Introduction to MLFLOW

Accelerating the Machine Learning Lifecycle

plain concepts

LONDON

Who am I?

Fernando Ortega Gallego

Data Enginer @ Plain Concepts UK

- Interests & hobbies:
  - Machine Learning and NLP
  - Python, Azure and my daughter
  - Honda biker

@FernanOrtega

fgallego@plainconcepts.com

plain concepts

Roadmap

Introduction

MLFlow Tracking

MLFlow Projects

MLFlow Models

Conclusions

Roadmap

Introduction

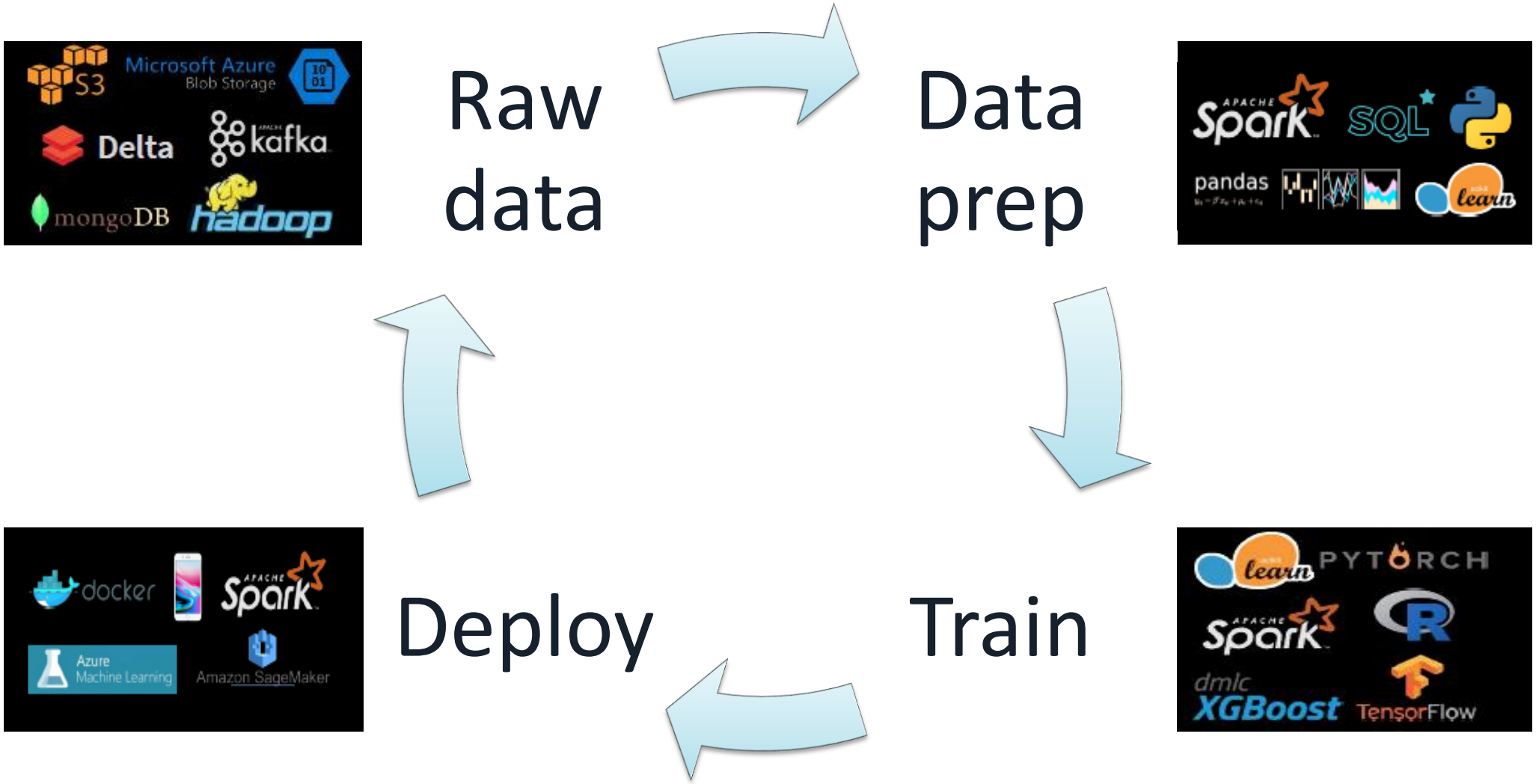
MLFlow Tracking

MLFlow Projects

MLFlow Models

Conclusions

Machine Learning Lifecycle



Problems

- Reproduce experiments
- Compare experiments
- Fine tune previous experiments across teams
- Share data, parameters or metrics
- Deploy trained models

It is difficult to productionize and share

Custom ML Platforms

- Facebook FBLearner, Google TFX, Uber Michelangelo
- Advantages:
  - Standarise the ML loop
- Disadvantages:
  - Limited to a few algorithms or frameworks
  - Tied to the company's infrastructure

Are there any similar solutions in an open manner?

mlflow

Introducing MLFlow

- It works with any ML library & language
- It runs the same way anywhere
- It is designed to be useful both for 1 person, small teams or big teams

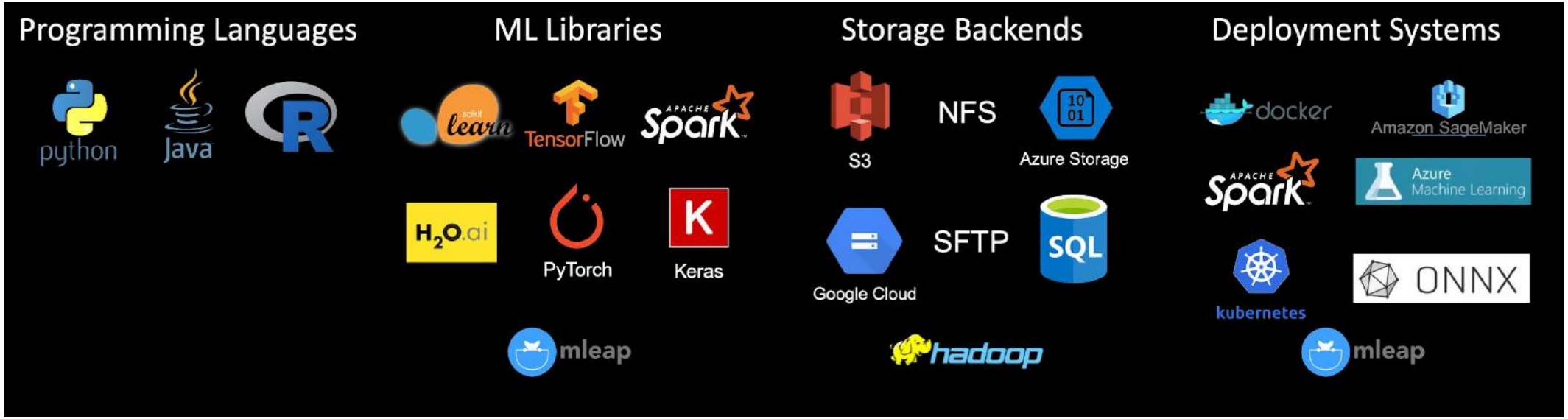
MLFlow community

600k
monthly downloads

100+
code contributors

40
contributing organizations

Supported Integrations: June’19



MLFlow components

mlflow™ Tracking

mlflow™ Projects

mlflow™ Models

Roadmap

Introduction

MLFlow Tracking

MLFlow Projects

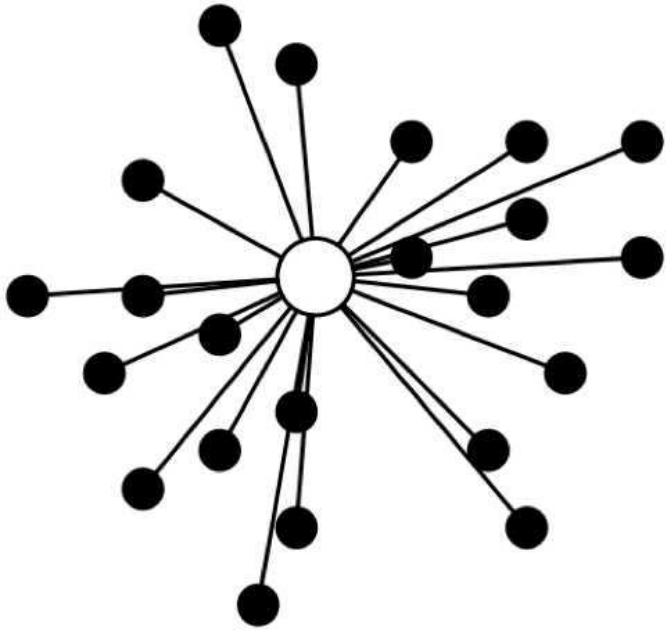
MLFlow Models

Conclusions

MLFlow Tracking

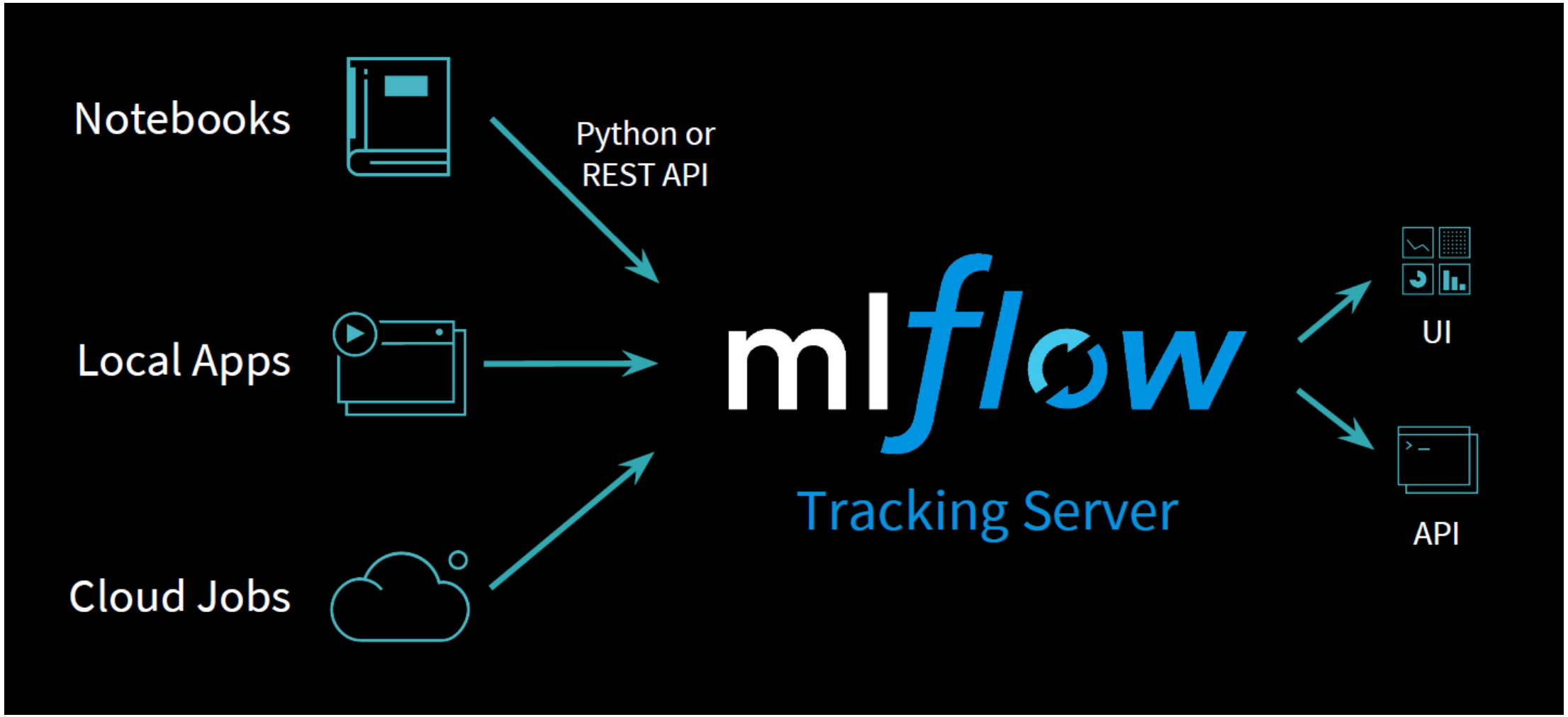
- Record and query experiments:
  - Data
  - Code
  - Model parameters
  - Results (performance metrics)
  - Model

Motivation



Centralised repository of useful information to analyse several runs of training.

How it is working



Key concepts

- Parameters
- Metrics
- Tags and notes
- Artifacts
- Source
- Version

Code example

```python
import mlflow

with mlflow.start_run():
    mlflow.log_param("layers", layers)
    mlflow.log_param("alpha", alpha)

    # train model
    model = train_model(layers, alpha)

    mlflow.log_metric("mse", model.mse())
    mlflow.log_artifact("plot", model.plot(test_df))
    mlflow.tensorflow.log_model(model
```

Demo

Roadmap

Introduction

MLFlow Tracking

MLFlow Projects

MLFlow Models

Conclusions

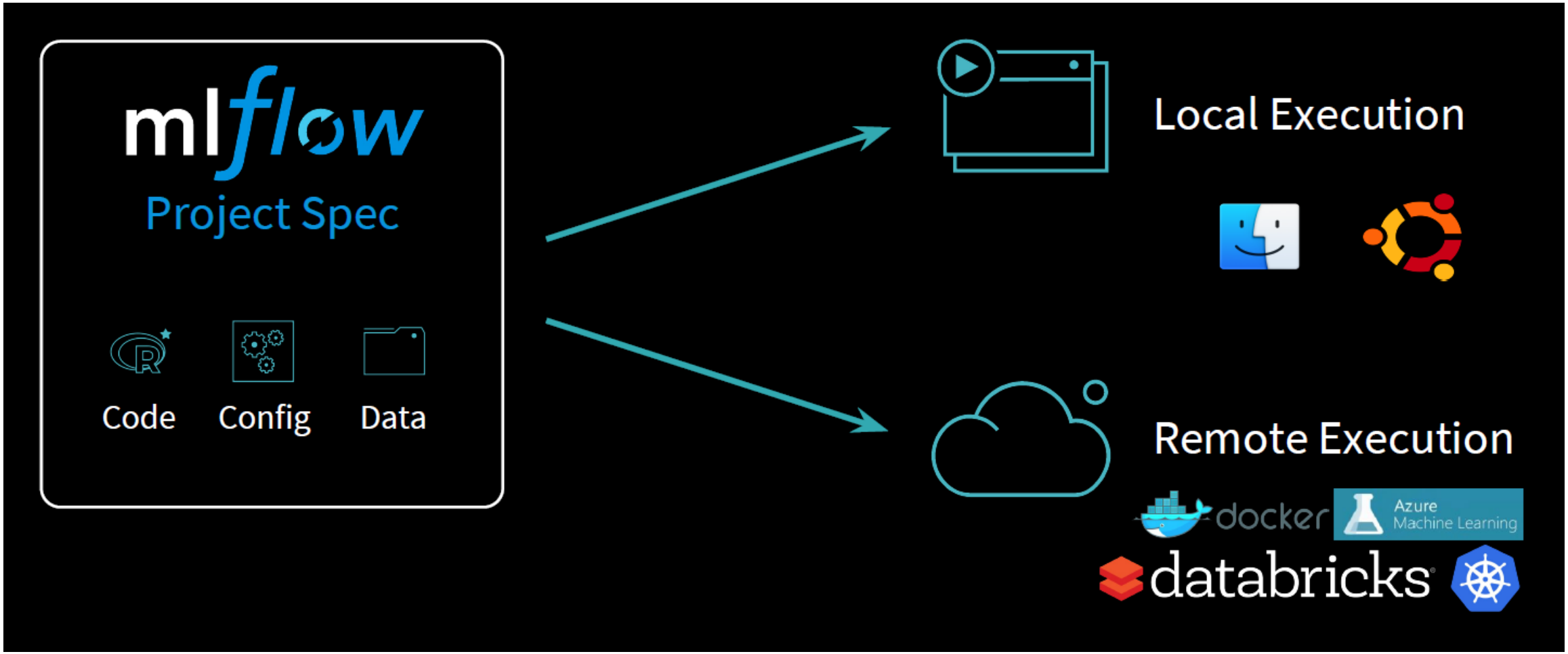
MLFlow Projects

- Packaging format for reproducible ML runs
- Defines dependencies for reproducibility
- Execution API for running projects locally or remote

Motivation

Diverse set of tools and environments involves difficulty to productionalize and share ML work

How it is working



Key concepts

- MLproject file
- Entry points
- Environments
  - Conda
  - Docker
  - System
- Run

Code example

```
my_project/
  |- MLproject
  |
  |
  |
  |
  |
  |- conda.yaml
  |- main.py
  |- model.py
  ...
```

```yaml
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@github.com:mlflow/mlflow-example.git -P alpha=0.5

mlflow run <uri> -m databricks --cluster-spec <json-cluster-spec>
```

Demo

Roadmap

Introduction

MLFlow Tracking

MLFlow Projects

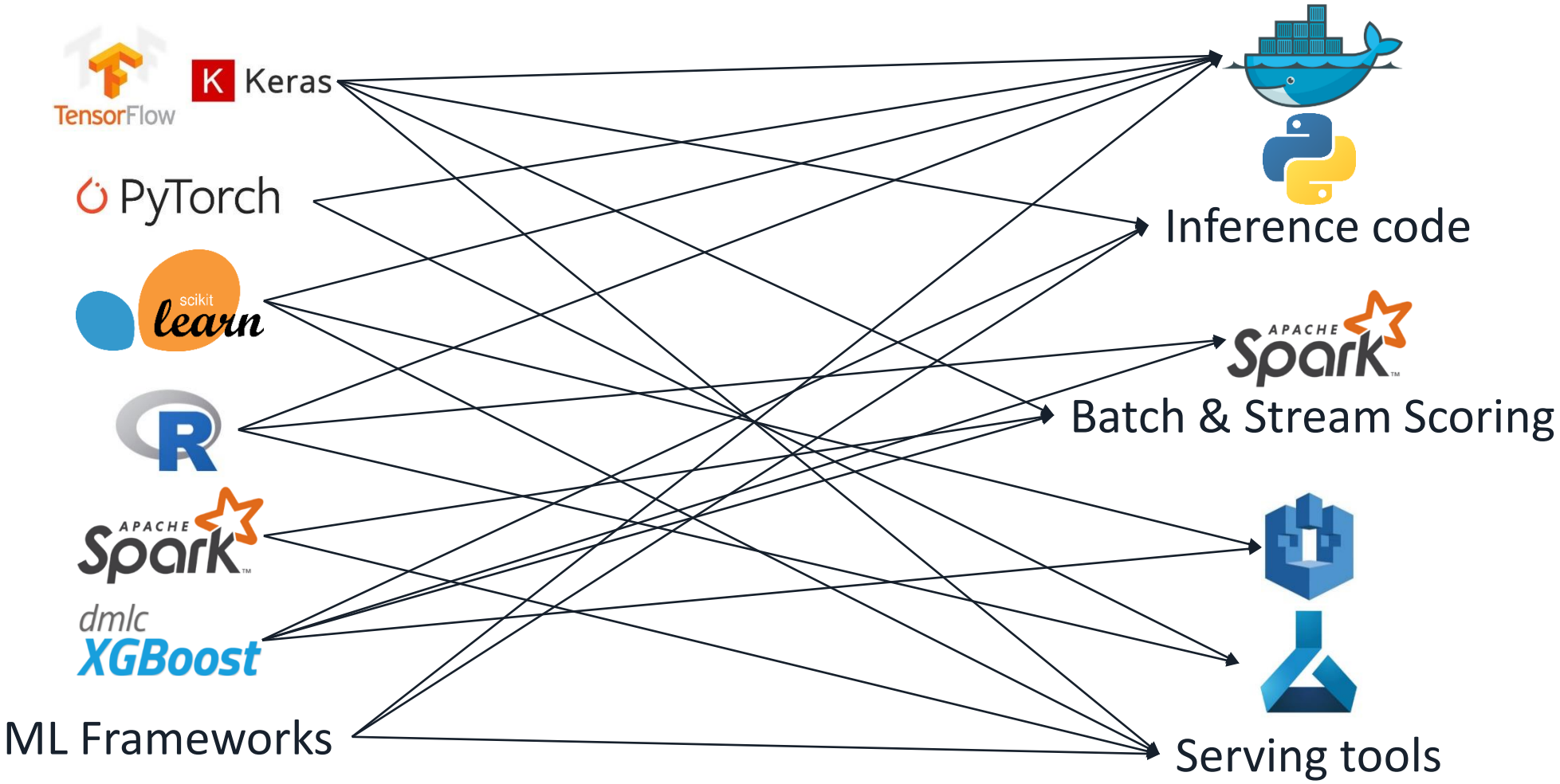
MLFlow Models

Conclusions

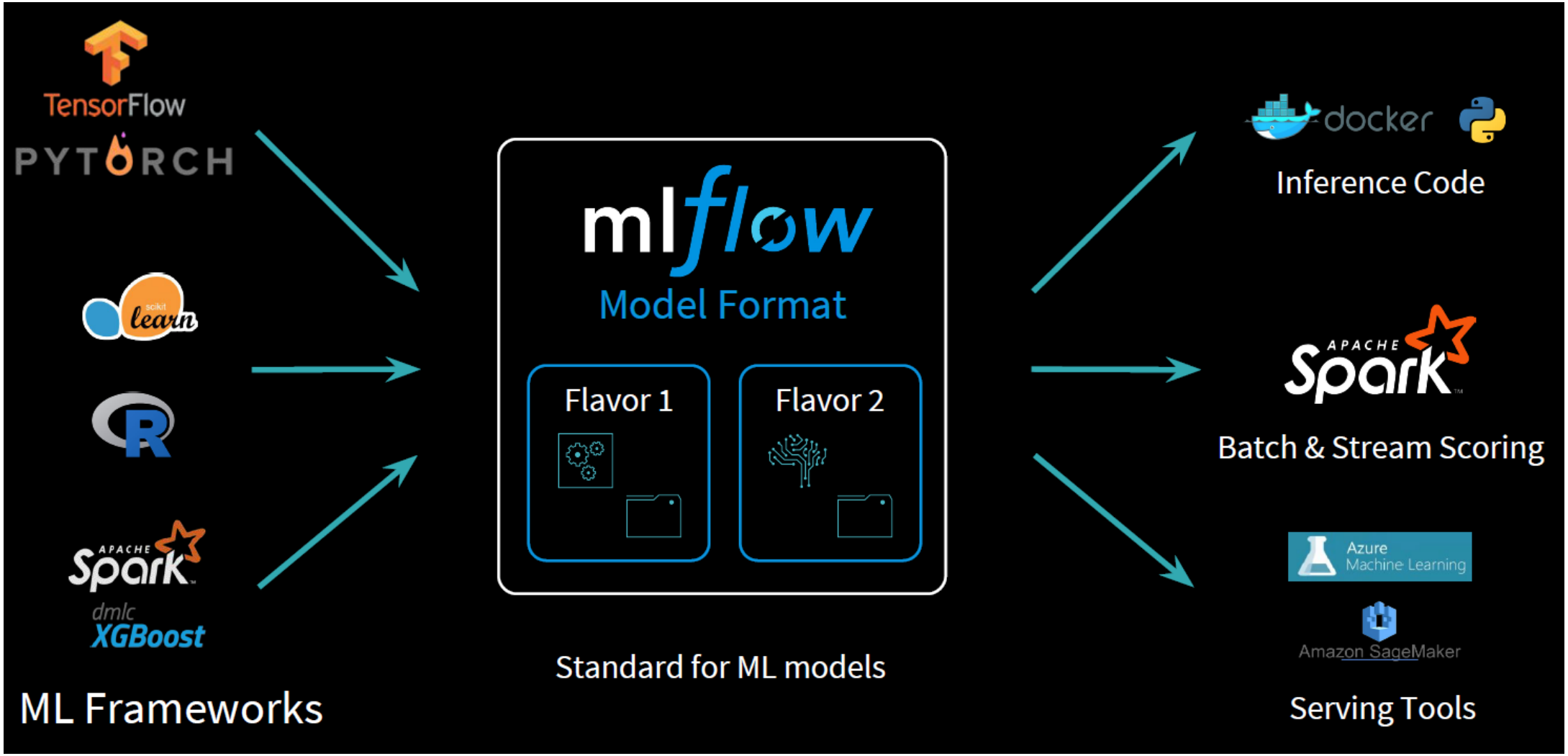
MLFlow Models

- Packaging format for ML Models
- Defines dependencies for reproducibility
- Deployment APIs

Motivation


TensorFlow
Keras
PyTorch
scikit learn
R
APACHE Spark
dmlc XGBoost
ML Frameworks
Inference code
APACHE Spark
Batch & Stream Scoring
Serving tools

How it is working



Key concepts

- MLmodel file
- Storage format
- Entry points
- Flavours
- Custom model

Code example

```
my_model/
├── MLmodel
│
│
│
│
│
└── estimator/
    ├── saved_model.pb
    └── variables/
        ...
```

```
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
```

Usable by tools that understand TensorFlow model format

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

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

Demo

Roadmap

Introduction

MLFlow Tracking

MLFlow Projects

MLFlow Models

Conclusions

plain concepts

LONDON

MLFlow rocks!

- **Log** important parameters, metrics, and other data that is important to the machine learning model
- **Track** the environment a model is run on
- **Run any** machine learning codes on that environment
- **Deploy and export** models to various platforms with multiple packaging formats

MLFlow 1.0 (4-jun)

- Support for step tracking
- Improved Search features
- Batched logging of metrics
- Support for HDFS
- Windows support for the client
- Build Docker images to deploy
- ONNX model flavour

MLFlow last updates (1.4: 31-oct)

- Windows support
- Tags and descriptions
- Google Cloud run models
- Log directories as artifacts
- CLI command to export to CSV
- Keras compatibility with TF 2.0
- Model Registry in preview

Future of MLFlow

PySEVILLA

plain concepts

LONDON