BAO ZUN

GraphQL

前后端协同从未如丝顺滑

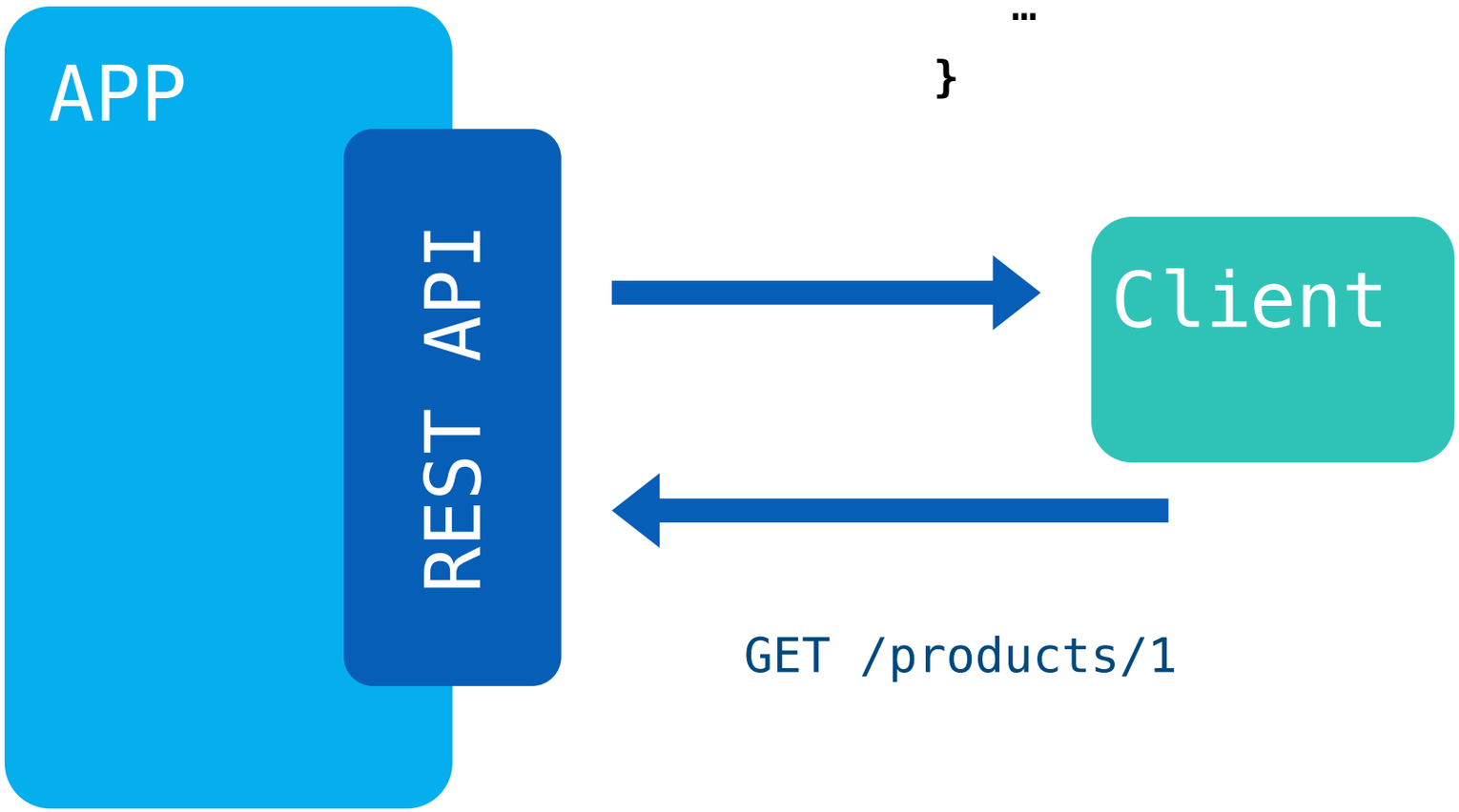
2018.07.14

陈嘉

TOPIC

1. 什么是 GraphQL
2. 我们如何使用 GraphQL
3. GraphQL in 前端

GraphQL




```
{
  "id": "1",
  "name": "product name",
  "price": 1.23,
   …
}
```

BAO ZUN

GraphQL

OVER FETCHING

Tech Fleece cotton-blend jersey hooded top

```
[
  …
  {
    "id": "1111",
    "name": "Tech Fleece cotton-blend",
    "price": 143.00,
    "image": "http://…",
    "description": "Nike's hooded top is made ..."
    "skus": [
      {
        …
        "size": "L"
      }
    ]
  }
  …
]
```

BAO ZUN

GraphQL

OVER FETCHING

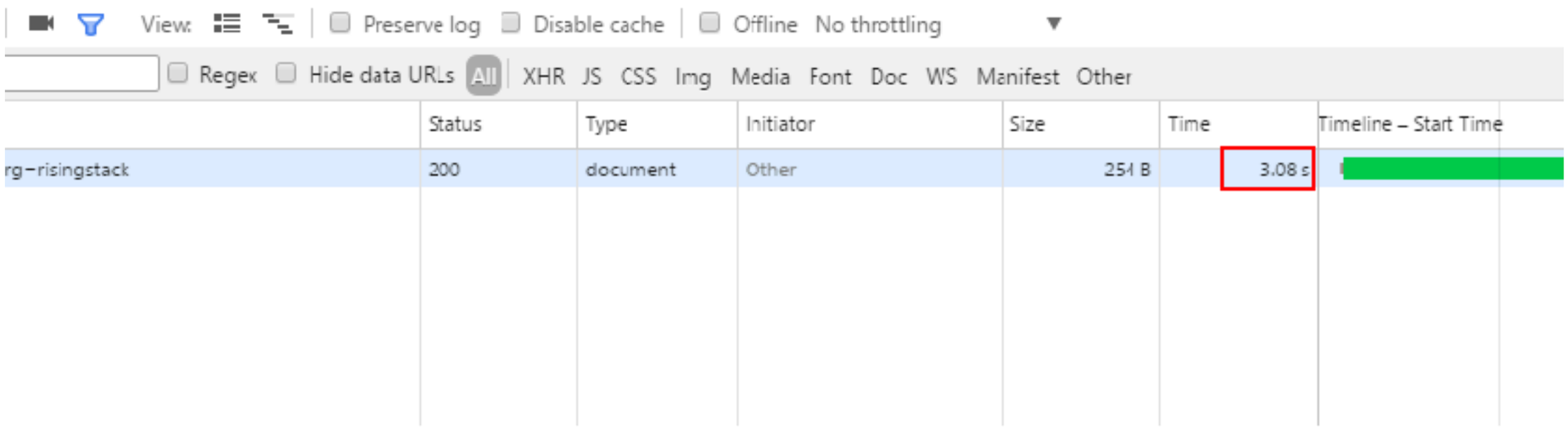
Tech Fleece cotton-blend jersey hooded top

```
[
  …
  {
    "id": "1111",
    "name": "Tech Fleece cotton-blend",
    "price": 143.00,
    "image": "http://…",
    "description": "Nike's hooded top is made ..."
    "skus": [
      {
        …
        "size": "L"
      }
    ]
  }
  …
]
```

BAO ZUN

GraphQL

OVER FETCHING



BAO ZUN

GraphQL

UNDER FETCHING

NIKE
Tech Fleece cotton-blend jersey hooded top
$143

```
[
  …
  {
    "id": "1111",
    "name": "Tech Fleece cotton-blend",
    "price": 143.00,
    “image": "http://…",
    "description": "Nike's hooded top is made ..."
    "skus": [
      {
        …
        "size": "L"
      }
    ]
  }
  …
]
```

BAO ZUN

GraphQL

TypeError: Cannot read property 'name' of undefined

App.render
src/App.js:14

```
  11 |       <img src={logo} className="App-logo" alt="logo" />
  12 |     </div>
  13 |     <p className="App-intro">
> 14 |       {user.profile.name}
  15 |     </p>
  16 |   </div>
  17 | );
```

View compiled

▼ 19 stack frames were expanded.

(anonymous function)
node_modules/react-dom/lib/ReactCompositeComponent.js:795

measureLifeCyclePerf
node_modules/react-dom/lib/ReactCompositeComponent.js:75

ReactCompositeComponentWrapper._renderValidatedComponentWithoutOwnerOrContext
node_modules/react-dom/lib/ReactCompositeComponent.js:794

ReactCompositeComponentWrapper._renderValidatedComponent
node_modules/react-dom/lib/ReactCompositeComponent.js:821

ReactCompositeComponentWrapper.performInitialMount
node_modules/react-dom/lib/ReactCompositeComponent.js:361

ReactCompositeComponentWrapper.mountComponent
node_modules/react-dom/lib/ReactCompositeComponent.js:257

问题

同志，快醒醒，API少了一个字段

问题

POST /users 创建用户

Implementation Notes

根据User对象创建用户

Response Class (Status 200)

Response Content Type */*

Parameters

Parameter	Value	Description	Parameter Type	Data Type
user	(required)	用户详细实体user	body	Model | Model Schema

```
{
  "age": 0,
  "id": 0,
  "name": "string"
}
```

Parameter content type: application/json

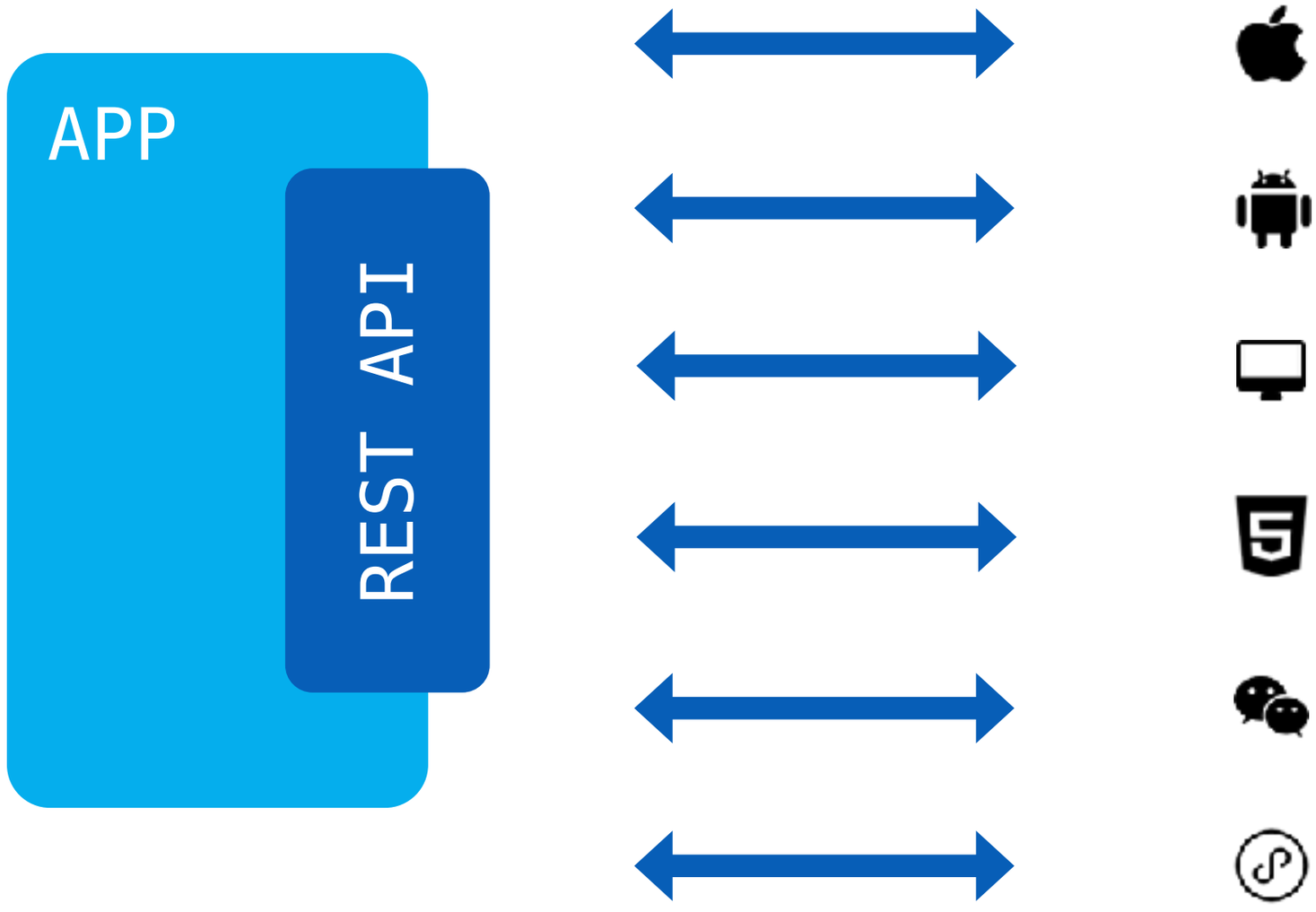
Click to set as parameter value

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

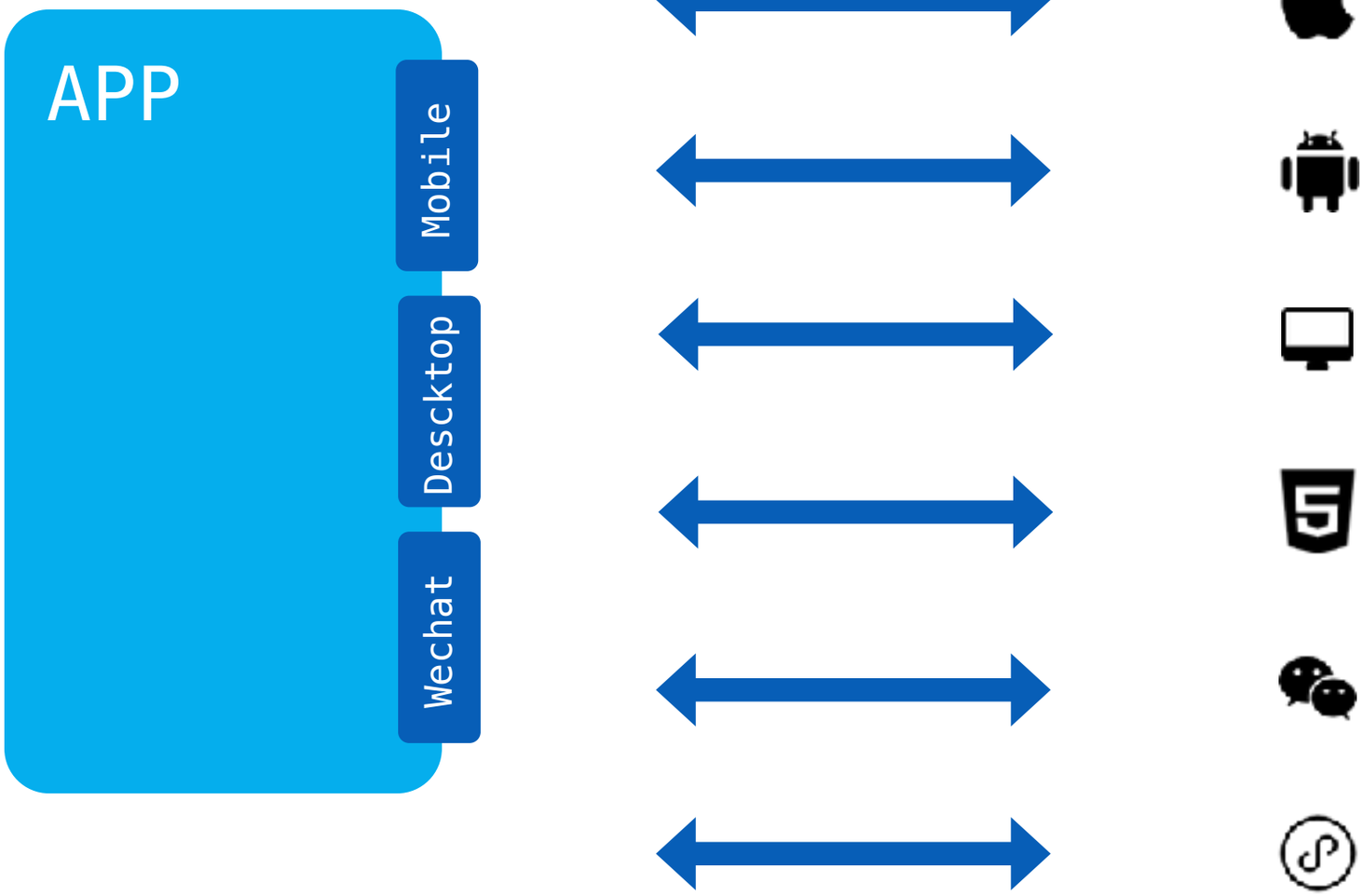
Try it out!

移动时代



GraphQL

移动时代

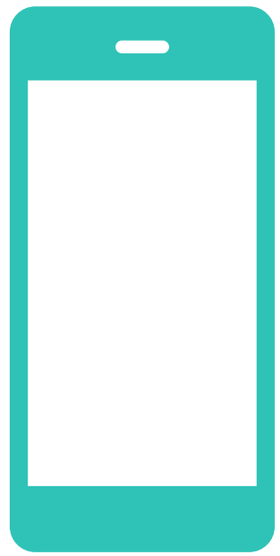


BAO ZUN

GraphQL

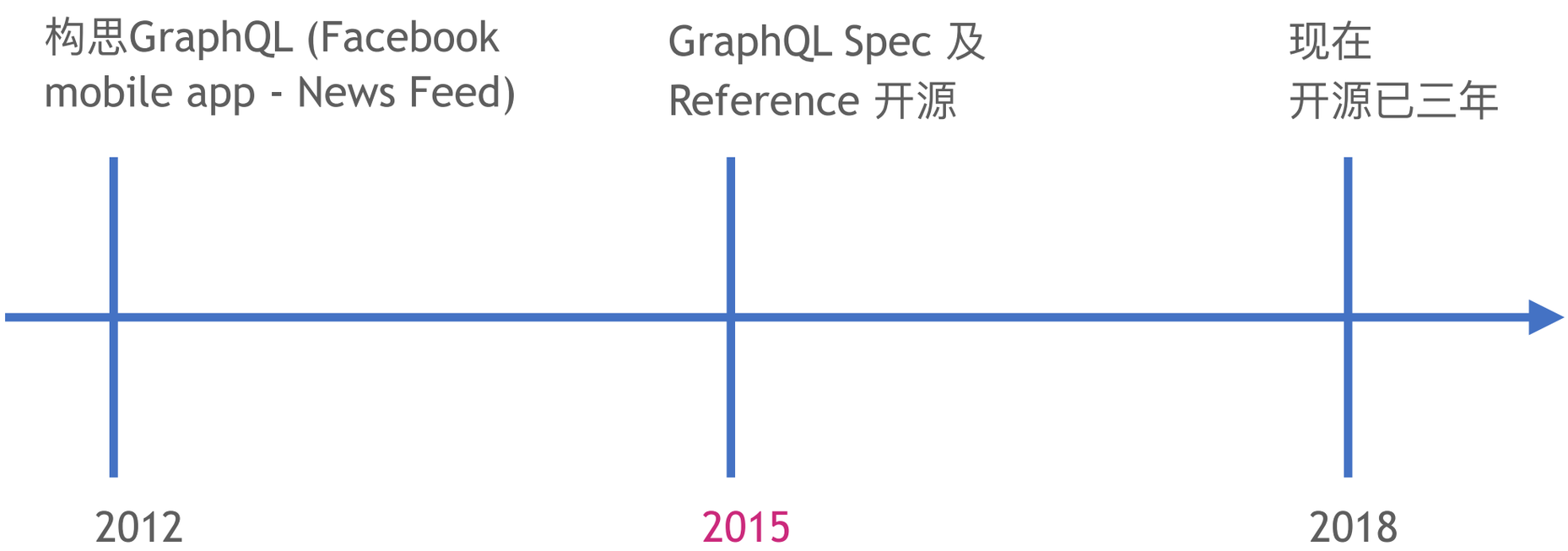
FQL

```
Select * from timelines;
```

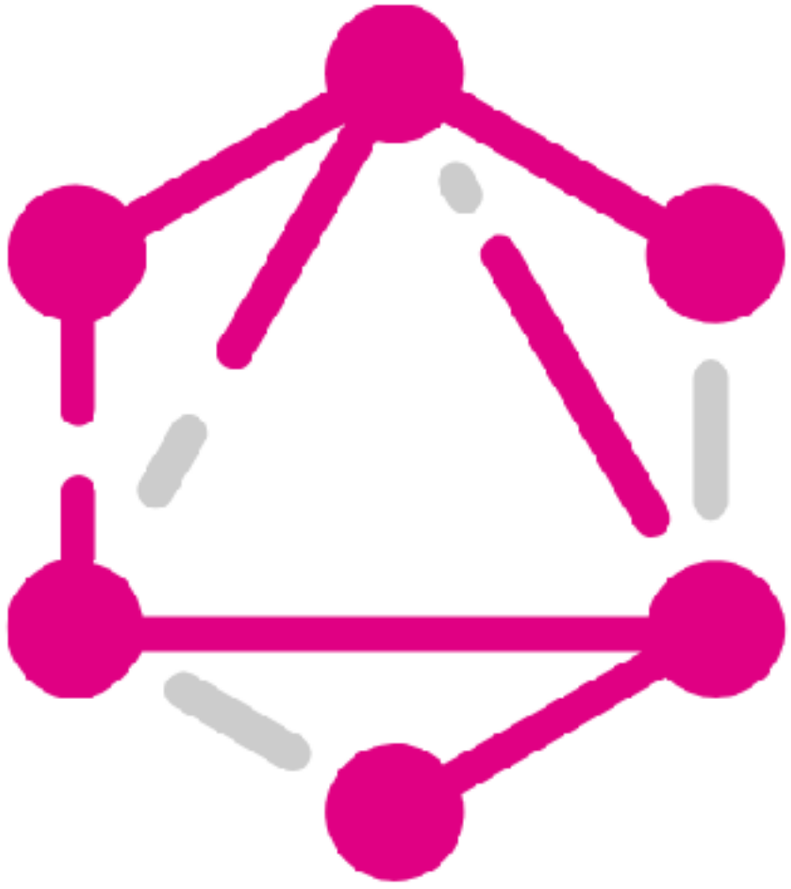


```
{
  “timelines": {…},
}
```

BAO ZUN

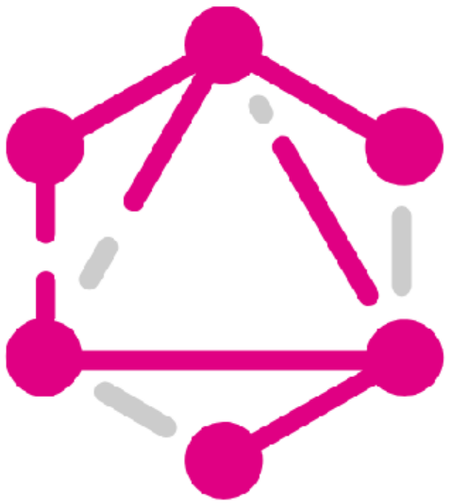

构思GraphQL (Facebook mobile app - News Feed)
GraphQL Spec 及 Reference 开源
现在 开源已三年
2012
2015
2018

GraphQL



GraphQL

BAO ZUN



GraphQL 是一个数据查询语言

GraphQL 不是什么！

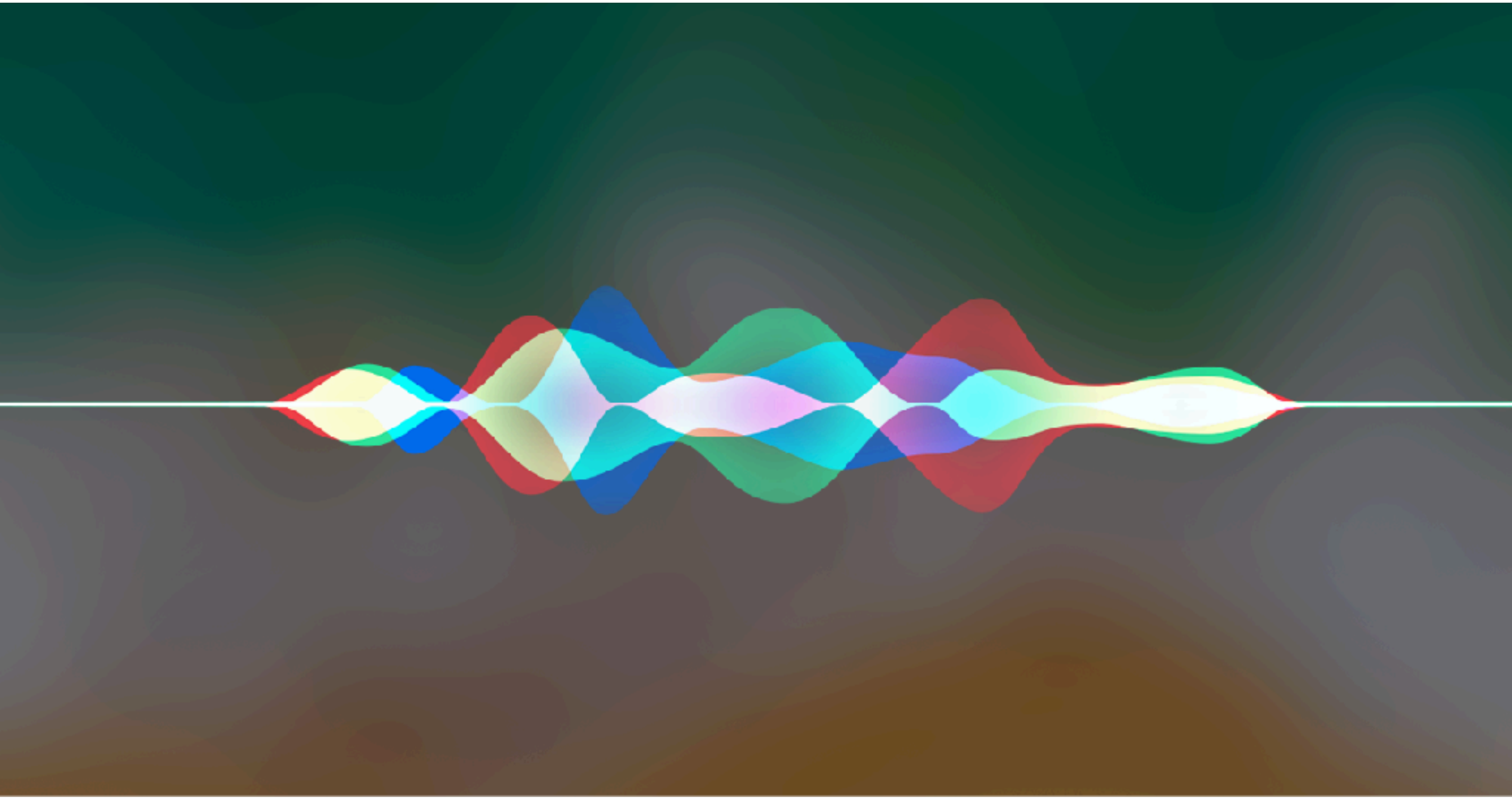
GraphQL 不是一个编程语言

GraphQL 不是数据库

GraphQL 不是某一个框架

GraphQL 不限制于某一个语言

GraphQL



Graphql - response 结构

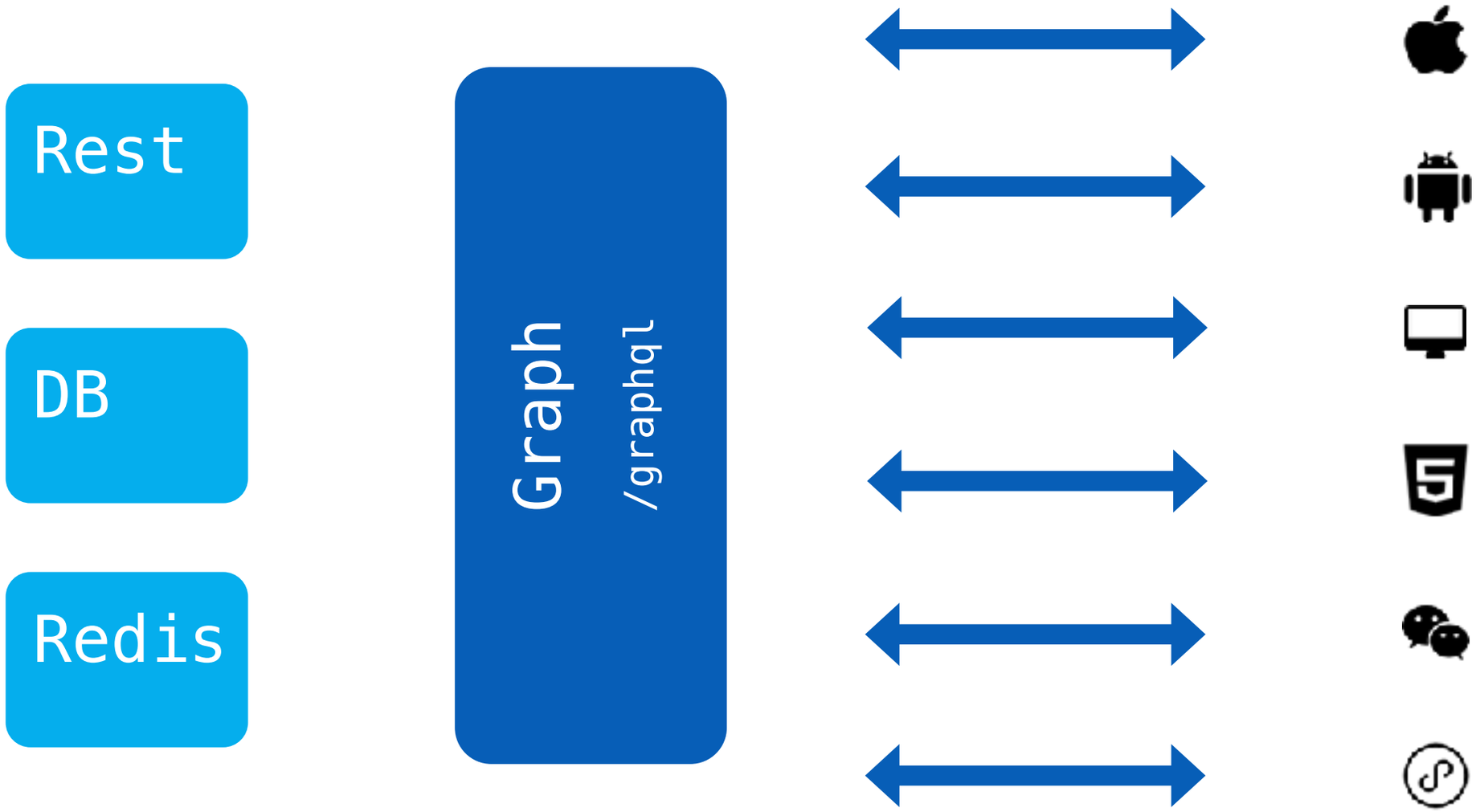
客户端驱动开发： 可预测的响应结构

```
query productList {
  product(id: 123) {
    name
    description
    picture {
      url
    }
  }
}
```

```
{
  "data": {
    "product": {
      "name": “官方AIR JORDAN 1”,
      "description": "Just do it!"
      "picture": {
        "url": "http://..."
      }
    }
  }
}
```

BAO ZUN

GraphQL



BAO ZUN

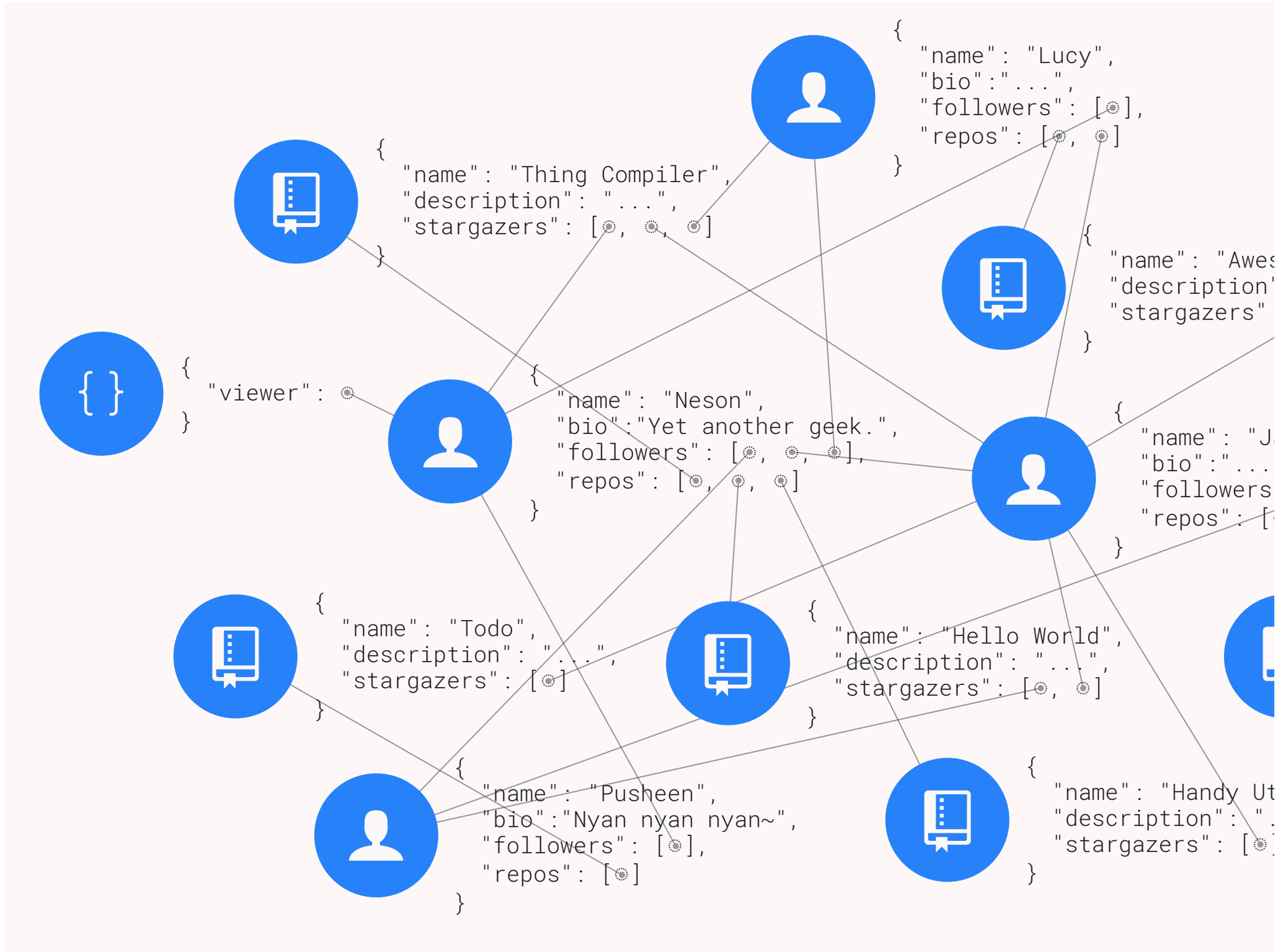
Graphql - response 结构

```
query products {
  product1: product(id: 1) {
    name
    price
  }
  product2: product(id: 2) {
    description
  }
  categories: {
    categories {
      name
    }
  }
}
```

```
type Product {
  id: String!
  name: String!
  description: String
  price: Float
}
```

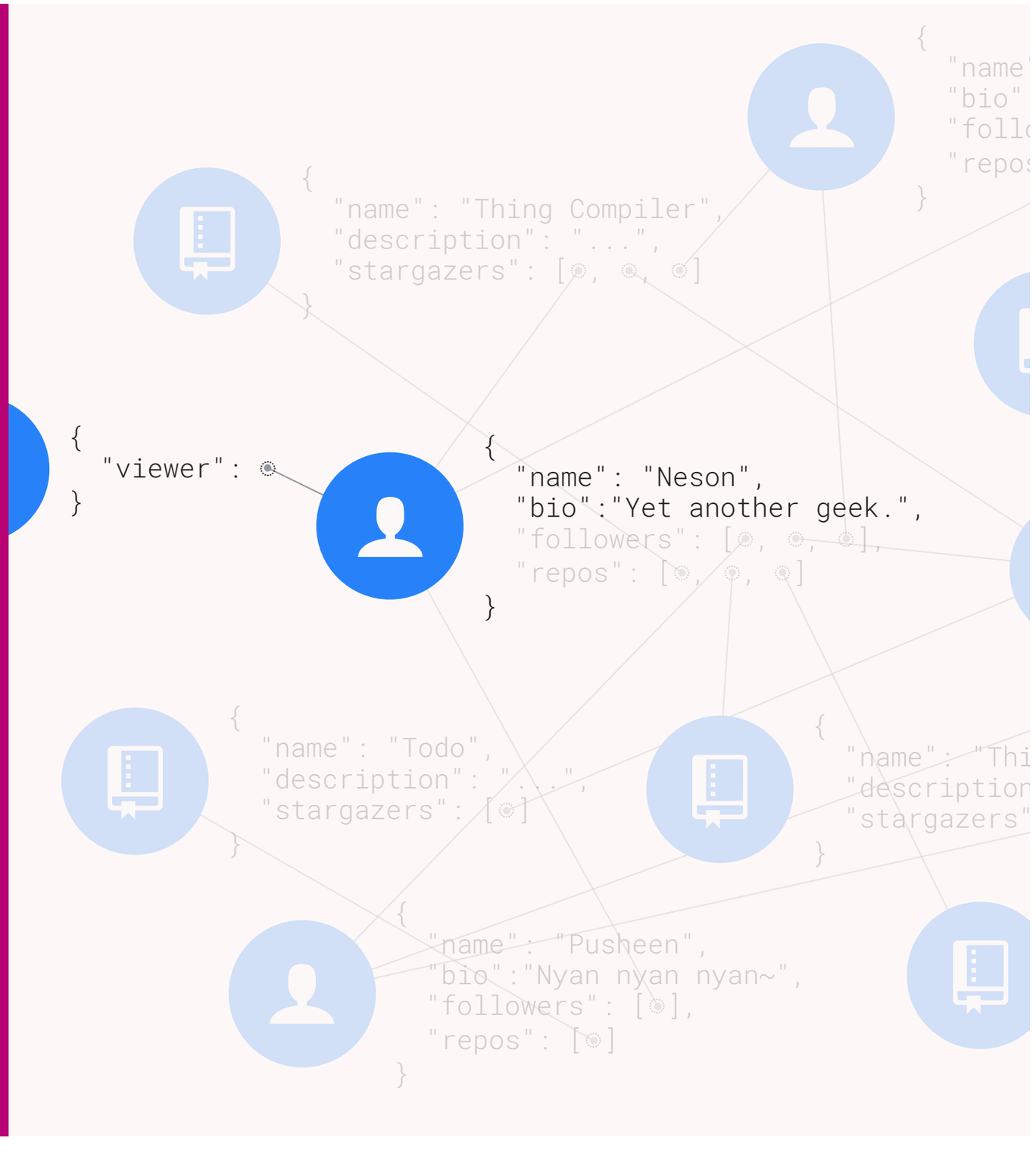
```
type Category {
  id: String!
  name: String!
  products: [Product!]
}
```

Graphql - type system



Graphql - type system

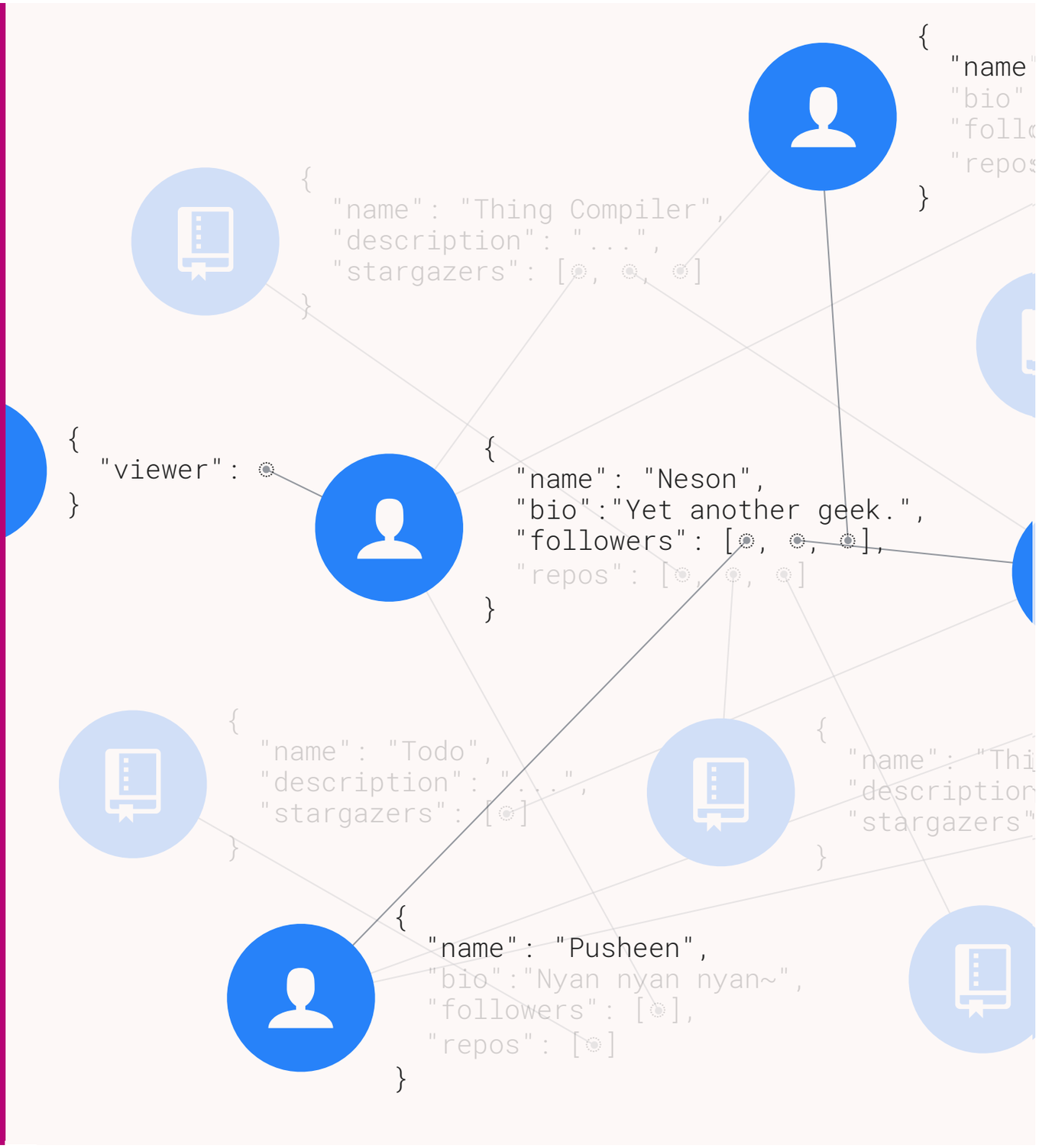
```
query {
  viewer {
    name
    bio
  }
}
```



BAO ZUN

Graphql - type system

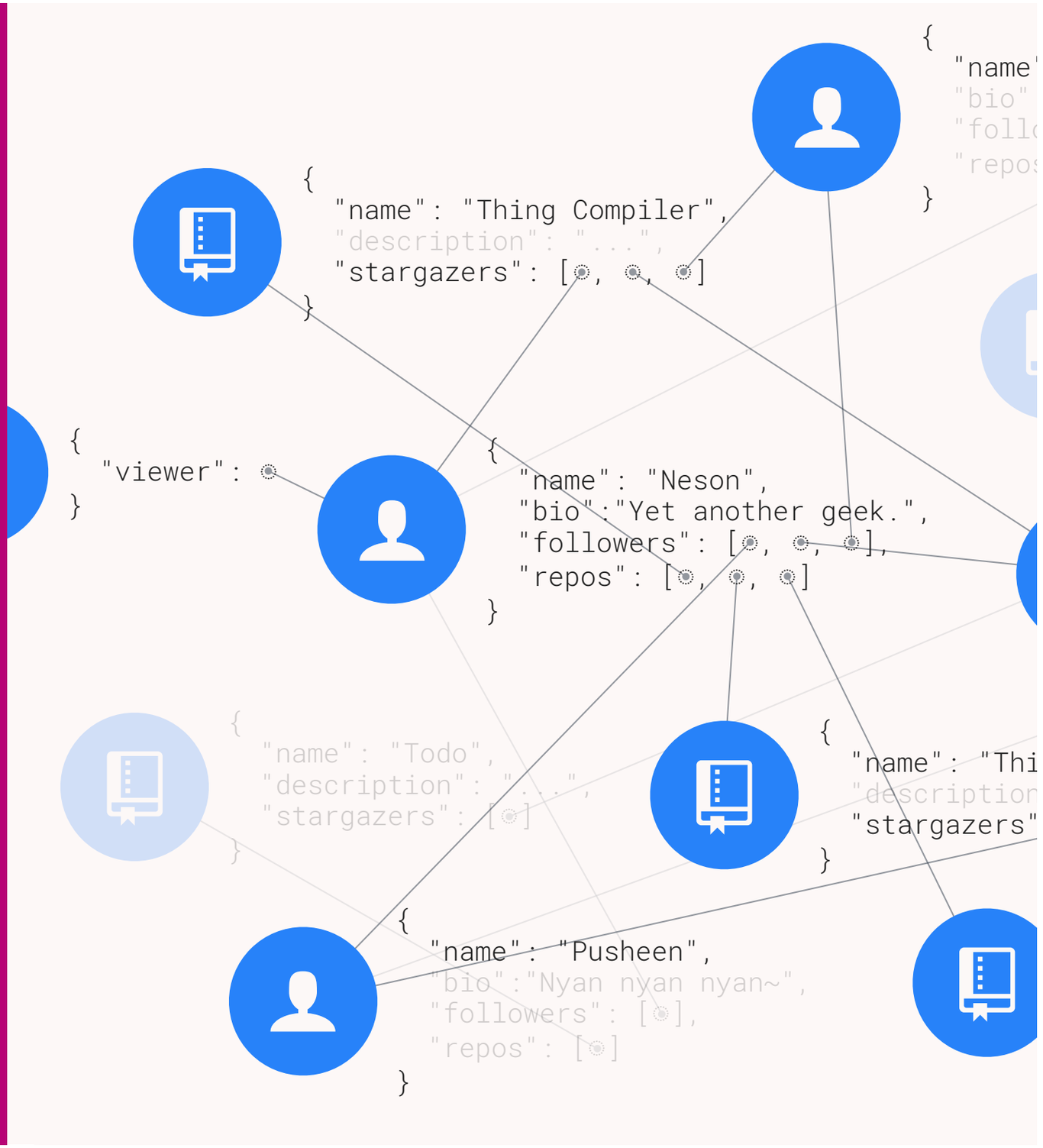
```
query {
  viewer {
    name
    bio
    followers {
      name
    }
  }
}
```



BAO ZUN

Graphql - type system

```
query {
  viewer {
    name
    bio
    followers {
      name
    }
    repos {
      name
      stargazers {
        name
      }
    }
  }
}
```



BAO ZUN

Graphql - type system

```
query {
  viewer {
    name
    bio
    followers {
      name
    }
    repos {
      name
      stargazers {
        name
      }
    }
    girlfriend {
      name
    }
  }
}
```


BAO ZUN

Graphql - type system

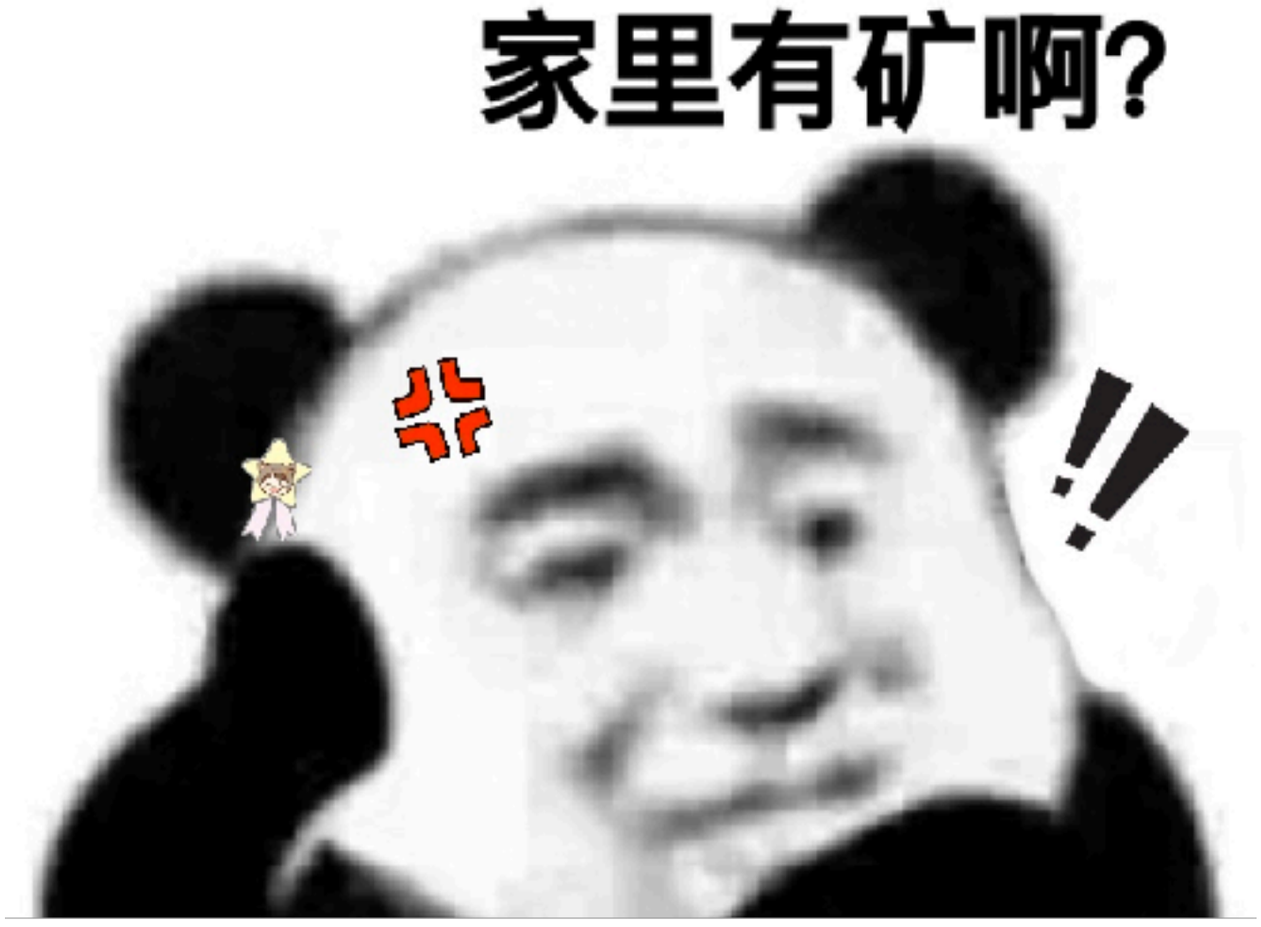
```
query {
  viewer {
    name
    bio
    followers {
      name
    }
    repos {
      name
      stargazers {
        name
      }
    }
    girlfriend {
      name
    }
  }
}
```

```
{
  "errors": [
    {
      "message": "Field 'girlfriend' doesn't
                  exist on type ‘User’",
      ...
    }
  ]
}
```

BAO ZUN

我们为什么要用

问题



人，时间，资源是有限的


CapitalOne
IBM
KLM
facebook
The New York Times
KHANACADEMY
yelp
GitHub
coursera
twitch
shopify
twitter

Graphql

大家都在用

BAO ZUN

Graphql

awesome-graphql

我们怎样使用

GraphQL

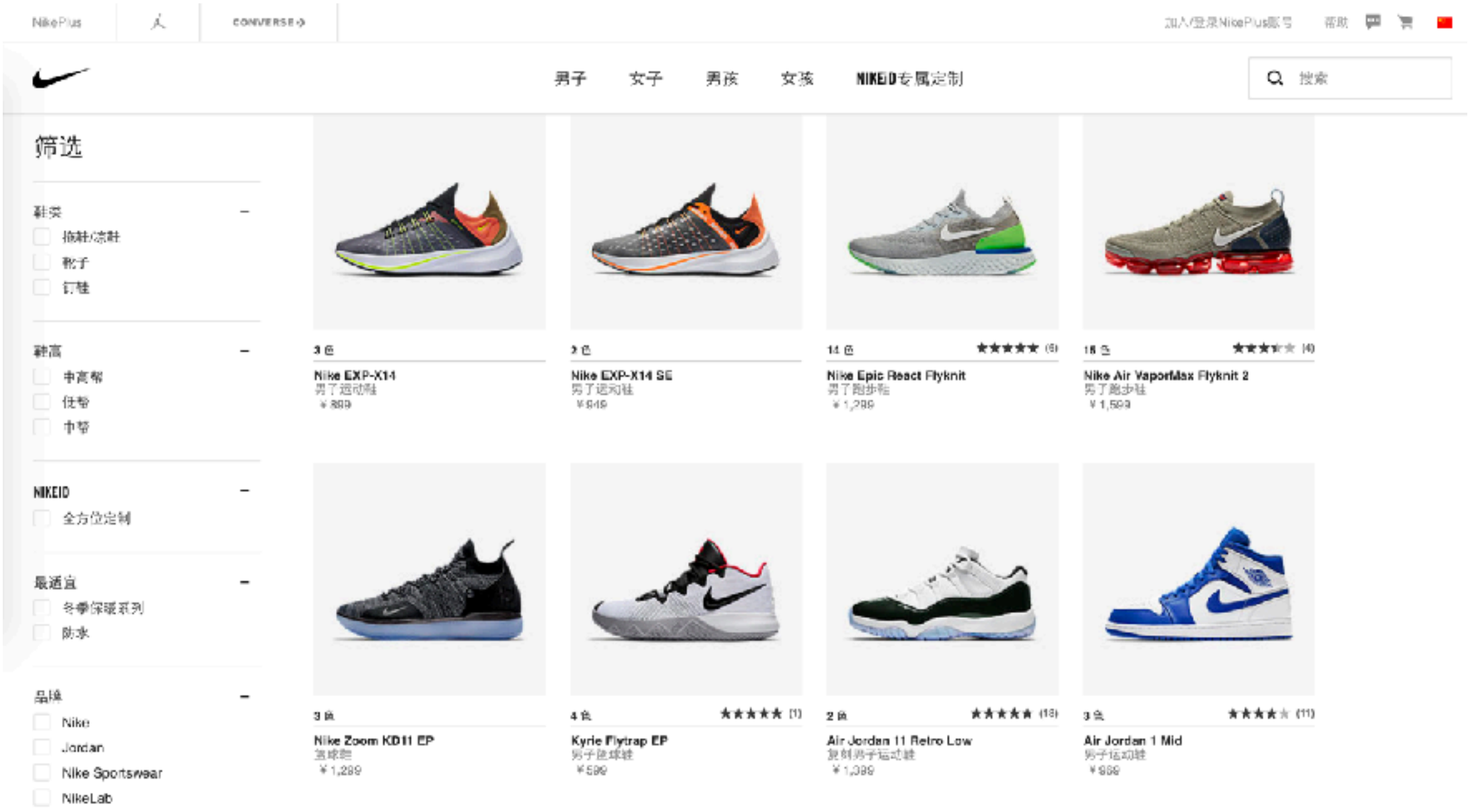
构建前后端通用语言

Schema First Development

BAO ZUN

Graphql - type system

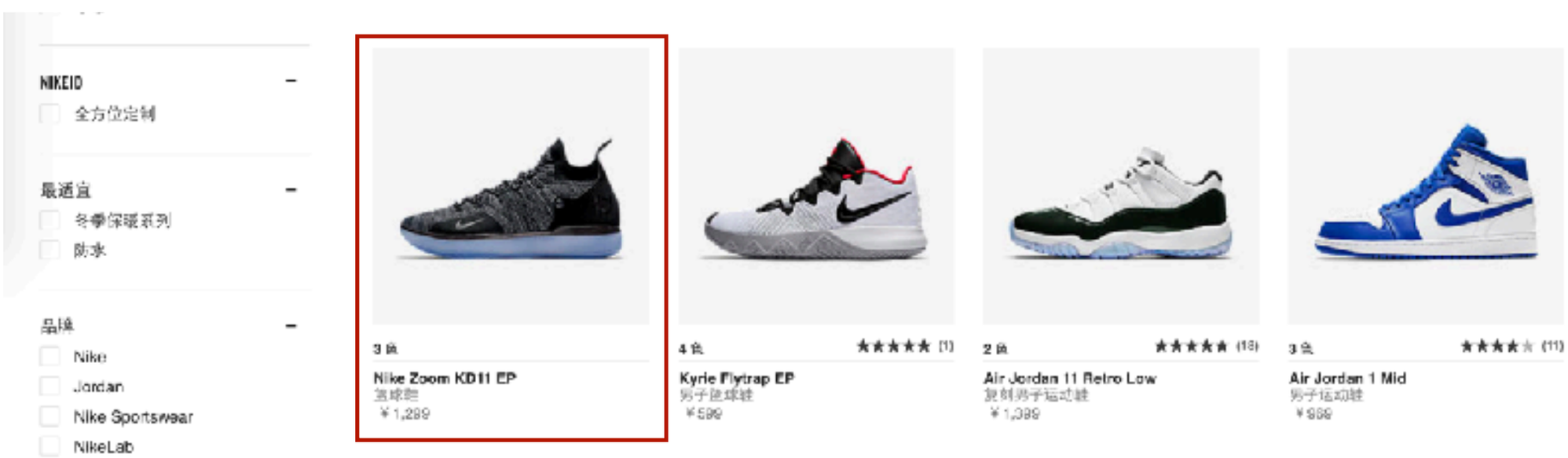
Graphql - type system



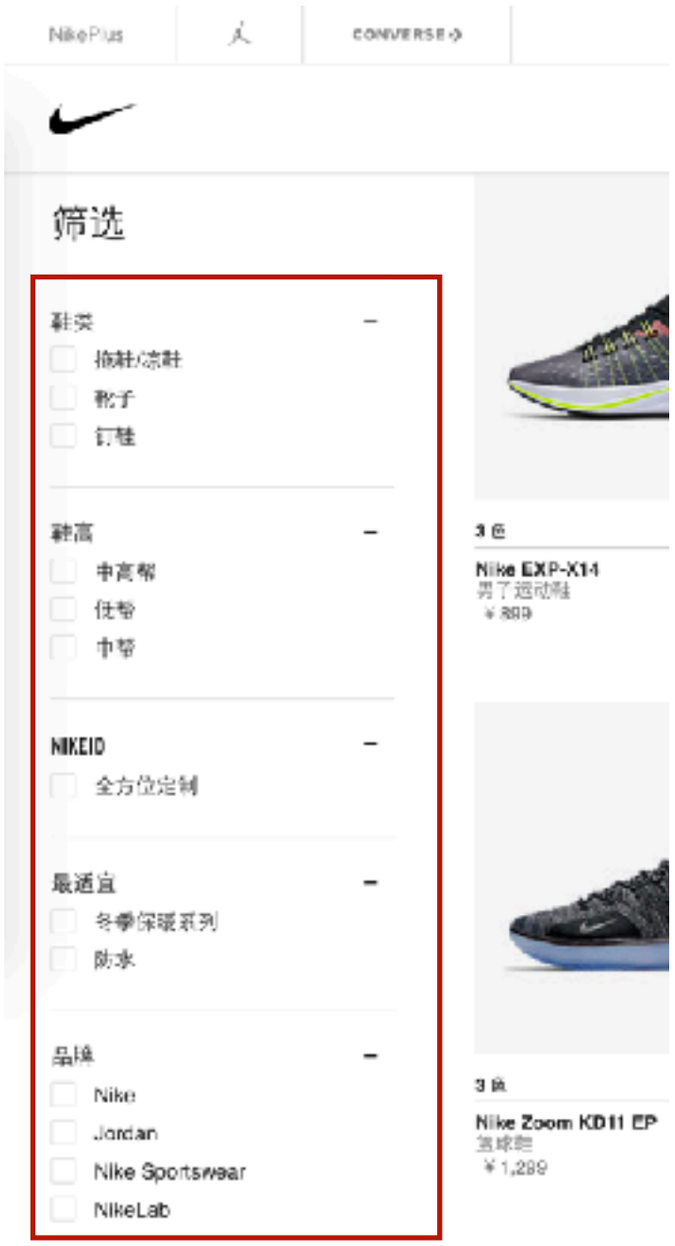
Graphql - type system

```
# 商品信息
type Product {
  id: ID!
  name: String!
  description: String
  image: String
  price: Float!
  rating: Rating
}
```

```
# 商品评分
type Rating {
  count: Int!
  average: Float!
}
```



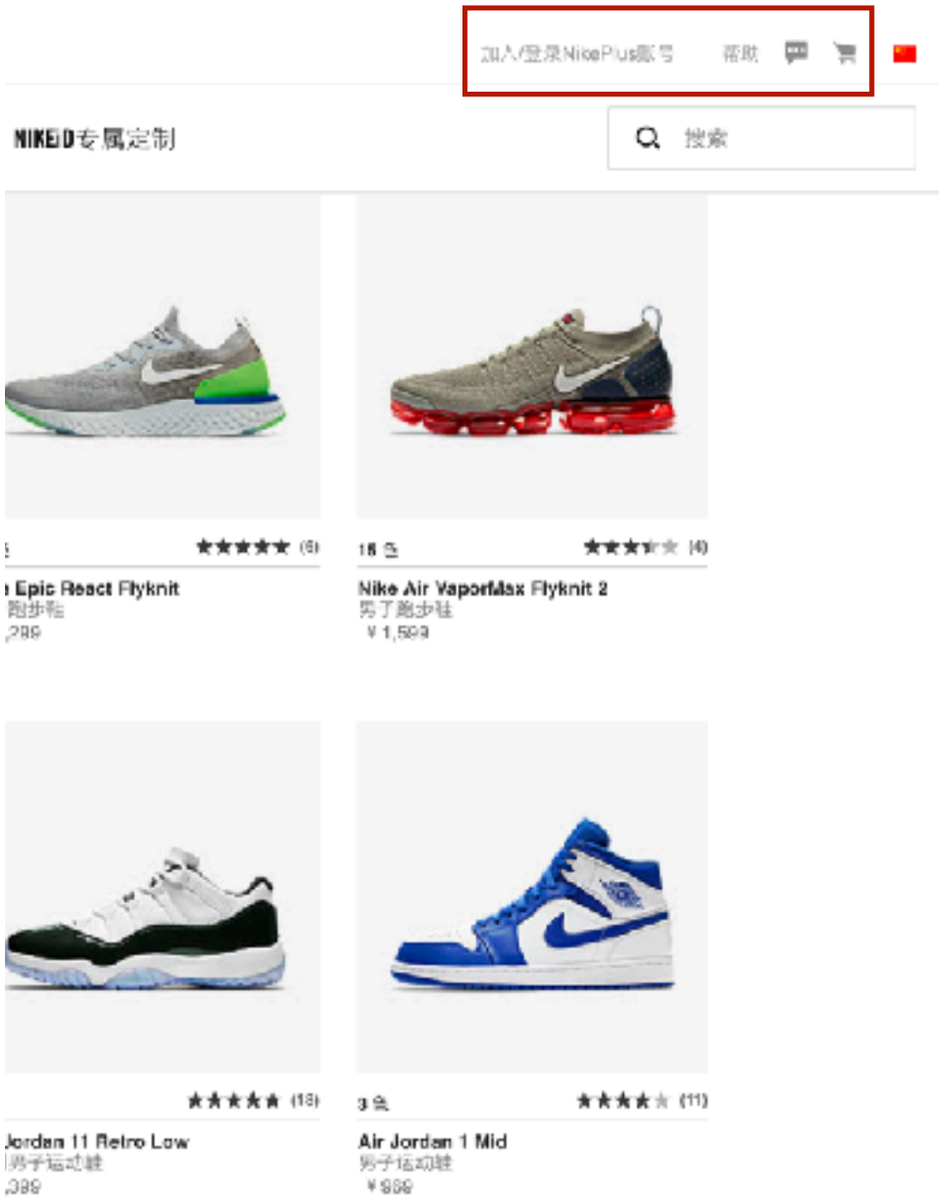
Graphql - type system




```
# 商品分类
type Category {
  # 分类名字
  name: String!
  # 子分类
  subCategory: [Category!]
  # 商品
  products:  [Product!]
}
```

Graphql - type system

```
# 用户信息
type User {
  name: String!
  cart: Cart!
}
# 购物车
type Cart {
  owner: User
  cartItem: [CartItem!]!
}
# 购物车项
type CartItem {
  product: Product!
  # 数量
  amount: Int!
}
```

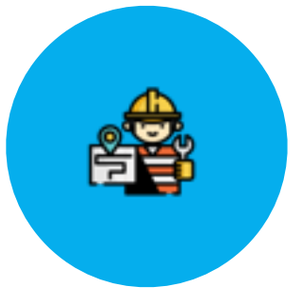


Graphql - type system

```
# 商品信息
type Product {
  id: ID!
  name: String!
  description: String
  image: String
  price: Float!
  rating: Rating
}
```

前端

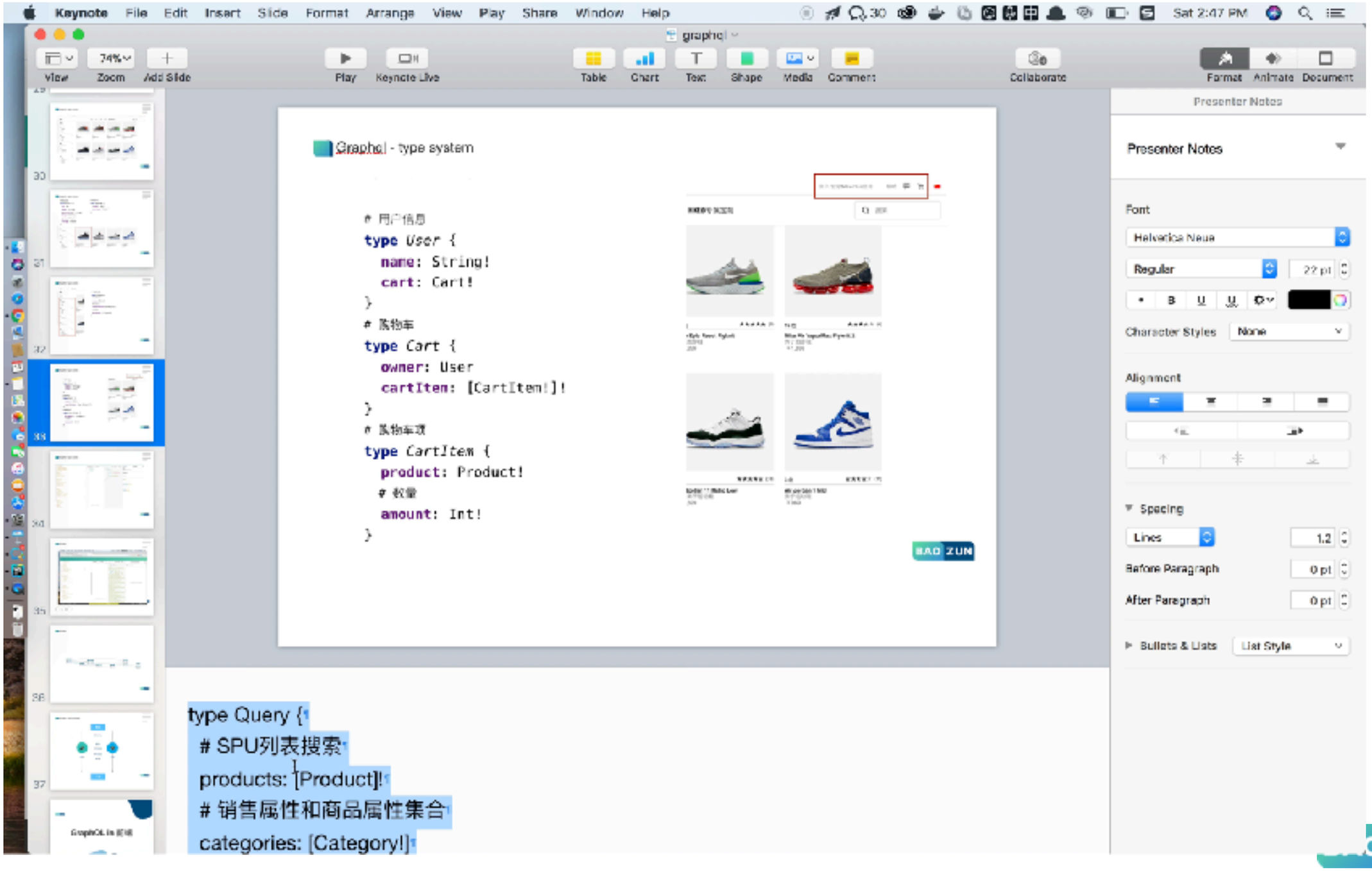
API



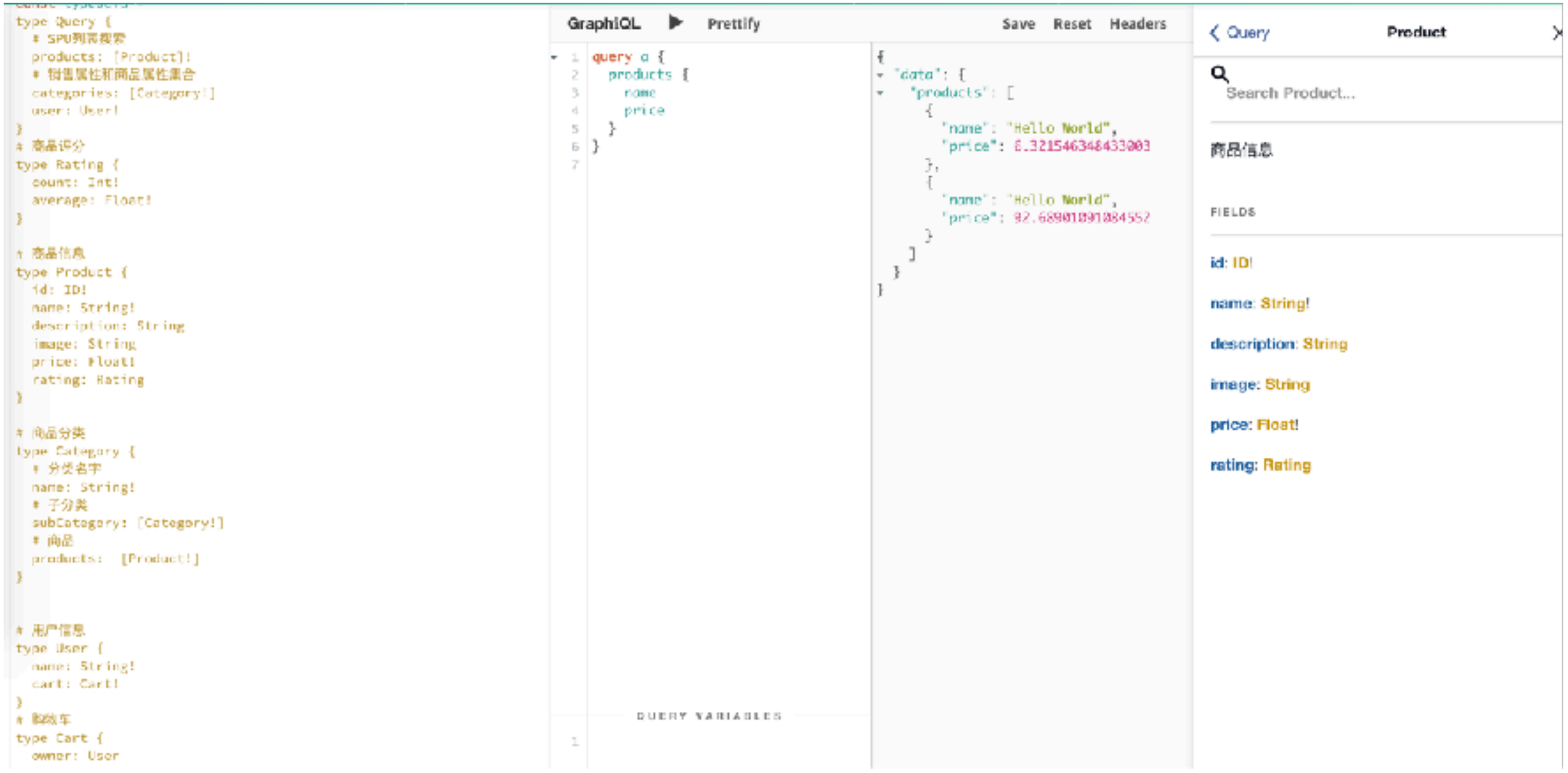
后端

resolver

Graphql


Graphql - type system
# 用户信息
type User {
name: String!
cart: Cart!
}
# 购物车
type Cart {
owner: User
cartItem: [CartItem!]!
}
# 购物车项
type CartItem {
product: Product!
数量
amount: Int!
}
type Query {
# SPU列表搜索
products: [Product]!
销售属性和商品属性集合
categories: [Category!]
Presenter Notes
Font
Helvetica Neue
Regular
Character Styles
None
Alignment
Spacing
Lines
1.2
Before Paragraph
0 pt
After Paragraph
0 pt
Bullets & Lists
List Style

Graphql - type system

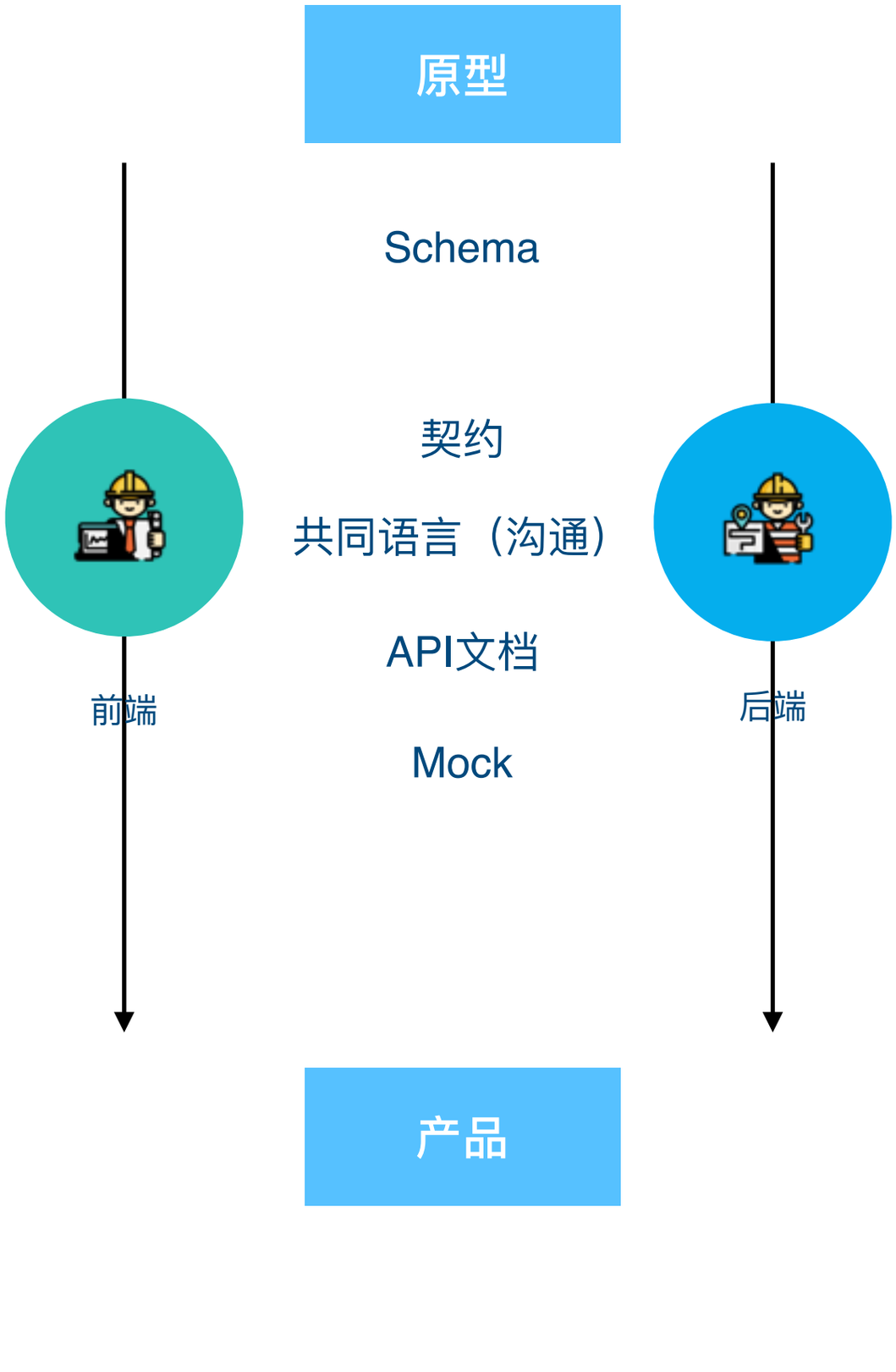


BAO ZUN

Graphql

BAO ZUN

Graphql - response 结构

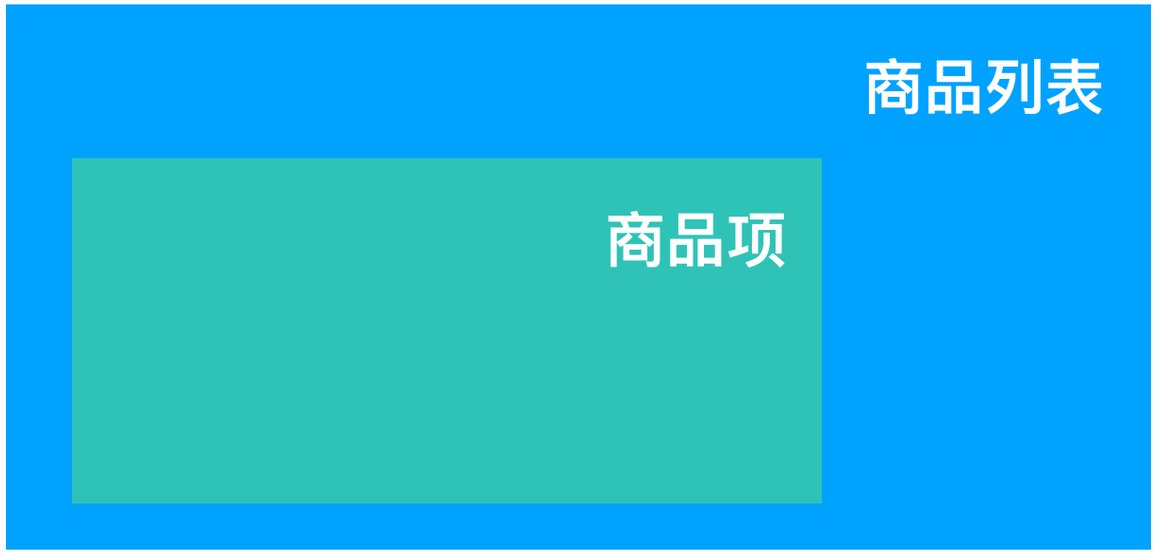


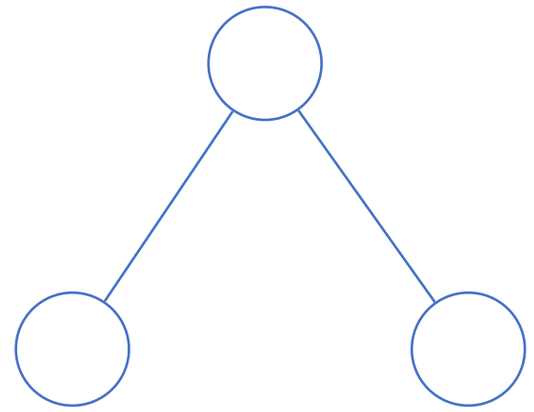
共同目标 + 相互理解 + 没有埋怨 + 相互独立

BAO ZUN

GraphQL in 前端

- **Relay**
- **Lokka**
- **Apollo**
- **....**
- **http post**

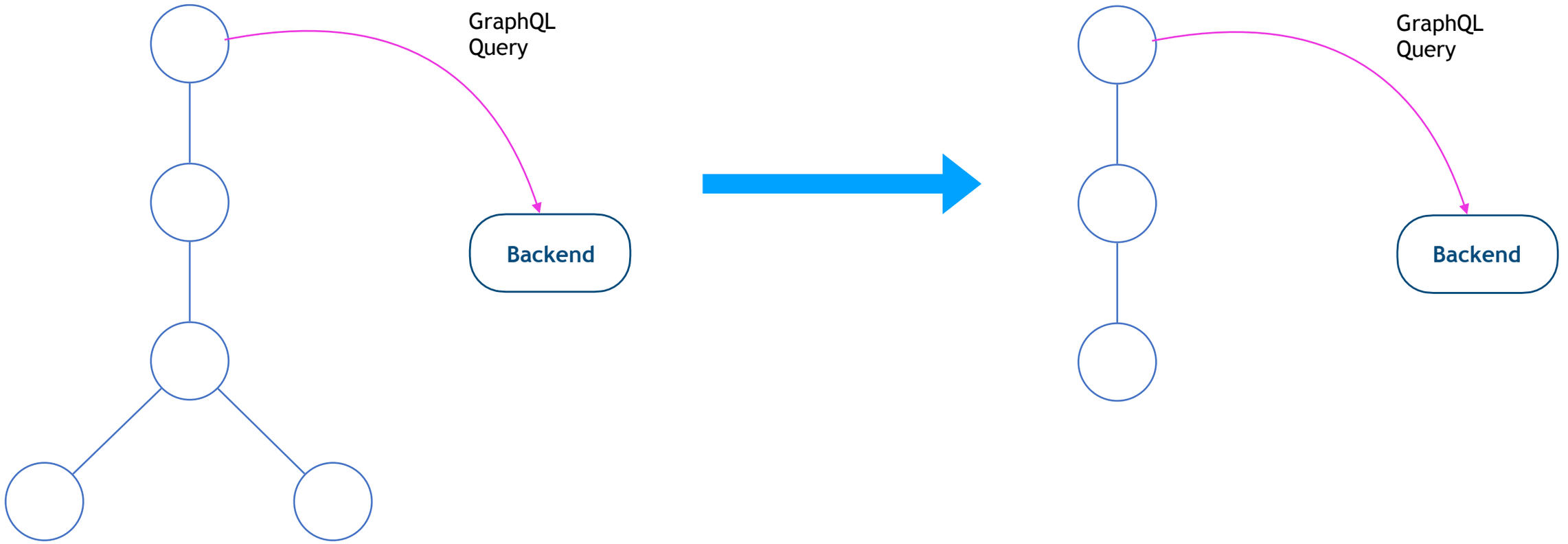




```
const Product = ({ product }) => (
  <div>
    <h1>{product.name}</h1>
  </div>
)

Product.fragment = gql`
  fragment _ on Product {
    name
    description
  }
`
```

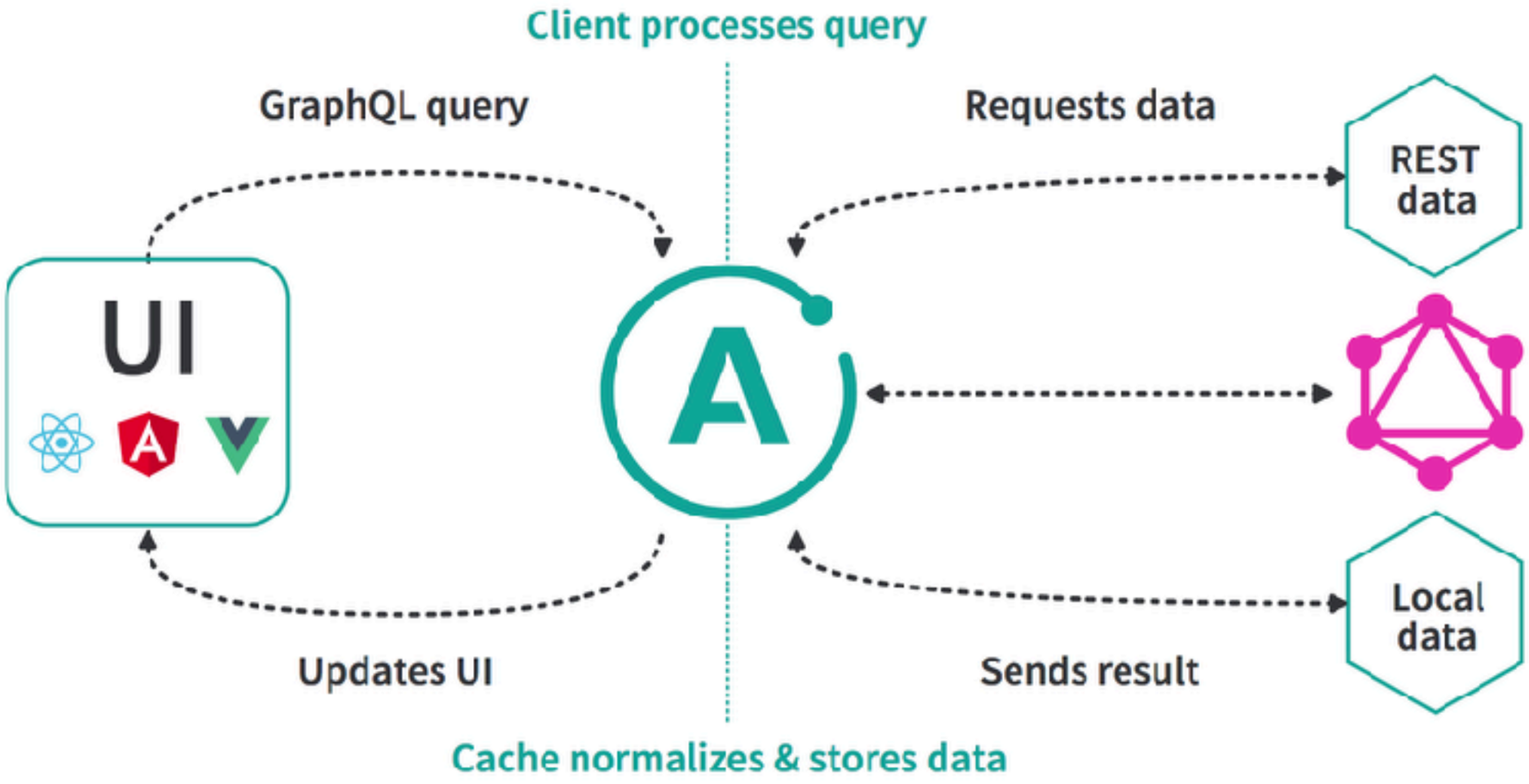
```
const ProductList = ({ products }) => (
  <div>{
    products.map(product =>{
      return <Product product={product} />
    })
  }</div>
)

ProductList.fragment = gql`
  fragment _ on ProductEdge {
    node {
      ${Product.fragment}
    }
  }
`
```

Graphql in 前端



BAO ZUN

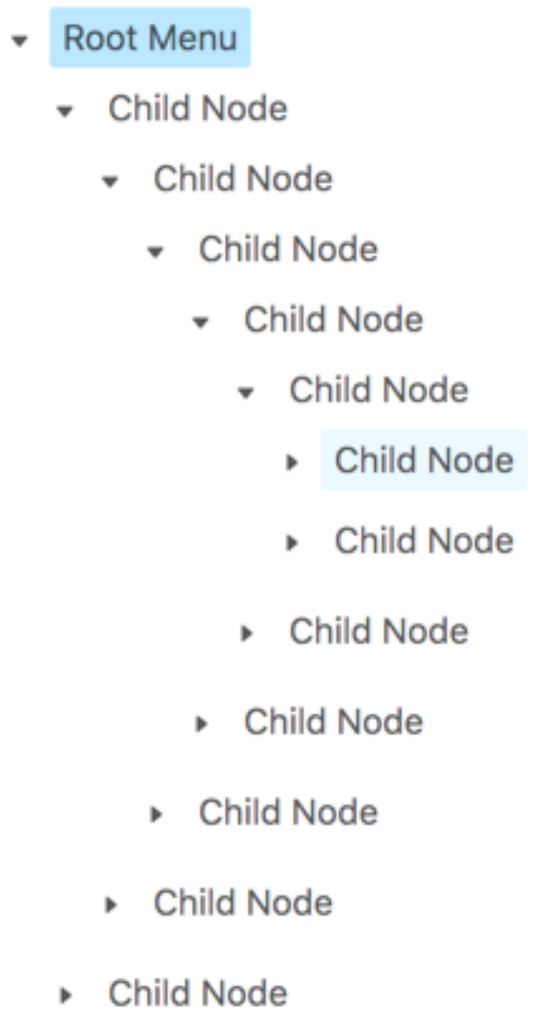
问题

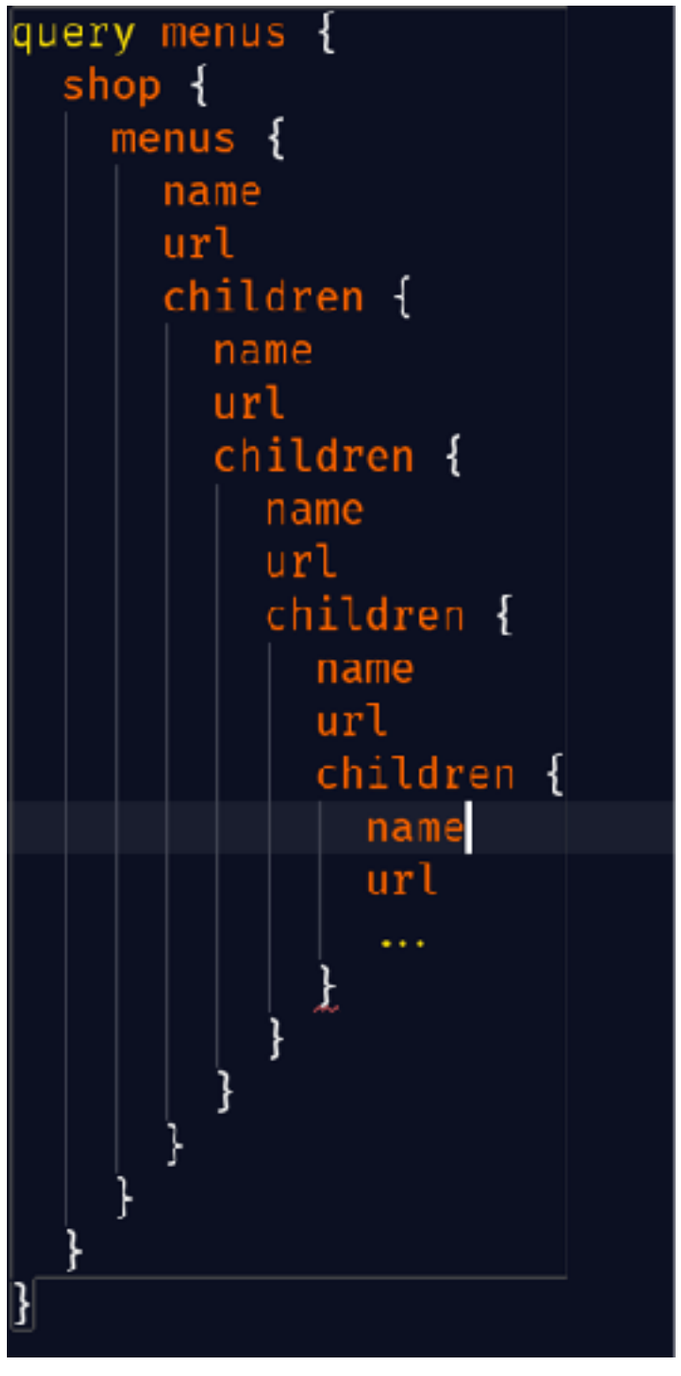

Client processes query
GraphQL query
Requests data
REST data
UI
Local data
Updates UI
Sends result
Cache normalizes & stores data

不是银弹

问题

One query get unknown depth tree





文件上传

BAO ZUN

True or False

GraphQL 是针对图数据库设计的么?

GraphQL

GraphQL 是针对图数据库设计的么?

不是

BAO ZUN

GraphQL 与编程语言无关的么?

GraphQL 与编程语言无关的么?

是

GraphQL 可以返回客户端希望的数据结构?

GraphQL

GraphQL 可以返回客户端希望的数据结构?

是

BAO ZUN

GraphQL

Facebook Relay 是唯一的 GraphQL 客户端么

GraphQL

Facebook Relay 是唯一的 GraphQL 客户端么

不是

BAO ZUN

GraphQL

GraphQL 当道
REST 已死？

BAO ZUN

GraphQL

GraphQL 当道
REST 已死？

你怎么看?

BAO ZUN

GraphQL

谢谢

BAO ZUN