TensorFlow

TensorFlow Extended (TFX)

An End-to-End ML Platform

Clemens Mewald

TensorFlow Extended (TFX): An End-to-End ML Platform

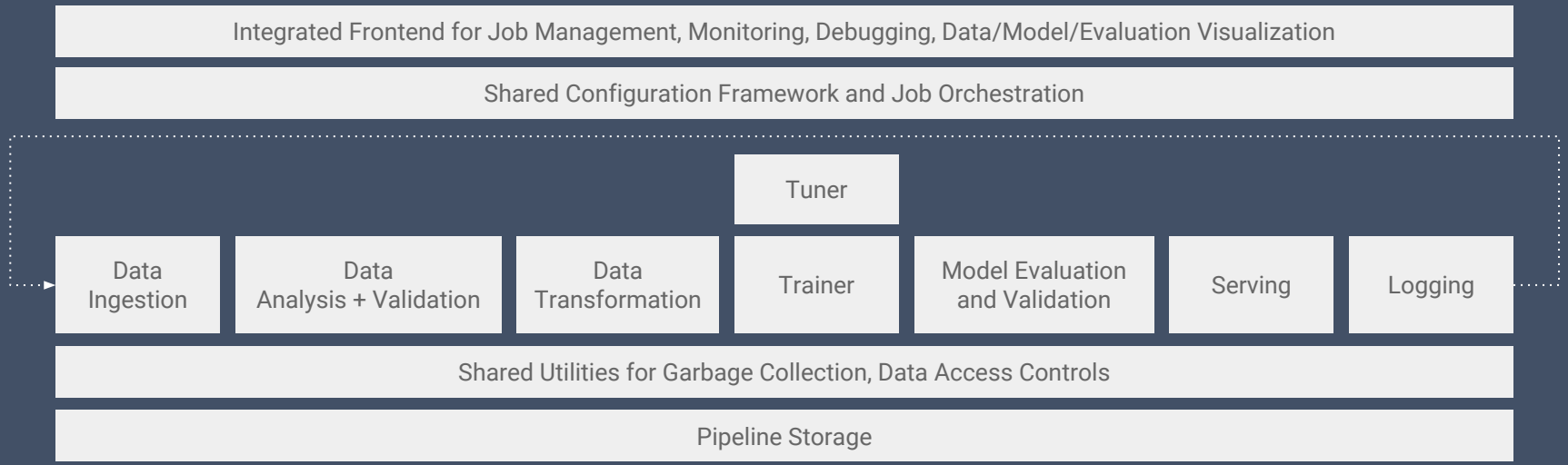
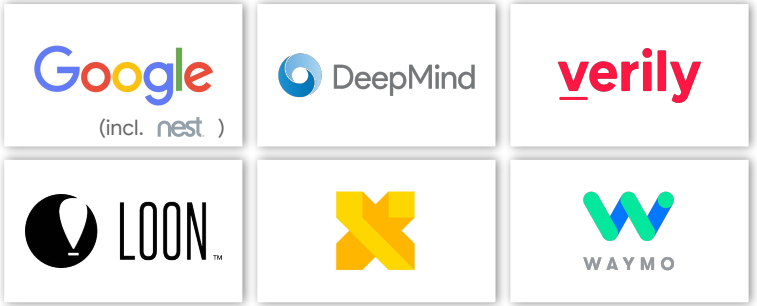


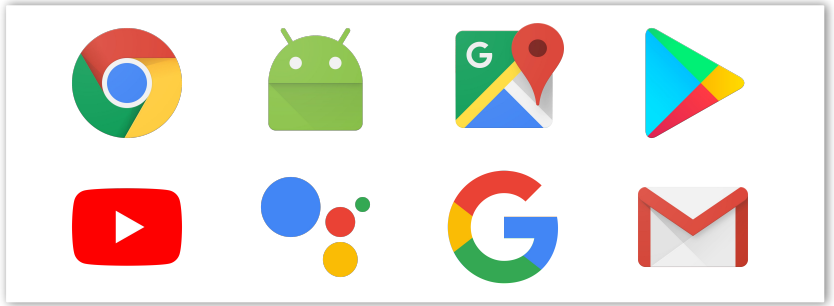
Figure 1: High-level component overview of a machine learning platform.

TFX powers our most important bets and products...

AlphaBets



Major Products



So far, we've made some of our libraries available.

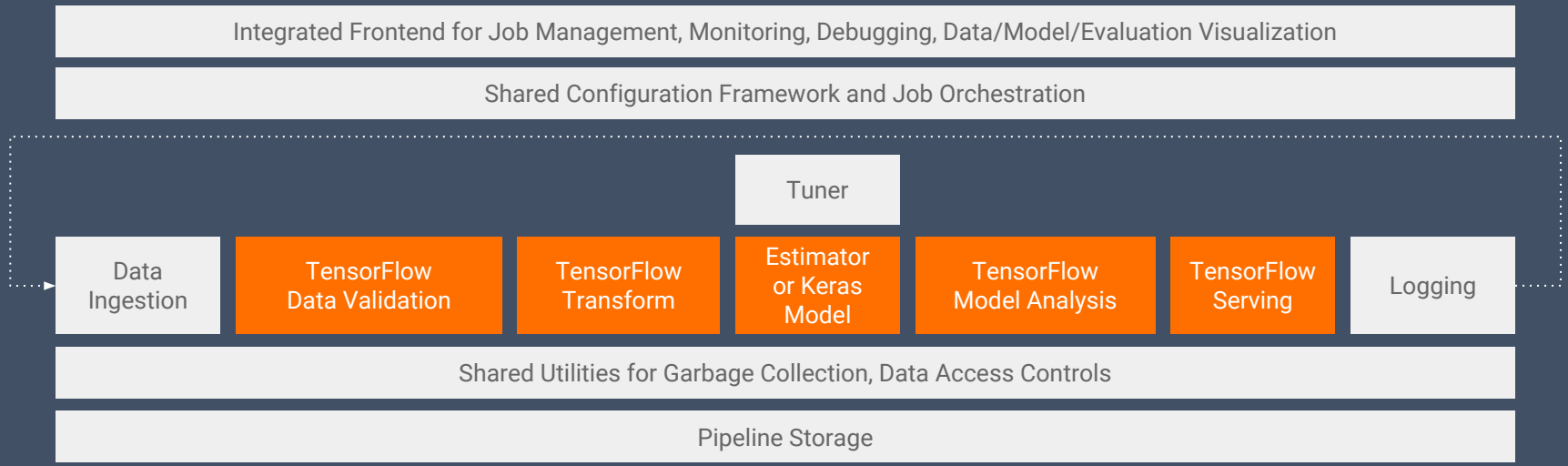
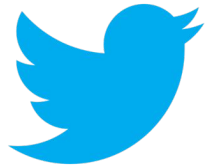


Figure 1: High-level component overview of a machine learning platform.

... and some of our most important partners.



So far, we've made some of our libraries available.

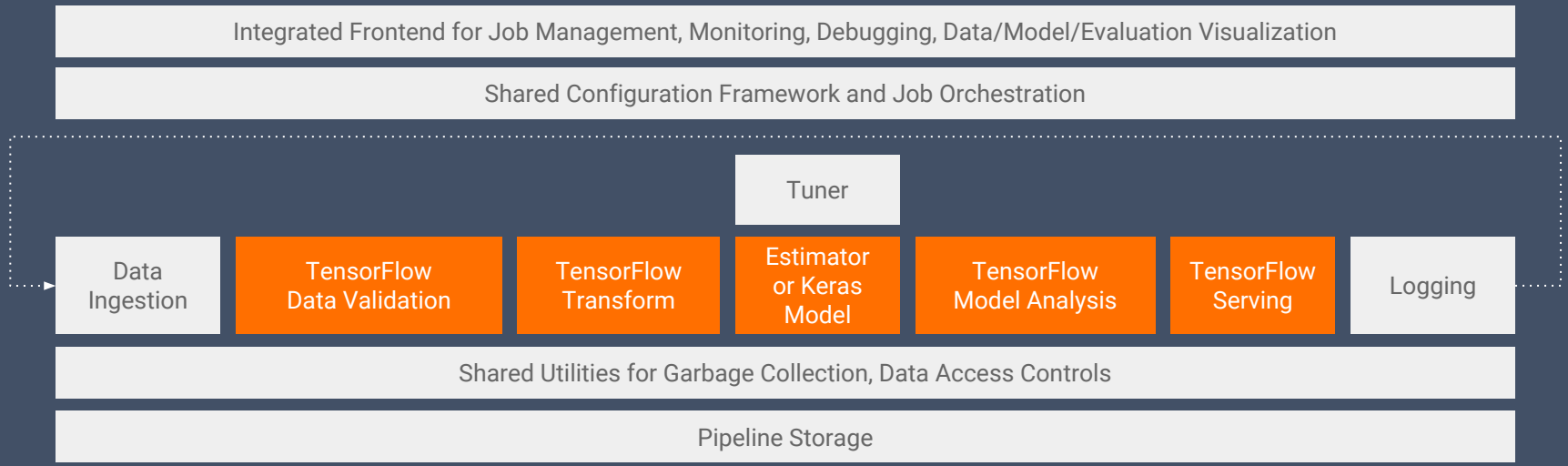


Figure 1: High-level component overview of a machine learning platform.

Today, we share the horizontal layers that integrate libraries in one product.

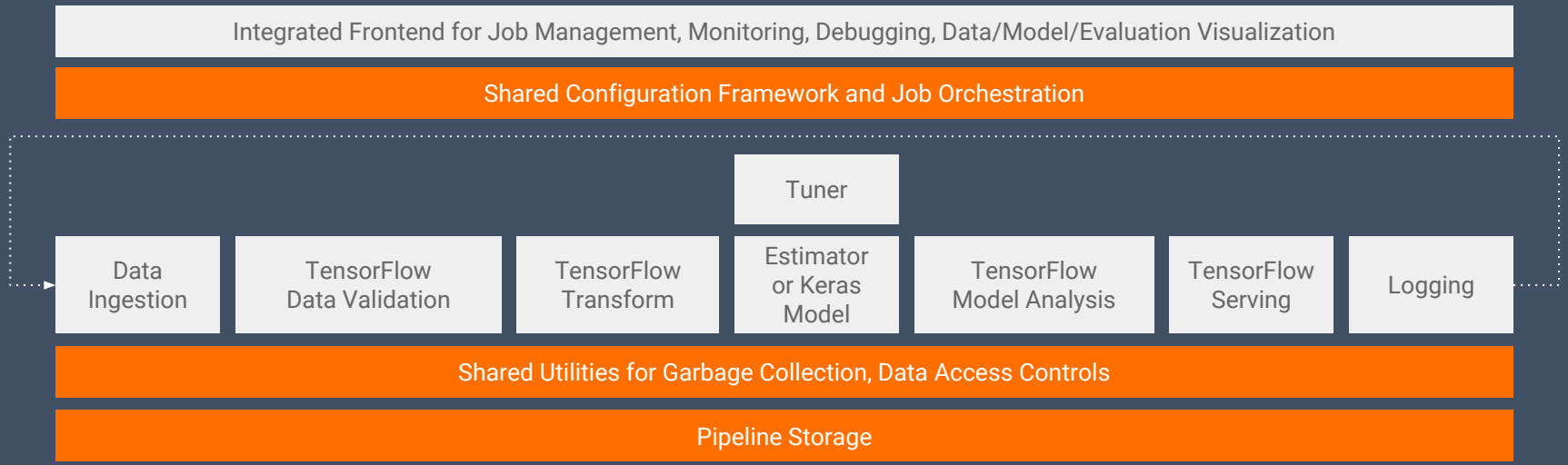
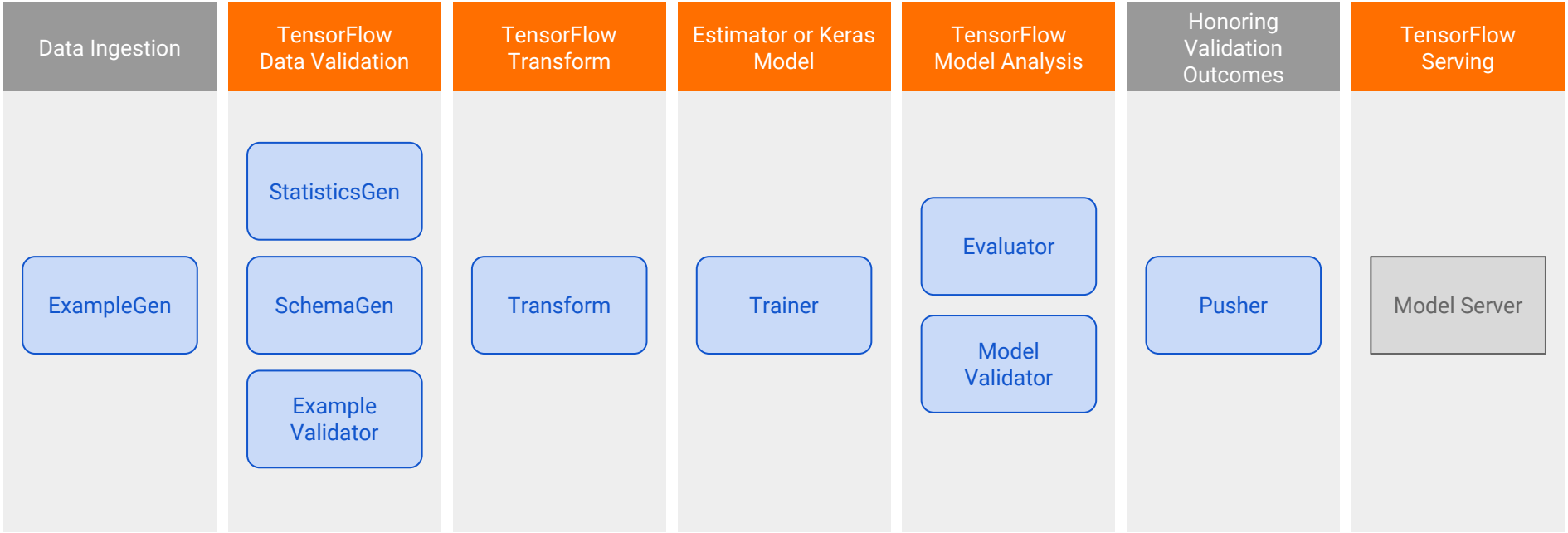
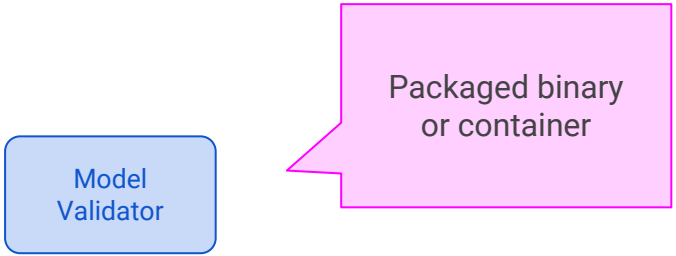


Figure 1: High-level component overview of a machine learning platform.

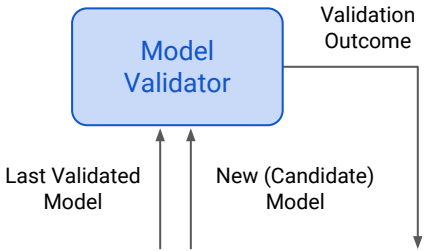
Building Components out of Libraries



What makes a Component

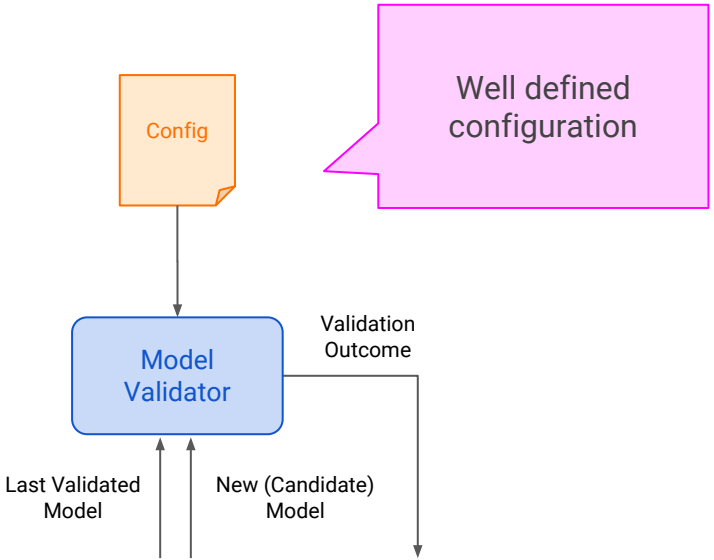


What makes a Component

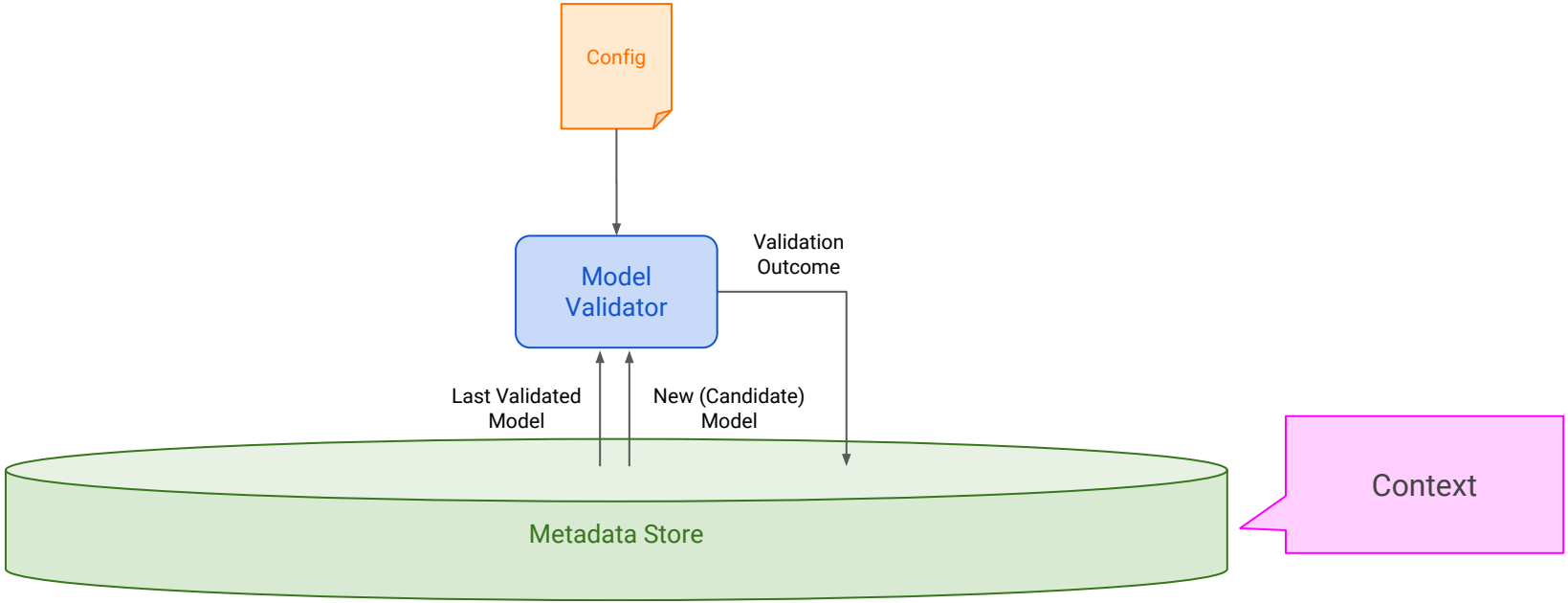


Well defined
inputs and outputs

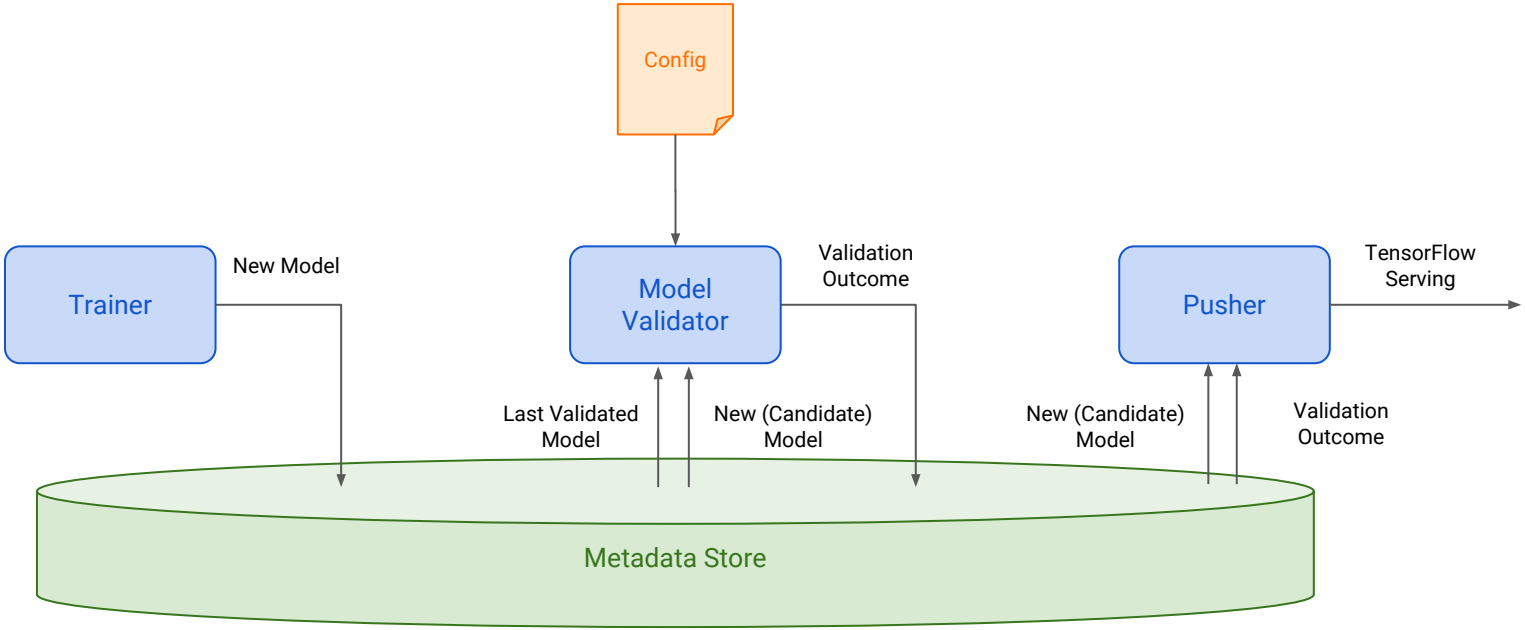
What makes a Component



What makes a Component



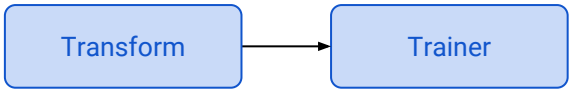
What makes a Component



Metadata Store? That’s new

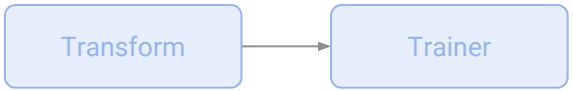
Metadata Store? That's new

Task-Aware Pipelines

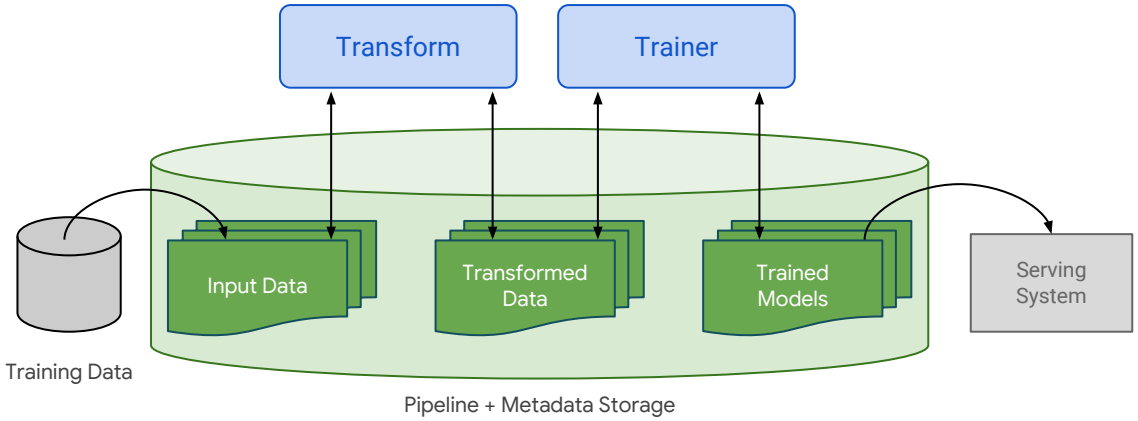


Metadata Store? That's new

Task-Aware Pipelines



Task- **and** Data-Aware Pipelines



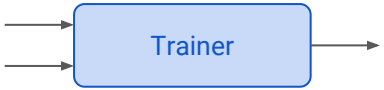
What's in the Metadata Store?

Type definitions of Artifacts and their Properties

E.g., Models, Data, Evaluation Metrics

What's in the Metadata Store?

Type definitions of Artifacts and their Properties
E.g., Models, Data, Evaluation Metrics

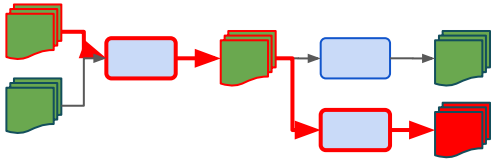


Execution Records (Runs) of Components
E.g., Runtime Configuration, Inputs + Outputs

What's in the Metadata Store?

Type definitions of Artifacts and their Properties
E.g., Models, Data, Evaluation Metrics

Execution Records (Runs) of Components
E.g., Runtime Configuration, Inputs + Outputs



Lineage Tracking Across All Executions
E.g., to recurse back to all inputs of a specific artifact

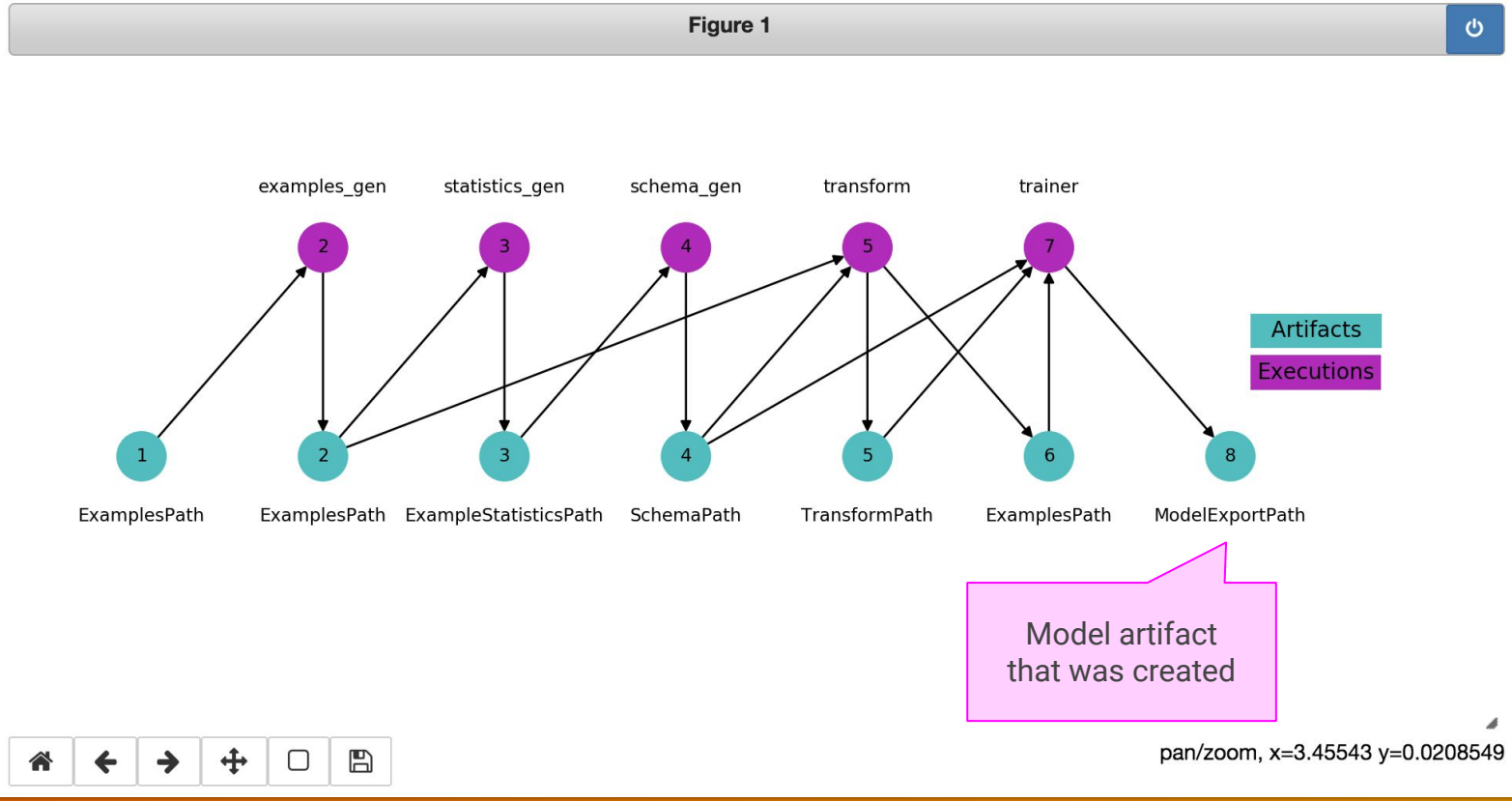
List all training runs and attributes

```
# Visualize all TFX Trainer runs.
display(get_executions_of_type_df(TFX_EXECUTION_TRAINER))
```

ID	EVAL_STEPS	TRAIN_STEPS	LOG_ROOT	WARM_START_FROM	WARM_STARTING	STATE	CHECKSUM_M
7	5000	10000	/var/tmp/tfx/logs/chicago_taxi_pipeline_local/...	None	True	complete	1477864c3749c2c49466624f83a163
17	5000	10000	/var/tmp/tfx/logs/chicago_taxi_pipeline_local/...	None	True	complete	1477864c3749c2c49466624f83a163

Visualize lineage of a specific model

```
# Visualize the lineage of model_id.
num_hops = None
get_and_plot_artifact_lineage(model_id, num_hops)
```



Visualize data a model was trained on

```
# Visualize stats for data that was used to generate model_id.
display_data_stats_for_model(model_id)
```

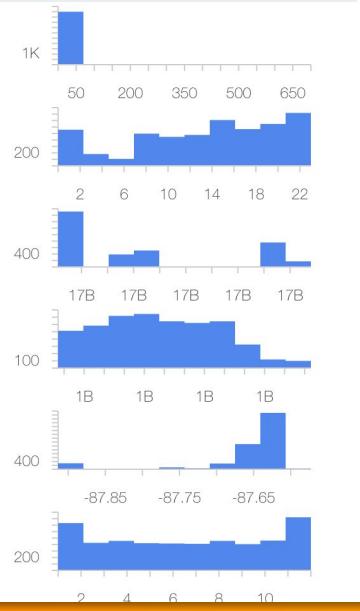
Sort by: Feature order | Reverse order | Feature search (regex enabled)

Features: int(5) | float(5) | variable-length floats(5) | string(1) | fixed-length strings(1) | variable-length strings(1)

Numeric Features (15)

Chart to show: Standard (log, expand)

	count	missing	mean	std dev	zeros	min	median	max
fare	5,009	0%	11.97	14.12	0.12%	0	7.85	700.07
trip_start_hour	5,009	0%	13.51	6.73	4.13%	0	15	23
dropoff_census_tract	5,009	0%	17.0B	328k	0%	17.0B	17.0B	17.0B
trip_start_timestamp	5,009	0%	1.41B	29.0M	0%	1.36B	1.41B	1.48B
pickup_longitude	5,009	0%	-87.66	0.07	0%	-87.91	-87.63	-87.57
trip_start_month	5,009	0%	6.6	3.4	0%	1	7	12



Visualize sliced eval metrics associated with a model

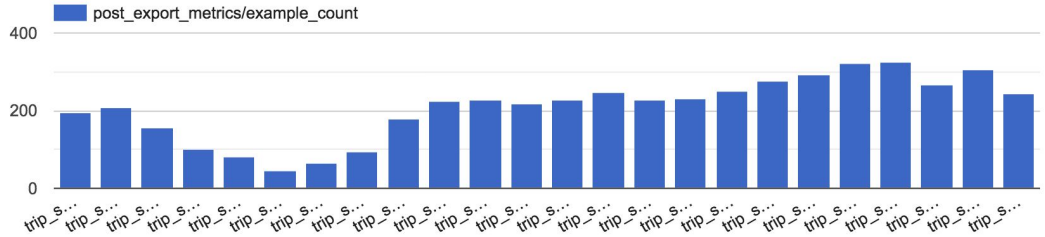
```
# Visualize TFMA analysis for model_id.
display_tfma_analysis(model_id, slicing_column='trip_start_hour')
```

Visualization: Slices Overview

Examples (Weighted) Threshold: 0

Show: post_export_metrics/example_count

Sort by: Slice

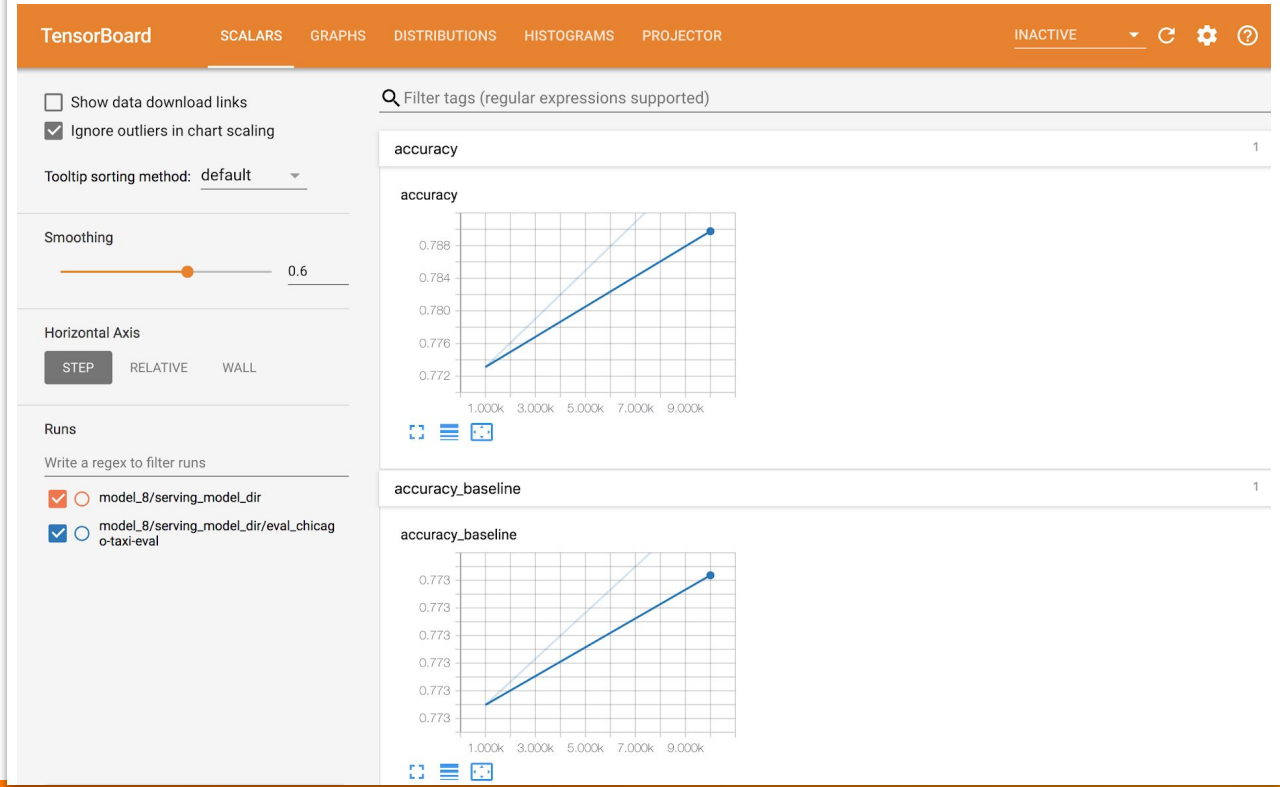


feature	accuracy	accuracy_baseline	auc	auc_precision_recall	average_loss
trip_start_hour:1	0.78462	0.76923	0.95548	0.81083	0.34285
trip_start_hour:0	0.74879	0.72464	0.95678	0.79841	0.36254
trip_start_hour:3	0.79000	0.80000	0.90781	0.59816	0.34760
trip_start_hour:2	0.76774	0.76129	0.92396	0.67323	0.36896
trip_start_hour:5	0.70455	0.70455	0.80645	0.58333	0.44569
trip_start_hour:4	0.80000	0.78750	0.96872	0.84782	0.31579
trip_start_hour:7	0.87368	0.86316	0.94887	0.61116	0.27600

Launch TensorBoard for a specific model run

```
# Open up Tensorboard for model_id.
print(display_tensorboard(model_id))
```

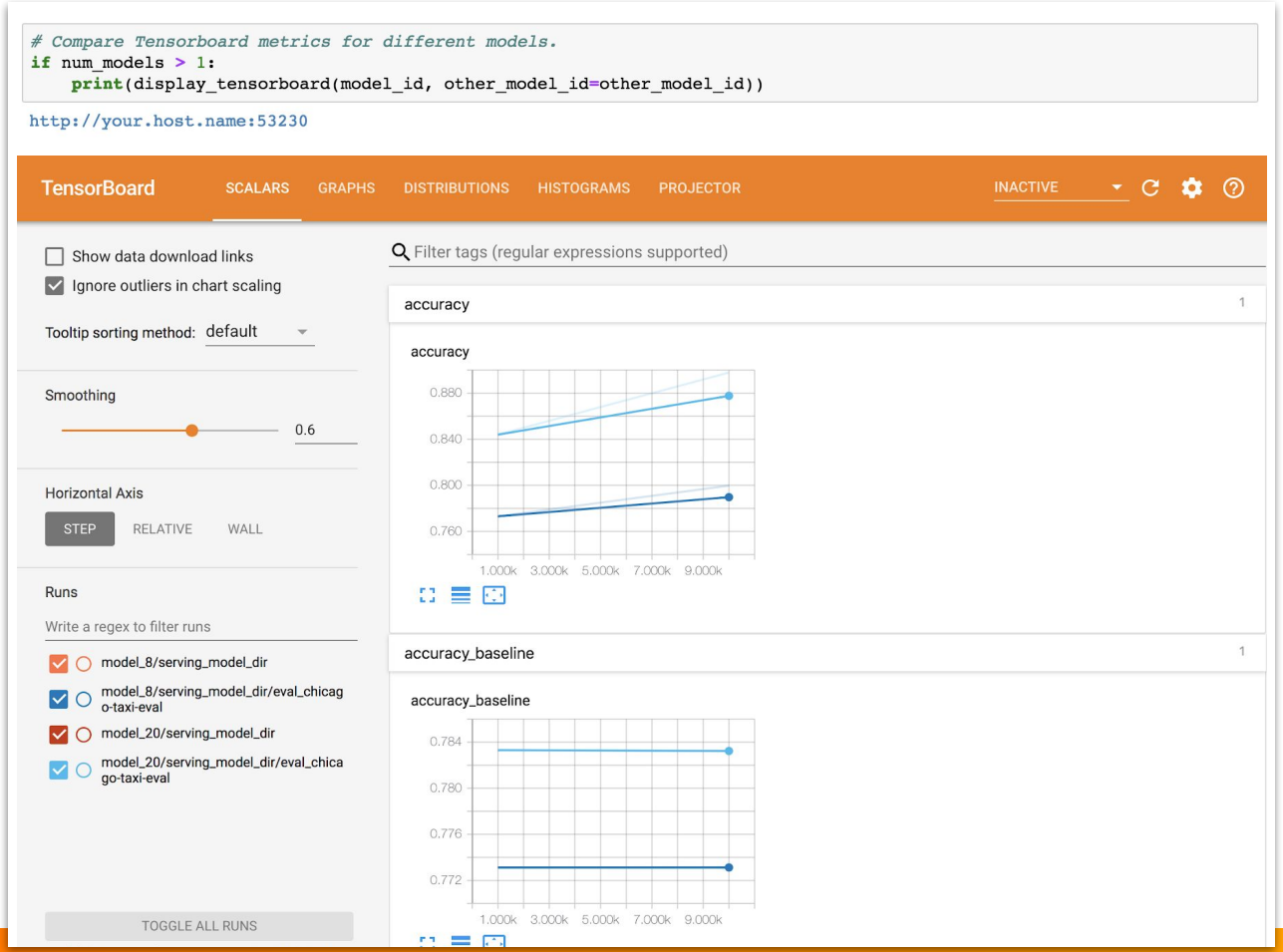
```
http://your.host.name:53143
```



Launch TensorBoard to compare multiple model runs

```
# Compare Tensorboard metrics for different models.
if num_models > 1:
    print(display_tensorboard(model_id, other_model_id=other_model_id))
```

http://your.host.name:53230



Compare data statistics for multiple models

```
# Visualize stats for data that was used to generate model_id and other_model_id.
if num_models > 1:
    display_data_stats_for_model(model_id, other_model_id=other_model_id)
```

Sort by: Feature order | Reverse order | Feature search (regex enabled)

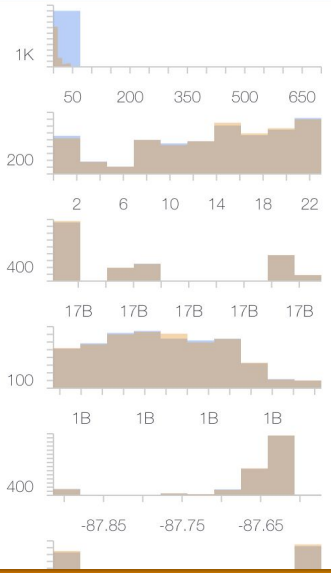
Features: int(5) float(5) variable-length floats(5) string(1) fixed-length strings(1) variable-length strings(1)

Model 8's data Model 20's data

Numeric Features (15)

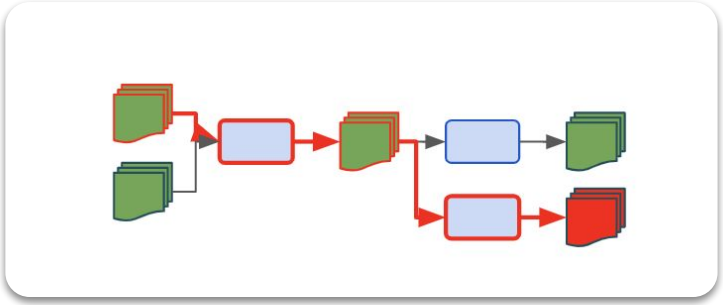
Chart to show: Standard — log, expand, percentages

	count	missing	mean	std dev	zeros	min	median	max
fare								
	5,009	0%	11.97	14.12	0.12%	0	7.85	700.07
	5,016	0%	11.81	10.35	0.16%	0	7.85	112.65
trip_start_hour								
	5,009	0%	13.51	6.73	4.13%	0	15	23
	5,016	0%	13.62	6.63	4.13%	0	15	23
dropoff_census_tract								
	5,009	0%	17.0B	328k	0%	17.0B	17.0B	17.0B
	5,016	0%	17.0B	328k	0%	17.0B	17.0B	17.0B
trip_start_timestamp								
	5,009	0%	1.41B	29.0M	0%	1.36B	1.41B	1.48B
	5,016	0%	1.41B	29.1M	0%	1.36B	1.41B	1.48B
pickup_longitude								
	5,009	0%	-87.66	0.07	0%	-87.91	-87.63	-87.57
	5,016	0%	-87.66	0.07	0%	-87.91	-87.63	-87.57
trip_start_month								



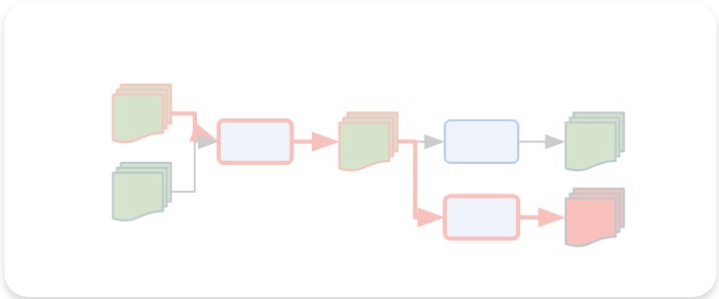
Examples of Metadata-Powered Functionality

Use-cases enabled by lineage tracking

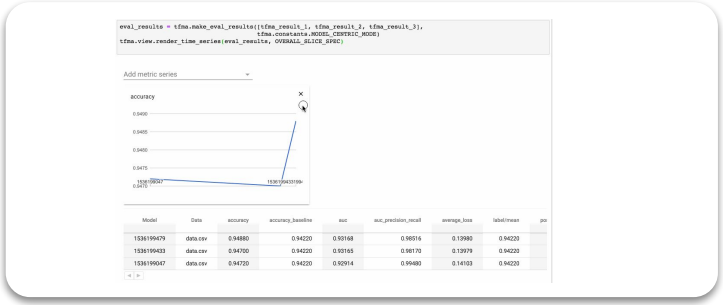


Examples of Metadata-Powered Functionality

Use-cases enabled by lineage tracking

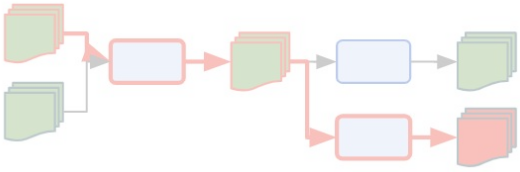


Compare previous model runs



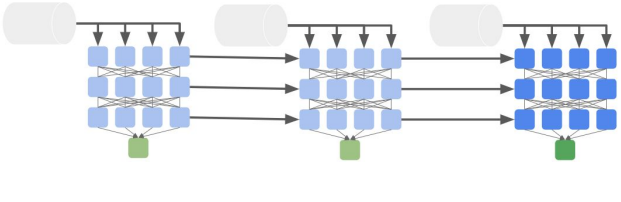
Examples of Metadata-Powered Functionality

Use-cases enabled by lineage tracking



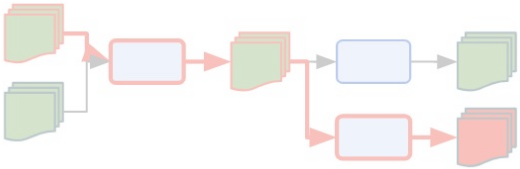
Compare previous model runs

Carry-over state from previous models



Examples of Metadata-Powered Functionality

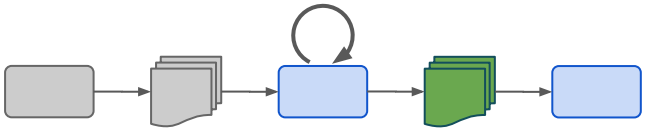
Use-cases enabled by lineage tracking



Compare previous model runs

Carry-over state from previous models

Re-use previously computed outputs

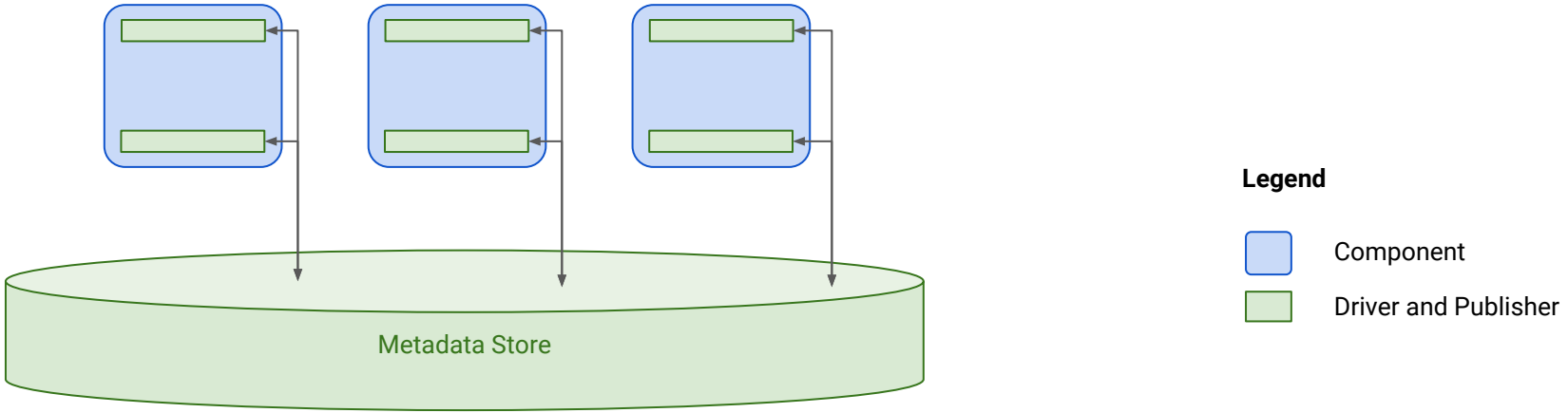


How do we orchestrate TFX?

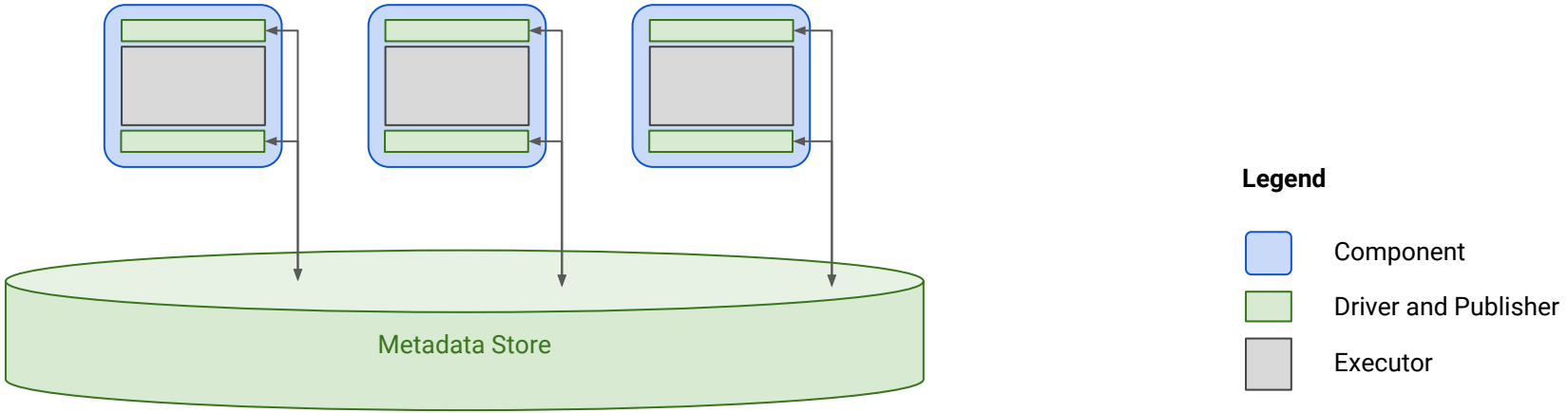
Legend

Component

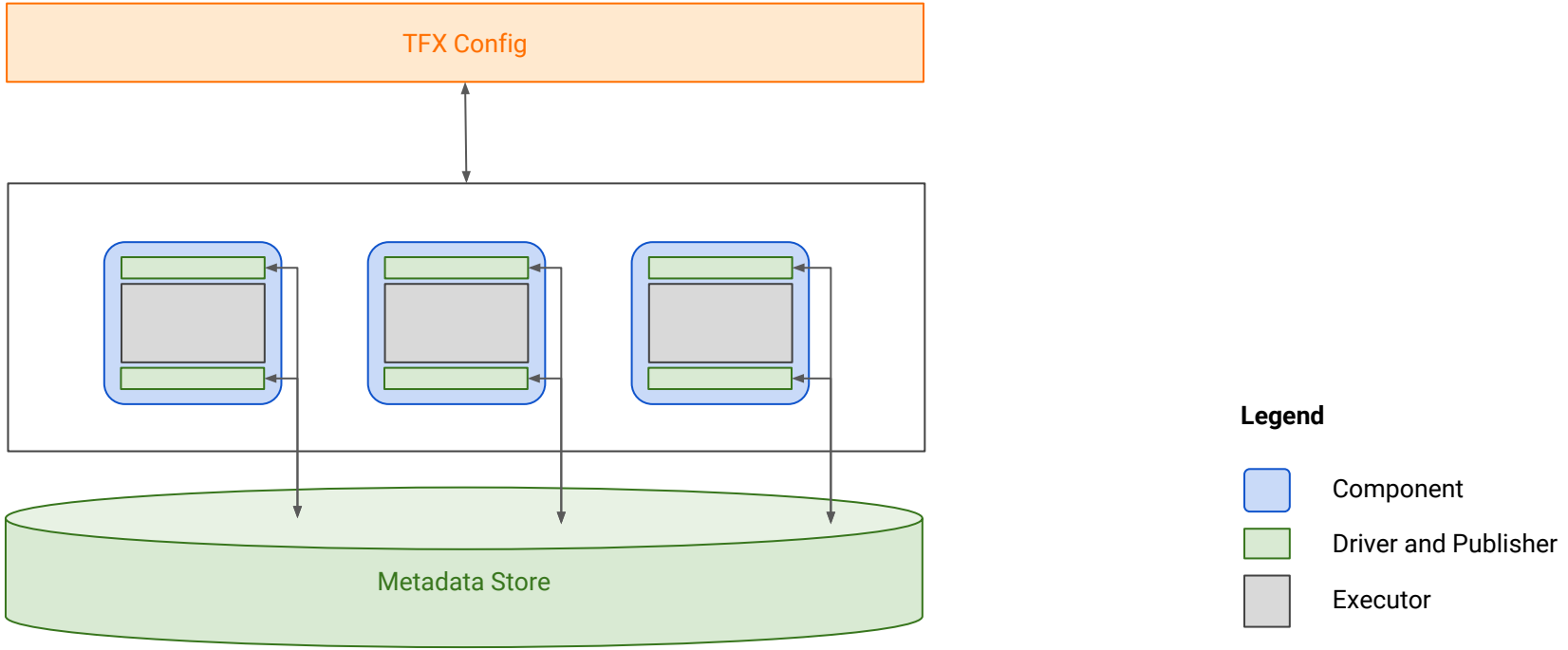
How do we orchestrate TFX?



How do we orchestrate TFX?



How do we orchestrate TFX?

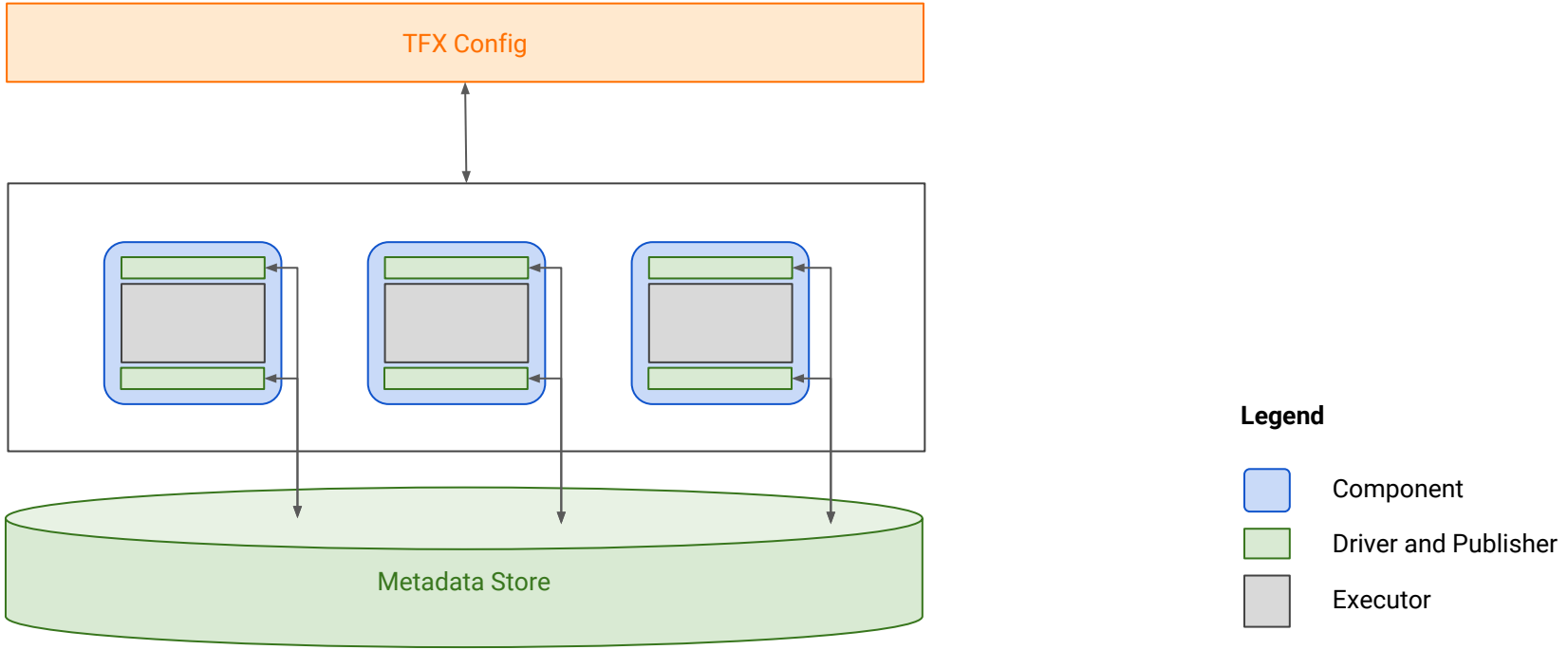


```
def create_pipeline():
  """Implements the chicago taxi pipeline with TFX."""
  examples = csv_input(os.path.join(data_root, 'simple'))
  example_gen = CsvExampleGen(input_base=examples)

  statistics_gen = StatisticsGen(input_data=...)
  infer_schema = SchemaGen(stats=...)
  validate_stats = ExampleValidator(stats=..., schema=...)
  transform = Transform(input_data=..., schema=..., module_file=...)

  trainer = Trainer(
      module_file=taxi_module_file,
      transformed_examples=transform.outputs.transformed_examples,
      transform_output=transform.outputs.transform_output,
      schema=infer_schema.outputs.output,
      train_steps=10000,
      eval_steps=5000,
      warm_starting=True)

  model_analyzer = Evaluator(examples=..., model_exports=...)
  model_validator = ModelValidator(examples=..., model=...)
  pusher = Pusher(model_export=..., model_blessing=..., serving_model_dir=...)
  return [example_gen, statistics_gen, infer_schema, validate_stats, transform, trainer,
          model_analyzer, model_validator, pusher]

pipeline = TfxRunner(airflow_config).run(create_pipeline())
```

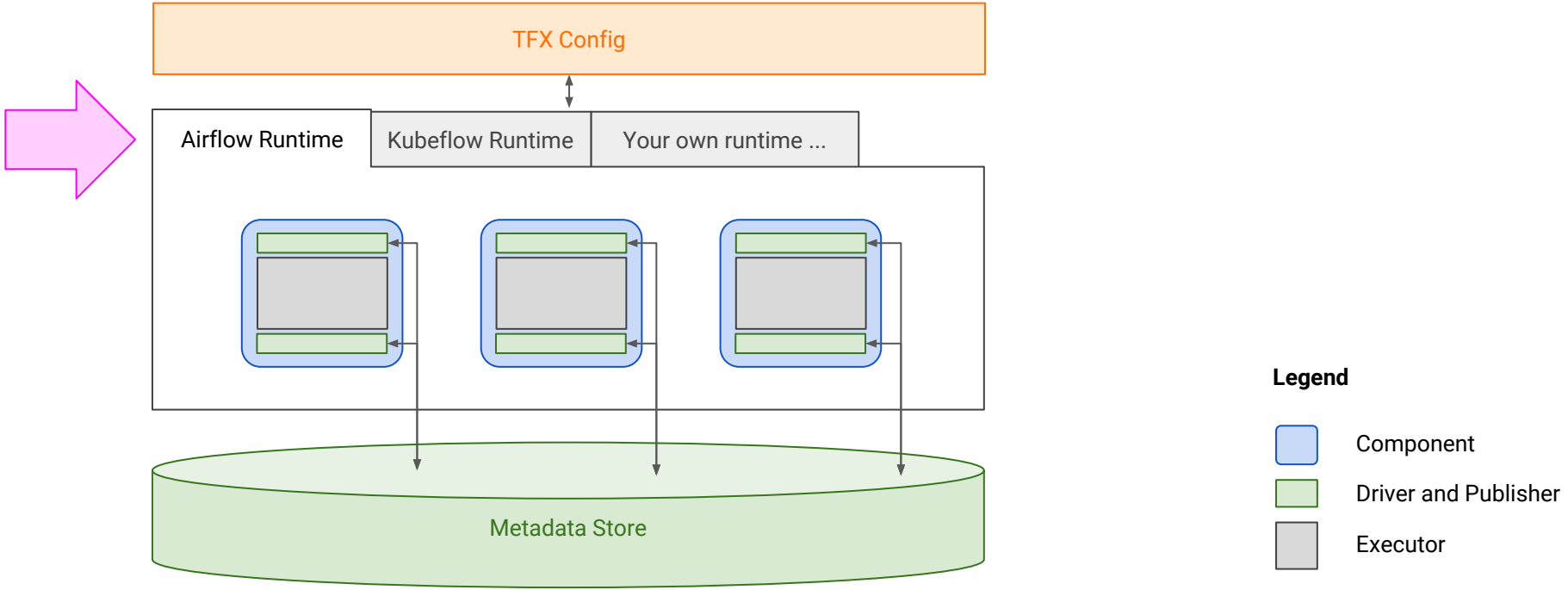
```
def train_and_maybe_evaluate(hparams):
  schema = taxi.read_schema(hparams.schema_file)

  train_input = lambda: model.input_fn(...)
  eval_input = lambda: model.input_fn(...)
  serving_receiver_fn = lambda: model.example_serving_receiver_fn(...)

  train_spec = tf.estimator.TrainSpec(...)
  eval_spec = tf.estimator.EvalSpec(...)

  exporter = tf.estimator.FinalExporter('chicago-taxi', serving_receiver_fn)

  run_config = tf.estimator.RunConfig(
      save_checkpoints_steps=999, keep_checkpoint_max=1)

  estimator = model.build_estimator(...)

  tf.estimator.train_and_evaluate(estimator, train_spec, eval_spec)

  return estimator
```

```
def build_estimator(tf_transform_dir, config, hidden_units=None):

  # receive Schema and Transform metadata
  metadata_dir = os.path.join(tf_transform_dir,
                              transform_fn_io.TRANSFORMED_METADATA_DIR)
  transformed_metadata = metadata_io.read_metadata(metadata_dir)
  transformed_feature_spec = transformed_metadata.schema.as_feature_spec()

  transformed_feature_spec.pop(taxi.transformed_name(taxi.LABEL_KEY))

  real_valued_columns = [...]
  categorical_columns = [...]

  return tf.estimator.DNNLinearCombinedClassifier(
      config=config,
      linear_feature_columns=categorical_columns,
      dnn_feature_columns=real_valued_columns,
      dnn_hidden_units=hidden_units or [100, 70, 50, 25])
```

How do we orchestrate TFX?

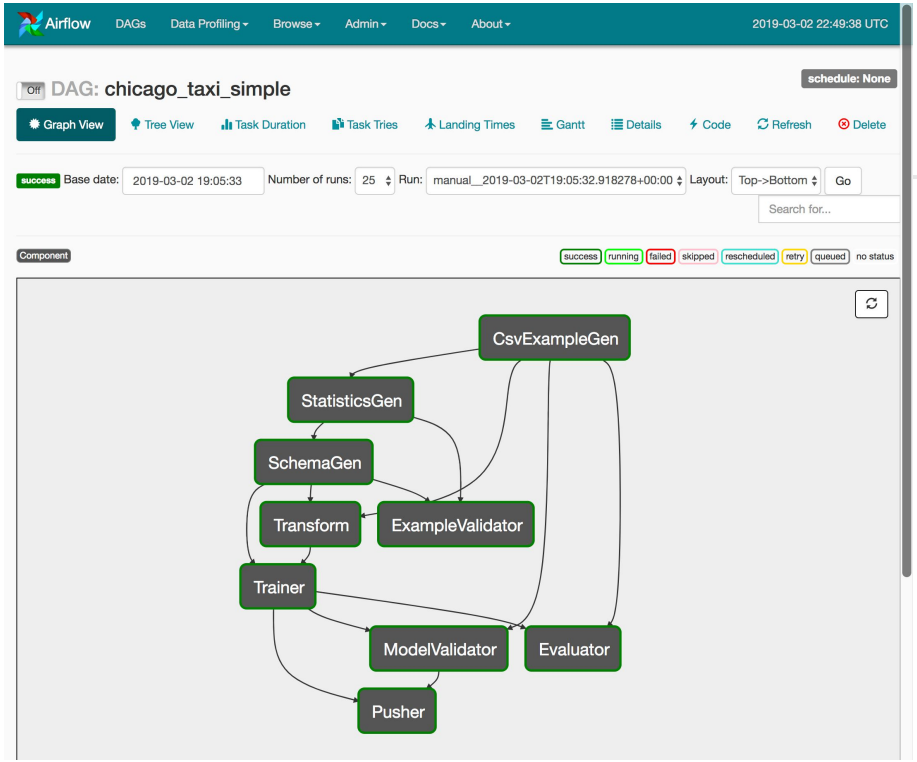


Bring your very own favorite orchestrator

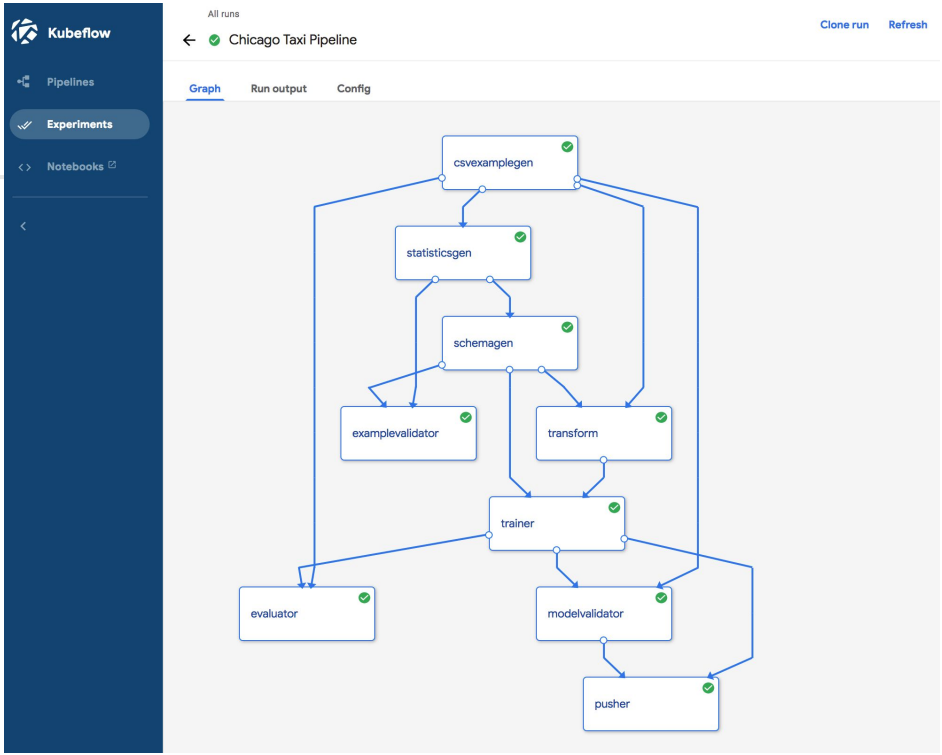


Examples of orchestrated TFX pipelines

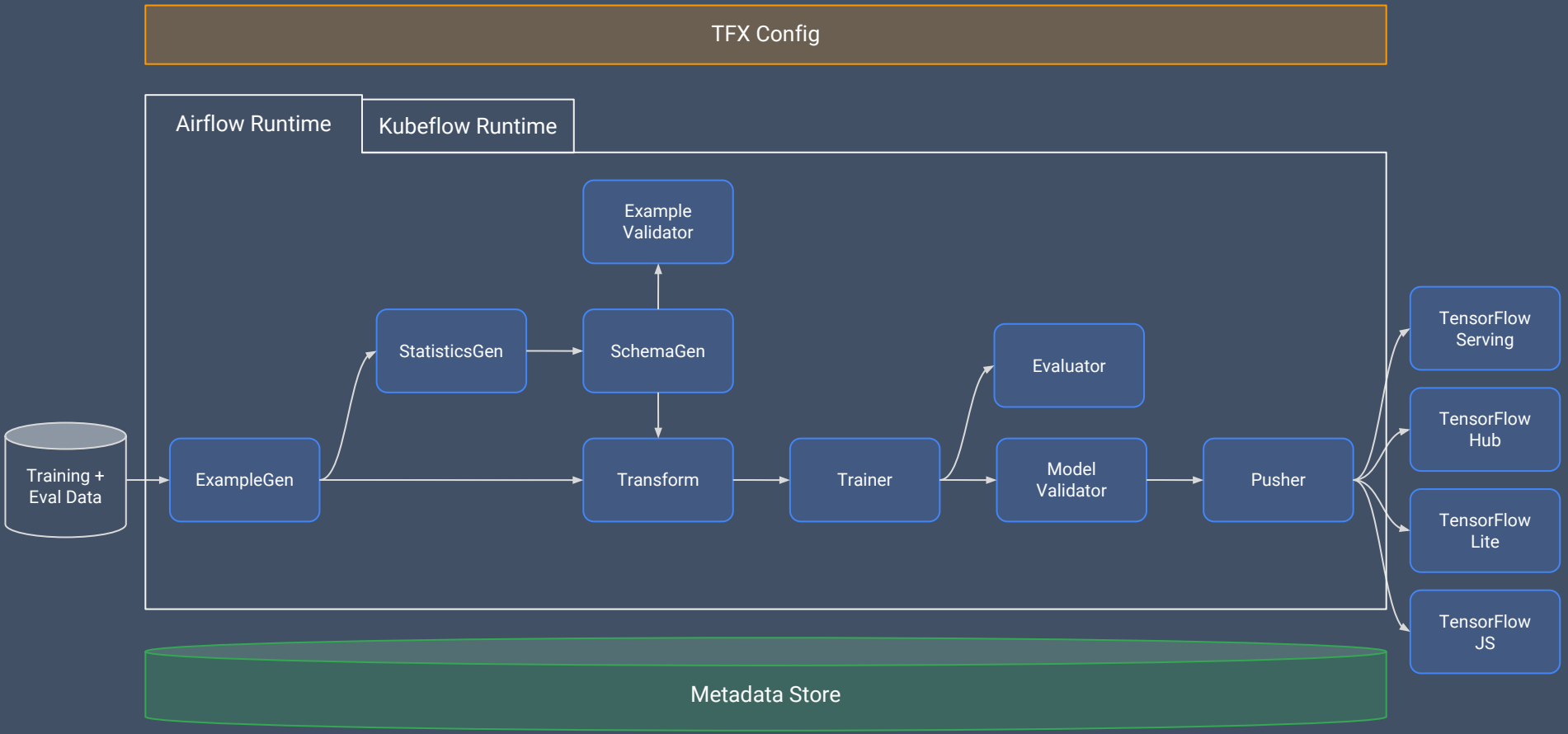
Airflow



Kubeflow Pipelines



TFX: Putting it all together.



TensorFlow

Get started with TensorFlow Extended (TFX)

An End-to-End ML Platform

github.com/tensorflow/tfx

tensorflow.org/tfx

TFX End-to-End Example

Chicago Taxi Cab Dataset

TFX End-to-End Example

Chicago Taxi Cab Dataset

Features

Categorical Features
trip_start_hour
trip_start_day
trip_start_month
pickup/dropoff_census_tract
pickup/dropoff_community_area

Bucket Features
pickup_latitude
pickup_longitude
dropoff_latitude
dropoff_longitude

Vocab Features
payment_type
company

Dense Float Features
trip_miles
fare
trip_seconds

Label = tips > (fare * 20%)

TFX End-to-End Example

Chicago Taxi Cab Dataset

Features

Categorical Features	Bucket Features	Vocab Features	Dense Float Features
trip_start_hour	pickup_latitude	payment_type	trip_miles
trip_start_day	pickup_longitude	company	fare
trip_start_month	dropoff_latitude		trip_seconds
pickup/dropoff_census_tract	dropoff_longitude		
pickup/dropoff_community_area			

Transforms

bucketize	string_to_int	scale_to_z_score

Label = tips > (fare * 20%)

TFX End-to-End Example

Chicago Taxi Cab Dataset

Features

Categorical Features	Bucket Features	Vocab Features	Dense Float Features
trip_start_hour	pickup_latitude	payment_type	trip_miles
trip_start_day	pickup_longitude	company	fare
trip_start_month	dropoff_latitude		trip_seconds
pickup/dropoff_census_tract	dropoff_longitude		
pickup/dropoff_community_area			

Transforms

bucketize | string_to_int | scale_to_z_score

Model (Wide+Deep)

Label = tips > (fare * 20%)

TFX End-to-End Example

Chicago Taxi Cab Dataset

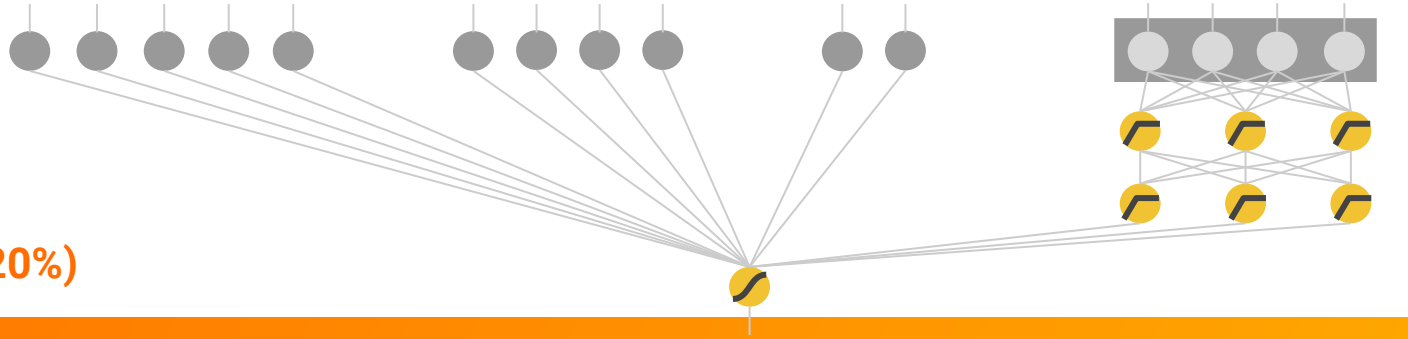
Features

Categorical Features	Bucket Features	Vocab Features	Dense Float Features
trip_start_hour	pickup_latitude	payment_type	trip_miles
trip_start_day	pickup_longitude	company	fare
trip_start_month	dropoff_latitude		trip_seconds
pickup/dropoff_census_tract	dropoff_longitude		
pickup/dropoff_community_area			

Transforms

bucketize | string_to_int | scale_to_z_score

Model (Wide+Deep)



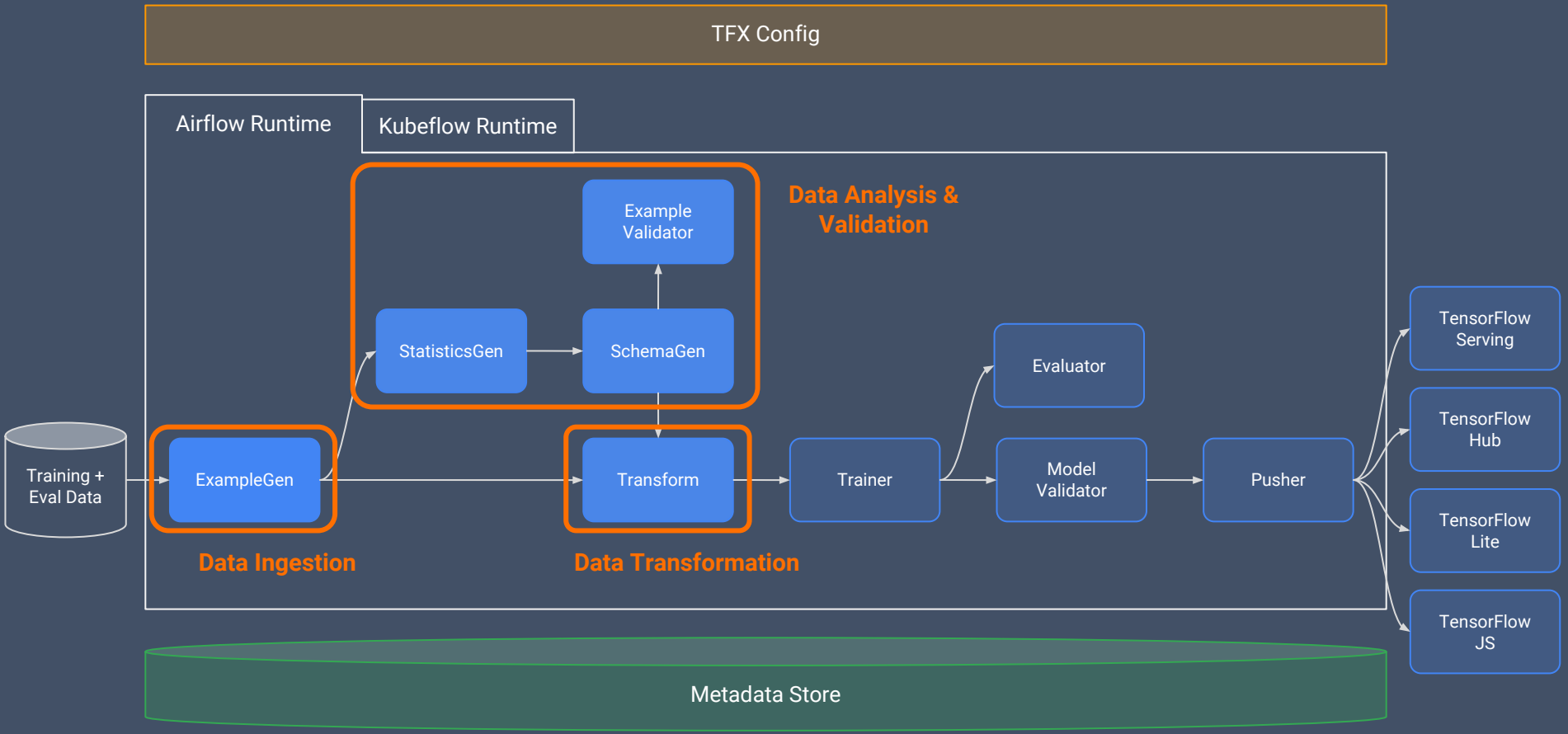
Label = tips > (fare * 20%)

TensorFlow

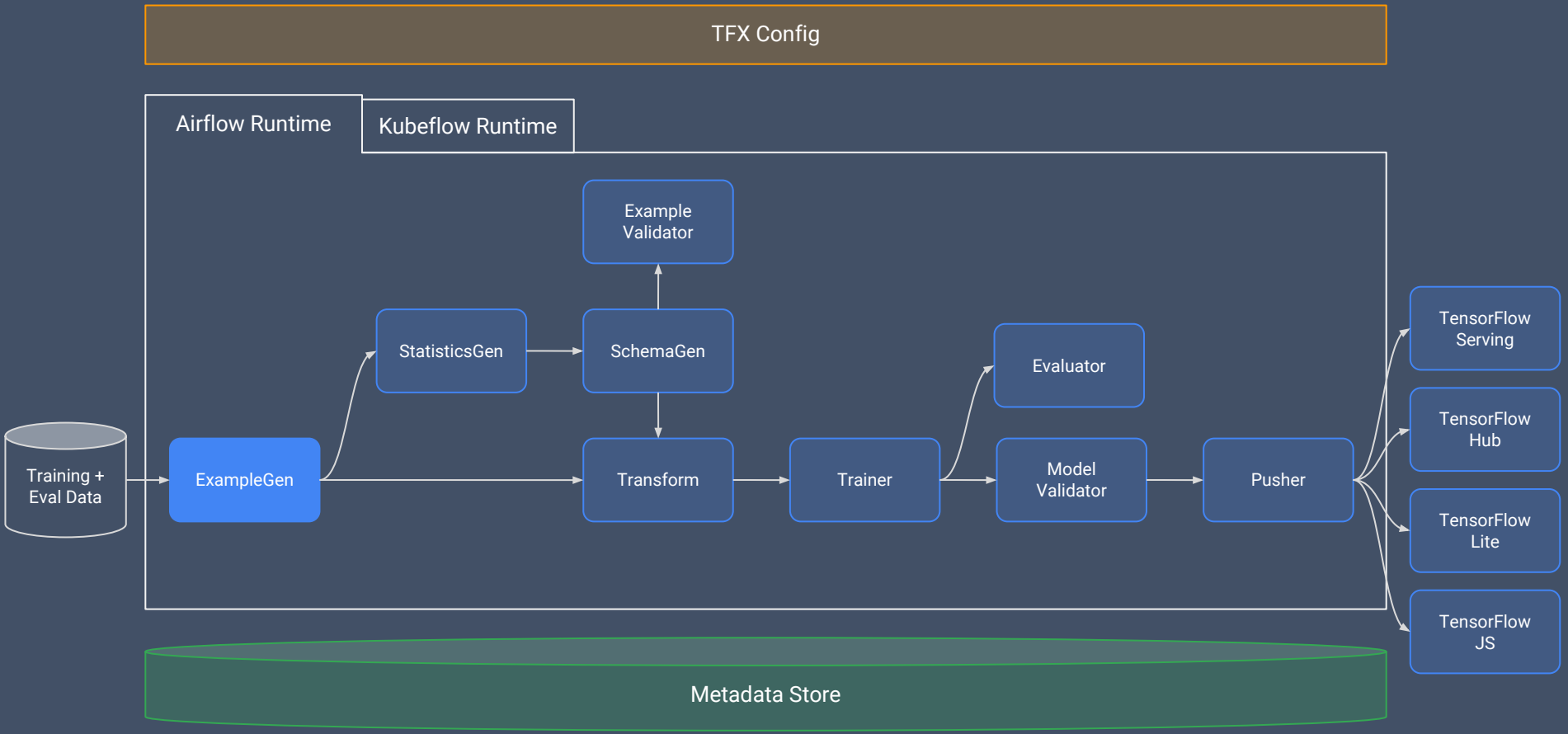
Data Validation and Transformation

Clemens Mewald

Overview

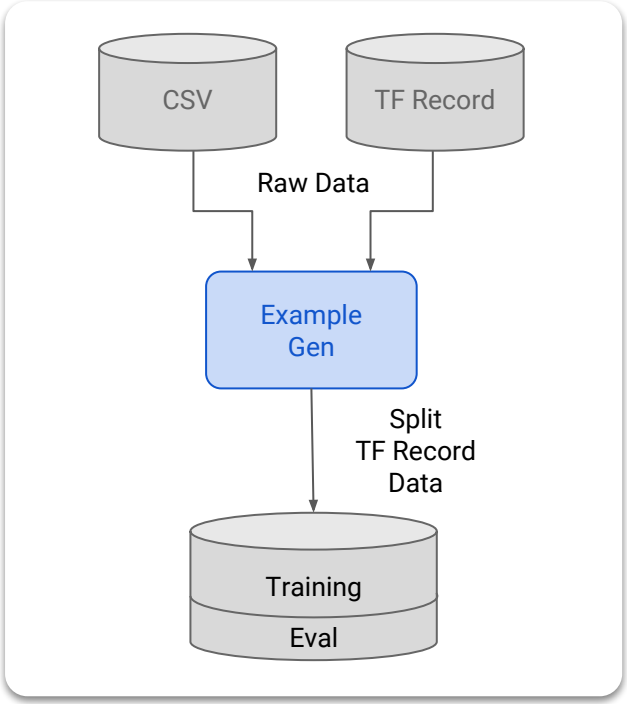

TFX Config
Airflow Runtime
Kubeflow Runtime
Example Validator
Data Analysis & Validation
StatisticsGen
SchemaGen
Evaluator
TensorFlow Serving
Training + Eval Data
ExampleGen
Transform
Trainer
Model Validator
Pusher
TensorFlow Hub
TensorFlow Lite
TensorFlow JS
Data Ingestion
Data Transformation
Metadata Store

ExampleGen


TFX Config
Airflow Runtime
Kubeflow Runtime
Example Validator
StatisticsGen
SchemaGen
Evaluator
Training + Eval Data
ExampleGen
Transform
Trainer
Model Validator
Pusher
TensorFlow Serving
TensorFlow Hub
TensorFlow Lite
TensorFlow JS
Metadata Store

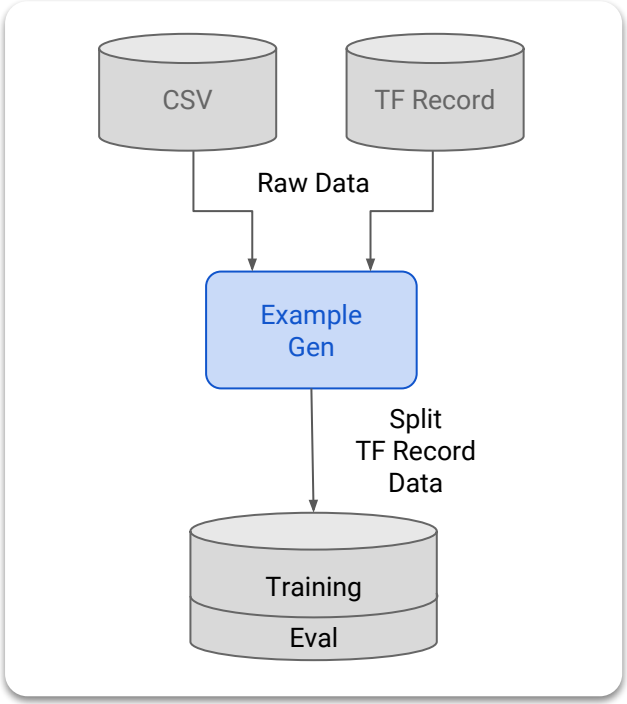
Component: ExampleGen

Inputs and Outputs



Component: ExampleGen

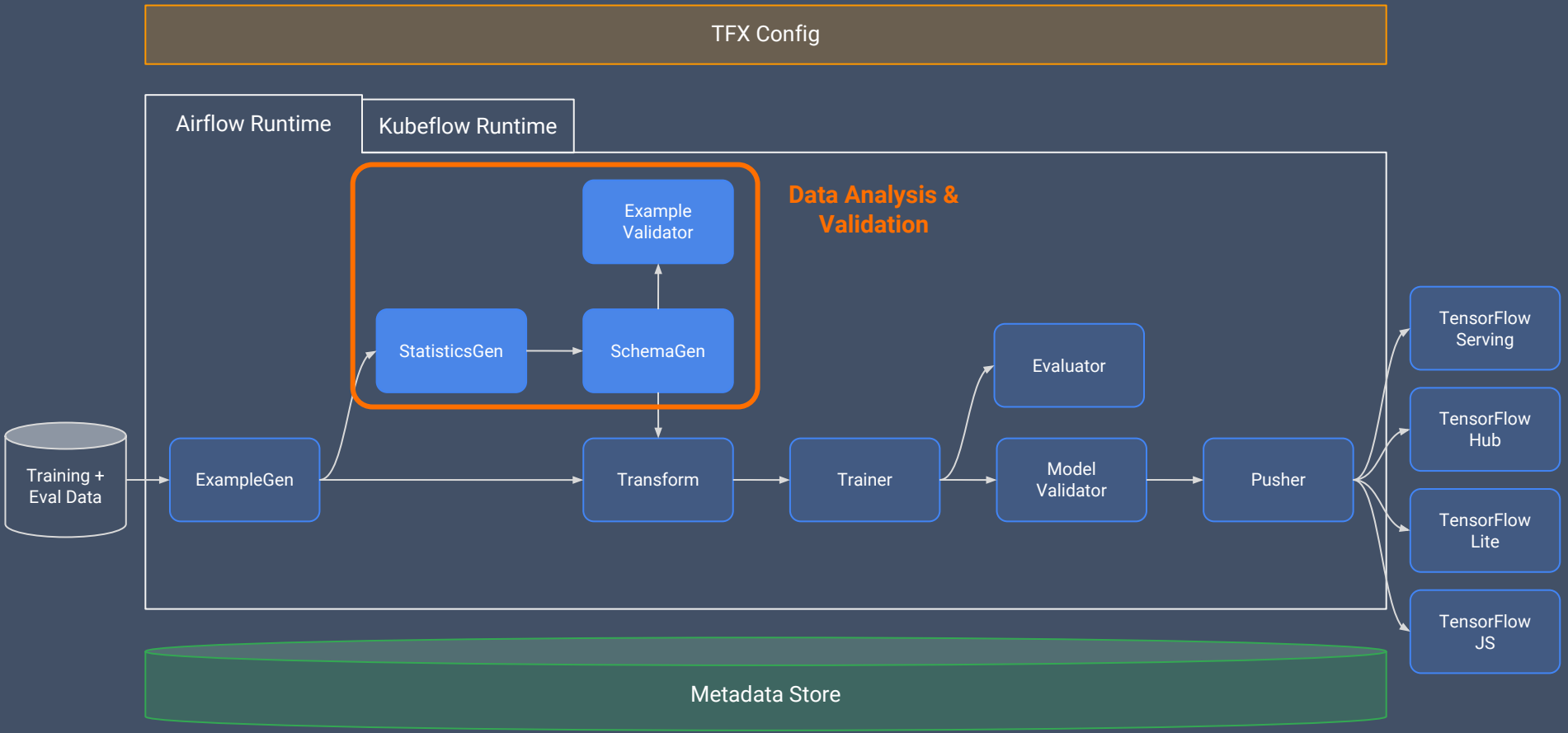
Inputs and Outputs



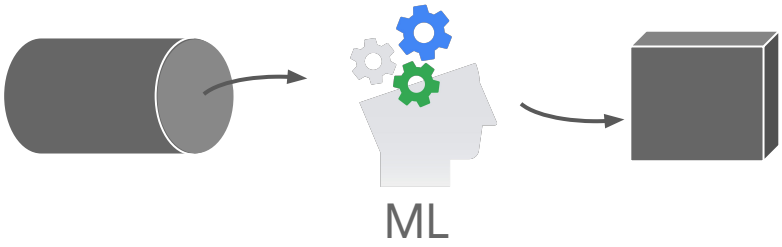
Configuration

```
examples = csv_input(os.path.join(data_root, 'simple'))

example_gen = CsvExampleGen(input_base=examples)
```

Data Analysis & Validation



Why Data Validation is important



Why Data Validation is important

Why Data Validation is important

- Data understanding is important for model understanding

“Why are my tip predictions bad in the morning hours?”

Why Data Validation is important

Data understanding is important for model understanding

- Treat data as you treat code

“What are expected values for payment types?”

Why Data Validation is important

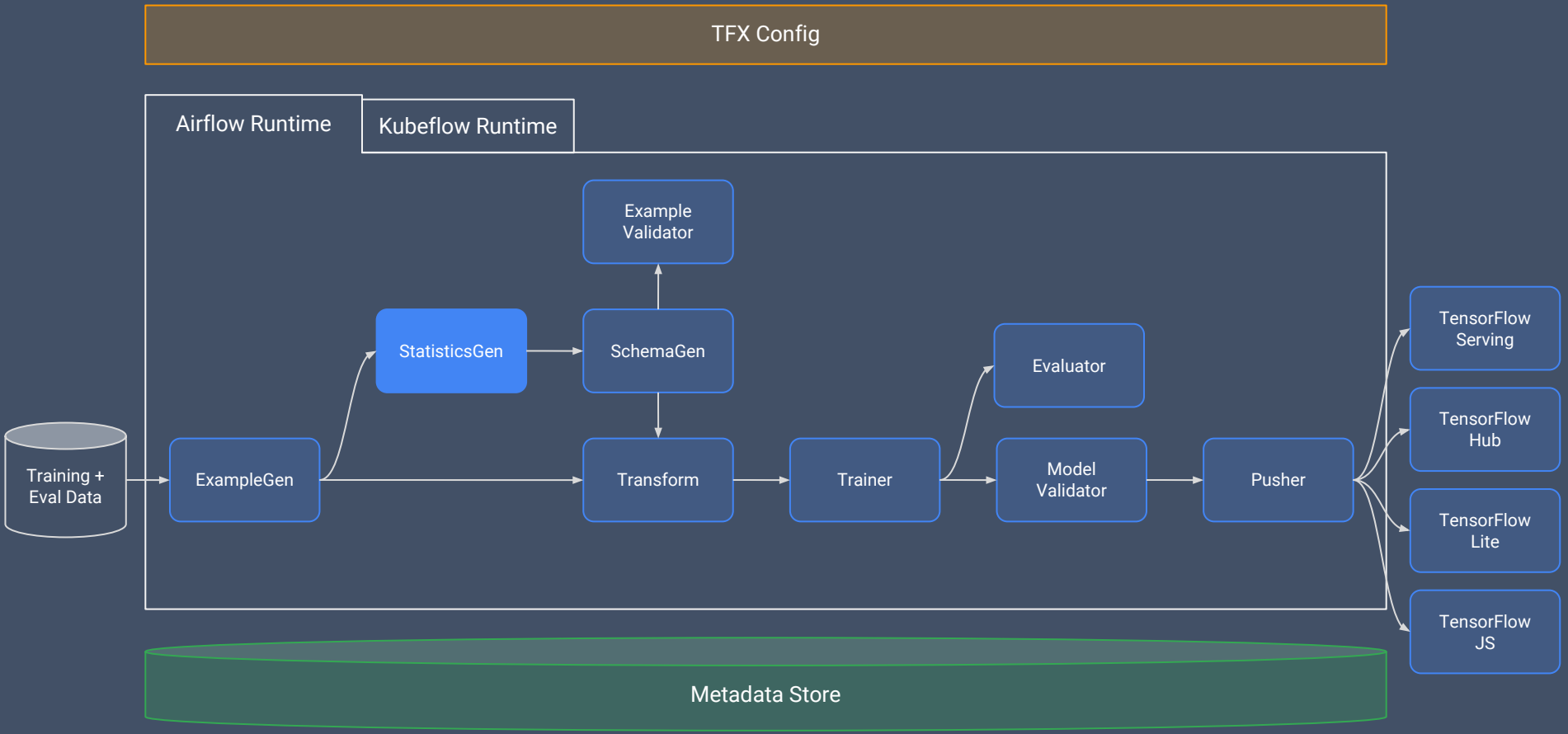
Data understanding is important for model understanding

Treat data as you treat code

Catching errors early is critical

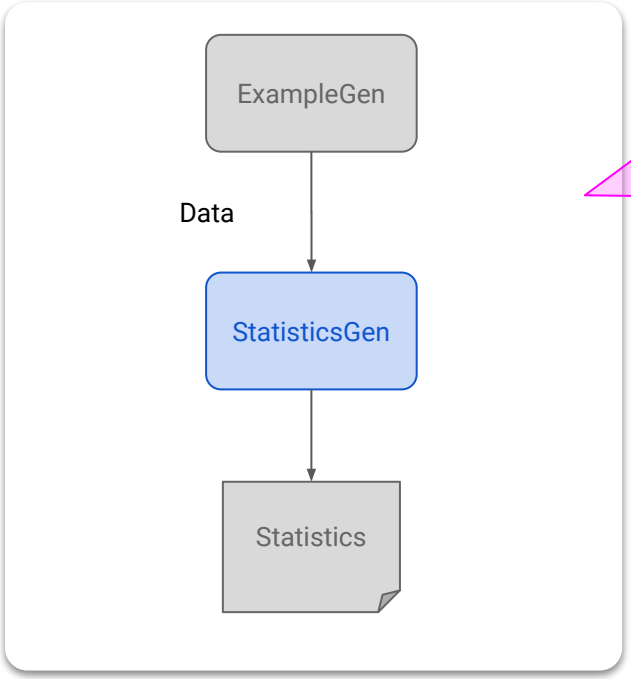
“Is this new taxi company name a typo or a new company?”

StatisticsGen



Component: StatisticsGen

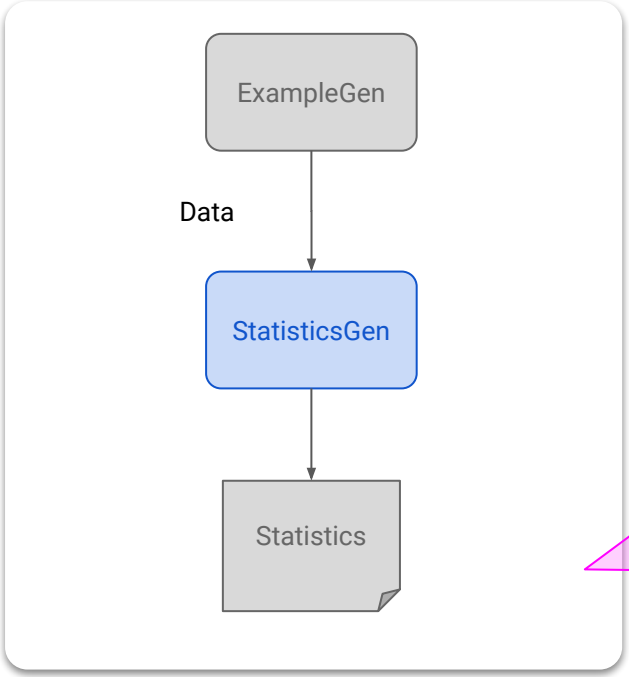
Inputs and Outputs



- Training
- Eval
- Serving logs (for skew detection)

Component: StatisticsGen

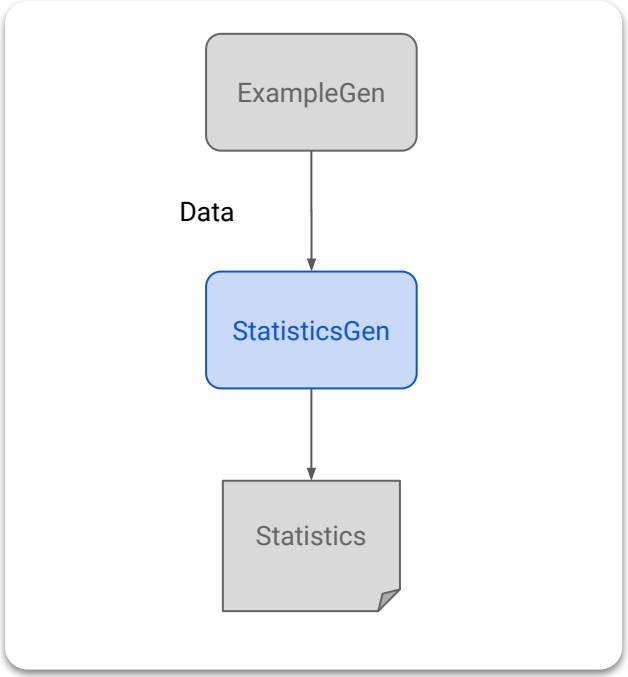
Inputs and Outputs



- Captures shape of data
- Visualization highlights unusual stats
- Overlay helps with comparison

Component: StatisticsGen

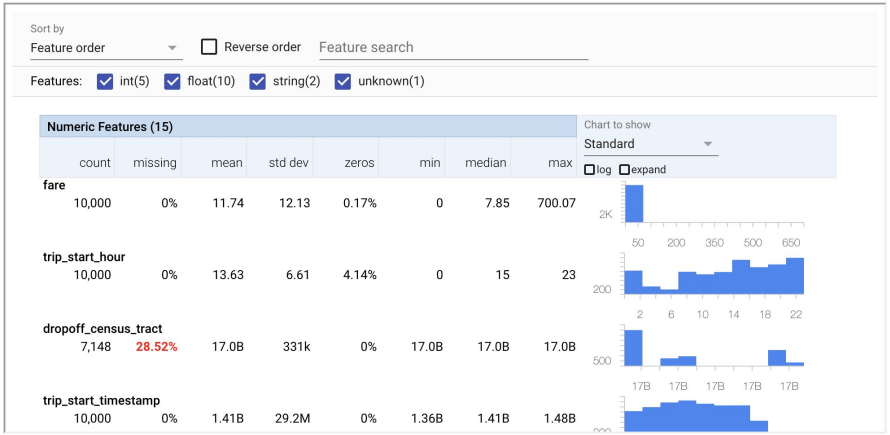
Inputs and Outputs



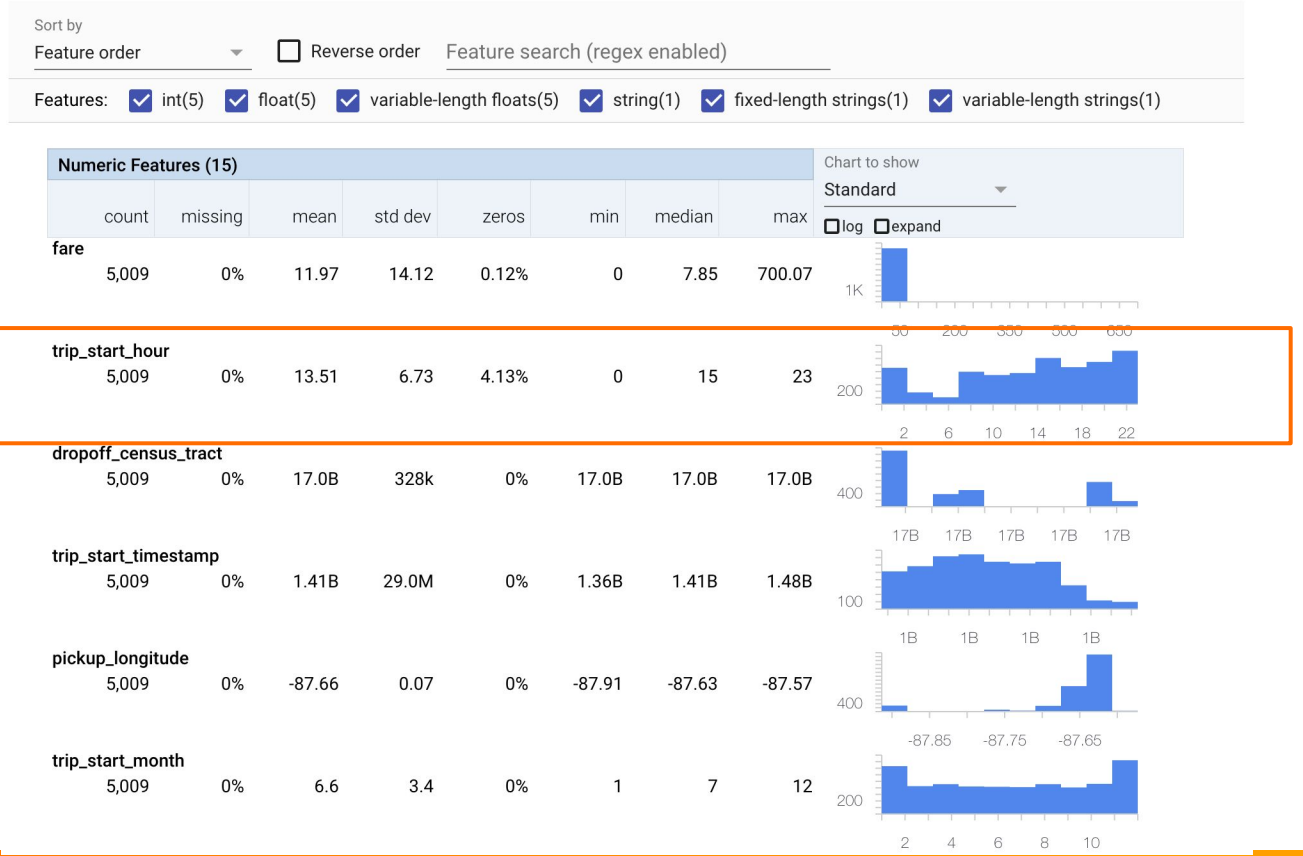
Configuration

```
statistics_gen =
  StatisticsGen(input_data=example_gen.outputs.examples)
```

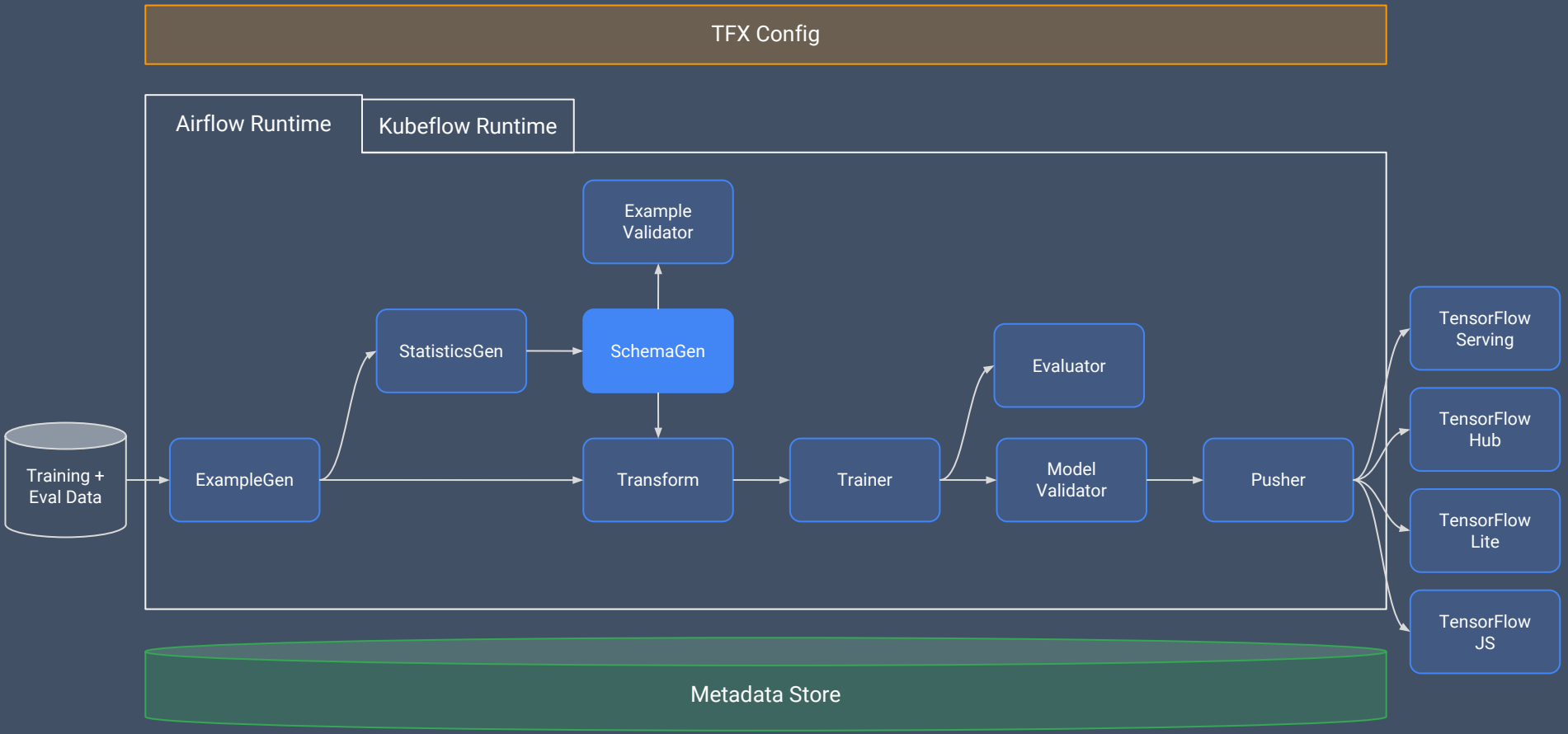
Visualization



Why are my tip predictions bad in the morning hours?

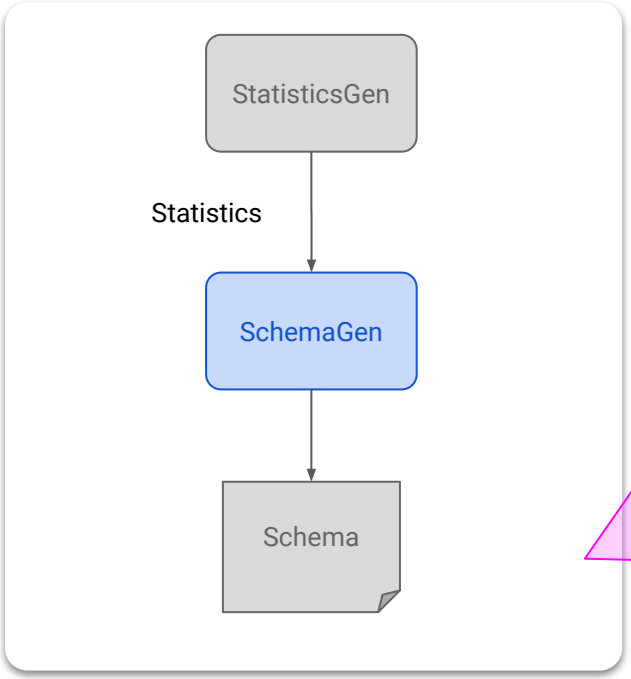


SchemaGen


TFX Config
Airflow Runtime
Kubeflow Runtime
Example Validator
StatisticsGen
SchemaGen
Evaluator
TensorFlow Serving
TensorFlow Hub
TensorFlow Lite
TensorFlow JS
Training + Eval Data
ExampleGen
Transform
Trainer
Model Validator
Pusher
Metadata Store

Component: SchemaGen

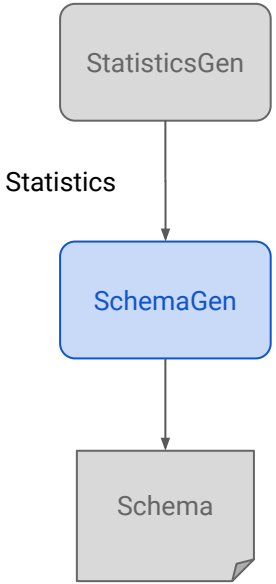
Inputs and Outputs



- High-level description of the data
  - Expected features
  - Expected value domains
  - Expected constraints
  - and much more!
- Codifies expectations of “good” data
- Initially inferred, then user-curated

Component: SchemaGen

Inputs and Outputs



Configuration

```
infer_schema = SchemaGen(stats=statistics_gen.outputs.output)
```

Visualization

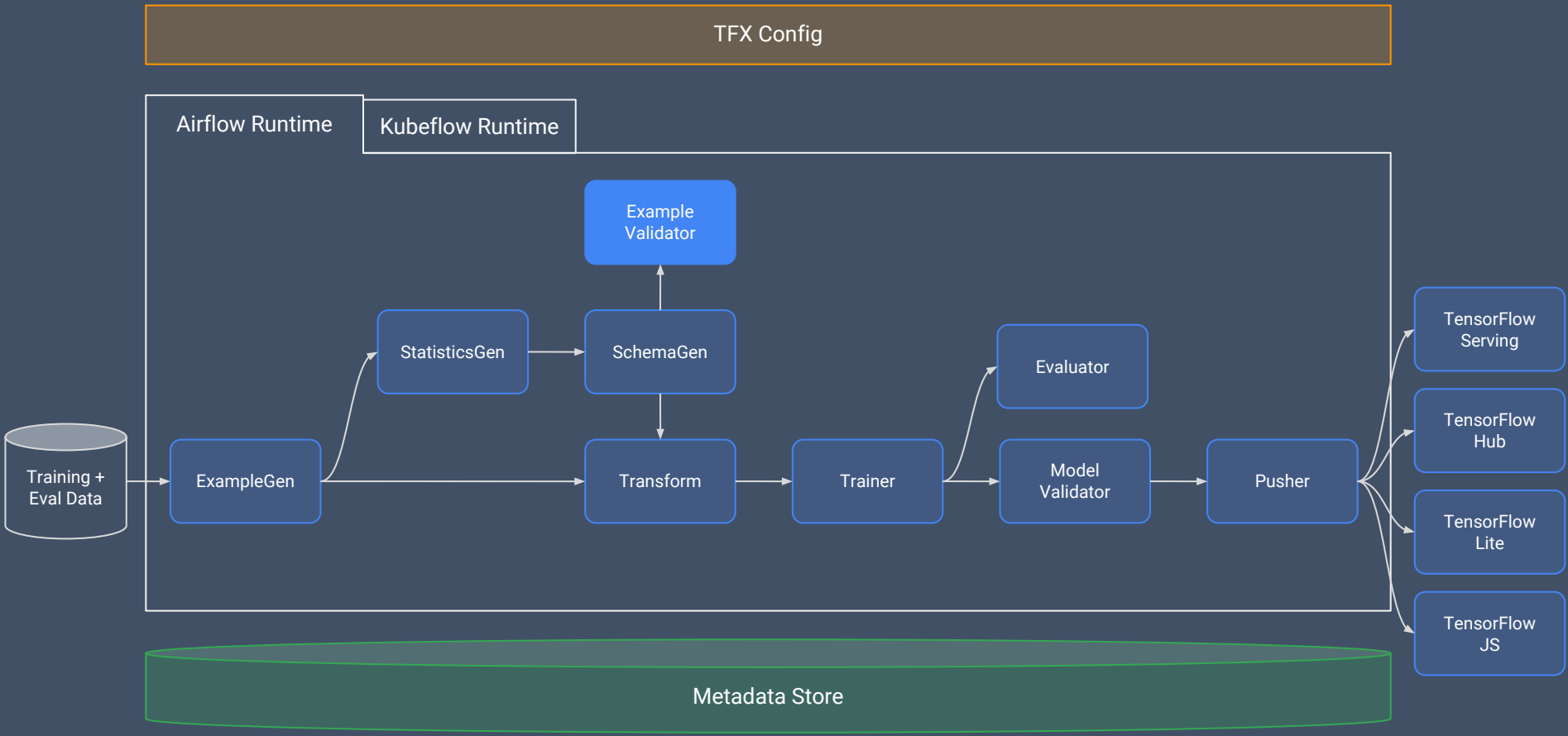
Feature name	Type	Presence	Valency	Domain
'fare'	FLOAT	required	single	-
'trip_start_hour'	INT	required	single	-
'pickup_census_tract'	BYTES	optional		-
'dropoff_census_tract'	FLOAT	optional	single	-
'company'	STRING	optional	single	'company'
'trip_start_timestamp'	INT	required	single	-

What are expected values for payment types?

```
# Infer a schema from the training data stats.
schema = tfdv.infer_schema(statistics=train_stats, infer_feature_shape=False)
tfdv.display_schema(schema=schema)
```

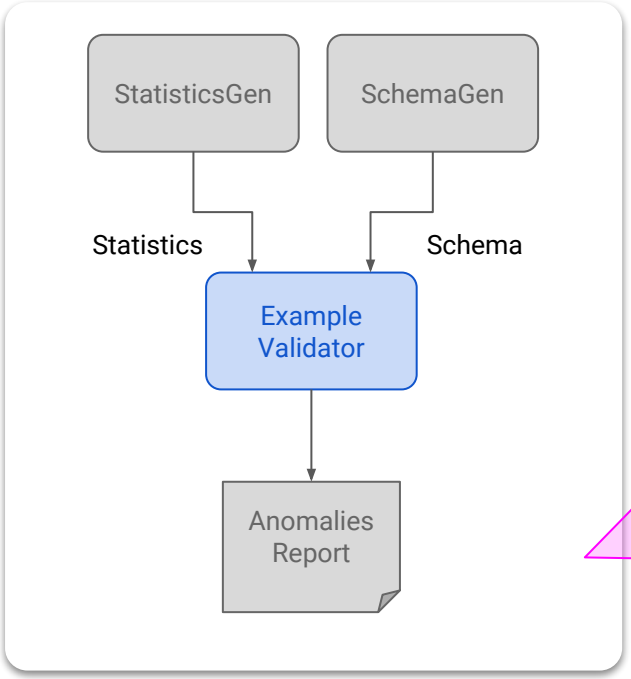
Feature name	Type	Presence	Valency	Domain
'fare'	FLOAT	required	single	-
'trip_start_hour'	INT	required	single	-
'pickup_census_tract'	BYTES	optional		-
'dropoff_census_tract'	FLOAT	optional	single	-
'company'	STRING	optional	single	'company'
'trip_start_timestamp'	INT	required	single	-
'pickup_longitude'	FLOAT	required	single	-
'trip_start_month'	INT	required	single	-
'trip_miles'	FLOAT	required	single	-
'dropoff_longitude'	FLOAT	optional	single	-
'dropoff_community_area'	FLOAT	optional	single	-
'pickup_community_area'	INT	required	single	-
'payment_type'	STRING	required	single	'payment_type'
'trip_seconds'	FLOAT	optional	single	-
'trip_start_day'	INT	required	single	-

ExampleValidator



Component: ExampleValidator

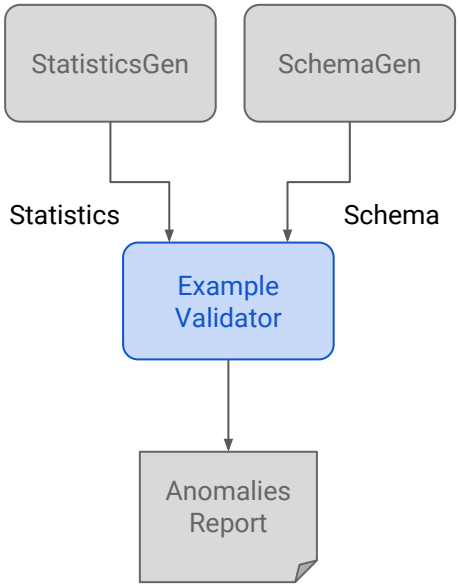
Inputs and Outputs



- Missing features
- Wrong feature valency
- Training/serving skew
- Data distribution drift
- ...

Component: ExampleValidator

Inputs and Outputs



Configuration

```
validate_stats = ExampleValidator(
    stats=statistics_gen.outputs.output,
    schema=infer_schema.outputs.output)
```

Visualization

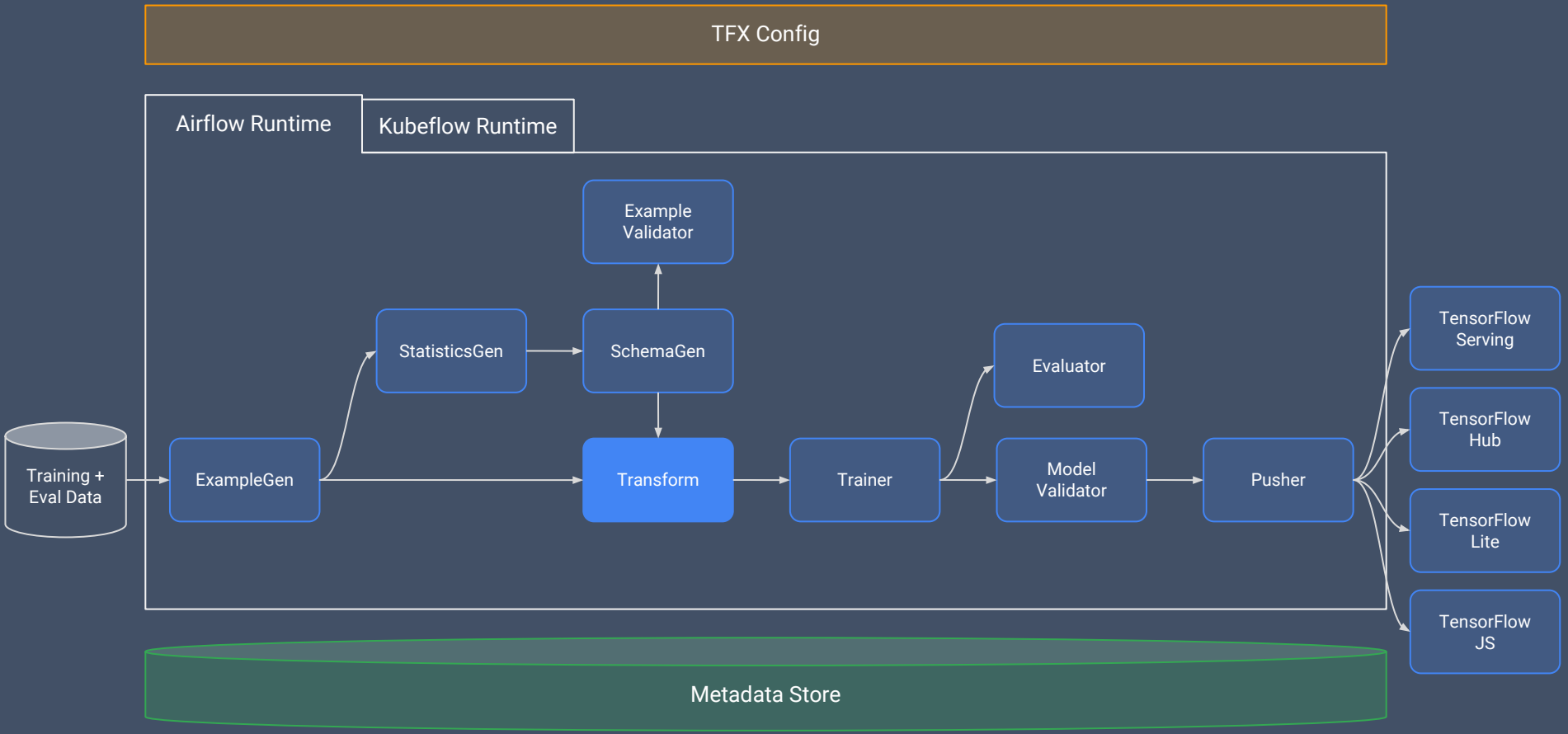
Feature name	Anomaly short description	Anomaly long description
'payment_type'	Unexpected string values	Examples contain values missing from the schema: Prcard (<1%).
'company'	Unexpected string values	Examples contain values missing from the schema: 2092 - 61288 Sbeih company (<1%), 2192 - 73487 Zeymane 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%).

Is this new taxi company name a typo or a new company?

```
# Check eval data for errors by validating the eval data stats using the previously inferred schema.
anomalies = tfdv.validate_statistics(statistics=eval_stats, schema=schema)
tfdv.display_anomalies(anomalies)
```

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%).

Transform


TFX Config
Airflow Runtime
Kubeflow Runtime
Example Validator
StatisticsGen
SchemaGen
Evaluator
Training + Eval Data
ExampleGen
Transform
Trainer
Model Validator
Pusher
TensorFlow Serving
TensorFlow Hub
TensorFlow Lite
TensorFlow JS
Metadata Store

Recap: End-to-End Example

Chicago Taxi Cab Dataset

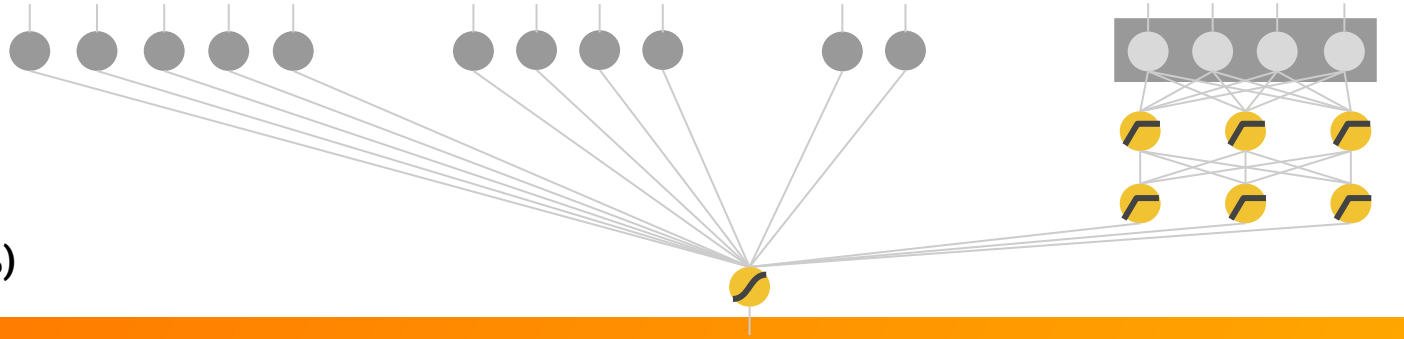
Features

Categorical Features	Bucket Features	Vocab Features	Dense Float Features
trip_start_hour	pickup_latitude	payment_type	trip_miles
trip_start_day	pickup_longitude	company	fare
trip_start_month	dropoff_latitude		trip_seconds
pickup/dropoff_census_tract	dropoff_longitude		
pickup/dropoff_community_area			

Transforms

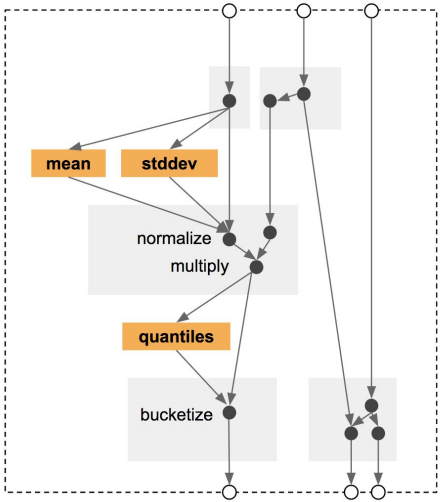
bucketize | string_to_int | scale_to_z_score

Model (Wide+Deep)

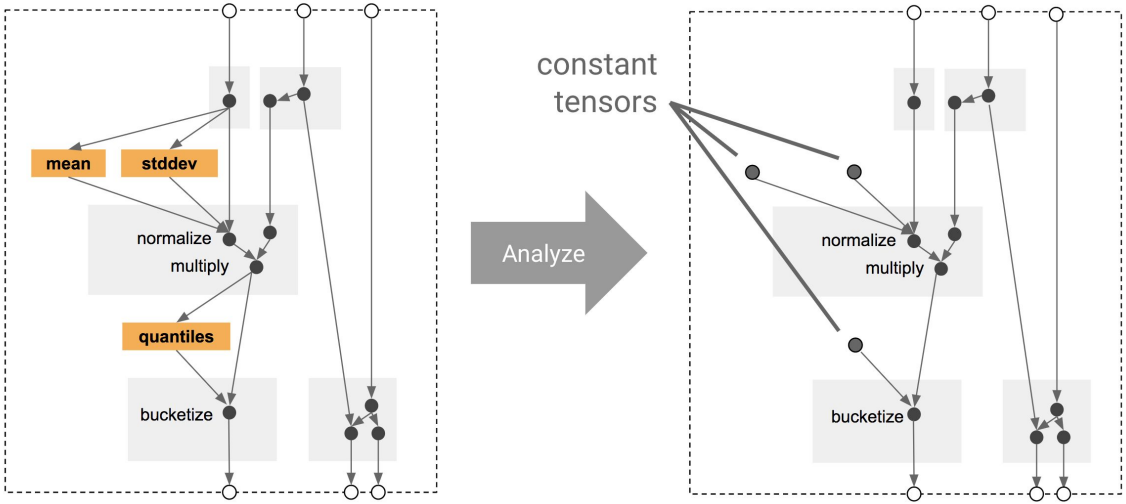


Label = tips > (fare * 20%)

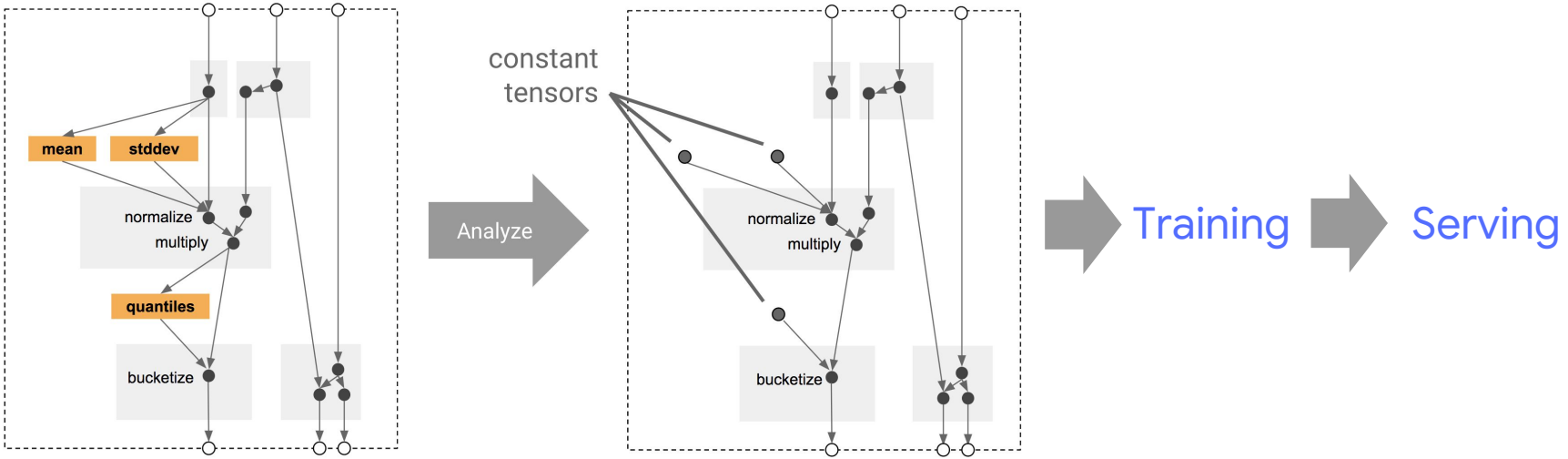
Using tf.Transform for feature transformations.



Using tf.Transform for feature transformations.

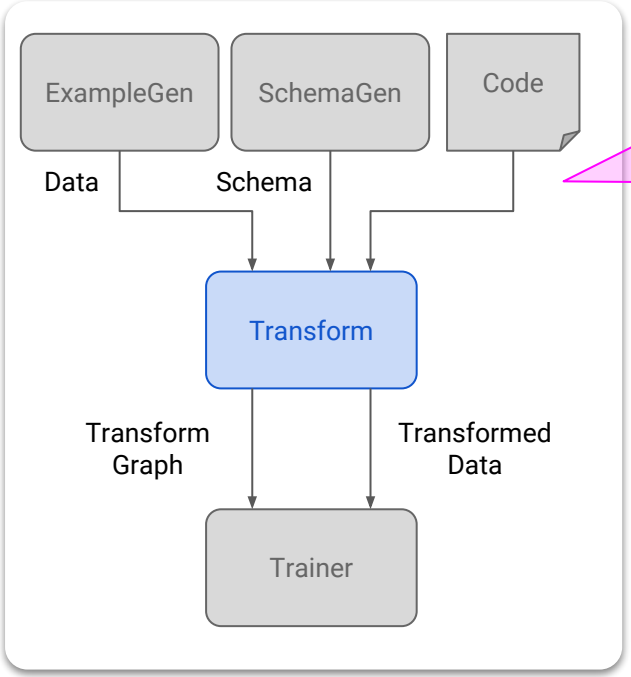


Using tf.Transform for feature transformations.



Component: Transform

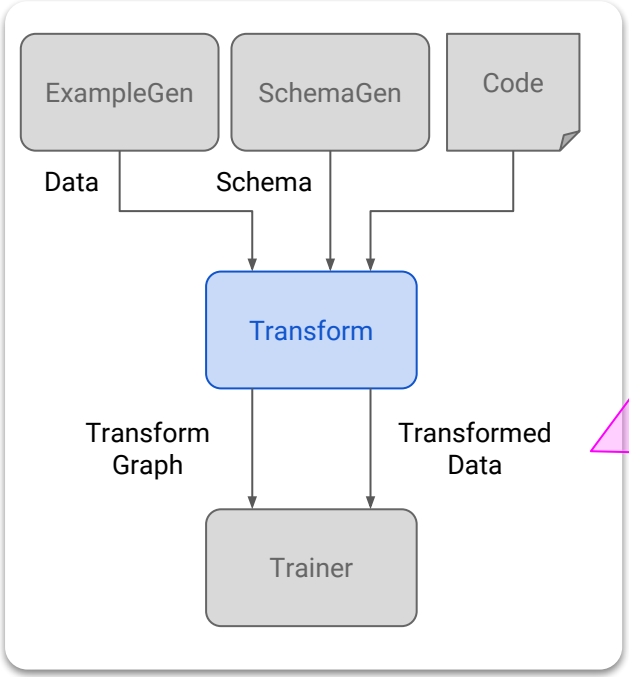
Inputs and Outputs



- User-provided transform code (TF Transform)
- Schema for parsing

Component: Transform

Inputs and Outputs



Transform Graph

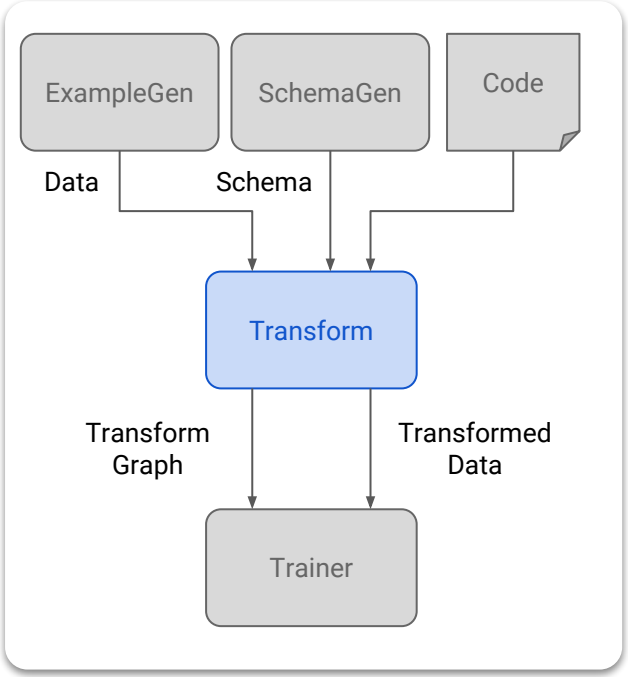
- Applied at training time
- Embedded in serving graph

(Optional) Transformed Data

- For performance optimization

Component: Transform

Inputs and Outputs



Configuration

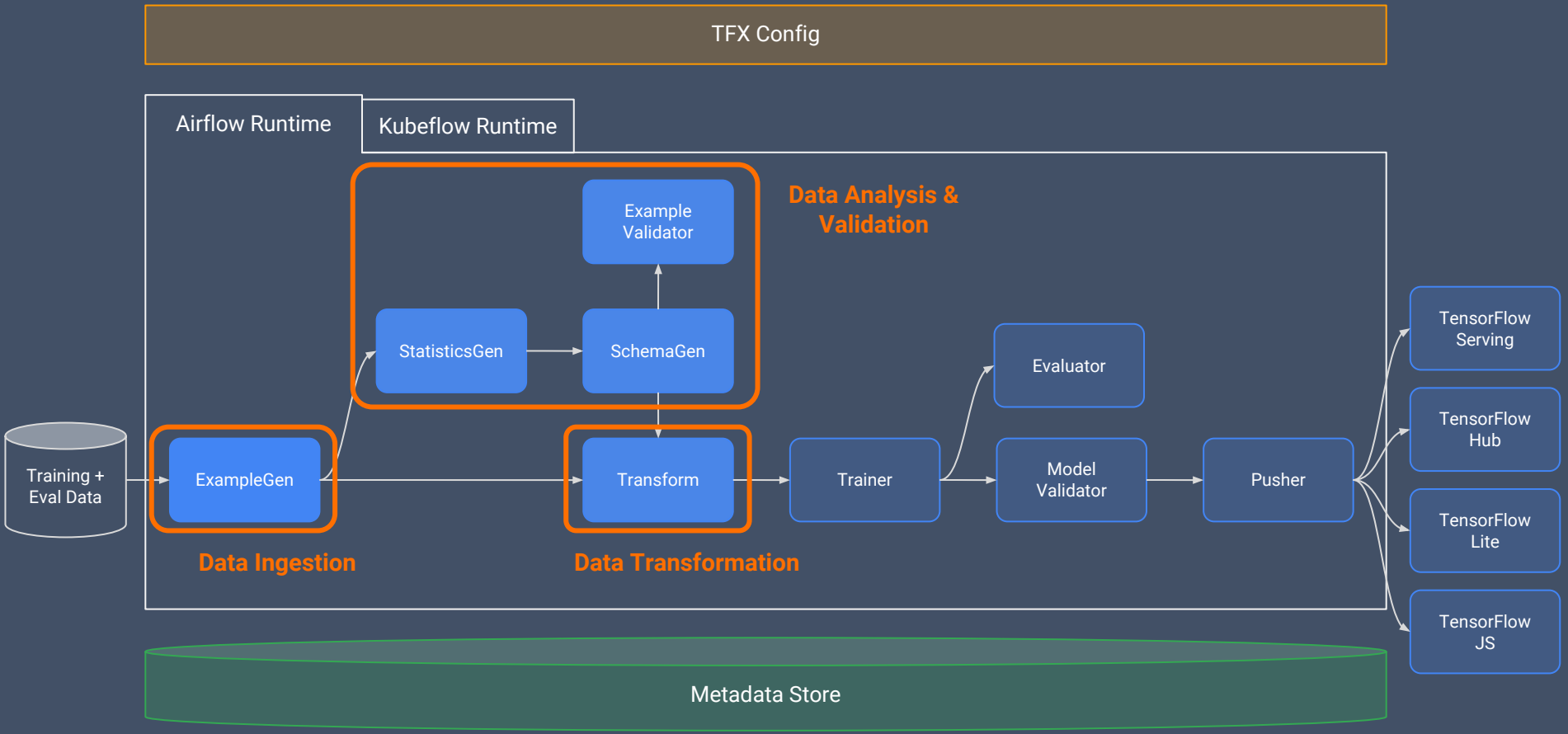
```
transform = Transform(
    input_data=example_gen.outputs.examples,
    schema=infer_schema.outputs.output,
    module_file=taxi_module_file)
```

Code

```
for key in _DENSE_FLOAT_FEATURE_KEYS:
    outputs[_transformed_name(key)] = transform.scale_to_z_score(
        _fill_in_missing(inputs[key]))
# ...

outputs[_transformed_name(_LABEL_KEY)] = tf.where(
      tf.is_nan(taxi_fare),
      tf.cast(tf.zeros_like(taxi_fare), tf.int64),
      # Test if the tip was > 20% of the fare.
      tf.cast(
          tf.greater(tips, tf.multiply(taxi_fare, tf.constant(0.2))), tf.int64))

# ...
```

```
def preprocessing_fn(inputs):
  ...
     for key in taxi.DENSE_FLOAT_FEATURE_KEYS:
      outputs[key] = transform.scale_to_z_score(inputs[key])

    for key in taxi.VOCAB_FEATURE_KEYS:
      outputs[key] = transform.string_to_int(inputs[key],
          top_k=taxi.VOCAB_SIZE,
          num_oov_buckets=taxi.OOV_SIZE)

    for key in taxi.BUCKET_FEATURE_KEYS:
      outputs[key] = transform.bucketize(inputs[key],
          taxi.FEATURE_BUCKET_COUNT)
  ...
```

Overview

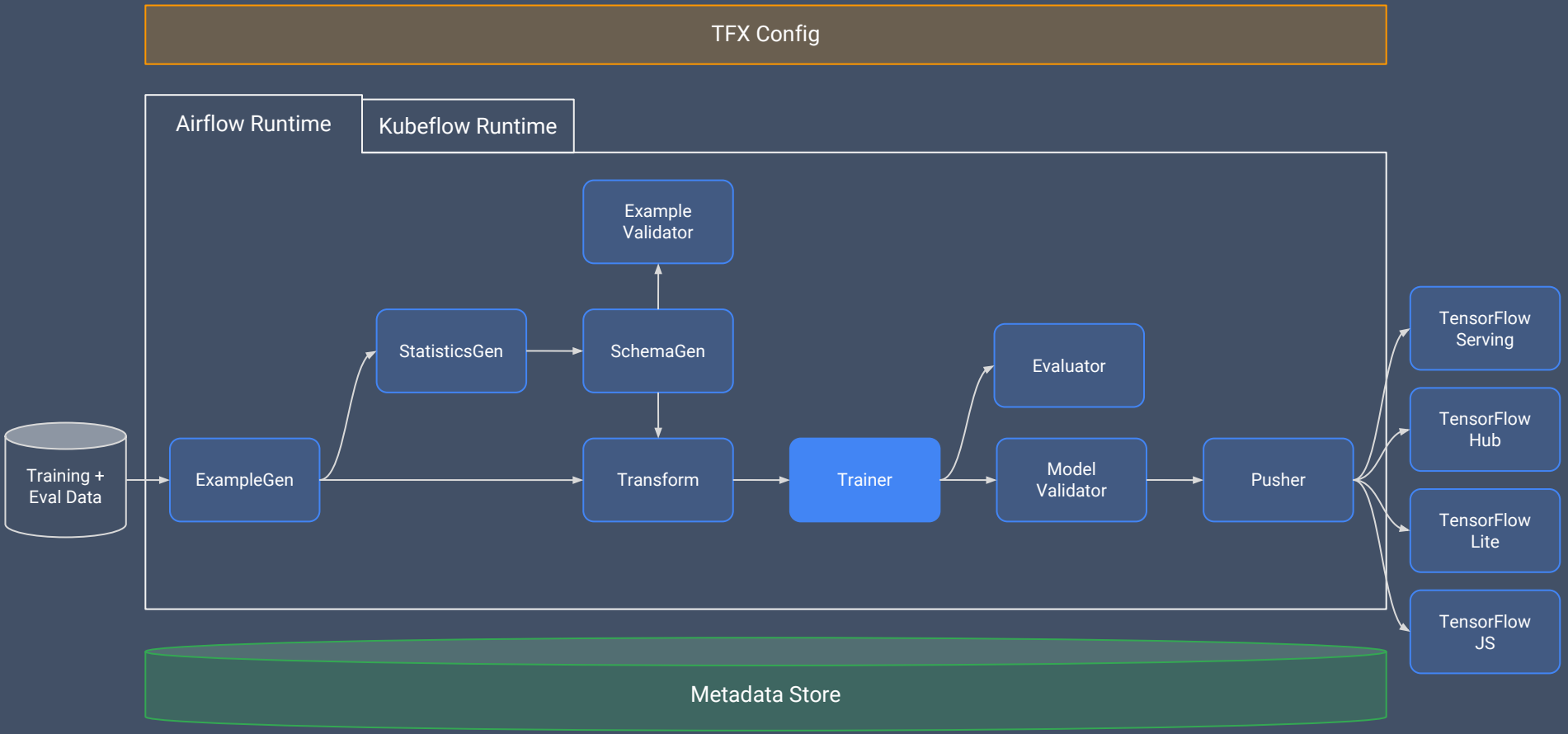

TFX Config
Airflow Runtime
Kubeflow Runtime
Example Validator
Data Analysis & Validation
StatisticsGen
SchemaGen
Evaluator
TensorFlow Serving
Training + Eval Data
ExampleGen
Transform
Trainer
Model Validator
Pusher
TensorFlow Hub
TensorFlow Lite
TensorFlow JS
Data Ingestion
Data Transformation
Metadata Store

TensorFlow

Training

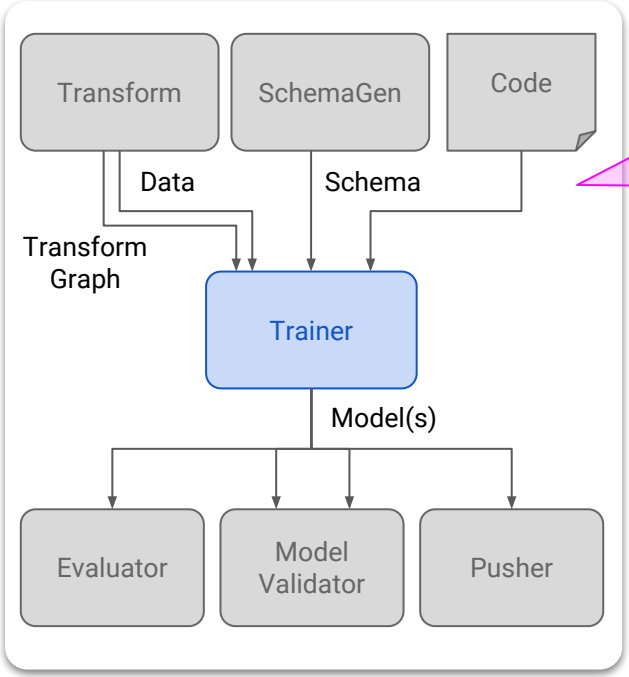
Clemens Mewald

Trainer


TFX Config
Airflow Runtime
Kubeflow Runtime
Example Validator
StatisticsGen
SchemaGen
Evaluator
TensorFlow Serving
Training + Eval Data
ExampleGen
Transform
Trainer
Model Validator
Pusher
TensorFlow Hub
TensorFlow Lite
TensorFlow JS
Metadata Store

Component: Trainer

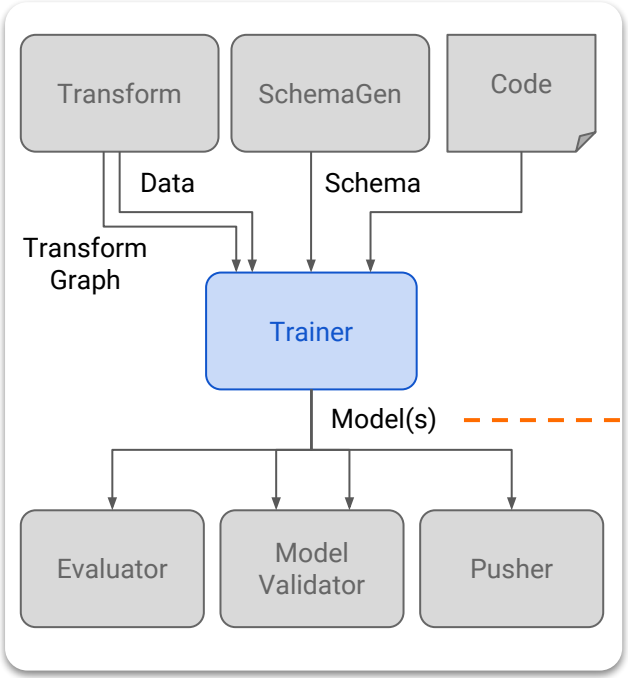
Inputs and Outputs



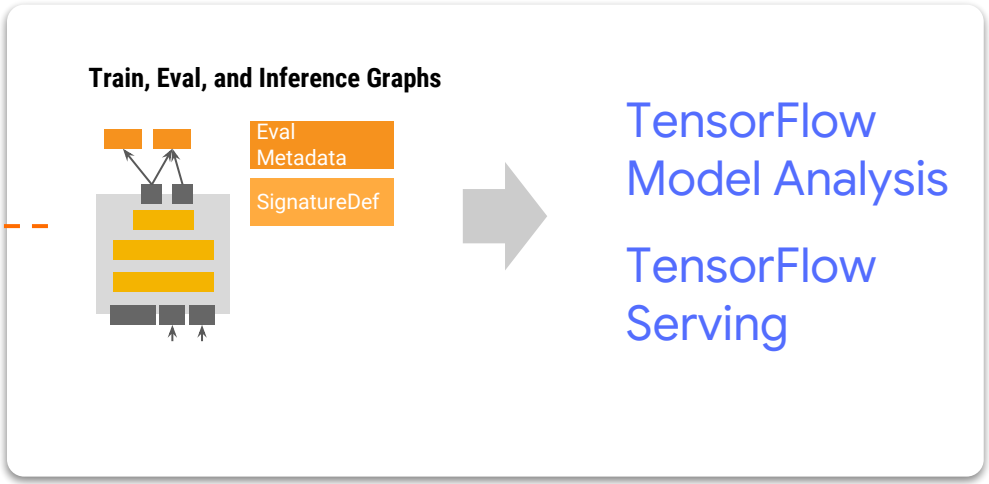
- User-provided training code (TensorFlow)
- Optionally, transformed data

Component: Trainer

Inputs and Outputs

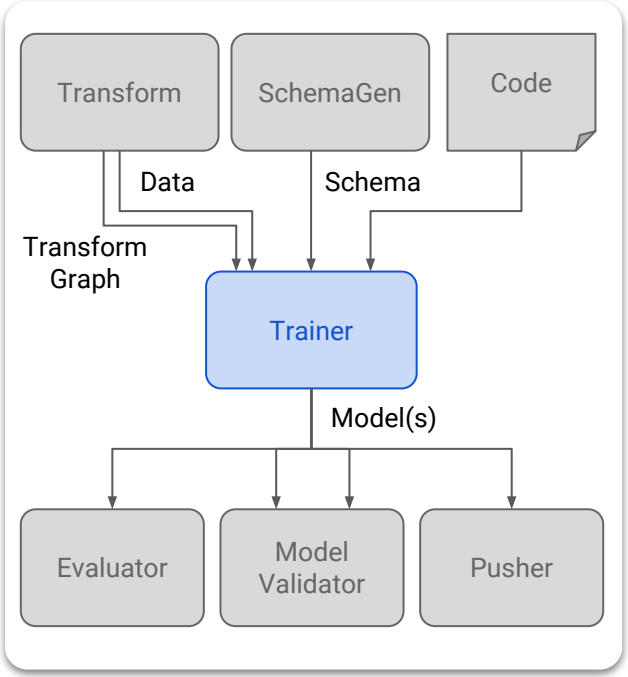


Highlight: SavedModel Format



Component: Trainer

Inputs and Outputs



Configuration

```
trainer = Trainer(
    module_file=taxi_module_file,
    transformed_examples=transform.outputs.transformed_examples,
    schema=infer_schema.outputs.output,
    transform_output=transform.outputs.transform_output,
    train_steps=10000,
    eval_steps=5000,
    warm_starting=True)
```

Code: Just TensorFlow :)

```
def train_and_maybe_evaluate(hparams):
  schema = taxi.read_schema(hparams.schema_file)

  train_input = lambda: model.input_fn(...)
  eval_input = lambda: model.input_fn(...)
  serving_receiver_fn = lambda: model.example_serving_receiver_fn(...)

  train_spec = tf.estimator.TrainSpec(...)
  eval_spec = tf.estimator.EvalSpec(...)

  exporter = tf.estimator.FinalExporter('chicago-taxi', serving_receiver_fn)

  run_config = tf.estimator.RunConfig(
      save_checkpoints_steps=999, keep_checkpoint_max=1)

  estimator = model.build_estimator(...)

  tf.estimator.train_and_evaluate(estimator, train_spec, eval_spec)

  return estimator
```

```
def build_estimator(tf_transform_dir, config, hidden_units=None):

  # receive Schema and Transform metadata
  metadata_dir = os.path.join(tf_transform_dir,
                              transform_fn_io.TRANSFORMED_METADATA_DIR)
  transformed_metadata = metadata_io.read_metadata(metadata_dir)
  transformed_feature_spec = transformed_metadata.schema.as_feature_spec()

  transformed_feature_spec.pop(taxi.transformed_name(taxi.LABEL_KEY))

  real_valued_columns = [...]
  categorical_columns = [...]

  return tf.estimator.DNNLinearCombinedClassifier(
      config=config,
      linear_feature_columns=categorical_columns,
      dnn_feature_columns=real_valued_columns,
      dnn_hidden_units=hidden_units or [100, 70, 50, 25])
```

Keras: TF 2.0

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(512, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10, activation='softmax')
])
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```

Going big: tf.distribute.Strategy

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Dense(64, input_shape=[10]),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(10, activation='softmax')])

model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

Going big: Multi-GPU

```
strategy = tf.distribute.MirroredStrategy()

with strategy.scope():
  model = tf.keras.models.Sequential([
      tf.keras.layers.Dense(64, input_shape=[10]),
      tf.keras.layers.Dense(64, activation='relu'),
      tf.keras.layers.Dense(10, activation='softmax')])

  model.compile(optimizer='adam',
                loss='categorical_crossentropy',
                metrics=['accuracy'])
```

Coming soon: Multi-node synchronous

```
strategy = tf.distribute.experimental.MultiWorkerMirroredStrategy()

with strategy.scope():
  model = tf.keras.models.Sequential([
      tf.keras.layers.Dense(64, input_shape=[10]),
      tf.keras.layers.Dense(64, activation='relu'),
      tf.keras.layers.Dense(10, activation='softmax')])

  model.compile(optimizer='adam',
                loss='categorical_crossentropy',
                metrics=['accuracy'])
```

To SavedModel and beyond

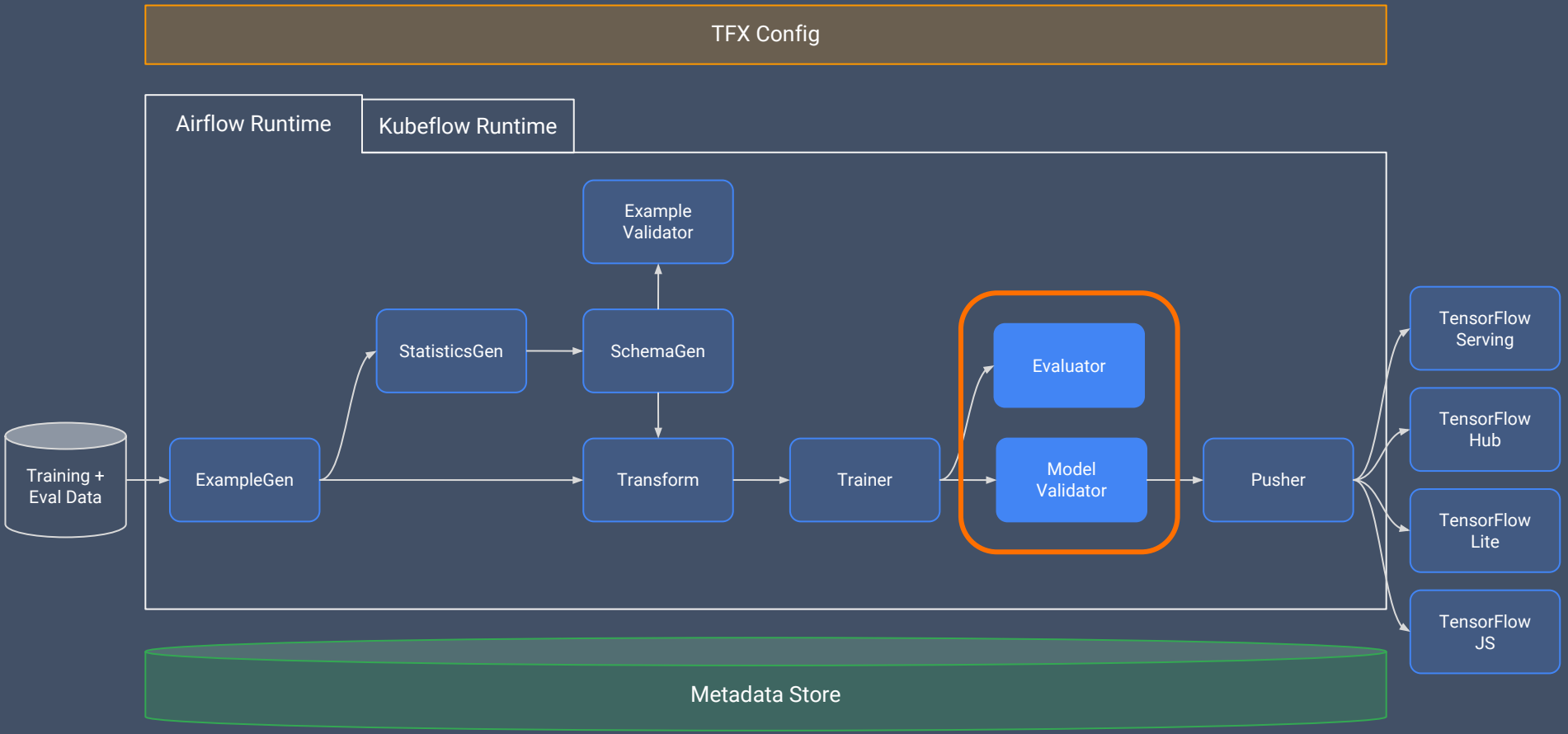
```
saved_model_path = tf.keras.experimental.export_saved_model(
    model, '/path/to/model')

new_model = tf.keras.experimental.load_from_saved_model(
    saved_model_path)

new_model.summary()
```

TensorFlow

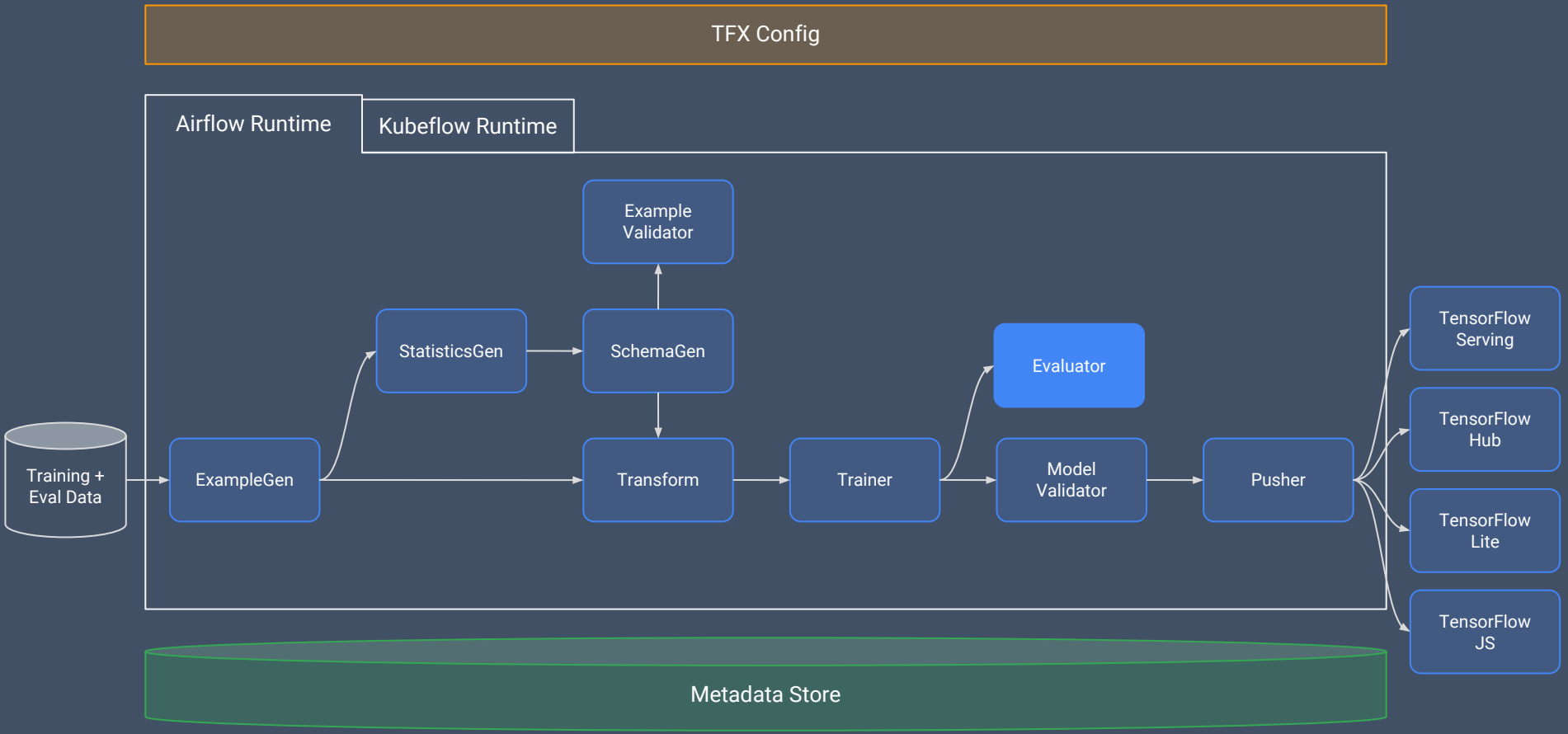
Model Evaluation and Analysis

Clemens Mewald

Model Analysis & Validation



Evaluator

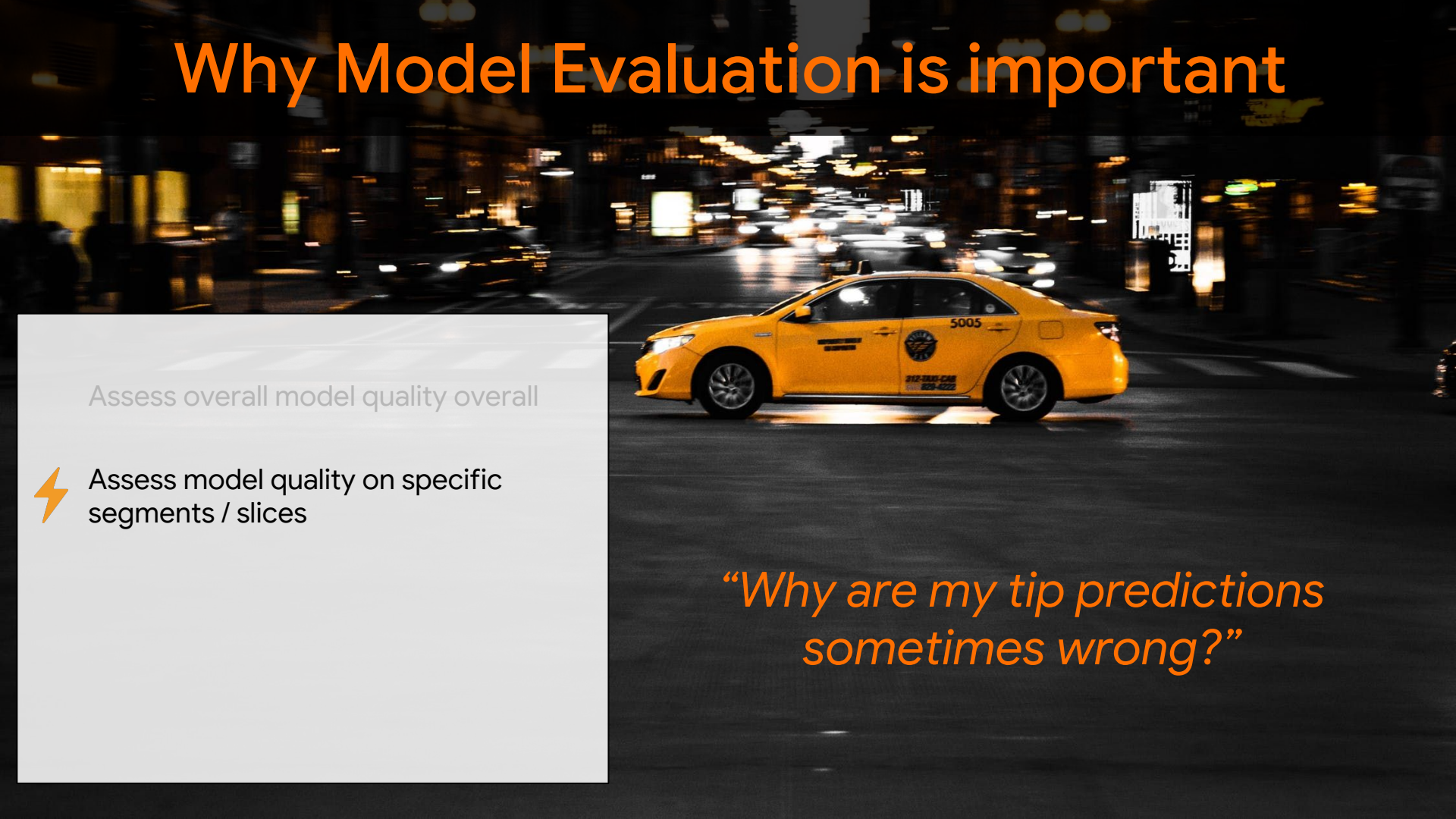

TFX Config
Airflow Runtime
Kubeflow Runtime
Example Validator
StatisticsGen
SchemaGen
Evaluator
Training + Eval Data
ExampleGen
Transform
Trainer
Model Validator
Pusher
TensorFlow Serving
TensorFlow Hub
TensorFlow Lite
TensorFlow JS
Metadata Store

Why Model Evaluation is important

- Assess overall model quality overall

“How well can I predict trips that result in tips > 20%?”

Why Model Evaluation is important



Assess overall model quality overall

- Assess model quality on specific segments / slices

“Why are my tip predictions sometimes wrong?”

Why Model Evaluation is important

Assess overall model quality overall

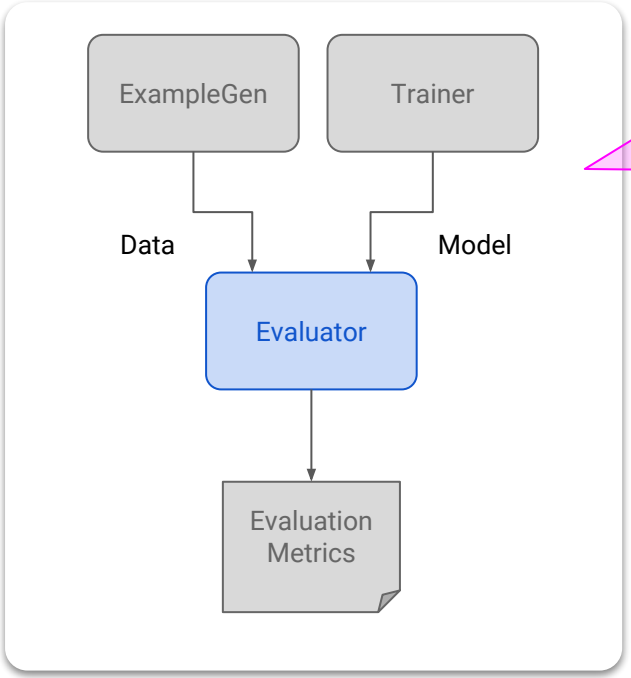
Assess model quality on specific segments / slices

Track performance over time

“Am I getting better at predicting trips with tips > 20%?”

Component: Evaluator

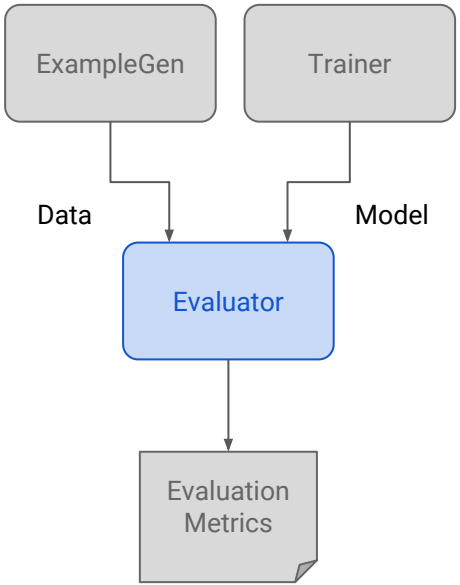
Inputs and Outputs



- Evaluation split of data
- Eval spec for slicing of metrics

Component: Evaluator

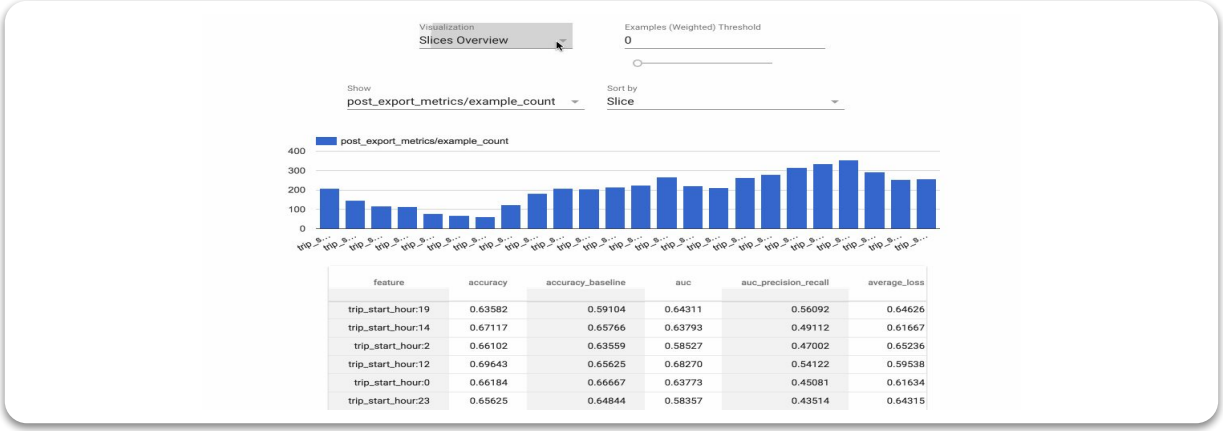
Inputs and Outputs



Configuration

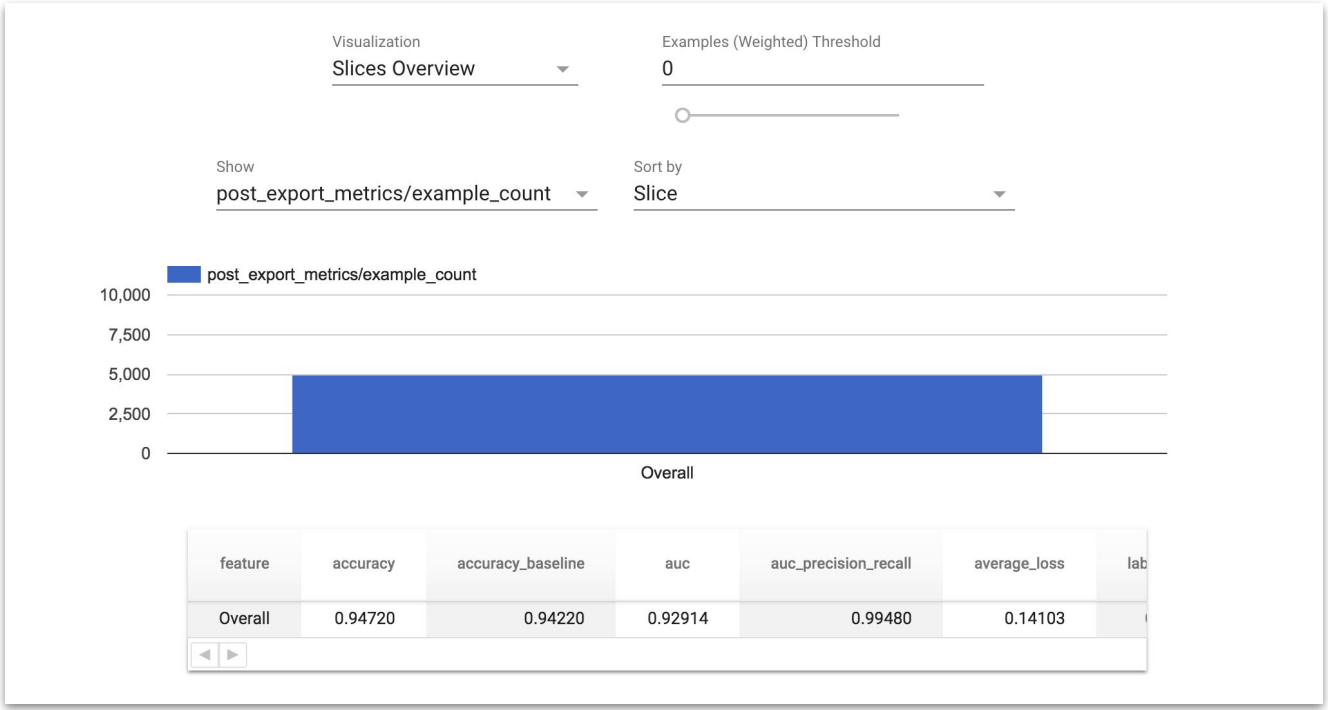
```
model_analyzer = Evaluator(
    examples=examples_gen.outputs.output,
    eval_spec=taxi_eval_spec,
    model_exports=trainer.outputs.output)
```

Visualization

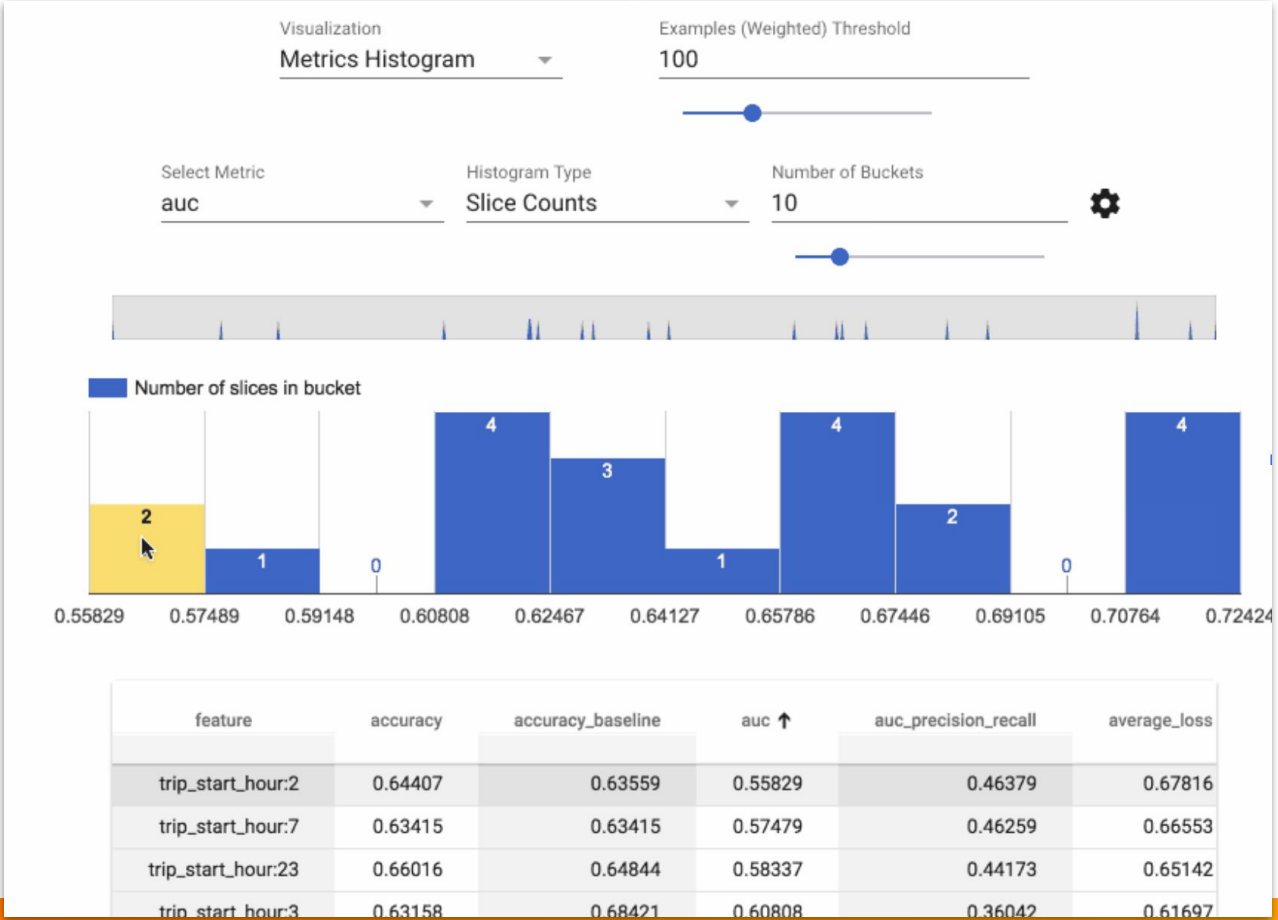


feature	accuracy	accuracy_baseline	auc	auc_precision_recall	average_loss
trip_start_hour:19	0.63582	0.59104	0.64311	0.56092	0.64626
trip_start_hour:14	0.67117	0.65766	0.63793	0.49112	0.61667
trip_start_hour:2	0.66102	0.63559	0.58527	0.47002	0.65236
trip_start_hour:12	0.69643	0.65625	0.68270	0.54122	0.59538
trip_start_hour:0	0.66184	0.66667	0.63773	0.45081	0.61634
trip_start_hour:23	0.65625	0.64844	0.58357	0.43514	0.64315

How well can I predict trips that result in tips > 20%?



Why are my tip predictions sometimes wrong?

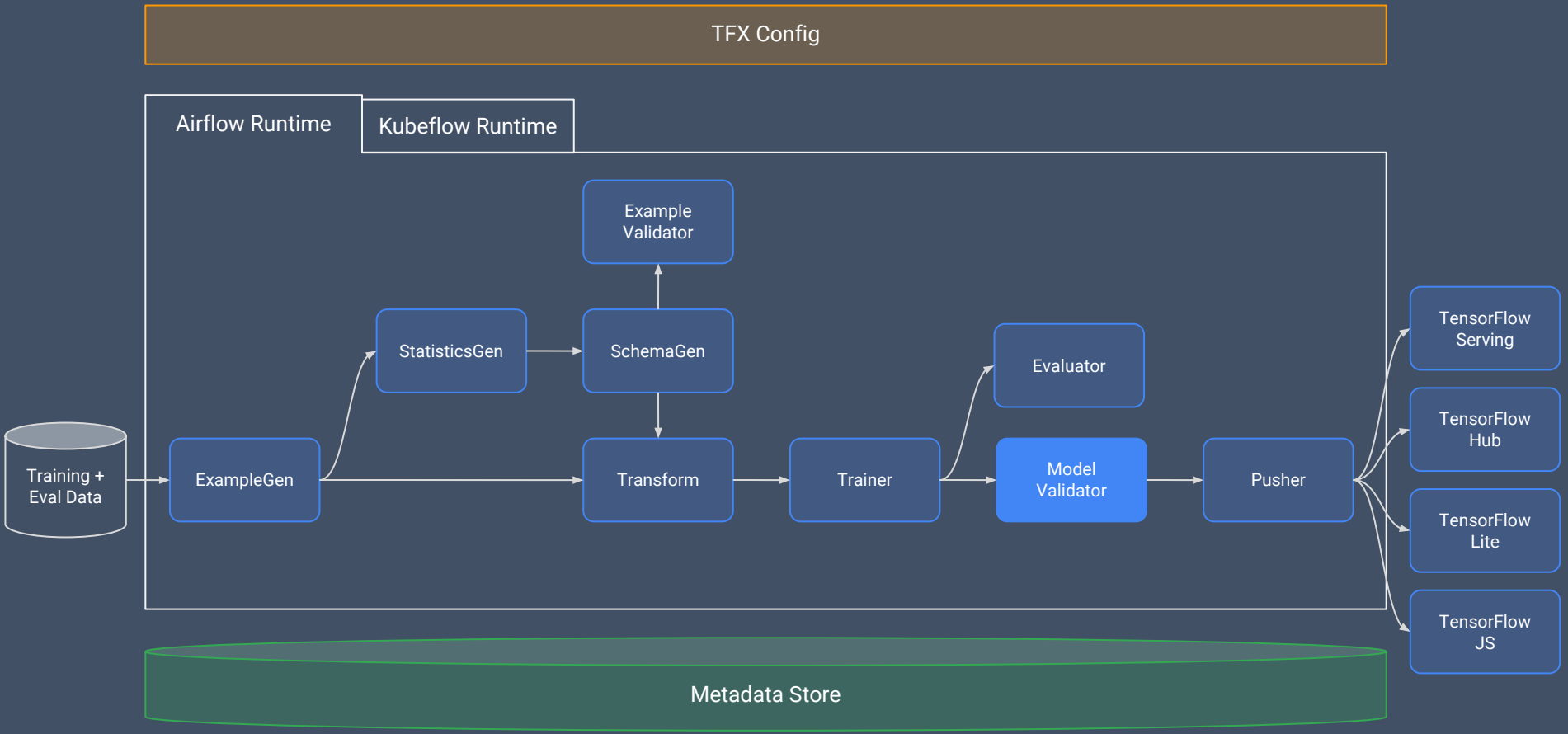


feature	accuracy	accuracy_baseline	auc ↑	auc_precision_recall	average_loss
trip_start_hour:2	0.64407	0.63559	0.55829	0.46379	0.67816
trip_start_hour:7	0.63415	0.63415	0.57479	0.46259	0.66553
trip_start_hour:23	0.66016	0.64844	0.58337	0.44173	0.65142
trip_start_hour:3	0.63158	0.68421	0.60808	0.36042	0.61697

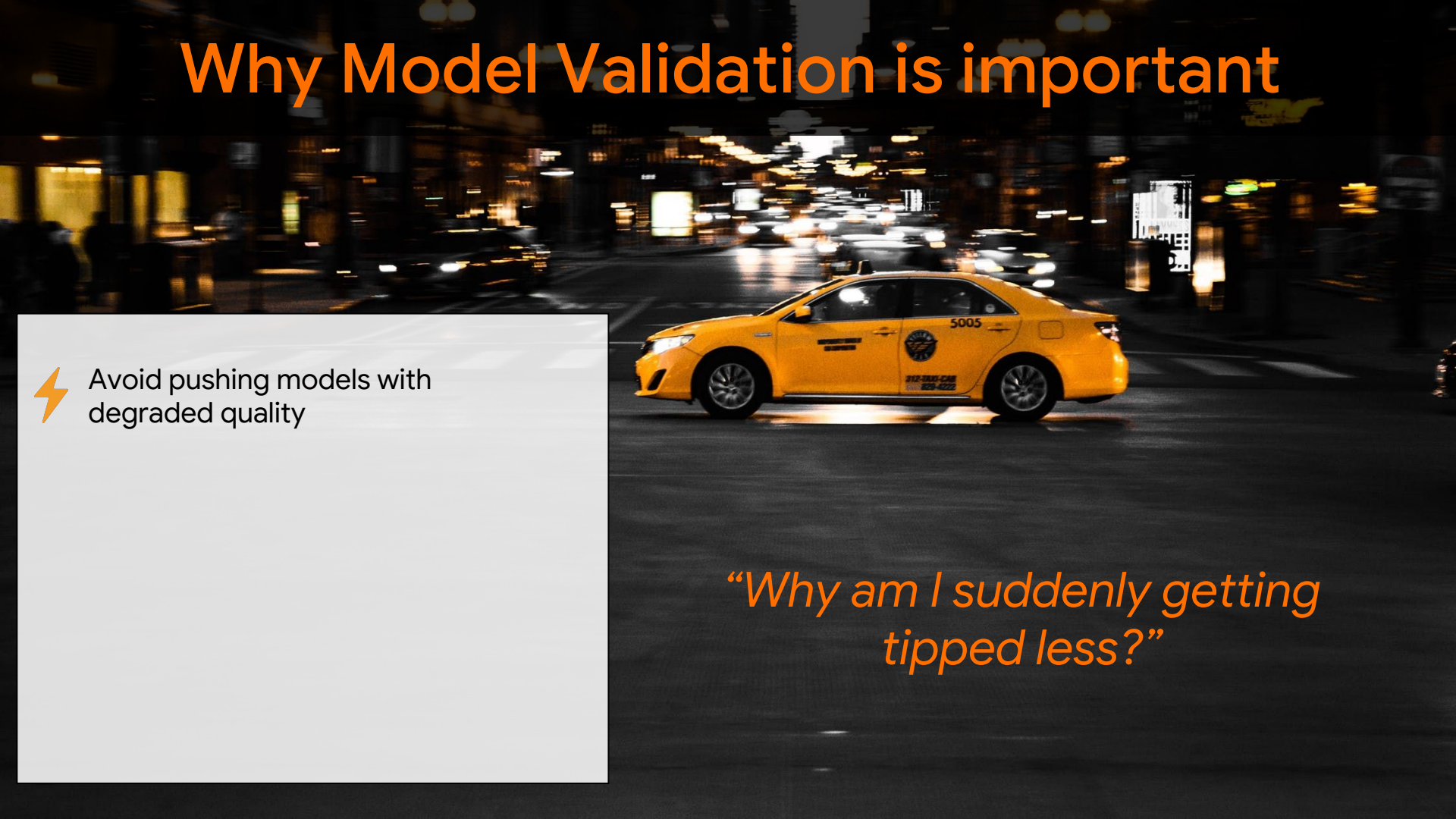
Am I getting better at predicting trips with tips > 20%?

Model	Data	accuracy	accuracy_baseline	auc	auc_precision_recall	average_loss	label/mean	po
1536199479	data.csv	0.94880	0.94220	0.93168	0.98516	0.13980	0.94220	
1536199433	data.csv	0.94700	0.94220	0.93165	0.98170	0.13979	0.94220	
1536199047	data.csv	0.94720	0.94220	0.92914	0.99480	0.14103	0.94220	

ModelValidator



Why Model Validation is important



- Avoid pushing models with degraded quality

“Why am I suddenly getting tipped less?”

Why Model Validation is important

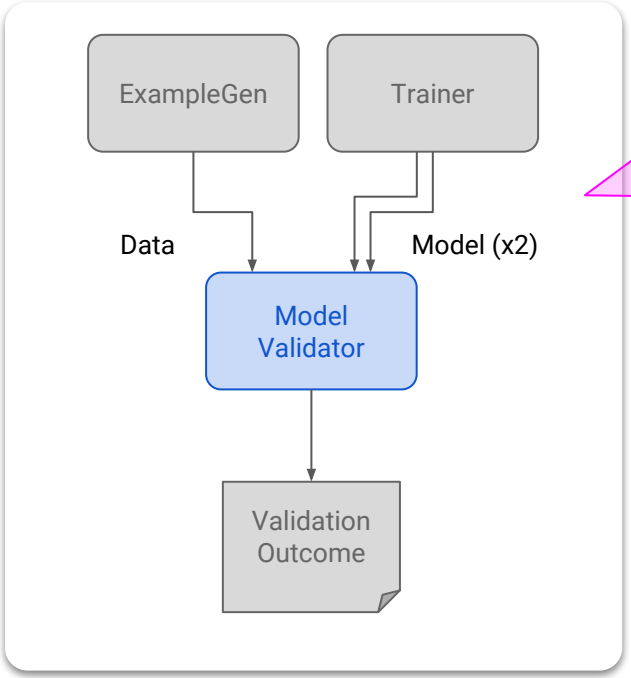
Avoid pushing models with degraded quality.

Avoid breaking downstream components (e.g. serving)

"Why am I no longer getting recommendations for trips?"

Component: ModelValidator

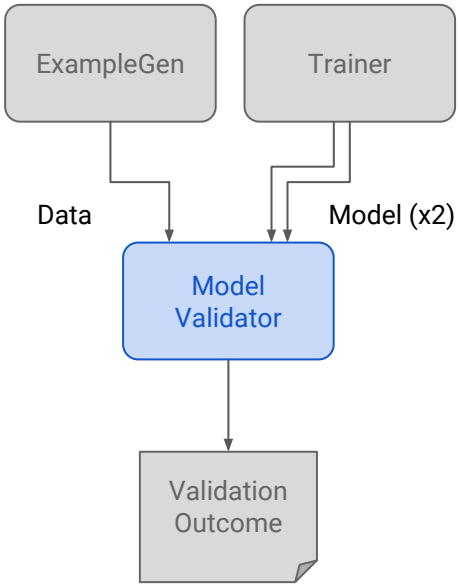
Inputs and Outputs



- Evaluation split of data
- Last validated model
- New candidate model

Component: ModelValidator

Inputs and Outputs

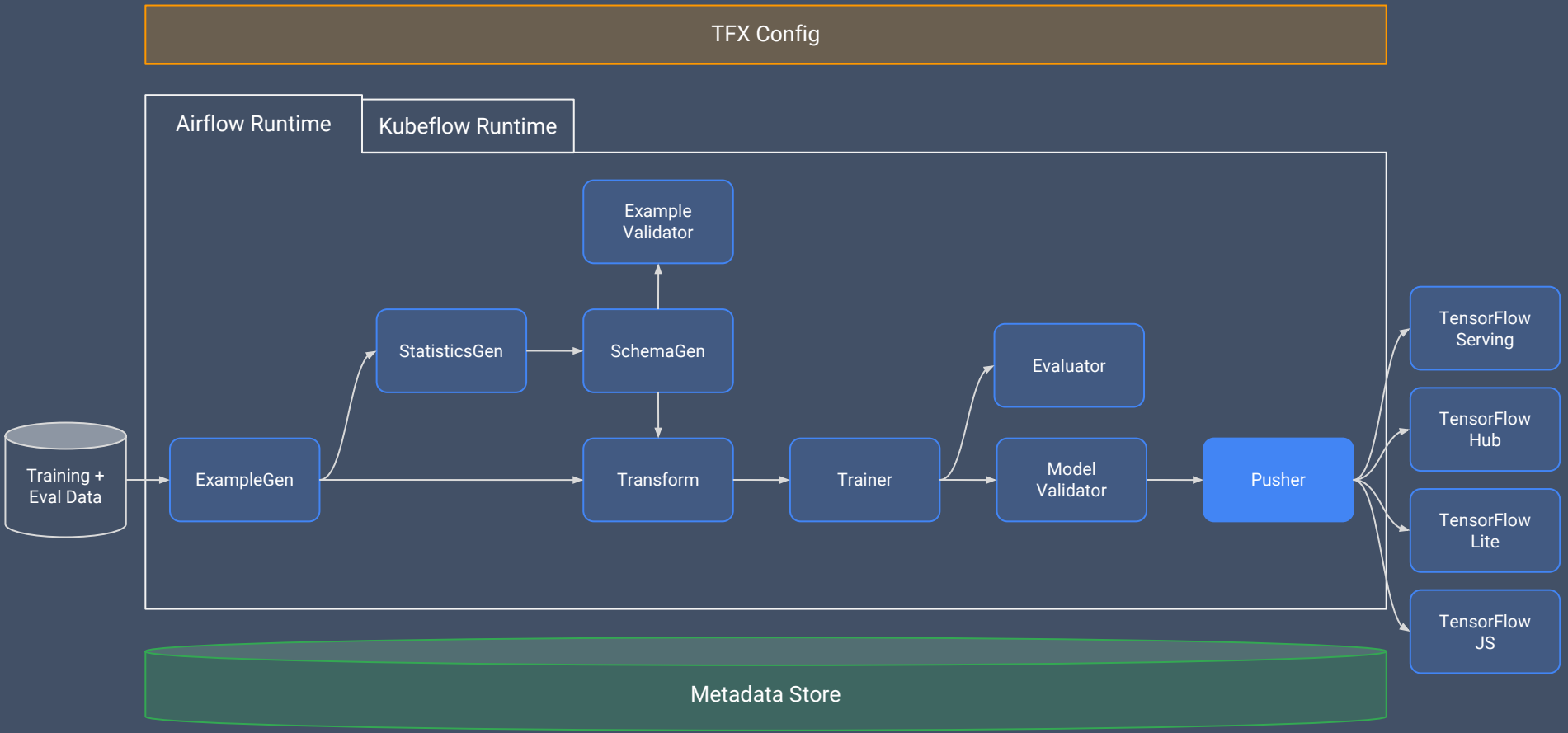


Configuration

```
model_validator = ModelValidator(
    examples=examples_gen.outputs.output,
    model=trainer.outputs.output,
    eval_spec=taxi_mv_spec)
```

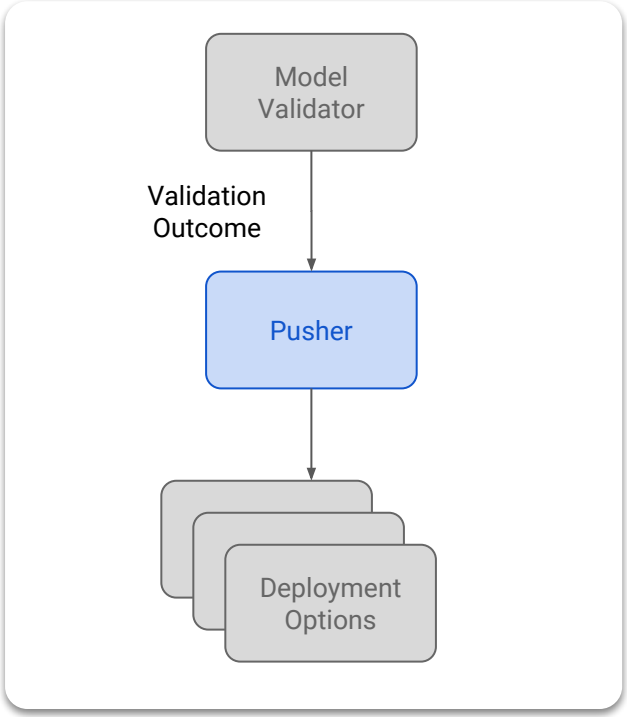
- Configuration options
  - Validate using current eval data
  - “Next-day eval”, validate using unseen data

Pusher


TFX Config
Airflow Runtime
Kubeflow Runtime
Example Validator
StatisticsGen
SchemaGen
Evaluator
Training + Eval Data
ExampleGen
Transform
Trainer
Model Validator
Pusher
TensorFlow Serving
TensorFlow Hub
TensorFlow Lite
TensorFlow JS
Metadata Store

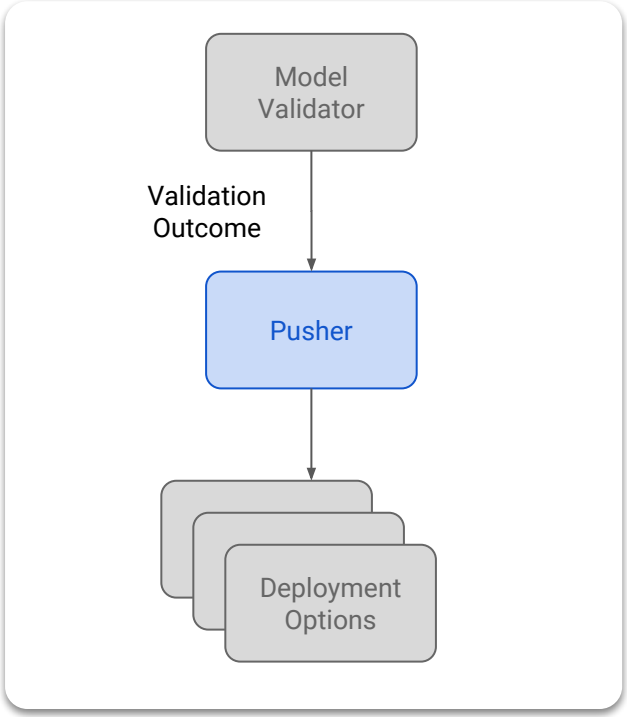
Component: Pusher

Inputs and Outputs



Component: Pusher

Inputs and Outputs



Configuration

```
pusher = Pusher(
    model_export=trainer.outputs.output,
    model_blessing=model_validator.outputs.blessing,
    serving_model_dir=serving_model_dir)
```

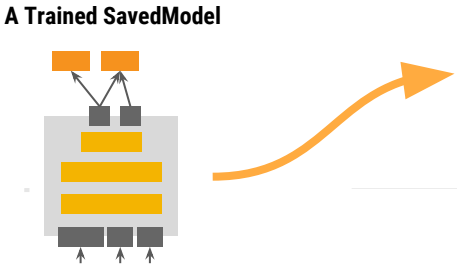
- Block push on validation outcome
- Push destinations supported today
  - Filesystem
  - TF Serving model server

TensorFlow

Model Deployment

Clemens Mewald

This is where you are ...




```
assets/
variables/
    variables.data-*****-of-*****
    variables.index
saved_model.pb
```

This is where you want to be ...

```
$ curl -d '{"instances": [60, 27.05, ...]}' -X POST
  http://localhost:8501/v1/models/chicago_taxi:predict
{ "results": [1.0] }
```

Deploy anywhere

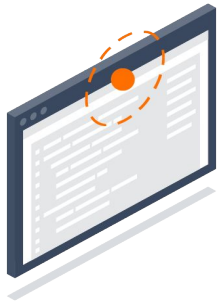
Servers

TensorFlow
Serving

Edge devices

TensorFlow
Lite

JavaScript



TensorFlow
.JS

TensorFlow Serving

Flexible

- Multi-tenancy
- Optimize with GPU and TensorRT
- gRPC or REST API

TensorFlow Serving

High-Performance

- Low-latency
- Request Batching
- Traffic Isolation

TensorFlow Serving

Production-Ready

- Used for years at Google, millions of QPS
- Scale in minutes
- Dynamic version refresh

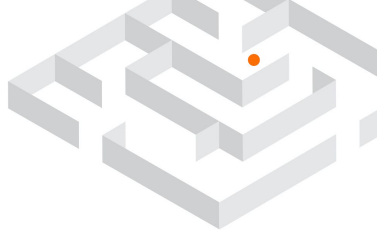
Deploy a REST API for your model in minutes ..

```
$ docker run -p 8501:8501 \
    -v '/path/to/savedmodel':/models/chicago_taxi
    -e MODEL_NAME=chicago_taxi -t tensorflow/serving
```

... using Docker ...

... or locally on your host ...

```
$ apt-get install tensorflow-model-server
$ tensorflow_model_server
    --port=8501
    --model_name=chicago_taxi
    --model_base_path='/path/to/savedmodel'
```



Easily enable* hardware acceleration

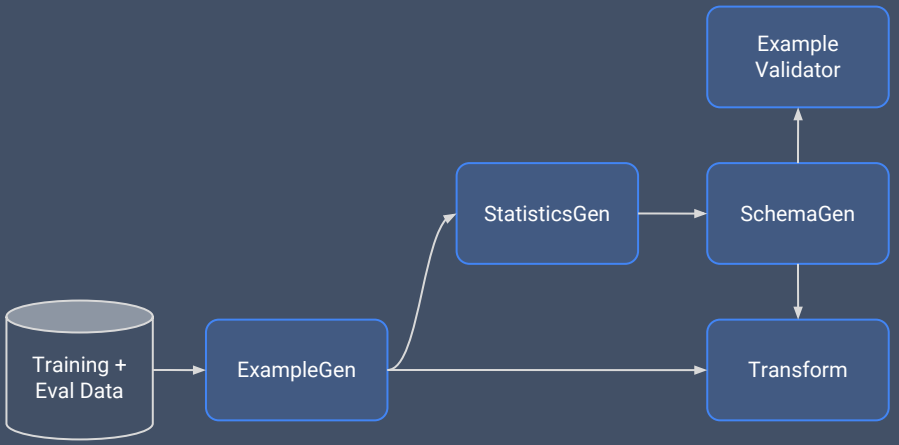
```
$ docker run -p 8501:8501 \
    -v '/path/to/savedmodel':/models/chicago_taxi
    -e MODEL_NAME=chicago_taxi -t tensorflow/serving

$ docker run --runtime=nvidia -p 8501:8501 \
    -v /path/to/savedmodel:/models/chicago_taxi
    -e MODEL_NAME=chicago_taxi -t tensorflow/serving:latest-gpu
```

* Only possible if running on a host equipped with a GPU and nvidia-docker installed

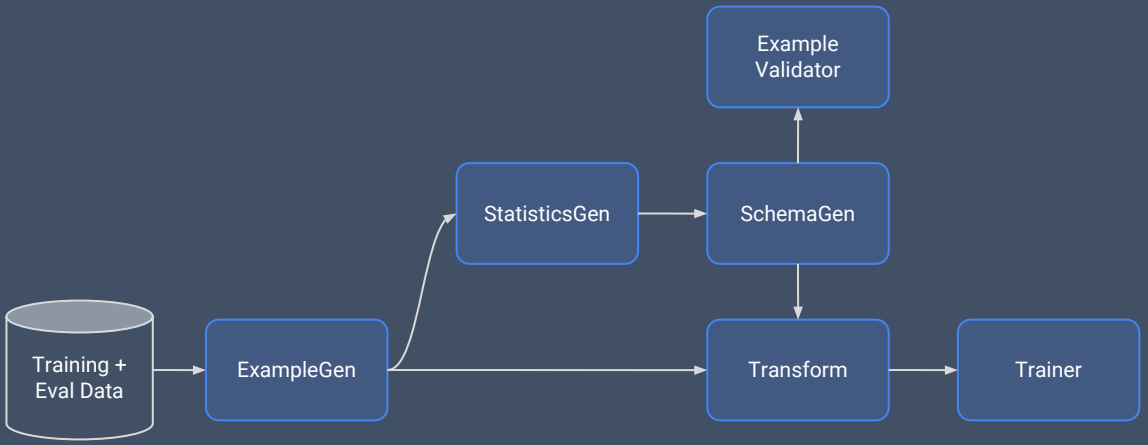
Optimize your model for serving using TensorRT

```
$ saved_model_cli convert --dir '/path/to/savedmodel'
--output_dir '/path/to/trt-savedmodel' --tag_set serve tensorrt


$ docker run --runtime=nvidia -p 8501:8501 \
    -v /path/to/trt-savedmodel:/models/chicago_taxi
    -e MODEL_NAME=chicago_taxi -t tensorflow/serving:latest-gpu
```

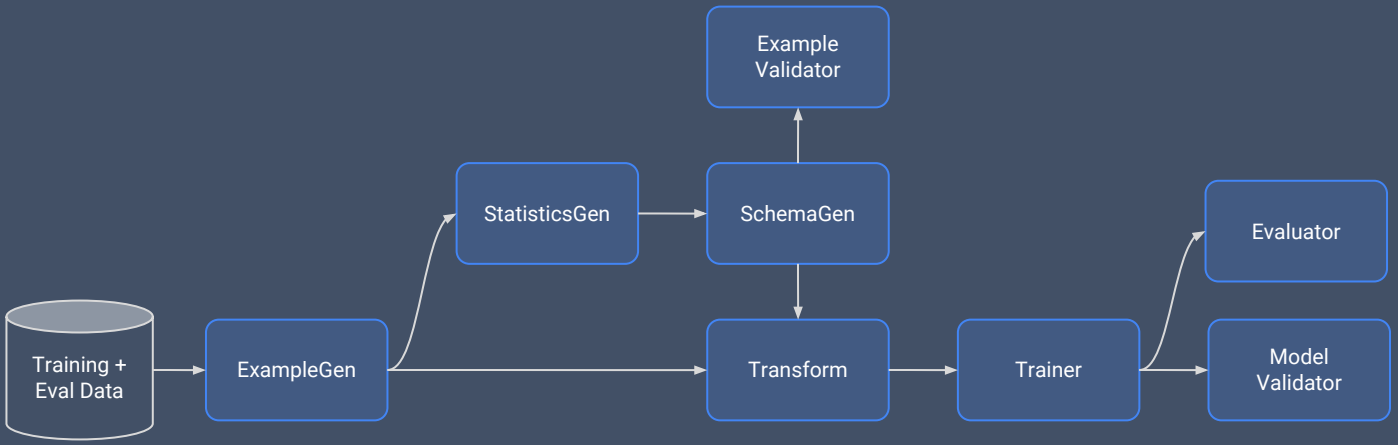
Putting it all together again



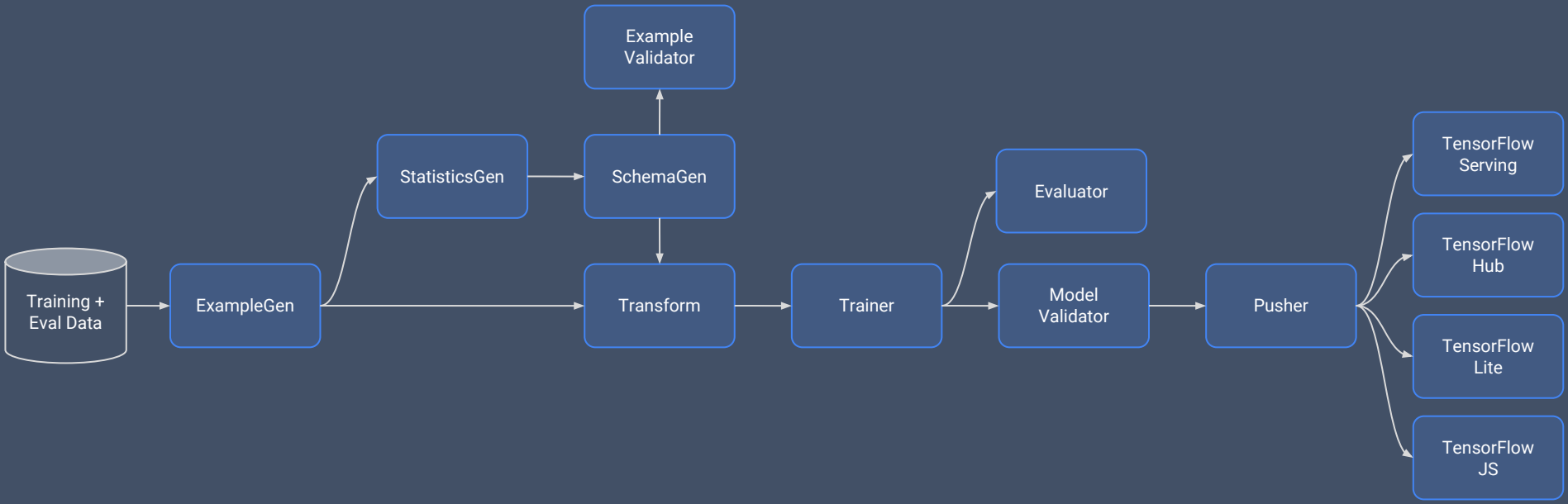
Putting it all together again



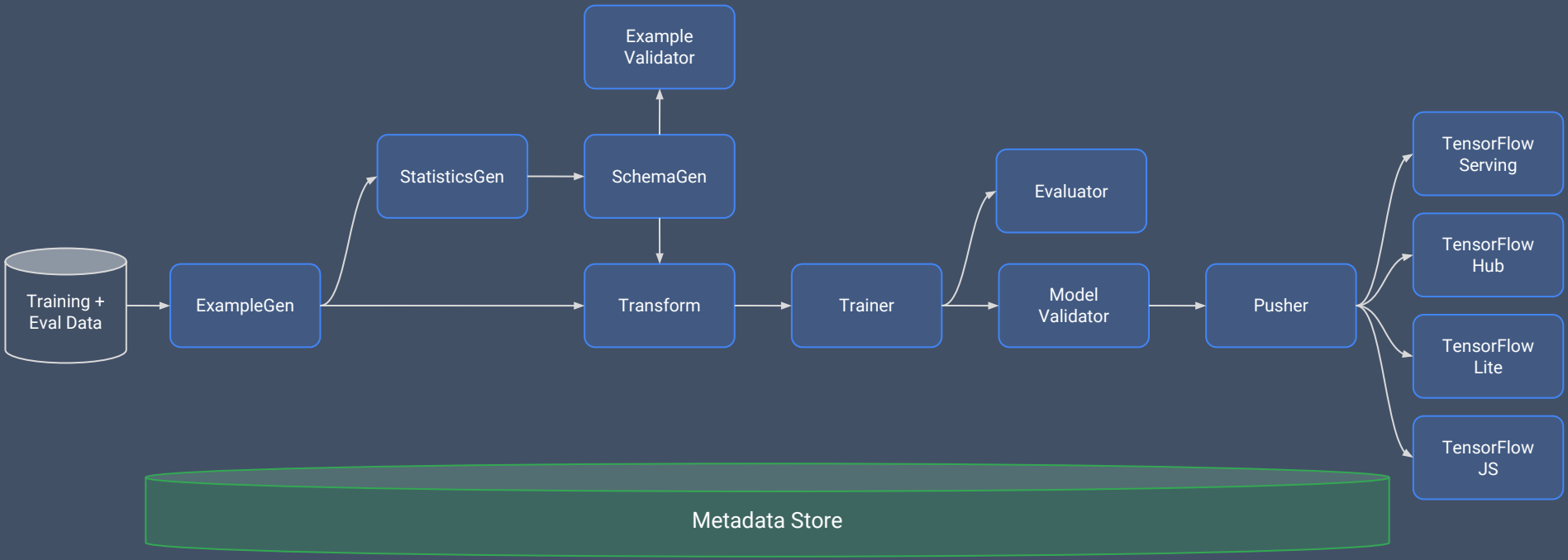
Putting it all together again



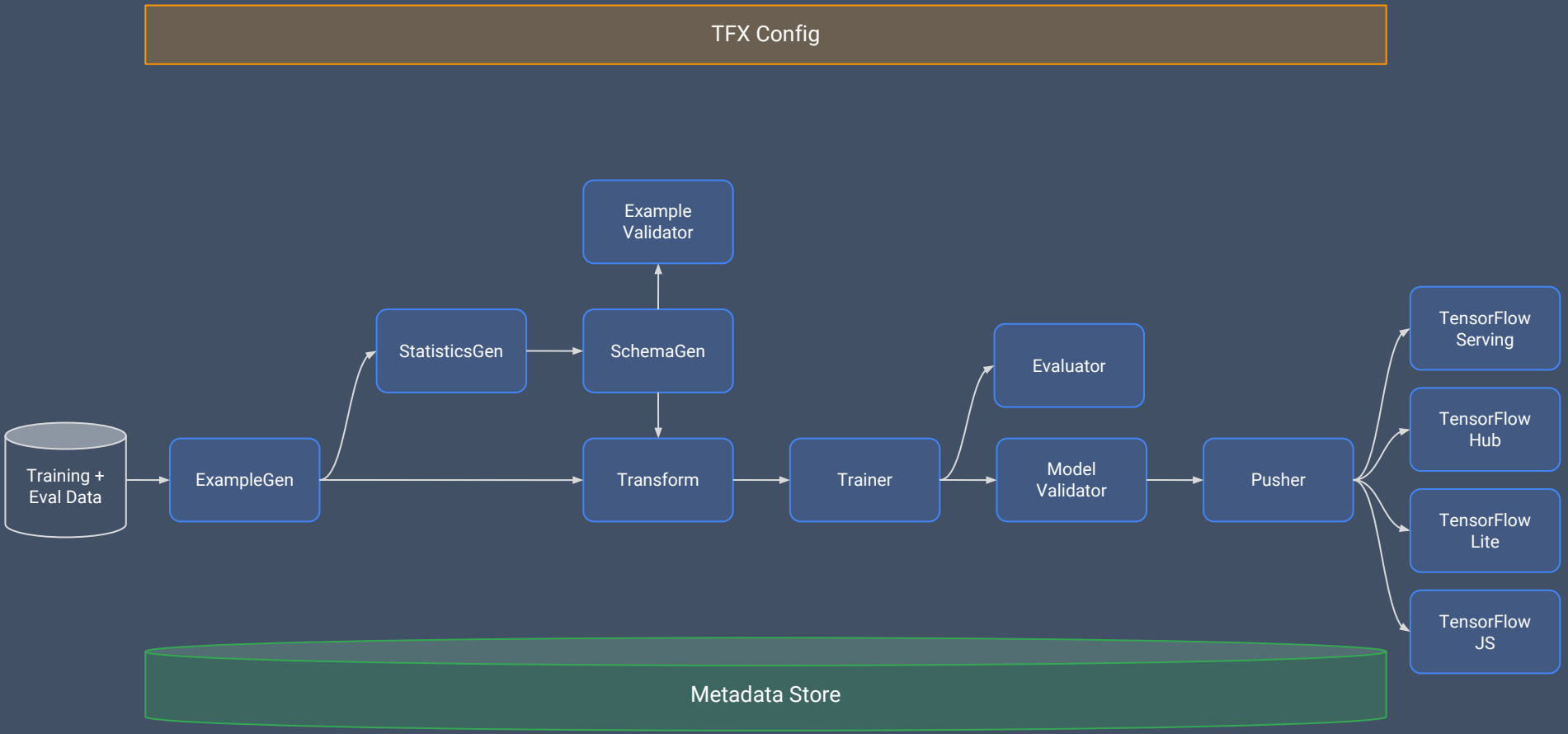
Putting it all together again



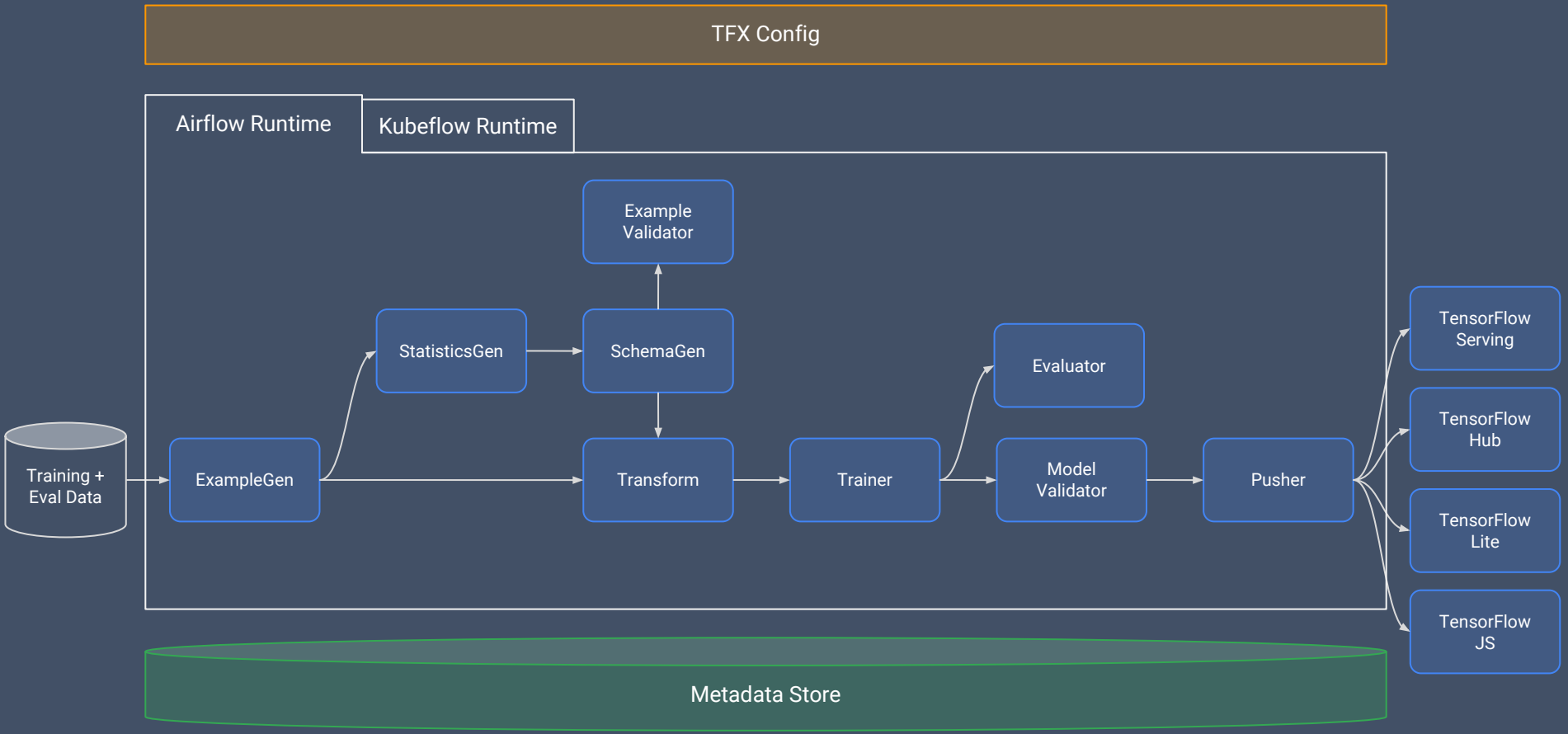
Putting it all together again



Putting it all together again



Putting it all together again



TensorFlow

Get started with TensorFlow Extended (TFX)

An End-to-End ML Platform

github.com/tensorflow/tfx

tensorflow.org/tfx