

BI metric consistency

Define a metric once. Compile models and meaning into every BI tool.

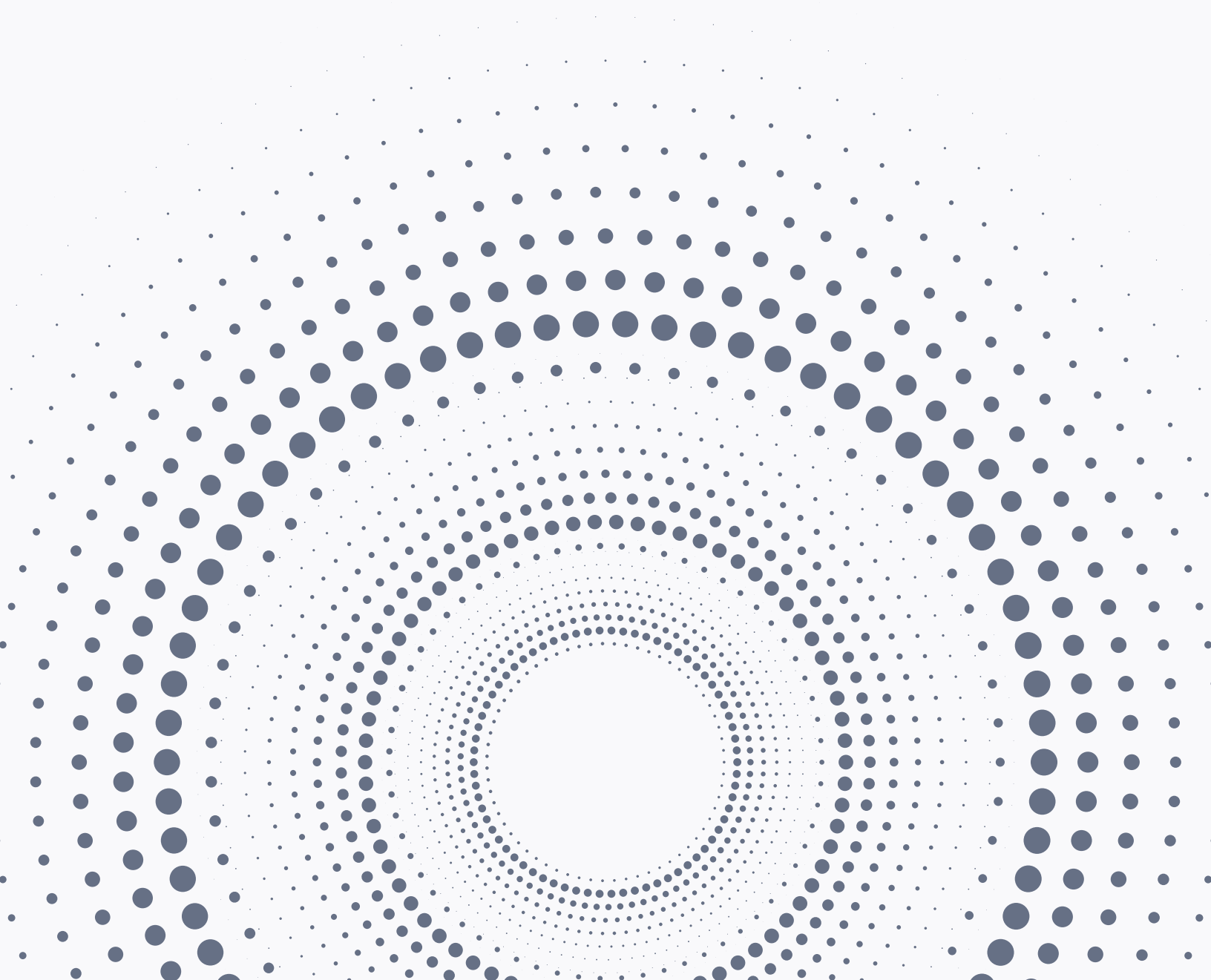


Table of Contents

Where documentation and middleware stop	03
Write once, compile anywhere.	04
How it's proven	04
The metric that moves	05
Where this leaves the estate	05

The executive complaint is always the same: "the numbers don't match."

Finance's ARR doesn't match Sales'. The board deck pulls from Power BI and shows a third number. Everyone is computing revenue, and every system computes it differently, so the meeting that was supposed to review performance turns into a debate about which dashboard is right.

Every BI platform stores metric logic inside itself. Power BI gets the calculation in DAX, embedded in the PBIX file. Tableau gets it again as a calculated field. When Snowflake or Databricks surfaces the same metric, it's defined a third time.

Each definition drifts a little. Different filters, different date logic, different treatment of edge cases, usually coded by different people at different times against documentation that went stale months earlier.

The result is a portfolio of BI tools that each produce their own version of the same metric. Each version is defensible on its own terms, and the versions disagree with each other.

Analytics teams put roughly 20% of every sprint into reconciling numbers instead of building. That number is a property of the architecture: when metric logic lives inside each tool, fragmentation is the default outcome.

Where documentation and middleware stop

The standard response is to write the definition down somewhere, in a Confluence page or a shared spreadsheet or a wiki entry that says "revenue is defined as X." Documentation describes what should be true. Compilation makes it true. The next engineer who builds a dashboard reads the page, interprets it a little differently, and ships a fourth version.

Some teams reach for a semantic layer. That middleware sits between the BI tools and the data sources and translates at query time. The BI tools keep their own embedded logic underneath, so the fragmentation survives the purchase, and now there's a run-time dependency sitting in the query path.

Under both approaches, every BI tool still holds its own metric definition, and every one of those definitions can drift.

Write once, compile anywhere.

Semantic Hub holds the definition in one governed place, outside the individual BI tools.

A governed metric definition resides in MetaKarta's shared metadata repository: version-controlled, stewardship-approved, traceable to the business rules behind it. Semantic Hub compiles that single definition into the native artifacts of every target platform at once: Snowflake Semantic Views, Tableau logical models, Power BI TMDL, Databricks Metric Views, LookML, and more.

Every BI tool consumes the compiled output. The definition inside Power BI is the same definition as the one inside Tableau, because both were compiled from the same source.

When the definition changes, one governed update compiles and deploys to every target. The team stops filing a ticket to the Power BI owner, a second ticket to the Tableau owner, and then waiting to see whether the two implementations agree.

Data Governance carries attribution and version history for every change: who made it, when, what the justification was, what the previous version said. When an executive asks what "revenue" meant in Q3 of last year, the answer is a query against the version history.

How it's proven

Run the full sequence against two BI targets in a single session:

1. Import and reverse-engineer an existing Power BI semantic model or Tableau data source into a governed definition.
2. Edit that definition once, in the shared metadata repository.
3. Compile to both targets simultaneously and open the native artifact inside each tool.
4. Run the same query in both, and compare the numbers.

The compiled artifact is the proof.

The metric that moves

The number to track is sprint time spent reconciling. Most teams enter this conversation somewhere near 20%. Measure it before the first compiled definition ships and again a quarter later, because the reconciliation work is downstream of tools disagreeing, and compiled definitions give them one governed source to agree on.

Two second-order effects are worth tracking as well. Engineering hours shift from reconciliation to new work. And every definition change carries its governance forward on the next compilation, so the audit trail accumulates on its own.

Where this leaves the estate

Governed metric definitions are defined once in the shared metadata repository. Every target BI tool in the estate receives a compiled version of that one definition, in its own native format, so consistency becomes a property of the architecture. The sprint-by-sprint coordination that used to hold the numbers together is no longer what's holding them together.

That holds as the estate changes. A new BI tool receives the governed definition compiled to its native format on day one, and a migration starts from definitions that are already governed and versioned.

Request a demo to see one governed definition compiled into Power BI and Tableau side by side, and the same query return the same number in both.

[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