WHITE PAPER

# Governing the Data Path

A security architecture for frontier AI in the enterprise — because the model was never the risk you had to own.

**Towfeeque Aalam**

Principal Cloud Security Architect · Ex-AWS

Frontier AI · Governance · Security Architecture

**towfeeque.com**

## Abstract

*Enterprise AI programs spend their governance energy in the wrong place. Model selection gets the committee; the data flowing into and out of the model gets an integration ticket. Yet nearly every AI incident an enterprise will actually experience — leakage of confidential data into prompts, over-privileged agents acting on systems of record, ungoverned outputs landing in customer channels — occurs on the data path, not inside the model. This paper proposes a boundary-first security architecture for adopting frontier AI: an AI gateway as a control point, identity and entitlement propagation into retrieval, blast-radius design for agentic systems, and an assurance model that treats AI as a new class of workload rather than a new class of magic.*

## 1. The misdirected governance debate

Ask an enterprise AI steering committee what they are evaluating and you will hear model names and benchmark scores. Ask what data the pilot is wired to, under whose identity it retrieves documents, and what systems its outputs can touch, and the room goes quiet. The frontier model — trained, aligned and operated by a vendor with more safety engineering than any enterprise will field — is usually the most governed component in the stack. The integration around it is usually the least.

This paper takes a deliberately unfashionable position: for enterprise adoption, model choice is a commodity decision revisited quarterly. The data path — what goes in, what comes out, who can see it, and what it is allowed to act on — is the architecture decision that persists across every model you will ever use.

## 2. Four boundaries that matter

Every enterprise AI system, from a chat assistant to an autonomous agent, has the same four boundaries. Each needs an owner, a control, and evidence.

- Ingress — what enters the context window: prompts, retrieved documents, tool results. The control question: can confidential or regulated data reach a model context that its human requester is not entitled to see?
- Egress — what leaves: completions, generated artefacts, tool invocations. The control question: what channels can output reach without a human decision, and what damage can a wrong output do there?
- Entitlement — whose identity governs retrieval and action. The control question: does the AI system act with the caller's permissions, or with a service account that quietly aggregates everyone's?
- Action — what the system can change in the world: APIs called, records written, messages sent. The control question: what is the blast radius of the worst plausible action, and is it survivable?

Most AI incidents trace to exactly one of these boundaries being implicit rather than designed. Making all four explicit is the whole game; the rest of this paper is implementation detail.

## 3. The AI gateway pattern

Enterprises learned long ago not to let every application call the internet directly; they will learn the same about model APIs. An AI gateway — a policy enforcement point through which all model traffic flows — is the single highest-leverage control an adopting enterprise can build.

### What the gateway enforces

- Data-loss prevention on ingress: classification-aware inspection of prompts and retrieved context, so a document marked board-confidential does not silently become training material for an answer to an intern.
- Output handling on egress: routing model responses through content policy, watermarking or human review appropriate to the destination channel.
- Model allow-listing and version pinning: which teams may use which models, in which regions, for which data classifications — enforced, not documented.
- Complete audit: every prompt, context payload and completion logged with the caller's identity, retained to the same standard as any other regulated system of record.

The gateway also solves the procurement problem quietly: when the better model ships next quarter — and it will — you re-point the gateway, rerun your evaluations, and no application changes.

## 4. Retrieval is an authorisation problem

RAG turned every document store into an API, and most enterprises wired that API to a service account with read access to everything. The result is a machine that launders entitlements: the user asks a question they could not answer with their own permissions, and the pipeline answers it for them.

The principle is simple to state and demanding to build: retrieval must execute under the effective entitlements of the requesting human, evaluated at query time. Index-time filtering is not enough — permissions change, people move teams, documents get reclassified. The retrieval layer must consult the source system's ACLs, or a faithfully synchronised authorisation index, on every query.

Treat any RAG deployment that cannot answer 'who was allowed to see the documents behind this answer?' as an open finding, not a phase-two enhancement.

## 5. Agentic systems and blast radius

Agents change the risk class. A model that writes text produces bad drafts; a model that calls APIs produces state changes. Governance built for chat does not transfer.

- Give every agent its own workload identity — never a shared service account — so its actions are attributable and individually revocable.
- Scope tools to the minimum verb set: an agent that files tickets needs create, not delete; an agent that reads a database does not need the write endpoint 'for later'.
- Set blast-radius budgets per agent: the maximum records writable, spend committable, or messages sendable per task — enforced by the platform, not by the prompt.
- Require human confirmation on irreversible actions, and make 'irreversible' a property tagged on the tool, not a judgement left to the model.
- Log the full reasoning-action trace. When an agent does something surprising, the trace is the difference between an incident review and a shrug.

The discipline mirrors what we already do for human operators: least privilege, separation of duties, and an audit trail. The novelty of agents is not the controls they need; it is how easy it is to forget to apply them to something that talks like a colleague.

## 6. Assurance: AI as a workload class

The instinct to create a parallel AI-governance universe — new committees, new registers, new everything — produces shelfware and slows adoption without reducing risk. The alternative is to treat AI as a new workload class inside the existing security operating model: threat-model it, pen-test it (prompt injection is input validation's new face), monitor it, and fold its evidence into the same compliance machinery as everything else.

Two additions are genuinely new and worth building: evaluation pipelines that regression-test model behaviour against your specific misuse cases on every model or prompt change, and an inventory of AI systems with their four boundary definitions on record. Both are extensions of engineering practice, not replacements for it.

## 7. Conclusion

Frontier AI adoption is won at the boundaries. Enterprises that govern the data path — ingress, egress, entitlement and action — will ship AI capability quickly and defensibly, swapping models underneath a stable control plane as the frontier moves. Enterprises that govern the model will hold excellent committee meetings about a component they were never going to operate anyway, while their actual risk ships in an integration ticket. Govern the path, not the model.

## About the author

Towfeeque Aalam is a Principal Cloud Security Architect with 23 years of experience across aviation, finance, government and healthcare. Formerly a Cloud Architect at Amazon Web Services, he has directed transformation programs up to $1.1 billion, and currently leads the cloud and application security uplift and Frontier AI initiatives at Virgin Australia. He writes at towfeeque.com and can be reached at taufiqaalam@gmail.com.