TensorFlow

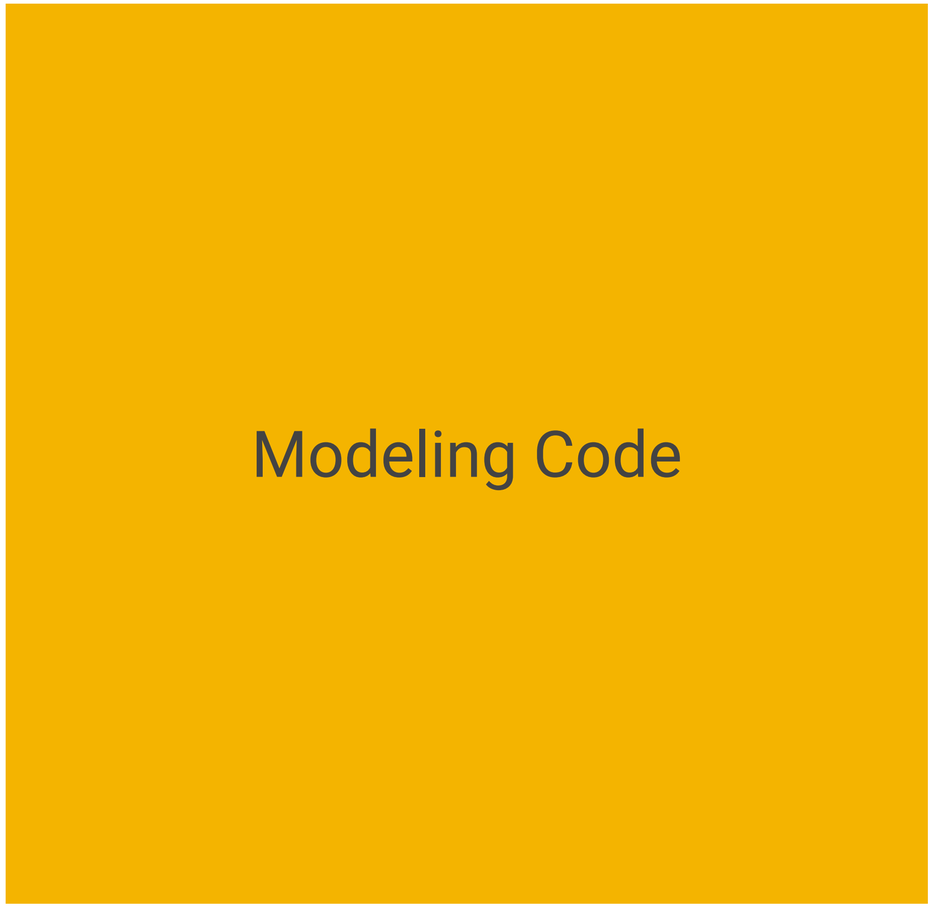
TensorFlow

TensorFlow Extended (TFX)

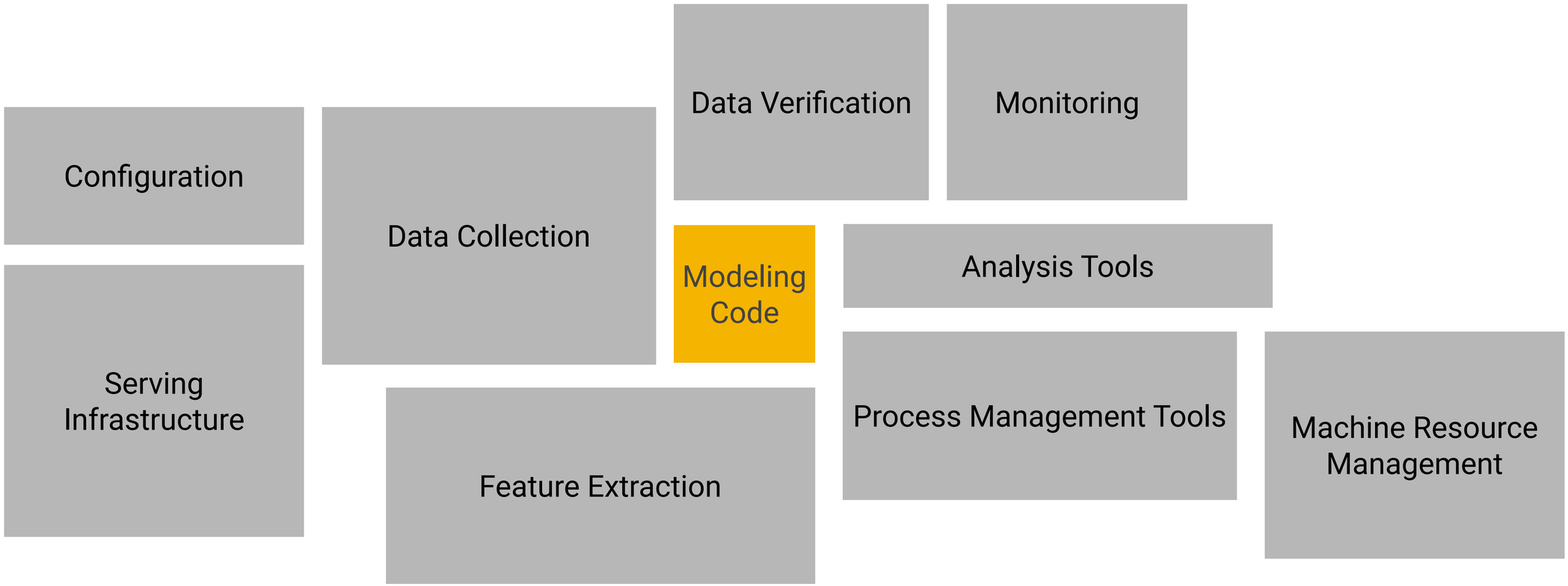
Real World Machine Learning in Production

Robert Crowe
TensorFlow Developer Advocate
@robert_crowe

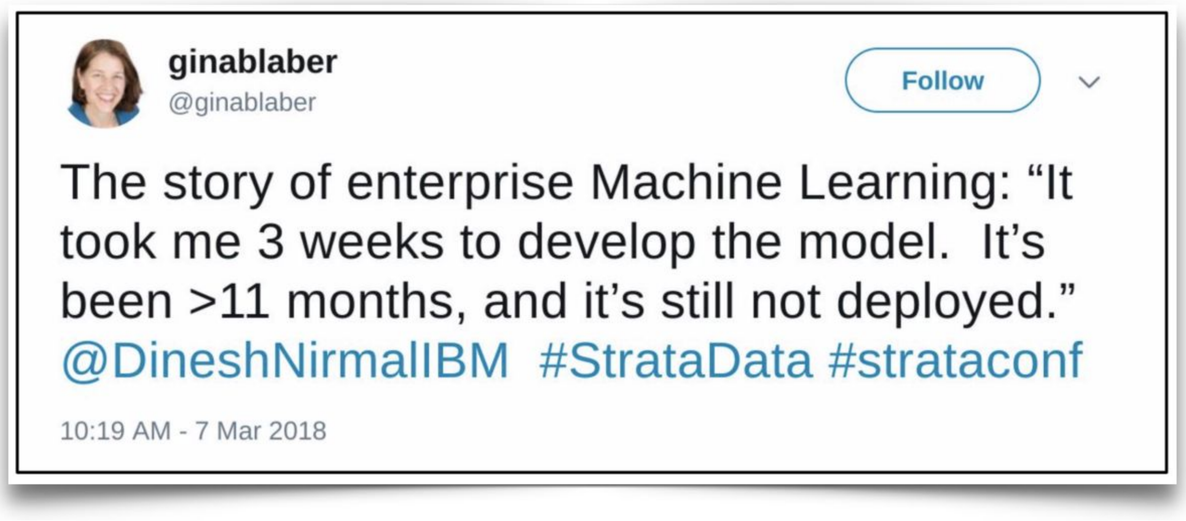
In addition to training an amazing model ...



… a production solution requires so much more

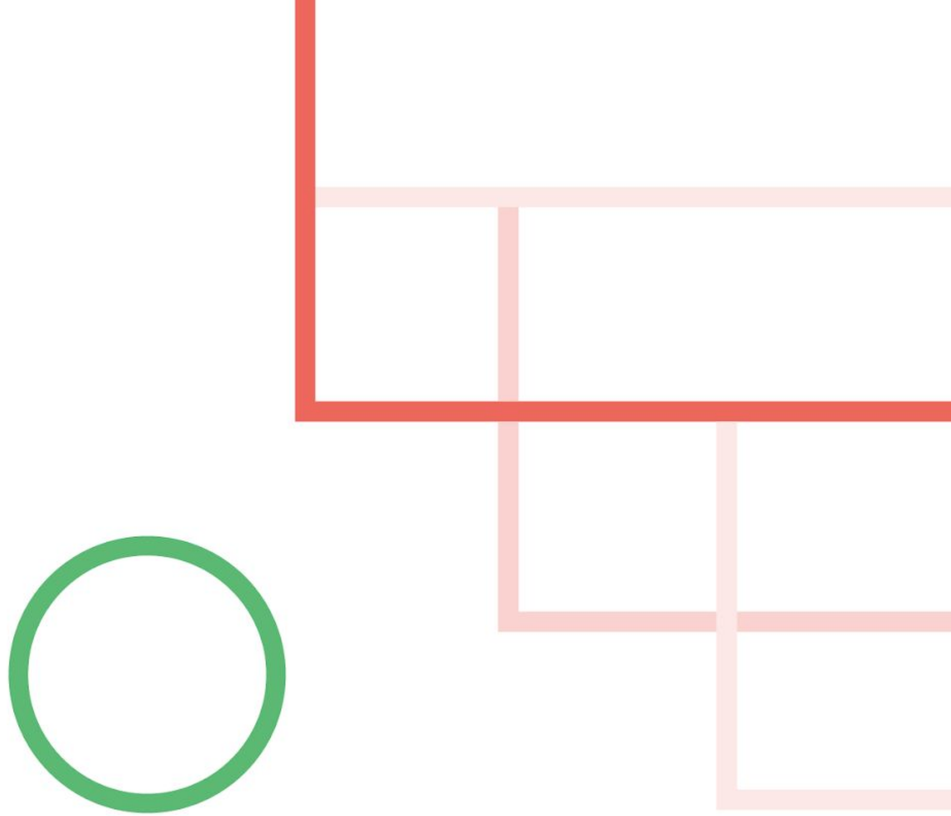


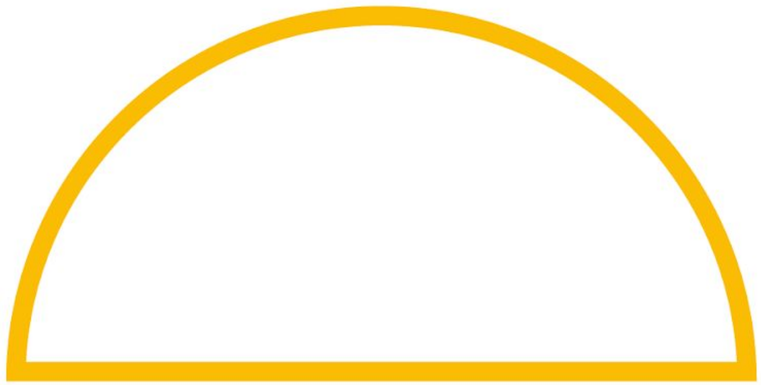
Tales From The Trenches



https://twitter.com/ginablaber/status/971450218095943681

Tensorflow Extended (TFX)





Tensorflow Extended (TFX)

Powers Alphabet's most important bets and products

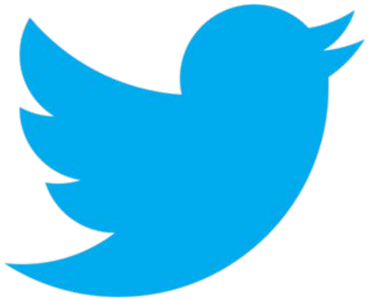









... and some of our most important partners.



"... we have re-tooled our machine learning platform to use TensorFlow. This yielded significant productivity gains while positioning ourselves to take advantage of the latest industry research."

Ranking Tweets with TensorFlow - Twitter blog post

Production Machine Learning

Machine Learning Development

- Labeled data
- Feature space coverage
- Minimal dimensionality
- Maximum predictive data
- Fairness
- Rare conditions
- Data lifecycle management

Production Machine Learning

Machine Learning Development

- Labeled data
- Feature space coverage
- Minimal dimensionality
- Maximum predictive data
- Fairness
- Rare conditions
- Data lifecycle management

Modern Software Development

- Scalability
- Extensibility
- Configuration
- Consistency & Reproducibility
- Modularity
- Best Practices
- Testability
- Monitoring
- Safety & Security

Production Machine Learning

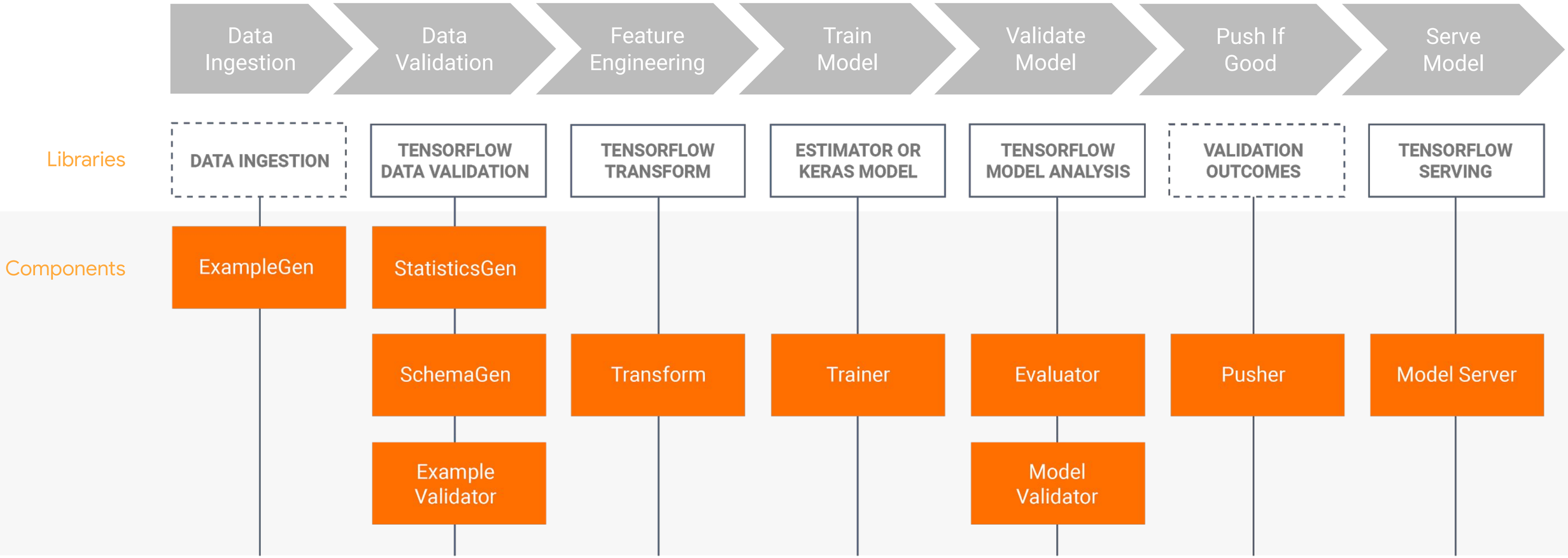
“Hidden Technical Debt in Machine Learning Systems”

NIPS 2015

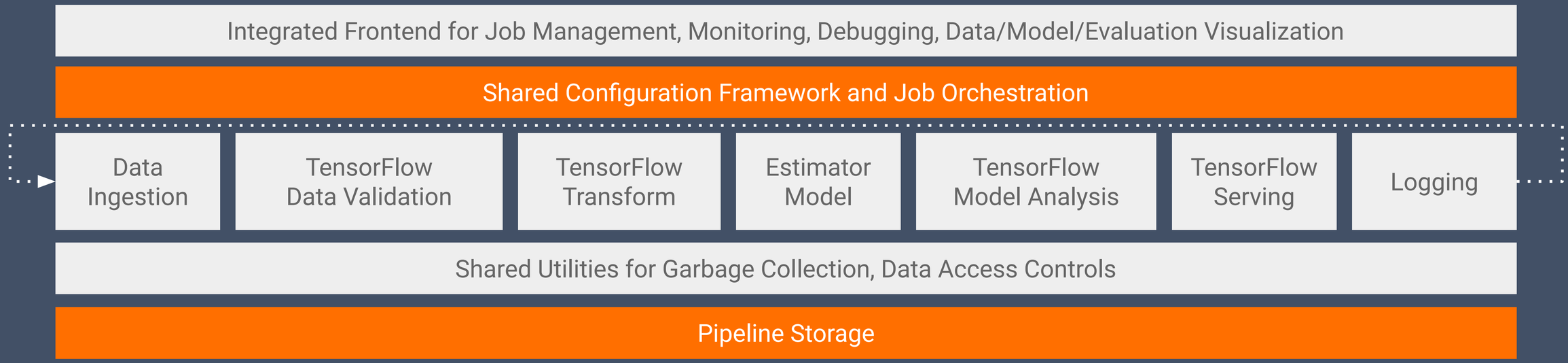
http://bit.ly/ml-techdebt

Data Lifecycle

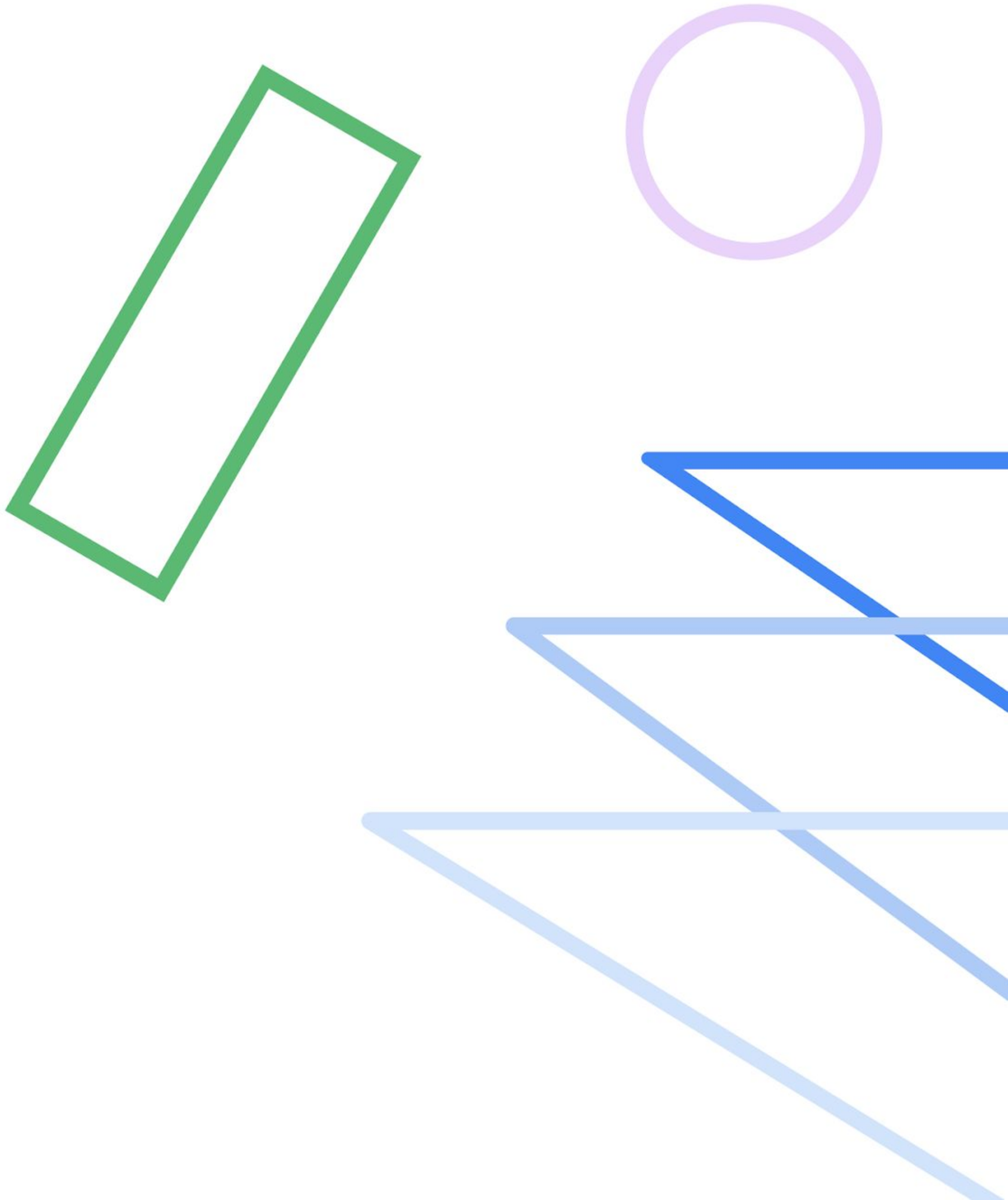
TFX Production Components

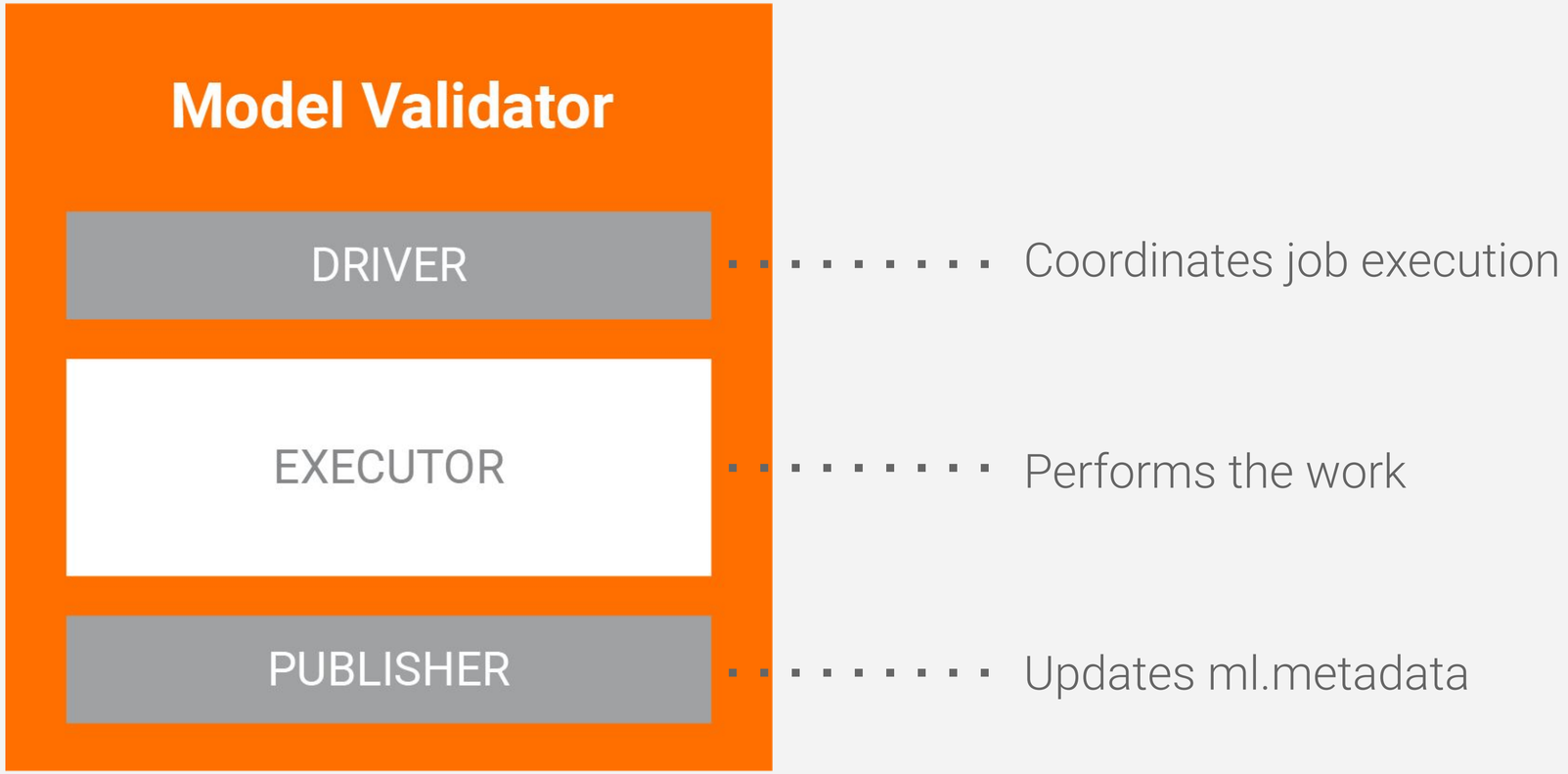


Horizontal Layers Coordinate Components

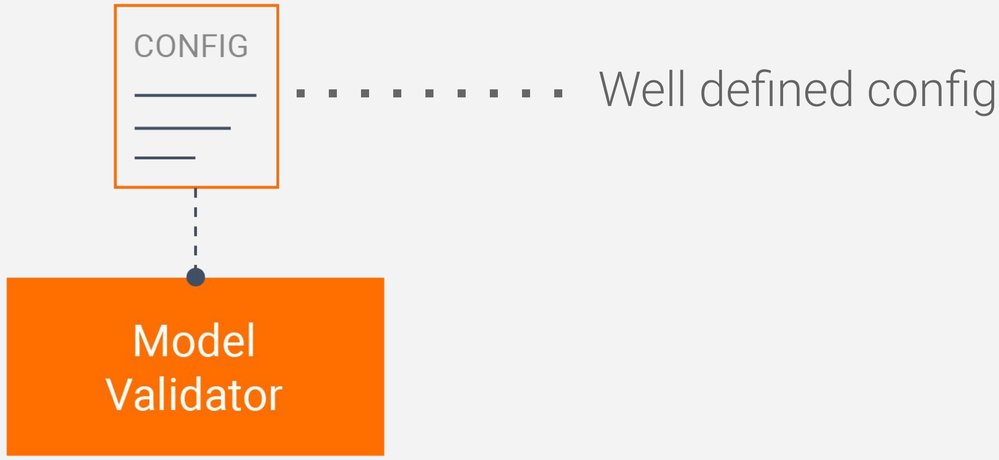


What is a Component?

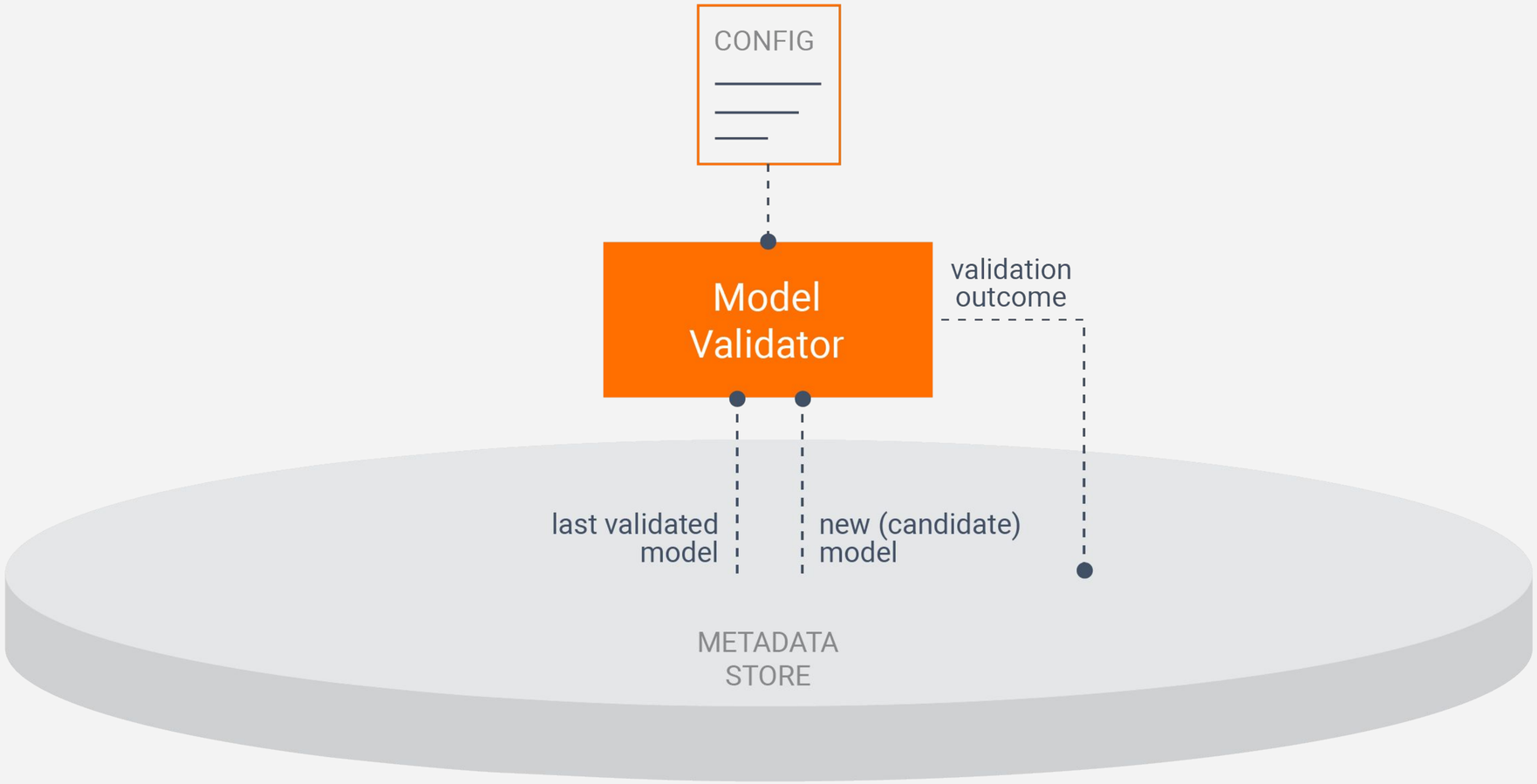




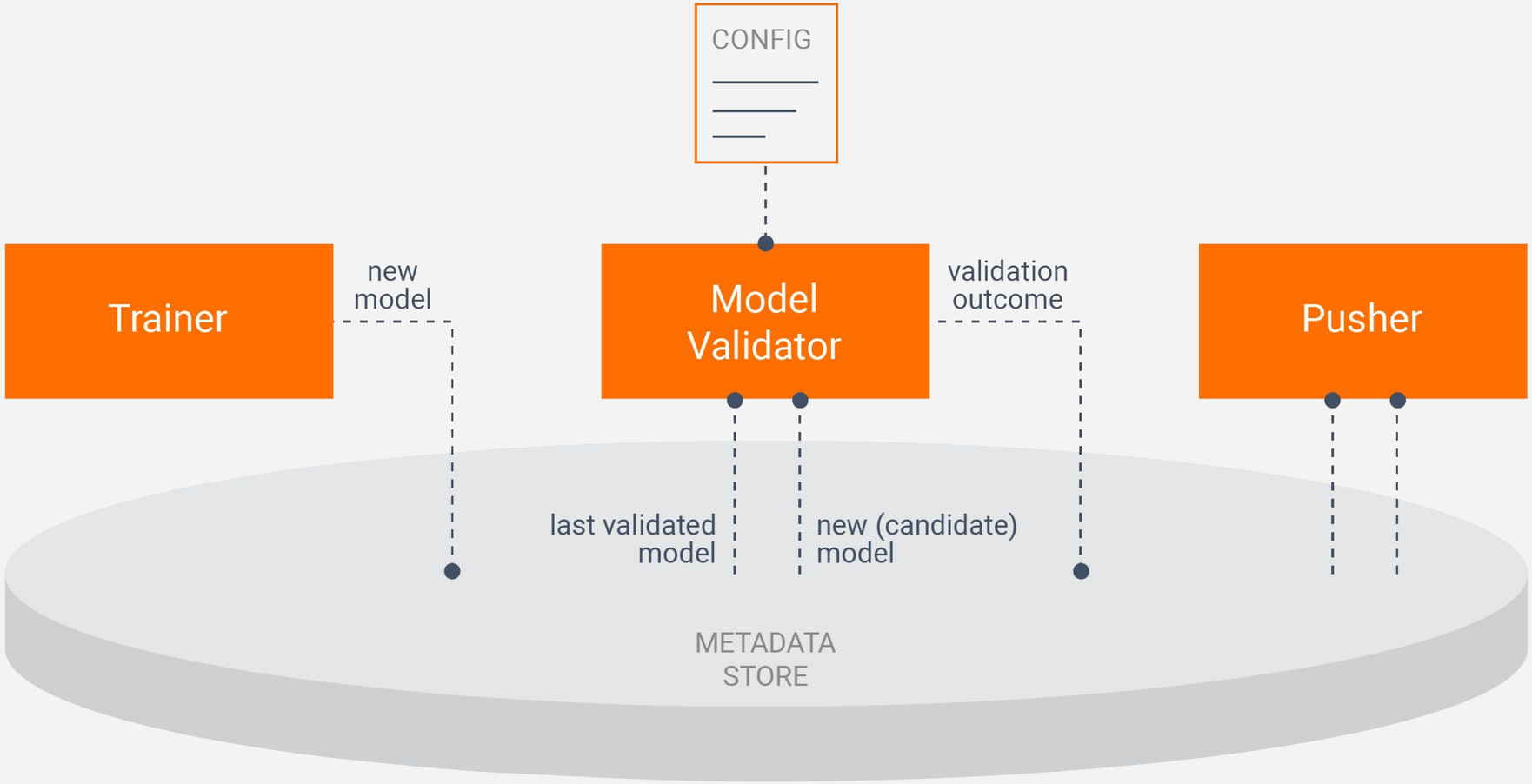
What makes a Component



What makes a Component?

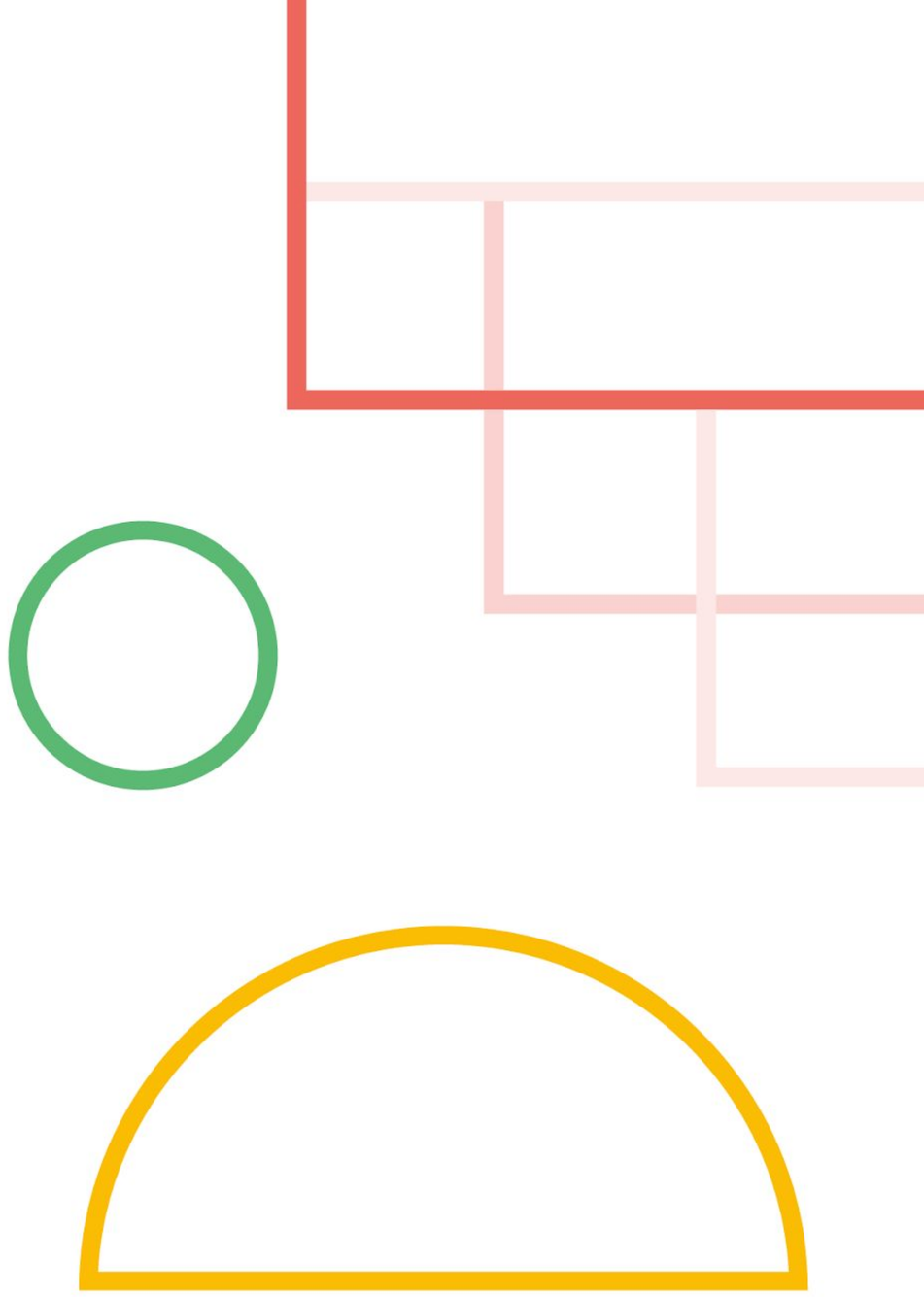


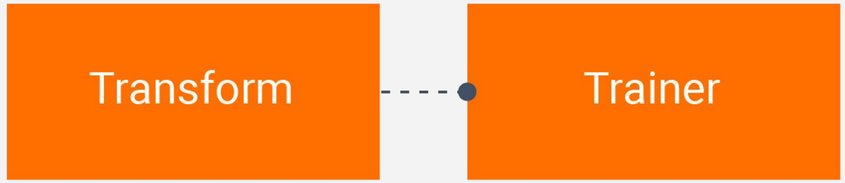
What makes a Component?



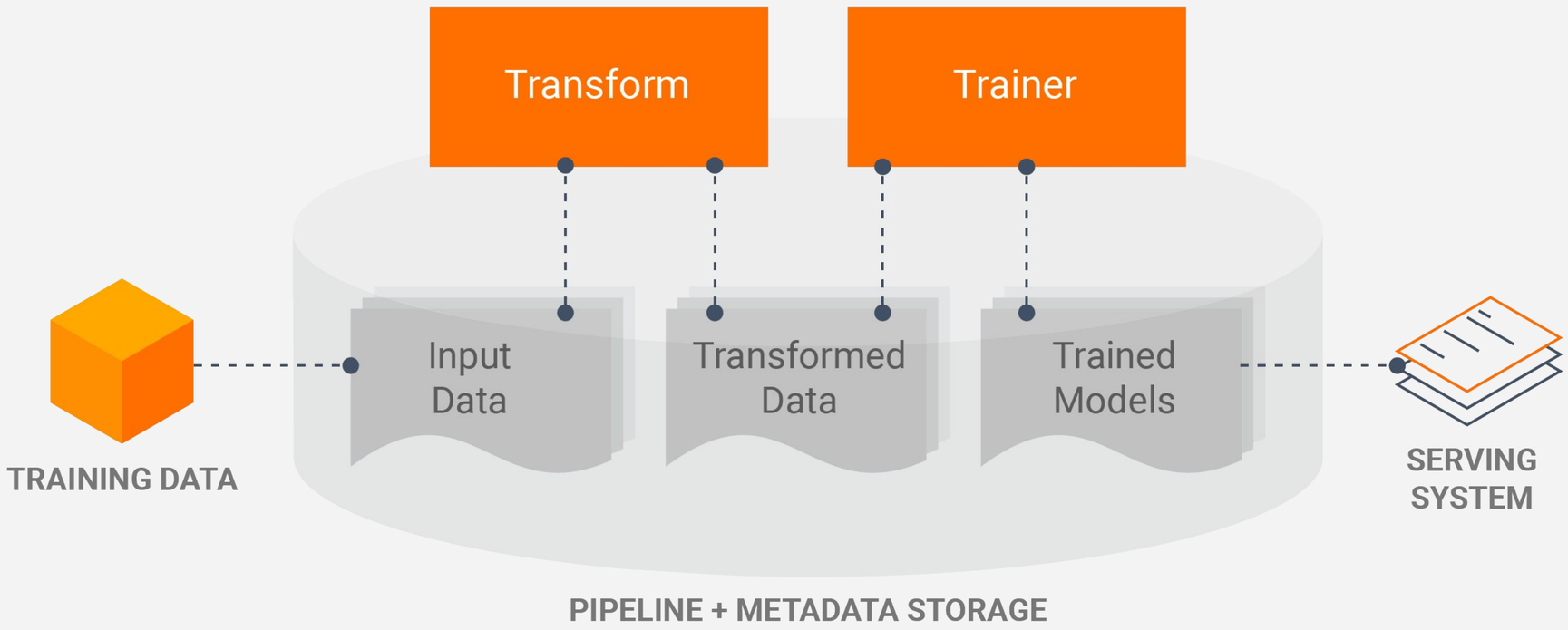
What makes a Component?

Orchestration Styles





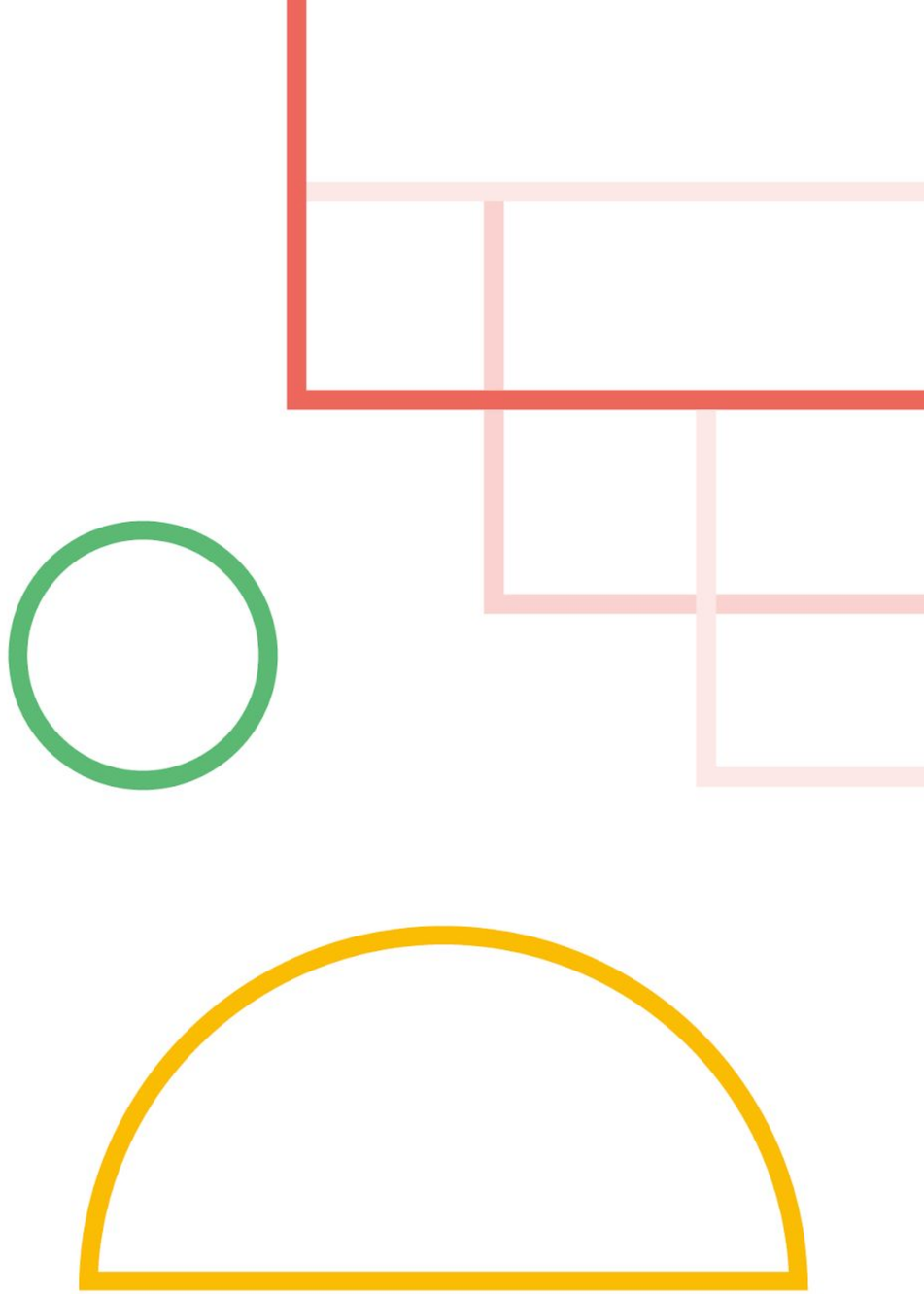
Task-Aware Pipelines

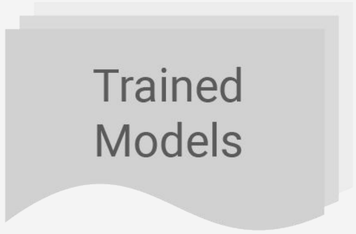


Task- and Data-Aware Pipelines

Metadata Store

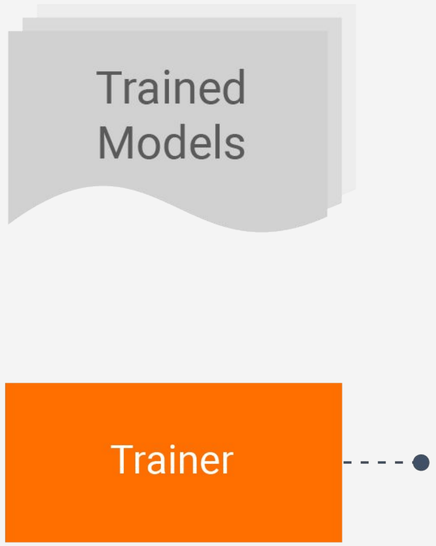
TFX: Metadata Store
What does it contain?





What is in Metadata Store?

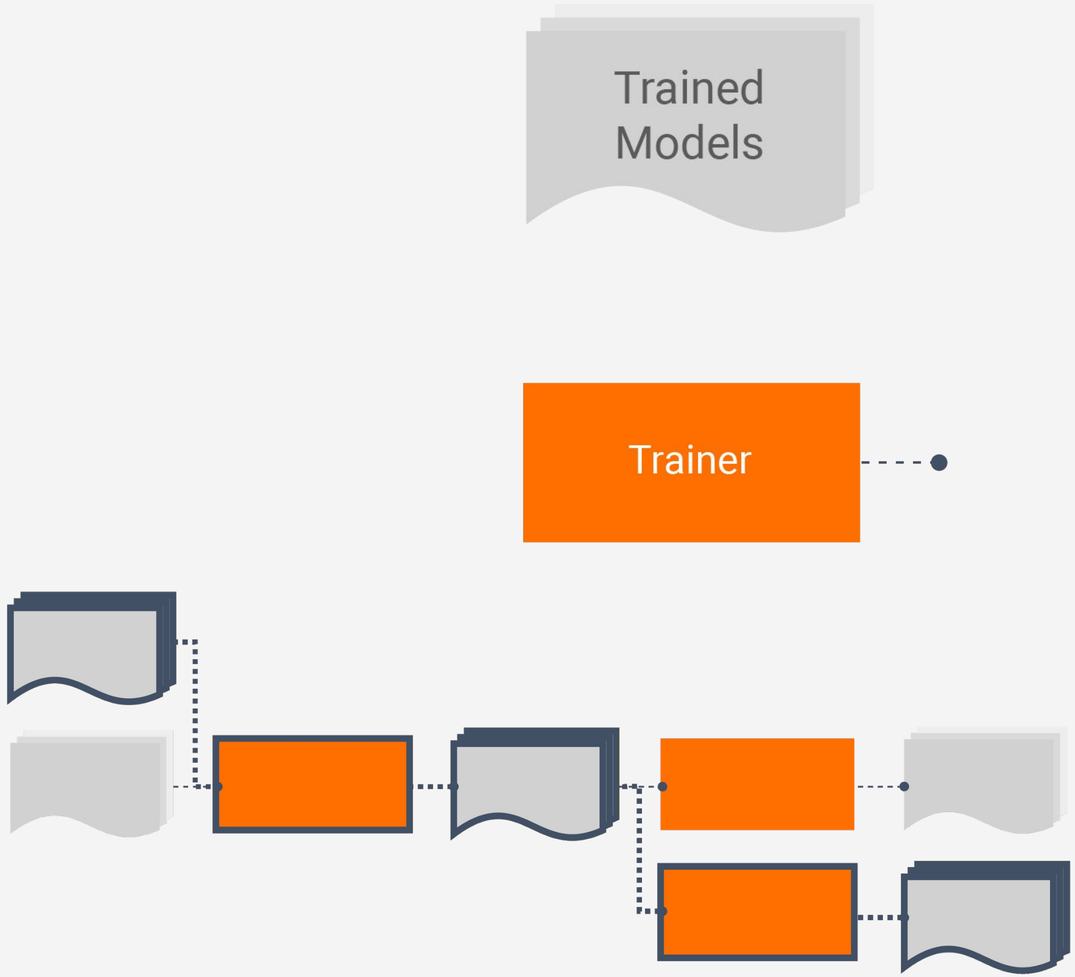
Type definitions of Artifacts and their Properties



What is in Metadata Store?

Type definitions of Artifacts and their Properties

Execution Records (Runs) of Components



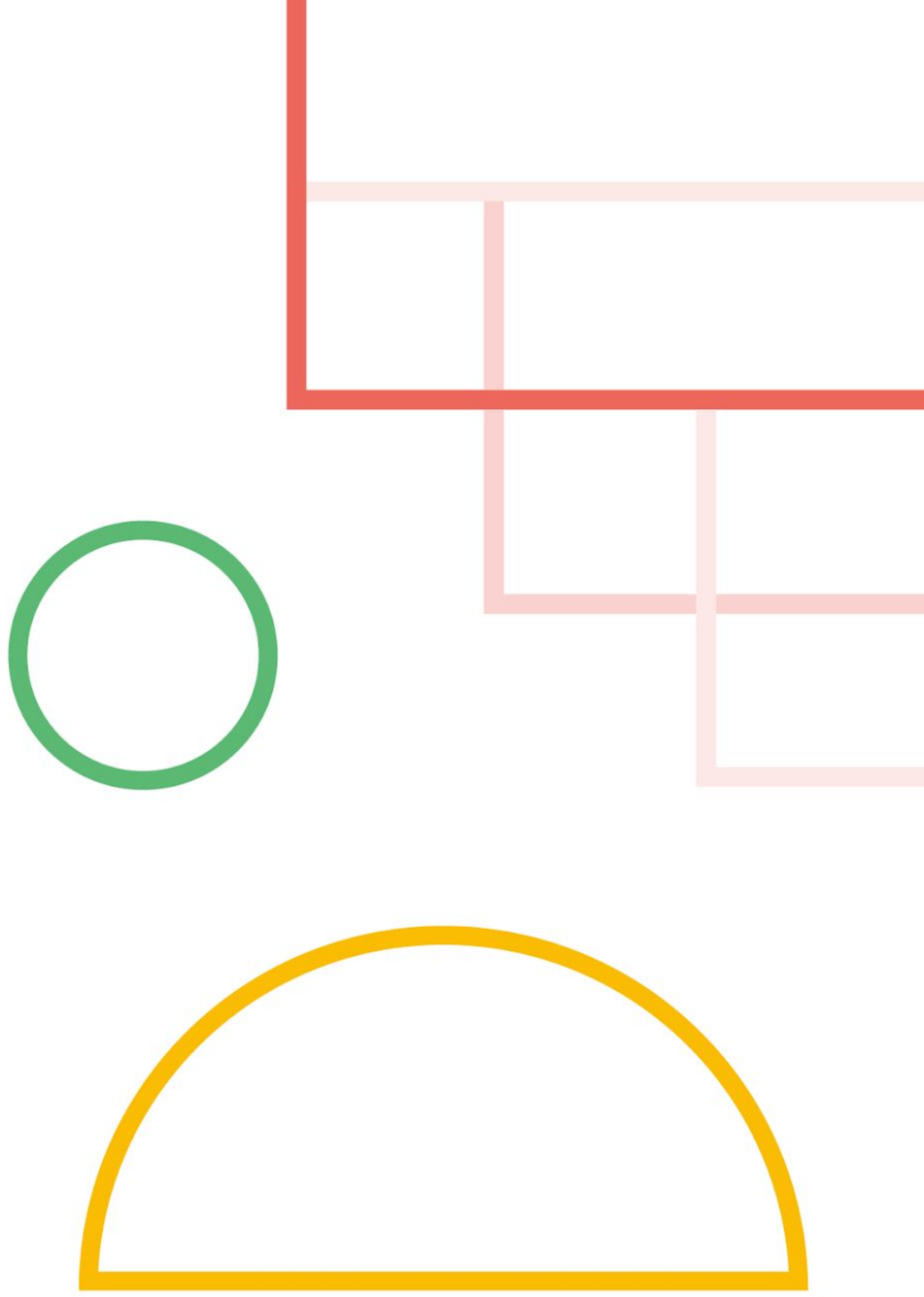
What is in Metadata Store?

Type definitions of Artifacts and their Properties

Execution Records (Runs) of Components

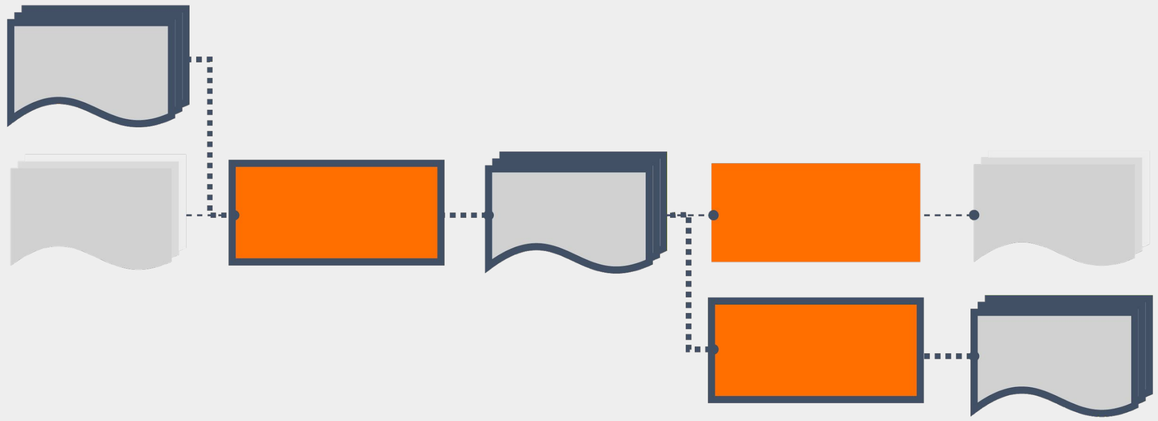
Data Provenance Across All Executions

Metadata-Powered Functionality



Metadata-Powered Functionality

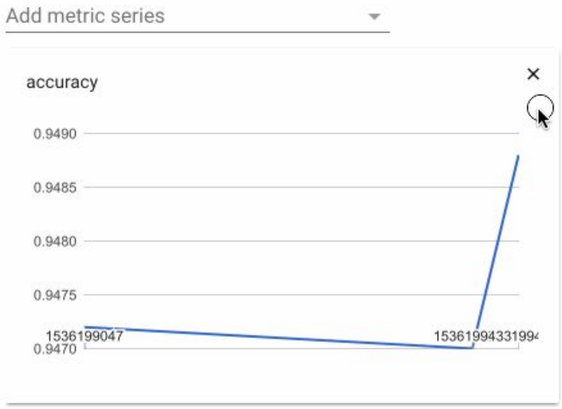
Find out which data a model was trained on



Metadata-Powered Functionality

Compare previous model runs

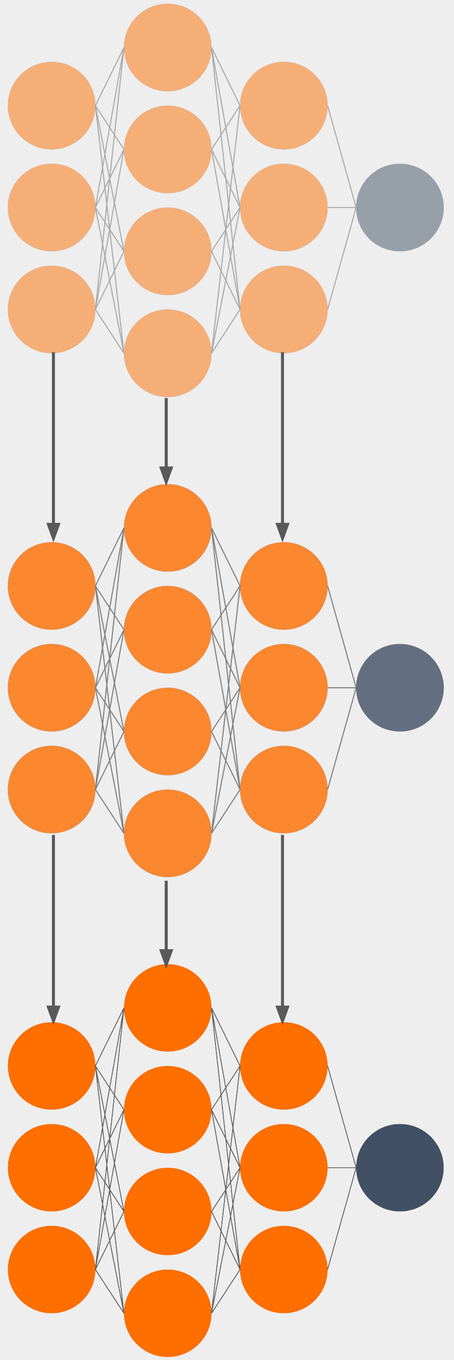
```
eval_results = tfma.make_eval_results([tfma_result_1, tfma_result_2, tfma_result_3],
                                      tfma.constants.MODEL_CENTRIC_MODE)
tfma.view.render_time_series(eval_results, OVERALL_SLICE_SPEC)
```



Model	Data	accuracy	accuracy_baseline	auc	auc_precision_recall	average_loss	label/mean	pos
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	

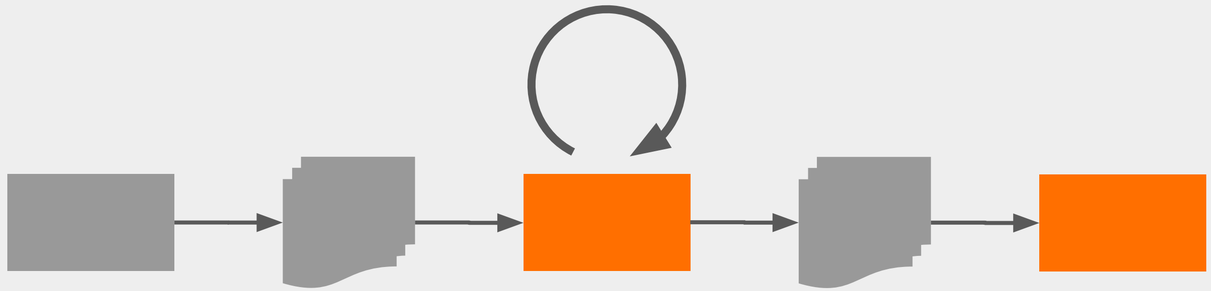
Metadata-Powered Functionality

Carry-over state from previous model runs

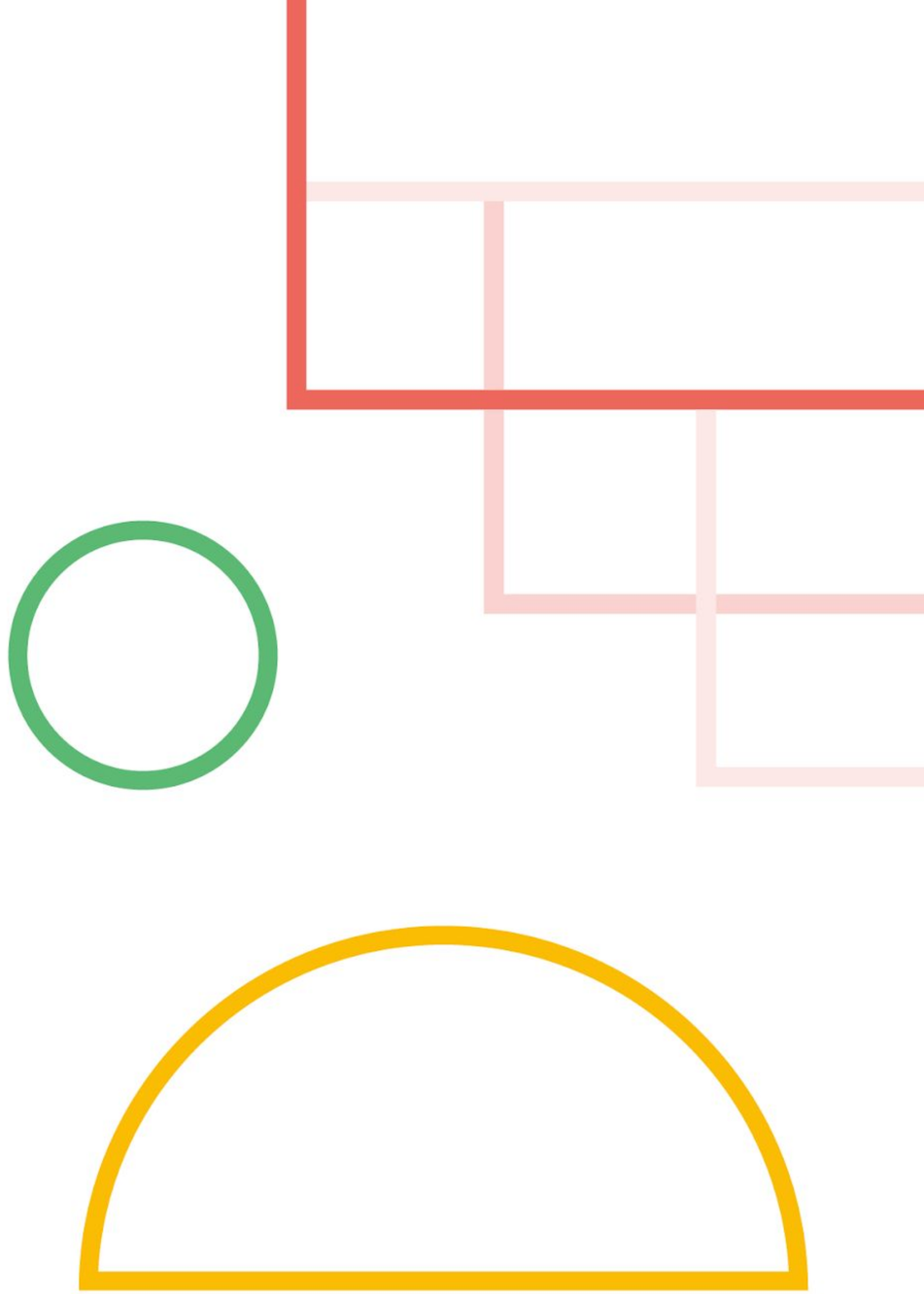


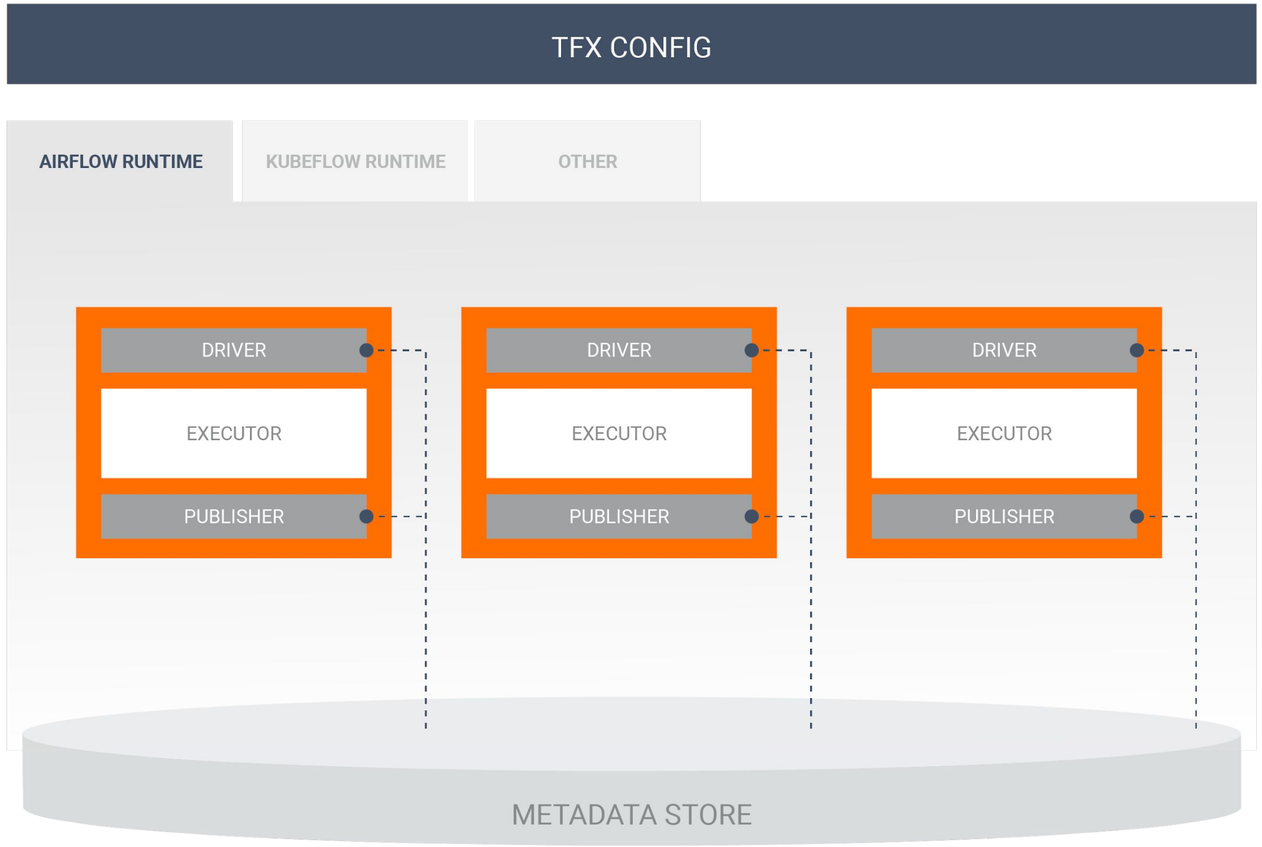
Metadata-Powered Functionality

Re-use previously computed outputs



TFX Orchestration



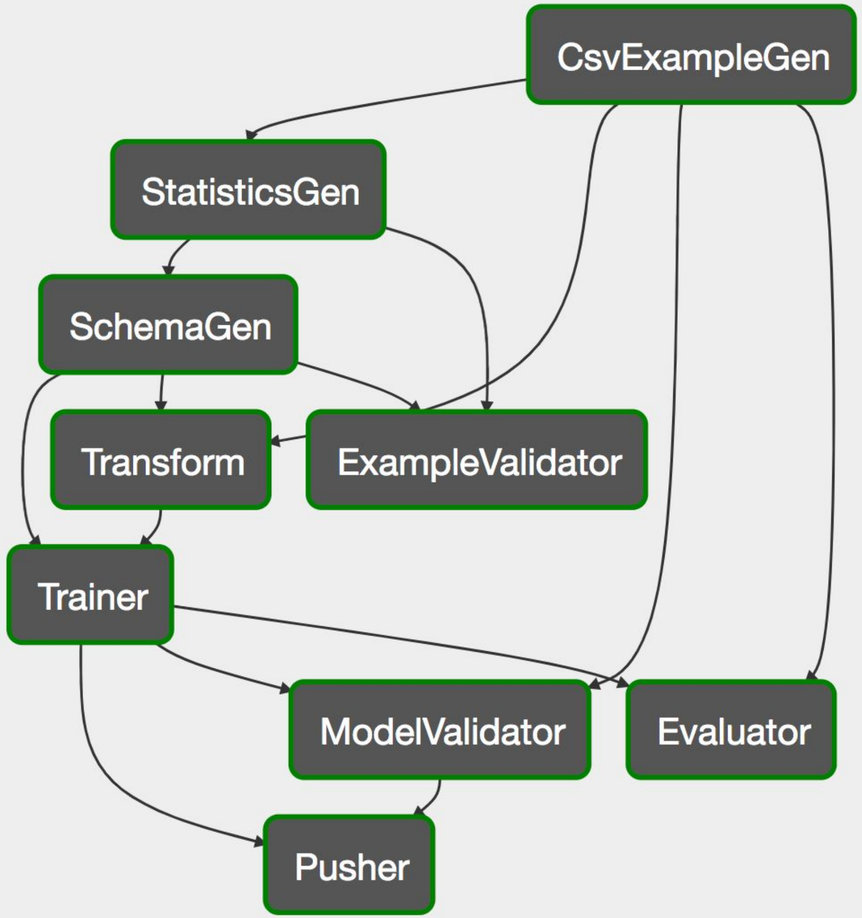


Bring your own Orchestrator

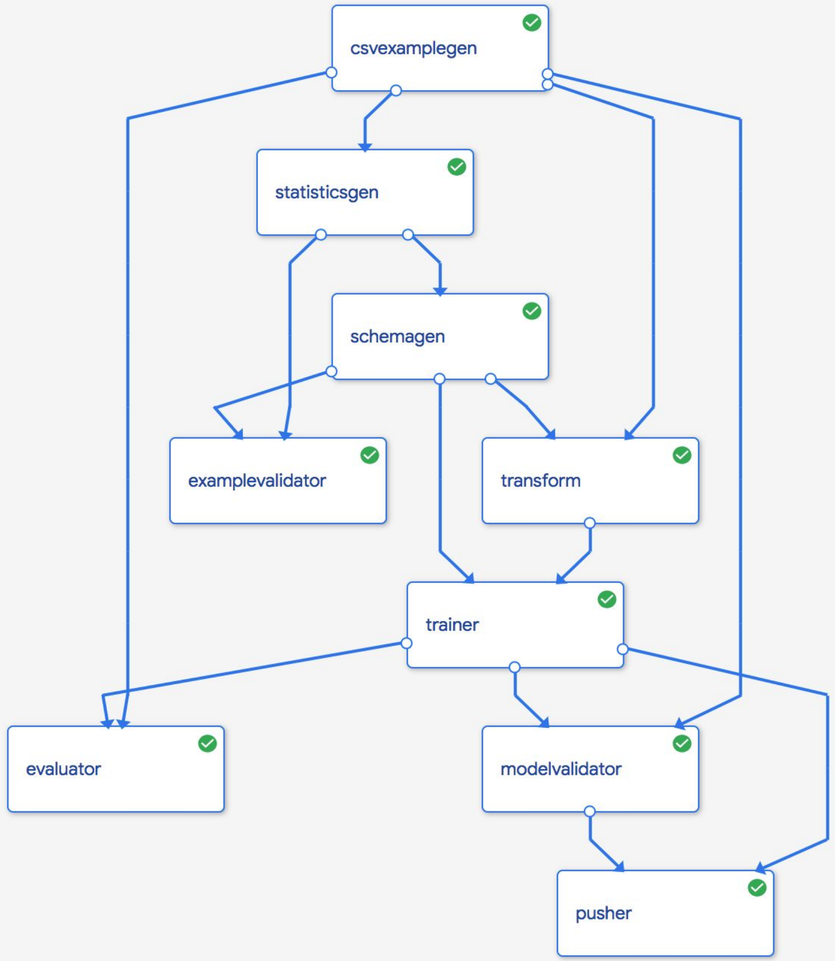
Flexible runtimes run components in the proper order using orchestration systems such as Airflow or Kubeflow

Orchestrators and DAGs

Airflow



Kubeflow Pipelines



TFX and Kubeflow

TensorFlow Extended (TFX)

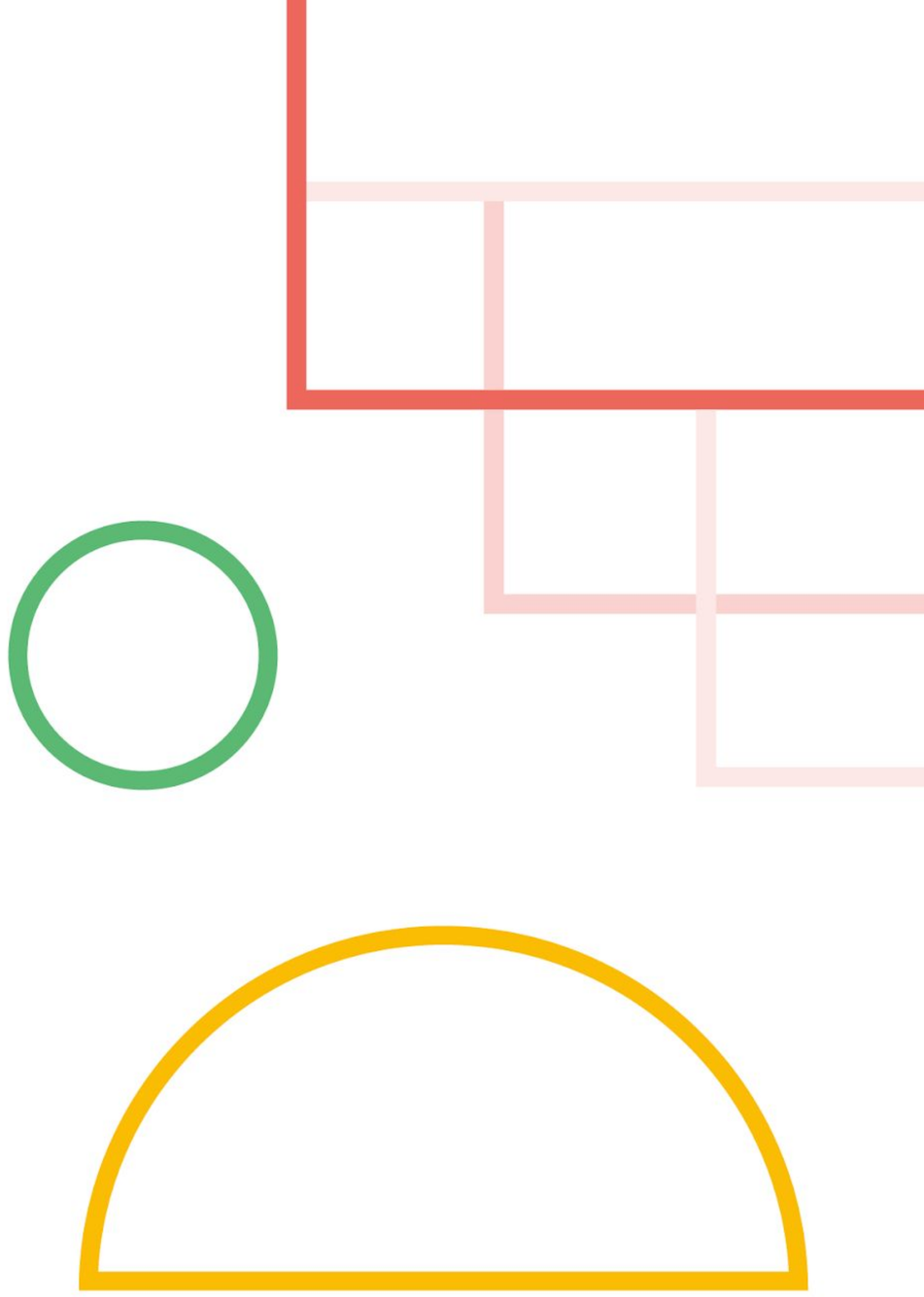
- Open-source version of what Google uses internally for Production ML
- Currently supported orchestrators:
  - Kubeflow
  - Apache Airflow
  - Apache Beam
  - We're adding more
  - You can add more

Kubeflow

- Open-source Production ML on Kubernetes
- Includes TFX
- Container set
- Management
- Monitoring
- Not just ML

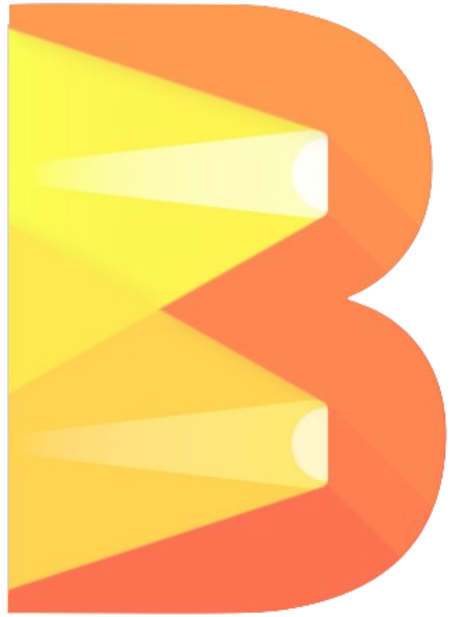
Distributed Pipeline Processing:

Apache Beam



What is Apache Beam?

- A unified **batch** and stream distributed processing API
- A set of **SDK frontends**: Java, **Python**, Go, Scala, SQL
- A set of **Runners** which can execute Beam jobs into various backends: **Local, Apache Flink**, Apache Spark, Apache Gearpump, Apache Samza, Apache Hadoop, **Google Cloud Dataflow**, ...



Apache Beam

Java

```
input.apply(
  Sum.integersPerKey())
```

Python

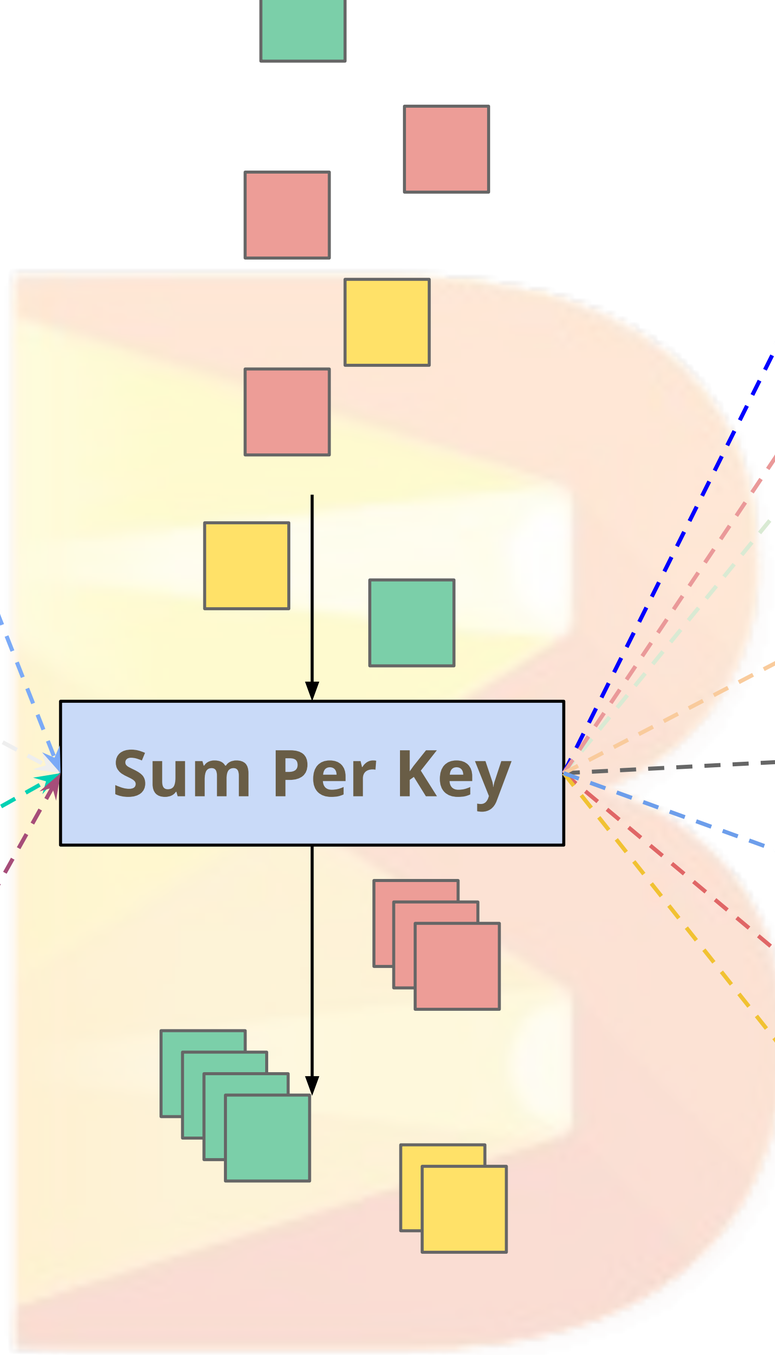
```
input | Sum.PerKey()
```

Go

```
stats.Sum(s, input)
```

SQL

```
SELECT key, SUM(value)
FROM input GROUP BY key
```



Cloud Dataflow

Apache Flink

Apache Spark

Apache Apex

Gearpump

IBM Streams

Apache Samza

Apache Nemo (incubating)

⋮

Beam Portability Framework

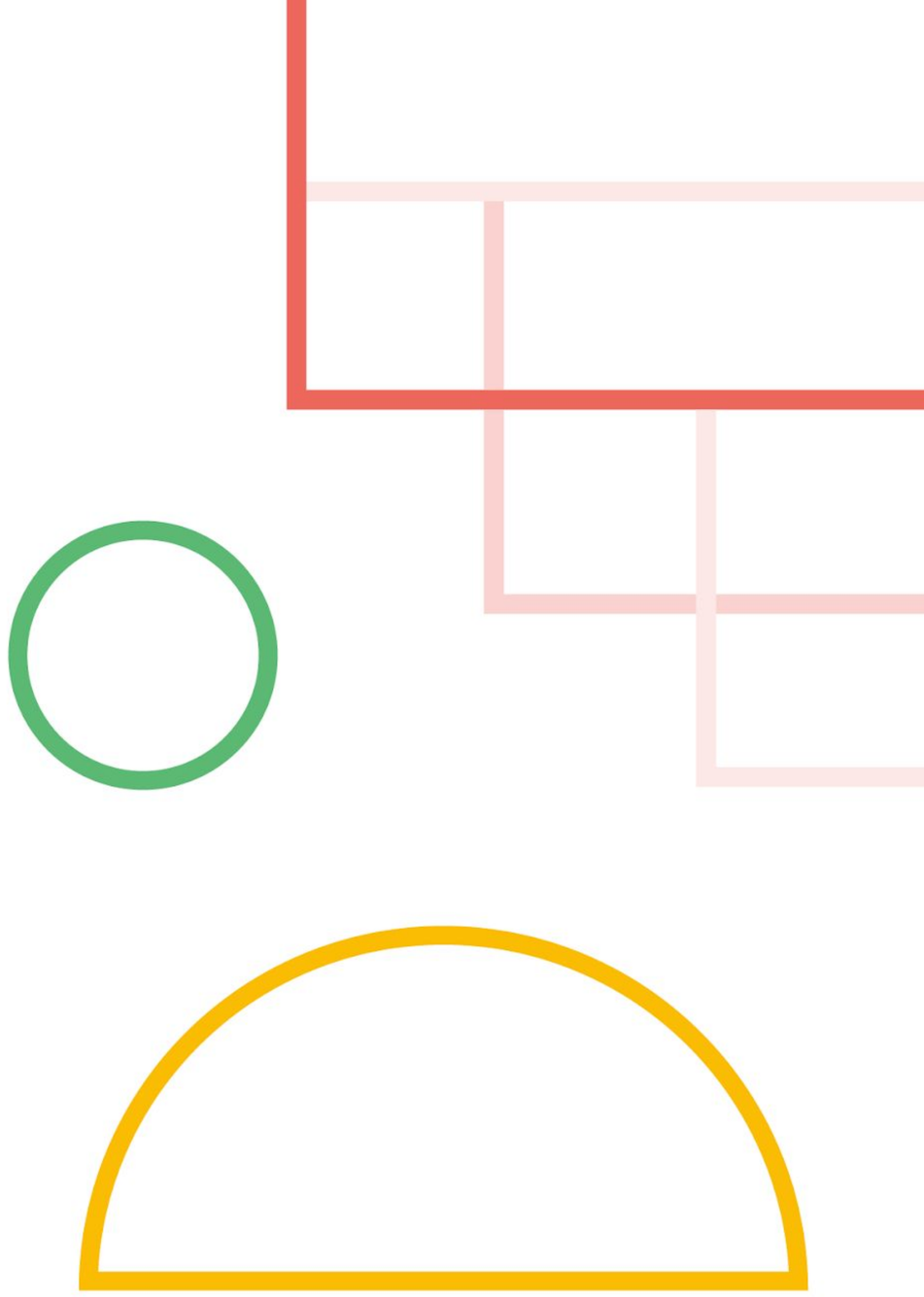
- **Currently most runners support the Java SDK only**
- Portability framework aims to provide full interoperability across the Beam ecosystem
- Portability API
  - Protobufs and gRPC for broad language support
  - Job submission and management: The Runner API
  - Job execution: The SDK harness
- **Python Flink and Spark runners use Portability Framework**

Beam Portability Support Matrix

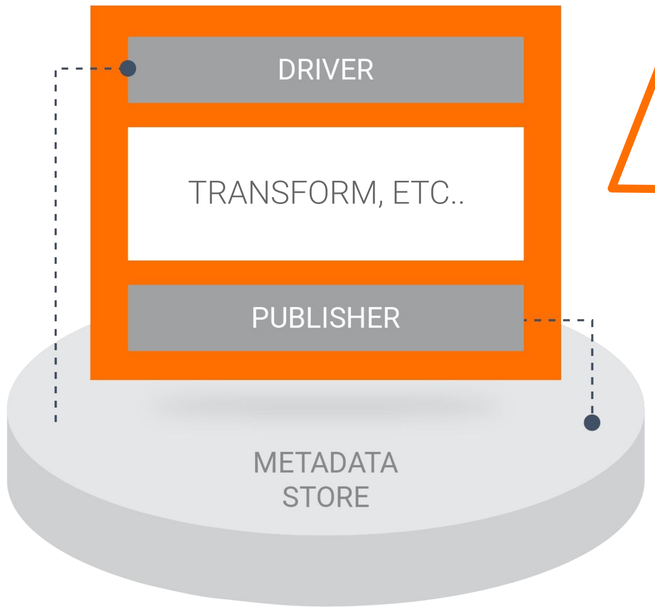
Apache Beam Portability Support Matrix			https://s.apache.org/apache-beam-portability-support-table					
			Flink (master)		**Spark (master)**		**Dataflow**	
			Python		Python		Python	
FEATURE			Batch	Streaming	Batch	Streaming	Batch	Streaming
	Impulse							
	ParDo							
		w/ side input						
		w/ multiple output						
		w/ user state					BEAM-2902	BEAM-2902
		w/ user timers						
		w/ user metrics						
	Flatten							
		w/ explicit flatten						
	Combine							
		w/ first-class rep						
		w/ lifting						
	SDF							
		w/ liquid sharding						
	GBK							
	CoGBK							
	WindowInto							
		w/ sessions						
		w/ custom windowfn						
Legend								
	Works, based on manual verification. Test desirable.							
BEAM-xxx	Partially works. Cell should contain JIRA.							
BEAM-xxx	Does not work. Cell should contain JIRA.							
	No information. To be evaluated.							

TFX Components:

What's in the box?

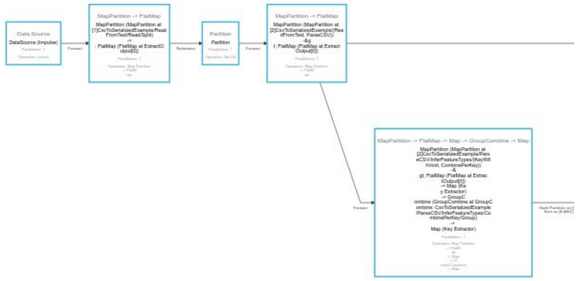


Executors do the work

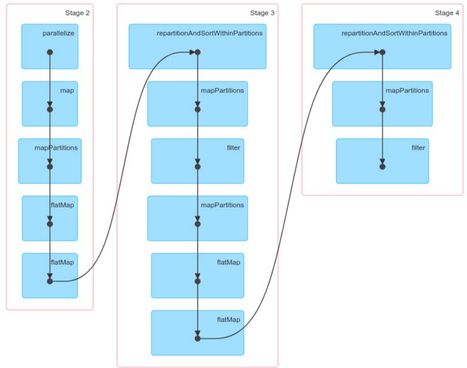


Beam

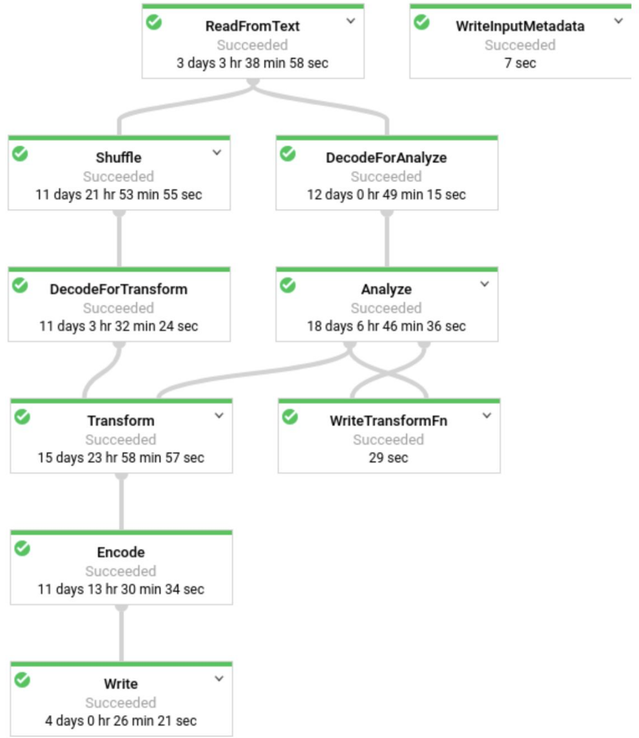
Flink



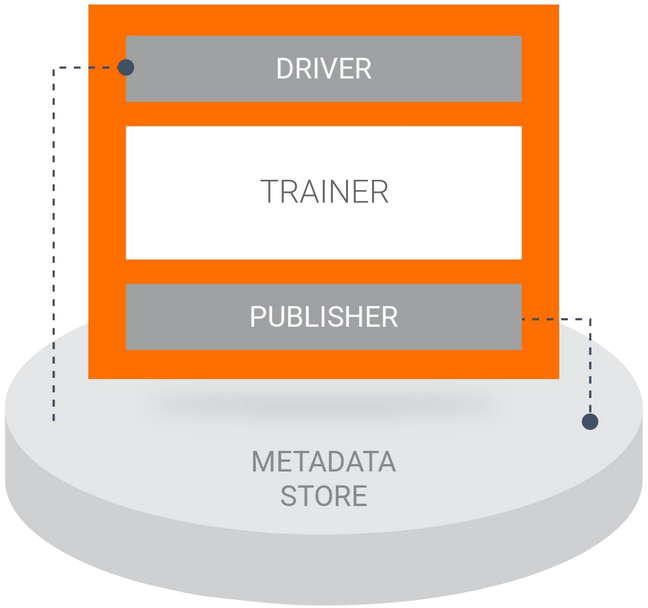
Spark

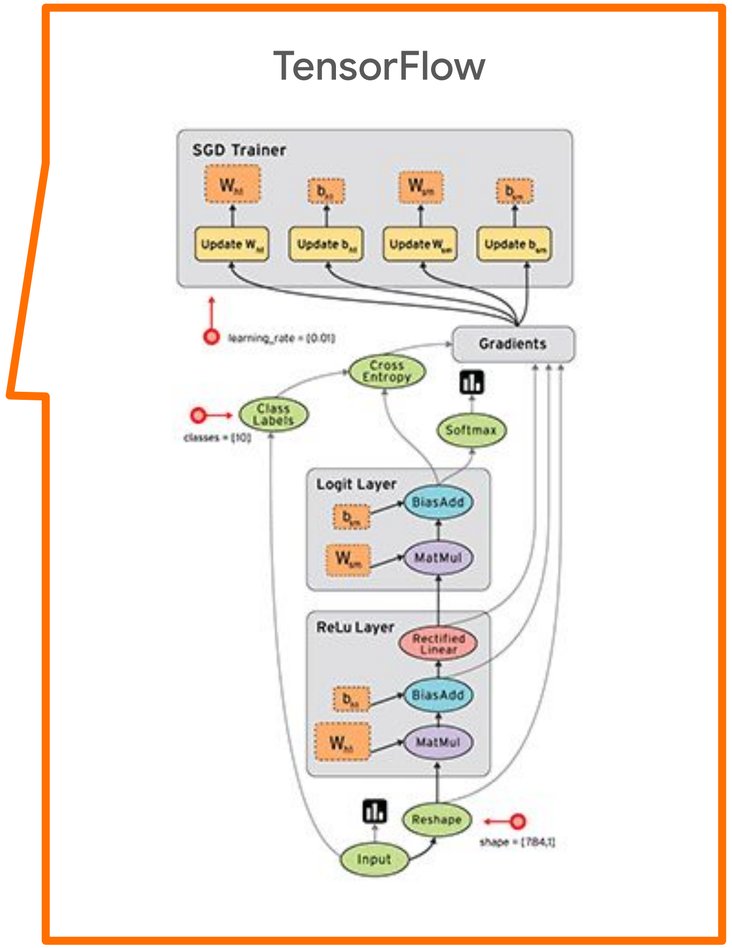


Dataflow

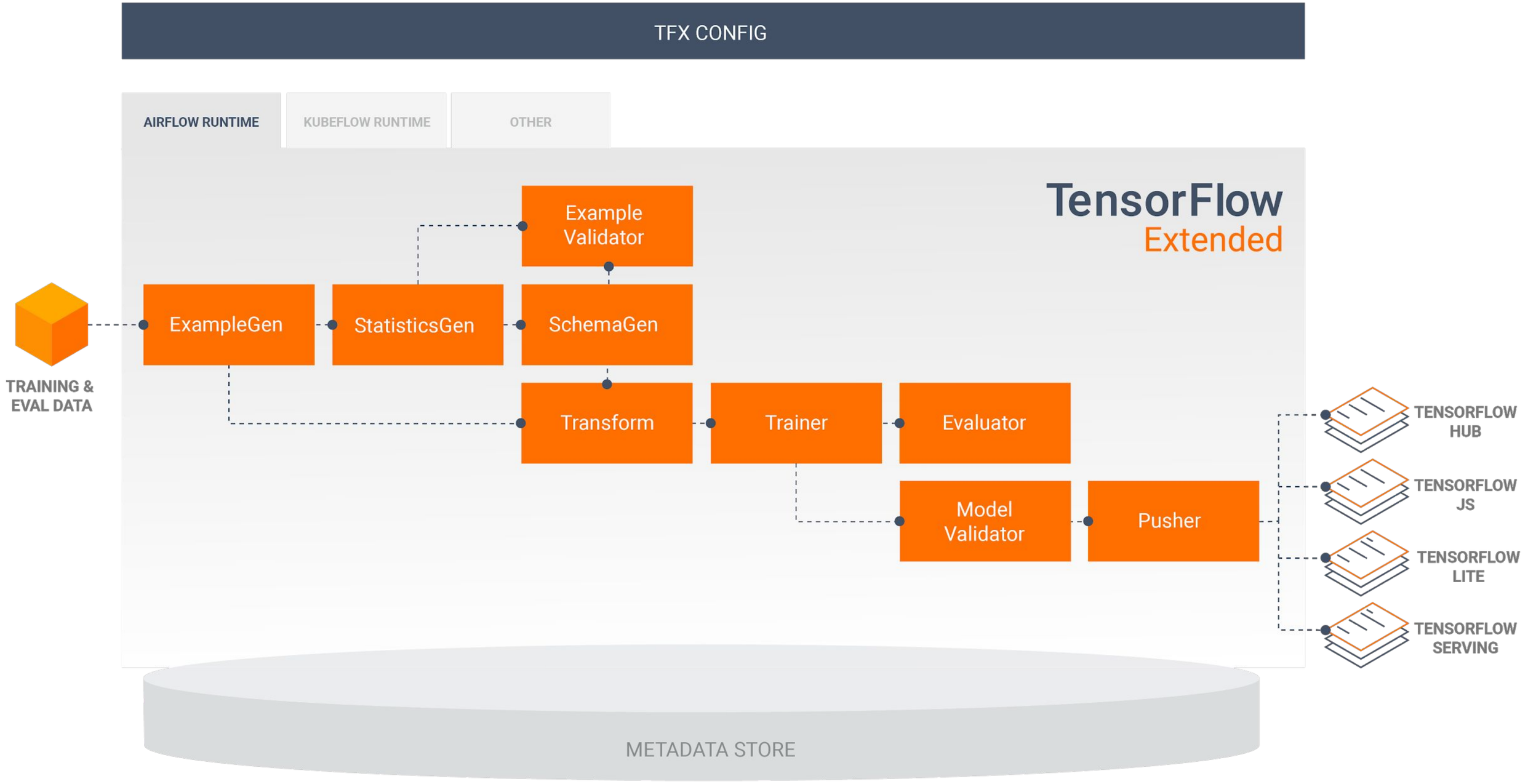


Executors do the work



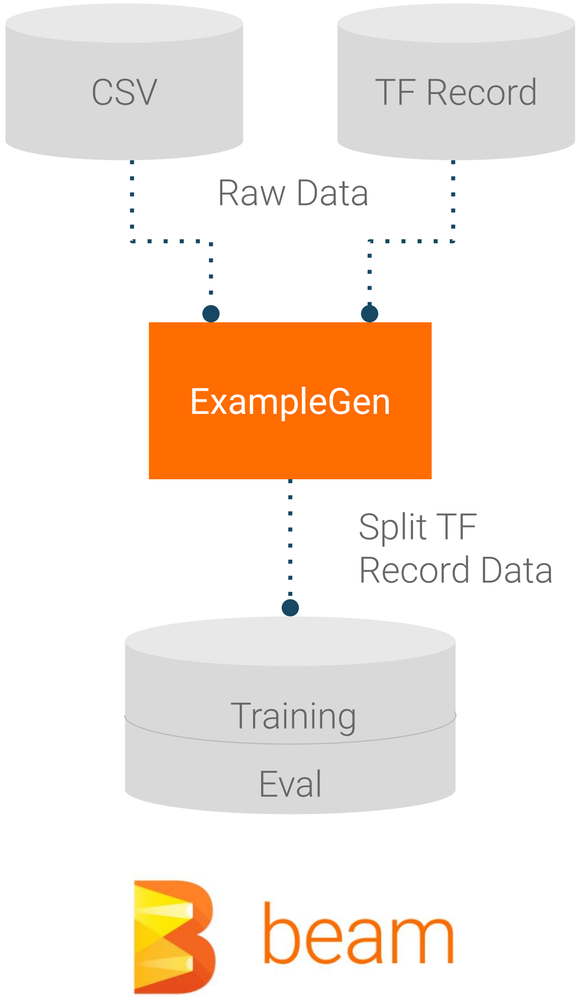


Executors do the work


TFX CONFIG
AIRFLOW RUNTIME
KUBEFLOW RUNTIME
OTHER
TensorFlow
Extended
Example Validator
ExampleGen
StatisticsGen
SchemaGen
TRAINING & EVAL DATA
Transform
Trainer
Evaluator
Model Validator
Pusher
TENSORFLOW HUB
TENSORFLOW JS
TENSORFLOW LITE
TENSORFLOW SERVING
METADATA STORE

Component: ExampleGen

Inputs and Outputs

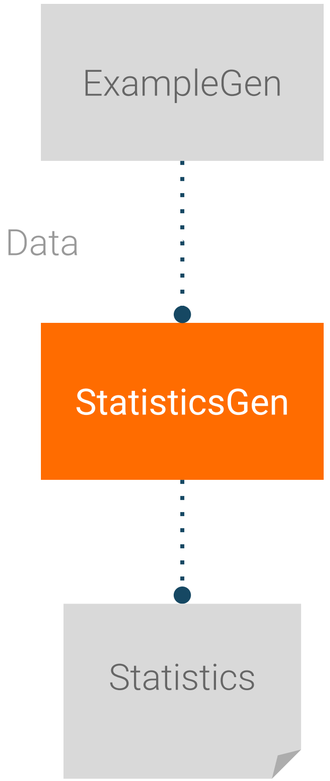


Configuration

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

example_gen = CsvExampleGen(input_base=examples)
```

Component: StatisticsGen

Inputs and Outputs

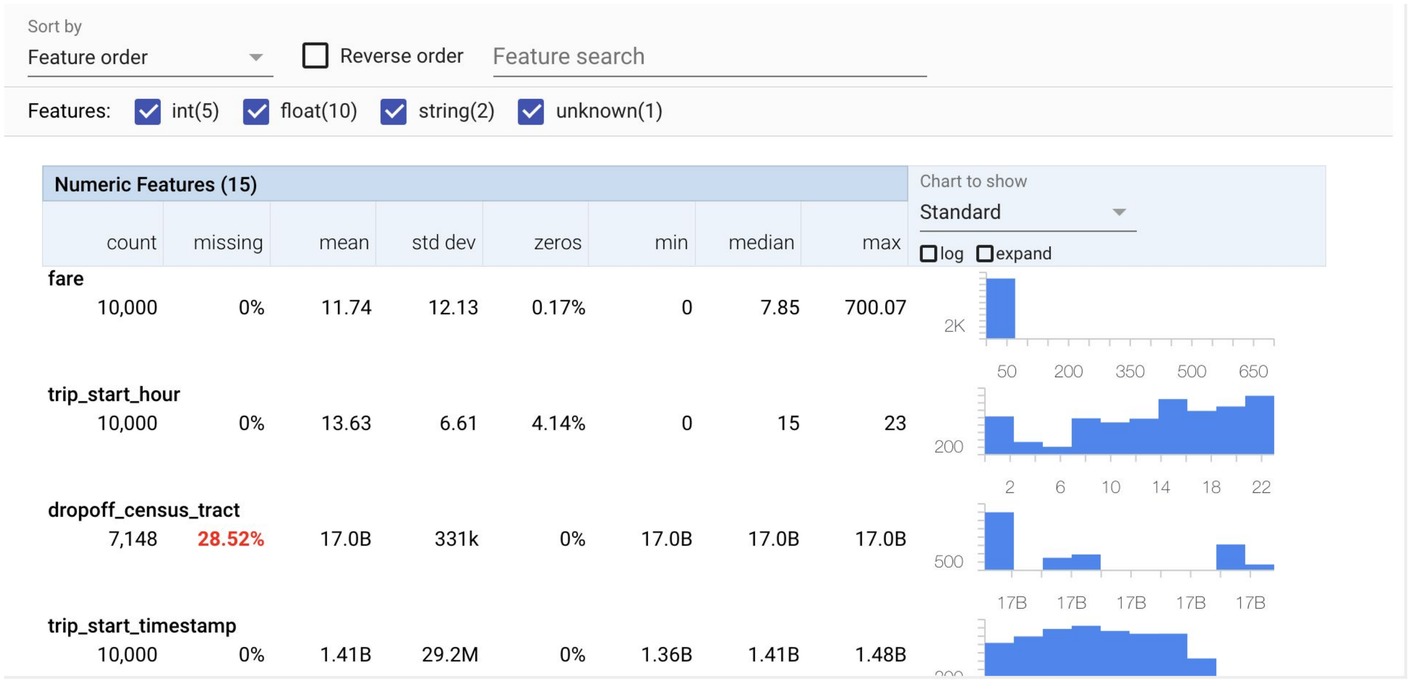


beam

Configuration

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

Visualization



Analyzing Data with TensorFlow Data Validation

Sort by

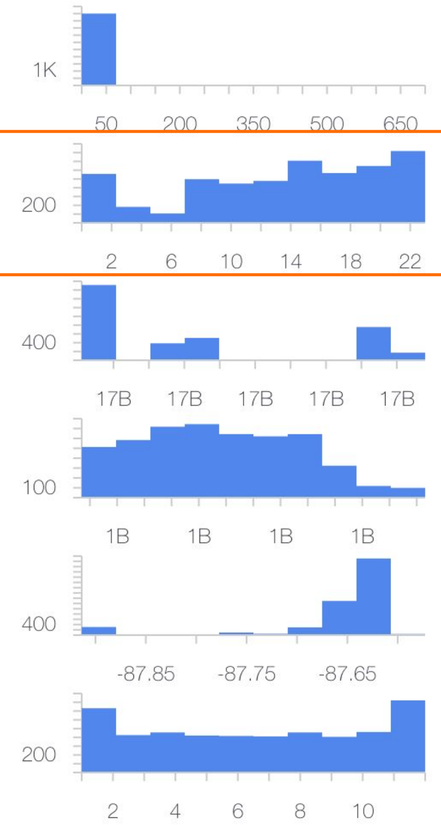
Feature order ▾ ☐ Reverse order Feature search (regex enabled)

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

Numeric Features (15)

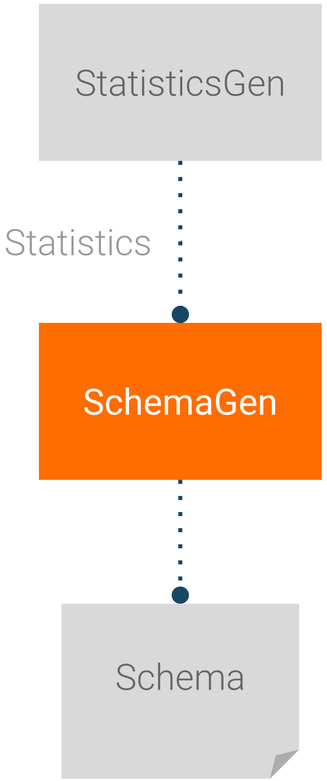
Chart to show: Standard ▾ ☐ log ☐ expand

	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



Component: SchemaGen

Inputs and Outputs



Configuration

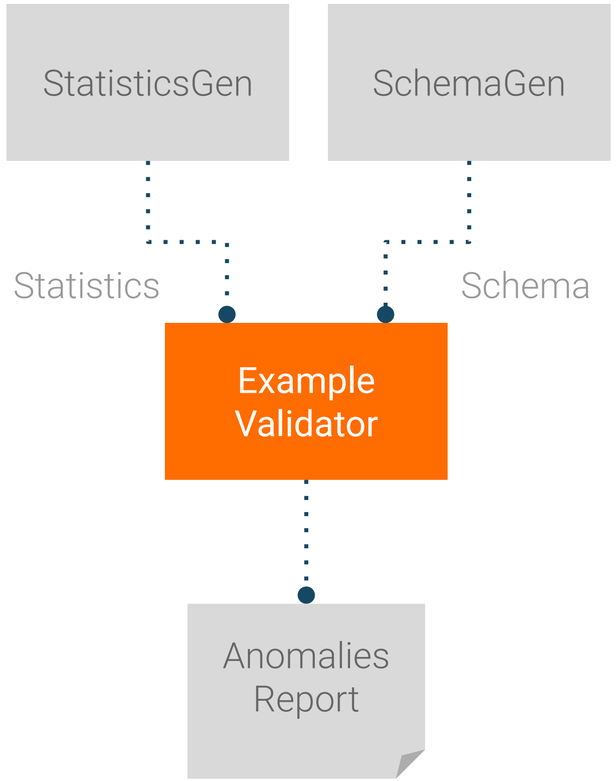
```
infer_schema = SchemaGen(stats=statistics_gen.outputs.output)
```

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'

Component: ExampleValidator

Inputs and Outputs



Configuration

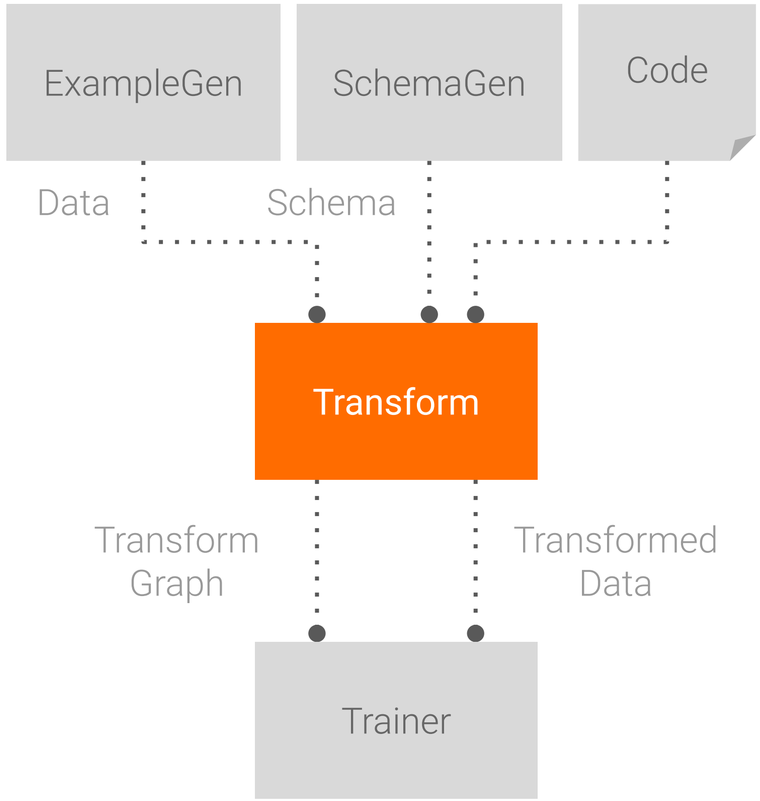
```
validate_stats = ExampleValidator(
    stats=statistics_gen.outputs.output,
    schema=infer_schema.outputs.output)
```

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%).

Component: Transform

Inputs and Outputs



beam

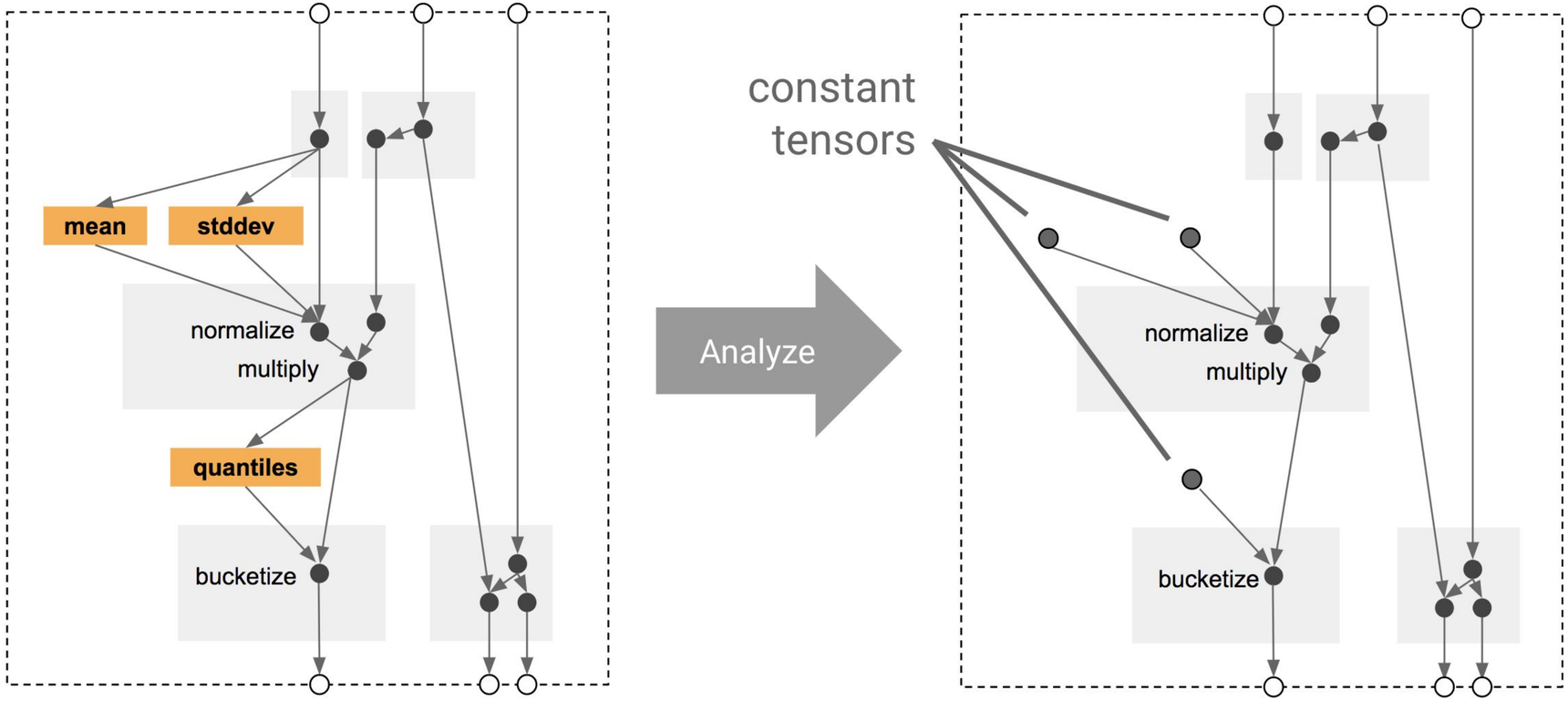
Configuration

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

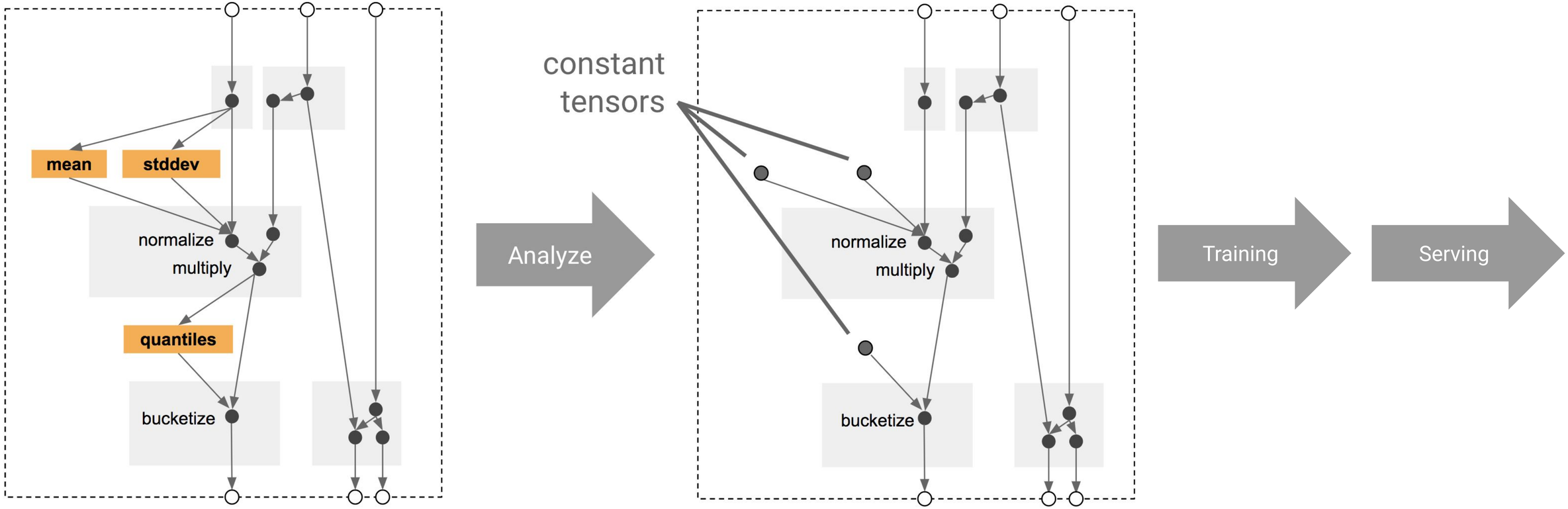
Code

```
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))

# ...
```

Using TensorFlow Transform for Feature Engineering

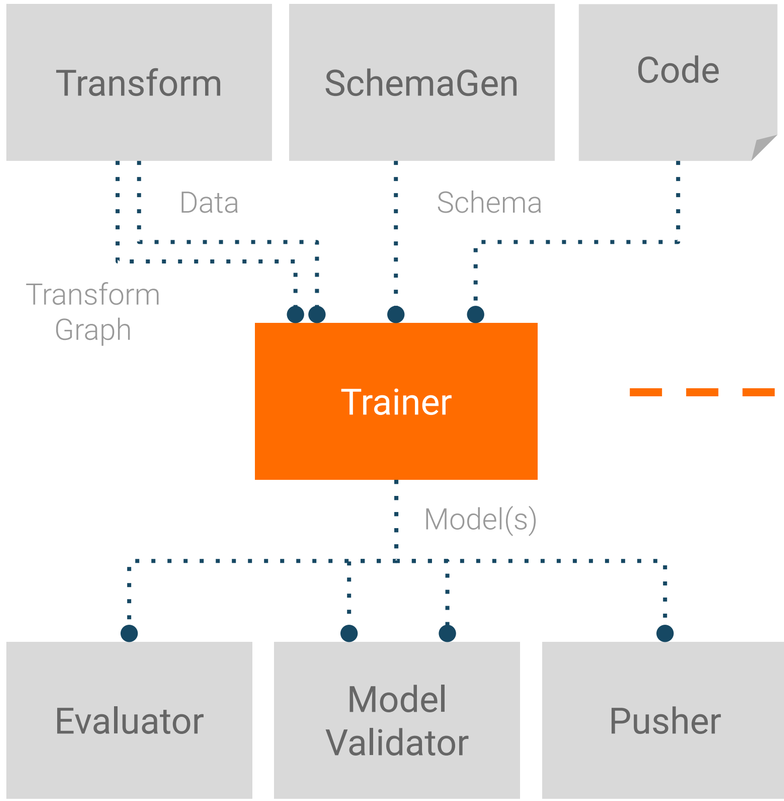


Using TensorFlow Transform for Feature Engineering



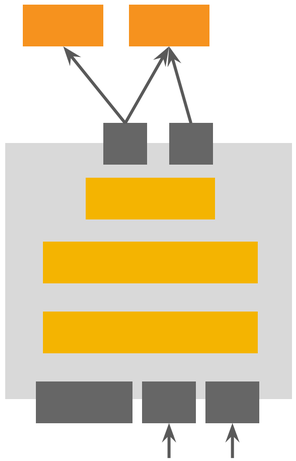
Component: Trainer

Inputs and Outputs



Highlight: SavedModel Format

Train, Eval, and Inference Graphs



EvalSavedModel

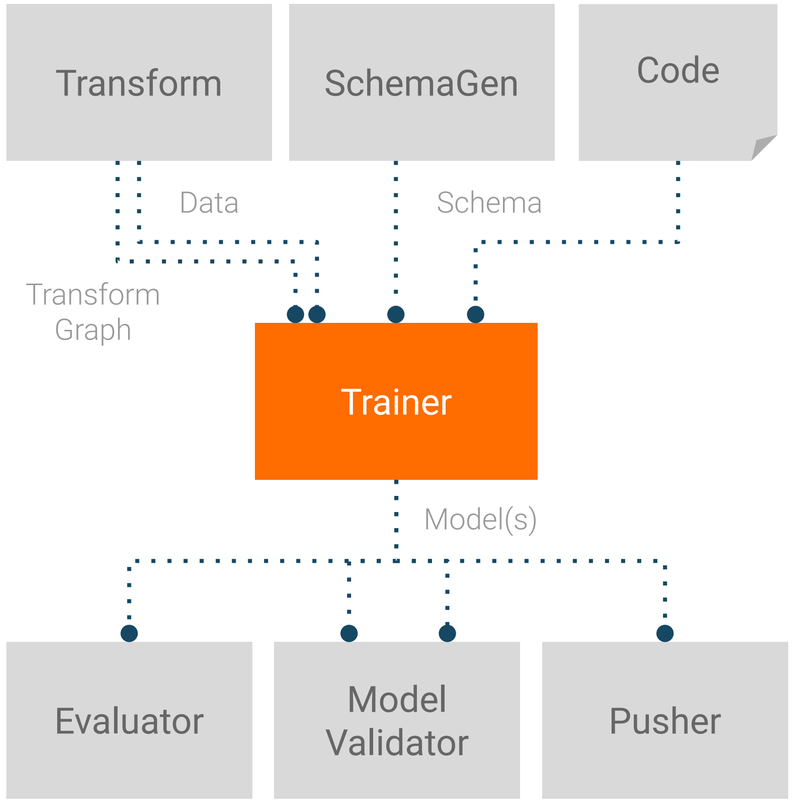
SavedModel

TensorFlow Model Analysis

TensorFlow Serving

Component: Trainer

Inputs and Outputs



Configuration

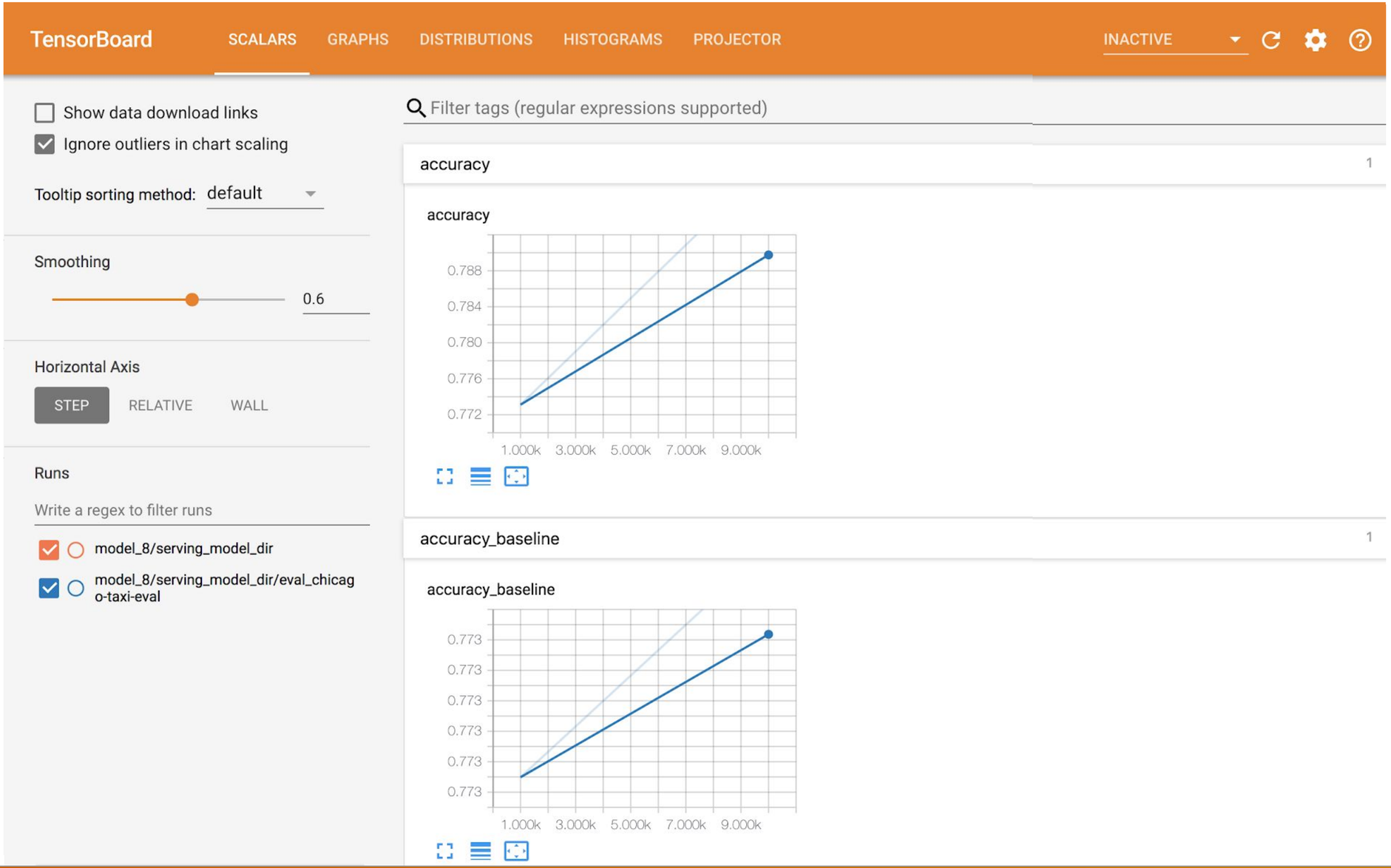
```
trainer = Trainer(
    module_file=taxi_module_file,
    transformed_examples=transform.outputs.transformed_examples,
    schema=infer_schema.outputs.output,
    transform_output=transform.outputs.transform_output,
    train_steps=10000,
    eval_steps=5000,
    warm_starting=True)
```

Code

Just TensorFlow :)

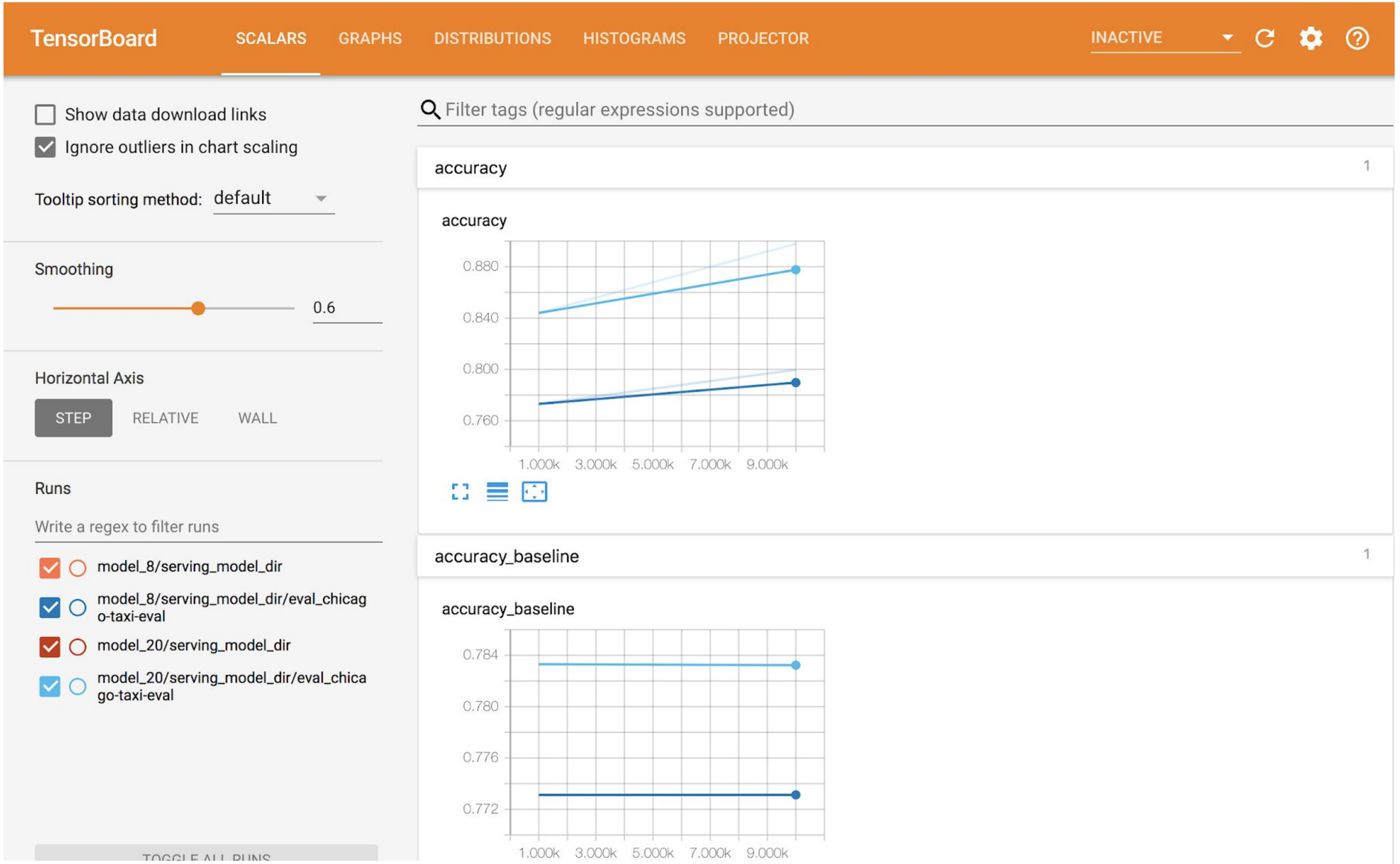
```
# Open up Tensorboard for model_id.
print(display_tensorboard(model_id))
```

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



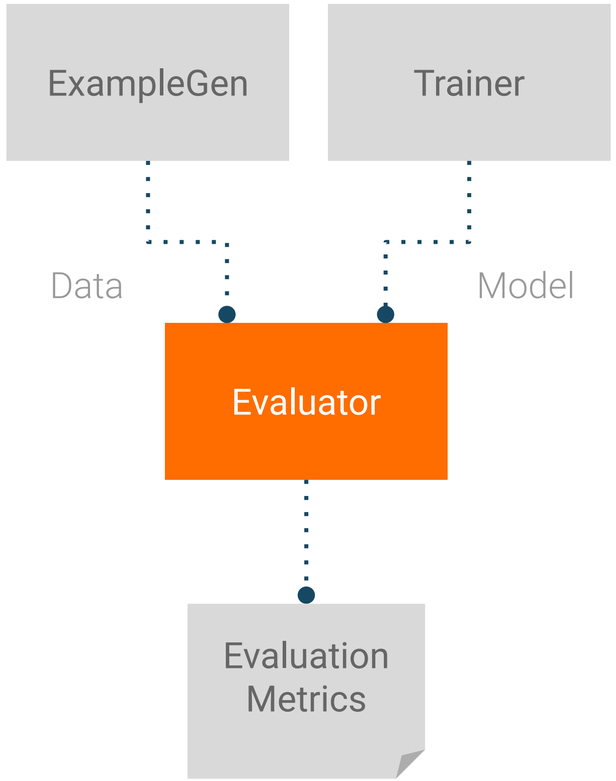
```
# Compare Tensorboard metrics for different models.
if num_models > 1:
    print(display_tensorboard(model_id, other_model_id=other_model_id))
```

```
http://your.host.name:53230
```



Component: Evaluator

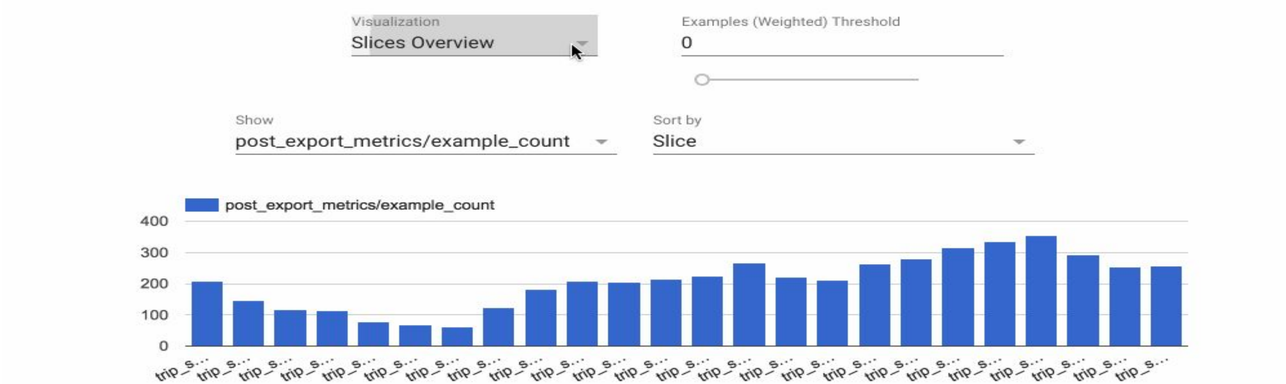
Inputs and Outputs



Configuration

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

Visualization

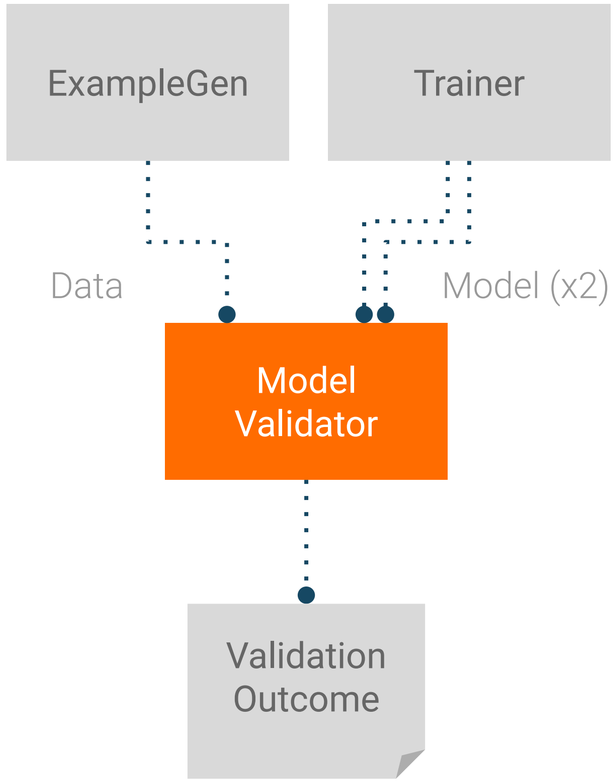


feature	accuracy	accuracy_baseline	auc	auc_precision_recall	average_loss
trip_start_hour:19	0.63582	0.59104	0.64311	0.56092	0.64626
trip_start_hour:14	0.67117	0.65766	0.63793	0.49112	0.61667
trip_start_hour:2	0.66102	0.63559	0.58527	0.47002	0.65236
trip_start_hour:12	0.69643	0.65625	0.68270	0.54122	0.59538
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

beam

Component: ModelValidator

Inputs and Outputs



Configuration

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

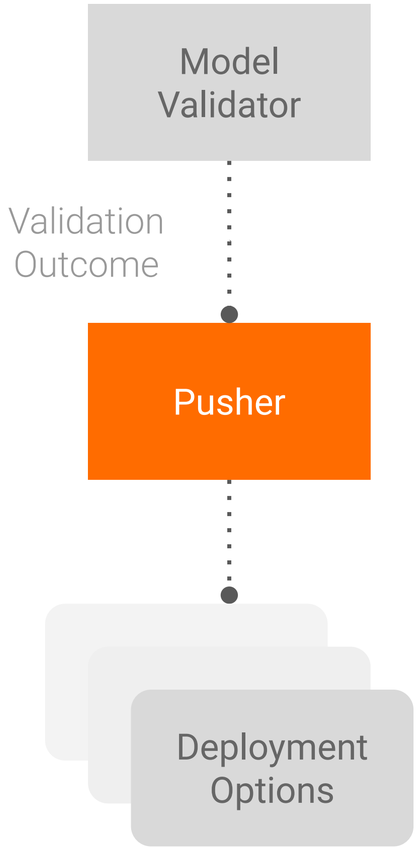
Configuration Options

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

beam

Component: Pusher

Inputs and Outputs



Configuration

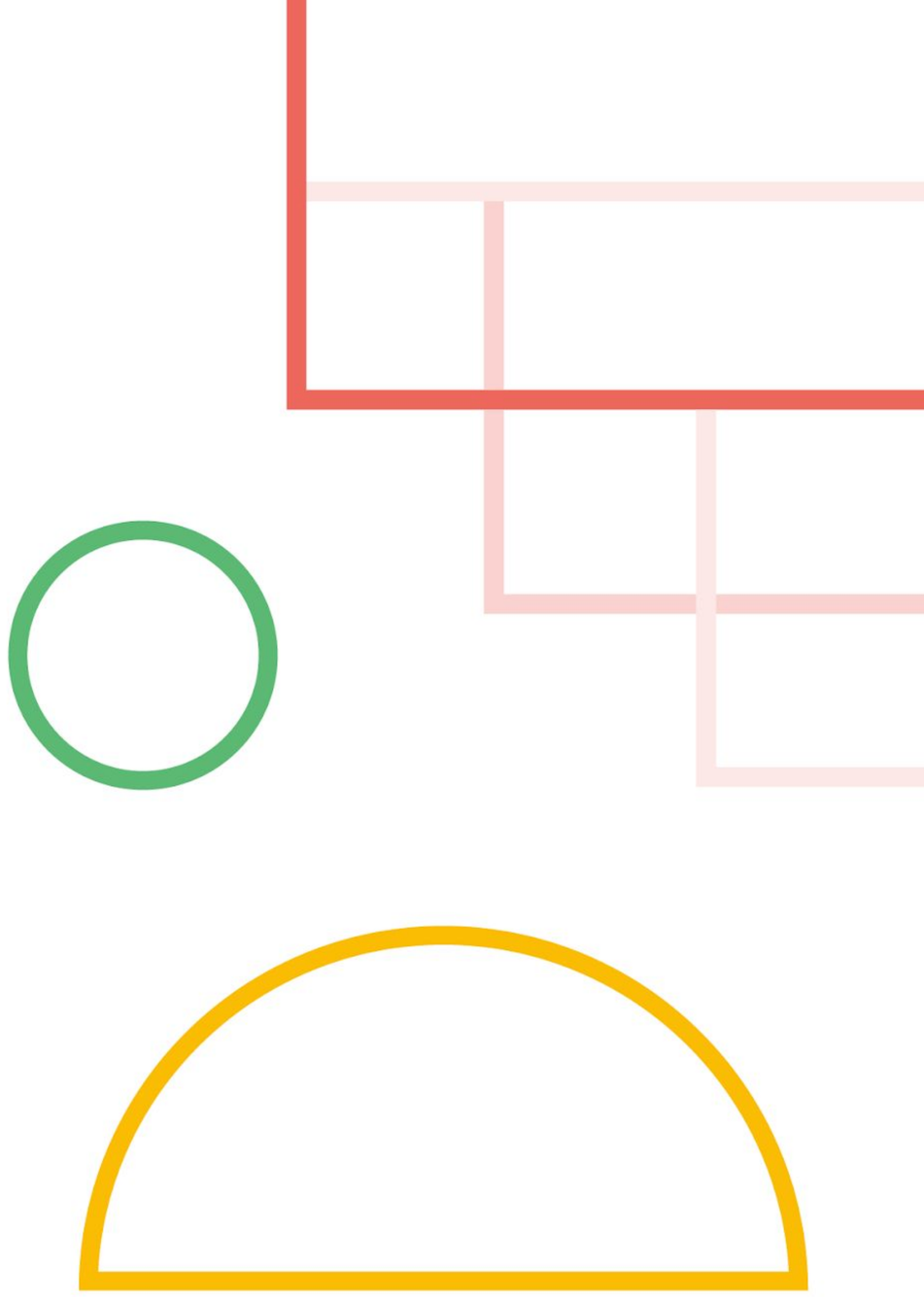
```
pusher = Pusher(
    model_export=trainer.outputs.output,
    model_blessing=model_validator.outputs.blessing,
    serving_model_dir=serving_model_dir)
```

Block push on validation outcome

Push destinations supported today

- Filesystem (TensorFlow Lite, TensorFlow JS)
- TensorFlow Serving

Model Understanding



Online Retailer Selling Shoes ...

Your model predicts **click-through rates (CTR)**, helping you decide how much inventory to order.

When all-of-a-sudden!

You discover that AUC and prediction accuracy have **dropped** on men's dress shoes!

Why “Understand” the model?

Mispredictions do not have uniform **cost** to your business.

The **data you have** is rarely the data you wish you had.

Model objective is nearly always a **proxy** for your business objectives

The real world **doesn’t stand still**.

ML Insights Triangle

ML Insights Triangle

Some **assumption** was violated, but which one?

Business Realities Changed?

Business Realities Changed?

Bad Data?

Business Realities Changed?

Model Needs Improvement?

Bad Data?

First Things First

Check your data with the ExampleValidator component and the tools in TensorFlow Data Validation:

- No outliers
- No missing features
- Minimal distribution shift

Sort by

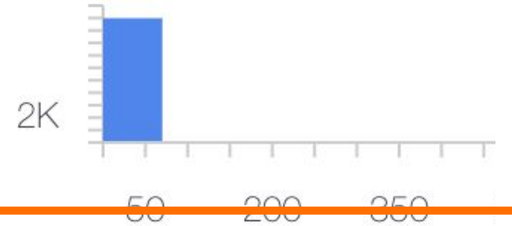
Feature order ▾ ☐ Reverse order Feature search

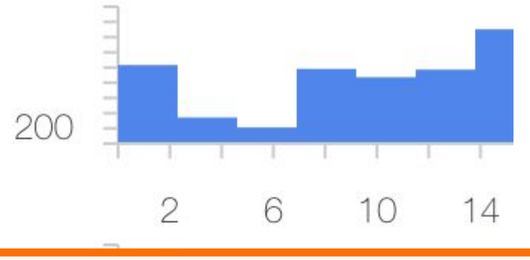
Features: ☑ int(5) ☑ float(10) ☑ string(2) ☑ unknown(1)

Numeric Features (15)

Chart to show: Standard ▾ ☐ log ☐ expand

	count	missing	mean	std dev	zeros	min	median	max
price	10,000	0%	11.74	12.13	0.17%	0	7.85	700.07
shoe_size	10,000	0%	13.63	6.61	4.14%	0	15	23





Feature Attributions

Query for other examples by matching on those important features

- Maybe the model overgeneralized from too few examples with this particular feature combo?
- Add features to help create distinctions you'd like the model to make.
- Collect more examples with that feature combo if possible!

Analyze and Compare

Check your model performance with the built-in TF Model Analysis component:

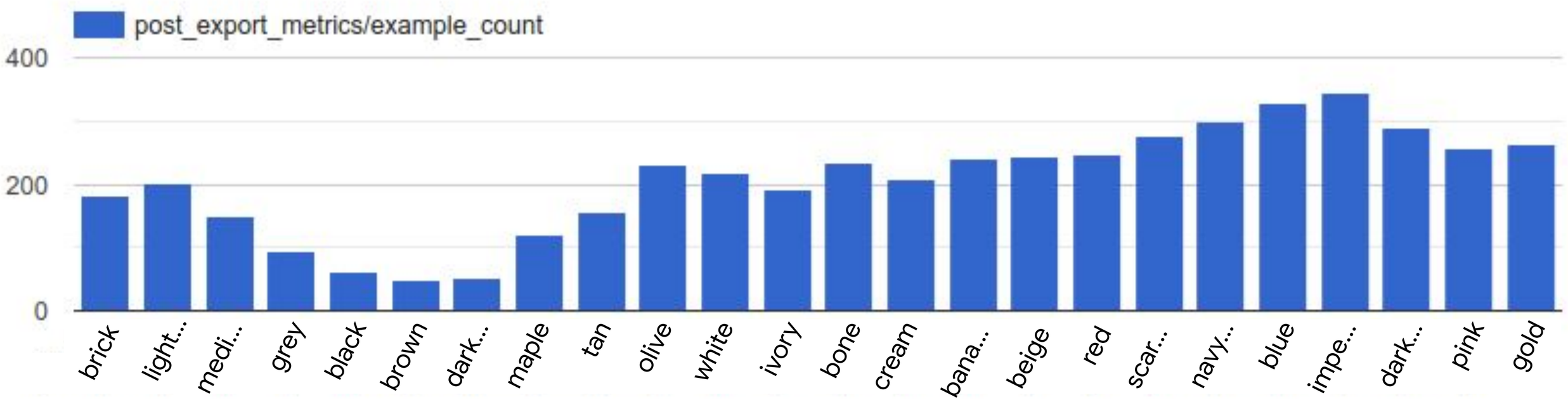
- How does the model perform on different slices of data?
- How does the current model performance compare to previous versions?

Show

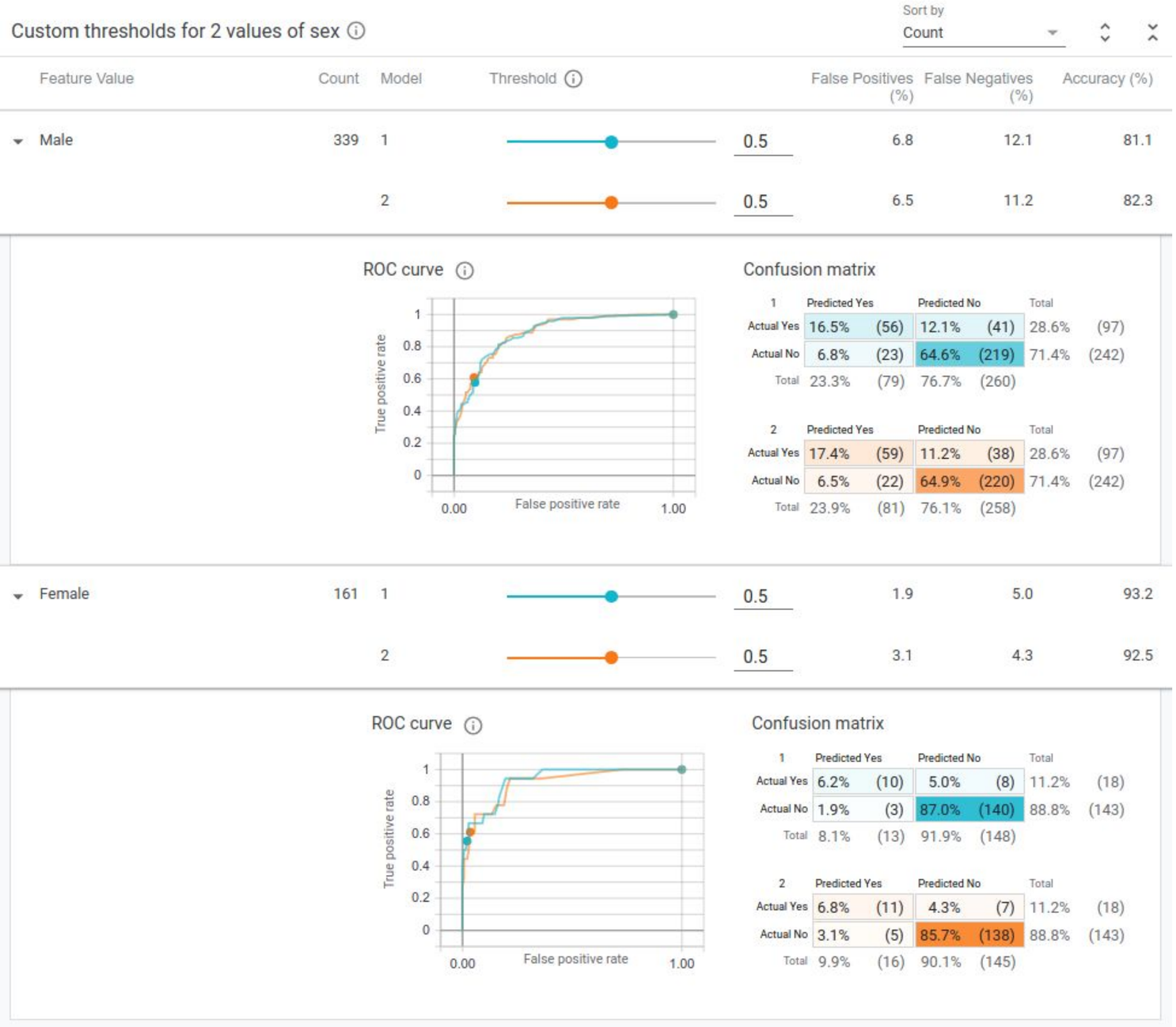
post_export_metrics/example_count

Sort by

Slice



feature	accuracy	accuracy_baseline	auc	auc_precision_recall	average_lo
brick	0.74586	0.74033	0.95943	0.80808	0.358
light grey	0.79310	0.78325	0.95648	0.74818	0.324



Explore your model and data

What-if tool

Understand the input your model is receiving

Ask and answer “what-if” questions about your model’s output

Compare model performance across different slices of your data

Compare performance across multiple models

Quantify the Cost

Remember, **CTR is just the model's proxy objective**!

- Your actual business objectives depend on: revenue, cost, your supply, etc.
- To analyze **misprediction cost**, join your model's predictions with the rest of your business data

TensorFlow Extended (TFX)

Out-of-the-box components for your production model needs

Flexible orchestration and metadata

Extensible with custom components

Visit us at https://tensorflow.org/tfx and show us how you've used and extended TFX!

TFX | TensorFlow

https://www.tensorflow.org/tfx

TensorFlow Install Learn API Resources Community Why TensorFlow Search GitHub

For Production

Overview Tutorials Guide API

Apply to speak at TensorFlow World. Deadline April 23rd. Propose talk

TensorFlow Extended (TFX) is an end-to-end platform for deploying production ML pipelines

When you're ready to move your models from research to production, use TFX to create and manage a production

https://www.tensorflow.org/tfx

Thank you!

Helpful resources

Web	https://tensorflow.org/tfx
Repo	https://github.com/tensorflow/tfx
Community	http://bit.ly/tfx-forum

Robert Crowe
TensorFlow Developer Advocate

@robert_crowe

Demo

You can run it too!

Developer Tutorial:

https://www.tensorflow.org/tfx/tutorials/tfx/airflow_workshop

Please rate this session

TFX: Production ML pipelines with TensorFlow

Robert Crowe (Google)
11:55am–12:35pm Wednesday, September 11, 2019
Location: 230 A
Implementing AI

Secondary topics: Deep Learning tools

RATE THIS SESSION

Who is this presentation for?

- Data scientists, machine learning engineers, ML Ops, ML management, and DevOps

https://goo.gle/AIC-SJ-TFX