TensorFlow

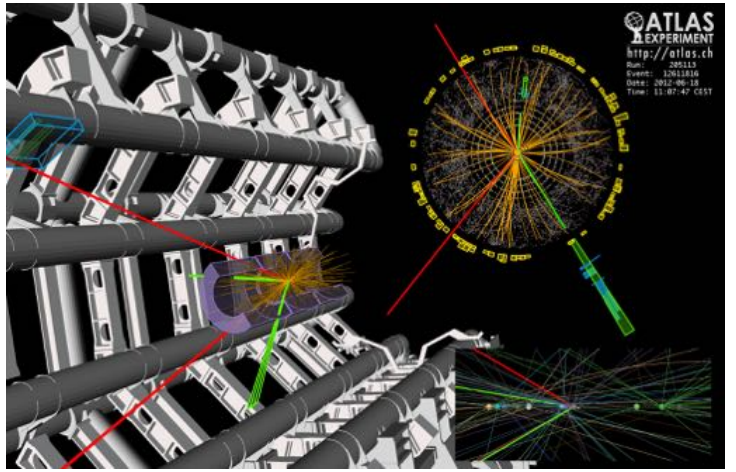
TensorFlow Ecosystem

Machine Learning: From Development To Production

Fabien Tarrade
Sr. Data Scientist and Applied ML Scientist
@fabtar

About me

- Sr. Data Scientist and Applied ML Scientist at AXA
- Working on Machine Learning & NLP at scale
  - Machine Learning pipeline
  - Deployment on the Cloud
- Particle physicist for 10 years @CERN
  - Big Data (Petabyte)
  - Statistical analyses
- Contact me:
  https://twitter.com/fabtar
  https://www.linkedin.com/in/fabientarrade/



Disclaimer

- I am not working for Google
- This is not an official talk about TensorFlow Ecosytem
- Views are my own

- My interest is running ML in production and at scale
- I will only cover some parts [highly bias selection]

- Always refer to the official documentation [latest information]
- A lot of new features/improvements/tools every few months

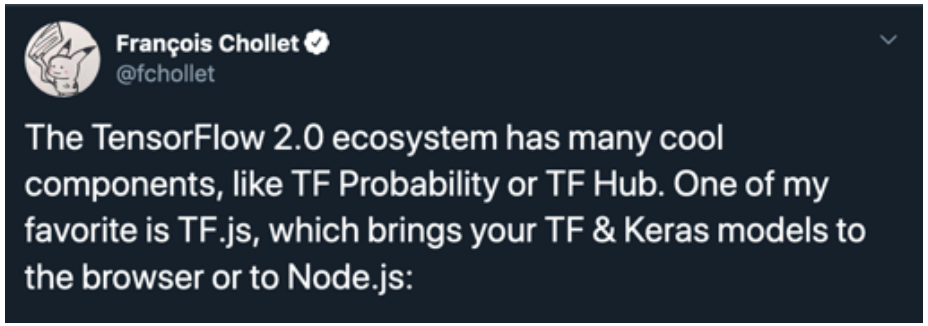
- Collected a lot of material from various sources [TF documentation, Googlers, GDE, Blogs, videos .., see the full list of references at the end]

Agenda

local dev

- High level API with **Keras/Tensorflow**
- Consumption of reusable parts of ML models **TF Hub**
- Monitoring with **TensorBoard**
- Easy Scaling with TensorFlow
- High-Performance ML with GPU/Cloud TPUs
- TensorFlow Extended **TFX**

production

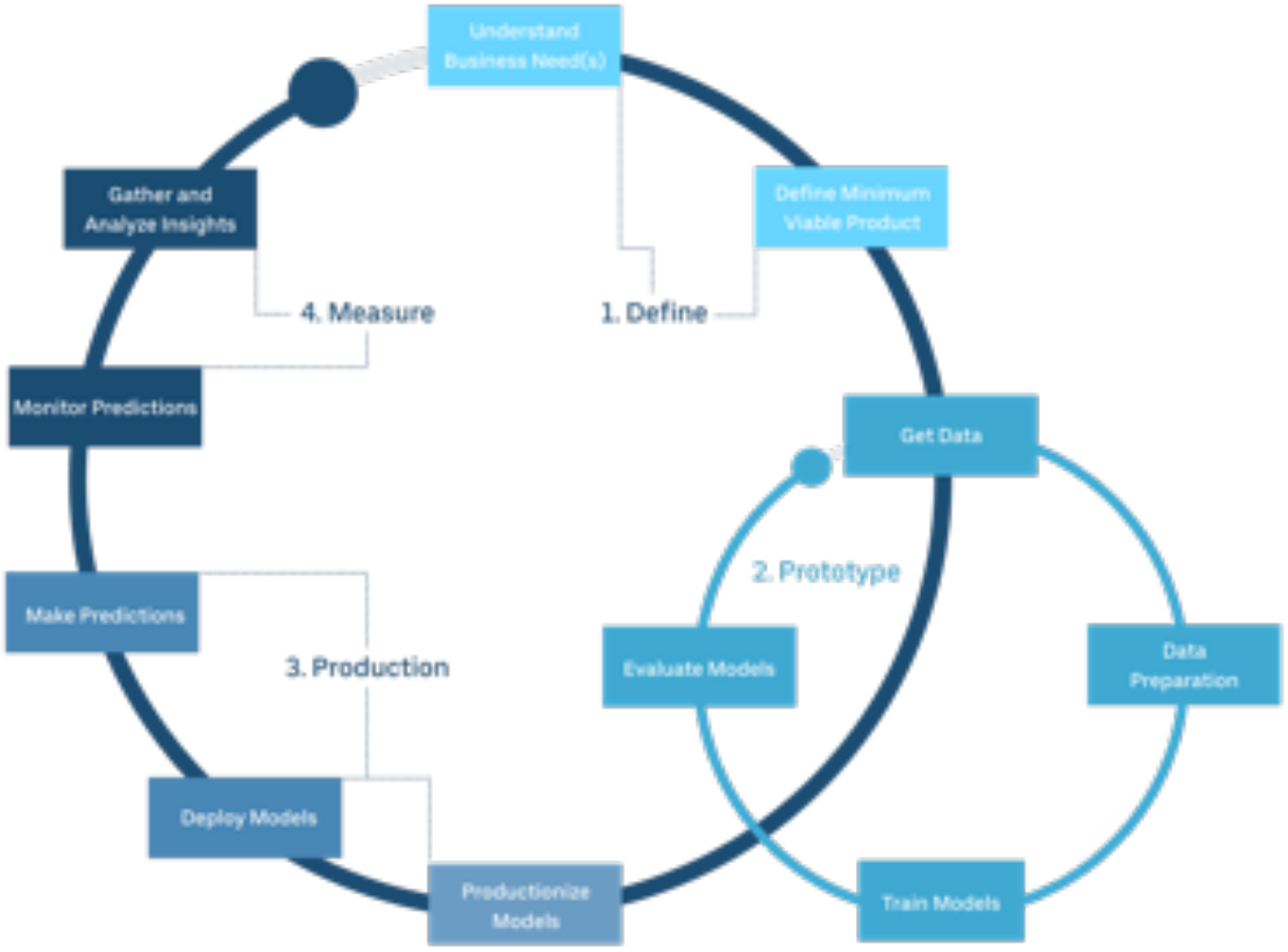


Everything is with TensorFlow 2.X except TFX (TensorFlow<=1.15)

TensorFlow

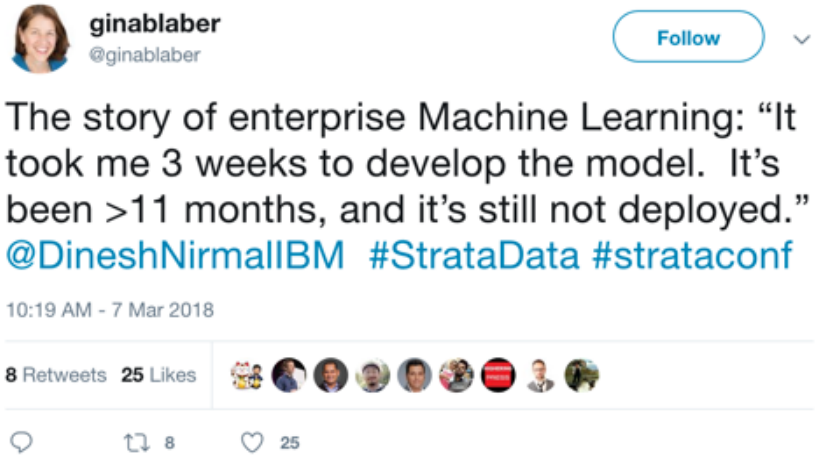
Machine Learning workflow and pipelines

Workflow of a ML Project



Uber Machine Learning

https://eng.uber.com/scaling-michelangelo/



ginablaber
@ginablaber

Follow

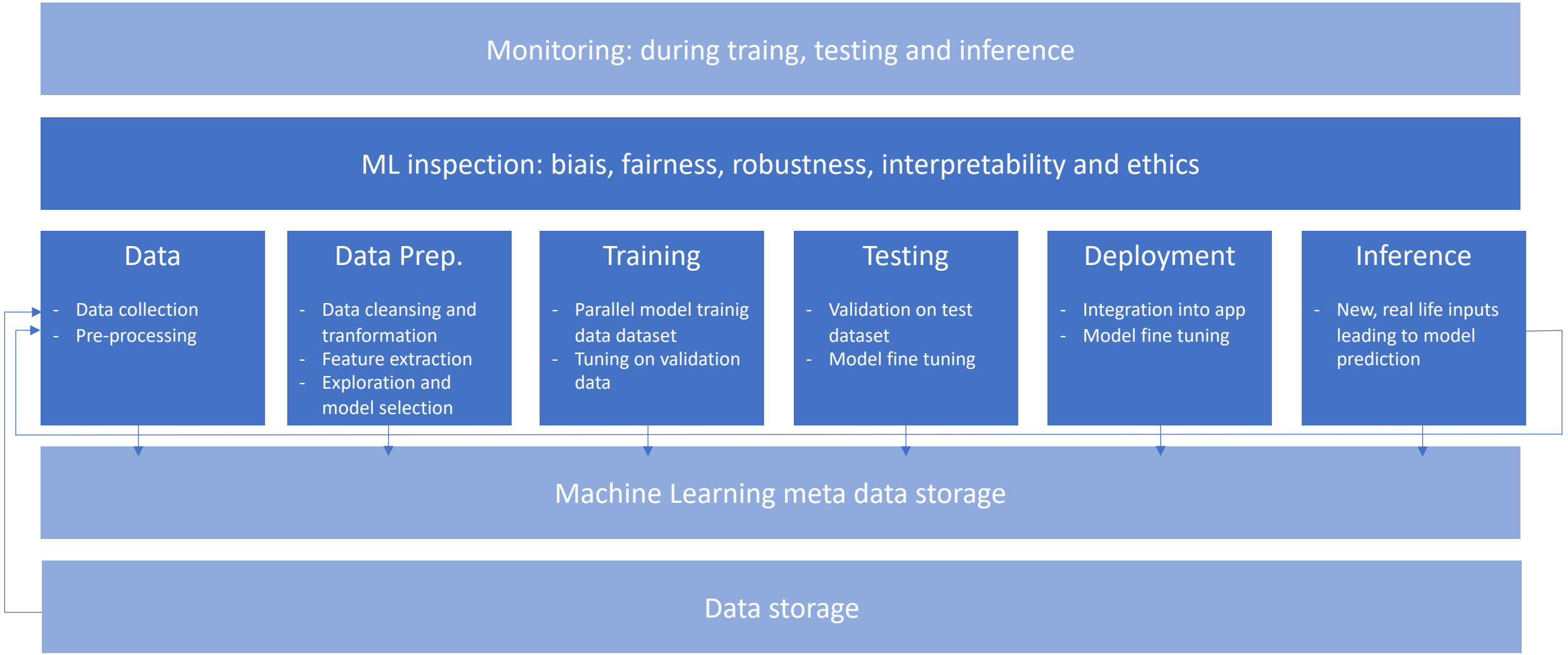
The story of enterprise Machine Learning: “It took me 3 weeks to develop the model. It’s been >11 months, and it’s still not deployed.” @DineshNirmalIBM #StrataData #strataconf

10:19 AM - 7 Mar 2018

8 Retweets 25 Likes

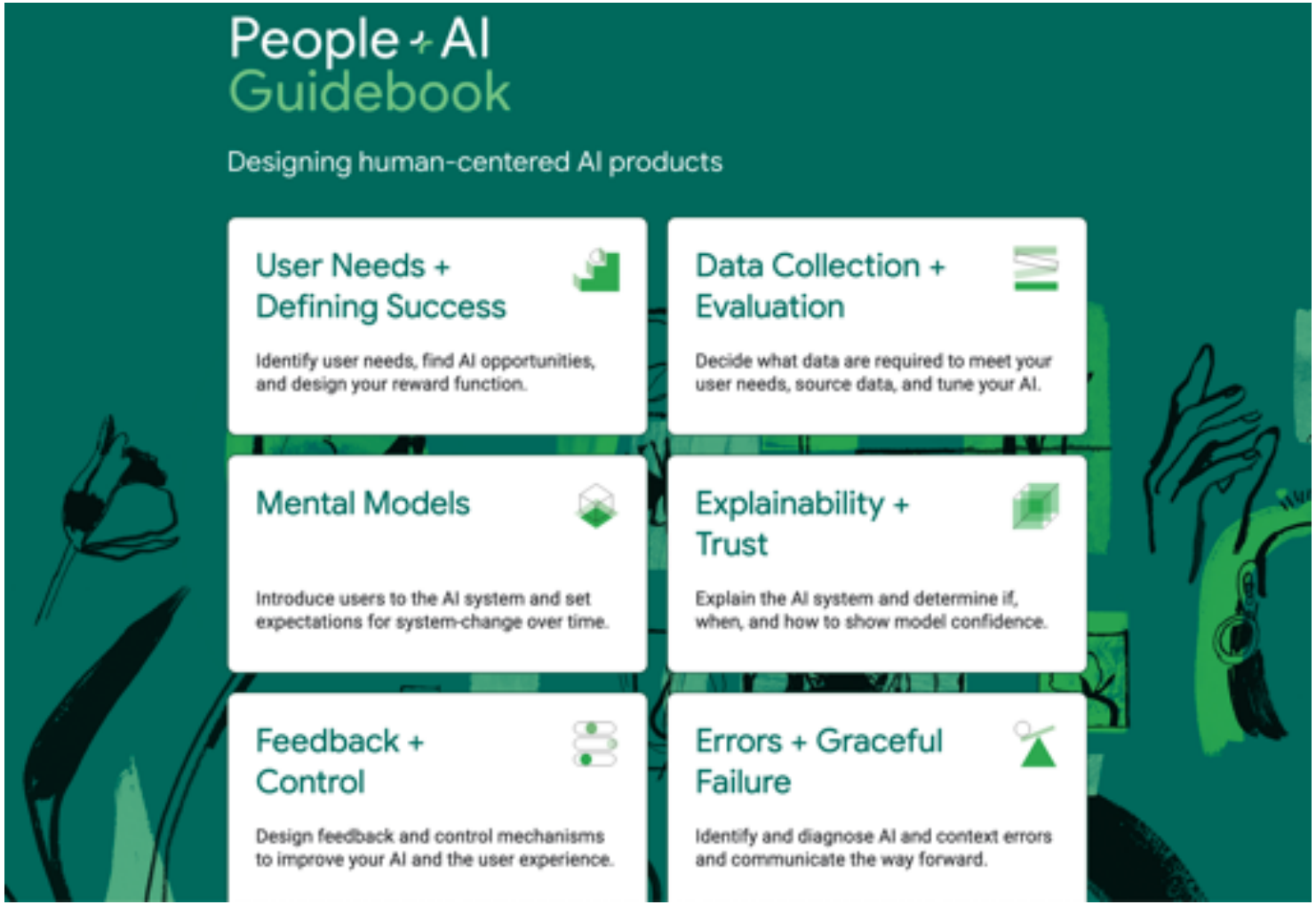
8 25

Machine Learning Pipelines



https://www.agilestacks.com/tutorials/ml-pipelines

Designing human-centered AI products



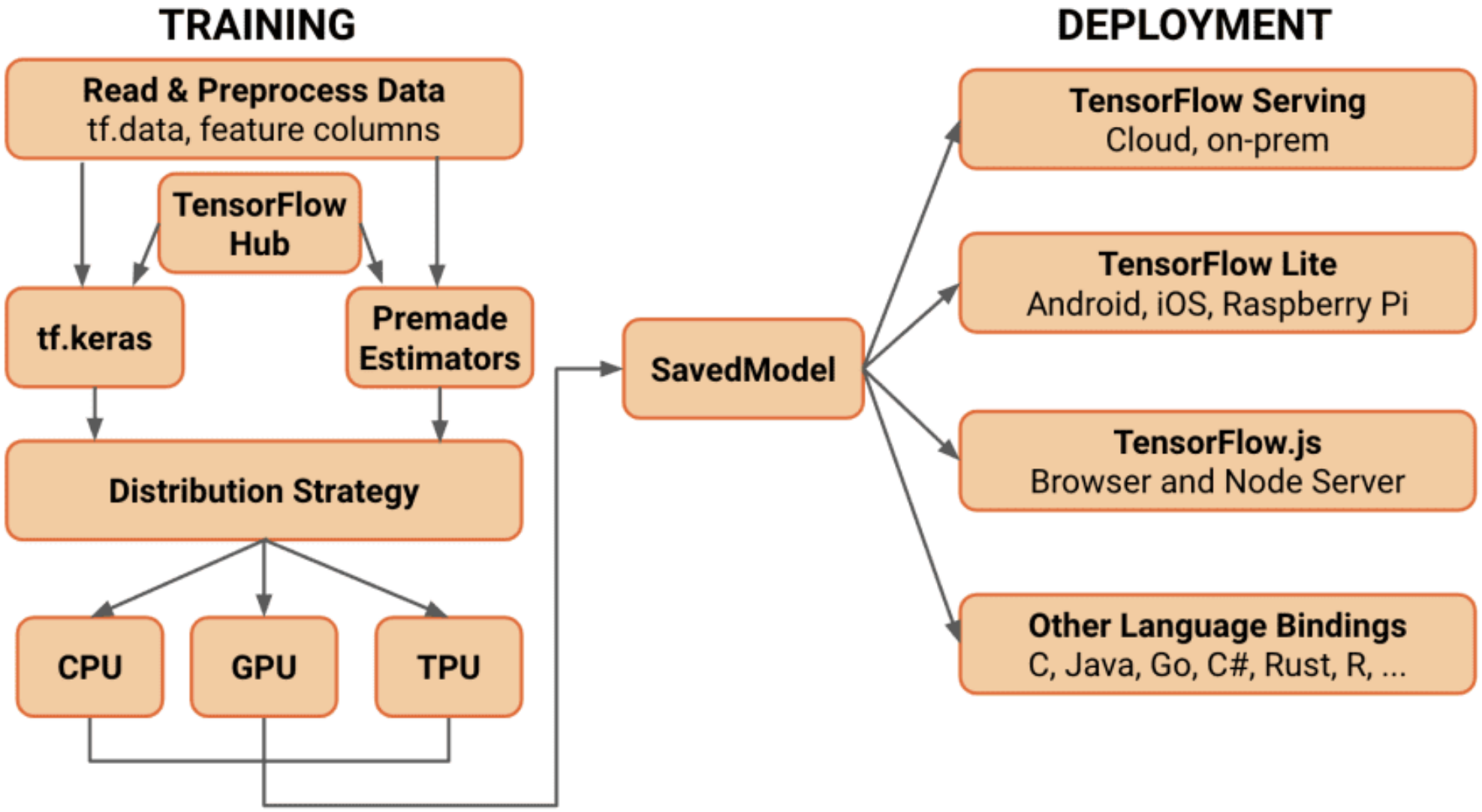
https://pair.withgoogle.com/

[Google People + AI Research (PAIR) team]

TensorFlow

TensorFlow Ecosystem Overview

TensorFlow Ecosystem [incomplete]



TensorFlow Ecosystem [incomplete]

TensorFlow 2.0

Easy

Simplified APIs.
Focused on Keras and
eager execution

Powerful

Flexibility and performance.
Power to do cutting edge research
and scale to > 1 exaflops

Scalable

Tested at Google-scale.
Deploy everywhere

Deploy anywhere

Servers

TensorFlow
Extended

Edge devices

TensorFlow
Lite

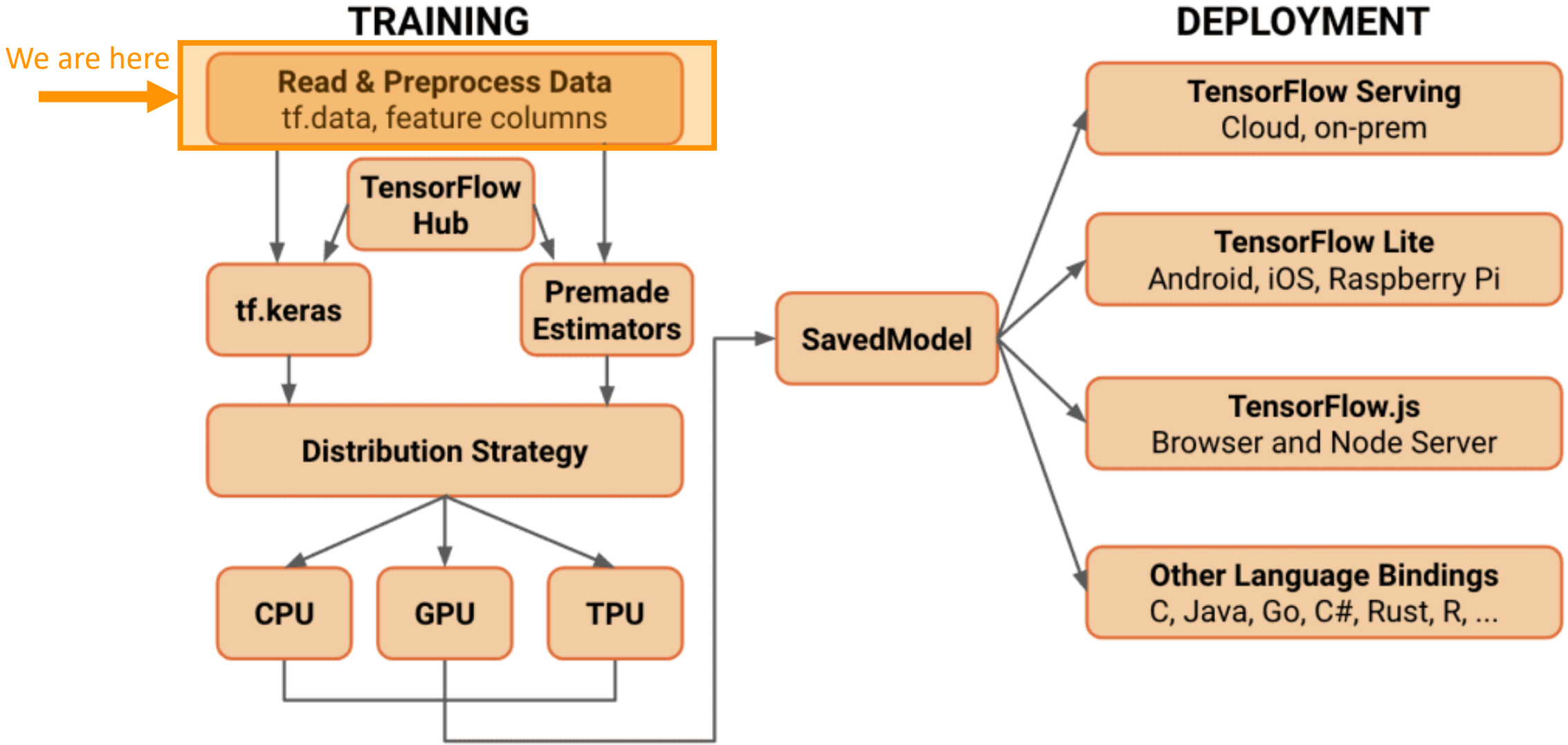
JavaScript

TensorFlow
.JS

```
pip install tensorflow==2.1.0
```

TensorFlow

TensorFlow - Keras


TRAINING
We are here
Read & Preprocess Data
tf.data, feature columns
TensorFlow Hub
tf.keras
Premade Estimators
Distribution Strategy
CPU
GPU
TPU
SavedModel
DEPLOYMENT
TensorFlow Serving
Cloud, on-prem
TensorFlow Lite
Android, iOS, Raspberry Pi
TensorFlow.js
Browser and Node Server
Other Language Bindings
C, Java, Go, C#, Rust, R, ...

TensorFlow Datasets: a huge collection of datasets

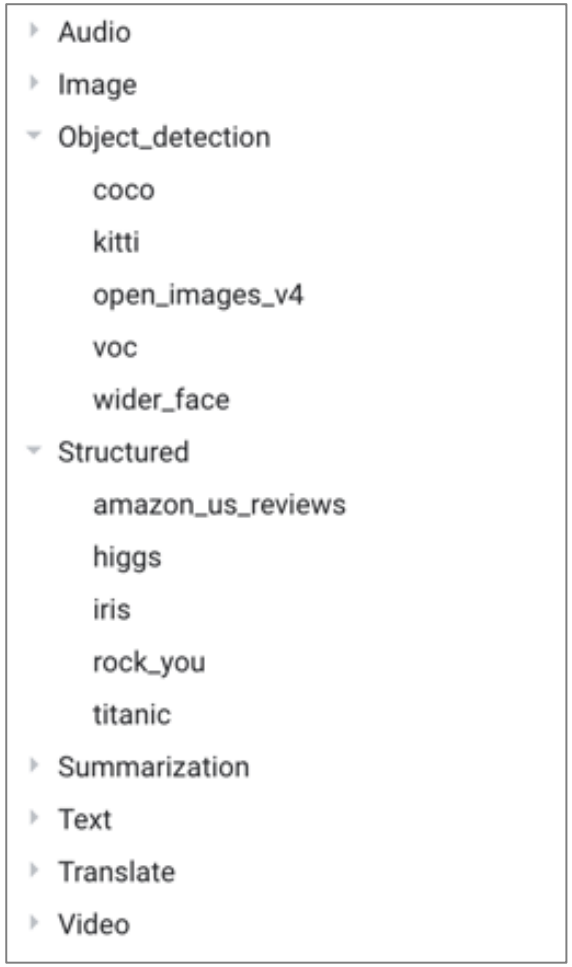
- A collection of datasets ready to use with TensorFlow

```python
import tensorflow_datasets as tfds

# Download the dataset and create a tf.data.Dataset
ds, info = tfds.load("mnist", split="train", with_info=True)

# Access relevant metadata with DatasetInfo
print(info.splits["train"].num_examples)
print(info.features["label"].num_classes)

# Build your input pipeline
ds = ds.batch(128).repeat(10)
```

https://blog.tensorflow.org/2019/02/introducing-tensorflow-datasets.html

- Today 101 datasets available from mnist to wikipedia

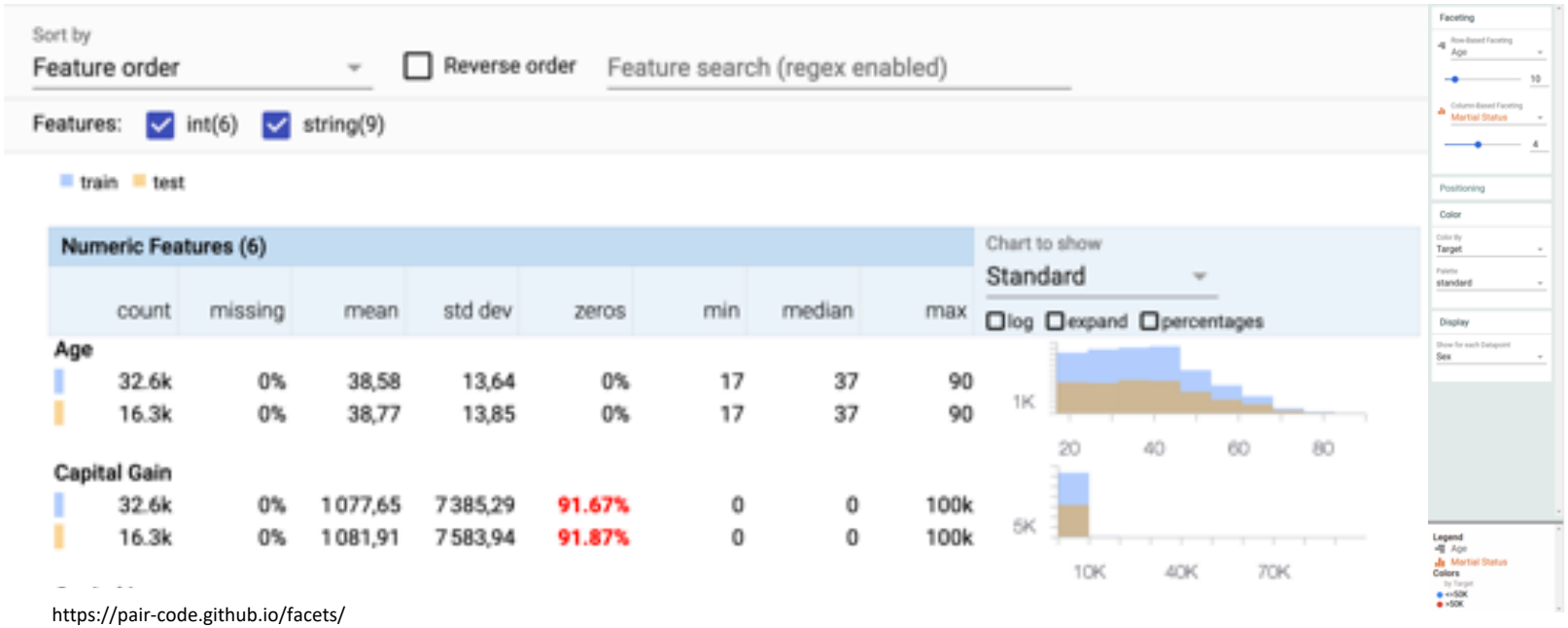
pip install tensorflow-datasets



Data visualization: Facet Overview and Dive

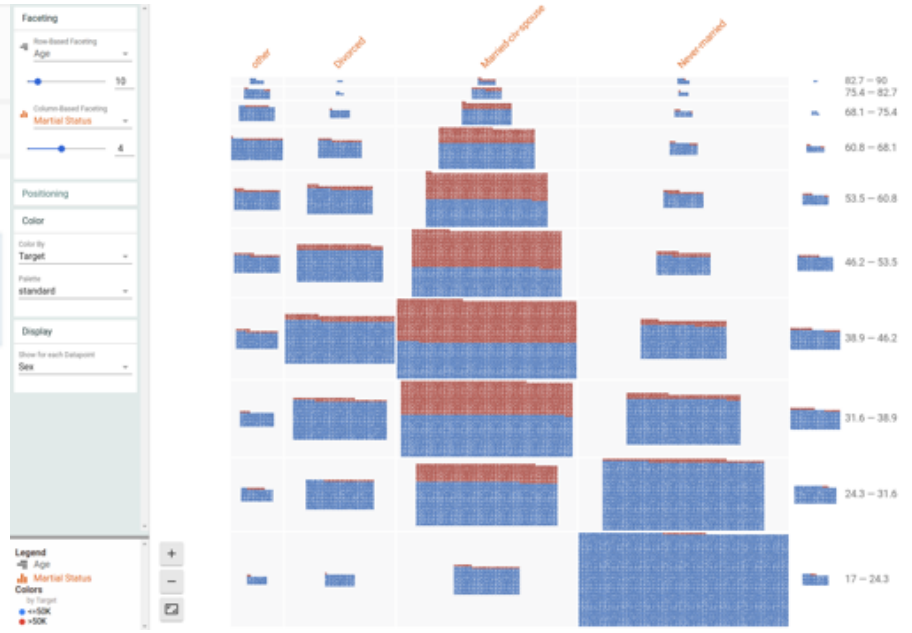
- Overview [Google PAIR] gives users a quick understanding of the distribution of values across the features of their dataset

Facet Overview



https://pair-code.github.io/facets/

Facet Dive



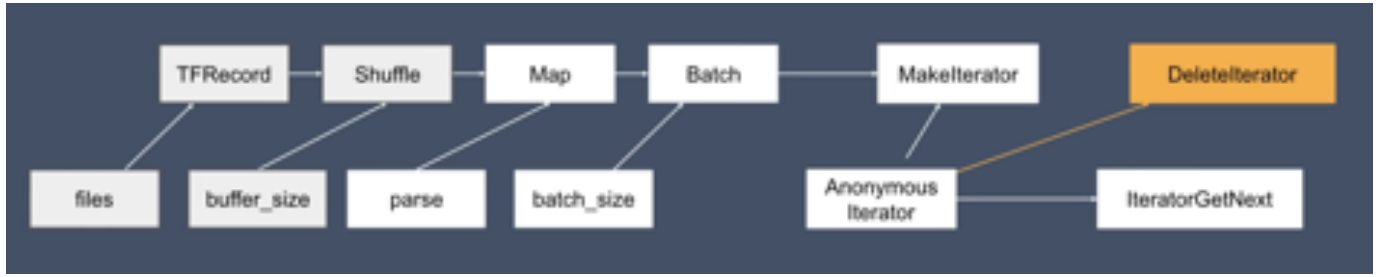
https://github.com/PAIR-code/facets

- Dive [Google PAIR is a tool for interactively exploring up to tens of thousands of multidimensional data points, allowing users to seamlessly switch between a high-level overview and low-level details.

pip install facets-overview

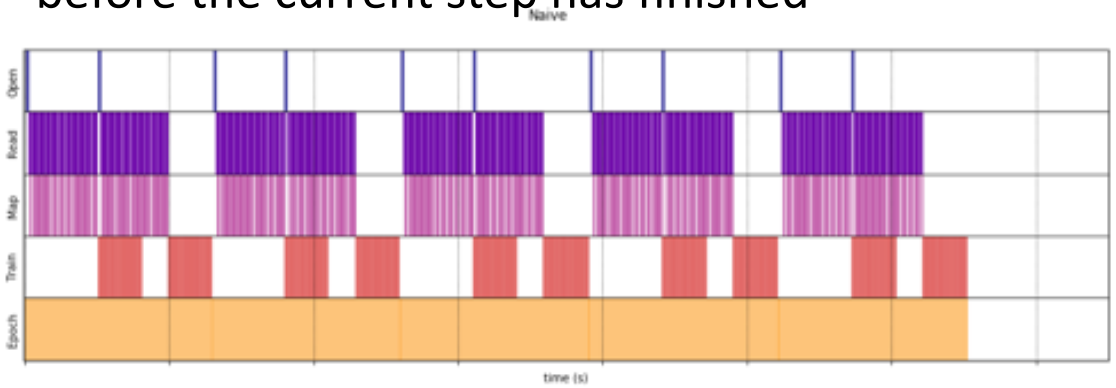
Tensorflow tf.data: build input pipelines at scale

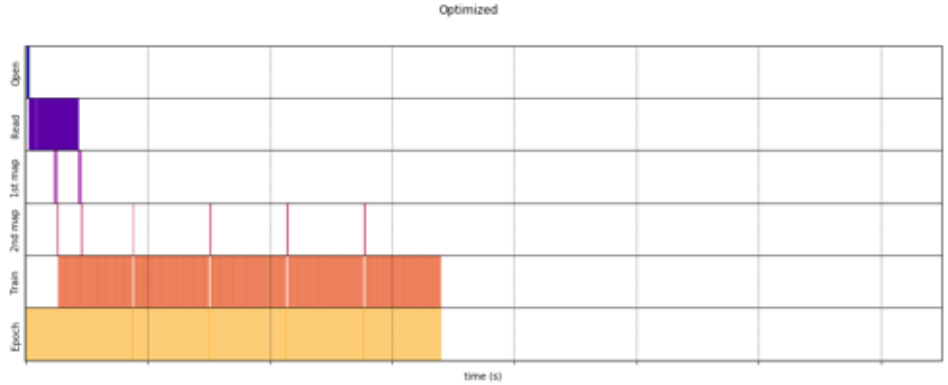
- Dataset usage follows a common pattern:
  1- Create a source dataset from your input data
  2- Apply dataset transformations to preprocess the data
  3- Iterate over the dataset and process the elements




```
dataset = tf.data.TFRecordDataset(files)

dataset = dataset.shuffle(buffer_size=X)

dataset = dataset.map(lambda record: parse(record))

dataset = dataset.batch(batch_size=Y)


for element in dataset:

  ...
```

- Achieving peak performance requires an efficient input pipeline that delivers data for the next step before the current step has finished





TensorFlow Text: preprocessing for text-based models

- The library can perform the preprocessing regularly required by text-based models: unicode, tokenization, wordshape, N-grams & Sliding Window

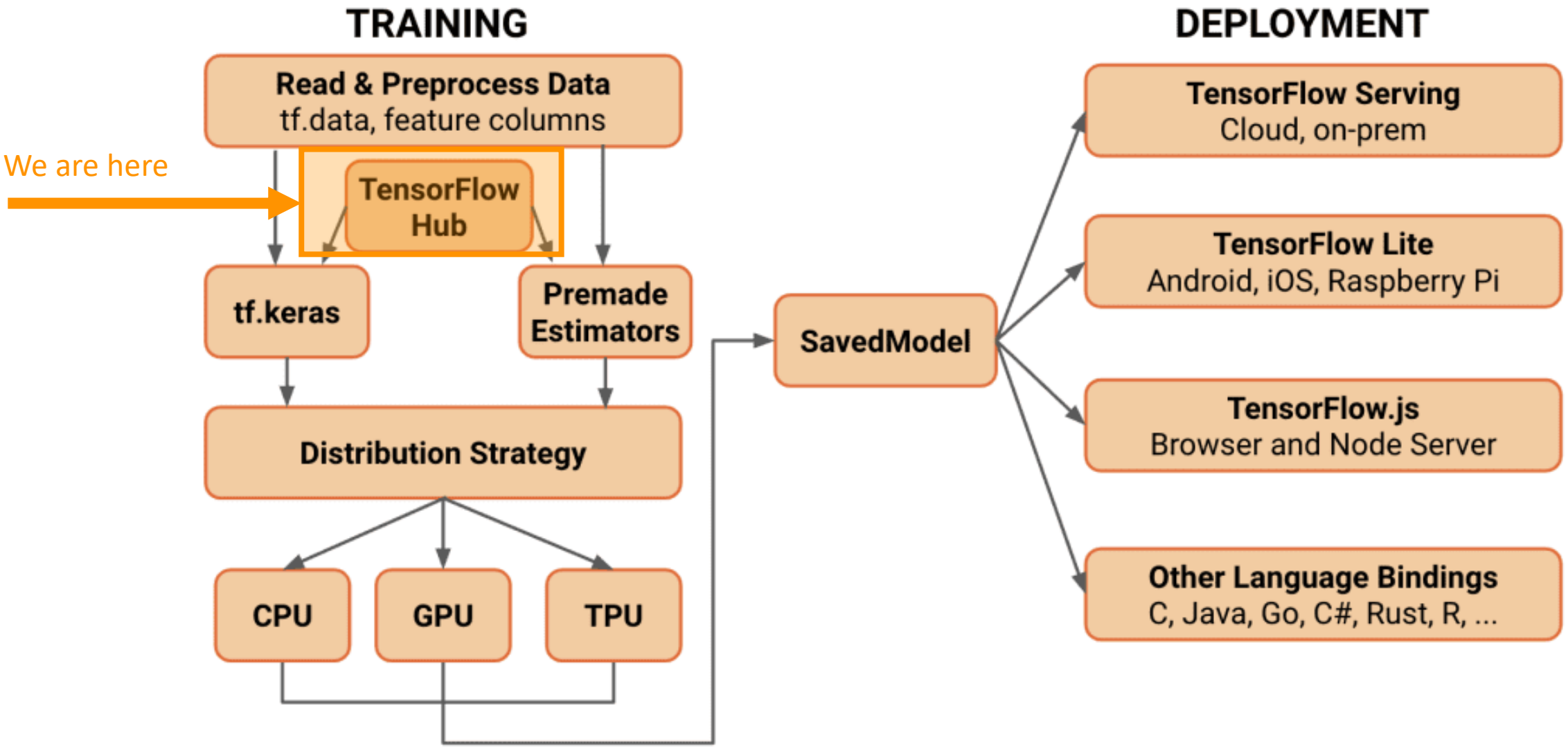
```
tokenizer = text.WhitespaceTokenizer()
tokens = tokenizer.tokenize(['Everything not saved will be lost.', u'Sad☹'.encode('UTF-8')])

# Ngrams, in this case bi-gram (n = 2)
bigrams = text.ngrams(tokens, 2, reduction_type=text.Reduction.STRING_JOIN)
```

https://www.tensorflow.org/tutorials/tensorflow_text/intro

- The library contains implementations of text-similarity metrics such as ROUGE-L, required for automatic evaluation of text generation models.

- Keras provide also some text processing functionalities: Tokenizer, hashing_trick, one_hot and text_to_word_sequence

pip install tensorflow-text


TRAINING
DEPLOYMENT
We are here
Read & Preprocess Data
tf.data, feature columns
TensorFlow Hub
tf.keras
Premade Estimators
Distribution Strategy
CPU
GPU
TPU
SavedModel
TensorFlow Serving
Cloud, on-prem
TensorFlow Lite
Android, iOS, Raspberry Pi
TensorFlow.js
Browser and Node Server
Other Language Bindings
C, Java, Go, C#, Rust, R, ...

TensorFlow Hub: reusable machine learning modules

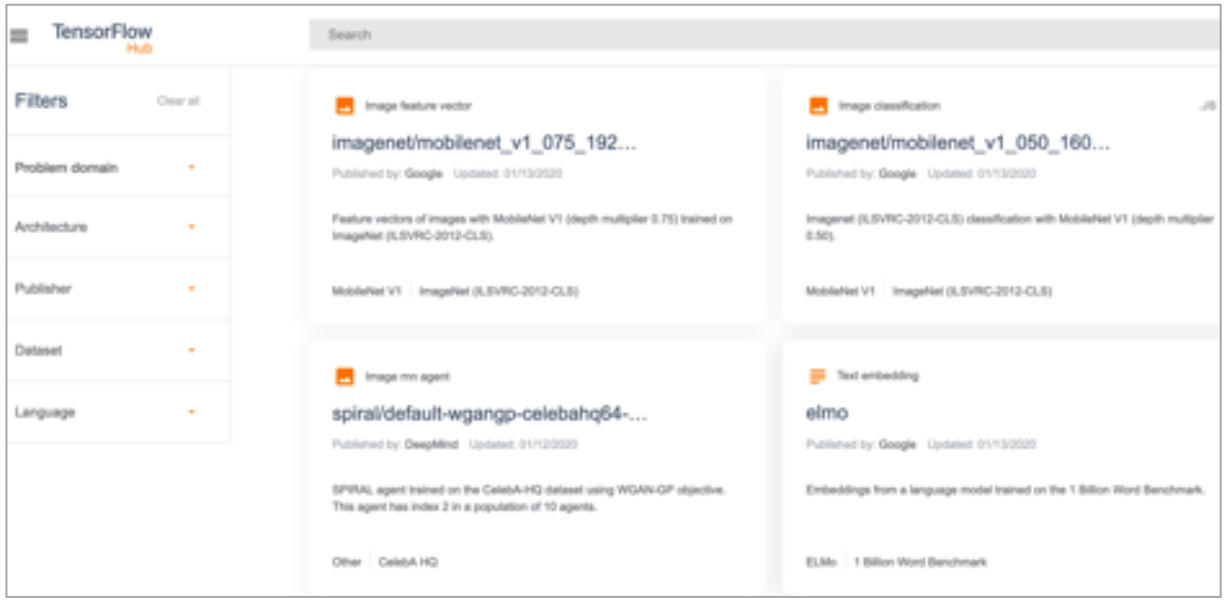
- Consumption of reusable parts of ML models

```
model = "https://tfhub.dev/google/tf2-preview/gnews-swivel-20dim/1"
hub_layer = hub.KerasLayer(model, output_shape=[20], input_shape=[],
                           dtype=tf.string, trainable=True)
hub_layer(train_examples[:3])
```

Let's now build the full model:

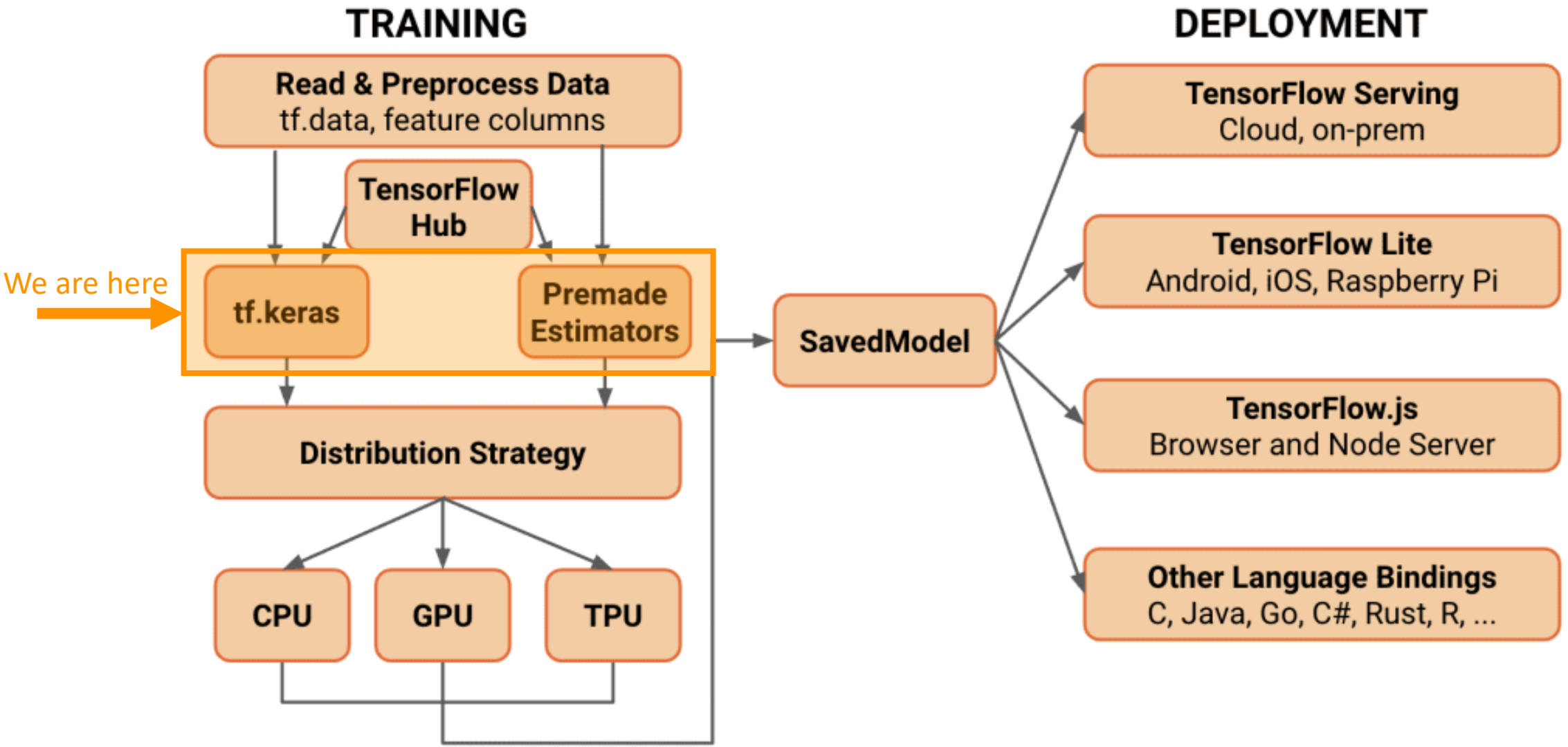
```
model = tf.keras.Sequential()
model.add(hub_layer)
model.add(tf.keras.layers.Dense(16, activation='relu'))
model.add(tf.keras.layers.Dense(1, activation='sigmoid'))

model.summary()
```

https://github.com/tensorflow/hub/blob/master/examples/colab/tf2_text_classification.ipynb



https://tfhub.dev/

- Why:
  - NASNet took thousands of GPU-hours to train
  - by sharing the learned weights, a model developer can make it easier for others to reuse
- Some examples:
  - Bert: tfhub link
  - Inception/ResNet: tfhub link
  - today 453 pre-trained models !


TRAINING
DEPLOYMENT
Read & Preprocess Data
tf.data, feature columns
TensorFlow Hub
We are here
tf.keras
Premade Estimators
Distribution Strategy
CPU
GPU
TPU
SavedModel
TensorFlow Serving
Cloud, on-prem
TensorFlow Lite
Android, iOS, Raspberry Pi
TensorFlow.js
Browser and Node Server
Other Language Bindings
C, Java, Go, C#, Rust, R, ...

Keras Tuner: Hyperparameter tuning

- New tool to do hyperparameter tuning

```
from tensorflow import keras
from tensorflow.keras import layers
from kerastuner.tuners import RandomSearch


def build_model(hp):
    model = keras.Sequential()
    model.add(layers.Dense(units=hp.Int('units',
                                        min_value=32,
                                        max_value=512,
                                        step=32),
                           activation='relu'))
    model.add(layers.Dense(10, activation='softmax'))
    model.compile(
        optimizer=keras.optimizers.Adam(
            hp.Choice('learning_rate',
                      values=[1e-2, 1e-3, 1e-4])),
        loss='sparse_categorical_crossentropy',
        metrics=['accuracy'])
    return model
```

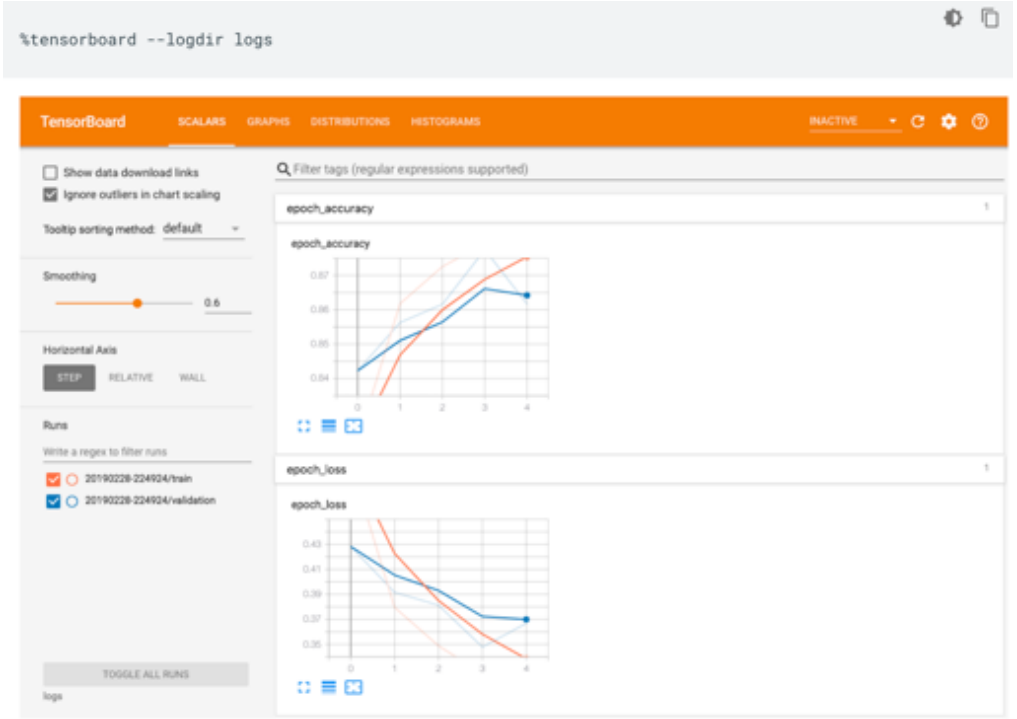
```
tuner = RandomSearch(
    build_model,
    objective='val_accuracy',
    max_trials=5,
    executions_per_trial=3,
    directory='my_dir',
    project_name='helloworld')
```

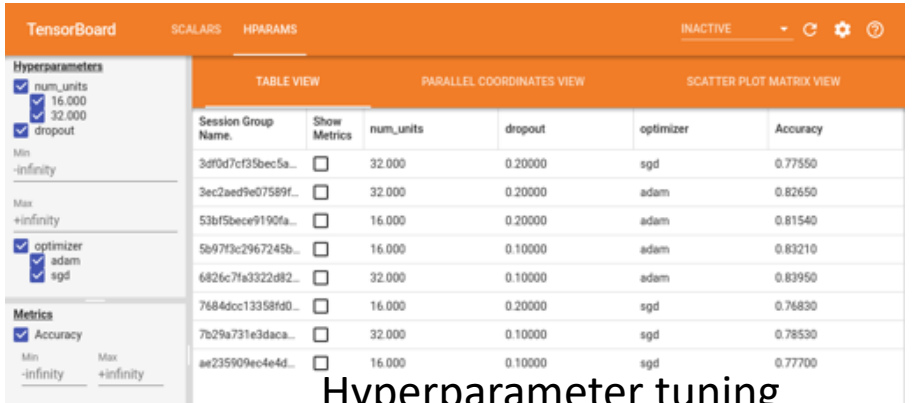
- Tuner available: RandomeSearch and Hyperband (paper)

pip install keras-tuner

TensorBoard:TensorFlow's visualization toolkit

- Now TensorBoard can also run in a Jupyter cell





Hyperparameter tuning

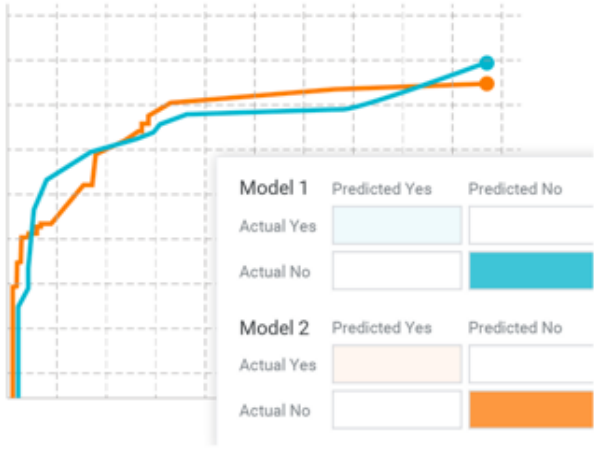
Profiling

- Allow visualisation of:
  scalars/metric, data, model graphs, hyperparameter tuning, what-if, fairness indicator, profiling
- TensorBoard.dev
  lets you upload and share your ML experiment results with anyone

Biais, fairness, robustness, interpretability and ethics in ML

- What-if tool [Google PAIR], coming together with TensorBoard

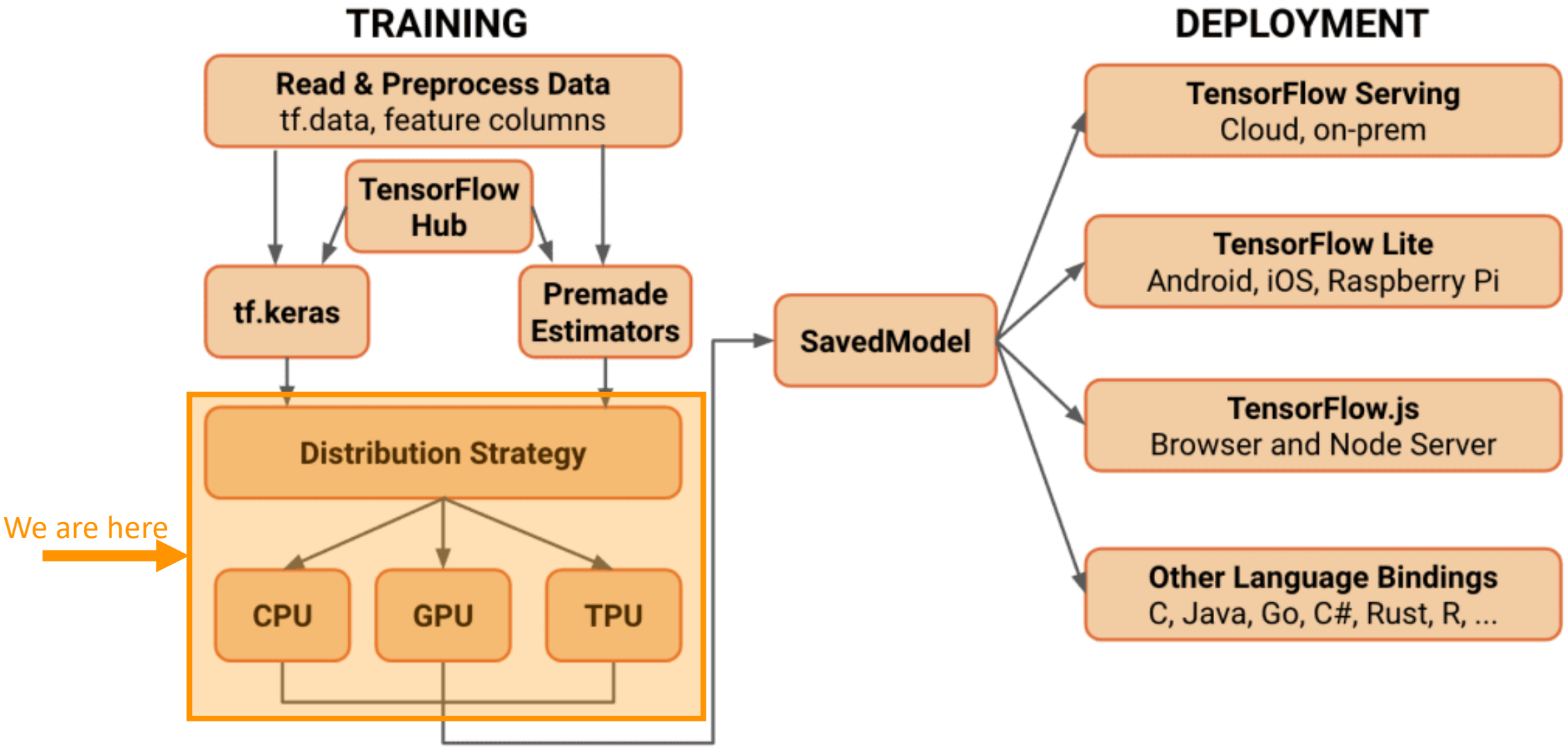
Compare multiple models within the same workflow



Test algorithmic fairness constraints

- Lime (Local Interpretable Model-Agnostic Explanations) non specific to TF
- Shap (SHapley Additive exPlanations) non specific to TF
- tf-explain: offers interpretability methods to ease neural network's understanding non official TF pkg
- Many other open source packages …

pip install lime | pip install shap | pip install tf-explain


TRAINING
Read & Preprocess Data
tf.data, feature columns
TensorFlow Hub
tf.keras
Premade Estimators
Distribution Strategy
CPU
GPU
TPU
We are here
SavedModel
DEPLOYMENT
TensorFlow Serving
Cloud, on-prem
TensorFlow Lite
Android, iOS, Raspberry Pi
TensorFlow.js
Browser and Node Server
Other Language Bindings
C, Java, Go, C#, Rust, R, ...

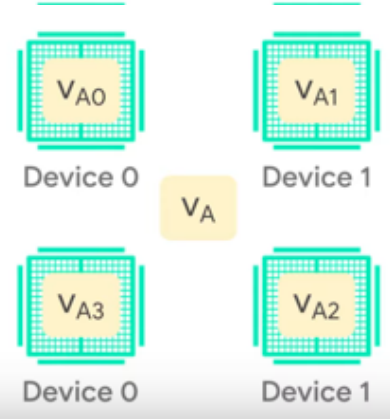
Distributed strategy: training at scale

- tf.distribute.Strategy is a TensorFlow API to distribute training across multiple GPUs, multiple machines or TPUs

```
strategy = tf.distribute.MirroredStrategy()
with strategy.scope():

  train_dataset = tf.data.Dataset(...)
  model = tf.keras.applications.ResNet50()
  optimizer = tf.keras.optimizers.SGD(learning_rate=0.1)
  model.compile(loss="...", optimizer=optimizer)
  model.fit(train_dataset, epochs=10)
```

Mirrored Strategy

- Every replica keeps its own local copy of every variable
- Kept in sync by applying identical updates



https://www.youtube.com/watch?v=jKV53r9-H14&feature=youtu.be

- Various option available [check the doc for up to date option and status]

Training API	MirroredStrategy	TPUStrategy	MultiWorkerMirroredStrategy	CentralStorageStrategy	ParameterServerStrategy
Keras API	Supported	Experimental support	Experimental support	Experimental support	Supported planned post 2.0
Custom training loop	Experimental support	Experimental support	Support planned post 2.0	Support planned post 2.0	No support yet
Estimator API	Limited Support	Not supported	Limited Support	Limited Support	Limited Support

TensorFlow Estimator

- TensorFlow Estimators: « Managing Simplicity vs. Flexibility in High-Level Machine Learning Frameworks » paper

- tf.estimator is a high-level TensorFlow API. Estimators encapsulate the following actions:
  - training
  - evaluation
  - prediction
  - export for serving

- Estimators provide a safe distributed training loop that controls how and when to:
  - load data
  - handle exceptions
  - create checkpoint files and recover from failures
  - save summaries for TensorBoard

- Type of estimator: pre-made and custom (You can convert existing Keras models to Estimators)

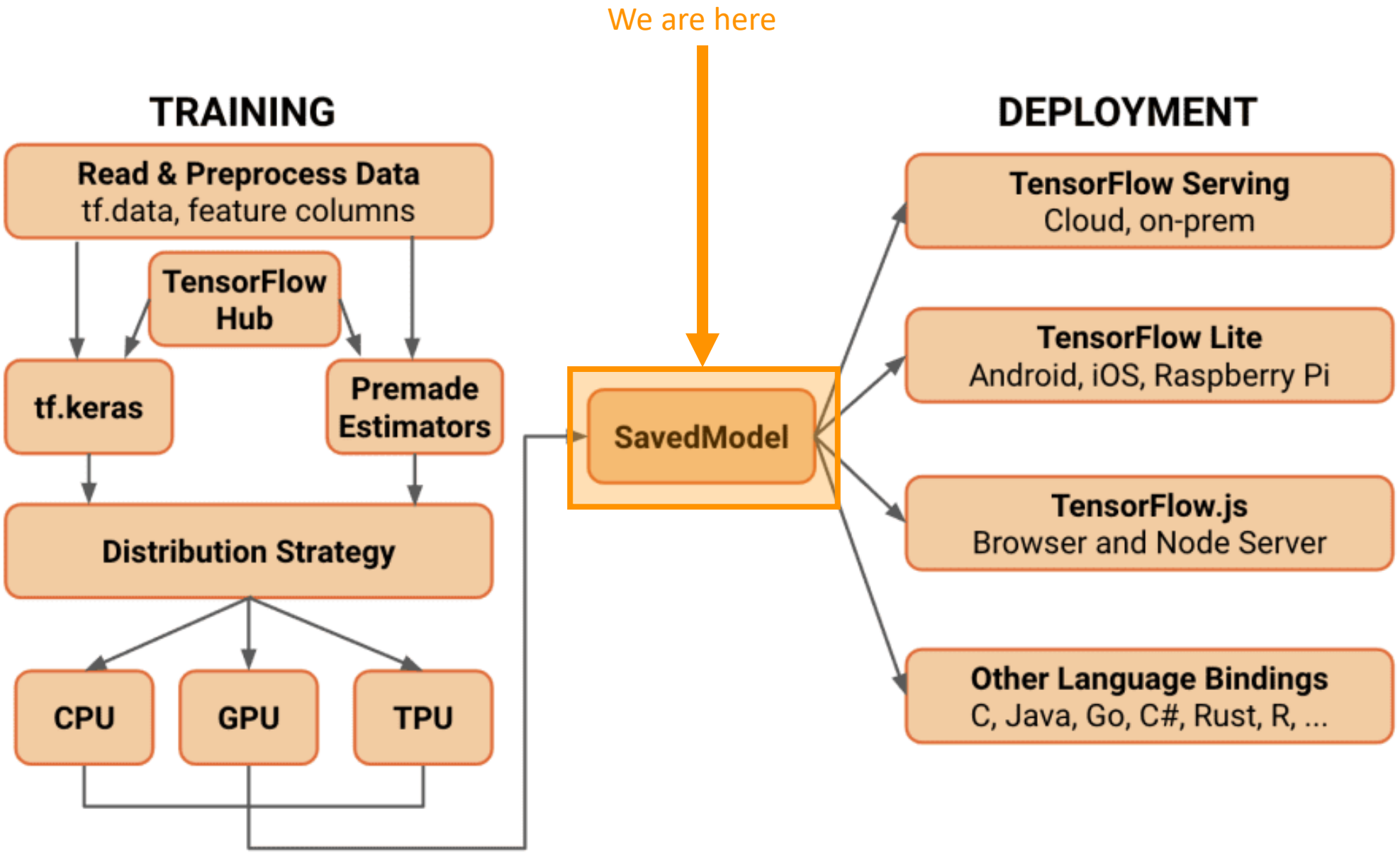
- You don't need to change our code at all to use this distributed config

```
gcloud ml-engine jobs submit training $JOB_NAME --scale-tier
`SCALE_TIER_STANDARD_1` \
     --runtime-version 1.4 --job-dir $GCS_JOB_DIR \
     --module-name trainer.task --package-path trainer/ \
     --region us-central1 \
     --train-steps 5000 --train-files $GCS_TRAIN_FILE --eval-files $GCS_EVAL_FILE --eval-steps 100
```

with Google Cloud Platform

Not sure sure about the future of estimator

https://cloud.google.com/blog/products/gcp/easy-distributed-training-with-tensorflow-using-tfestimatortrain-and-evaluate-on-cloud-ml-engine


We are here
TRAINING
Read & Preprocess Data
tf.data, feature columns
TensorFlow Hub
tf.keras
Premade Estimators
Distribution Strategy
CPU
GPU
TPU
SavedModel
DEPLOYMENT
TensorFlow Serving
Cloud, on-prem
TensorFlow Lite
Android, iOS, Raspberry Pi
TensorFlow.js
Browser and Node Server
Other Language Bindings
C, Java, Go, C#, Rust, R, ...

Save and load models

- Save checkpoints during training
  You can use a trained model without having to retrain it, or pick-up training where you left off—in case the training process was interrupted.
- Save the entire model
  Call model.save to save the a model's architecture, weights, and training configuration in a single file/folder.
  This allows you to export a model so it can be used without access to the original Python code*.
  Since the optimizer-state is recovered, you can resume training from exactly where you left off.

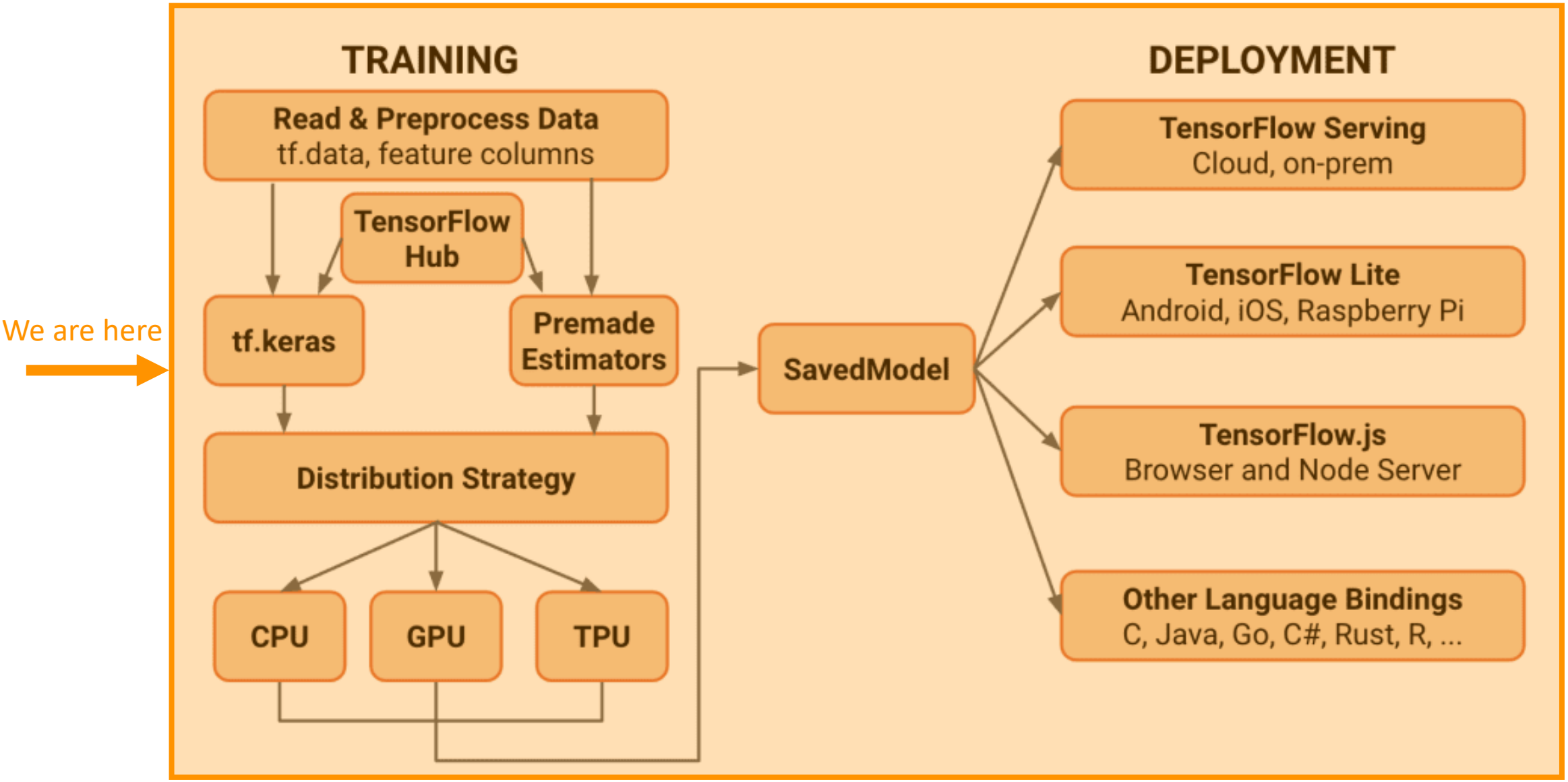
```
# Save the model
model.save('path_to_my_model.h5')

# Recreate the exact same model purely from the file
new_model = keras.models.load_model('path_to_my_model.h5')
```

```
assets/
assets.extra/
variables/
    variables.data-?????-of-?????
    variables.index
saved_model.pb|saved_model.pbtxt
```

where:

- `assets` is a subfolder containing auxiliary (external) files, such as vocabularies. Assets are copied to the SavedModel location and can be read when loading a specific `MetaGraphDef`.
- `assets.extra` is a subfolder where higher-level libraries and users can add their own assets that co-exist with the model, but are not loaded by the graph. This subfolder is not managed by the SavedModel libraries.
- `variables` is a subfolder that includes output from `tf.train.Saver`.
- `saved_model.pb` or `saved_model.pbtxt` is the SavedModel protocol buffer. It includes the graph definitions as `MetaGraphDef` protocol buffers.

Probably a bit different for TF 2.X
Same between Keras/Estimator


TRAINING
DEPLOYMENT
Read & Preprocess Data
tf.data, feature columns
TensorFlow Hub
tf.keras
Premade Estimators
Distribution Strategy
CPU
GPU
TPU
SavedModel
TensorFlow Serving
Cloud, on-prem
TensorFlow Lite
Android, iOS, Raspberry Pi
TensorFlow.js
Browser and Node Server
Other Language Bindings
C, Java, Go, C#, Rust, R, ...
We are here

TensorFlow

TensorFlow Extended (TFX)

An End-to-End ML Platform

a very high level overview …

Machine Learning in production

- Having a model trained is just a important tiny part
- A production solution requires so much more:

Developing Production ML Pipelines by Aurélien Géron @TensorFlow World 2019

- Paper from Google « Hidden Technical Debt in Machine Learning Systems » NIPS 2015 paper

Machine Learning in production

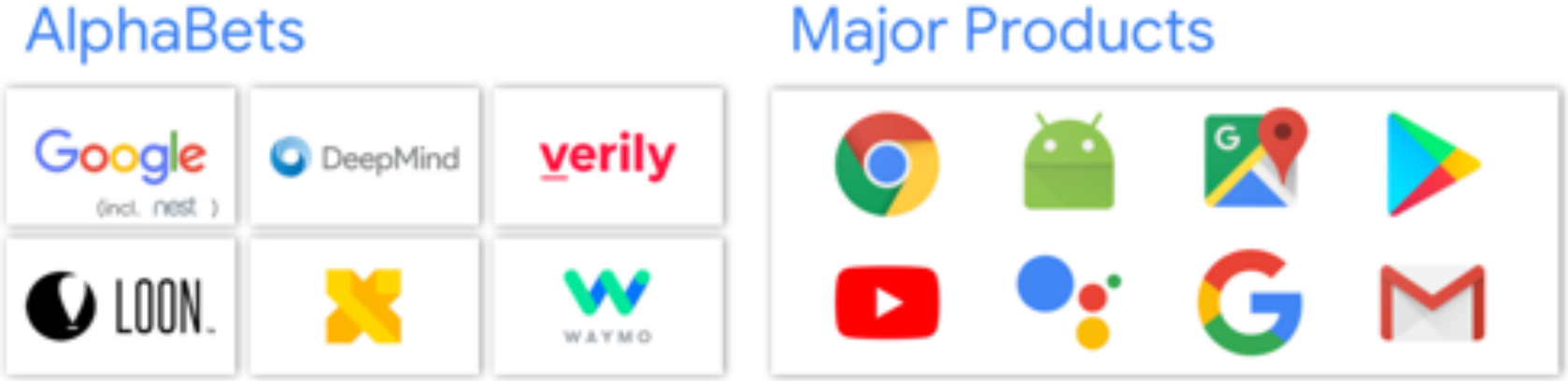
- Require high quality software

ML Programming **in the small** (Coding)	**ML** Programming **in the large** (Engineering)
Monolithic code	Modular design and implementation
Non-reusable code	Libraries for reuse (ideally across languages)
Undocumented code	Well documented contracts and abstractions
Untested code	Well tested code (exhaustively and at scale)
Unbenchmarked or hack-optimized once code	Continuously benchmarked and optimized code
Unverified code	Reviewed and peer verified code
Undebuggable code or adhoc tooling	Debuggable code and debug tooling
Uninstrumented code	Instrumentable and instrumented code

TensorFlow Extented (TFX)

- TFX is a Google-production-scale machine learning platform based on TensorFlow. It provides a configuration framework and shared libraries to integrate common components needed to define, launch, and monitor your machine learning system

- TFX powers Google most import bets and products

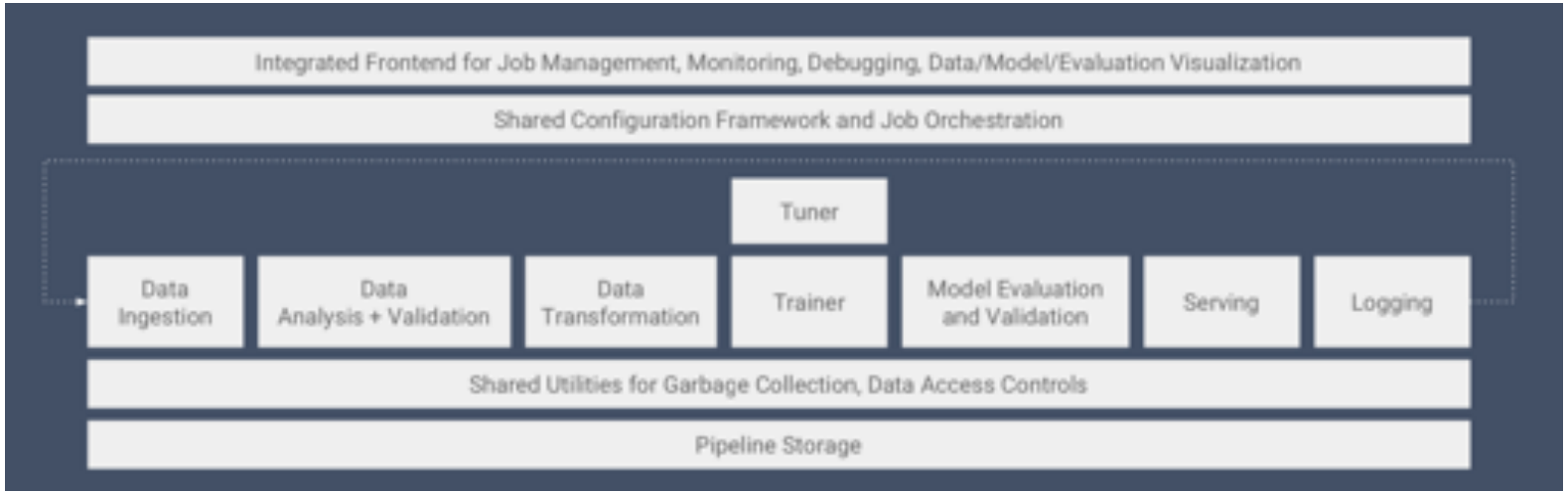


- Also used by other companies

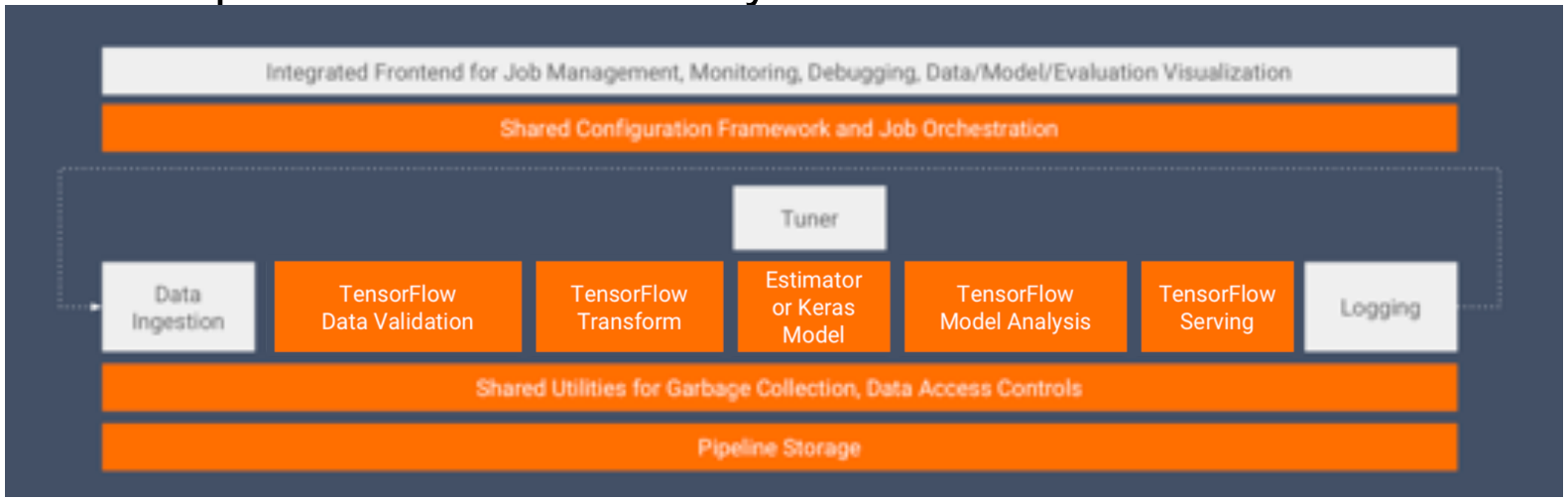
pip install tfx==0.15.0

TFX Overview

- View of all components when having ML model in production



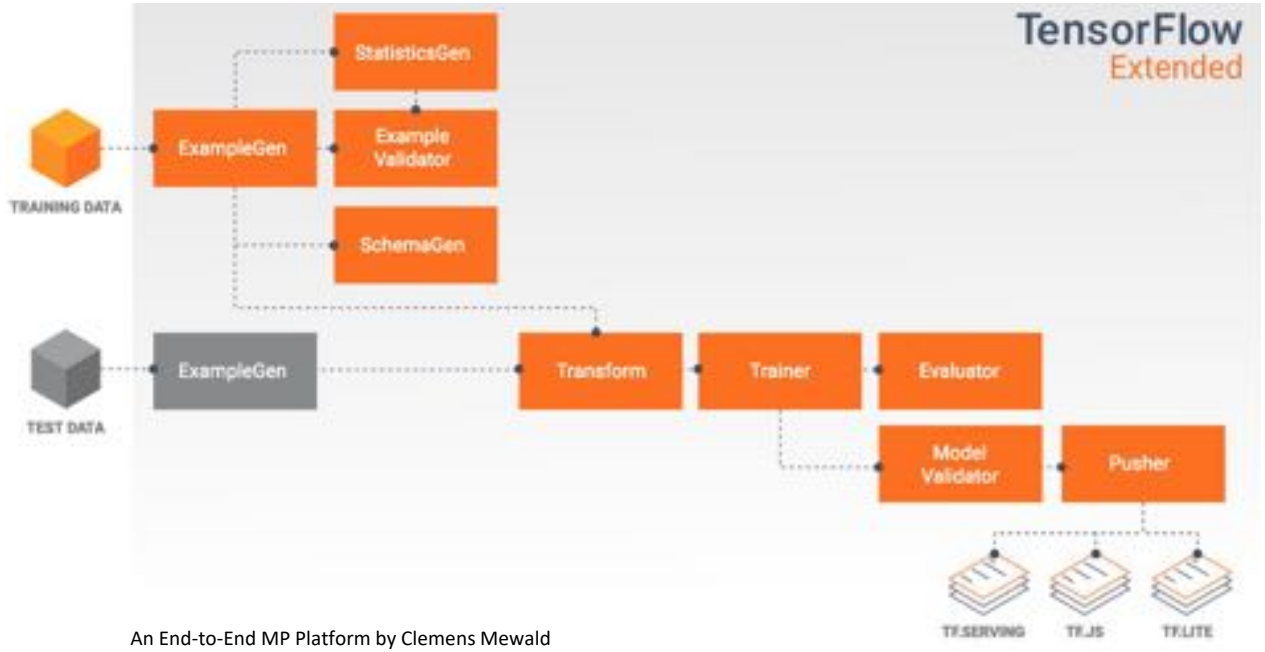
- TFX components available today



An End-to-End MP Platform by Clemens Mewald

TFX: building components of libraries

- TFX pipeline typically includes the following components:



pipeline components

An End-to-End MP Platform by Clemens Mewald

- TFX libraries which are used to create pipeline components.

TFX libraries

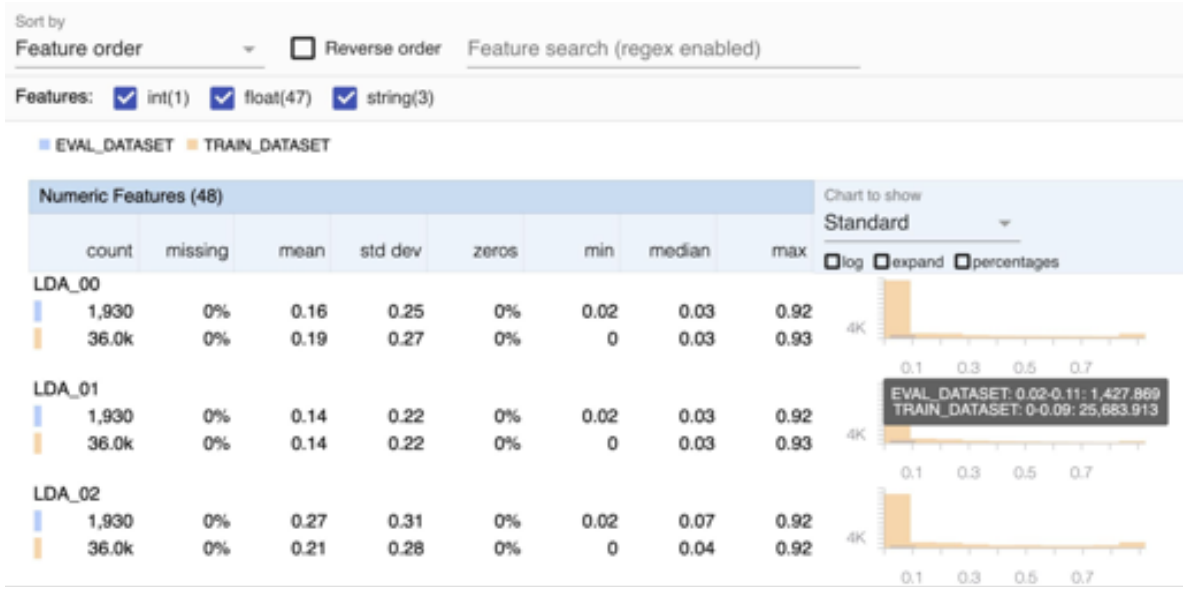
pipeline components

TFX: data ingestion and data validation

- Data ingestion:
  ExampleGen: is the initial input component of a pipeline that ingests and optionally splits the input dataset.

- TensorFlow Data Validation (TFDV)
  StatisticsGen calculates statistics for the dataset.
  SchemaGen examines the statistics and creates a data schema.
  ExampleValidator looks for anomalies and missing values in the dataset.

Facet overview



Developing Production ML Pipelines by Aurélien Géron @TensorFlow World 2019

Example of data schema

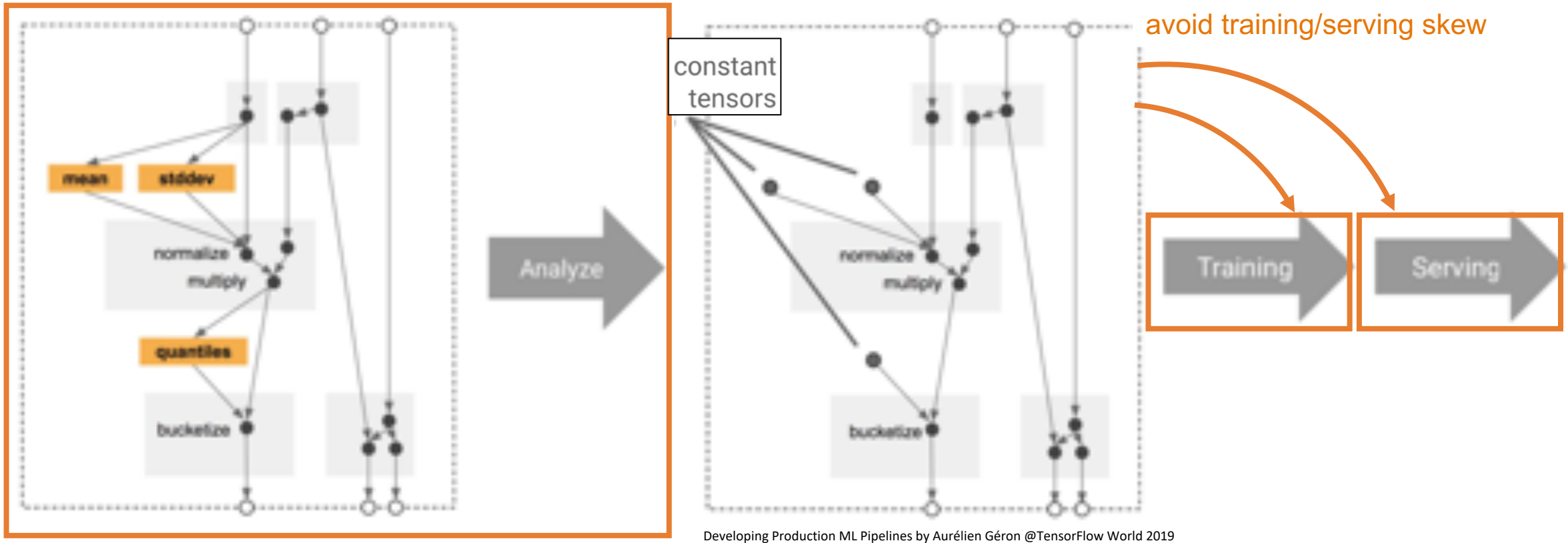
Feature name	Type	Presence	Valency	Domain
'data_channel'	STRING	required		'data_channel'
'slug'	BYTES	required		-
'date'	BYTES	required		-
'n_hrefs'	FLOAT	required		-
'kw_max_avg'	FLOAT	required		-
'n_imgs'	FLOAT	required		-
'n_non_stop_unique_tokens'	FLOAT	required		-
'kw_min_max'	FLOAT	required		-
'self_reference_max_shares'	FLOAT	required		-

Example of data validation

Feature name	Anomaly short description	Anomaly long description
'data_channel'	Unexpected string values	Examples contain values missing from the schema: Fun (<1%).

TFX: transform

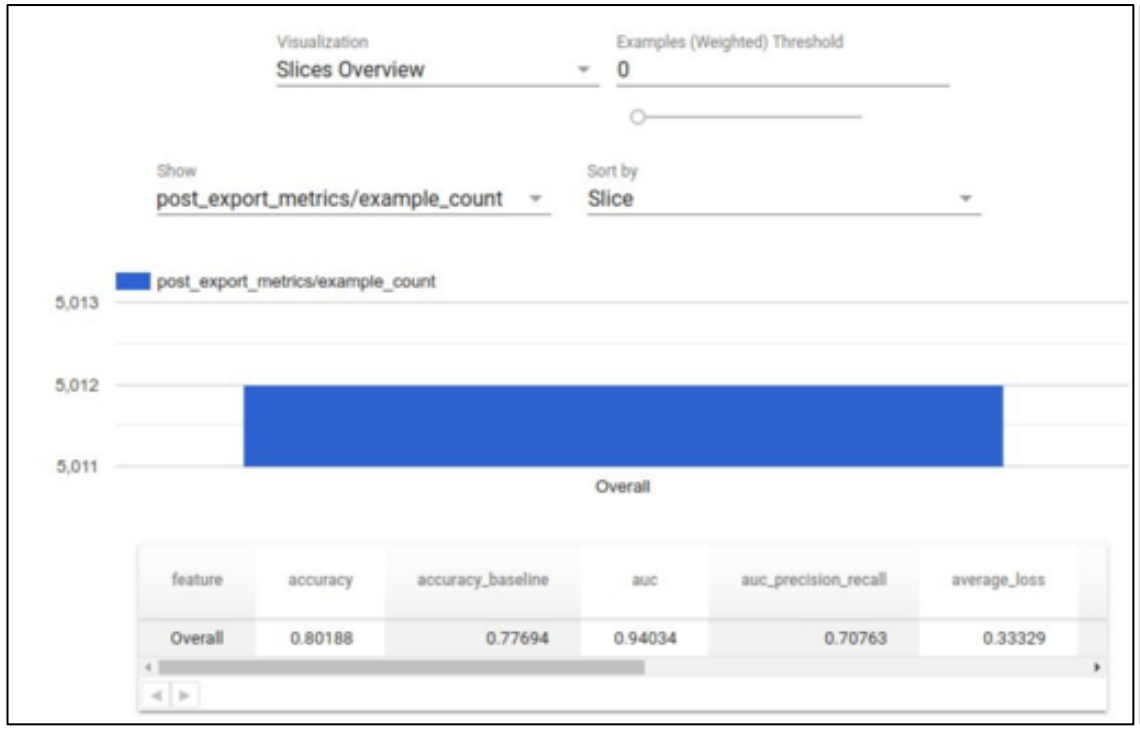
- TensorFlow Transform (TFT)
  Tranform performs feature engineering on the dataset
  - Fill in missing values
  - Normalize features
  - Bucketize features
  - Zoom/Crop images
  - Augment images
  - Feature crosses
  - Vocabularies
  - Embeddings
  - PCA
  - Categorical encoding



Developing Production ML Pipelines by Aurélien Géron @TensorFlow World 2019

TFX: training

- TensorFlow (TF)
  Trainer trains the model
- TensorFlow (TFMA)
  Evaluator performs deep analysis of the training results
  ModelValidator helps you validate your exported models, ensuring that they are "good enough" to be pushed to production



Developing Production ML Pipelines by Aurélien Géron @TensorFlow World 2019

An End-to-End MP Platform by Clemens Mewald

TFX: serving (deploy a REST API for your model in minutes)

- Pusher
  Pusher deploys the model on a serving infrastructure
- TensorFlow Serving (TFS)
  Serving provides out-of-the-box integration with TensorFlow models

TensorFlow Serving
Flexible

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

TensorFlow Serving
High-Performance

- Low-latency
- Request Batching
- Traffic Isolation

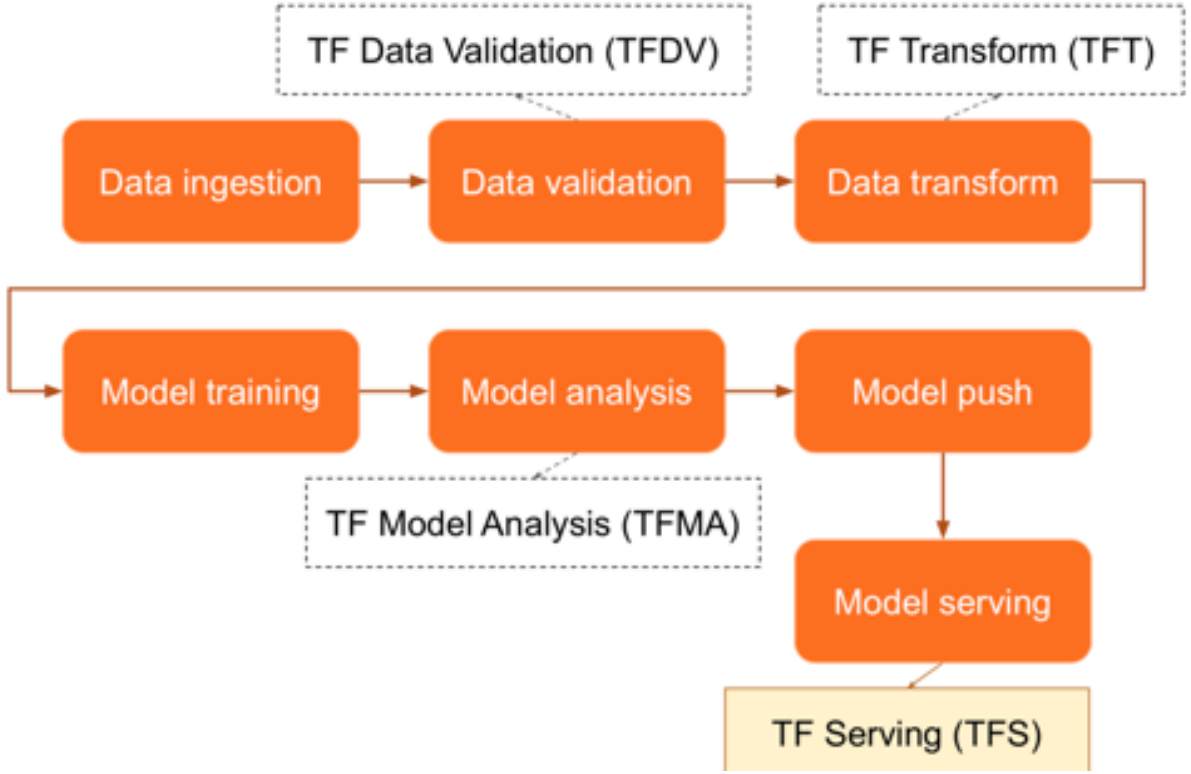
TensorFlow Serving
Production-Ready

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

An End-to-End MP Platform by Clemens Mewald

TFX recap

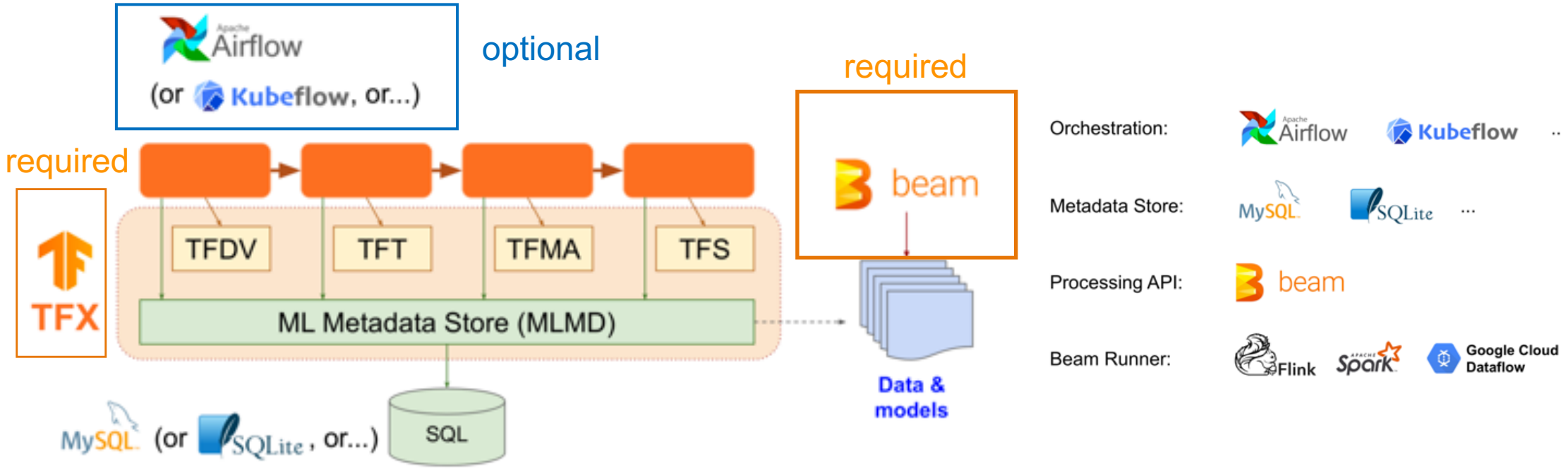
- ML pipeline and components



- Need some special underlying component/infrastructure
  can run on a laptop but require many components and design for big infrastructure

TFX: supporting technologies

- Overview of underlying core technologies needed



- Design for public/private cloud infrastructure do do ML at scale
  Kubeflow is an open source Kubernetes-native platform for developing, orchestrating, deploying, and running scalable and portable machine learning workloads.

  Kubernetes is a vendor-agnostic cluster and container management tool, open-sourced by Google in 2014. It provides a "platform for automating deployment, scaling, and operations of application containers across clusters of hosts"

TFX 2.X migration 

- Not so clear about the exact timeline from the documentation
- Goal for H1 2020

Usability

- Support of TensorFlow 2.0 in two phases:
  i. The first phase will provide the following: Existing TFX pipelines can continue to use TensorFlow 1.X. To switch to TensorFlow 2.X, see the TensorFlow migration guide. New TFX pipelines should use Keras (via `tf.keras.estimator.model_to_estimator()`) and TensorFlow 2.X.
  ii. The second phase will enable the remainder of TensorFlow 2.X functionality, including tf.distribute and Keras without Estimator.

https://github.com/tensorflow/tfx/blob/master/ROADMAP.md

TensorFlow

TensorFlow Ecosystem : not cover

TensorFlow Ecosystem

TensorFlow Hub

A library for the publication, discovery, and consumption of reusable parts of machine learning models.

Model Optimization

The TensorFlow Model Optimization Toolkit is a suite of tools for optimizing ML models for deployment and execution.

TensorFlow Federated

A framework for machine learning and other computations on decentralized data.

Neural Structured Learning

A learning paradigm to train neural networks by leveraging structured signals in addition to feature inputs.

TensorFlow Graphics

A library of computer graphics functionalities ranging from cameras, lights, and materials to renderers.

Serving

A TFX serving system for ML models, designed for high-performance in production environments.

Probability

TensorFlow Probability is a library for probabilistic reasoning and statistical analysis.

MLIR

MLIR unifies the infrastructure for high-performance ML models in TensorFlow.

XLA

A domain-specific compiler for linear algebra that accelerates TensorFlow models with potentially no source code changes.

SIG Addons

Extra functionality for TensorFlow, maintained by SIG Addons.

SIG IO

Dataset, streaming, and file system extensions, maintained by SIG IO.

TensorFlow Ecosystem

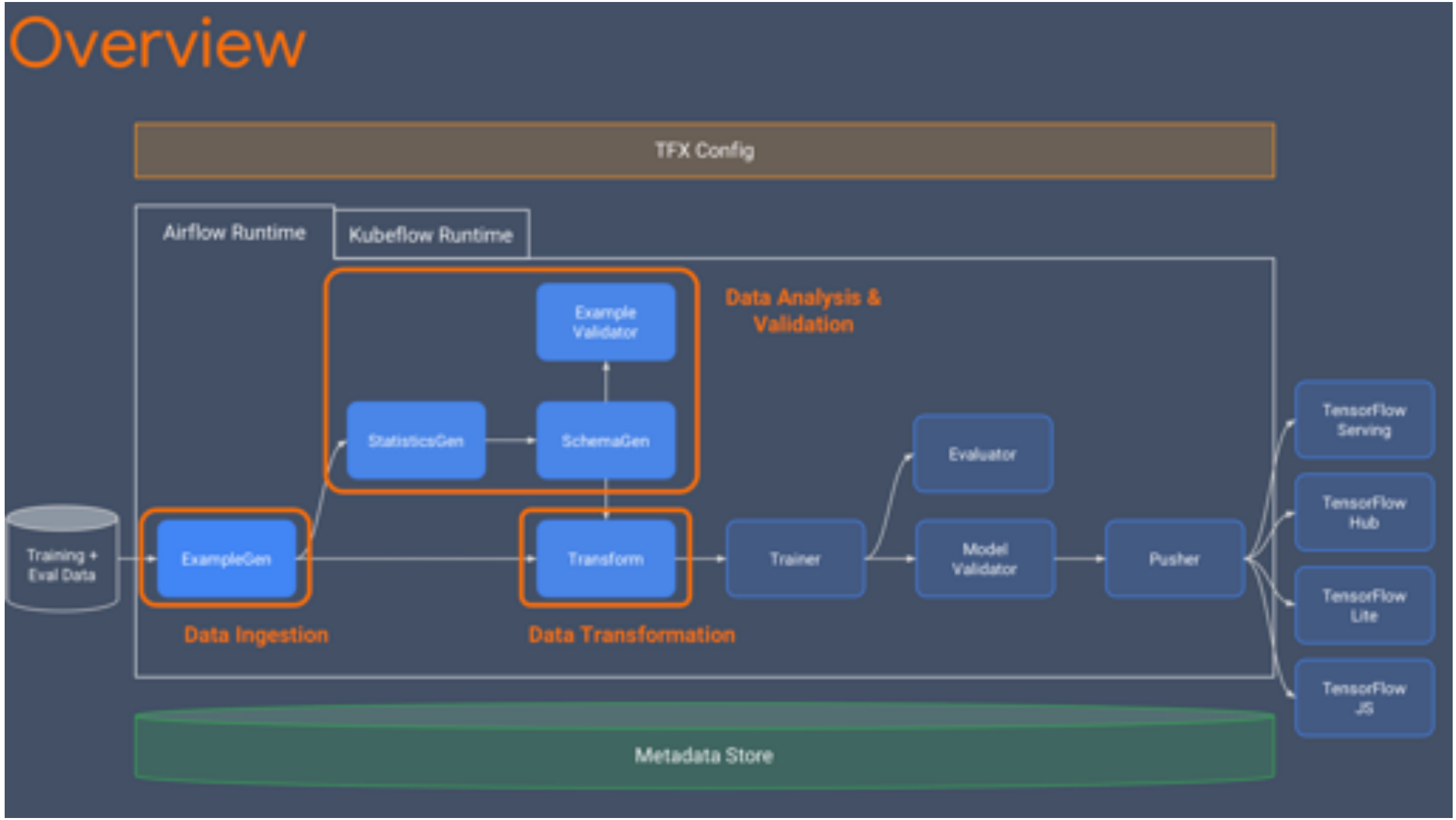
- TF Agents
- Tensor2Tensor
- TF Ranking
- TF Privacy
- Swift for TensorFlow
- TensorFlow.js
- TensorFlow Lite
- ….

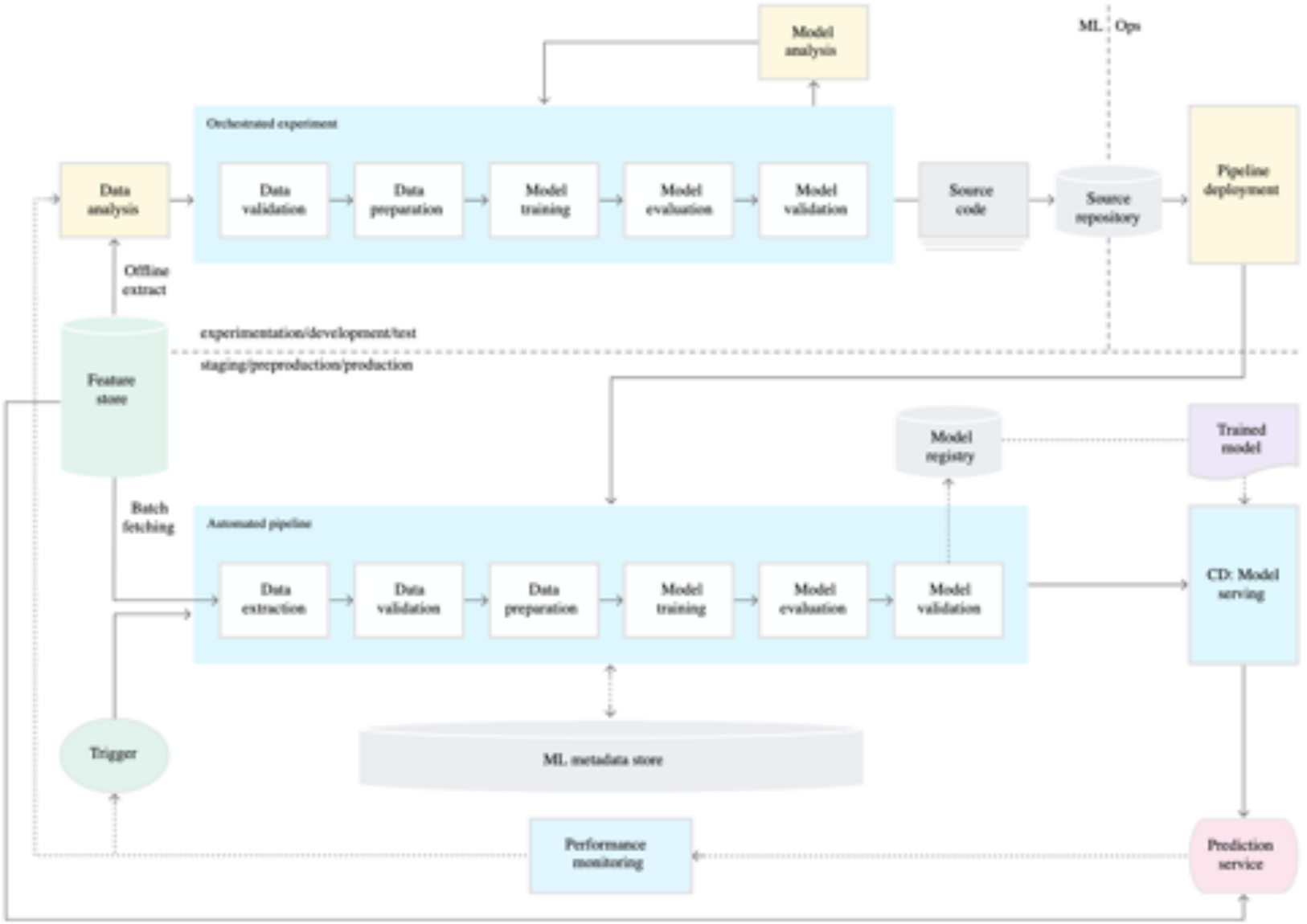
TensorFlow

Thank You!

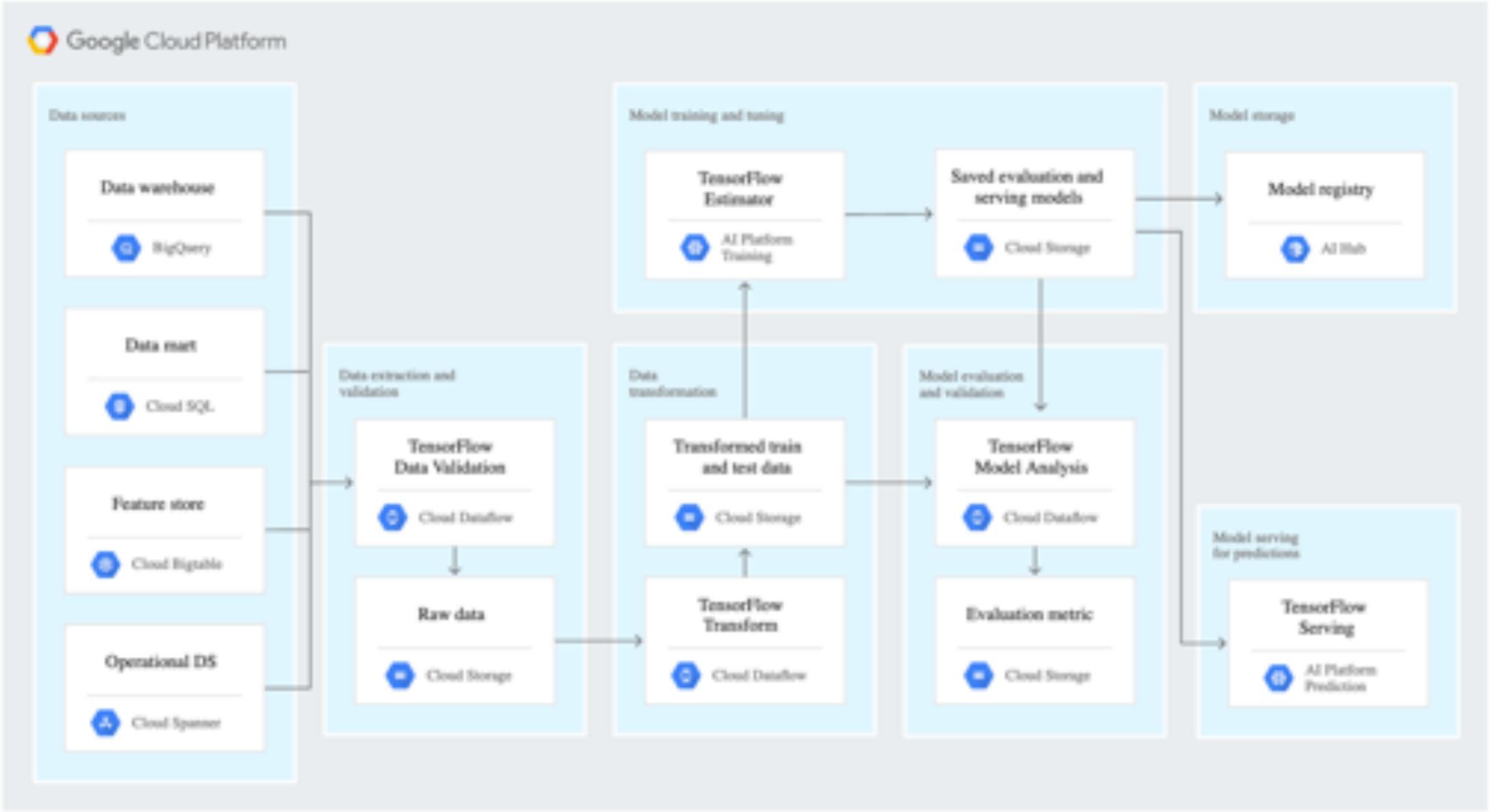
TensorFlow

Backup and references

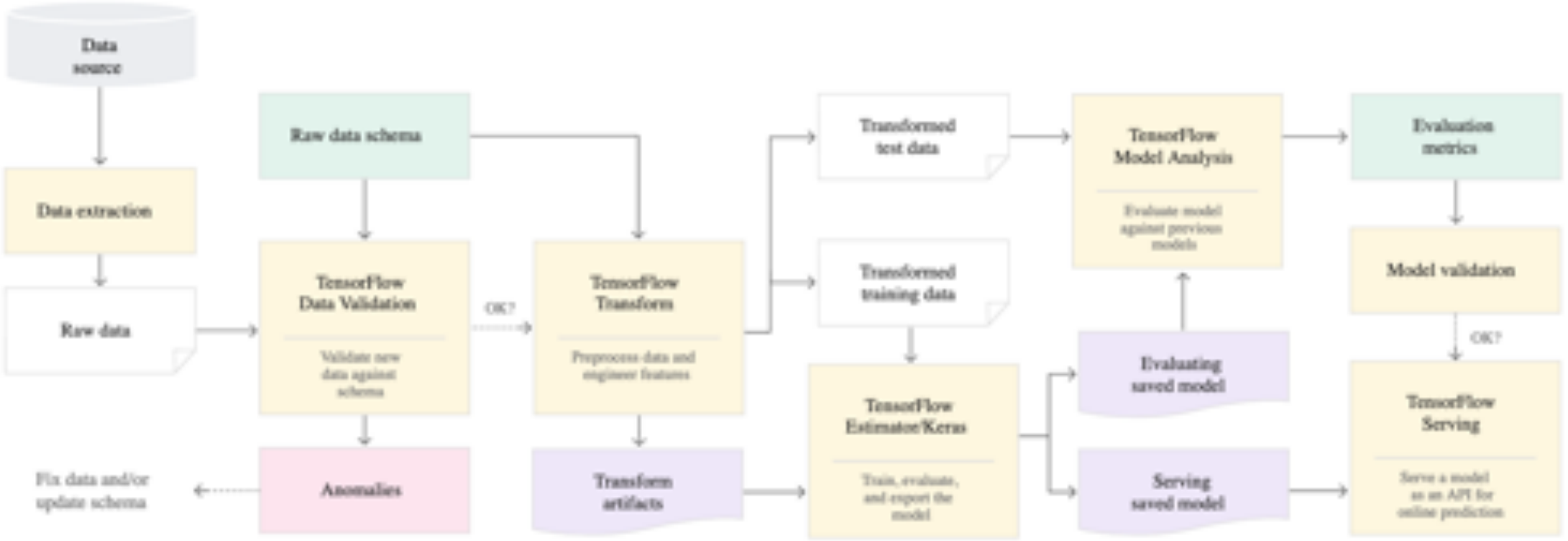

Overview
TFX Config
Airflow Runtime
Kubeflow Runtime
Data Analysis & Validation
ExampleValidator
StatisticsGen
SchemaGen
Training + Eval Data
ExampleGen
Transform
Trainer
Evaluator
Model Validator
Pusher
TensorFlow Serving
TensorFlow Hub
TensorFlow Lite
TensorFlow JS
Data Ingestion
Data Transformation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



https://cloud.google.com/solutions/machine-learning/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning



https://cloud-dot-google-developers.appspot.com/solutions/machine-learning/architecture-for-mlops-using-tfx-kubeflow-pipelines-and-cloud-build



https://cloud-dot-google-developers.appspot.com/solutions/machine-learning/architecture-for-mlops-using-tfx-kubeflow-pipelines-and-cloud-build

References

- Uber workflow link
- ML pipelines link
- Designing human-centered AI products link
- PAIR Facets documentation Github repository blog
- What-if tool link
- Lime documentation
- Shab Github repository

References

- Keras (TensorFlow high level API) link
- TensorFlow documentation Github repository
- TensorFlow Datasets link blog
- TensorFlow tf.data documentation documentation blog
- TensorFlow Text documentation
- Keras Text processing documentation
- Tensorflow Hub documentation modules
- TensorFlow Distributed Strategy documentation
- TensorFlow Estimator documentation
- TensorFlow saved and load documentation blog

References

- Keras Tuner documentation Github repository
- TensorBoard link
- TensorBoard.dev link
- tf-explain documenation
- TensorFlow Extended (TFX) documentation blog
- « Developing Production ML Pipelines » by Aurélien Géron @TensorFlow World 2019 link
- « An End-to-End MP Platform » by Clemens Mewald link

References

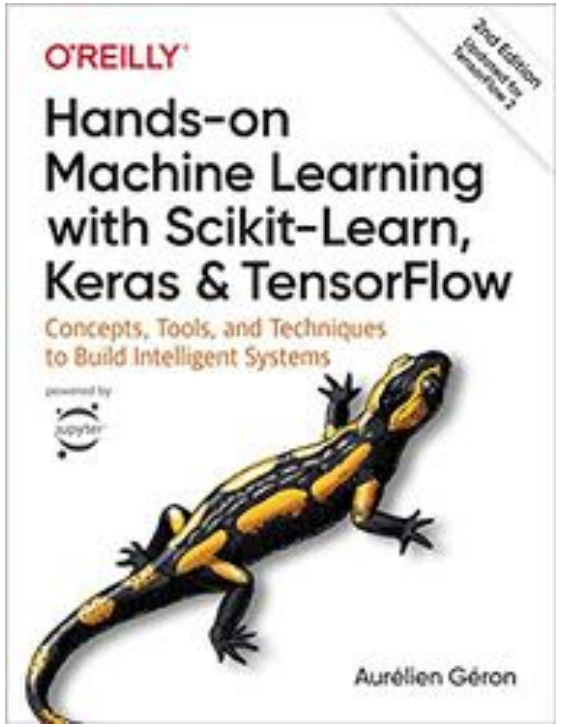
- Inside TensorFlow [videos] link
- TensorFlow World 2019 [videos] link
- TensorFlow Dev Summit 2019 [videos] link
- TensorFlow tutorial [video] link
- TensorFlow tutorial [video] link

- Cambridge Google Developers ML Summit 2019 [videos] link
- Kirkland Google Developers ML Summit 2019 [videos] link

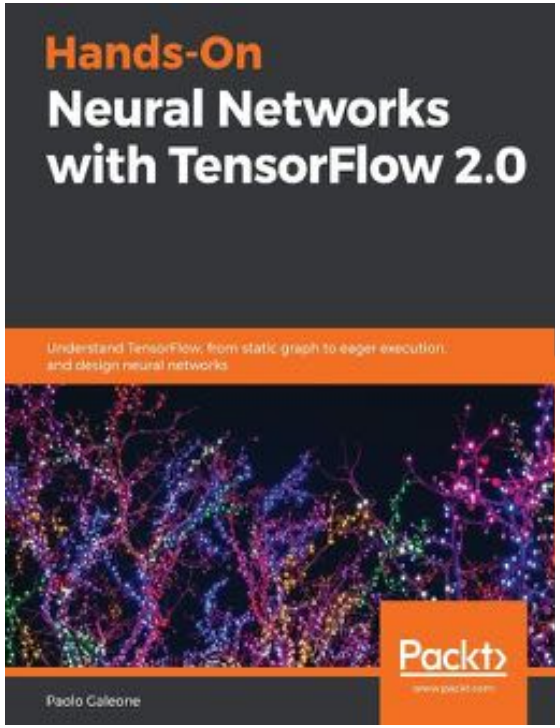
- AI Adventure: ML on Google Cloud Platform [videos] link

Books

TensorFlow 2.0

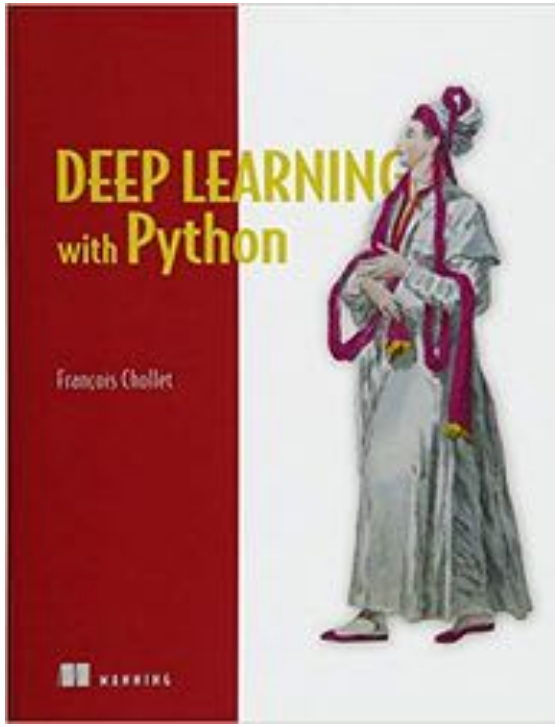


link



link

Updated version with TensorFlow 2.0 coming soon ?



link