Accelerating CDC Data Ingestion with Apache Beam: A Qlik-to-BigQuery Journey

Bipin Upadhyaya
Cloud Data Engineer
Google Cloud Consulting

BEAM SUMMIT

September 4-5, 2024

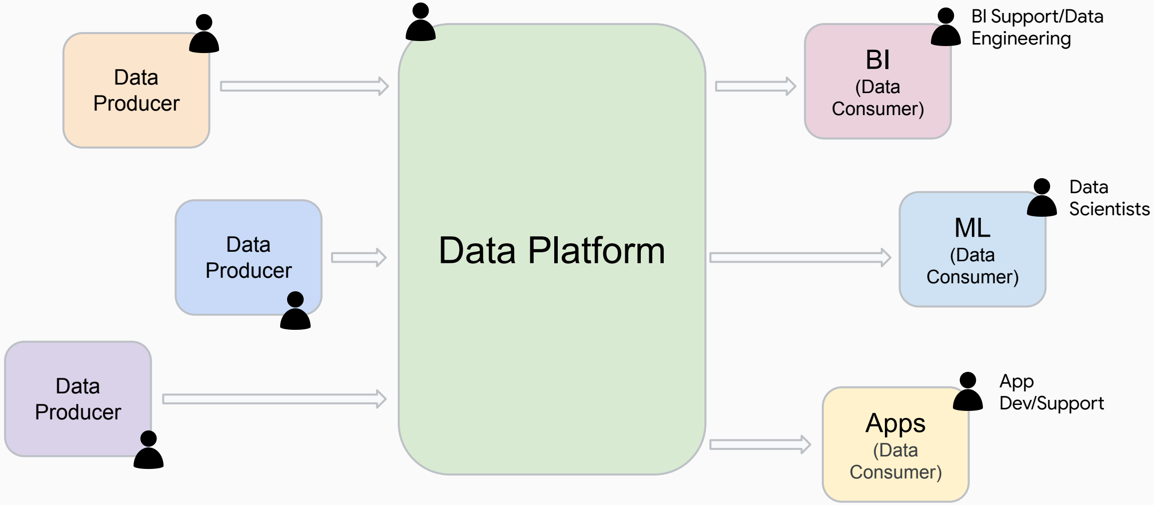
Sunnyvale, CA. USA

Agenda

- Overview
- Guiding Principles
- Overall Architecture
- Ingesting CDC to BigQuery
  - Qlik Data Format
  - Error Handling
- Observation and Optimization

Overview

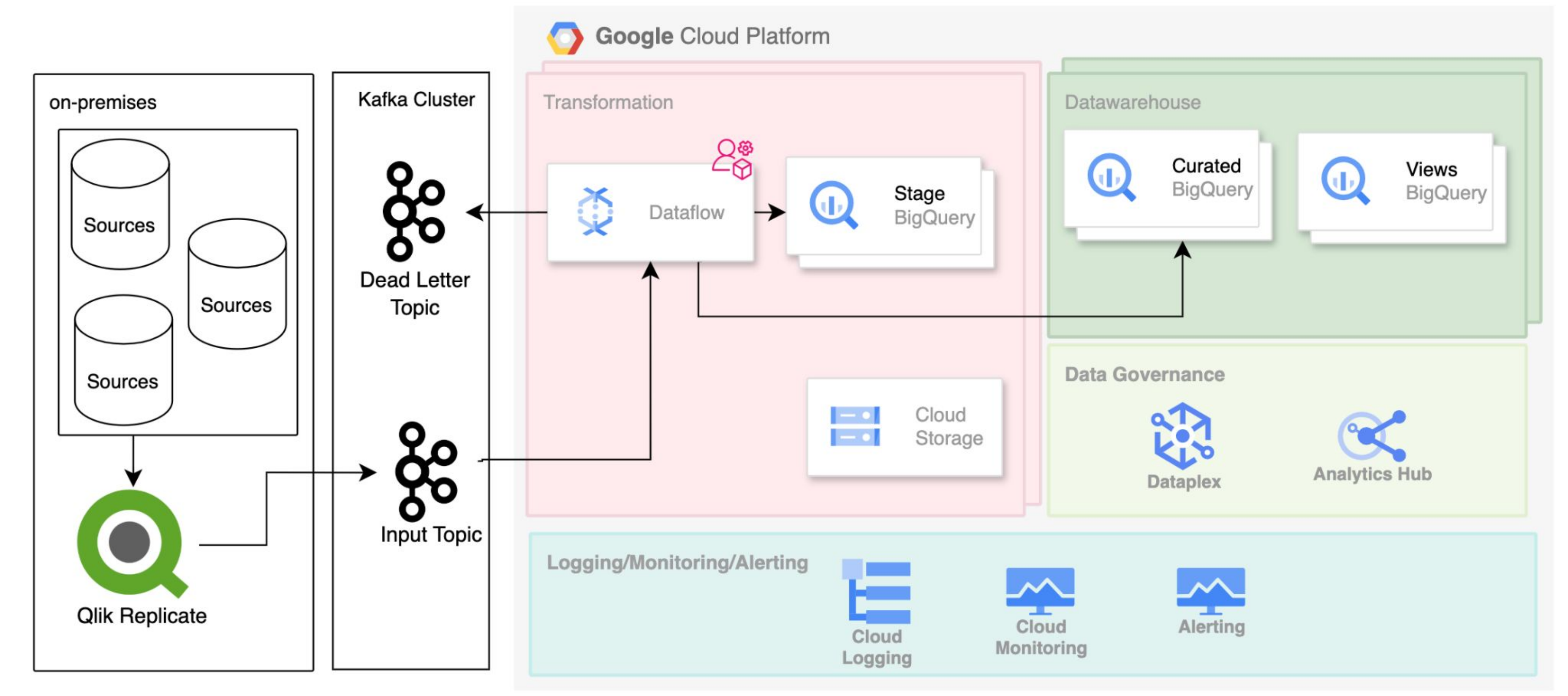
- **Ingest** data from different RDBMS sources (e.g., change data captures)
- **Vend out curated data** useful to **multiple** eligible consumers.
- Decoupled systems & clean **Operational Ownership**.



Guiding Principles

- Managed services
  - Scalable framework
  - Petabyte scale storage and Queries
- Code Reusability
  - Build Once. Run for all CDC ingestion
- Data security and privacy first
- Making data available for Business Analysts within seconds
- Infrastructure as Code
- Operational observability

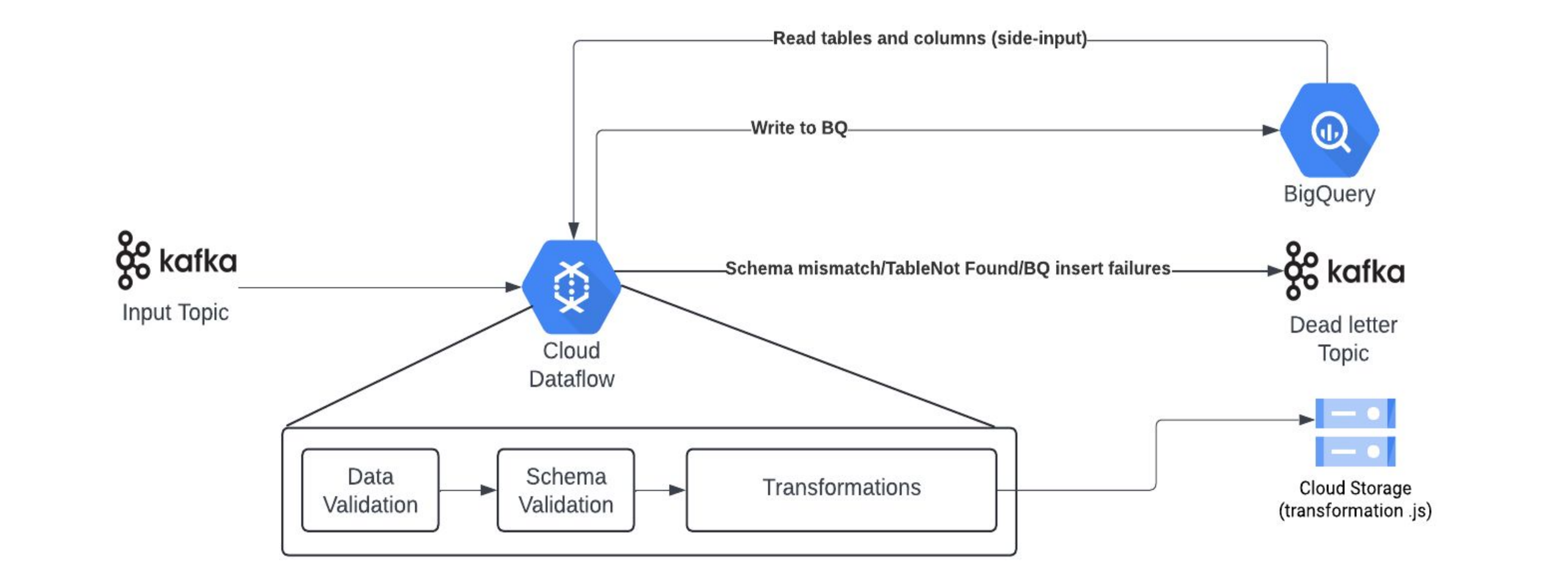
Overall Architecture


Google Cloud Platform
on-premises
Sources
Sources
Sources
Qlik Replicate
Kafka Cluster
Dead Letter Topic
Input Topic
Transformation
Dataflow
Stage
BigQuery
Cloud
Storage
Datawarehouse
Curated
BigQuery
Views
BigQuery
Data Governance
Dataplex
Analytics Hub
Logging/Monitoring/Alerting
Cloud
Logging
Cloud
Monitoring
Alerting

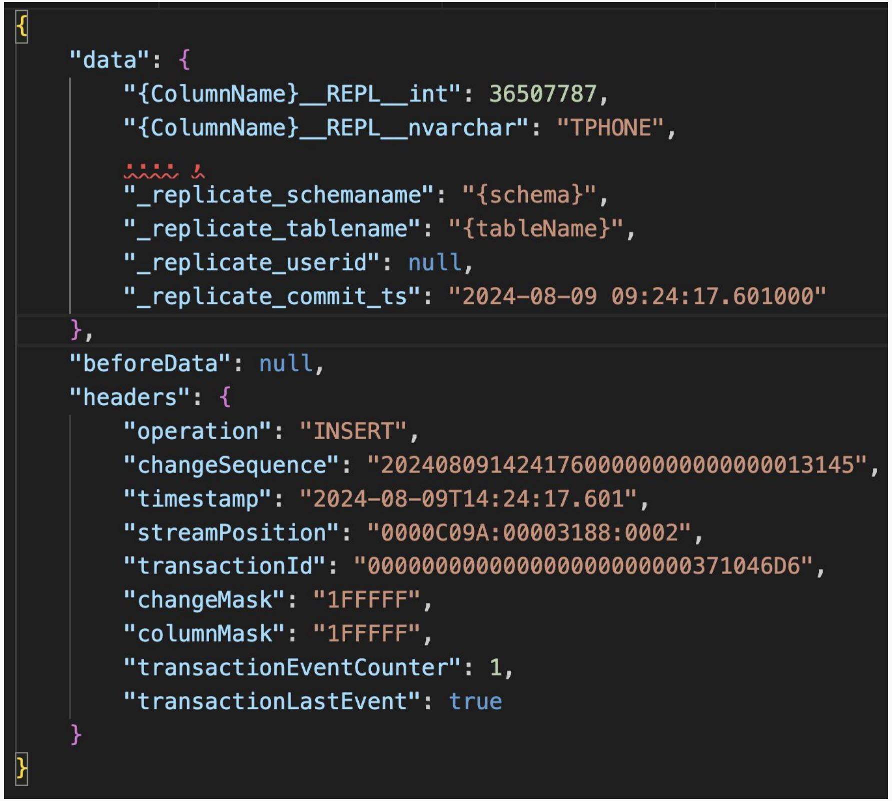
Collaborate & Prep

- Cross collaboration between **Data governance team, Data privacy team, Data owners** and **Data stewards.**
- Extract Schema from the sources
  - Create BigQuery dataset and table (use SQL translation service)
    - Append extra columns (such as *replicate*_timestamp, sequence#)
- Access controls
  - Define policy tags for the columns
  - Define row level access controls
- Define tag templates for data discoverability

Ingesting CDC Changes to BigQuery


Read tables and columns (side-input)
Write to BQ
BigQuery
kafka
Input Topic
Cloud Dataflow
Schema mismatch/TableNot Found/BQ insert failures
kafka
Dead letter Topic
Data Validation
Schema Validation
Transformations
Cloud Storage (transformation .js)

Qlik Data Format



```
{
    "data": {
        "{ColumnName}__REPL__int": 36507787,
        "{ColumnName}__REPL__nvarchar": "TPHONE",

        .... ,
        "_replicate_schemaname": "{schema}",
        "_replicate_tablename": "{tableName}",
        "_replicate_userid": null,
        "_replicate_commit_ts": "2024-08-09 09:24:17.601000"
    },
    "beforeData": null,
    "headers": {
        "operation": "INSERT",
        "changeSequence": "20240809142417600000000000000013145",
        "timestamp": "2024-08-09T14:24:17.601",
        "streamPosition": "0000C09A:00003188:0002",
        "transactionId": "000000000000000000000000371046D6",
        "changeMask": "1FFFFF",
        "columnMask": "1FFFFF",
        "transactionEventCounter": 1,
        "transactionLastEvent": true
    }
}
```

Iteration I

{columnName}__REPL__{format}

- Schema validation to check the Column Name exists in BigQuery table
- Dataflow pipeline casts the value to proper format

Iteration II

{columnName}

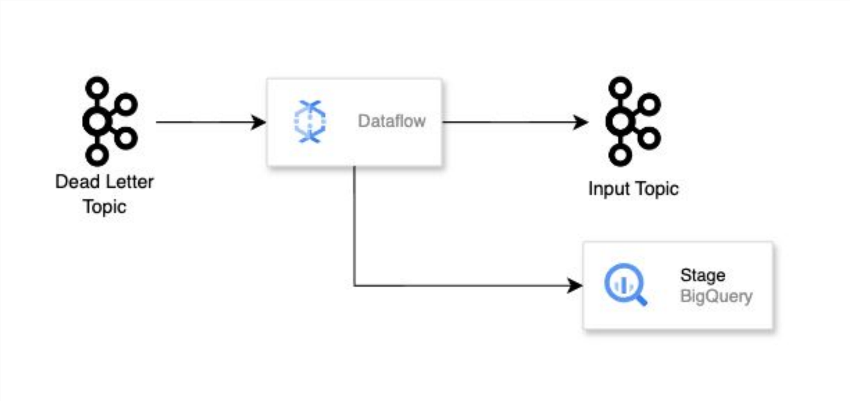
- Drop REPL part.
- Use the schema information from BQ tables

Error handling

Records with data-related errors are persist in BQ stage dataset (Manual inspection required)

- Bad data/Unserializable data
- Format conversion Errors

Other errors such as table not found and schema mismatch are resolved during replay.



Observations and Optimizations

- Enable Autosharding in BigQuery
- Enable cloud profiling (in dev/test) to identifying any bottlenecks in your functions
- Leverage DoFn lifecycle to speed up per element processing when external API is involved.
- Number of partitions in kafka topic

Observations and Optimizations

- Kafka consumer configurations
  - **unboundedReaderMaxReadTimeSec**: Use lower for low latency pipeline
  - **unboundedReaderMaxElements**: Use higher number if pipeline performs aggregation
- Restrict excessive logging in Dataflow pipelines
  - defaultWorkerLogLevel
  - --workerLogLevelOverrides={"<package/class>":"<level>","<package/class>":"<level>"}
- Similar deployment configurations
  - Run ingestion pipelines using the same worker type configuration
  - Capping the maximum workers number to avoid very large fleets
- [Future] Use BigQuery CDC

Thank you!

Questions?

Bipin Upadhyaya
Data Engineer
Google Cloud Consulting