databricks

Reliable Data Pipelines Made Easy

AGENDA

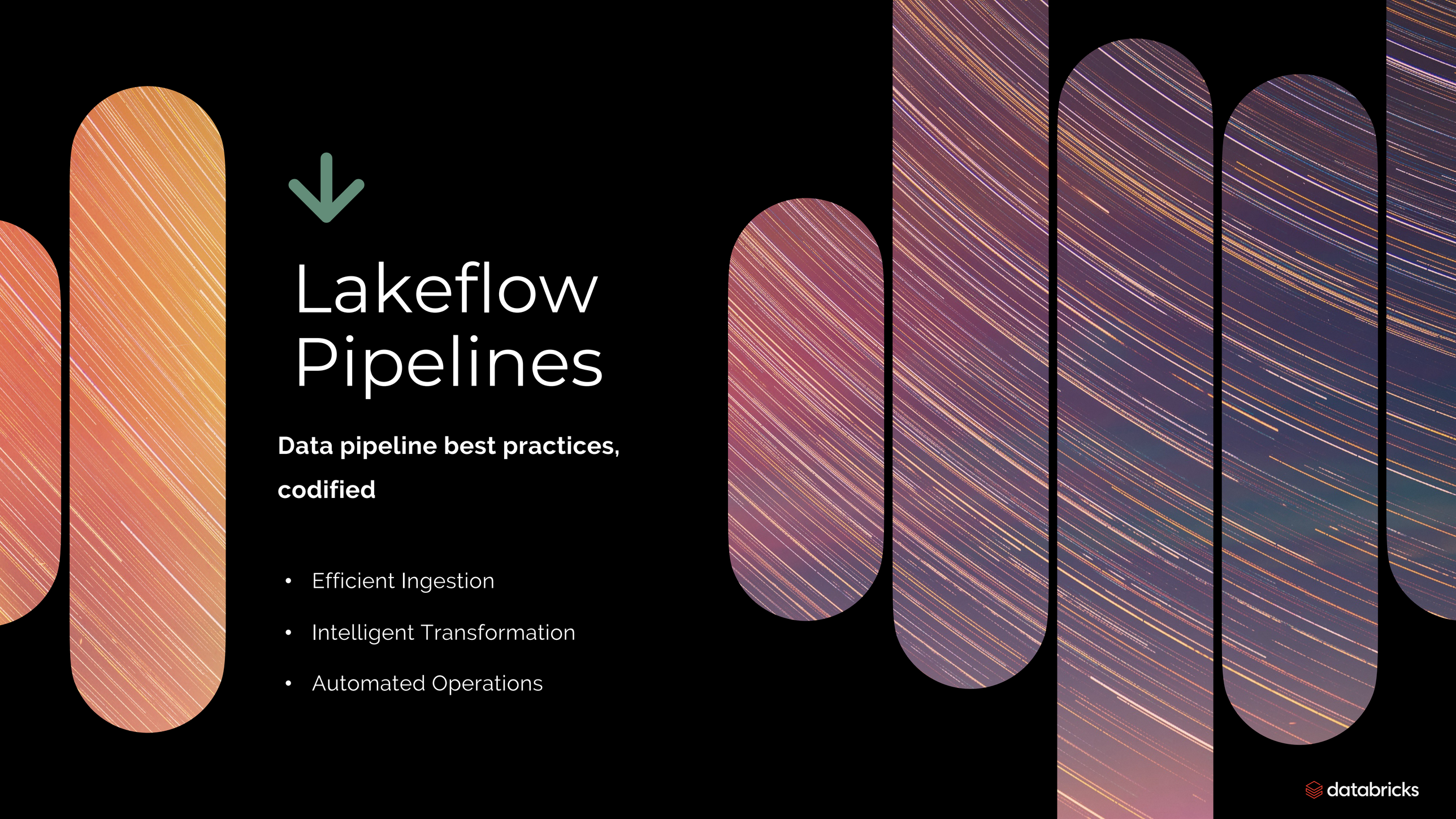
Lakeflow Pipelines

LAKEFLOW PIPELINES OVERVIEW

In the first section we will cover what Lakeflow Pipelines are, how they're used and provide an introduction to declarative development

EXTENDING PIPELINES & DASHBOARDING

In the second section we will go further with our pipeline exploring data quality and how to make pipelines robust and reliable. We will also be customizing a dashboard.



Lakeflow Pipelines

Data pipeline best practices, codified

- Efficient Ingestion
- Intelligent Transformation
- Automated Operations

databricks

CAPABILITIES

Built to Simplify Data Pipelining

Unified Batch & Streaming

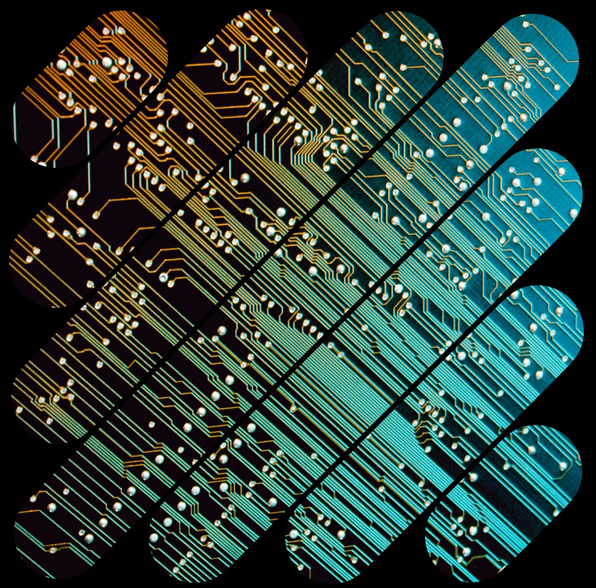
Build with standard design patterns in mind for enforced best practices. Build pipelines once and decide between batch processing and streaming on-the-fly

Incremental Processing

Process only new data as it arrives, reducing compute overhead and cost overruns.

Load Data from Anywhere

Any data can be loaded and processed with Lakeflow Pipelines. Everything from simple files to dynamic API ingestion with managed and un-managed connectors



Automated Data Quality

Constraints and schema evolution support data quality assurance. Invalid data can be deleted, quarantined or cleaned on-the-fly.

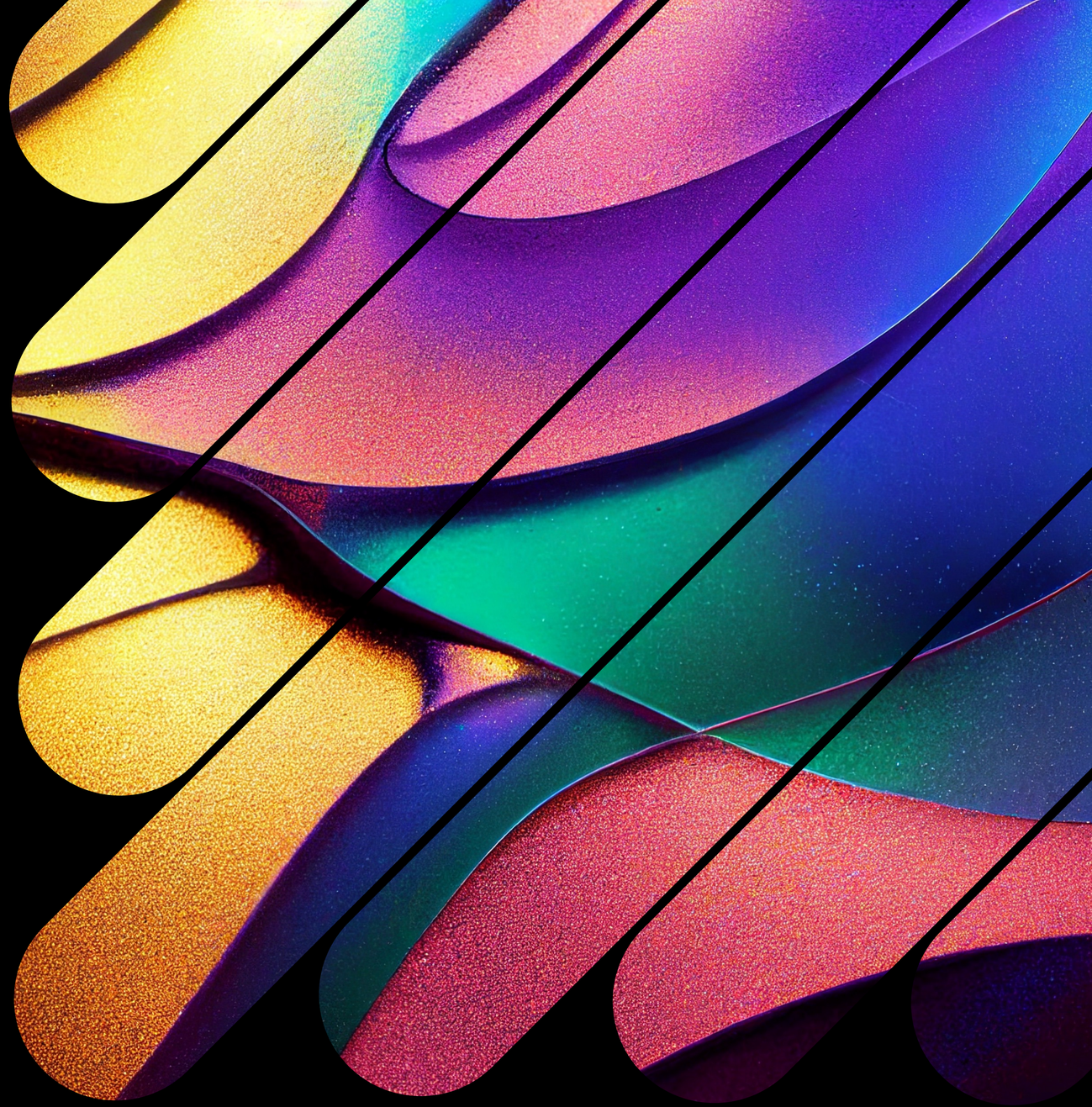
Integrated Pipeline Development

Pipelines are easily built as an assembly of knowledge objects in the form of flows and tables, with standard easy-to-understand language.

Unified Governance & Storage

Building pipelines on Unity Catalog enables out-of-box lineage and governance as well as automated access controls

Declarative Development



WHAT IS IT?

Declarative development is the telling of "what" you want rather than the "how"

WHY DO IT?

Declarative development is typically more efficient and less error-prone. When we use libraries to define the "what" we let the system decide how to be compose the underlying functions for us.

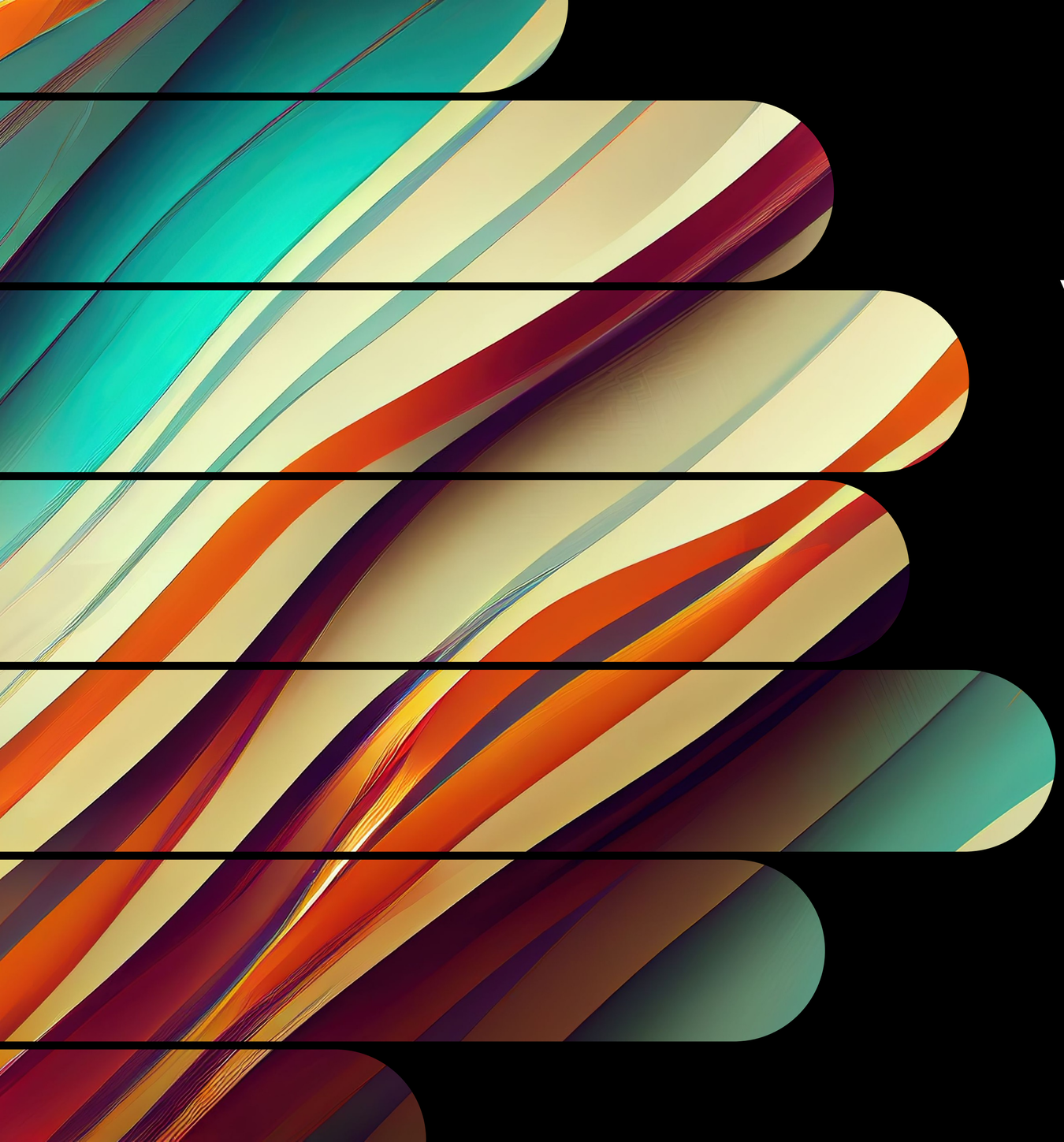
WHEN TO DO IT?

When we use standard libraries (e.g., numpy, stringparser etc.) we are taking a declarative approach to developing systems. Declarative approaches to development tend to be the most resilient and reliable.

Pipeline Development

Constraints and Expectations

Constraints enforce data shape and format to be of a known factor. This is important to ensure high-quality data as it's processed throughout the pipeline. As data moves through the pipeline and is processed, certain expectations of what that data looks like can and should be monitored for reliability. Lakeflow Pipelines adds this capability when defining knowledge objects within the pipeline which can be analyzed, validated and reported on.



Streaming Tables & Materialized Views

STREAMING TABLES

Streaming tables are similar to traditional tables in that they have a defined structure, however streaming tables are “unbounded” meaning they can be updated on-the-fly with new data and support aggregation capabilities such as watermarking. Streaming tables are good for data that is frequently flowing in but not being too disruptive to the overall structure.

MATERIALIZED VIEWS

Materialized views are stored computations and perspectives of data from tables. Materialized views are persisted on disk much like tables but have the distinct advantage of being more performant for typical down-sampled reports since they are ‘pre-calculated’. Materialized views also tend to have auto-optimization capabilities built-in.

INTEGRATION

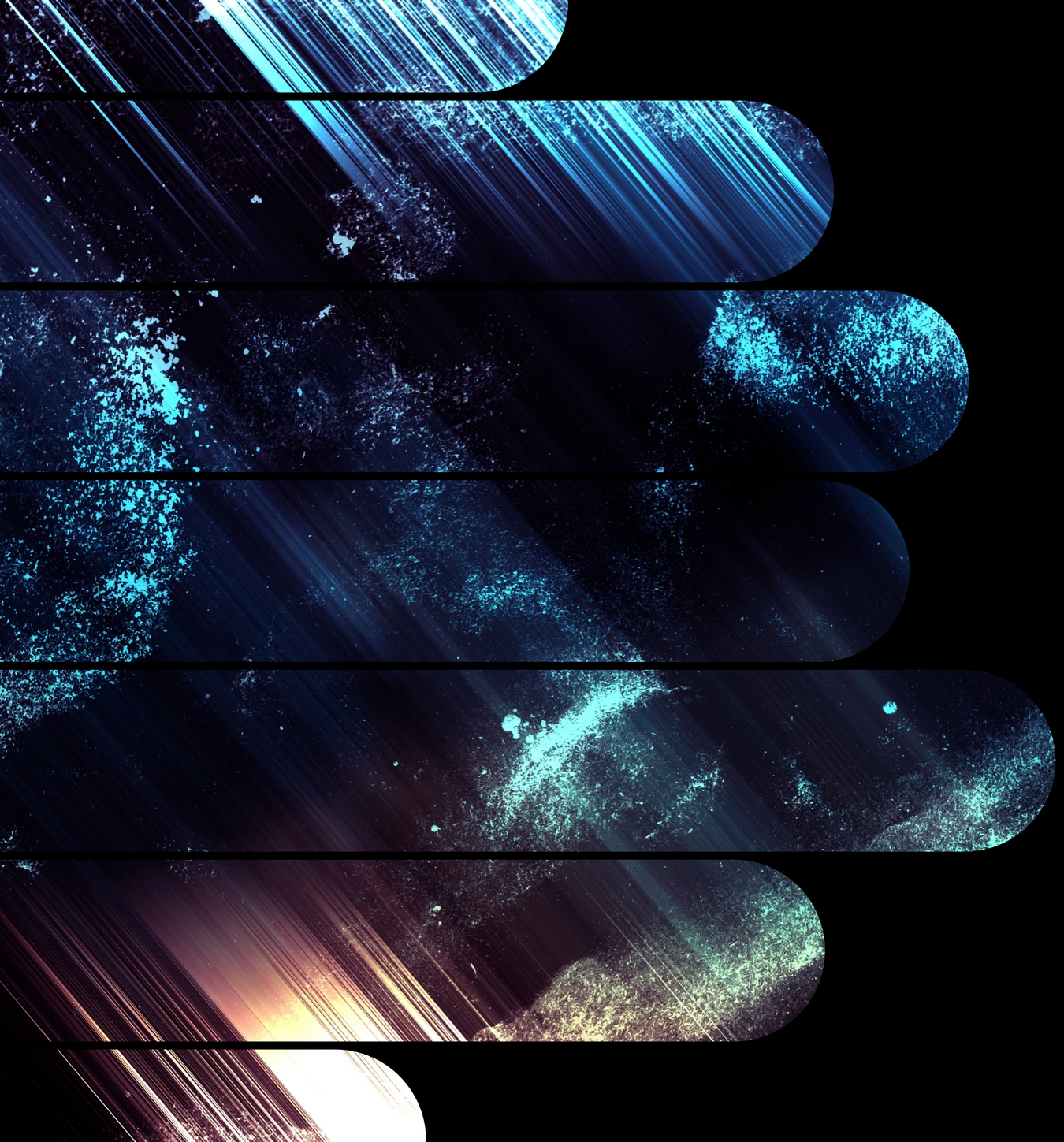
Visualizations & Data Sharing

STREAMING TABLES

Streaming tables are ideal for rendering atomic data, or data that is highly variable (e.g., time series charts). Adding streaming tables to visualizations requires direct-connect compute such as a serving warehouse.

MATERIALIZED VIEWS

Materialized views offer more flexible sharing and export options. While they can be shared with direct-connect compute, they can also be shared with Delta Shares.



COMMON USE CASES

Lakeflow Pipelines in Energy

Common use cases in energy for data pipelines include:

- Ingestion of well operating metrics
- Ingestion of vendor and supplier invoices & contracts
- Ingestion of SCADA / PLC data from field IoT
- Ingestion of occupational, health & safety data


databricks