GraphQL at the Edge

Jake Dawkins

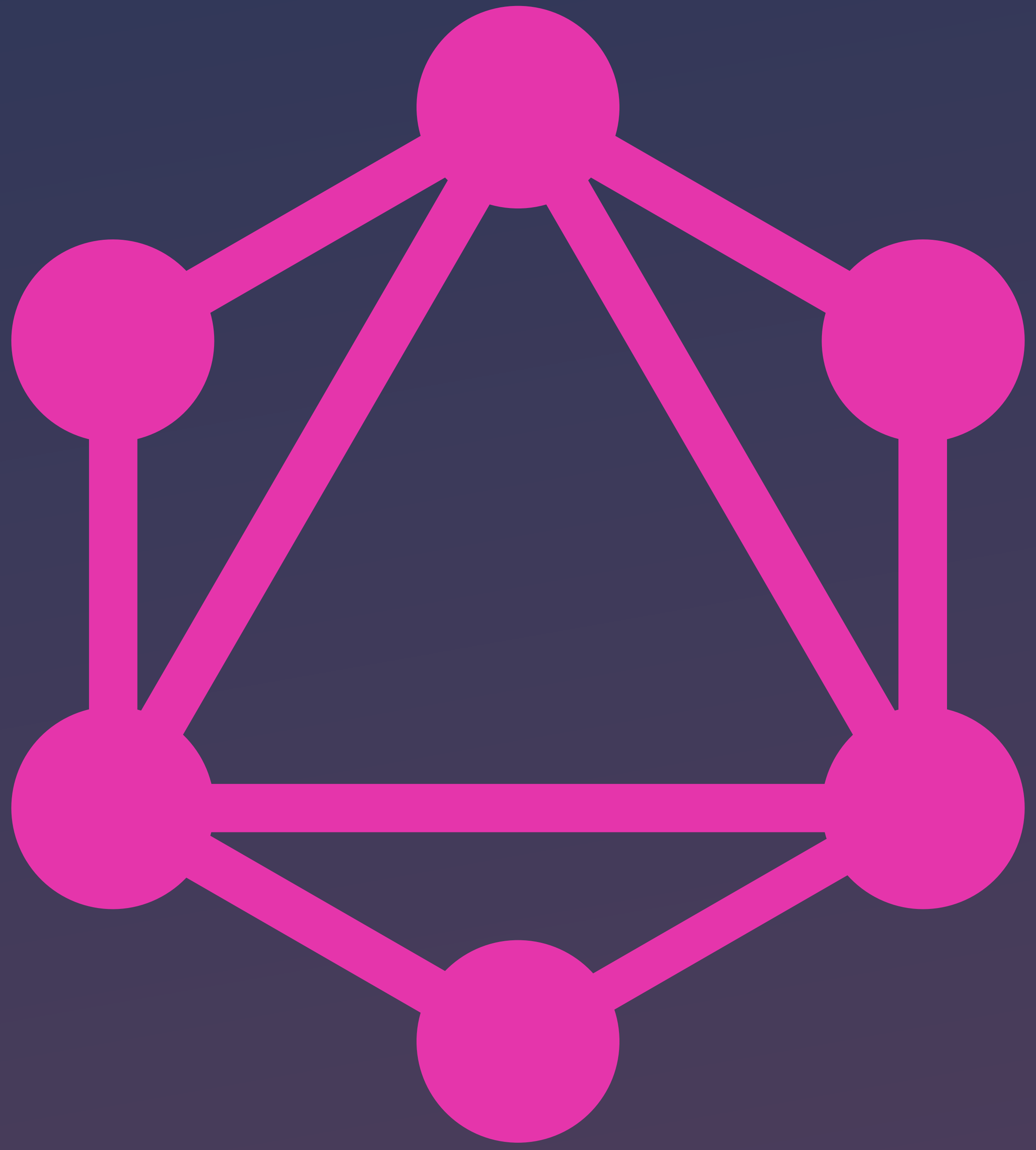
Apollo Architect

@jakedawkins

APOLLO

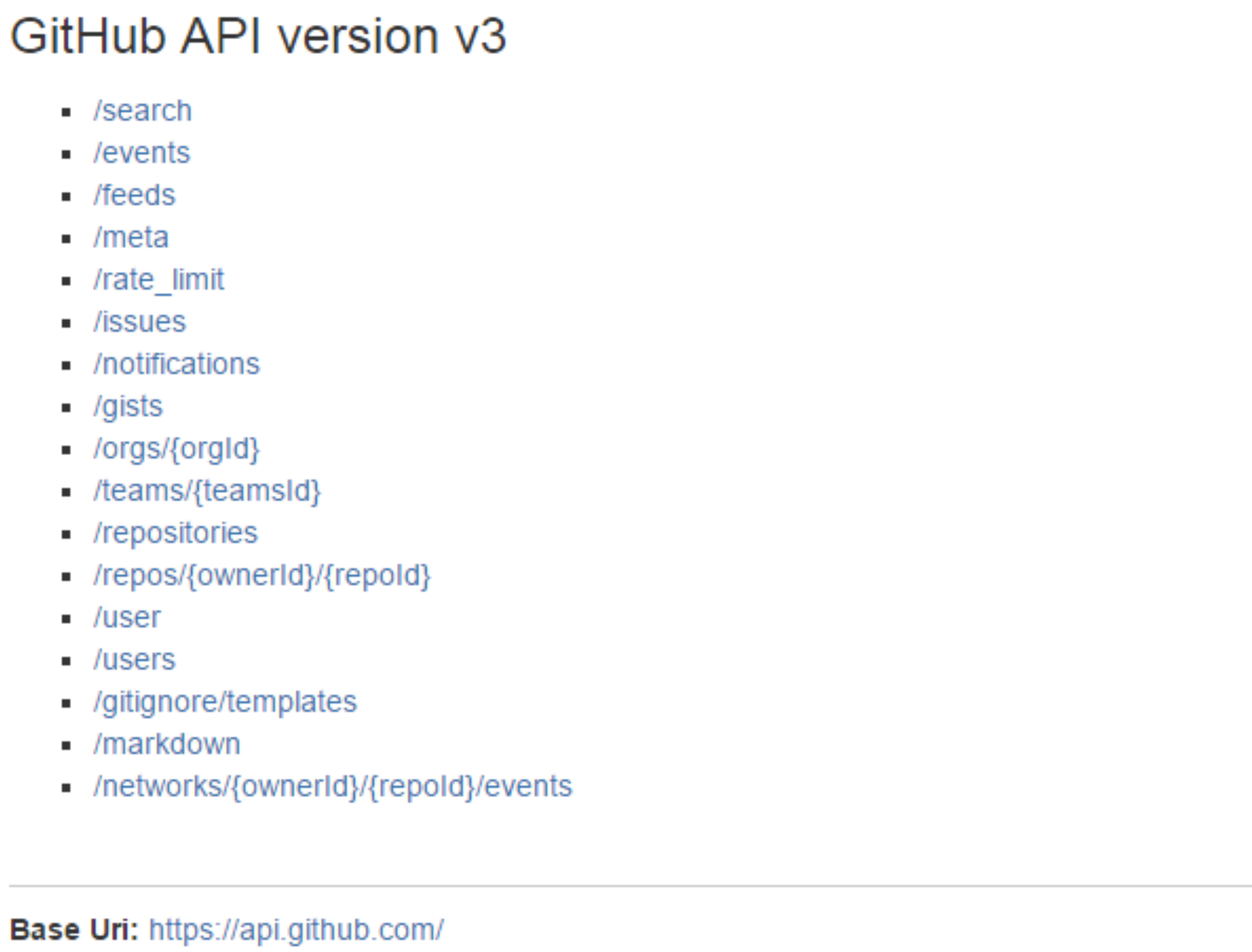
www.apollographql.com

GitHub
MLS
The New York Times
AMERICAN EXPRESS
CNN
EXPRESS
intuit
airbnb
OpenTable
ticketmaster
yelp
twitch
coursera
Product Hunt
Pinterest
shopify
UDACITY
@WalmartLabs
Expo
credit karma
heroku



GraphQL

What would an ideal API for apps look like?



GitHub API version v3

- /search
- /events
- /feeds
- /meta
- /rate_limit
- /issues
- /notifications
- /gists
- /orgs/{orgId}
- /teams/{teamsId}
- /repositories
- /repos/{ownerId}/{repoId}
- /user
- /users
- /gitignore/templates
- /markdown
- /networks/{ownerId}/{repoId}/events

Base Uri: https://api.github.com/

Overfetching

CLIENT

SERVER

GET users/apollographql/repos

```
[
  {
    "id": 67563416,
    "name": "android-docs",
    "full_name": "apollostack/android-docs",
    "owner": {
      "login": "apollostack",
      "id": 17189275,
      "avatar_url": "https://avatars.githubusercontent.com/u/17189275?v=3",
      "gravatar_id": "",
      "url": "https://api.github.com/users/apollostack",
      "html_url": "https://github.com/apollostack",
      "followers_url": "https://api.github.com/users/apollostack/followers",
      "following_url": "https://api.github.com/users/apollostack/following{/other_user}",
      "gists_url": "https://api.github.com/users/apollostack/gists{/gist_id}",
      "starred_url": "https://api.github.com/users/apollostack/starred{/owner}{/repo}",
      "subscriptions_url": "https://api.github.com/users/apollostack/subscriptions",
      "organizations_url": "https://api.github.com/users/apollostack/orgs",
      "repos_url": "https://api.github.com/users/apollostack/repos",
      "events_url": "https://api.github.com/users/apollostack/events{/privacy}",
      "received_events_url": "https://api.github.com/users/apollostack/received_events",
      "type": "Organization",
      "site_admin": false
    },
    "private": false,
    "html_url": "https://github.com/apollostack/android-docs",
```

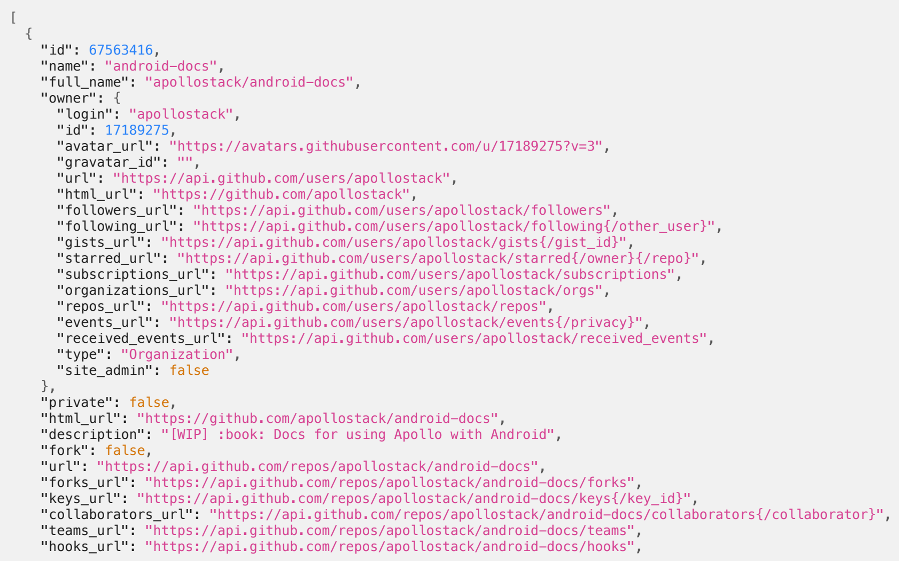
```
{
  "id": 67563416,
  "name": "android-docs",
  "full_name": "apollostack/android-docs",
  "owner": {
    "login": "apollostack",
    "id": 17189275,
    "avatar_url": "https://avatars.githubusercontent.com/u/17189275?v=3",
    "gravatar_id": "",
    "url": "https://api.github.com/users/apollostack",
    "html_url": "https://github.com/apollostack",
    "followers_url": "https://api.github.com/users/apollostack/followers",
    "following_url": "https://api.github.com/users/apollostack/following{/other_user}",
    "gists_url": "https://api.github.com/users/apollostack/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/apollostack/starred{/owner}{/repo}",
    "subscriptions_url": "https://api.github.com/users/apollostack/subscriptions",
    "organizations_url": "https://api.github.com/users/apollostack/orgs",
    "repos_url": "https://api.github.com/users/apollostack/repos",
    "events_url": "https://api.github.com/users/apollostack/events{/privacy}",
    "received_events_url": "https://api.github.com/users/apollostack/received_events",
    "type": "Organization",
    "site_admin": false
  },
  "private": false,
  "html_url": "https://github.com/apollostack/android-docs",
  "description": "[WIP] :book: Docs for using Apollo with Android",
  "fork": false,
  "url": "https://api.github.com/repos/apollostack/android-docs",
  "forks_url": "https://api.github.com/repos/apollostack/android-docs/forks",
```

Extra roundtrips

CLIENT

SERVER

GET users/apollographql/repos



```
[
  {
    "id": 67563416,
    "name": "android-docs",
    "full_name": "apollostack/android-docs",
    "owner": {
      "login": "apollostack",
      "id": 17189275,
      "avatar_url": "https://avatars.githubusercontent.com/u/17189275?v=3",
      "gravatar_id": "",
      "url": "https://api.github.com/users/apollostack",
      "html_url": "https://github.com/apollostack",
      "followers_url": "https://api.github.com/users/apollostack/followers",
      "following_url": "https://api.github.com/users/apollostack/following{/other_user}",
      "gists_url": "https://api.github.com/users/apollostack/gists{/gist_id}",
      "starred_url": "https://api.github.com/users/apollostack/starred{/owner}{/repo}",
      "subscriptions_url": "https://api.github.com/users/apollostack/subscriptions",
      "organizations_url": "https://api.github.com/users/apollostack/orgs",
      "repos_url": "https://api.github.com/users/apollostack/repos",
      "events_url": "https://api.github.com/users/apollostack/events{/privacy}",
      "received_events_url": "https://api.github.com/users/apollostack/received_events",
      "type": "Organization",
      "site_admin": false
    },
    "private": false,
    "html_url": "https://github.com/apollostack/android-docs",
    "description": "[WIP] :book: Docs for using Apollo with Android",
    "fork": false,
    "url": "https://api.github.com/repos/apollostack/android-docs",
    "forks_url": "https://api.github.com/repos/apollostack/android-docs/forks",
    "keys_url": "https://api.github.com/repos/apollostack/android-docs/keys{/key_id}",
    "collaborators_url": "https://api.github.com/repos/apollostack/android-docs/collaborators{/collaborator}",
    "teams_url": "https://api.github.com/repos/apollostack/android-docs/teams",
    "hooks_url": "https://api.github.com/repos/apollostack/android-docs/hooks",
```

GET repos/apollographql/apollo-client/contributors

[{"login":"stubailo","id":448783,"avatar_url":"https://avatars.githubusercontent.com/u/448783?v=3","gravatar_id":"","url":"https://api.github.com/users/stubailo","html_url":"https://github.com/stubailo","followers_url":"https://api.github.com/users/stubailo/followers","following_url":"https://api.github.com/

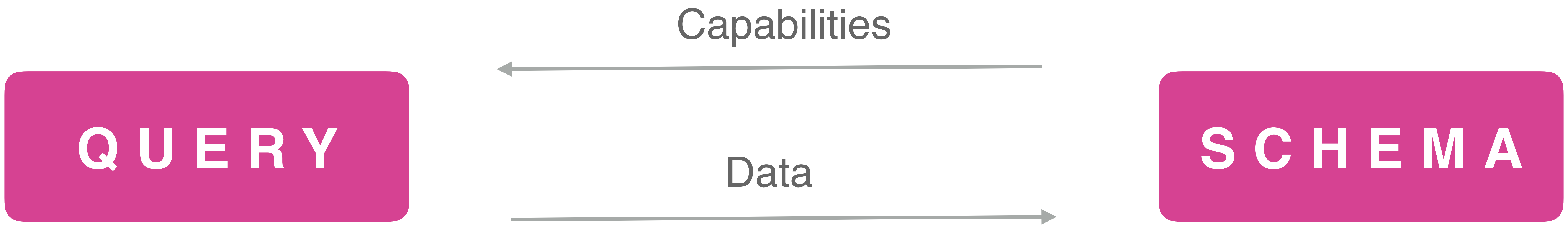
GET repos/apollographql/react-apollo/contributors

[{"login":"stubailo","id":448783,"avatar_url":"https://avatars.githubusercontent.com/u/448783?v=3","gravatar_id":"","url":"https://api.github.com/users/stubailo","html_url":"https://github.com/stubailo","followers_url":"https://api.github.com/users/stubailo/followers","following_url":"https://api.github.com/

GET repos/apollographql/apollo-ios/contributors

[{"login":"stubailo","id":448783,"avatar_url":"https://avatars.githubusercontent.com/u/448783?v=3","gravatar_id":"","url":"https://api.github.com/users/stubailo","html_url":"https://github.com/stubailo","followers_url":"https://api.github.com/users/stubailo/followers","following_url":"https://api.github.com/

Let clients specify their own **data needs** against the **capabilities** exposed by the server



QUERY

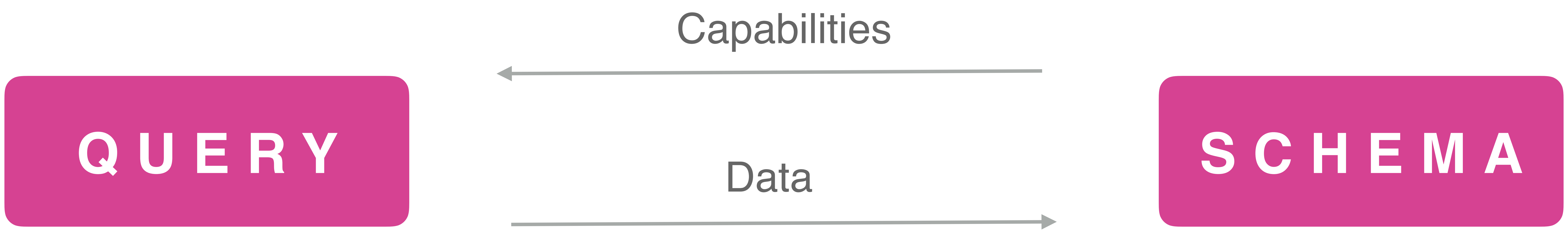
Capabilities

Data

SCHEMA

```
query {
  posts {
    title
  }
}
```

```
type Post {
  id: Int!
  title: String
  author: Author
  votes: Int
}
```



QUERY

Capabilities

Data

SCHEMA

```
query {
  posts {
    title
    author {
      firstName
    }
  }
}
```

```
type Post {
  id: Int!
  title: String
  author: Author
  votes: Int
}

type Author {
  id: Int!
  firstName: String
  lastName: String
  posts: [Post]
}
```

Not just about performance

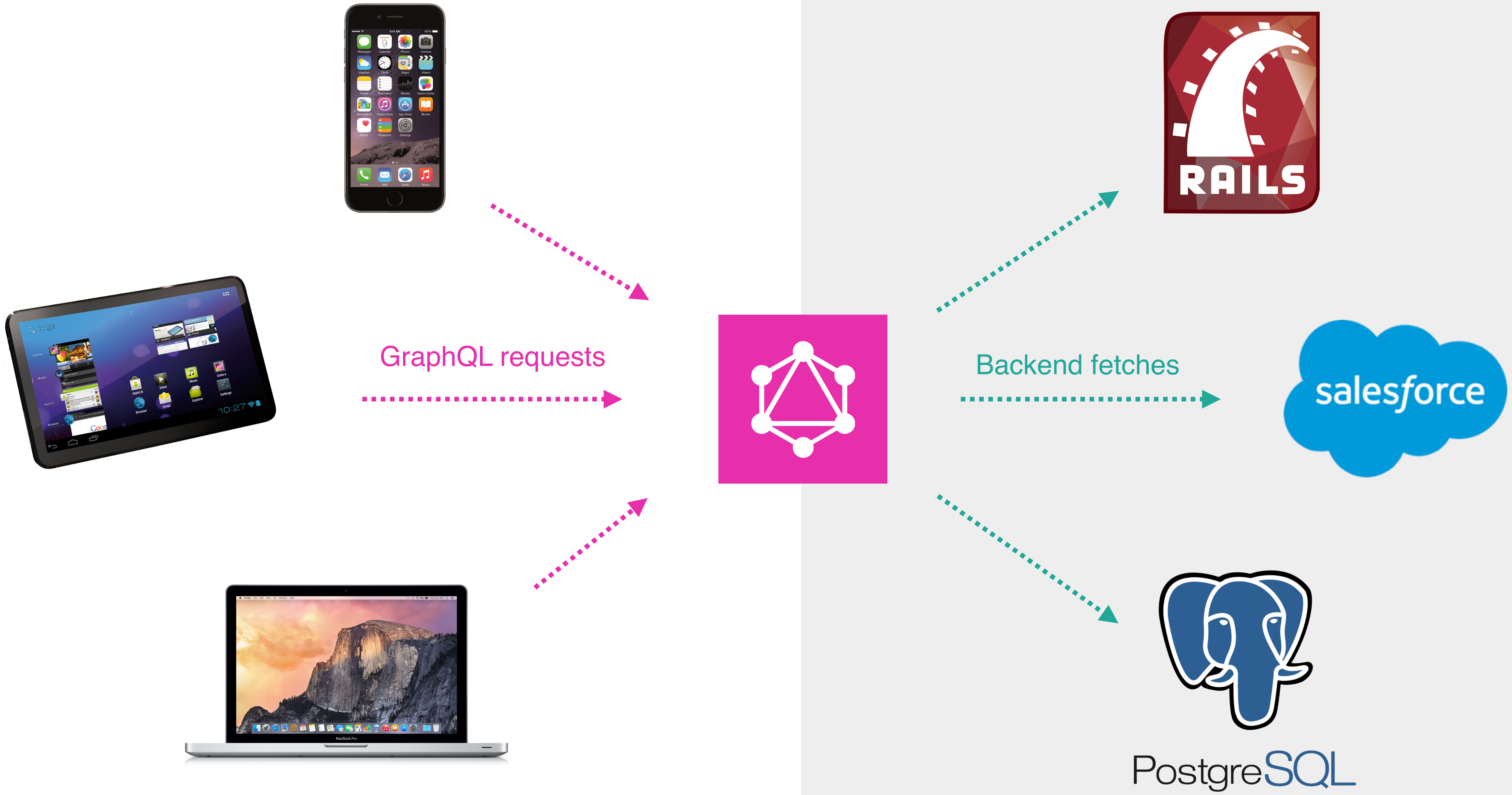
GraphQL has been designed
from the start with
developer experience in mind

APOLLO SERVER

You likely have existing backends you need to integrate with

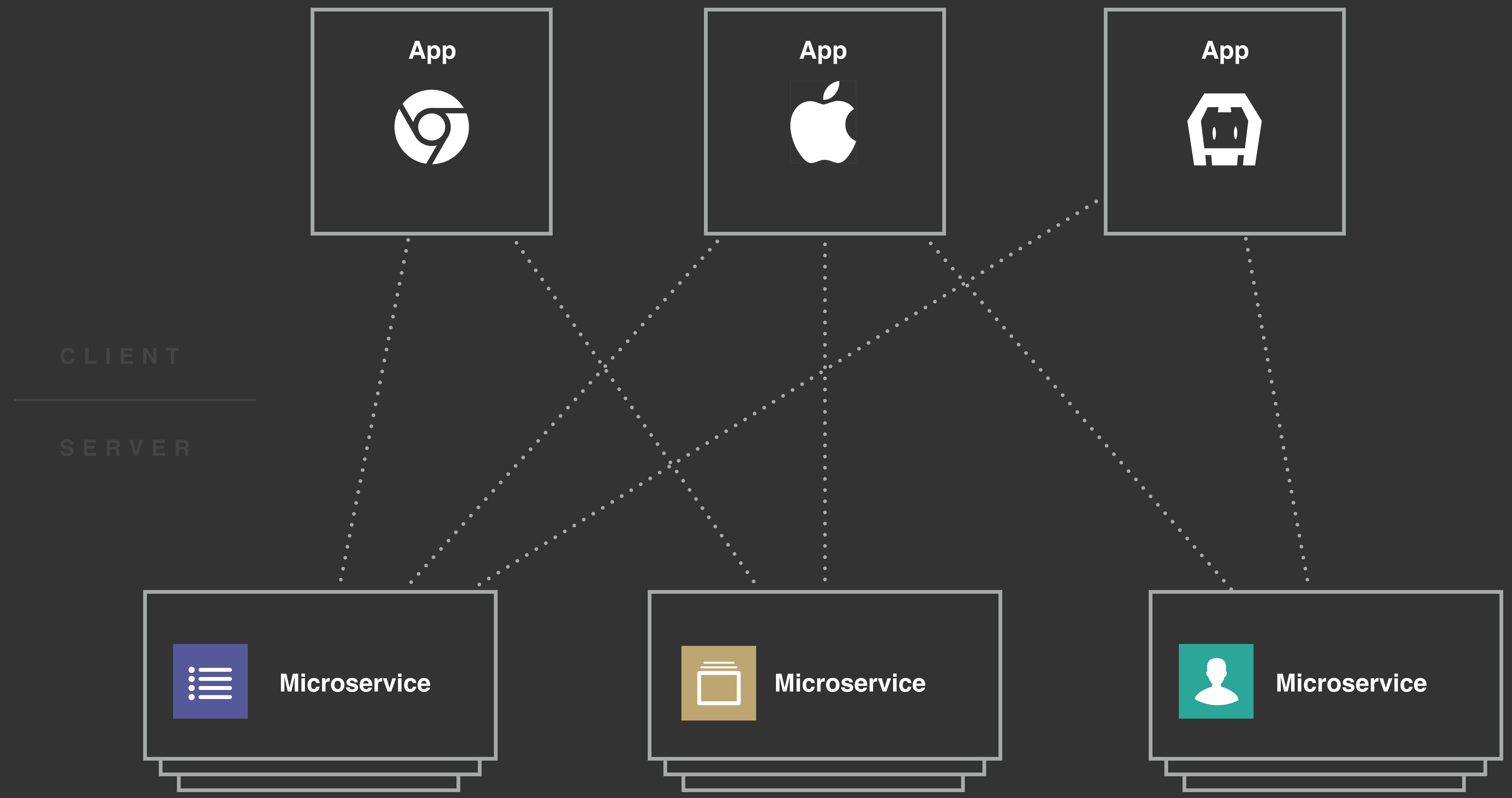
That’s a strength, not a weakness

GraphQL is at its strongest as a technology for building **gateways**

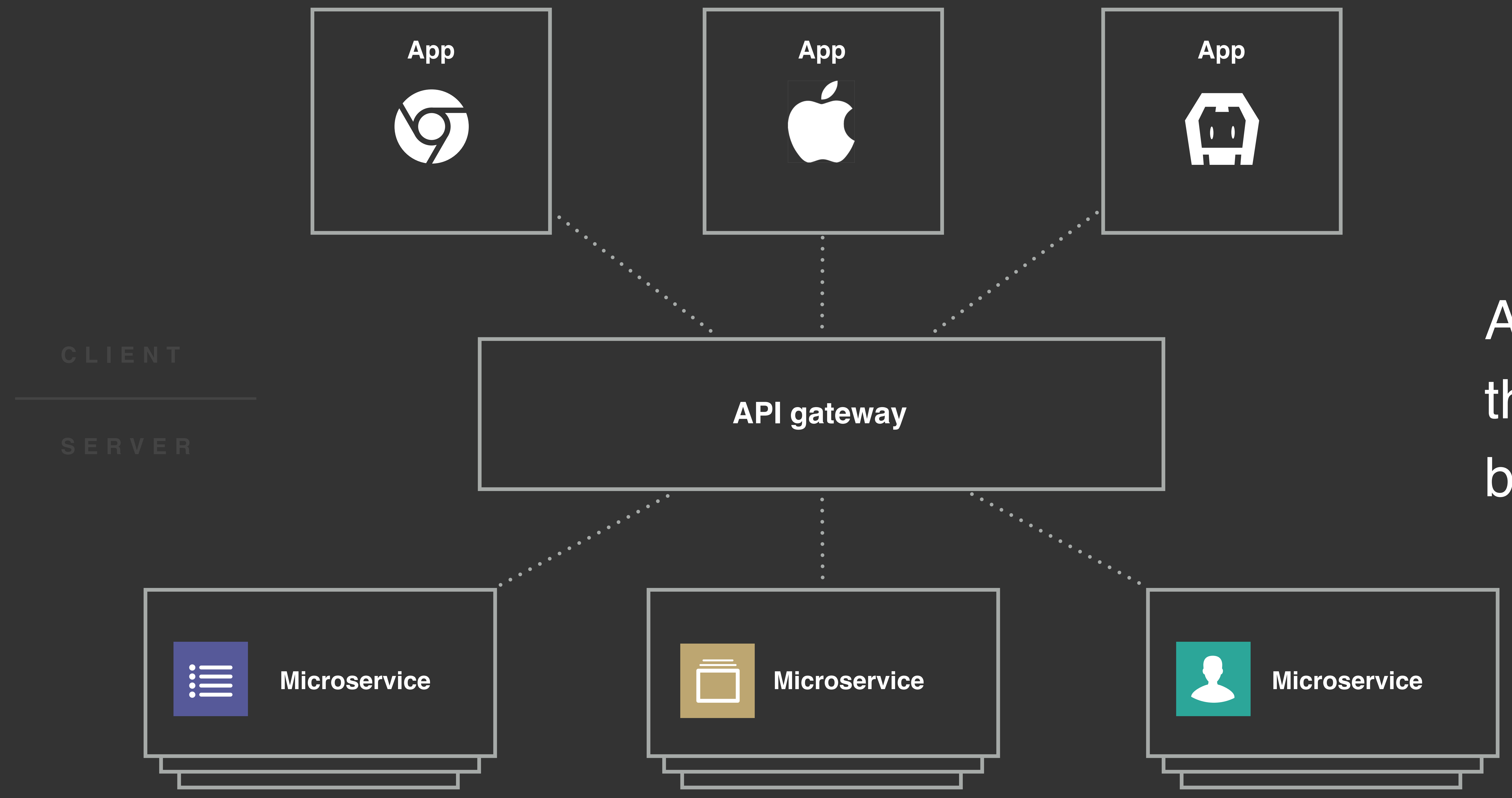


GraphQL requests

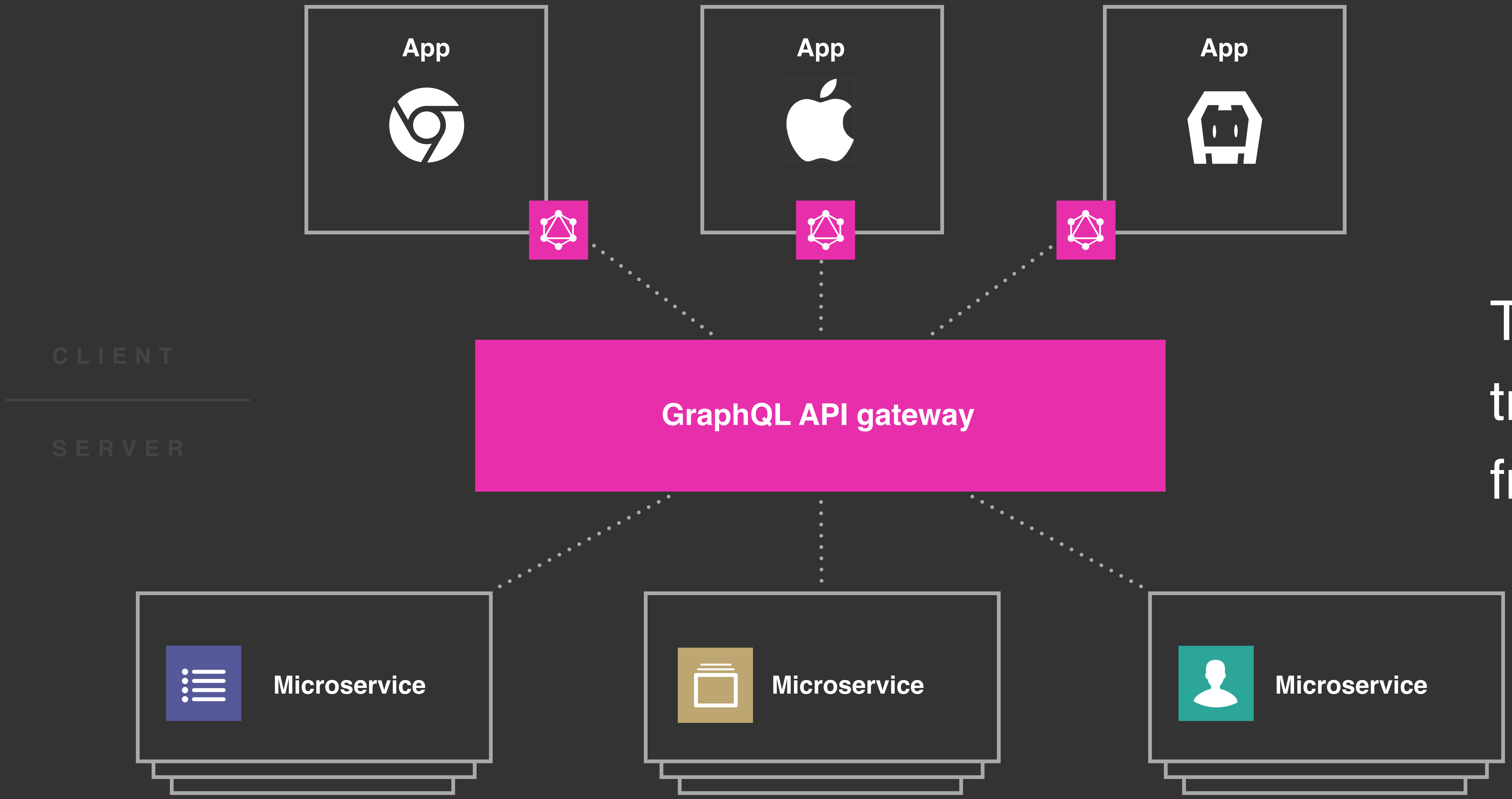
Backend fetches



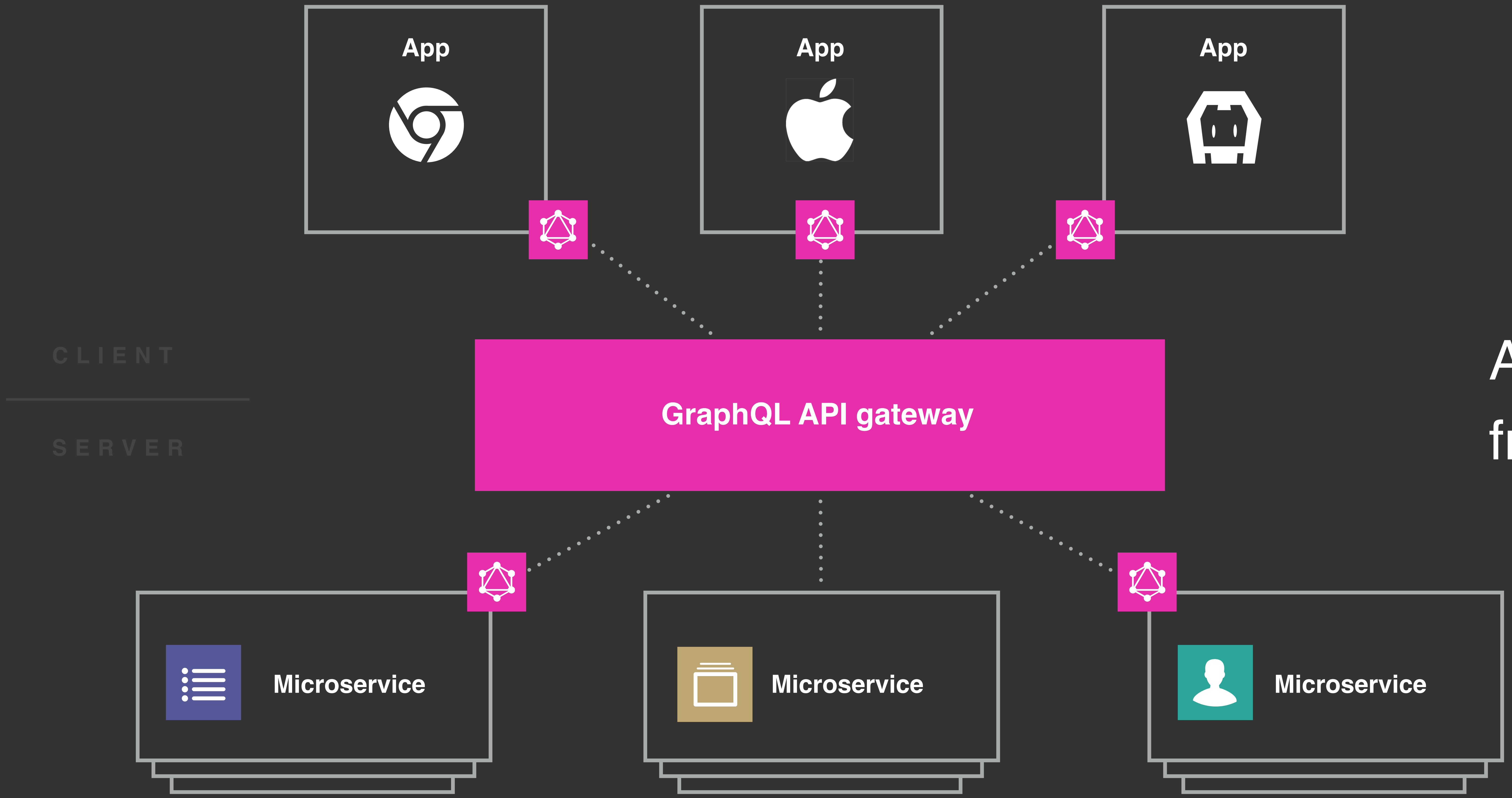
Splitting into microservices, waterfall loading and multiple APIs



Add an API gateway, now there’s a development bottleneck

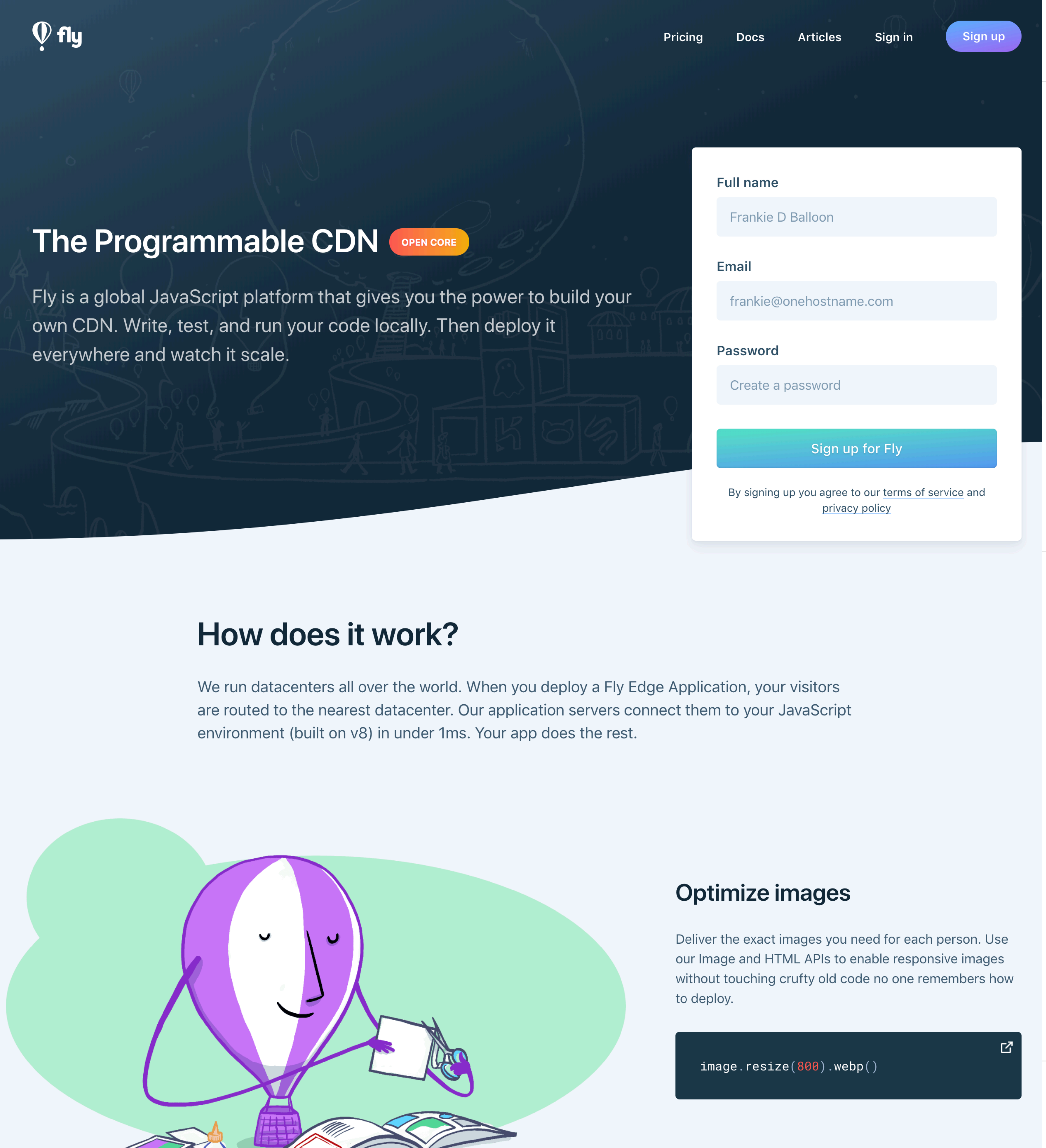


The solution: GraphQL as the translation layer between frontend and backend



Alternative: Stitch schemas from different services together

closedweather
SHOW ME WHAT YOU GOT!



The Programmable CDN

Fly is a global JavaScript platform that gives you the power to build your own CDN. Write, test, and run your code locally. Then deploy it everywhere and watch it scale.

Full name: Frankie D Balloon

Email: frankie@onehostname.com

Password: Create a password

By signing up you agree to our terms of service and privacy policy

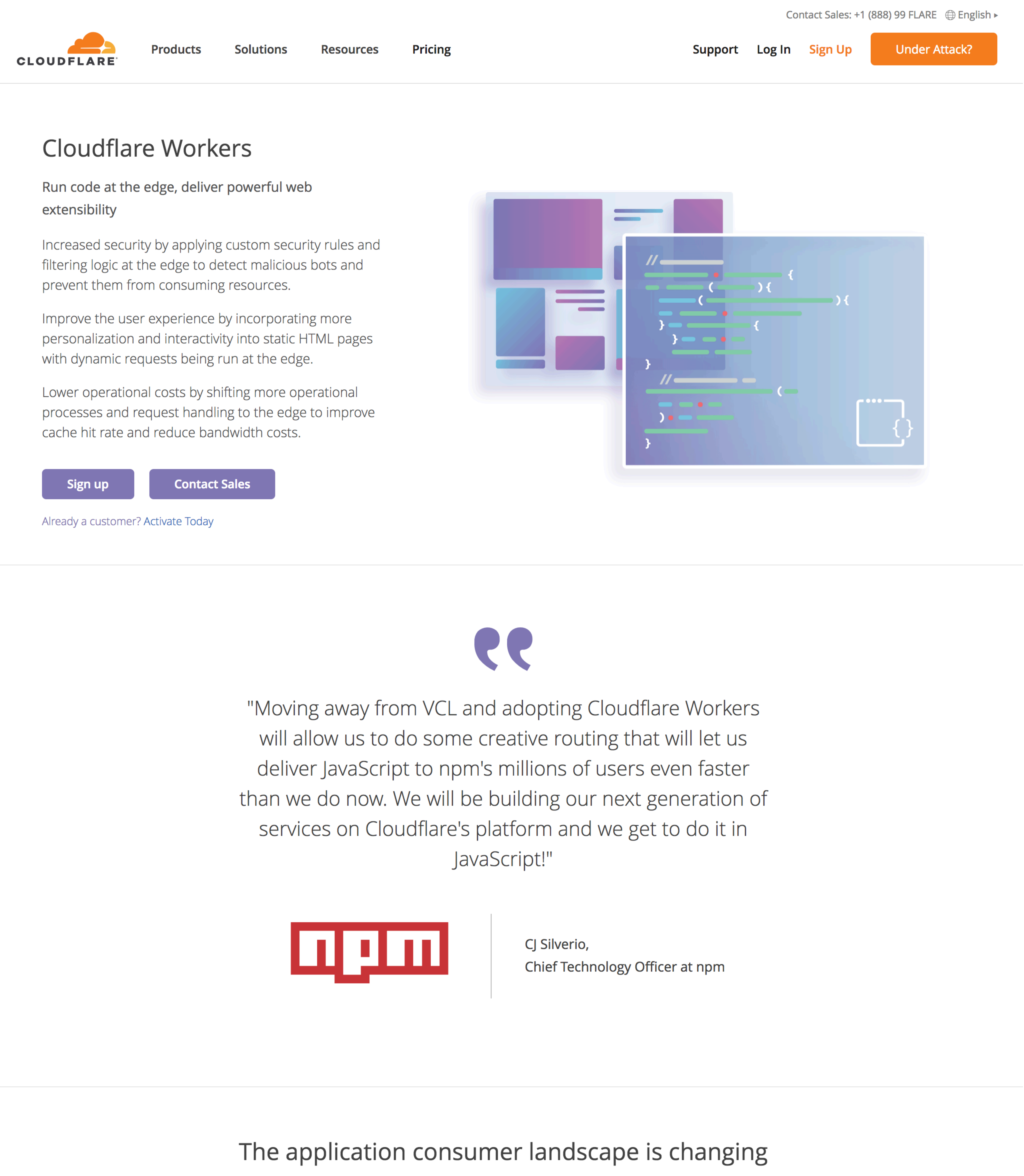
How does it work?

We run datacenters all over the world. When you deploy a Fly Edge Application, your visitors are routed to the nearest datacenter. Our application servers connect them to your JavaScript environment (built on v8) in under 1ms. Your app does the rest.

Optimize images

Deliver the exact images you need for each person. Use our Image and HTML APIs to enable responsive images without touching crufty old code no one remembers how to deploy.

```
image.resize(800).webp()
```



Cloudflare Workers

Run code at the edge, deliver powerful web extensibility

Increased security by applying custom security rules and filtering logic at the edge to detect malicious bots and prevent them from consuming resources.

Improve the user experience by incorporating more personalization and interactivity into static HTML pages with dynamic requests being run at the edge.

Lower operational costs by shifting more operational processes and request handling to the edge to improve cache hit rate and reduce bandwidth costs.

Already a customer? Activate Today

> "Moving away from VCL and adopting Cloudflare Workers will allow us to do some creative routing that will let us deliver JavaScript to npm's millions of users even faster than we do now. We will be building our next generation of services on Cloudflare's platform and we get to do it in JavaScript!"

CJ Silverio,
Chief Technology Officer at npm

The application consumer landscape is changing

DEMO 🎉

CLOUDFLARE
Preview
Testing
Documentation
POST
https://tutorial.cloudfla
Headers
+ Add New Header
Body
Run Test
Console
Sources
7
Worker
Filter
Default levels
Group similar
Update (⌘s)
Share

Transform requests on the edge using JavaScript.

Learn more about Cloudflare Workers

```
!function(e,n){"object"==typeof exports&&"undefined"!=typeof module?n(exports):"function"==typeof

const { ApolloServer, gql } = this.apollo;

const server = new ApolloServer({
  typeDefs: gql`
    type Query {
      popularMovies: [Movie]!
    }

    type Movie {
      id: ID
      title: String
      overview: String
    }
  `,
  resolvers: {
    Query: {
      async popularMovies() {
        const response = await fetch("https://api.themoviedb.org/3/discover/movie?api_key=4e911a0
        const data = await response.json();
        return data.results;
      }
    },
  },
  enableIntrospection: true
});

server.listen()
```

```
status: 200 OK
content-length: 892
content-type: application/json

{"data":{"popularMovies":[{"id":"299536","title":"Avengers: Infinity War"},{"id":"337167"
```

Content-Type	application/json

```
Got request                                                        worker.js:10
Got response                                                       worker.js:12
Got request                                                        worker.js:10
Got response                                                       worker.js:12
Got request                                                        worker.js:10
Got response                                                       worker.js:12
Got request                                                        worker.js:10
Got response                                                       worker.js:12
Got request                                                        worker.js:10
Got response                                                       worker.js:12
⚠ Script modified; context reset.
4 ▶Called .text() on an HTTP body which does not appear to be text. The body's Content-Type is    worker.js:21
  "application/json;charset=utf-8". The result will probably be corrupted. Consider checking the Content-Type header before
  interpreting entities as text.
2 ▶Called .text() on an HTTP body which does not appear to be text. The body's Content-Type is    worker.js:21
  "application/json;charset=utf-8". The result will probably be corrupted. Consider checking the Content-Type header before
  interpreting entities as text.
>
```

I like what you got. Good job.
[adult swim]

We’re not done yet

APOLLO

www.apollographql.com

github.com/apollographql

@apollographql

@jakedawkins