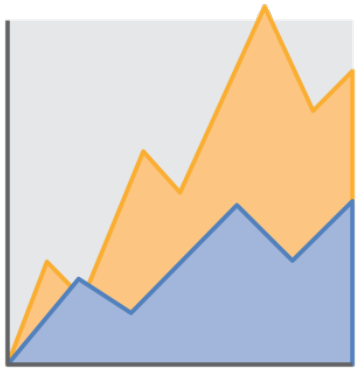
Machine Learning & Data Lake for IoT scenarios on AWS

John Chang
Technology Evangelist
October 2016

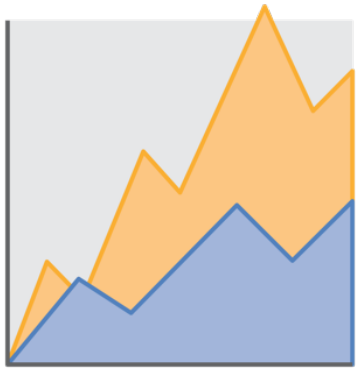
Three types of data-driven development



Retrospective
analysis and
reporting

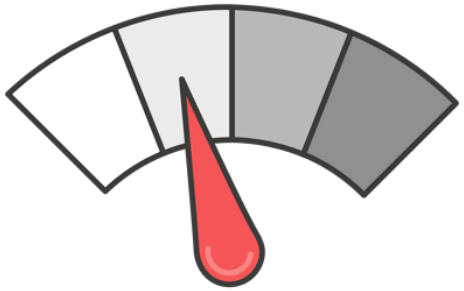
Amazon Redshift,
Amazon RDS
Amazon S3
Amazon EMR

Three types of data-driven development



Retrospective
analysis and
reporting

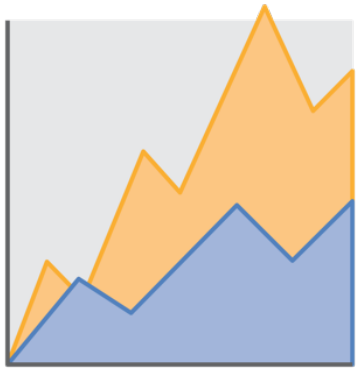
Amazon Redshift,
Amazon RDS
Amazon S3
Amazon EMR



Here-and-now
real-time processing
and dashboards

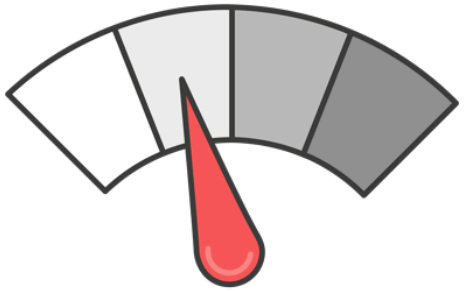
Amazon Kinesis
Amazon EC2
AWS Lambda

Three types of data-driven development



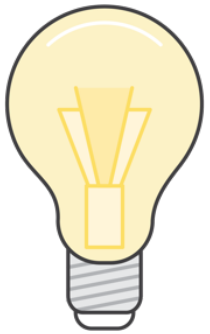
Retrospective
analysis and
reporting

Amazon Redshift,
Amazon RDS
Amazon S3
Amazon EMR



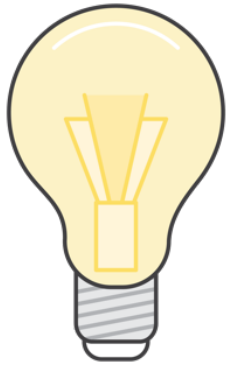
Here-and-now
real-time processing
and dashboards

Amazon Kinesis
Amazon EC2
AWS Lambda



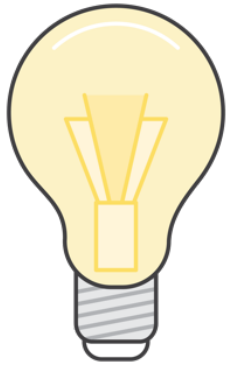
Predictions
to enable smart
applications

Machine learning and smart applications



Machine learning is the technology that automatically finds patterns in your data and uses them to make predictions for new data points as they become available.

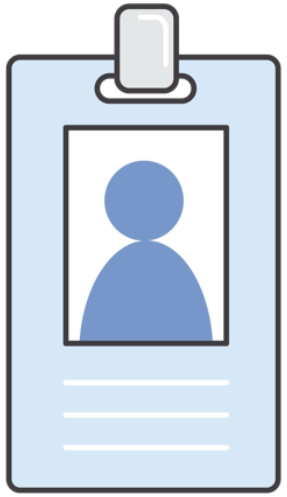
Machine learning and smart applications



Machine learning is the technology that automatically finds patterns in your data and uses them to make predictions for new data points as they become available.

Your data + machine learning = smart applications

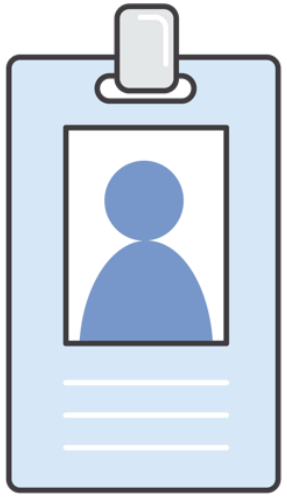
Smart applications by example



Based on what you know about the **user**:

Will they use your product?

Smart applications by example



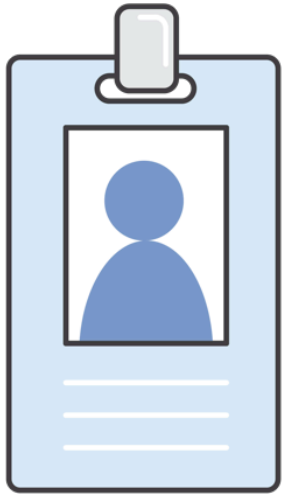
Based on what you know about the **user**:

Will they use your product?

Based on what you know about an **order**:

Is this order fraudulent?

Smart applications by example

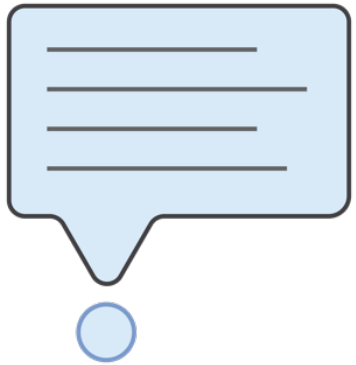


Based on what you know about the **user**:

Will they use your product?

Based on what you know know about an **order**:

Is this order fraudulent?



Based on what you know about a **news article**:

What other articles are interesting?

And a few more examples…

Fraud detection	Detecting fraudulent transactions, filtering spam emails, flagging suspicious reviews,…
Personalization	Recommending content, predictive content loading, improving user experience,…
Targeted marketing	Matching customers and offers, choosing marketing campaigns, cross-selling and up-selling,…
Content classification	Categorizing documents, matching hiring managers and resumes,…
Churn prediction	Finding customers who are likely to stop using the service, upgrade targeting,…
Customer support	Predictive routing of customer emails, social media listening,…

Smart applications by *counterexample*

```
Dear Alex,

This awesome quadcopter is on sale
for just $49.99!
```

Smart applications by *counterexample*

```
SELECT  c.ID
FROM    customers c
        LEFT JOIN orders o
            ON c.ID = o.customer
GROUP   BY c.ID
HAVING  o.date > GETDATE() – 30
```

We can start by sending the offer to all customers who placed an order in the last 30 days

Smart applications by *counterexample*

```
SELECT   c.ID
FROM     customers c
         LEFT JOIN orders o
             ON c.ID = o.customer
GROUP    BY c.ID
HAVING   O.CATEGORY = 'TOYS'
   AND   o.date > GETDATE() – 30
```

…let's narrow it down to just customers who bought toys

Smart applications by *counterexample*

```
SELECT  c.ID
FROM    customers c
        LEFT JOIN orders o
            ON c.ID = o.customer
        LEFT JOIN PRODUCTS P
            ON P.ID = O.PRODUCT
GROUP   BY c.ID
HAVING  o.category = 'toys'
   AND ((P.DESCRIPTION LIKE '%HELICOPTER%'
         AND O.DATE > GETDATE() - 60)
     OR (COUNT(*) > 2
         AND  SUM(o.price) > 200
         AND  o.date > GETDATE() – 30)
       )
```

…and expand the query to customers who purchased other toy helicopters recently, or made several expensive toy purchases

Smart applications by *counterexample*

```
SELECT  c.ID
FROM    customers c
        LEFT JOIN orders o
            ON c.ID = o.customer
        LEFT JOIN products p
            ON p.ID = o.product
GROUP   BY c.ID
HAVING  o.category = 'toys'
   AND ((p.description LIKE '%COPTER%'
         AND o.date > GETDATE() - 60)
     OR (COUNT(*) > 2
         AND  SUM(o.price) > 200
         AND  o.date > GETDATE() – 30)
       )
```

…but what about quadcopters?

Smart applications by *counterexample*

```
SELECT  c.ID
FROM    customers c
        LEFT JOIN orders o
            ON c.ID = o.customer
        LEFT JOIN products p
            ON p.ID = o.product
GROUP   BY c.ID
HAVING  o.category = 'toys'
   AND ((p.description LIKE '%copter%'
         AND o.date > GETDATE() - 120)
     OR (COUNT(*) > 2
         AND  SUM(o.price) > 200
         AND  o.date > GETDATE() – 30)
       )
```

…maybe we should go back further in time

Smart applications by *counterexample*

```
SELECT  c.ID
FROM    customers c
        LEFT JOIN orders o
            ON c.ID = o.customer
        LEFT JOIN products p
            ON p.ID = o.product
GROUP   BY c.ID
HAVING  o.category = 'toys'
   AND ((p.description LIKE '%copter%'
         AND o.date > GETDATE() - 120)
     OR (COUNT(*) > 2
         AND  SUM(o.price) > 200
         AND  o.date > GETDATE() – 40)
       )
```

…tweak the query more

Smart applications by *counterexample*

```
SELECT  c.ID
FROM    customers c
        LEFT JOIN orders o
            ON c.ID = o.customer
        LEFT JOIN products p
            ON p.ID = o.product
GROUP   BY c.ID
HAVING  o.category = 'toys'
   AND ((p.description LIKE '%copter%'
         AND o.date > GETDATE() - 120)
     OR (COUNT(*) > 2
         AND  SUM(o.price) > 150
         AND  o.date > GETDATE() – 40)
       )
```

…again

Smart applications by *counterexample*

```
SELECT  c.ID
FROM    customers c
        LEFT JOIN orders o
            ON c.ID = o.customer
        LEFT JOIN products p
            ON p.ID = o.product
GROUP   BY c.ID
HAVING  o.category = 'toys'
   AND ((p.description LIKE '%copter%'
         AND o.date > GETDATE() - 90)
     OR (COUNT(*) > 2
         AND  SUM(o.price) > 150
         AND  o.date > GETDATE() – 40)
       )
```

…and again

Smart applications by *counterexample*

```
SELECT  c
FROM      comers c
          LEFT JOIN orders
               ON c.ID =      comer
          LEFT JOIN p      s p
               ON p      o.product
GRO       BY c.I
HAVI      o.c      y = 'toys'
    AN        cription LIKE '%      er%'
              o.date > GETDA      - 90)
       OR (C
           AND  Sum(      ice) > 150
           AND  o.date > GETDATE() - 40)
         )
```

Use machine learning technology to **learn** your business rules from data!

Why aren't there *more* smart applications?

1. Machine learning expertise is **rare**.

2. Building and scaling machine learning technology is **hard**.

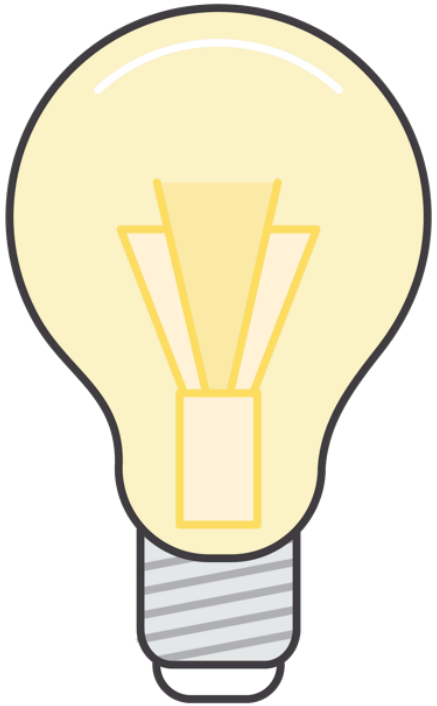
3. Closing the gap between models and applications is **time-consuming and expensive**.

Building smart applications today

Expertise	**Technology**	**Operationalization**
Limited supply of data scientists	Many choices, few mainstays	Complex and error-prone data workflows
Expensive to hire or outsource	Difficult to use and scale	Custom platforms and APIs
	Many moving pieces lead to custom solutions *every time*	Reinventing the model lifecycle management wheel

What if there were a better way?

Introducing Amazon Machine Learning



Easy-to-use, managed machine learning service built for developers

Robust, powerful machine learning technology based on Amazon's internal systems

Create models using your data already stored in the AWS cloud

Deploy models to production in seconds

Easy-to-use and developer-friendly

Use the intuitive, powerful service console to build and explore your initial models

- Data retrieval
- Model training, quality evaluation, fine-tuning
- Deployment and management

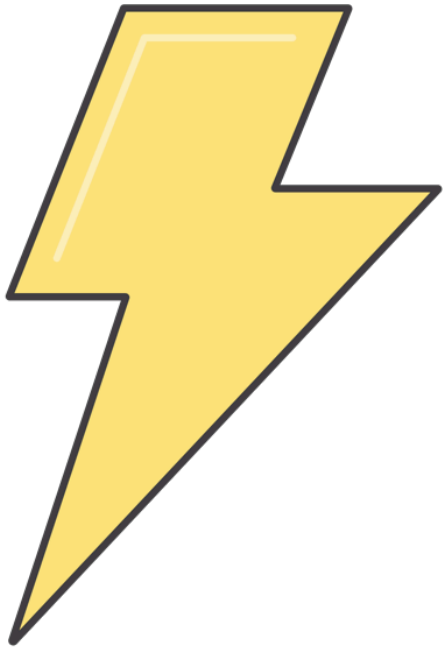
Automate model lifecycle with fully featured APIs and SDKs

- Java, Python, .NET, JavaScript, Ruby, PHP

Easily create smart iOS and Android applications with AWS Mobile SDK

Powerful machine learning technology

Based on Amazon's battle-hardened internal systems



Not just the algorithms:

- Smart data transformations
- Input data and model quality alerts
- Built-in industry best practices

Grows with your needs

- Train on up to 100 GB of data
- Generate billions of predictions
- Obtain predictions in batches or real-time

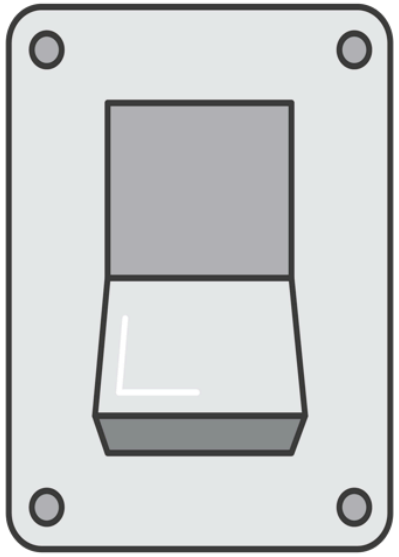
Integrated with the AWS data ecosystem

Access data that is stored in Amazon S3, Amazon Redshift, or MySQL databases in Amazon RDS

Output predictions to Amazon S3 for easy integration with your data flows

Use AWS Identity and Access Management (IAM) for fine-grained data access permission policies

Fully-managed model and prediction services



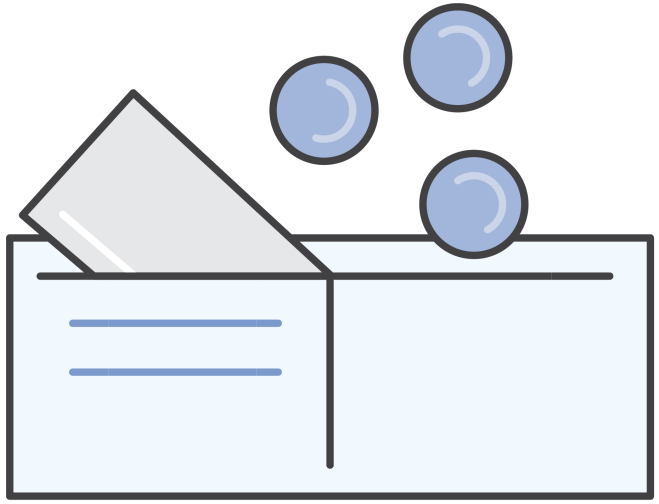
End-to-end service, with no servers to provision and manage

One-click production model deployment

Programmatically query model metadata to enable automatic retraining workflows

Monitor prediction usage patterns with Amazon CloudWatch metrics

Pay-as-you-go and inexpensive



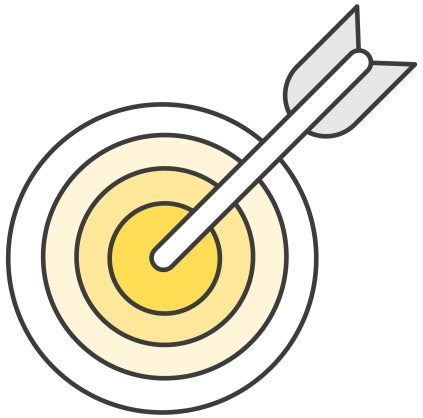
Data analysis, model training, and evaluation: **$0.42/instance hour**

Batch predictions: **$0.10/1000**

Real-time predictions: **$0.10/1000**

+ hourly capacity reservation charge

Three supported types of predictions



Binary classification

Predict the answer to a Yes/No question

Multiclass classification

Predict the correct category from a list

Regression

Predict the value of a numeric variable

Building smart applications with Amazon ML

1

Train model

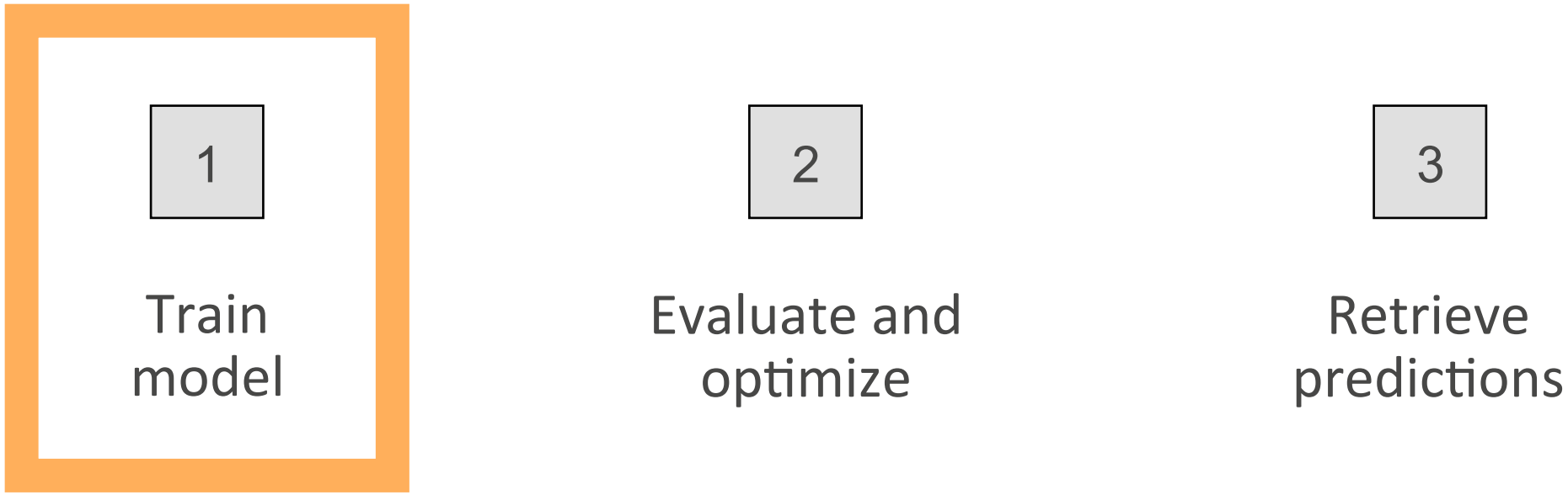
2

Evaluate and optimize

3

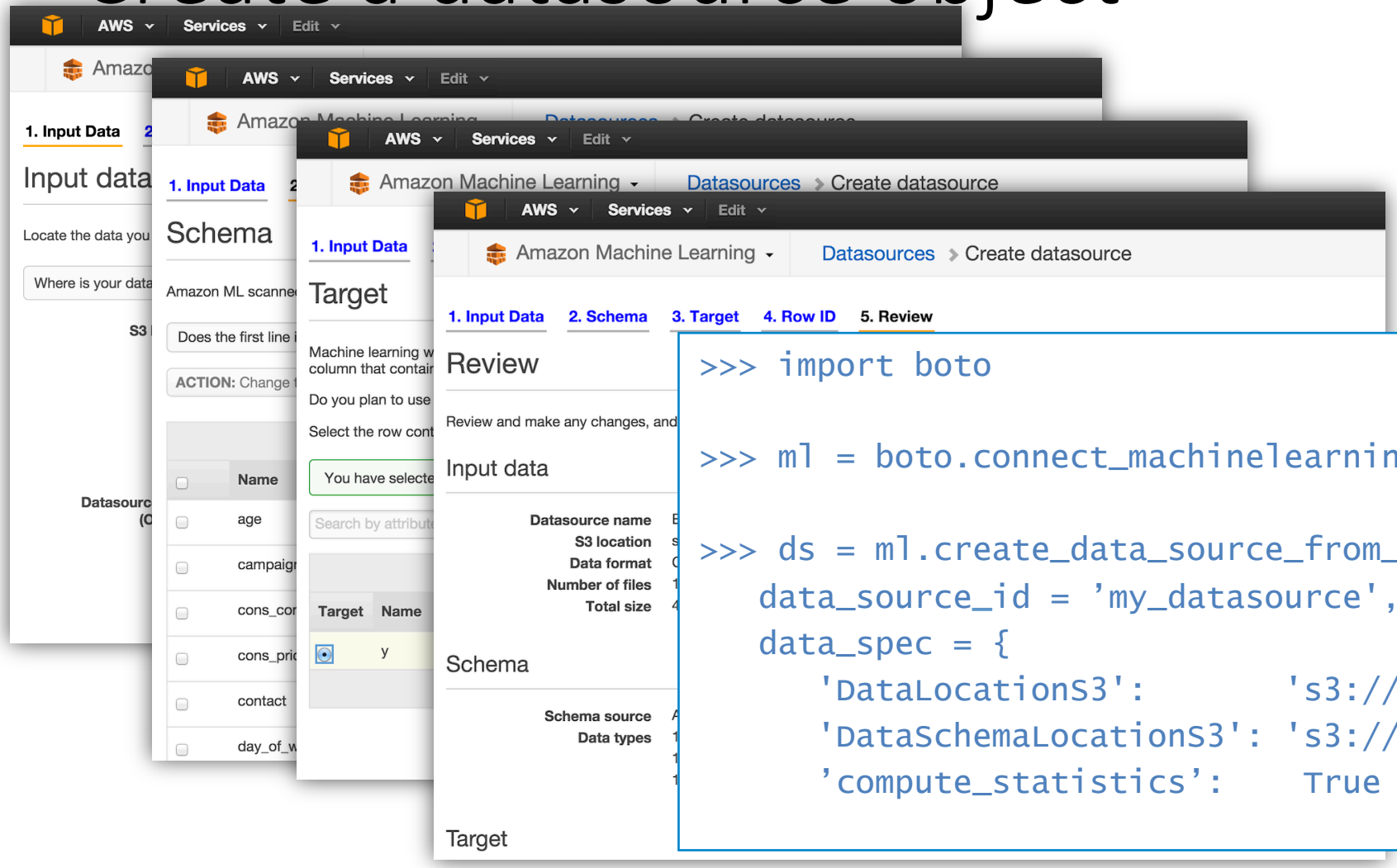
Retrieve predictions

Building smart applications with Amazon ML

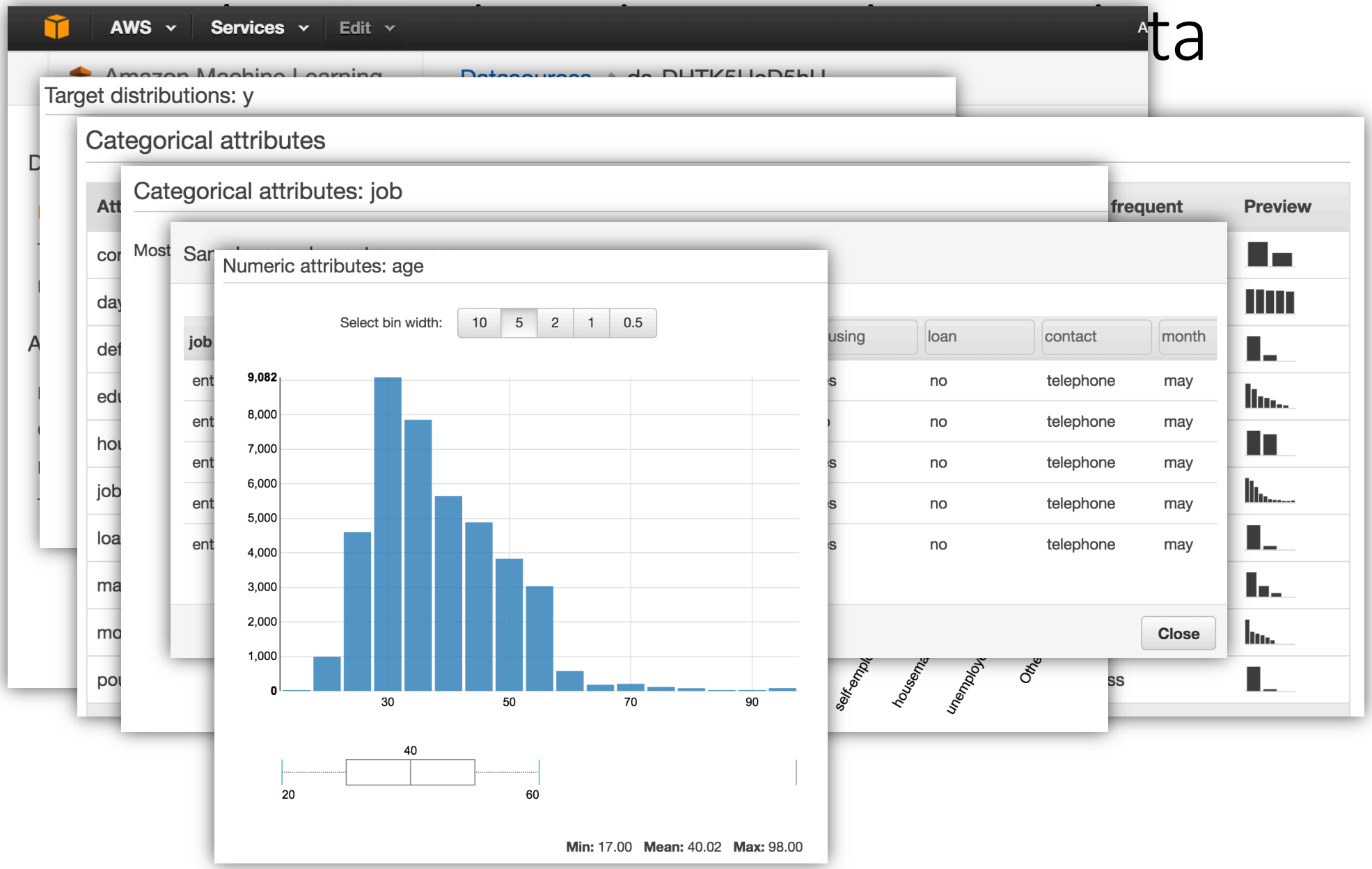


- Create a datasource object pointing to your data
- Explore and understand your data
- Transform data and train your model

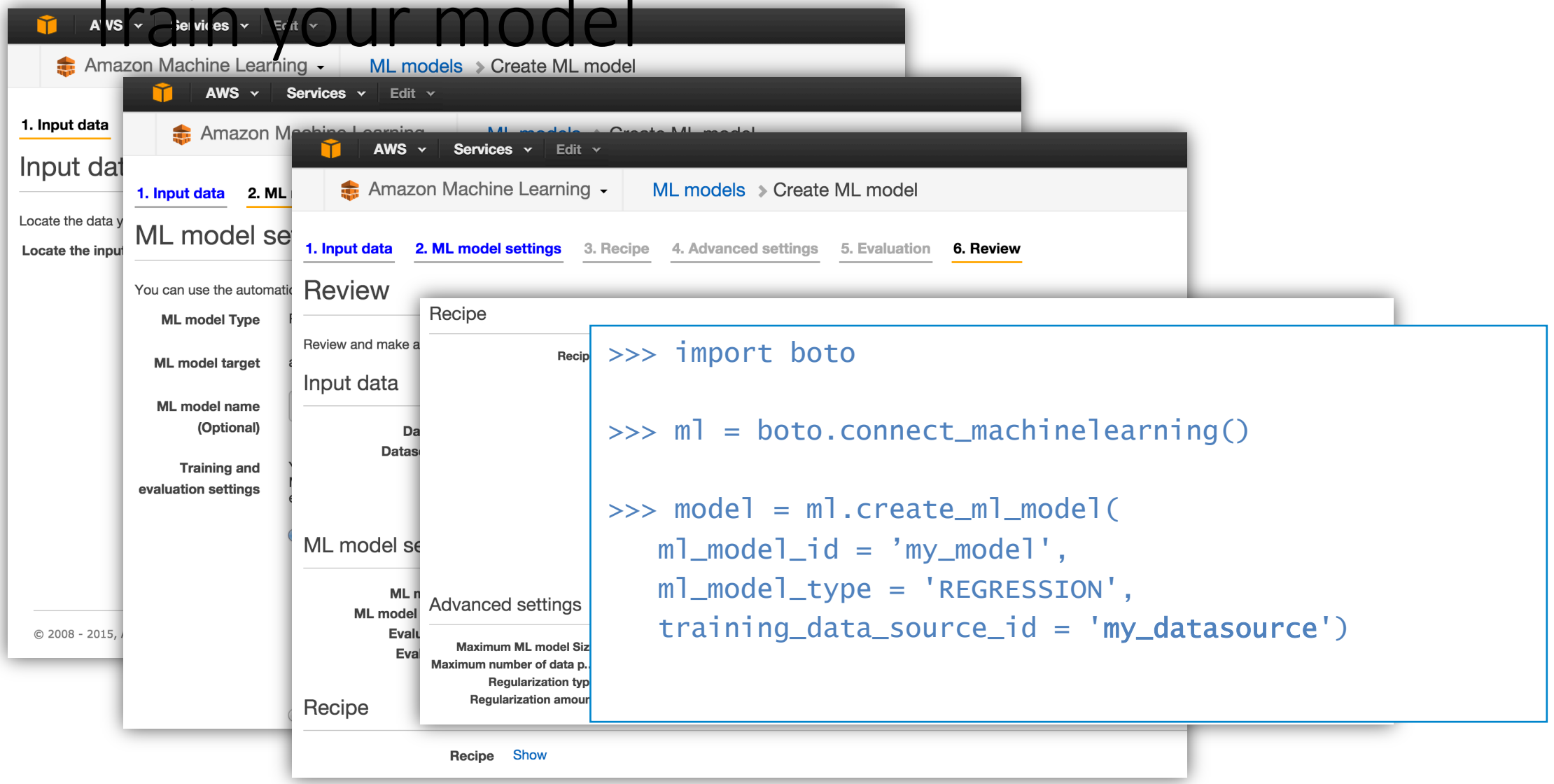
Create a datasource object



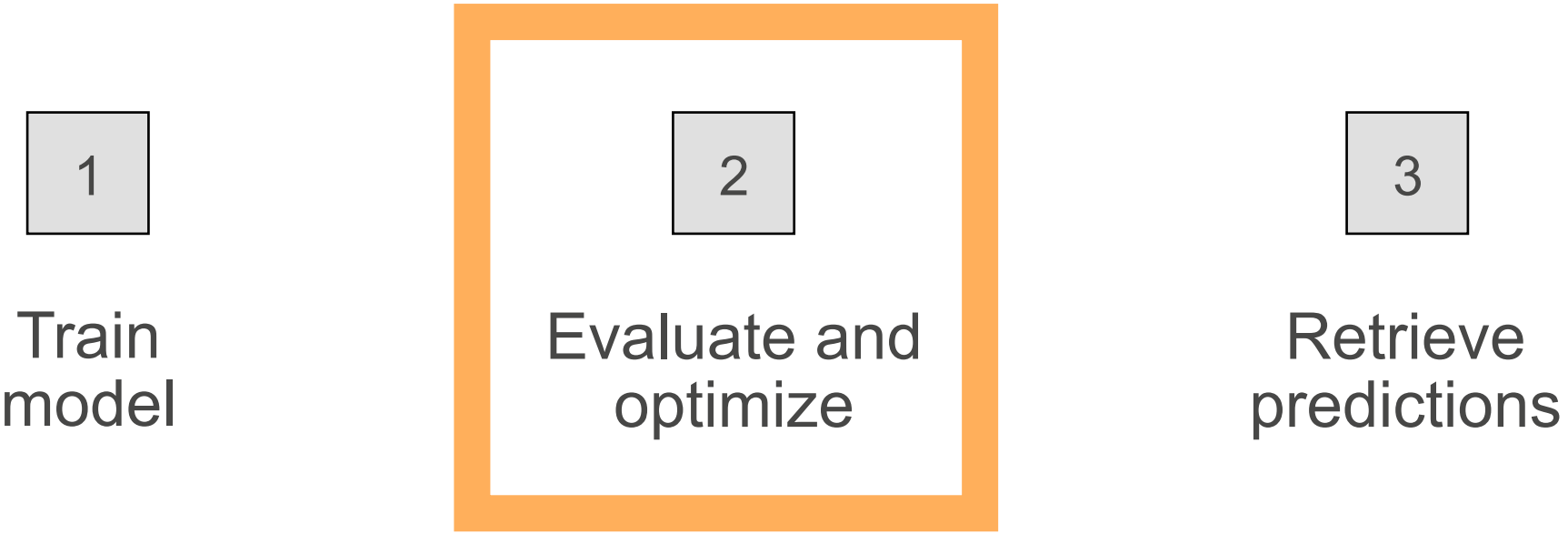

```
>>> import boto

>>> ml = boto.connect_machinelearning()

>>> ds = ml.create_data_source_from_s3(
   data_source_id = 'my_datasource',
   data_spec = {
      'DataLocationS3':       's3://bucket/input/data.csv',
      'DataSchemaLocationS3': 's3://bucket/input/data.schema',
      'compute_statistics':    True } )
```


AWS
Services
Edit
Target distributions: y
Categorical attributes
Categorical attributes: job
frequent
Preview
Numeric attributes: age
Select bin width:
10
5
2
1
0.5
9,082
8,000
7,000
6,000
5,000
4,000
3,000
2,000
1,000
0
30
50
70
90
40
20
60
Min: 17.00 Mean: 40.02 Max: 98.00
job
loan
contact
month
no
telephone
may
Close

Train your model

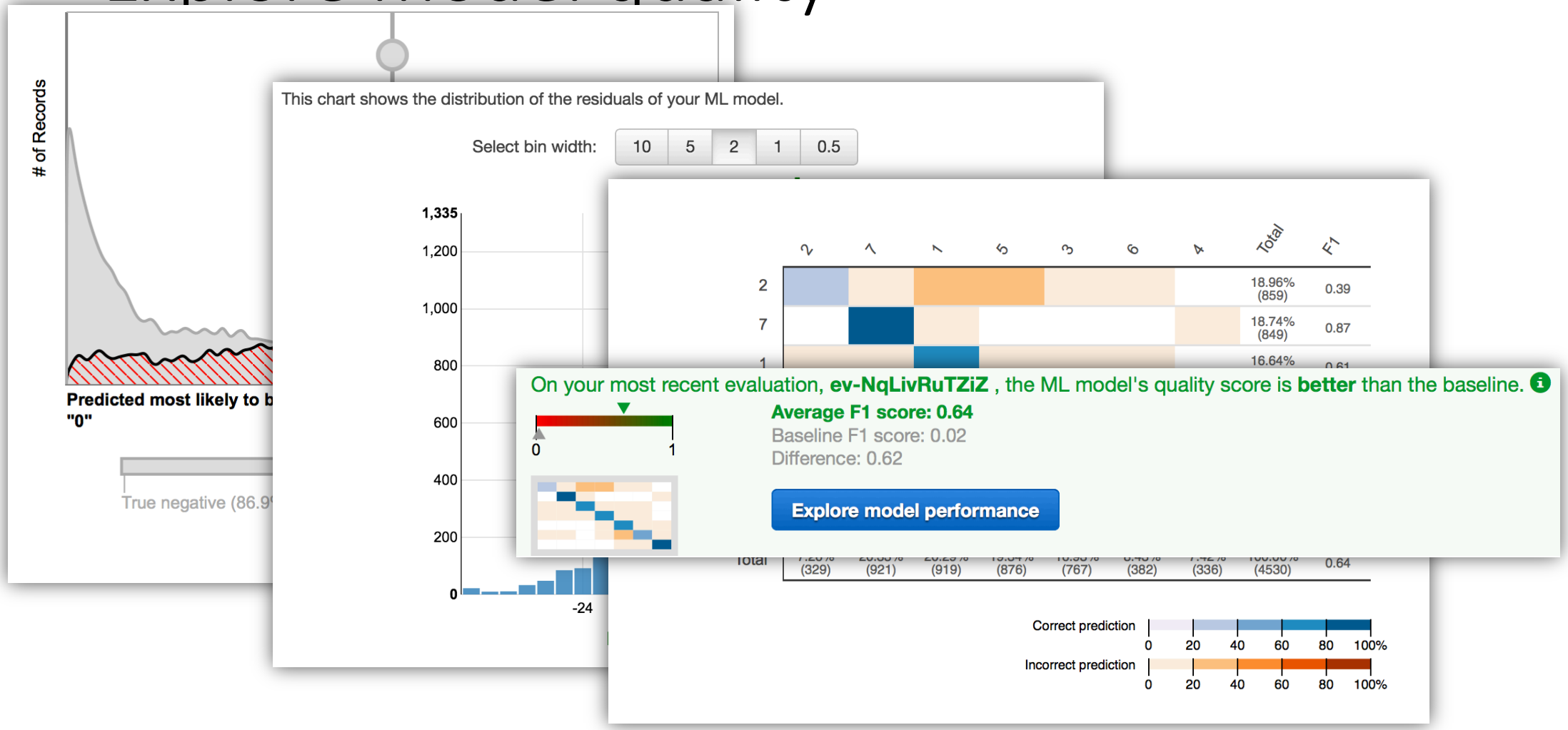


Building smart applications with Amazon ML

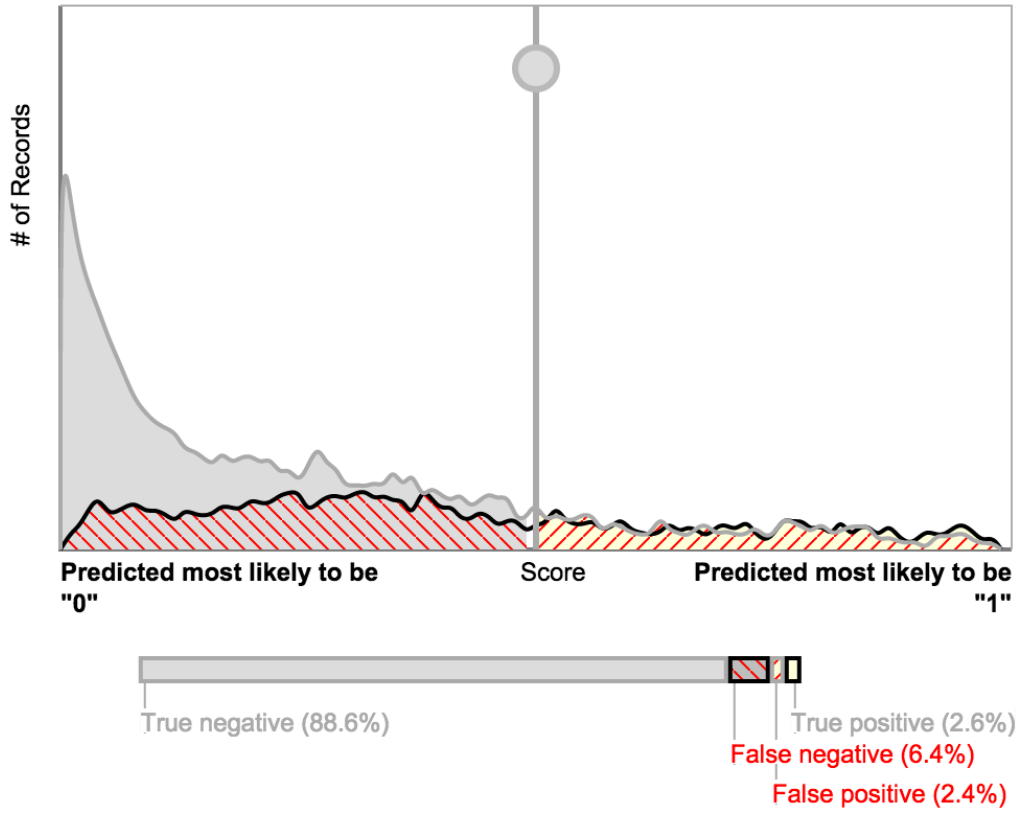


- Measure and understand model quality
- Adjust model interpretation

Explore model quality



Fine-tune model interpretation



Trade-off based on score threshold 0.5

Reset score threshold (0.5)

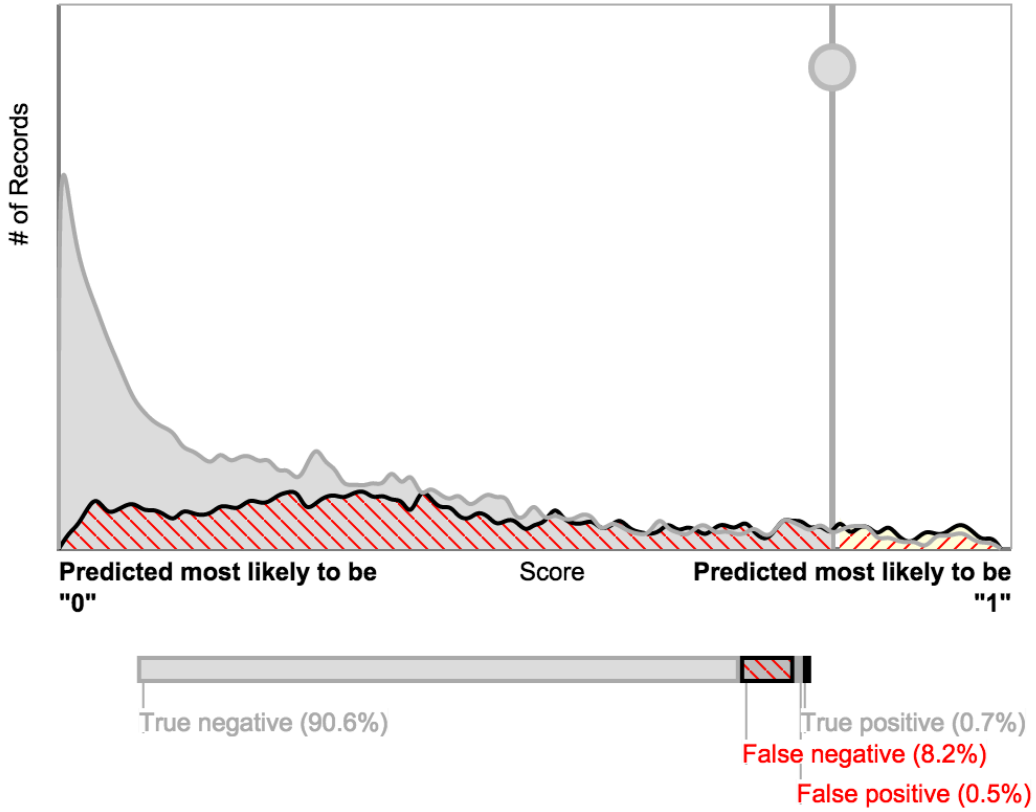
- **91% are correct**
  607 true positive
  20,933 true negative
- **9% are errors**
  567 false positive
  1,507 false negative

- 5% of the records are predicted as "1"
- 95% of the records are predicted as "0"

Save score threshold at 0.50

› Advance metrics

Fine-tune model interpretation



Trade-off based on score threshold 0.81

Reset score threshold (0.5)

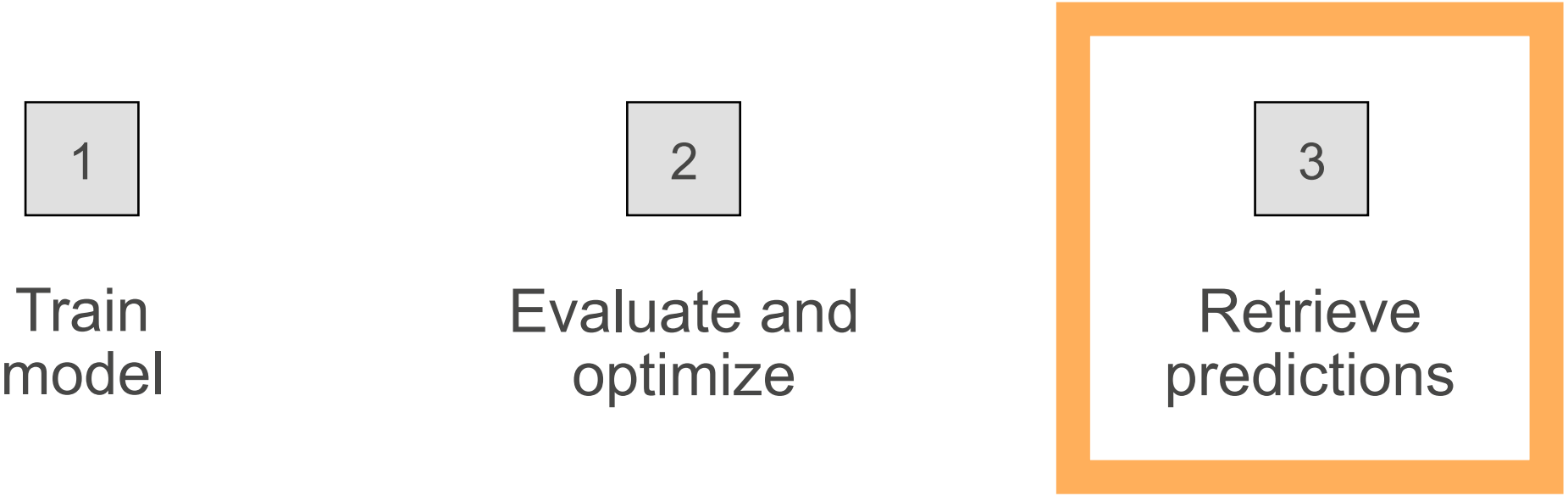
- **91% are correct**
  166 true positive
  21,385 true negative
- **9% are errors**
  115 false positive
  1,948 false negative

- 1% of the records are predicted as "1"
- 99% of the records are predicted as "0"

Save score threshold at 0.81

› Advance metrics

Building smart applications with Amazon ML



- Batch predictions
- Real-time predictions

Batch predictions

Asynchronous, large-volume prediction generation

Request through service console or API

Best for applications that deal with batches of data records

```
>>> import boto

>>> ml = boto.connect_machinelearning()

>>> model = ml.create_batch_prediction(
    batch_prediction_id = 'my_batch_prediction',
    batch_prediction_data_source_id = 'my_datasource',
    ml_model_id = 'my_model',
    output_uri = 's3://examplebucket/output/')
```

```
bestAnswer,score
0,3.93914E-1
0,1.654963E-2
1,0.832306
0,2.143189E-2
0,9.23001E-3
0,0.714461
0,9.772378E-3
1,0.525307
1,0.710729
```

Real-time predictions

Synchronous, low-latency, high-throughput prediction generation

Request through service API, server, or mobile SDKs

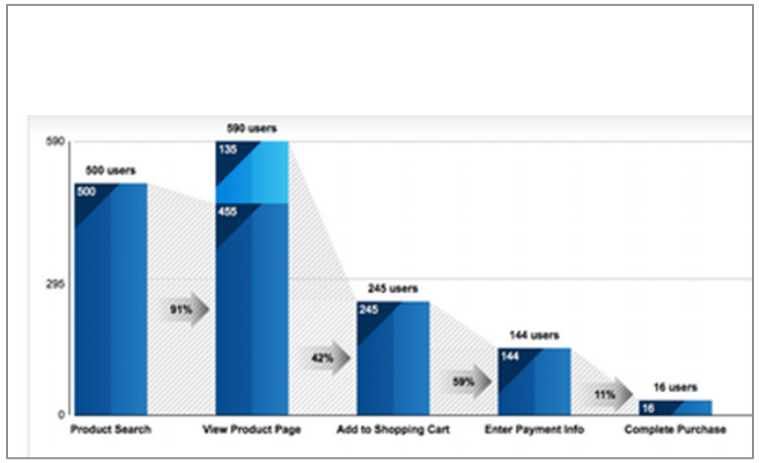
Best for interaction applications that deal with individual data records

```
>>> import boto

>>> ml = boto.connect_machinelearning()

>>> ml.predict(
   ml_model_id = 'my_model',
   predict_endpoint = 'example_endpoint',
   record = {'key1':'value1', 'key2':'value2'})
```

```
{
  'Prediction': {
    'predictedValue': 13.284348,
    'details': {
      'Algorithm': 'SGD',
      'PredictiveModelType': 'REGRESSION'
    }
  }
}
```

Data Lake

Retailers need to deliver continuous differentiation

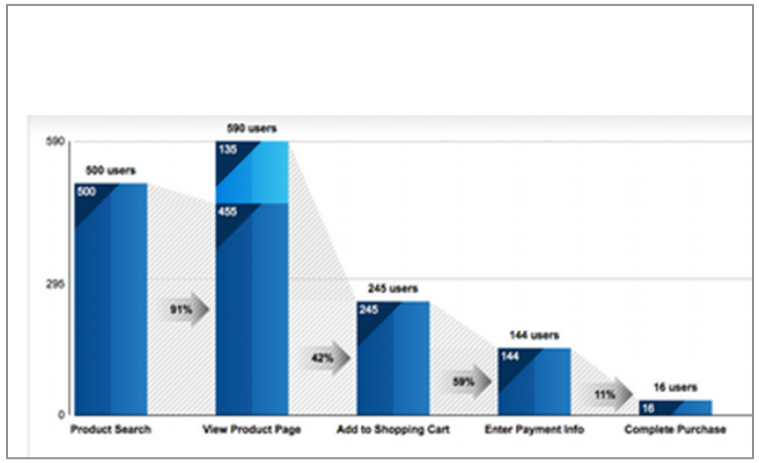


Real-time engagement

Personalization

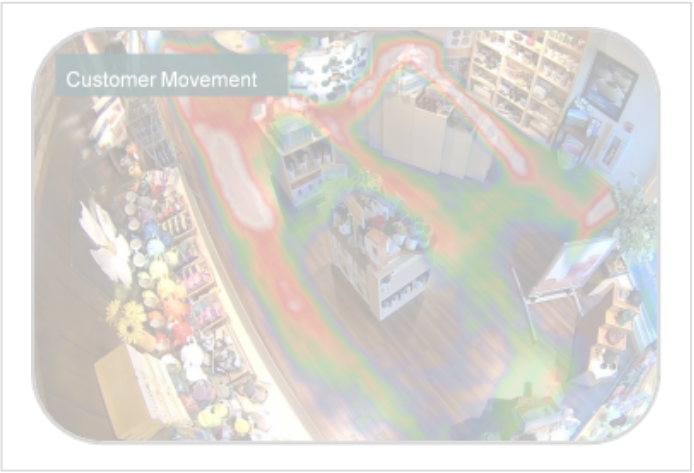
Merchandising

Retailers need to deliver continuous differentiation



Real-time engagement

Personalization



Merchandising

REDFIN®

A full-service residential real estate brokerage

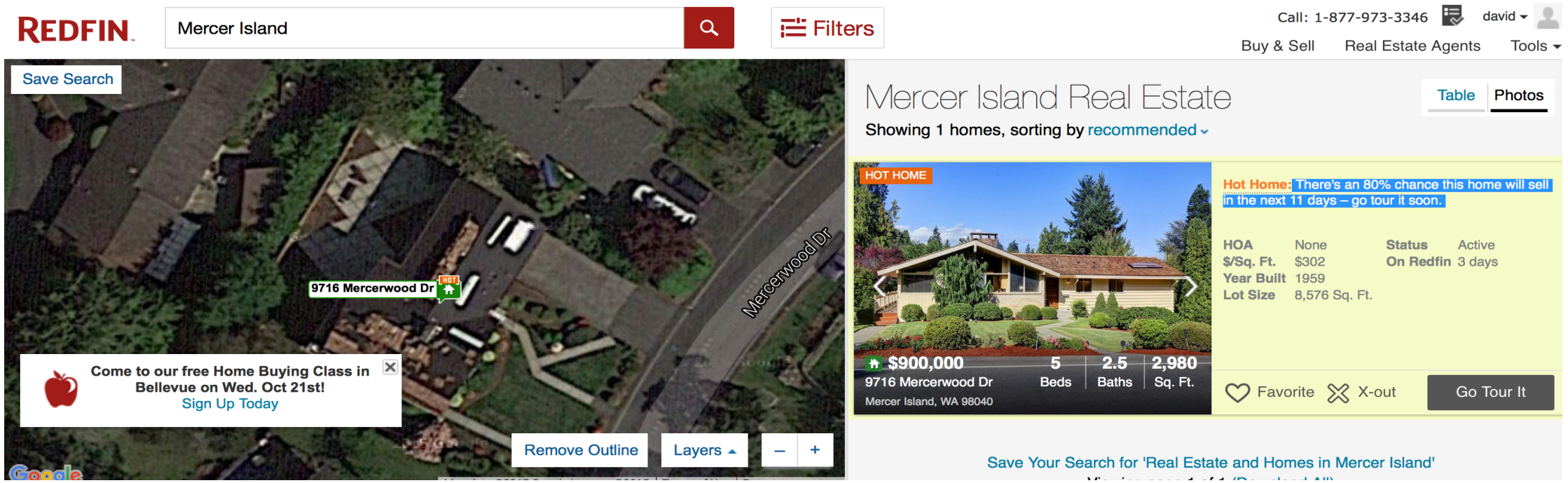
Redfin manages data on **hundreds of millions of properties** and millions of customers

The Hot Homes algorithm automatically calculates the likelihood by analyzing more than 500 attributes of each home

Was fully AWS-native **since day one**

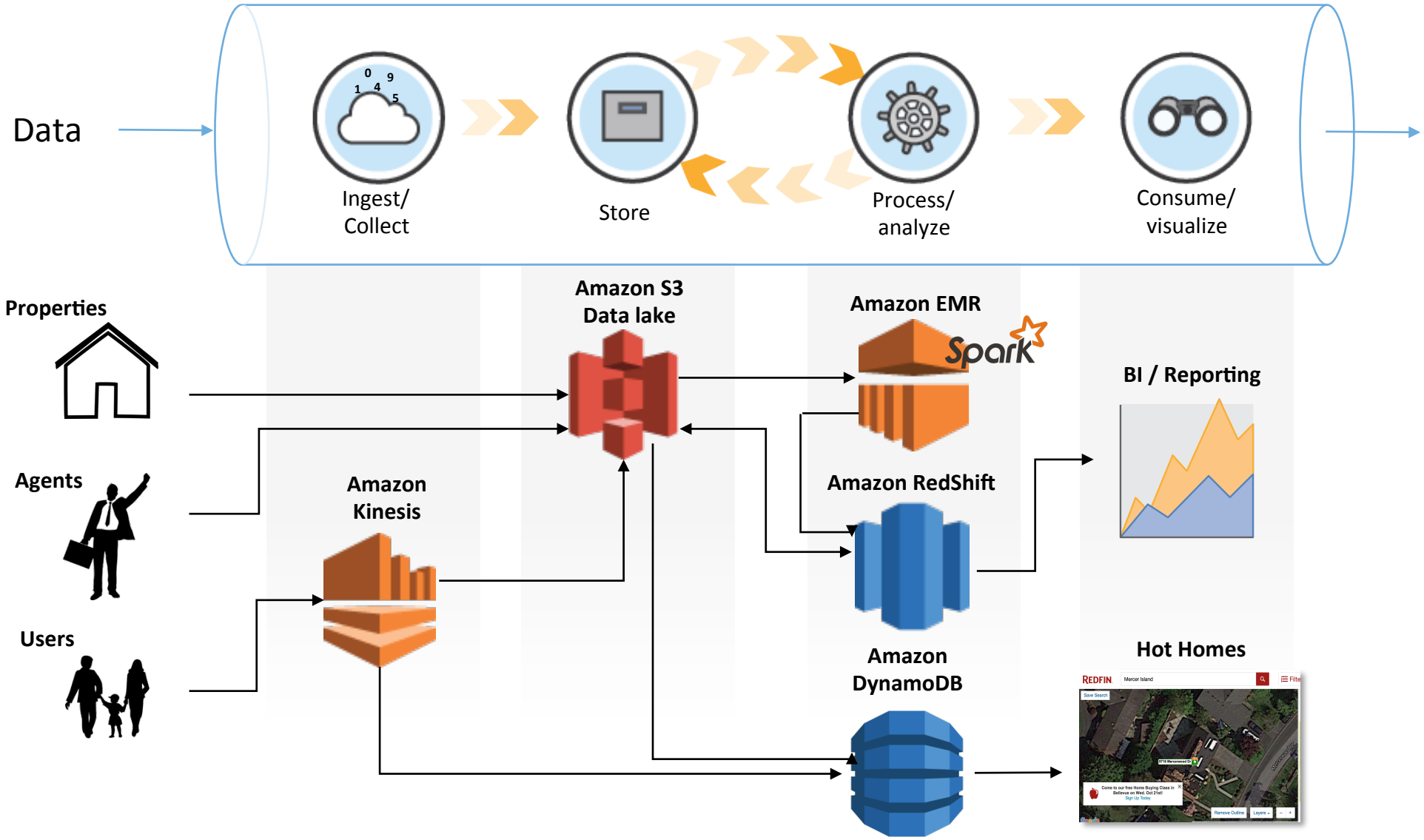
https://aws.amazon.com/solutions/case-studies/redfin/

REDFIN Hot Homes



There's an 80% chance this home will sell in the next 11 days – go tour it soon.

REDFIN

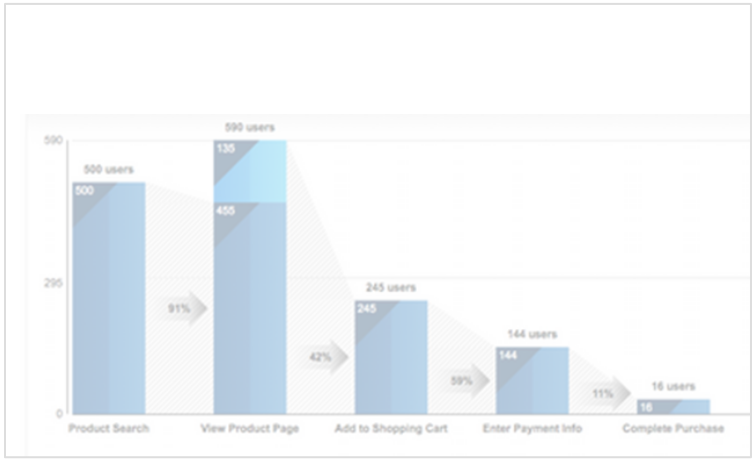

Data
Ingest/
Collect
Store
Process/
analyze
Consume/
visualize
Answers &
Insights
Properties
Agents
Users
Amazon S3
Data lake
Amazon EMR
Spark
Amazon
Kinesis
Amazon RedShift
BI / Reporting
Amazon
DynamoDB
Hot Homes
User Profile
Recommendation
Hot Homes
Similar Homes
Agent Follow-up
Agent Scorecard
Marketing
A/B Testing
Real Time Data
…

Redfin Manages Data on Hundreds of Millions of Properties Using AWS

> Once we solved the infrastructure problem, we could dream a little bigger. Now we can deliver results without worrying about how to scale.
>
> Yong Huang, Director, Big Data and Analytics
>
> REDFIN

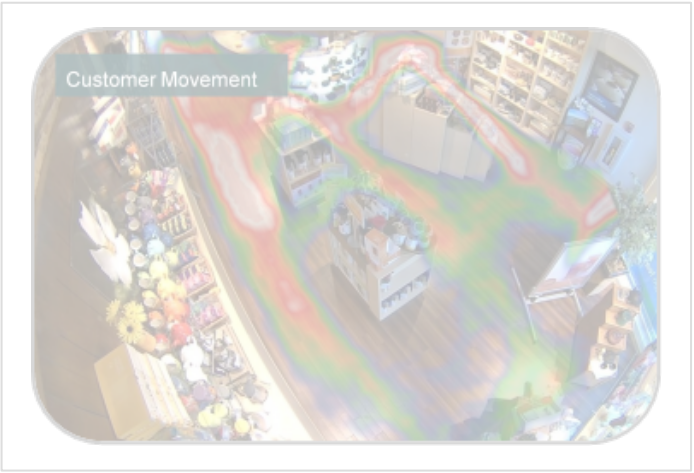
- Zero on-premises infrastructure
- Using spot pricing for EC2, Redfin saved 90% compared to running on-demand
- Using AWS, Redfin maintains a small technical team, allowing much simplified server management and allowing the transition to DevOps
- Redfin is able to launch products like Hot Homes to greatly increase the buyer experience, by leveraging the agility and scale of AWS

Retailers need to deliver continuous differentiation



Real-time engagement

Personalization



Merchandising

NORDSTROM

American upscale fashion retailer

Nordstrom has **323 stores** operating in 38 of the United States and also in Canada; the **largest in number of stores and geographic footprint** of its retail competitors

Fashion retailer that sells clothing, shoes, cosmetics, and accessories

Nordstrom is **going all in** on AWS

https://aws.amazon.com/solutions/case-studies/nordstrom/

NORDSTROM

Search by keyword or item #

Change Selections

Designer Collections | Women | Men | Shoes | Handbags | Accessories | Beauty | Trend | Kids | Home | Gifts | Sale | Brands | POP-IN

Home / Get Recommendations

Recommendations for You

Need a little shopping inspiration? Take a look at recommendations for favorite brands, bestsellers, new items and more.

Bestsellers

Madewell 'Whisper' Cotton V-Neck Tee

KRW 24,149.58

★★★★½ (80)

Longchamp 'Large Le Pliage' Tote

KRW 179,573.80

★★★★½ (883)

Converse Chuck Taylor® 'Shoreline' Sneaker (Women)

KRW 61,860.08

★★★★½ (376)

Vince Camuto 'Lavette' Perforated Peep Toe Bootie (Women)

KRW 148,550.88

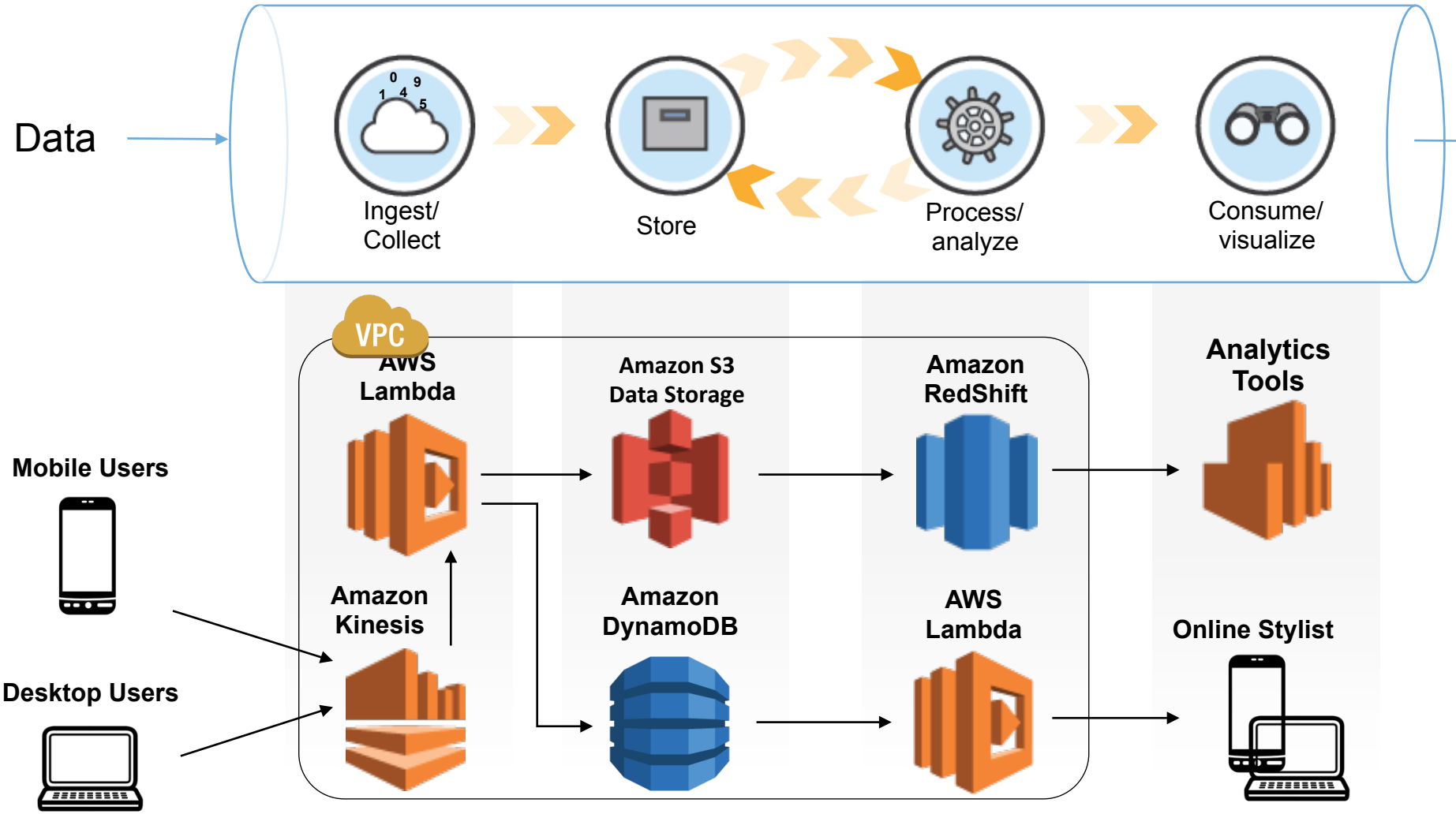
★★★½☆ (13)

Eliza J Embellished Tiered Chiffon Halter Gown

KRW 245,211.12

★★★★½ (152)

NORDSTROM



Personalized recommendations within seconds (from 15-20 min)

Scale the expertise of stylists to all shoppers

Reduce costs by 2X order of magnitude

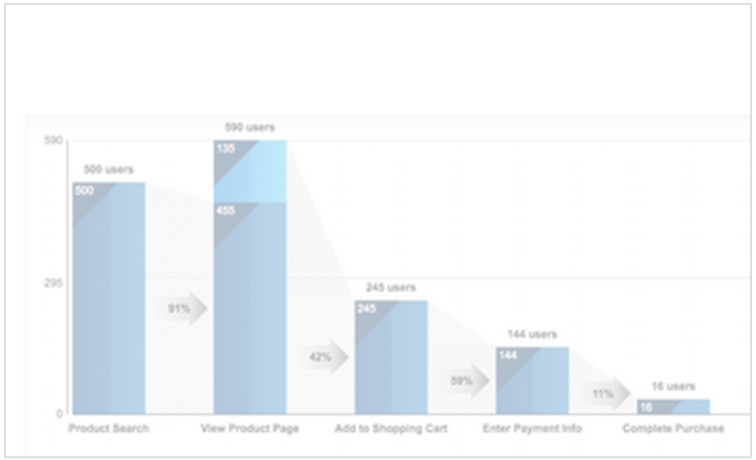
…

Nordstrom gives personalized style recommendations in seconds

> Alert me when the internet is down ...
>
> Keith Homewood
> Cloud Product Owner, Nordstrom
>
> **Nordstrom**

- Nordstrom Recommendation is the online version of a stylist. It can analyze and deliver personalized recommendations in seconds
- Going All-In on AWS has resulted in reducing costs by 2X
- Continuous delivery allows Nordstrom to deliver multiple production launches a day in a single application
- Can now create a personalized recommendation in seconds, in what used to take 15-20 minutes of processing
- Nordstrom Cloud Product Owner finds the reliability and availability of AWS so suitable that as long as the internet is working, Nordstrom Recommendation is working

Retailers need to deliver continuous differentiation



Real-time engagement

Personalization

Merchandising

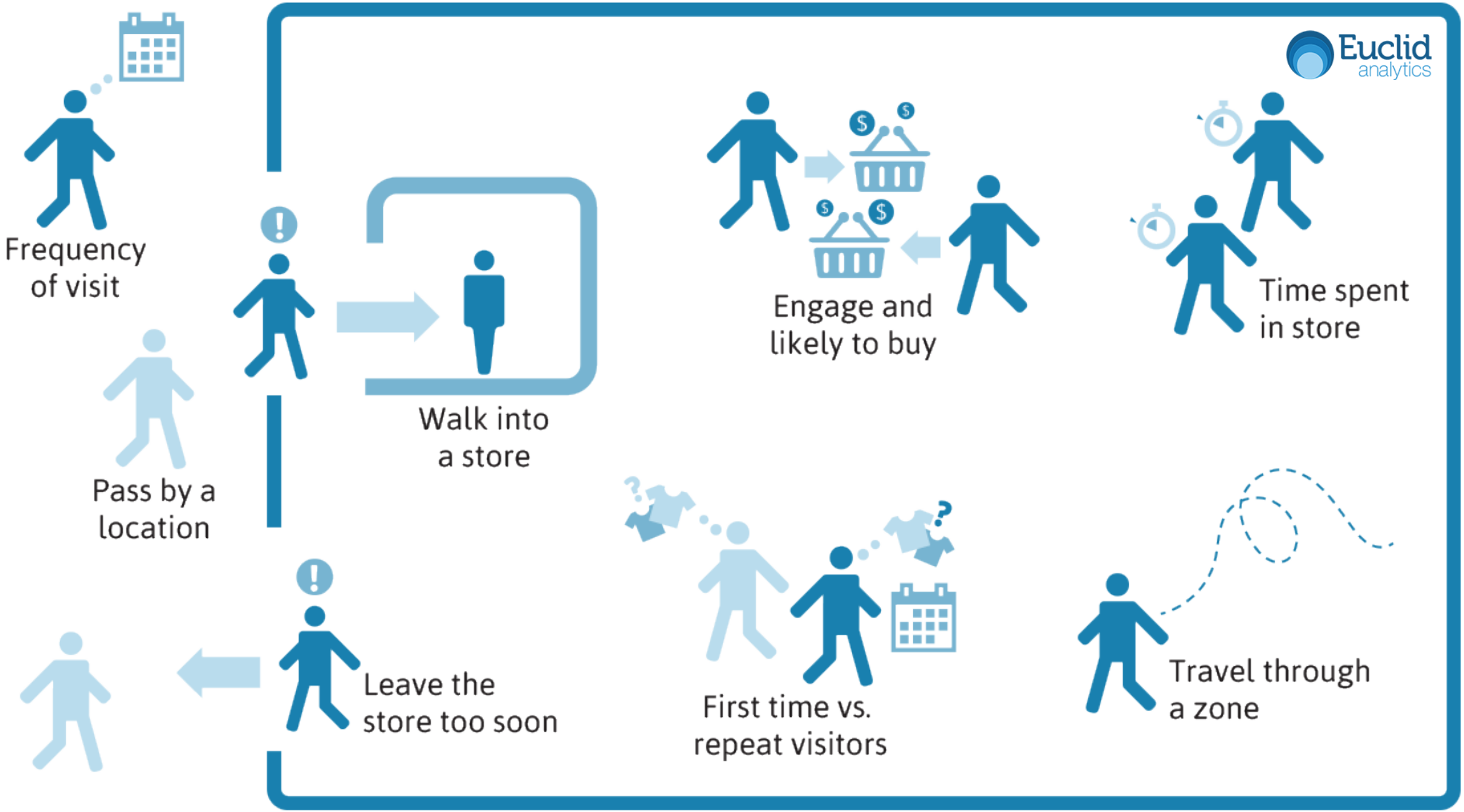
Technology that helps brick-and-mortar retailers optimize performance

Trusted by over
500 global brands in
45 countries worldwide
and counting

Euclid analyzes customer movement data to correlate traffic with marketing campaigns and to help retailers optimize hours for peak traffic

Was fully AWS-native
since day one

https://aws.amazon.com/solutions/case-studies/euclid/


Euclid analytics
Frequency of visit
Walk into a store
Engage and likely to buy
Time spent in store
Pass by a location
Leave the store too soon
First time vs. repeat visitors
Travel through a zone

Euclid analytics

Data

Ingest/ Collect

Store

Process/ analyze

Consume/ visualize

Answers & Insights

- Walk-Bys
- New & Return Visitors
- Visit Duration
- Engagement Rate
- Bounce Rate
- Storefront Potential & Conversion
- Customer segmentation and loyalty assessment
- Regional and categorical roll-up reporting
- Zoning for large-format locations

Campaigns

WiFi - Foot traffic

Transactions

Elastic Load Balancing

Amazon EC2

Amazon Elastic Beanstalk

Amazon S3 Data lake

Amazon EMR

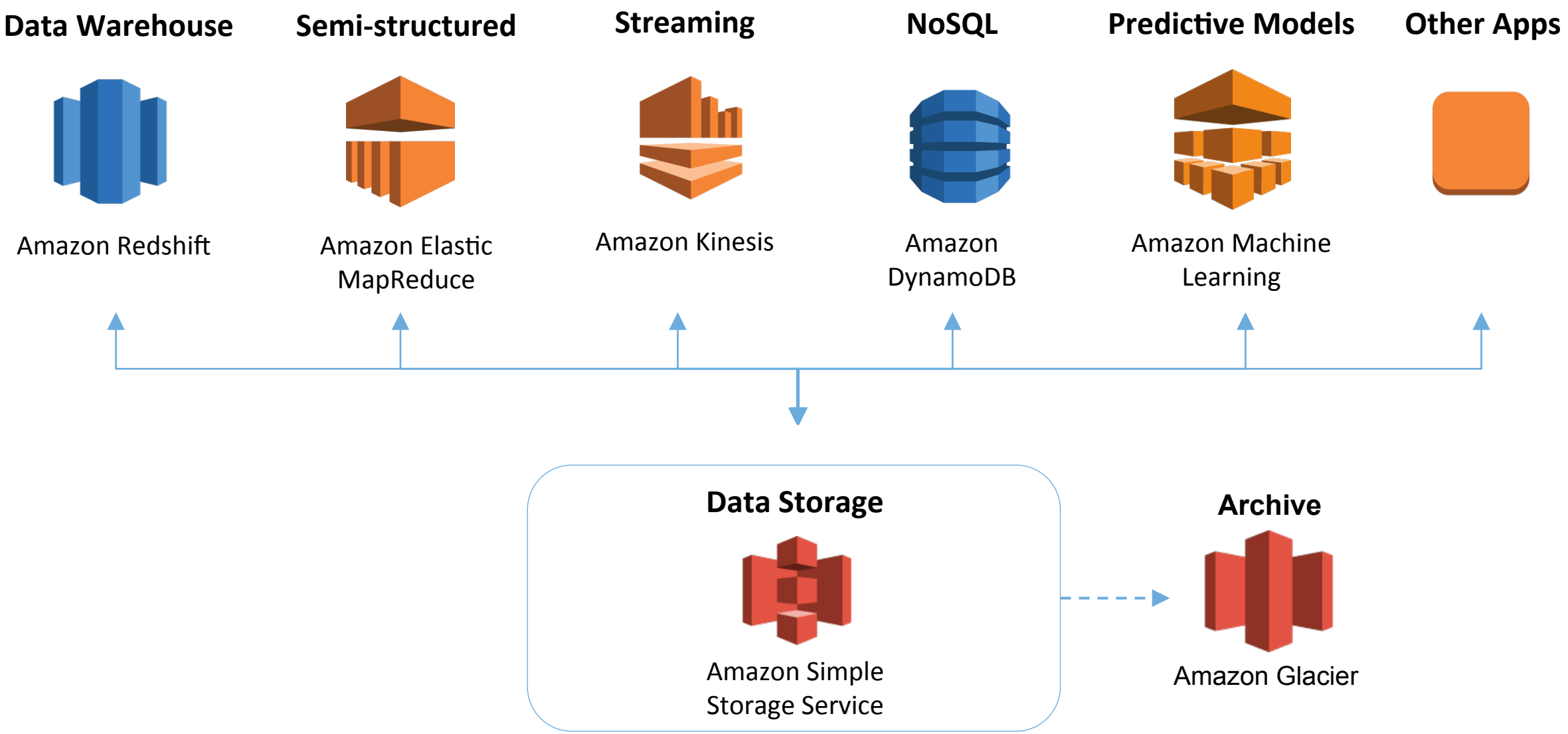
Amazon RedShift

Amazon RDS for MySQL

Euclid EventIQ

Euclid Analytics

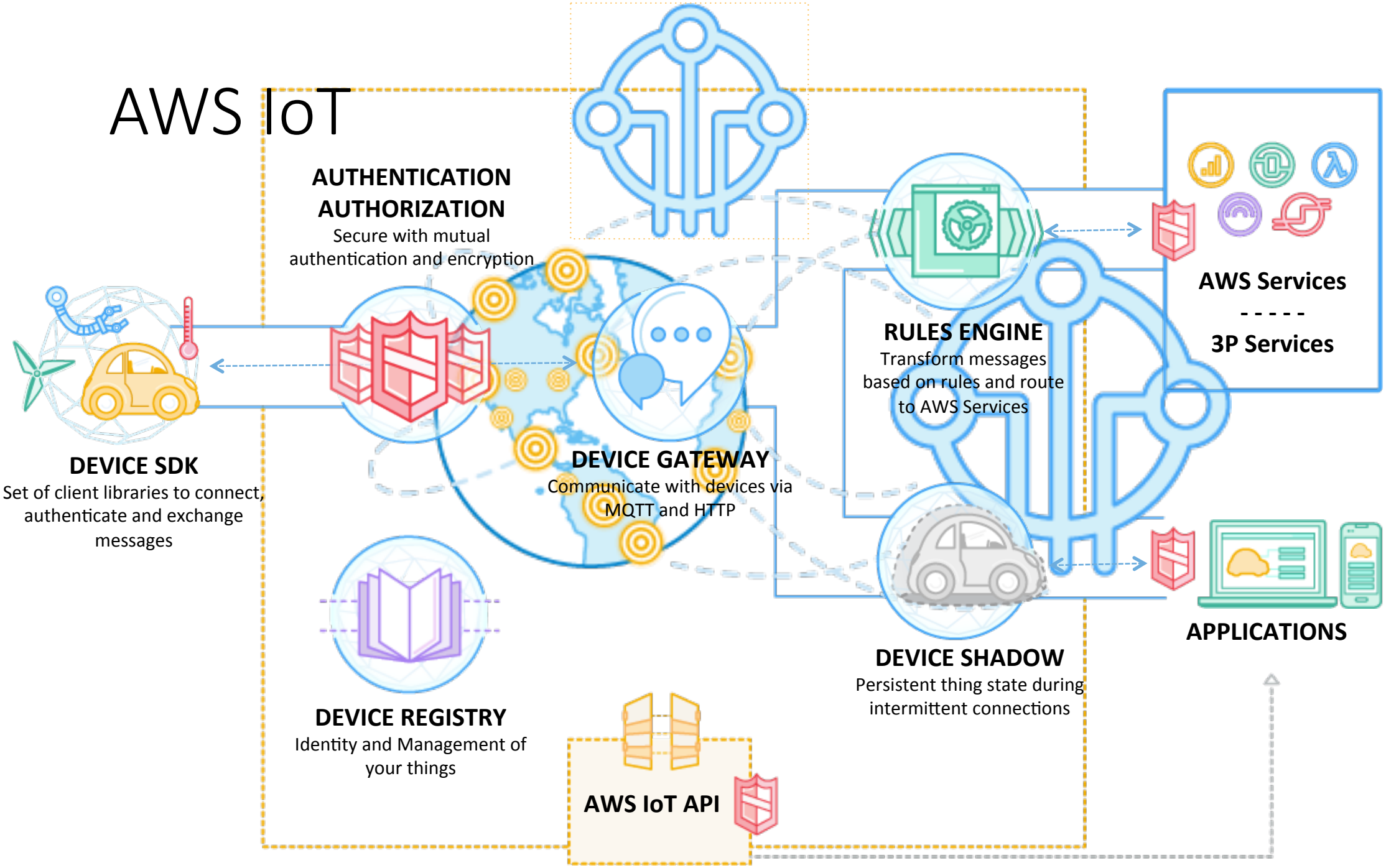
Use an optimal combination of highly interoperable services



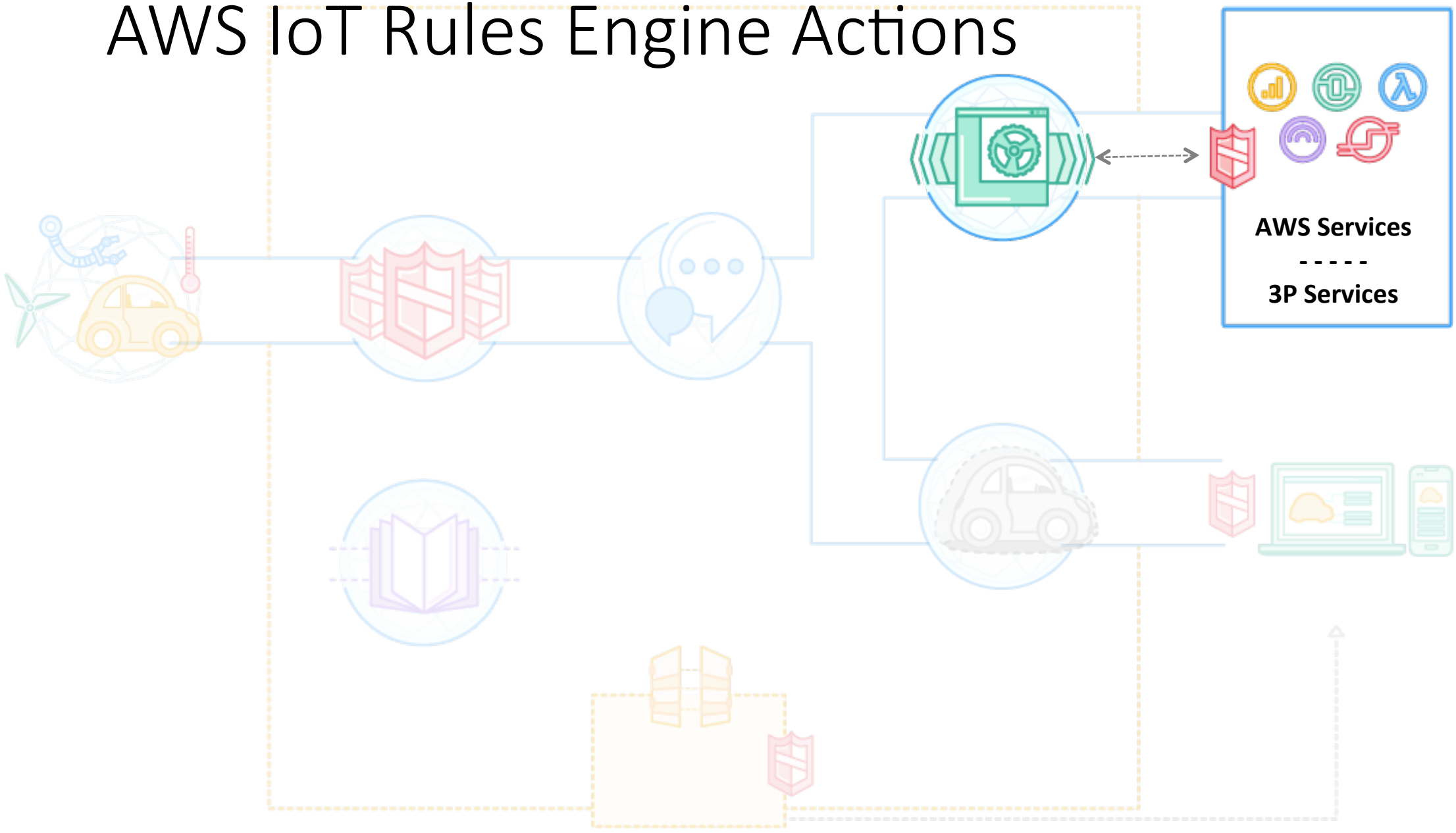
AWS IoT

“Securely connect one or one billion devices to AWS,
so they can interact with applications and other devices”

AWS IoT

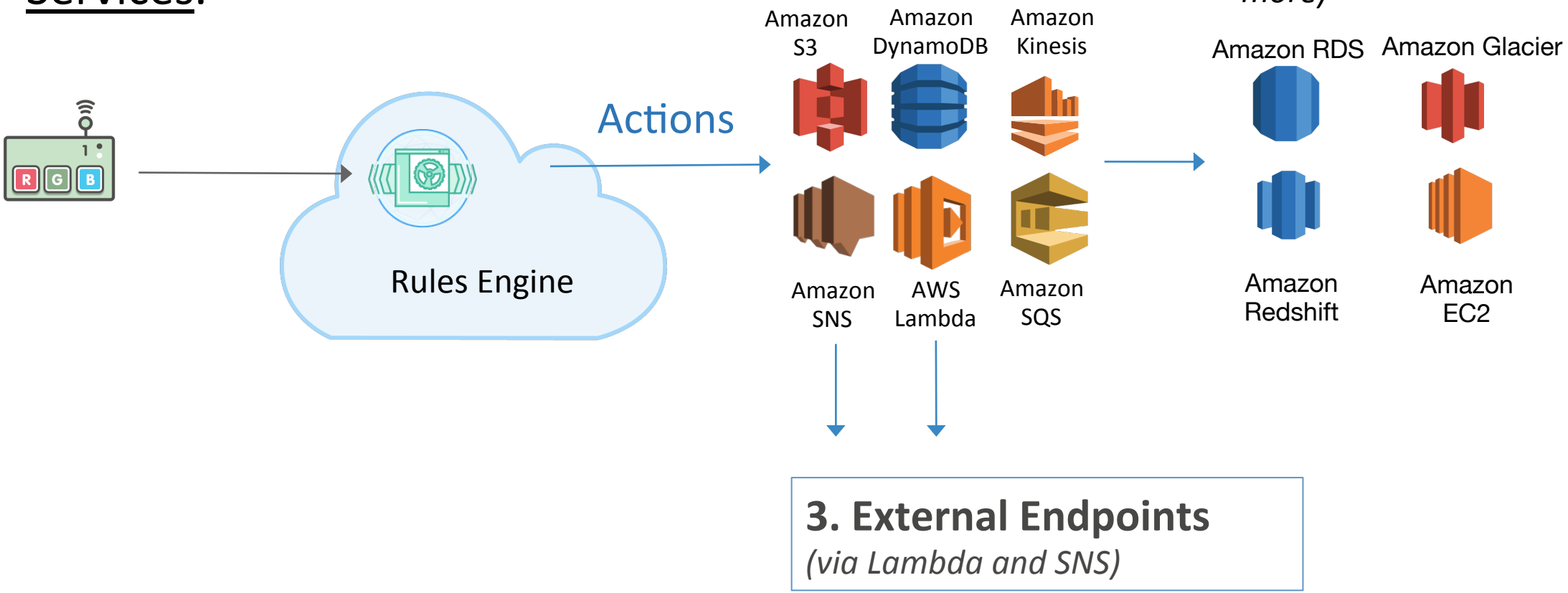

AUTHENTICATION
AUTHORIZATION
Secure with mutual authentication and encryption
RULES ENGINE
Transform messages based on rules and route to AWS Services
AWS Services
- - - - -
3P Services
DEVICE SDK
Set of client libraries to connect, authenticate and exchange messages
DEVICE GATEWAY
Communicate with devices via MQTT and HTTP
APPLICATIONS
DEVICE SHADOW
Persistent thing state during intermittent connections
DEVICE REGISTRY
Identity and Management of your things
AWS IoT API

AWS IoT Rules Engine Actions



AWS IoT Rules Engine

Rules Engine connects AWS IoT to External Endpoints and AWS Services.



AWS IoT Rules Engine Actions

Rules Engine evaluates inbound messages published into AWS IoT, transforms and delivers to the appropriate endpoint based on business rules.

Actions

<u>External endpoints</u> can be reached via Lambda and Simple Notification Service (SNS).

Invoke a Lambda function

Put object in an S3 bucket

Insert, Update, Read from a DynamoDB table

Publish to an SNS Topic or Endpoint

Amazon Elasticsearch

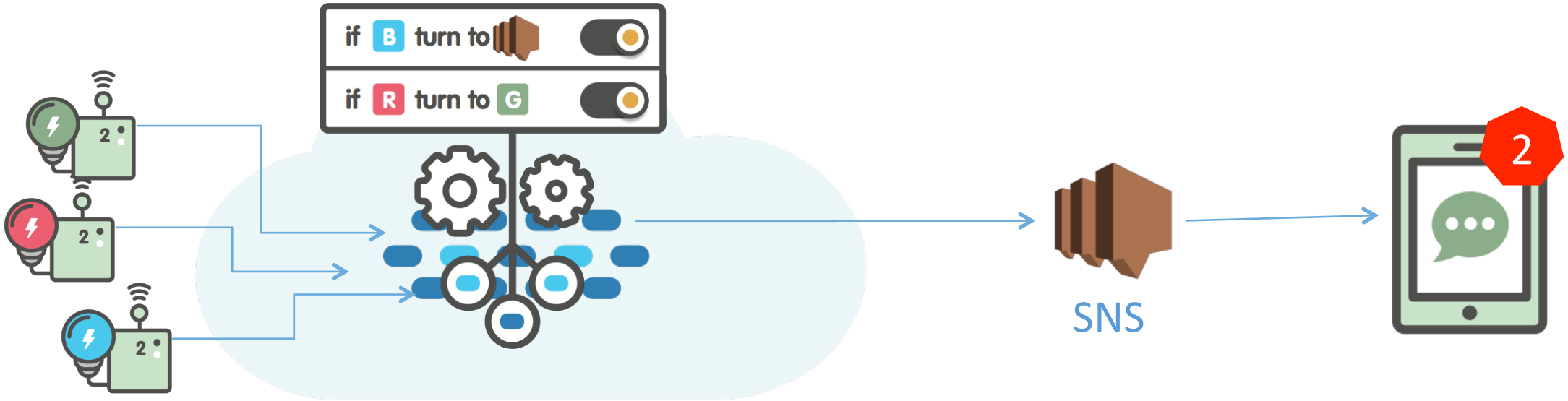
Amazon Machine Learning

Publish to an Amazon Kinesis stream

Amazon Kinesis Firehose

Republish to AWS IoT

AWS IoT Rules Engine & Amazon SNS



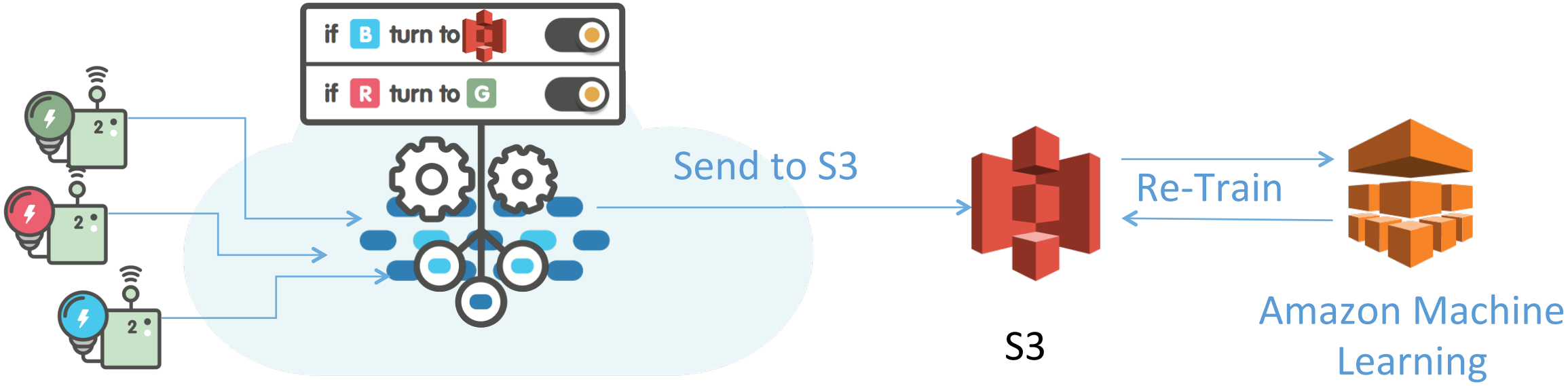
Push Notifications

Apple APNS Endpoint, Google GCM Endpoint, Amazon ADM Endpoint, Windows WNS

Amazon SNS -> HTTP Endpoint (Or SMS or Email)

Call HTTP based 3rd party endpoints through SNS with subscription and retry support

AWS IoT Rules Engine for Machine Learning



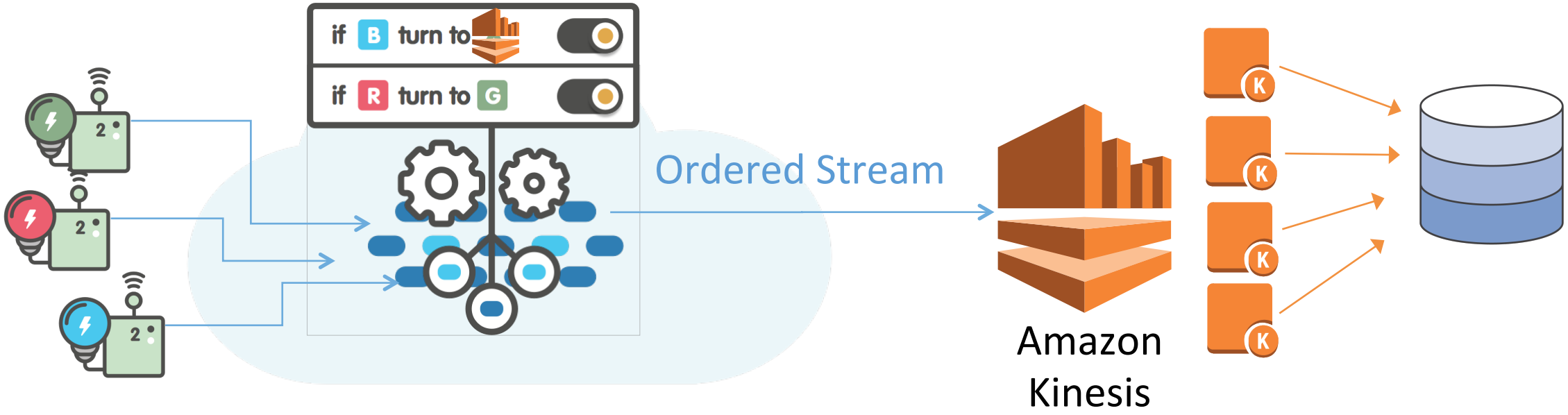
Anomaly Detection

Amazon Machine Learning can feed predictive evaluation criteria to the Rules Engine

Continuous Improvement Around Predication

Continuously look for outliers and re-calibrate the Amazon Machine Learning models

AWS IoT Rules Engine & Stream Data



N:1 Inbound Streams of Sensor Data **(Signal to Noise Reduction)**

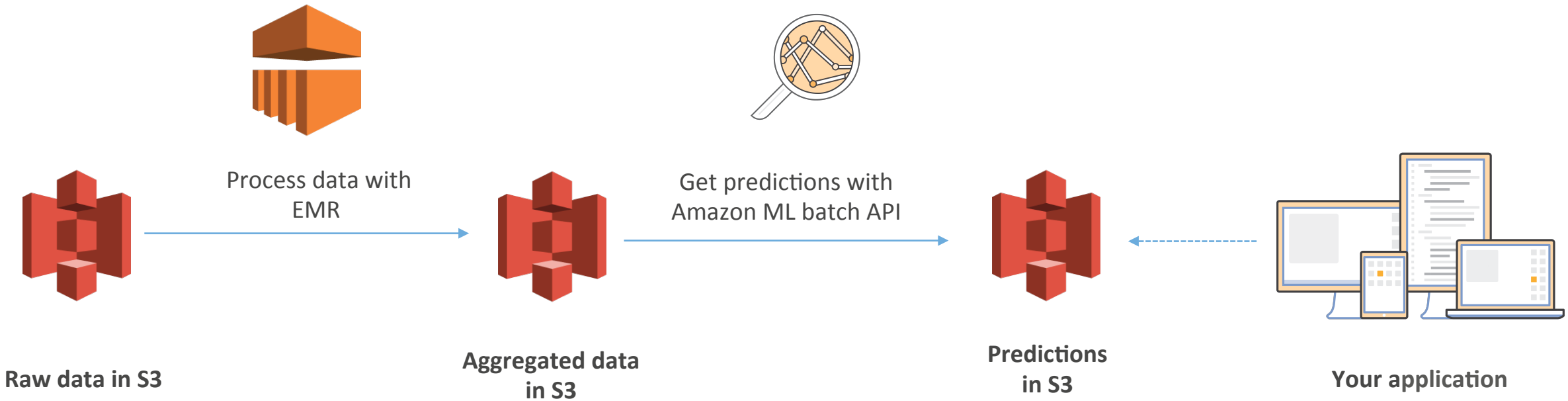
Rules Engine filters, transforms sensor data then sends aggregate to Amazon Kinesis

Amazon Kinesis Streams to Enterprise Applications

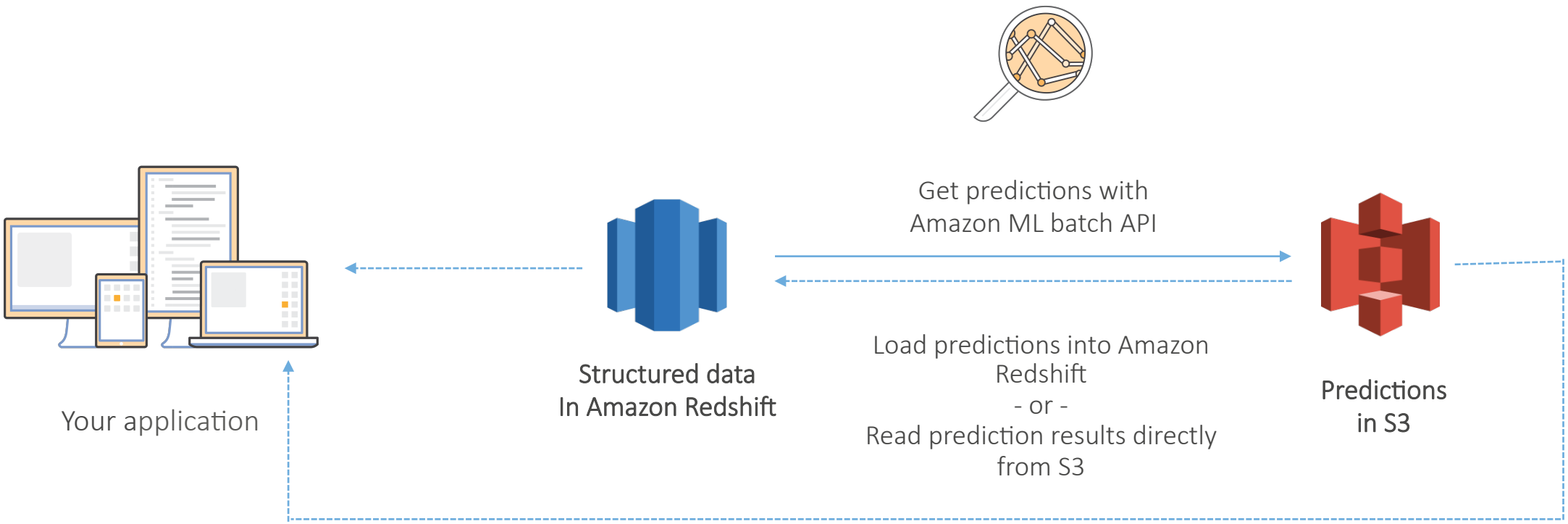
Simultaneously stream processed data to databases, applications, other AWS Services

Thank you!

Batch predictions with EMR



Batch predictions with Amazon Redshift



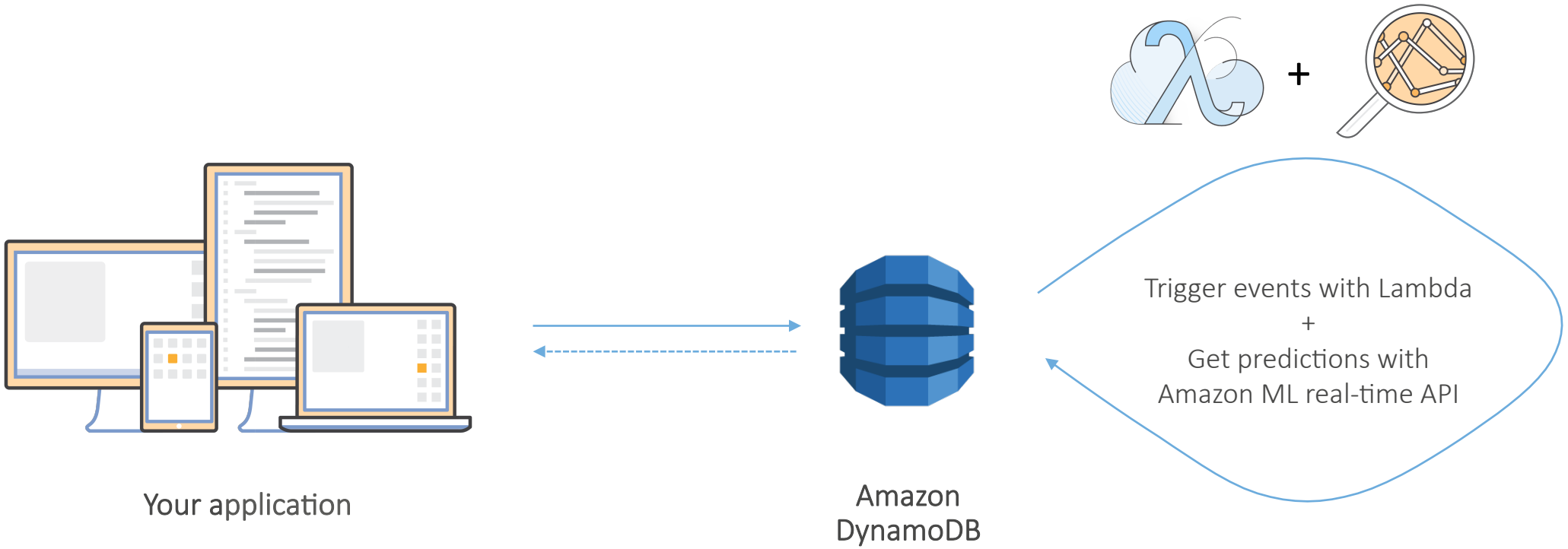
Real-time predictions for interactive applications

Your application

Get predictions with
Amazon ML real-time API

Amazon ML service

Adding predictions to an existing data flow



Recommendation engine