Is GraphQL the end of RESTful APIs?

...

Joakim Lundborg, Wrapp

(the dominance of)

Is GraphQL the end of RESTful APIs?

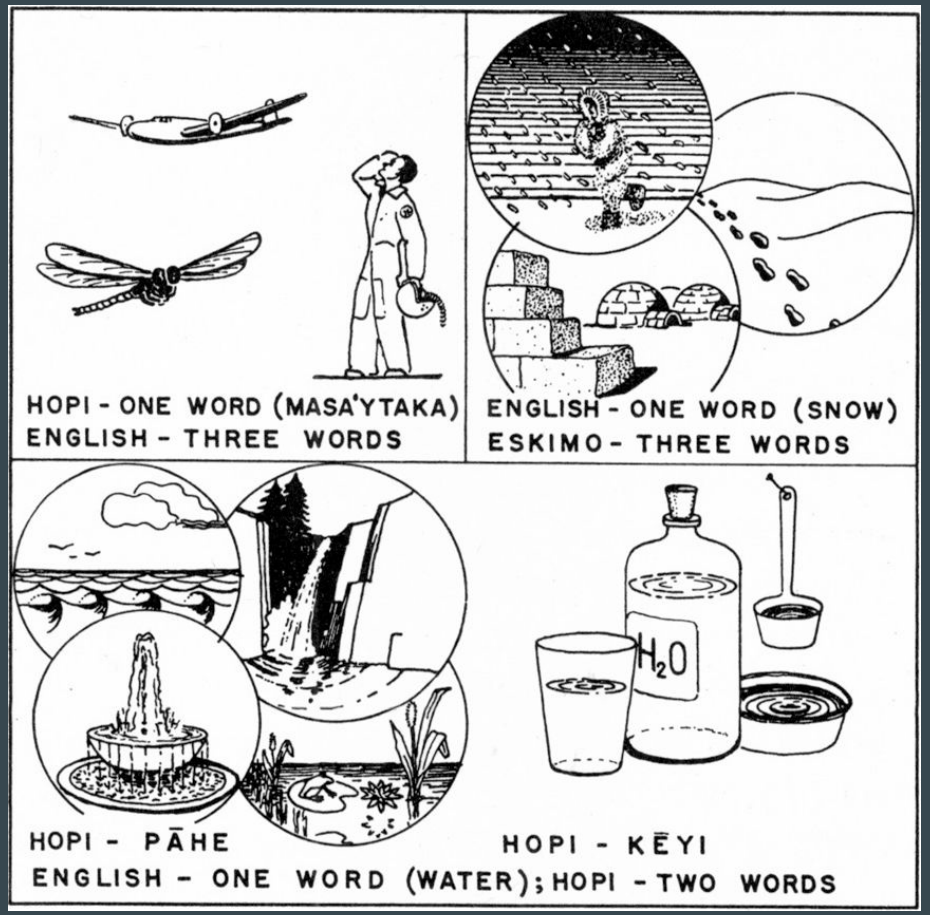
(and other similar technologies)

...

Maybe, let's take a look

Joakim Lundborg
@joakimlundborg
github.com/cortex

Team Shopper @Wrapp



The Sapir-Whorf hypothesis

Image from the article “Language and Science”, originally published in the MIT Technology Review, 1940. Image copyright MIT Press.

An API is a language in which we converse with a system

APIs structure our thinking

RESTful APIs

•••

Everything is a resource

REST nouns

•••

Points out a single resource

REST verbs

...

GET “give me this resource
PUT “here is a resource”
DELETE “remove this document”
POST “~~create a document”~~ “JUST DO SOMETHING”

HATEOAS

•••

Hypermedia drives state

...sounds great, why would anyone want to change this?

...


Myspace
Vimeo
Skype
WhatsApp
Vine
YouTube
Tumblr
LinkedIn
Facebook
Twitter
Instagram
Google+

Fat clients & ever changing data

Round trip time

...

...maybe HTTP/2 will fix this

Type safety

...

MIME if you are lucky,
roll your own otherwise
(openapi, JSON-schema)

Over/underfetching

•••

Unnecessary bloat, or extra round trips, pick any two

Versioning

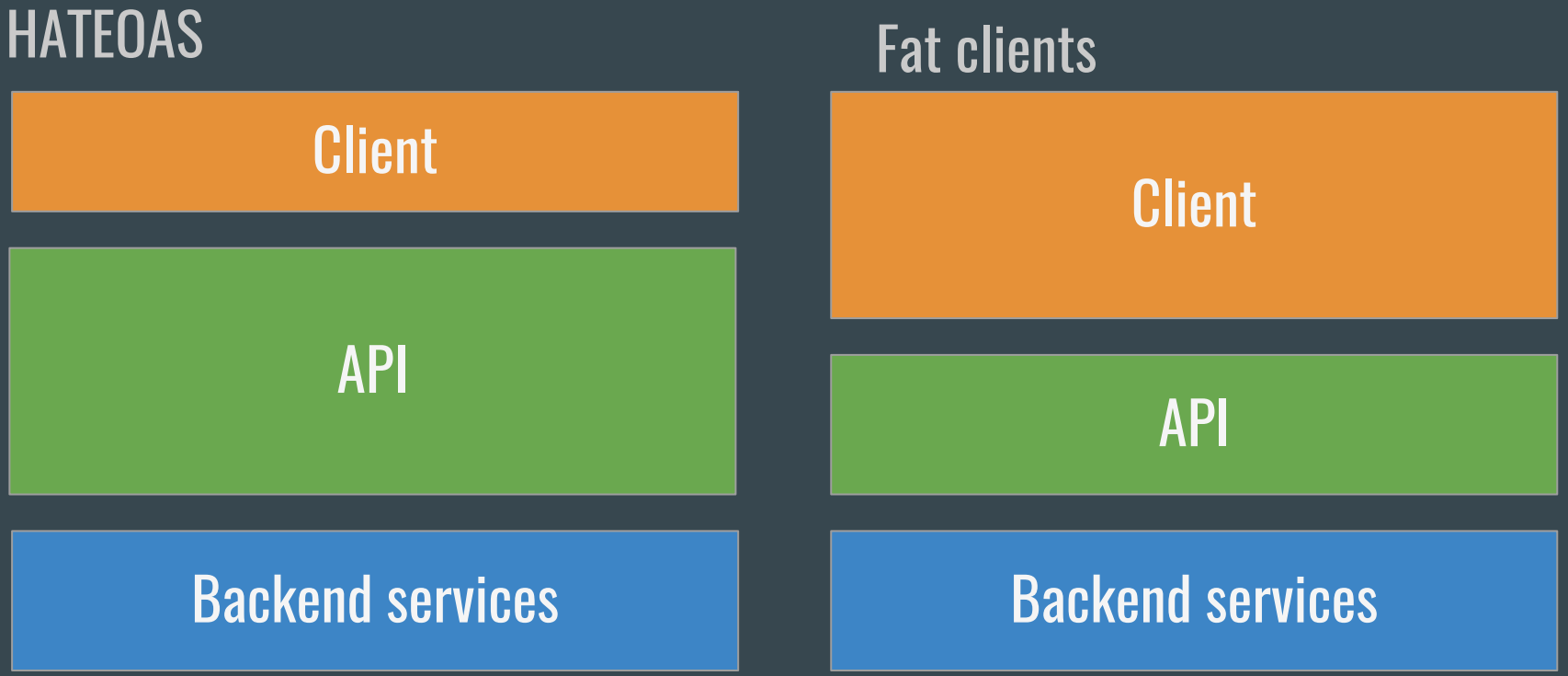
...

Have fun!

Versions are great, let's make more of them

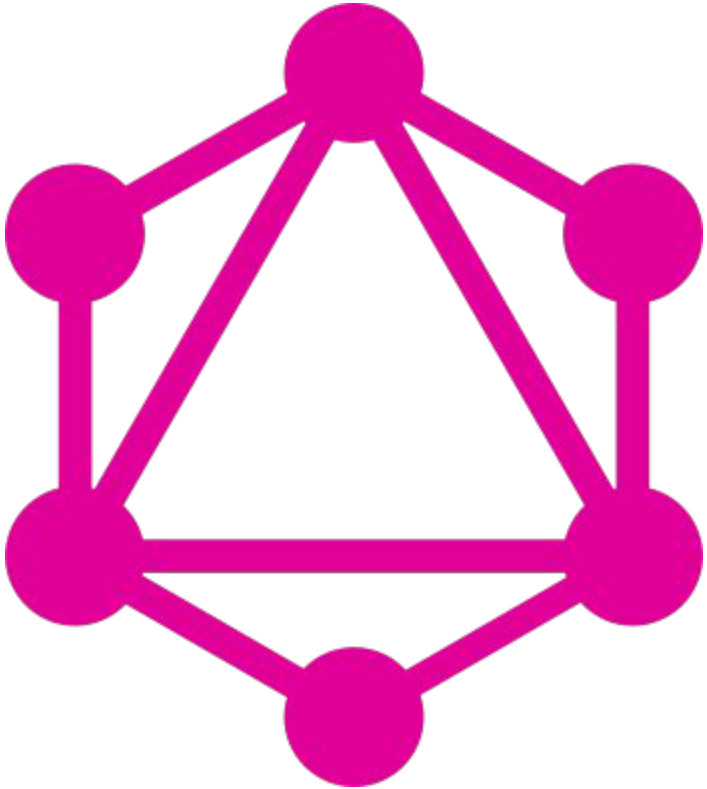
```
$ cat endpoints.py | grep route
  @app.route('/v3/fail', methods=['GET'])
  @app.route('/v3/client-upgrade-required', methods=['GET'])
  @app.route('/v3/ping', methods=['GET'])
  @app.route('/v3/status', methods=['GET'])
  @app.route("/v3/auth/sms/request-code", methods=['POST'])
  @app.route("/v3/auth/sms", methods=['POST'])
  @app.route('/v3/auth/session-token', methods=['POST'])
  @app.route('/v1/users/me/validate', methods=['POST'])
  @app.route('/v3/users/me/register', methods=['POST'])
  @app.route('/v3/users/me/discover', methods=['GET'])
  @app.route('/v4/users/me/discover', methods=['GET'])
  @app.route('/v4/users/me/nearby-offers', methods=['GET'])
  @app.route('/v3/offers/<string:offer_id>', methods=['GET'])
  @app.route('/v4/offers/<string:offer_id>/activate', methods=['POST'])
  @app.route('/v3/offers/<string:offer_id>/activate', methods=['POST'])
  @app.route('/v3/offers/<string:offer_id>/like', methods=['POST'])
  @app.route('/v3/offers/<string:offer_id>/unlike', methods=['POST'])
  @app.route('/v3/merchants/<string:merchants_id>/punchcard', methods=['GET'])
  @app.route('/v3/impressions', methods=['POST'])
  @app.route('/v3/users/me', methods=['GET'])
  @app.route('/v3/users/me/entercard-requested-confirmed', methods=['POST'])
  @app.route("/v3/purchases/<string:purchase_id>", methods=['GET'])
  @app.route('/v3/purchases/<string:purchase_id>/location', methods=['POST'])
  @app.route('/v3/users/me/events', methods=['GET'])
  @app.route('/v3/users/me/purchases', methods=['GET'])
  @app.route('/v3/users/me/dashboard', methods=['GET'])
  @app.route('/v3/users/me/last-purchase', methods=['GET'])
  @app.route('/v3/users/me/profile', methods=['GET'])
  @app.route('/v3/users/me/category-status', methods=['GET'])
  @app.route('/v3/users/me/stats', methods=['GET'])
  @app.route('/v3/users/me/payouts', methods=['GET'])
  @app.route('/v3/users/me/bank-wish/<string:bank_id>', methods=['post'])
  @app.route('/v3/integrations/<integration>/register', methods=['POST'])
  @app.route('/v5/integrations', methods=['GET'])
  @app.route('/v4/terms', methods=['GET'])
  @app.route('/v3/integrations/new-nordea/terms', methods=['GET'])
  @app.route('/v3/integrations/nordea/terms', methods=['GET'])
  @app.route('/v3/integrations/nordea/supported-bins', methods=['GET'])
```

Thin and fat clients



GraphQL

Everything is a graph



GraphQL is a declarative query language

Declarative

•••

Say what you want to get, let the server figure it out

A simple example

```
{
  me {
    name
    role
    company
    handle
    url
  }
}
```

```
{
  "data": {
    "me": {
      "name": "Joakim Lundborg",
      "role": "Team lead",
      "company": "Wrapp",
      "handle": "@joakimlundborg",
      "url": "/speakers/joakim-lundborg/"
    }
  }
}
```

```
GET /users/me
```

```
{
      "name": "Joakim Lundborg",
      "role": "Team lead",
      "company": "Wrapp",
      "handle": "@joakimlundborg",
      "url": "/speakers/joakim-lundborg/"
}
```

GraphQL can do mutation too

```
mutation me {
  changeRole(“GraphQL evangelist){
      name
      role
   }
}
```

```
{
  "data": {
    "me": {
      "name": "Joakim Lundborg",
      "role": "GraphQL evangelist",
     }
  }
}
```

```
POST /users/me
```

```
{
      "name": "Joakim Lundborg",
      "role": "Team lead",
      "company": "Wrapp",
      "handle": "@joakimlundborg",
      "url": "/speakers/joakim-lundborg/"
}
```

Summary

	REST	GraphQL
Round trips	Many	Few
Type safety	No	Yes
Discoverability	HATEOAS/Swagger/etc..	Yes
Performance	Proxies! Caching! CDN!	You'd better have good servers & engineers
Architecture	Thin client/fat Server	Fat client/Fat server

GraphQL

•••

As a language

GraphQL

...

Is in some ways simpler than REST

Two verbs

•••

Query and Mutation

A sophisticated type system

...

With introspection

This declarative thing seems familiar...

SQL

Everything is a ~~table~~ relation

(at least the result set)

GraphQL vs SQL

...

GraphQL is much simpler,
maps to afamiliar tree structure

JSON is just a transport format

...

Anything that can express a tree-structure would work

Selectors can be seen as paths

…

Except you don’t pay for the roundtrips

DEMO

...

https://github.com/cortex/graphql-nordicapis

npm install

node init.js

GraphQL and the future

...

Microservices <3 GraphQL

...

Untangling a micro-web

GraphQL as a library

...

Even fewer roundtrips

The open web

•••

How do you crawl a GraphQL service?

Open Data

A great fit for GraphQL

GraphQL aggregator proxies?

•••

A future direction for open data APIs?

Is GraphQL the end of RESTful APIs?

(and other similar technologies)

(the dominance of)

...

Yes, but we will probably see more ~~churn~~ innovation

Questions

...