



Data + AI
Online Meetup Group

mlflow

Platform for Complete Machine
Learning Lifecycle

Jules S. Damji
@2twitme

San Francisco | May 13, 2020: Part 2 of 3 Series

Outline – Introduction to MLflow: Understanding MLflow Projects and Models- Part 2

- Review & Recap Part 1: MLflow Tracking
- Concepts and Motivations
- MLFlow Component
 - MLflow Projects & Models
 - Use MLflow on Databricks Community Edition (DCE)
 - Explore MLflow UI
 - Tutorials
- Q & A

<https://dbricks.co/mlflow-part-2>

Traditional Software vs. Machine Learning

Traditional Software

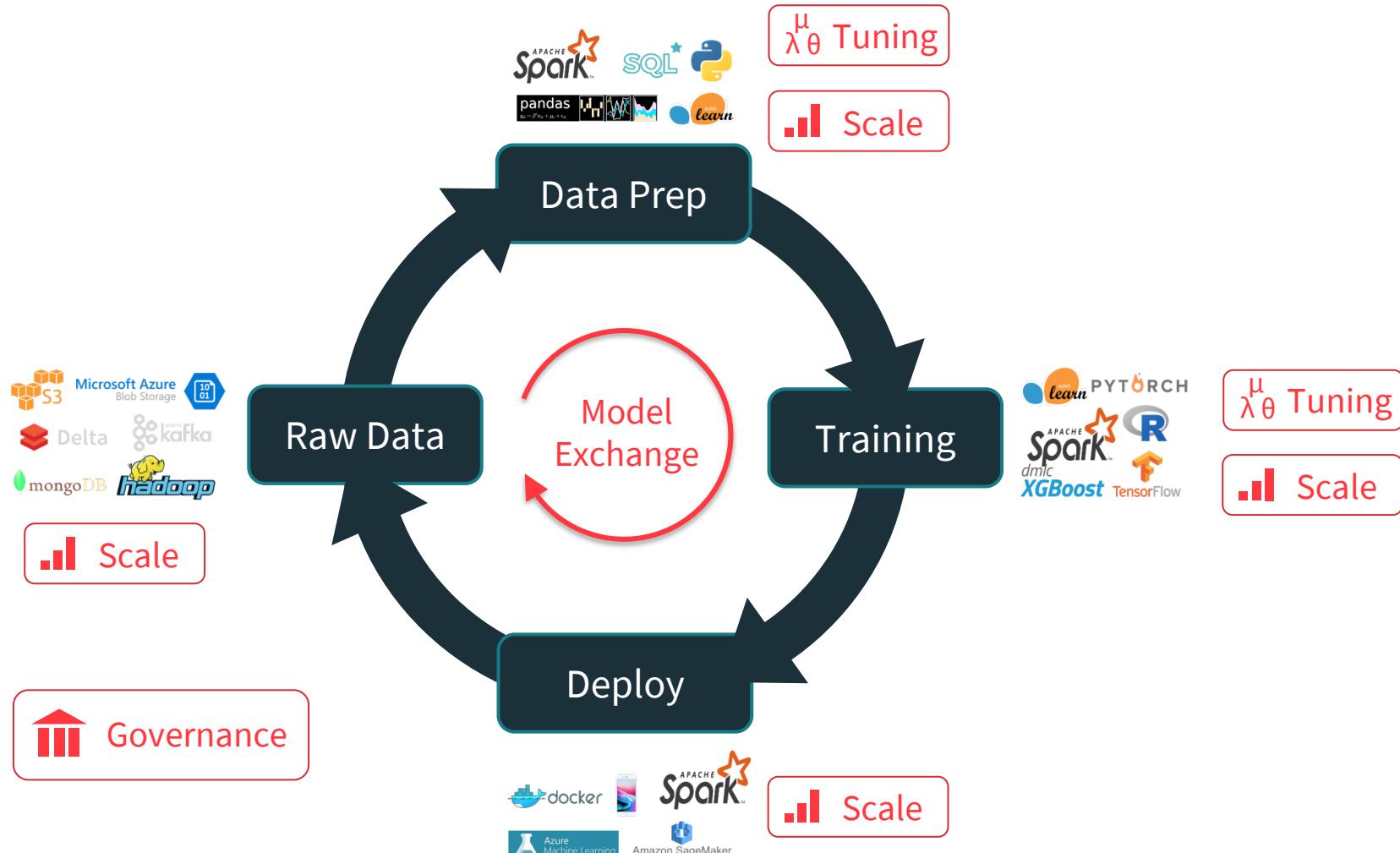
- Goal: Meet a functional specification
- Quality depends only on code
- Typically pick one software stack w/ fewer libraries and tools

Machine Learning

- Goal: Optimize metric(e.g., accuracy). Constantly experiment to improve it
- Quality depends on input data and tuning parameters
- Compare + combine many libraries, model



Machine Learning Lifecycle



MLflow Components

mlflow Tracking

Record and query experiments: code, data, config, and results

mlflow Projects

Package data science code in a format that enables reproducible runs on any platform

mlflow Models

Deploy machine learning models in diverse serving environments environments

new

mlflow Model Registry

Store, annotate and manage models in a central repository

[databricks.com
/mlflow](https://databricks.com/mlflow)



mlflow.org



github.com/mlflow



twitter.com/MLflow

Key Concepts in MLflow Tracking

Parameters: key-value inputs to your code

Metrics: numeric values (can update over time)

Tags and Notes: information about a run

Artifacts: files, data, and models

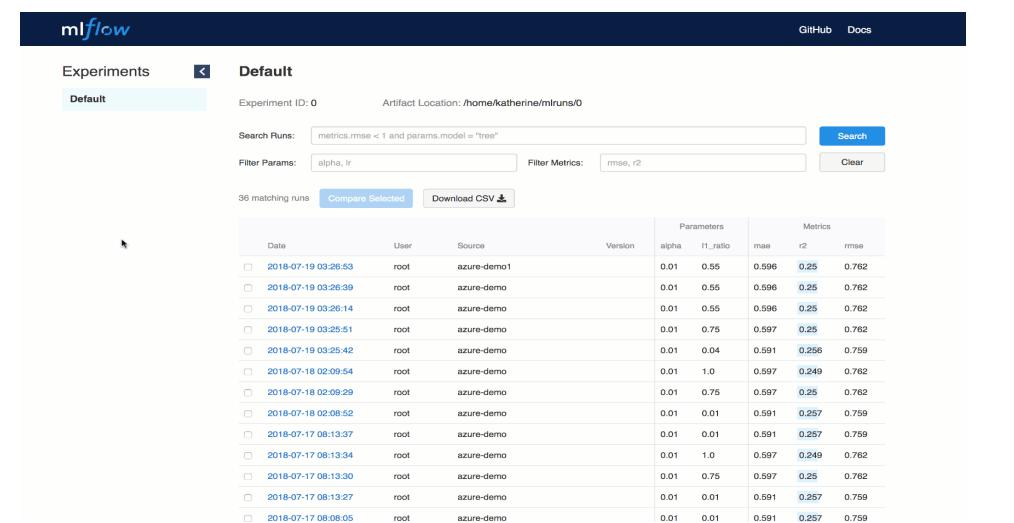
Source: what code ran?

Version: what of the code?

Model Development with MLflow is Simple!

```
data      = load_text(file)
ngrams   = extract_ngrams(data, N=n)
model    = train_model(ngrams,
                      learning_rate=lr)
score    = compute_accuracy(model)
with mlflow.start_run() as run:
    mlflow.log_param("data_file", file)
    mlflow.log_param("n", n)
    mlflow.log_param("learn_rate", lr)
    mlflow.log_metric("score", score)
    mlflow.sklearn.log_model(model)
```

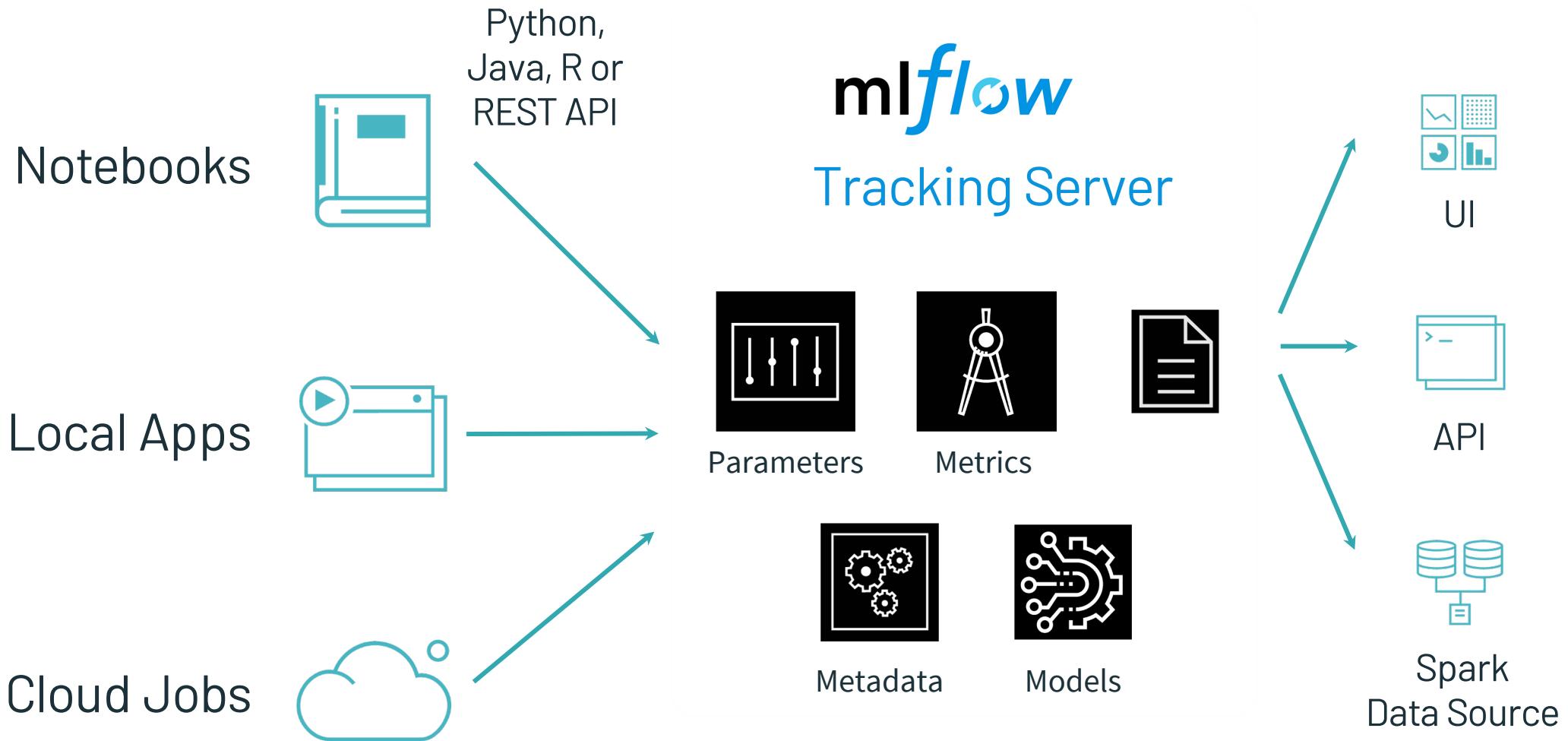
```
$ mlflow ui
```



The screenshot shows the MLflow UI interface. At the top, there's a search bar with the query "metrics.rmse < 1 and params.model = 'tree'". Below the search bar, there are two filter inputs: "Filter Params: alpha, lr" and "Filter Metrics: rmse, r2". A "Clear" button is also present. The main area displays a table of 36 matching runs. The columns in the table are Date, User, Source, Version, Parameters, and Metrics. The Parameters column includes alpha and l1_ratio, while the Metrics column includes rmse, r2, and rmse.

Date	User	Source	Version	Parameters	Metrics
2018-07-19 03:26:53	root	azure-demo1	0.01	0.55 0.01	0.596 0.25 0.762
2018-07-19 03:26:39	root	azure-demo	0.01	0.55	0.596 0.25 0.762
2018-07-19 03:26:14	root	azure-demo	0.01	0.55	0.596 0.25 0.762
2018-07-19 03:25:51	root	azure-demo	0.01	0.75	0.597 0.25 0.762
2018-07-19 03:25:42	root	azure-demo	0.01	0.04	0.591 0.256 0.759
2018-07-18 02:09:54	root	azure-demo	0.01	1.0	0.597 0.249 0.762
2018-07-18 02:09:29	root	azure-demo	0.01	0.75	0.597 0.25 0.762
2018-07-18 02:08:52	root	azure-demo	0.01	0.01	0.591 0.257 0.759
2018-07-17 08:13:37	root	azure-demo	0.01	0.01	0.591 0.257 0.759
2018-07-17 08:13:34	root	azure-demo	0.01	1.0	0.597 0.249 0.762
2018-07-17 08:13:30	root	azure-demo	0.01	0.75	0.597 0.25 0.762
2018-07-17 08:13:27	root	azure-demo	0.01	0.01	0.591 0.257 0.759
2018-07-17 08:08:05	root	azure-demo	0.01	0.01	0.591 0.257 0.759

MLflow Tracking



```
$ export MLFLOW_TRACKING_URI <URI>  
mlflow.set_tracking_uri(URI)
```

MLflow Components

mlflow Tracking

Record and query experiments: code, data, config, and results

mlflow Projects

Package data science code in a format that enables reproducible runs on any platform

mlflow Models

Deploy machine learning models in diverse serving environments environments

new

mlflow Model Registry

Store, annotate and manage models in a central repository

databricks.com
/mlflow



mlflow.org



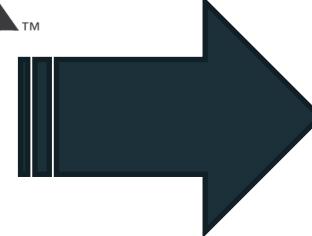
github.com/mlflow



twitter.com/MLflow

MLflow Projects Motivation

Diverse set of tools



Diverse set of environments

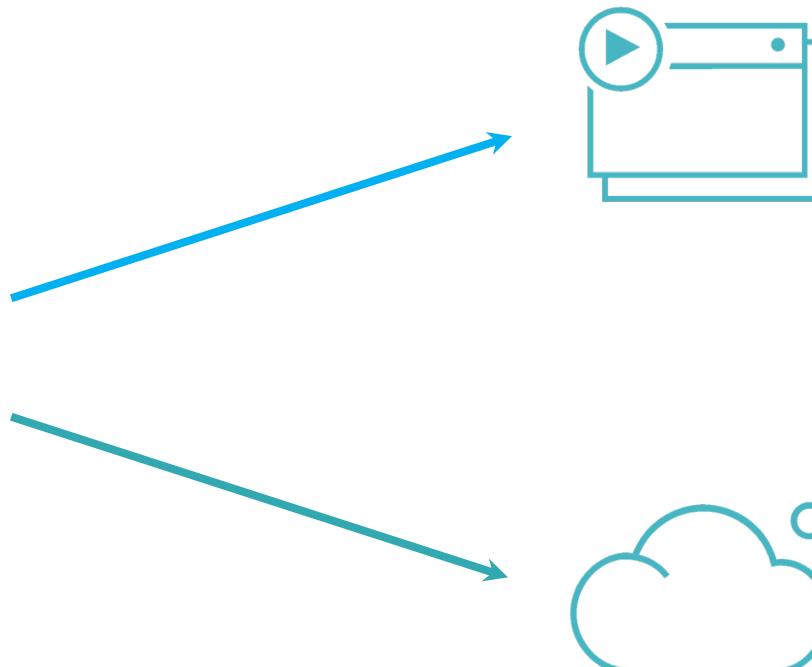
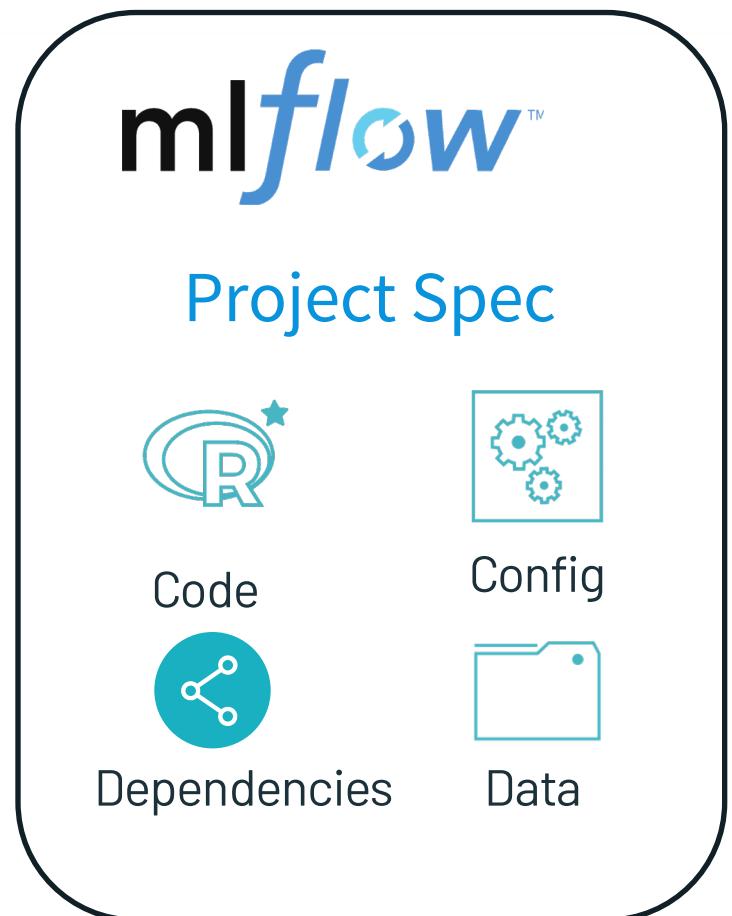


mlflow™
Projects

Package data science
code in a format that
enables reproducible runs
on any platform

Challenge: ML results difficult to reproduce

MLflow Projects



Local Execution



Remote Execution



Example MLflow Project

```
my_project/
└── MLProject
    ├── conda.yaml
    ├── main.py
    └── model.py
...

```

```
conda_env: conda.yaml

entry_points:
  main:
    parameters:
      training_data: path
      lambda: {type: float, default: 0.1}
  command: python main.py {training_data} {lambda}
```

```
$ mlflow run git://<my_project> -P lambda=0.2
mlflow.run("git://<my_project>", ...)
mlflow run . -e main -P lambda=0.2
```

Example

```
my_project/
    └── MLproject
        |
        └── conda.yaml
        └── main.py
        └── model.py
    ...

```

```
channels:
  - defaults
dependencies:
  - python=3.7.3
  - scikit-learn=0.20.3
  - pip:
    - mlflow
    - cloudpickle==0.8.0
name: mlflow-env
```

MLflow Components

mlflow Tracking

Record and query experiments: code, data, config, and results

mlflow Projects

Package data science code in a format that enables reproducible runs on any platform

mlflow Models

Deploy machine learning models in diverse serving environments environments

new

mlflow Model Registry

Store, annotate and manage models in a central repository

[databricks.com
/mlflow](https://databricks.com/mlflow)



mlflow.org



github.com/mlflow



twitter.com/MLflow

MLflow Model Motivations



ML Frameworks

N x M
Combination of
Model support for
all Serving tools



Inference Code

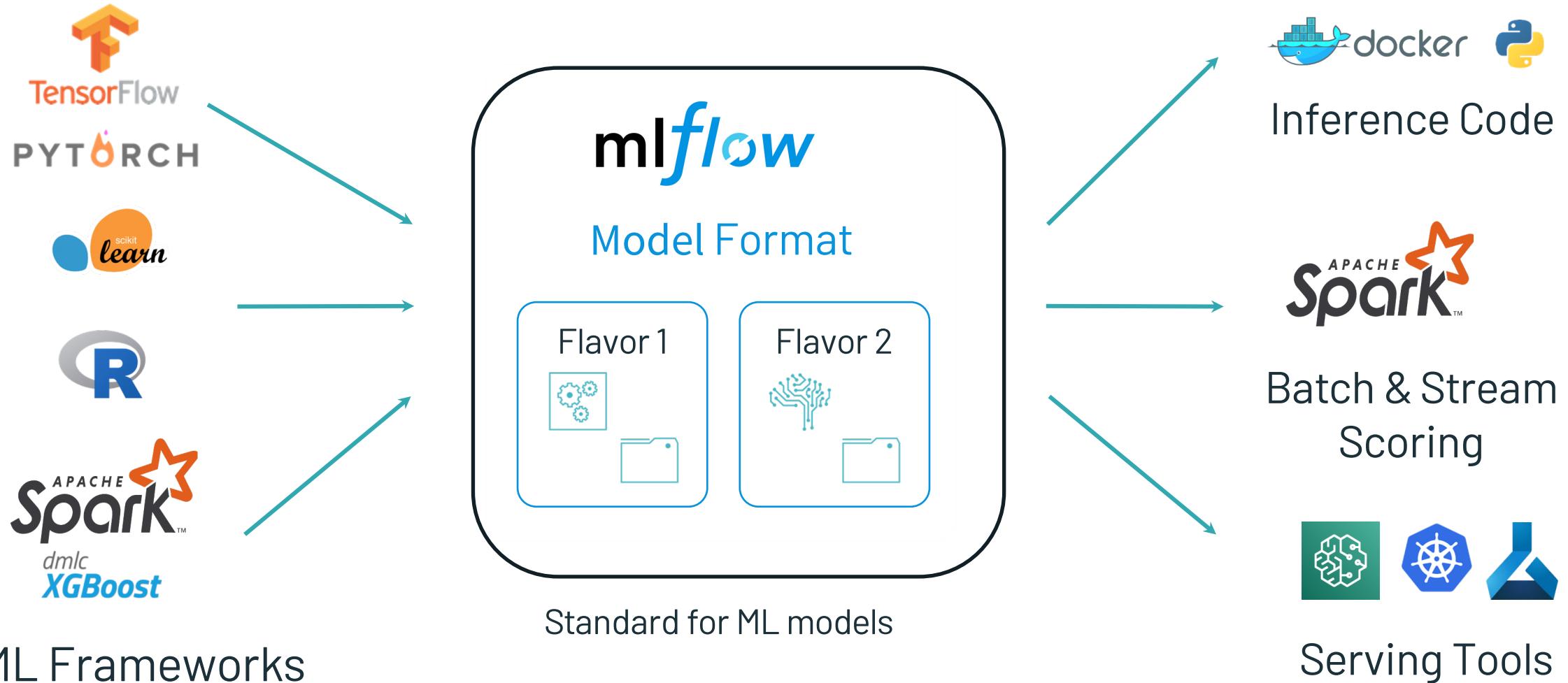


Batch & Stream Scoring

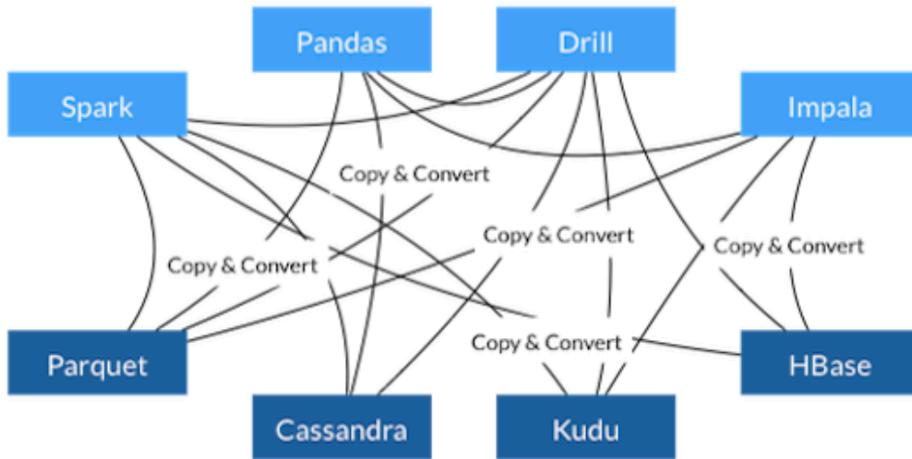


Serving Tools

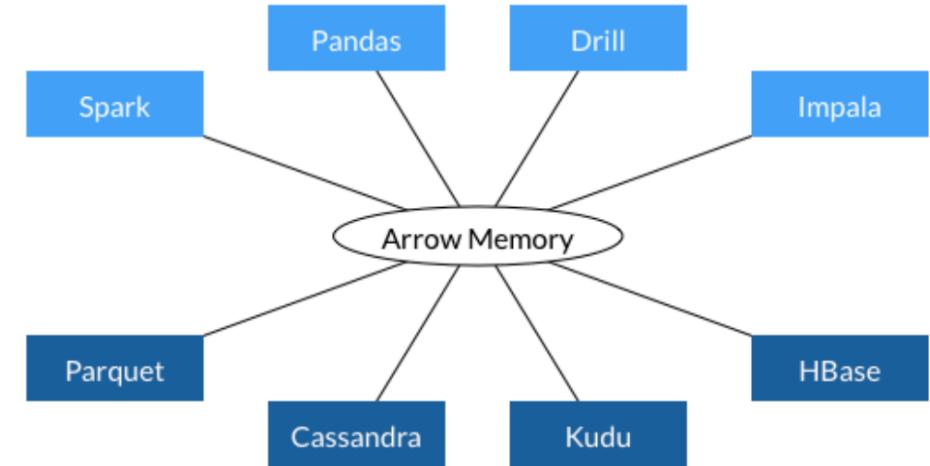
MLflow Models



Advantages of a Common Data Layer



- Each system has its own internal memory format
- 70-80% computation wasted on serialization and deserialization
- Similar functionality implemented in multiple projects



- All systems utilize the same memory format
- No overhead for cross-system communication
- Projects can share functionality (eg, Parquet-to-Arrow reader)

Example MLflow Model

```
mlflow.tensorflow.log_model(...)
```

```
my_model/  
└── MLmodel
```

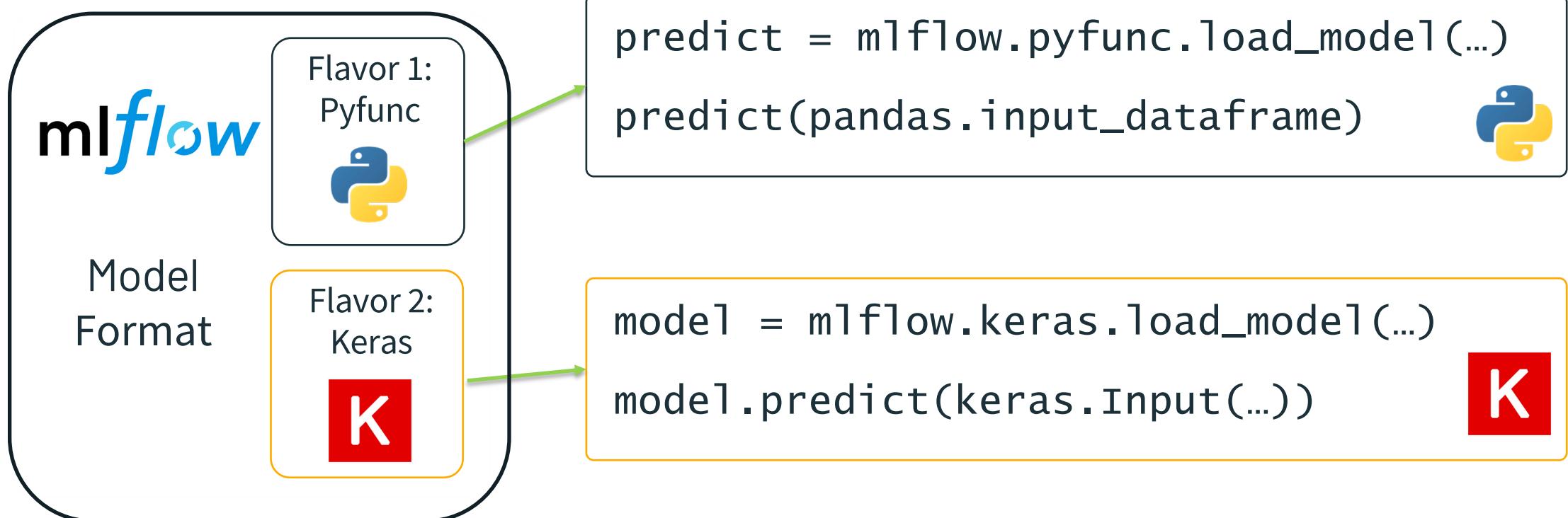
```
run_id: 769915006efd4c4bbd662461  
time_created: 2018-06-28T12:34  
flavors:  
    tensorflow:  
        saved_model_dir: estimator  
        signature_def_key: predict  
    python_function:  
        loader_module: mlflow.tensorflow
```

```
estimator/  
└── saved_model.pb  
└── variables/  
...  
...
```

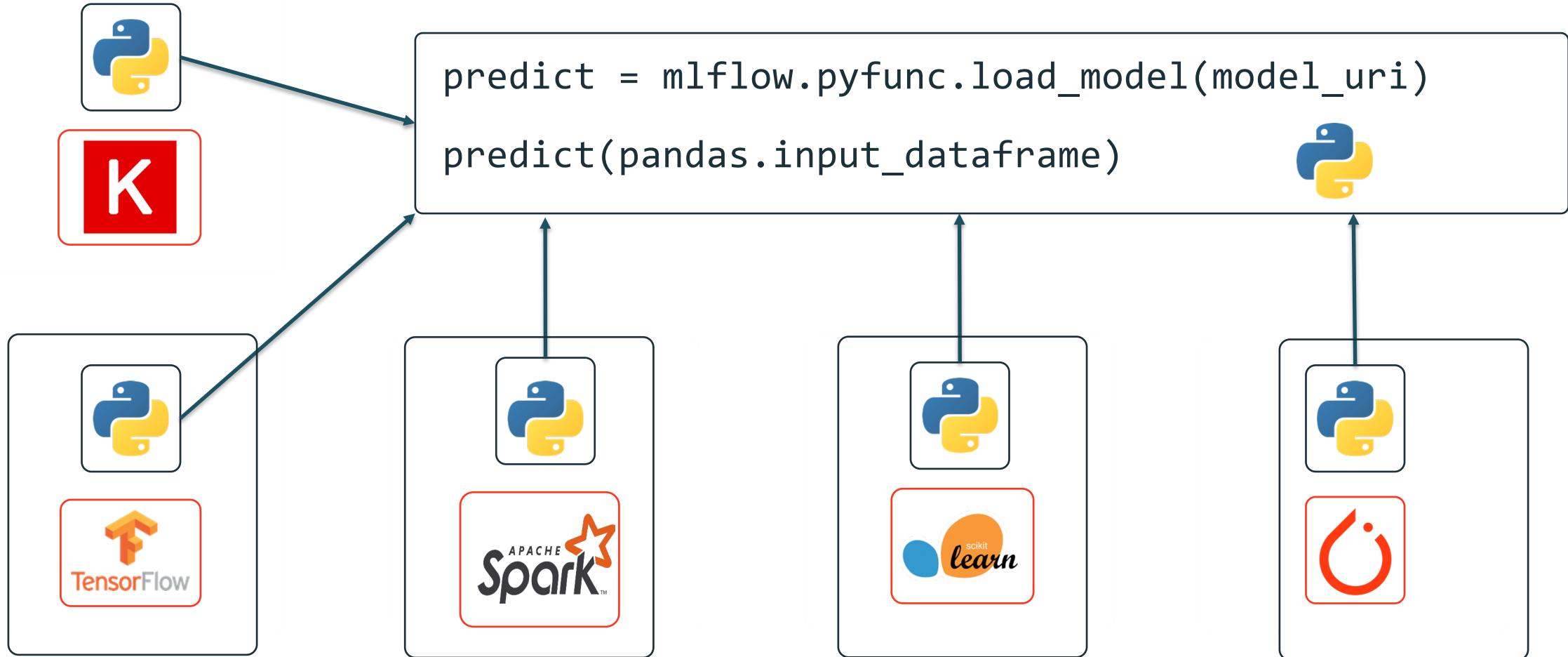
} Usable by tools that understand TensorFlow model format
}

} Usable by any tool that can run Python (Docker, Spark, etc!)

Model Flavors Example



Model Flavors Example



MLflow Project & Models Tutorials

Tutorials: <https://github.com/dmatrix/mlflow-workshop-part-2>

MLflow Project Keras Example:

<https://github.com/dmatrix/mlflow-workshop-project-expamle-1>

Learning More About MLflow

- pip install mlflow to get started
- Find docs & examples at mlflow.org
- Peruse code at [MLflow Github](https://github.com/mlflow/mlflow)
- Join the [Slack channel](#)
- More [MLflow tutorials](#)

Thank you! 😊

Q & A

jules@databricks.com
@2twitme

<https://www.linkedin.com/in/dmatrix/>