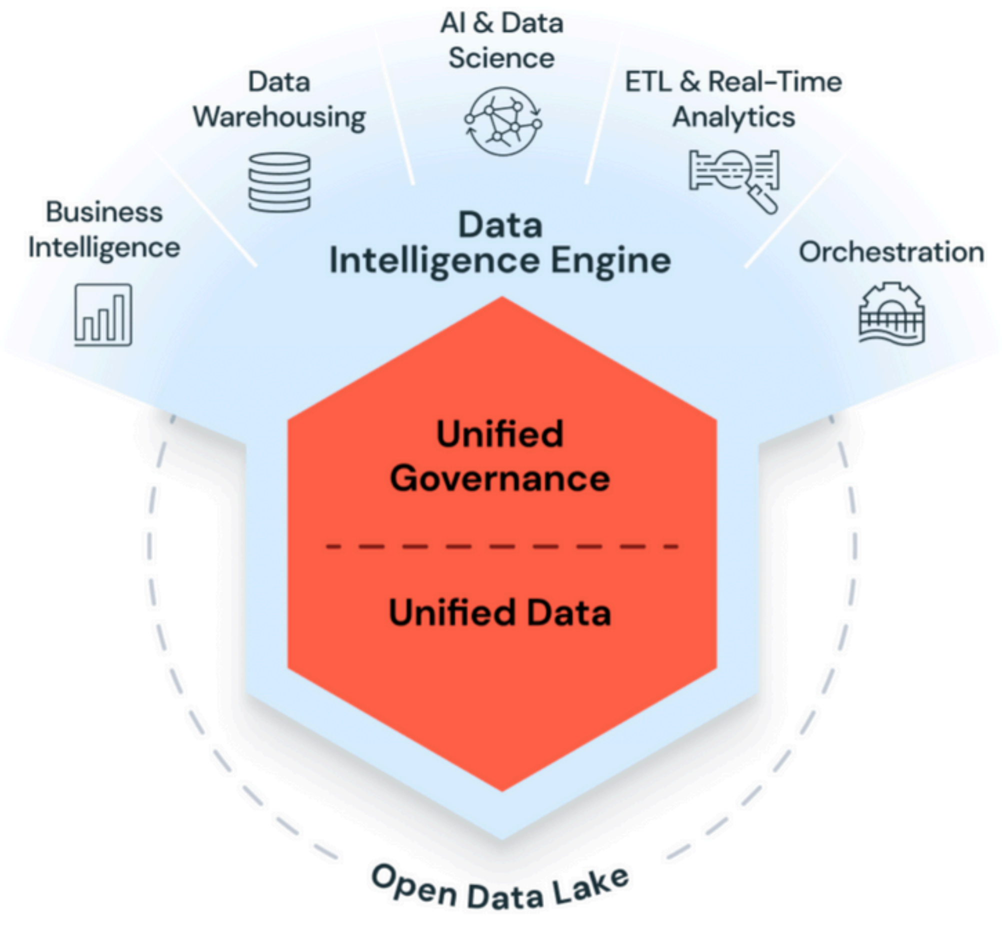
Two Titans of Modern Data Infrastructure: Snowflake & Databricks

What do they have in common? Cloud-native, scalable, and powerful. What makes them different? A whole lot — and that’s the highlight of it. From big data engineering to powerful SQL analytics, these two platforms power the modern data stack.



01

What is Databricks?

Databricks is a unified Lakehouse platform built on Apache Spark.

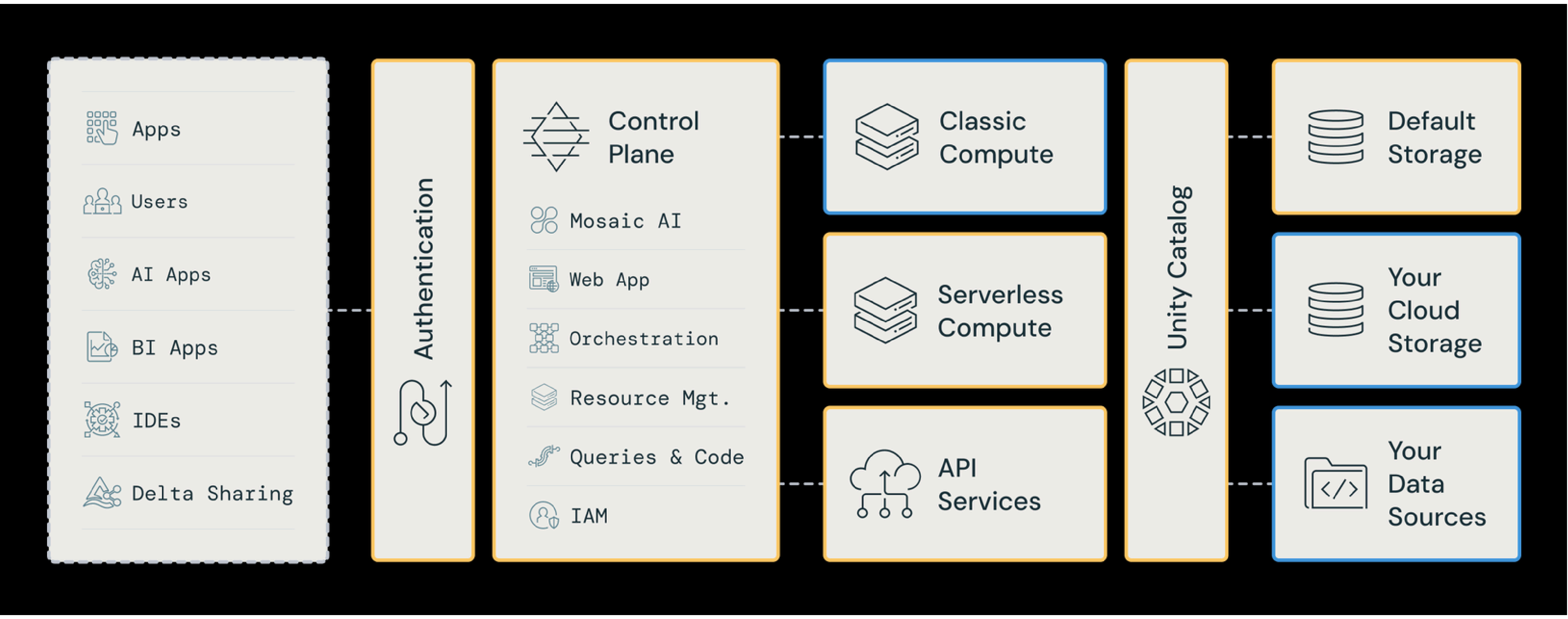
- Supports SQL, Python, Scala, R
- Handles batch + streaming data
- Combines Data Lake flexibility + Data Warehouse performance
- Ideal for ML, AI, and massive-scale ETL

KEY FEATURE:

DELTA LAKE - BRINGS ACID TRANSACTIONS TO YOUR DATA LAKE.

→

Databricks Distributed Architecture



Databricks splits its platform into two planes for optimized performance, security, and governance

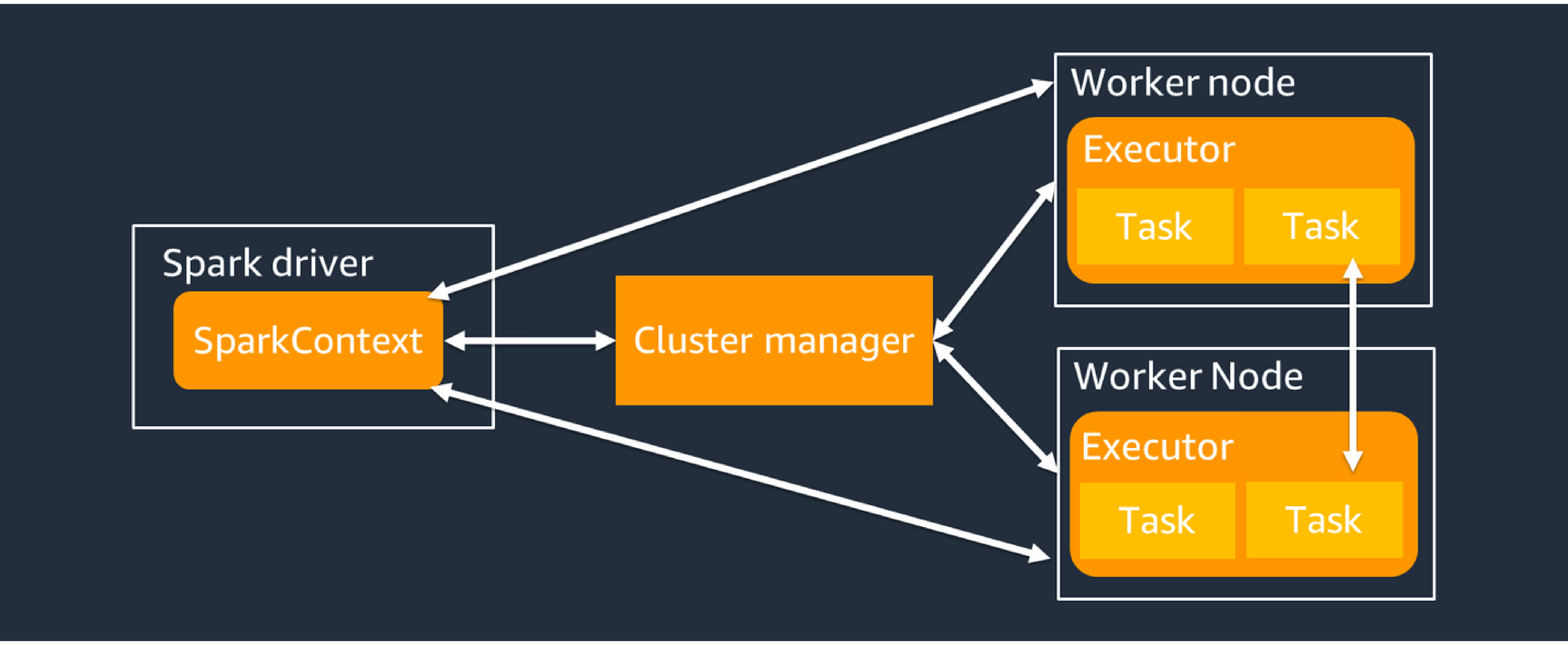
Control Plane	Compute Plane
Hosts workspace UI, notebooks, cluster configurations, and job scheduling	Dedicated to running data processing workloads: Spark jobs, SQL queries, ML workflows
Manages access control, collaboration, and platform logic	In serverless mode, compute runs inside your Databricks account, isolated from cloud provider
Central hub for governance & security policy enforcement	Protects data exfiltration at network & application layers

Key Architecture Features

- Encryption at rest & in transit
- Fine-grained access control
- Unity Catalog for governance & data lineage
- Audit logging and secure cluster connectivity

03

Under the Hood
Apache Spark Architecture



Apache Spark is the powerful distributed engine at the core of Databricks, designed for speed, scale, and simplicity.

Driver Program	**Cluster Manager** (**YARN, Kubernetes, or Standalone**)	**Executors**
Orchestrates the execution	Manages resource allocation	Run tasks in parallel on worker nodes
Constructs the DAG (Directed Acyclic Graph)	Launches and monitors executors	Cache and manage intermediate data
Schedules tasks across the cluster	Handles node failures and task retries to ensure fault tolerance	Report task status back to the driver

04 How Queries Run in Databricks

Discover how your code executes behind the scenes using Apache Spark.
→ Fast, parallel, fault-tolerant.

Let’s say you have a large CSV file of sales transactions from hundreds of retail stores. Here’s how Spark handles it behind the scenes:

Step-by-Step:

Step 1: You upload a CSV of transactions to Databricks and run a PySpark cell to filter, join, and aggregate sales.

Step 2: The Driver parses the logic, builds a DAG (Directed Acyclic Graph), and requests compute resources.

Step 3: The Cluster Manager (e.g., YARN or Kubernetes) launches multiple Executors to process your data in parallel.

Step 4: Spark computes total revenue, top-selling products, and region-wise breakdowns — all in-memory for speed.

Step 5: The results are returned and visualized directly in your Databricks notebook.

Even with millions of rows, your query runs fast, at scale - without writing any complex infrastructure code.

05

What is Snowflake?

Snowflake is a cloud-based data warehouse built for SQL analytics at scale.

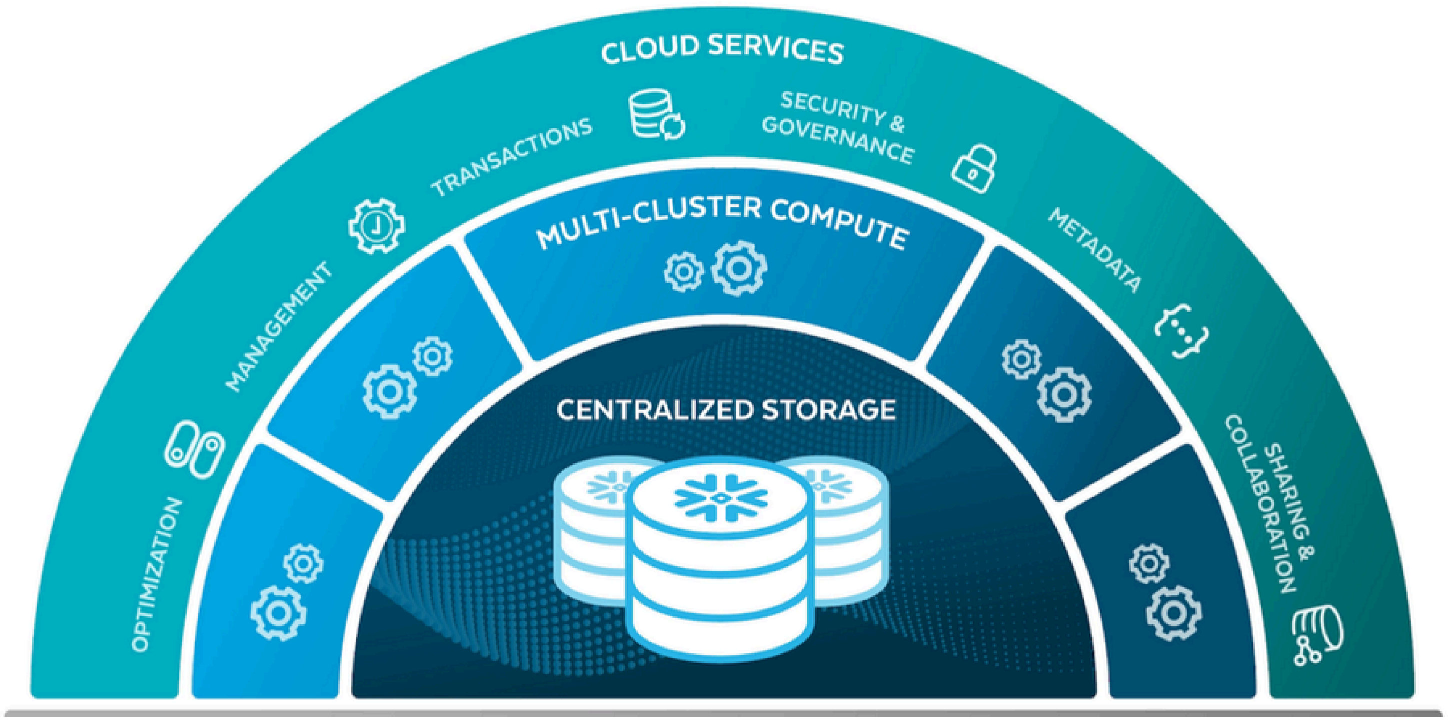
- Supports: Structured + Semi-structured data
- Handles: ELT, BI, dashboarding
- Cloud-native: Works on AWS, Azure, GCP
- Scales independently across storage & compute

KEY FEATURE:
TIME TRAVEL – QUERY DATA AS IT EXISTED AT ANY POINT IN THE PAST (UP TO 90 DAYS).

→

06 SnowflakeArchitecture

Snowflake separates responsibilities into three independent layers, each designed to handle a specific set of tasks.



Storage	Compute	Cloud Services
This layer is responsible for storing all data—structured and semi-structured—in a highly compressed, columnar format.	This layer handles all data processing tasks, including querying, loading, transforming, and analytics.	Orchestrates the system and manages essential services such as metadata, query parsing and optimization, user authentication, and access control.
Key features: • Cloud-native and decoupled from compute • Automatically scaled and optimized by Snowflake • Supports large-scale storage with minimal management **Benefit:** Enables centralized, low-cost data storage without impacting compute performance	**Key features:** • Uses Virtual Warehouses • Each warehouse can scale up/down or pause without affecting others • Multiple users and workloads can run concurrently **Benefit:** Enables elastic compute power that scales independently of storage—ideal for handling bursty or varied workloads.	**Key features:** • Includes the optimizer, metadata manager, security services, and governance controls • Provides seamless user experience with minimal configuration **Benefit:** Centralized intelligence and coordination that supports efficient, secure, and governed access to data.

07 How Queries Run in Snowflake

Curious how your SQL query runs behind the scenes in Snowflake?
→ Cloud-native. Scalable. Smart.

Let’s say you’re working with a massive sales transactions table from thousands of retail stores. Here’s how Snowflakeprocesses your query efficiently — behind the scenes:

Step-by-Step:

Step 1: Parsed & Optimized
Your query is sent to the Cloud Services Layer, where it’s parsed and turned into an efficient execution plan.

Step 2: Sent to Compute
The plan goes to a Virtual Warehouse. If it’s paused, it automatically resumes.

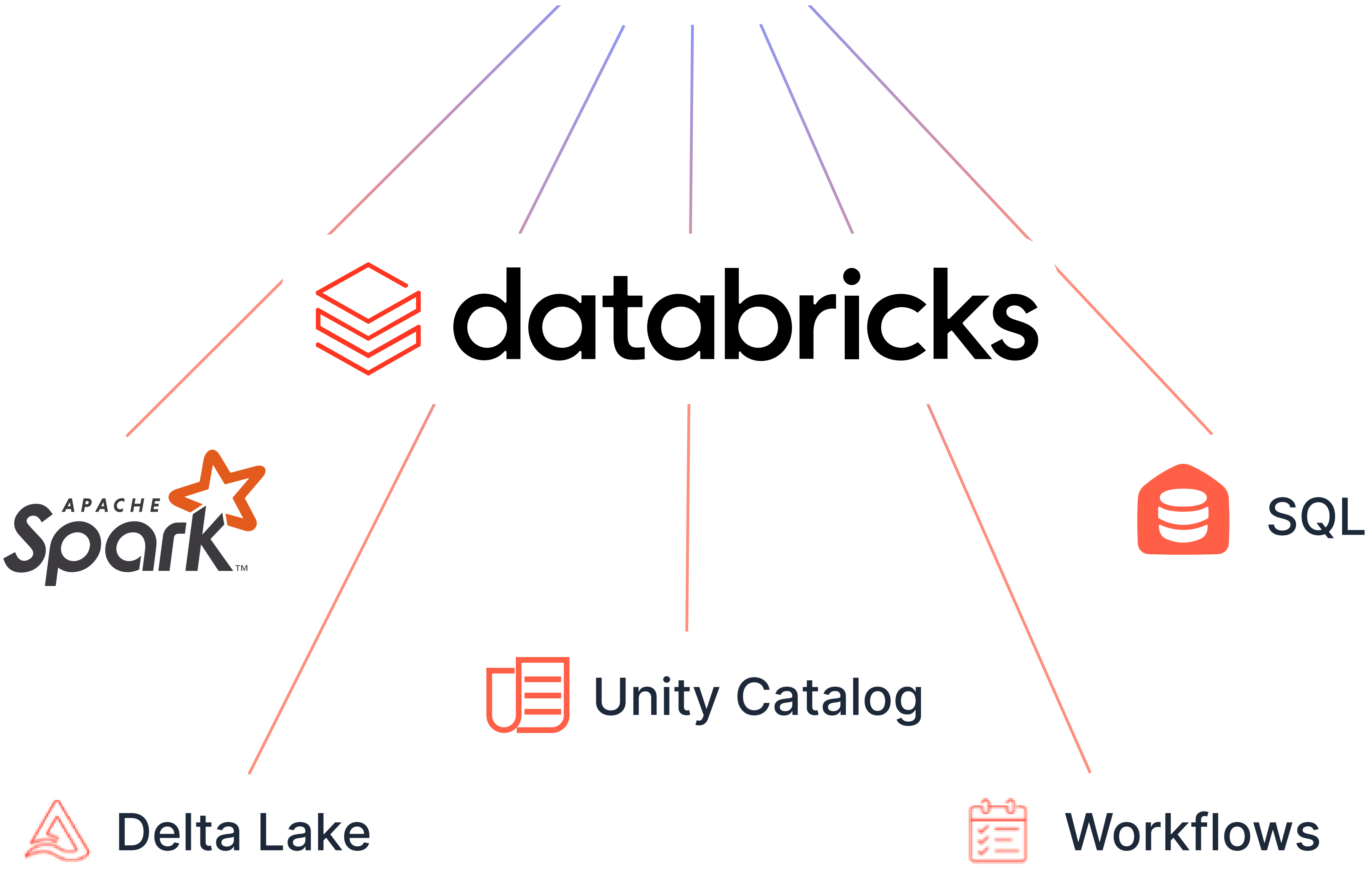
Step 3: Data Retrieved
The warehouse pulls only the needed data from the Storage Layer in compressed, columnar format.

Step 4: Results Returned
Results are processed and returned in milliseconds.

Bonus: Compute auto-scales, suspends, and resumes based on demand—saving time and cost.

When to Use Databricks

- 🔧 Need powerful data transformations with PySpark
- 🔧 Building real-time or large-scale batch pipelines
- 🔧 Doing ML/AI feature engineering
- 🔧 Handling unstructured or streaming data
- 🔧 Best for engineering-heavy, full-stack data workflows.



databricks

Apache Spark

SQL

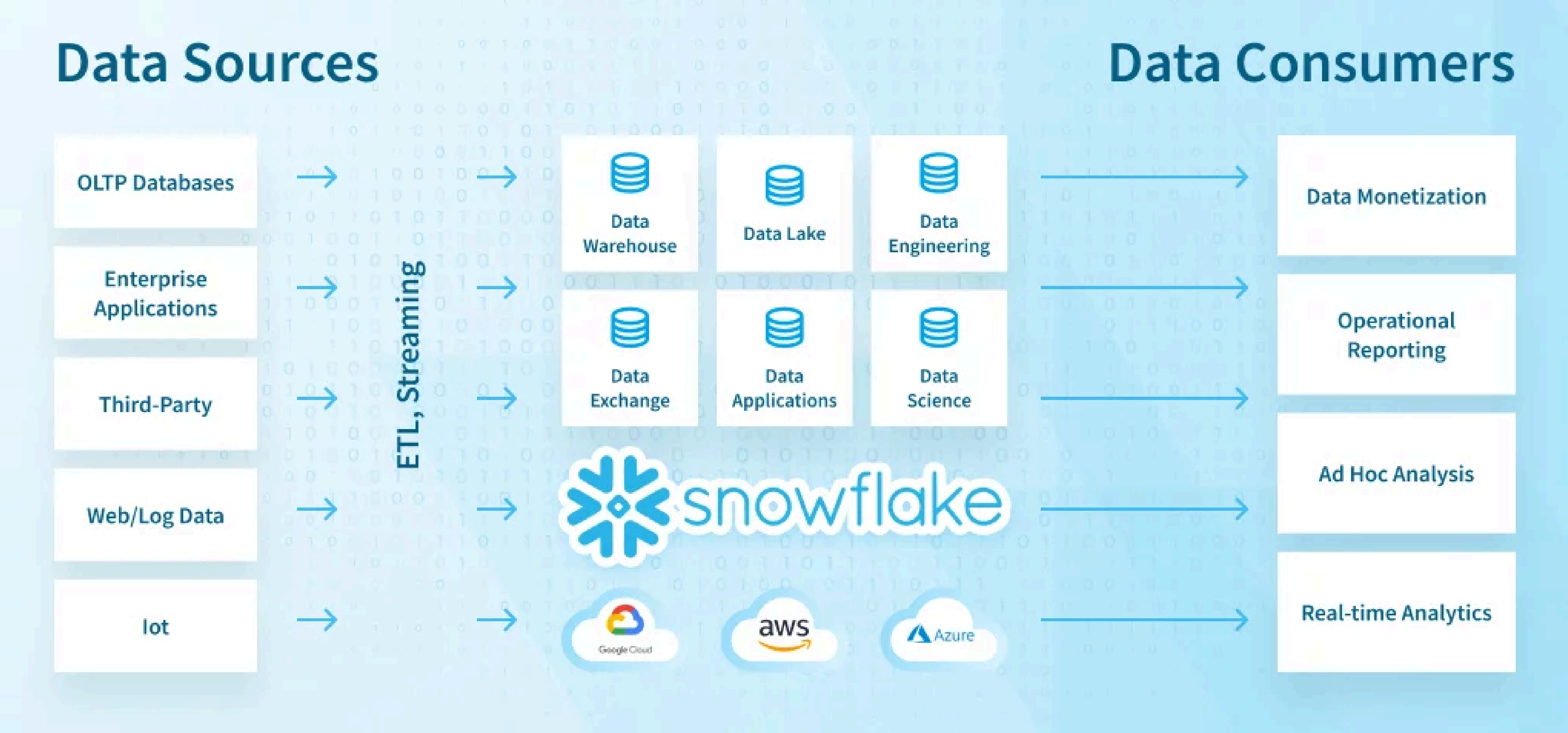
Unity Catalog

Delta Lake

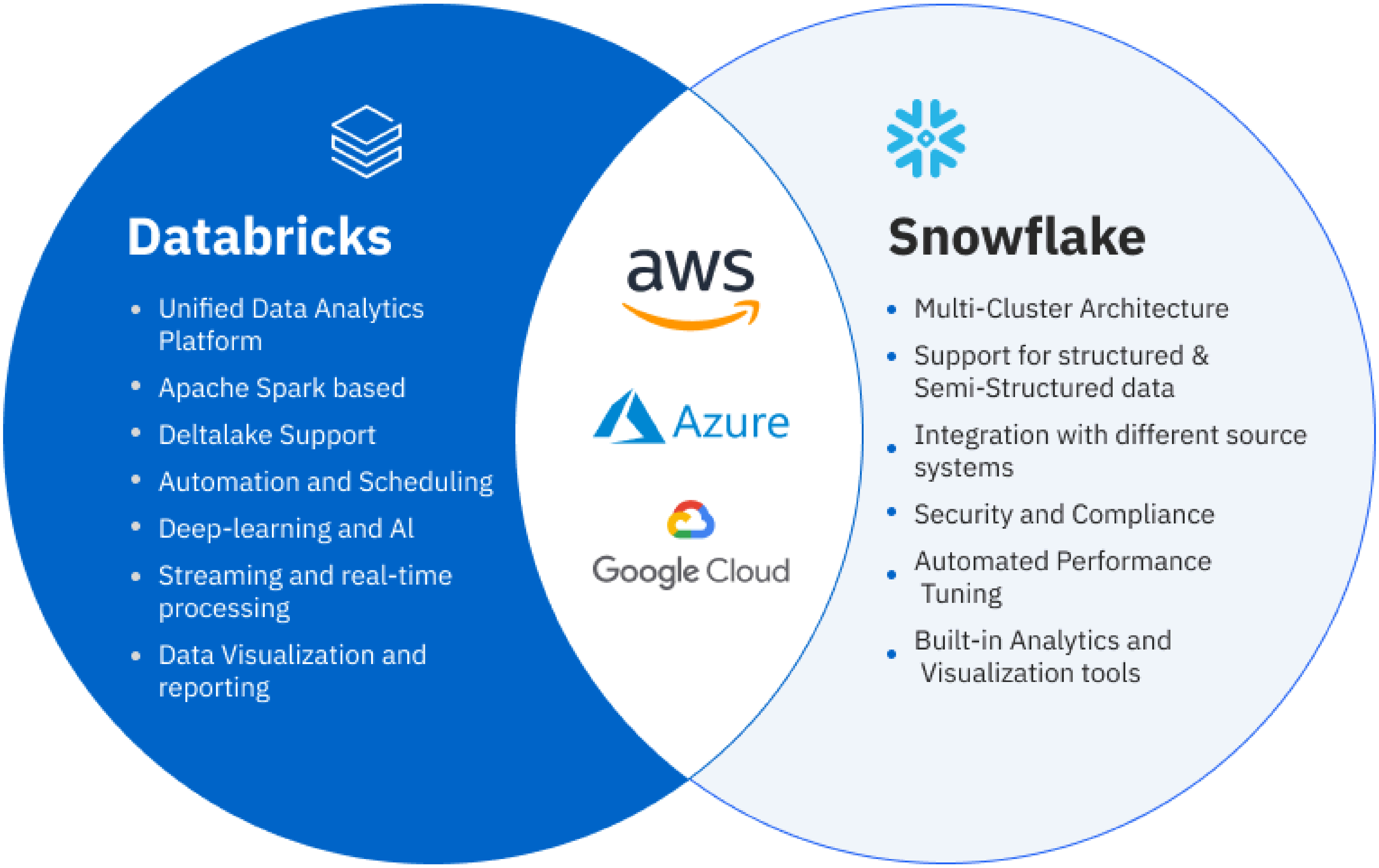
Workflows

When to Use Snowflake

- You want fast SQL querying with no infra headaches
- You're loading clean data into BI tools like Power BI or Tableau
- You prioritize performance, cost-efficiency, and high concurrency
- You need features like Time Travel and Zero-Copy Cloning



Final Takeaway



Databricks

- Unified Data Analytics Platform
- Apache Spark based
- Deltalake Support
- Automation and Scheduling
- Deep-learning and AI
- Streaming and real-time processing
- Data Visualization and reporting

aws

Azure

Google Cloud

Snowflake

- Multi-Cluster Architecture
- Support for structured & Semi-Structured data
- Integration with different source systems
- Security and Compliance
- Automated Performance Tuning
- Built-in Analytics and Visualization tools

Don't think of Databricks vs. Snowflake — think of Databricks + Snowflake.
Use Databricks for scalable processing and ML
Use Snowflake for fast querying and reporting
Together, they supercharge your data pipeline