



AI & Machine Learning

Prevent online fraud while ensuring a frictionless customer experience

Anjan Biswas

Sr. AI Services Solutions Architect, AWS

Fraud Leads to Friction

SURGE OF FRAUDULENT ACCOUNT REGISTRATIONS



Fake credentials, stolen identities,
and bots make creating
fake accounts fast and easy

SPIKE IN FRAUD ALERTS INCREASES ACCOUNT REVIEWS



Customer service teams,
overwhelmed by alert volumes,
struggle to keep up

FRICTION ADDED TO STOP FRAUDULENT REGISTRATIONS



Friction slows and lowers
account registrations
from legitimate customers

Use Case: Coupon Shopper



A Digital Platform

An online rewards platform that allows users to earn Gift Cards, rewards and coupons for online shopping.



Growing User Base

A growing user base with currently ~50k monthly active users. Awards millions of dollars worth of Gift Cards and coupons annually.

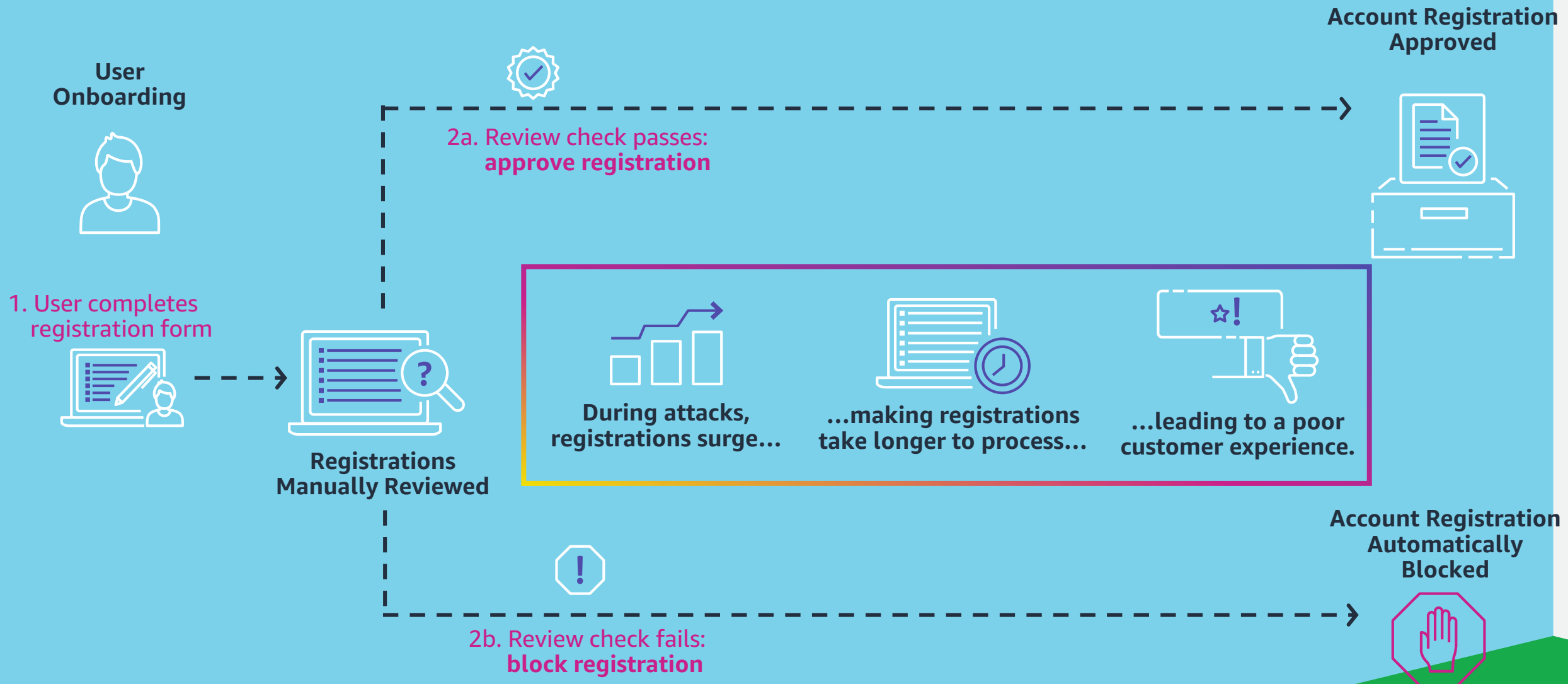


Increasing Online Fraud

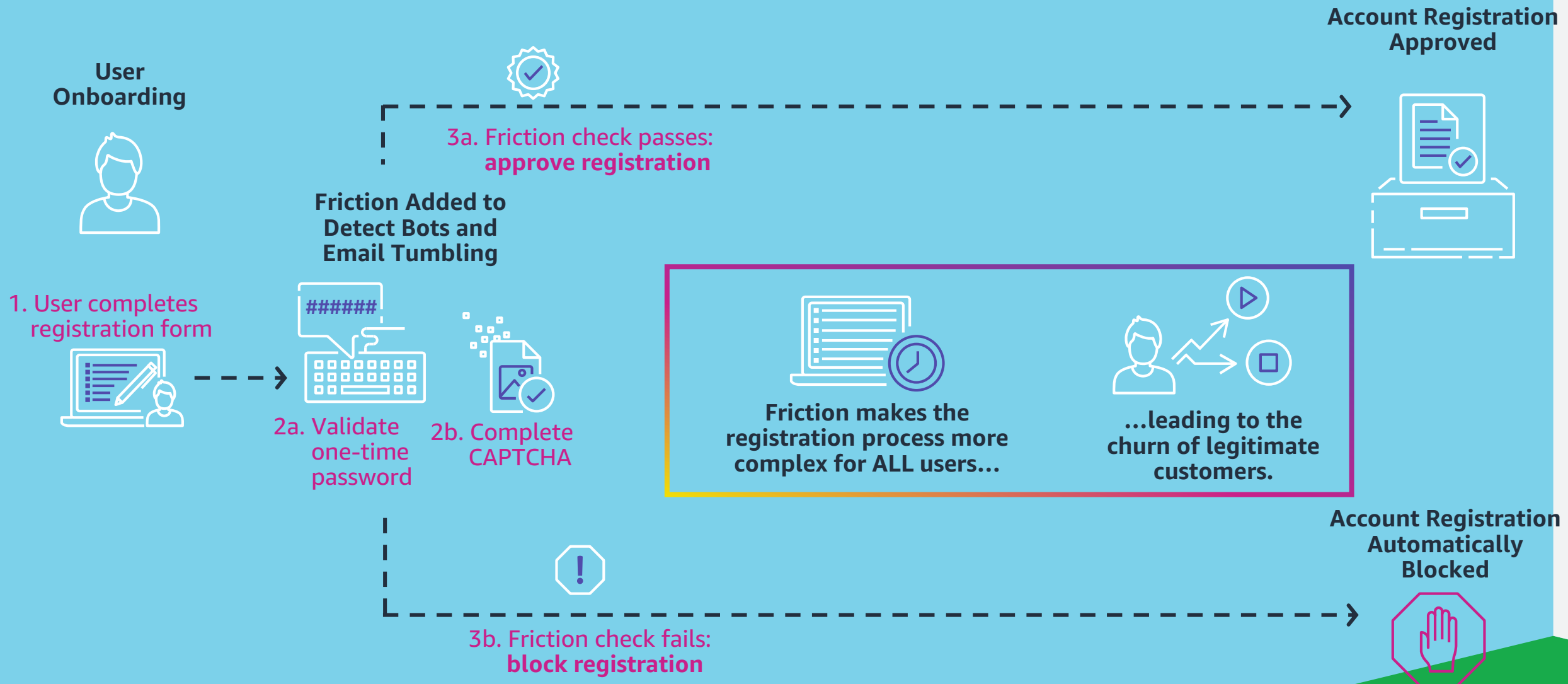
Increasing fraudulent user registrations from bad actors, automated accounts/bots, who try to game the system to earn Gift Cards and coupons.

**Net annual loss of
~\$1.5 million**

User Onboarding: with manual review

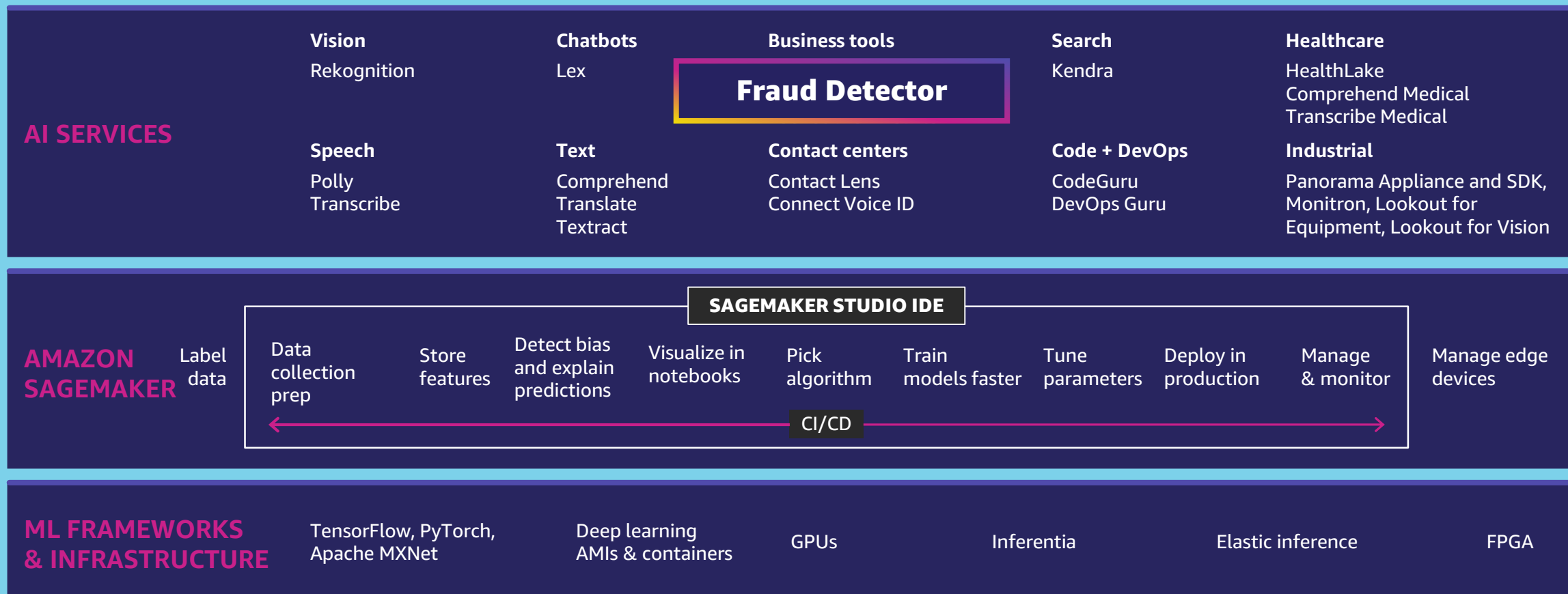


User Onboarding: with automated friction



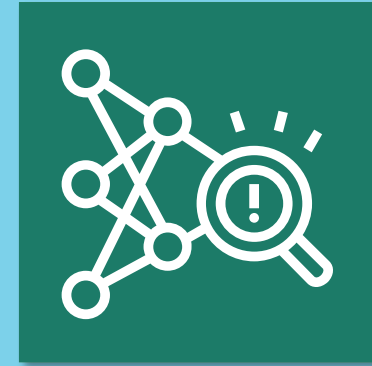
The AWS ML stack

Broadest and most complete set of machine learning capabilities



Amazon Fraud Detector

A fully managed service that makes it easy to use machine learning to detect potential online fraud in real-time, at scale

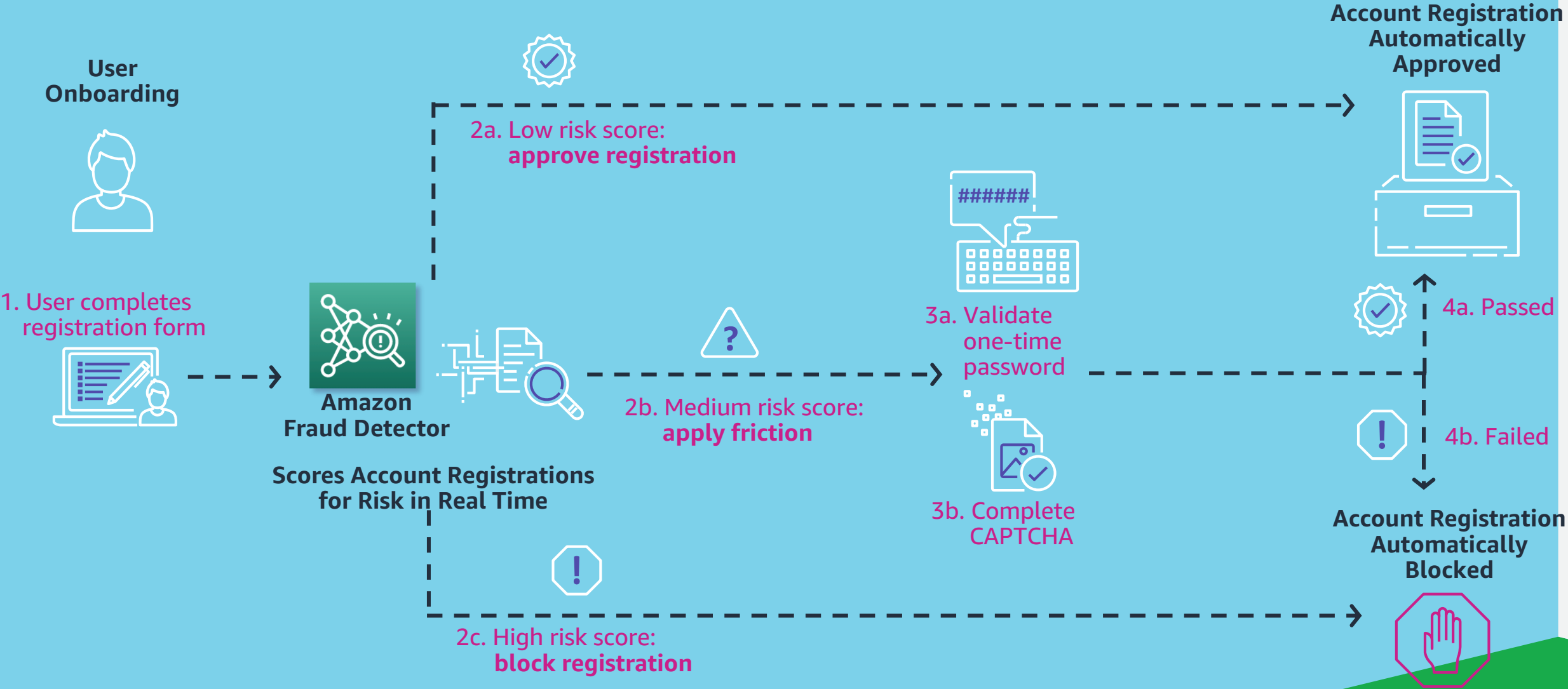


Two Model Types

Online Fraud Insights

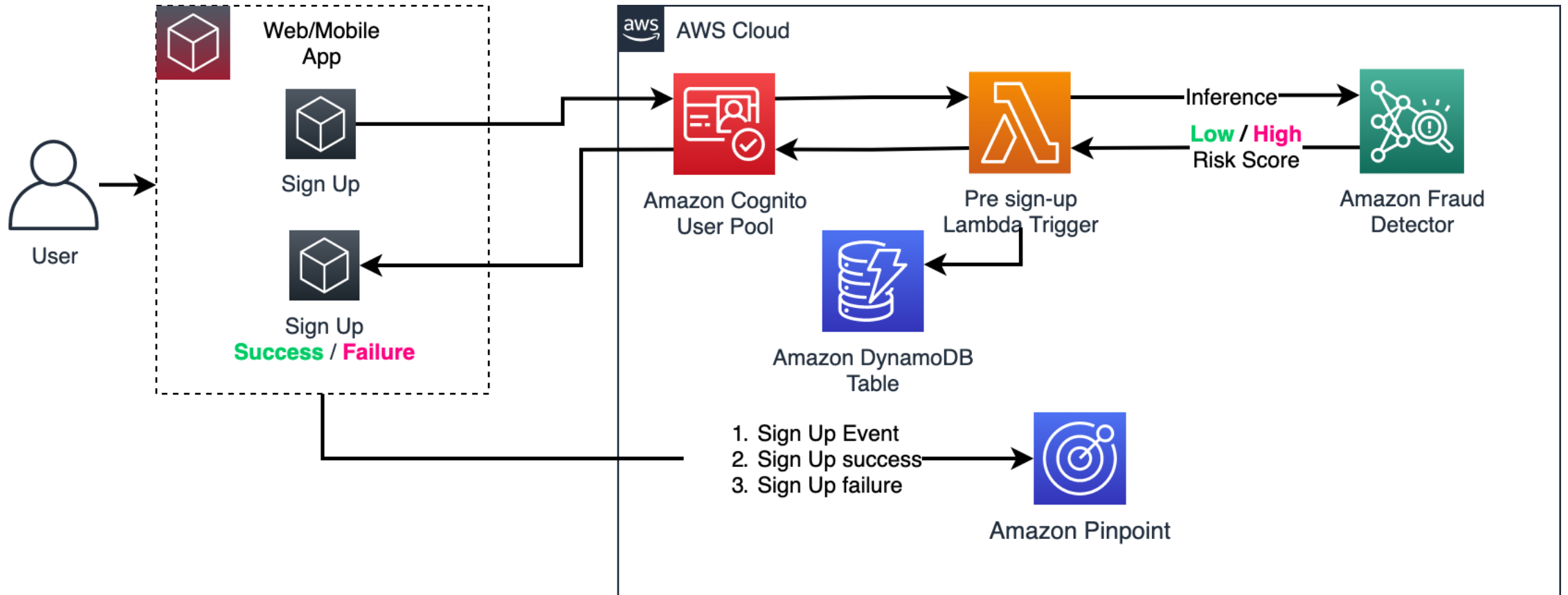
Transaction Fraud Insights

User Onboarding: with Amazon Fraud Detector

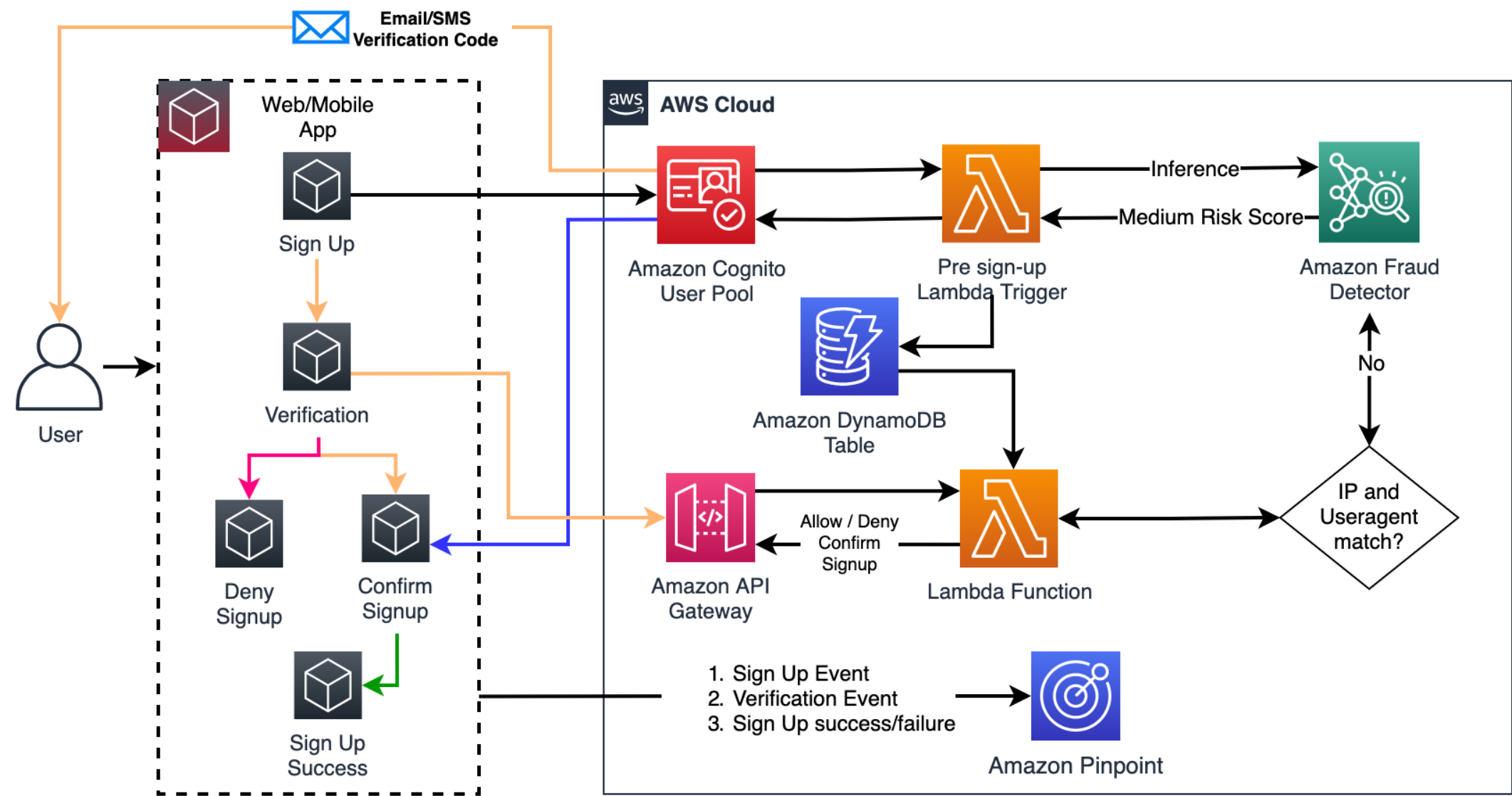


Coupon Shopper Demo

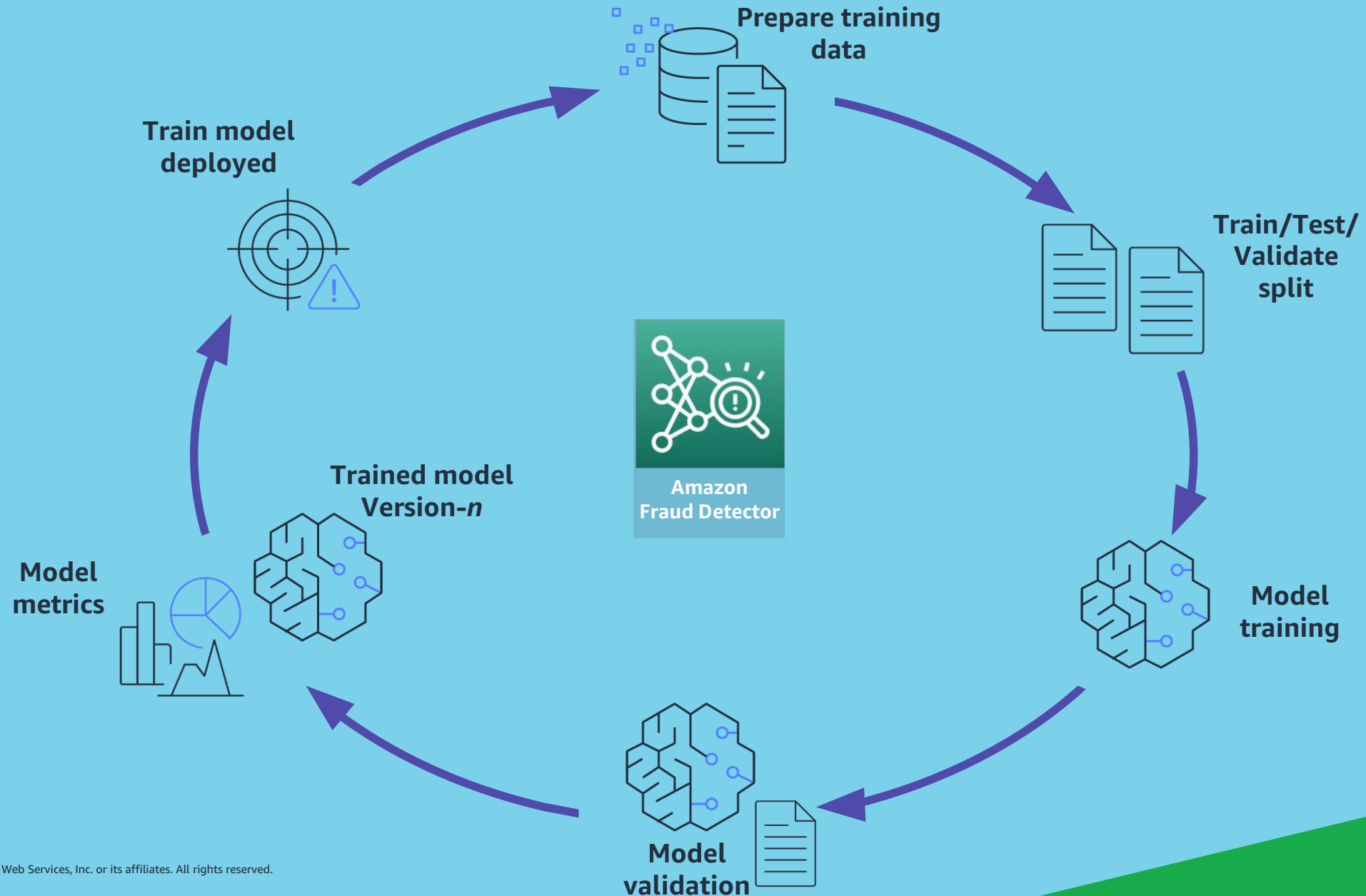
Solution Architecture



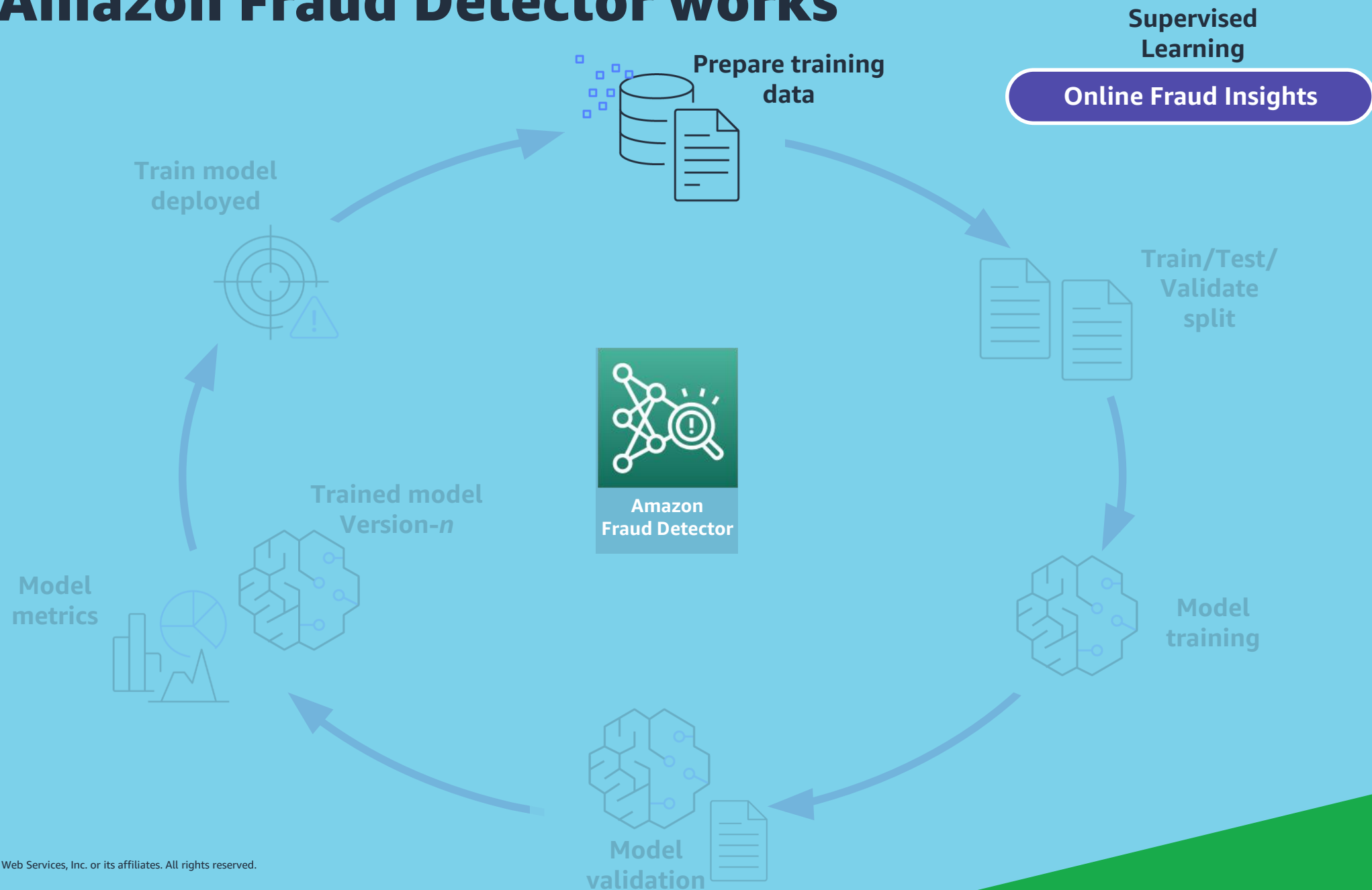
Solution Architecture



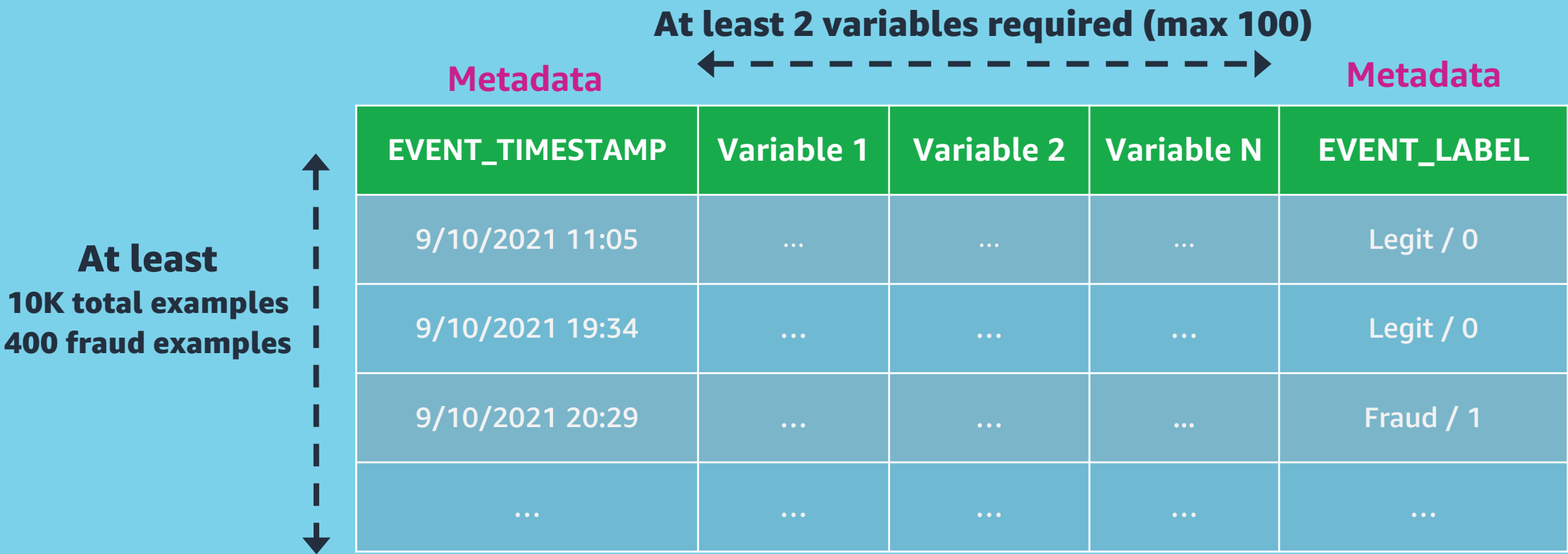
How Amazon Fraud Detector works



How Amazon Fraud Detector works



General Training Data Guidelines



- Data must be in **CSV** format – store in Amazon S3 bucket or stream directly.
- 2 required **Event Metadata** variables for **Online Fraud Insights**: **EVENT_TIMESTAMP** and **EVENT_LABEL**
- Recommend at least **3-6 months** of data for training the model
- Too many NULLs and missing values in a column means **it won't be used**

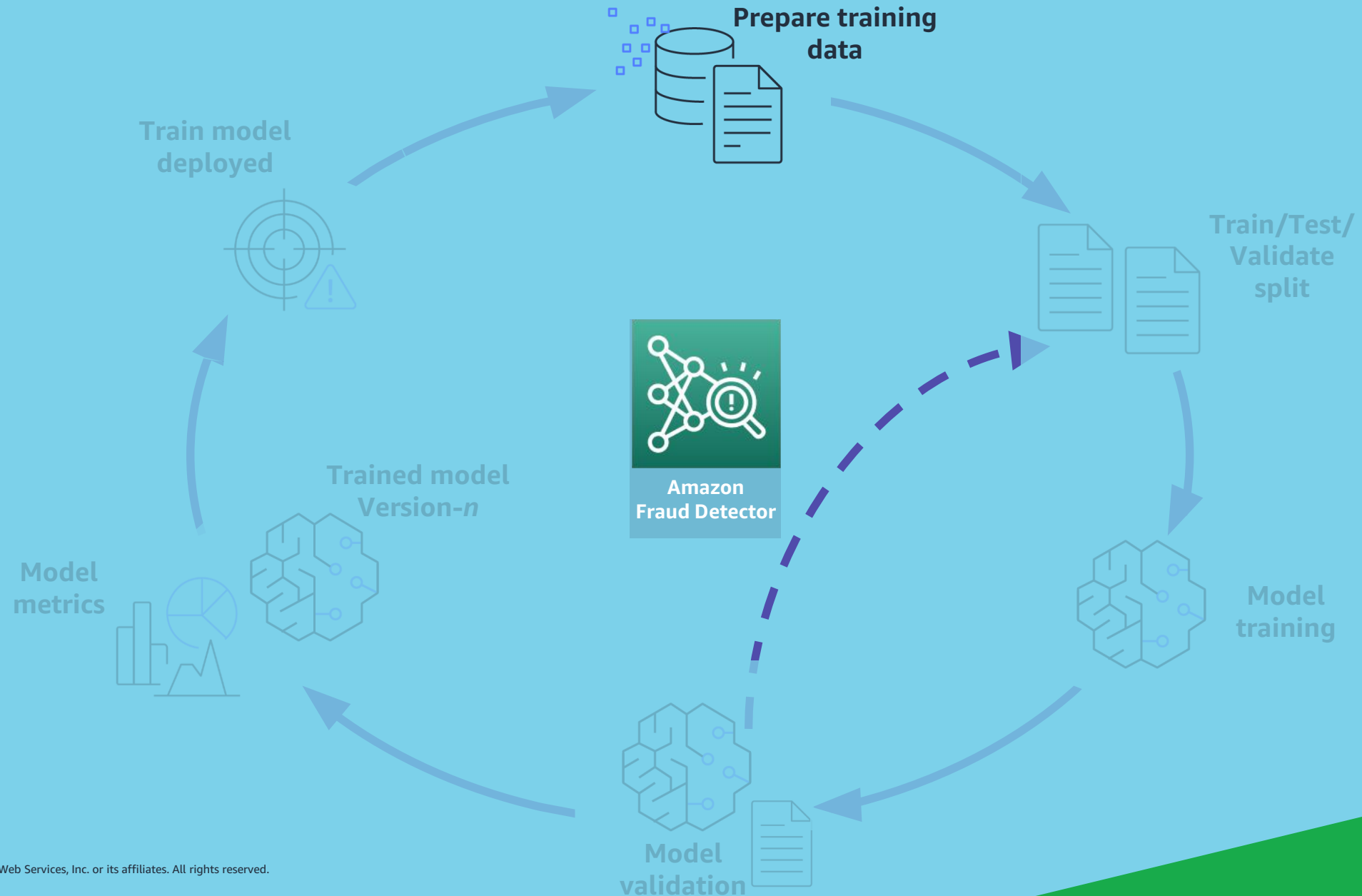
Sample Data

Mandatory variables

6 Additional variables

EVENT_TIMESTAMP	EVENT_LABEL	IP_ADDRESS	EMAIL	PHONE	ADDRESS	STATE	POSTAL
4/29/19 8:34	fraud	192.94.136.134	synth_crossdaniel@gmail.com	(555)715 - 6254	43424 Davis Inlet Apt. 764	NY	32155
4/29/19 8:41	fraud	192.31.192.222	synth_wilsonjustin@yahoo.com	(555)740 - 7222	55403 Jackson Hill	ND	32843
4/29/19 8:43	fraud	192.174.61.17	synth_julie09@hotmail.com	(555)514 - 6454	68642 Leslie Rest Suite 087	NY	34109
4/29/19 8:48	legit	192.88.115.133	synth_max43@hotmail.com	(555)303 - 9960	403 Linda Parkway	KS	33965
4/29/19 8:51	legit	73.60.135.213	synth_billy51@gmail.com	(555)992 - 9688	2566 Henderson Creek Apt. 138	LA	32156
4/29/19 9:28	legit	198.51.101.133	synth_qharris@hotmail.com	(555)672 - 1421	7295 Peterson Ports	ND	33090
4/29/19 9:30	legit	181.20.252.125	synth_john02@yahoo.com	(555)463 - 8749	2404 Huff Pine Apt. 018	MD	34109
...
5/2/19 7:36	legit	123.143.40.137	synth_ashley90@hotmail.com	(555)200 - 9565	64680 Solomon Causeway	LA	32678
5/2/19 7:38	legit	192.88.107.202	synth_barrettaskley@hotmail.com	(555)562 - 3998	4556 Jordan River Suite 917	AK	32967
5/2/19 7:47	fraud	192.175.40.168	synth_colemariah@gmail.com	(555)491 - 8229	046 Joel Turnpike Suite 872	CA	34205

How Amazon Fraud Detector works



Assessing Model Performance

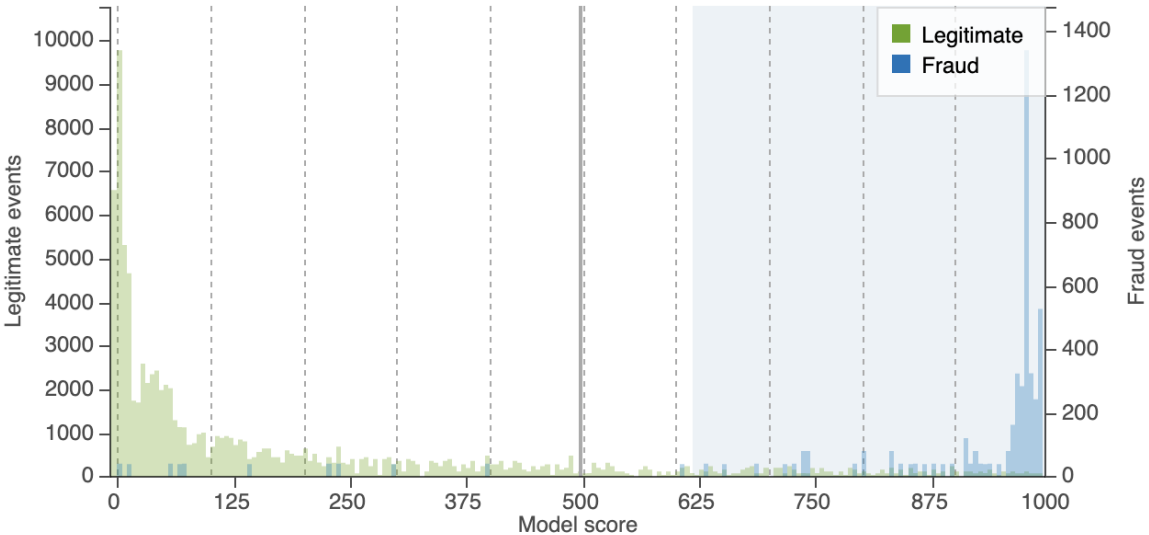
Model performance **AUC 0.95**

Charts

Table

Score distribution

By writing a rule using a model score threshold of **500**, you will succeed in catching **91.9%** of all fraudulent events (TPR) while accepting a risk that **12.7%** of legitimate events are incorrectly labeled as fraud (FPR).



Click anywhere on the chart above to select a model threshold score and determine the TPR and FPR.

Confusion matrix


For the selected model score threshold, the confusion matrix represents the expected outcome given 100,000 sample events (95,012 legitimate, 4,988 fraud).

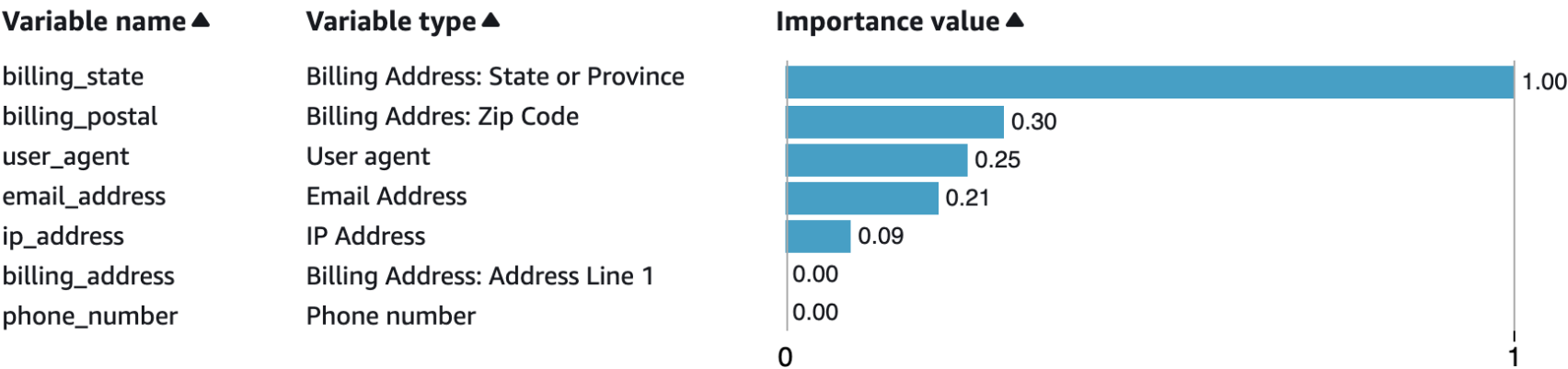
		Predicted		
		Fraud	Legitimate	
Actual	Fraud	True positive 4582	False negative 406	TPR 91.9%
	Legitimate	False positive 12084	True negative 82928	FPR 12.7%

Numbers are based on a sample of 100,000 events.

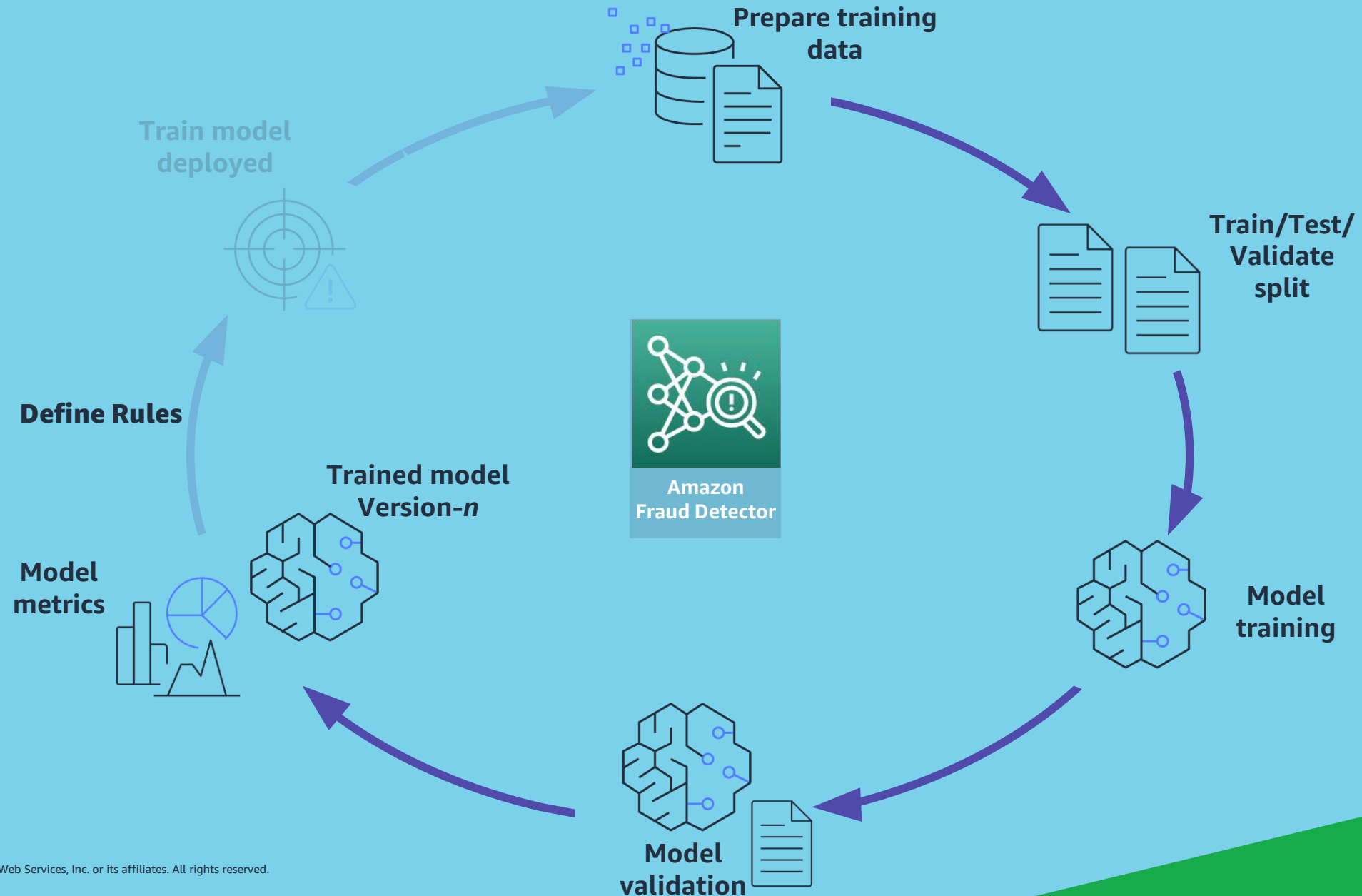
Assessing Model Performance

Model variable importance - *new*

Variable Importance gives you an understanding of how each variable is contributing to your model's performance. The chart below lists model input variables in the order of their importance to the model, indicated by the number. A variable with a much higher number relative to the rest could indicate that the model might be overfitting on it, while variables with relatively lowest numbers could just be noise. For details, [refer to documentation](#) 

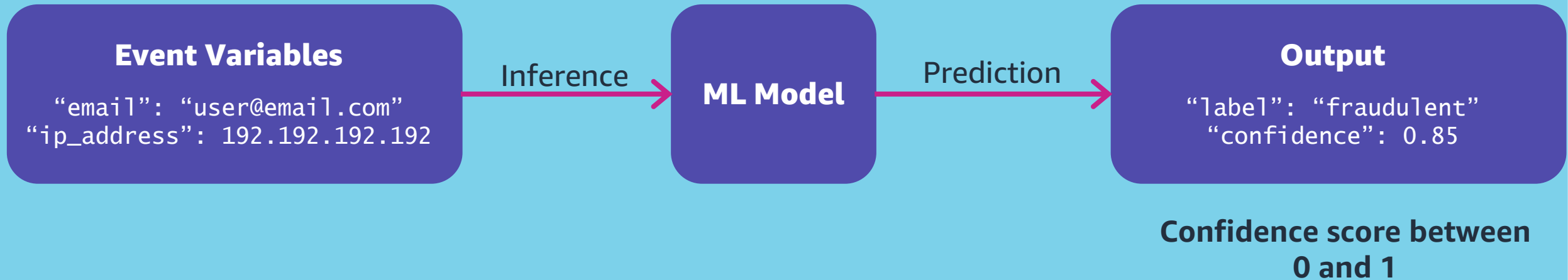


How Amazon Fraud Detector works



Defining Rules

A typical ML model outcome



Defining Rules

Version details

Model name

sample_fraud_detection_model

Model type

Online Fraud Insights

Performance (AUC)

0.95

Event type

sample_registration

Status

Active

Date created

Sun, Sep 19, 2021, 10:27 AM PDT

Output variable

sample_fraud_detection_model_insightscore

ARN

arn:aws:frauddetector:us-east-2:576090071164:model-version/ONLINE_FRAUD_INSIGHTS/sample_fraud_detection_model/1.0

Model Score	Estimated FPR
975	0.05%
950	1%
900	2%
860	3%
775	5%
700	7%
600	10%

Rule language

\$variable < 100

\$variable in [5, 10, 25, 100]

\$variable != "US"

\$variable == 1000

Associated rules (3)

The rules listed below have been added to the sample_detector detector

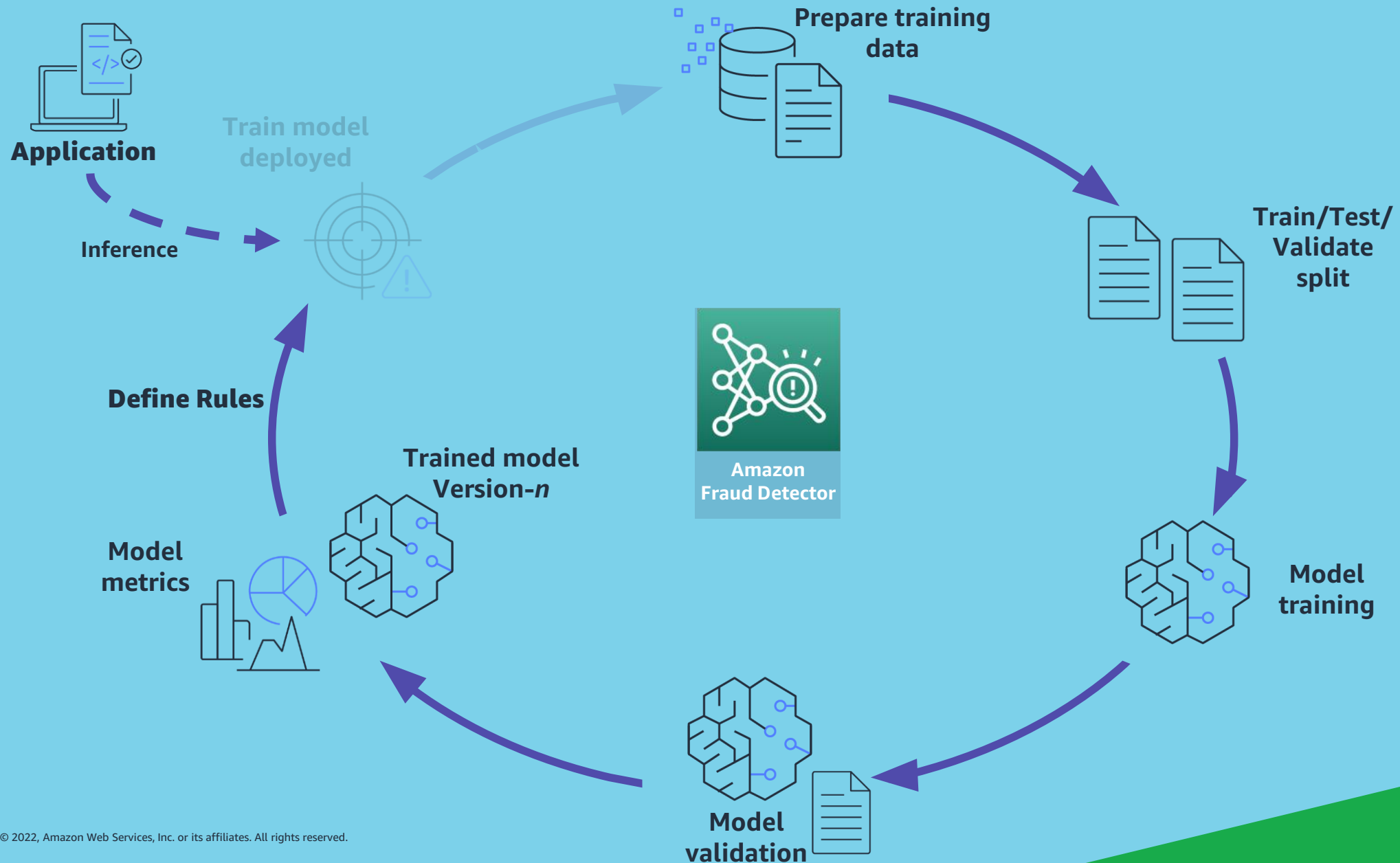
Find rules

Create rule

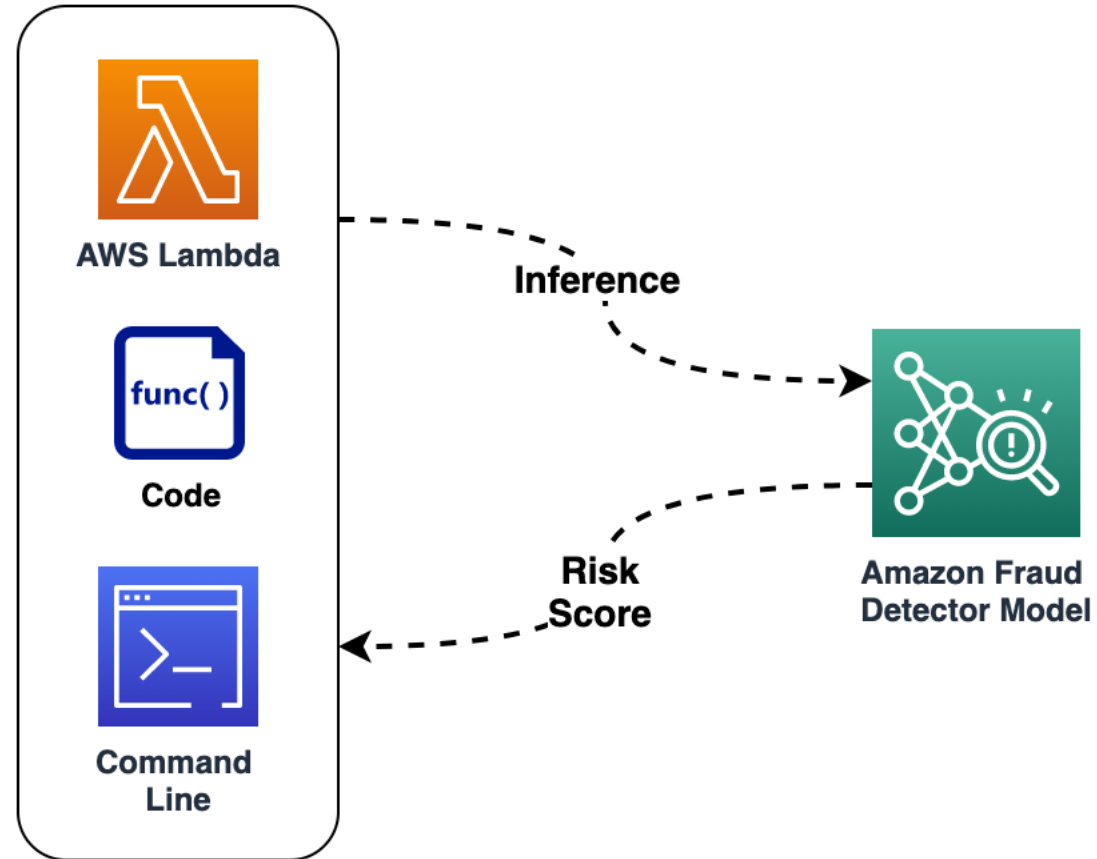
< 1 >

Rule name	Version	Description	Expression	Outcomes
high_fraud_risk	2	This rule captures events with a high ML model score	\$sample_fraud_detection_model_insightscore > 900	review_customer
low_fraud_risk	1	Low risk of fraud	\$sample_fraud_detection_model_insightscore <= 700	approve
medium_fraud_risk	1	Medium Fraud Risk	\$sample_fraud_detection_model_insightscore <= 900 and \$sample_fraud_detection_model_insightscore > 700	review

How Amazon Fraud Detector works



Generating predictions with a deployed model



Generating predictions with a deployed model

AWS SDK

Python, NodeJS, Java, C++, .Net etc.

GetEventPrediction API

Python example

```
response = client.get_event_prediction(
    detectorId='sample_detector',
    detectorVersionId='1',
    eventId='20220207',
    eventTypeName='user_registration',
    entities=[
        {
            'entityType': '20220207',
            'entityId': 'customer_entity'
        },
    ],
    eventTimestamp='20220207120903',
    eventVariables={
        'email_address': 'user@email.com',
        'ip_address': '192.192.192.192'
    }
)
```

AWS CLI

```
aws frauddetector get-event-prediction --detector-id sample_detector \
--detector-version-id 1 \
--event-id 20220207 \
--event-type-name user_registration \
--entities entityType=20220207,entityId=customer_entity \
--event-timestamp 20220207120903 \
--event-variables { \
    "email_address": "user@email.com", \
    "ip_address": "192.192.192.192" \
}
```


Sample Prediction

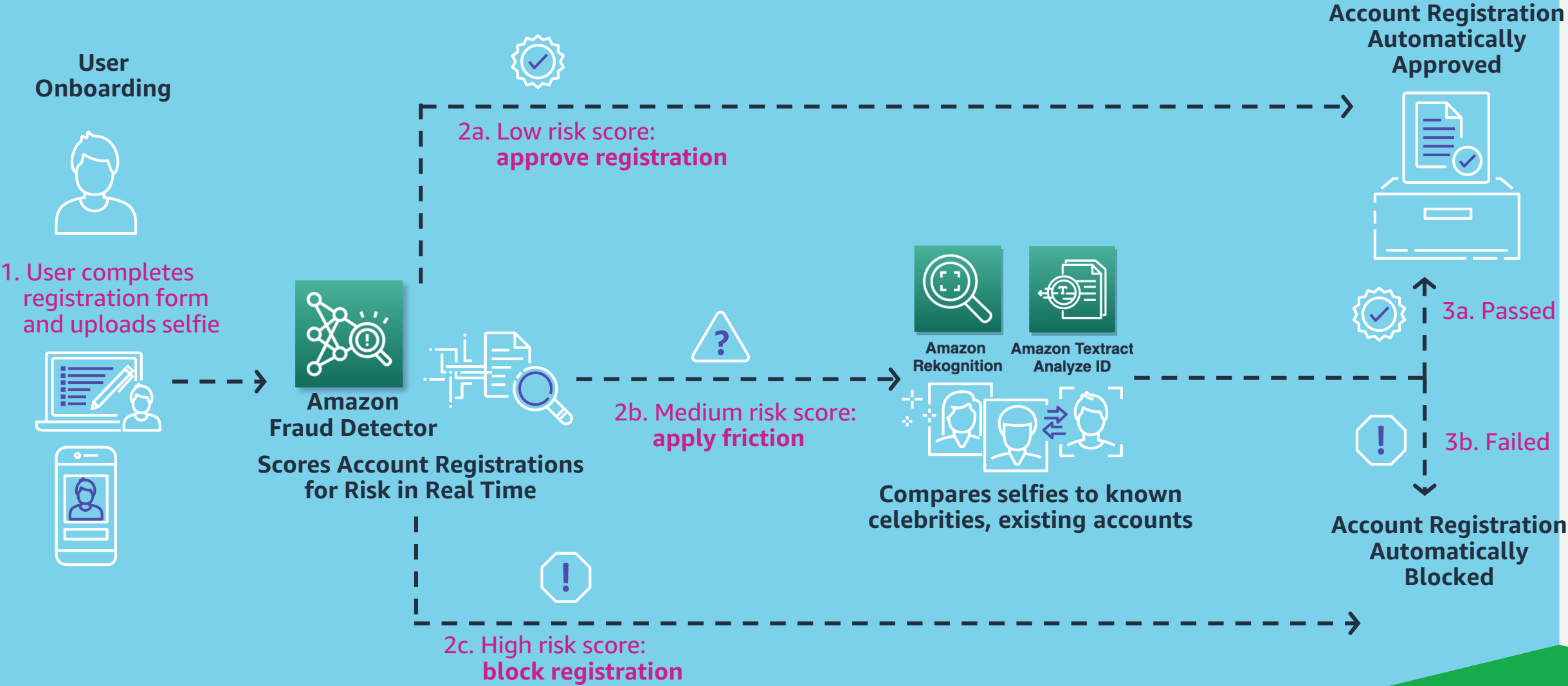
Model Score = 984

```
{
  "modelScores": [
    {
      "modelVersion": {
        "modelId": "sample_fraud_detection_model",
        "modelType": "ONLINE_FRAUD_INSIGHTS",
        "modelVersionNumber": "1.0"
      },
      "scores": {
        "sample_fraud_detection_model_insightscore": 984
      }
    }
  ],
  "ruleResults": [
    {
      "ruleId": "high_fraud_risk",
      "outcomes": [
        "review_customer"
      ]
    }
  ],
  "externalModelOutputs": []
}
```

High Fraud Risk

Fraud Detection

Combined with facial recognition & ID verification





Wuzzon

Wuzzon is an app marketing agency that is committed to aid app owners grow and activate their user acquisition by helping them set up a complete marketing plan, including user acquisition, app store optimization and re-engagement strategies.

“

By implementing Amazon Fraud Detector into the WuzzTrack system, Wuzzon now has a much more robust and reliable fraud detection solution, which can also detect more novel fraud techniques. The implementation was quick and easy and the results were even better than initially hoped. In some extreme cases there was a **decrease in false positives of up to 43%** (when compared to the previous rule-based solutions), while for other sources, **the true positive rate increased by 11-14%**.

”

Justin Westerveld

CTO

What Comes Next?

Try It Yourself

Blog:

Prevent fake account sign-ups in real time with AI using Amazon Fraud Detector

<https://a.co/4cgDed2>



Talk With Us

Deep Dive:

Contact your AWS Account Team to get help with your specific situation

Mention this session!

Learn More

For documentation, code samples and pricing visit:

aws.amazon.com/fraud-detector





Thank you!

Anjan Biswas

[linkedin.com/in/anjanavabiswas](https://www.linkedin.com/in/anjanavabiswas)