AWS
re:Invent

Modernize REST with GraphQL APIs and AWS AppSync

Derek Bingham

Partner Solutions Architect
Amazon Web Services

Lucas Schejtman

Solutions Architect
Amazon Web Service

AWS re:Invent

aws

Before we jump in

- There will be coding (solutions are provided)

- You need an AWS account (provided)

- You need enough IAM permissions to create and delete (provided)

Workshop objectives

- An understanding of GraphQL and its benefits

- How it fits into the overall evolution of APIs

- Introduce AWS AppSync

- Introduce AWS Amplify

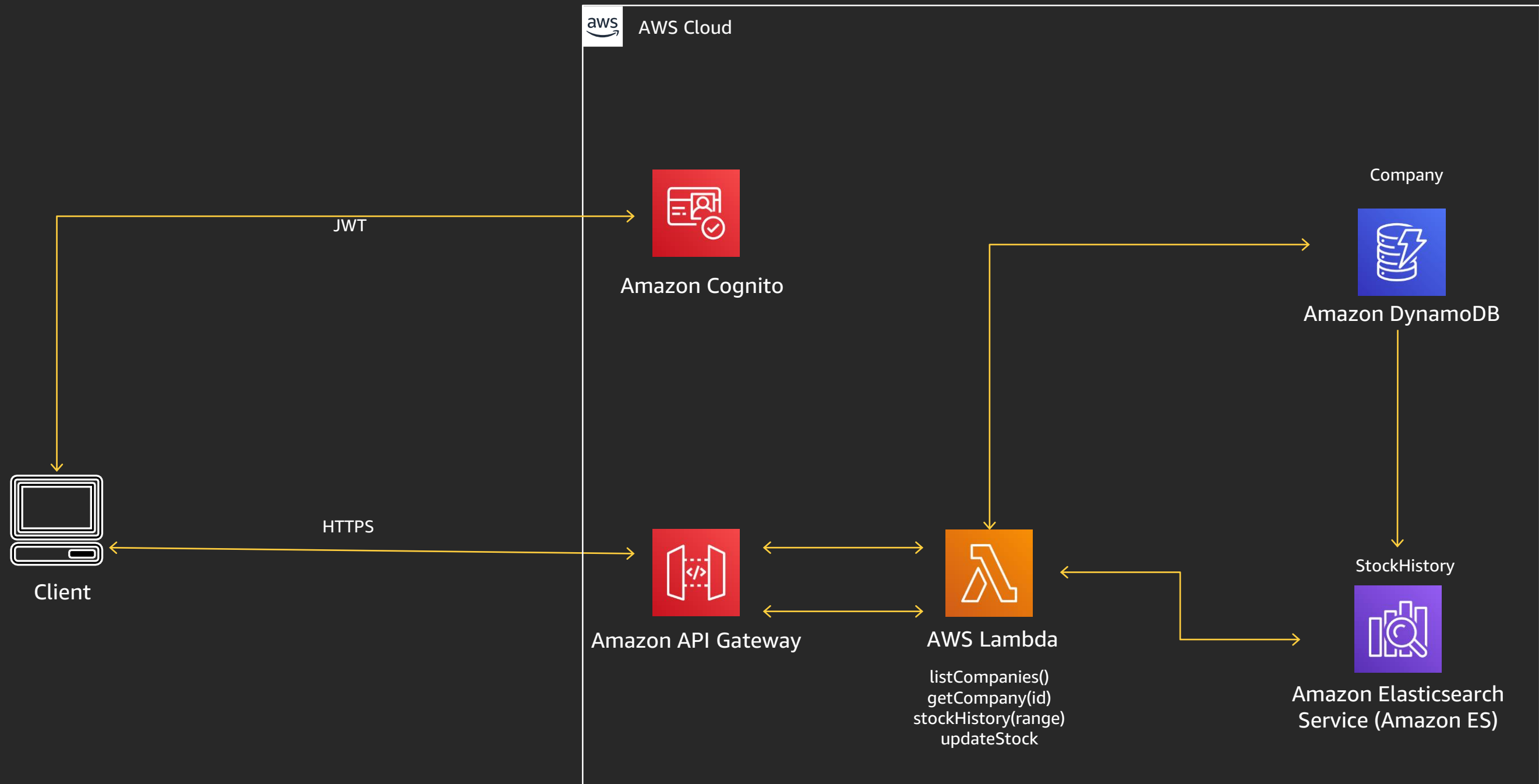- Show how we can use these tools to transform an existing RESTful API to use the benefits GraphQL

Part 1 – Deploy existing app (10 minutes)

https://www.appsyncworkshop.com

Complete the Setting Up section

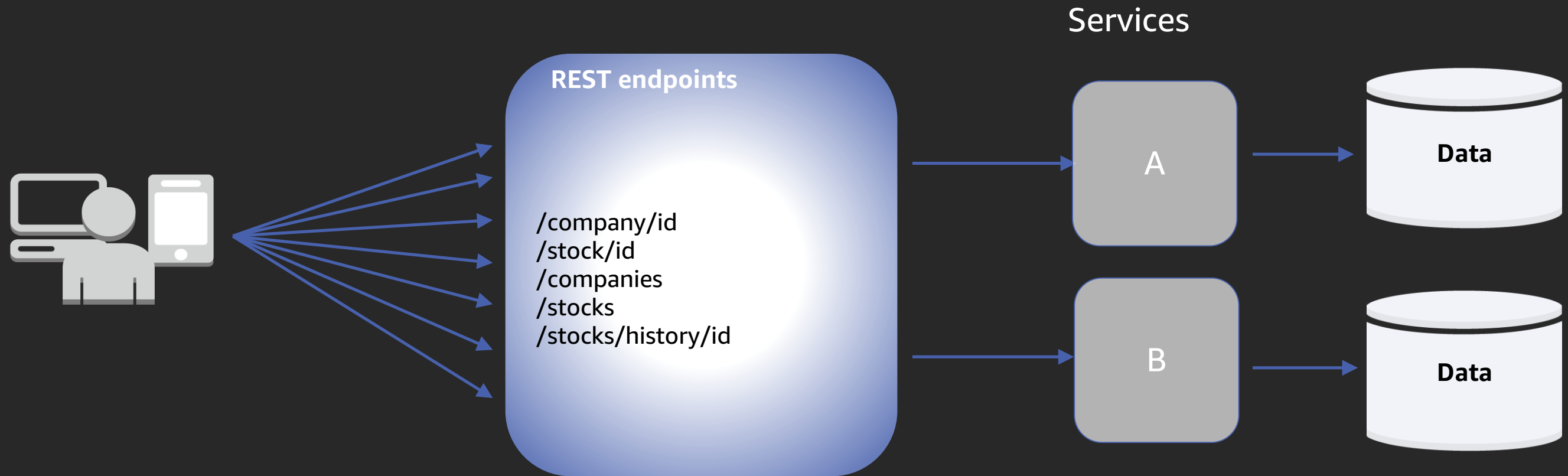# Demo existing stocks app demo

Initial architecture

New version product backlog

- ☐ Provide real-time stock updates
- ☐ Provide mobile platform solution
- ☐ Don't affect existing users
- ☐ Reduce application initial load times

What API options do we have ?

Option 1. Continue with REST

Services

REST endpoints

/company/id
/stock/id
/companies
/stocks
/stocks/history/id

A

Data

B

Data

✓ Trivial to set up
✓ Standard HTTP Calls

✗ Relationships
✗ Lists with reduced information
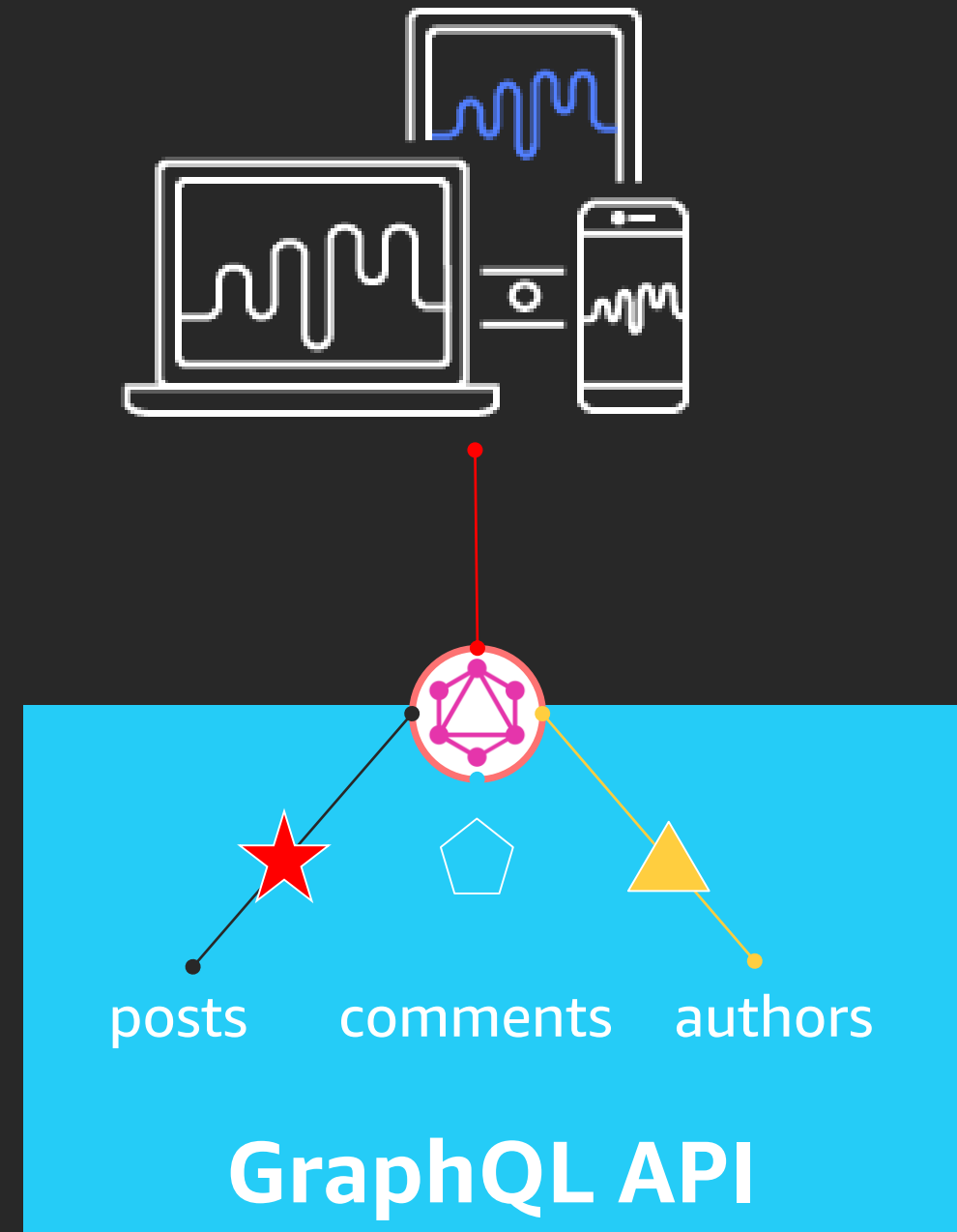✗ Query support
✗ Ordering and pagination
✗ Notifications

Option 2. GraphQL

GraphQL server

Schema

HTTPS

Web sockets

Resolver

Resolver

Services

A

Data

Data
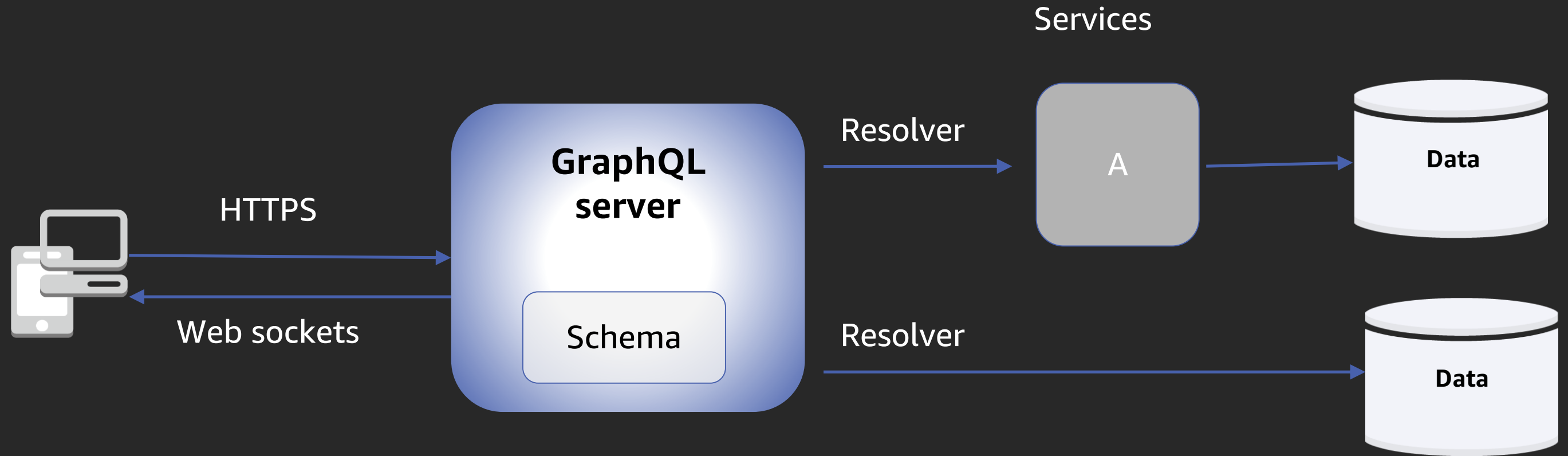
✓ Single endpoint
✓ Supports near real-time updates
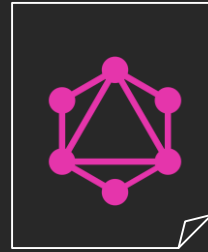✓ Caters for varying clients (mobile)

❌ Not as well used/known as REST
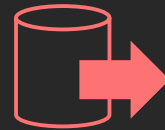
What is GraphQL

Query language for APIs and a runtime

GraphQL schema and operations

Types Queries Mutations Subscriptions

```
type User {
    id: ID!
    username: String!
    firstName: String
    lastName: String
    daysActive: Int
}
```

A query language

Queries

```
query GetPost {
  getPost(id: "1") {
    id
    title
  }
}
```
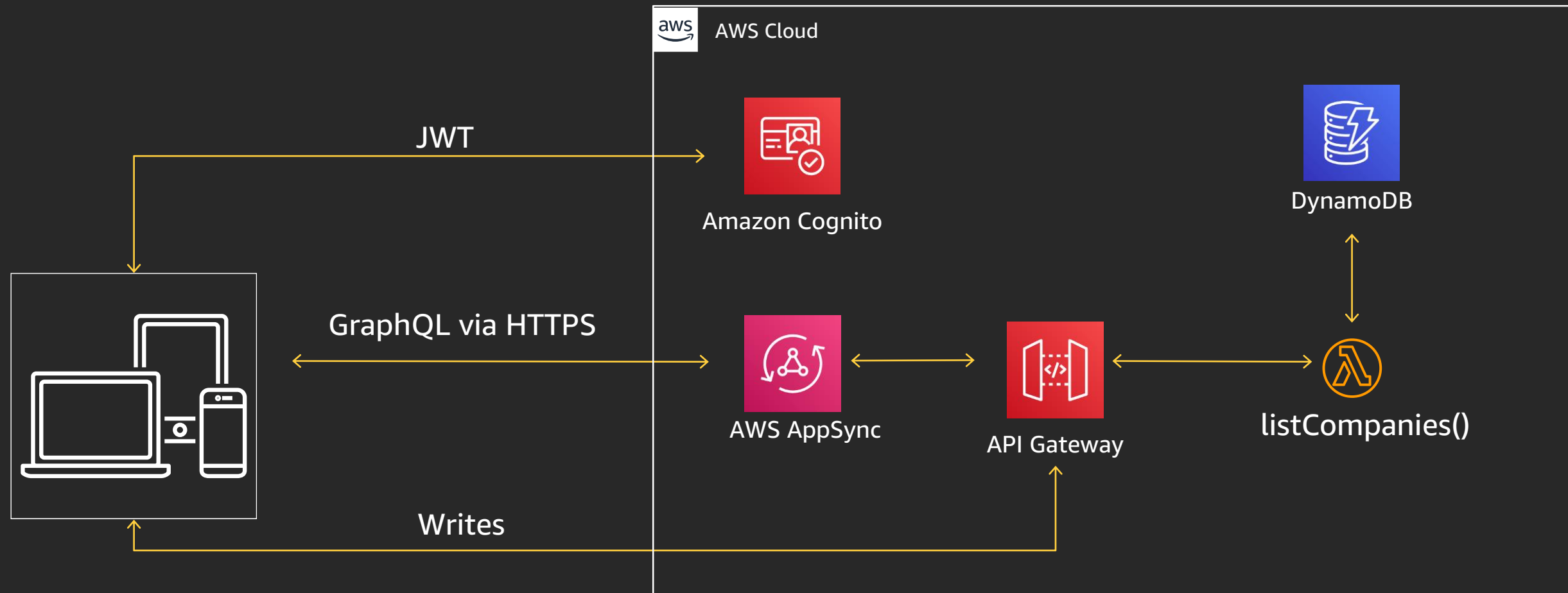
Mutations

```
mutation CreatePost {
  createPost(title: "Summit") {
    id
    title
  }
}
```

Subscriptions

```
subscription OnCreatePost {
  onCreatePost {
    id
    title
  }
}
```
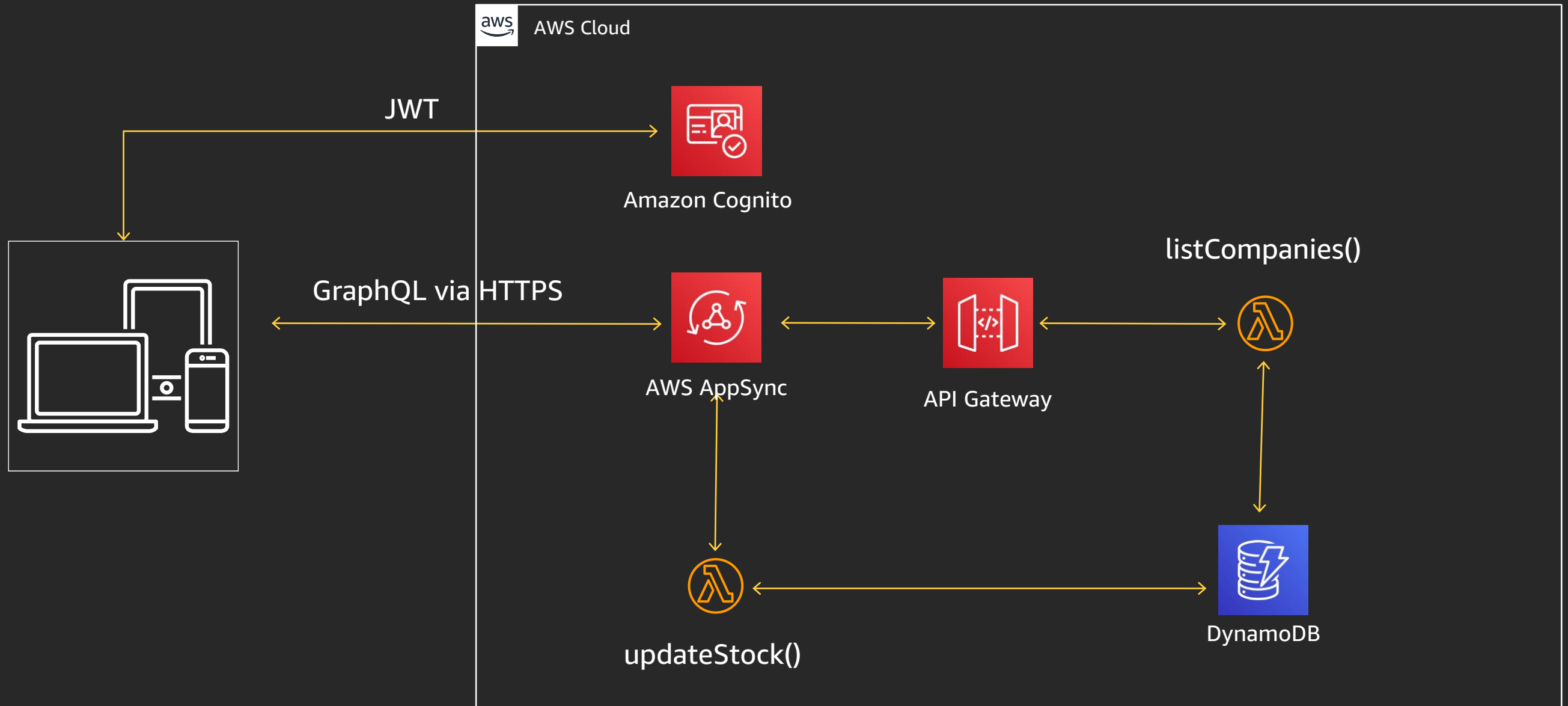
Part 1: Queries

Migrate an existing REST endpoint

- ✓ Large legacy API to deprecate
- ✓ Reduces client chattiness
- ✓ Client can control data **exactly**
- ✓ Reduce initial load times
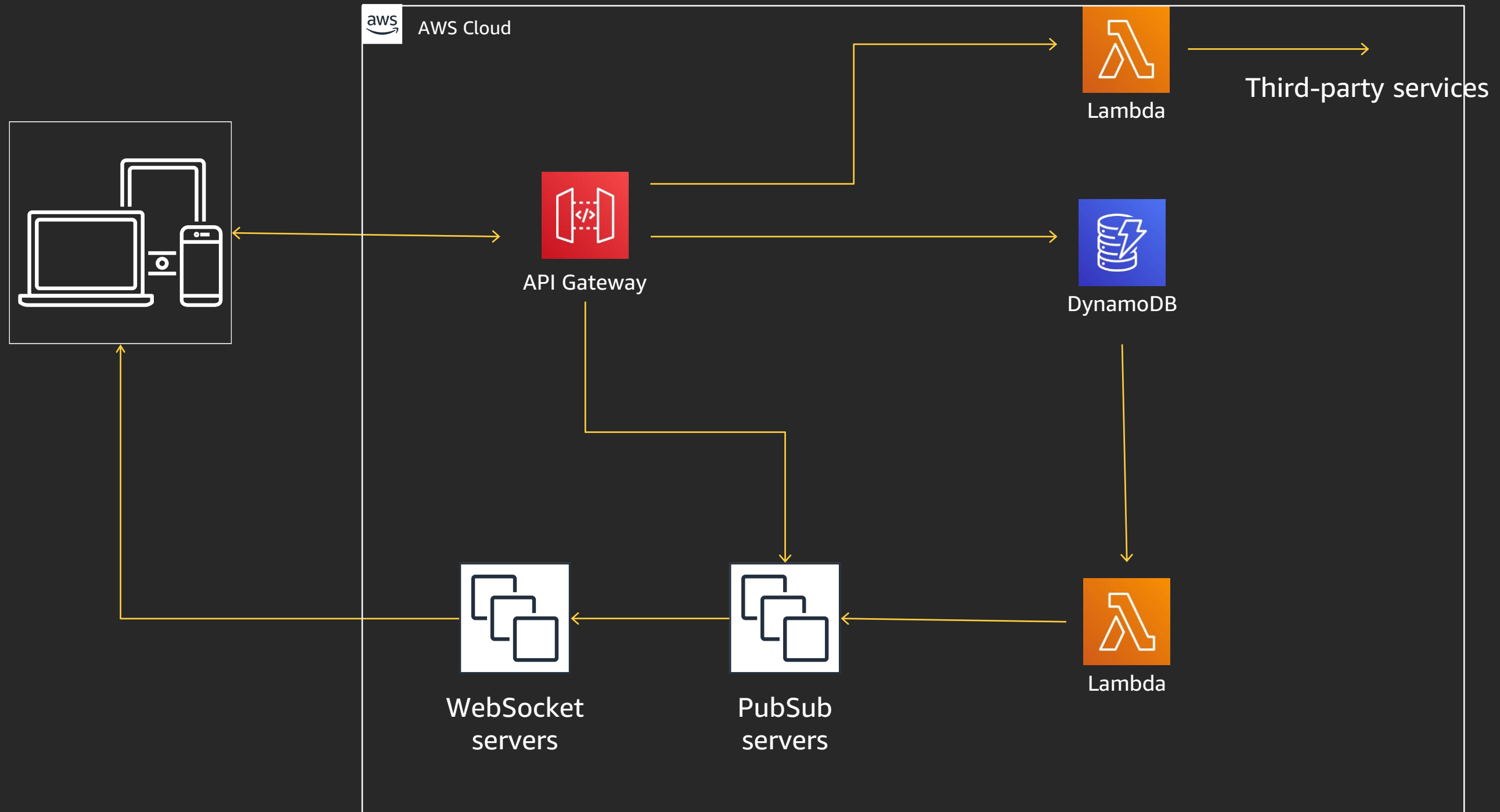- ✓ Clients can continue to use API Gateway

Part 2: Mutations
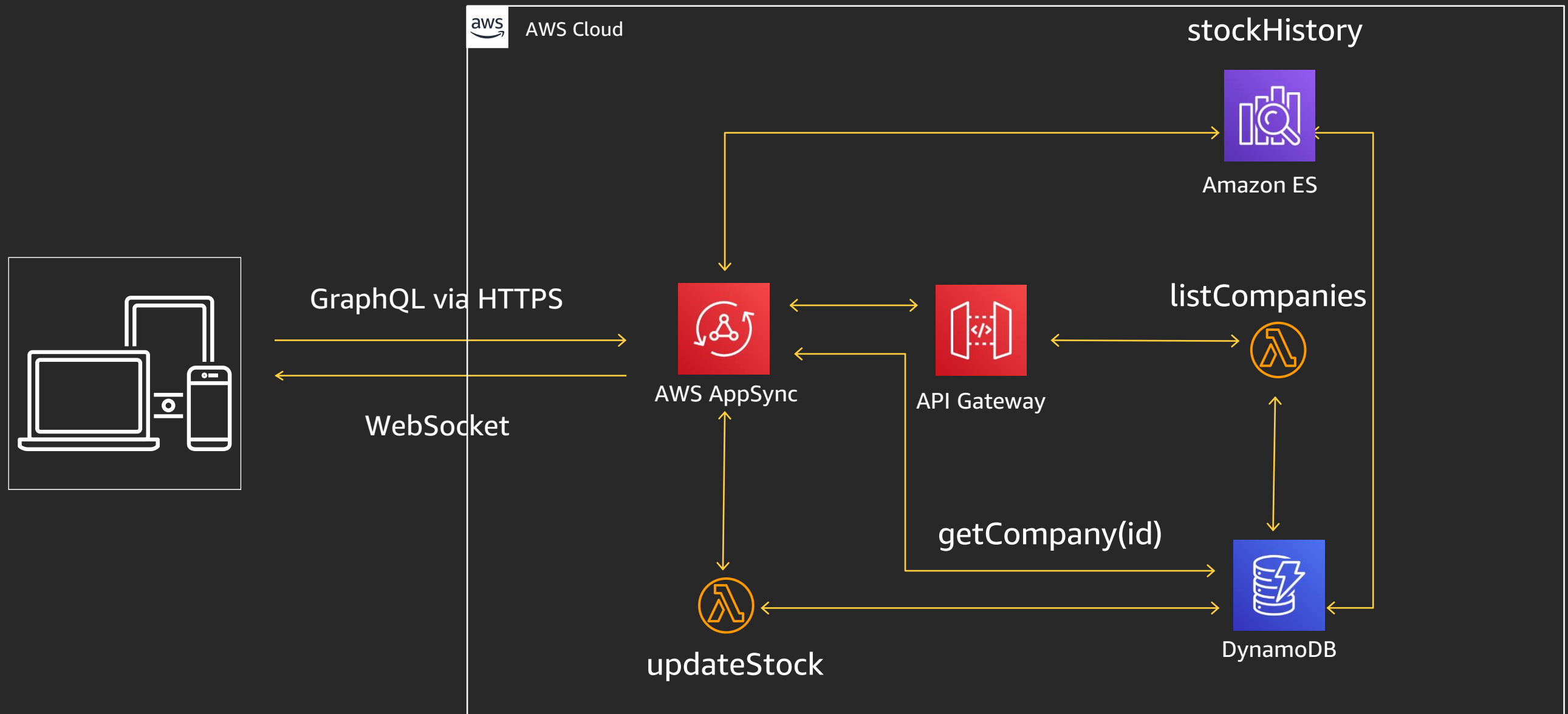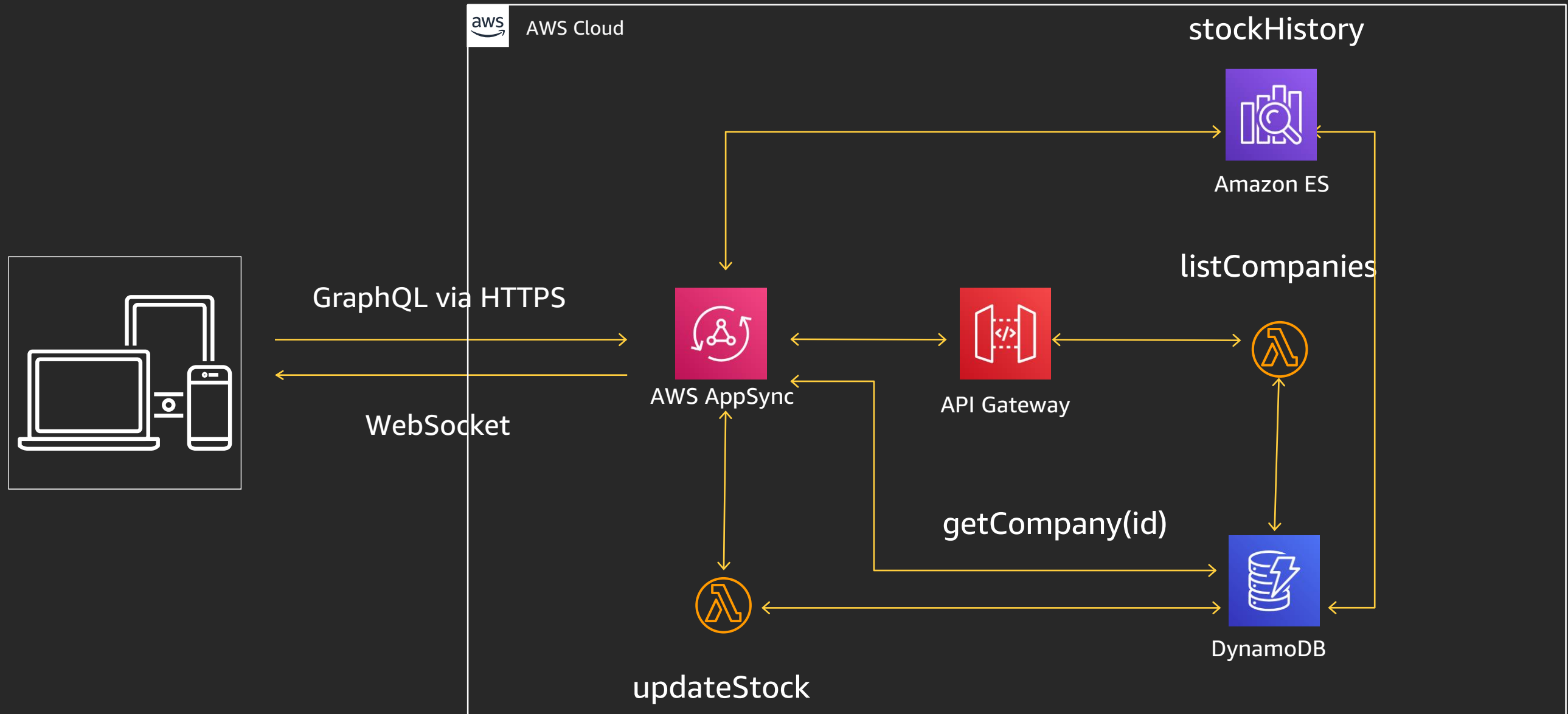
Migrate buy/sell stock

Part 3: Subscriptions

Traditional real-time updates

With AWS AppSync

Final State

Thank you!

AWS
re:Invent

aws

Please complete the session survey in the mobile app.