Delta Lake & Databricks

BI-BIG, 29.4.2025, Jan Lukány

DATAMOLE

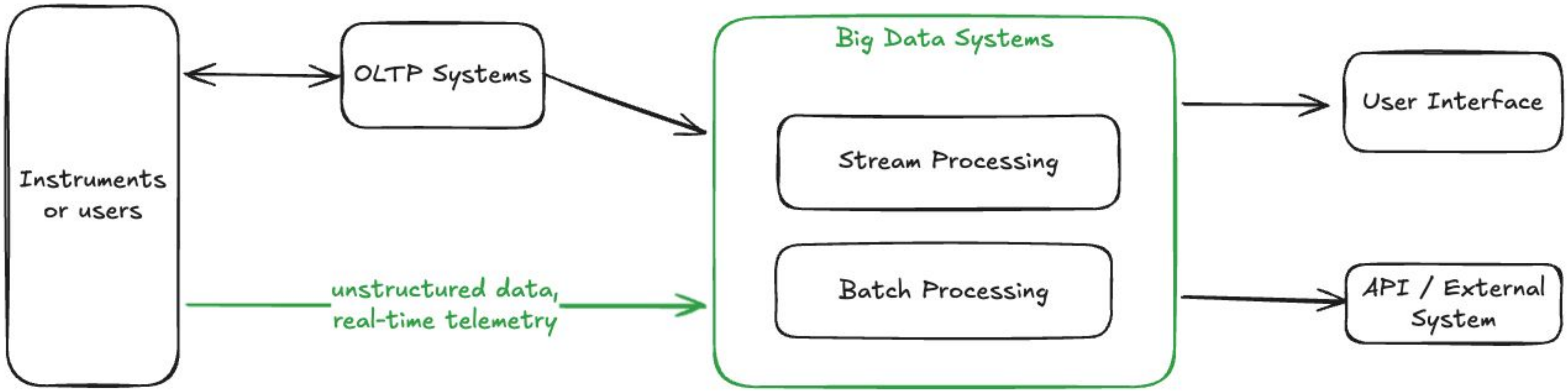
About Speaker

- Jan Lukany
- lukany.jan@gmail.com
- https://www.linkedin.com/in/jan-lukany/
- CTU FIT 2014-2020 (Bc Theoretical Informatics, Mgr Knowledge Engineering)
- Datamole 2015-
  - 2015-2017 Junior Software Developer
  - 2017-2020 Data Scientist
  - 2020-2024 ML Engineer
  - 2024- ML Engineering Team lead

DATAMOLE

datamole.ai

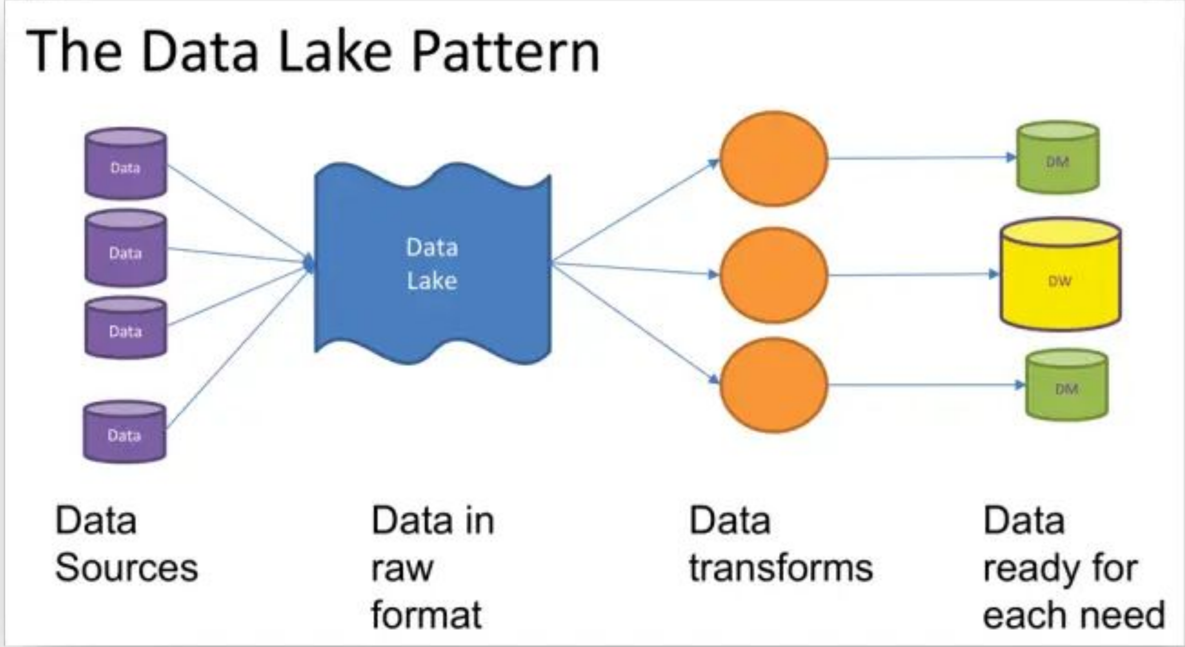
a Czech **data and AI company** of **80+ people** founded in **2015.** We **develop custom AI, IoT & UI solutions** that innovate industrial companies worldwide – in agriculture, machinery or biotech.


Big Data Systems
Instruments or users
OLTP Systems
unstructured data, real-time telemetry
Stream Processing
Batch Processing
User Interface
API / External System

DATAMOLE

This Lecture

- Delta Lake – *Reliable* Storage Protocol/Format
  - Data Lake with ACID
  - Core concepts and practical usage (Python Polars and Pyspark)
- Databricks – Data Intelligence Platform (Spark + Delta Lake)
  - Practical example with free Community Edition


The Data Lake Pattern
Data
Data
Data
Data
Data
Lake
DM
DW
DM
Data
Sources
Data in
raw
format
Data
transforms
Data
ready for
each need

DATAMOLE

DELTA LAKE

Delta Lake

Key Features

ACID Transactions
Protect your data with serializability, the strongest level of isolation

Scalable Metadata
Handle petabyte-scale tables with billions of partitions and files with ease

Time Travel
Access/revert to earlier versions of data for audits, rollbacks, or reproduce

Open Source
Community driven, open standards, open protocol, open discussions

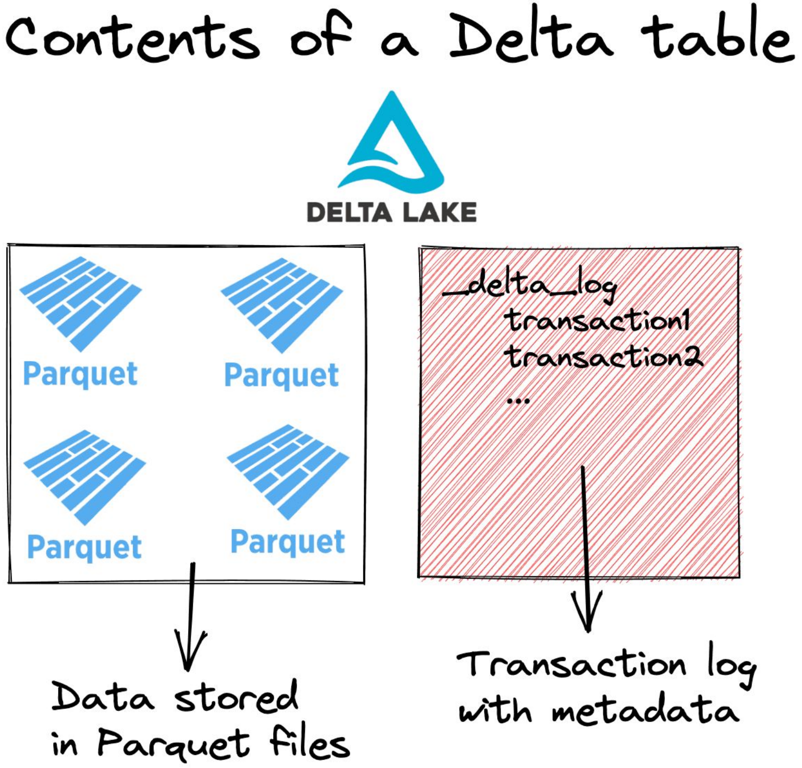
Unified Batch/Streaming
Exactly once semantics ingestion to backfill to interactive queries

Schema Evolution / Enforcement
Prevent bad data from causing data corruption

Audit History
Delta Lake log all change details providing a fill audit trail

DML Operations
SQL, Scala/Java and Python APIs to merge, update and delete datasets

Delta Lake – Delta Table



Delta Lake – Demo

Delta Lake – on cheap cloud storages

- Azure Blob Storage
- Amazon S3
- Google Cloud Storage

Delta Lake – ACID

- **A**tomicity – process all or nothing
- **C**onsistency – always valid state
- **I**solation – read while writing, safe multiple writes
- **D**urability – transactions are permanent

Guarantees **over one table only**

Delta Lake – scalability

- Data file skipping using statistics (e.g. min/max)
- Columnar format of parquets
  - Read only parts of data files
- Optimization techniques
  - Storing similar data close together
  - Liquid clustering, Z-Ordering, partitioning

Delta Lake – time travel

- transaction log
- selecting specific version from the log
- optimize and vacuum operations
  - optimize – compact multiple small files into large ones
  - vacuum – delete unused data files

```
df = spark.read.format("delta").option("versionAsOf", 0).load("data/delta_census")
```

Delta Lake – Change Data Capture/Feed

- Tracking row-level changes for updates and deletes
  - no need for append-only (no update/delete) – transaction log is enough

SQL

```sql
ALTER TABLE myDeltaTable SET TBLPROPERTIES (delta.enableChangeDataFeed = true)
```

```python
# providing a starting version
spark.readStream.format("delta") \
  .option("readChangeFeed", "true") \
  .option("startingVersion", 0) \
  .table("myDeltaTable")
```

DATAMOLE

Delta Lake – schema evolution and enforcement

- allow adding new columns and changing (some) data types
- prevent bad data from being written

Delta Lake – Unified Batch and Streaming

- Batch – historical data processing in bulk
- Streaming – real-time data processing
- Same API for both

Example Batch Processing with Spark

```
# Read data from Delta table
input_df = spark.read.format("delta").load("input-table-path")

# Perform simple processing (e.g., filtering and adding a column)
processed_df = (
    input_df
    .filter(col("column_name") > 100)
    .withColumn("new_column", col("column_name") * 2)
)

# Write processed data back to a Delta table
(
    processed_df
    .write.format("delta")
    .mode("overwrite")
    .save("output-table-path")
)
```

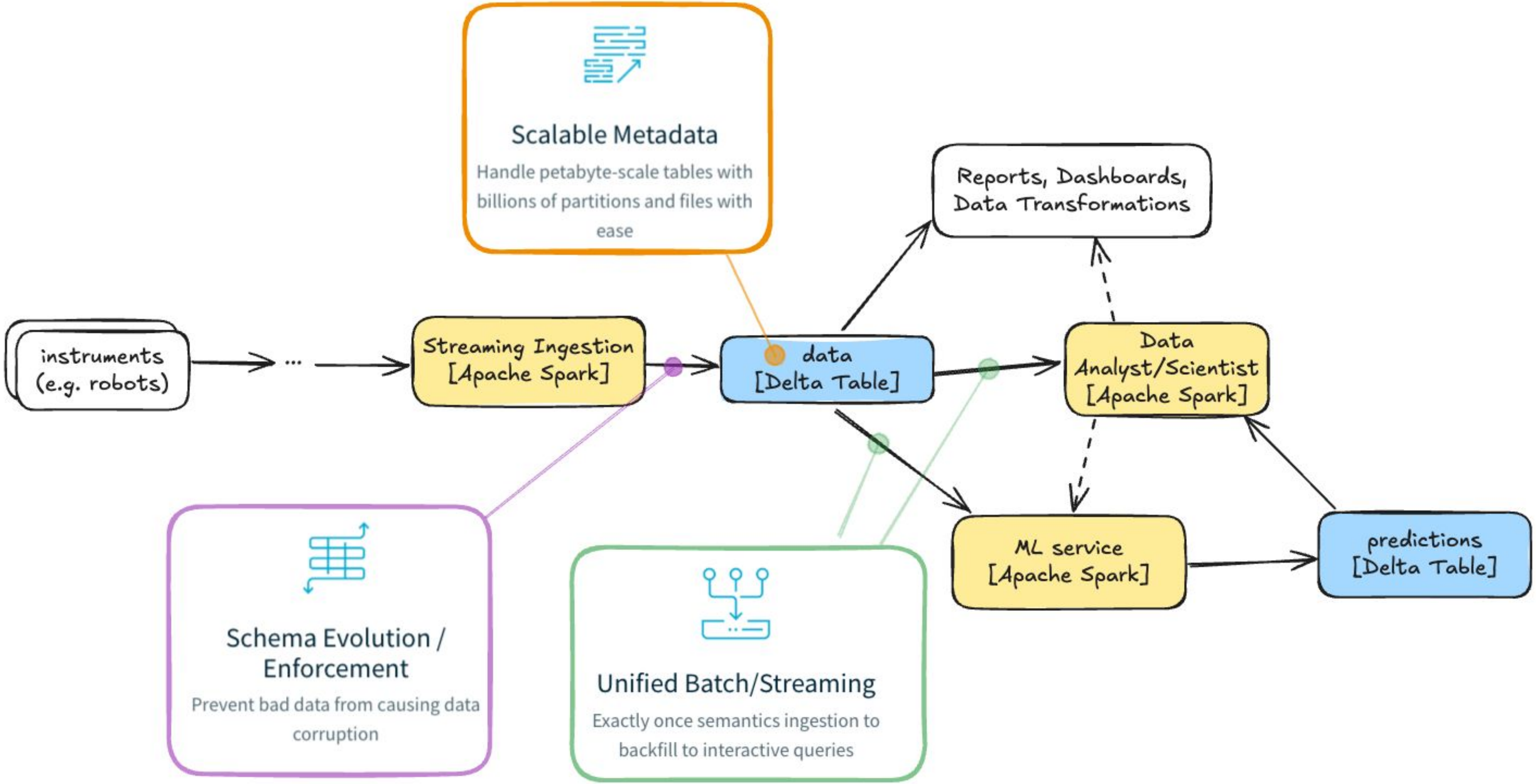
Example Stream Processing with Spark

```python
# Read data from Delta table as a streaming source
input_stream = spark.readStream.format("delta").load("input-table-path")

# Perform simple processing (e.g., filtering and adding a column)
processed_stream = (
    input_stream.filter(col("column_name") > 100)
    .withColumn("new_column", col("column_name") * 2)
)

# Write processed data to Delta table as a streaming sink
query = (
    processed_stream.writeStream.format("delta")
    .outputMode("append")
    .option("checkpointLocation", "/path/to/checkpoint-dir")
    .start("output-table-path")
)
```

DATAMOLE


Scalable Metadata
Handle petabyte-scale tables with billions of partitions and files with ease
Reports, Dashboards, Data Transformations
instruments (e.g. robots)
...
Streaming Ingestion [Apache Spark]
data [Delta Table]
Data Analyst/Scientist [Apache Spark]
ML service [Apache Spark]
predictions [Delta Table]
Schema Evolution / Enforcement
Prevent bad data from causing data corruption
Unified Batch/Streaming
Exactly once semantics ingestion to backfill to interactive queries

DATAMOLE

databricks

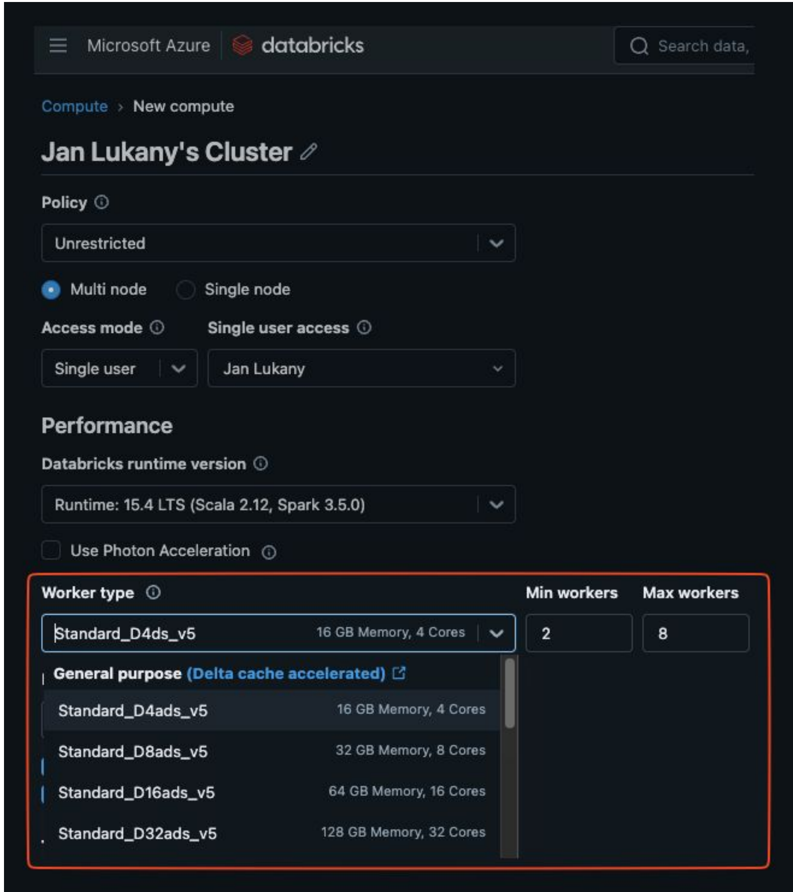
DATAMOLE

Databricks

- Compute
  - Apache Spark cluster management
  - Built-in orchestration for workflows
- Data Lakehouse
  - Delta Lake as default storage layer
  - Unity Catalog – “one” database, data governance
    - Delta tables
    - External volumes
    - Federated access
- Other
  - Development workspace (similar to Jupyter Lab) with collaborative features
  - Machine learning support (training, deployment)
  - Native integrations with major cloud providers (Azure, AWS, GCP)

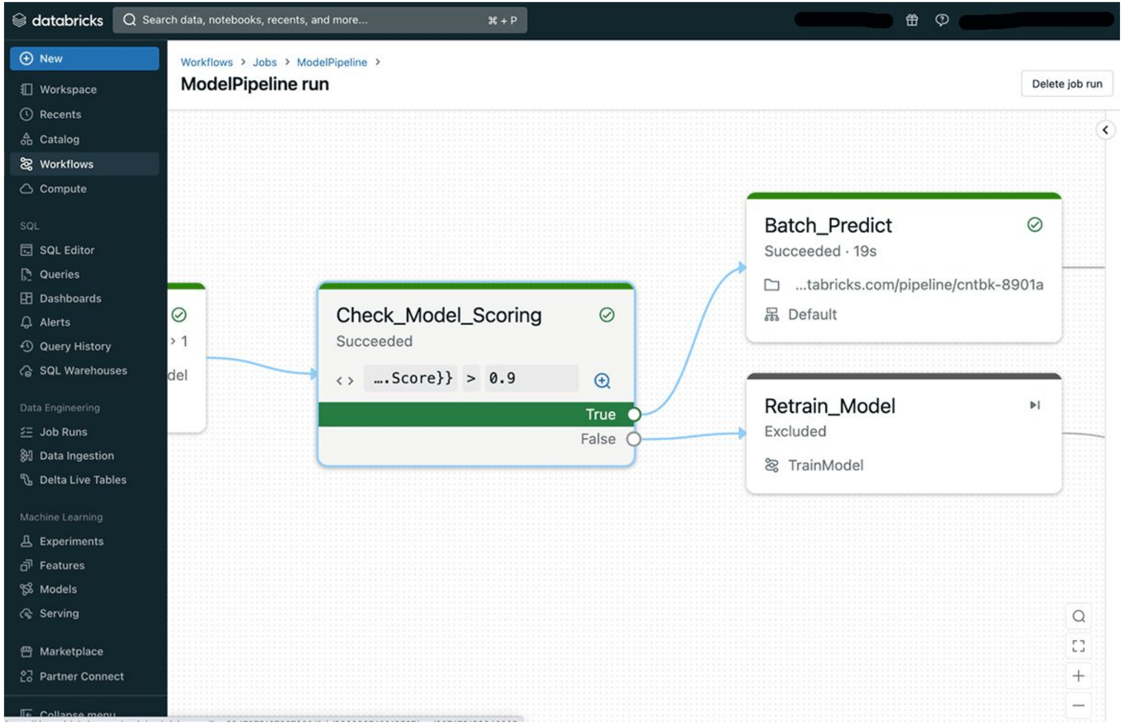
Databricks – Apache Spark cluster management

- Azure VMs
- AWS EC2
- Google Compute Engine



DATAMOLE

Databricks – Workflows (job orchestration)



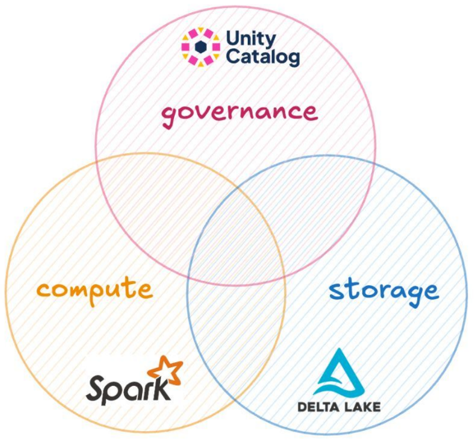
DATAMOLE

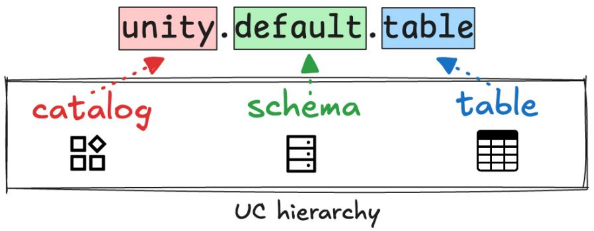
Databricks – Delta Tables on Cloud

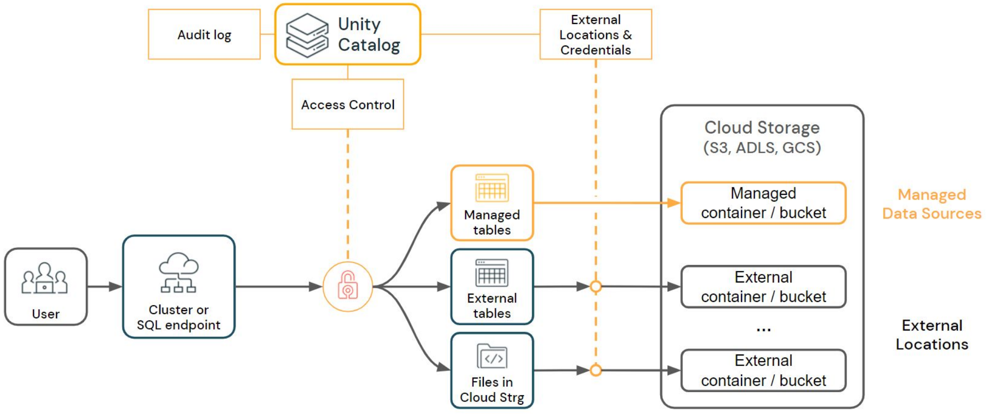
- Amazon S3
- Azure Blob Storage
- Google Cloud Platform

Databricks – Unity Catalog

- "one" database from the perspective of consumers
- data governance at one place
- can include e.g.:
    - Delta tables
    - ML models
    - Unstructured data on cloud storages
    - External databases







DATAMOLE

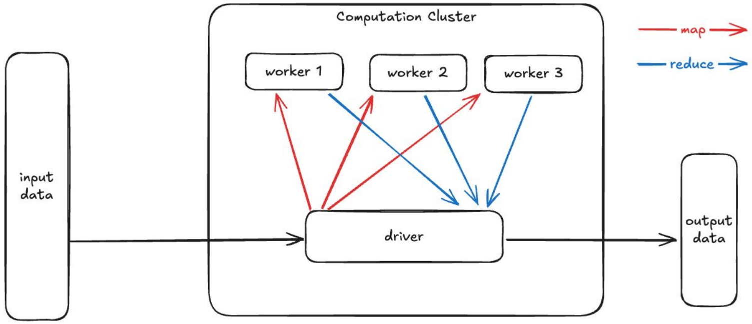
Databricks – Demo (community edition)

Summary

- Delta Lake
  - Reliable, ACID
  - Integration: Apache Spark, Python Polars, delta-rs
  - Open-source
- Unity Catalog
  - Data governance (e.g. easy access, discoverability)
  - Open-source
- Databricks
  - Cloud platform on top of Apache Spark, Delta Lake and Unity Catalog
  - Unity Catalog
  - ML experiments and serving
  - Closed-source / paid

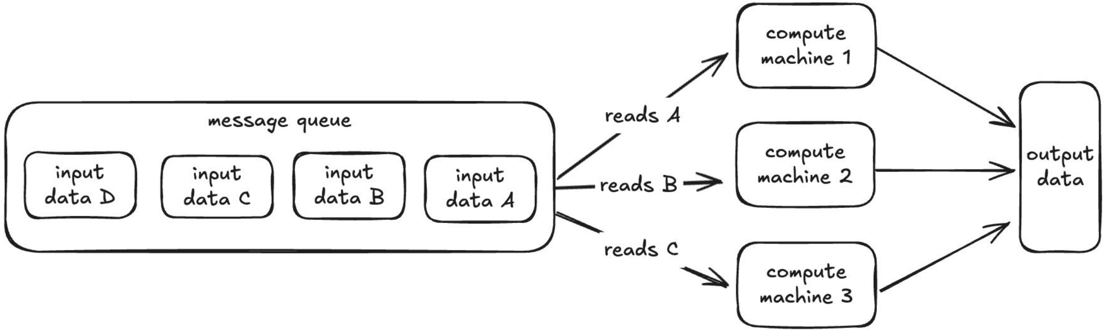
Q&A

Bonus: Map-Reduce vs Competing Consumers



Example tech stack:

- Apache Spark
- Delta Lake
- (Databricks)



Example tech stack:

- Azure Event Hub
- Kubernetes, Docker, Python
- Azure Tables (NoSQL)