GO-JEK's Journey: Moving to BigQuery, Our Next Gen Data Warehouse

Dimas Natajiwa
Johanes Alexander

Business Intelligence @ GO-JEK

Google Cloud

Agenda

Introduction
Journey
The Takeaway

Google Cloud

Confidential & Proprietary

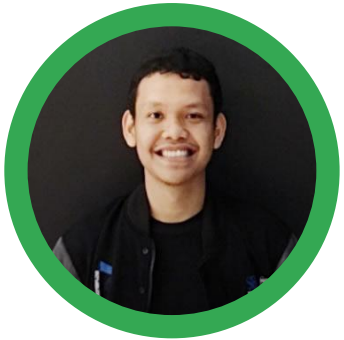
Introduction

Google Cloud

Hi Everyone!

Dimas Natajiwa

Data Warehouse (DWH) Manager
Business Intelligence @ GO-JEK



Johanes Alexander

Lead Solutions Architect
Business Intelligence @ GO-JEK

Google Cloud

GO-JEK

How GO-JEK started

Mobile App for Daily Needs

The GO-JEK app offers various services such as transport, food delivery, courier, instant shopping, professional massage, to payments

2010 Call-center for ojek* services

2015 App launched with 3 services

2016 Expansion and new services

*ojek is an Indonesian term of motorcycle ride hailing

Google Cloud

GO-JEK

Born to Provide Solutions

Main challenges for the informal sector in Indonesia:

- Inefficiency and competition
- Limited access to customer
- Limited access to financial services (unbankable)

Google Cloud

Our Solution for Every Customer's Needs

- GO-RIDE
- GO-CAR
- GO-BLUEBIRD
- GO-FOOD
- GO-MART
- GO-BUSWAY
- GO-PULSA
- GO-MART
- GO-SHOP
- GO-SEND
- GO-MASSAGE
- GO-TIX
- GO-GLAM
- GO-BOX
- GO-AUTO
- GO-MED
- GO PAY
- GO POINTS
- GO BILLS

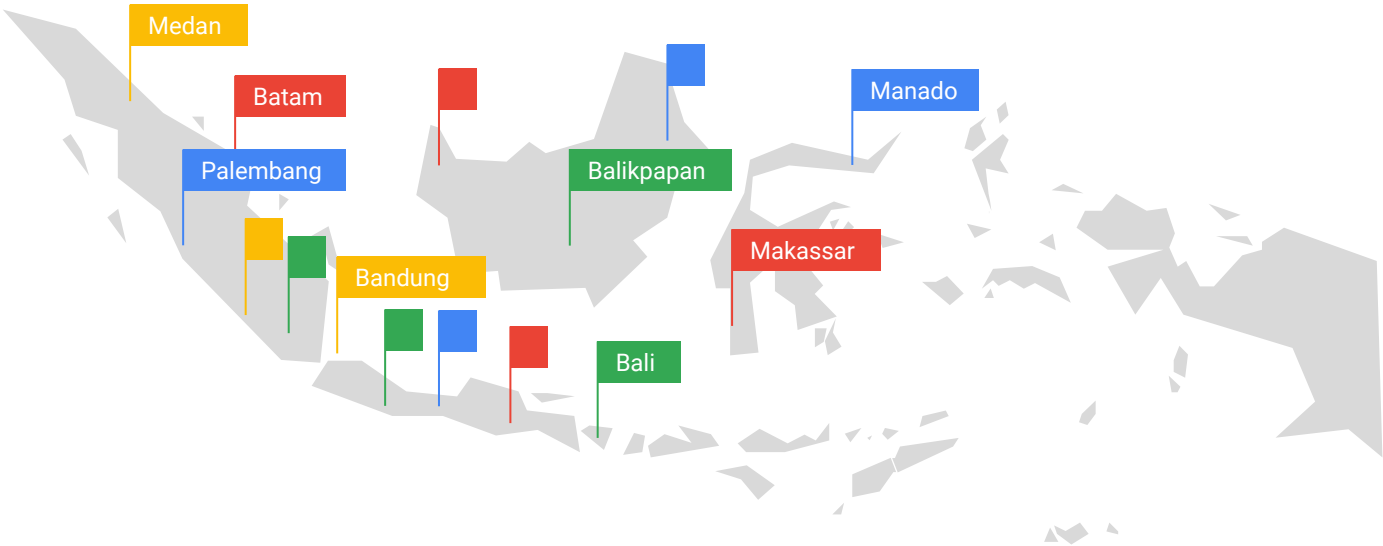
*Visit https://www.go-jek.com/ for more information

Google Cloud

GO JEK

GO-JEK's Footprint Nationwide

Operating in 50 cities throughout Indonesia



70m app downloads

+125k merchants

50 cities

+900k drivers

2m families

Google Cloud

GO-JEK

GO JEK

Journey

Google Cloud

The good old days

We started building the data warehouse from scratch in Q3 of 2016

It started based on the needs of the Business Intelligence team

Labour-intensive work and tons of time to deliver business insights

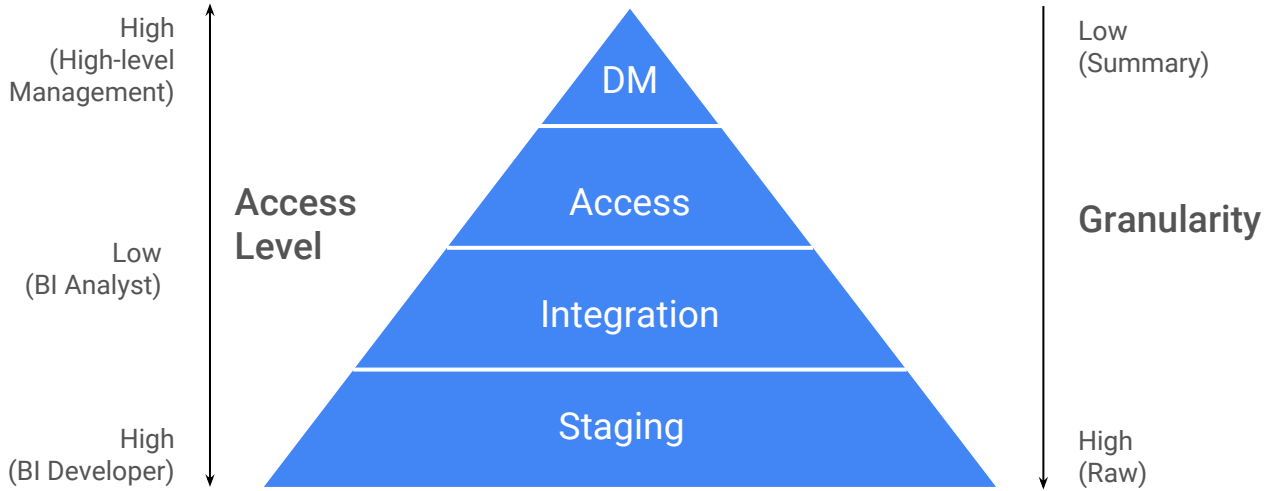
There was no single place to hold data

Single Version of Truth (SVOT) of business datasets was needed

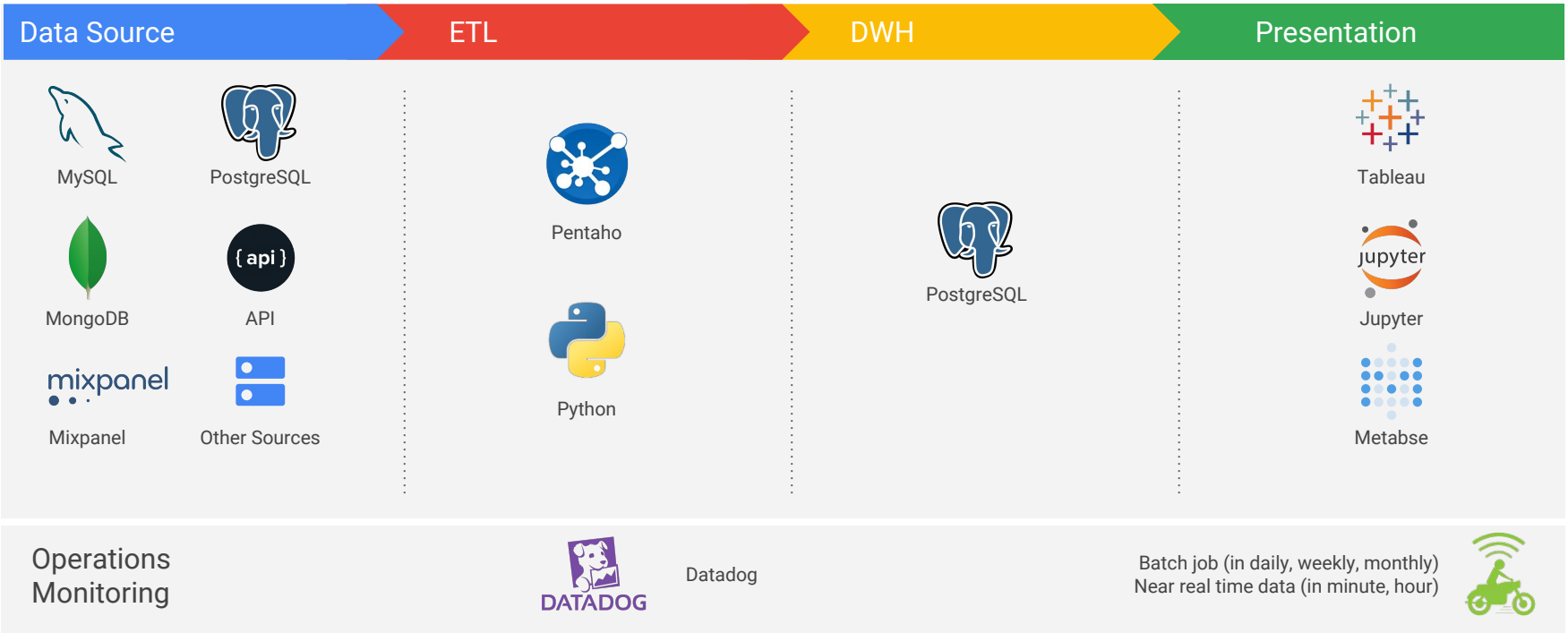
Various backends across multiple products

Needed to deliver hassle-free, timely and uniform data products

Design Rationale



DW Architecture (end of 2016)



The Stats in Q1 2017

~27%

Growing Data Volume per Month

*This is only business metrics data collected by BI

> 4000

Metabase Cards & Tableau Sheets

> 100

Low Resolution Dataset

*Contains reusable summary and roll-up datasets.

> 400

Average Daily Metabase & Tableau Users

*Everyone just loves data!

Google Cloud

GO JEK

The ~~good~~ old days

The pain starts coming in...

Performance

Everyone wants access to data to perform analysis on their own

Scalability

The growth of data is *extraordinary*

Operational

“With great power, comes great responsibility”

Uncle Ben

Then, what’s next?

What we want

- High performance, scalable, minimum operational maintenance
- Full resolution datasets
- Easy data discovery process

And then we realized

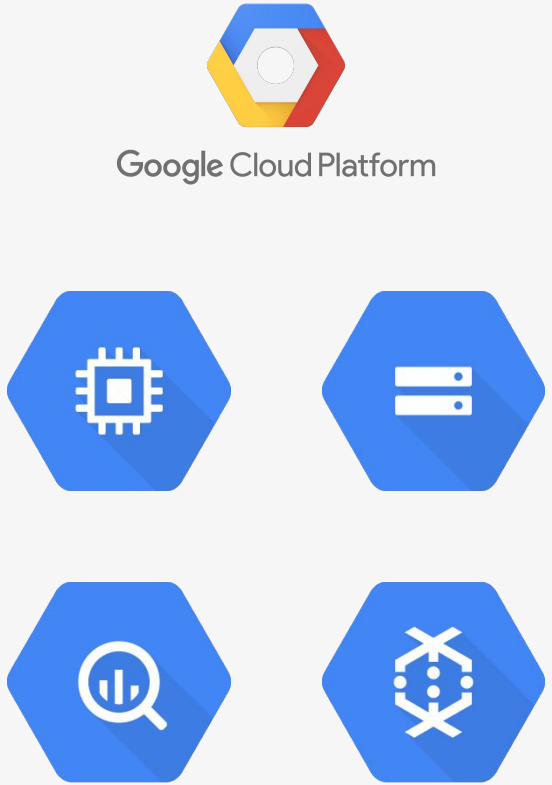
GO JEK

GO SEND GO CAR

GO TIX GO RIDE

GO FOOD

Google Cloud

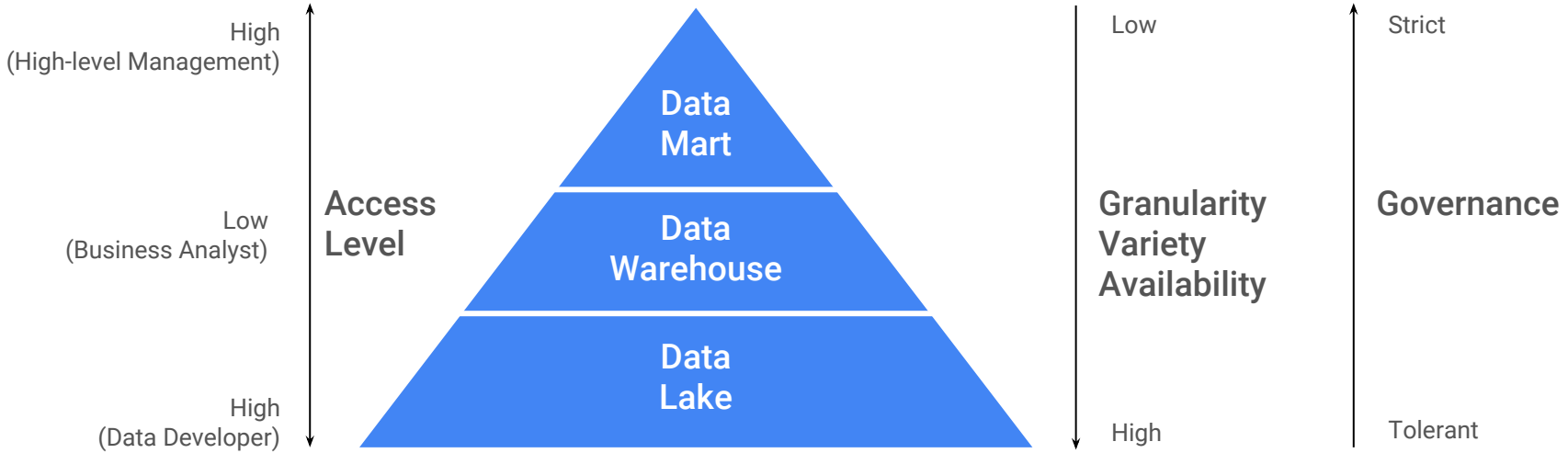


GO JEK

We called it:
“Next Gen DWH”

Google Cloud

Design Rationale



Next Gen DWH

~~What we want~~ What we achieved

- High performance, scalable, minimum operational maintenance
- Full resolution datasets
- Easy data discovery process

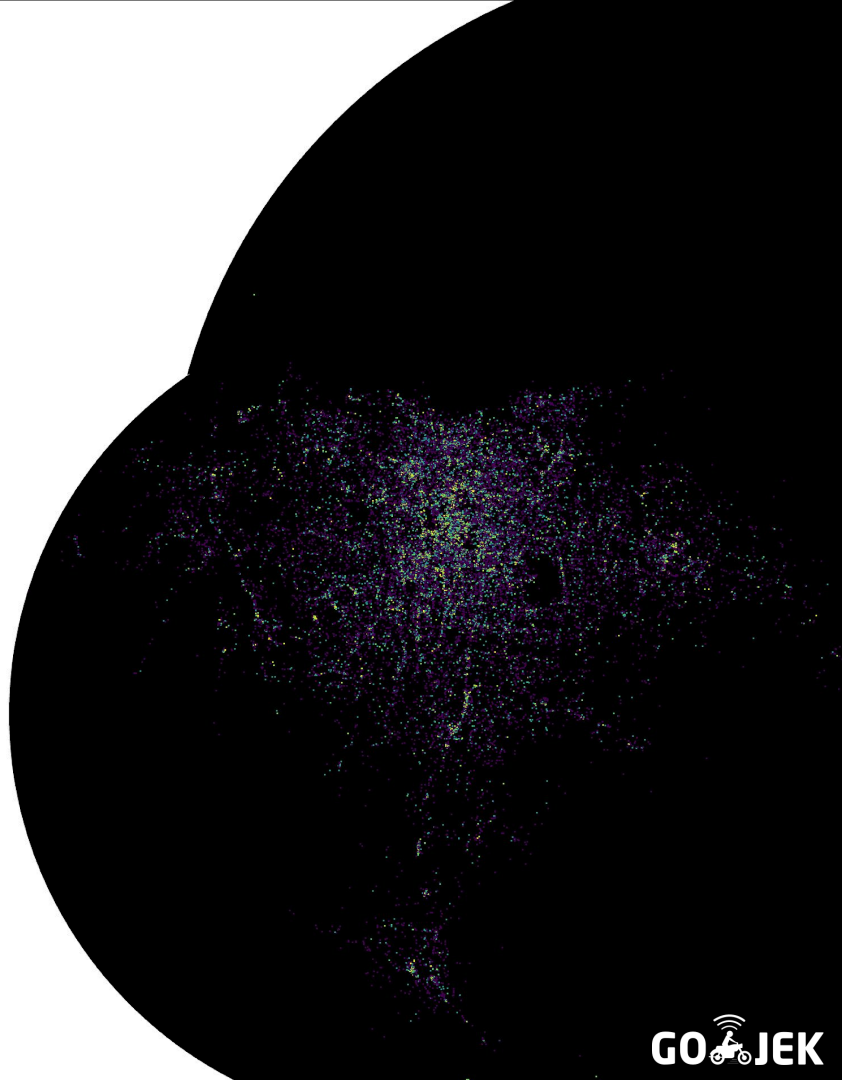
What did we build?

- *Wrappers* for universal job executors
- *Airflow* for job dependencies and orchestration
- *Denormalized tables* for prefered data modeling
- *BigQuery* for leveraging a cloud data warehouse
- *Datadex* for a universal data dictionary

Let’s go

On this iteration, we’re focusing on several aspects:

- Ingestion and transformation
- Storage
- Data Modeling
- Data Governance
- Operations



Google Cloud

GO-JEK

Ingestion and transformation

Batch

Pros

- Simple data processing (with the notion of *bounded data*)
- Data completeness is guaranteed
- Good enough to cover most of the analytics processes needed by the business

Cons

- Data availability is slow; most data is available in D+1
- Huge processing power is needed to process the growing data
- Huge operational efforts (e.g. migrating data from hundreds of DBs to the DWH everyday)

NEW! Streaming

Pros

- All the cons in *Batch* are handled

Cons

- Need to rethink the approach of architecture and business logic implementation
- Learning curve is a little steep for most to use and build (considering the streaming concept itself)

Ingestion and transformation - cont’d

Batch

What did we improve?

- ELT process in BigQuery *for simplicity*
- ETL process in Dataproc with Spark for *complex transformations*
- Improving data availability from daily to hourly

NEW! Streaming

What we’ve built so far?

- Work together with product teams to make their data available in Kafka/PubSub
- General ingestor from Kafka/PubSub to BigQuery
- Focusing on data quality and integrating business logic






Google Cloud

GO JEK

Storage

Old DWH

Pros

- Handles transactional data as is
- Indexing capabilities
- Granular access levels

Cons *(related to our usage)*

- Huge operational efforts
- Limited performance
- Limited scalability

NEW! Data Lake

Pros

- No-ops
- Handles all data types
- Stores data as it is

Cons *(related to our usage)*

- *Unknown in our implementation*

NEW! Next Gen DWH

Pros

- No-ops
- High performance
- Perfectly designed for OLAP

Cons *(related to our usage)*

- Loss of indexing capabilities
- No ACL on a table level
- Not the best solution to store transactional data

Google Cloud

GO JEK

Data Modeling

Star Scheme

Customer Dimension		
id	*name*	*phone_no*
123	Dika	628112345678

Order Fact			
id	*id_customer*	*id_driver*	id_merchant
10001	123	458	1

Merchant Dimension		
id	*name*	*category*
1	Warung Bu Iis	Indonesian

Item Fact			
id	*id_order*	*name*	*price*
101	10001	Nasi Goreng	30000
102	10001	Es Teh Manis	5000

Driver Dimension		
id	*name*	*gender*
456	Asep	M
457	Doni	M
458	Siti	F

Driver Search Fact			
id	*id_driver*	*name*	*status*
1	456	Asep	Rejected
2	457	Doni	Rejected
3	458	Siti	Accepted

Data Modeling - cont’d

Nested Denormalized

Nested Denormalized Dataset													
id_order	*id_customer*	*customer_name*	*phone_no*	*id_merchant*	*merchant_name*	*id_item*	*item_name*	*item_price*	*id_bid*	*id_driver*	*driver_name*	*driver_gender*	*bid_status*
10001	123	Dika	628112345678	1	Warung Bu Iis	101	Nasi Goreng	30000	1	456	Asep	M	Rejected
						102	Es Teh Manis	5000	2	457	Doni	M	Rejected
									3	458	Siti	F	Accepted

- Clean and structured data model
- Everything is in one place
- Complete insight of end to end data flow

Google Cloud

GO JEK

Data Governance

User mapping

Who gets access to data?	High Level Management	Business Analyst (and everyone else)	Data Developer
What kind of data granularity?	Low — business summary data	Medium — combination of summary and raw data	High — raw data
Why do they need it?	Decision Support System	Hypothesis Analysis	Exploratory Analysis, System Creation
How do they access it?	Worksheets or reports with simple slice and dice operations	Worksheets or reports with simple slice and dice operations	SQL queries for complex analysis
	SQL queries on structured datasets	SQL queries for more complex analysis	Custom code for statistical analysis, visualization, machine-learning, etc.

Data Governance - cont’d

High Level Management

- Identity Access Management
  - BigQuery User
- Google Cloud Storage
  - None
- BigQuery Dataset
  - Datamart

Business Analyst

- Identity Access Management
  - BigQuery User
- Google Cloud Storage
  - Read and Write Playground bucket
- BigQuery Dataset
  - Datamart
  - Data Warehouse

Data Developer

- Identity Access Management
  - BigQuery User
- Google Cloud Storage
  - Read and Write Playground + Production buckets
- BigQuery Dataset
  - Datamart
  - Data Warehouse
  - Data Lake

Data Governance - cont'd

Data Discovery

- BigQuery has The capability to write descriptions directly in the tables
- Not only that, we are able to extract table details information into files
- From these files, we can build Data Discovery tools to help us pointing out specific metrics from hundreds of tables or thousands data points

Table Details: comments

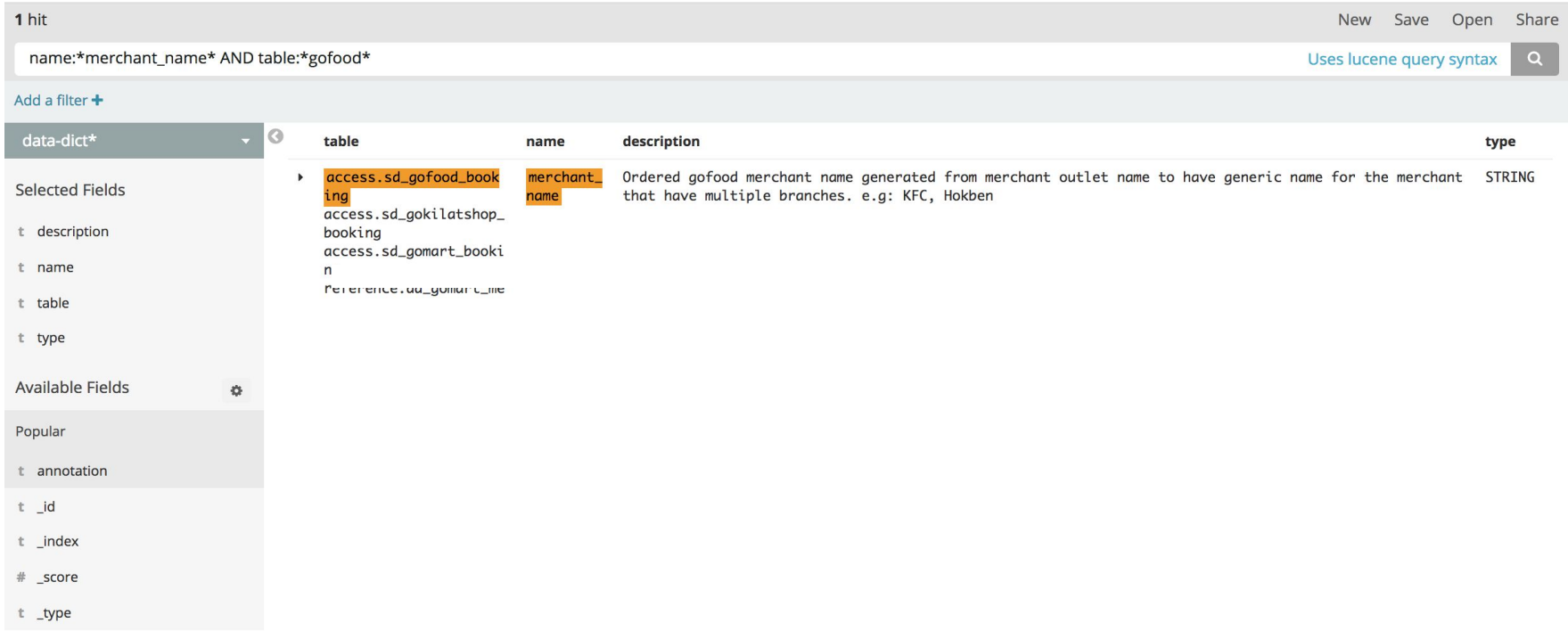
Schema	Details	Preview

id	INTEGER	NULLABLE	Unique comment ID
by	STRING	NULLABLE	Username of commenter
author	STRING	NULLABLE	Username of author
time	INTEGER	NULLABLE	Unix time
time_ts	TIMESTAMP	NULLABLE	Human readable time in UTC (format: YYYY-MM-DD hh:mm:ss)
text	STRING	NULLABLE	Comment text
parent	INTEGER	NULLABLE	Parent comment ID
deleted	BOOLEAN	NULLABLE	Is deleted?
dead	BOOLEAN	NULLABLE	Is dead?
ranking	INTEGER	NULLABLE	Comment ranking

Google Cloud

GO JEK

DataDex Prototype



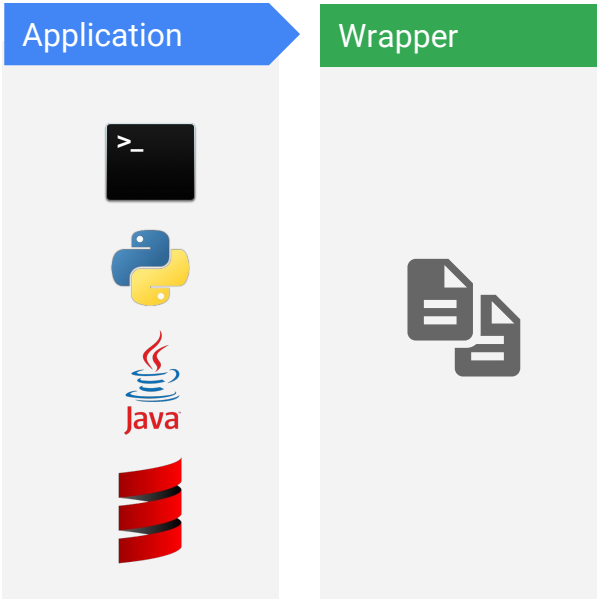
Google Cloud

GO JEK

Operations

Application Management

- Define rules and mandatory functions
  - To help manage applications with multiple programming languages
- Create a universal executor framework
  - Uniformness in running jobs



Operations - cont’d

Wrapper

- Define rules and mandatory functions
- Create universal executor framework for any programming language
- Provide easy execution processes

Dependency Management

- Manage dependencies for hundreds of jobs
- Manage job scheduler and orchestration
- Job monitoring and alerting

Monitoring

- Monitoring for Infrastructure health
- Alerting for certain thresholds or conditions
- Deep dive analysis for specific timeframes or incidents

Google Cloud

GO JEK

Next Gen DWH - Recap

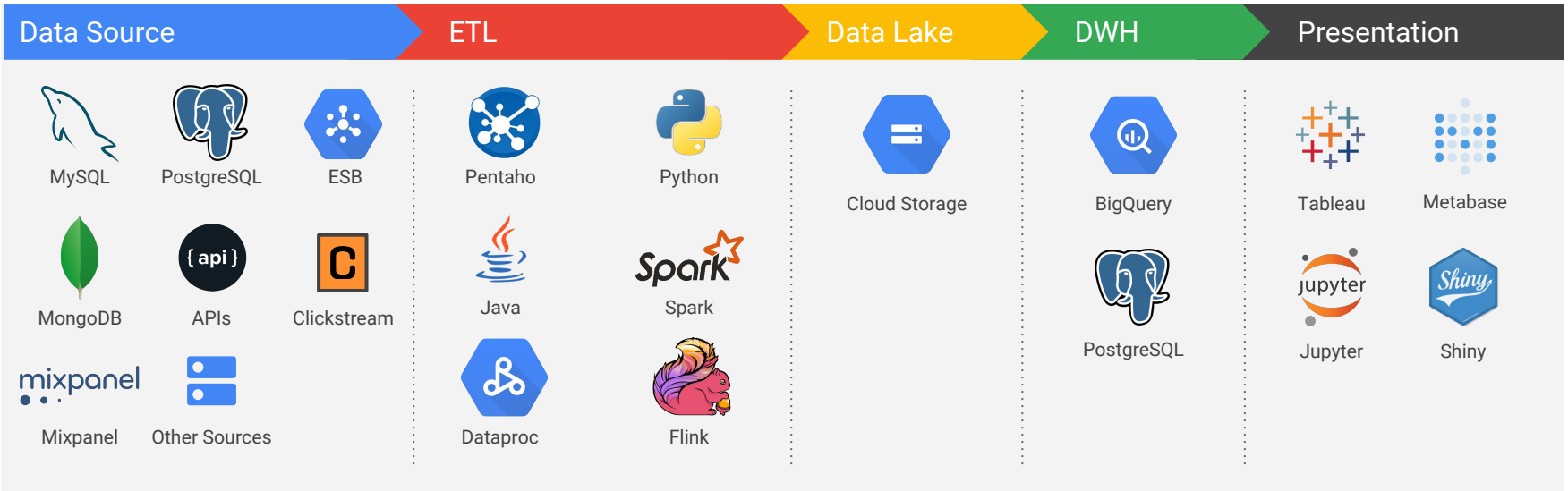
Google Cloud

Summary

Google Cloud

GO JEK

DW Architecture (end of 2017)



Current Stats of Q4 of 2017

>30%

Growing Data Volume per Month

*This is only business metrics data collected by BI

>150

Multi Resolution Datasets

*Contains full resolution, reusable summary and roll-up dataset

>1000

Data Points

>6000

Metabase Cards & Tableau Dashboards

>500

Average Daily Metabase & Tableau Users

Google Cloud

GO JEK

So, it’s all good?

Not really - the migration itself, is ~~crazy~~ super hard:

Almost 2 quarters of migration activity

- Migrating hundreds of tables and thousands of data points
- Requires careful planning (e.g. step by step migration without downtime)

Defying the logic

- Different way of doing things while applying it to new architecture (e.g. putting transactional data into an OLAP DB like BigQuery)

Halp! Need backup!

- We need dedicated resources to do the migration

The Takeaway

Google Cloud

The Takeaway

Some helpful tips based on our journey:

Make sure every use case is considered before you decide to do DWH migration

You'll probably need a complementary system to cover all of your possible use cases

Put data governance as your priority from the start

Defining data governance is never easy, but opening your data without governance is a big mistake

Application management is important to have

Make sure you have operational uniformity between across applications with various programming languages

Thank you

Google Cloud

GO JEK