databricks

University Alliance

Data Science at Scale

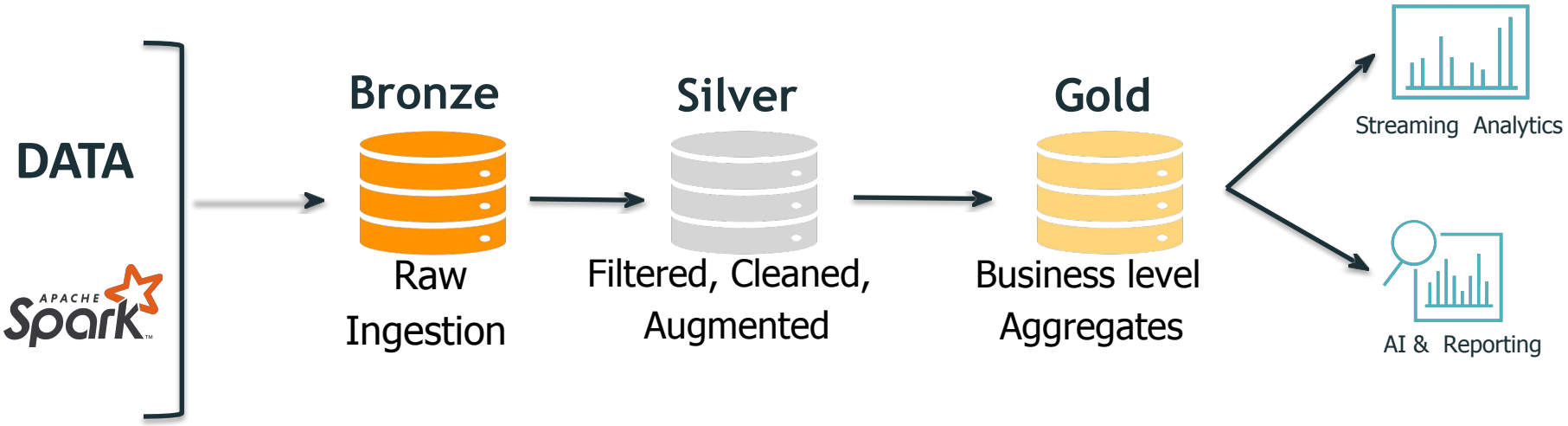
DSCC 202/402
March 16th 2022

Delta Lake Architecture Review

Elements of Delta Lake

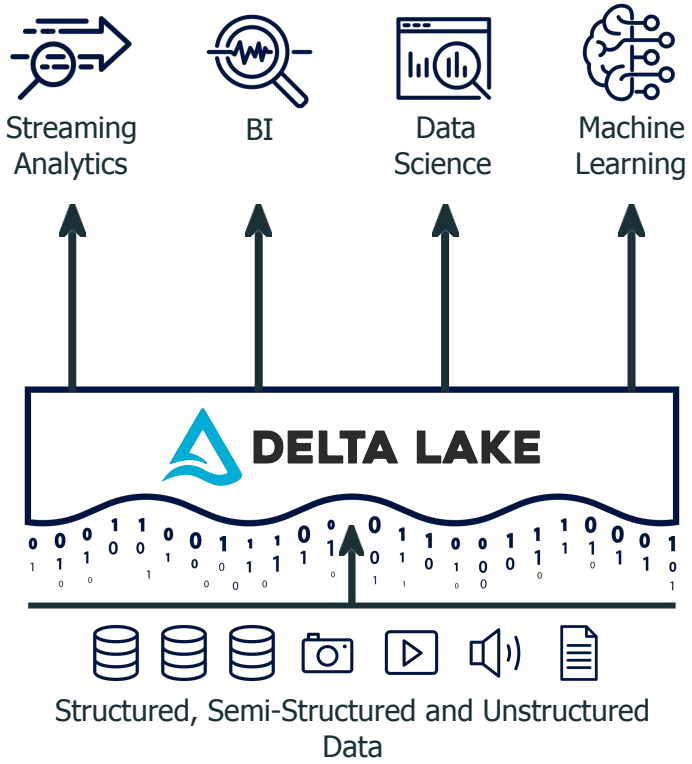
- Delta Architecture
- Delta Storage Layer
- Delta Engine

Delta architecture



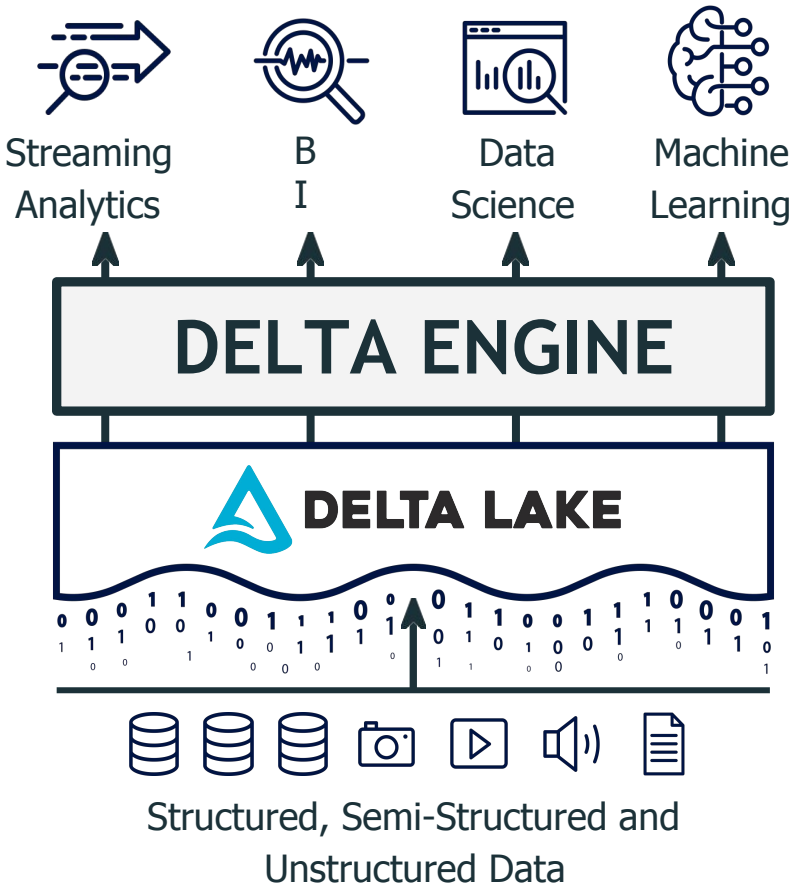
Delta Storage Layer

- Guarantee data is consistent
- Track metadata
- Automatically handle variations in schema
- Enables version control and rollbacks
- Merge and update data as it arrives



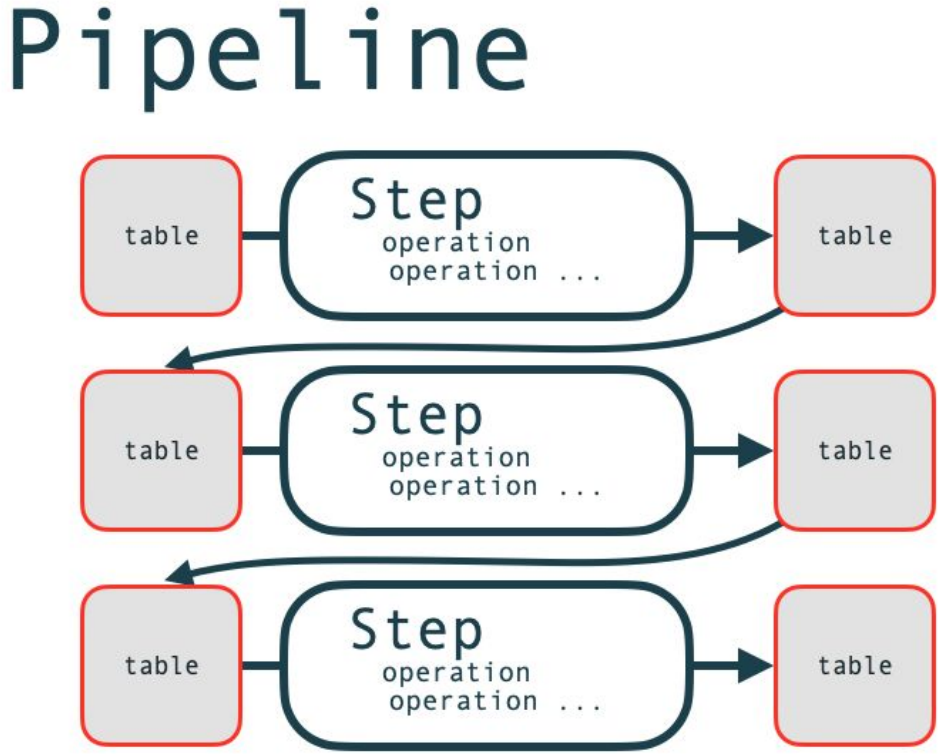
Delta Engine

- File management optimizations
- Performance optimization Caching
- Dynamic File Pruning
- Adaptive Query Execution



Writing software with Databricks

Pipelines-Steps-Operations



Data Pipelines on Databricks


```
├── plus
│   ├── 02_bronze_to_silver
...
    └── includes
        ├── configuration
        └── main
            └── python
                └── operations
```

The Includes Software Pattern

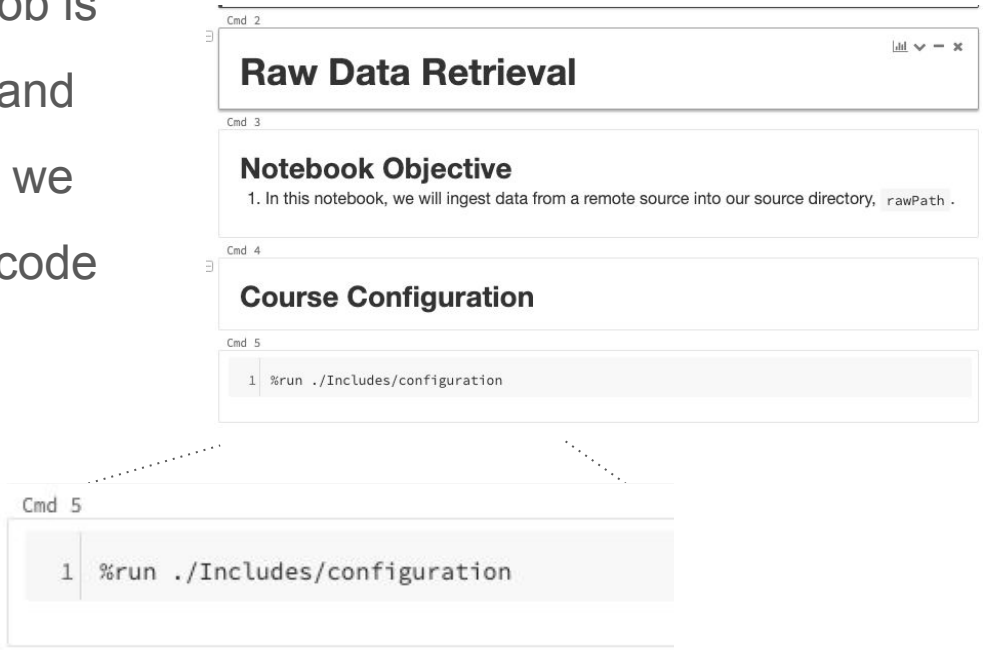
- Classically, the “Includes” pattern inserts the contents of another file into the source code

- In Python

```
from pyspark.sql import DataFrame
from pyspark.sql.functions import (
  col, current_timestamp, from_json,
  from_unixtime, lag, lead, lit,
  mean, stddev, max
)
```

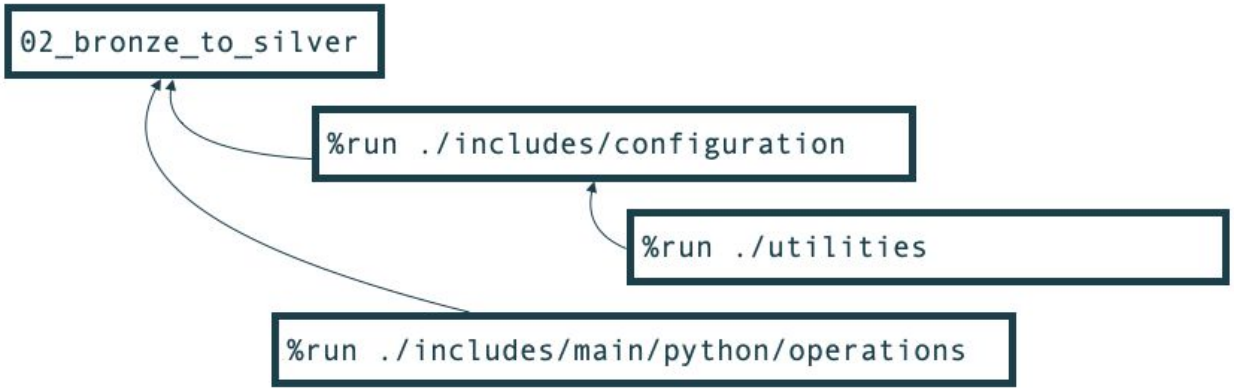
The Includes Software Pattern in Databricks

- The best practice in including source code in a Databricks job is to use the `%run` magic command
- In each of our job notebooks, we use this command to source code for the job



Inclusion Dependency

- Each step includes the `configuration` file
  - The `configuration` file includes the `utilities` file
- If necessary, the step includes the `operations` file



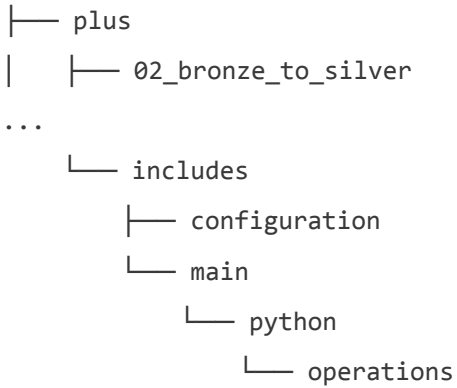
The configuration File

- The configuration file is used to do the following:
  - Define a unique username to be used across the project
  - Define the file pathways to be used across the project including:
    - storage locations for our Delta files
    - streaming checkpoints
  - Create and use a unique Database

```
├── plus
│   ├── 02_bronze_to_silver
...
    └── includes
        ├── configuration
        └── main
            └── python
                └── operations
```

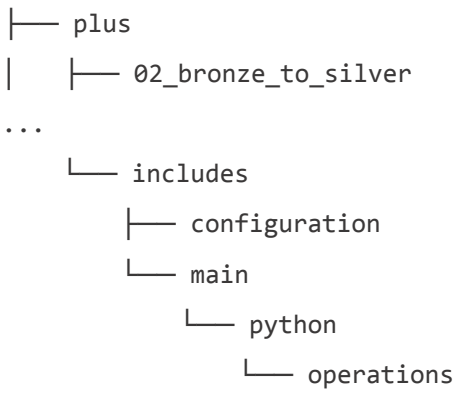
The `utilities` File

- The `utilities` file is used to define and source the following utility functions:
  - `retrieve_data`
    - used to ingest raw files into our system
  - `stop_all_streams`
    - stops all running streams
  - `stop_named_stream`
    - stops a running stream with a given name



The operations File

- The operations file is used to define and source the following composable operation functions:
  - create_stream_writer
  - read_stream_delta
  - read_stream_raw
  - update_silver_table
  - transform_bronze
  - transform_raw
  - transform_silver_mean_agg_last_thirty



Planning Your Data Pipeline: Moovio+

The raw data

- Multi-line JSON files
- Resemble the strings passed by Kafka
- Each file consists of five users:
  - Heart rate measured each hour, 24 hours a day, every day

Health tracker data sample


```
{"device_id":0,"heartrate":52.8139067501,"name":"Deborah Powell","time":1.5778368E9}
{"device_id":0,"heartrate":53.9078900098,"name":"Deborah Powell","time":1.5778404E9}
{"device_id":0,"heartrate":52.7129593616,"name":"Deborah Powell","time":1.577844E9}
{"device_id":0,"heartrate":52.2880422685,"name":"Deborah Powell","time":1.5778476E9}
{"device_id":0,"heartrate":52.5156095386,"name":"Deborah Powell","time":1.5778512E9}
{"device_id":0,"heartrate":53.6280743846,"name":"Deborah Powell","time":1.5778548E9}
```


Note that each line is a valid JSON object.

Health Tracker Data Schema

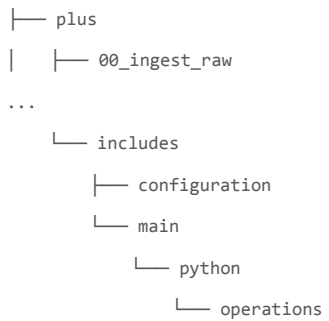
```
name: string
heartrate: double
device_id: long
time: float
```

Planning your data pipeline

- Where is the data coming from?
- How much data exists?
- What is this type of data?
- What are the SLA requirements around how it will be used?
- How frequently is it updated?
- What kind of inconsistencies or uncertainties might you anticipate?
- What might the raw → bronze → silver → gold levels look like?

The operations File

- The operations file is used to define and source the following composable pipeline functions:
  - create_stream_writer
  - read_stream_delta
  - read_stream_raw
  - update_silver_table
  - transform_bronze
  - transform_raw
  - transform_silver_mean_agg_last_thirty



Moovio Plus Delta Architecture