BGL BURAK GALİBA
Thoughts on AI Transformation

ESSAY · ENTERPRISE AI

Your AI Agents Are Stalled at IT — The Access Layer Nobody Built

Your tier model is published, but agents have nothing to connect to. Why direct database access fails, and how a thin, governed access layer fixes it.

AGENTIC AI | GOVERNANCE | AI STRATEGY

Burak Galiba · July 27, 2026 · 4 min read
https://burakgaliba.com/blog/agents-need-data-the-missing-access-layer

SCAN TO READ ONLINE

You did the work. You published the tiered access model from the paved road: sandbox, read-only production, write-with-approval, scoped autonomy. The first hackathon agent filed its package, named an owner, cleared review for Tier 1. It needs customer orders.

Then someone asks the question that stalls everything: connect to what?

The orders live in a database owned by a fifteen-year-old line-of-business system. No API. No MCP server. The application talks straight to its own tables, and the only integration paths are a nightly export, a shared read replica, and — if someone is feeling generous — a database login.

Your tier model is published. Your agents have nothing to plug into. An MIT NANDA report found that 95% of enterprise GenAI pilots deliver zero P&L impact — and in my experience, "the agent couldn't reach the data" is exactly the kind of failure that hides inside a number like that. This is where the agentic production gap lives: not in the model. In the plumbing.

Three doors, two of them traps

Door one: hand the agent database credentials. Fast, familiar, and the anti-pattern. SQL grants are coarse — schema-level or table-level, with row- and column-level security existing mostly on paper. The agent sees every column, including PII it never needed. The audit trail records queries, not intent: you know a SELECT ran at 2 a.m., not which task ran it or why. And writes that bypass the application bypass every validation and business rule the application enforces.

Here is the part that ends the debate: a tier model is unenforceable when the only credential is a database login. You cannot build Tier 1 — read-only, scoped, attributable — on top of `SELECT *`.

Door two: wait for the vendors. MCP — the Model Context Protocol, an emerging open standard for connecting agents to tools and data — is real, and support is growing. But most enterprise systems don't have an MCP server today, and your homegrown systems never will unless IT builds one. Waiting is a plan with no date on it, and you know what undated plans do to sponsors.

Door three: build a thin, governed access layer. The answer — and smaller than it sounds.

Demand-driven, not ocean-boiling

Do not API-ify the estate. That's a three-year platform program, and it will die in scoping. The agents already waiting at your door define the scope for you — the queue is the backlog. In most organizations that's two or three systems — orders, inventory, tickets — each with a named agent and a named data need attached. Build interfaces for those. Ignore the rest until an agent shows up wanting them.

Read path first

Start with reads — what most waiting agents actually need, and the risk you can bound.

- Build a database view — or point at a read replica — exposing only the columns the task requires. Filter columns and mask PII in the view itself, not in the client.
- Wrap it in a curated read endpoint, or an MCP server, offering a handful of scoped operations.
- Give each agent its own least-privilege service account. Never a human's credentials, never a shared login.
- Log every call from day one, attributed to the agent and the task it was serving.

A view plus a thin facade is days of work, not a platform program — and already more governance than most estates have ever shipped.

Tools mirror business operations, not tables

Expose `get-customer-orders(customer_id, date_range)`. Never expose `run-sql`.

The interface is two things at once. It's the security boundary: the agent cannot ask for what the tools don't offer. And it's the semantic layer: an agent reasoning over business-shaped operations is safer *and* more accurate than one reasoning over raw schemas, because there are no joins to get wrong and no cryptic column names to misread. And the audit log finally records intent, because the tool call *is* the intent.

Writes go through the application — or through a human

Where an application API exists, the agent calls it and inherits every validation and business rule for free. Where none exists, writes land in an approval queue that a human reviews and executes: the agent proposes, a person commits. That's Tier 2 from the paved road, made real.

Direct database writes by agents are banned. Full stop.

Every interface gets an owner

Each exposed interface needs a named system owner and a documented scope: which data, which operations, for which agents. That's the governance committee's published standard doing its job. Put bluntly, the tiers describe what agents may do; the access layer is what physically prevents them from doing more. Without it, your tier model is a poster.

One more thing, for the budget conversation: none of this is agent-specific plumbing. A governed access layer serves every future integration, analytics request, and vendor swap. It's the data-foundations leg of AI readiness — agents are just the first consumers impatient enough to force the issue.

This week

Pick one system — the one the most waiting agents need. Ship a read-only view with PII masking, wrap it in a thin facade or MCP server with two or three business-shaped tools, issue the first agent its own service account, and turn on logging. Assign an owner before it goes live. That's days of work, and it turns your tier model from a document into an enforced boundary.

To see where your data foundations stand next to your pilots, talent, and governance, my free AI Readiness Score takes about ten minutes — 20 questions across all four — and shows which gap will bite first. The tier model tells your agents what they're allowed to do. The access layer is what lets them do it.

Read the essay online at https://burakgaliba.com/blog/agents-need-data-the-missing-access-layer

I built a free AI Readiness Score — 20 questions across pilots, data, talent and governance, about ten minutes, no email required: https://burakgaliba.com/readiness