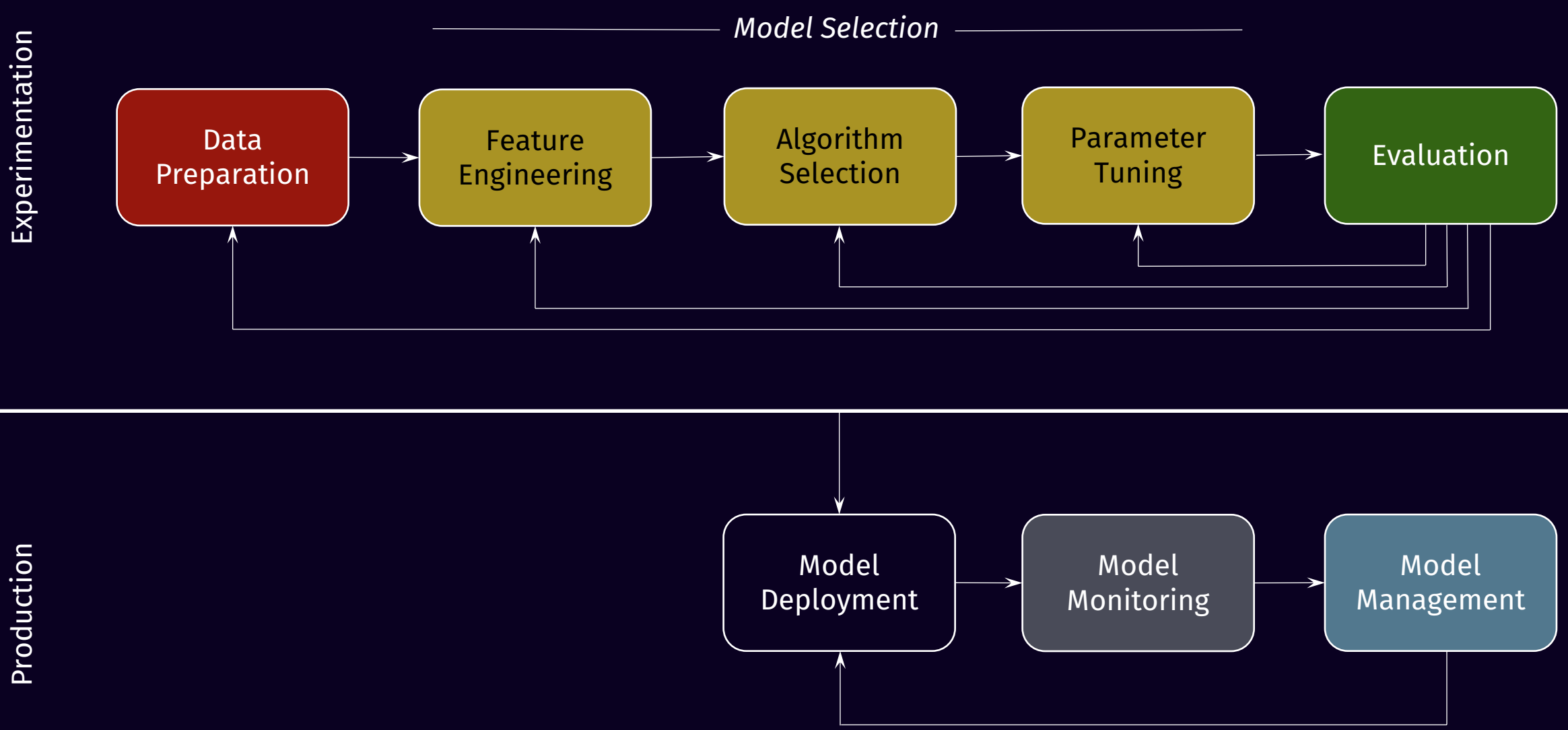
machine learning conference

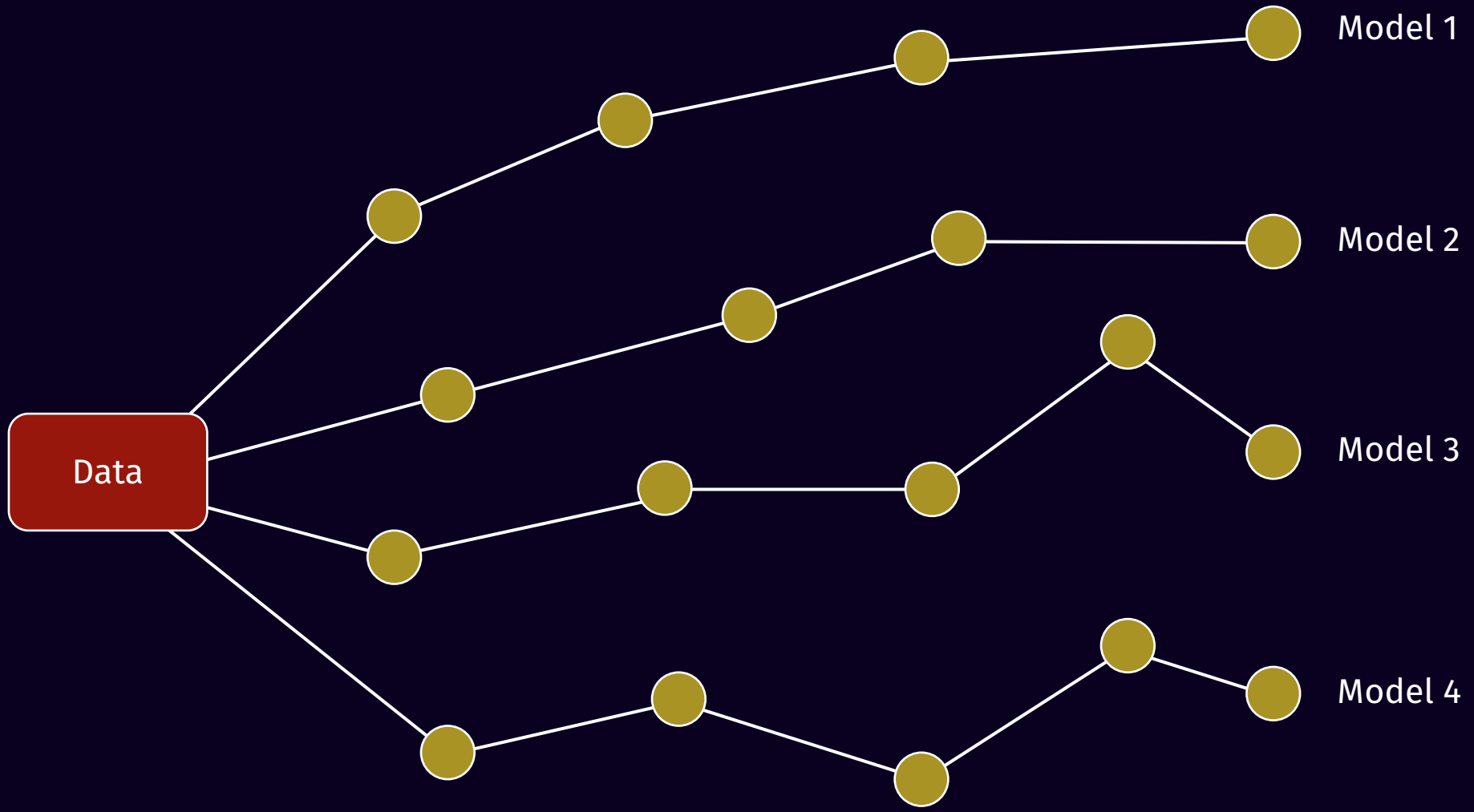
Towards Automated Tracking of Machine Learning Experiments

Joos-Hendrik Boese | Johannes Kirschnick | Thoralf Klein | **Stephan Seufert**

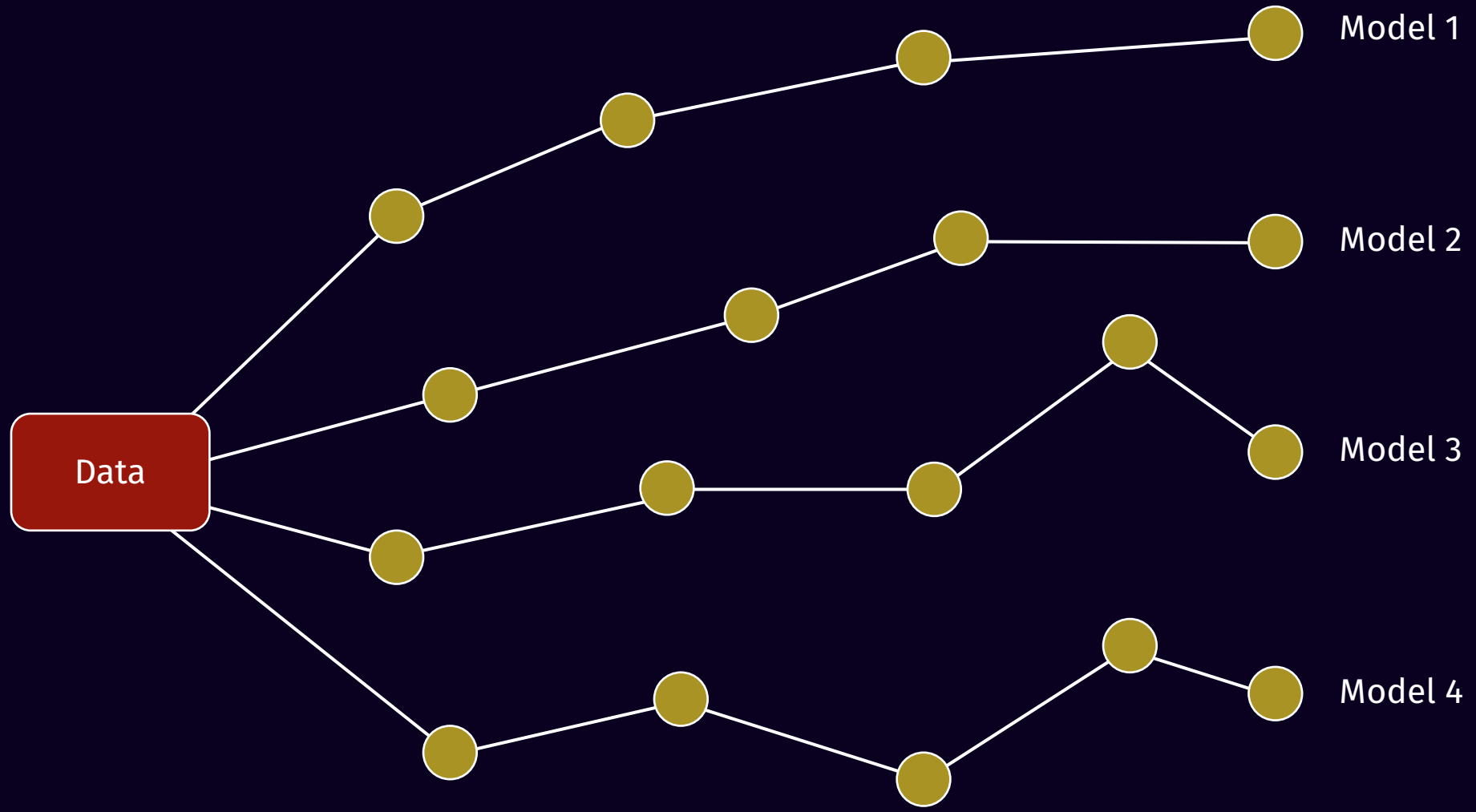
Ryck ExperimentTracker | Core AI



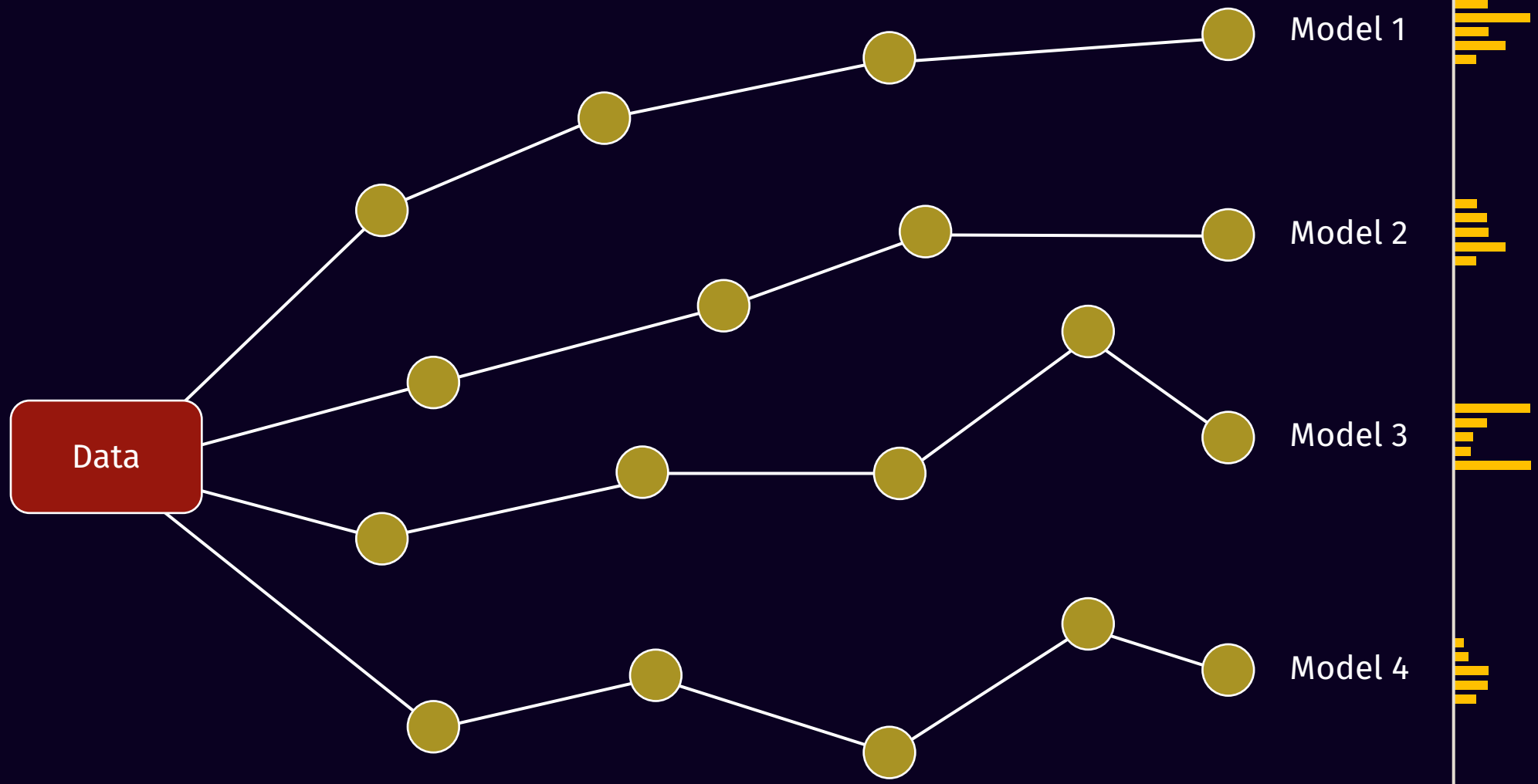
Experimentation



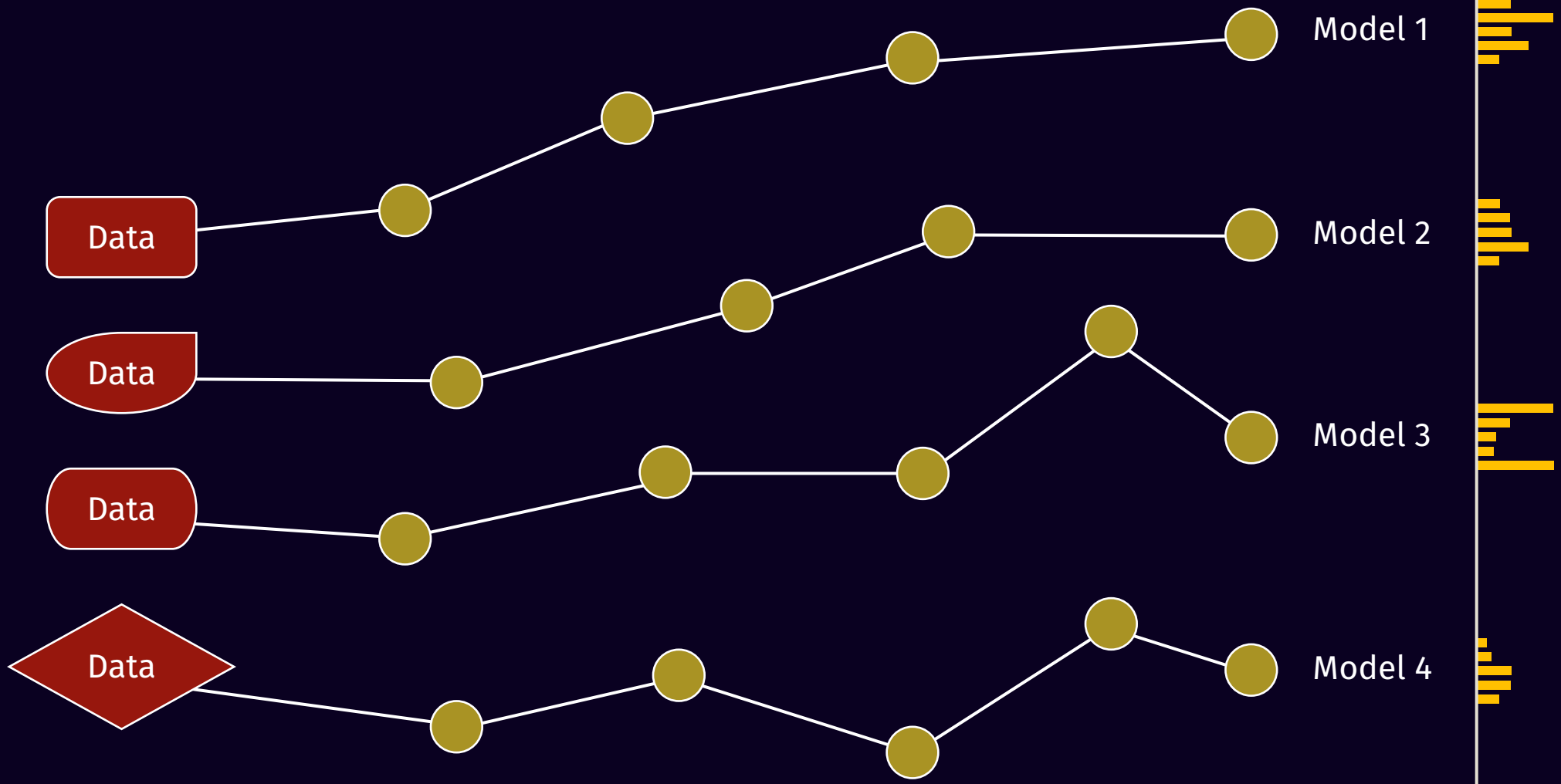
Experimentation



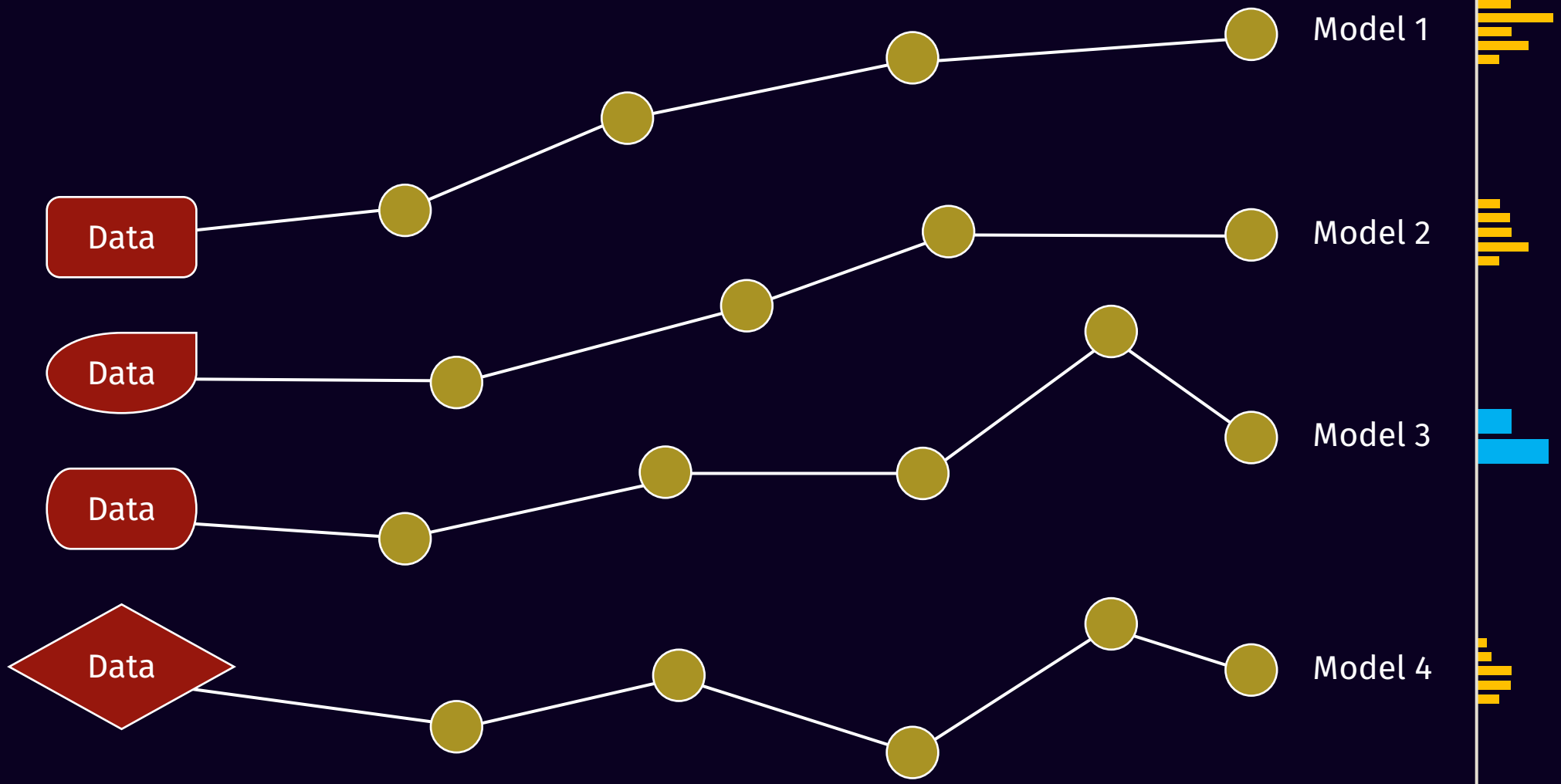
Experimentation



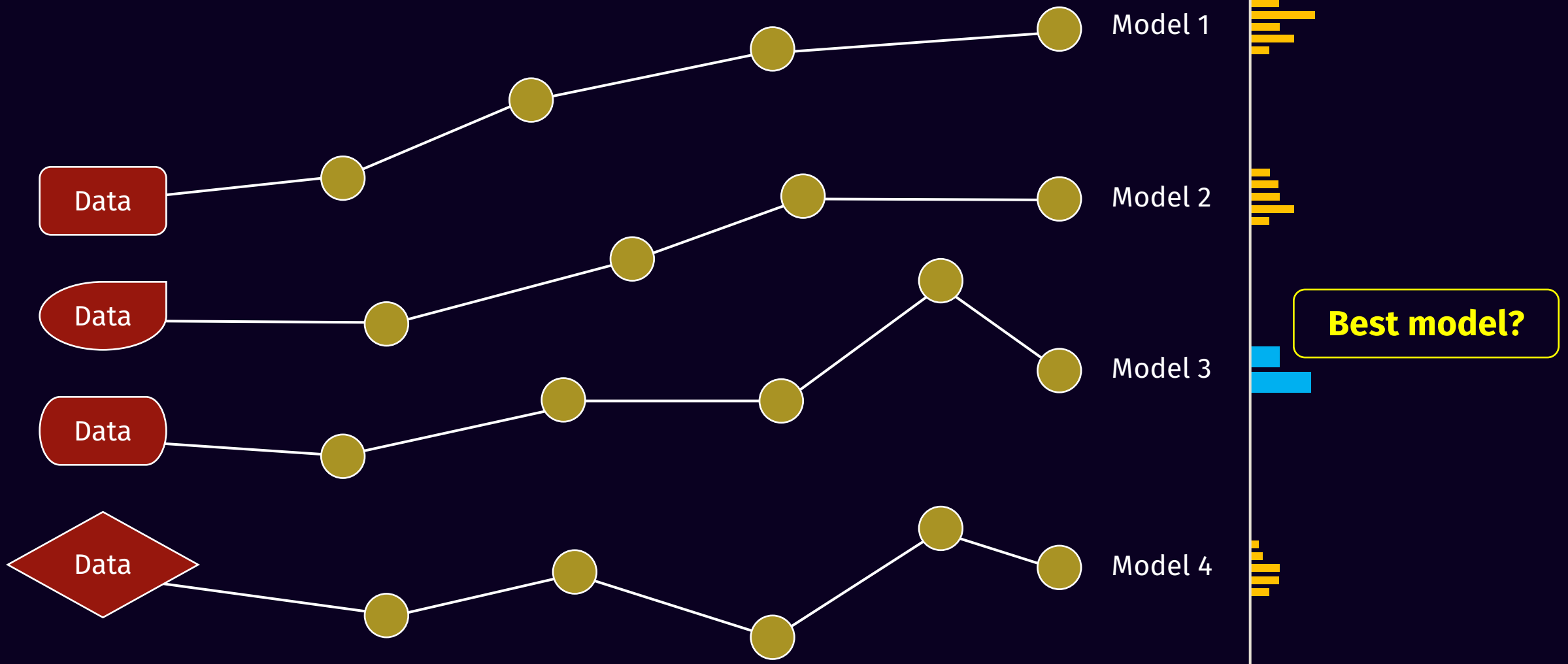
Experimentation



Experimentation

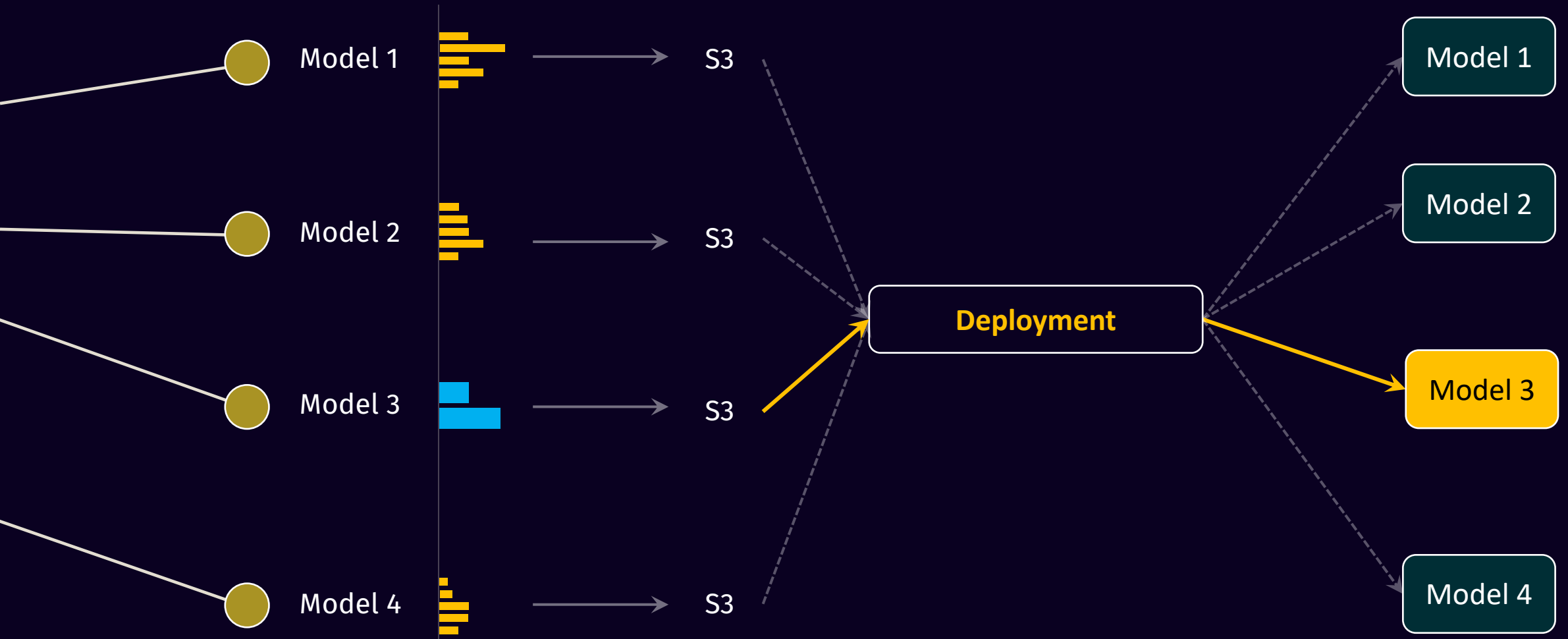


Experimentation



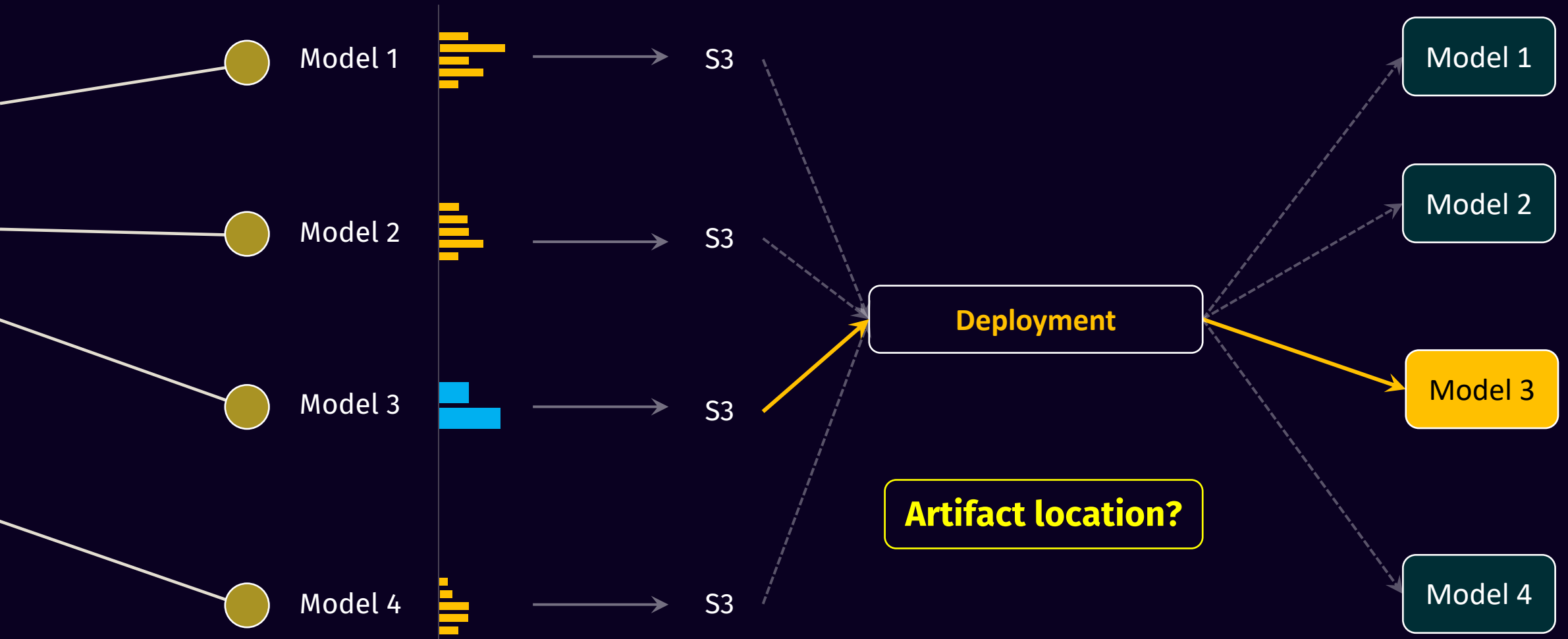
Experimentation

Production

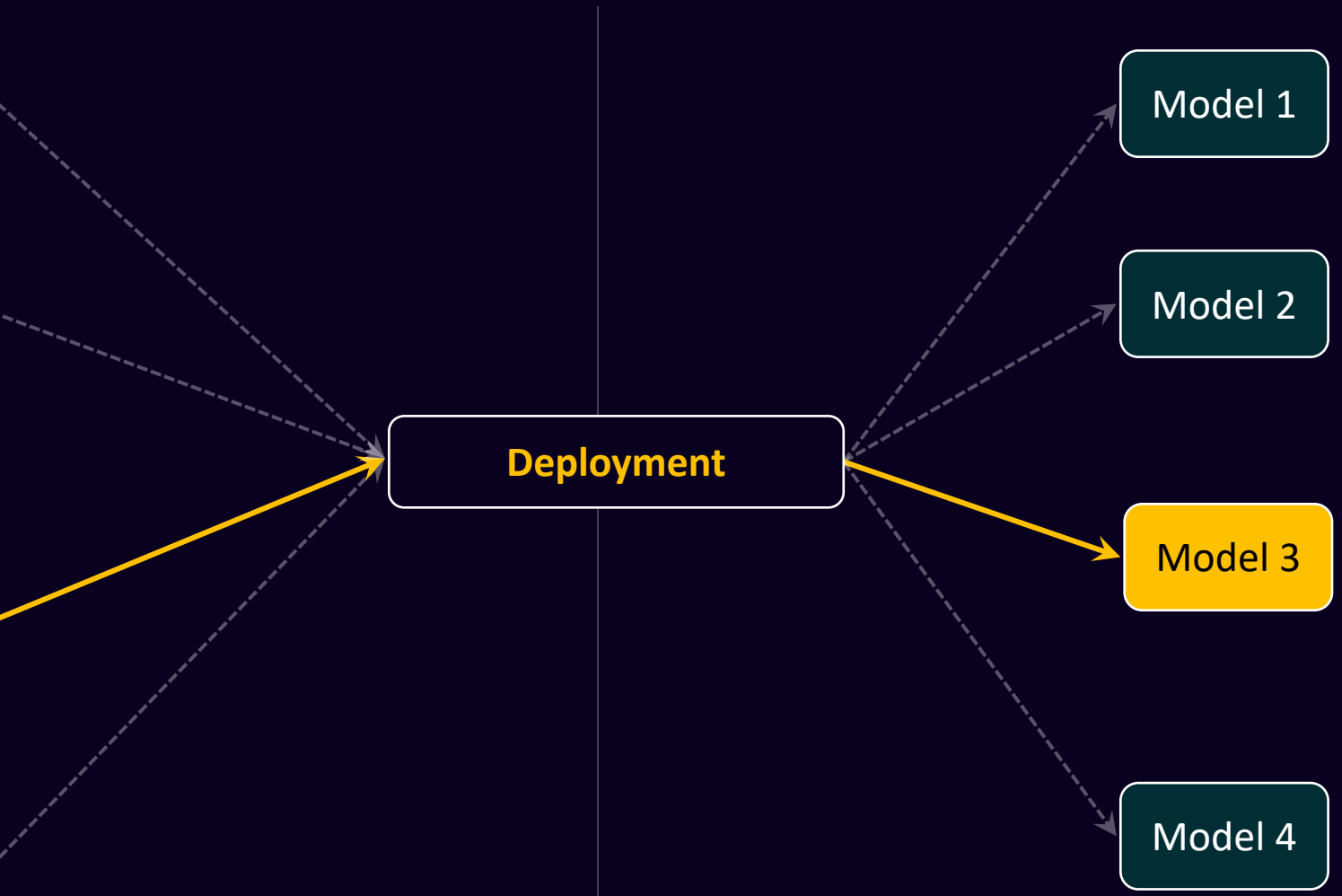


Experimentation

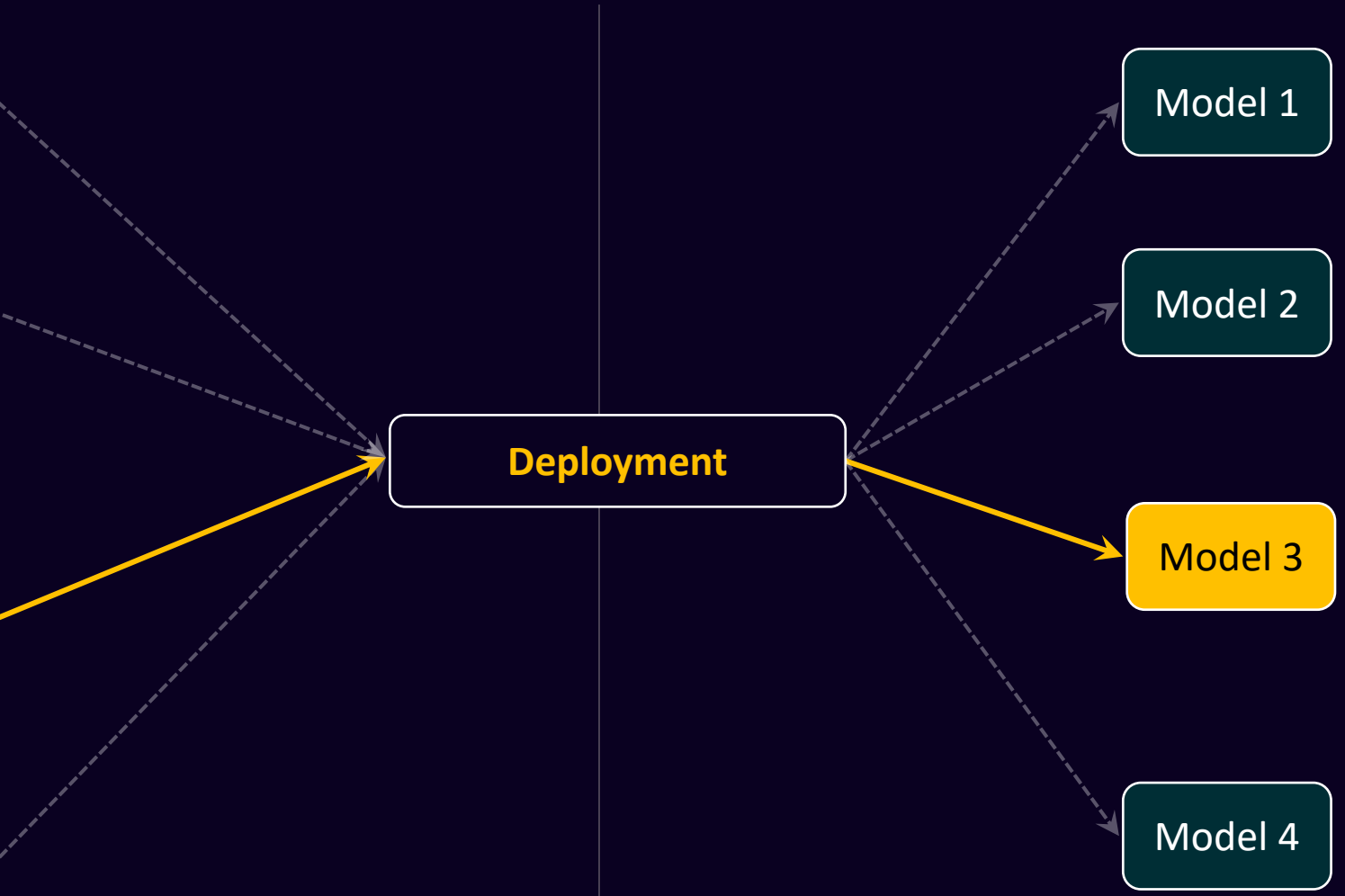
Production



Production

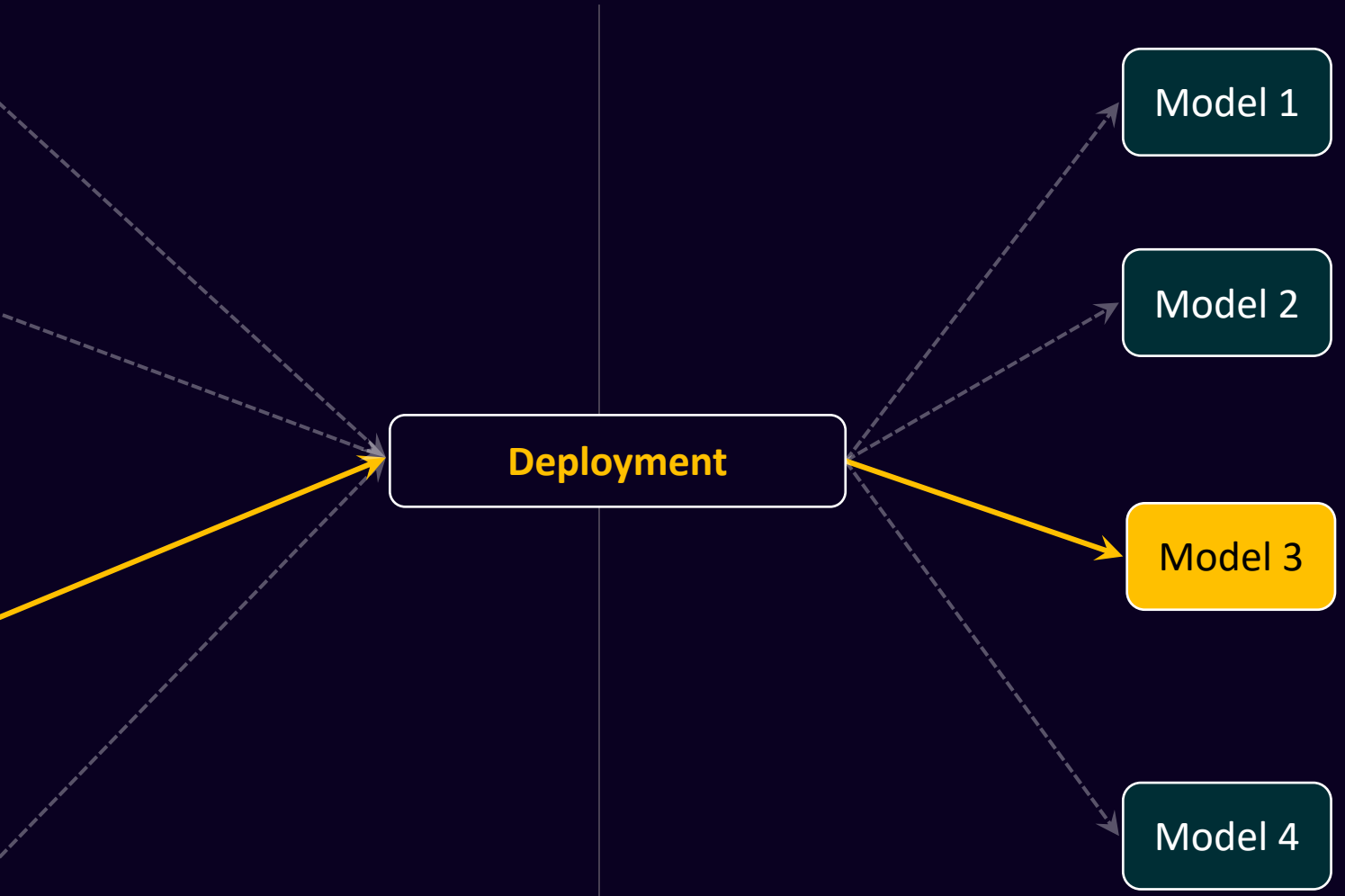


Production



Which model is currently deployed?

Production

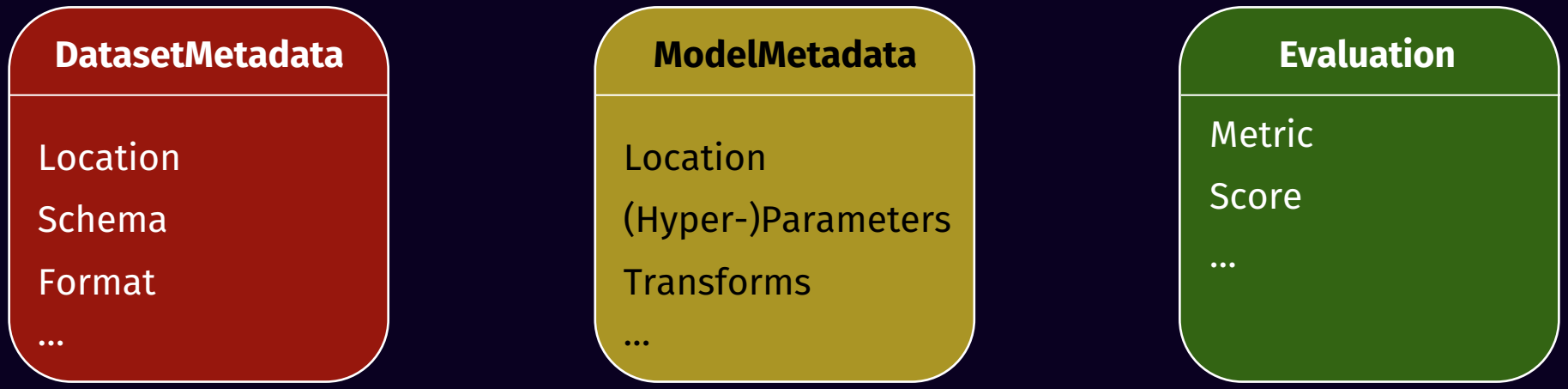


Which model is currently deployed?

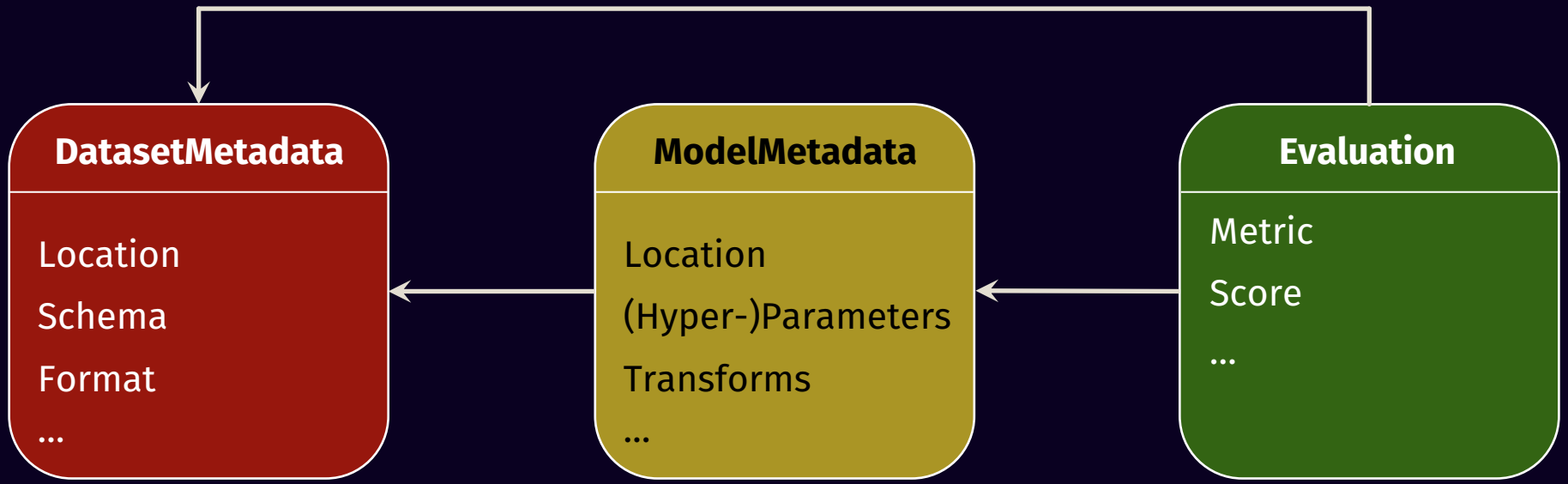
Which model was deployed at time X?

ExperimentTracker

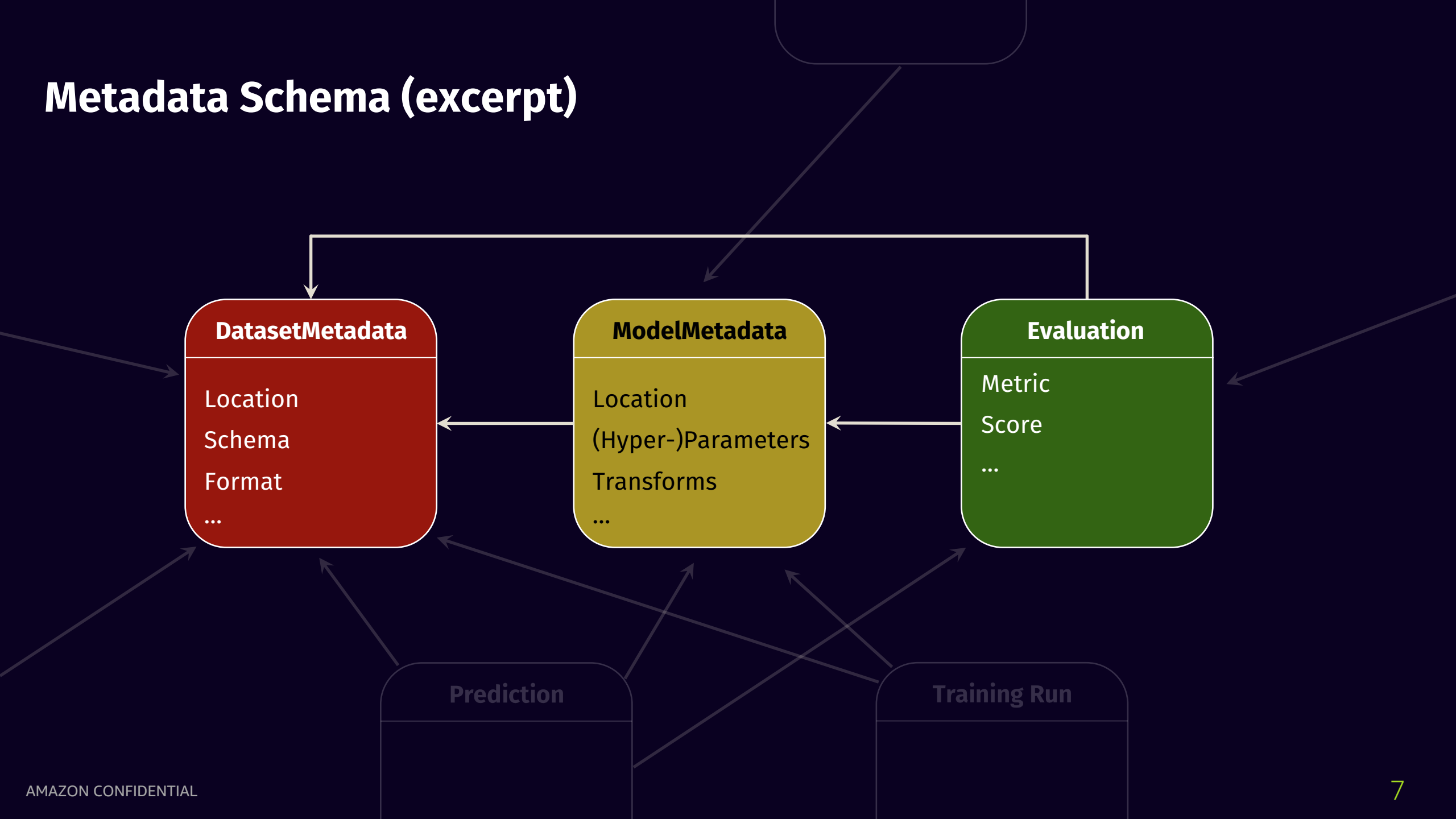
Metadata Schema (excerpt)



Metadata Schema (excerpt)

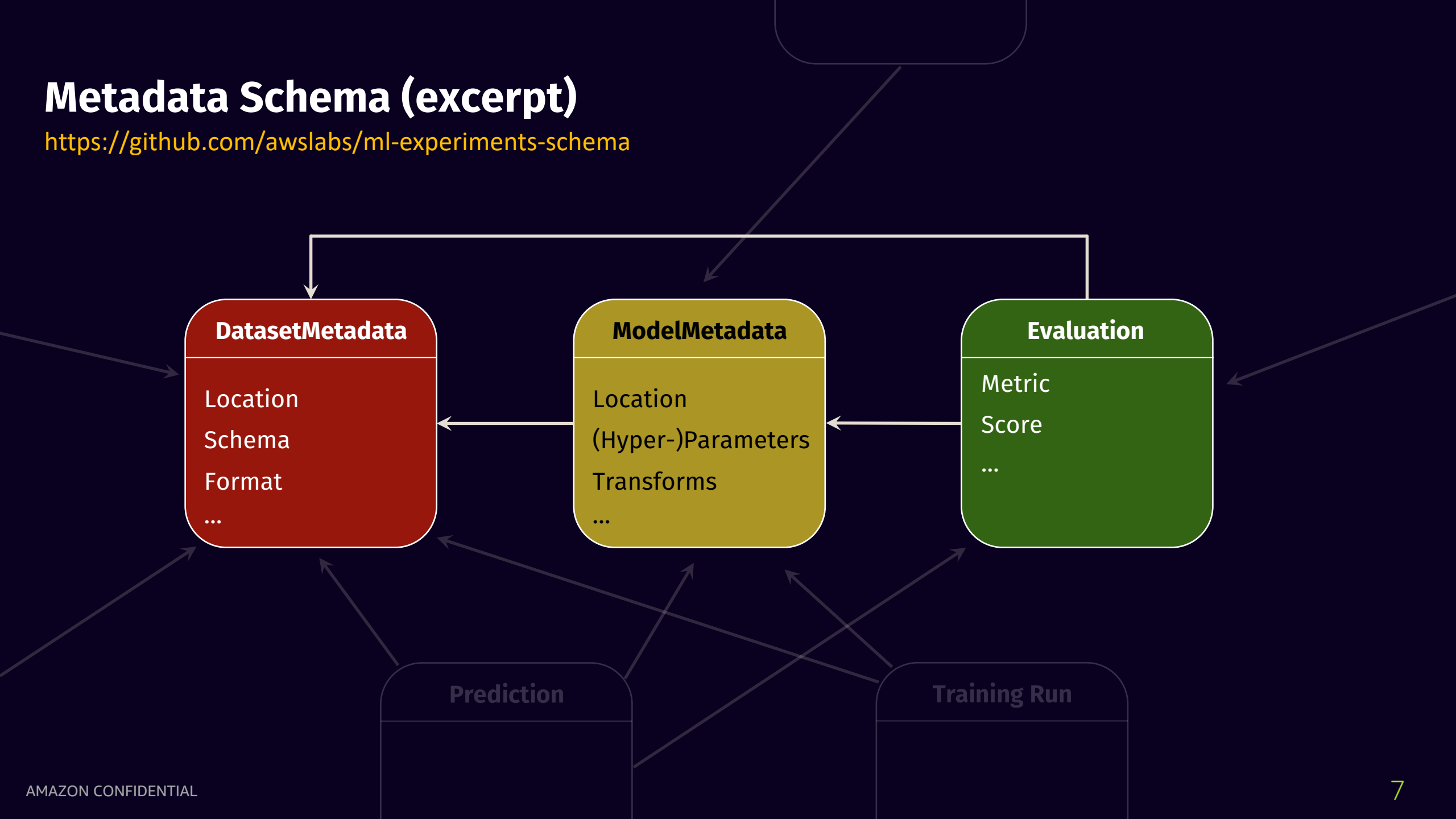


Metadata Schema (excerpt)



Metadata Schema (excerpt)

https://github.com/awslabs/ml-experiments-schema

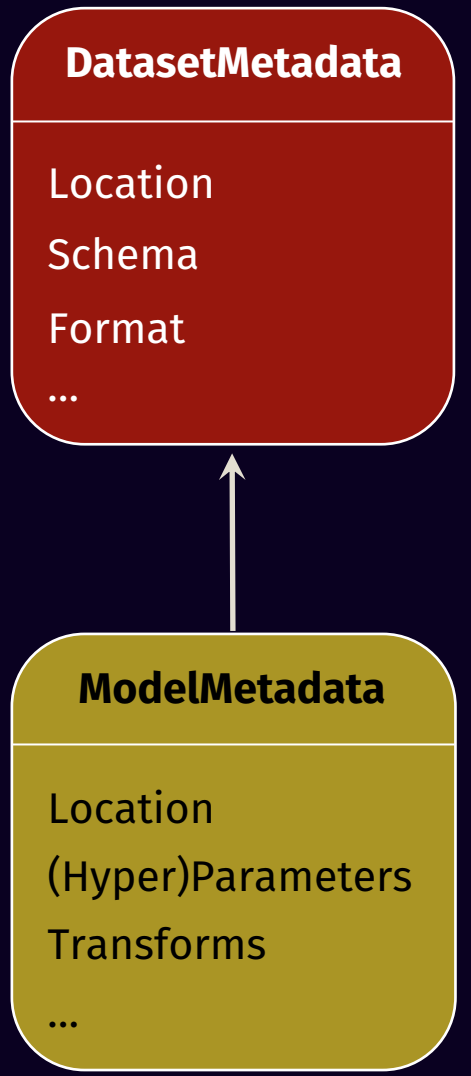


Automated Metadata Extraction

```
val df: DataFrame = …



val pipeline: Pipeline = …

val model: PipelineModel = pipeline.fit(df)
```

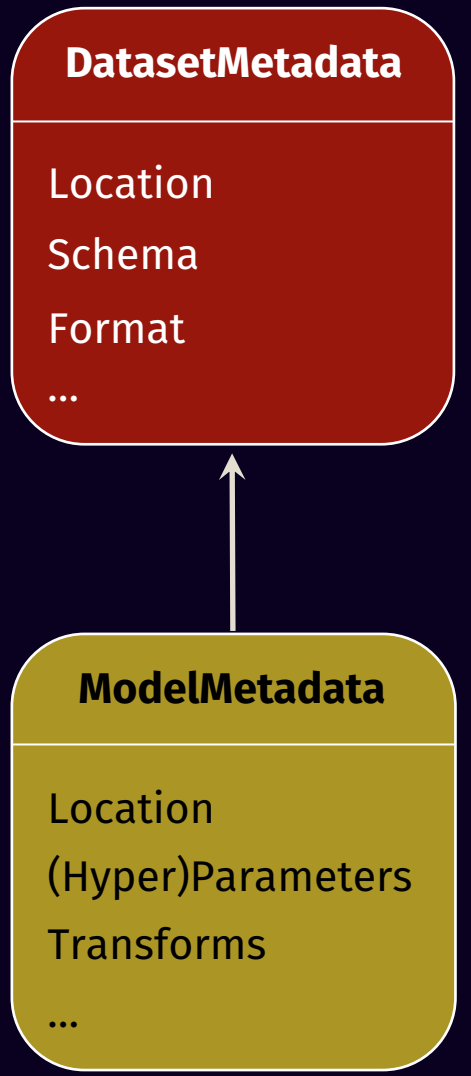
Automated Metadata Extraction

```
val df: DataFrame = …


val pipeline: Pipeline = …

val model: PipelineModel = pipeline.fit(df)
```



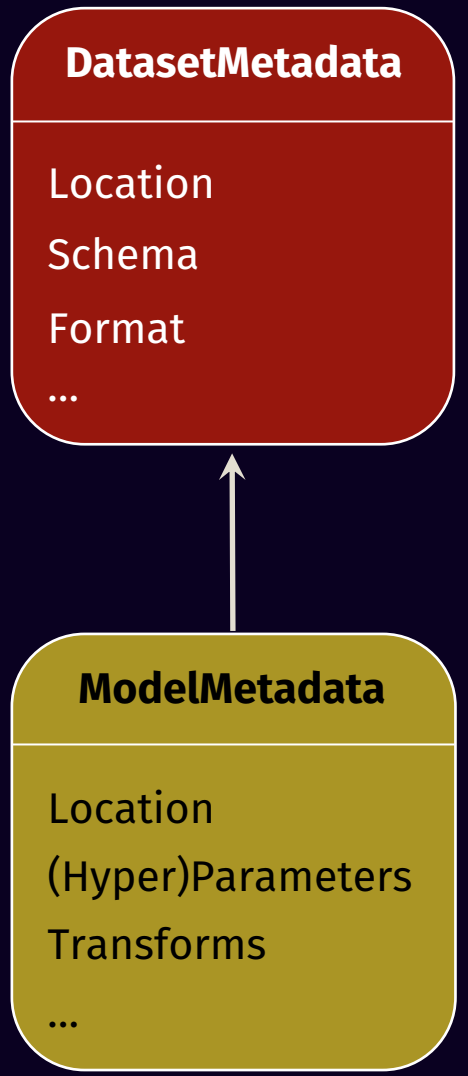
Automated Metadata Extraction

```
val df: DataFrame = …

tracker.trackDataset(df)


val pipeline: Pipeline = …

val model: PipelineModel = pipeline.fit(df)

tracker.trackModel(model)
```

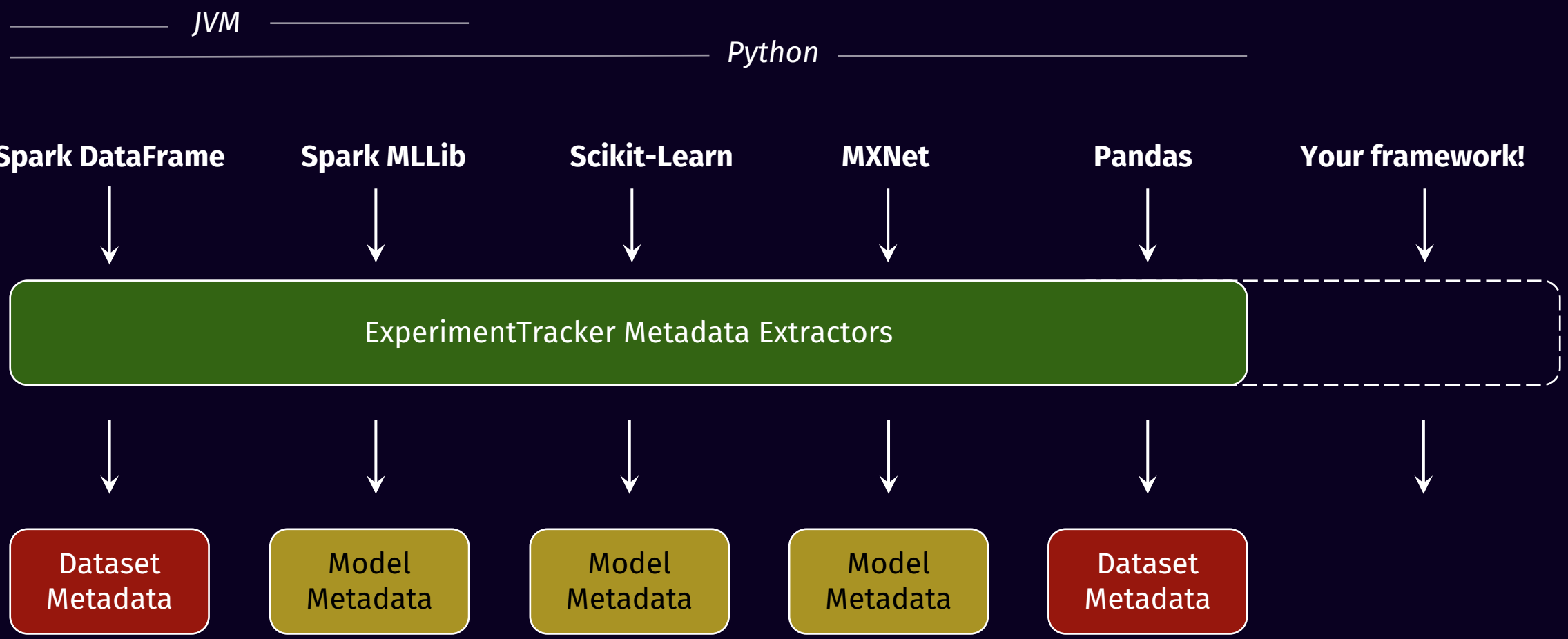


Automated Metadata Extraction

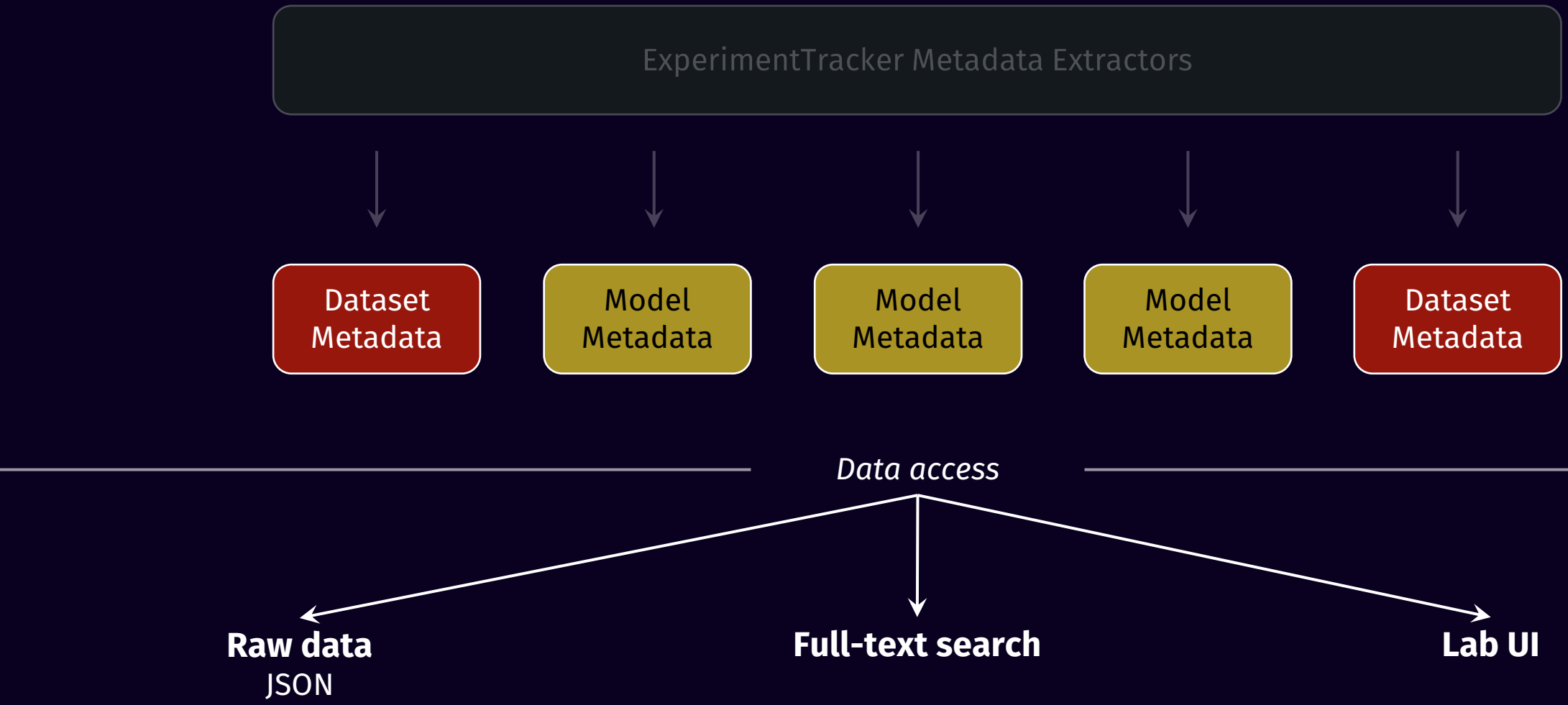
```
val df: DataFrame = …

tracker.trackDataset(df)


val pipeline: Pipeline = …

val model: PipelineModel = pipeline.fit(df)

tracker.trackModel(model)
```



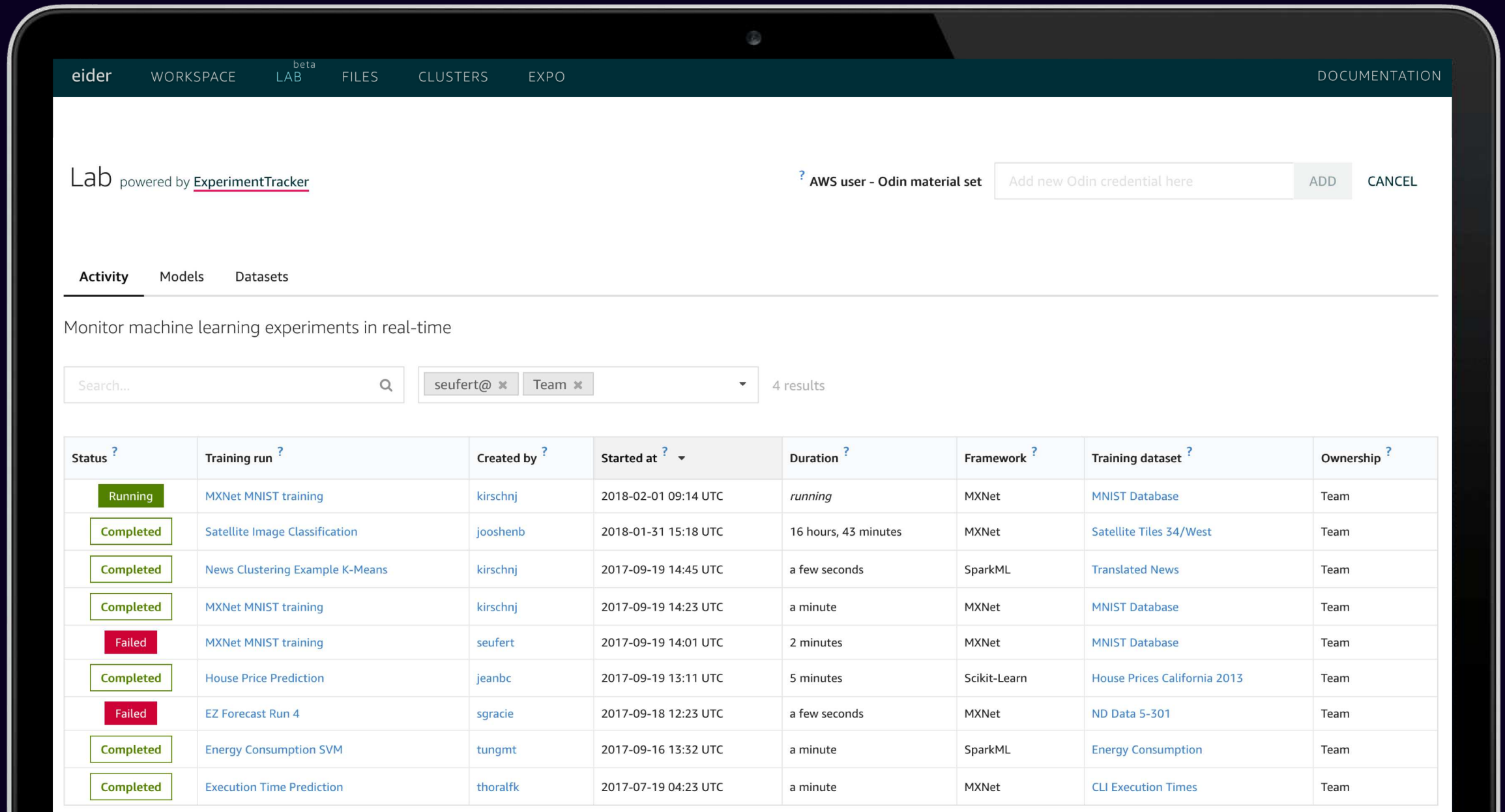
Automated Metadata Extraction



Data Access


ExperimentTracker Metadata Extractors
Dataset Metadata
Model Metadata
Model Metadata
Model Metadata
Dataset Metadata
Data access
Raw data
JSON
Full-text search
Lab UI

Browse experiments



Monitor model training

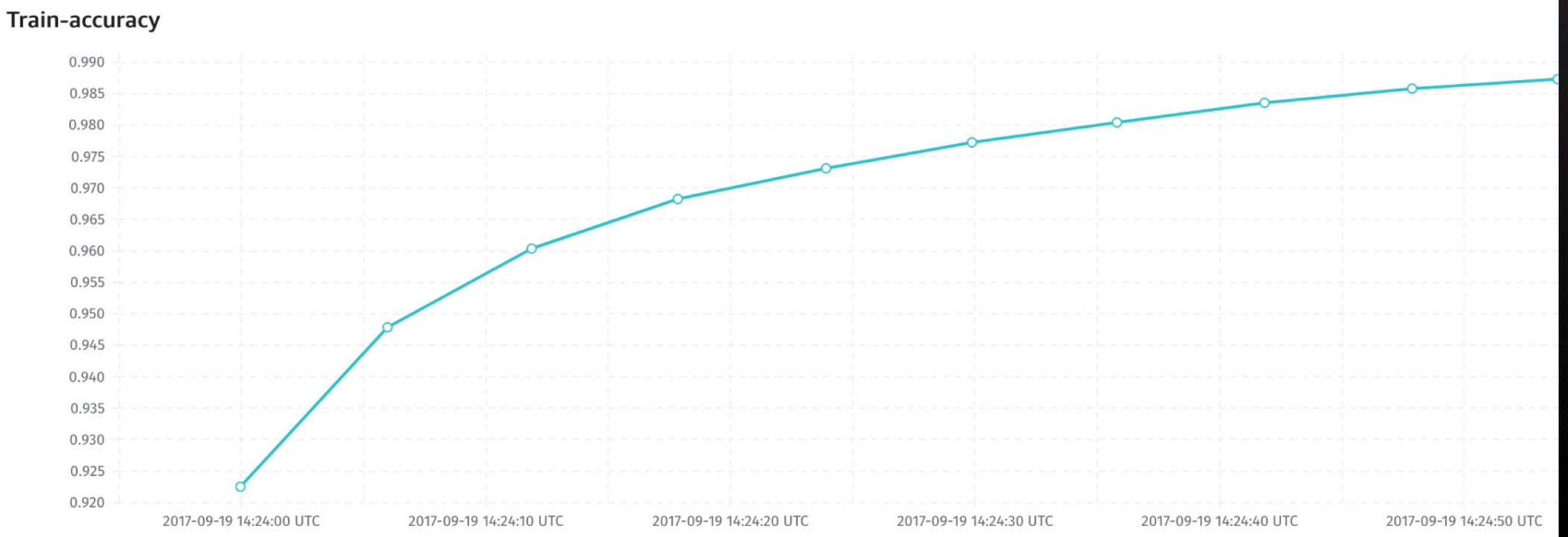
eider WORKSPACE LAB beta FILES CLUSTERS EXPO DOCUMENTATION

BACK

Training run: MXNet MNIST training

Training Run Info | Training Progress | Source

Train Accuracy



Epoch	Timestamp	Value
0	2017-09-19 14:23:59:993 UTC	0.92241

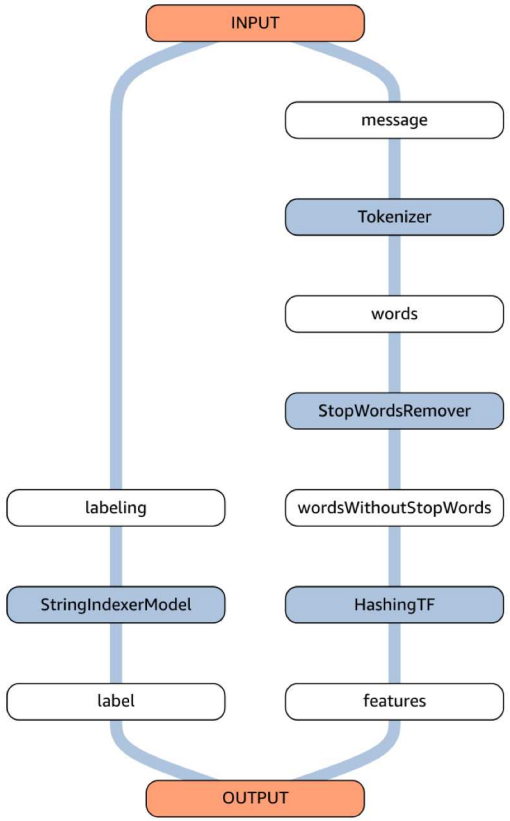
Inspect artifacts

eider WORKSPACE LAB beta FILES CLUSTERS EXPO DOCUMENTATION

BACK

Model: SMS Spam Classification (Logistic Regression)

Model Info | Hyperparameters | Pipeline | Evaluations | Source



Transform details

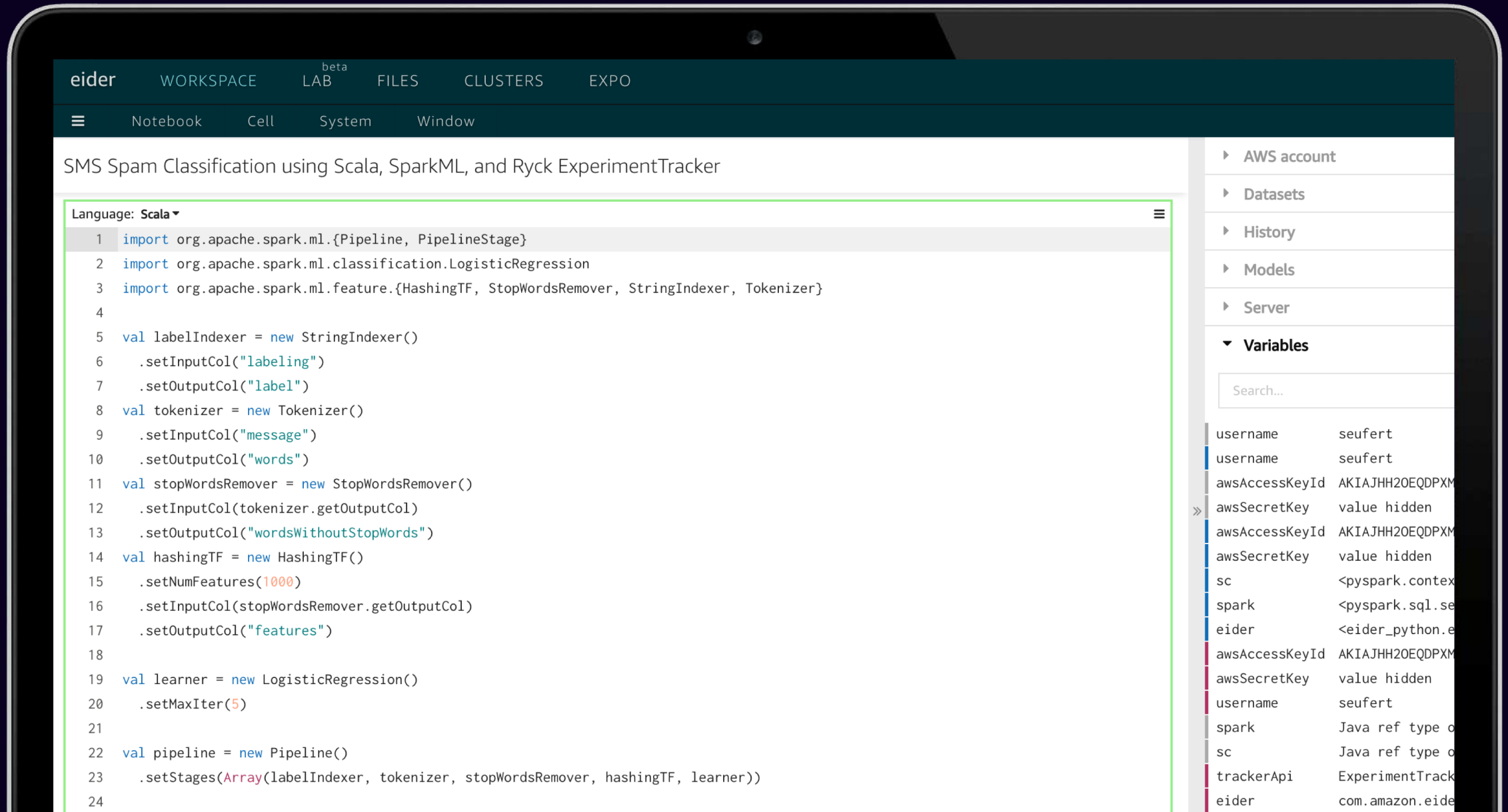
Id	89a07589-5cc6-4022-a109-3824e0100124
Name	HashingTF
Framework	

Transform parameters

Parameter	Value
numFeatures	1000
binary	false

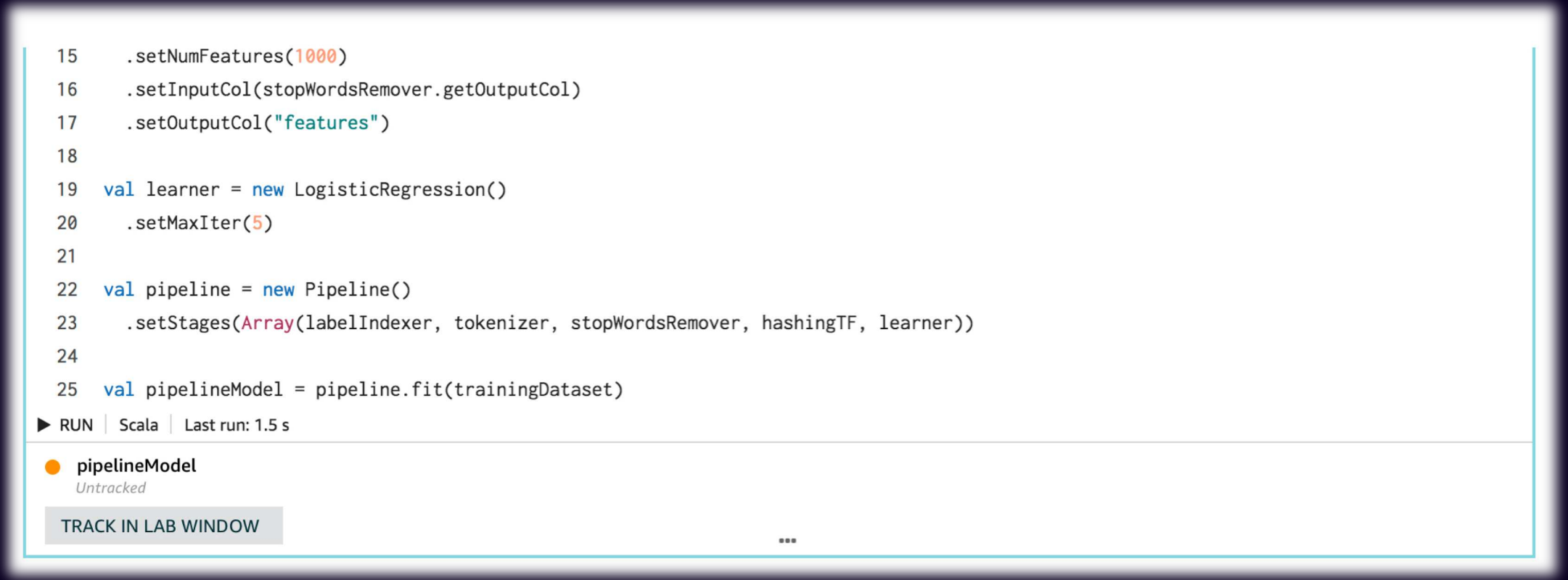
1. Interactive Tracking

Eider Workspace Integration



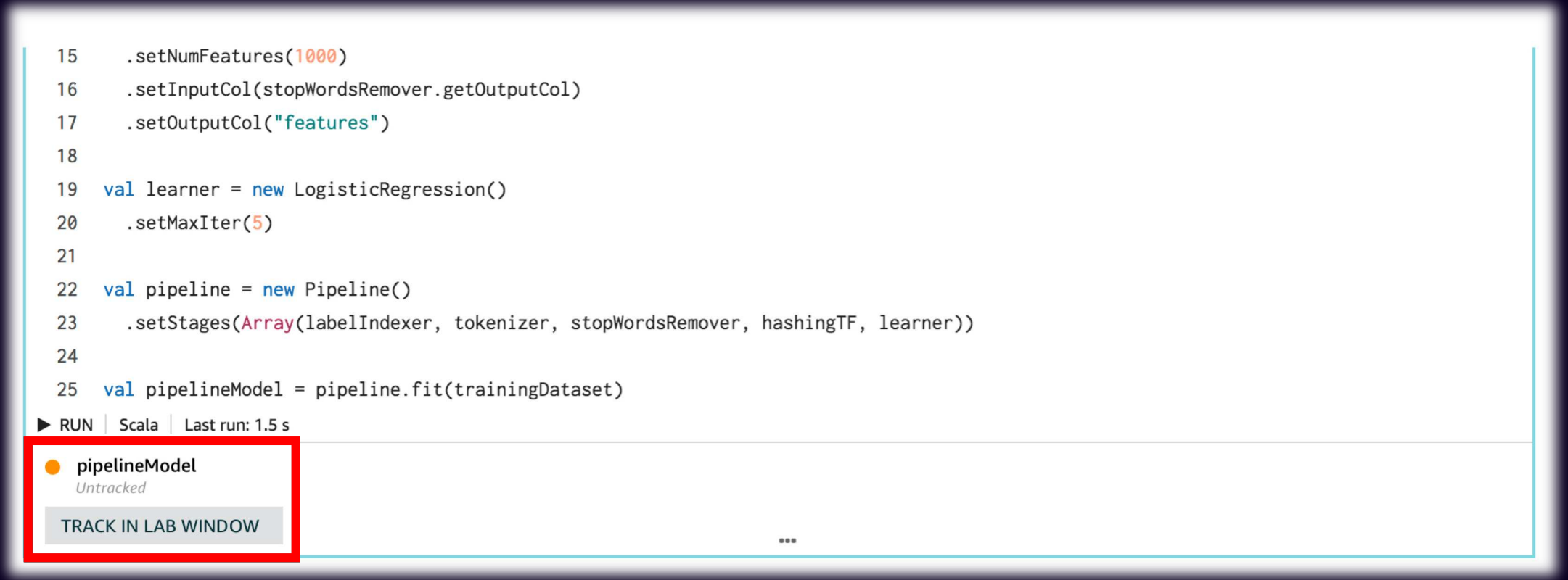
Eider: Interactive Tracking

New design – coming soon



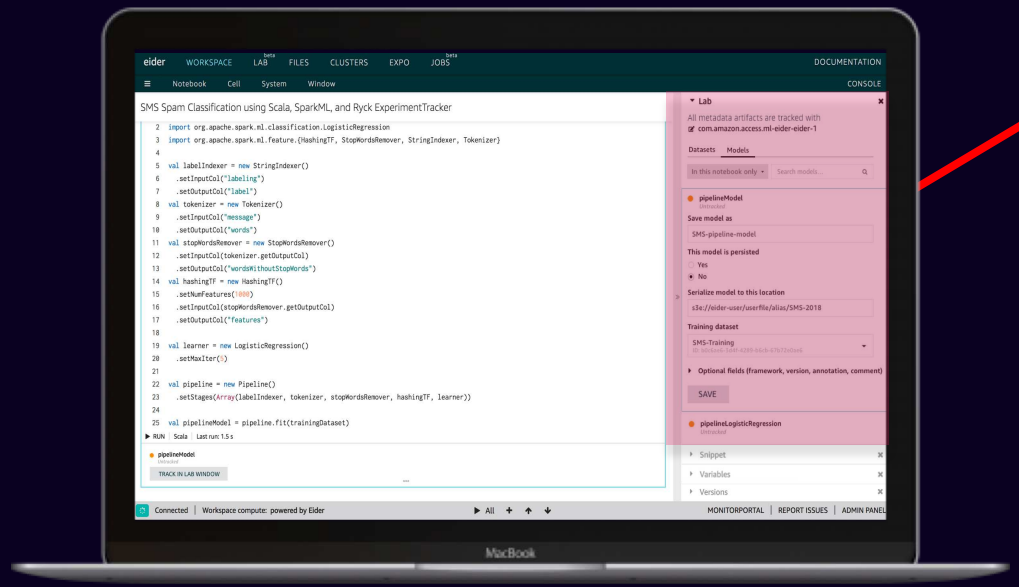
Eider: Interactive Tracking

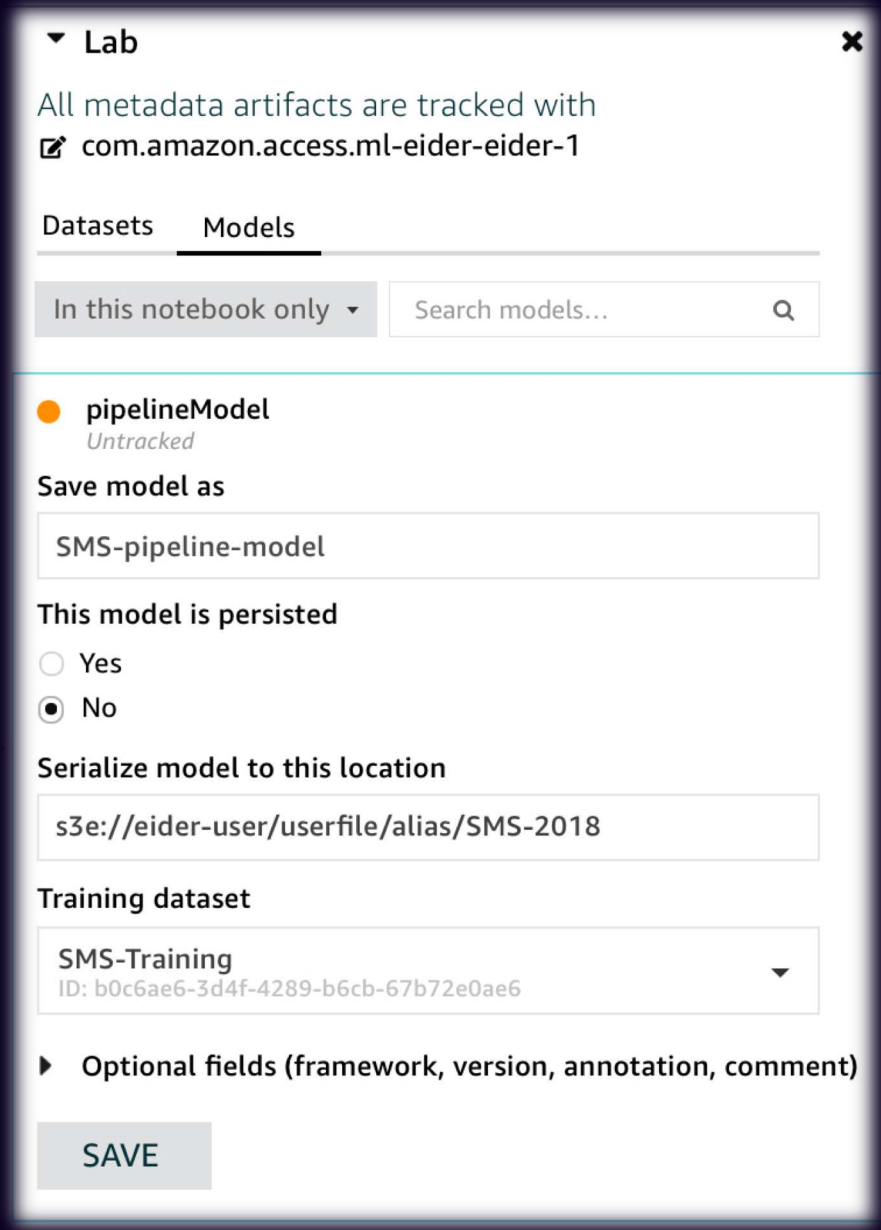
New design – coming soon



Eider: Interactive Tracking

New design – coming soon





2. Log Analysis

Logexp: Non-Intrusive Experiment Tracking

```
./run-mxnet-experiment.py
```

Logexp: Non-Intrusive Experiment Tracking

```
./run-mxnet-experiment.py

INFO:root:Epoch[0] Batch [100] Speed: 83831.42 samples/sec  accuracy=0.10432
INFO:root:Epoch[0] Batch [200] Speed: 59385.85 samples/sec  accuracy=0.11060
INFO:root:Epoch[0] Batch [300] Speed: 62468.41 samples/sec  accuracy=0.11310
INFO:root:Epoch[0] Batch [400] Speed: 68188.53 samples/sec  accuracy=0.11480
INFO:root:Epoch[0] Batch [500] Speed: 40556.61 samples/sec  accuracy=0.12300
INFO:root:Epoch[0] Train-accuracy=0.300101
INFO:root:Epoch[0] Time cost=1.210
INFO:root:Epoch[0] Validation-accuracy=0.457100
INFO:root:Epoch[1] Batch [100] Speed: 38430.77 samples/sec  accuracy=0.547426
. . .
```

Logexp: Non-Intrusive Experiment Tracking

```
> pip install logexp
```

```
./run-mxnet-experiment.py

INFO:root:Epoch[0] Batch [100] Speed: 83831.42 samples/sec  accuracy=0.10432
INFO:root:Epoch[0] Batch [200] Speed: 59385.85 samples/sec  accuracy=0.11060
INFO:root:Epoch[0] Batch [300] Speed: 62468.41 samples/sec  accuracy=0.11310
INFO:root:Epoch[0] Batch [400] Speed: 68188.53 samples/sec  accuracy=0.11480
INFO:root:Epoch[0] Batch [500] Speed: 40556.61 samples/sec  accuracy=0.12300
INFO:root:Epoch[0] Train-accuracy=0.300101
INFO:root:Epoch[0] Time cost=1.210
INFO:root:Epoch[0] Validation-accuracy=0.457100
INFO:root:Epoch[1] Batch [100] Speed: 38430.77 samples/sec  accuracy=0.547426
. . .
```

Logexp: Non-Intrusive Experiment Tracking

```
> pip install logexp

> logexp-run ./run-mxnet-experiment.py

                INFO:root:Epoch[0] Batch [100] Speed: 83831.42 samples/sec  accuracy=0.10432
                INFO:root:Epoch[0] Batch [200] Speed: 59385.85 samples/sec  accuracy=0.11060
                INFO:root:Epoch[0] Batch [300] Speed: 62468.41 samples/sec  accuracy=0.11310
                INFO:root:Epoch[0] Batch [400] Speed: 68188.53 samples/sec  accuracy=0.11480
                INFO:root:Epoch[0] Batch [500] Speed: 40556.61 samples/sec  accuracy=0.12300
                INFO:root:Epoch[0] Train-accuracy=0.300101
                INFO:root:Epoch[0] Time cost=1.210
                INFO:root:Epoch[0] Validation-accuracy=0.457100
                INFO:root:Epoch[1] Batch [100] Speed: 38430.77 samples/sec  accuracy=0.547426
                . . .
```

Logexp: Non-Intrusive Experiment Tracking

```
> pip install logexp

> logexp-run ./run-mxnet-experiment.py

INFO:root:Epoch[0] Batch [100] Speed: 83831.42 samples/sec  accuracy=0.10432
INFO:root:Epoch[0] Batch [200] Speed: 59385.85 samples/sec  accuracy=0.11060
INFO:root:Epoch[0] Batch [300] Speed: 62468.41 samples/sec  accuracy=0.11310
INFO:root:Epoch[0] Batch [400] Speed: 68188.53 samples/sec  accuracy=0.11480
INFO:root:Epoch[0] Batch [500] Speed: 40556.61 samples/sec  accuracy=0.12300
INFO:root:Epoch[0] Train-accuracy=0.300101
INFO:root:Epoch[0] Time cost=1.210
INFO:root:Epoch[0] Validation-accuracy=0.457100
INFO:root:Epoch[1] Batch [100] Speed: 38430.77 samples/sec  accuracy=0.547426
. . .
```

Information Extraction

Logexp: Non-Intrusive Experiment Tracking

```
> pip install logexp
> logexp-run ./run-mxnet-experiment.py
```

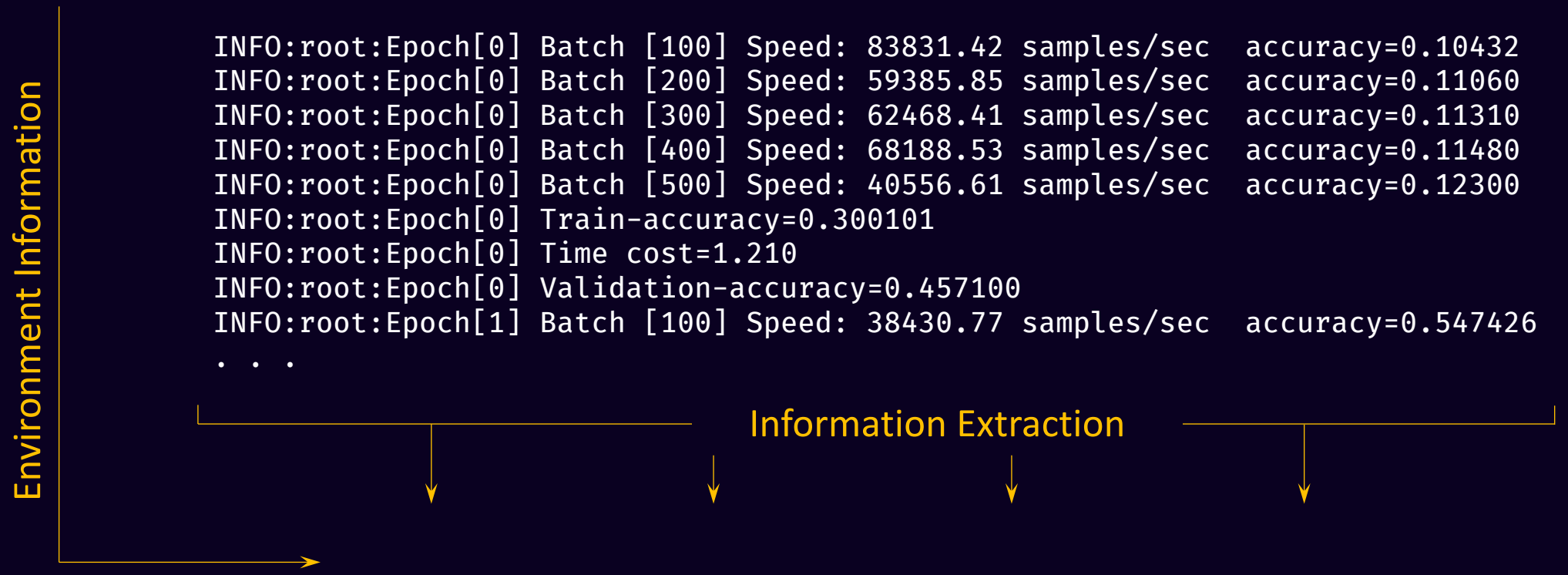
Environment Information

```
INFO:root:Epoch[0] Batch [100] Speed: 83831.42 samples/sec  accuracy=0.10432
INFO:root:Epoch[0] Batch [200] Speed: 59385.85 samples/sec  accuracy=0.11060
INFO:root:Epoch[0] Batch [300] Speed: 62468.41 samples/sec  accuracy=0.11310
INFO:root:Epoch[0] Batch [400] Speed: 68188.53 samples/sec  accuracy=0.11480
INFO:root:Epoch[0] Batch [500] Speed: 40556.61 samples/sec  accuracy=0.12300
INFO:root:Epoch[0] Train-accuracy=0.300101
INFO:root:Epoch[0] Time cost=1.210
INFO:root:Epoch[0] Validation-accuracy=0.457100
INFO:root:Epoch[1] Batch [100] Speed: 38430.77 samples/sec  accuracy=0.547426
. . .
```

Information Extraction

Logexp: Non-Intrusive Experiment Tracking

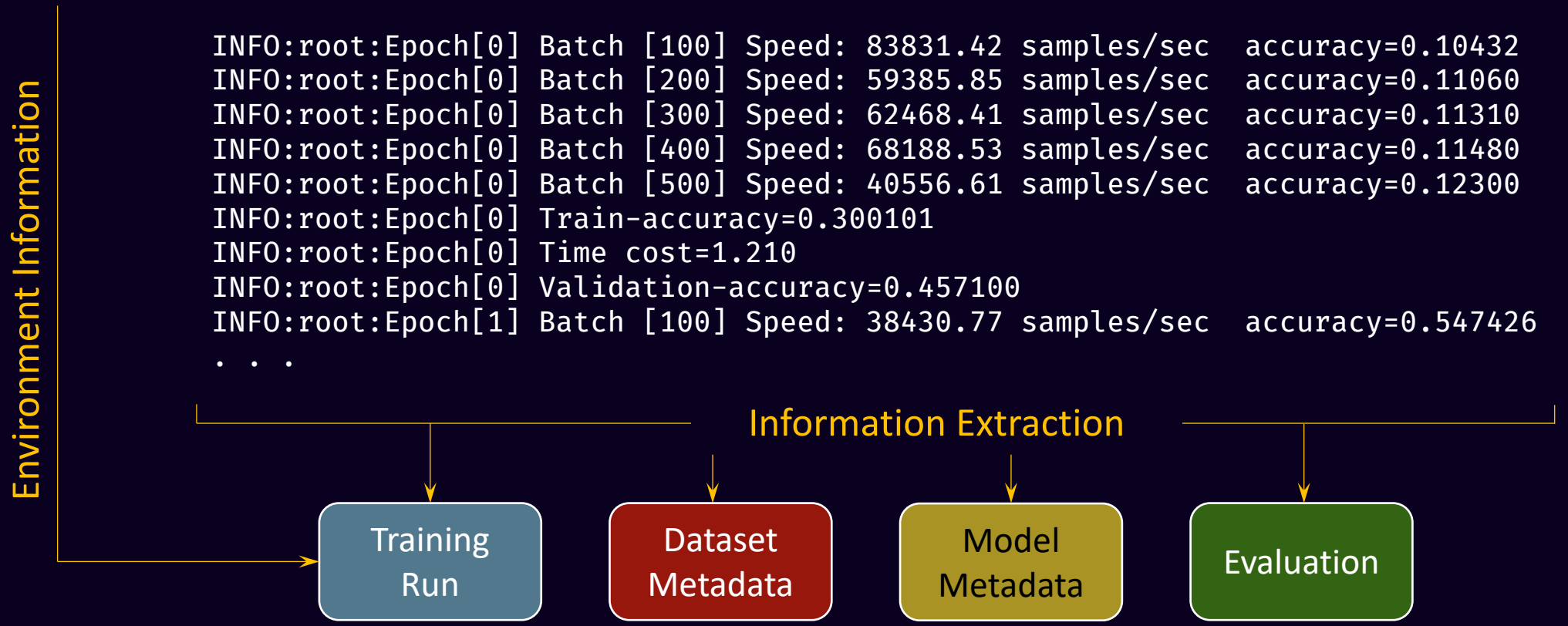
```
> pip install logexp
> logexp-run ./run-mxnet-experiment.py
```



Logexp: Non-Intrusive Experiment Tracking

```
> pip install logexp
> logexp-run ./run-mxnet-experiment.py
```

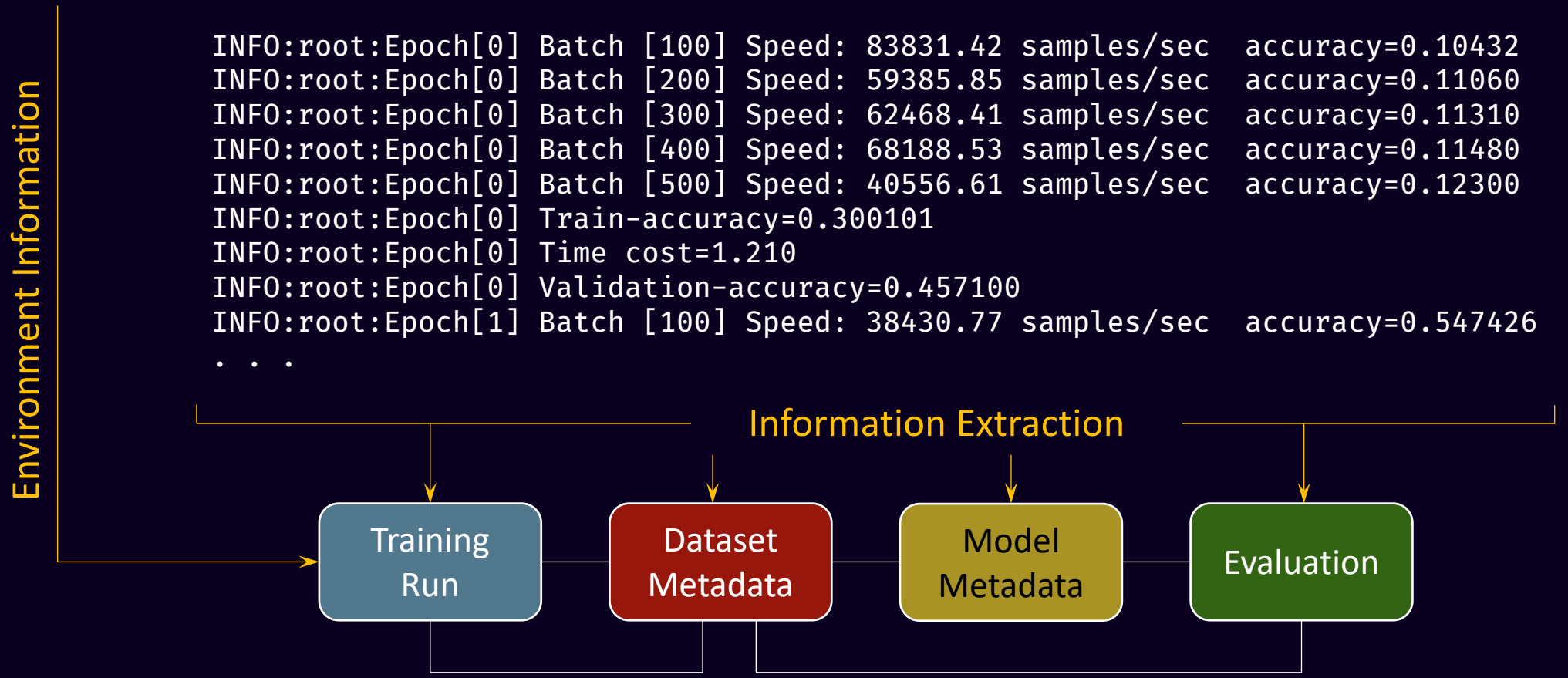
```
INFO:root:Epoch[0] Batch [100] Speed: 83831.42 samples/sec  accuracy=0.10432
INFO:root:Epoch[0] Batch [200] Speed: 59385.85 samples/sec  accuracy=0.11060
INFO:root:Epoch[0] Batch [300] Speed: 62468.41 samples/sec  accuracy=0.11310
INFO:root:Epoch[0] Batch [400] Speed: 68188.53 samples/sec  accuracy=0.11480
INFO:root:Epoch[0] Batch [500] Speed: 40556.61 samples/sec  accuracy=0.12300
INFO:root:Epoch[0] Train-accuracy=0.300101
INFO:root:Epoch[0] Time cost=1.210
INFO:root:Epoch[0] Validation-accuracy=0.457100
INFO:root:Epoch[1] Batch [100] Speed: 38430.77 samples/sec  accuracy=0.547426
. . .
```



Logexp: Non-Intrusive Experiment Tracking

```
> pip install logexp
> logexp-run ./run-mxnet-experiment.py
```

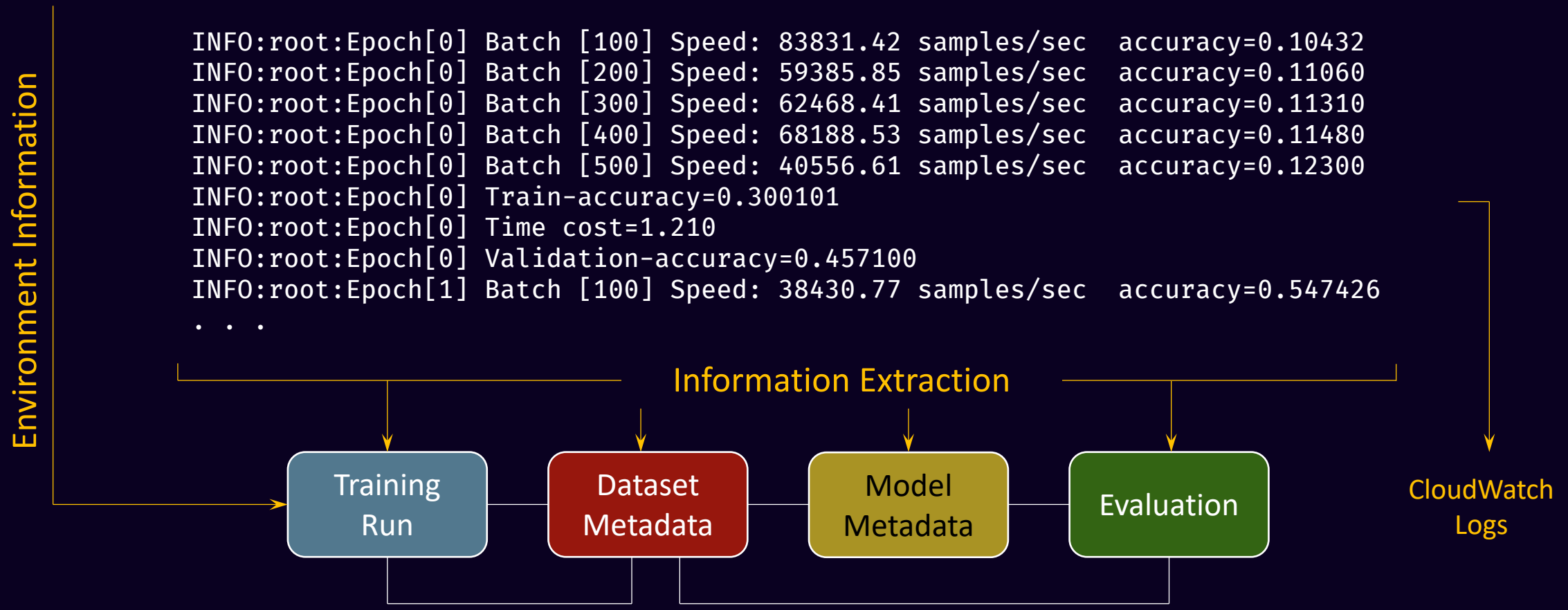
```
INFO:root:Epoch[0] Batch [100] Speed: 83831.42 samples/sec  accuracy=0.10432
INFO:root:Epoch[0] Batch [200] Speed: 59385.85 samples/sec  accuracy=0.11060
INFO:root:Epoch[0] Batch [300] Speed: 62468.41 samples/sec  accuracy=0.11310
INFO:root:Epoch[0] Batch [400] Speed: 68188.53 samples/sec  accuracy=0.11480
INFO:root:Epoch[0] Batch [500] Speed: 40556.61 samples/sec  accuracy=0.12300
INFO:root:Epoch[0] Train-accuracy=0.300101
INFO:root:Epoch[0] Time cost=1.210
INFO:root:Epoch[0] Validation-accuracy=0.457100
INFO:root:Epoch[1] Batch [100] Speed: 38430.77 samples/sec  accuracy=0.547426
. . .
```



Logexp: Non-Intrusive Experiment Tracking

```
> pip install logexp
> logexp-run ./run-mxnet-experiment.py
```

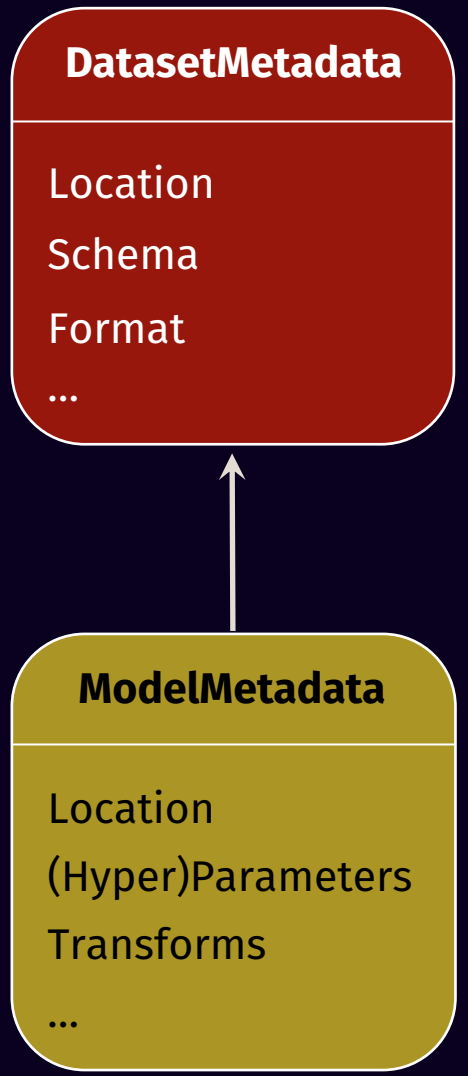
```
INFO:root:Epoch[0] Batch [100] Speed: 83831.42 samples/sec  accuracy=0.10432
INFO:root:Epoch[0] Batch [200] Speed: 59385.85 samples/sec  accuracy=0.11060
INFO:root:Epoch[0] Batch [300] Speed: 62468.41 samples/sec  accuracy=0.11310
INFO:root:Epoch[0] Batch [400] Speed: 68188.53 samples/sec  accuracy=0.11480
INFO:root:Epoch[0] Batch [500] Speed: 40556.61 samples/sec  accuracy=0.12300
INFO:root:Epoch[0] Train-accuracy=0.300101
INFO:root:Epoch[0] Time cost=1.210
INFO:root:Epoch[0] Validation-accuracy=0.457100
INFO:root:Epoch[1] Batch [100] Speed: 38430.77 samples/sec  accuracy=0.547426
. . .
```



3. Transparent Tracking

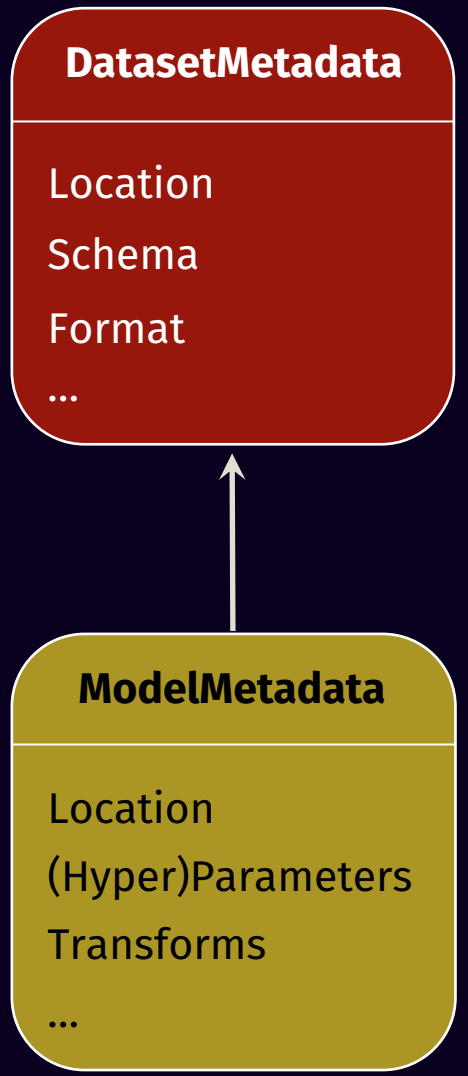
Automated Metadata Extraction (Recap)

```
val df: DataFrame = …

tracker.trackDataset(df)


val pipeline: Pipeline = …

val model: PipelineModel = pipeline.fit(df)

tracker.trackModel(model)
```



Automated Metadata Extraction (Recap)

```
val df: DataFrame = …

tracker.trackDataset(df)


val pipeline: Pipeline = …

val model: PipelineModel = pipeline.fit(df)

tracker.trackModel(model)
```



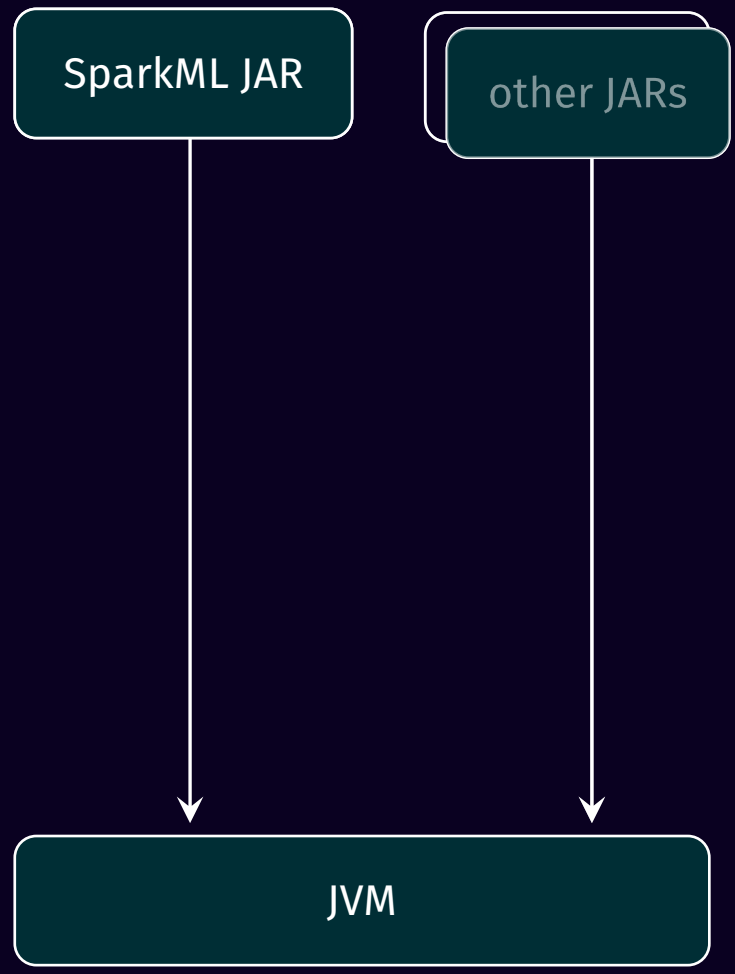
Transparent Tracking - JVM

Transparent Tracking - JVM

```
> java –cp sparkml.jar:… . . .
```

Transparent Tracking - JVM

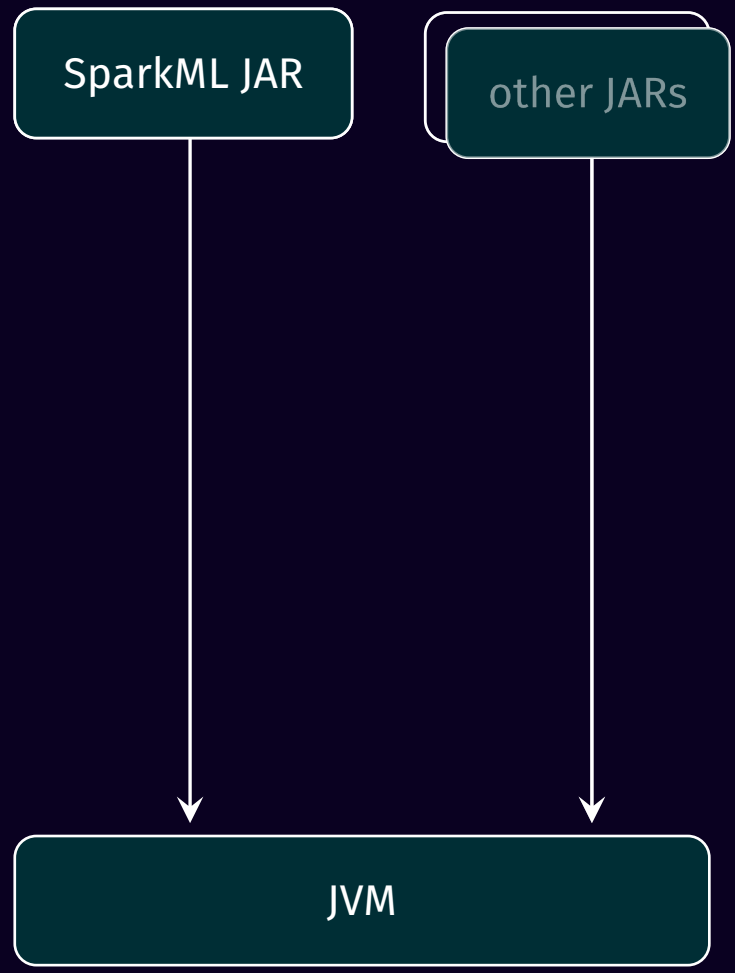
```
> java –cp sparkml.jar:… . . .
```



Transparent Tracking - JVM

```
> java –cp sparkml.jar:… . . .
```

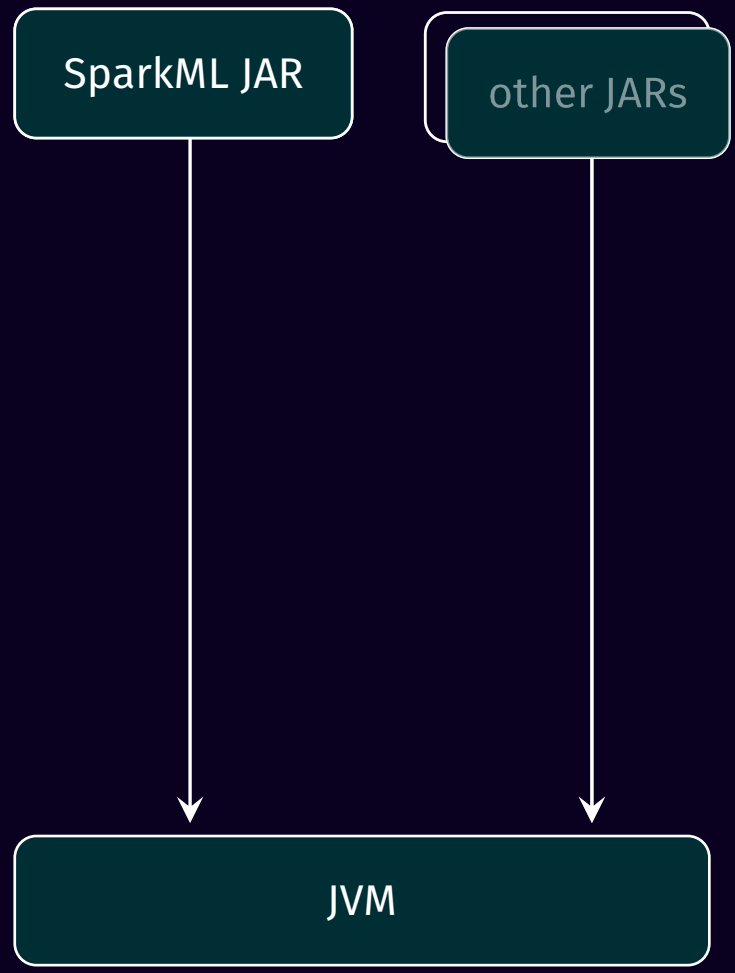
```
import org.apache.spark.ml.Pipeline

val df: DataFrame = …


val pipeline: Pipeline = …

val model: PipelineModel = pipeline.fit(df)
```



Transparent Tracking - JVM

```
> java -javaagent:tracker.jar -cp sparkml.jar:… . . .
```

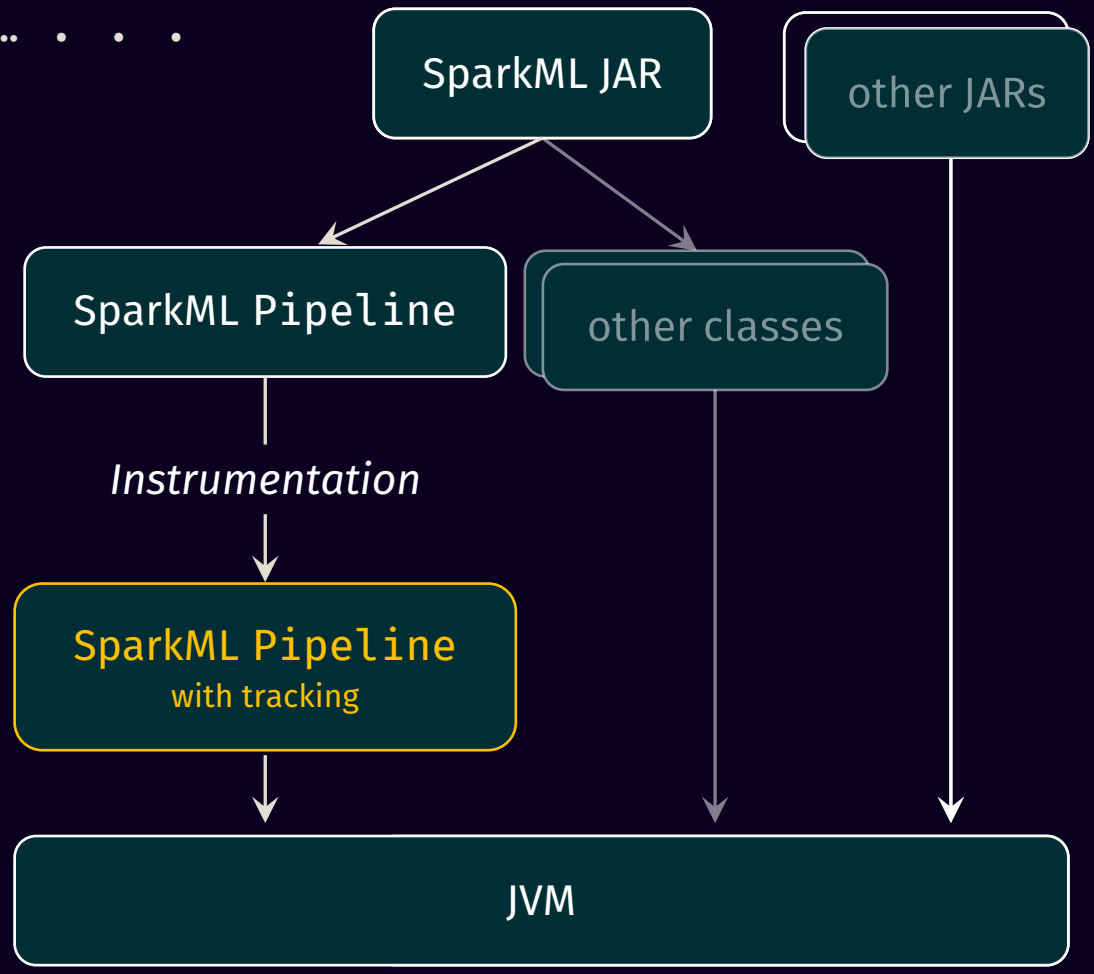
```
import org.apache.spark.ml.Pipeline

val df: DataFrame = …


val pipeline: Pipeline = …

val model: PipelineModel = pipeline.fit(df)
```



Transparent Tracking - JVM

```
> java -javaagent:tracker.jar -cp sparkml.jar:… . . .
```

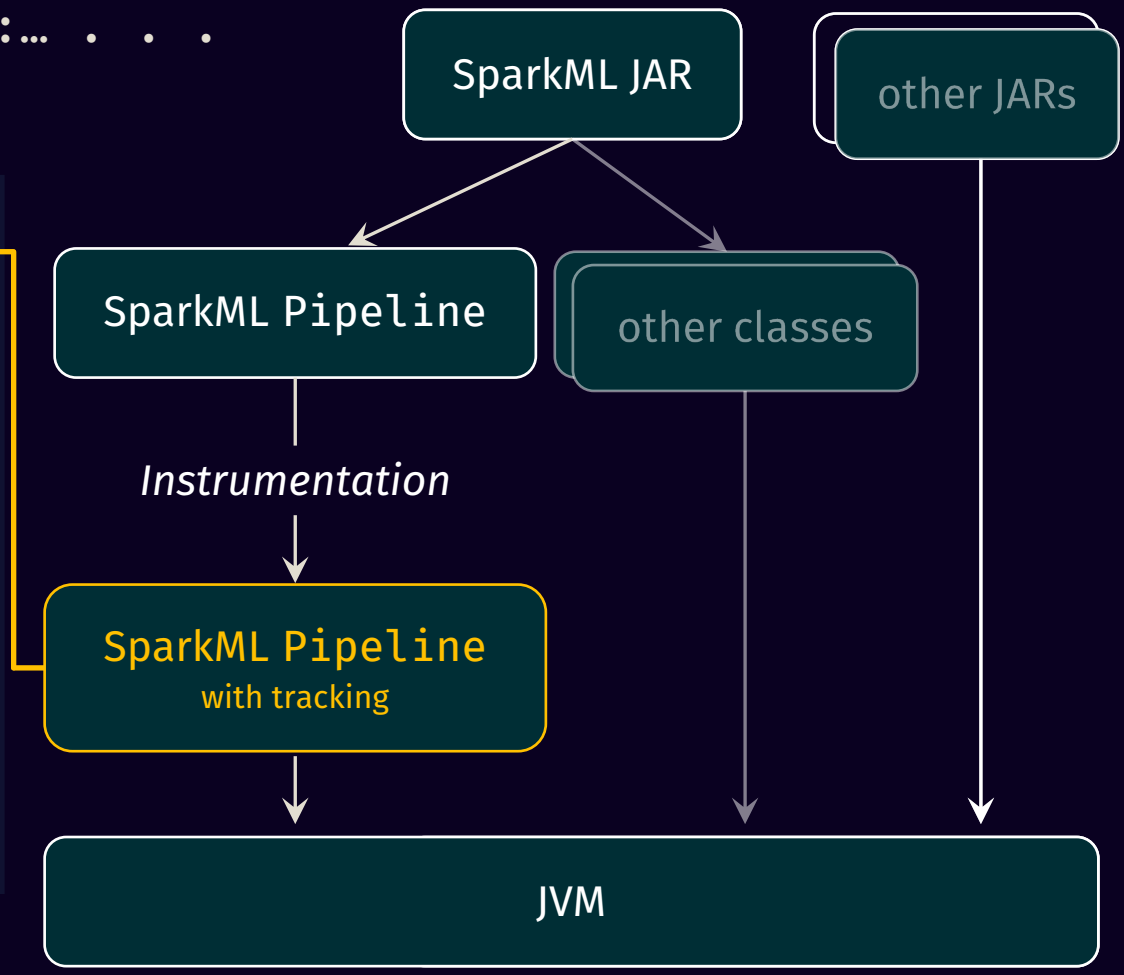
```
import org.apache.spark.ml.Pipeline

val df: DataFrame = …


val pipeline: Pipeline = …

val model: PipelineModel = pipeline.fit(df)
```



Transparent Tracking - JVM

```
> java -javaagent:tracker.jar -cp sparkml.jar:… . . .
```

```
import org.apache.spark.ml.Pipeline

val df: DataFrame = …


val pipeline: Pipeline = …

val model: PipelineModel = pipeline.fit(df)
```



Transparent Tracking - JVM

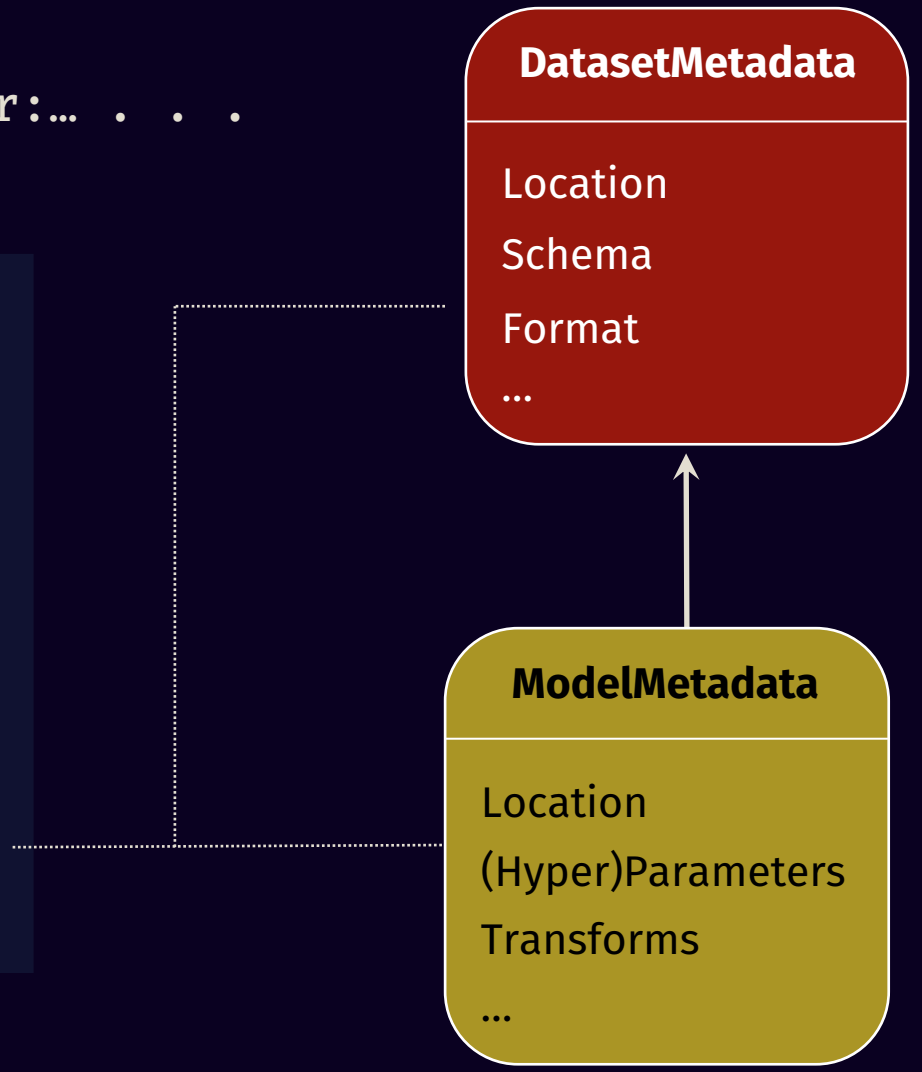
```
> java -javaagent:tracker.jar -cp sparkml.jar:… . . .
```

```
import org.apache.spark.ml.Pipeline

val df: DataFrame = …


val pipeline: Pipeline = …
val model: PipelineModel = pipeline.fit(df)
```

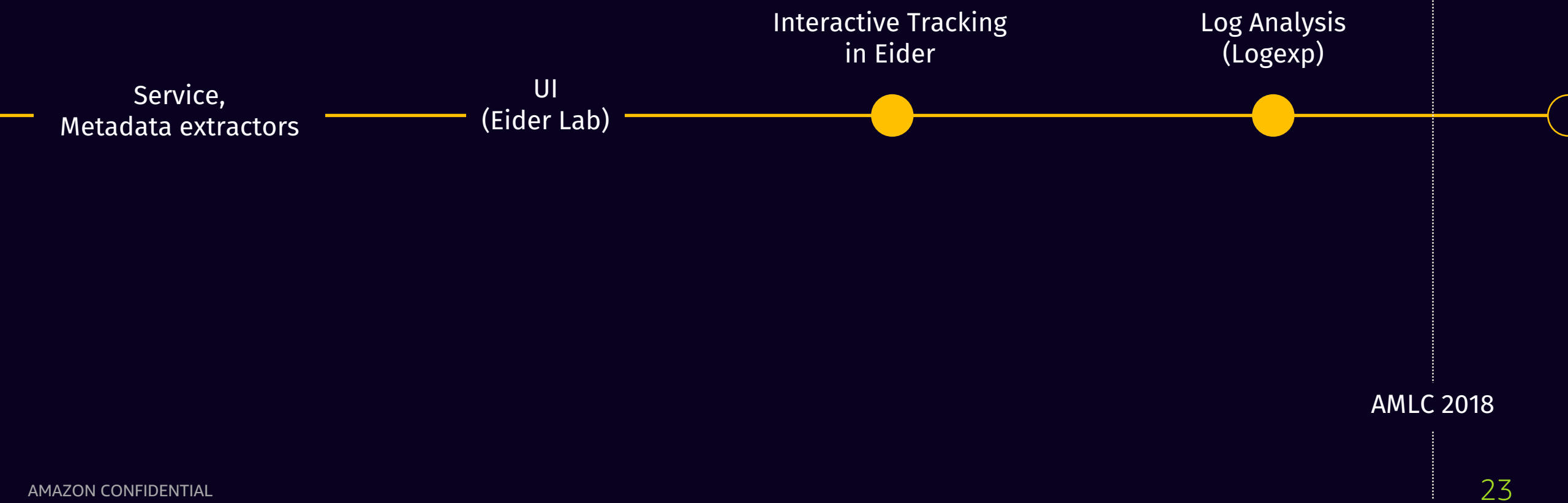
Transparent Tracking - JVM

```
> java -javaagent:tracker.jar -cp sparkml.jar:… . . .
```

```
import org.apache.spark.ml.Pipeline

val df: DataFrame = …


val pipeline: Pipeline = …
val model: PipelineModel = pipeline.fit(df)
```

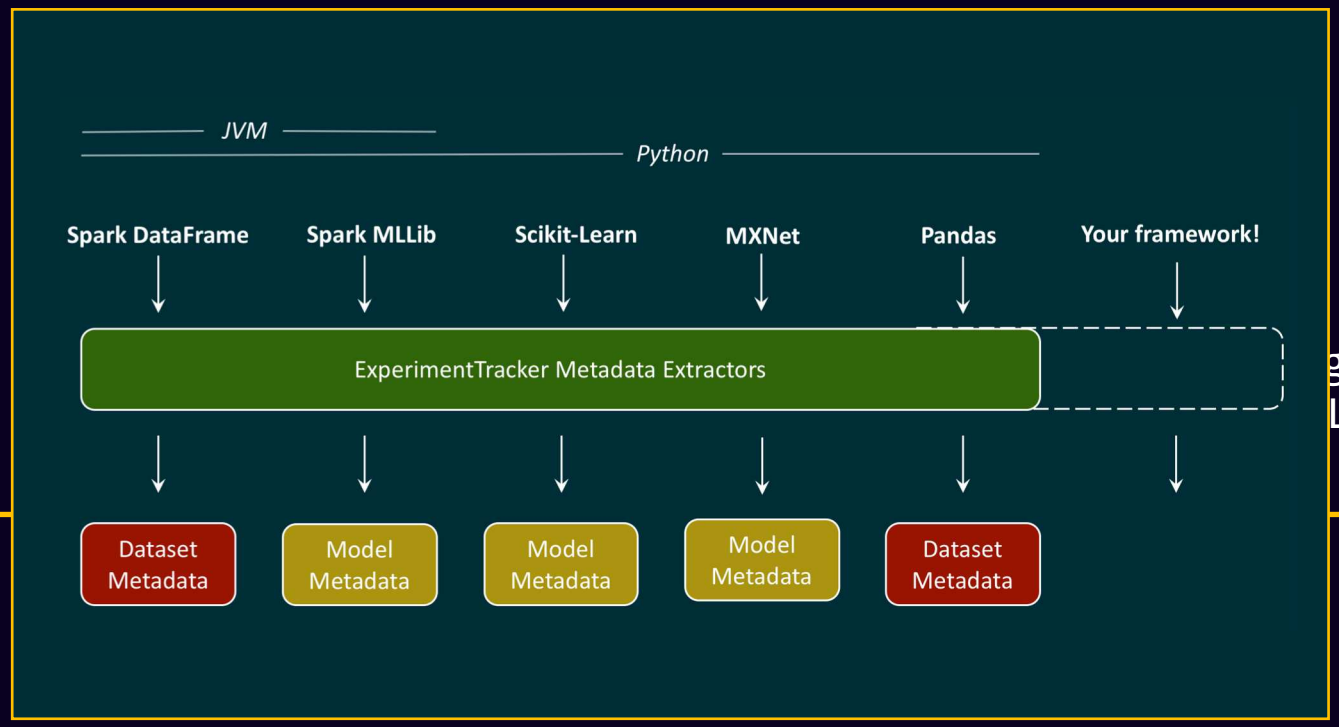


Summary & Timeline



Summary & Timeline

Service,
Metadata extractors



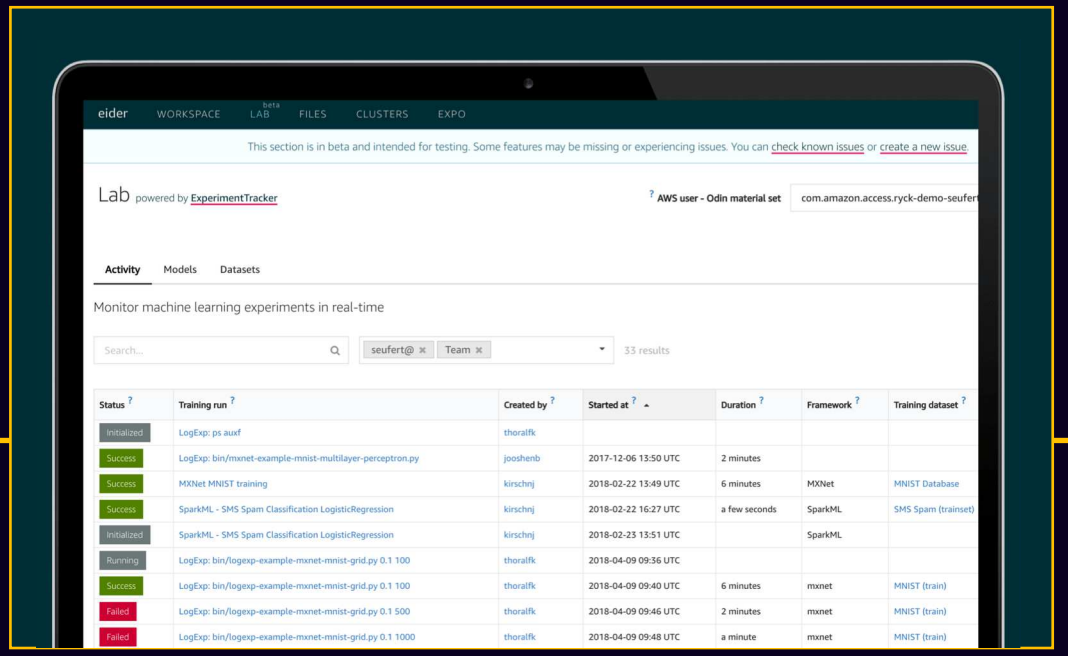
g Analysis
Logexp)

AMLC 2018

Summary & Timeline

Service,
Metadata extractors

UI
(Eider Lab)



AMLC 2018

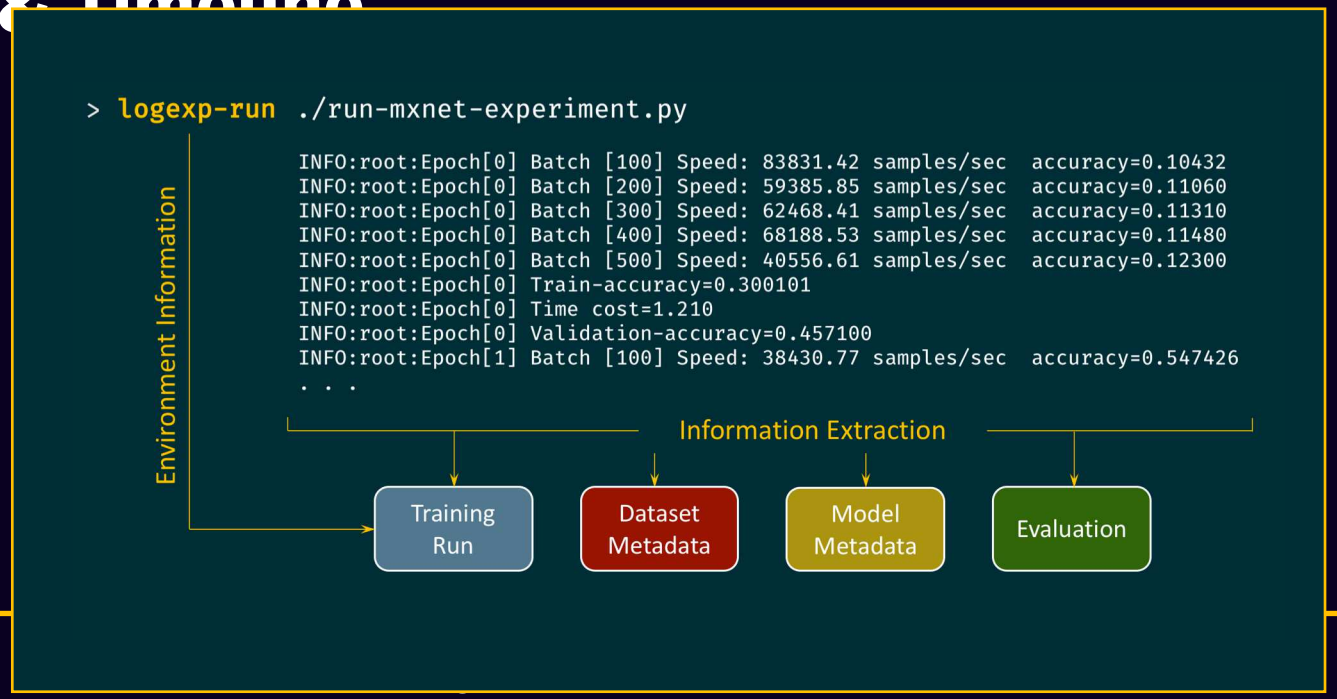
Summary & Timeline

Interactive Tracking in Eider

Log Analysis (Logexp)

AMLC 2018

Summary & Timeline



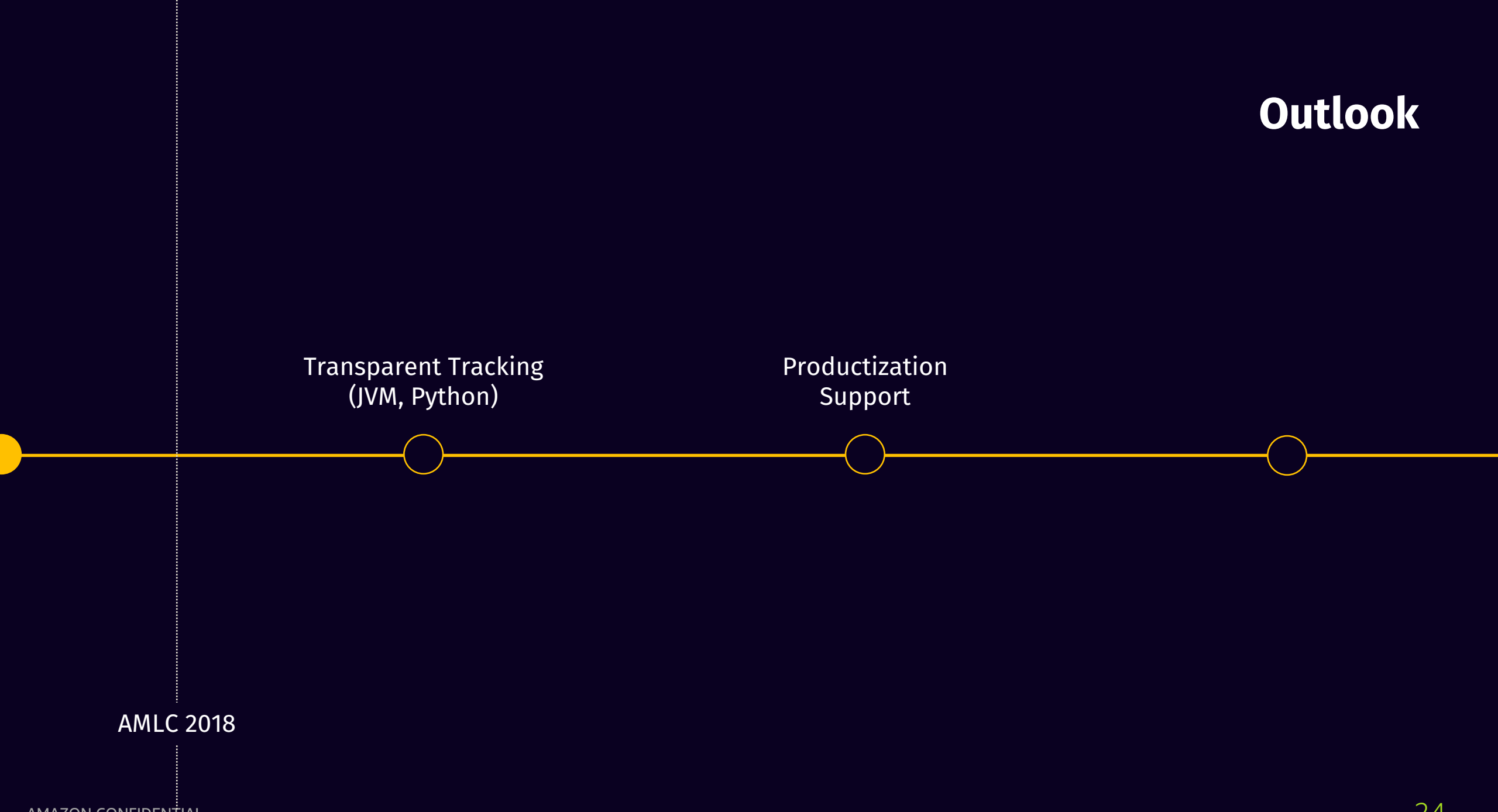
Log Analysis (Logexp)

Service, Metadata extractors

(Eider Lab)

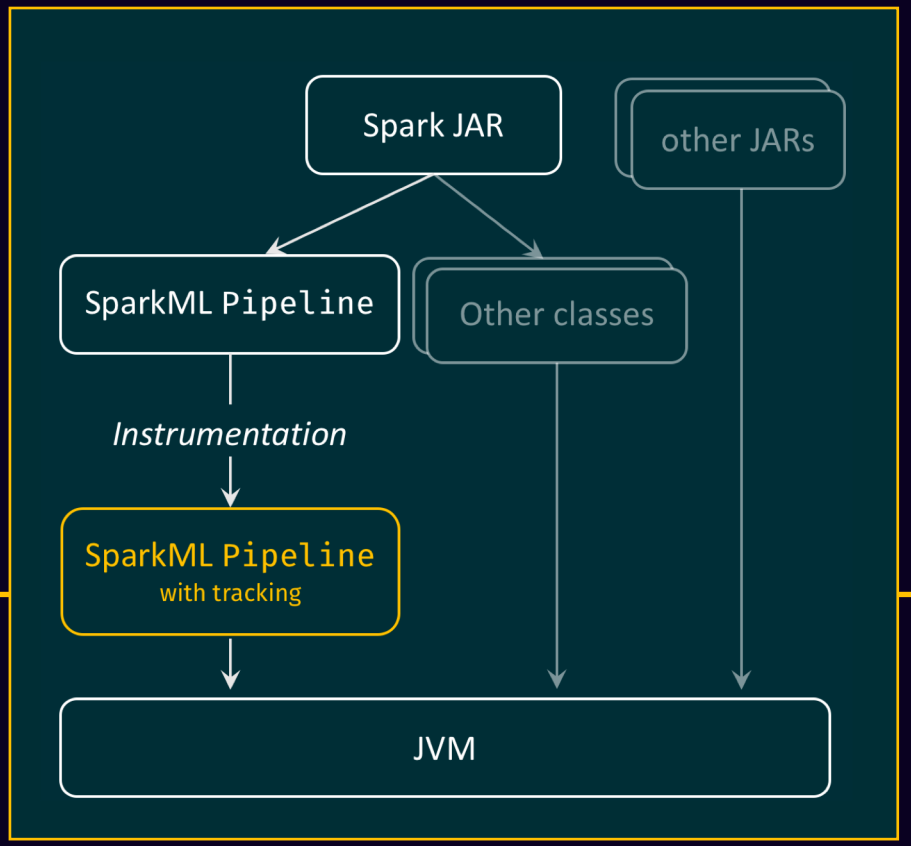
AMLC 2018

Outlook

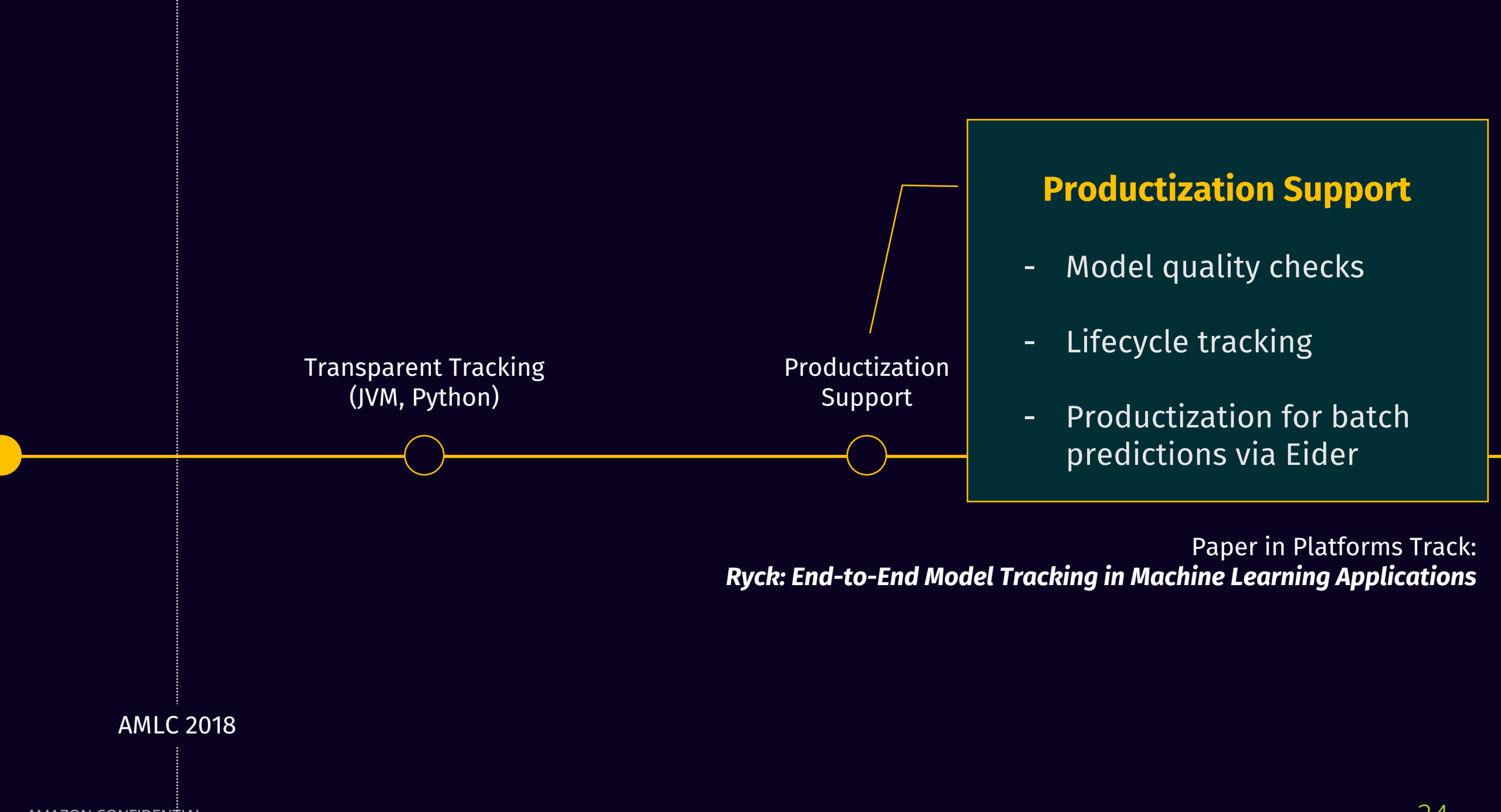


Outlook

Transparent Tracking
(JVM, Python)



AMLC 2018



Paper in Platforms Track:
Ryck: End-to-End Model Tracking in Machine Learning Applications

Team

ExperimentTracker (Ryck)

Joos-Hendrik Boese
Manager AS
Berlin

Johannes Kirschnick
SDE
Berlin

Thoralf Klein
SDE
Berlin

Stephan Seufert
SDE
Berlin

Gabriel Zimmerman
Sr. SDE
New York City

Eider

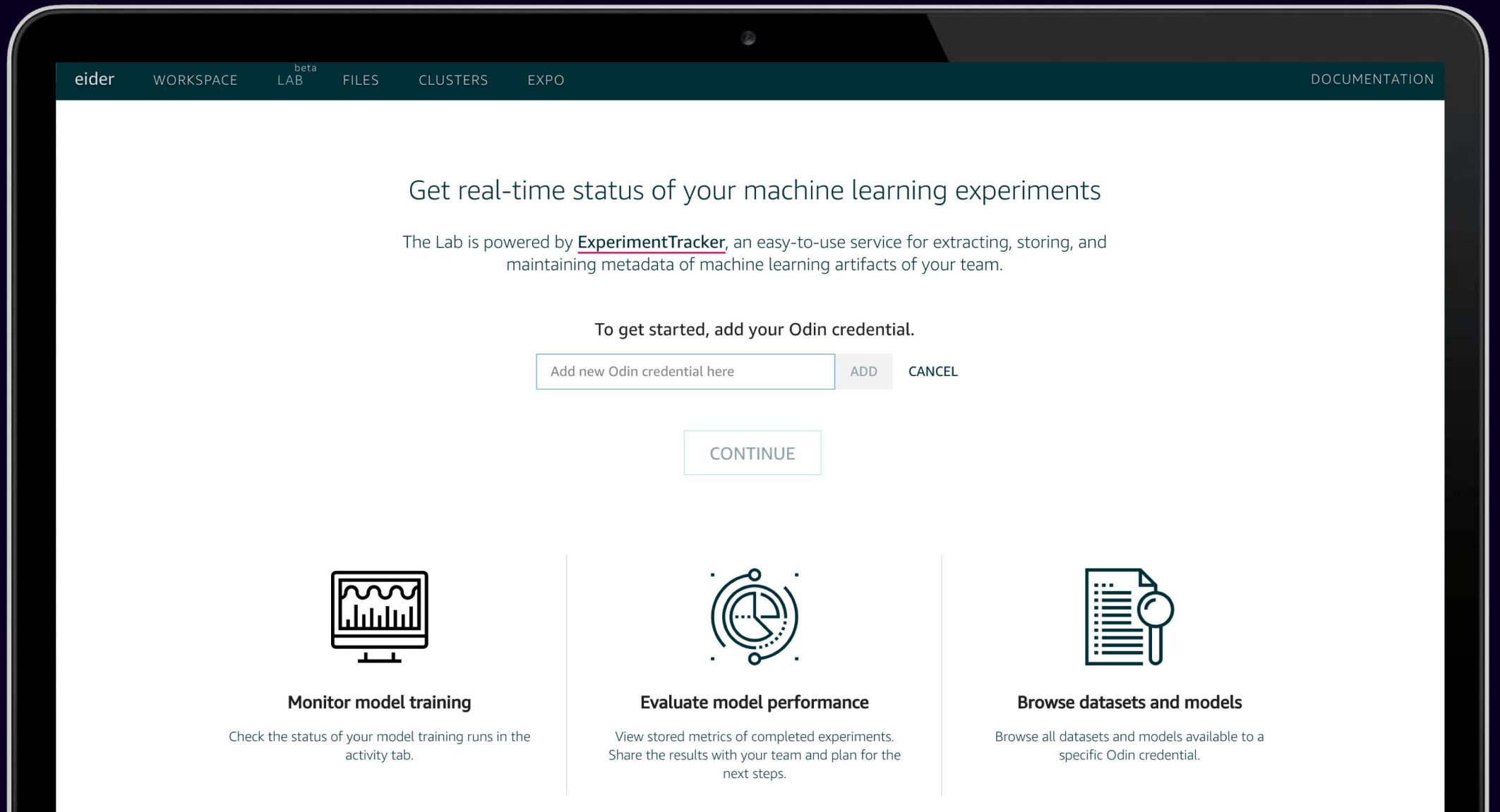
Sam Gracie
Sr. UX Designer
Seattle

Margaret Tung
UX Designer
Seattle

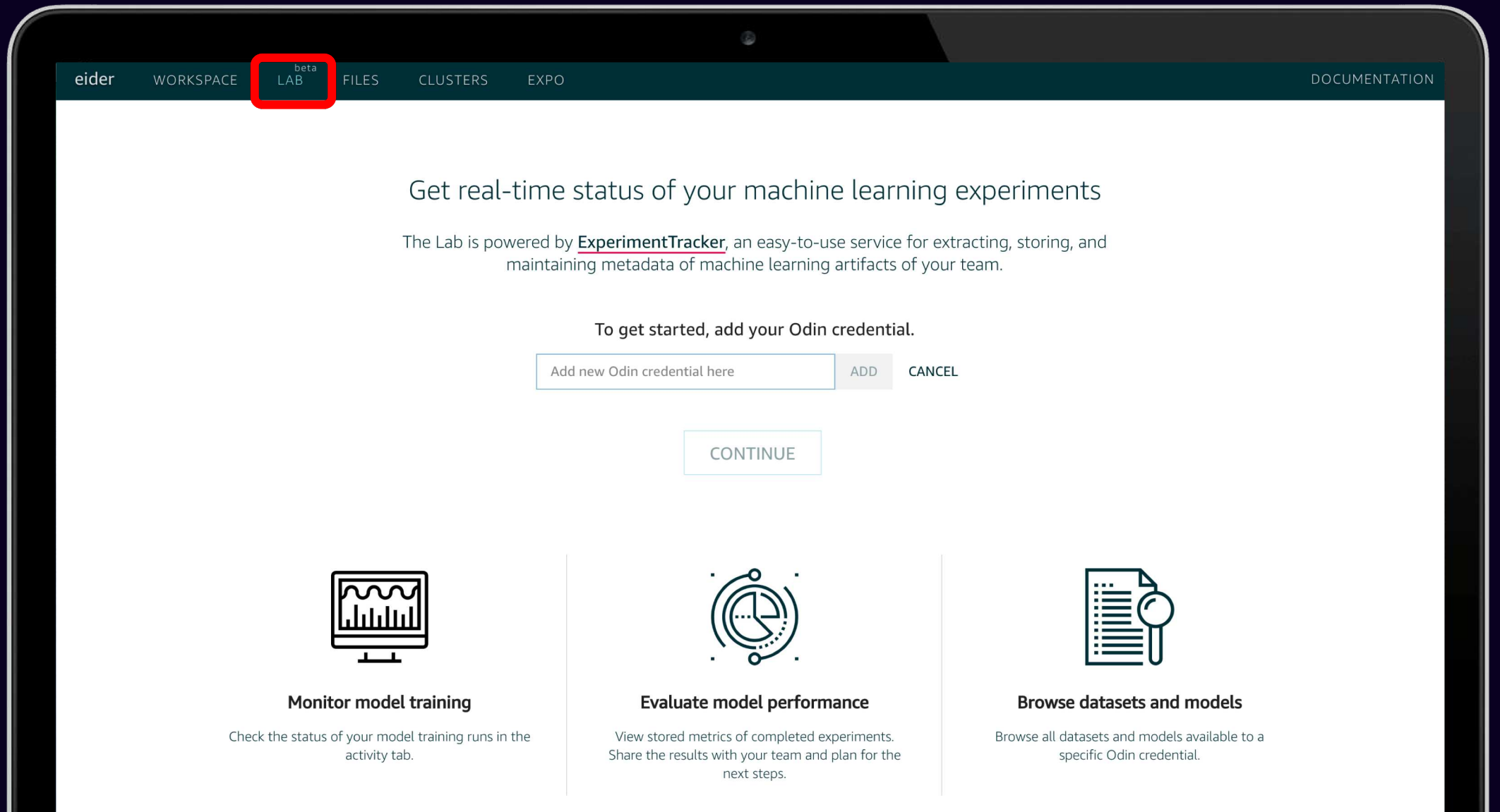
40+
teams

160,000+
entities

eider.corp.amazon.com/lab



eider.corp.amazon.com/lab



eider.corp.amazon.com/lab

eider WORKSPACE LAB beta FILES CLUSTERS EXPO DOCUMENTATION

Get real-time status of your machine learning experiments

The Lab is powered by ExperimentTracker, an easy-to-use service for extracting, storing, and maintaining metadata of machine learning artifacts of your team.

To get started, add your Odin credential.

Add new Odin credential here ADD CANCEL

CONTINUE

Monitor model training

Check the status of your model training runs in the activity tab.

Evaluate model performance

View stored metrics of completed experiments. Share the results with your team and plan for the next steps.

Browse datasets and models

Browse all datasets and models available to a specific Odin credential.

eider.corp.amazon.com/lab