Design

Introduction

LLMs are increasingly embedded into products that draft contracts, summarize policies, recommend medical actions, support customer service, and power autonomous agents. In these settings, the primary risk is not that the model generates an imperfect sentence; it is that the organization cannot reliably determine whether an output is safe to use, and cannot demonstrate that determination to auditors, regulators, or customers.

Traditional mitigations, prompt engineering, RLHF, safety filters, and retrieval-augmented generation (RAG), improve behaviour but do not provide enforceable guarantees. Most controls are heuristic, non-versioned, and difficult to audit. When failures occur, post-mortems frequently lack a portable artifact that captures: the user query, the model output, the evidence consulted, the decision rule applied, and the rationale for allowing or blocking the output.

AI Proof Layer addresses these gaps through a simple systems principle: models generate; platforms decide. Proof Layer is the decision layer that decides whether a model output can be used in a given context, and produces an evidence artifact that makes this decision inspectable and repeatable.

The Hallucination and Compliance Problem

Operational Definition of Hallucination

For enterprise deployment, hallucination should be defined operationally as: a confident assertion that lacks adequate support in authoritative evidence for the current context. This definition is systems-friendly: it can be measured, thresholded, and audited.

Why Hallucinations Persist Even with RAG and Guardrails

RAG reduces hallucination by providing relevant documents, but it does not guarantee that the model uses them correctly. Guardrails may detect obvious unsafe content, but they rarely provide a quantitative, versioned decision record. Self-evaluation (e.g., 'check your answer') is circular: it asks the model to validate itself.

High-risk contexts require stronger properties

- Explicit outcome guarantees
- Deterministic enforcement
- Portable evidence artifacts and
- Continuous regression tracking over time.

Architectural Overview

High-level System Architecture

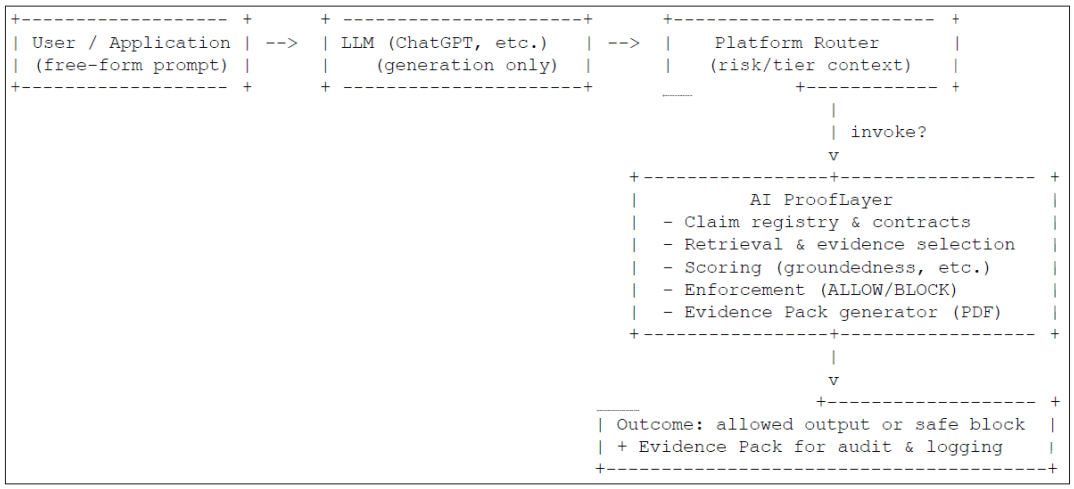


Figure 1: AI Proof Layer sits between model output and real-world impact. The model never self-certifies; the platform invokes Proof Layer based on policy and context.

Design goals

Goal	What it means in practice
Outcome guarantees	Every decision maps to an explicit claim with measurable thresholds.
Deterministic enforcement	Same input + same evidence + same contract → same decision.
Portable evidence	Each decision yields an immutable Evidence Pack suitable for audits.
Model-agnostic integration	Works across vendors and model versions without retraining.
Continuous regression tracking	Detects drift by comparing decisions and scores over time.

Enforcement Decision Contract

In *Enforce* mode, AI Proof Layer applies a deterministic and binary decision rule that governs whether an AI-generated output is allowed to propagate beyond the platform router.

AI Proof Layer computes a measurable *groundedness* score that quantifies the degree to which an output is supported by verifiable evidence aligned with a declared outcome claim (e.g., financial accuracy, clinical grounding, policy completeness).

A domain- or claim-specific *threshold* defines the minimum acceptable level of grounding required for safe output propagation.

The enforcement rule is defined as follows:

- If **groundedness** < **threshold**, the output is **BLOCKED**
- If **groundedness** ≥ **threshold**, the output is **ALLOWED**

This rule is evaluated after model generation but before output propagation, ensuring that outputs failing to meet explicit outcome guarantees cannot reach downstream users or systems.

The enforcement decision is invariant and auditable given identical inputs. For each decision, Proof Layer generates an immutable evidence artifact that records the user query, computed scores, threshold, decision rationale, and referenced evidence.

In contrast, *Observe* mode executes the same evaluation logic without gating output propagation, allowing organizations to assess risk, measure policy compliance, and detect regressions prior to enforcement.

This explicit separation between observation and enforcement enables gradual adoption of outcome assurance while preserving full auditability and control.

Core Abstractions

Claims and Contracts

A **claim** is a named, measurable guarantee attached to an AI interaction. Examples include: **finance accuracy, policy completeness, clinical grounding, and contract consistency**.

Each claim defines

- Required evidence sources
- Scoring metrics
- Enforcement thresholds and
- A safe fallback behaviour.

```
Claim Contract (example: finance_accuracy)
------------------------------------------
Evidence sources: enterprise_policy_corpus, legal_guidelines
Metrics: groundedness, semantic_similarity, citation_ratio
Thresholds: groundedness >= 0.75 AND citation_ratio >= 0.20
Enforcement: if below thresholds and output asserts facts -> BLOCK
Safe fallback: allow only if response explicitly states uncertainty + routes to authority (e.g., Lega
Artifact: Evidence Pack v1 (PDF) stored with run_id and timestamps
```

Figure 2: A claim contract defines what "good" means, how it is measured, and what happens when it is not met

Evidence Packs

Every evaluation produces an immutable Evidence Pack containing the user query, decision, thresholds, scores, evidence excerpts, and an answer preview. This transforms ad-hoc prompt outputs into auditable records.

Decision Logic: How Proof Layer Stops Hallucinations

Key Insight: Safe Uncertainty should Pass

A common failure mode of simplistic gating is to block any low-confidence output.

Proof Layer distinguishes between

- Low evidence with confident assertions (unsafe), and
- Low evidence with explicit uncertainty (safe).

Therefore, the policy should be: **block confident speculation; allow evidence-aware uncertainty.**

```
Decision rules (simplified)
---------------------------
Let G = groundedness score
Let T = claim threshold
Let A = assertion_strength (confident vs uncertain)

IF mode = observe:
    ALLOW (log artifact)
ELSE IF G >= T:
    ALLOW
ELSE IF G < T AND A = confident_assertion:
    BLOCK (INSUFFICIENT_GROUNDING)
ELSE IF G < T AND A = explicit_uncertainty:
    ALLOW (with safe escalation language)
```

Figure 3: AI Proof Layer blocks hallucinations without suppressing safe escalation behaviour

Worked Example (Policy QA)

User Query: What exceptions apply to enterprise refunds under international contracts?

Assume the retrieved enterprise refund policy does not mention international contract exceptions. A hallucinating model may invent exceptions. AI Proof Layer forces the system to either (a) demonstrate grounding, or (b) state uncertainty and escalate.

Scenario	Model behavior	Proof Layer outcome
Hallucination	Lists fictional exceptions not present in policy.	BLOCK + Evidence Pack citing Missing support
Safe uncertainty	States policy does not specify exceptions; routes	ALLOW + Evidence Pack

API Contract and Integration

Minimal Evaluate Contract

```
POST /evaluate
{
  "model": "gpt",
  "claim": "finance_accuracy",
  "mode": "enforce",              // observe | enforce
  "threshold": 0.75,
  "user_query": "....",
  "raw_answer": "...."            // optional, if generated upstream
}

Response
{
  "run_id": "pl-<id>",
  "decision": "ALLOW|BLOCK",
  "blocked_reason": "INSUFFICIENT_GROUNDING|null",
  "scores": {
     "groundedness": 0.406,
     "semantic_similarity": 0.565,
     "citation_ratio": 0.167
  },
  "evidence_pdf": "artifacts/pl-<id>.pdf"
}
```

Figure 4: AI Proof Layer is invoked by the platform router. It returns a decision plus an immutable Evidence Pack location

When to invoke AI ProofLayer (Router Rules)

The model should not decide when oversight is needed. The invoking policy lives within the platform router and is auditable. Common invocation policies include: domain-based routing (finance/health/legal), product tier (Certified Mode), and actionability (if output triggers transactions or customer-facing commitments).

Safety and Regulatory Compliance

AI Proof Layer supports compliance-by-design by producing traceable decisions and artifacts. In regulated contexts, the key requirement is not perfect accuracy; it is demonstrable control: the organization must show how it constrained risk, logged decisions, and responded to uncertainty.

Compliance need	How ProofLayer supports it
Traceability	Run IDs, timestamps, evidence sources, and decision rules captured per interaction.
Audit artifacts	Evidence Pack PDFs can be attached to tickets, audits, and incident reports.
Risk-based controls	Router invokes ProofLayer only for high-risk domains or actions.
Change management	Claim contracts can be versioned; regressions tracked over time.
Accountability	Separates generator (model) from enforcer (ProofLayer) for clearer responsibility.

Regression Over Time (Drift Detection)

AI Proof Layer can record scores and decisions over time for the same claim and query templates. This enables drift detection: if groundedness or citation ratio degrades after a model update or retrieval change, AI Proof Layer flags regression and produces comparable artifacts.

```
Regression Example (same query over time)
-----------------------------------------
Date        Model     Groundedness   Decision
2026-02-01  gpt-X     0.81           ALLOW
2026-03-10  gpt-Y     0.62           BLOCK (INSUFFICIENT_GROUNDING)
Action: investigate retrieval corpus, model update, or threshold tuning; attach Evidence Packs
```

Figure 5: Proof Layer supports operational monitoring by converting quality into a time-series of decisions and artifacts.

Limitations and Future Work

This paper describes a reference architecture and artifact contract. Production deployments should extend the system with

- Stronger claim registries and policy DSLs
- Cryptographic signing of Evidence Packs
- Tenant-aware evidence controls
- Calibrated metrics and adversarial testing and
- Integration with governance frameworks and incident management

Conclusion

Hallucinations persist because LLMs cannot prove correctness. AI Proof Layer addresses this by adding an external, enforceable outcome-assurance layer that evaluates outputs against explicit claims, blocks unsafe assertions, and generates immutable evidence artifacts. This separation of generation from permission enables safer deployments, stronger governance, and regulatory-grade auditability—without sacrificing model innovation.

Copyright: ©2026 Patrick Casimir. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.