Elegant metadata for ELT

Factories and Data systems

Modern data systems process massive amounts of information every day. Much like manufacturing companies, they rely on repeatable steps to transform raw materials into finished products.

The industrial revolution unlocked efficiency through assembly lines and standardized machinery. A single machine could process thousands of identical units, raising throughput and lowering cost.

Data systems face the same challenge and the mass production of analytical assets do seem to share many traits with physical ones. It then seems very fitting that terms such as **AI factories** and products such as **Azure Data factory** have come along. Think about it, if every dataset required a unique process, efficiency would collapse. Imagine Coca-Cola needing a bespoke machine for each bottle it produced. The lesson from the factory floor is clear: scalable systems demand generalization and repeatability.

Figure 1. *"From Factory Floors to Data Factories"*. The similarities of manufacturing floors and producing analytical products in data systems (*AI generated image)*

This is where **metadata-driven, parameterized pipelines** come in — the assembly lines of the modern data world.

The Problem of Hardcoded Pipelines

It's not rare to have data systems that ingest data from 5 source systems each with 50 tables.

Consider the medallion architecture, where data flows from:

- **Bronze** → raw, ingested data
- **Silver** → cleaned, enriched data
- **Gold** → curated, business-ready data

If every source table required its own ingestion pipeline, managing this flow across five systems with fifty tables each would demand 250 unique pipelines. The result is the "spaghetti factory" of data engineering: slow to scale, expensive to maintain, and prone to error. A metadata-driven approach collapses that to a handful of reusable pipelines, parameterized by source and table metadata.

This isn't just elegant — it's the only way to scale in a world of ever-expanding data.If each table, dataset, or transformation requires a separate pipeline or notebook.

Metadata and Parameters: The Machinery of Data Factories

In manufacturing, machines don't change with every unit — they rely on settings, molds, and specifications. In data systems, **metadata and parameters** play the same role.

- **Metadata** describes the structure and rules of your data.
- **Parameters** allow a single pipeline to adapt dynamically based on that metadata.

Instead of building 250 pipelines for 250 tables, we can build **one generalized pipeline** that accepts the table name, system name, and other details as parameters. That pipeline ingests the raw data, applies consistent cleaning rules (e.g., column name normalization), and lands it in the right storage layer.

This design dramatically reduces complexity, accelerates development, and ensures consistency across the enterprise.

Implementation Insights

Whether you use **Azure Data Factory, dbt, Airflow, or another orchestration tool**, the principle is the same:

- Treat pipelines as general-purpose machinery.
- Push variability into metadata and parameters.
- Centralize control with metadata repositories.
- Loop over metadata to dynamically extract, transform, and load only the tables you need.

Almost every modern orchestration tool supports this approach — through parameterized activities, dynamic sources and sinks, and reusable components.

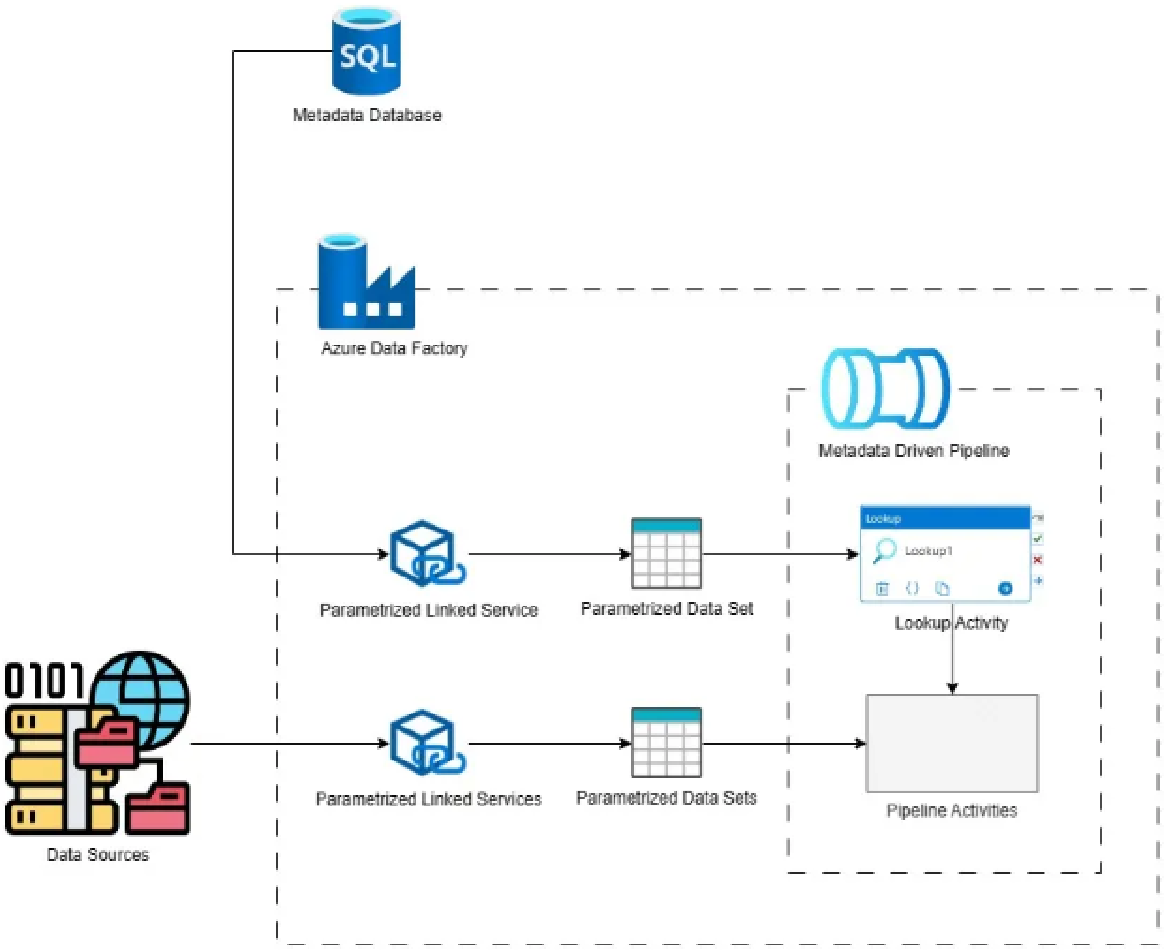

Figure 2. Metadata driven pipelines in Azure Data Factory. Source: What are Metadata-Driven Pipelines? – Systems, Architecture and Systems Architecture

The Future: Mass Production of Data Products

The data world is moving toward mass production of analytical products — dashboards, machine learning features, APIs, and more. Just as factories could not scale without assembly lines, data systems cannot scale without metadata-driven pipelines.

The takeaway is simple:

- Hardcoding pipelines leads to inefficiency and fragility.
- Metadata and parameters unlock scalability, reusability, and speed.

- The organizations that master this pattern will be the ones capable of producing data products at industrial scale.

Just as the industrial revolution was powered by assembly lines, the data revolution will be powered by metadata.

I’d love to hear your experiences. Share your thoughts in the comments, or pass this along to someone who’s wrestling with pipeline sprawl. Let’s keep the conversation going on how to build truly scalable *data factories*.