DS-2002: Data Systems

Overview of Data Warehouse Systems

Prof. Jon Tupitza – Fall 2022

Modern Data Platform: Solution Scenarios

Big (Unstructured and/or Poly-Schematic) Data Integration and Advanced Analytics

"We want to integrate all our data into our data warehouse"

Modern Data Warehousing

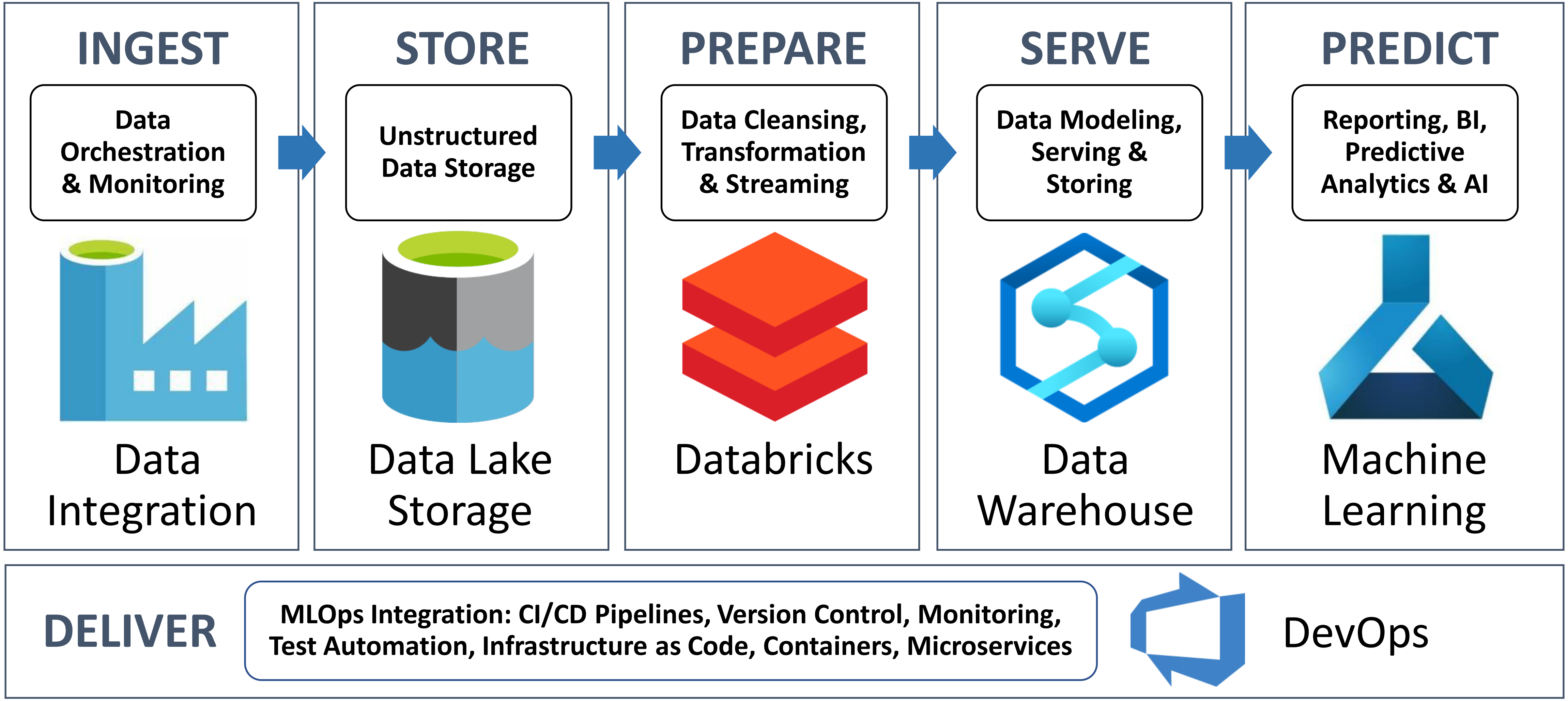
"We're trying to predict which of our customers will churn"

Advanced Analytics

"We're trying to get insights from our devices in real-time"

Real-Time Analytics

Modern Data Platform: Data Services Pipeline



INGEST

Data Orchestration & Monitoring

Data Integration

STORE

Unstructured Data Storage

Data Lake Storage

PREPARE

Data Cleansing, Transformation & Streaming

Databricks

SERVE

Data Modeling, Serving & Storing

Data Warehouse

PREDICT

Reporting, BI, Predictive Analytics & AI

Machine Learning

DELIVER

MLOps Integration: CI/CD Pipelines, Version Control, Monitoring, Test Automation, Infrastructure as Code, Containers, Microservices

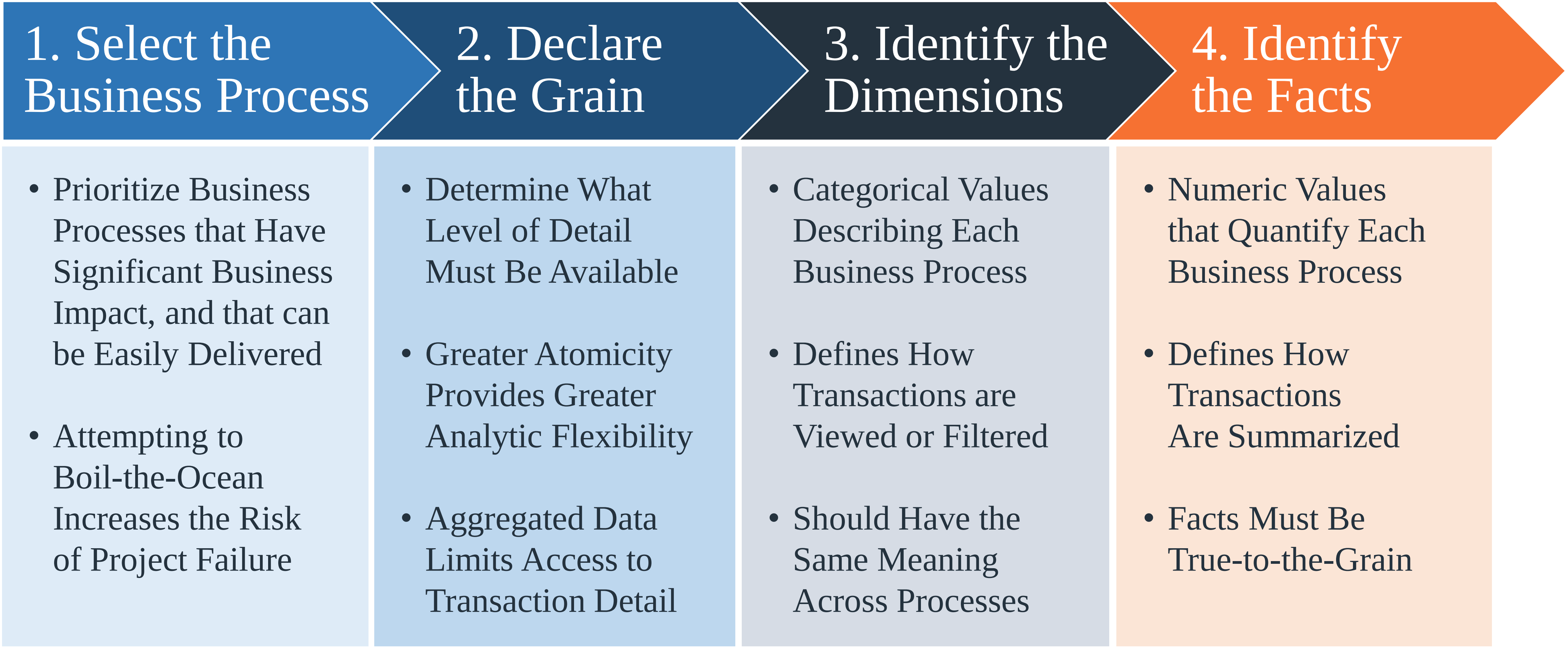
DevOps

The Data Warehouse Process

How to Approach Designing and Building a Data Warehouse

The Four-Step Dimensional Design Process

A Time-Honored and Tested Methodology for Delivering Data Marts & Data Warehouses



1. Select the Business Process

- Prioritize Business Processes that Have Significant Business Impact, and that can be Easily Delivered
- Attempting to Boil-the-Ocean Increases the Risk of Project Failure

2. Declare the Grain

- Determine What Level of Detail Must Be Available
- Greater Atomicity Provides Greater Analytic Flexibility
- Aggregated Data Limits Access to Transaction Detail

3. Identify the Dimensions

- Categorical Values Describing Each Business Process
- Defines How Transactions are Viewed or Filtered
- Should Have the Same Meaning Across Processes

4. Identify the Facts

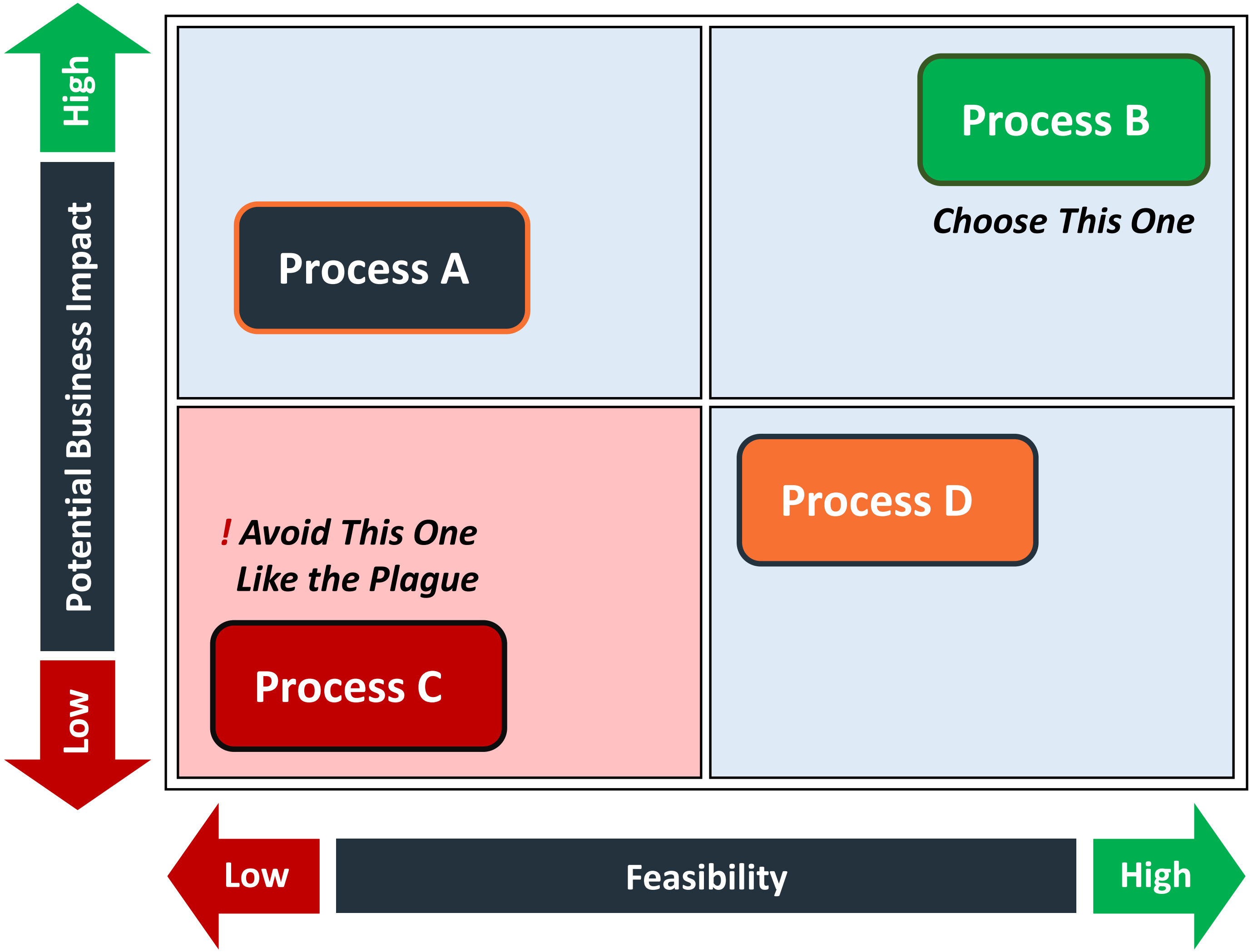
- Numeric Values that Quantify Each Business Process
- Defines How Transactions Are Summarized
- Facts Must Be True-to-the-Grain

The Data Warehouse Toolkit, *by Ralph Kimball*

Selecting Business Processes: Prioritizing Requirements

Quadrant Analysis for Prioritizing Requirements:

- **Business Process A:**
  - High Potential Business Impact
  - Extremely Difficult to Implement
- **Business Process B:**
  - **High Potential Business Impact**
  - **Highly Feasible**
- **Business Process C:**
  - Very Little Business Impact
  - Extremely Difficult to Implement
- **Business Process D:**
  - Little Business Impact
  - Highly Feasible



The Data Warehouse Toolkit, *by Ralph Kimball*

Identifying Dimensions: Data Warehouse Bus Matrix

Using the Same Dimensions Across Multiple Business Processes Enforces a Unified View of the Truth

	Common *(Conformed)* Dimensions			
Business Processes	Date	Product	Store	Promotion
Sales	✓	✓	✓	✓
Inventory	✓	✓	✓	
Deliveries	✓	✓	✓	

The Data Warehouse Toolkit, *by Ralph Kimball*

Identifying Dimensions: Data Warehouse Bus Matrix

Using the Same Dimensions Across Multiple Business Processes Enforces a Unified View of the Truth

	Common *(Conformed)* Dimensions			
Business Processes	**Date**	**Product**	**Store**	**Promotion**
Sales	✓	✓	✓	✓
Inventory	✓	✓	✓	
Deliveries	✓	✓	✓	
WH Inventory	✓	✓		
WH Deliveries	✓	✓		
Purchase Orders	✓	✓		

The Data Warehouse Toolkit, *by Ralph Kimball*

Identifying Dimensions: Data Warehouse Bus Matrix

Using the Same Dimensions Across Multiple Business Processes Enforces a Unified View of the Truth

	Common *(Conformed)* Dimensions							
Business Processes	Date	Product	Store	Promotion	Warehouse	Vendor	Contract	Shipper
Sales	✓	✓	✓	✓				
Inventory	✓	✓	✓					
Deliveries	✓	✓	✓					
WH Inventory	✓	✓			✓	✓		
WH Deliveries	✓	✓			✓	✓		
Purchase Orders	✓	✓			✓	✓	✓	✓

The Data Warehouse Toolkit, *by Ralph Kimball*

Data Integration

How to Approach Populating a Data Warehouse

Data Processing: Extract-Transform-Load (ETL)

Frequently, Data Must Be Moved from Sources to a Database and/or Data Lake

Extract

- This is the step where sensors wait for upstream data sources to land. Once available, we transport the data from their source locations to further transformations.

Transform

- The heart of any ETL job: apply business logic, perform actions such as filtering, grouping, and aggregation to translate raw data into analysis-ready datasets.

Load

- Load the processed data and transport to a final destination. Can now be consumed directly by end-users or treated as yet another upstream dependency.

Data Processing: Batch versus Streaming

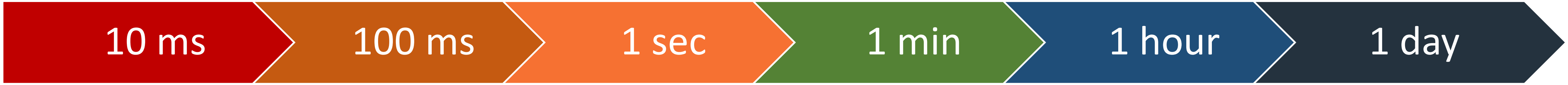
- Data Motion:
  - At-Rest Data: Data that has settled
  - In-Motion Data: Data where new events arrive at some continuous interval
- Datasets:
  - Bounded Datasets: Data of a known & finite size; having a start point and endpoint
  - Unbounded Datasets: Data wherein events are continuously added to the dataset
- Data Processing Engines:
  - Batch Processing Engines: Only capable of processing data after it has settled
  - Streaming Processing Engines: Capable of processing data in-motion as it’s arriving

Data Processing Paradigms: Latency Requirements

Latency & Response:

The speed at which clients require new insights determines the frequency at which new data must be processed

1. Batch
2. Continuous/Streaming
3. Real-time



Low-Latency Real-Time

- Spark-less, highly-available prediction server

Real-Time

- Prediction server with Spark

Micro-Batch

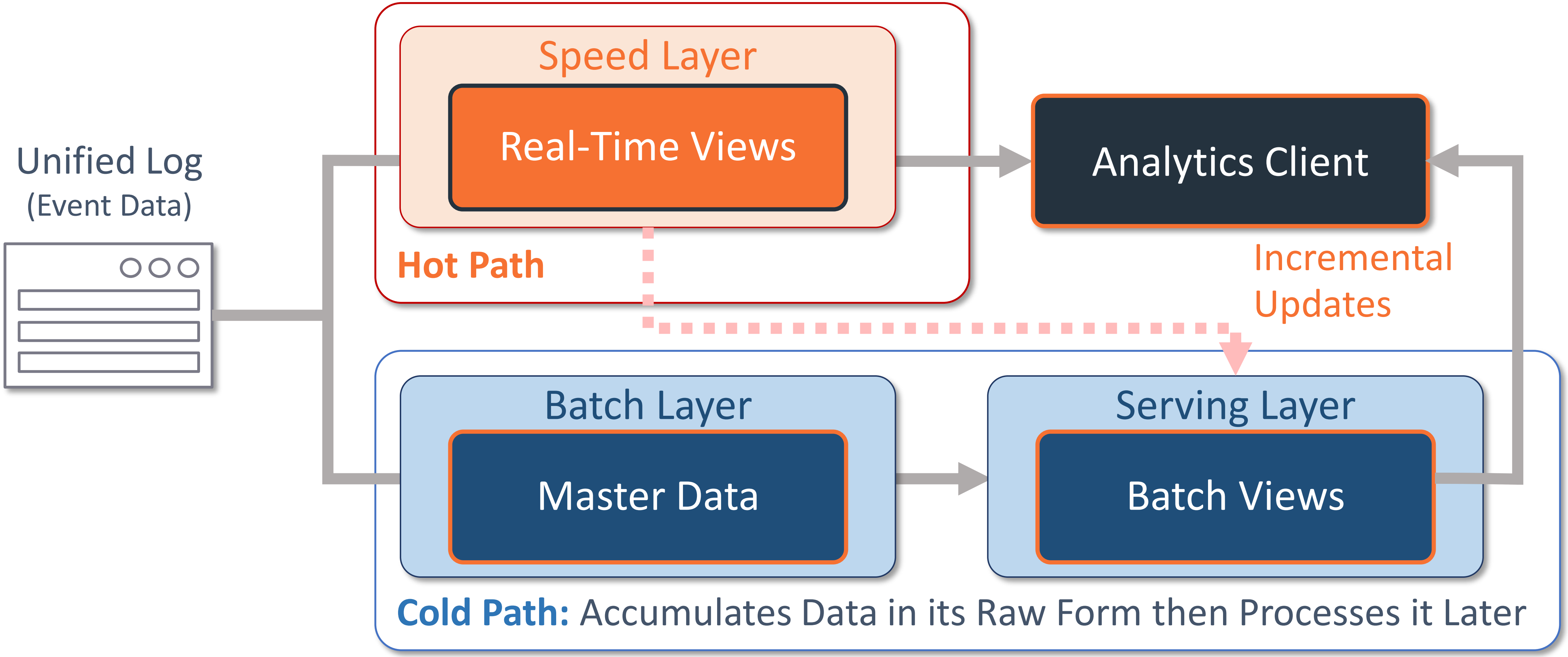
- Structured Streaming

Batch

- Spark batch processing

Data Processing Paradigms: Lambda Architecture

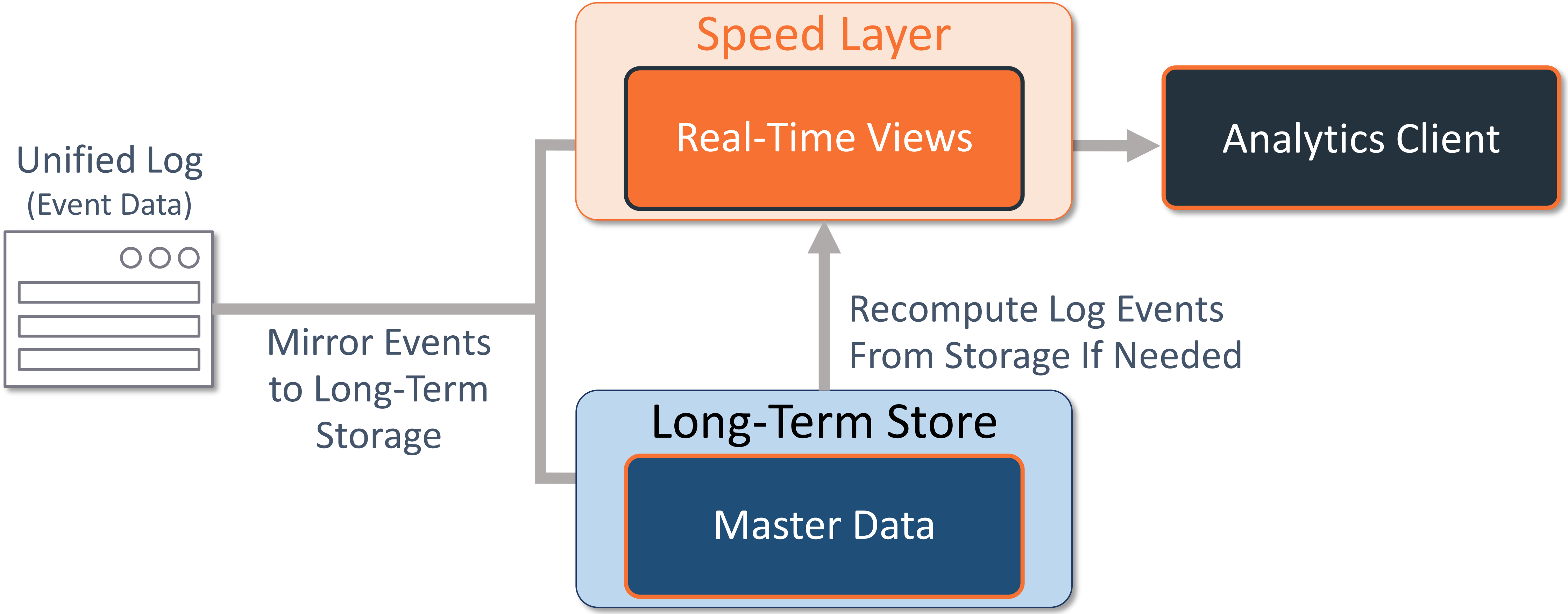
All Data Flows Through One of Two Paths: Hot or Cold



Cold Path: Accumulates Data in its Raw Form then Processes it Later

Data Processing Paradigms: Kappa Architecture

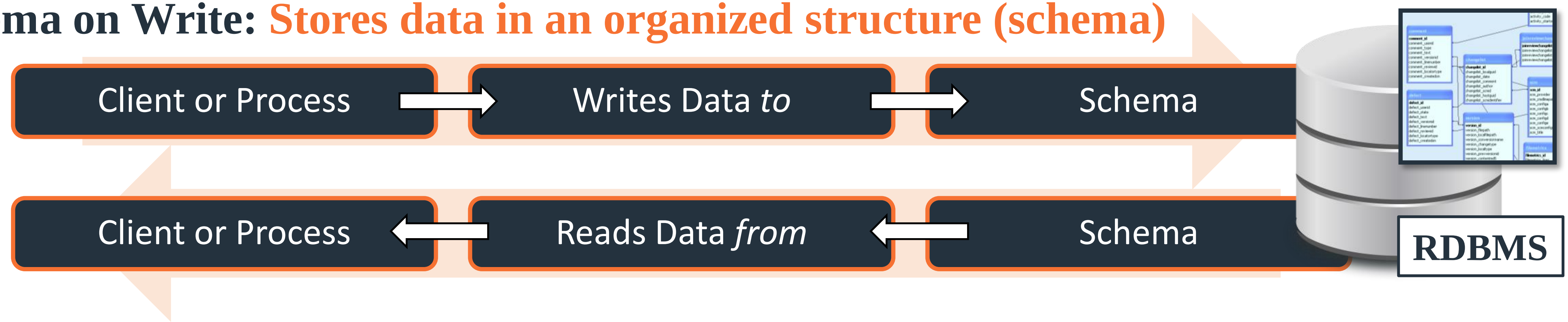
All Data Flows Through One of Two Paths: Hot or Cold



Paradigms: Data Storage and Retrieval

Schema on Write versus Schema on Read

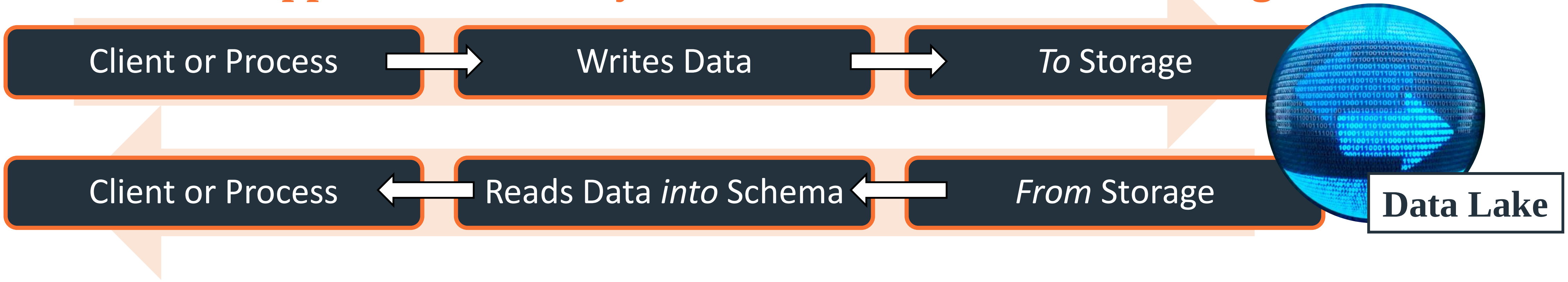
Schema on Write: Stores data in an organized structure (schema)



Client or Process → Writes Data *to* → Schema

Client or Process ← Reads Data *from* ← Schema

Schema on Read: Applies schema only when read, data stored in its original format

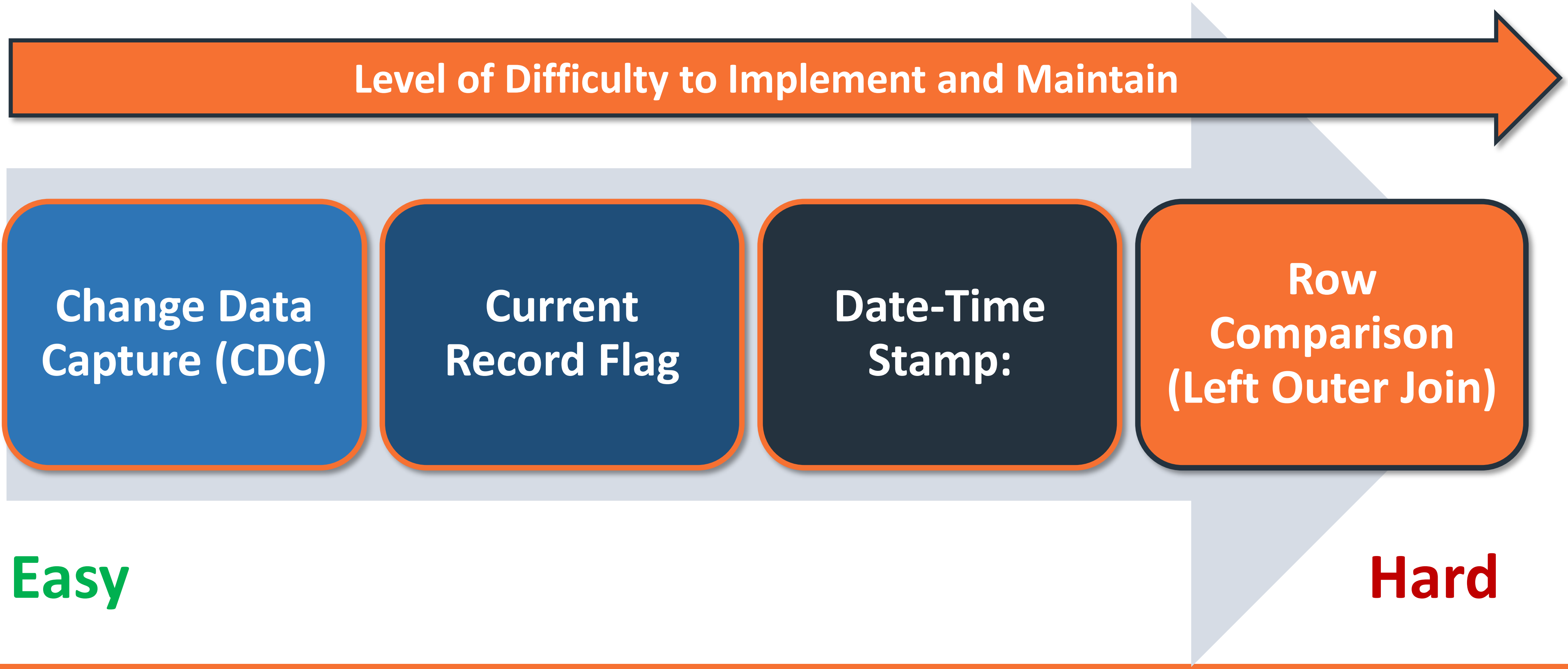


Client or Process → Writes Data → *To* Storage

Client or Process ← Reads Data *into* Schema ← *From* Storage

ETL Processing: Incremental Extraction

Techniques for Minimizing Data Movement: Extract Only the Changes



Level of Difficulty to Implement and Maintain

Change Data Capture (CDC)

Current Record Flag

Date-Time Stamp:

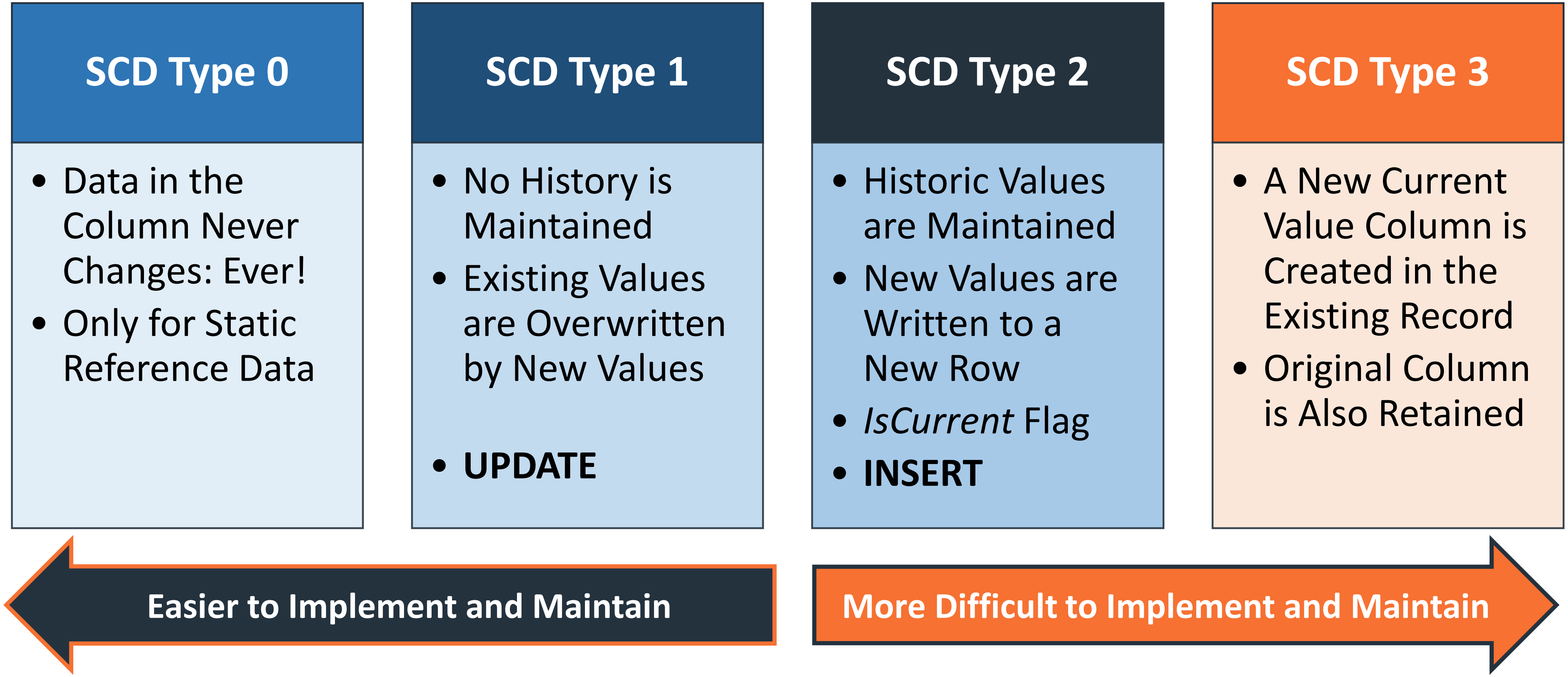
Row Comparison (Left Outer Join)

Easy

Hard

Data Integration Patterns: Dimensional Data

Slowly Changing Dimension Update Strategies: Handling Variable Rates of Change



SCD Type 0

- Data in the Column Never Changes: Ever!
- Only for Static Reference Data

SCD Type 1

- No History is Maintained
- Existing Values are Overwritten by New Values
- **UPDATE**

SCD Type 2

- Historic Values are Maintained
- New Values are Written to a New Row
- *IsCurrent* Flag
- **INSERT**

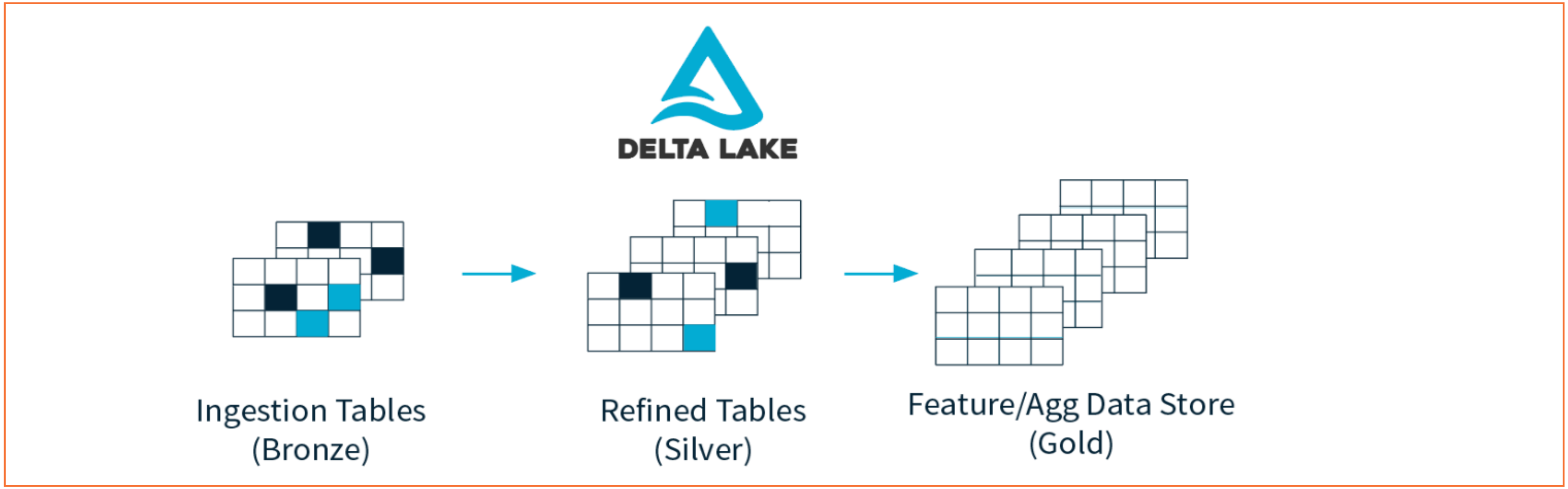
SCD Type 3

- A New Current Value Column is Created in the Existing Record
- Original Column is Also Retained

Easier to Implement and Maintain

More Difficult to Implement and Maintain

Databricks: Delta Lake at Scale



ACID Transaction Guarantees

Atomic, Consistent, Isolated, Durable

Versioned Parquet Files

Delta transaction log keeps track of all operations

Efficient Upserts

MERGE, DELETE, UPDATE

Time Travel

Audit history, pipeline debugging, data reproducibility

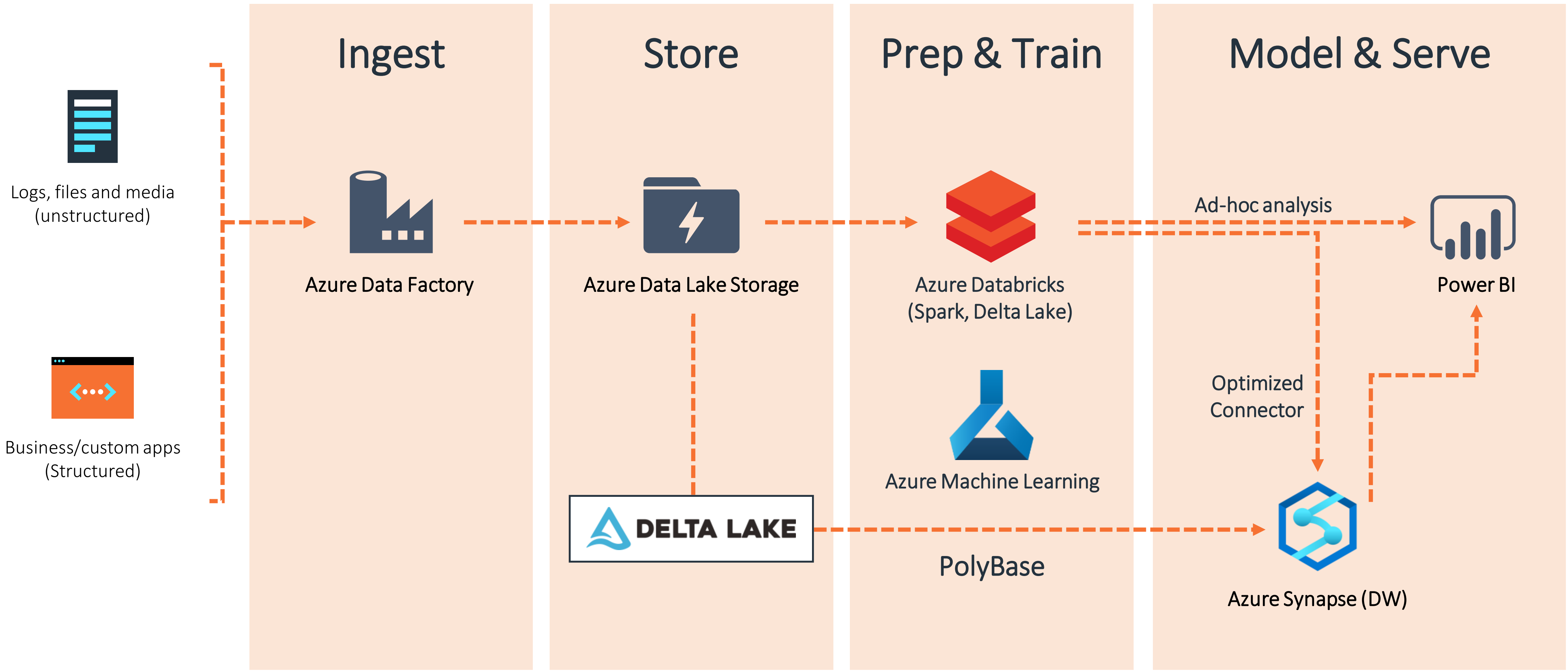
Small file compaction with no interrupt to availability

OPTIMIZE and VACUUM

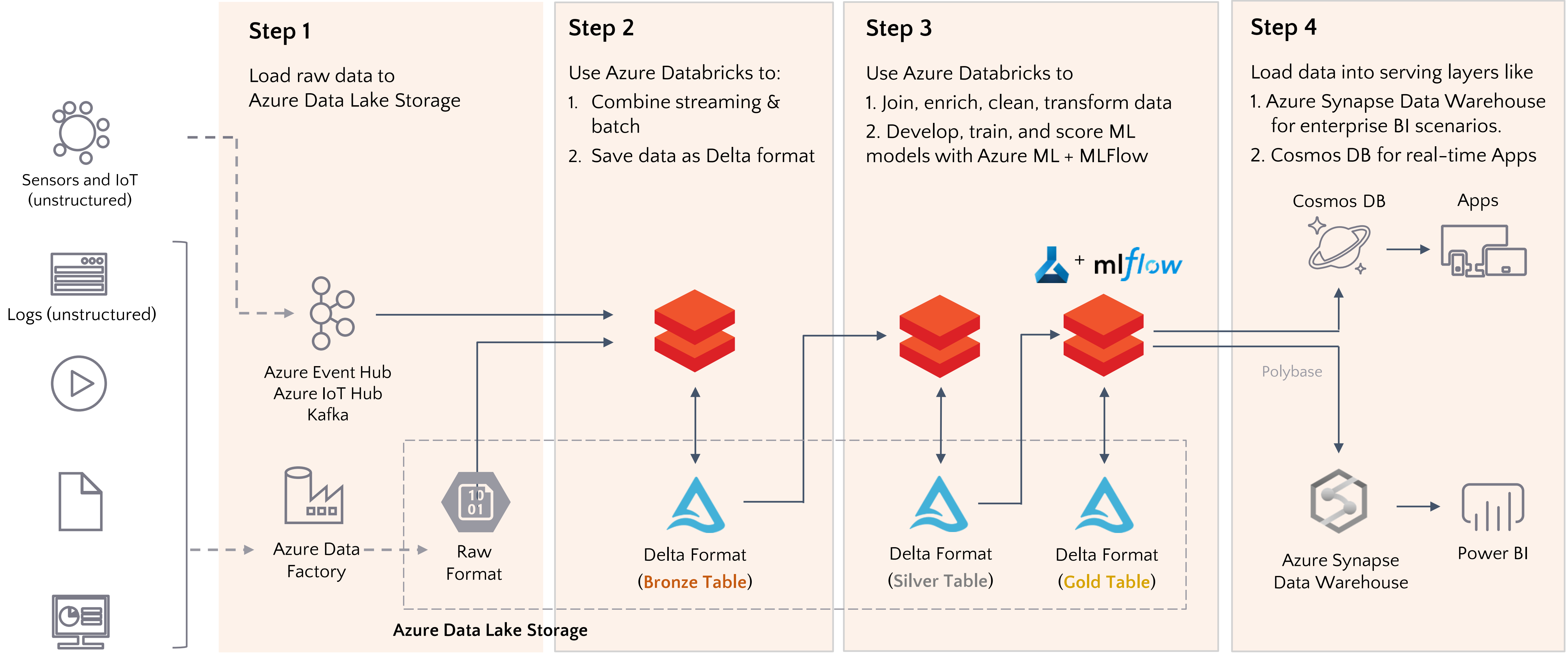
Z-Order partitioning with up to 100x perf

New multidimensional partitioning enables data skipping

Data Engineering… for Data Science



Design Pattern: Modern Data Warehousing



Step 1

Load raw data to
Azure Data Lake Storage

Step 2

Use Azure Databricks to:

1. Combine streaming & batch
2. Save data as Delta format

Step 3

Use Azure Databricks to

1. Join, enrich, clean, transform data
2. Develop, train, and score ML models with Azure ML + MLFlow

Step 4

Load data into serving layers like

1. Azure Synapse Data Warehouse for enterprise BI scenarios.
2. Cosmos DB for real-time Apps

Q & A

An Overview of Data Warehouse Systems