CDATA

Whitepaper

The 25% Accuracy Gap:

MCP Provider Performance Across Enterprise Workloads

Results from 378 queries across CRM, project management, data warehouse, and ERP platforms

Why This Benchmark Exists

MCP (Model Context Protocol) is becoming the default interface between AI agents and enterprise software. As organizations move from chatbot-style copilots toward autonomous agents that read, write, and act on live business data, MCP providers serve as the translation layer: they take a natural language query, convert it into the right API calls, and return structured results.

The promise is straightforward. A sales lead asks an AI agent to pull pipeline data; the agent queries the CRM through an MCP provider and returns the answer. A project manager asks what's in the sprint backlog; the agent queries the project management platform and surfaces the relevant issues. No manual filtering, no context-switching between tools.

But how well does this actually work? Most evaluations of MCP providers focus on connection coverage—how many platforms they support—rather than accuracy: does the provider return the right data for a given query? That's a meaningful gap. An MCP provider that connects to your CRM but misinterprets "deals closing this quarter" is worse than no connection at all, because the wrong answer looks like the right one.

This benchmark attempts to fill that gap. We tested five MCP providers across four enterprise platforms with 378 real-world queries and measured a simple thing: did the provider return correct results, or didn't it?

What We Tested

We evaluated five MCP providers representing the major architectural approaches in the market: CData Connect AI (relational/semantic layer), a unified API provider, a CRM-native MCP server, an MCP gateway, and an iPaaS-based provider. Each was tested against four enterprise platforms: CRM, project management, data warehouse, and ERP.

Each provider executed 16 standardized prompts per platform, ranging from simple lookups to multi-filter queries to write operations. That produced 378 total test runs. An evaluator scored each response against pre-established ground truth—either the MCP provider returned correct data, or it didn't. There was no partial credit.

The prompts were designed to reflect actual enterprise work:

- **"Tell me how many deals I have set to close this quarter"** — requires fiscal calendar awareness and relative date calculation
- **"List all issues assigned to Alex Johnson that are in 'To Do' status"** — requires multi-filter logic across different fields
- **"Find all issues in sprint 112, sorted by highest priority"** — requires sprint navigation and semantic understanding of text-based priority enumerations
- **"Change the deal 'Acme Corp Deal' to the stage contract sent"** — requires workflow validation for write operations
- **"Find all orders created in the last 30 days"** — requires relative date calculation and correct field mapping

These aren't edge cases. They're the kinds of questions sales teams, project managers, and finance analysts ask every day.

Overall Results

Most MCP providers returned incorrect results 15–42% of the time. That translates to roughly one error every two to three queries—a failure rate that's manageable for a suggestion engine but untenable for an autonomous agent.

Platform	CData Accuracy	Other Providers	CData Gap
CRM	100%	75%-100%	Consistent
Project Mgmt	94%	45%-50%	+45–50 pp
Data Warehouse	100%	75%	+25 pp
ERP	100%	20%	+80 pp
Overall	98.5%	65%-75%	+25 pp

CData Connect AI achieved 98.5% accuracy (67 of 68 correct responses). The other providers ranged from 65% to 75%. The 25+ percentage point gap was consistent across platforms, though the specifics varied.

What stood out as much as the aggregate scores was the variance. CData maintained roughly the same accuracy everywhere. Other providers swung widely: MCP Gateway dropped from 95% on CRM t o 50% on project management. The unified API provider fell from 100% to 50%. The iPaaS provider ranged from 75% down to 45%. For organizations deploying AI across multiple systems, that inconsistency is as much of a problem as the raw accuracy numbers.

Where Providers Break Down

The failures weren't random. They clustered into four recurring patterns.

Relative Date Logic

Fifteen failures involved queries with relative dates. "Deals closing this quarter" and "orders created in the last 30 days" require the provider to calculate date ranges from context. The iPaaS provider consistently asked for clarification on what "this quarter" means, even though Q4 2025 was unambiguous. Other providers used the wrong date field—pulling ORDER_D ATE when the query specified CREATED _DATE—or omitt ed date filters entirely, returning datasets spanning multiple years.

For a sales executive reviewing pipeline, this means either getting no results or getting a data dump that requires manual filtering—exactly the work the AI was supposed to eliminate.

Multi-Filter Queries

Twelve failures involved queries requiring two or more simultaneous filters. "Issues in backlog for project Acme Project" requires matching both project ID and status. The MCP gateway and unified API providers returned all 50 project issues instead of the 6 in backlog—they applied the project filter but dropped the status condition entirely.

The practical result: project managers sort through irrelevant tickets manually, spending time on exactly the kind of busywork the tool was meant to handle.

Semantic Interpretation

Eight failures came from priority and sorting logic. Project management platforms typically use text-based priority values—Highest, High, Medium, Low, Lowest—rather than numbers. When asked for "Highest priority" issues, providers either returned "High" priority items (close but wrong), mixed all priority levels together, or failed the query outright. Even CData got this wrong once, returning "High" instead of "Highest" —its only error across all platforms.

The downstream effect is that engineering teams work on the wrong things. Issues that should be addressed immediately get buried in a mixed-priority list.

Write Operations

Seven write operations failed. Changing deal stages, updating order statuses, and moving issues between workflow states all require understanding platform-specific validation rules—not just what data to send, but what transitions are allowed.
The CRM-native MCP server failed every deal stage update despite having direct platform access. The iPaaS provider showed inconsistent behavior: failing on a first attempt, then succeeding on retry, suggesting flaky validation handling.

When write operations fail, business processes stall. Deals don't progress.
Statuses don't update. The autonomous agent creates a queue of manual follow-ups.

Why Architecture Matters

These failure patterns aren't quirks to be patched. They reflect a fundamental architectural divide in how MCP providers work.

Most providers translate natural language queries directly into REST API calls.
This works fine for simple lookups—"get contact by ID"—but falls apart when the query requires any interpretation: relative date math, multi-condition filtering, understanding that "Highest" is a specific enumeration value and not a synonym for "High," or knowing which deal stage transitions are valid in a given CRM workflow.

CData's approach uses a relational abstraction layer with a semantic model that understands entity relationships, business logic, and platform conventions. That's what enables it to handle "this quarter" without asking for clarification, apply multiple filters correctly, and validate write operations against workflow rules.

The ERP results illustrate this clearly. The native ERP MCP server—built by the ERP vendor's own engineers—failed completely, returning "Record 'account' was not found" on simple lookups. The iPaaS managed 40% accuracy but was wildly inconsistent largely due to poorly constructed queries. CData, working through its connector infrastructure, achieved 100%. Having platform access isn't sufficient. The abstraction layer between the query and the API call is where accuracy is won or lost.

What This Means for AI Deployment

The accuracy threshold depends on the use case. For copilot-style tools where a human reviews every suggestion before acting, 75% accuracy is probably workable —annoying, but workable. The human catches the errors.

But the industry is moving toward autonomous agents: AI that updates CRM records, triggers workflows, modifies financial data, and acts without a human in the loop. At 75% accuracy, that agent fails one out of every four actions. Date logic errors waste time. Multi-filter failures bury important items. Semantic mistakes cause teams to work on the wrong priorities. Write operation failures block business processes. And importantly, that inaccuracy compounds: 75% accuracy across a 5-step process ends in less than 24% of processes succeeding. **75% accuracy just turned into a 75% failure rate.**

For autonomous agents to be viable, MCP providers need to clear roughly 98% accuracy. Below that, the agent creates more cleanup work than it saves.

The architectural patterns that get there are now fairly clear: a semantic layer that interprets business logic rather than just translating syntax, an abstraction layer that can construct sophisticated queries, and mature connectors that understand platform-specific conventions. Organizations evaluating MCP providers should test with their own data and queries, but the benchmark results suggest that architectural approach is the strongest predictor of accuracy.

About CData

CData provides connectivity, context, and governance for AI-to-data interactions within organizations. The platform offers live access and data replication across 350+ sources, with semantic intelligence designed to improve the accuracy of AI workloads. CData supports AI infrastructure for Anthropic, Databricks, Microsoft, Google, Palantir, and more than 10,000 customers.

Disclaimer

This benchmark was conducted internally by CData Software, Inc. (""CData"") to provide accuracy data for the MCP ecosystem. All testing, evaluation, and analysis were performed by CData employees. Results have not been independently verified.

Key Limitations:

- Testing was conducted in Q4 2025 using specific configurations described in the Methodology section
- Sample sizes varied across MCP servers based on platform support capabilities
- Binary (all-or-nothing) scoring was used; partial correct responses were counted as failures
- The ""Efficiency Score"" is a CData-developed metric, not an industry standard
- Test complexity profiles differed by platform
- Results reflect controlled test conditions and may not represent performance in all production environments

Performance Variability: Actual performance may vary significantly based on deployment configuration, data characteristics, network conditions, query patterns, LLM model versions, API changes, and platform-specific factors. The platforms and MCP servers tested may have been updated since testing was conducted.

Not Professional Advice: This report is provided for informational purposes only and does not constitute professional, legal, or technical advice. Organizations should conduct their own independent testing using data and queries representative of their specific use cases before making purchasing or implementation.