
AWS
re:Invent

AIM320

Detecting fraud faster with Amazon SageMaker, featuring Coinbase

Erik Reppel
Engineering Manager, ML Platform
Coinbase

Soji Adeshina
Machine Learning Engineer
AWS

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Fraud detection
- using machine learning

Amazon SageMaker features
- model building
- model deployment

Building a fraud prevention system using AWS at Coinbase
- Model Development lifecycle
- Platform requirements
- What we built

Fraud detection solutions on Amazon SageMaker



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon SageMaker solutions for fraud detection

Get started in minutes

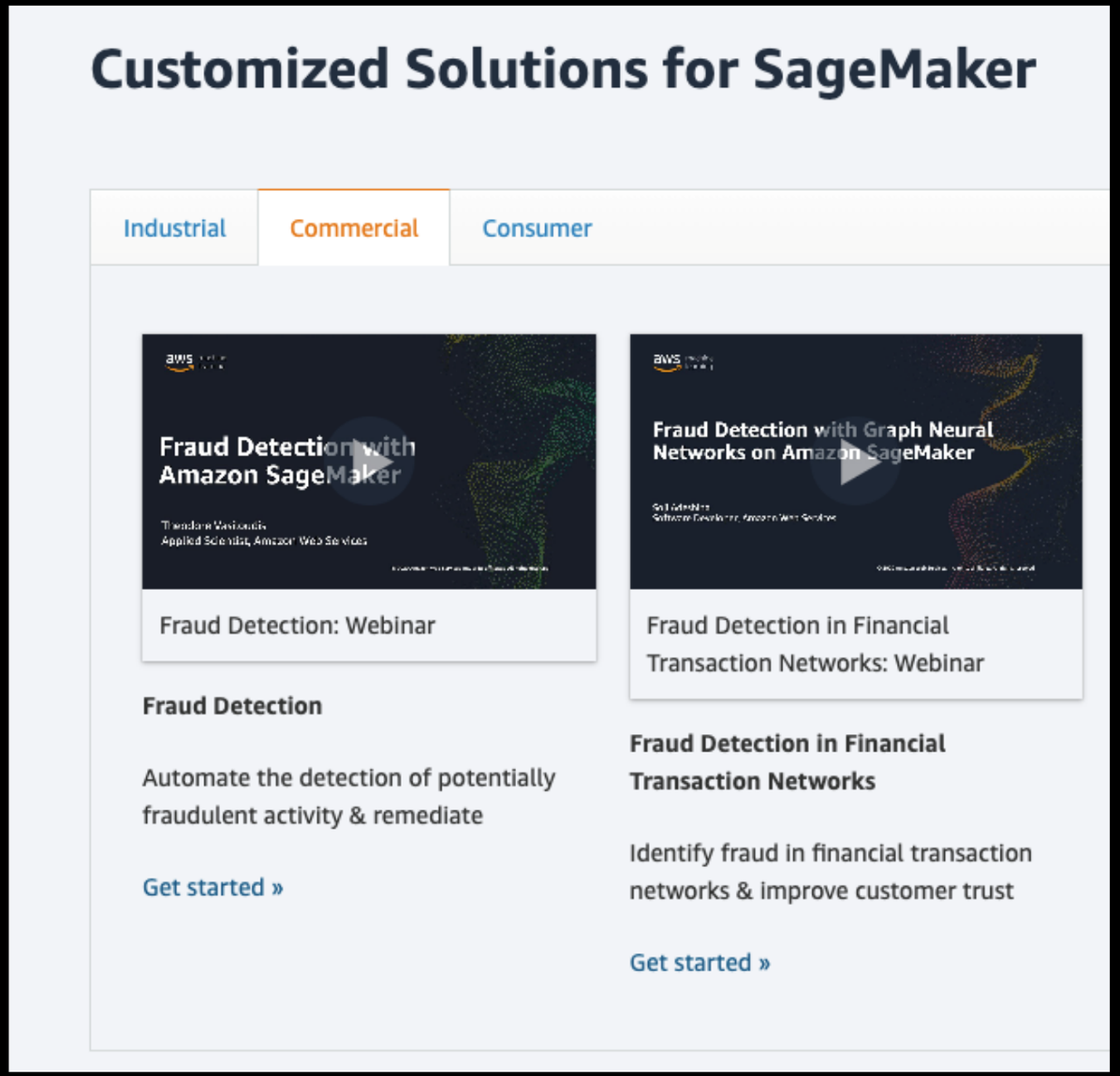
Leverage Amazon SageMaker and other AWS services

Build an end to end framework for fraud detection

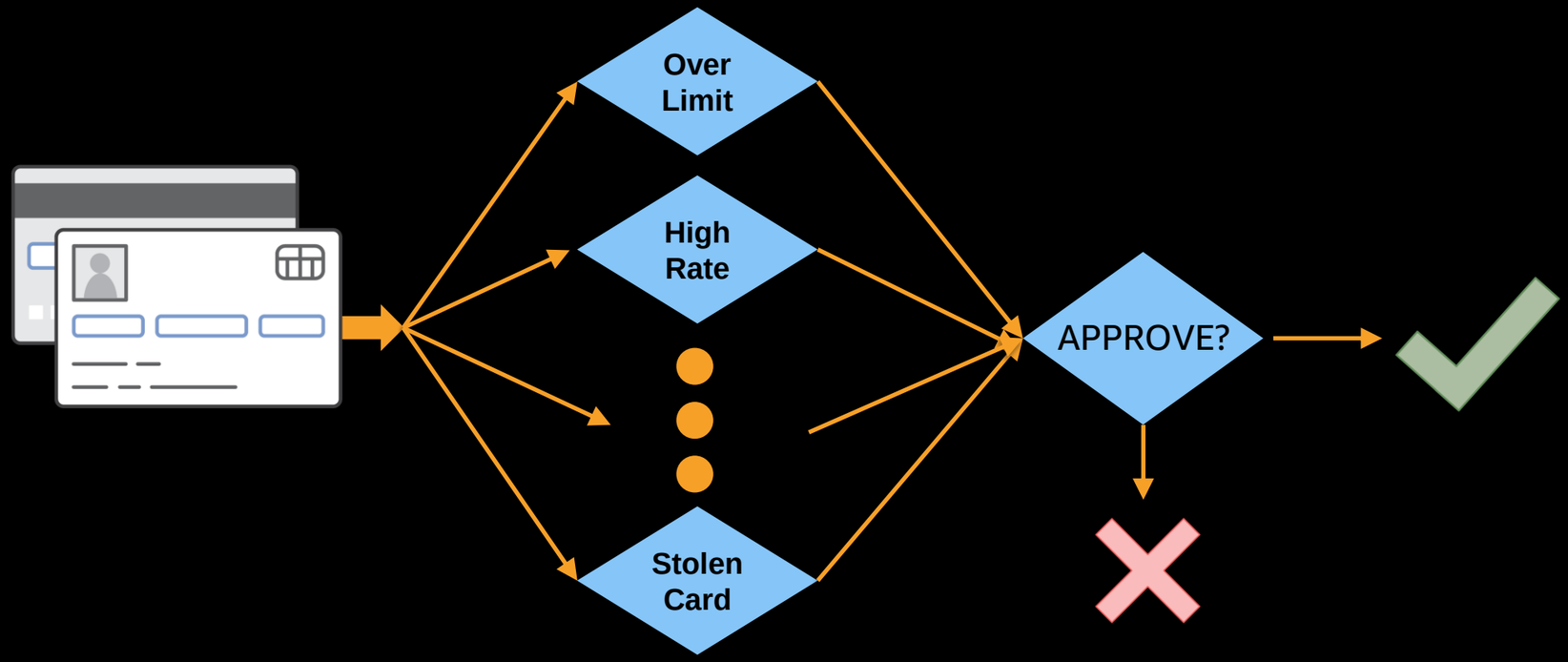
Flexible machine learning modelling techniques

Adapt to your needs and use-cases

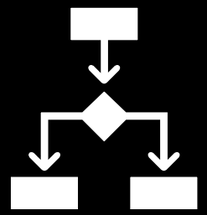
https://aws.amazon.com/sagemaker/getting-started/



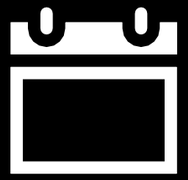
Typical approach to combat fraud: Rules based systems



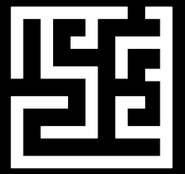
Rules based fraud detection - limitations



Static



Reactive

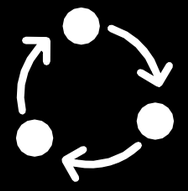


Complicated

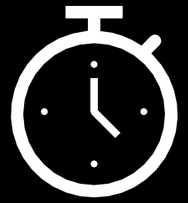


Scales poorly

Machine learning for fraud detection



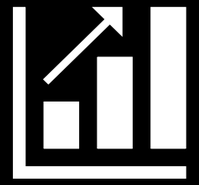
Dynamic



Real-time



Maintainable



Scalable

Fraud detection on AWS

Amazon provides customers with choice when considering implementing Fraud detection solution for their business



Amazon Fraud Detector

- Makes it quick and easy for developers and builders to implement AI and improve business outcomes
- No ML experience required



Amazon SageMaker

- 1P end to end solutions for fraud detection
- Build, train, and deploy your own ML models quickly using a visual IDE
- All tools required for ML in a single service
- Prebuilt models, algorithms, and solutions plus AutoML
- Fully managed for scalability, performance, and cost-effectiveness

Building Fraud detection models on Amazon SageMaker






Amazon SageMaker Autopilot

Automatically create machine learning models

Building Fraud detection models on Amazon SageMaker



Amazon SageMaker Built-in Algorithms

IP insights, XGBoost, Linear Learner, Random Cut Forest

Building Fraud detection models on Amazon SageMaker









Amazon SageMaker Training Toolkit

PyTorch, TensorFlow, Apache MXNet ,

Scikit-learn, Deep Graph Library

Building Fraud detection models on Amazon SageMaker









AWS Marketplace

Hundreds of algorithms and pre-trained models

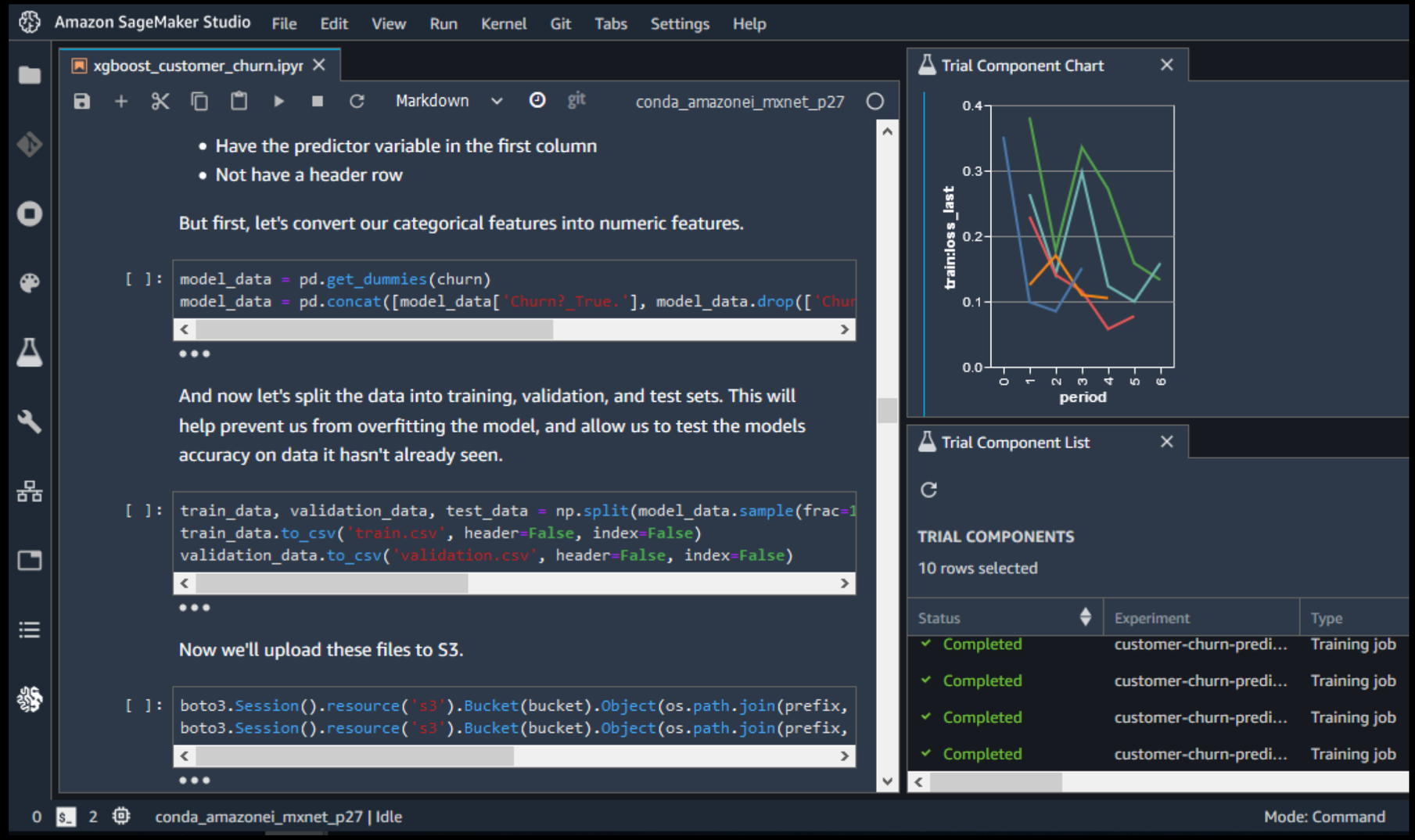
Amazon SageMaker features – Model development

SageMaker Studio

Integrated IDE for building models

SageMaker Experiments

Log and compare training runs



Amazon SageMaker features – Model deployment



Amazon SageMaker Model Monitor

monitor model predictions and detect drift



Amazon Augmented AI

add human review of model predictions

Amazon SageMaker features – Model deployment



Amazon SageMaker Model Monitor

monitor model predictions and detect drift



Amazon Augmented AI

add human review of model predictions

Building a fraud prevention system at Coinbase using SageMaker and AWS

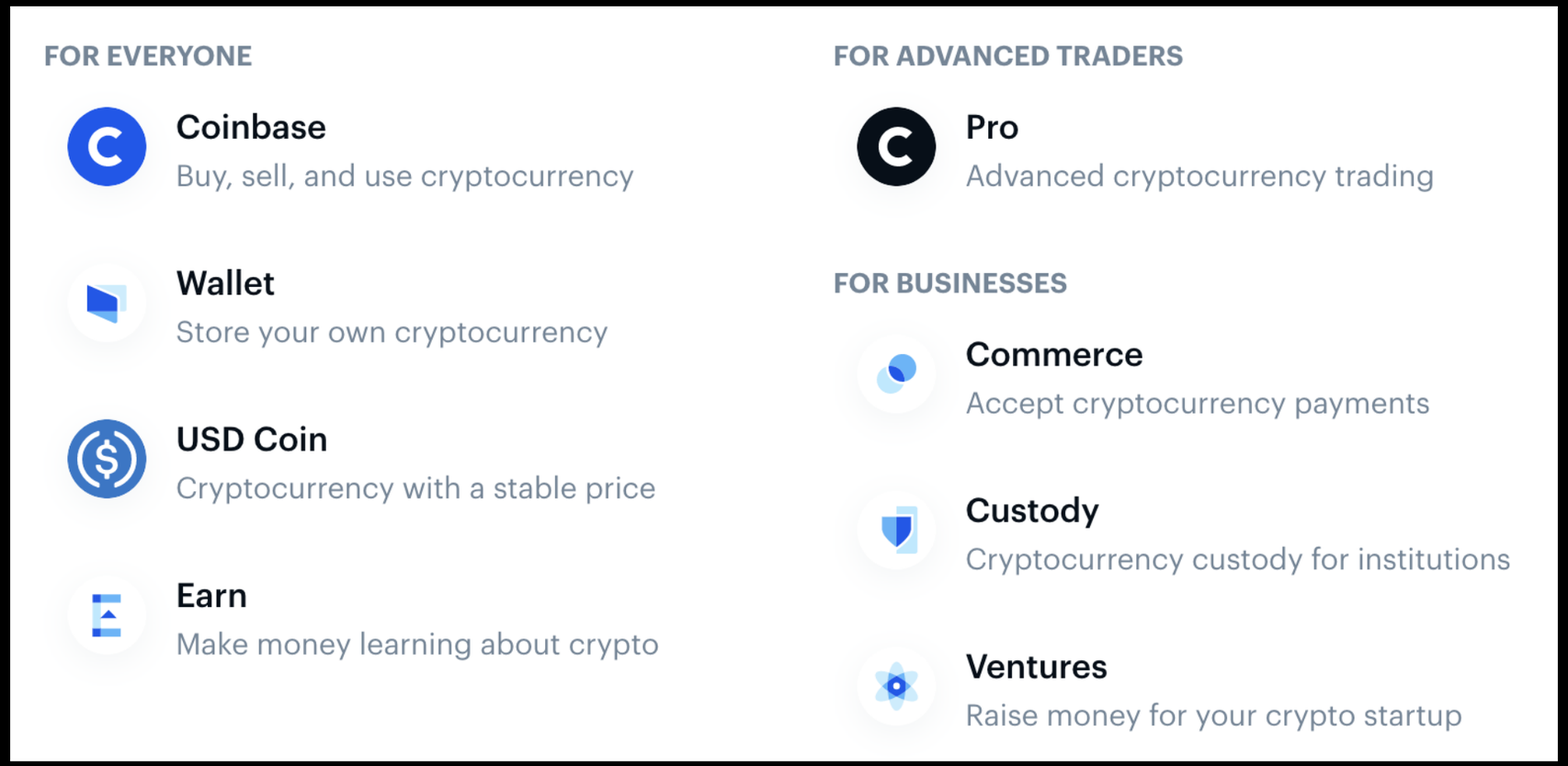


AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

About Coinbase

- Stewards more than 35 million accounts
- Total trading volume since launch in 2012: $320B+
- Two main priorities: be the most trusted and easiest to use



Combating fraud @ Coinbase

- Sophisticated attackers
- High stakes
- Fraud can occur in both the fiat and crypto worlds
- Blockchains are immutable, limited recourse
- Strong Real time requirement

Combating fraud @ Coinbase

Risk Analysts

High skilled specialists in fraud prevention

Rules Engine

Developed by analysts using data

ML Models

Models are developed in house by ML Engineers

Overall impact: Contributed to 50% reduction in fraud rate

Coinbase fraud prevention ML models

- User risk model
  - *Ensemble of CatBoost and Pytorch Transformer model for event sequences*
- Transfer Risk model
  - *CatBoost Model with sequence aggregate features*
- Credit risk model
  - *CatBoost Model with sequence aggregate features*
- Account takeover model
  - *Ensemble of CatBoost and Pytorch Transformer model for event sequences*

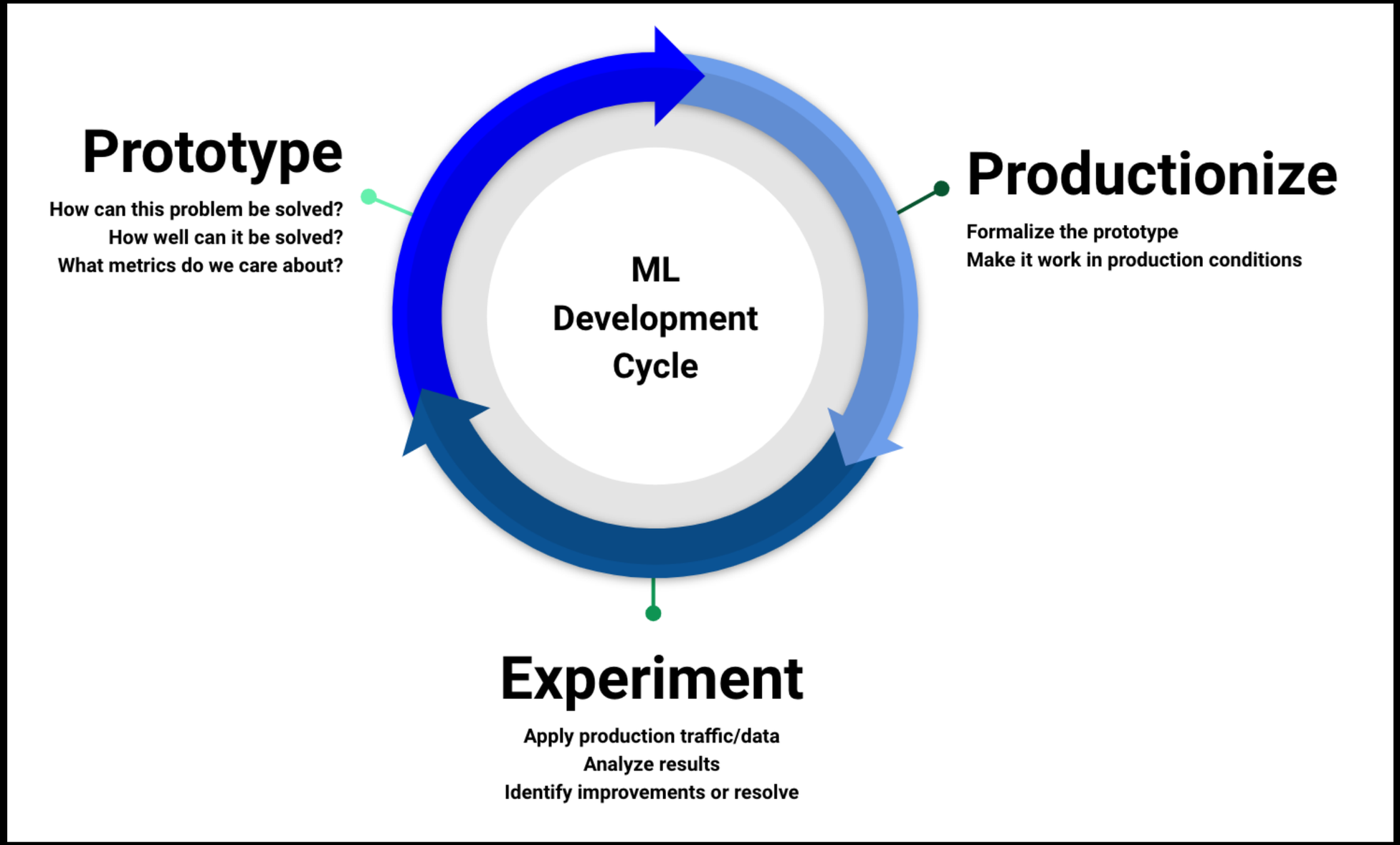
Coinbase model development lifecycle and platform requirements



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Model development lifecycle



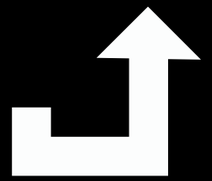
Coinbase ML Fraud Detection Platform requirements



Secure



Low Latency



Flexible

Coinbase ML Fraud Detection Platform requirements



Secure



Low Latency



Flexible

Access limited to those who need it

No unreviewed code deployed to production

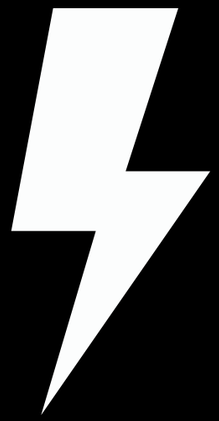
Consensus mechanisms to guard against bad actors

Meet Coinbase strict security requirements

Coinbase ML Fraud Detection Platform requirements



Secure



Low Latency



Flexible

ML Models are blocking in critical UX paths requiring SLAs as low as p99 of 80ms e2e latency

Data must be as up to date as possible, feature vectors user latest actions

Sophisticated fraudsters know the site well, often want a transactional guarantee (read from app db)

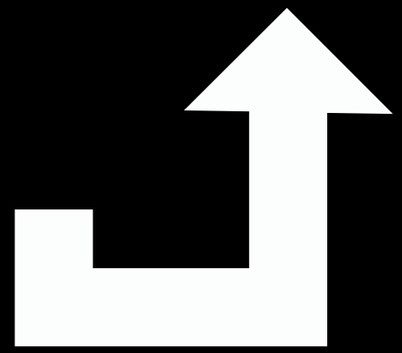
Coinbase ML Fraud Detection Platform requirements



Secure



Low Latency



Flexible

Using bleeding edge tech from academia gives us an advantage

Keep time from concept to production as low as possible

Don’t be coupled to any 1 model framework Ex: using CatBoost over XGBoost gave a 4% AUC boost,

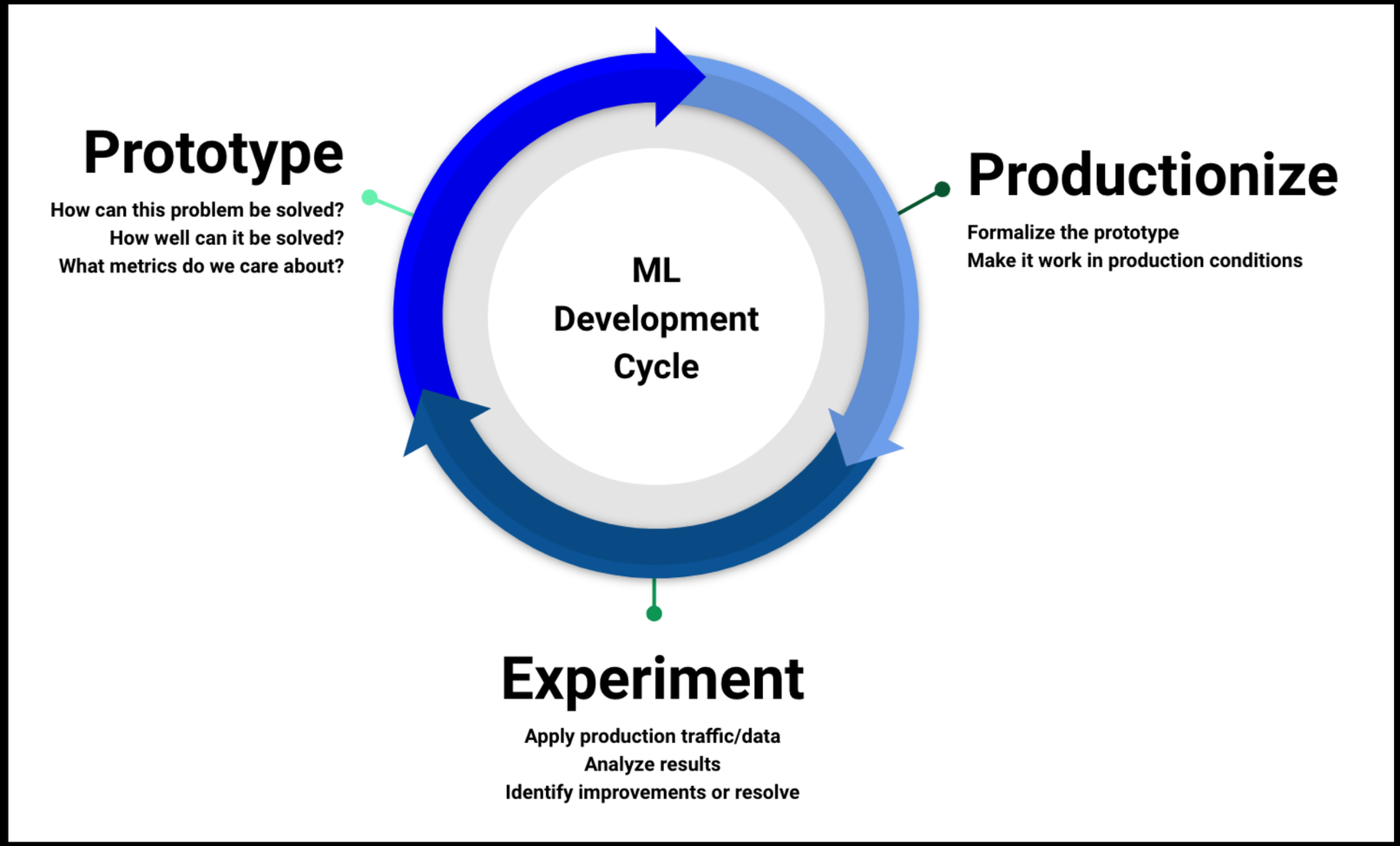
“Be able to deploy any model we need, to solve problems for the business”

In summary

Requirements

- Secure
- Low Latency
- Flexible

Workflow

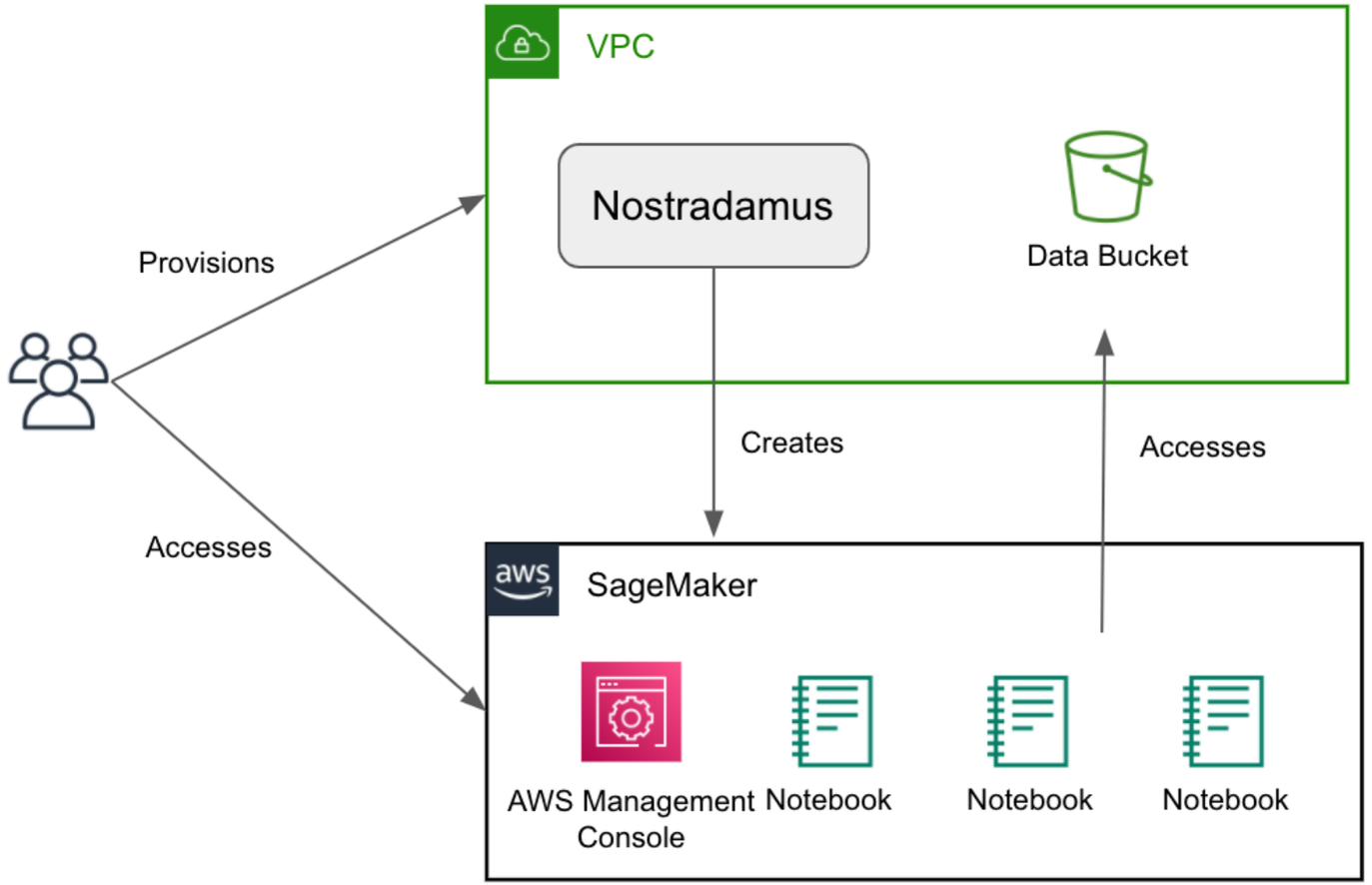


What we built



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Prototype using Jupyter Notebooks on Amazon SageMaker



- Notebook accessed via Roles
- VPC peered with required CB infra
- Bucket access tightly controlled
- Codified configs / security

Productionizing models

cbml BaseModel - our abstraction over the lifecycle of a model

```
from cbml.models import BaseModel

class NewModel(BaseModel):
    def train(self, data: pd.DataFrame) -> Any:
        ...

    def predict(self, data: pd.DataFrame):
        ...

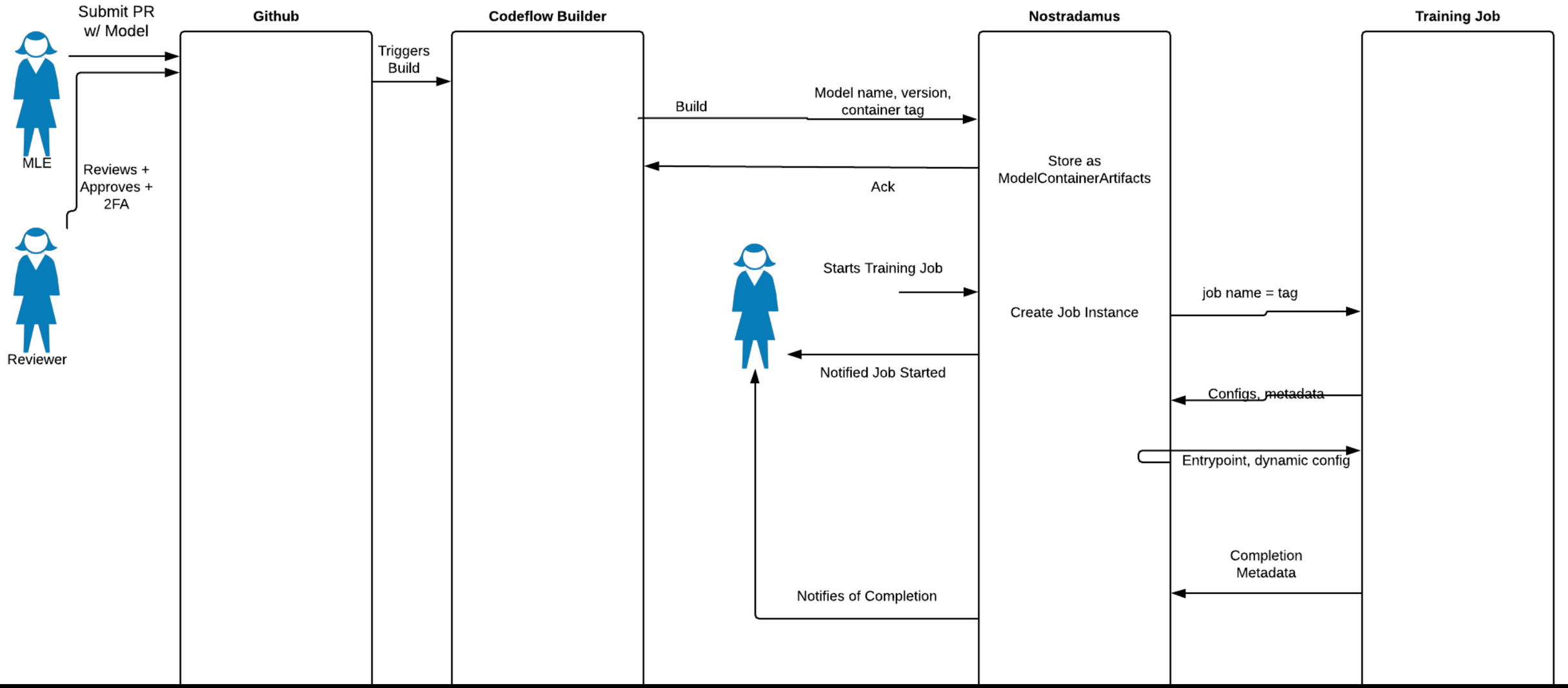
    def performance_summary(self, data: Optional[pd.DataFrame] = None) -> dict:
        ...

    def load(self, state):
        ...
```

Training models in production using Amazon SageMaker

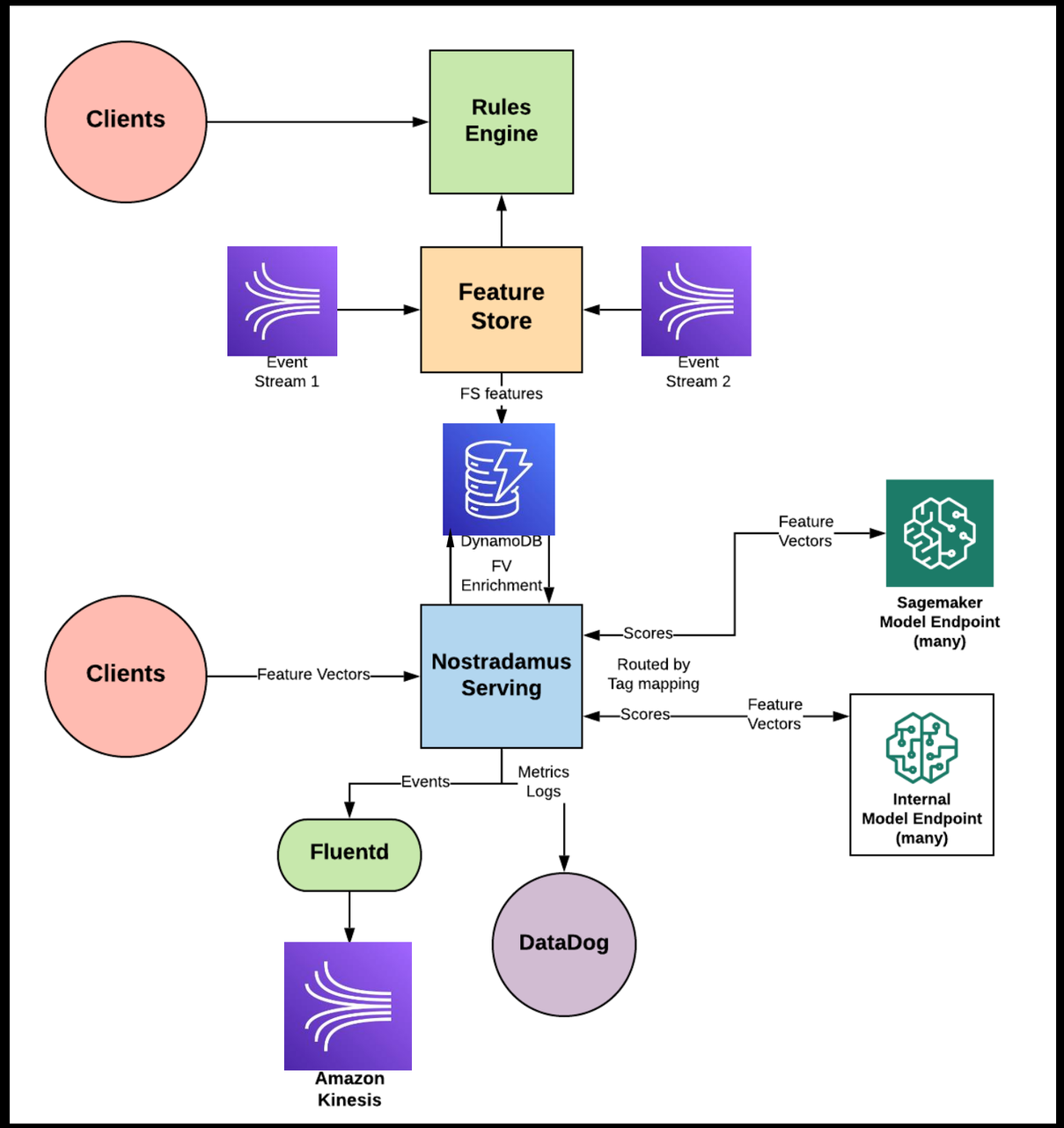
- Amazon SageMaker Bring Your Own Container for all training
- Data & model artifacts out of leverage
- Metadata & performance stored in Nostradamus control plane
- Only reviewed model code in production jobs
- Only artifacts from production jobs used for production inference

Checking in models and achieving consensus



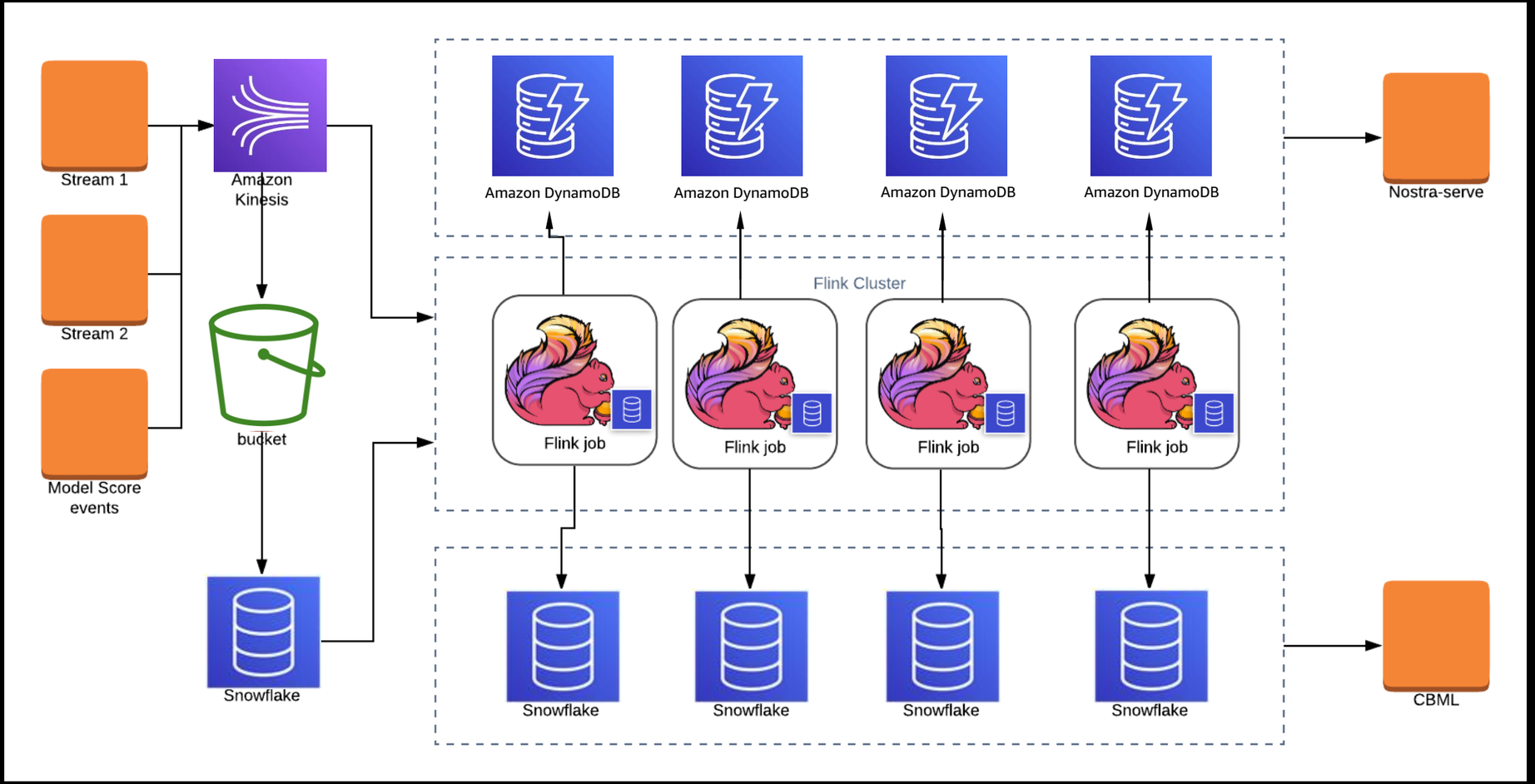
Deploying real time models

- Models deployed as Amazon SageMaker Endpoints with training containers
- Score requests proxied through central gateway (Nostradamus Serving)
- Same BaseModel + training job container used for inference



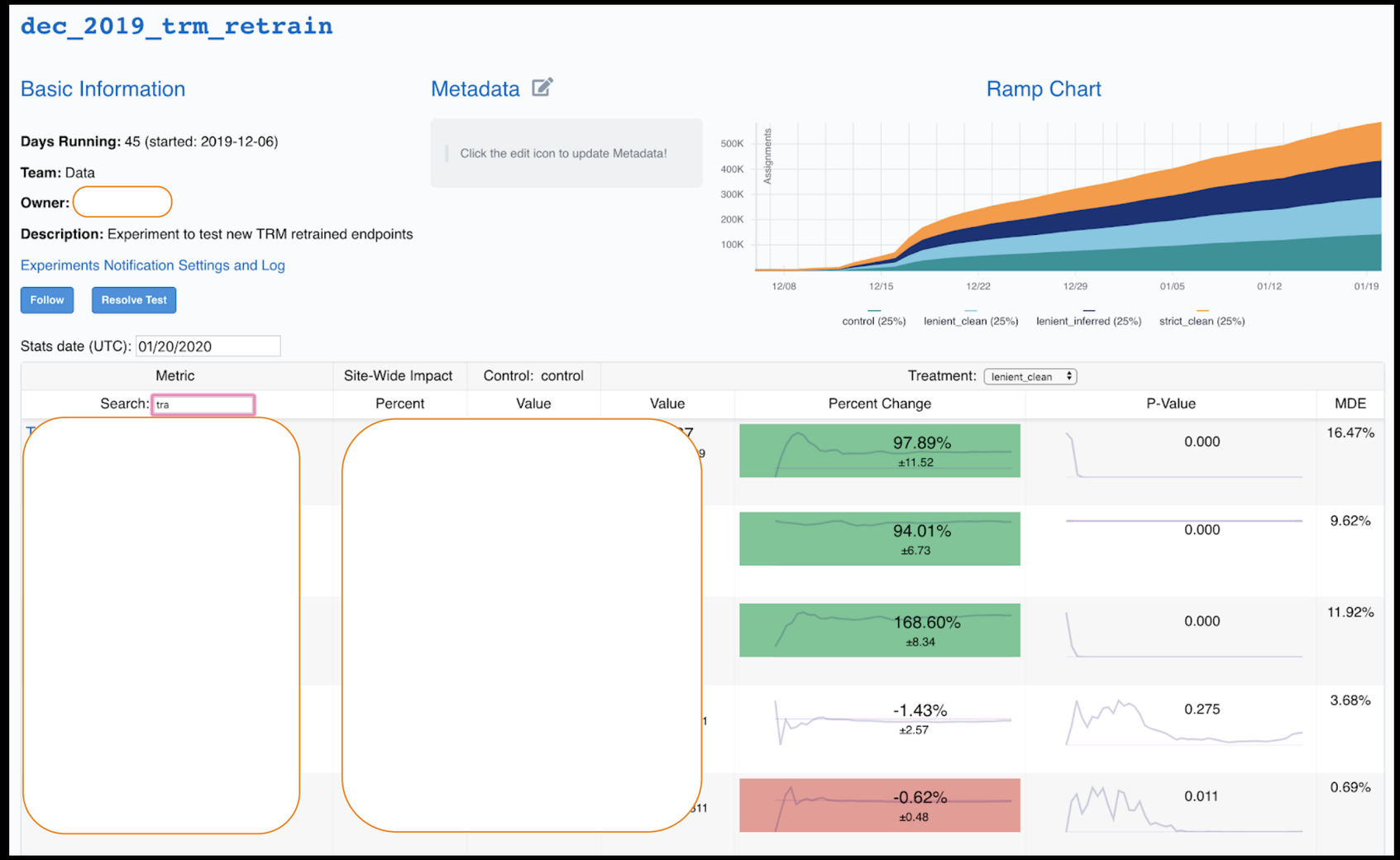
Feature store

- Streaming features using Flink via AWS EMR
- Computed off event stream, stored in DynamoDB for real time inference
- Feature Vectors replicated to S3 / Snowflake for training
- Merged with analytic and real time features using entity IDs



Experimentation

- All model rollouts are strictly managed by experimentation
- Impact measured against target and guardrail metrics
- Multiple SageMaker endpoints deployed and experimented over



Summary



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Meeting Coinbase ML Fraud Detection Platform requirements

- **Secure**
  - Amazon SageMaker Notebooks provide us a secure way to prototype models
  - Amazon SageMaker training jobs with BYOC fits our high security standards
- **Low-Latency**
  - Amazon SageMaker endpoints respond in real time, models can be optimized
- **Flexible**
  - BYOC gives us the freedom to use any modeling approach we want
  - Use Amazon SageMaker to create endpoint and deploy any model
  - Endpoints can be integrated with internal Feature Store for powerful, cutting edge modeling techniques

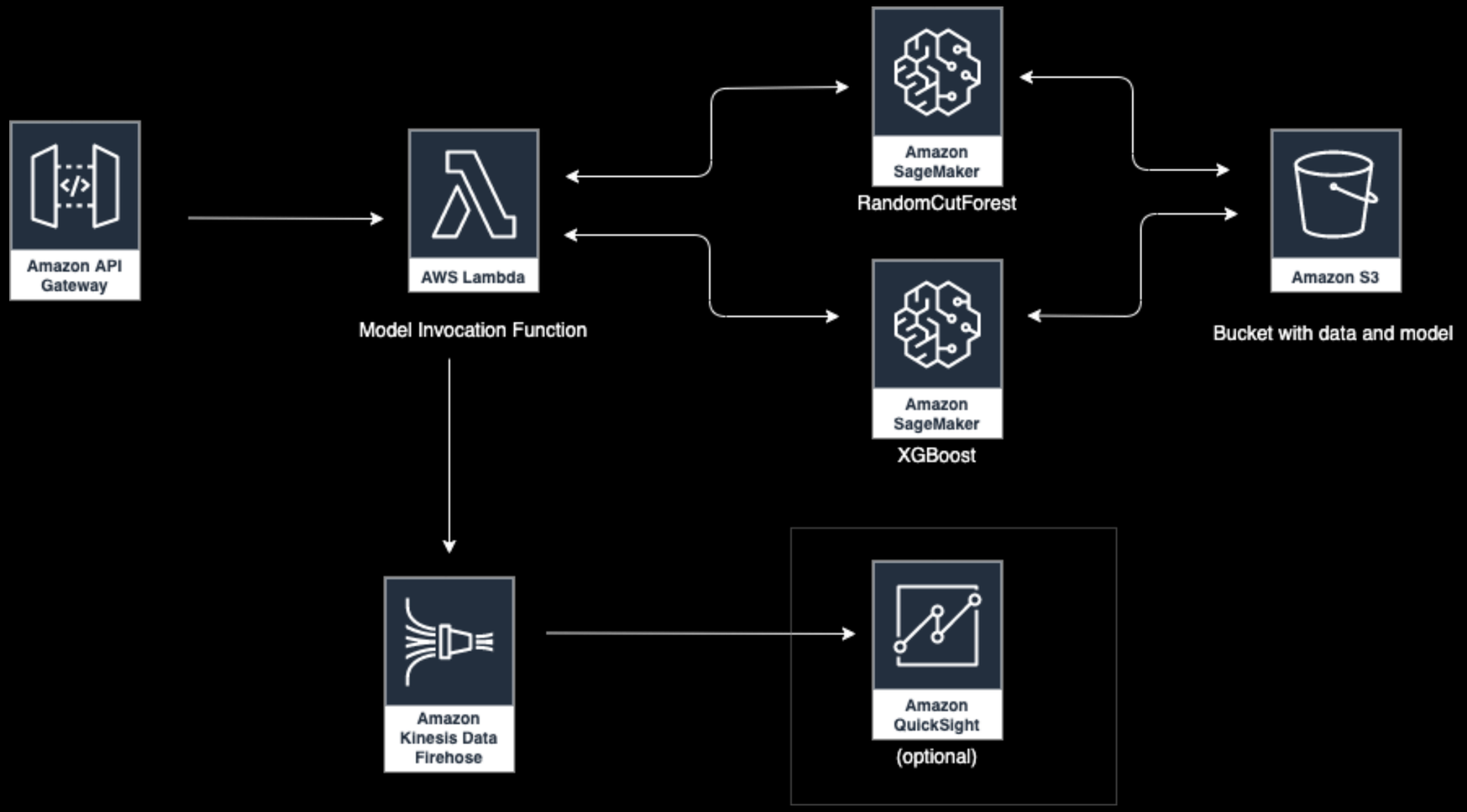
Amazon SageMaker solutions for fraud detection



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

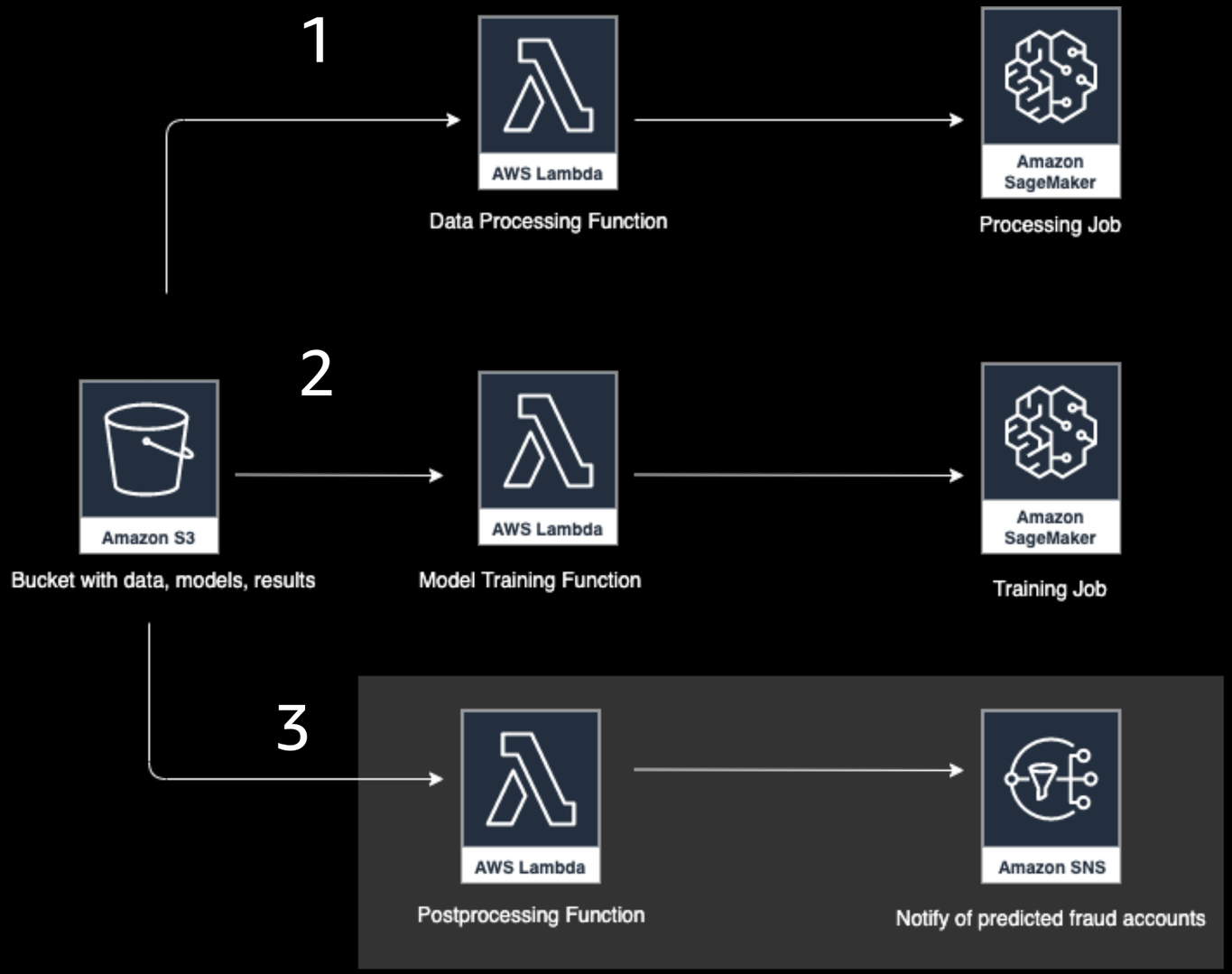
SageMaker solution – Fraud detection on tabular data



- Supervised and unsupervised learning
- Deploy real-time endpoint

https://github.com/awslabs/fraud-detection-using-machine-learning

SageMaker solution: Fraud detection using Graph Neural Networks



Leverage interaction network

Custom graph neural network model

https://github.com/awslabs/sagemaker-graph-fraud-detection

Thank you!



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.