
PROFORMA GLOBAL RESEARCH

Is AI Part of IT?

Matt Rollings, Founder and Principal, Proforma Global

Whitepaper · 2026-06-01

Executive Summary

Background. AI runs on the infrastructure IT already owns so leaders reach for IT the way they would for any other system. IT does own part of AI, but only one part. An AI program splits into four separate jobs that no single owner holds at once. IT owns the architecture the agents run on. The business function owns the use cases worth building and the meaning of its own data. No one outside that function can read either one correctly. A new AI team builds and runs the agents themselves. Leadership hands all four to IT because the infrastructure is IT's and the work looks technical. The real question survives that hand-off untouched. Leaders still need to decide who owns each of the four jobs and whether AI reports to IT at all.

Approach. This whitepaper splits an AI capability into four responsibilities and shows why only architecture belongs to IT. Leaders worry first about the org chart. AI can sit inside IT or Finance or a function of its own without changing the work underneath. The four assignments decide the outcome. A working program gives those four owners one place to run them. Enterprise systems settled this same question over thirty years. The last section reads off their answer.

1. The Question Behind the Question

"Is AI part of IT" is a question about ownership. Most companies answer it before they understand the work they are placing. IT earns the default fairly because it already concentrates the infrastructure and security the rest of the business leans on. AI runs on that same foundation so the classical instinct files it under IT alongside every other system. The instinct breaks the moment AI stops behaving like those systems.

An agent does not behave like the systems IT usually runs. Nobody provisions it once and then walks away. You embed it inside the work of one function and shape it around how that function operates. An agent kept at arm's length from the business it serves does the wrong job well. A leader who asks which existing box AI fits hands a cross-functional effort to one owner before anyone has named the work.

A serious AI deployment carries four separate responsibilities, and no single function holds more than two. The architecture, the use cases, the meaning of the data, and the agents themselves each answer to a different owner.

2. Architecture Belongs to IT

Architecture belongs to IT and always has. The infrastructure an agent runs on sits with IT the same way the databases and the security model do. No other function can hold it. An AI team that stands up its own data platform off to the side has not escaped IT so much as built a second one that is worse at the job and answers to no one.

IT already runs these controls as a profession, and a program built inside that boundary moves faster rather than slower. Architecture poses no real difficulty here. Leaders create the real problem one step later when they read ownership of the architecture as ownership of the AI. The rest of this paper takes that misreading apart.

3. What Gets Built Belongs to the Business Function

The business function decides what to build, in what order, and whether the work paid off. The function holds the use cases because it knows which tasks cost the most and which are safe to change. The function also controls the budget and answers for every dollar of it. IT cannot prioritize a finance use case it does not live with. An AI team cannot rule that automating reconciliation matters more than automating a forecast. The function that owns the process owns the decision about what the process becomes.

Organizations strip this responsibility from the people who should hold it more often than any of the other three. A program run on budget control rather than outcome ownership optimizes for what is buildable instead of what is valuable.

4. What the Data Means Belongs to the Business Function

The function owns the meaning of its own data as a second responsibility, and people keep misreading that one as technical. IT's databases hold the rows and columns, but their meaning lives somewhere else. Only the function that uses the data can author its definitions, the processes that produce it, and the metadata that makes it readable. Finance decides what counts as a closed period, how the books recognize revenue, and which adjustments fall inside policy. The team that administers the storage answers none of those questions. IT enforces who may see which data through the access model. The function defines what the data means. A working program needs both jobs because they do different work on the same asset.

These definitions form the semantic layer, the work that decides whether an agent can reason about a business or only push its raw numbers around. We cover the engineering of that layer in [Semantic Layers in Enterprise Agent Systems](#) and the broader requirement across the [Data Architecture for Enterprise Agents](#) series. The definitions belong to the function because no one else can write them correctly.

5. Agent Development Is the AI Function

The fourth responsibility belongs to neither IT nor the business because it never existed as a standing job before. A new AI function designs the agents, builds them, instruments them with telemetry and observability, and trains the people who will use and extend them. That function takes a use case validated on IT's architecture and the function's data and turns it into a system the team can watch and correct in production.

The architect owns this work, the role we describe in [Your Board Wants AI. Where Do You Start?](#). Agent development breaks in ways classic software delivery does not and no output earns trust until someone verifies it. The work stays too technical for a business role and too unlike software for IT's delivery practice. A team builds an AI capability rather than configuring a packaged product, so it stands up as a new discipline rather than an extension of either side.

The AI function also owns the standard beyond building the agents. That standard sets the model and tooling choices, the verification and observability practices, and the platform pattern every function's agents are built on. The same standard keeps distributed ownership from fragmenting into a sprawl of incompatible systems. AI risk, compliance, and cost ride

across all four responsibilities, so the AI function holds them through the standard rather than parking them in any one place. We map how the capability should grow by stage in [The Agentic AI Maturity Curve in Enterprise Finance](#).

6. Who Should Own AI in the Enterprise? No Single Function Holds All Four

The four responsibilities laid side by side answer the original question plainly. IT holds one. The business function holds two. The new AI capability holds the fourth, and it carries the cross-cutting concerns of risk, compliance, and cost on top. No group on the current org chart holds all four, which makes AI ownership cross-functional by construction rather than by preference.

Any single owner handed the whole program fails in a specific way. IT alone builds what is technically tractable and never settles what the data means or what is worth automating. A business function alone underestimates the architecture and the engineering. A standalone AI team with no authority over data or use cases produces demonstrations no function adopts. The work splits across all three owners, so the structure has to reflect the split.

7. What Enterprise Systems Already Showed

The structure that follows is not new. Enterprise systems settled into it over thirty years. The most effective organizations embed ownership of a system in the department that uses it and treat IT as the partner that makes it run.

System	Owned by	What IT supports
ERP (general ledger, procurement, supply chain)	Finance and Operations	Managing master data, and feeding transactions to the warehouse
EPM (planning, consolidation, close: ARCS, FCCS, EPBCS)	Finance	Integrating actuals from the source systems into the model
HRIS (Workday, SuccessFactors)	HR	Securing employee data, and managing identity and access

System	Owned by	What IT supports
CRM (Salesforce)	Sales and Revenue	Moving customer data into the analytics layer

The department that lives with the outcome holds the system, sets its priorities, and answers for whether it returns. Look closely at IT's column. The vendor now owns the platform, because a modern SaaS application ships its own hosting and much of its own integration, so IT no longer stands the system up or wires it to the next one. IT keeps the data architecture beneath these systems, the lakes and warehouses and ETL that pull their data into one place, plus identity and security. IT contributes the data layer rather than the application.

Architecture still belongs to IT here, even though the vendor handles the platform for a system the enterprise buys. IT's role narrows to the data layer beneath that kind of system. AI arrives differently. A team engineers AI in-house, so the architecture under it stays the fuller one named earlier, the data platform, the security model, the environments, and the access controls, and every piece of it belongs to IT. A single row in this table understates IT's role for AI rather than defining it.

AI does not fit in any single row of the table. Each system above touches one function most, so each settled with a single owner. AI touches nearly every function and over time integrates with all of those systems. It reads from the ERP, plans through the EPM, acts in the CRM, and learns from the data each of them holds. AI runs across the whole line instead of sitting in it as the next single-function system. Two reasons now rule out a single owner. No one function holds all four responsibilities, and the capability itself belongs to no function because AI serves them all.

8. So Where Should AI Sit, IT, Finance, or Its Own Function?

AI sits most clearly as a sister to IT. IT concentrates skills, infrastructure, security, and systems in one place to support the rest of the organization. The AI function stands beside it and concentrates the skills that make those systems intelligent. The two functions work as peers. Peer status describes how the work divides and where accountability sits, and the reporting line can vary without changing either.

Most organizations land on the same practical answer. A small senior AI function owns agent development and the standard, partners with IT for architecture, and works with each business function on its use cases and data. The organization can place that function inside IT, inside

Finance, or on its own. The org chart stays negotiable, but the split of responsibilities beneath it holds firm because the work itself fixes it.

IT can own AI well when its culture and staffing fit the work. The culture decides the call. AI rewards rapid iteration and deployment. A group that runs on the release cadence and change control a system of record demands will slow AI without meaning to. The group AI sits in has to move at the speed adoption needs and stay close enough to the business to keep the use cases honest, wherever it reports.

The structure has to resolve contention, because shared ownership with no tie-breaker is no ownership at all. Each business function sets priority within its own domain. IT holds the architecture all of them depend on. The AI function owns the standard and arbitrates across functions when they fight for the same capacity or when a use case outruns the data behind it. When the AI function and IT themselves disagree, IT's architecture and security constraints win and the standard yields to them. The data layer makes accountability joint by design. The function owns what an agent may do, IT owns what it may reach, and a misbehaving agent becomes both their problem. This arrangement does not pit AI against the IT department. AI runs enabled by IT and ends up owned where the work and the accountability actually sit.

9. Closing

AI works as an enablement function, a sister to IT, and its ownership stays cross-functional. The ownership question answers itself once you name the work. Architecture belongs to IT. Functionality and data belong to the business function. Agent development and the standard form their own discipline. The AI function holds the cross-cutting concerns through that standard. The organizations that deliver stop assigning AI to one existing owner and start assigning each responsibility to the group that can hold it, with IT enabling the whole.

An enterprise that gets this right builds systems that are AI-native rather than AI-attached. The enterprise designs the parts it owns for agents to operate from the start, its data architecture, its semantic layer, and the agent platform the AI function builds. Bought systems come from vendors whose platforms are built the same way rather than from agents bolted on afterward. The ownership question exists to enable exactly that work, which is why a company should settle it now before the first program settles it by default. We take up where that begins, and how to prove the first use cases before any of this settles at scale, in [Your Board Wants AI. Where Do You Start?](#) and [Implementing Your First AI Agent Pilot](#).

[DOWNLOAD PDF](#)

Work with Proforma Global

The thinking in this paper is public; the methods that turn it into a working system are not. If it fits a problem your team is working on, that is what we bring to an engagement. Start a conversation: info@proforma.global.

© 2026 Proforma Global. All rights reserved.

This paper is published as Proforma Global Research. The text and figures are the property of Proforma Global.

Brief excerpts may be quoted under fair use with attribution to Proforma Global Research and a link to the canonical URL. Permission requests: info@proforma.global.