METAKARTA

Solution Brief

How far does one column name really reach?

Data architects and engineers already know the estate better than any tool. MetaKarta puts that knowledge in one shared metadata repository, computes lineage from the code itself, and shows the blast radius of a change before it deploys.

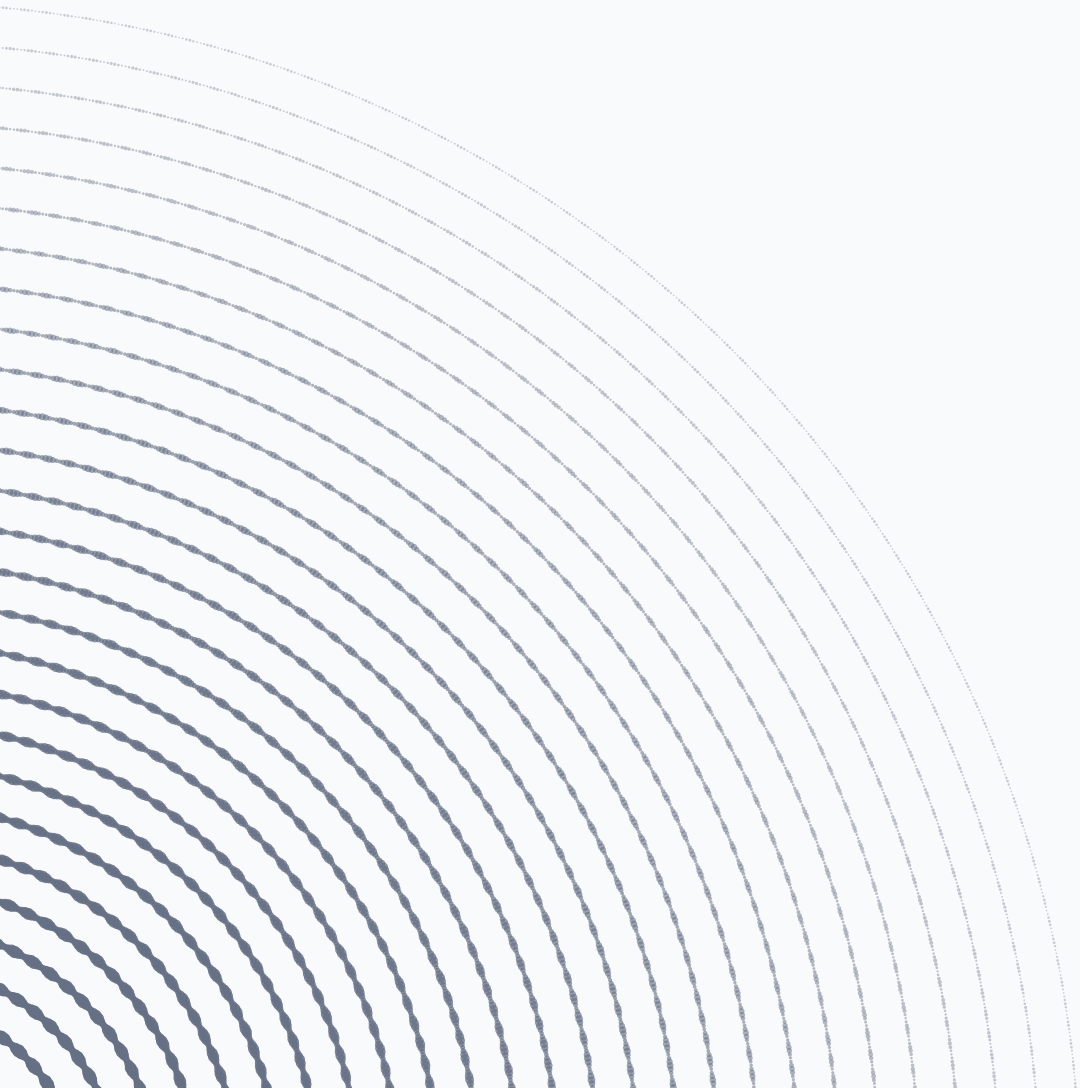


Table of Contents

What changes with MetaKarta 03

How it works 04

See it on your own estate 05

Built on nearly 30 years of metadata engineering 06

Most estates run lineage, catalog, governance, and semantics in four separate tools, joined by integration code someone on your team wrote and still maintains. The tools disagree, so you carry the reconciled view in your head.

That holds until a schema change ships on a Friday. Three gaps decide whether you hear about it first or the ticket queue does:

Observation.

Log-based lineage sees only the jobs that ran. A quarter-end stored procedure or a dormant SSIS package stays off the map until it breaks.

Propagation.

A renamed column cascades through SQL views, Informatica mappings, and Power BI measures, and impact analysis means weeks of manual cross-referencing.

Reconciliation.

Each tool keeps its own copy of the metadata, so the lineage graph, the catalog entry, and the policy for the same column drift apart.

What changes with MetaKarta

Lineage by design, not by observation.

MetaKarta parses SQL, stored procedures, ETL mappings, and BI models across 400+ native connectors. A job that hasn't run in six months is mapped the same way as one that ran last night.

Know the blast radius before the first ticket lands.

Propose a change and MetaKarta traces every downstream column, view, mapping, and report it touches, at column level. Ask MetaKarta answers "What breaks if I rename this column?" in plain language, with drilldown to the lineage behind the answer.

One repository under every capability.

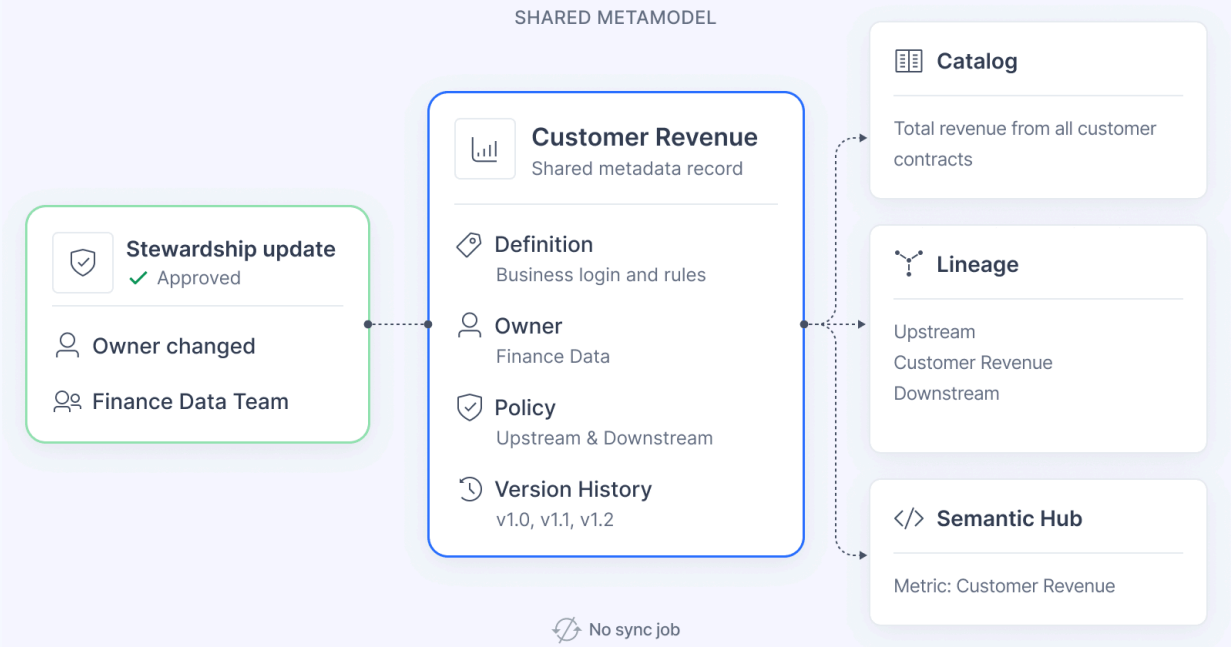
Lineage, catalog, governance, and semantics read from and write to the same shared metadata repository. The catalog entry and the lineage graph describe the same column because they're the same record, and there's no sync job for you to maintain.

Definitions compiled into the systems you run.

Semantic Hub compiles governed definitions into native Snowflake Semantic Views, Databricks Metric Views, and Power BI semantic models. We don't sit in your query path. We've already compiled into your database and tools.

How it works

MetaKarta runs on one shared metadata repository, so lineage, catalog, governance, and semantics describe the same estate.



Parser-based lineage: Column-level lineage computed from SQL, stored procedures, ETL mappings, and BI models, including jobs that haven't run this quarter

- **400+ native connectors**: Metadata harvested directly from cloud warehouses, legacy databases, ETL tools, and BI platforms, on-prem and in the cloud
- **Impact analysis and Ask MetaKarta**: Downstream impact of a proposed change at column level, queryable in natural language through Anthropic, Google, Ollama, OpenAI, or Azure OpenAI
- **Version and configuration management**: Every metadata change logged, attributed, and reversible across the full estate
- **MetaKarta MCP**: Governed lineage, catalog metadata, native artifacts in target systems, and compiled context to agents, with per-user access tokens
- **Semantic Hub**: Governed definitions compiled into native warehouse and BI artifacts, with Git and CI/CD integration

See it on your own estate

Three checks make a useful evaluation of any metadata platform, including ours:

The dormant-job check.

Pick a stored procedure or SSIS package that hasn't run this quarter and ask for its column-level lineage. Lineage computed from code answers it; lineage observed from query logs covers only the jobs that ran.

The blast-radius check.

Name one column you plan to rename. Ask the vendor to list every downstream view, mapping, and BI measure it reaches before the change ships.

The one-record check.

Pick one column and ask to see its lineage, its catalog entry, and the policy that governs it. Then ask how many systems those three answers came from.

Built on nearly 30 years of metadata engineering

MetaKarta's metadata technology runs inside the products of Microsoft Purview, Informatica from Salesforce, IBM, Oracle, and Qlik Talend. MetaKarta brings that engine to your architecture directly, with definitions that stay portable as native artifacts in the tools you already own.

When lineage comes from the code, impact analysis moves from a multi-week project to a check you run before the change deploys. The number to watch is how many schema changes reach production with their blast radius already mapped.

Get in touch to learn more about how parser-based lineage maps a schema change before it deploys.

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