Track Machine Learning Applications by mlflow Tracking

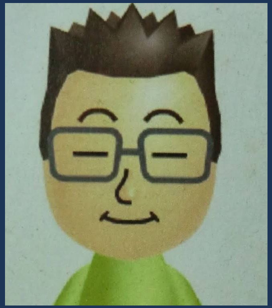
Shuhsi Lin @ PyConTW 2020

PYCON TW20

About Me

Working in a manufacturing company
With data and people
Focus on

- Agile/Engineering culture
- IoT applications
- Streaming process
- Data visualization



Shuhsi Lin

sucitw@gmail.com

https://medium.com/@suci/

Lurking in PyHug, Taipei.py and various Meetups

Agenda -What we will focus on

Logging

Logging & ML monitoring

ML Life Cycle

ML Life Cycle

MLOPS

MLflow Tracking

Track ML

- Basic logging
- Model logging
- Auto-logging

What we will not focus on

1. Details of ML Platform/ MLOps
    - Deployment/Operation/Administration
2. MLflow Projects/Models/Model Registry
3. Infrastructure Details
4. Comparison of Different Tools
5. Machine Learning Algorithms or Frameworks

Logging

Everything

Why Logging is important

Business Analytics

Stakeholder

- Auditing for business
- Product improvement from log statistics

Problem Solving

End-User

- Self-Troubleshooting

Developer

- Profiling for performance
- Debugging

Sysadmin

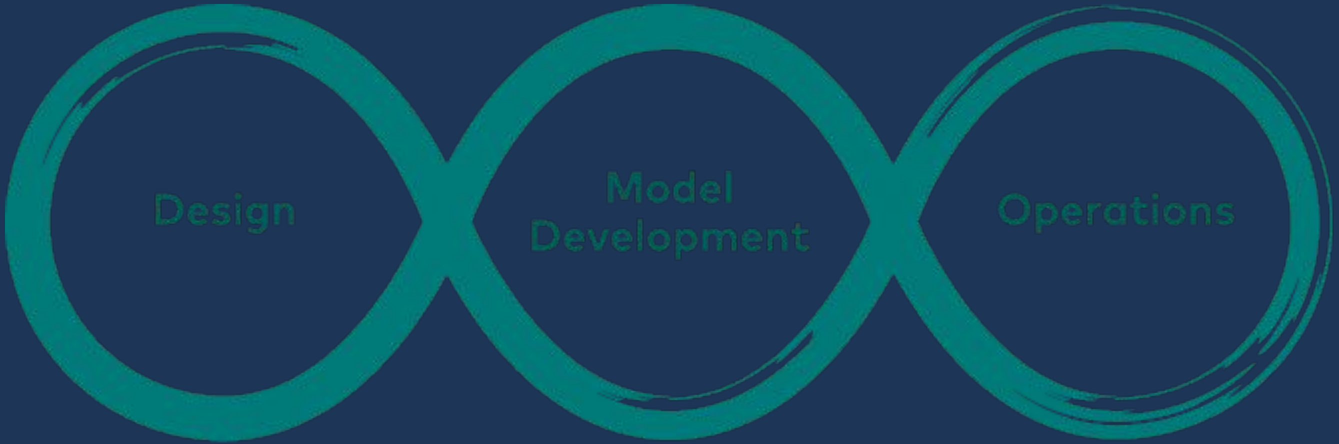
- Stability monitoring
- Troubleshooting

Security

- Auditing for security

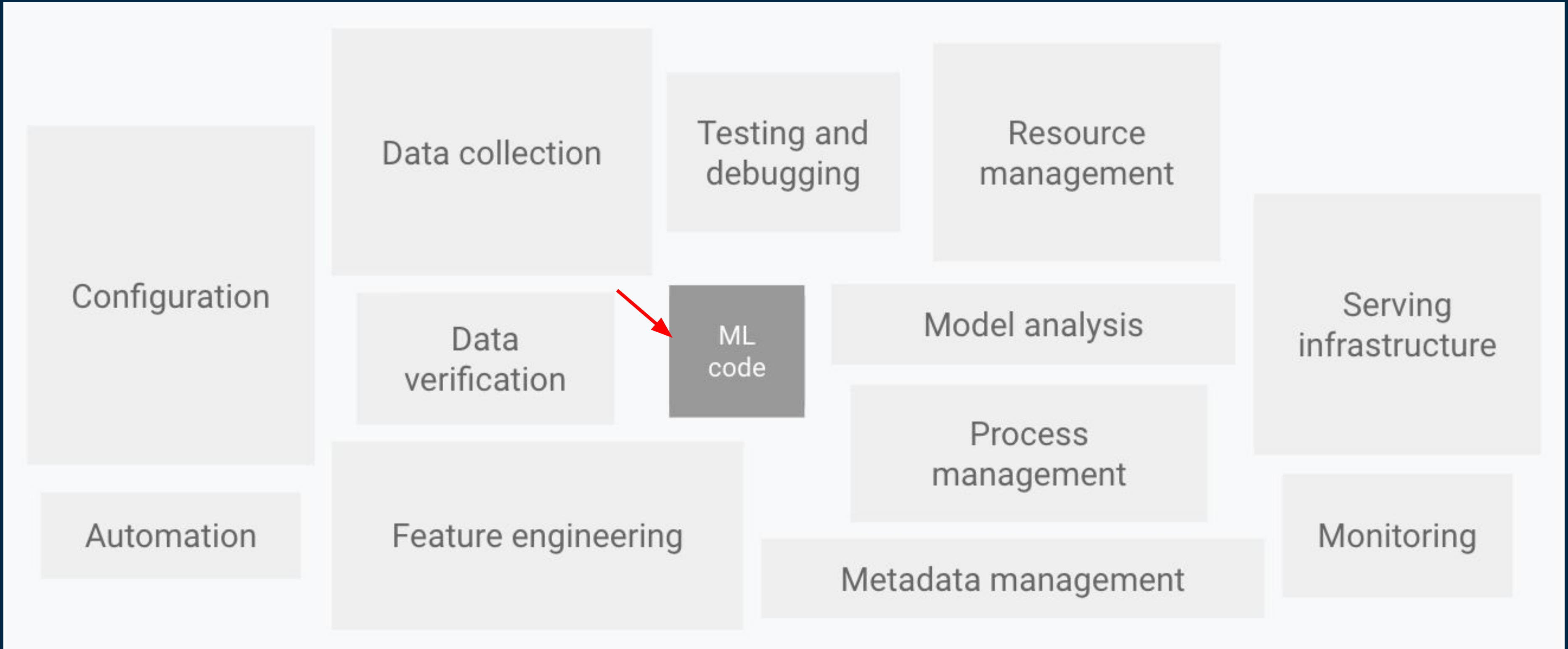
What is Logging in Machine Learning

ML Life Cycle



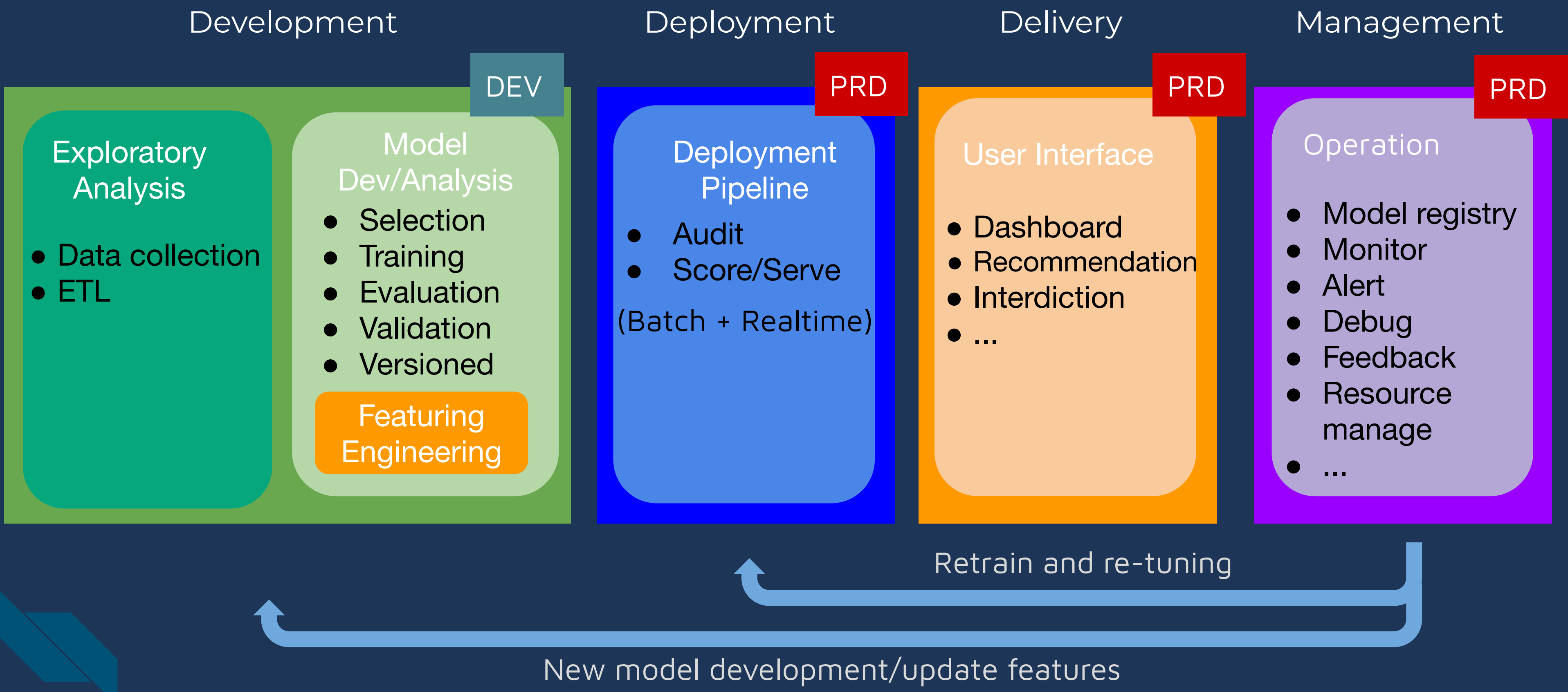
ML is’t just code

Elements for ML systems



MLOps: Continuous delivery and automation pipelines in machine learning
(originally adapted from Hidden Technical Debt in Machine Learning Systems)

ML Life Cycle



Inspired from "Enabling Scalable Data Science Pipelines with MLflow and Model Registry at Thermo Fisher Scientific"

The Maturity of the MLOPS Process

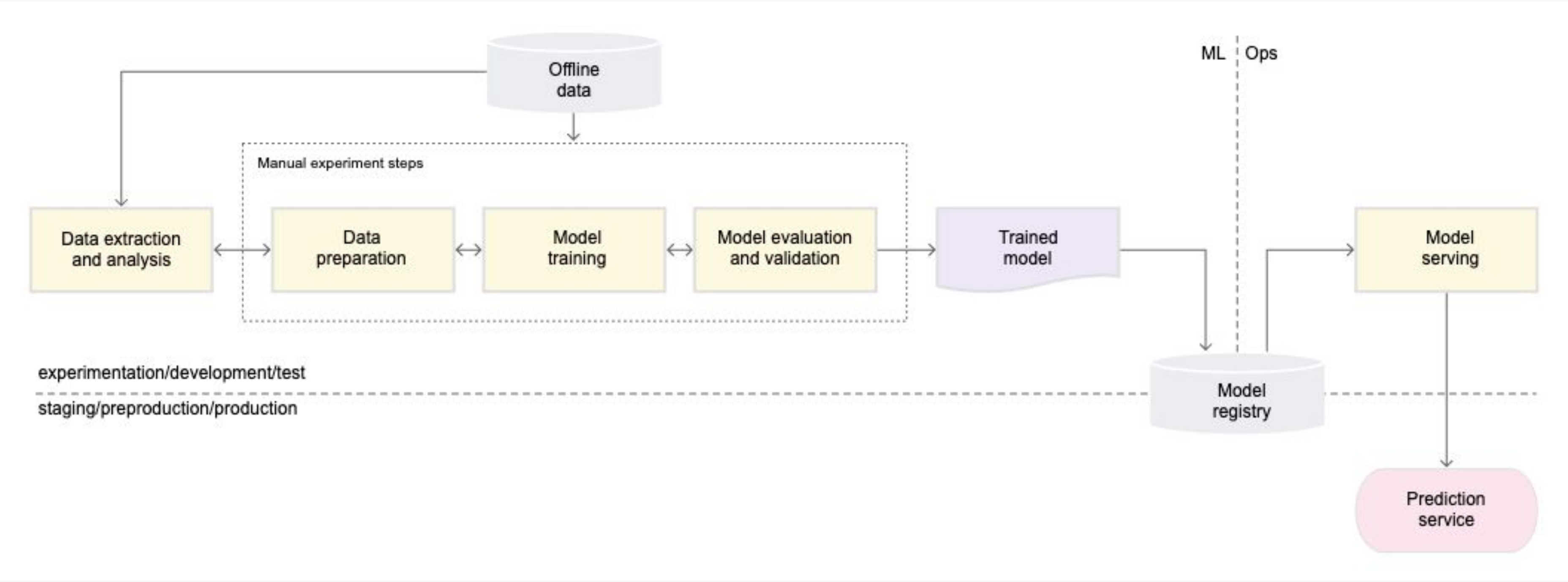
level 0	**Manual process**
level 1	**ML pipeline automation**
level 2	**CI/CD pipeline automation**

MLOps: DevOps principles to ML systems

- *MLOps* is an ML engineering culture and practice that aims at unifying ML system development (Dev) and ML system operation (Ops).
- Practicing MLOps means that you advocate for automation and monitoring at all steps of ML system construction, including integration, testing, releasing, deployment and infrastructure management.

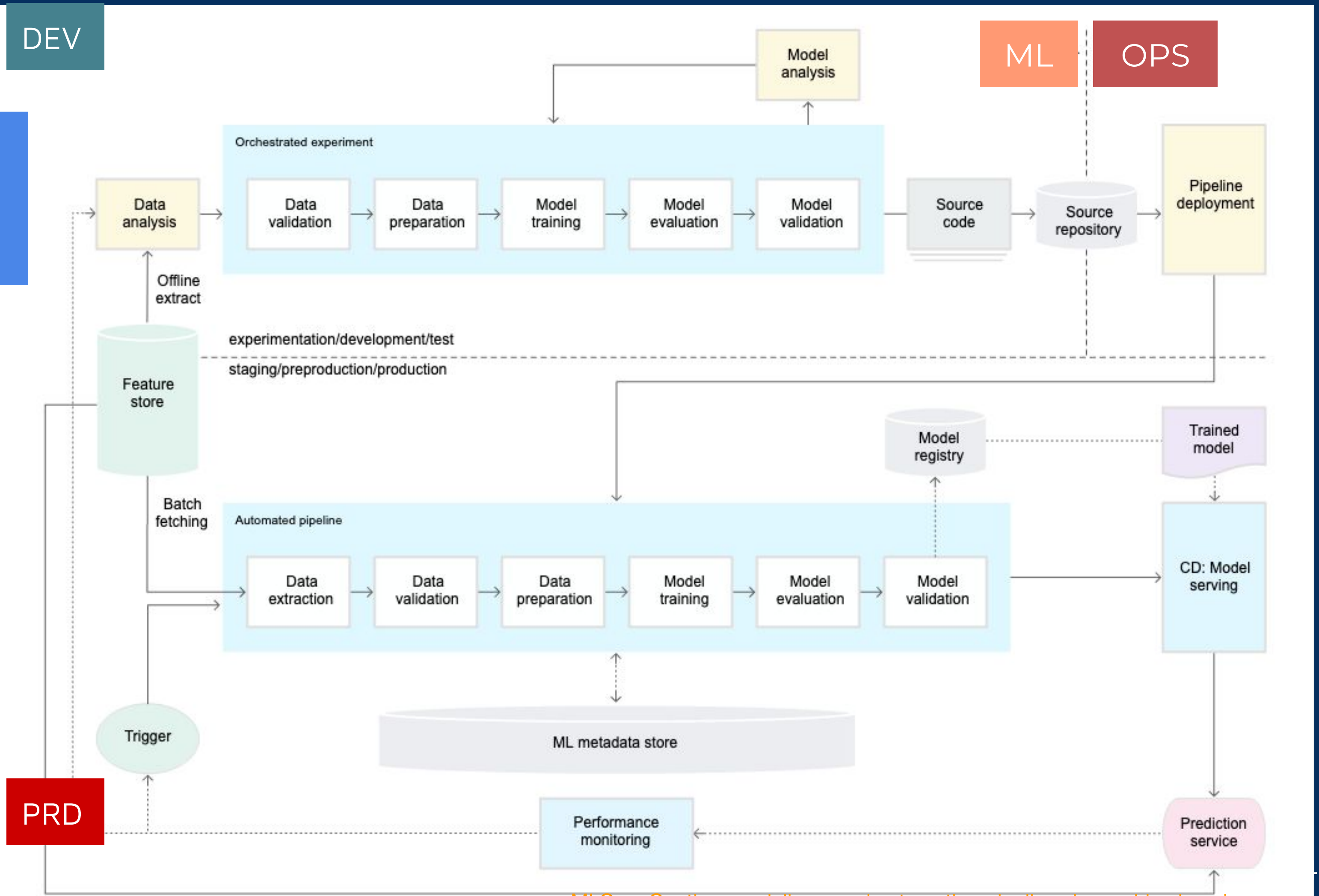
MLOps: Continuous delivery and automation pipelines in machine learning

MLOps level 0: Manual process



[MLOps: Continuous delivery and automation pipelines in machine learning]

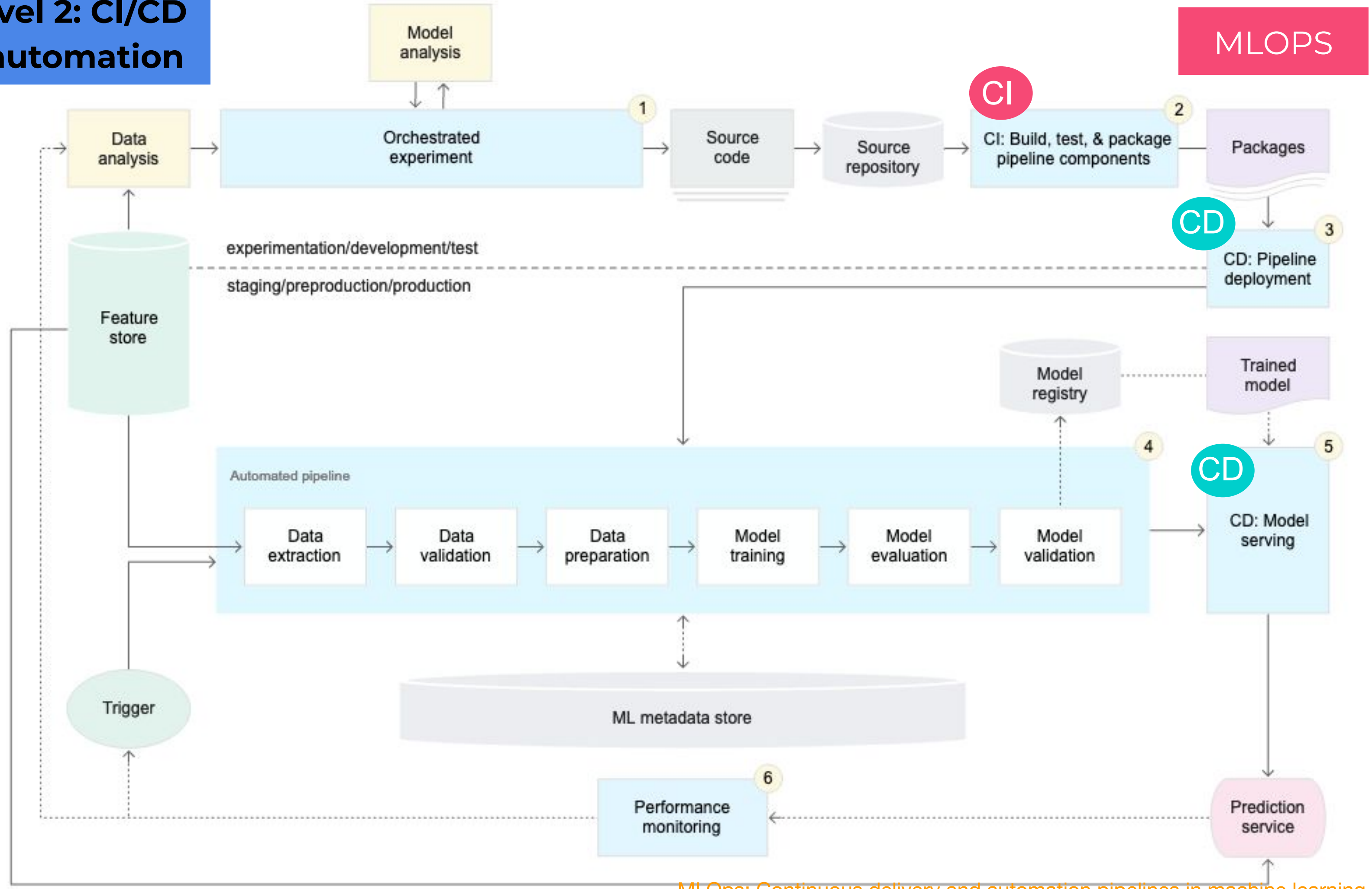
MLOps level 1: ML pipeline automation



[MLOps: Continuous delivery and automation pipelines in machine learning]()

MLOps level 2: CI/CD pipeline automation

MLOPS



MLOps: Continuous delivery and automation pipelines in machine learning

Experiment Tracking

Model development & Post-Deployment

- **Prove value of experiment**
  - Need baseline to show and compare
- **Collaborate**
  - Need to refer and access models and artifacts from other members
- **Reproduce work**
  - Need same parameters and model of ex-run

What we should log/track in ML

Log day-to-day work in ML life cycle

- Hyper parameters
- Training/modeling performances
- Model
  - Type
  - Building environment
  - Modeling version
- and so on

Evaluation metrics

- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- Root Mean Squared Error (RSME)
- R-squared (r2)
- ...

parameters

- Convolutional filter
- Kernel_size
- Max pooling
- Dropout
- Dense
- Batch_size
- Epochs
- ....

ML is complex
and need to be tracked

mlflow

https://github.com/mlflow

- Since 2018 from DataBricks (Main contributor)
- An open platform for the machine learning lifecycle
- Python Library; runs locally and on the cloud
- Built-in UI for experiment visualization
- Logging integrations for major frameworks: scikit-learn, PyTorch, TF,..

Collaboration with MLflow

Built-in integrations:

TensorFlow, PyTorch, Keras, Apache Spark, scikit learn, H2O.ai, python, R, Java, mleap, ONNX, GLUON, XGBoost, LightGBM, fast.ai, CONDA, kubernetes, docker, Amazon SageMaker, Azure Machine Learning, Google Cloud, databricks

Organizations using and contributing to MLflow:

databricks, Microsoft, RStudio, University of Washington, Virgin hyperloop one, Booking.com, TOYOTA Southeast Toyota Finance, Zillow, SK telecom, accenture, criteo, Outreach, splice MACHINE, Telia Company, TEKION, Bering, SCRY ANALYTICS, cnrs, WIX.com, ABRAXAS Intelligence, TIGER SHARK, virtusa, Health 2 Sync, mined.ai, dL, SYSTEMATIC, despegar.com, ModelWorks, Brandfolder, ASML, fractal, Acerta, ENABLE AI, Linde, TENPOINT7, plain concepts, xpand it, Kobalt, CognitOps, DATA INSIGHTS, MOTOROLA SOLUTIONS, TIGER ANALYTICS, JULO, publicis sapient, agremo, servian, BLECKWEN, landscape, edgeverve, solocal, nunam, Kardiolabs AI, STRATIO, Quantmetry, EMERTONDATA

mlflow Components

MLflow is an open source platform to manage the ML lifecycle, including experimentation, reproducibility, deployment, and a central model registry. MLflow currently offers four components:

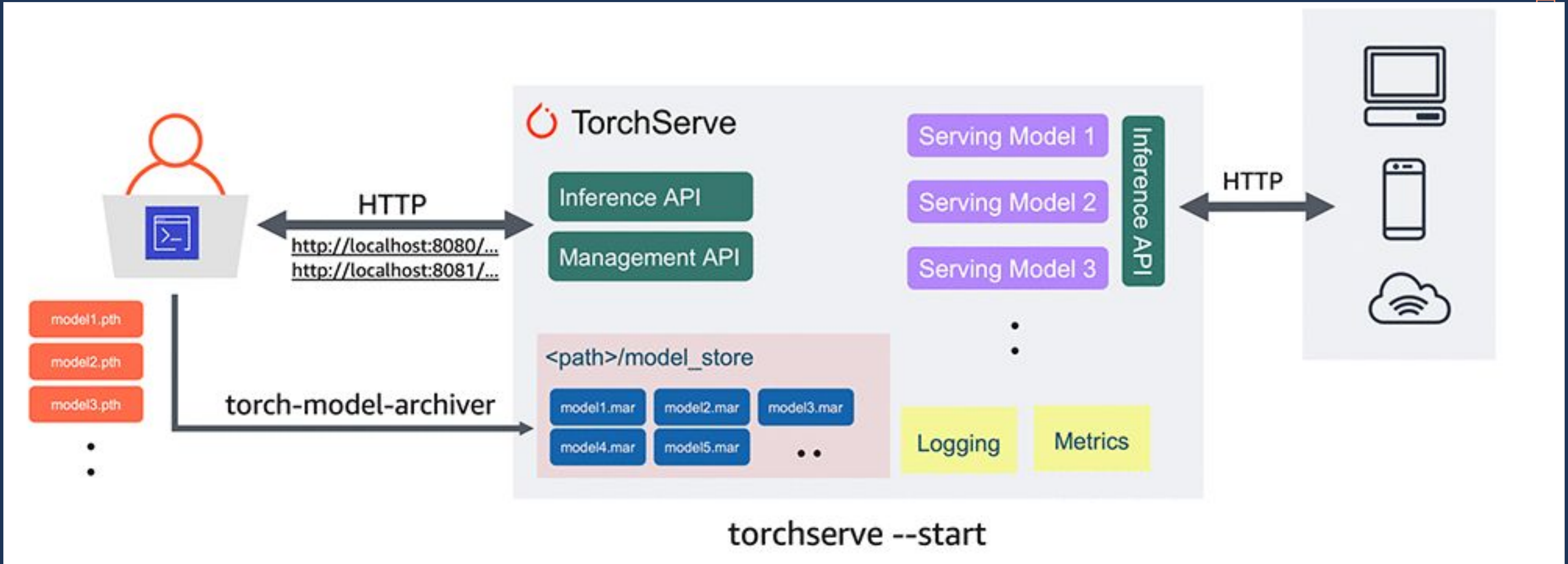
MLflow Tracking	MLflow Projects	MLflow Models	Model Registry
Record and query experiments: code, data, config, and results	Package data science code in a format to reproduce runs on any platform	Deploy machine learning models in diverse serving environments	Store, annotate, discover, and manage models in a central repository
Read more	Read more	Read more	Read more

Main focus of this sharing!

Similar Tools

- Neptune (commercial)
- Tensorboard (+MLflow.tensorflow)
- TorchServe (+ MLflow.pytorch)
- Kubeflow (Meta)
- Data Science Workbench
- ...

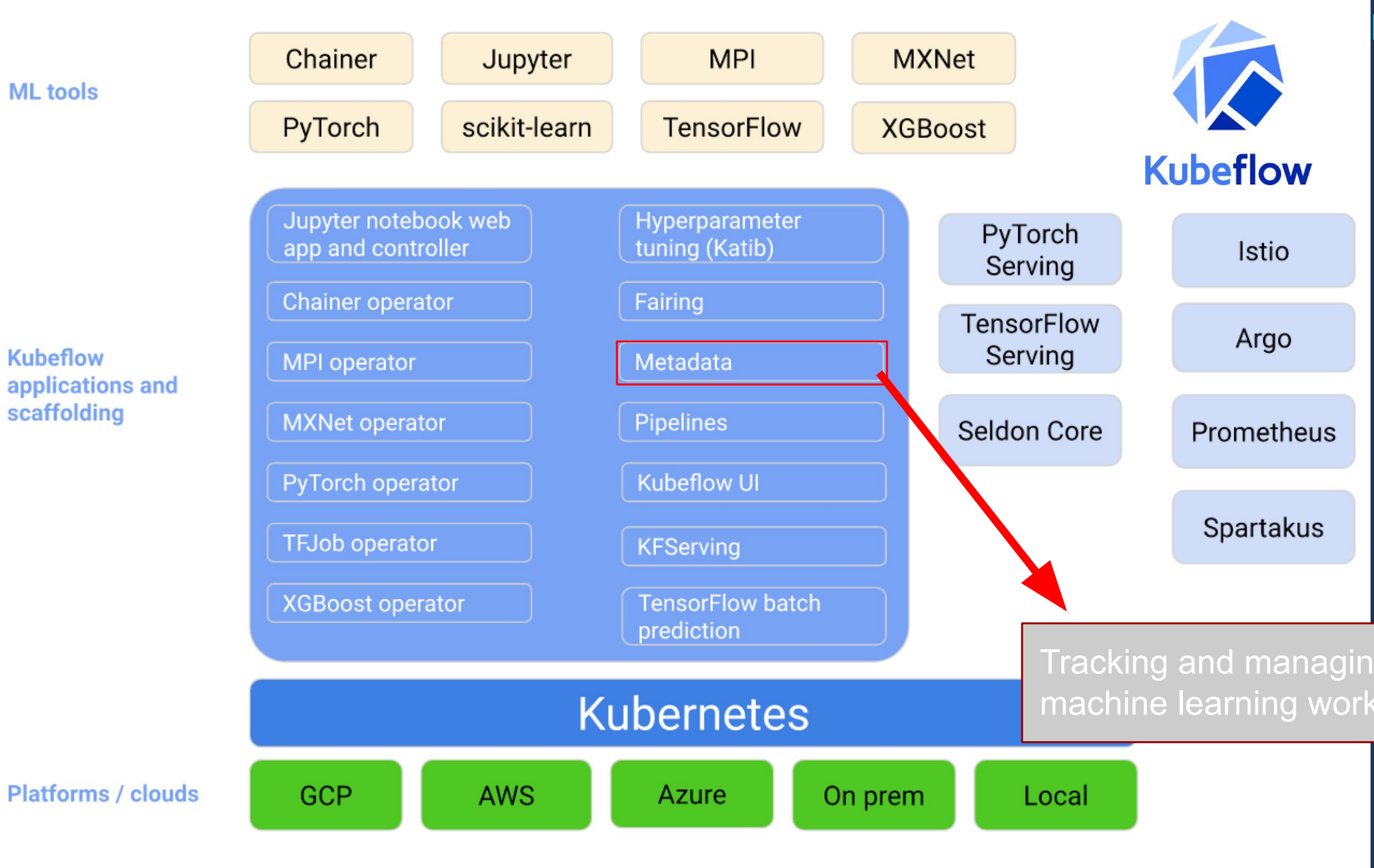
TorchServe



Facebook + AWS

https://aws.amazon.com/tw/blogs/machine-learning/deploying-pytorch-models-for-inference-at-scale-using-torchserve/

Kubeflow



Tracking and managing metadata of machine learning workflows in Kubeflow

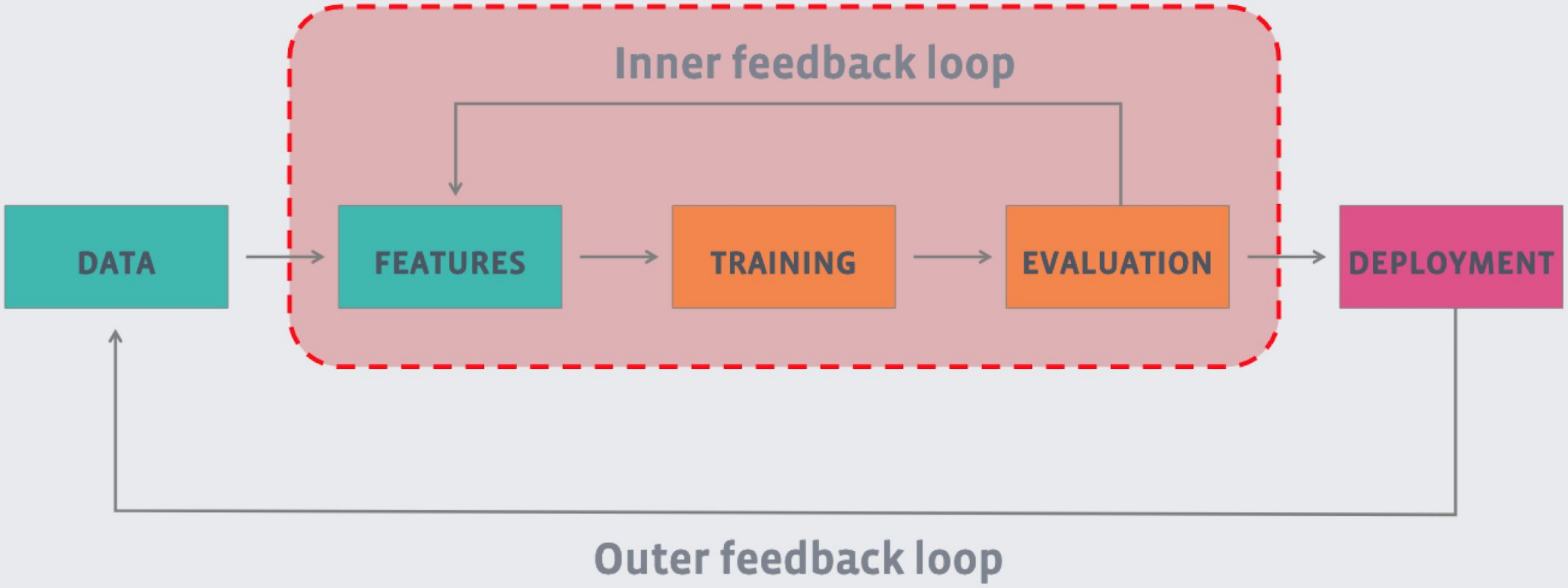
https://www.kubeflow.org/docs/started/kubeflow-overview/

Getting Started with

mlflow Tracking

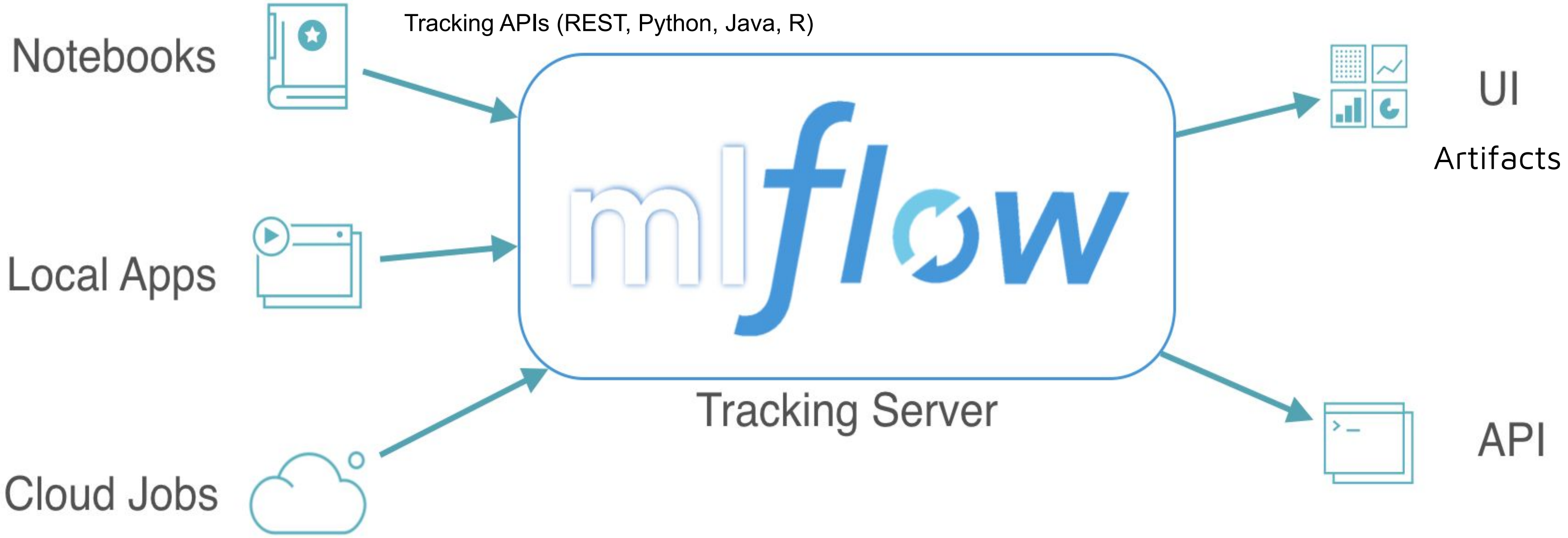
MLflow Components

Experiment tracking



Experiment/Production pipeline

Experiment and metric tracking



Terminology

Project

Experiments

Run Run Run Run

Run Run Run Run

Run

Entity

- Code version
- Start and end time
- Source
- (Hyper) Parameters
- Metrics
- Tags/Notes

Backend stores

- File store
- Database

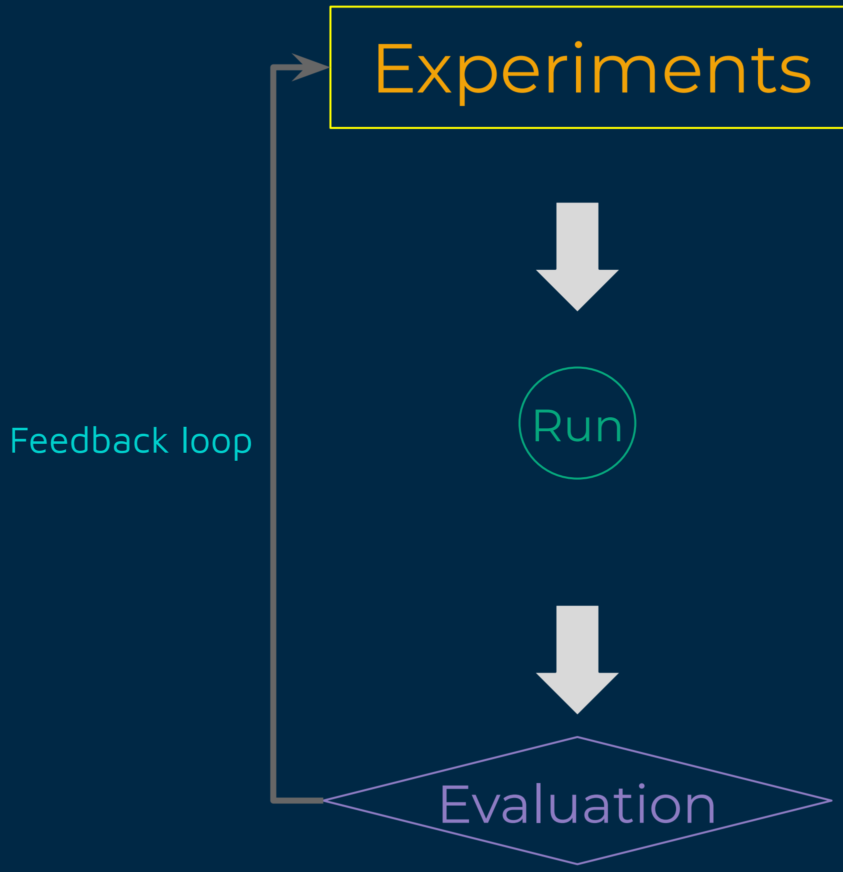
Artifacts

- Output files

a. Images
b. Pickled models
c. Data files...

File storage

- Amazon S3
- Azure Blob Storage
- Google Cloud Storage
- FTP server
- SFTP Server
- NFS
- HDFS



Setup & Initialize

- Data preparation
- Setup MLflow experiment

Train & Inference

- Start a Run
- Log (Hyper)parameter
- Log metrics
- Log artifact
- Log model

Post-analysis

- Compare runs/ Tuning (parameters/models)

MLflow Tracking for ML Development

Tracking.API

- start_Run()
- log_param()
- log_metric()
- log_artifact()
- end_Run()

Output

- Parameters
- Metrics
- Output file
    - Artifact
- Code version
- ...

- log_model

- Model

mlflow Tracking

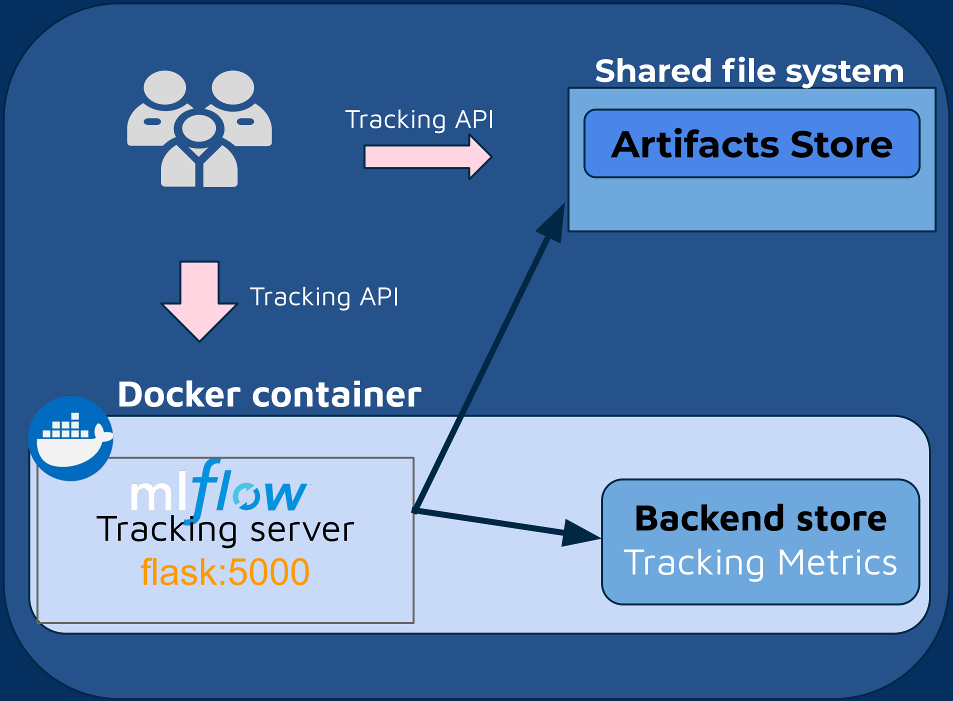
~~DEMO~~ Examples

https://github.com/sucitw/mlflow_tracking

Example Architecture

```
mlflow server \
    --backend-store-uri  $FILE_STORE \
    --default-artifact-root $ARTIFACT_STORE \
    --host $SERVER_HOST \
    --port $SERVER_PORT
```

```
docker run -d -p 5000:5000 \
        -v /tmp/artifactStore:/tmp/mlflow/artifactStore \
        --name mlflow-tracking-server \
        suci/mlflow-tracking
```



https://github.com/sucitw/mlflow_tracking

Tracking- Experiments

Setup & Initialize

```
# Setup & Initialize MLflow experiment
 experiment_name = "PyconTW 2020 Demo "
 tracking_server = "http://localhost:5000"

 mlflow.set_tracking_uri(tracking_server)
 mlflow.set_experiment(experiment_name)
```

```
#System Env setting

  #backend-store-uri
     $FILE_STORE
  #default-artifact-root
     $ARTIFACT_STORE
  #host
     $SERVER_HOST
  #port
     $SERVER_PORT
```

mlflow

Basic Logging

Tracking- Run

Train & Inference

```
with mlflow.start_run() as run:

    # Log a parameter (key-value pair)
    log_param("param1", randint(0, 100))

    # Log a metric; metrics can be updated throughout the run
    log_metric("metricsA", random())
    log_metric("metricsA", random() + 1)
    log_metric("metricsA", random() + 2)
    log_metric("metricsB", random() + 2)

    # Log an artifact (output file)
    with open("outputs/test.txt", "w") as f:
        f.write("hello world! Run id:{}".format(type(mlflow.active_run().info)))

    log_artifacts("outputs")
```

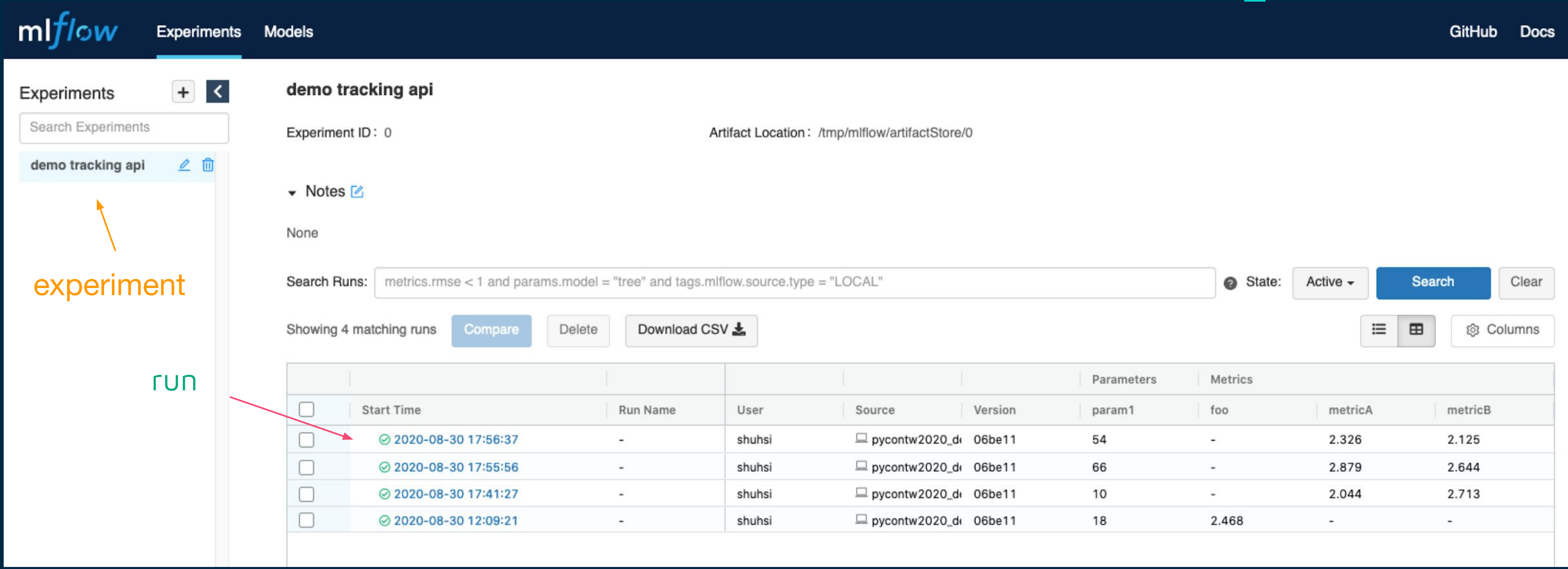
→ **Log (Hyper)parameter**

→ **Log metrics**

→ **Log artifact**

mlflow

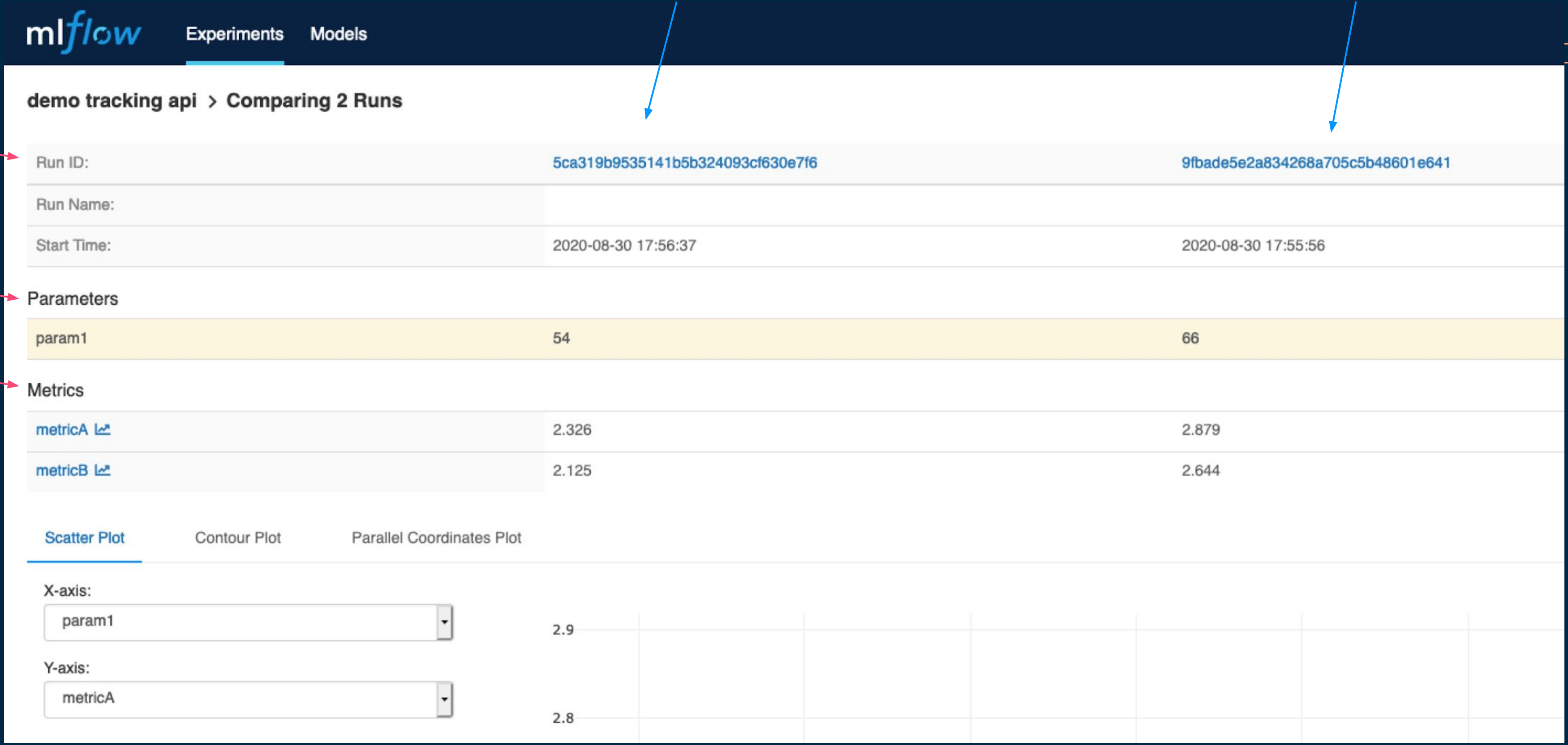
Tracking UI


mlflow
Experiments
Models
GitHub
Docs
Experiments
Search Experiments
demo tracking api
experiment
run
demo tracking api
Experiment ID : 0
Artifact Location : /tmp/mlflow/artifactStore/0
Notes
None
Search Runs:
metrics.rmse < 1 and params.model = "tree" and tags.mlflow.source.type = "LOCAL"
State:
Active
Search
Clear
Showing 4 matching runs
Compare
Delete
Download CSV
Columns
Parameters
Metrics
Start Time
Run Name
User
Source
Version
param1
foo
metricA
metricB
2020-08-30 17:56:37 - shuhsi pycontw2020_d 06be11 54 - 2.326 2.125
2020-08-30 17:55:56 - shuhsi pycontw2020_d 06be11 66 - 2.879 2.644
2020-08-30 17:41:27 - shuhsi pycontw2020_d 06be11 10 - 2.044 2.713
2020-08-30 12:09:21 - shuhsi pycontw2020_d 06be11 18 2.468 - -

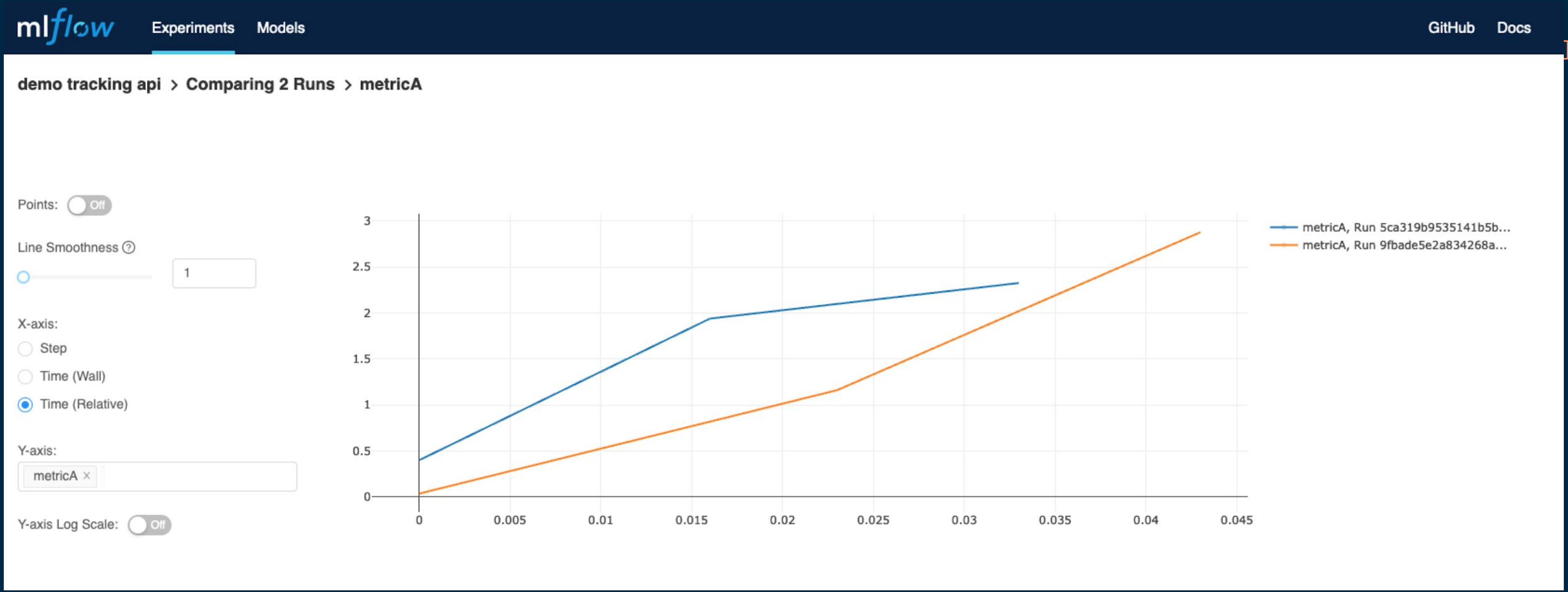
Compare Two Runs

Run A

Run B



Compare Two Runs



Model Logging

Model

Log->Load->deploy

```
mlflow.<model-type>.log_model(model, ...)

mlflow.<model-type>.load_model(modelpath)

mlflow.<model-type>.deploy()
```

```
mlflow.sklearn.log_model(lr, "model")
```

Built-In Model Flavors <model-type>

- Python Function (python_function)
- R Function (crate)
- H_2O (h2o)
- Keras (keras)
- MLeap (mleap)
- PyTorch (pytorch)
- Scikit-learn (sklearn)
- Spark MLlib (spark)
- TensorFlow (tensorflow)
- ONNX (onnx)
- MXNet Gluon (gluon)
- XGBoost (xgboost)
- LightGBM (lightgbm)
- Spacy(spaCy)
- Fastai(fastai)

Artifacts

- model
  - MLmodel
  - conda.yaml
  - model.pkl

Full Path: /tmp/mlflow/artifactStore/4/557a222e5c8e454689a112cd…

Size: 349B


```yaml
artifact_path: model
flavors:
  python_function:
    env: conda.yaml
    loader_module: mlflow.sklearn
    model_path: model.pkl
    python_version: 3.7.1
  sklearn:
    pickled_model: model.pkl
    serialization_format: cloudpickle
    sklearn_version: 0.23.2
run_id: 557a222e5c8e454689a112cd54dd7ff7
utc_time_created: '2020-09-04 00:21:11.388959'
```


https://github.com/sucitw/mlflow_tracking/blob/master/pytw2020_demo_SK_model.py

Artifacts

- model
  - MLmodel
  - conda.yaml
  - model.pkl

Full Path: /tmp/mlflow/artifactStore/4/557a222e5c8e454689a112cd...
Size: 150B


```yaml
channels:
- defaults
- conda-forge
dependencies:
- python=3.7.1
- scikit-learn=0.23.2
- pip
- pip:
  - mlflow
  - cloudpickle==1.5.0
name: mlflow-env
```


https://github.com/sucitw/mlflow_tracking/blob/master/pytw2020_demo_SK_model.py

mlflow

Automatic Logging

Automated MLflow Tracking

Currently supported libraries:

TensorFlow, Keras, GLUON, LightGBM, dmlc XGBoost, Apache Spark, fast.ai

Coming soon:

PyTorch, scikit learn

https://www.mlflow.org/docs/latest/tracking.html#automatic-logging

Auto-Logging - TF

Manually logging

```
import mlflow

mlflow.log_param("layers", layers)
model = train_model()
mlflow.log_metric("mse", model.mse())
mlflow.log_artifact("plot", plot(model))
mlflow.tensorflow.log_model(model)
```

With autologging

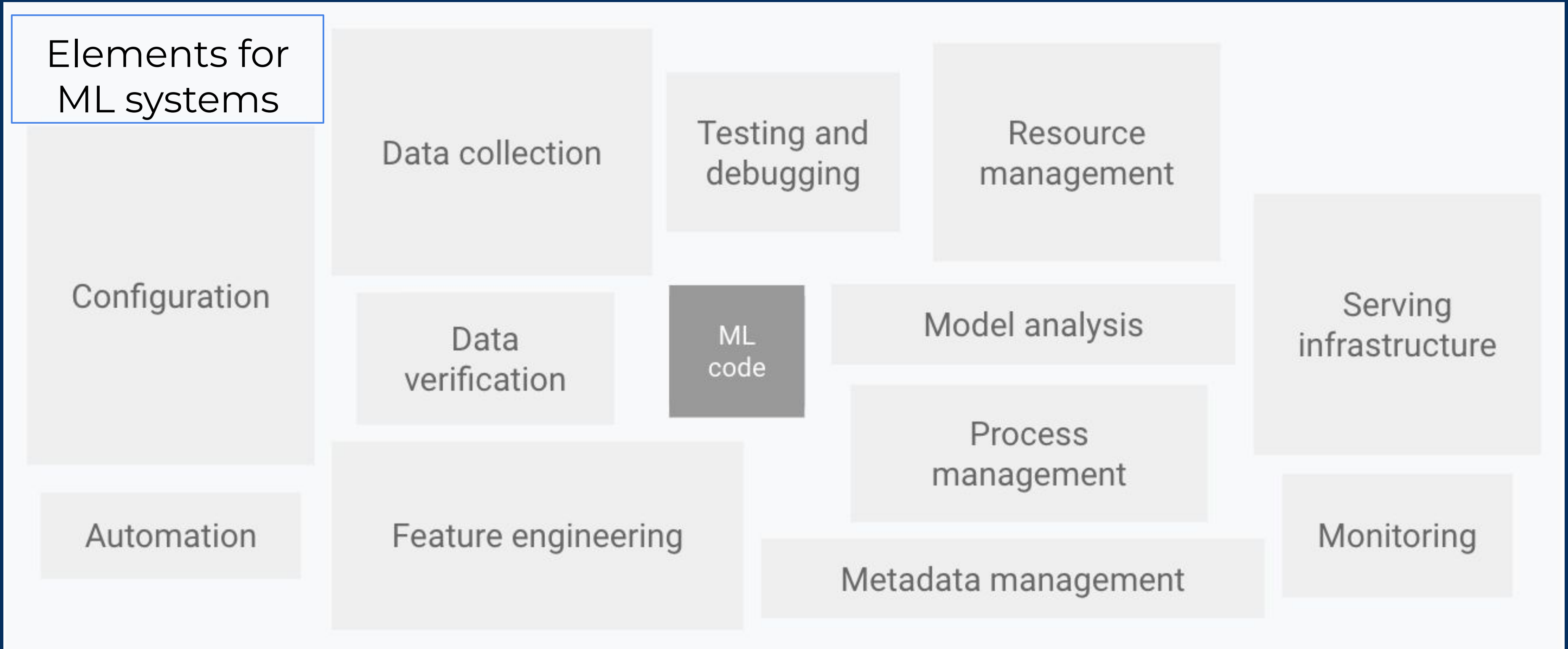
```
import mlflow

mlflow.tensorflow.autolog()
model = train_model()
```

Capture TensorBoard metrics

https://cs.stanford.edu/people/matei/papers/2020/deem_mlflow.pdf

Recap

ML is COMPLEX and need to be Tracked



MLOps: Continuous delivery and automation pipelines in machine learning
(originally adapted from Hidden Technical Debt in Machine Learning Systems)

What mlflow can help

Managing the machine learning lifecycle - (Experiment) Tracking

Easy use in at the same way

- Remotely /cloud or locally
- Individual, team or large orgs

Tracking Metrics

- Simplified tracking for ML models means faster time to insights and value
- Integrated with popular ML library & languages

Model management

- Launch of model registry enhances governance and core proposition of model management.

More mlflow

- MLflow Tracking
- MLflow Project
- MLflow Models
- MLflow Model registry
- MLflow Deployment

Experiment tracking

Model deployment/serving

Model governance

Google Cloud Platform

RedisAI

kubernetes

Amazon SageMaker

Azure
Machine Learning

Reference

- MLflow official Doc
- MLflow tracking
- Learning MLflow
    - 2020_Workshop | Managing the Complete Machine Learning Lifecycle with MLflow (DataBricks)
- MLOps: Continuous delivery and automation pipelines in machine learning
- 2020 Spark Summit: Enabling Scalable Data Science Pipelines with MLflow and Model Registry at Thermo Fisher Scientific
- 2020 DEEM workshop: Developments in MLflow: A System to Accelerate the Machine Learning Lifecycle
- Example codes of this talk (Github)

Thanks!