Governing Claude in the Enterprise: The Missing Data Access Layer

Anthropic's Compliance API tells you what Claude said. It cannot tell you what Claude touched. This paper explains the gap, and how to close it at the data layer without waiting for anyone's roadmap.

EXECUTIVE SUMMARY

The Question Your Auditor Will Ask

Claude is now embedded across the enterprise: Claude Code in engineering, Claude connectors in business teams, Claude-powered agents in production workflows. Anthropic has responded to enterprise governance demands with a **Compliance API**, giving security and compliance vendors programmatic access to usage data, conversation records, and policy hooks. A growing ecosystem of vendors now integrates with it.

This is genuine progress. And it still misses the question that determines regulatory liability:

> **"What did Claude actually access in our datastores?"**
> Not what was typed into a prompt. Not what the model answered. What rows, records, and fields left your systems of record, and under what justification.

The Compliance API operates at the *conversation layer*. But when Claude does real work, it acts through **tool calls**: MCP (Model Context Protocol) servers and internal APIs that query databases, vector stores, document repositories, and SaaS systems. The data returned by those calls is summarized, transformed, and compressed before it ever appears in a conversation record. A conversation log showing *"I've summarized the customer accounts you asked about"* is not evidence of which 40,000 rows were read to produce it.

What this paper covers

- **The visibility gap:** why conversation-layer compliance tooling, including Anthropic's Compliance API and the vendors built on it, structurally cannot see data access.
- **The data-layer answer:** how SmartVerify's egress control plane sits in front of your MCP servers and datastores, inspecting every Claude tool call in under 50 milliseconds.
- **How to do it:** a five-step deployment path from audit-only visibility to full enforcement and behavioral intelligence, including why enterprise-hosted MCP gateways are the deployment pattern that makes this work.

EXECUTIVE SUMMARY, CONTINUED

Key Takeaways

- Claude is not a special case. It is a new, very high-volume agent class flowing through the same data paths your other AI agents use. Governing it requires no Anthropic partnership and no application changes.
- Anthropic's Compliance API and SmartVerify are **complementary**: one governs the conversation layer, the other governs the data layer. Regulators and auditors will ask about both.
- Evidence-grade audit of agent data access is becoming a regulatory expectation, not a differentiator. NIST's AI Agent Standards Initiative and the NSA/CISA joint guidance on agentic AI adoption (April 2026) both point at runtime data controls.

PART ONE

The Visibility Gap

Enterprise Claude deployments have three planes. The **conversation plane** is where prompts and completions live; this is what the Compliance API exposes. The **orchestration plane** is where Claude decides which tools to invoke. The **data plane** is where those tools actually execute against your systems. Compliance evidence lives in the third plane; today's tooling watches the first.

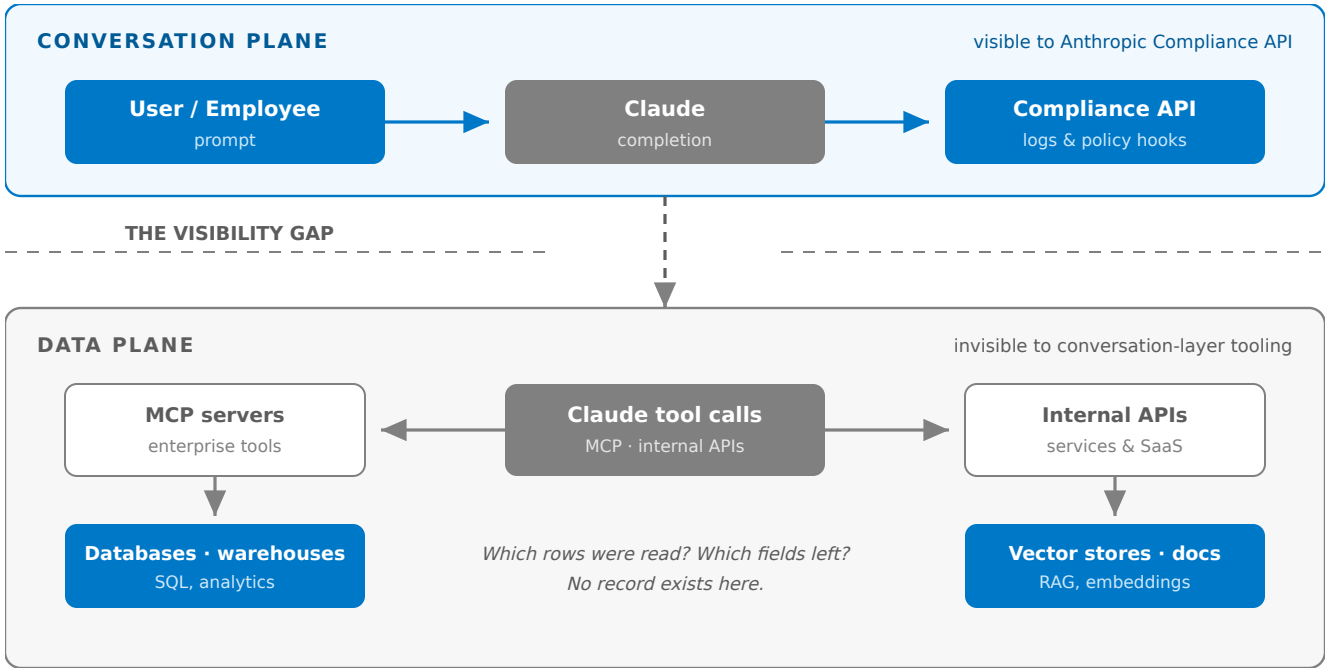

Figure 1: The Compliance API governs the conversation plane. Claude's actual data access happens below the line, where no conversation-layer record exists.

What Each Layer Can Answer

This is not a criticism of Anthropic's design. A model vendor *cannot* see inside your datastores; that boundary is exactly where their visibility should end. But it means conversation-layer records are structurally incomplete as compliance evidence: they capture intent and summary, not access. The table below maps the questions a security or compliance team must answer against what each layer can actually see.

Compliance question	**Anthropic Compliance API** (conversation layer)	**SmartVerify** (data layer)
Who used Claude, when, and in which workspace?	**Yes**	**Partial: per agent identity**
What was said in the conversation?	**Yes**	**No, by design**
Did an employee paste sensitive text into a prompt?	**Yes (with DLP vendors)**	**Out of scope**
Which tools did Claude invoke, with which arguments?	**Summary only**	**Yes: full JSON-RPC parse**
Which rows / records / fields were returned from datastores?	**No**	**Yes: payload-level**
Was sensitive data masked or blocked before reaching Claude?	**No enforcement at data path**	**Yes: inline, sub-50ms**
Does this access map to NIST AI RMF / state AI acts?	**Usage-level only**	**Yes: per-query mapping**
Did deleted user data stay deleted across RAG and caches?	**No**	**Yes: DROP enforcement**
Is this agent's data access drifting from its baseline?	**No**	**Yes: behavioral baseline**

> **The complementary picture:** run the Compliance API (and its vendor ecosystem) for conversation governance, and SmartVerify for data access governance. Together they close the loop an auditor actually walks: *who asked, what the agent did, what data left*.

Why this matters now

Regulatory momentum has shifted from "govern which AI you deploy" to "prove what your AI did with data at runtime," and current guidance converges on one expectation: **an immutable, per-query record of agent data access, enforced at the data path.** Conversation logs do not satisfy it.

PART TWO

The Data-Layer Answer

SmartVerify is an **egress control plane**: a sub-50ms Envoy-based proxy deployed in your own VPC (the Hybrid Gateway model), sitting directly in front of your MCP servers, internal APIs, and datastores. To SmartVerify, Claude is not a special integration; it is a new agent class flowing through the same data paths as every other agent. Every Claude tool call receives the full treatment:

- **MCP-aware inspection:** the proxy parses MCP's JSON-RPC structure, so policy sees *which tool, which arguments, what returned*, not opaque HTTP.
- **Intent classification:** each request is scored against the stated purpose of the agent, generating dataset-level policy on the fly.
- **Inline enforcement:** sensitive fields are masked, tokenized, or blocked on the wire before they reach Claude; attribute-based access control applies per MCP server and API endpoint.
- **DROP enforcement:** deletion requests propagate across RAG pipelines, vector stores, and caches, and stay enforced when Claude queries them.
- **Behavioral baselining:** each Claude deployment gets its own baseline; silent drift and slow exfiltration patterns surface even when each individual query looks legitimate.

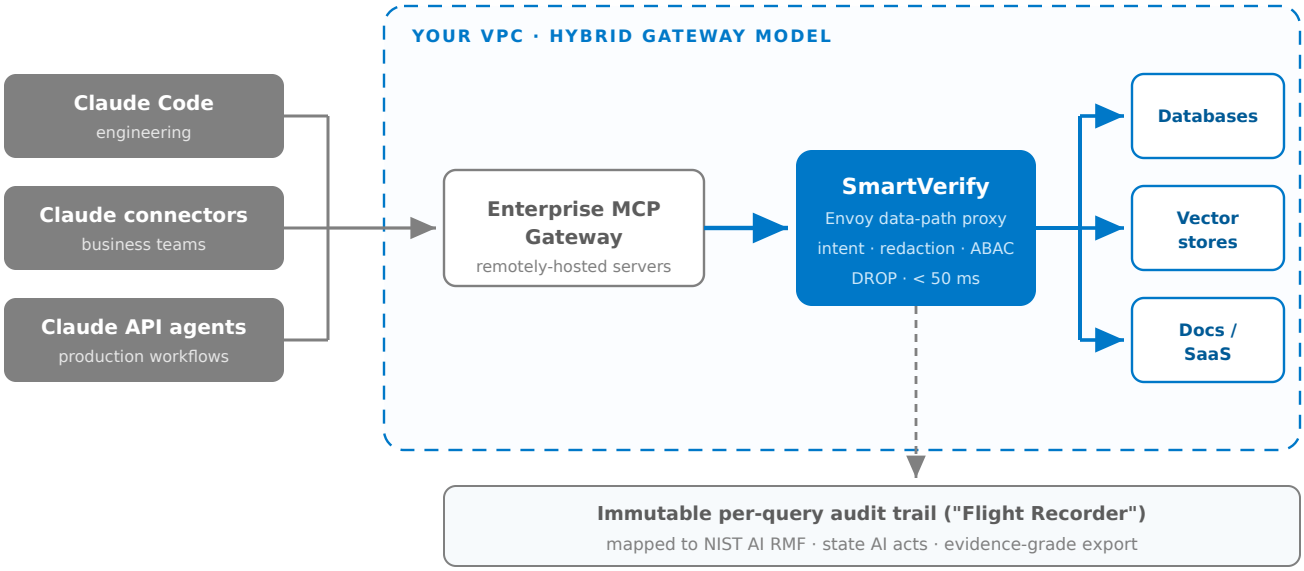

Figure 2: Every Claude surface flows through the enterprise MCP gateway; SmartVerify inspects and enforces on the data path inside your VPC.

Because enforcement happens in your infrastructure, on your traffic, this requires **no Anthropic partnership, no Claude configuration changes, and no application code changes**. It works on any customer's Claude traffic today, and on the next model vendor's traffic tomorrow.

Seeing Inside the Tool Call

The technical crux is making the data path **MCP-protocol-aware**. MCP tool calls are JSON-RPC messages; to a conventional network proxy they are opaque HTTP POST bodies. SmartVerify's wire-speed filter parses the protocol itself, so classification and policy operate on the semantic triple that matters: *which tool, which arguments, what returned.*

```
// What a conventional proxy sees:
POST /mcp HTTP/1.1        <- opaque body, allow or deny

// What SmartVerify sees, and records:
{
  "method": "tools/call",
  "params": {
    "name": "query_customer_db",        <- which tool
    "arguments": {
      "sql": "SELECT name, ssn, balance FROM accounts …"  <- what was asked
    }
  }
}
// Response payload inspected before returning to Claude:
rows_returned: 40,212   sensitive_fields: [ssn -> MASKED, balance -> ALLOWED]
intent_score: 0.91 (matches stated purpose: quarterly reconciliation)
audit_record: sv-2026-07-20-8842-c1 (immutable)
```

Figure 3: MCP-aware inspection turns an opaque tool call into a policy decision and an evidence-grade audit record.

An honest coverage note: local stdio MCP servers

One deployment pattern is outside any network proxy's reach: MCP servers running *locally on developer laptops via stdio*. Those calls never touch the network, so no data-path control (SmartVerify's or anyone else's) can see them.

> **The prerequisite that is also best practice:** route Claude's enterprise data access through **remotely-hosted, enterprise-managed MCP gateways**. This is the pattern enterprises are already converging on, and the one CISOs already want for credential centralization, version control, and kill-switch capability. Local stdio servers should be limited to non-sensitive developer tooling by policy. Centralized MCP hosting is not a workaround for SmartVerify; it is the security architecture that makes agent access governable at all.

This gives security teams a clean, defensible position: *sensitive datastores are reachable only through the gateway; the gateway is fronted by the egress control plane; everything else is out of policy by construction.*

PART THREE

How to Do It: Five Steps

1 Centralize MCP hosting

Stand up (or designate) an enterprise MCP gateway for every server that touches sensitive systems. Inventory existing Claude Code and connector configurations; migrate stdio servers that access production data to remote hosting. Publish a policy: sensitive datastores are reachable only via the gateway.

2 Deploy the SmartVerify data plane in your VPC

The Envoy-based proxy deploys in front of your MCP gateway, internal APIs, and datastore endpoints. It is Kubernetes-native, adds under 50ms of latency, and requires zero application code changes. Your data never leaves your environment (Hybrid Gateway model).

3 Run in Audit mode first

Start observation-only. Within days you get the Flight Recorder: an immutable, per-query record of every Claude tool call (tool, arguments, rows and fields returned, sensitivity labels) automatically mapped to NIST AI RMF sub-categories and applicable state AI acts. This alone typically unblocks Legal/Risk approval for broader Claude rollouts.

4 Turn on Enforcement where evidence justifies it

Promote observed patterns into policy: mask or tokenize sensitive fields on the wire, apply attribute-based access control per MCP server and API endpoint, block out-of-intent queries, and enforce deletion propagation (DROP) across RAG pipelines and vector stores. Enforcement is incremental: one datastore, one agent class at a time.

5 Add Intelligence: baselines and drift detection

Each Claude deployment gets a behavioral baseline. Intent scoring adapts to how autonomous agents actually behave, catching slow-drift exfiltration patterns that span hours and multiple agents: the cases where every individual query looks fine and only the trajectory is wrong.

Pairing with the Anthropic Compliance API

Teams already integrating conversation-layer vendors should keep them. The join key is identity and time: SmartVerify's per-query audit records correlate with Compliance API conversation records to produce the full evidentiary chain, from *who prompted* to *what Claude decided to do* to *what data actually moved.* That chain is what an auditor, a regulator, or an incident responder actually needs.

CONCLUSION

Claude Is the Forcing Function, Not the Exception

Claude's enterprise footprint made the question urgent, but the question is bigger than any one vendor: **when an authorized agent touches your data, can you prove what it accessed, and stop what it shouldn't?** Every conversation-layer tool, including Anthropic's Compliance API and the vendor ecosystem around it, stops at the boundary of your infrastructure. That is precisely where the liability begins.

SmartVerify closes the gap at the only place it can be closed: the data path. One deployment governs Claude Code, Claude connectors, Claude API agents, and the next agent platform your business adopts, with no new integration work.

> **Secure the Data. Liberate the AI.** Start in Audit mode, get evidence-grade visibility in days, and let your security team say yes to the Claude roadmap instead of slowing it down.

About SmartVerify

SmartVerify is the first Egress Control Plane for the Agentic Era, built by the team that built AWS security and compliance services from the ground up. The platform decouples data security from application logic, giving organizations an immutable audit trail and real-time enforcement at the data layer, where agent risk actually lives.

Want to learn more? Contact us at **smartverify.ai/contact**