Serverless Architecture 101

Ellipsis Tech Series 2020 Hackathon
Pre-Hackathon Workshop

Glendon Thaiw Yong Neng

3 September 2020

Introduction

Glendon Thaiw Yong Neng

Solutions Architect Intern, ASEAN Digital Customers & Region Expansion at AWS

Renaissance Engineering Programme, Nanyang Technological University

: Glendonthaiw

: Glendont

Workshop Overview

- Serverless 101
- Case Study – Realtor.com
- Hands-on Project with AWS
- Q&A

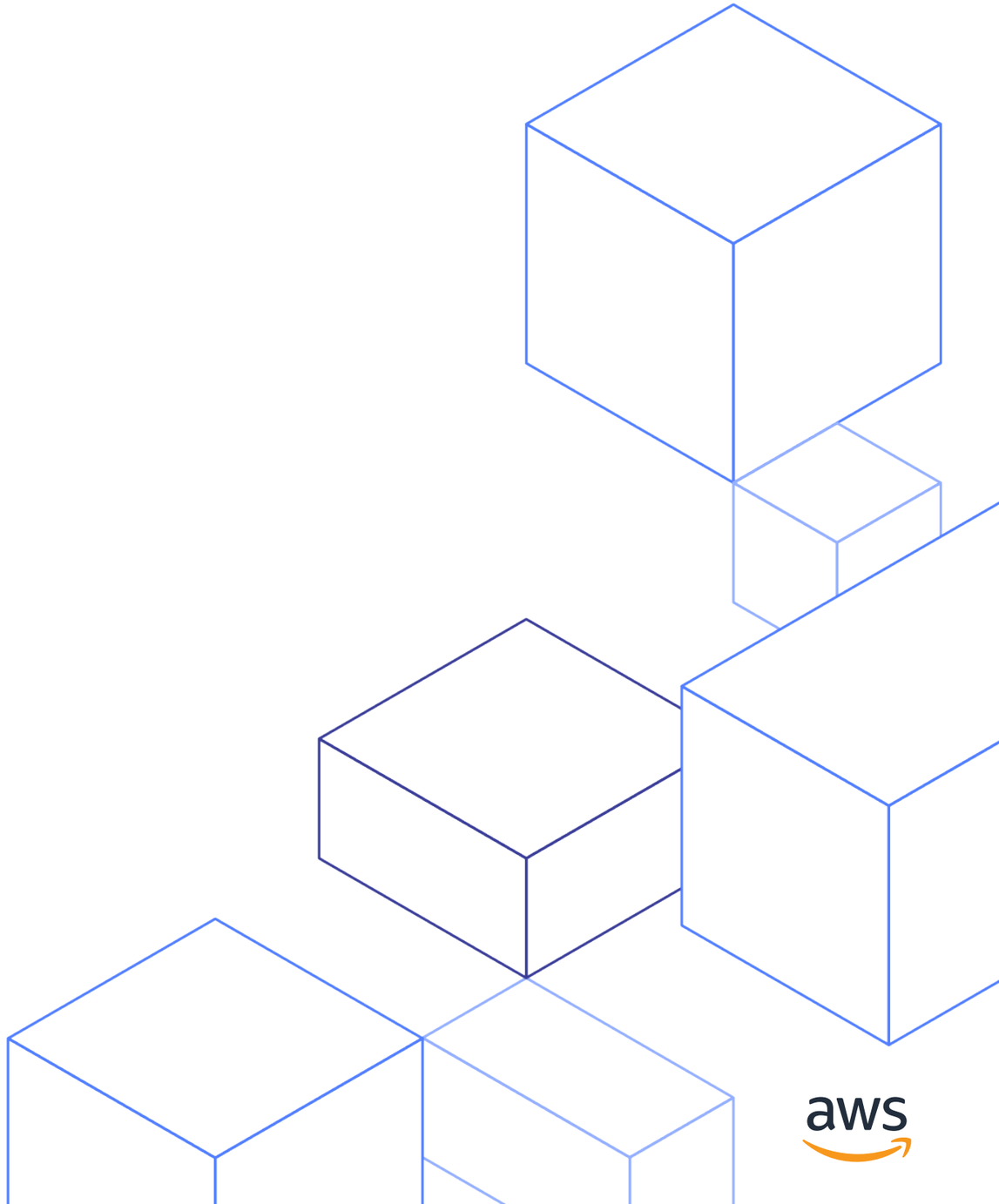
Resources (code snippets and workshop slide deck) available at:

https://github.com/glendont/aws-serverless-workshop

*** Please direct your technical issues to Okkar Min through direct message*

aws

Serverless 101



What is serverless?

rackspace

Data Centers

In 2006,
EC2 is introduced

Amazon EC2

Infrastructure
As a service

In 2011,
EB is introduced

Elastic BeanStalk

Platform
As a service

In 2014,
Lambda is introduced

AWS Lambda

Serverless
"Function-as-a-service"

aws

What is serverless?

Data Centers

Coffee-machine-as-a-Service

Infrastructure
As a service

Platform
As a service

Coffee-as-a-service

Serverless
“Function-as-a-service”

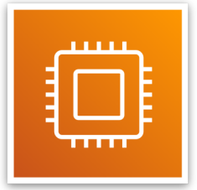
Why use serverless?

No infrastructure provisioning, no server management

Automatic, Flexible Scaling

Pay for Value

Highly available and secure



aws

AWS Serverless Services

Amazon API Gateway

Amazon Lambda

Amazon Simple Notification Service (SNS)

Amazon Simple Queue Service (SQS)

Amazon Fargate

Amazon EventBridge

Amazon Step Functions

Amazon Serverless Application Model (SAM)

aws

Three-tier application architecture

Web servers

Presentation

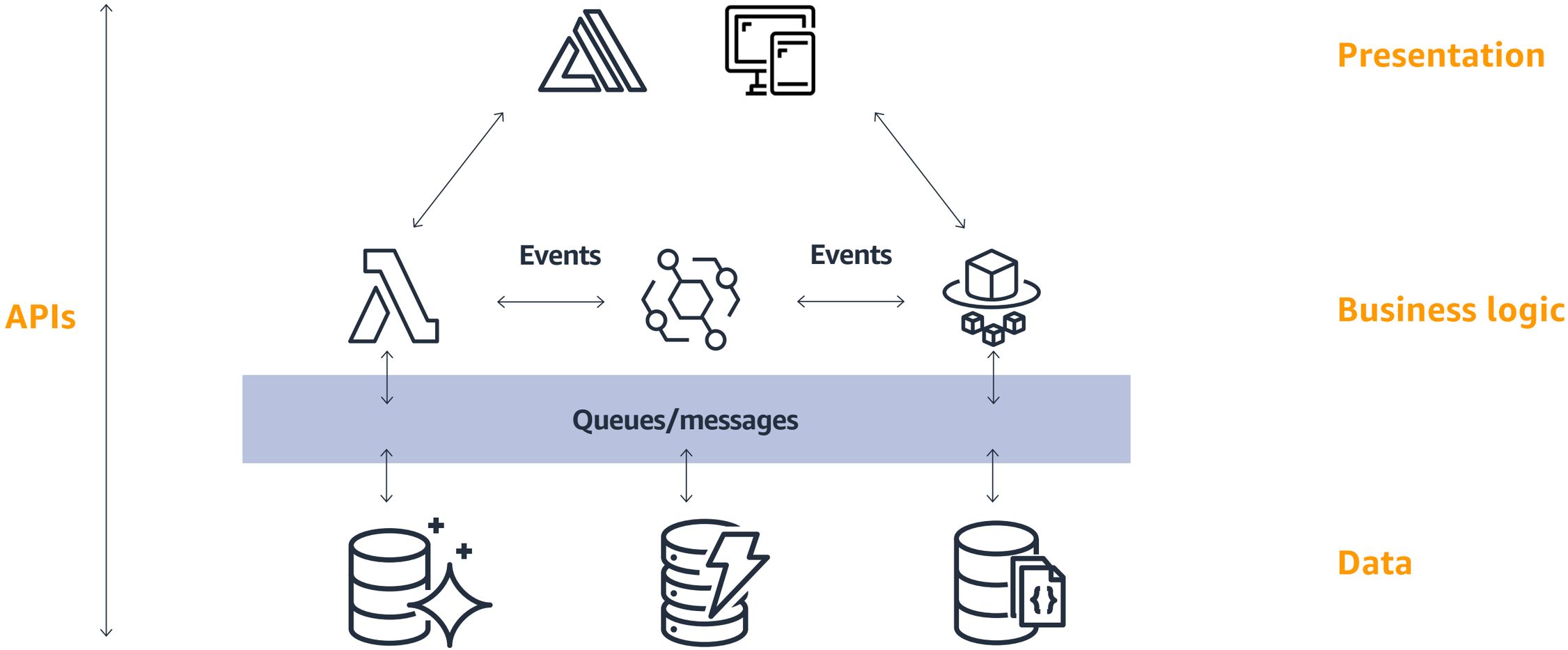
Application servers

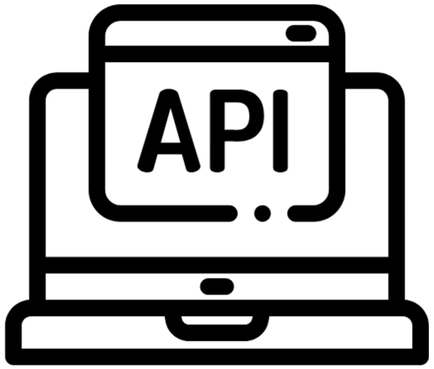
Business

Database servers

Data layer

A modern three-tier web application architecture

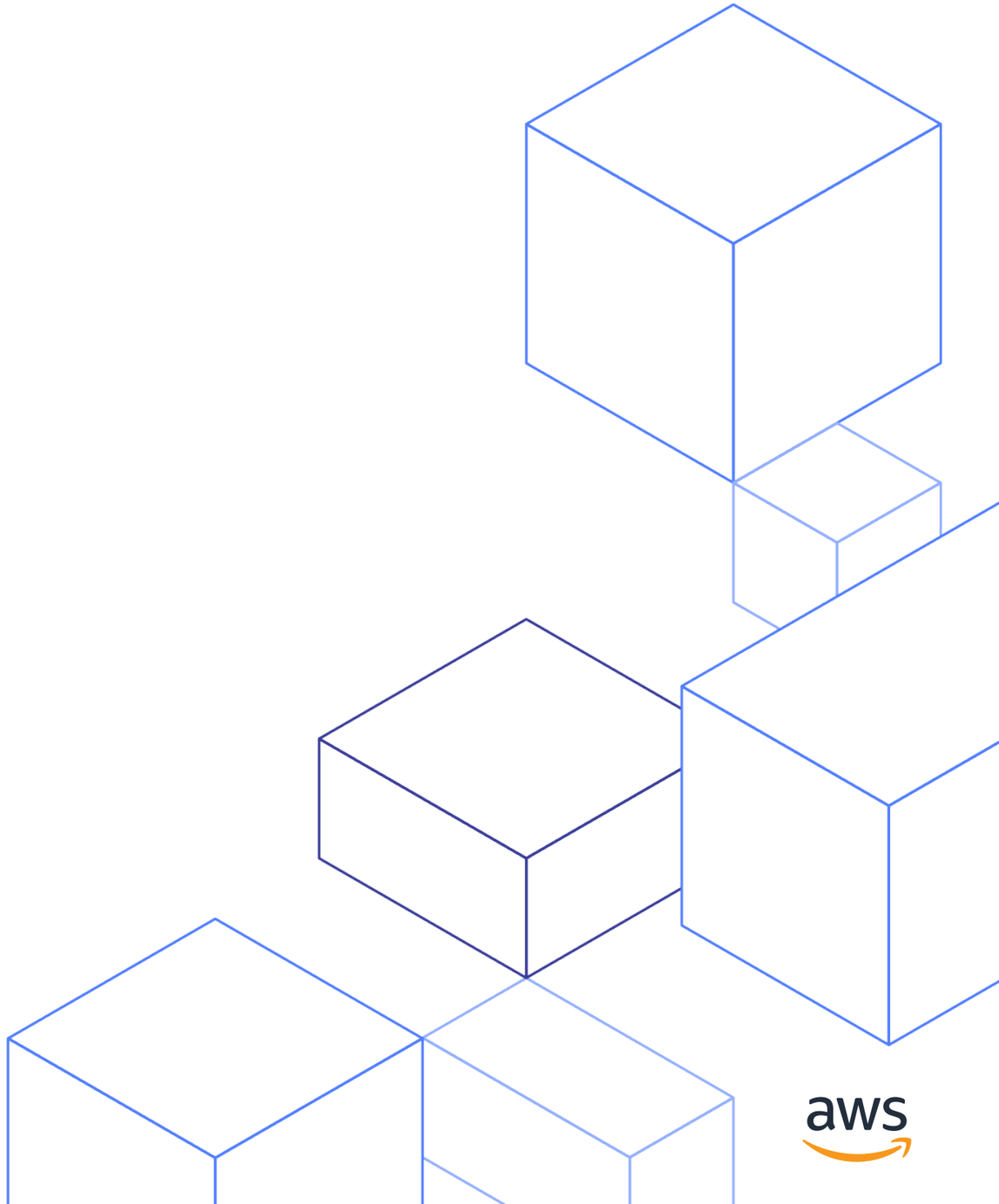




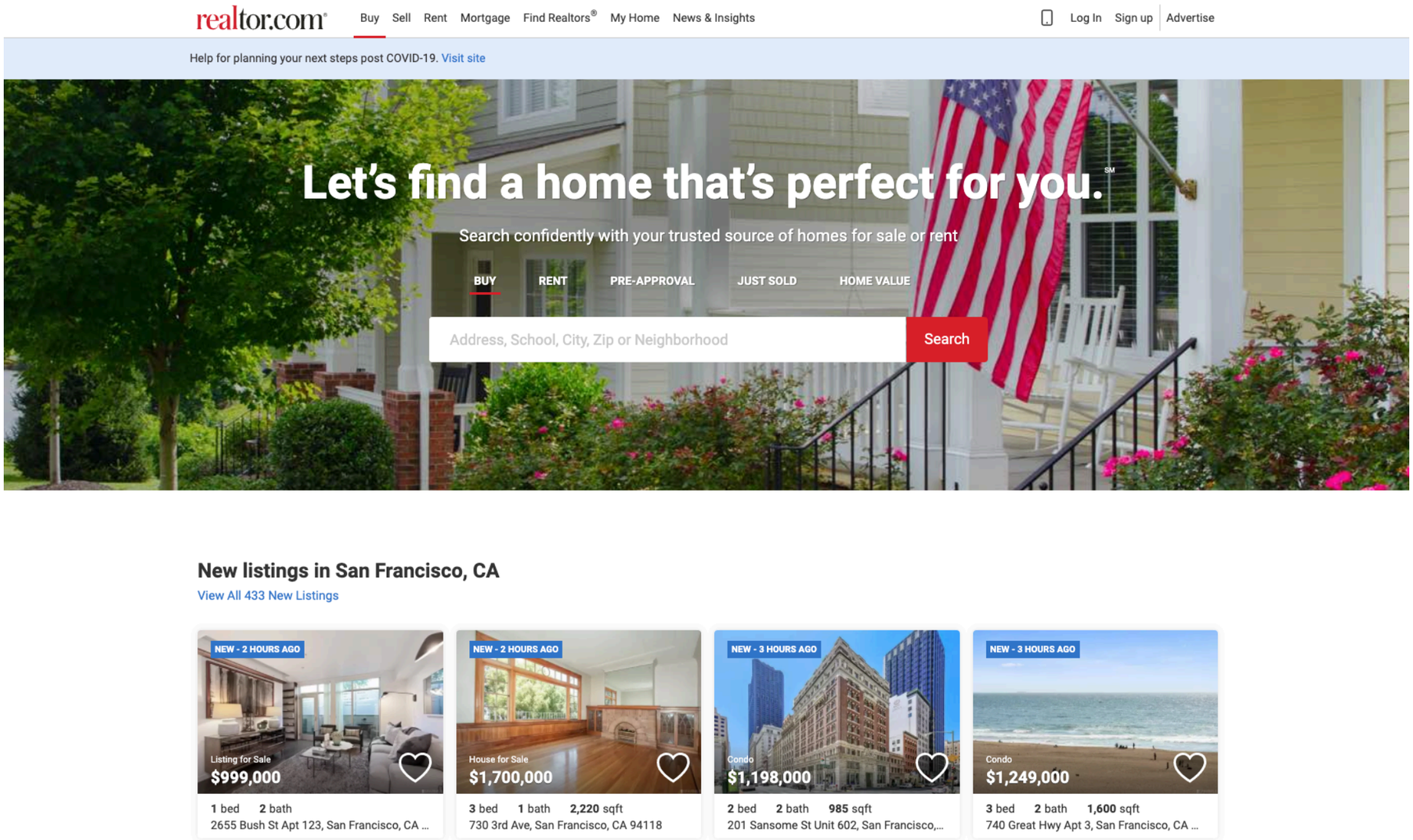
APIs are the front door of microservices

Case Study

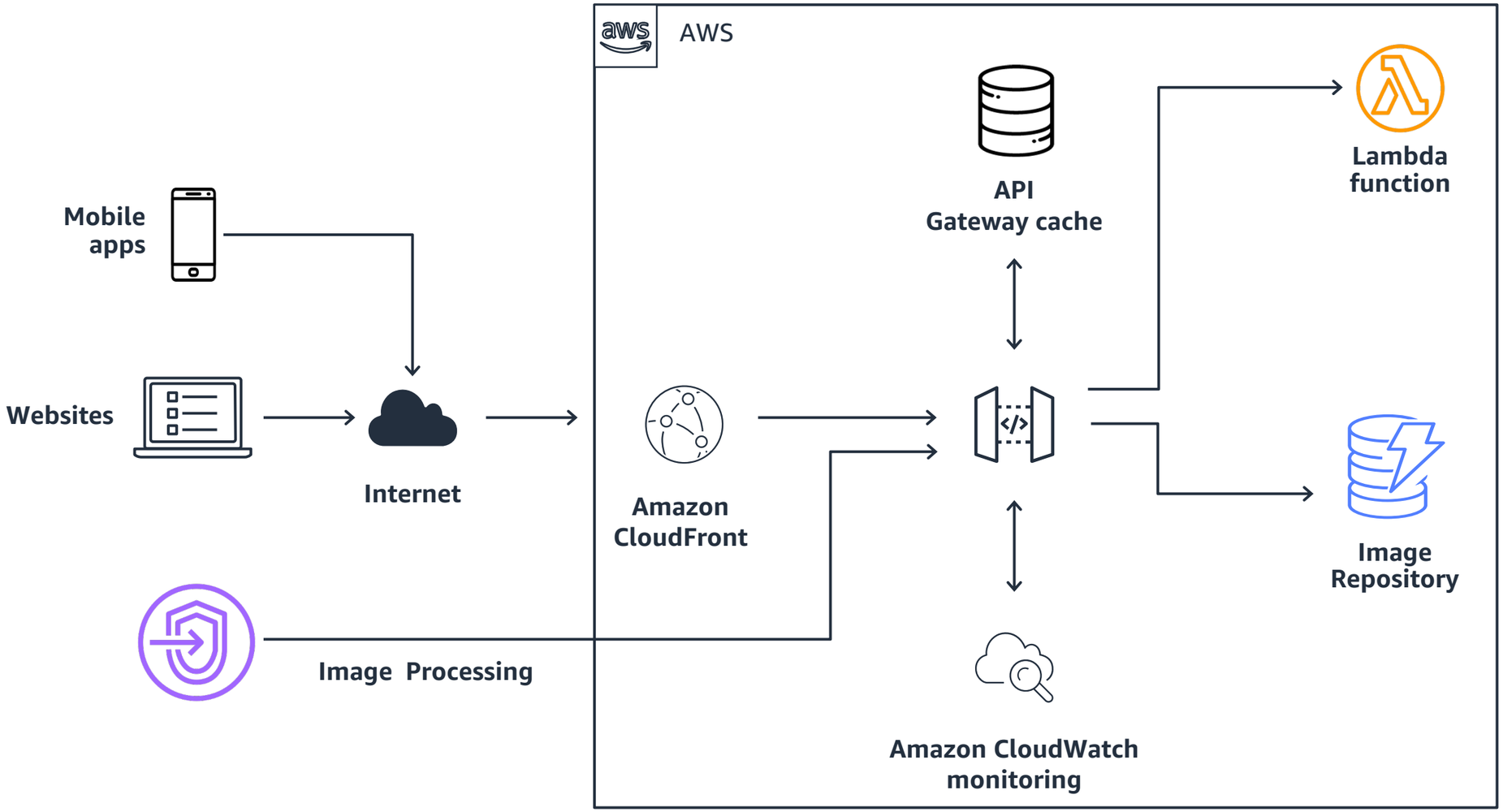
Realtor.com



Case Study: Realtor.com uses APIs between services



Case Study: Realtor.com uses APIs between services

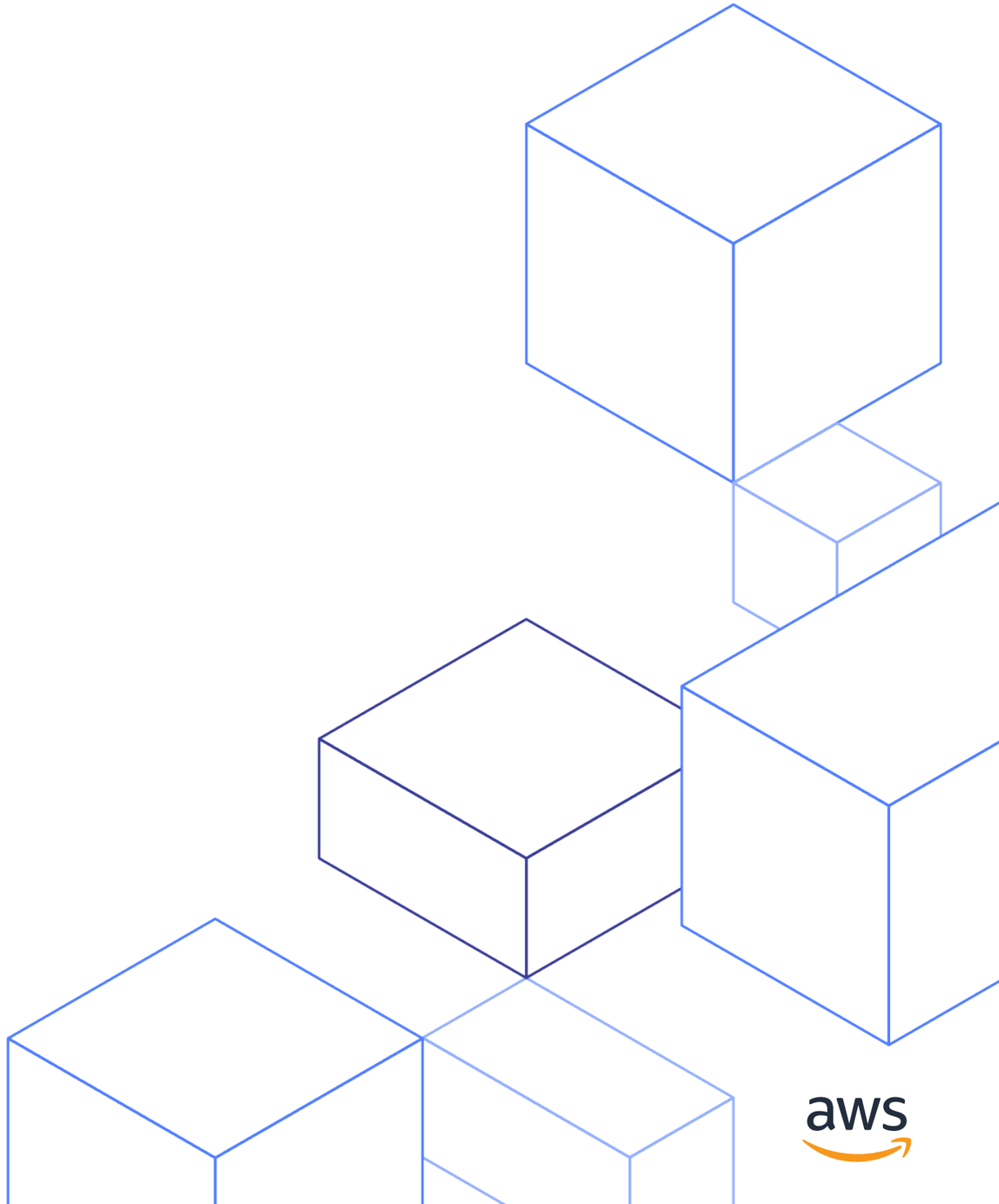


“We process 800 million images per day through Amazon API Gateway...”

Kuntal Shah,
SVP Engineering,
Realtor.com

realtor.com®

Hands-on project with AWS



Serverless Web Application

WildRydes

- Ride Hailing Web Application

Core Functionalities

- Account creation and user authentication
- Request a ride on the app by submitting a pick-up location
- Data persistence on a database to store records and transactions

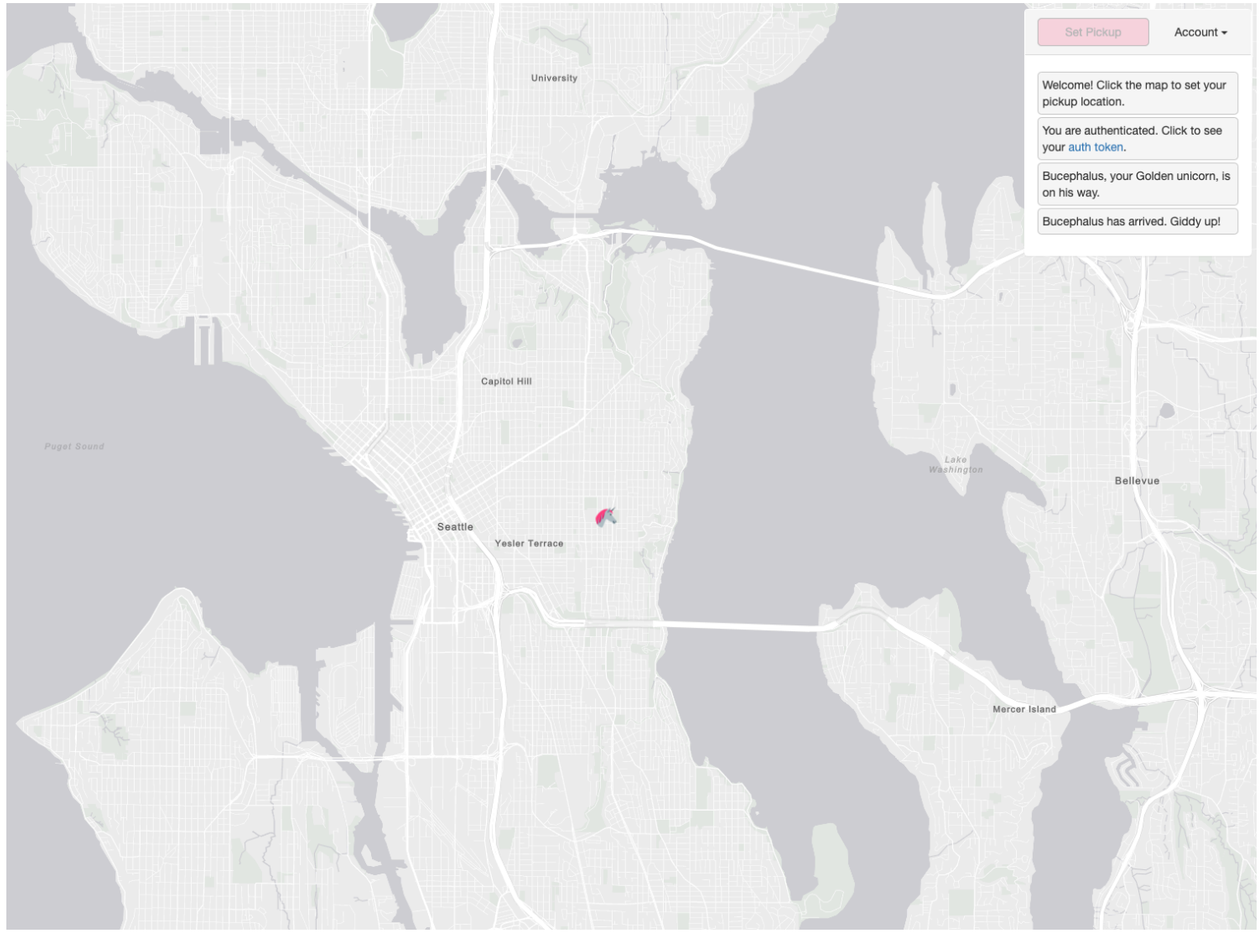
Serverless Web Application

WildRydes

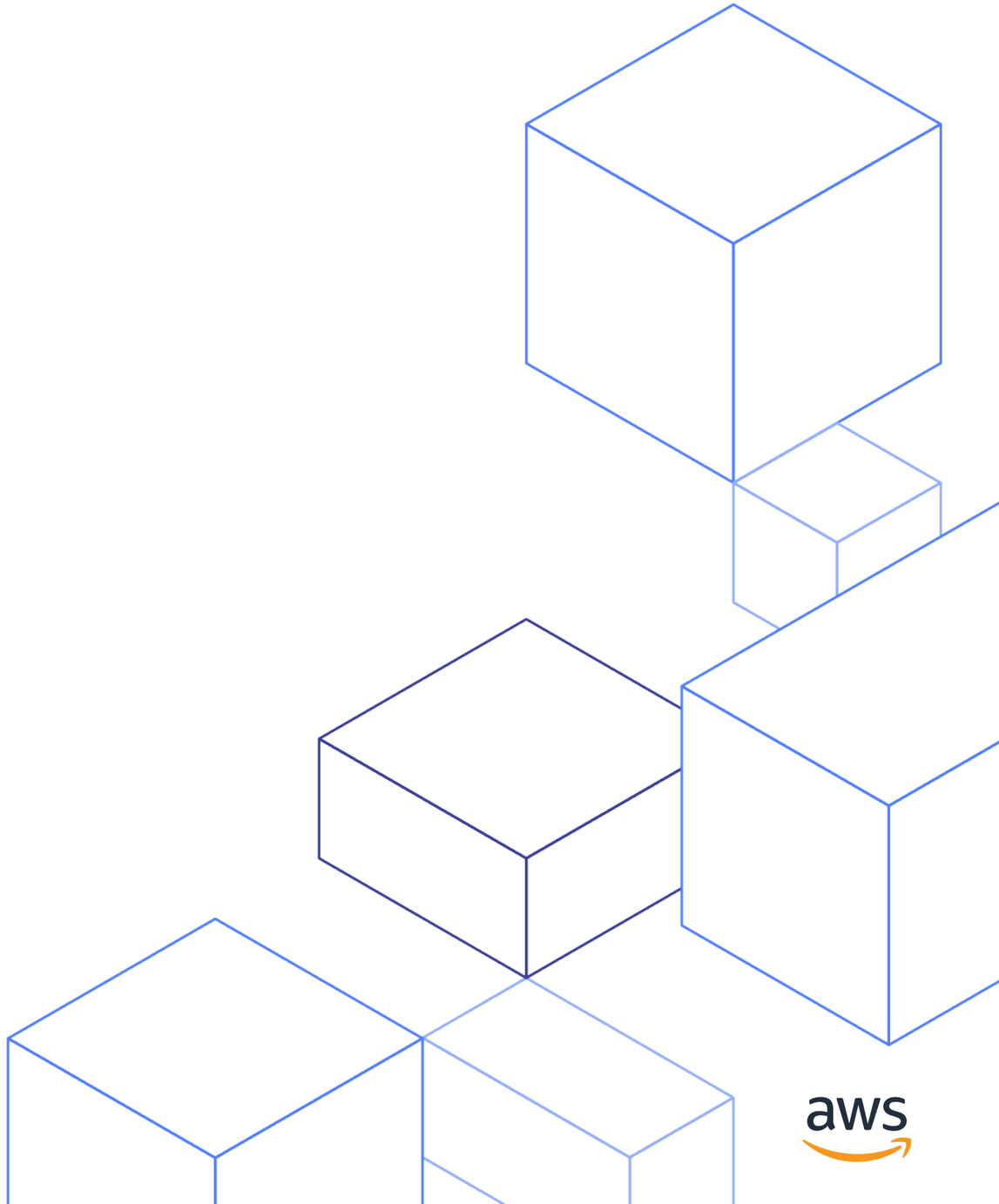
- Ride Hailing Web Application

Core Functionalities

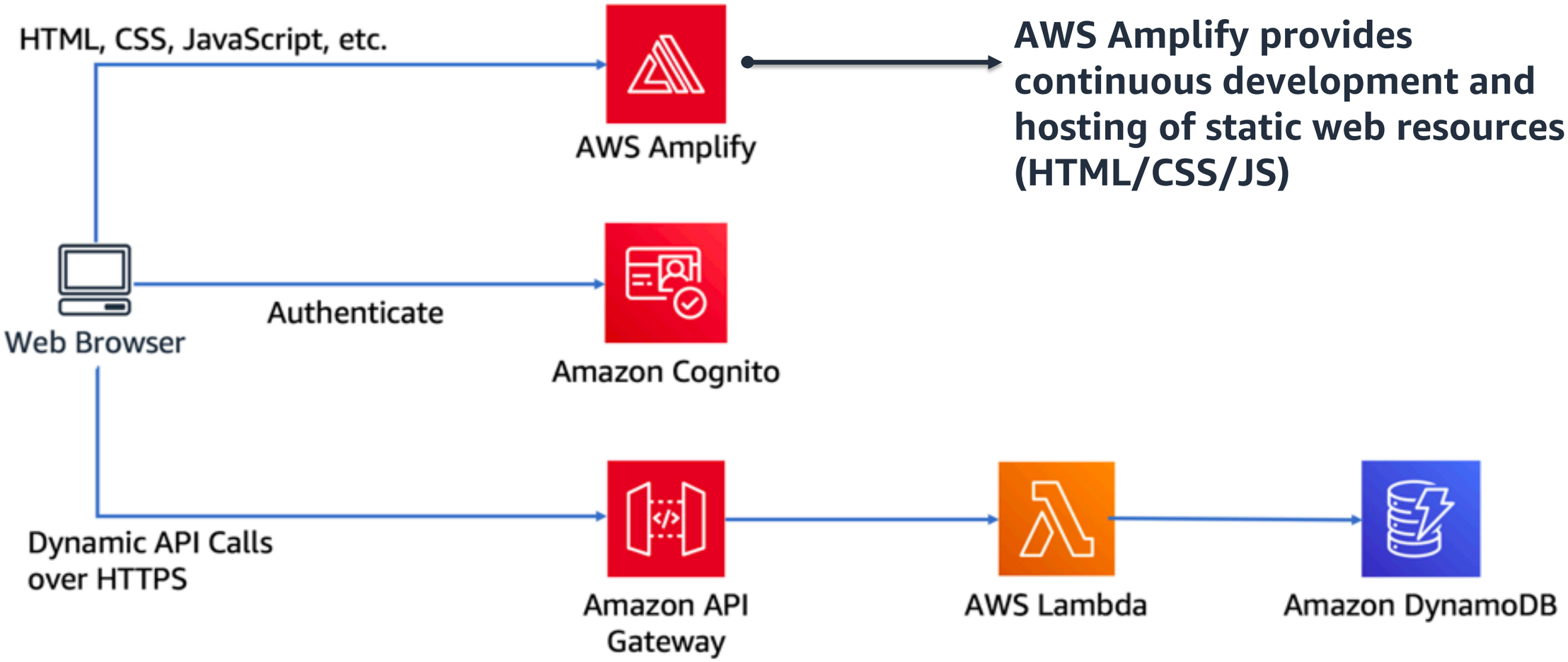
- Account creation and user authentication
- Request a ride on the app by submitting a pick-up location
- Data persistence on a database to store records and transactions



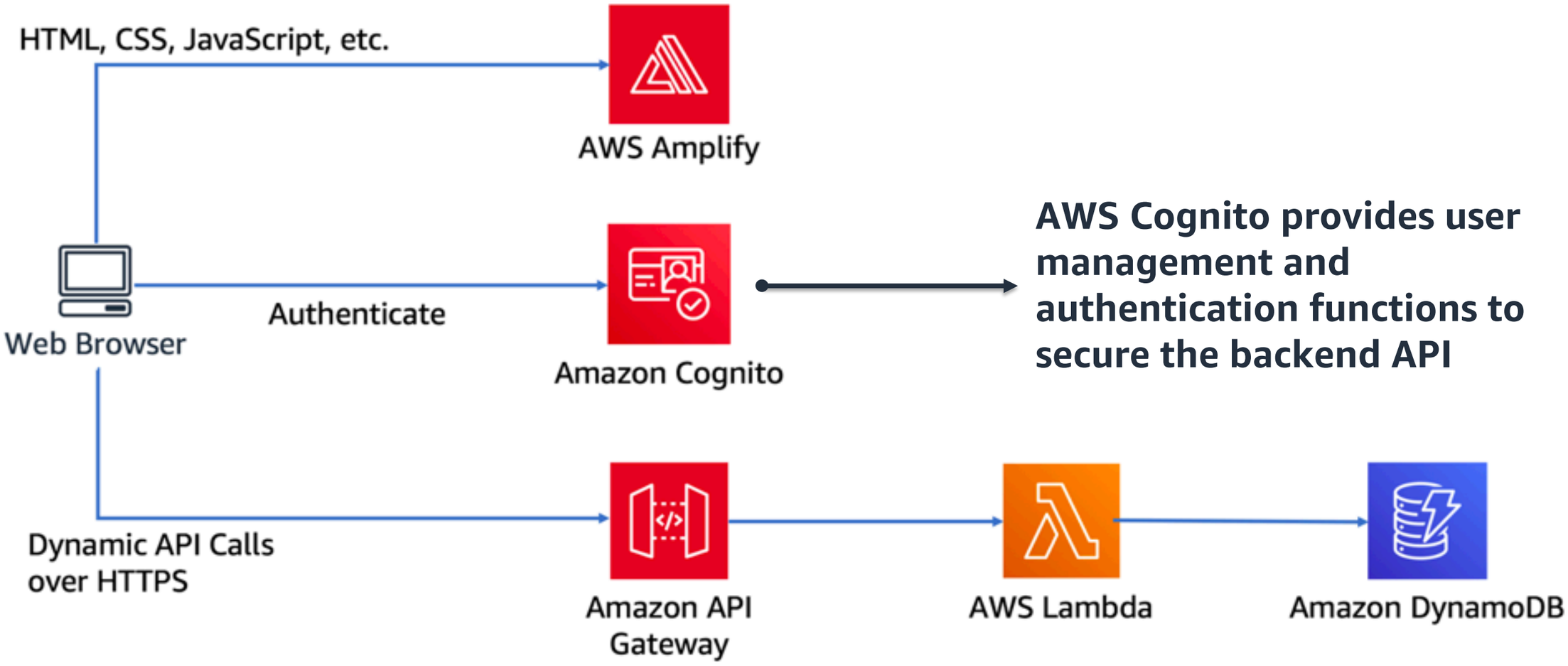
App Demo



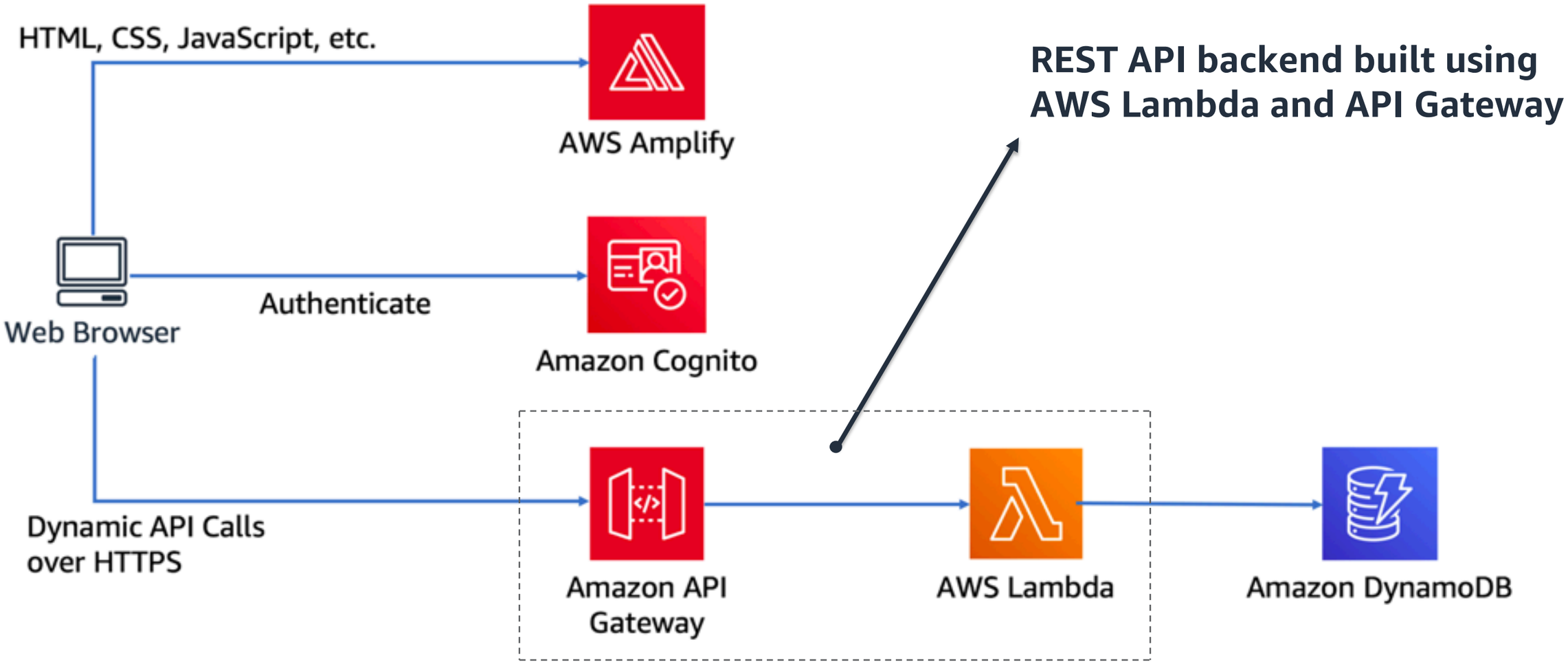
WildRydes Architecture Diagram



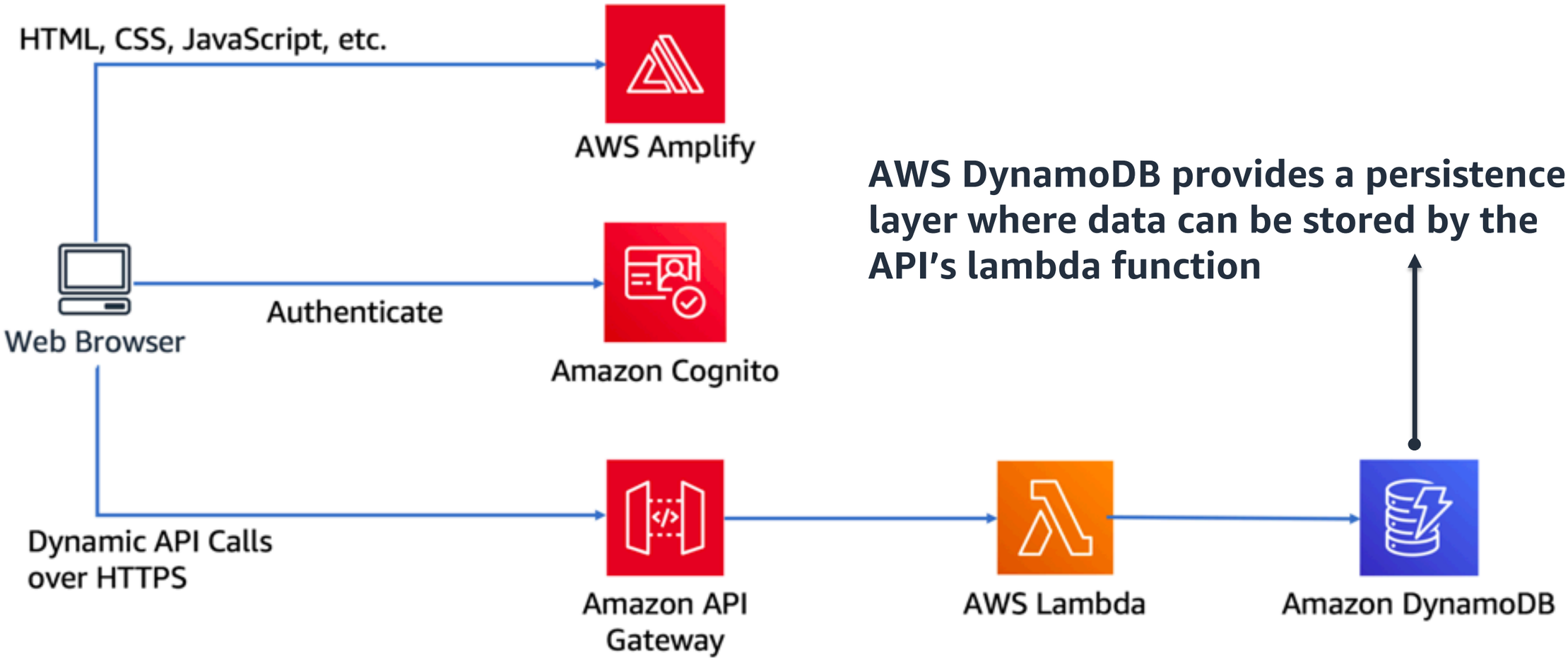
WildRydes Architecture Diagram



WildRydes Architecture Diagram



WildRydes Architecture Diagram



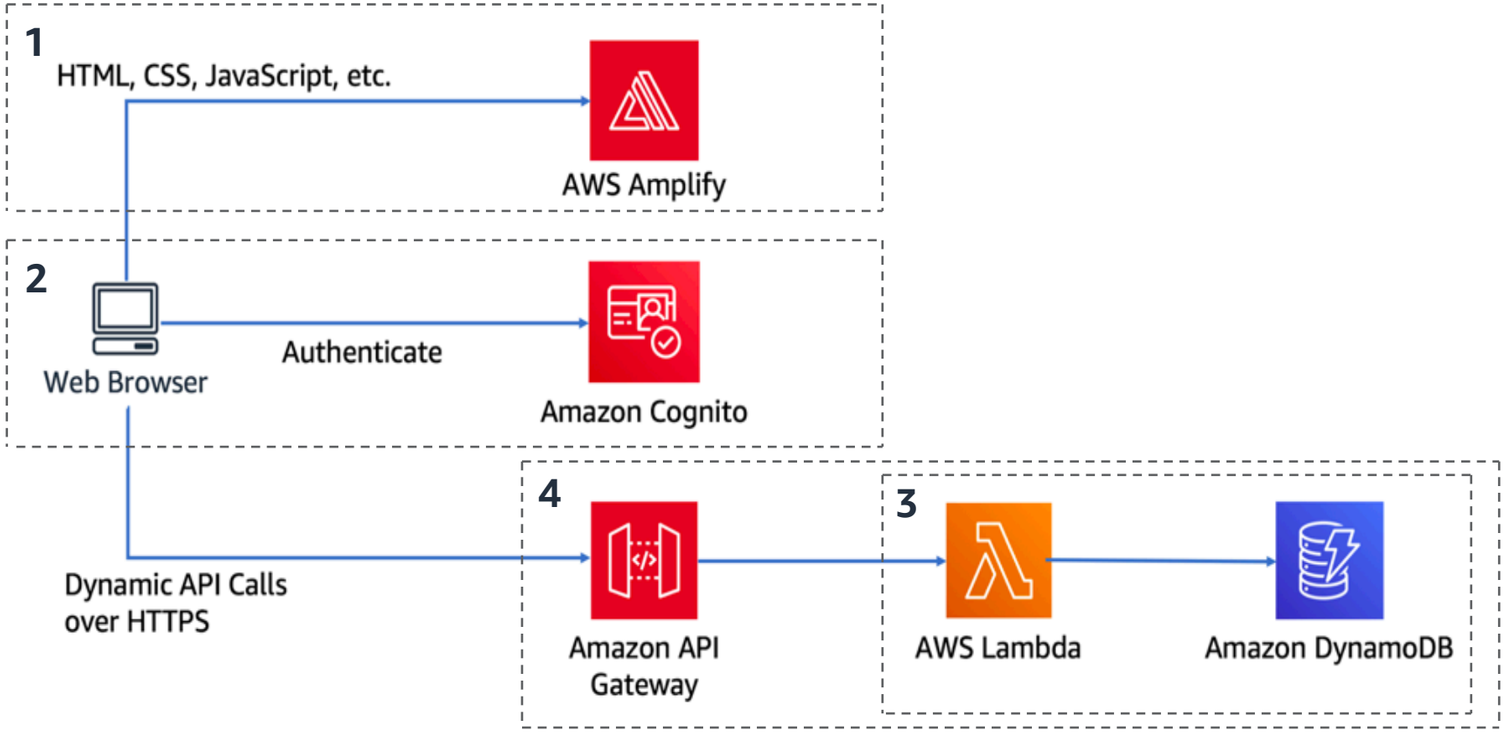
Built Sequence

1) Host front-end static site on AWS Amplify

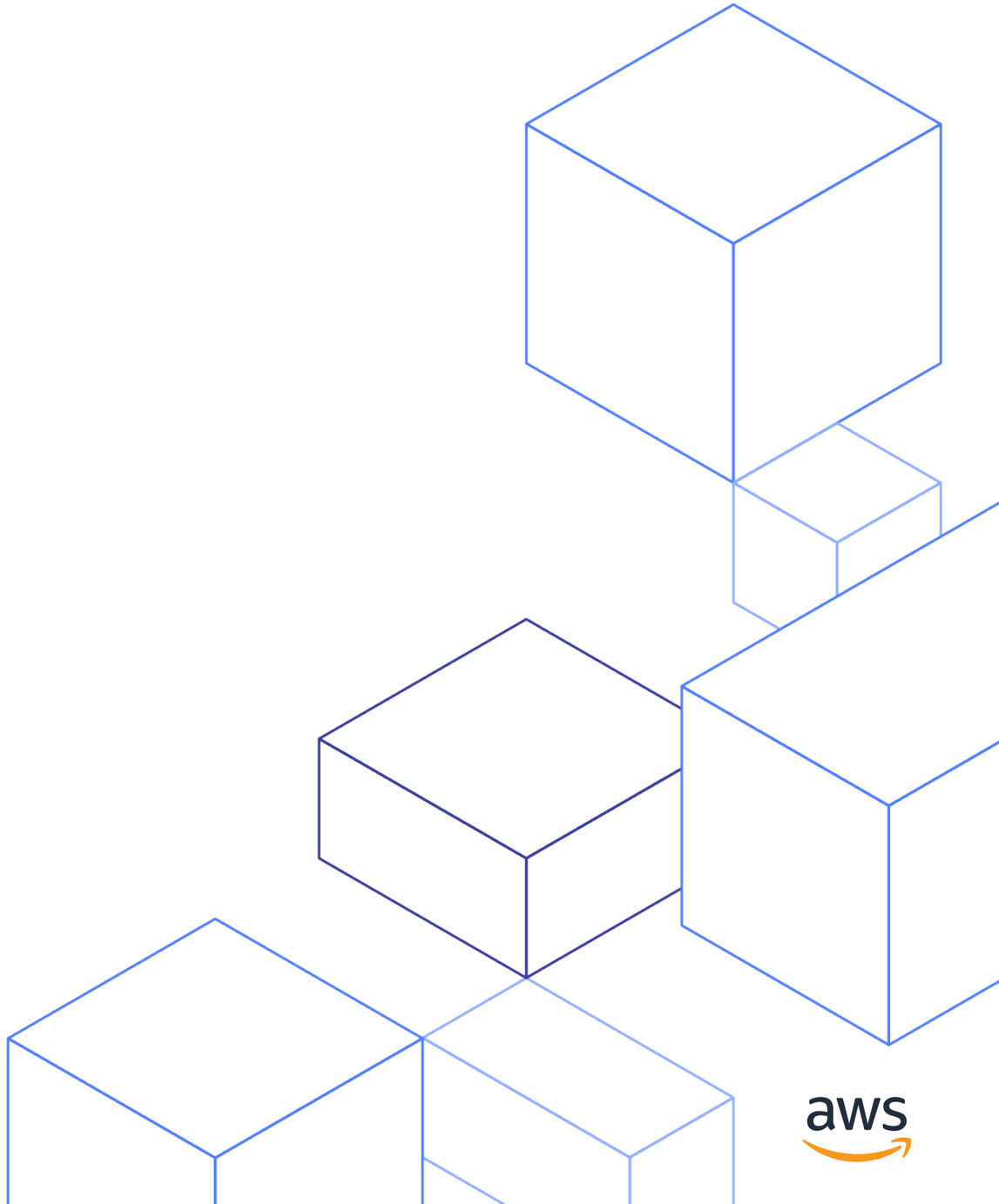
2) Set up user authentication with AWS Cognito

3) Build a serverless backend with AWS Lambda and DynamoDB

4) Deploy AWS Lambda function as a REST API with API Gateway



Let’s get buildin’!



Q&A

Glendon Thaiw Yong Neng

Resources (code snippets and workshop slide deck) available at:
https://github.com/glendont/aws-serverless-workshop

https://docs.google.com/forms/d/e/1FAIpQLSdpRfVhLYOnzggHASXlbOjkZoYi4t72qrACiPvwuSBxBVHOiw/viewform?usp=sf_link