

CS 스터디 21주차

GraphQL



GraphQL

GraphQL의 정의



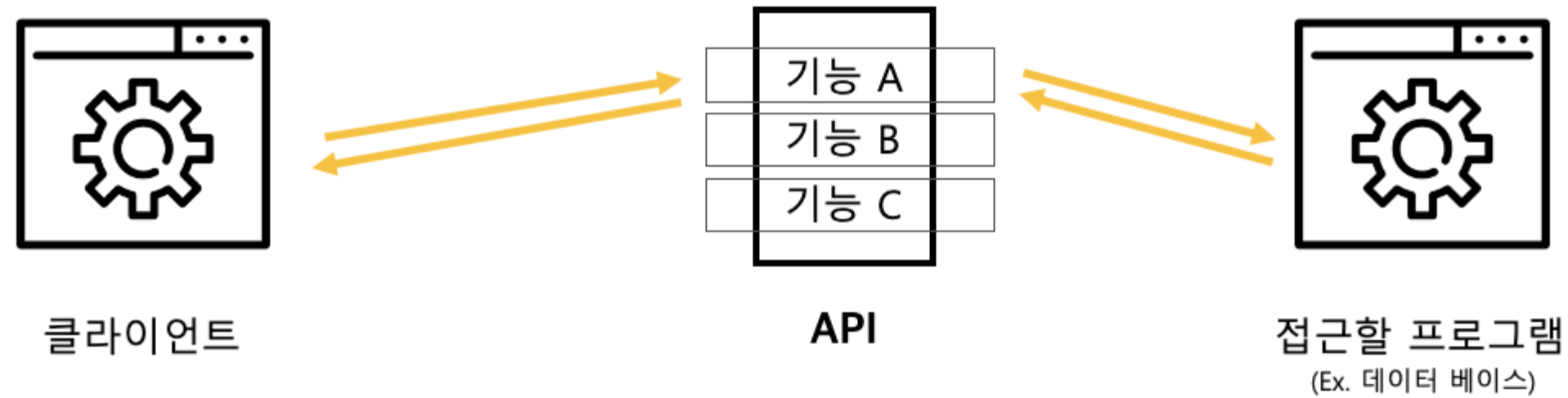
“정보를 사용하는 측에서 원하는 대로 정보를 가져올 수 있고,
보다 편하게 정보를 수정할 수 있도록 하는 표준화된 Query language”

API를 만들 때 사용할 수 있는 쿼리 언어

GraphQL

API?

프로그램과 프로그램을 연결시켜주는 매개체



서로 다른 프로그램끼리 통신을 하는 경우 사용하는 방법
API는 규칙들의 집합

REST API

REST API?

REpresentational S tate T ransfer
자원의 상태를 표현해 전송하는 방법

GraphQL vs REST API

왜 GraphQL?

REST API의 문제점

1. Over-Fetching

2. Under-Fetching

GraphQL vs REST API

Over-Fetching

사용자의 id와 name만을 가져오고 싶다.

```
{  
  "id" : 1,  
  "name" : "eunji",  
  "age" : 26,  
  "address" : "korea"  
}
```

필요 없는 데이터도 호출하게 됨

리소스의 낭비

```
{  
  "id":1,  
  "name":"eunji"  
}
```

이런식의 API가 무수히 많이 생길 것

개발 속도가 느려짐

GraphQL vs REST API

Under-Fetching

사용자의 정보와 해당 사용자가 작성한 글을 가져오고 싶다.

/user/info/1
/user/board/1

비효율적

GraphQL vs REST API

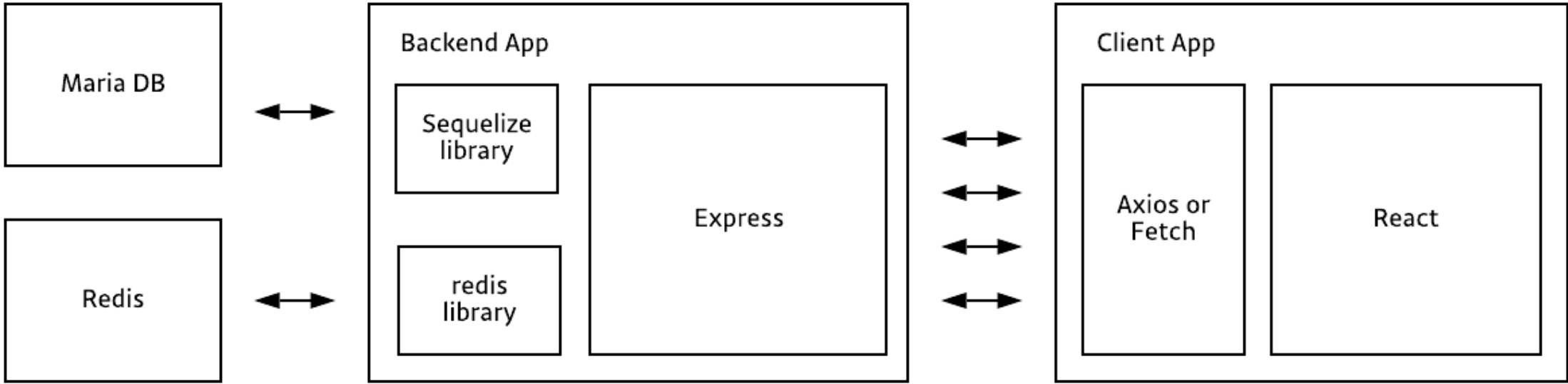
왜 GraphQL?

GraphQL은 하나의 API로 원하는 정보를 불러올 수 있다

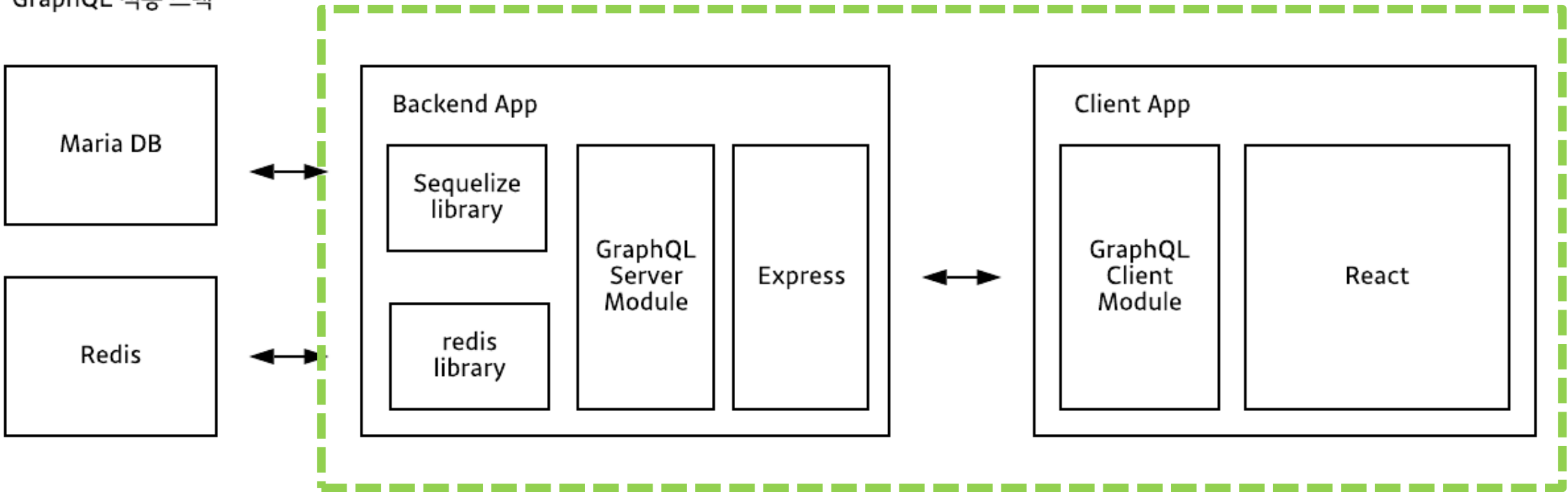
GraphQL vs REST API

왜 GraphQL?

일반 HTTP API 적용 스택

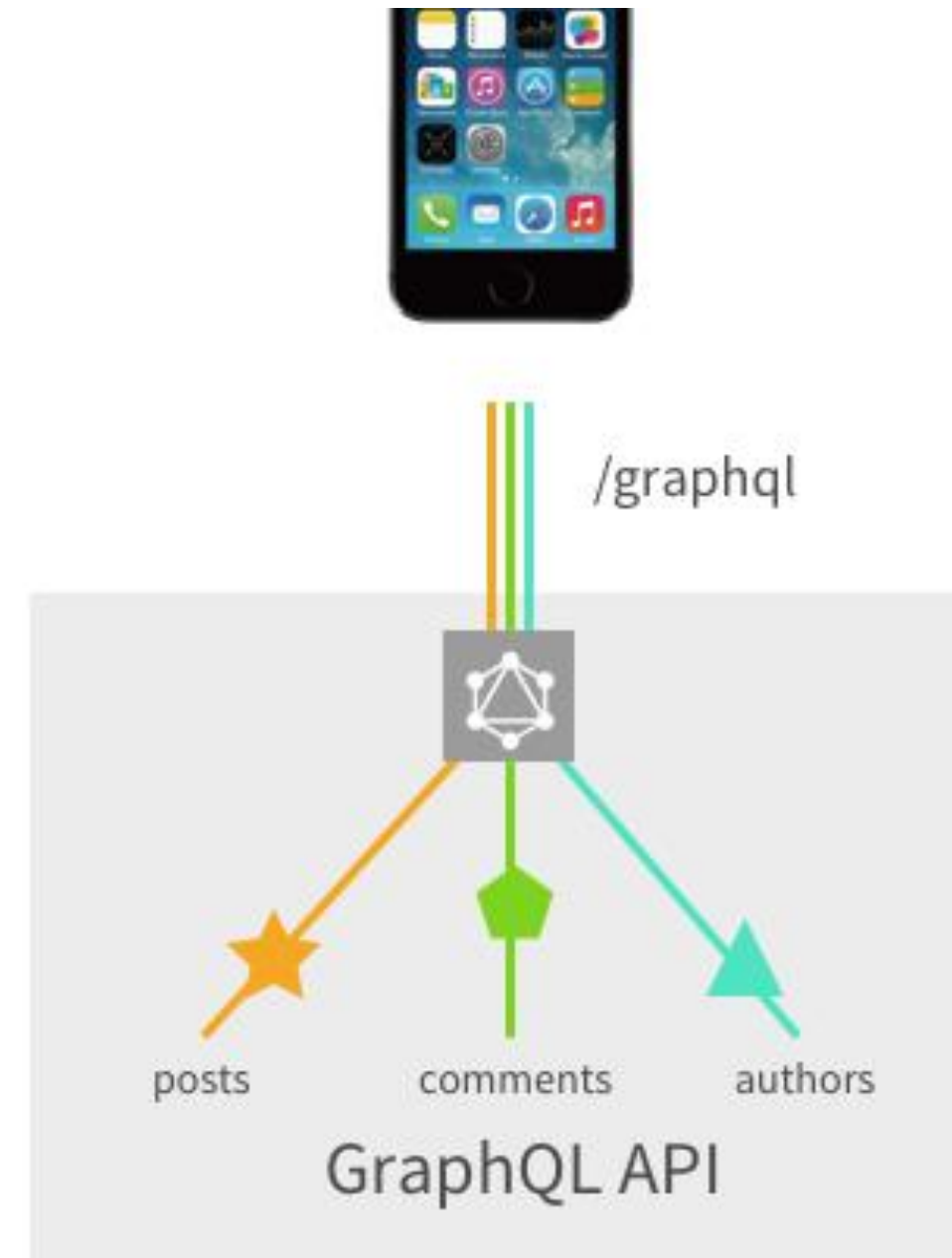


GraphQL 적용 스택



GraphQL vs REST API

왜 GraphQL?



GraphQL

GraphQL의 생김새

```
query {  
  person(personID: 1) {  
    name  
    height  
    area  
  }  
}
```

클라이언트 쿼리

```
{  
  "data": {  
    "person": {  
      "name": "Eunji",  
      "height": 160,  
      "area": "Daegu"  
    }  
  }  
}
```

응답

GraphQL

GraphQL의 생김새

```
// GraphQL

// Endpoint
query | mutation: /graphql

// GET
query {
  post(id: 1) {
    id
    post_header
  }
}

// POST, DELETE, PATCH|PUT
mutation {
  updatePost(id: 1, post_header: "바꿀 제목") {
    id
  }
}
```

GET과 같은 read 역할을 하는 요청은 query,
나머지 객체에 변화를 주는 요청은 mutation

GraphQL

GraphQL의 장점

HTTP 응답의 Size를 줄일 수 있다.

HTTP 요청의 횟수를 줄일 수 있다.

서버의 부하를 줄일 수 있다.

GraphQL

GraphQL의 단점

백엔드와 클라이언트 개발자 모두 러닝 커브가 있다.

단순한 서비스에서는 오히려 사용하기 복잡하다.

요청이 text로 날아가 File 전송을 구현하기 어렵다.

GraphQL

GraphQL or REST API

GraphQL

서로 다른 모양의 다양한 요청들에 대해 응답하고 싶을 때
대부분의 요청이 CRUD일 때

REST API

HTTP와 HTTPS에 대한 캐싱을 사용하고 싶다
File 전송 등 단순한 Text로 처리되지 않을 때
요청의 구조가 정해져 있을 때

GraphQL

GraphQL or REST API

둘 다 사용 가능하다!

끝 :)

