AWS re:Invent

ADM302-R1

End-to-end machine learning using Spark and Amazon SageMaker

Dmitri Tchikatilov
Head, WW Business Development
Advertising & Marketing
Amazon Web Services

François Guerraz
Systems Architect
Smaato

Janko Kovačević
Machine Learning Engineer
Smaato

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

10x
Developer

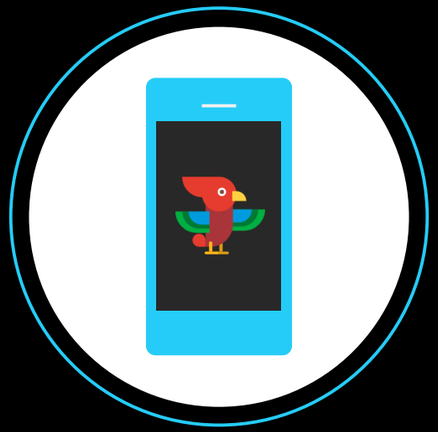


smaato

smaato

10x
Developer

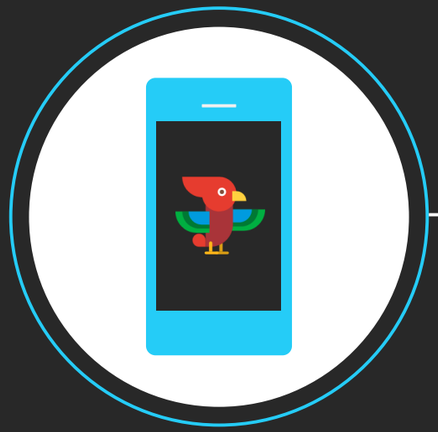




Hungry
Birds

10x
Developer





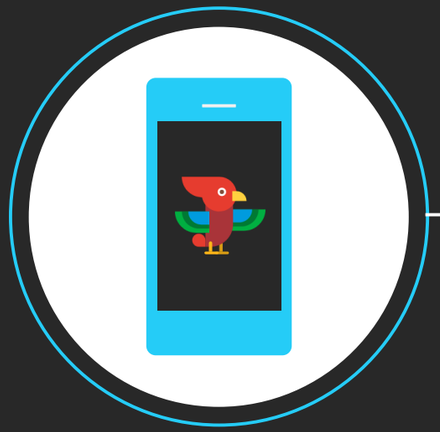
Hungry
Birds

Google Play
& App Store

smaato

10x
Developer





Hungry
Birds

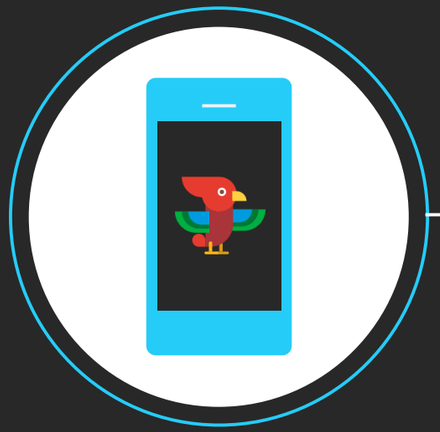
Google Play
& App Store

smaato

smaato

10x
Developer





Hungry
Birds

Google Play
& App Store

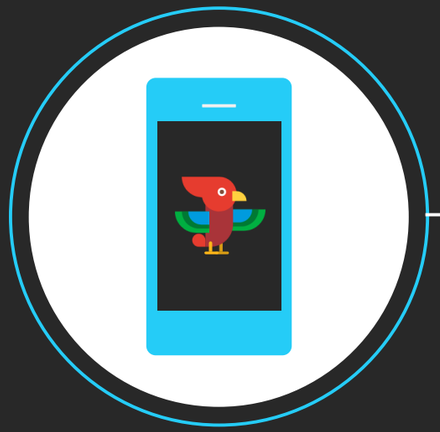






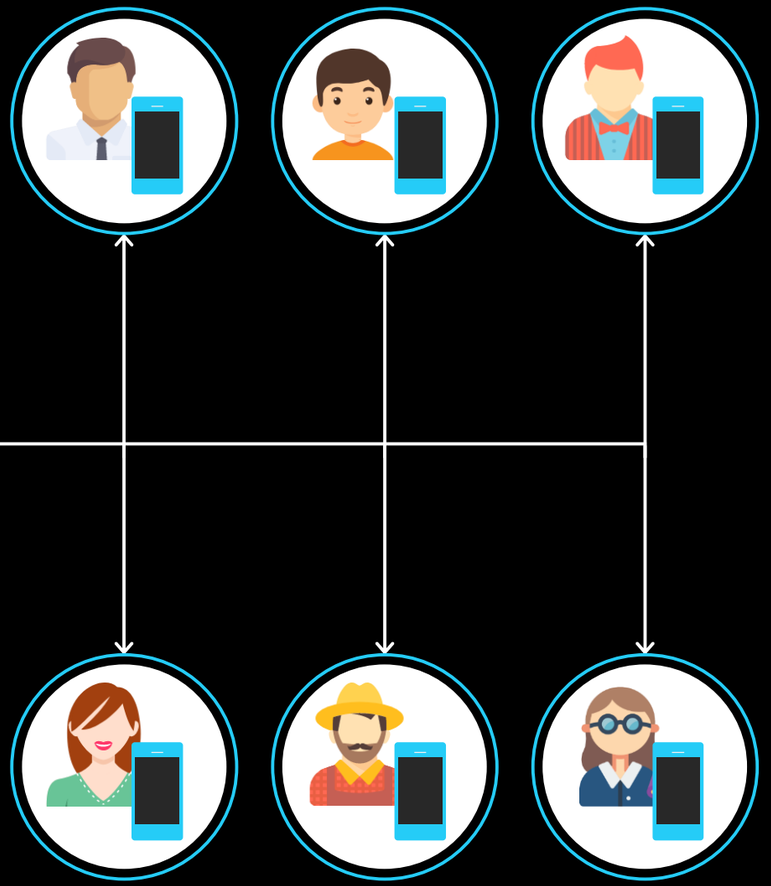
10x
Developer





Hungry
Birds

Google Play
& App Store



smaato

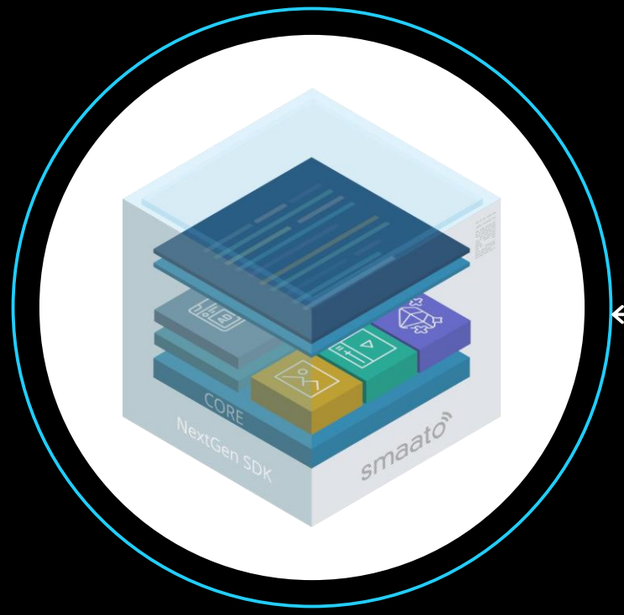
smaato





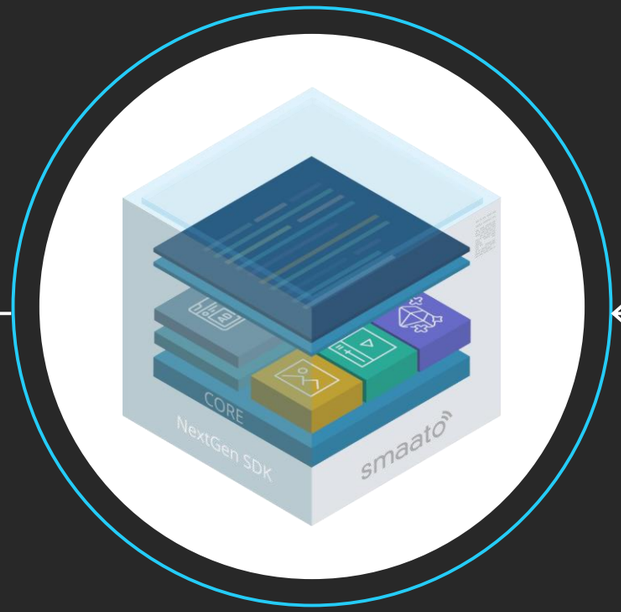
smaato





smaato

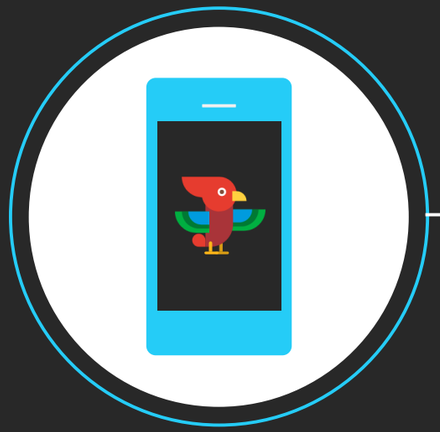



CORE
NextGen SDK
smaato


smaato

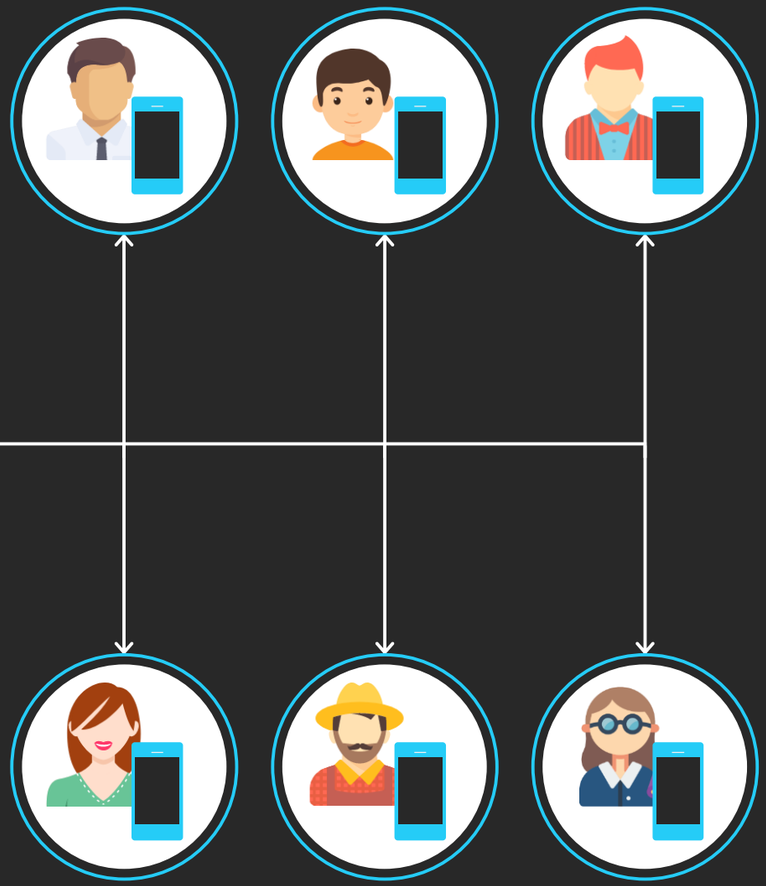
10x
Developer





Hungry
Birds

Google Play
& App Store



smaato

smaato

Hungry Birds



Request details

Sex: Male
Age: 15–25
Country: US
Device: iPhone 11

smaato

Hungry Birds



Core auction platform

Request details

Sex: Male
Age: 15–25
Country: US
Device: iPhone 11

smaato

Hungry Birds



Core auction platform

Demand partner 100

Demand partner 101

Demand partner 102

Demand partner 103

Request details

Sex: Male
Age: 15–25
Country: US
Device: iPhone 11

smaato

Hungry Birds



Request details

Sex: Male
Age: 15–25
Country: US
Device: iPhone 11

Core auction platform

Auction

Demand partner 100

Demand partner 101

Demand partner 102

Demand partner 103

smaato

Hungry Birds



Request details

Sex: Male
Age: 15–25
Country: US
Device: iPhone 11

Core auction platform

Auction

Request details

Demand partner 100

Demand partner 101

Demand partner 102

Demand partner 103

smaato

Hungry Birds



Request details

Sex: Male
Age: 15–25
Country: US
Device: iPhone 11

Core auction platform

Auction

$10 Demand partner 100

$12 Demand partner 101

$0 Demand partner 102

$7 Demand partner 103

smaato

Hungry Birds



Request details

Sex: Male
Age: 15–25
Country: US
Device: iPhone 11

Core auction platform

Auction

DSP 100: $10,00
DSP 101: $12,00
DSP 102: No bid
DSP 103: $7,00

$10 — Demand partner 100

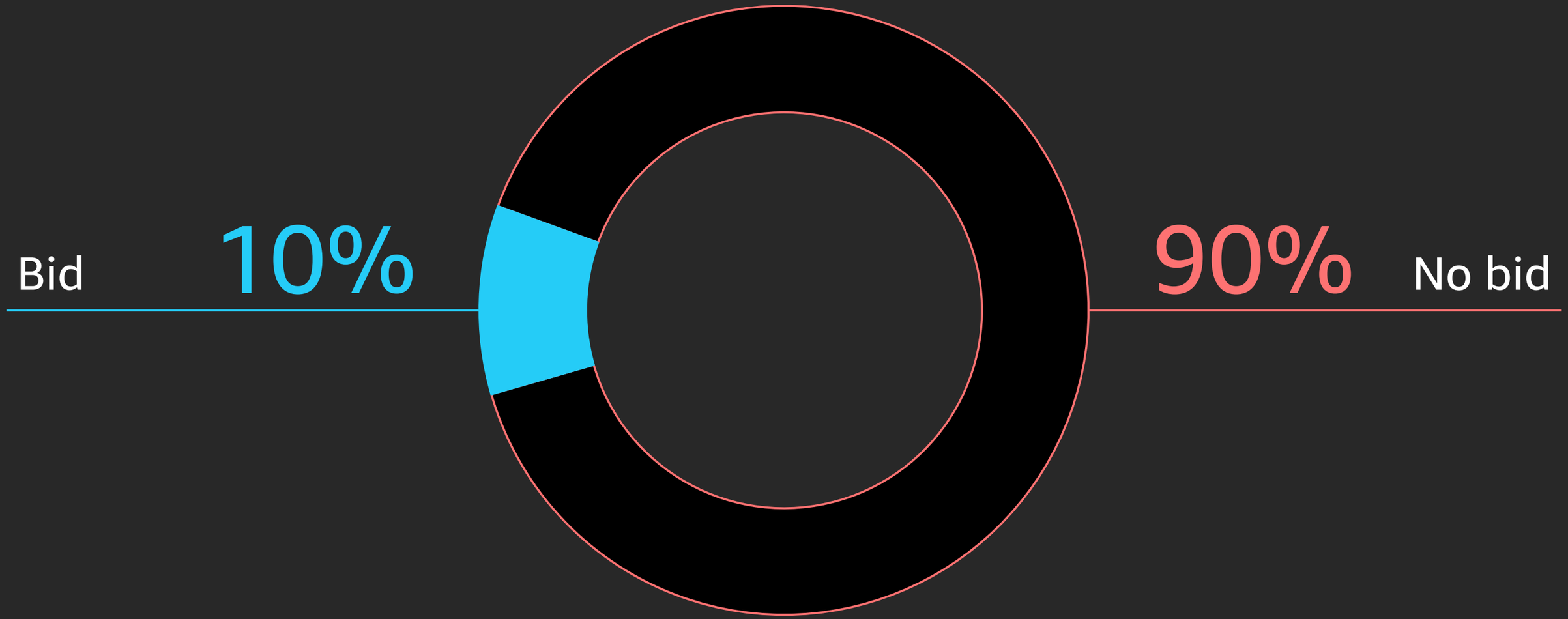
$12 — Demand partner 101

$0 — Demand partner 102

$7 — Demand partner 103

Share of ad requests that receive no bids

smaato



smaato

Hungry Birds



Request details

Sex: Male
Age: 15–25
Country: US
Device: iPhone 11

Core auction platform

Auction

No winner



$0 Demand partner 100

$0 Demand partner 101

$0 Demand partner 102

$0 Demand partner 103

Hungry Birds



Request details

Sex: Male
Age: 15–25
Country: US
Device: iPhone 11

Core auction platform

Auction

No winner



smaato

$0 Demand partner 100

$0 Demand partner 101

$0 Demand partner 102

$0 Demand partner 103

Waste of time | Waste of CPU | Waste of bandwidth

smaato

Hungry Birds



Request details

Sex: Male
Age: 15–25
Country: US
Device: iPhone 11

Core auction platform

Request details

Traffic-filtering service

Demand partner 100

Demand partner 101

Demand partner 102

Demand partner 103

smaato

Hungry Birds



No ad

Core auction platform

Request details

Traffic-filtering service

Demand partner 100

Demand partner 101

Demand partner 102

Demand partner 103

Request details

Sex: Male
Age: 15–25
Country: US
Device: iPhone 11

smaato

Hungry Birds



Request details

Sex: Male
Age: 15–25
Country: US
Device: iPhone 11

Ad

Core auction platform

Request details

Traffic-filtering service

Request details

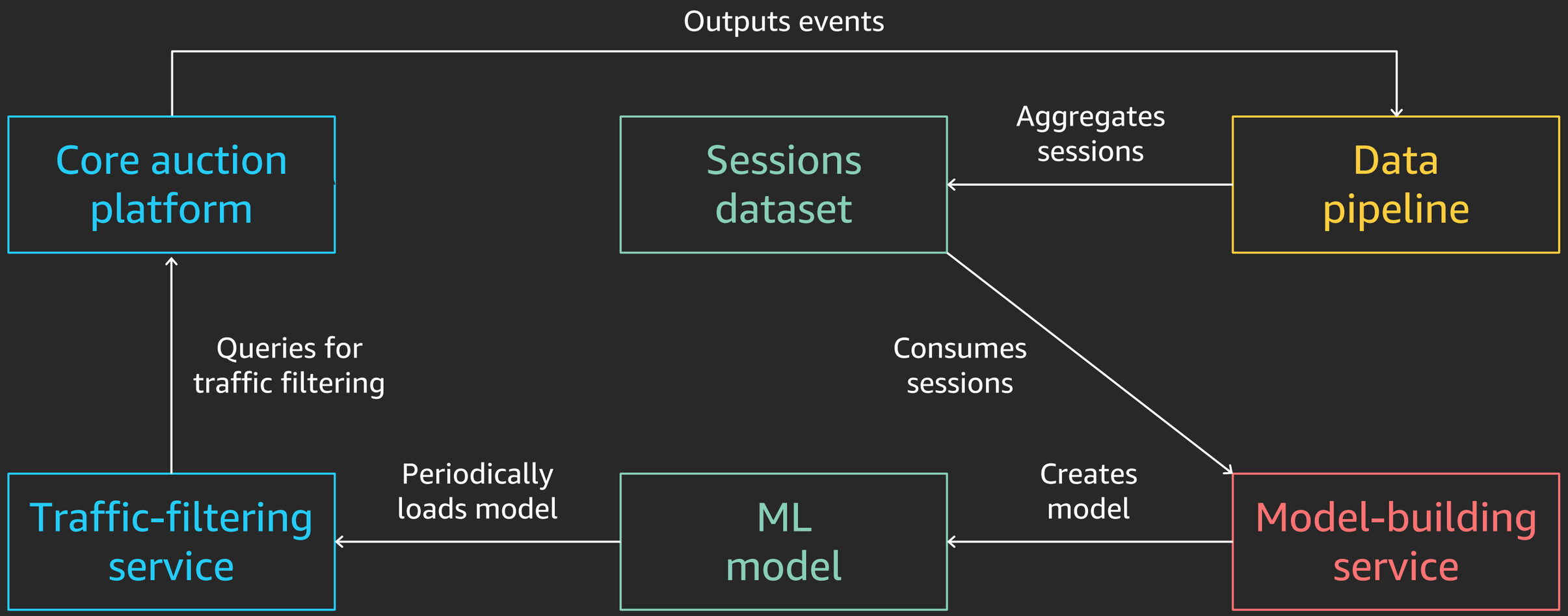
Demand partner 100

Demand partner 101

Demand partner 102

Demand partner 103

smaato


Outputs events
Core auction platform
Sessions dataset
Aggregates sessions
Data pipeline
Queries for traffic filtering
Consumes sessions
Traffic-filtering service
Periodically loads model
ML model
Creates model
Model-building service

smaato

Compute



Core auction platform

Traffic-filtering service



Amazon S3

Sessions dataset

ML model



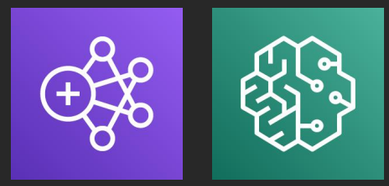
Amazon Kinesis



Data pipeline

ML

Model-building service

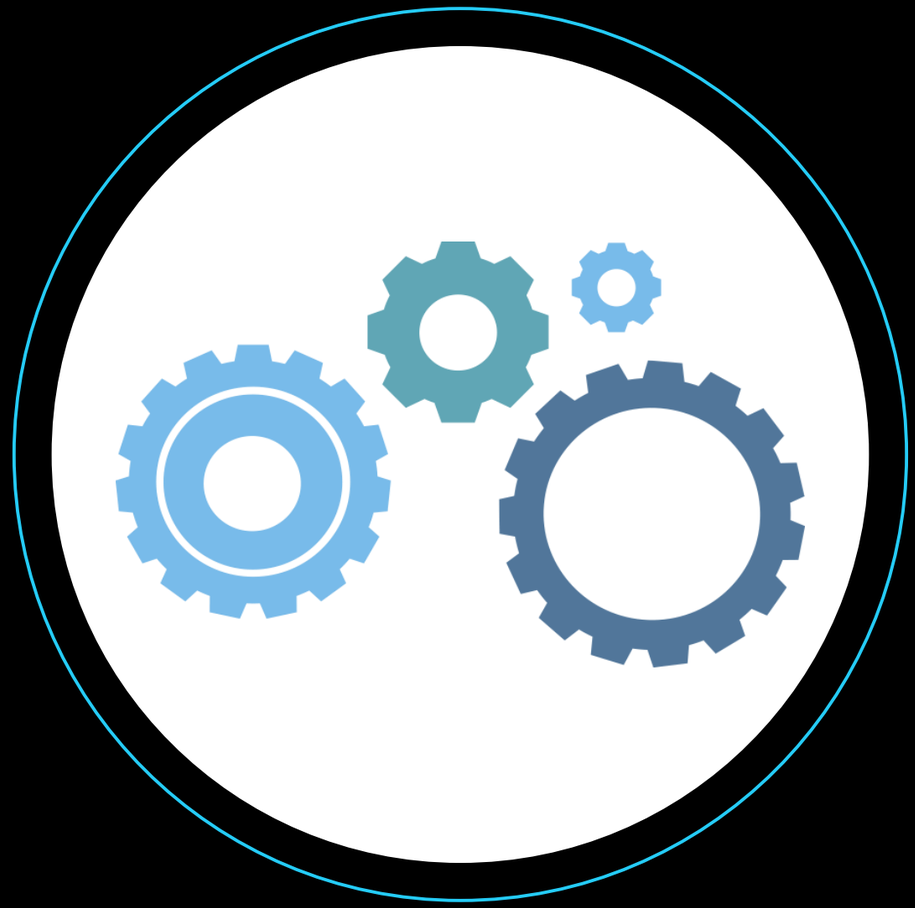


smaato

Improve
traffic filtering

Reuse existing
infrastructure

Use best software
development practices

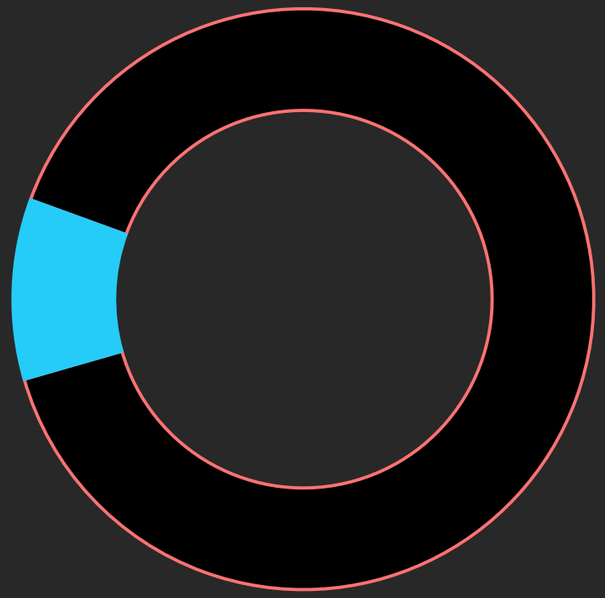


Challenge introduction

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

90% of incoming ad requests do not generate any revenue

The model needs to predict
which ad requests will receive bids



Bid 10%

No bid 90%



smaato

Incoming ad request example

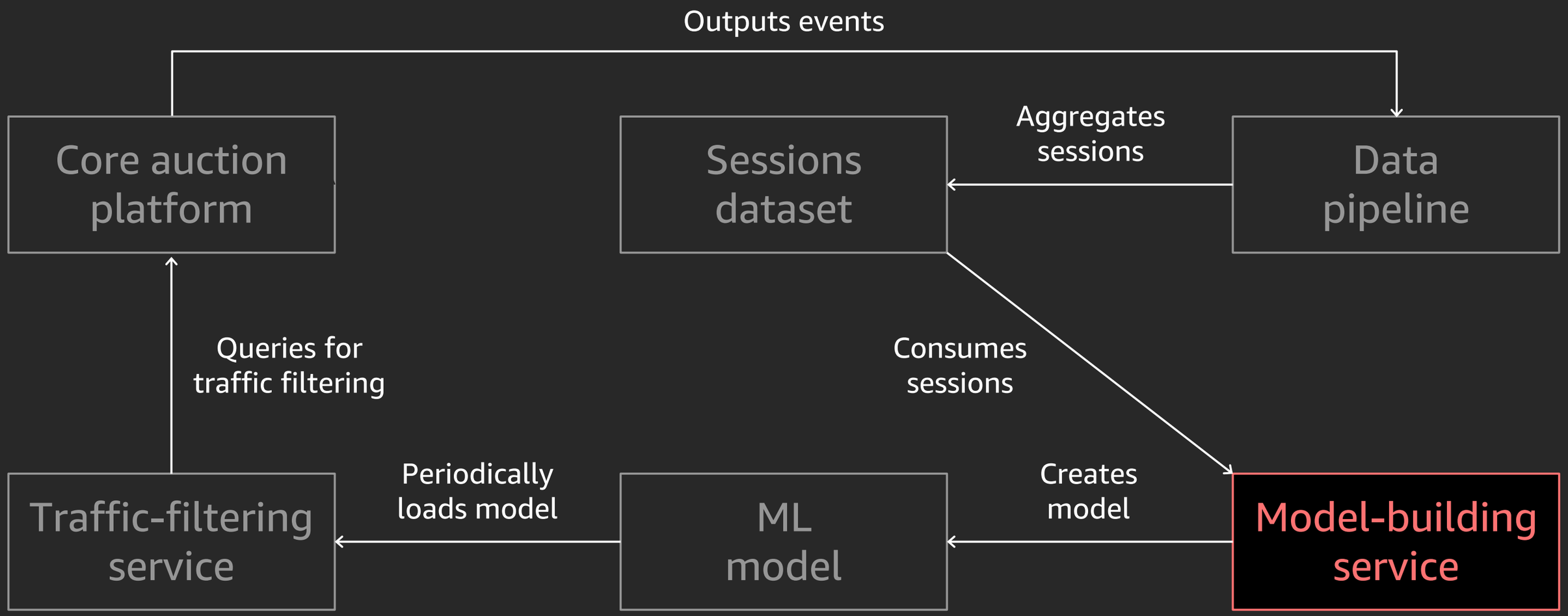
smaato

Open RTB protocol

Request time	Epoch time format
Country code	US, DE, GB, FR, HR
City name	Las Vegas, Hamburg, Zagreb
Application name	Angry Birds, Diablo Immortal
Device brand	Samsung, Apple, Xiaomi
Ad type	Banner, native, video
Ad resolution	320x50, 300x250, 1440x2560

+70 more features

smaato


Outputs events
Core auction platform
Sessions dataset
Aggregates sessions
Data pipeline
Queries for traffic filtering
Consumes sessions
Traffic-filtering service
Periodically loads model
ML model
Creates model
Model-building service

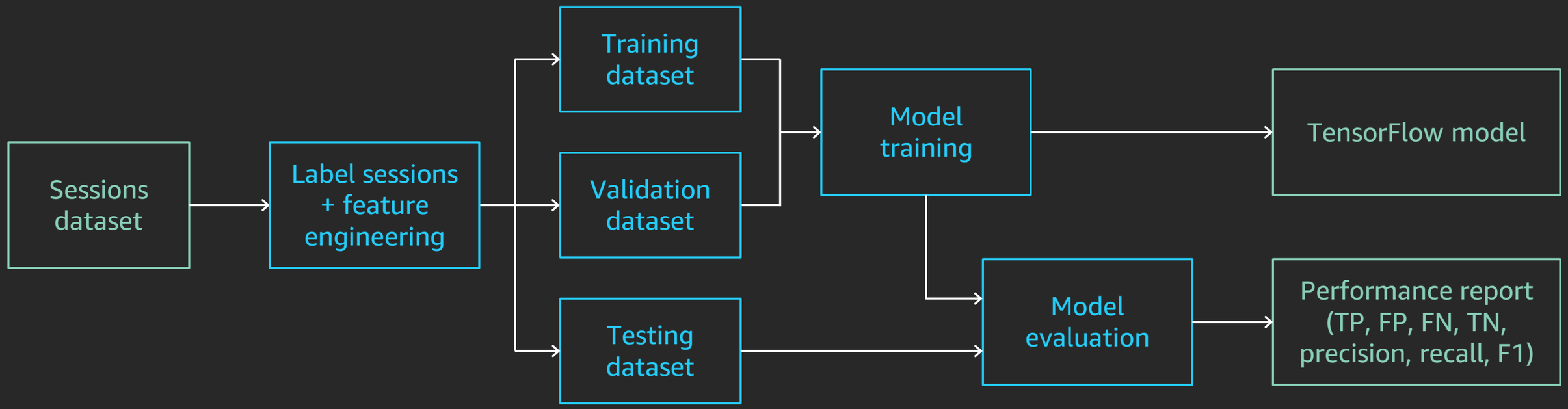
Machine learning pipeline

AWS re:Invent

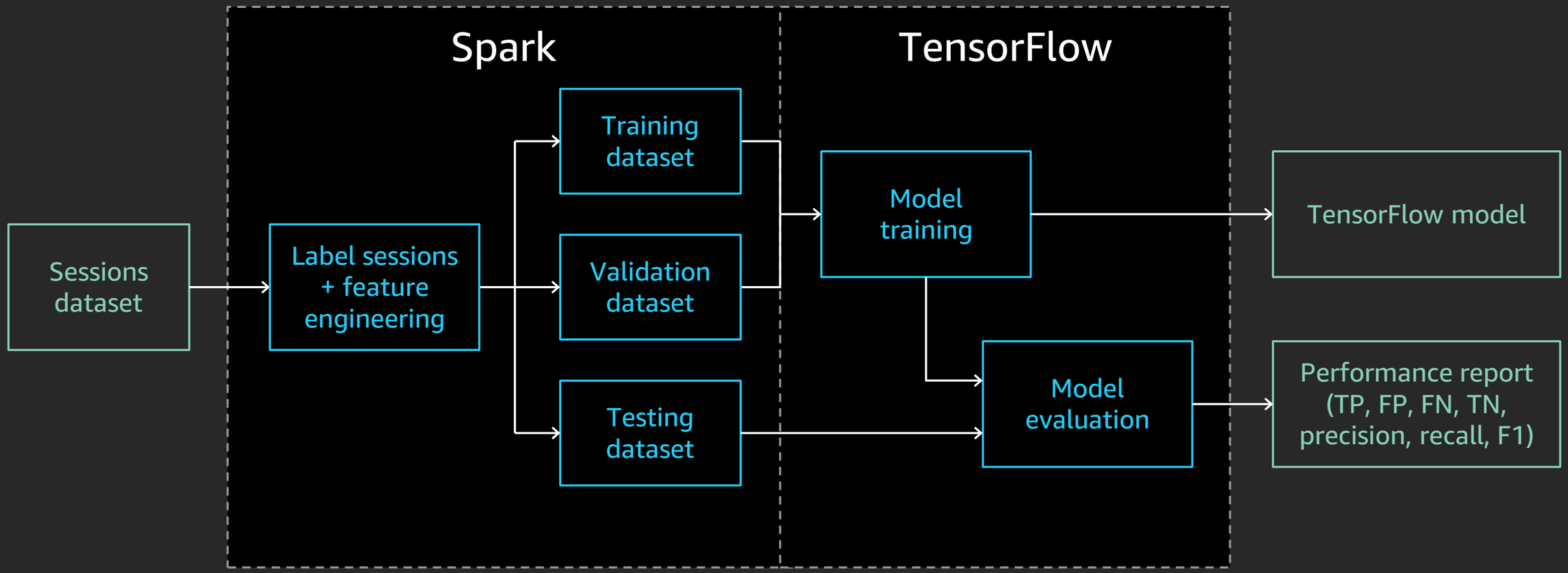
© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

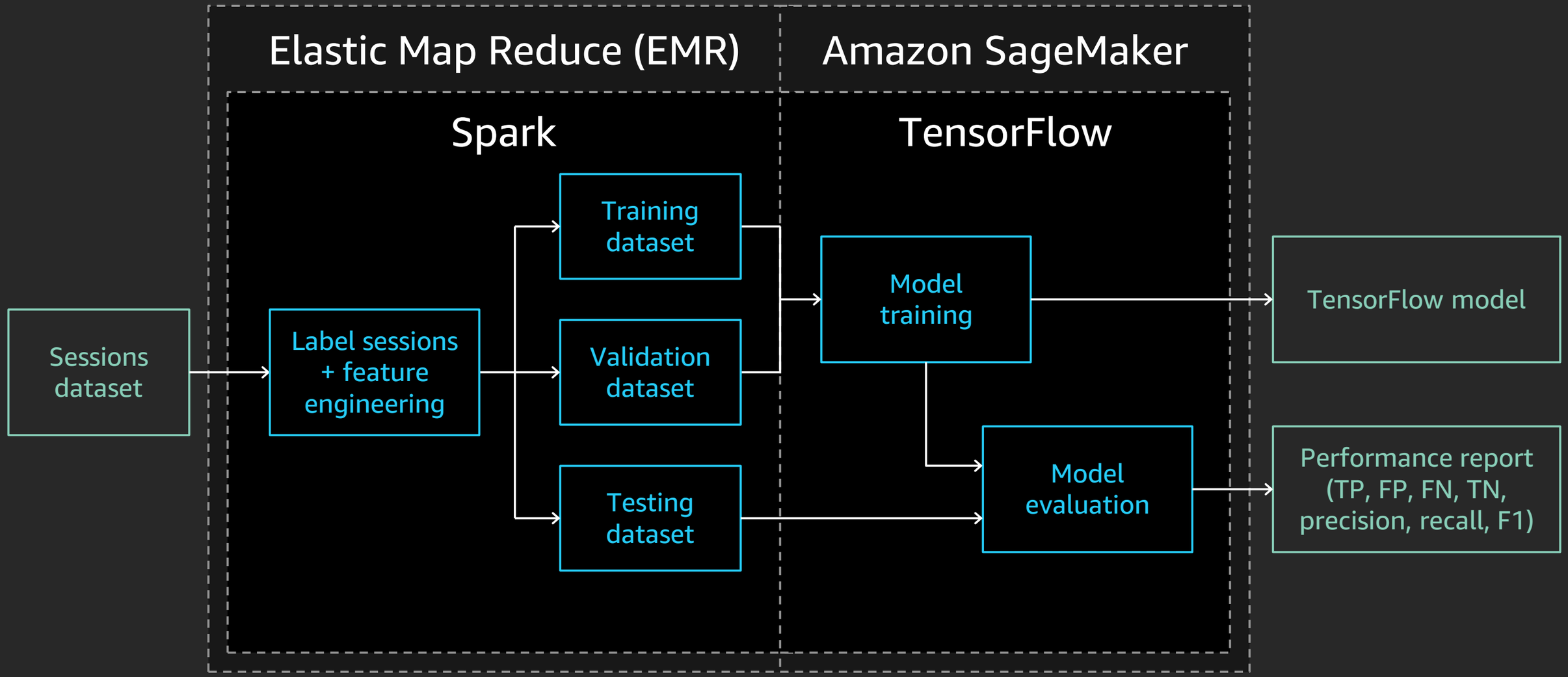
aws

smaato


Sessions dataset
Label sessions + feature engineering
Training dataset
Validation dataset
Testing dataset
Model training
TensorFlow model
Model evaluation
Performance report (TP, FP, FN, TN, precision, recall, F1)

smaato


Spark
TensorFlow
Sessions dataset
Label sessions + feature engineering
Training dataset
Validation dataset
Testing dataset
Model training
Model evaluation
TensorFlow model
Performance report (TP, FP, FN, TN, precision, recall, F1)


Elastic Map Reduce (EMR)
Amazon SageMaker
Spark
TensorFlow
Sessions dataset
Label sessions + feature engineering
Training dataset
Validation dataset
Testing dataset
Model training
Model evaluation
TensorFlow model
Performance report (TP, FP, FN, TN, precision, recall, F1)

How to develop machine learning model
(cookbook)

1. Dataset labeling
2. Feature engineering
3. Model selection
4. Model training
5. Model testing
6. Repeat steps 2 to 6 until model performs well



Problems?

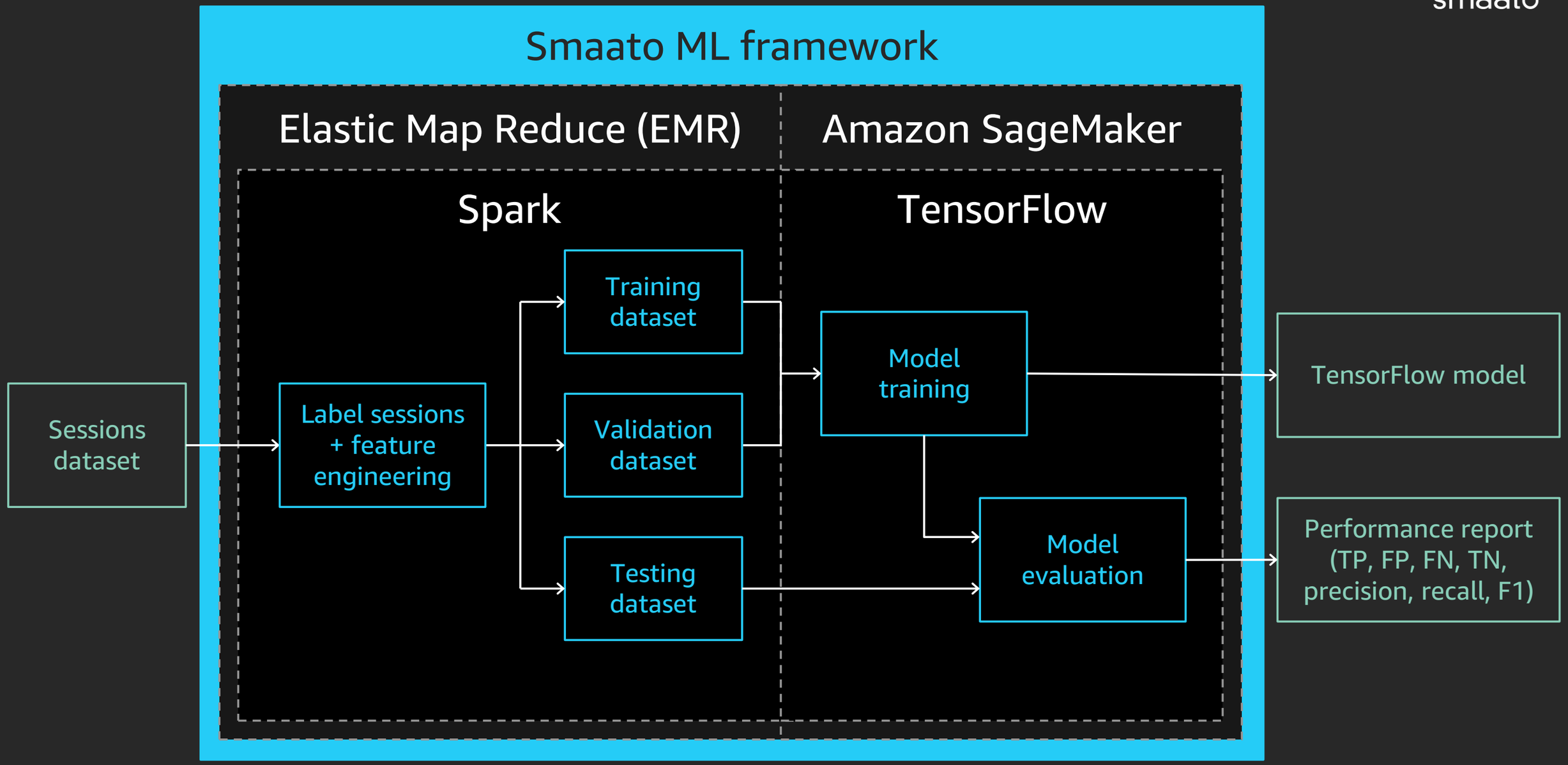
EMR cluster creation/bootstrap	5 min
Spark execution time	10–60 min
Spark logs to arrive on Amazon S3	5 min
Amazon SageMaker cluster creation/bootstrap	5 min
TensorFlow training job	10–60 min

One development iteration between 30 and 120 minutes

smaato



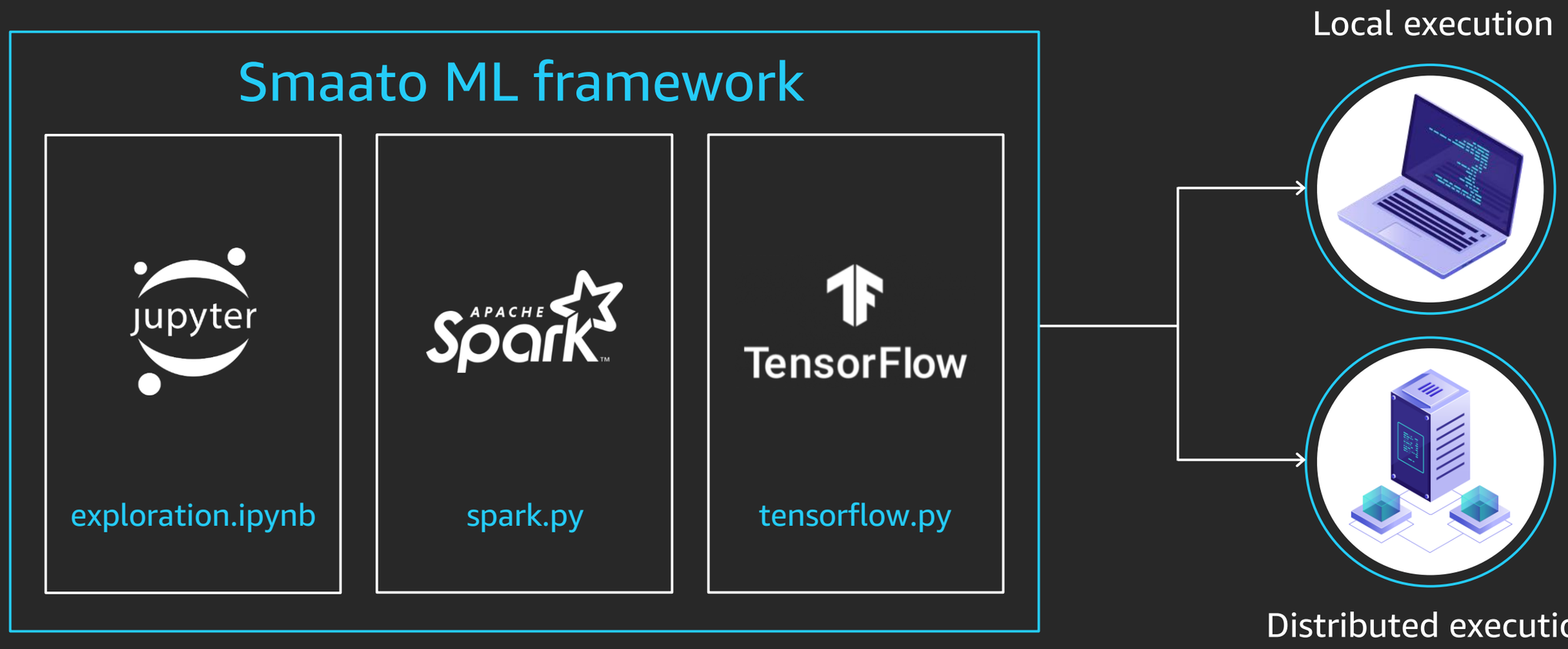
smaato


Smaato ML framework
Elastic Map Reduce (EMR)
Amazon SageMaker
Spark
TensorFlow
Sessions dataset
Label sessions + feature engineering
Training dataset
Validation dataset
Testing dataset
Model training
Model evaluation
TensorFlow model
Performance report (TP, FP, FN, TN, precision, recall, F1)

Smaato ML framework provides:

smaato

Project structure for production environment | Fast feedback loops, fast development speed | Ability to invoke services that scale computation | Ability to focus on business problem (not plumbing)



smaato

Smaato ML Framework

jupyter

exploration.ipynb

APACHE Spark™

spark.py

TensorFlow

tensorflow.py

Local execution





Distributed execution

smaato

Smaato ML Framework

jupyter

exploration.ipynb

APACHE Spark™

spark.py

TensorFlow

tensorflow.py

Local execution





Distributed execution

```
        "range": [
-        -74.27088747797062,
-        -73.68075052202938
        ],
        "title": {
         "text": "LONGITUDE"

       }
      }
     },
-    "image/png":
"iVBORw0KGgoAAAANSUhEUgAABI8AAAHCCAYAAABmC+kXAAAgAElEQVR4Xuy9CbgcVZn//63qTnITshNIwpawSSK7gARcwqKssjqA4AiCoMLKssjqA4AiCCoMLCsygouJ3fXAAAAQSgJqm0Zu2E5pCZUAqV2AEEOAZz8p3fa/g0GBD8WLXKxGTmaSGBjrNJMQgUMBzRFgQOGvQ1nH7HfQ5J7z0Csc3uyjk5h+1ghp/zUYj+6JG8UFpaRmgPHxDBxRCSbtRcgg1kxQ/EC8PBJgGdtEvbQ8ugWsYM9SLQQIIDvj+WdPF0D6KJjMFTC8pkdzhXKOJgONAZBdzcSYv56fN0VdCqIb4WImTpyzNS3LD2ZP3pmyNJq
/7Hsf3fR8SIAESIAESIAESIAESIAESIAESIAESMBCwKE84rogARIgARIgARIgARIgARIgARIgARIgARJIIkB5xLVBAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRA
AAiRAAiRAAiRAAiRAAiSQSIDyiIuDBEiABEiABEiABEiABEiABEiABEiABEiABEiABEiABEiABEiA8ohrgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIoH4CzDyq
nxlHkAAJkAAJkAAJkAAJkAAJkAAJkAAJkEDLEKA8aplQ80FJgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIoH4ClEf1M+MIEiABEiABEiABEiABEiABEiABEiABE
iABEiABEiABEmgZApRHLRNqPigJkAAJkAAJkAAJkAAJkAAJkAAJkAAJkAAJkAAJkAAJ1E+A8qh+Zh+ZhxBAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1DgPKoZU
LNByUBEiABEiABEiABEiABEiABEiABEiCB+glQHtXPjCNIgARIgARIgARIgARIgARIgARIgARIgARIoGUIUB61TKj5oCRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiR
AAiRAAiRAAiRQPwHKo
/qZcQQJkAAJkAAJkAAJkAAJkAAJkAAJkAAJtAwByqOWCTUflARIgARIgARIgARIgARIgARIgARIgATqJ0B5VD8zjiABEiABEiABEiABEiABEi
ABEiABEiABEiCBliFAedQyoeaDkgAJkAAJkAAJkAAJkAAJkAAJkAAJkED9BCiP6mfGfESRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiTQMgQoj1o
m1HxQEiABEiABEiABEiABEiABEiABEiABEqifAOVR
/cw4ggRIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARAvUToDyqnxlHkAAJkAAJkAAJkAAJk
AAJkAAJkAAJkEDLEKA8aplQ80FJgARIgARIgARIgARIgARIgARIgARIoH4CzDyqnxlHkAAJkAAJkAAJkAAJkAAJkAAJkAAJkEDLEKA8aplQ80FJgARIgARIgARIgARIgARIgARIgARIoH4ClEf1M+MIEiABEiABEiABEiABEiABEiABEiABEmgZApRHLR
NqPigJkAAJkAAJkAAJkAAJkAAJkAAJkAAJ1E+A8qh+ZhxBAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1DgPKoZULNByUBEiABEiABEiABEiABEiABEiABEiCB+glQHtXPjCNIgARIgARIgARIgARIgARIgARIgARIgARIoGUIUB61TKj5oCRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRQPwHKo
/qZcQQJkAAJkAAJkAAJkAAJkAAJkAAJkAAJtAwByqOWCTUflARIgARIgARIgARIgARIgARIgARIgATqJ0B5VD8zjiABEiABEiABEiABEiABEiABEiABEiABEiCBliFAedQyoeaDkgAJkAAJkAAJkAAJkAAJkAAJkAAJkED9BCiP6mfGfESRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiTQMgQoj1om1HxQEiABEiABEiABEiABEiABEiABEiABEqifAOVR
/cw4ggRIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARIgARAvUToDyqnxlHkAAJkAAJkAAJkAAJkAAJkAAJkAAJkEDLEKA8aplQ80FJgARIgARIgARIgARIgARIgARIgARIoH4CzDyqnxlHkAAJkAAJkAAJkAAJkAAJkAAJkAAJkEDLEKA8aplQ80FJgARIgARIgARIgARIgARIgARIgARIoH4ClEf1M+MIEiABEiABEiABEiABEiABEiABEiABEmgZApRHLRNqPigJkAAJkAAJkAAJkAAJkAAJkAAJkAAJ1E+A8qh+ZhxBAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1DgPKoZULNByUBEiABEiABEiABEiABEiABEiABEiCB+glQHtXPjCNIgARIgARIgARIgARIgARIgARIgARIgARIoGUIUB61TKj5oCRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAiRQPwHKo
/qZcQQJkAAJkAAJkAAJkAAJkAAJkAAJkAAJtAwByqOWCTUflARIgARIgARIgARIgARIgARIgARIgATqJ0B5VD8zjiABEiABEiABEiABEiABEi
```



```
        "range": [
+        -74.2709047142857,
+        -73.6807332857143
        ],
        "title": {
         "text": "LONGITUDE"

       }
      }
     },
+    "image/png": "iVBORw0KGgoAAAANSUhEUgAABIgAAAHCCAYAAACE1/JuAAAgAElEQVR4XuydCbgcVZm
/f1Xdyb03JGQDsrAkQTSJCCTsoMOqLKJsozCgA4JK/qCC4IiKMzgy4zoDCDPAwIysIxhQFBAQISgQMIa9kQEErYsJOSGbDf7J7ar
/852qU326+1R331t3qer+1fPMkNyu5Zz3032l1Xr7vO47v047v+z54kAAJkAAJkAAJkAAJkAAJkAAJkAAJkAAJtCwBh4KoZWPPiZMACZZAACZAACZAACZAACZAACZAACZCAIkBBxIVAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1OgIKoxRcAp08CJEAfAXD6JEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEBBxDVAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1OgIKoxRcAp08CJEAfAXD6JEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEBBxDVAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1OgIKoxRcAp08CJEAfAXD6JEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEBBxDVAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1OgIKoxRcAp08CJEAfAXD6JEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEBBxDVAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1OgIKoxRcAp08CJEAfAXD6JEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEBBxDVAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1OgIKoxRcAp08CJEAfAXD6JEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEBBxDVAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1OgIKoxRcAp08CJEAfAXD6JEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEBBxDVAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1OgIKoxRcAp08CJEAfAXD6JEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEBBxDVAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1OgIKoxRcAp08CJEAfAXD6JEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEBBxDVAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1OgIKoxRcAp08CJEAfAXD6JEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEBBxDVAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1OgIKoxRcAp08CJEAfAXD6JEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEBBxDVAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1OgIKoxRcAp08CJEAfAXD6JEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEBBxDVAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1OgIKoxRcAp08CJEAfAXD6JEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEBBxDVAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1OgIKoxRcAp08CJEAfAXD6JEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEBBxDVAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1OgIKoxRcAp08CJEAfAXD6JEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEBBxDVAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1OgIKoxRcAp08CJEAfAXD6JEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEBBxDVAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1OgIKoxRcAp08CJEAfAXD6JEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEBBxDVAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1OgIKoxRcAp08CJEAfAXD6JEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEBBxDVAAiRAAiRAAiRAAiRAAiRAAiRAAiRAAi1OgIKoxRcAp08CJEAfAXD6JEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJEACJ
```

smaato

Smaato ML Framework

jupyter

exploration.ipynb

APACHE Spark™

spark.py

TensorFlow

tensorflow.py

Local execution

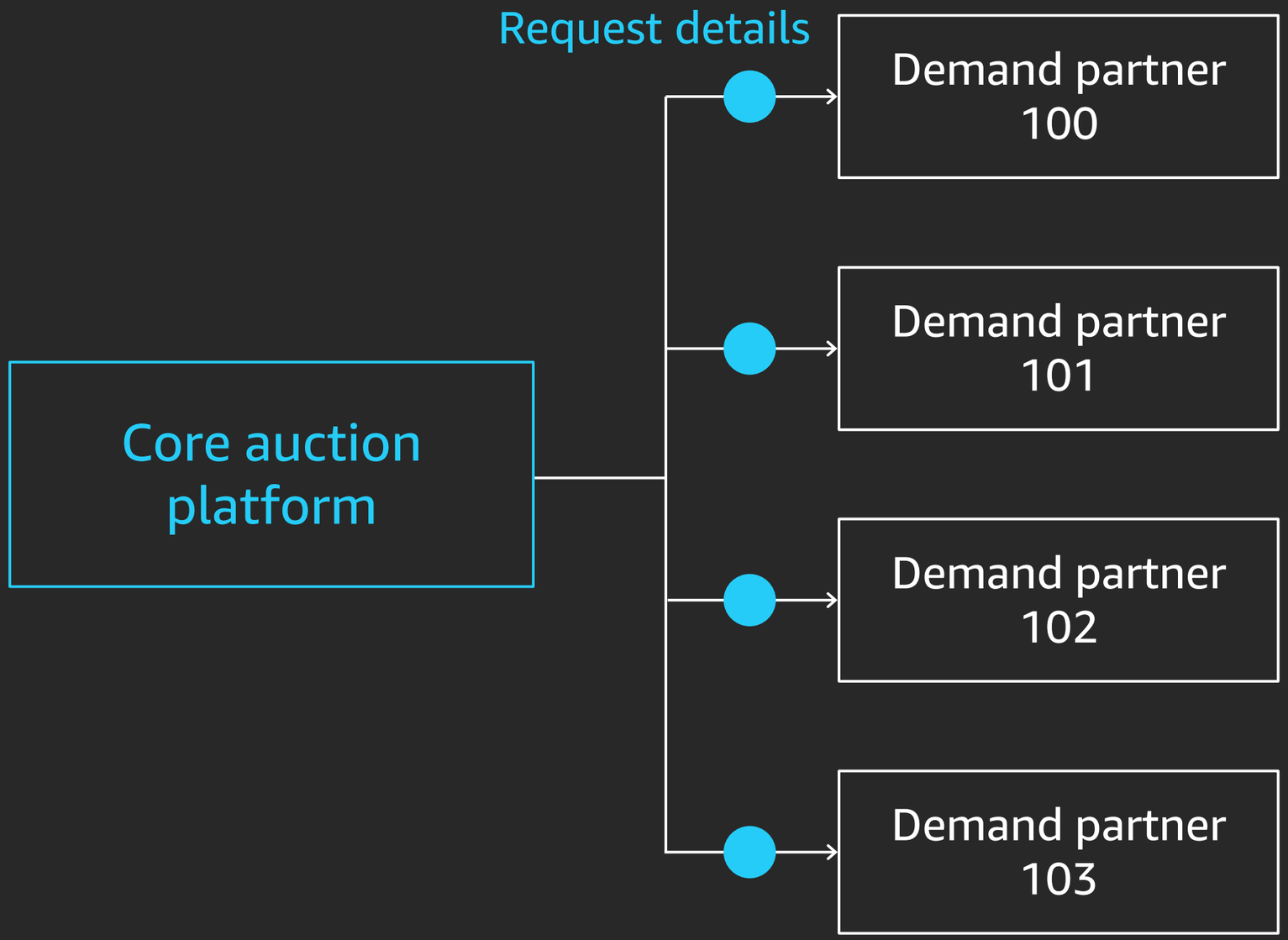




Distributed execution

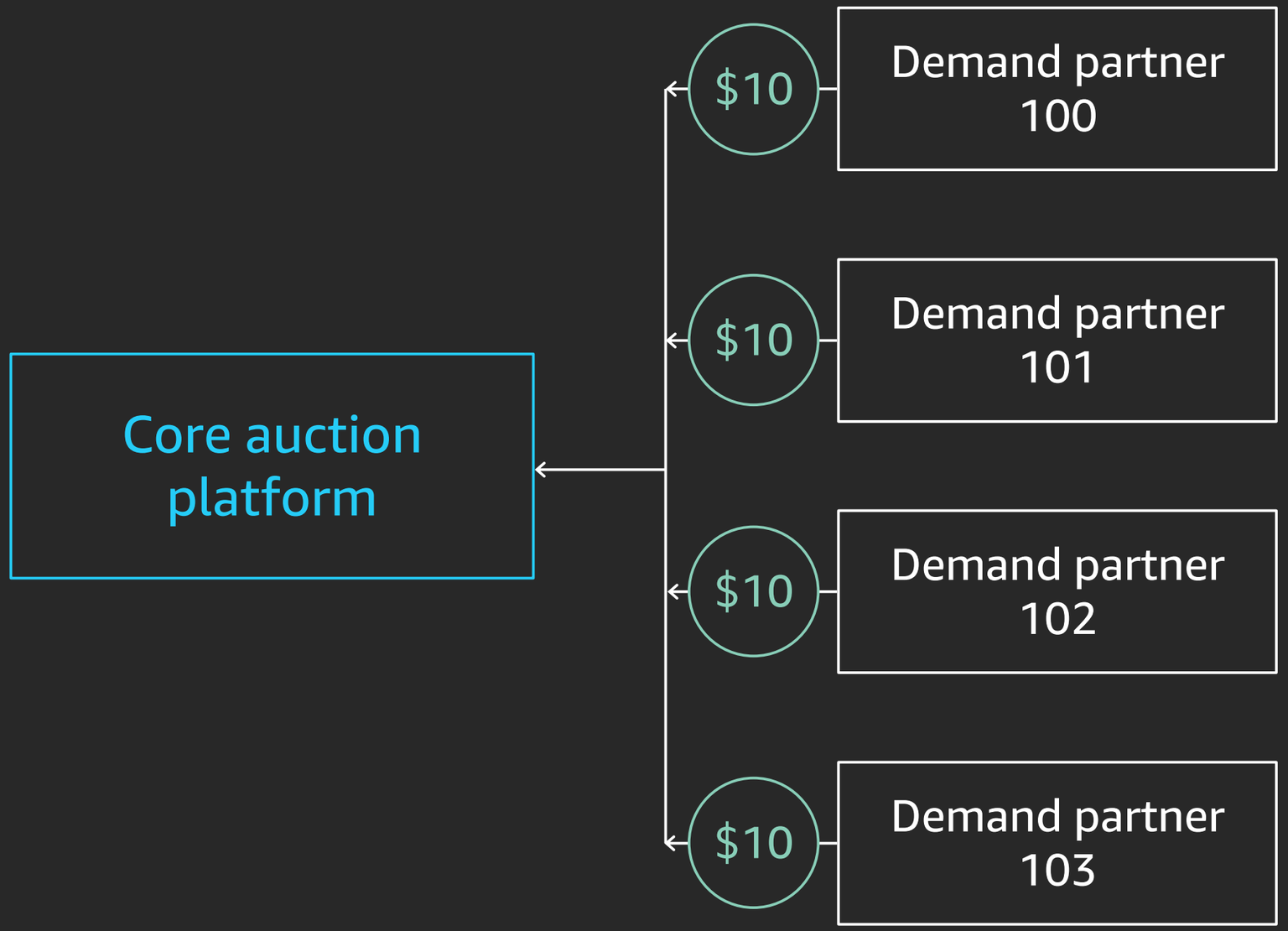
What did we do specifically?

smaato



What did we do specifically?

smaato

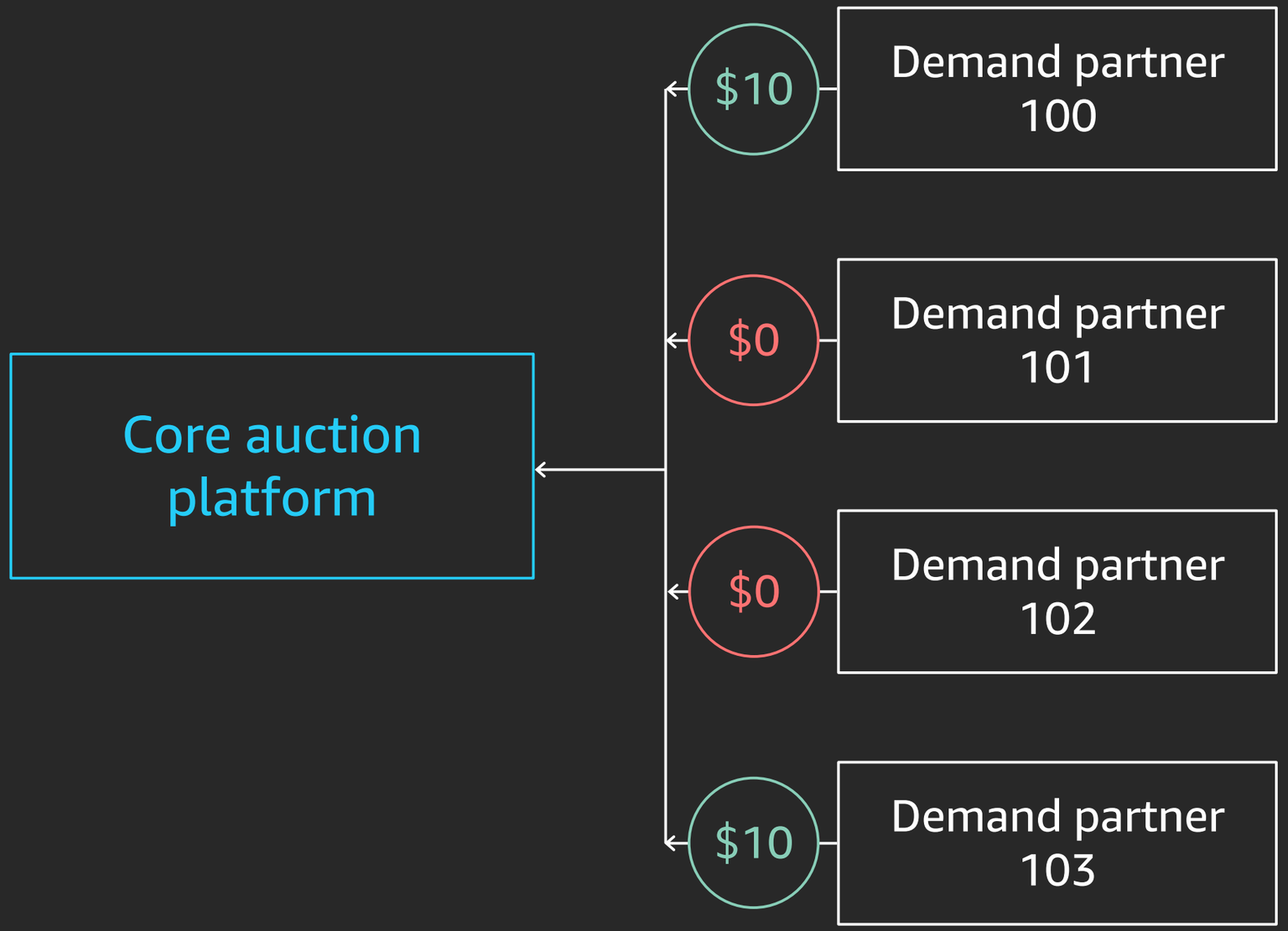


Session label



What did we do specifically?

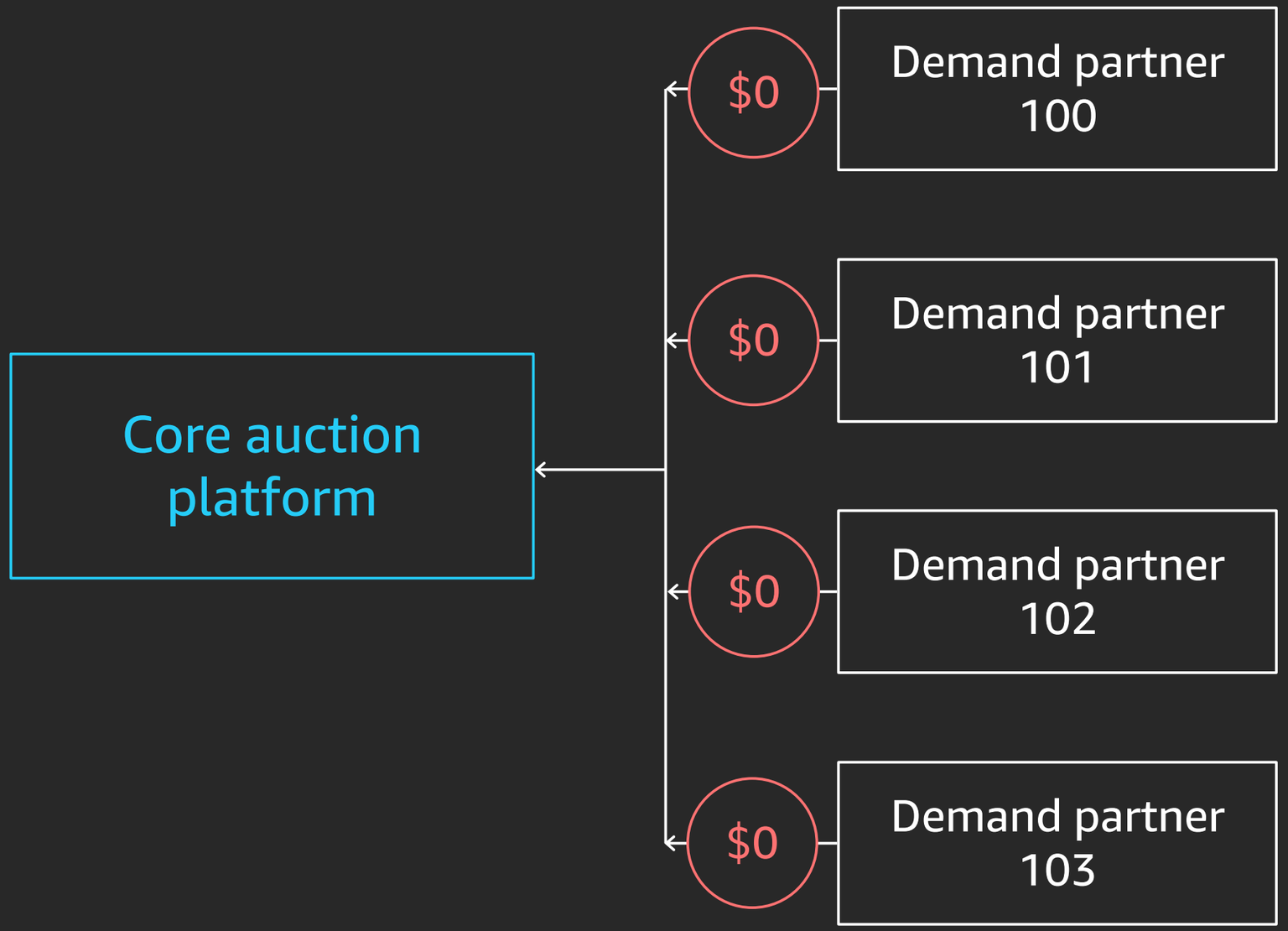
smaato



Session label



What did we do specifically?



Session label



What did we do specifically?

smaato

Continuous

- Request time

Low Cardinality

- Country code
- Connection type
- Device model
- Device brand
- Advertisement type
- Adspace resolution

High Cardinality

- Publisher ID
- Adspace ID
- Application ID

What did we do specifically?

smaato

Time extraction

- Extract day of week
- Extract hour of day

OneHot Encoding

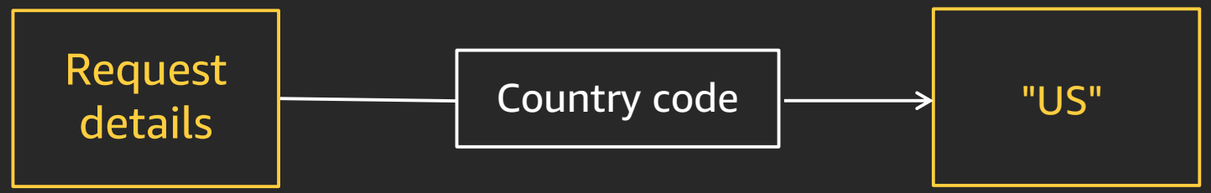
- Country code
- Connection type
- Device model
- Device brand
- Advertisement type
- Adspace resolution

Feature Hashing

- Publisher ID
- Adspace ID
- Application ID

OneHot Encoding example

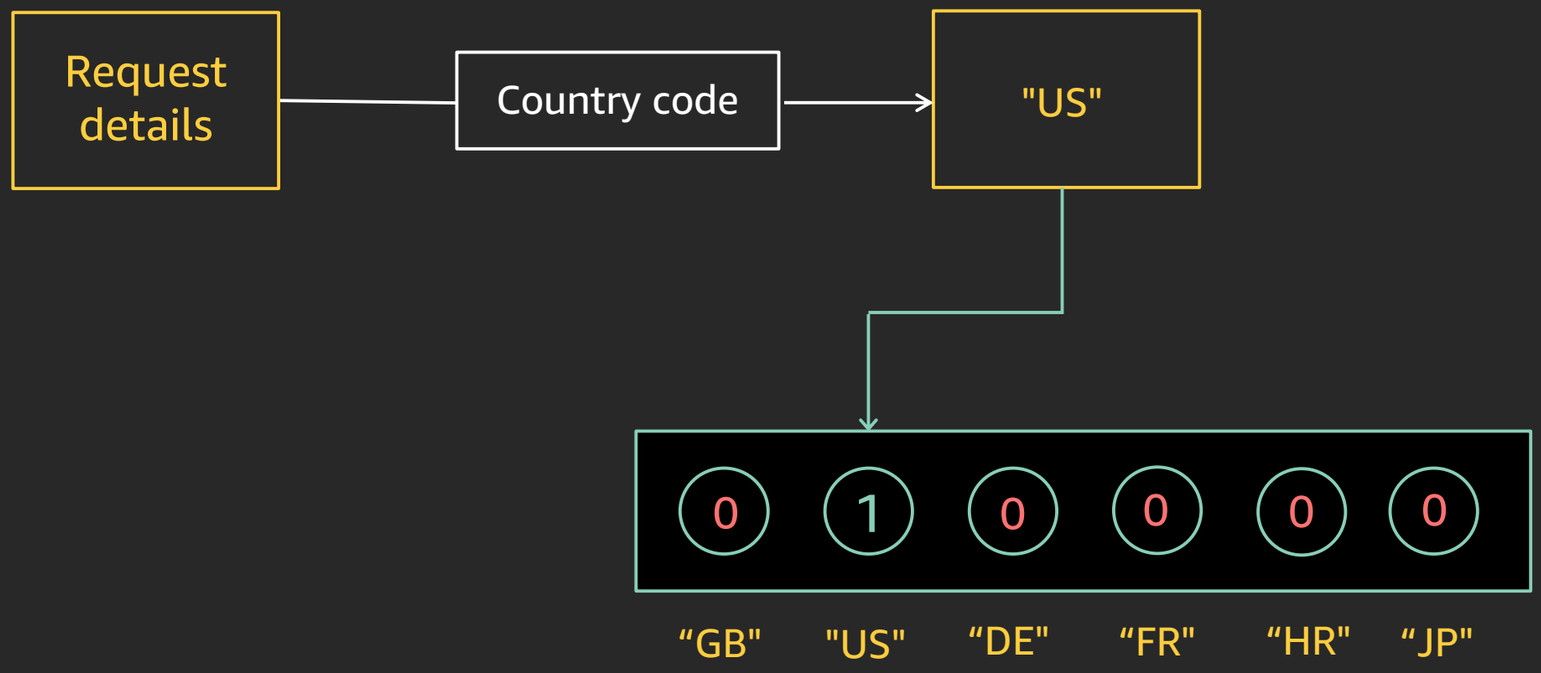
smaato





OneHot Encoding example

smaato



Feature Hashing example

smaato



smaato

Smaato ML Framework

jupyter

exploration.ipynb

APACHE Spark™

spark.py

TensorFlow

tensorflow.py

Local execution

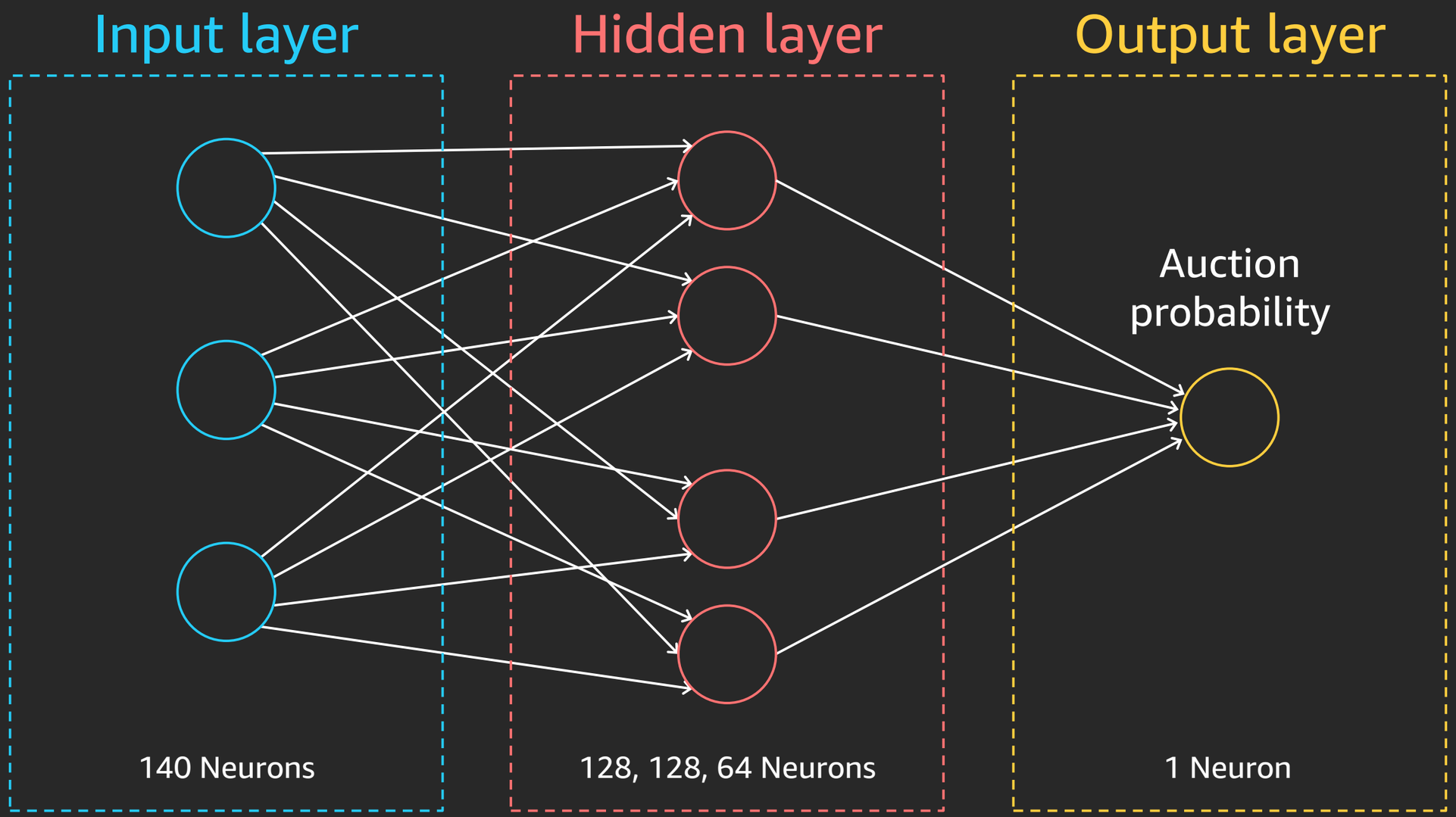




Distributed execution

What did we do specifically?

smaato



Why we chose Amazon EMR and SageMaker

smaato



Managed AWS service

Effortlessly scalable

Optimized infrastructure

Demo

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

x-terminal-emulator

[15:43:39] SM393 /home/janko/PyProjects/smart-platform-sagemaker > _ [STF/develop]

[shell] 1.zsh

54% - SMTHH 1000 Mbit/s 40% 100% 96% Friday, 29 November - 15:43:41

```
159 def load_datasets(config):...
168
169
170 def label_sessions(df: DataFrame):...
181
182
183 def engineer_features(df_train: DataFrame, df_valid: DataFrame, df_test: DataFrame):...
224
225
226 def save_datasets(config, df_train, df_valid, df_test, meta_data):...
245
246
247 def run():
248     config = parse_cli_arguments()
249     initialize_spark_context()
250
251     df_train, df_valid, df_test = load_datasets(config)
252
253     df_train = label_sessions(df_train)
254     df_valid = label_sessions(df_valid)
255     df_test = label_sessions(df_test)
256
257     df_train, df_valid, df_test, meta_data = engineer_features(df_train, df_valid, df_test)
258
259     save_datasets(config, df_train, df_valid, df_test, meta_data)
260
261
262 if __name__ == '__main__':
263     run()
264
```

```
159 def load_datasets(config):...
168
169
170 def label_sessions(df: DataFrame):...
181
182
183 def engineer_features(df_train: DataFrame, df_valid: DataFrame, df_test: DataFrame):...
224
225
226 def save_datasets(config, df_train, df_valid, df_test, meta_data):...
245
246
247 def run():
248     config = parse_cli_arguments()
249     initialize_spark_context()
250
251     df_train, df_valid, df_test = load_datasets(config)
252
253     df_train = label_sessions(df_train)
254     df_valid = label_sessions(df_valid)
255     df_test = label_sessions(df_test)
256
257     df_train, df_valid, df_test, meta_data = engineer_features(df_train, df_valid, df_test)
258
259     save_datasets(config, df_train, df_valid, df_test, meta_data)
260
261
262 if __name__ == '__main__':
263     run()
264
```


x-terminal-emulator
[16:02:19] SM393 /home/janko/PyProjects/smart-platform-sagemaker > _
[STF/develop]
[shell] 1.zsh
64% - SMTHH 1000 Mbit/s 45% 100% 96% Friday, 29 November - 16:03:16

```
158
159 def load_datasets(config):...
168
169
170 def label_sessions(df: DataFrame):...
181
182
183 def engineer_features(df_train: DataFrame, df_valid: DataFrame, df_test: DataFrame):...
224
225
226 def save_datasets(config, df_train, df_valid, df_test, meta_data):...
245
246
247 def run():
248     config = parse_cli_arguments()
249     initialize_spark_context()
250
251     df_train, df_valid, df_test = load_datasets(config)
252
253     df_train = label_sessions(df_train)
254     df_valid = label_sessions(df_valid)
255     df_test = label_sessions(df_test)
256
257     df_train, df_valid, df_test, meta_data = engineer_features(df_train, df_valid, df_test)
258
259     save_datasets(config, df_train, df_valid, df_test, meta_data)
260
261
262 if __name__ == '__main__':
263     run()
```

```
100 @with_failure_traceback(log, propagate=is_executed_on_sagemaker())
101 def run():
102     config = parse_cli_arguments()
103     meta_data = read_meta_data(config)
104
105     df_train, df_valid, df_test = load_datasets(config, meta_data)
106     model = build_model(meta_data)
107
108     callbacks = []
109     if on_master_node():...
114
115     log.info(
116         f"Start model training (epochs: {config['tensorflow.epochs']}, batch_size: {config['tensorflow.batch_size']}, "
117         f"batch per epoch: {meta_data['train_size'] // config['tensorflow.batch_size']}, "
118         f"train_size: {meta_data['train_size']}, valid_size: {meta_data['valid_size']})")
119
120     model.fit(
121         *df_train,
122         epochs=config['tensorflow.epochs'],
123         steps_per_epoch=meta_data['train_size'] // config['tensorflow.batch_size'],
124         validation_data=df_valid,
125         validation_steps=meta_data['valid_size'] // config['tensorflow.batch_size'],
126         verbose=config['tensorflow.keras_verbosity'],
127         callbacks=callbacks)
128
129     if on_master_node():
130         log.info(f'Starting model evalution on test dataset...')
131         evaluate_model(model, ds=df_test, step_size=1)
132         save_model(model=model)
133
```

```
100 @with_failure_traceback(log, propagate=is_executed_on_sagemaker())
101 def run():
102     config = parse_cli_arguments()
103     meta_data = read_meta_data(config)
104
105     df_train, df_valid, df_test = load_datasets(config, meta_data)
106     model = build_model(meta_data)
107
108     callbacks = []
109     if on_master_node():...
114
115     log.info(
116         f"Start model training (epochs: {config['tensorflow.epochs']}, batch_size: {config['tensorflow.batch_size']}, "
117         f"batch per epoch: {meta_data['train_size'] // config['tensorflow.batch_size']}, "
118         f"train_size: {meta_data['train_size']}, valid_size: {meta_data['valid_size']})")
119
120     model.fit(
121         *df_train,
122         epochs=config['tensorflow.epochs'],
123         steps_per_epoch=meta_data['train_size'] // config['tensorflow.batch_size'],
124         validation_data=df_valid,
125         validation_steps=meta_data['valid_size'] // config['tensorflow.batch_size'],
126         verbose=config['tensorflow.keras_verbosity'],
127         callbacks=callbacks)
128
129     if on_master_node():
130         log.info(f'Starting model evalution on test dataset...')
131         evaluate_model(model, ds=df_test, step_size=1)
132         save_model(model=model)
133
```

```
182
183 def engineer_features(df_train: DataFrame, df_valid: DataFrame, df_test: DataFrame):...
224
225
226 def save_datasets(config, df_train, df_valid, df_test, meta_data):...
245
246
247 def run():
248     config = parse_cli_arguments()
249     initialize_spark_context()
250
251     df_train, df_valid, df_test = load_datasets(config)
252
253     df_train = label_sessions(df_train)
254     df_valid = label_sessions(df_valid)
255     df_test = label_sessions(df_test)
256
257     df_train, df_valid, df_test, meta_data = engineer_features(df_train, df_valid, df_test)
258
259     save_datasets(config, df_train, df_valid, df_test, meta_data)
260
261
262 if __name__ == '__main__':
263     run()
264
```

```
182
183 def engineer_features(df_train: DataFrame, df_valid: DataFrame, df_test: DataFrame):...
224
225
226 def save_datasets(config, df_train, df_valid, df_test, meta_data):...
245
246
247 def run():
248     config = parse_cli_arguments()
249     initialize_spark_context()
250
251     df_train, df_valid, df_test = load_datasets(config)
252
253     df_train = label_sessions(df_train)
254     df_valid = label_sessions(df_valid)
255     df_test = label_sessions(df_test)
256
257     df_train, df_valid, df_test, meta_data = engineer_features(df_train, df_valid, df_test)
258
259     save_datasets(config, df_train, df_valid, df_test, meta_data)
260
261
262 if __name__ == '__main__':
263     run()
264
```

```
config = dict()

config['spark.sessionizer.train
config['spark.sessionizer.valid
config['spark.sessionizer.test'
# config['spark.sessionizer.train
# config['spark.sessionizer.valid
# config['spark.sessionizer.test'
config['spark.repartition'] = 6

config['tensorflow.source_directory'] = 's3://smart-platform-test/users/janko/traffic-filtering'
# config['tensorflow.source_directory'] = 'engineered_datasets/'
config['tensorflow.model.output'] = 'models/'
config['tensorflow.epochs'] = 4
config['tensorflow.batch_size'] = 50
config['tensorflow.log_batch_freq'] = 1000
config['tensorflow.learning_rate'] = 0.01
config['tensorflow.optimizer'] = 'adam'
config['tensorflow.default_threshold'] = 0.5
config['tensorflow.keras_verbosity'] = 0  # 0 = silent, 1 = progress bar, 2 = one line per epoch.

config['emr.cluster_name'] = 'Prometheus-PySpark-On-EMR-jk'
config['emr.project_location'] = 's3://smart-platform-test/users/janko/emr'
```

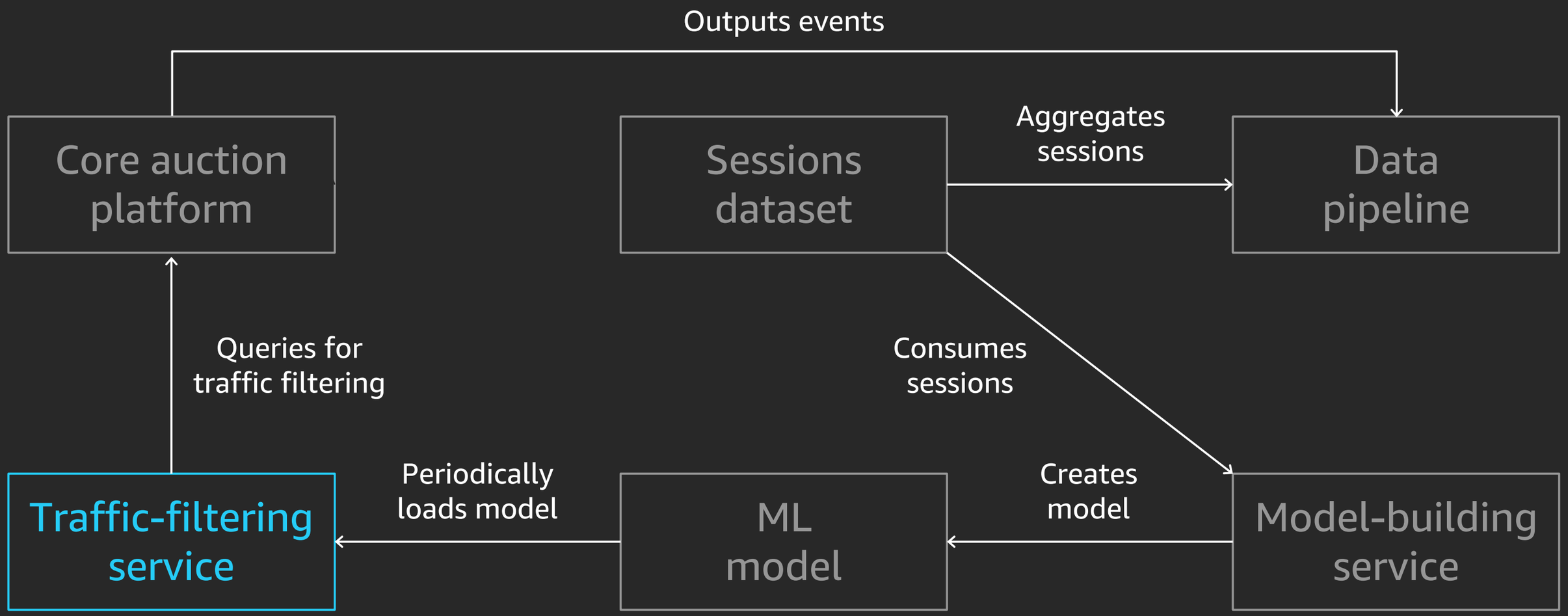
Demo

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Integration with traffic-filtering service

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

smaato


Outputs events
Core auction platform
Sessions dataset
Aggregates sessions
Data pipeline
Queries for traffic filtering
Consumes sessions
Traffic-filtering service
Periodically loads model
ML model
Creates model
Model-building service

Traffic filtering service

smaato

Hosts models for traffic filtering

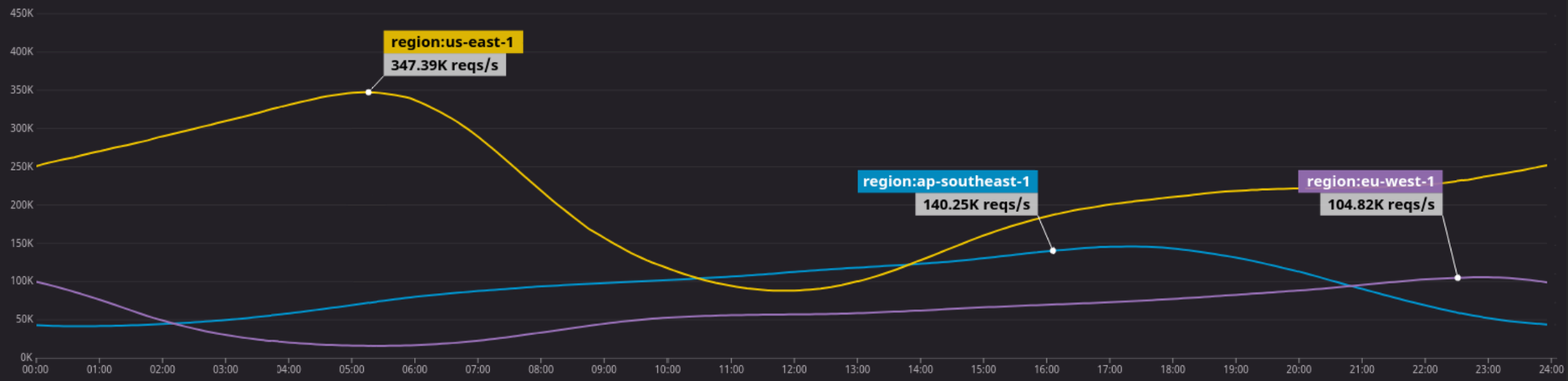
Transforms ad requests

Abides by business rules



smaato

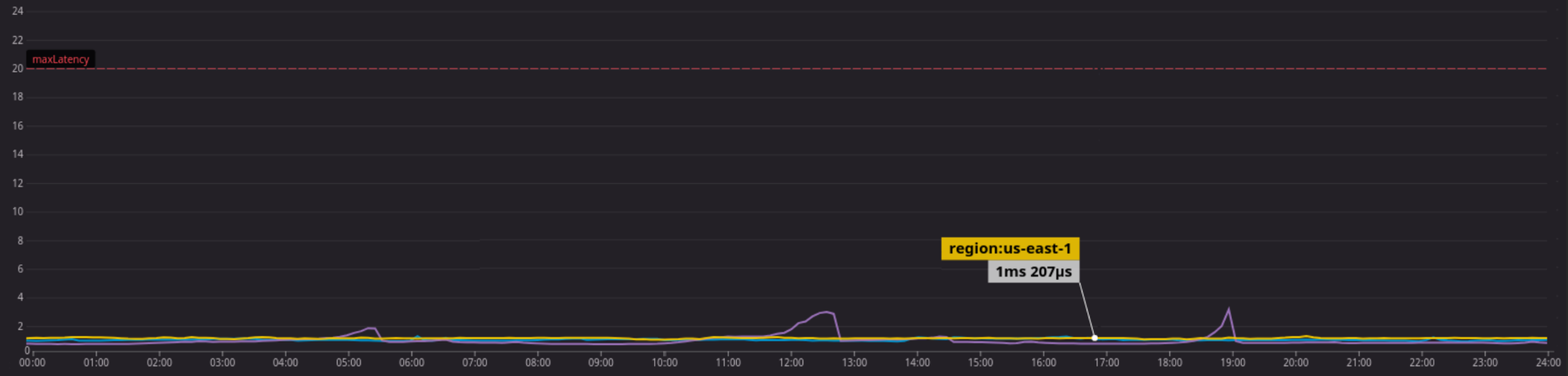
Incoming ad requests [req/sec]



Hosts per region

US	10–50
EU	5–15
APAC	5–20

Traffic filtering service latency [ms]



Incoming ad requests per host

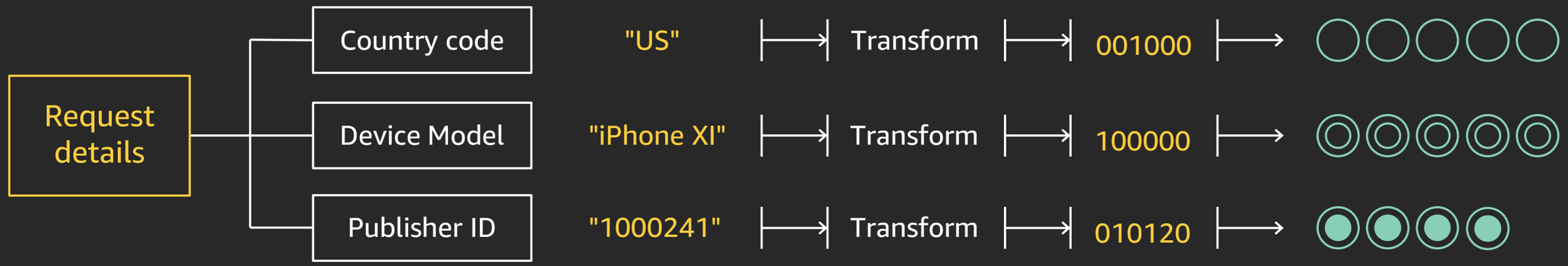
10,000–15,000

Friday, 20th of September

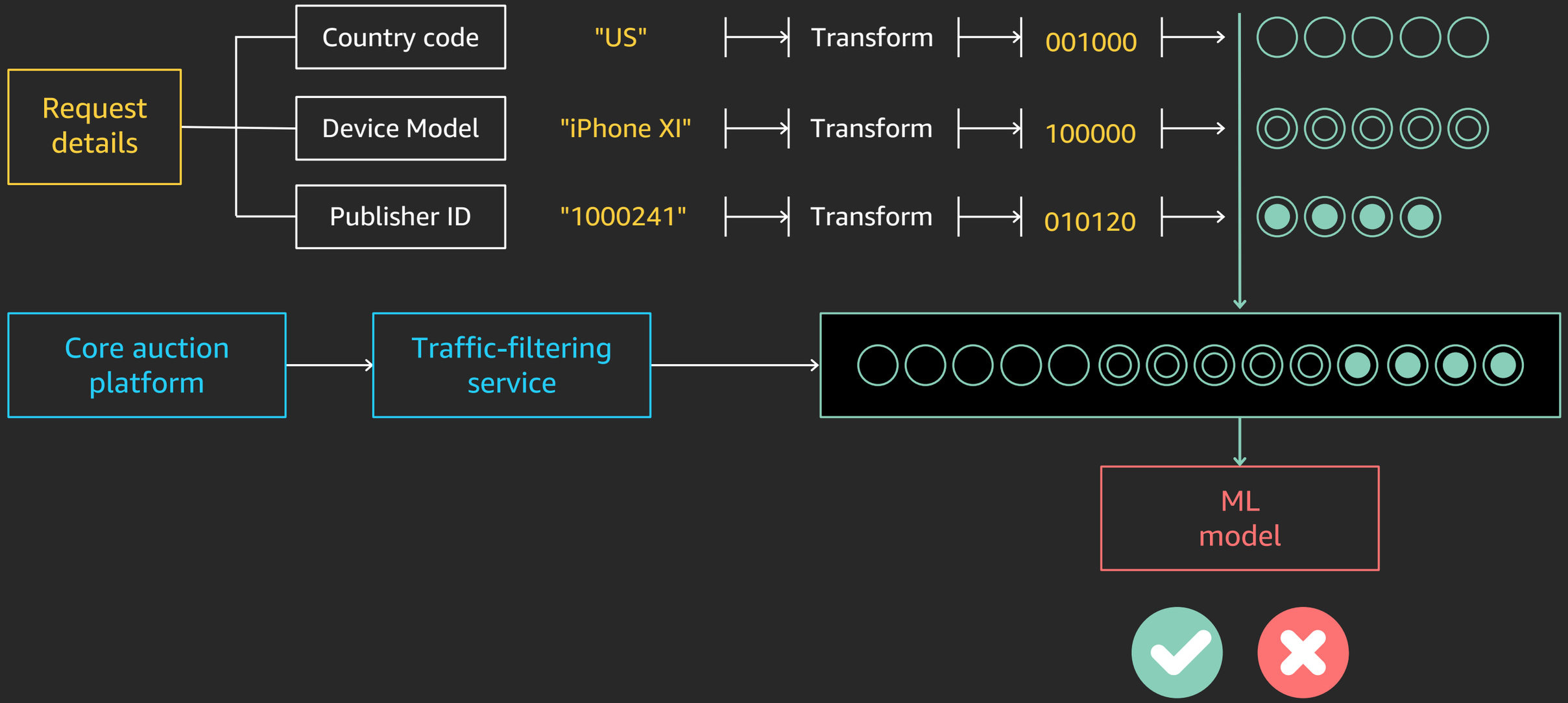
Model integration scenarios

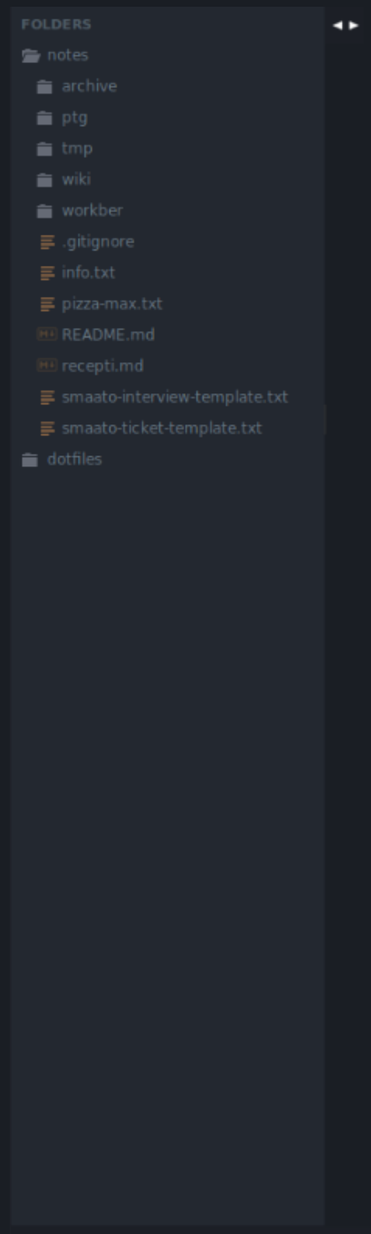
smaato

Model building	Model inference
Python	Python - offline
Python	Python - online
Python	Java / C# / JS - online


Request details
Country code
Device Model
Publisher ID
"US"
Transform
001000
"iPhone XI"
Transform
100000
"1000241"
Transform
010120

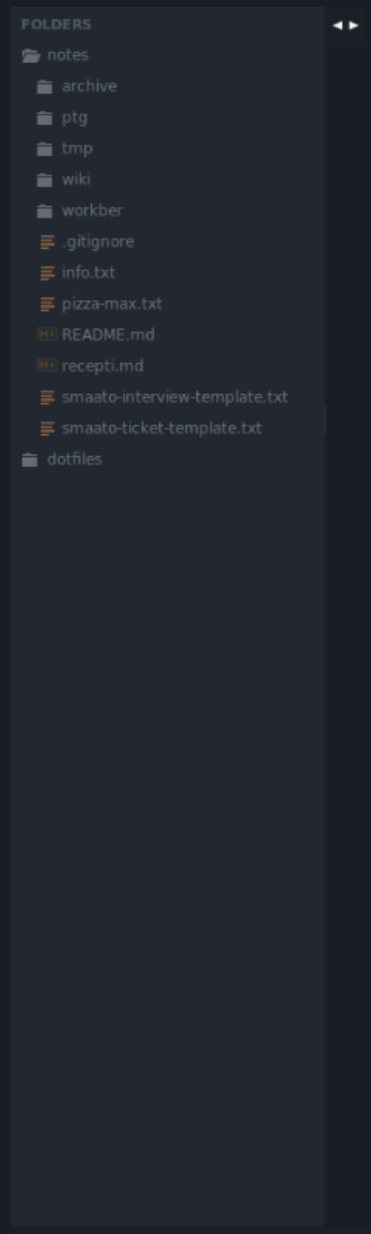
smaato


Request details
Country code
Device Model
Publisher ID
"US"
"iPhone XI"
"1000241"
Transform
Transform
Transform
001000
100000
010120
Core auction platform
Traffic-filtering service
ML model



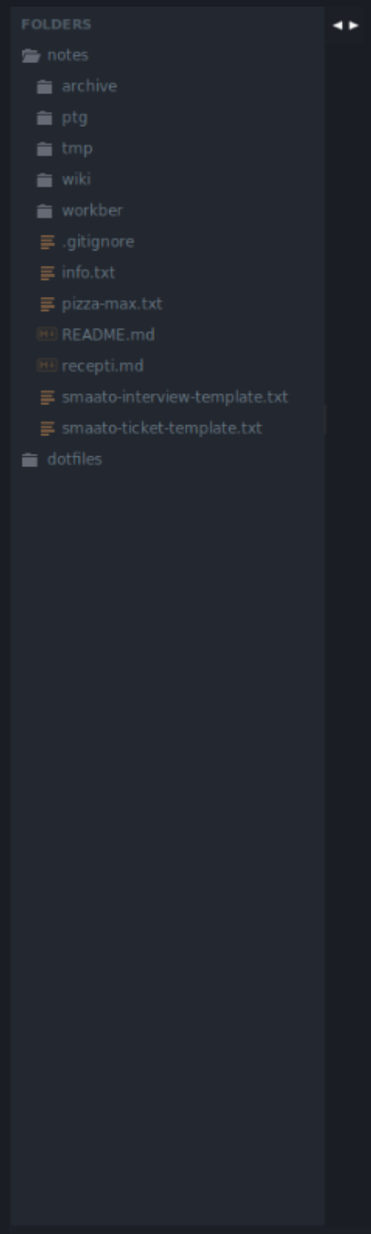
Feature meta data

```
{
    "train_data": "year=2019, month=9, day=10, hour=10, delta=-24",
    "valid_data": "year=2019, month=9, day=10, hour=11, delta=0",
    "test_data": "year=2019, month=9, day=10, hour=12, delta=0",
    “train_size": 5,403,718,092,
    "valid_size": 21,958,208,
    "test_size": 23,482,946,
    "model_input_size": 140,
    "transformations" : [
        (...)
    ]
}
```



Feature hashing transformation

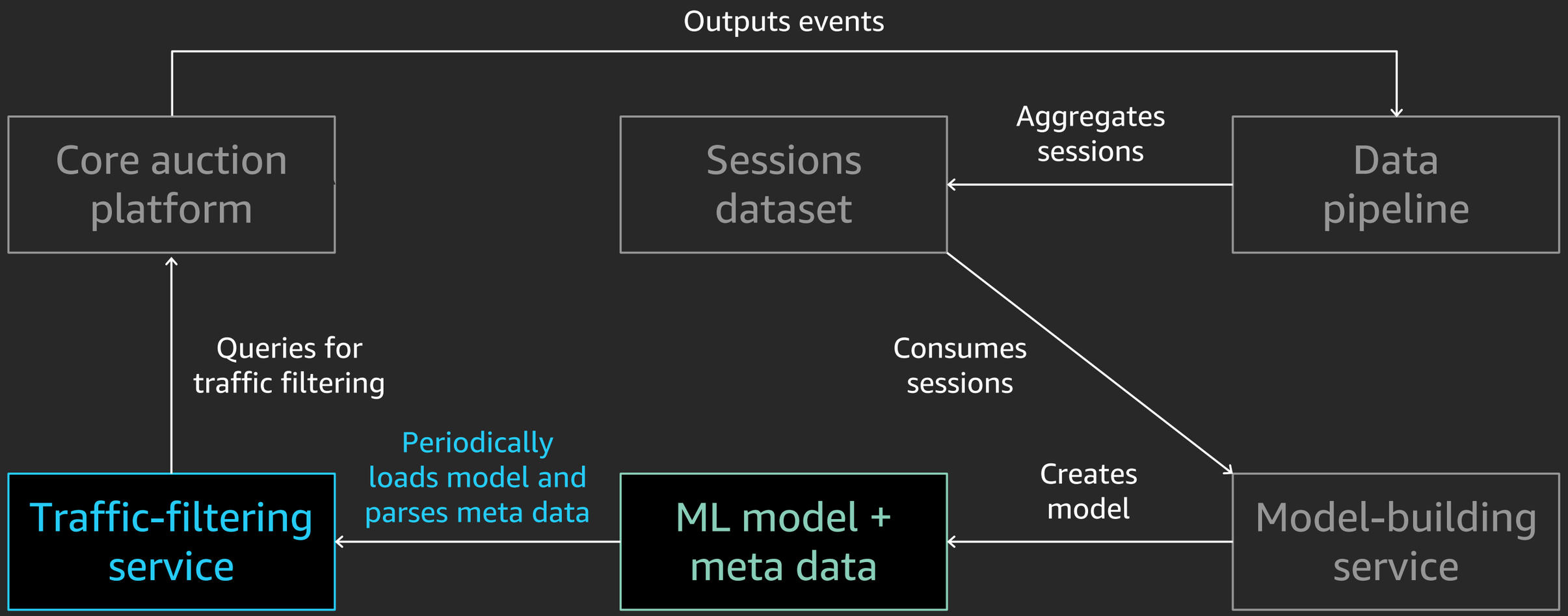
```
{
    ...
    "transformations" : [
        {
            "name": "hashing",
            "features": [ "adspaceId", "publisherId", "applicationId“ ],
            "index_start": 0,
            "index_end": 8
        },
    ]
}
```



OneHot encoding transformation

```
{
    ...
    "transformations" : [
        {
            "name": "onehot",
            "features": [ "ccod" ],
            “mapping”: { "us": 0, "in": 1, "de": 2, ..., "hr": 40, "jp": 41 }
            "index_start": 8,
            "index_end": 50
        },
    ]
}
```

smaato


Outputs events
Core auction platform
Sessions dataset
Aggregates sessions
Data pipeline
Queries for traffic filtering
Consumes sessions
Periodically loads model and parses meta data
Creates model
Traffic-filtering service
ML model + meta data
Model-building service

Results

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Traffic filtering improvements

smaato

Additional 15% sessions blocked

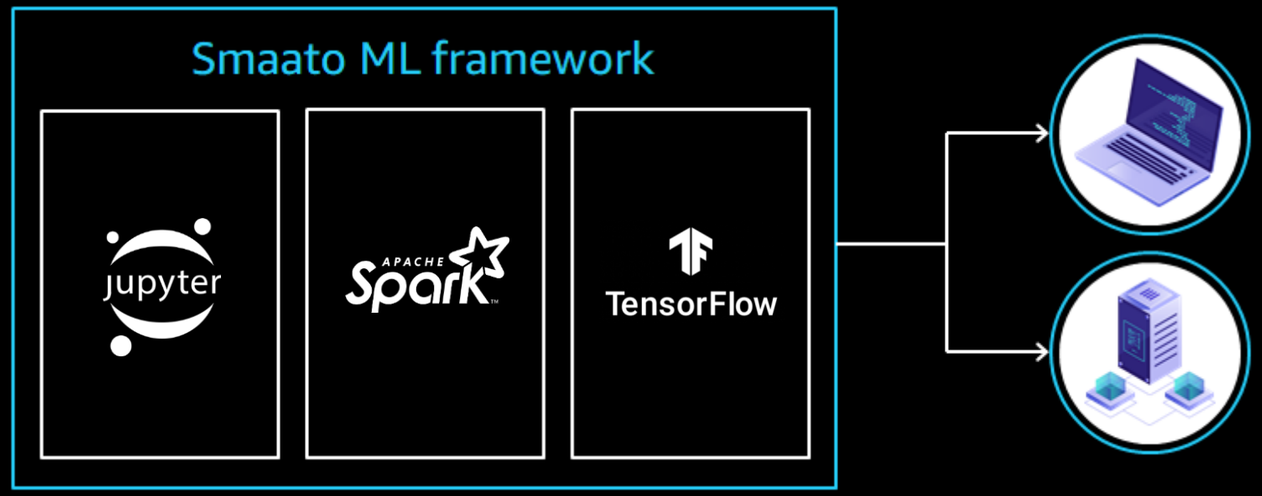
Infrastructure cost greatly reduced

No negative impact on revenue



Engineering improvements

smaato



Significantly improved speed to market

Lowered the barrier of entry for ML engineers

Completely streamlined development workflow

Thank you!

Dmitri Tchikatilov
dmitrit@amazon.com

François Guerraz
LinkedIn: francoisguerraz

Janko Kovačević
LinkedIn: jankovacevic

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws



Please complete the session survey in the mobile app.

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.