Google Cloud

Partner Certification Academy

Professional Machine Learning Engineer

We will begin in:

<<15:00->>

The information in this presentation is classified:

Google confidential & proprietary

⚠️ This presentation is shared with you under NDA.

- Do **not** record or take screenshots of this presentation.
- Do **not** share or otherwise distribute the information in this presentation with anyone **inside** or **outside** of your organization.

Thank you!

Source Materials

Some of this program's content has been sourced from the following resources:

- Google Cloud certification site
- Google Cloud documentation
- Google Cloud console
- Google Cloud courses and workshops
- Google Cloud white papers
- Google Cloud Blog
- Google Cloud YouTube channel
- Google Cloud samples
- Google codelabs
- Google Cloud partner-exclusive resources

Google Cloud

Google Cloud Skills Boost for Partners

- Google Cloud Fundamentals: Core Infrastructure
- Logging, Monitoring and Observability in Google Cloud

Google Cloud
Partner Advantage

- Identity Management Technical Deep Dive
- Access Management Technical Deep Dive
- Cloud Foundations: Cost Control Technical Deep Dive [PSO Y22]

Session Logistics

- When you have a question, please:
  - Click the Raise hand button in Google Meet.
  - Or add your question to the Q&A section of Google Meet.
  - Please note that answers may be deferred until the end of the session.

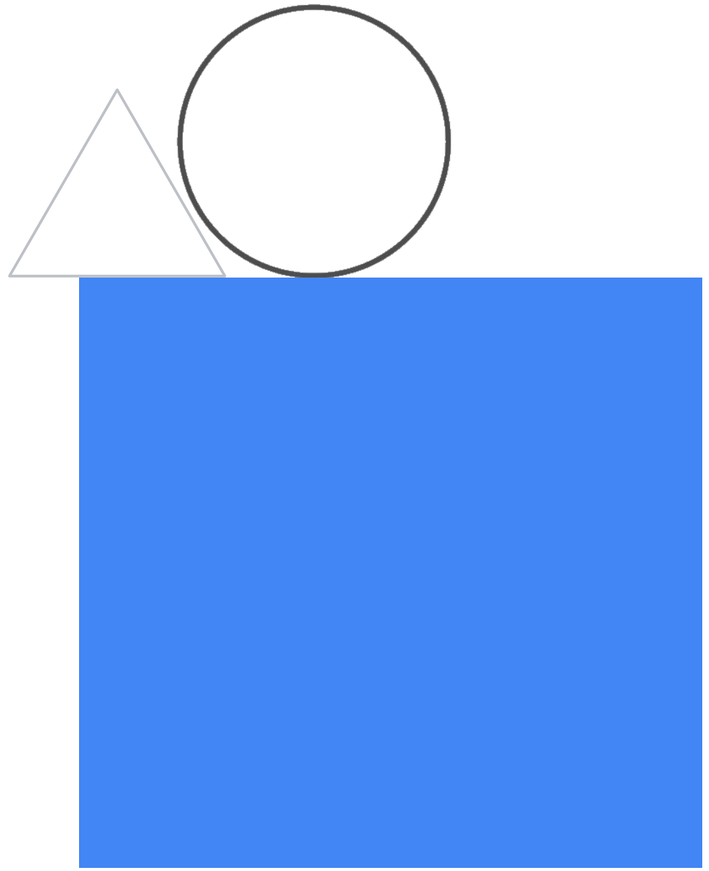
- These slides are available in the Student Lecture section of your Qwiklabs classroom.

- The session is **not recorded.**

- Google Meet does not have persistent chat.
  - If you get disconnected, you will lose the chat history.
  - Please copy any important URLs to a local text file as they appear in the chat.

Google Cloud Partner Learning Programs

- Partner Certification Academy
- Partner Delivery Readiness Index (DRI)
- Cloud Skills Boost for Partners
- Partner Advantage



Learner Commitment

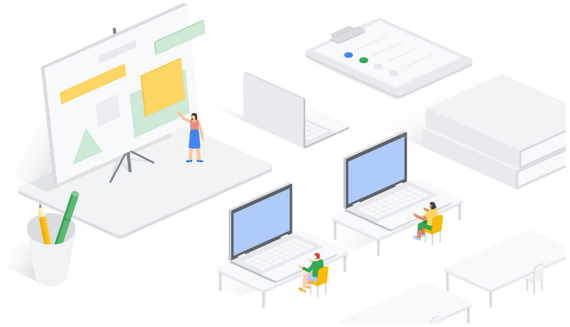
Each week, learners are to complete the learning path's course content, Cloud Skills Boost for Partner Quests/Challenge Labs and material that the mentor has recommended that will support learning.

- **Workshop Day:** Meet for the cohort's weekly 'general session'. **(≅ 2 hours)**
- **During the week:** Complete the week's course, perform hands-on labs, review any additional material suggested material for the week. **(≅ 8 - 16 hours)**

➔ **Important:** Learners must allocate time between each weekly session to study and familiarize themselves with any prerequisite knowledge they may lack. It is also recommended that learners complete the next week's course prior to the scheduled workshop.

Path to Service Excellence

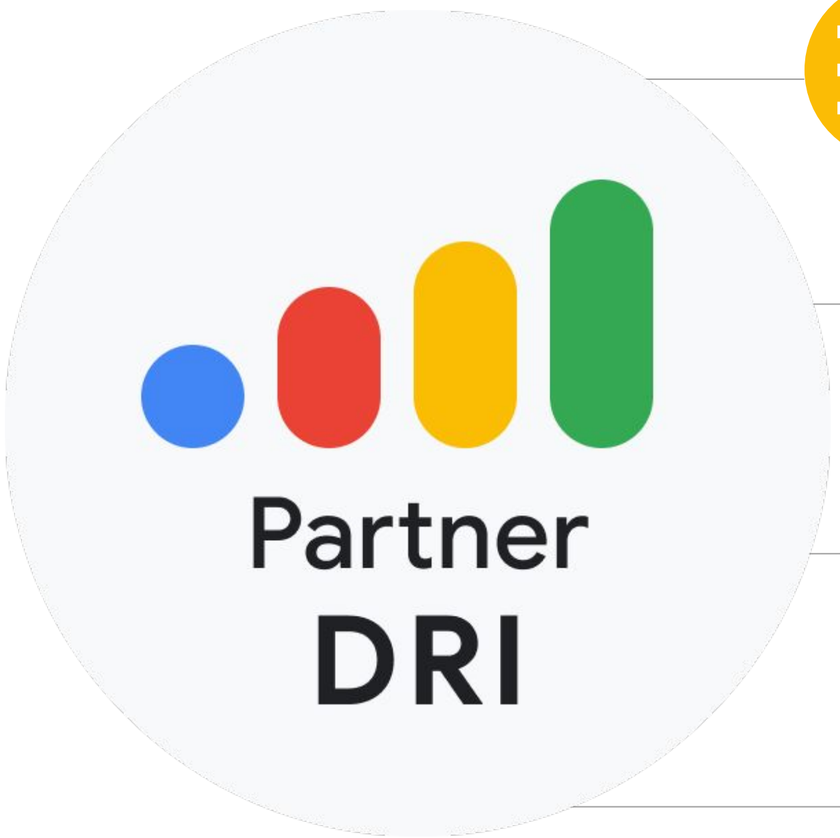
Certification



Advanced Solutions Training

Delivery Readiness Index

Benchmark your skills with DRI



Assess: Partner Proficiency and Delivery Capability
Benchmark Partner individuals, project teams and practices GCP capabilities

Analyze: Individual Partner Consultants' GCP Readiness
Showcase Partner individuals GCP knowledge, skills, and experience

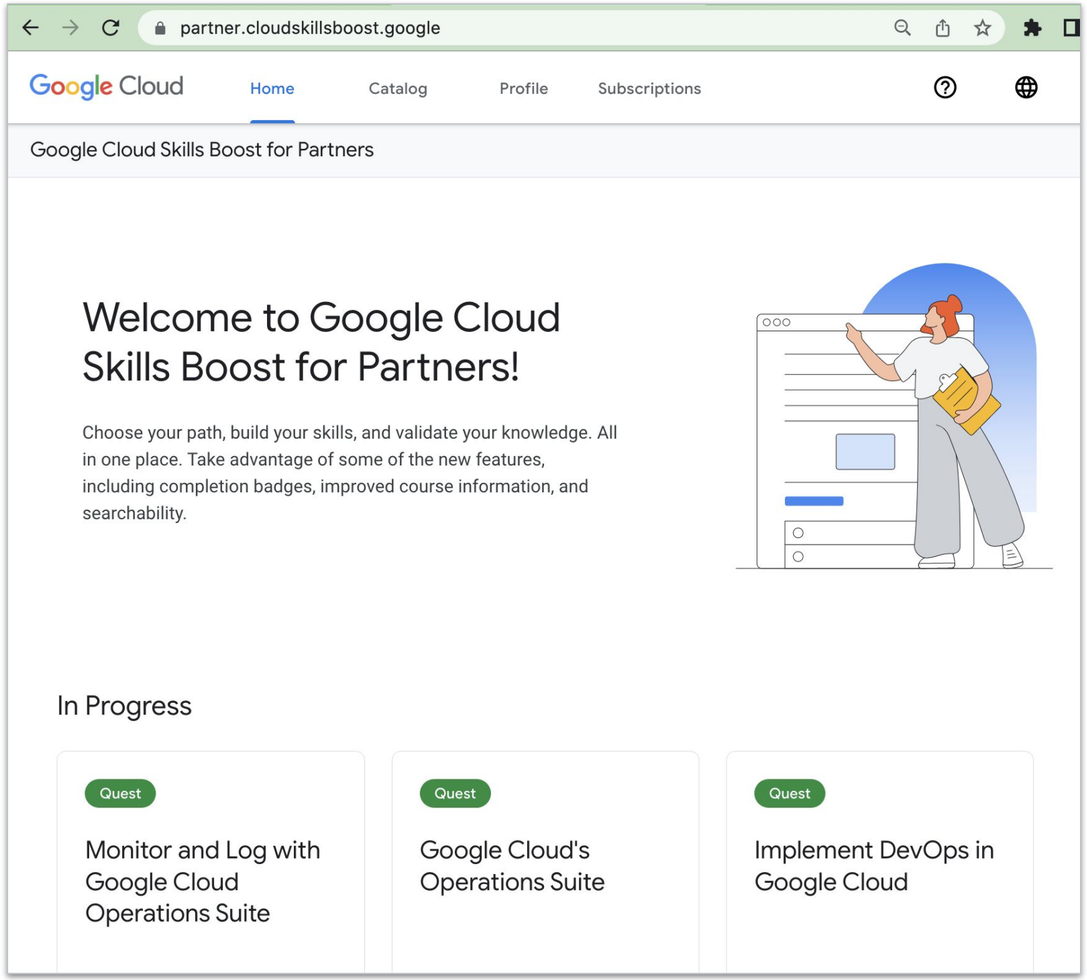
Advise: Google Assurance for Partner Delivery
Packaged offerings to bridge specific capability gaps

Action: Tailored L&D Plan for Account Based Enablement
Personalized learning & development recommendations per individual consultant

Google Cloud Skills Boost for Partners

https://partner.cloudskillsboost.google/

- On-demand course content
- Hands-on labs
- Skill Badges
- **FREE** to Google Cloud Partners!



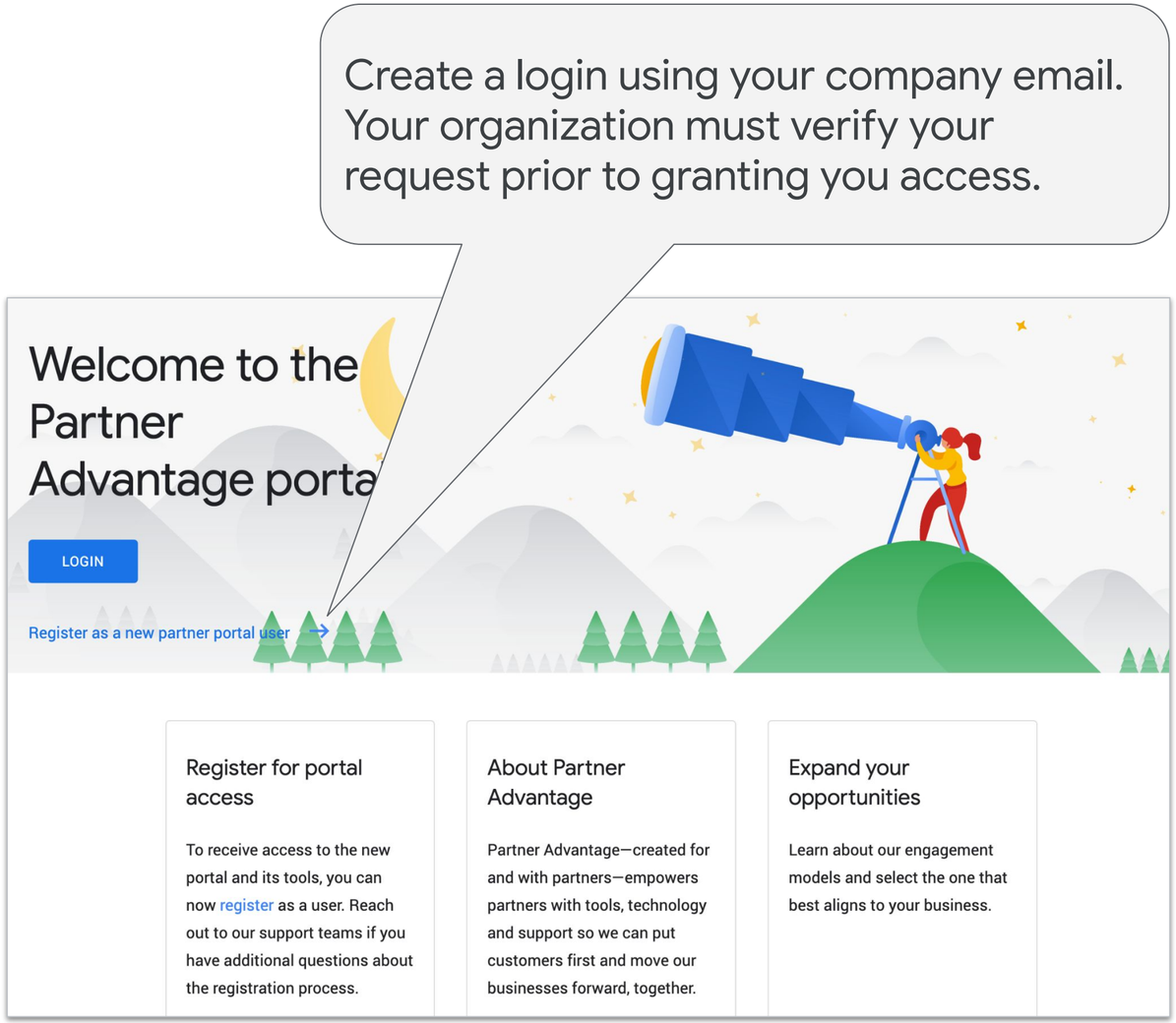
Google Cloud

Partner Advantage

- Resources for Google Cloud partner organizations:
  - Recent announcements
  - Solutions/role-based training
  - Live/pre-recorded webinars on various topics
    - Partner Advantage Live Webinars

- Complements the certification self-study material presented on Google Cloud Skills Boost for Partners

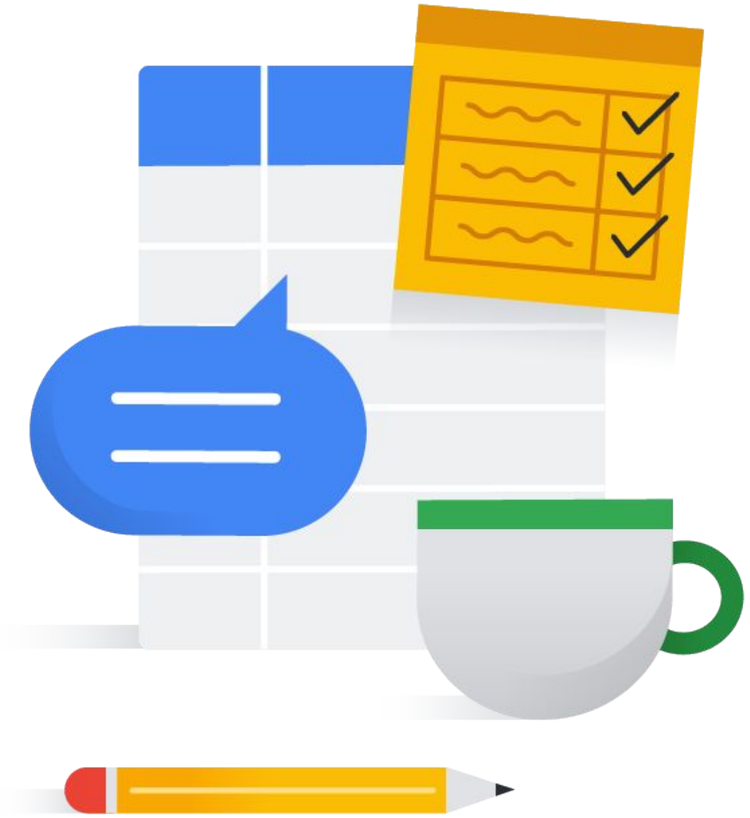
- Helpful Links:
  - Getting started on Partner Advantage
  - Join Partner Advantage
  - Get help accessing Partner Advantage



https://www.partneradvantage.goog

Program issues or concerns?

- Problems with **accessing** Cloud Skills Boost for Partners
  - partner-training@google.com
- Problems with **a lab** (locked out, etc.)
  - support@qwiklabs.com
- Problems with accessing Partner Advantage
  - https://support.google.com/googlecloud/topic/9198654



Session 4:

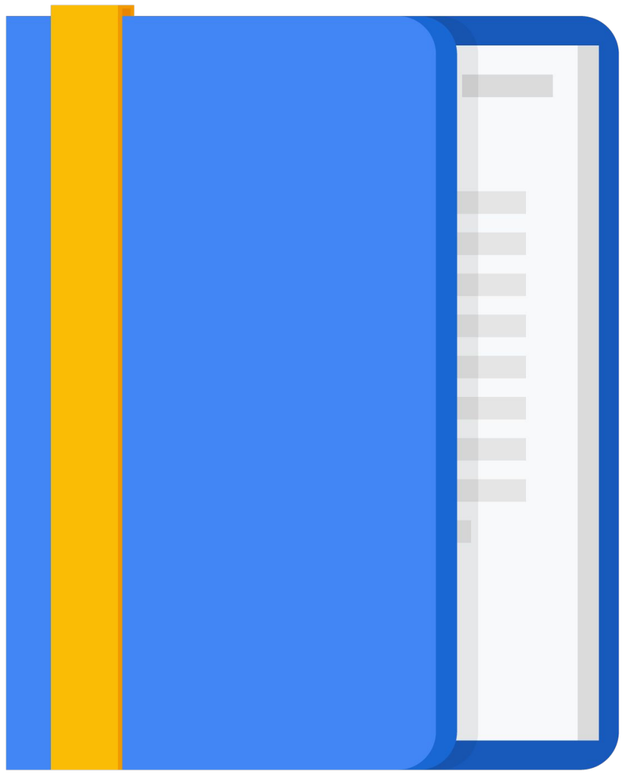
MLOps& Tensorflow on Google Cloud

Agenda

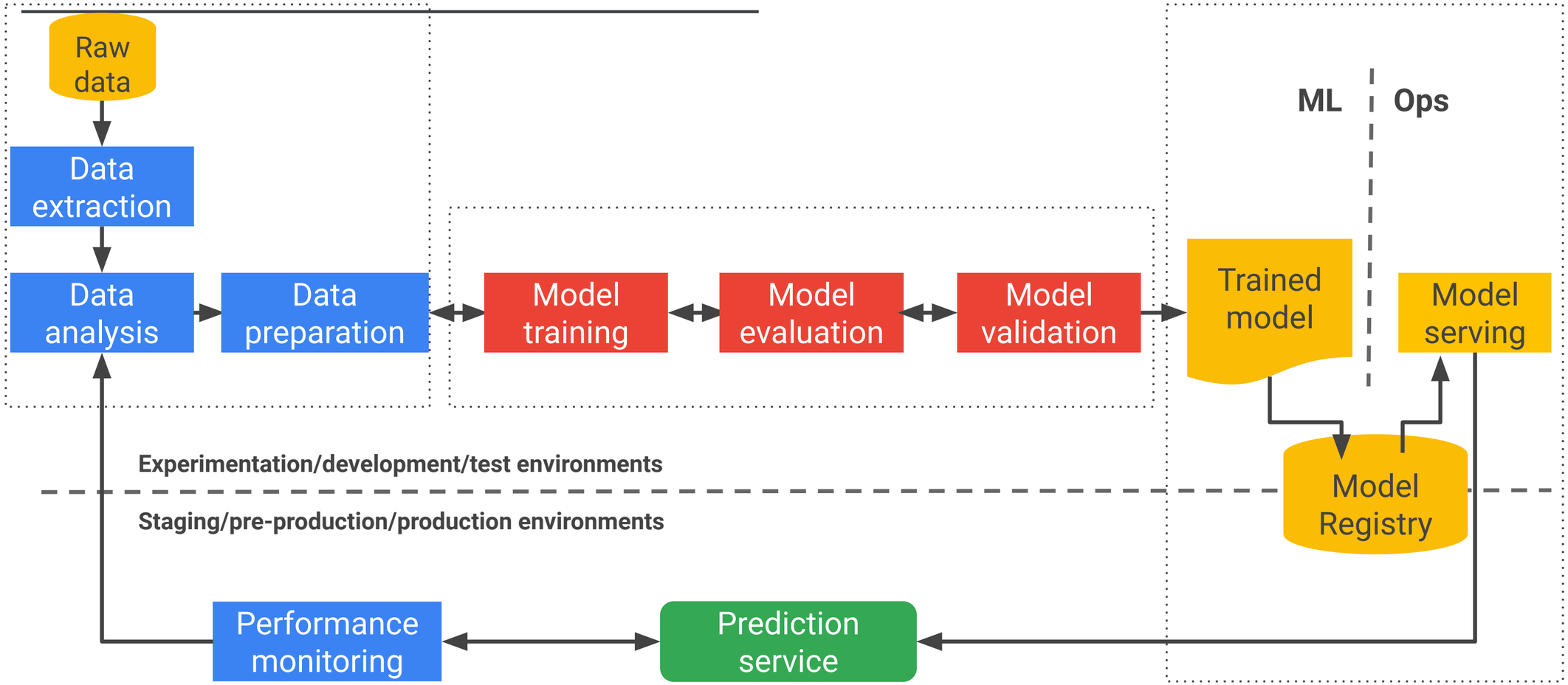
MLOps Introduction

MLOps with Kubeflow

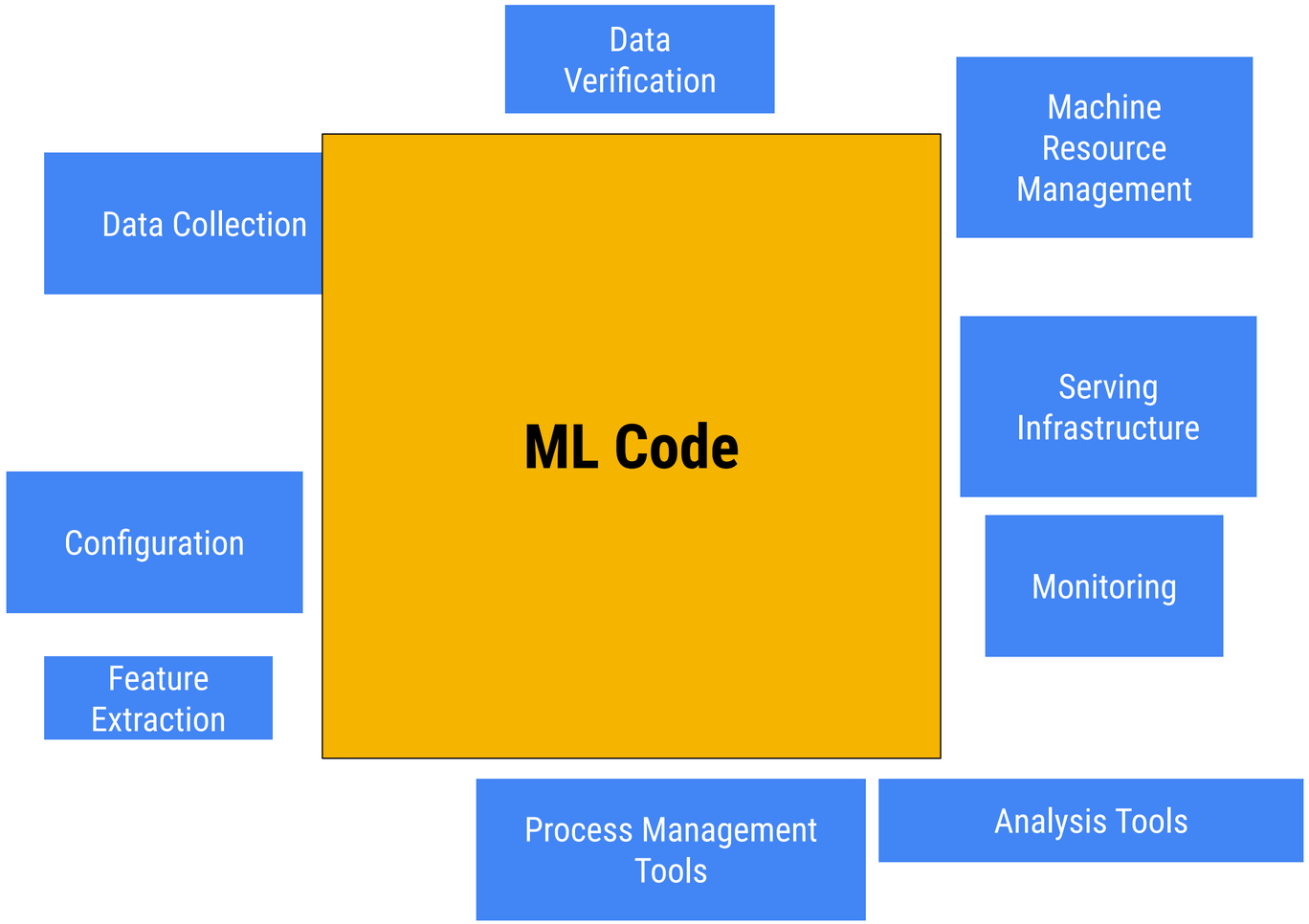
MLOps with TFX



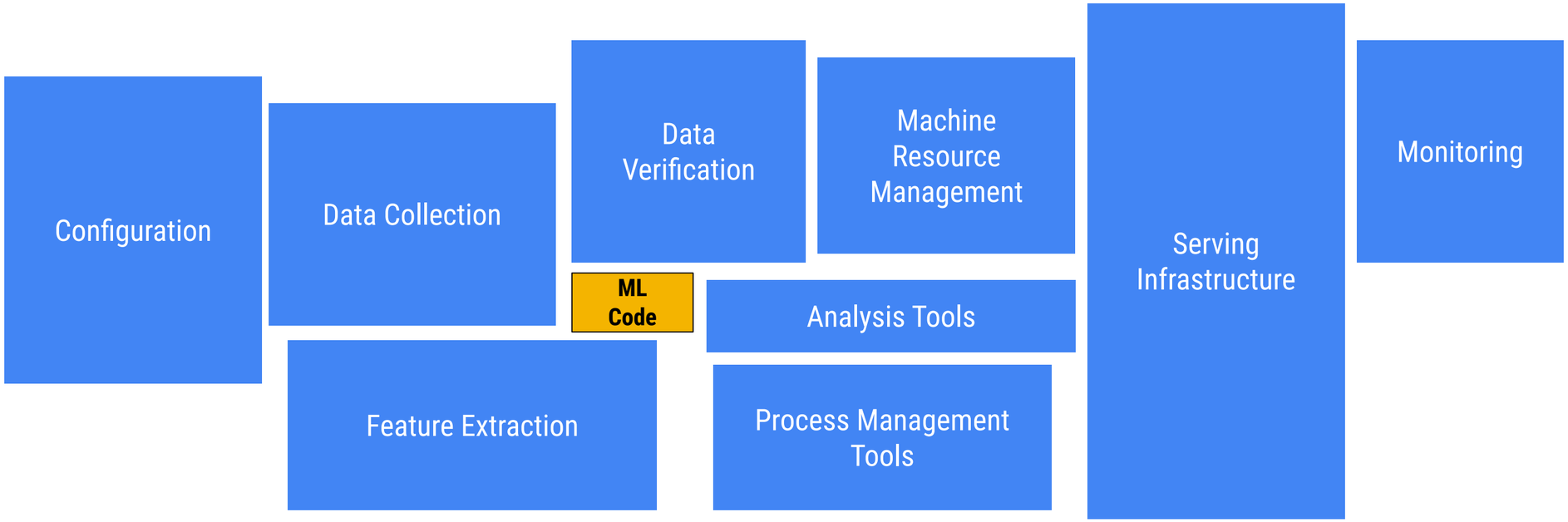
An ML Pipeline Recap



Perception: ML products are mostly about ML

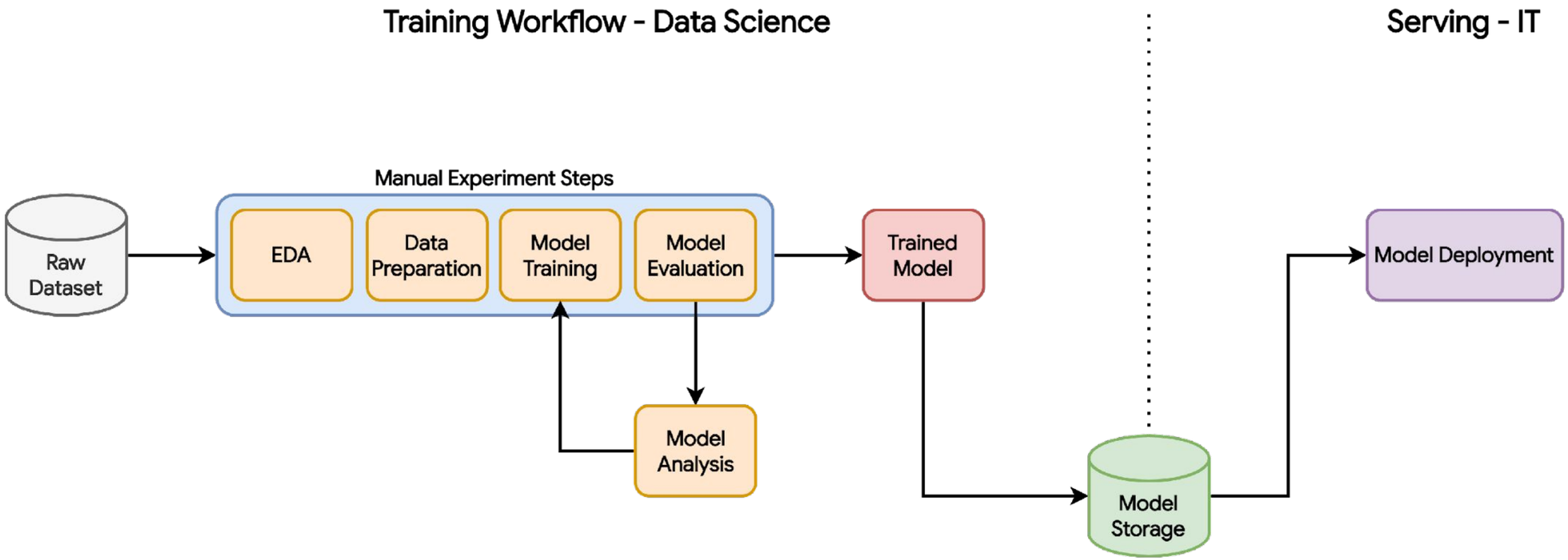


Reality: ML Requires lots of DevOps

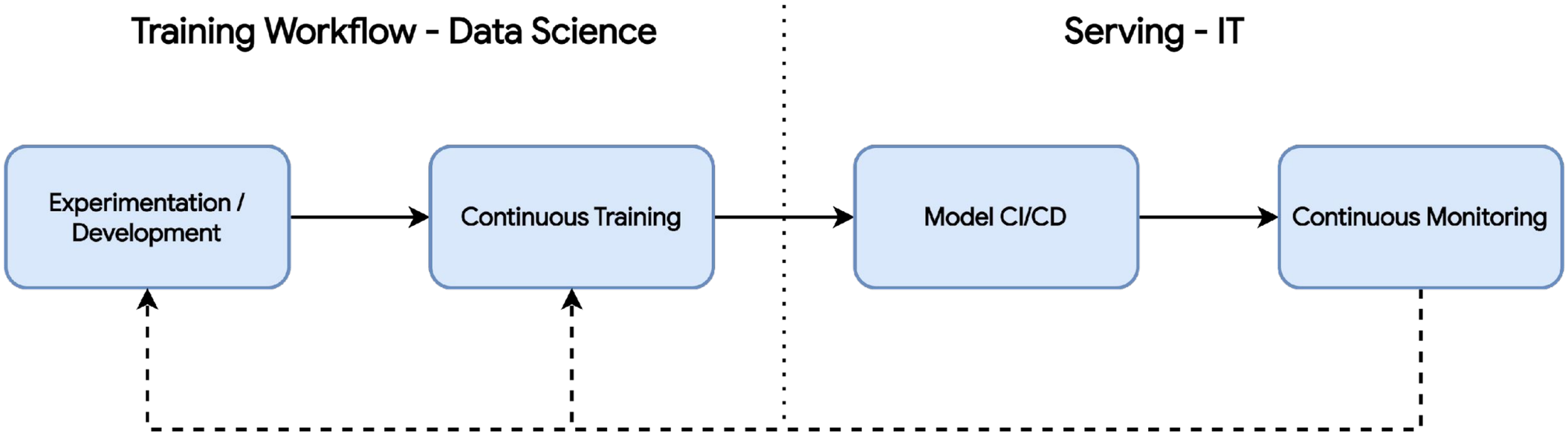


Source: Sculley et al.: Hidden Technical Debt in Machine Learning Systems

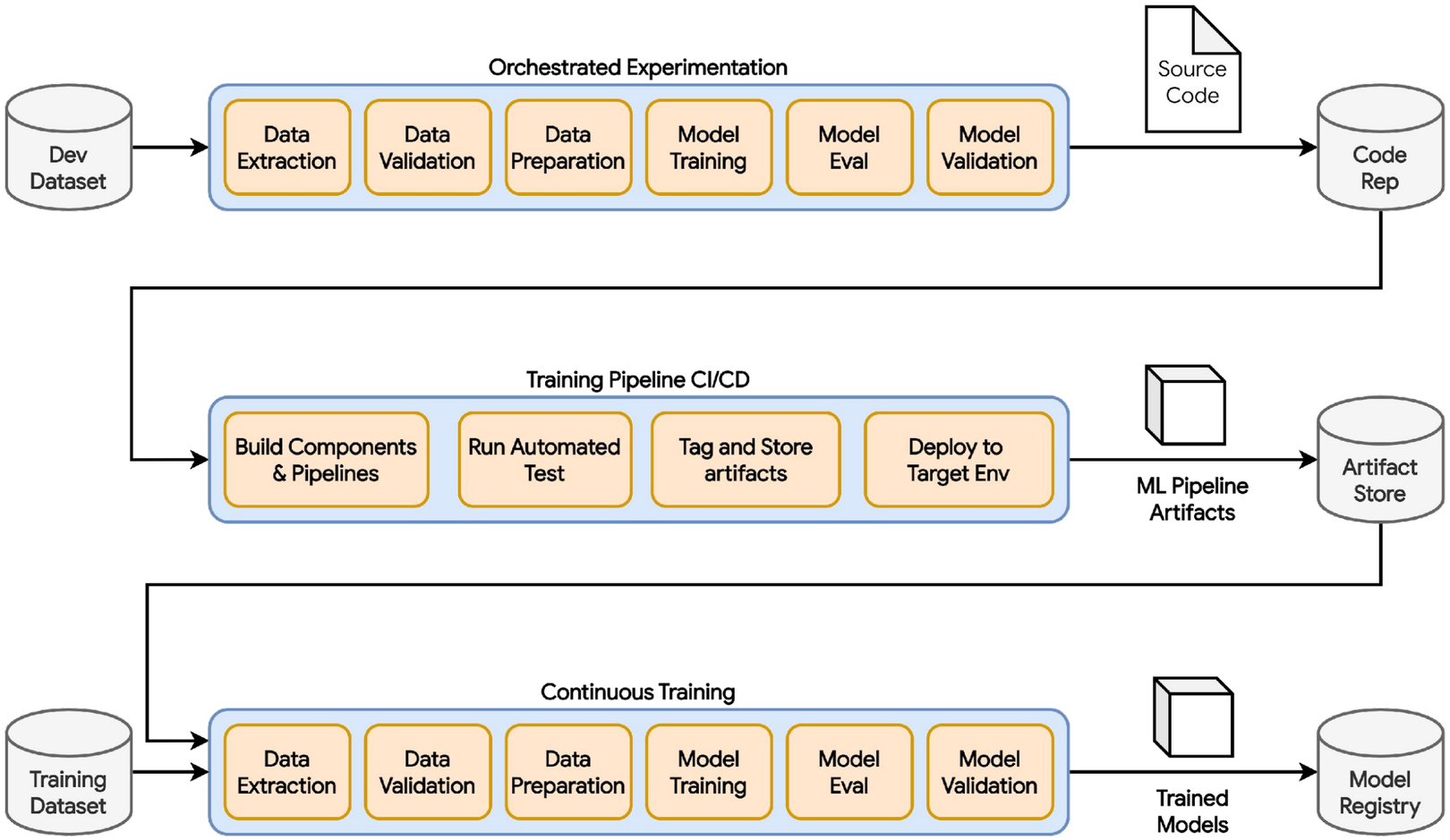
Traditional ML Workflow



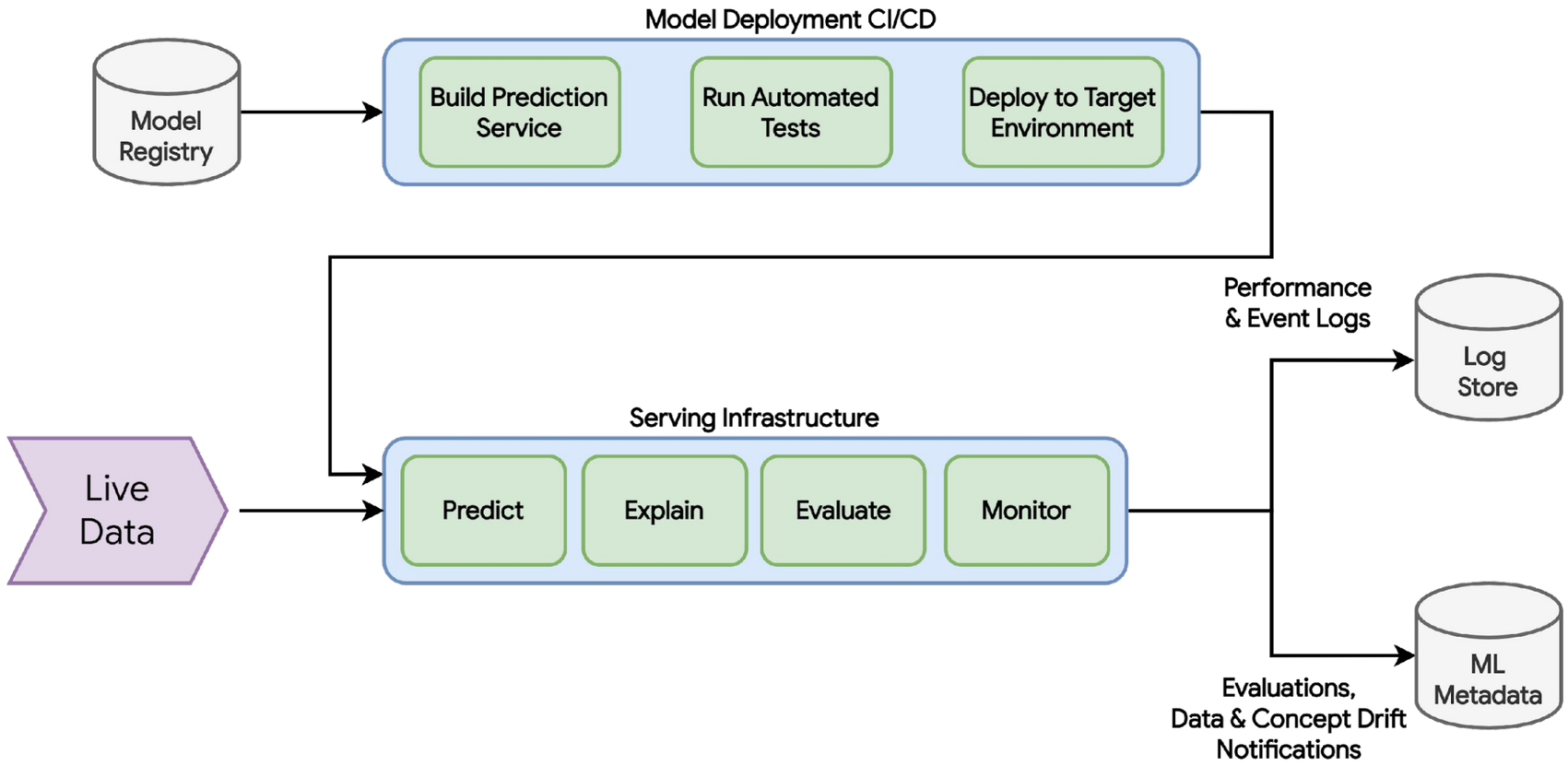
MLOps Workflow



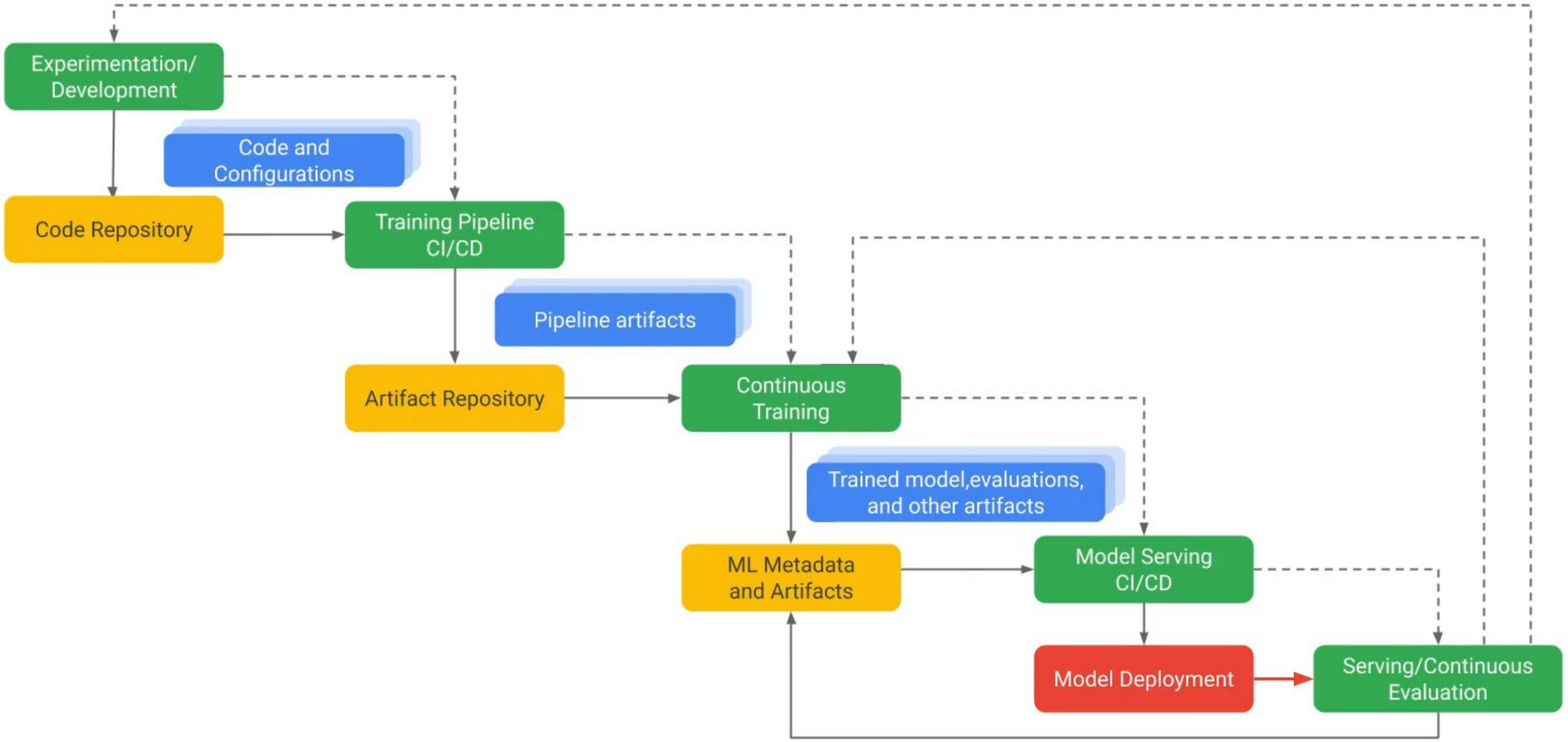
Model Development in MLOps



Model Serving in MLOps



End-to-End view



A COMPARISON OF KUBEFLOW & TFX

Google Cloud

Main Differences:

- **Multi-framework support:** If you need to use multiple machine learning frameworks (e.g., TensorFlow, PyTorch, Scikit-learn) in your workflow, Kubeflow Pipelines can be a good fit.
- **Already running kubeflow:** Kubernetes-native: If your organization is already using Kubernetes extensively, adopting Kubeflow Pipelines can lead to easier integration and a more consistent experience across your infrastructure.
- **Open Sourced:** Extensible to other platforms that can run Kubernetes

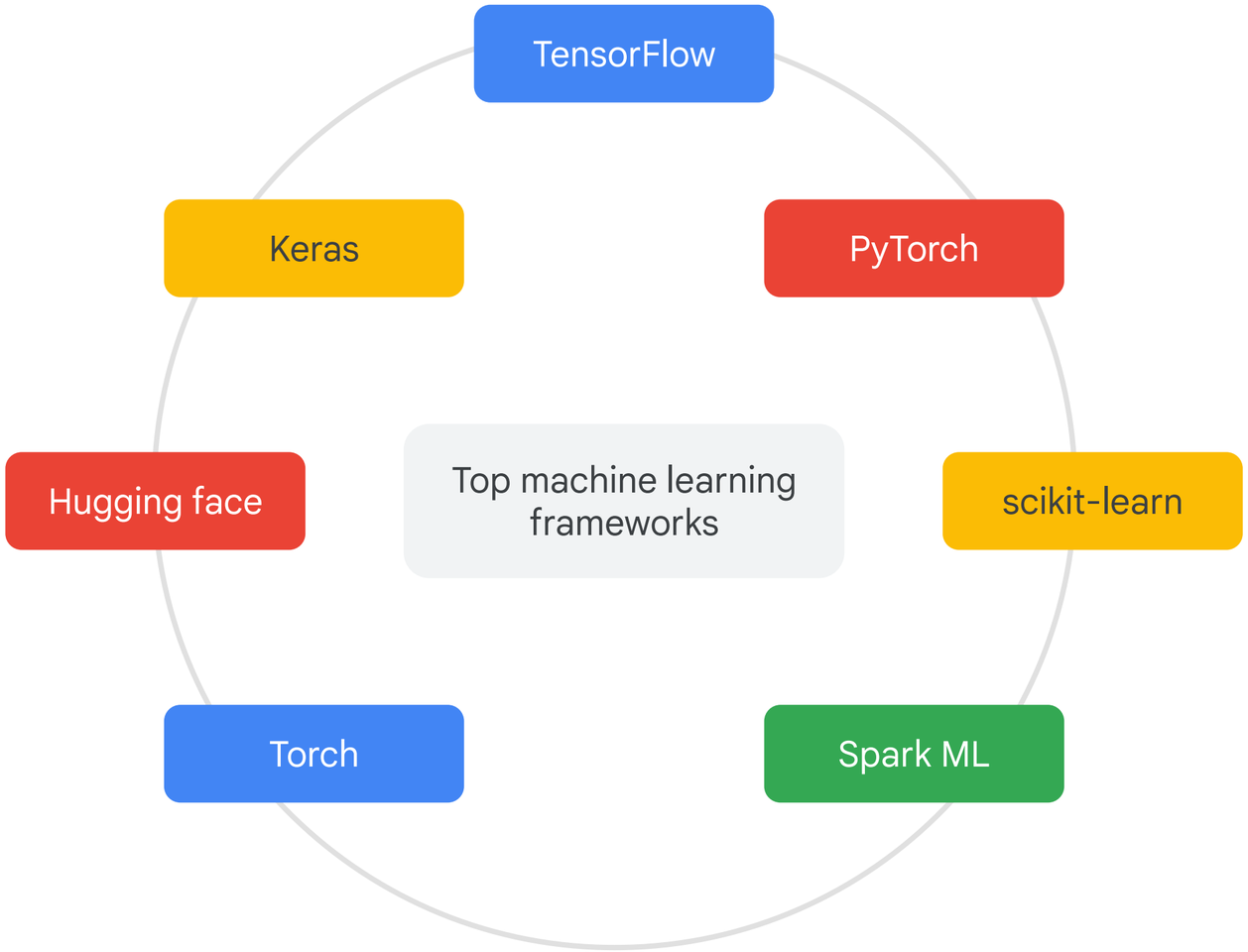
- **TensorFlow-centric:** If your project relies primarily on TensorFlow for model training and serving, TFX Pipelines can provide a more cohesive and streamlined experience.
- **Pre-built components:** TFX offers a set of standard components (e.g., ExampleGen, Transform, Trainer, Evaluator) that follow best practices and can save time in developing your ML workflows.
- **Advanced features:** TFX Pipelines provide advanced features such as model analysis, validation, and drift detection, which can help ensure your models are robust and maintain high performance over time.

example business use cases:

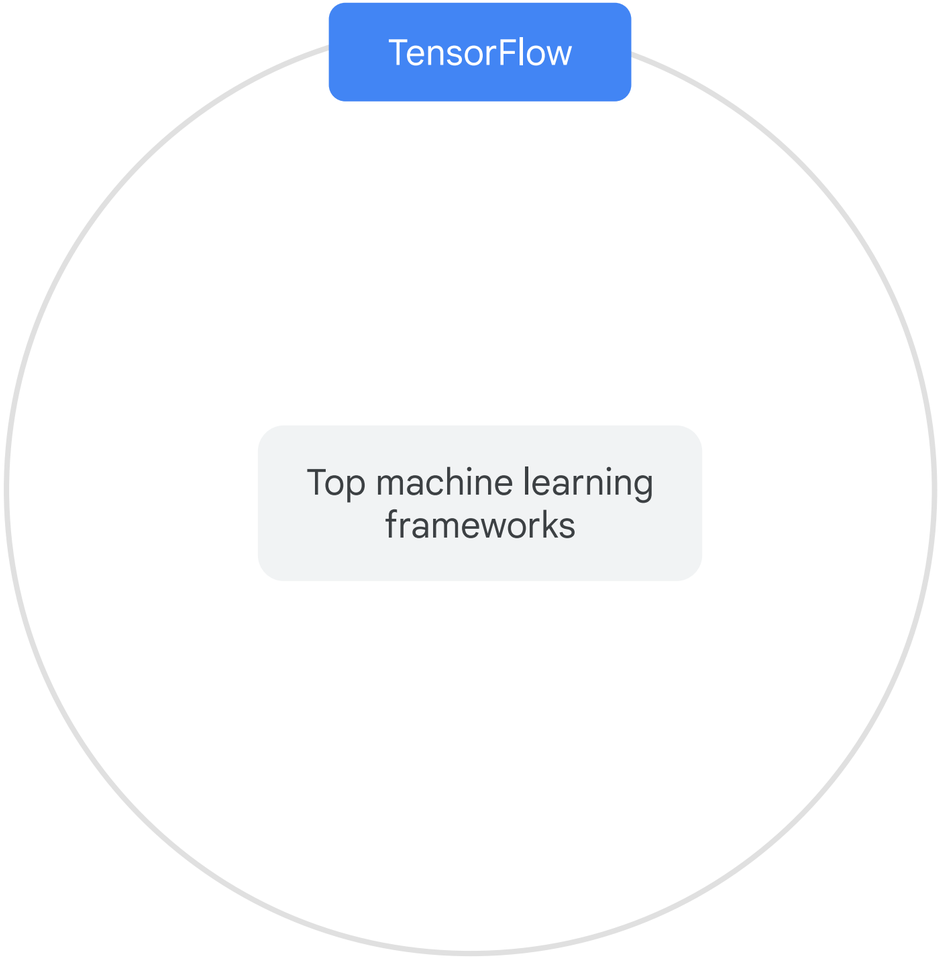
- **Fraud Detection:** A financial institution wants to build a fraud detection system that uses multiple ML frameworks to analyze transactions.
- **Predictive Maintenance:** A manufacturing company wants to predict when their machines might require maintenance based on sensor data.
- **Customer Churn Prediction:** A telecommunications company wants to predict which customers are likely to churn, using multiple ML frameworks.

- **Personalized Recommendations:** An e-commerce company wants to provide personalized product recommendations to their customers using TensorFlow.
- **Sentiment Analysis:** A social media company wants to analyze user-generated content to determine sentiment
- **Image Classification:** A social media company wants to analyze user-generated content to determine sentiment

Kubeflow




TensorFlow

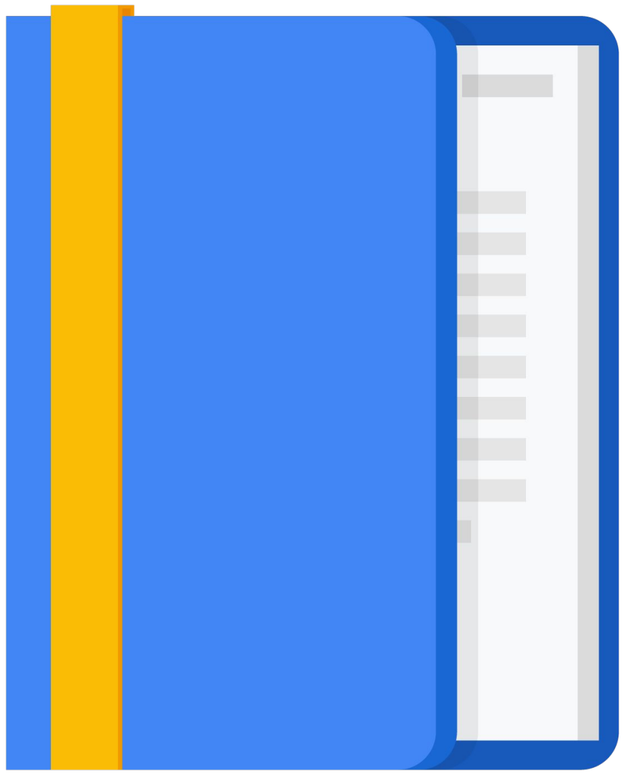

TensorFlow
Top machine learning frameworks

Agenda

MLOps Introduction

MLOps with Kubeflow

MLOps with TFX



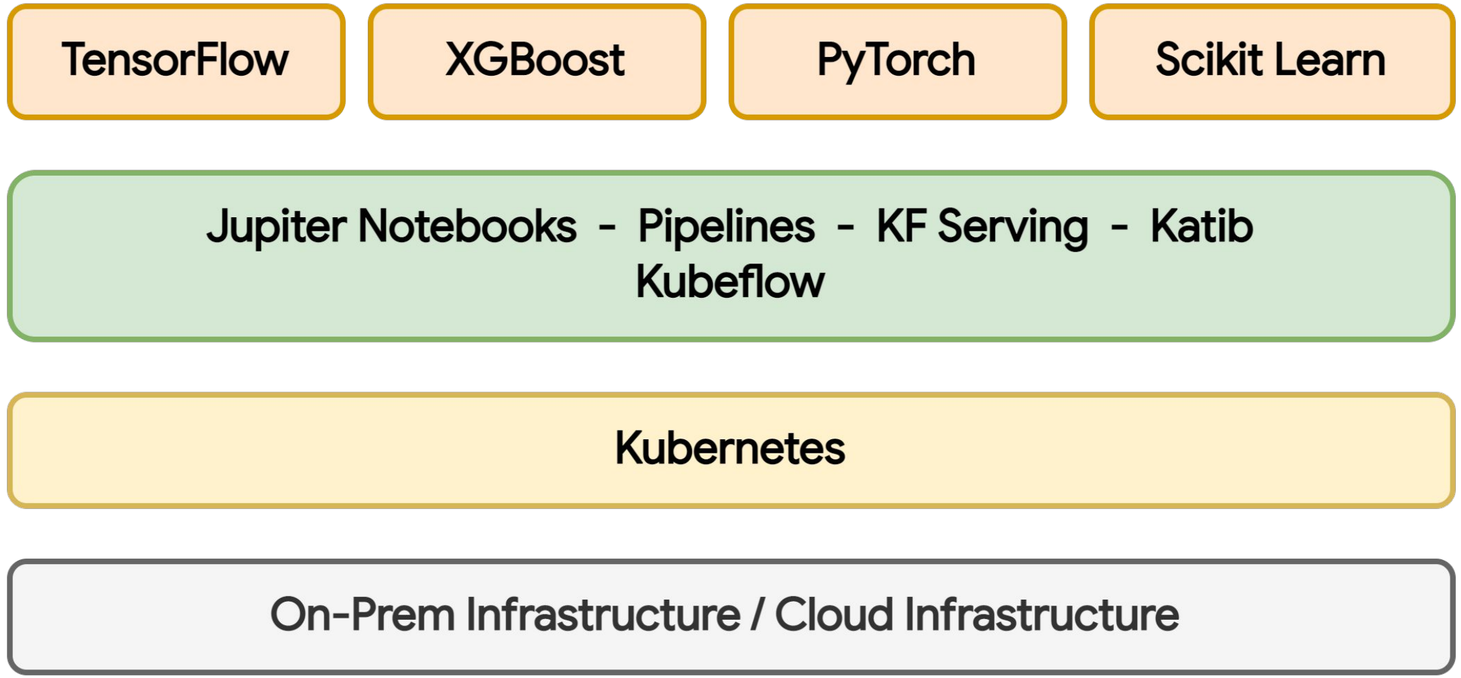
Kubeflow provides a platform for building ML products

- Leverage containers and Kubernetes to solve the challenges of building ML products
- Kubeflow = Cloud Native, multi-cloud solution for ML.
- Kubeflow provides a platform for composable, portable and scalable ML pipelines
- If you have a Kubernetes conformant cluster, you can run Kubeflow

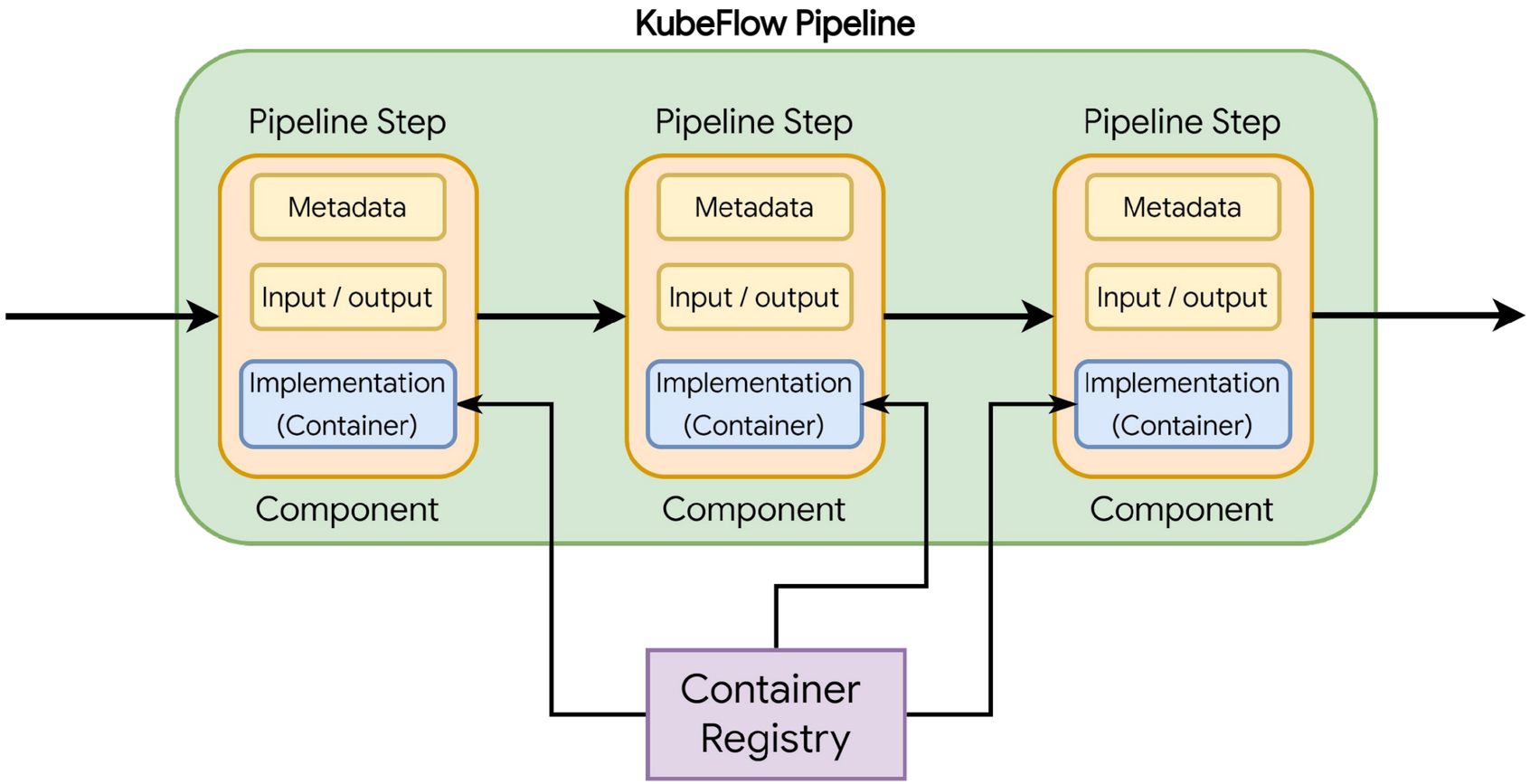
Kubernetes is a great platform for ML

- Containers
- Scaling built in
- Unified architecture
- Easy to integrate building blocks
  - ML APIs
  - Dataflow
- Lots of options for CI/CD
- Portability
  - Dev, On-Prem, Multi-cloud: same stack

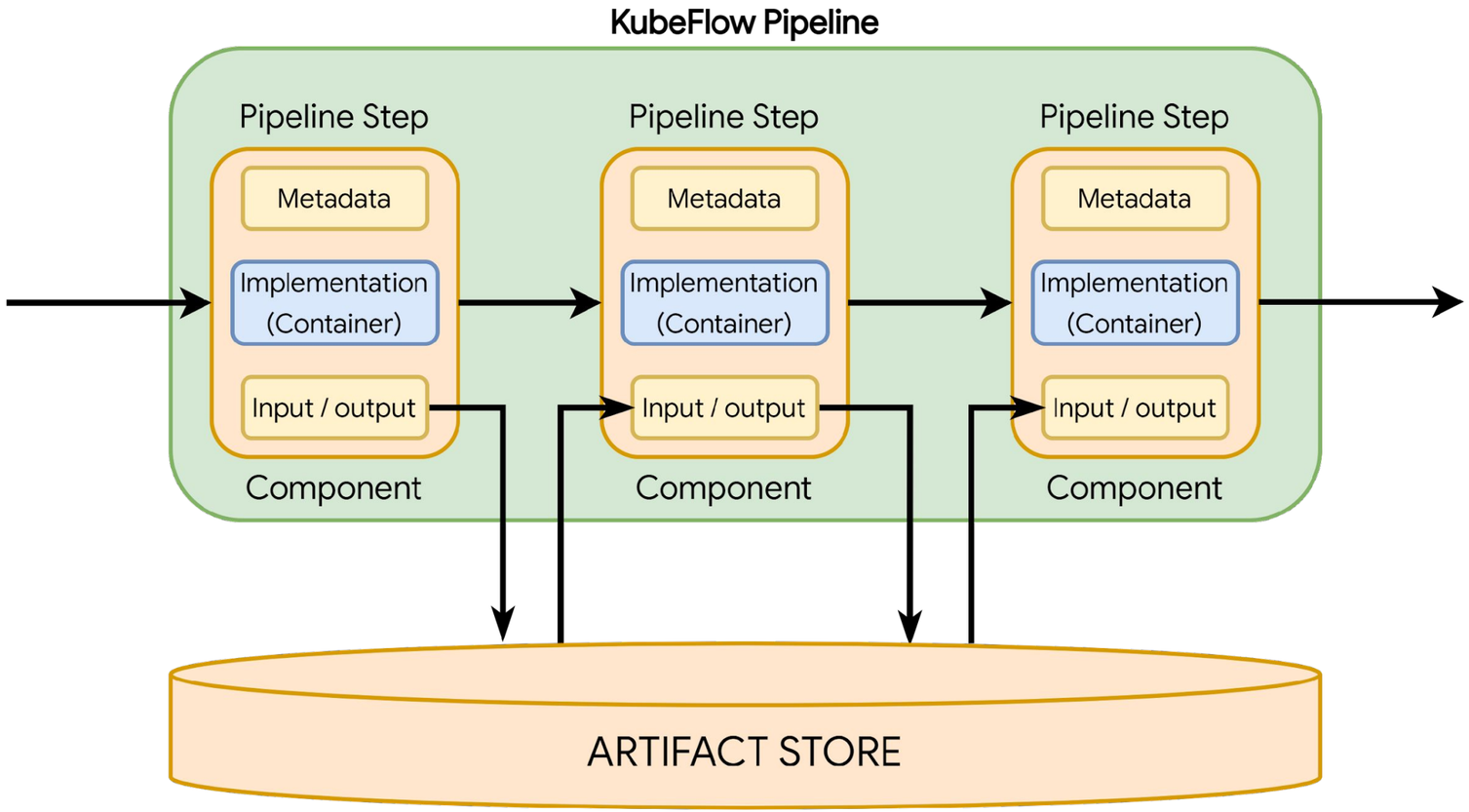
MLOps with Kubeflow



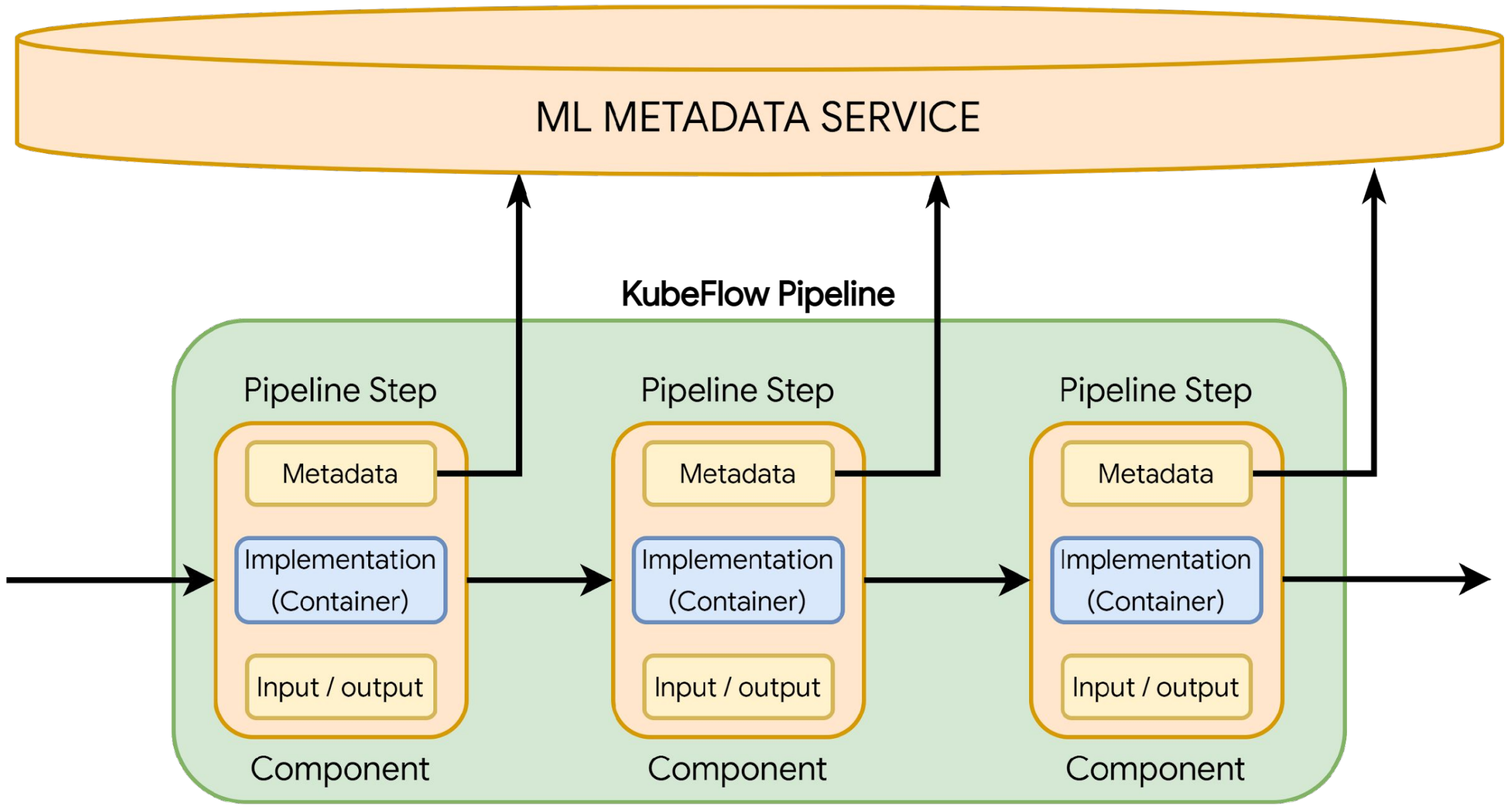
Kubeflow Pipeline



Kubeflow Pipeline Storage



Kubeflow Pipeline Storage

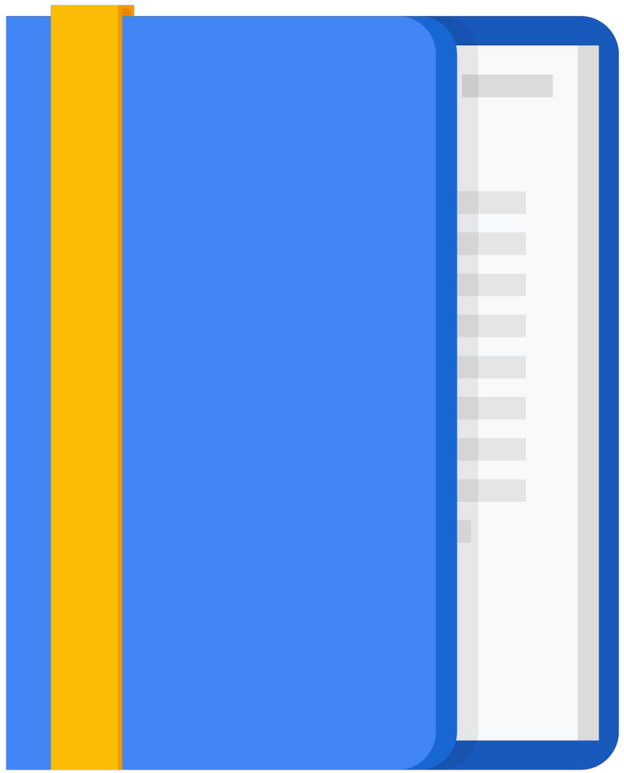


Agenda

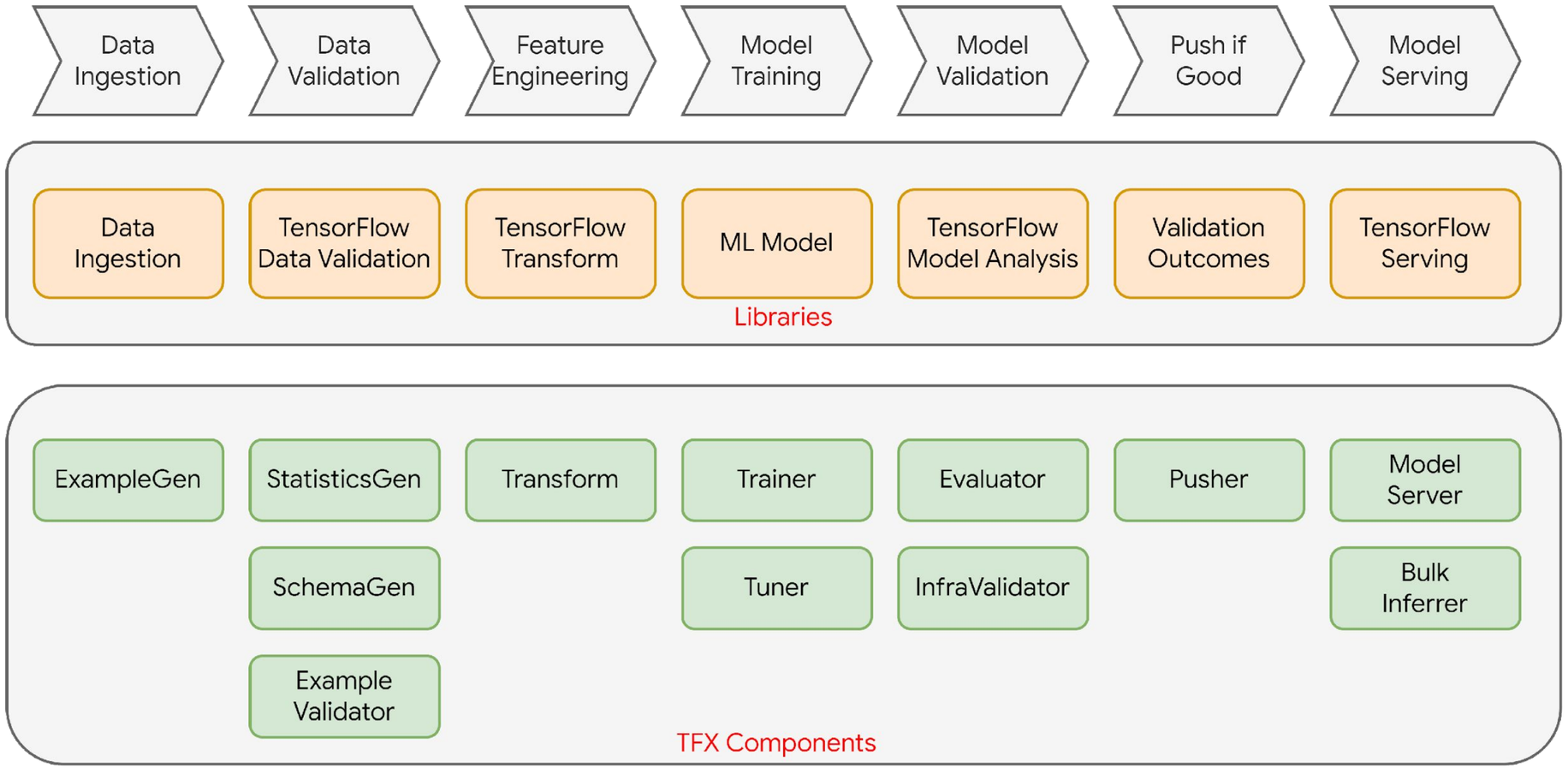
MLOps Introduction

MLOps with Kubeflow

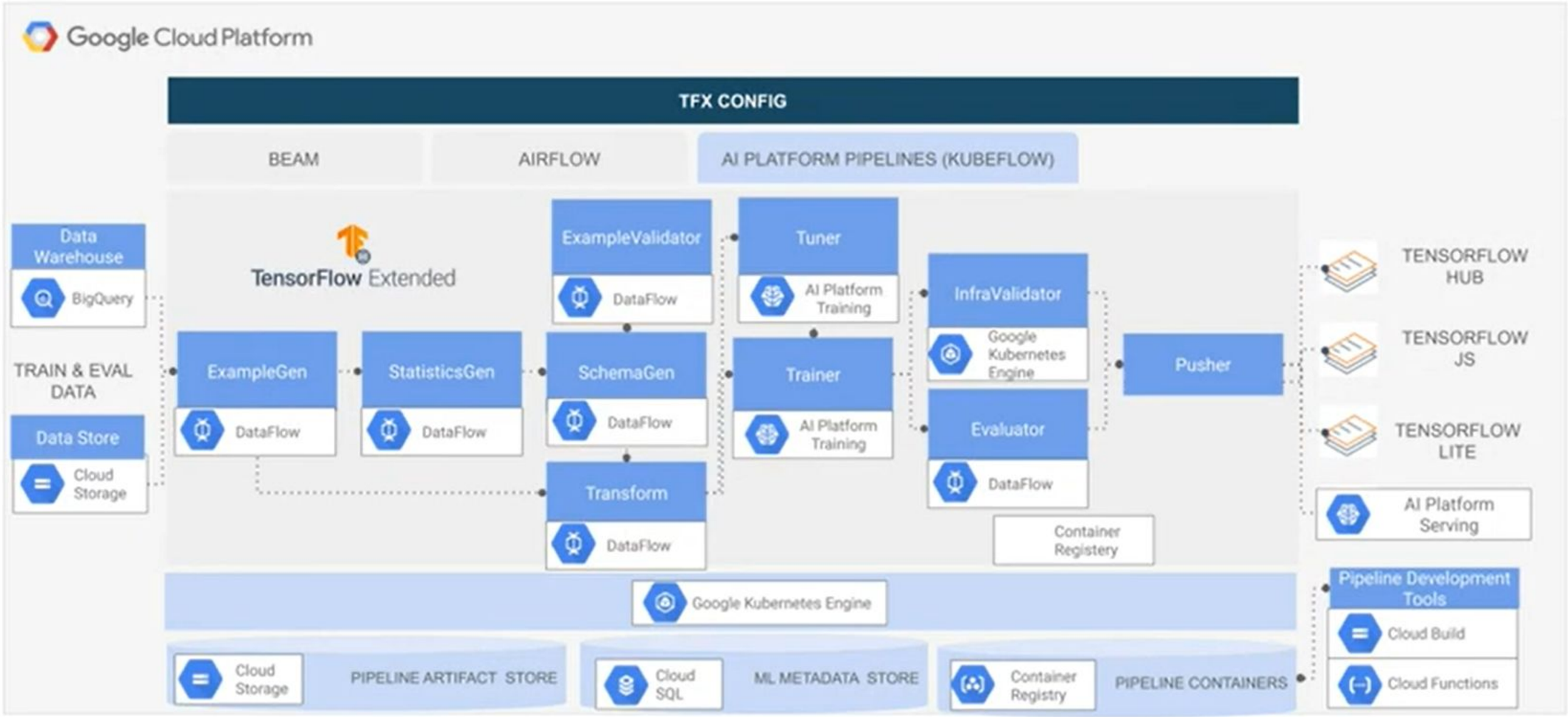
MLOps with TFX



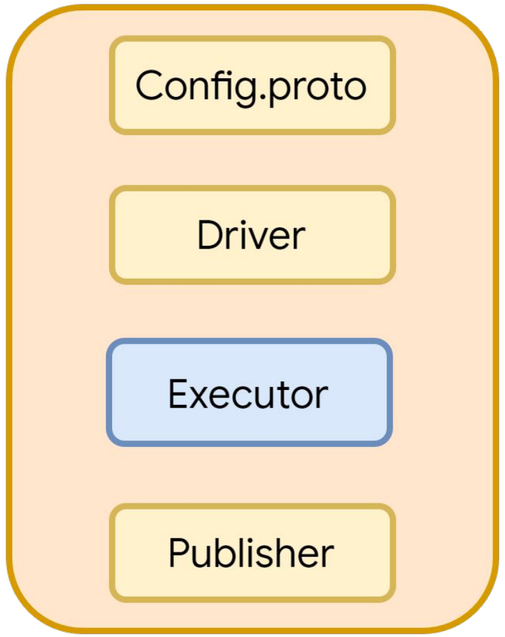
TFX Components



TFX: Google's production machine learning platform



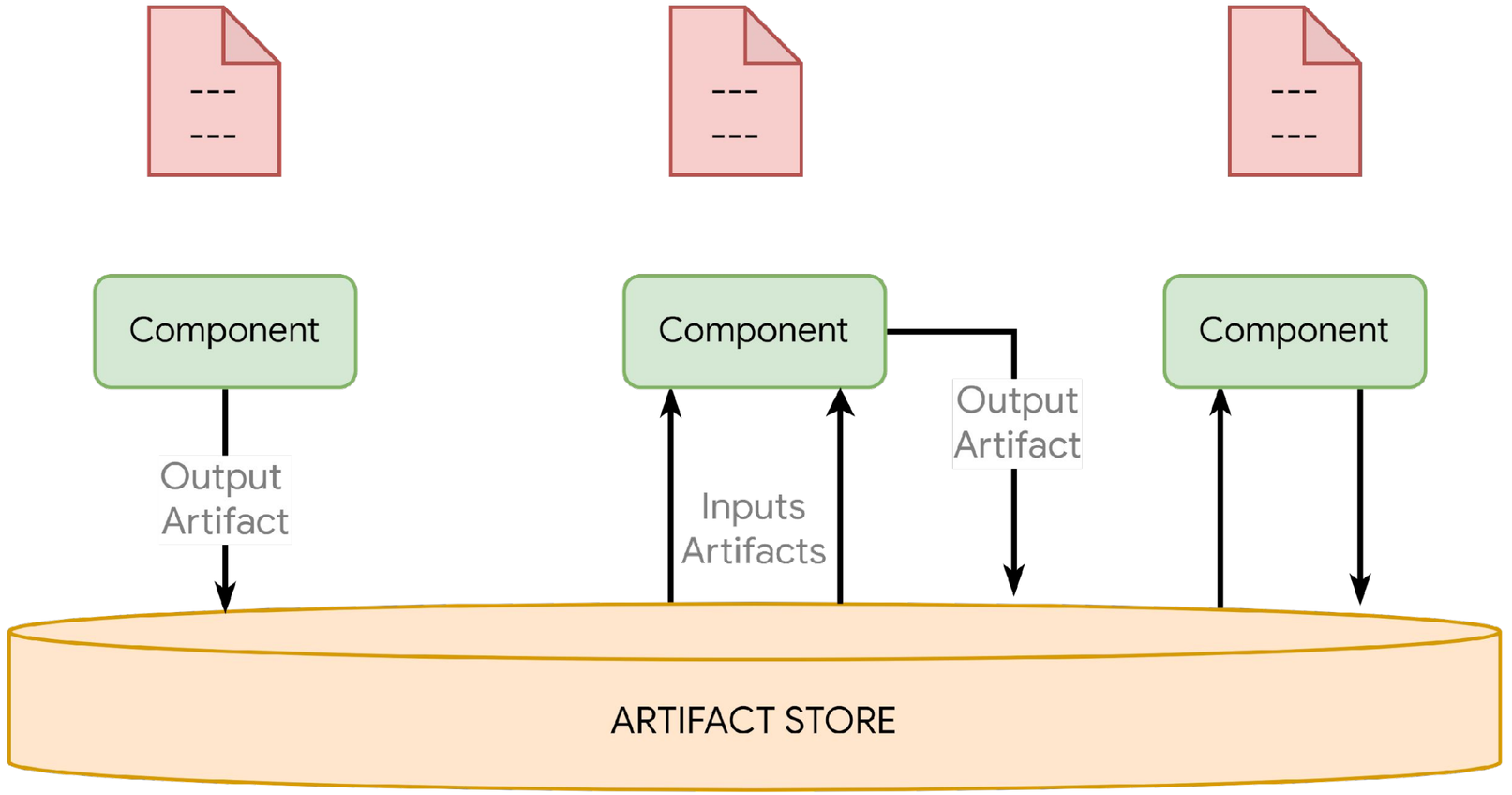
TFX Component



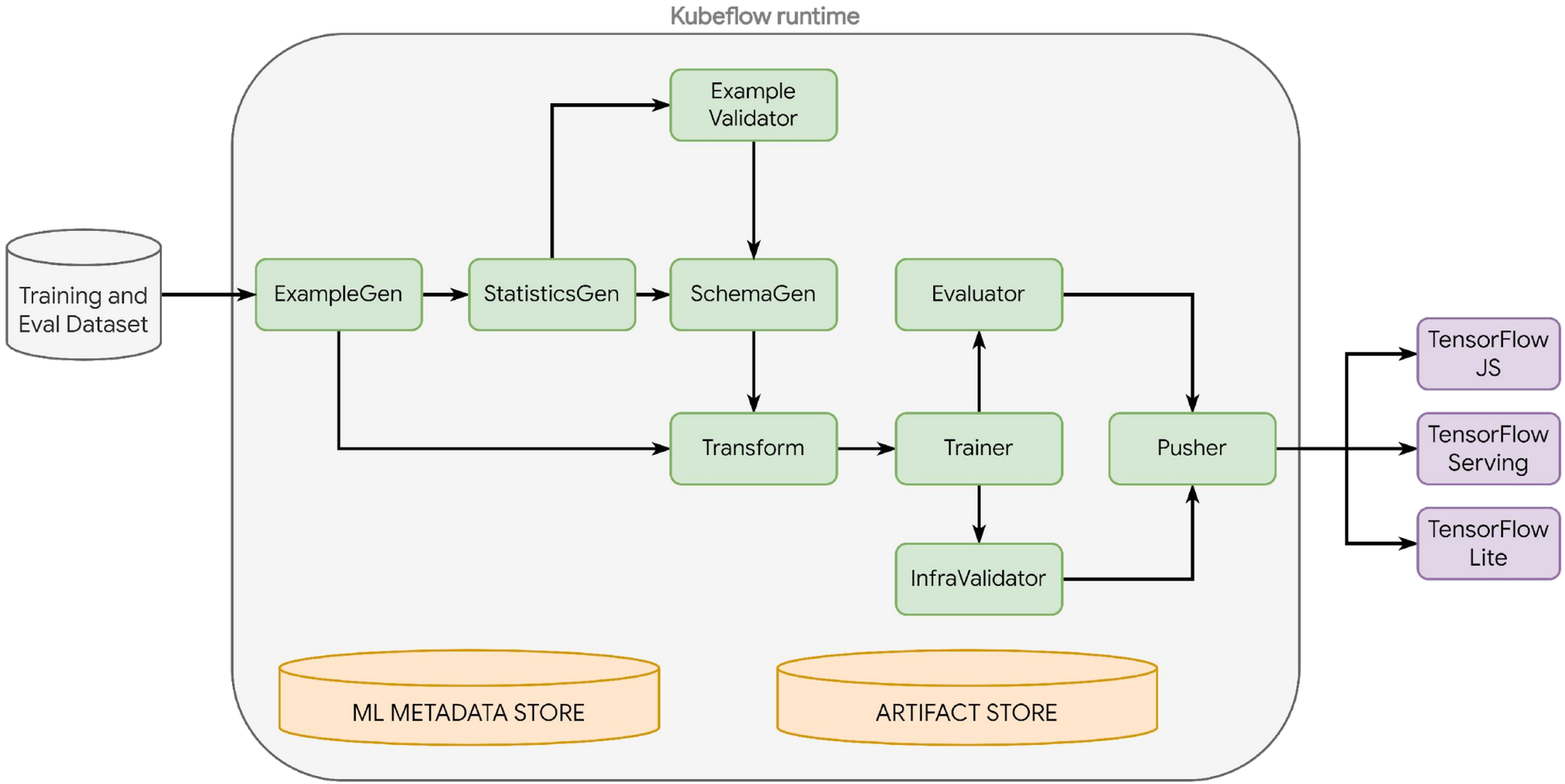
TFX Component

- **Config**: Specifies how components communicates between each other via input and output artifact channels. It also includes component parameters such as hyperparameters.
- **Driver**: gets the artifact locations from the ML metadata store and retrieves artifact from pipeline storage.
- **Executor**: “the workhorse”: implements the actual code of the component such as ingestion/transform/train
- **Publisher**: Updates the ML Metadata store

TFX Component Communication

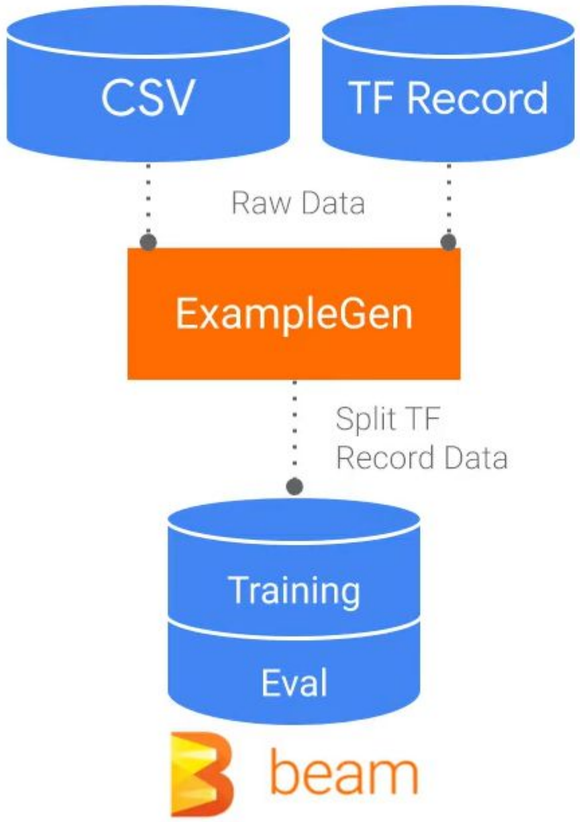


TFX DAG


Kubeflow runtime
Example Validator
Training and Eval Dataset
ExampleGen
StatisticsGen
SchemaGen
Evaluator
TensorFlow JS
Transform
Trainer
Pusher
TensorFlow Serving
TensorFlow Lite
InfraValidator
ML METADATA STORE
ARTIFACT STORE

Component: **ExampleGen**

Inputs and Outputs



Configuration

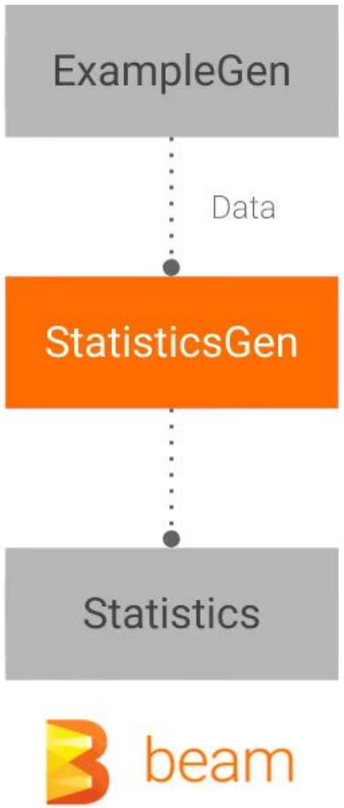
```
output_config = example_gen_pb2.Output(
    split_config=example_gen_pb2.SplitConfig(splits=[
        example_gen_pb2.SplitConfig.Split(name='train', hash_buckets=4),
        example_gen_pb2.SplitConfig.Split(name='dev', hash_buckets=1)
    ]))

example_gen = tfx.components.CsvExampleGen(
    instance_name='Data_Extraction_Spliting',
    input=external_input(DATA_ROOT),
    output_config=output_config
)
```

ML project lifecycle benefits

- Configurable and reproducible data partitioning and shuffling
- External CSV, Avro, Parquet, BigQuery, and TFRecord ingestion
- Apache Beam supported for scalable data ingestion
- Customizable to new input data formats and ingestion methods

Component: **StatisticsGen**

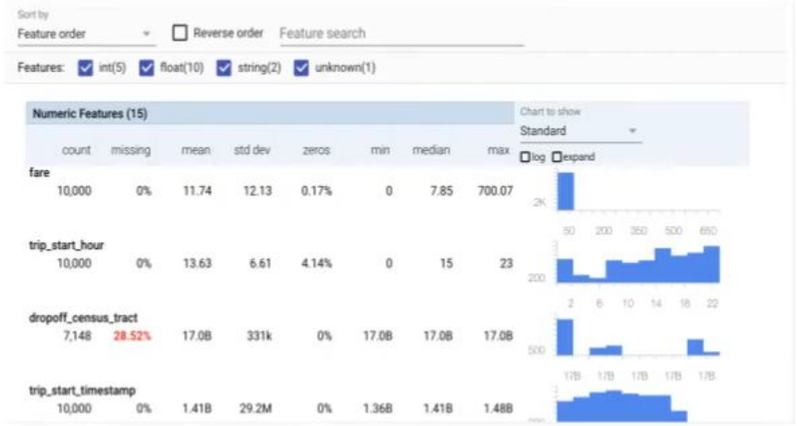
Inputs and Outputs



Configuration

```
statistics_gen = tfx.components.StatisticsGen(
    instance_name='Statistics_Generation',
    examples=example_gen.outputs['examples'])
```

TFDV Feature Statistics Visualization

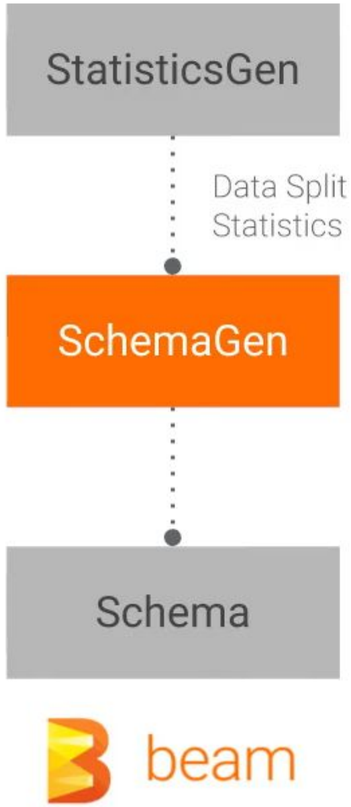


ML project lifecycle benefits

- TensorFlow Data Validation (TFDV) library for calculating feature statistics
- Scalable full-pass dataset feature statistics processing with Apache Beam

Component: **SchemaGen**

Inputs and Outputs



Configuration

```
schema_gen = SchemaGen(
    statistics=statistics_gen.outputs['statistics'],
    infer_feature_shape=False)
```

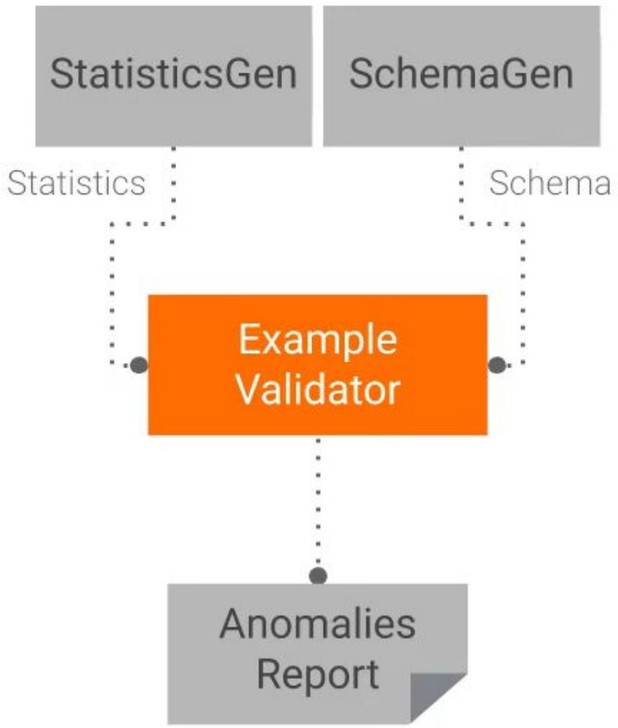
Feature name	Type	Presence	Valency	Domain
'fare'	FLOAT	required	single	-
'trip_start_hour'	INT	required	single	-
'pickup_census_tract'	BYTES	optional		-
'dropoff_census_tract'	FLOAT	optional	single	-
'company'	STRING	optional	single	'company'

ML project lifecycle benefits

- TensorFlow Data Validation (TFDV) creates common data description for pipeline components
- A Schema enables continuous data monitoring and validation during pipeline training

Component: **ExampleValidator**

Inputs and Outputs



Configuration

```
example_validator = ExampleValidator(
    statistics=statistics_gen.outputs['statistics'],
    schema=schema_importer.outputs['result'],
    instance_name="Data_Validation"
)
```

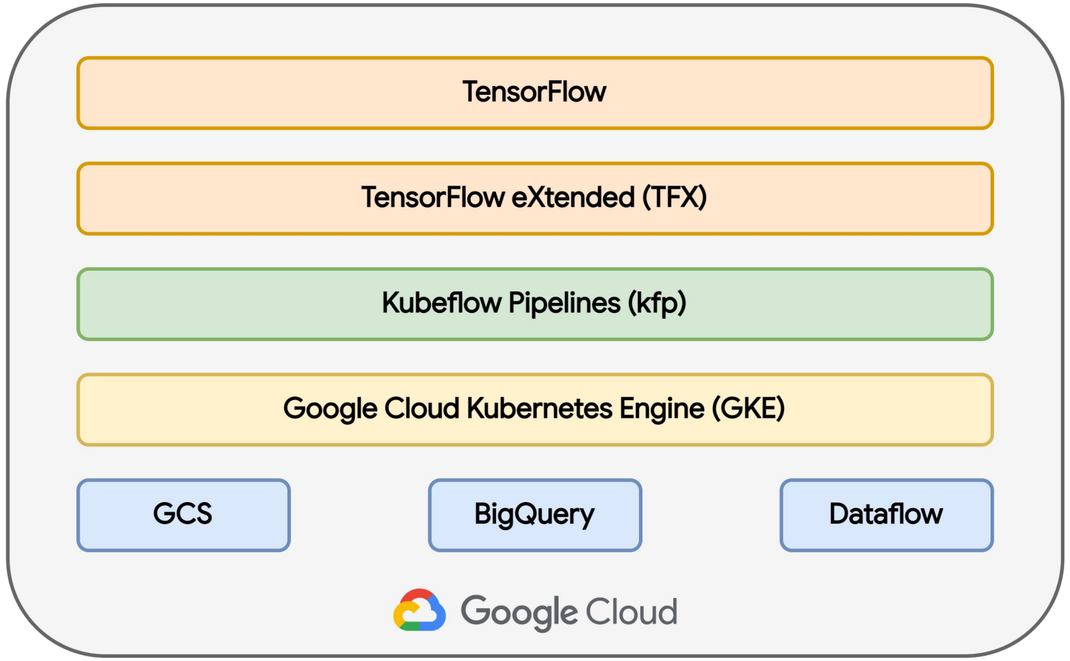
Visualization

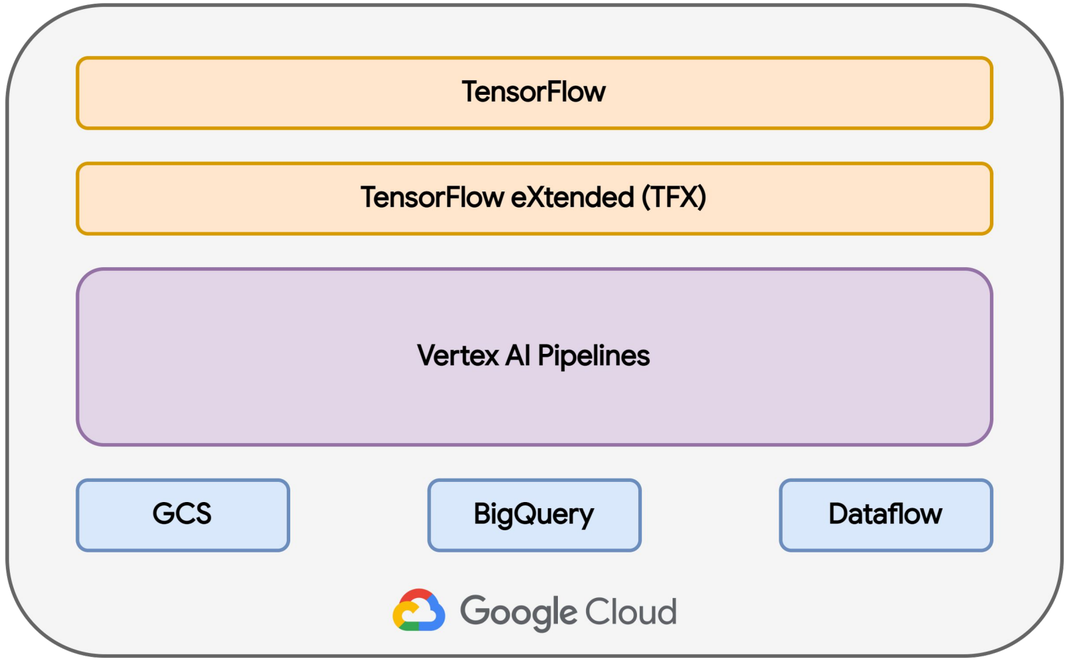
Feature name	Anomaly short description	Anomaly long description
'payment_type'	Unexpected string values	Examples contain values missing from the schema: Prcard (<1%).
'company'	Unexpected string values	Examples contain values missing from the schema: 2092 - 61288 Sbeih company (<1%), 2192 - 73487 Zeymane Corp (<1%), 2192 - Zeymane Corp (<1%), 2823 - 73307 Seung Lee (<1%), 3094 - 24059 G.L.B. Cab Co (<1%), 3319 - CD Cab Co (<1%), 3385 - Eman Cab (<1%), 3897 - 57856 Ilie Malec (<1%), 4053 - 40193 Adwar H. Nikola (<1%), 4197 - Royal Star (<1%), 585 - 88805 Valley Cab Co (<1%), 5874 - Sergey Cab Corp. (<1%), 6057 - 24657 Richard Addo (<1%), 6574 - Babylon Express Inc. (<1%), 6742 - 83735 Tasha ride inc (<1%).

ML project lifecycle benefits

- Data monitoring for detection of anomalies, train-serving skew, and data drift

MLOps with TFX in Google Cloud, (can be used in either)

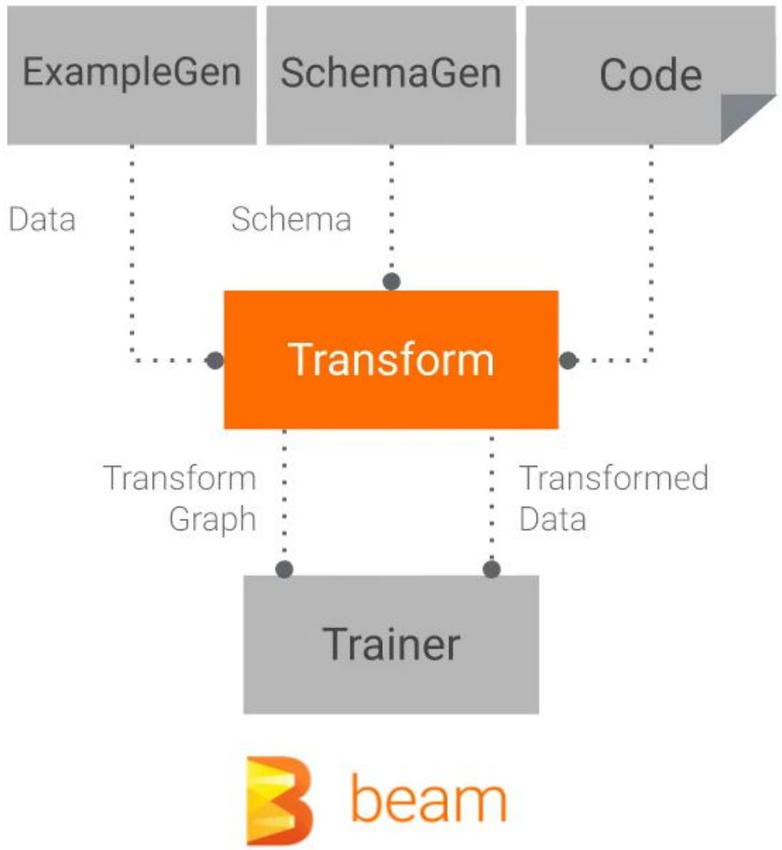




Google Cloud

Component: **Transform**

Inputs and Outputs



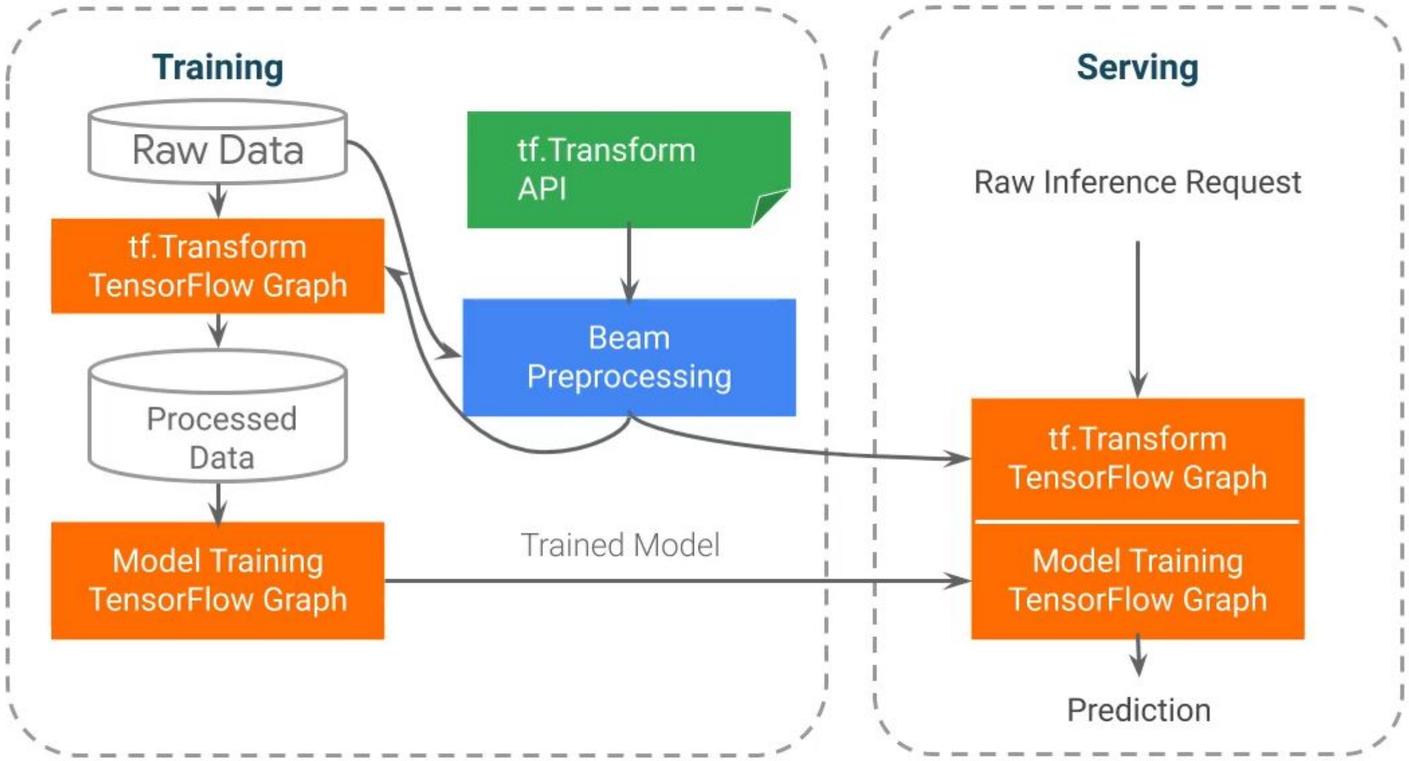
Configuration

```
transform = Transform(
    examples=example_gen.outputs['examples'],
    schema=schema_importer.outputs['result'],
    module_file=TRANSFORM_MODULE)
```

Tensorflow Transform (TFT) library code in pipeline/preprocessing.py

```
def preprocessing_fn(inputs):

  outputs = {}

  # Scale numerical features
  for key in features.NUMERIC_FEATURE_KEYS:
    outputs[features.transformed_name(key)] = tft.scale_to_z_score(
        _fill_in_missing(inputs[key]))

  # Generate vocabularies and maps categorical features
  for key in features.CATEGORICAL_FEATURE_KEYS:
    outputs[features.transformed_name(key)] = tft.compute_and_apply_vocabulary(
        x=_fill_in_missing(inputs[key]), num_oov_buckets=1, vocab_filename=key)

  outputs[features.transformed_name(features.LABEL_KEY)] = _fill_in_missing(
      inputs[features.LABEL_KEY])

  return outputs
```

Component: **Transform**

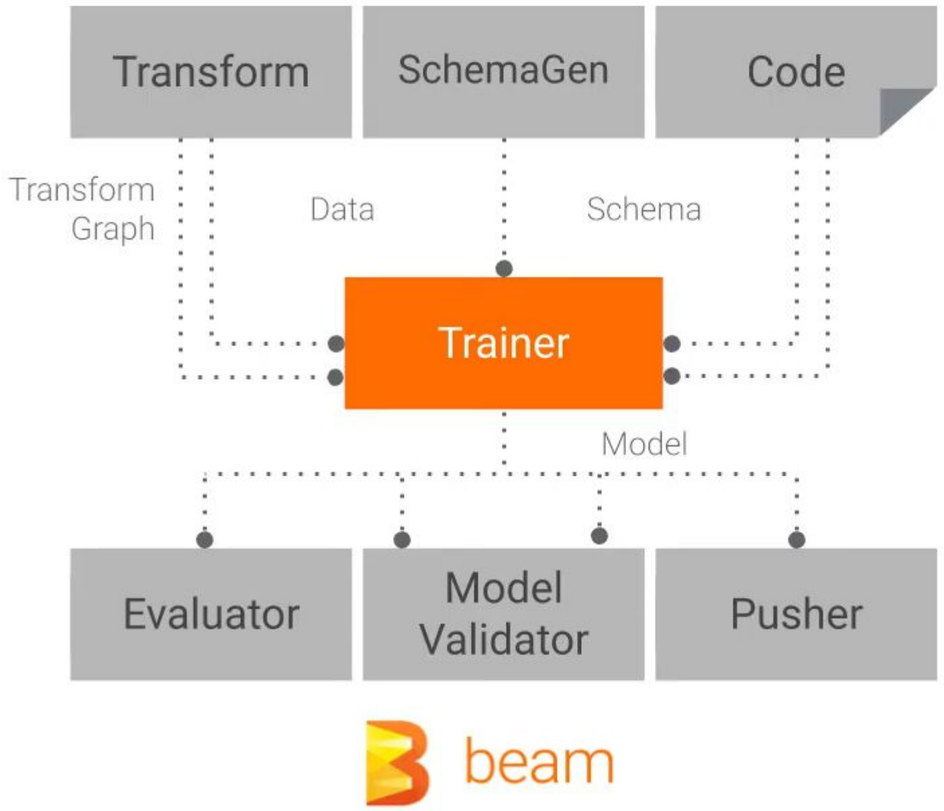


ML project lifecycle benefits

- Generates SavedModel for consistent training and serving feature engineering
- Backed by Apache Beam for scalable distributed data processing to large datasets

Component: **Trainer**

Inputs and Outputs

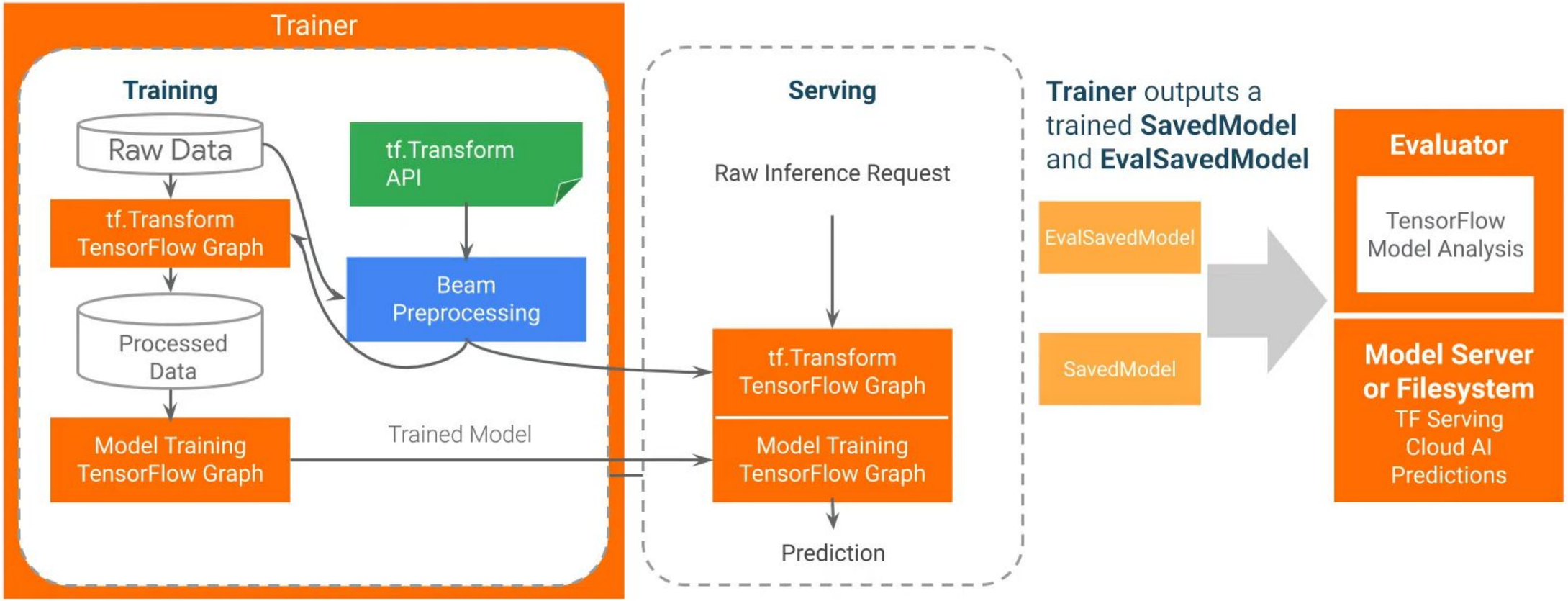


Configuration

```
trainer = Trainer(
custom_executor_spec=executor_spec.ExecutorClassSpec(trainer_executor.Gen
ericExecutor),
    module_file=model.py,
    transformed_examples=transform.outputs["transformed_examples"],
    schema=schema_importer.outputs["result"],
    transform_graph=transform.outputs["transform_graph"],
    train_args=trainer_pb2.TrainArgs(num_steps=5000),
    eval_args=trainer_pb2.EvalArgs(num_steps=1000))
```

Code -> your TensorFlow code in a pipeline/models/model.py

Component: **Trainer**

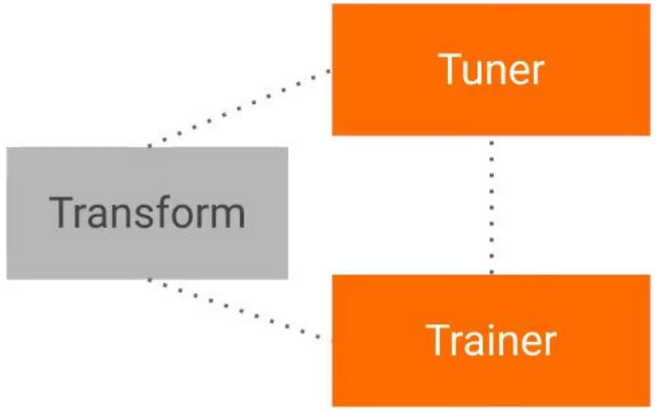


ML project lifecycle benefits

- Produces standardized TensorFlow SavedModel model artifact for sharing and easier deployment
- Configurable with Generic Trainer for any TensorFlow model API such Estimator, tf.Keras, and TFLite

Component: **Tuner**

Tuner works with Trainer



Configuration

```
tuner = Tuner(
    module_file=module_file,  # Contains `tuner_fn`.
    examples=transform.outputs['transformed_examples'],
    transform_graph=transform.outputs['transform_graph'],
    train_args=trainer_pb2.TrainArgs(num_steps=20),
    eval_args=trainer_pb2.EvalArgs(num_steps=5))
```

Best hyperparameters added to Trainer in pipeline

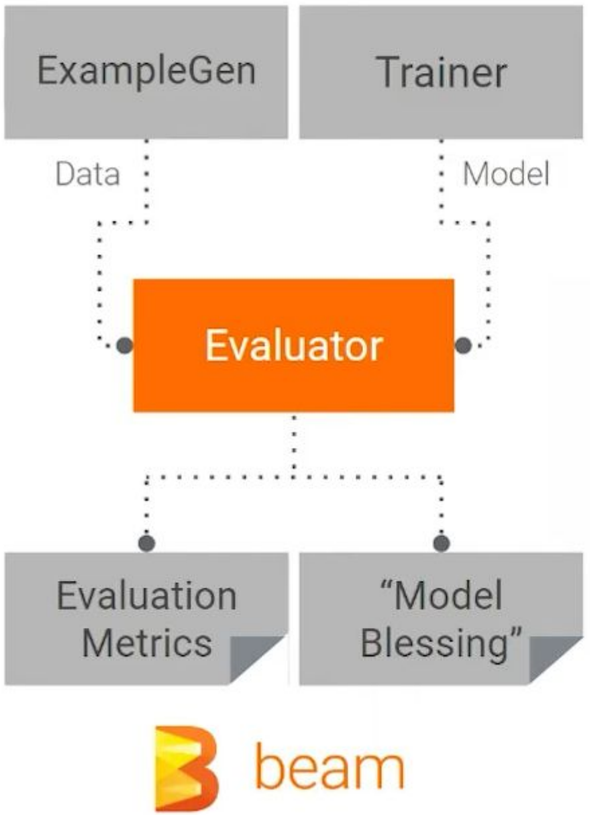
```
trainer = Trainer(
    module_file=module_file,  # Contains `run_fn`.
    custom_executor_spec=executor_spec.
        ExecutorClassSpec(GenericExecutor),
    examples=transform.outputs['transformed_examples'],
    transform_graph=transform.outputs['transform_graph'],
    schema=schema_gen.outputs['schema'],
    # This will be passed to `run_fn`.
    hyperparameters=tuner.outputs['best_hyperparameters'],
    train_args=trainer_pb2.TrainArgs(num_steps=100),
    eval_args=trainer_pb2.EvalArgs(num_steps=5))
```

ML project lifecycle benefits

- Supports KerasTuner library for hyperparameter tuning and integration with TensorFlow Trainer
- Can use Google Cloud AI Platform Optimizer for distributed tuning

Component: **Evaluator**

Inputs and Outputs



beam

Configuration

```
model_analyzer = Evaluator(
    examples=example_gen.outputs.examples,
    model=trainer.outputs.model,
    baseline_model=model_resolver.outputs.model,
    eval_config=eval_config)
```

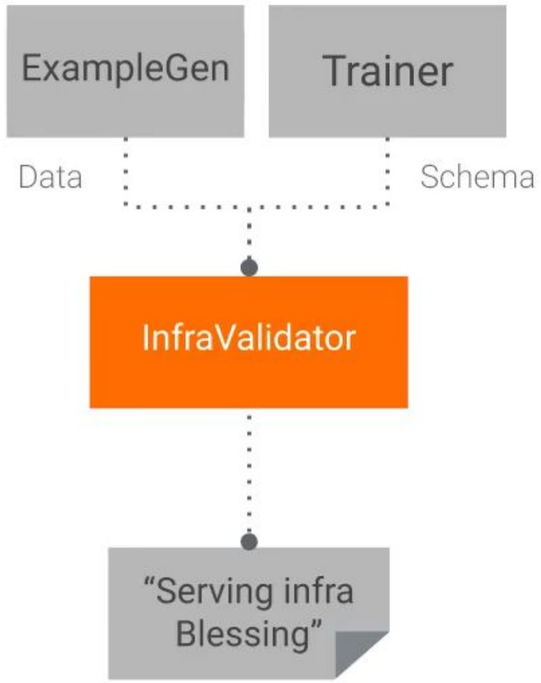
TensorFlow Model Analysis (TFMA) library for visualizing model evaluation

ML project lifecycle benefits

- Uses TensorFlow Model Analysis (TFMA) and Apache Beam for scalable model evaluation across data splits and slices.
- Validate model performance "good enough" to be pushed to production

Component: **InfraValidator**

Inputs and Outputs



Configuration

```
infra_validator = InfraValidator(
    model=trainer.outputs['model'],
    examples=example_gen.outputs['examples'],
    serving_spec=infra_validator_pb2.ServingSpec(...),
  validation_spec=infra_validator_pb2.ValidationSpec(...),
  request_spec=infra_validator_pb2.RequestSpec(...)
)
```

Configuration options

ServingSpec: Type of model server and infrastructure to test with e.g. TF Serving

ValidationSpec: Adjusts the infra validation criteria or workflow.

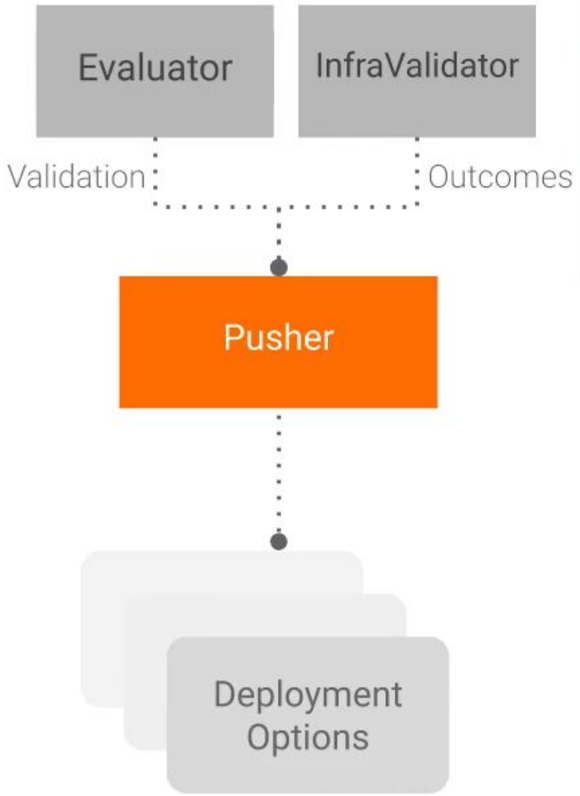
RequestSpec: Which model signature, how many examples to test.

ML project lifecycle benefits

- Validate model in model serving environment before pushing to production
- Configurable to mirror different model serving environments such as Kubernetes and TF Serving

Component: **Pusher**

Inputs and Outputs



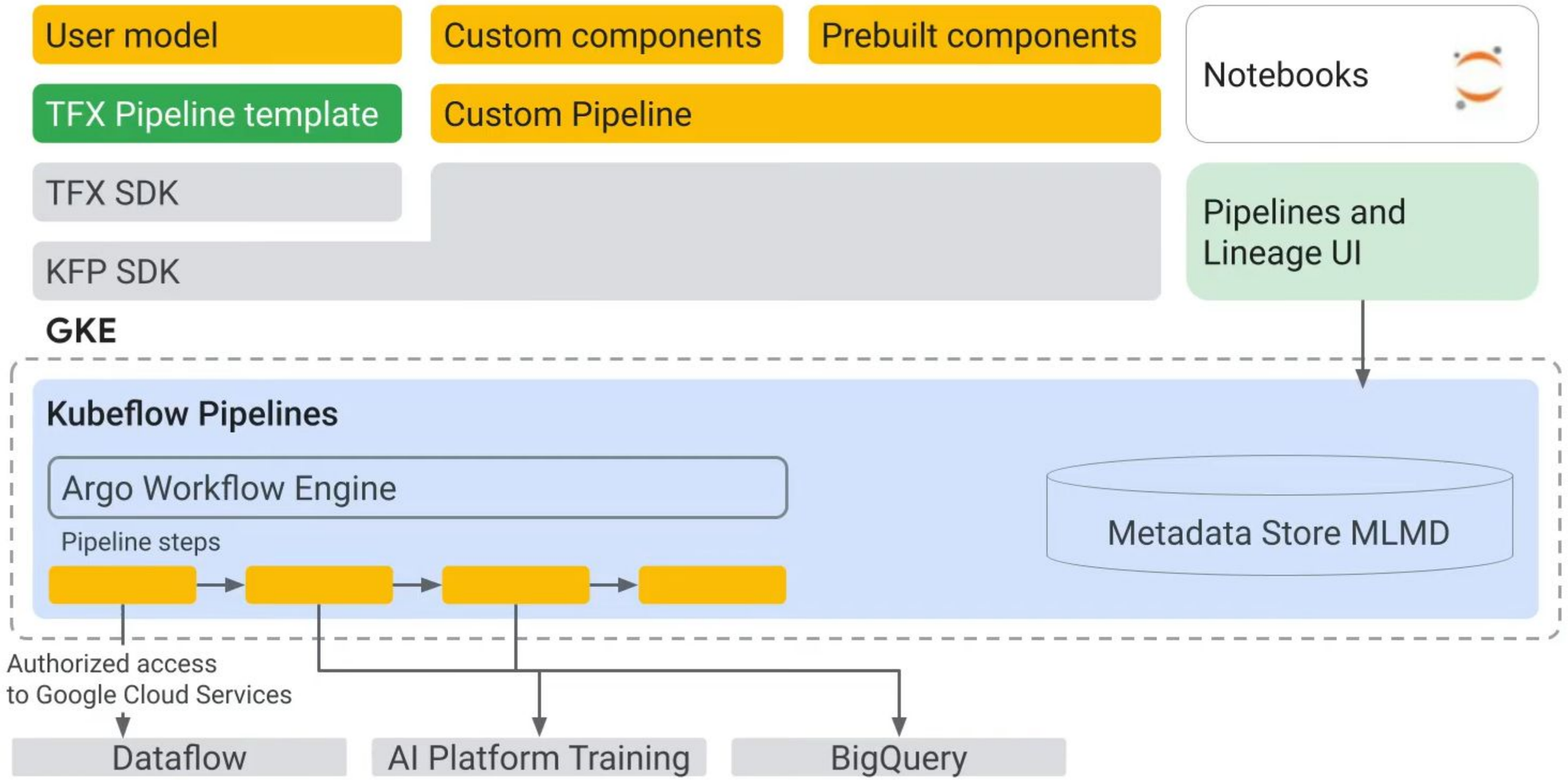
Configuration

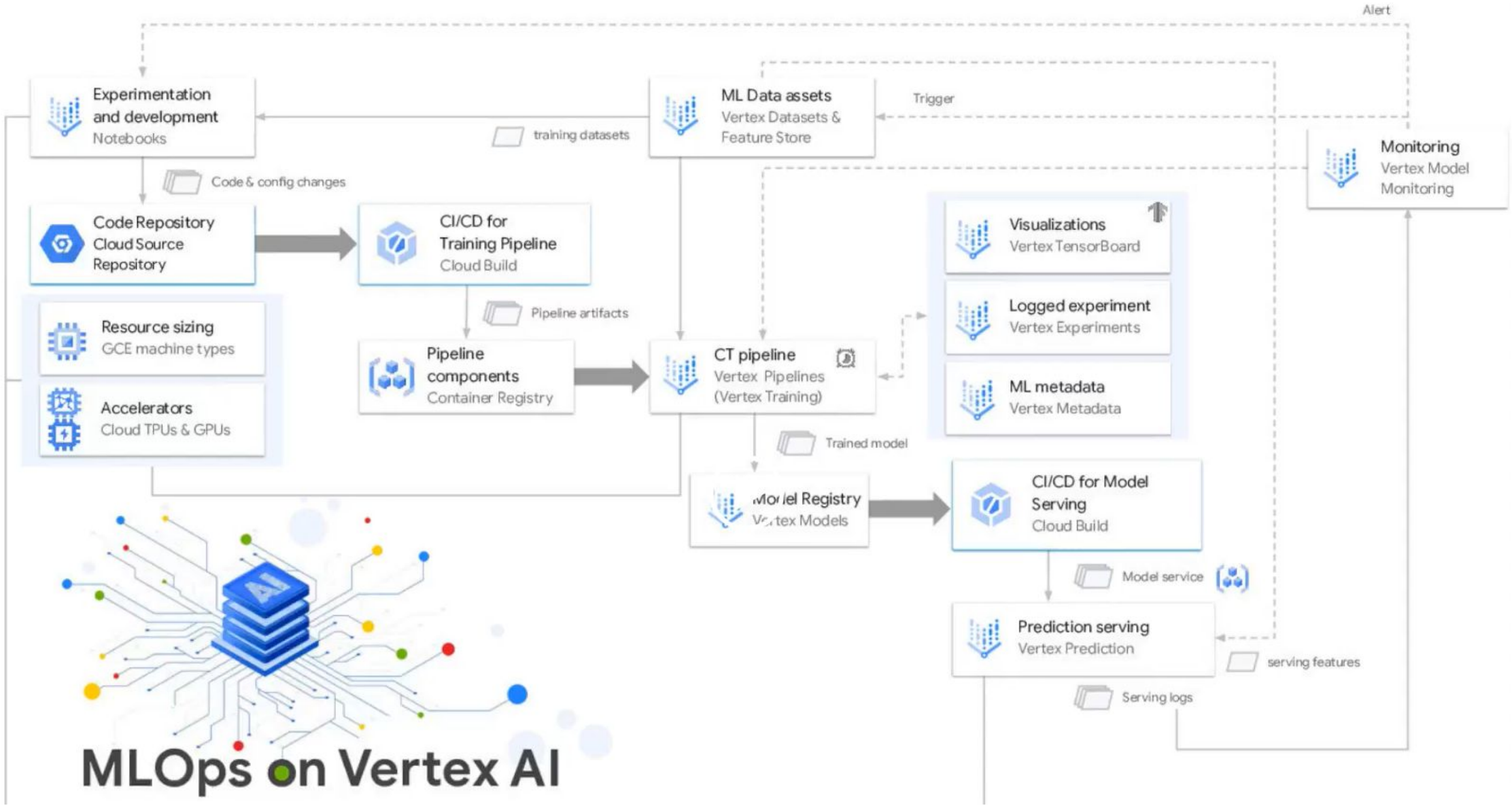
```
pusher = Pusher(
    model=trainer.outputs['model'],
    model_blessing=model_analyzer.outputs['blessing'],
    infra_blessing=infra_validator.outputs['blessing'],
    push_destination=pusher_pb2.PushDestination(
        filesystem=pusher_pb2.PushDestination.Filesystem(
            base_directory=SERVING_MODEL_DIR)))
```

ML project lifecycle benefits

- Validation checks on model performance and serving ability before pushing to production.
- Reusable model export code that is configurable for different push destinations:
  - Filesystem (TensorFlow Lite, TensorFlow JS)
  - Model Server (TensorFlow Serving)

Details view: TFX pipelines run on Google Cloud





Google Cloud

Thank you
End of Session 4

Google Cloud


Google Cloud