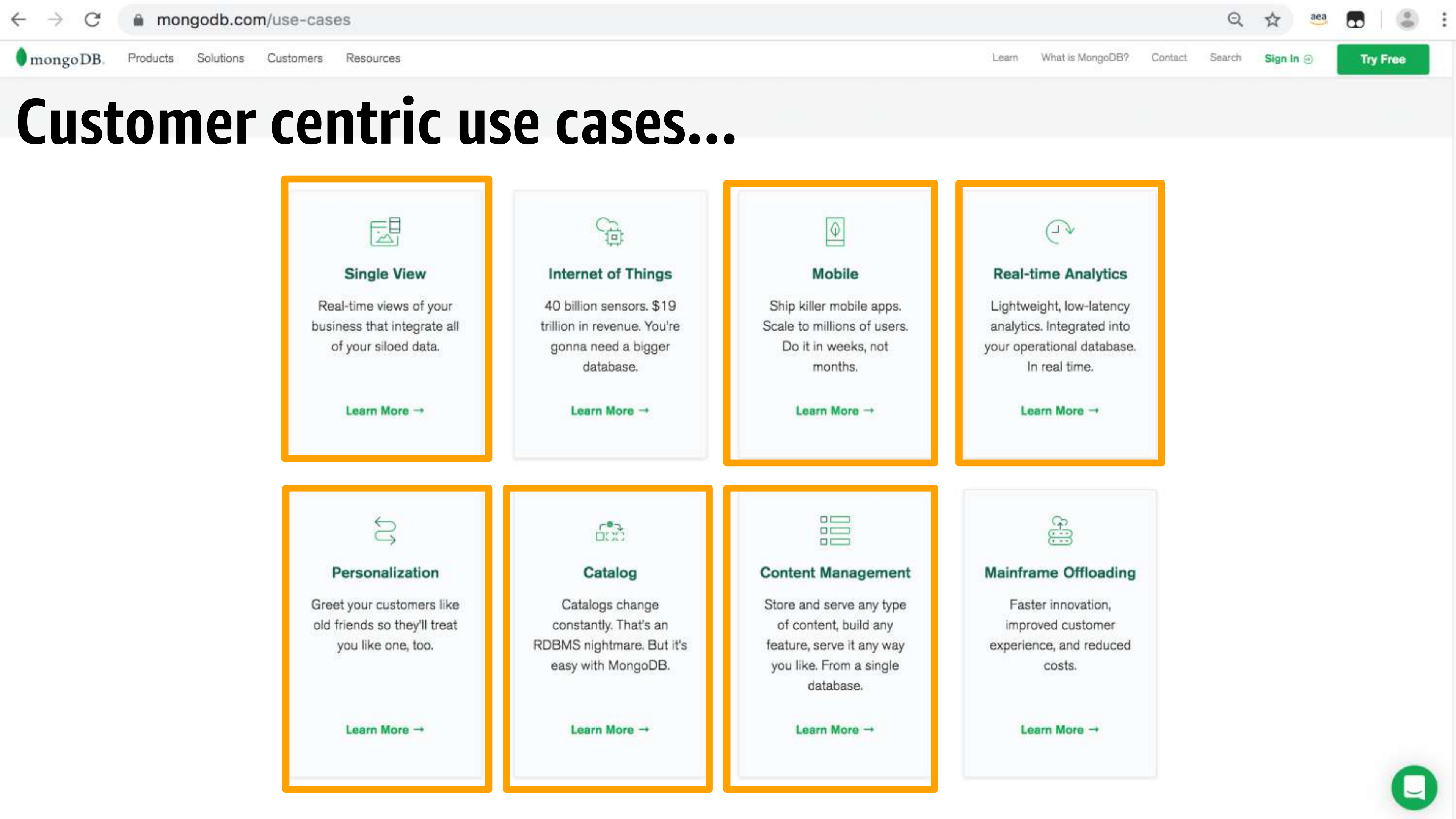
AWS | Using AWS to Transform Customer Data in MongoDB into AI-driven Personalization

Dylan Tong, Machine Learning Partner Solutions Architect

aws machine learning

Customer centric use cases...



Single View

Real-time views of your business that integrate all of your siloed data.

Learn More →

Internet of Things

40 billion sensors. $19 trillion in revenue. You're gonna need a bigger database.

Learn More →

Mobile

Ship killer mobile apps. Scale to millions of users. Do it in weeks, not months.

Learn More →

Real-time Analytics

Lightweight, low-latency analytics. Integrated into your operational database. In real time.

Learn More →

Personalization

Greet your customers like old friends so they'll treat you like one, too.

Learn More →

Catalog

Catalogs change constantly. That's an RDBMS nightmare. But it's easy with MongoDB.

Learn More →

Content Management

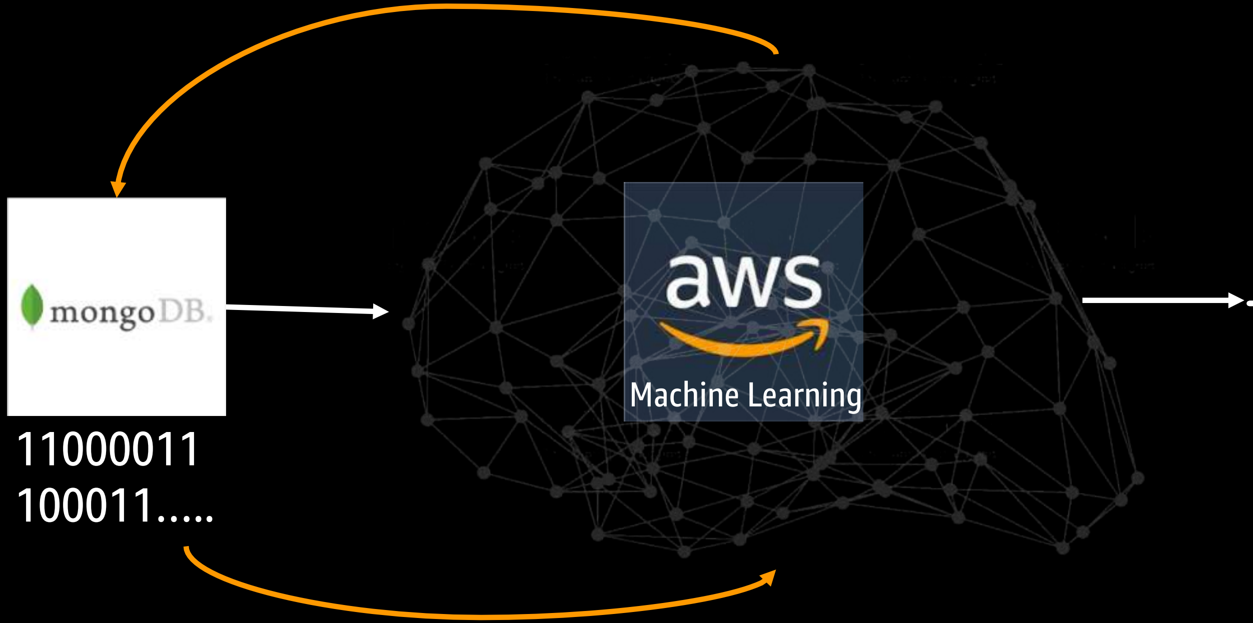
Store and serve any type of content, build any feature, serve it any way you like. From a single database.

Learn More →

Mainframe Offloading

Faster innovation, improved customer experience, and reduced costs.

Learn More →



Many possibilities…

Forecasting

Customer Churn Prediction

Personalization

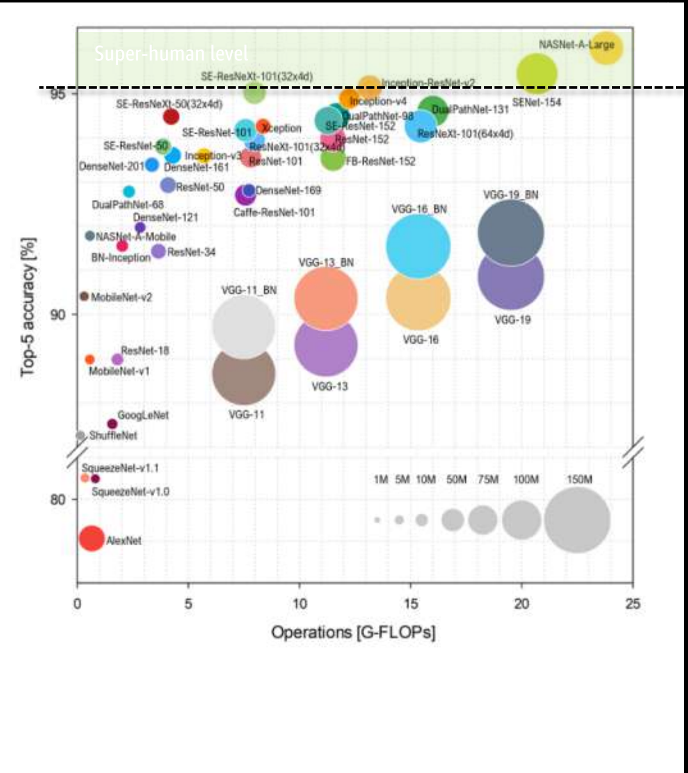
.

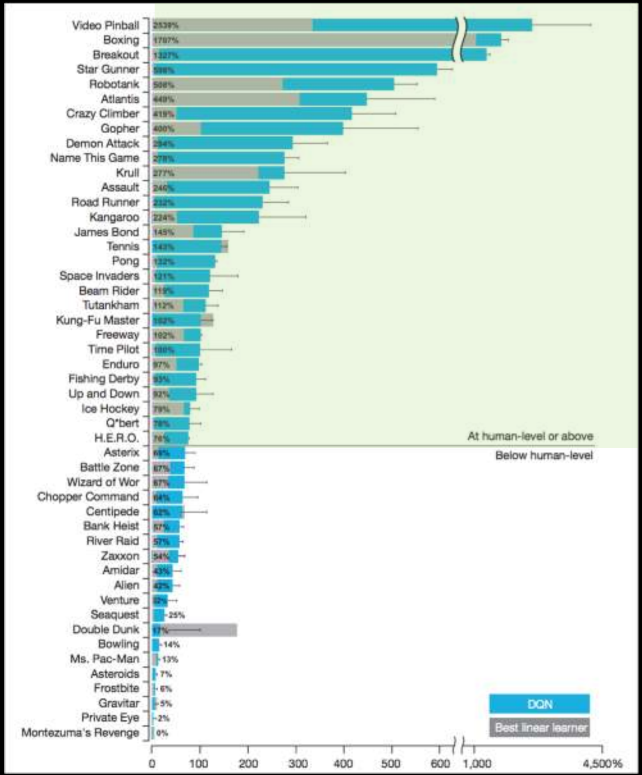
.

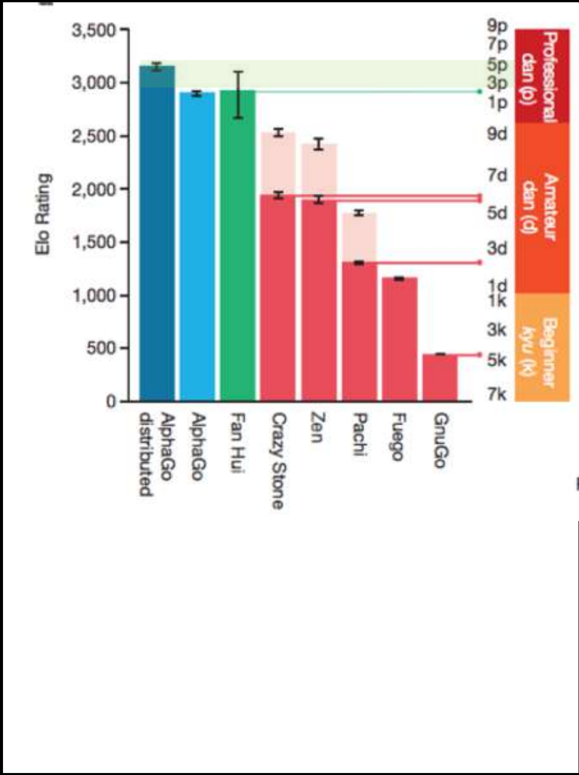
.

Deep Convolutional Neural Networks

Deep Reinforcement Learning



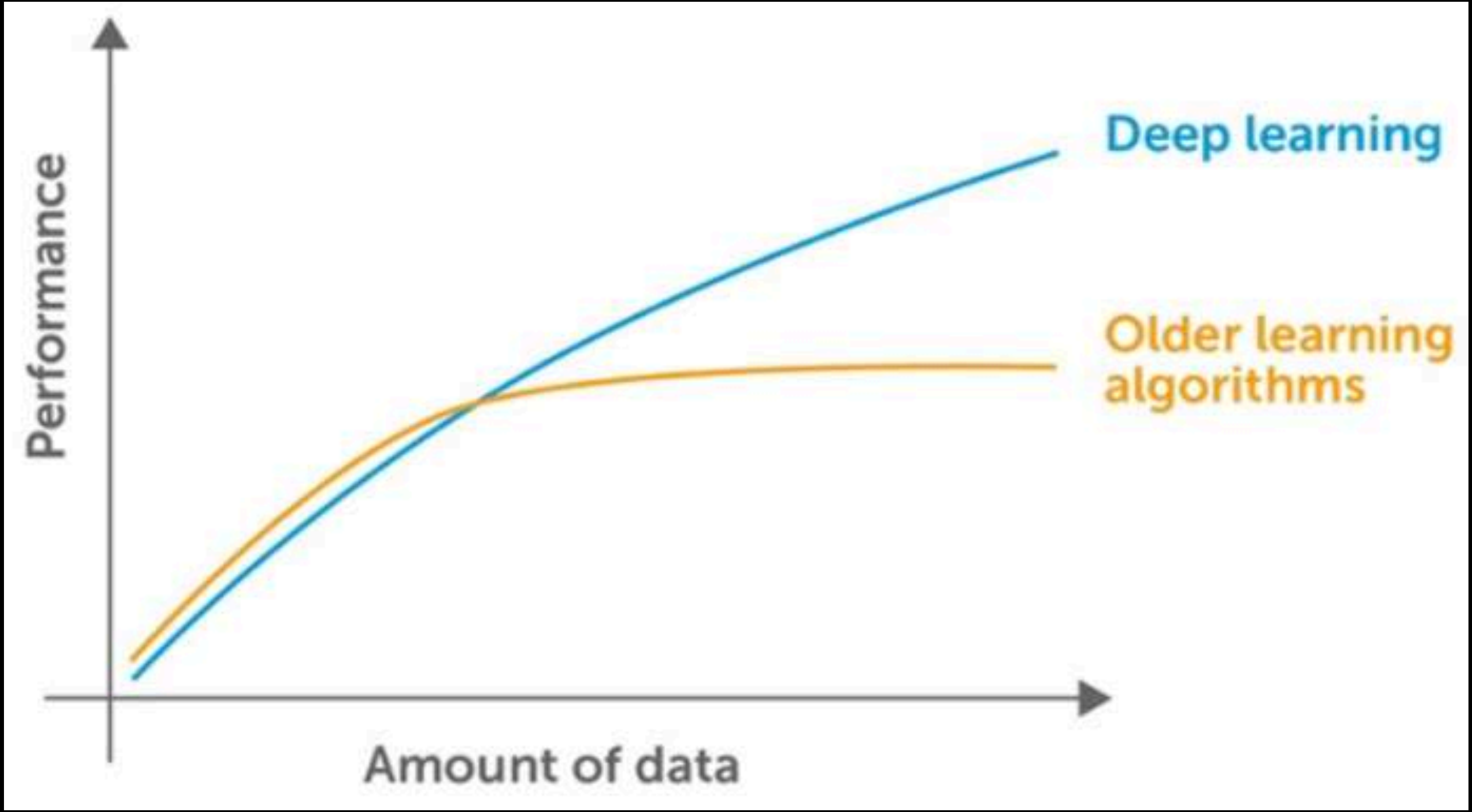




Source: arxiv.org/pdf/1810.00736.pdf, 2018

Source: DeepMind Research, 2015

Source: DeepMind Research, 2016



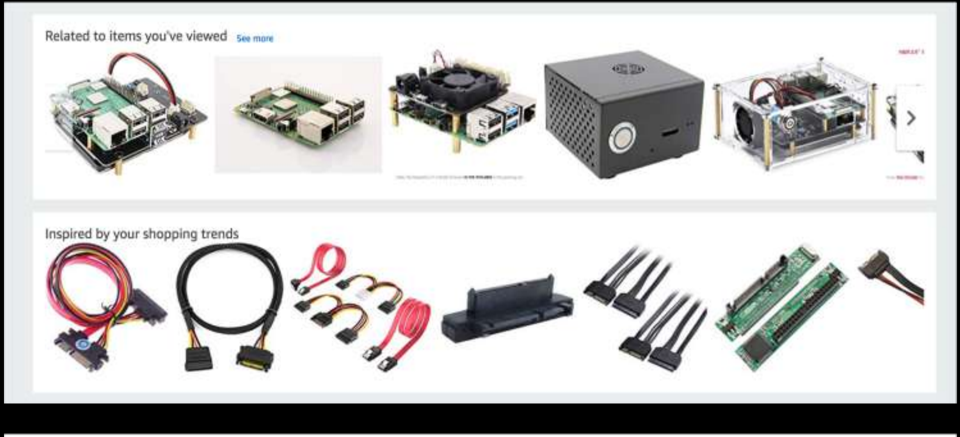
Source: https://www.researchgate.net/figure/Andrew-Ngs-30-graph-shows-how-deep-learning-is-said-to-outperforms-traditional_fig6_324476862

Reinvent the Customer Experience

mongoDB

Amazon Personalize

- Session based recommendations
- Predictive Customer Analytics



Related to items you've viewed See more

Inspired by your shopping trends

Amazon SageMaker RL

- Contextual Bandits: uplift conversion rates

Conversational AI

- Humanize your apps with life-like voices

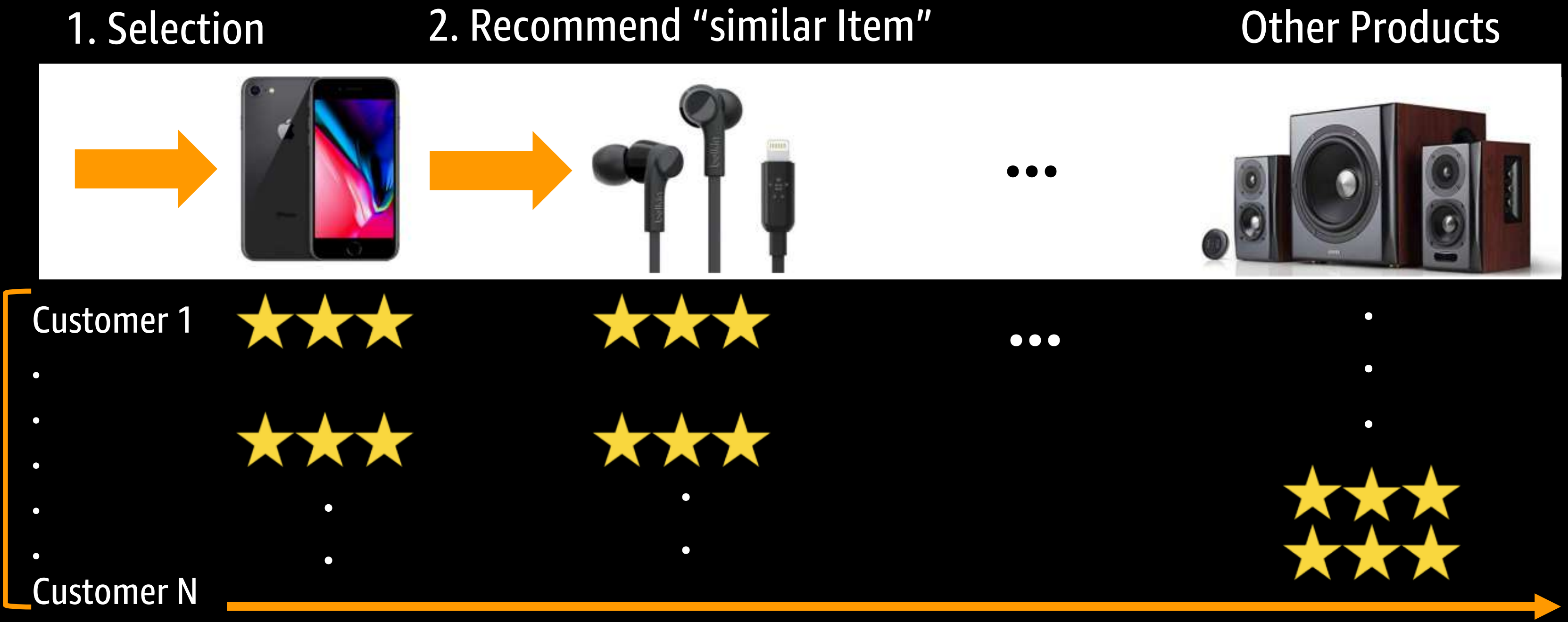
Recommender Modernization

Related to items you've viewed

Inspired by your shopping trends

Classic Recommenders: Item-Item Collaborative Filtering

Calculate "rating vector" for each product and calculate vector distance to measure similarity



1. Selection

2. Recommend "similar Item"

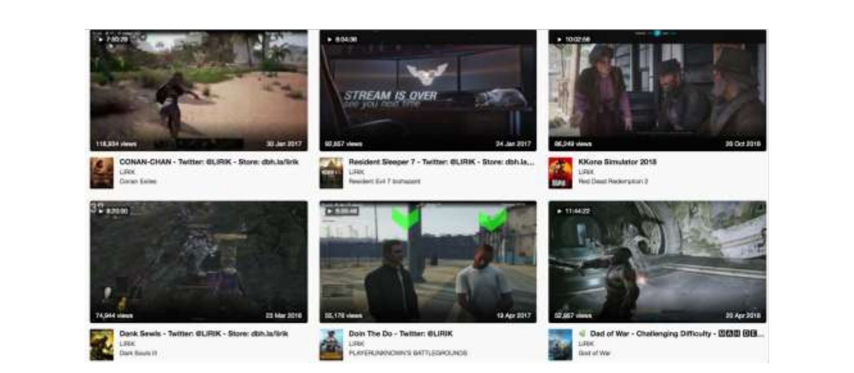
Other Products

Customer 1

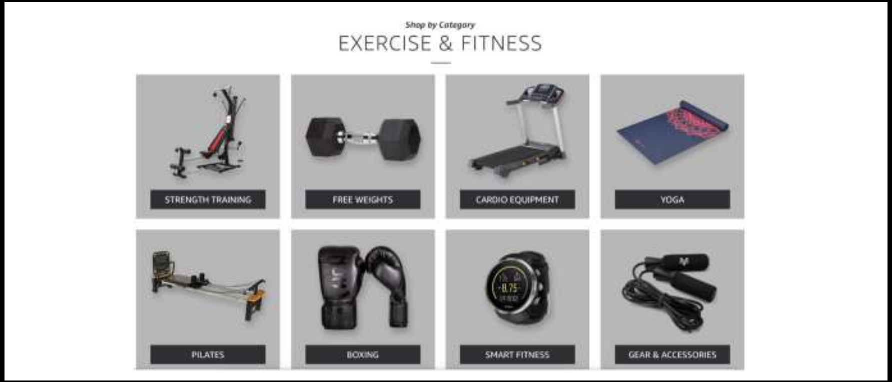
Customer N

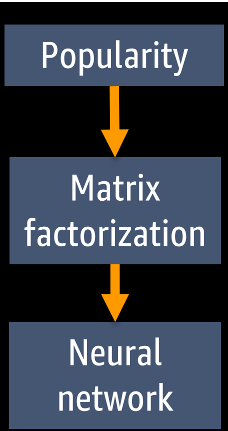
Degree of similarity by co-rating

Deep learning techniques have a direct impact on the bottom line



STREAM IS OVER
118,834 views
30 Jan 2017
CONAN-CHAN - Twitter: @LIRIK - Store: dbh.la/lirik
LIRIK
Conan Exiles
92,857 views
24 Jan 2017
Resident Sleeper 7 - Twitter: @LIRIK - Store: dbh.la,...
LIRIK
Resident Evil 7 biohazard
86,249 views
30 Oct 2018
KKona Simulator 2018
LIRIK
Red Dead Redemption 2
74,944 views
23 Mar 2016
Dank Sewls - Twitter: @LIRIK - Store: dbh.la/lirik
LIRIK
Dark Souls III
55,176 views
19 Apr 2017
Doin The Do - Twitter: @LIRIK
LIRIK
PLAYERUNKNOWN'S BATTLEGROUNDS
52,857 views
20 Apr 2018
Dad of War - Challenging Difficulty - ...
LIRIK
God of War





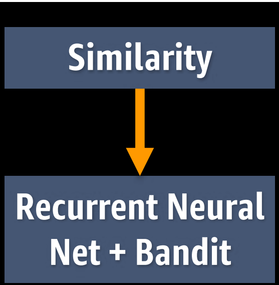
Popularity

Matrix factorization

Neural network

+15.4%
Engagement

+7.4%
Engagement



Similarity

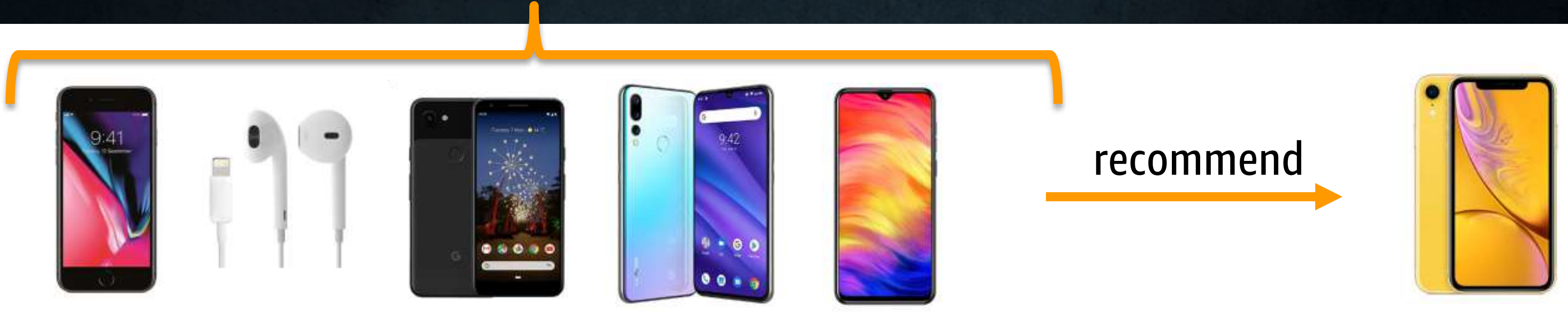
Recurrent Neural Net + Bandit

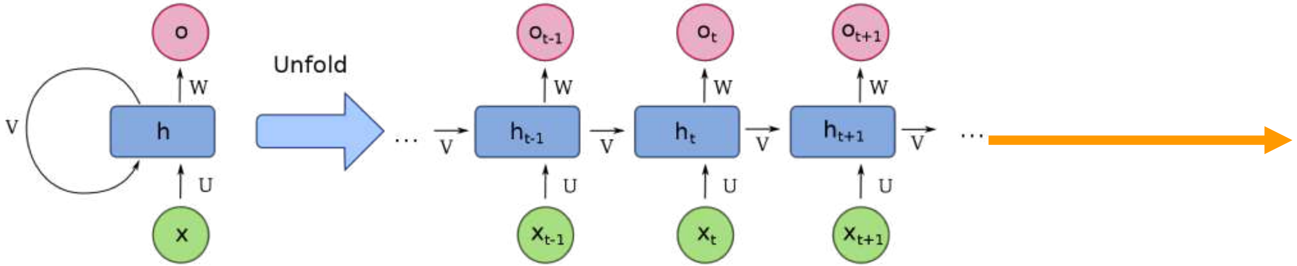
+20%
Click Through

https://www.slideshare.net/AmazonWebServices/add-realtime-personalization-and-recommendations-to-your-applications-aim395-aws-reinvent-2018

RNN: History and User Representation

Customers interaction history: clicks, ratings, purchases…



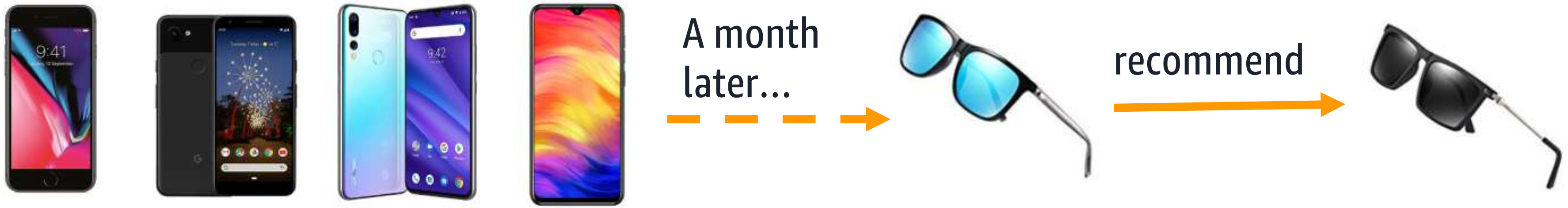


User Representation

Image: François Deloche (https://commons.wikimedia.org/wiki/File:Recurrent_neural_network_unfold.svg), https://creativecommons.org/licenses/by-sa/4.0/legalcode

HRNN: Modeling Sessions

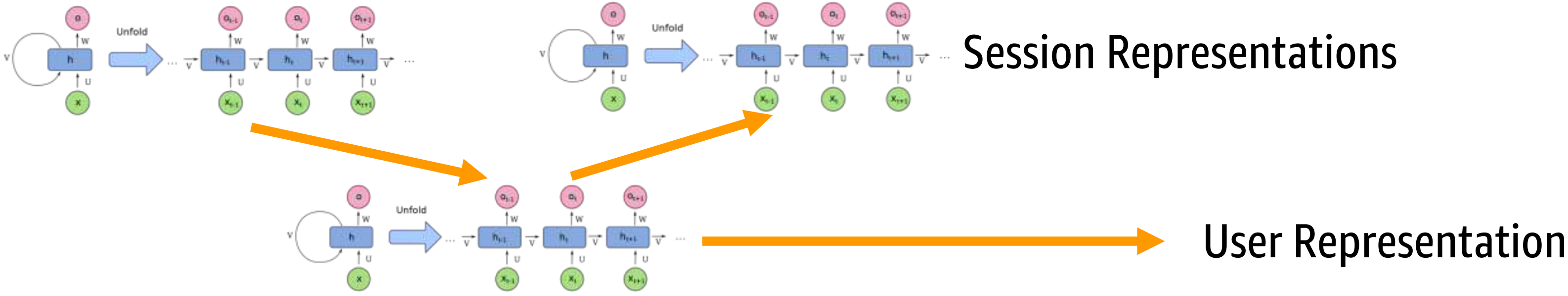
Insight: Evolution of interests and disinterests predict future preferences…



A month later…

recommend

Interactions, ordering and timing all matter…



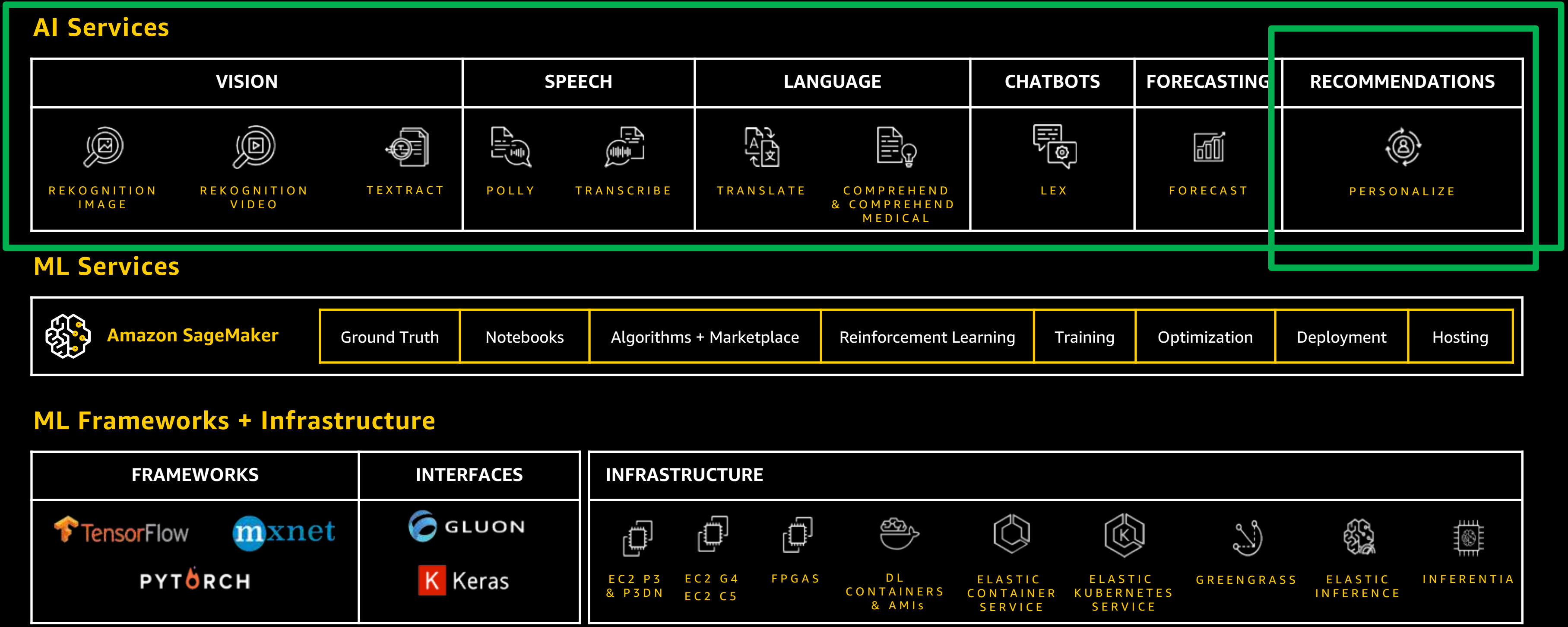
Session Representations

User Representation

Image: François Deloche (https://commons.wikimedia.org/wiki/File:Recurrent_neural_network_unfold.svg), https://creativecommons.org/licenses/by-sa/4.0/legalcode

THE AWS ML STACK

Broadest and deepest set of capabilities



AI Services

VISION			SPEECH		LANGUAGE		CHATBOTS	FORECASTING	RECOMMENDATIONS
REKOGNITION IMAGE	REKOGNITION VIDEO	TEXTRACT	POLLY	TRANSCRIBE	TRANSLATE	COMPREHEND & COMPREHEND MEDICAL	LEX	FORECAST	PERSONALIZE

ML Services

Amazon SageMaker

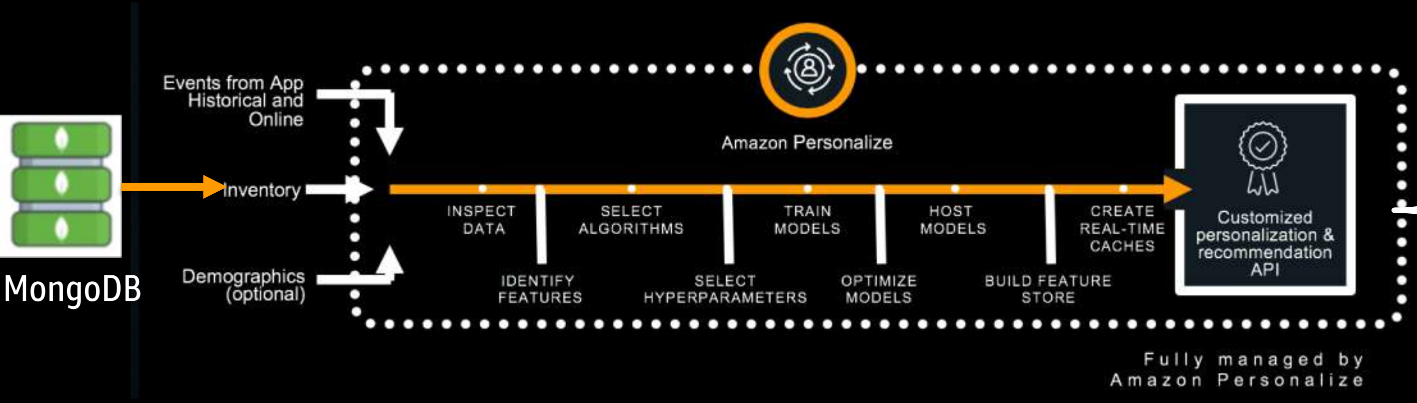
Ground Truth	Notebooks	Algorithms + Marketplace	Reinforcement Learning	Training	Optimization	Deployment	Hosting

ML Frameworks + Infrastructure

FRAMEWORKS	INTERFACES
TensorFlow mxnet PYTORCH	GLUON Keras

INFRASTRUCTURE								
EC2 P3 & P3DN	EC2 G4 EC2 C5	FPGAS	DL CONTAINERS & AMIs	ELASTIC CONTAINER SERVICE	ELASTIC KUBERNETES SERVICE	GREENGRASS	ELASTIC INFERENCE	INFERENTIA

Amazon Personalize: AutoML



Real-time Recommendations API

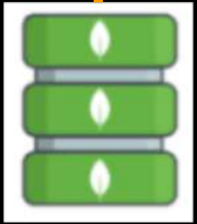
```
GetRecommendations:

{ "campaignArn": "string",
"itemId": "string",
"numResults": number,
"userId": "string" }

GetRankedList:

{ "campaignArn": "string",
"inputList": [ "string" ],
"userId": "string" }
```

Cold starts and Online Learning

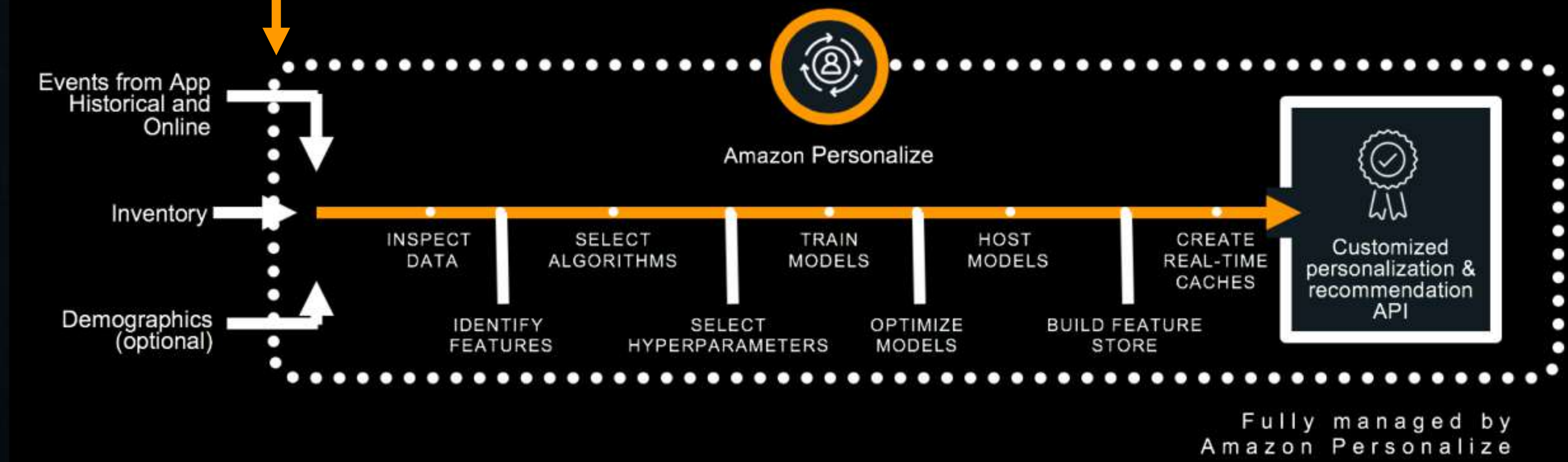
Application

AWS SDK:
Event Recorder

```
personalize_events.put_events(
trackingId = 'tracking_id’,
userId= 'USER_ID’,
sessionId = 'session_id’,
eventList = [{ 'sentAt': TIMESTAMP,
'eventType': 'EVENT_TYPE’,
'properties': "{\"itemId\": \"ITEM_ID\"}" }])
```



MongoDB

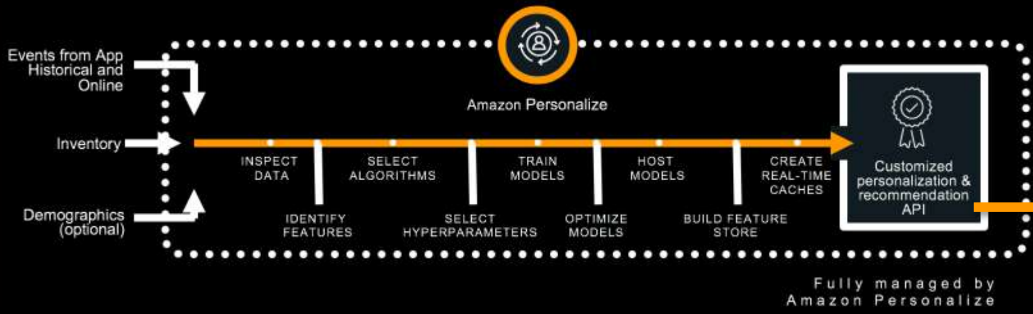


Fully managed by
Amazon Personalize

Predictive Customer Insights

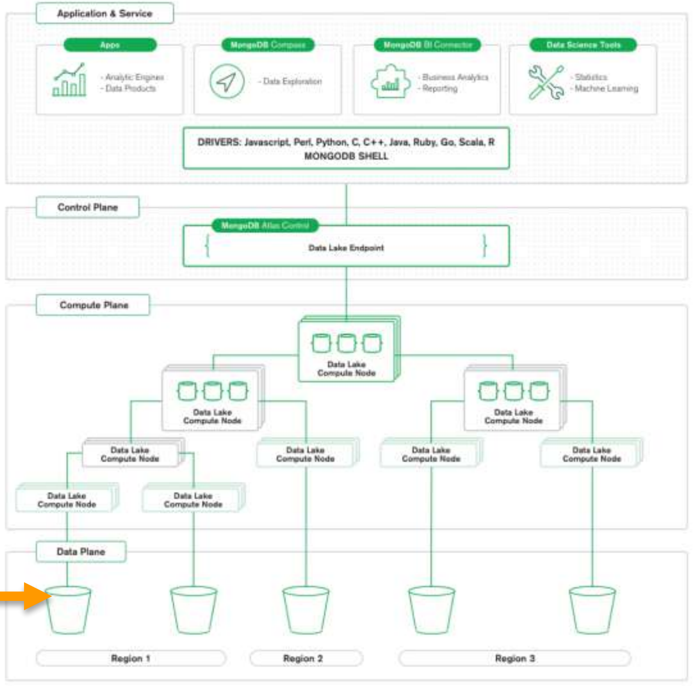
- Predictive Customer-level Marketing

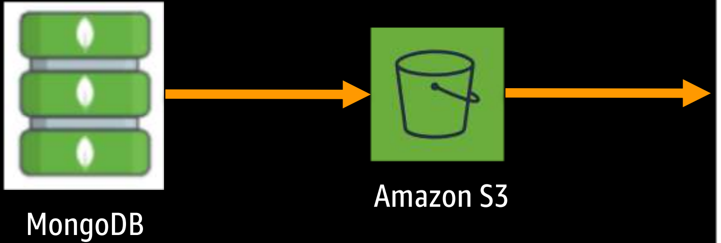
- Reverse Recommendations: query the users most likely to be interested in product(s).

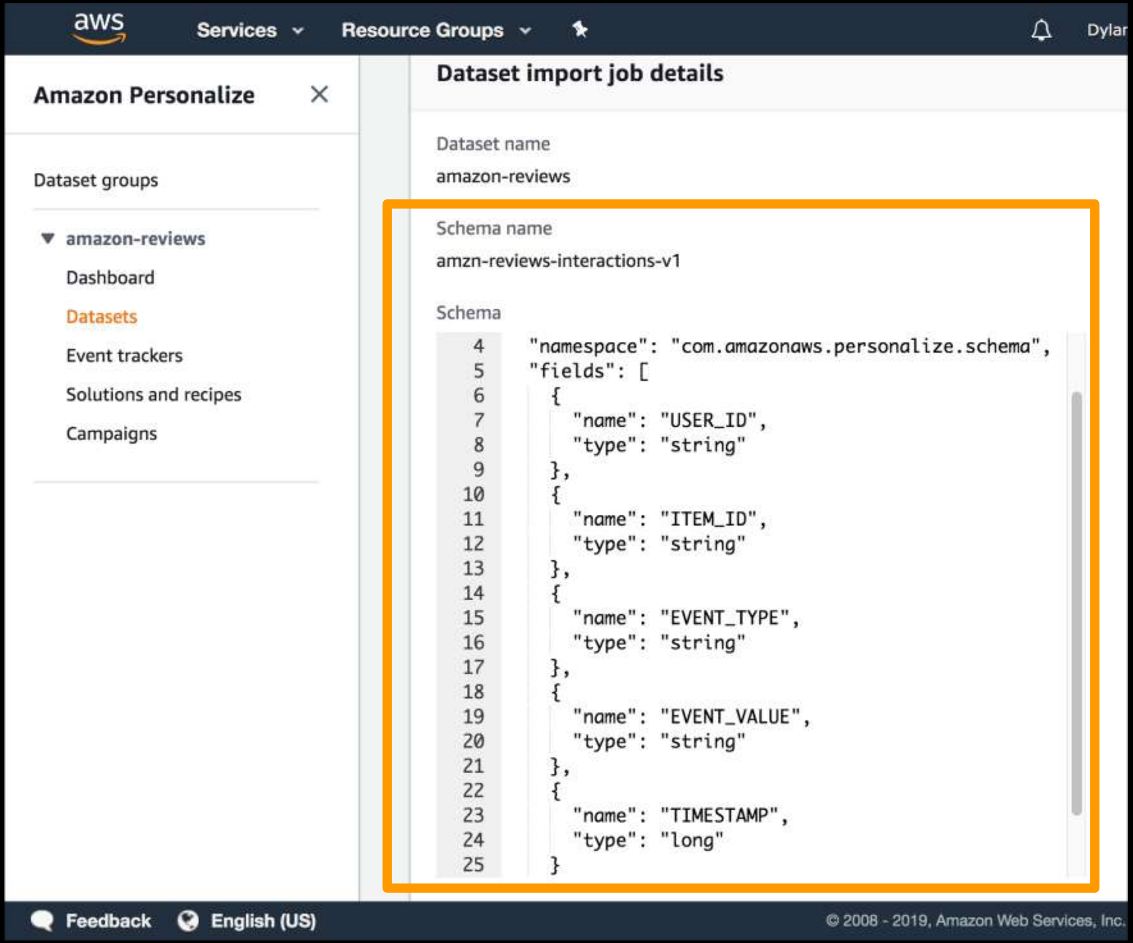


Batch Scores

MongoDB Atlas Data Lake Architecture







Dataset import job details

Dataset name

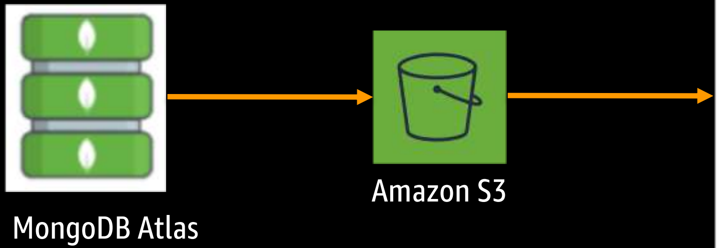
amazon-reviews

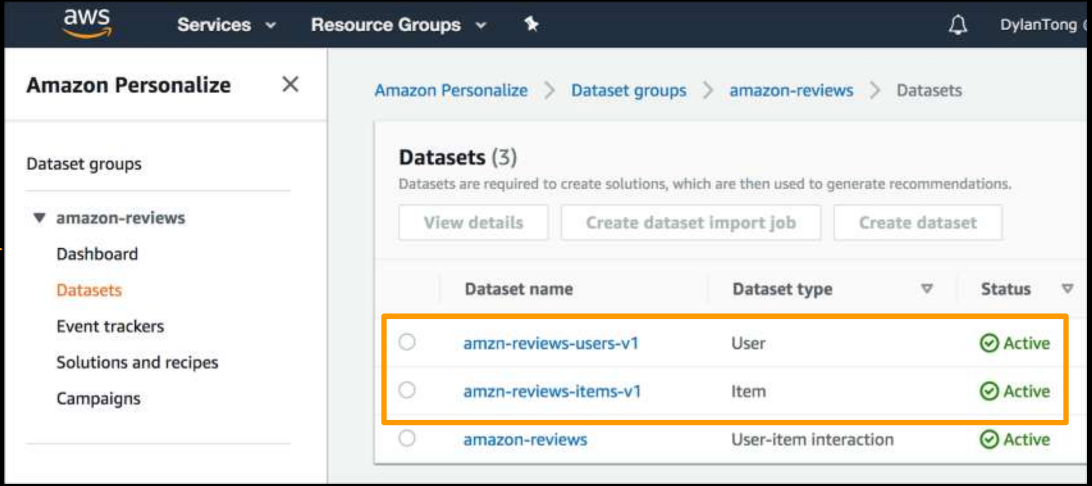
Schema name

amzn-reviews-interactions-v1

Schema

```
  "namespace": "com.amazonaws.personalize.schema",
  "fields": [
    {
      "name": "USER_ID",
      "type": "string"
    },
    {
      "name": "ITEM_ID",
      "type": "string"
    },
    {
      "name": "EVENT_TYPE",
      "type": "string"
    },
    {
      "name": "EVENT_VALUE",
      "type": "string"
    },
    {
      "name": "TIMESTAMP",
      "type": "long"
    }
```

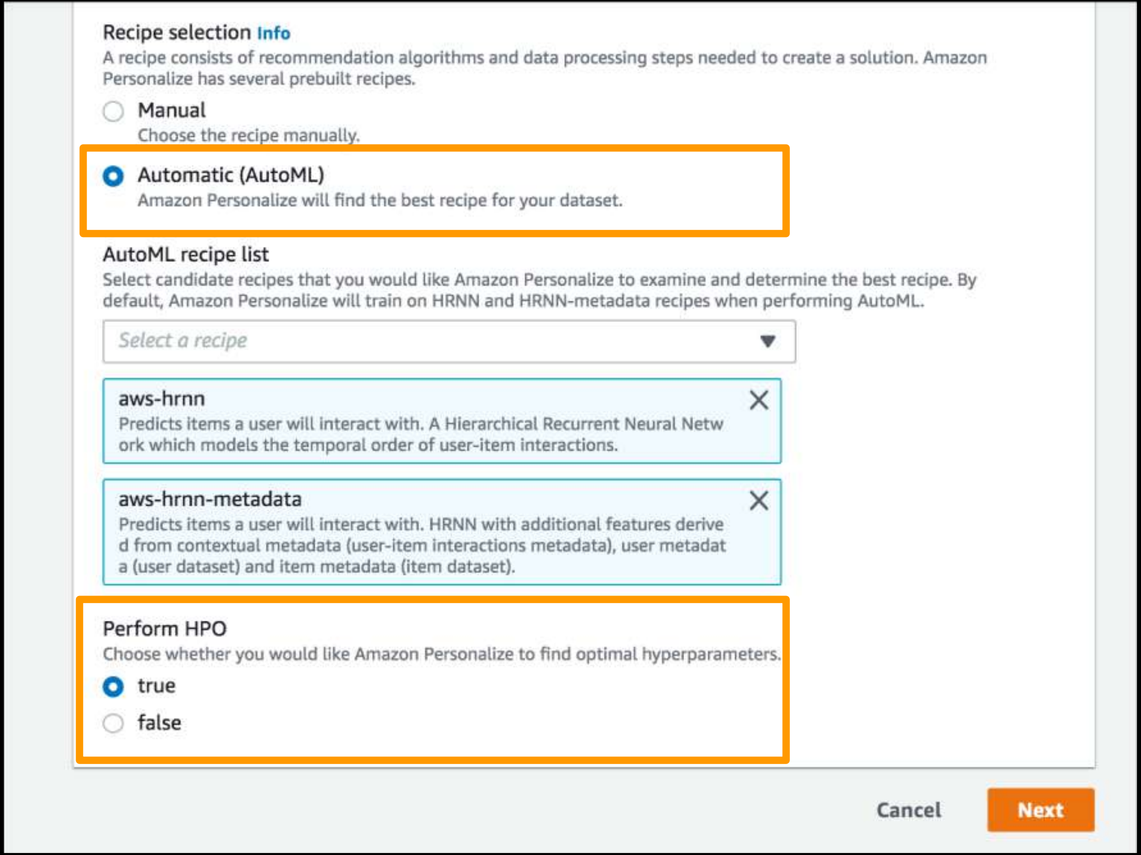




Datasets (3)

Datasets are required to create solutions, which are then used to generate recommendations.

	Dataset name	Dataset type	Status
○	amzn-reviews-users-v1	User	✓ Active
○	amzn-reviews-items-v1	Item	✓ Active
○	amazon-reviews	User-item interaction	✓ Active



Recipe selection Info

A recipe consists of recommendation algorithms and data processing steps needed to create a solution. Amazon Personalize has several prebuilt recipes.

○ Manual
Choose the recipe manually.

● Automatic (AutoML)
Amazon Personalize will find the best recipe for your dataset.

AutoML recipe list

Select candidate recipes that you would like Amazon Personalize to examine and determine the best recipe. By default, Amazon Personalize will train on HRNN and HRNN-metadata recipes when performing AutoML.

Select a recipe ▼

aws-hrnn ✕
Predicts items a user will interact with. A Hierarchical Recurrent Neural Netw
ork which models the temporal order of user-item interactions.

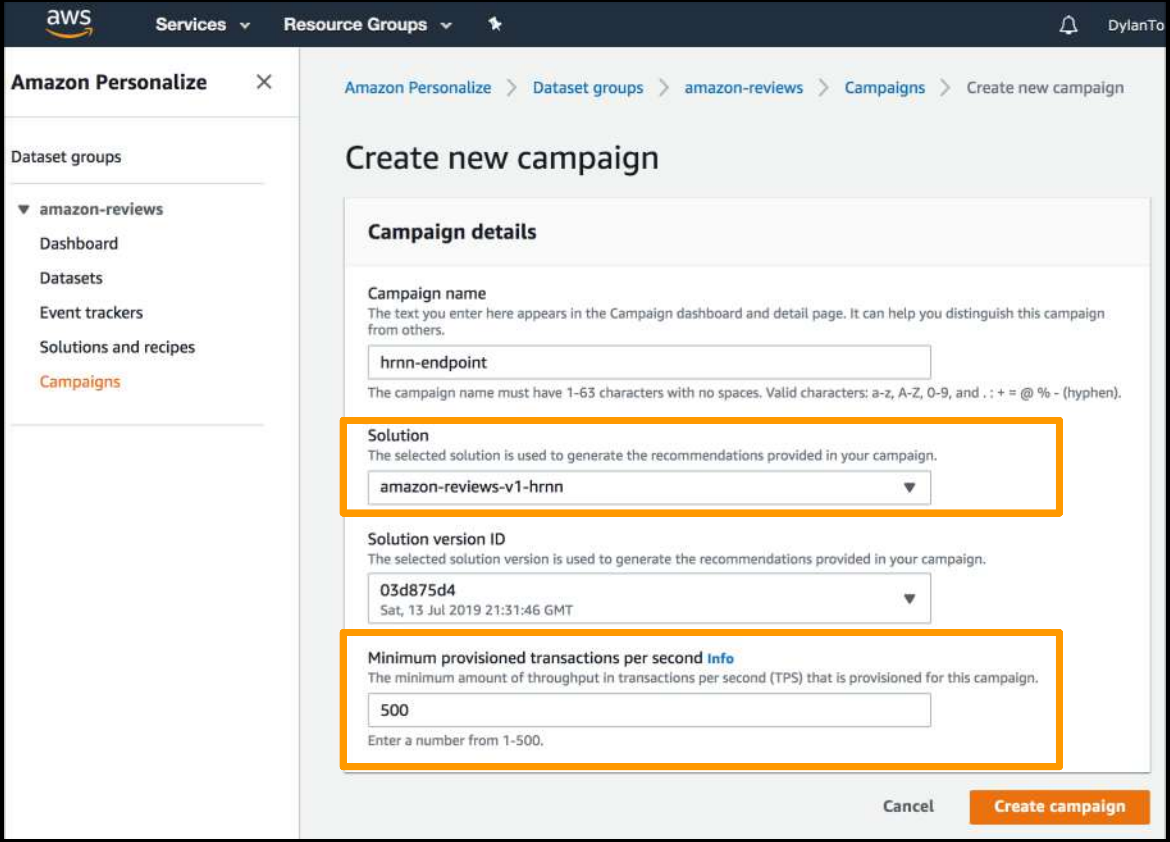
aws-hrnn-metadata ✕
Predicts items a user will interact with. HRNN with additional features derive
d from contextual metadata (user-item interactions metadata), user metadat
a (user dataset) and item metadata (item dataset).

Perform HPO

Choose whether you would like Amazon Personalize to find optimal hyperparameters.

● true

○ false



Create new campaign

Campaign details

Campaign name
The text you enter here appears in the Campaign dashboard and detail page. It can help you distinguish this campaign from others.

hrnn-endpoint

The campaign name must have 1-63 characters with no spaces. Valid characters: a-z, A-Z, 0-9, and . : + = @ % - (hyphen).

Solution
The selected solution is used to generate the recommendations provided in your campaign.

amazon-reviews-v1-hrnn ▼

Solution version ID
The selected solution version is used to generate the recommendations provided in your campaign.

03d875d4
Sat, 13 Jul 2019 21:31:46 GMT ▼

Minimum provisioned transactions per second Info
The minimum amount of throughput in transactions per second (TPS) that is provisioned for this campaign.

500

Enter a number from 1-500.

Multivariate Optimization

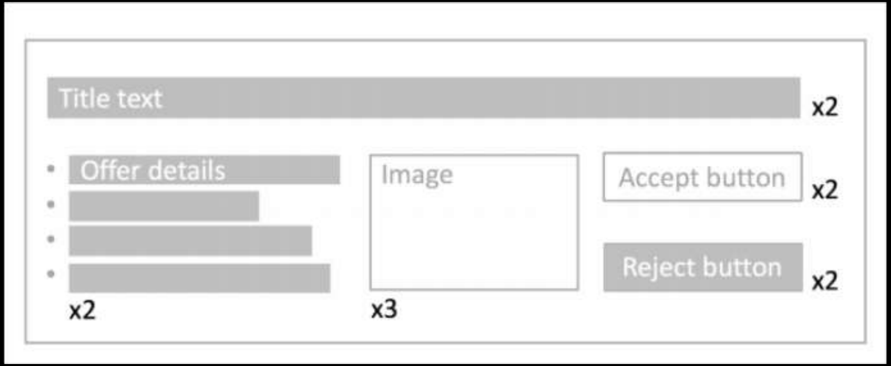
amazon
amazon
amazon

Source: https://www.kdd.org/kdd2017/papers/view/an-efficient-bandit-algorithm-for-realtime-multivariate-optimization

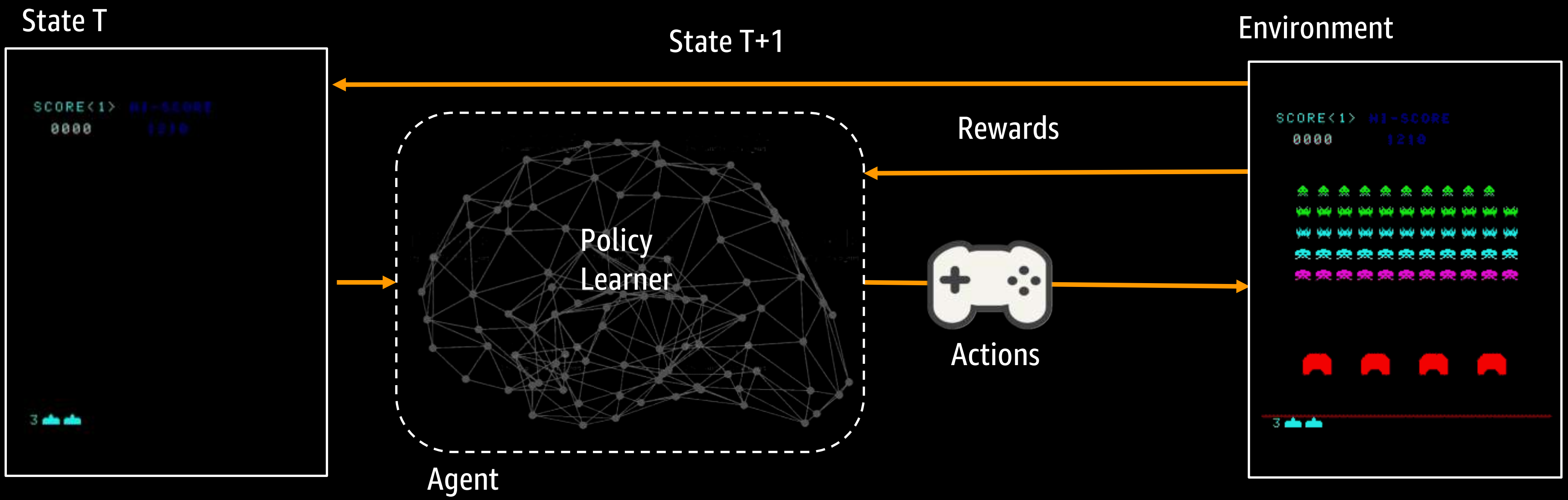
Contextual Bandits

After only a single week of online optimization, we saw a 21% conversion increase compared to the median layout…

Amazon.com: https://arxiv.org/pdf/1810.09558.pdf



Reinforcement Learning (RL)



State T

State T+1

Environment

Rewards

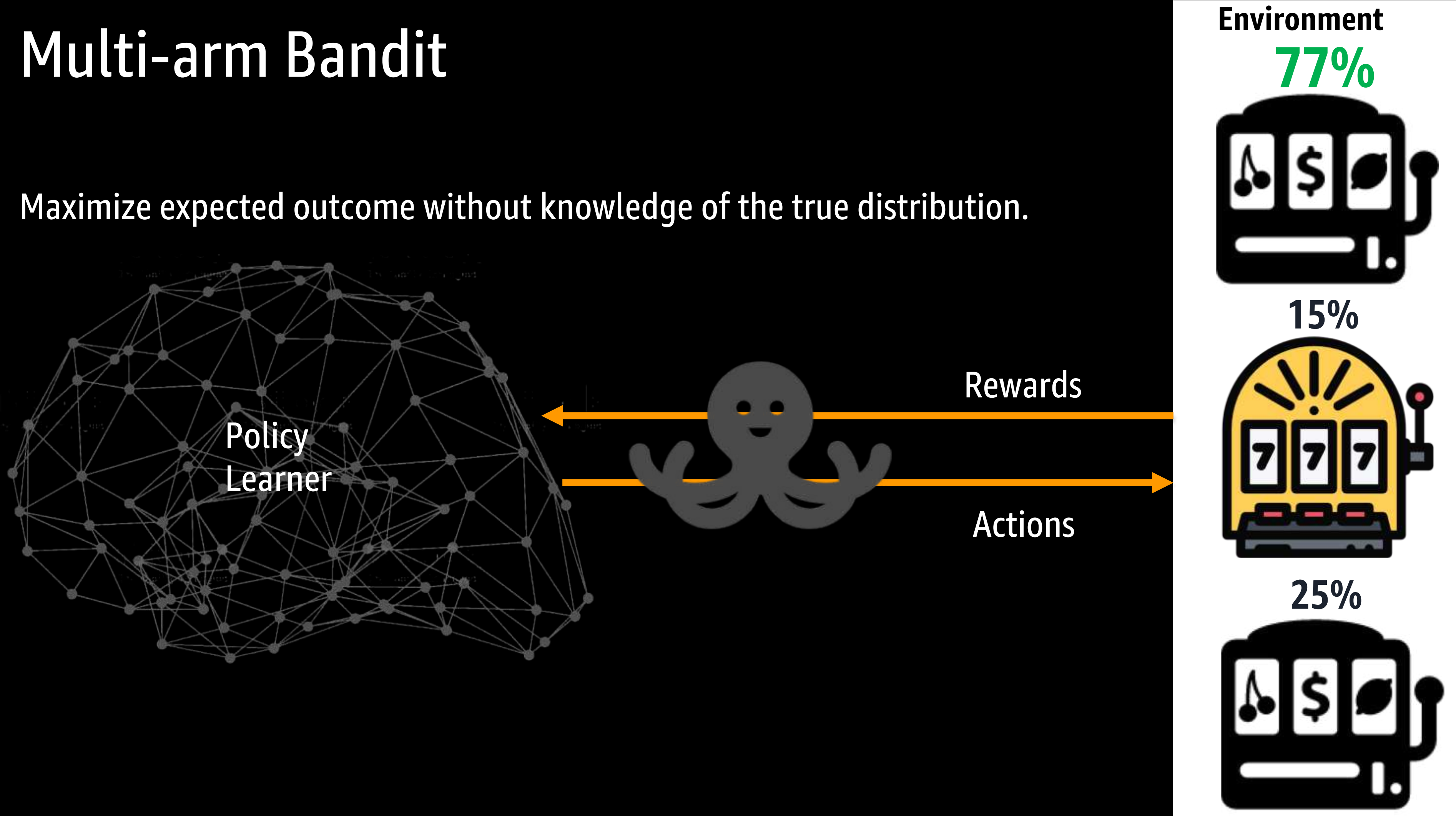
Policy Learner

Actions

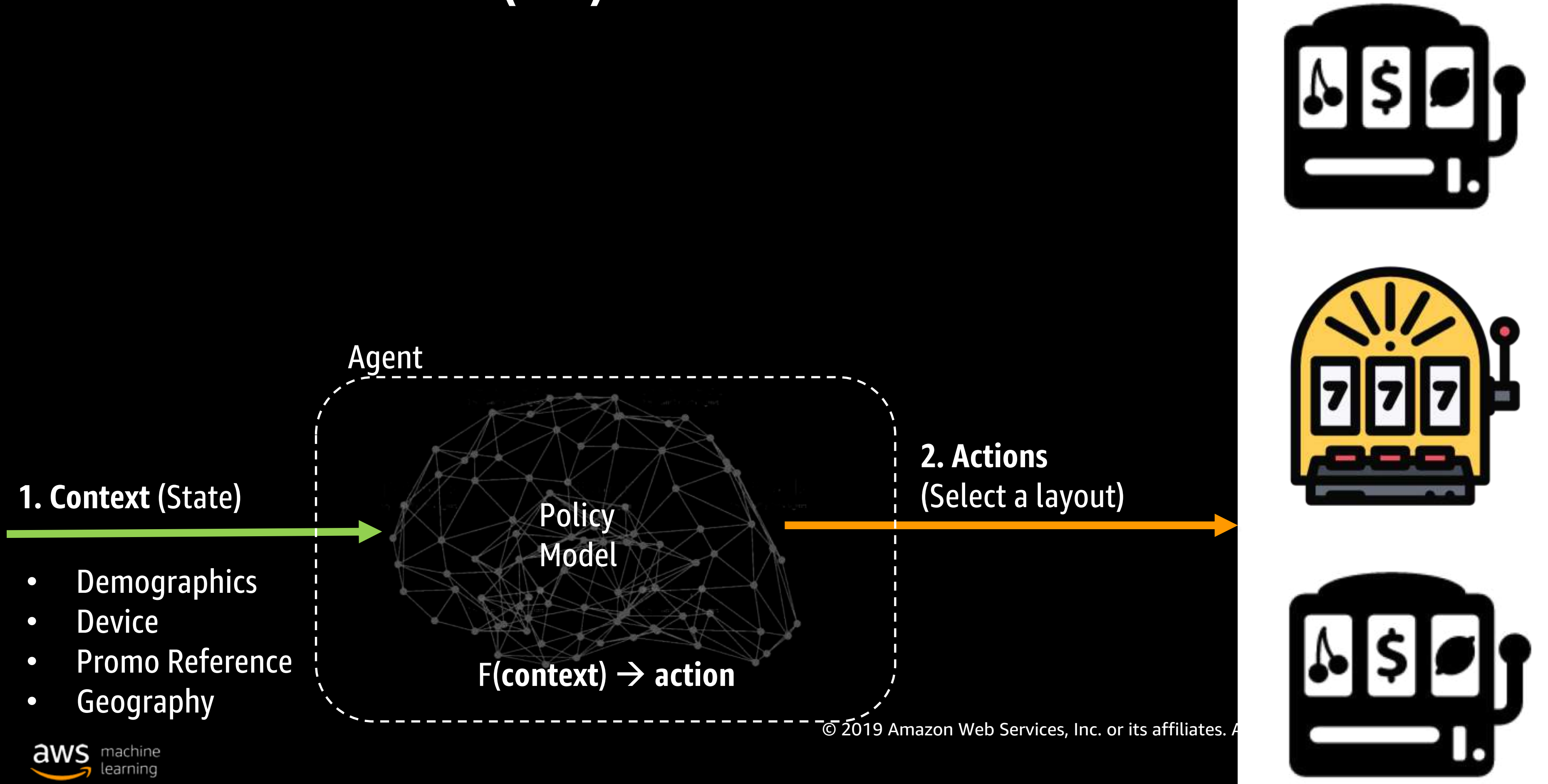
Agent

Multi-arm Bandit

Maximize expected outcome without knowledge of the true distribution.

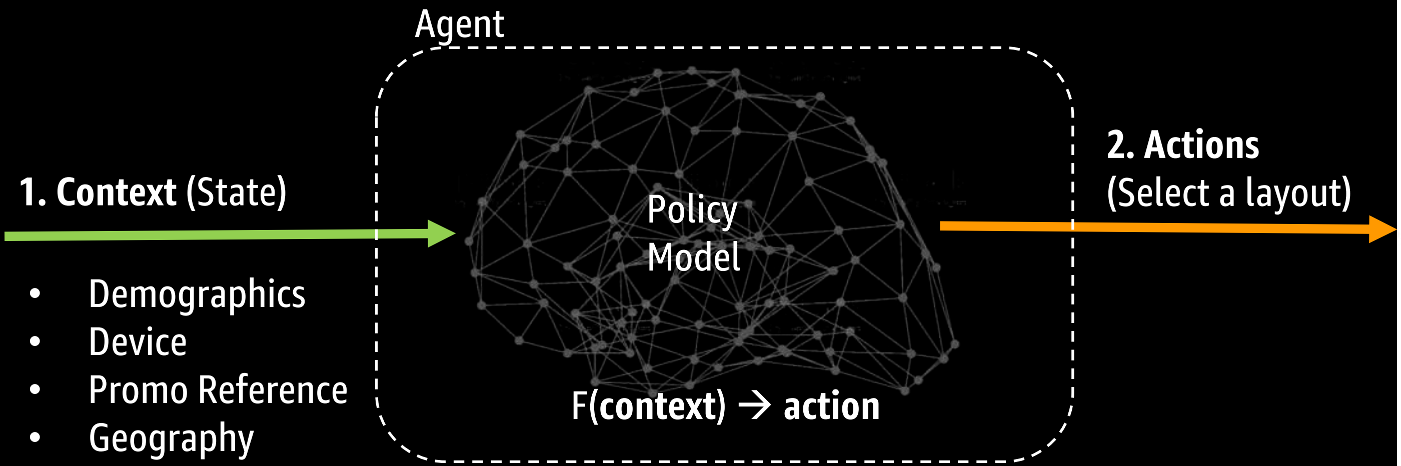


Contextual Bandits (CB)



- Demographics
- Device
- Promo Reference
- Geography

Contextual Bandits: Multivariate Testing

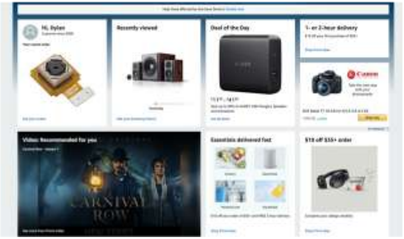


1. Context (State)

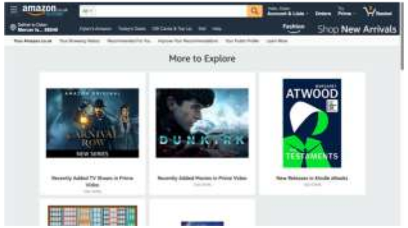
- Demographics
- Device
- Promo Reference
- Geography

2. Actions
(Select a layout)

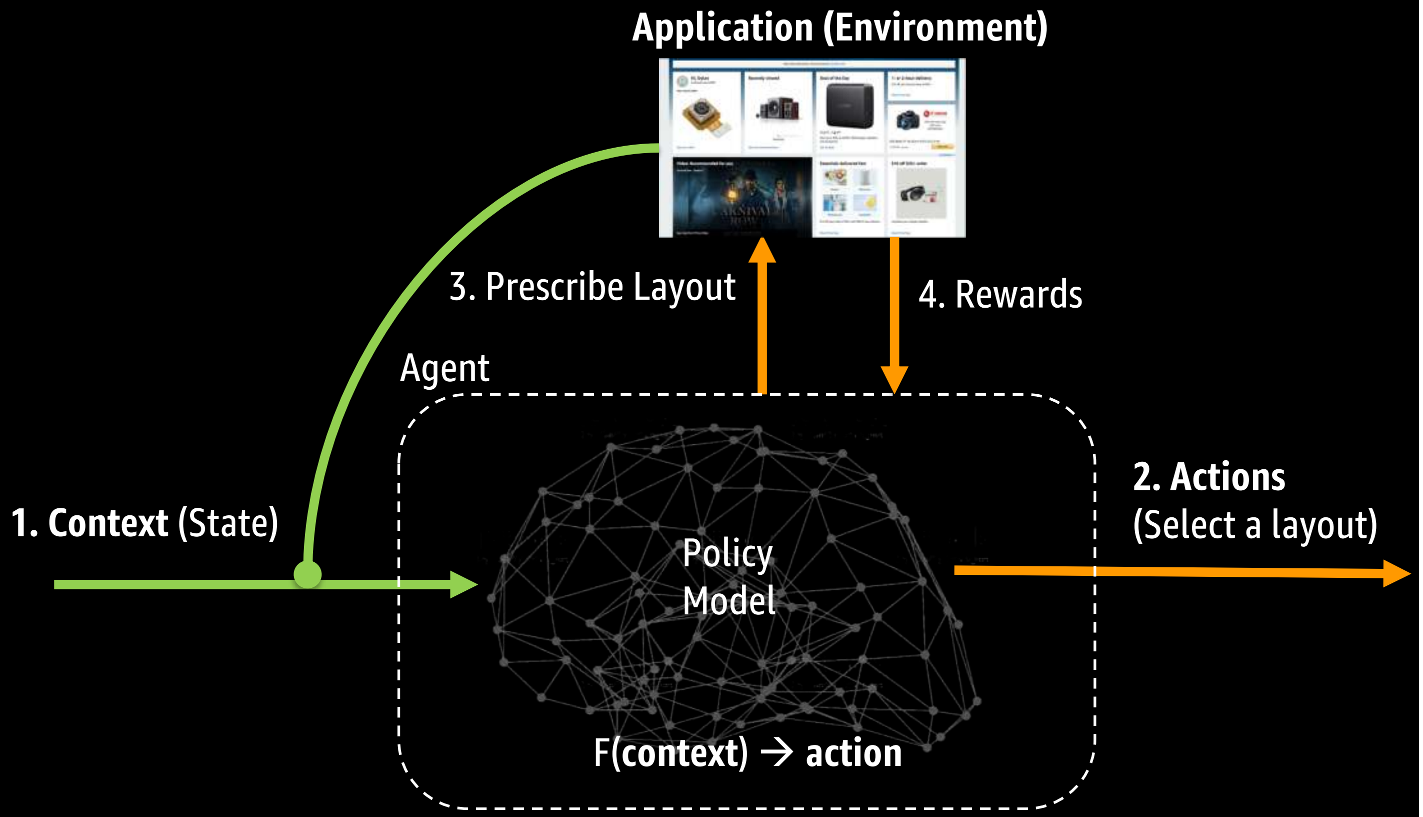
Arms = Layout Variations



•
•
•



Contextual Bandits and RL



Application (Environment)

3. Prescribe Layout

4. Rewards

Agent

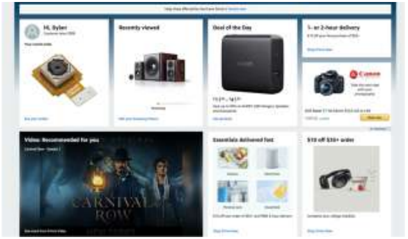
1. **Context** (State)

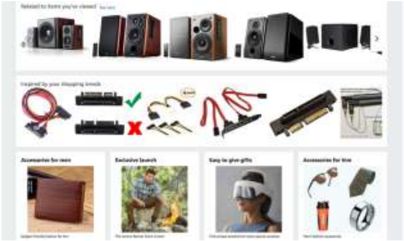
Policy Model

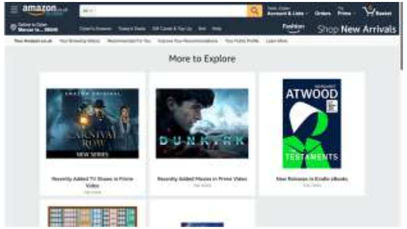
F(**context**) → **action**

2. Actions
(Select a layout)

Arms = Layout Variations







THE AWS ML STACK

Broadest and deepest set of capabilities

AI Services

VISION	SPEECH	LANGUAGE	CHATBOTS	FORECASTING	RECOMMENDATIONS
REKOGNITION IMAGE REKOGNITION VIDEO TEXTRACT	POLLY TRANSCRIBE	TRANSLATE COMPREHEND & COMPREHEND MEDICAL	LEX	FORECAST	PERSONALIZE

ML Services

Amazon SageMaker	Ground Truth	Notebooks	Algorithms + Marketplace	Reinforcement Learning	Training	Optimization	Deployment	Hosting

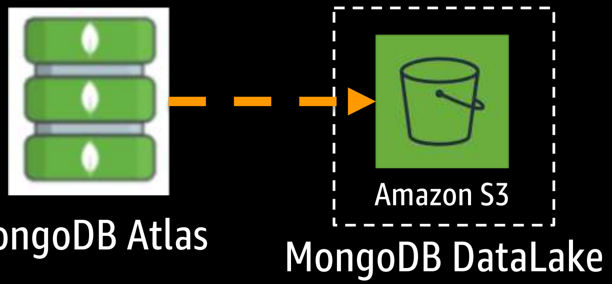
ML Frameworks + Infrastructure

FRAMEWORKS	INTERFACES
TensorFlow mxnet PYTORCH	GLUON Keras

INFRASTRUCTURE								
EC2 P3 & P3DN	EC2 G4 EC2 C5	FPGAS	DL CONTAINERS & AMIs	ELASTIC CONTAINER SERVICE	ELASTIC KUBERNETES SERVICE	GREENGRASS	ELASTIC INFERENCE	INFERENTIA

Training Initial Model: Warm Starts (…*if data exists*)

1. Experience data is prep and made available in the data lake.



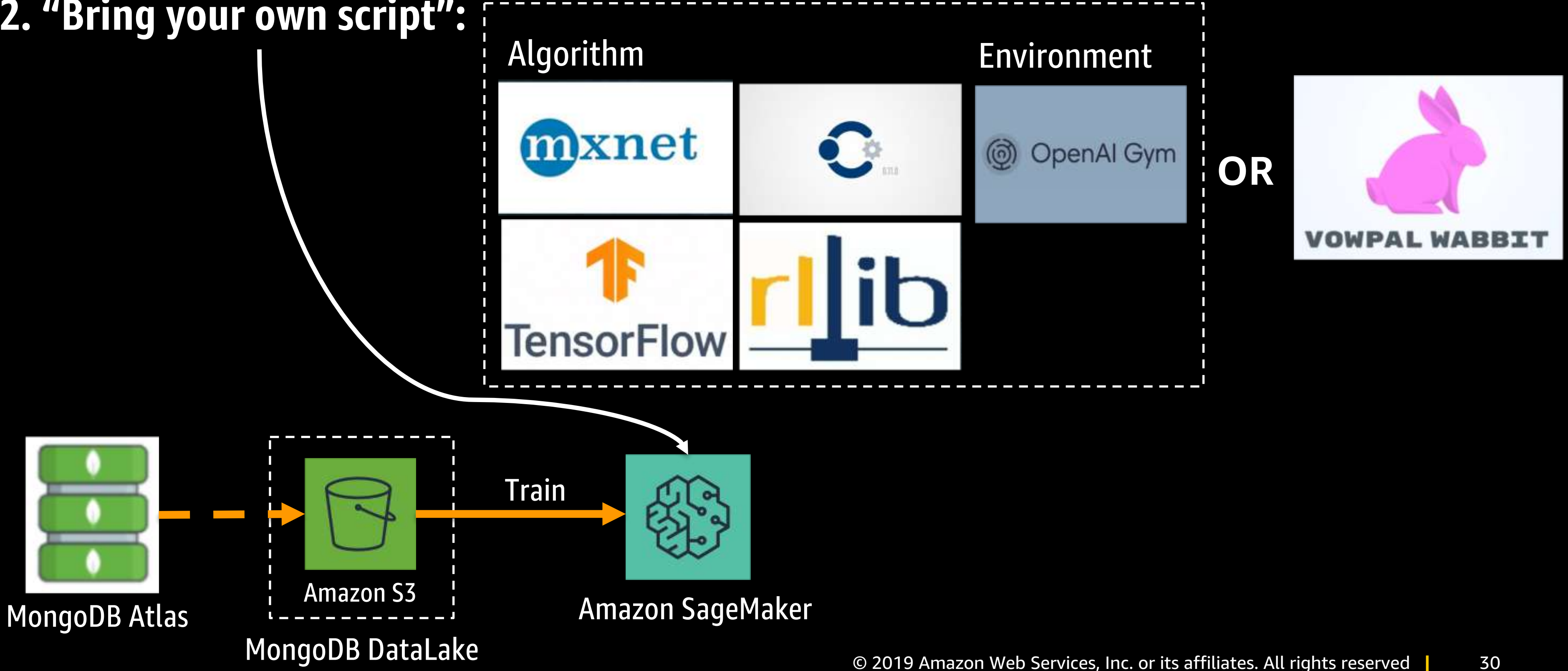
Experience Data:

Source: web and application logs:

- **Context features (state):** eg. device, geo, promo referrer…etc.
- **Action:** One of N layout variations
- **Action Probability:** chance that action is prescribed given the context for unbiasing the data.
- **Reward/Cost:** Selected value for a positive outcome. For instance, +1 for a click.

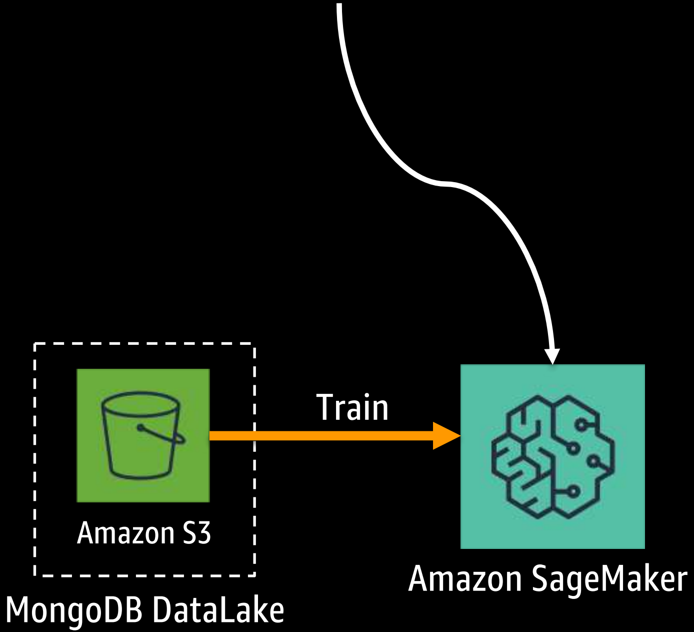
Amazon SageMaker Training: “BYOS” Approach

2. “Bring your own script”:



Bring Your Own Script for Vowpal Wabbit

2. “Bring your own script”:



Train

Amazon S3

MongoDB DataLake

Amazon SageMaker

[Vowpal Wabbit Contextual Bandits]()

Usage: ./vw -d train.dat **--cb_explore 10** --epsilon 0.1

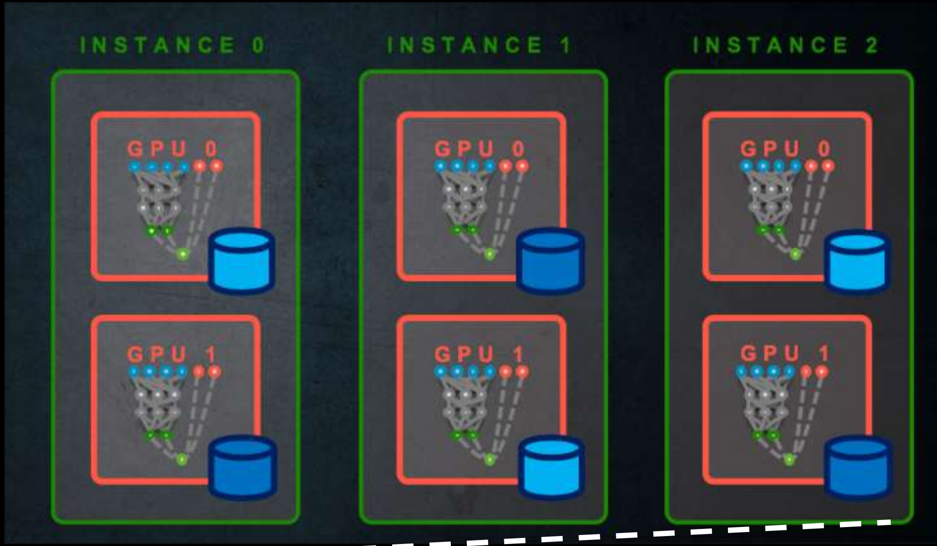
[Amazon SageMaker Examples:]()

[VW Python Scripts (CLI wrapper)]()

Amazon SageMaker Training

3. **Launch Training Jobs:** SageMaker provisions a cluster and runs the training job—only pay for what you use.

```
estimator = RLEstimator(entry_point="train-vw.py",
                        source_dir='src',
                        dependencies=["common/sagemaker_rl"],
                        image_name=custom_image_name,
                        role=role,
                        train_instance_type=instance_type,
                        train_instance_count=1,
                        output_path=s3_output_path,
                        base_job_name=job_name_prefix,
                        hyperparameters = {…}
                    )
```



```
estimator.fit(…)
```

Amazon SageMaker Hosting

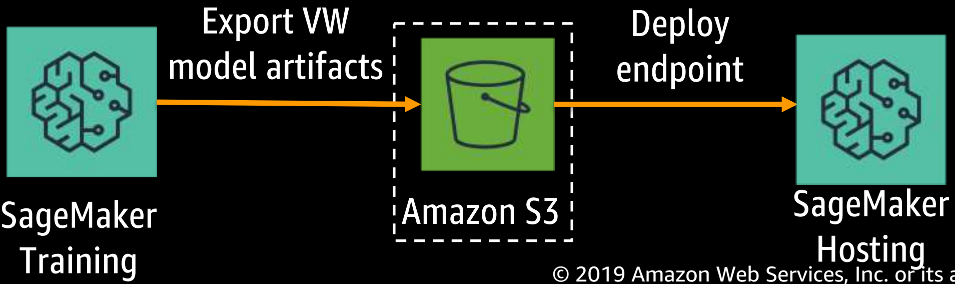
4. Deploy the model for real-time inference

I. Register model:

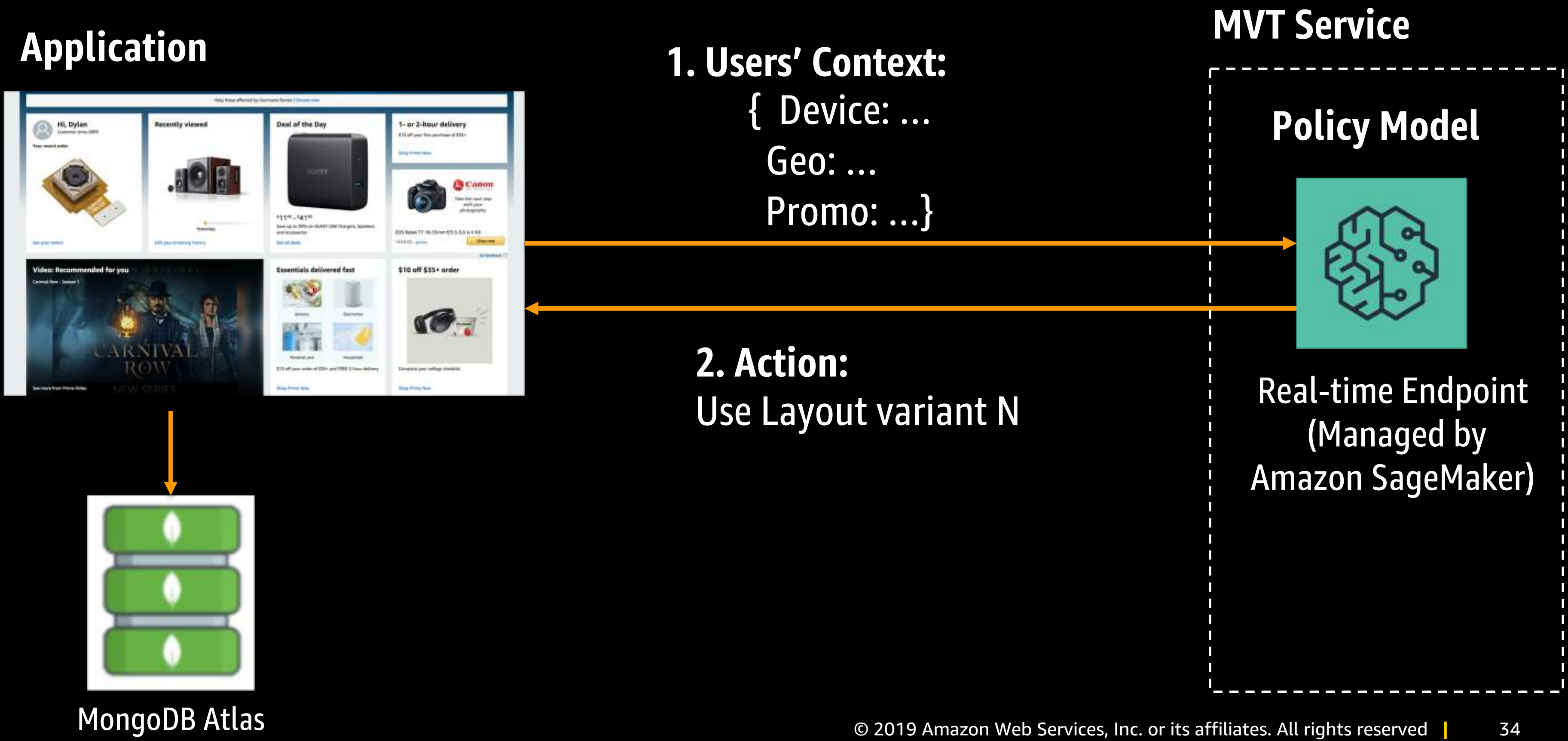
```
sagemaker_model = sagemaker.model.Model(
        image=self.image,
        role=self.resource_manager.iam_role_arn,
        name=model_id,
        model_data=model_record["s3_model_output_path"],
        sagemaker_session=self.sagemaker_session,
        env=environ_vars)
```

II. Deploy endpoint:

```
sagemaker_model.deploy(
        initial_instance_count=hosting_instance_count,
        instance_type=hosting_instance_type,
        endpoint_name=self.experiment_id)
```



Multivariate Testing in Production



Application

MVT Service

1. Users' Context:
{ Device: …
Geo: …
Promo: …}

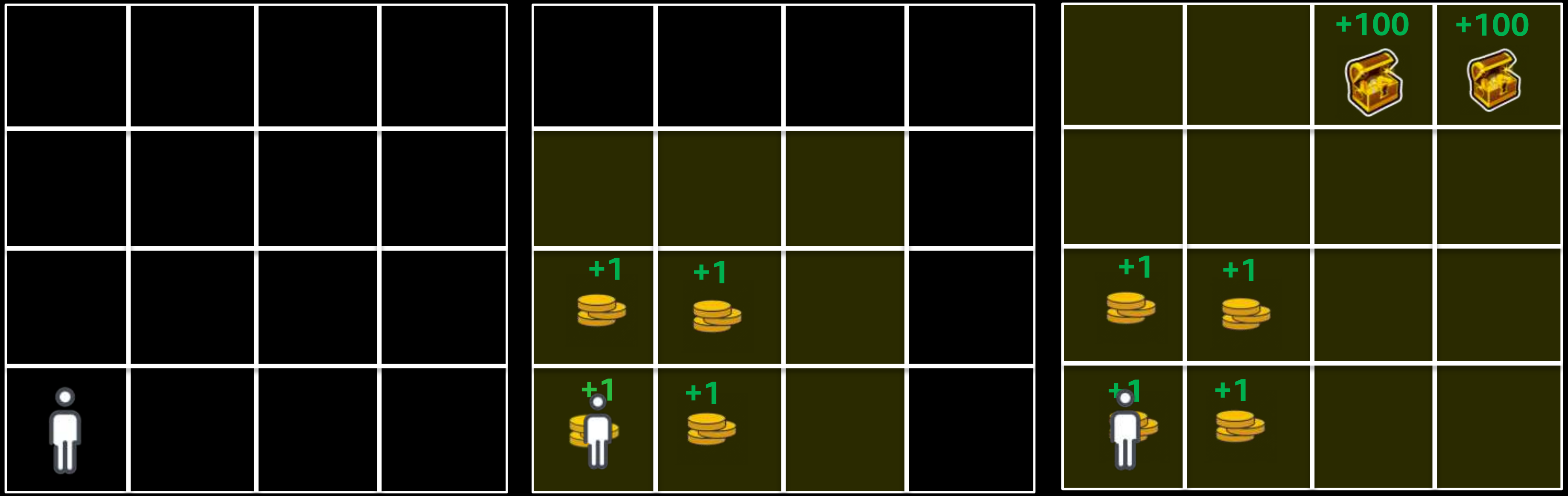
2. Action:
Use Layout variant N

Policy Model

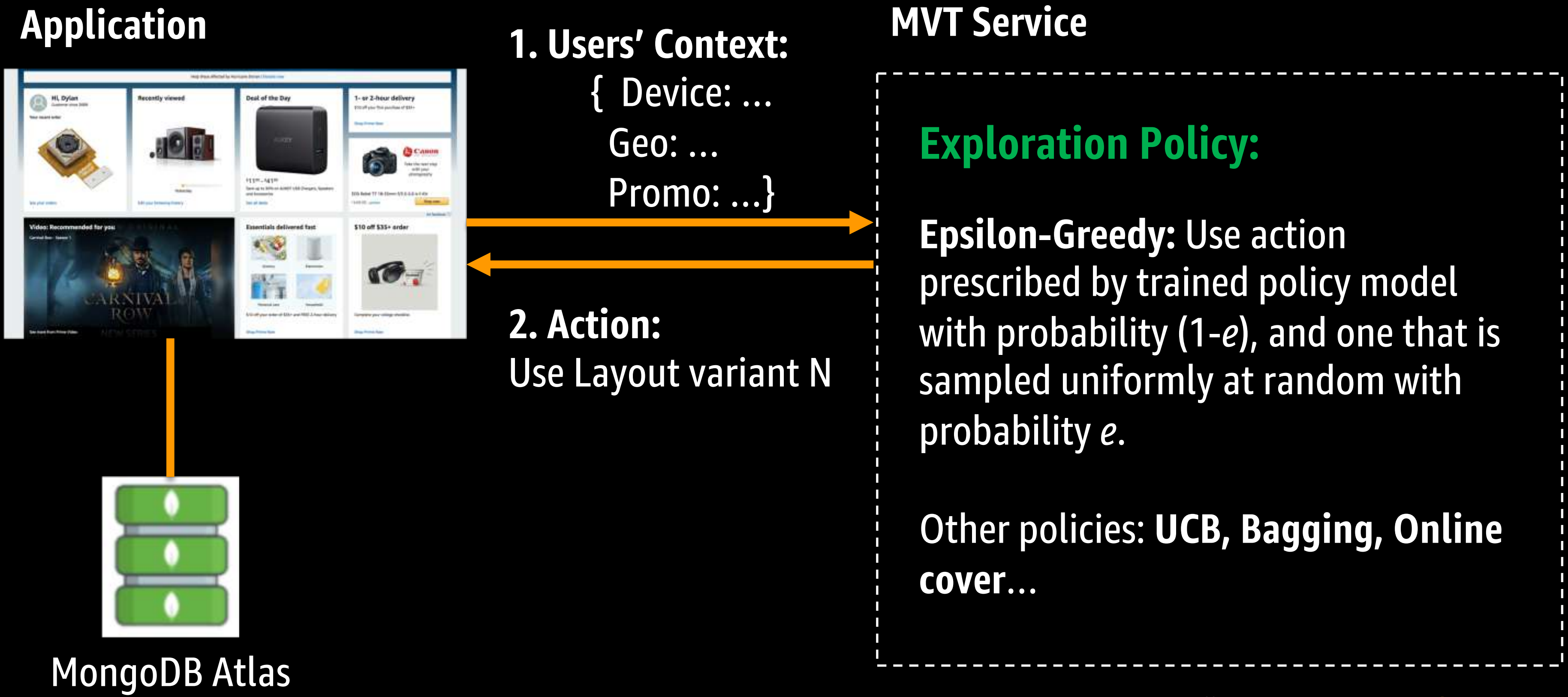
Real-time Endpoint
(Managed by
Amazon SageMaker)

MongoDB Atlas

Exploration and Exploitation



Exploration Policy



Application

1. Users' Context:
{ Device: …
Geo: …
Promo: …}

2. Action:
Use Layout variant N

MongoDB Atlas

MVT Service

Exploration Policy:

Epsilon-Greedy: Use action prescribed by trained policy model with probability (1-*e*), and one that is sampled uniformly at random with probability *e*.

Other policies: **UCB, Bagging, Online cover…**

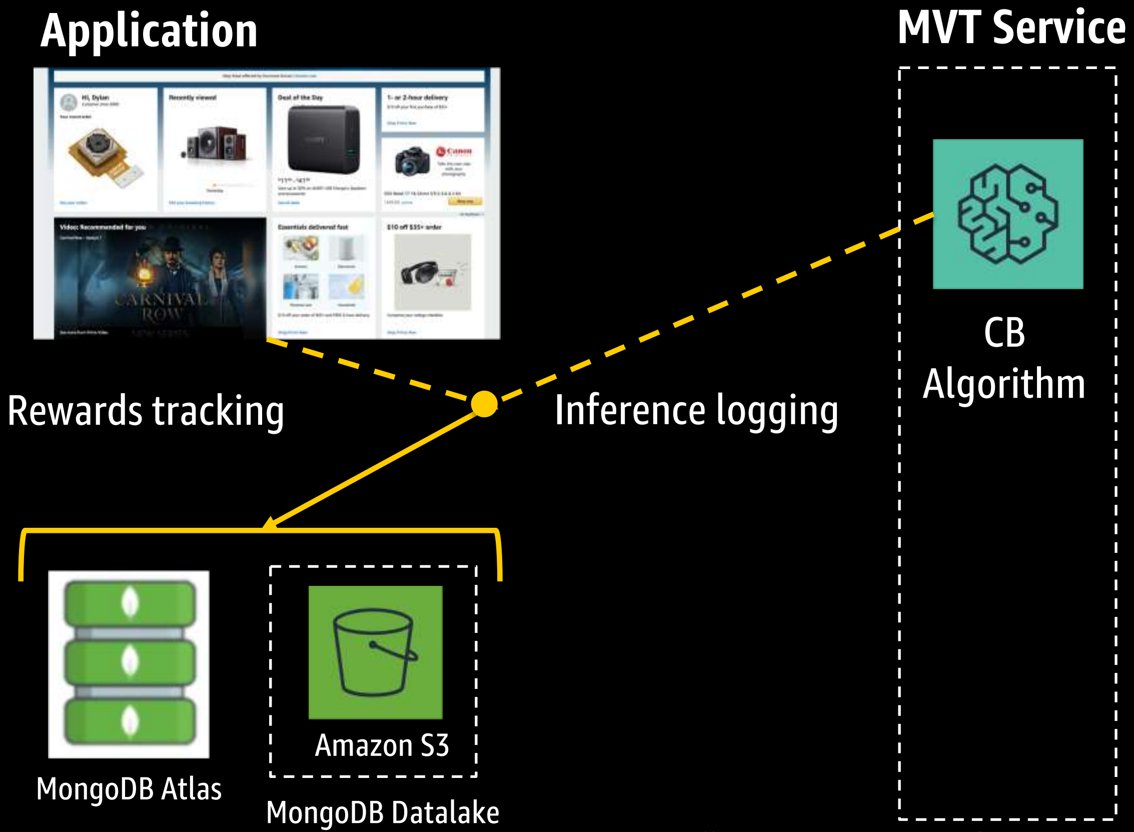
Inference and Experience Capture

3. Capture Experiences:

I. **Reward Tracking:** Event Id, Reward/Cost

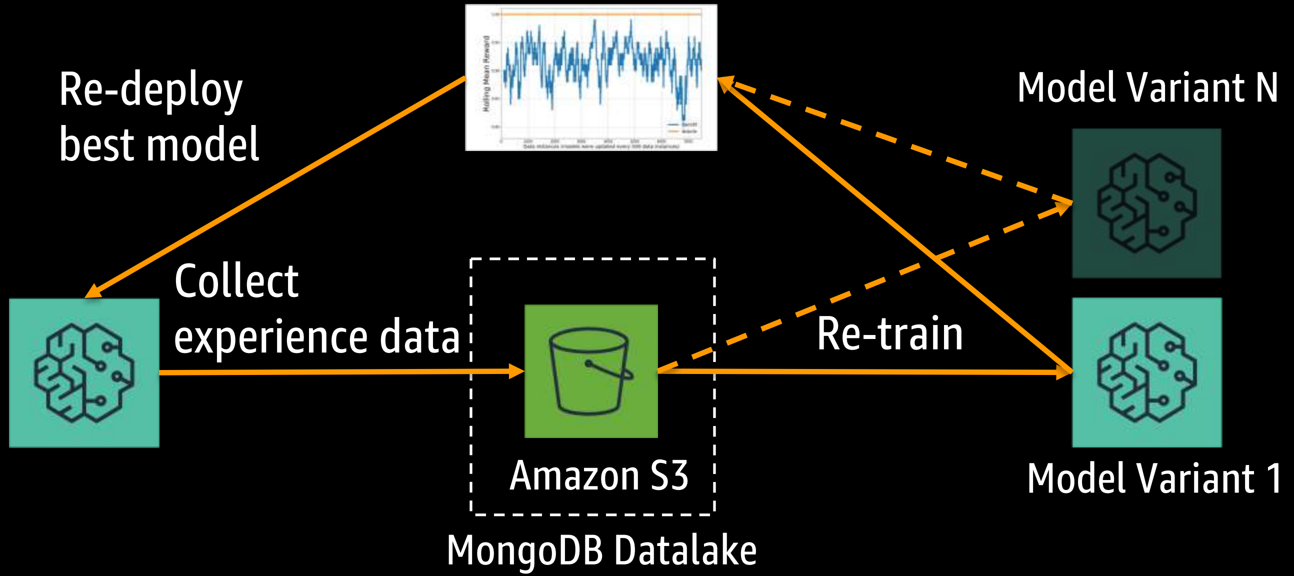
II. **Inference Logging:** Event Id, Context, Action, Action Probability

III. **Associate** rewards with inference events to augment the training set (experience data).



Re-train, Evaluate and Re-deploy

Offline Evaluation (Replay): Compare new and old models with varying policies



Re-deploy best model

Collect experience data

Amazon S3

MongoDB Datalake

Re-train

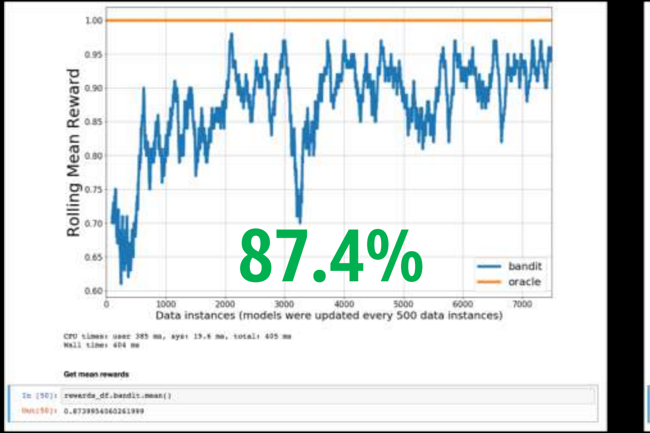
Model Variant N

Model Variant 1

Minimize cumulative "regret" (opportunity cost)

$$\sum_{t=1}^{T} r_t(\pi(x_t)) - r_t(a_t)$$

Converges Towards an Optimal Policy

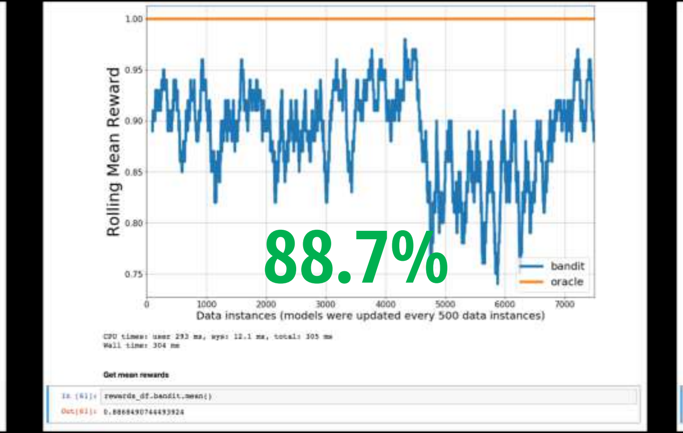


87.4%

```
CPU times: user 385 ms, sys: 19.6 ms, total: 405 ms
Wall time: 404 ms
```

Get mean rewards

```
In [50]: rewards_df.bandit.mean()
Out[50]: 0.8739954060261999
```

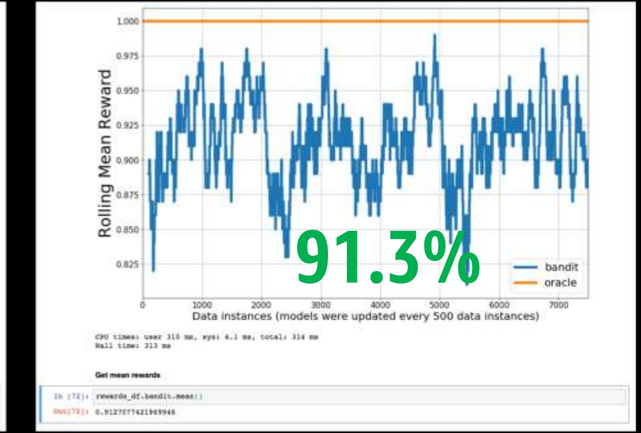


88.7%

```
CPU times: user 293 ms, sys: 12.1 ms, total: 305 ms
Wall time: 304 ms
```

Get mean rewards

```
In [61]: rewards_df.bandit.mean()
Out[61]: 0.8868490744493924
```

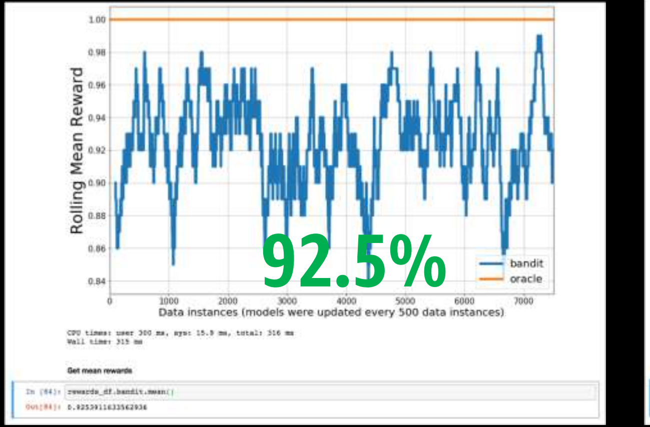


91.3%

```
CPU times: user 310 ms, sys: 4.1 ms, total: 314 ms
Wall time: 313 ms
```

Get mean rewards

```
In [72]: rewards_df.bandit.mean()
Out[72]: 0.9127077421969946
```

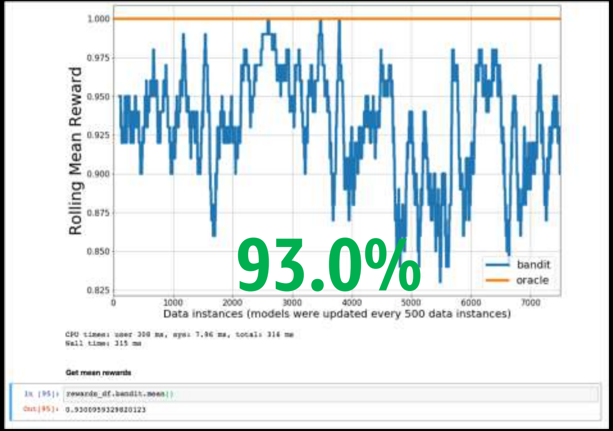


92.5%

```
CPU times: user 300 ms, sys: 15.9 ms, total: 316 ms
Wall time: 315 ms
```

Get mean rewards

```
In [84]: rewards_df.bandit.mean()
Out[84]: 0.9253911633562936
```



93.0%

```
CPU times: user 308 ms, sys: 7.96 ms, total: 316 ms
Wall time: 315 ms
```

Get mean rewards

```
In [95]: rewards_df.bandit.mean()
Out[95]: 0.9300959329820123
```

Sample SageMaker Notebook

• • •

Humanize and Personalize Conversational AI

THE AWS ML STACK

Broadest and deepest set of capabilities

AI Services

VISION			SPEECH		LANGUAGE		CHATBOTS	FORECASTING	RECOMMENDATIONS
REKOGNITION IMAGE	REKOGNITION VIDEO	TEXTRACT	POLLY	TRANSCRIBE	TRANSLATE	COMPREHEND & COMPREHEND MEDICAL	LEX	FORECAST	PERSONALIZE

ML Services

Amazon SageMaker

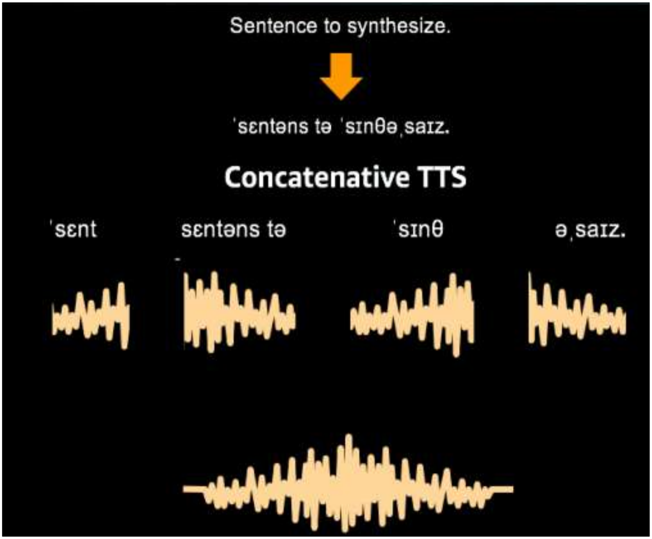
Ground Truth	Notebooks	Algorithms + Marketplace	Reinforcement Learning	Training	Optimization	Deployment	Hosting

ML Frameworks + Infrastructure

FRAMEWORKS	INTERFACES
TensorFlow mxnet PYTORCH	GLUON Keras

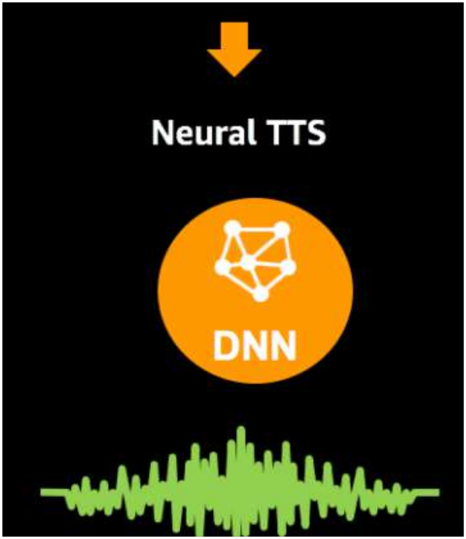
INFRASTRUCTURE								
EC2 P3 & P3DN	EC2 G4 EC2 C5	FPGAS	DL CONTAINERS & AMIs	ELASTIC CONTAINER SERVICE	ELASTIC KUBERNETES SERVICE	GREENGRASS	ELASTIC INFERENCE	INFERENTIA

Amazon Polly: Humanize Your Apps using Neural TTS



US English Joanna voice

"President Donald Trump said on March 13 his administration was ordering the grounding of all Max 8 and 9 models, hours after Canada said it was grounding the planes after analyzing new satellite tracking data."



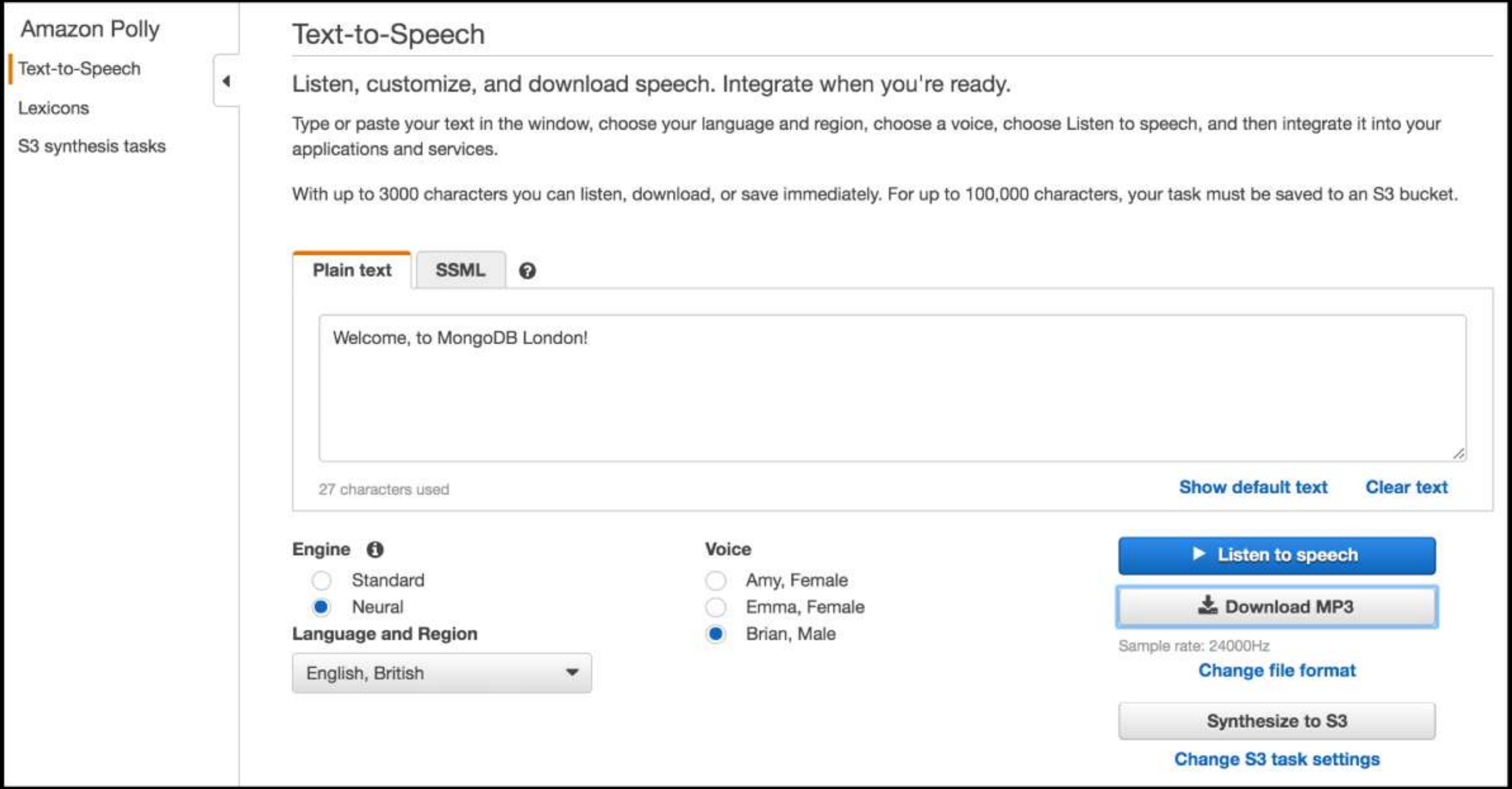
Standard

Newscaster NTTS

Amazon Polly: Personalize Your Voices

Justin
English (US)
Male, Child

Brian
English (UK)
Male, Adult



Amazon Polly

Text-to-Speech

Lexicons

S3 synthesis tasks

Text-to-Speech

Listen, customize, and download speech. Integrate when you're ready.

Type or paste your text in the window, choose your language and region, choose a voice, choose Listen to speech, and then integrate it into your applications and services.

With up to 3000 characters you can listen, download, or save immediately. For up to 100,000 characters, your task must be saved to an S3 bucket.

Welcome, to MongoDB London!

27 characters used

Engine
○ Standard
● Neural

Language and Region
English, British

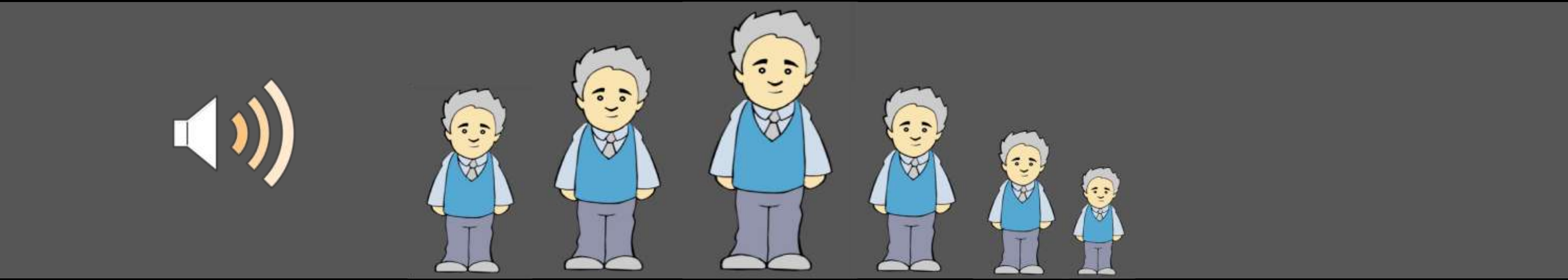
Voice
○ Amy, Female
○ Emma, Female
● Brian, Male

Sample rate: 24000Hz

aws machine learning

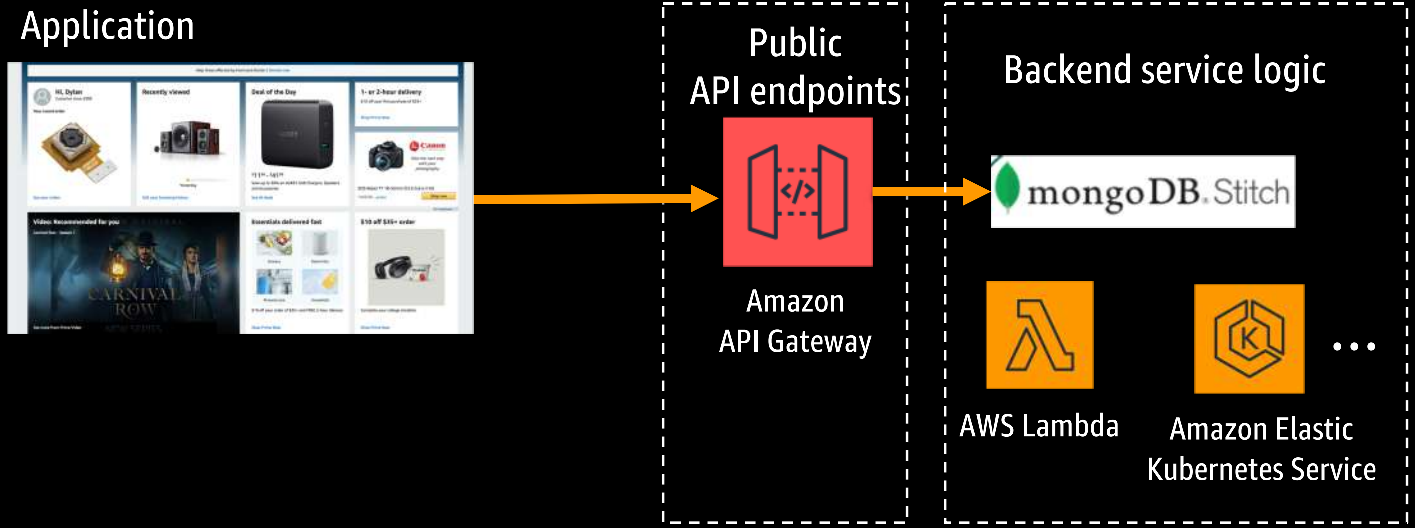
Voice Modification

```
<speak>
        This is Brian without any voice modifications.
        <amazon:effect vocal-tract-length="+15%"> Imagine now that I got bigger… </amazon:effect>
        <amazon:effect vocal-tract-length="+25%"> Suppose that I got even bigger still… </amazon:effect>
        Now let's go back and hear the effect when I go in the opposite direction.
        <amazon:effect vocal-tract-length="-15%"> Can you tell that I'm getting smaller? </amazon:effect>
        <amazon:effect vocal-tract-length="-25%"> Now I'm even smaller than before. </amazon:effect>
 </speak>
```



Deploy as a microservice

Your choice of compute...



Application

Public API endpoints

Amazon API Gateway

Backend service logic

AWS Lambda

Amazon Elastic Kubernetes Service

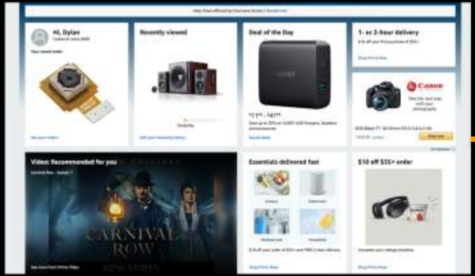
...

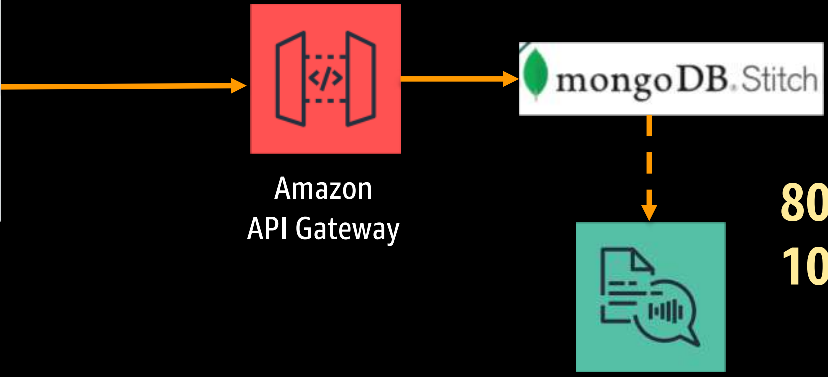
Know your performance requirements

```
Params =
{ "Engine": "string",
"LanguageCode": "string",
"LexiconNames": [ "string" ],
 "OutputFormat": "string",
"SampleRate": "string",
"SpeechMarkTypes": [ "string" ],
"Text": "string",
"TextType": "string",
"VoiceId": "string" }
```

→

```
Polly.synthesizeSpeech
      (params, (err, data) => {…}
```





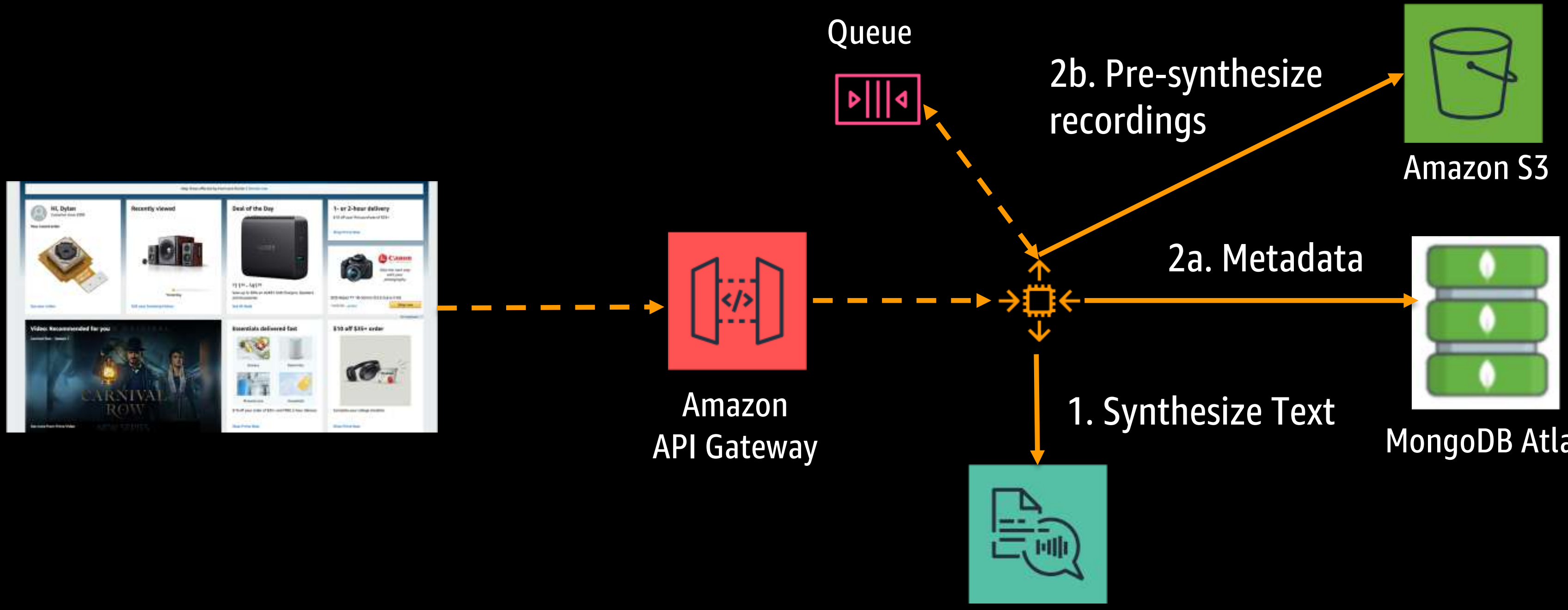
Amazon API Gateway

Amazon Polly

80 TPS

100 TPS (burst)

Build in Caching and Pre-processing



Queue

2b. Pre-synthesize recordings

Amazon S3

2a. Metadata

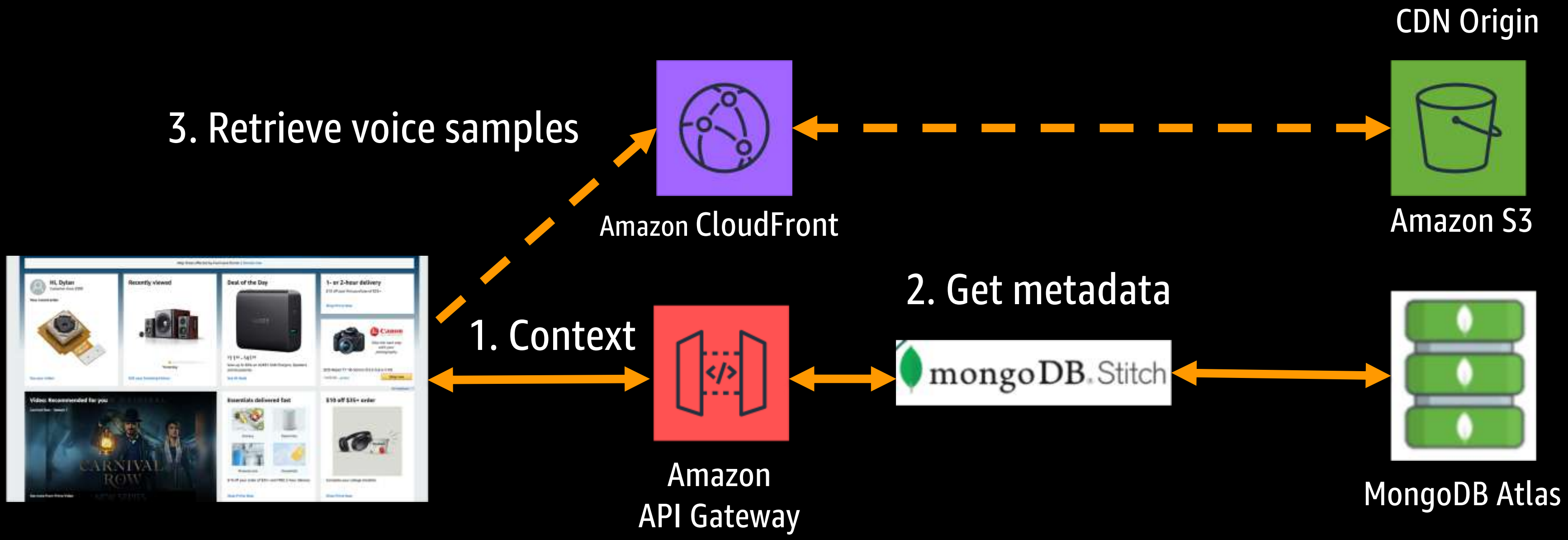
Amazon API Gateway

1. Synthesize Text

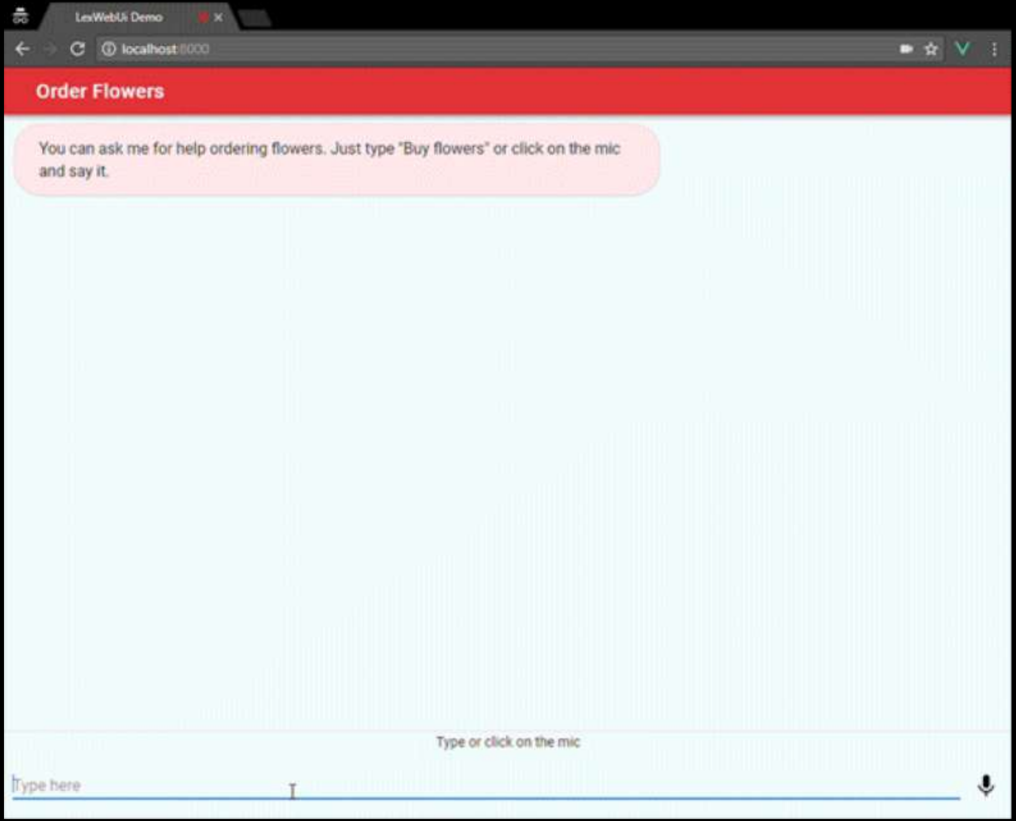
MongoDB Atlas

Amazon Polly

Cache and stream from the edge



Branding Voice: chat bots with personality



You can ask me for help ordering flowers. Just type "Buy flowers" or click on the mic and say it.

Type or click on the mic

Type here

Github: Lex Chatbot Example

MONGODB.LOCAL

Using AWS to Transform Customer Data in MongoDB into AI-driven Personalization

Dylan Tong + Igor Alekseev

https://www.surveymonkey.com/r/8L63FGV

#MDBlocal