

Spark vs. BigQuery vs. Trino

Shopify's journey of SQL transformation at scale
Presented by Kate Darnell



What makes a strong data platform?

- Process petabytes of data at scale
- Support 500+ data scientists
- Constantly deliver fresh data
- Track data lineage

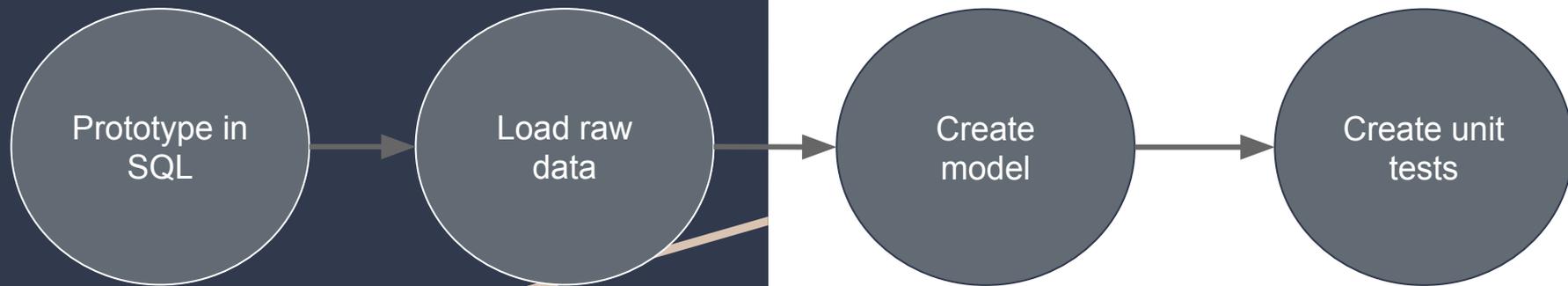
Shopify's Batch Transformation Strategy

Legacy approach: pyspark on hadoop

Modern approach: Using DBT to orchestrate SQL transformation pipelines in Trino and BQ

Looking Forward: SparkSQL orchestrated by DBT?

Model Development Cycle



PySpark

```
from pyspark.sql import functions as F

return select_named_columns(
    table1
    .join(table2, on='store_id', how='inner')
    .filter(F.col('content').isNotNull()),
    {
        'Customer_id': F.col('Customer_id'),
        'store_id': F.col('store_id'),
        'content': F.col('content')
    })
)
```

DBT jinja SQL Template

```
{{ seamster_config('view', engines=["trino", "BigQuery"],
trino_session_properties=[]) }}

WITH tmp_table AS (

    SELECT customer_id, store_id, content

    FROM {{ ref(table1) }} JOIN {{ ref(table2) }}

    WHERE content is not null

)

SELECT *

FROM tmp_table
```

PySpark

```
from pyspark.sql.types import BooleanType, DecimalType,
DoubleType, LongType, StringType, TimestampType

partner_marketing_attributes': Contract({
    'store_id':{'type': StringType(),
                'nullable': True
    }
})
```

DBT schema.yml Template

```
columns:
- name: store_id
  description: store id
  data_type: STRING
  tests:
  - unique
```

Type contracts

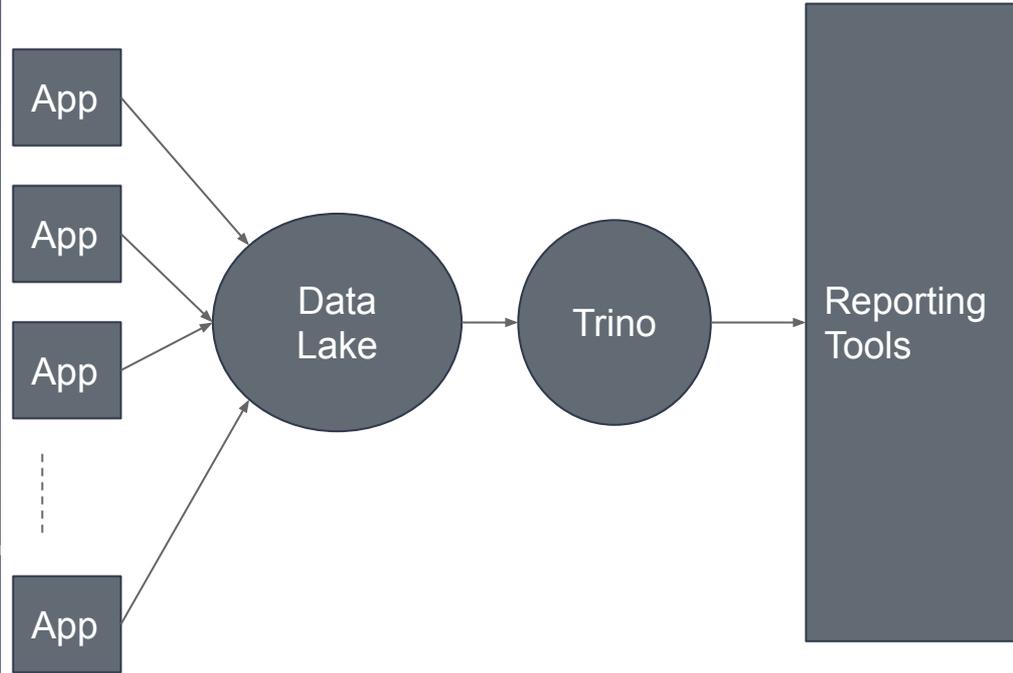


Data Ingestion Trino

Accumulo
Atop
BigQuery
Black Hole
Cassandra
ClickHouse
Delta Lake
Druid
Elasticsearch
Google Sheets
Hive
Iceberg
JMX
Kafka
Kinesis
Kudu
Local File

MariaDB
Memory
MongoDB
MySQL
Oracle
Phoenix
Pinot
PostgreSQL
Prometheus
Redis
Redshift
SingleStore (MemSQL)
SQL Server
System
Thrift
TPCDS

30+ Connectors available to
popular 3rd party data
applications



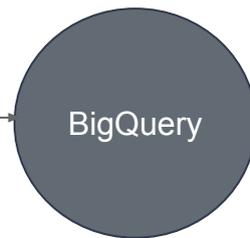
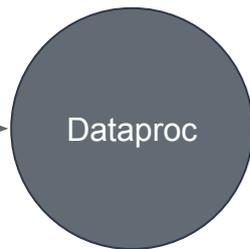
Data Ingestion BigQuery

Google dataproc

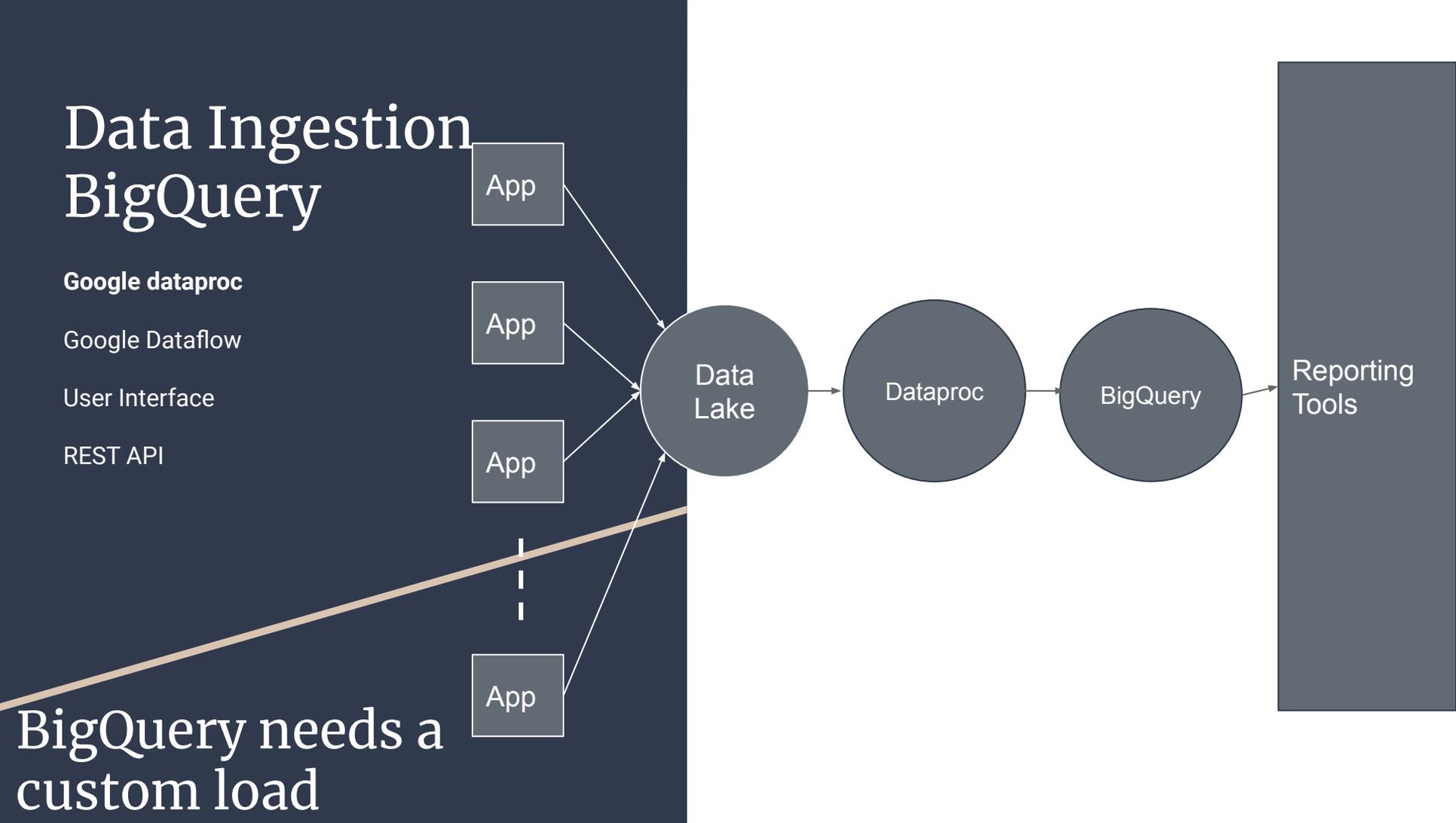
Google Dataflow

User Interface

REST API



BigQuery needs a
custom load



Performance

BigQuery -Fast

BigQuery's **internal storage format**,
conductor

Resources allocated on BigQuery's pay **per query** plan.

Trino - depends on data storage format and cluster Tuning

Trino connects directly to data stores and performs queries on **raw files**

Improved performance with **Iceberg**.

Scaling the cluster can also impact performance

Storage

BigQuery

We use BigQuery's **internal tables** which store **duplicated data** from data lake.

4000 **partition limit** means storing data on an hourly grain will limit to 6 months of data

Trino

We use external tables in GCS.

User Frustrations

Ease of Use

- Supporting parallel systems leads to user confusion
- Data needs a custom load job to get into BQ
- Trino requires training an in-house development and reliability teams

Features

- We're hitting lots of BigQuery limits for large models
- Some Trino features have limited support

Debugging

- BigQuery support tickets can take ~2 days for a solution
- BigQuery's logging system is limited

Summary

	Spark	BigQuery	Trino
Developer Experience	★ ★ Pyspark syntax requires SMEs to have data engineering experience	★ ★ Dataproc job to load data into BQ delays development	★ ★ ★
Cost	Team: 🧑🏠 🧑🏠 Infrastructure: 💰	Team: Infrastructure: 💰 💰 💰	Team: 🧑🏠 🧑🏠 Infrastructure: 💰
Reliability	★ ★ ★ Mature product maintained in-house and supported by Google Dataproc	★ ★ ★ Fully managed by Google	★ ★ Has been down a few times as in house team is learning to maintain and scale.
Features and open source	★ ★ ★ Has been supported by the open source community since 2014	★ ★ Closed source, and limited integrations with other data products.	★ ★ ★ Has been supported by the open source community since 2019. Shopify has made contributions.
Useage	~800,000 jobs/day	890 active users/week ~450,000 queries/week	3,000 active users/week ~200,000 queries/week

Thank you

