
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

AIM339

Store features across teams with Amazon Feature Store, featuring Zalando

Mark Roy

Principal ML Specialist SA
AWS

Romi Datta, PhD

Sr. Manager, Product Management
AWS

Sergii Pylypchuk

Principal Engineer
Zalando

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Feature engineering and its challenges

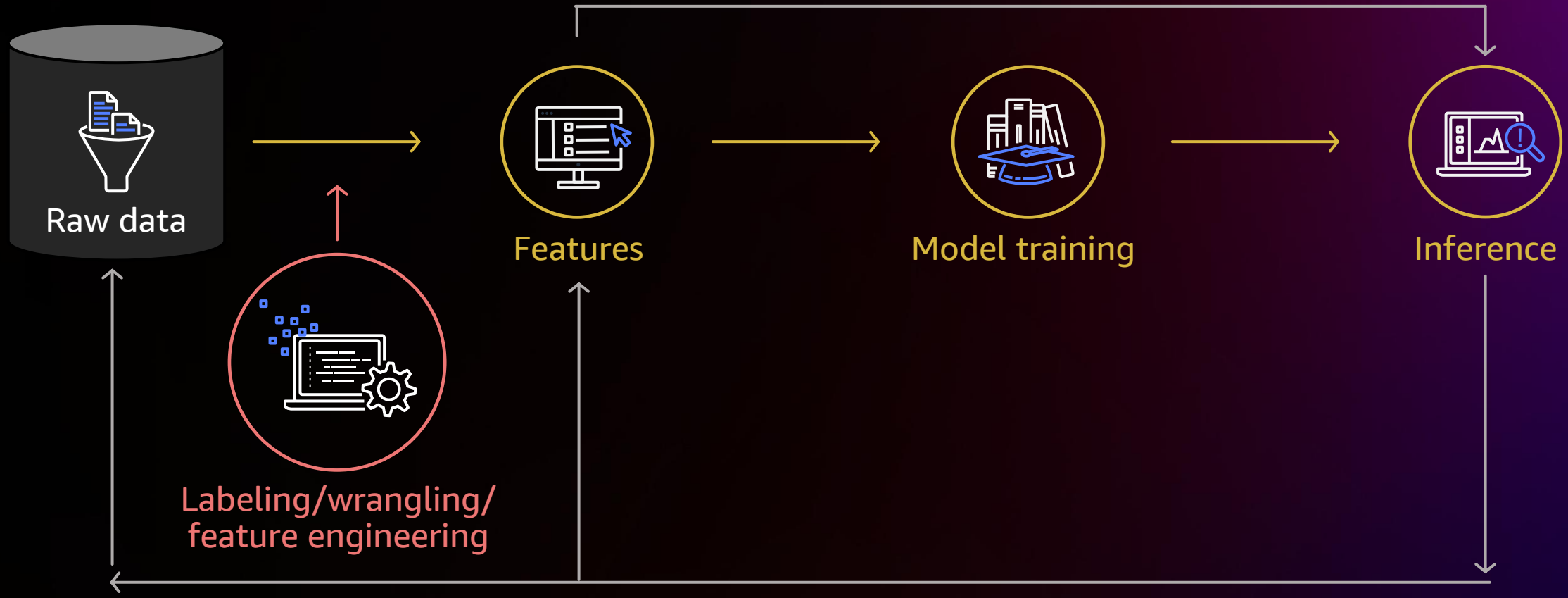
Amazon SageMaker Feature Store introduction

SageMaker Feature Store deep dive and demo

Zalando customer use case with SageMaker Feature Store

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Machine learning workflow



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Common challenges with feature engineering

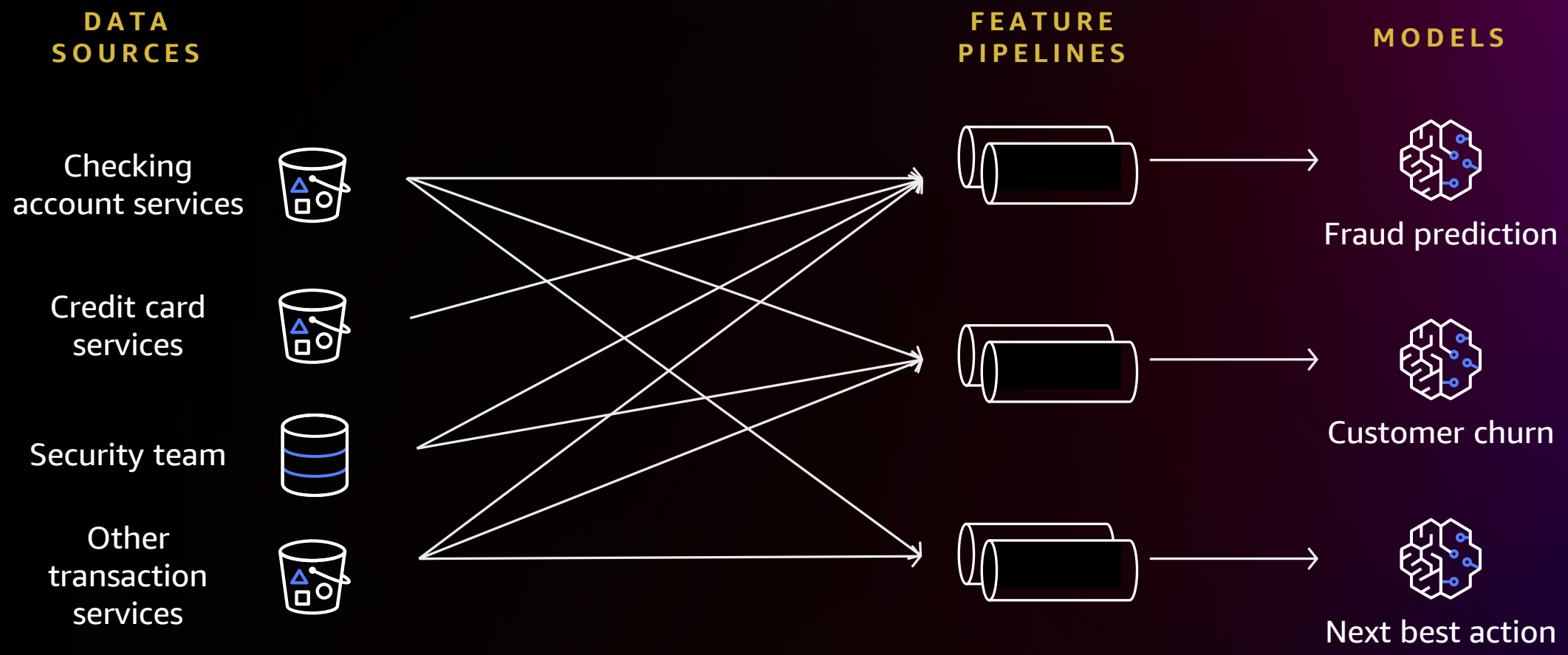
1. Time consuming and resource intensive
2. Redundant feature pipelines
3. No sharing or discovery mechanism
4. Troubleshooting overhead due to training-serving skew

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Without a feature store

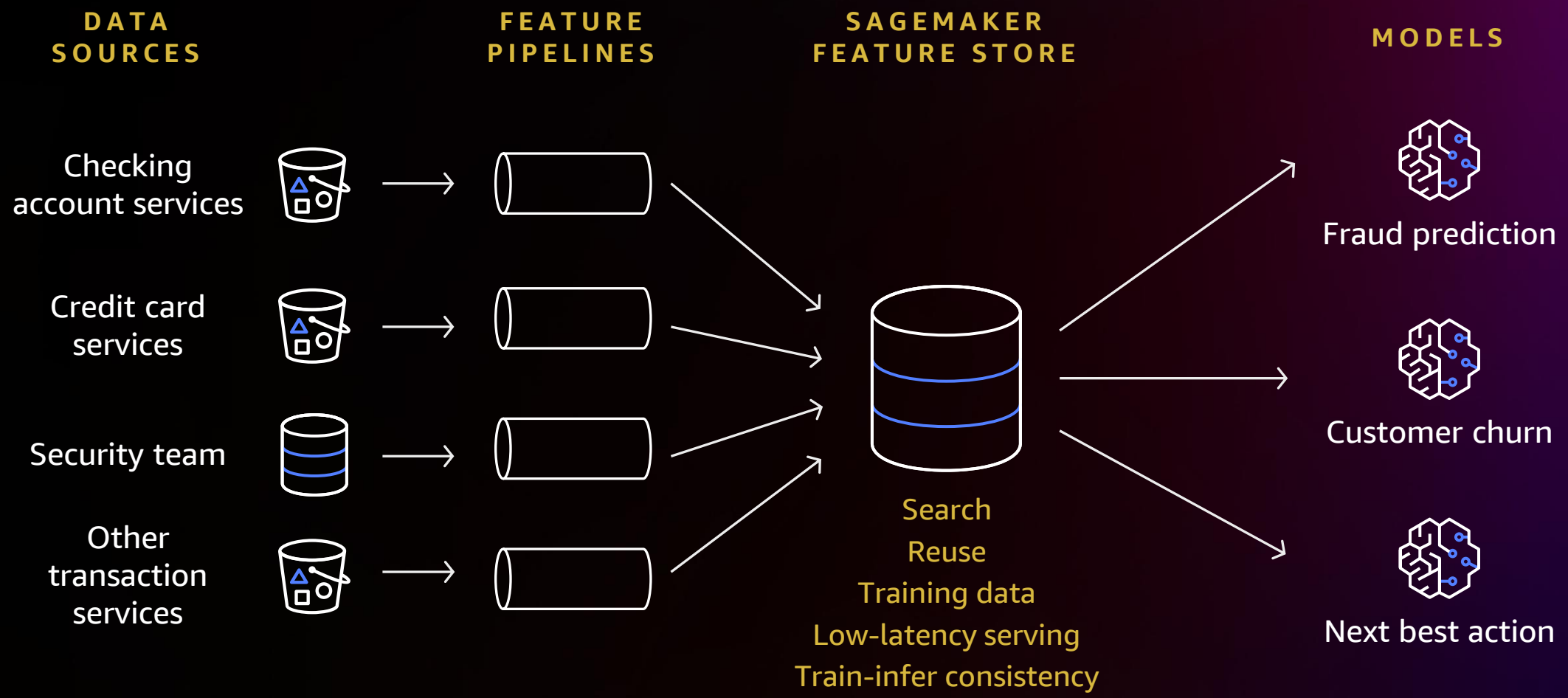
STANDALONE FEATURE ENGINEERING FOR EACH NEW MODEL



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

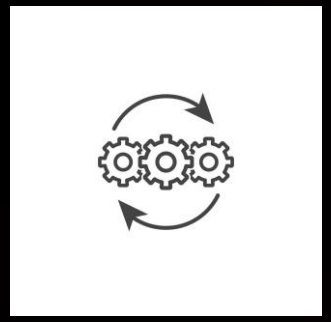
With SageMaker Feature Store...

BUILD FEATURES ONCE AND REUSE THEM ACROSS TEAMS AND MODELS



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SageMaker Feature Store common business use cases



Business process optimization



Personalization



Fraud detection

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SageMaker Feature Store

Accelerate machine learning with a fully managed feature store



Batch and streaming ingestion

High-throughput writes and reads for features



Feature search and discovery

Store metadata for features; leverage automatic data cataloging



Online and offline features

Online features for real-time prediction; offline features for model training and batch prediction



Feature consistency

Service maintains consistency between online and offline store to avoid training-inference skew



Security and access control

Access control for feature data and feature metadata



Serve features with low latency

Millisecond-latency serving of features for real-time inference

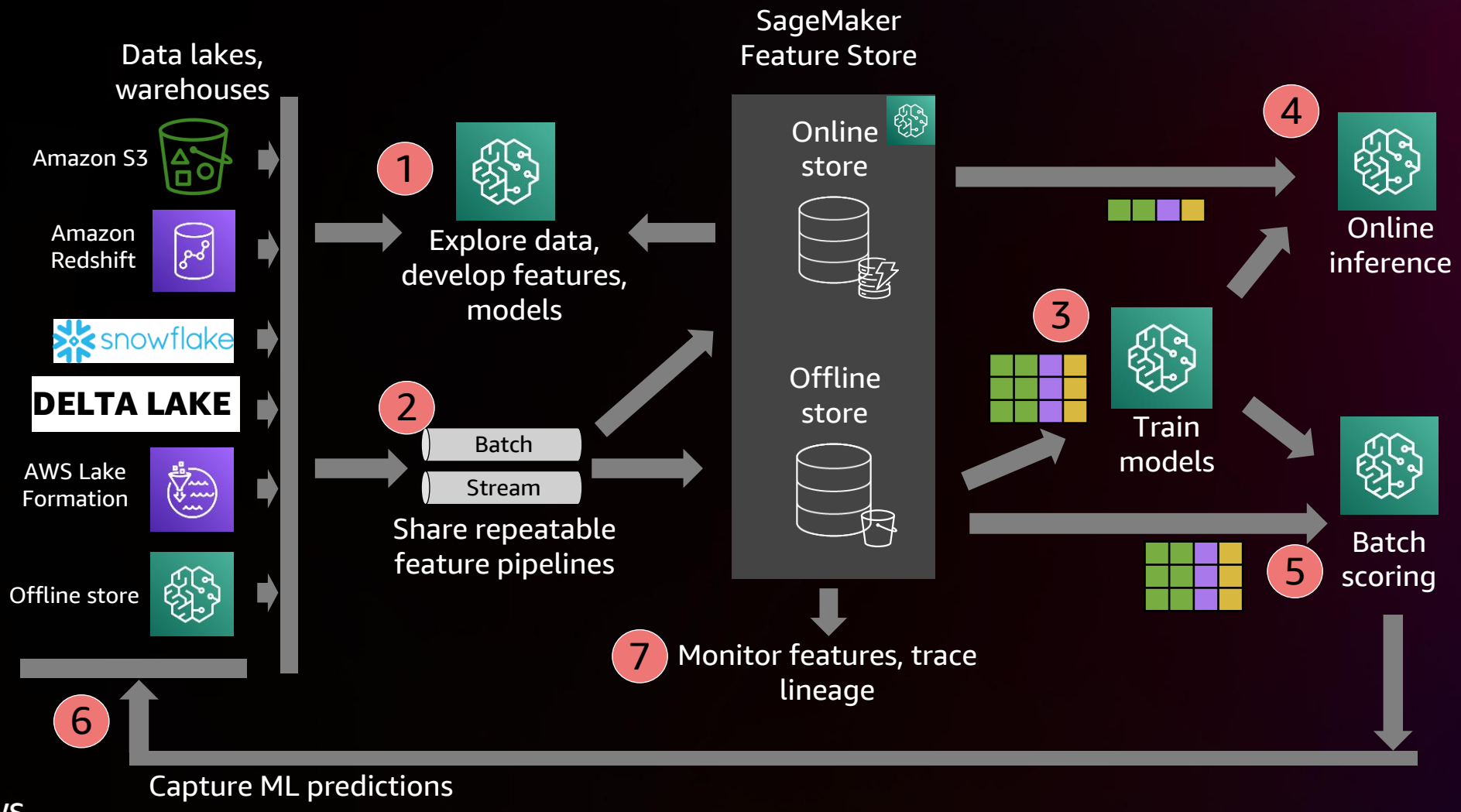
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights r

Deep dive

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Using Feature Store with data lakes, warehouses



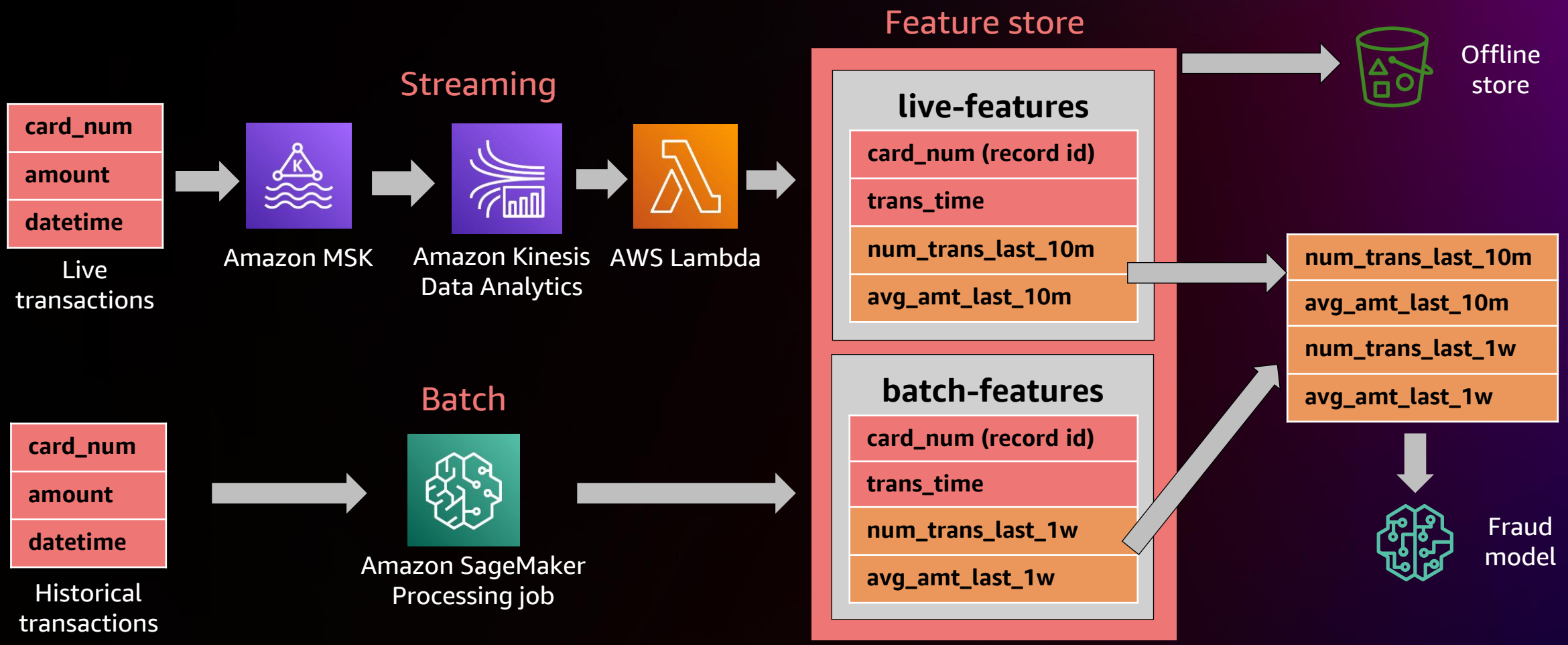
Steps

1. Use SageMaker Studio notebooks or SageMaker Data Wrangler to explore existing data, experiment with features and models
2. Create repeatable feature pipelines, share features
3. Train models from available features, with point-in-time feature accuracy
4. Make low-latency online predictions without train-serve skew
5. Make batch predictions with latest feature values
6. Capture prediction results to drive downstream use cases
7. Monitor features and trace lineage

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Features are materialized at different frequencies

ONLINE MODELS CAN PULL LATEST FEATURES WITH LOW LATENCY



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Feature consistency with online and offline store

```
# uniquely identified records
put_record('id-1', t1, 0.1)

put_record('id-2', t2, 0.2)

put_record('id-3', t3, 0.3)


# new records for existing ID
put_record('id-1', t4, 0.4)

put_record('id-1', t5, 0.5)
```

Feature group after all PutRecord calls

Online store

id	event time	feature
id-1	t5	0.5
id-2	t2	0.2
id-3	t3	0.3



Offline store

id	event time	feature
id-1	t1	0.1
id-2	t2	0.2
id-3	t3	0.3
id-1	t4	0.4
id-1	t5	0.5

- **Latest feature values** for each ID
- Values available **immediately**

- Appends new records, **history of feature values**
- <15 minute read-after-write consistency

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How do I find available features?



Data scientists

What features are already built?

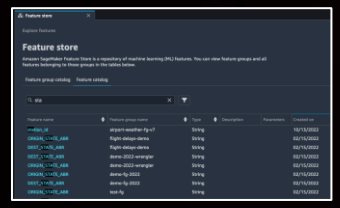
SageMaker Feature Store

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

How do I find available features?

SageMaker Feature Store

SageMaker Studio feature search



Data scientists

Search API

Feature catalog

Feature groups

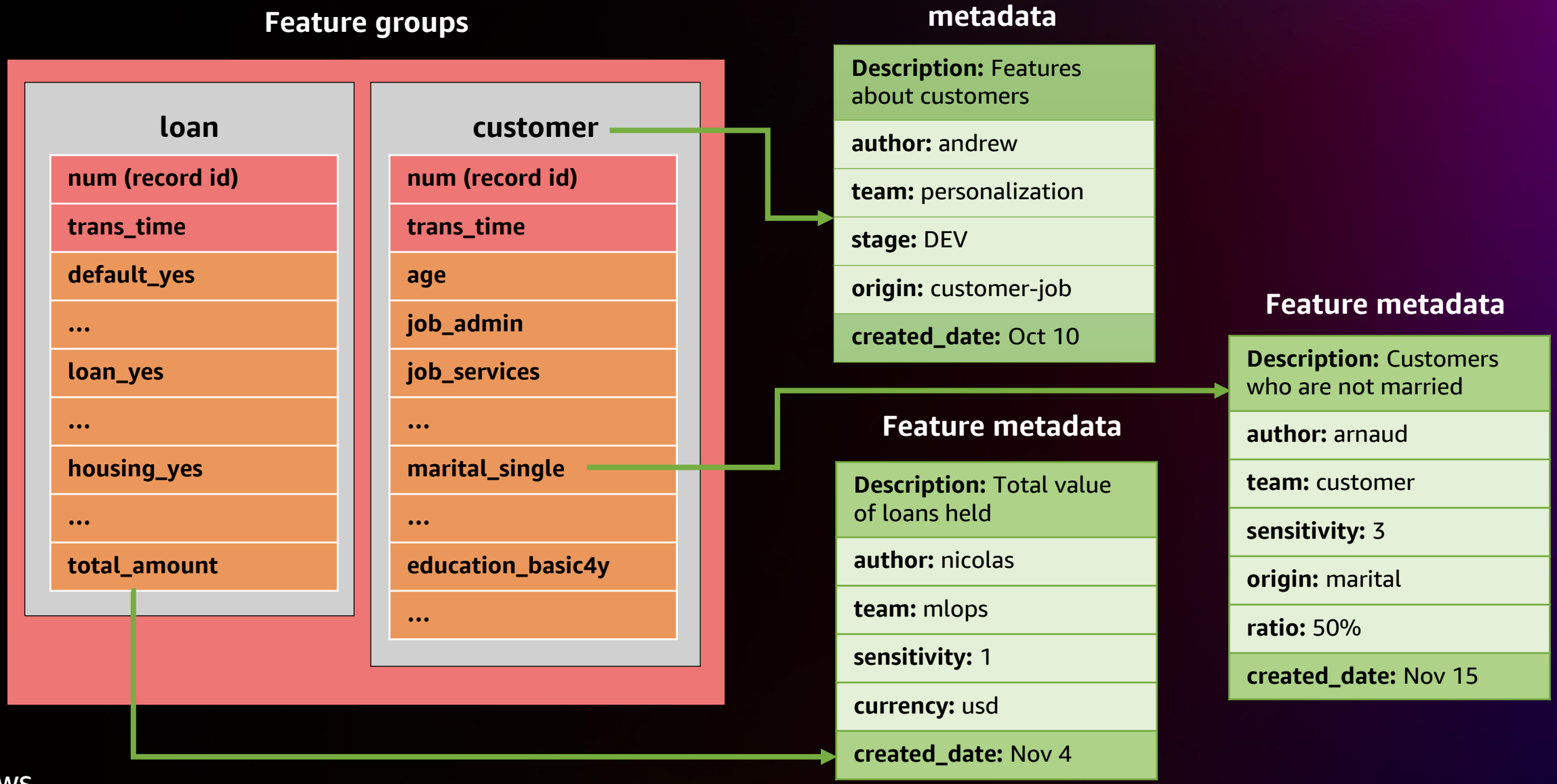
- Descriptions
- Metadata

Features

- Descriptions
- Metadata

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Metadata example



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Feature ingestion APIs

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)
```

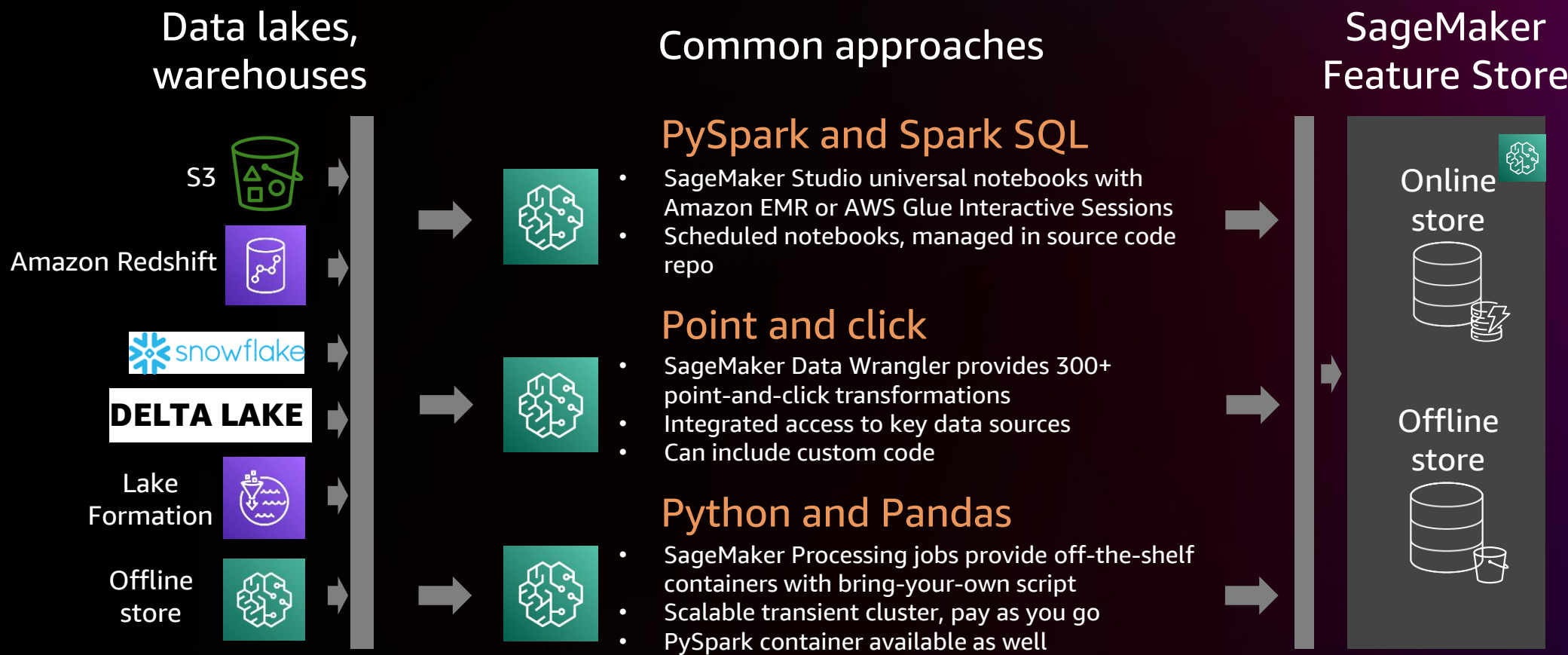
Spark connector

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

aws

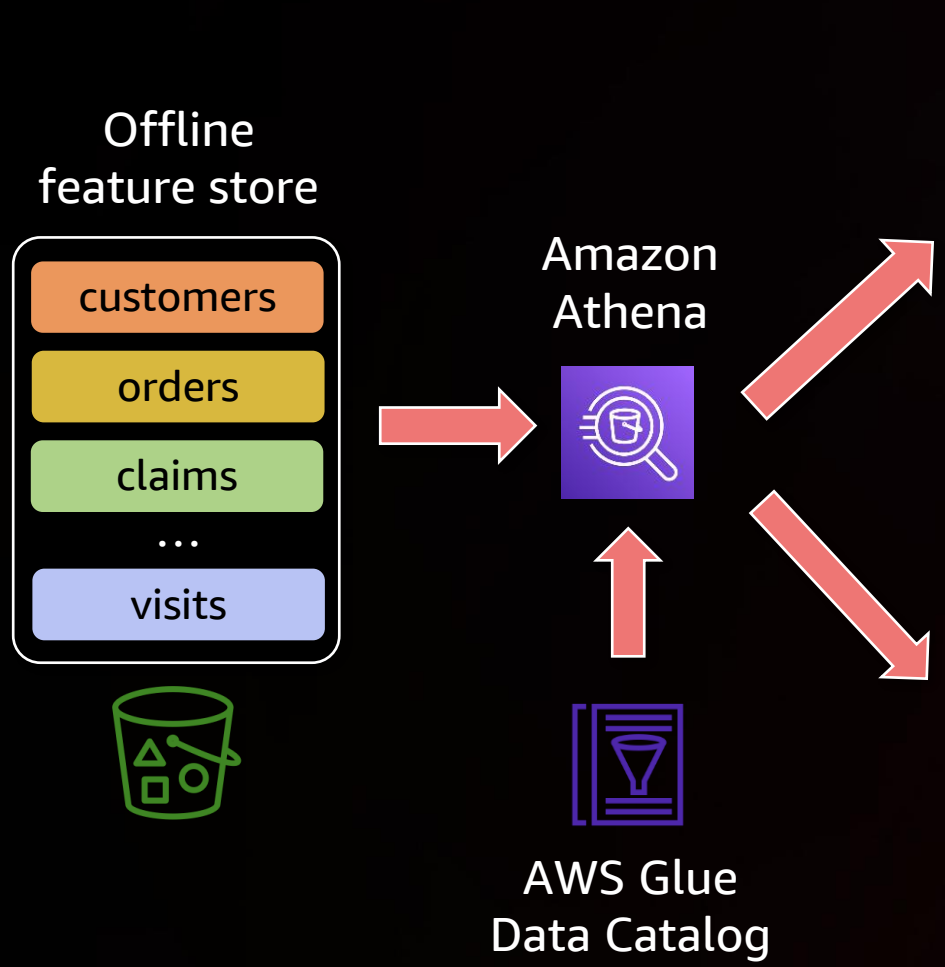
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Feature pipeline automation



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Use SQL to create training and scoring datasets



Training

```
SELECT   loyalty_status,  last_purch_date,
         last_purch_amt,  churn_label
FROM     customers
```

customer_train.csv → SageMaker training

Batch scoring

```
SELECT   id,  loyalty_status,
         last_purch_date, last_purch_amt
FROM     customers
```

customer_score.csv → SageMaker batch transform

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Feature retrieval APIs

GetRecord API

```
sm_fs.get_record(FeatureGroupName='customer-fg',
                 RecordIdentifierValueAsString='CUST-001',
                 FeatureNames=['spend-last7d', 'tenure-days'])
```

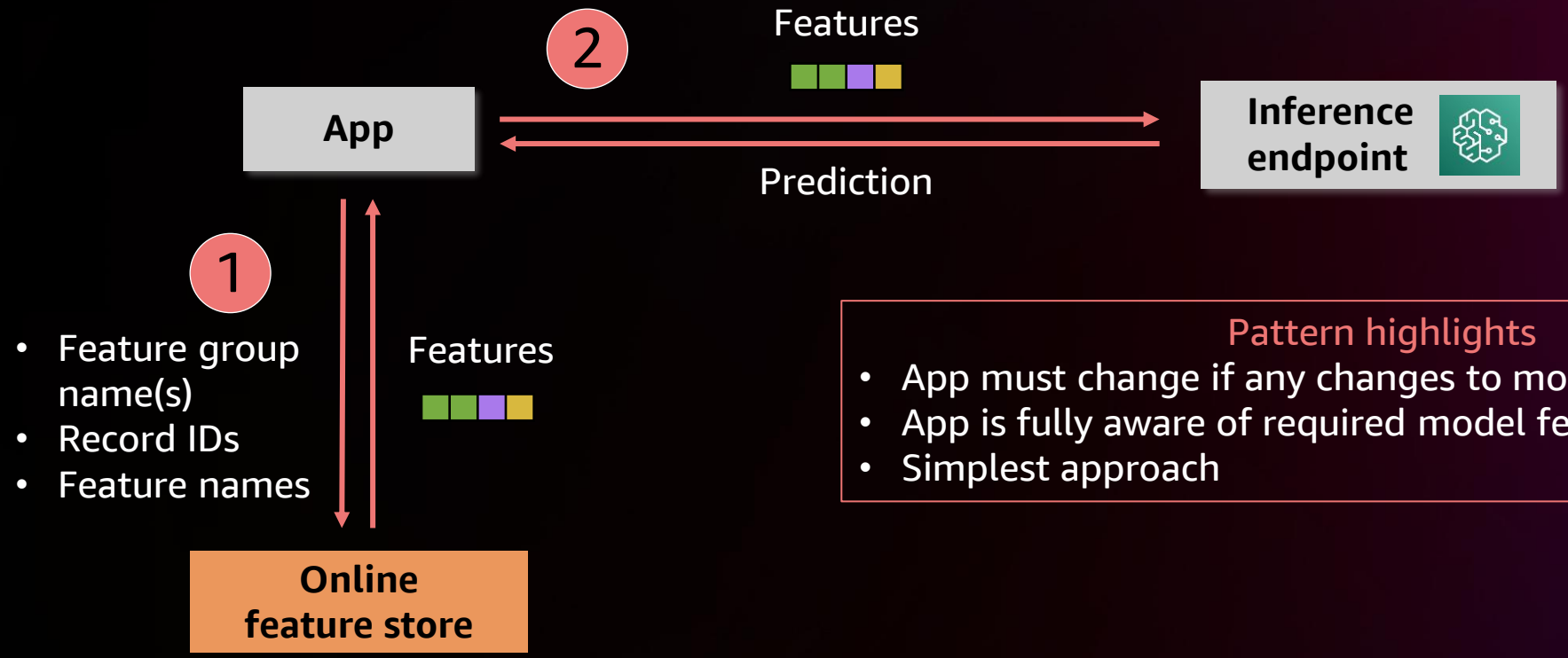
BatchGetRecord API

```
sm_fs.batch_get_record(Identifiers=[
    {'FeatureGroupName': 'customer-fg',
        'RecordIdentifiersValueAsString': [
            'CUST-001', 'CUST-002', 'CUST-003'],
        'FeatureNames': ['spend-last7d', 'tenure-days']},
    {'FeatureGroupName': 'product-fg',
        'RecordIdentifiersValueAsString': [
            'P-100', 'P-200'],
        'FeatureNames': ['orders-last7d', 'daily-revenue']}])
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Inference pattern 1: App-based feature lookup

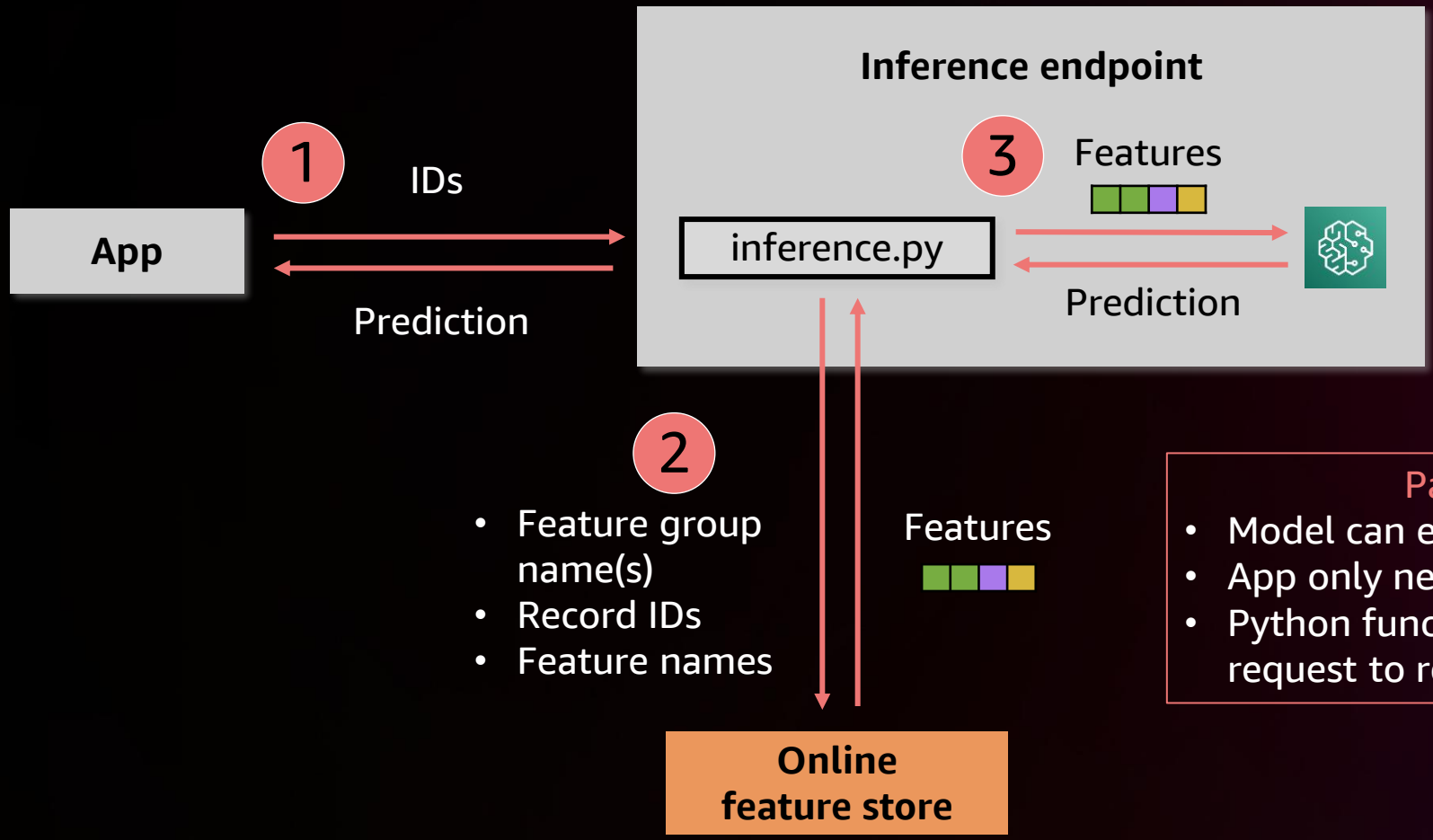


Pattern highlights

- App must change if any changes to model input features
- App is fully aware of required model features
- Simplest approach

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Inference pattern 2: Endpoint-based feature lookup

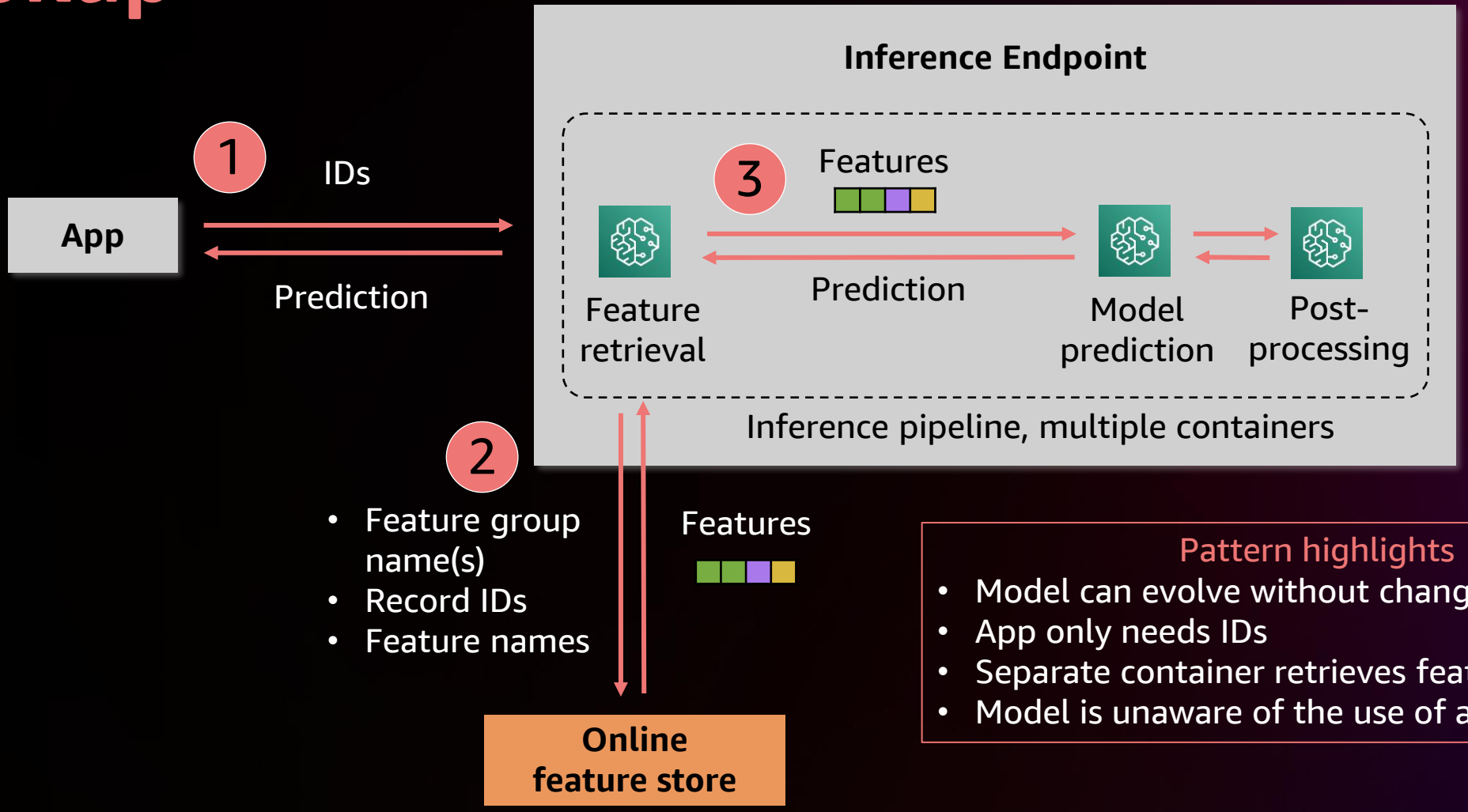


Pattern highlights

- Model can evolve without changing app
- App only needs IDs
- Python function intercepts prediction request to retrieve latest features

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Inference pattern 3: Inference pipeline feature lookup



Pattern highlights

- Model can evolve without changing app
- App only needs IDs
- Separate container retrieves features
- Model is unaware of the use of a feature store

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Zalando journey towards SageMaker Feature Store

Sergii Pylypchuk (he/him)

Principal Software Engineer
Zalando Payments GmbH

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SageMaker Feature Store at Zalando

This presentation is a case study of the Zalando experience

This is not an endorsement or recommendation of any vendor, product, or service

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

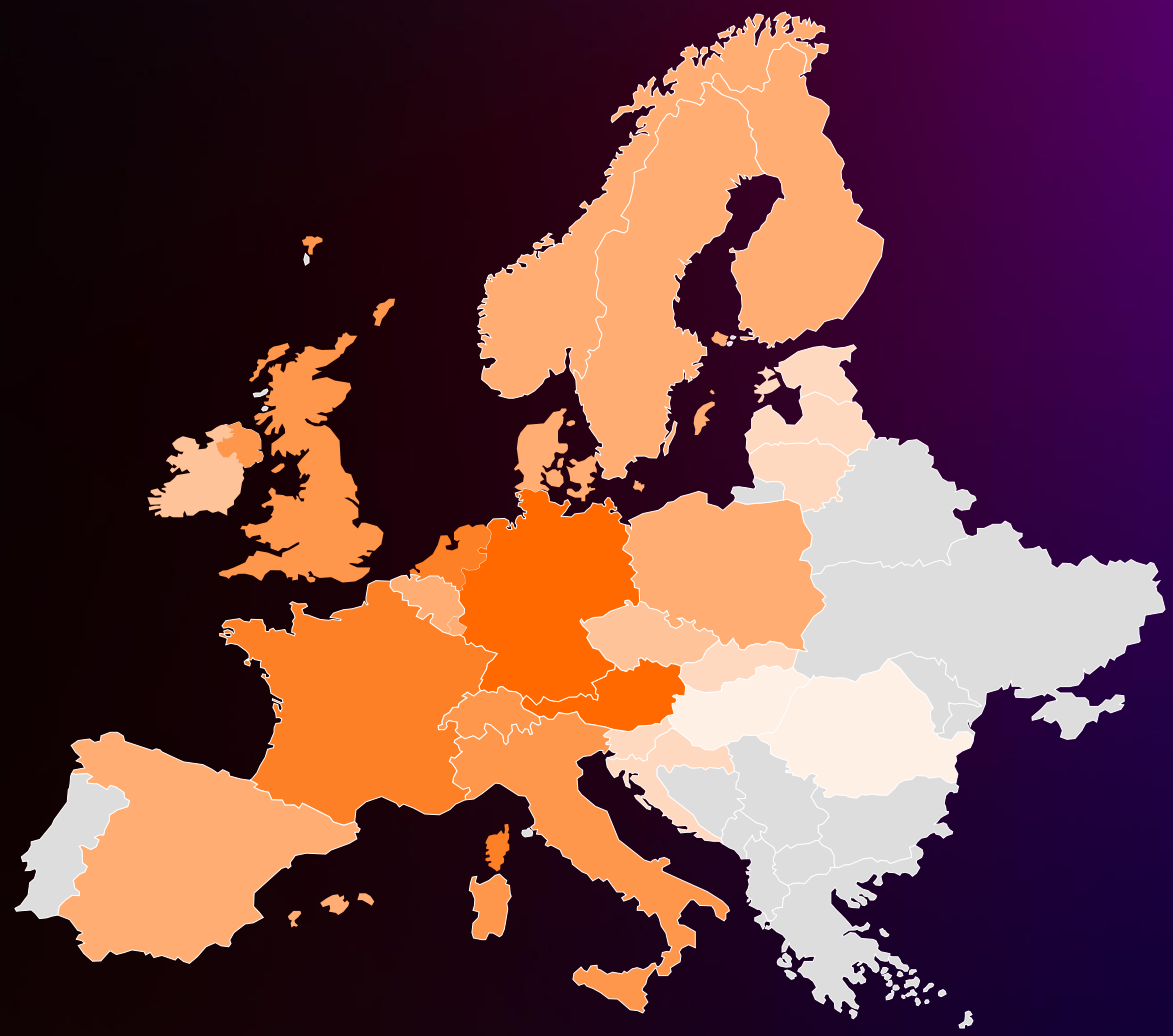
Introduction

Bringing fashion to 25 countries

More than 49M active customers

More than 250M orders last year

More than 20 payment methods



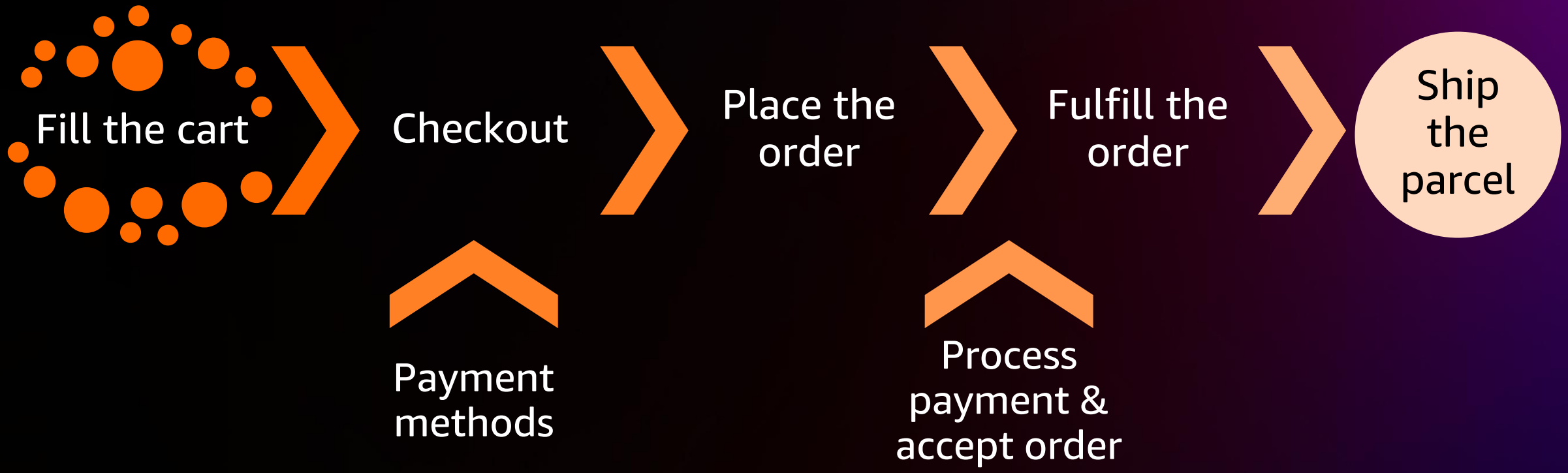
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Introduction: Payments



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

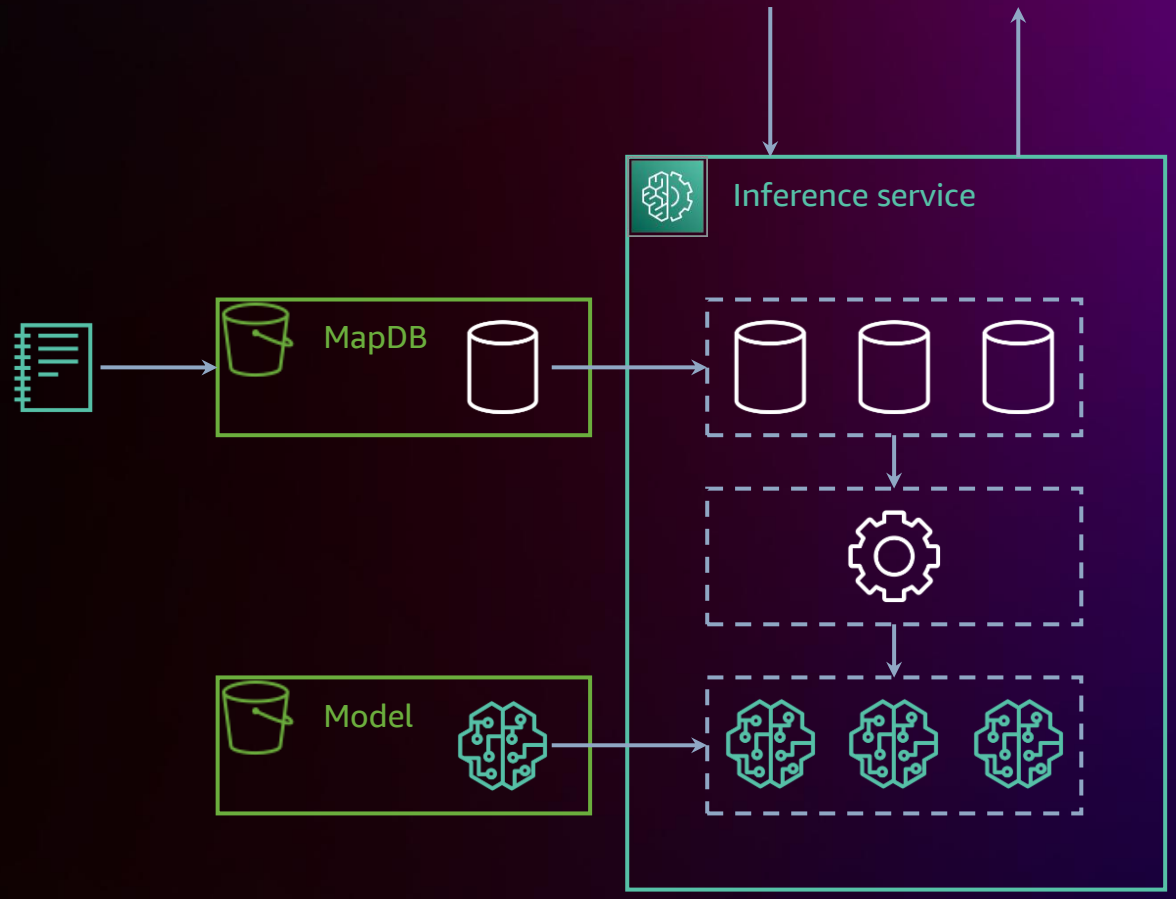
Introduction: Payments



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Problem statement

- Home-built ML infrastructure
- Batch-computed features are available for ML inference only
- Similar feature implementations in different teams
- High memory consumption
- Long startup time



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

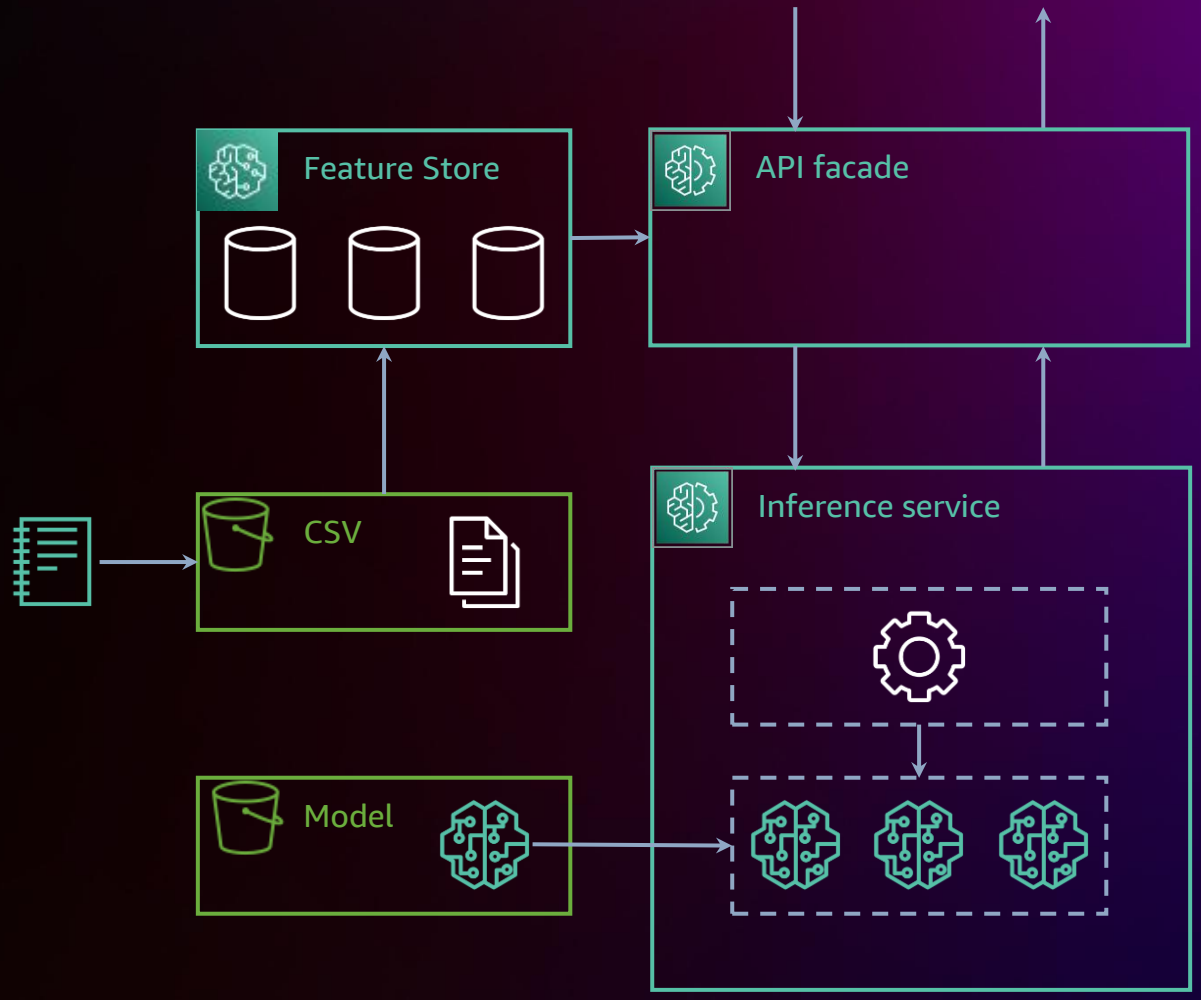
Requirements for the replacement

- Managed solution that can fulfill our performance requirements
  - Low latency for live inference
  - High availability
- Easy feature discovery
- Reusable features across department
- Easy process for adding new features
- Source for training data

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SageMaker Feature Store: Integration

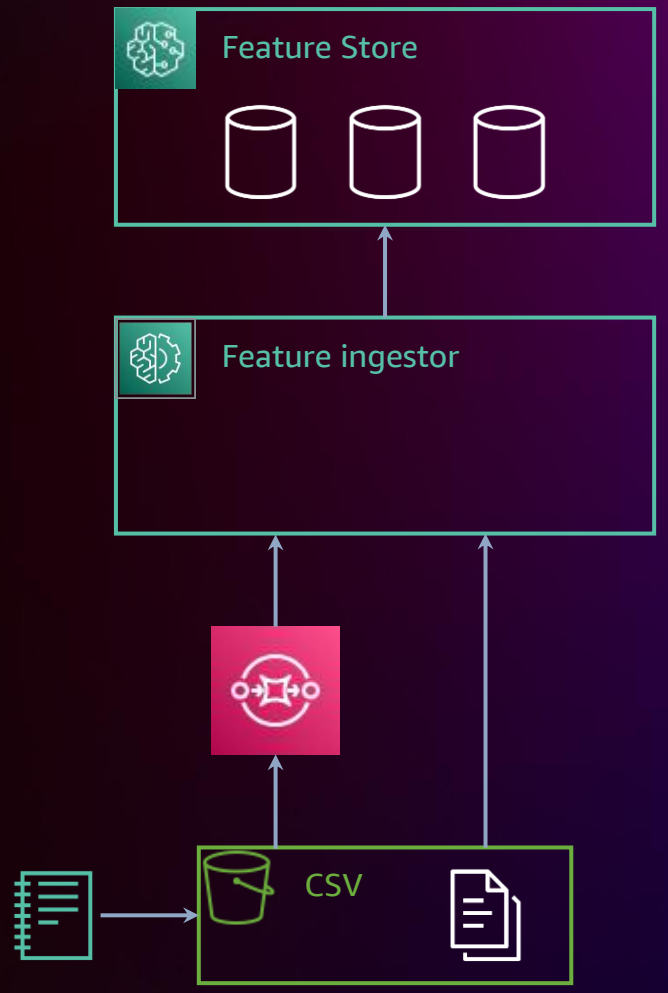
- Features are batch computed in scheduled notebooks
- Automatically ingested into Feature Store
- Available through common API facade



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SageMaker Feature Store: Ingestion

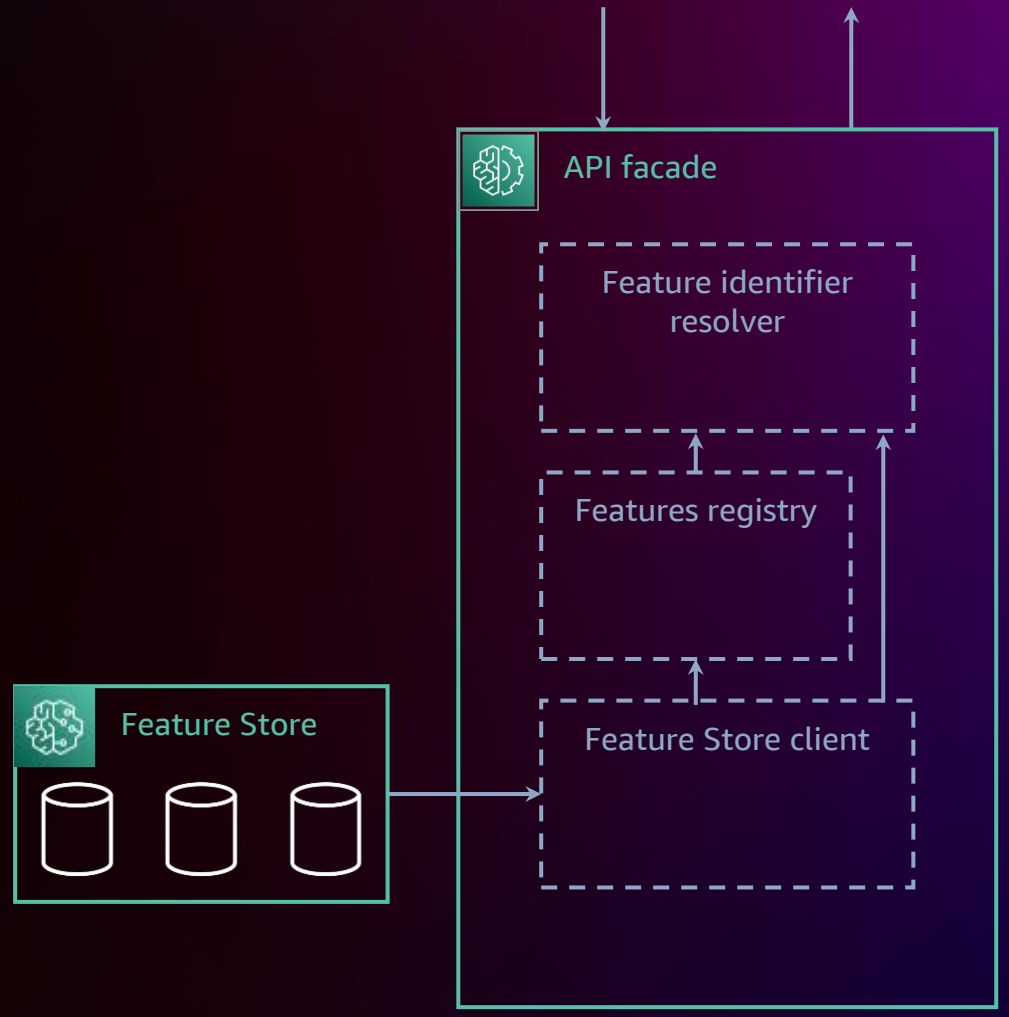
- Notebook generates features on a schedule
- Ingestion is triggered through Amazon SQS
- New feature group is automatically created if it does not exist
- SQS dead letter queue is used to ensure no missing records



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SageMaker Feature Store: Serving

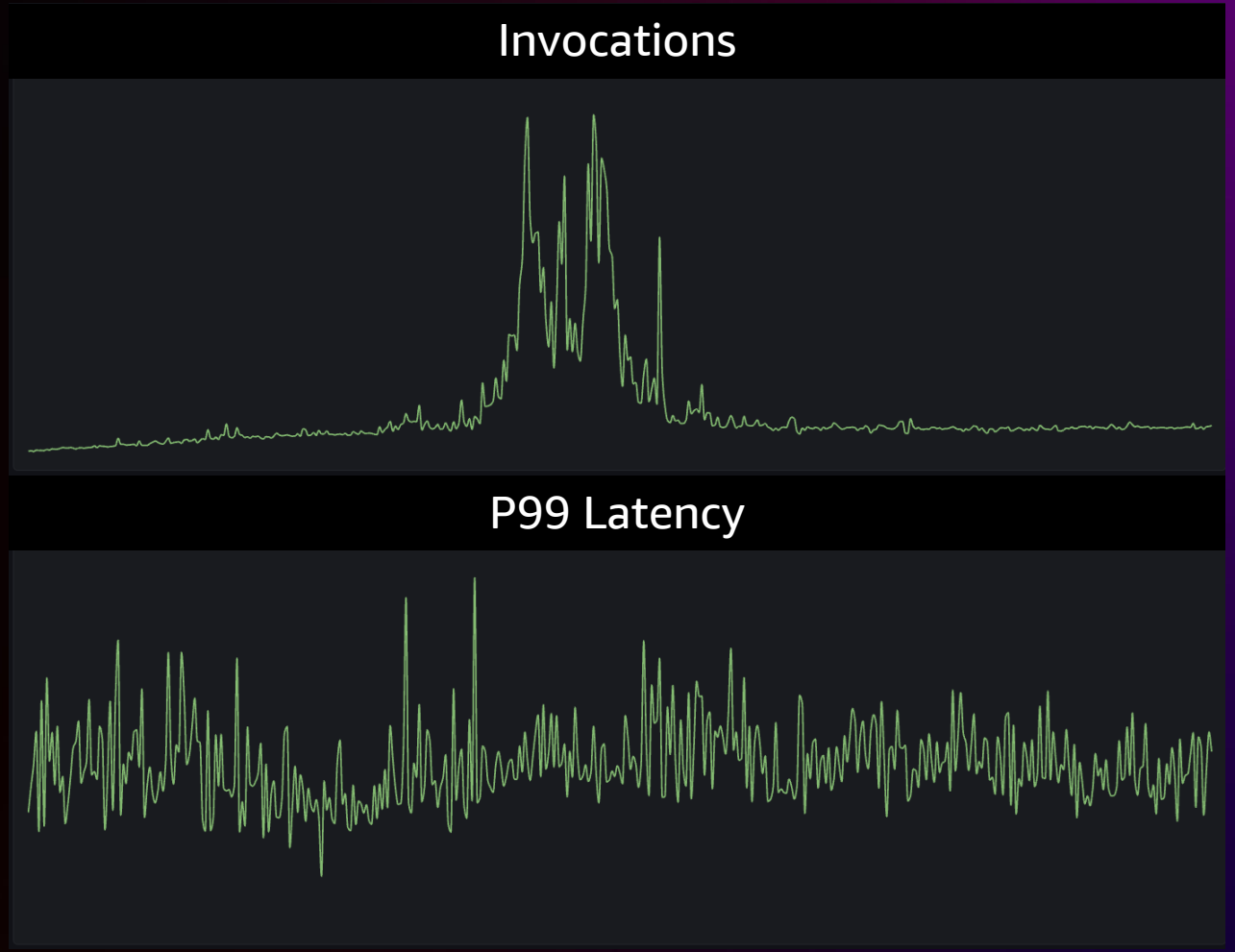
- Available features are queried based on incoming request
- Features are fetched in batches from Feature Store



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SageMaker Feature Store: Performance

- We see stable, low P99 latency
- No problems observed during request bursts
- Availability Zone fault tolerant



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Service improvements during project

- Limited monitoring capabilities
  - Requests, errors, throttled request counts, and latency metrics were added
- Search over existing features was limited
  - Feature-level descriptions, metadata, and search added

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Our wins

- Applied scientists can add new batch-computed features without constraints
- Mitigated operational risk in our legacy application
- Multiple teams in the department can benefit from batch-computed features availability through the HTTP API

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Our plans for expanding Feature Store usage

- Add more features into SageMaker Feature Store
- Evaluate switching training data source to offline Feature Store
- Migrate completely to SageMaker for model training and inference

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Special thanks to Zalando teams
that helped make this happen!

Team NextGen Team Gatekeepers Team Turing

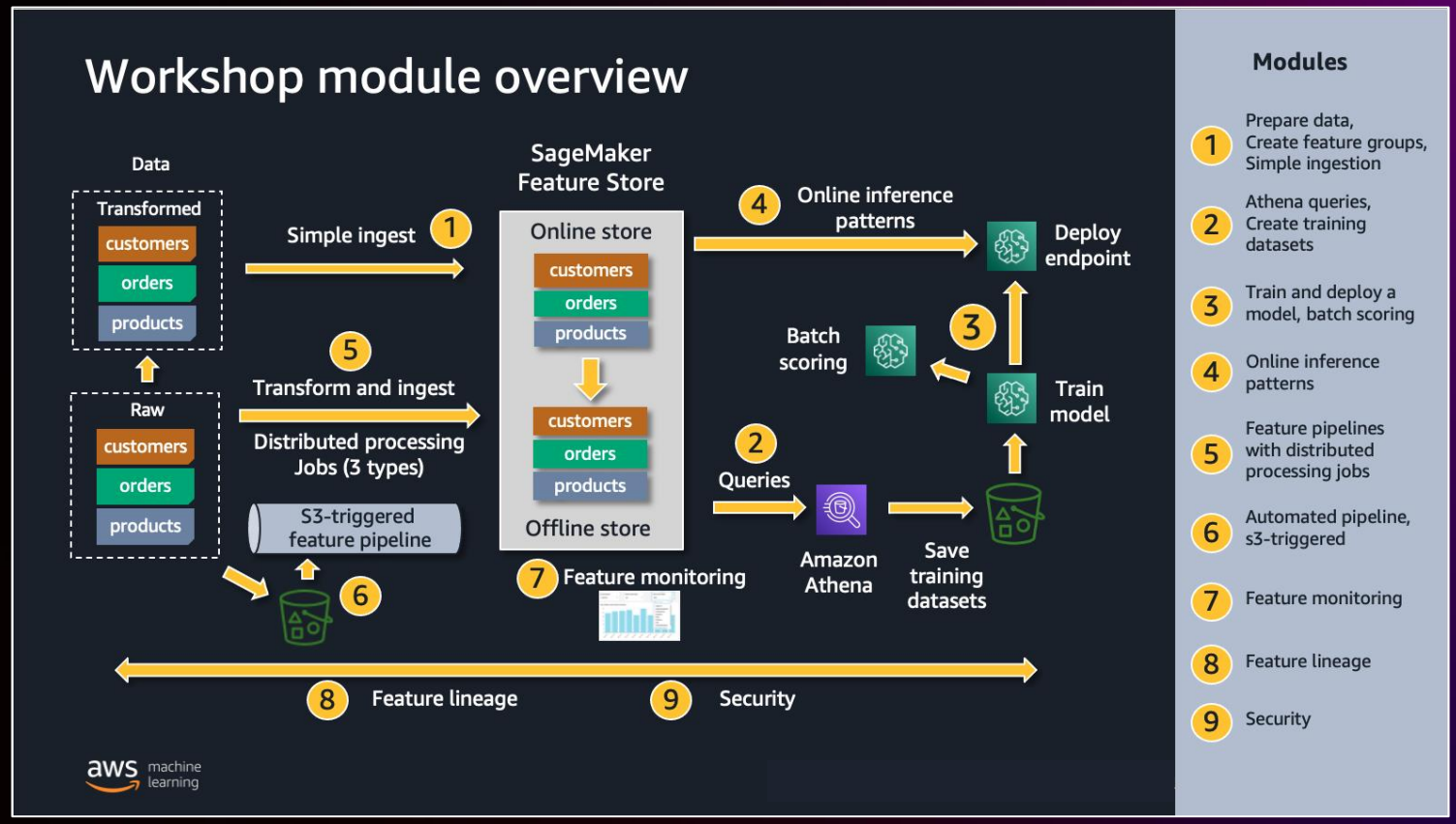
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Resources

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SageMaker Feature Store workshop

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



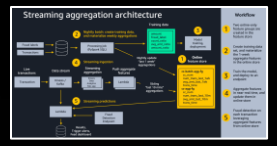
https://github.com/aws-samples/amazon-sagemaker-feature-store-end-to-end-workshop

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

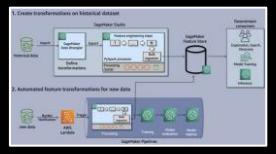
SageMaker Feature Store blog posts



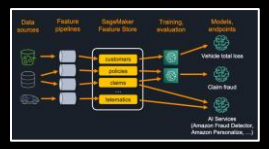
Understanding key capabilities – link



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

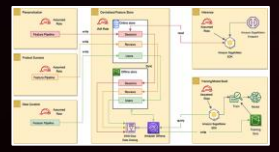
filename = strftime("%Y%m%dT%H%M%SZ_", gm_time)
filename += ''.join(random.choice(string.ascii_upperc
filename += '.parquet'

df.to_parquet(filepath + filename)
```

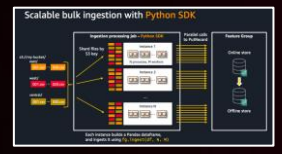
Directly ingesting historical feature data to S3 – link



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



Enabling feature reuse across accounts – link



Scaling batch ingestion – link



Monitor ML features – link



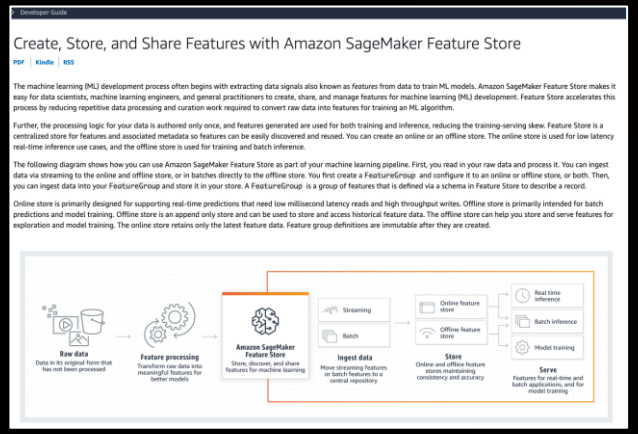
```
// Create feature group
FeatureGroupOperations.createFeatureGroups(
    sageMakerClient, FEATURE_GROUP_NAMES, FEATURE_GROUP_DESCRIPTION,
    onlineStoreConfig, EVENT_TIME_FEATURE_NAME, offlineStoreConfig,
    columnDefinitions, RECORD_IDENTIFIER_FEATURE_NAME,
    SAGEMAKER_ROLE_ARN);
```

Using feature store in a Java environment – link

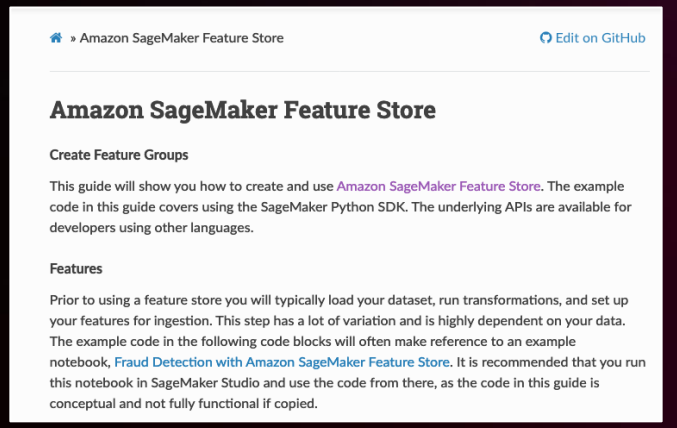
aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

SageMaker Feature Store documentation



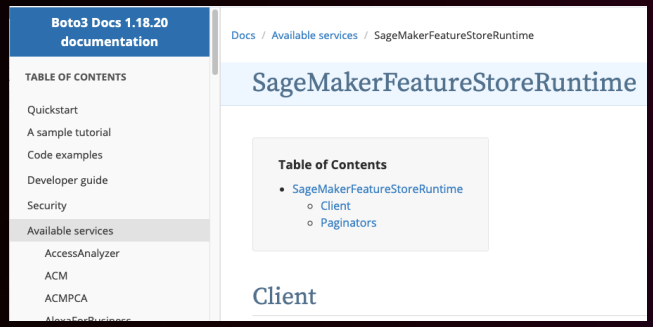
Developer guide
link



SageMaker Python SDK
link



Boto3 control plane API
link



Boto3 runtime API
link

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Mark Roy
roymark@amazon.com

Romi Dotta, PhD
romidat@amazon.com

Sergii Pylypchuk
sergii.pylypchuk@zalando.de



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.