The Last Mile to Agentic Value

Closer Than We Think

Rajesh Iyer
iyer70@gmail.com

Abstract

Enterprise agentic AI adoption is stalling not because the technology fails, but because practitioners cannot produce evidence that satisfies examiners. Debug traces tell engineering what executed. Compliance replay tells regulators why a decision was made, what alternatives were rejected, and under what authority the system acted. This paper defines compliance replay as the missing evidentiary primitive for agentic systems—a standardized, auditable representation for third parties, analogous to what financial statements are for firms. We specify the minimal five-element construct (Who, What, State, Branch Logic, Counterfactual), illustrate the transformation with concrete banking and insurance examples, and call for practitioner alignment on this standard. Compliance replay cannot be retrofitted. Every agentic system designed today without the required infrastructure is accumulating technical debt that will block deployment later. The gap is bridgeable, but only through deliberate design.

1 Introduction

The agentic AI movement is drowning in demos that cannot ship. Gartner predicts that over 40% of agentic AI projects will be cancelled by the end of 2027, citing "escalating costs, unclear business value, or inadequate risk controls" [1]. Senior Director Analyst Anushree Verma notes that "most agentic AI projects right now are early stage experiments or proof of concepts that are mostly driven by hype and are often misapplied."

The conventional diagnosis points to technology immaturity, integration complexity, or organizational resistance. This paper argues the root cause is simpler: practitioners cannot produce evidence that satisfies the people who must approve deployment.

Scope. For banks and insurers operating under SR 11-7 and NAIC frameworks, compliance replay is mandatory—examiners will eventually require it explicitly, and this paper gets ahead of that inevitability. For general enterprises deploying agentic systems in consequential contexts (hiring, pricing, resource allocation), compliance replay is optional but valuable: it accelerates stakeholder approval and provides defensibility if decisions are later challenged. For low-stakes automation (content generation, internal tooling), the full primitive may be unnecessary.

Debug traces—the standard output of agent orchestration frameworks—tell engineering teams what executed. They answer: which tool was called, what parameters were passed, what response was returned. This is necessary for development. It is insufficient for deployment.

Compliance replay answers a different question: why was this decision made and not another? What state was visible at decision time? What constraints governed the action space? What alternatives were evaluated and rejected? Under what authority did the system act?

We introduce compliance replay as a new compliance abstraction—a standardized, auditable representation for third parties, analogous to what financial statements are for firms. It sits alongside logs (what happened), models (how decisions are made), and controls (what is permitted) as the fourth layer: *why this decision was made*.

Critically, compliance replay cannot be retrofitted. The State element requires content-addressable storage infrastructure. The Counterfactual element requires agents designed with structured deliberation. Every agentic system being built today without these capabilities is accumulating technical debt that will block deployment when examiners ask questions the debug traces cannot answer. The 9-month remediation cycles we observe in practice are not bugs—they are the predictable consequence of designing for engineering observability without designing for regulatory evidence.

We present the specification, illustrate the transformation with concrete examples, and call for practitioner alignment so that those who can build production agentic systems are no longer drowned out by those who cannot.

2 The Evidence Problem

Enterprise agentic AI operates in contexts where decisions must be explained to stakeholders who did not build the system and do not trust it by default. These stakeholders include:

Examiners (OCC, NAIC, internal audit) who ask: "Why wasn't this escalated, declined, or flagged?" Their job is to find the decision that should not have been made autonomously. They need the counterfactual—what alternatives existed and why they were rejected.

Compliance officers who must attest: "Was this action within the constraint envelope at decision time?" They cannot sign off on a system that cannot prove what constraints were active when a decision was made.

Model risk managers who validate: "Did this com-

ponent version behave as documented under these inputs?" SR 11-7 requires them to perform effective challenge—critical analysis by objective parties. They cannot challenge what they cannot reconstruct.

Current observability tools—LangSmith, Weights & Biases, custom telemetry—capture execution traces optimized for debugging. They record what happened. They do not record the evidentiary chain required to prove what happened was appropriate.

Tool Category	What It Proves
Observability	Execution truth (what ran)
Compliance Replay	Decision legitimacy (why appropriate)

Table 1: Observability vs. compliance replay

The result is a deployment bottleneck. Engineering completes the agent. Product validates the capability. Legal and compliance cannot sign off because the evidence artifact does not exist. This is not hypothetical: 61% of BFSI executives cite regulatory risk as the primary barrier to AI deployment [4]. The 40% cancellation rate Gartner predicts is not a technology problem—it is an evidence problem.

2.1 The Anti-Pattern: Explainability as Evidence

A common misconception holds that explainability solves the evidence problem. It does not.

Explainability explains models. Compliance replay explains decisions.

XAI techniques (SHAP, LIME, attention visualization) answer "why did the model output X?" They do not answer "why did the system take action Y instead of action Z, and was that within policy?"

The distinction is critical:

- **Explainability**: Post-hoc rationalization of model internals
- **Compliance replay**: Point-in-time evidence of decision basis, alternatives rejected, and authority exercised

Examiners do not want explanations. They want reconstructible evidence.

2.2 A Failure Case

Consider a real pattern from model risk examinations:

A bank deployed an LLM-assisted credit decisioning agent. When examiners asked "why was customer X approved for $50K when similar customers were declined?", the team produced: application logs showing the approval event, LLM prompt/response pairs from the decision, and a post-hoc SHAP analysis of feature importance.

The examiner's response: "None of this tells me what the system knew at decision time, what alternatives it considered, or whether the approval was within delegated authority."

The finding: *Insufficient evidence of effective challenge capability.*

The deployment was paused pending remediation. The remediation took 9 months—not because the agent didn't work, but because the evidence infrastructure had to be built from scratch.

3 The Compliance Replay Primitive

Compliance replay is the minimal evidence artifact that transforms a debug trace into stakeholder-acceptable proof. It comprises five elements:

Element	Definition	Failure Mode
Who	Identity & version of components	Cannot attribute
What	Action & business effect	Cannot assess harm
State	Hash of inputs at decision time	Cannot prove known
Branch	Constraints & criteria	Cannot verify policy
Counter	Alternatives & rejections	Cannot answer "why not?"

Table 2: Five-element compliance replay specification

Each element maps to a specific examiner question. Table 3 shows the persona-to-element correspondence.

Persona	Question	Elements
Examiner	Why not escalated?	Counter, Branch
Compliance	Within policy?	State, Branch
Model Risk	Behaved as doc'd?	Who, What, State

Table 3: Persona to element mapping

3.1 Design-Time Requirements

Compliance replay cannot be retrofitted. Two elements impose design-time constraints:

State capture requires content-addressable storage. The State element assumes infrastructure for hash-to-

content resolution—the ability to retrieve actual data from a hash years after decision time. Organizations without existing CAS infrastructure face a platform investment, not a configuration change.

Counterfactual capture requires structured deliberation. The Counterfactual element requires agents to explicitly enumerate alternatives considered and reasons for rejection. Agents must be designed to deliberate visibly, not retrofitted to explain post-hoc. The pattern:

```
Before deciding, enumerate alternatives:
- Option A: [description] →
[selected/rejected]
- Option B: [description] →
[selected/rejected]
- Rationale: [why selected option
prevailed]
```

Without this structure, the Counterfactual element cannot be populated from execution traces alone.

3.2 Human Override Capture

Compliance replay applies equally to human decisions within agentic workflows. When a human overrides, modifies, or approves an agent recommendation, that decision also requires a compliance replay record.

The Who element captures the human actor (role, authorization level). The Counterfactual element captures what the agent recommended and why the human diverged:

```
COUNTERFACTUAL:
- Agent recommendation: APPROVE $15,000
- Human override: REDUCED to $10,000
Reason: "Customer called with hardship
claim;
applying discretionary reduction per policy
4.2.1"
- Full decline: NOT WARRANTED
Reason: Human confirmed eligibility
criteria met
```

This is critical for examiner defensibility. Examiners will ask not only "why did the agent decide X?" but also "why did the human disagree with the agent?" Both questions require the same evidentiary structure.

4 Debug Trace vs. Compliance Replay

The transformation from debug trace to compliance replay is not a matter of adding fields. It is a reframing of purpose. The same execution event yields two different artifacts for two different audiences.

4.1 Banking Example: Credit Limit Increase

A customer requests a credit limit increase from $8,000 to $15,000. The agent approves $12,000.

Debug Trace (engineering artifact):

```
[2026-01-14T09:23:41Z] AgentRouter v2.3.1
→ CreditPolicyAgent invoked
[2026-01-14T09:23:41Z] Tool:
get_customer_profile
→ customer_id: 847291
→ response: {income: 78000, tenure:
4.2yr,
utilization: 0.72, dti: 0.31}
[2026-01-14T09:23:42Z] Tool:
get_bureau_data
→ response: {score: 721, derogs: 0,
inq_6mo: 2}
[2026-01-14T09:23:42Z] LLM: claude-3-opus
→ prompt_tokens: 1847
→ completion_tokens: 234
[2026-01-14T09:23:43Z] Decision: APPROVE
→ approved_limit: 12000
→ confidence: 0.87
```

Compliance Replay (examiner artifact):

```
COMPLIANCE REPLAY: Credit Decision
CR-2026-847291
================================================
WHO:
- Orchestrator: AgentRouter v2.3.1
(artifact: sha256: 8f3a2b...)
- Policy Agent: CreditPolicyAgent v1.8.0
(artifact: sha256: 2c4e7f...)
- Reasoning: claude-3-opus-20240229
(temperature: 0.1, seed: 847291)

WHAT:
- Action: Credit limit increase
- Request: $8,000 → $15,000
- Approved: $12,000
- Business Effect: $4,000 incremental
exposure

STATE (at decision time):
- State Hash: sha256:9d8c7b6a5e4f3d2c1b0a...
- Customer: income=$78K, tenure=4.2yr,
utilization=72%, DTI=31%
- Bureau: score=721, derogatory=0
- Policy Version: credit-policy-2024-q4-v3

BRANCH LOGIC:
- Constraint Envelope:
* Max increase: min(2x current, $25K)
* DTI threshold: ≤0.40
* Bureau floor: ≥680
* Utilization cap for full approval:
≤0.65
- Decision Path:
* Customer eligible: YES
* Full amount blocked: util 72% > 65%
* Partial approved per high-util policy

COUNTERFACTUAL:
- Full $15,000: REJECTED
```

```
Reason: Utilization 72% > 65% threshold
- Decline: NOT WARRANTED
Reason: All hard constraints satisfied;
score 721 > 680, DTI 31% < 40%
- Human escalation: NOT TRIGGERED
Reason: Within $25K autonomous ceiling
```

The debug trace tells engineering the agent worked. The compliance replay tells the examiner why this customer got $12,000 and not $15,000, why it was not declined, and why no human was required.

4.2 Insurance Example: Auto Claim Triage

A policyholder submits a collision claim with photos and a police report. The agent assigns severity "moderate" and routes to standard adjuster queue.

Compliance Replay (examiner artifact):

```
COMPLIANCE REPLAY: Claim Triage
CT-2026-00892
==========================================
WHO:
- Triage Agent: ClaimsTriageAgent v3.1.0
(artifact: sha256: 4a5b6c...)
- Vision Model: damage-assess-v2.1
(artifact: sha256: 7d8e9f...)
- Document Parser: police-report-extractor
v1.2
(artifact: sha256: 1a2b3c...)

WHAT:
- Action: Claim routed to standard queue
- Severity Assignment: Moderate
- Business Effect: 3-5 day SLA, no
expedite

STATE (at decision time):
- State Hash: sha256:3f4e5d6c7b8a9...
- Images: 2 (front_damage.jpg,
rear_quarter.jpg)
- Police Report: Fault=0%, No injuries
- Policy: Comprehensive, $500 deductible
- Claim History: 1 claim in 5 years (2021,
$340)

BRANCH LOGIC:
- Triage Rules (NAIC-compliant v2024.1):
* Total loss threshold: est > 70% ACV
* Expedite triggers: injury, fatality,
>$25K estimate, fraud flag
* SIU referral: fraud score > 0.7
- Decision Path:
* Severity: Moderate (est $4,200-$6,800)
* No expedite triggers present
* Fraud score: 0.12 (below threshold)

COUNTERFACTUAL:
- Expedite Queue: NOT WARRANTED
Reason: No injury, est below $25K
- SIU Referral: NOT WARRANTED
Reason: Fraud score 0.12 < 0.7 threshold
- Total Loss: NOT APPLICABLE
Reason: Est $4,200-$6,800 « 70% ACV
- Human Review: NOT TRIGGERED
Reason: All automated criteria satisfied
```

4.3 Multi-Agent Example

In orchestrated multi-agent systems, compliance replay captures at decision boundaries, not agent boundaries. Consider a workflow where Agent A proposes and Agent B approves.

Compliance Replay (at Agent B's decision):

```
COMPLIANCE REPLAY: Policy Approval
PA-2026-00341
Trace ID: tr-7f8e9d0c-1a2b-3c4d
==========================================
WHO:
- Recommender: UnderwritingAgent v2.1.0
- Approver: ApprovalAuthority v1.3.2

STATE (at decision time):
- State Hash: sha256:4d5e6f7a8b9c...
- Upstream Recommendation:
trace_ref: tr-7f8e9d0c-1a2b-3c4d/node-A
recommended_premium: $2,400

COUNTERFACTUAL:
- Accept recommendation: SELECTED
- Override to $2,800: REJECTED
Reason: Risk score within tolerance
```

The trace ID correlation allows examiners to reconstruct the full chain. Each agent's compliance replay is self-contained; the correlation enables end-to-end audit.

5 Regulatory Context

For general enterprises, compliance replay accelerates stakeholder approval. For banks and insurers, it is a regulatory requirement.

SR 11-7 (Federal Reserve, OCC, 2011) establishes that "a guiding principle for managing model risk is 'effective challenge' of models, that is, critical analysis by objective, informed parties who can identify model limitations and assumptions and produce appropriate changes" [5]. Effective challenge requires the ability to reconstruct the decision basis—precisely what compliance replay provides.

NAIC Model Bulletin on AI (adopted December 2023, now implemented in 24 states [6]) establishes that "decisions made by Insurers are not inaccurate, arbitrary, capricious, or unfairly discriminatory." The bulletin explicitly contemplates examination: "An Insurer can expect to be asked about its development, deployment, and use of AI Systems... and outcomes resulting from them."

The question for BFSI institutions is not whether to implement compliance replay, but how quickly they can

close the gap.

6 Implementation

Debug trace and compliance replay are parallel outputs from the same execution event—not sequential transformations.

Element	Hook Point	Implementation
Who	Agent init	Manifests, configs
What	Decision exit	Payload + effect
State	Pre-decision	SHA-256 + CAS ref
Branch	Policy load	Constraint envelope
Counter	Post-decision	Rejection rationale

Table 4: Implementation mapping

The compliance layer intercepts at decision boundaries and enriches the trace with the five elements. The primitive is framework-agnostic.

7 Limitations

This paper proposes a specification intended to drive practitioner alignment, not an established standard.

The specification assumes decision-based architectures where discrete choice points can be identified. Continuous or emergent behaviors may require extensions.

Compliance replay cannot be retrofitted. Historical decisions remain evidentiarily incomplete.

The appendix storage projections assume 60-80% deduplication. High-cardinality customer data deduplicates poorly—the cost delta between 60% and 30% deduplication is approximately 2x at 7-year retention scales.

8 The Path Forward

The agentic movement's credibility problem is not technical. Gartner projects 33% of enterprise software will include agentic AI by 2028 [3]. The question is whether adoption will be blocked at the compliance gate.

Compliance replay is the primitive that separates demo from deployment. It is the *Decision Evidence Layer*—the missing abstraction between agent frameworks and regulatory examination. It sits alongside:

- **Logs**: What happened (engineering)
- **Models**: How decisions are made (model risk)
- **Controls**: What is permitted (compliance)
- **Compliance Replay**: Why this decision (evidence)

The call to action: align on this primitive. Build it into frameworks. Specify it in procurement. Let practitioners who can produce compliance replay be distinguished from those who cannot.

The last mile is shorter than it appears.

Acknowledgements

This paper was developed with GenAI collaboration (ChatGPT, Claude) through an interview-driven refinement process.

References

[1] Gartner. (2025). *Gartner Predicts Over 40% of Agentic AI Projects Will Be Canceled by End of 2027*.

[2] MIT Sloan Management Review & BCG. (2024). *Achieving Individual and Organizational Value with AI*.

[3] Gartner. (2024). *Predicts 2025: AI Agents Emerge as Digital Colleagues*.

[4] PYMNTS Intelligence. (2024). *Financial Services Executives See Higher Stakes in GenAI Deployments*.

[5] Board of Governors of the Federal Reserve System & OCC. (2011). *SR 11-7: Guidance on Model Risk Management*.

[6] NAIC. (2023). *Model Bulletin on the Use of AI Systems by Insurers*.

[7] Iyer, R. (2026). *Forensic Telemetry for Regulated Agentic Systems*. corpXiv.

Appendix: Implementation Specification

Engineering Detail for the Compliance Replay Primitive

This appendix provides engineering specifications for the compliance replay primitive: JSON schemas, a LangGraph reference implementation, performance benchmarks, and deployment guidance.

A Schema Definitions

A.1 Master Record

```
{
  "record_id": "uuid",
  "timestamp": "ISO-8601",
  "trace_id": "string",
  "who": {"$ref": "WhoElement"},
  "what": {"$ref": "WhatElement"},
  "state": {"$ref": "StateElement"},
  "branch_logic": {"$ref": "BranchElement"},
  "counterfactual": {"$ref": "CfElement"}
}
```

A.2 WHO Element

```
{
  "components": [{
    "name": "CreditPolicyAgent",
    "version": "1.8.0",
    "artifact_ref": {
      "uri": "s3://models/1.8.0.tar.gz",
      "hash": "sha256:2c4e7f..."
    },
    "role": "policy_agent",
    "config": {"temperature": 0.1}
  }],
  "decision_authority": {
    "type": "autonomous",
    "ceiling": "$25K"
  }
}
```

A.3 WHAT Element

```
{
  "action_type": "credit_limit_increase",
  "action_details": {
    "customer_id": "847291",
    "current_limit": 8000,
    "approved_limit": 12000
  },
  "business_effect": {
    "description": "$4K exposure",
    "impact_category": "financial",
    "reversibility": "partial"
  }
}
```

A.4 STATE Element

```
{
  "state_hash": "sha256:9d8c7b...",
  "capture_ts": "2026-01-14T09:23:42Z",
  "input_refs": [{
    "source": "customer_profile",
    "hash": "sha256:abc123...",
    "storage_ref": "s3://state/abc",
    "ttl_days": 2555
  }],
  "policy_refs": [{
    "name": "credit-policy",
    "version": "2024-q4-v3"
  }]
}
```

Requires content-addressable storage.

A.5 BRANCH_LOGIC Element

```
{
  "policy_version": "credit-2024-q4-v3",
  "constraints": [{
    "id": "min-bureau-score",
    "threshold": {"op":">=","val":680},
    "actual": 721,
    "eval": "passed"
  }],
  "decision_path": [{
    "step": "eligibility",
    "outcome": "eligible"
  }]
}
```

A.6 COUNTERFACTUAL Element

```
{
  "alternatives": [{
    "alt": "Full approval ($15K)",
    "outcome": "rejected",
    "reason": "Util 72% > 65%",
    "blocking": ["util-cap"]
  }],
  "escalation": {"warranted": false}
}
```

B Reference Implementation

The compliance layer intercepts at decision boundaries, capturing data in parallel with debug traces.

B.1 Architecture

```
LangGraph: [A]->[B]->[Decision]->[D]
                      |
          Compliance Capture Layer
            - WHO: component versions
            - STATE: hash inputs
            - BRANCH: constraints
            - COUNTER: alternatives
            - WHAT: action
                 |              |
           [LangSmith]    [Compliance DB]
```

B.2 Core Data Classes

```
@dataclass
class ComponentRef:
    name: str
    version: str
    artifact_hash: str
    role: str
    config: Dict[str, Any]

@dataclass
class InputRef:
    source: str
    hash: str
    storage_ref: str
    ttl_days: int = 2555

@dataclass
class Constraint:
    id: str
    threshold: Dict[str, Any]
    actual: Any
    evaluation: str

@dataclass
class Alternative:
    alternative: str
    outcome: str
    reason: str
    blocking: List[str]
```

B.3 State Hash (Merkle Tree)

```
def compute_state_hash(inputs):
    hashes = [
        sha256(json.dumps(i, sort_keys=True)
               .encode()).hexdigest()
        for i in inputs
    ]
    while len(hashes) > 1:
        if len(hashes) % 2:
            hashes.append(hashes[-1])
        hashes = [
            sha256((hashes[i] + hashes[i+1])
                   .encode()).hexdigest()
            for i in range(0, len(hashes), 2)
        ]
    return f"sha256:{hashes[0]}"
```

B.4 Counterfactual Prompt

```
PROMPT = """Before deciding, enumerate:
<alternatives>
- Alt: [description]
- Outcome: [selected/rejected]
- Reason: [explanation]
</alternatives>"""
```

B.5 Callback Handler

```
class Handler(BaseCallbackHandler):
    def on_chain_start(self, ser, inp, **kw):
        self._trace = kw.get("run_id")
        self._inputs = []

    def on_tool_end(self, out, **kw):
        self._inputs.append(json.loads(out))

    def on_llm_end(self, resp, **kw):
        self._llm = [
            x.text for g in resp.generations
            for x in g
        ]

    def on_chain_end(self, out, **kw):
        if kw.get("name") in self.nodes:
            self._capture(out)
```

C Performance

C.1 Latency

Operation	P50	P99
State hash	2ms	8ms
S3 storage	15ms	45ms
Counterfactual parse	5ms	12ms
Record write	8ms	25ms
Total	**30ms**	**90ms**

Table 3: Latency by operation

C.2 Storage (1M decisions/mo)

Component	Per Rec	7-Year
Compliance record	8 KB	672 GB
State snapshots*	15 KB	500 GB
Index overhead	2 KB	168 GB

*Table 4: Storage (*60-80% deduplication)*

C.3 Cost (AWS)

```
S3 Standard:      $30/mo
PostgreSQL RDS:   $200/mo
Total:            $240/mo (1M decisions)
```

D Testing

D.1 Reconstruction Test

```
def test_reconstruction(record):
    for ref in record.state.input_refs:
        content = store.get(ref.storage_ref)
        assert sha256(content).hex() == \
               ref.hash[7:]
    result = graph.invoke(rebuild(record))
    assert result["decision"] == \
           record.what.action_type
```

D.2 Examiner Simulation

```
def test_examiner(record):
    alts = record.counterfactual.alternatives
    decline = [a for a in alts if
               "decline" in a.alt.lower()]
    assert decline[0].reason
    sources = {r.source for r in
               record.state.input_refs}
    assert "customer_profile" in sources
```

E Deployment Checklist

```
Infrastructure:
[ ] Content-addressable storage
[ ] Compliance DB + indexes
[ ] 7-year retention policy
[ ] Monitoring dashboards

Agent Integration:
[ ] Decision nodes annotated
[ ] Callback handler integrated
[ ] Counterfactual prompts added
[ ] Component registry populated

Validation:
[ ] Reconstruction tests passing
[ ] Examiner simulation passing
[ ] Performance within SLA (<100ms P99)
```

F Quick Reference

Element	Examiner Question	Hook
WHO	What decided?	chain_start
WHAT	What action?	chain_end
STATE	What known?	tool_end
BRANCH	In policy?	policy
COUNTER	Why not X?	LLM parse

Table 5: Element to implementation mapping