Using a Lakehouse Data Layer for Clinical Data Review

Paul Fioole
Joris Dekinder

28 Sept 2023

Peter Sas
Spring in the Forest

janssen | PHARMACEUTICAL COMPANIES OF Johnson & Johnson

Background

- Need for a datamart of our Data Review Environment project
  - Templates in visualization tool
  - Performance
  - Separating the datalayer from the visualization tooling
- Proposed solution based on Databricks
  - Flexible and future proof
- Potential to cover more than just Datamart
  - Data pipelines from ingestion to analysis & Data Science/ML

Purpose is to share our current experiences with Databricks Lakehouse

Storing clinical data – J&J approach

Best of breed

Modular implementation

Open environment

Ownership of data

Cloud first

Integrations

Performance

Validated environment

Future & functionality proof

Storing clinical data

- Traditionally: collecting files
- Data processing, transforming, checking, wrangling, reporting, visualizing
- Versions, comparisons, structural drift, data drift
- Traditional dataflow based on moving files, tracking versions, keeping copies in separate data containers (e.g. processing vs review)
- Data review calls for data warehouse functionalities

Modern technologies allow for bringing together best of both worlds: flexibility of files vs functionality of databases

Data Lake

Lakehouse

One platform to unify all of your data, analytics, and AI workloads

Data Warehouse

Data Lake

DELTA LAKE

An open approach to bringing **data management and governance** to data lakes

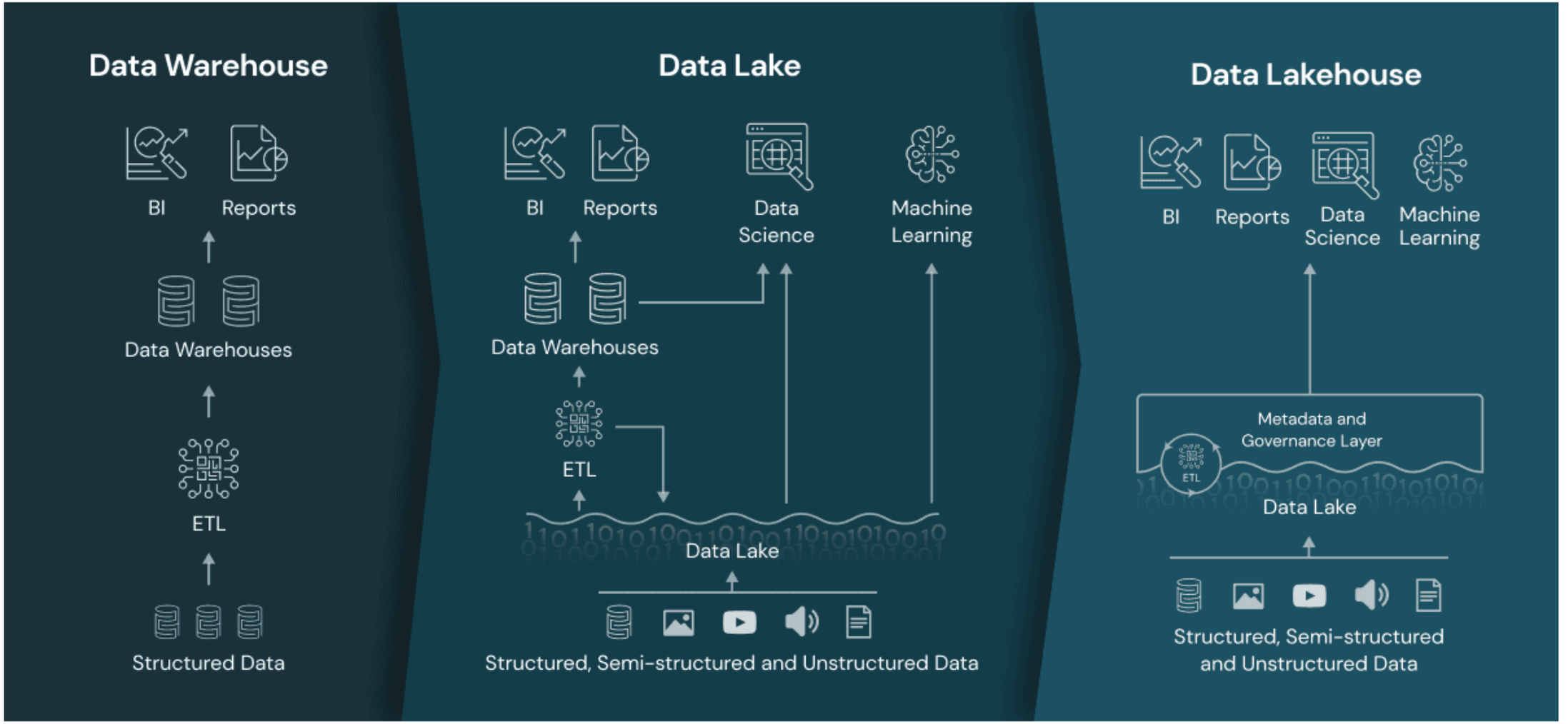
Better reliability with transactions

48x faster data processing with indexing

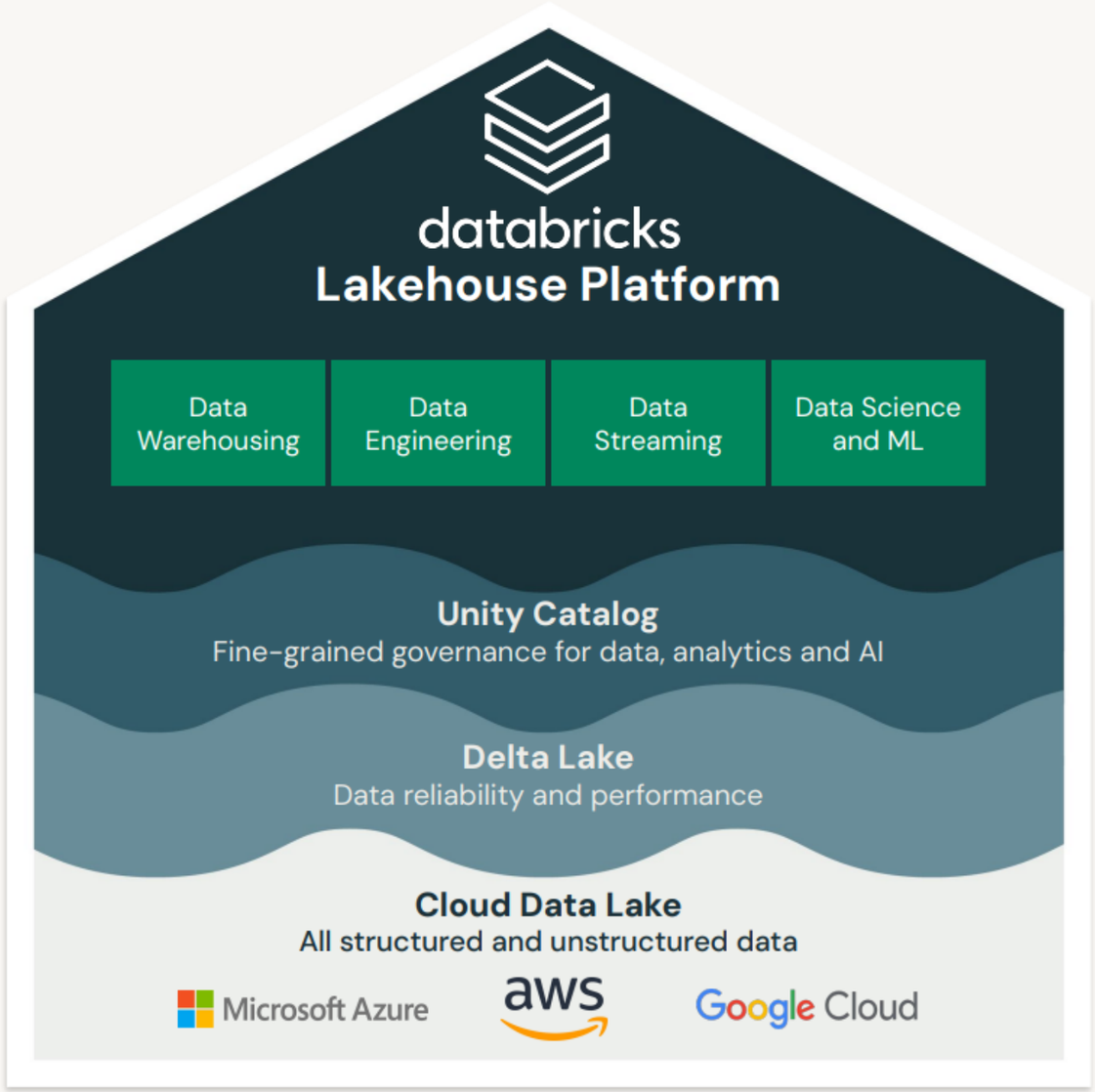
Data governance at scale with fine-grained access control lists

Data Warehouse

Lakehouse vs datalake vs database



Databricks architecture



Databricks Lakehouse Platform

Simple

Unify your data warehousing and AI use cases on a single platform

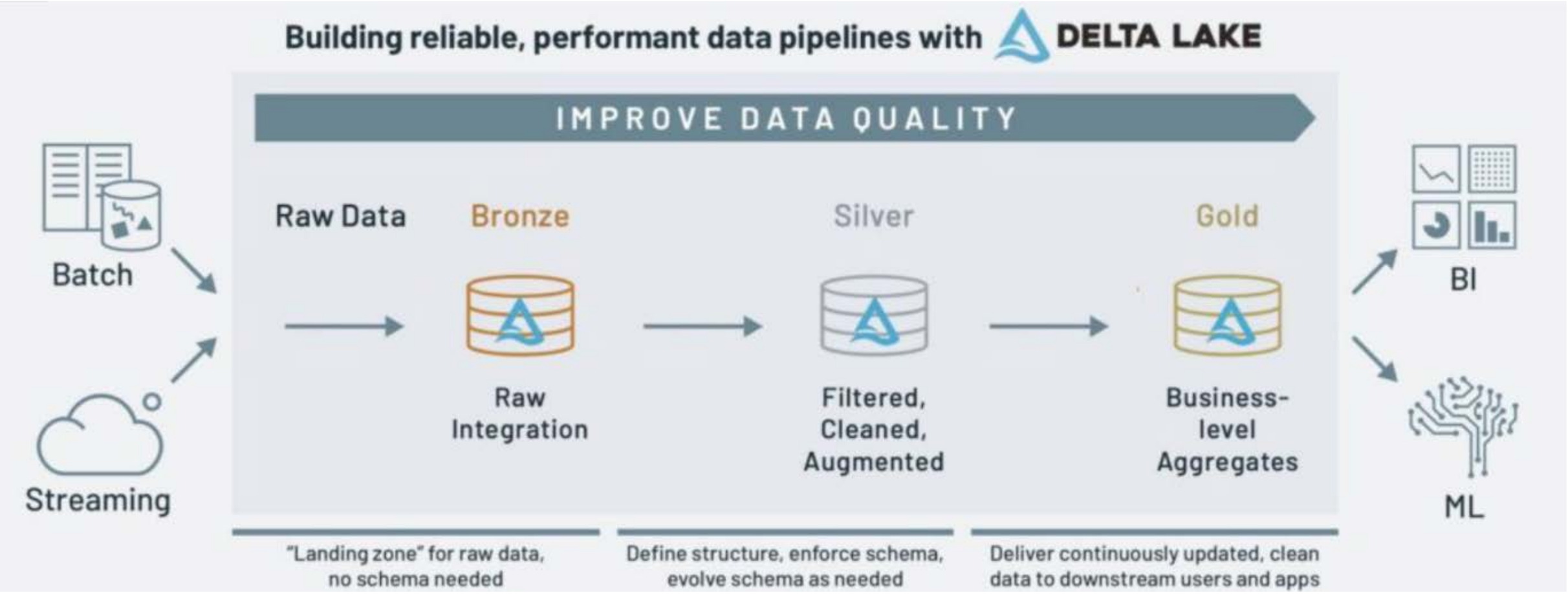
Multicloud

One consistent data platform across clouds

Open

Built on open source and open standards

Medallion architecture



Imported / RAW data

Versioned RAW data, id's

Versioned converted Data with provenance

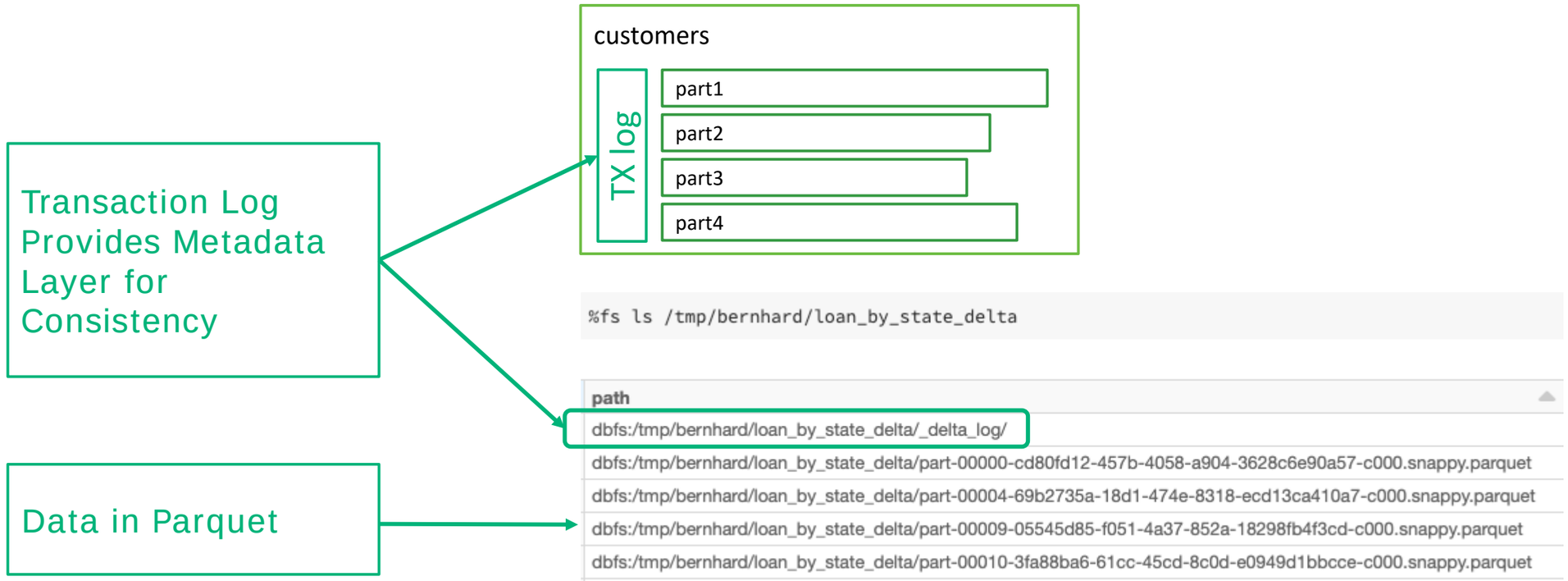
Current reporting datasets

Consumption (incl external)

Leveraging Delta Lake internals

- Delta Lake is based on cost-effective AWS S3 storage
- Delta tables
  - Parquet format – Columnar structure
  - maintaining delta's
- ACID transactions
- Time travel and rollback

Delta tables

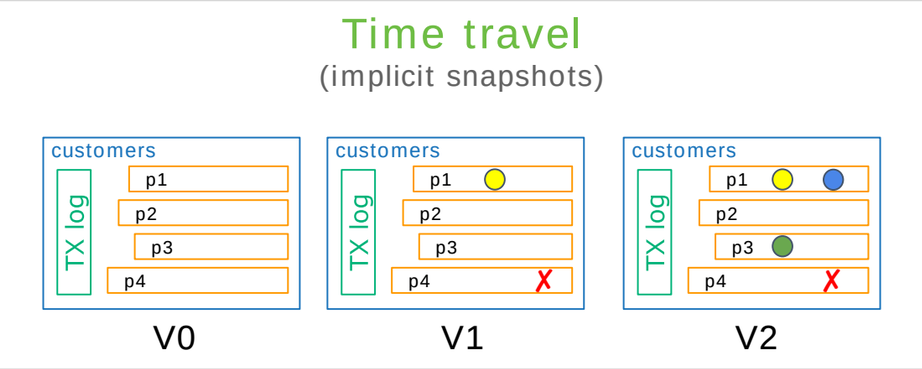


https://databricks.com/blog/2019/08/21/diving-into-delta-lake-unpacking-the-transaction-log.html

Time Travel

Key Features

- Perform “point in time” queries on your data
- See how it has evolved over time

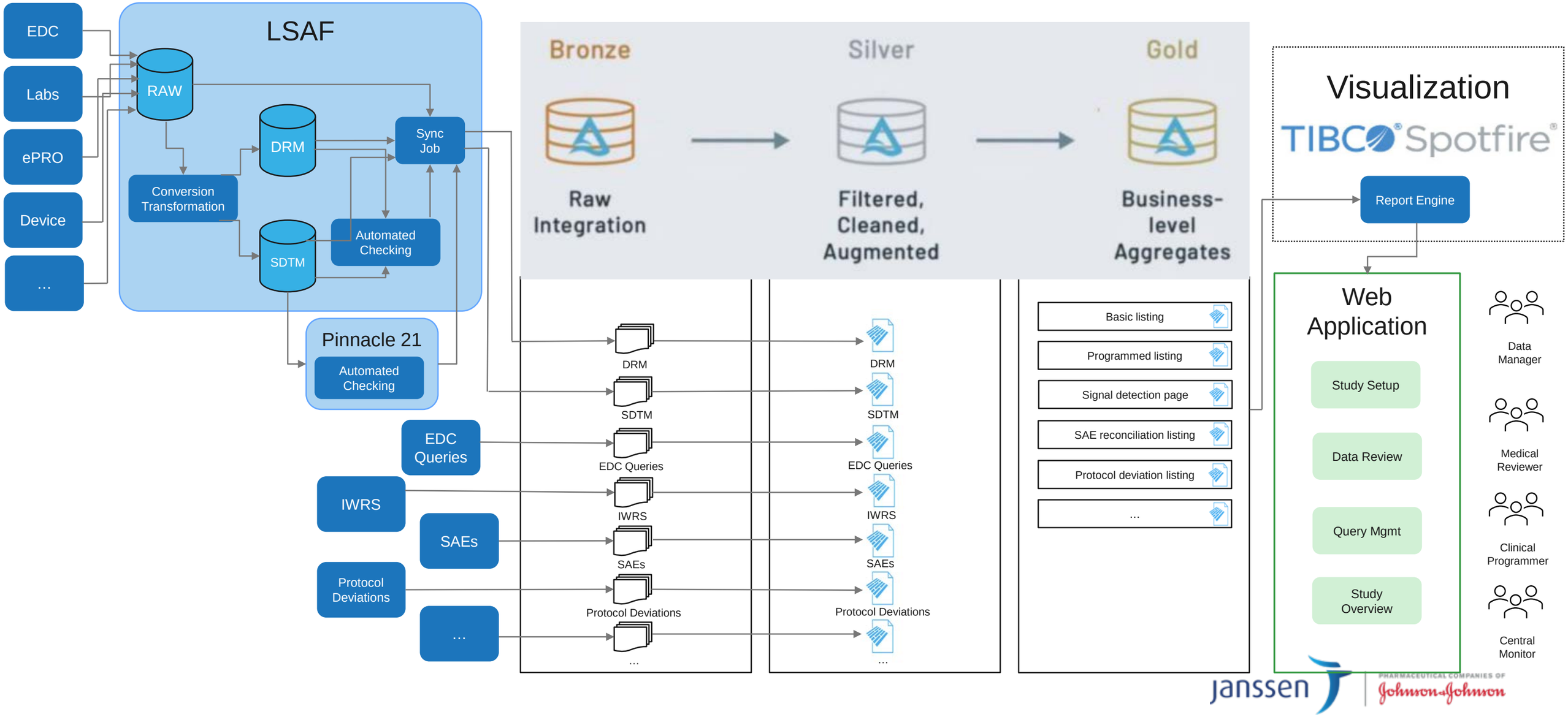


Syntax

```
SELECT *
FROM customers -- Delta table
TIMESTAMP AS OF
[timestamp_expression]
```

```
SELECT *
FROM customers -- Delta table
VERSION AS OF [version]
```

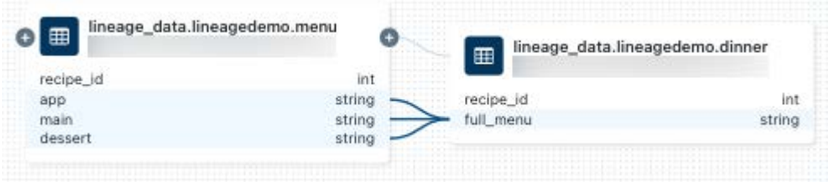
Our current dataflow in Databricks



Leveraging databricks: provenance

Data Lineage

- Table-level and column level are natively supported by Databricks



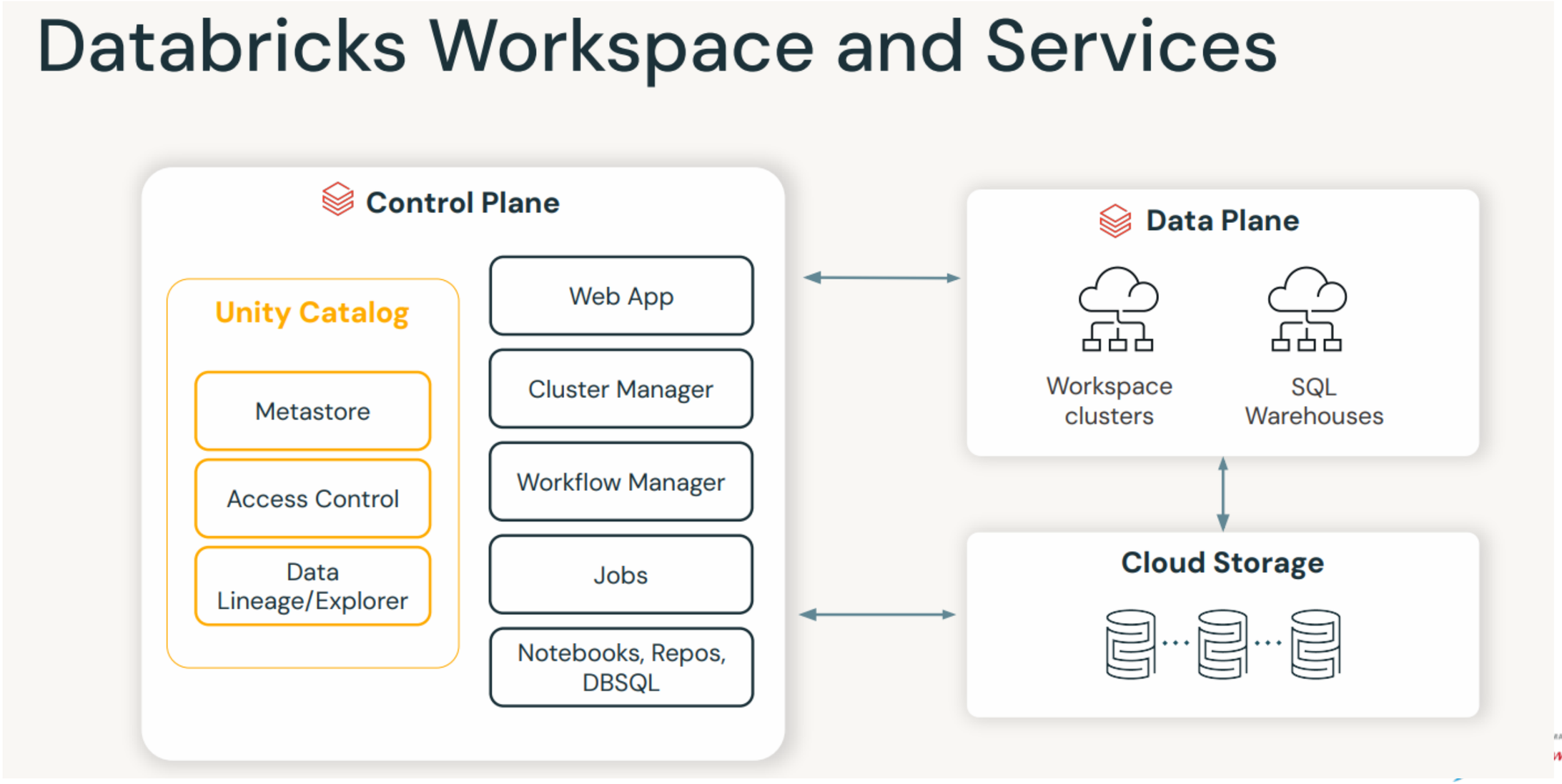
- Row-level required to show raw data and link inquiries to source records

Data versioning

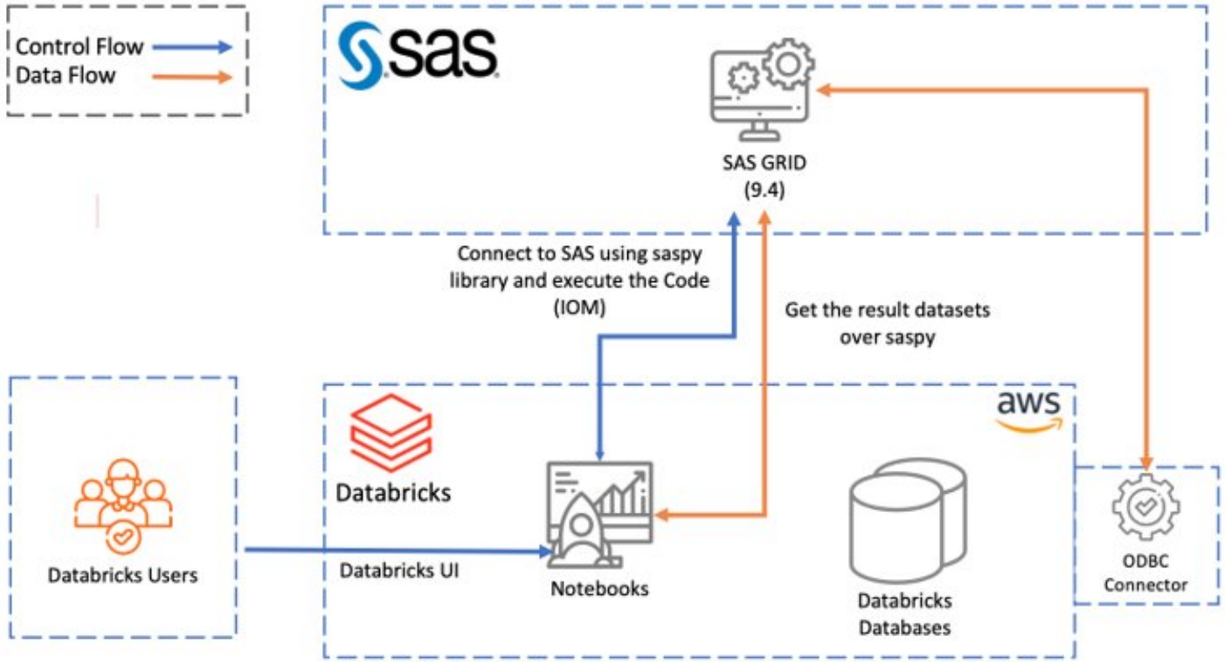
- Row level is natively supported with time travel and change data capture

- Cell-level required for reviewing incremental updates

Leveraging databricks



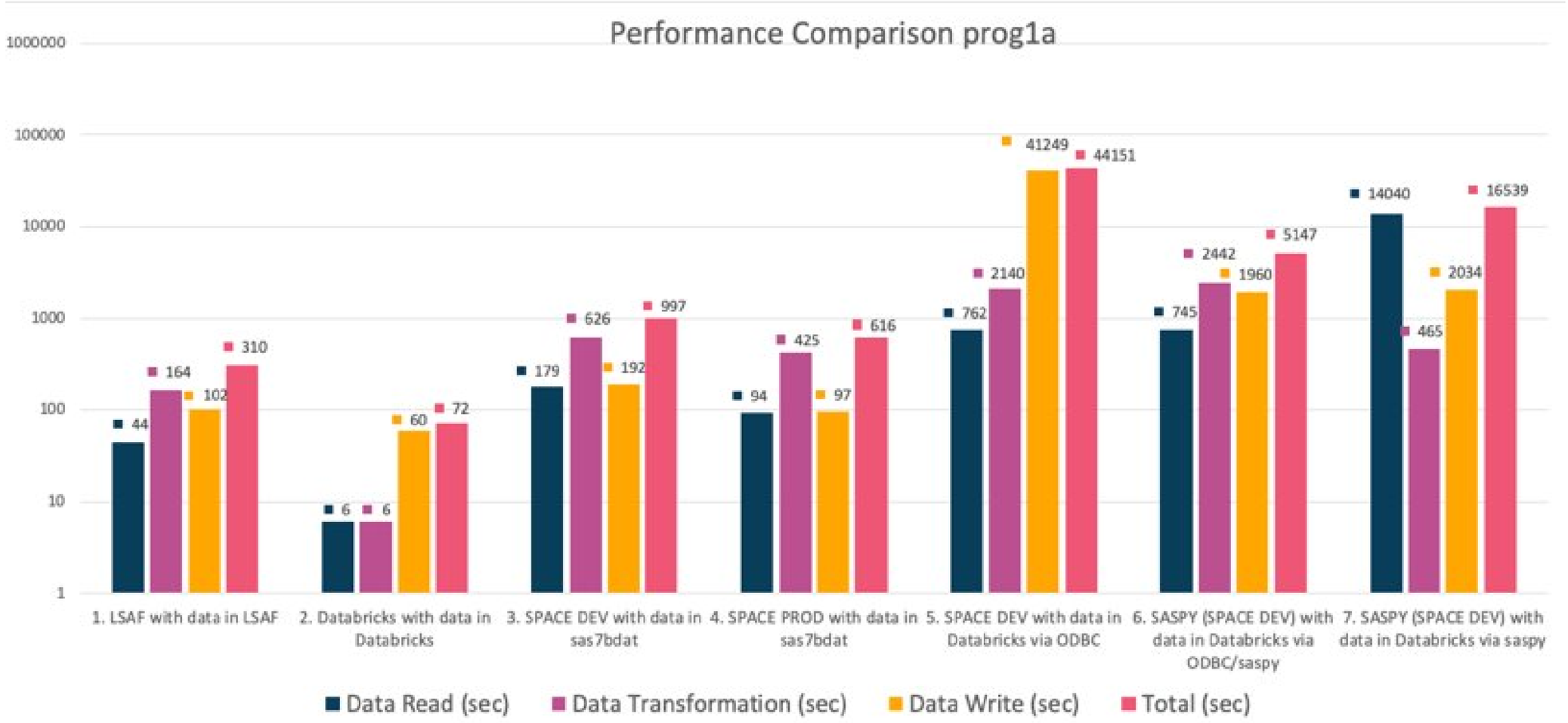
SAS



- Large base of clinical programs and processes
- Integration and performance tests
- Straightforward integration using ODBC
  - Write performance limitations
- Preliminary conclusions
  - LSAF performed much better than SAS SPACE GRID on Databricks (ODBC)
    - Look into custom integration opportunities / eliminate jumpbox architecture
    - Same High speed network
  - Functional Integration Databricks – SAS Grid works well
  - Native databricks outperforms both LSAF and SAS grid in the tests

https://www.databricks.com/blog/2022/03/16/how-to-speed-up-data-flow-between-databricks-and-sas.html

Performance comparisons



Current experiences

- Versioning
  - Separated increments -> time travel
- Notebooks & Workflows
  - Multiple languages, repositories
  - Easy to deploy / maintain
- Data Provenance (Lineage + History)
  - Data provenance : Change data feed / URECID composite key
  - Databricks native functionality vs specific business needs
- Pending full live performance testing
- Costing pattern
  - Quite promising
  - Consumption based – many tuning options


janssen
PHARMACEUTICAL COMPANIES OF
Johnson&Johnson