Approaches to governing AI, compared

Your staff are already using AI. The controls on the market differ less in their feature lists than in **where they sit** and **what they can see**. Where a control sits determines what it can catch. This document maps the common approaches and the gap each one leaves, beginning with the gap that matters most.

> The hardest problem is not the AI you approved. It is the AI you did not: the unsanctioned tools, personal accounts, browser sessions, and command-line clients that never touch a sanctioned path. This is shadow AI, and most controls cannot see it.

The common approaches, and where each leaves a gap

Acceptable-use policy alone

A written rule about what staff may and may not do with AI.

Gap: a document does not enforce anything. It cannot see use, redact a prompt, or produce evidence. Shadow AI continues unobserved.

Network or DNS blocking

A firewall or DNS filter that allows or denies whole destinations.

Gap: the decision is all-or-nothing. Block a provider and staff route around it through another tool or a personal device; allow it and everything passes unseen. It cannot read inside an encrypted session, redact, or keep a usable record.

An API gateway that requires integration

A proxy that applications are configured to send their AI calls through.

Gap: it governs only the traffic that was wired to it. A new tool, a personal account, a browser, or a command-line client that was never configured bypasses it entirely. It typically terminates the session centrally and sees plaintext, which makes the operator a custodian of your data. **It misses shadow AI by design.**

A closed or walled-garden assistant

An enterprise AI suite that governs its own sanctioned assistant.

Gap: it governs one tool. The moment a user opens a different provider, it is blind. Coverage equals a single application, not the institution.

A browser extension or per-app plugin

A control installed into one surface, usually the browser.

Gap: it covers that one surface. Desktop apps, IDEs, command-line clients, and other browsers slip past, and the user can often disable it.

Data-centric DLP or CASB

Tools built to watch files and SaaS usage, often by sampling.

Gap: they are not aware of the AI decision itself. They observe data movement, frequently after it has already happened, rather than deciding before content leaves the device.

Provider-side controls

Relying on the model provider's own settings and logging.

Gap: by the time those controls apply, the data has already left your boundary. The record belongs to the provider, not to you, it is not independent or tamper-evident, and it does not span the other providers your staff use.

Where Verillian sits

Verillian intercepts AI traffic **on the device, at the TLS layer, before egress**. Because the control sits at the point where traffic leaves the machine, it sees what every other approach has to assume away.

- **It catches shadow AI.** It sees the AI traffic leaving the device regardless of the application, the provider, or whose account it is. Nothing has to be wired up to be seen.
- **No integration, no application changes.** The control does not depend on each tool being configured to cooperate.
- **It decides before egress.** Policy is applied at the moment of execution: allow, redact, or block. When a decision is uncertain it fails closed. A blocked request never leaves.
- **The keys and the data stay with you.** Content is encrypted under your institution's key. The server stores ciphertext it cannot read. No outside operator becomes the custodian.
- **The evidence is yours and it is tamper-evident.** Every event is Ed25519 signed and SHA-256 hash-chained into a record your institution holds. Any modification breaks the chain.
- **It spans providers, and it works air-gapped.** One control, every model, no dependency on a vendor's connectivity.

The same question, answered side by side

APPROACH	CATCHES SHADOW AI	NO INTEGRATION NEEDED	DECIDES BEFORE EGRESS	REDACTS, NOT JUST BLOCKS	YOU HOLD TAMPER-EVIDENT EVIDENCE	SPANS ALL PROVIDERS
Acceptable-use policy	○	●	○	○	○	○
Network / DNS blocking	◐	●	◐	○	○	◐
API gateway (integrated)	○	○	●	◐	◐	◐
Closed assistant	○	●	◐	◐	○	○
Browser / app plugin	◐	●	◐	◐	○	○
DLP / CASB	◐	◐	○	◐	◐	◐
Provider-side controls	○	●	○	◐	○	○
Verillian	●	●	●	●	●	●

● meets it ◐ partial or conditional ○ does not

The point

Where a control sits determines what it can catch. Sit downstream of the decision and you inherit every blind spot upstream of it.

Most approaches govern the AI you already approved. The risk lives in the AI you did not. A control at the device, deciding before egress, with evidence you hold, is the only position that sees the whole picture. The model proposes. Verillian decides.