BYOK-Always: An Economic and Trust Architecture for Multi-Tenant AI

Abstract

Multi-tenant AI platforms almost universally intermediate inference: the platform holds keys, funds tokens, and proxies model calls on the customer's behalf. We argue this default couples three concerns that should be separated — *paying for intelligence*, *holding the credential*, and *proving what happened* — and that decoupling them yields a materially stronger economic, trust, and safety design. We describe **BYOK-Always**, the doctrine implemented in ARQERA: every model call runs on the *tenant's own* provider key, routed *directly* to the provider, never funded by the platform and never proxied through a platform gateway; the platform sells the operating system, orchestration, and trust layer per seat, not the tokens. We ground the design in production data — **138 tenant-owned provider configurations across 23 providers (18 with live adapters)**, ~397,000 recorded inference-usage rows — and in a concrete failure that BYOK-Always structurally prevents: a platform-owned inference gateway whose expired auth token silently broke *every* customer's model call, while the customers' own keys sat valid and unused. We formalise the separation of concerns, show how an append-only hash-chained evidence ledger (**11.2 M rows**) supplies the "prove what happened" leg without the platform ever touching the customer's key, and discuss the safety consequence: a platform that cannot spend the customer's tokens cannot be economically incentivised to over-call models, and a platform that never holds the key removes an entire class of credential-orphaning failures.

---

1. The default and its coupling

The standard multi-tenant AI platform is a metered reseller of intelligence. It holds model-provider keys, meters the customer's usage, marks up the tokens, and routes calls through its own infrastructure. This bundles three genuinely distinct concerns:

1. **Payment** — who pays the model provider for the tokens.
2. **Custody** — who holds the API credential that authorises the call.
3. **Provenance** — who can later prove what was asked, decided, and produced.

The reseller model couples all three onto the platform: the platform pays, the platform holds the key, and the platform is the only party positioned to attest to what happened. Each coupling has a cost.

- **Payment coupling** creates a margin incentive. If the platform earns on token markup, its incentives are (weakly) aligned with *more* inference, not *better-governed* inference. It also caps the platform's own margin to the spread over provider cost, and exposes it to provider price moves.
- **Custody coupling** makes the platform a credential honeypot and a single point of credential failure — if the platform's key or gateway breaks, *every* tenant breaks at once (§4).

- **Provenance coupling** ties "what happened" to the same party that ran the call, which is exactly the party a sceptical auditor least wants as the sole witness.

BYOK-Always decouples them.

---

2. The doctrine

> **BYOK-Always.** Every inference call runs on the *tenant's own* provider credential, sent *directly* to the provider. The platform never funds a token and never proxies the credential through a platform-owned gateway. The platform charges for the operating system, orchestration, and trust layer — per seat — not for the intelligence. Intelligence is the tenant's, always.

Three consequences fall out immediately:

- **Payment is the tenant's**, so the platform has *no* token-margin incentive. It earns on seats and orchestration, whose incentives point at *value delivered per governed action*, not tokens burned.
- **Custody is the tenant's**, and the credential goes *direct to the provider*, so the platform is neither a honeypot nor a shared point of failure for inference.
- **Provenance is separated** from custody and payment: the platform proves *what happened* through an independent, append-only evidence ledger (§3) without ever holding the key.

In ARQERA this is not aspirational plumbing. The production database holds **138 tenant-owned provider configurations** spanning **23 distinct providers**, of which **18 have live direct adapters** (Google, OpenAI, Anthropic, Groq, DeepSeek, Mistral, Fireworks, Together, Cerebras, xAI, Perplexity, Anyscale, HuggingFace, AI21, Baidu Ernie, MiniMax, OpenRouter, Ollama) and 5 are declared-but-pending. Roughly **397,000 inference-usage records** and **685,000 token-allocation rows** are attributed to tenant keys, not a platform pool.

2.1 The routing rule that enforces it

The doctrine reduces to a single, checkable routing invariant:

> A request's destination base URL is determined by its **key source**. A `BYOK` key routes to the provider's *native* endpoint (`api.groq.com`, `generativelanguage.googleapis.com`, …). Only a `PLATFORM_MANAGED` key may use a platform gateway.

This one rule is where the economics become a safety property, as §4 shows.

---

3. Provenance without custody: the evidence ledger

Decoupling provenance from payment and custody requires an independent witness. ARQERA's is an **append-only, hash-chained evidence ledger**: every governed action emits an artifact carrying a `payload_hash`, a `chain_hash` linking it to its predecessor (`chain_prev_hash`), and an HMAC signature. The chain is third-party verifiable — anyone holding the signing parameters can recompute the hashes and detect any insertion,

deletion, or mutation — and it is replicated across three stores (a relational database, and two object stores) so that no single store is a point of permanent loss.

At production scale this is **11.2 million evidence artifacts / 17 GB**, spanning **252 distinct artifact types over 174 days** at roughly **64,000 artifacts per day**. Critically, *none of this requires the platform to hold the tenant's model key*. The ledger records the governed decision and its outcome — the intent, the authority check, the action, the result — not the provider credential. Provenance is thus supplied by a party (the ledger) structurally separate from the party that ran the inference (the tenant's own key, direct to the provider). This is the auditor-friendly configuration the reseller model cannot offer.

(The evidence ledger's engineering — and the request-path scaling problems it created — is treated separately in the companion paper on intelligence wieldability.)

---

4. When custody coupling breaks everyone at once

The clearest argument for BYOK-Always is a failure that it structurally prevents, observed in production.

Before the doctrine was fully enforced on the streaming path, the platform's AI assistant returned a bare failure to users. Tracing to ground truth: a tenant with a *valid, working* Google/Gemini key was never served by it. The router sent the request through the platform's own managed AI gateway — and that **gateway's auth token had expired**. Because the gateway is a shared platform credential, *every* provider routed through it returned `401 Unauthorized`. The customer's own key, which would have hit the provider directly and worked, was never tried; the fallback tried only *another* gateway-routed provider, which also `401`'d, yielding an internal "all providers failed."

This is custody coupling's signature failure: **one platform-held credential expiring silently breaks every tenant's inference at once**, while every tenant's own valid credential sits unused. The fix was to enforce the §2.1 invariant — `BYOK` keys go direct to the provider; only `PLATFORM_MANAGED` keys touch the gateway. After the fix the assistant streamed real tokens from the tenant's key directly (verified live: a response streamed token-by-token from the provider's native endpoint where it previously failed), and a native-endpoint fallback correctly reached a second provider (returning a clean provider-side quota response, not a gateway `401` — proof the request had left the gateway entirely).

The general lesson: **every layer that proxies or re-holds a customer's credential is a place the platform can break the customer while the customer's own asset is fine.** BYOK-Always removes the layer.

---

5. The economic model

If the platform does not sell tokens, what does it sell? It sells the OS, the orchestration, and the trust layer, per seat — a Microsoft-365-style ladder rather than a metered-intelligence bill. In ARQERA the ladder is five tiers (per seat, monthly): Personal, Team, Business, Pro, and a contact-sales Enterprise, differentiated by *platform* capabilities — seat and connection limits, evidence-retention depth, governance depth (from basic auto/soft/hard tiers up to approval chains, policies, compliance, and audit), and features like SSO, shared workforce, and pods. **No tier includes intelligence; a validated provider key is required to activate any account.** Agents and workforces are seller-priced items in a marketplace; the platform plan never gates on them.

This has three properties the reseller model lacks:

1. **Incentive alignment.** With no token margin, the platform earns strictly on *governed value per seat*. It has no reason to prefer more model calls; if anything it benefits from the tenant's inference being efficient (a happier tenant renews the seat). The platform's and the tenant's incentives on inference volume are *aligned to minimise waste*, not opposed.
2. **Margin independence from provider prices.** Seat revenue does not compress when a model provider raises prices, nor does it depend on maintaining a spread. Provider price wars are the tenant's upside, not the platform's risk.
3. **A hard activation gate as a safety floor.** Requiring a validated key to activate means there is no anonymous free-inference floor for the platform to fund — removing both a cost sink and an abuse surface, and making every call attributable to a real, paying credential holder.

A second-order effect worth noting: because the enforcement of these limits must match what the pricing page *displays*, ARQERA derives both the enforced limits and the customer-facing catalog from a single canonical plan definition, and derives the payment-provider price objects from that same definition (self-healing by metadata rather than by hand-set identifiers). We found the failure mode this prevents in production: a stale, hand-set price identifier orphaned checkout with a "no such price" error and no signal. The canonical-source design — one definition, everything derived, a deploy gate that fails closed if any purchasable tier cannot resolve to a live price — makes display-vs-enforcement drift structurally impossible. This is the pricing analogue of BYOK-Always's core move: *one source of truth, everything else derived, no hand-maintained duplicate to drift.*

---

6. Discussion

BYOK-Always is best understood as **the separation of "paying for intelligence," "holding the credential," and "proving what happened" into three independent parties** — the tenant, the tenant, and an append-only ledger, respectively — where the reseller default fuses all three onto the platform.

The design generalises beyond model tokens to any metered third-party capability a multi-tenant platform intermediates (storage, compute, messaging): the same coupling analysis applies, and the same decoupling — the customer's own credential, direct to the provider, with an independent provenance ledger — yields the same incentive, custody, and provenance benefits. The trade the platform accepts is real: it forgoes token margin and the lock-in that a held credential creates. We argue this is the *right* trade for a governance-and-trust platform, because its product is precisely the thing the reseller model compromises — a credible, independent account of what the customer's AI did, produced by a party with no incentive to have it call the model more.

Limitations

The incentive-alignment argument (§5) is a design argument, not a measured behavioural result; we have not run a controlled comparison of token consumption under reseller vs BYOK-Always pricing. The 138 provider configurations and ~397K usage rows demonstrate that BYOK is *live and load-bearing*, but do not by themselves prove tenants are better off economically than under a hypothetical resale plan — that would require a cost study across representative tenants. The activation-gate-as-safety claim (§5.3) assumes a validated key is a meaningful identity signal; a determined abuser can still supply a real key. Finally, BYOK-Always shifts the operational burden of key validity onto the tenant: a tenant whose own key expires or exhausts quota sees their AI stop — which is *honest* (the failure is theirs and legible) but is a support and UX surface the reseller model hides. The

right handling is an explicit, actionable activation state ("add or refresh your key") rather than a silent failure — which is itself a wieldability requirement.

---

7. Conclusion

The multi-tenant AI default — the platform pays for, holds, and proxies the customer's intelligence — couples payment, custody, and provenance in a way that misaligns incentives, concentrates credential risk, and compromises the very audit trail a trust platform sells. BYOK-Always decouples them: the tenant pays, the tenant holds the key, it goes direct to the provider, and an independent hash-chained ledger proves what happened. The production evidence — 138 tenant-owned provider configs across 23 providers, ~397K attributed usage rows, an 11.2 M-row evidence ledger, and a concrete gateway-custody failure that the doctrine prevents — suggests the decoupled design is not only cleaner but safer and better-aligned. A platform that never buys a token has no reason to burn one; a platform that never holds the key cannot lose it for you.

Appendix A — Headline figures

Figure	Value
Tenant-owned provider configurations	138
Providers in routing registry / with live adapters	23 / 18
Recorded inference-usage rows	~397,000
Token-allocation rows	~685,000
Evidence ledger (independent provenance)	11.2 M rows / 17 GB
Evidence artifact types / span	252 / 174 days
Platform pricing tiers (per seat, no intelligence bundled)	5
Gateway-custody failure prevented	1 platform token → all tenants `401`

All figures measured live on the production platform, 23 July 2026.