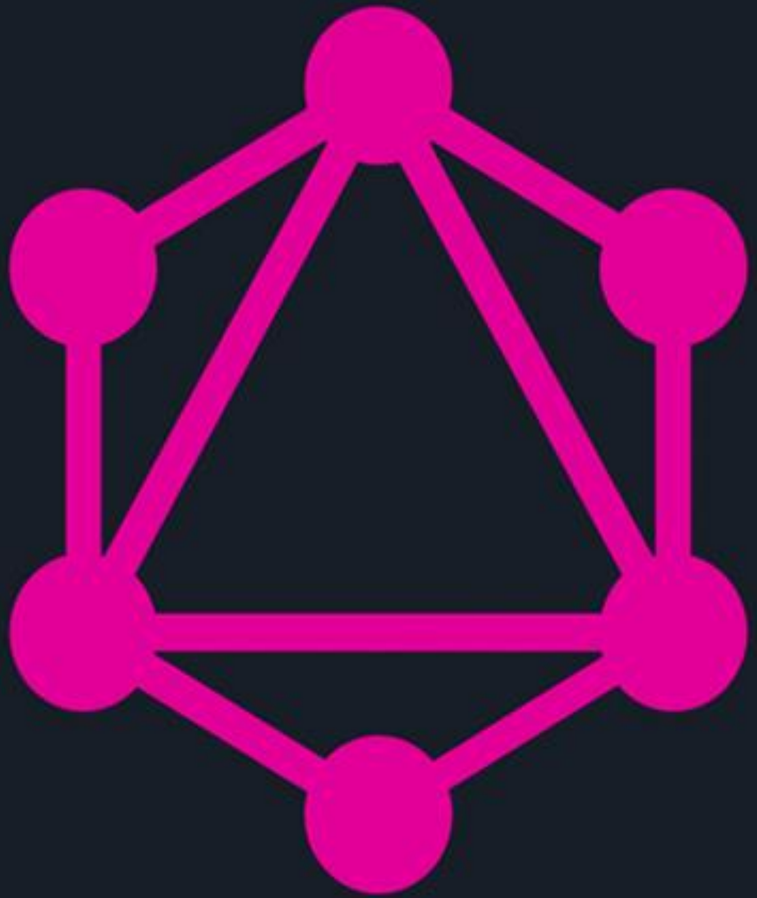


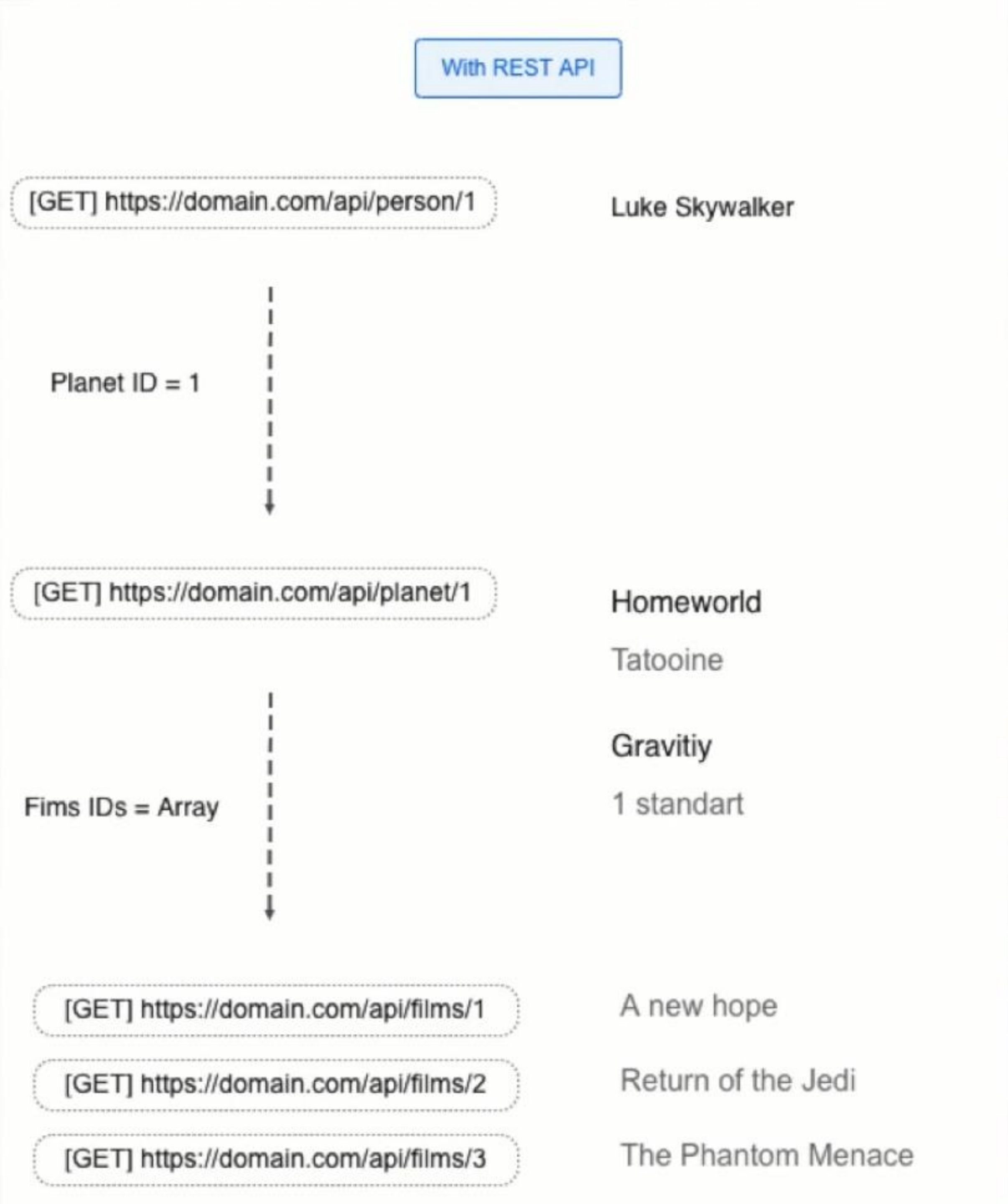
GraphQL

Star Wars REST API

https://swapi.dev/

Star Wars GraphQL API

https://graphql.github.io/swapi-graphql/



With REST API

[GET] https://domain.com/api/person/1

Luke Skywalker

Planet ID = 1

[GET] https://domain.com/api/planet/1

Homeworld

Tatooine

Gravitiy

1 standart

Fims IDs = Array

[GET] https://domain.com/api/films/1

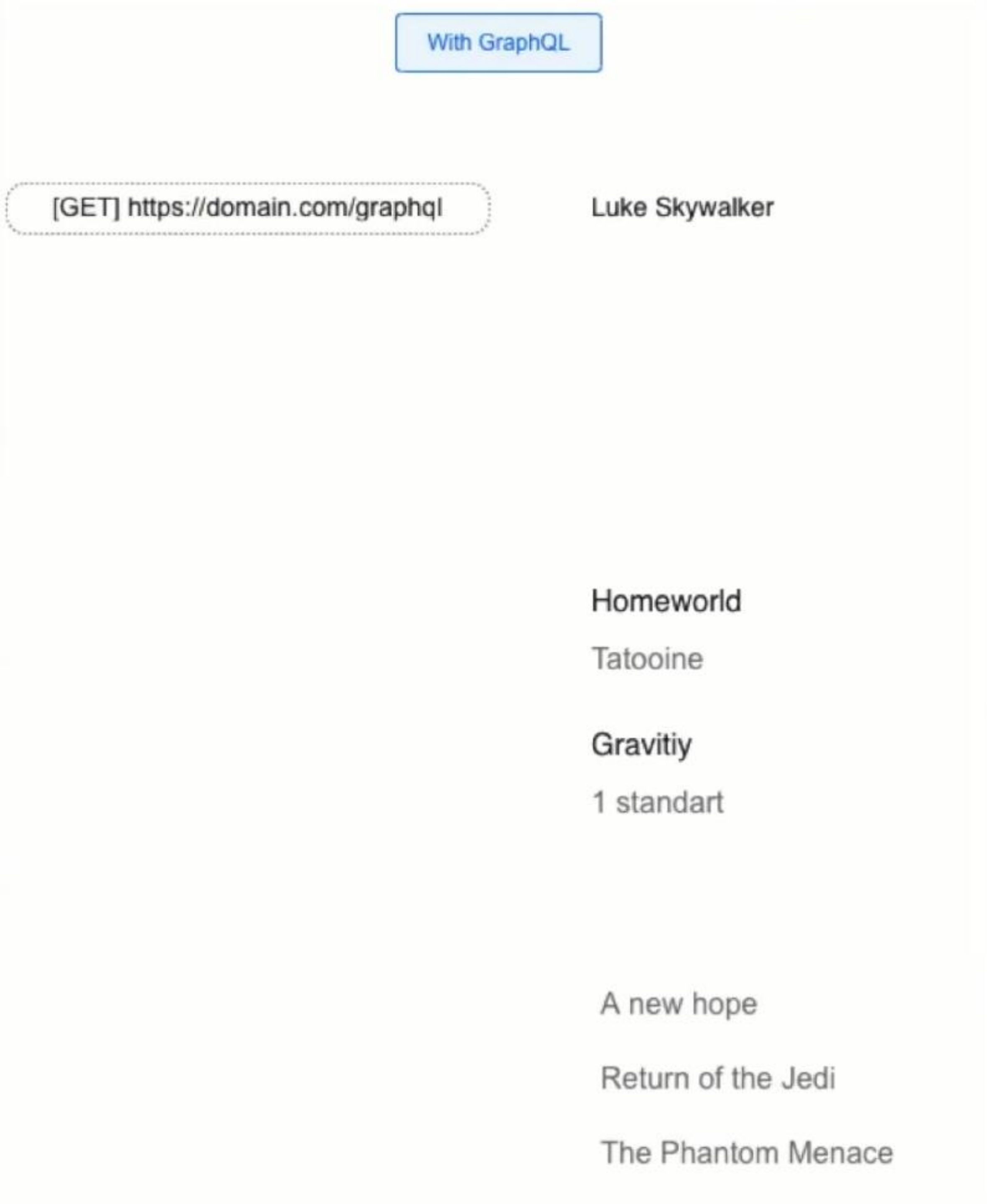
A new hope

[GET] https://domain.com/api/films/2

Return of the Jedi

[GET] https://domain.com/api/films/3

The Phantom Menace



[GET] https://domain.com/graphql

Luke Skywalker

Homeworld

Tatooine

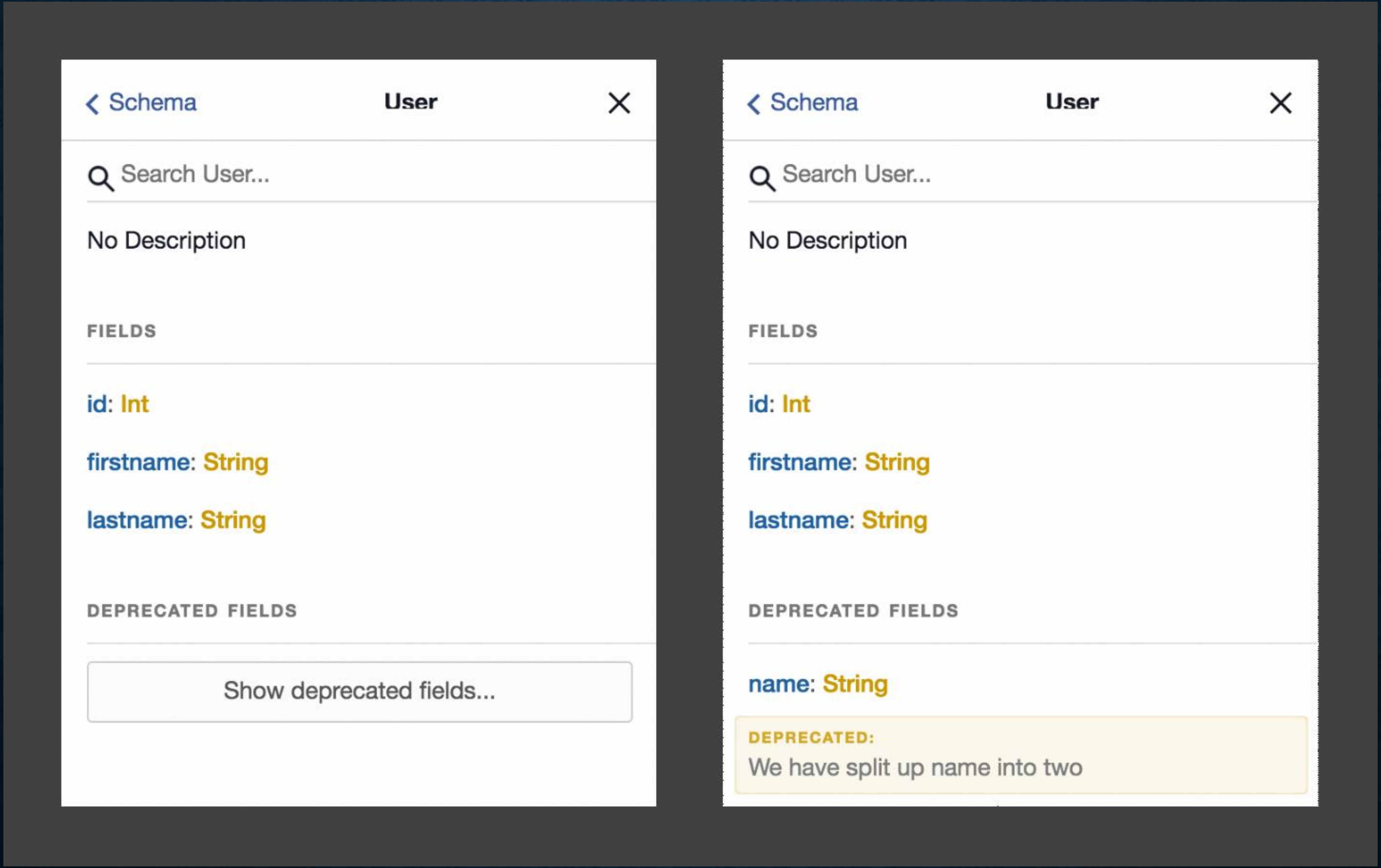
Gravitiy

1 standart

A new hope

Return of the Jedi

The Phantom Menace



< Schema User ✕

Search User...

No Description

FIELDS

id: Int

firstname: String

lastname: String

DEPRECATED FIELDS

< Schema User ✕

Search User...

No Description

FIELDS

id: Int

firstname: String

lastname: String

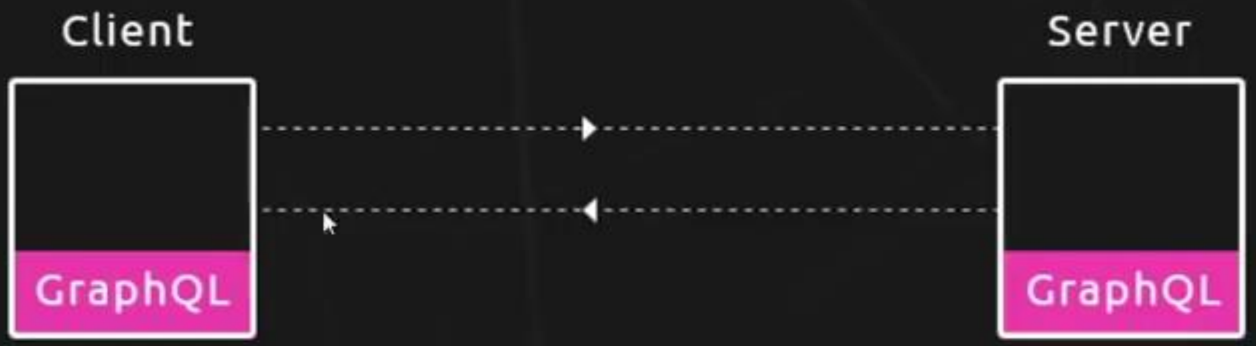
DEPRECATED FIELDS

name: String

DEPRECATED:
We have split up name into two

What is GraphQL?

- GraphQL is a powerful query language



- Allows for a more flexible & efficient approach than REST

A RESTful Approach...

- Endpoint for getting a particular book:

  domain.com/books/:id

  title, genre, reviews, authorId

- Endpoint for getting the author info of that book:

  domain.com/authors/:id

  name, age, biography, bookIds

A GraphQL Approach...

- Query to get book data and it’s author data (AND the other books):

```
{
    book(id: 123) {
        title
        genre
        reviews
        author {
            name
            bio
            books {
                name
            }
        }
    }
```

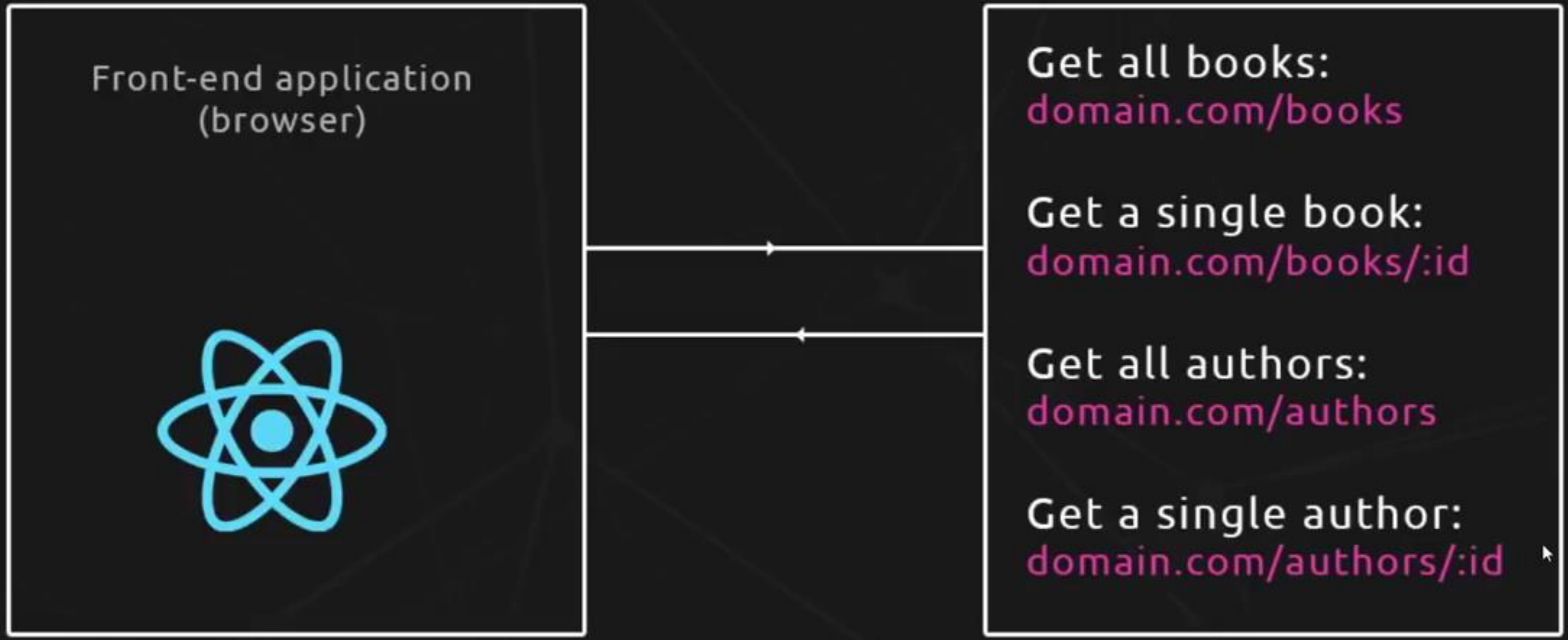
A GraphQL Approach...

- Query to get book data and it’s author data (AND the other books):

```
{
    book(id: 123) {
        title
        genre
        reviews
        author {
            name
            bio
            books {
                name
            }
        }
    }
```

```
{
    book(id: 123) {
        title
        author {
            name
            books {
                name
            }
        }
    }
}
```

REST API Endpoints



Front-end application
(browser)

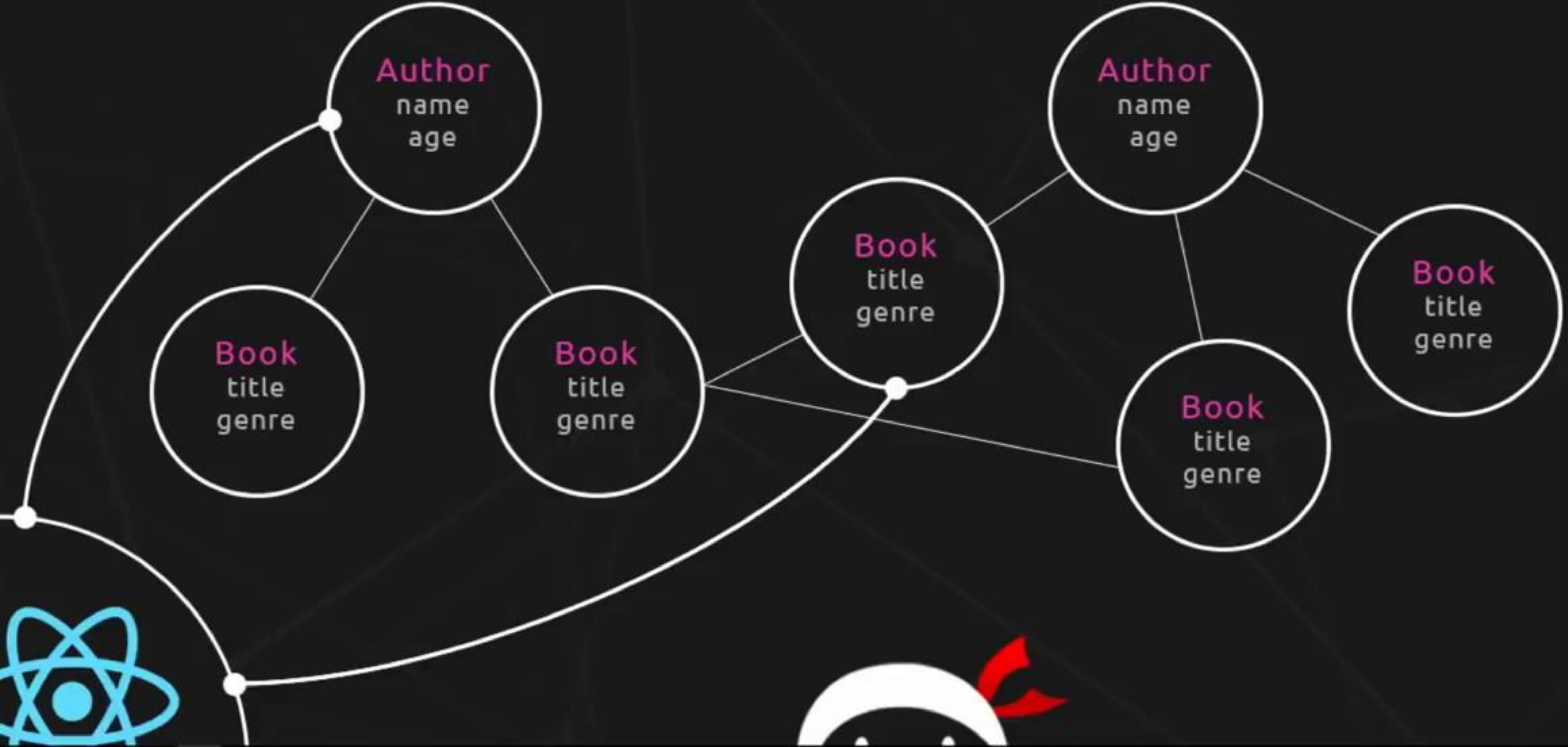
Get all books:
domain.com/books

Get a single book:
domain.com/books/:id

Get all authors:
domain.com/authors

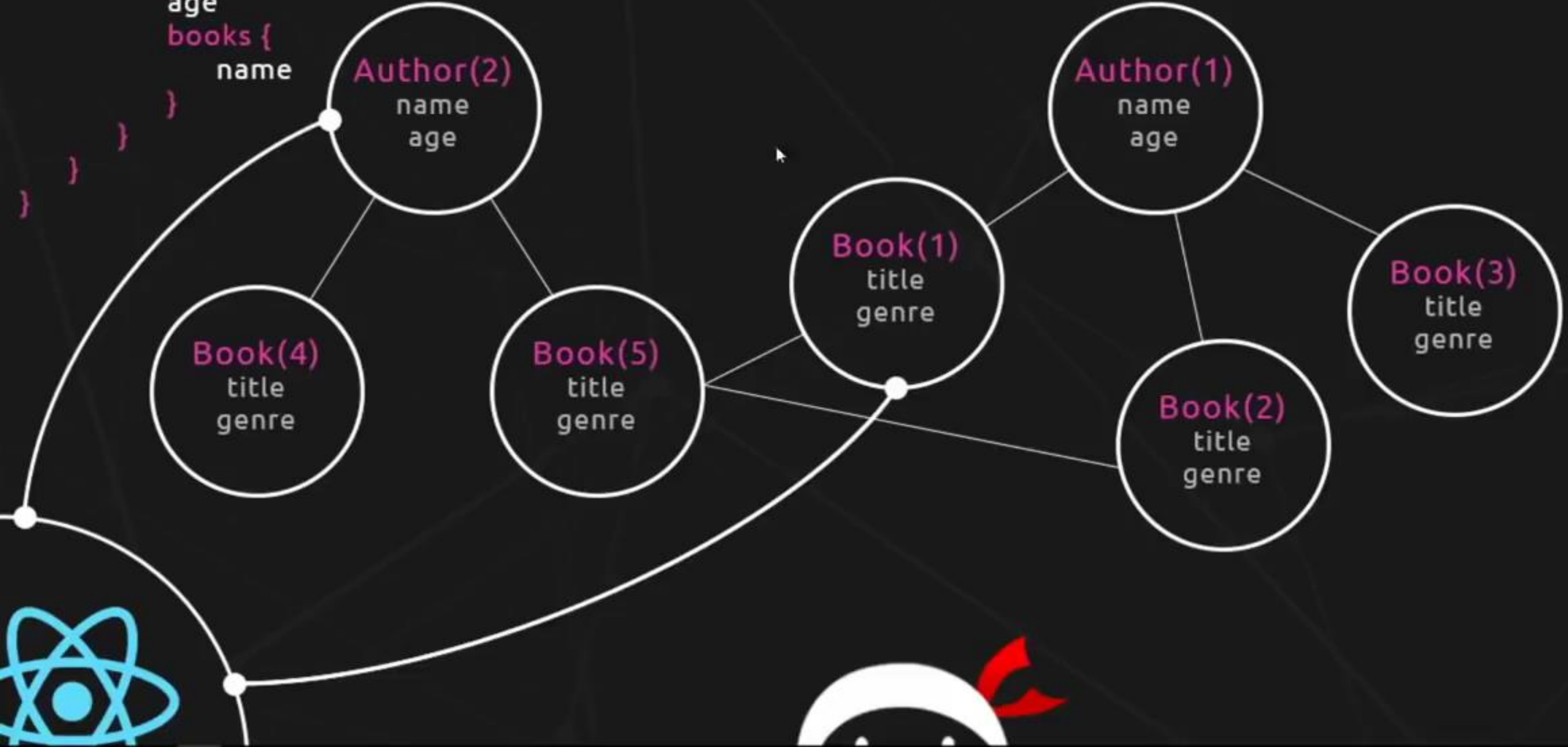
Get a single author:
domain.com/authors/:id

Walking the Graph...



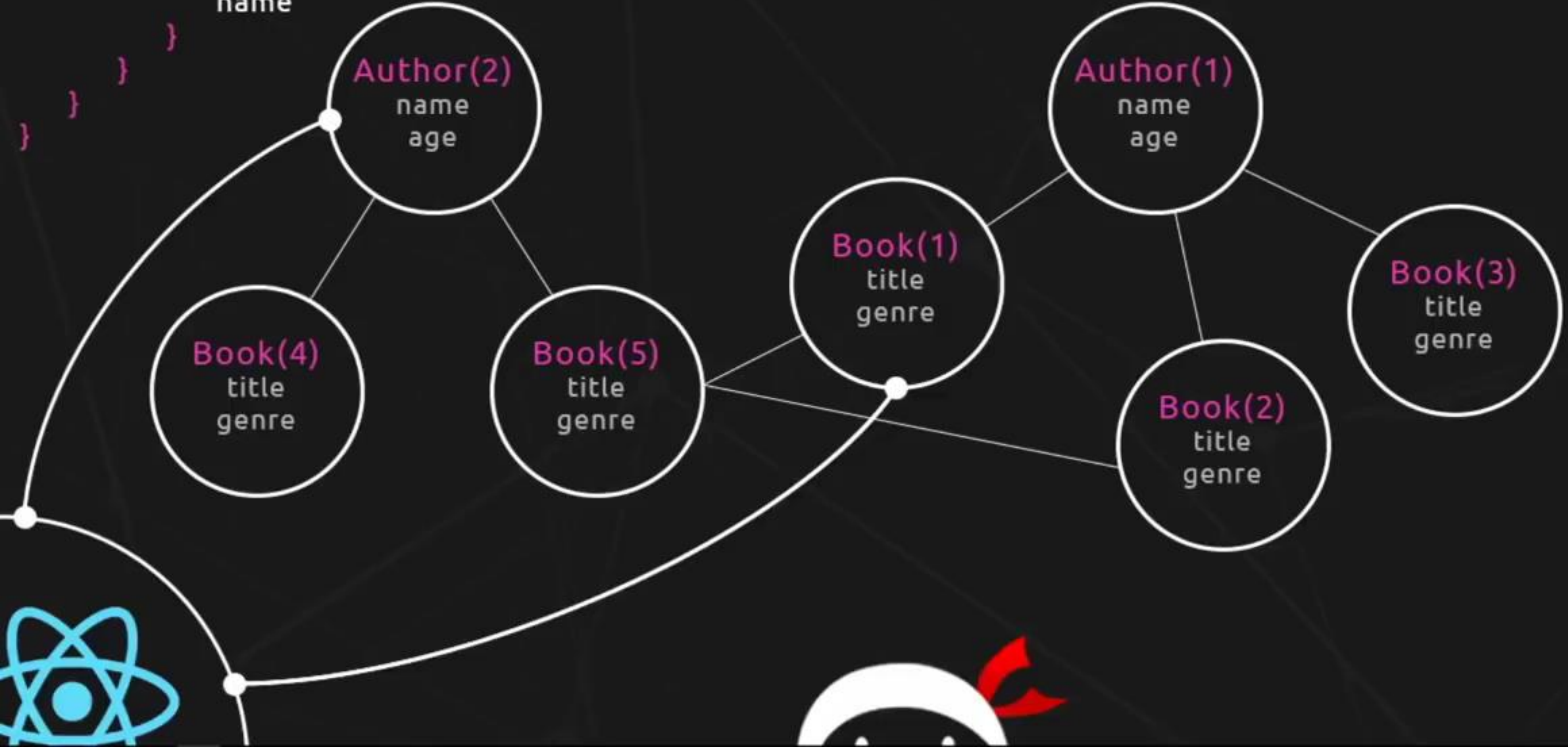
```
{
  book(id: 1) {
    title
    genre
    author {
      name
      age
      books {
        name
      }
    }
  }
}
```

Project Overview



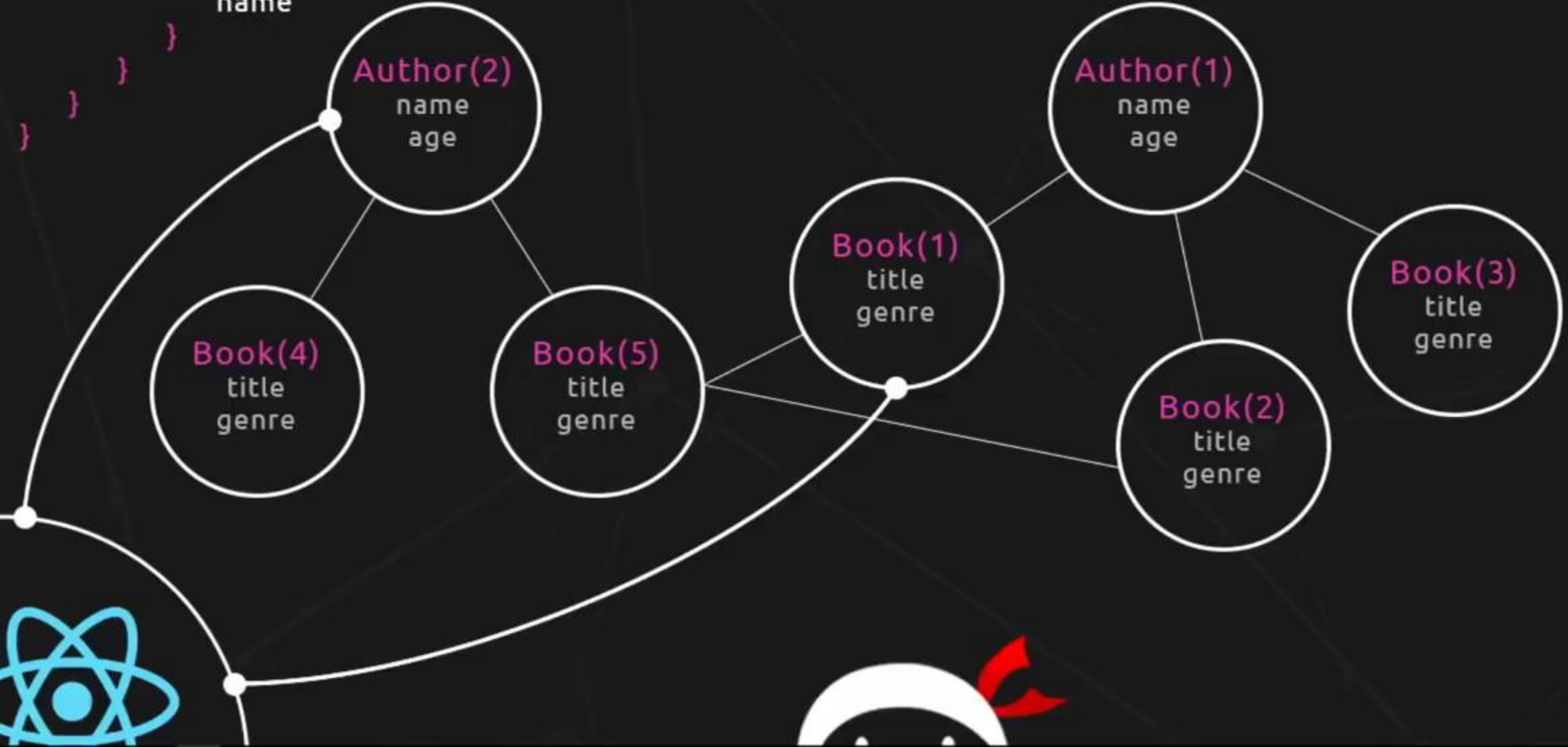
Project Overview

```
{
    Author(id: 2) {
        name
        books {
            name
            similar {
                name
            }
        }
    }
}
```



Project Overview

```
{
    Author(id: 2) {
        name
        books {
            name
            similar {
                name
            }
        }
    }
}
```


Author(2)
name
age
Author(1)
name
age
Book(1)
title
genre
Book(4)
title
genre
Book(5)
title
genre
Book(3)
title
genre
Book(2)
title
genre