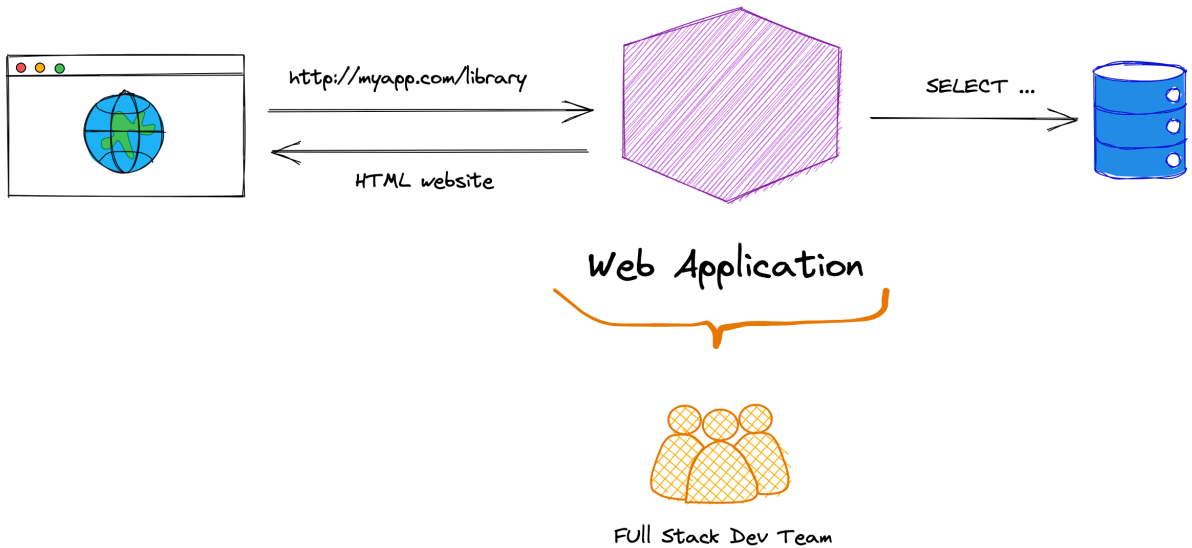
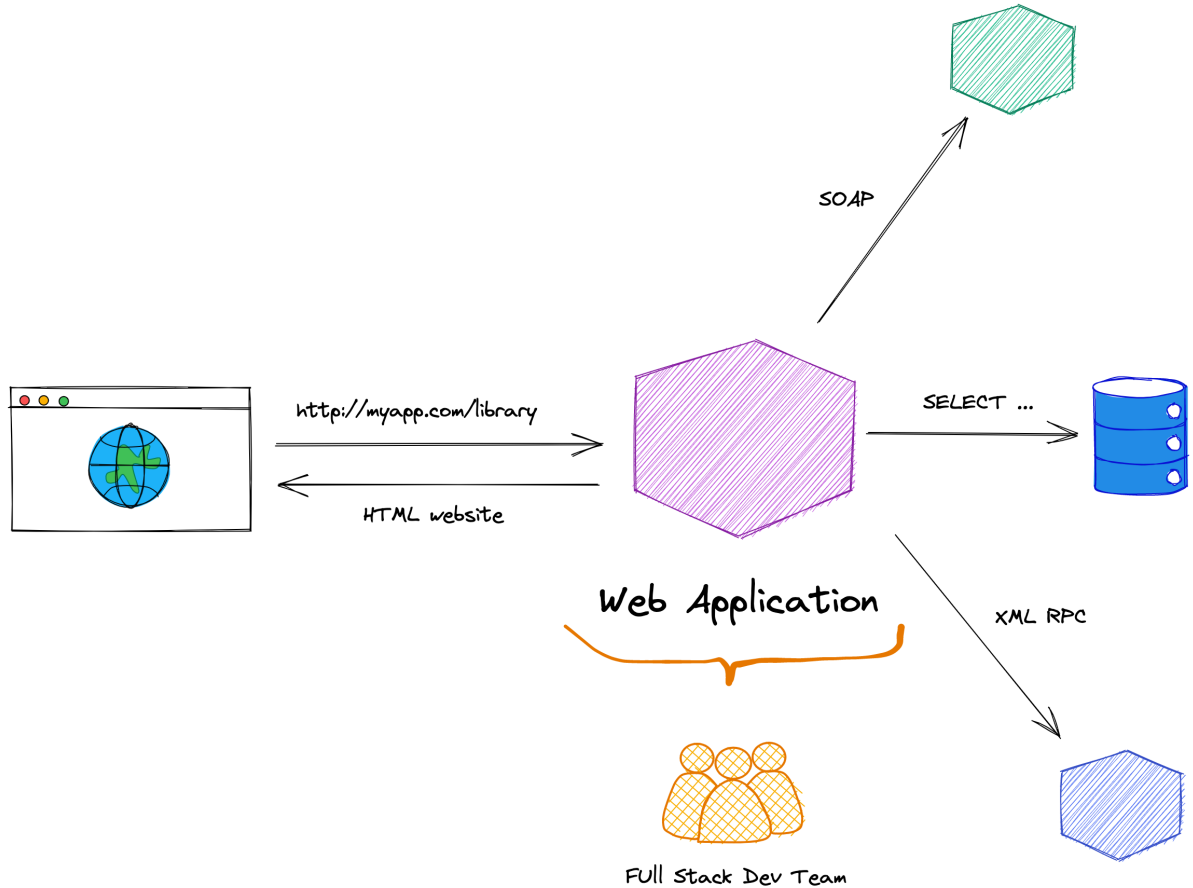

Spring GraphQL
@maciejwalkowiak

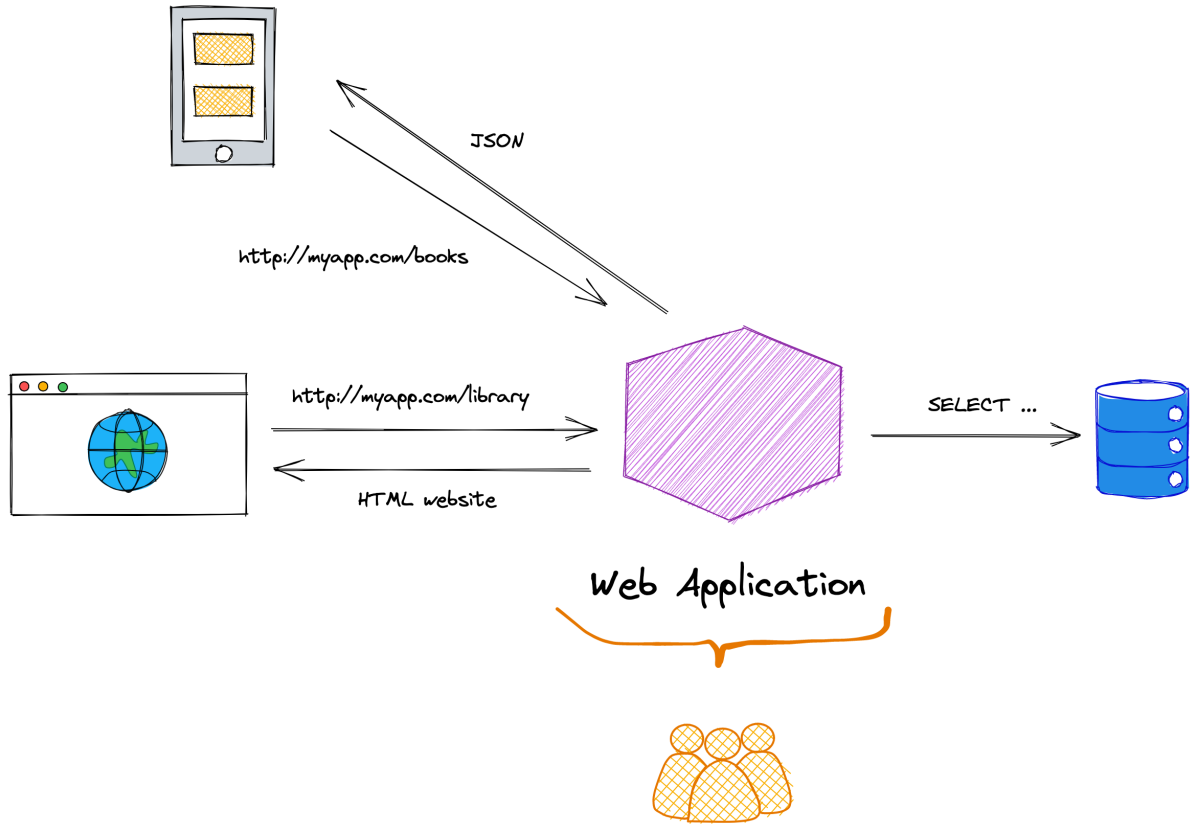
Web Applications < 2010



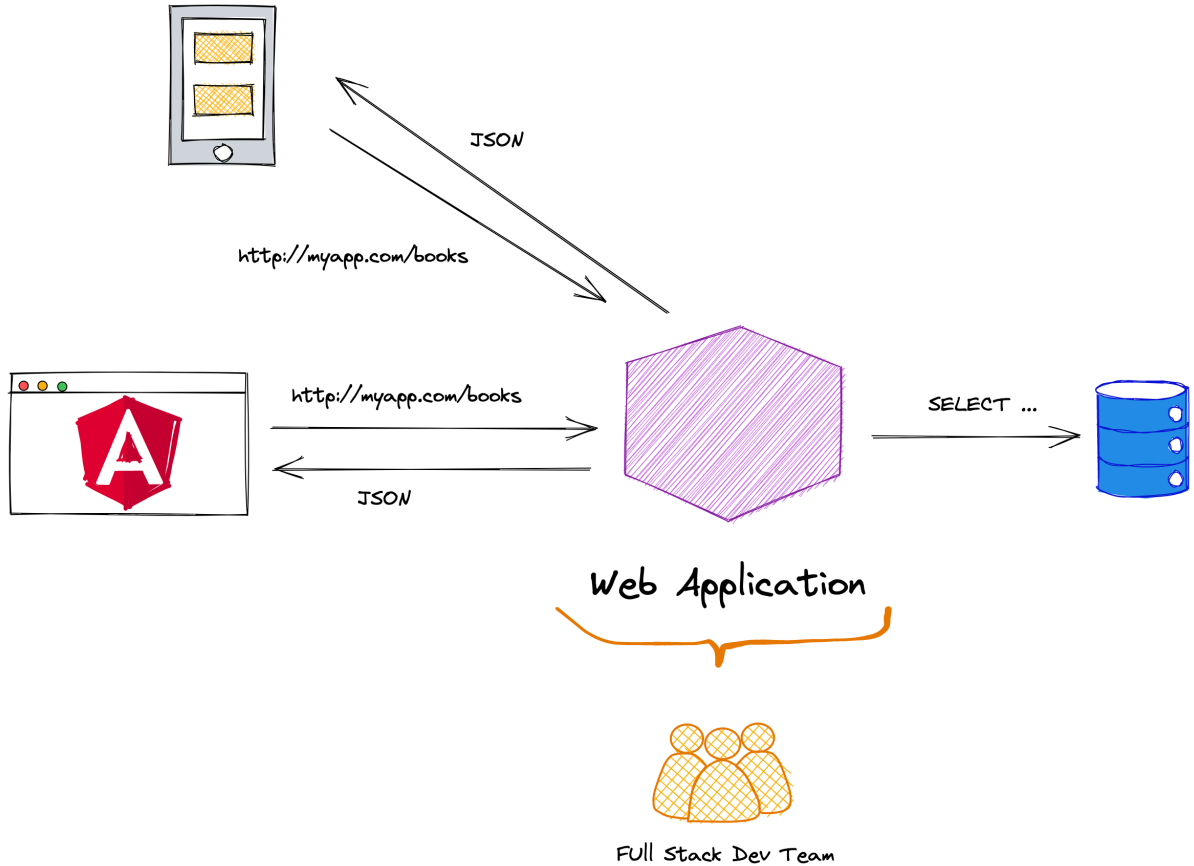
Enterprise Applications < 2010



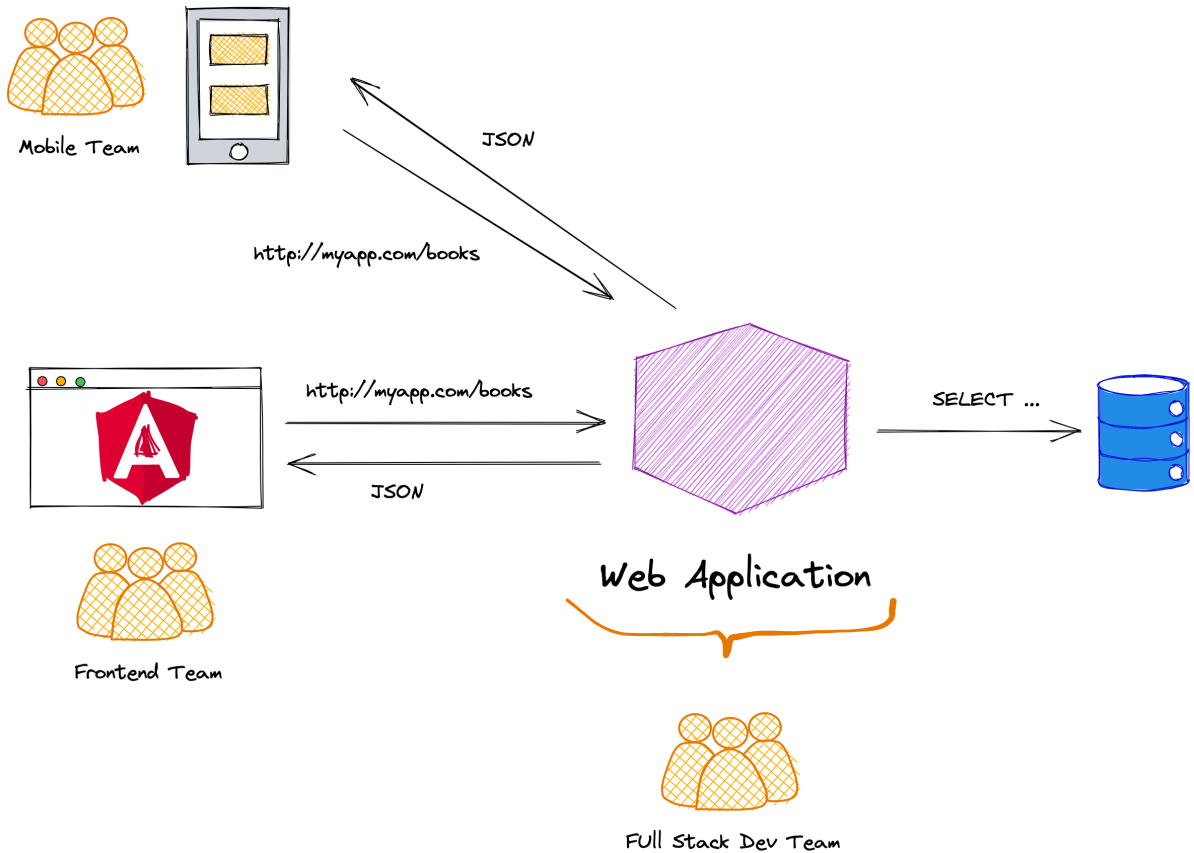
Mobile Apps > 2007



Angular & SPA > 2011



Angular & SPA > 2011



Different Clients - Different Needs

- browser displays more data and more types of data - payload size is not a huge issue
- mobile client displays small amount of data - small payload size is critical for UX

Web:

```
GET /books
[
  {
    "id": 122,
    "title": "Modern Java",
    "publicationYear": 2021,
    "coverUrl": "https://...",
    "authorId": 344,
    "description": "... long description text ..."
  },
  ...
]
```

Mobile:

```
GET /books
[
  {
    "id": 122,
    "title": "Modern Java",
    "publicationYear": 2021,
    "coverUrl": "https://...",
    "authorId": 344
  },
  ...
]
```

Data Composition

- client is responsible for data composition
- multiple requests sent from client (potential N+1 problem)

Fetch books:

```
GET /books
[
  {
    "id": 122,
    "title": "Modern Java",
    "publicationYear": 2021,
    "coverUrl": "https://...",
    "authorId": 344,
  },
  {
    "id": 123,
    "title": "Spring in Action",
    "publicationYear": 2020,
    "coverUrl": "https://...",
    "authorId": 345,
  }
]
```

Fetch authors:

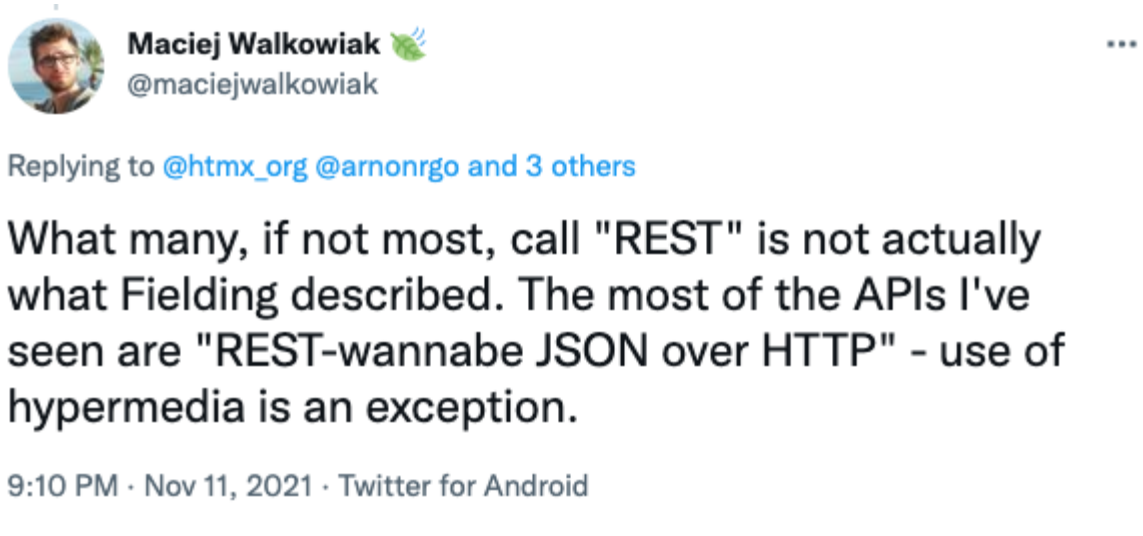
```
GET /authors/344
{
  "id": 344,
  "name": "John Smith",
  "avatarUrl": "https://..."
}

GET /authors/345
{
  "id": 344,
  "name": "John Smith",
  "avatarUrl": "https://..."
}
```

Problem with REST(ish) APIs

- REST is an architecture style
- no spec - just principles
- difficult to get "right"
- overfetching
- expensive data composition
- no schema
- hypermedia isn’t what developers want

This is not a REST critique

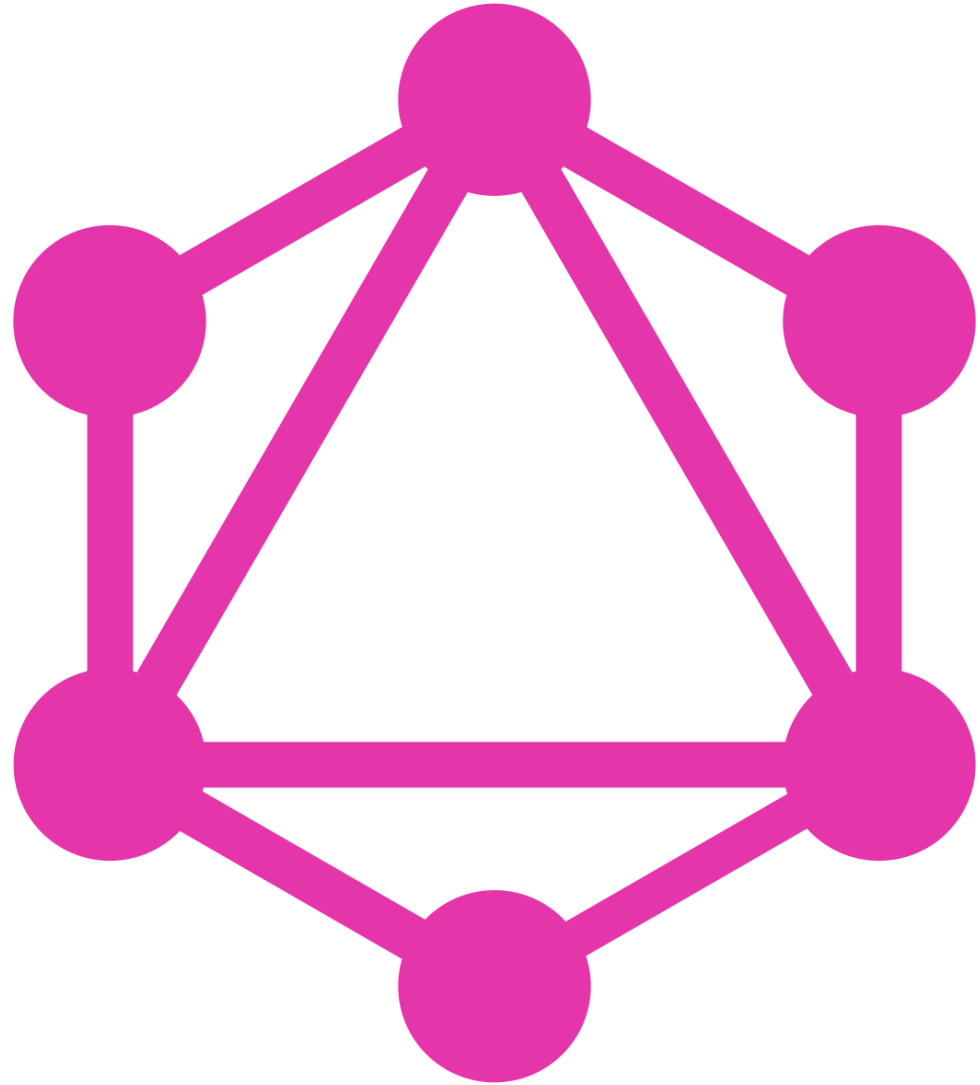


Also, look at https://jsonapi.org/.

Hello GraphQL

- Forget everything you know about REST
- Ignore HTTP methods, status codes, caching

> A query language for your API GraphQL is a query language for APIs and a runtime for fulfilling those queries with your existing data. GraphQL provides a complete and understandable description of the data in your API, gives clients the power to ask for exactly what they need and nothing more, makes it easier to evolve APIs over time, and enables powerful developer tools.



GraphQL

Schema:

```
type Query {
    findAuthors: [Author]
    author(id: ID!): Author
}

type Author {
    id: ID!
    name: String!
    age: Int
    books: [Book]
}

type Book {
    id: ID!
    title: String!
    author: Author!
    publicationYear: Int
}
```

Query:

```
query {
  findAuthors {
    id
    name
    books {
      title
    }
  }
}
```

Results:

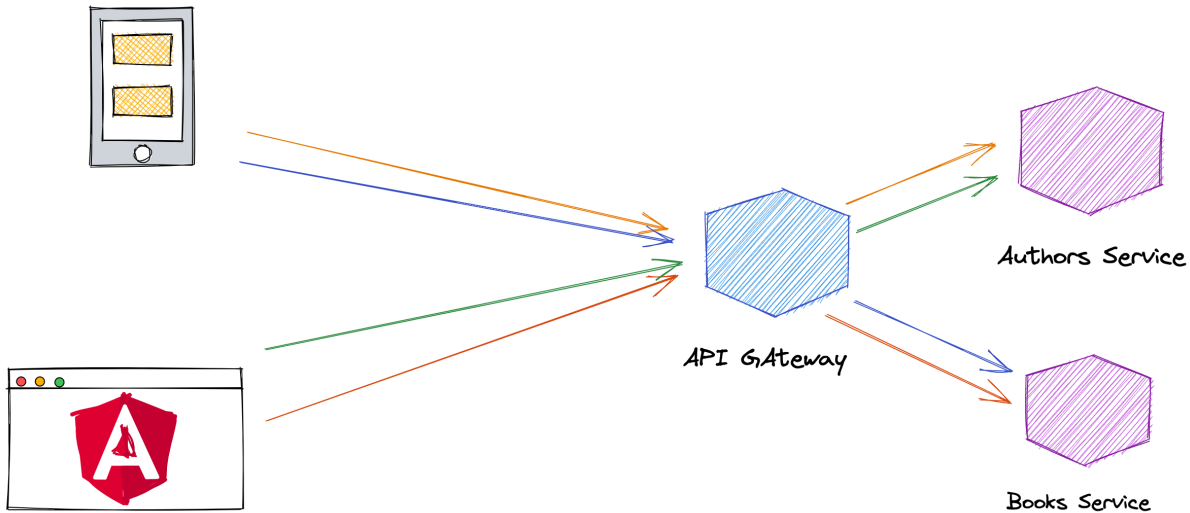
```
{
  "data": {
    "findAuthors": [
      {
        "id": "1",
        "name": "maciej",
        "books": [
          {
            "title": "book 1"
          },
```

DEMO

We have just scratched the surface

Challenges

- microservies & single `/graphql` endpoint?



- caching?
- database transactions?

Continue Learning

- https://www.graphql-java.com/
- https://spring.io/projects/spring-graphql
- https://www.howtographql.com/
- https://youtube.com/springacademy <-- that's me
- follow @maciejwalkowiak on twitter

Thank you!