Introduction to GraphQL

By Jorigis Šnaras and Tom Hastjarjanto

What is GraphQL?

GraphQL

What is GraphQL?

+

{ ? }

Graphs

Queries

GraphQL

What is GraphQL?

A means to

define data as

make tailored

Graphs

Queries

What is GraphQL?

"[…] a **query language** for your API, […] using a **type system** you define for your data."

– graphql.org

GraphQL

What is GraphQL?

Define your data

```
type Project {
  name: String
  tagline: String
  contributors: [User]
}
```

(a type)

GraphQL

What is GraphQL?

Ask for what you want

```
{
  project(name: "GraphQL") {
    tagline
  }
}
```

(a query)

GraphQL

What is GraphQL?

Get predictable results

```
{
  "project": {
    "tagline": "A query language for APIs"
  }
}
```

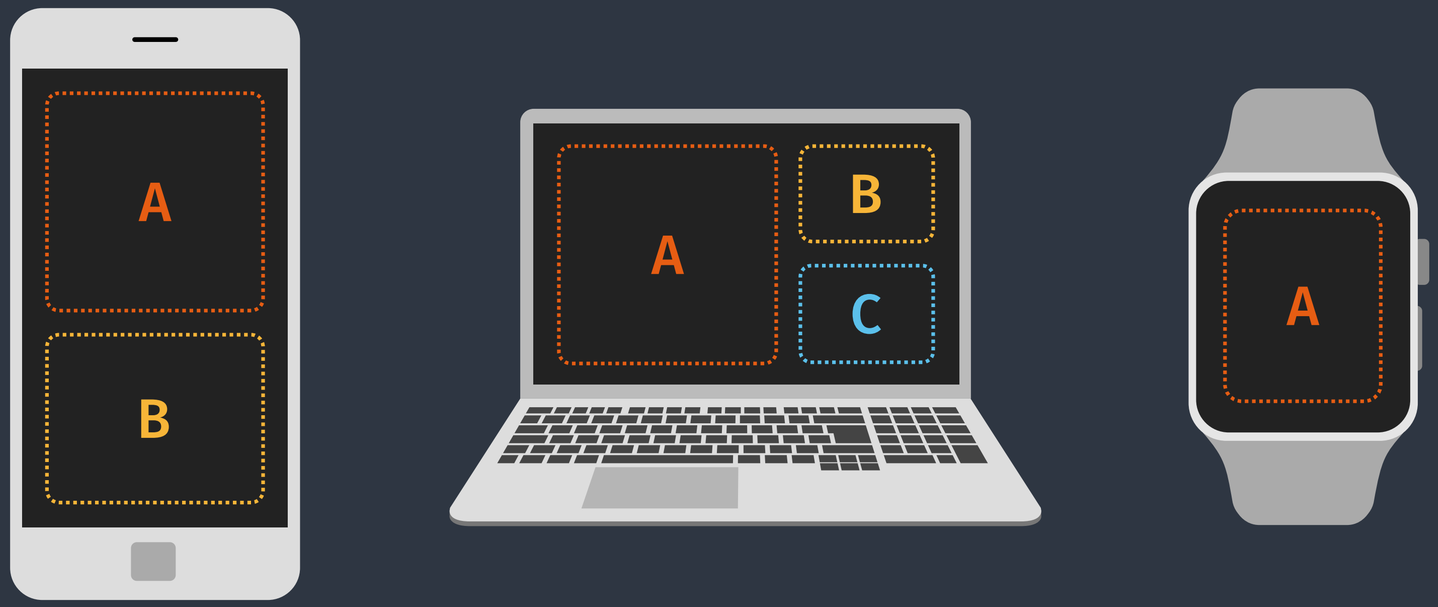
(result JSON)

Use Cases for GraphQL

GraphQL

Use Cases for GraphQL

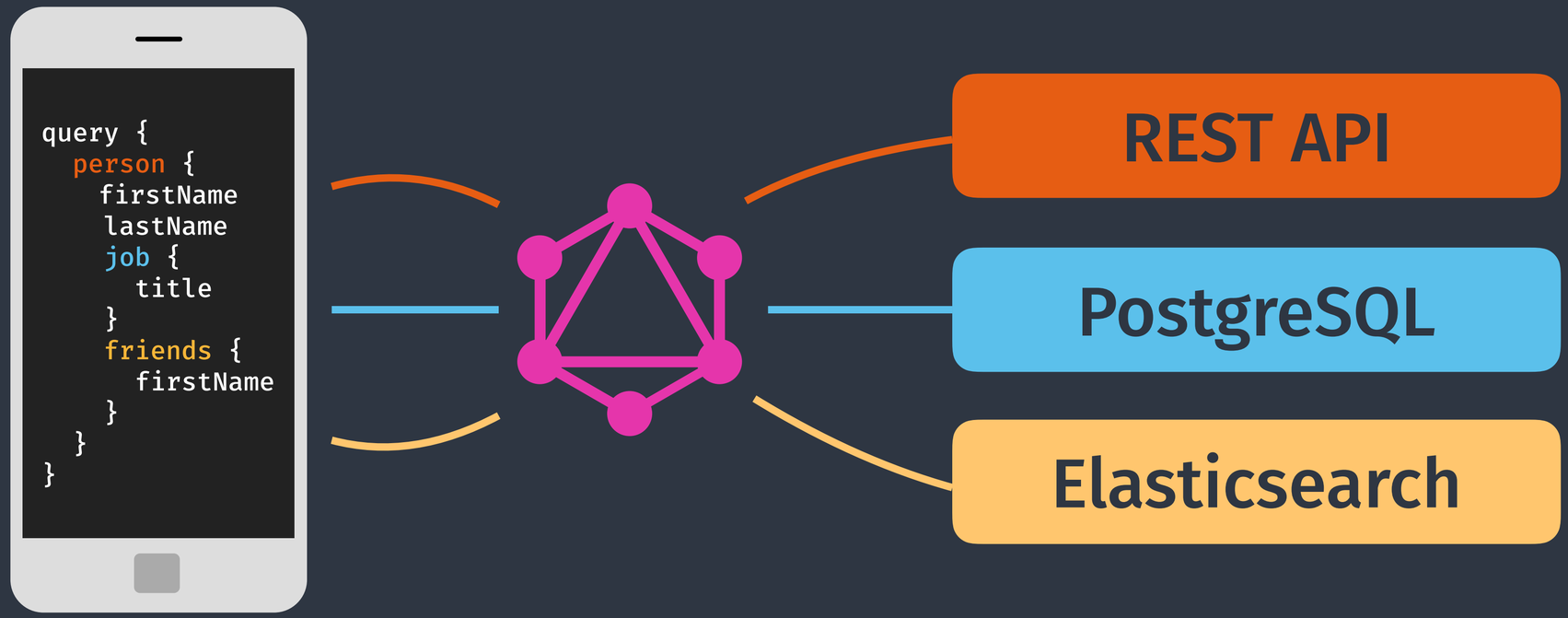
Multiple clients



GraphQL

Use Cases for GraphQL

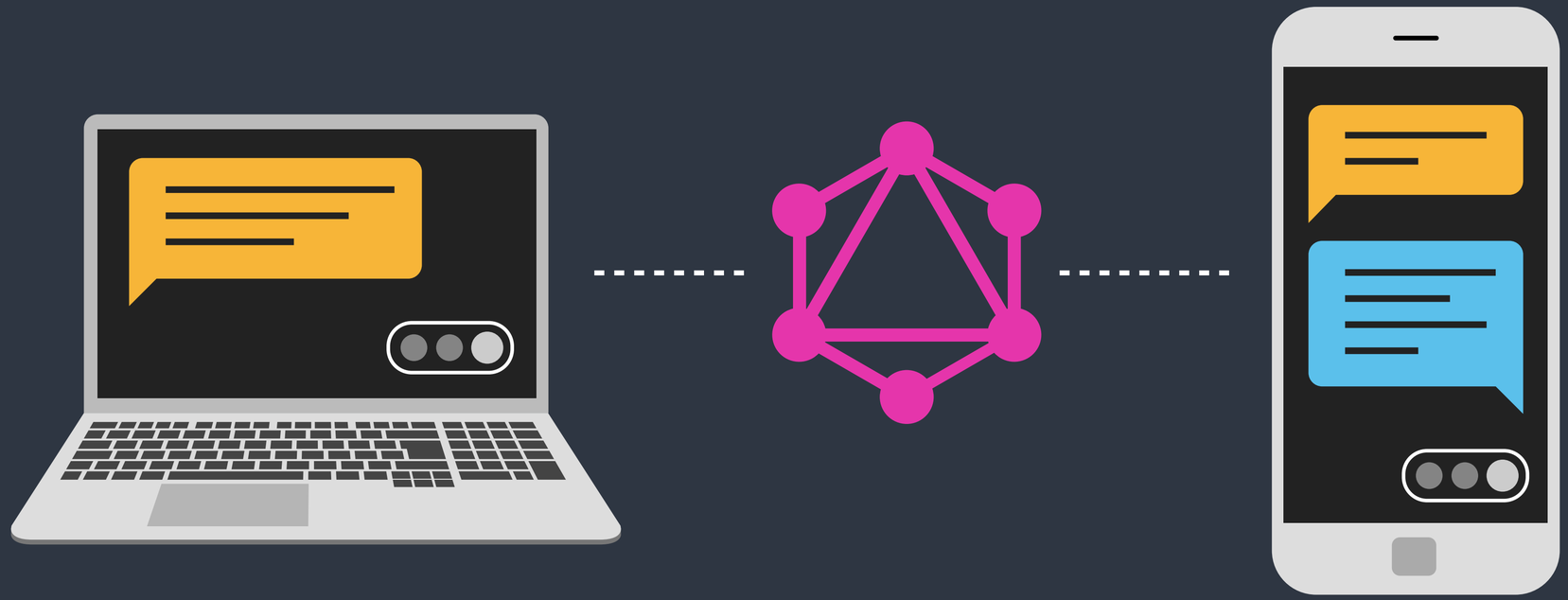
Aggregation of APIs



GraphQL

Use Cases for GraphQL

Real-time APIs



GraphQL

Use Cases for GraphQL

Public APIs

Benefits of GraphQL

Benefits of GraphQL

Optimized frontend development

One backend for many frontends

Focus on what should be displayed

Same protocol for everything

Superb out-of-the-box tooling and libraries

Benefits of GraphQL

Reduced workload for API teams

No need to make custom endpoints

Less focus on specific use cases

The schema is the documentation

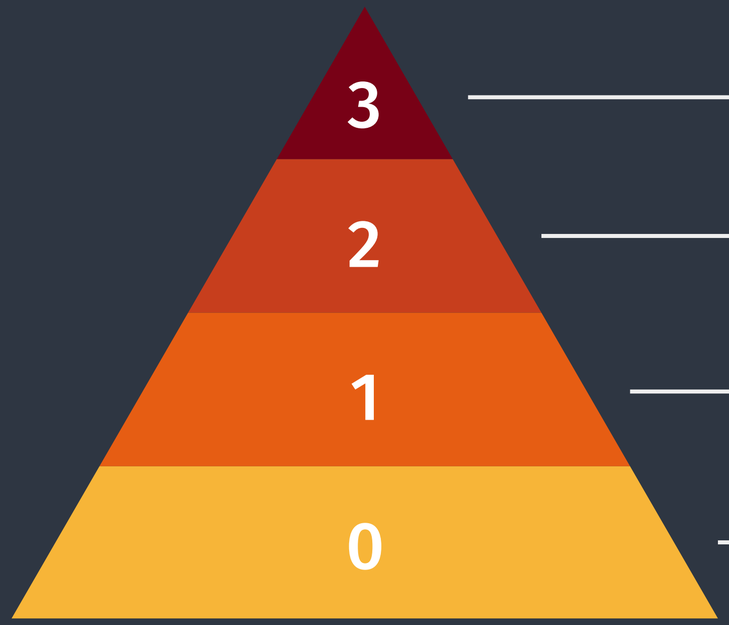
One clean API between backends and frontends

GraphQL vs REST

GraphQL

GraphQL vs REST

Proper REST is hard



HATEOAS

Multiple URI-based resources and verbs

Multiple URI-based resources and single verbs

Single URI and a single verb

Richardson Maturity Model

GraphQL

Proper REST is complex

```
{
  "departmentId": 10,
  "departmentName": "Administration",
  "links": [
    {
      "href": "10/employees",
      "rel": "employees",
      "type": "GET"
    }
  ]
}
```

GraphQL

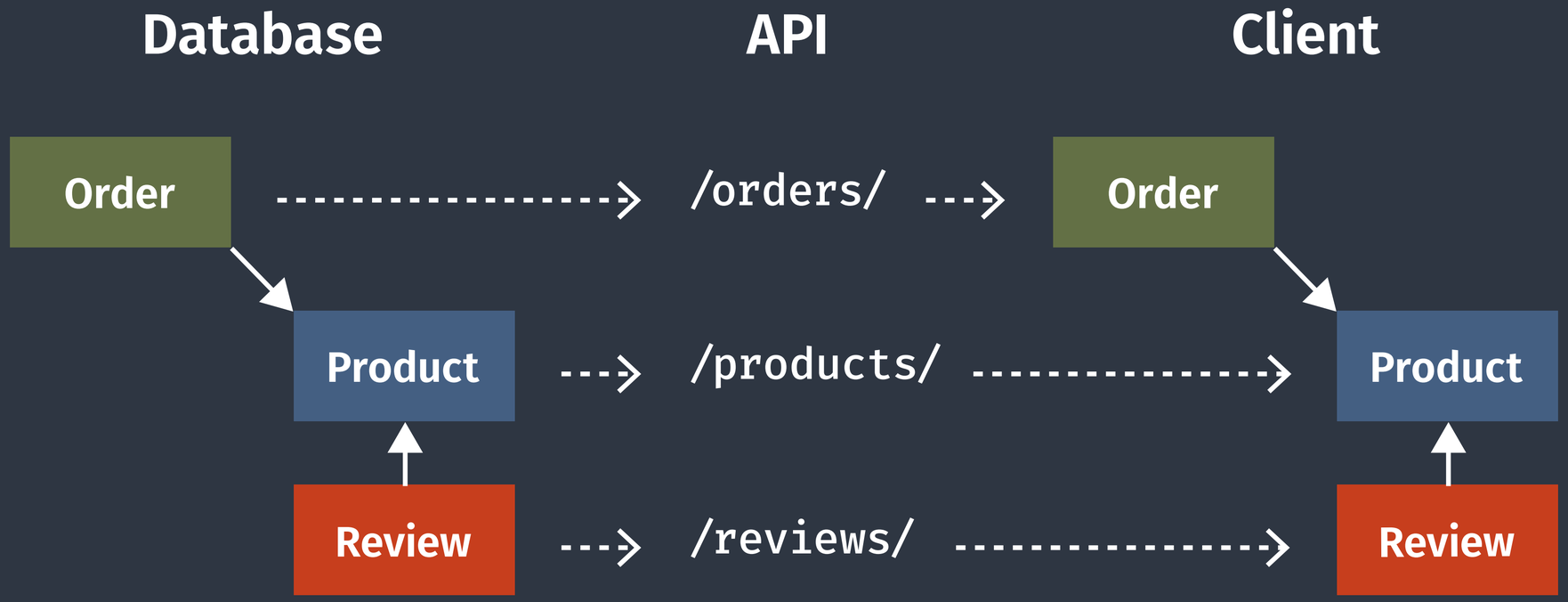
Proper REST is complex

```
<appointment>
  <slot id="1234" doctor="mjones" start="1400" end="1450"/>
  <patient id="jsmith"/>
  <link rel="/linkrels/appointment/cancel"
    uri="/slots/1234/appointment"/>
  <link rel="/linkrels/appointment/addTest"
    uri="/slots/1234/appointment/tests"/>
  <link rel="self"
    uri="/slots/1234/appointment"/>
  <link rel="/linkrels/appointment/changeTime"
    uri="/doctors/mjones/slots?date=20100104&status=open"/>
  <link rel="/linkrels/appointment/updateContactInfo"
    uri="/patients/jsmith/contactInfo"/>
  <link rel="/linkrels/help"
    uri="/help/appointment"/>
</appointment>
```

GraphQL

GraphQL vs REST

Proper REST is complex



GraphQL

Proper REST is complex

Use case: deactivate a resource

Property?

Operation?

POST? PUT? PATCH?

Is GraphQL easier?

GraphQL vs REST

Is GraphQL easier?

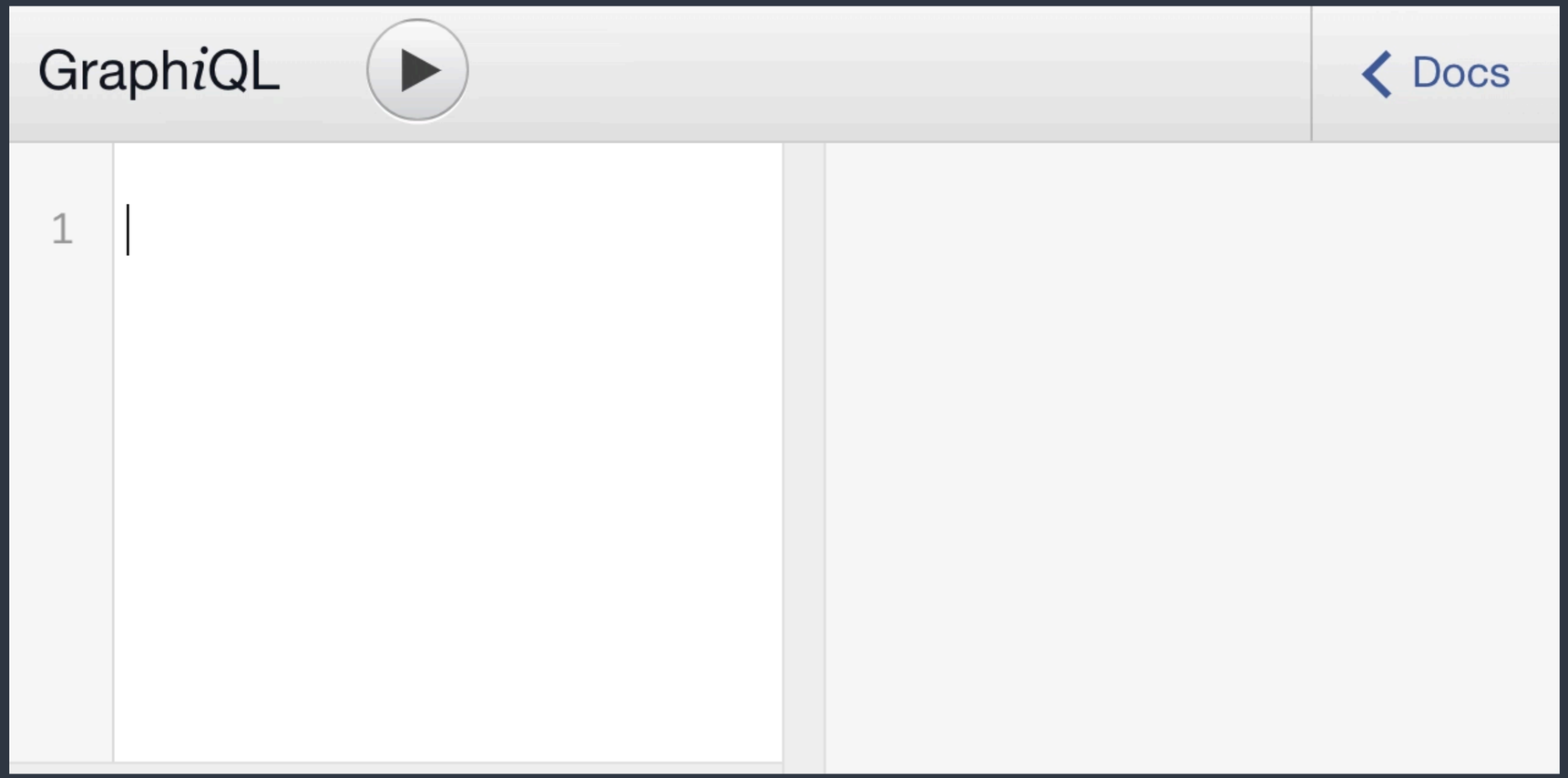
HTTP as “dumb” protocol

`/graphql`

GraphQL

GraphQL vs REST

Is GraphQL easier?



Caveats of GraphQL

Caveats of GraphQL

Ecosystem is young, moves fast

Less experience in the wild

No drop-in replacement for REST

Uses HTTP differently

Caching, routing, load balancing

TL;DR

TL;DR

GraphQL is

defined in a spec

a query (and schema definition) language

a client-server communication pattern

TL;DR

GraphQL is used for

serving multiple clients from one backend

API aggregation

exposing public APIs

Thank you!