Legally vs. Technically Enforced

2026-07-09

A framework for telling a legally-enforced sovereignty promise from a technically-enforced one, applied across BYOK, RBAC, and AI-agent access. Part 4 of 6.

The same distinction has come up three times already in this series without being named directly: BYOK in Part 1, two vendors' own statements in Part 2, and Kubernetes RBAC against a written `CLAUDE.md` rule in Part 3. This post makes it explicit, adds a third rung most sovereignty marketing doesn't reach yet, and applies it across everything the series has found so far — no new research, just naming a pattern that kept repeating.

What makes a control legally-enforced

A legally-enforced control is a promise, a contract, a corporate policy, or a written rule that someone — a company, an employee, an AI agent — is *expected* to follow. It can be tested, reinterpreted, overridden by legislation, renegotiated after an acquisition, or simply not followed. Part 3's finding that many AI agents in production have no real audit log is the same category of fragility: a written rule ("never run `git push` without asking") backed by partial technical friction, not a guarantee that holds regardless of what the agent — or a court, or a new owner — decides to do.

What makes a control technically-enforced

A technically-enforced control means the actor is *architecturally incapable* of the action, independent of intent. Part 1 already had the cleanest example: OVH's SSE-C model has the customer supply the raw encryption key on every request, and OVH's own documentation states that losing the key makes the data permanently unreadable — because OVH itself never stores it. That's not "we promise not to look." It's "we cannot look, because we do not have the key." Part 3's Kubernetes RBAC and LiteLLM virtual-key examples work the same way for access instead of encryption: the API server or gateway rejects an out-of-scope call regardless of what the caller intends, the way a written rule alone never can.

Three rungs, not two

Treating this as a strict binary undersells it. BYOK is real technical enforcement — and it still isn't the strongest rung available:

- **Rung 0 — legally-enforced only.** A policy promise, a contract, an internal process someone is supposed to follow. Most of what this series has found so far sits here: Microsoft's "Data Guardian" claim, AWS's "Qualified… Staff" language, Bleu's request-based right to inspect specific portions of Microsoft's source code.
- **Rung 1 — technically-enforced at rest.** BYOK, SSE-C, customer-managed keys in a KMS. The provider cannot read stored data without the customer's key — but its compute layer still handles

the decrypted plaintext in memory while a query or workload actually runs. The guarantee covers data sitting still, not data being processed.

- **Rung 2 — technically-enforced at rest *and* in use.** Confidential computing: workloads run inside a hardware-backed enclave that keeps memory encrypted even from the host operating system and hypervisor, with remote attestation letting a customer verify what code is actually running before trusting it with anything. In principle, this closes the Rung 1 gap — the operator's own infrastructure never sees plaintext at all, not even briefly. Described here as a general technical concept, not a claim about any specific vendor's current implementation — none of the cases this series has covered so far were found operating at this rung.

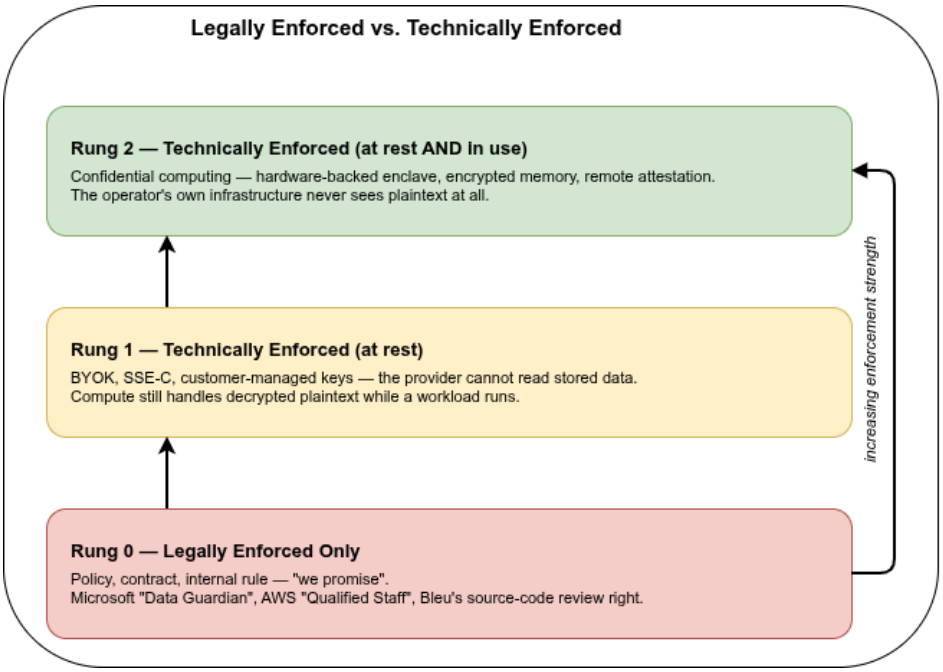

Figure 1: Diagram: three rungs of enforcement — legally-enforced only, technically-enforced at rest, and technically-enforced at rest and in use

A synthesis table across the series

Mechanism	Where it appeared	Rung
BYOK / SSE-C (AWS, GCP, Azure, OVH)	Part 1, Question 3	1 — technically-enforced at rest
Kubernetes RBAC	Part 3	Technical (access axis) — API server enforces regardless of intent
LiteLLM per-user virtual keys	Part 3	Technical (access axis) — gateway enforces regardless of intent
Microsoft "Data Guardian"	Part 2	0 — legally/organizationally enforced, self-reported, unaudited
AWS "Qualified… Staff"	Part 2	0 — legally/organizationally enforced
Bleu's request-based source-code review right	Part 2	0 — legally/organizationally enforced, on-request only
`CLAUDE.md` Git Safety Protocol	Part 3	0 — written policy, partial technical friction via permission prompts

Reading the table straight: the technically-enforced rows are the ones an outside party doesn't have to trust — they hold regardless of who's asking or what a court later decides. Everything at Rung 0 depends on the promise being kept.

Why the CLOUD Act is the sharpest real-world test of this

A legally-enforced "we don't share EU data with foreign authorities" claim depends on that claim surviving contact with a jurisdiction that asserts extraterritorial reach — which is exactly what Microsoft France's own director of public and legal affairs, Anton Carniaux, testified under oath to the French Senate in Part 2: asked directly whether he could guarantee French citizens' data would never reach US authorities, he answered "No, I cannot guarantee it." A technically-enforced "we cannot read it" claim doesn't need to survive that same test, because complying with an order to hand over unreadable data isn't a matter of willingness — the provider does not have what's being asked for. This blog is not qualified to assess how CLOUD Act jurisdiction actually plays out in a specific legal dispute (see the disclaimer already established in Part 1 and Part 2) — what it can point to is which category a given control falls into, and specific statements and incidents already on the public record for a reader to weigh for themselves.

That kind of dispute isn't hypothetical. In February 2025, a US executive order sanctioning ICC chief prosecutor Karim Khan reportedly left his Microsoft Outlook account inaccessible, prompting the ICC to move its email to Proton Mail, a Swiss provider (Computer Weekly, heise). Microsoft's president publicly denied suspending the account and said the company stayed in contact with the ICC throughout (Techzine). What actually happened is contested — and that's itself the point: a legally-enforced control's behavior under real political pressure can end up disputed even after the fact, in a way a technically-enforced one leaves no room for. The provider either has the key or it doesn't; there's no equivalent disagreement to have about that.

None of this is a recommendation for any specific rung. A workload with no realistic exposure to foreign-jurisdiction requests may not need Rung 1, let alone Rung 2; a regulated workload handling

data under strict national-security constraints might need Rung 2 and not have it available from any provider today. Which rung is "enough" depends entirely on what's actually being protected and from whom — this post names the categories, it doesn't rank them.

Next in this series: Part 5 — Can You Leave? Data Portability & Egress →

Related

- Sovereign-Cloud-Washing: Five Questions — Question 3, where this framework first appeared
- Who Builds the Platform? Ownership vs. Stack — the Microsoft/AWS statements and Bleu's source-code review right
- Who — or What — Has Access? — RBAC, LiteLLM keys, and the `CLAUDE.md` policy example
- Log Archiving Security & Compliance — the full BYOK/SSE-C provider comparison behind Rung 1