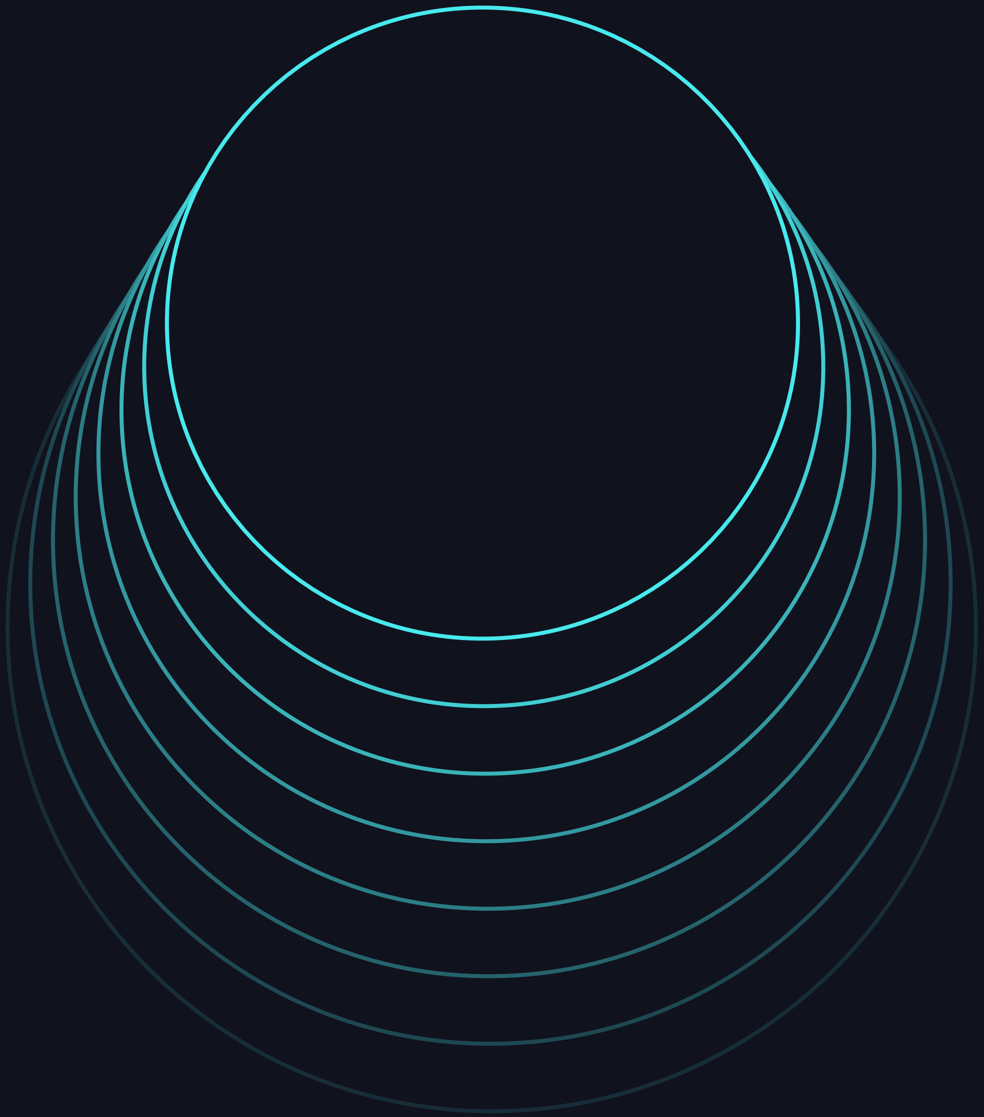
DATA+AI SUMMIT BY databricks

SUCCESSFULLY MIGRATING A SNOWFLAKE DATA WAREHOUSE TO THE DATABRICKS DATA INTELLIGENCE PLATFORM



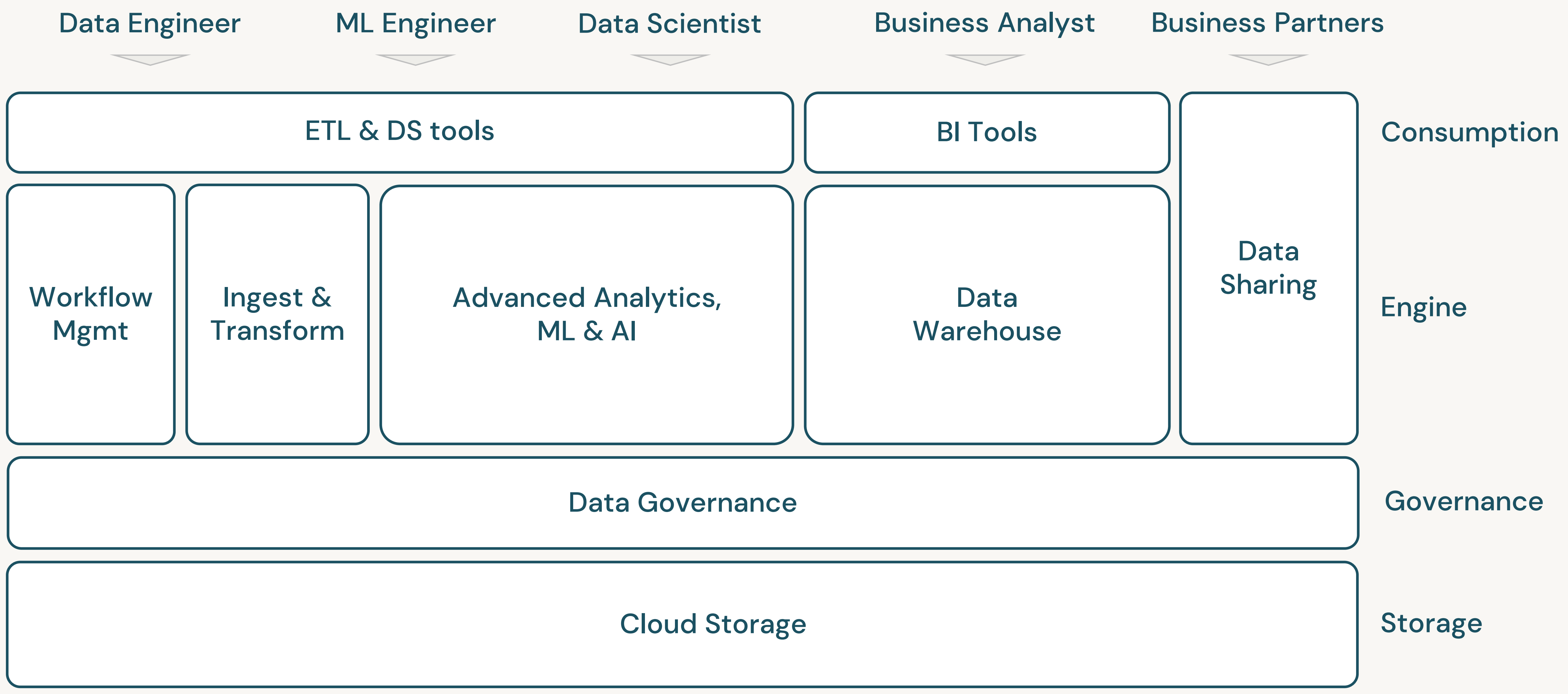
Kevin Barlow
Sr. Solution Architect, Databricks

Product safe harbor statement

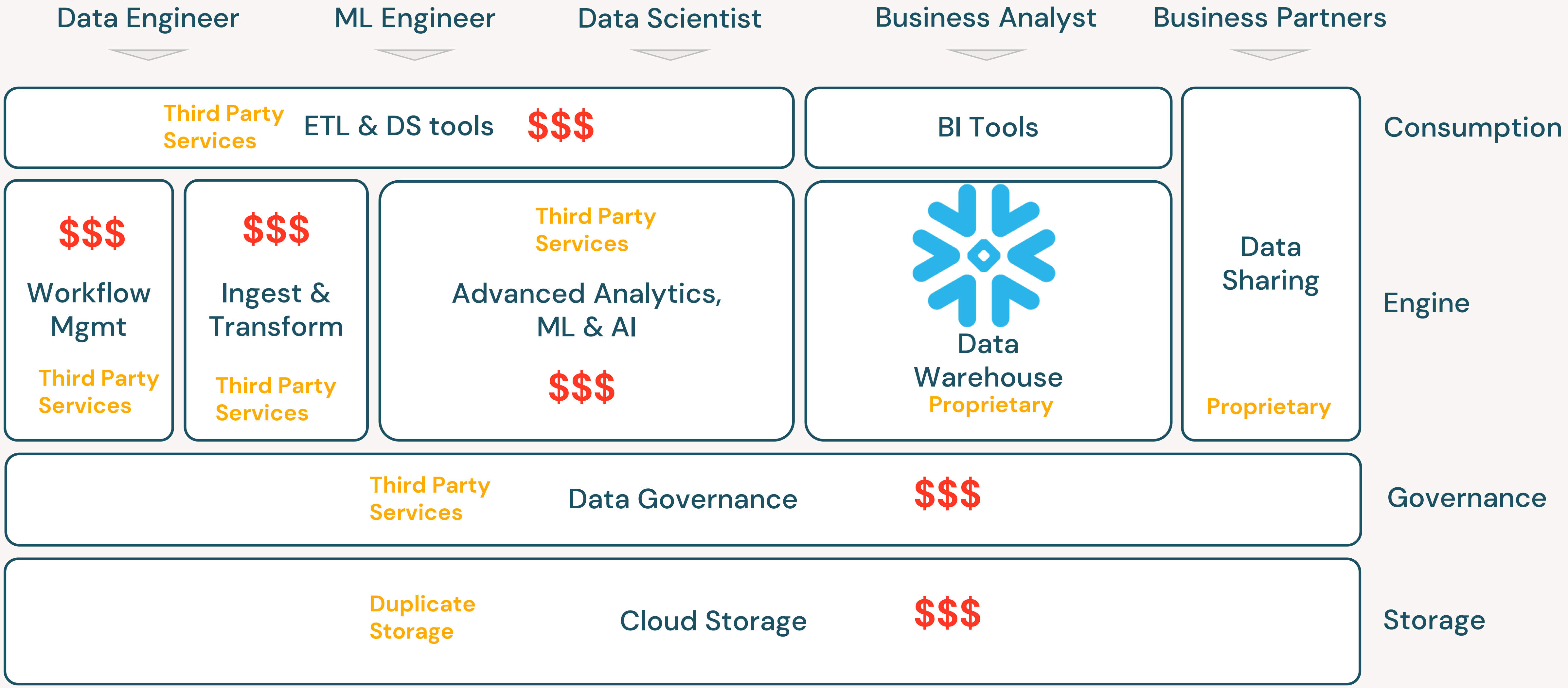
This information is provided to outline Databricks' general product direction and is **for informational purposes only.** Customers who purchase Databricks services should make their purchase decisions relying solely upon services, features, and functions that are currently available. Unreleased features or functionality described in forward-looking statements are subject to change at Databricks discretion and may not be delivered as planned or at all.

Snowflake Data Warehouse Migrations to Databricks

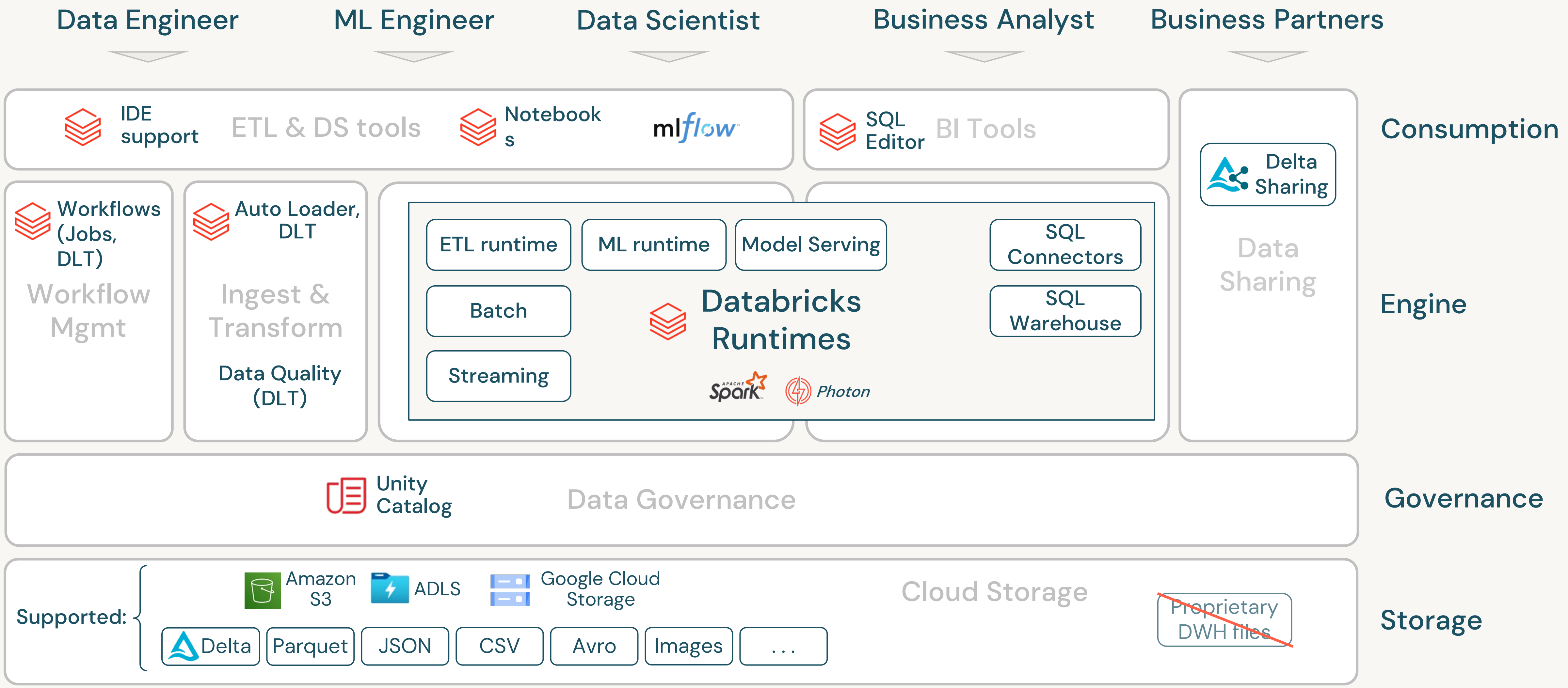
Cloud Data Analytics Framework



Common situation with Snowflake CDW



Cloud Data Analytics Framework with Databricks



Areas of optimization and best practices

Prefer open data formats

Use the Delta data format

Data stored in a proprietary data warehouse format leads to high storage costs and lock-in

DELTA LAKE

Open formats like Delta Lake (now with UniForm) allow for access across your organization

Actions

During data migration convert your data from the proprietary warehouse format it is currently stored in to Delta

Pro Tip
Use the `COPY INTO` command to offload data from Snowflake warehouses to object storage in parquet format

Unify data management & data security

Migrate to Unity Catalog (UC)

Unity Catalog

According to IDC Research –

"People spend 60% to 80% of their time trying to find data. It's a huge productivity loss."

Dan Vesset
Group Vice President, IDC

Actions

Have a data lake governance strategy while you plan for migration

Migrate to Unity Catalog, track and understand the usage

Pro Tip
Plan and build your security and data policies on UC before you migrate data

Leverage UC integration with catalog providers from Databricks Partner Connect

Define standards for integration

Use Databricks Delta Live Tables (DLT) & Autoloader

Often more time is spent on tooling than the actual transformation of data in Data Warehouse Extract Transform & Load (ETL) pipelines

Actions

Microbatch or streaming is the best way to do ingestion (files, Kafka, Kinesis, Change Data Capture, ...)

Automatically manage infrastructure and accelerate ETL development using DLT

Pro Tip
Use Autoloader for both historical data migration and incremental data from sources and use DLT for SCD Type 2 requirements

Design workloads for performance

Using Databricks features for performance efficiency

Photon

A common reason for high costs and performance issues on data warehouse platforms is using a SQL based compute approach for all purposes

Using the correct engine for ETL pipelines, data science apps, streaming and so on allows for better performance & cost control

Actions

Split workloads by type and use optimal compute resources to get best performance and lower cost

Use Unity Catalog for a single data governance model irrespective of the engine used

Pro Tip
Use SQL Warehouses for interactive analysis and dashboarding
Use clusters for Data Science and Engineering

Eliminate orchestration struggles

Using Databricks Workflows

"Data pipelines are growing in size, volume, and complexity, with multistage processing and dependencies between various data assets."

Gartner Data Engineering Essentials, Patterns and Best Practices, September 2022

Orchestrate processes across all data, analytics and AI use cases

Actions

Use Databricks Workflows either alone or combined with Airflow or other external orchestration tools

Integrate with your favorite IDEs

Pro Tip

Use Databricks Jobs to orchestrate workloads composed of a single task or multiple data processing and analysis tasks This can include Delta Live Tables (DLT) ingestion and transformation

Developer Experience

Using Databricks IDE support and partner connect

Engineers want the full power of lakehouse from their favorite IDEs and development tools.

Do not compromise and have your engineers to choose one tool. Instead provide flexibility with any programming language based on personas.

Actions

Databricks Partner Connect supports native integration with many Data Management tools - dbt, Fivetran, etc.

Use Databricks IDE plugins such as VS Code Extension to build locally and manage CI/CD

Pro Tip
Data ingestion jobs in DBT, Fivetran, Matillion can be refactored easily to write to Delta instead of Snowflake or other cloud warehouses

Migration Approach

Migration Planning

Some key guidelines

Governance

Build a Lakehouse governance model with Unity Catalog at its core

Use cases & prioritisation

Migrate use cases and business value, not tables and pipelines. Use this to drive the implementation strategy

Progress & Deliverables

Tangible deliverables along the process (assessments, mappings, architecture, delivery kit, MVPs)

Personas

Understand the various data users and their preferred methods of working & tools

Code

Reuse working code and refactor where needed. Leverage automated tooling.

Enablement

Leverage training, best practices and migration guides

Migration Methodology

Phase 1 Discovery

Inventory database objects and workloads

Profile Workload (ETL vs BI)

Identify Warehouse Dependencies

List current gaps being addressed

Phase 2 Assessment

Analyze Code Complexity

List out SLAs

Calculate TCO

Assess automated code conversion viability

Phase 3 Strategy

Design Target State Architecture

Perform Technology Mapping [by features]

Identify MVPs

Design Migration Plan

Devise Implementation Strategy

Phase 4 Pilot Onboarding

Develop validation criteria and design testing approach

Generate Success Report

Prioritization of MVPs and sprint schedule

Create training and enablement plan

Phase 5 Go-Live

Develop change management plan

Post migration Testing and Validation plan

Roll-out and cutover plan

Migration Execution Pillars

Architecture & Infrastructure

Establish deployment Architecture

Implement Security and Governance framework

Map legacy features to Databricks capabilities

Data Migration

Map Data Structures and Layout

Complete One time load

Implement incremental load approach

Code Migration

Migrate transformation and pipeline code, orchestration and jobs

Speedup your migration using Automation tools

Validate your results with On Prem data and expected results

Cut-over plan

BI & Analytics

Re-point reports and analytics for Business Analysts and Business Outcomes

Connect to reporting and downstream applications

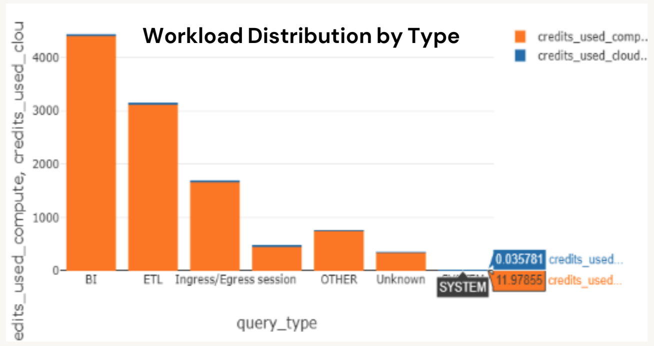
Data Science & ML

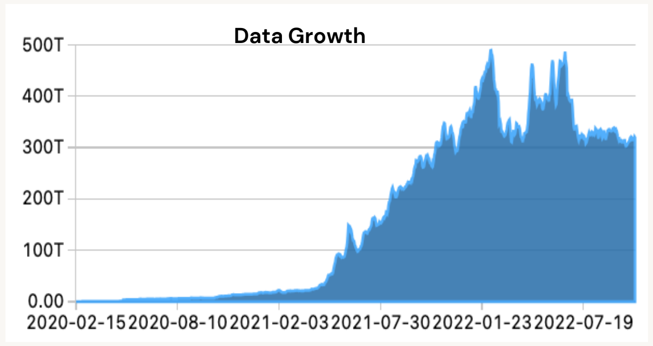
Onboard Data Science teams

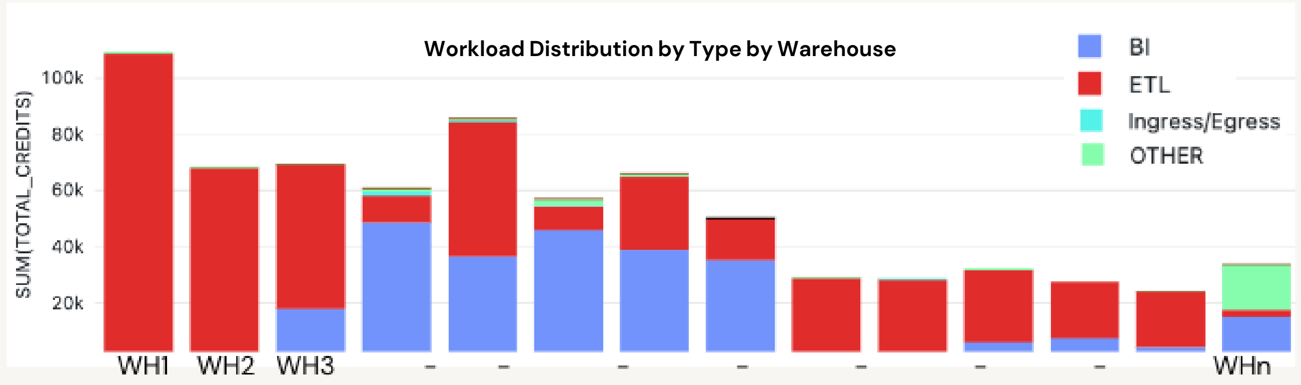
Implement AI.ML use cases using data in the Lakehouse

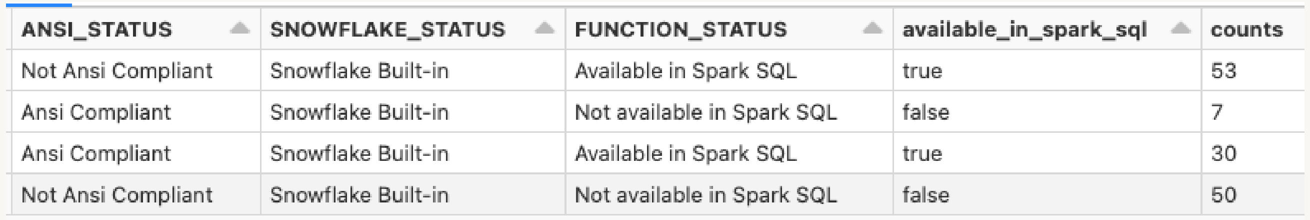
Technical Enablement, Onboarding and Change Management

Profile usage and assess size and complexity









ANSI_STATUS	SNOWFLAKE_STATUS	FUNCTION_STATUS	available_in_spark_sql	counts
Not Ansi Compliant	Snowflake Built-in	Available in Spark SQL	true	53
Ansi Compliant	Snowflake Built-in	Not available in Spark SQL	false	7
Ansi Compliant	Snowflake Built-in	Available in Spark SQL	true	30
Not Ansi Compliant	Snowflake Built-in	Not available in Spark SQL	false	50

Profile the usage in your environment to answer core questions

What is the breakdown of my warehouse usage by category?

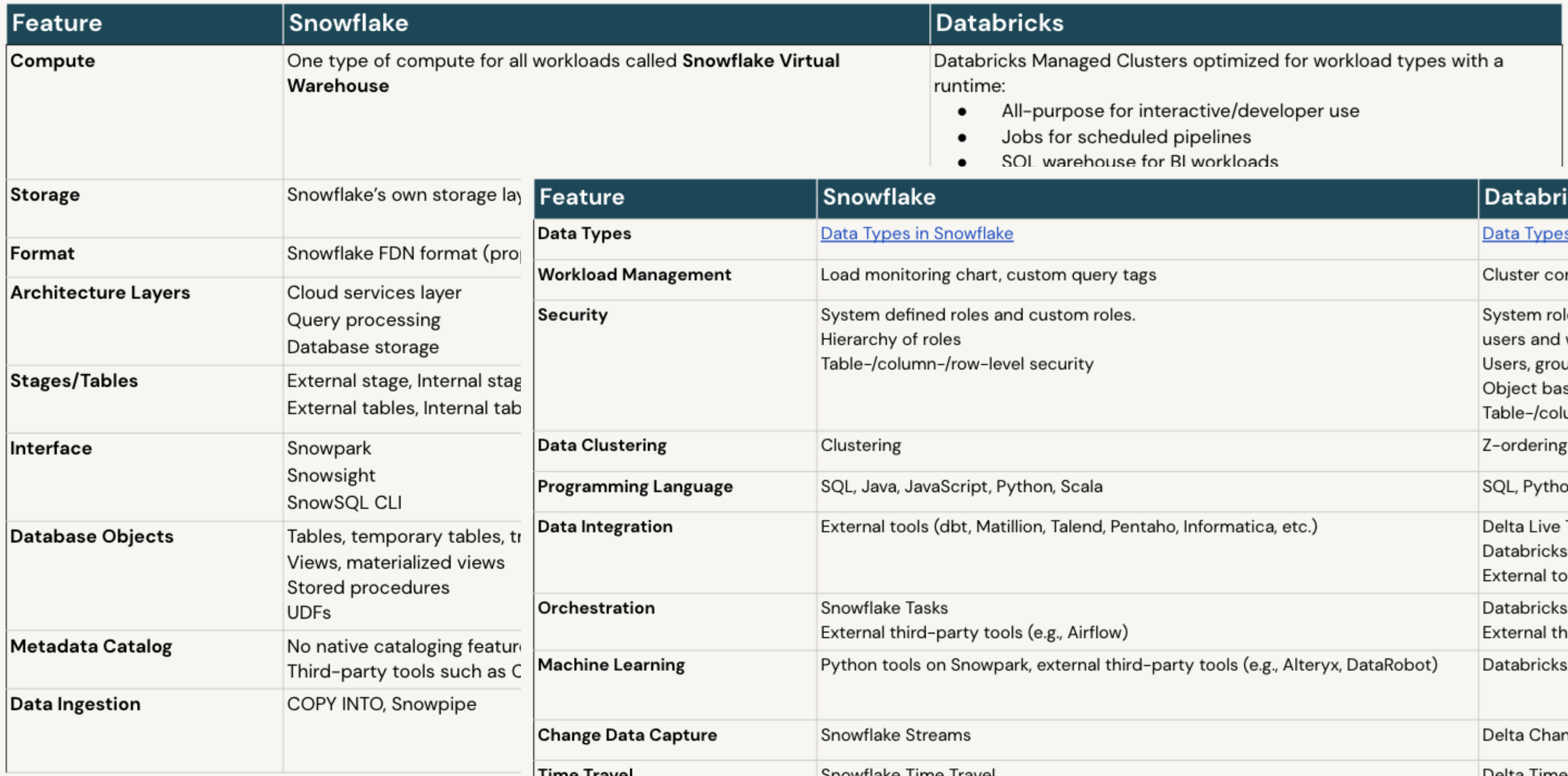
Where are these workloads running? (Which warehouse, which users?)

How do we expect these costs to grow over time?

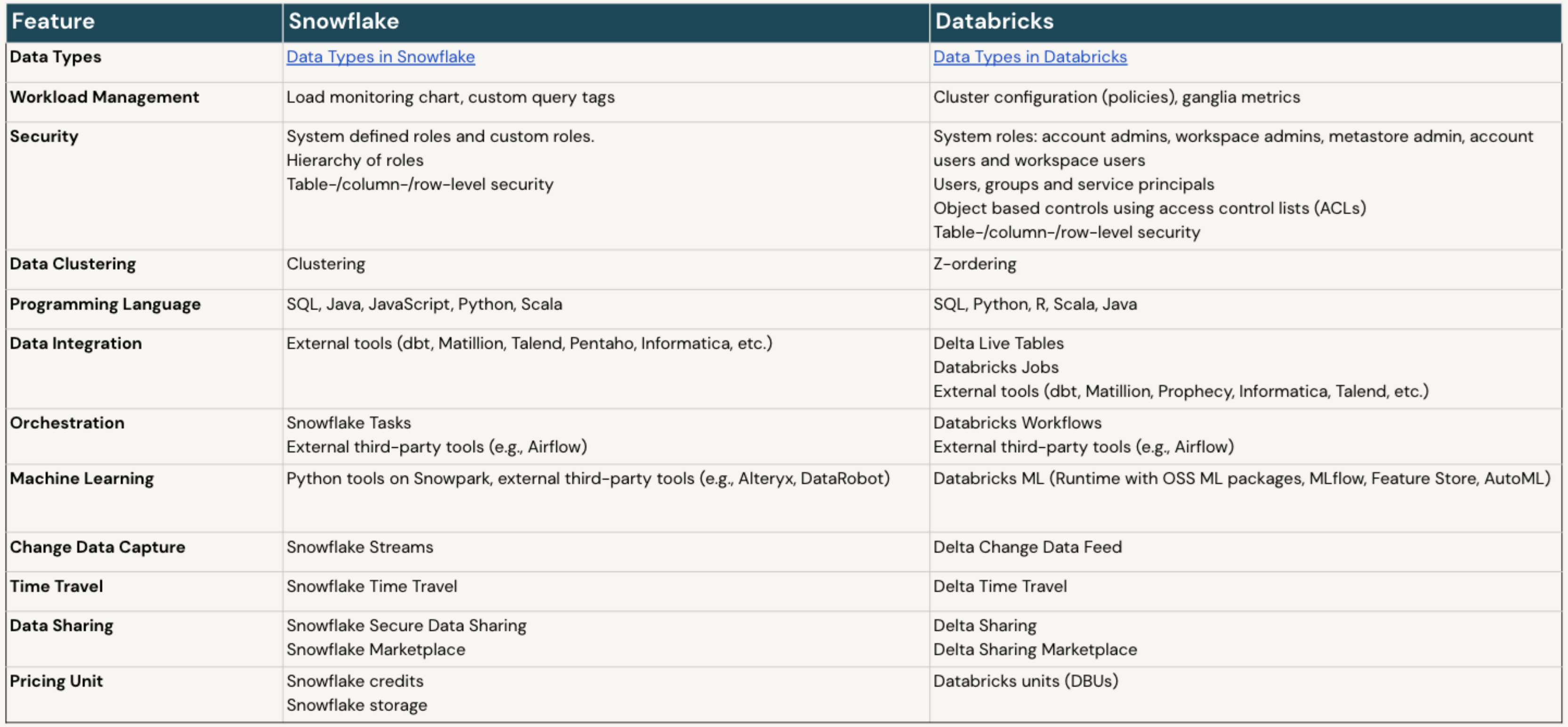
What are the long running jobs and queries?

What is the level of compatibility of the legacy SQL functions in Databricks?

Snowflake to Databricks: Technical Mapping



Feature	Snowflake	Databricks
Compute	One type of compute for all workloads called **Snowflake Virtual Warehouse**	Databricks Managed Clusters optimized for workload types with a runtime: • All-purpose for interactive/developer use • Jobs for scheduled pipelines • SQL warehouse for BI workloads
Storage	Snowflake's own storage lay	
Format	Snowflake FDN format (pro	
Architecture Layers	Cloud services layer Query processing Database storage	
Stages/Tables	External stage, Internal stag External tables, Internal tab	
Interface	Snowpark Snowsight SnowSQL CLI	
Database Objects	Tables, temporary tables, tr Views, materialized views Stored procedures UDFs	
Metadata Catalog	No native cataloging featur Third-party tools such as C	
Data Ingestion	COPY INTO, Snowpipe	



Feature	Snowflake	Databricks
Data Types	Data Types in Snowflake	Data Types in Databricks
Workload Management	Load monitoring chart, custom query tags	Cluster configuration (policies), ganglia metrics
Security	System defined roles and custom roles. Hierarchy of roles Table-/column-/row-level security	System roles: account admins, workspace admins, metastore admin, account users and workspace users Users, groups and service principals Object based controls using access control lists (ACLs) Table-/column-/row-level security
Data Clustering	Clustering	Z-ordering
Programming Language	SQL, Java, JavaScript, Python, Scala	SQL, Python, R, Scala, Java
Data Integration	External tools (dbt, Matillion, Talend, Pentaho, Informatica, etc.)	Delta Live Tables Databricks Jobs External tools (dbt, Matillion, Prophecy, Informatica, Talend, etc.)
Orchestration	Snowflake Tasks External third-party tools (e.g., Airflow)	Databricks Workflows External third-party tools (e.g., Airflow)
Machine Learning	Python tools on Snowpark, external third-party tools (e.g., Alteryx, DataRobot)	Databricks ML (Runtime with OSS ML packages, MLflow, Feature Store, AutoML)
Change Data Capture	Snowflake Streams	Delta Change Data Feed
Time Travel	Snowflake Time Travel	Delta Time Travel
Data Sharing	Snowflake Secure Data Sharing Snowflake Marketplace	Delta Sharing Delta Sharing Marketplace
Pricing Unit	Snowflake credits Snowflake storage	Databricks units (DBUs)

This shows example set of features between both platforms. A similar exercise comparing all features relevant for you environment must be performed. Taken from the snowflake-to-databricks-migration-guide

Strategies for Data Migration

One-time loads, catch-up loads , Real-time vs Batch Ingestion

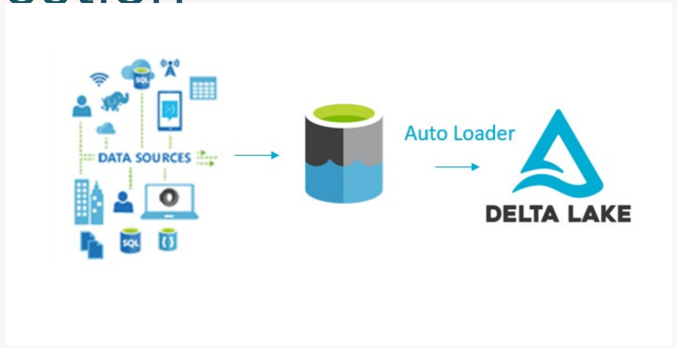
1. **[Recommended]** Leveraging Snowflake's COPY INTO command to push data out of Snowflake and into cloud storage (S3, ADLS, GCS) in *Parquet* format, then reading this data into Databricks using one of these methods:

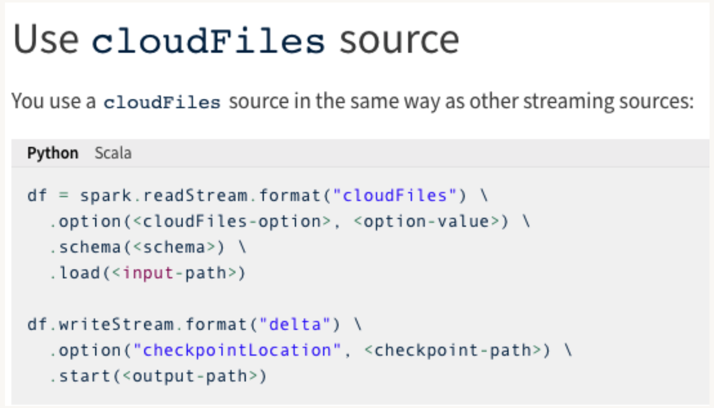
   - Auto Loader
   - Databricks COPY INTO command
   - Spark batch/streaming APIs

1. Leveraging the Snowflake Connector for Spark to read data from Snowflake and write to Delta Lake format tables.

1. Leveraging the Snowflake Stream Reader library to ingest data from Snowflake in a CDC fashion in cloud storage and into Delta Lake format tables using Auto Loader

1. For sophisticated synchronization requirements and accelerate data migration, leverage Real-Time CDC Ingestion using **Arcion** (now part of Databricks)





Use `cloudFiles` source

You use a `cloudFiles` source in the same way as other streaming sources:

Python Scala

```python
df = spark.readStream.format("cloudFiles") \
  .option(<cloudFiles-option>, <option-value>) \
  .schema(<schema>) \
  .load(<input-path>)

df.writeStream.format("delta") \
  .option("checkpointLocation", <checkpoint-path>) \
  .start(<output-path>)
```

A Well Architected Lakehouse

Dimensional Modeling

Staging

- Raw data in its original format (temporarily)

Ingestion

- Raw data converted to Delta (from Avro, CSV, parquet, XML, JSON format in Landing)

Integration –Physical data model

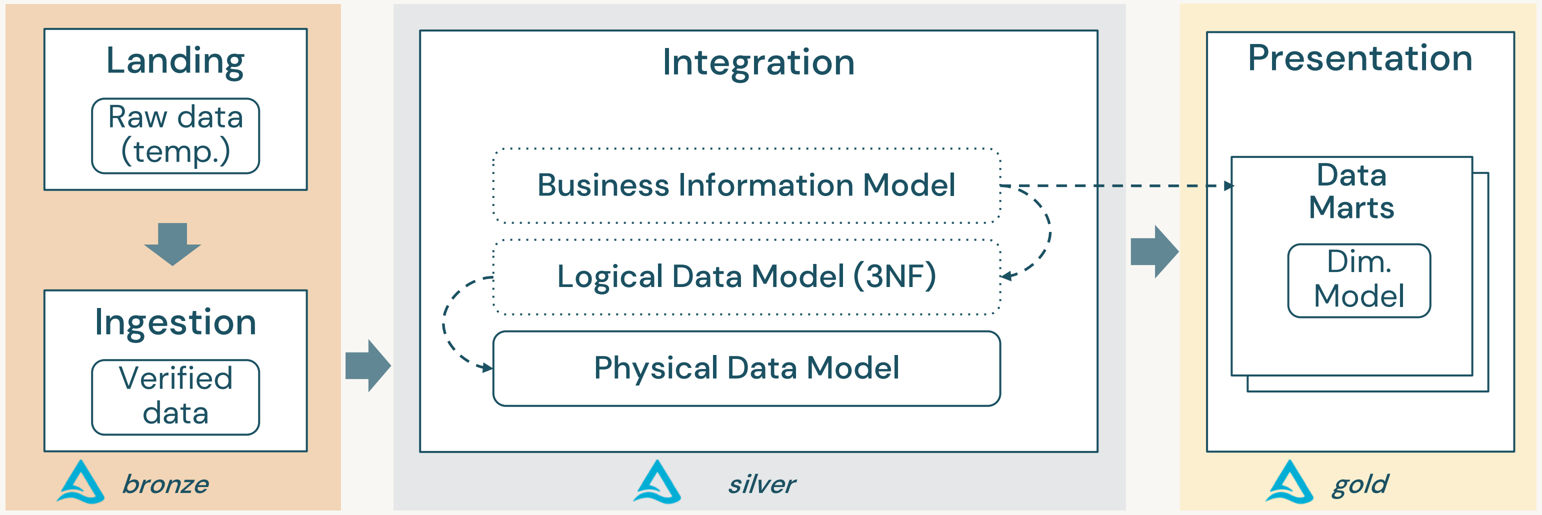
- Detailed information covering multiple subject areas
- Integrates all data sources
- Does not necessarily use a dimensional model but feeds dimensional models.

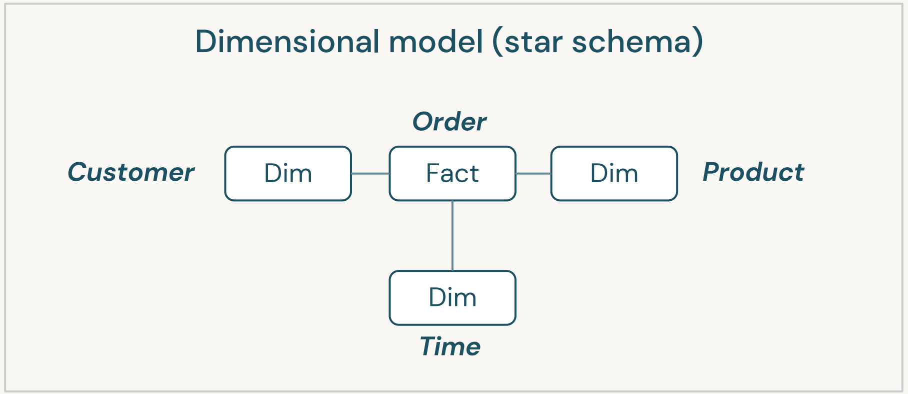
Data Mart

- Subset of the Integrated layer, sometimes or aggregated data
- Focus on dimensional modeling with star schema
- Typically oriented to a specific business line or team

Implementing Data Modeling Techniques in the Databricks Lakehouse Platform

Data Modeling Best Practices in the Databricks Lakehouse Platform





Code Migration

Core Components

PIPELINE MIGRATION

Consider re-engineering using Databricks Notebooks or Delta Live Tables with Databricks Workflows for orchestration

Or use GUI-based ETL tool Partners: Matallion, Prophecy, etc.

If continuing to use external tools (airflow, Matallion, dbt) - repoint existing pipelines to Databricks SQL Warehouses

MANAGING INGESTION

Simply re-configure ingestion pipelines (custom or tools such as Fivetran) to save data in Delta format

Ingestion pipelines using Snowpipe can be replaced with Databricks Auto Loader

Native Spark integrations (i.e. Kafka) are leveraged to refactor the streams.

QUERY MIGRATION & REFACTORING

Spark SQL supports ANSI SQL allowing a lot of code to work as-is

Use automation tools from vendors (BladeBridge, LeapLogic) or open source tools such as SQLGlot

Identify code inefficiencies and refer to best join strategies and practices in Spark and Delta Lake

TESTING

Keep pipelines as-is to simplify testing and evolve pipelines after migration

Establish unit testing for sink-to-sink comparison

Run pipelines in parallel for a week or two to ensure successful migration

Report modernization to Databricks

Run Semantic Layer & Analytics directly on your data in the Lakehouse

As easy as repointing your reports to schemas in Databricks using JDBC/ODBC driver connections

Use Photon and Cloudfetch technology

Use PowerBI or Power BI Premium Large Models

Integration with Tableau

Integration with OLAP cube providers like Microstrategy, Atscale etc.

Reporting and Semantic layer partner options in Databricks partner connect

Power BI ATSCALE TIBCO Spotfire ThoughtSpot

tableau Qlik MicroStrategy Looker

Walkthrough

Summary

Delivering a Successful Migration

PROFILE

"workload types, size, use cases, complexity, costs, roadmap, dependencies, integrations,...."

PLATFORM

"governance with Unity Catalog, open format, native orchestration, ingestion with DLT,...."

PLAN

"prioritize use cases, tangible deliverables, automation, training and change management,..."

Value realized in migrating to Databricks

- Cost Effective
- Open formats & Data Sharing
- Real-time streaming
- Scalable Data Platform
- AI & Machine Learning

databricks Lakehouse

All your data: Structured, Semistructured, Unstructured databricks