PROJECT:	**MICROSOFT FABRIC ECOSYSTEM**
SHEET:	**01**
TITLE:	**CONCEPTUAL FRAMEWORK**

The Microsoft Fabric Blueprint

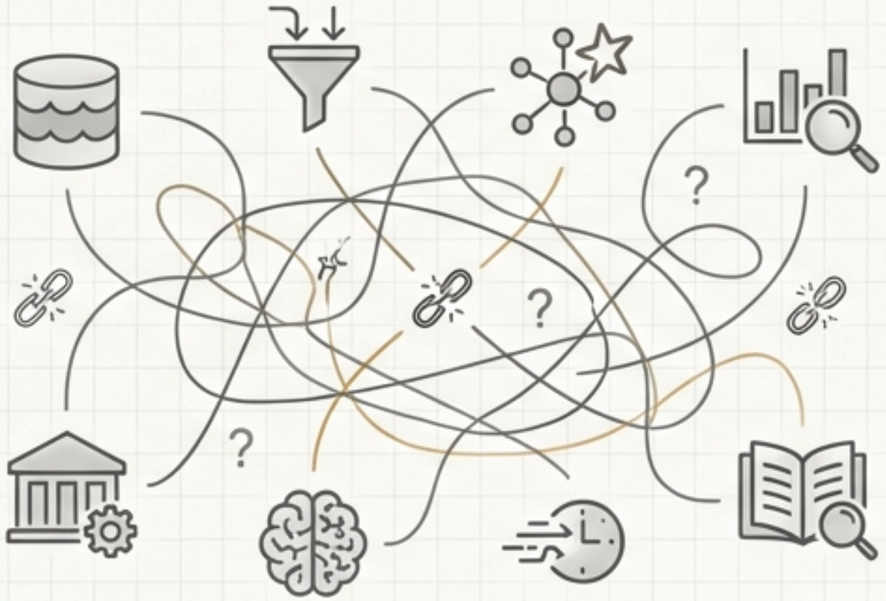
A Structured Guide to the Modern Data Journey

From Scattered Facts to
a Coherent Framework.

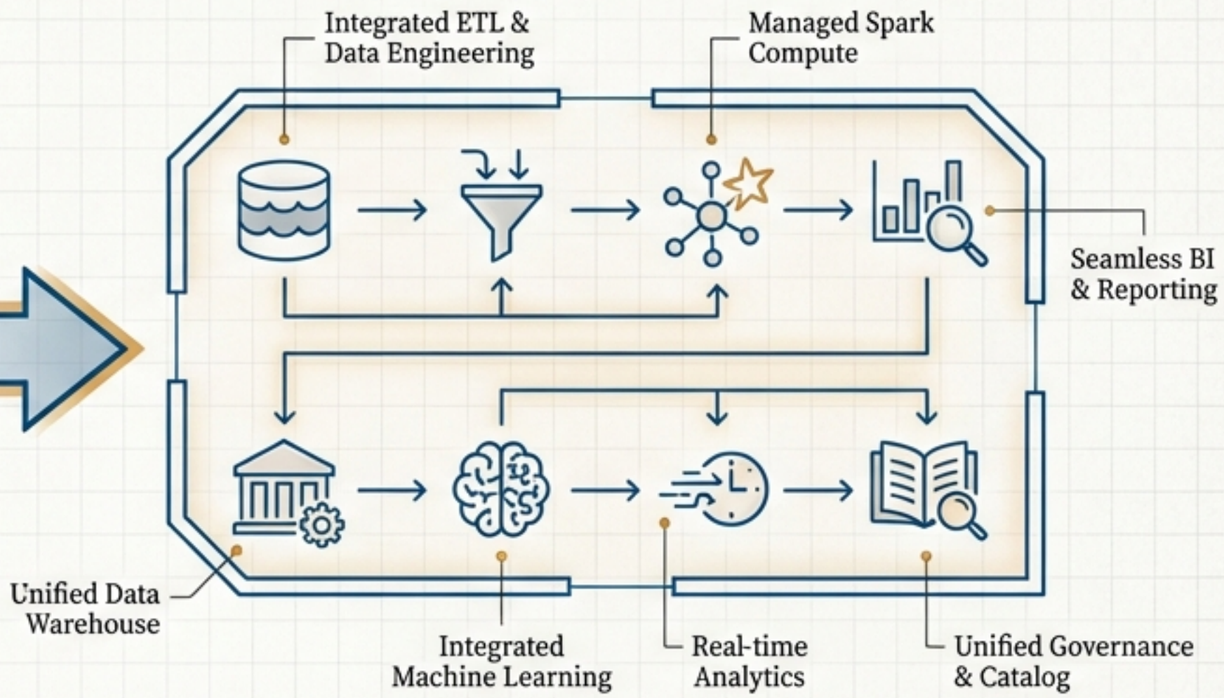
The Foundation: A Single, Integrated Environment for Data

The primary benefit of Microsoft Fabric is providing a single, integrated environment for collaboration on data projects. This eliminates the need to stitch together multiple disparate services.

BEFORE



AFTER: FABRIC



Eliminates

Data duplication across systems.

Enables

Seamless collaboration between data professionals (engineers, analysts, scientists).

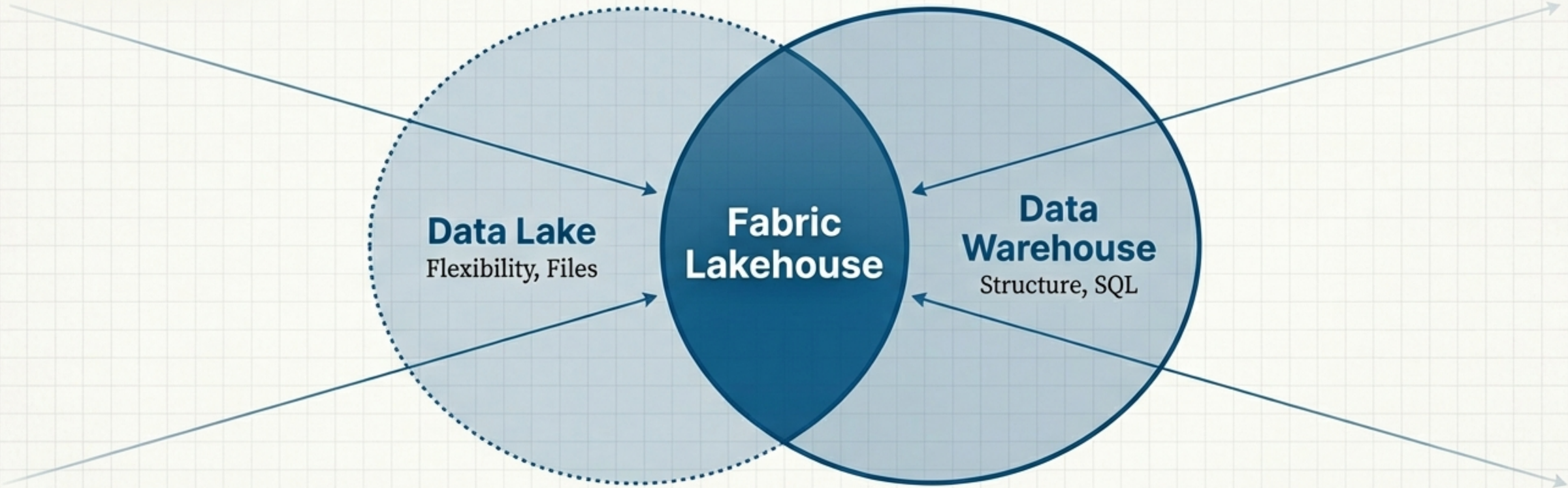
Creates

A unified architecture from ingestion to insight.

At the Heart of Fabric is the Lakehouse

What is a Lakehouse?

An analytical store that combines the file storage flexibility of a data lake with the SQL-based query capabilities of a data warehouse.



Built on OneLake, a single, unified data lake for the entire organization.

Supports both structured and unstructured data.

Provides a single source of truth for all data workloads.

The Default Storage Format is Delta-Parquet

All data in Fabric is stored in the Delta-Parquet format by default.

What is the Delta format?

A relational storage layer for Apache Spark that supports tables based on Parquet files. It adds a transactional log on top of Parquet files, enabling ACID transactions and enhanced reliability.

Why it Matters

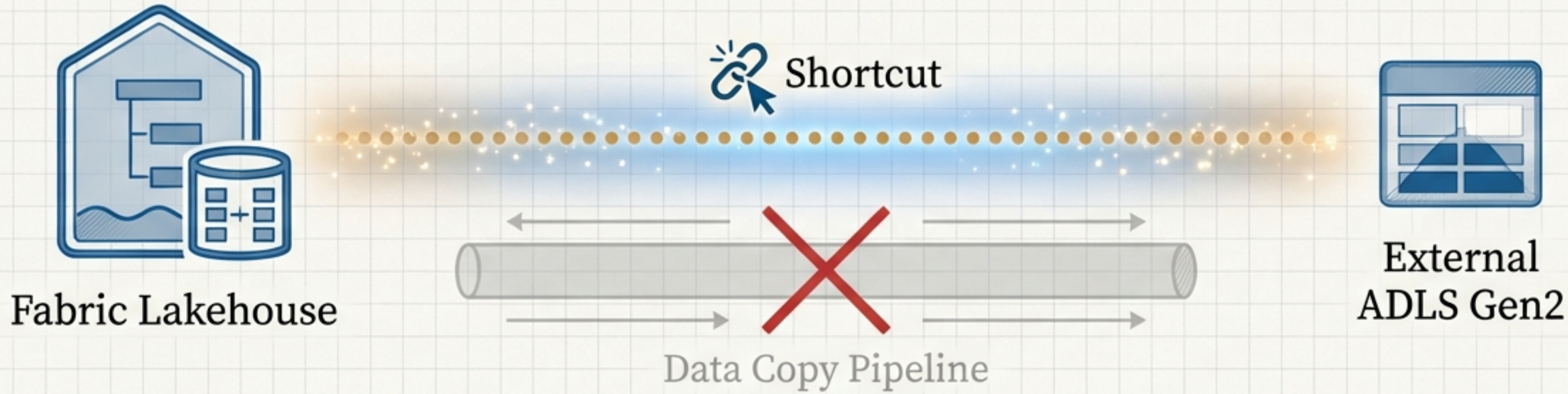
- **Reliability:** Supports ACID transactions for data consistency.
- **Performance:** Optimized for Spark and large-scale analytics.
- **Unified Standard:** Ensures compatibility across all Fabric engines (Spark, SQL).

Practical Application: When writing a Spark dataframe to create a lakehouse table, you must use the `DELTA` format.

Accessing External Data Without Copying: The Role of Shortcuts

The Challenge: How do you include data from an external Azure Data Lake Store Gen2 location in your lakehouse without creating a copy?

The Fabric Solution: Create a Shortcut.

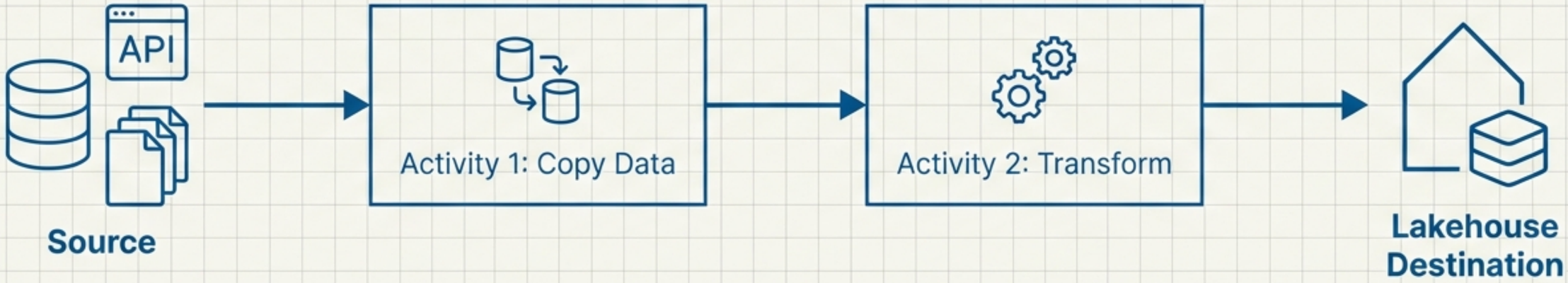


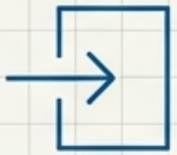
Key Takeaway: Shortcuts act as symbolic links, allowing you to virtualize data and treat external sources as if they were natively inside your lakehouse. This is critical for maintaining a single source of truth and minimizing data movement.

The Pathways In: Ingestion and Orchestration with Data Factory

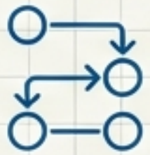
Core Experience: The Data Factory workload in Fabric is used to move and transform data.

What is a Pipeline? A sequence of activities designed to orchestrate a data ingestion or transformation process.





Ingest: Pull data from hundreds of sources.



Orchestrate: Define complex, multi-step workflows.

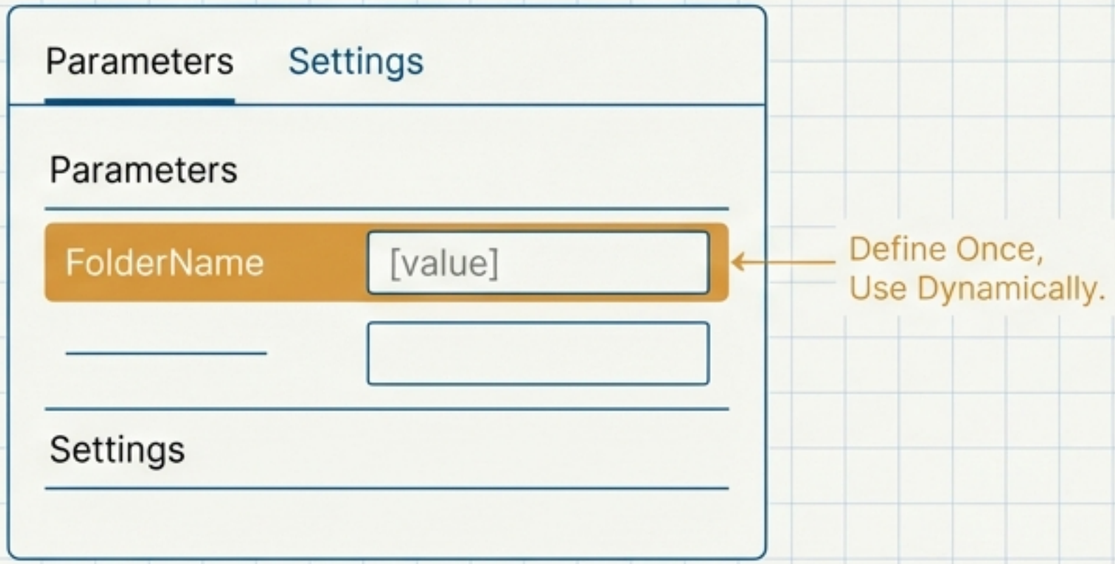
Monitor: Track the execution and performance of data processes.

Building Dynamic and Monitorable Pipelines

Scenario 1: Dynamic Execution

Problem: You need to copy data to a different folder for each pipeline run.

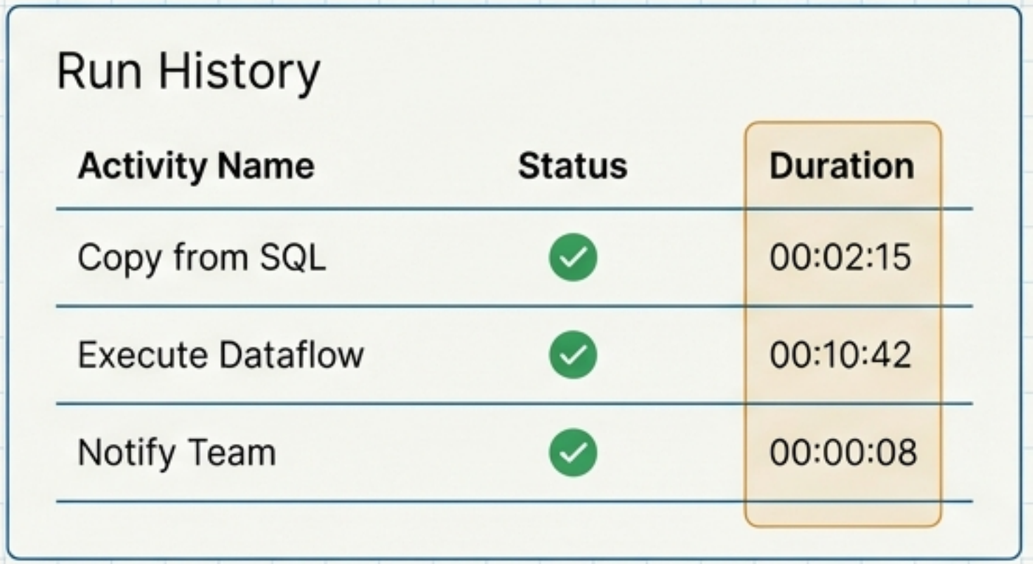
Solution: Add a parameter to the pipeline. Use the parameter to specify the folder name dynamically for each run. This avoids creating multiple static pipelines.



Scenario 2: Performance Monitoring

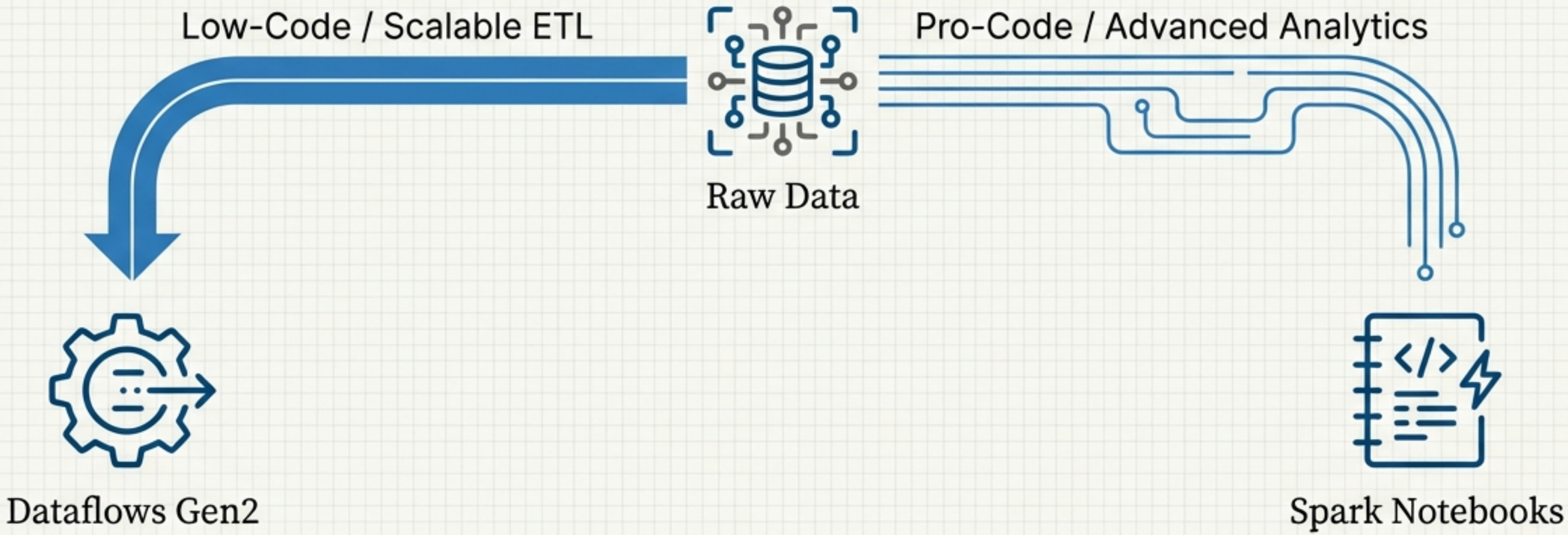
Problem: You need to know how long each individual activity within a completed pipeline took to run.

Solution: View the run details in the pipeline's run history. The history provides a granular breakdown of each activity's duration and status.



The Refinery: Choosing Your Path for Data Transformation

Fabric offers two primary tools for large-scale data transformation, catering to different user preferences and technical requirements.



Which tool is best suited for your transformation needs? The answer depends on your comfort with code and the nature of your analysis. The following slides will detail each path.

The Low-Code Path: Transforming Data with Dataflows Gen2

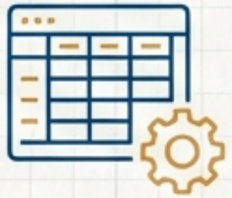
What is a Dataflow Gen2?

A tool to import and transform data using the familiar Power Query Online interface.

When to Use It:

Dataflows Gen2 is the best-suited tool for data transformation when dealing with large-scale data that will continue to grow, especially for users who prefer a graphical interface over code.

1. Navigate to the **Data Factory** workload.



2. Create a new **Dataflow Gen2** to define your transformation steps.

3. Add your Fabric **Lakehouse** as the data destination.

The Pro-Code Path: Interactive Analysis with Spark Notebooks

When you want to use Apache Spark to interactively explore and analyze data in a file within the lakehouse.

Core Concepts

The Simplest Starting Point:

To analyze data in a CSV file, the simplest first step is to **load the file into a Spark dataframe**.

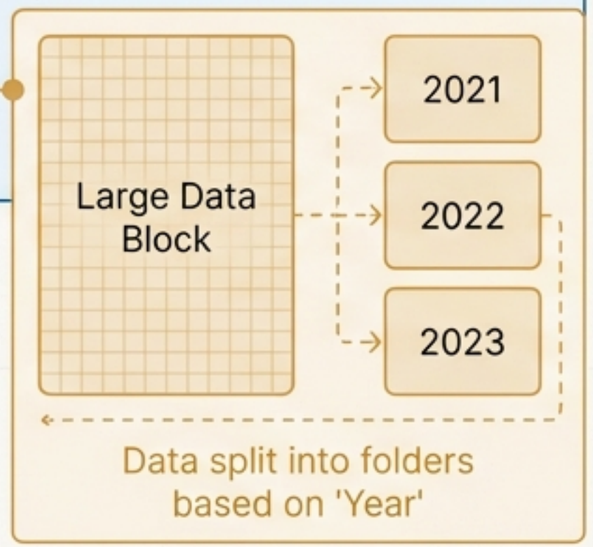
An Optimization Technique:

When saving a large dataframe, use the `partitionBy` method to split the data across different folders. This can significantly improve query performance by allowing Spark to skip reading irrelevant data partitions.

Visual Code Snippet

```
# 1. Load data into a dataframe
df = spark.read.csv("path/to/your/file.csv")

# 2. Save with partitioning
df.write.format("delta") \
    .partitionBy("Year") \
    .save("path/to/table")
```

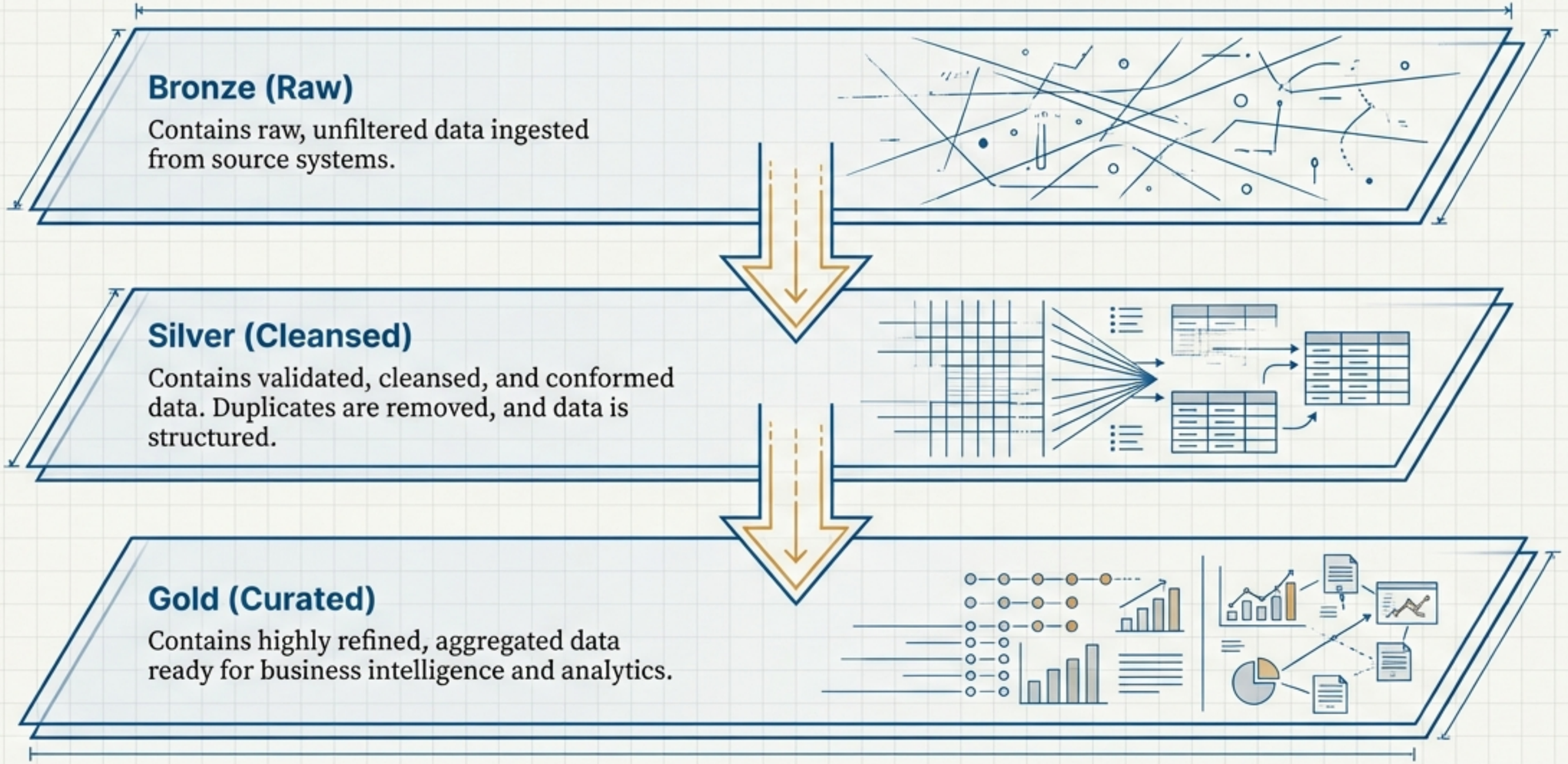


Decision Point: Dataflows Gen2 vs. Spark Notebooks

Feature	Dataflows Gen2	Spark Notebooks
Primary Use Case	Scalable data ingestion and transformation (ETL/ELT)	Interactive data exploration, complex analysis, machine learning
User Interface	Low-code, visual (Power Query Online)	Pro-code (Python, Scala, SQL, R)
Ideal User	Data Analysts, BI Developers, Engineers preferring GUI	Data Scientists, Data Engineers comfortable with Spark
Best For...	Large-scale, repeatable transformation of growing data	Ad-hoc analysis, custom logic, and advanced algorithms
Underlying Engine	Power Query engine, scales out with Spark	Direct access to Apache Spark clusters

The Architectural Blueprint: Organizing the Lakehouse with Layers

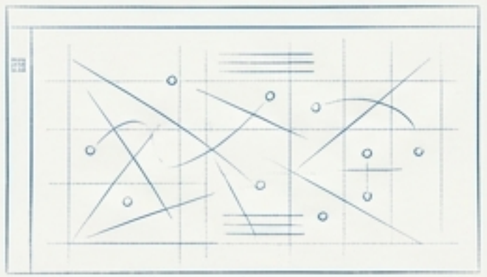
A best practice for structuring a lakehouse is to use a multi-layered approach to progressively refine data.



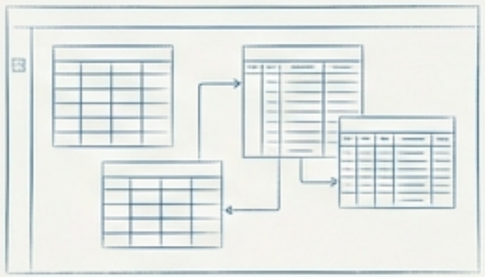
Implementing Layers with Separate Workspaces

Strategy: A highly effective approach is to store the different layers of your lakehouse (Bronze, Silver, Gold) in separate Fabric workspaces.

WS_Bronze



WS_Silver



WS_Gold

Enhanced Security

Apply different access controls and permissions to each layer. For example, restrict access to the raw Bronze layer while providing broader access to the curated Gold layer.

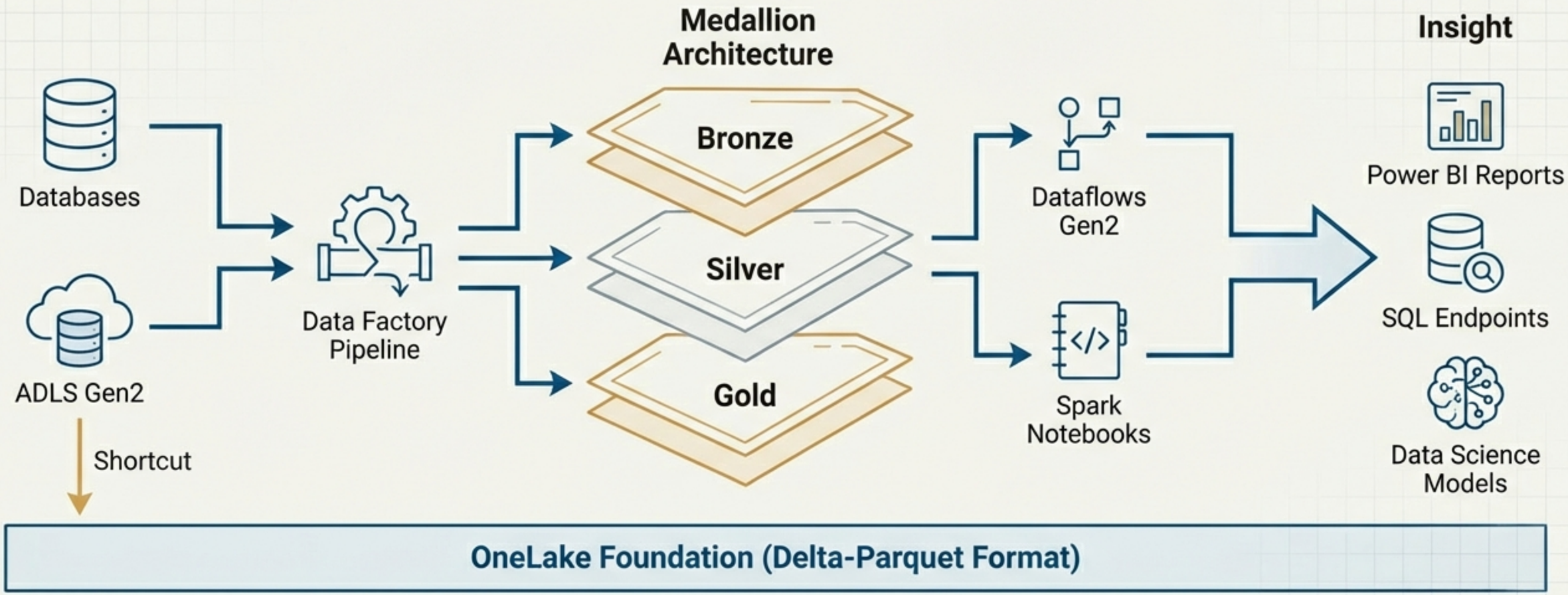
Manage Capacity Use

Isolate workloads and manage Fabric capacity consumption more effectively for different stages of the data pipeline.

Optimize Cost-Effectiveness

Track and attribute costs more easily to different teams or business units responsible for each data layer.

Your Fabric Blueprint: From Ingestion to Insight



Microsoft Fabric provides a complete, integrated blueprint for building modern data solutions, enabling teams to collaborate effectively across the entire data lifecycle.