Databricks additions

Delta storage

Use of external blobs

Key vault integration

Delta storage (delta lake)

Open Source Reliability for Data Lake with Apache Spark by Michael Armbrust

Clip slide

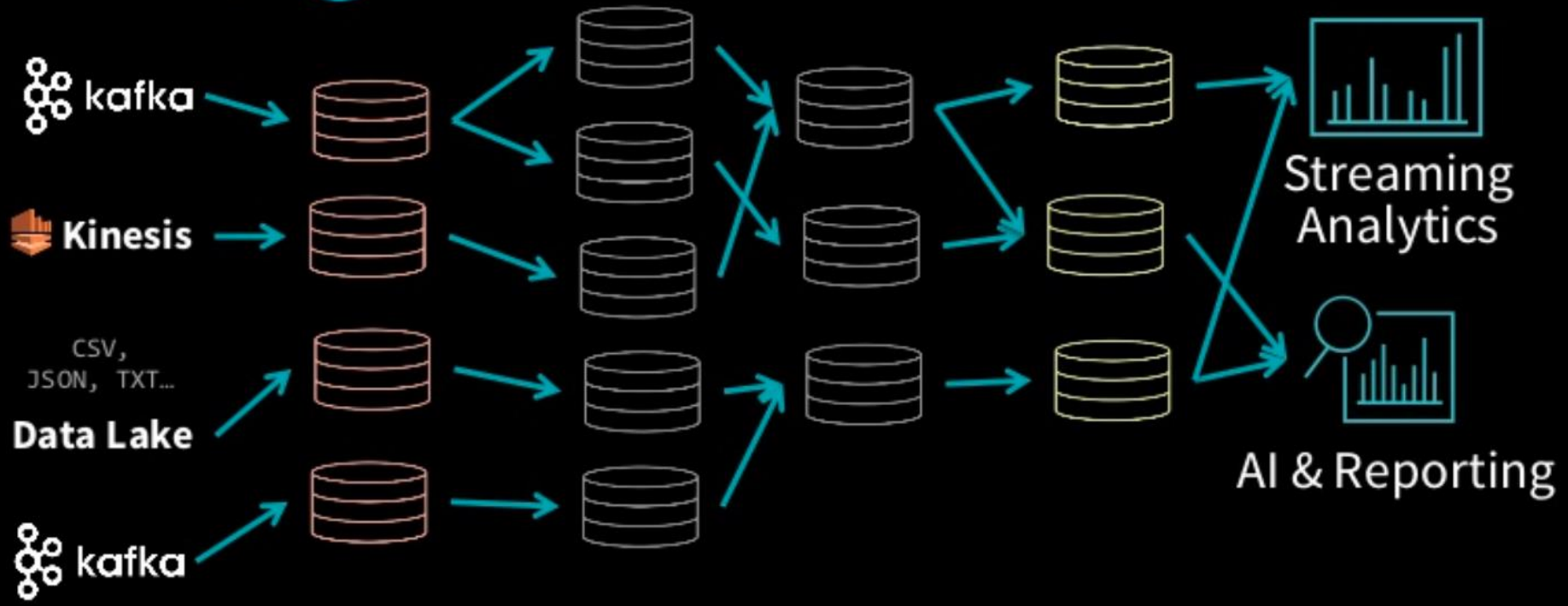
Data Lake Distractions

No atomicity means failed production jobs leave data in corrupt state requiring tedious recovery

No quality enforcement creates inconsistent and unusable data

No consistency / isolation makes it almost impossible to mix appends and reads, batch and streaming

The DELTA LAKE Architecture

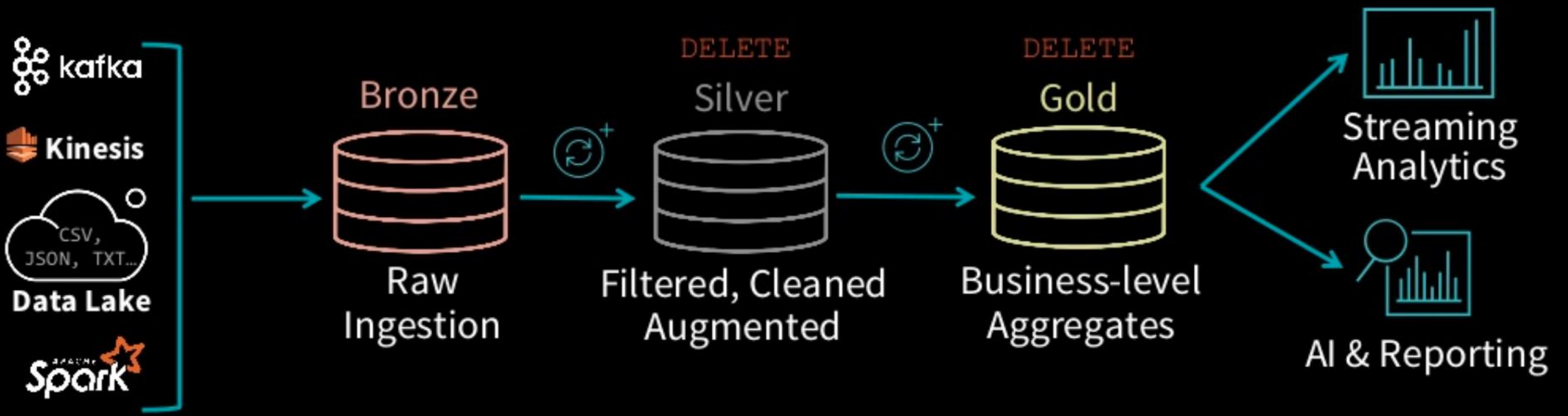


Full ACID Transaction

Focus on your data flow, instead of worrying about failures.

Clip slide

The DELTA LAKE



Easy to recompute when business logic changes:

- Clear tables
- Restart streams

Clip slide

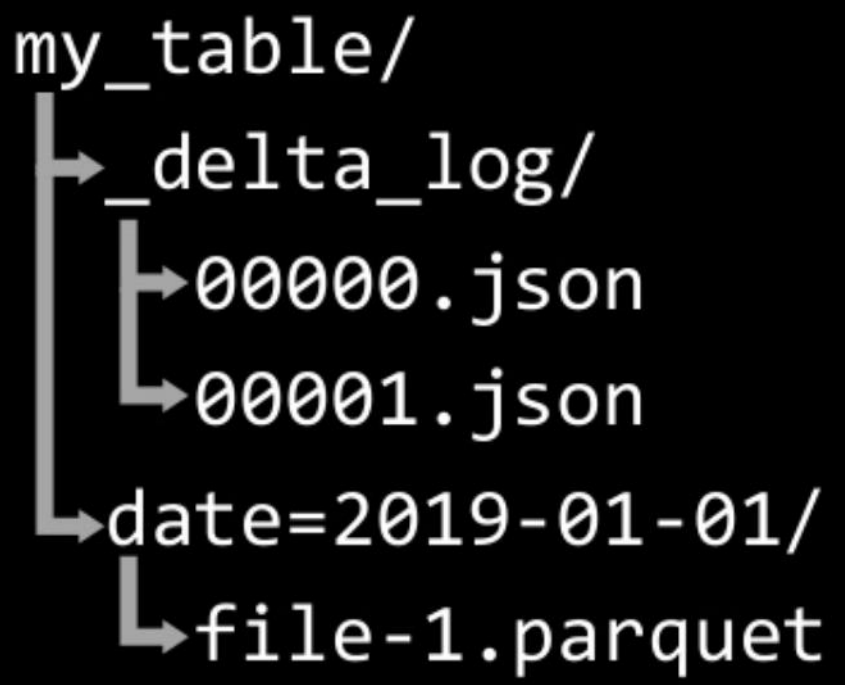
Delta On Disk

Transaction Log

Table Versions

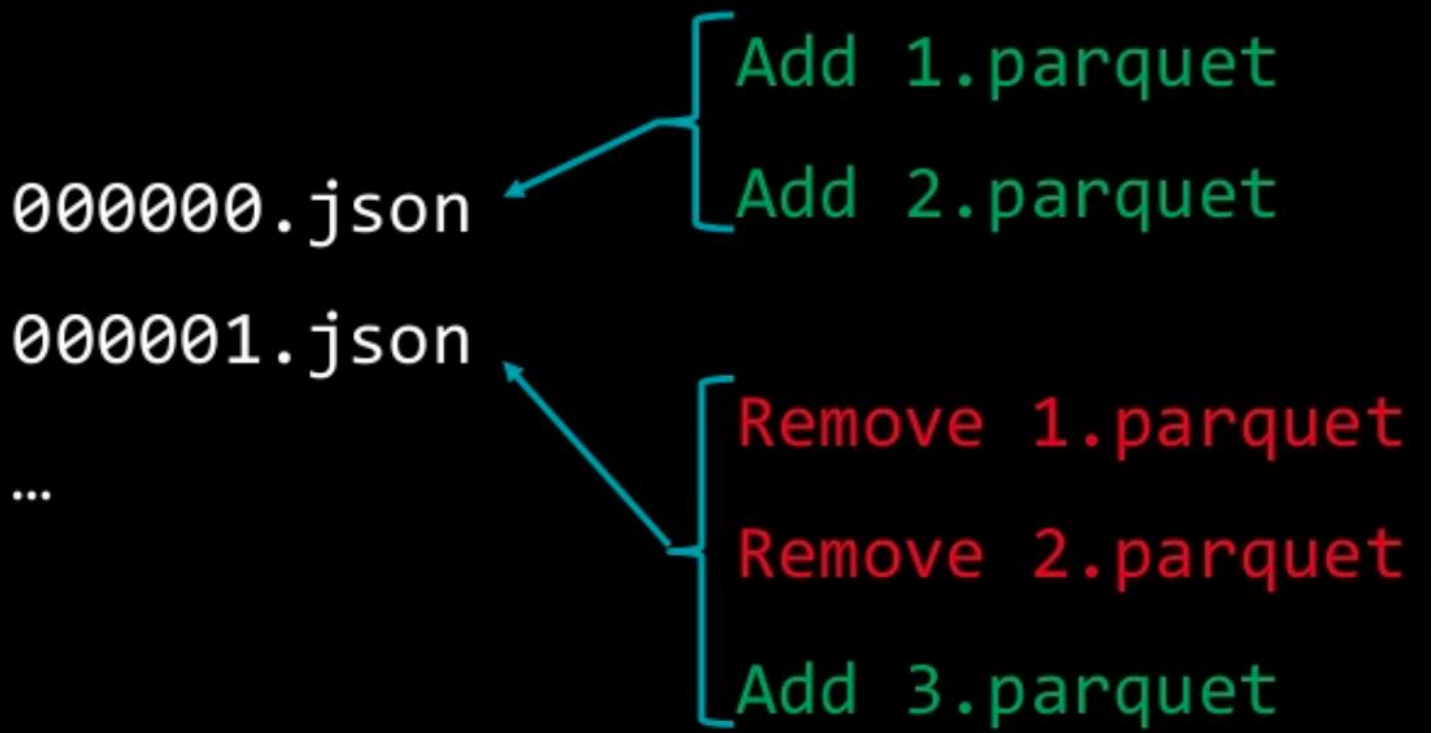
(Optional) Partition Directories

Data Files



Implementing Atomicity

Changes to the table are stored as *ordered, atomic* units called commits



External Blobs / Key vault integration

Python

```
dbutils.fs.mount(
  source = "wasbs://<container-name>@<storage-account-name>.blob.core.windows.net",
  mount_point = "/mnt/<mount-name>",
  extra_configs = {"<conf-key>":dbutils.secrets.get(scope = "<scope-name>", key = "<key-name>")})
```

https://docs.databricks.com/spark/latest/data-sources/azure/azure-storage.html