

How to turn a query into an audit answer

Use version history, policy, and lineage to answer what a definition meant, who changed it, and where it flowed on any date.

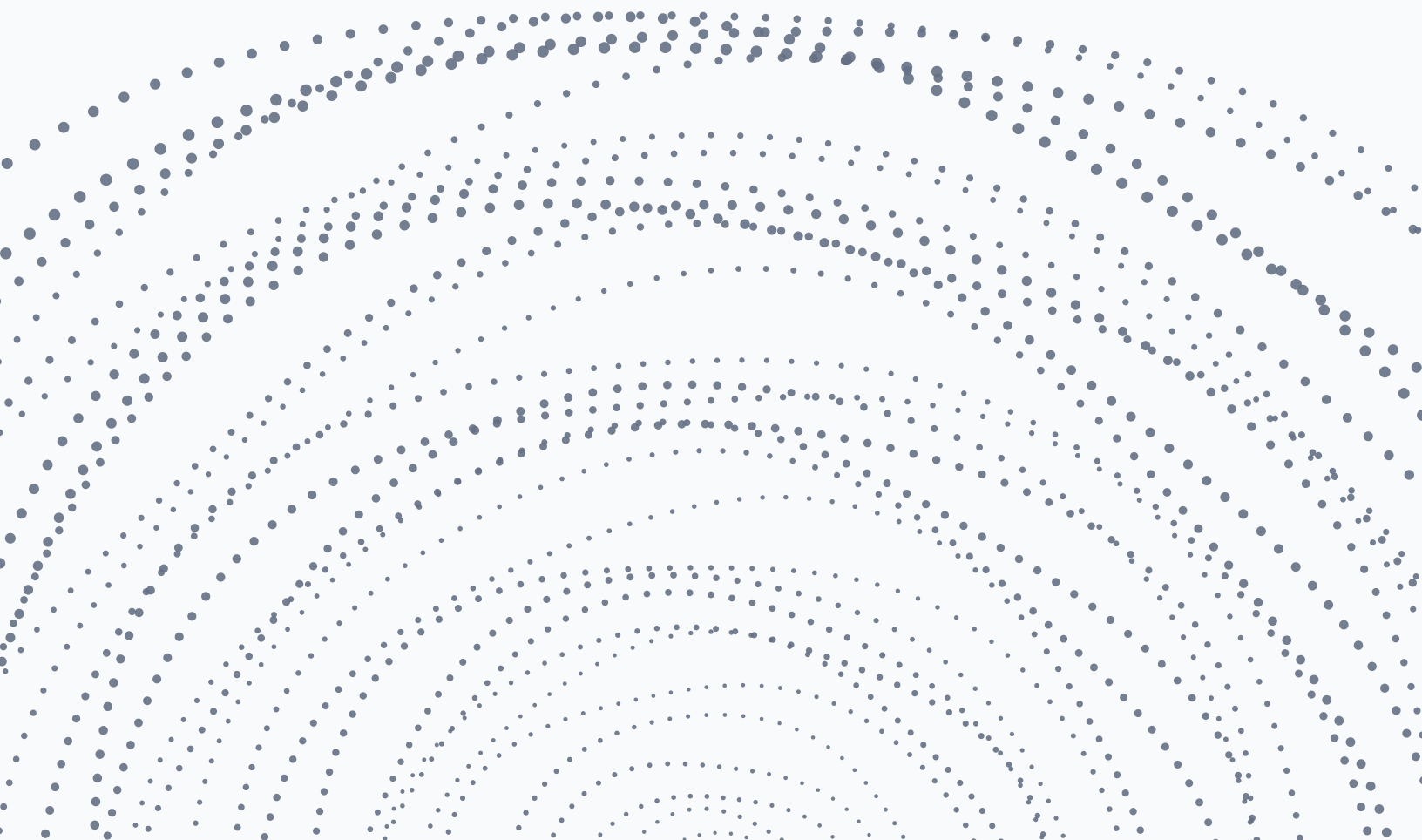


Table of Contents

The record was never kept in the first place	03
The questions got more specific	04
Three things have to be true at once	04
One number, three questions	05
Policy has to hold where the data is used	06
The enforced definition is the audited definition	07
What changes about the cycle	08
Where to start	09

The auditor asks three questions about one number.

What did this field mean on the transaction date? How was it calculated? Who approved the calculation, and when?

Every question is reasonable. Every question is answerable in principle. Together they start a project that runs for weeks, pulls senior people off their roadmap, and ends with a reconstruction assembled from email threads, a wiki page somebody updated in 2024, and one analyst's memory.

Then the cycle closes, the reconstruction gets filed, and the next audit starts the same archaeology from the beginning.

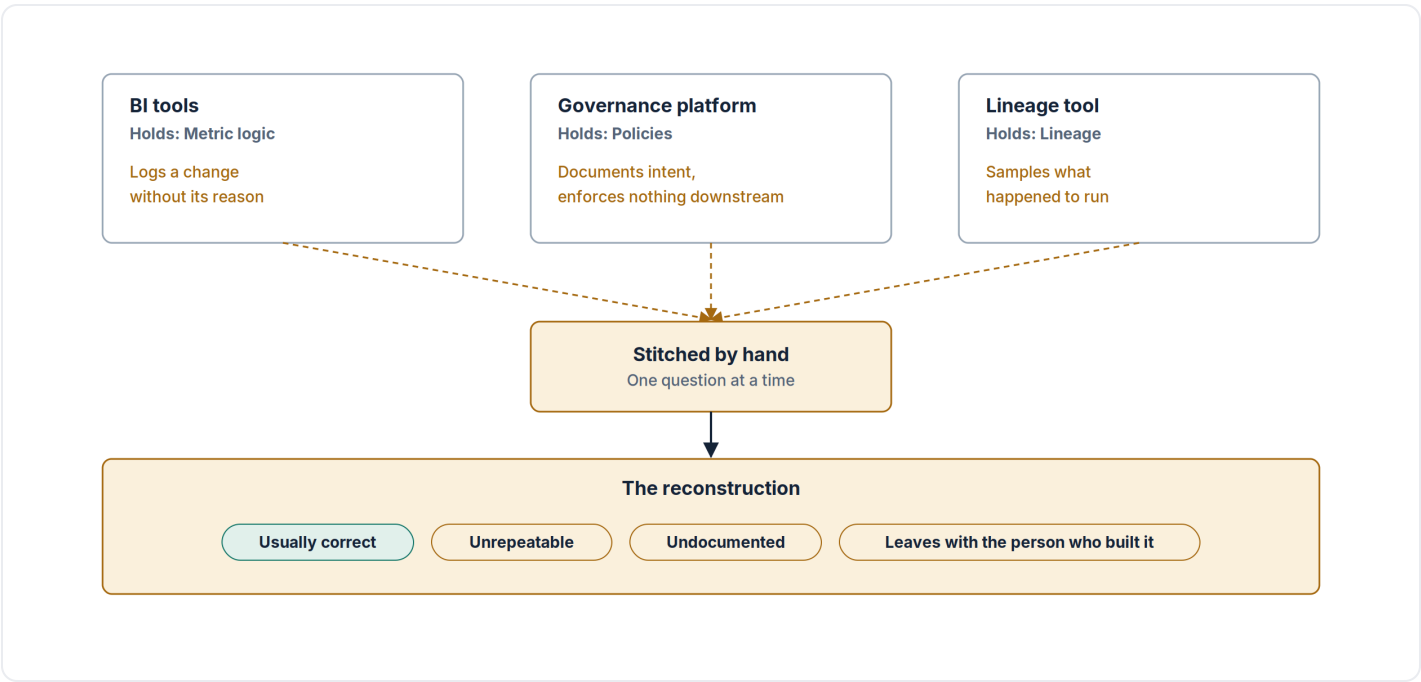
The record was never kept in the first place

Audit prep is a project because the estate holds no continuous record of what its definitions meant over time.

Metric logic lives inside BI tools that log a change without logging the reason. Policies live in a governance platform that documents intent and enforces nothing downstream. Lineage lives in a third system that samples what happened to run.

That's the assembled metadata solution. Every tool in it owns its own metadata, so answering one question means pulling threads across all of them and stitching the result by hand.

The reconstruction is usually correct. It's also unrepeatable, undocumented, and gone the moment the person who assembled it changes jobs.



The questions got more specific

Regulators used to ask whether a control existed. Increasingly they ask what a specific number meant on a specific day.

SOX 302 puts an executive signature on the accuracy of a reported figure. BCBS 239 asks banks to demonstrate that risk data can be aggregated accurately and traced.

CSRD and ISSB extend the same expectation to sustainability figures, which most organizations compute from data that never went through a financial control. Emerging AI governance mandates add provenance requirements for outputs that were generated rather than reported.

The common thread is a shift from policy to evidence. A documented governance program answers the old question. The new one wants the trace.

Three things have to be true at once

The audit answer is a query. That holds when the estate has three properties at the same time, which is what defensible governance looks like as an architecture.



Every definition has a version history.

MetaKarta places the entire metadata estate under full version and configuration management: lineage, catalog entries, governance policies, and semantic definitions together. Every change is logged, attributed, timestamped, and reversible, so what a metric meant on any past date is a state the system already holds.



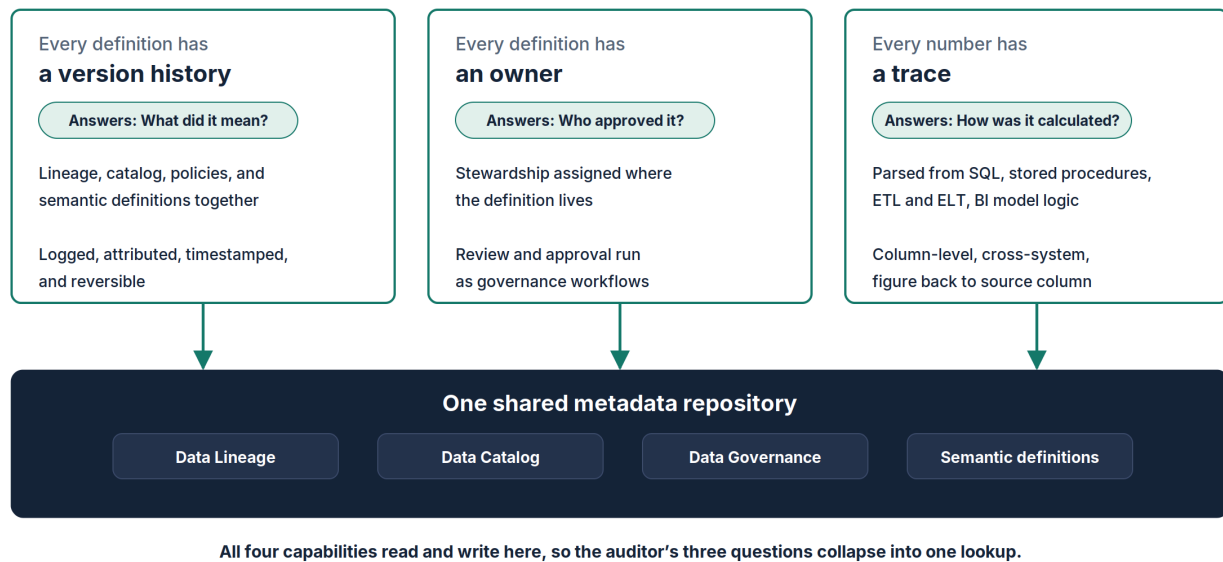
Every definition has an owner.

Stewardship is assigned in the system that holds the definition, with review and approval cycles running as governance workflows. The name attached to a change is recorded when the change happens, which is the only time anyone reliably knows it.



Every number has a trace.

MetaKarta computes lineage by parsing the code artifacts that define the transformation chain: SQL, stored procedures, ETL and ELT mappings, BI model logic. Column-level and cross-system, from a figure in a report back to the source column and the transformation that produced it. Lineage by design, not by observation.



All three hold at once because all four capabilities read from and write to one shared metadata repository. That's the structural reason the three questions the auditor asked collapse into one lookup.

One number, three questions

Take the revenue figure from a Q2 filing and run the auditor's questions against it.



What did it mean on the reporting date.

Open the metric's version history, select the date, and read the definition that was live: the expression, the dimensions it was valid against, and the physical columns it read.



How was it calculated.

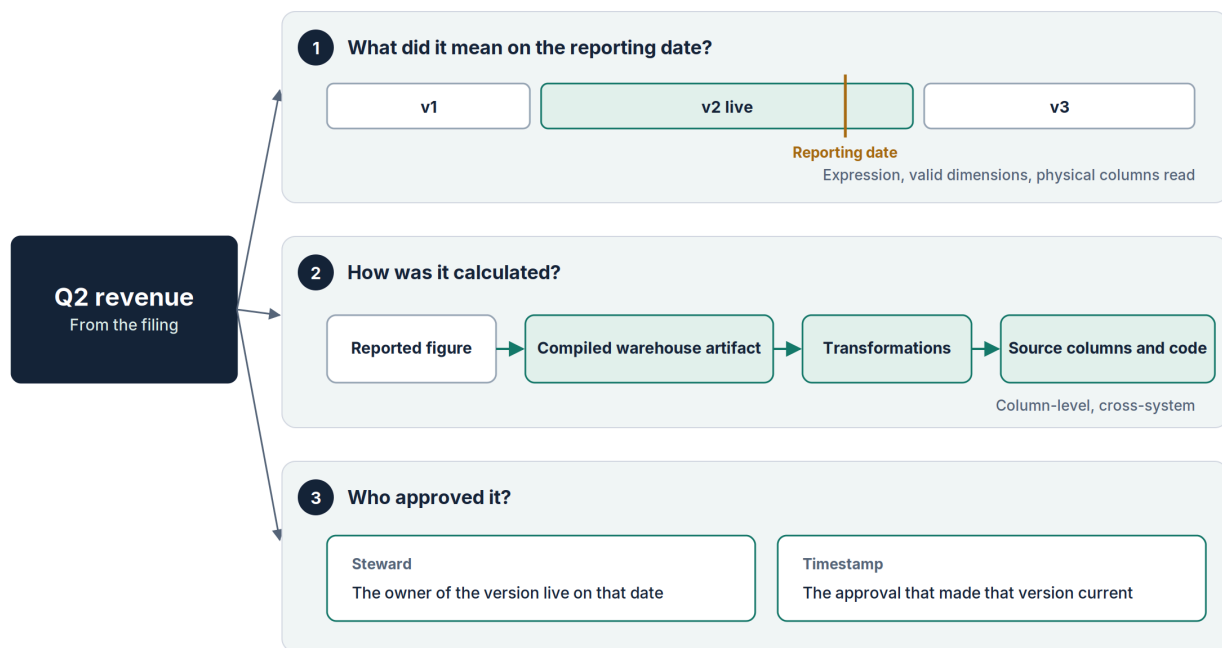
The trace runs from the reported figure back through the compiled artifact in the warehouse, through the transformations that produced its inputs, down to the source columns and the code that transformed them.



Who approved it.

The version live on that date carries the steward who owned it and the timestamp of the approval that made it current.

Three lookups against one repository, in about the time it takes to describe them.



Policy has to hold where the data is used

Version history explains what a definition meant. Auditors also ask who could see it.

Row-level security and column masking are defined once in the governed model and compiled into the native security engine of every target the model deploys to. The database enforces it with its own engine. The BI tool enforces it with its own engine.

Policy applied where the data is used, versioned where it changed.

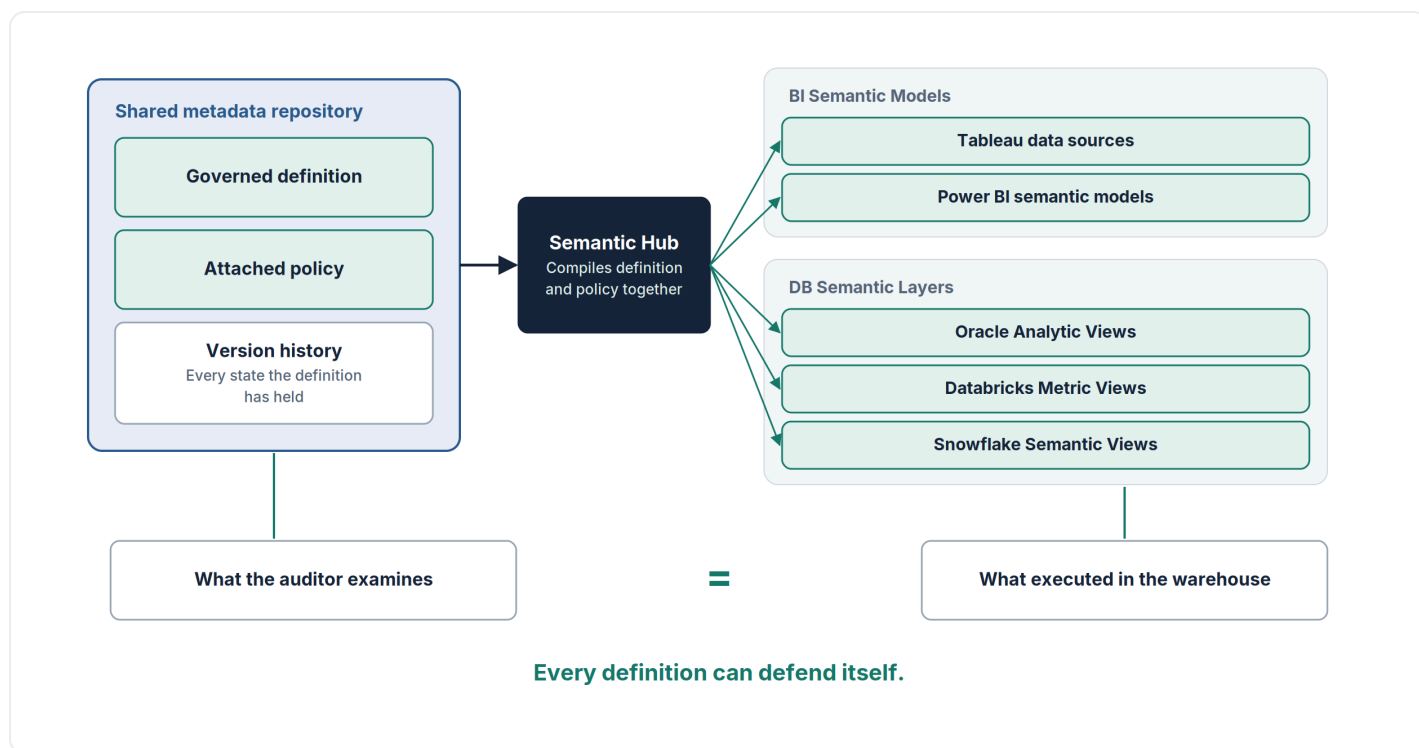
That second half is what makes it auditable. A policy change is a recorded event with a date, an author, and a diff, so the question of who could see regional revenue in Q2 has the same shape of answer as the question of what regional revenue meant in Q2.

The enforced definition is the audited definition

Here's the part that closes the gap between a governance program and an audit finding.

Semantic Hub compiles governed definitions into the native artifact each target platform requires: Snowflake Semantic Views, Databricks Metric Views, Oracle Analytic Views, Power BI semantic models, Tableau data sources. The policy attached to the definition compiles alongside it.

So the definition an auditor examines in the repository is the definition that executed in the warehouse. One artifact, produced by one compilation, from one governed source, with a version history behind it.

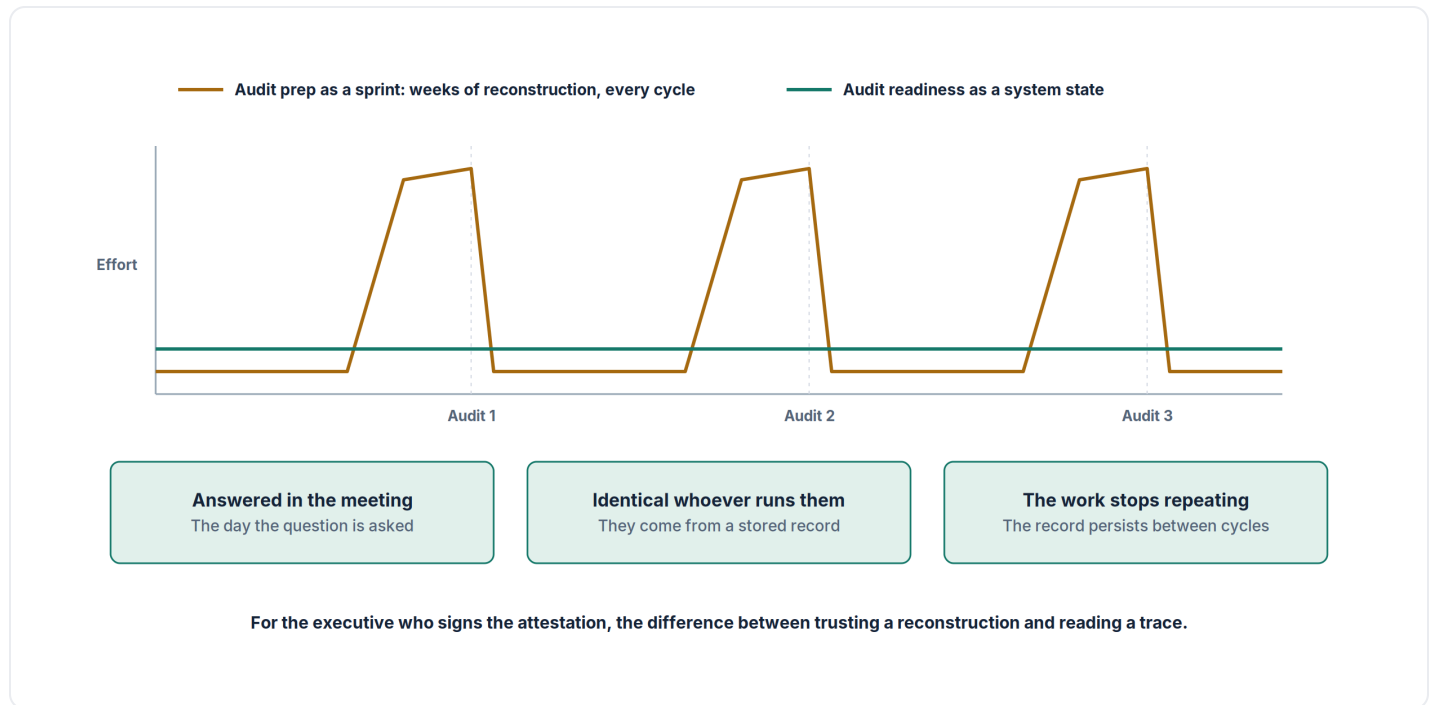


Documentation describes what should be true. Compilation makes it true.

Every definition can defend itself.

What changes about the cycle

Audit readiness shifts from a sprint before the auditor arrives to a permanent system state.



The practical difference shows up in three places. The questions get answered in the meeting rather than weeks after it. The answers are identical whoever runs them, because they come from a stored record, and the work stops repeating every cycle because the record persists between them.

For the executive who signs the attestation, that's the difference between trusting a reconstruction and reading a trace.

Where to start

Take one number from your last board deck and ask your team the auditor's three questions about it. What it meant on the reporting date, how it was calculated, and who approved that calculation.

Time the answer. Whatever that number is, it's your current audit cost per question, and an audit is a lot of questions.

MetaKarta v12 is available now. Request a demo to learn how to use it to answer audit questions on demand.

[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