Why GraphQL?

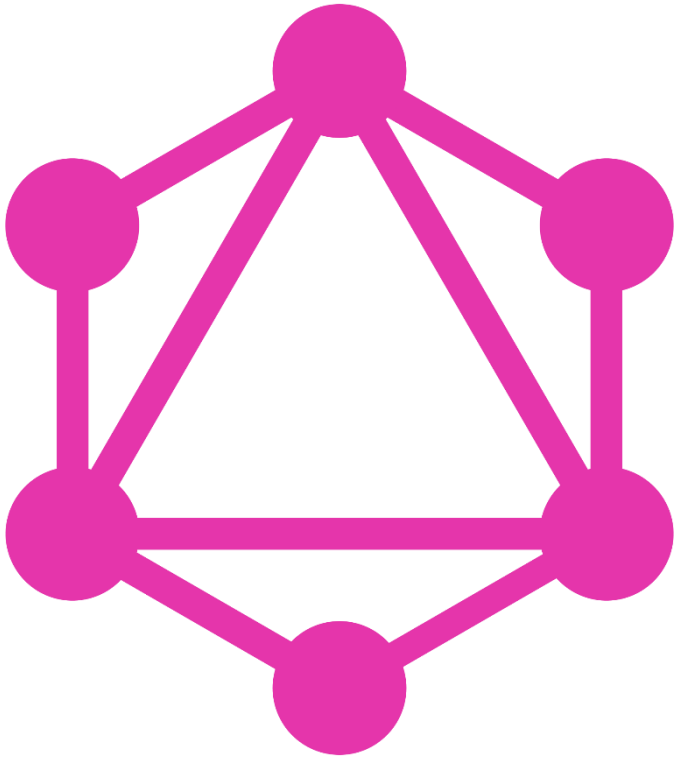
ADHITHI RAVICHANDRAN
SOFTWARE CONSULTANT

@AdhithiRavi www.adhithiravichandran.com

Why GraphQL?

Declarative Data Fetching

Benefits of Declarative Data Fetching



Avoids round-trips to fetch data

No more over-fetching or under-fetching of data

You specify exactly the data you need and GraphQL gives you exactly what you asked for

Multiple Round Trips Using REST

/ps/author/<id>

/ps/author/<id>/courses

/ps/author/<id>/rating

/ps/author/<id>/topics

GraphQL Single Request and Response

```
{
  author (id : 2100) {
    name
    courses {
      title
    }
    rating
    topics (last : 3) {
      name
    }
  }
}
```

```
{
  "data" : {
    "author" : {
     "name": "Adhithi Ravichandran",
     "courses": [
       { title: "React Native: The Big Picture" },
       { title: "GraphQL : The Big Picture" }
     ],
     "rating": "4.5",
     "topics" : [
       { name : "React" },
       { name : "React Native" },
       { name : "GraphQL" }
     ]
    }
  }
```

Over-fetching in REST API

/api/user

```
user {
  firstName: "Adhithi",
  lastName: "Ravichandran",
  gender: "female",
  createAt: "05/15/2019",
  updatedAt: "05/16/2019",
  posts: [],
  comments: []
}
```

The REST API over-fetches data by returning the whole user object.

GraphQL No Over-Fetching

```
user {
  firstName
  lastName
  gender
}
```

```
user {
  firstName: "Adhithi",
  lastName: "Ravichandran",
  gender: "female"
}
```

GraphQL fetches only what you ask for. No more over-fetching data.

Strongly Typed Schema

GraphQL is Strongly Typed

GraphQL is a strongly-typed language, and its schema should have types for all objects that it uses

The schema serves as a contract between client and server

Relies on a query language with a type system

Benefits of Strongly Typed GraphQL Schema

Code is predictable

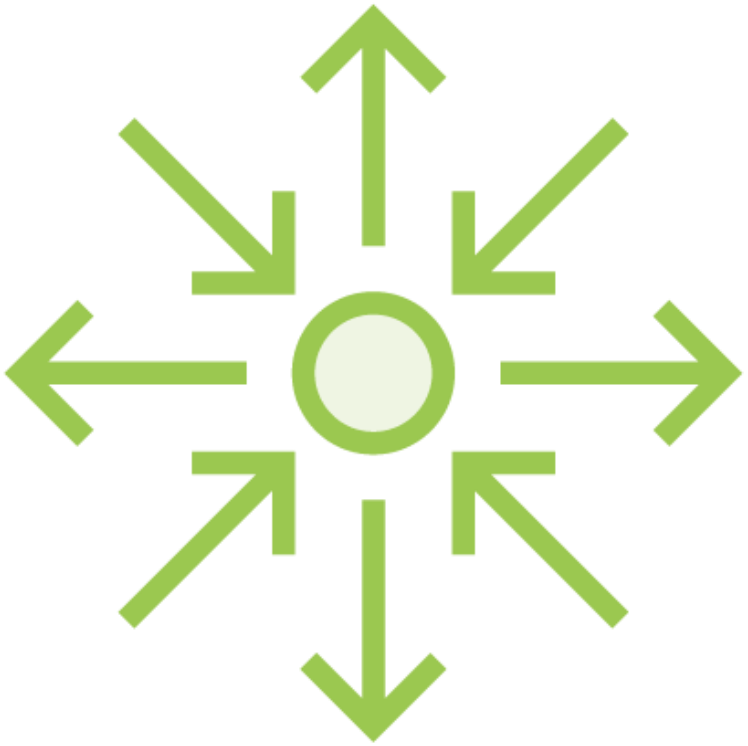
Frontend and backend teams work independently

Earlier detection of errors and speeds development

Facilitates building high quality client tools

Superior Developer Experience

Flexibility



GraphQL offers a lot of flexibility

No versioning

Non-breaking changes

GraphQL – Teams Work Independently

In the REST World

Well, we can't start coding until we get the new endpoints

Frontend dev team

We need it code reviewed, tested and deployed, it will be a few weeks.

Backend API dev team

GraphQL – Teams Work Independently

Frontend and backend teams can work independently

With strictly-typed schema, teams can work in parallel

Frontend can continue work with mocked version of API and also test it

Evolve Your API Without Versioning

Add new fields and types to GraphQL API without breaking changes

Provides apps continuous access to new features

Developers don’t have to maintain multiple versions of the API

Modern Ecosystem and Tooling

Tools like GraphiQL IDE provide excellent developer experience

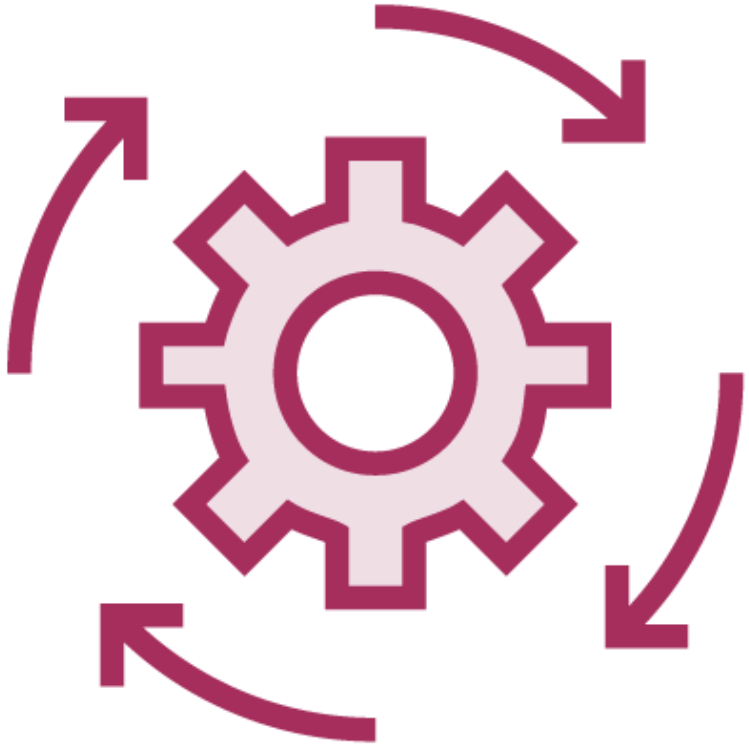
Hours of documentation time saved with in-built self-documentation in the schema

Plenty of GraphQL clients for every UI framework (Apollo, Relay)

Lots of server-side libraries, client libraries and services across several languages

Cost Effective

Rapid product development cycle



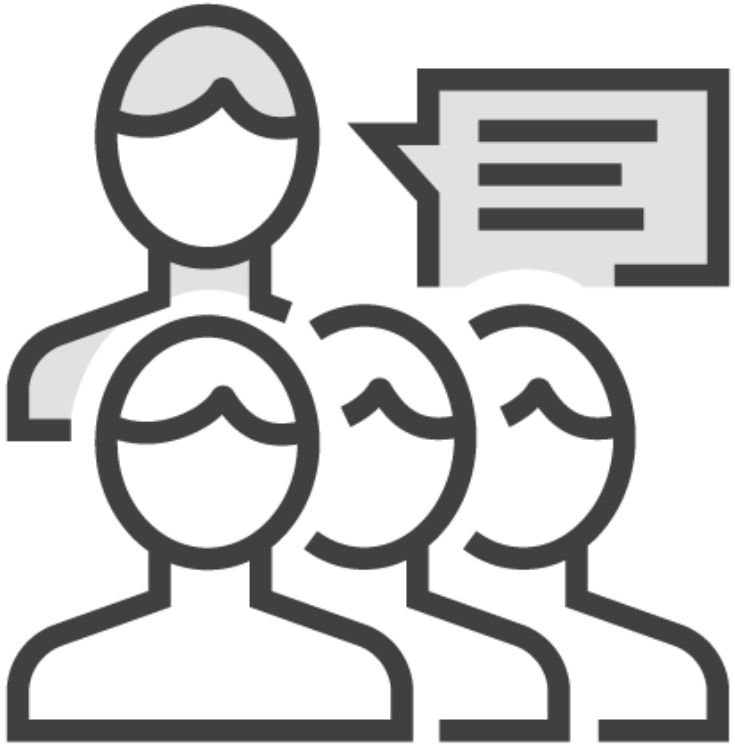
Multiple teams work
in parallel

Self documentation of API

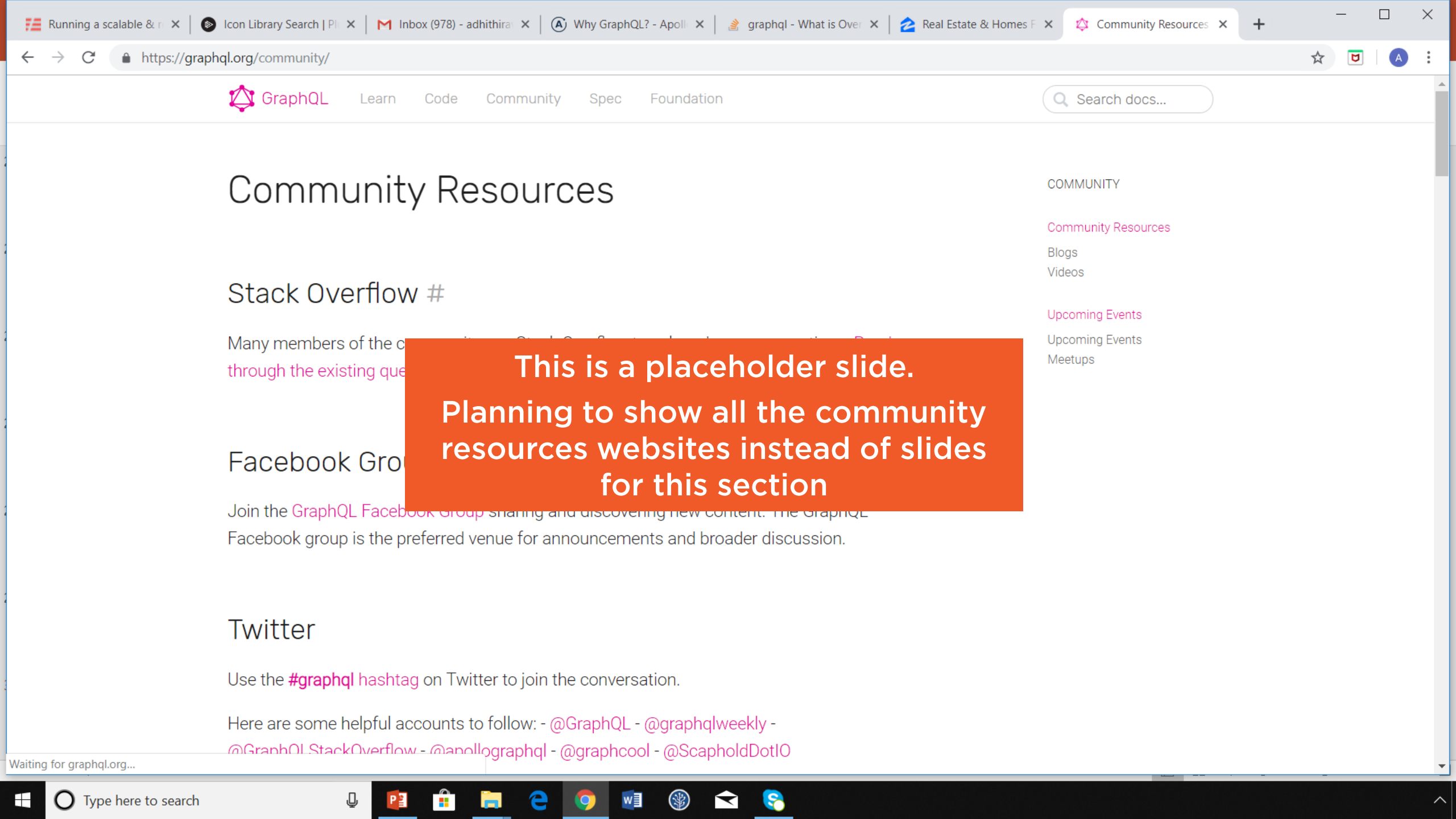
Tons of hours saved from writing elaborate API documentation

Save money with minimal training

Easy to onboard developers on GraphQL



Growing Community


Running a scalable &
Icon Library Search | Pl
Inbox (978) - adhithira
Why GraphQL? - Apoll
graphql - What is Over
Real Estate & Homes F
Community Resources
https://graphql.org/community/
GraphQL
Learn
Code
Community
Spec
Foundation
Search docs...
Community Resources
COMMUNITY
Community Resources
Blogs
Videos
Upcoming Events
Upcoming Events
Meetups
Stack Overflow #
Many members of the c
through the existing que
Facebook Gro
Join the GraphQL Facebook Group
Facebook group is the preferred venue for announcements and broader discussion.
Twitter
Use the #graphql hashtag on Twitter to join the conversation.
Here are some helpful accounts to follow: - @GraphQL - @graphqlweekly -
@GraphQLStackOverflow - @apollographql - @graphcool - @ScapholdDotIO
This is a placeholder slide.
Planning to show all the community resources websites instead of slides for this section
Waiting for graphql.org...
Type here to search

Summary

1. Declarative Data Fetching
2. Strongly Typed Schema
3. Superior Developer Experience
4. Cost Effective
5. Excellent community

Next Module: GraphQL Ecosystem