AWS AppSync GraphQL APIs

GraphQL vs REST

- Strongly Typed Schema
- Reduce over fetching (or under fetching) data
- Single endpoint to serve your API
- Faster Iterations
    - Strict schema helps with this
    - You don't need to know how your backend implements the API

GraphQL on AWS

- Access a variety of Datasources (DynamoDB, RDS, OpenSearch, Lambda, Eventbridge, Etc.)
- Merge several APIs into a single source API

AWS AppSync Components

Schema

- Defines the shape of your data

```
type Person {
    id: ID!
    name: String
    email: AWSEmail!
}

type Query {
    people: [Person]
    person(id: ID!): Person
}

type Mutation {
    addPerson(id: ID!, name: String, email: AWSEmail!): Person
    deletePerson(id: ID!):
```

AWS AppSync Components

Data sources

The actual data your are interacting with via your schema

- DynamoDB Table
- RDS Database
- OpenSearch
- HTTP APIs
- Lambda
- None Data source

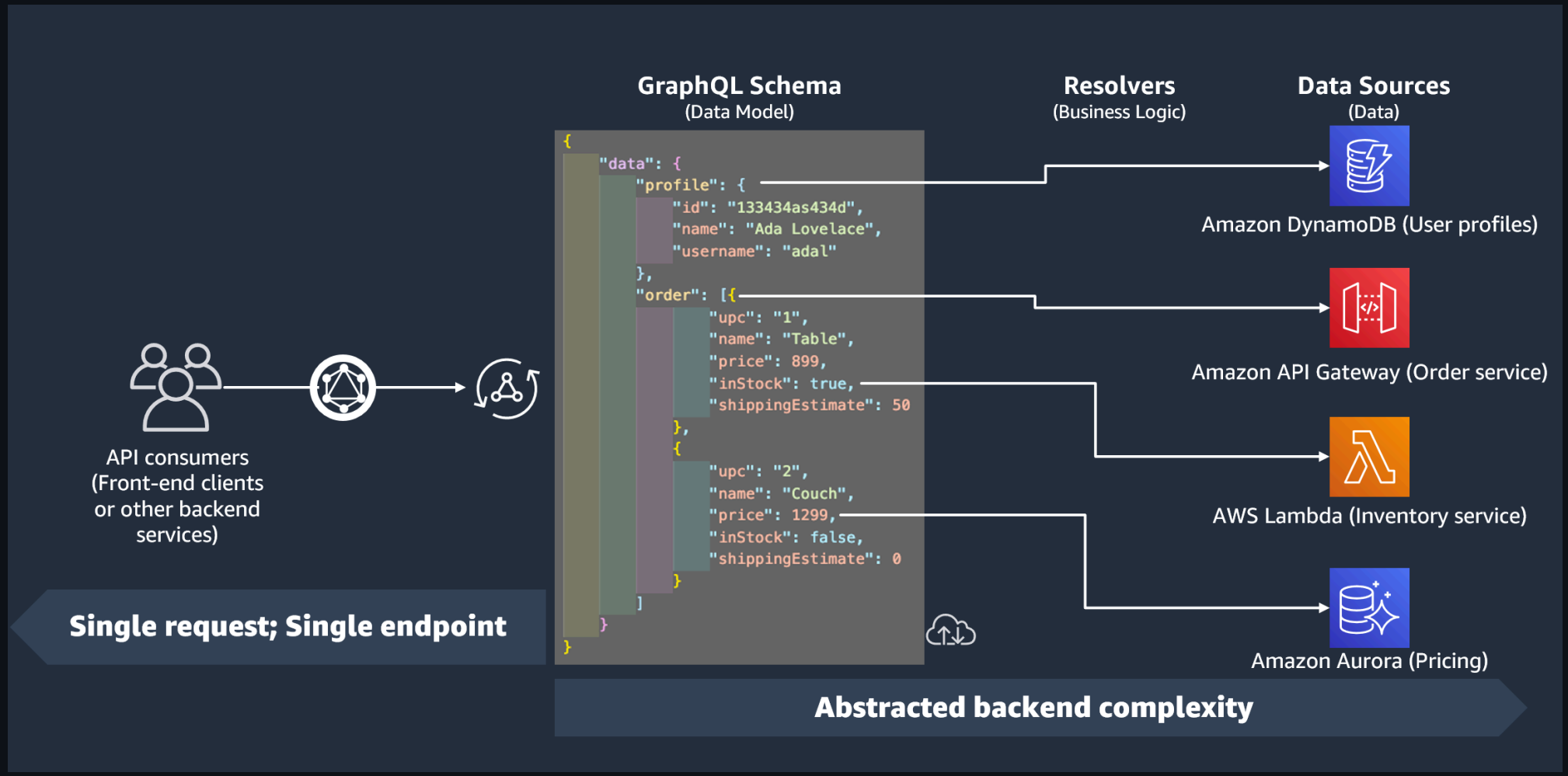
AWS AppSync Components

Resolvers

How to interact with your Data sources

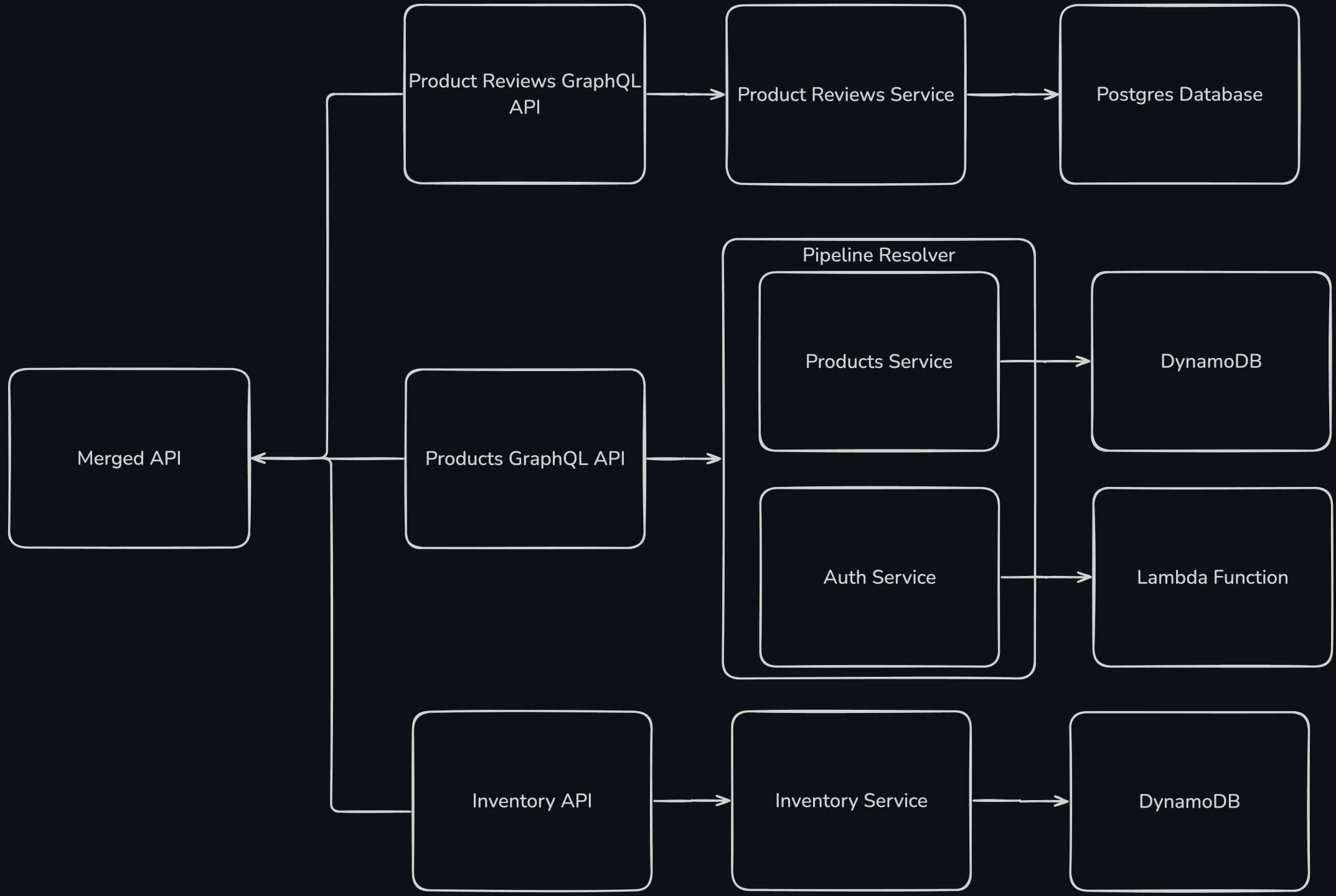
- Unit/Direct Resolvers or Pipeline Resolvers
- JavaScript or VTL

Putting It All together



Source

Demo Time


Product Reviews GraphQL API
Product Reviews Service
Postgres Database
Pipeline Resolver
Products Service
DynamoDB
Merged API
Products GraphQL API
Auth Service
Lambda Function
Inventory API
Inventory Service
DynamoDB