Prepare data for ML with ease, speed, and accuracy

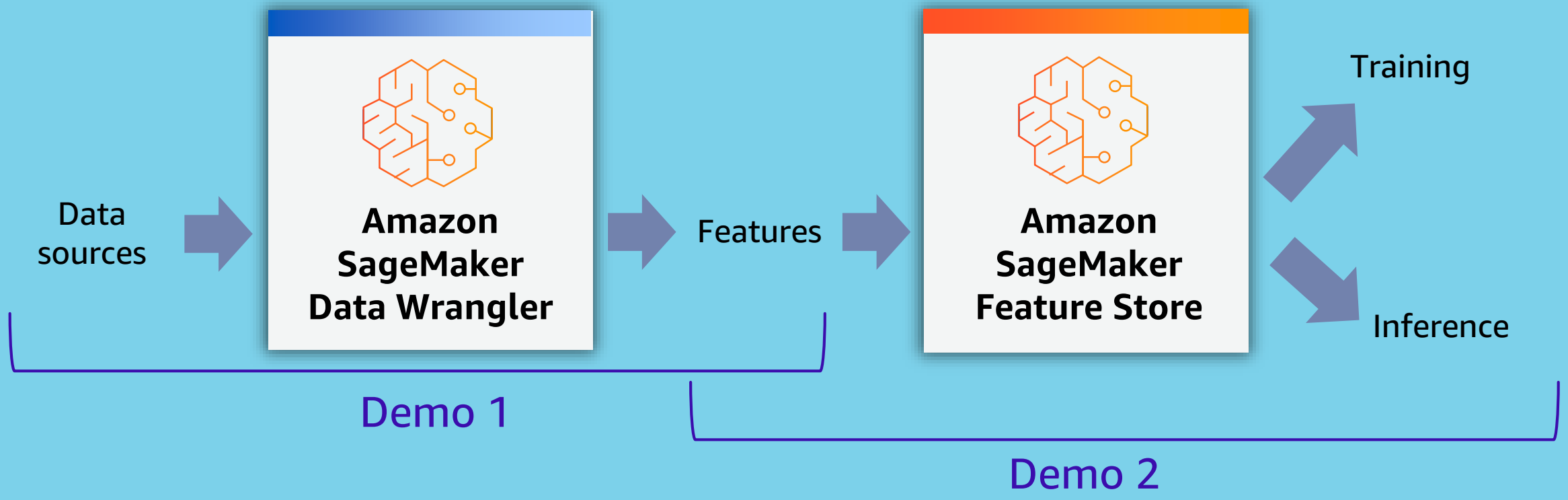
Mark Roy

Principal ML Architect

Agenda

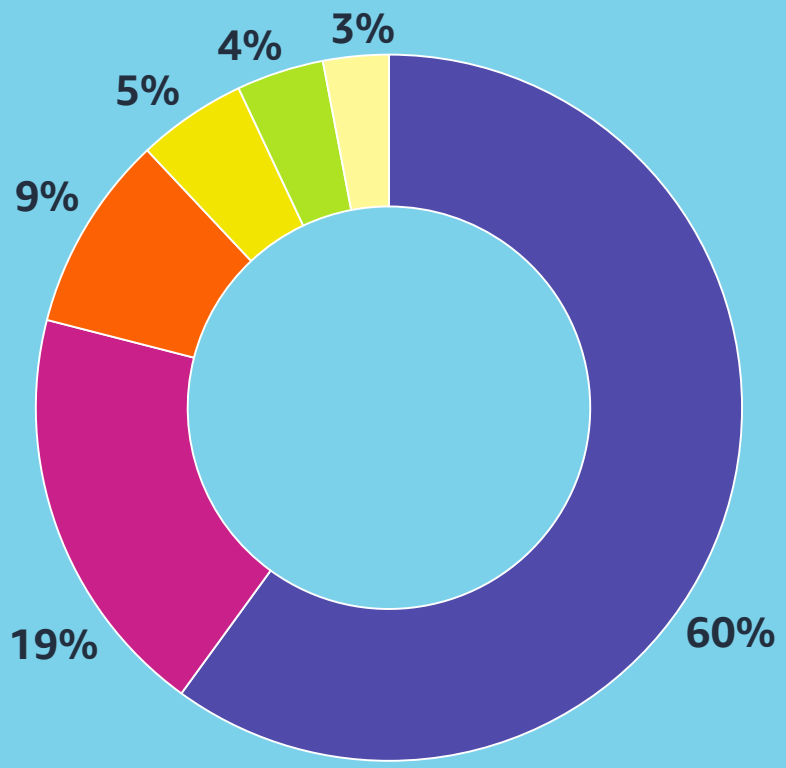
Overview of SageMaker Data Wrangler and Feature Store

Deep dive and demos



Overview of SageMaker Data Wrangler and SageMaker Feature Store

60% of time spent on data prep



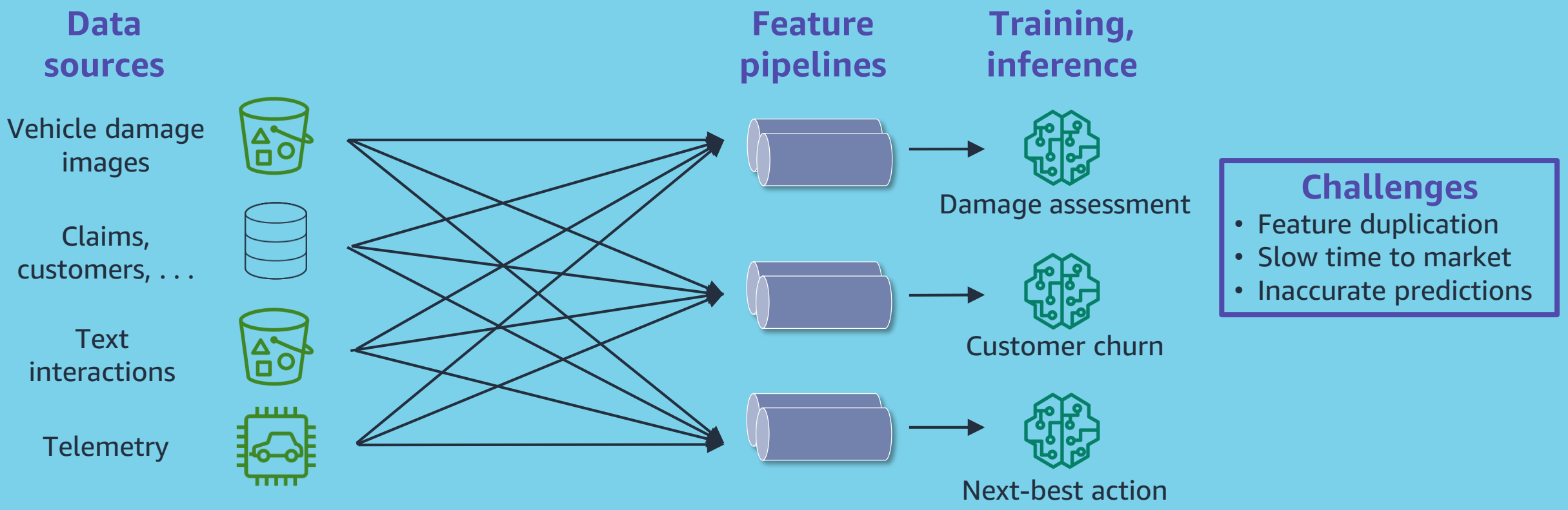
What data scientists spend the most time doing

- Cleaning and organizing data
- Collecting data sets
- Mining data for patterns
- Other
- Refining algorithms
- Building training sets

Source: Forbes survey of 80 data scientists, March 2016

… and teams typically start from scratch

Standalone feature engineering for each new model



Challenges

- Feature duplication
- Slow time to market
- Inaccurate predictions

So, how does SageMaker help?

Amazon SageMaker Data Wrangler

The fastest and easiest way to prepare data for machine learning

Quickly select and query data

Support for data from multiple sources

Easily transform data with built-in transformations

Use built-in data transformations to covert raw data to features for machine learning

Customize data transformations

Complete flexibility to bring your own custom transformations in PySpark, SQL, or Pandas

Understand data visually

Quickly detect outliers or extreme values – all without writing code

Quickly estimate machine learning model accuracy

Diagnose potential issues in data preparation workflows that could hinder machine learning model accuracy

Easily deploy data preparation workflows

Manage all steps of the data preparation workflow through a single visual interface to quickly operationalize workflows into production settings

Amazon SageMaker Feature Store

Securely store, discover, and share features for both real-time inference and training

Batch and streaming ingestion

High throughput writes for ingesting features

Online and offline features

Online features for real-time prediction, and offline features for historical data for model training and batch prediction

Feature metadata and data cataloging

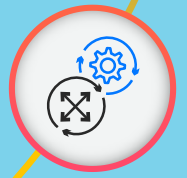
Store metadata for features and leverage automatic data cataloging to easily query and extract feature data

Feature discovery and reuse

Search for feature discovery

Security and access control

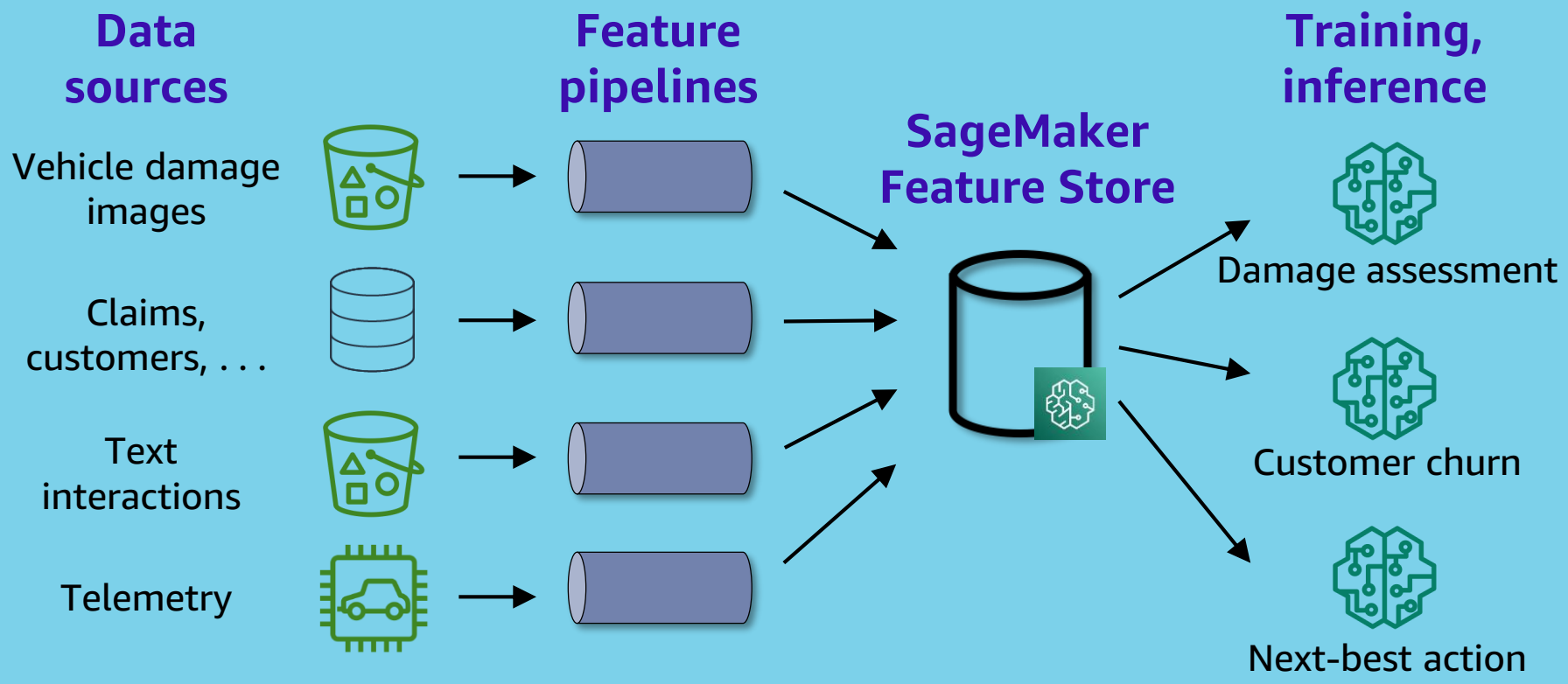
Access control for feature data and feature metadata, and support for encryption at rest, Amazon VPC, and AWS PrivateLink



Fully managed

Online features cached in low-latency store; maintain consistency between online and offline store to avoid train-infer skew

Build features once and reuse them



Benefits

- Feature reuse
- Reproducible features
- Accurate training datasets
- Low latency inference
- Consistent features for training and inference

Flight delay prediction

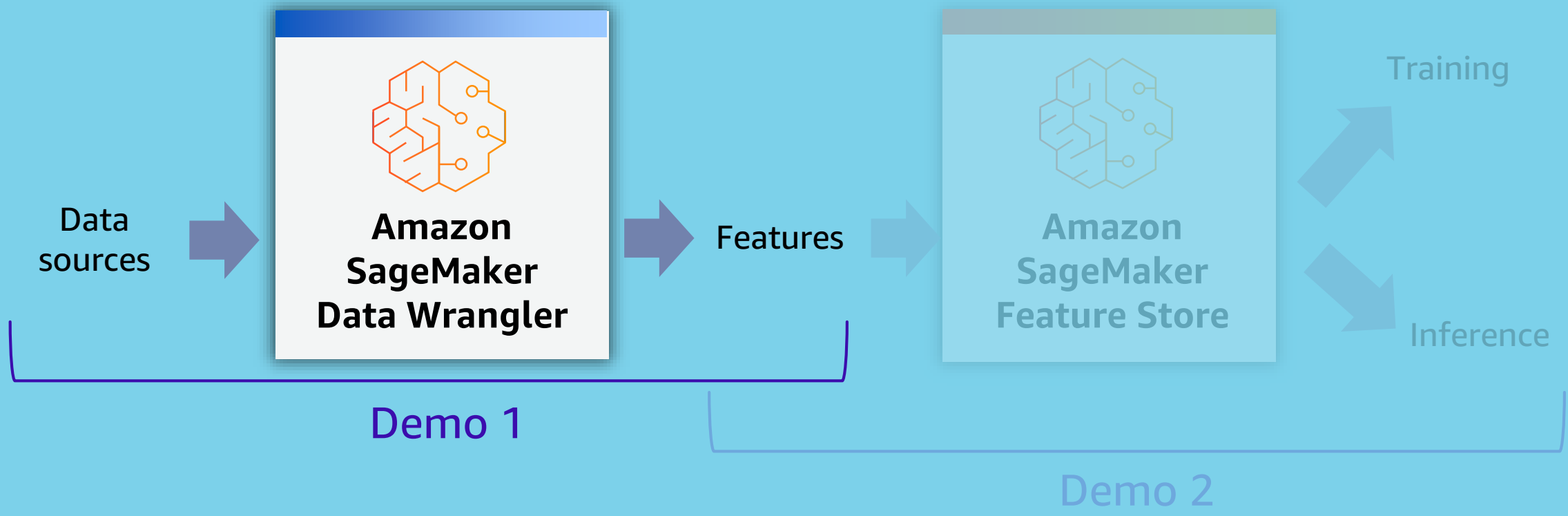
Raw flight delay historical data

Flight	Date	Origin	Destination	…	Delayed more than 15 minutes?
AA329	2020-02-01	BOS	SFO	…	1 (yes)
AA329	2020-02-02	BOS	SFO	…	0 (no)
…	…	…	…	…	…
SW85	2020-03-30	LAX	SEA	…	1 (yes)

3 months of US flights,
1.7 million rows, 26 columns

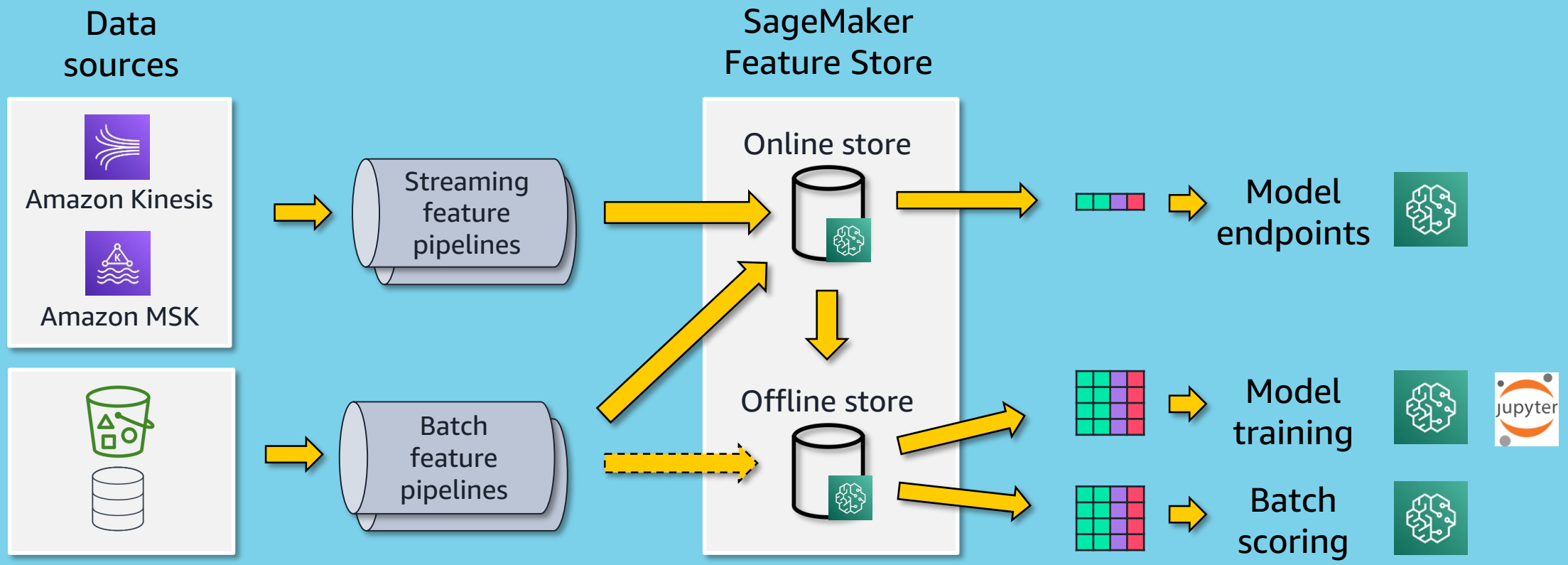
Will my flight be delayed by more than 15 minutes?

Demo 1: SageMaker Data Wrangler



Deep dive on feature ingestion

Feature store in context



Feature pipeline options

Key implementation considerations

- Existing ecosystem
- Data scientist skillset
- ML engineering capacity
- Scale

Feature ingestion API's

PutRecord API

```
record = [{'FeatureName': 'feature_1', 'ValueAsString': 'val_1'},
                    …
          {'FeatureName': 'feature_N', 'ValueAsString': 'val_N'}]

sm_fs.put_record(FeatureGroupName='my-fg-name', Record=record)
```

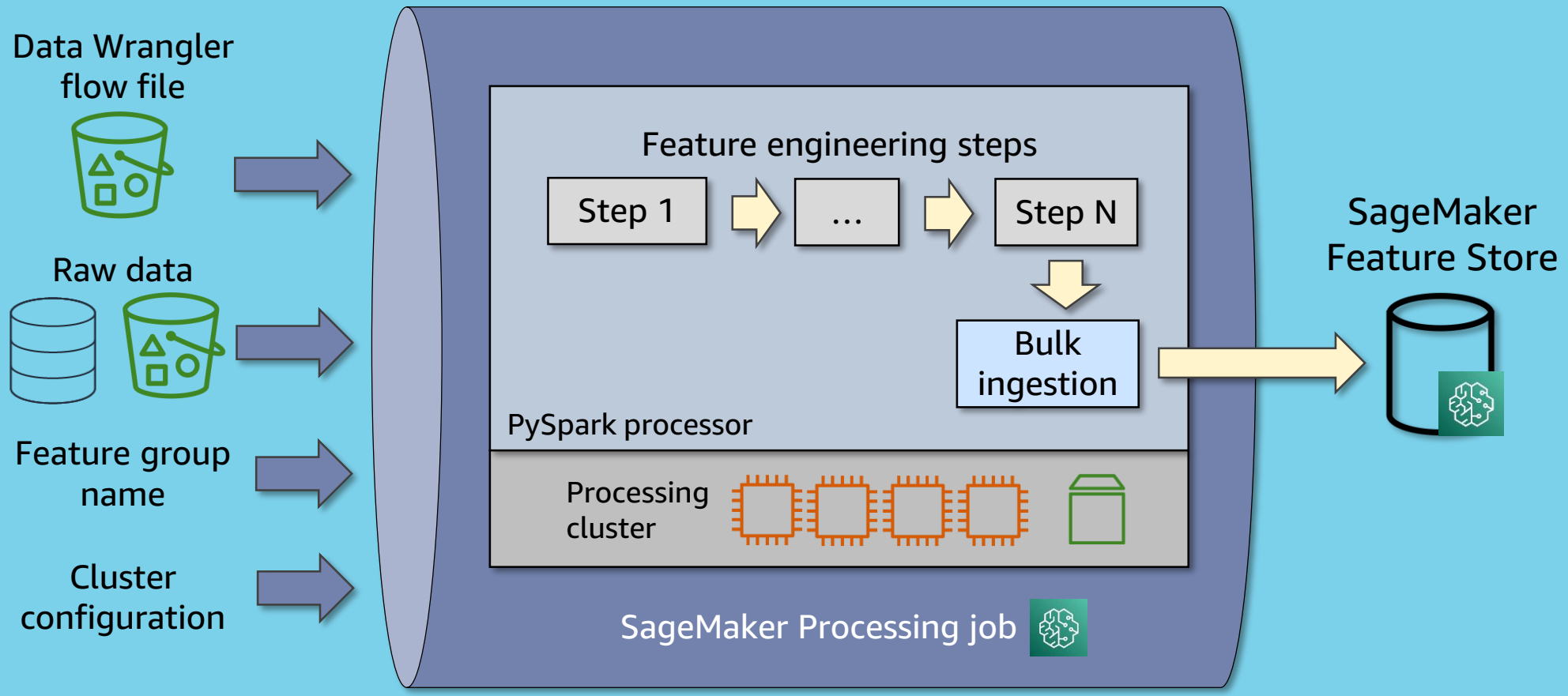
Python SDK

```
fg = FeatureGroup(name='my-fg-name',
                  sagemaker_session=fs_session)

fg.ingest(df, max_processes=20, max_workers=4)
```

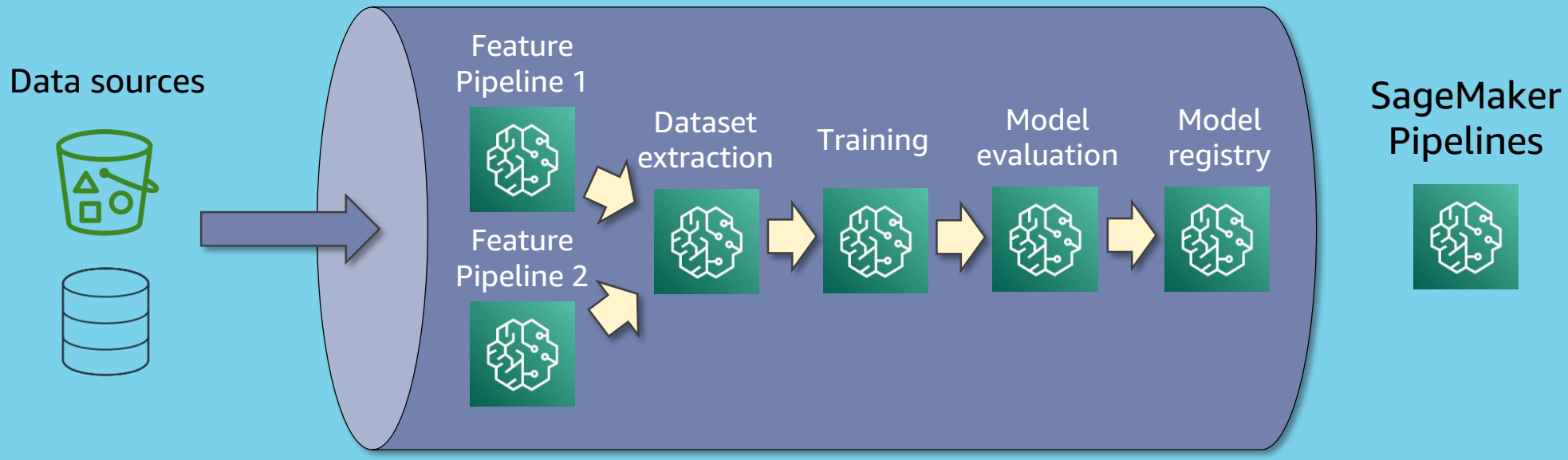
Spark Connector

```
fs_manager.ingest_data(input_data_frame=spark_df,
                       feature_group_arn=fg_arn)
```

Data Wrangler feature pipeline



Feature pipeline can be part of larger ML pipeline



Pipeline execution

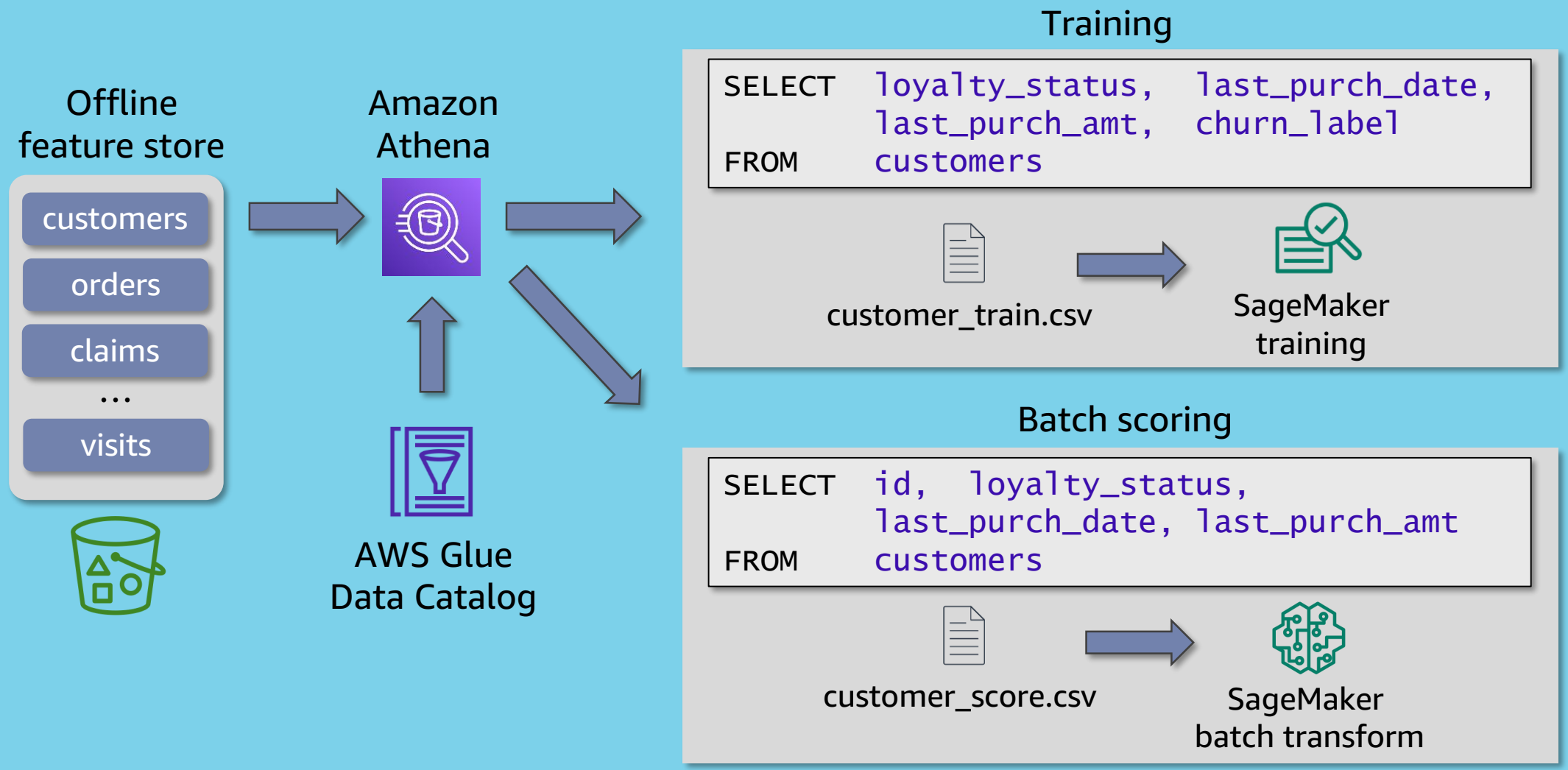
- On demand
- Scheduled
- Triggered by code check-in
- Triggered by new raw data

AWS Lambda

Amazon EventBridge

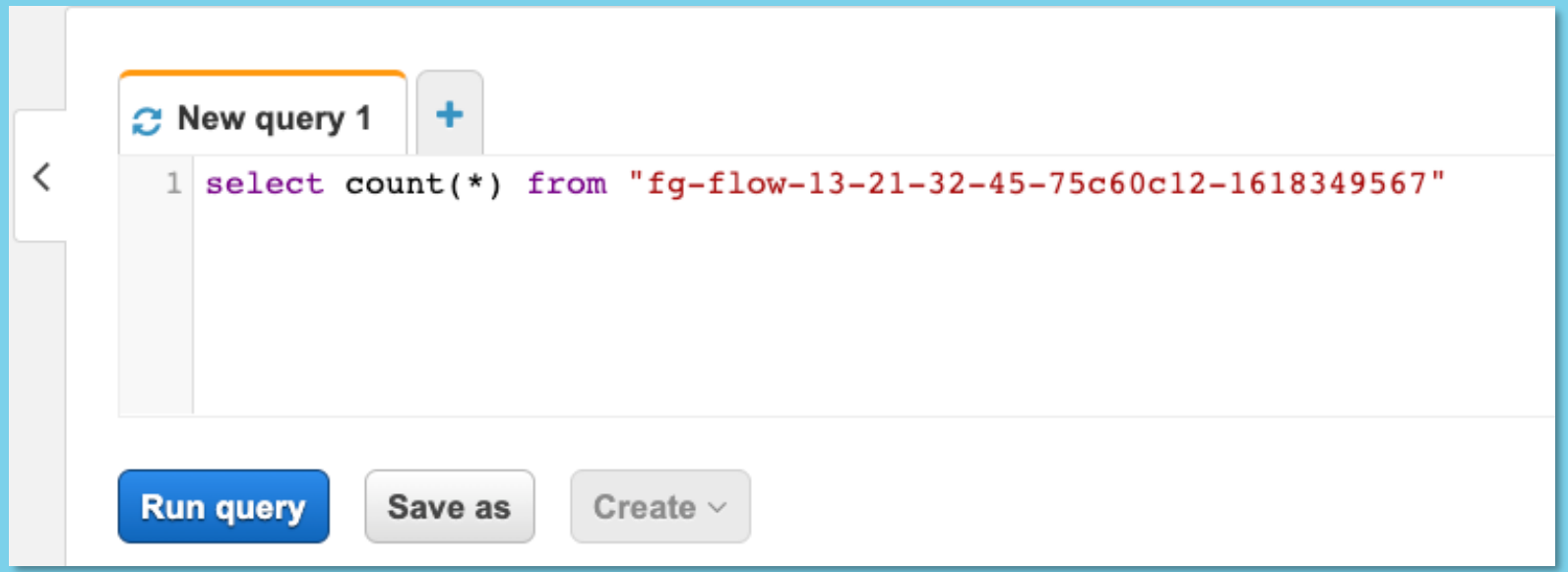
Working with offline features

Using Amazon Athena to query features



Querying features interactively or with Python SDK

Athena console



Python SDK

```
s = f'SELECT COUNT(*) FROM "{fg.athena_query().table_name}" ' + \
          'WHERE fl_date = \'2020-03-31\''
q = feature_group.athena_query()
q.run(s, output_location=output_location)
q.wait()
df = q.as_dataframe()
```

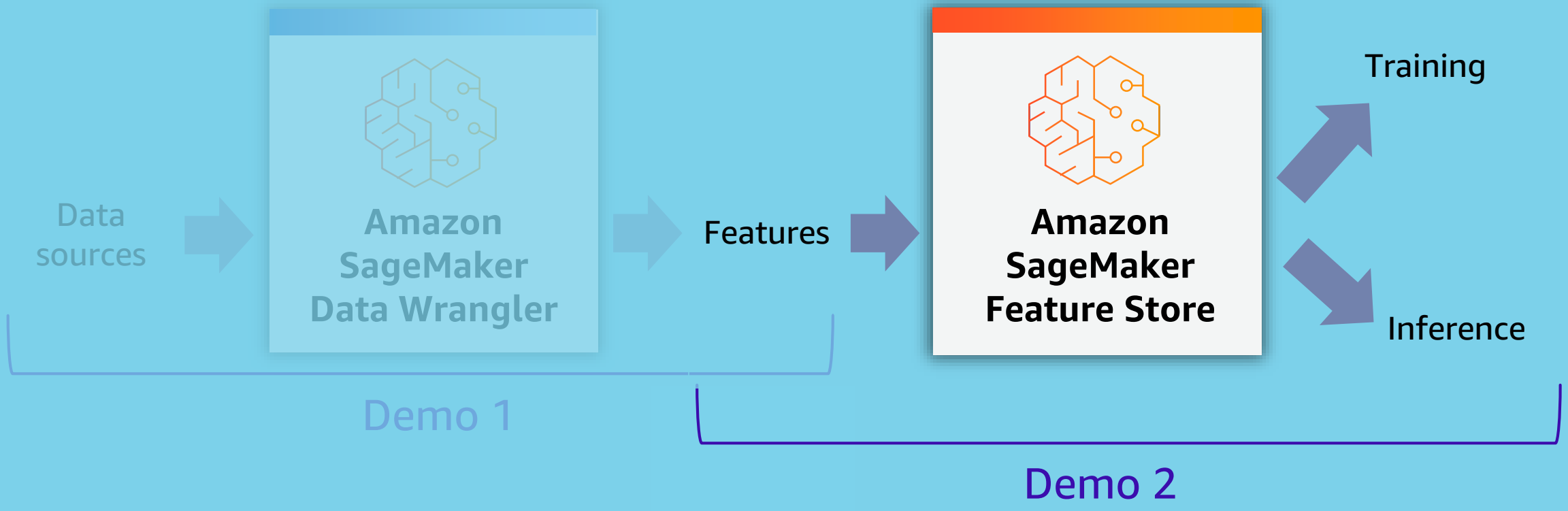
Offline store, under the hood

- Compressed parquet files
- Partitioned by event time
- Feature value history for point-in-time correct datasets

```
s3://…/offline-store/flight-delays-fg/data/
    year=2020/
        month=03/
            day=31/
                hour=00/
```

Name
20200331T000000Z_dD6TOLNJDGnjyCvv.parquet
20200331T000000Z_dRks1omtxu7Q8Ht8.parquet
20200331T000000Z_dTMroptdkVWCP5aR.parquet
20200331T000000Z_dz1vwlFZ9Qd3sDoj.parquet
20200331T000000Z_Eg4XDeRIDgMqbBBx.parquet
20200331T000000Z_GiVBADnjVrUVpcQN.parquet
20200331T000000Z_GnGo1iEvidVAbAIN.parquet
20200331T000000Z_gYS938zYCb9KdVYm.parquet

Demo 2: SageMaker Feature Store, feature pipelines, pulling features for model training



Summary

Prepare data for machine learning with ease, speed, and accuracy

SageMaker Data Wrangler

The fastest and easiest way to prepare data for machine learning

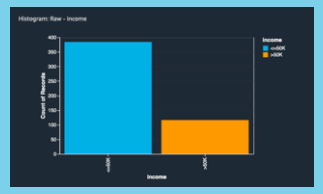
SageMaker Feature Store

Securely store, discover, and share features for both real-time inference and training

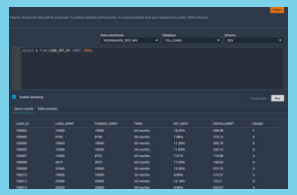
SageMaker Data Wrangler blog posts

Data Wrangler overview - link

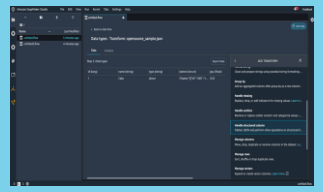
Patient readmission use case - link



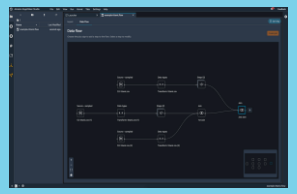
Handling imbalanced data - link



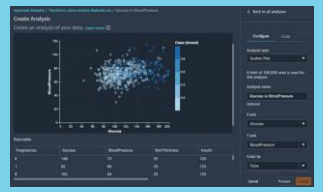
Preparing data from Snowflake - link



Prepare data from JSON and ORC files - link



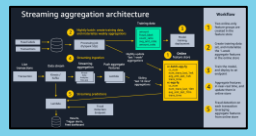
Launch processing jobs - link



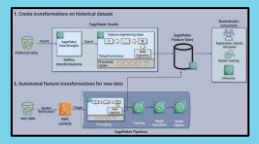
Data Wrangler with SageMaker Autopilot - link

SageMaker Feature Store blog posts

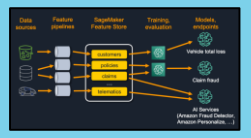
Understanding key capabilities - link



Using streaming ingestion to make ML-backed decisions in near-real time - link



Automating feature pipelines - link

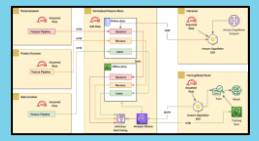


Extending ML lineage to include features - link

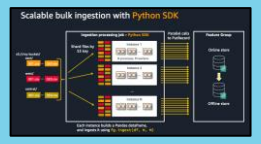
```
import random, string

filename = strftime("%Y%m%d%H%M%S2_", gm_time)
filename += ''.join(random.choice(string.ascii_uppercase
filename += '.parquet'

df.to_parquet(filepath + filename)
```

Directly ingesting historical feature data to S3 - link

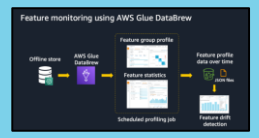
Building accurate training datasets using point-in-time queries - link



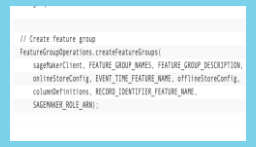
Enabling feature reuse across accounts - link



Scaling batch ingestion - link



Monitor ML features - link

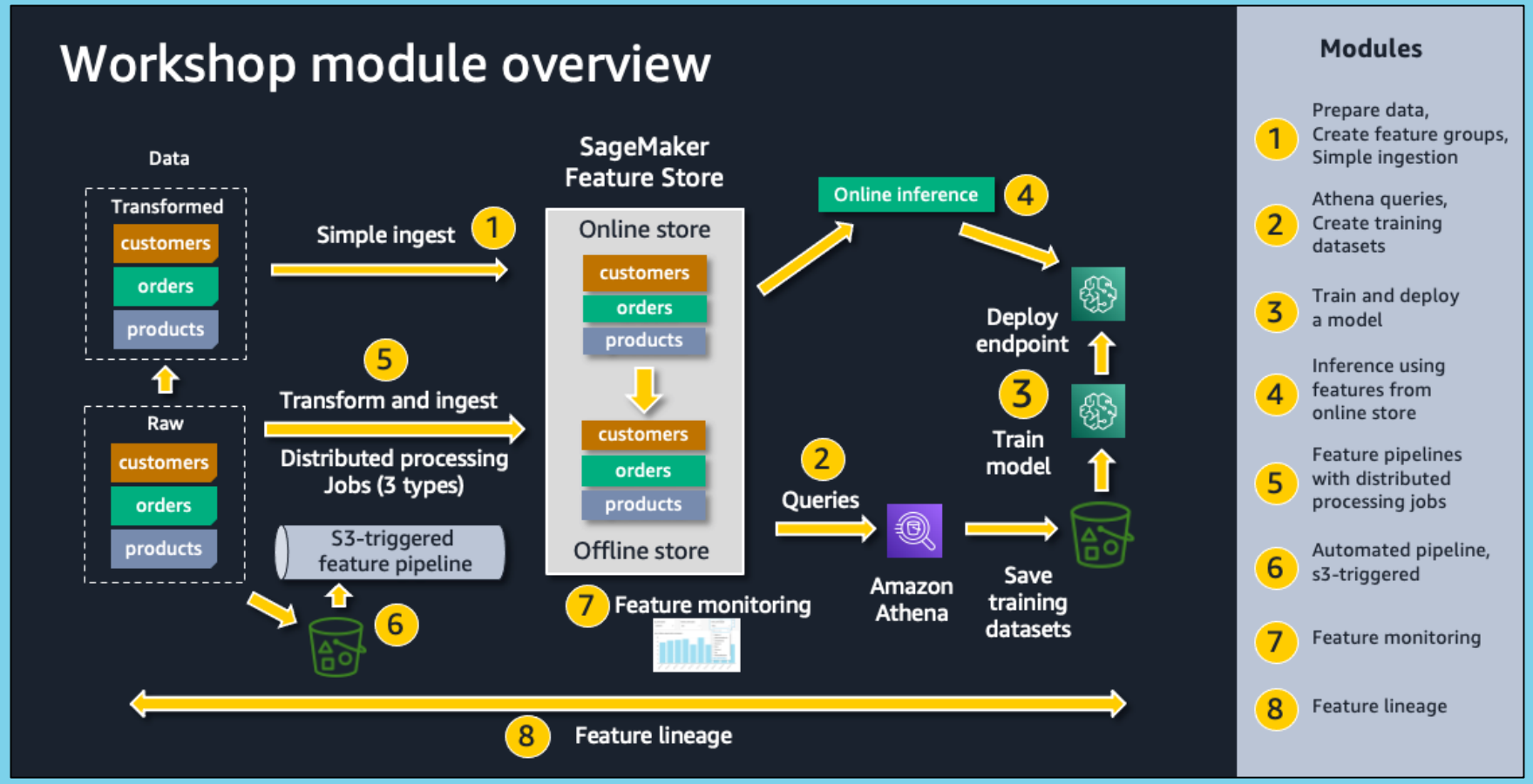


```
// Create feature group
FeatureGroupOperations.createFeatureGroups(
    sageMakerClient, FEATURE_GROUP_NAMES, FEATURE_GROUP_DESCRIPTION,
    onlineStoreConfig, EVENT_TIME_FEATURE_NAME, offlineStoreConfig,
    columnDefinitions, RECORD_IDENTIFIER_FEATURE_NAME,
    SAGEMAKER_ROLE_ARN);
```

Using feature store in a Java environment - link

SageMaker Feature Store workshop

This workshop gives you an end-to-end hands-on introduction to SageMaker Feature Store - link



https://github.com/aws-samples/amazon-sagemaker-feature-store-end-to-end-workshop

Thank you!