mlflow Platform for Complete Machine Learning Lifecycle

Jules S. Damji
@2twitme

San Francisco | May 20, 2020: Part 3 of 3 Series

databricks

Outline – Introduction to MLflow: Model Registry Workflows Explained – Part 3

- Review & Recap Part 2: MLflow Projects & Models
- Concepts and Motivations
- MLFlow Component
  - MLflow Model Registry
  - Model Registry UI & API Workflow
  - Managed MLflow Model Registry Demo
    - Tutorials on local host
- Q & A

https://dbricks.co/mlflow-part-3

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

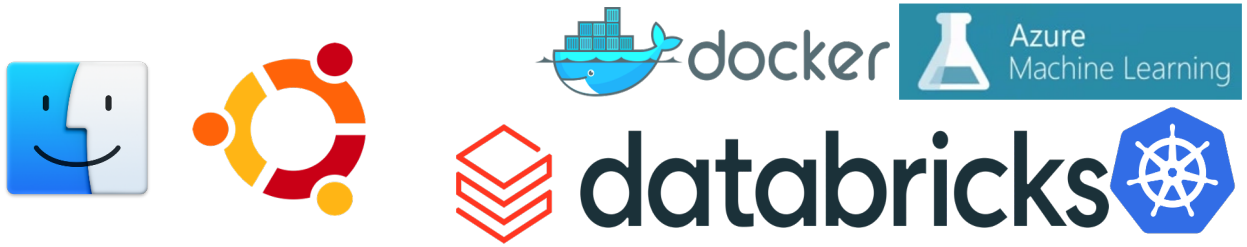
twitter.com/MLflow

databricks

MLflow Projects Motivation

Diverse set of tools

Diverse set of environments



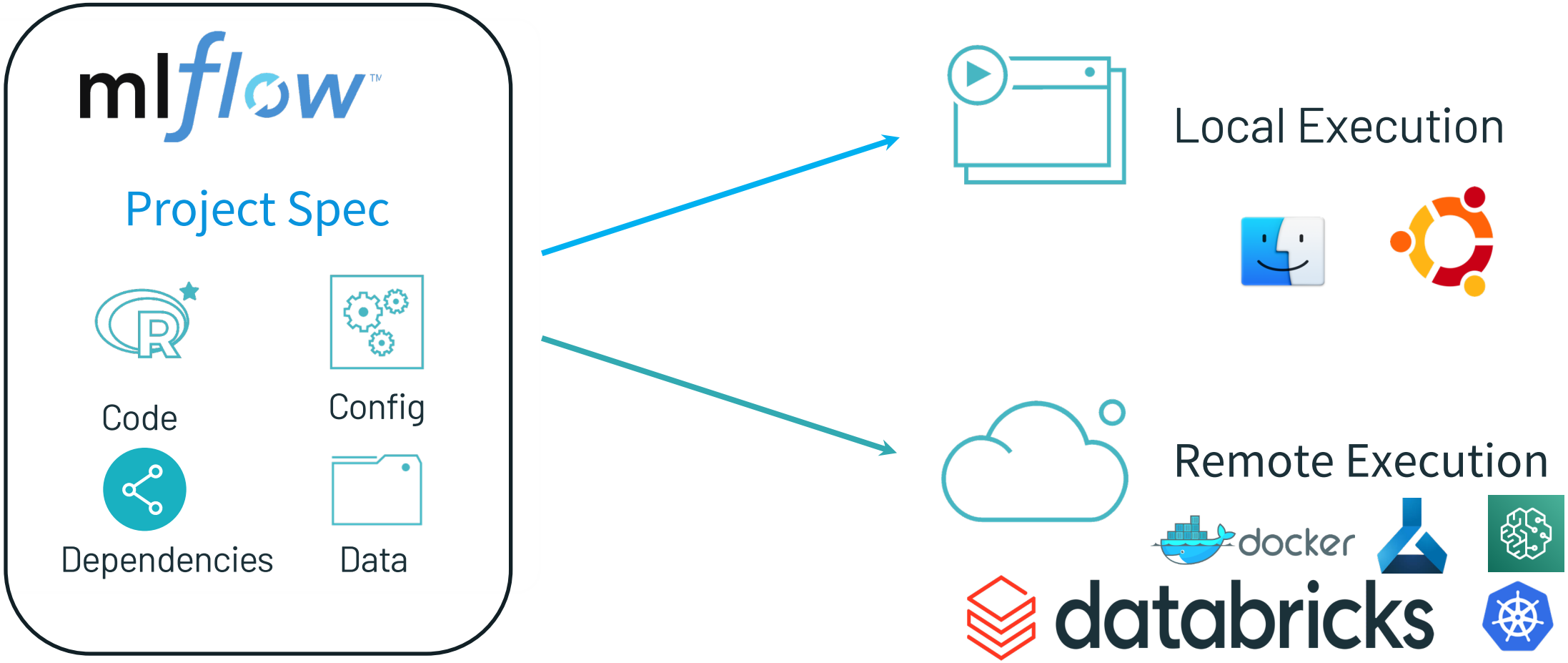
mlflow

Projects

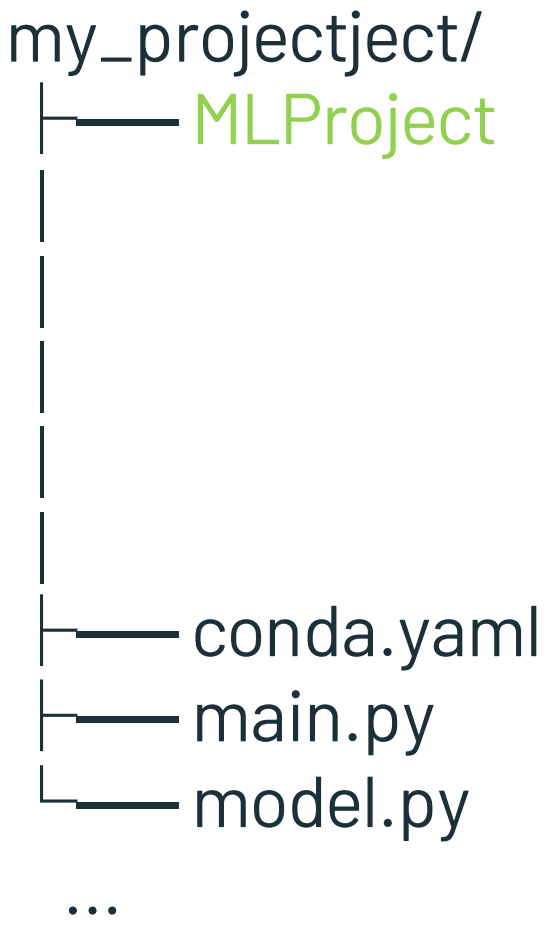
Package data science code in a format that enables reproducible runs on any platform

Challenge: ML results difficult to reproduce

MLflow Projects



Example MLflow Project




```
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 Model Motivations

ML Frameworks

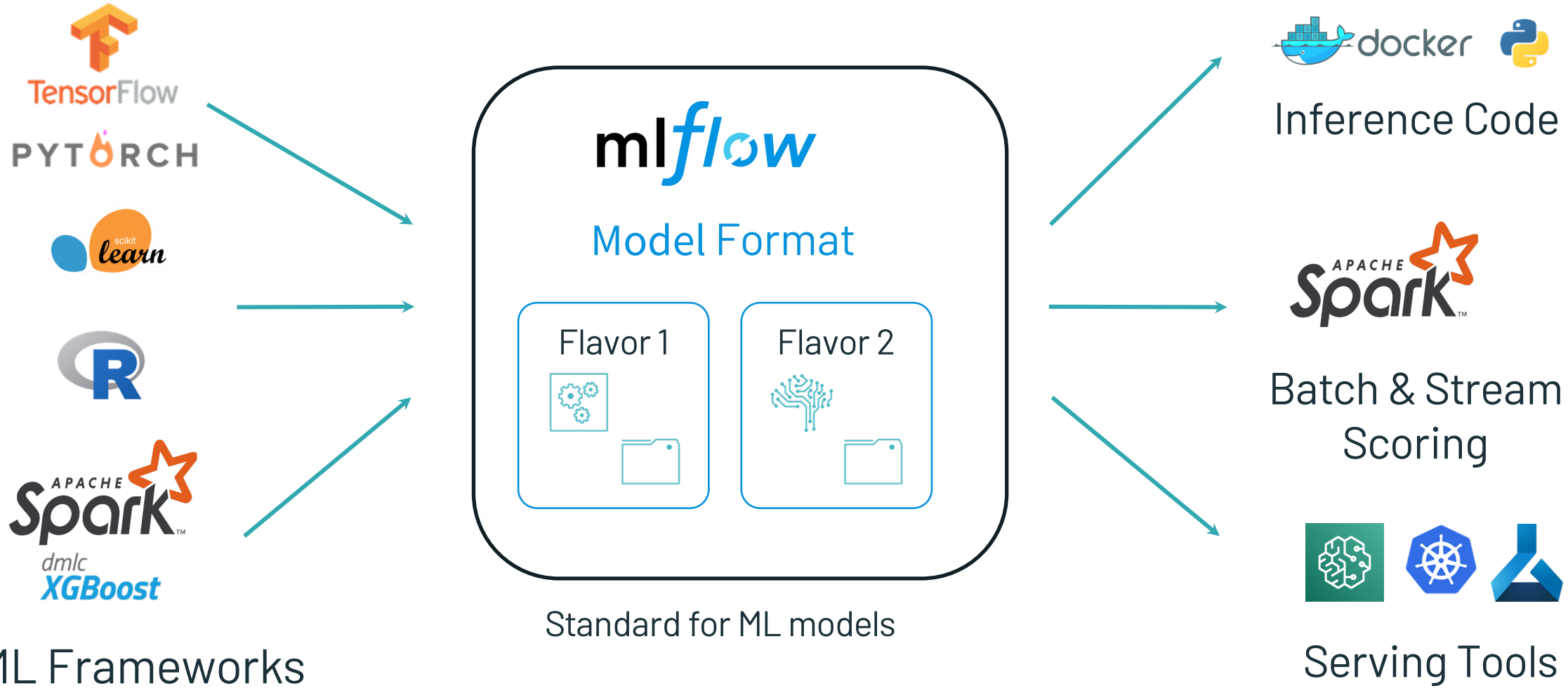
N x M
Combination of
Model support for
all Serving tools

Inference Code

Batch & Stream Scoring

Serving Tools

MLflow Models



Example MLflow Model

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

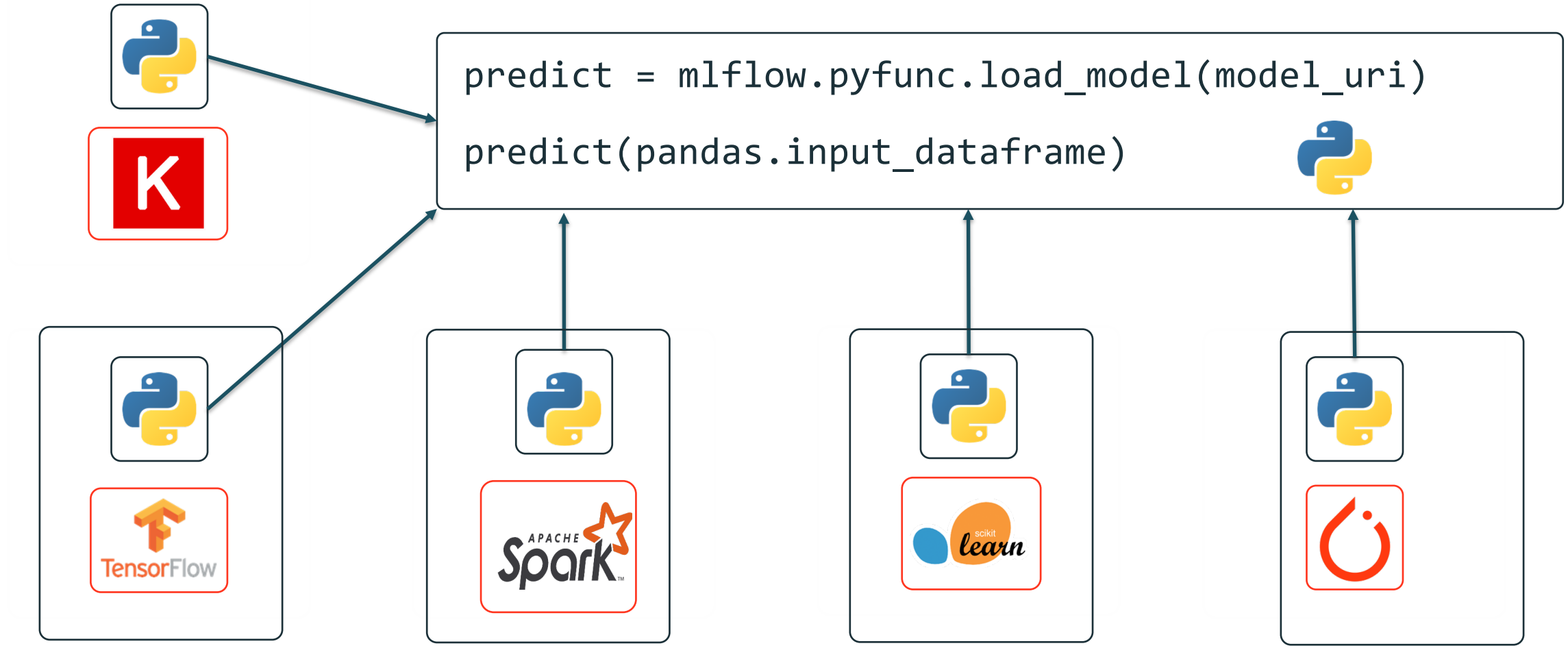
```
my_model/
├── MLmodel
└── estimator/
    ├── saved_model.pb
    └── variables/
        ...
```

```
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!)

Model Flavors Example



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

databricks

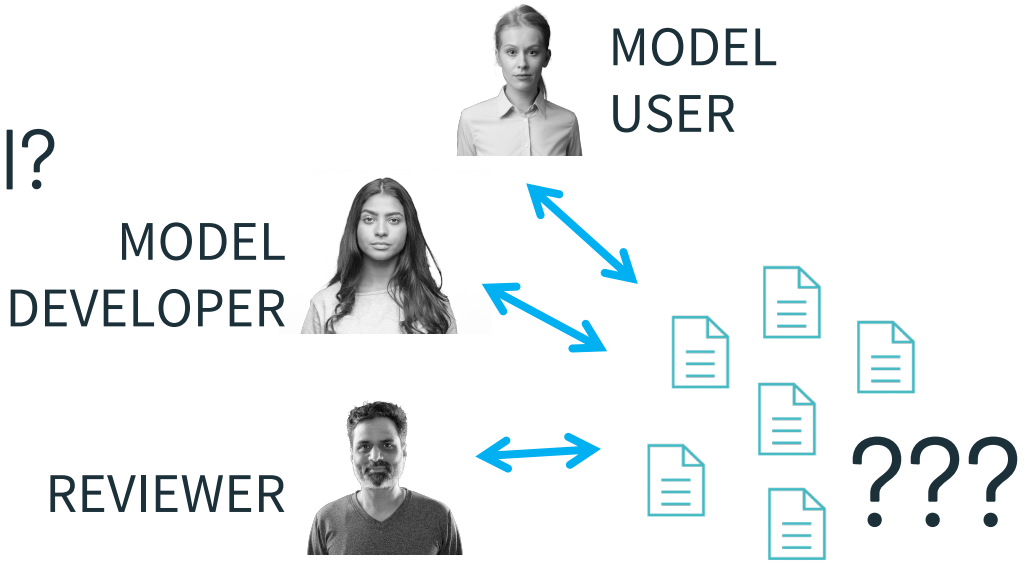
The Model Management Problem

When you're working on one ML app alone, storing your models in files is manageable

The Model Management Problem

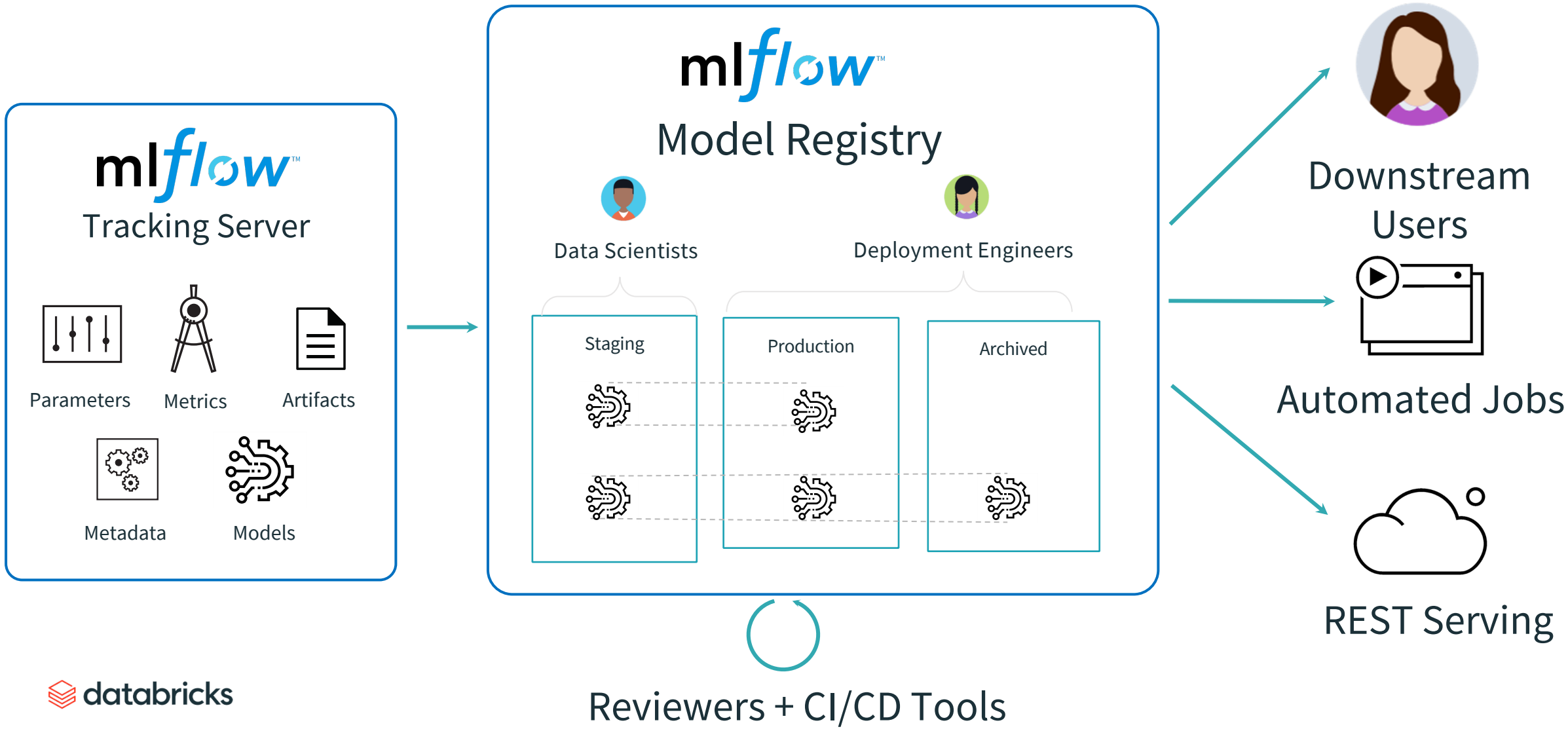
When you work in a large organization with many models, many data teams, management becomes a major challenge:

- Where can I find the best version of this model?
- How was this model trained?
- How can I track docs for each model?
- How can I review models?
- How can I integrate with CI/CD?



mlflow Model Registry

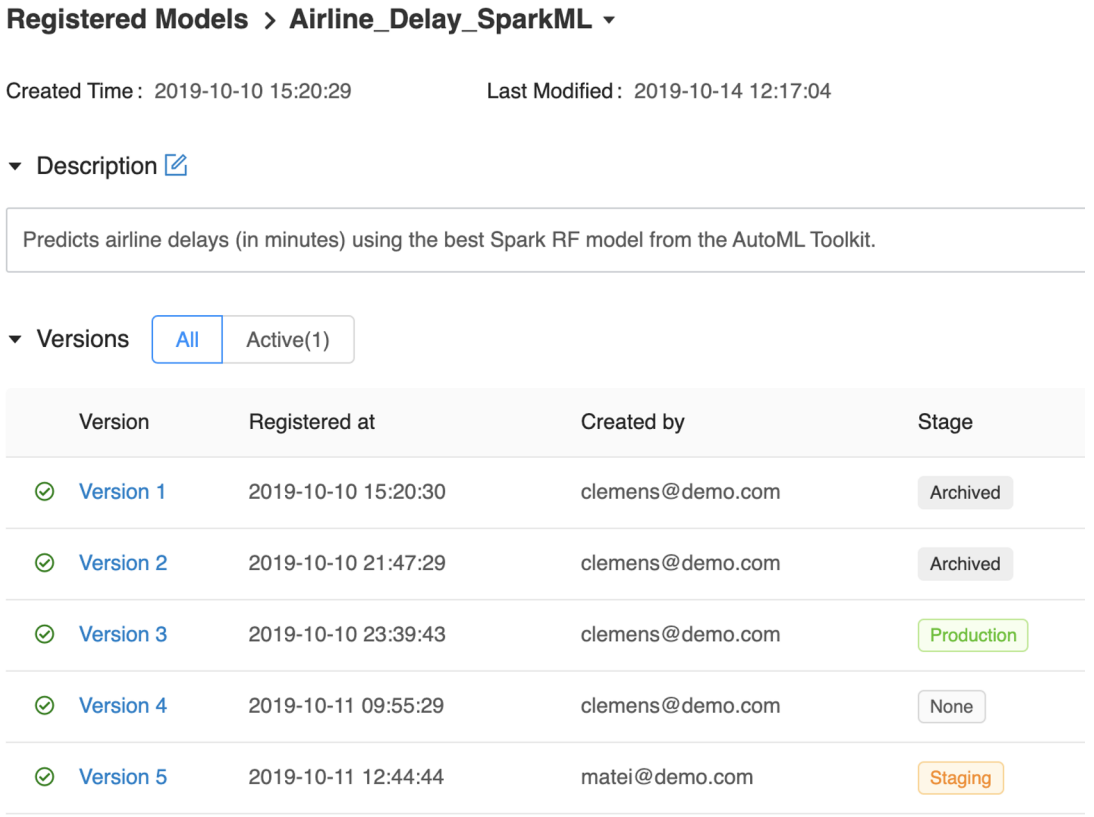
VISION: Centralized and collaborative model lifecycle management



databricks

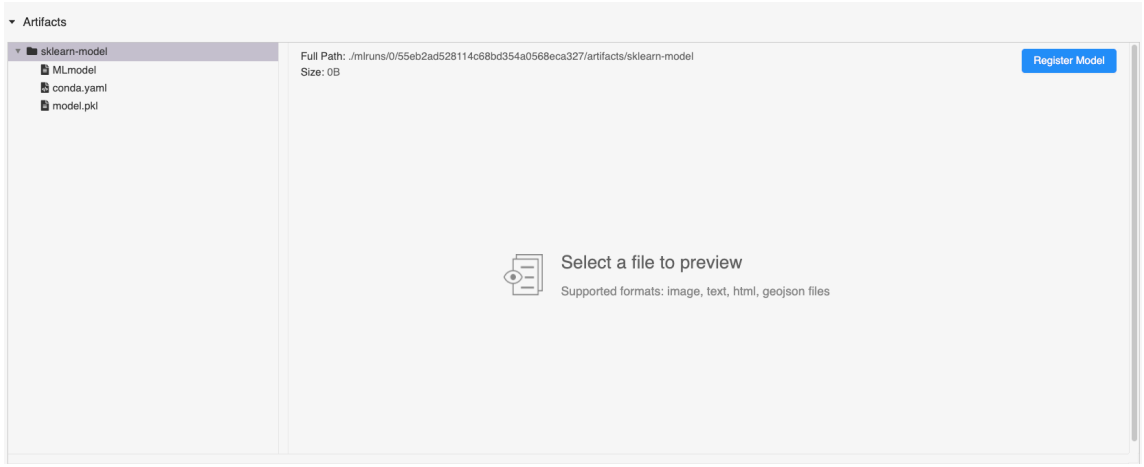
MLflow Model Registry

- Repository of named, versioned models with comments & tags
- Track each model's stage: none, staging, production, or archived
- Easily inspect a specific version and its run info
- Easily load a specific version
- Provides Model description, lineage & activities

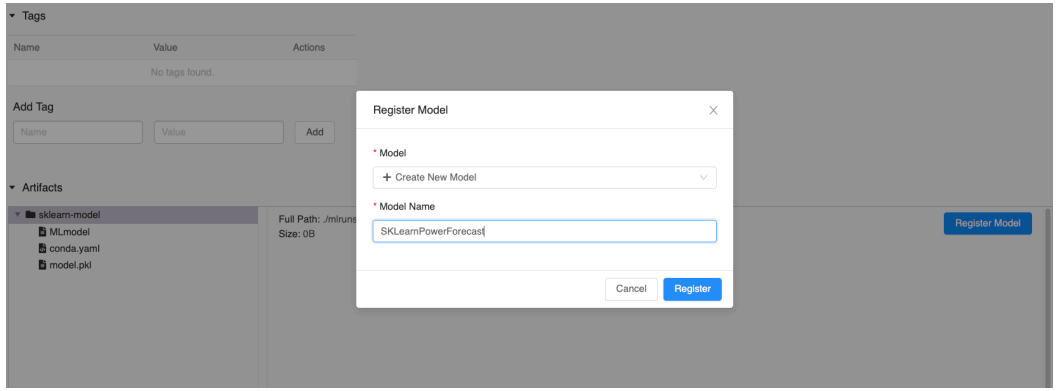


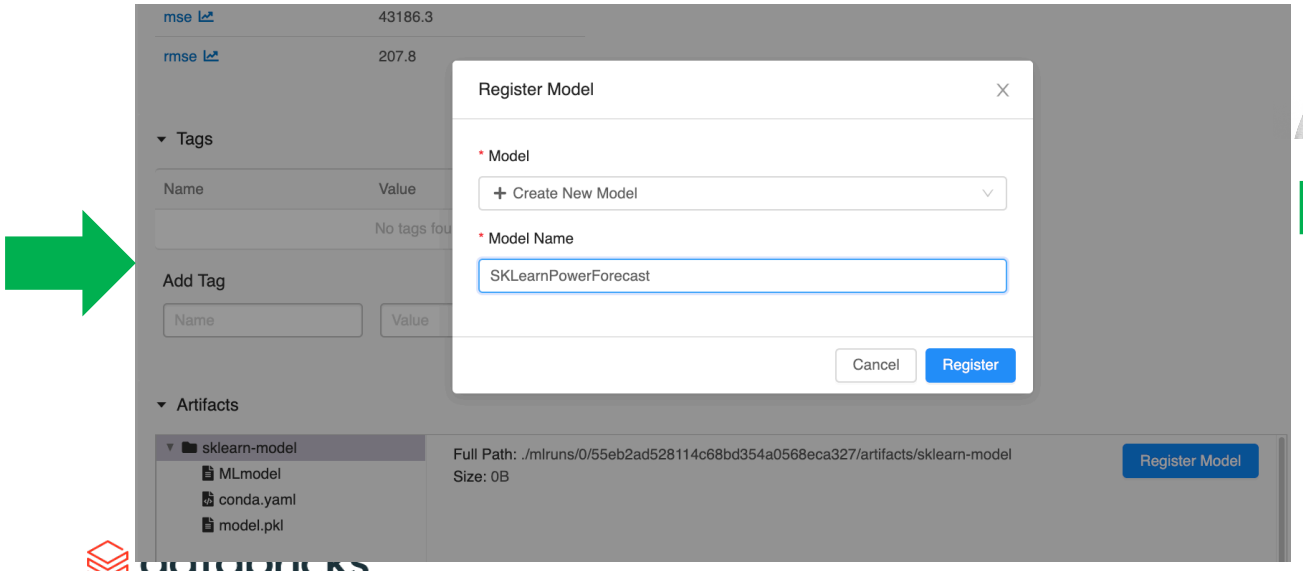
databricks

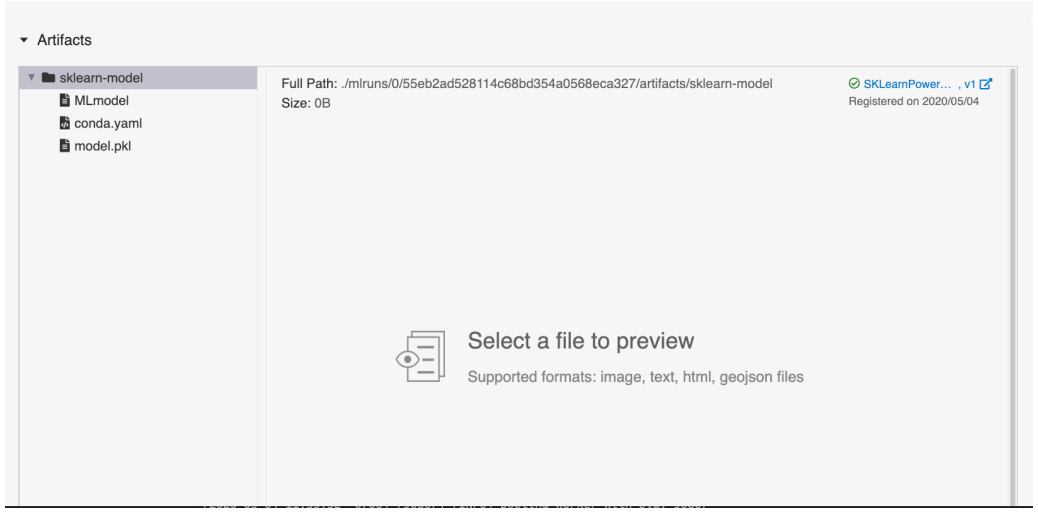
Model Registry Workflow UI



MODEL DEVELOPER

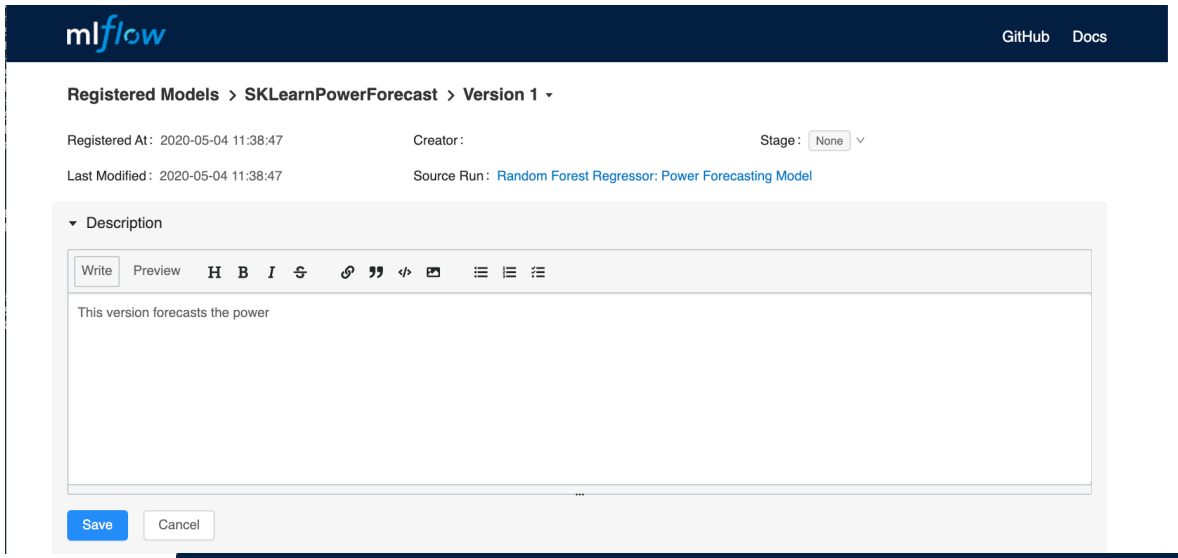




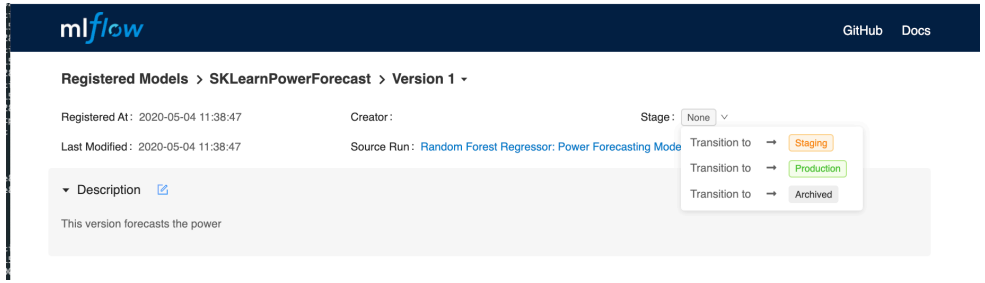


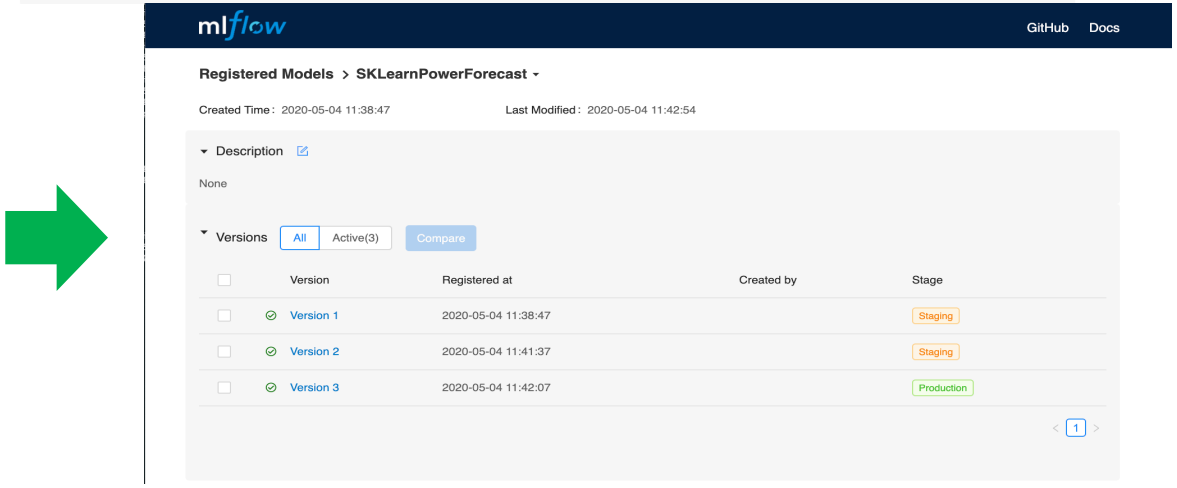
databricks

Model Registry Workflow UI



MODEL REVIEWER





DOWNSTREAM USERS

AUTOMATED JOBS

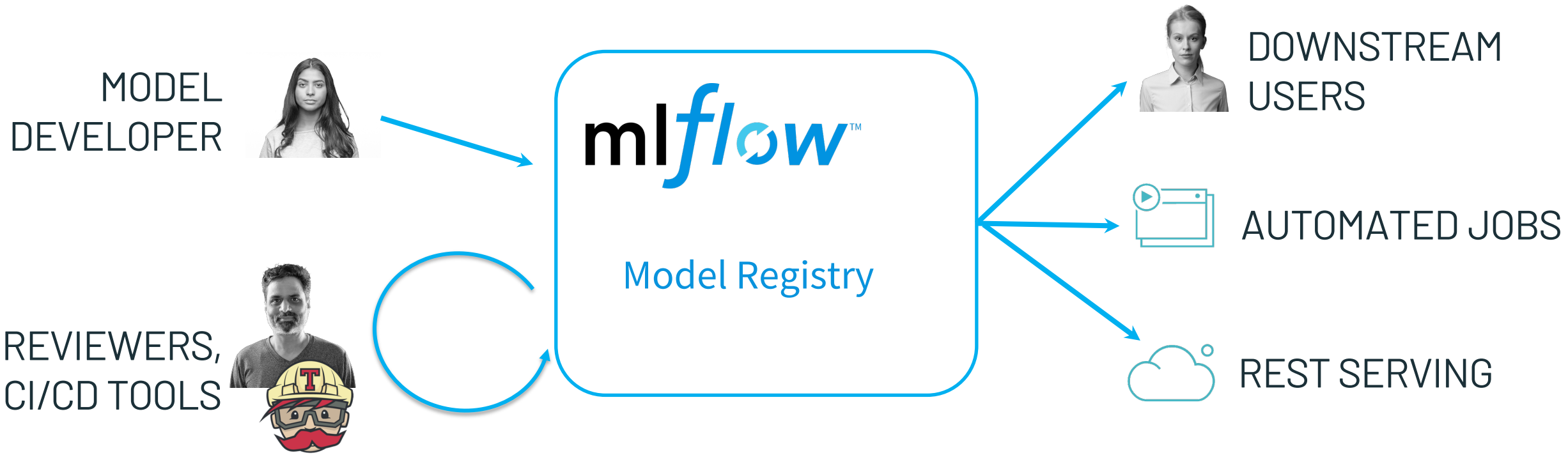
REST SERVING

databricks

Model Registry Workflow API

```
mlflow.register_model(model_uri,"WeatherForecastModel")
```

```
model_uri= "models:/{model_name}/production".format(
          model_name="WeatherForecastModel")
model_prod =  mlflow.sklearn.load_model(model_uri)
```




```
client = mlflow.tracking.Mlflowclient()
client.transition_model_version_stage(name=”WeatherForecastModel”,
                                      version=5,
                                      stage="Production")
```

databricks

MLflow Backend Registry Stores

Entity (Metadata) Store and Models

- SQLStore (via SQLAlchemy)
  - PostgreSQL, MySQL, SQLite
  - Default is mlruns.db file locally
- Set programmatically for locally
- `mlflow.set_tracking_uri("sqlite:///mlruns.db")`
- `sqlite3 ./mlruns.db (on local host)`

Artifact Store

- Local Filesystem
  - mlruns directory
- S3 backed store
- Azure Blob storage
- Google Cloud Storage
- DBFS artifact repo
- Managed MLflow on Databricks
  - MySQL on AWS and Azure

```
(tutorials) ➜  src git:(master) ✗ sqlite3 ./mlruns.db
SQLite version 3.30.1 2019-10-10 20:19:45
Enter ".help" for usage hints.
sqlite> .databases
main: /Users/julesdamji/gits/mlflow-workshop-part-3/src/./mlruns.db
sqlite> .tables
alembic_version    latest_metrics     params             tags
experiment_tags    metrics            registered_models
experiments        model_versions     runs
sqlite>
```

What Did We Talk About? mlflow™

- Modular Components greatly simplify the ML lifecycle
- Easy to install and use
- Great Developer Experience
- Develop & Deploy locally and track locally or remotely
- Available APIs: Python, Java & R (Soon Scala)
- Visualize experiments and compare runs
- Centrally register and mange model lifecycle

MLflow Model Registry Demo

Tutorials: https://github.com/dmatrix/mlflow-workshop-part-3

Thank you! ☺

Q & A

jules@databricks.com
@2twitme
https://www.linkedin.com/in/dmatrix/