Course 2: Production ML Systems

Module 1: Architecting Production ML Systems

Lesson Title: **Introduction**

Presenter: Max Lotstein

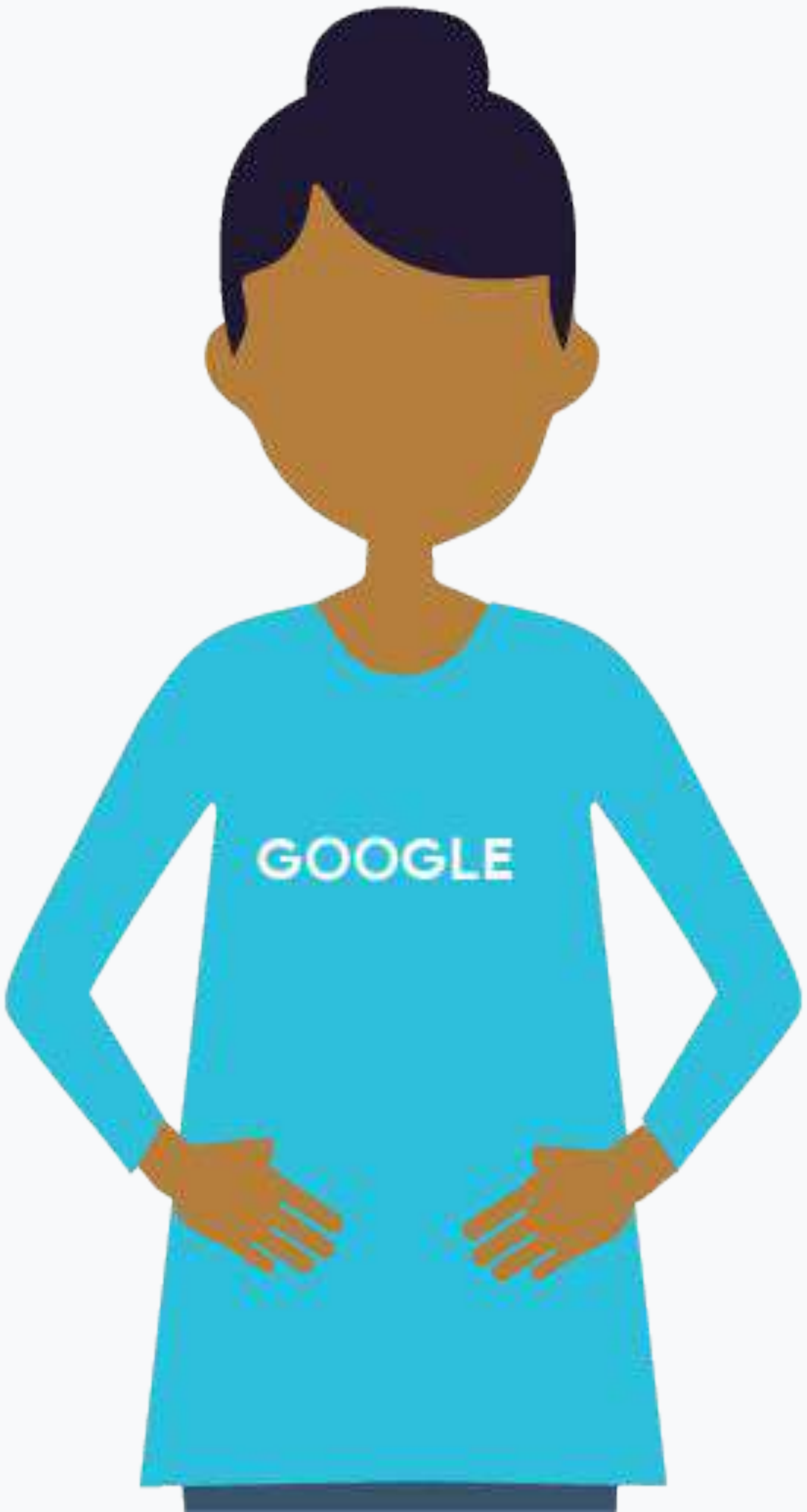
Format: Talking Head

Video Name: T-PSML-O_1_l1_introduction

Google Cloud

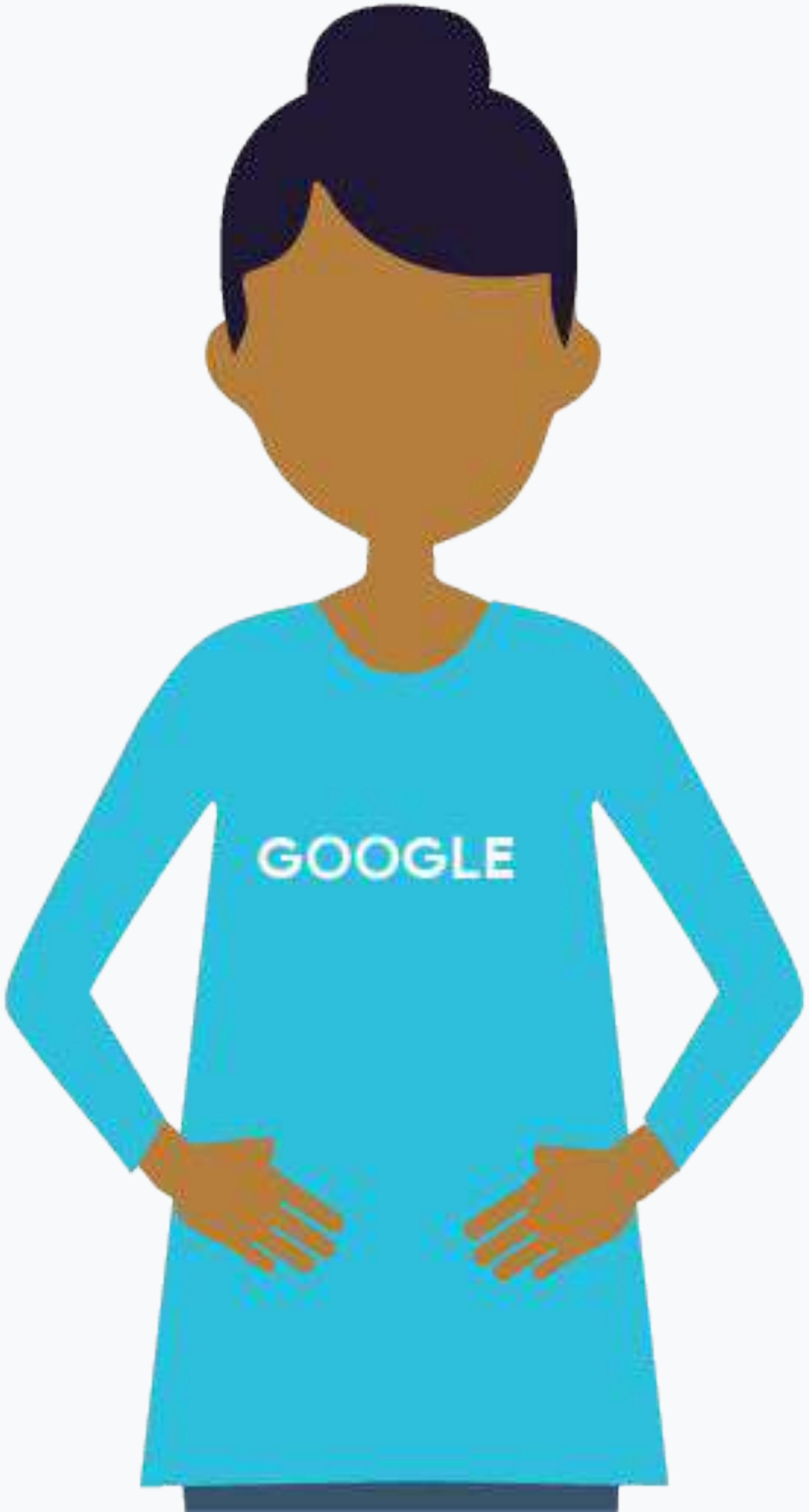
Architecting ML Systems

Max Lotstein



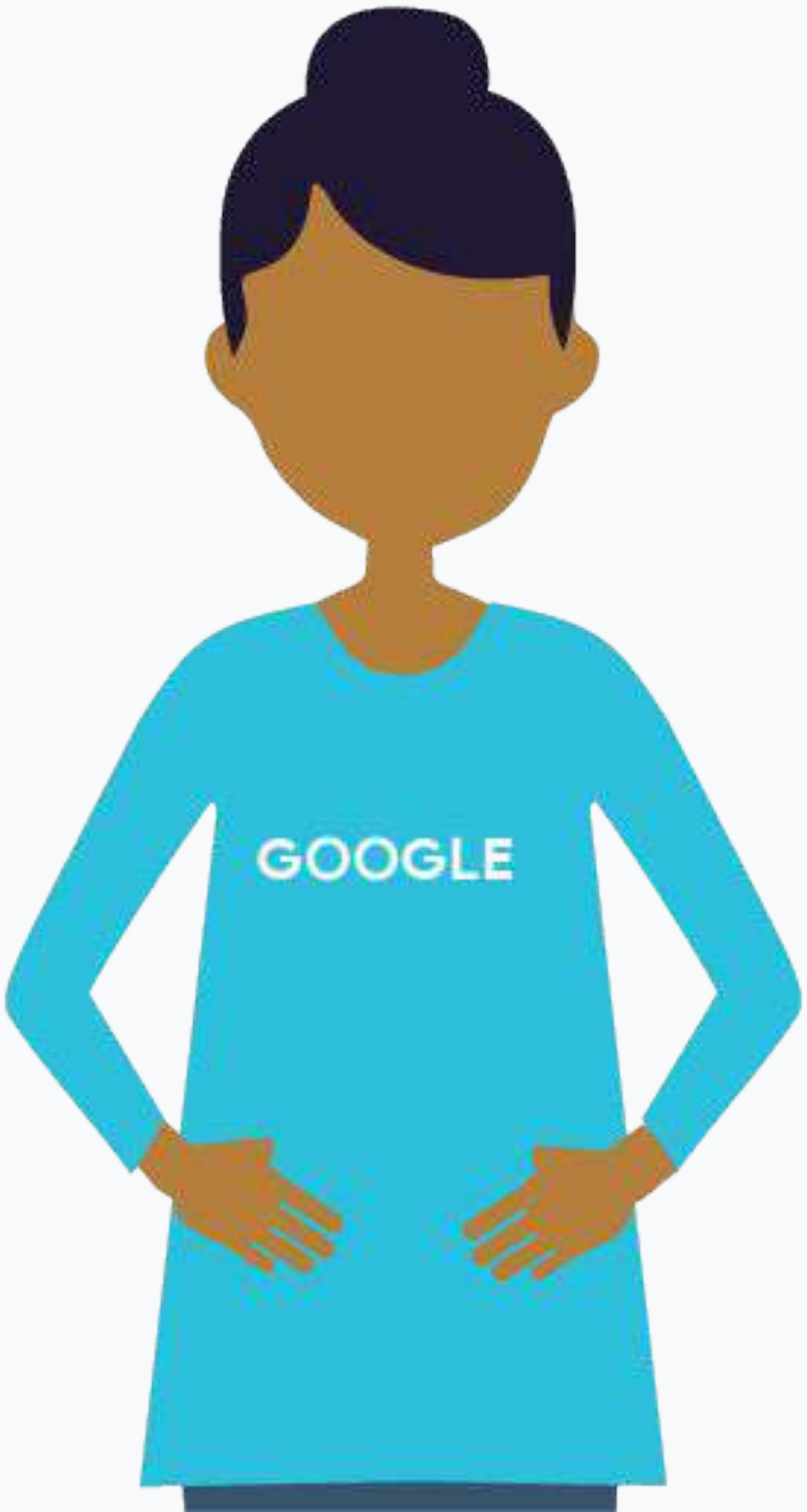
Quiz: What percent of system code does the ML model account for?

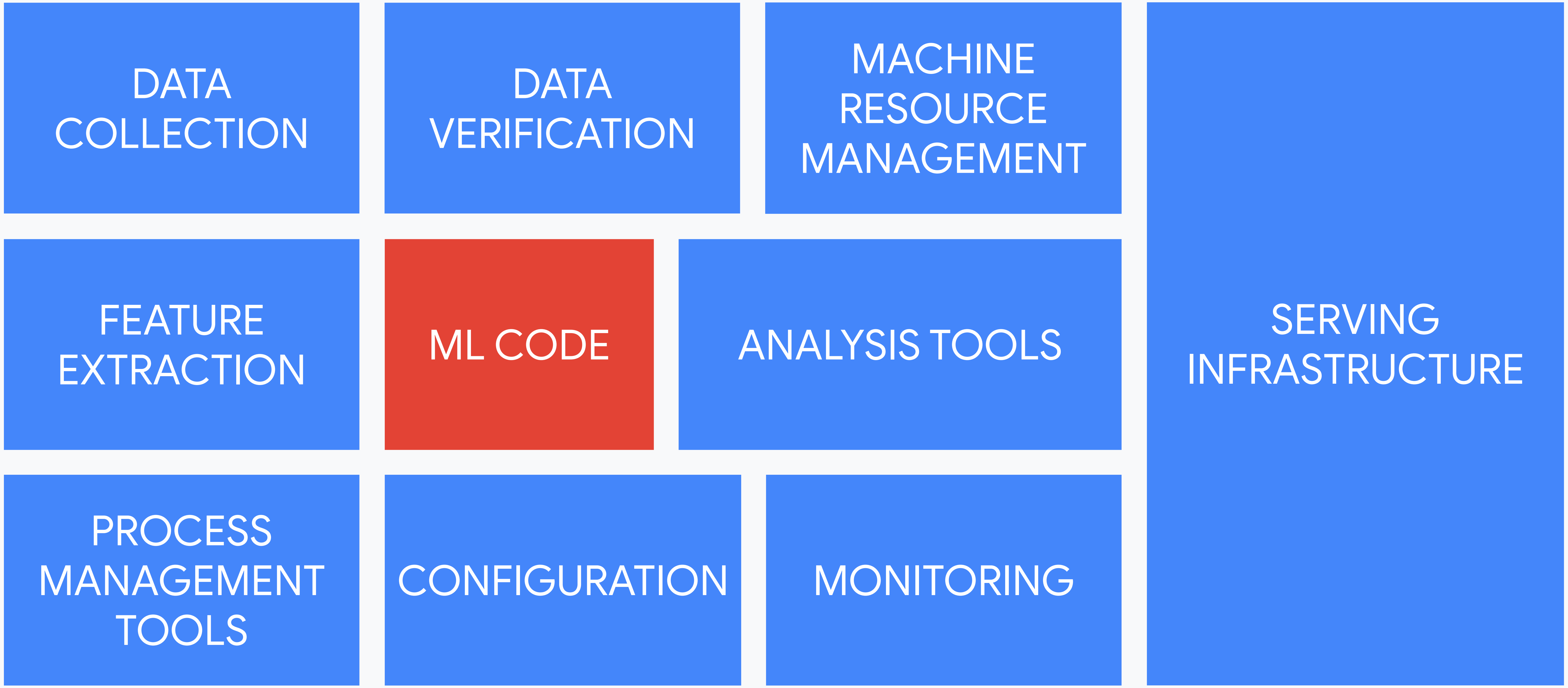
(a) 5%
(b) 25%
(c) 50%
(d) 90%



Quiz: What percent of system code does the ML model account for?

(a) 5%
(b) 25%
(c) 50%
(d) 90%




DATA COLLECTION
DATA VERIFICATION
MACHINE RESOURCE MANAGEMENT
SERVING INFRASTRUCTURE
FEATURE EXTRACTION
ML CODE
ANALYSIS TOOLS
PROCESS MANAGEMENT TOOLS
CONFIGURATION
MONITORING

Agenda

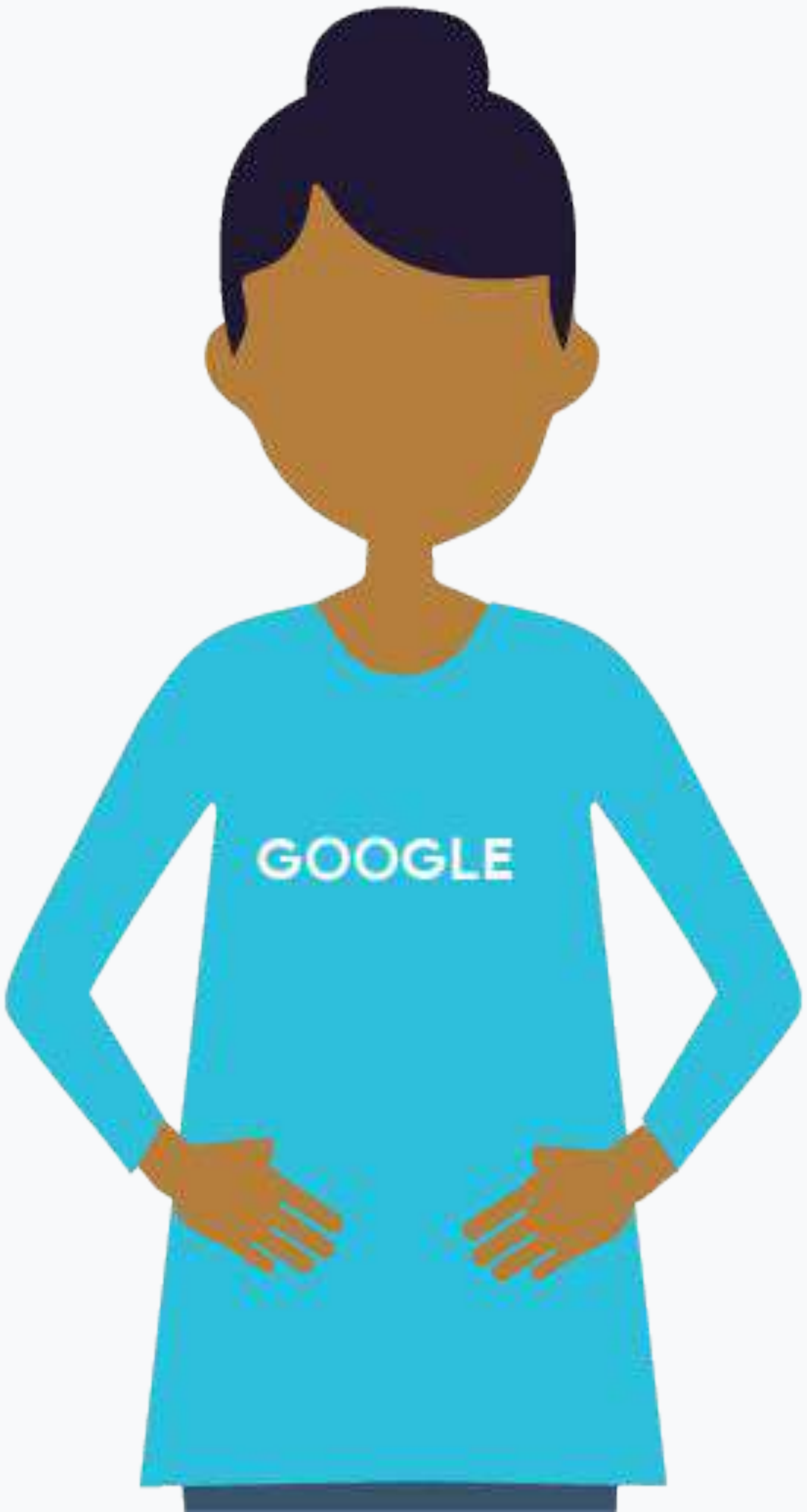
What's in a Production ML System

Training Design Decisions

Serving Design Decisions

Serving on CMLE

Designing an Architecture from Scratch



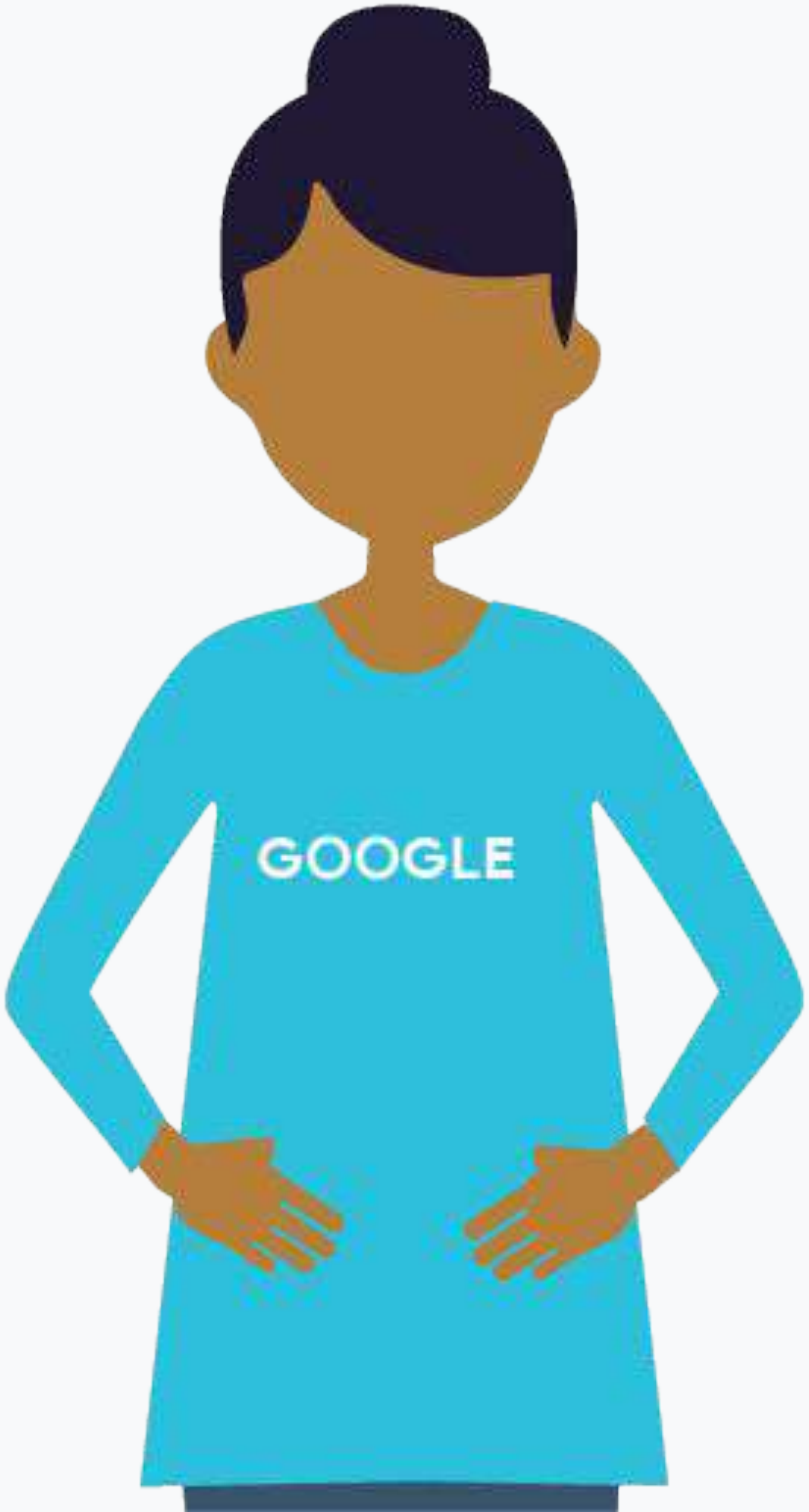
Learn how to...

Choose the appropriate training paradigm

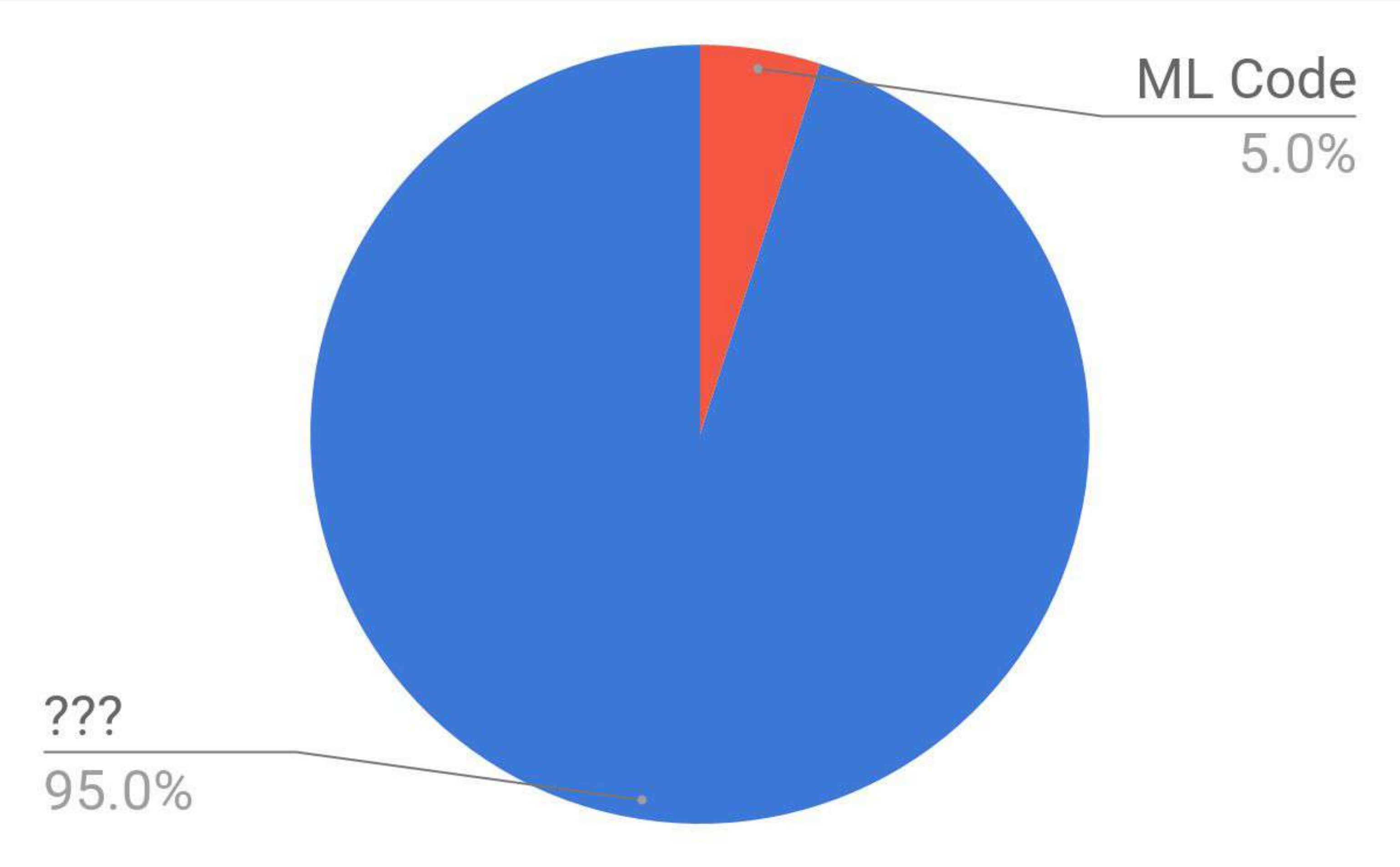
Choose the appropriate serving paradigm

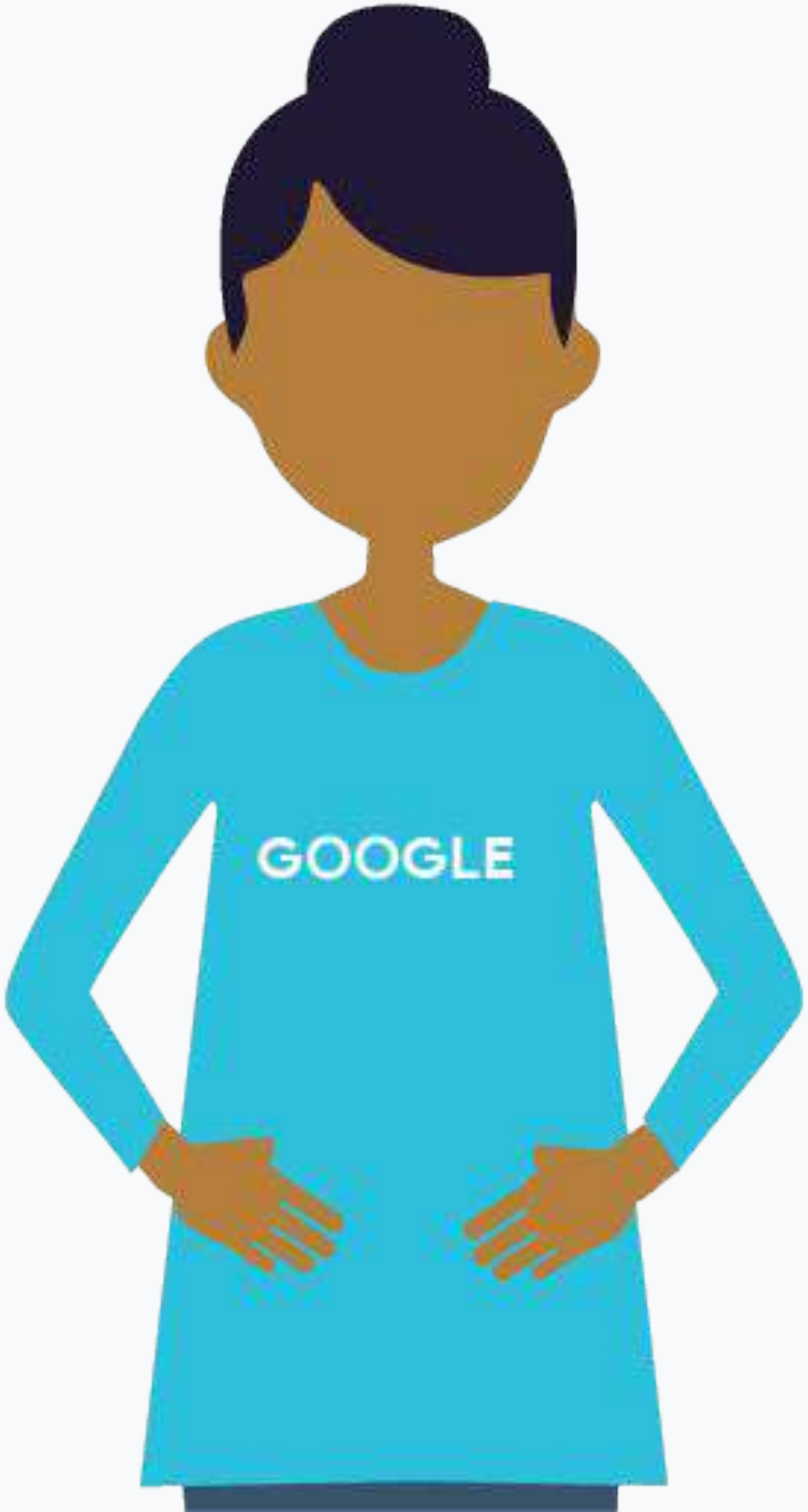
Serve ML models scalably

Design an architecture from scratch

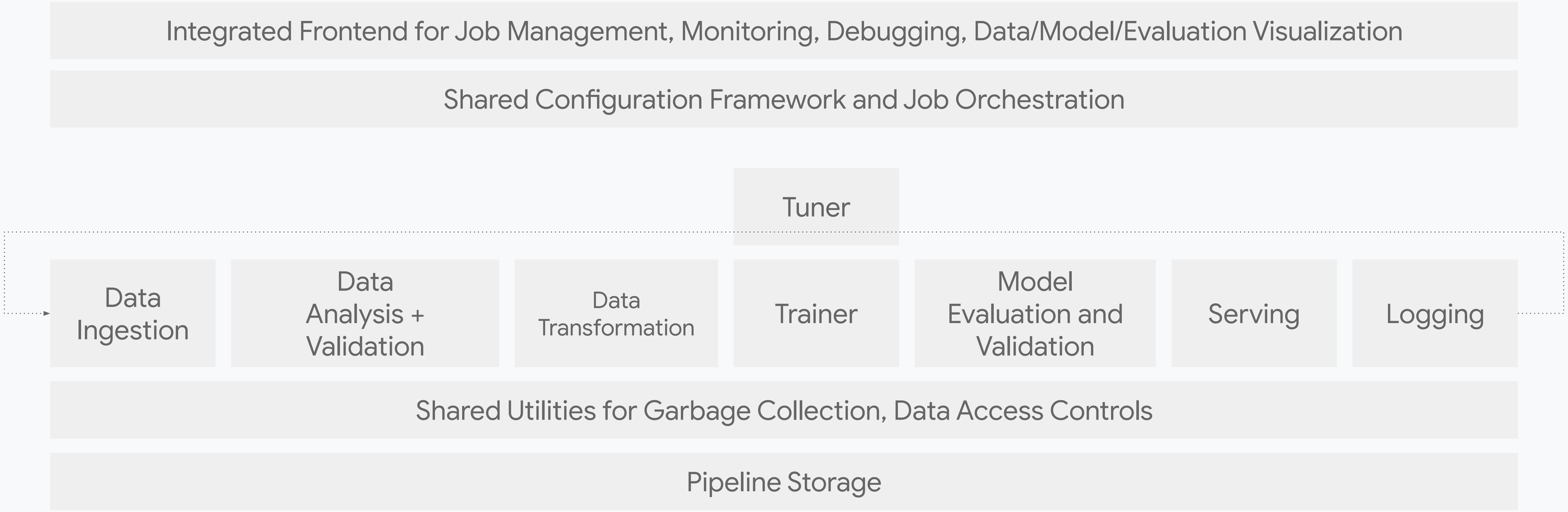


What's the other 95% of system code?

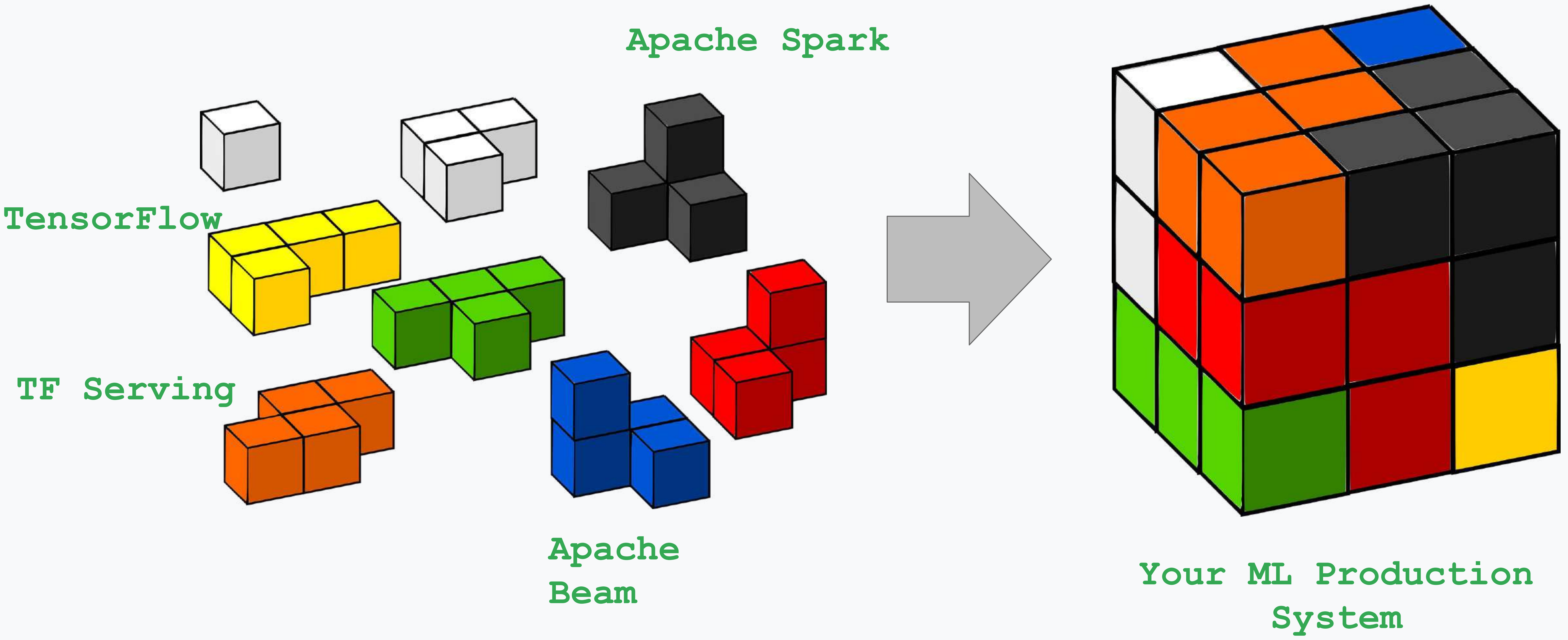




Other Components in a Production ML System



Reuse generic software frameworks whenever you can



Managed services handle infrastructure for you

Cloud Dataproc

Cloud Dataflow

Cloud Machine Learning Engine

Course 2: Production ML Systems

Module 1: Architecting Production ML Systems

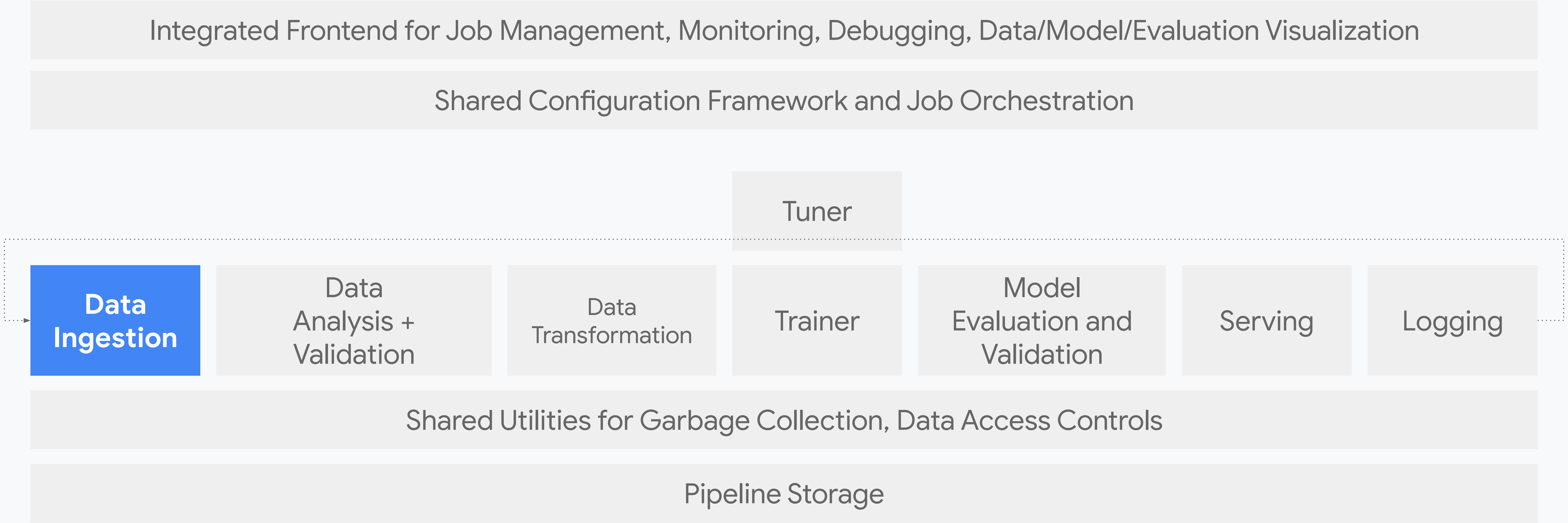
Lesson Title: **The Components of an ML System**

Presenter: Max Lotstein

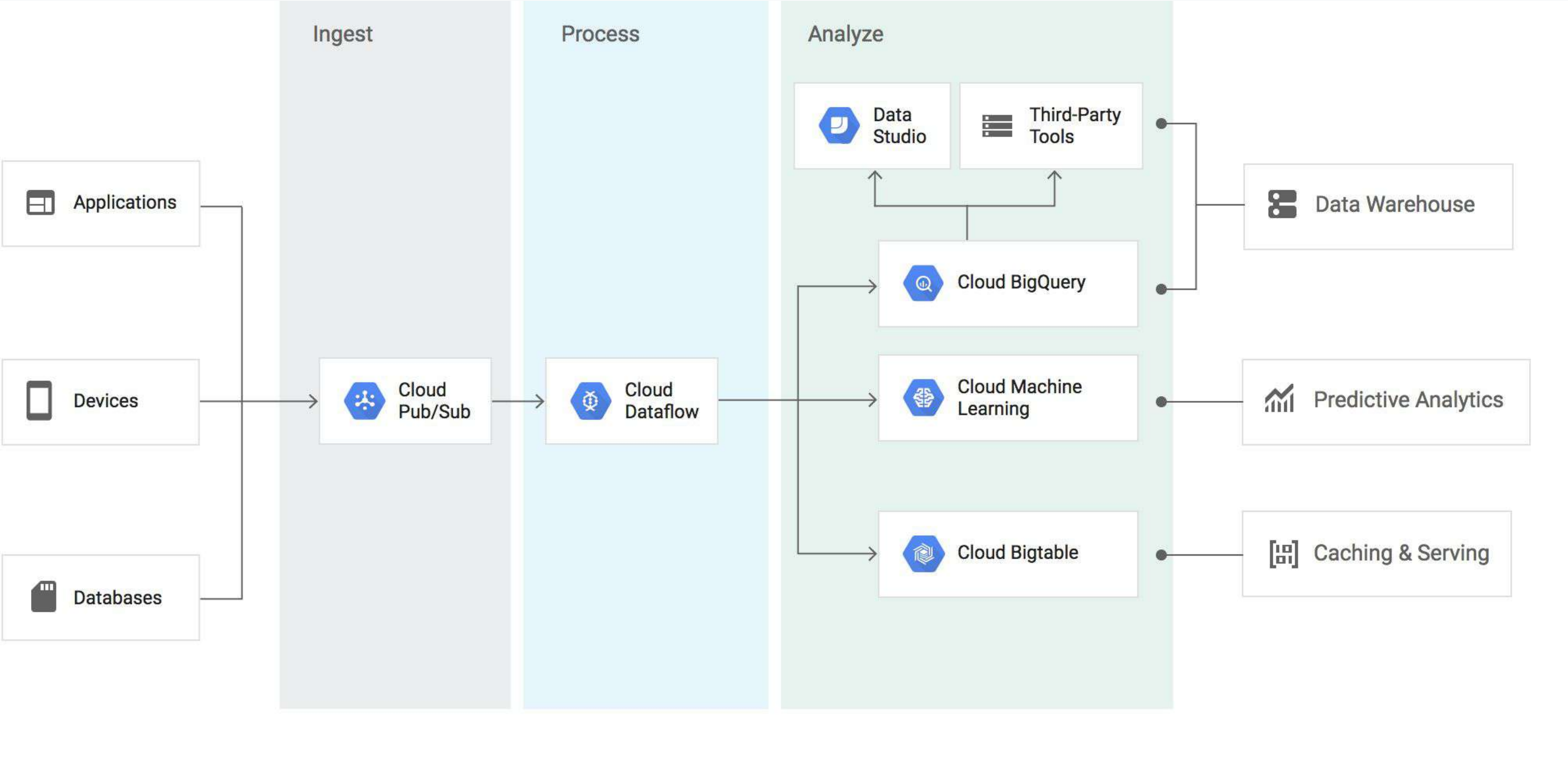
Format: Talking Head

Video Name: T-PSML-O_1_l2_the_components_of_an_ml_system

Production ML System Component: Data Ingestion



Streaming Data Ingestion Pipeline Architecture



Structured Batch Data Ingestion with BigQuery

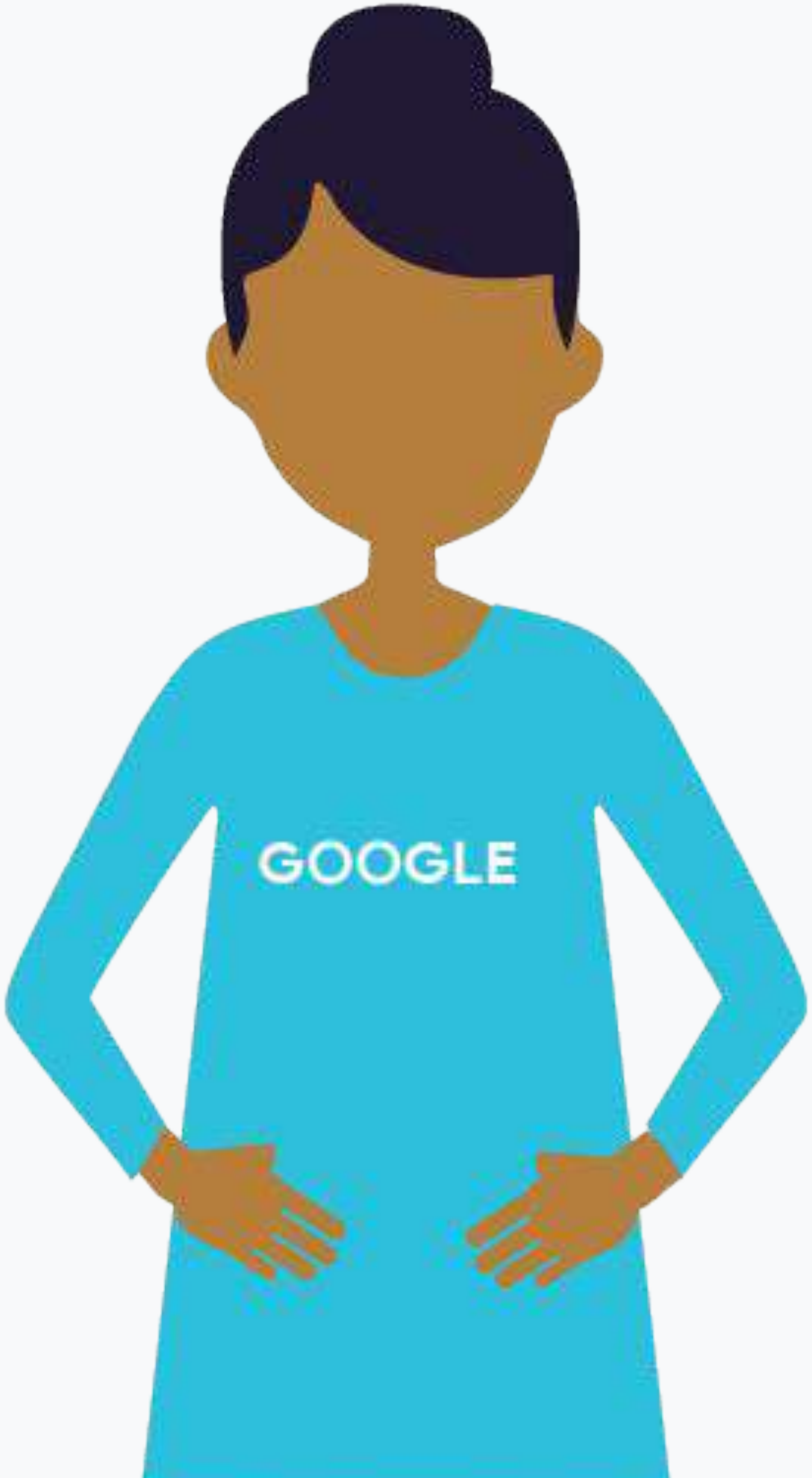
```python
# Assume a BigQuery has the following schema,
#     name      STRING,
#     age       INT,


# Create the parse_examples list of features.
features = dict(
  name=tf.FixedLenFeature([1], tf.string),
  age=tf.FixedLenFeature([1], tf.int32))

# Create a Reader.
reader = bigquery_reader_ops.BigQueryReader(project_id=PROJECT,
                                            dataset_id=DATASET,
                                            table_id=TABLE,
                                            timestamp_millis=TIME,
                                            num_partitions=NUM_PARTITIONS,
                                            features=features)

# Populate a queue with the BigQuery Table partitions.
queue = tf.train.string_input_producer(reader.partitions())

# Read and parse examples.
row_id, examples_serialized = reader.read(queue)
examples = tf.parse_example(examples_serialized, features=features)

# Process the Tensors examples["name"], examples["age"], etc...
```

BigQuery

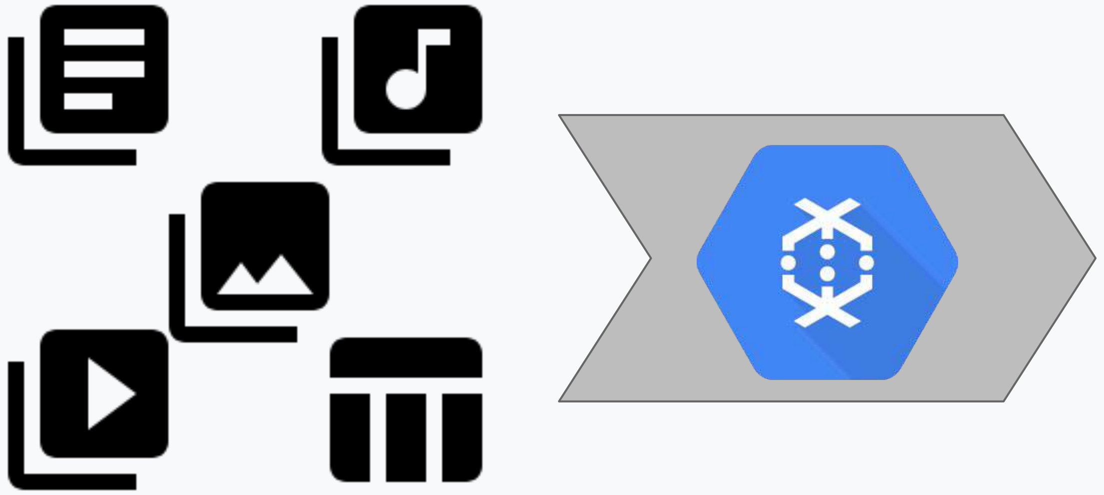
Structured Batch Data Ingestion with Cloud DataFlow

```
p = beam.Pipeline(argv=pipeline_args)
   input = p | 'ReadFromBQ' >> beam.io.Read(beam.io.BigQuerySource(known_args.input))
   predictions = input | 'Prediction' >> beam.ParDo(PredictDoFn(), known_args.model)
   predictions | 'WriteToBQ' >> beam.io.Write(beam.io.BigQuerySink(
       known_args.output,
       schema=schema,
       create_disposition=beam.io.BigQueryDisposition.CREATE_IF_NEEDED,
       write_disposition=beam.io.BigQueryDisposition.WRITE_TRUNCATE))
logging.getLogger().setLevel(logging.INFO)
p.run()
```

DataFlow

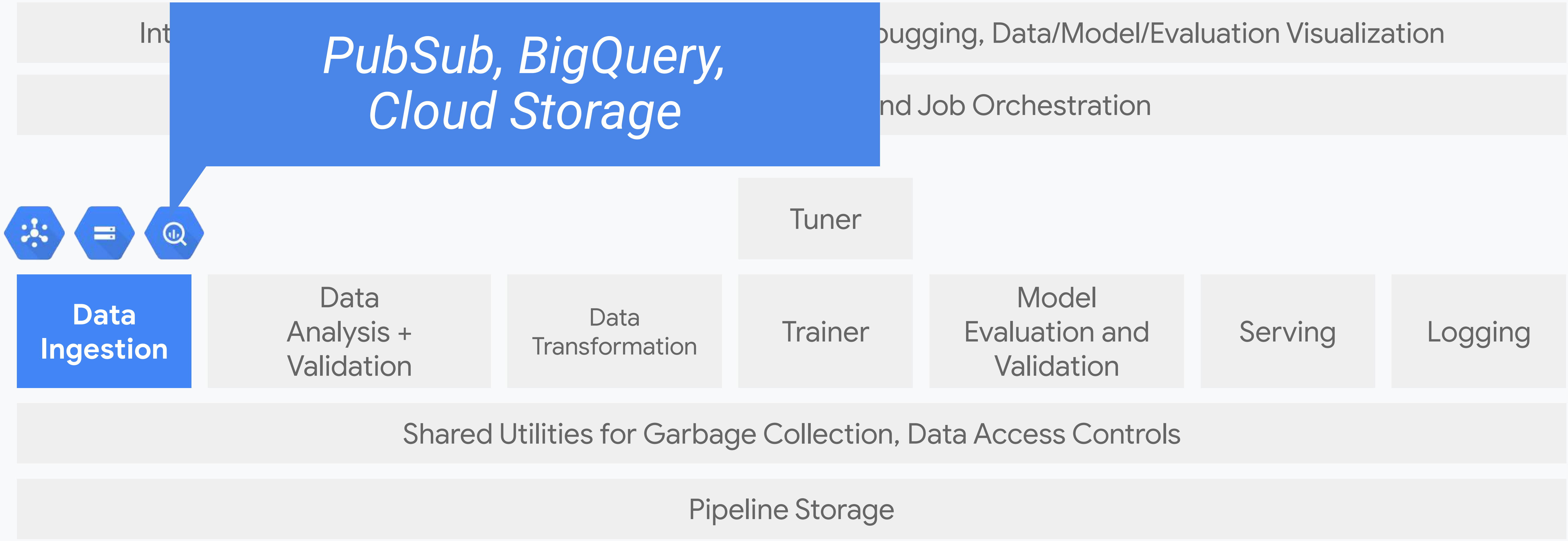


General Data Ingestion



TFRecord
CSV

Production ML System Component: Data Ingestion



Course 2: Production ML Systems

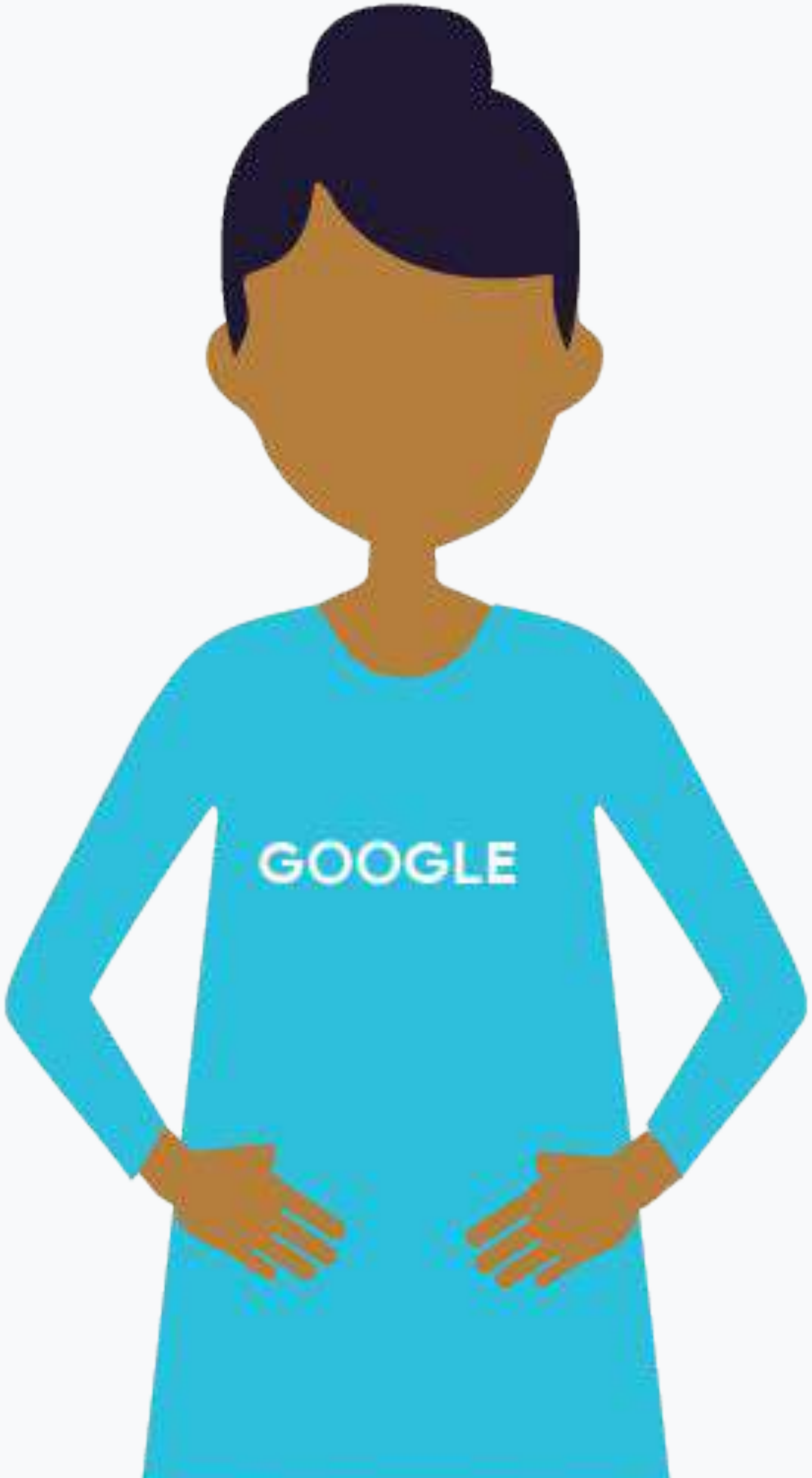
Module 1: Architecting Production ML Systems

Lesson Title: **The Components of an ML System: Data Analysis and Validation**

Presenter: Max Lotstein

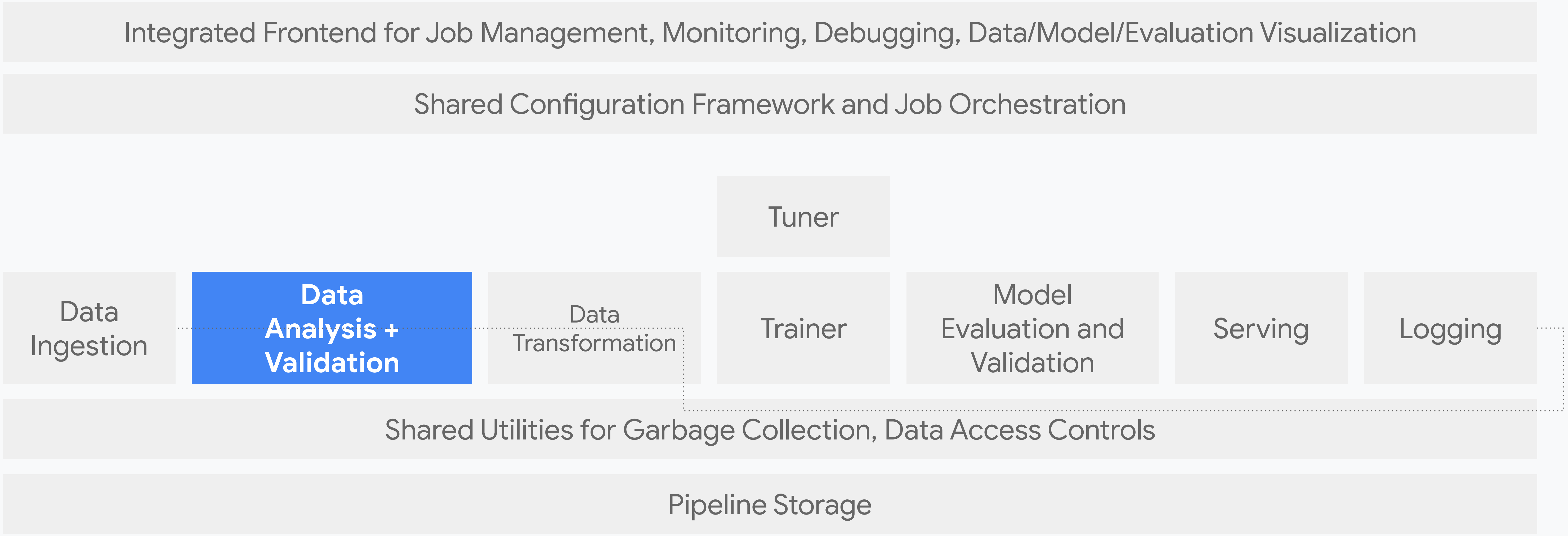
Format: Talking Head

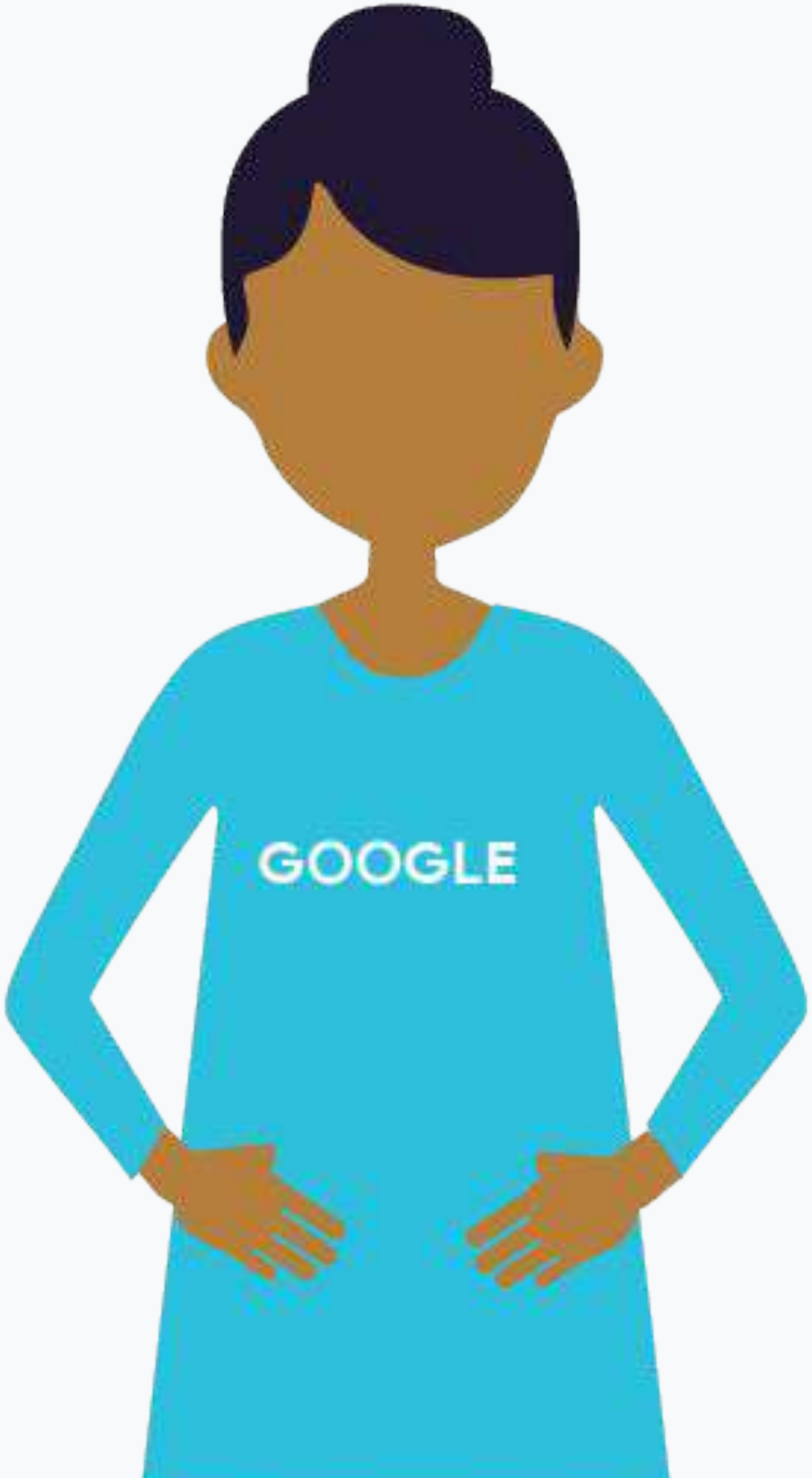
Video Name: T-PSML-O_1_l3_the_components_of_an_ml_system:_data_analysis_and_validation_



Data Analysis and Validation

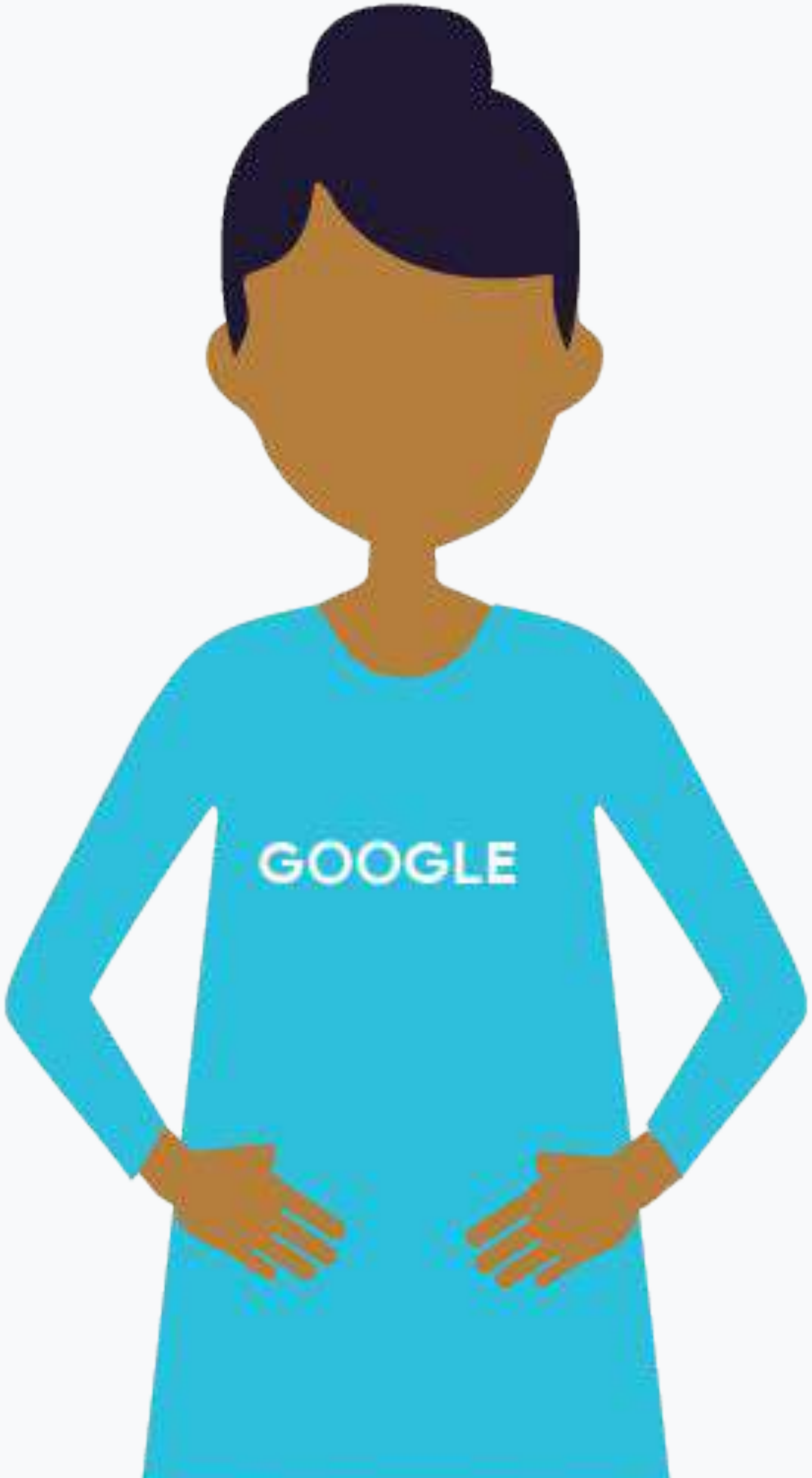
Production ML System Component: Data Analysis and Validation





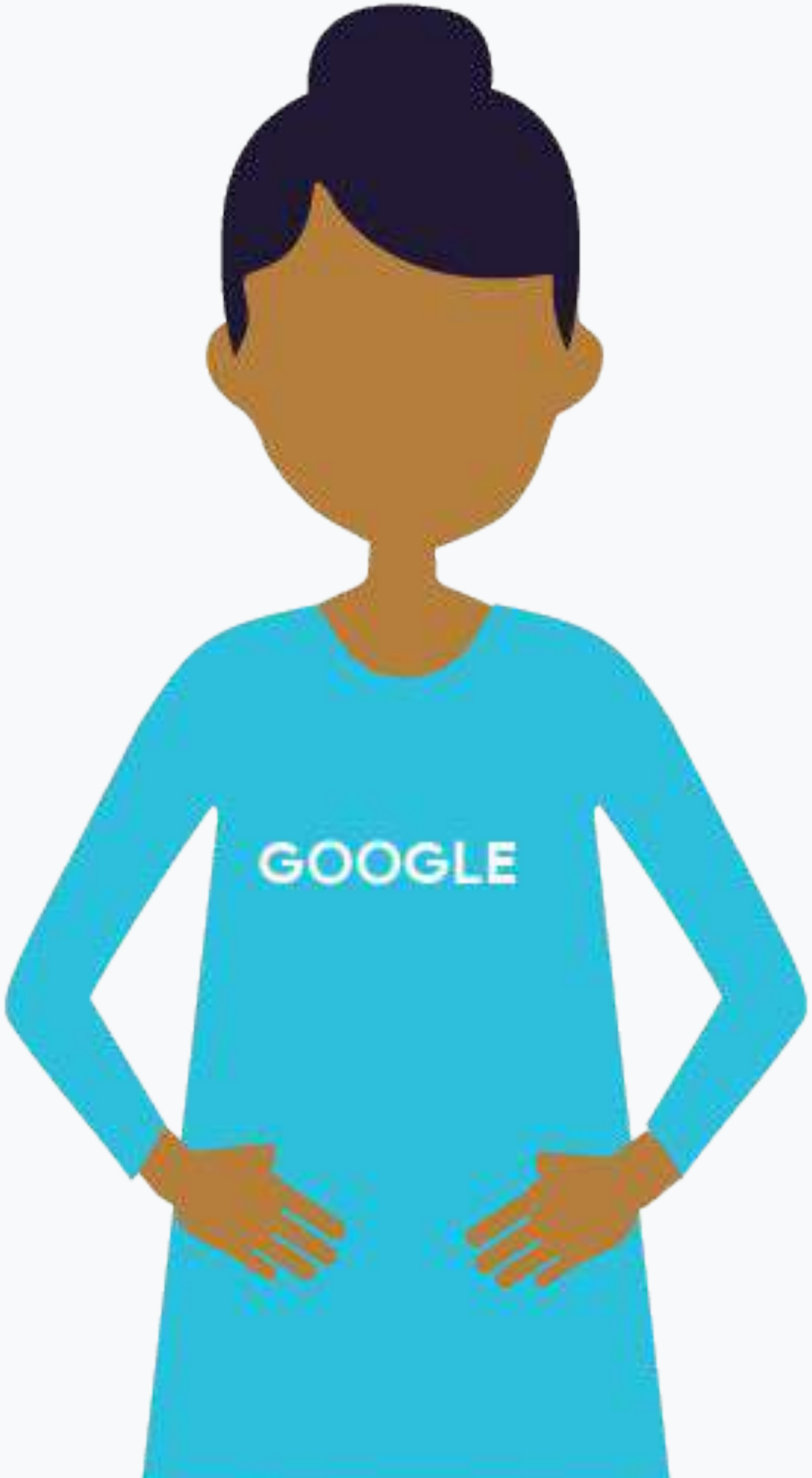
Data Analysis

Product Number	Product Name
112	Blue T-Shirt
231	Dog Frisbee
1333	Mobile Phone Charge

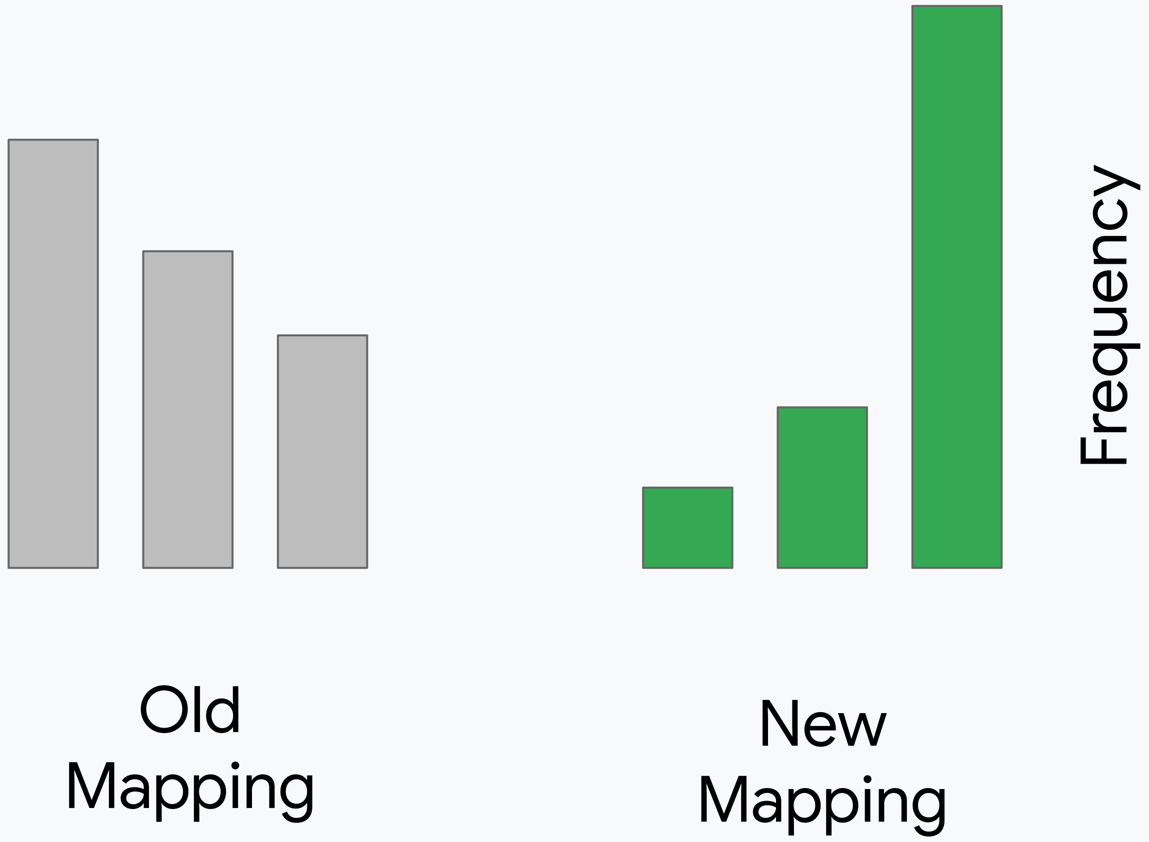


Data Analysis

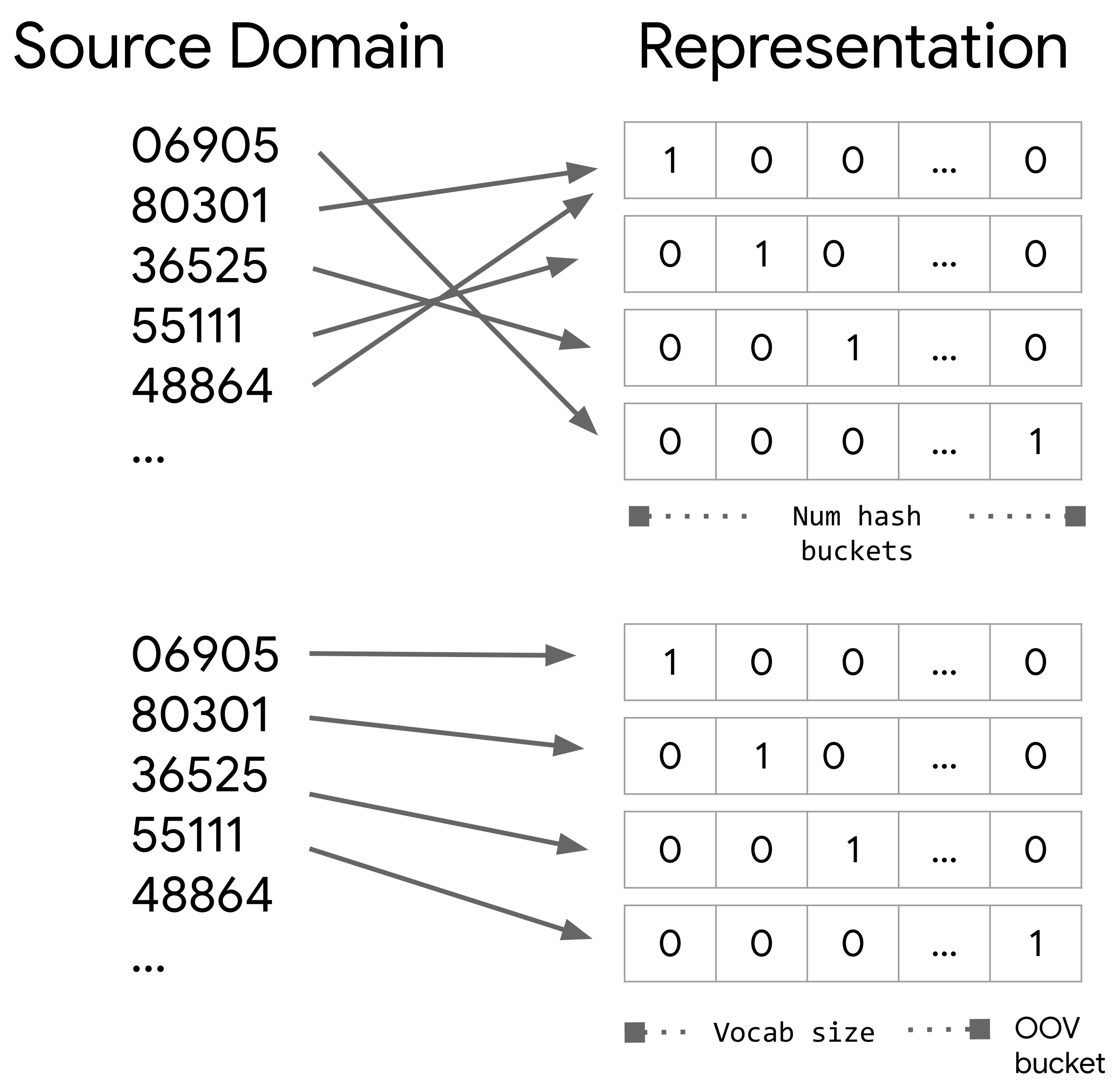
Product Number	Product Number	Product Name
112	231	Blue T-Shirt
231	231231231	Dog Frisbee
1333	112	Mobile Phone Charger

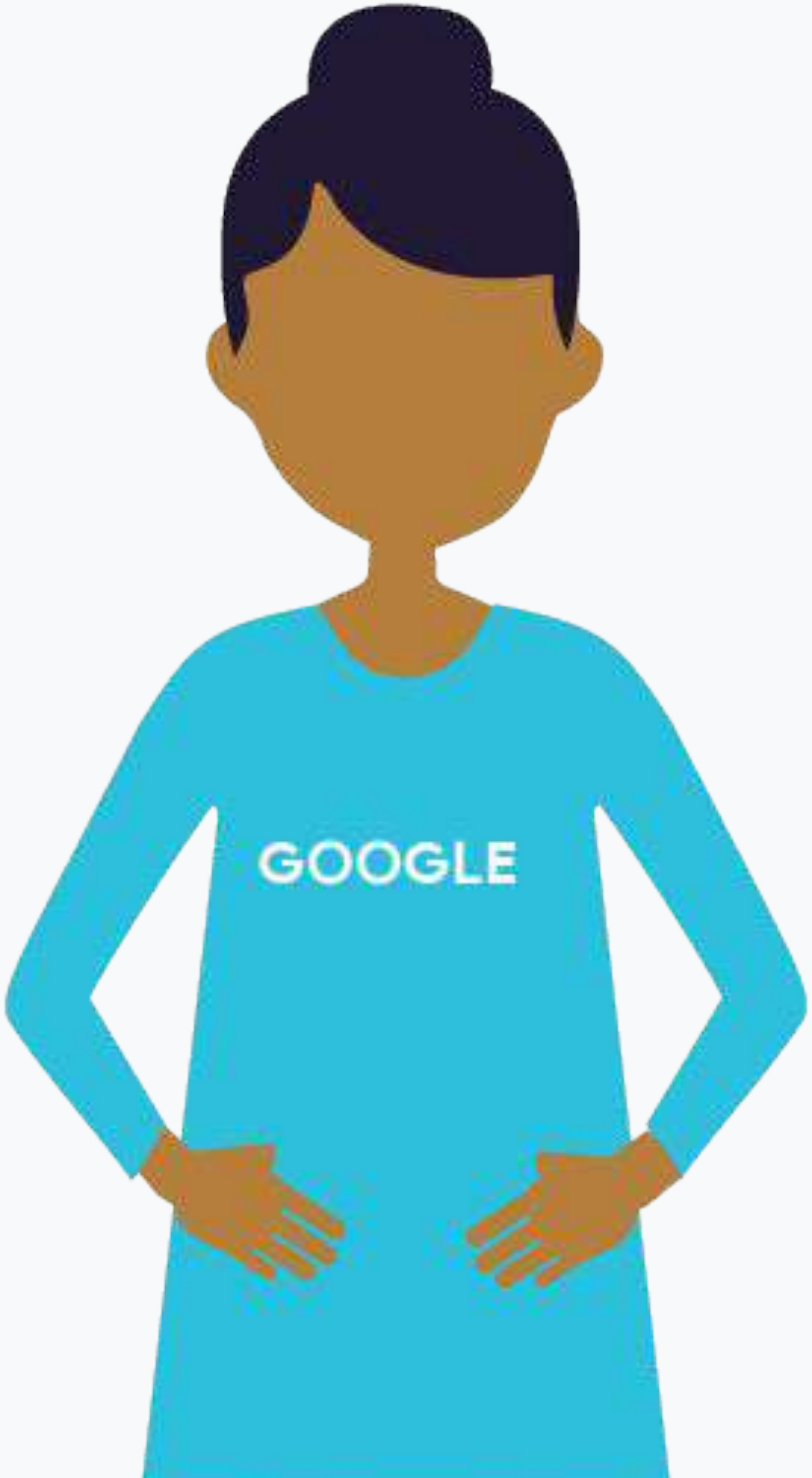


Data Analysis

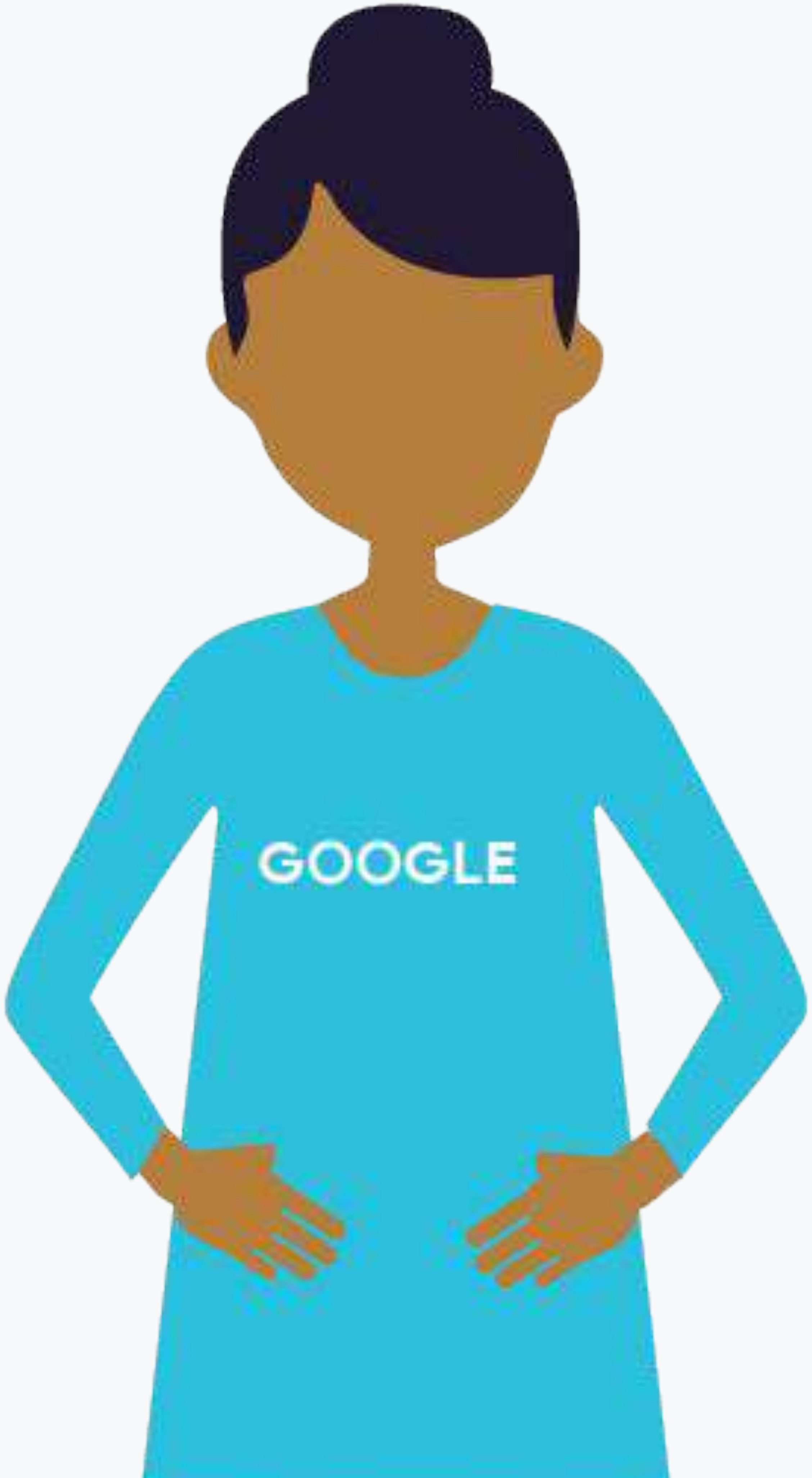


Data Analysis



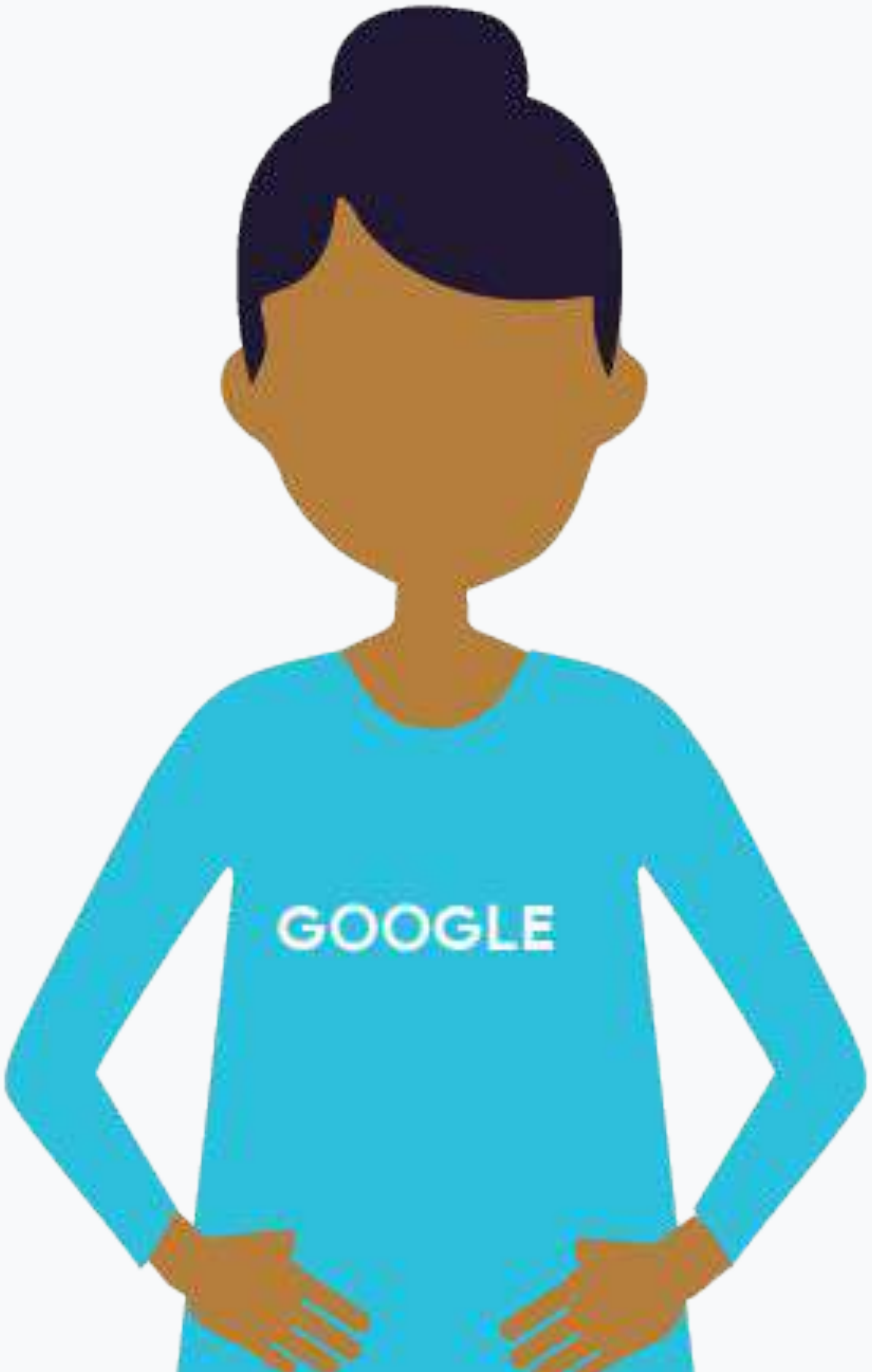


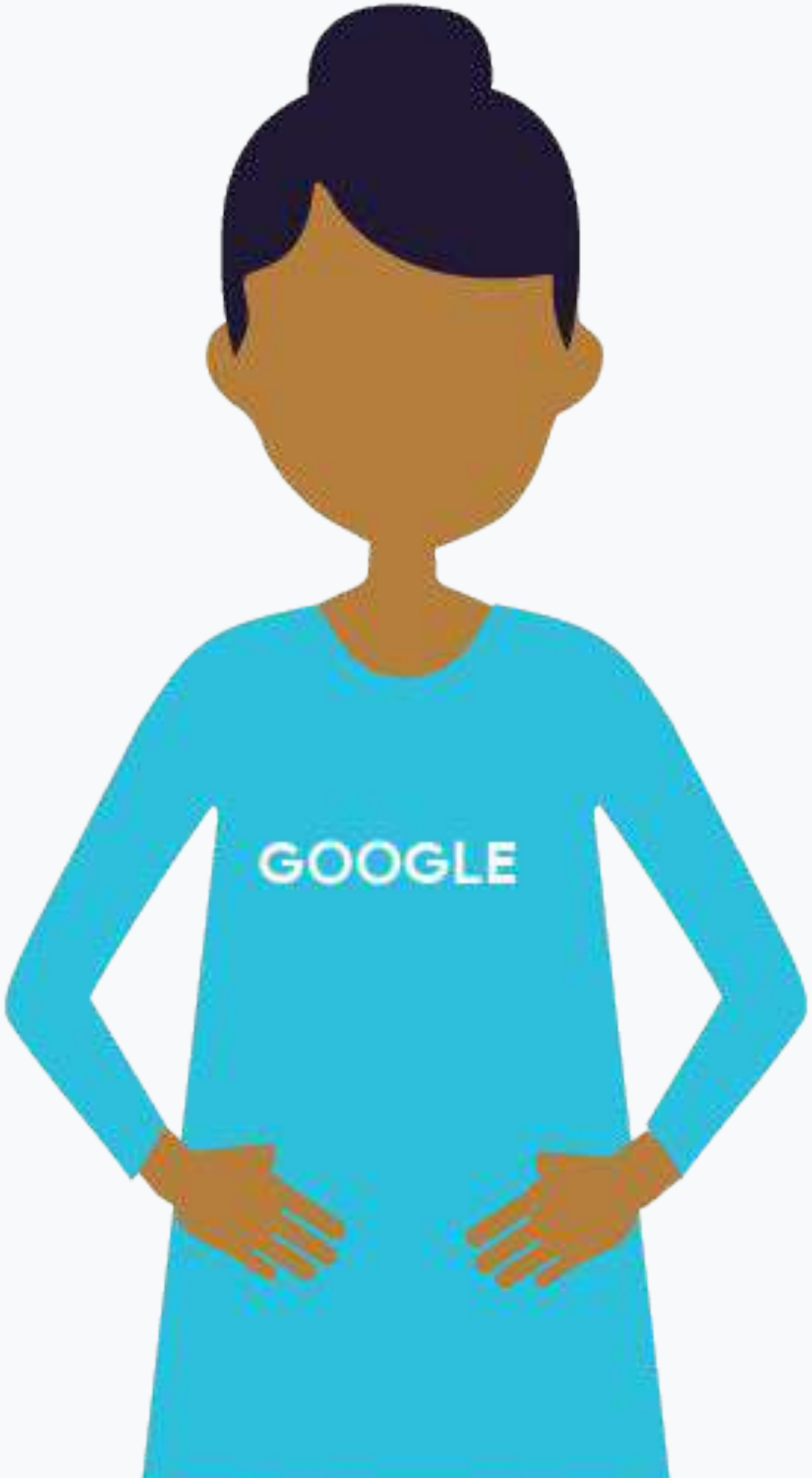
Data Validation:
Is the data healthy or not?



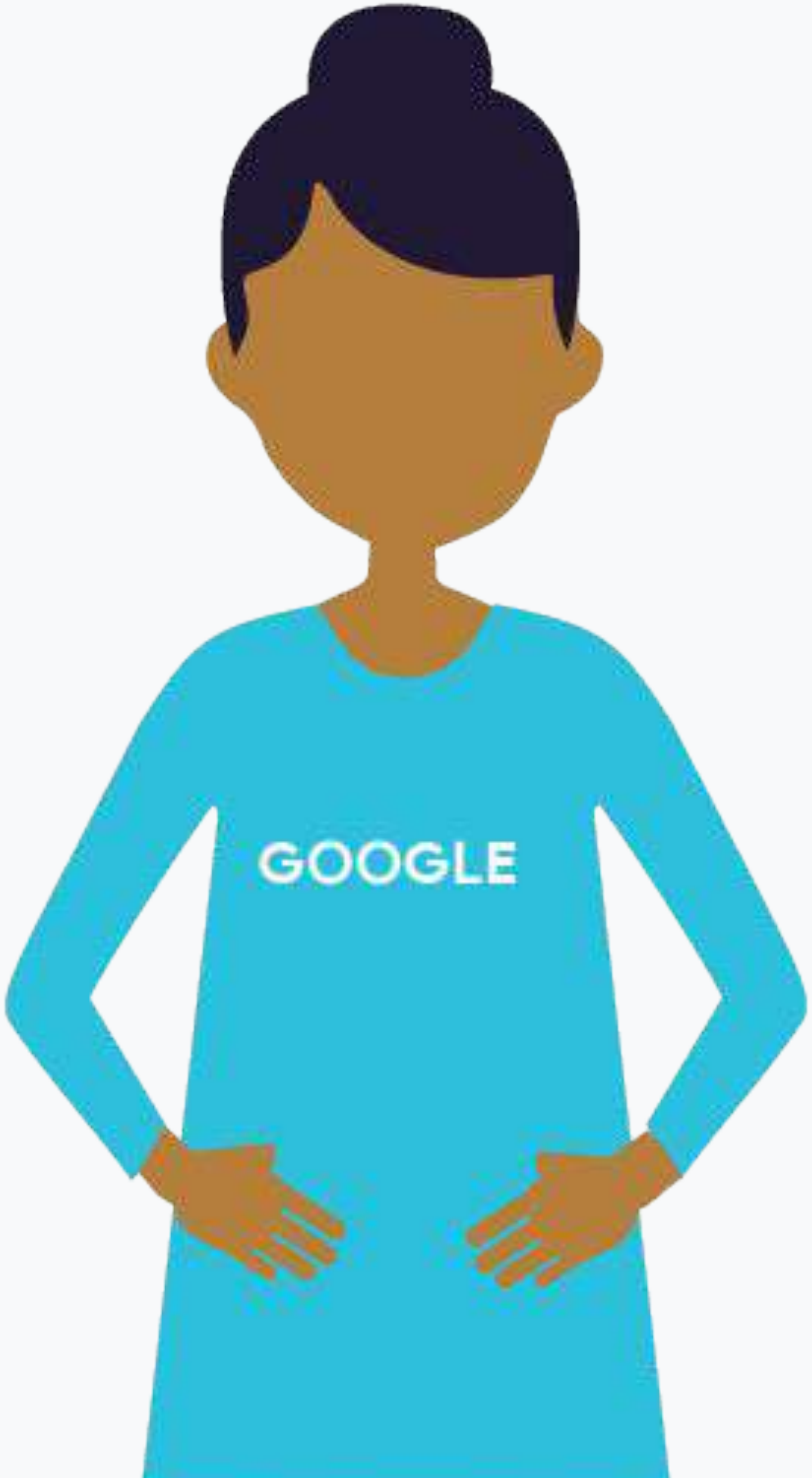
Data Validation:
Is the data healthy or not?

1) Is the new distribution similar enough to the old one?
2) Are all expected features present?
3) Are any unexpected features present?
4) Does the feature have the expected type?
5) Does an expected proportion of the examples contain the feature?
6) Do the examples have the expected number of values for feature?

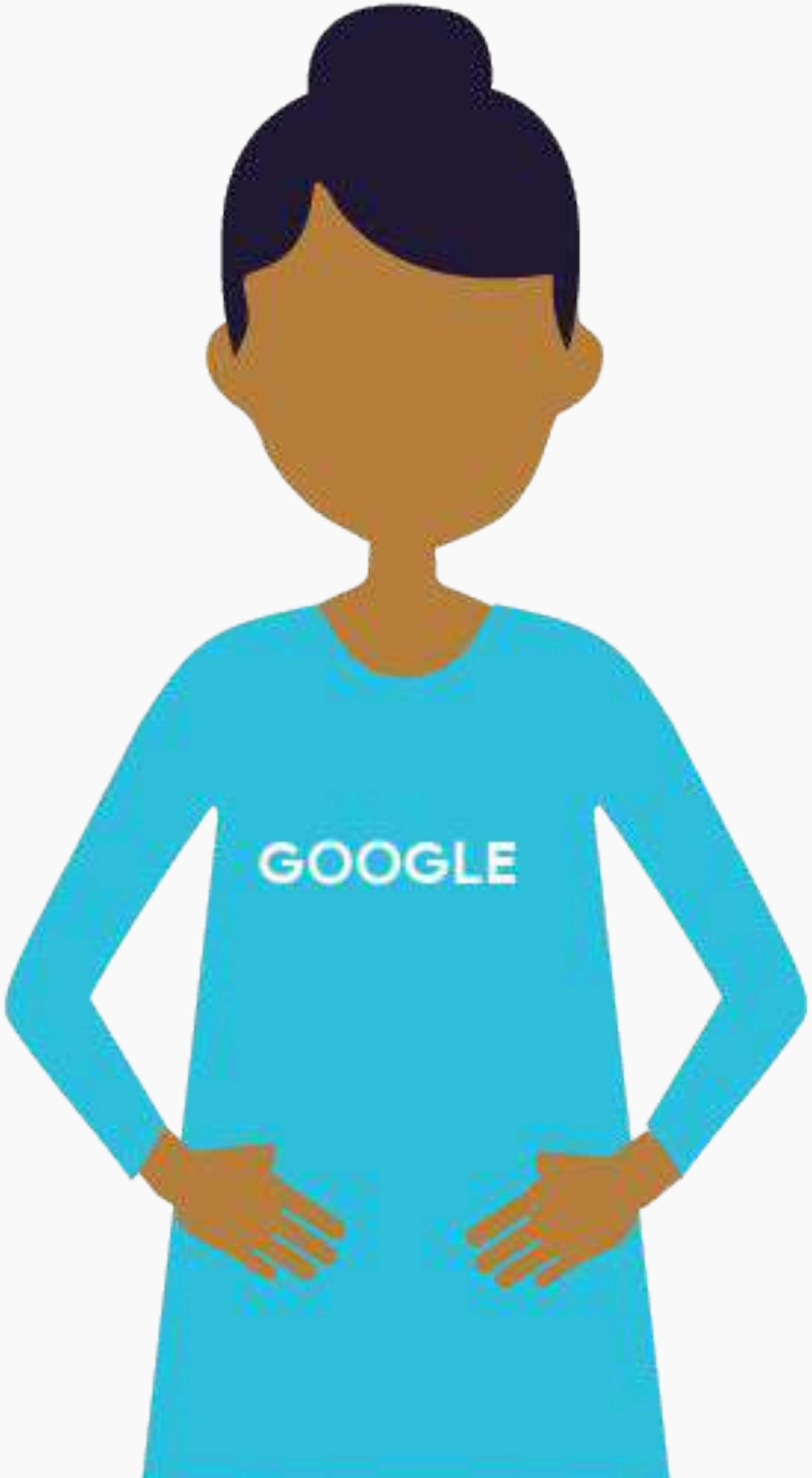

GOOGLE



Data Validation



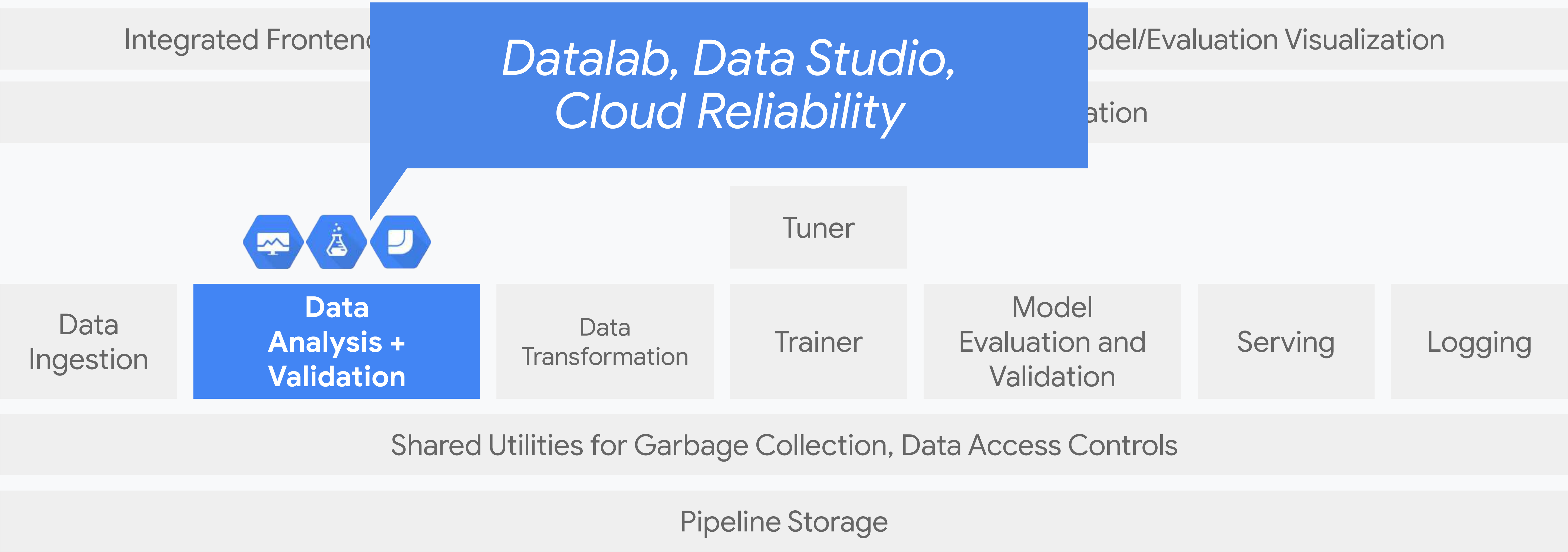
Data Validation



Quiz: Which tests would catch this error?

1) Is the new distribution similar enough to the old one?
2) Are all expected features present?
3) Are any unexpected features present?
4) Does the feature have the expected type?
5) Does an expected proportion of the examples contain the feature?
6) Do the examples have the expected number of values for the feature?

Production ML System Component: Data Analysis and Validation



Course 2: Production ML Systems

Module 1: Architecting Production ML Systems

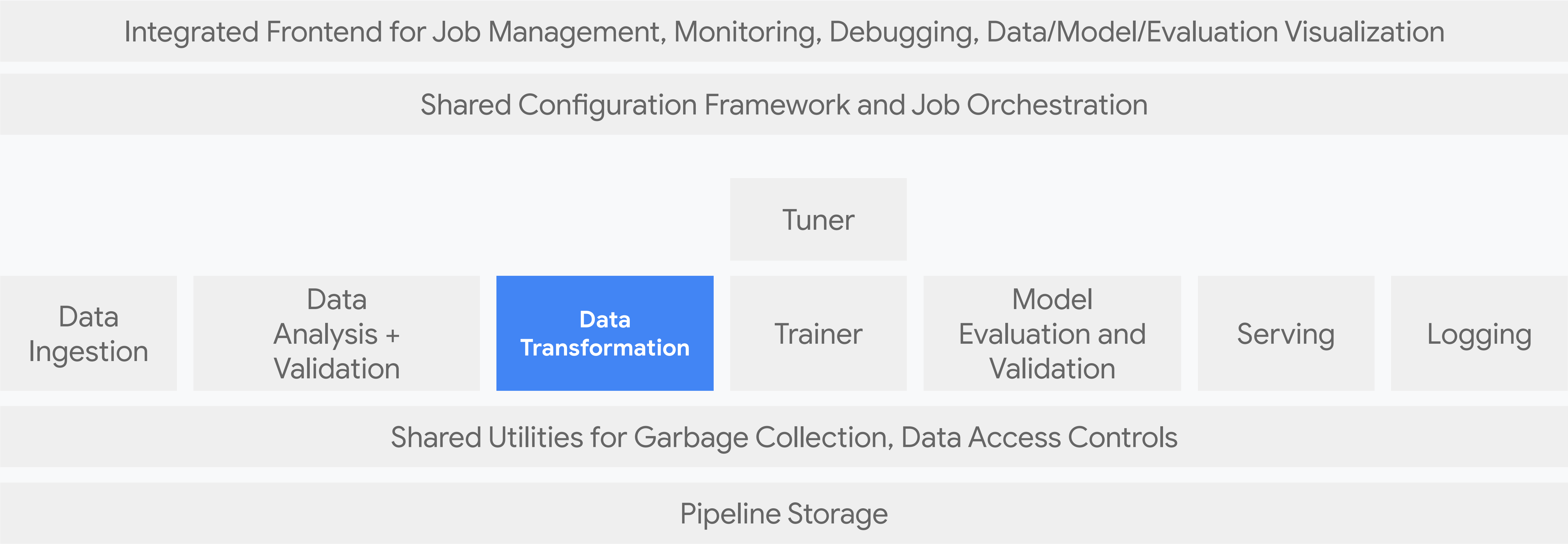
Lesson Title: **The Components of an ML System: Data Transformation + Trainer**

Presenter: Max Lotstein

Format: Talking Head

Video Name: T-PSML-O_1_l4_the_components_of_an_ml_system:_data_transformation_+_trainer

Production ML System Component: Data Transformation



Production ML System Component: Data Transformation

Integrated Frontend for Job Management, Monitoring, Debugging, Data/Model/Evaluation Visualization

Shared Configuration Fr

Dataflow, Dataproc, Dataprep

Tuner

Data Ingestion

Data Analysis + Validation

Data Transformation

Trainer

Model Evaluation and Validation

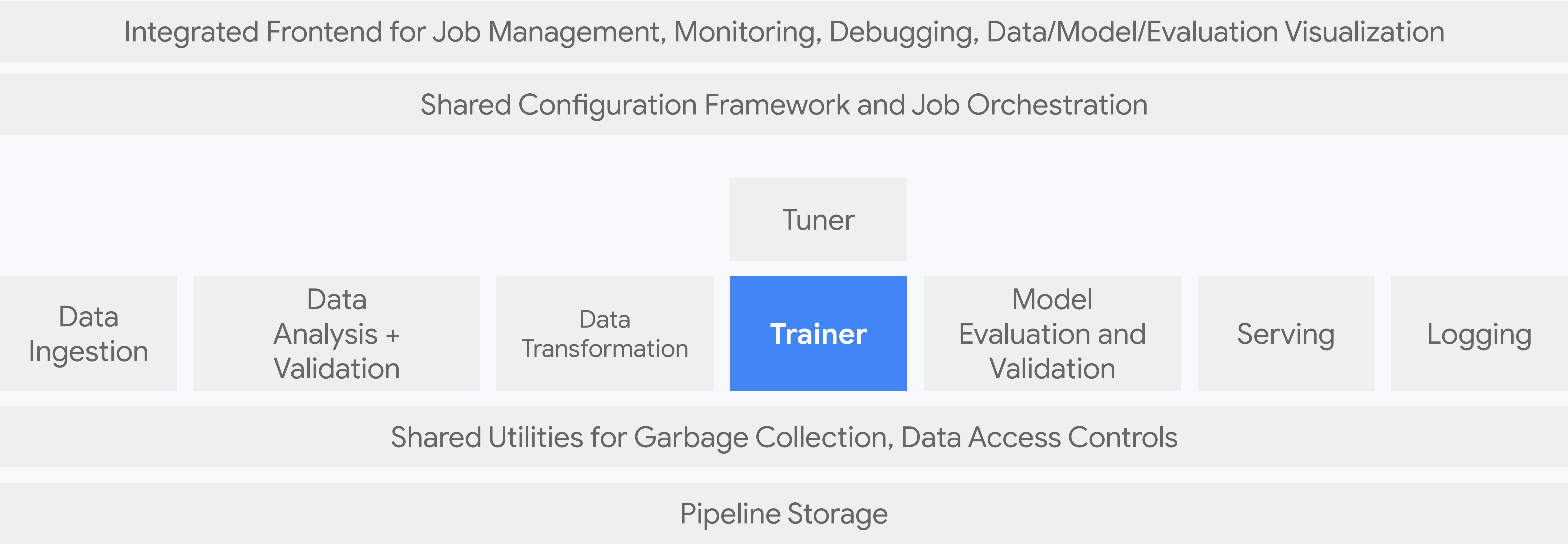
Serving

Logging

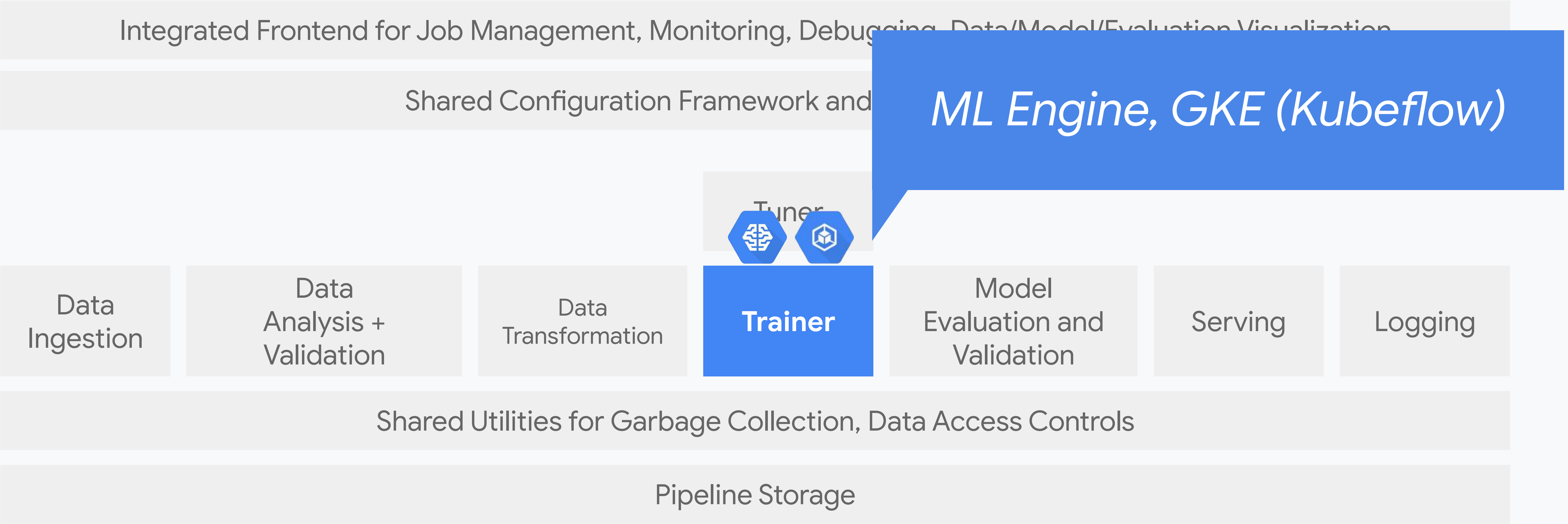
Shared Utilities for Garbage Collection, Data Access Controls

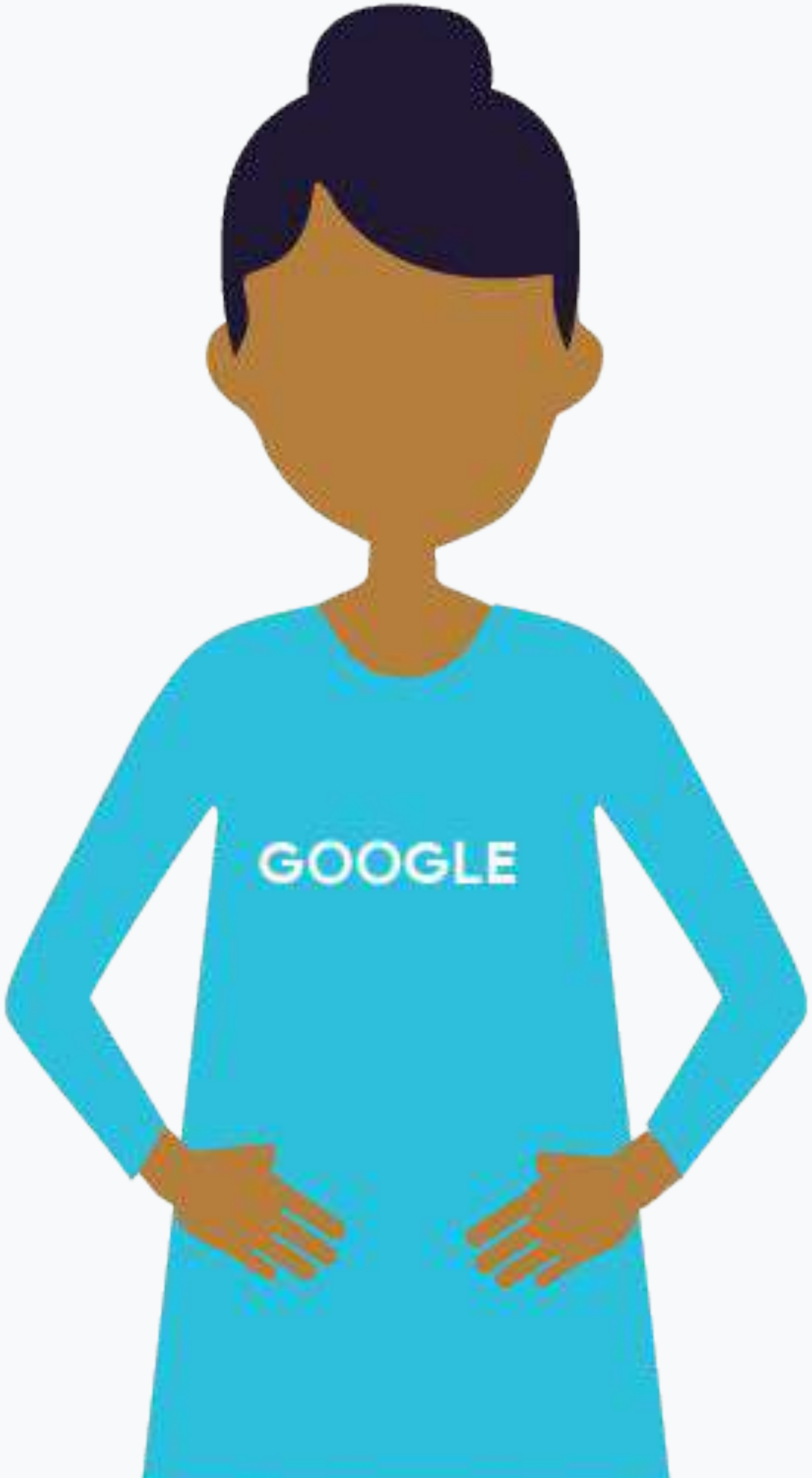
Pipeline Storage

Production ML System Component: Trainer



Production ML System Component: Trainer





Cloud ML Engine

1) Scalable
2) Integrated with Tuner, Logging, Serving components
3) Experiment-oriented
4) Open

Course 2: Production ML Systems

Module 1: Architecting Production ML Systems

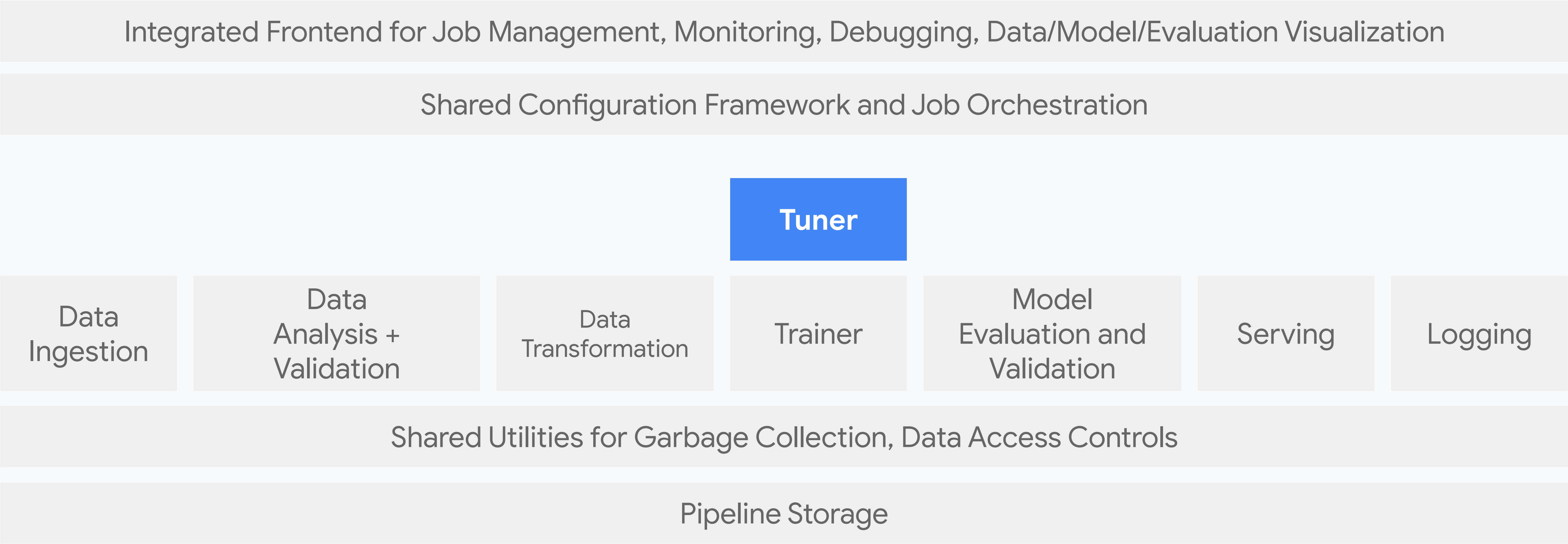
Lesson Title: **The Components of an ML System: Tuner + Model Evaluation and Validation**

Presenter: Max Lotstein

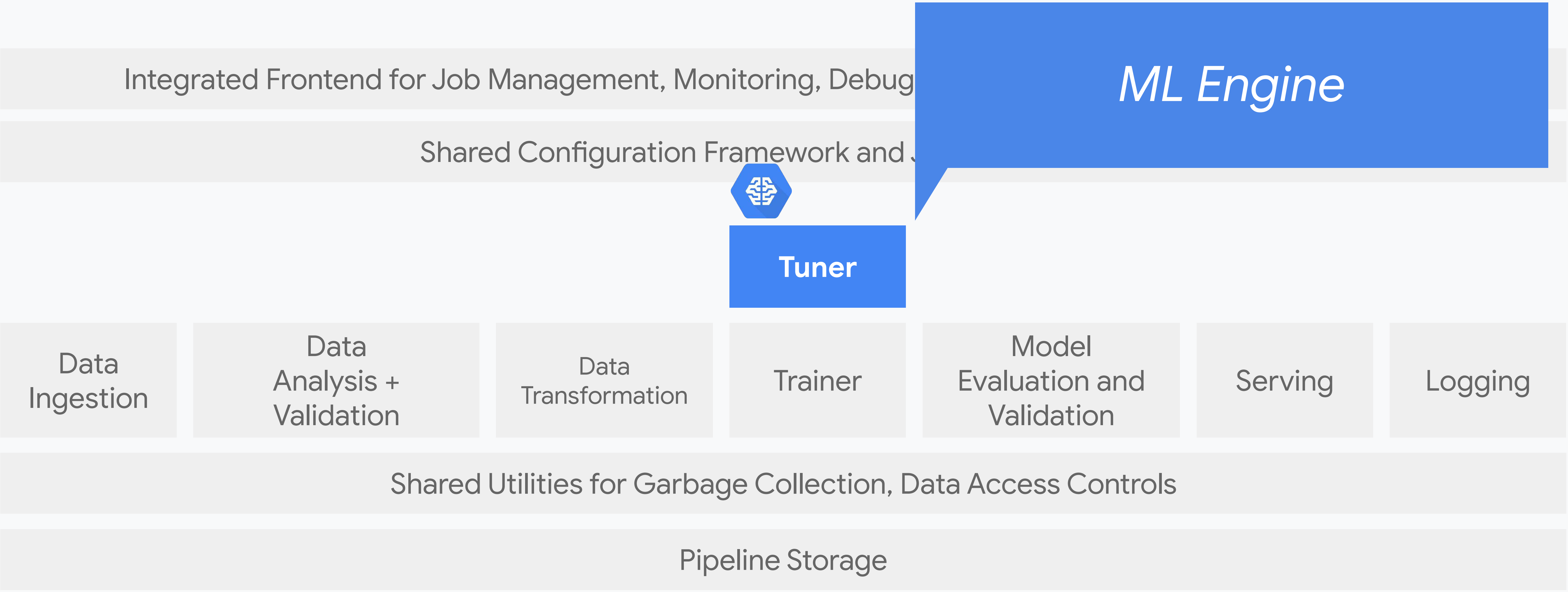
Format: Talking Head

Video Name
T-PSML-O_1_l5_the_components_of_an_ml_system:_tuner_+_model_evaluation_and_validation:

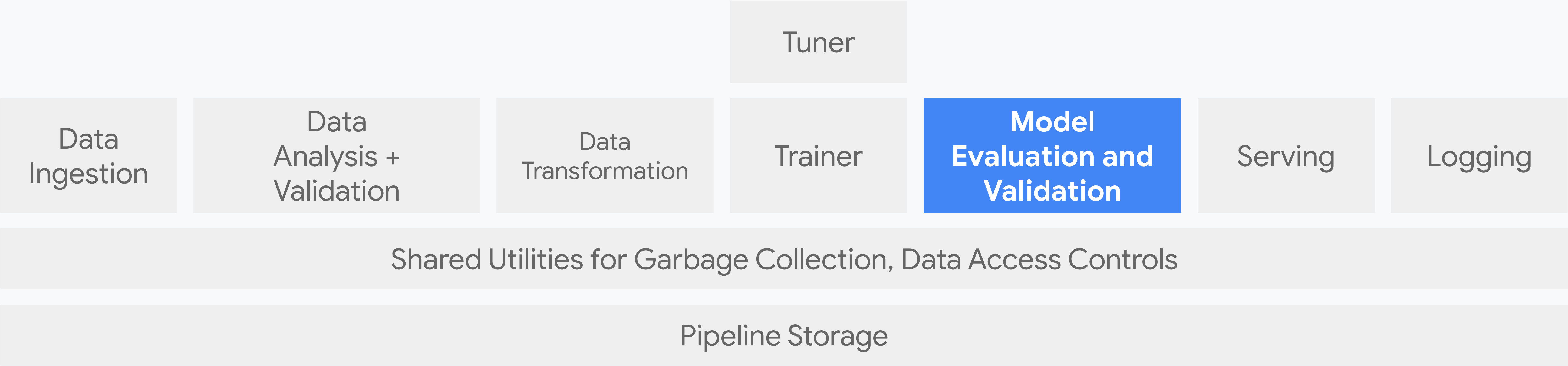
Production ML System Component: Tuner

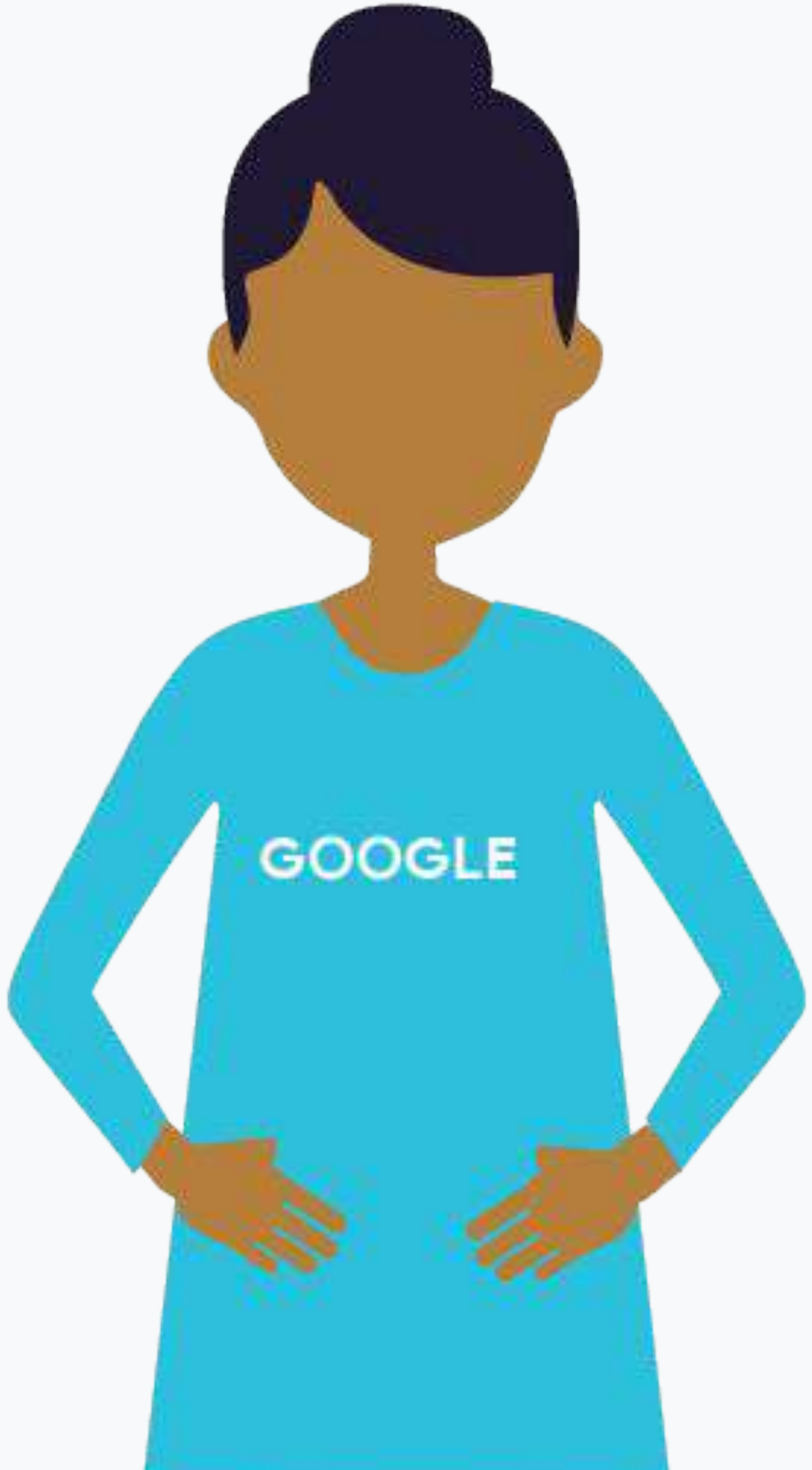


Production ML System Component: Tuner



Production ML System Component: Model Evaluation and Validation



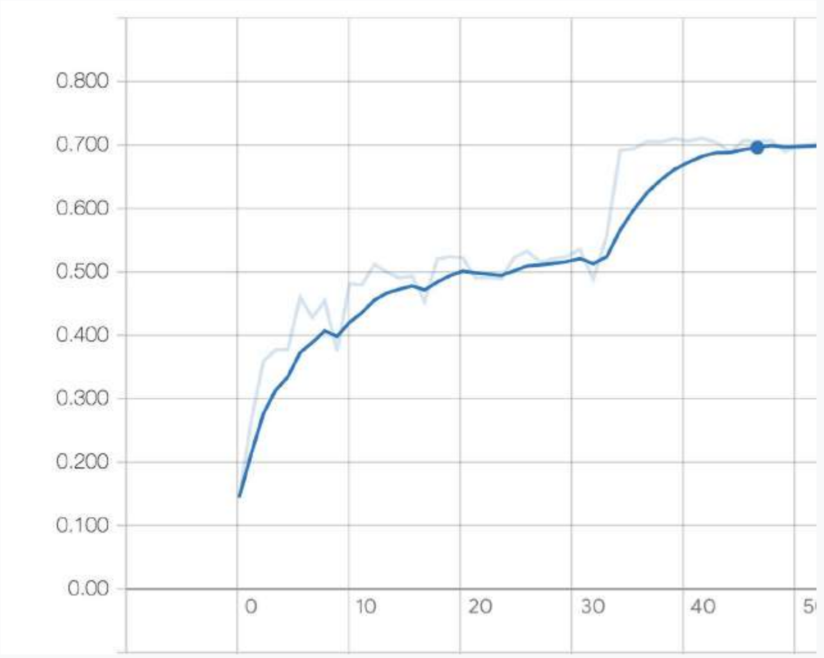


A good model is hard to find

Model Safeness

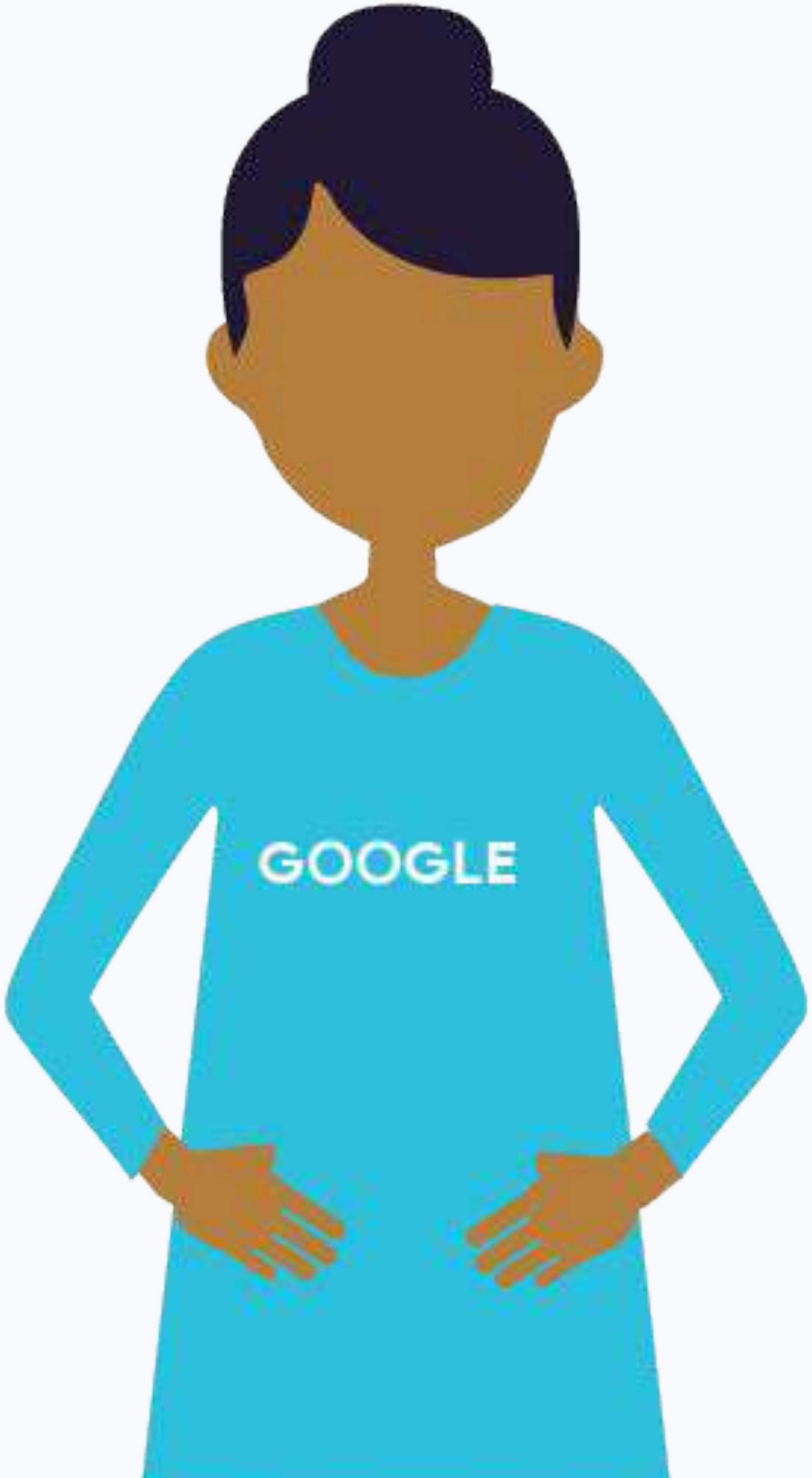
Likeliness to crash

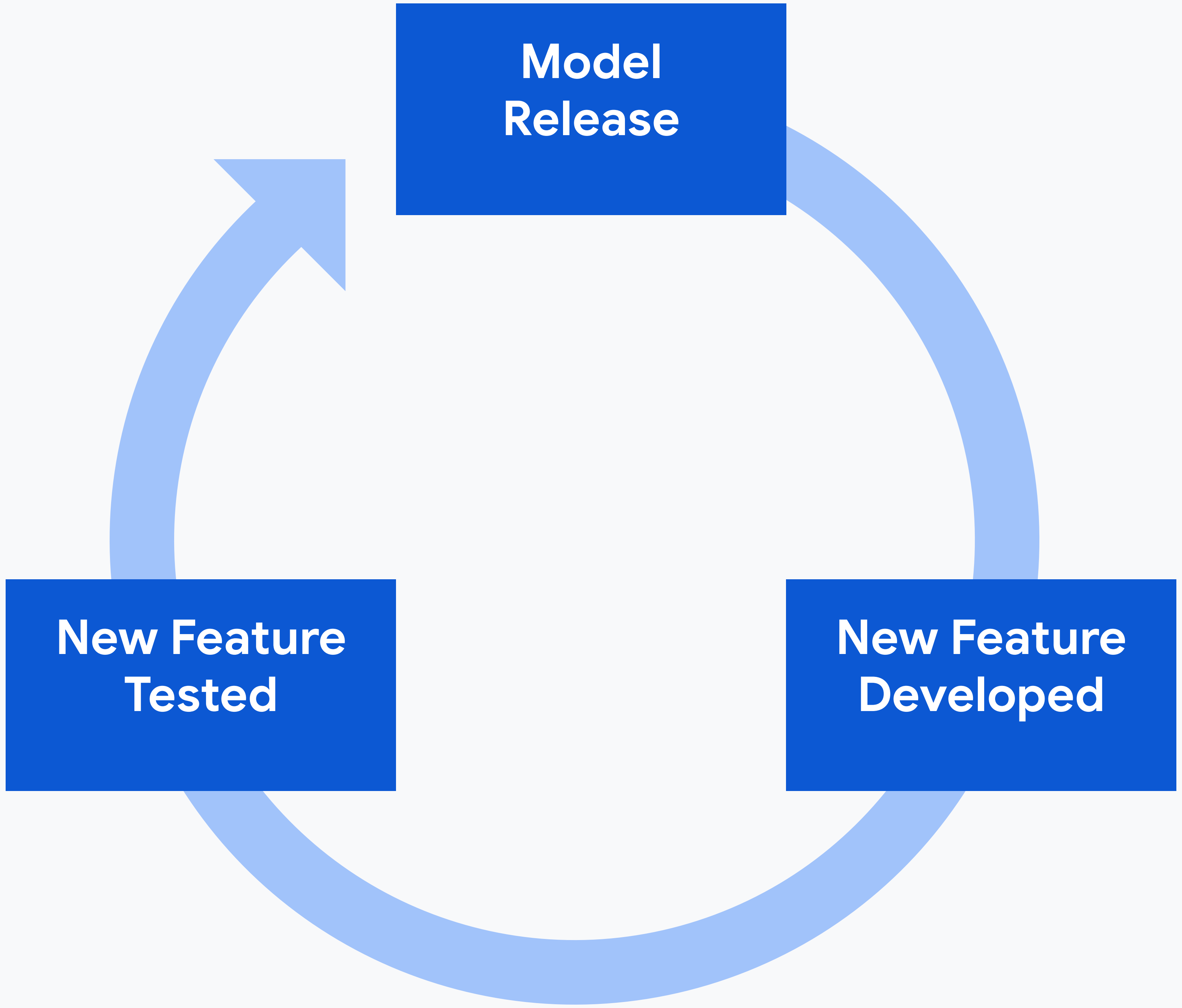
Prediction Quality

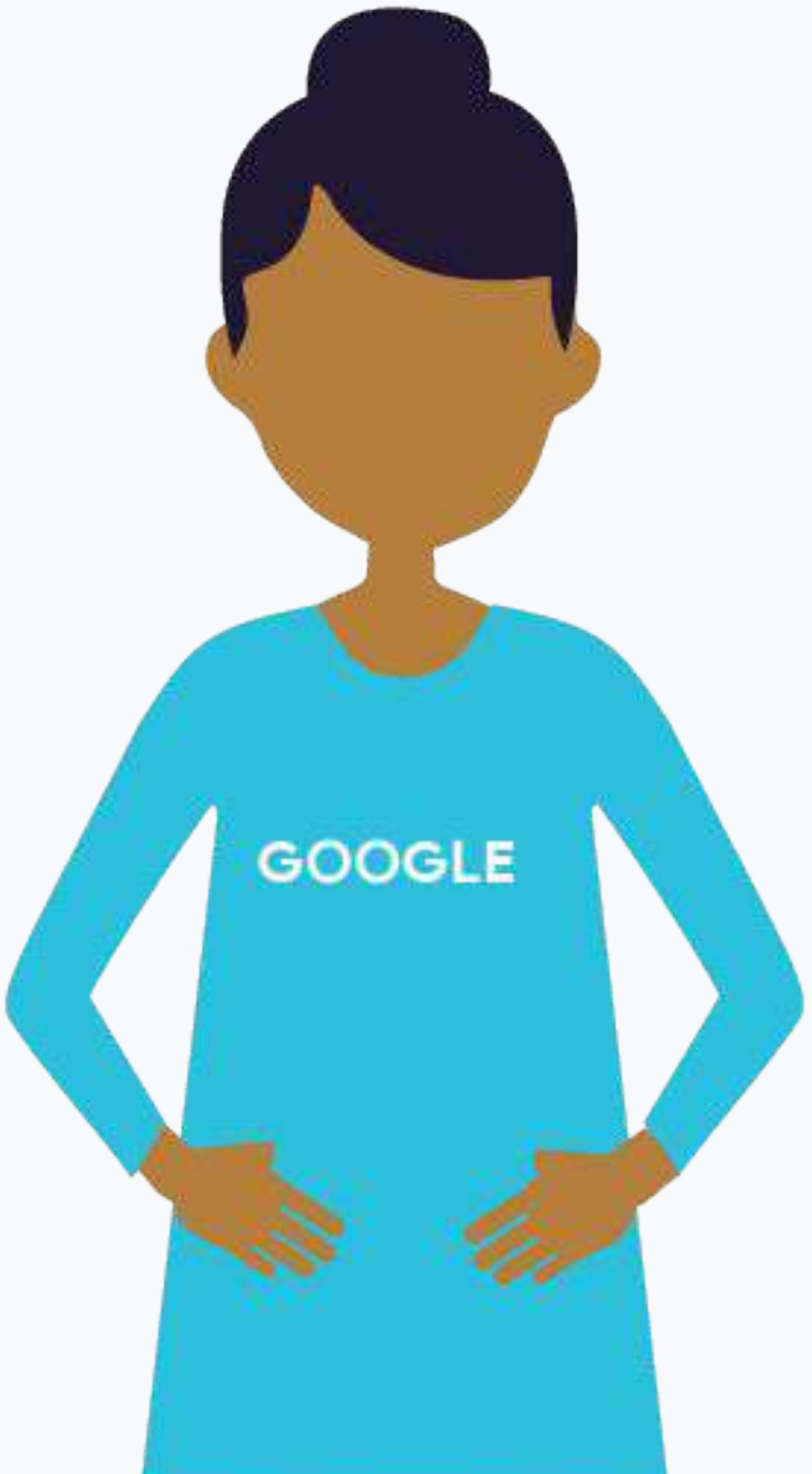


Accuracy vs Time

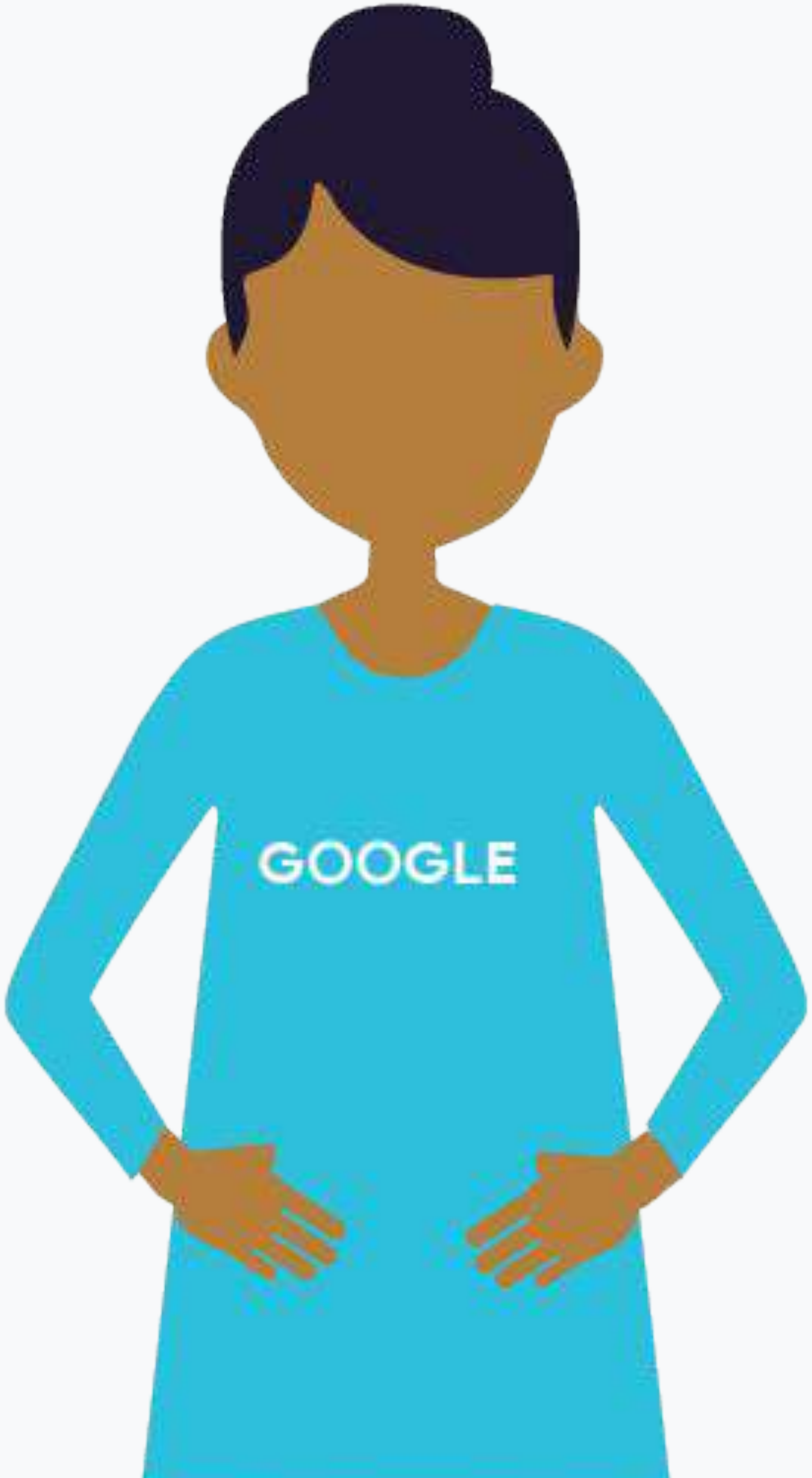
Model evaluation

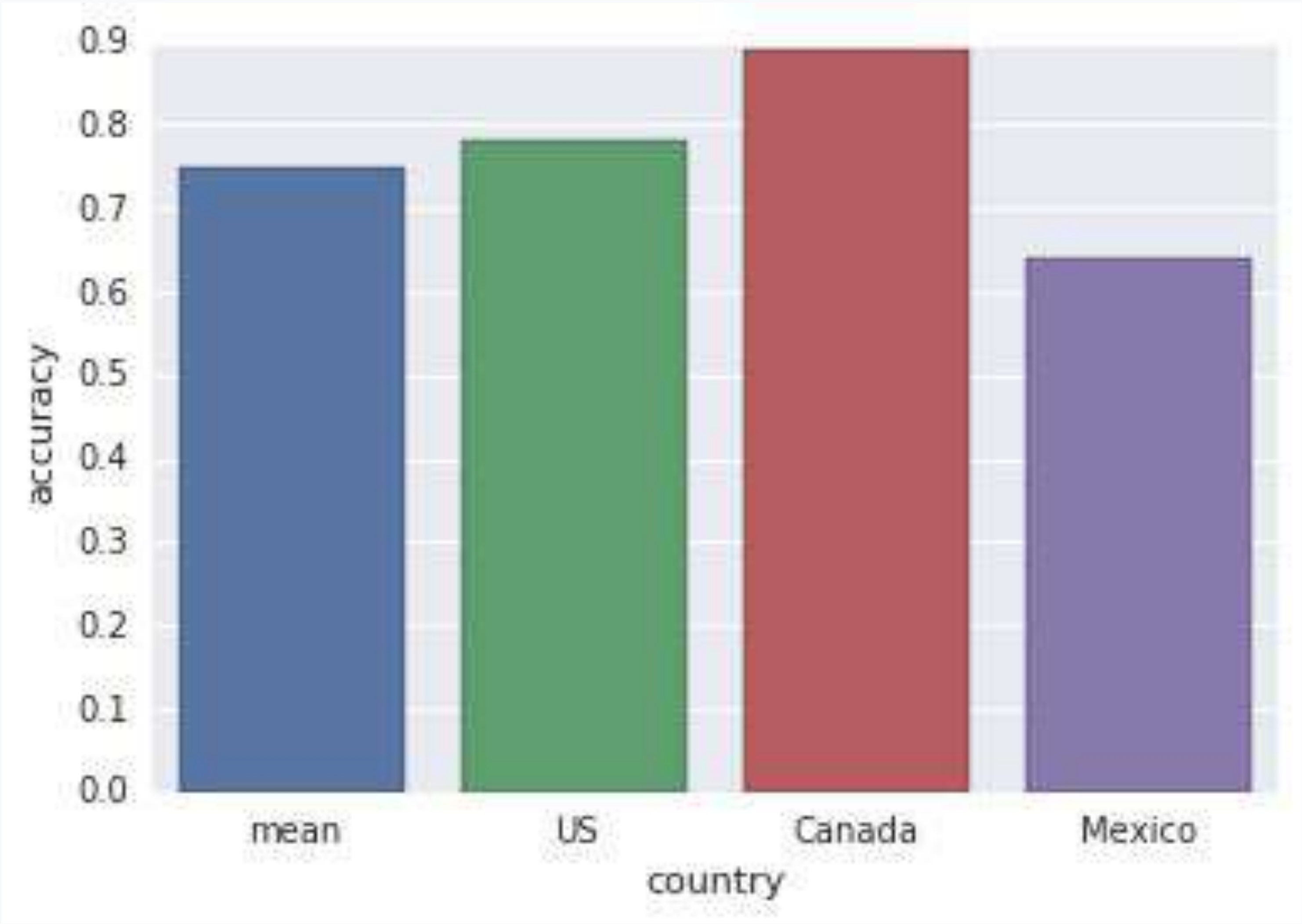





GOOGLE

Model Validation





Production ML System Component: Model Evaluation and Validation

Integrated Frontend for Job Management, Monitoring, Debugging, Data/Model/Evaluation Visualization

Shared Configuration Framework and Job Orchestration

TFX Model Analysis

Tuner

Data Ingestion

Data Analysis + Validation

Data Transformation

Trainer

Model Evaluation and Validation

Serving

Logging

Shared Utilities for Garbage Collection, Data Access Controls

Pipeline Storage

Course 2: Production ML Systems

Module 1: Architecting Production ML Systems

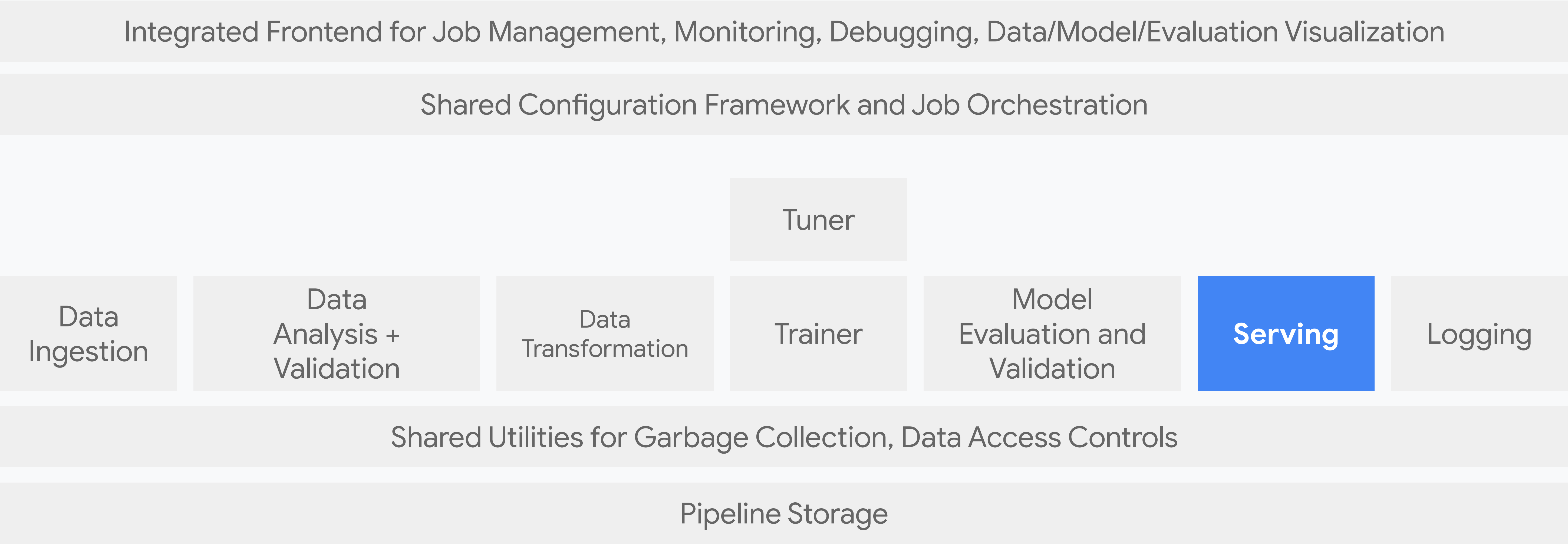
Lesson Title: **The Components of an ML System: Serving**

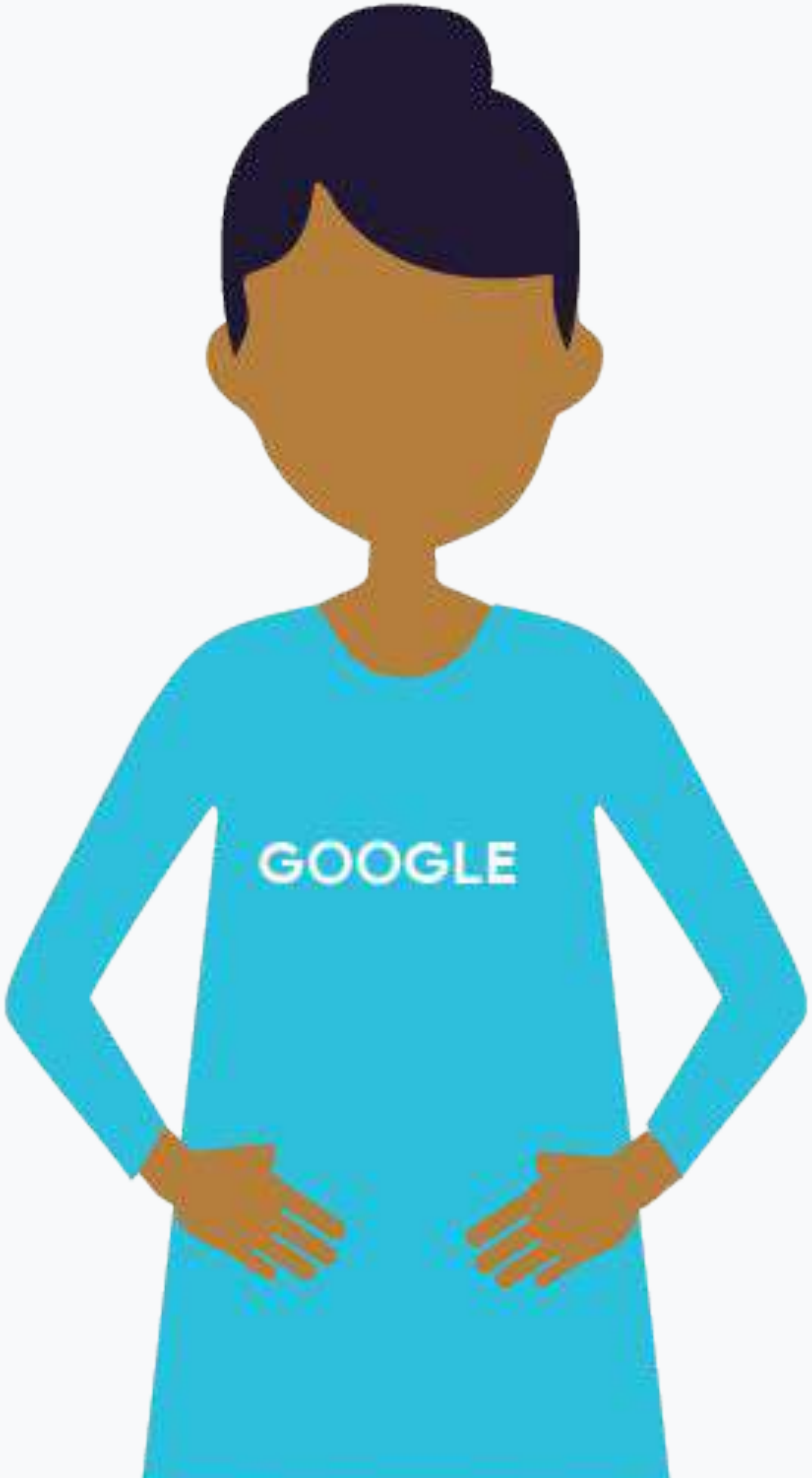
Presenter: Max Lotstein

Format: Talking Head

Video Name:
T-PSML-O_1_l6_the_components_of_an_ml_system:_serving

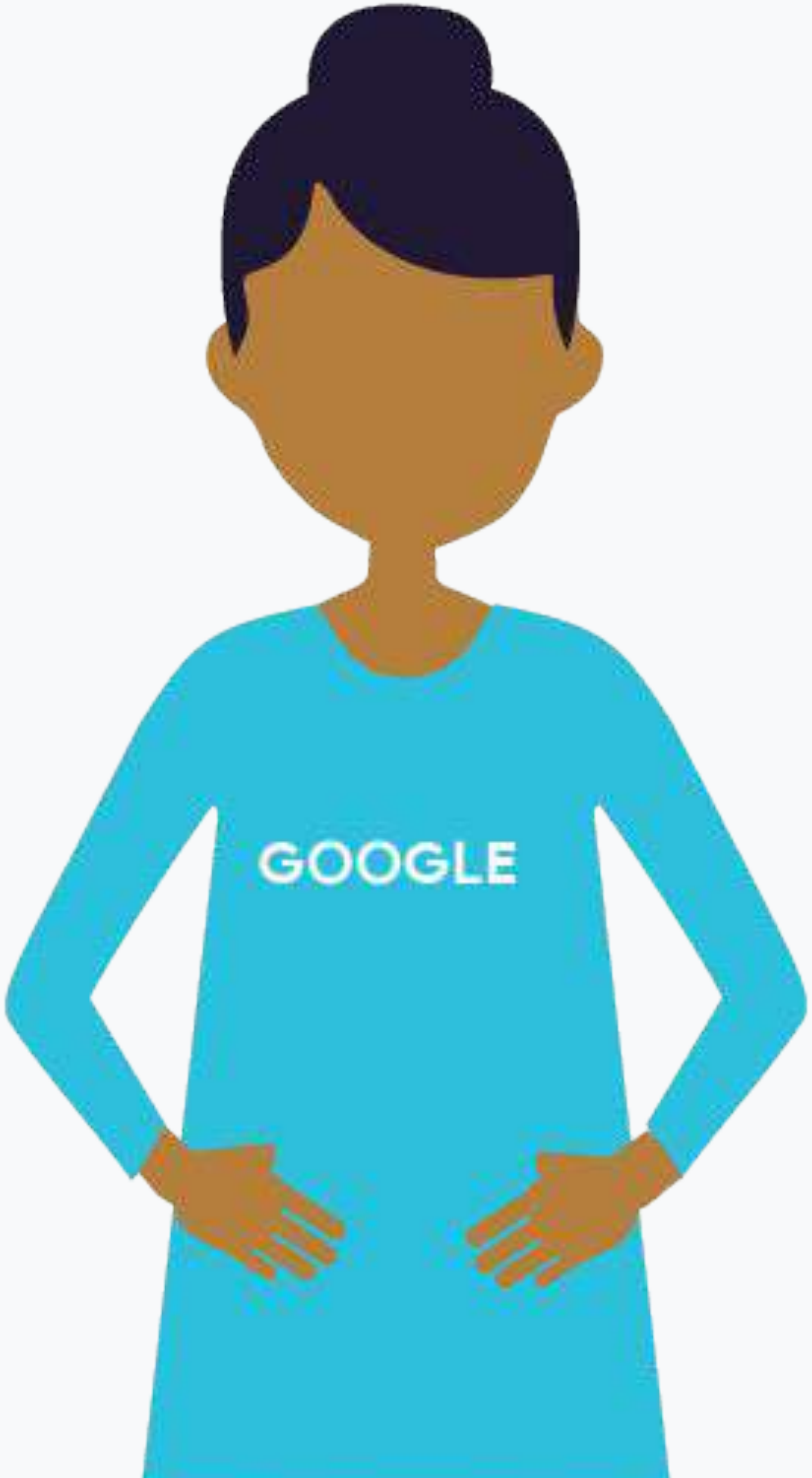
Production ML System Component: Serving





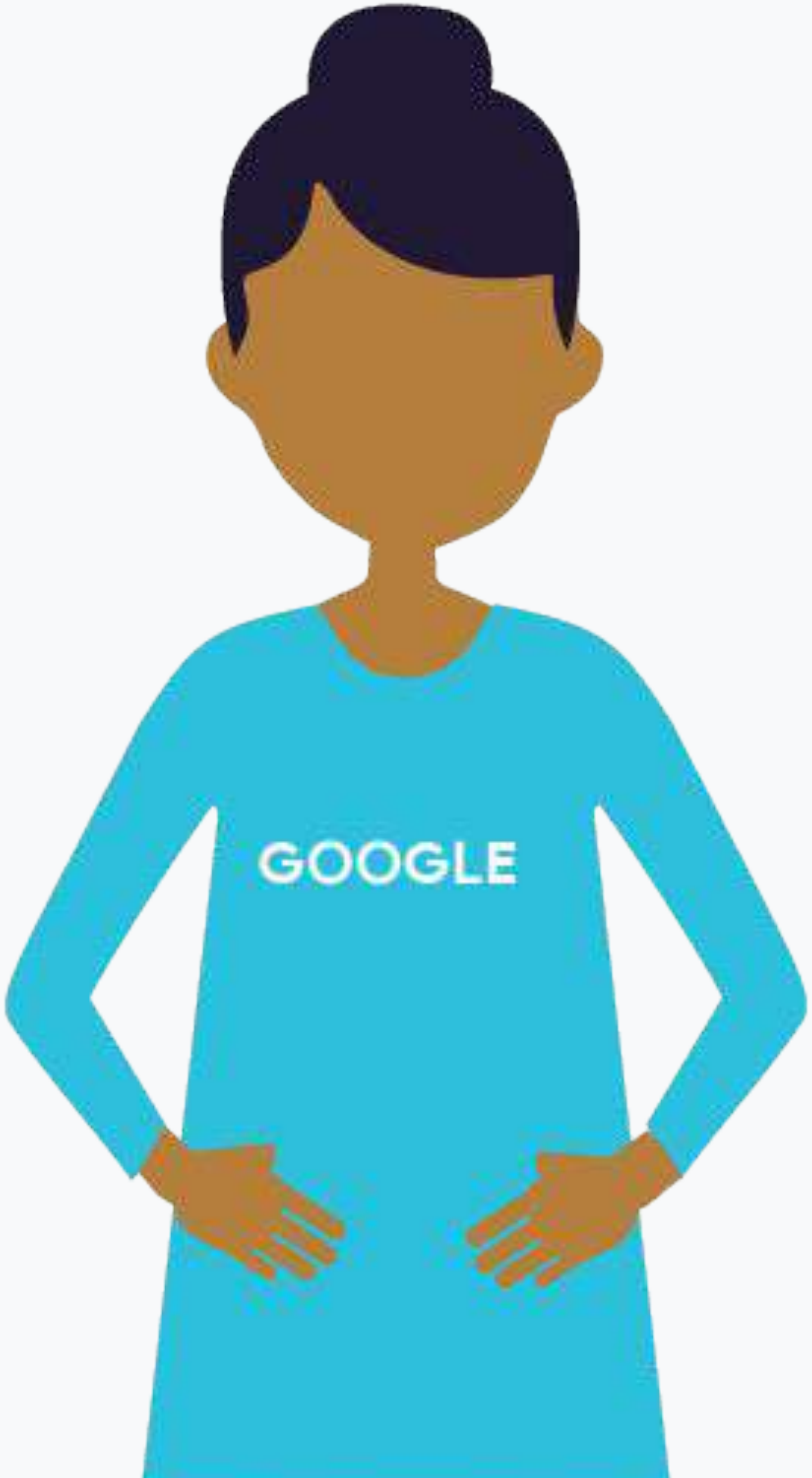
Serving Component must be:

- Low-latency



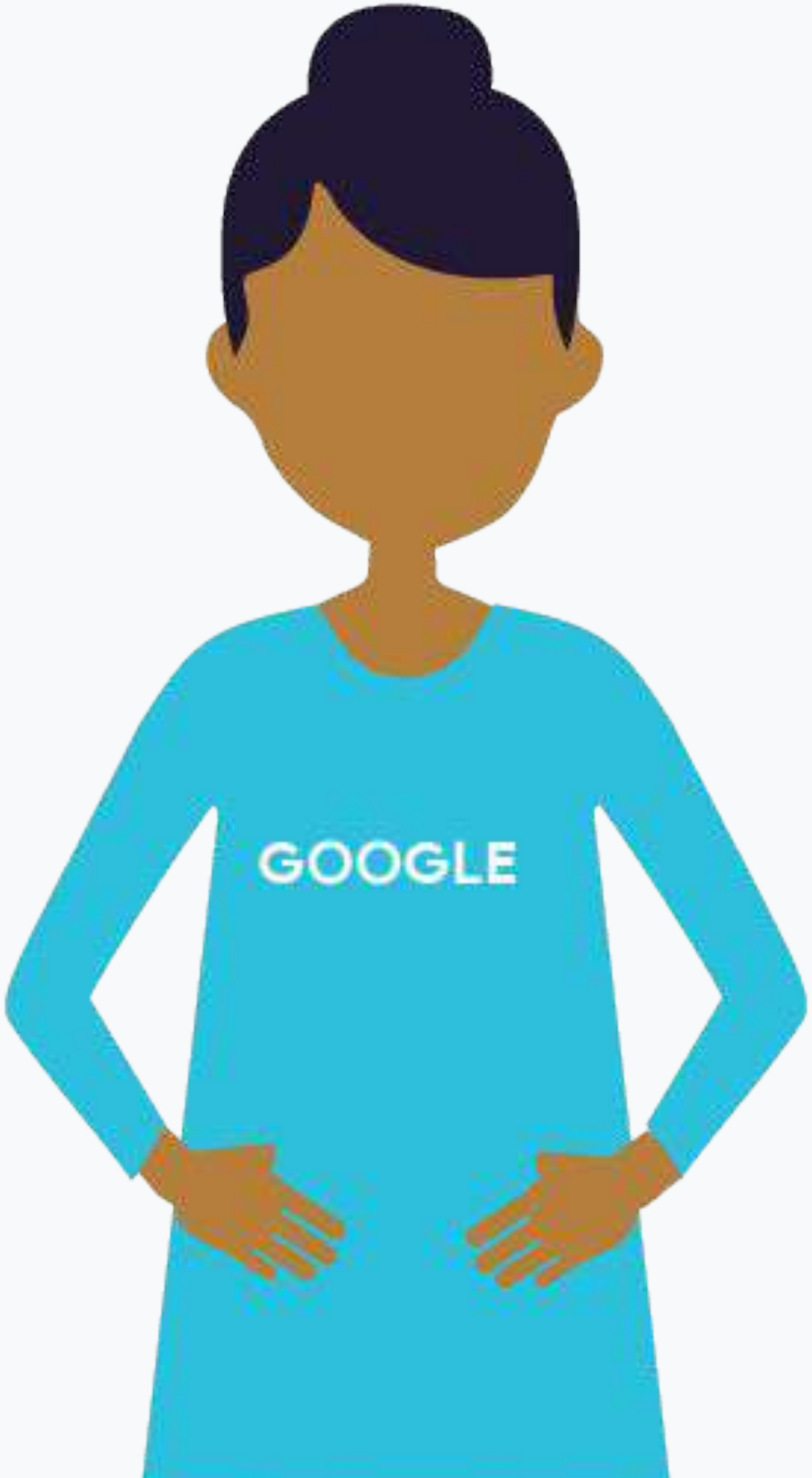
Serving Component must be:

- Low-latency
- Highly efficient



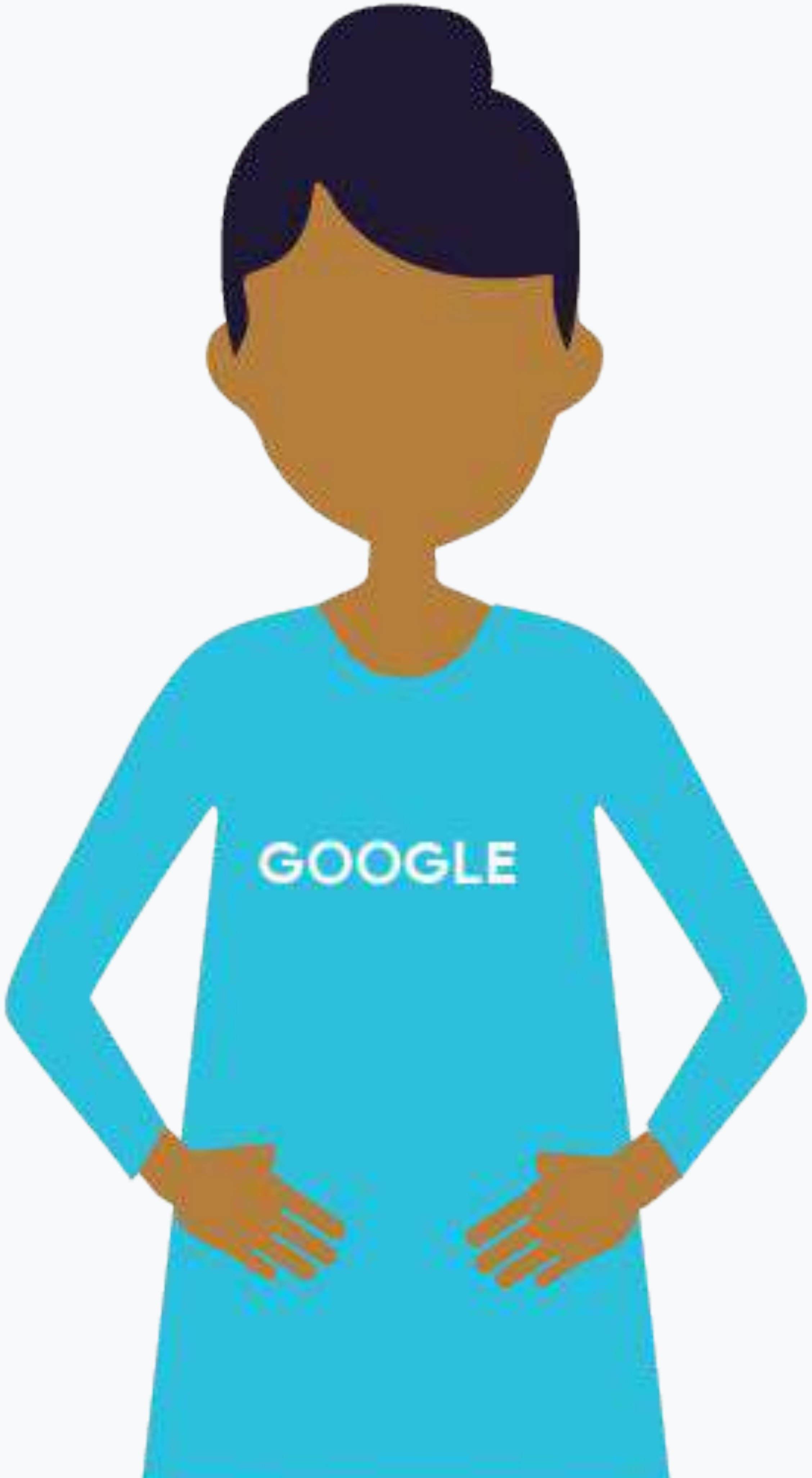
Serving Component must be:

- Low-latency
- Highly efficient
- Scale Horizontally



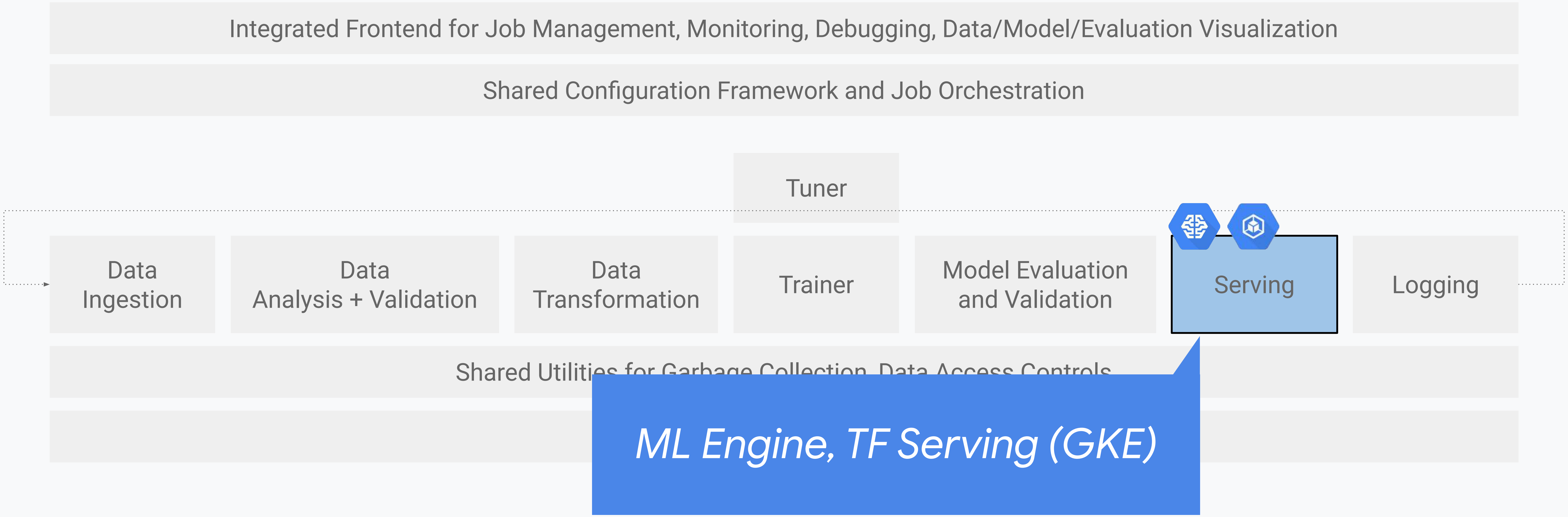
Serving Component must be:

- Low-latency
- Highly efficient
- Scale Horizontally
- Reliable and robust



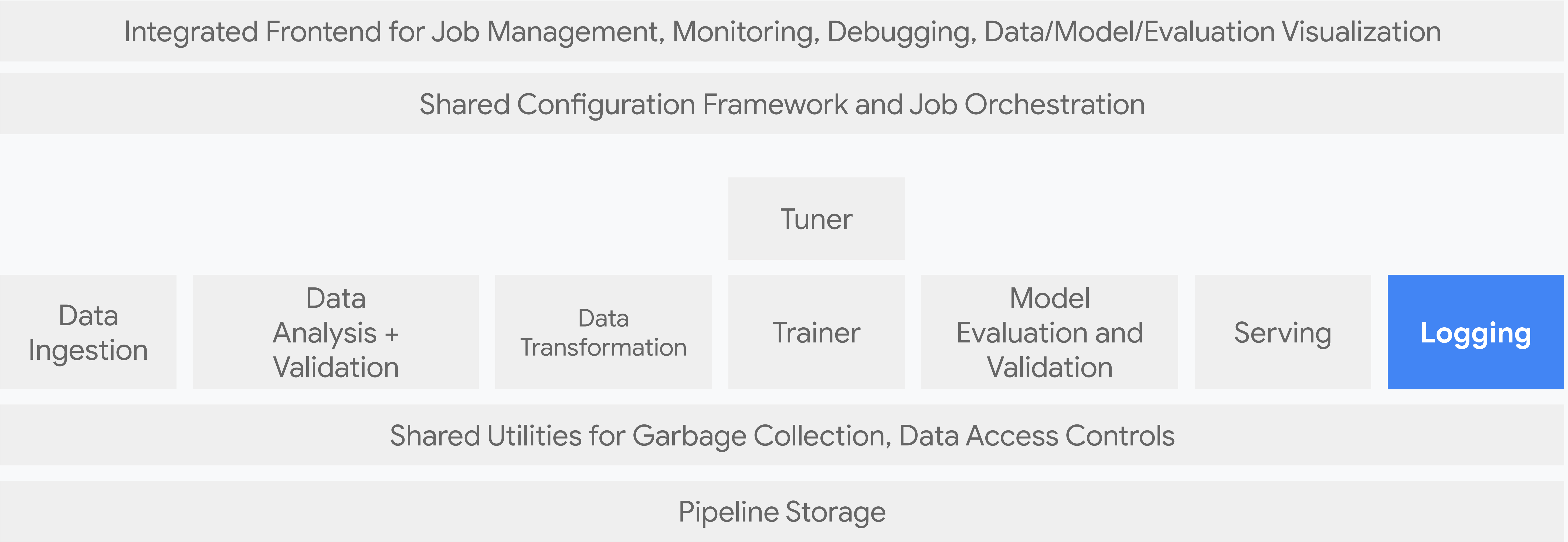
Serving Component
must be:

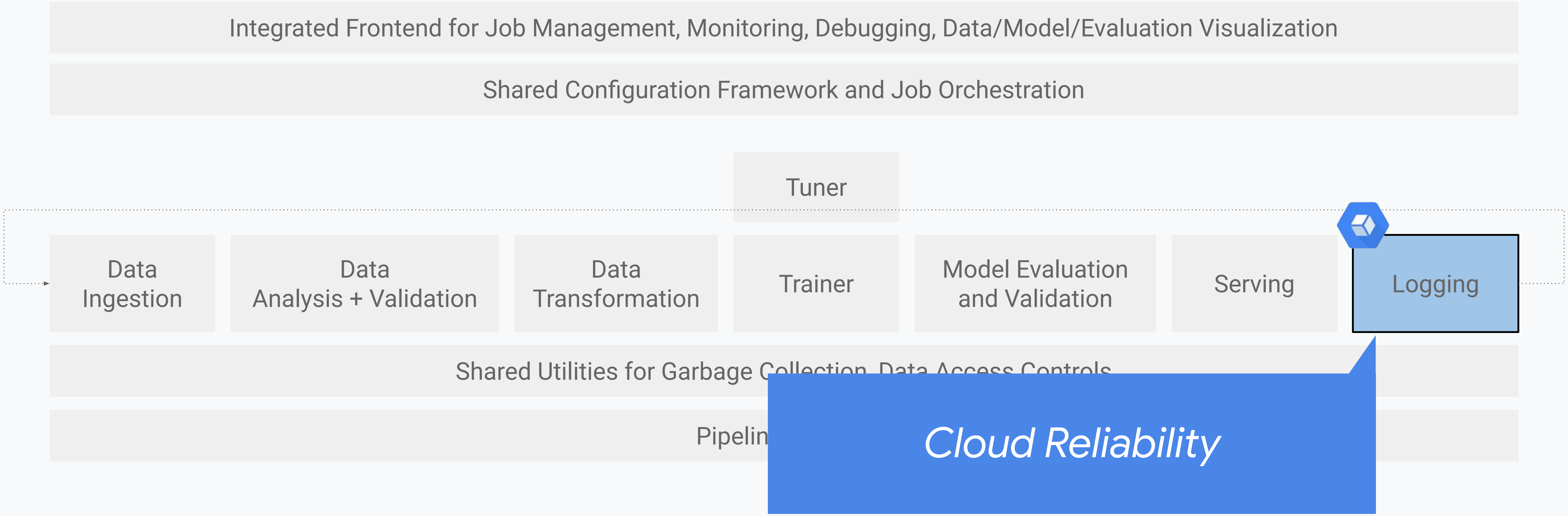
- Low-latency
- Highly efficient
- Scale Horizontally
- Reliable and robust
- Easy to update versions



High-level component overview of a machine learning platform.

Production ML System Component: Logging





High-level component overview of a machine learning platform.

Course 2: Production ML Systems

Module 1: Architecting Production ML Systems

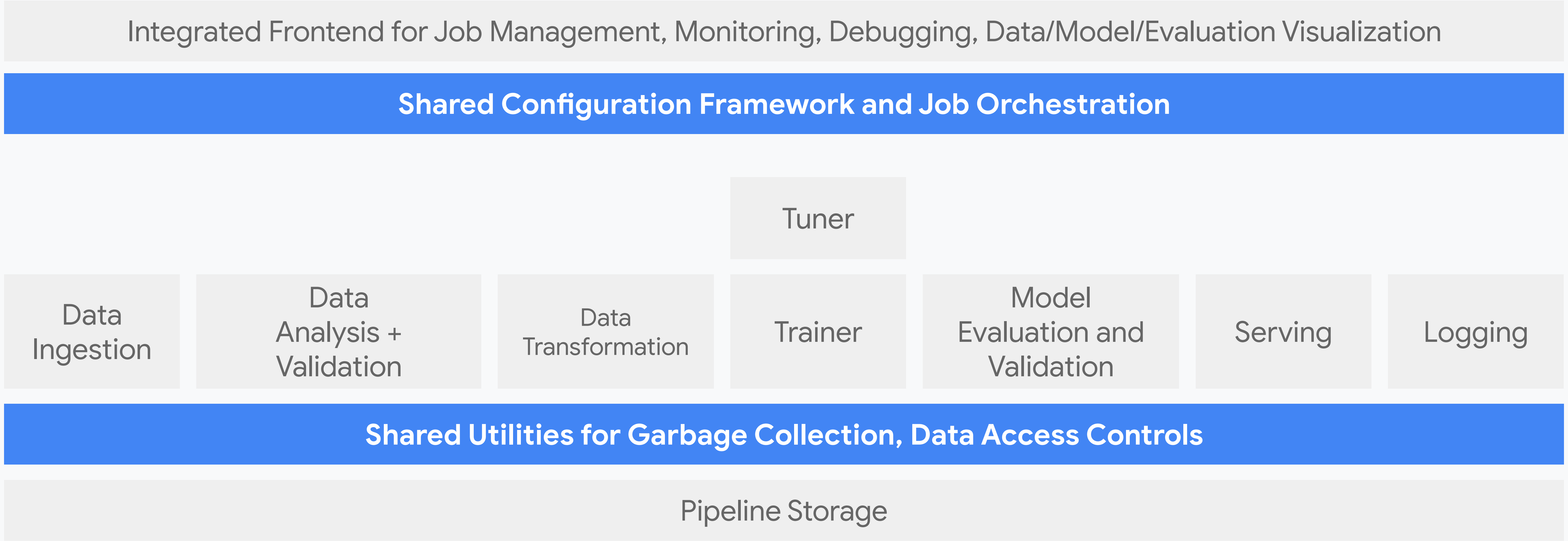
Lesson Title: **The Components of an ML System: Orchestration + Workflow**

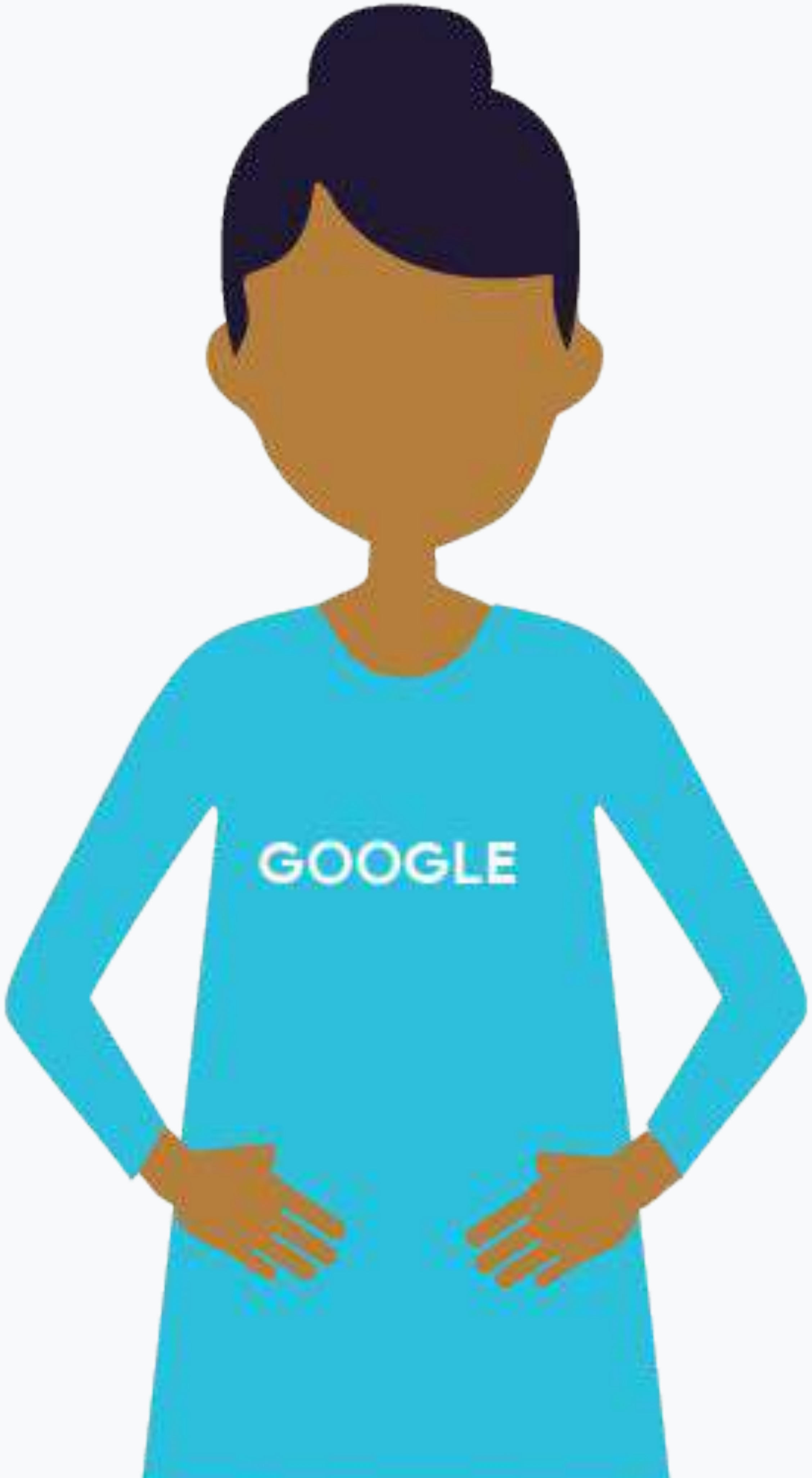
Presenter: Max Lotstein

Format: Talking Head

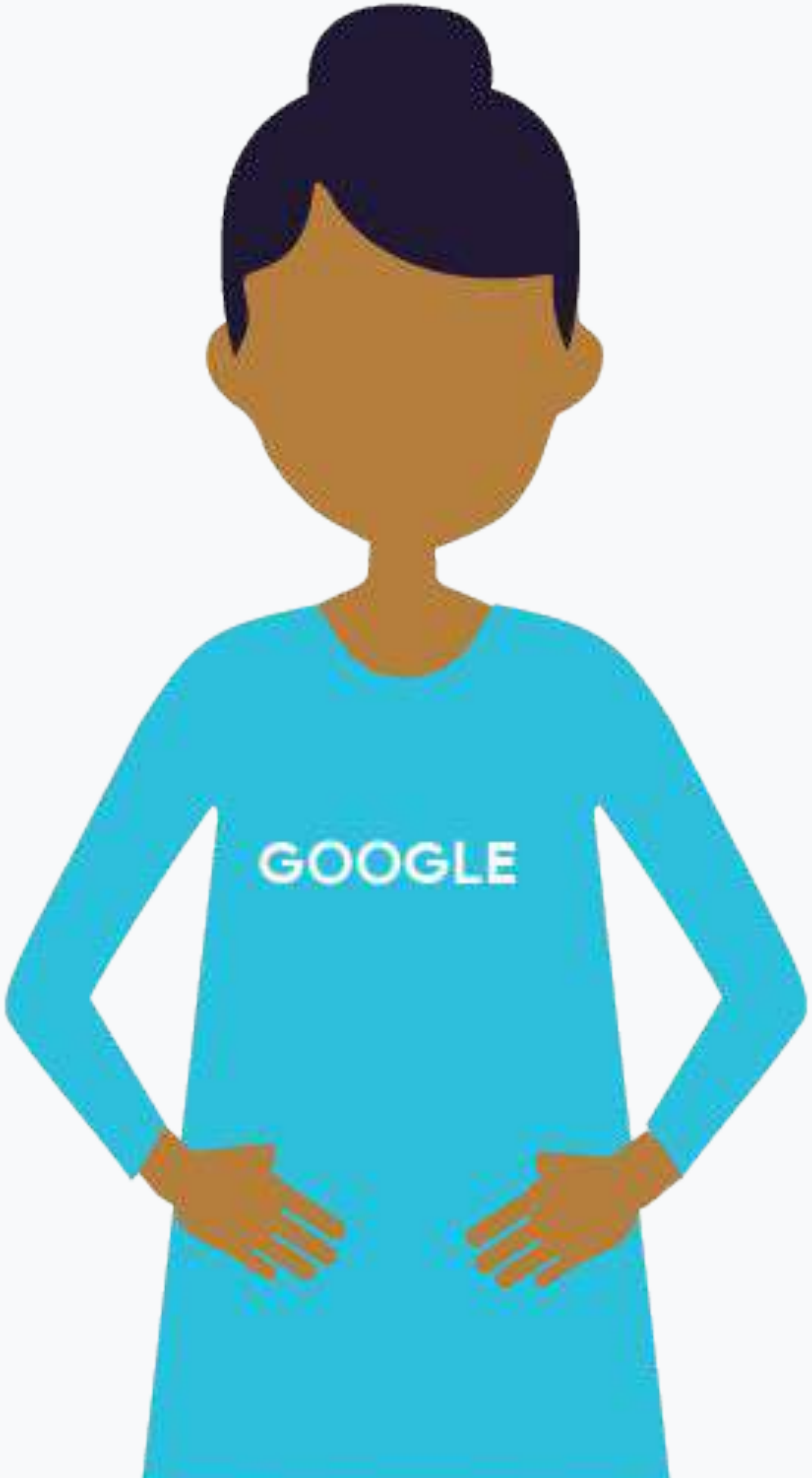
Video Name: T-PSML-O_1_l8_the_components_of_an_ml_system:_orchestration_+_workflow

Production ML System Component: Shared Config and Utilities





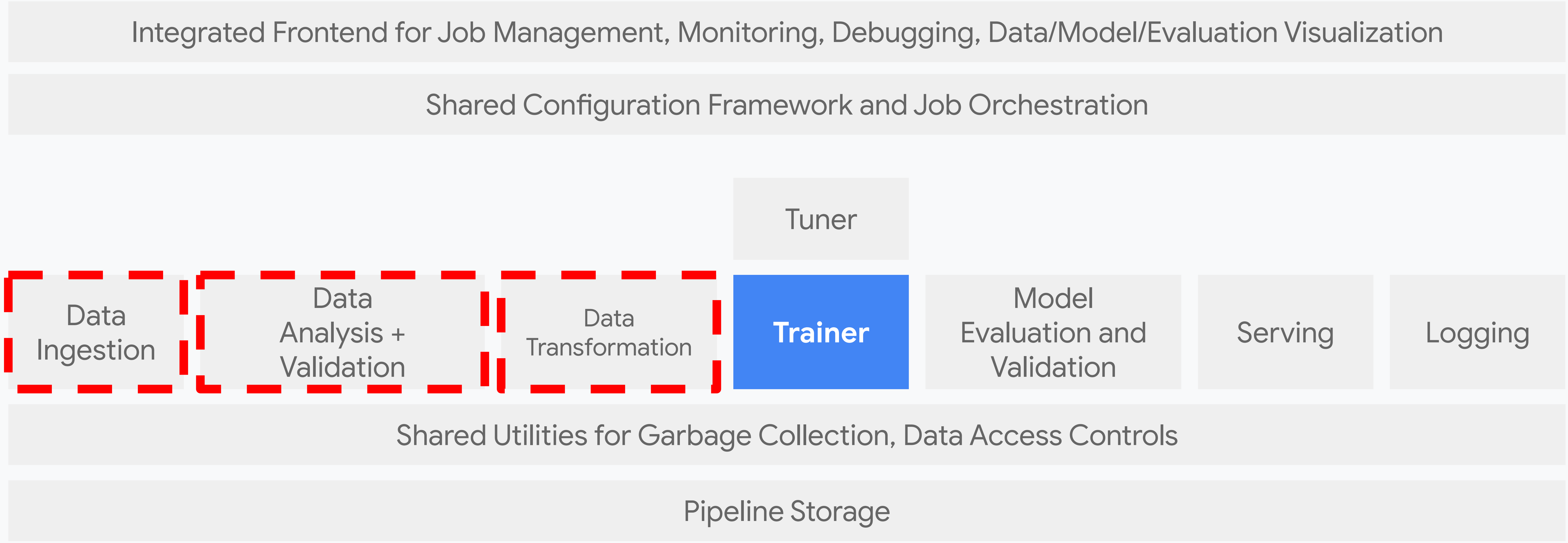
Quiz: If changes are made to the trainer, what component(s) might also need to change?



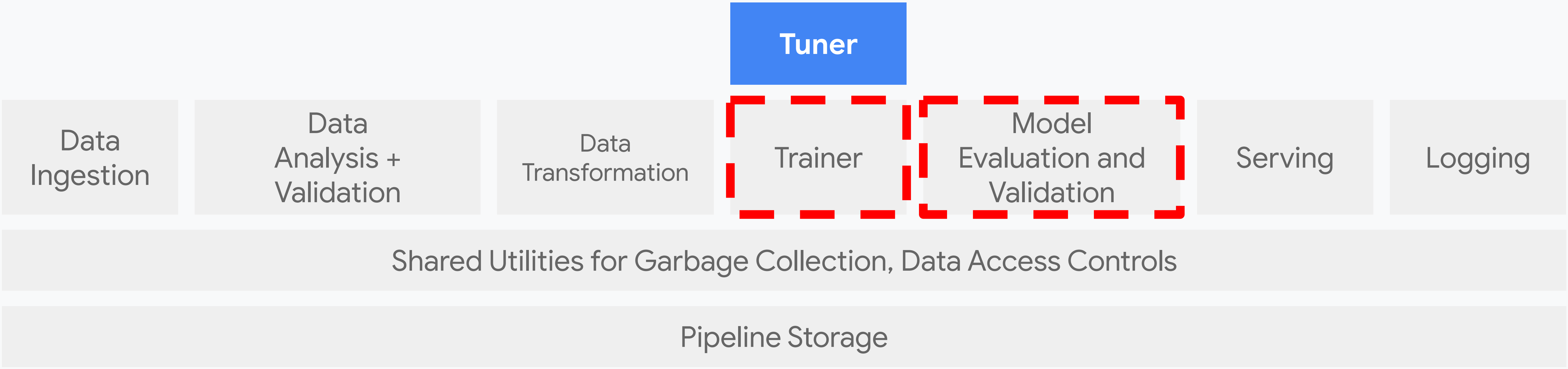
Answer:

Potentially all of them

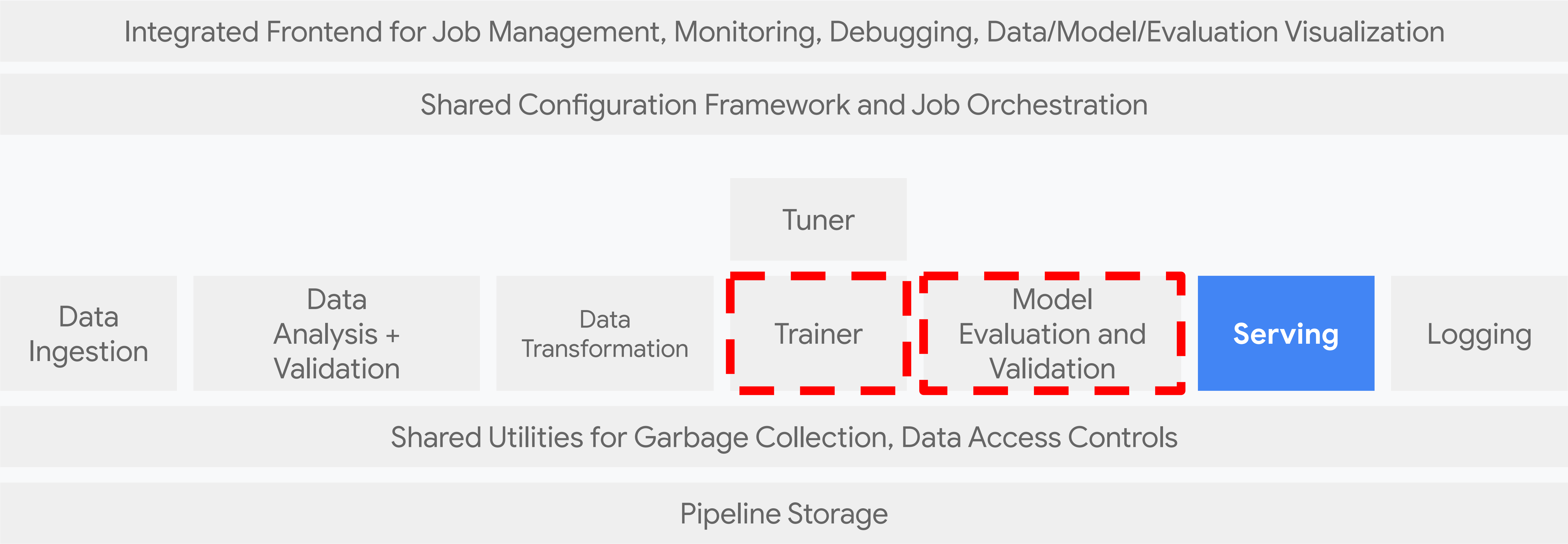
Changing Anything Changes Almost Everything



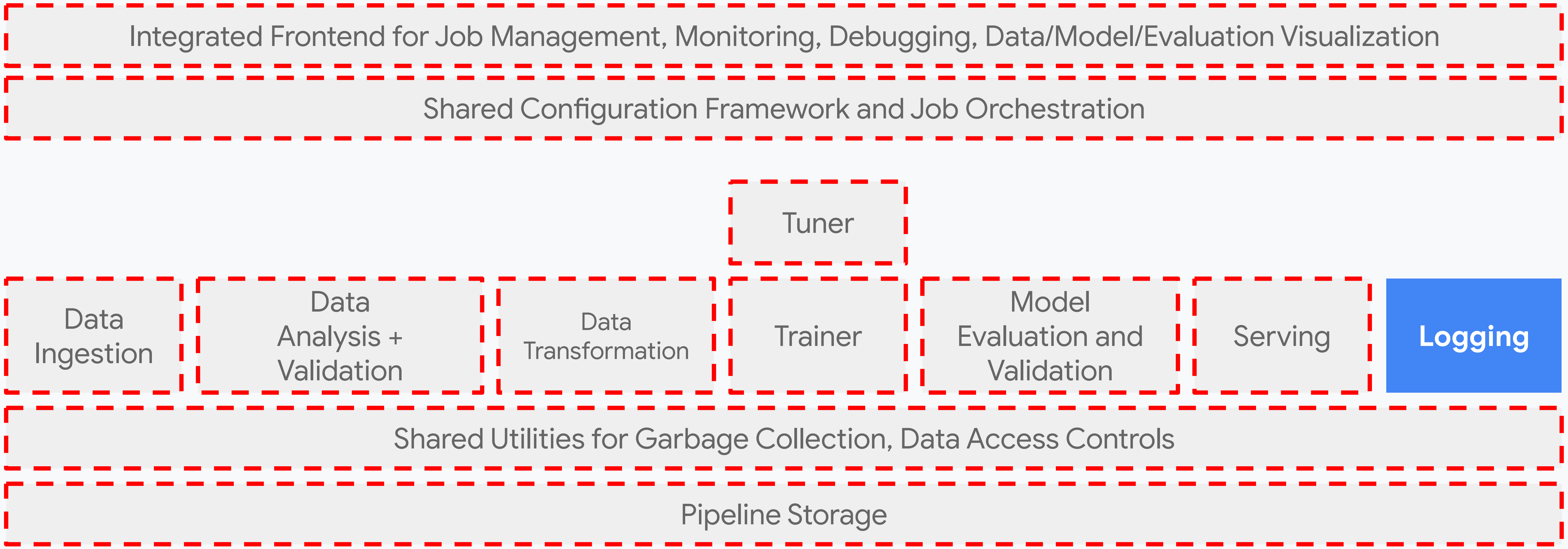
Changing Anything Changes Almost Everything

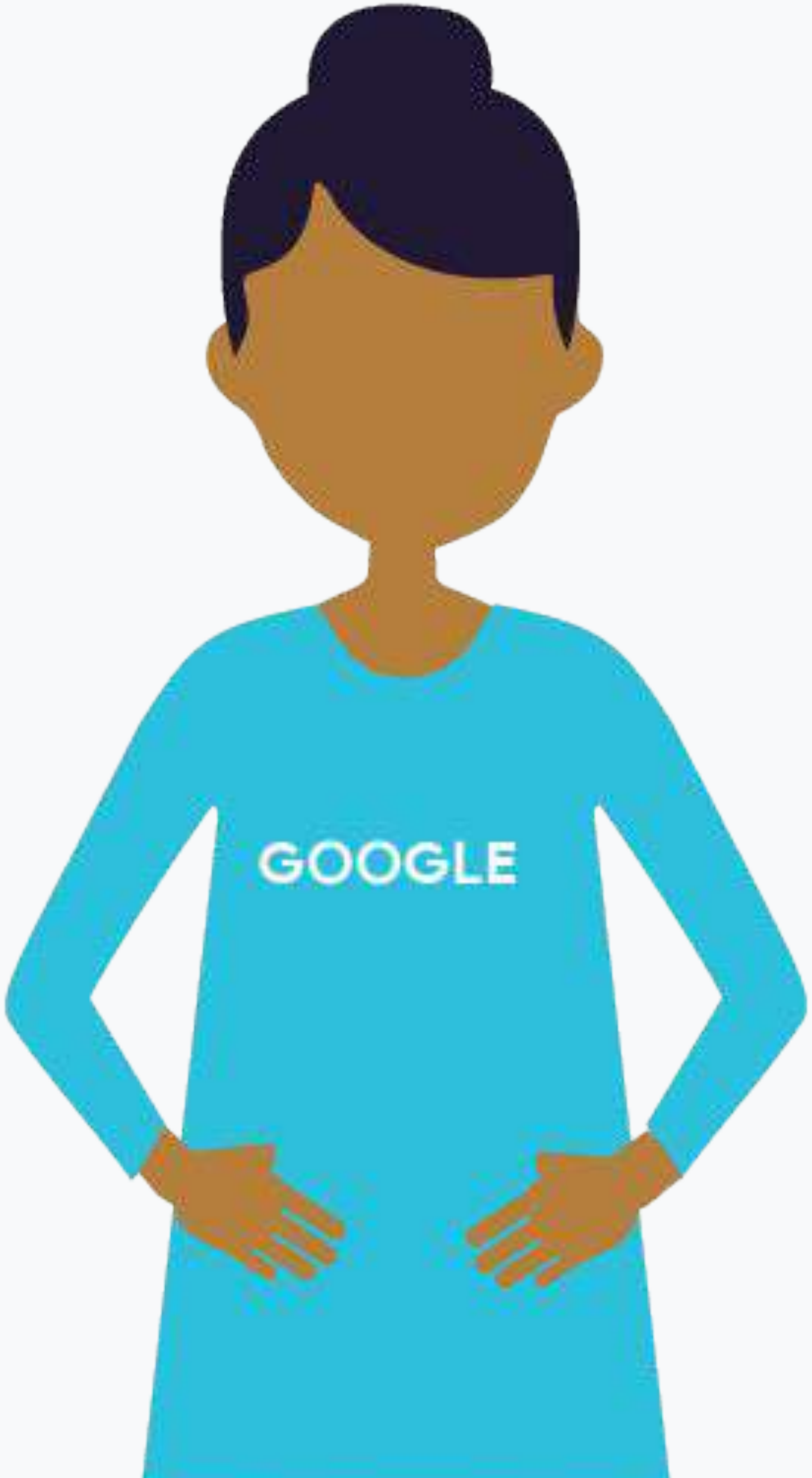


Changing Anything Changes Almost Everything

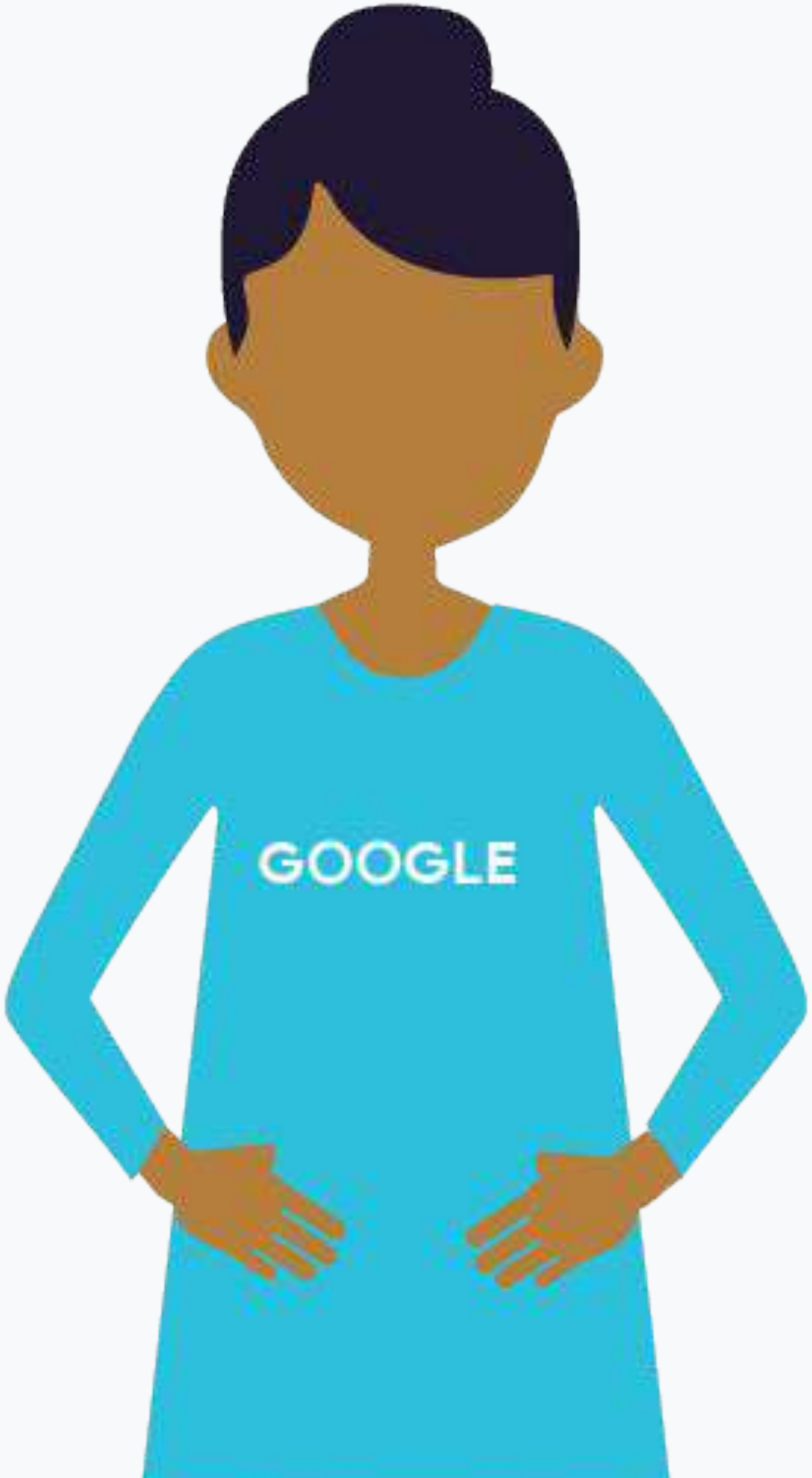


Changing Anything Changes Almost Everything

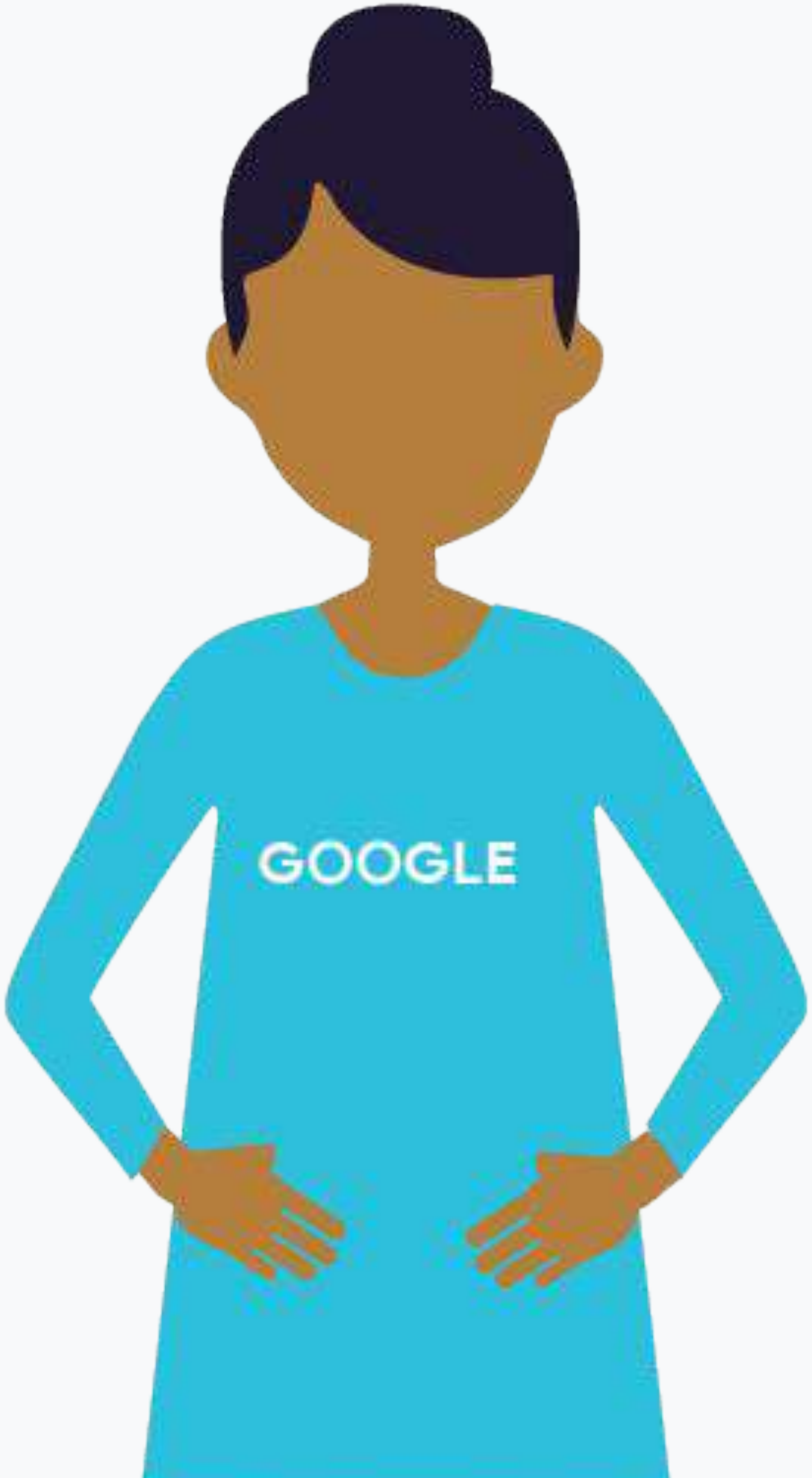




Configuration: A Potential Source of Debt

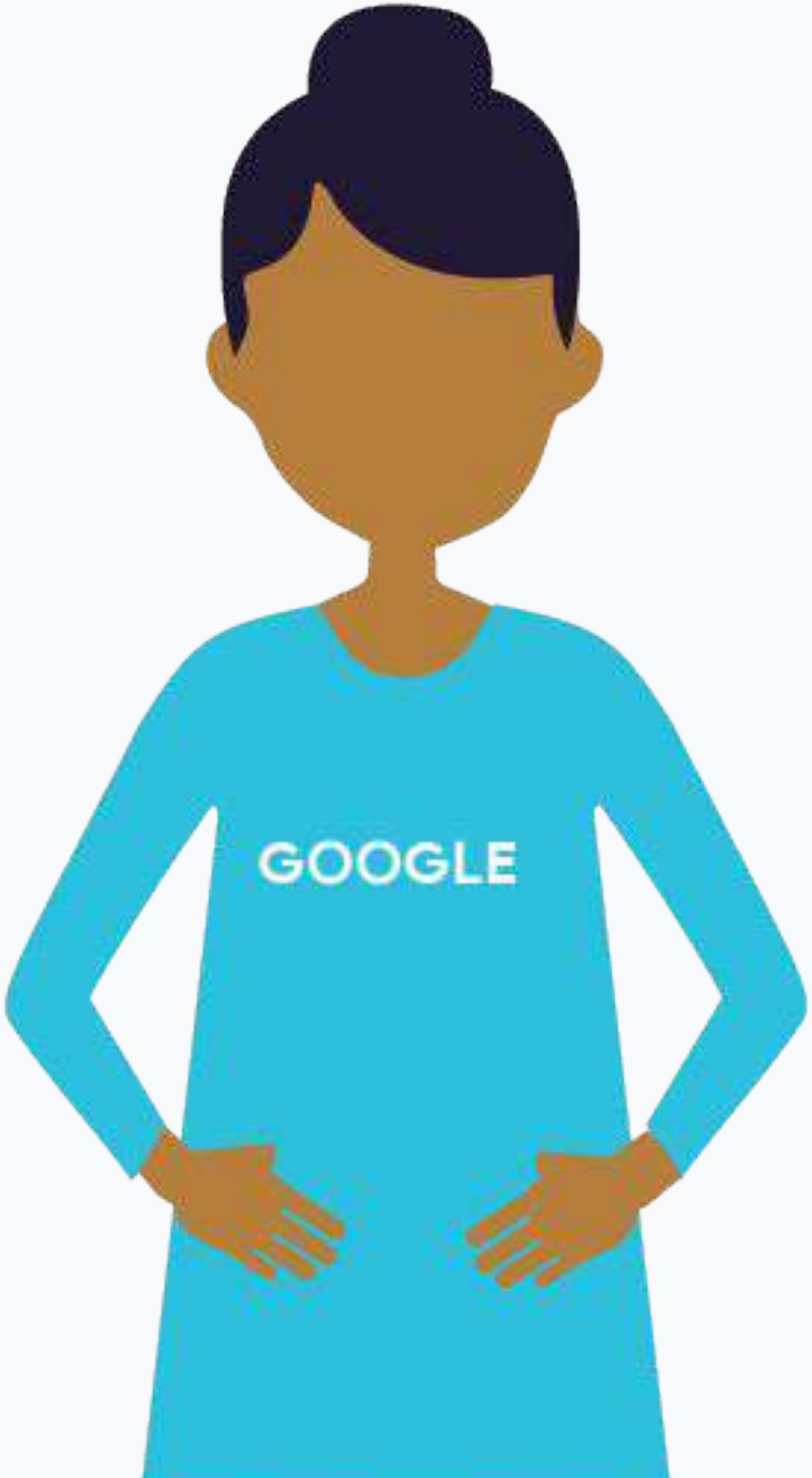


Configuration Remedies



Configuration Remedies

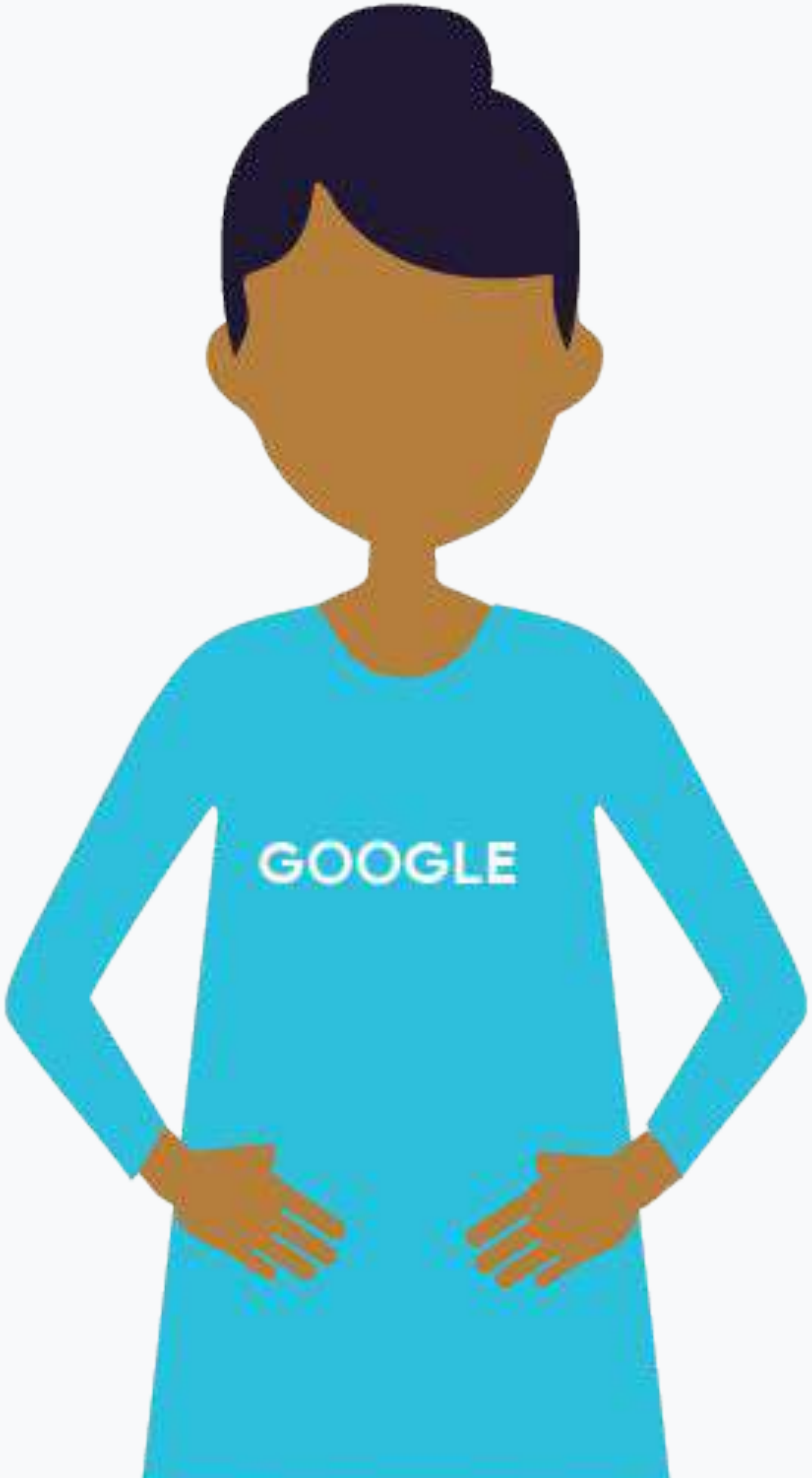
1) Establish a common architecture for both R&D and production deployment



Configuration Remedies

1) Establish a common architecture for both R&D and production deployment

2) Embed the teams together, so that engineering can influence the design of code from its inception



Orchestration glues all the components together

Production ML System Component: Orchestration

Cloud Composer, Argo (GKE)

Integrated Frontend for Job Management, Monitoring, Debugging, Dat

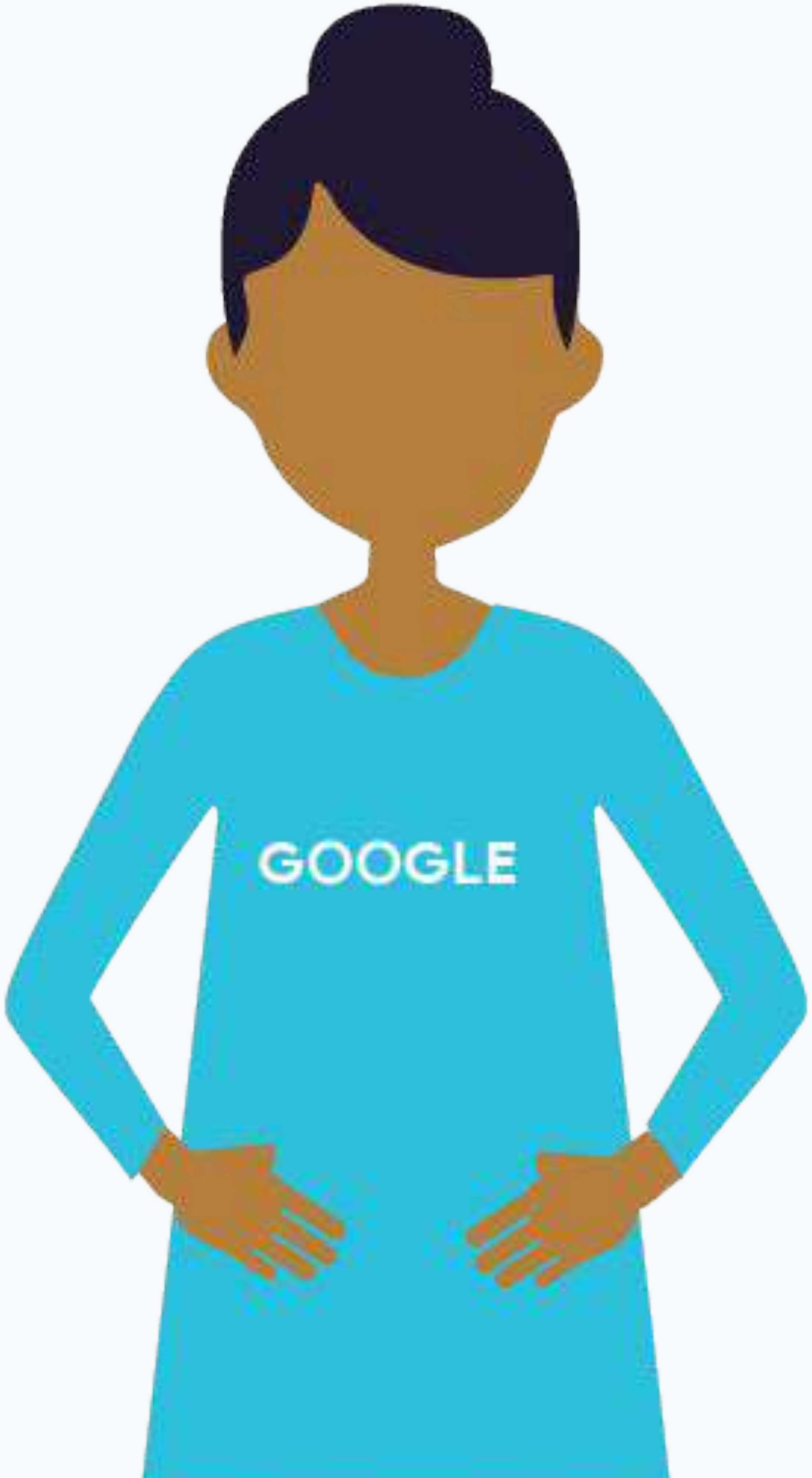
Shared Configuration Framework and Job Orchestration

Tuner

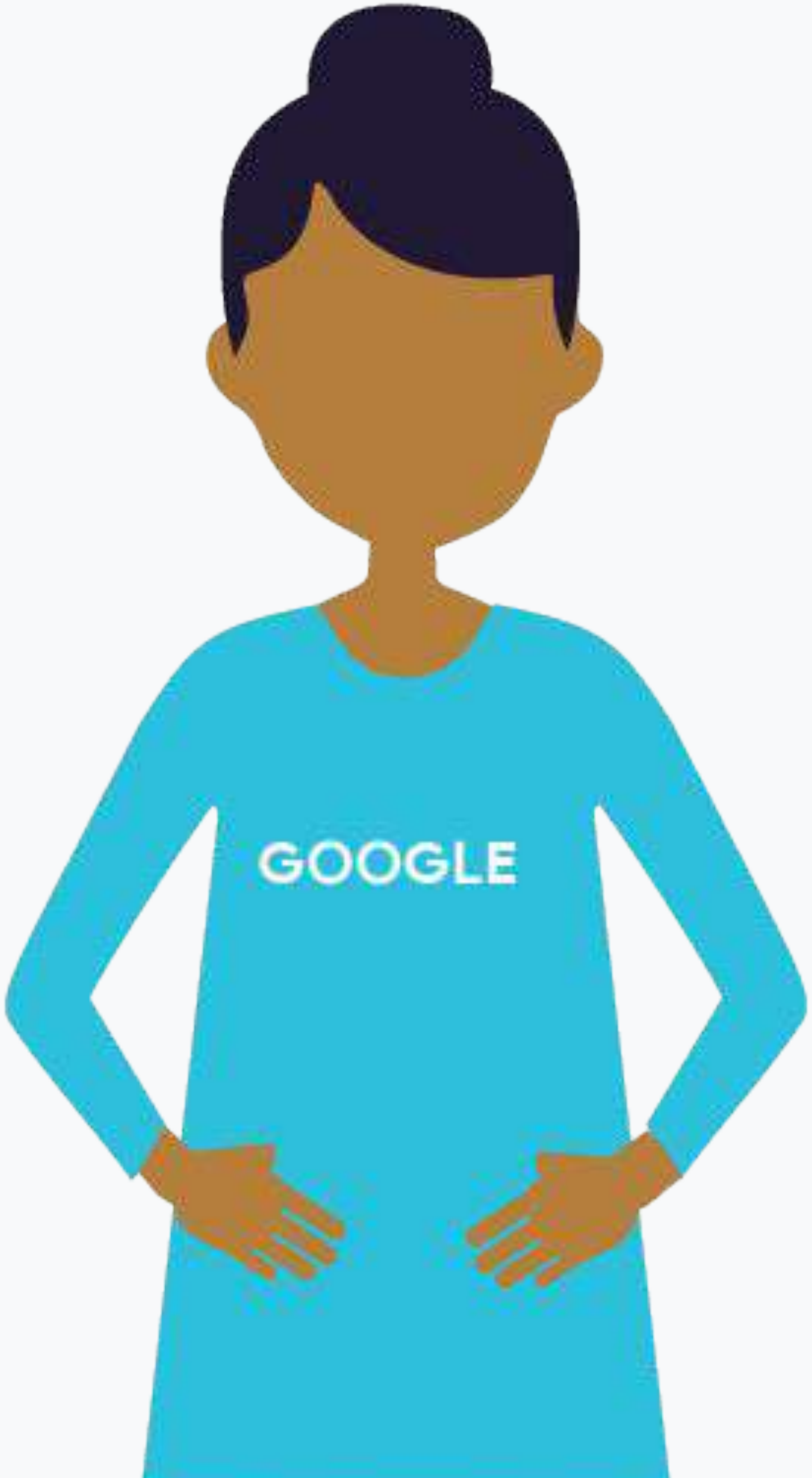
Data Ingestion	Data Analysis + Validation	Data Transformation	Trainer	Model Evaluation and Validation	Serving	Logging

Shared Utilities for Garbage Collection, Data Access Controls

Pipeline Storage

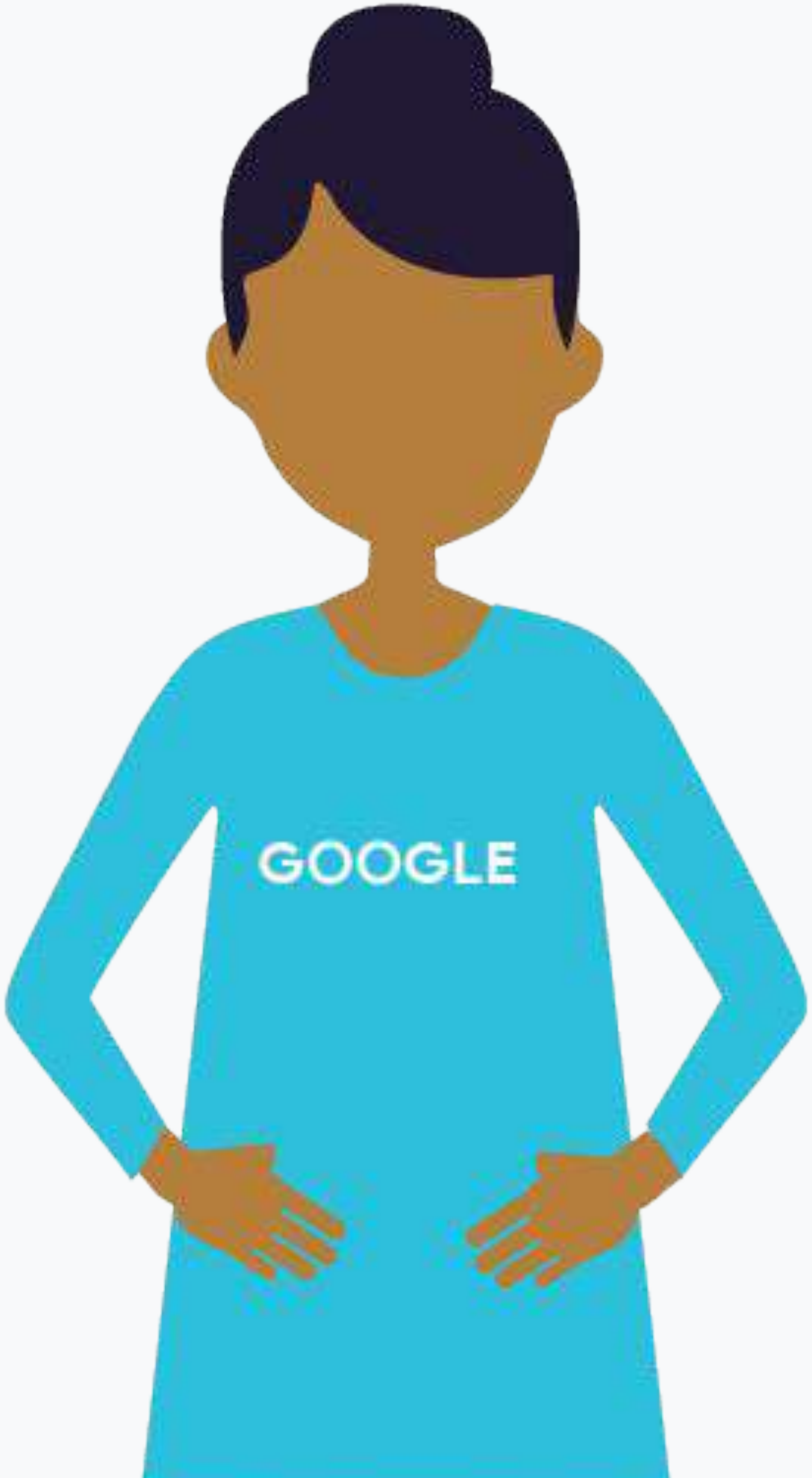


Steps to Compose a Workflow in Cloud Composer



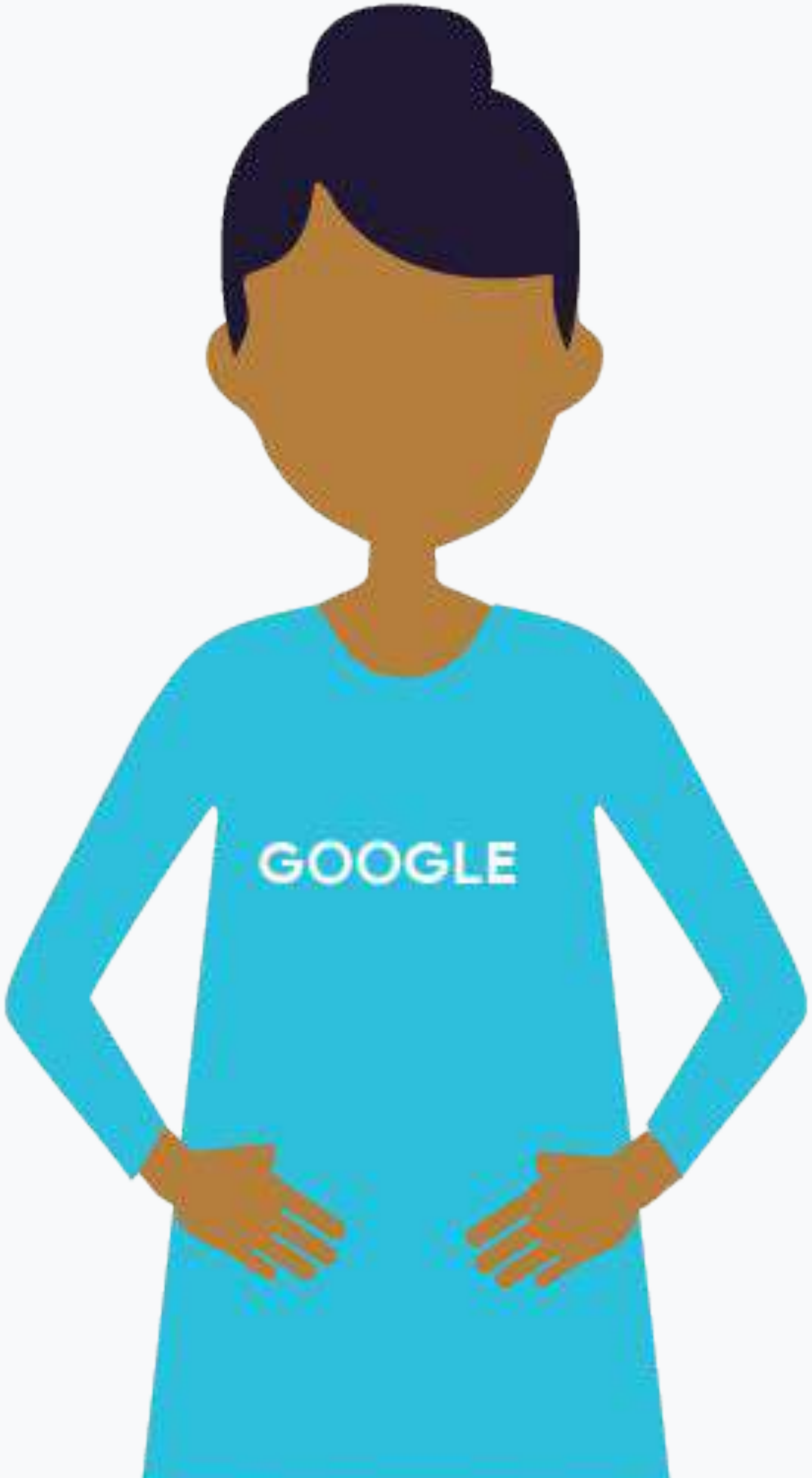
Steps to Compose a Workflow in Cloud Composer

1) Define the Ops



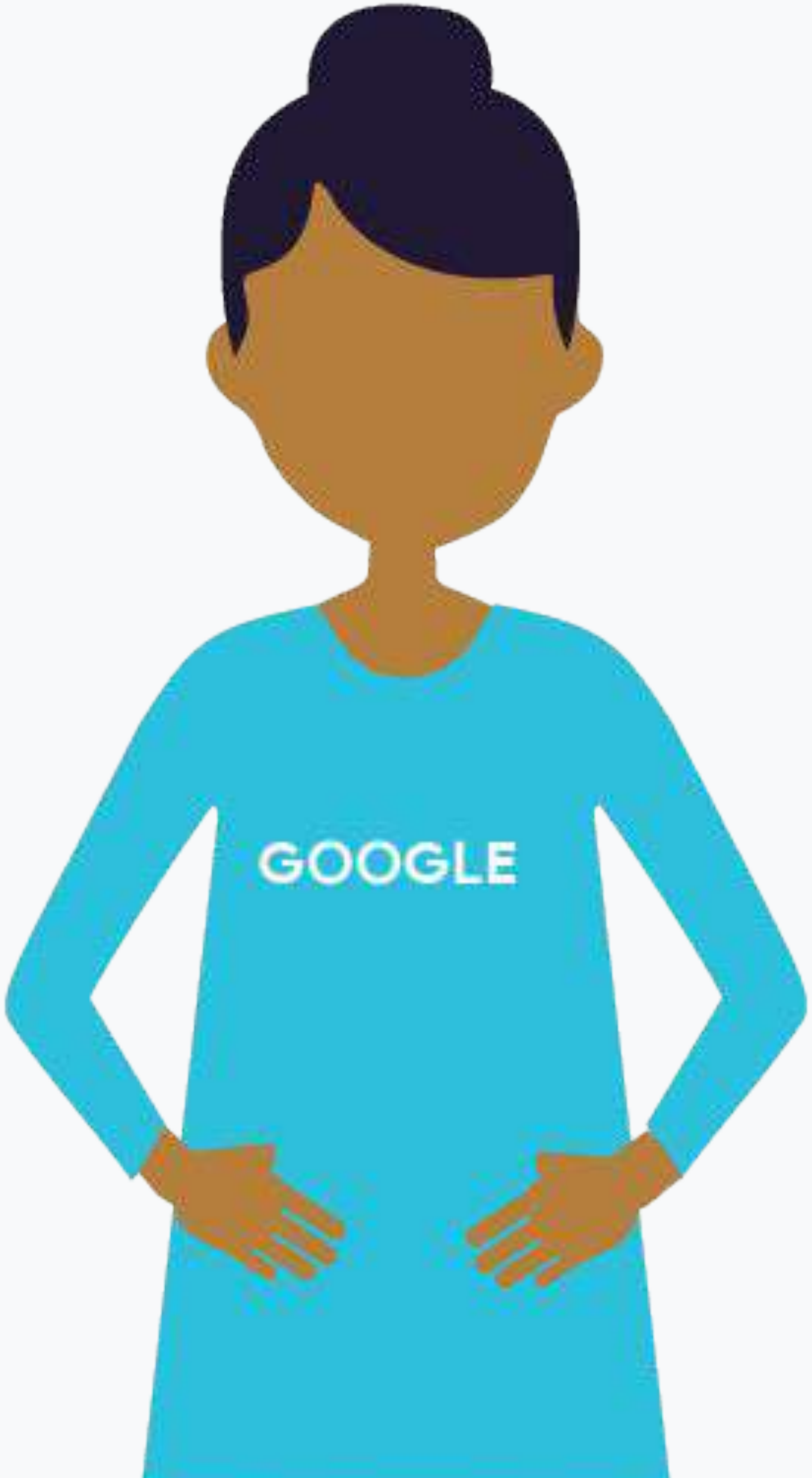
Steps to Compose a Workflow in Cloud Composer

1) Define the Ops
2) Arrange into a DAG



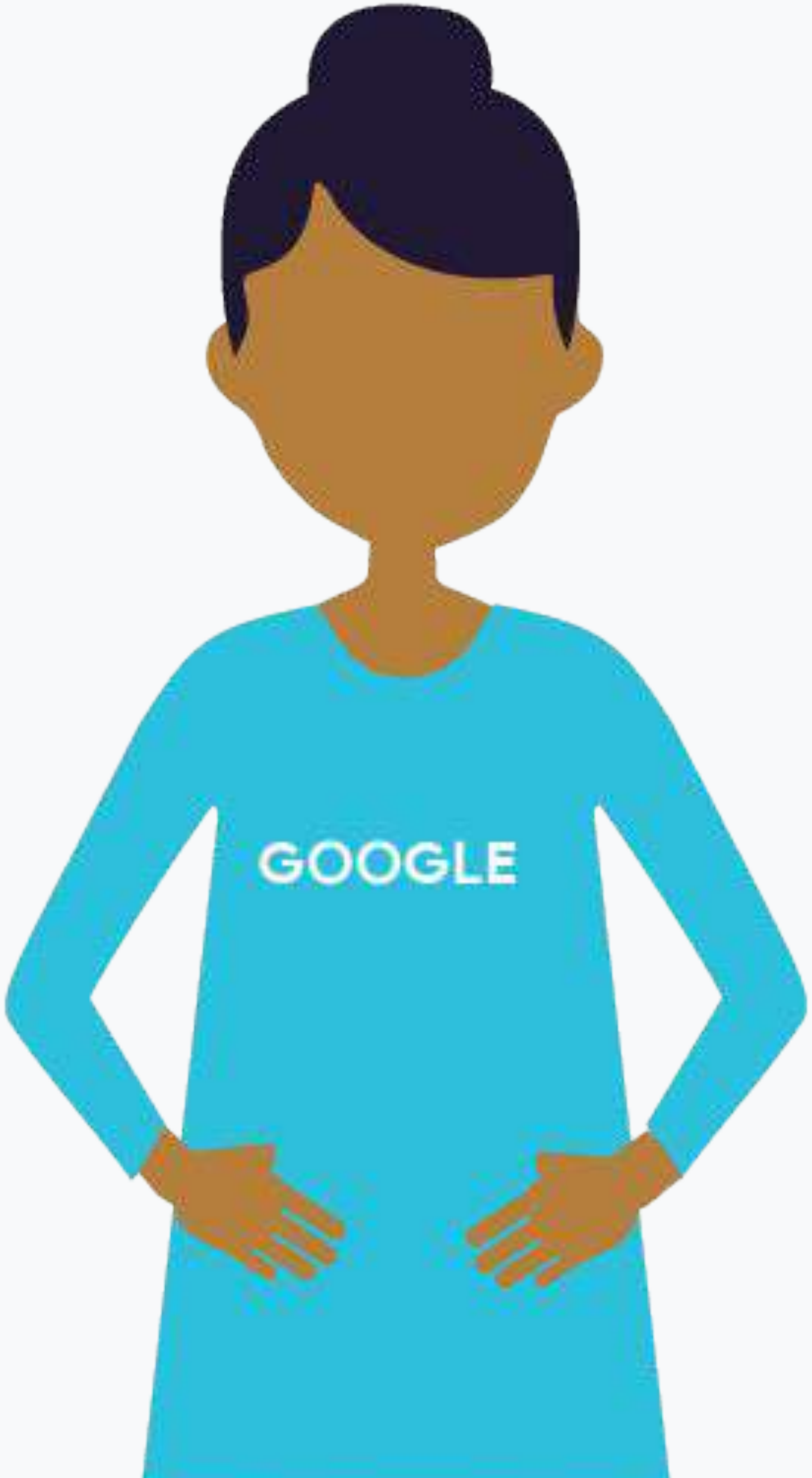
Steps to Compose a Workflow in Cloud Composer

1) Define the Ops
2) Arrange into a DAG
3) Upload to Environment



Steps to Compose a Workflow in Cloud Composer

1) Define the Ops
2) Arrange into a DAG
3) Upload to Environment
4) Explore DAG Run in Web UI



A basic workflow

```
# BigQuery training data query
t1 = BigQueryOperator(params)

# BigQuery training data export to GCS
t2 = BigQueryToCloudStorageOperator(params)

# ML Engine training job
t3 = MLEngineTrainingOperator(params)

# App Engine deploy new version
t4 = AppEngineVersionOperator(params)

# Establish dependencies
t1 >> t2 >> t3 >> t4
```

Course 2: Production ML Systems

Module 1: Architecting Production ML Systems

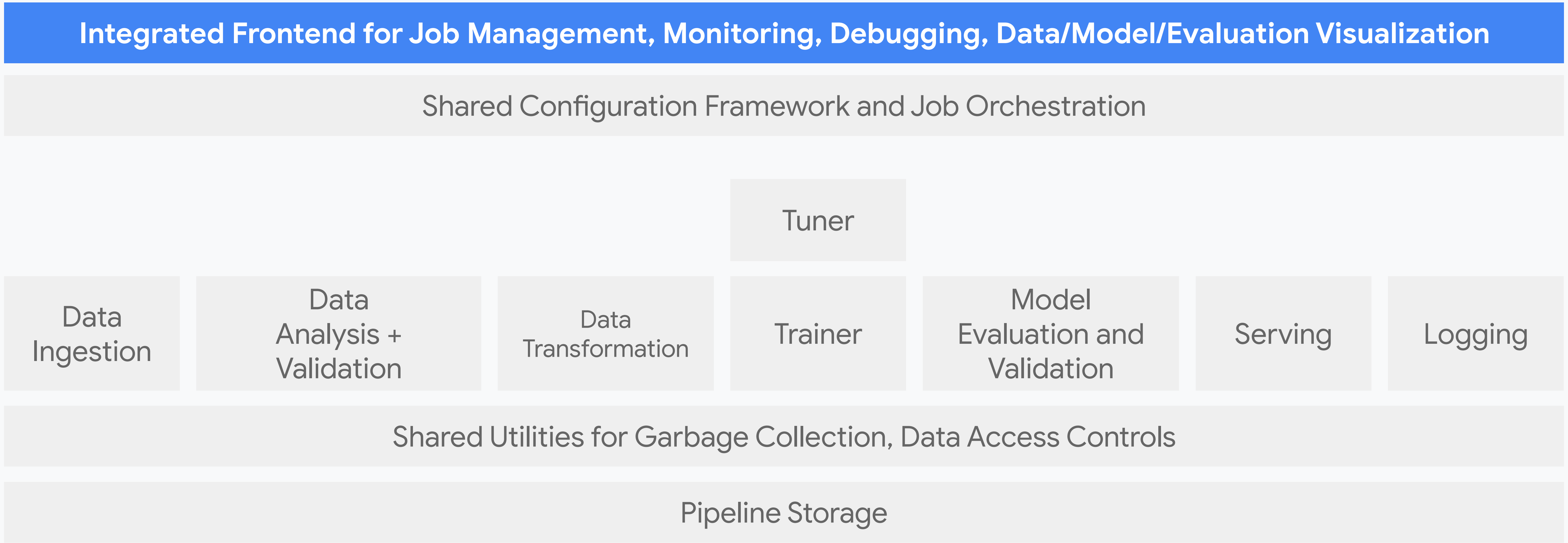
Lesson Title: **The Components of an ML System: Integrated Frontend + Storage**

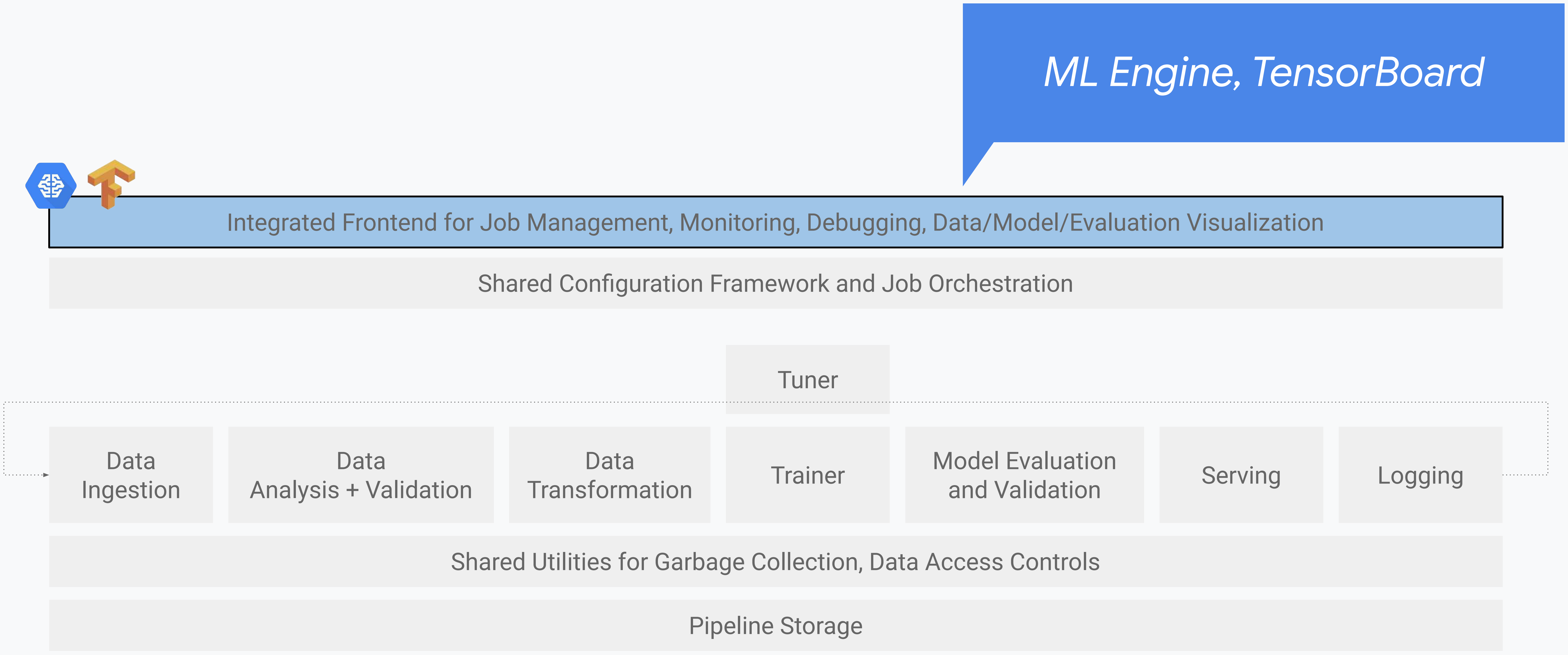
Presenter: Max Lotstein

Format: Talking Head

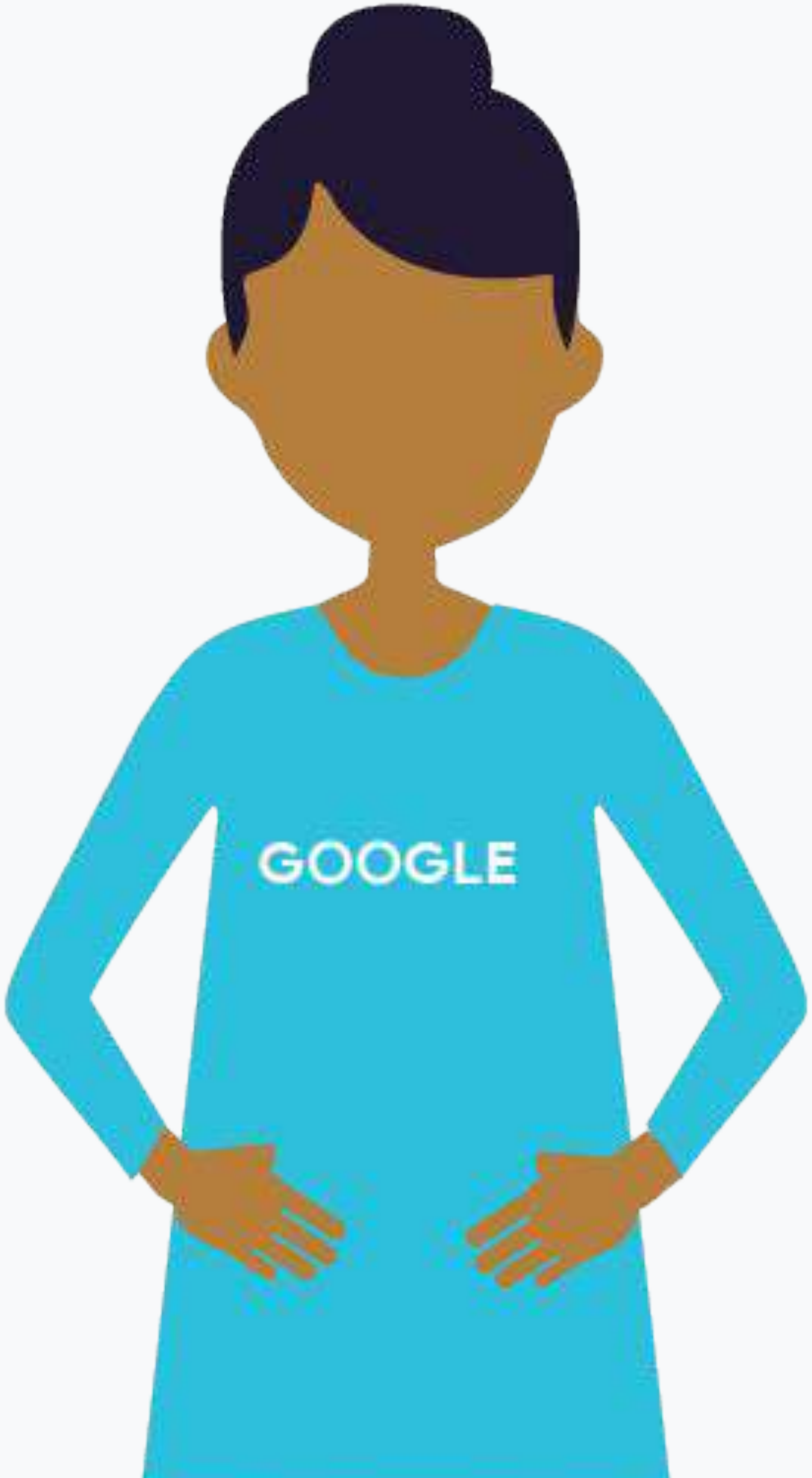
Video Name: T-PSML-O_1_l9_the_components_of_an_ml_system:_integrated_frontend_+_storage

Production ML System Component: Integrated Frontend

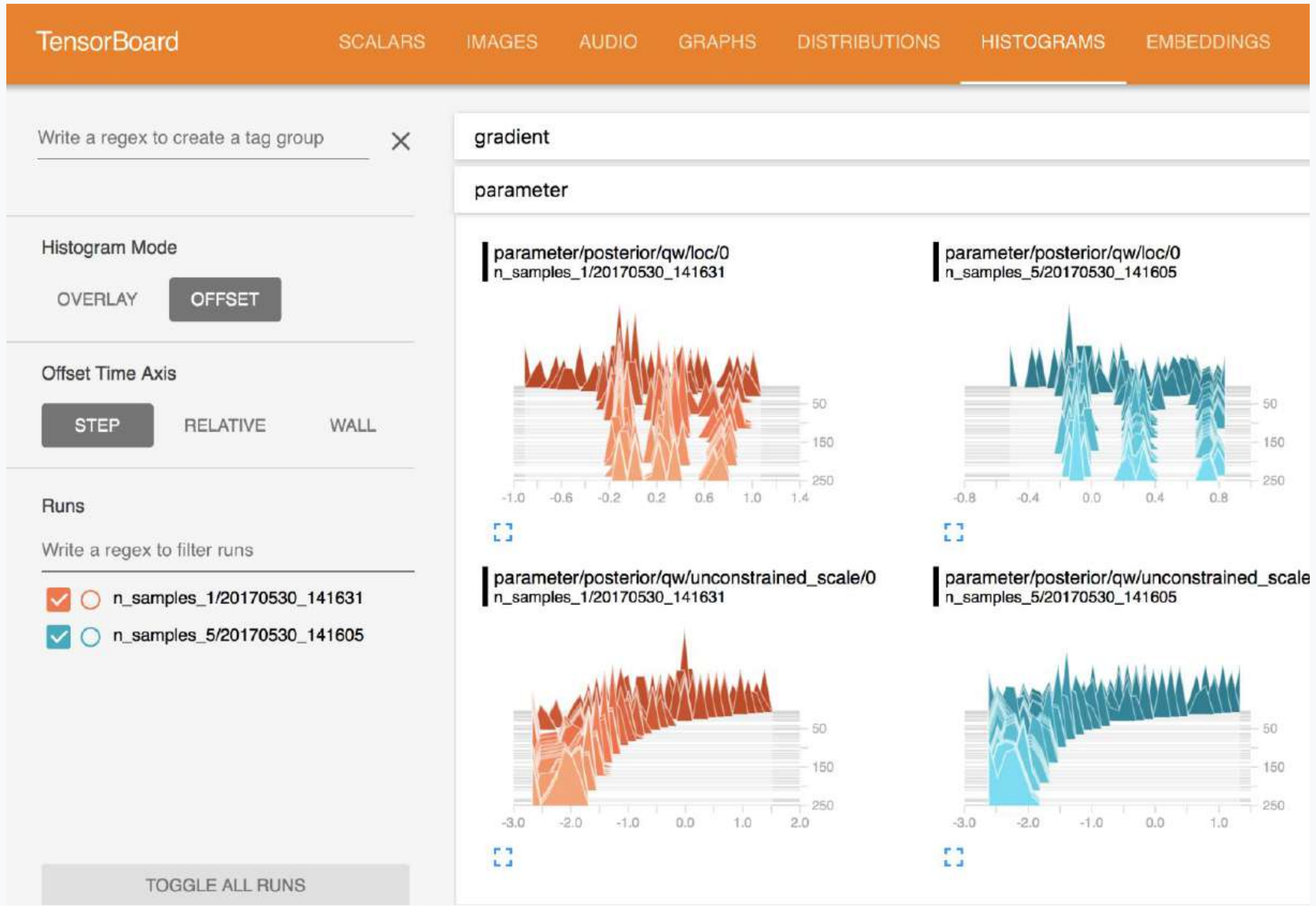


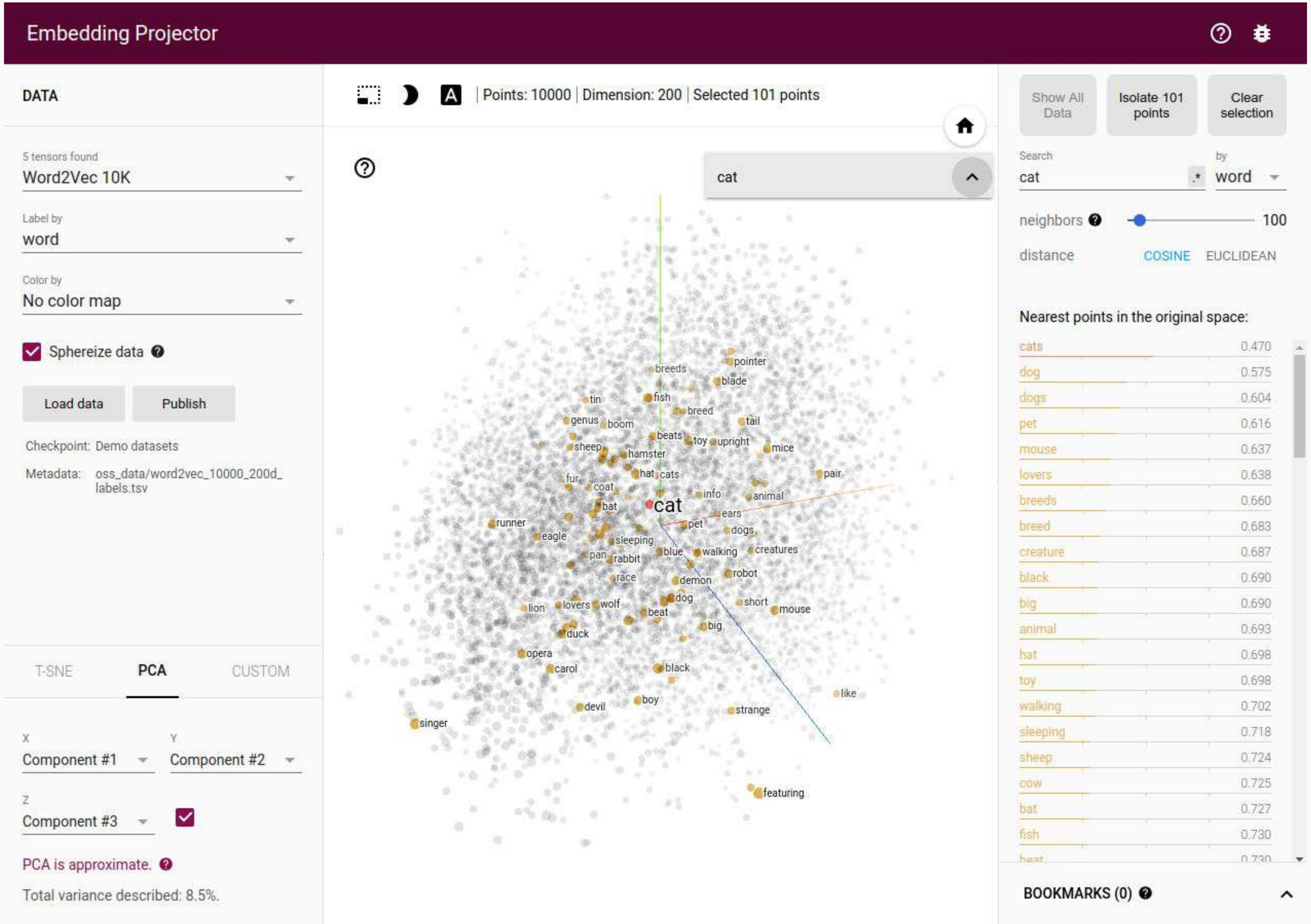


High-level component overview of a machine learning platform.

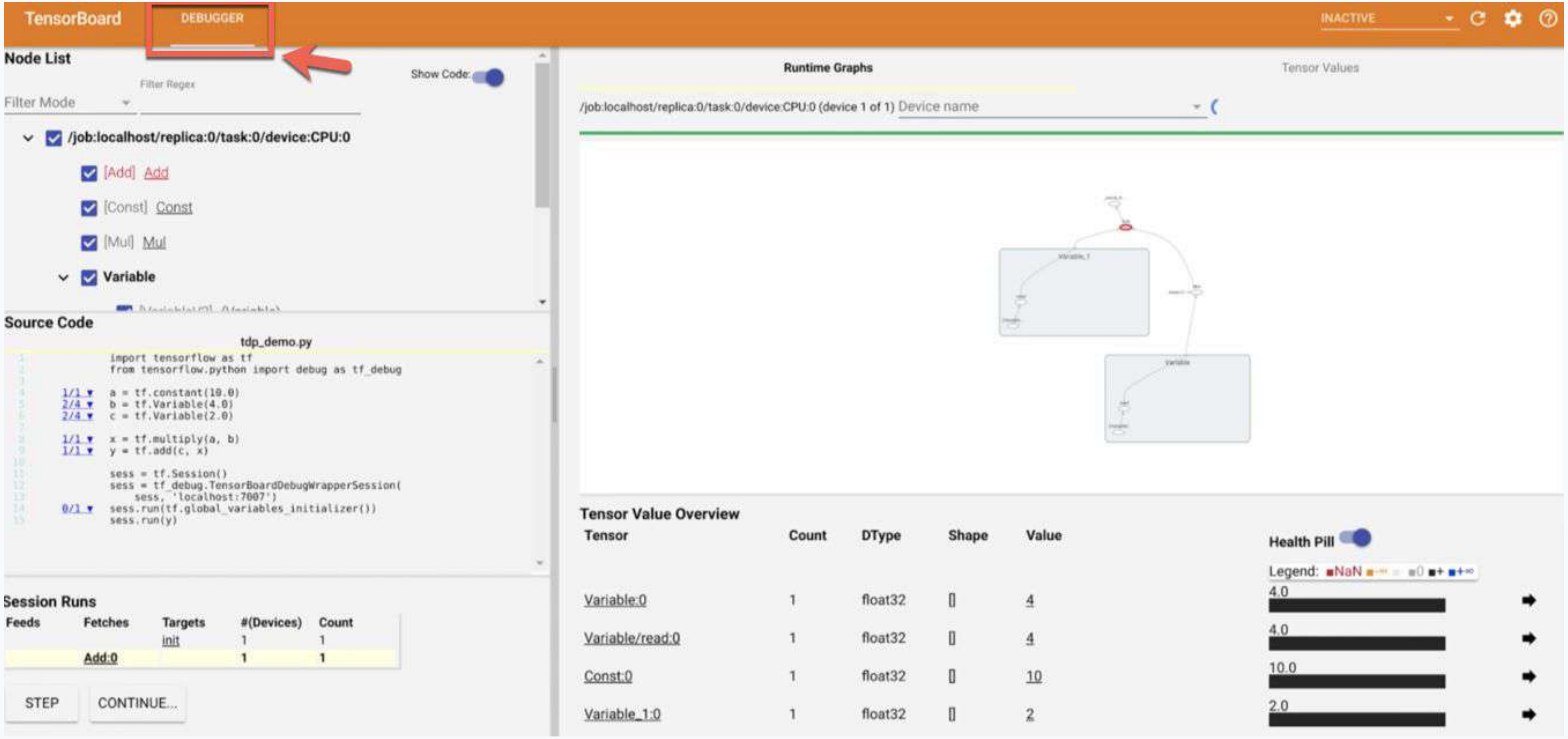


TensorBoard Provides Rich and Extendible Visualizations

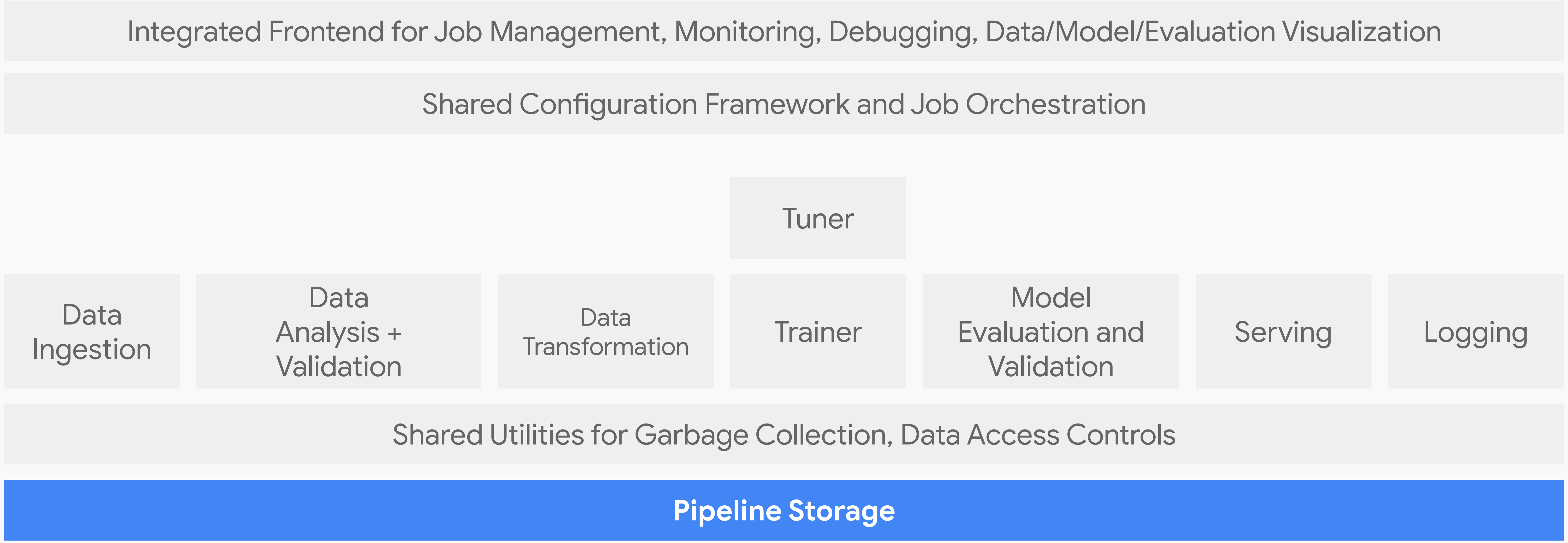



Embedding Projector
DATA
5 tensors found
Word2Vec 10K
Label by
word
Color by
No color map
Sphereize data
Load data
Publish
Checkpoint: Demo datasets
Metadata: oss_data/word2vec_10000_200d_labels.tsv
T-SNE
PCA
CUSTOM
X
Component #1
Y
Component #2
Z
Component #3
PCA is approximate.
Total variance described: 8.5%.
Points: 10000 | Dimension: 200 | Selected 101 points
cat
Show All Data
Isolate 101 points
Clear selection
Search
cat
by
word
neighbors
100
distance
COSINE
EUCLIDEAN
Nearest points in the original space:
cats 0.470
dog 0.575
dogs 0.604
pet 0.616
mouse 0.637
lovers 0.638
breeds 0.660
breed 0.683
creature 0.687
black 0.690
big 0.690
animal 0.693
hat 0.698
toy 0.698
walking 0.702
sleeping 0.718
sheep 0.724
cow 0.725
bat 0.727
fish 0.730
BOOKMARKS (0)

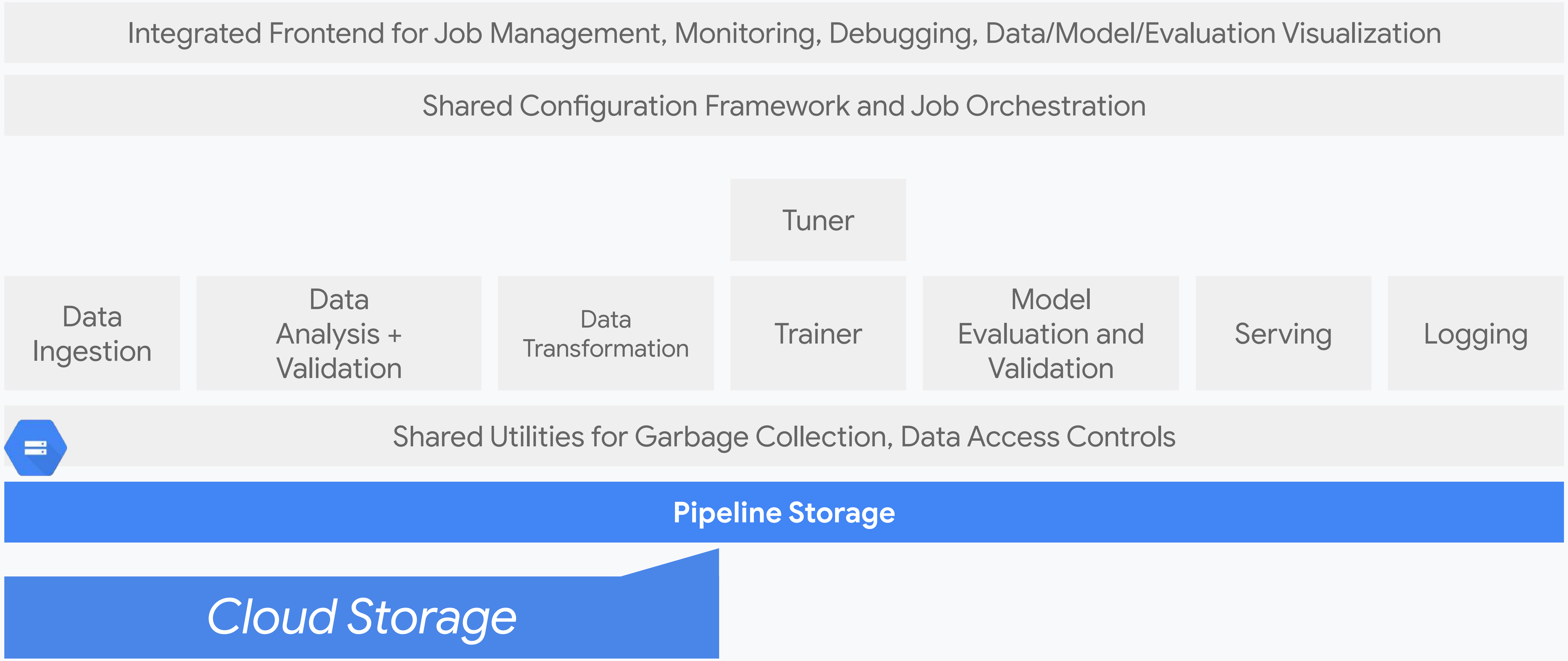
Debug TensorFlow in real-time



Production ML System Component: Pipeline Storage



Production ML System Component: Pipeline Storage



Course 2: Production ML Systems

Module 1: Architecting Production ML Systems

Lesson Title: **Training Design Decisions**

Presenter: Max Lotstein

Format: Talking Head

Video Name: T-PSML-O_1_l10_training_design_decisions

Agenda

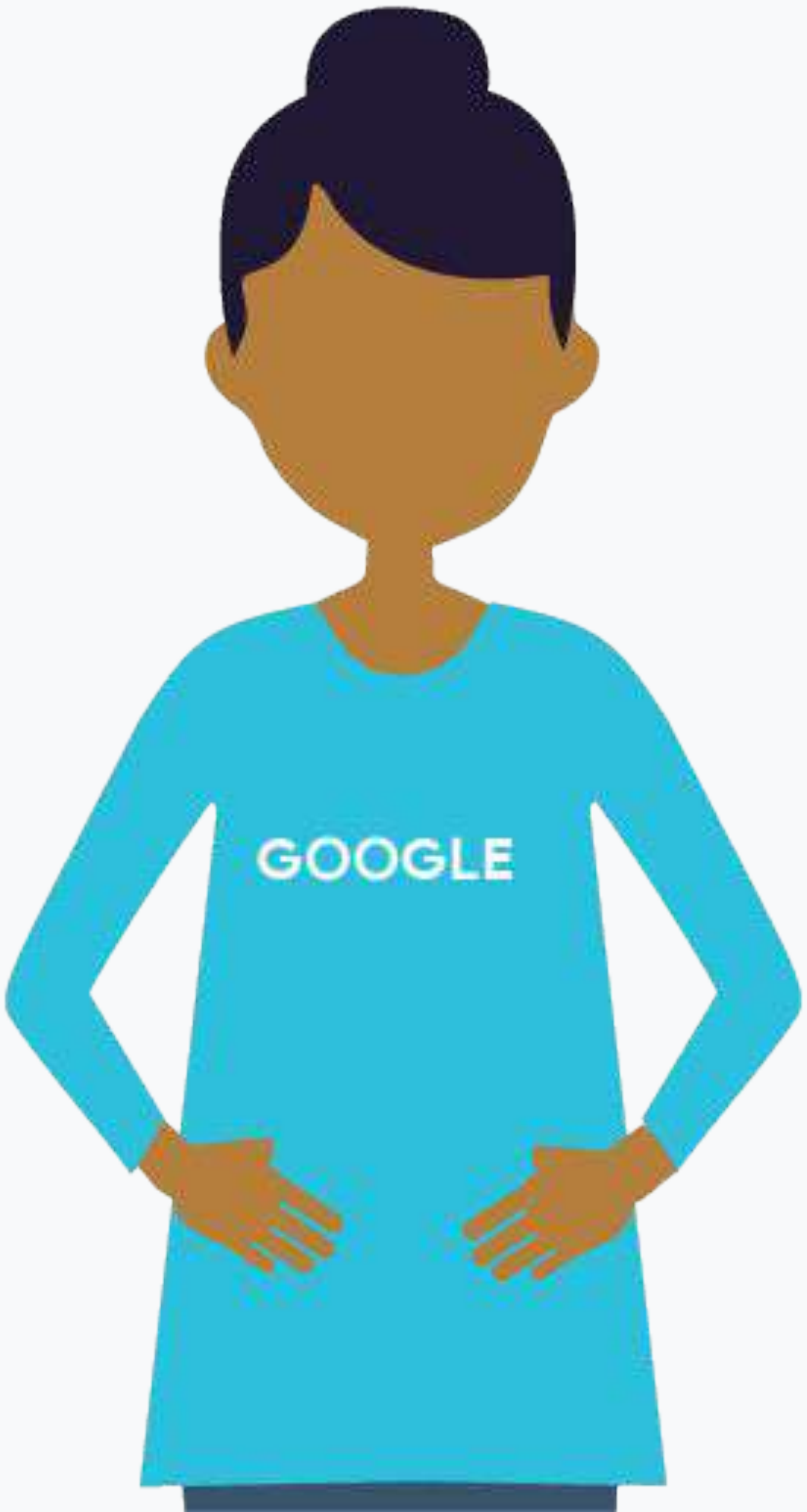
What's in a Production ML System

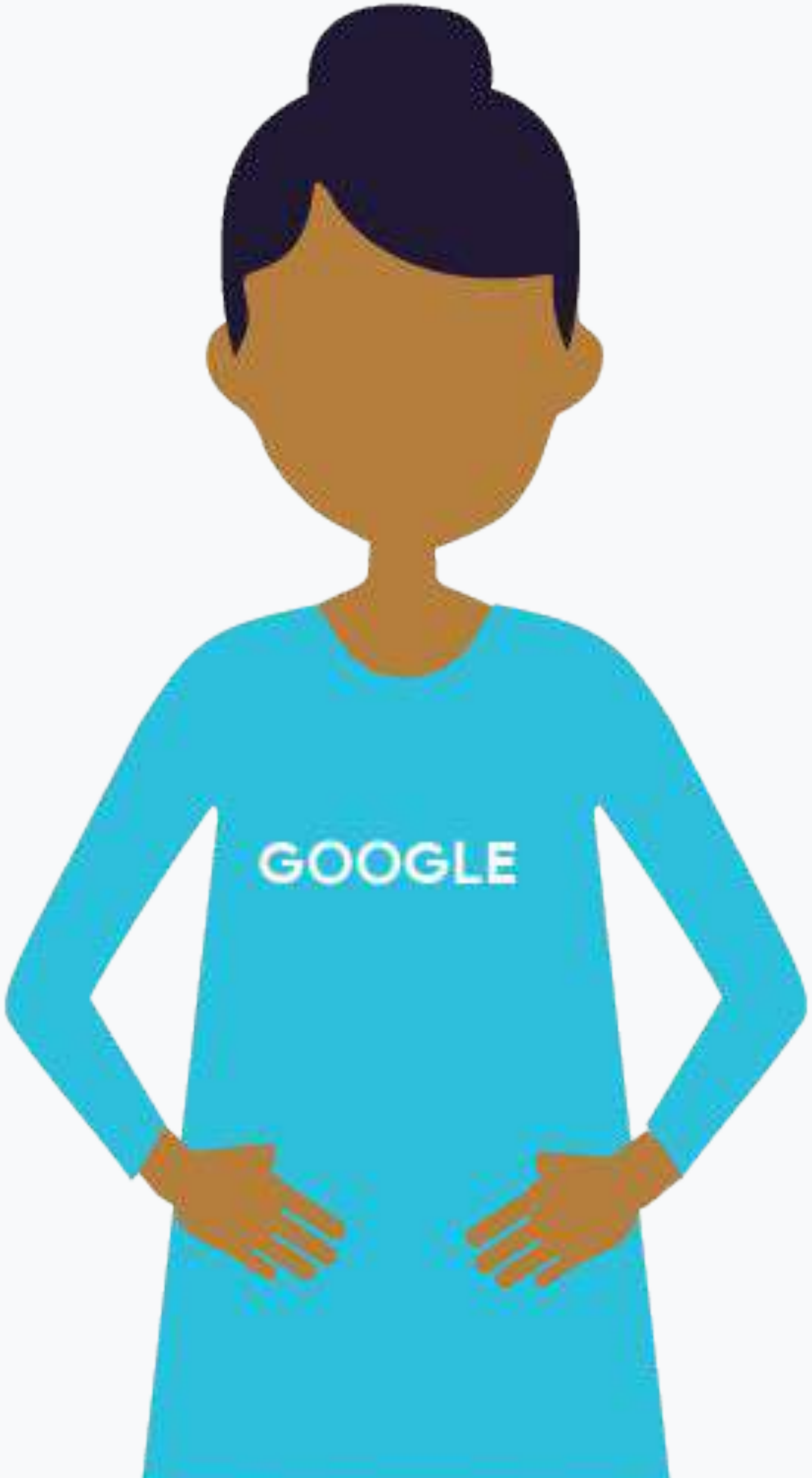
Training Design Decisions

Serving Design Decisions

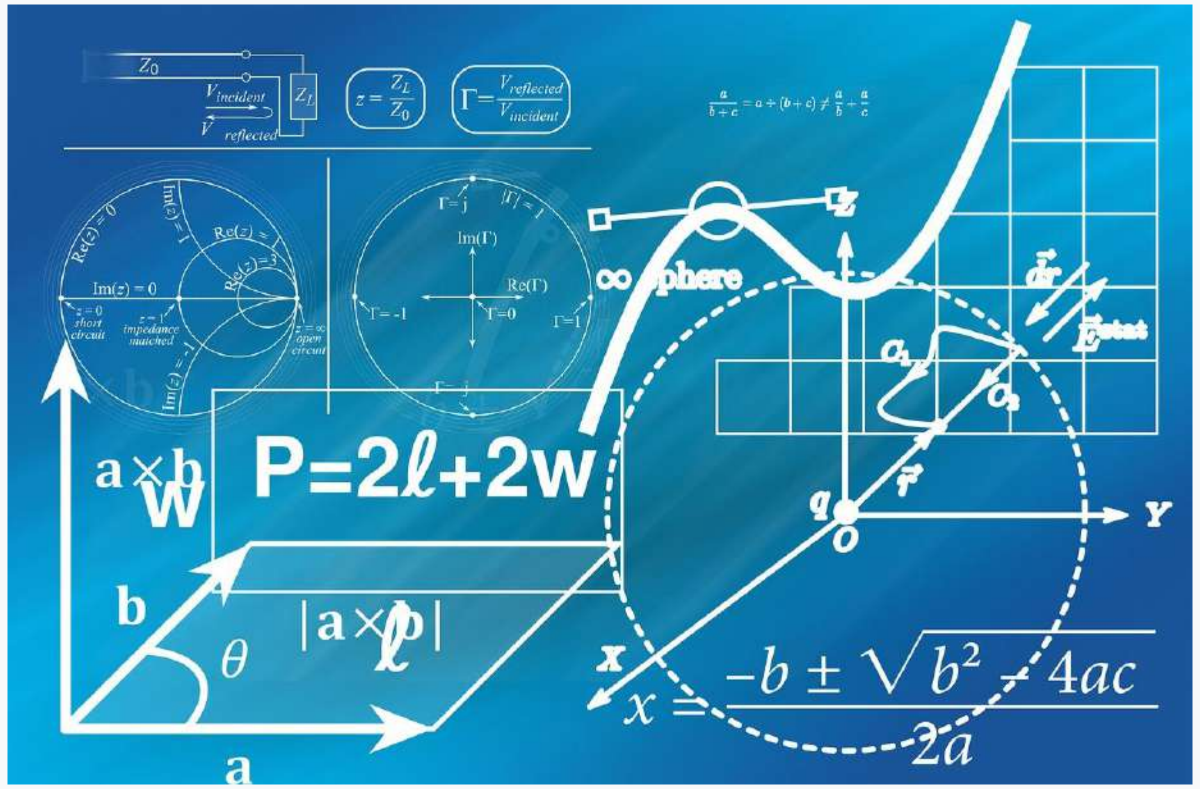
Serving on CMLE

Designing an Architecture from Scratch

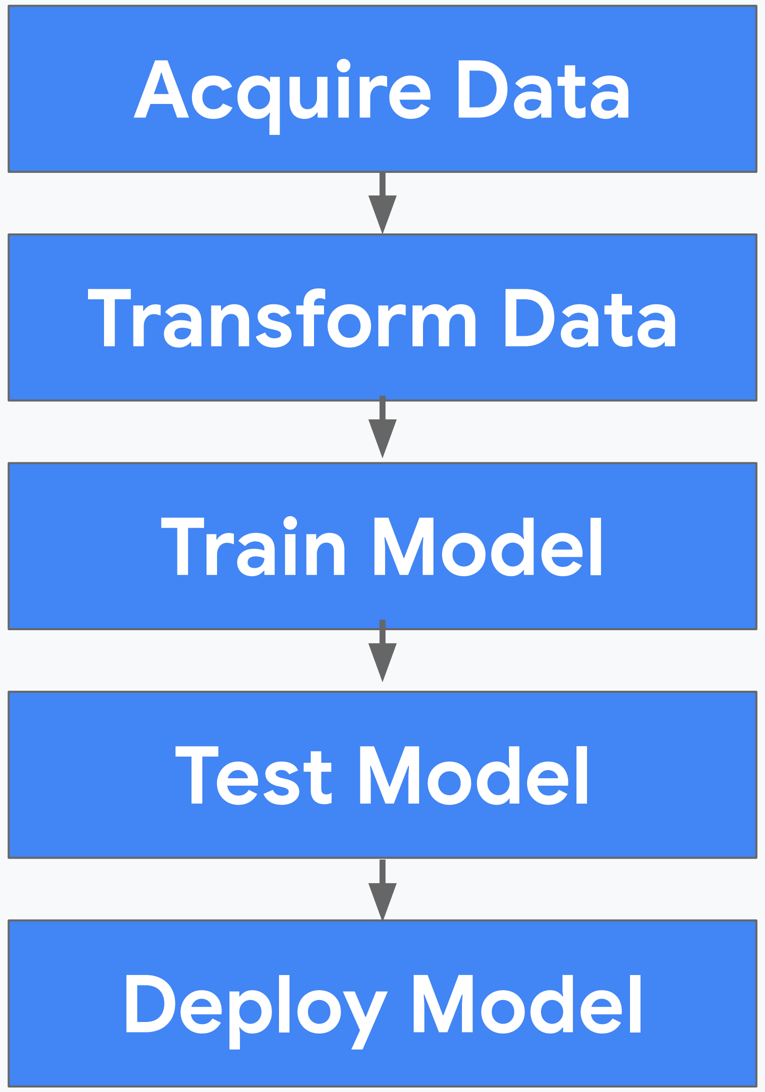




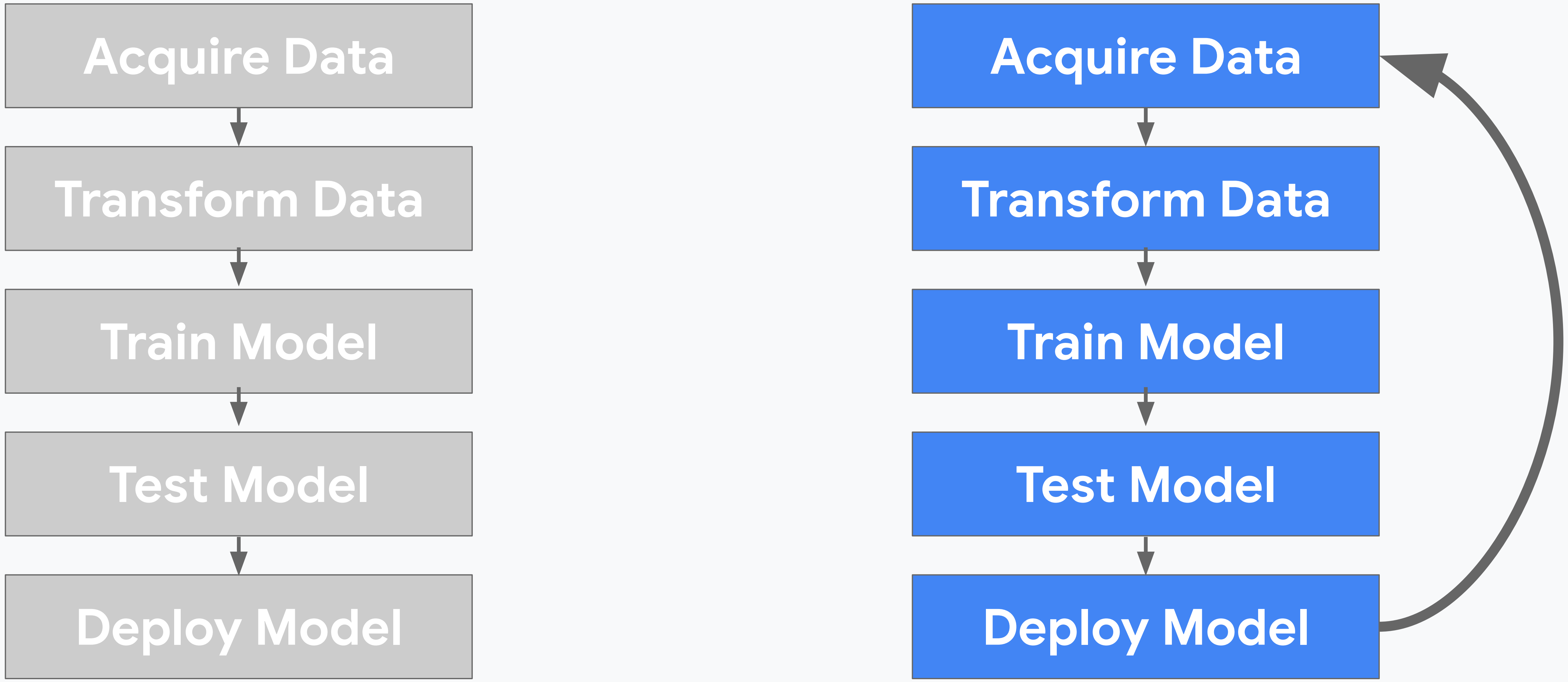
Physics vs Fashion

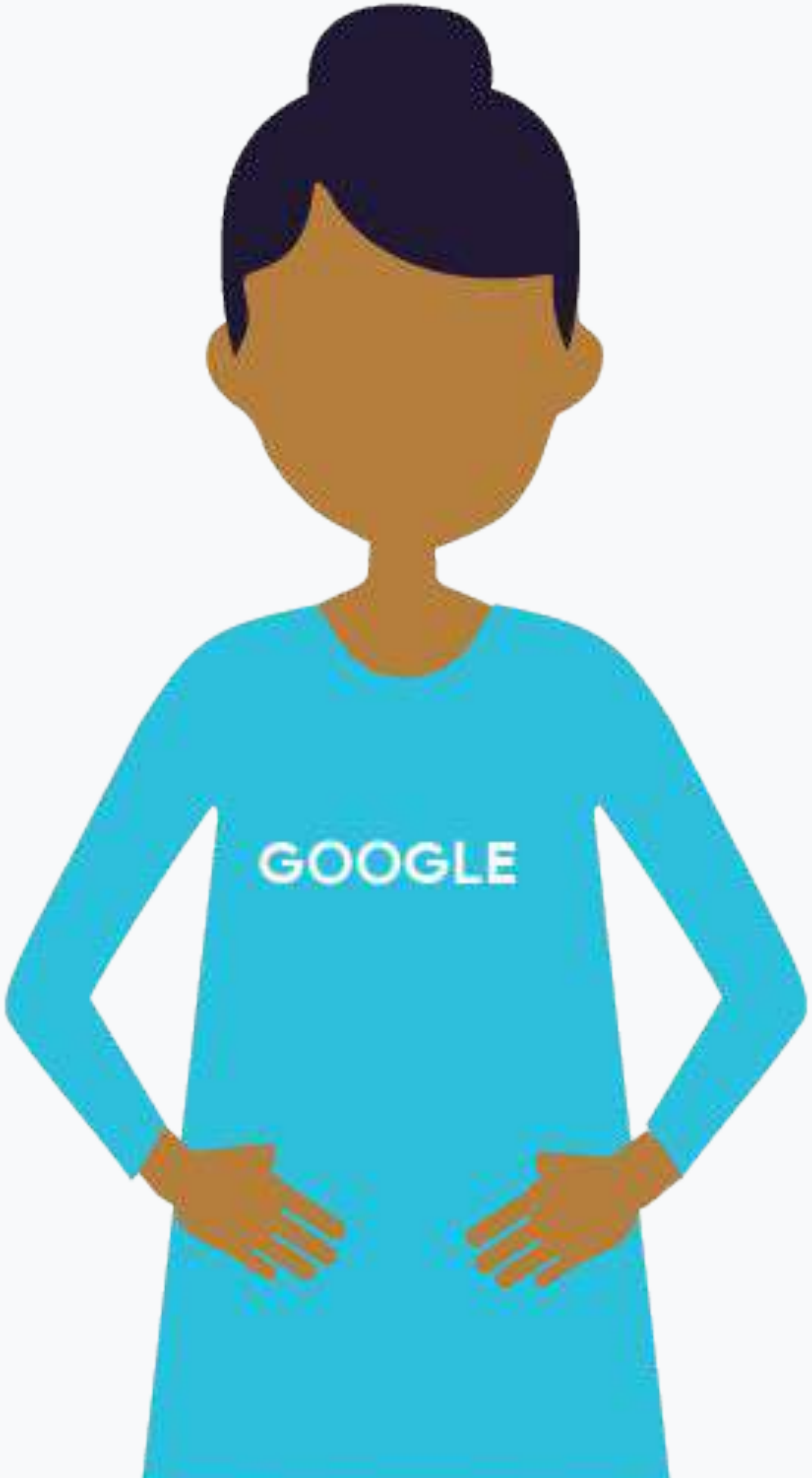


Static vs Dynamic Training



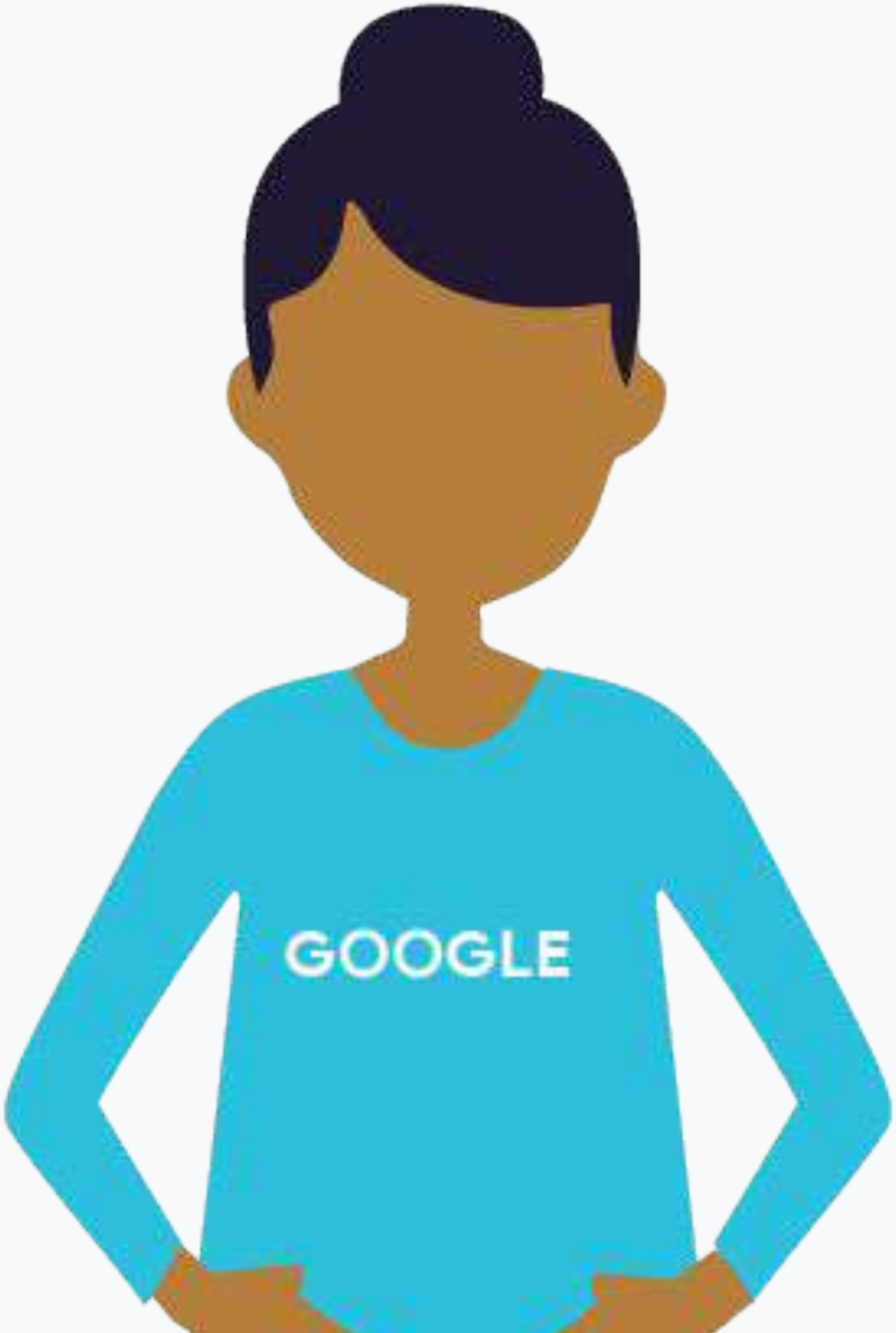
Static vs Dynamic Training

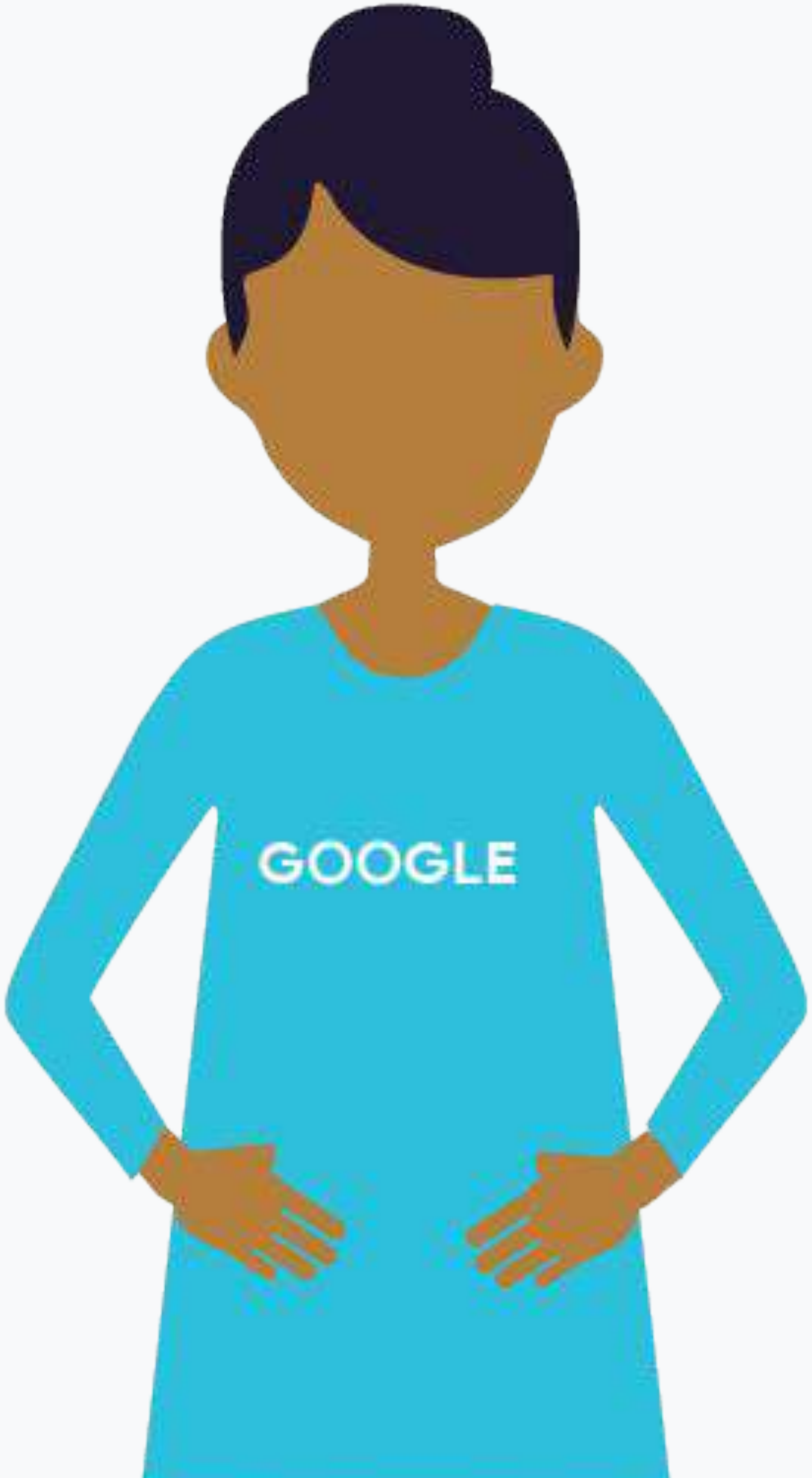




Static vs Dynamic Training

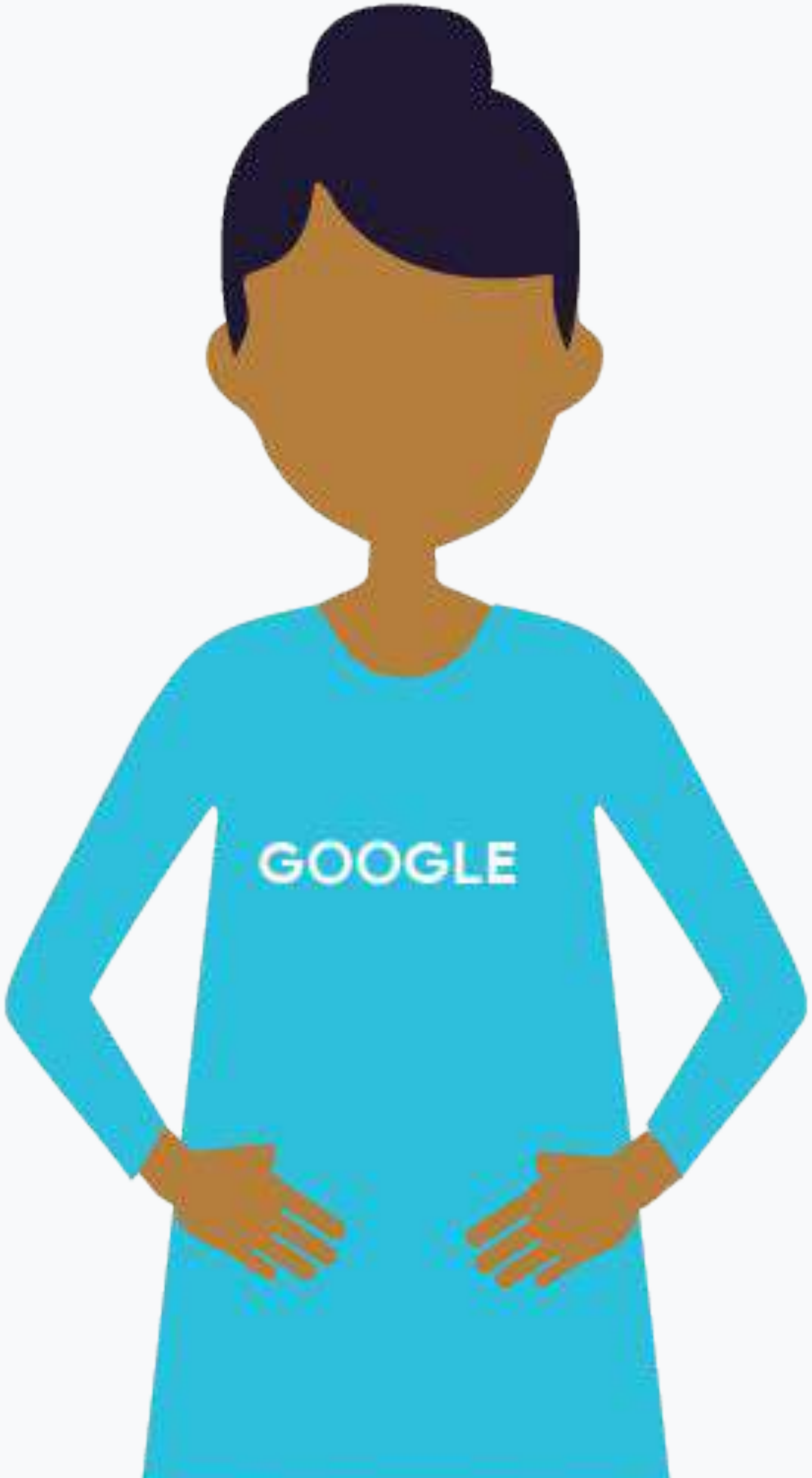
Statically Trained Models	Dynamically Trained Models
Trained once, offline	Add training data over time
Easy to build and test	Engineering is harder Have to do progressive validation
Easy to let become stale	Regularly sync out updated version Will adapt to changes


GOOGLE



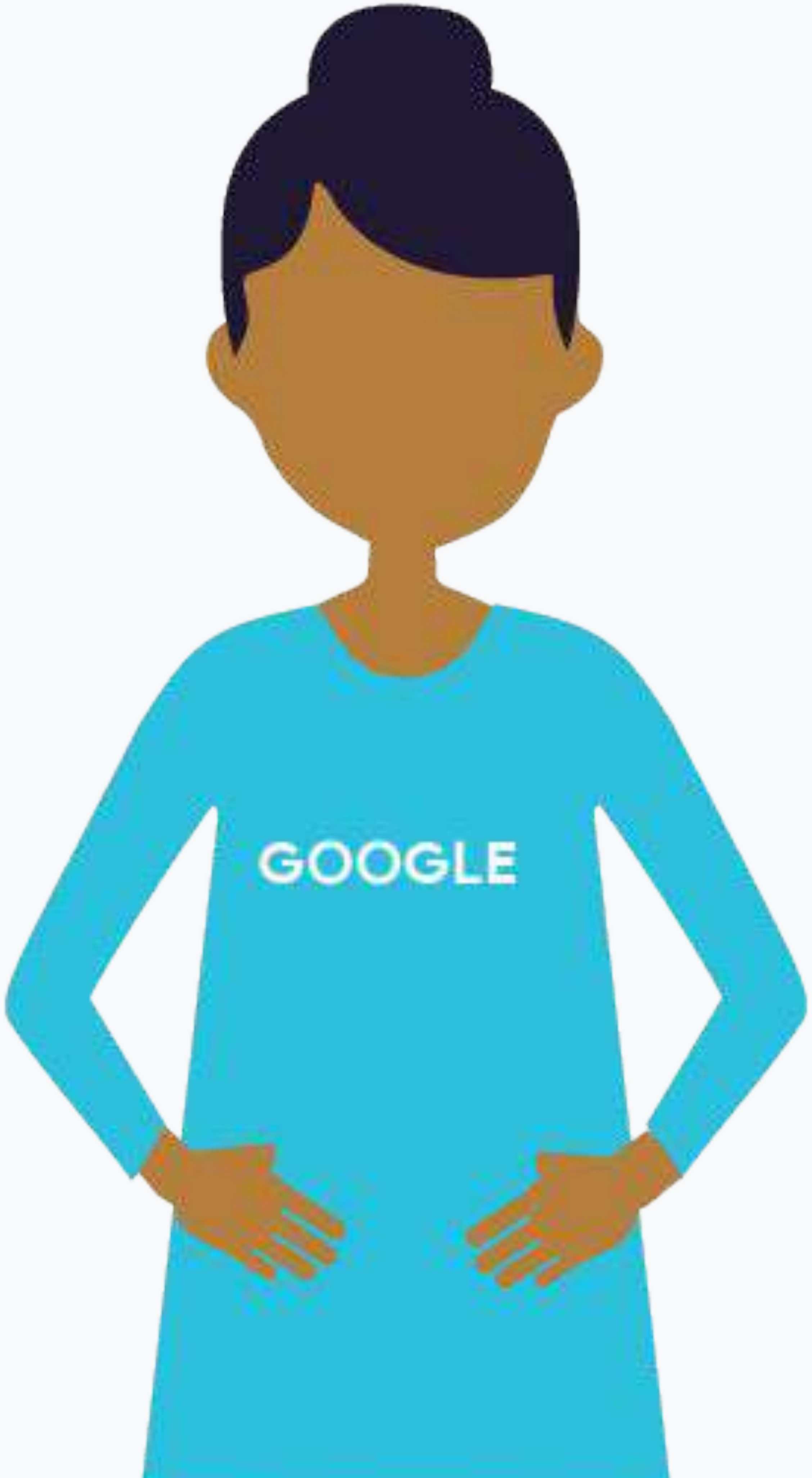
Lab: Estimate training needs

Problem	Training style (static or dynamic?)
Predict whether email is spam	
Android voice to text	
Shopping ad conversion rate	



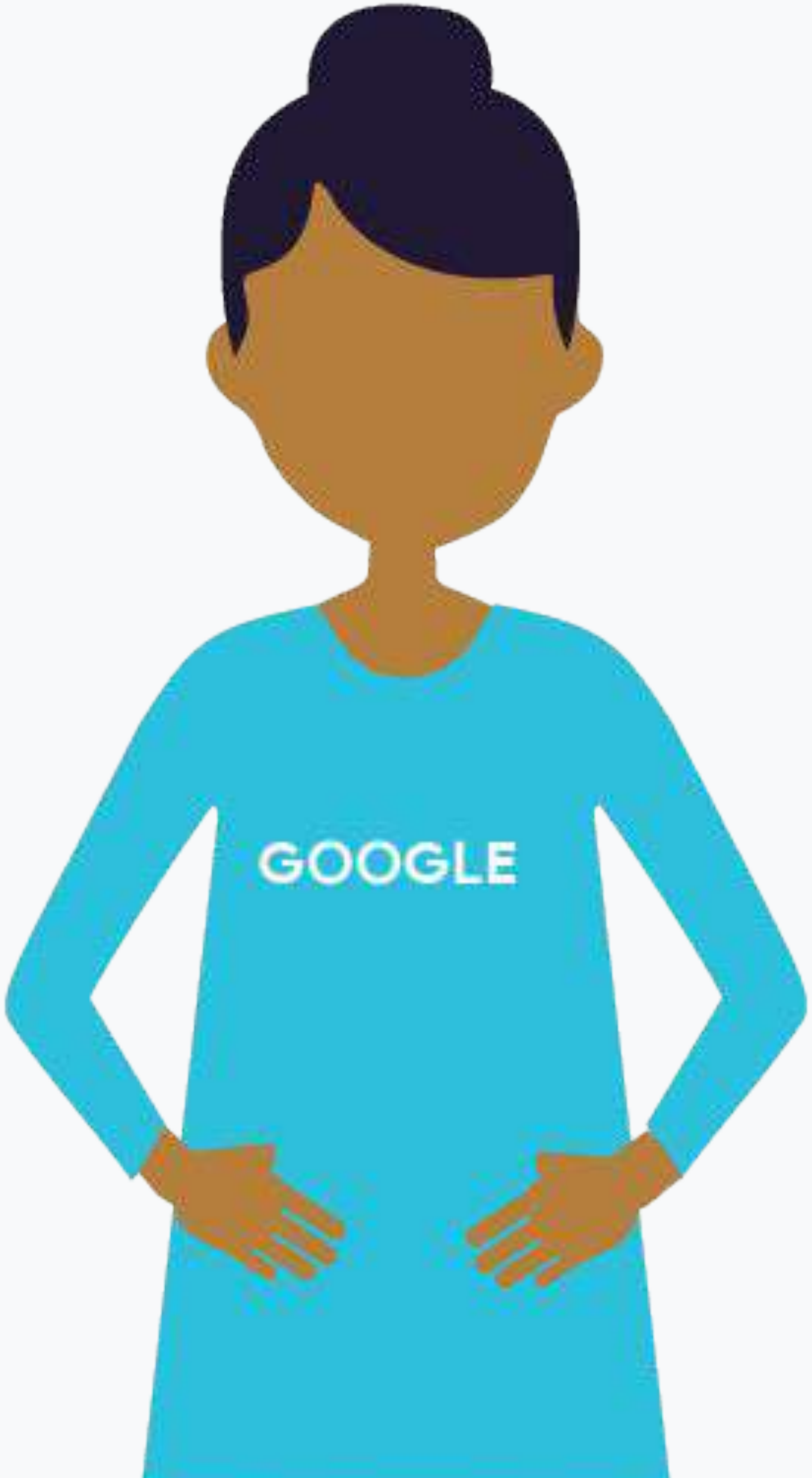
Lab: Estimate training needs

Problem	Training style (static or dynamic?)
Predict whether email is spam	**Static or Dynamic (How quickly spammers change)**
Android voice to text	
Shopping ad conversion rate	



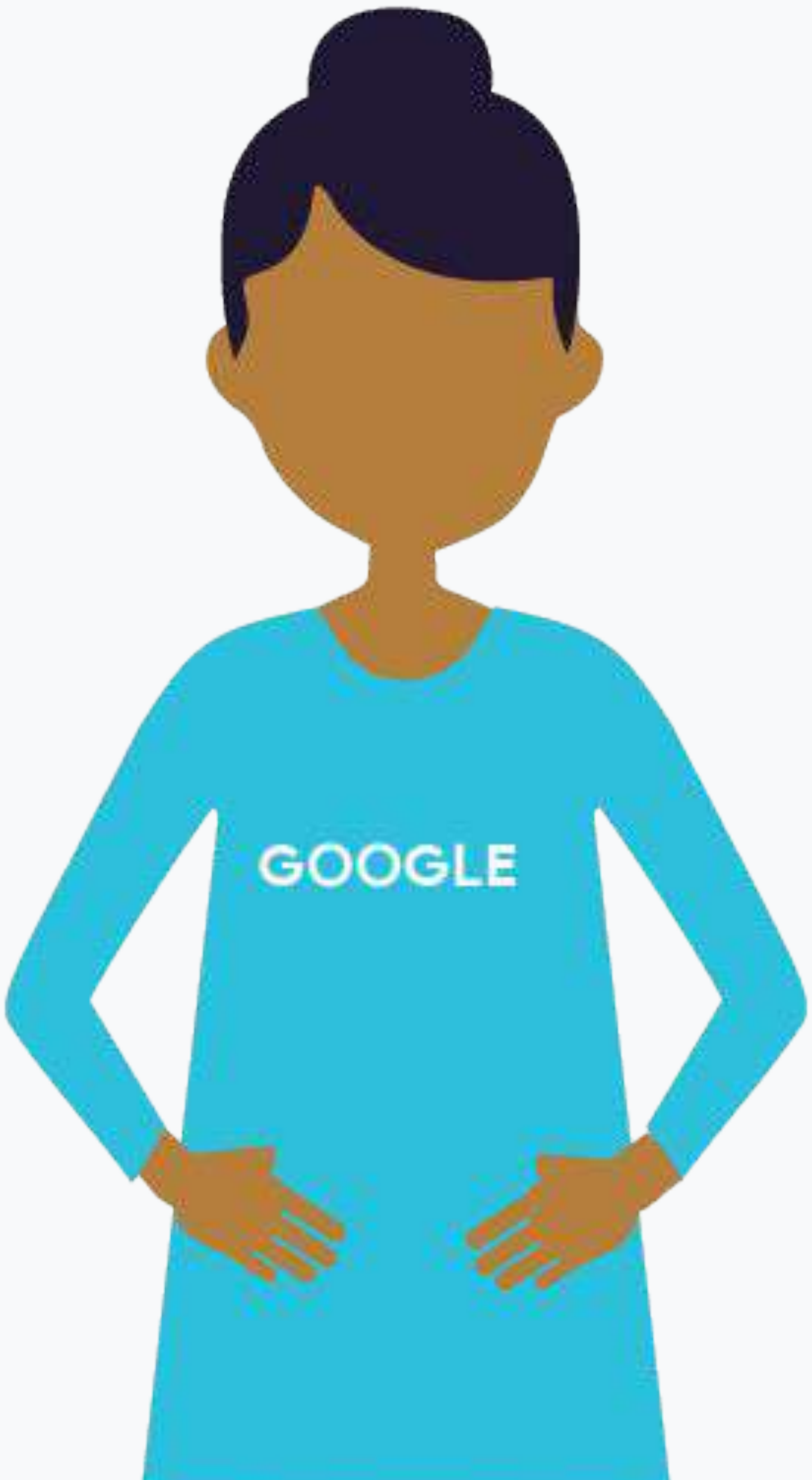
Lab: Estimate training needs

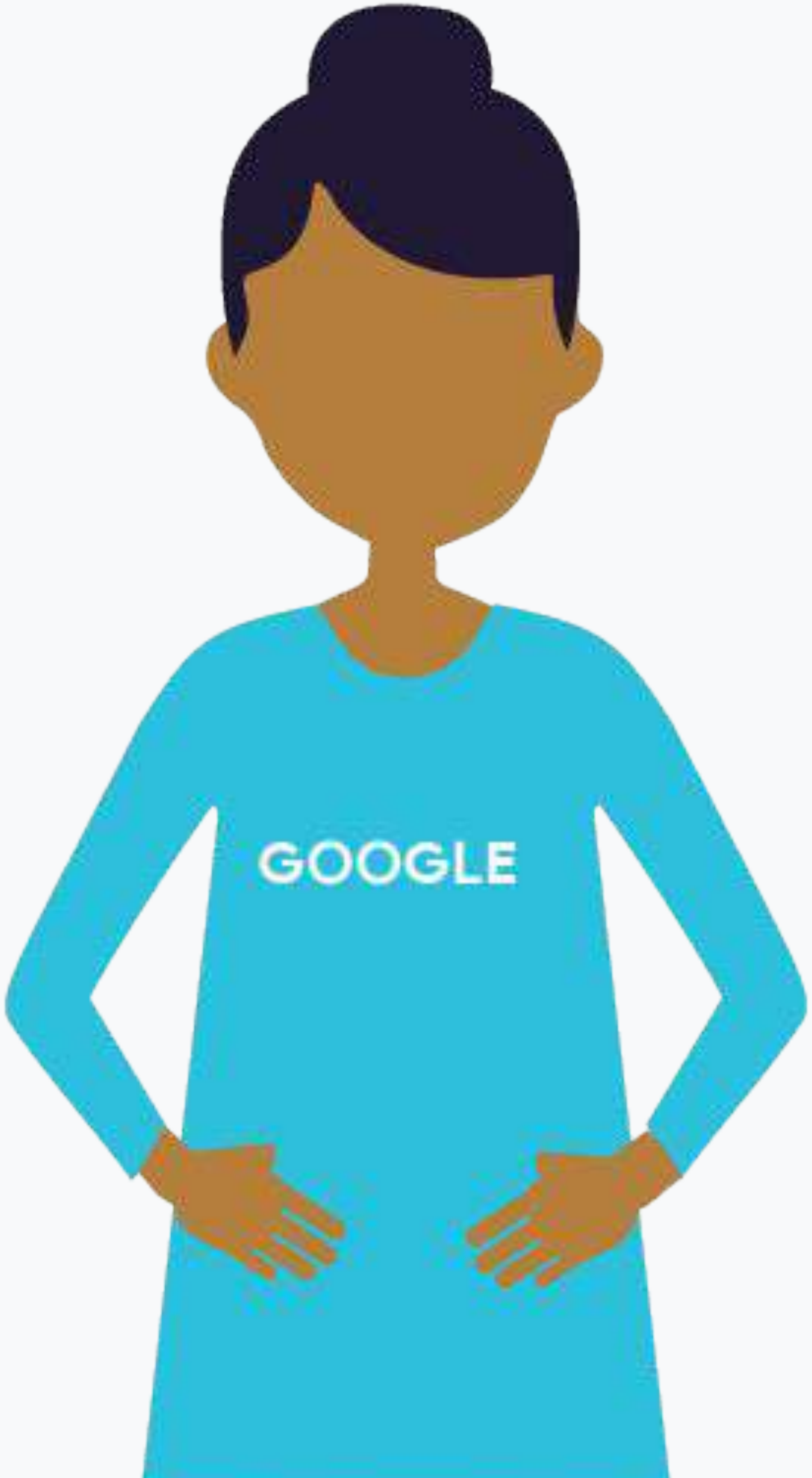
Problem	Training style (static or dynamic?)
Predict whether email is spam	**Static or Dynamic (How quickly spammers change)**
Android voice to text	**Static or Dynamic (Global vs personalized)**
Shopping ad conversion rate	



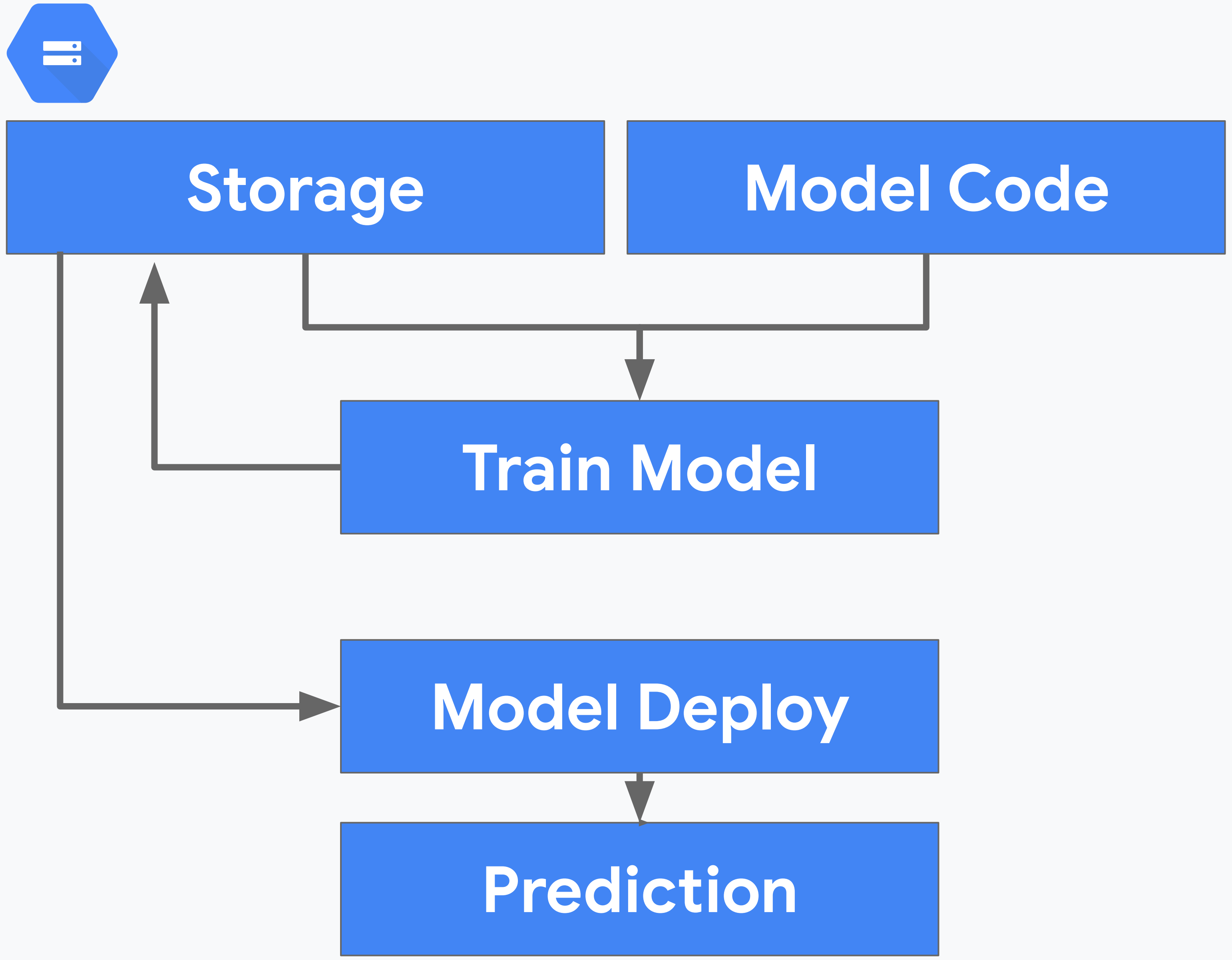
Lab: Estimate training needs

Problem	**Training style (static or dynamic?)**
Predict whether email is spam	**Static or Dynamic (How quickly spammers change)**
Android voice to text	**Static or Dynamic (Global vs personalized)**
Shopping ad conversion rate	**Static**


GOOGLE



Reference architecture for static training



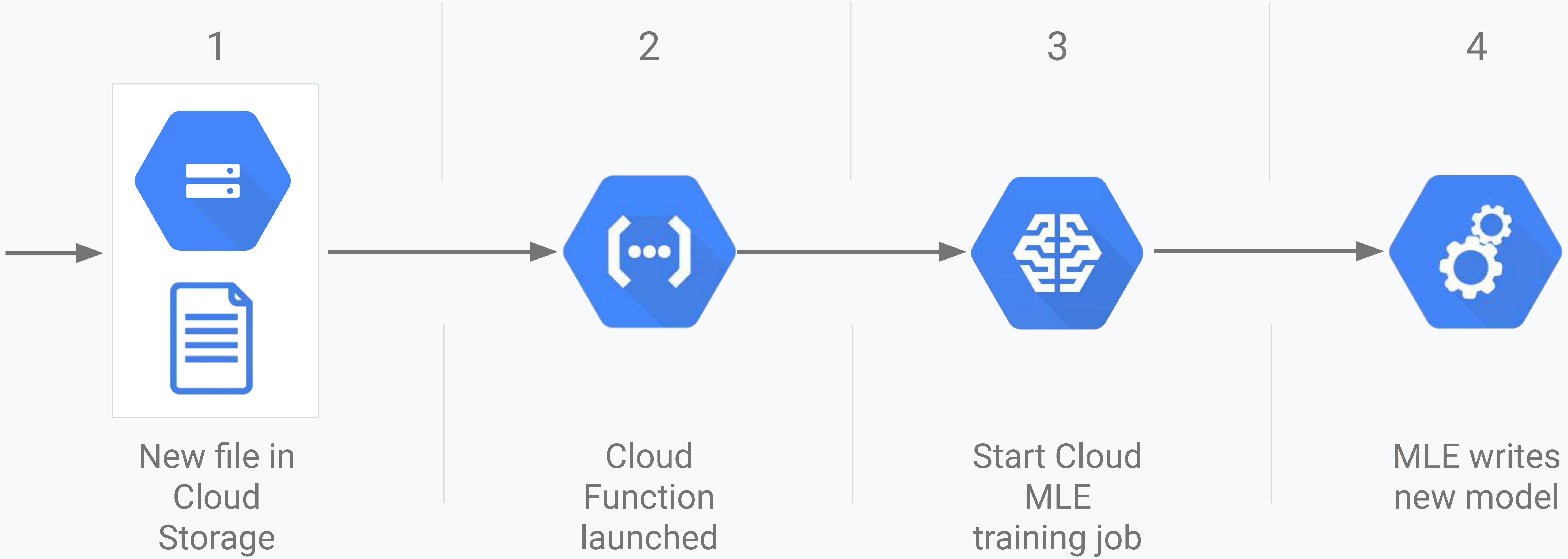
Three potential architectures for dynamic training

Cloud Functions
for asynchronous training jobs

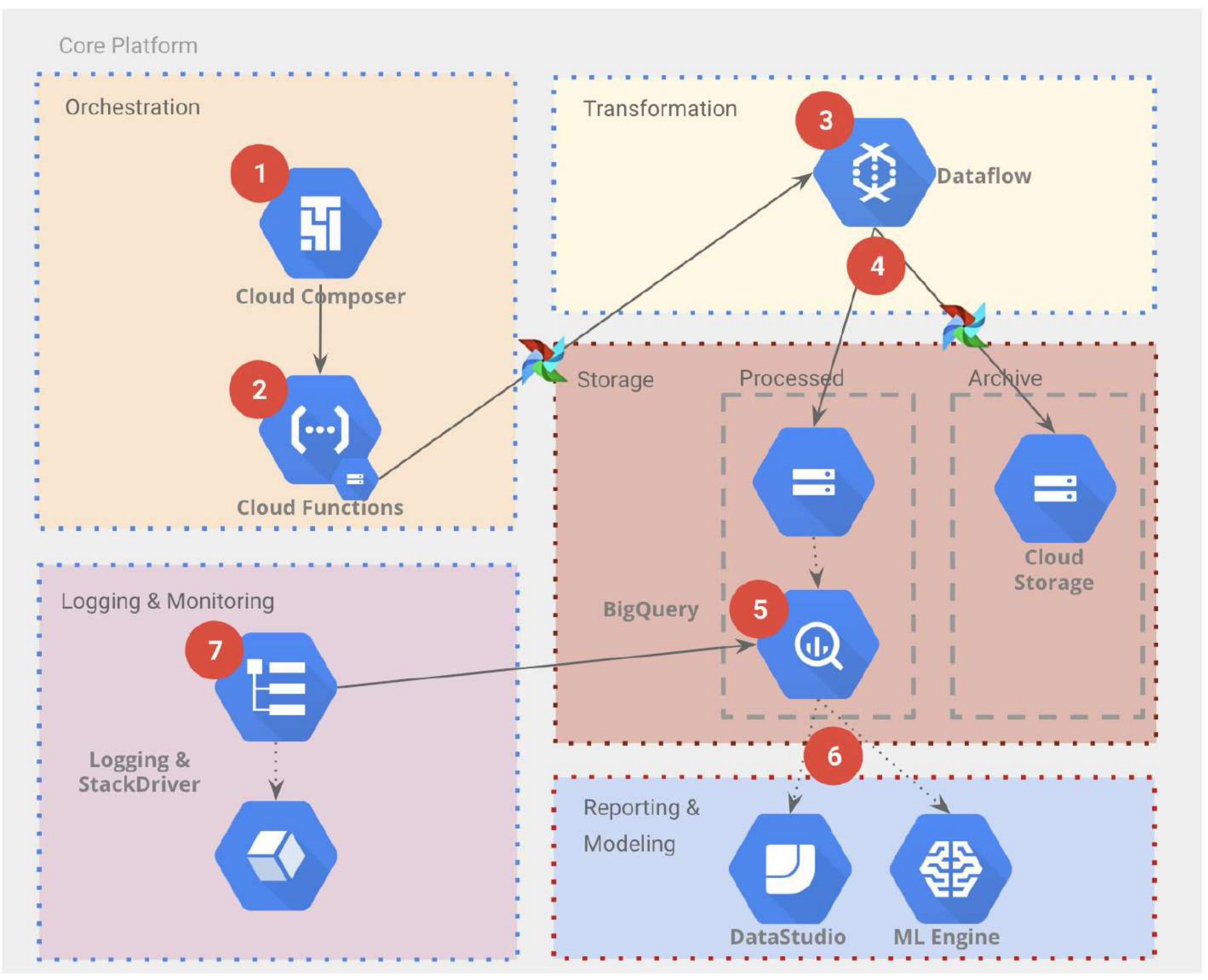
App Engine
for user-triggered training jobs

Cloud Dataflow
for continuous training

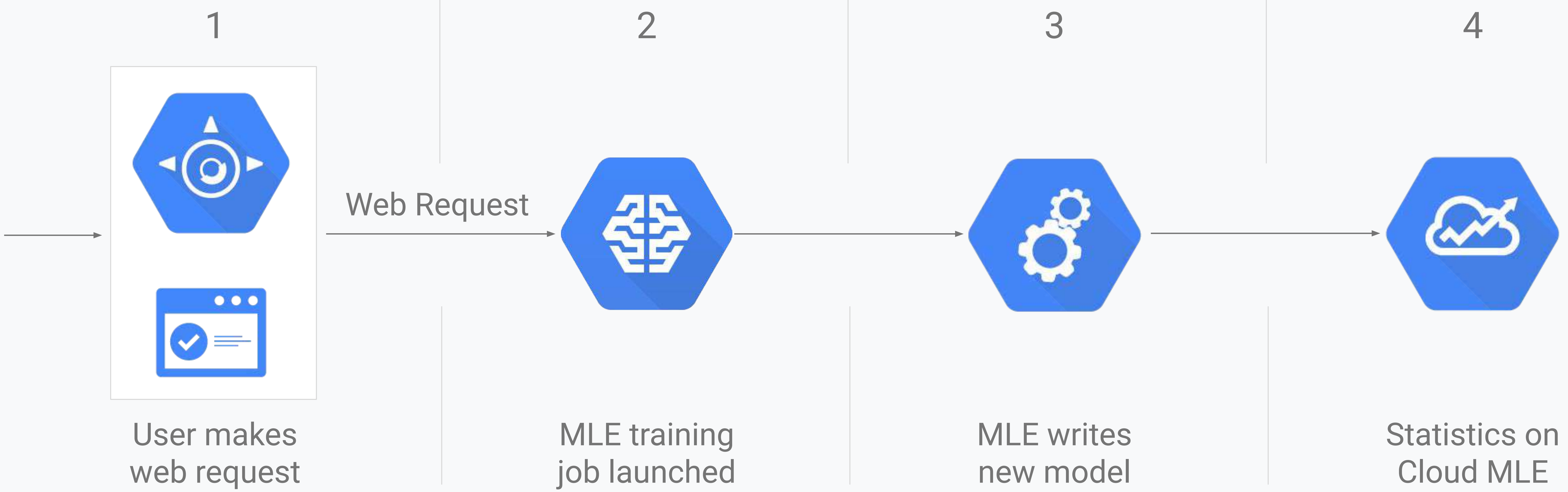
Reference architecture for dynamic training



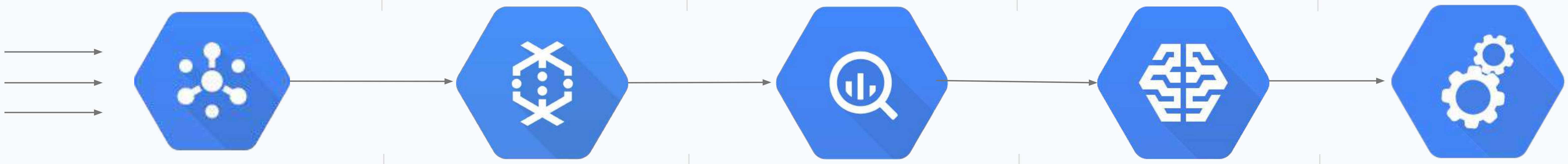
Use Cloud Composer to Orchestrate Jobs



AppEngine can be used for user-triggered training jobs



Dataflow can be used for continuous training



1. Messages into Pub/Sub
2. Messages aggregated with Dataflow
3. Aggregated data is stored into BigQuery
4. Cloud MLE launched on new data in BigQuery
5. Updated model is deployed

Course 2: Production ML Systems

Module 1: Architecting Production ML Systems

Lesson Title: **Serving Design Decisions**

Presenter: Max Lotstein

Format: Talking Head

Video Name: T-PSML-O_1_l11_serving_design_decisions

Agenda

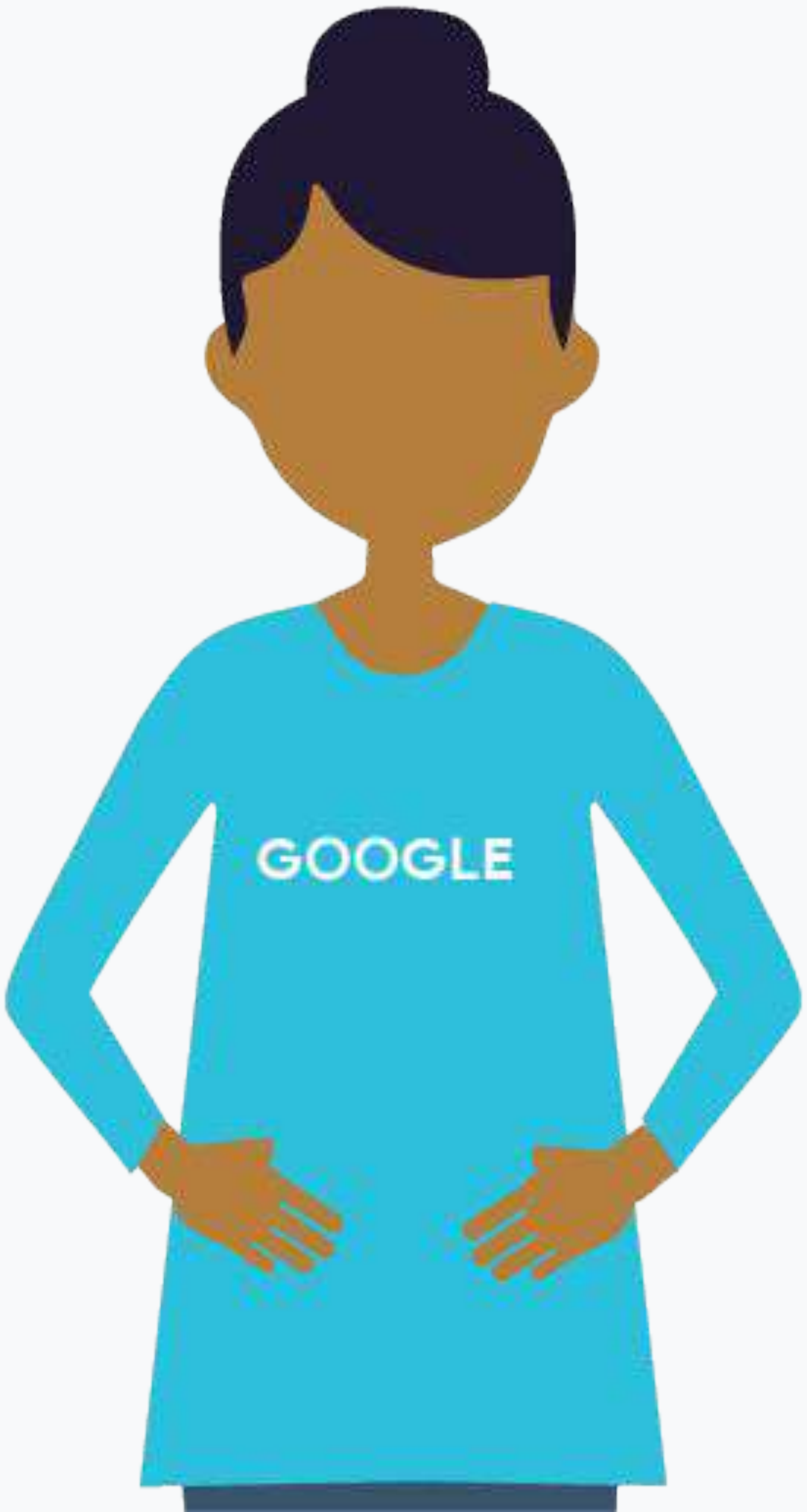
What's in a Production ML System

Training Design Decisions

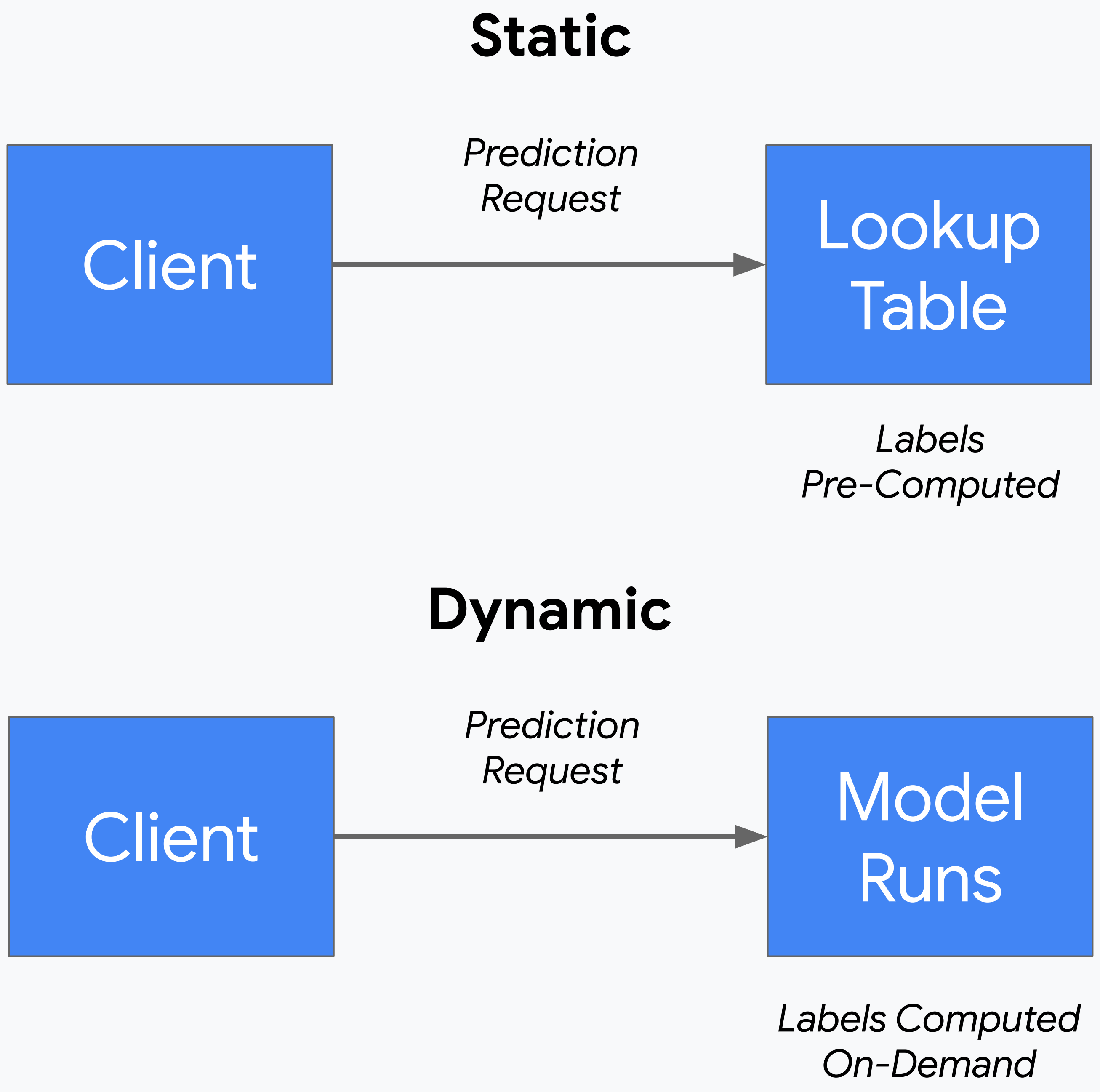
Serving Design Decisions

Serving on CMLE

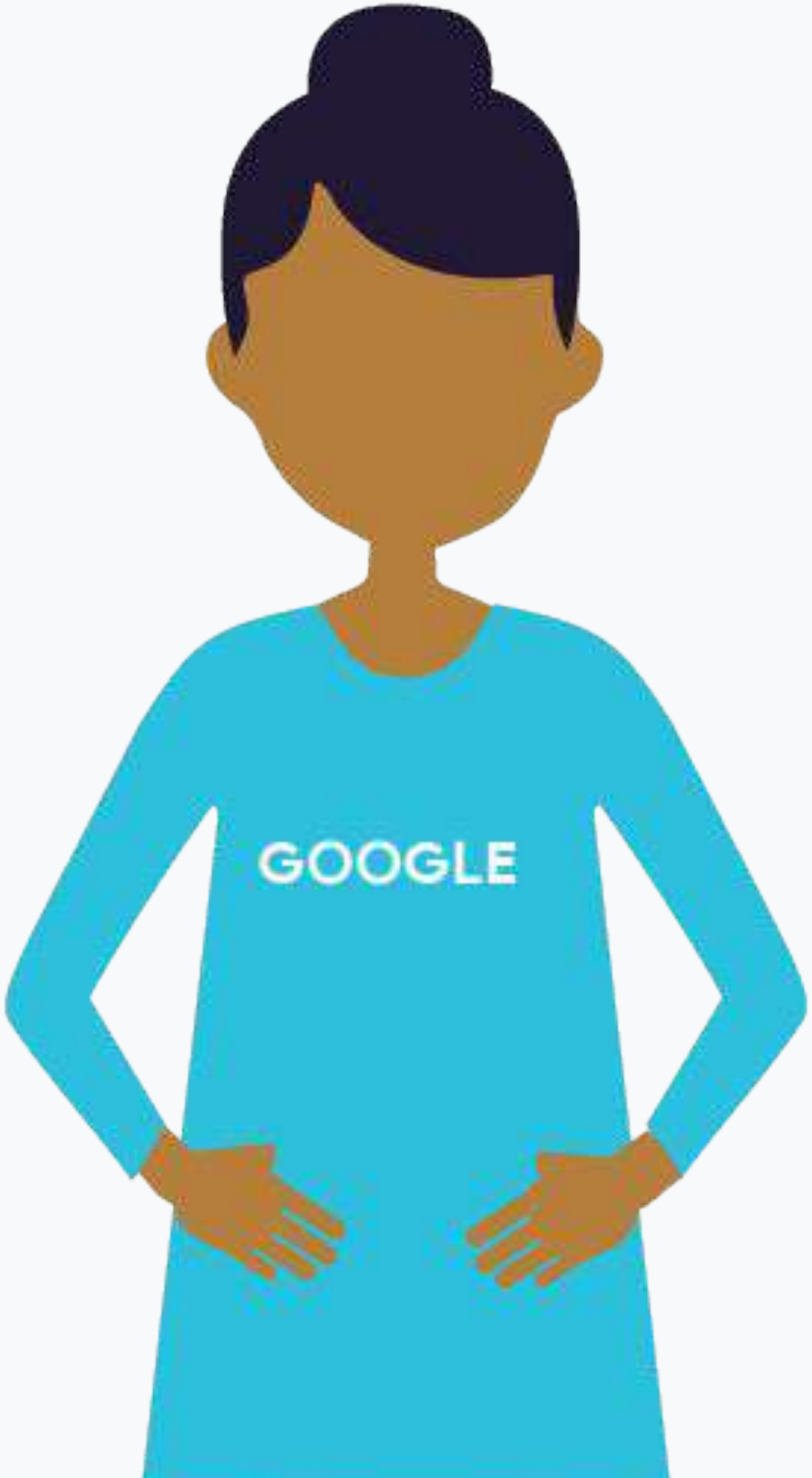
Designing an Architecture from Scratch



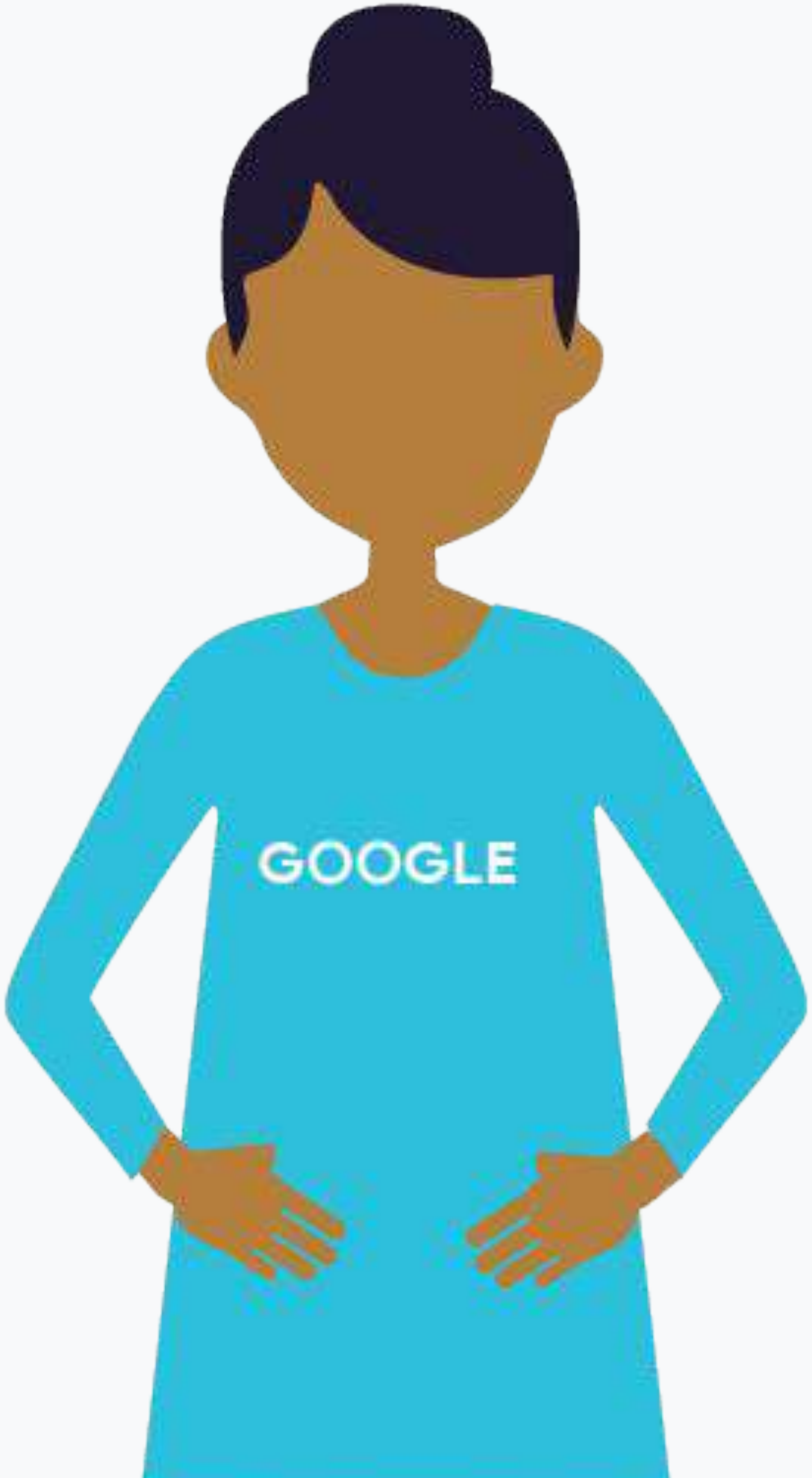
Static vs Dynamic Serving



Static	Dynamic
Higher Storage Cost	Lower Storage Cost
Low, Fixed Latency	Variable Latency
Lower Maintenance	Higher Maintenance
Space intensive	Compute intensive

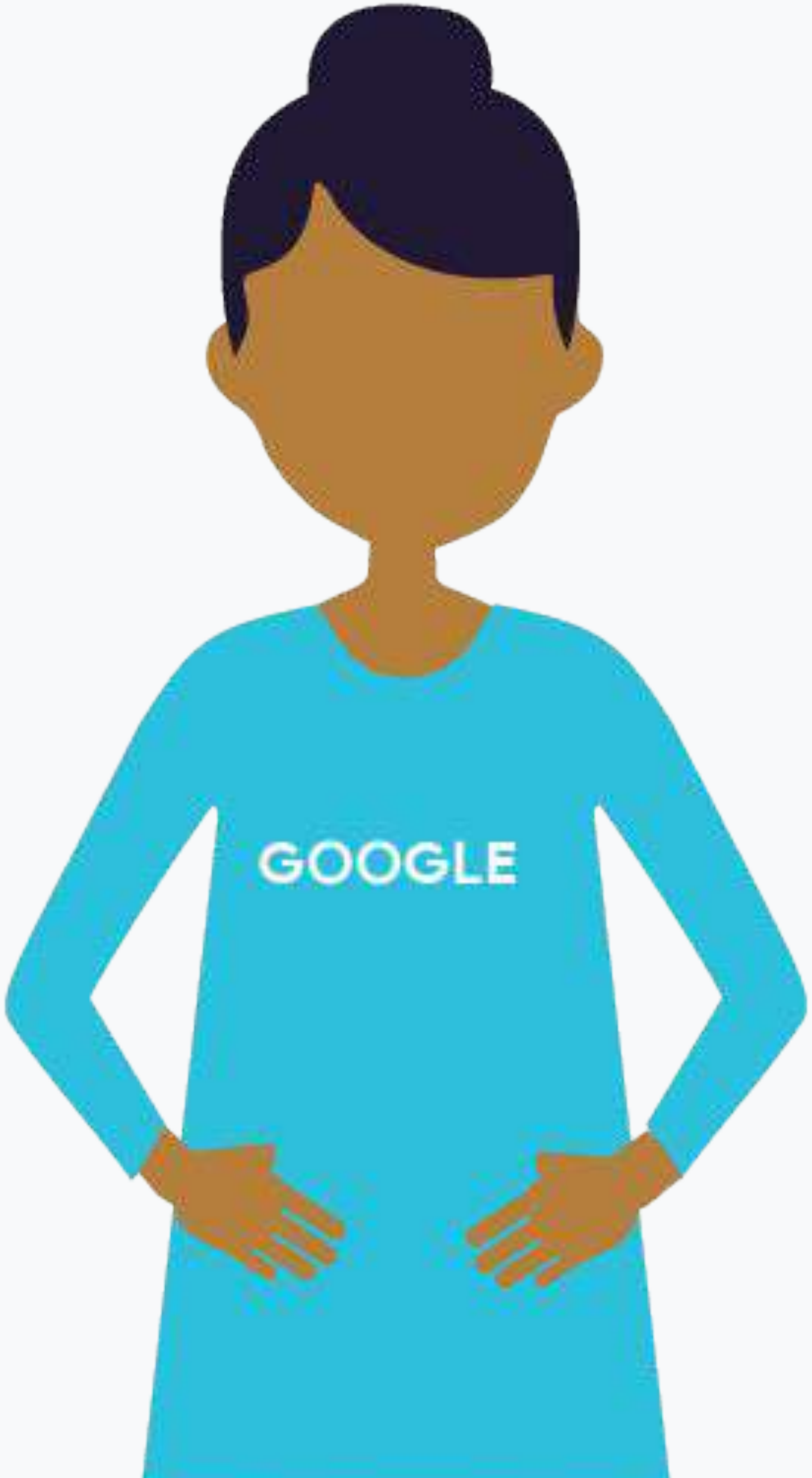


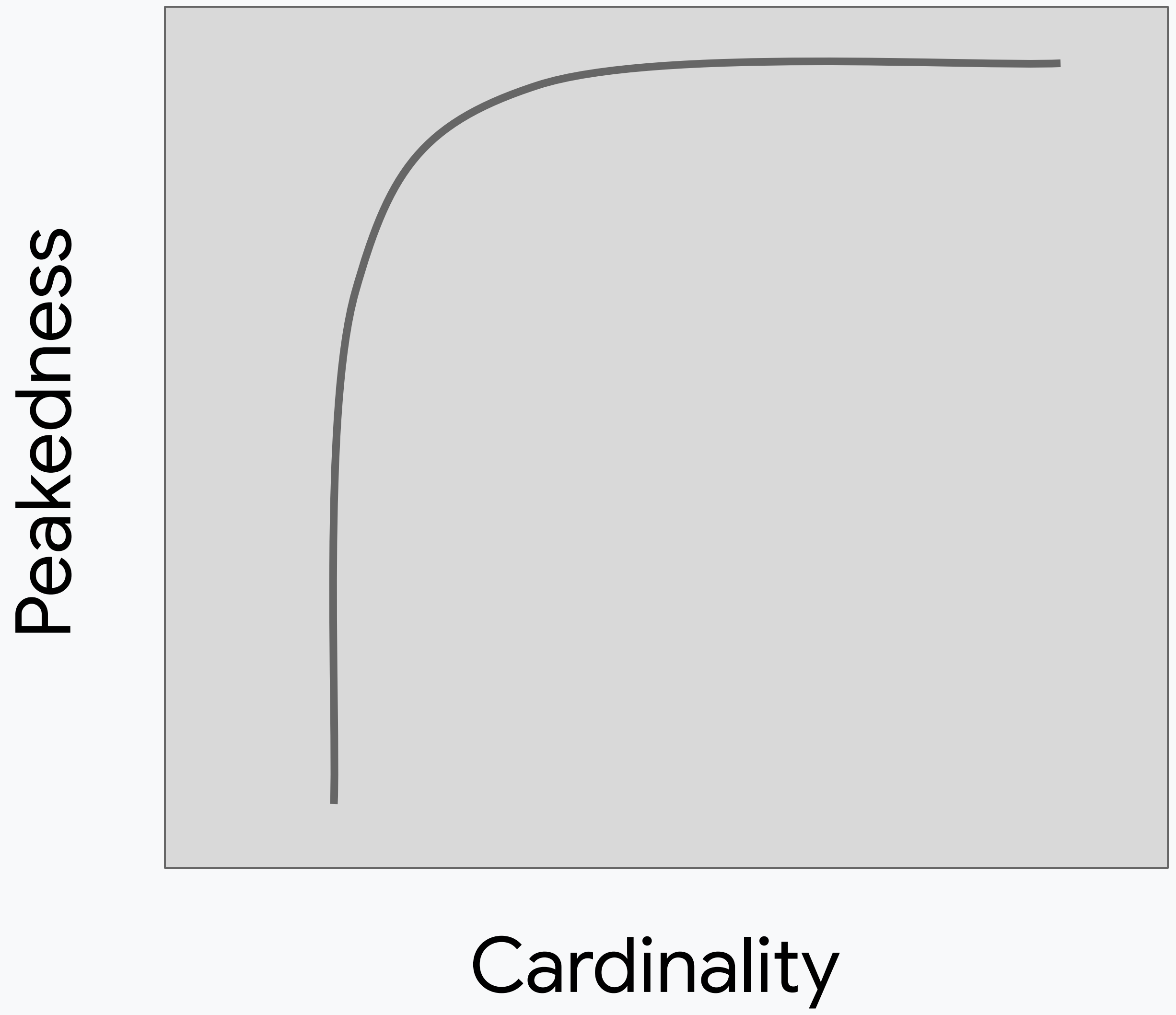
Peakedness is how concentrated the distribution is

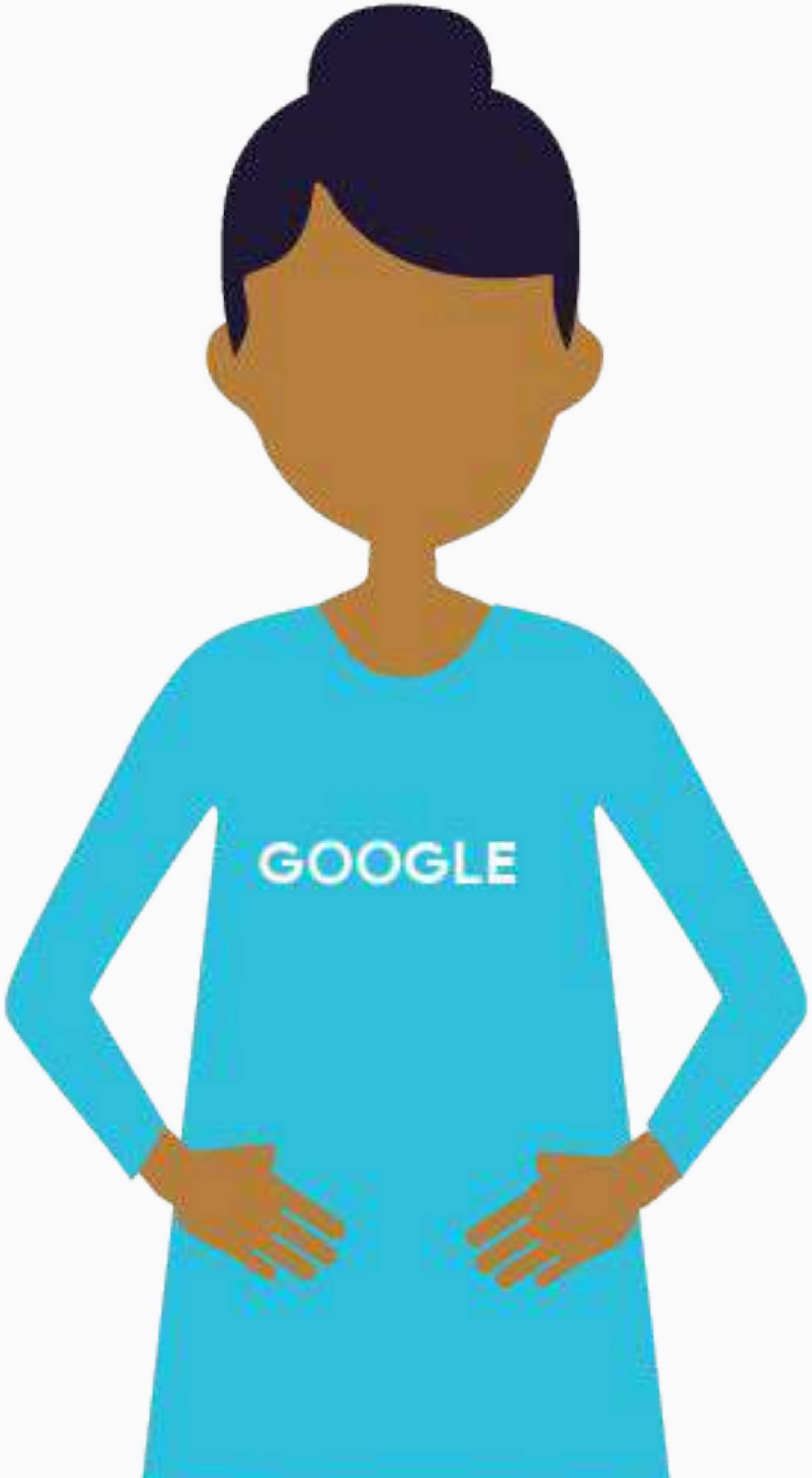


Cardinality is the number of values in the set

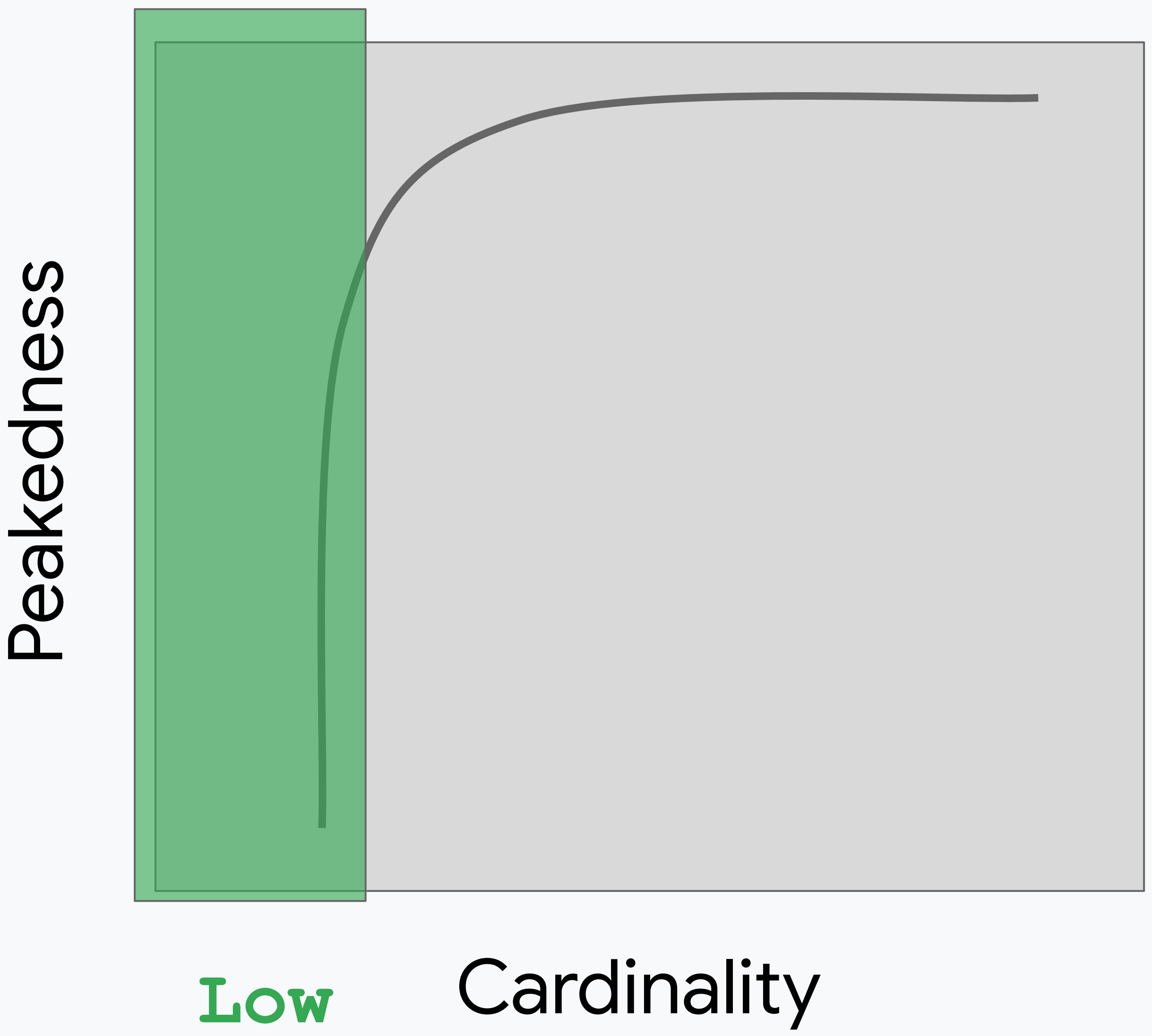
Peakedness and Cardinality space

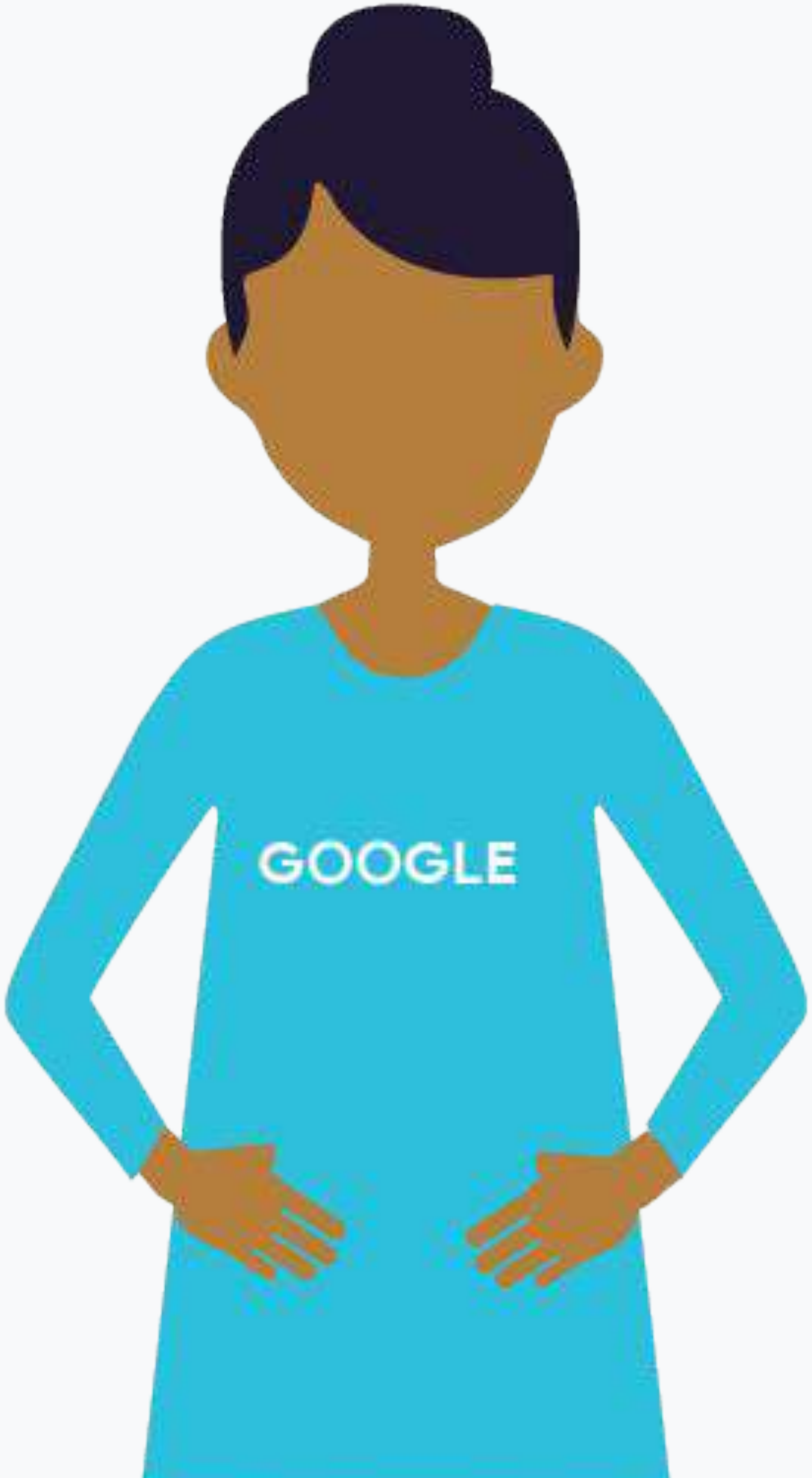




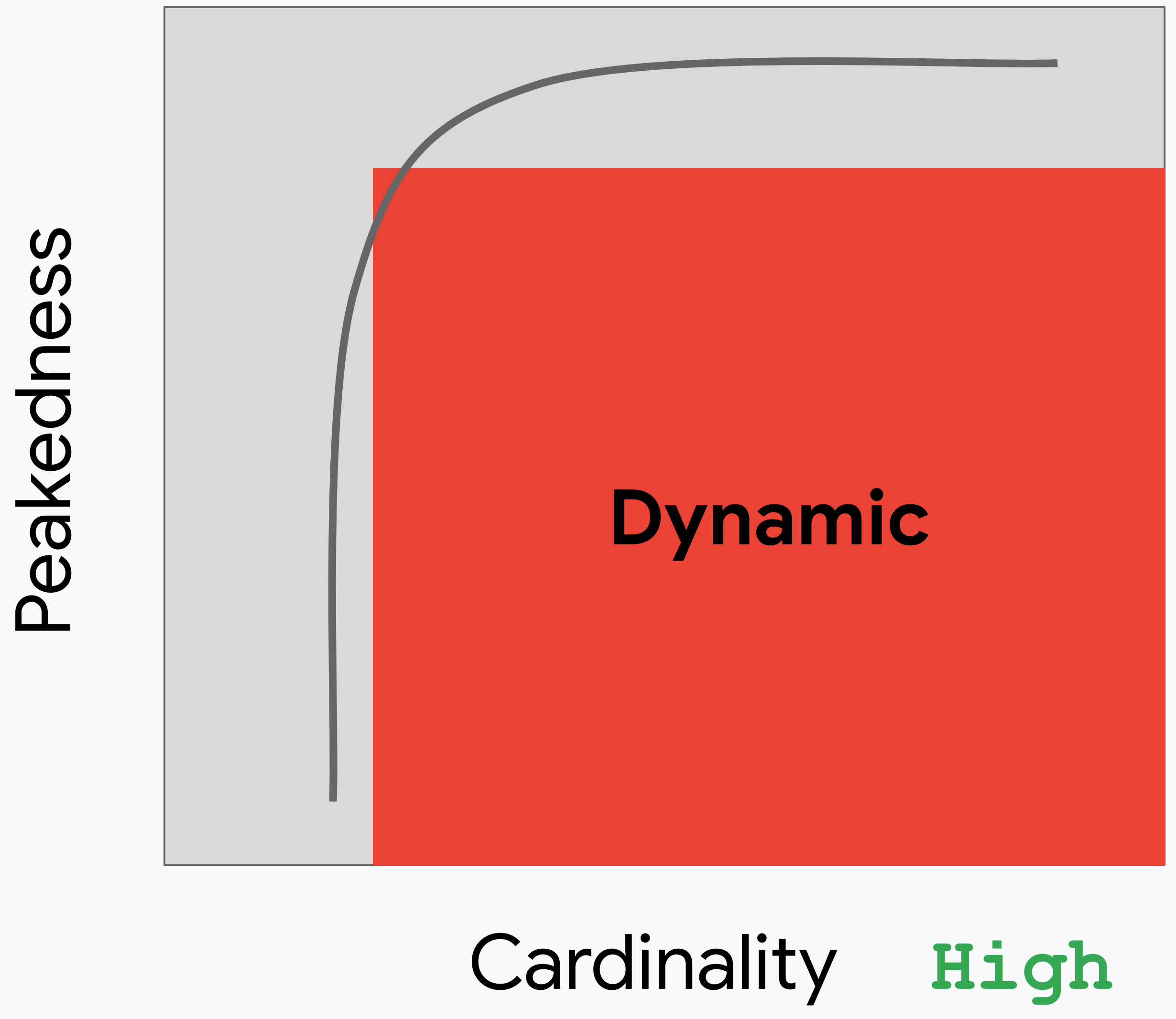


Peakedness and Cardinality space

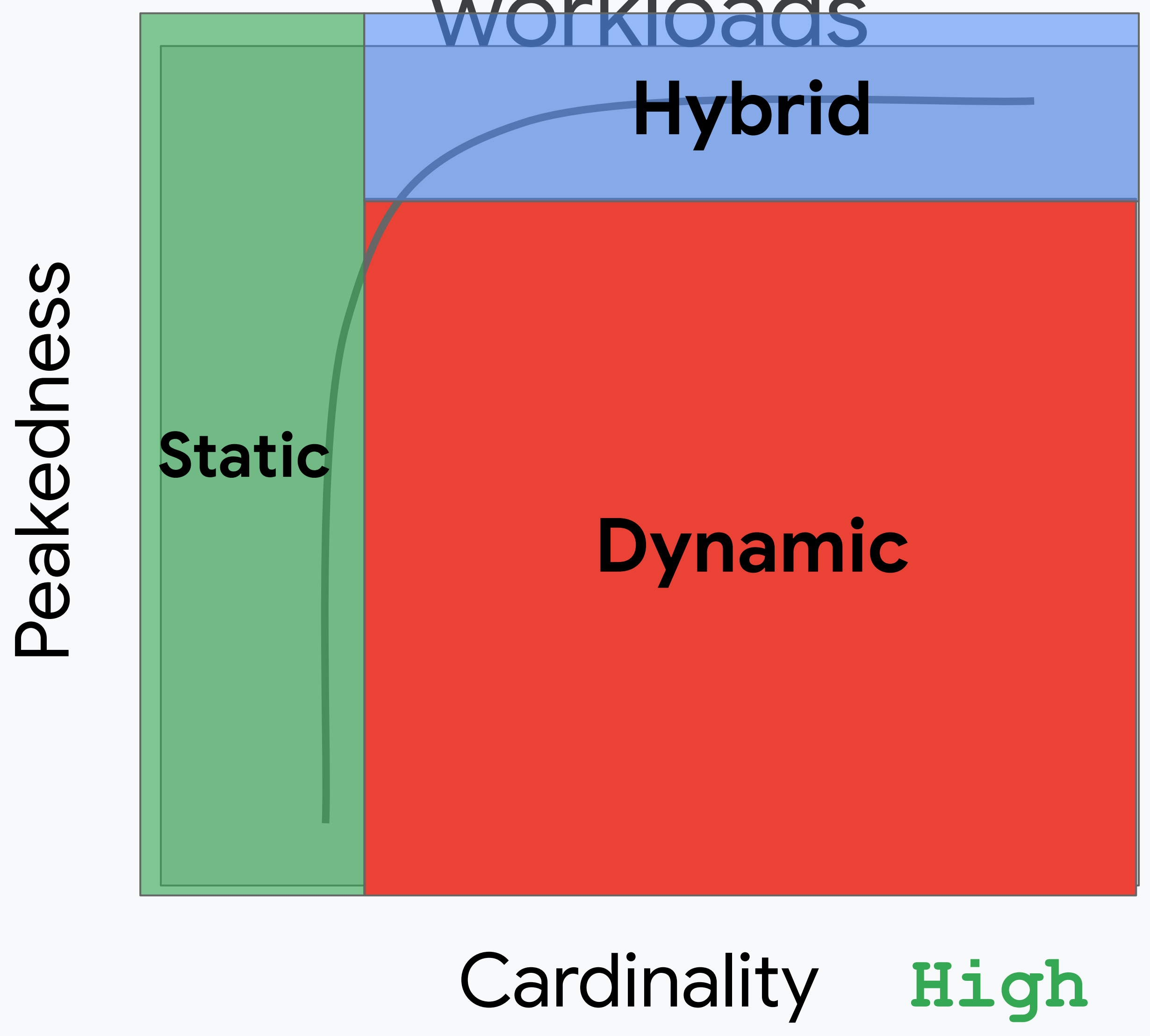


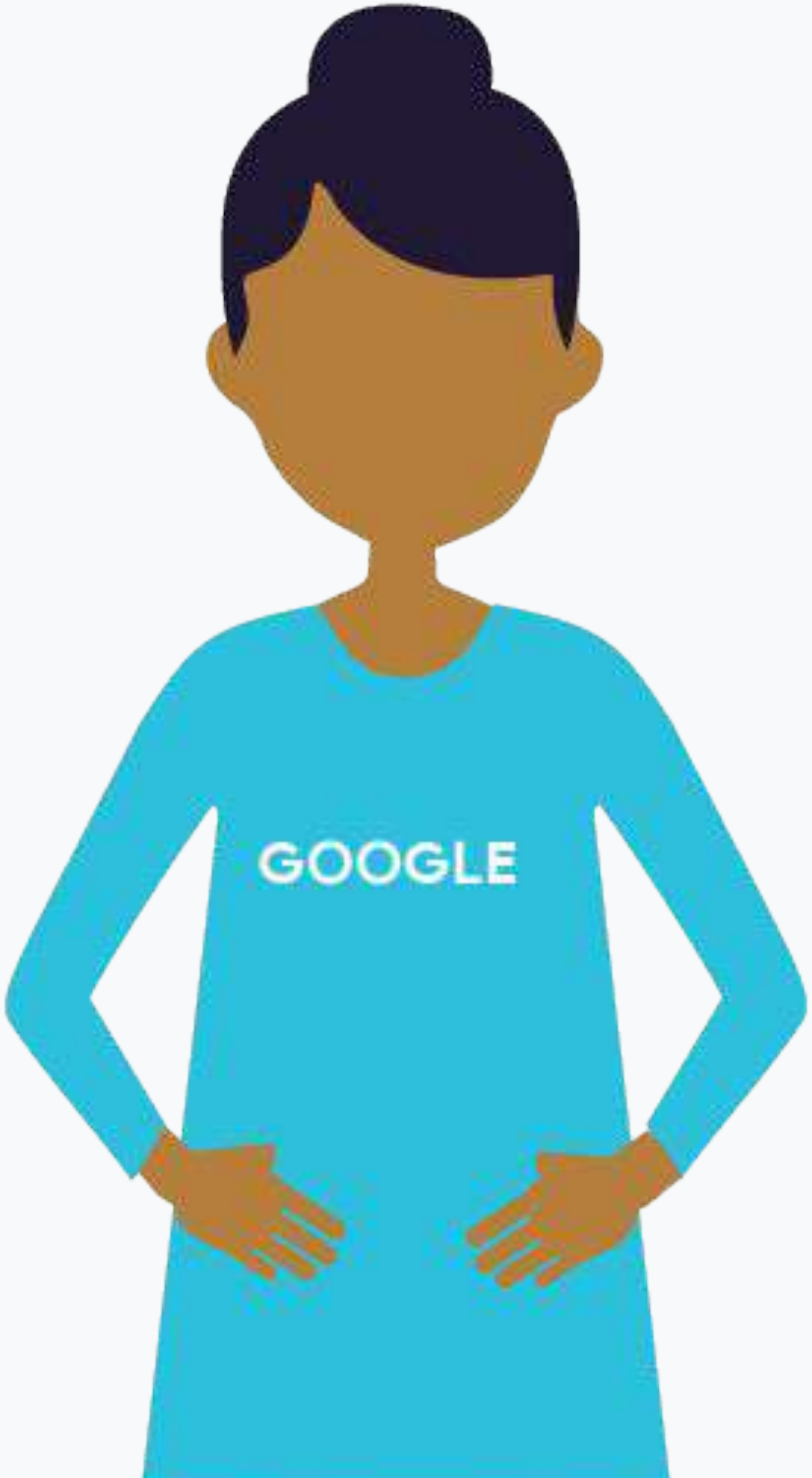


Peakedness and Cardinality space



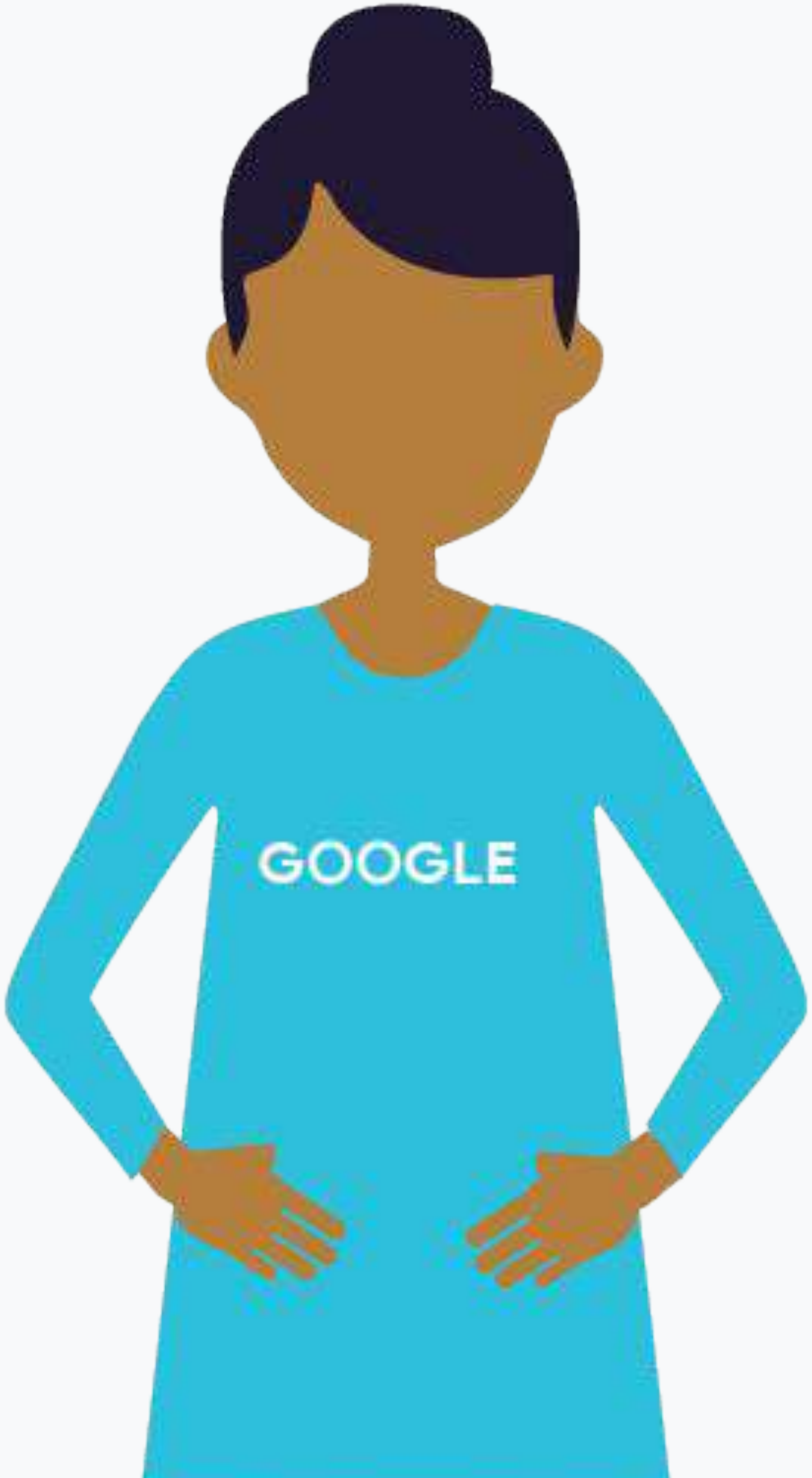
Hybrid solutions optimize for both types of prediction workloads





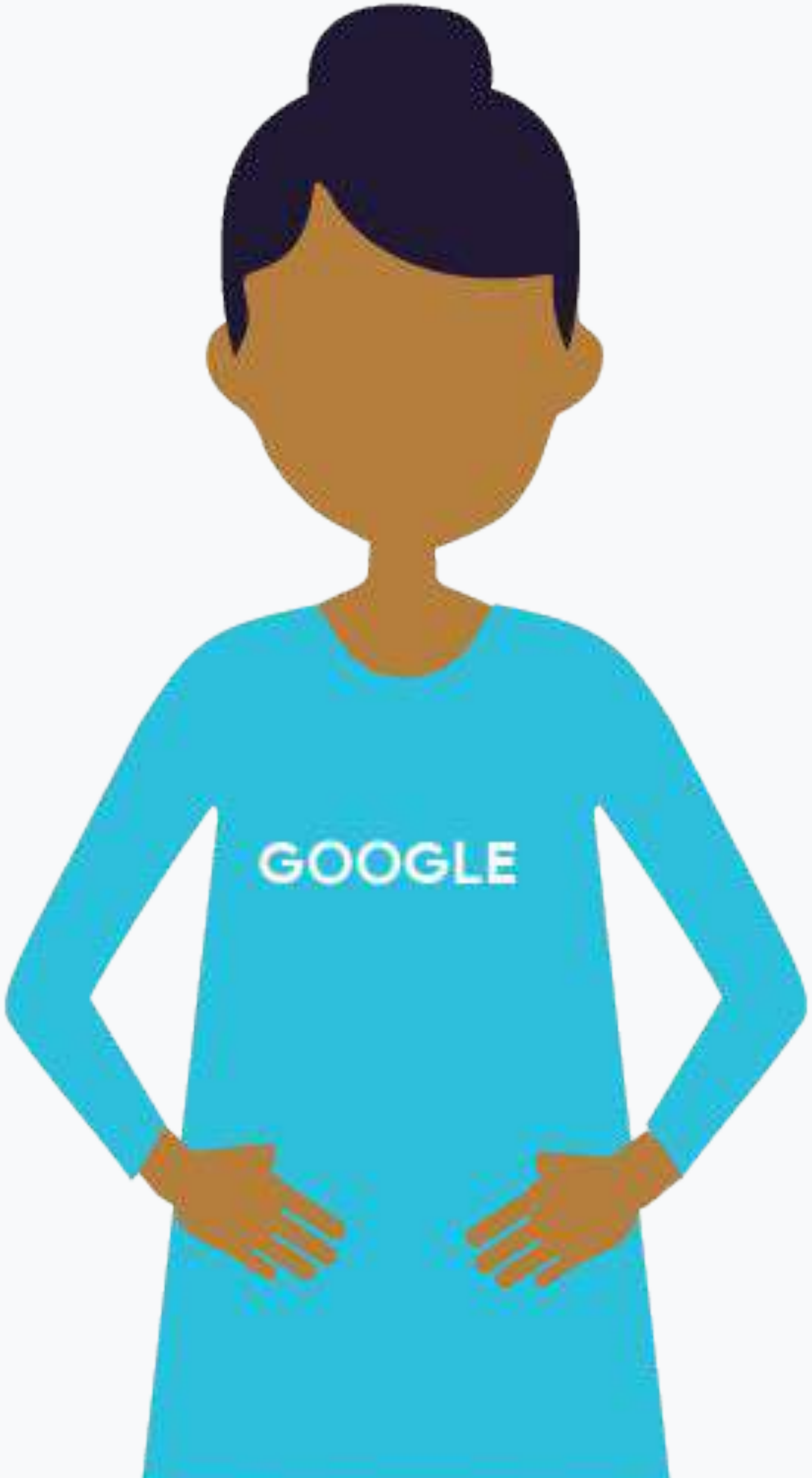
Lab: Estimate training and inference needs

Problem	Inference style (static or dynamic?)
Predict whether email is spam	
Android voice to text	
Shopping ad conversion rate	



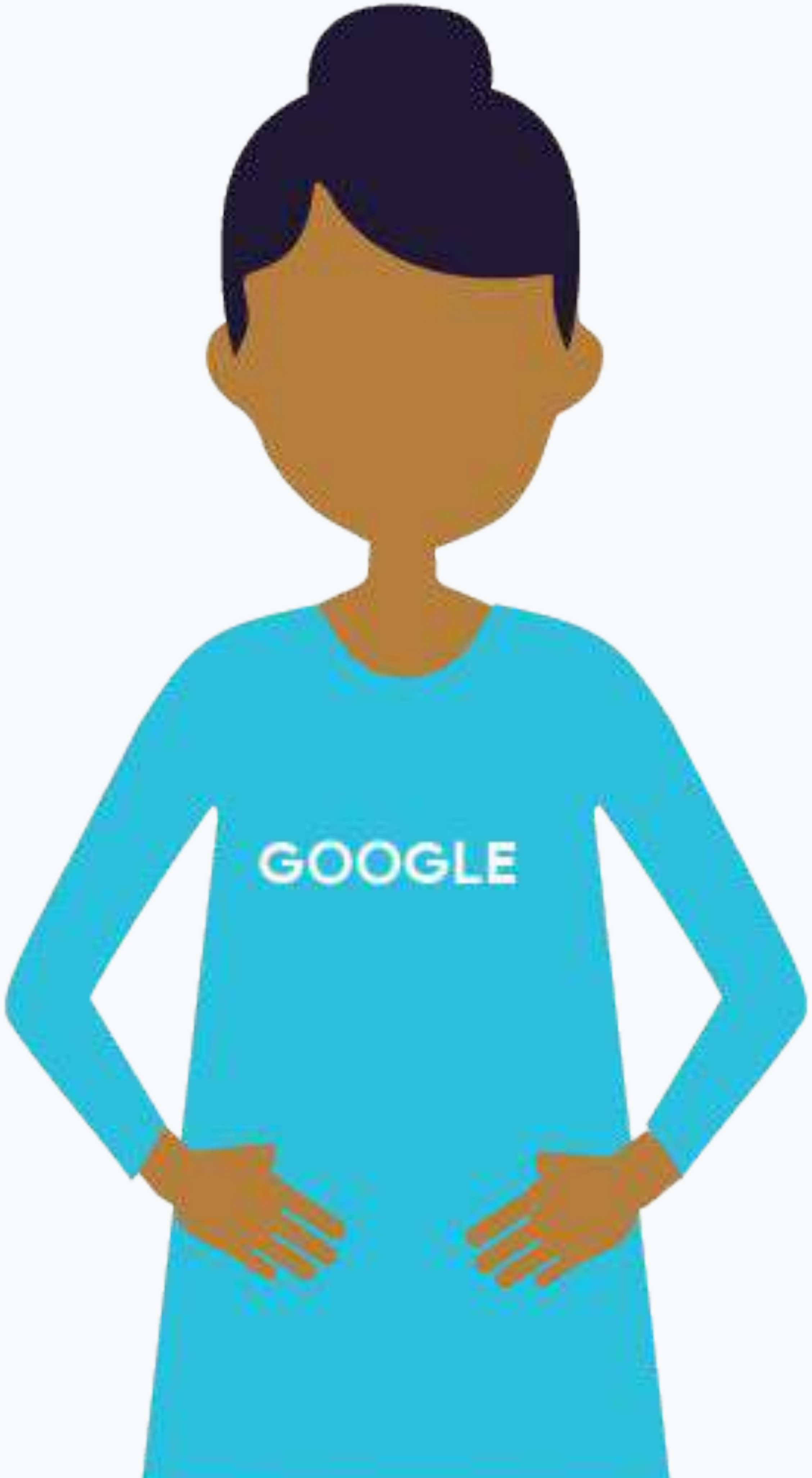
Lab: Estimate training and inference needs

Problem	Inference style (static or dynamic?)
Predict whether email is spam	**Dynamic**
Android voice to text	
Shopping ad conversion rate	



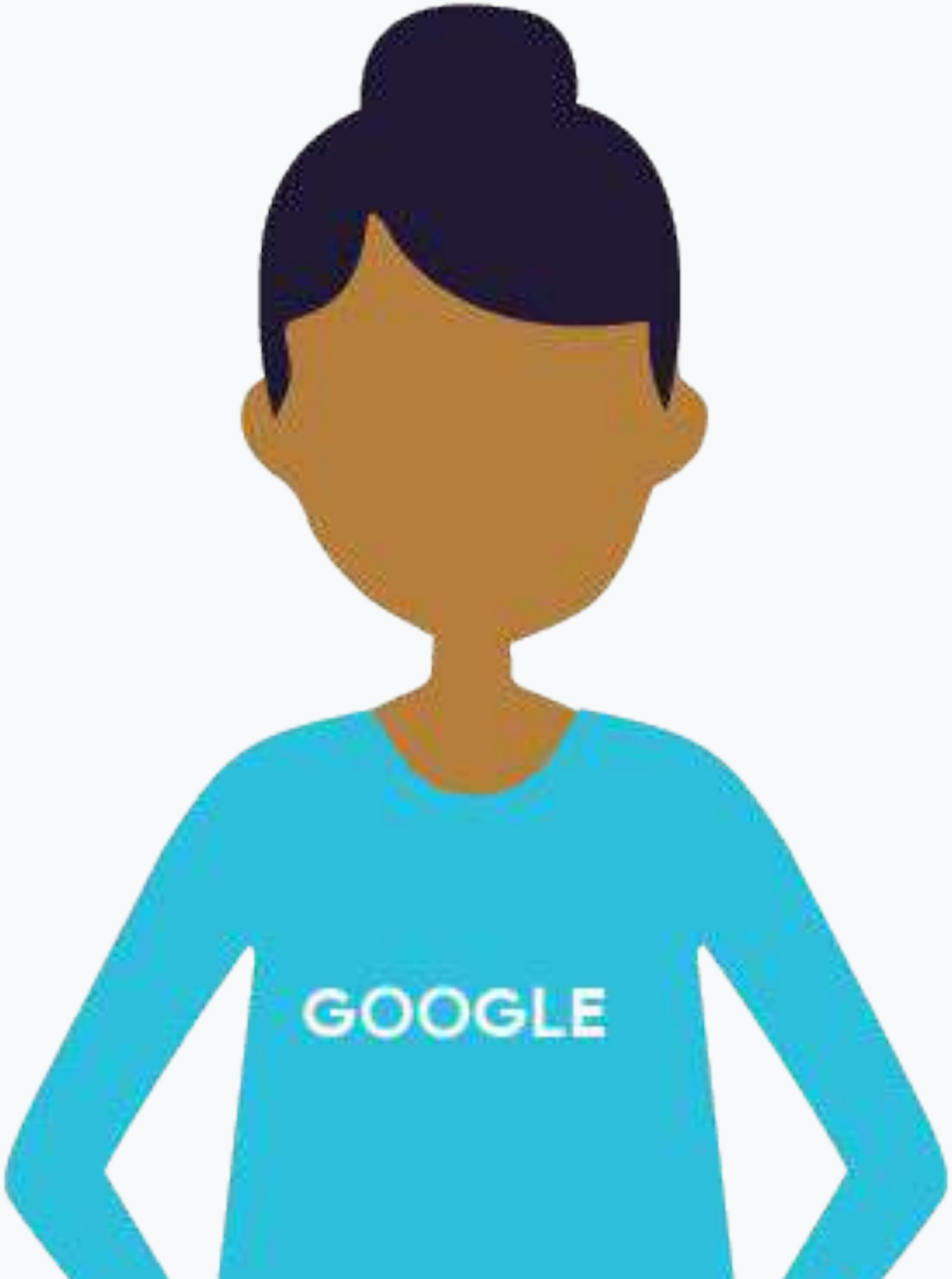
Lab: Estimate training and inference needs

Problem	Inference style (static or dynamic?)
Predict whether email is spam	**Dynamic**
Android voice to text	**Dynamic / Hybrid**
Shopping ad conversion rate	

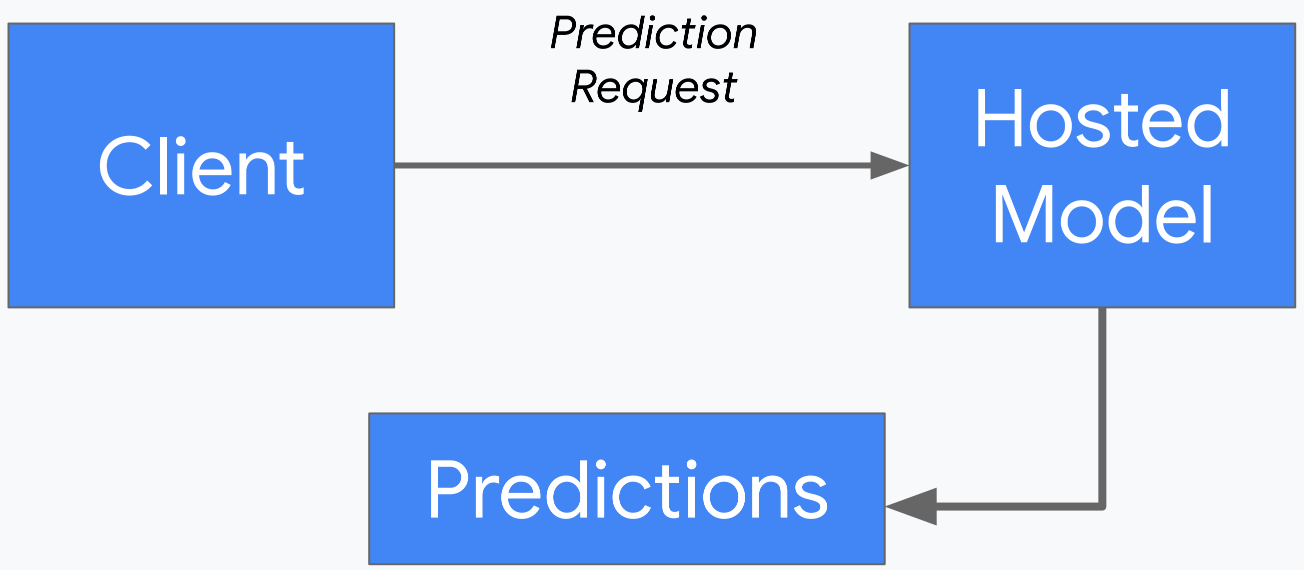


Lab: Estimate training and inference needs

Problem	Inference style (static or dynamic?)
Predict whether email is spam	**Dynamic**
Android voice to text	**Dynamic / Hybrid**
Shopping ad conversion rate	**Static**


GOOGLE

Dynamic




```
gcloud ml-engine predict --model $MODEL_NAME  \
                   --version $VERSION_NAME \
                   --json-instances $INPUT_DATA_FILE
```

Architecting a Static Serving Model

1. Change Cloud MLE from online to batch prediction job

2. Model accepts and passes keys as input

3. Write predictions to a data warehouse (e.g. BigQuery)

Course 2: Production ML Systems

Module 1: Architecting Production ML Systems

Lesson Title: **Serving on CMLE**

Presenter: Max Lotstein

Format: Talking Head

Video Name: T-PSML-O_1_l12_serving_on_cloud_mle

Agenda

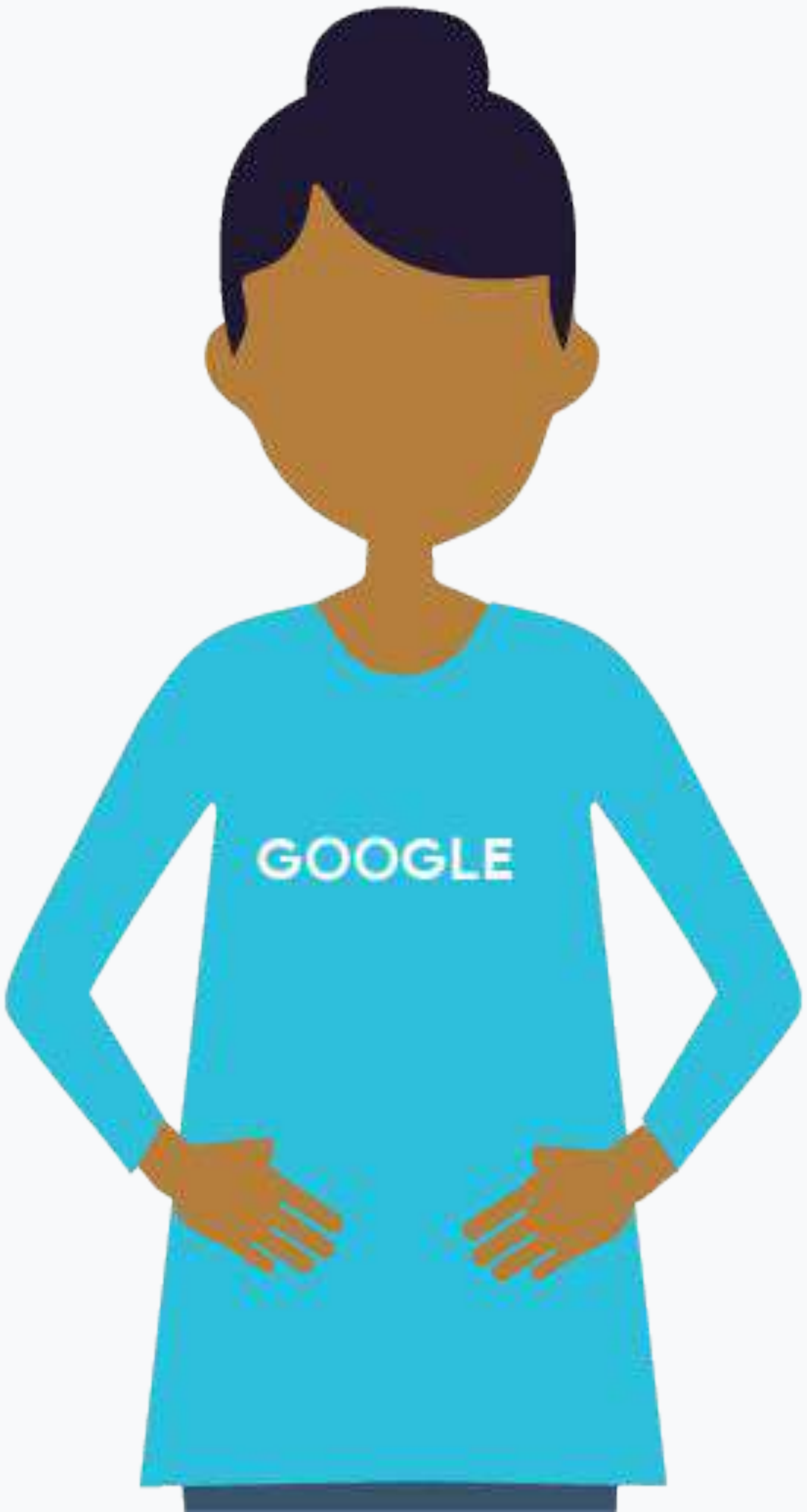
What's in a Production ML System

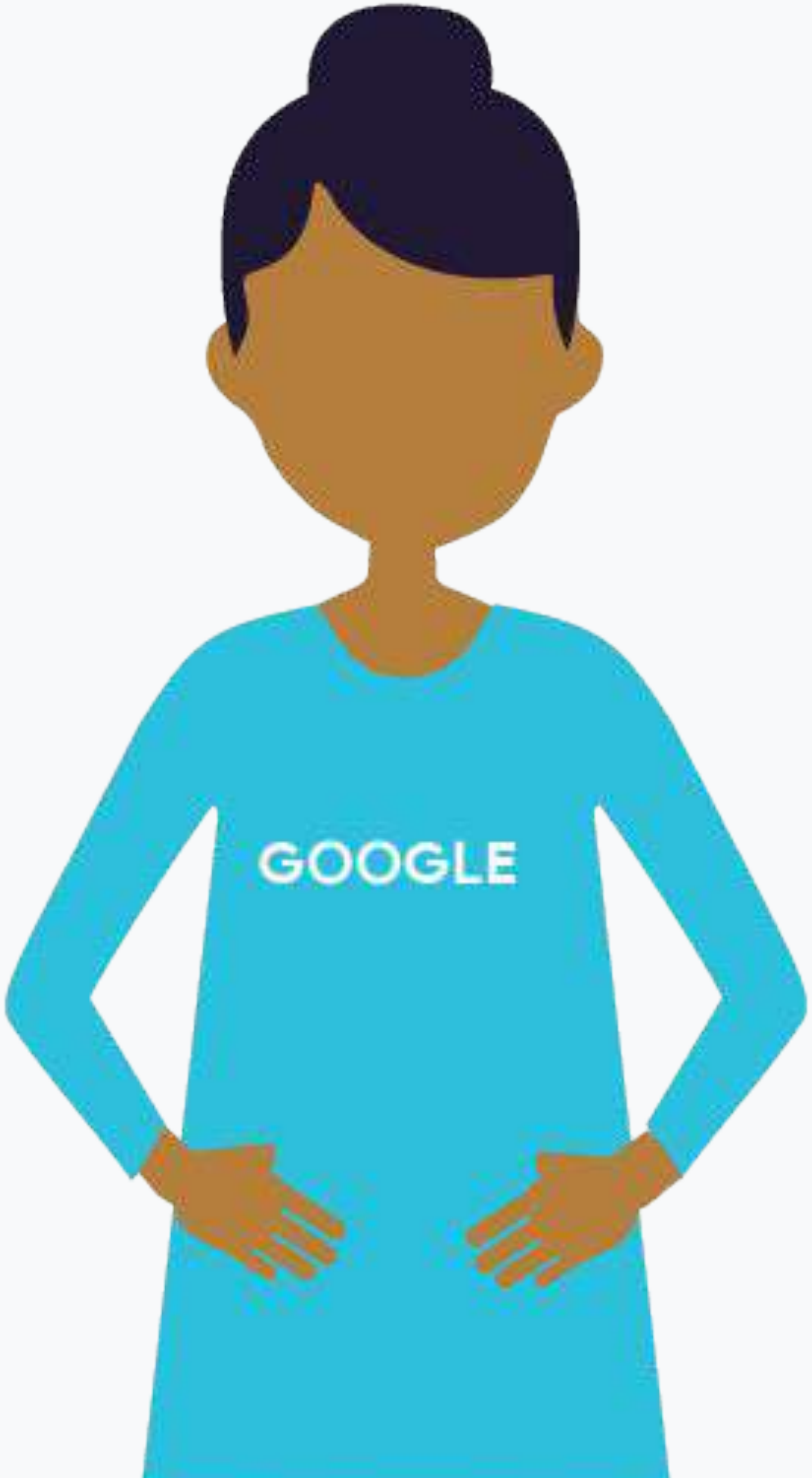
Training Design Decisions

Serving Design Decisions

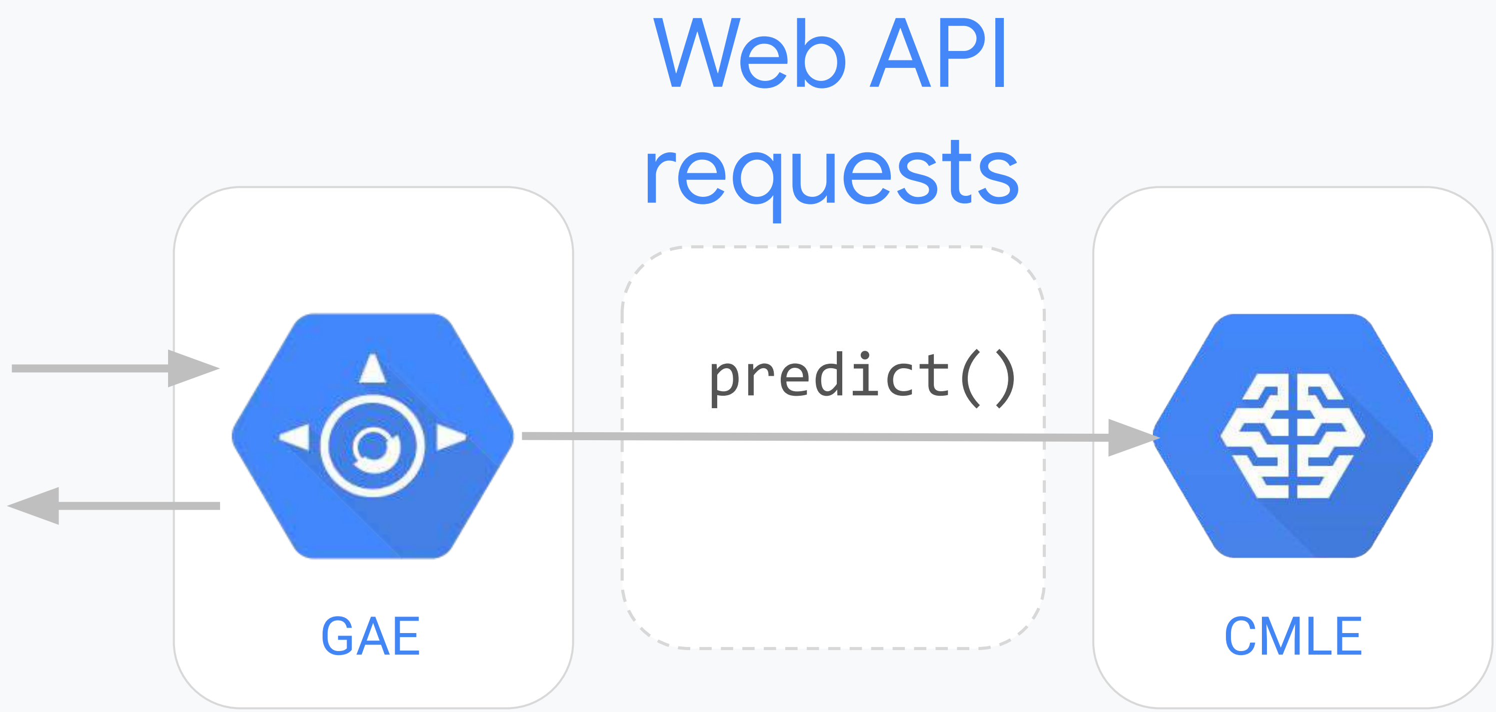
Serving on CMLE

Designing an Architecture from Scratch





Lab: Invoking ML Predictions with AppEngine



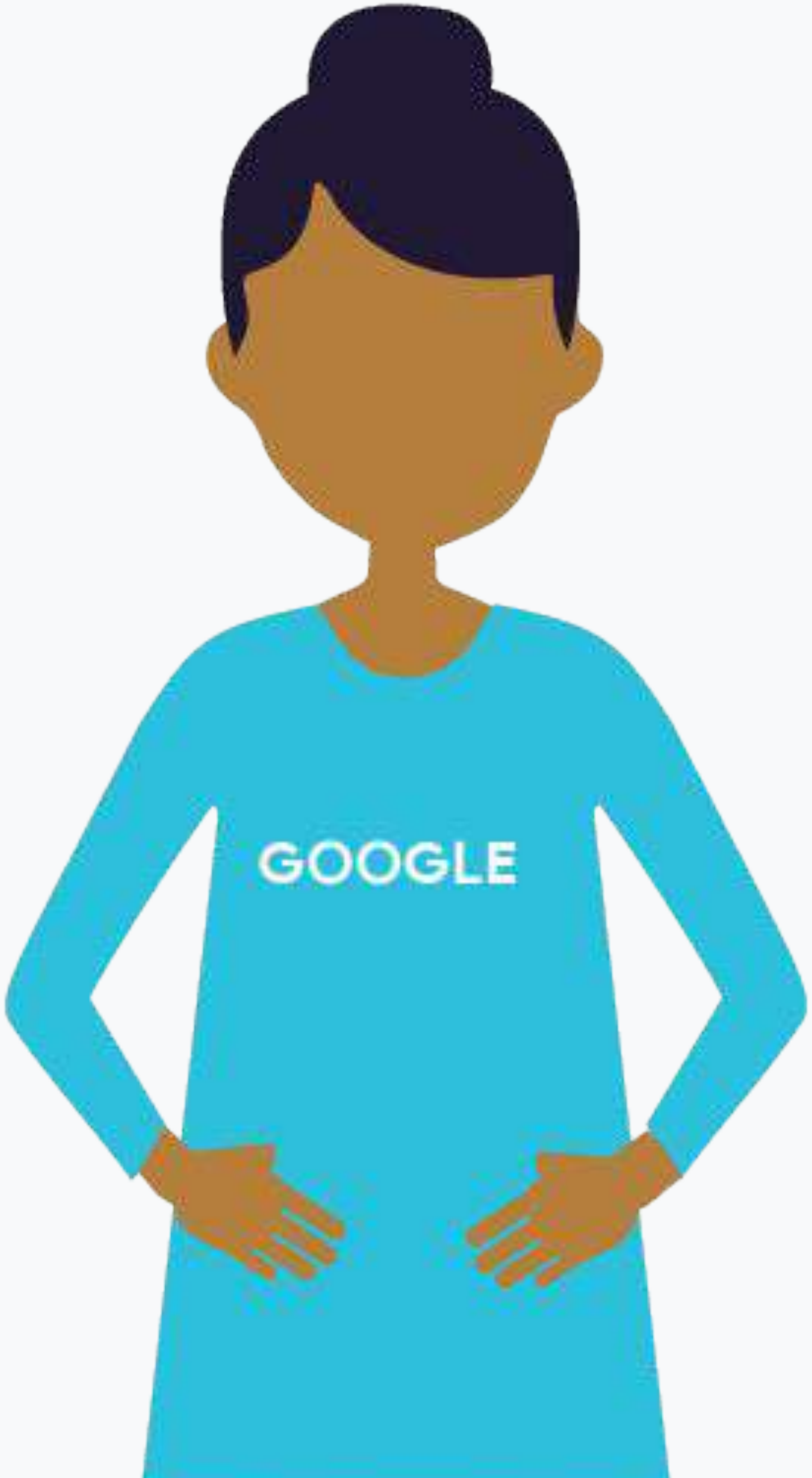
Lab

Build an AppEngine app to serve ML predictions

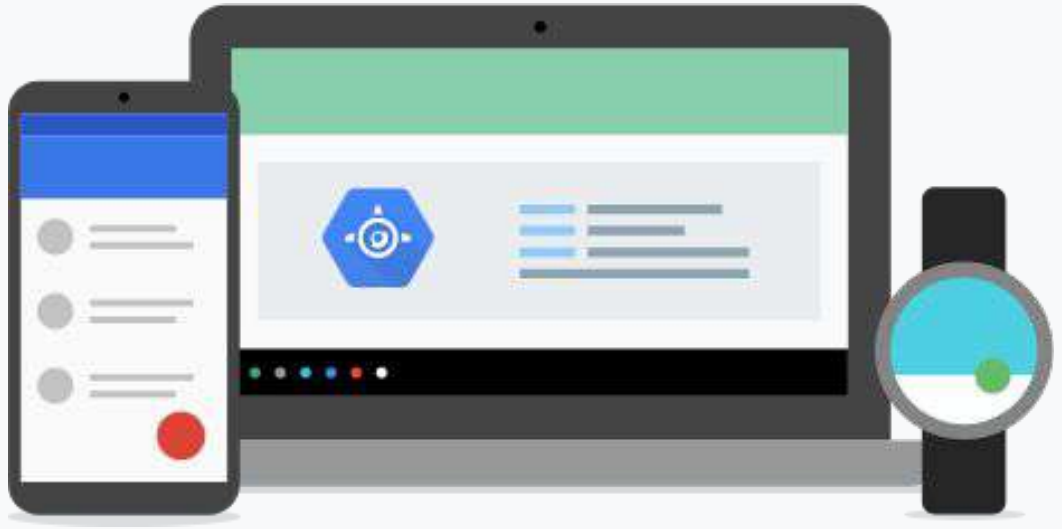
Max Lotstein

Title Safe >

< Action Safe

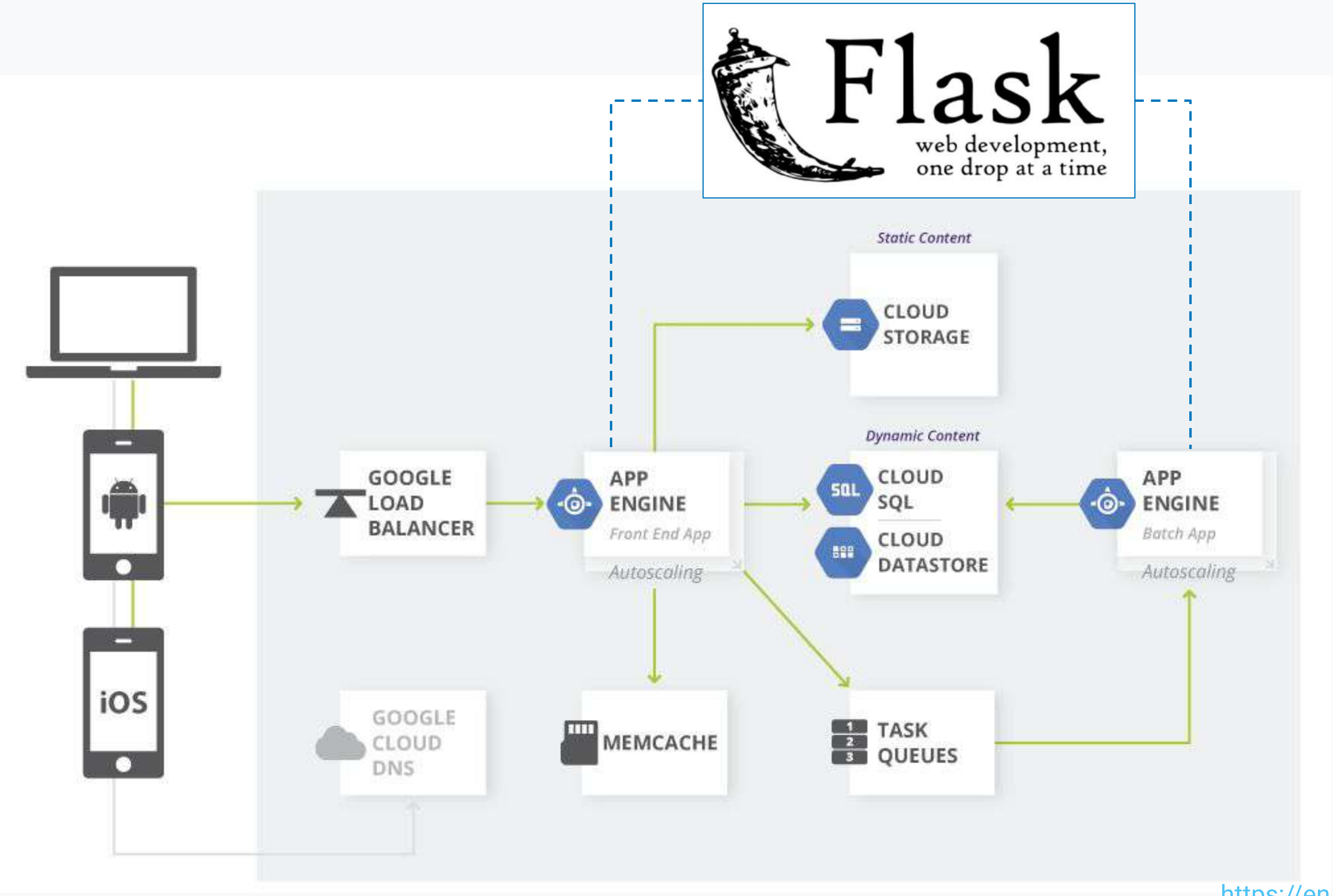


Google App Engine is a fully-managed service for building web backends



Supports Java, Node.js, Ruby, C#, Go, Python, and PHP

The lab's App Engine application uses Flask to build backend



Flask is a Python framework that allows you to build web applications

Flask Logo
https://en.wikipedia.org/wiki/File:Flask_logo.svg


Baby weight predictor
Example application to predict a baby's weight.
Mother's race
Select
Mother's age
Gestation weeks
Plurality
Select
Baby's gender
Male
Female
Unmarried
Cigarette use
Alcohol use
PREDICT
Prediction

Course 2: Production ML Systems

Module 1: Architecting Production ML Systems

Lesson Title: **Lab Intro: Serving CMLE**

Presenter: Max Lotstein

Format: Screencast

Video Name:
T-PSML-O_1_l13_lab_intro:_serving_on_cloud_mle

Course 2: Production ML Systems

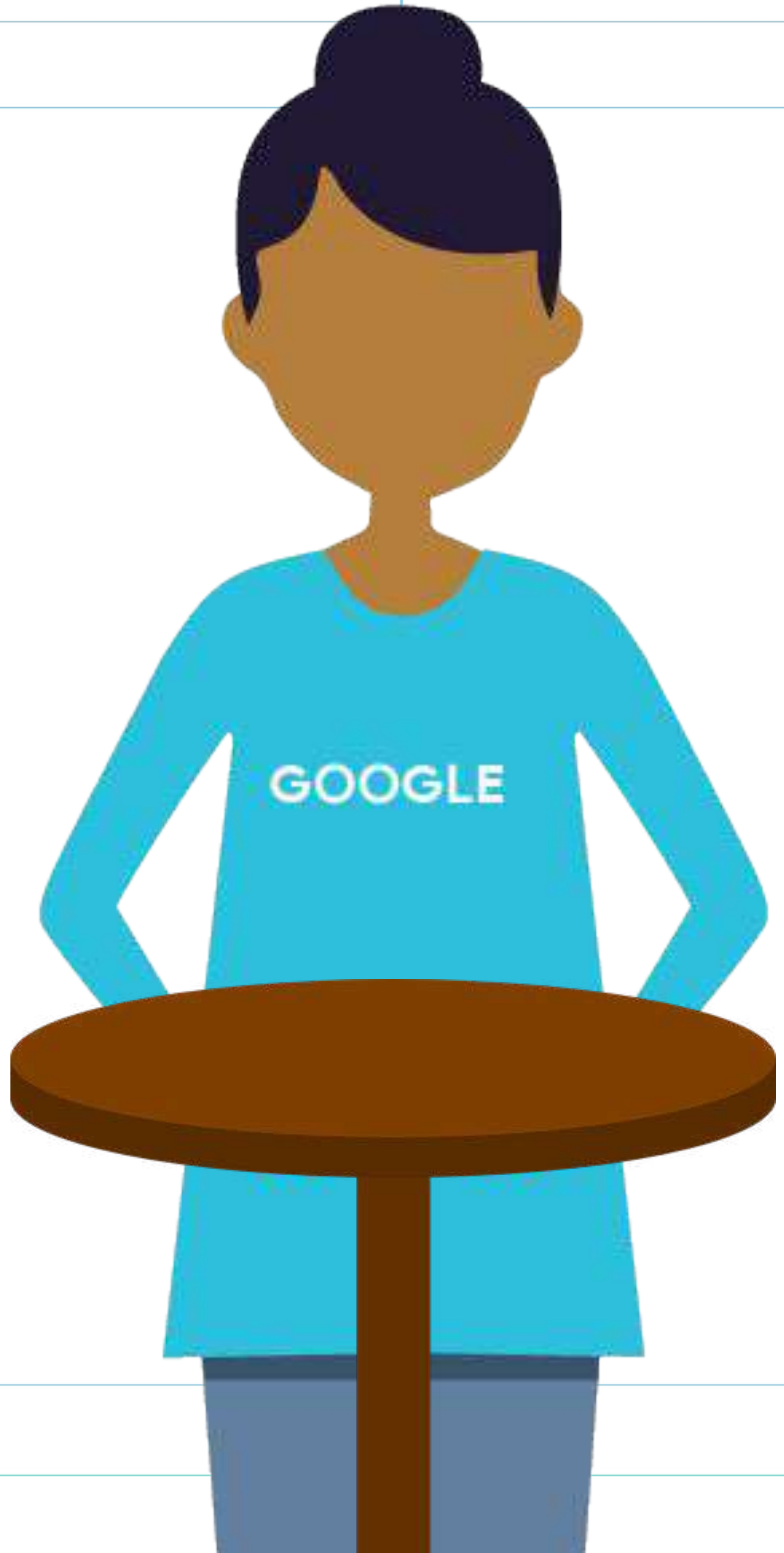
Module 1: Architecting Production ML Systems

Lesson Title: **Lab Solution: Serving CMLE**

Presenter: Max Lotstein

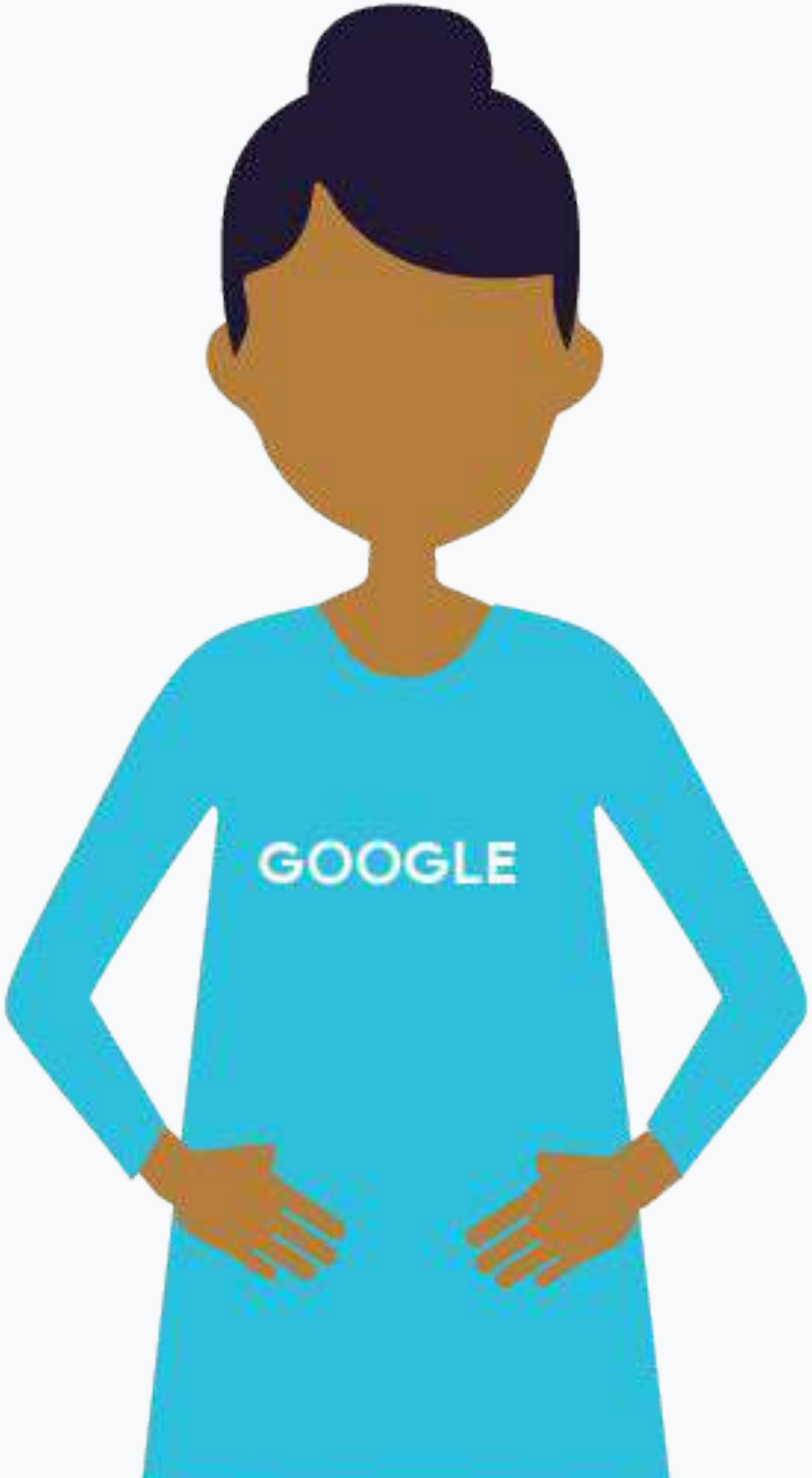
Format: Screencast

Video Name:
T-PSML-O_1_l14_lab_solution:_serving_on_cloud_mle

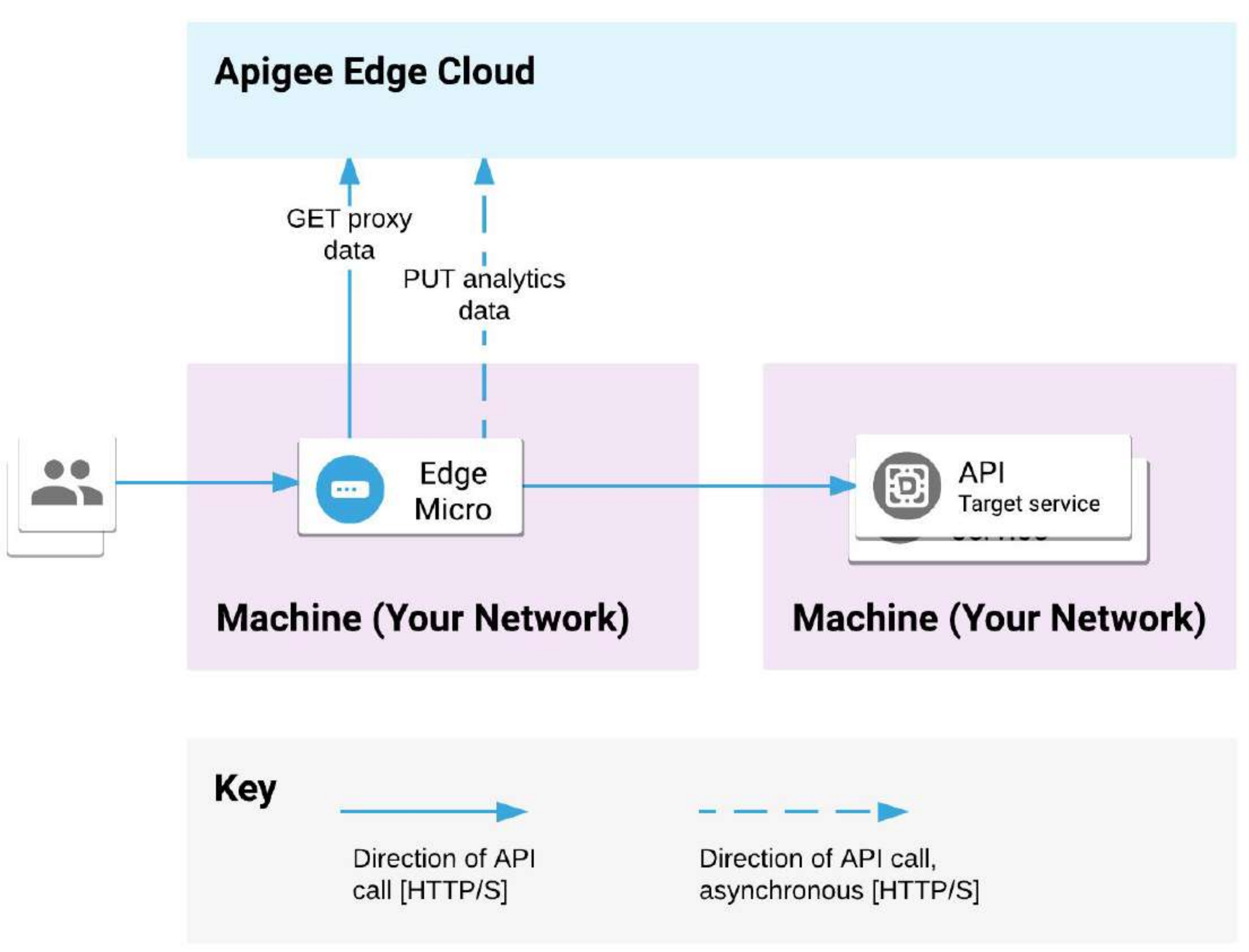


Title Safe >

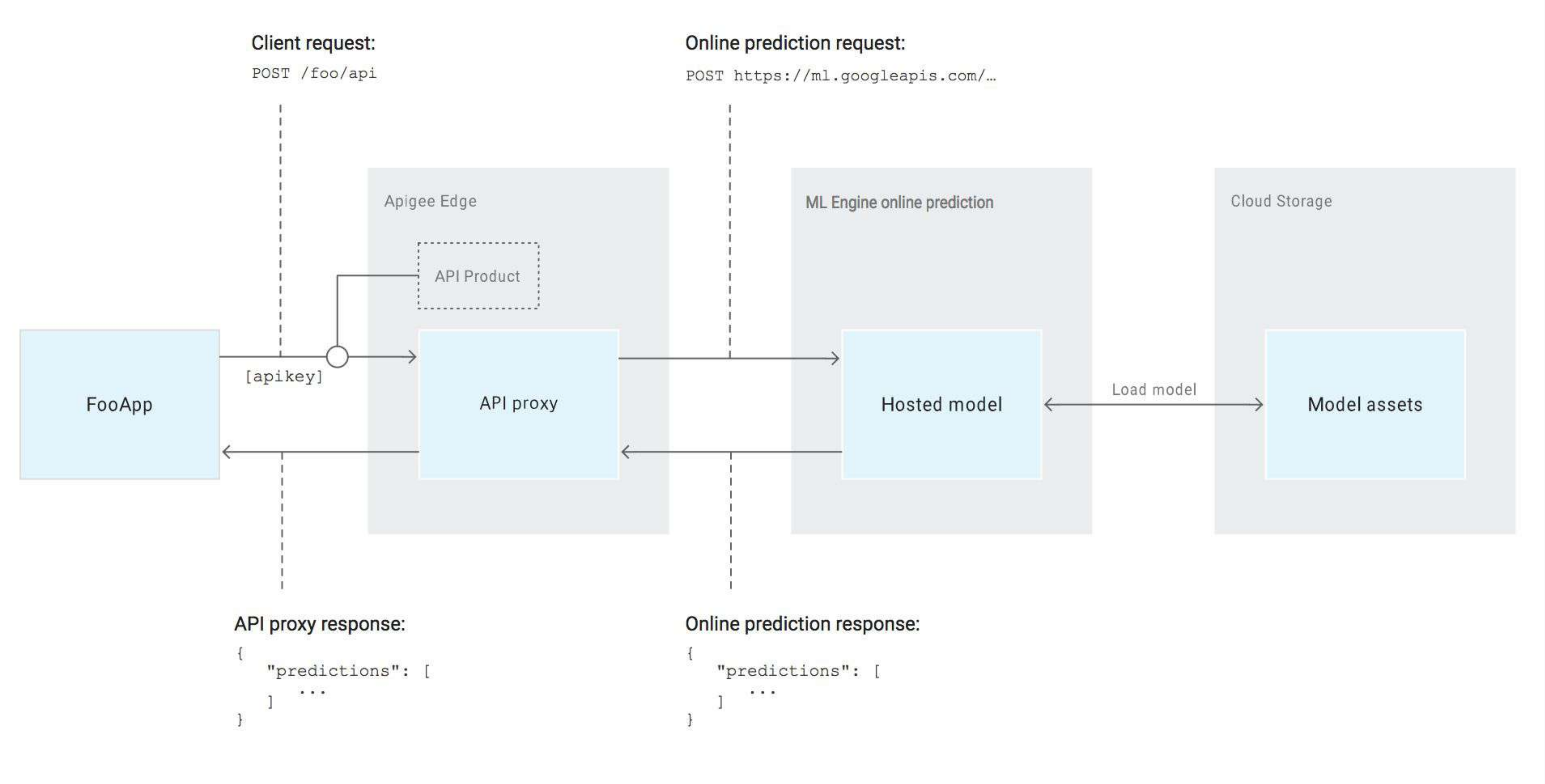
< Action Safe



Use Apigee Edge for full-fledged APIs



Serving ML Models Using Apigee Edge and Cloud ML Engine



Course 2: Production ML Systems

Module 1: Architecting Production ML Systems

Lesson Title: **Designing from Scratch**

Presenter: Max Lotstein

Format: Screencast

Video Name: T-PSML-O_1_l15_designing_from_scratch

Agenda

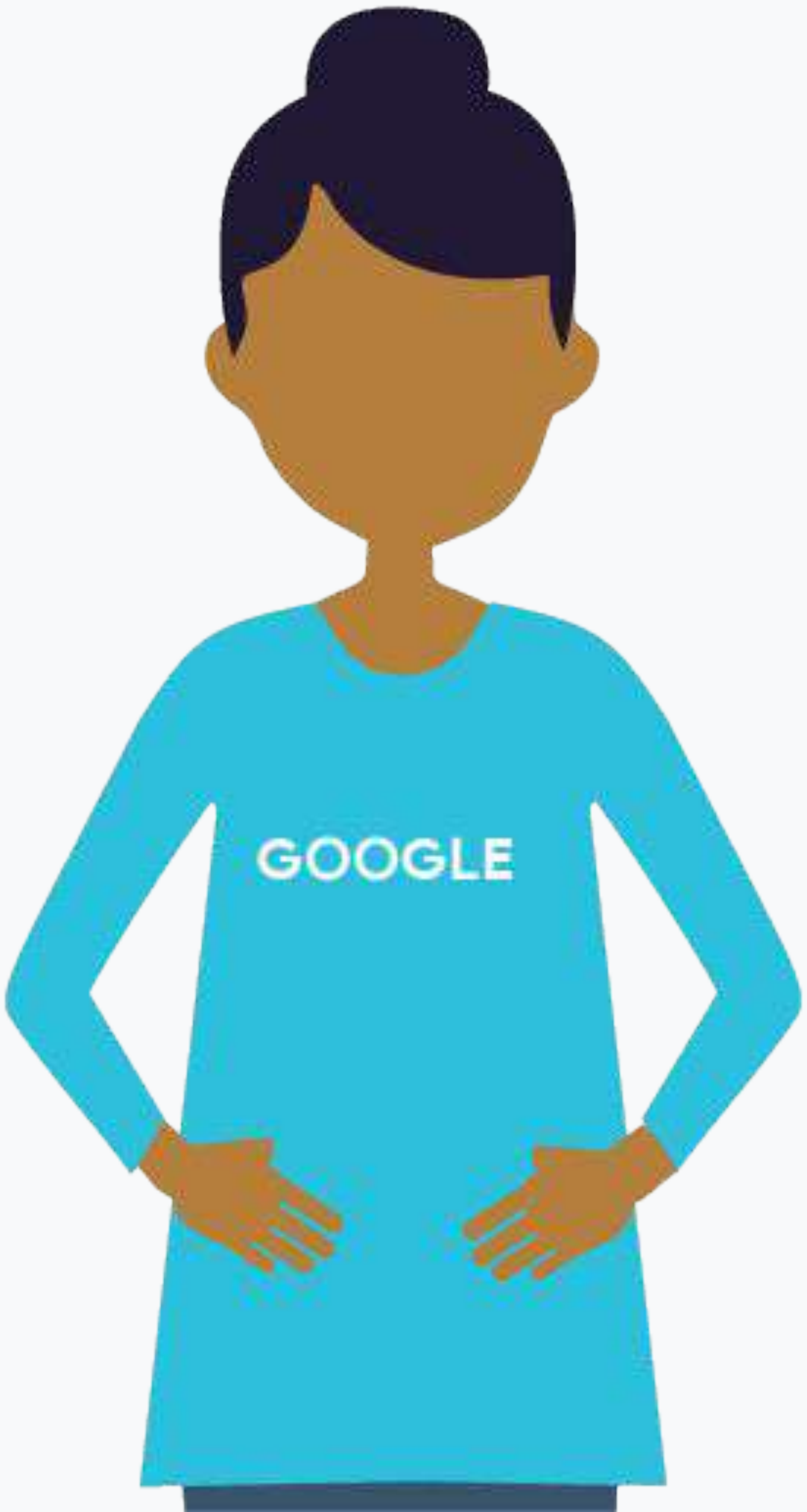
What's in a Production ML System

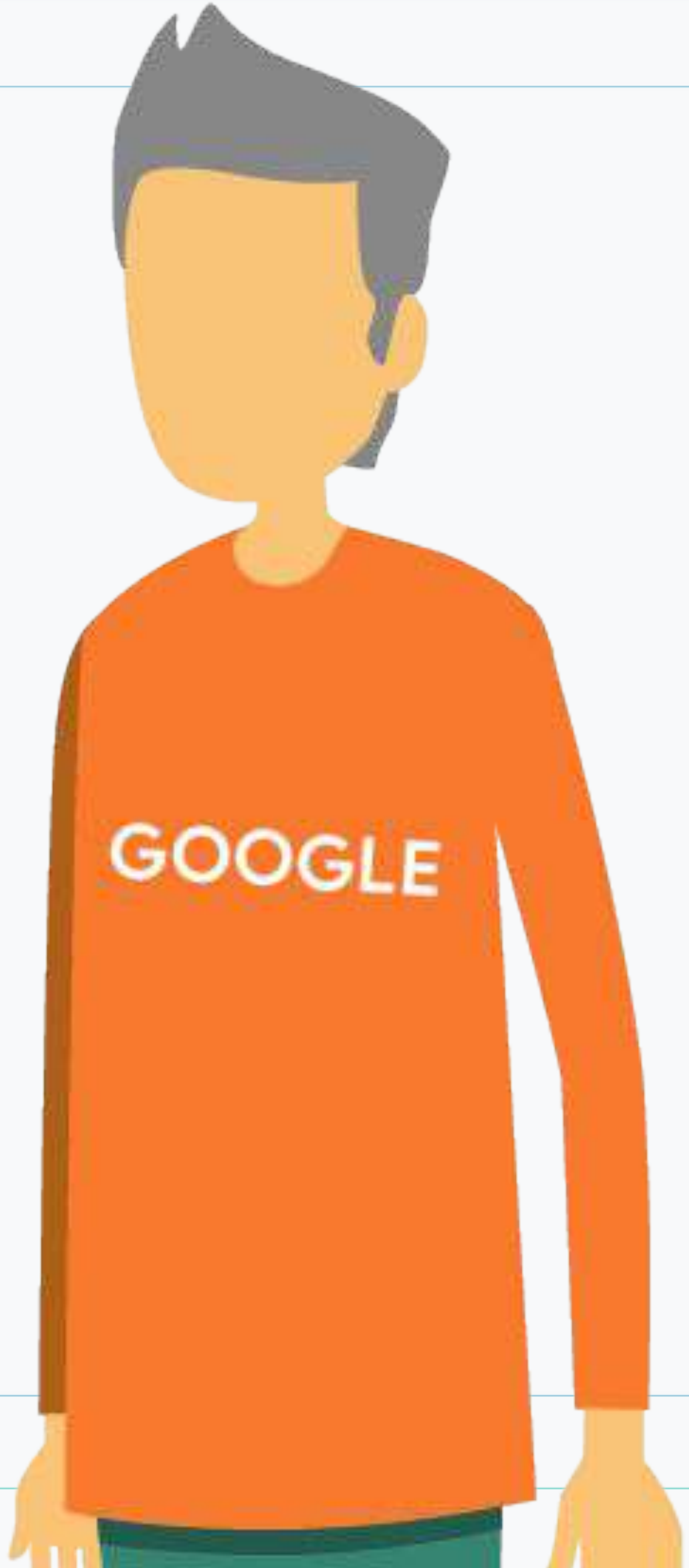
Training Design Decisions

Serving Design Decisions

Serving on CMLE

Designing an Architecture from Scratch

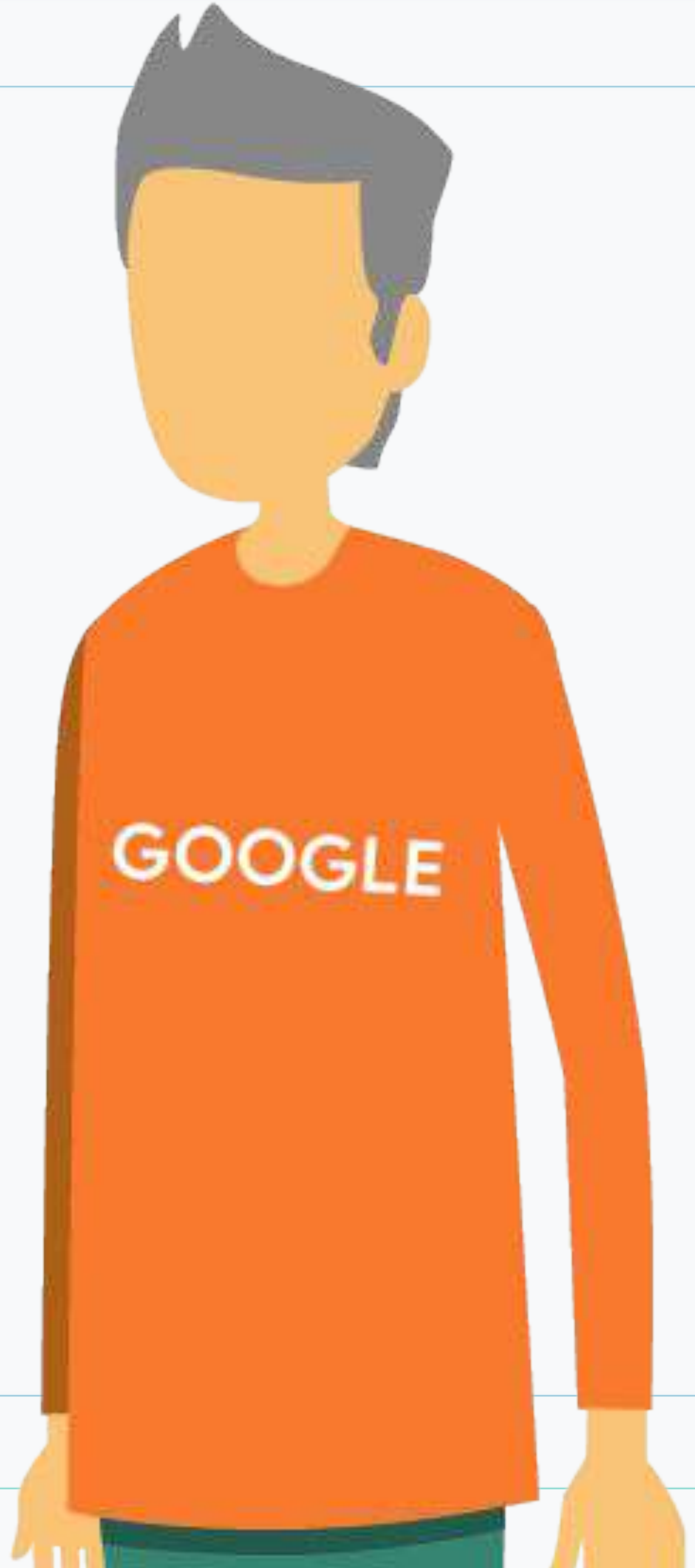




Lab: Build a system that predicts the traffic levels on roads

Title Safe >

< Action Safe



Lab: Build a system that predicts the traffic levels on roads

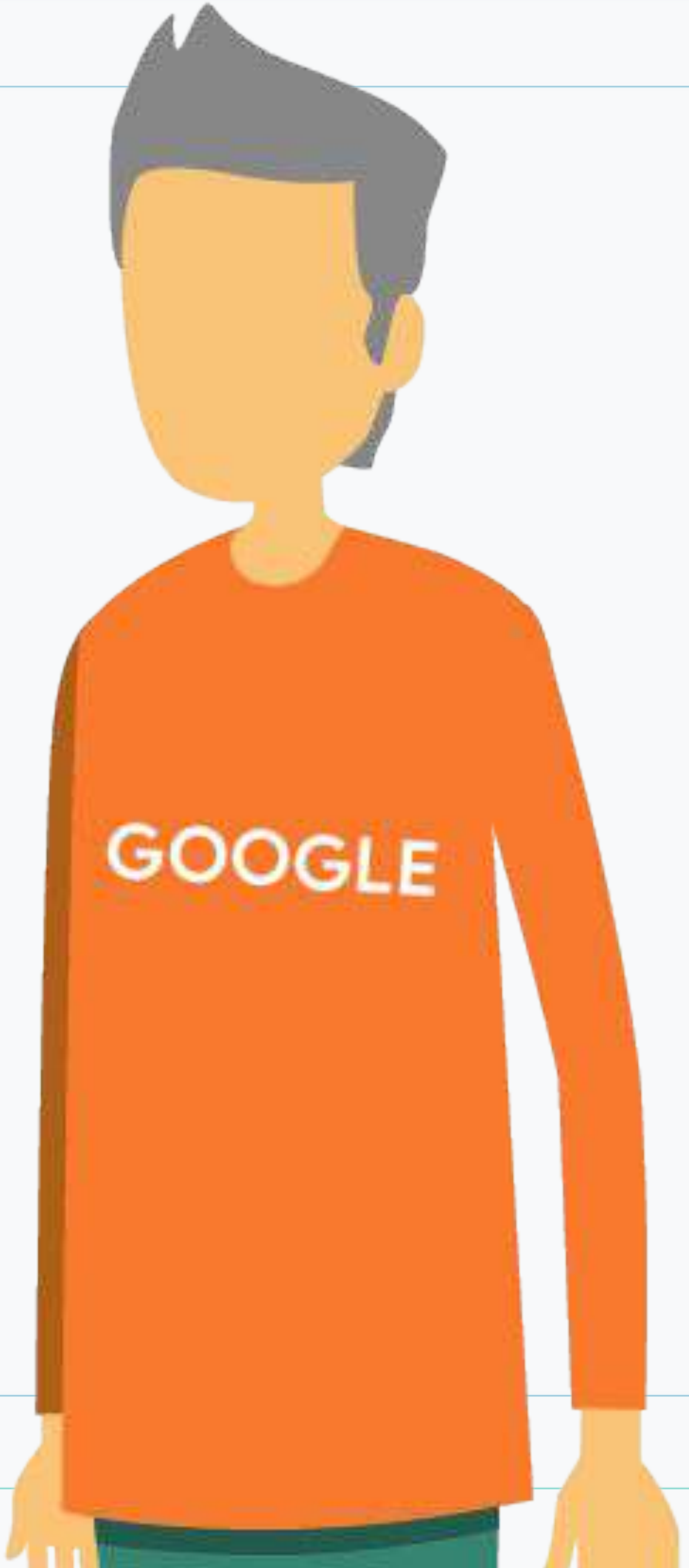
Available data: Traffic sensors deployed all over the city

Title Safe >

< Action Safe

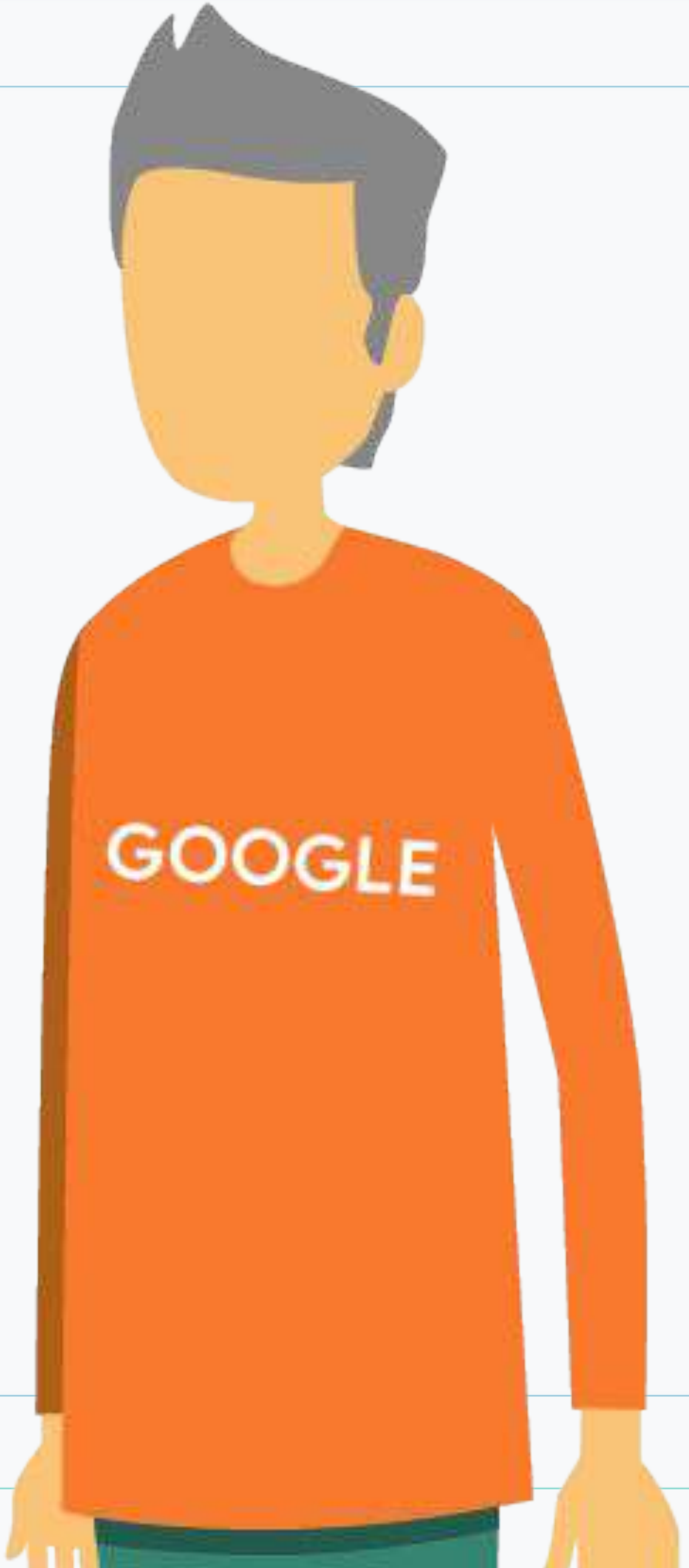
Lab: Build a system that predicts the traffic levels on roads

What sort of training architecture is appropriate?



Title Safe >

< Action Safe

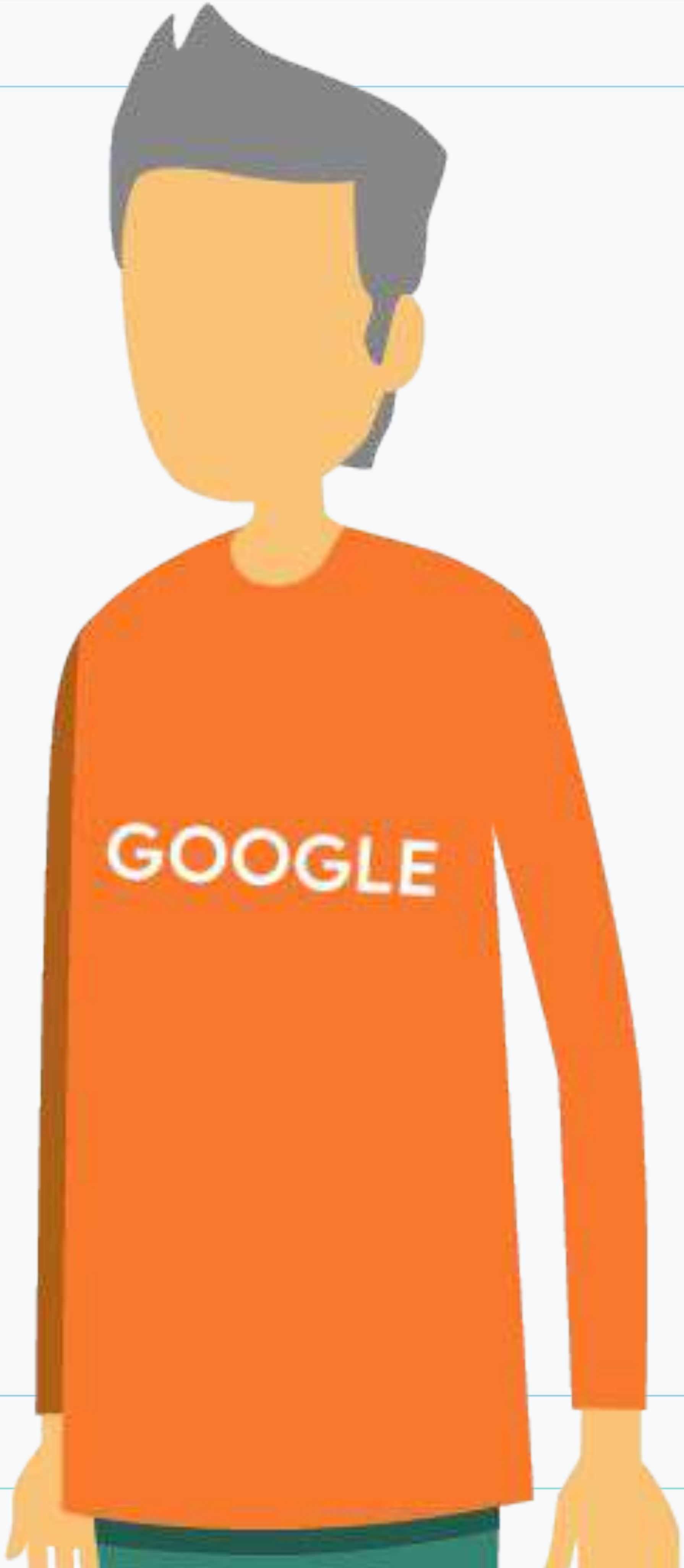


Lab: Build a system that predicts the traffic levels on roads

What is the relationship between the features and labels like?

Title Safe >

< Action Safe

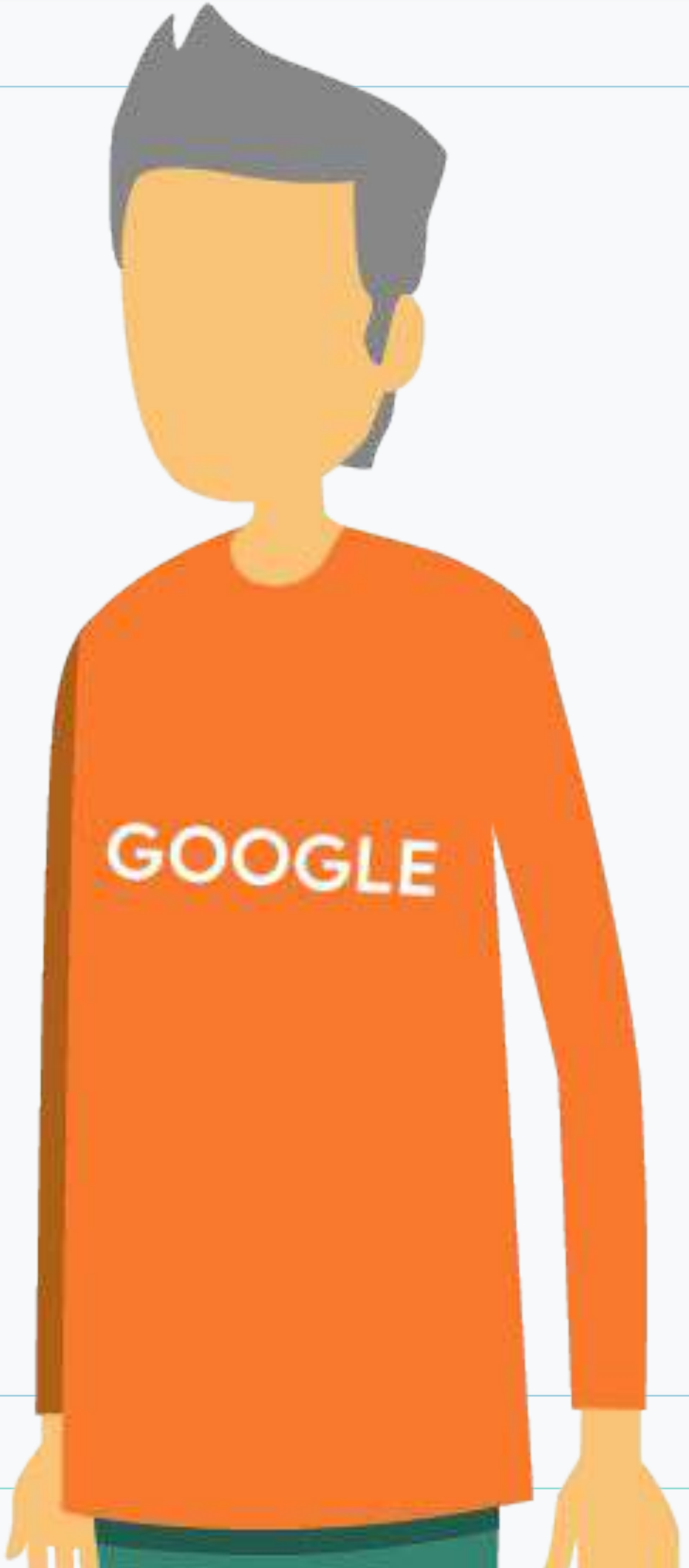


Lab: Build a system that predicts the traffic levels on roads

Which sort of serving architecture is appropriate?

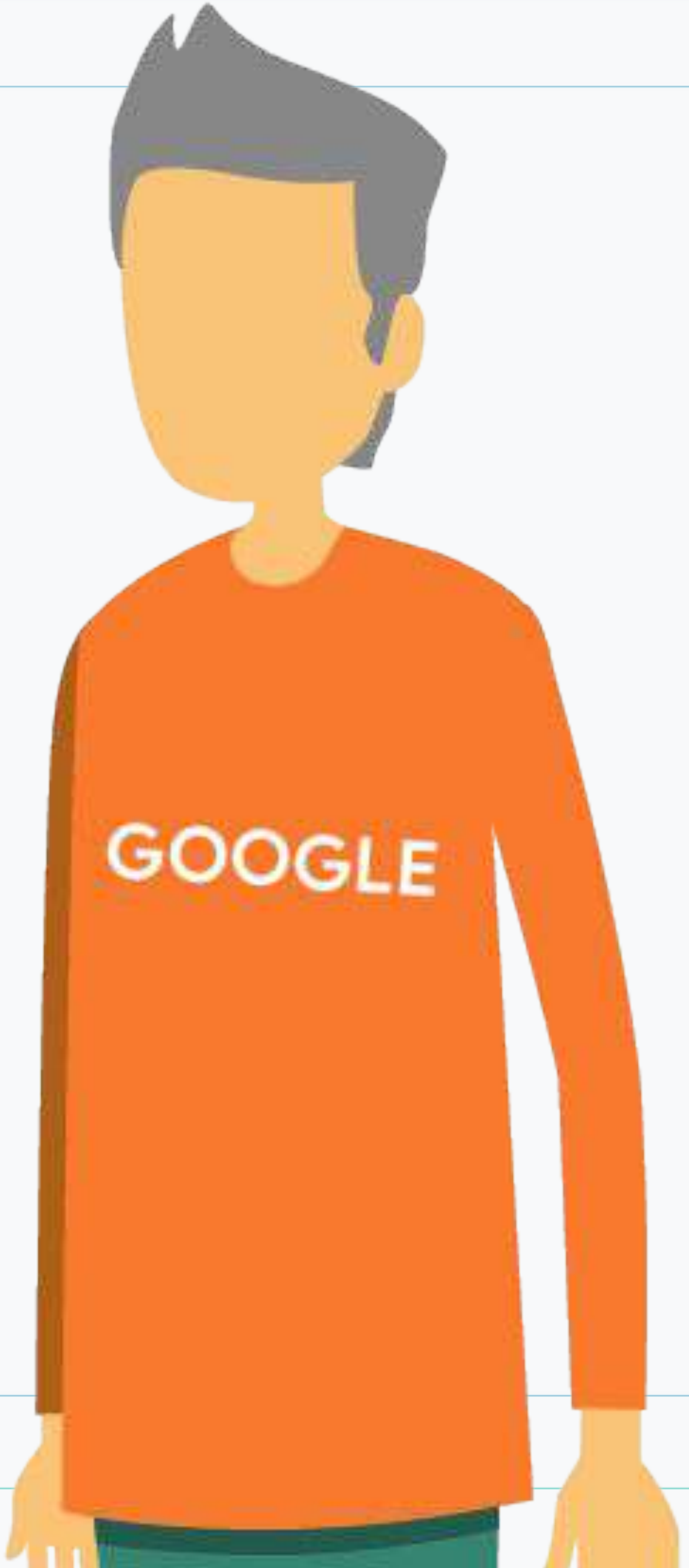
Title Safe >

< Action Safe



Lab: Build a system that predicts the traffic levels on roads

Is the distribution of prediction requests likely to be more peaked or more flat?

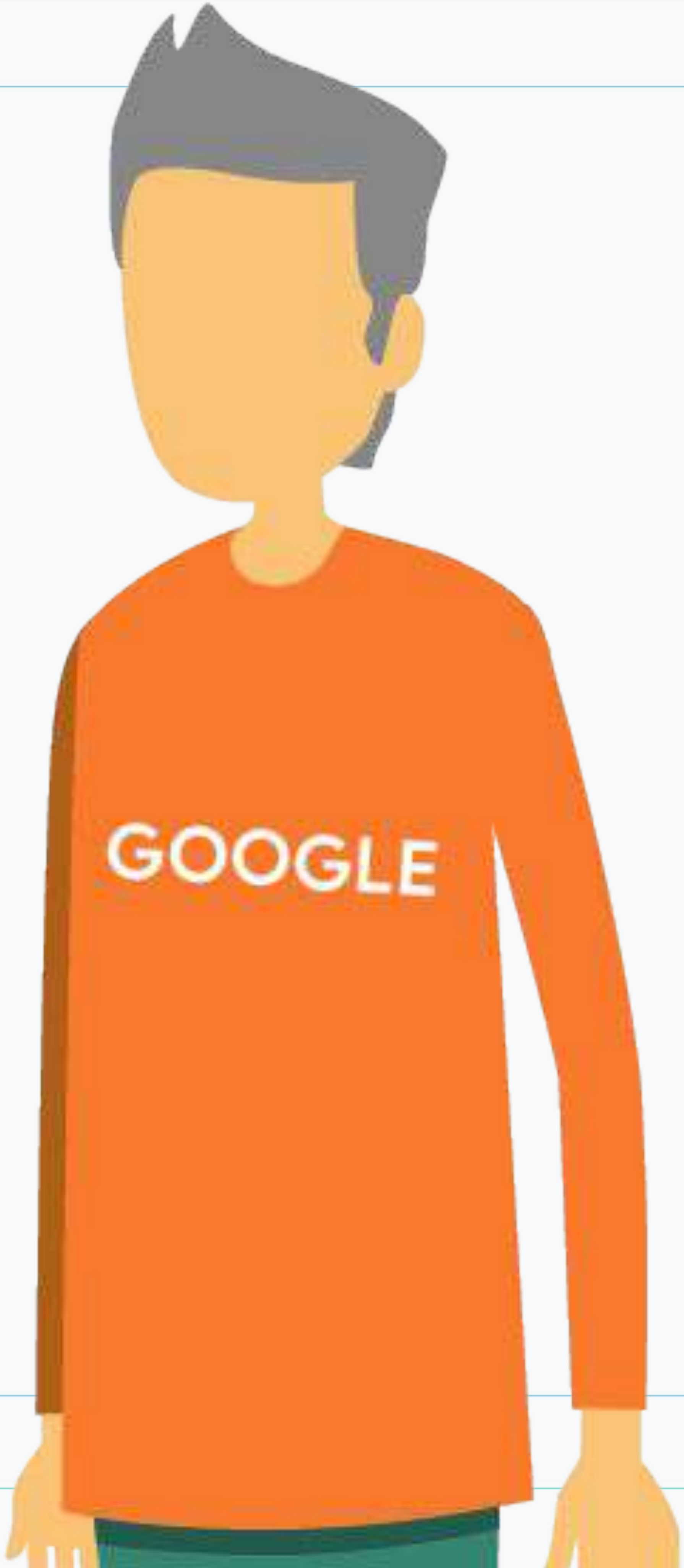


Lab: Build a system that predicts the traffic levels on roads

Is the cardinality of the set of all prediction requests likely to be low, moderate, high, need more info?

Title Safe >

< Action Safe



Lab: Build a system that predicts the traffic levels on roads

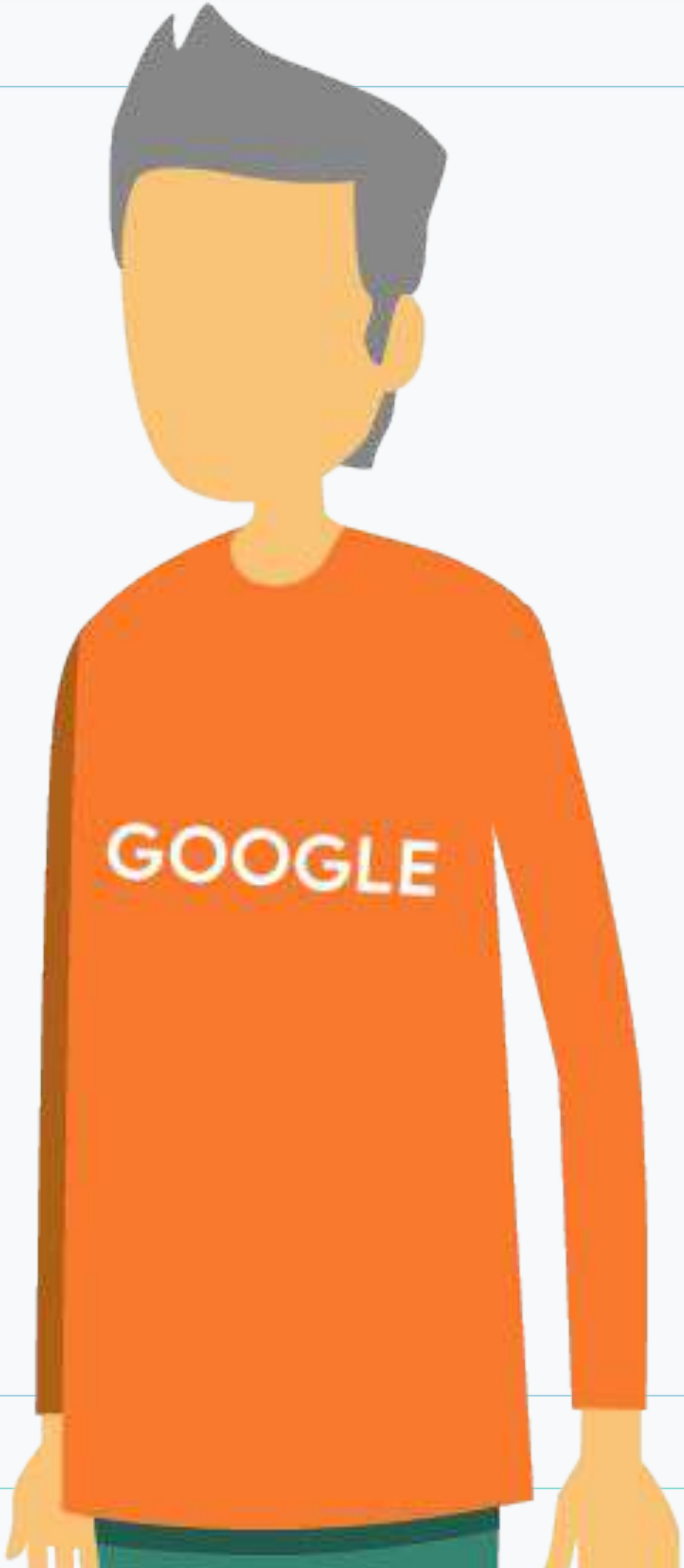
Is the cardinality of the set of all prediction requests likely to be low, moderate, high, need more info?

What does it depend on?

A) Historical traffic data
B) Problem framing
C) Variance of Traffic Levels

Title Safe >

< Action Safe



Lab: Build a system that predicts the traffic levels on roads

Is the cardinality of the set of all prediction requests likely to be low, moderate, high, need more info?

What does it depend on?

A) **<u>Historical traffic data</u>**
B) **<u>Problem framing</u>**
C) Variance of Traffic Levels

Title Safe >

< Action Safe