trino iceberg connector

Powering Fast Changing Data Marts
at blinkit & zomato’s scale

trino blinkit zomato

Speakers

Bhanu Mittal

Data Platform Engineer

Blinkit - India

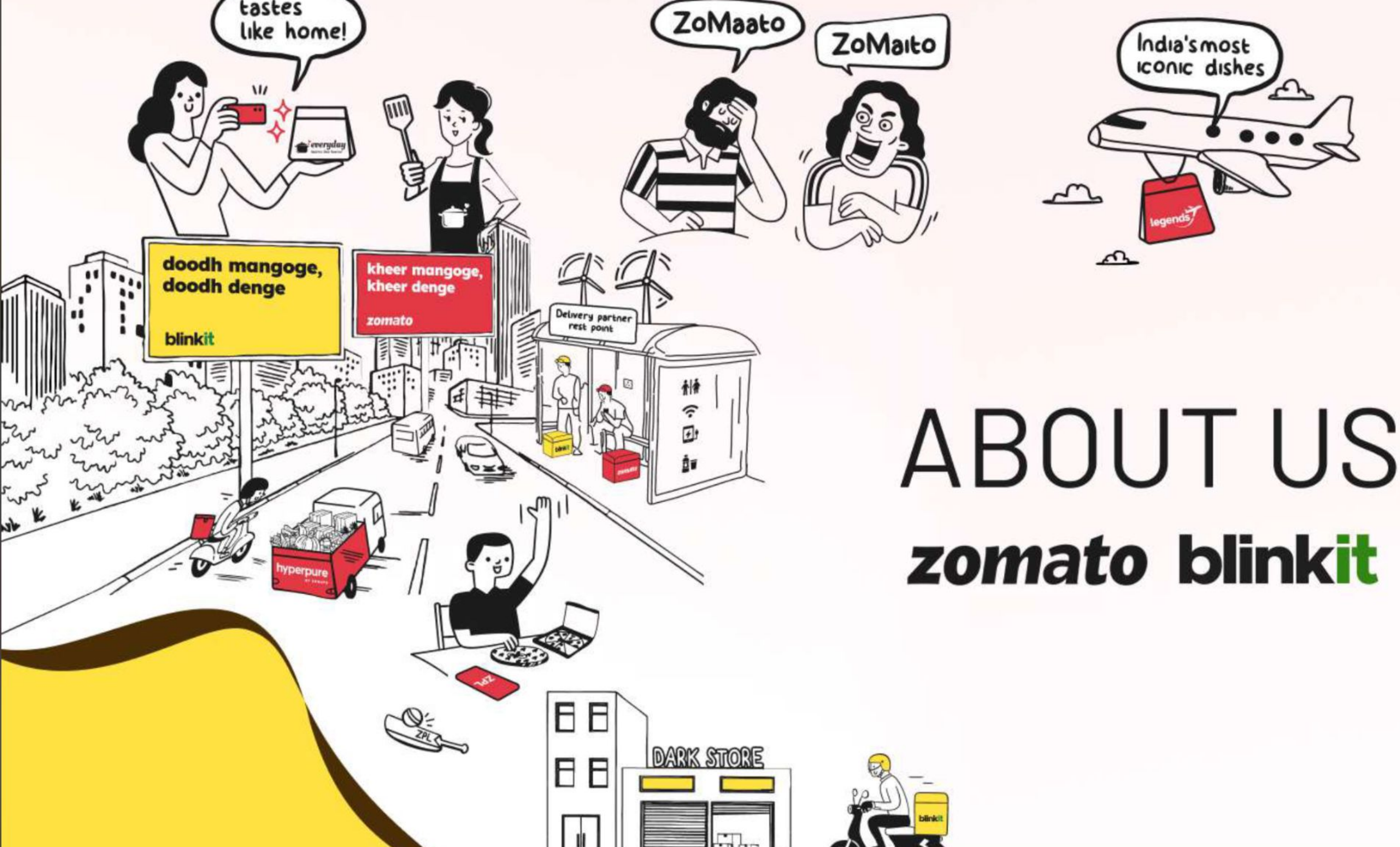
Shubham Gupta

Data Platform Engineer

Blinkit - India

Agenda

- Blinkit & Zomato
- Why data marts?
- Why Trino?
- Implementation
- Impact



ABOUT US

zomato blinkit

Zomato at Glance

Better food for more people

~18M
customers

~230K
restaurant partners

~350K
delivery partners

~50M
orders

* These are average monthly statistics from Q1 FY24
* All content provided here is for informational and educational purposes

Blinkit at Glance

re-imagining quick commerce to make it local, curated, efficient and possible in the blink of an eye

~4M

monthly transacting customers

~12M

monthly orders

19

cities

400

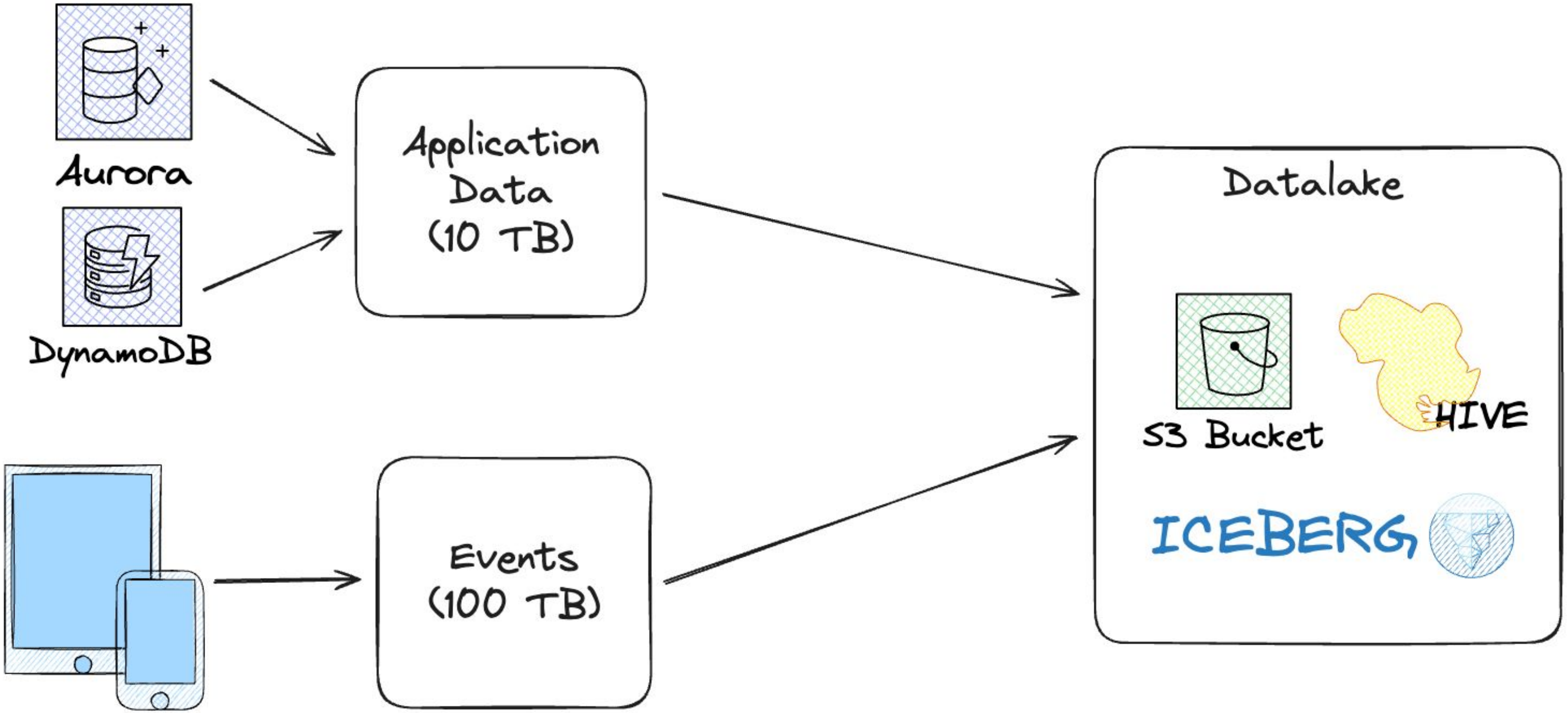
stores

* These are average statistics of Q1 FY24

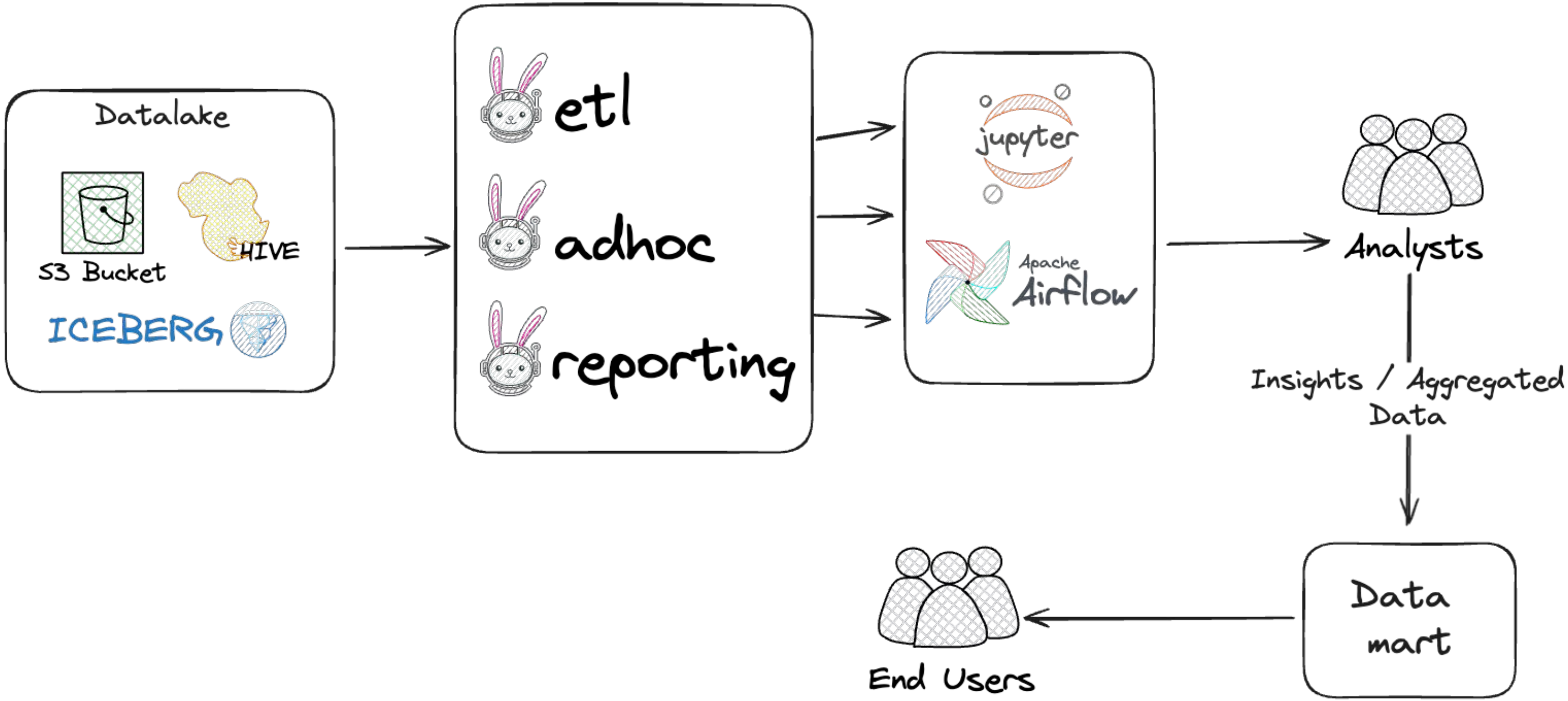
blinkit

Why do we need data marts?

Fast Scalable Analytics at Low Cost



Fast Scalable Analytics at Low Cost

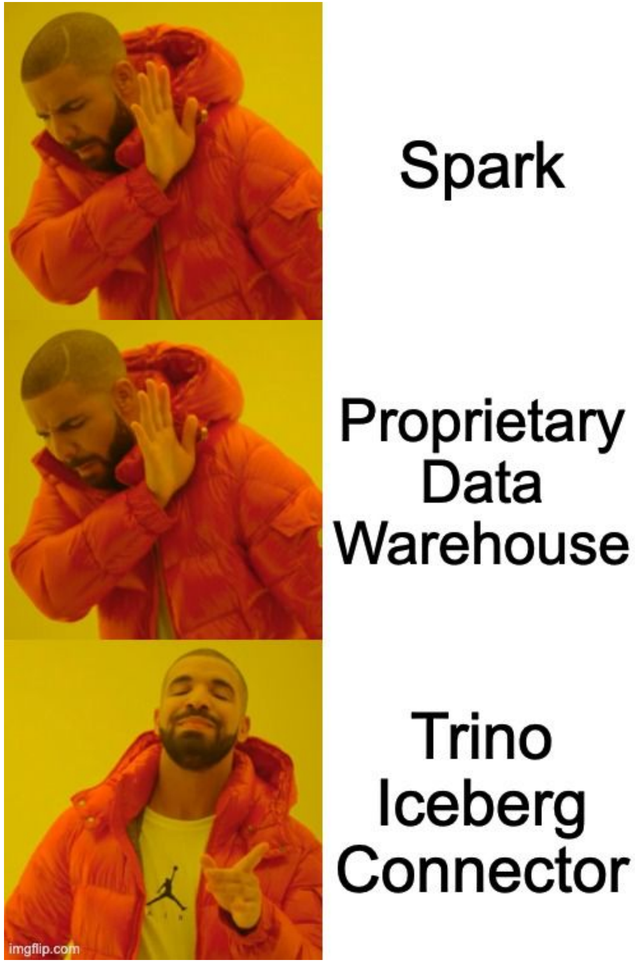


blinkit

Problem statements

- 1000+ data marts
- Refreshed every 15-60 minutes
- Ease of use

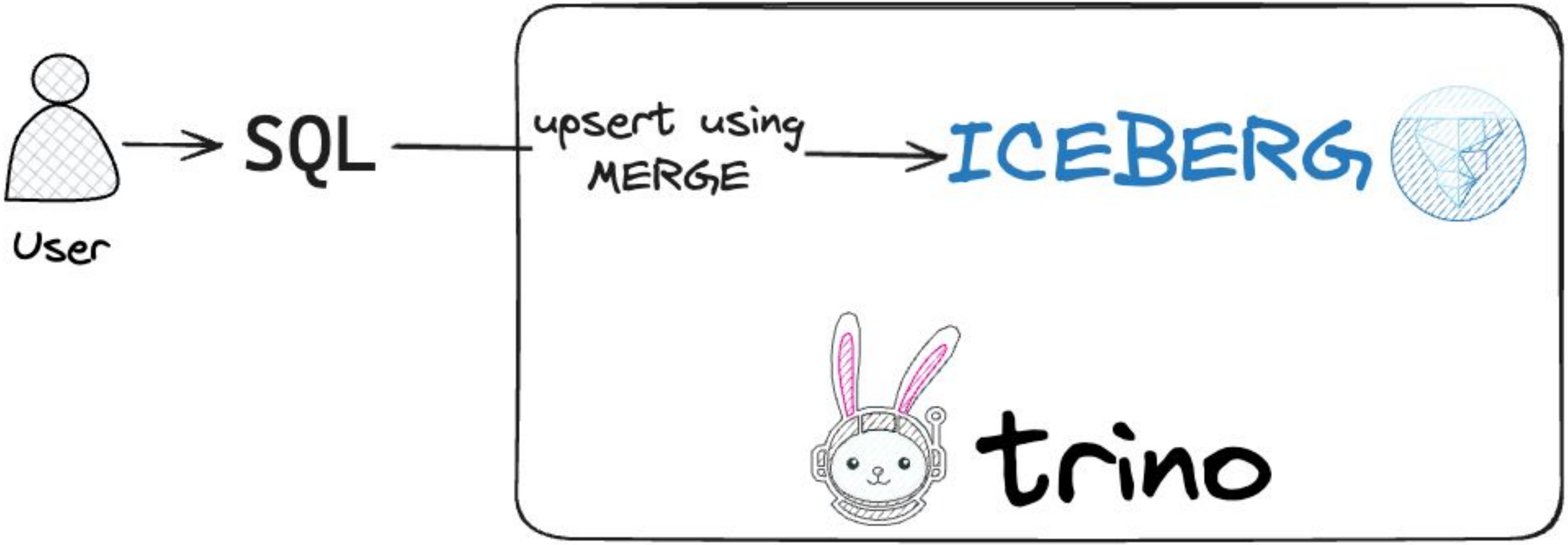
Solution?

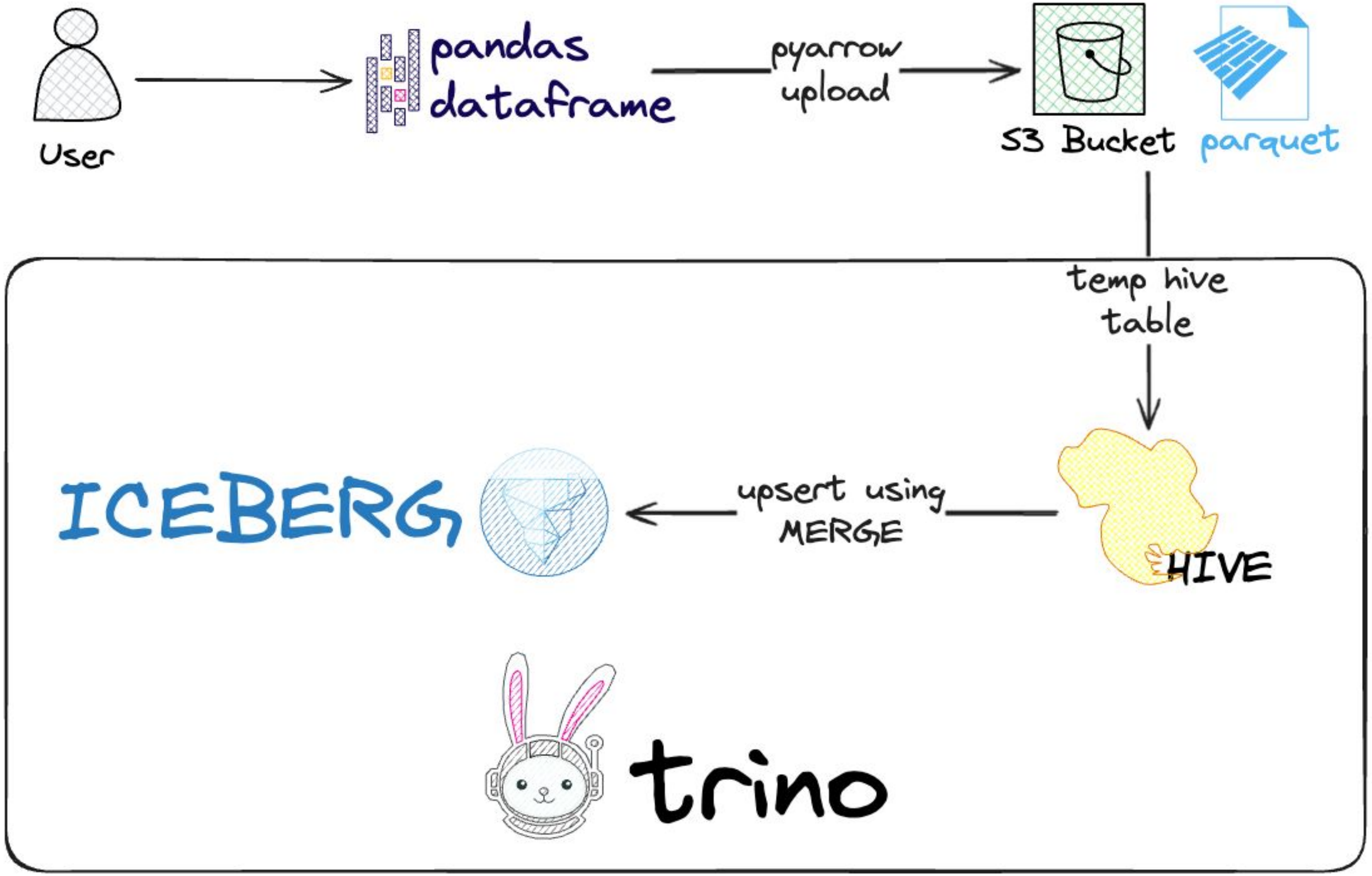

Spark
Proprietary Data Warehouse
Trino Iceberg Connector
imgflip.com

	Managed Warehouses	Spark (Managed)	Spark (Open source)	Trino Iceberg
Ease of Use	🔥	❌	❌	🔥
Performance	✅	🔥	🔥	🔥
Cost	❌	❌	✅	🔥
Scalability	❌	✅	🔥	✅

Implementation

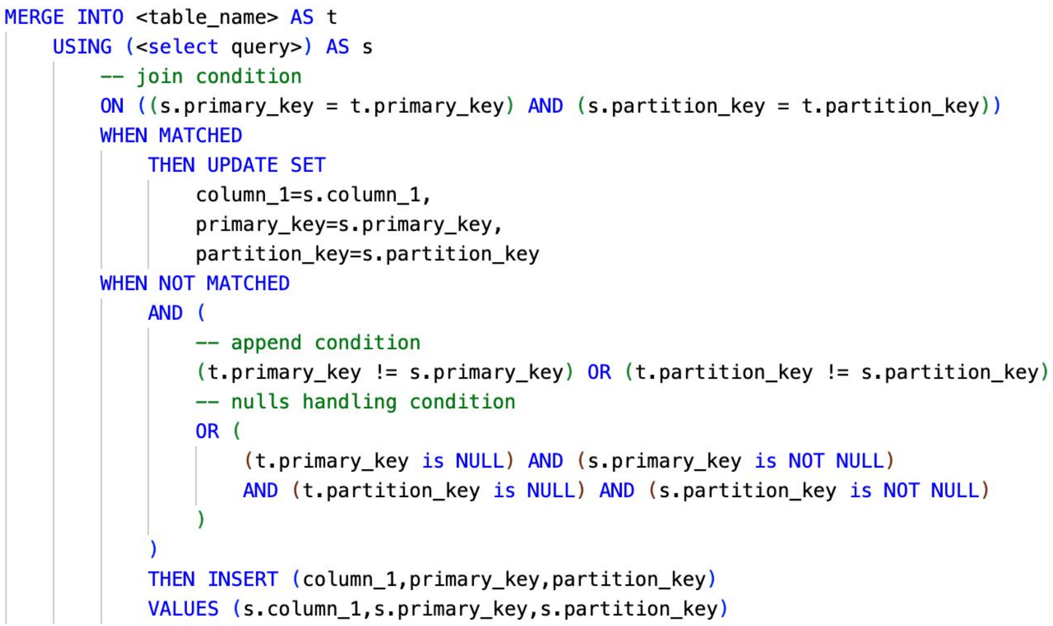
- SQL Flow
- Dataframe Flow


User
SQL
upsert using MERGE
ICEBERG
trino


User
pandas dataframe
pyarrow upload
S3 Bucket
parquet
temp hive table
HIVE
upsert using MERGE
ICEBERG
trino

Capabilities

- Load types
    - Upsert
    - Append
    - Truncate
    - Partition Overwrite
- Schema evolution
- Incremental key support



```sql
MERGE INTO <table_name> AS t
    USING (<select query>) AS s
        -- join condition
        ON ((s.primary_key = t.primary_key) AND (s.partition_key = t.partition_key))
        WHEN MATCHED
            THEN UPDATE SET
                column_1=s.column_1,
                primary_key=s.primary_key,
                partition_key=s.partition_key
        WHEN NOT MATCHED
            AND (
                -- append condition
                (t.primary_key != s.primary_key) OR (t.partition_key != s.partition_key)
                -- nulls handling condition
                OR (
                    (t.primary_key is NULL) AND (s.primary_key is NOT NULL)
                    AND (t.partition_key is NULL) AND (s.partition_key is NOT NULL)
                )
            )
            THEN INSERT (column_1,primary_key,partition_key)
            VALUES (s.column_1,s.primary_key,s.partition_key)
```

A Glimpse into the action 🚀

Active tables

1000+

Largest upsert table

~5TB

(10M daily upserts)

Largest append table

~50TB

(2B daily inserts)

Data written

~6TB/day

~44B rows/day

p90 upsert time

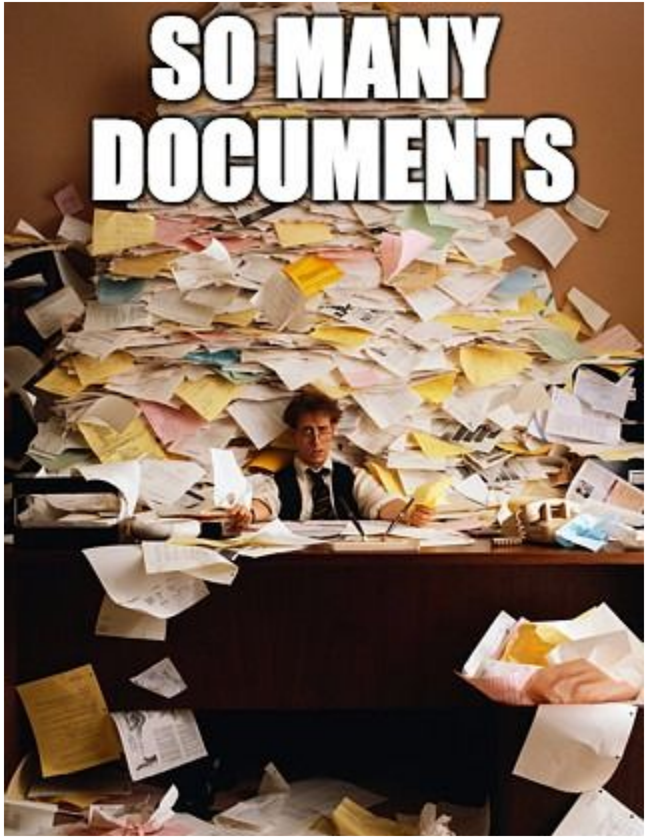
~5 minutes

~7 minutes

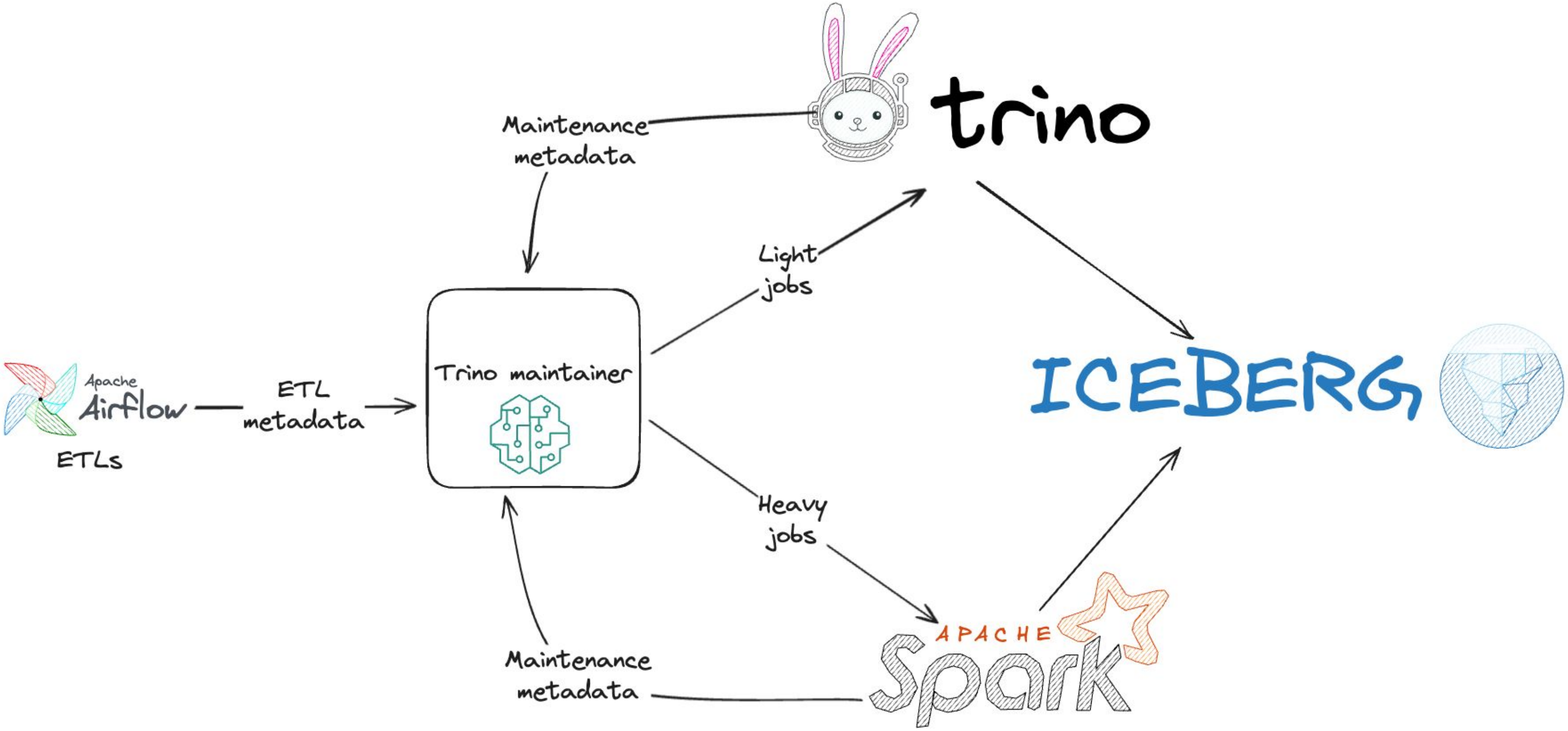
(with maintenance)

*Metrics are for a **30 nodes (8vCPU, 128GB)** general purpose read/write cluster

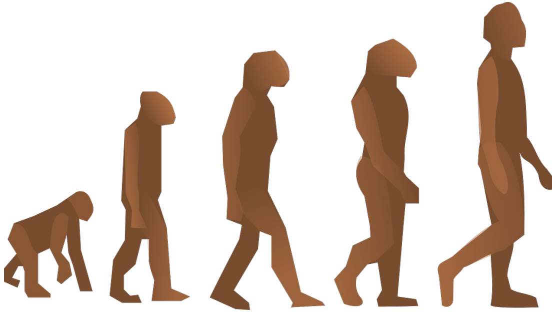
Need for Maintenance



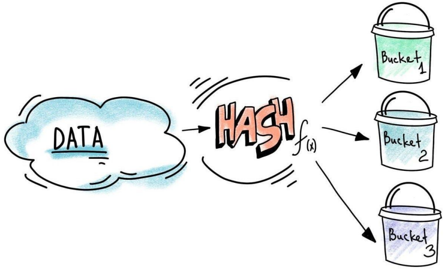
Iceberg Maintenance



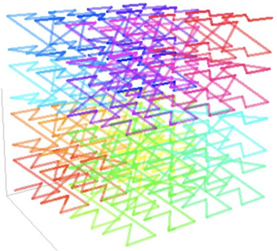
Future Scope?



Enhanced schema evolution



Bucket based partitioning



Z order and puffin index

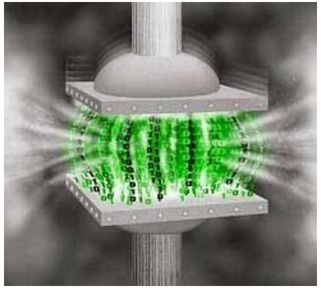


Table compression

Thank You

If you want to work on cool projects with us,
reach out to us at **yes@zomato.com**

Shubham Gupta **shubham.gupta2@blinkit.com**
Bhanu Mittal **bhanu.mittal@blinkit.com**