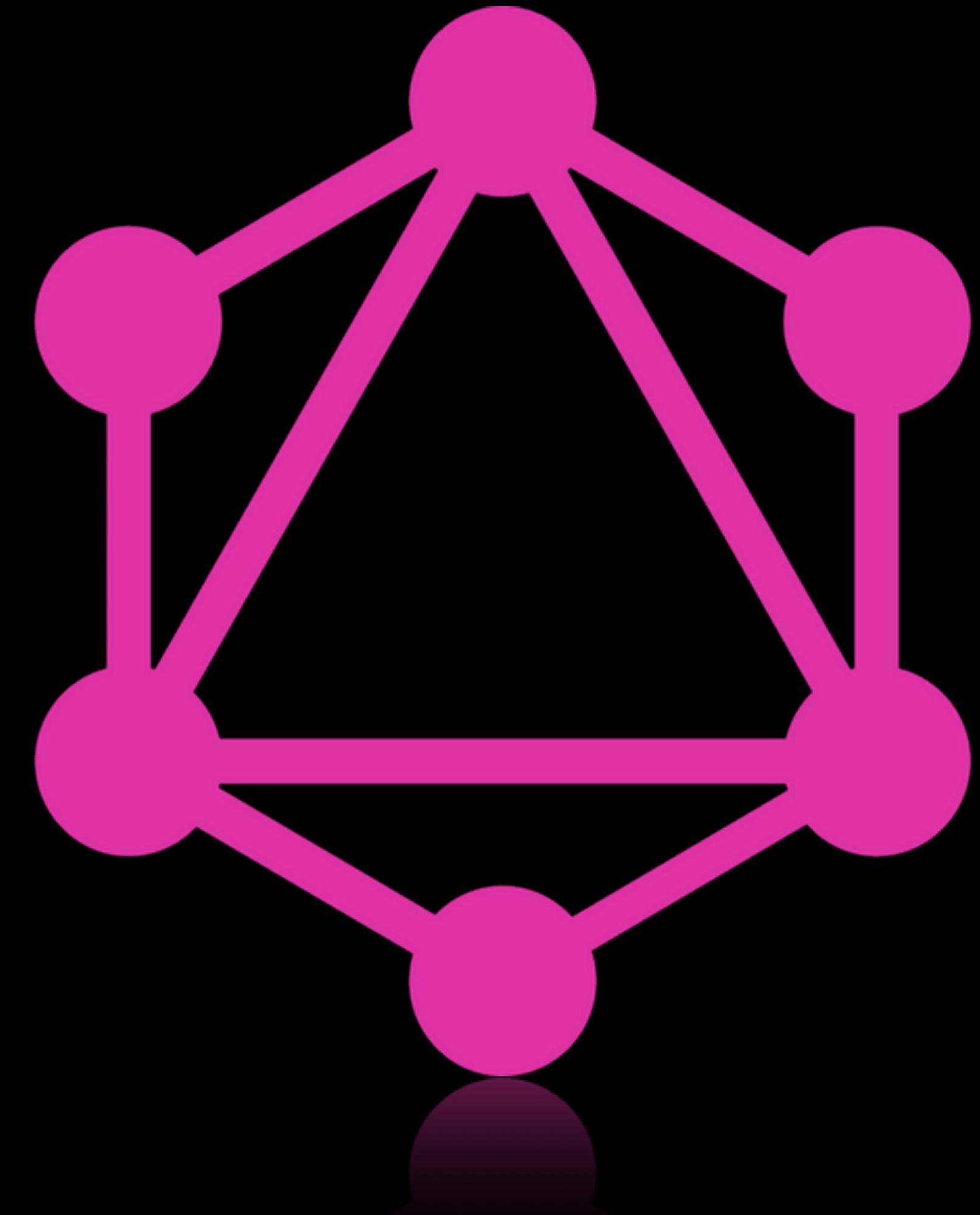


# GraphQL

소프트웨어 마에스트로



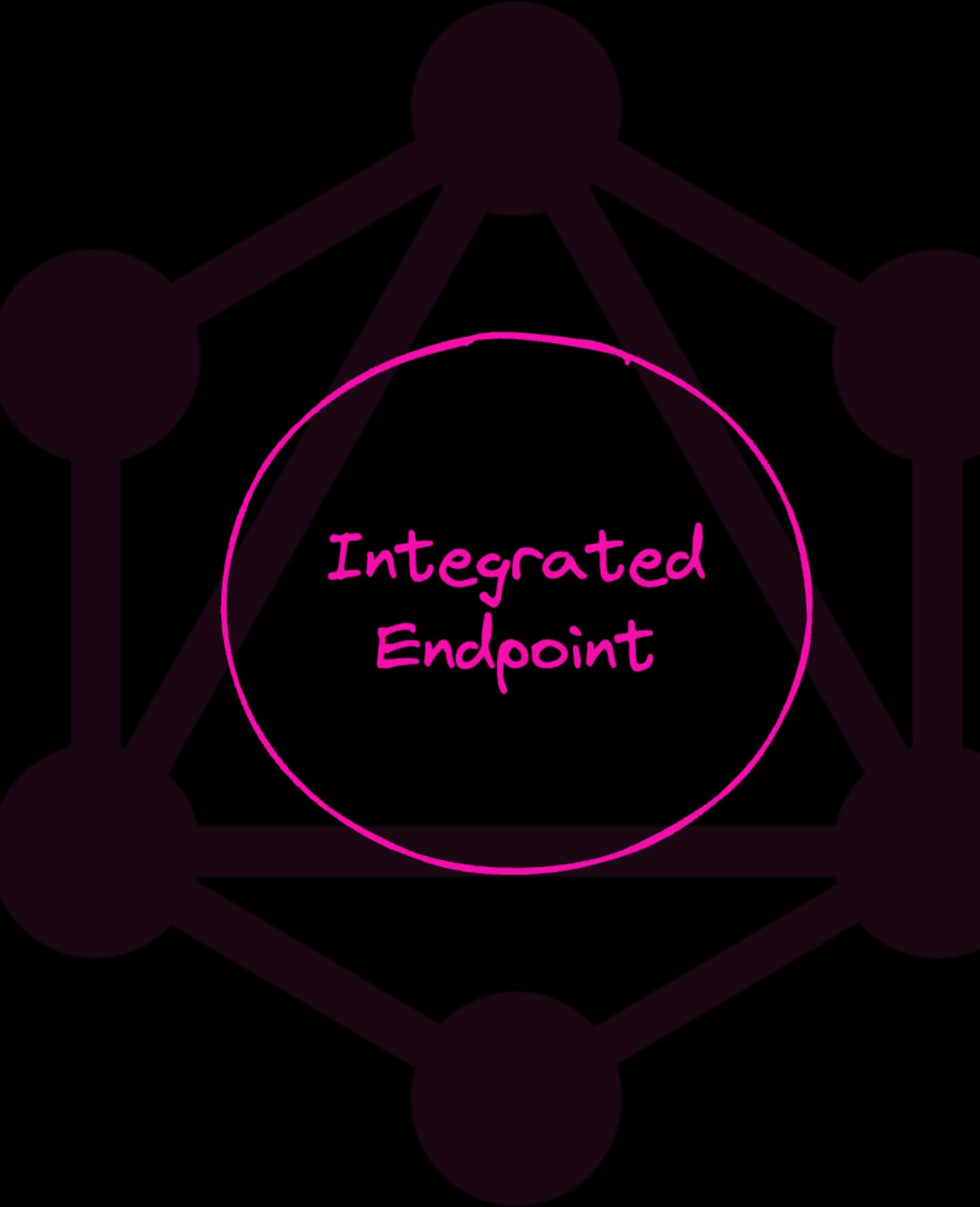
# Problem

Increased  
mobile  
usage

Variety  
of  
Platforms

Communication

# GraphQL



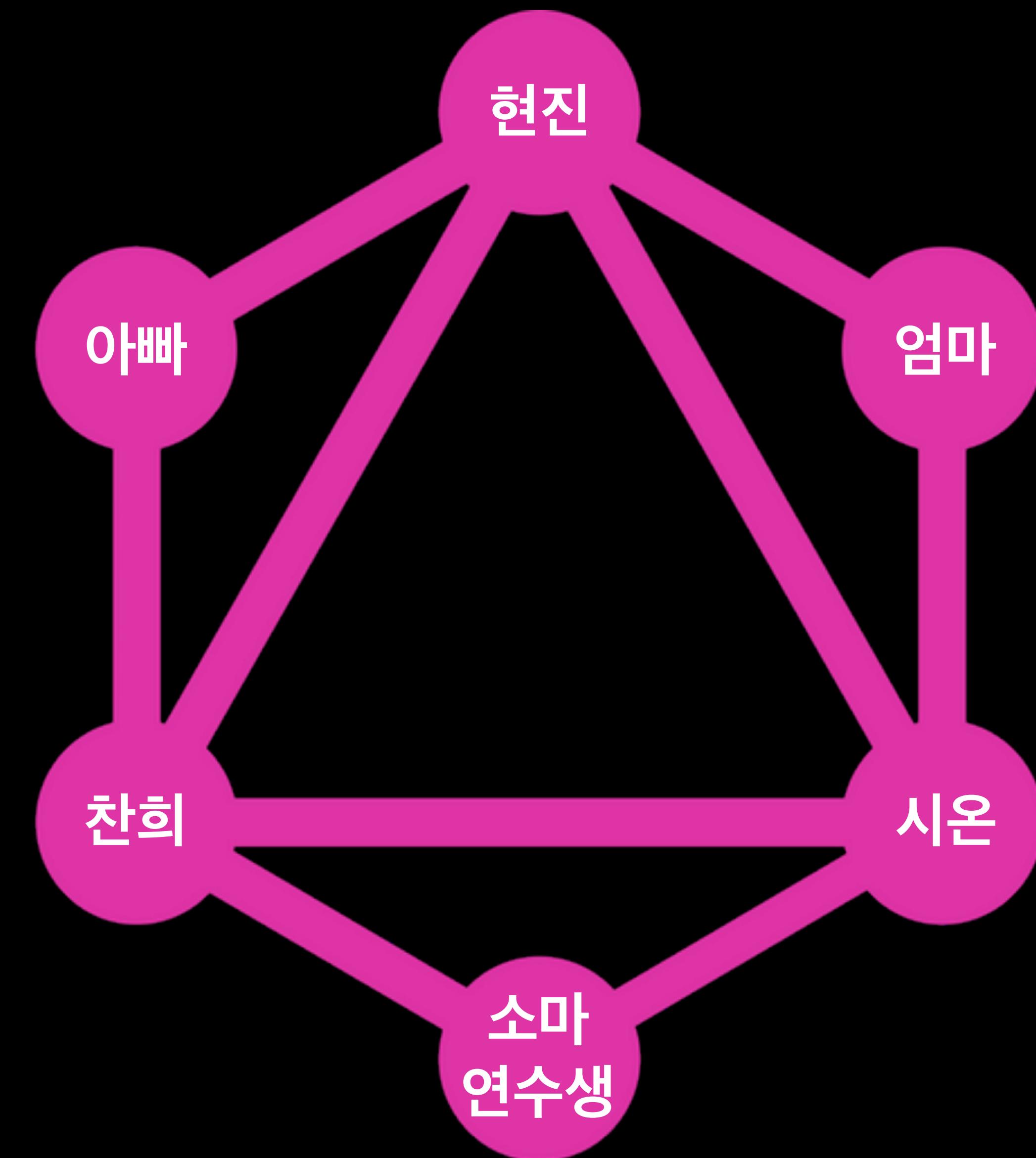
GET ALL  
IN ONCE

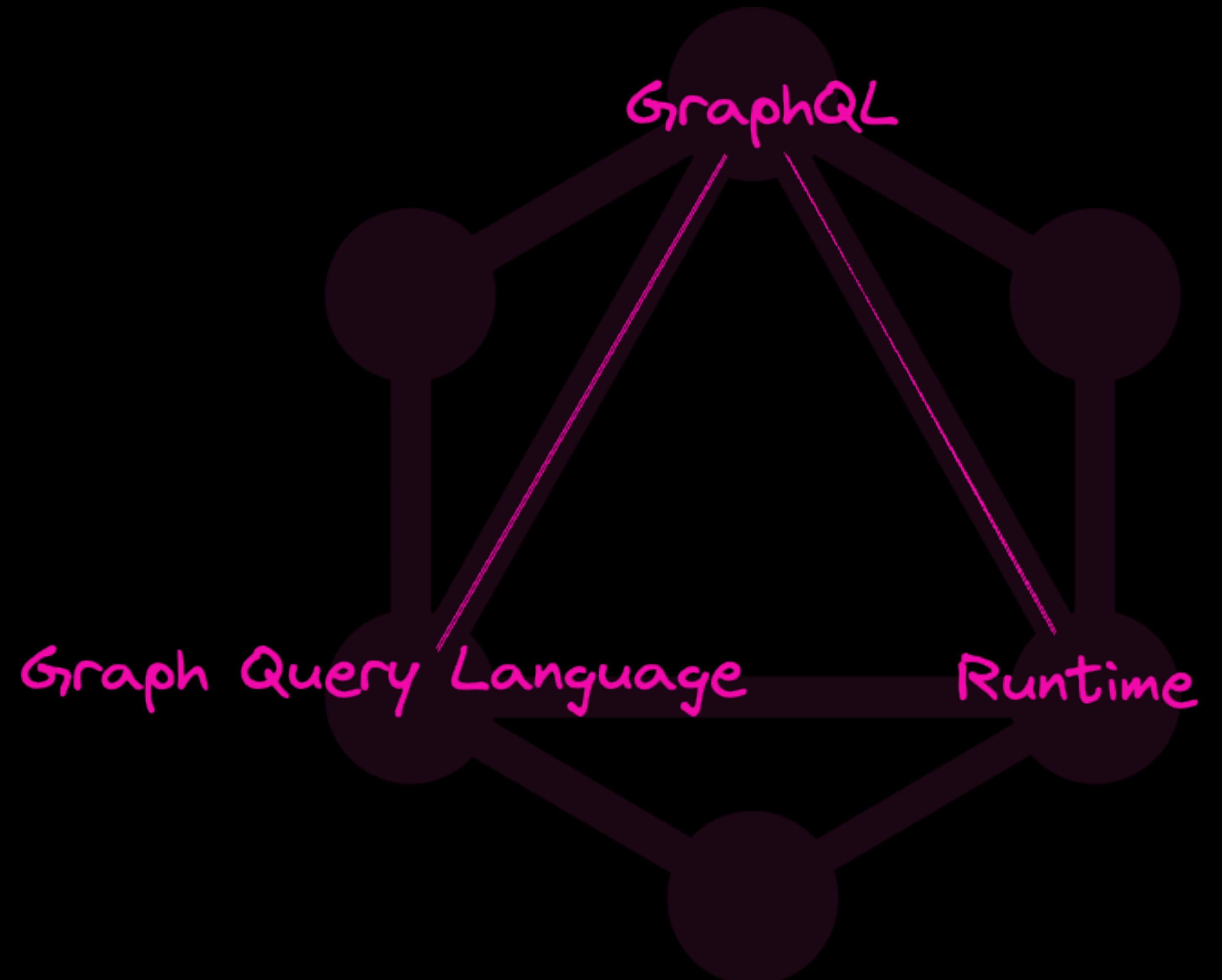
Integrated  
Endpoint

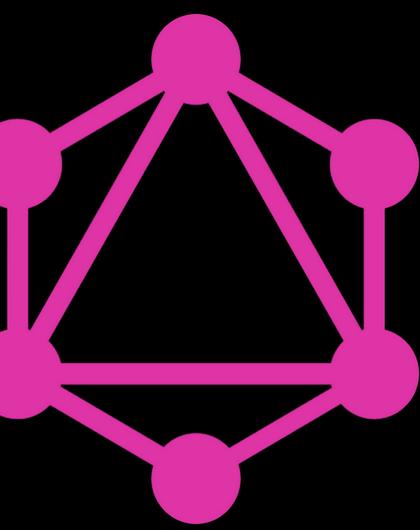
Less  
Communication

# GraphQL

- Facebook이 개발한 오픈 소스 언어(2012)
- 현실 세계의 데이터를 표현하는 가장 적합한 방법이 **Graph**라는 사실에 착안
- GraphQL은 API를 위한 **Query Language**
- GraphQL은 서버사이드에서 실행되는 쿼리를 해석하는 **Runtime**





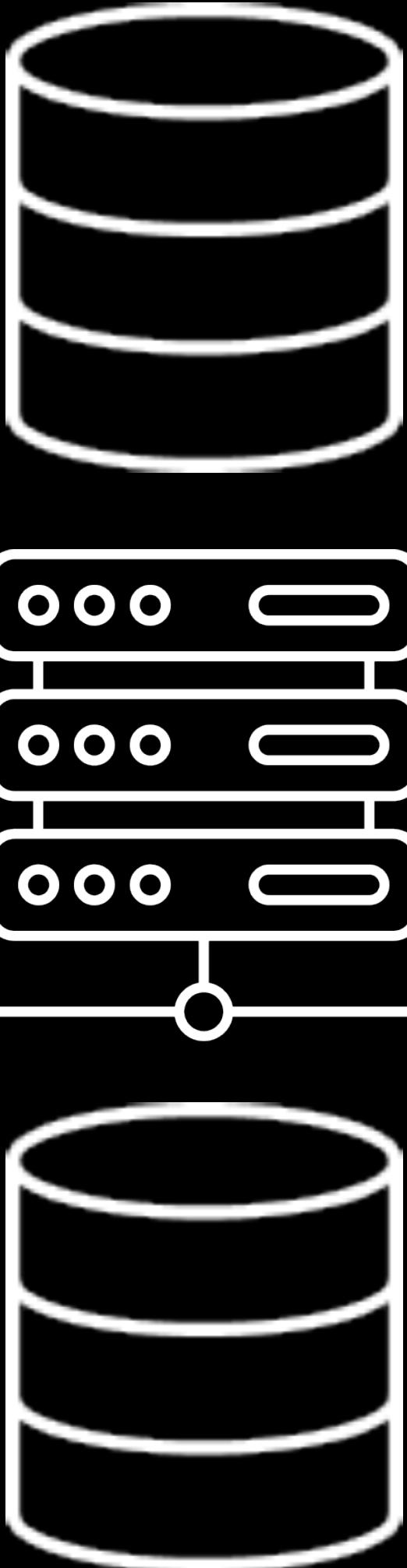


## Graph Query Language

GraphQL 언어는 텍스트 형태로 정확하게  
필요한 데이터를 요청



정확한 데이터 응답(예: JSON 형식)

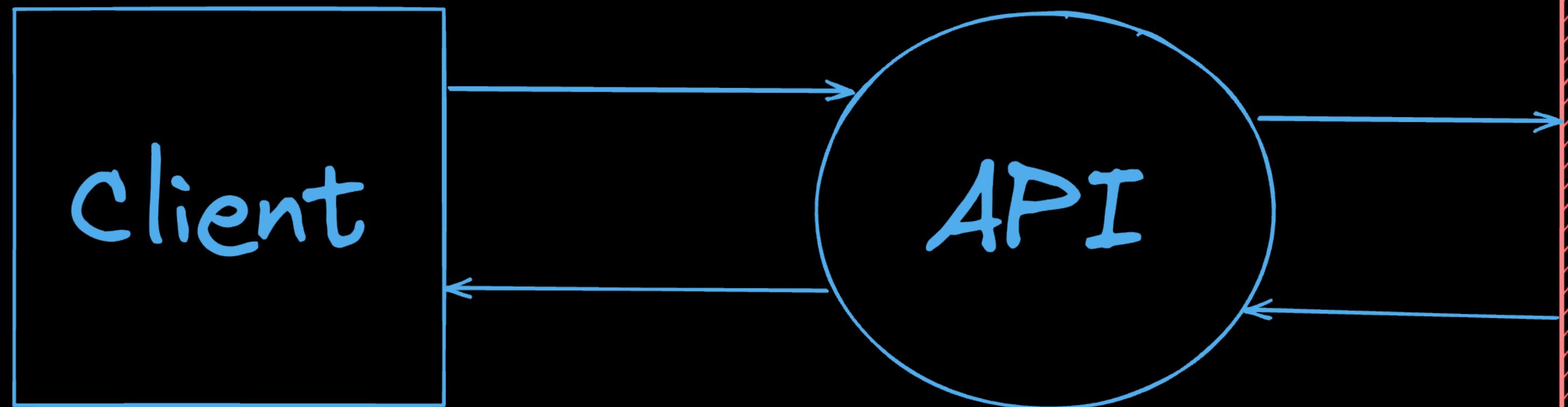


API 사용자

전송채널(예: HTTPS)

API 서비스

API



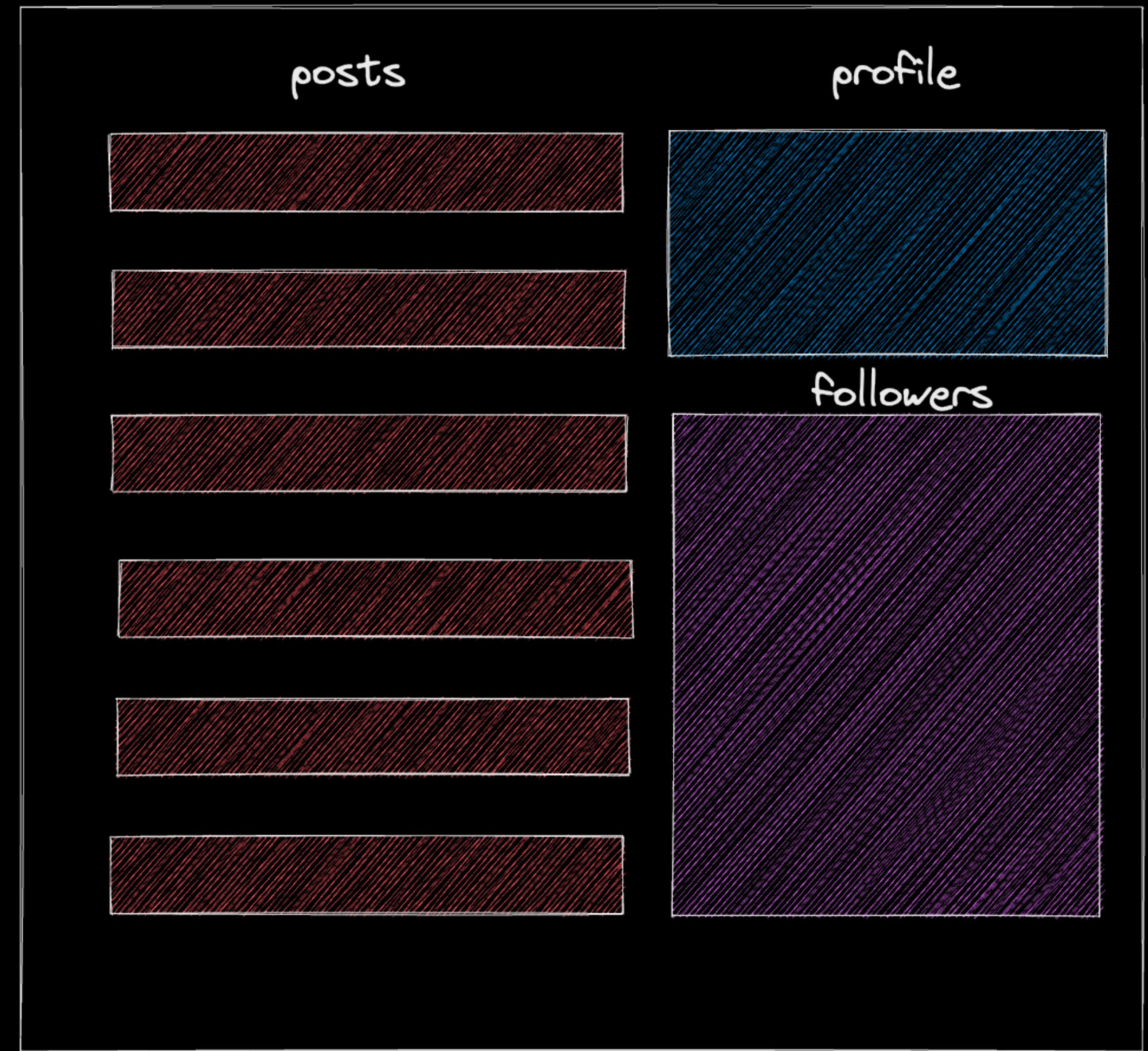
Something  
Amazing



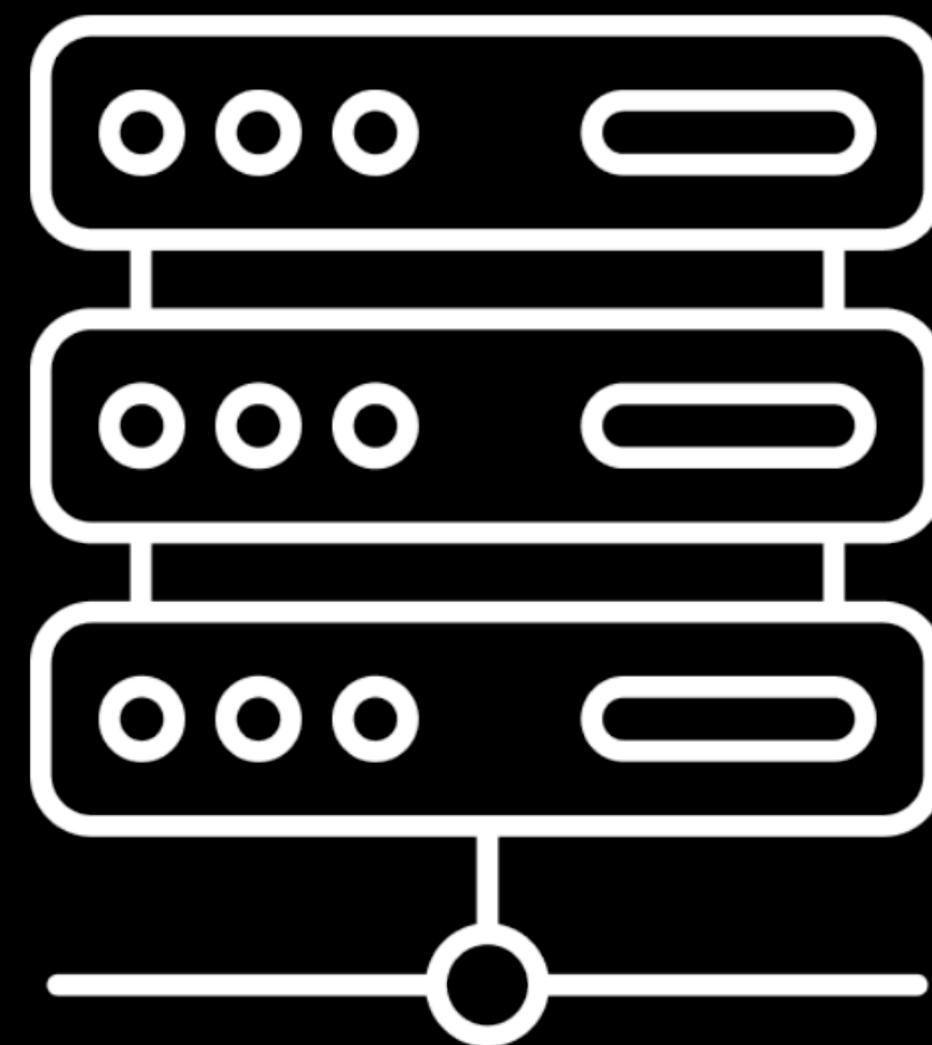
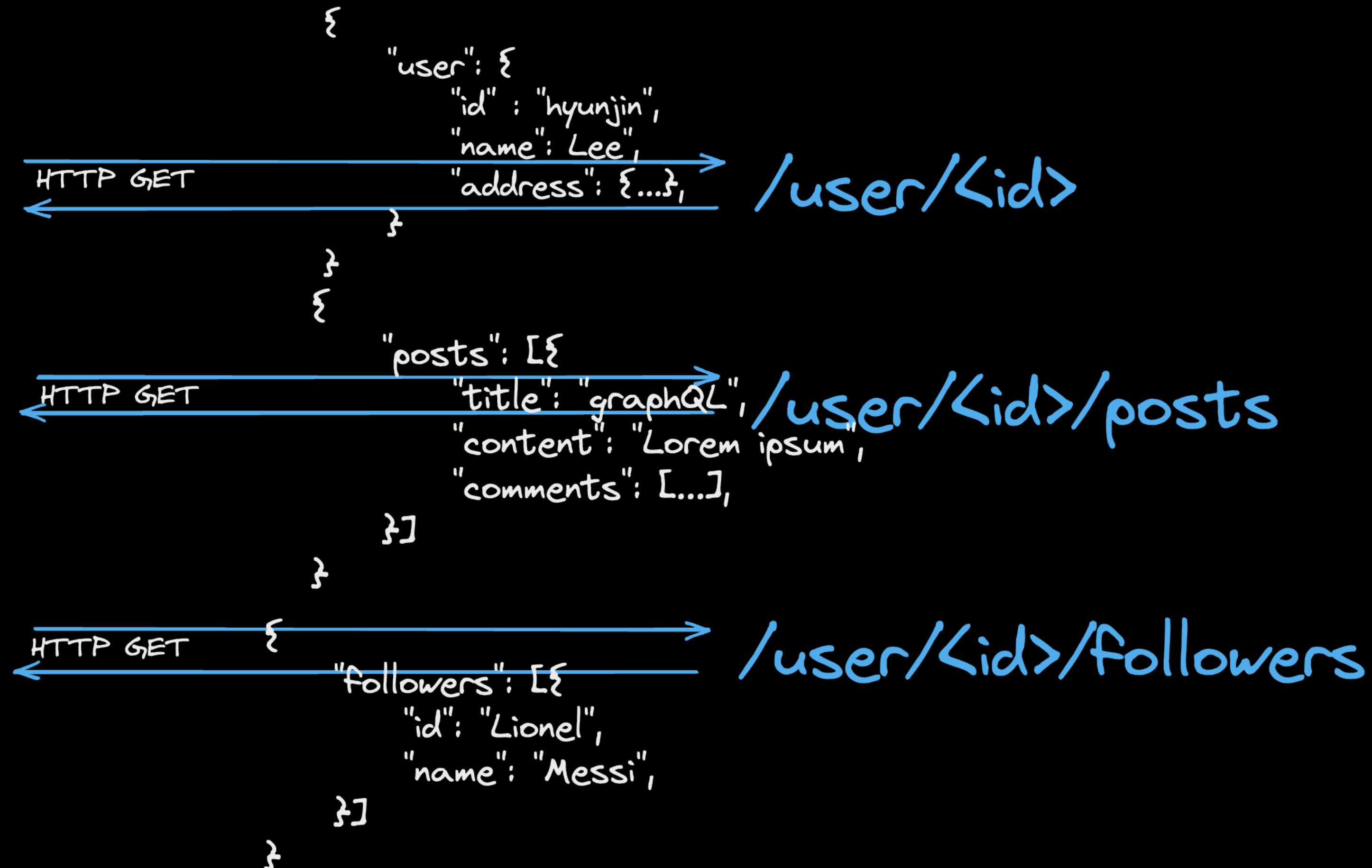
/user/<id>  
/user/<id>/followers

GET  
POST  
PUT  
DELETE

# Example



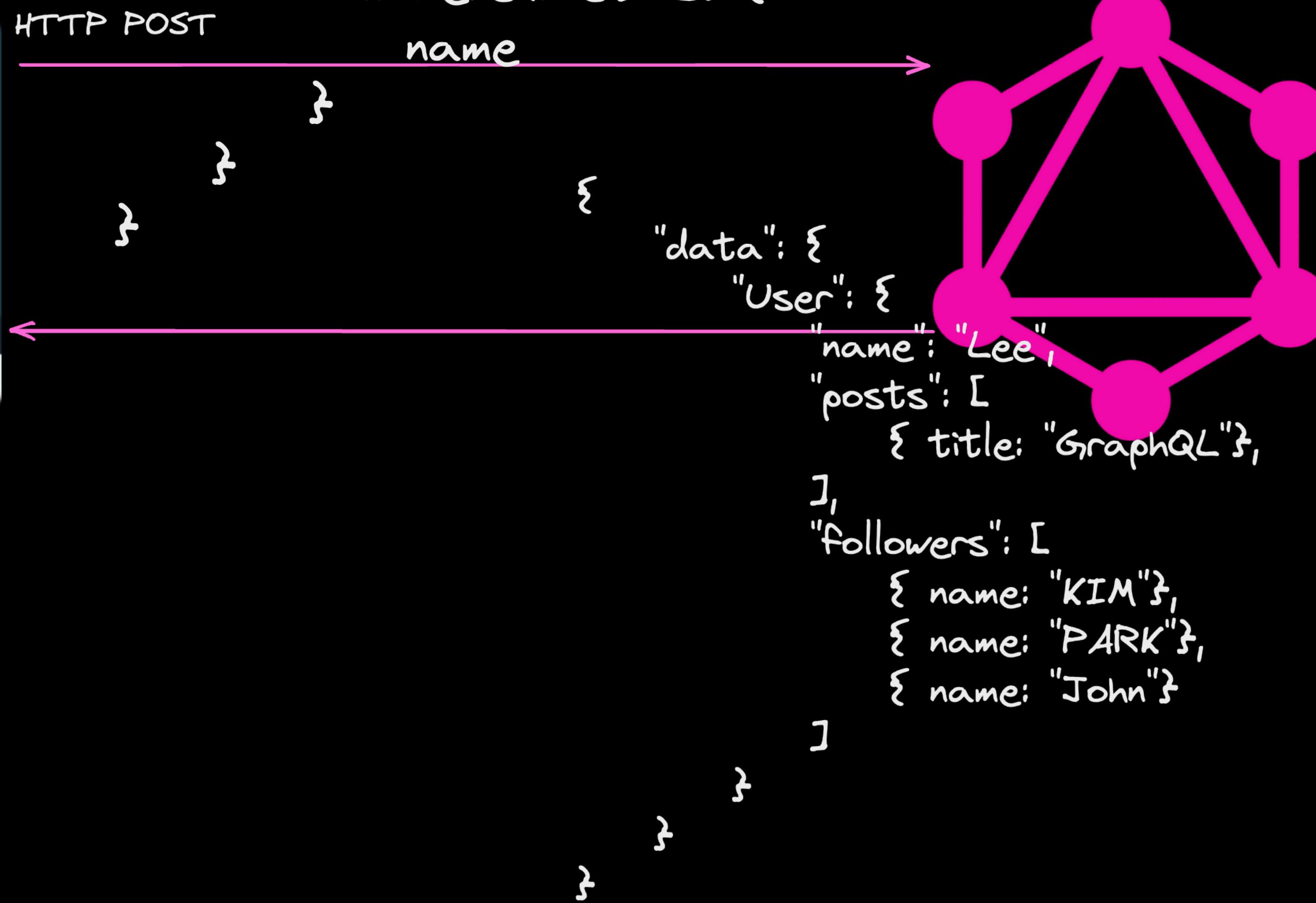
# REST



# GraphQL

```
query {  
  User(id: "hyunjin") {  
    name  
    posts {  
      title  
    }  
    followers(last: 3) {  
      name  
    }  
  }  
}
```

HTTP POST



# REQUEST

```
query {  
  User(id: "hyunjin") {  
    name  
    posts {  
      title  
    }  
    followers(last: 3) {  
      name  
    }  
  }  
}
```

# RESPONSE

```
{  
  "data": {  
    "User": {  
      "name": "Lee",  
      "posts": [  
        { title: "GraphQL"},  
        ...  
      ],  
      "followers": [  
        { name: "KIM"},  
        { name: "PARK"},  
        { name: "John"}  
      ]  
    }  
  }  
}
```

# REST

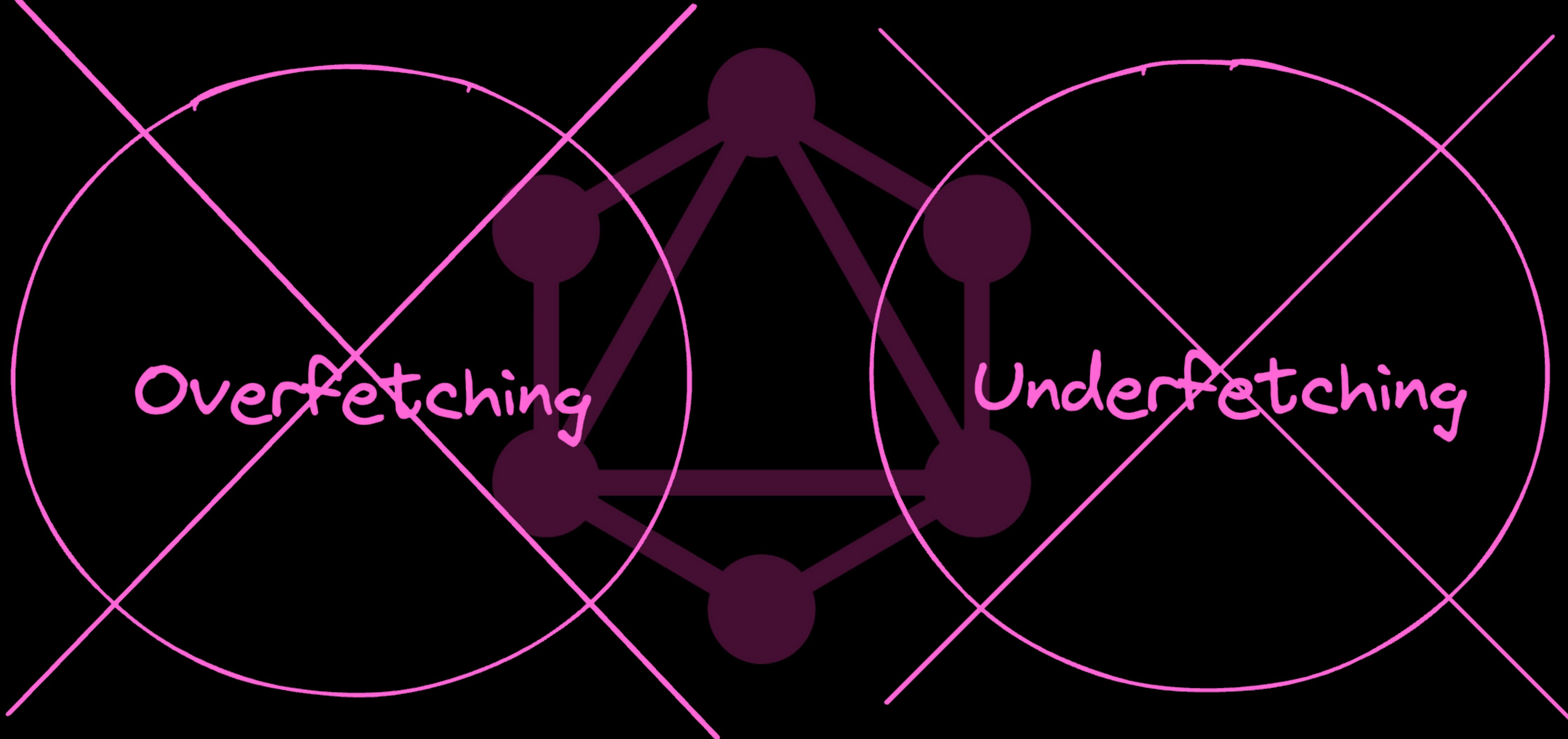
Overfetching

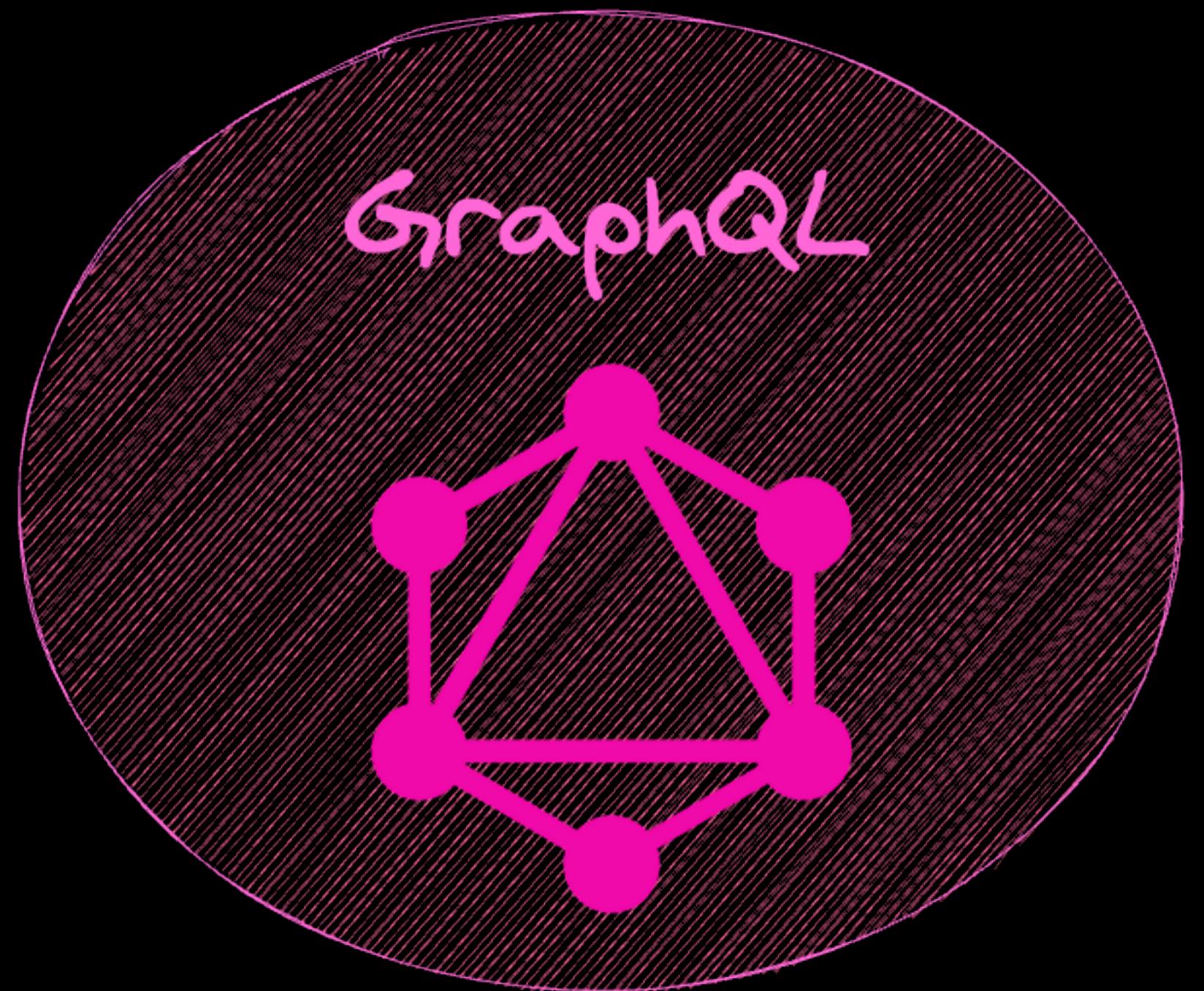
Underfetching

GraphQL

Overfetching

Underfetching



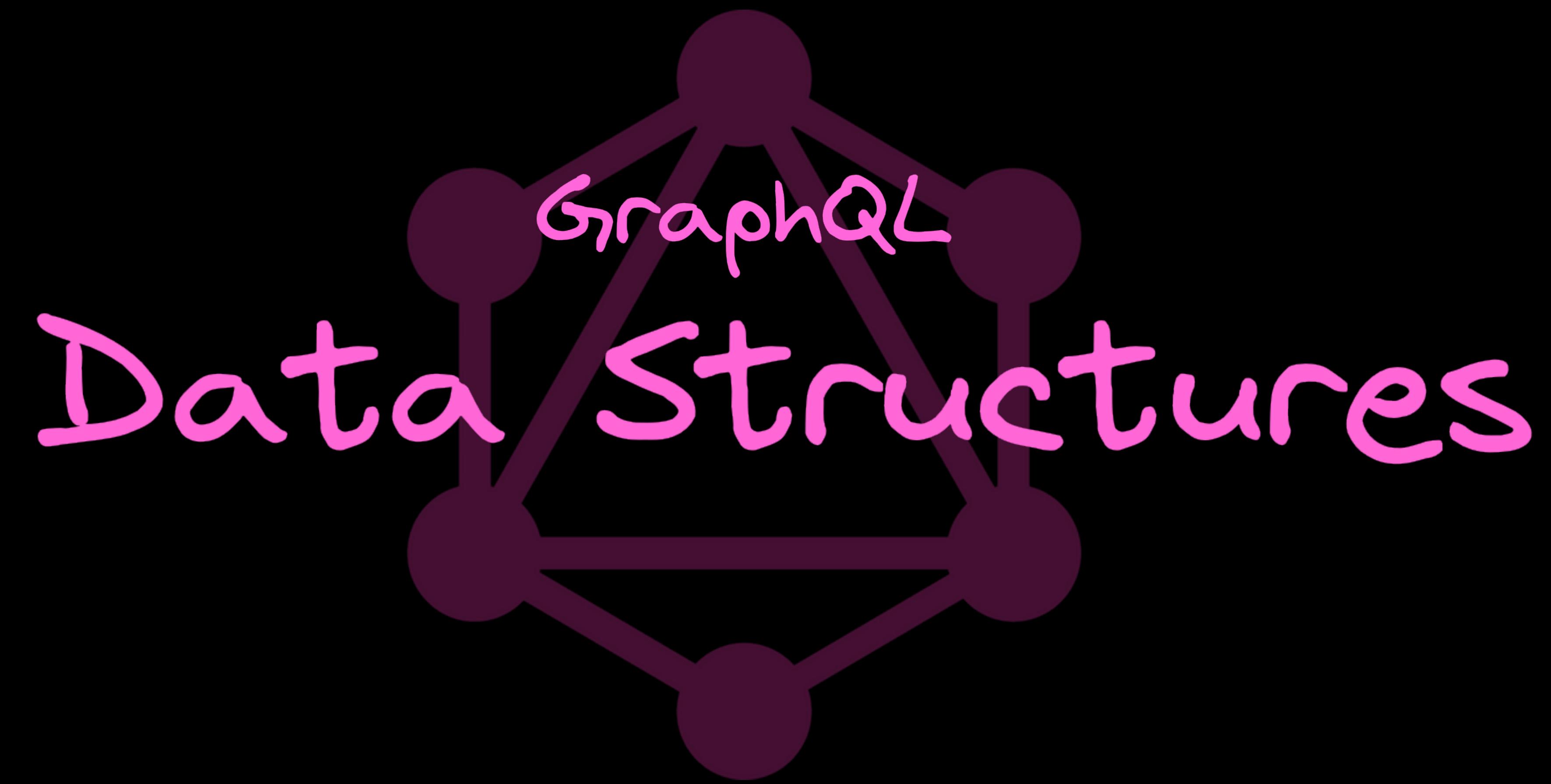


VS



# Communication





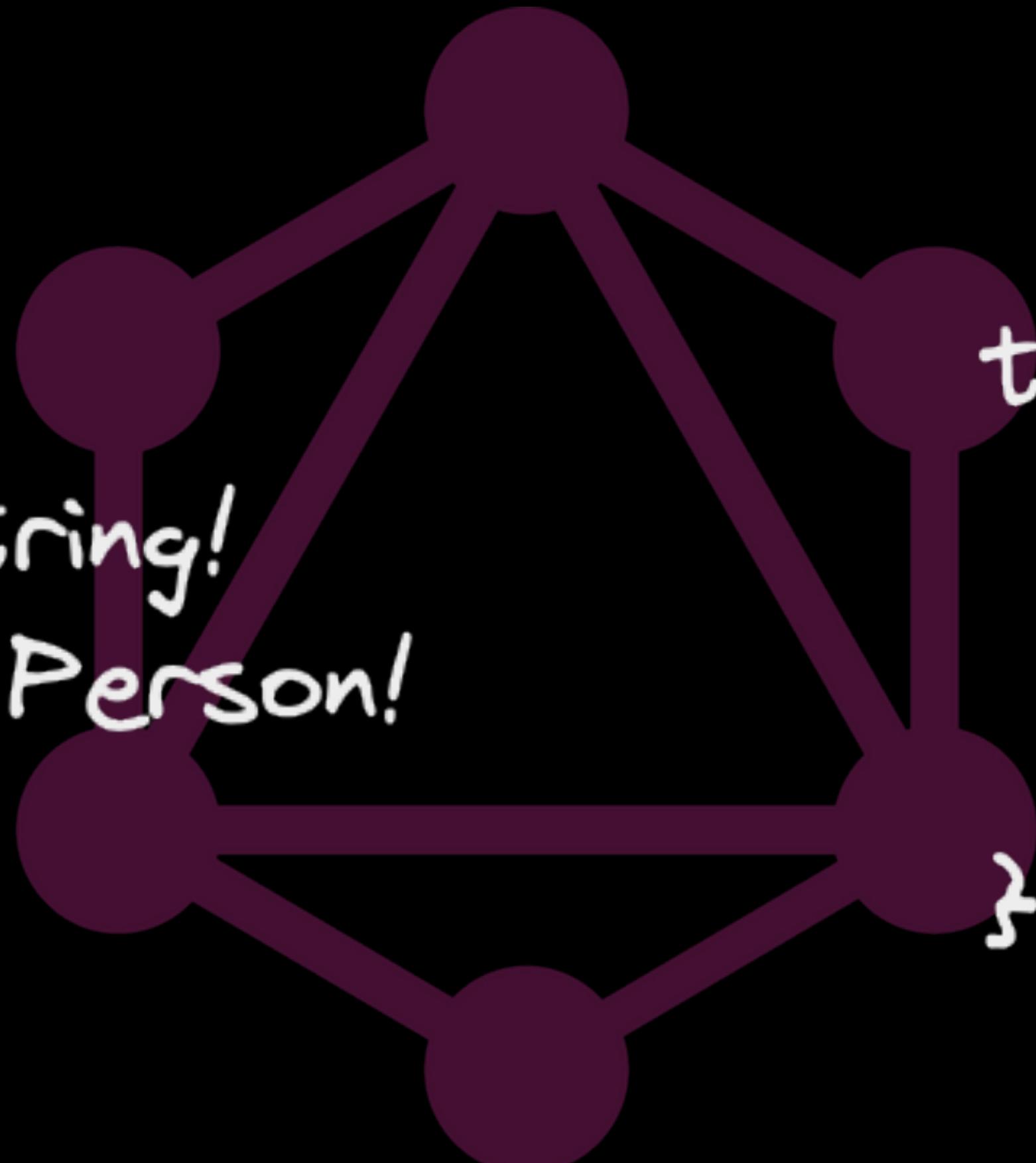
The logo features the word "GraphQL" in a pink, handwritten-style font positioned above the word "Data Structures". The "Q" in "GraphQL" overlaps with the "D" in "Data". The "S" in "Data" overlaps with the "t" in "Structures". Both words are rendered in a vibrant pink color. The background is black, and the text is surrounded by a network of dark purple circular nodes connected by lines, forming a graph structure.

# GraphQL

# Data Structures

# GraphQL Schema

```
type Post {  
    title: String!  
    author: Person!  
}
```



```
type Person {  
    name: String!  
    age: Int!  
    posts: [Post!]!
```

Frontend

Schema

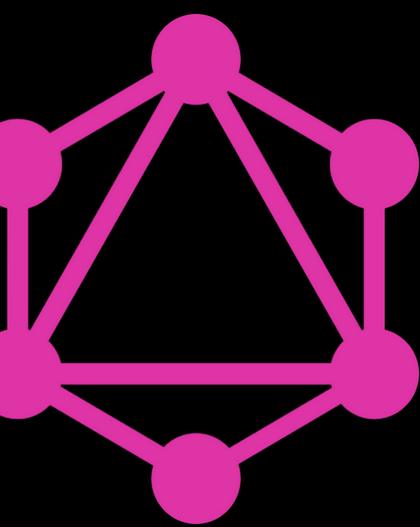


Backend

# GraphQL Resolvers

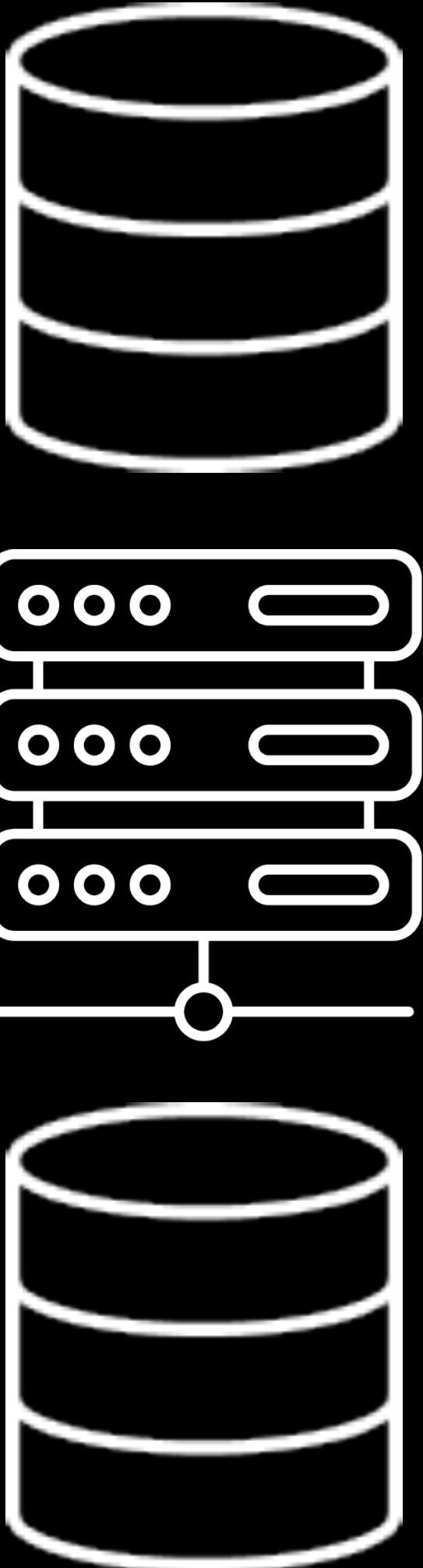
- 스키마의 각 필드는 리졸버 함수와 연동된다.
- 리졸버 함수는 어디서 어떻게 데이터를 가져올지를 지시한다.





*Graph Query Language*

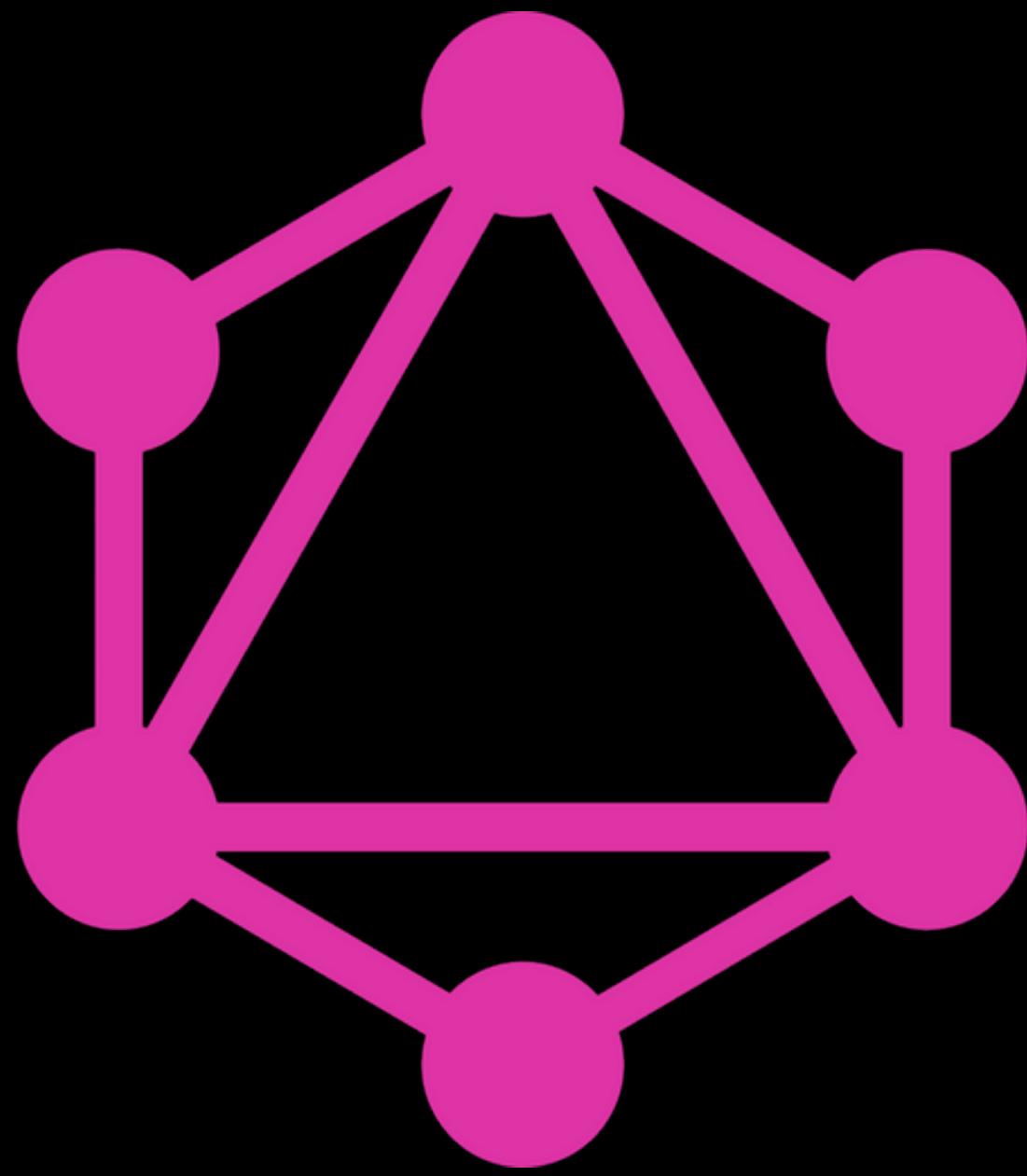
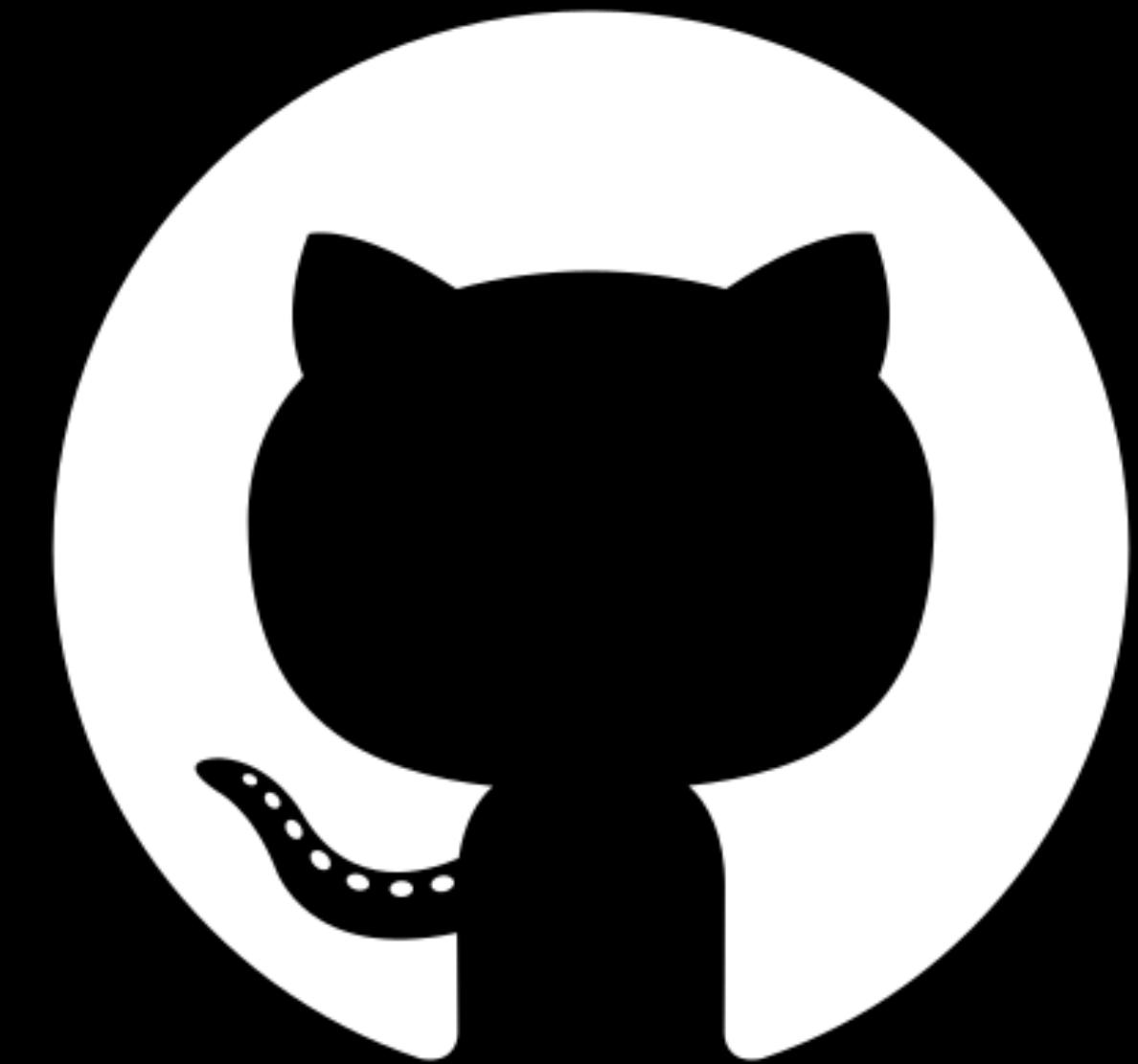
GraphQL 언어는 텍스트 형태로 정확하게  
필요한 데이터를 요청



API 사용자

전송채널(예: HTTPS)

API 서비스



Testing Comment Mutation · Issues

github.com/hyunjinee/GraphQL/issues/1

Pull requests Issues Marketplace Explore

hyunjinee / GraphQL Public

Code Issues 1 Pull requests Actions Projects Wiki Security Insights Settings

# Testing Comment Mutation #1

Open hyunjinee opened this issue 13 days ago · 2 comments

hyunjinee commented 13 days ago · edited

Owner

...

### Find Repo ID

```
{  
  repository(name:"GrahpQL", owner:"hyunjinee") {  
    id  
  }  
}
```

### Find ISSUE in Repo

```
query GetIssueInfo {  
  repository (owner:"hyunjinee", name: "GraphQL") {  
    issue(number: 1) {  
      id  
      title  
    }  
  }  
}
```

### Add Comment to Issue

```
mutation AddCommentToIssue {  
  addComment(input:{  
    subjectId:"I_kwD0H0hUV85Px2eu",  
    body:"안녕하세요 이슈에 대한 테스트입니다."}  
}
```

Assignees

hyunjinee

Labels

good first issue

Projects

None yet

Milestone

No milestone

Development

Create a branch for this issue or link a pull request.

Notifications

Customize

Unsubscribe

You're receiving notifications because you're watching this repository.

1 participant

<https://docs.github.com/en/graphql/overview/explorer>

```
mutation AddCommentToIssue {  
  addComment(input:{  
    subjectId:"I_kwDOH0hUV85Px2eu",  
    body:"안녕하세요 이현진입니다." # 여러분의 이름으로 바꿔주세요  
  }) {  
    commentEdge {  
      node {  
        createdAt  
      }  
    }  
  }  
}
```

<https://github.com/hyunjinee/GraphQL/issues/1>

Testing Comment Mutation · Issues · GitHub

github.com/hyunjinee/GraphQL/issues/1

Open Testing Comment Mutation #1

hyunjinee opened this issue 13 days ago · 1 comment

```
commentEdge {
  node {
    createdAt
  }
}
```

hyunjinee self-assigned this 13 days ago

hyunjinee added the `good first issue` label 13 days ago

hyunjinee commented 13 days ago

Owner Author ⋮

안녕하세요 이현진입니다.

Write Preview

H B I ↵ ↶ ↷ ↸ ↹ ↺

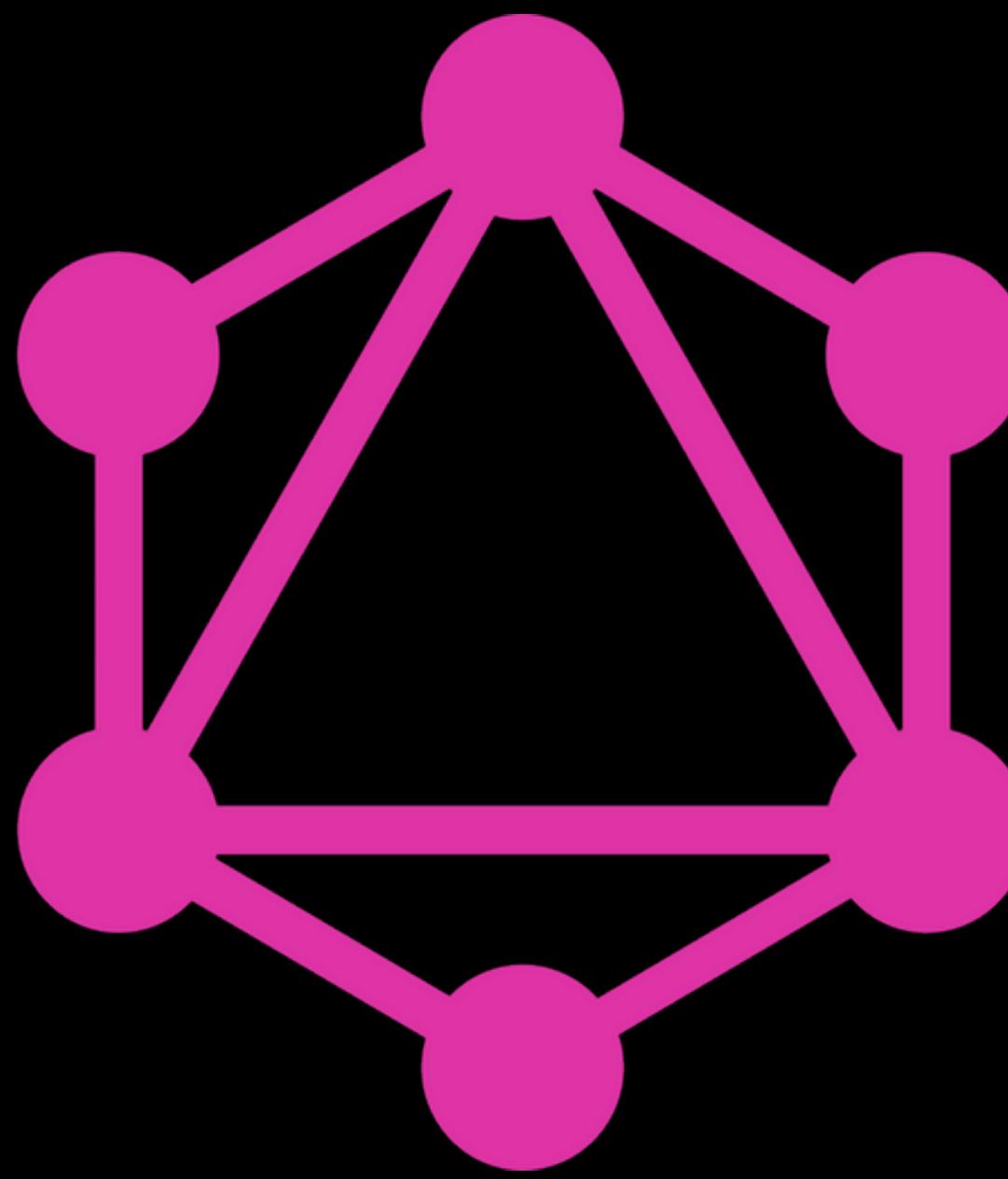
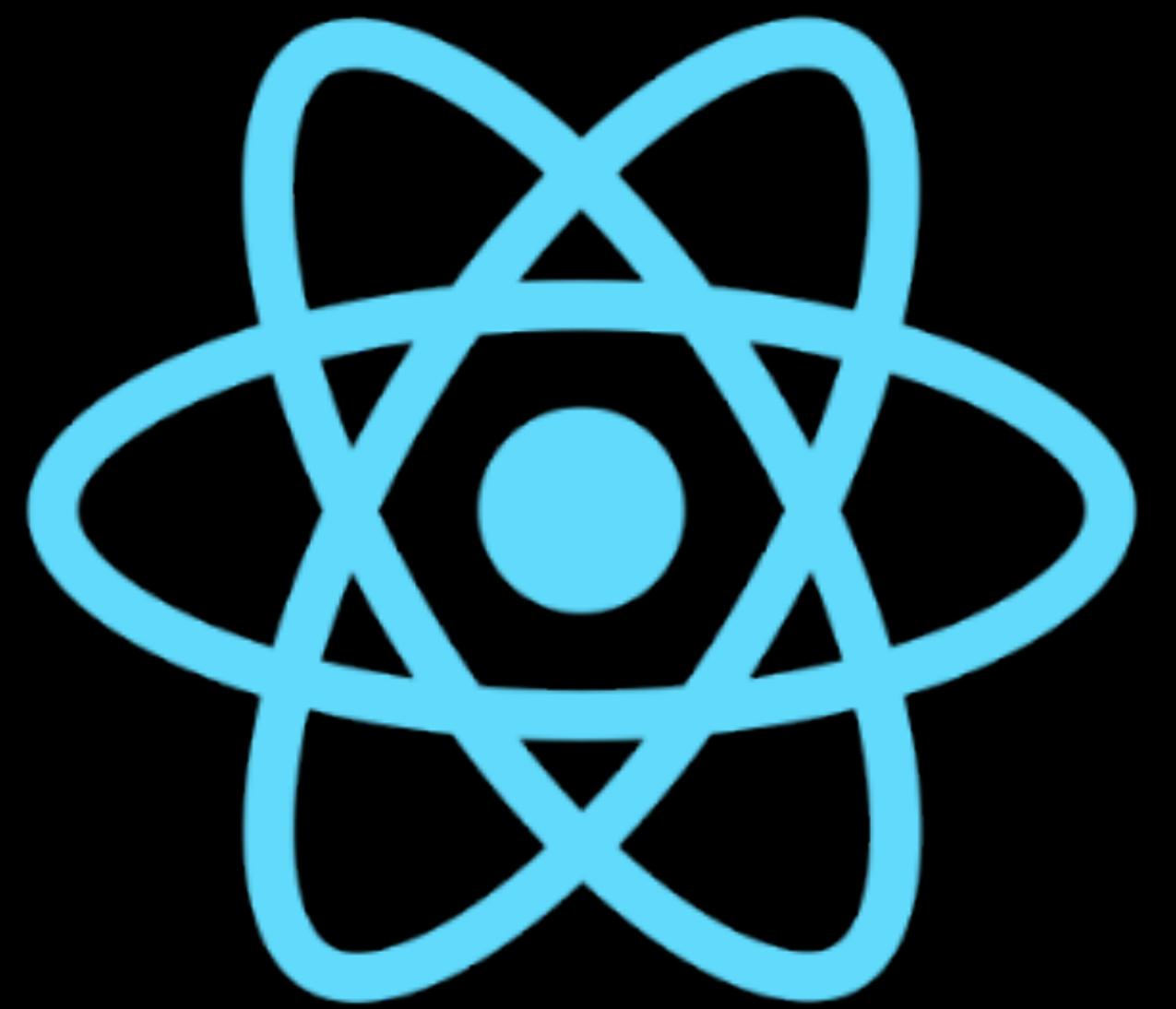
Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Close issue Comment

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About



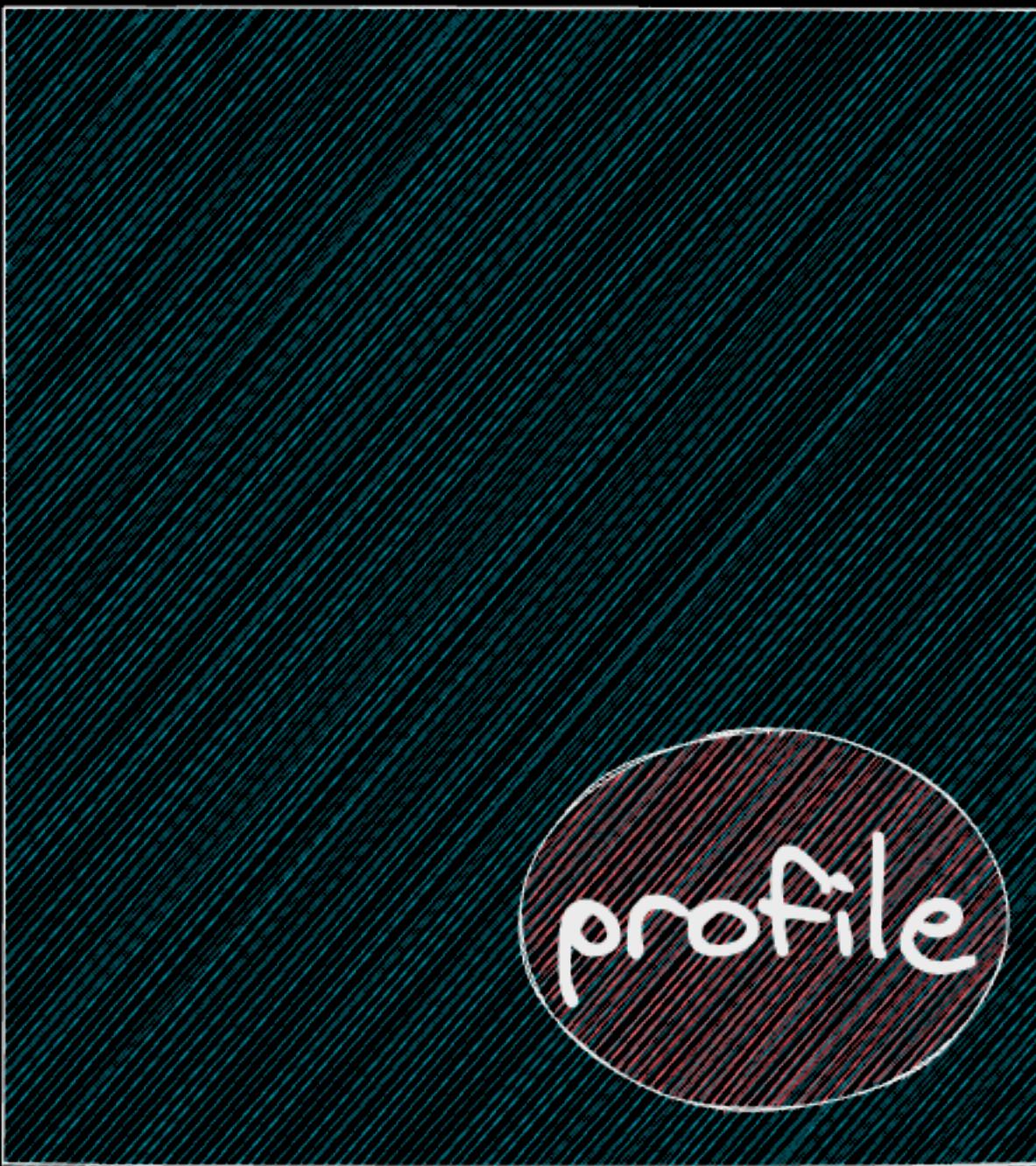
# Blog

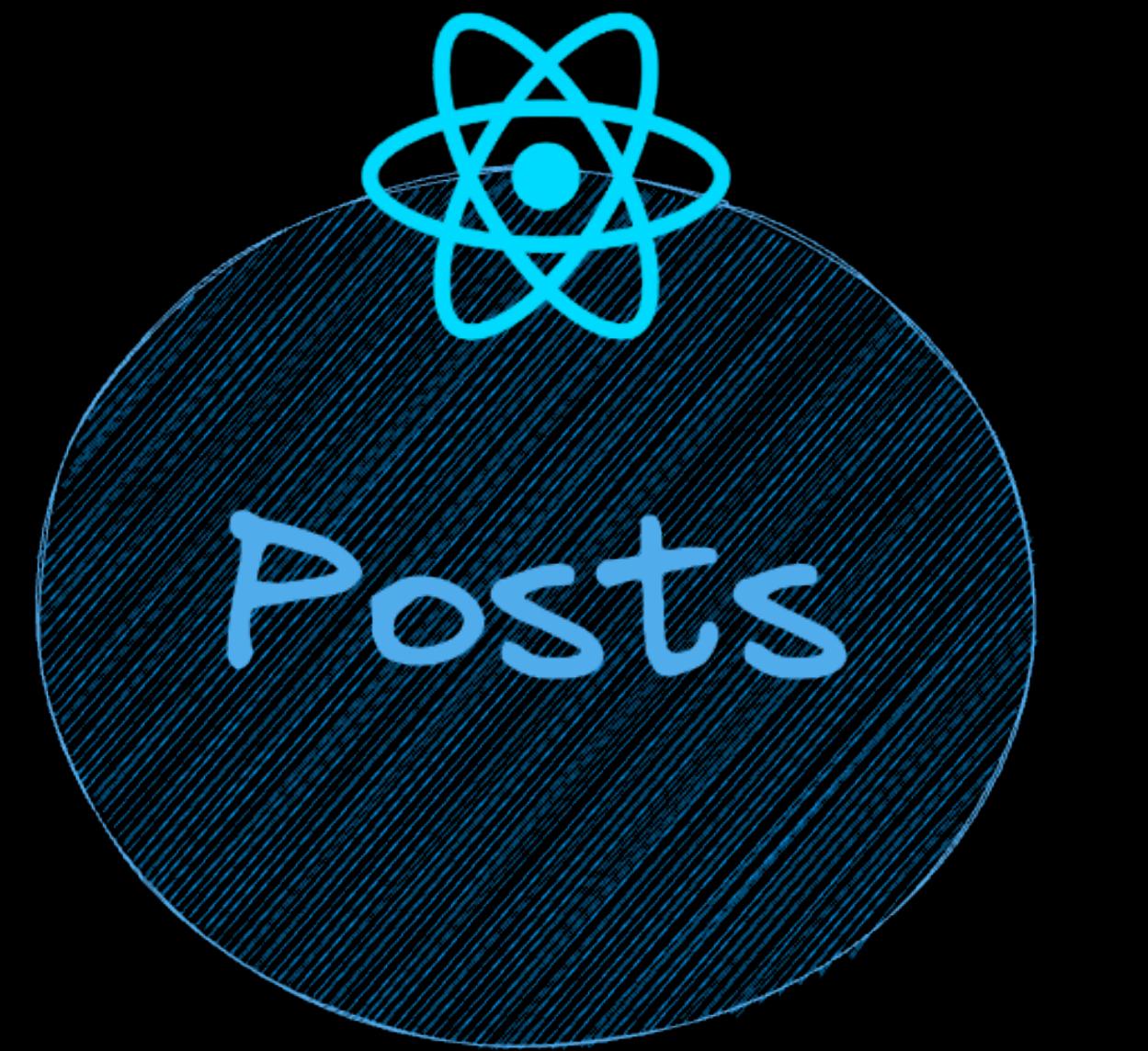
## Posts



# Blog

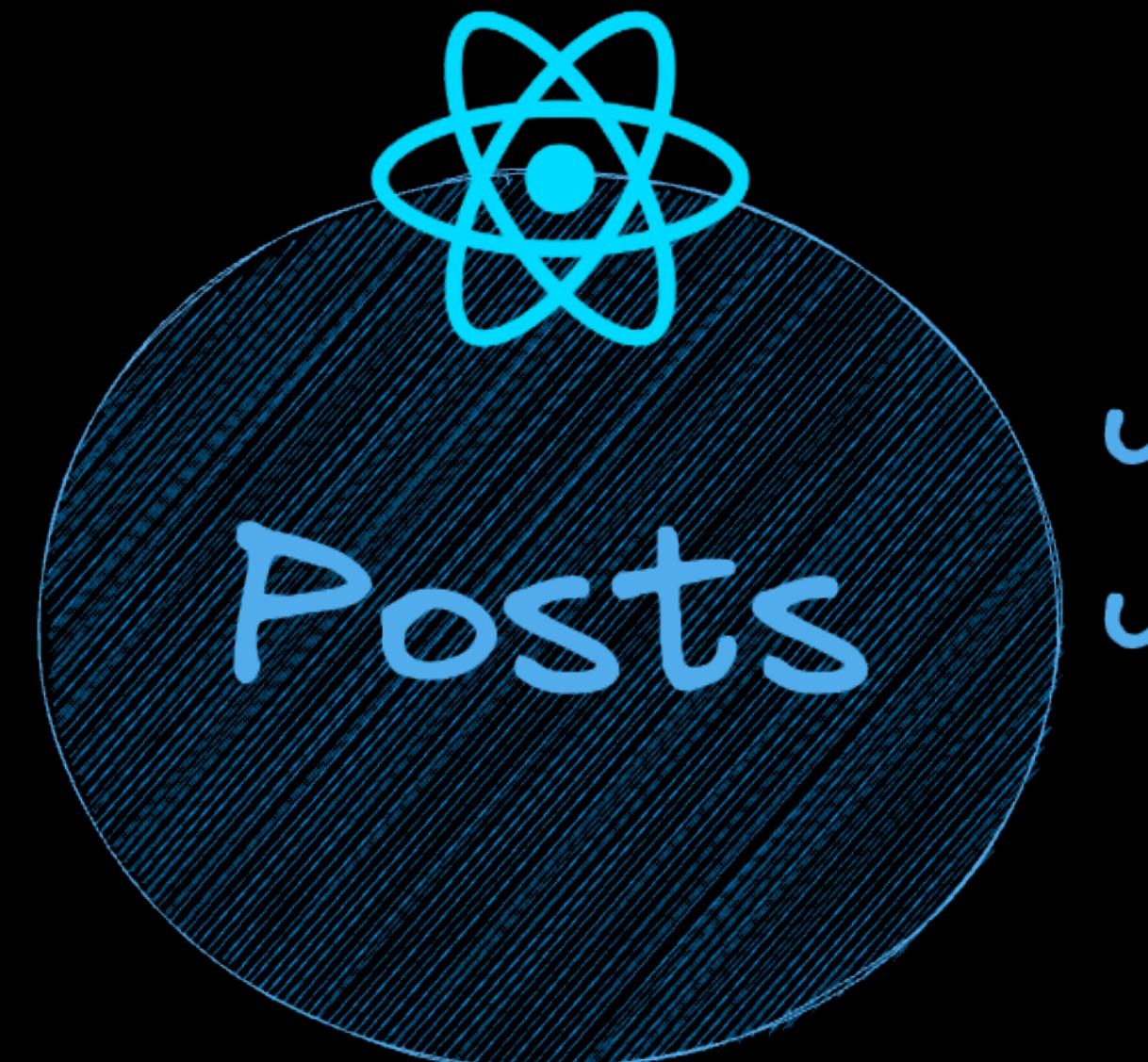
## Posts



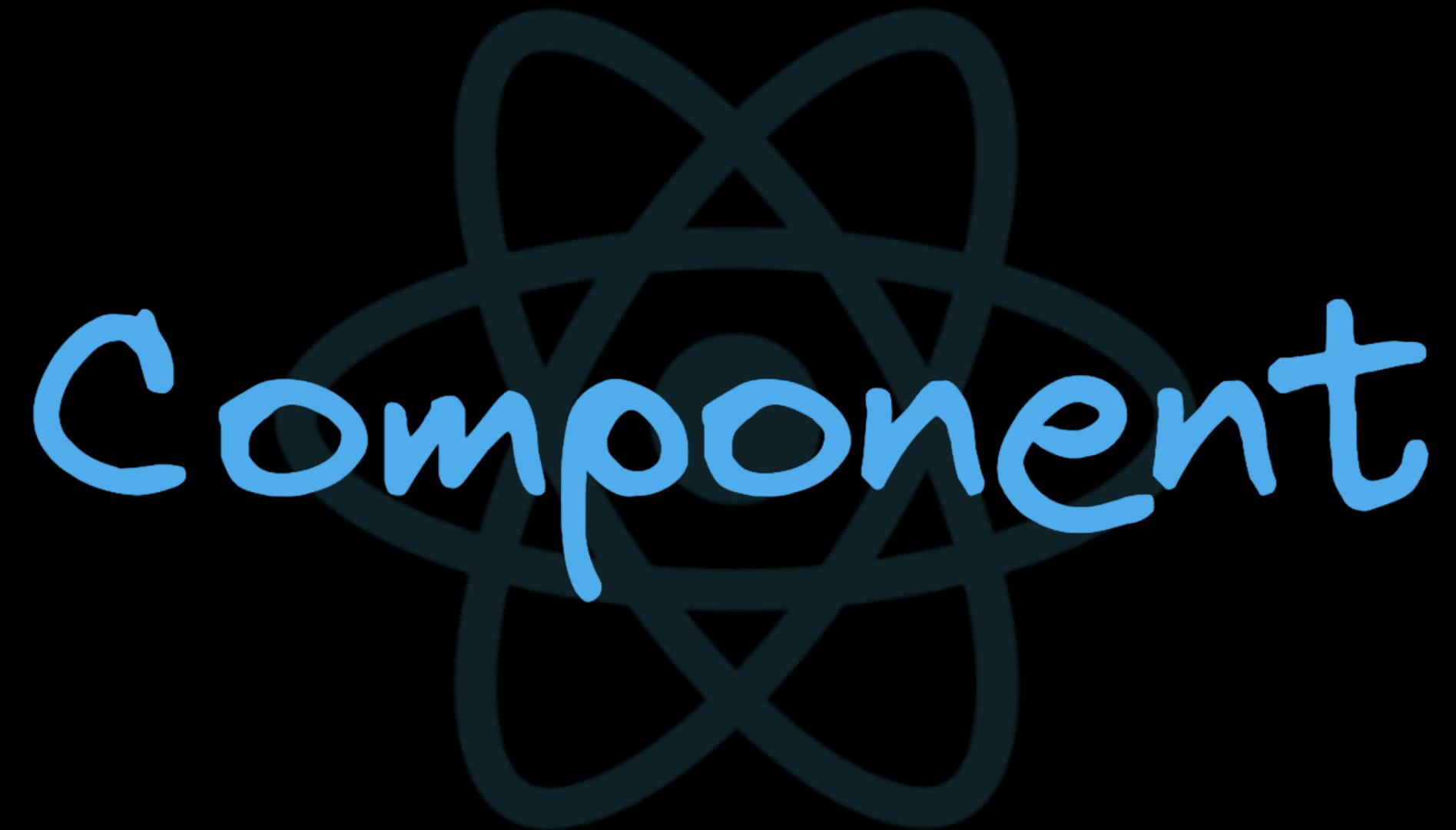


Network  
Waterfall

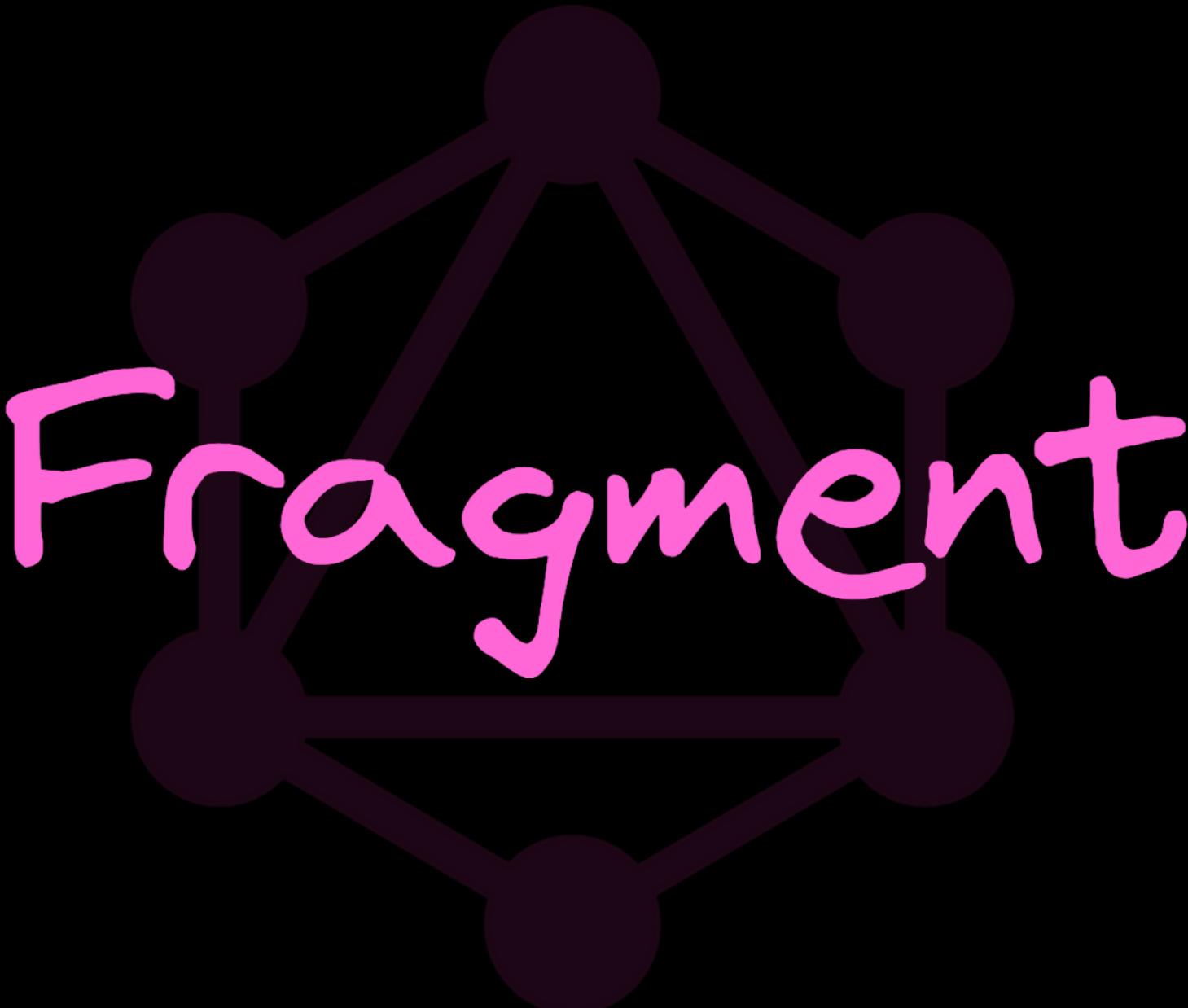
Profile



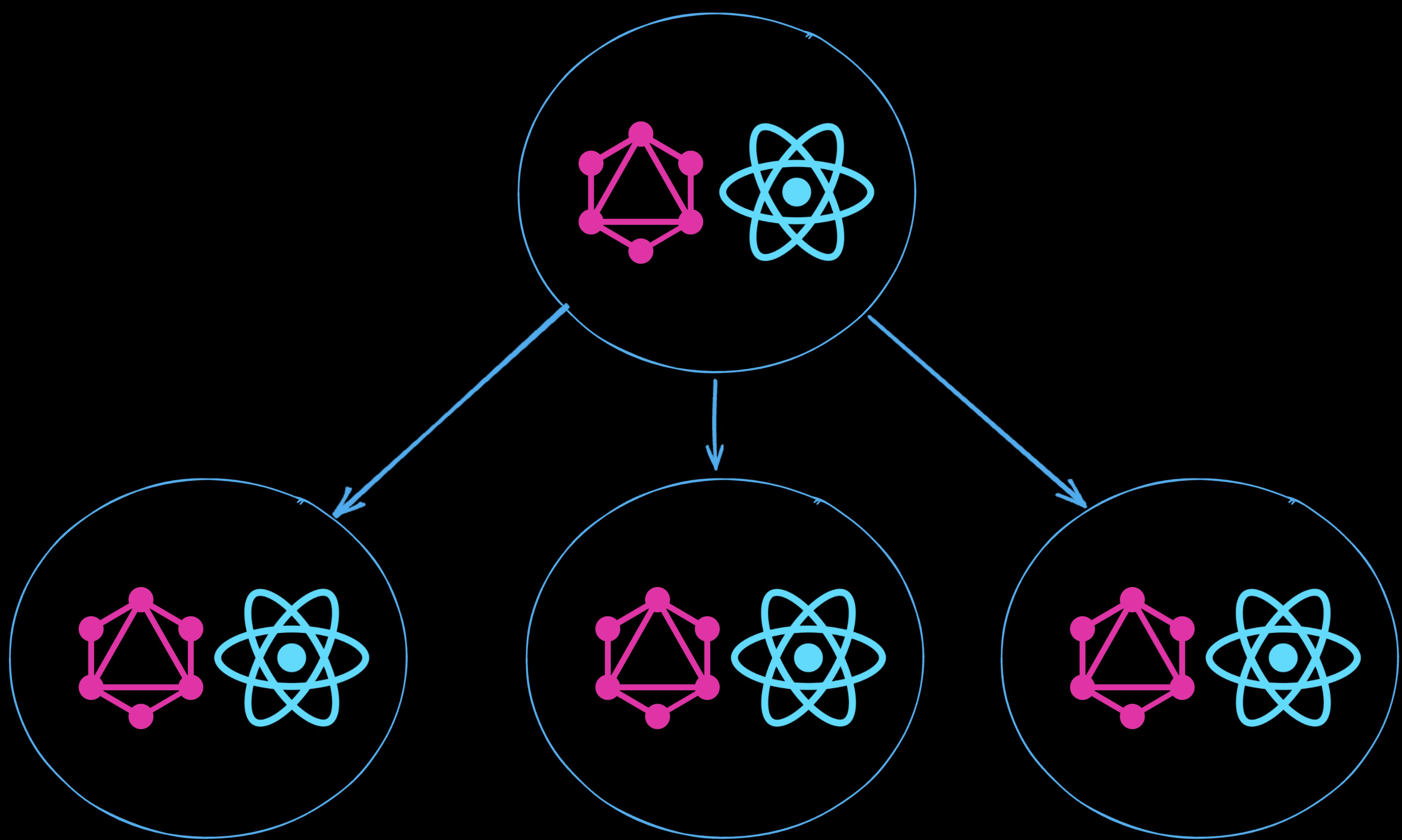
usePosts()  
useProfile()

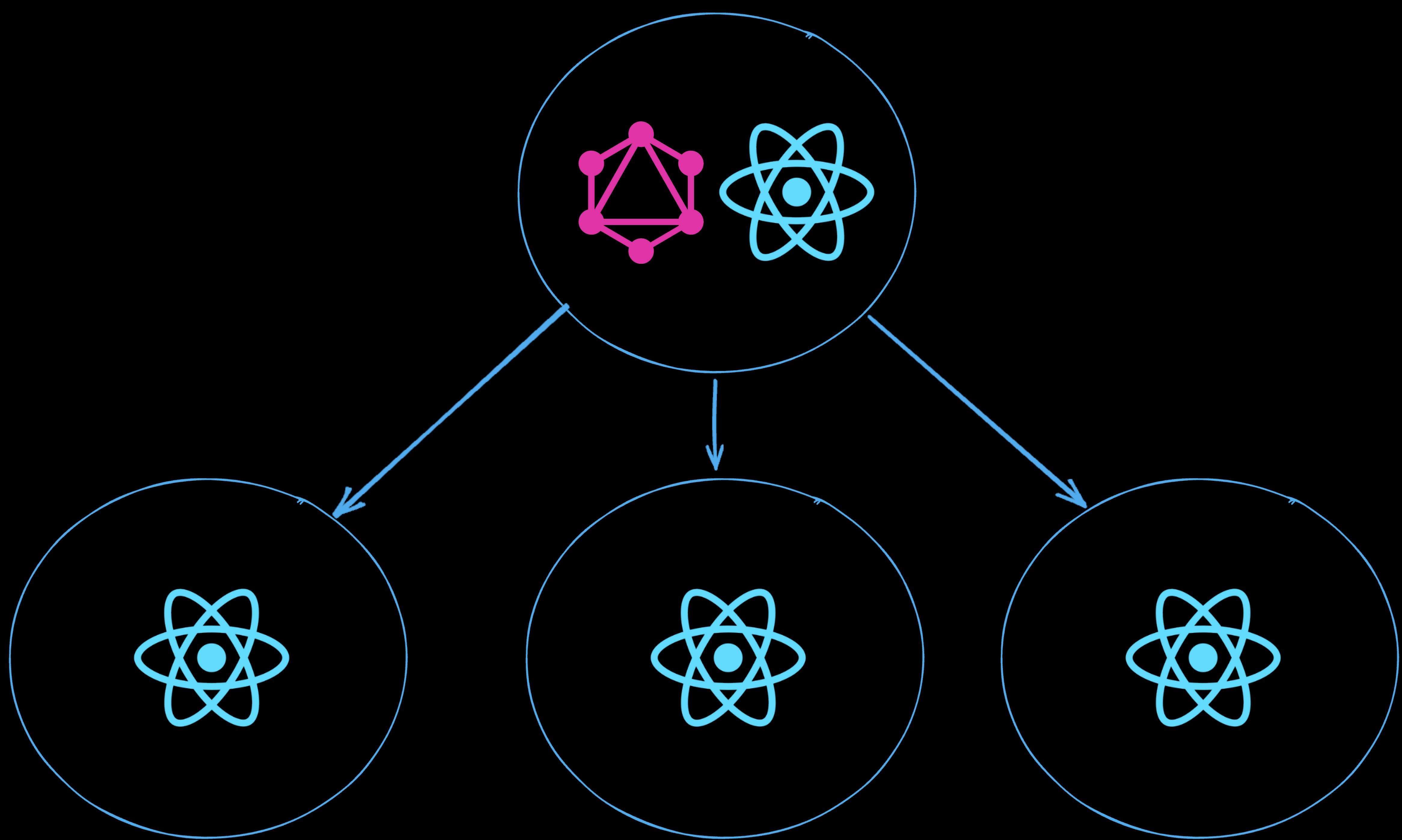


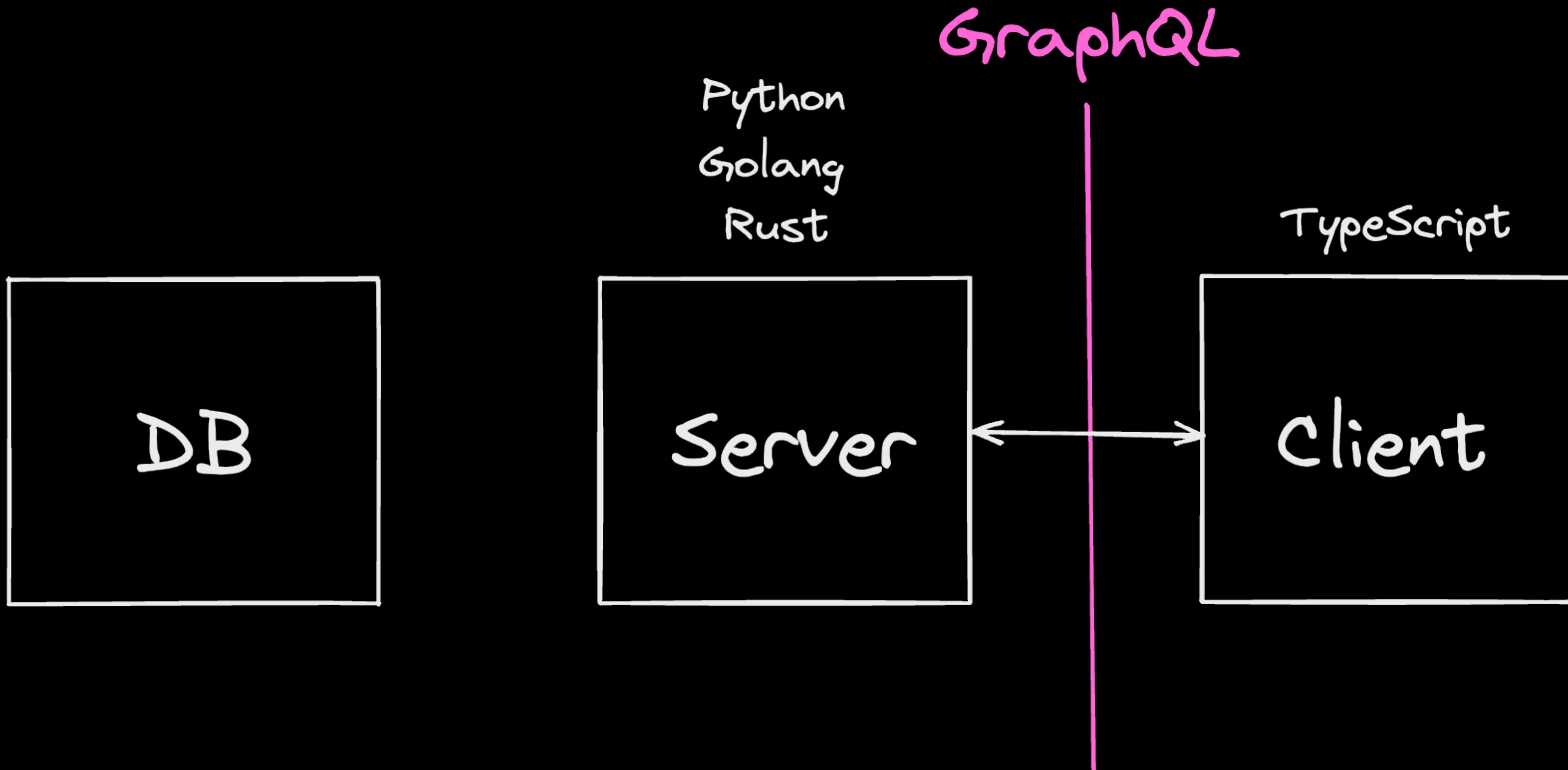
Component

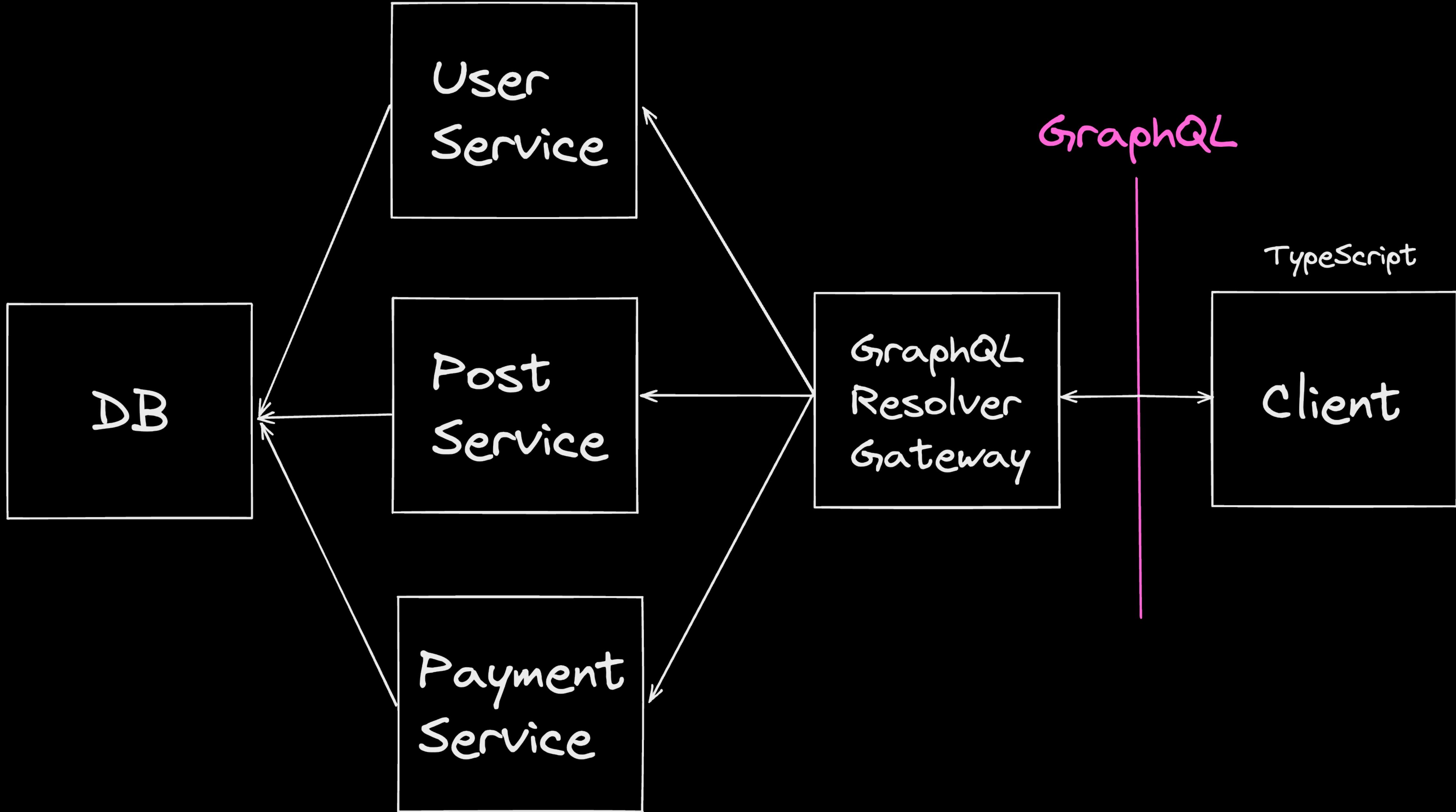


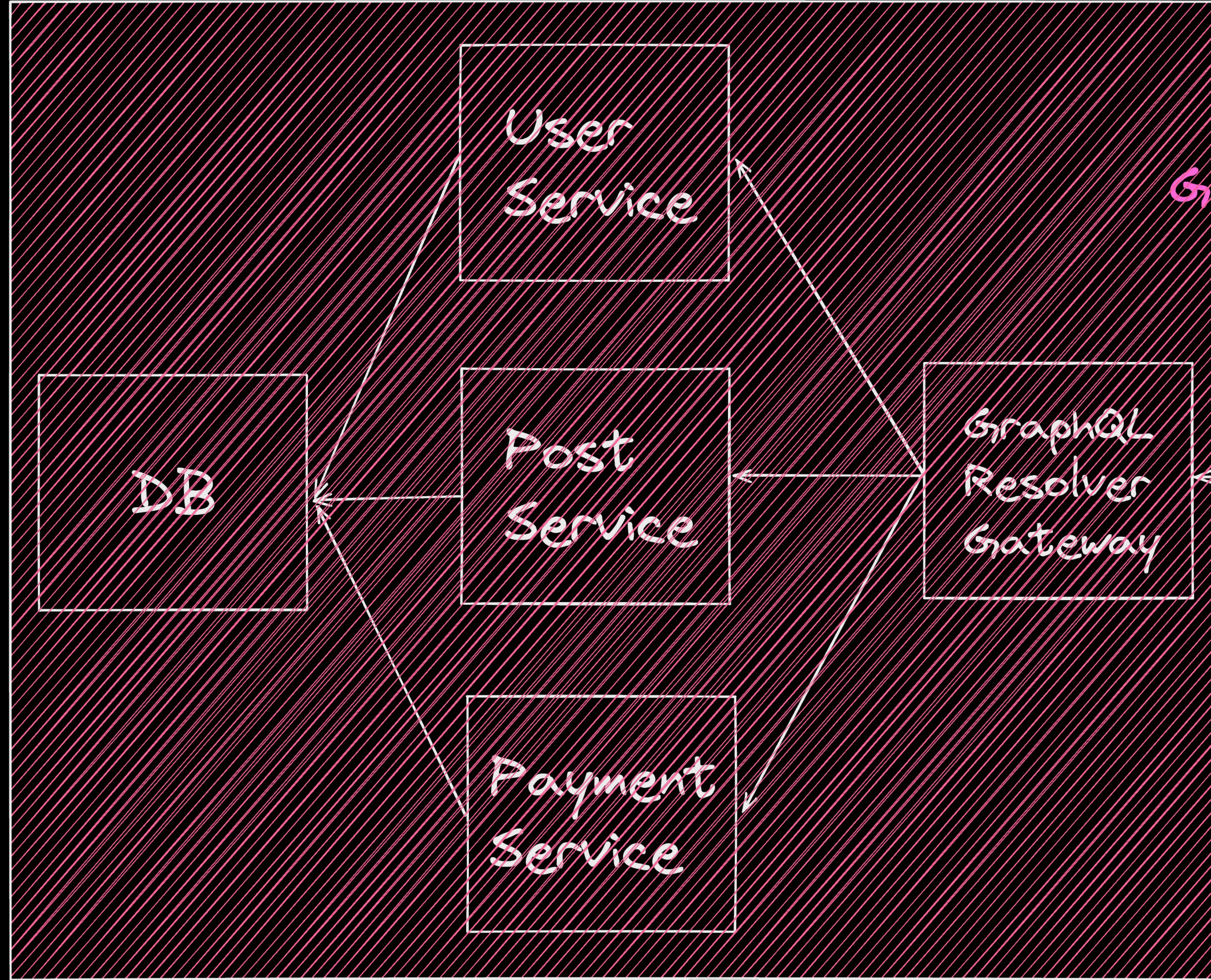
Fragment



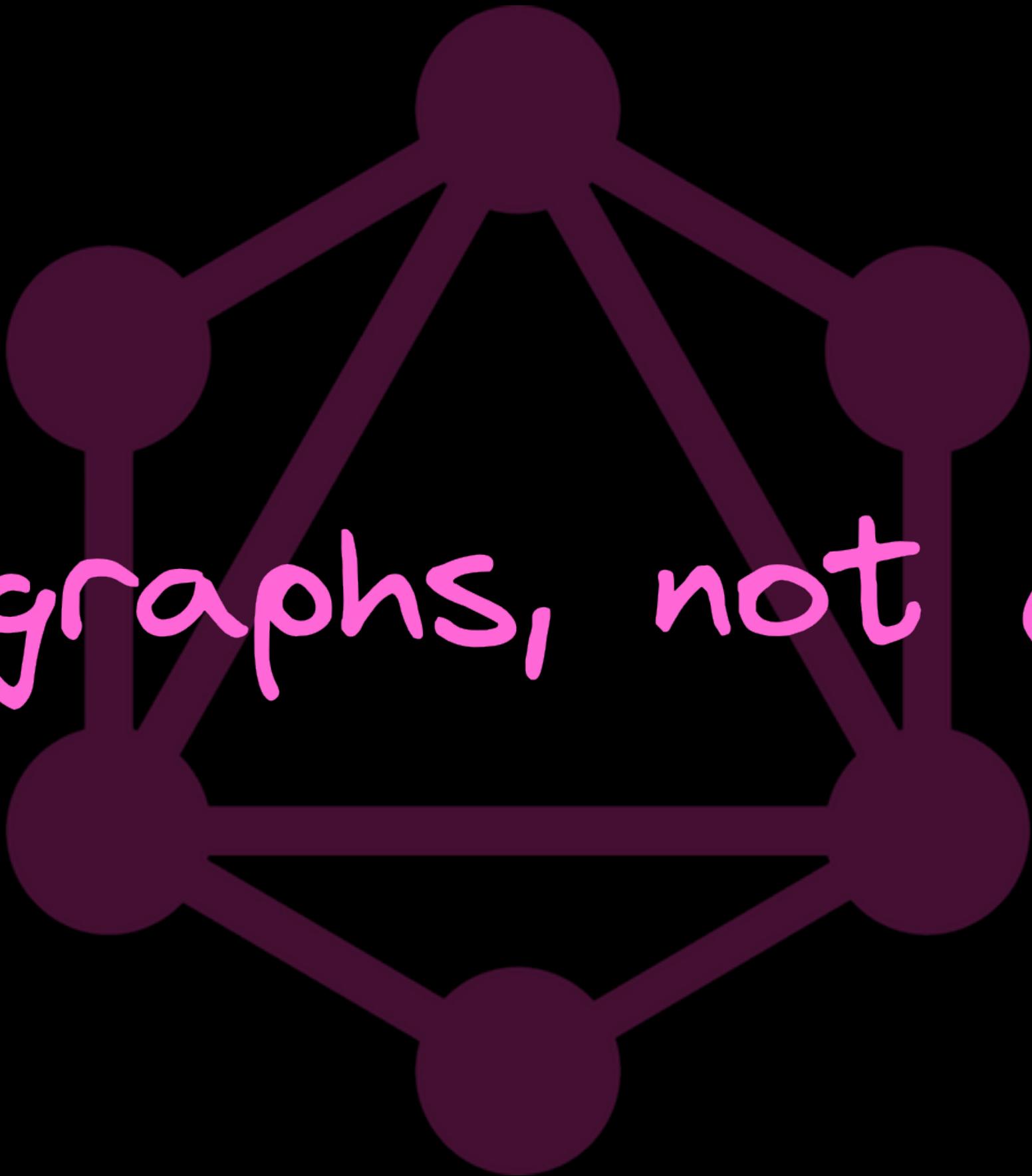






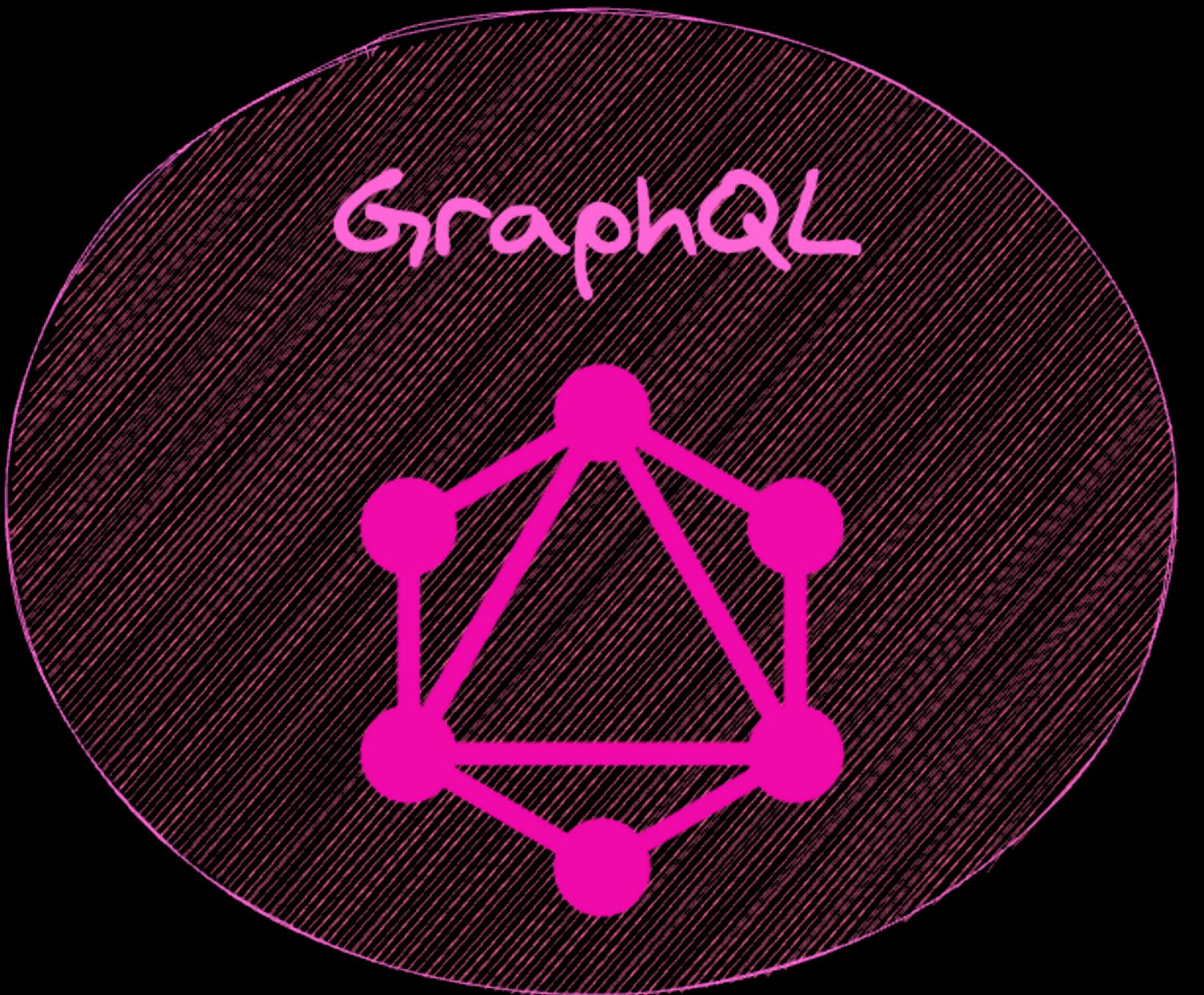


TypeScript



Think in graphs, not endpoints

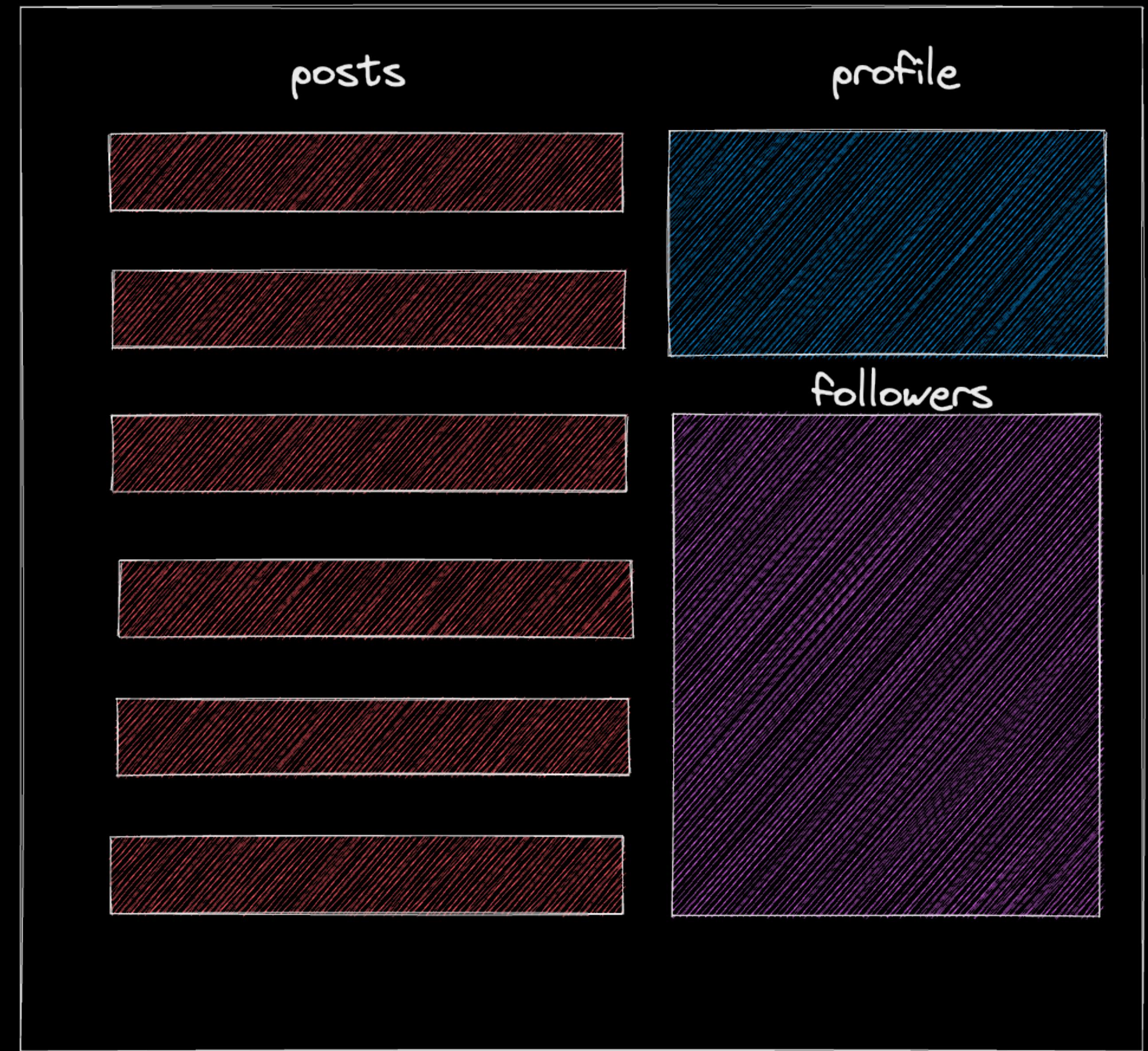
# GraphQL vs REST



VS

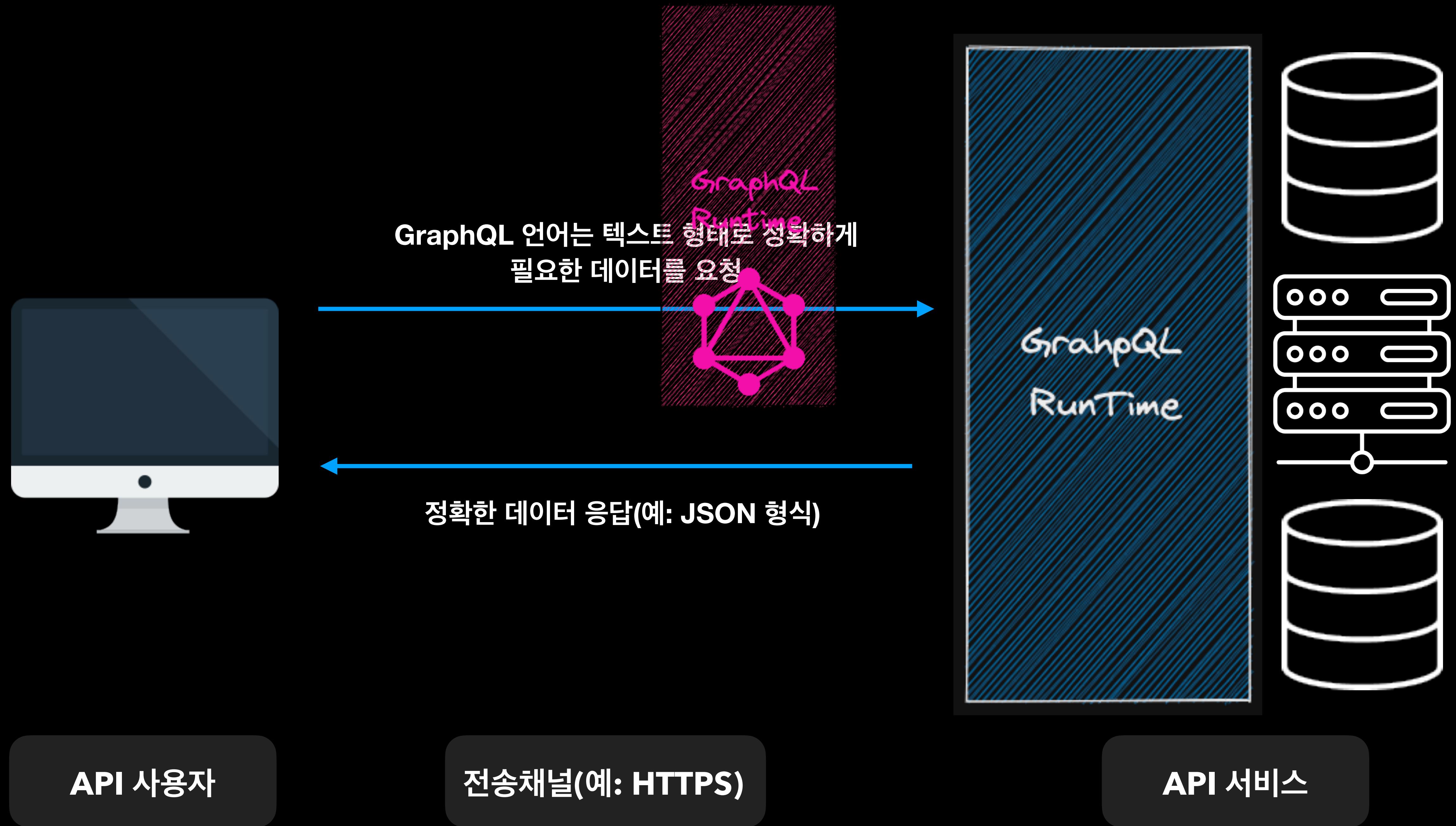


# Example



# Introduction to GraphQL

- Graph Query Language
- GraphQL is a Runtime

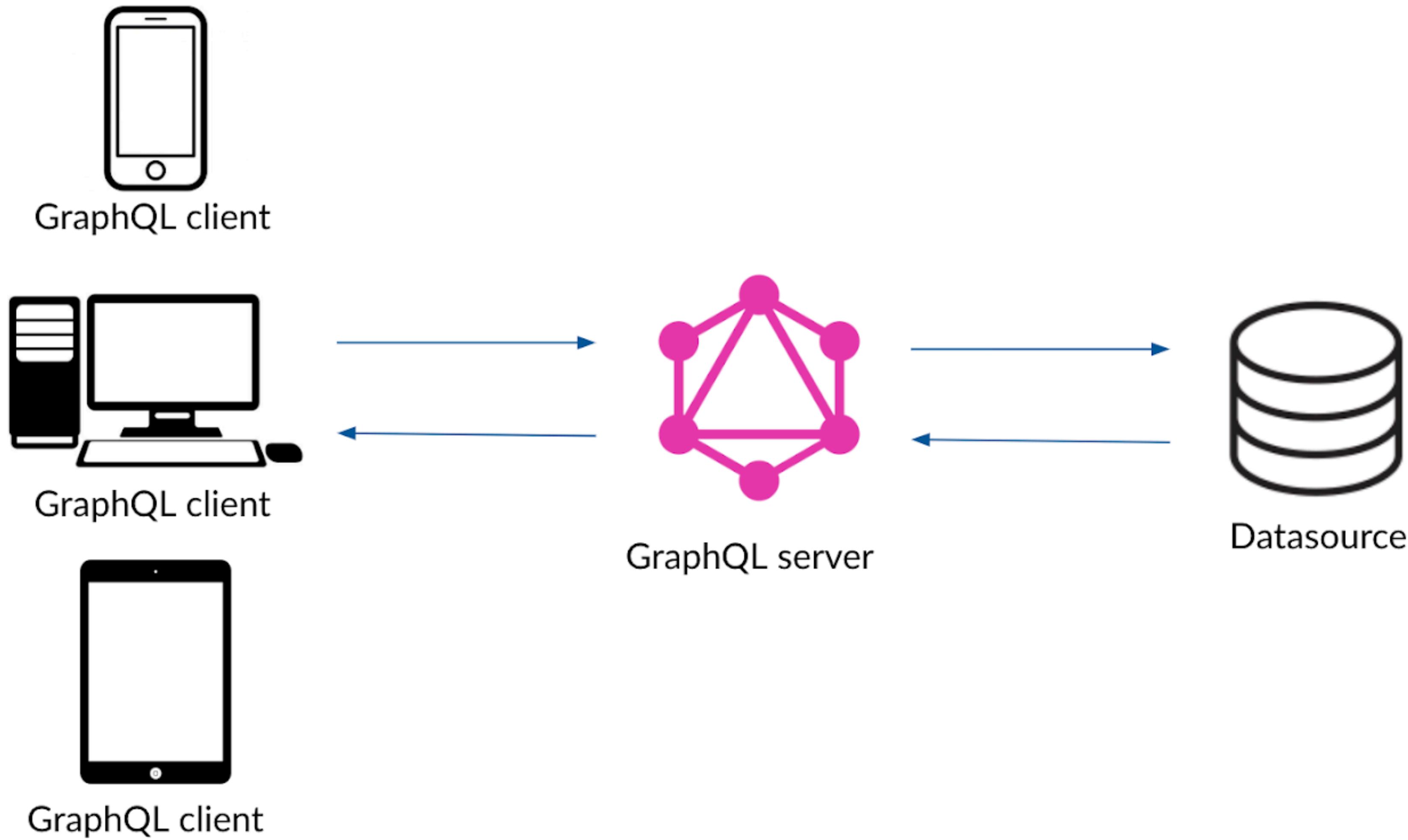


# Reference

- **GraphQL in Action.** Samer Buna
  - **How to GraphQL**
  - **Apollo Docs**
  - **Data Flows in React**
- 
- **GraphQL 기초 툴아보기.** devstone

REST API는 명령형

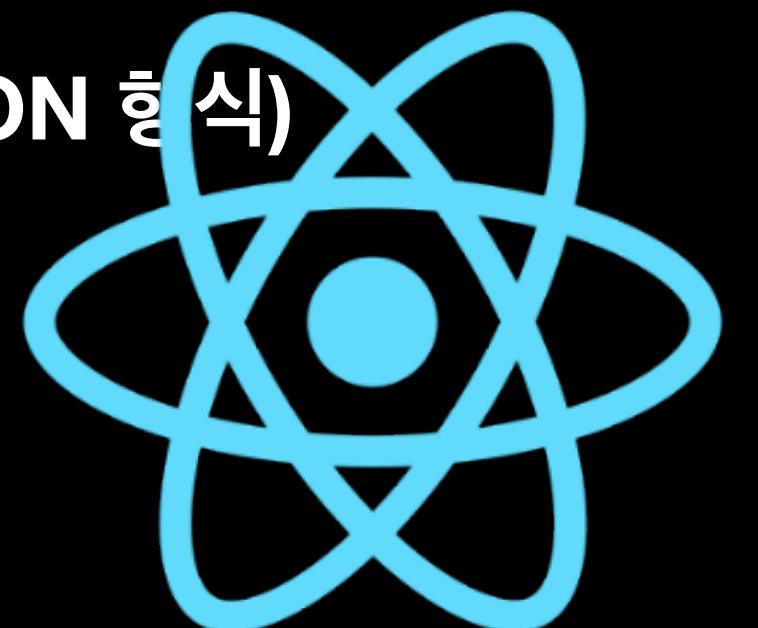
Graphql 선언형



**GraphQL** 언어는 텍스트 형태로 정확하게  
필요한 데이터를 요청



정확한 데이터 응답(예: JSON 형식)

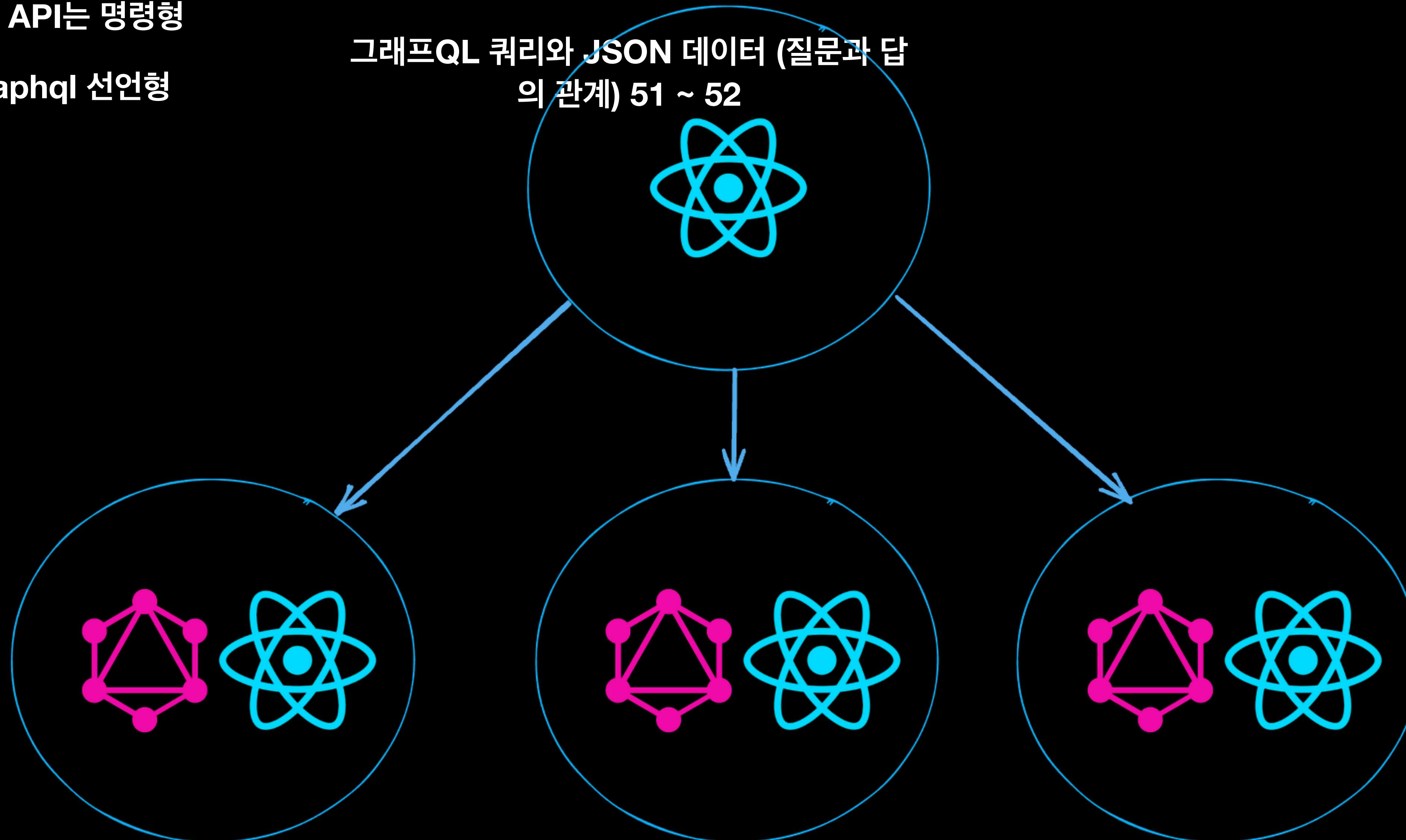


예제 들기 46 ~ 53

REST API는 명령형

Graphql 선언형

그래프QL 쿼리와 JSON 데이터 (질문과 답  
의 관계) 51 ~ 52

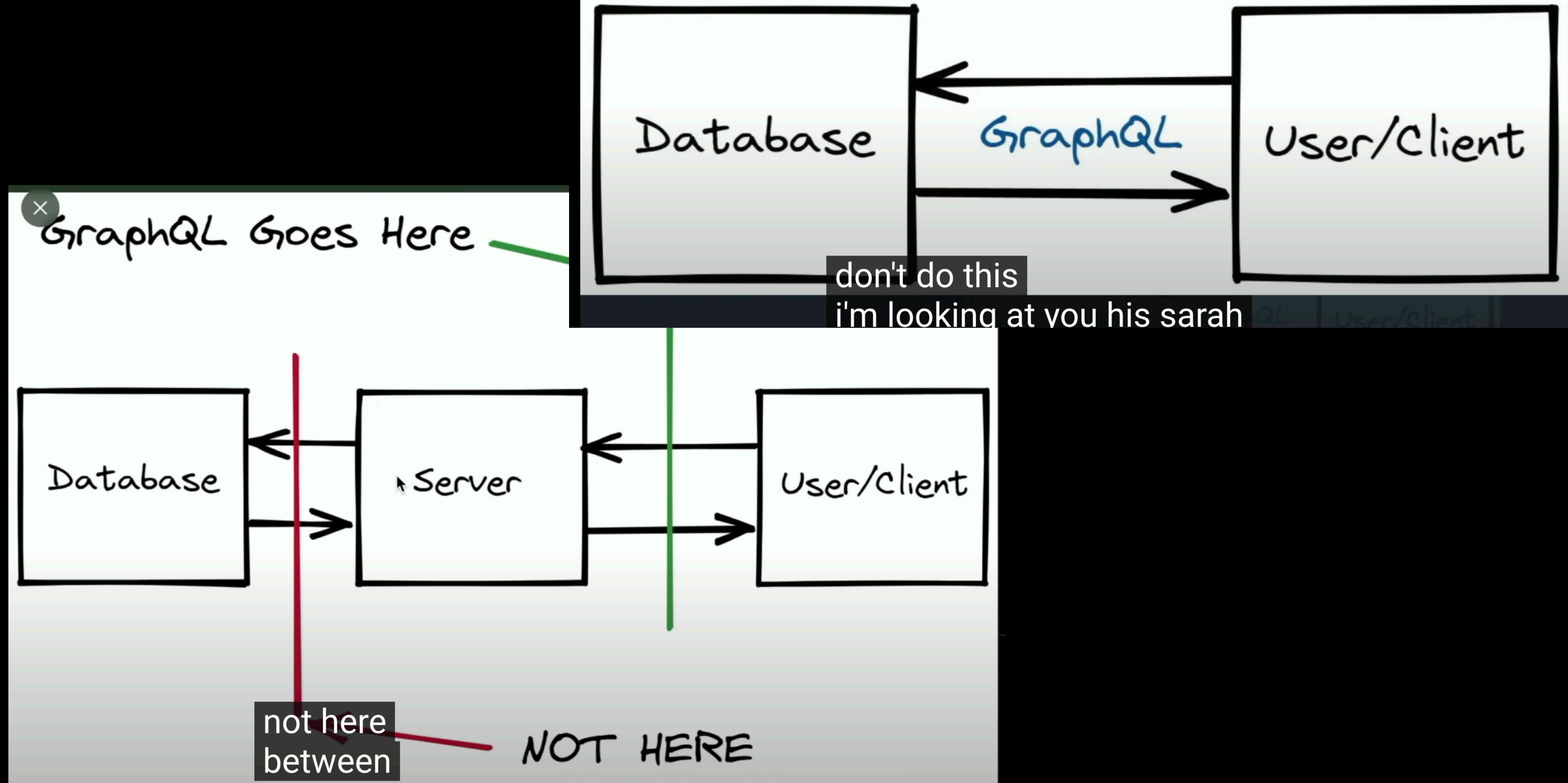


**useQuery**

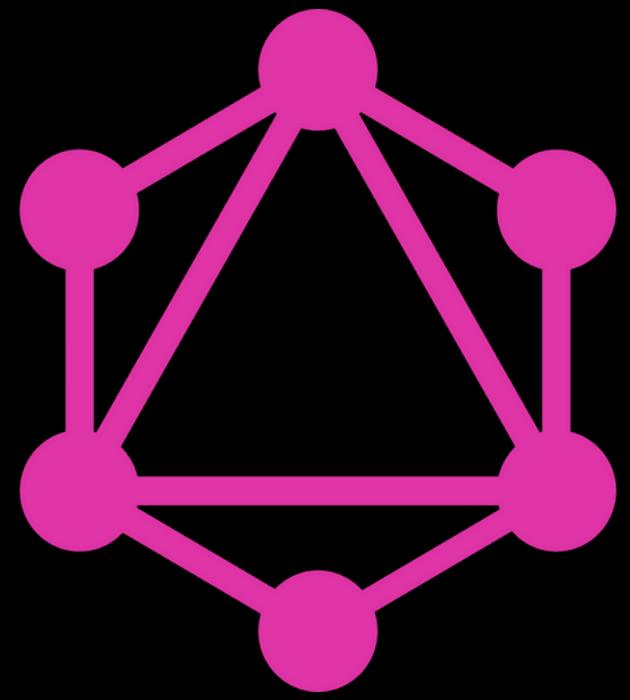


**useState**

This is just bad.







# Concepts

- Introduction to GraphQL
  - How to GraphQL
  - Apollo Docs
  - Data Flows in React
- 
- GraphQL 기초 투아보기. devstone

REST API는 명령형

Graphql 선언형