

AI context & governance

Ground every AI agent on governed, auditable definitions.

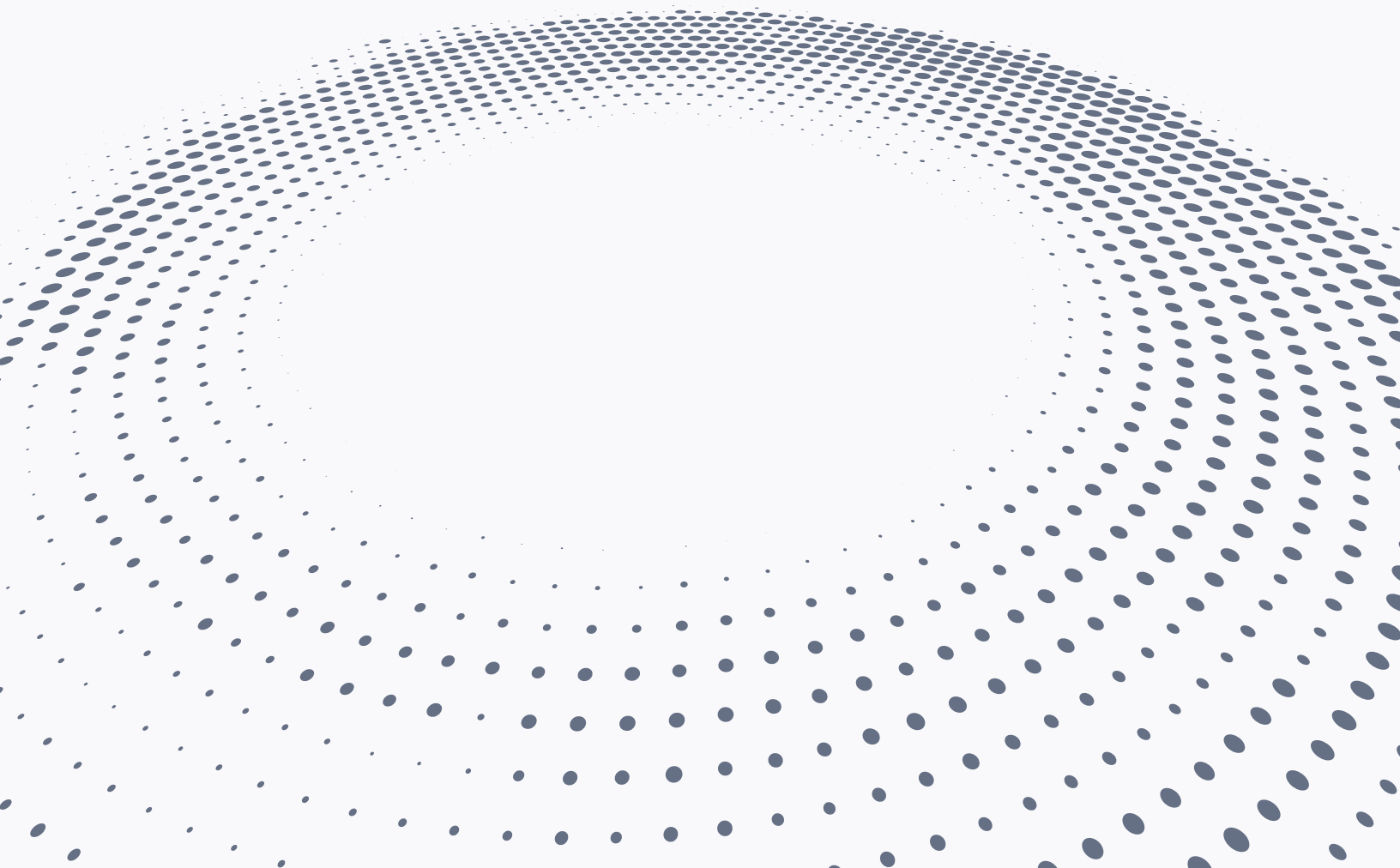


Table of Contents

Why prompt engineering stops heret	03
How MetaKarta serves compiled context	04
The proof to ask for	05
The metric that moves	05
Where this leaves the AI program	06

AI queries against raw schema produce inconsistent answers, and the plausible-looking wrong ones do the most damage, because those are the ones people act on.

The model infers what "ARR" means from column names, table structures, and whatever the prompt happens to carry, so the same question can land on the right definition one day and apply gross revenue logic the next where the correct definition uses net.

That inconsistency has a structural cause: the governed definition exists nowhere the model can read it, so the model closes the gap by guessing.

Text-to-SQL breaks on metric fragmentation for the same reason. When "revenue" is defined differently in three places, the SQL an agent writes depends on which definition it happened to pick up. Identical questions return different numbers on different days, and the architecture offers no way to trace which definition was applied.

Why prompt engineering stops here

Prompt engineering is the common first response. Add the context to the system prompt, describe the metric definitions inline, tell the model what "ARR" means every time you ask.

It stops scaling quickly. A large enterprise has thousands of governed metrics, hundreds of data sources, and agents querying across all of them, so maintaining prompt context for every possible query is operationally out of reach.

The audit problem is worse. When a regulator asks how the AI produced a specific output, a description that lived in a system prompt at query time is gone by the time the question arrives.

SOX, GDPR, BCBS 239, CSRD, and the emerging AI-specific frameworks all require data provenance: a path from an AI output back through the metric definition, through the lineage, to the source data. That path has to be persistent, attributable, and auditable at the infrastructure level.

How MetaKarta serves compiled context

MetaKarta MCP is a single server, built on the industry-standard Model Context Protocol, carrying two toolsets. Metadata Management Tools deliver lineage, catalog, and governance metadata. Semantic Hub Tools deliver compiled context and deterministic SQL.

Both read from the same shared metadata repository as every other platform capability, so the AI answer and the BI answer come from one source.



Compiled definitions.

When an agent queries "ARR," it resolves against the governed definition in the shared metadata repository, the same definition Semantic Hub compiled into Snowflake, Power BI, and Tableau.



Provenance.

Parser-based, column-level lineage carries the path from an output back through the definition, through the transformations, to the source data. Every response returns the metadata evidence that produced it.



Catalog context.

Classifications, data domains, ownership, access policies, and documentation reach the agent through the same interface, so the model reads what the catalog holds about a dataset.



Governance policy.

Classification labels, access controls, and sensitivity designations travel with the context, permissioned to the requesting user.

The proof to ask for

Context Sandbox validates the grounding before any query runs, so this part of the demo works as a test rather than a claim:

- ✓ Ask the same business question of the agent and of a governed BI report.
- ✓ Open the metadata evidence returned with the agent's answer and follow it back to the definition.
- ✓ Follow the column-level lineage from that definition to the source data.
- ✓ Change the definition through a stewardship workflow, ask again, and watch the answer move with it.

Test the claim yourself, on your data, with the grounding visible.

The metric that moves



Answer variance.

Ask the same question ten times across a week and count the distinct answers. That count measures how much the agent is inferring, and it falls as the agent resolves against a compiled definition.



Time to an audit answer.

How long it takes today to reconstruct how a specific AI output was produced on a specific date. With version history and column-level lineage in the shared metadata repository, the audit answer is a query.

Where this leaves the AI program

The program stops depending on how well each prompt was written. Agents resolve against definitions that are governed, versioned, and compiled into the same systems the business already reports from, so the AI answer and the BI answer agree because one definition produced both.

That makes the outputs defensible. A regulator's question resolves against version history and lineage, an executive's against the same definition the board deck used, and the AI initiative carries an audit trail either way.

It holds as the estate changes, too. A definition approved through a stewardship workflow is available to agents as soon as it clears, and lineage updates the provenance chain alongside it. Request a demo to see an agent answer resolve against a compiled definition, return the metadata evidence behind it, and trace column-level to source.

[Request Demo](#)

About Meta Integration Technology

Meta Integration Technology, Inc. (MITI) is a pioneer in metadata integration, semantic modeling, and data lineage with decades of experience serving global enterprises and technology providers. MITI's technology is embedded in products from leading data and analytics vendors and powers MetaKarta, its enterprise platform for metadata-driven data management. MetaKarta delivers data lineage, data cataloging, data governance, and semantic enforcement through a single integrated system, connecting the full enterprise data estate across 400+ native connectors, from legacy to modern and on-premises to the cloud.

650 Castro Street, Suite 120-431, Mountain View, California 94041-2055 USA
Phone: 650.429.8853

info@metakarta.com
www.metakarta.com