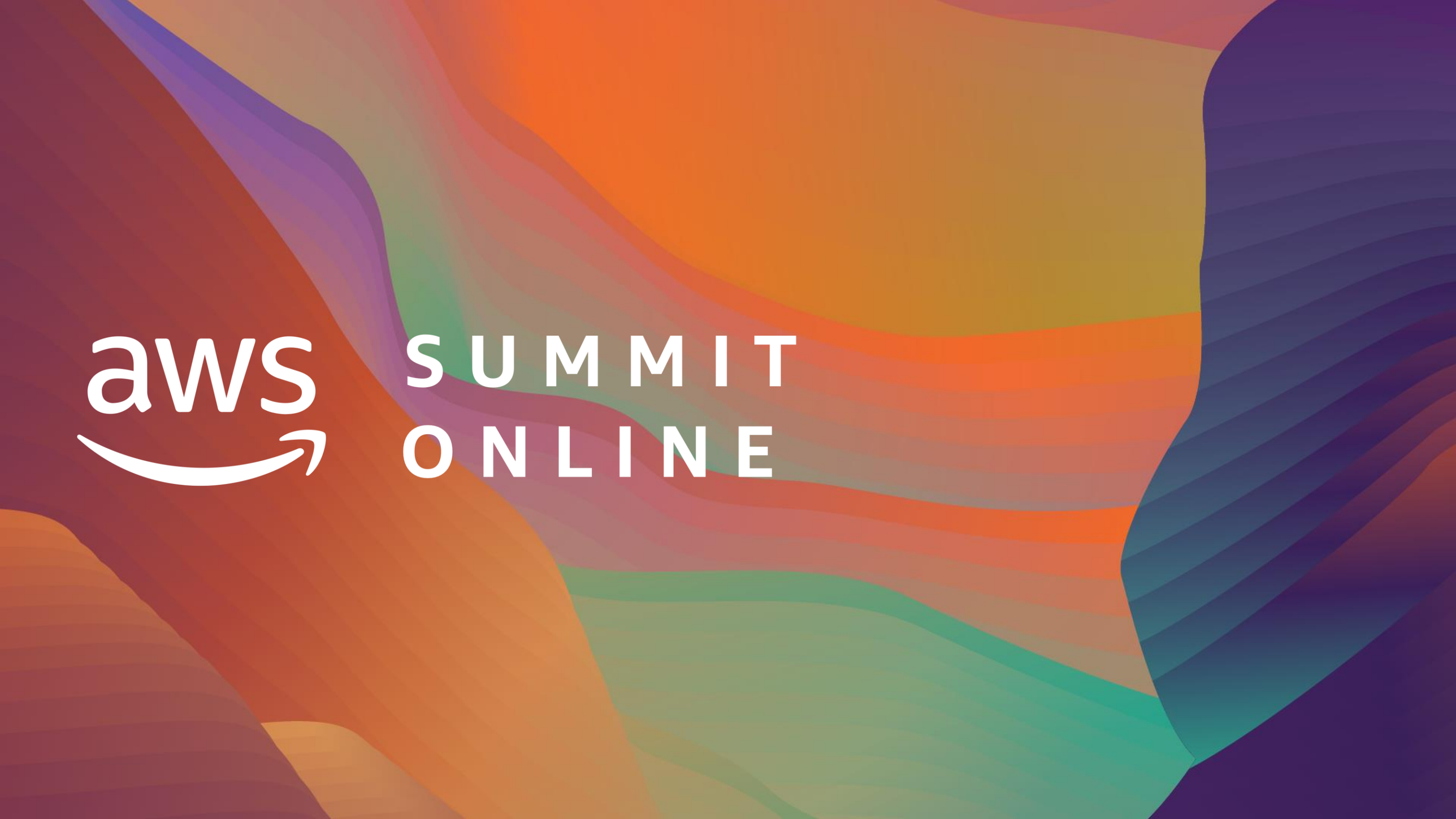

aws
SUMMIT
ONLINE

Building serverless applications with AWS Amplify

Derek Bingham

Senior Partner Solution Architect
Amazon Web Services

aws SUMMIT ONLINE

Quick recap: AWS Amplify

AWS Amplify: A development platform

Components

CLI

Client libs (iOS, Android, JS)

UI components

Open source framework

Amplify console – CI/CD and hosting

AWS Device Farm

AWS Managed Developer Services

AWS Amplify recap: CLI

```
# create new project
$ amplify init

# add feature
$ amplify add api

# test locally
$ amplify mock

# push changes
$ amplify push

# update feature
$ amplify update api
```

Convention over configuration

Manage single/multi-environment

Local mocking and testing

Code generation

native code

types & statements

AWS Amplify recap: Client

```
// import Amplify components
import { API } from 'aws-amplify'

// call Amazon API Gateway endpoint
const data = await API.get('orderApi', '/orders')
```

```
// import React component
import { withAuthenticator } from 'aws-amplify-react'

// main App component definition
class App extends React.Component {
    // your beautiful code
}

// add authentication
export default withAuthenticator(App)
```

Interact with services via client-side

Amplify Native for iOS and Android

JavaScript (JS) client for web and React Native

JS framework-specific components

AWS Amplify recap: Categories

DataStore

NEW

On-device persistent storage that automatically synchronises data between you apps and the cloud

Predictions

NEW

Add MI/ML capabilities to your app powered by cloud services

Analytics

Track user sessions, custom user attributes and in-app metrics

API

HTTP requests using REST and GraphQL with support for real-time data

Auth

AuthN + AuthZ library with prebuilt UI components for your app

Interactions

Conversational bots powered by deep learning technologies

PubSub

Connect your app to message-oriented middleware on the cloud

Notifications

Push notifications with campaign analytics and targeting

Storage

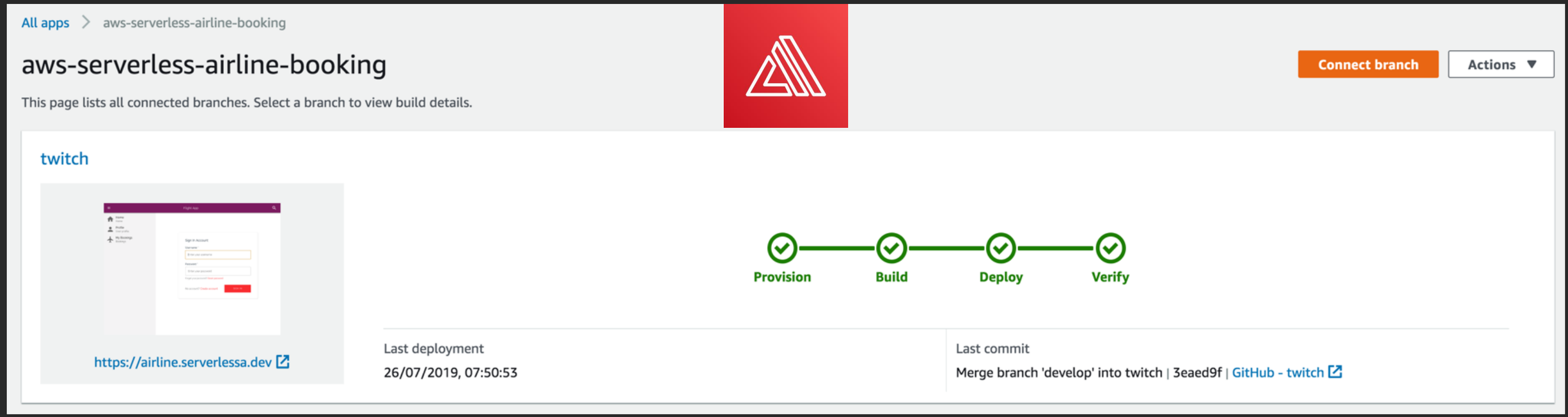
Manage user content securely in public, protected, and private storage

XR

Work with augmented reality and virtual reality content in your apps

AWS Amplify recap: Category adoption

AWS Amplify recap: Amplify Console



Git-based CI/CD for full-stack serverless apps

Jumpstart building serverless apps

What's new

aws SUMMIT ONLINE

What's new: Native SDKs

Amplify Clients

Use-case centric

Declarative abstractions

For example: Storage.put()

AWS Mobile SDKs

AWS service-centric

Low-level generated

For example: AWSS3TransferUtilityUploadExpression

What's new: Native categories

DataStore NEW

On-device persistent storage that automatically synchronises data between you apps and the cloud

Predictions NEW

Add MI/ML capabilities to your app powered by cloud services

Analytics

Track user sessions, custom user attributes and in-app metrics

API

HTTP requests using REST and GraphQL with support for real-time data

Auth

AuthN + AuthZ library with prebuilt UI components for your app

Interactions

Conversational bots powered by deep learning technologies

PubSub

Connect your app to message-oriented middleware on the cloud

Notifications

Push notifications with campaign analytics and targeting

Storage

Manage user content securely in public, protected, and private storage

XR

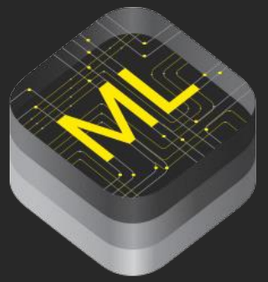
Work with augmented reality and virtual reality content in your apps

Example: Predictions in iOS

Custom models

Amazon SageMaker

Core ML models

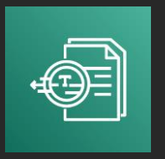


Pre-trained models

Amazon Rekognition

Amazon Comprehend

Amazon Polly



Amazon Textract

Amazon Transcribe

Amazon Translate

Escape hatch

For when convention isn't enough

Currently available categories

- Analytics
- Authentication
- Predictions
- Storage

```
let rekognitionService = Amplify.Predictions.getEscapeHatch
(key: .rekognition) as! AWSRekognition

let request = rekognitionService.AWSRekognitionCreateCollectionRequest()

rekognitionService.createCollection(request)
```

Demo: Adding predictions

aws SUMMIT ONLINE

Quick recap

What's new: Amplify DataStore

Multi-platform (iOS/Android/React Native/Web), on-device persistent storage engine that automatically synchronizes data between mobile/web apps and the cloud using GraphQL

Automatic versioning, conflict detection, and resolution in the cloud

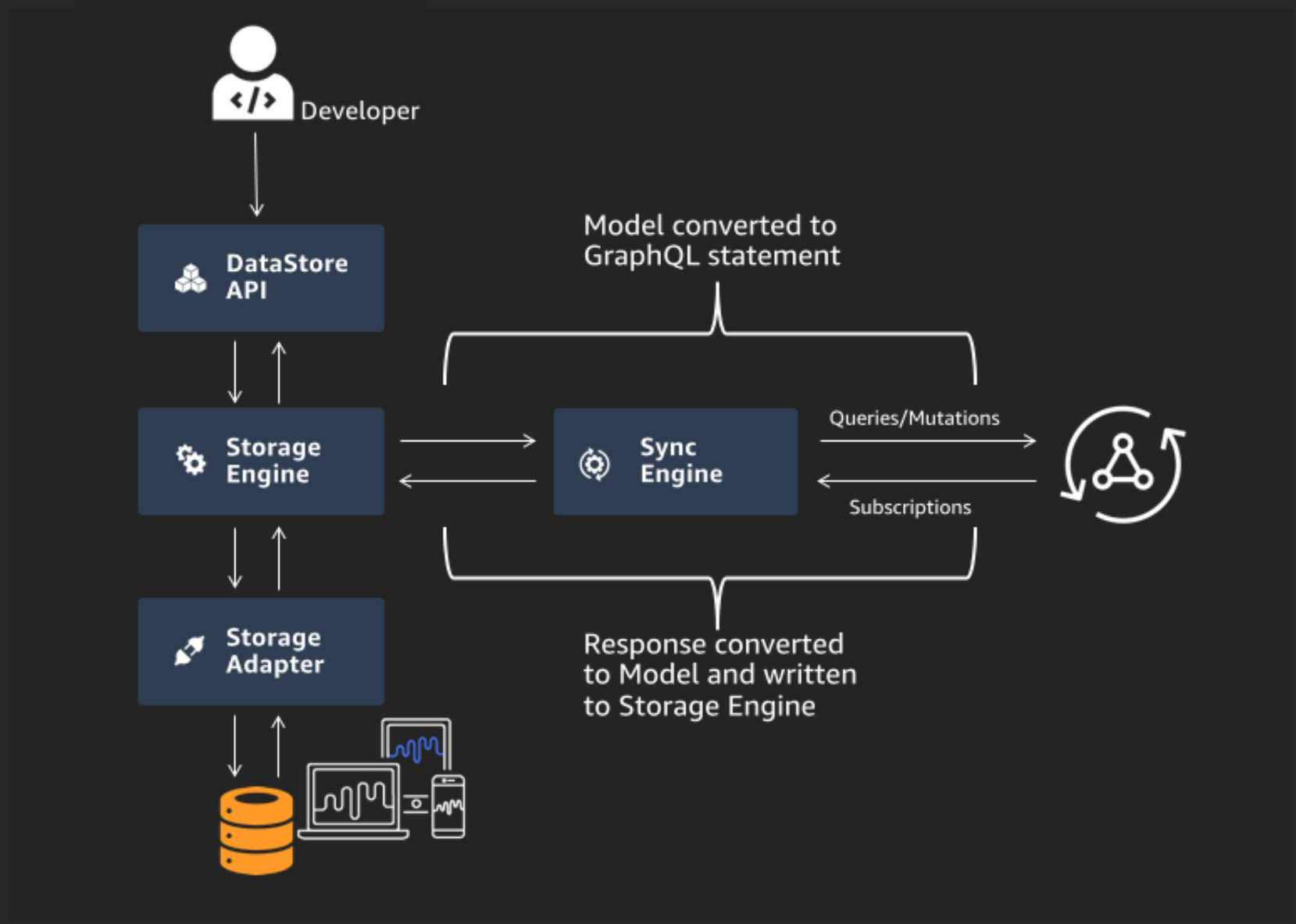
Familiar and local-first programming model

Delta Sync and Auto-Merge

Powered by AppSync and GraphQL

What’s new: Amplify DataStore

What's new: Amplify DataStore



before

```
function Posts({ params }) {
  const { subscribeToMore, ...result } = useQuery(POST_QUERY)

  return (
    <PostsPage
      {...result}
      subscribeToNewPosts={() =>
        subscribeToMore({
          document: POSTS_SUBSCRIPTION,
          updateQuery: (prev, { subscriptionData }) => {
            if (!subscriptionData.data) return prev;
            const newPost = subscriptionData.data.newPost;
            return {
              ...prev,
              {
                posts: [newPost, ...prev.posts]
              }
            }
          }
        })
      }
    />
  )
}
```

after

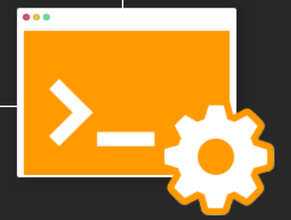
```
const subscription = DataStore.observe(Post)
  .subscribe(msg => fetchPosts())

async function fetchPosts() {
  const posts = await DataStore.query(Post)
  setState(current => ({ ...current, posts }))
}
```

What's new: Transformers

```
$ amplify add api
```

```
# schema.graphql
type Post @model {
  id: ID!
  title: String!
}
```



What's new: Transformers

`@model`
Top-level entity; creates Amazon DynamoDB table, resolvers, and additional schema (queries, mutations, and subscriptions) for base type

`@connection`
Enables relationships between `@model` types

`@auth`
Enables set of authorization rules

`@searchable`
Handles streaming the data of an `@model` object type to Amazon Elasticsearch Service and configures search resolvers

`@versioned`
Enables versioning

`@function`
Enables adding an AWS Lambda function as a data source

`@key`
Enables configuring custom indexes for `@model` types

`@predictions`
Access an orchestration of AI/ML services such as Amazon Rekognition, Amazon Translate, and/or Amazon Polly

Demo: Creating a new data model

aws SUMMIT ONLINE

Quick recap

6 tips and best practices

AWS Amplify tips and best practices (1)

Enable authorization with
@auth to protect models

AWS Amplify tips and best practices (2)

Take advantage of
field-level authorization

AWS Amplify tips and best practices (3)

Use local mocking and testing
to quickly test things out

AWS Amplify tips and best practices (4)

Use multiple environments
to test in production

AWS Amplify tips and best practices (5)

Amplify Console
branch previews

AWS Amplify tips and best practices (6)

Get to know @key directives for additional access patterns

Go build serverless apps

(with AWS Amplify)

Links and resources

https://amplify.aws/community/

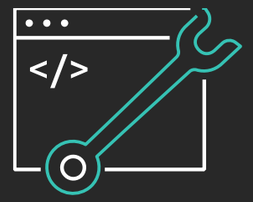
https://awsappsync.dev/

https://aws-amplify.github.io/

https://github.com/aws-amplify

Learn to build modern applications on AWS

Resources created by the experts at AWS to help you build and validate developer skills



Enable rapid innovation by developing your skills in designing, building, and managing modern applications

Learn to modernize your applications with free digital training and classroom offerings, including Architecting on AWS, Developing on AWS, and DevOps Engineering on AWS

Validate expertise with the AWS Certified DevOps – Professional or AWS Certified Developer – Associate exams

Visit the developer learning path at aws.amazon.com/training/path-developing

Thank you!

Derek Bingham

 @derekwbingham

 @deekob

aws SUMMIT ONLINE