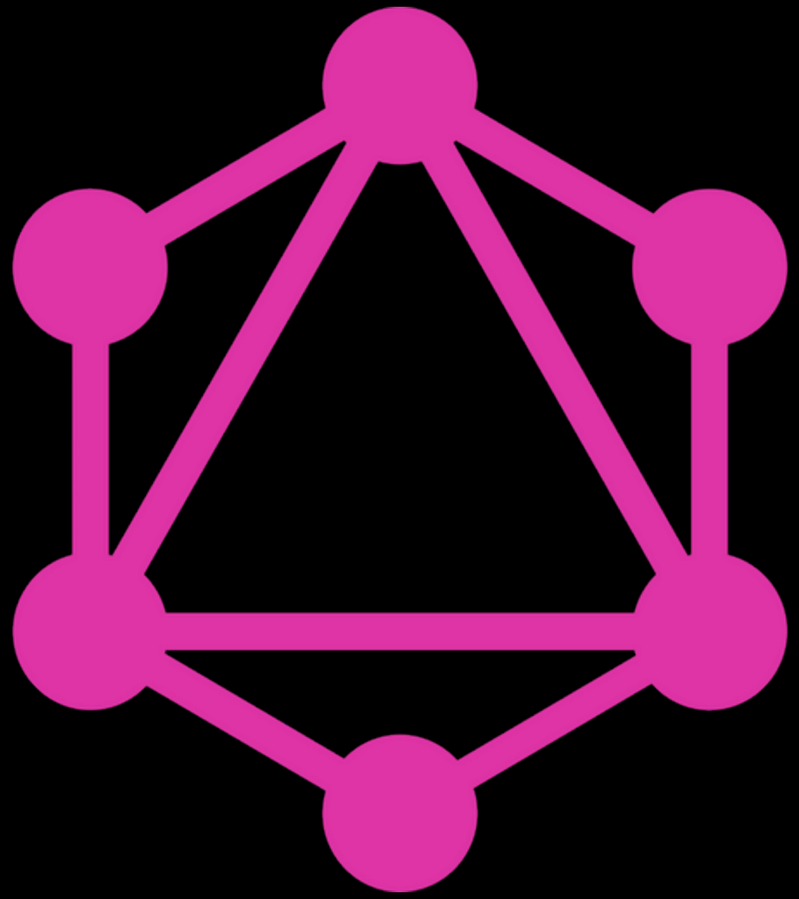
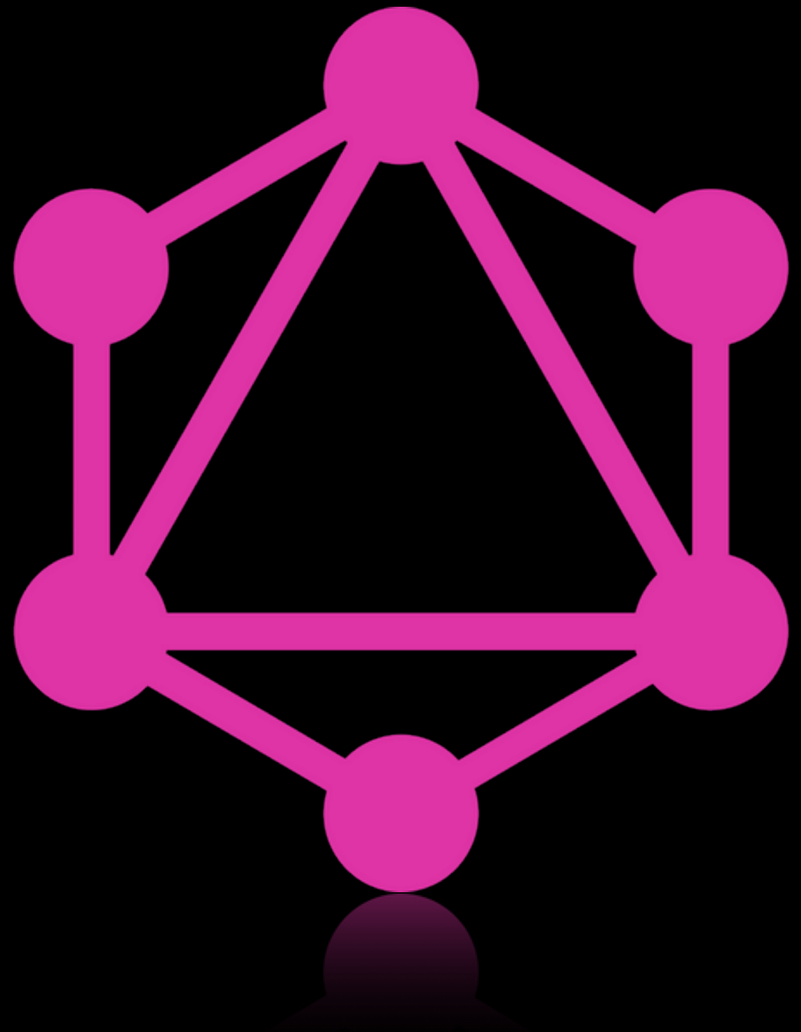


GraphQL

소프트웨어 마에스트로

SOFTWARE MAESTRO

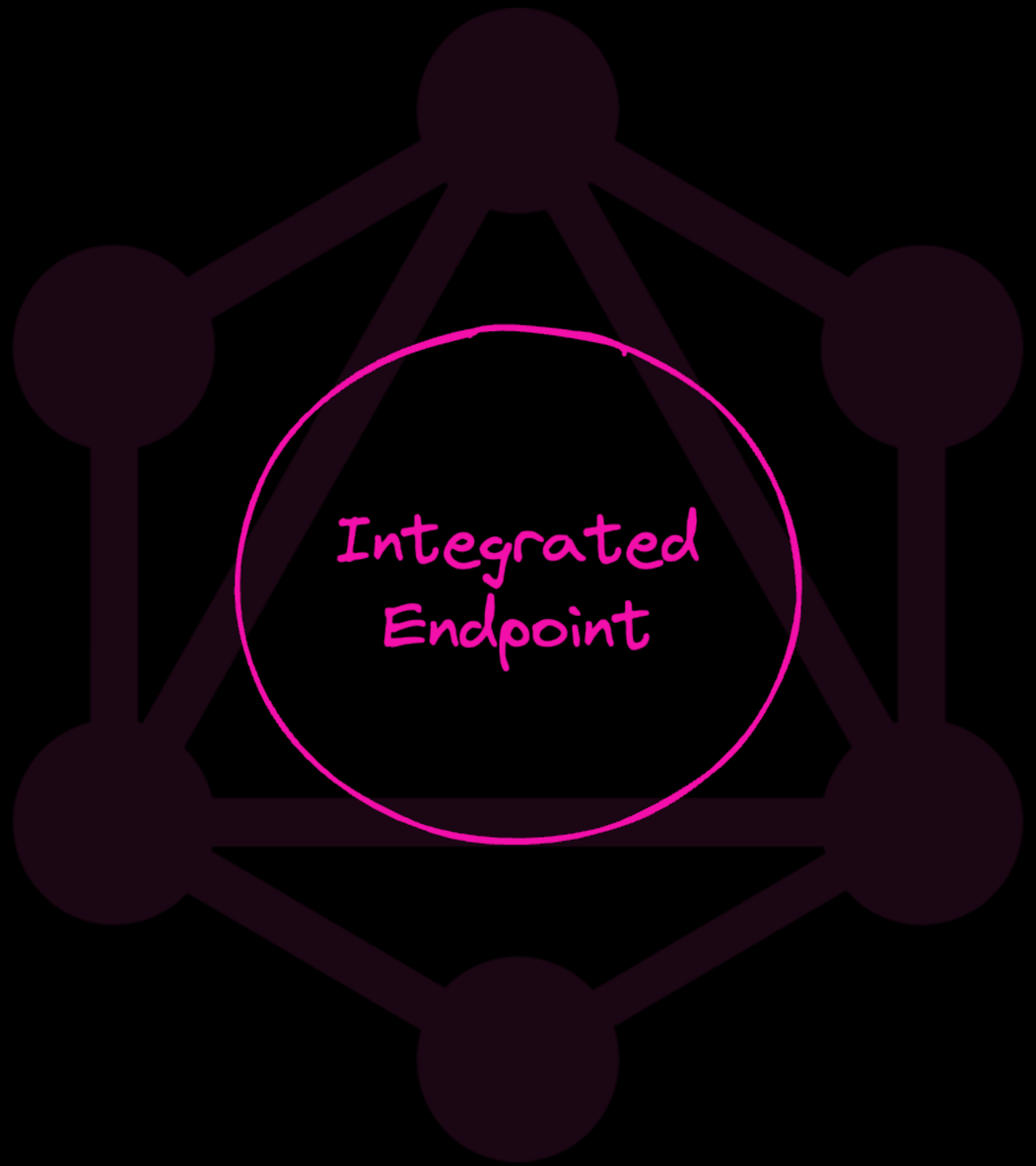
hyunjinee



Problem


Increased mobile usage
Variety of Platforms
Fast Development

GraphQL



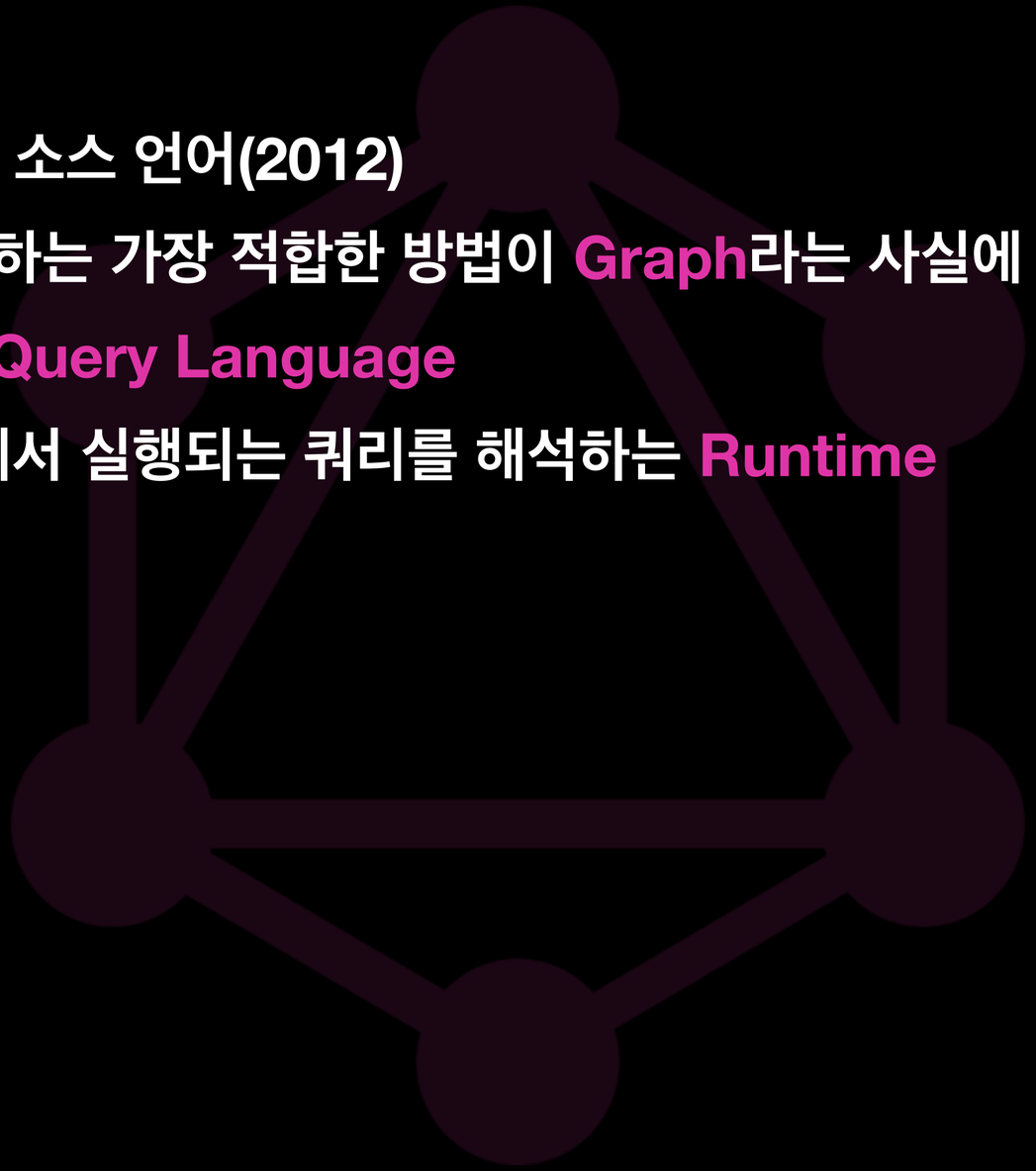
GET ALL
IN ONCE

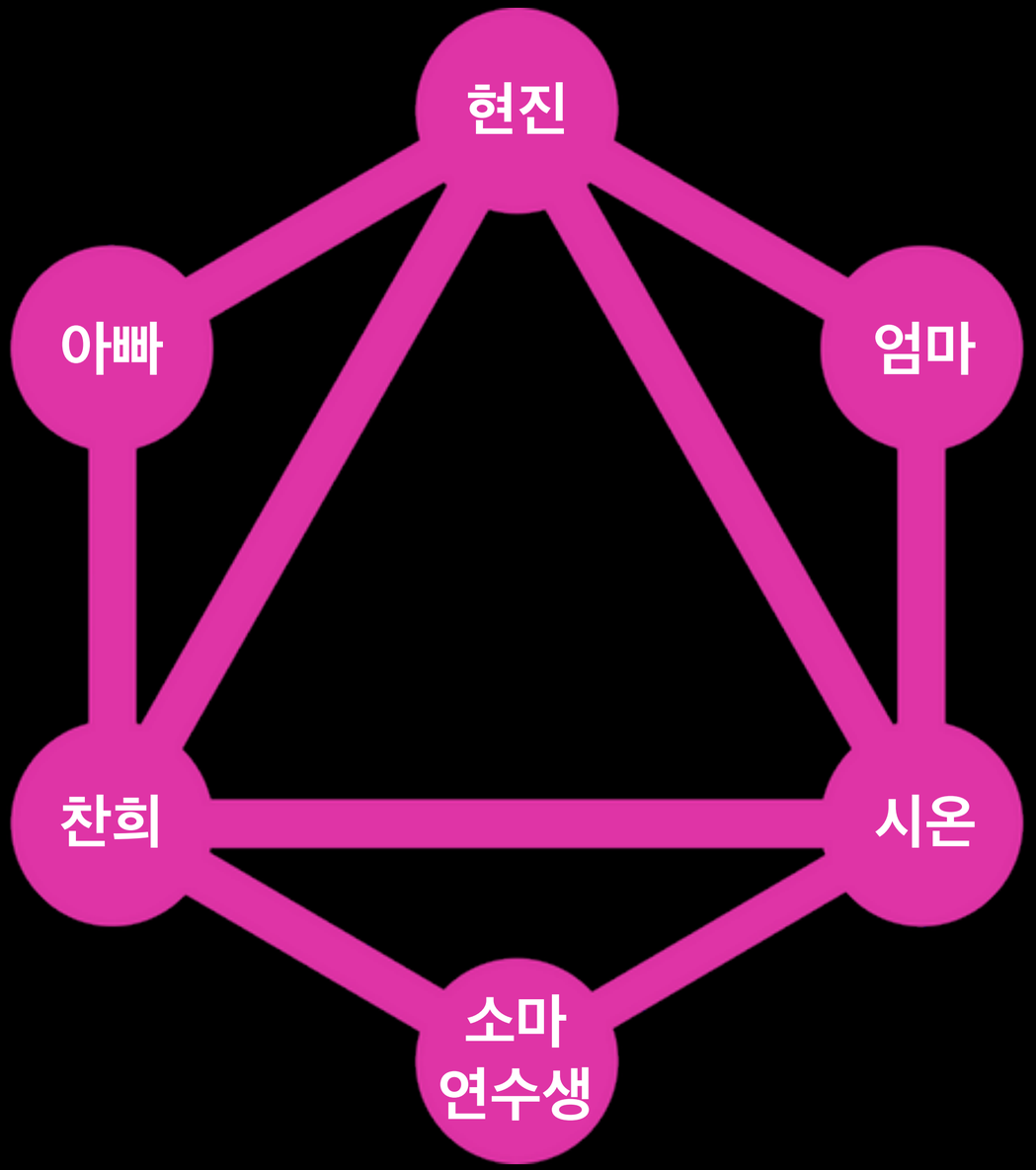
Integrated
Endpoint

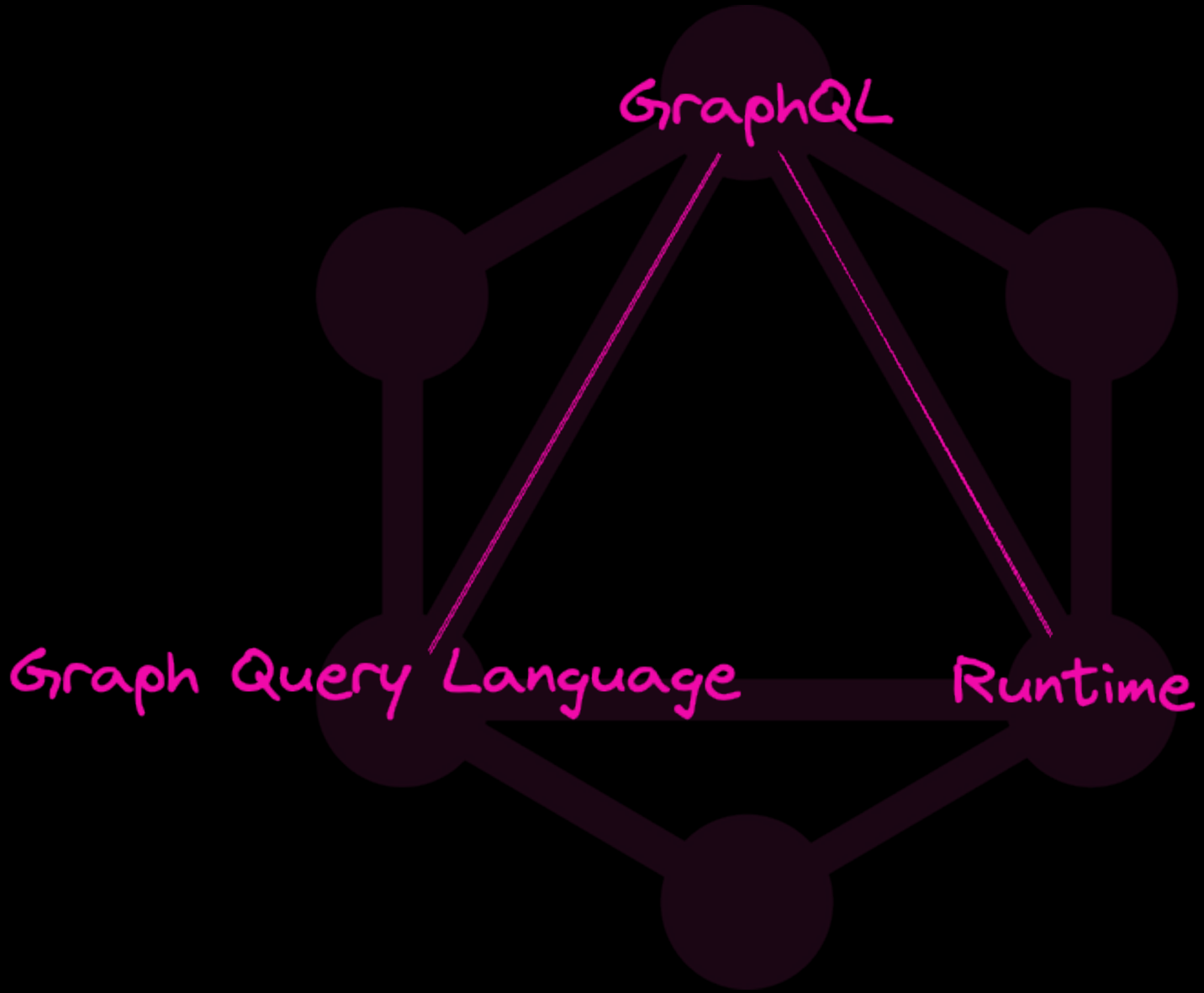
Less
Comunication

GraphQL

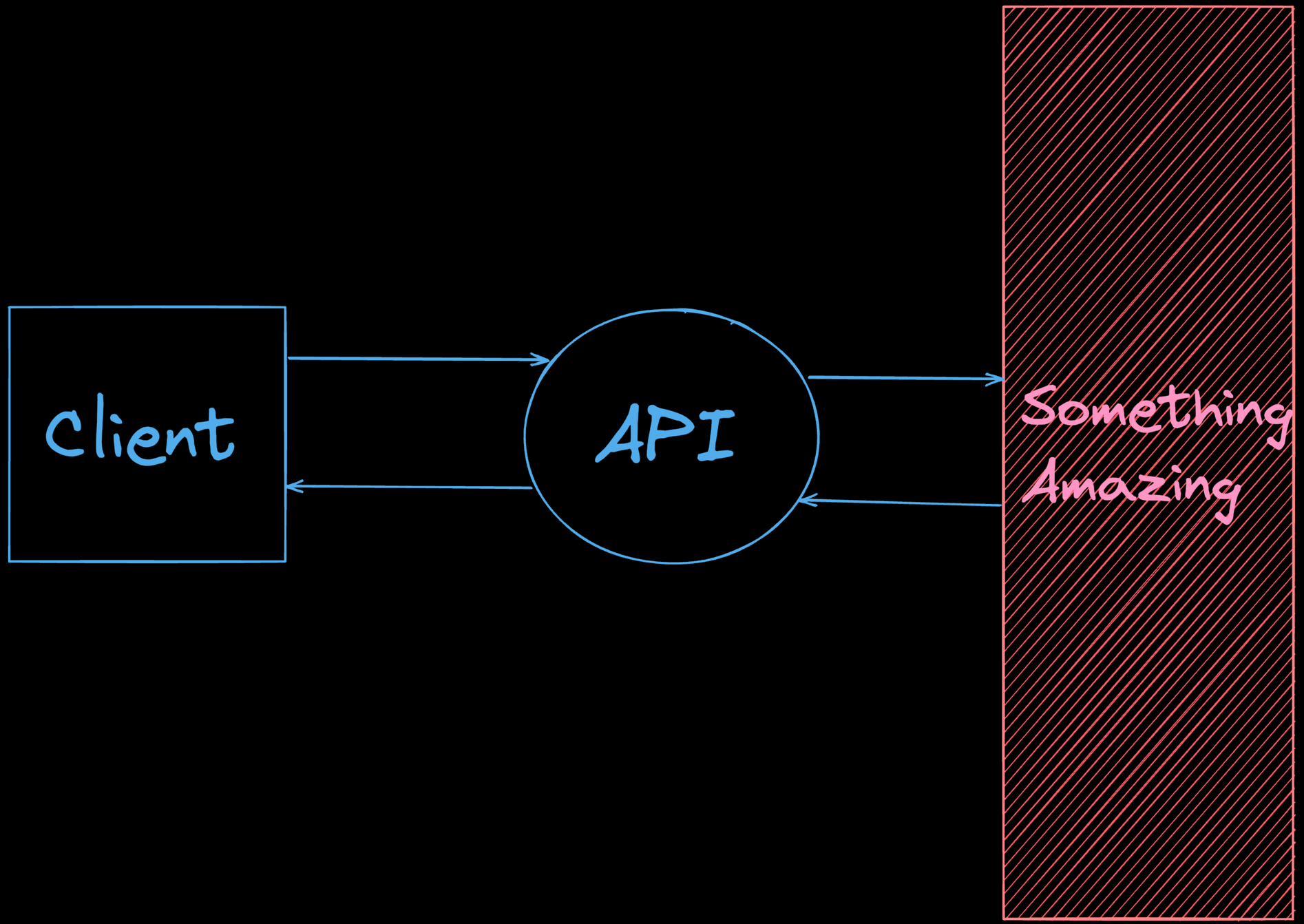
- Facebook이 개발한 오픈 소스 언어(2012)
- 현실 세계의 데이터를 표현하는 가장 적합한 방법이 Graph라는 사실에 착안
- GraphQL은 API를 위한 Query Language
- GraphQL은 서버사이드에서 실행되는 쿼리를 해석하는 Runtime




현진
아빠
엄마
찬희
시온
소마
연수생


GraphQL
Graph Query Language
Runtime

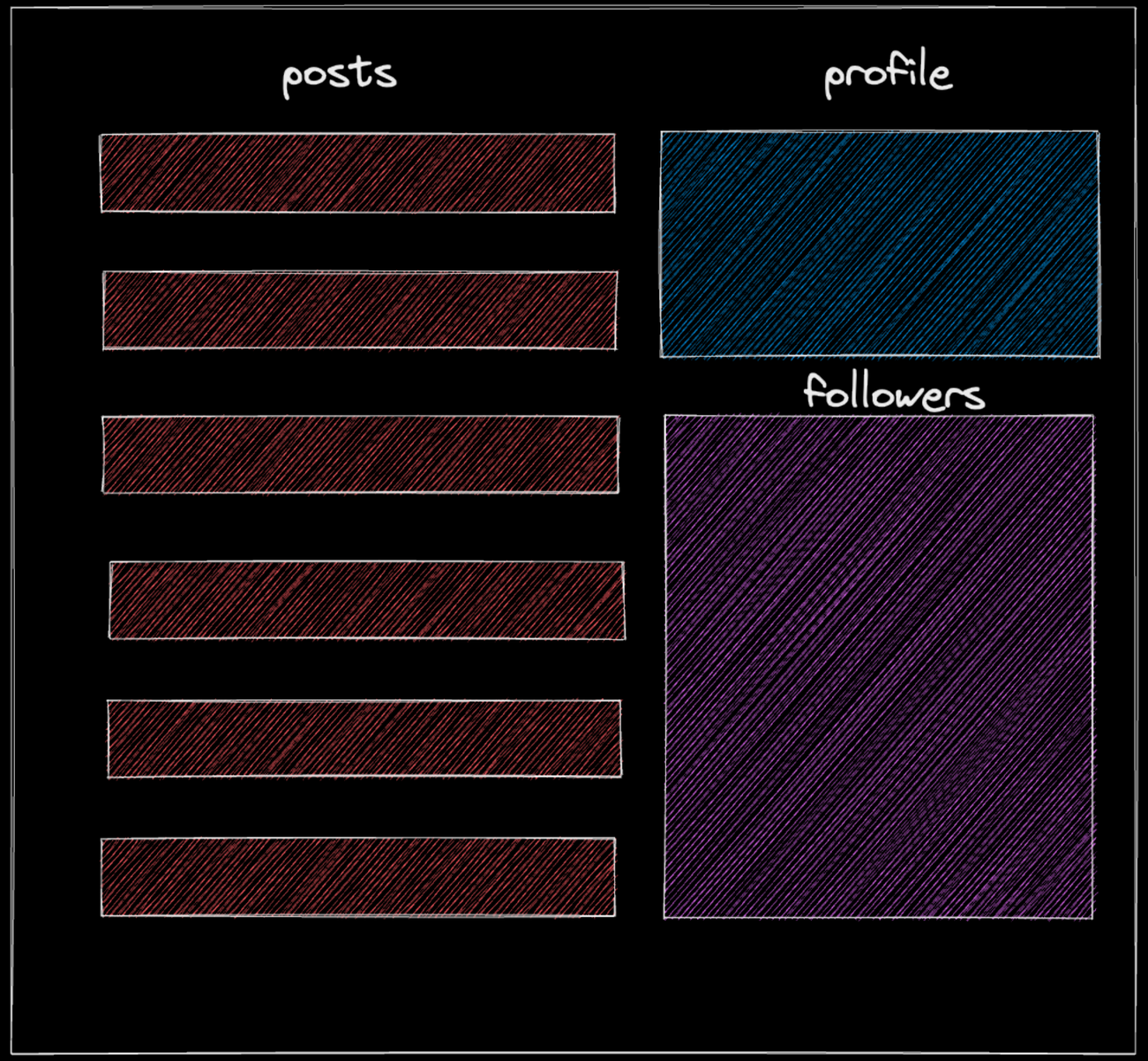
API



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

GET
POST
PUT
DELETE

Example



REST

HTTP GET → /user/<id>

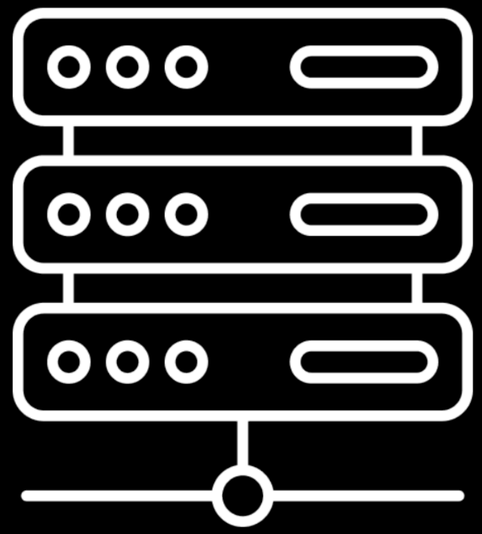
```
{
    "user": {
        "id" : "hyunjin",
        "name": Lee",
        "address": {...},
    }
}
```

HTTP GET → /user/<id>/posts

```
{
    "posts": [{
        "title": "graphQL",
        "content": "Lorem ipsum",
        "comments": [...],
    }]
}
```

HTTP GET → /user/<id>/followers

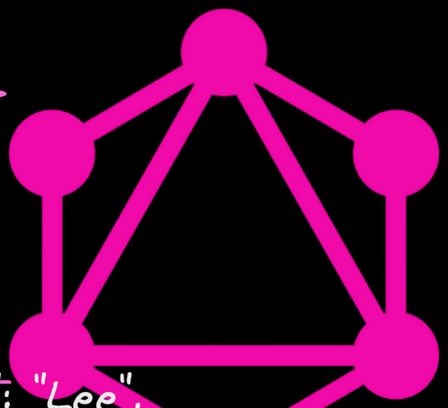
```
{
    "followers": [{
        "id": "Lionel",
        "name": "Messi",
    }]
}
```

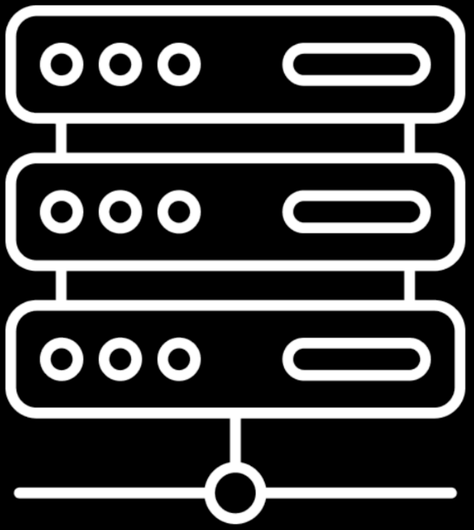


GraphQL

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

HTTP POST





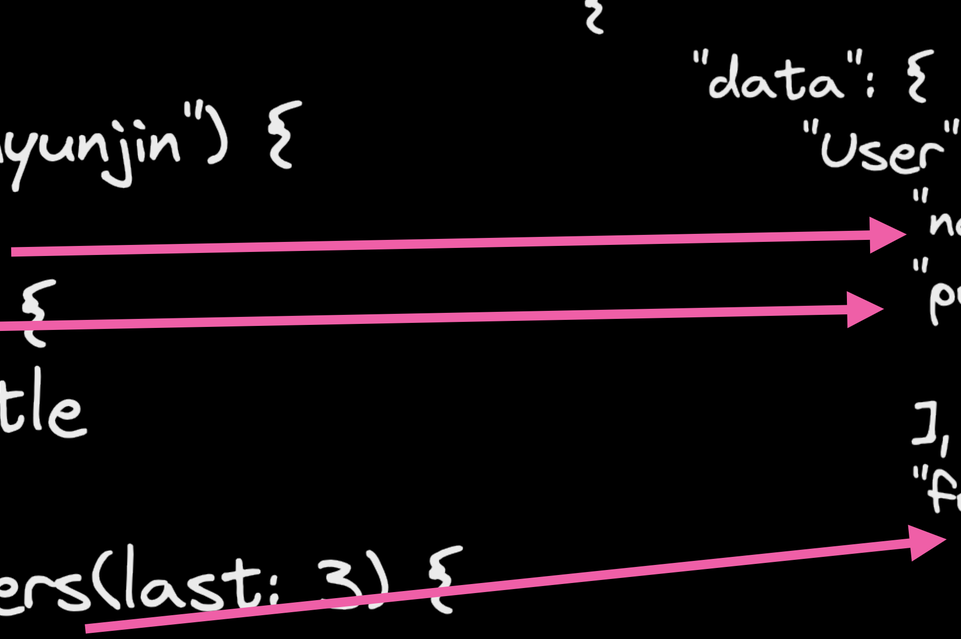
```
{
    "data": {
        "User": {
            "name": "Lee",
            "posts": [
                { title: "GraphQL"},
            ],
            "followers": [
                { name: "KIM"},
                { name: "PARK"},
                { name: "John"}
            ]
        }
    }
}
```

REQUEST

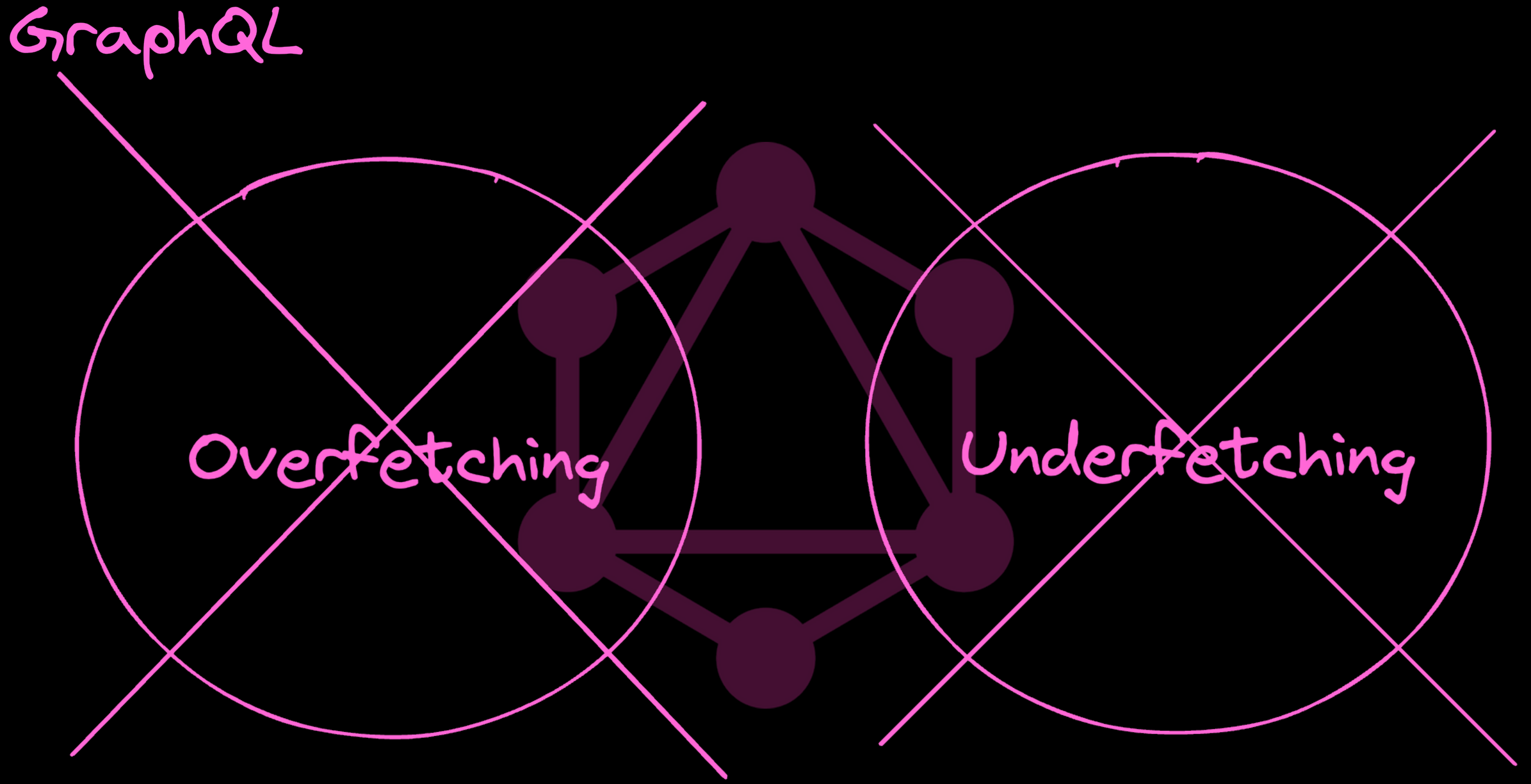
```
query {
    User(id: "hyunjin") {
        name
        posts {
            title
        }
        followers(last: 3) {
            name
        }
    }
}
```

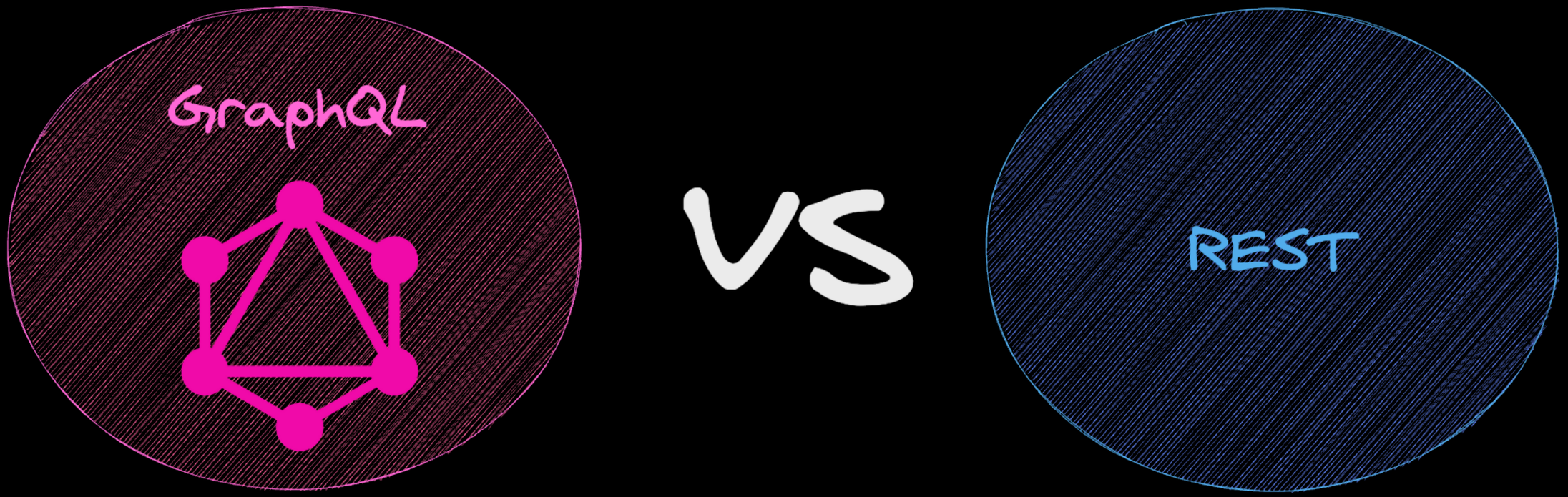
RESPONSE

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

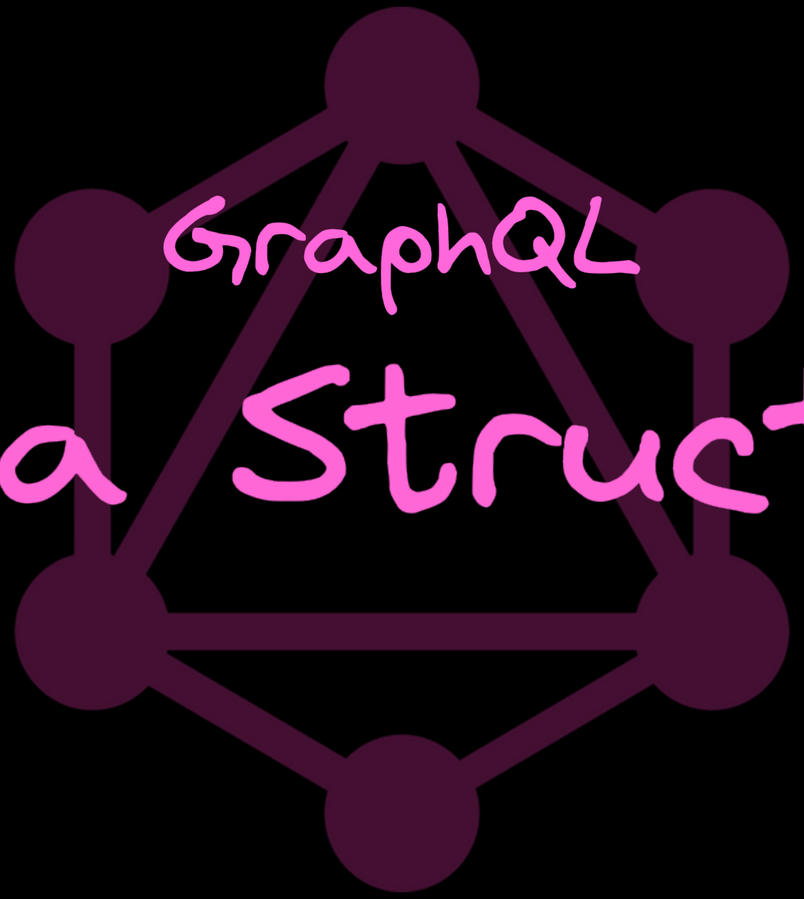


REST


GraphQL
Overfetching
Underfetching


GraphQL
VS
REST

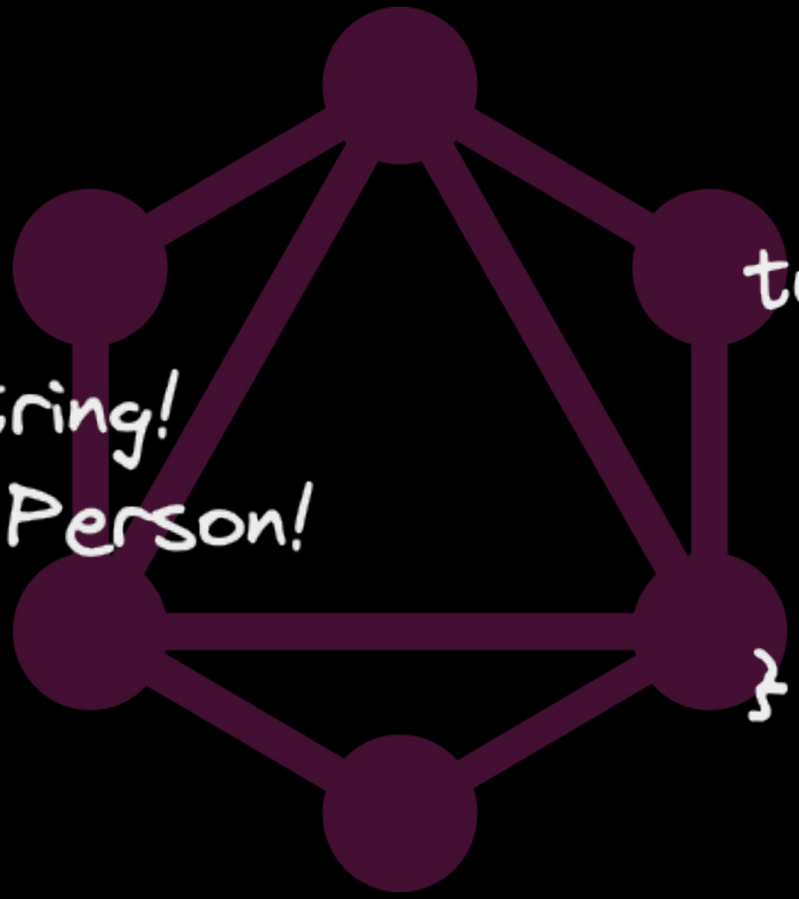
Communication



GraphQL

Data Structures

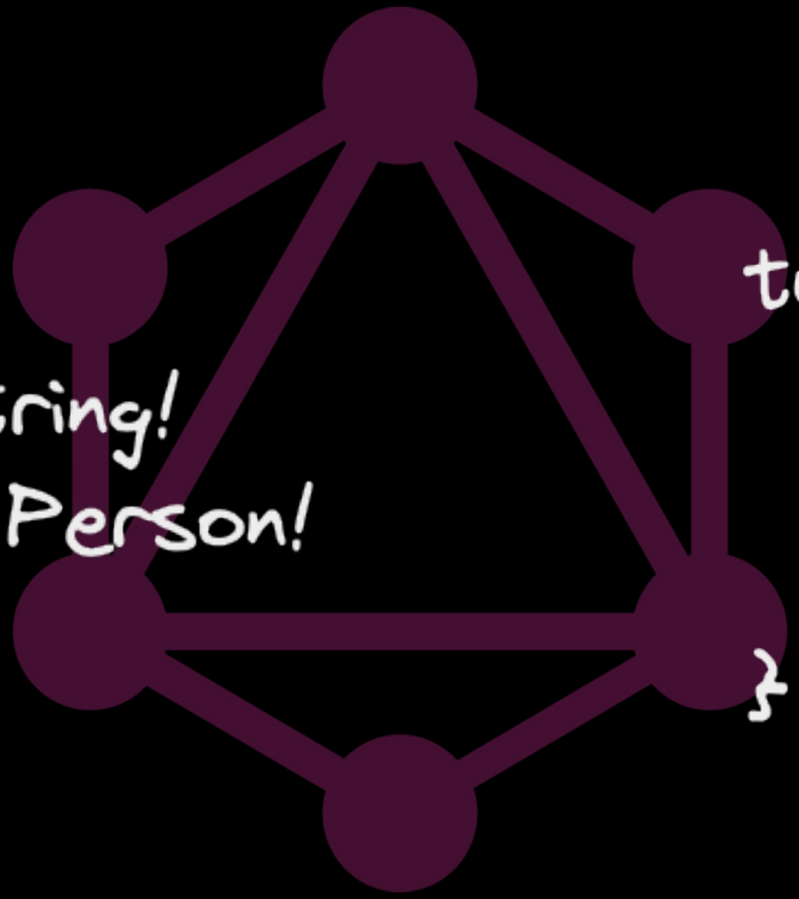
GraphQL Schema



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

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

GraphQL Schema



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

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

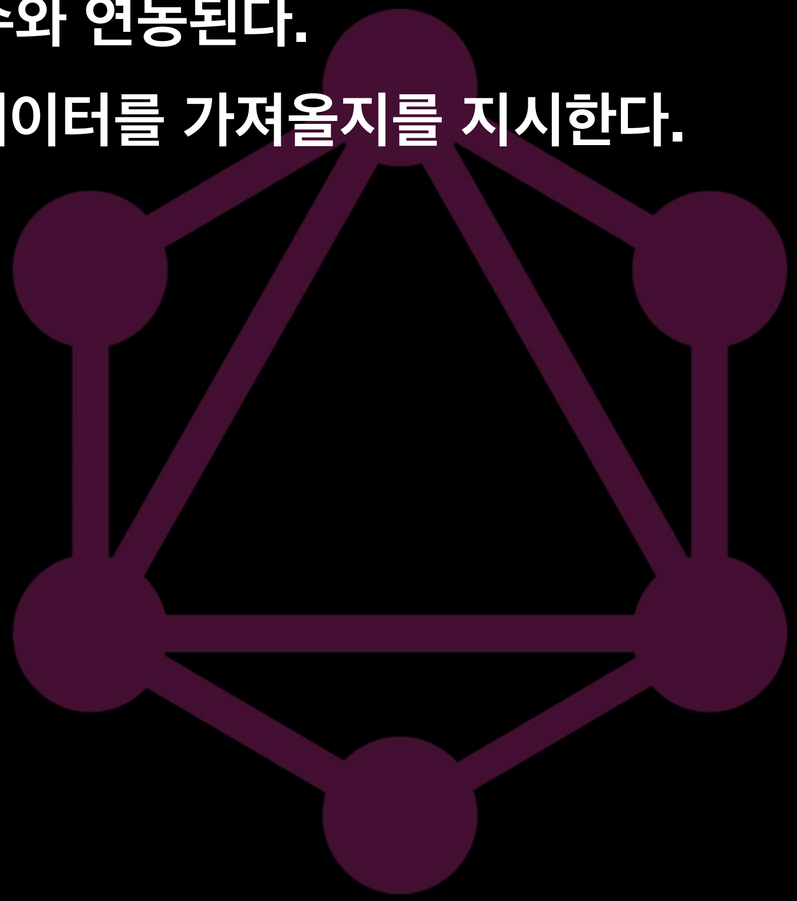
Schema

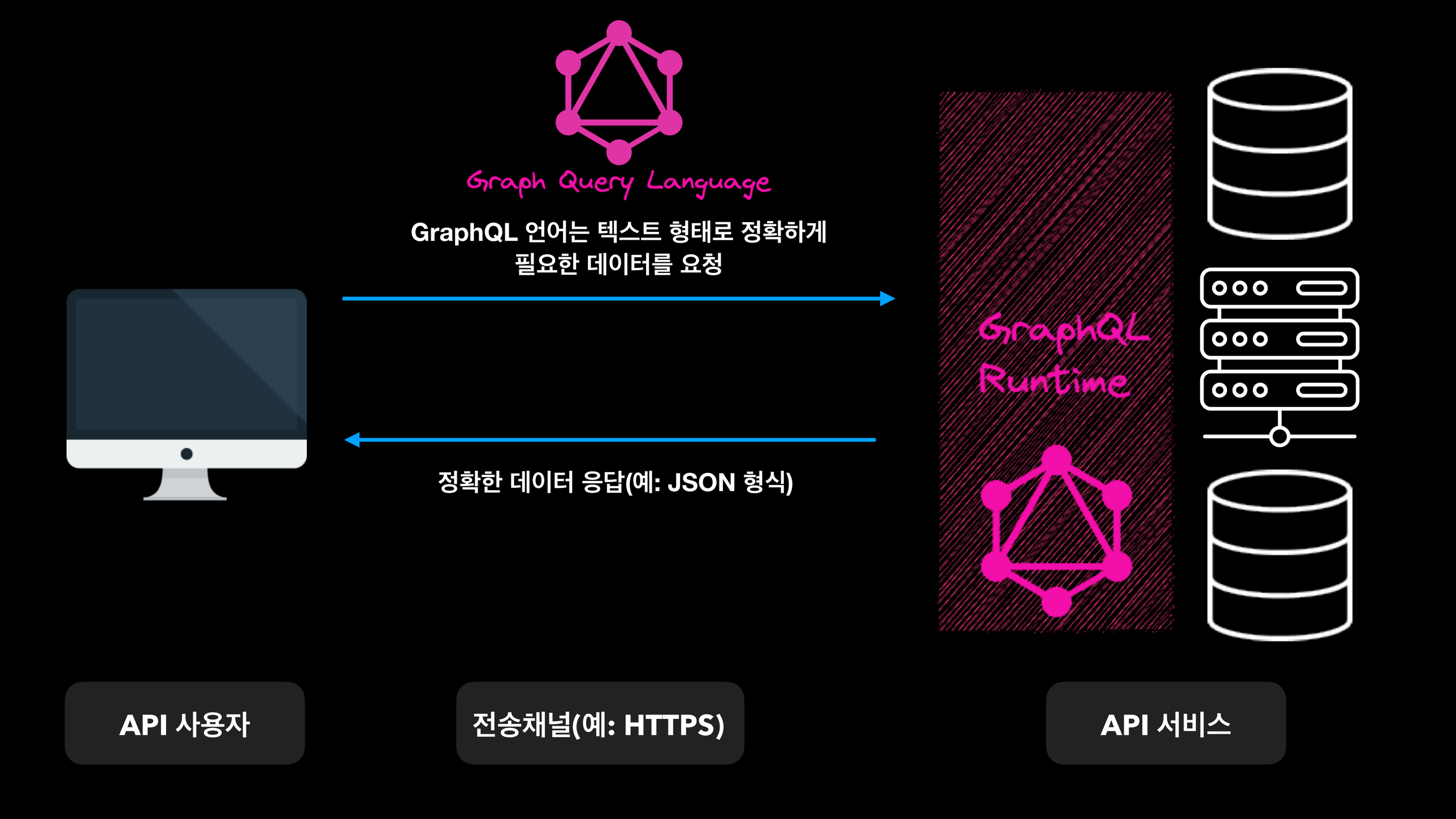
Frontend

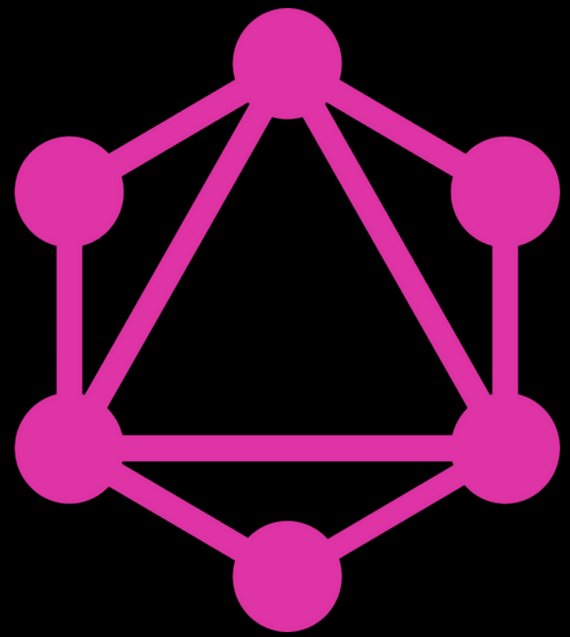
Backend

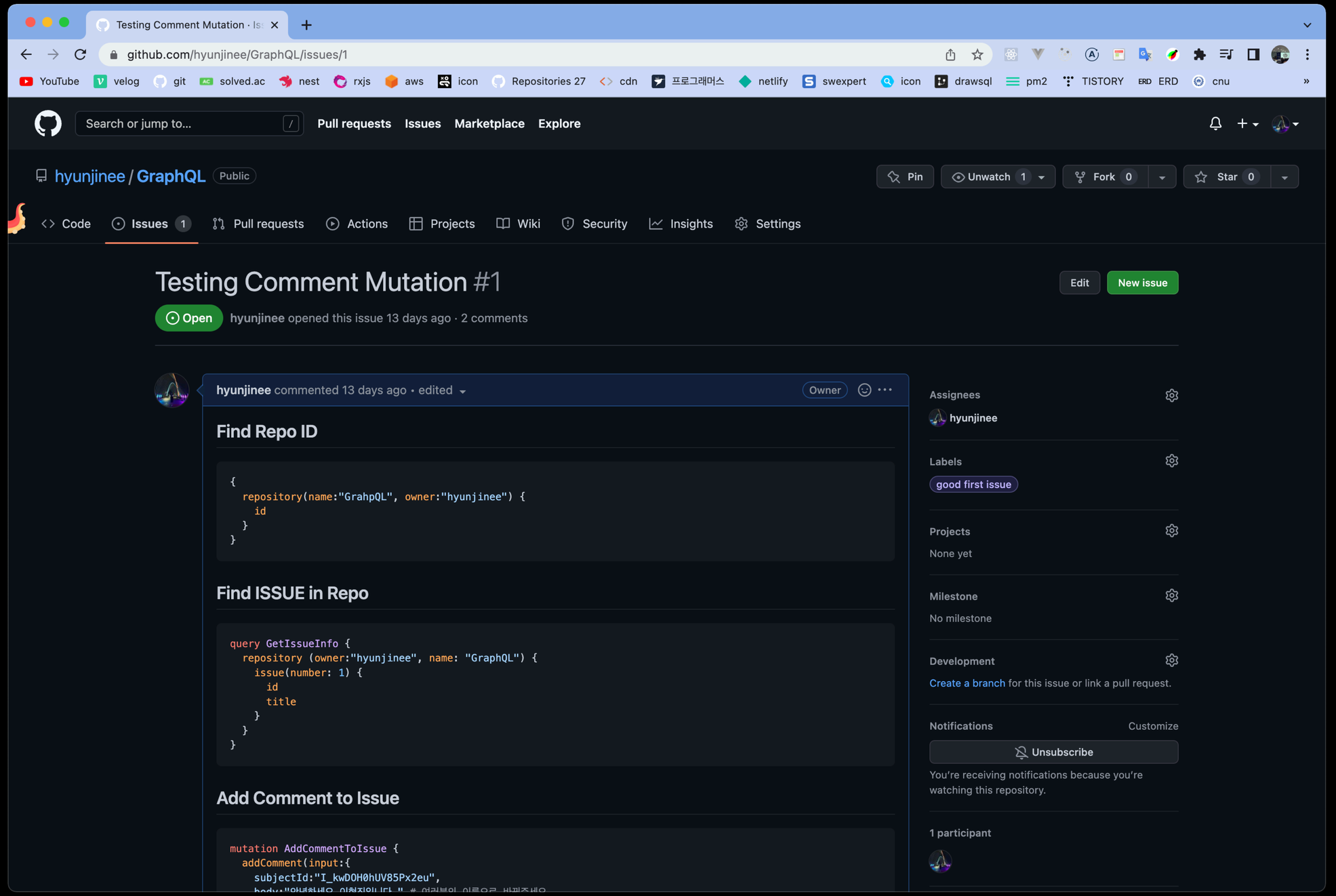
GraphQL Resolvers

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




Graph Query Language
GraphQL 언어는 텍스트 형태로 정확하게
필요한 데이터를 요청
정확한 데이터 응답(예: JSON 형식)
GraphQL
Runtime
API 사용자
전송채널(예: HTTPS)
API 서비스

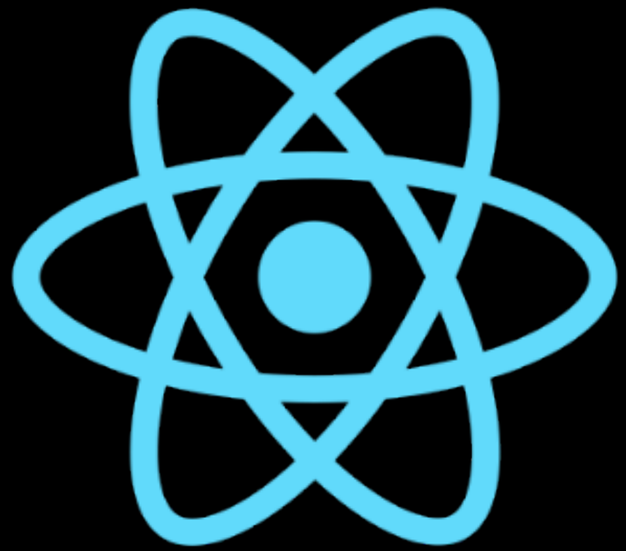


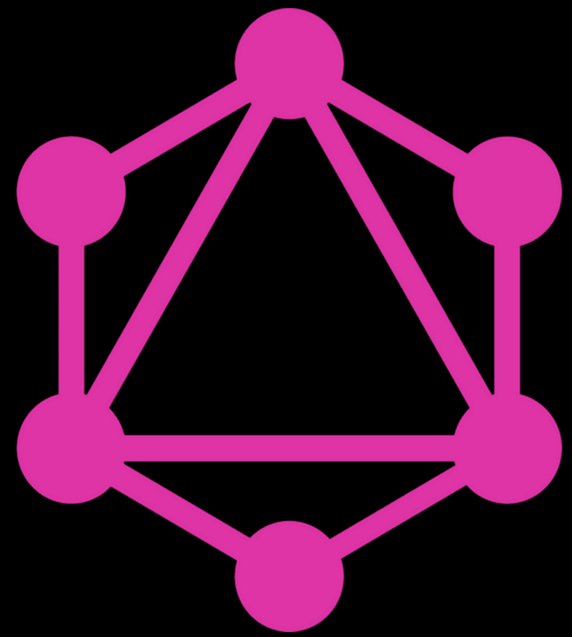

Testing Comment Mutation · Is
github.com/hyunjinee/GraphQL/issues/1
YouTube velog git solved.ac nest rxjs aws icon Repositories 27 cdn 프로그래머스 netlify swexpert icon drawsql pm2 TISTORY ERD ERD cnu
Search or jump to...
Pull requests Issues Marketplace Explore
hyunjinee / GraphQL Public
Pin Unwatch 1 Fork 0 Star 0
Code Issues 1 Pull requests Actions Projects Wiki Security Insights Settings
Testing Comment Mutation #1
Edit New issue
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_kwDOH0hUV85Px2eu",
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

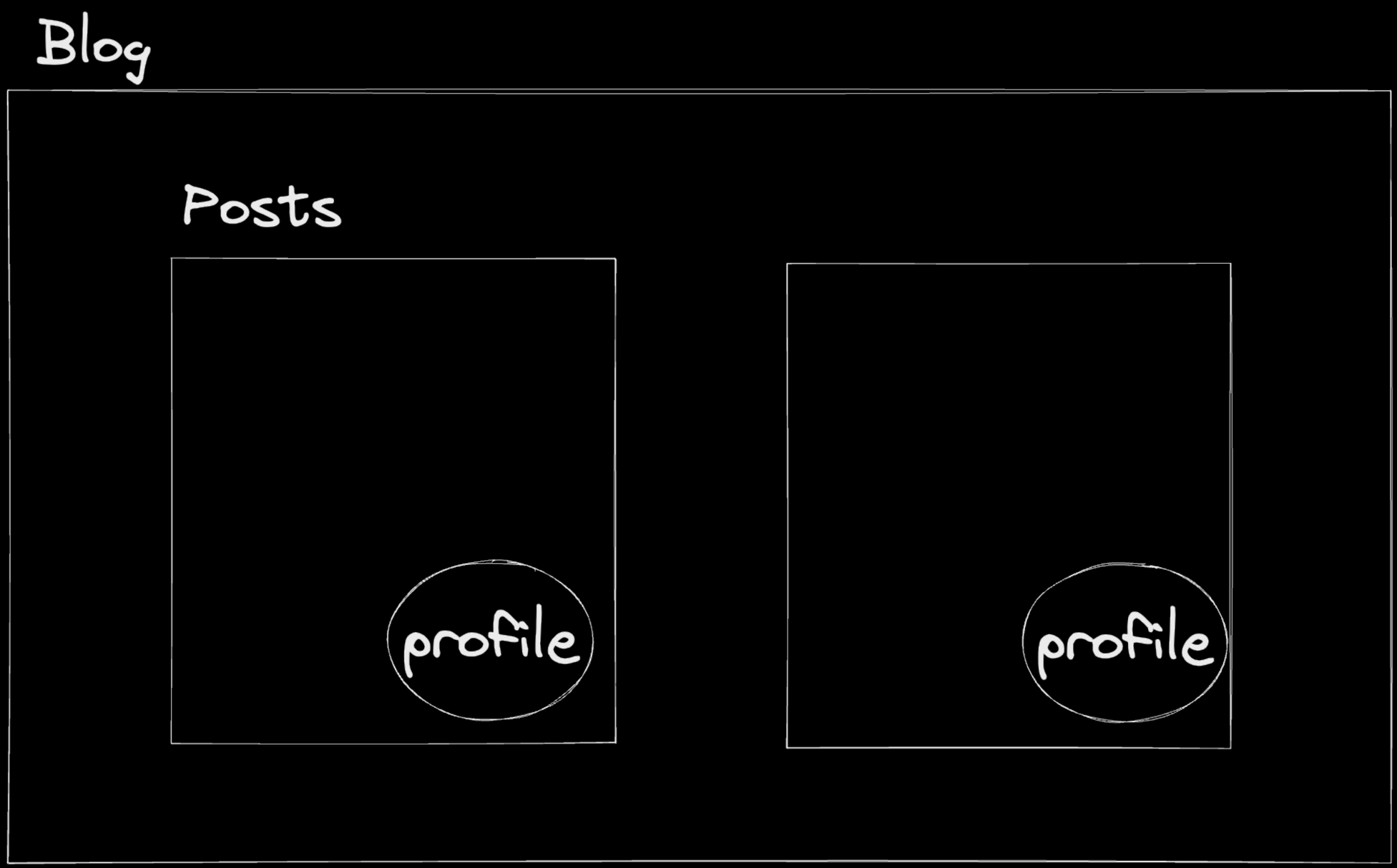
https://docs.github.com/en/graphql/overview/explorer

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

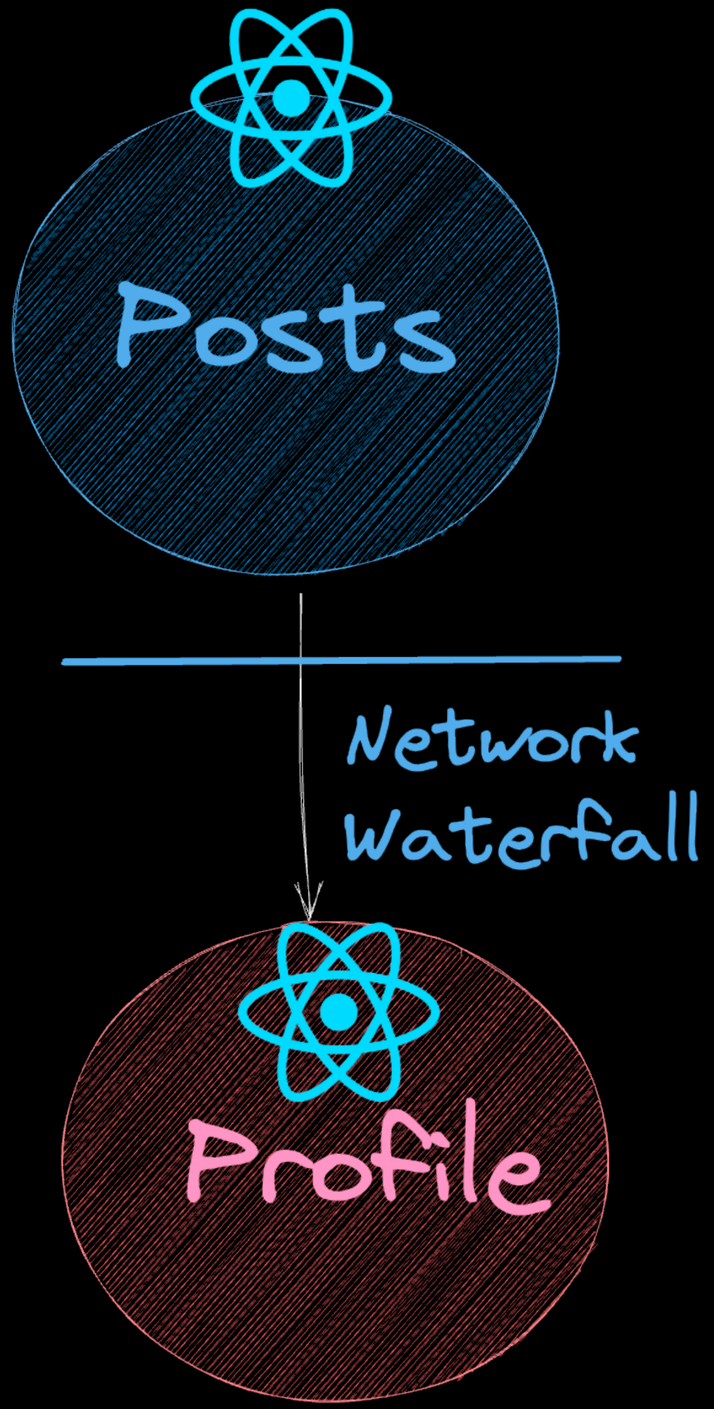
https://github.com/hyunjinee/GraphQL/issues/1

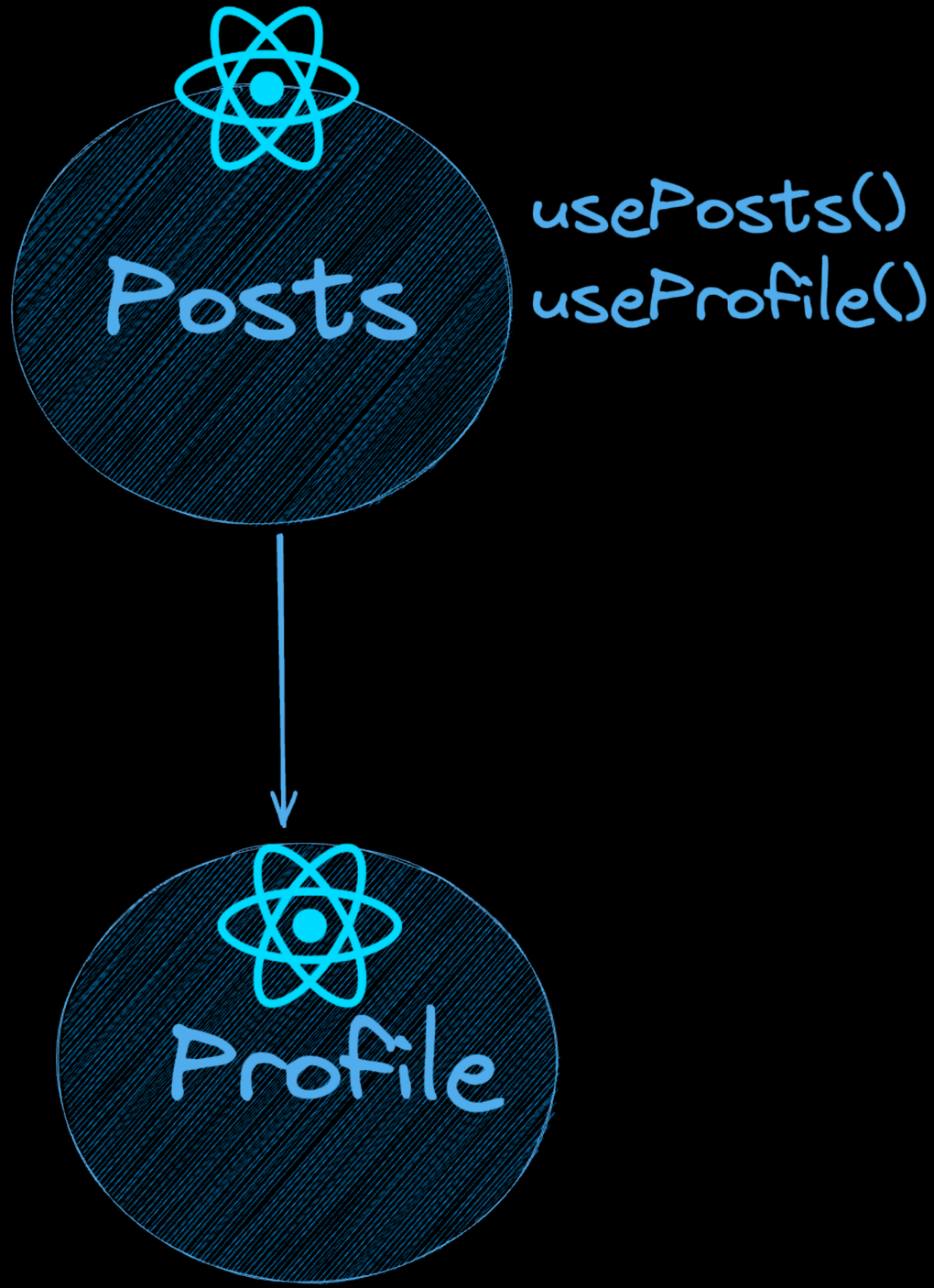


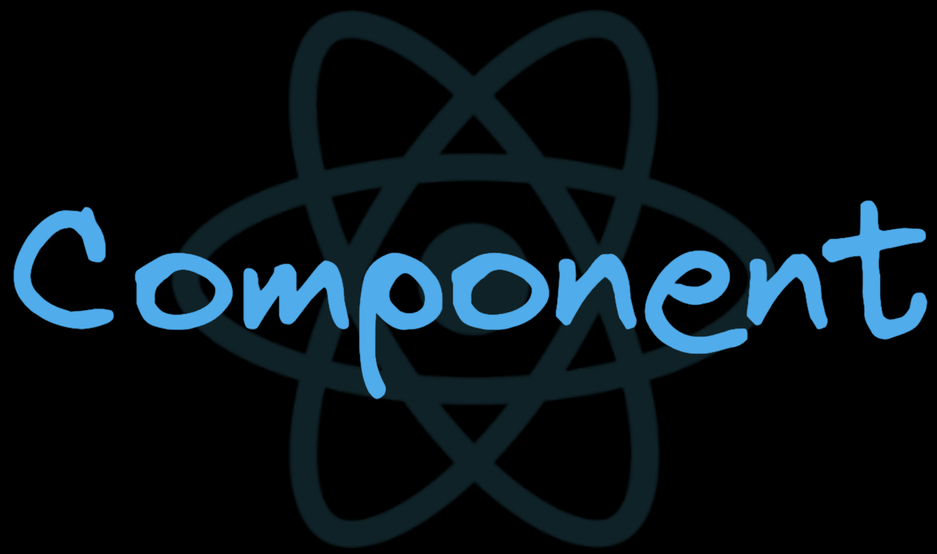


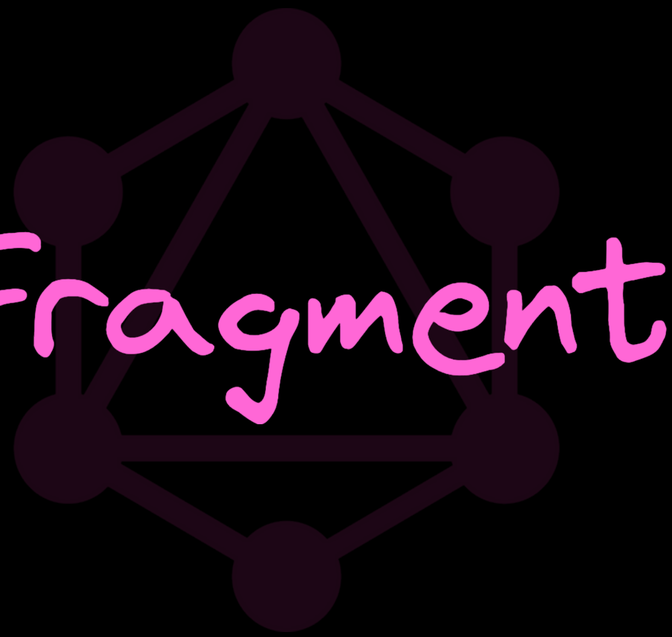

Blog
Posts
profile
profile

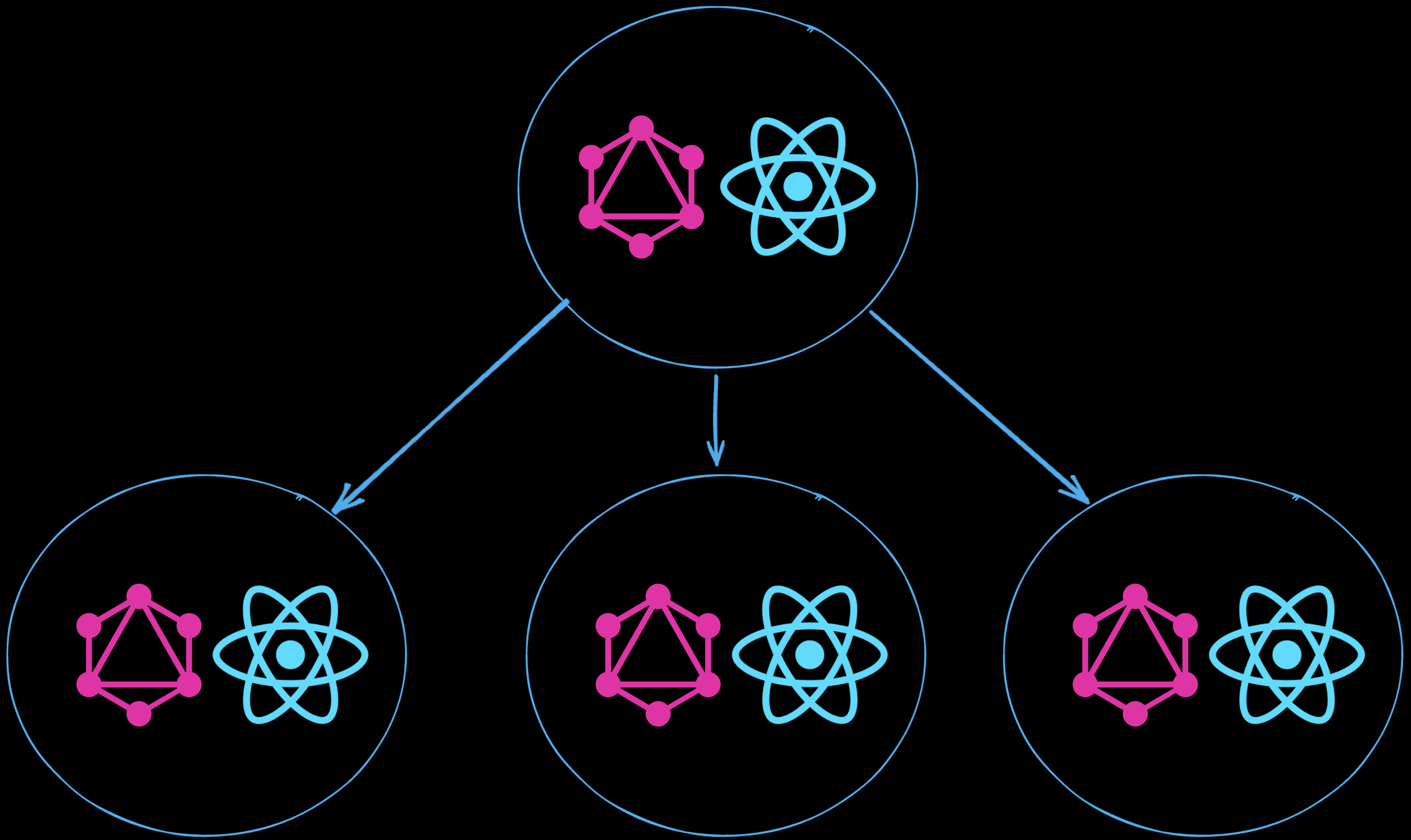

Blog
Posts
profile
profile

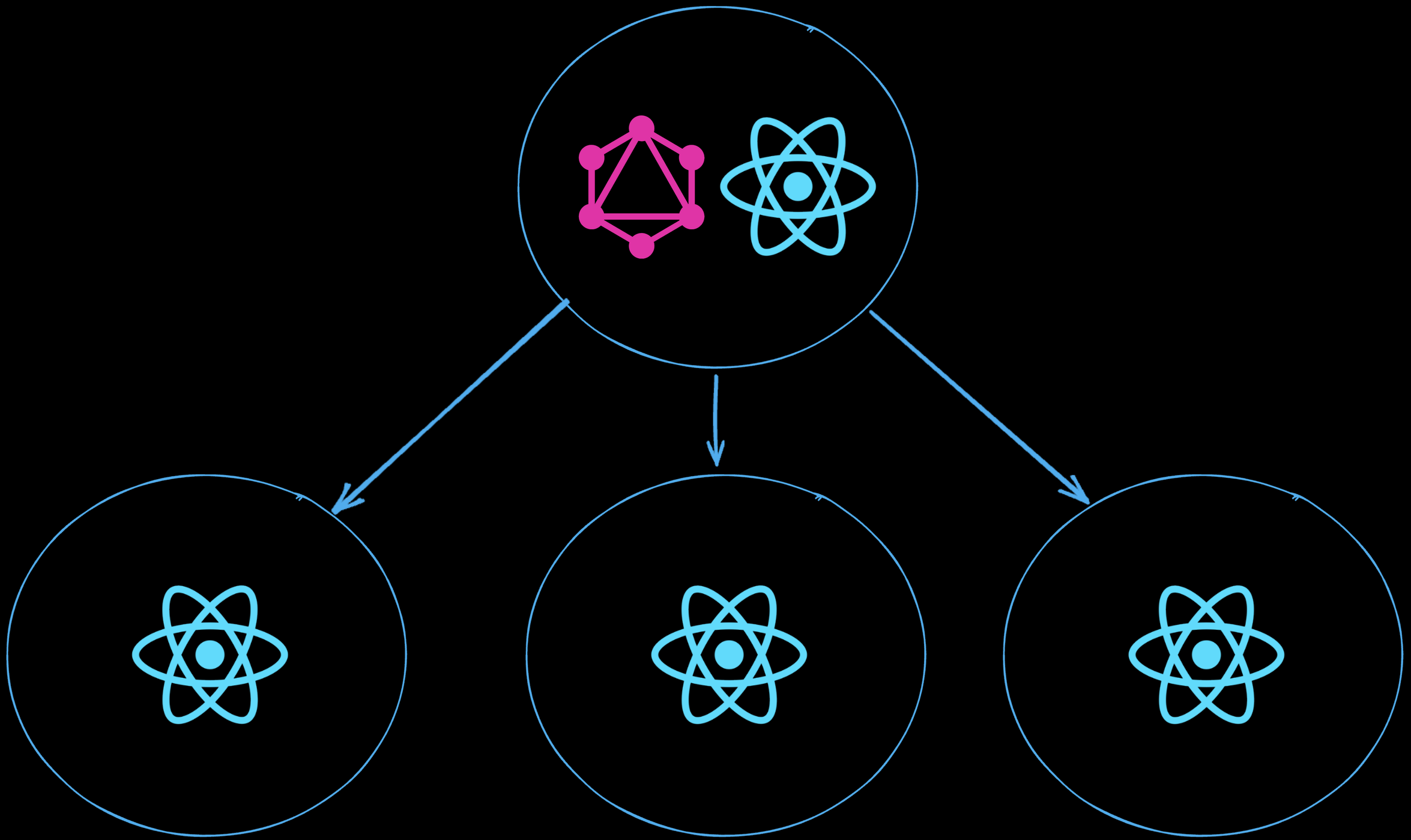

Posts
Network
Waterfall
Profile

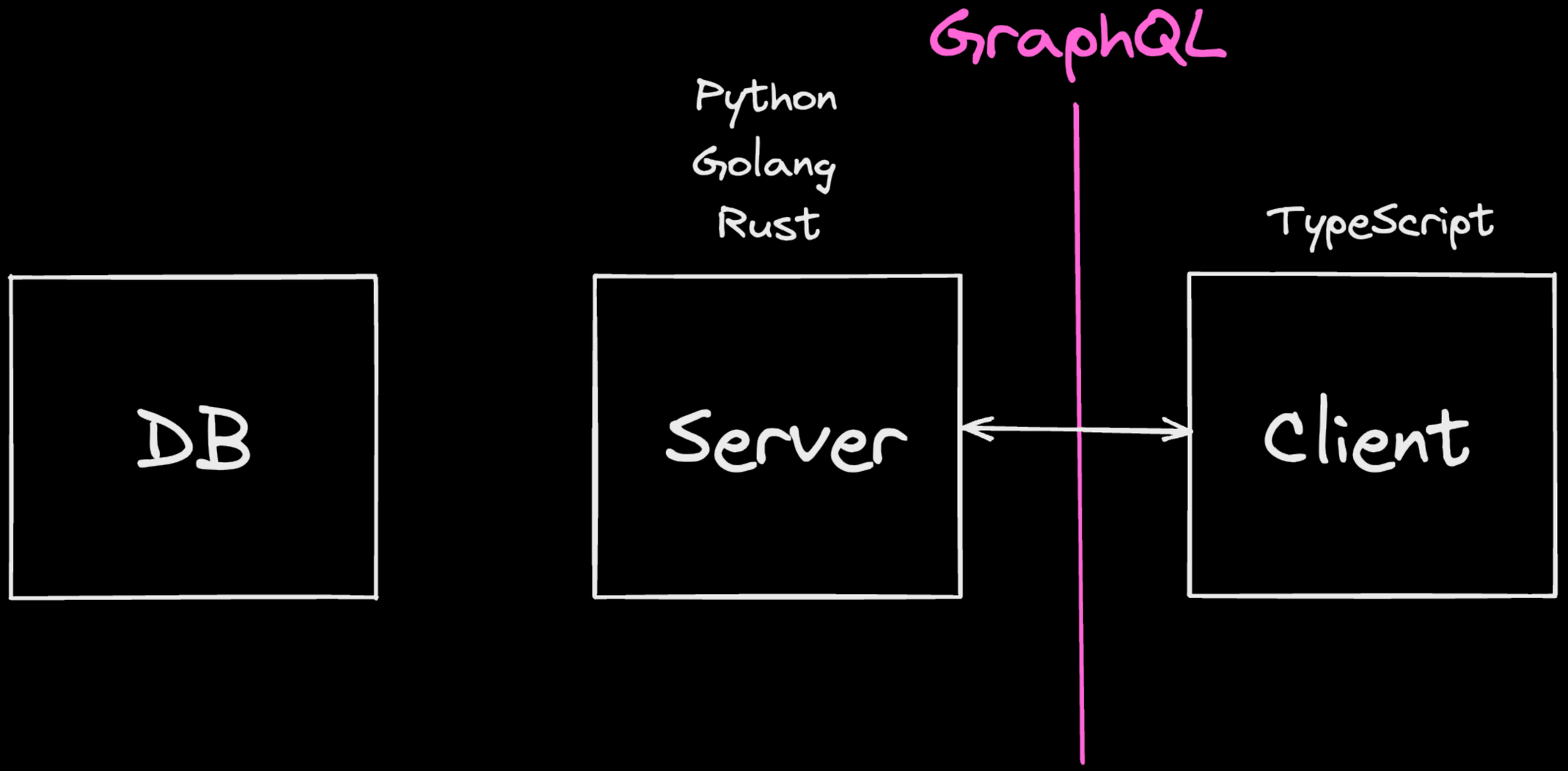

Posts
usePosts()
useProfile()
Profile

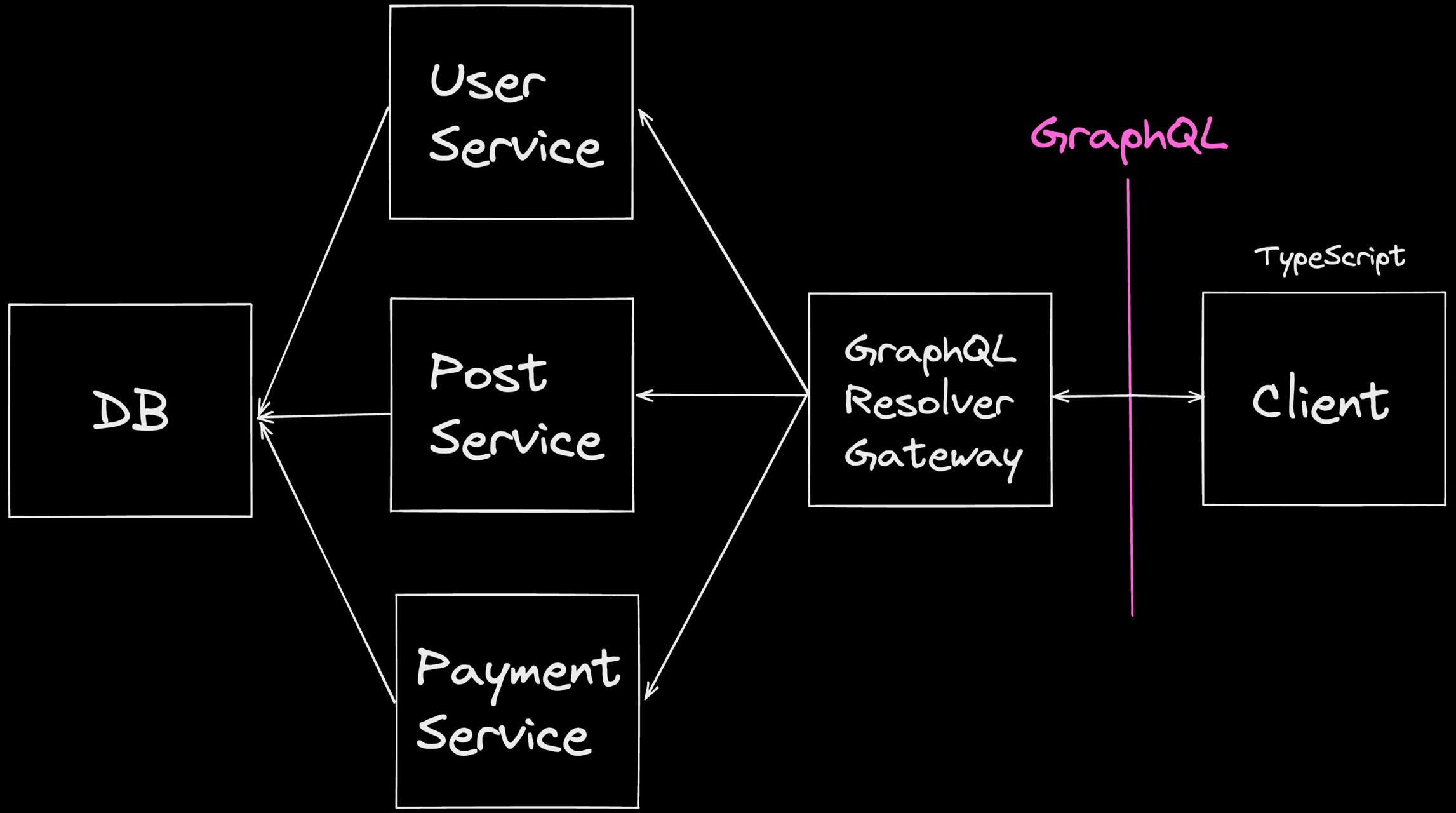

Component

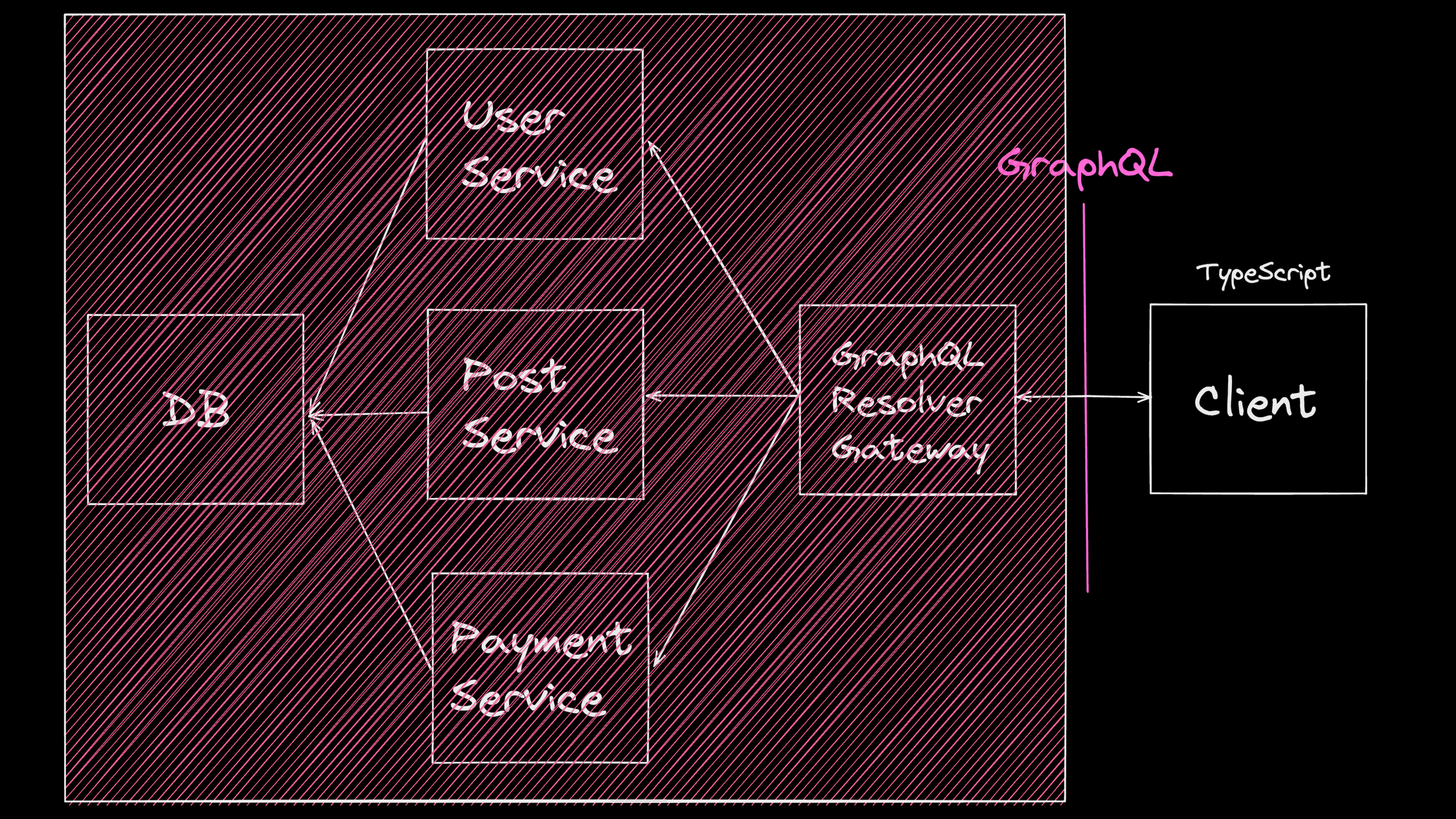

Fragment

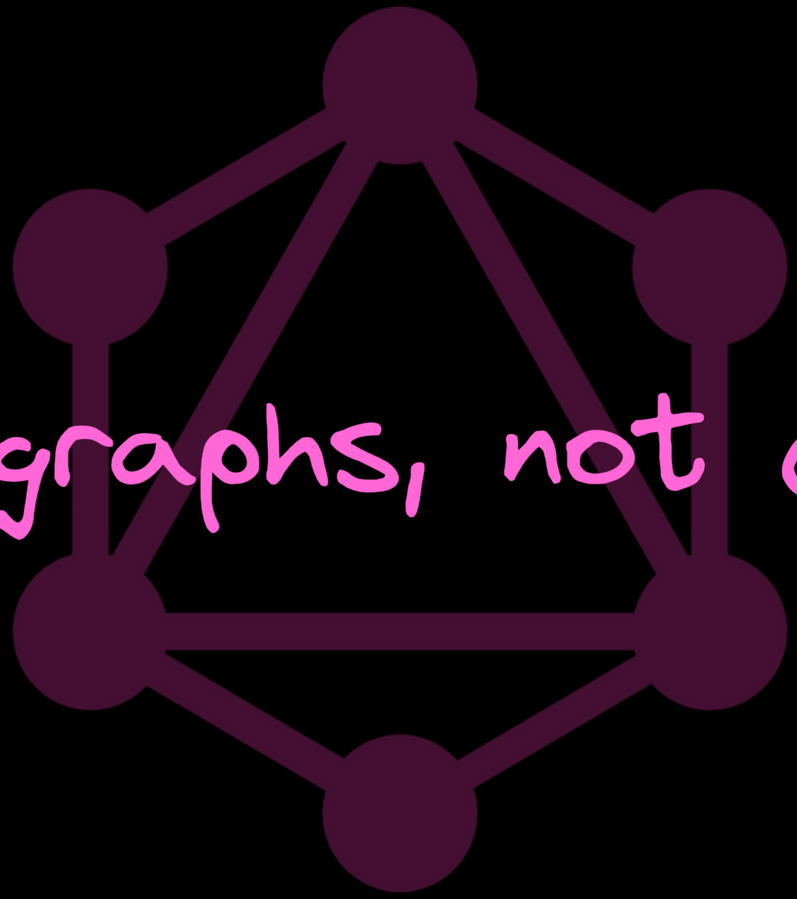





GraphQL
Python
Golang
Rust
TypeScript
DB
Server
Client


User Service
Post Service
Payment Service
DB
GraphQL Resolver Gateway
GraphQL
TypeScript
Client


User Service
GraphQL
TypeScript
DB
Post Service
GraphQL Resolver Gateway
Client
Payment Service



Think in graphs, not endpoints