

I 🙌 Francis

reconbot



I



BUSTLE

Hey Francis, How did everyone at
Bustle make the website so fast?

– People

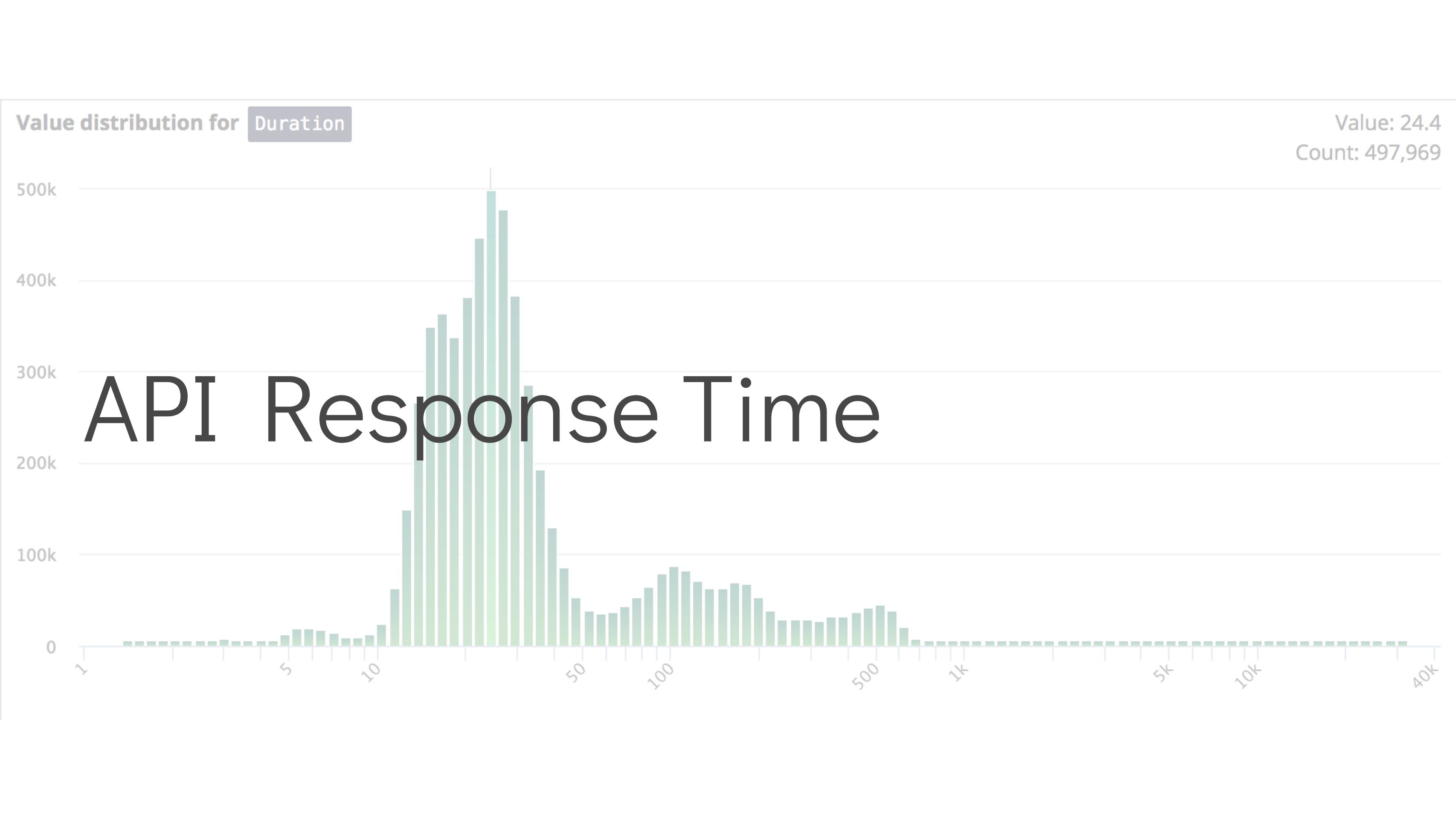
We put everything in Redis.

— Francis

We live in Memory

λ , Graphql, and Redis In Under 70ms

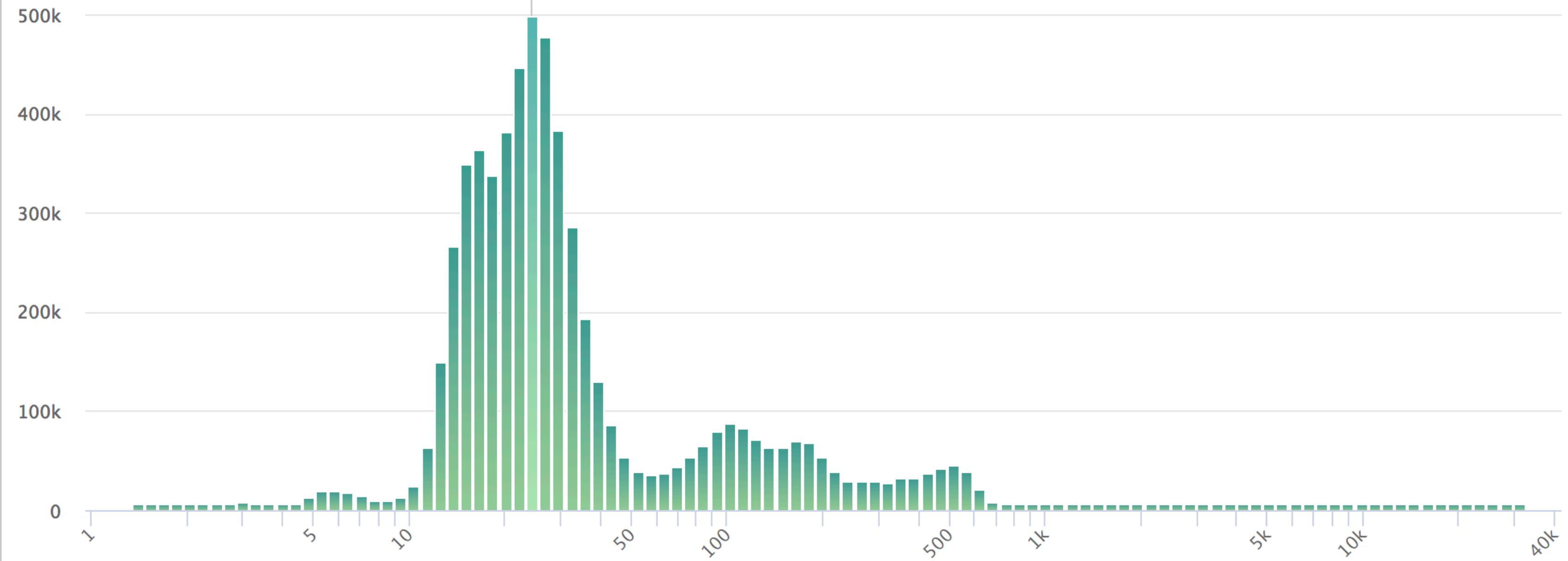
[https://github.com/reconbot/
we-live-in-memory](https://github.com/reconbot/we-live-in-memory)



Value distribution for Duration

Value: 24.4

Count: 497,969

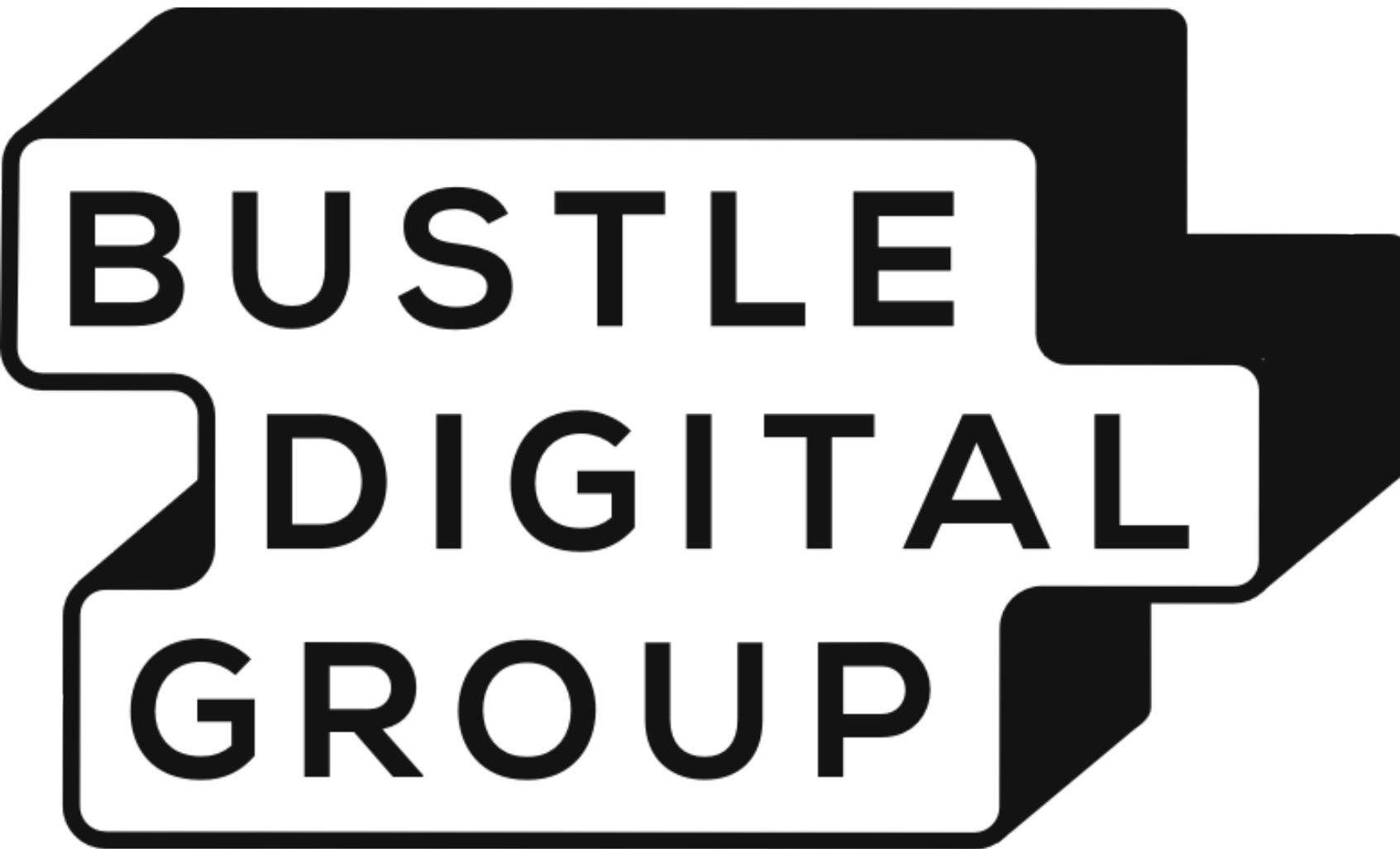


The Story

1. What is bustle doing
2. How is bustle doing it
3. How you can do it



Welcome To The New Authority



BDG is the largest
reaching publisher
for millennial women



bustle acquires



bustle acquires **elite daily**
bustle acquires **zoe report**
bustle acquires **gawker**

Google Search

I'm Feeling Lucky

Report inappropriate predictions

Platform Goals

- Best reader experience possible
- Best features for our writers and designers

Just like you 

Platform Strategies

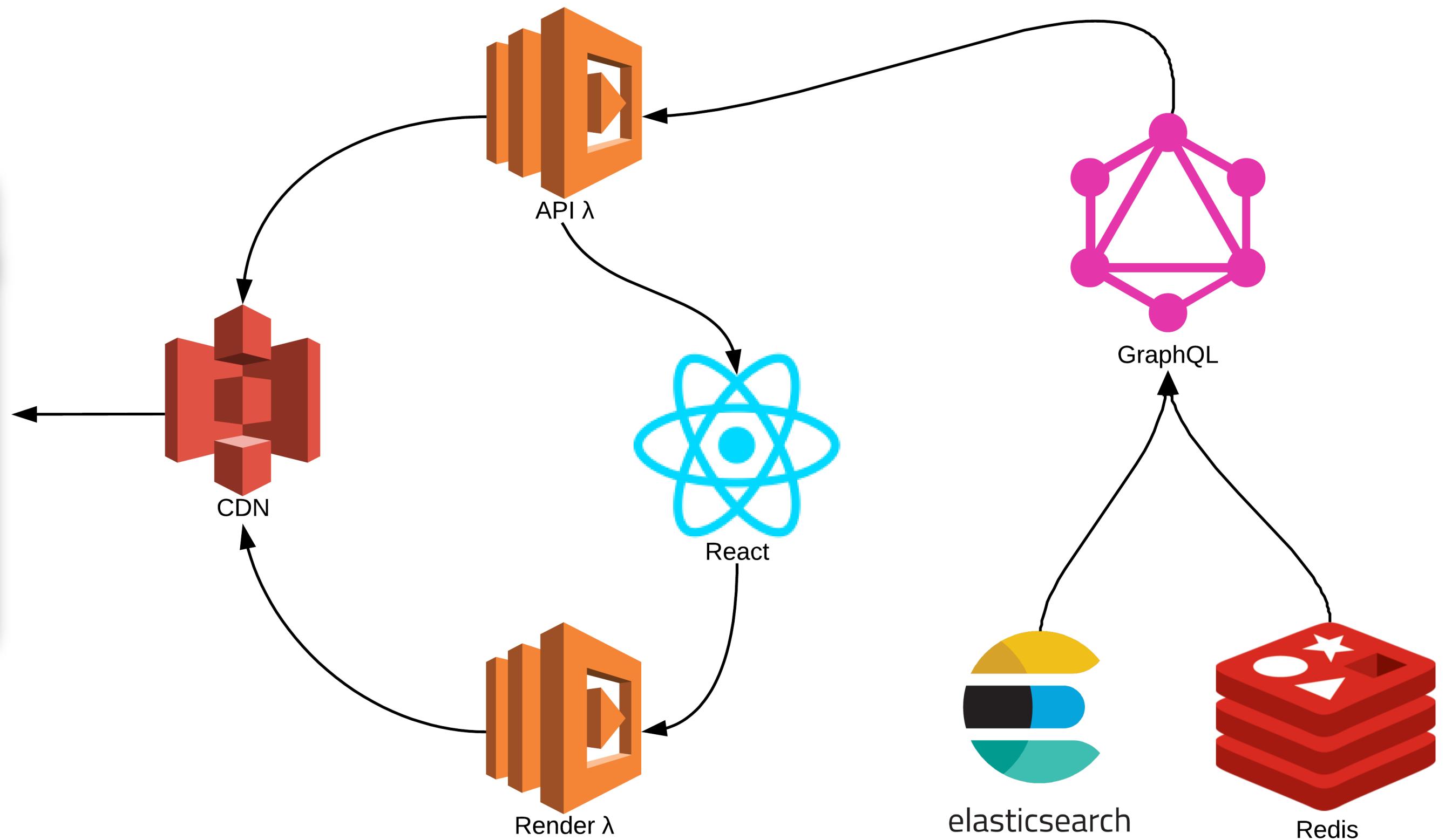
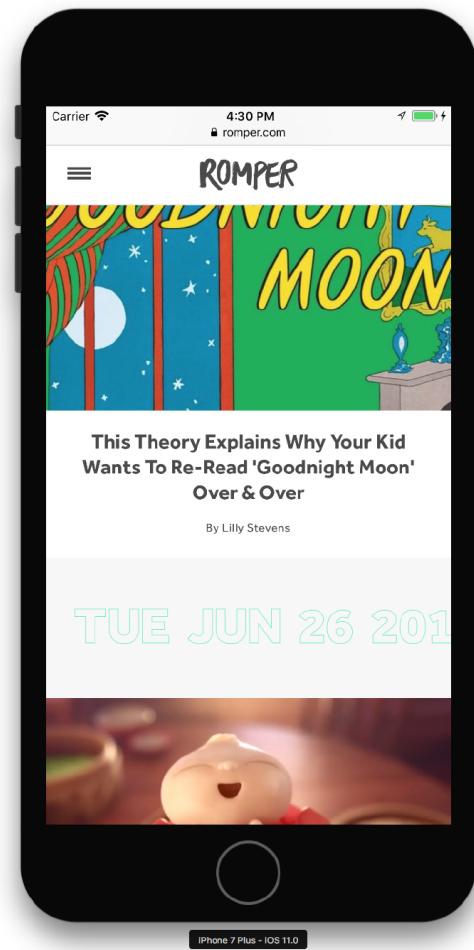
- Fastest page load
- Reduce the cost of change

What is AWS Lambda (λ)?

Why are we using lambda?

Ok Francis, but how does it work?

– Get to it already



Layers

1. CDN
2. Rendering
3. API
4. Database

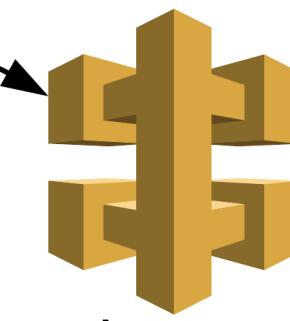
CDN

Takes an HTTP request and try to give a response out of cache, try really hard.

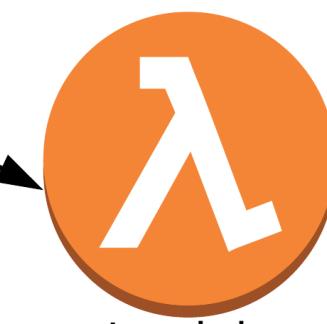
- API Gateway / Cloudfront
- Cloudflare
- Fastly

The CDN
Should Execute my Functions

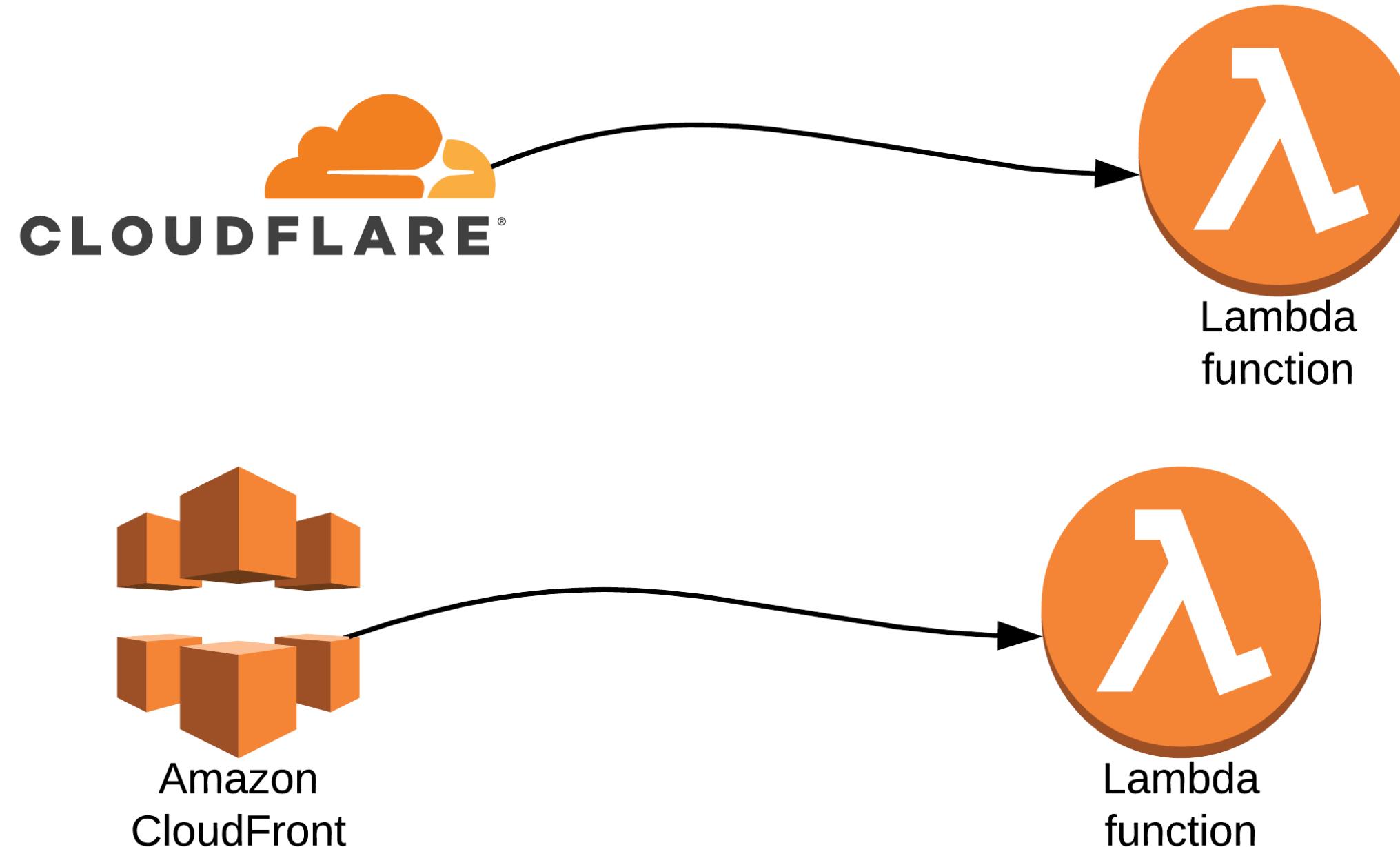
fastly.



Amazon
API
Gateway



Lambda
function

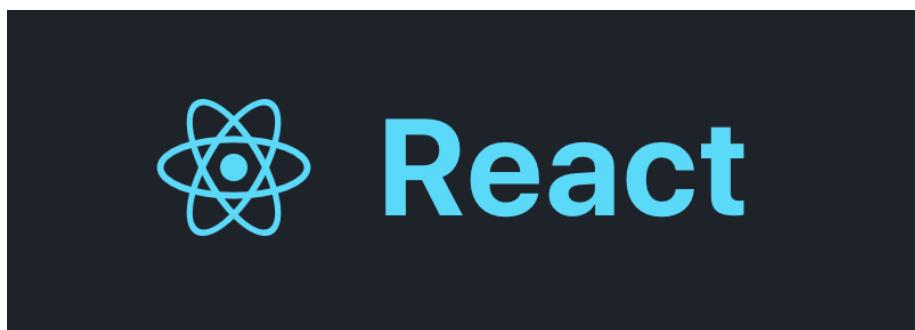


Rendering

Take an HTTP request, fetch data, and render out { status, body, headers }

- Server Side Render a page
- Client Side Render a page
- Be smart about Loading Stylesheets and Components

Rendering



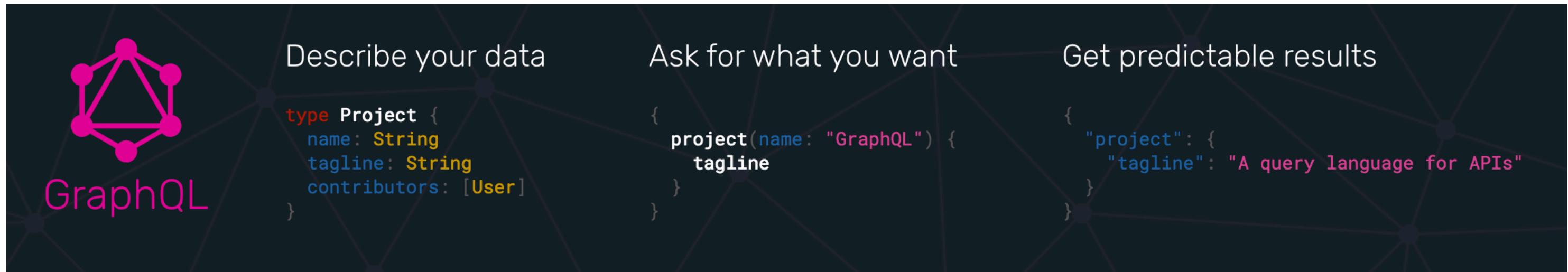
API

- Take a query and return some data
- Take some input and change state, then return some data
- Be strict about what types we return

API



API



Database

- Store the data safely
- Retrieve the data fast

Database

- Store the data safely
- Retrieve the data fast, in a consistent time

Redis is our Primary Data Store

Redis is an in-memory data
structure server. It supports,
hashes, lists, sets, sorted sets
with range queries.

- Redis.io (kinda)

Hey Francis,

Isn't that really dangerous?

– 50/50 chance you'll say this

No.

– 100% chance I'll say this

I don't believe you.

– 99% chance you're thinking this

Redis Persistance

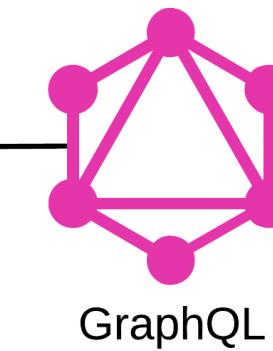
- 1s fsync of Append Only File (AOF)
- 1 hour snapshot of the Redis Database File (RDB) backed up to S3
- Read replicas ready to take over really fast
- Perfect for our read heavy load

This could be any database

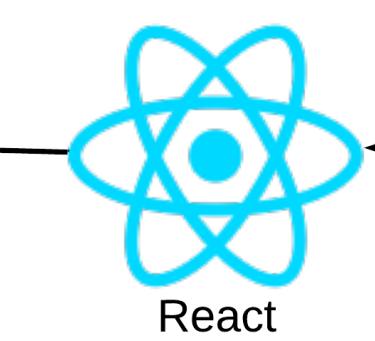
DATA LAYER



API LAYER

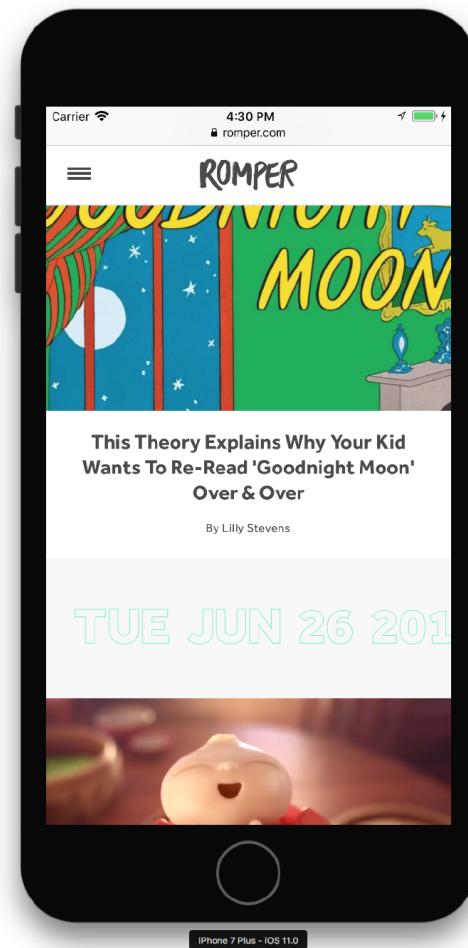


RENDER LAYER



CDN LAYER



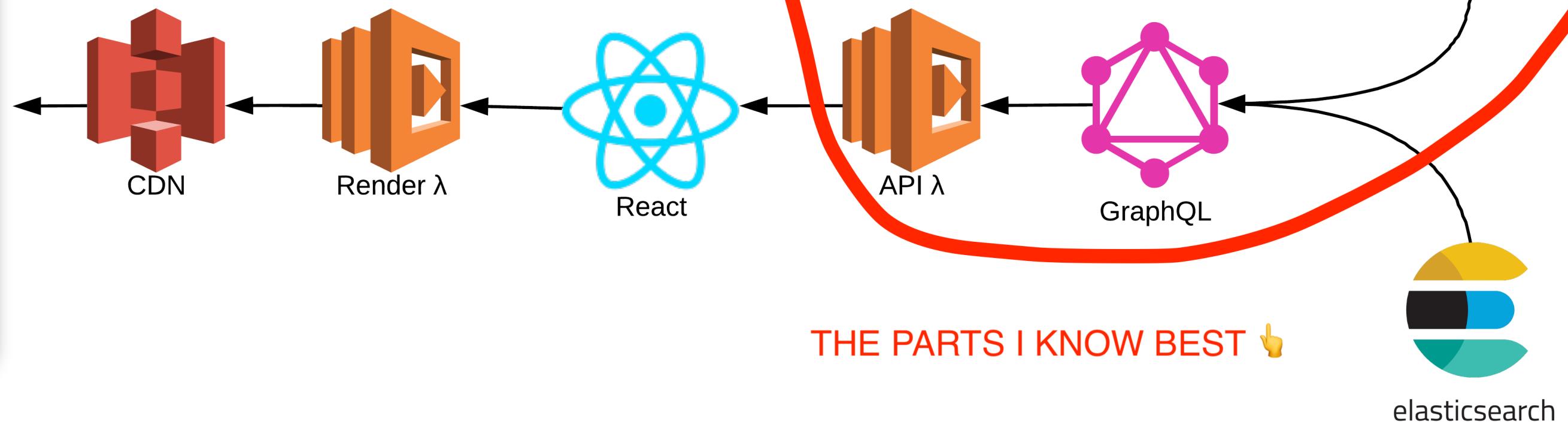


CDN
LAYER

RENDER
LAYER

API
LAYER

DATA
LAYER





how can I pull off dad shoes?|



Google Search

I'm Feeling Lucky

how can I pull off dad shoes? - X +

https://www.google.com/search?source=hp&ei=ZJg9W8OVGaKQ_Qb...

I Wore 'Dad Shoes' For A Week & They Were SO Much Cooler Than I ...

<https://www.bustle.com/.../i-wore-dad-shoes-for-a-week-they-were-so-much-cooler-th...> ▾

Mar 15, 2018 - My two reference points for the perfect **dad shoes** were a stylish Ryan ... I ended up pulling a blazer from the back of my closet and wearing a ...

'Dad' trainers are everywhere, but would you wear them?

<https://www.harpersbazaar.com/uk/fashion/.../chunky-dad-grandpa-trainers-trend/> ▾

Feb 7, 2018 - Spongy gym **shoes** are fast becoming the fashion trainer of choice. Here's how to pull them off.

8 "Dad Style" Moves That Are Actually Really Cool | GQ

<https://www.gq.com/story/dad-style-is-actually-cool> ▾

Jun 15, 2016 - These days, add "dad" to anything—dad jeans, dad bod, dad style—and you're implying that it's not hip, hopelessly **out of touch**. Then again ...

I Wore 'Dad Shoes' For A Week & They Were SO Much Cooler Than I Thought They'd Be

By DALE ARDEN CHONG | Mar 15 2018

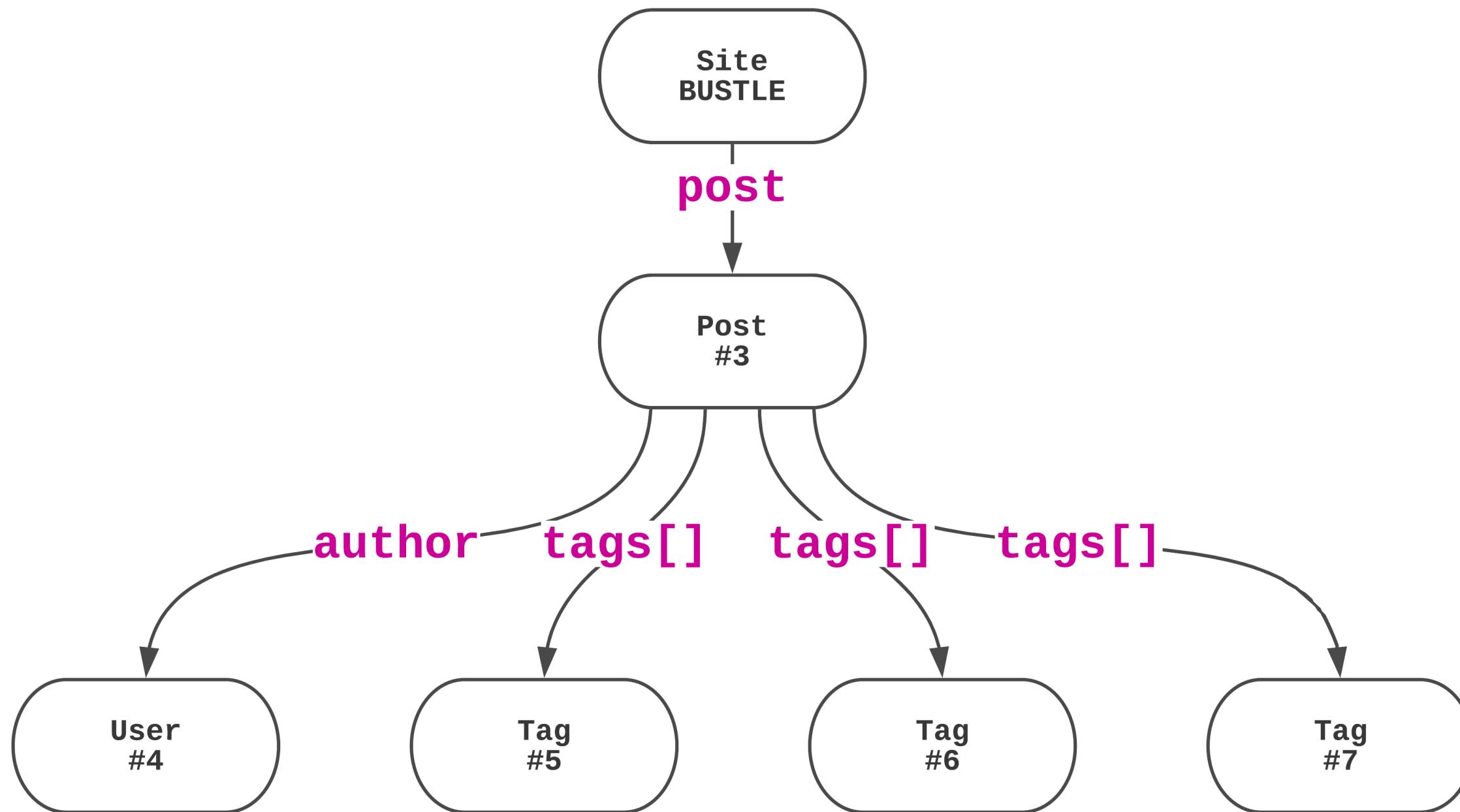
f

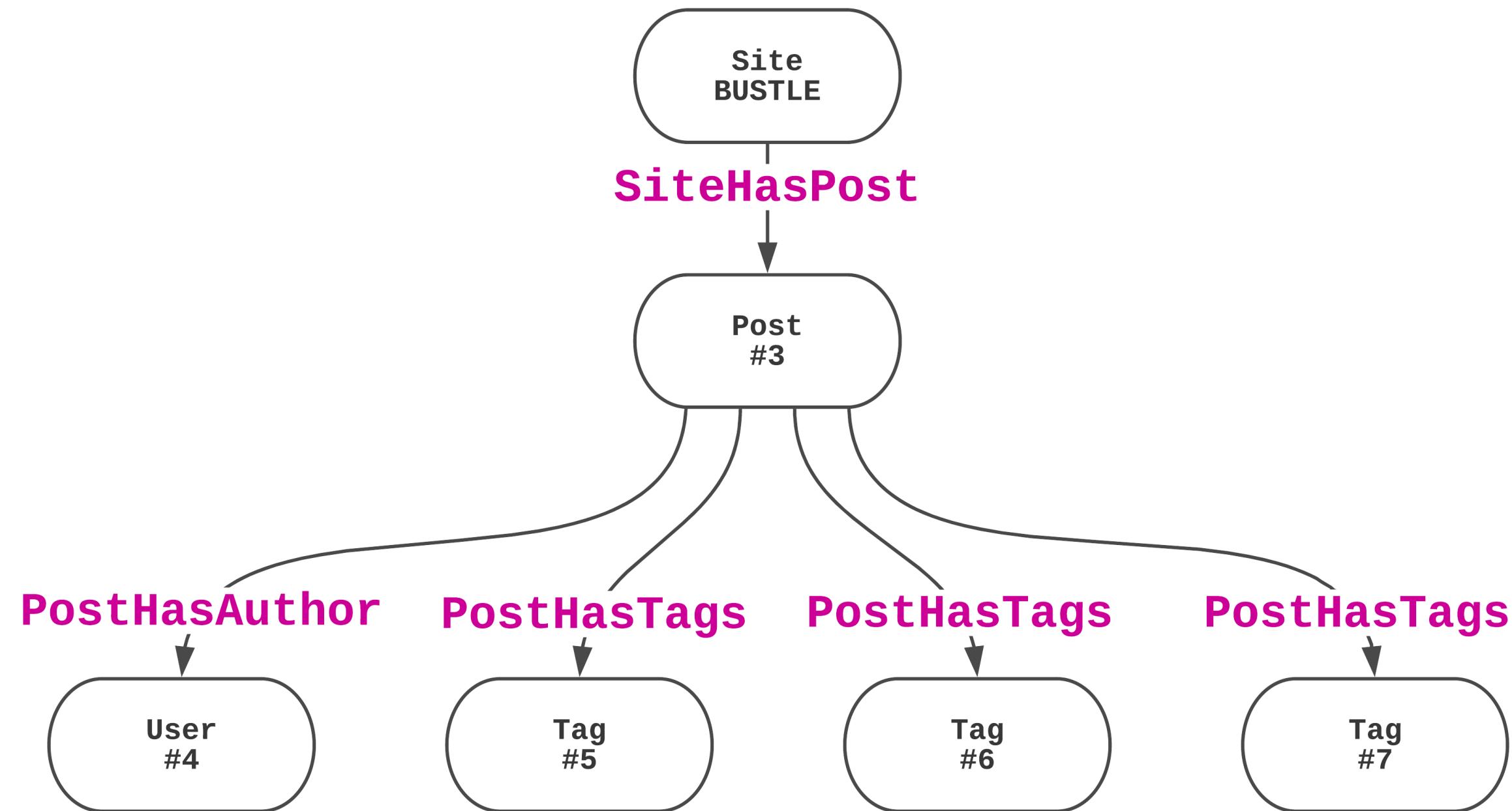


```
{  
  "site": {  
    "name": "BUSTLE",  
    "__typename": "Site",  
    "post": {  
      "id": "3",  
      "__typename": "Post",  
      "title": "I Wore 'Dad Shoes' For A Week & They Were SO Much Cooler Than I Thought",  
      "path": "/p/i-wore-dad-shoes-for-a-week-they-were-so-much-cooler-3",  
      "body": "99% of the JSON you'd be looking at, HI JSCONF!",  
      "author": {  
        "id": "4",  
        "__typename": "User",  
        "name": "Dale Arden Chong"  
      },  
      "tags": [  
        { "id": "5", "__typename": "Tag", "name": "homepage" },  
        { "id": "6", "__typename": "Tag", "name": "fashion" },  
        { "id": "7", "__typename": "Tag", "name": "freelancer" }  
      ]  
    }  
  }  
}
```

```
query postByPath {  
  site(name: BUSTLE) {  
    name  
    post(path: "/p/i-wore-dad-shoes-for-a-week-they-were-so-much-cooler-3") {  
      id  
      __typename  
      title  
      path  
      body  
      author {  
        id  
        __typename  
        name  
      }  
      tags {  
        id  
        __typename  
        name  
      }  
    }  
  }  
}
```

```
{  
  "site": {  
    "name": "BUSTLE",  
    "__typename": "Site",  
    "post": {  
      "id": "3",  
      "__typename": "Post",  
      "author": {  
        "id": "4",  
        "__typename": "User"  
      },  
      "tags": [  
        { "id": "5", "__typename": "Tag" },  
        { "id": "6", "__typename": "Tag" },  
        { "id": "7", "__typename": "Tag" }  
      ]  
    }  
  }  
}
```





We built our own Graph
Database



Oh you mean, like Neo4j right?

– 99% of you

Sure, but it's faster and doesn't do any
of the same things.

– Francis

Trains aren't slow they have one speed,
people get on & off, the people are slow.

Databases aren't slow they have one
speed, data goes in and out, the
queries are slow.

- Ikai (a DBA, who I guess never took the subway)

It's all about tradeoffs

Bustle Traded Query Flexibility
for Speed

Bustle **loves** to share & fund
Open Source

Konami präsentiert

NEMESIS

NEMESIS™

© 1985 Konami

Weltbekannt unter dem Namen GRADIUS
Schon lange erwartet – jetzt endlich auch in Deutschland
Seit Monaten in Japan ein Riesenerfolg
Garant für ausgezeichnete – vor allem aber beständige Kassen
Weltraum-action mit unendlichen Spielvariationen
Verkauf nur über den Fachgroßhandel



install

```
➤ npm i nemesis-db
```

- Really Fast Reads (0-1ms)
- Node and Edge Schemas with Types and Interfaces
- Weighted and Labeled Edges with Scanning
- Compression
- Still porting, not done yet!

Lets Make a GraphQL Server

1. The tools
2. The schema
3. The resolvers
4. The loaders

Tools

```
{  
  "dependencies": {  
    "apollo-server": "^2.0.5",  
    "apollo-server-lambda": "^2.0.4",  
    "graphql": "^0.13.2",  
    "nemesis-db": "^1.3.0-0"  
  }  
}
```

apollo-server

```
const { ApolloServer } = require('apollo-server')
const { readFileSync } = require('fs')
const resolvers = require('./resolvers')
const typeDefs = readFileSync('./lib/types.graphql', 'UTF8')

const server = new ApolloServer({ typeDefs, resolvers })

server.listen().then(({ url }) => {
  console.log(`🚀 Server ready at ${url}`)
});
```

apollo-server-lambda

```
const { ApolloServer } = require('apollo-server-lambda')
const { readFileSync } = require('fs')
const resolvers = require('./resolvers')
const typeDefs = readFileSync('./lib/types.graphql', 'UTF8')

const server = new ApolloServer({ typeDefs, resolvers })

exports.handler = server.createHandler() // 🚀
```

localhost:4000

site

PRETTIFY HISTORY http://localhost:4000/ COPY CURL SHARE PLAYGROUND

GraphiQL

```
1 ▼ querygetPost {
2   site(name: BUSTLE) {
3     post(path: "/dad-shoes-rock") {
4       id
5       __typename
6       title
7       body
8       path
9       author {
10         id
11         name
12       }
13       tags {
14         name
15         id
16       }
17     }
18   }
19 }
```

SCHEMA

▼ {
 ▼ "data": {
 ▼ "site": {
 ▼ "post": {
 "id": 3,
 "__typename": "Post",
 "title": "dad shoes are great",
 "body": "a long post about dad
shoes",
 "path": "/dad-shoes-rock",
 "author": {
 "id": 4,
 "name": "Dale"
 },
 "tags": [
 ▶

localhost:4000

site X +

PRETTIFY HISTORY http://localhost:4000/ COPY CURL SHARE PLAYGROUND

SCHEMA

```
1 ▼ querygetPost {
2 ▼   site(name: BUSTLE) {
3 ▼     post(path: "/dad-shoes-rock") {
4       id
5       __typename
6       title
7       body
8       path
9       author {
10         id
11         name
12       }
13       tags {
14         name
15         id
16       }
17     }
18   }
19 }
```

▼ {
▼ "data": {
▼ "site": {
▼ "post": {
"id": 3,
"__typename": "Post",
"title": "dad shoes are great",
"body": "a long post about dad
shoes",
"path": "/dad-shoes-rock",
"author": {
"id": 4,
"name": "Dale"
},
"tags": [
]

Search the schema ...

QUERIES

site(...): Site!

MUTATIONS

resetDevData: String!

site(
 name: SITE_NAME!
): Site!

TYPE DETAILS

type Site {

 id: Int!

 name: String!

 post(...): Post

}

ARGUMENTS

 name: SITE_NAME!

post(
 path: String!
): Post

TYPE DETAILS

type Post {

 id: Int!

 path: String!

 title: String!

 body: String!

 author: User!

 tags: [Tag!]!

}

ARGUMENTS

 path: String!

author: User!

TYPE DETAILS

type User {

 id: Int!

 name: String!

}

The Schema

- Defines what the data looks like
- Doesn't care how it behaves

The Schema

```
type Query {  
  site(name: SITE_NAME!): Site!  
}
```

```
enum SITE_NAME {  
  BUSTLE  
  ROMPER  
}
```

```
type Site {  
  id: Int!  
  name: String!  
  post(path: String!): Post  
}
```

The Schema

```
type Post {  
    id: Int!  
    path: String!  
    title: String!  
    body: String!  
    author: User!  
    tags: [Tag!]!  
}
```

The Schema

```
type User {  
    id: Int!  
    name: String!  
}
```

```
type Tag {  
    id: Int!  
    name: String!  
}
```

The Resolvers

- Only needed for lookups
- Return a full object

The Resolvers

```
module.exports = {
  Query: {
    site: (root, { name }) => { /*...*/ }
  },
  Site: {
    post: async (site, { path }) => { /*...*/ }
  },
  Post: {
    author: async post => { /*...*/ },
    tags: async post => { /*...*/ }
  }
}
```

The Site Resolver

```
{  
  Site: {  
    post: async (site, { path }) => {  
      const edge = await graph.findLabeledEdge({  
        subject: site.id,  
        predicate: 'path',  
        label: path  
      })  
      if (!edge) { return null }  
      const { object: postId } = edge  
      return graph.findNode(postId)  
    }  
  }  
}
```

The Post Resolver

```
{  
  Post: {  
    author: async ({ id }) => {  
      const [{ object: userId }] = await graph.findEdges({  
        subject: id,  
        predicate: 'PostHasAuthor'  
      })  
      return graph.findNode(userId)  
    },  
    tags: async ({ id }) => {  
      const edges = await graph.findEdges({  
        subject: id,  
        predicate: 'PostHasTags'  
      })  
      return Promise.all(edges.map(({ object }) => graph.findNode(object)))  
    }  
  }  
}
```

The N+1 problem

Command Batching

dataloader - lets you batch and dedupe all queries in a single tick

```
const loader = new DataLoader(ids => db.getBunchOfIds(ids))
await Promise.all([
  loader.load(1),
  loader.load(2),
  loader.load(2),
  loader.load(3)
])
// db.getBunchOfIds([1,2,3])
```

Command Batching

redis-loader - lets you pipeline all queries in a single tick.

```
const posts = await Promise.all([
  graph.findNode(1),
  graph.findNode(2),
  graph.findNode(2),
  graph.findNode(3)
])
```

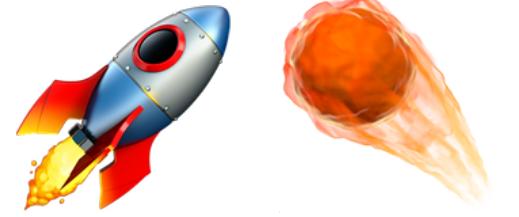
```
// redis
1535039766.77294 [2 127.0.0.1:55031] "hmget" "node:1" "c" "data"
1535039766.77397 [2 127.0.0.1:55031] "hmget" "node:2" "c" "data"
1535039766.77554 [2 127.0.0.1:55031] "hmget" "node:2" "c" "data"
1535039766.77725 [2 127.0.0.1:55031] "hmget" "node:3" "c" "data"
// 434 µs
```

```
// redis-loader
1535039810.951880 [13 127.0.0.1:55036] "multi"
1535039810.951898 [13 127.0.0.1:55036] "hmget" "node:1" "c" "data"
1535039810.951921 [13 127.0.0.1:55036] "hmget" "node:2" "c" "data"
1535039810.951940 [13 127.0.0.1:55036] "hmget" "node:3" "c" "data"
1535039810.951956 [13 127.0.0.1:55036] "exec"
// 76 µs
```

Lambda Tooling

- npm/sammie - "Serverless Application Model Made Infinitely Easier"
- Architect - <https://arc.codes>
- npm/shep

You're ready to rock and roll



A screenshot from the game Nemesis. The scene depicts a futuristic city at night, with buildings featuring red and blue lights. A massive alien mothership with multiple wings and a central hangar bay is hovering over the city. Several smaller alien ships are flying around the mothership, some dropping red energy beams or bombs onto the city. The sky is dark with scattered stars. In the foreground, there are large, jagged rock formations and debris. The overall atmosphere is one of a major space battle taking place over a populated urban area.

Nemesis isn't done yet, you can help

Open source

- bluestream Streams for Async functions
- mobiledoc-kit A toolkit for building WYSIWYG editors with Mobiledoc
- nemesis-db A fast redis graph database
- redis-loader An ioredis-like object that batches commands via dataloader
- sammie Serverless Application Model Made Infinitely Easier
- streaming-iterables Replace your streams with async iterators

Thank you



- I'm Francis / reconbot
- For slides and a short story
<https://github.com/reconbot/we-live-in-memory>
- For a great place to work
<https://bustle.company>