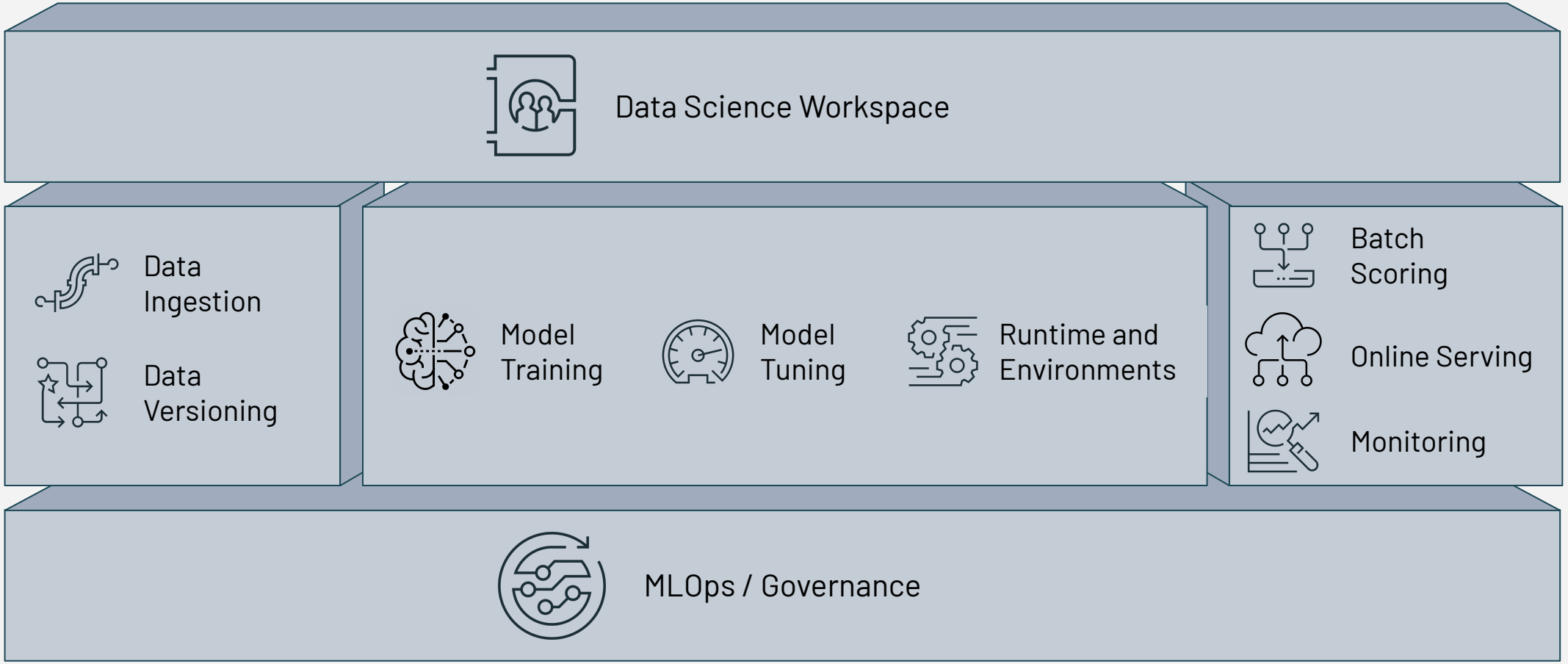
Databricks Overview for MLOps

Clemens Mewald
Director of Product Management

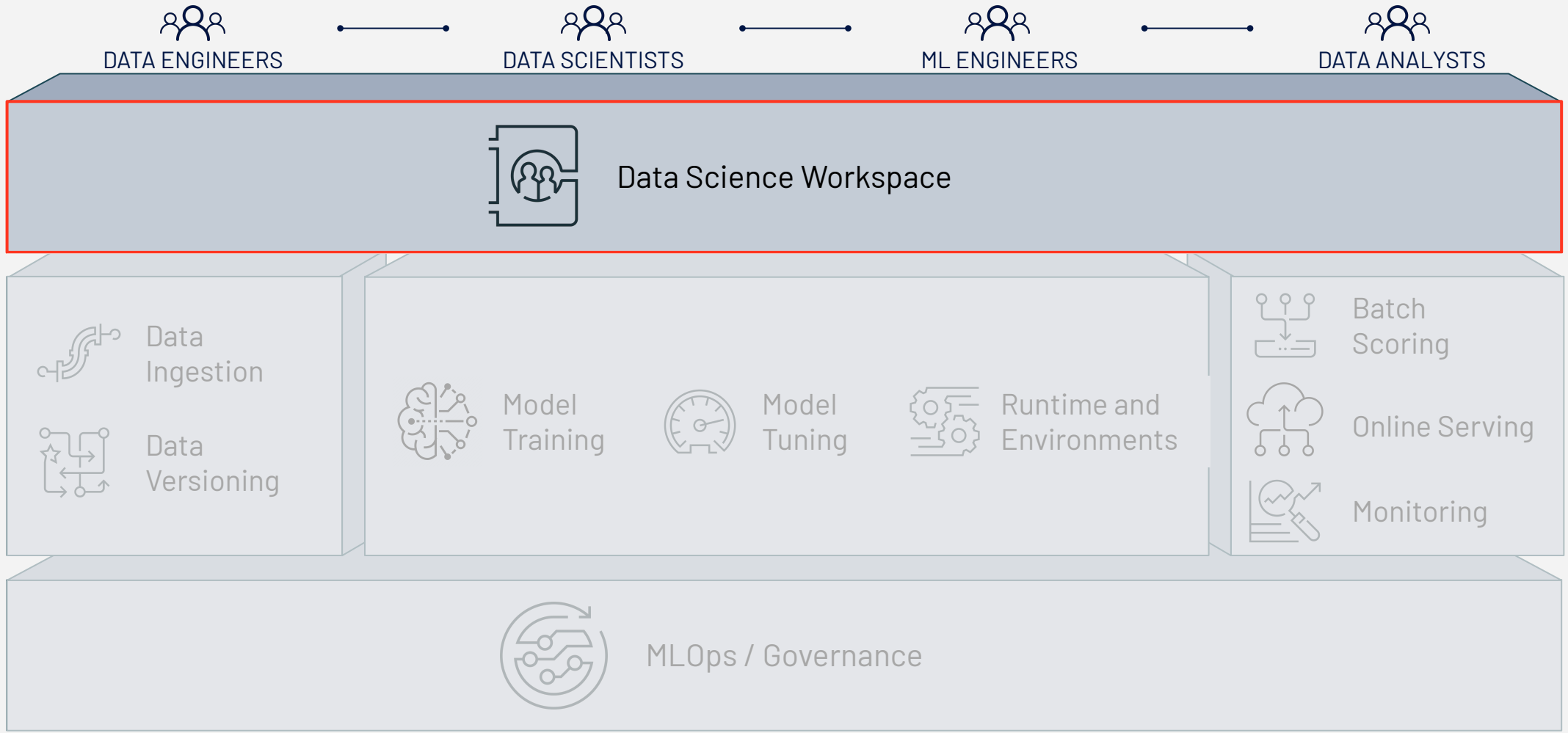
databricks

The Databricks ML Platform



databricks

Collaborative Data Science Workspace



databricks

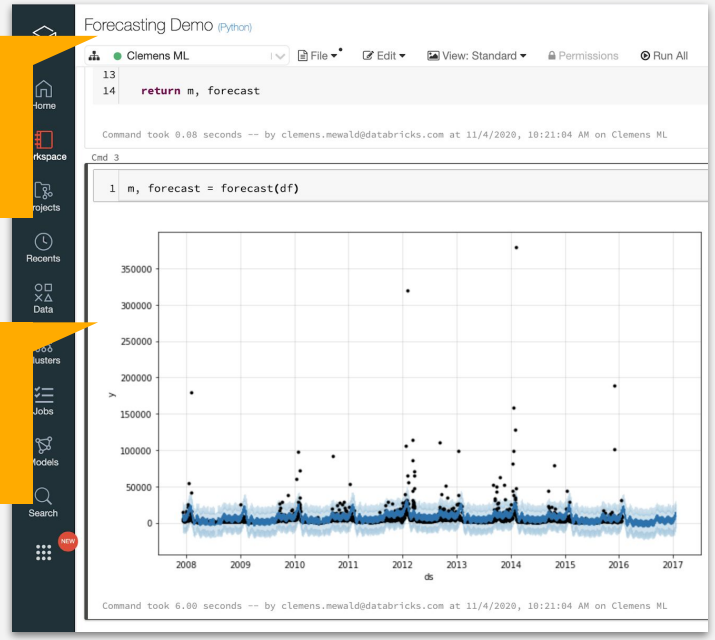
Databricks Notebooks

DATA ENGINEERS — DATA SCIENTISTS — ML ENGINEERS — DATA ANALYSTS

Data Science Workspace

Multi-Language
Scala, SQL, Python, R: All in one notebook.

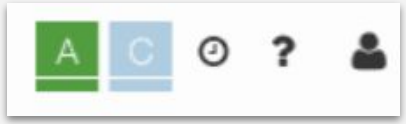
Collaborative
Realtime co-presence, co-editing, and commenting.



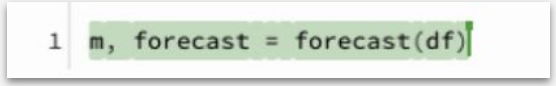
Cloud-native Collaboration Features

Commenting

Co-Presence

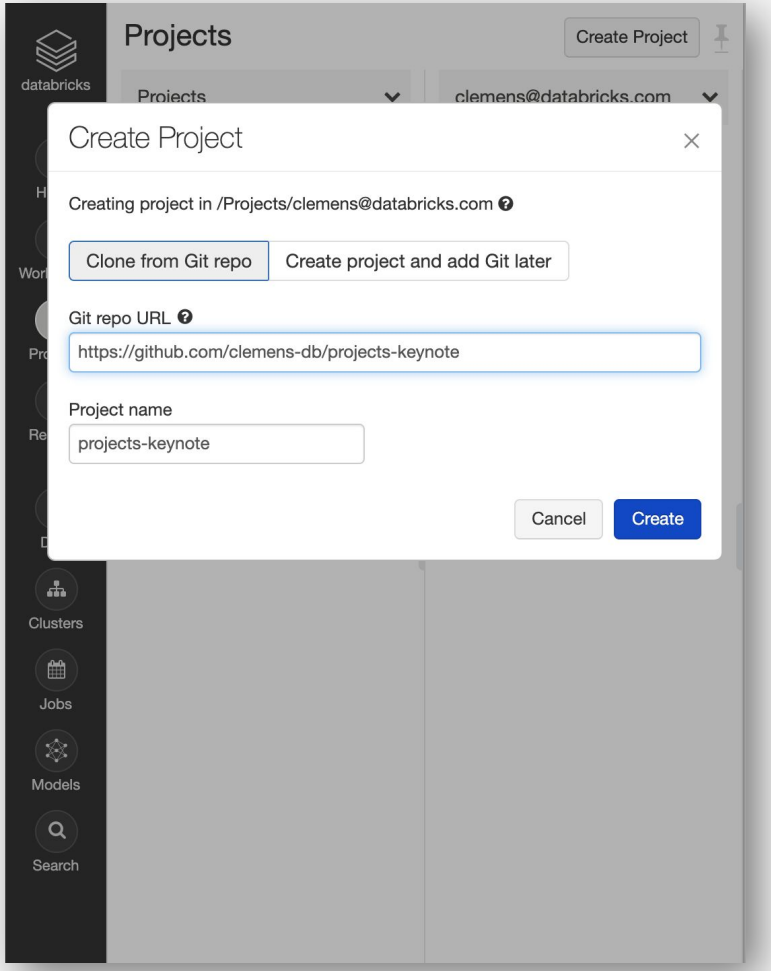


Co-Editing

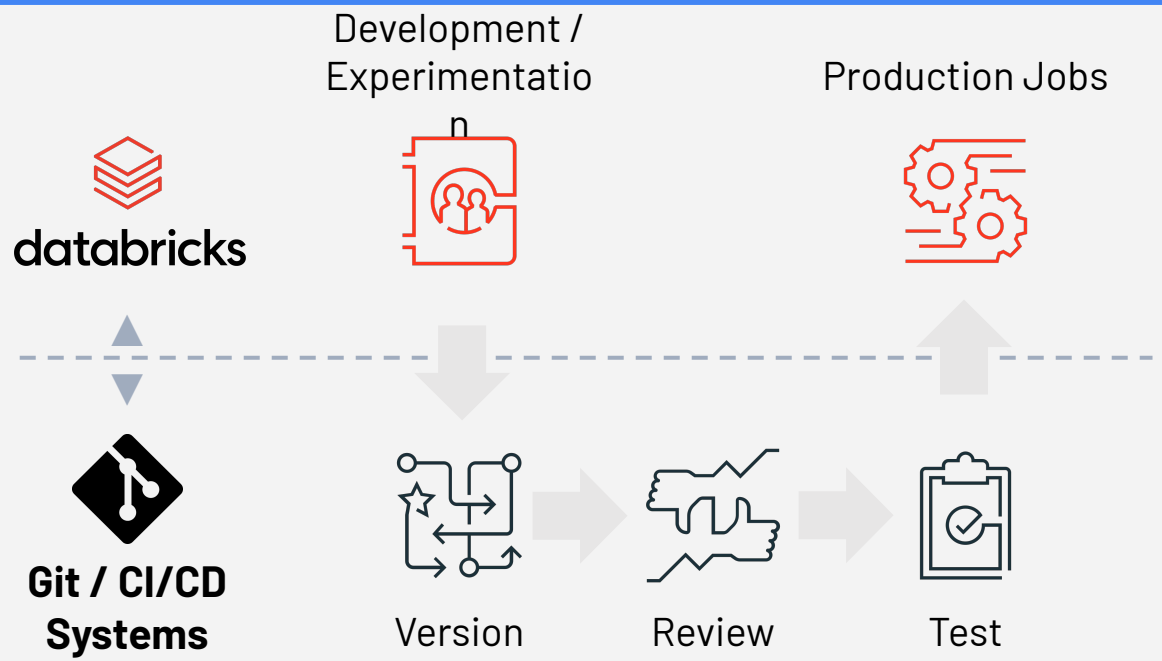


databricks

(Git-based) Projects



CI/CD Integration

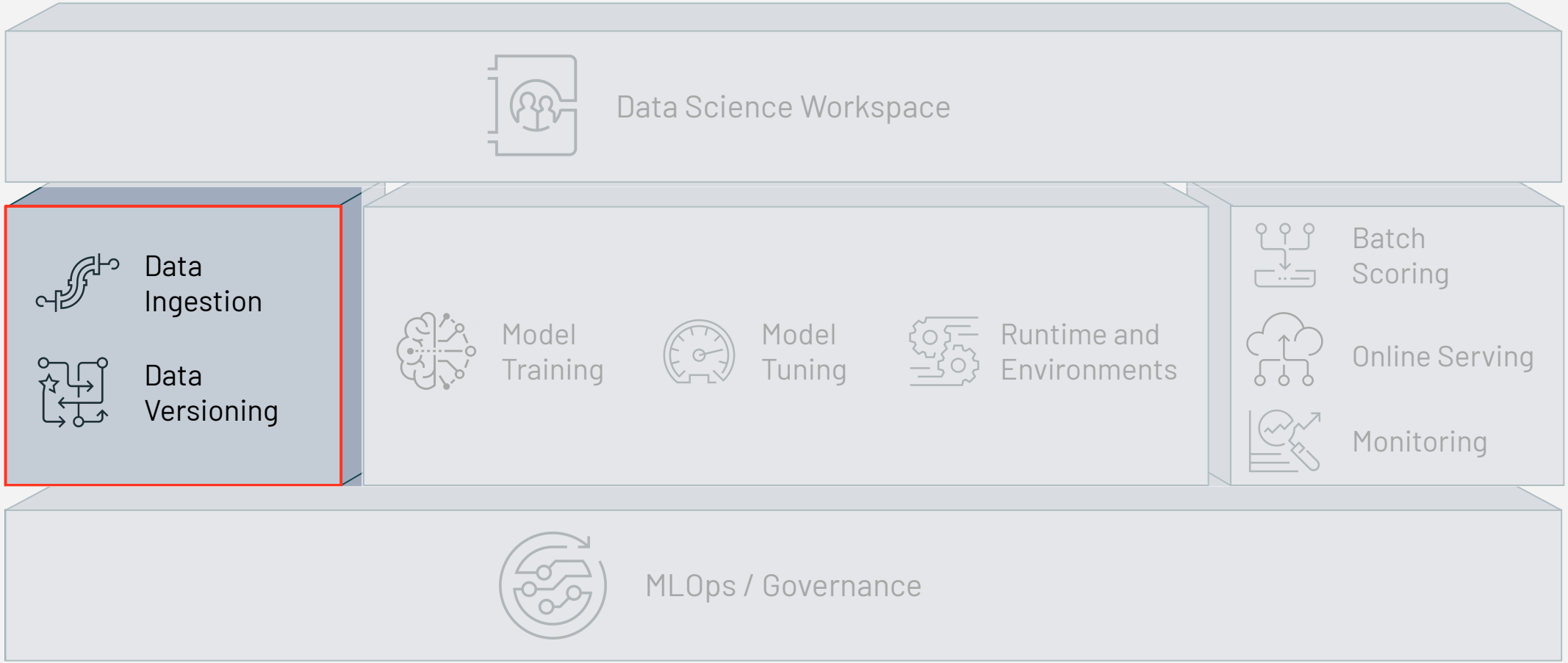


Supported Git Providers

Azure DevOps GitHub GitLab Bitbucket

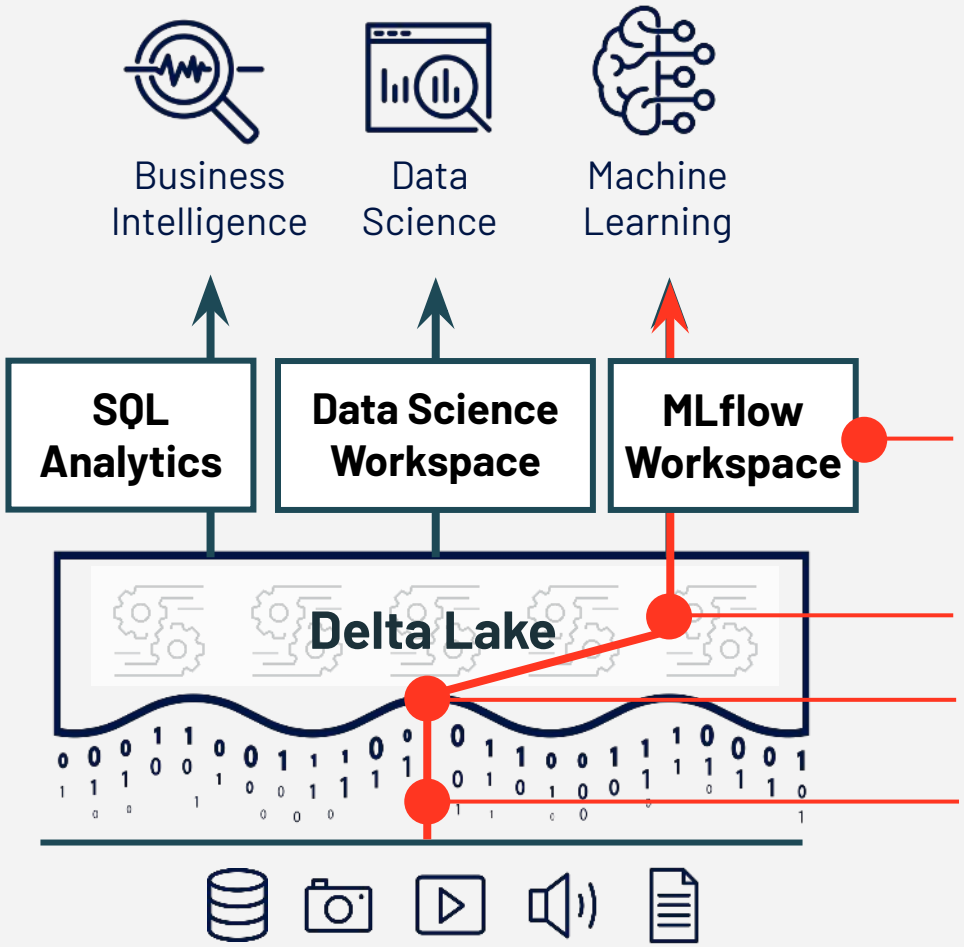
databricks

High Quality Data at Scale



databricks

High Quality Data at Scale

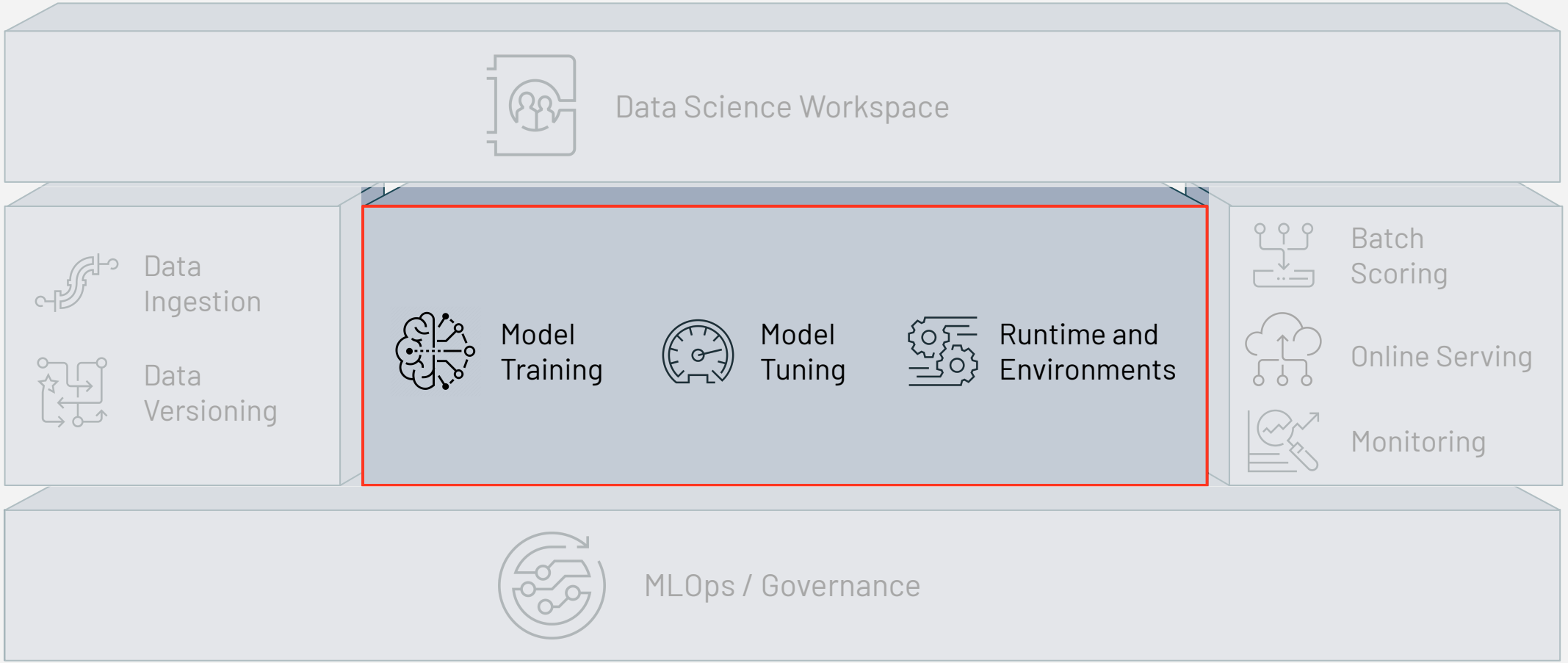


DELTA LAKE

- Automated logging of data + version information
- Versioning and time-travel built-in
- ACID transactions guarantee data validity
- Ingest any format at any scale from any source

databricks

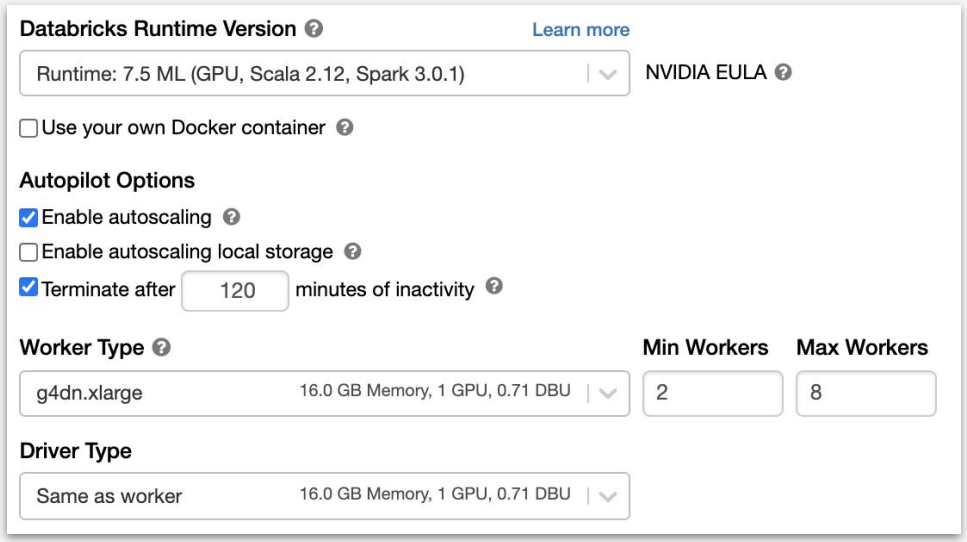
Turnkey ML Training at Scale



databricks

Turnkey ML Training at Scale

ML Runtime: DevOps-free Environment optimized for Machine Learning



Packages up the most popular ML Toolkits

Simplifies Distributed ML/DL

Distribute and scale any single-machine ML code to 1,000's of machines.

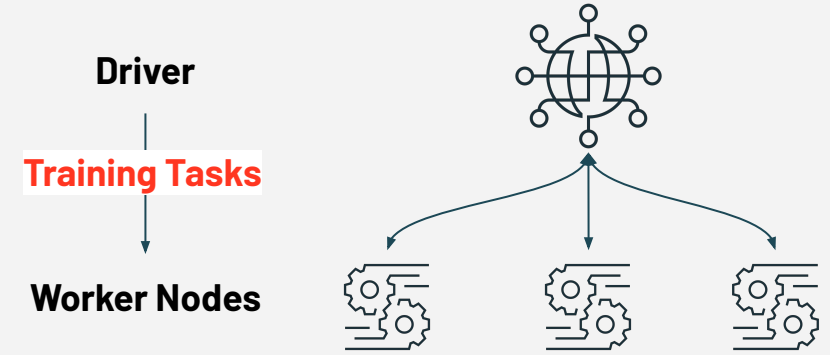
Built-in AutoML and Auto-Logging

Hyperparameter tuning, AutoML, automated tracking, and visualizations with MLflow

databricks

Distributed Training

- Built-in support in the ML Runtime



HorovodRunner (Keras, TensorFlow, and PyTorch)

```
hr = HorovodRunner(np=2)
hr.run(train)
```

TensorFlow native Distribution Strategy (Spark TensorFlow Distributor)

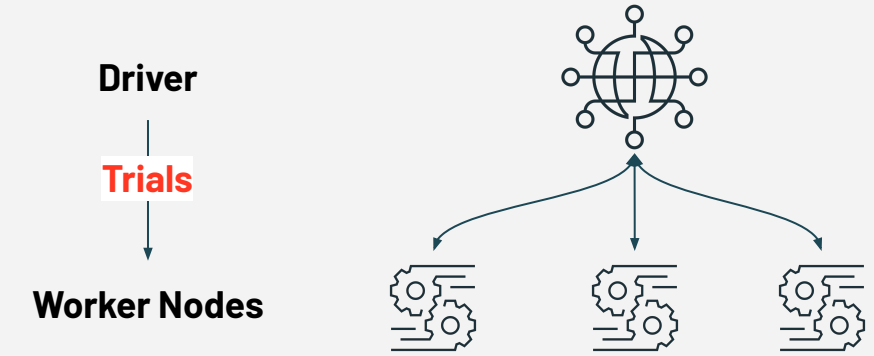
```
runner = MirroredStrategyRunner(num_slots=1, local_mode=True, use_gpu=USE_GPU)
runner.run(train)
```

Distributed Tuning

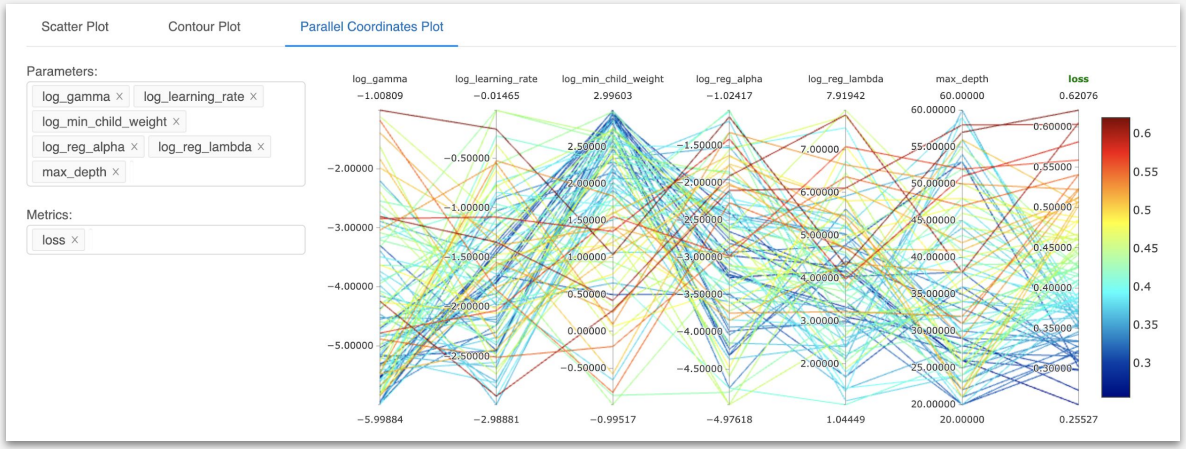
- Built-in support in the ML Runtime

```
search_space = {
  'max_depth':            hp.quniform('max_depth', 20, 60, 1),
  'log_learning_rate':    hp.uniform('log_learning_rate', -3, 0),
  'log_reg_alpha':        hp.uniform('log_reg_alpha', -5, -1),
  'log_reg_lambda':       hp.uniform('log_reg_lambda', 1, 8),
}

best_params = fmin(fn=train_model,
                   space=search_space,
                   algo=tpe.suggest,
                   max_evals=96,
                   trials=SparkTrials(parallelism=6))
```

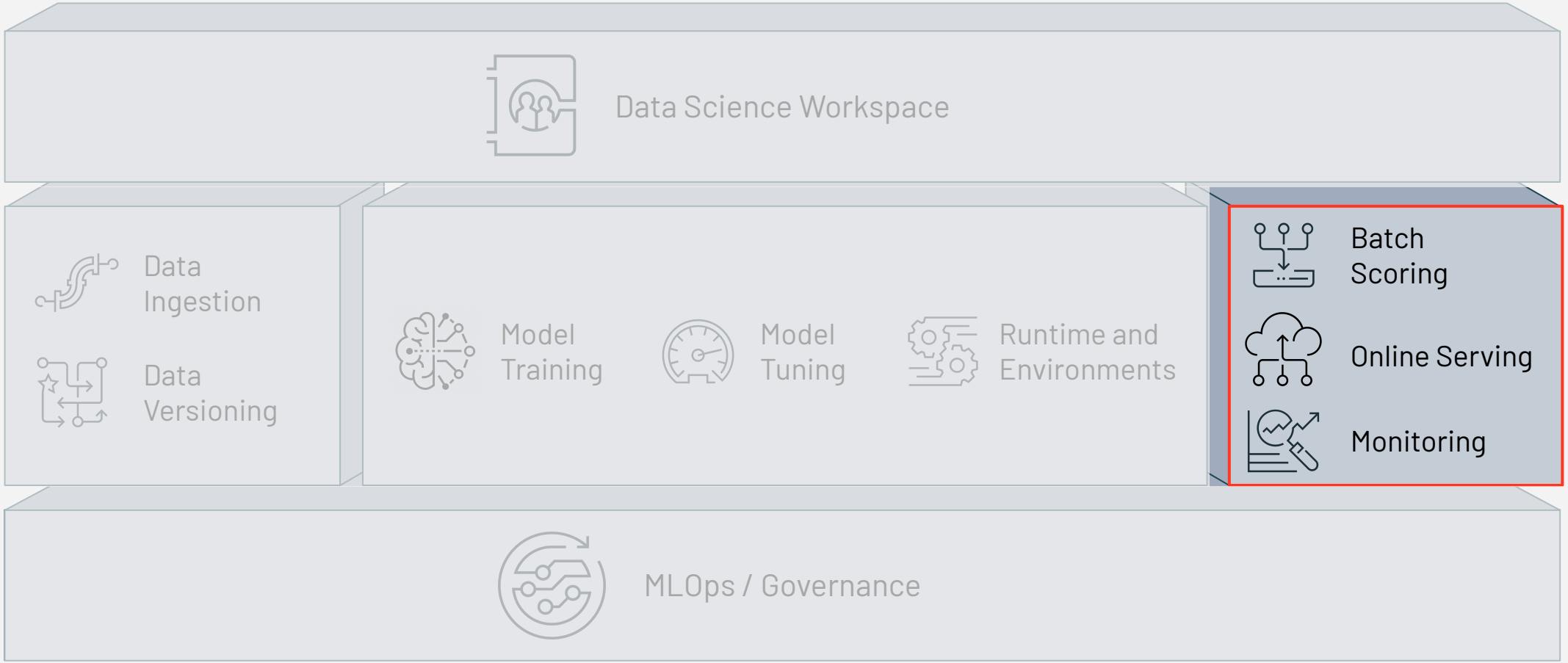


mlflow Integration



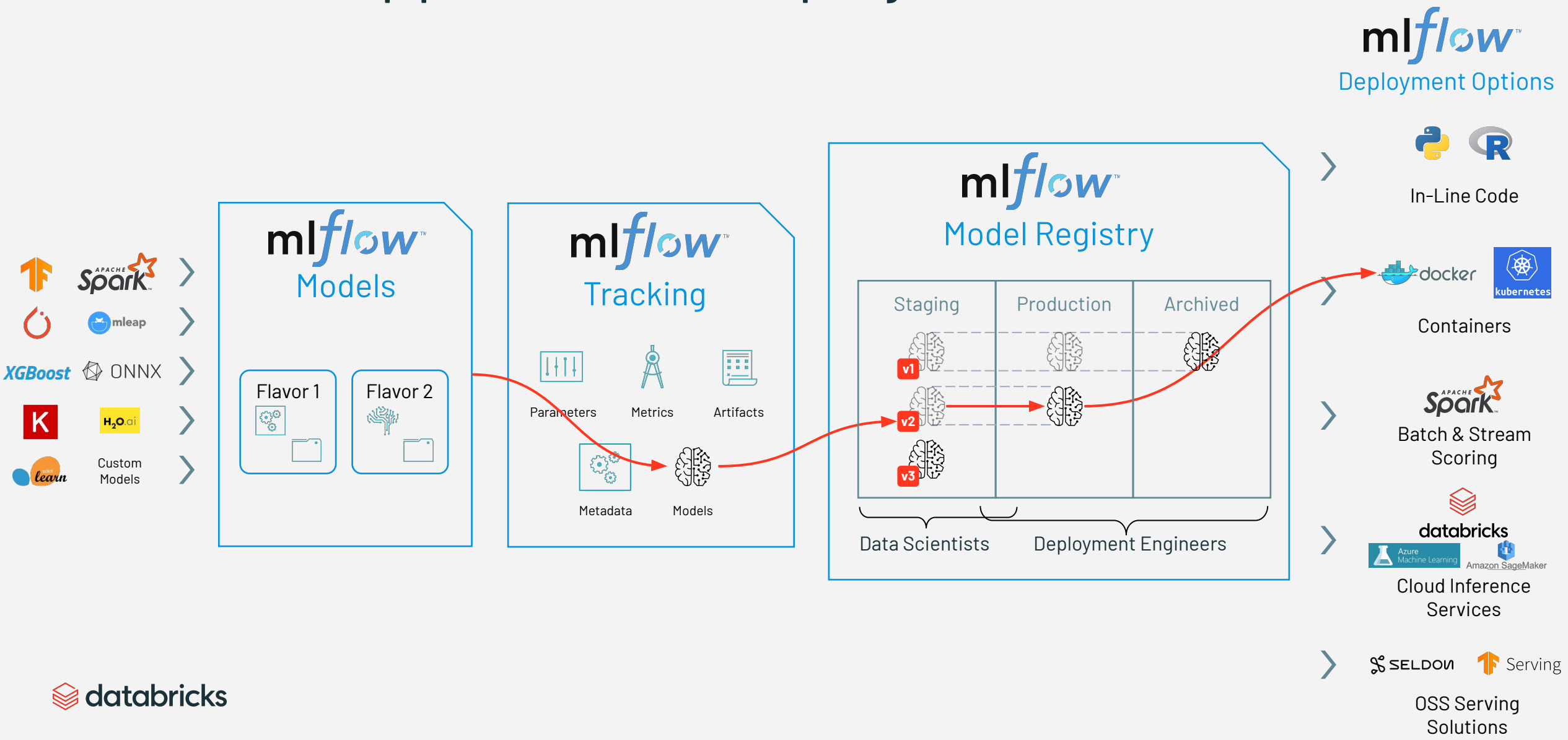
databricks

Support for all Deployment Modes



databricks

Support for all Deployment Modes



databricks

Support for all Deployment Modes

Deploying an MLLib model as a Spark UDF

```
model_udf = mlflow.pyfunc.spark_udf(spark, model_uri='models:/clemens-windfarm-signature/production')
df.with_column('prediction', model_udf)
```

Support for all Deployment Modes

Deploying an MLLib model as a Spark UDF

Deploying a Scikit Learn model as a Spark UDF

```
model_udf = mlflow.pyfunc.spark_udf(spark, model_uri='models:/clemens-windfarm-signature/production')
df.with_column('prediction', model_udf)
```

Support for all Deployment Modes

Deploying an MLLib model as a Spark UDF

Deploying a Scikit Learn model as a Spark UDF

Deploying a TensorFlow model as a Spark UDF

```
model_udf = mlflow.pyfunc.spark_udf(spark, model_uri='models:/clemens-windfarm-signature/production')
df.with_column('prediction', model_udf)
```

Support for all Deployment Modes

Deploying an MLLib model as a Spark UDF

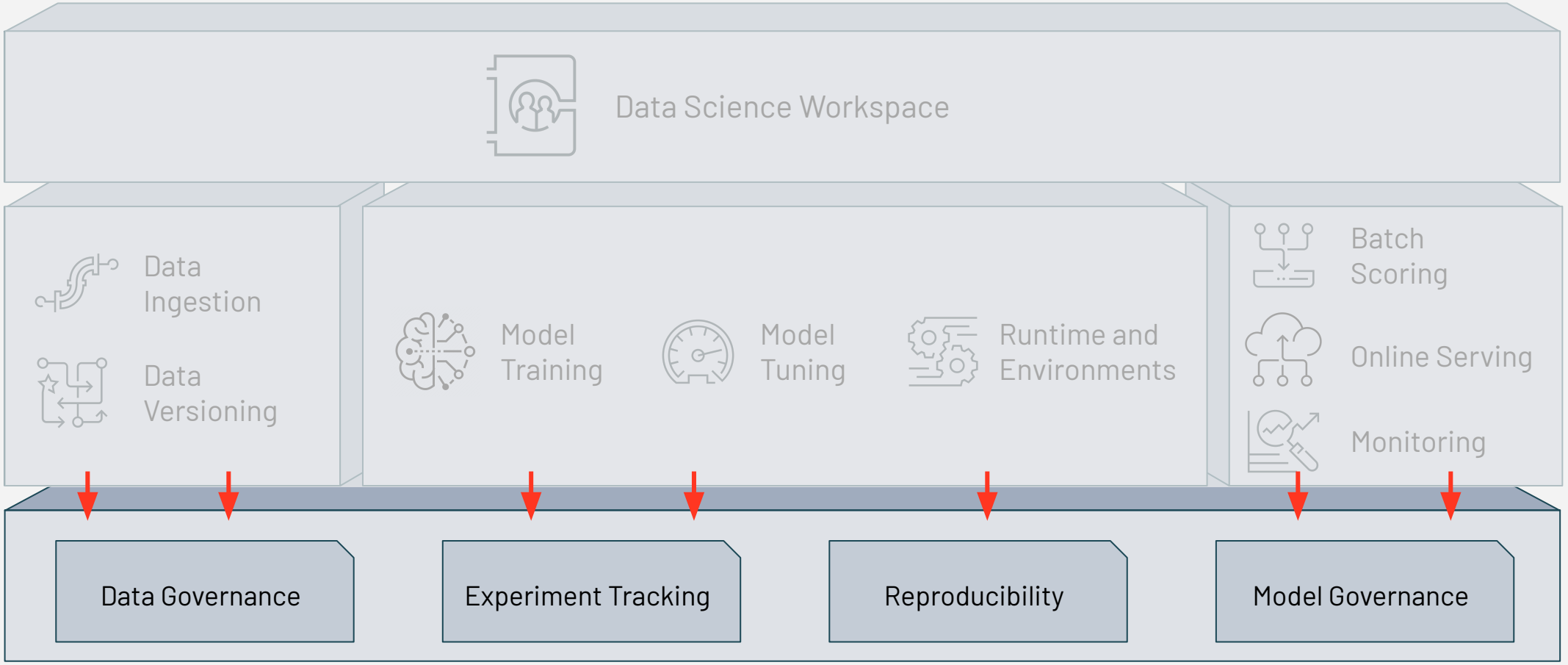
Deploying a Scikit Learn model as a Spark UDF

Deploying a TensorFlow model as a Spark UDF

```
model_udf = mlflow.pyfunc.spark_udf(spark, model_uri='models:/clemens-windfarm-signature/production')
df.with_column('prediction', model_udf)
```

Yes, they're all the same! As are the commands to deploy these models as Docker containers, etc.

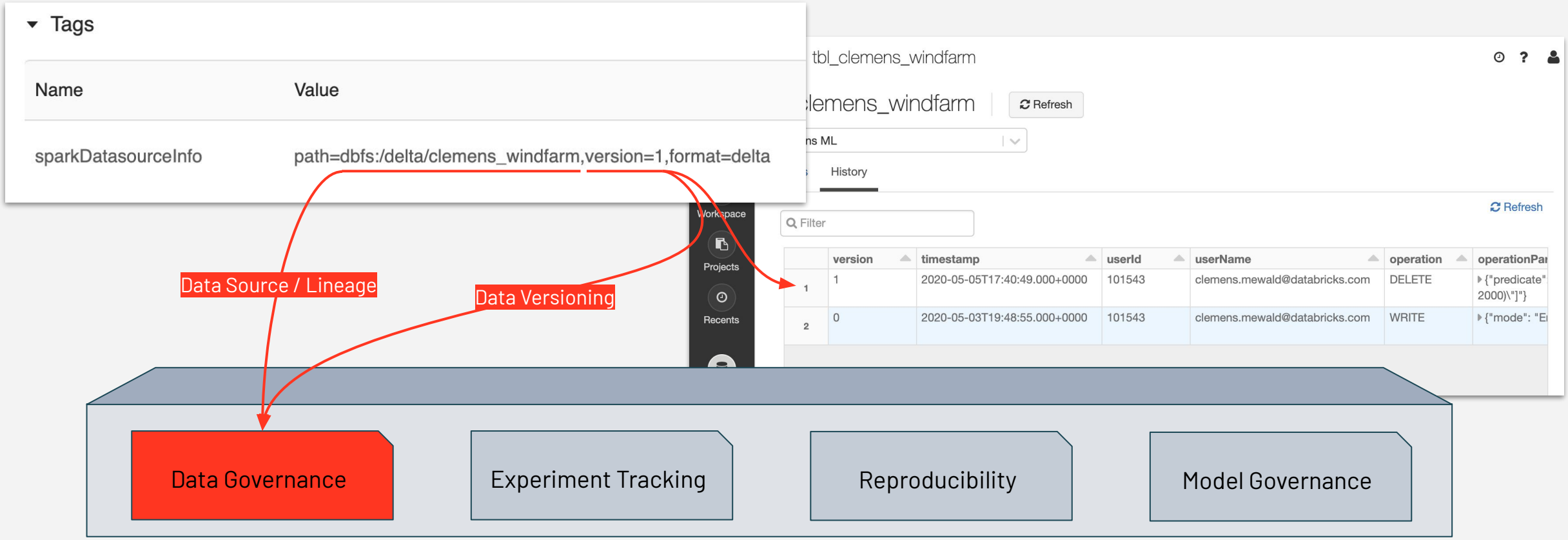
End-to-end MLOps / Governance



databricks

Powered by mlflow

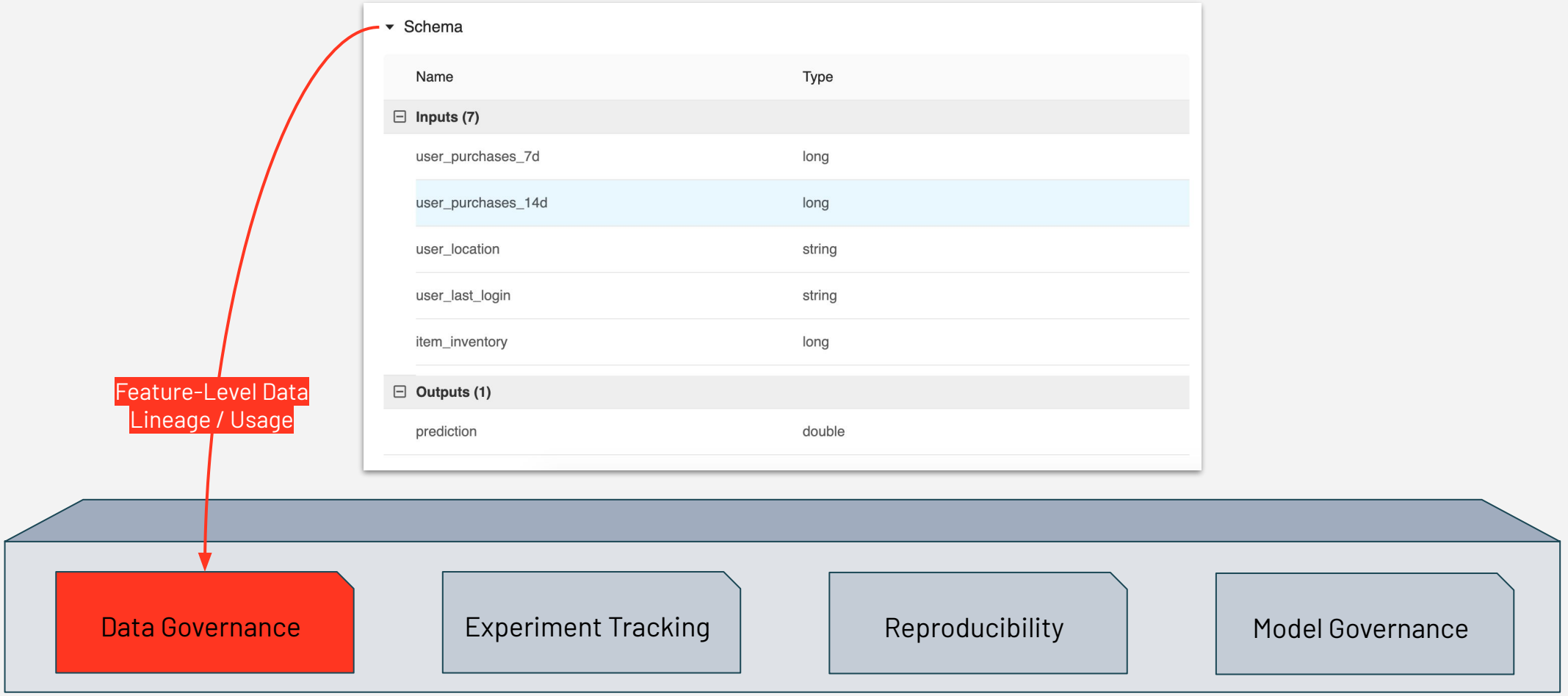
Automated Data Source capture and Versioning



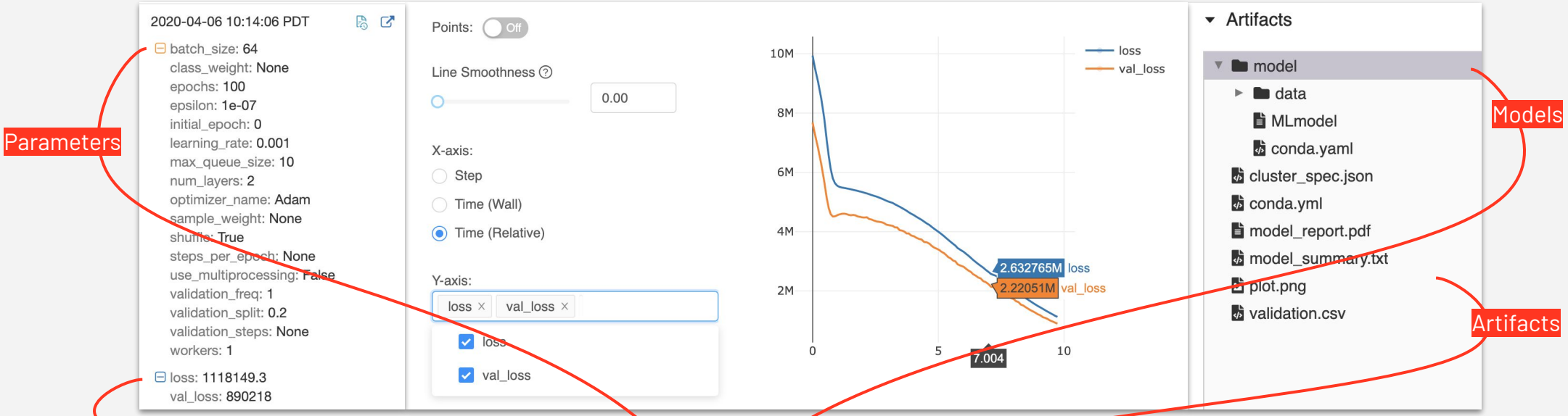
databricks

Powered by mlflow

Automated capture of Feature Usage



Automated capture of ML metrics, parameters, artifacts, etc.



Parameters

Metrics

Models

Artifacts

Data Governance

Experiment Tracking

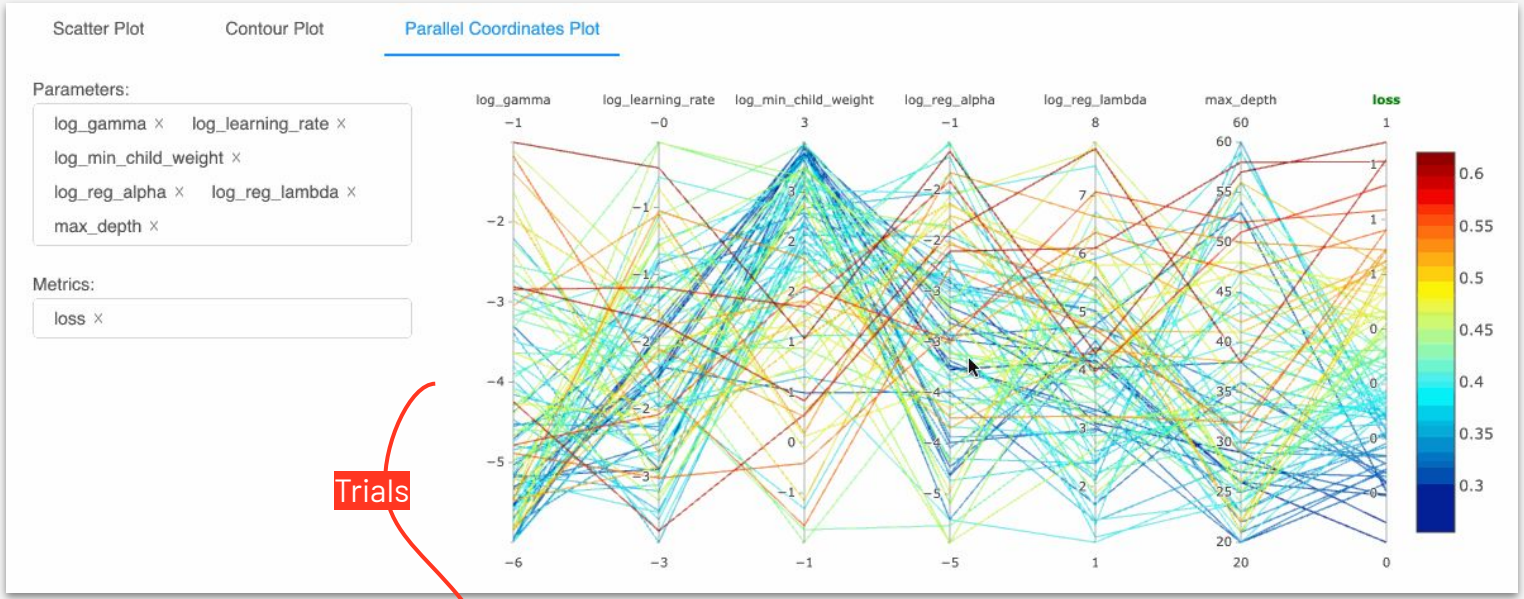
Reproducibility

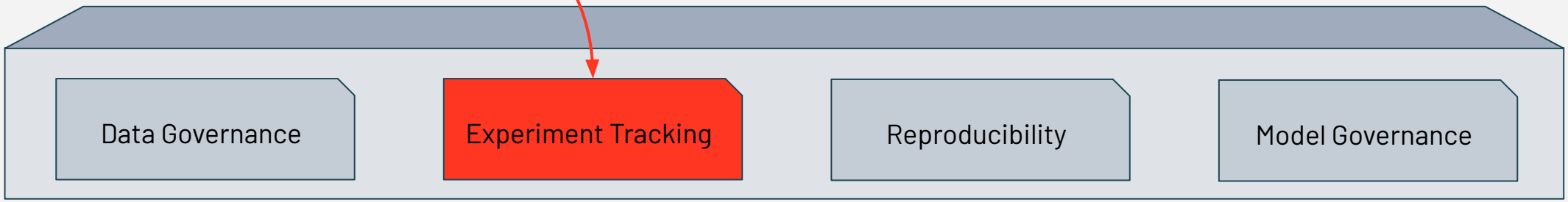
Model Governance

databricks

Powered by mlflow

Automated capture of Hyperparameter Search

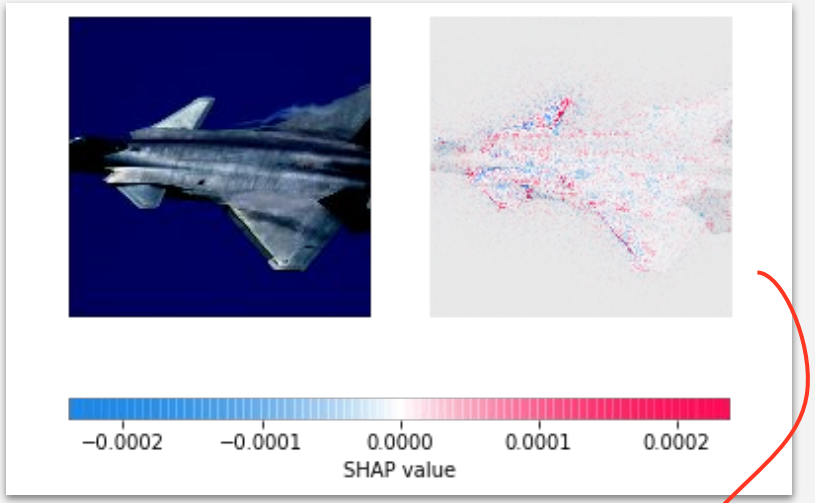


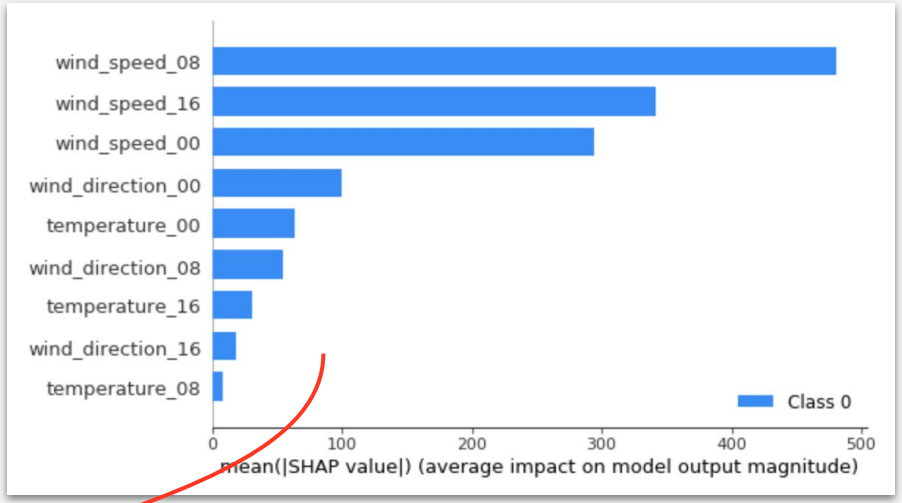


databricks

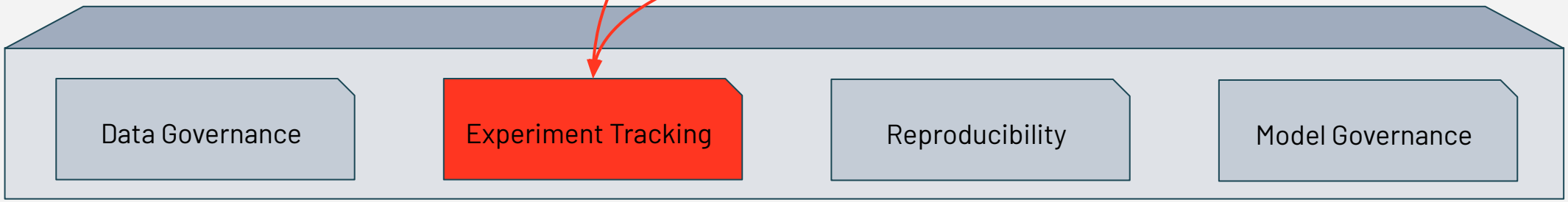
Powered by mlflow

Automated Model Interpretability





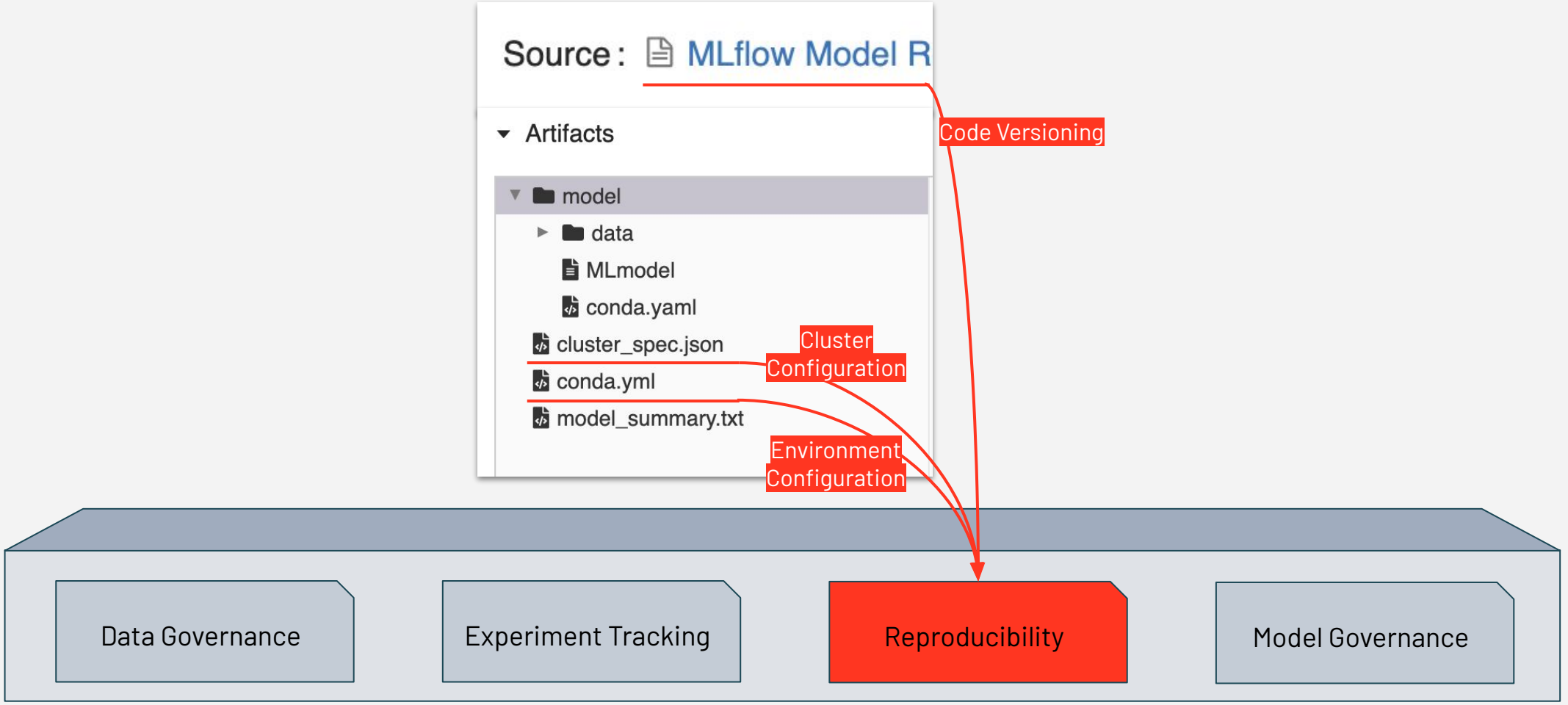
Model Interpretability



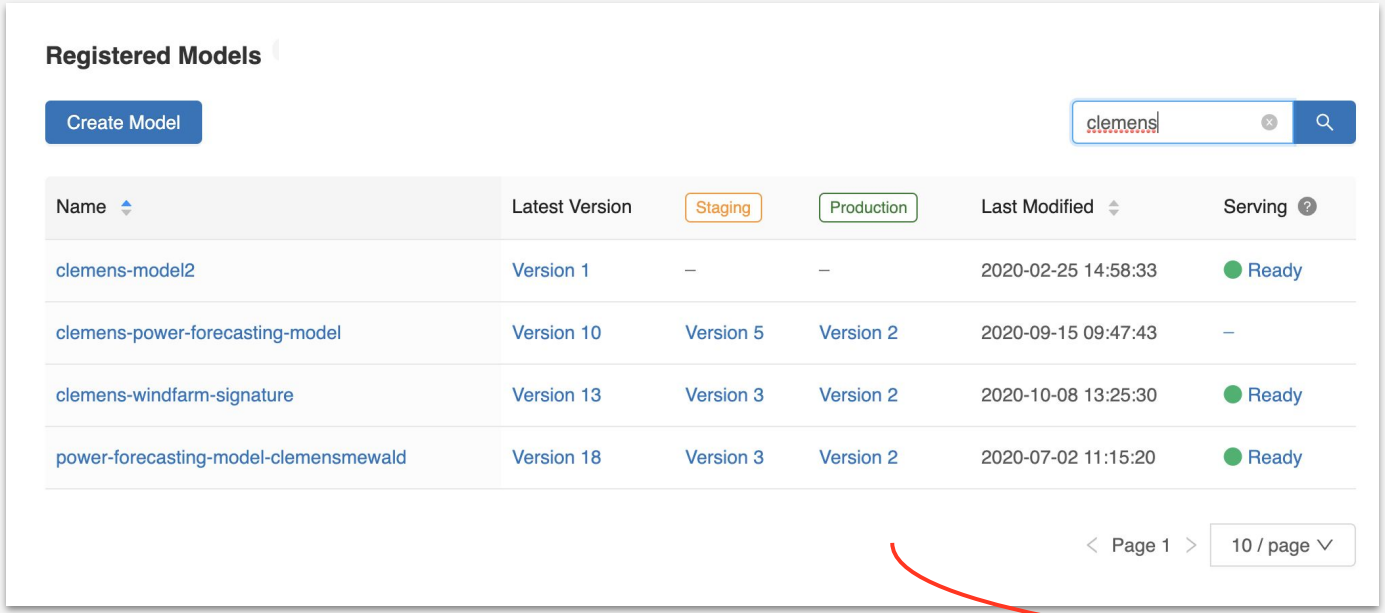
databricks

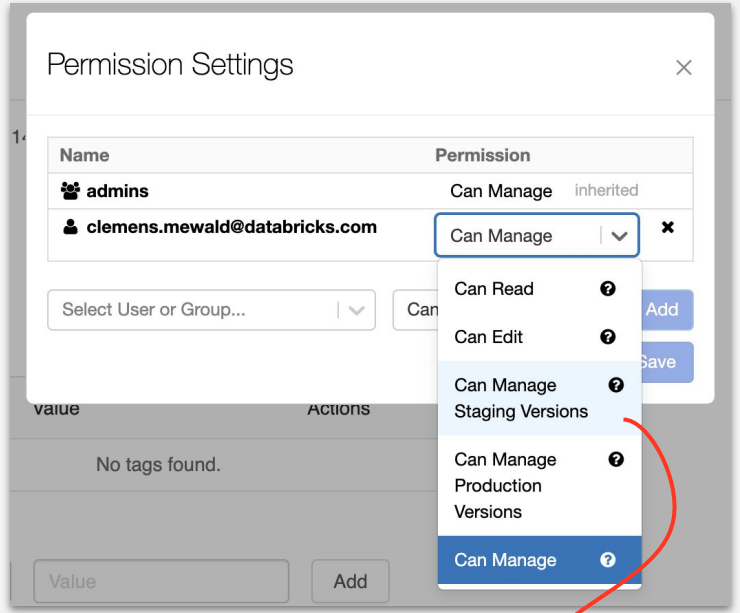
Powered by mlflow

Automated capture of Code, Environment and Cluster Specification



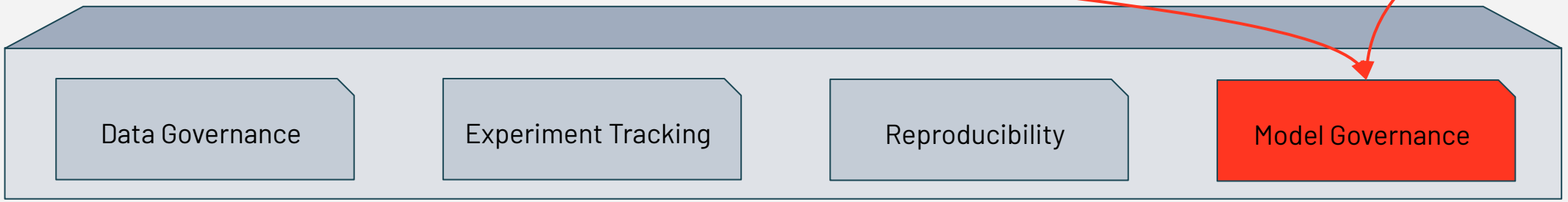
Model Sharing, Reuse, and ACLs





Model Discoverability

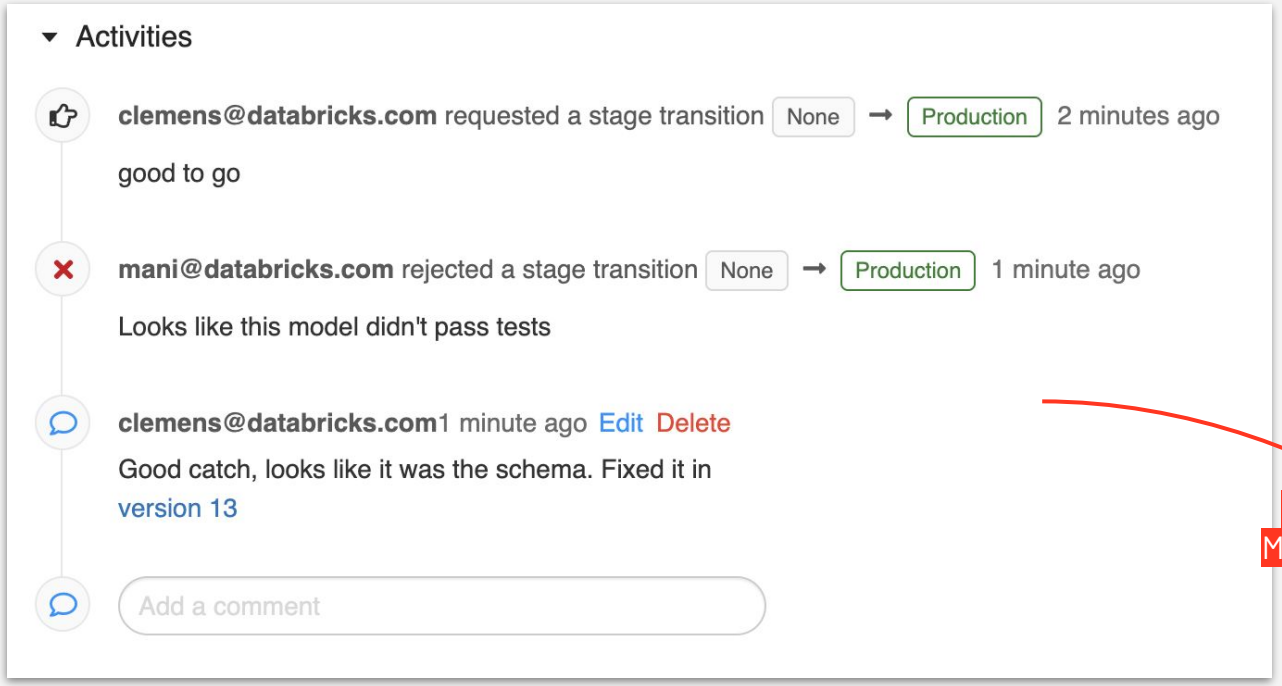
Model Stage-Based ACLs

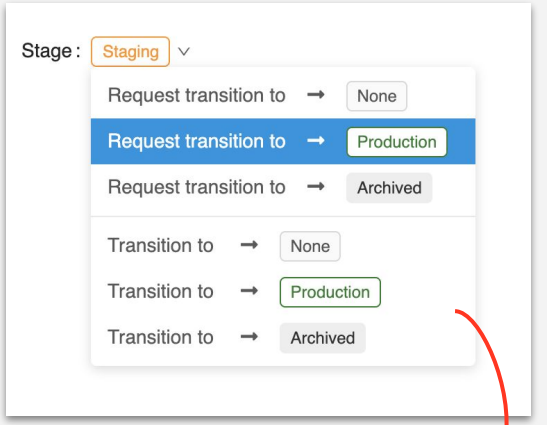


databricks

Powered by mlflow

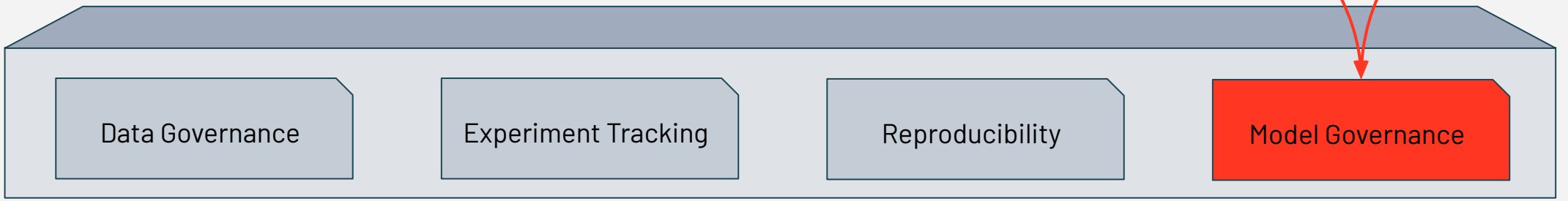
Automated Model Lineage and Governance





Audit Log of Model Changes

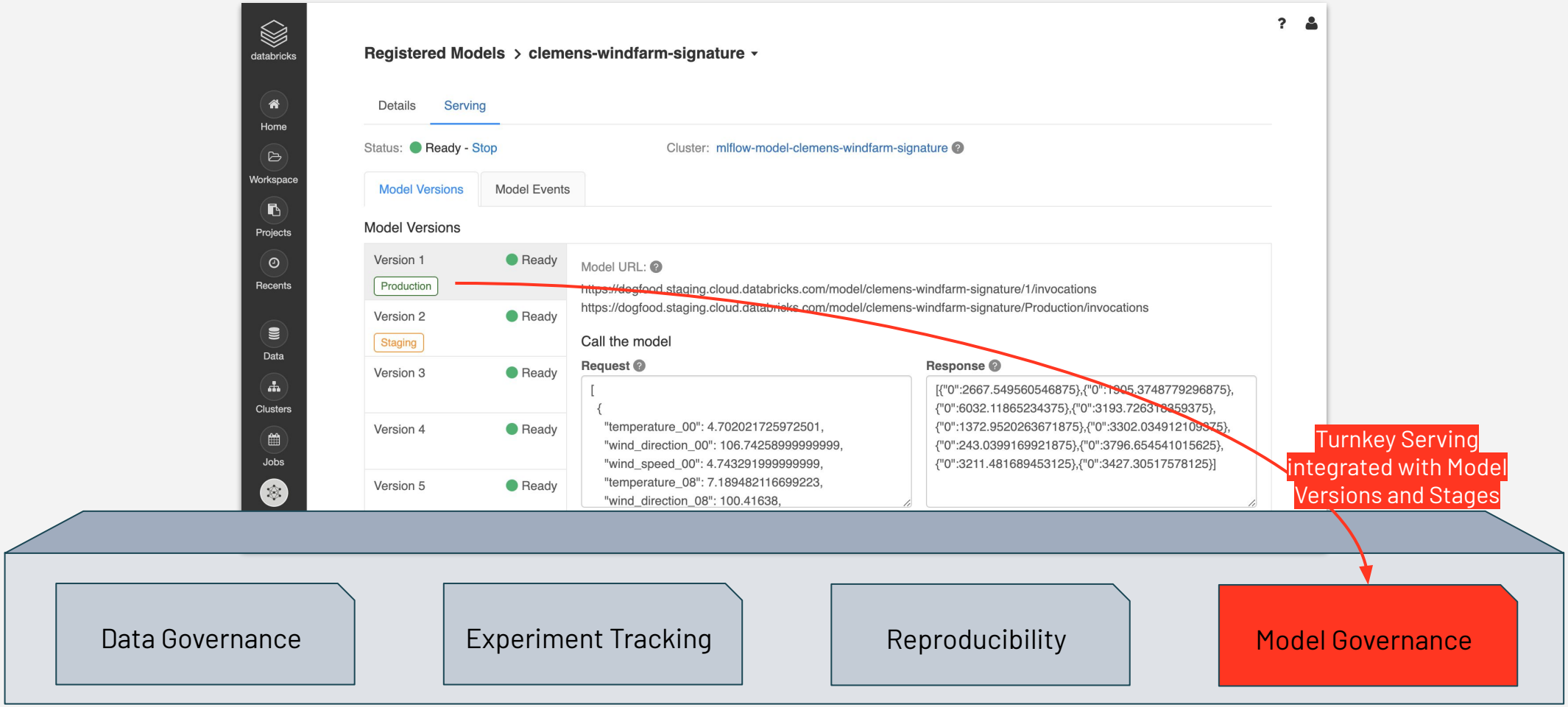
Approval Process for Stage Transitions



databricks

Powered by mlflow

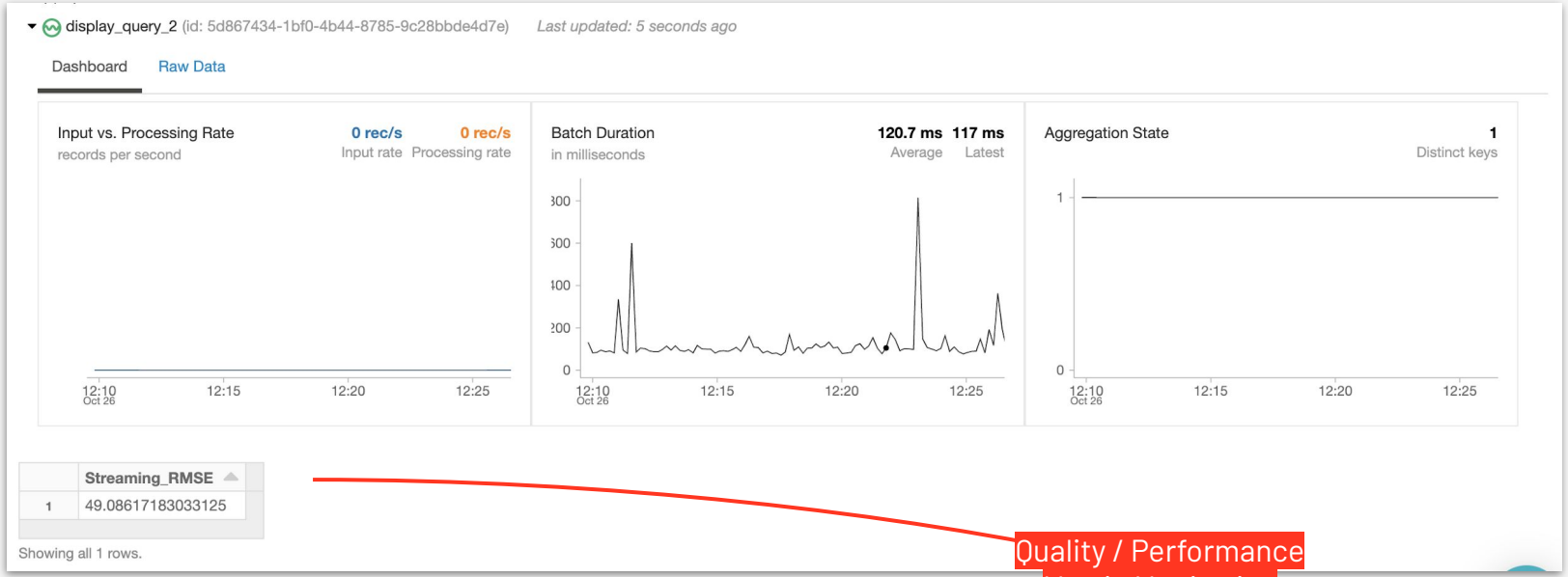
Turnkey Model Serving



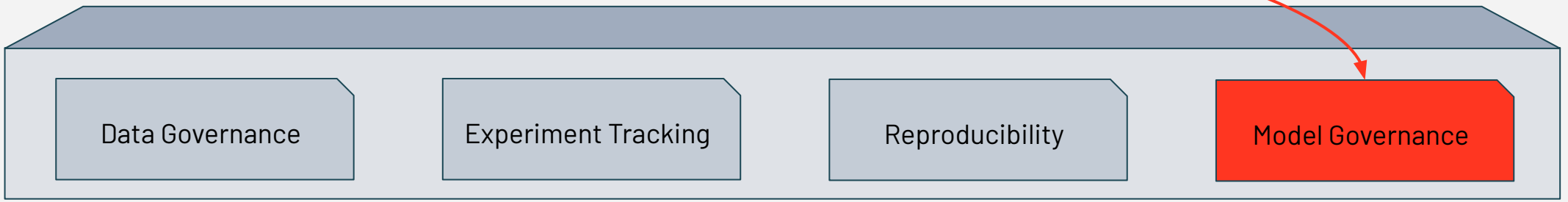
databricks

Powered by mlflow

Model Quality monitoring



Quality / Performance Metric Monitoring

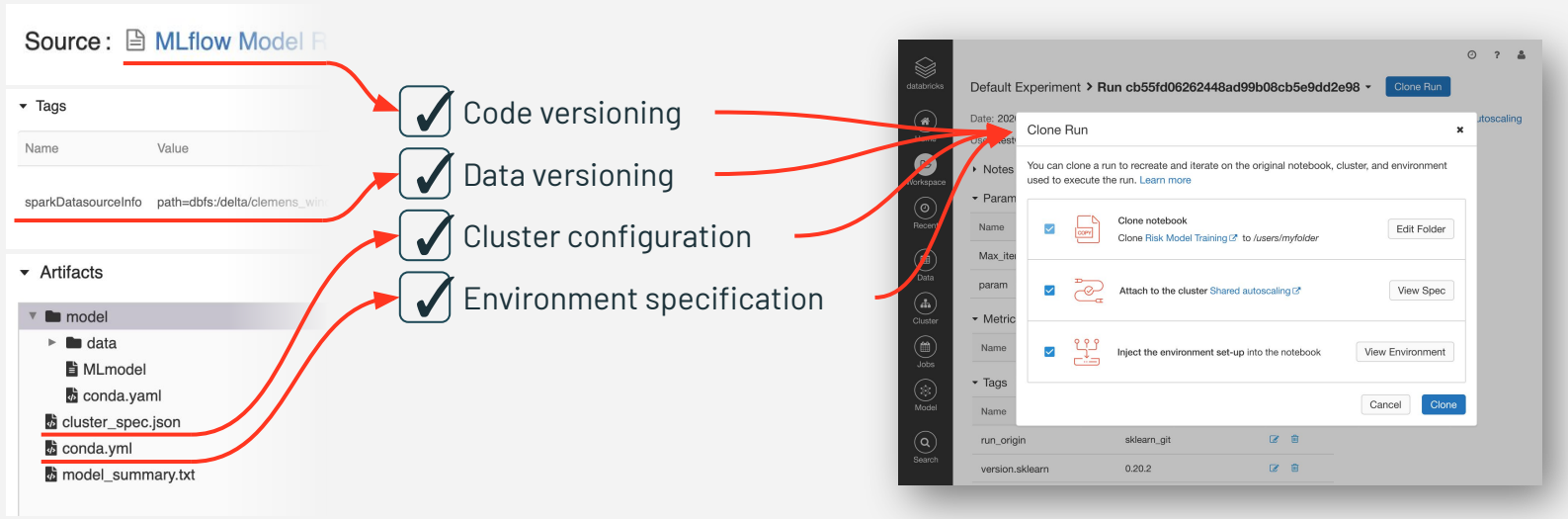


databricks

Powered by mlflow

The Result: Full End-to-End Governance and Reproducibility

Auto-Logging | **Reproducibility Checklist** | **Reproduce Run Feature**



Data Governance | Experiment Tracking | Reproducibility | Model Governance

databricks

Powered by mlflow

The Databricks ML Platform