High Performance Caching Strategies with Apollo Server

Peter Kellner
DEVELOPER, CONSULTANT AND AUTHOR

@pkellner linkedin.com/in/peterkellner99 ReactAtScale.com

Using cache and providing truth are enemies of each other.

Possible Problems Cache Brings

Harder to find and fix bugs in your apps

Features are harder to add

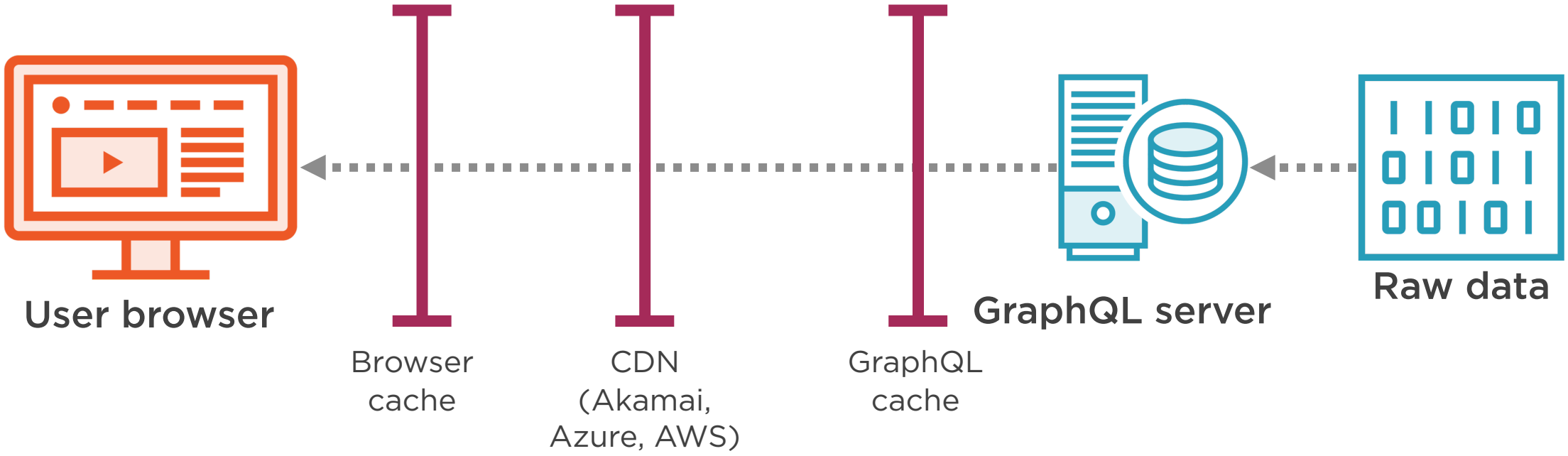
Managing production updates for consistency hard

Testing becomes much more difficult

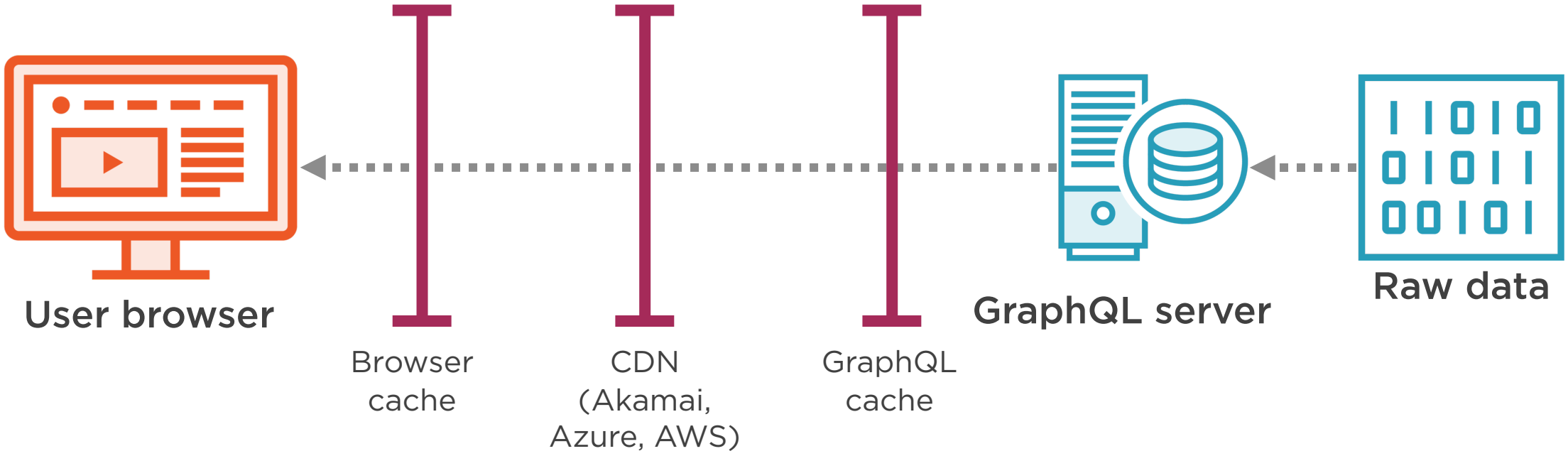
“Premature optimization is the root of all evil.”

Donald Knuth

Layers between User and GraphQL Server



Layers between User and GraphQL Server



What's Coming in This Module

Strategies for getting data as fast as possible to your user.

Caching always creates multiple versions of the truth.

Cache Hints

What to cache

How long to cache

Allow intermediate cache participation

Two Ways to Set Cache Hints

Declaratively

In schema

Imperatively

In resolvers

HTTP GET method
designed for caching.

Apollo invented their own protocol for supporing GraphQL queries over GET.

GraphQL Query String

```
speakers(offset: 5, limit: 77) {
  datalist {
    id
    first
    last
    favorite
    company
    twitterHandle
    bio
    sessions {
      id
      title
      eventYear
      room {
        id
        name
        capacity
      __typename
      }...
__
```

GraphQL Query String

```
speakers(offset: 5, limit: 77) {
  datalist {
    id
    first
    last
    favorite…
```

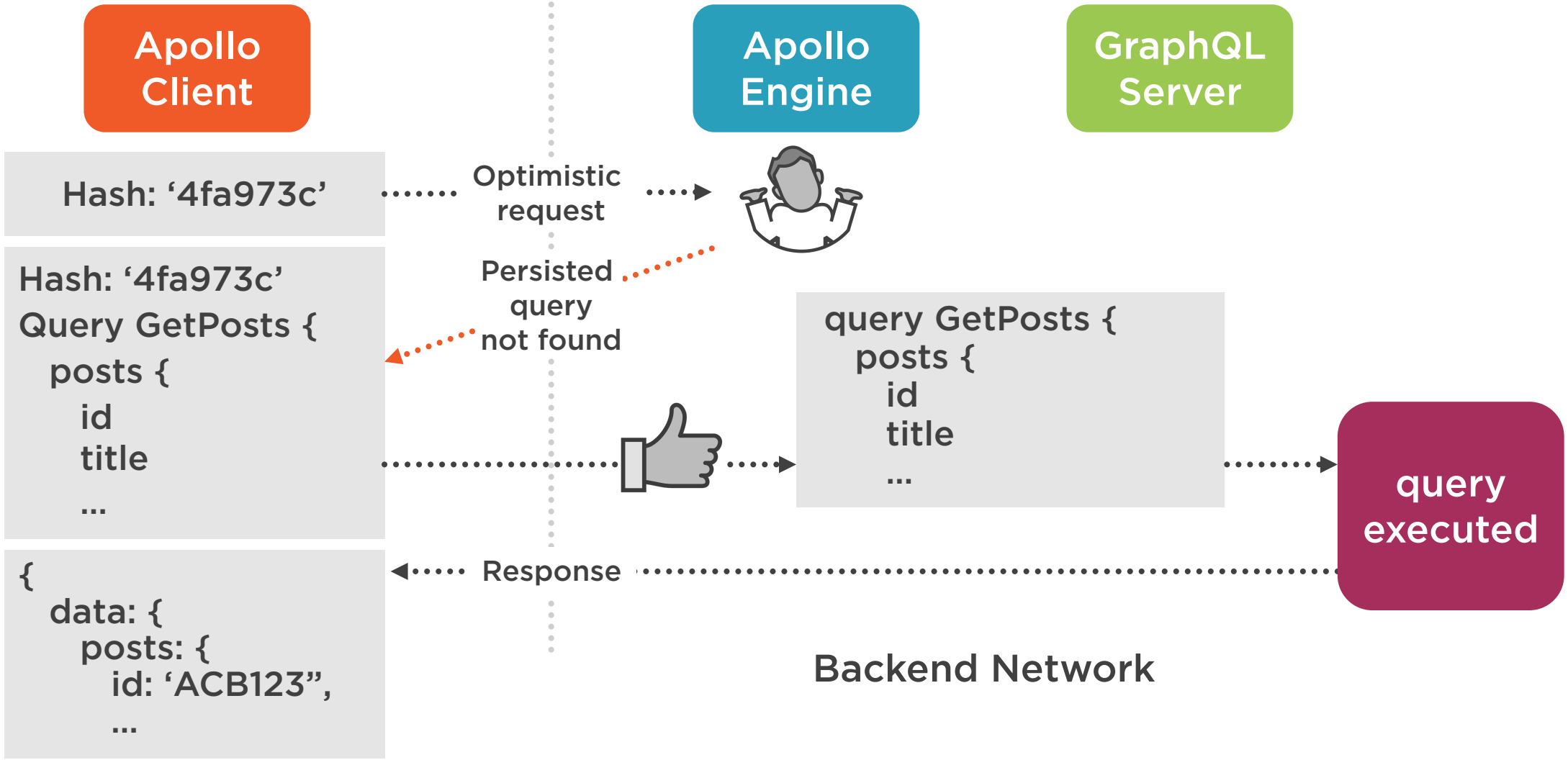
```
sha256Hash":"f8b4e2b2d68..."
```

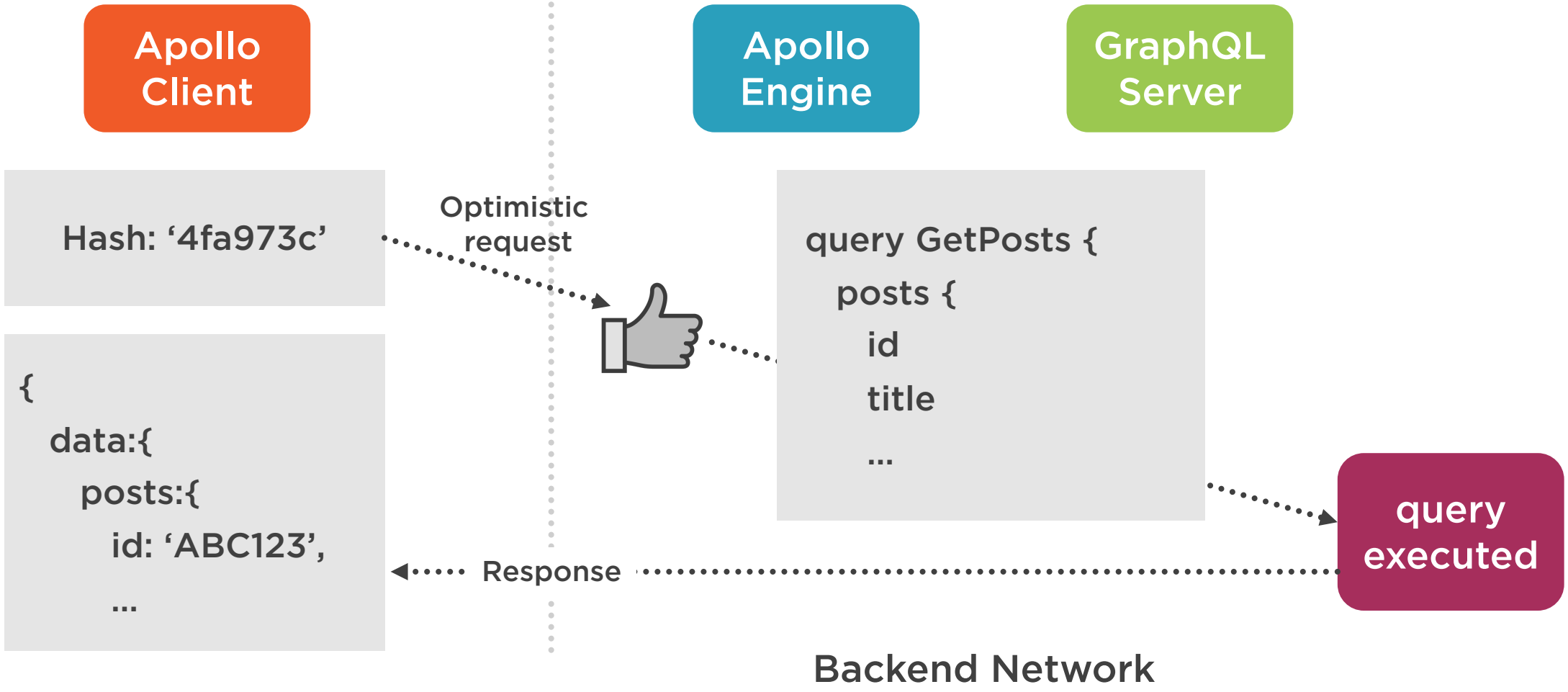
```
speakers(offset: 5, limit: 77) {
  datalist {
    id
    first
    last
    favorite…
```

NOT

```
sha256Hash":"f8b4e2b2d68..."
```

Automatic Persisted Queries: New Query Path




Apollo Client
Apollo Engine
GraphQL Server
Hash: '4fa973c'
Optimistic request
query GetPosts {
posts {
id
title
...
{
data:{
posts:{
id: 'ABC123',
...
Response
query executed
Backend Network

Apollo GraphQL queries over HTTP GET can be cached with a CDN.