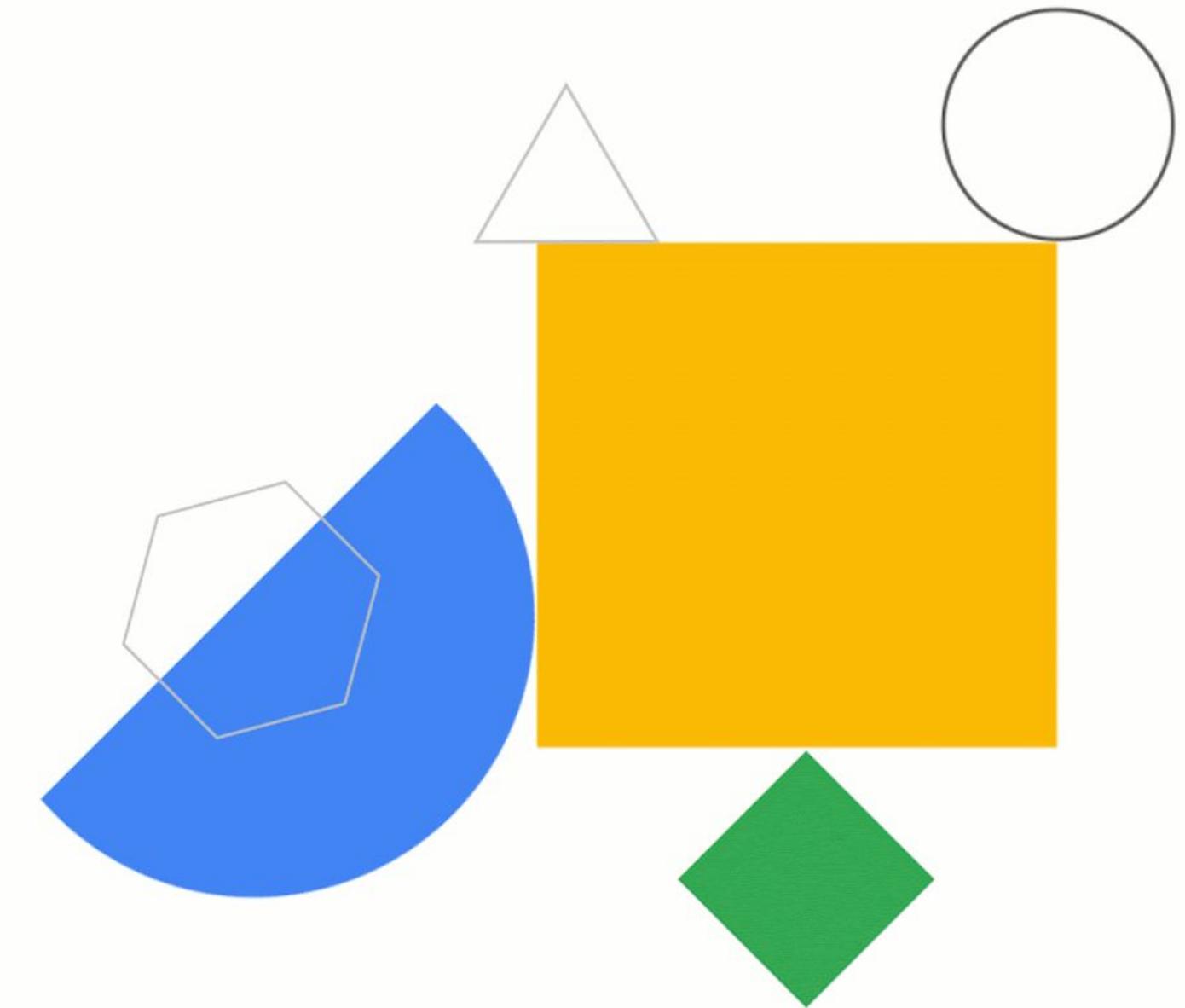


# Modernizing Data Lakes and Data Warehouses with Google Cloud



# Data Engineering on Google Cloud learning path

- 1 Modernizing Data Lakes and Data Warehouses with Google Cloud
- 2 Building Batch Data Pipelines on Google Cloud
- 3 Building Resilient Streaming Analytics Systems on Google Cloud
- 4 Smart Analytics, Machine Learning and AI on Google Cloud



# Day 1 agenda

- 01 Introduction to Data Engineering
- 02 Building a Data Lake
- 03 Building a Data Warehouse



# Introduction to Data Engineering

# Introduction to Data Engineering

- 01 The role of a data engineer
- 02 Data engineering challenges
- 03 Introduction to BigQuery
- 04 Data lakes and data warehouses
- 05 Transactional databases versus data warehouses
- 06 Partner effectively with other data teams
- 07 Manage data access and governance
- 08 Build production-ready pipelines
- 09 Google Cloud customer case study



# Introduction to Data Engineering

- 01 **The role of a data engineer**
- 02 Data engineering challenges
- 03 Introduction to BigQuery
- 04 Data lakes and data warehouses
- 05 Transactional databases versus data warehouses
- 06 Partner effectively with other data teams
- 07 Manage data access and governance
- 08 Build production-ready pipelines
- 09 Google Cloud customer case study



# A data engineer builds data pipelines to enable data-driven decisions

Get the data to where it can be useful

Get the data into a usable condition

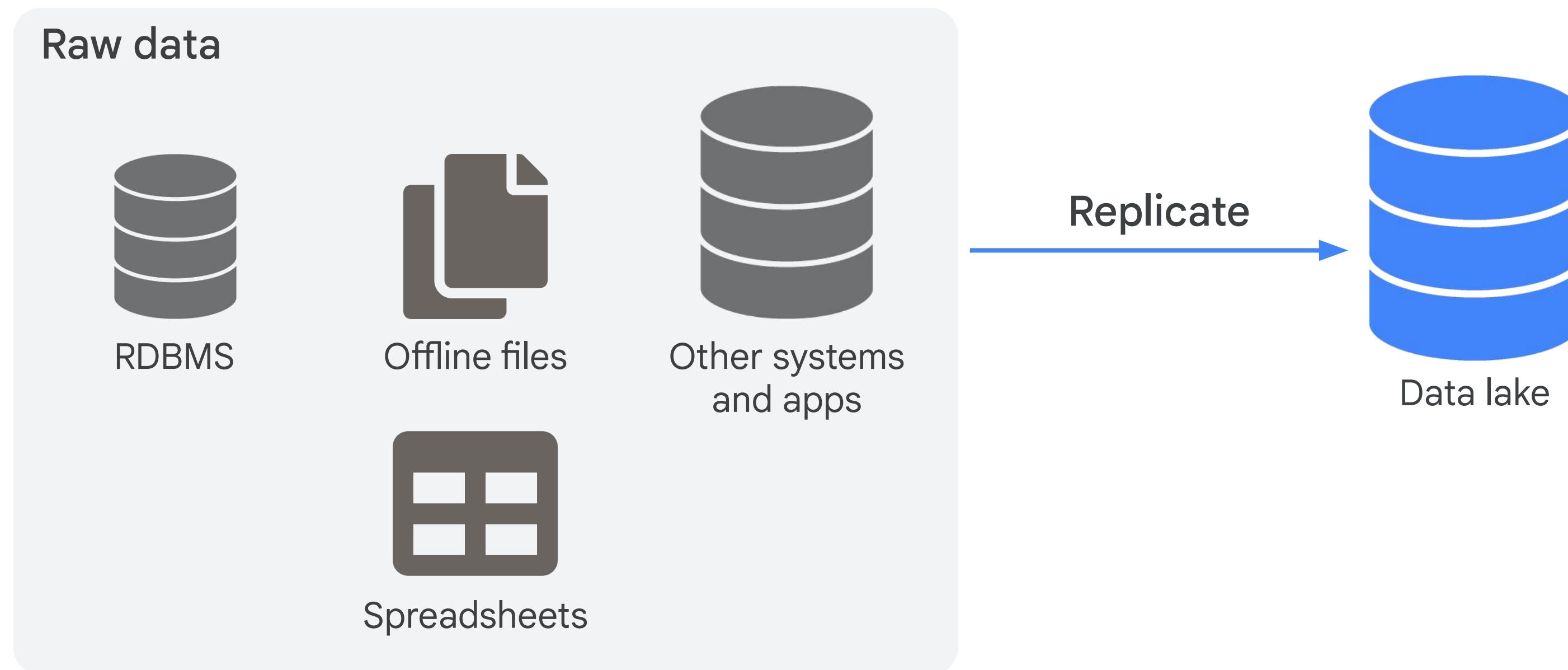
Add new value to the data

Manage the data

Productionize data processes

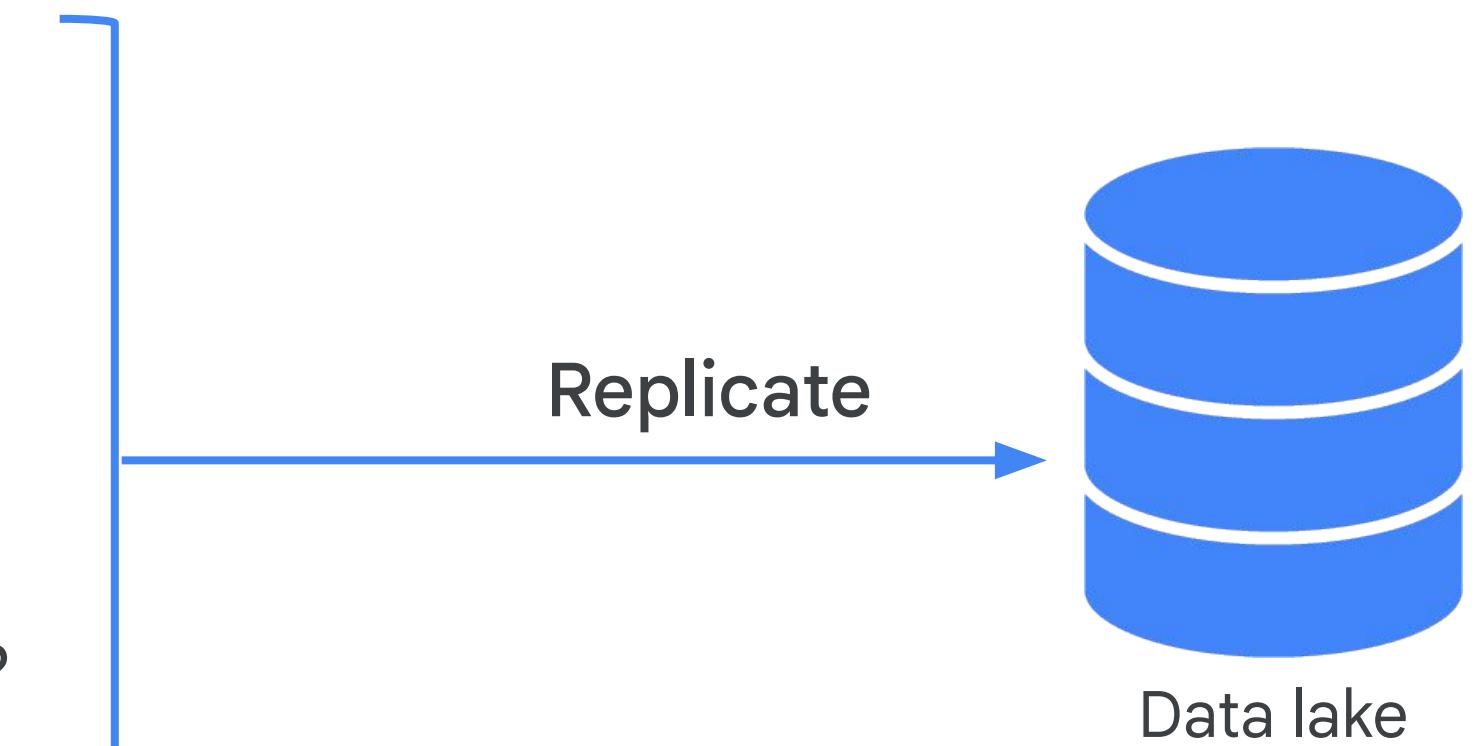
So... how do we get the raw data from multiple systems and where can we store it durably?

# A data lake brings together data from across the enterprise into a single location



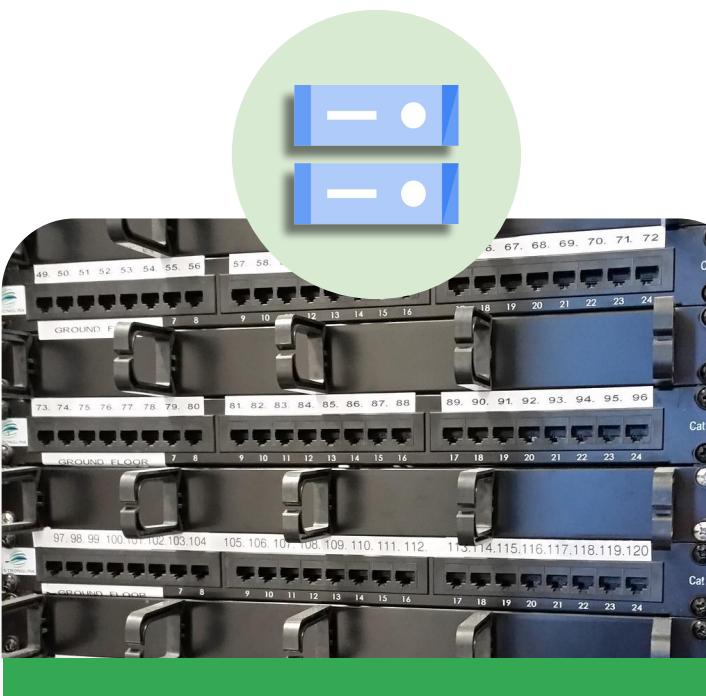
# Key considerations when building a data lake

1. Can your data lake handle all the types of data you have?
2. Can it scale to meet the demand?
3. Does it support high-throughput ingestion?
4. Is there fine-grained access control to objects?
5. Can other tools connect easily?



We need an elastic data container  
that is flexible and durable to  
stage all our data ...

# Cloud Storage is designed for 99.999999999% annual durability



Backup



Replace/decommission infrastructure



Analytics and ML



Content storage and delivery

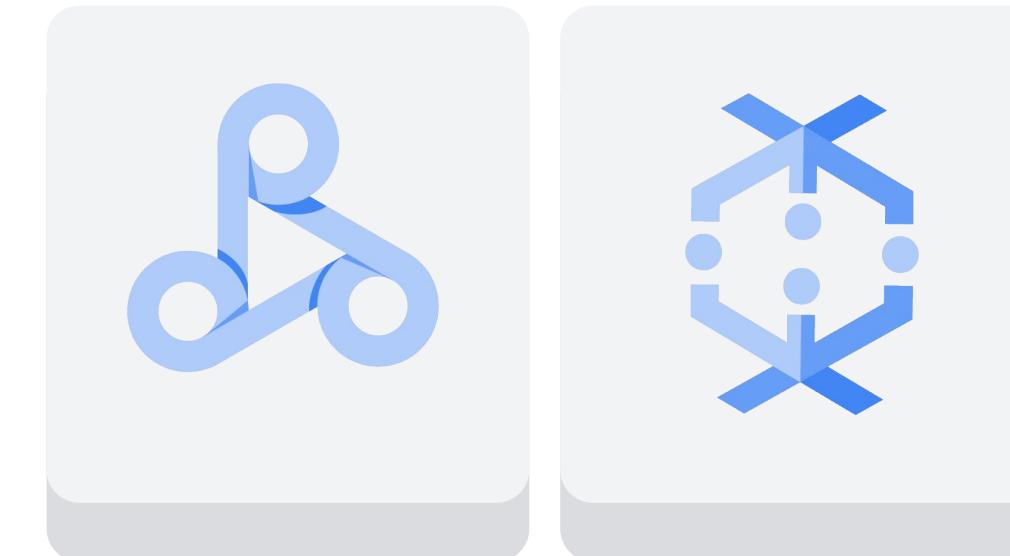
Quickly create buckets with Cloud Shell  
`gsutil mb gs://your-project-name`

# What if your data is not usable in its original form?



**Extract, Transform, and Load**

Data processing



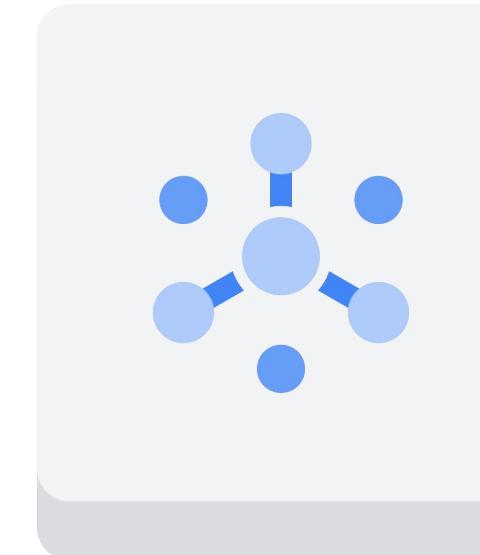
Dataproc

Dataflow

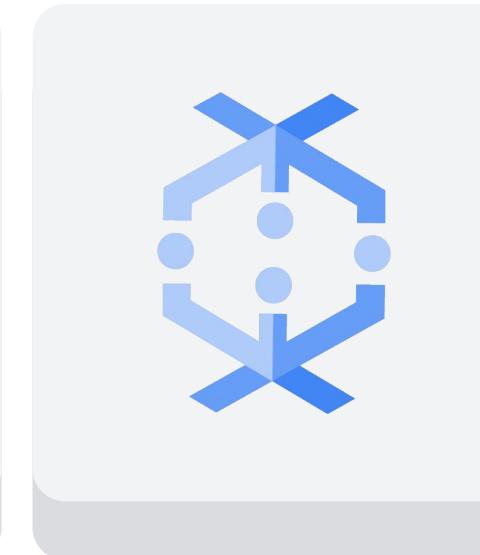
# What if your data arrives continuously and endlessly?



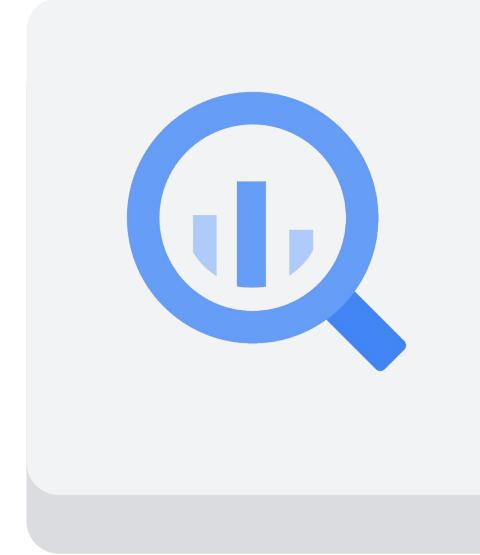
Streaming data processing



Pub/Sub



Dataflow



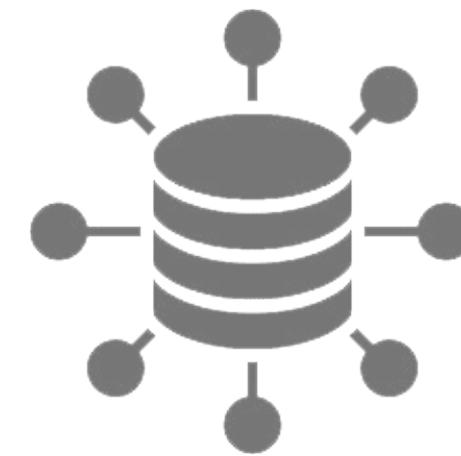
BigQuery

# Introduction to Data Engineering

- 01 The role of a data engineer
- 02 **Data engineering challenges**
- 03 Introduction to BigQuery
- 04 Data lakes and data warehouses
- 05 Transactional databases versus data warehouses
- 06 Partner effectively with other data teams
- 07 Manage data access and governance
- 08 Build production-ready pipelines
- 09 Google Cloud customer case study



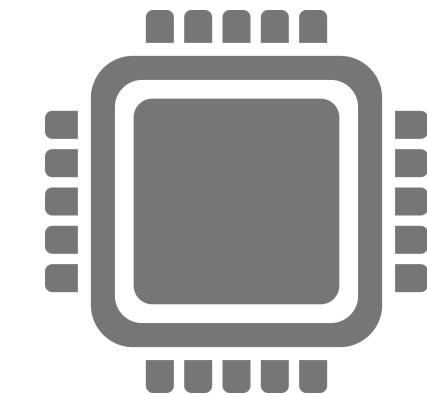
# Common challenges encountered by data engineers



Access to data



Data accuracy  
and quality

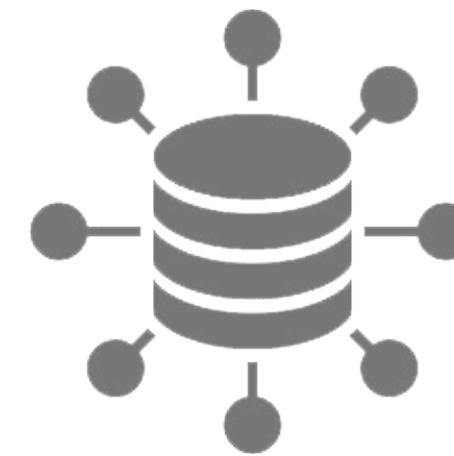


Availability of  
computational  
resources



Query  
performance

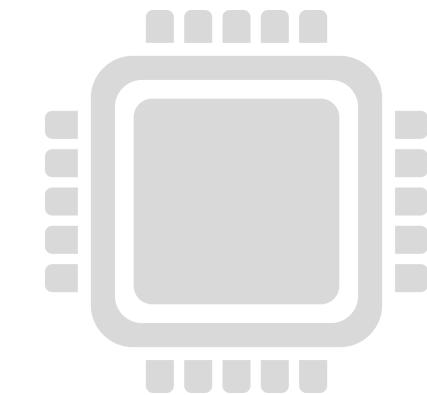
# Challenge: Consolidating disparate datasets, data formats, and manage access at scale



Access to data



Data accuracy  
and quality



Availability of  
computational  
resources



Query  
performance

# Getting insights across multiple datasets is difficult without a data lake

Data is scattered across Google Analytics 360, CRM, and Campaign Manager products, among other sources.

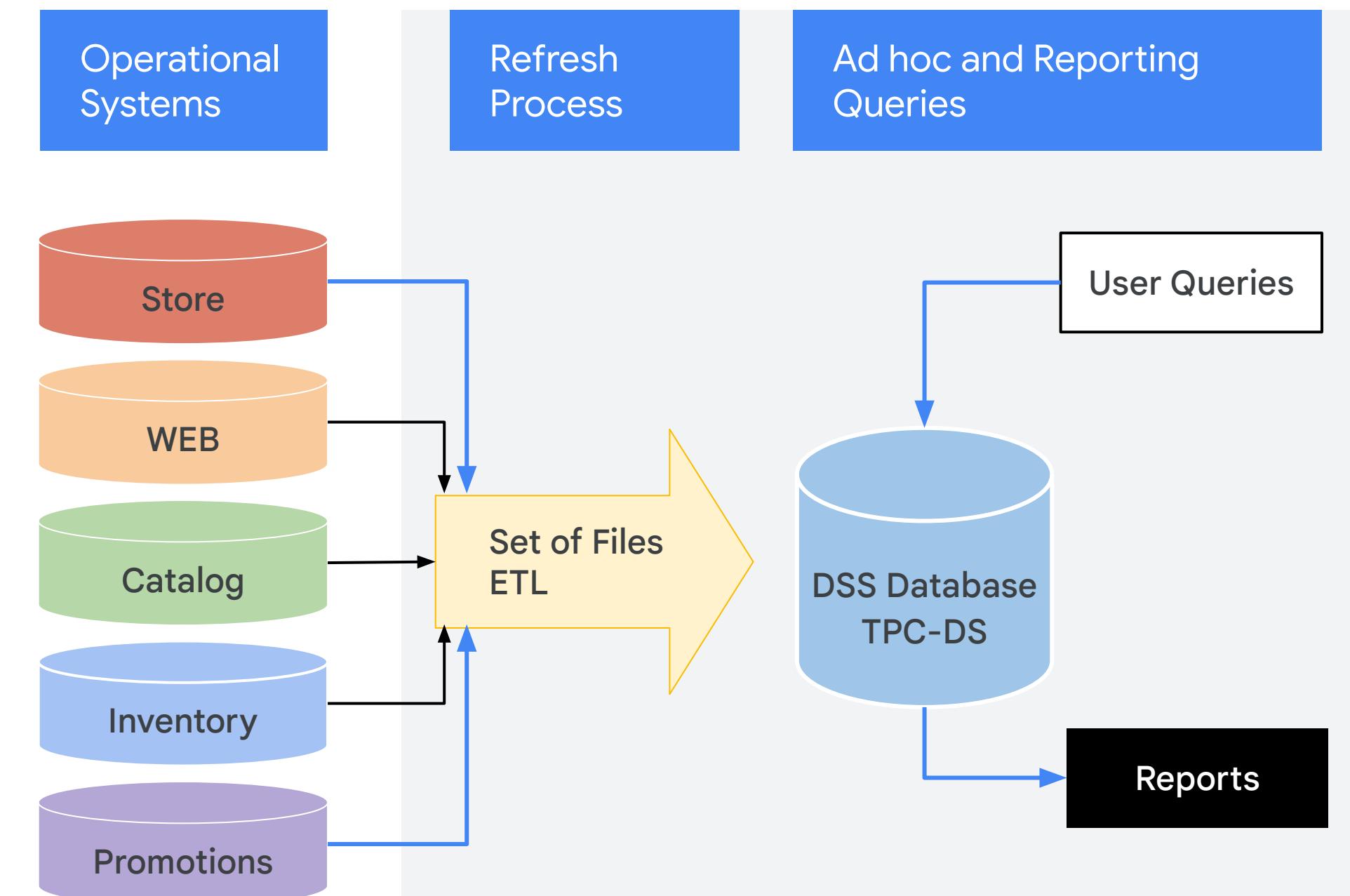
Customer and sales data is stored in a CRM system.



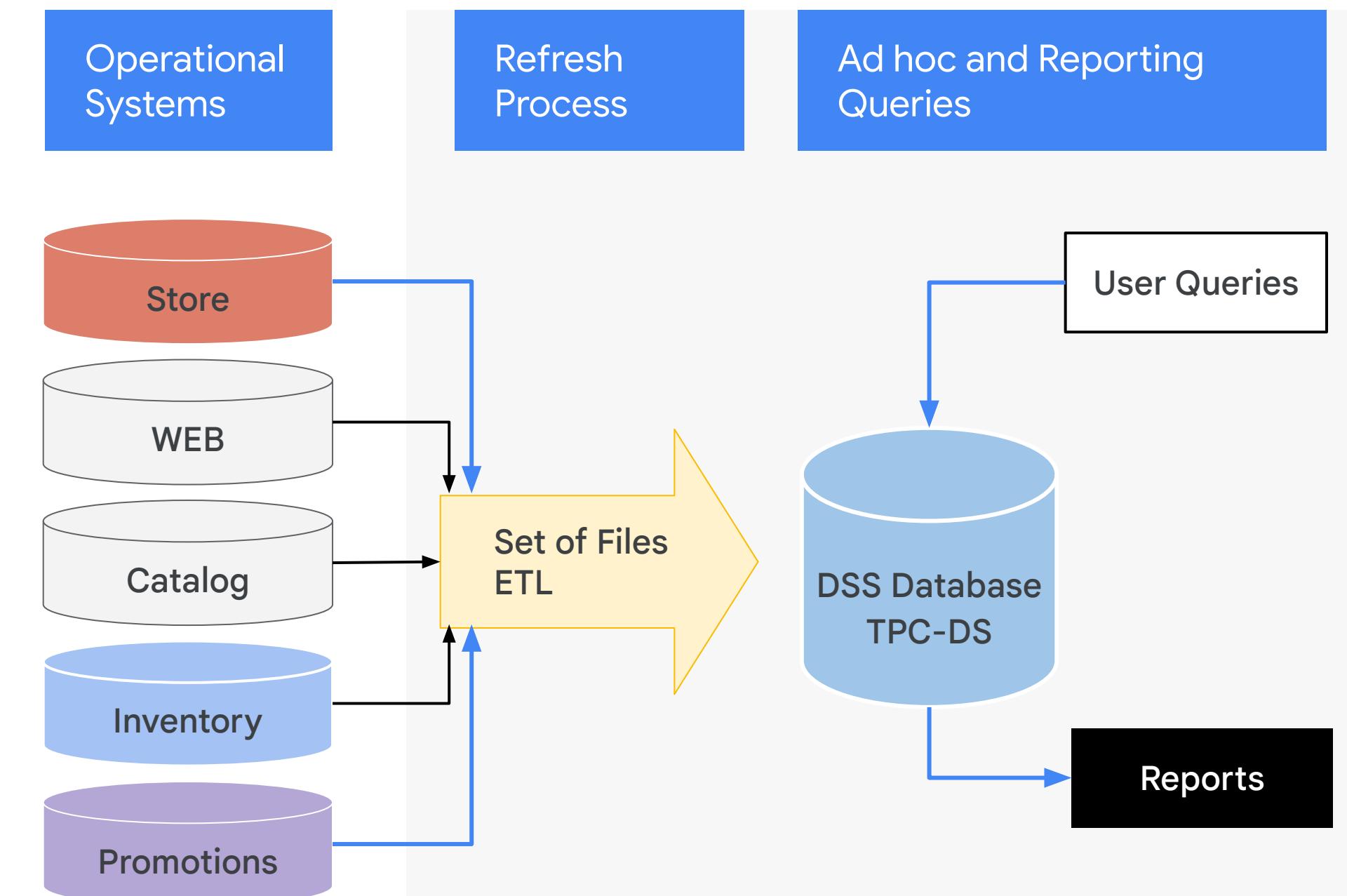
No common tool exists to analyze data and share results with the rest of the organization.

Some data is not in a queryable format.

# Data is often siloed in many upstream source systems



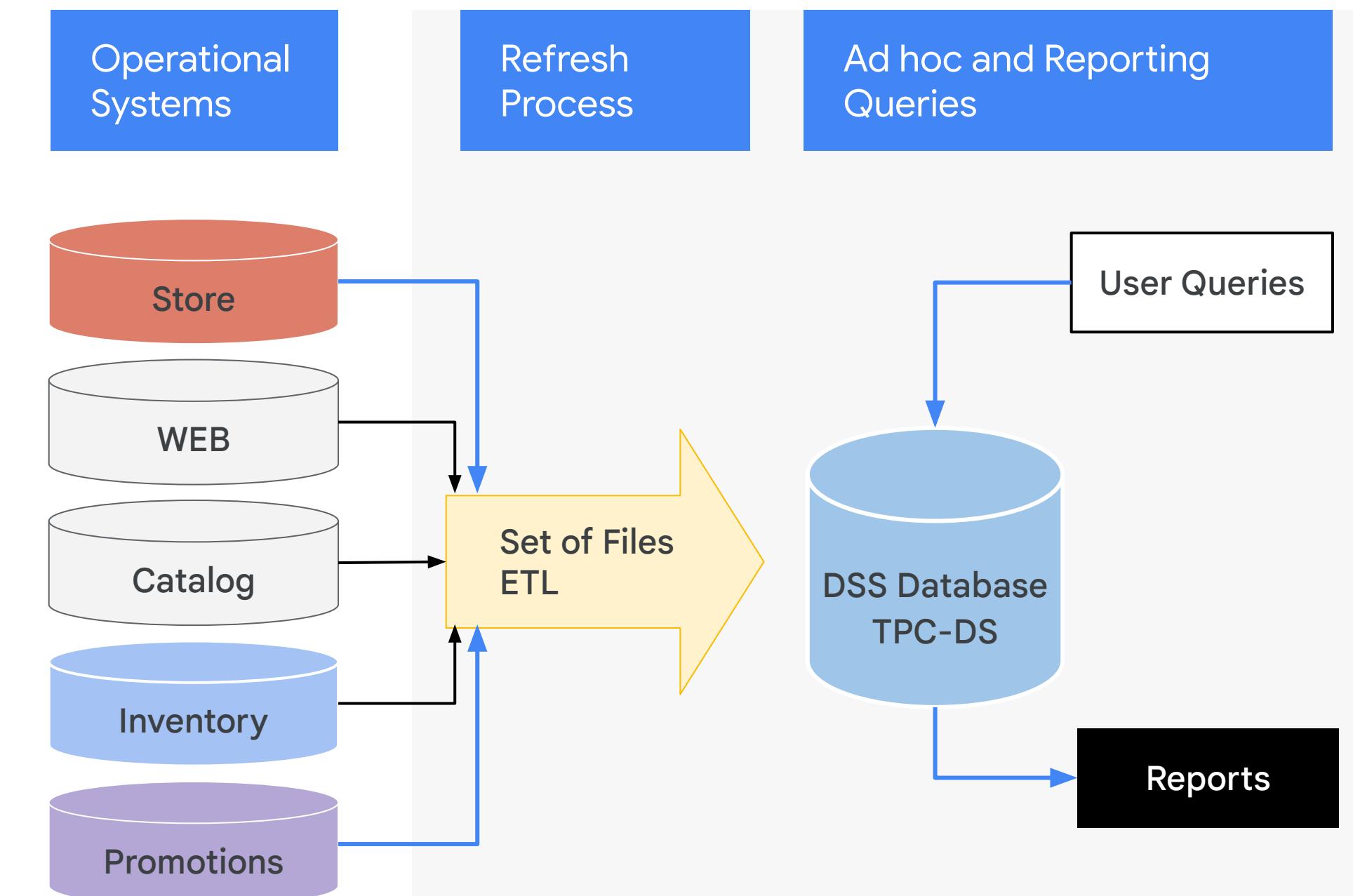
# Data is often siloed in many upstream source systems



# Data is often siloed in many upstream source systems

## Example query:

Give me all the in-store promotions for recent orders and their **inventory levels**.

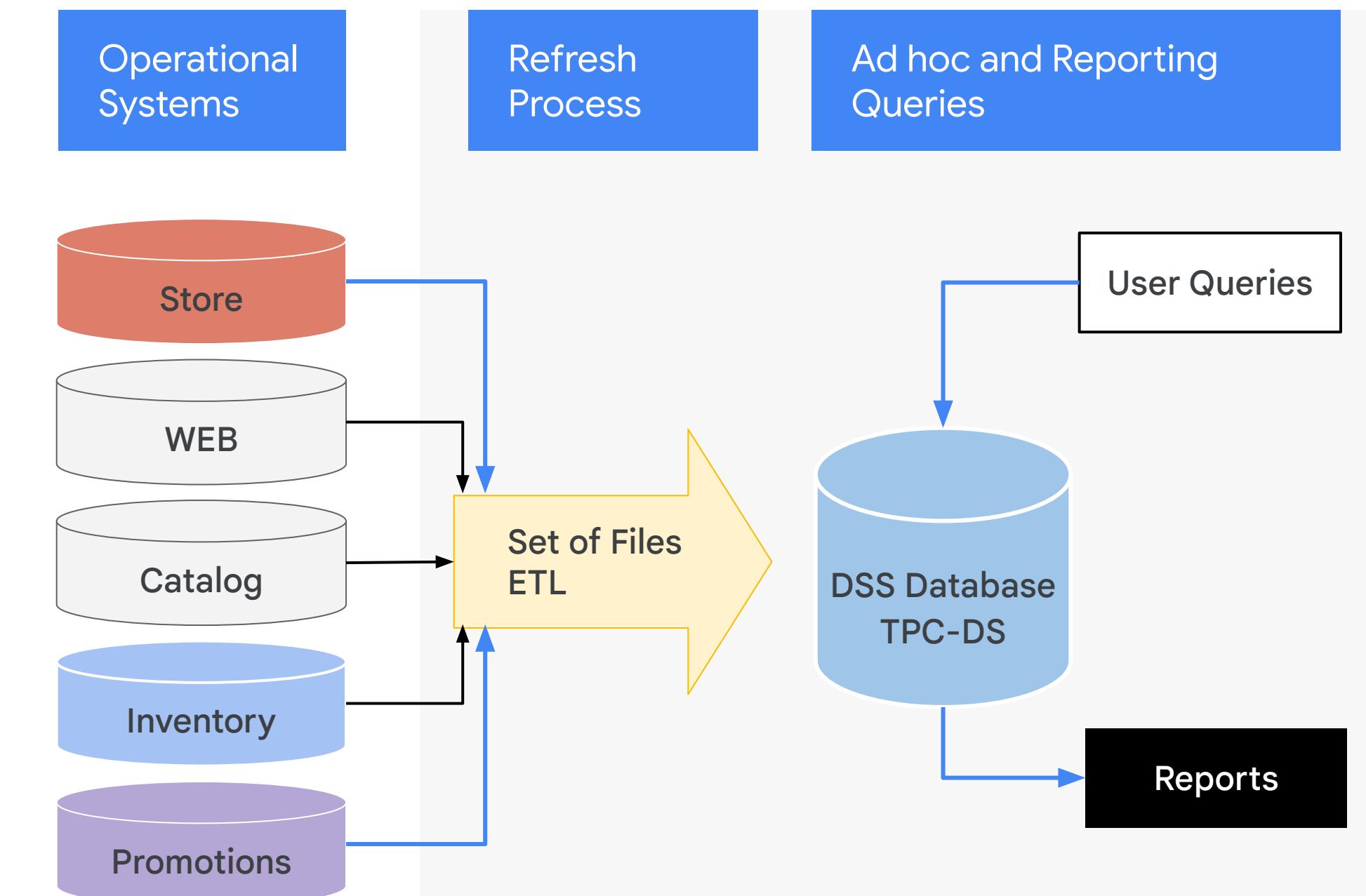


# Data is often siloed in many upstream source systems

## Example query:

Give me all the in-store promotions for recent orders and their **inventory levels**.

Stored in a separate system and restricted access



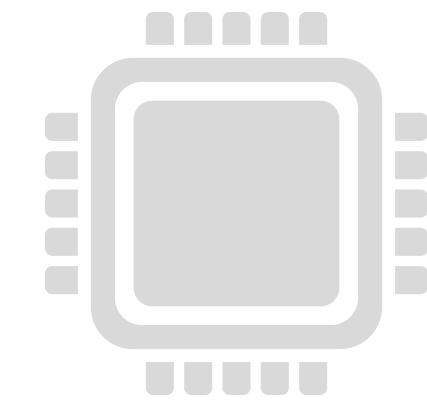
# Challenge: Cleaning, formatting, and getting the data ready for useful business insights in a data warehouse



Access to data



Data accuracy  
and quality

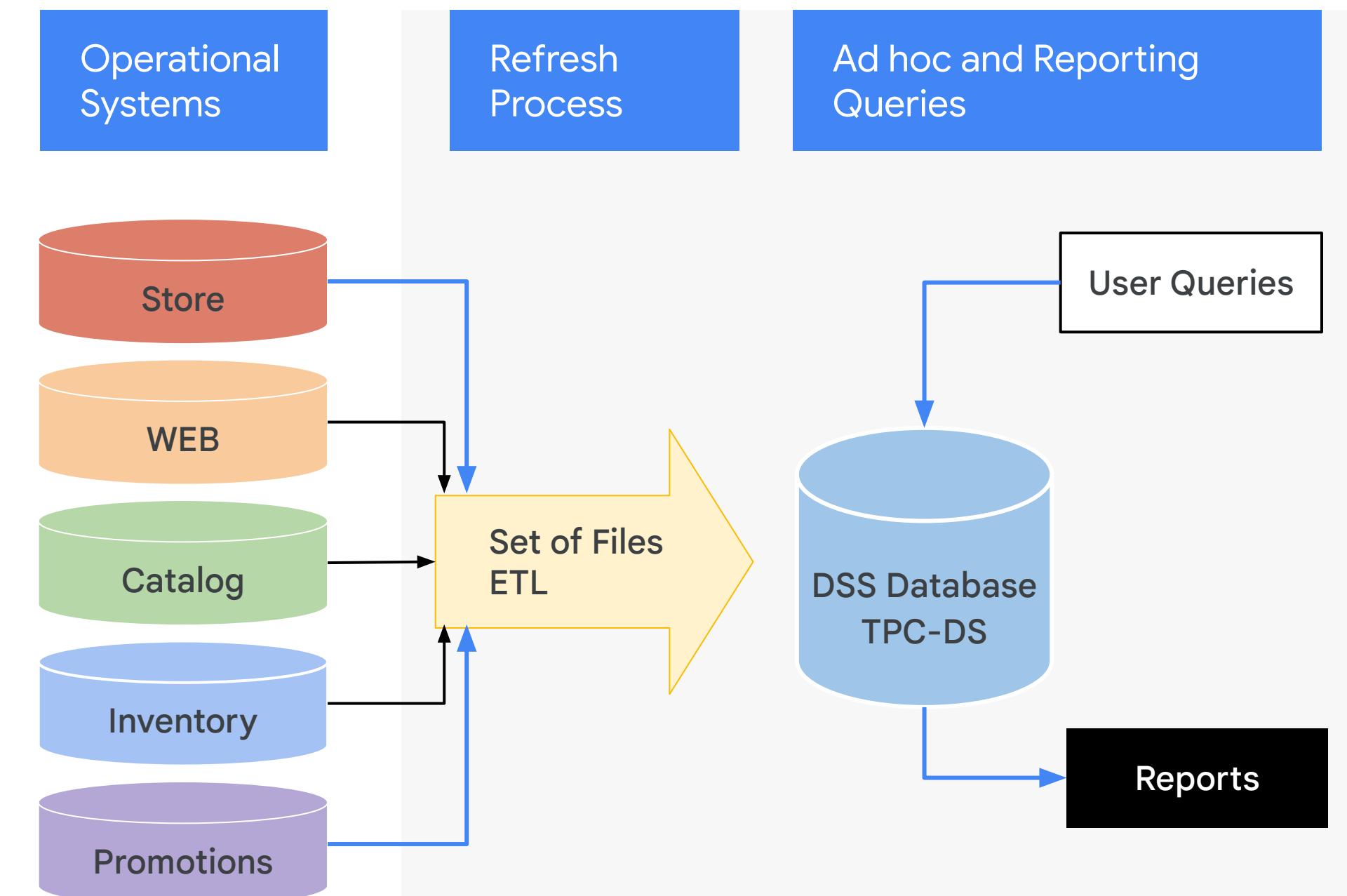


Availability of  
computational  
resources



Query  
performance

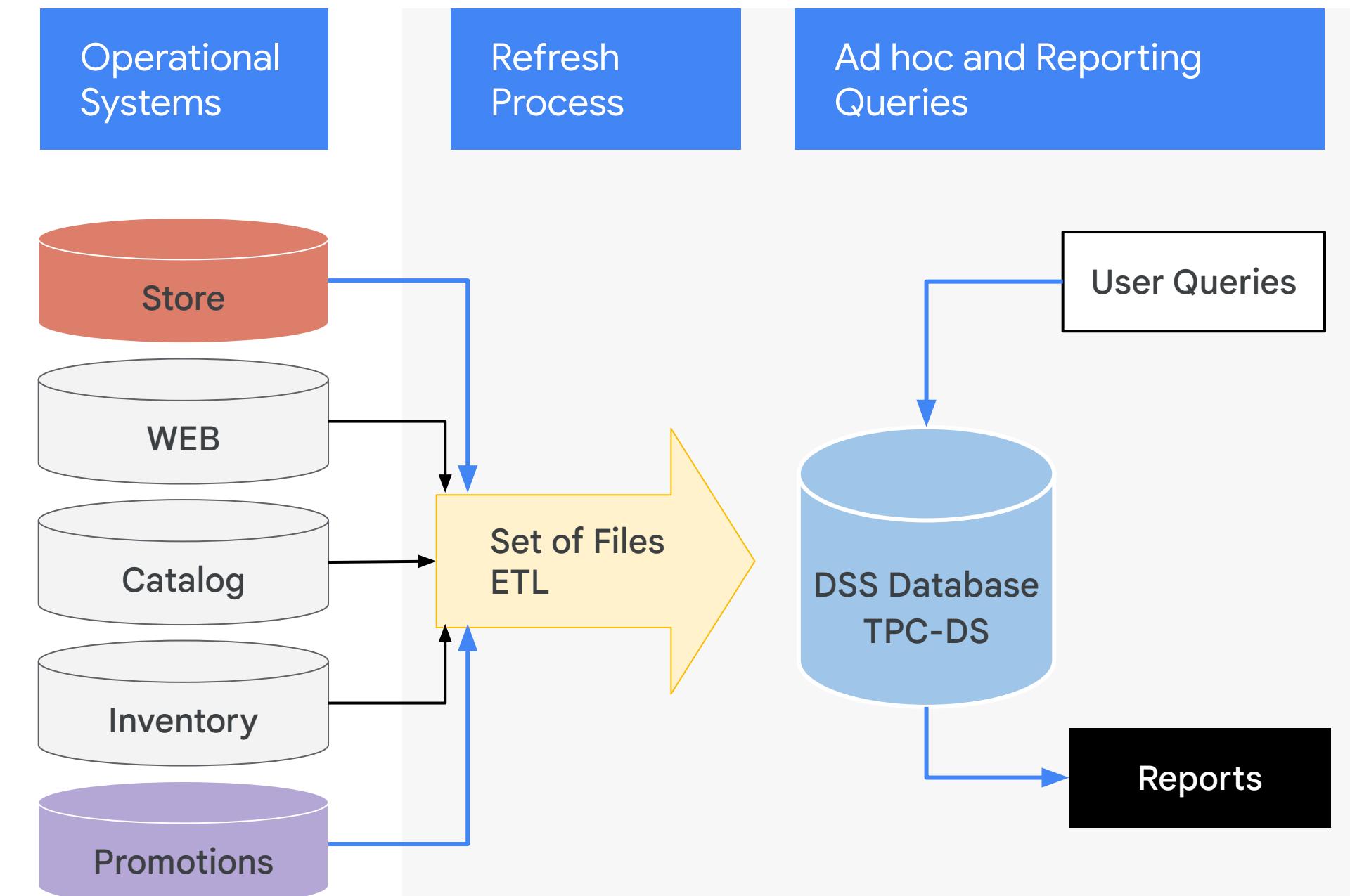
# Assume that any raw data from source systems needs to be cleaned, transformed, and stored in a data warehouse



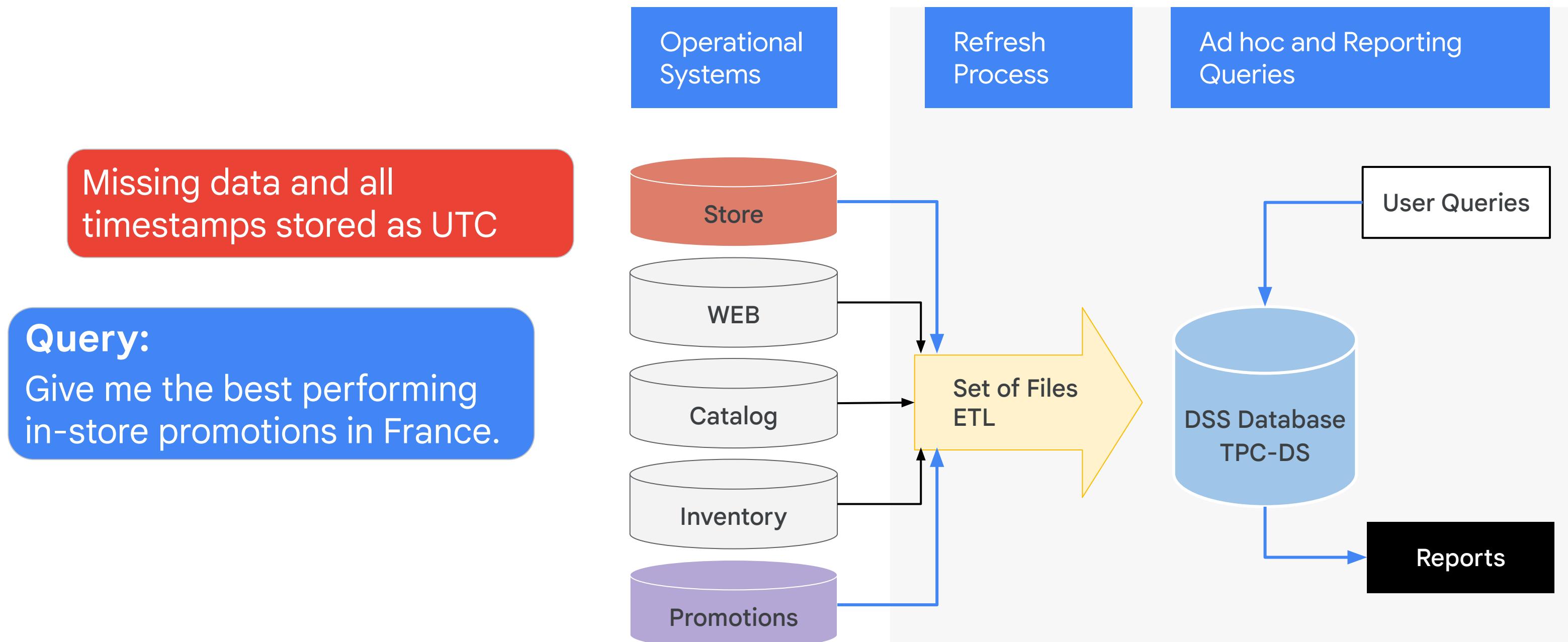
# Assume that any raw data from source systems needs to be cleaned, transformed, and stored in a data warehouse

**Query:**

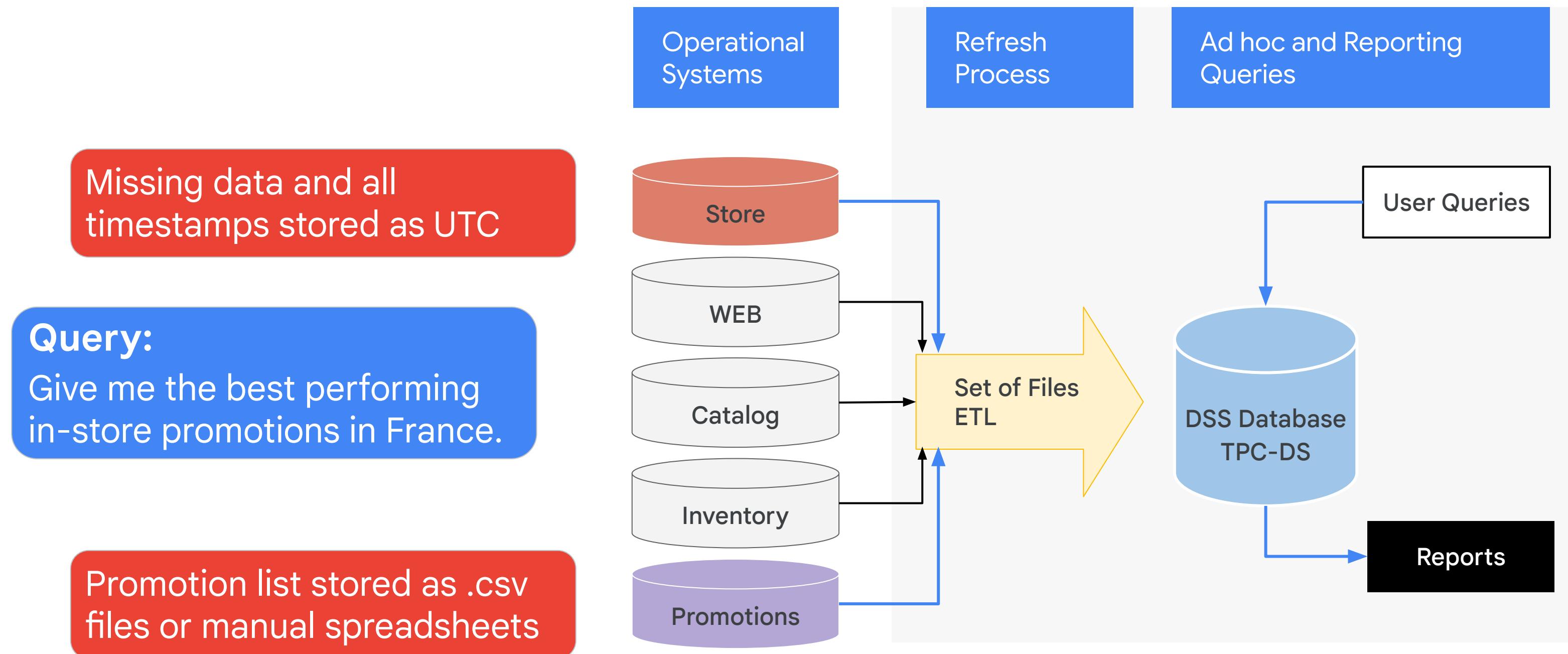
Give me the best performing in-store promotions in France.



# Assume that any raw data from source systems needs to be cleaned, transformed, and stored in a data warehouse



# Assume that any raw data from source systems needs to be cleaned, transformed, and stored in a data warehouse



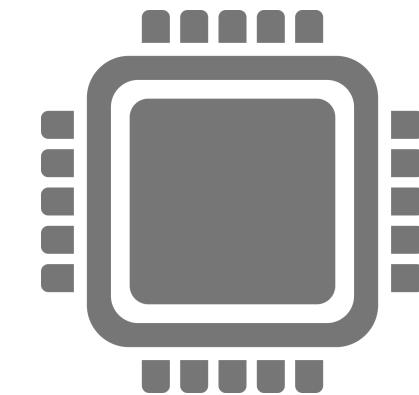
# Challenge: Ensuring you have the compute capacity to meet peak-demand for your team



Access to data



Data accuracy  
and quality

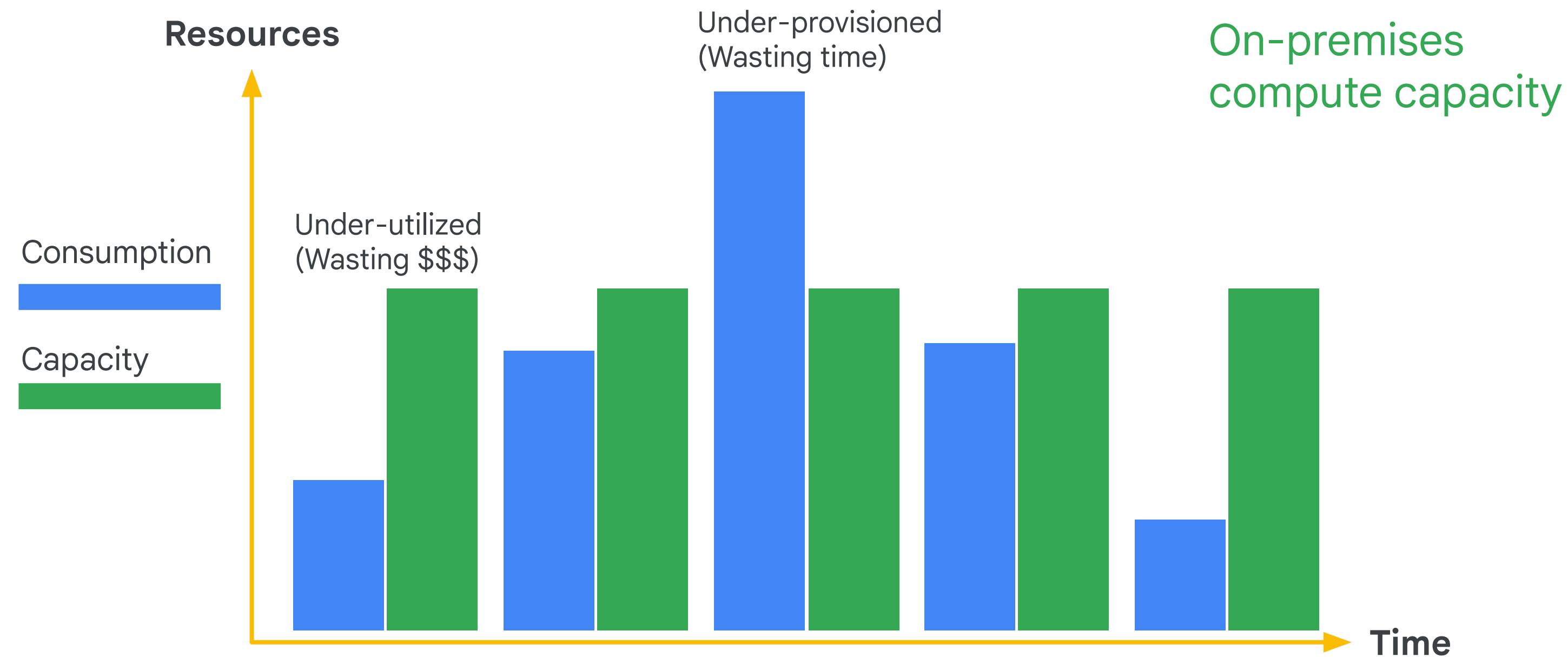


Availability of  
computational  
resources



Query  
performance

# Challenge: Data Engineers need to manage server and cluster capacity if using on-premise



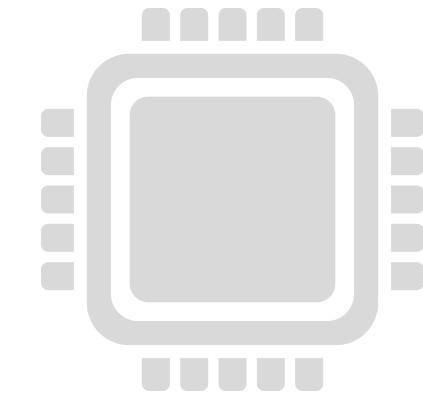
# Challenge: Queries need to be optimized for performance (caching, parallel execution)



Access to data



Data accuracy  
and quality



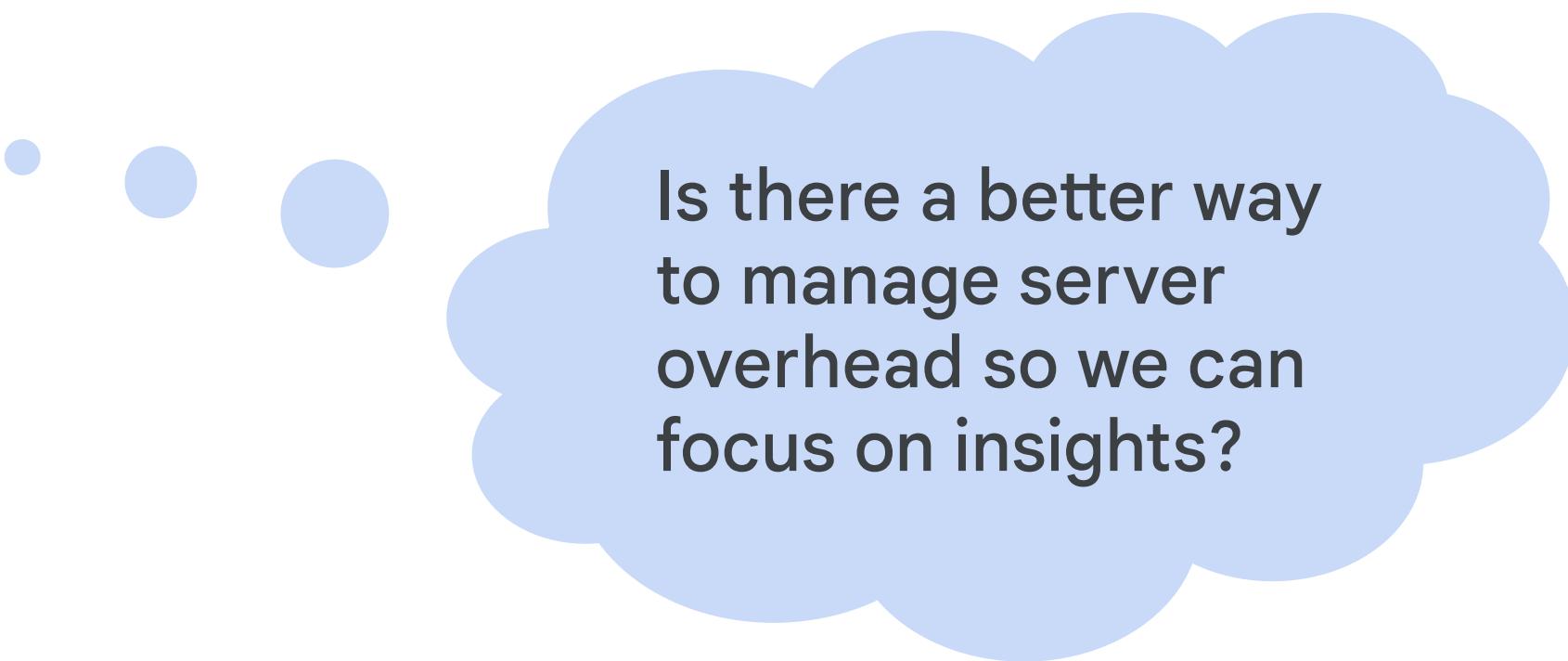
Availability of  
computational  
resources



Query  
performance

# Challenge: Managing query performance on-premise comes with added overhead

- Choosing a query engine.
- Continually patching and updating query engine software.
- Managing clusters and when to re-cluster.
- Optimize for concurrent queries and quota / demand between teams.

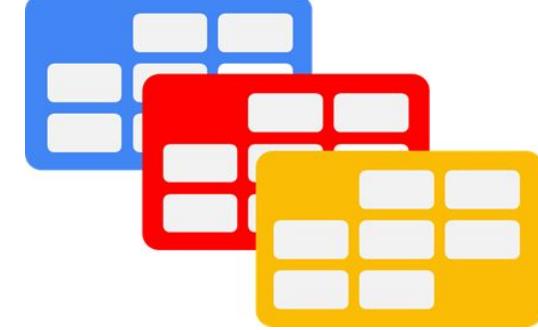
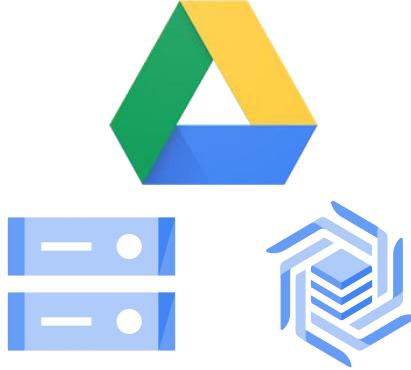
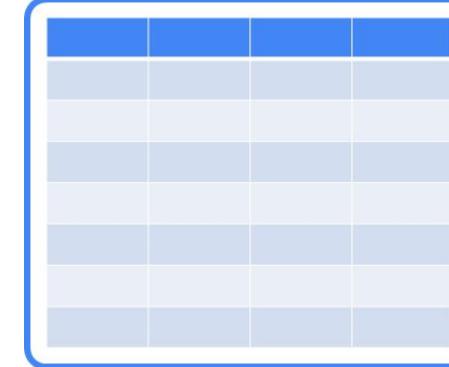


# Introduction to Data Engineering

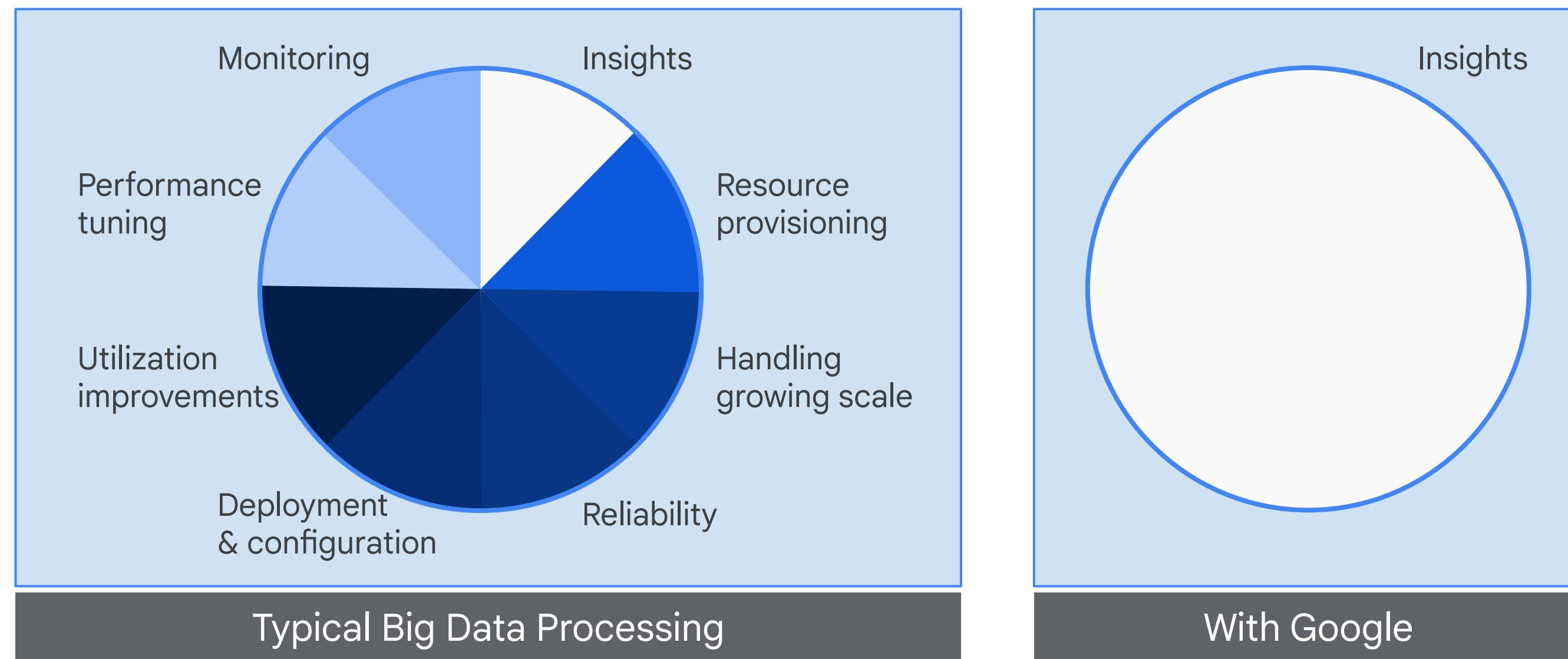
- 01 The role of a data engineer
- 02 Data engineering challenges
- 03 **Introduction to BigQuery**
- 04 Data lakes and data warehouses
- 05 Transactional databases versus data warehouses
- 06 Partner effectively with other data teams
- 07 Manage data access and governance
- 08 Build production-ready pipelines
- 09 Google Cloud customer case study



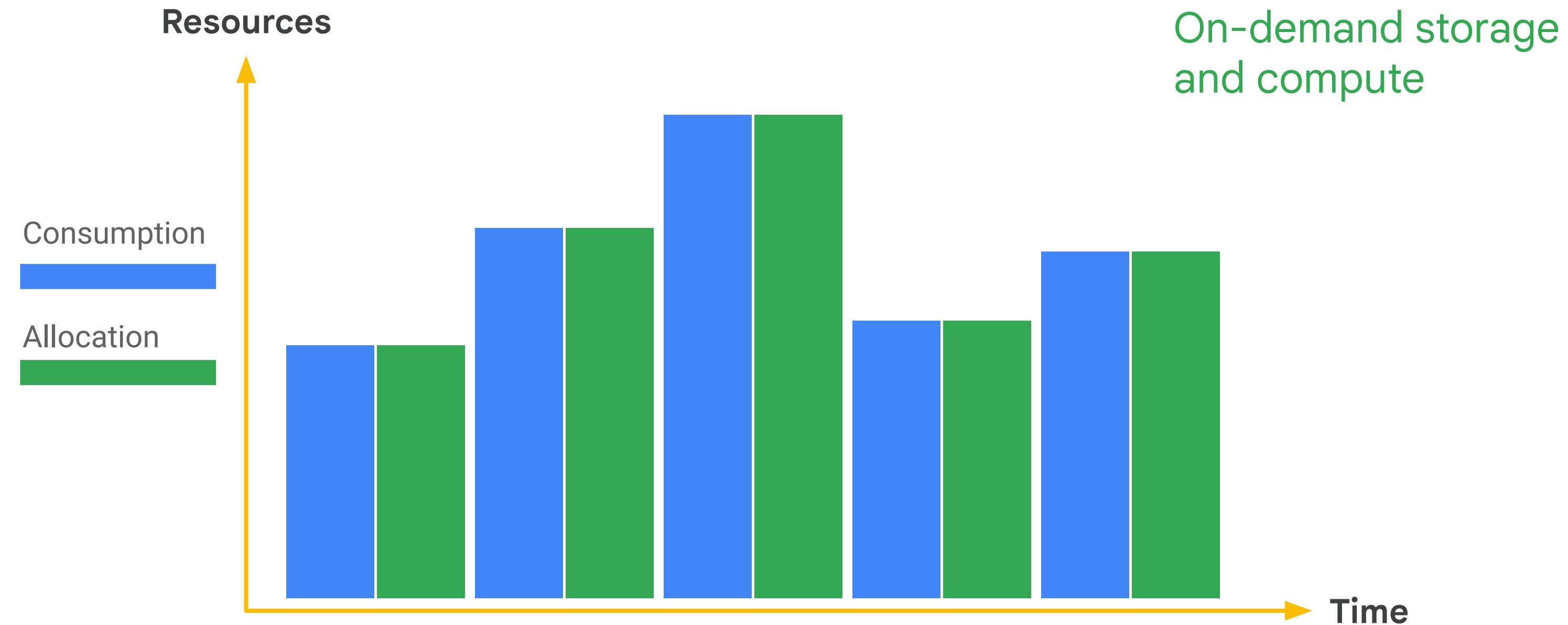
# BigQuery is Google's data warehouse solution

				
Data warehouse	Data mart	Data lake	Tables and views	Grants
BigQuery replaces a typical data warehouse hardware setup.	BigQuery organizes data tables into units called datasets.	BigQuery defines schemas and issues queries directly on external data sources.	Function the same way as in a traditional data warehouse.	IAM grants permission to perform specific actions.

# Cloud allows data engineers to spend less time managing hardware and enabling scale. Let Google do that for you



# You don't need to provision resources before using BigQuery



# Introduction to Data Engineering

- 01 The role of a data engineer
- 02 Data engineering challenges
- 03 Introduction to BigQuery
- 04 **Data lakes and data warehouses**
- 05 Transactional databases versus data warehouses
- 06 Partner effectively with other data teams
- 07 Manage data access and governance
- 08 Build production-ready pipelines
- 09 Google Cloud customer case study



# A data engineer gets data into a useable condition

Get the data to where it can be useful

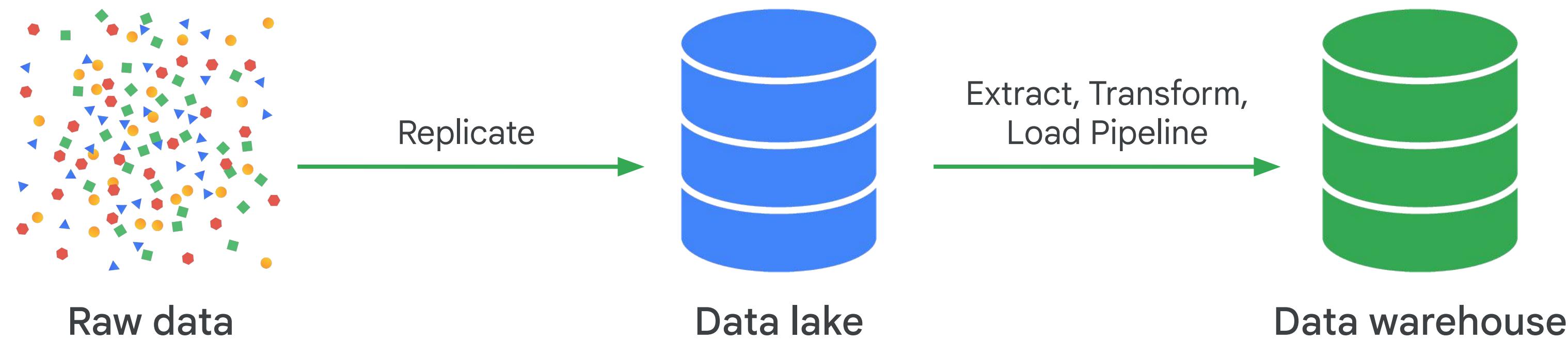
Get the data into a usable condition

Add new value to the data

Manage the data

Productionize data processes

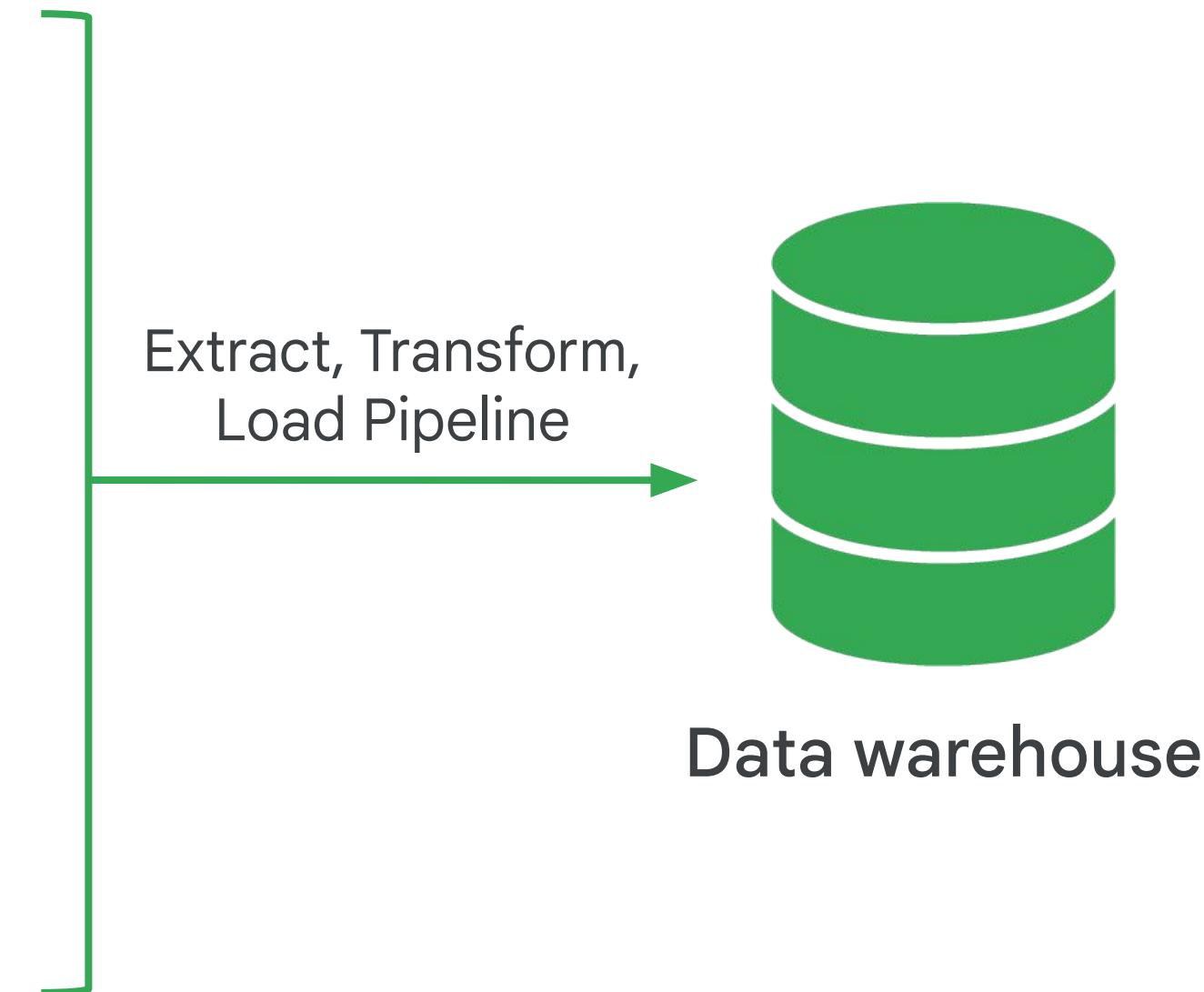
# A data warehouse stores transformed data in a usable condition for business insights



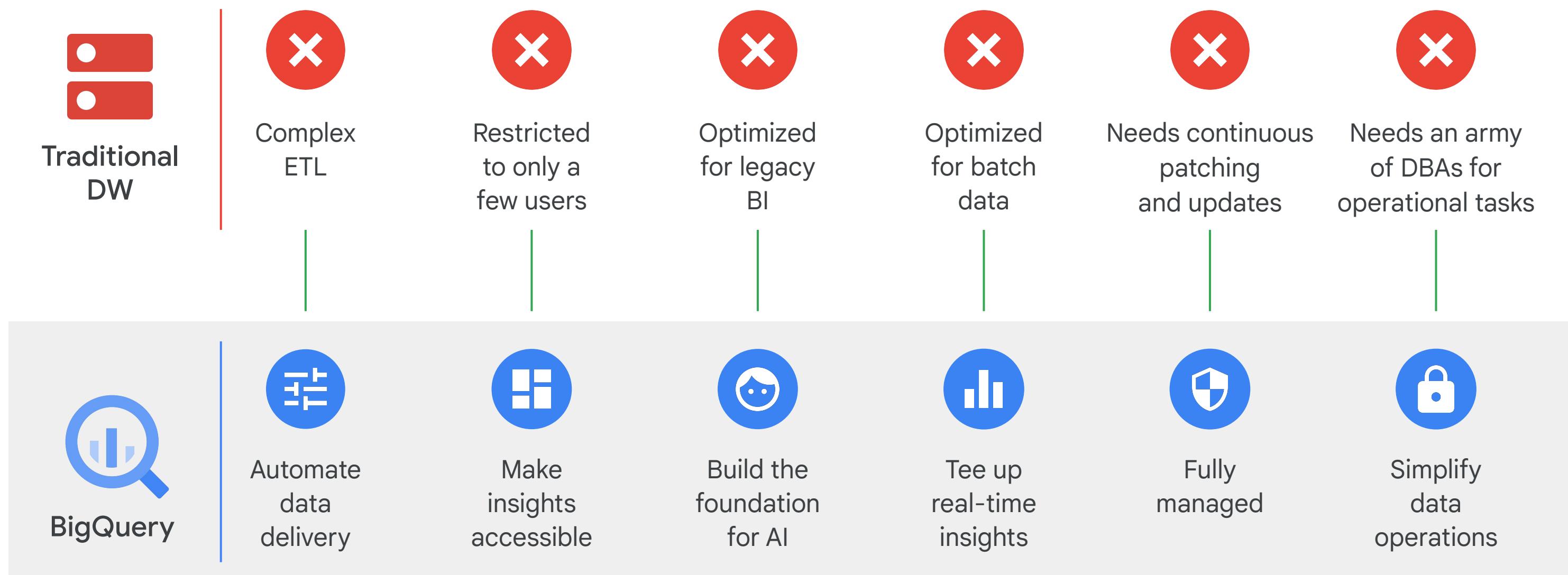
What are the key considerations when deciding between data warehouse options?

# Considerations when choosing a data warehouse

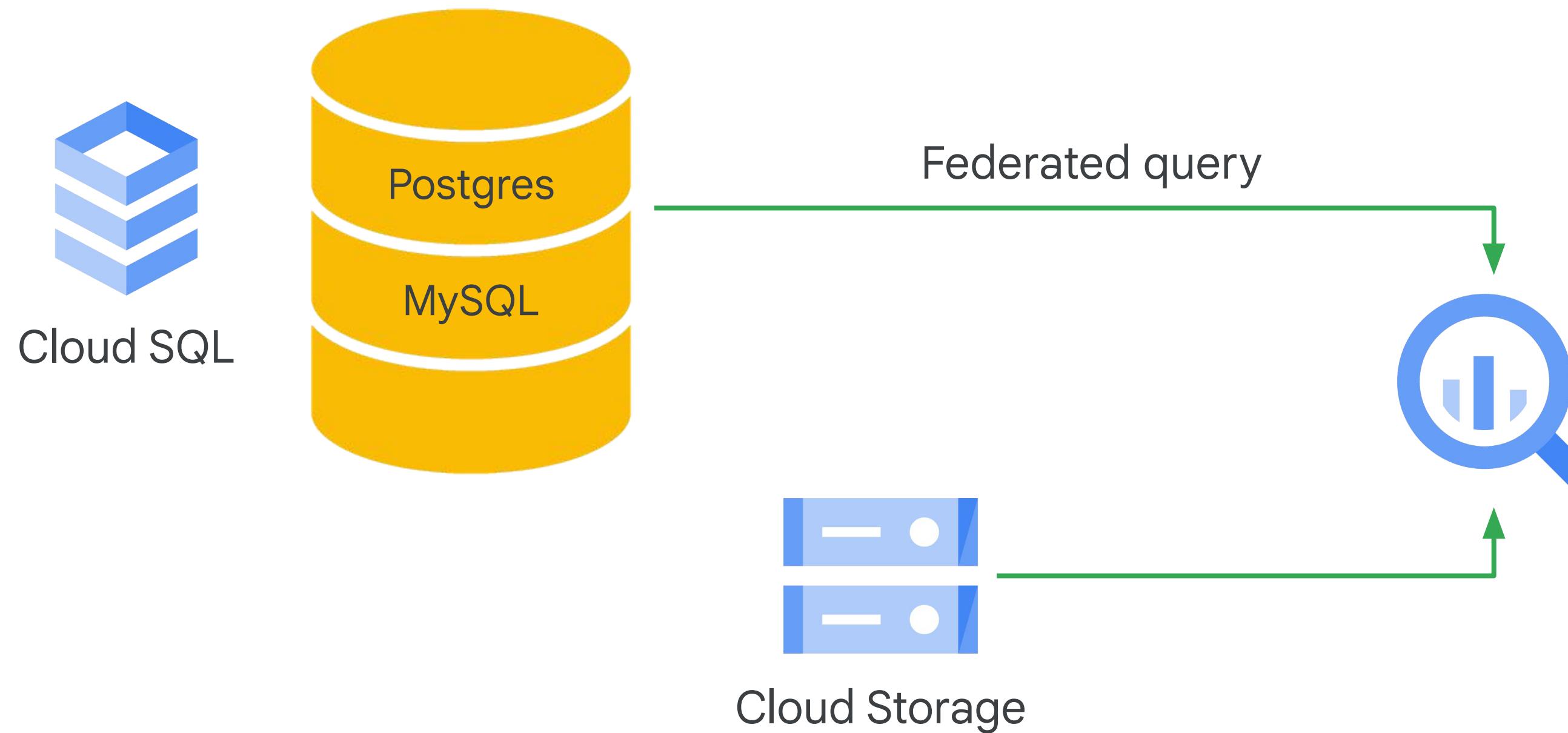
- Can it serve as a sink for both batch and streaming data pipelines?
- Can the data warehouse scale to meet my needs?
- How is the data organized, cataloged, and access controlled?
- Is the warehouse designed for performance?
- What level of maintenance is required by our engineering team?



# BigQuery is a modern data warehouse that changes the conventional mode of data warehousing

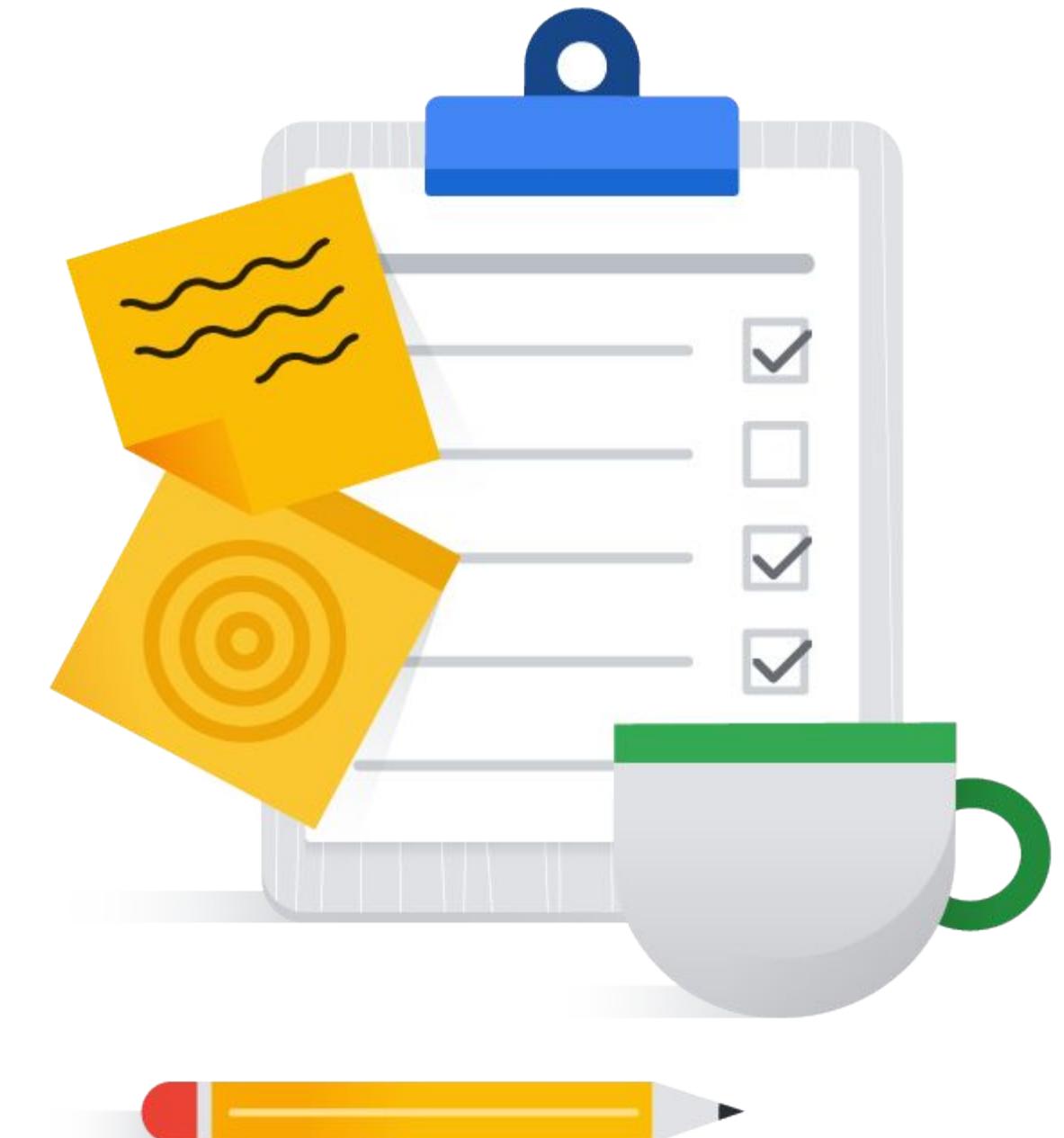


# You can simplify Data Warehouse ETL pipelines with external connections to Cloud Storage and Cloud SQL



# Introduction to Data Engineering

- 01 The role of a data engineer
- 02 Data engineering challenges
- 03 Introduction to BigQuery
- 04 Data lakes and data warehouses
- 05 **Transactional databases versus data warehouses**
- 06 Partner effectively with other data teams
- 07 Manage data access and governance
- 08 Build production-ready pipelines
- 09 Google Cloud customer case study



# Cloud SQL is fully managed SQL Server, Postgres, or MySQL for your Relational Database (transactional RDBMS)



Why not simply use Cloud SQL  
for reporting workflows?

- Automatic encryption
- 64 TB storage capacity
- 60,000 IOPS (read/write per second)
- Auto-scale and auto backup

# RDBMS are optimized for data from a single source and high-throughput writes versus high-read data warehouses



Cloud SQL \*

- Scales to GB and TB.
- Ideal for back-end database applications.
- Record-based storage.



BigQuery

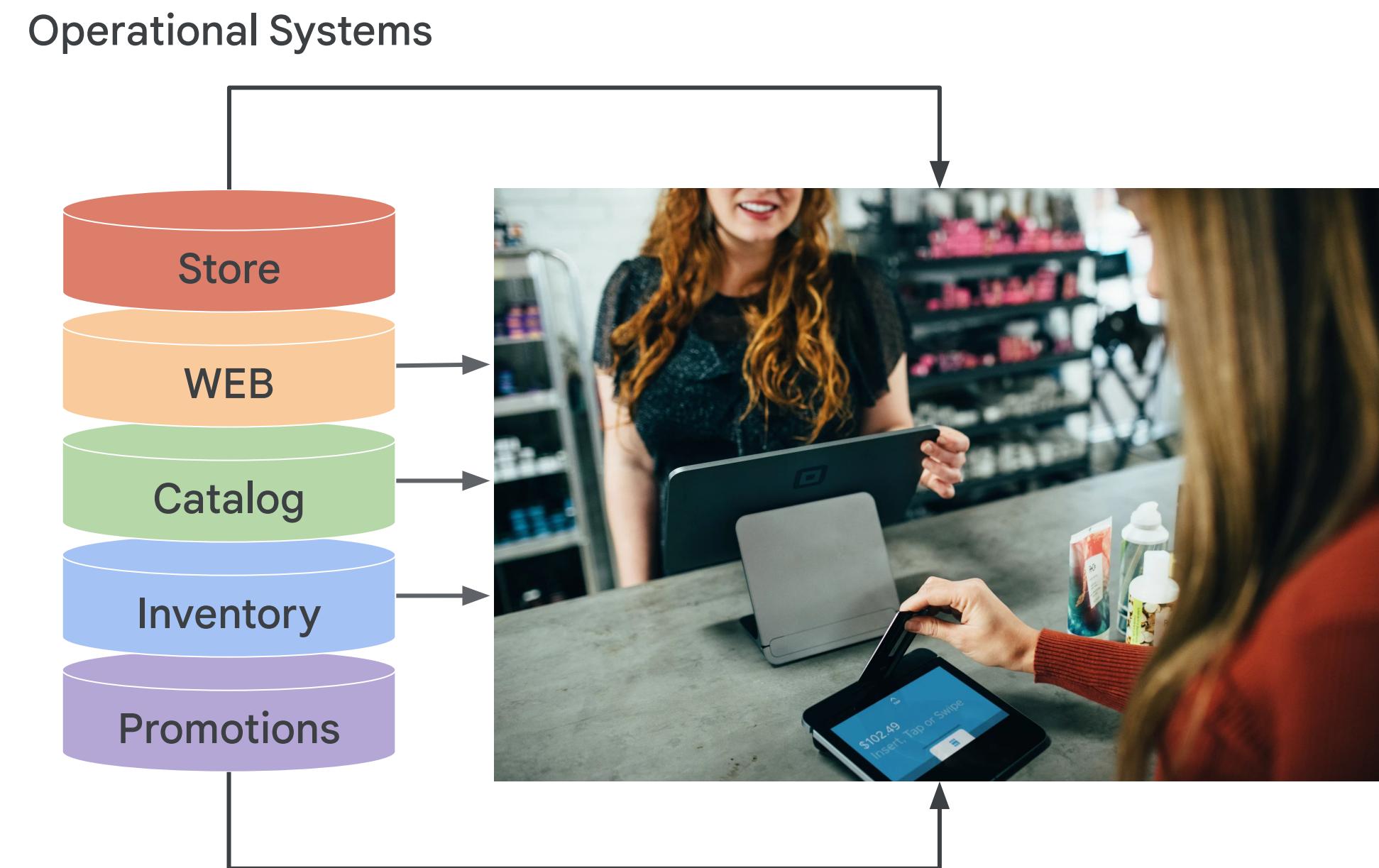
- Scales to PB.
- Easily connect to external data sources for ingestion.
- Column-based storage.

You will likely need and encounter both a database and data warehouse in your final architecture.

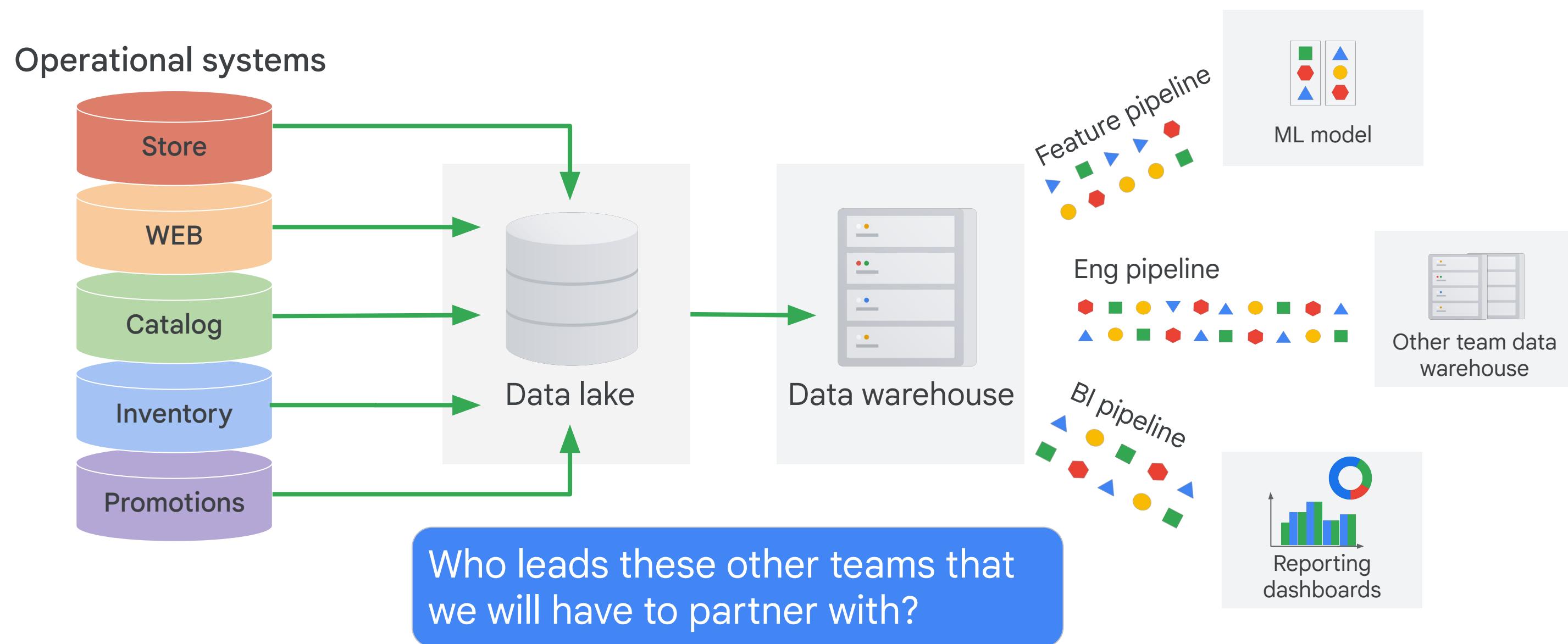
\* Cloud SQL is one of several RDBMS options on Google Cloud

# Relational database management systems (RDBMS) are critical for managing new transactions

RDBMS are optimized for high throughput **WRITES** to **RECORDS**.



# The complete picture: Source data comes into the data lake, is processed into the data warehouse and made available for insights



# Introduction to Data Engineering

- 01 The role of a data engineer
- 02 Data engineering challenges
- 03 Introduction to BigQuery
- 04 Data lakes and data warehouses
- 05 Transactional databases versus data warehouses
- 06 Partner effectively with other data teams
- 07 Manage data access and governance
- 08 Build production-ready pipelines
- 09 Google Cloud customer case study



# A data engineer builds data pipelines to enable data-driven decisions

Get the data to where it can be useful

Get the data into a usable condition

Add new value to the data

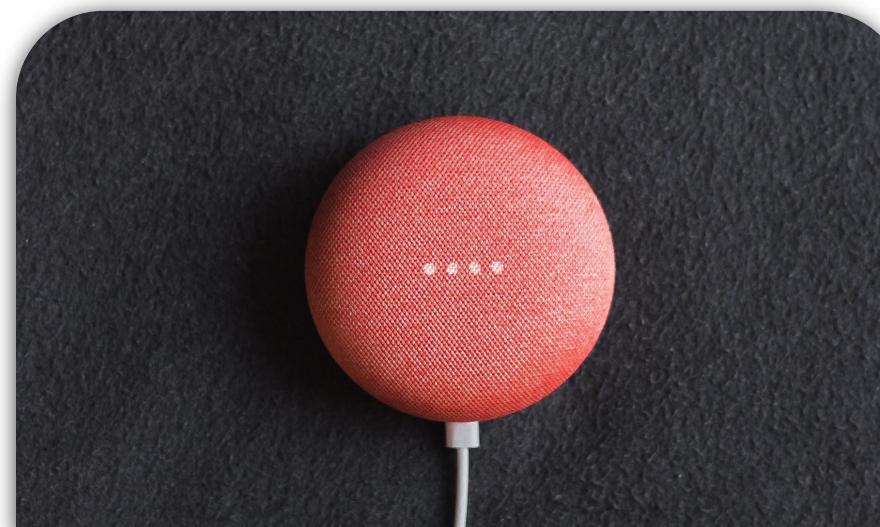
Manage the data

Productionize data processes

What teams rely on these pipelines?

# Many teams rely on partnerships with data engineering to get value out of their data

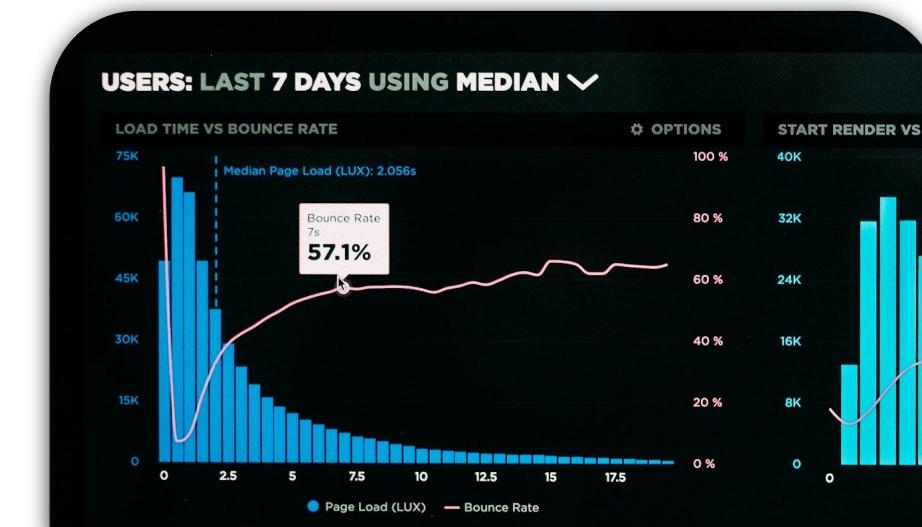
How might each of these teams rely on data engineering?



ML Engineer

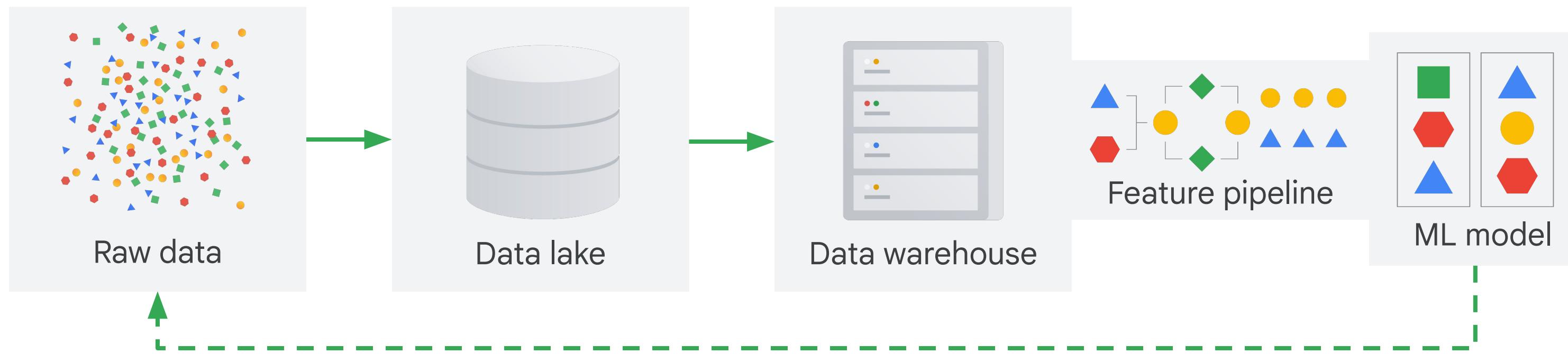


Data Analyst

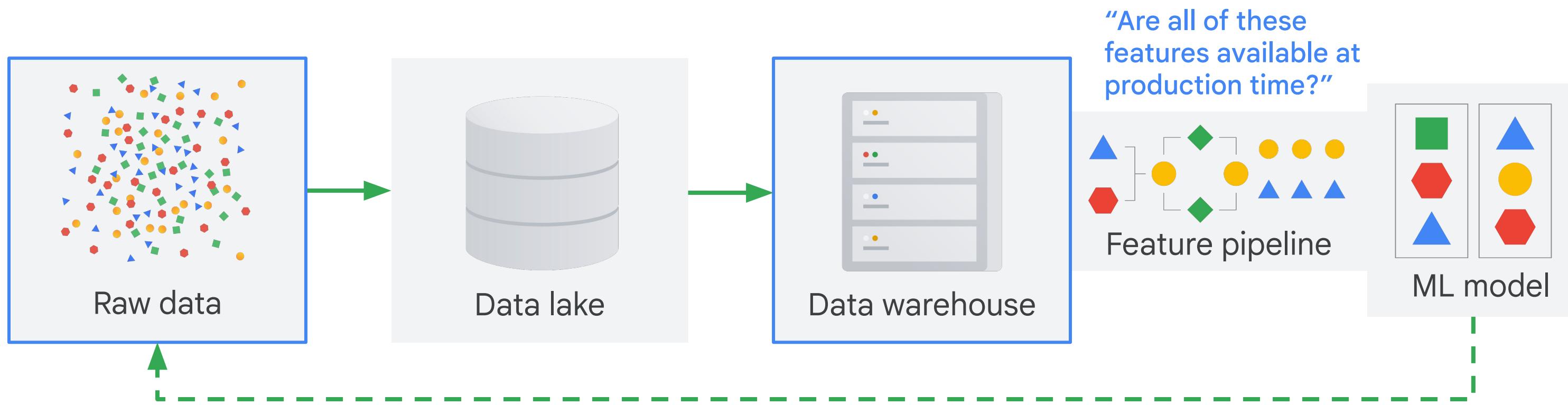


Data Engineer

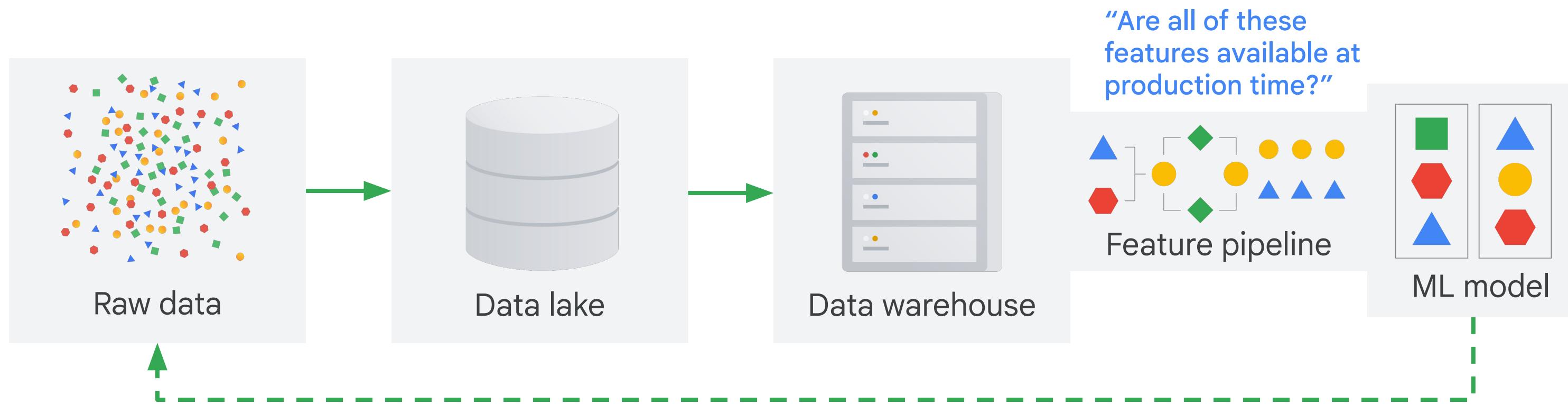
# Machine learning teams need data engineers to help them capture new features in a stable pipeline



# Machine learning teams need data engineers to help them capture new features in a stable pipeline

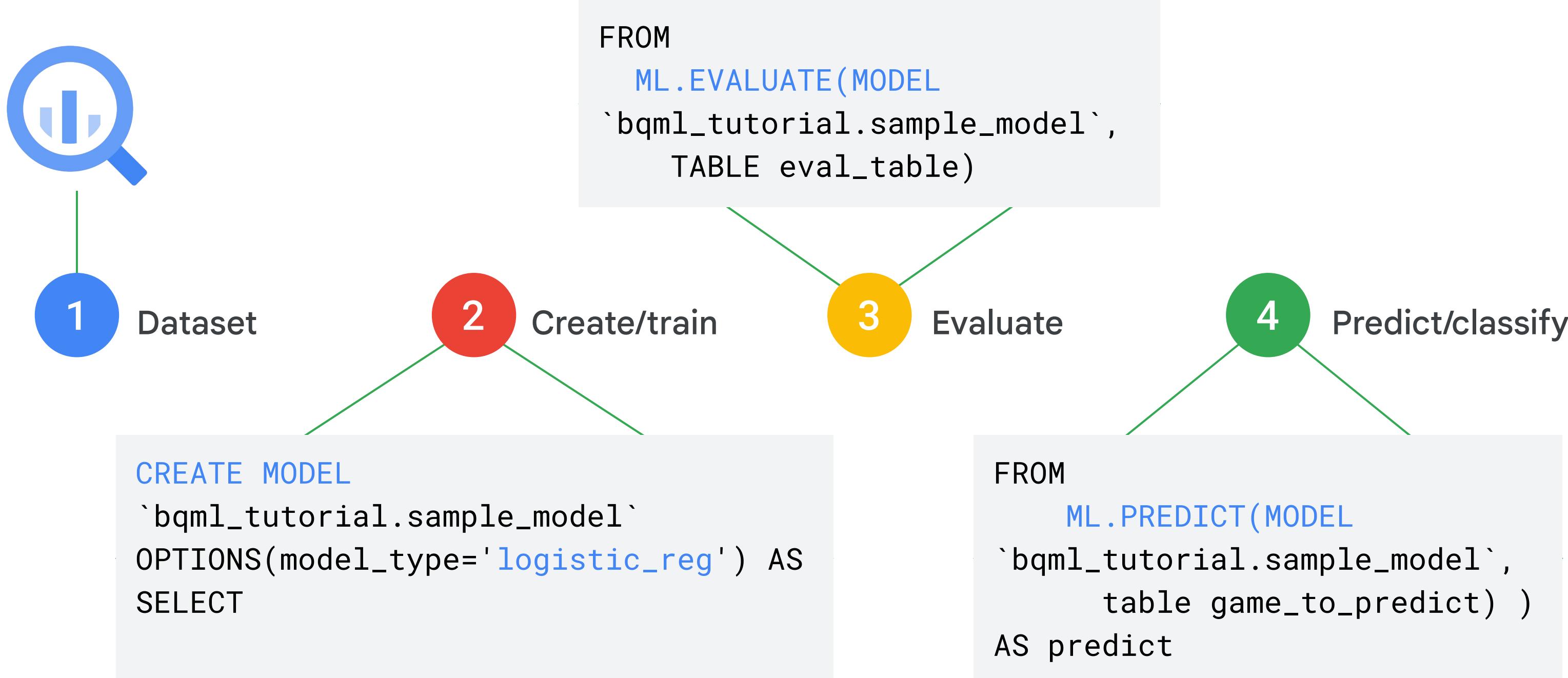


# Machine learning teams need data engineers to help them capture new features in a stable pipeline

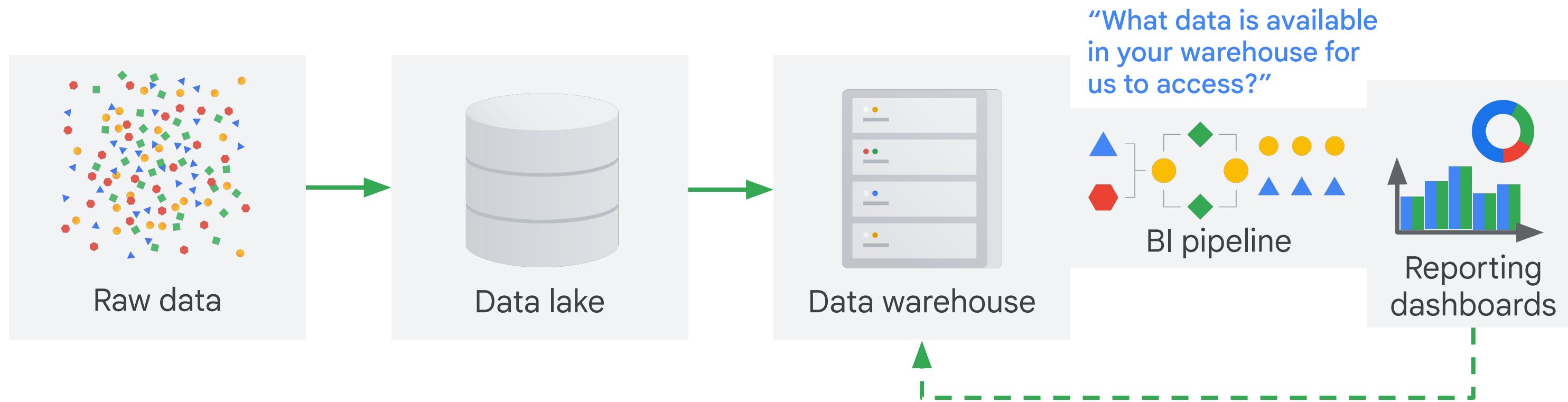


"Can you help us get more features (columns)  
of data for our machine learning model?"

# Add value: Machine learning directly in BigQuery



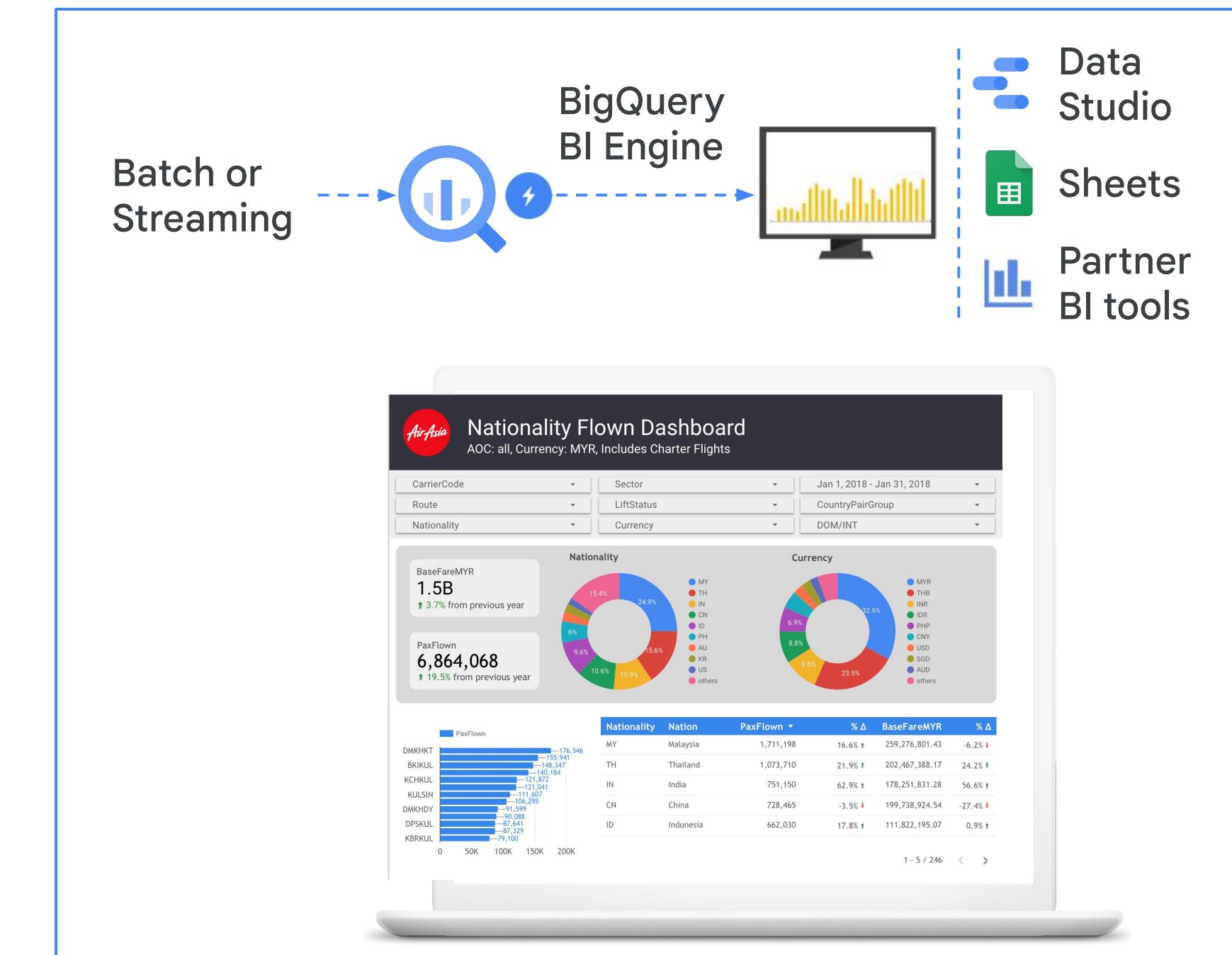
# Data analysis and business intelligence teams rely on data engineering to showcase the latest insights



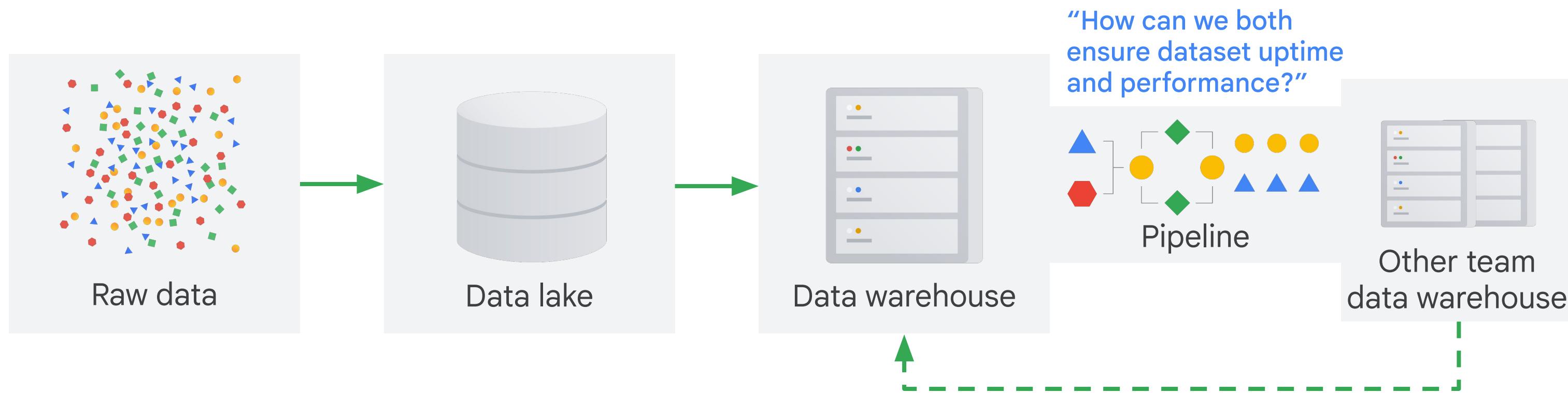
"Our dashboards are slow, can you help us re-engineer our BI tables for better performance?"

# Add value: BI Engine for dashboard performance

- No need to manage OLAP cubes or separate BI servers for dashboard performance.
- Natively integrates with BigQuery streaming for real-time data refresh.
- Column oriented in-memory BI execution engine.

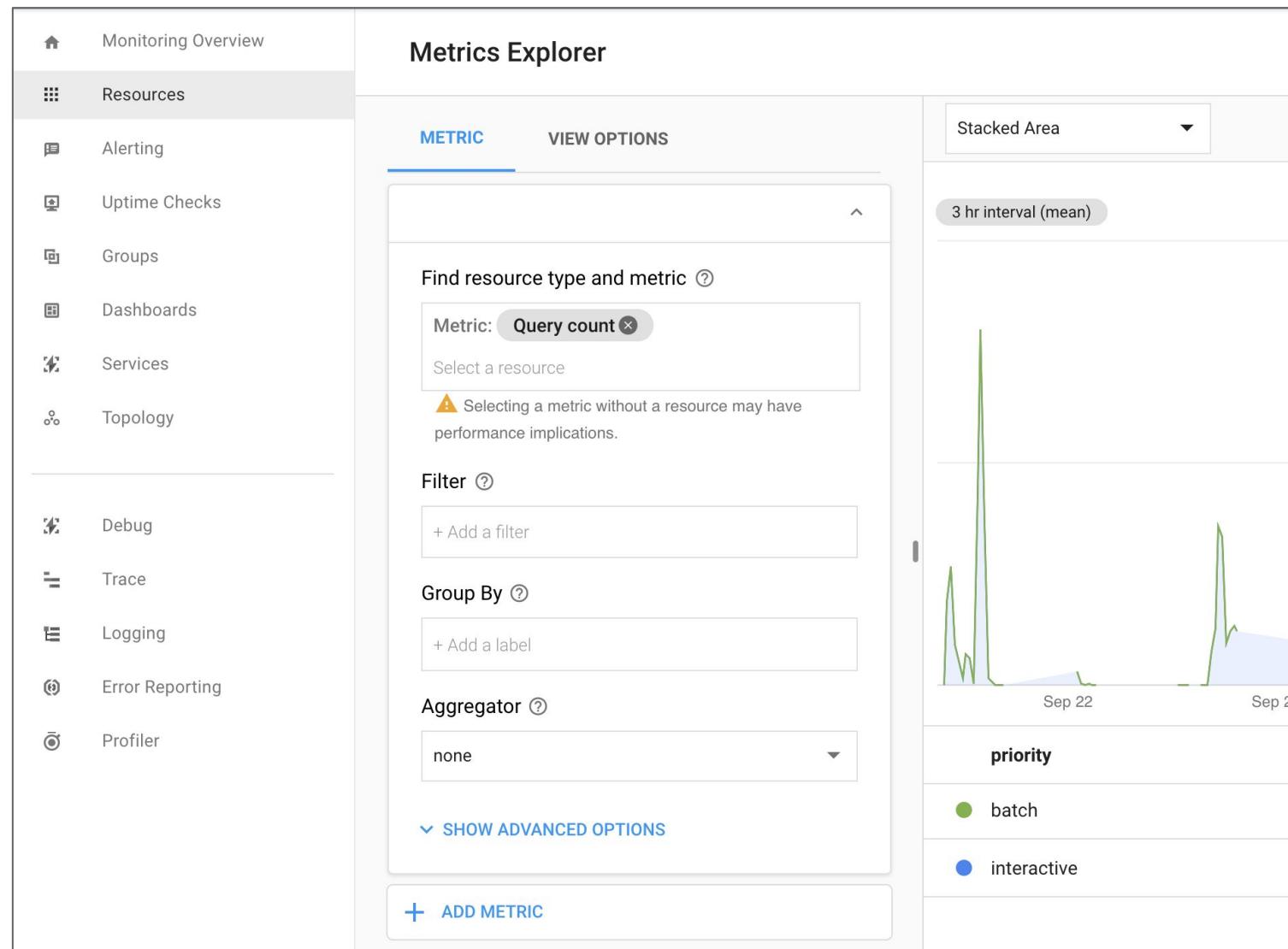


# Other data engineering teams may rely on your pipelines being timely and error free



"We're noticing high demand for your datasets -- be sure your warehouse can scale for many users"

# Add value: Cloud Monitoring for performance



- View in-flight and completed queries.
- Create alerts and send notifications.
- Track spending on BigQuery resources.
- Use [Cloud Audit Logs](#) to view actual job information (who executed, what query was ran).

# Introduction to Data Engineering

- 01 The role of a data engineer
- 02 Data engineering challenges
- 03 Introduction to BigQuery
- 04 Data lakes and data warehouses
- 05 Transactional databases versus data warehouses
- 06 Partner effectively with other data teams
- 07 **Manage data access and governance**
- 08 Build production-ready pipelines
- 09 Google Cloud customer case study



# A data engineer manages data access and governance

Get the data to where it can be useful

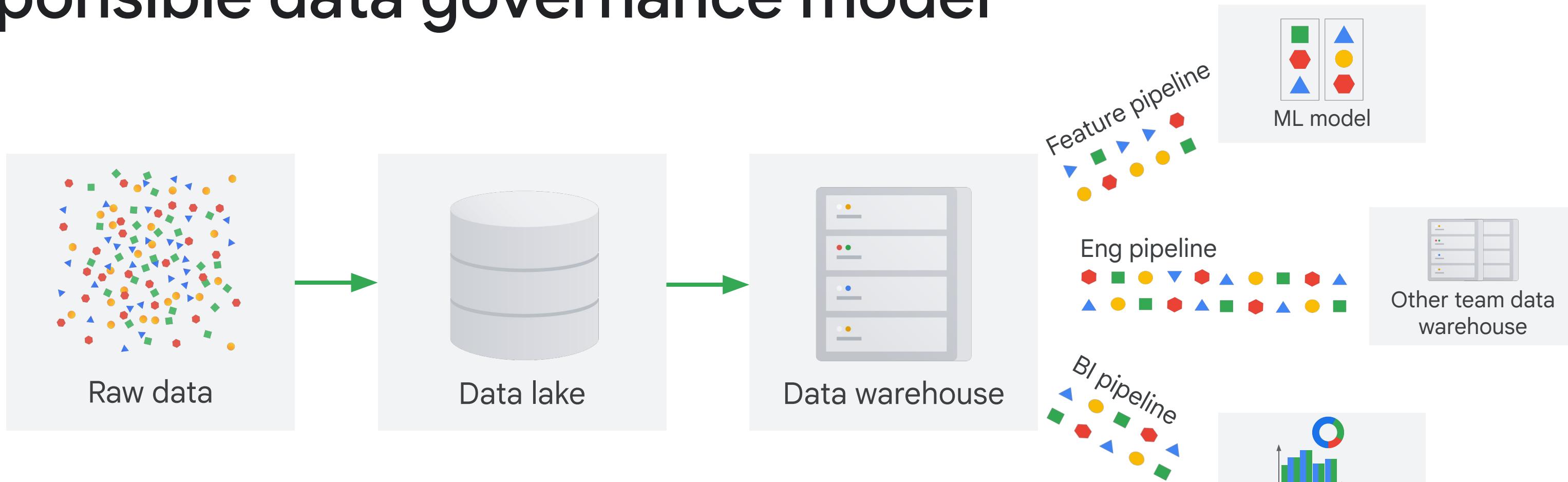
Get the data into a usable condition

Add new value to the data

Manage the data

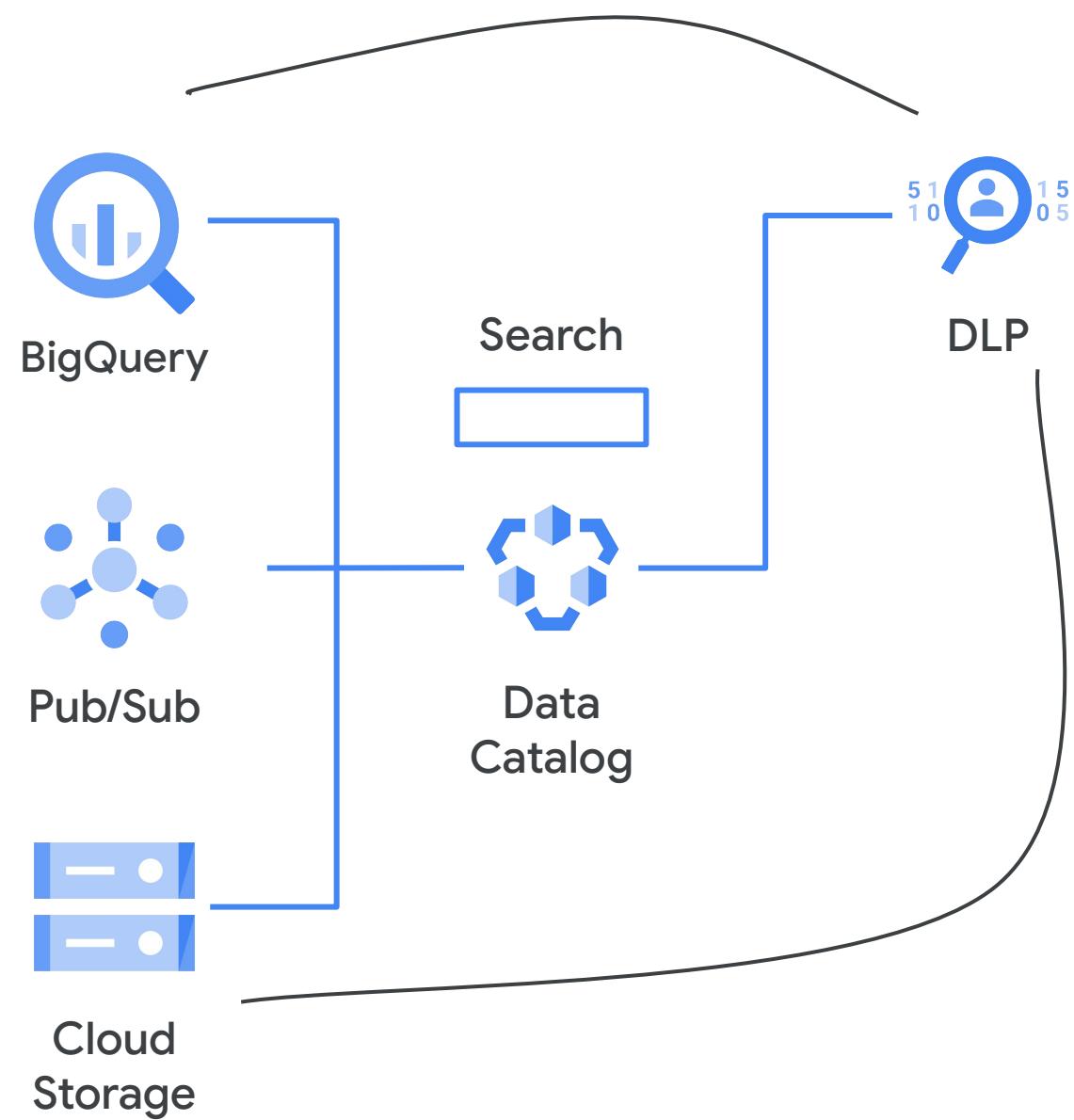
Productionize data processes

# Data engineering must set and communicate a responsible data governance model



- ✓ Who should have access?
- ✓ How is PII handled?
- ✓ How can we educate end-users on our data catalog?

# Cloud Data Catalog is a managed data discovery + Data Loss Prevention API for guarding PII



## Data Catalog

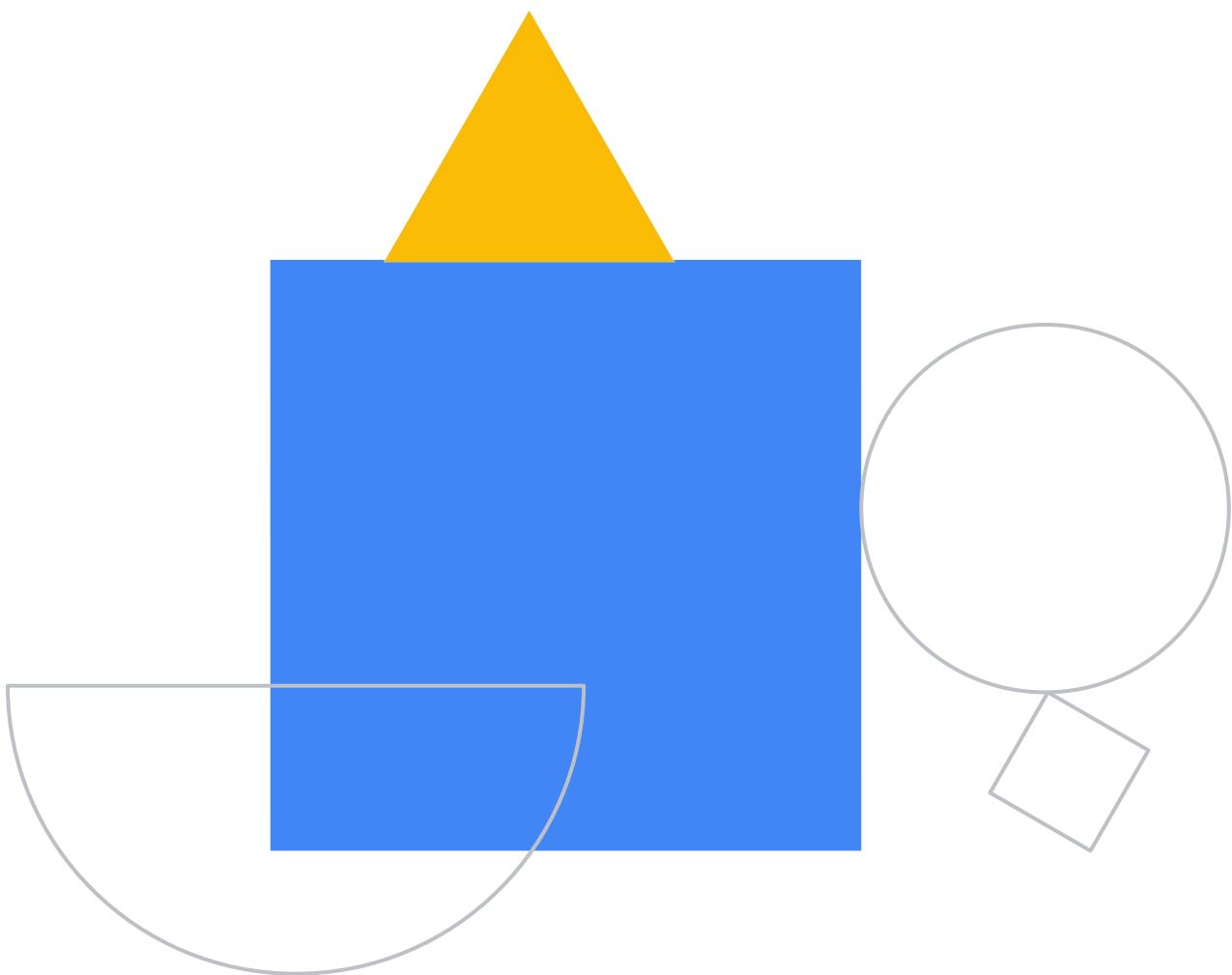
**Simplify data discovery at any scale:**  
Fully managed metadata management service with no infrastructure to set up or manage.

**Unified view of all datasets:**  
Central and secure data catalog across Google Cloud with metadata capture and tagging.

**Data governance foundation:**  
Security compliance with access level controls along with Cloud Data Loss Prevention integration for handling sensitive data.

# Demo

Finding PII in your dataset with  
the DLP API



# Introduction to Data Engineering

- 01 The role of a data engineer
- 02 Data engineering challenges
- 03 Introduction to BigQuery
- 04 Data lakes and data warehouses
- 05 Transactional databases versus data warehouses
- 06 Partner effectively with other data teams
- 07 Manage data access and governance
- 08 **Build production-ready pipelines**
- 09 Google Cloud customer case study



# A data engineer builds production data pipelines to enable data-driven decisions

Get the data to where it can be useful

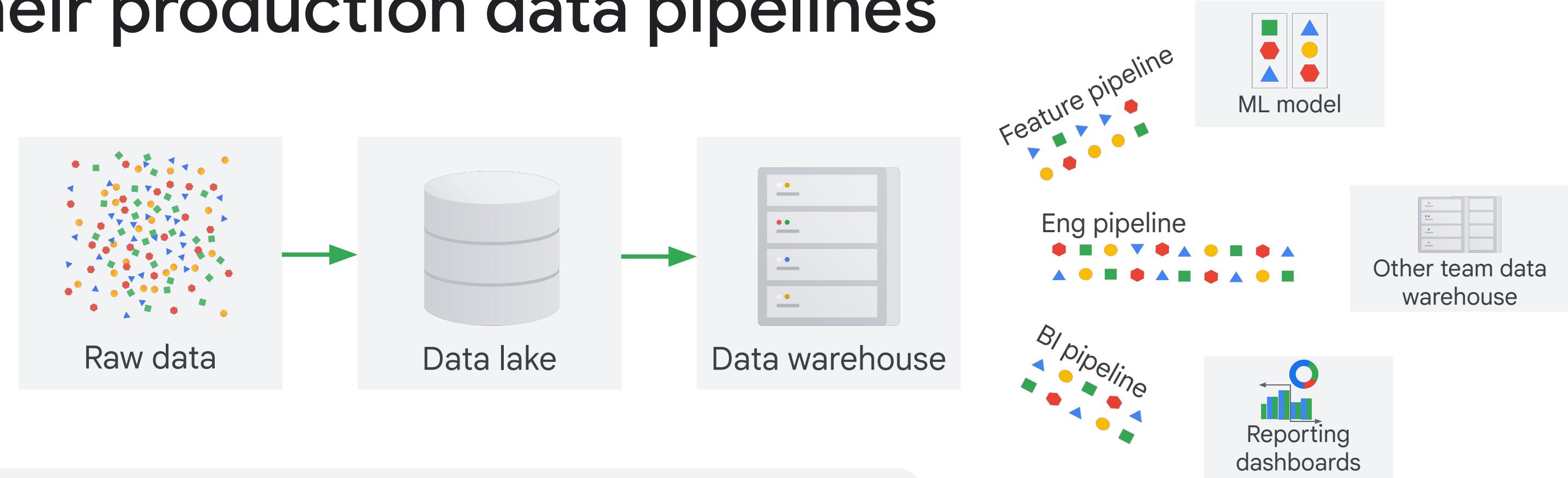
Get the data into a usable condition

Add new value to the data

Manage the data

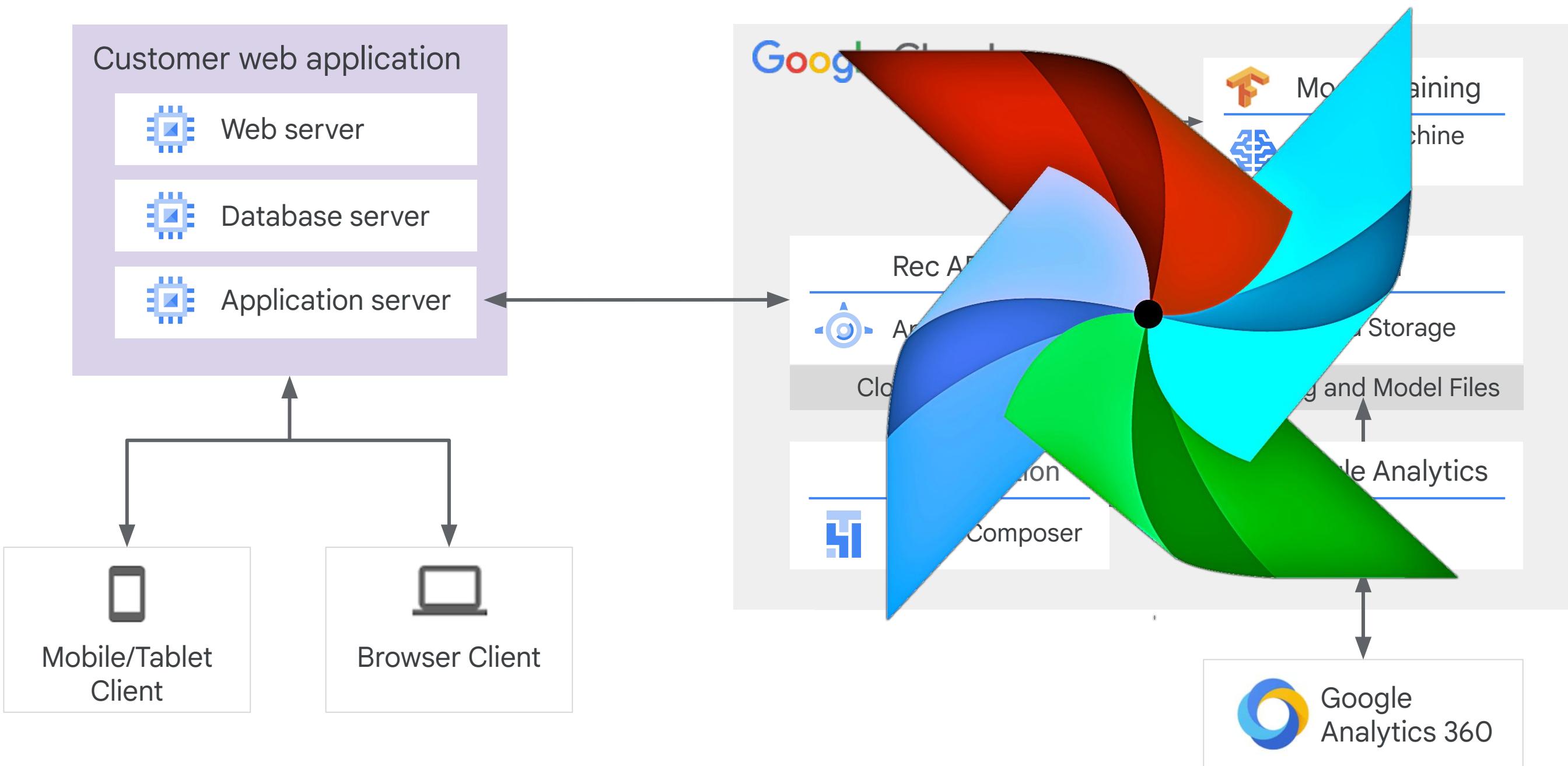
Productionize data processes

# Data engineering owns the health and future of their production data pipelines

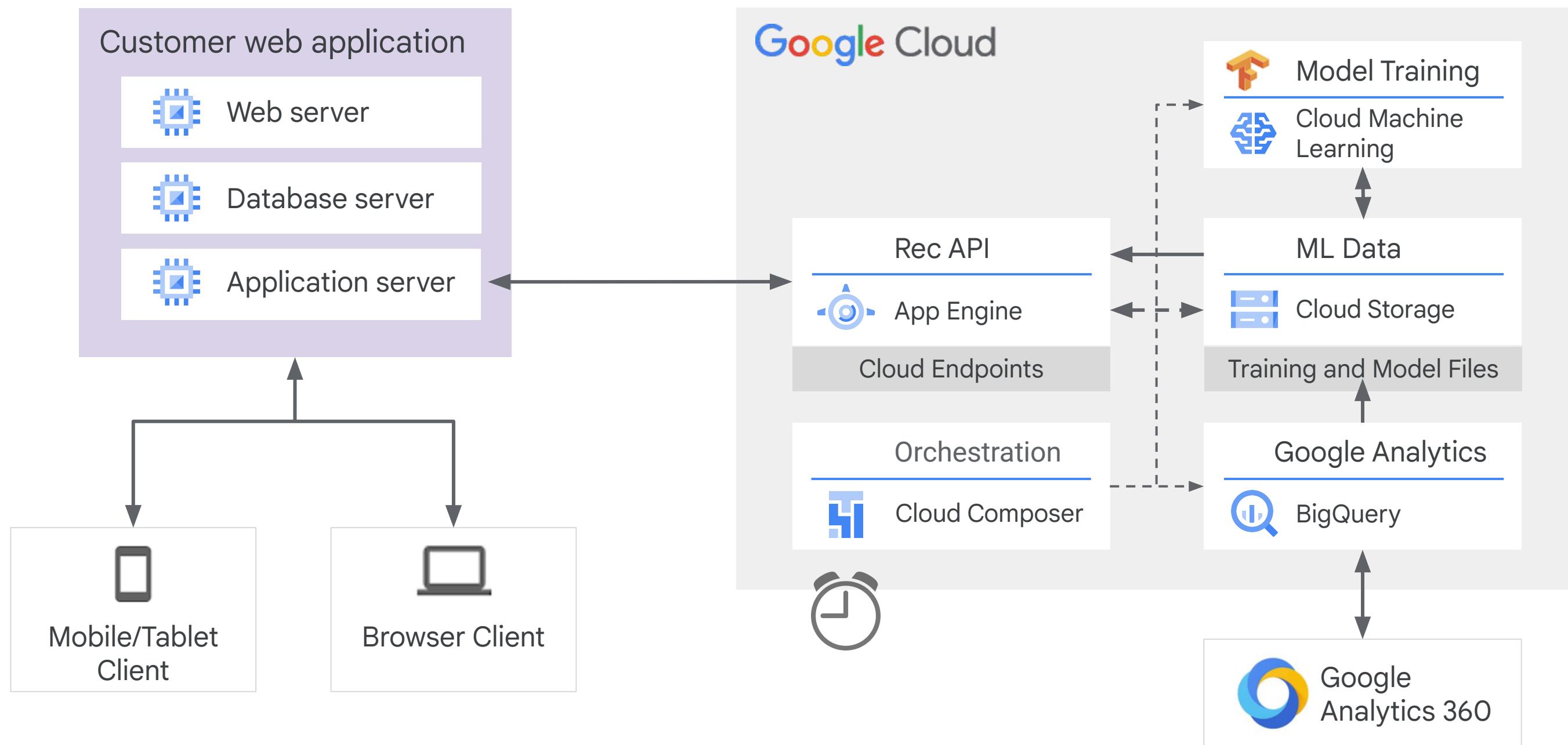


- ✓ How can we ensure pipeline health and data cleanliness?
- ✓ How do we productionalize these pipelines to minimize maintenance and maximize uptime?
- ✓ How do we respond and adapt to changing schemas and business needs?
- ✓ Are we using the latest data engineering tools and best practices?

**Cloud Composer (managed Apache Airflow) is used to orchestrate production workflows**



# Cloud Composer (managed Apache Airflow) is used to orchestrate production workflows

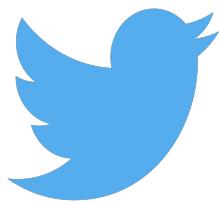


# Introduction to Data Engineering

- 01 The role of a data engineer
- 02 Data engineering challenges
- 03 Introduction to BigQuery
- 04 Data lakes and data warehouses
- 05 Transactional databases versus data warehouses
- 06 Partner effectively with other data teams
- 07 Manage data access and governance
- 08 Build production-ready pipelines
- 09 [Google Cloud customer case study](#)

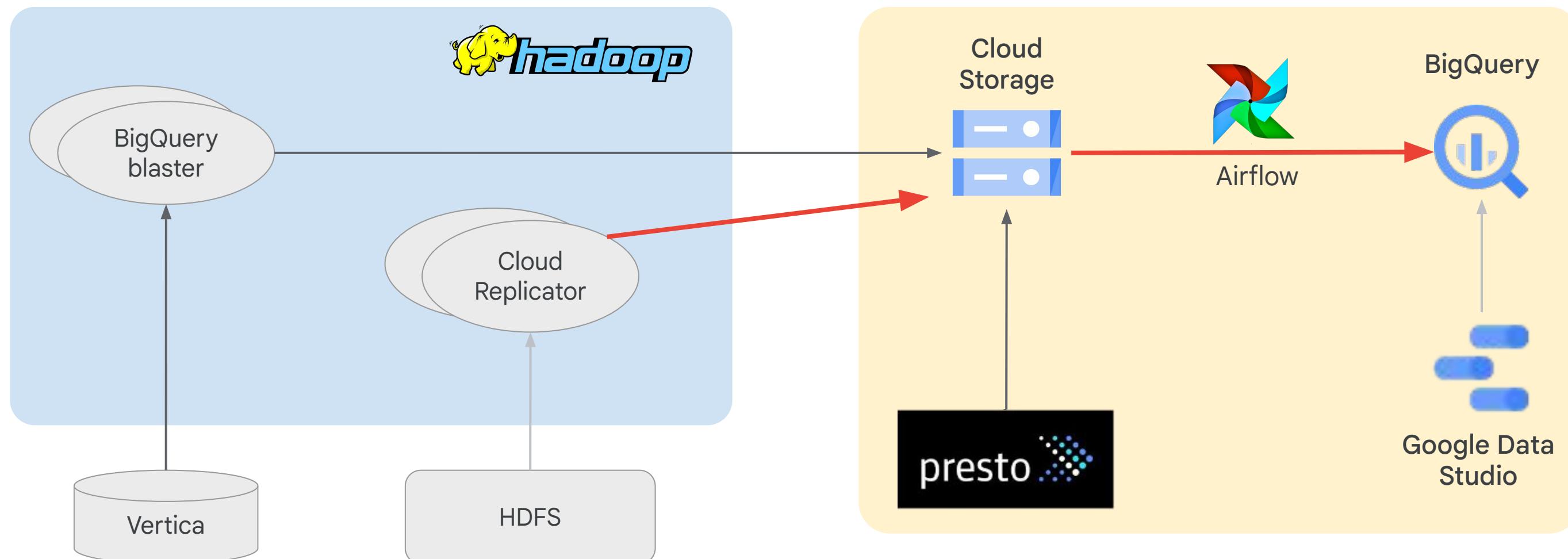


# Twitter democratized data analysis using BigQuery



"We believe that users with a wide range of technical skills should be able to discover data and have access to SQL-based analysis and visualization tools that perform well"

-- Twitter

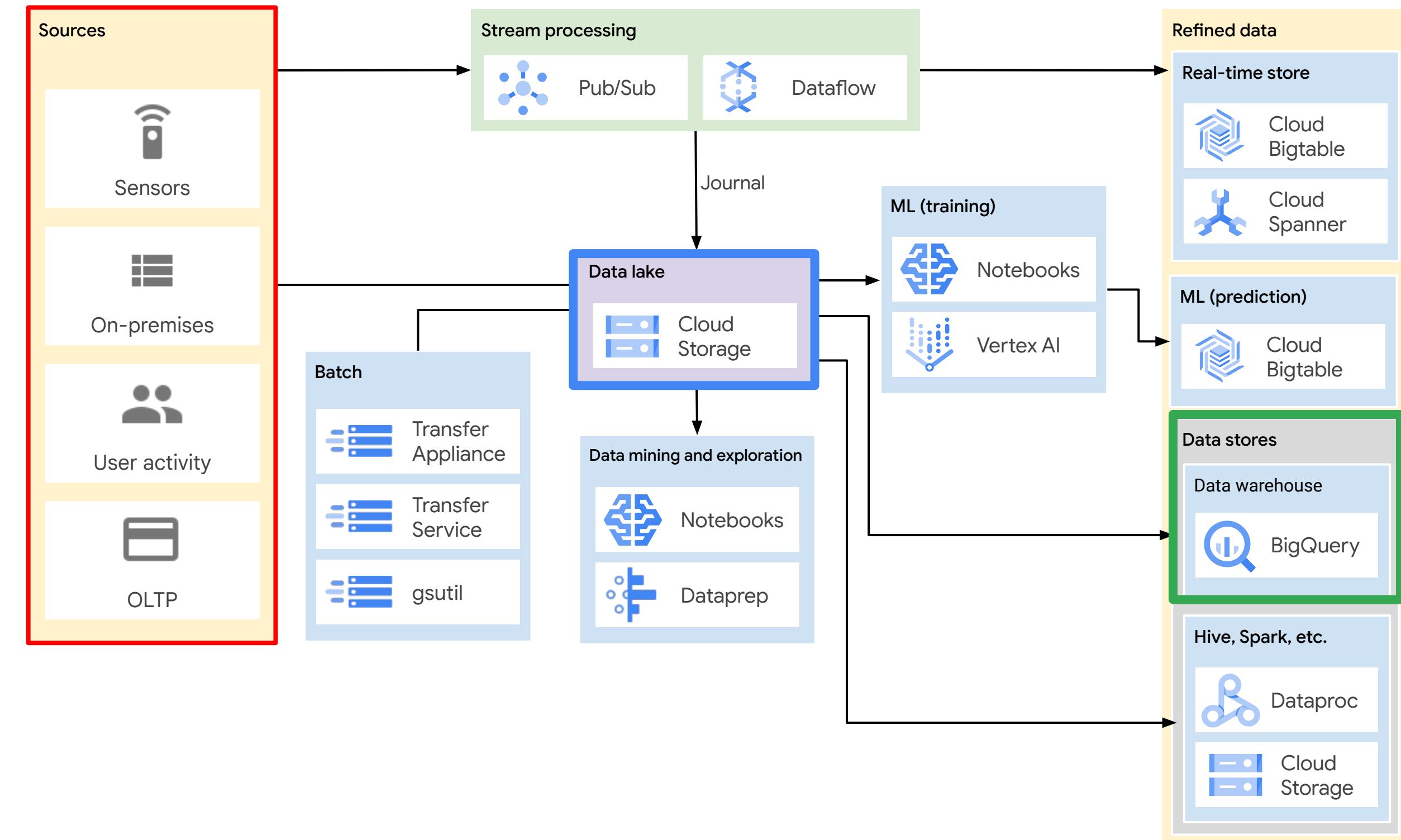


# Recap

- Data sources
- Data lakes
- Data warehouses
- Google Cloud solutions for Data Engineering

# Concept review

Data sources feed into a Data Lake and are processed into your Data Warehouse for analysis.



# Here's a useful guide for “Google Cloud products in 4 words or less”

## The Google Cloud Developer's Cheat Sheet

Updated continually by Google Developer Relations

### DATABASES

Cloud Bigtable	Petabyte-scale, low-latency, non-relational
Cloud Datastore	Horizontally scalable document DB
Cloud Firestore	Strongly-consistent serverless document DB
Cloud Memorystore	Managed Redis
Cloud Spanner	Horizontally scalable relational DB
Cloud SQL	Managed MySQL and PostgreSQL

### DATA AND ANALYTICS

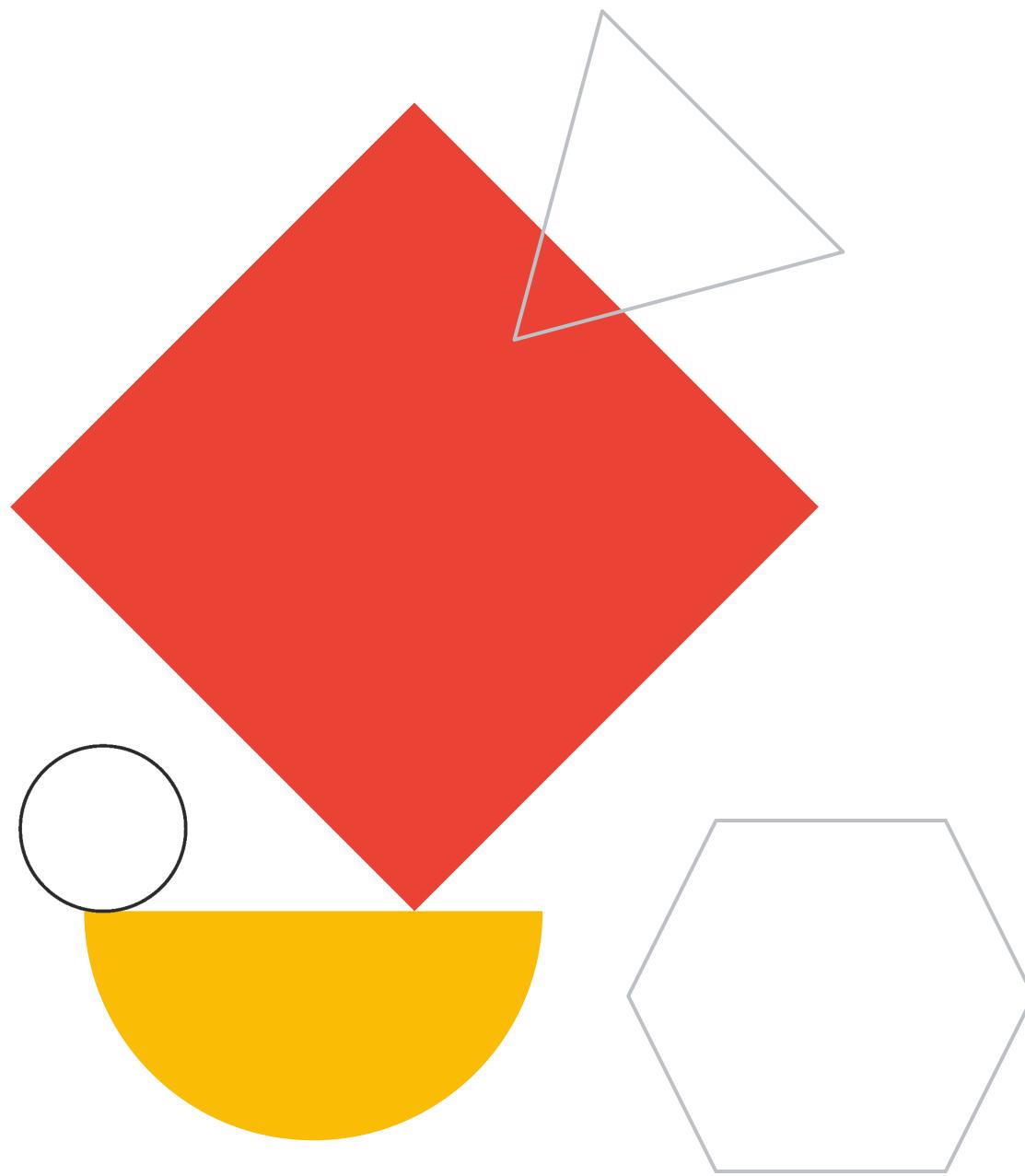
BigQuery	Data warehouse/analytics
BigQuery BI Engine	In-memory analytics engine
BigQuery ML	BigQuery model training/serving
Cloud Composer	Managed workflow orchestration service
Cloud Data Fusion	Graphically manage data pipelines
Cloud Dataflow	Stream/batch data processing
Cloud Datalab	Managed Jupyter notebook
Cloud Dataprep	Visual data wrangling
Cloud Dataproc	Managed Spark and Hadoop
Cloud Pub/Sub	Global real-time messaging
Data Catalog	Metadata management service
Data Studio	Collaborative data exploration/dashboarding
Genomics	Managed genomics platform

### AI/ML

AI Hub	Hosted AI component sharing
AI Platform	Managed platform for ML
AI Platform Data Labeling	Data labeling by humans
AI Platform Deep Learning VMs	Preconfigured VMs for deep learning
AI Platform Notebooks	Managed JupyterLab notebook instances
AI Platform Training	Parallel and distributed training

# Lab Intro

Using BigQuery to do Analysis



# Lab objectives

01

Execute interactive queries in the BigQuery console

02

Combine and run analytics on multiple dataset



