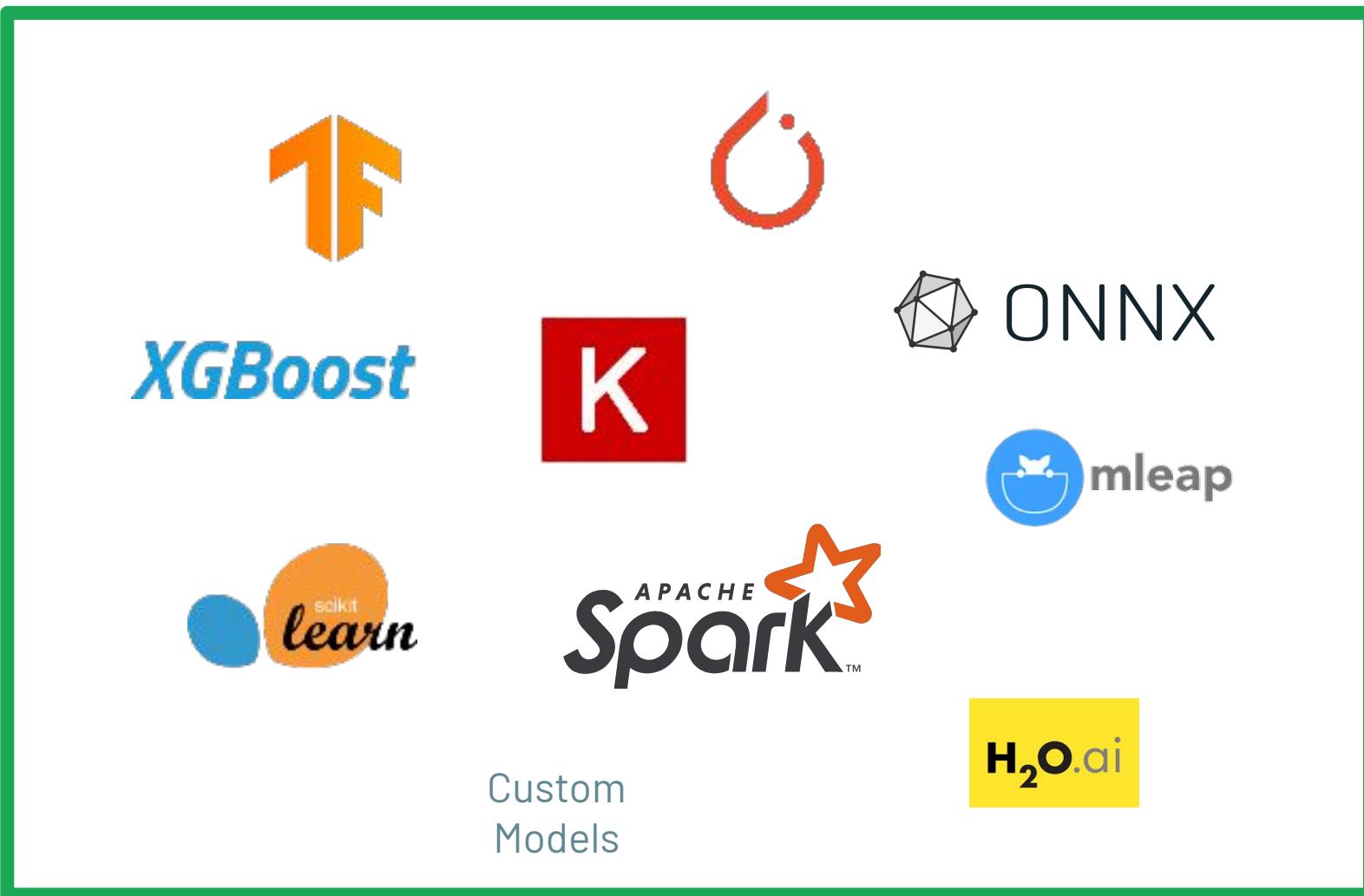


# MSFT- ADB Radio Series Part III

John O'Dwyer - Developer Advocate

Nikhil Gupta - Sr. Solution Architect

# Machine Learning and AI



# Hardest Part of AI isn't AI, it's Big Data

***"Hidden Technical Debt in Machine Learning Systems," Google NIPS 2015***

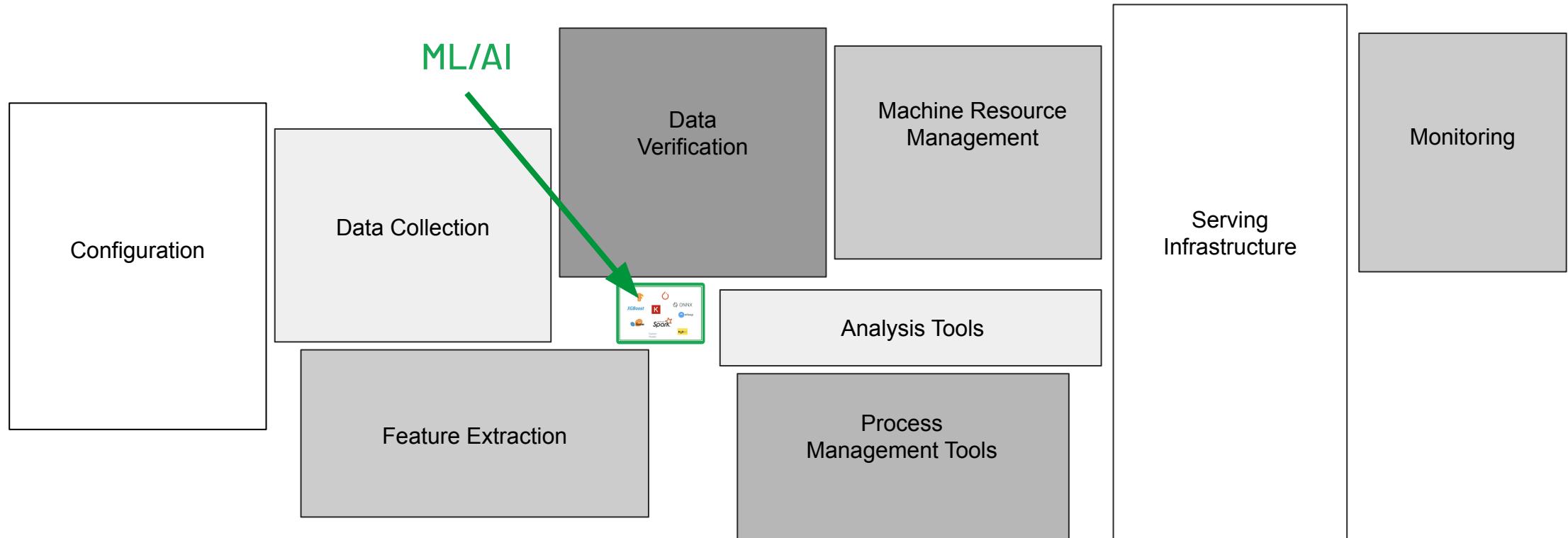
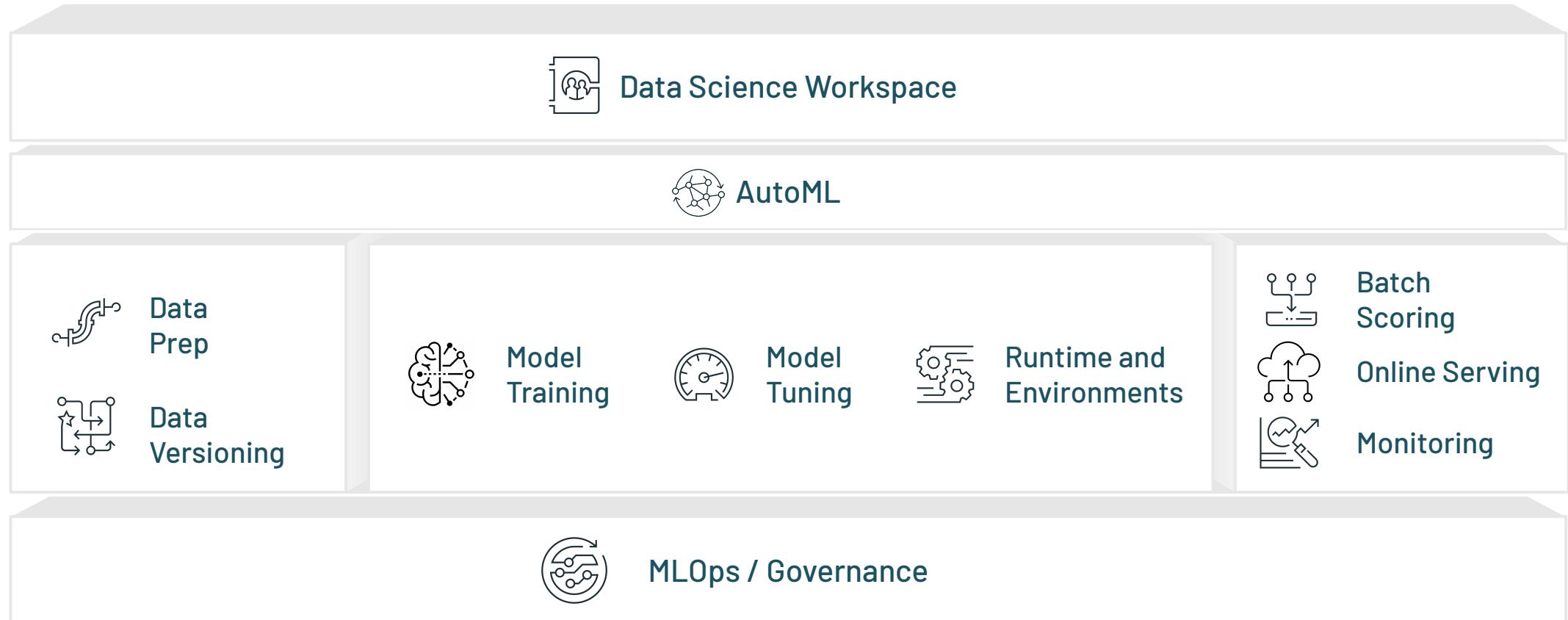


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small green box in the middle. The required surrounding infrastructure is vast and complex.

# Databricks Machine Learning

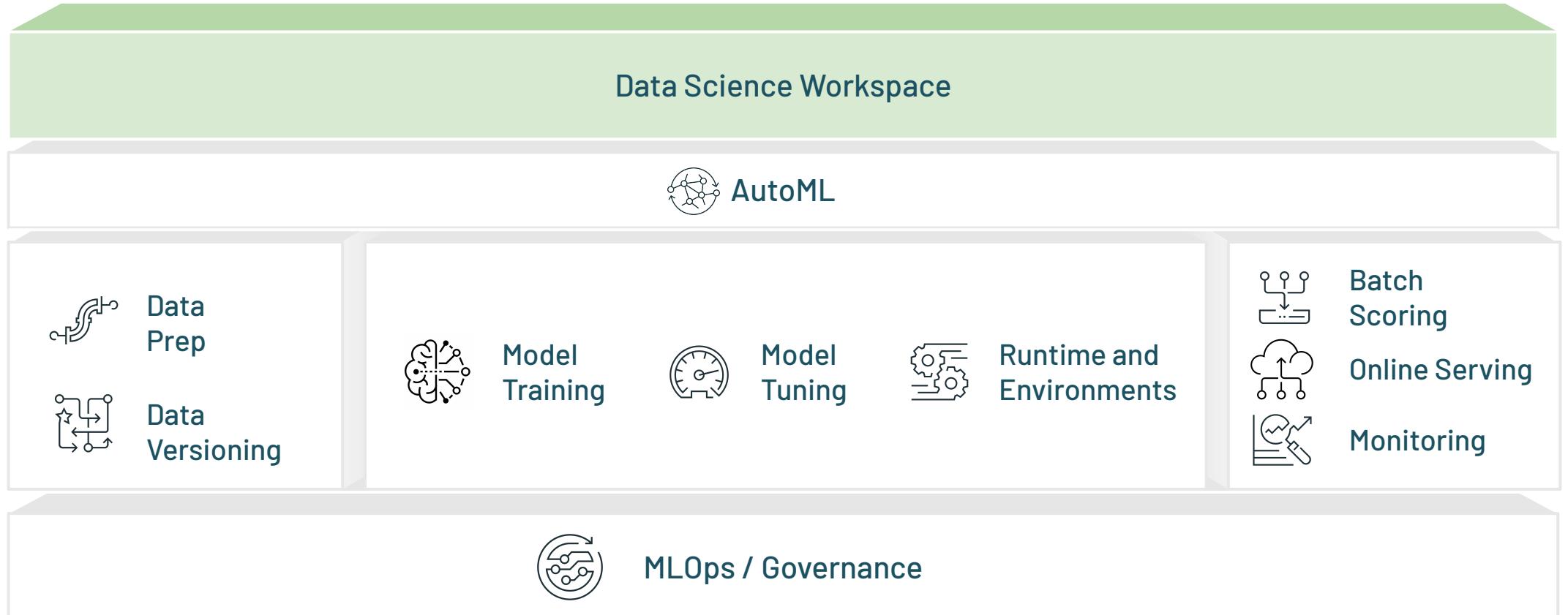
## A data-native and collaborative solution for the full ML lifecycle



Open Data Lake Foundation with



# Data Science Workspace



Open Data Lake Foundation with



# Databricks Notebooks

Provide a collaborative environment for Unified Data Analytics

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

Visualizations  
Built-in visualizations and support for the most popular visualization libraries (e.g. matplotlib, ggplot)

Experiment Tracking  
Built-in tracking of Data Science and ML experiments, with metrics, parameters, artifacts, and more

The screenshot shows the Databricks Notebook interface. On the left is a sidebar with icons for Home, Workspace, Projects, Recents, Data, Clusters, Jobs, Models, and Search. The main area has two code cells:

**Cmd 1:**

```
%scala  
val flightdelays_df = spark.read.format("delta").load("/ml/flightdelays_delta_all")  
  
display(flightdelays_df)
```

(1) Spark Jobs

flightdelays\_df: org.apache.spark.sql.DataFrame = [Year: integer, Month: integer ... 27 more fields]

Year	Month	DayofMonth	DayOfWeek	DepTime	CRSDepTime	ArrTime	CRSArrTime	UniqueCarrier	FlightNu
2002	1	1	2	1100	665	1619	970	DL	1034
2002	1	2	3	1117	665	1948	970	DL	1034
2002	1	3	4	1105	665	1635	970	DL	1034

Showing the first 1000 rows.

Command took 2.03 seconds clemens@databricks.com at 1/3/2020, 10:32:18 AM on Shared Autoscaling

**Cmd 2:**

```
%sql  
SELECT year, count(*) FROM flightdelays GROUP BY year ORDER BY year
```

(2) Spark Jobs

count(1)

year

A bar chart showing the count of flights per year from 1987 to 2007. The y-axis is labeled "count(1)" and ranges from 0 to 6M. The x-axis is labeled "year" and shows years from 1987 to 2007. The bars show a general upward trend with a notable dip around 2000.

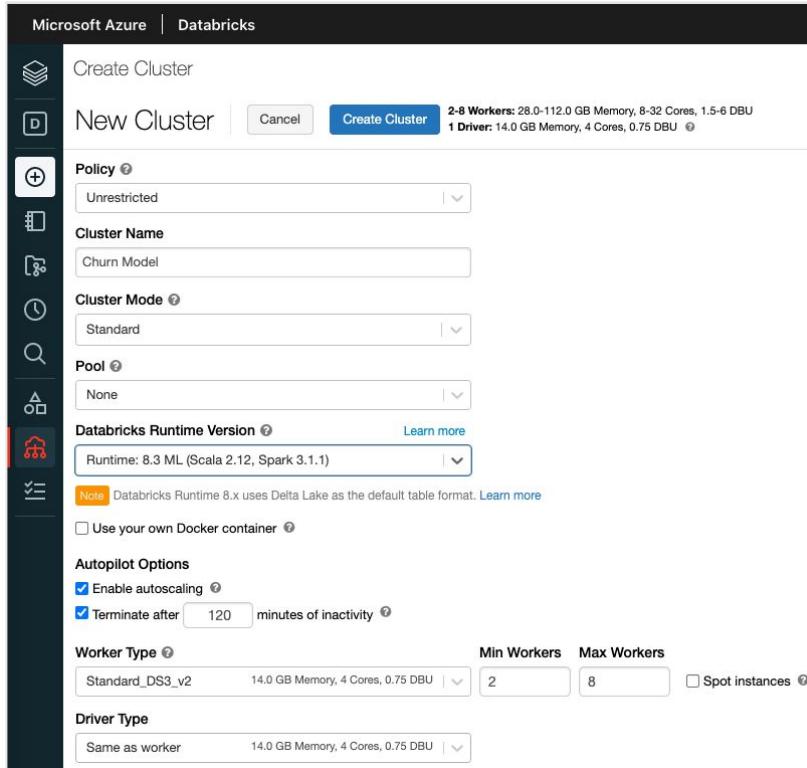
Reproducible  
Auto-logged revision history and Git integration with Azure Devops for version control

Collaborative  
Realtime co-editing and commenting

Enterprise Ready  
Enterprise grade access controls, identity pass-through, and auditability

# Hassle-free Compute Infrastructure

## Clusters for data processing and model creation



- Compatible with all supported instance/VM types
- User-friendly cluster management
- Optimized Spark performance
- Cost-optimized autoscaling
- GPU support
- Backward compatibility with automatic upgrades
- Data caching

Configure an optimized cluster with just a few clicks.

# Databricks ML Runtime

## Infrastructure Optimized for ML

Create Cluster

New Cluster

2-8 Workers: 61.0-244.0 GB Memory, 8-32 Cores, 2-8 DBU  
1 Driver: 30.5 GB Memory, 4 Cores, 1 DBU

Cluster Name: ml-cluster

Cluster Mode: Standard

Pool: None

Databricks Runtime Version: Runtime: 6.2 ML (Scala 2.11, Spark 2.4.4) [Learn more](#)

**Databricks Runtime**

6.2	Scala 2.11, Spark 2.4.4
6.2 Genomics	Scala 2.11, Spark 2.4.4
6.2 ML	GPU, Scala 2.11, Spark 2.4.4
<b>6.2 ML</b>	Scala 2.11, Spark 2.4.4
15 more	
<b>Deprecated</b>	
5.3 HLS Beta	Scala 2.11, Spark 2.4.1

Driver Type: Same as worker 30.5 GB Memory, 4 Cores, 1 DBU

Min Workers: 2 Max Workers: 8

## Packaged most common ML Tools

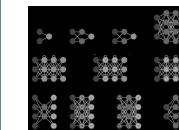


## Simplified Distributing ML/DL Libraries



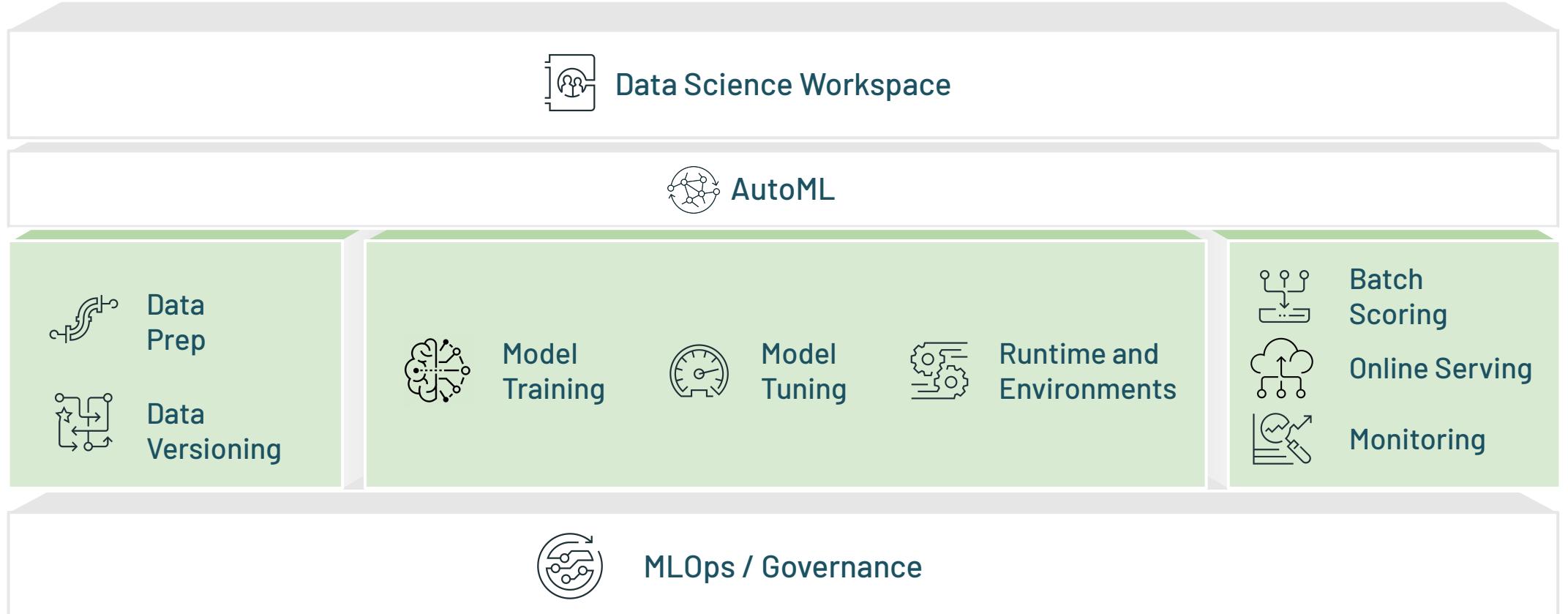
Distribute and Scale any Single-Machine ML Code to 1,000's of machines.

## Built-In AutoML and MLflow



AutoML and Tracking / Visualizations with MLflow

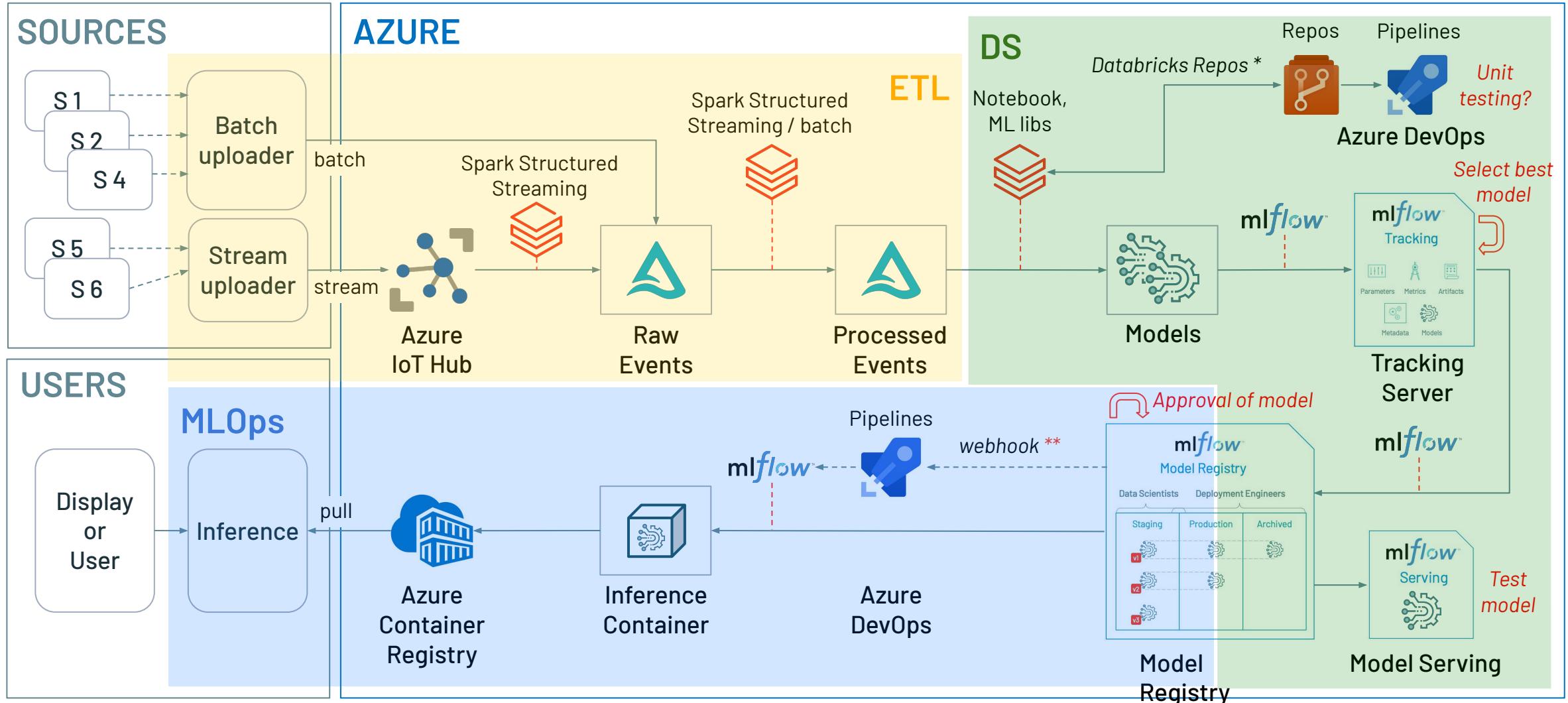
# MLflow - From data to model serving end to end



Open Data Lake Foundation with

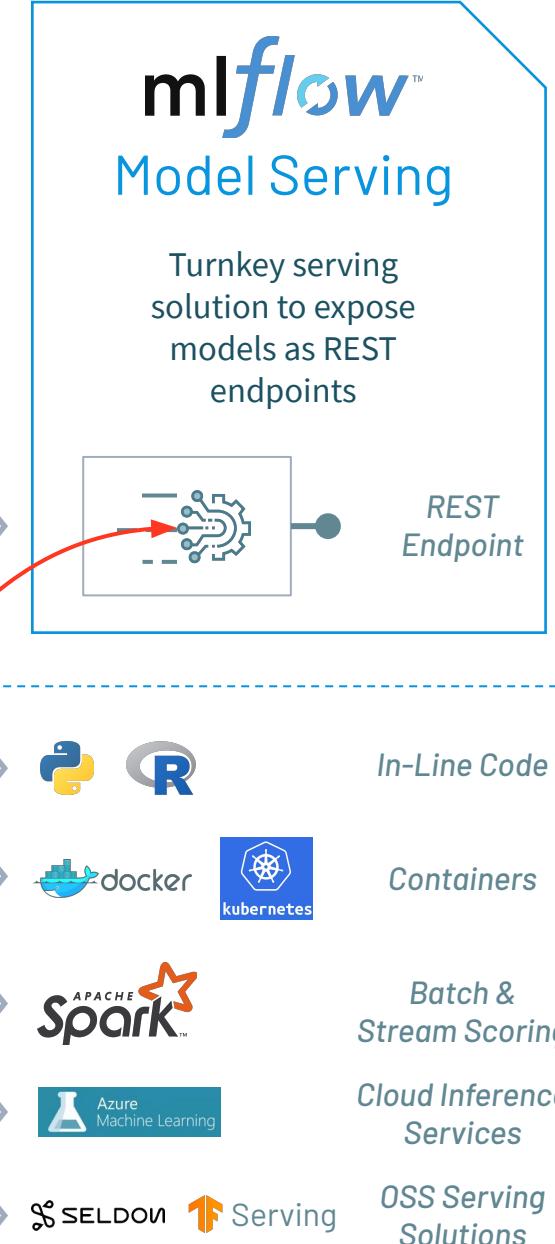
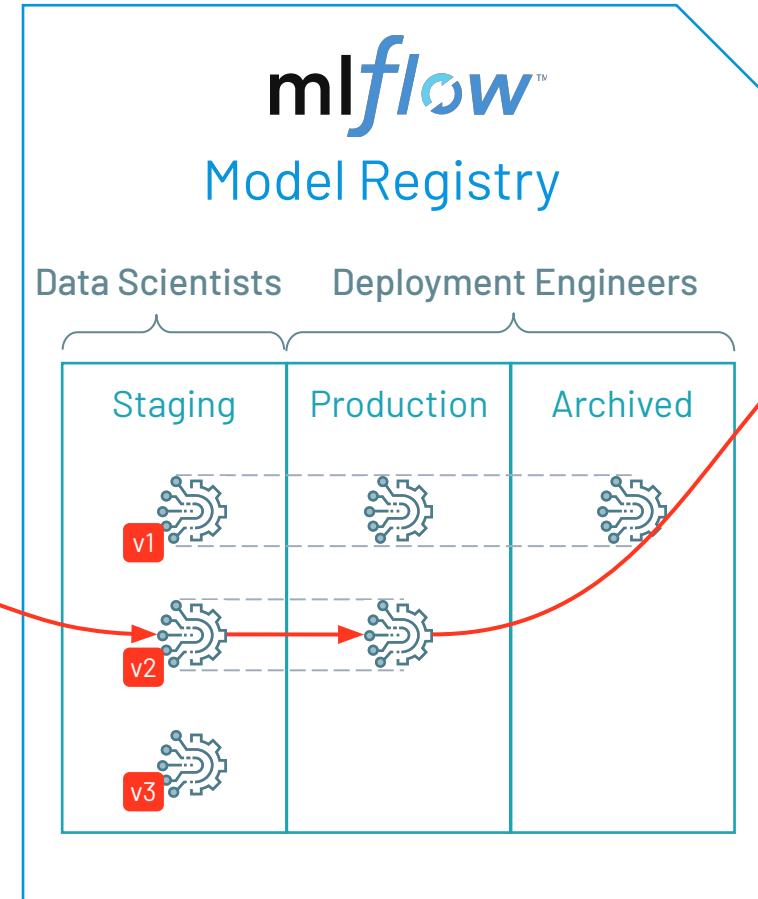
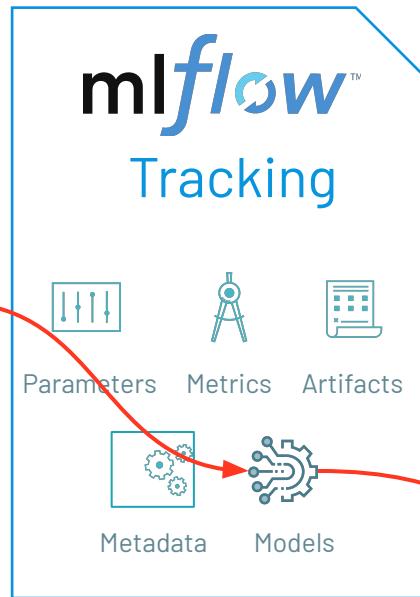
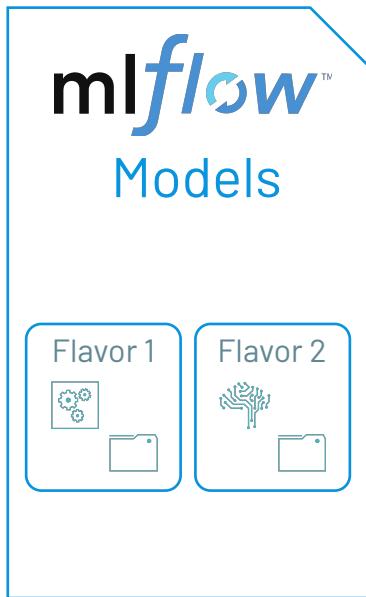


# The Big Picture end to end

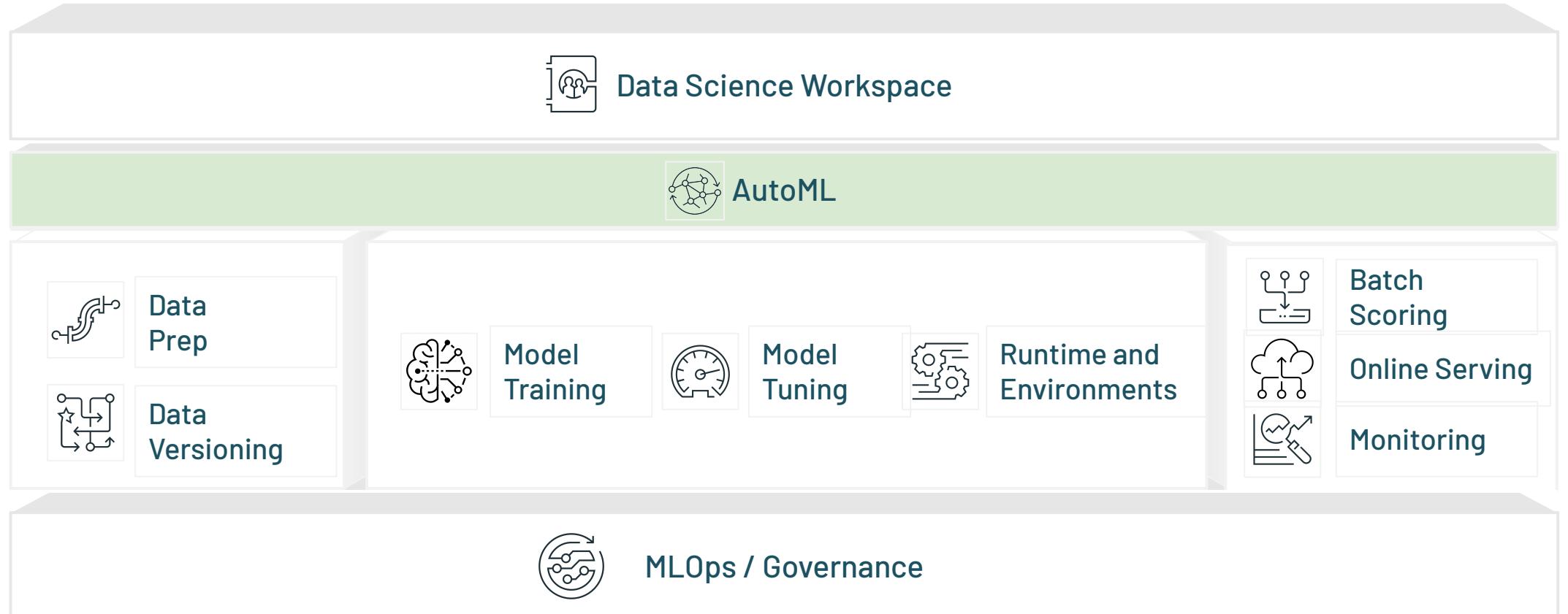


# mlflow™ Under the hood

TensorFlow  
Caffe  
XGBoost  
Keras  
TensorFlow learn  
Apache Spark™  
mleap  
ONNX  
H2O.ai  
Custom Models



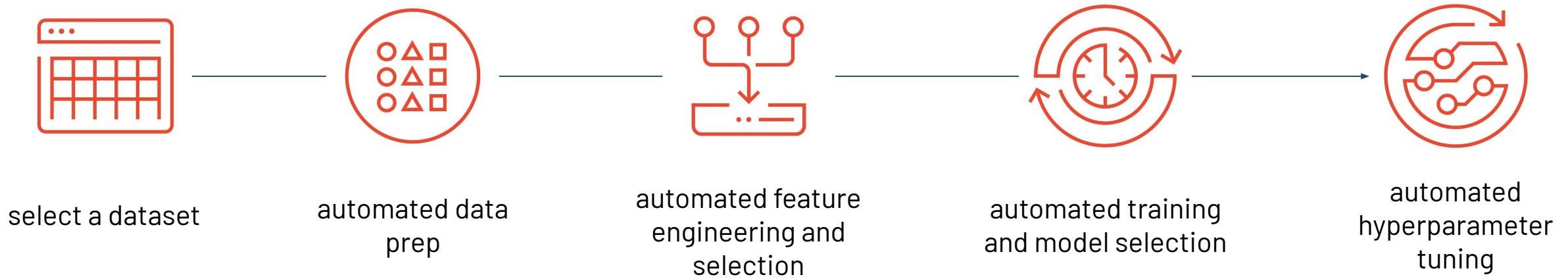
# AutoML - Glass box approach to Machine Learning



Open Data Lake Foundation with

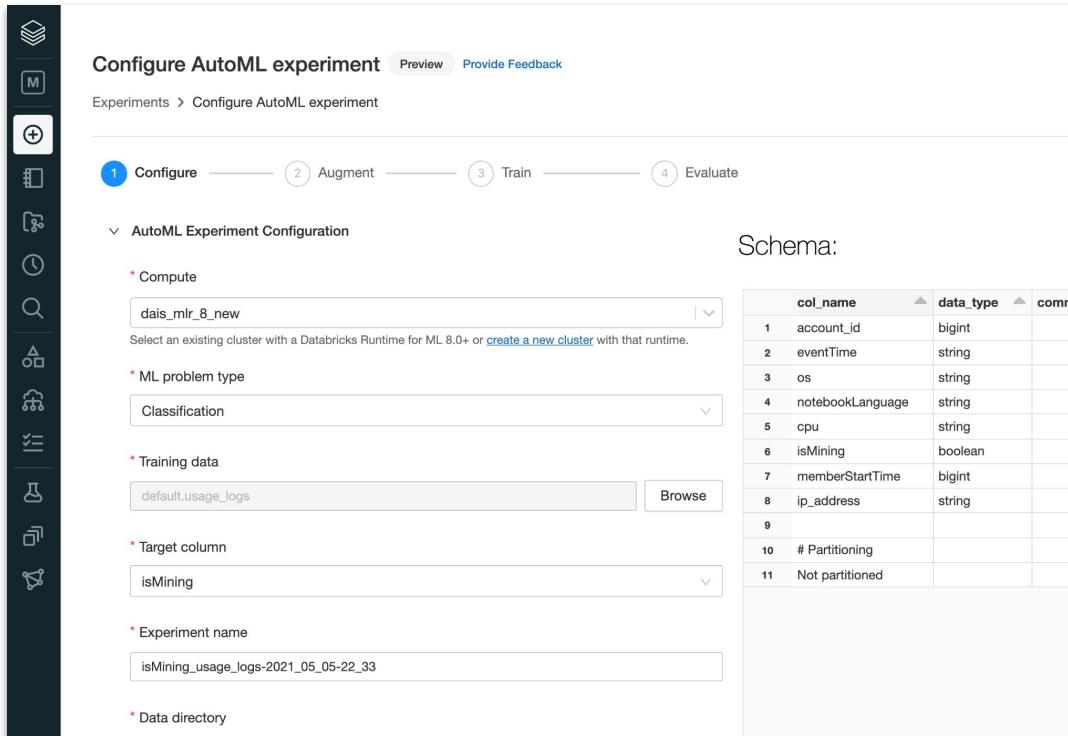


# Automating the ML workflow



# AutoML in the UI or using the API

## UI in Databricks Machine Learning



The screenshot shows the 'Configure AutoML experiment' page in the Databricks UI. The left sidebar contains various icons for data management, ML, and workspace. The main area has a header 'Configure AutoML experiment' with 'Preview' and 'Provide Feedback' buttons. Below the header, it says 'Experiments > Configure AutoML experiment'. A progress bar at the top indicates steps 1 through 4: 'Configure', 'Augment', 'Train', and 'Evaluate', with 'Configure' being the active step. The 'AutoML Experiment Configuration' section includes fields for 'Compute' (selected cluster 'dais\_mlr\_8\_new'), 'ML problem type' ('Classification'), 'Training data' ('default.usage\_logs'), 'Target column' ('isMining'), 'Experiment name' ('isMining\_usage\_logs-2021\_05\_05-22\_33'), and 'Data directory'. To the right, a 'Schema:' table lists the columns and their types from the selected dataset:

col_name	data_type	comment
1 account_id	bigint	
2 eventTime	string	
3 os	string	
4 notebookLanguage	string	
5 cpu	string	
6 isMining	boolean	
7 memberStartTime	bigint	
8 ip_address	string	
9		
10 # Partitioning		
11 Not partitioned		

```
databricks.automl.classify(df,  
    target_col='label',  
    timeout_minutes=60)
```

# AutoML output

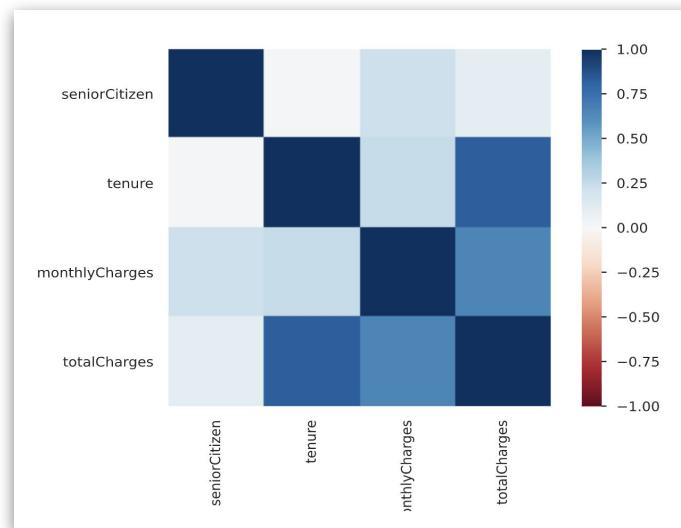
## MLflow experiment

Auto-created MLflow Experiment to track models and metrics

	Start Time	Run Name	User	Source
	2021-05-05 1	logistic_r...	kase...	<a href="#">Notebook</a>
	2021-05-05 1	logistic_r...	alkis...	<a href="#">21-05</a>
	2021-05-05 1	logistic_r...	alkis...	<a href="#">21-05</a>
	2021-05-05 1	logistic_r...	kase...	<a href="#">Notebook</a>
	2021-05-05 1	logistic_r...	kase...	<a href="#">Notebook</a>
	2021-05-05 1	logistic_r...	kase...	<a href="#">Notebook</a>
	2021-05-05 1	decision_...	kase...	<a href="#">Notebook</a>
	2021-05-05 1	random_f...	kase...	<a href="#">Notebook</a>

## Data exploration notebook

Generated notebook with feature summary statistics and distributions



## Reproducible trial notebooks

Generated notebooks with source code for every model

Generated Trial Notebook (Python)

dais\_mlr\_8\_new

Random Forest training

- Load Data
- Preprocessors
  - Numerical columns
    - One-hot encoding
    - Feature standardization
- Train classification mo...

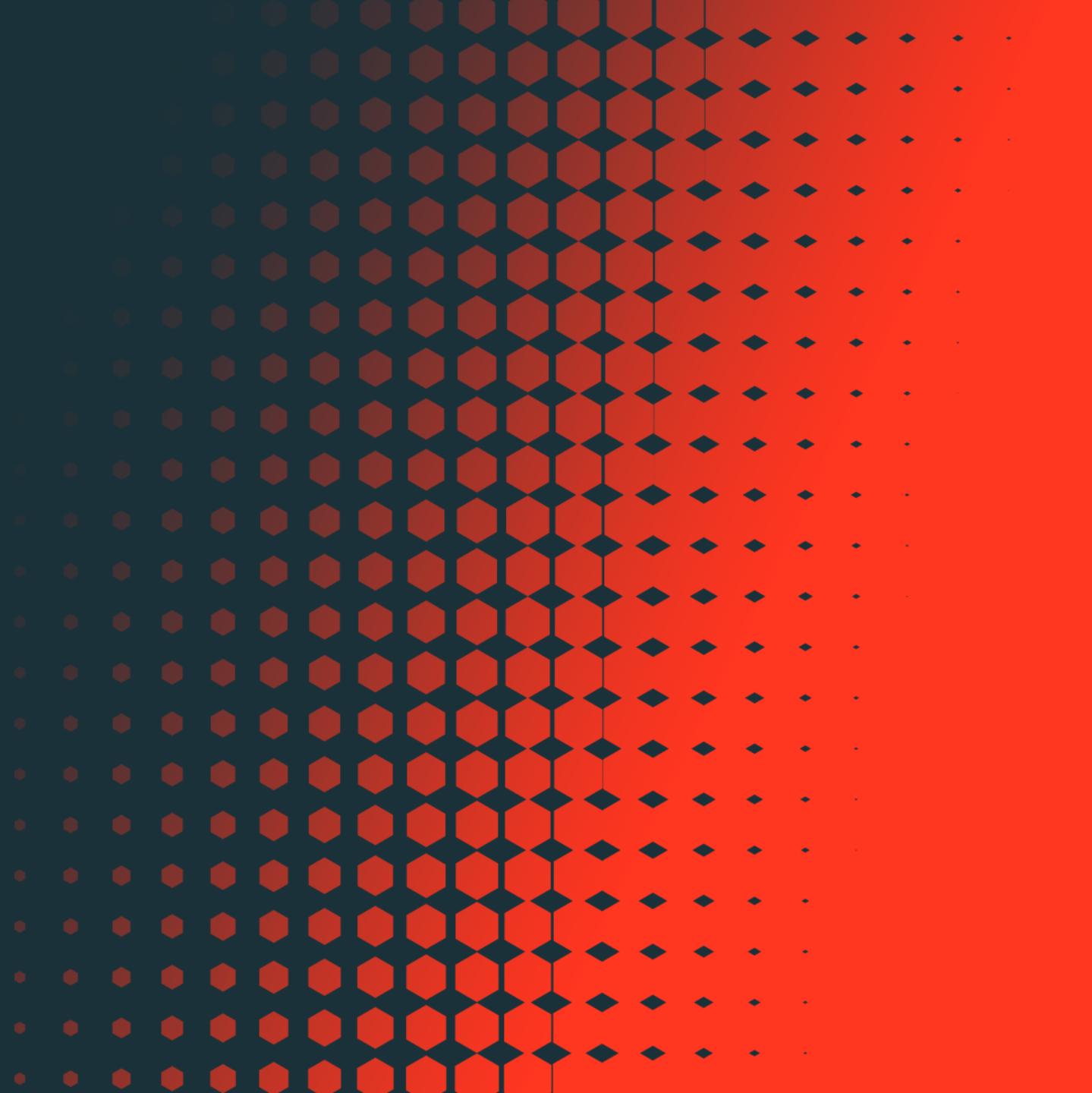
# Chosen results.  
# Use predi  
expla  
shap\_

6  
7  
8  
9  
10  
11  
12  
13  
14

# Cho  
resul  
examp  
# Use  
predi  
expla  
shap\_

# AutoML Features

Problem Types	Models / Tuning	Features	Tracking	Evaluation	Deployment
 Classification	 	 Numerical			 Batch Scoring
 Regression		 Categorical   Timestamp	 Metrics	 Parameters	 Artifacts   Models



Thank you