CDATA

The MCP Platform Advantage

The MCP Platform Advantage: Choosing the Right Path to Enterprise Scale

Enterprise AI architects are designing for something that didn't exist five years ago: a system that must simultaneously answer a question, execute a workflow, and act autonomously — governed, cost-contained, and correct — across dozens of live enterprise data sources. This presents an infrastructure problem, not a model or data pipeline problem, and the decisions architects and IT leaders make now will determine whether enterprise AI scales or stalls.

The core decision — attempt to build an access and control plane with source-native Model Context Protocol (MCP) servers, or incorporate a centrally managed MCP platform across all target data sources — largely comes down to required levels of data control, context, and connectivity; the need to support deterministic, non-deterministic, and autonomous workflows simultaneously; and the degree of pressure to keep token consumption in check.

While building with source-native MCPs may appeal on the surface, significant costs and trade-offs arise quickly that ultimately push enterprise AI architects and IT leaders toward a centrally managed MCP server approach. This paper walks through the AI architect's core infrastructure objectives, details the build vs. buy decision at hand, and discusses the considerations and trade-offs of the two fundamentally different approaches.

The enterprise AI architect's mission

The data infrastructure requirements for enterprise AI are a complex web that must all hold true at once. There are six core objectives AI architects must balance and design for in creating a scalable enterprise infrastructure.

Provide LLMs controlled accessibility to a wide breadth of rich enterprise data sources. The value of AI in the business is proportional to its access to actual enterprise data — Salesforce records, NetSuite financials, Jira tickets, BambooHR profiles, Zendesk cases. Furthermore, this data is often spread across cloud applications, custom data sources, and on-prem network drives, all of which store immense value if accessible by an LLM. As has been well documented, models without access to this data produce generic answers, while models with it enable specificity, consistency, and trust.

Deliver complete control, auditability, and enforceable governance. The ability to ask a question or initiate an action across systems is only possible if the data accessed and returned is governed. That means access controls, policy enforcement, audit trails, and visibility into what data was touched, by whom, and at what cost. According to Gartner's 2025 AI Ethics, Governance and Compliance findings, less than 25% of IT leaders are currently confident in their AI governance. This is a gap that shows up in practice as 69% of organizations have employees using prohibited generative AI tools, copying data into spreadsheets, or uploading it to AI systems outside sanctioned channels.

Support non-deterministic, deterministic, and autonomous workflows collectively. Enterprise AI isn't a monolith. Some use cases highlight discovery, requiring a flexible architecture that can adapt to unpredictable queries without sacrificing accuracy. On the other end of the spectrum are deterministic workflows, which are by nature more predictable and repeatable, but which require greater levels of intra-source access and governance. And then there's autonomous agent deployments, which require a

healthy dose of both open-ended discovery and business rule intelligence, and demand even greater levels of governance, control, and auditability. Enterprises routinely demand all three modes of AI autonomy and must deploy an architecture that supports each in tandem.

Design for "citizen builders" within lines of business. Business units (BUs) asking for AI support today are overwhelming IT teams with requests and "well, what about this idea." Rather than take on the burden of accommodating every BU request, AI architects and IT leaders are encouraged to design a system that places tools confidently in the hands of their business units, as systems requiring central IT involvement for every new use case will face a backlog that compounds indefinitely. The ideal infrastructure allows for delegating building to the business while maintaining control and governance centrally.

Maintain LLM cost control and token ROI. Token consumption has a real — and rapidly growing — cost. Sending unanalyzed raw data sets to the model for summarization is an expensive way to get answers that a SQL query could have pre-authored for a fraction of the price. Architects need a system that can push analysis to the source and return only the context the model actually needs.

Ensure accuracy above all else. A wrong answer delivered quickly is worse than no answer at all. In a governed enterprise context, an inaccurate AI response doesn't just waste time, it creates downstream decisions made on bad information. Accuracy isn't a nice-to-have; it's the condition on which every other goal depends. And as multi-step workflows and agent systems come online, these inaccuracies compound.

`Accuracy compounds against you`

An industry average accuracy rate of 75% across a five-step workflow results in only 24% of processes completing correctly.

Two approaches to delivering on this mission: build or buy

AI architects still evaluating enterprise AI data infrastructure are typically weighing two options.

Path 1: Adopt a centrally managed MCP platform

A managed MCP platform is a pre-built set of connectors and controls that expose enterprise data sources to chosen AI clients and models through MCP servers. Without requiring custom connector development, data pipeline construction, or ongoing maintenance, MCP platforms handle schema resolution, semantic context, access control, query optimization, and audit logging across hundreds of sources from a single configuration layer. AI tools, agents, and automations connect to a governed, semantic-rich data layer rather than to individual sources.

Path 2: Build through single-source MCPs

While it has taken some time for legacy SaaS vendors to fully embrace the MCP standard, major providers like Salesforce and Atlassian have begun rolling out MCP servers for their native data sources. These source-native MCPs offer one-to-one connections between their data and an AI client, and each source integration becomes a separate project that must be implemented and maintained. Alternatively, some organizations choose to build the MCP server fully in-house. Governance, access control, and

semantic understanding are managed at the application layer, not the data layer, often requiring additional overhead and tools. For example, to implement multi-source AI applications through a source-native MCP architecture, additional tooling like MCP gateways and semantic retrieval layers must be designed, implemented, and managed.

While both approaches can work in a vacuum, the decision comes down to several factors and trade-offs at the enterprise level, outlined in the following considerations.

Considerations for every enterprise AI architect: context, control, connectivity

The differences between a managed MCP platform and a self-built single-source MCP approach aren't always visible at the start of a build. While single-source MCPs typically do not come with a price tag attached, costs quickly arise elsewhere and trade-offs become clear. Below, we walk through these trade-off decisions through the lens of context (bringing in the right data as completely and efficiently as possible), control (governing data access and human-or-agent user actions), and connectivity (securely accessing target data sources consistently and in real time).

Context

Multi-source semantic federation vs. single-source data pipelines. A native MCP approach exposes one source at a time, with the semantics of that source. A federated platform exposes all sources together, normalized to a consistent semantic model. When a sales analyst asks a question that touches Salesforce, NetSuite, and BambooHR simultaneously, the federated approach returns a coherent answer that accounts for the subtle — yet otherwise derailing — differences in terminology across systems. The native approach requires either a pre-built pipeline or a model able to reconcile inconsistent schemas on its own, which isn't a reliable assumption and, as we'll see below, is a costly alternative.

Tool calling: inherited source tools vs. a universal tool layer. Native MCPs give the model the tools the source vendor chose to expose. A managed platform like Connect AI provides not only source-specific tools but also universal tools across data sources, and the ability to add custom tools for your organization's specific workflows. The difference is whether your AI capabilities are defined only by your vendors or with your input as well.

Schema, metadata, permissions, and lineage — not just raw data. AI models make better and more efficient decisions when they understand the structure of the data they're working with, not just its contents. A managed MCP platform delivers data schemas, field-level metadata, relationship context, and permission structures across all connected sources alongside the rows and columns themselves.

Support for non-deterministic, deterministic, and autonomous workflows collectively. As outlined above, the ideal AI architecture supports each of these three workflow modes. Source-native MCP servers typically perform best with deterministic use cases confined to their source system, but they lack the external connectivity and governance controls to meaningfully support the other two modes. By bridging connections across systems, semantically federating and normalizing data, and controlling access and action permissions centrally, a managed MCP server provides the access and control plane to properly support each mode simultaneously.

Control

Platform-level role-based access control plus source-inherited permissions. Native MCPs may inherit the permissions of the authenticated user or service account. That's a starting point, but it offers an incomplete governance model. A managed platform lets you layer additional controls on top — role-based access control (RBAC), attribute-based access control (ABAC), row-level policies, column masking for personally identifiable information (PII) — without modifying the source system.

Policy enforcement at the data layer. For many regulated and sensitive industries, PII masking, data residency requirements, and field-level redaction need to happen before data reaches the model. Enforcing these policies in a prompt or at the application layer is fragile, inconsistent, and may violate end user agreements. Enforcing them at the data access layer is the more reliable and auditable path.

Read vs. read-write access with agent controls. Agentic workflows that write to source systems require different governance than read-only queries. A managed platform can scope write access by agent, by source, and by workflow, and can require human confirmation for write operations above a defined threshold. A build approach requires that logic to live and be managed in each agent's code and be patrolled manually.

Audit logging across the full stack. Native MCP audit logs typically only capture tool calls. A managed MCP server captures and logs data queries, token consumption by user and use case, user and agent identity, and policy decisions — all in one place.

Service accounts for agent access. Human users authenticate through identity providers. Agents authenticate through service accounts. A platform that treats both as first-class access patterns — with separate governance controls for each — is built for agentic AI. One that handles only human access, which is more common among native MCPs, is not.

Connectivity

Single-source vs. multi-source connectivity. While some use cases can be confined to a single source ("find all open enterprise deals for product X in Salesforce"), queries and agentic workflows increasingly require connectivity across multiple sources, and multiple cloud or on-prem environments, in rapid succession. Single-source MCPs, by their definition, offer a one-to-one relationship typically confined to their cloud-based application. A managed MCP platform offers connectivity across sources, regardless of their hosted environment.

Token cost optimization at the source. When managed MCP analysis runs at the source — via a semantic SQL layer unique to Connect AI that returns a summarized result — the model receives a precisely tuned answer, not a raw data dump. Sending a 10,000-row dataset to the model for summarization costs significantly more in tokens than sending the three-sentence summary that SQL already computed. At enterprise scale, this is a material trade-off both in token consumption and latency.

Metadata and relationship context. Answering "which open deals have had no activity in the last 30 days?" requires understanding the relationship between deal records, activity records, and timestamps in Salesforce, not just fetching rows from a table. A source-aware connector understands those relationships, and a semantic layer federates them across sources.

MCP maintenance over time. Native MCP servers are maintained by the source vendors. When Salesforce changes its MCP endpoint, NetSuite updates its API, or Zendesk deprecates a tool, each connection breaks independently. In a self-built architecture, someone on your team — or often a full team itself — owns that maintenance across every source, on each vendor's timeline. A managed

platform absorbs that maintenance; connections and authentication tokens stay current because the platform keeps them current.

LLM and vendor neutrality. An architecture tightly coupled to one SaaS vendor or LLM provider creates lock-in that becomes expensive if the model landscape shifts, which it has repeatedly over the past two years and shows no signs of slowing down. An MCP platform that's model-agnostic lets you route queries to the right model for the right use case, or switch providers, without rebuilding the data layer.

How Connect AI addresses enterprise context, control, and connectivity

CData's Connect AI is built around the recognition that enterprise AI data infrastructure must deliver on connectivity, context, and control simultaneously. When any one of the three is weak, accuracy suffers — and in the autonomous reasoning mode, inaccuracy has consequences before anyone can intervene.

In CData's *State of AI Connectivity Report*, enterprise technology leaders ranked their AI data integration priorities as follows: 42% chose real-time connectivity, 34% ranked semantic data layers, and 60% ranked governance and lineage. Those three priorities map directly to connectivity, context, and control, providing confirmation that the gaps practitioners feel in production are the same gaps addressed by the managed MCP server approach and often unaddressed through a native MCP architecture.

On context and accuracy. Connect AI exposes a semantic layer that resolves schema, structure, and relationships before the model runs. The model doesn't have to infer what a field means or how two objects relate — it's told. The direct impact is measurably higher answer accuracy, reduced latency, and dramatically increased token efficiency. The full analysis of how this translates to accuracy at scale is in CData's AI accuracy whitepaper.

On control and security. Connect AI enforces access policies, RBAC, PII masking, and audit logging at the platform layer, while also inheriting and enforcing pass-through user and service account permissions from each source application. Governance holds regardless of which model, agent, or interface is making the request. The architecture of that control layer is documented in CData's security best practices guide.

On connectivity and maintenance. Connect AI offers and maintains connections to hundreds of enterprise data sources — including custom fields, custom objects, and complex relational structures — across SaaS applications, databases, on-premises systems, and cloud platforms. The full connector library is at cdata.com/drivers.

To see this infrastructure in action, including a live order-to-cash agent demo running across multiple enterprise systems, watch the *AI agents and the future of digital work* webinar with Microsoft.

Build the foundation once

Single-source MCP servers offer an appealing path on the surface toward building an enterprise-ready AI architecture. While they may be the right fit for certain prototyping efforts, the breadth of connectivity, depth of context, and degree of governance and control required for meaningful, scalable enterprise AI value typically keeps source-native MCP architectures from ever reaching production.

By investing in a centrally managed MCP platform, AI architects and enterprise IT leaders are enabling the present and preparing for the future. By adopting this approach, they deliver to the business one

governed, federated, semantically aware data layer that AI can operate on directly across every workflow, at every level of autonomy, and without rebuilding the infrastructure every time a new use case appears.

That's what an AI access and control plane actually is. And the decision about how to build it is one of the few architecture choices that's genuinely hard — and costly — to undo later.