

GraphQL + Relay + Serverless

박형식 @ 텀블벅

tumblbug



텀블벅

필요한 리소스만을 요청하고 사용하는 API 만들기





장바구니

1. 장바구니에 담긴 상품

상품	설명	가격	배송비
	[REDACTED] [REDACTED] [REDACTED]	25,000	2,500
	[REDACTED] [REDACTED] [REDACTED]	12,000	2,500

2. 배송 주소

[REDACTED] [REDACTED] [REDACTED] [REDACTED]



장바구니

Cart

1. 장바구니에 담긴 상품

ProductList

상품	설명	ProductData	가격	배송비
	[REDACTED]	[REDACTED]	25,000	2,500
	[REDACTED]	[REDACTED]	12,000	2,500

2. 배송 주소

ShippingAddress

Cart

/carts/1

ProductList

ProductData

/products/1

ProductData

/products/2

ShippingAddress

/shippingAddresses/1



/carts/1?

fields=products, coupons, shipping_addresses&products[images, descriptions]

/carts/1?

fields=products, coupons, shipping_addresses&products[images[high, low]]&products[descriptions[reviews, contents]]

/carts_with_products_for_tablets

/carts_with_products_shipping_address_coupons

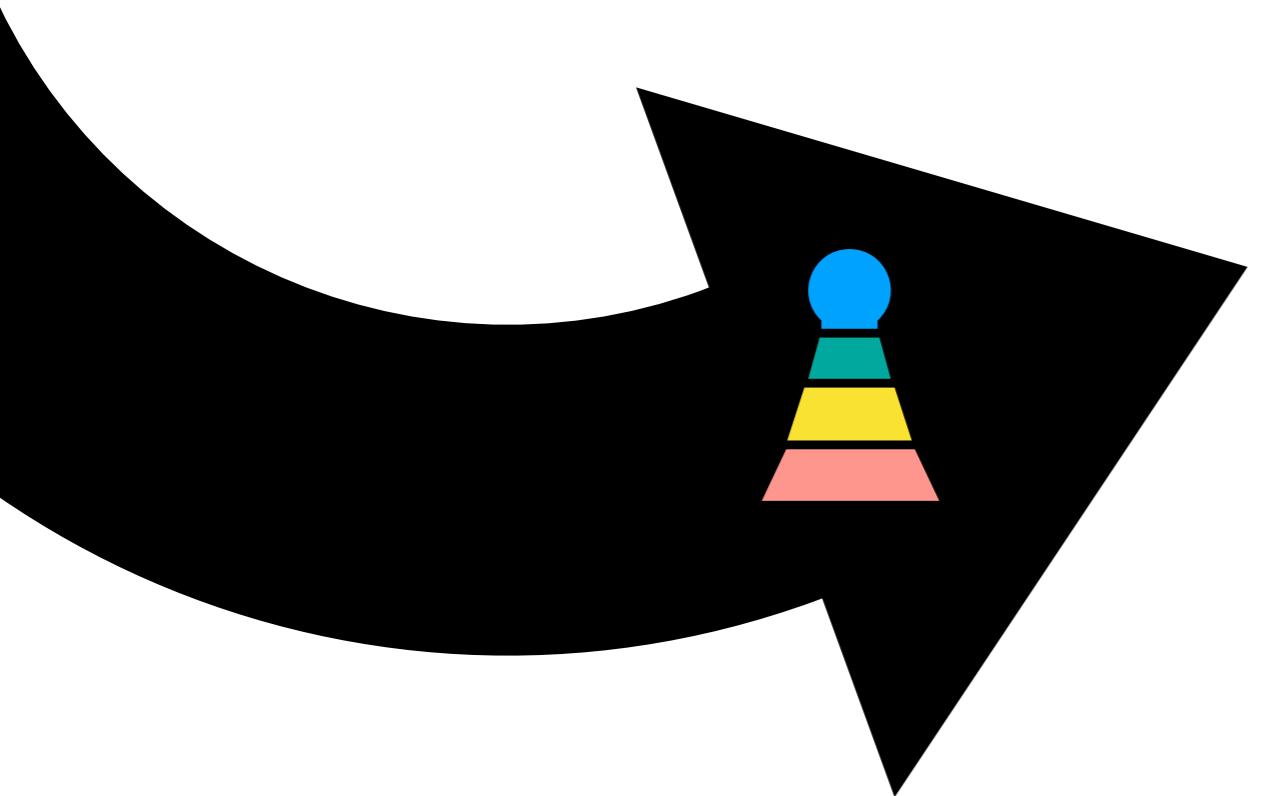
/carts_with_products_recommendations

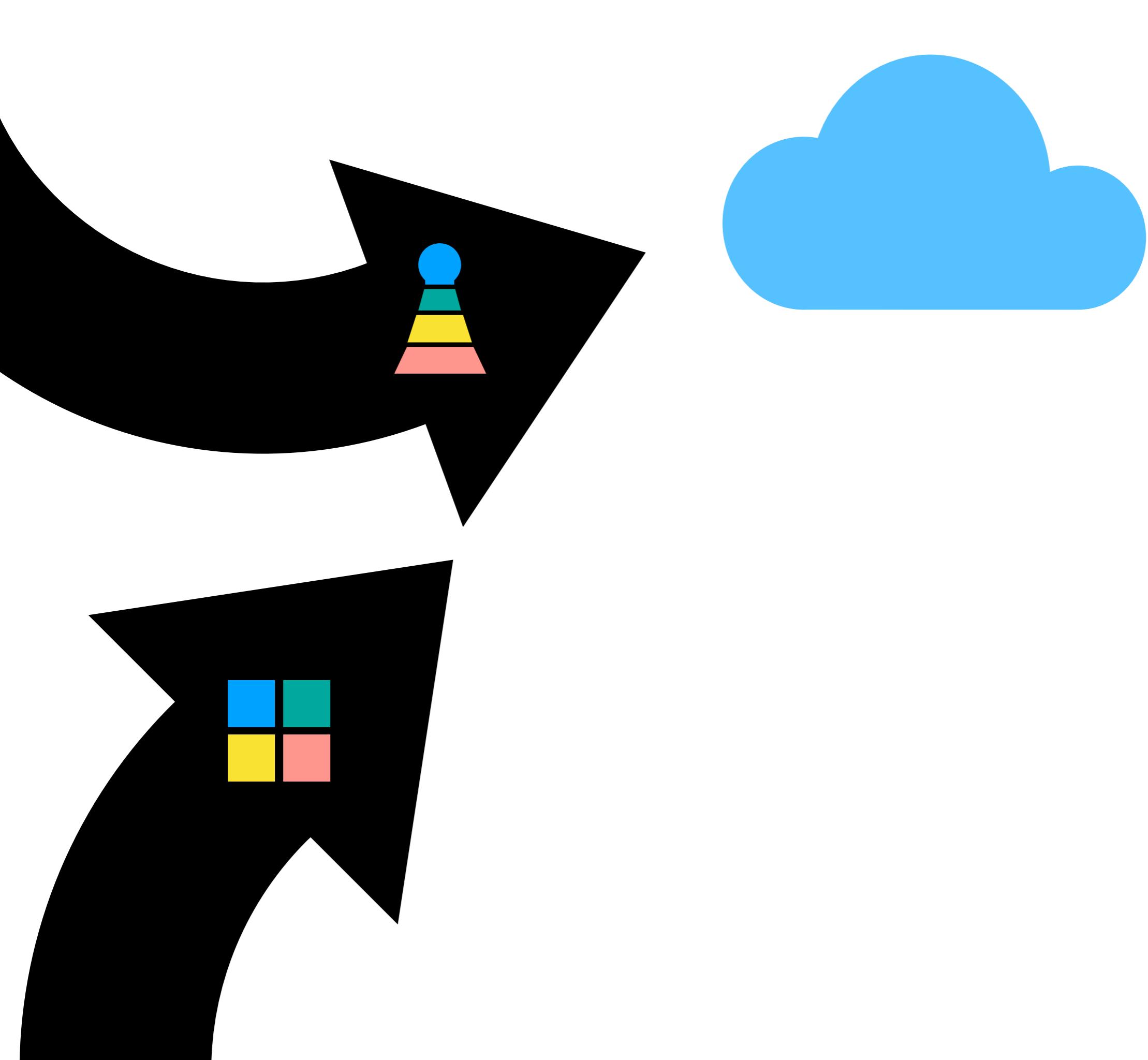
/carts_with_products_videos

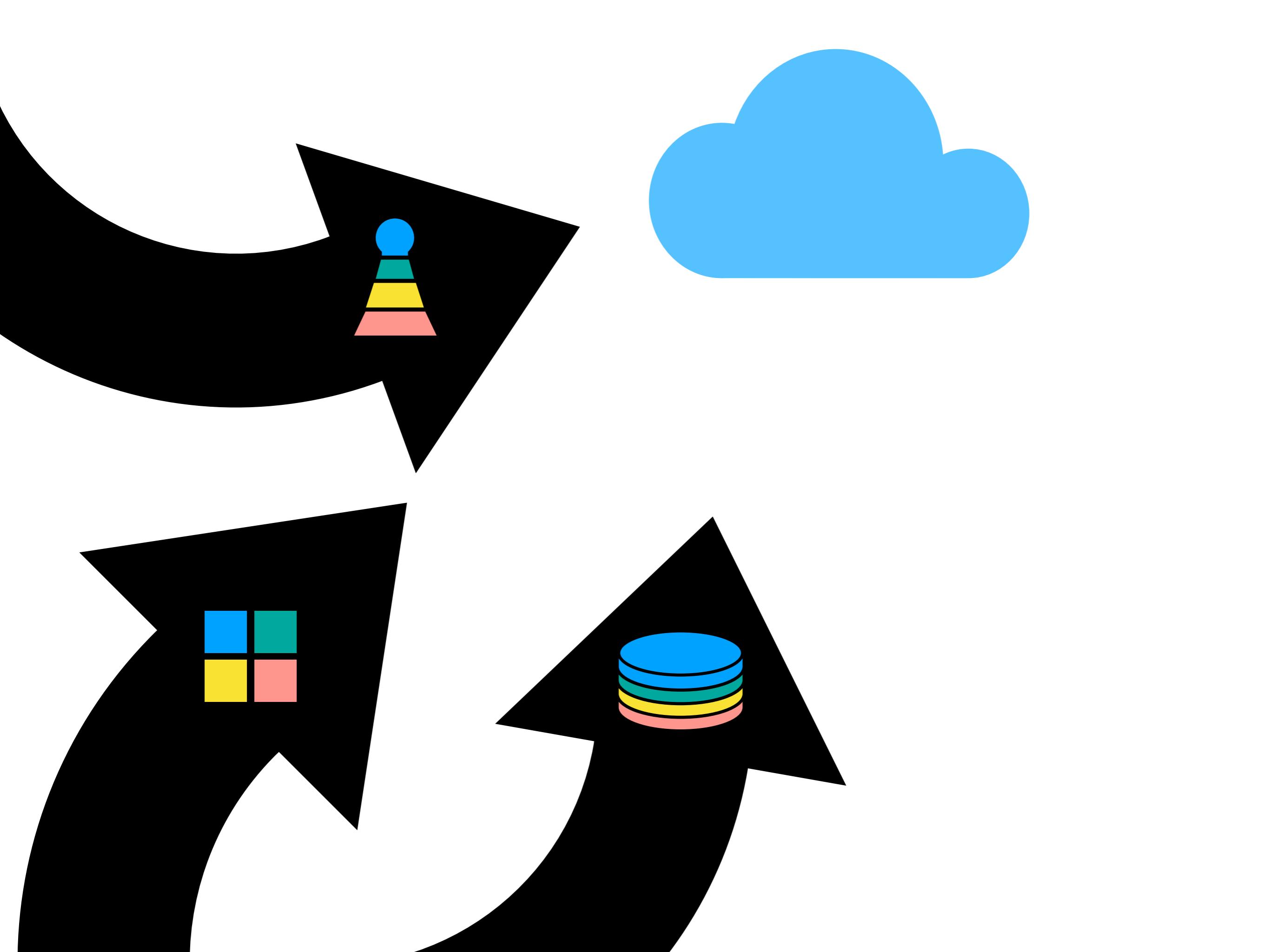
/carts_with_products_newsletter_signups

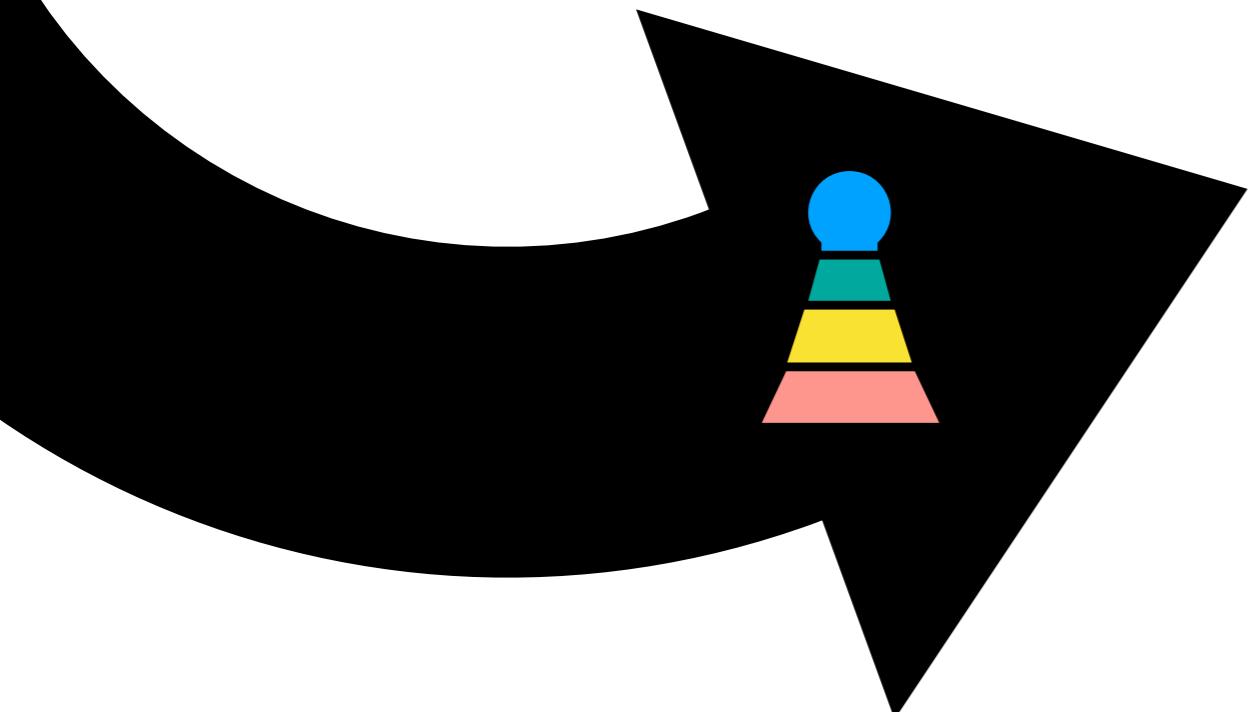
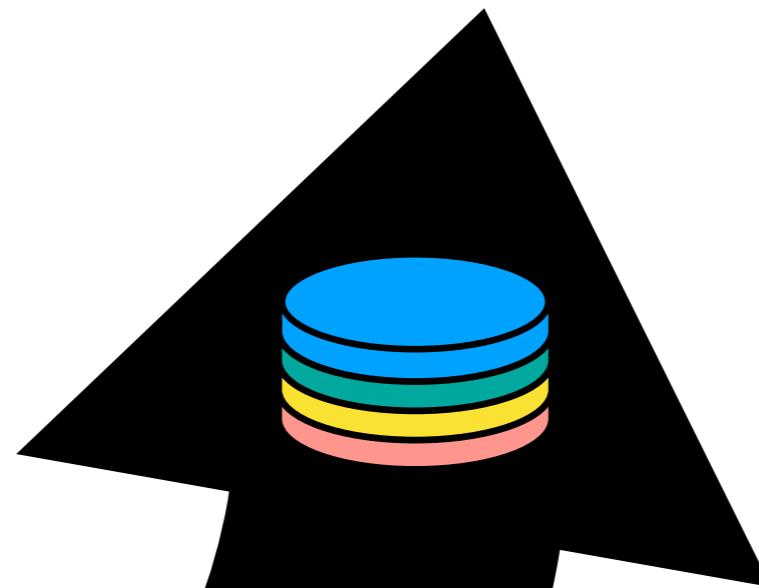
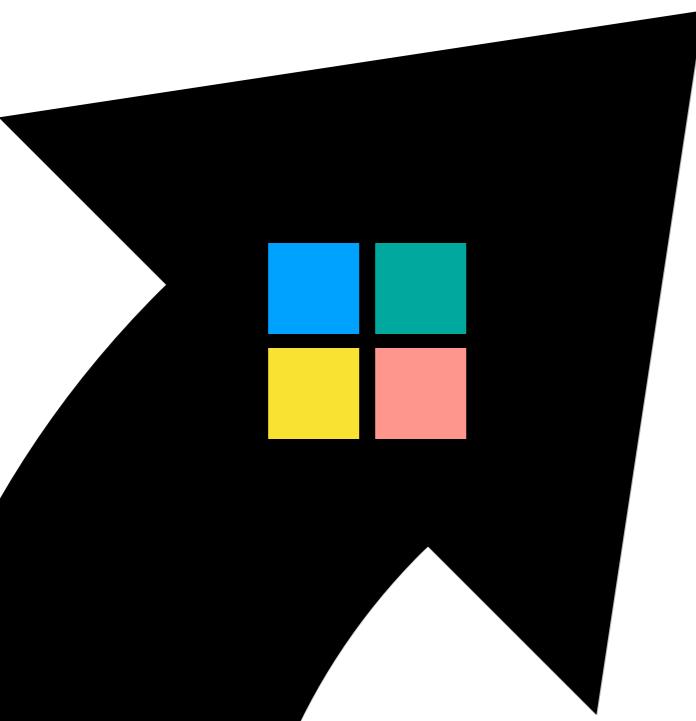
/carts_with_products_for_mobile

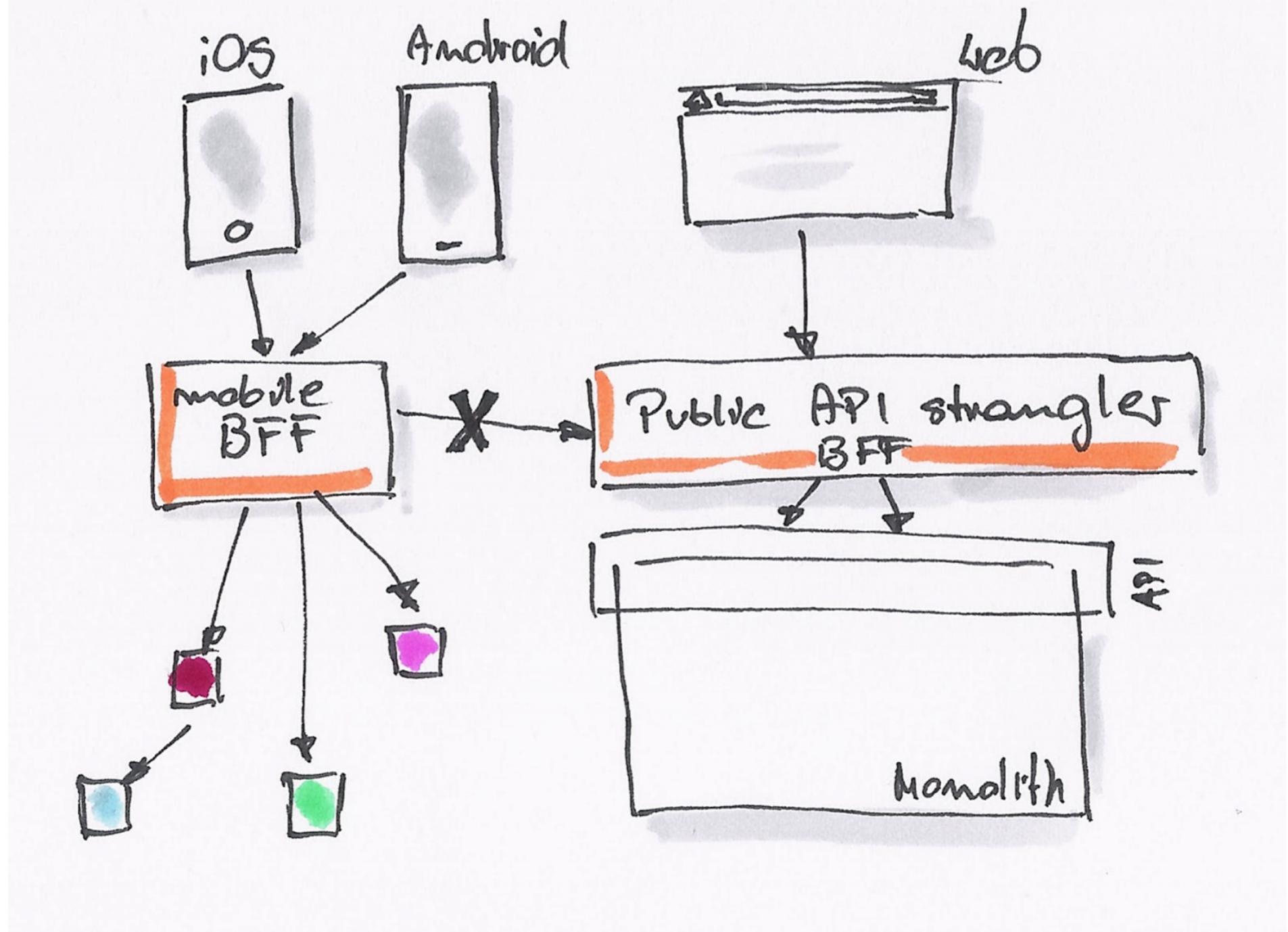
/carts_with_products_for_desktops





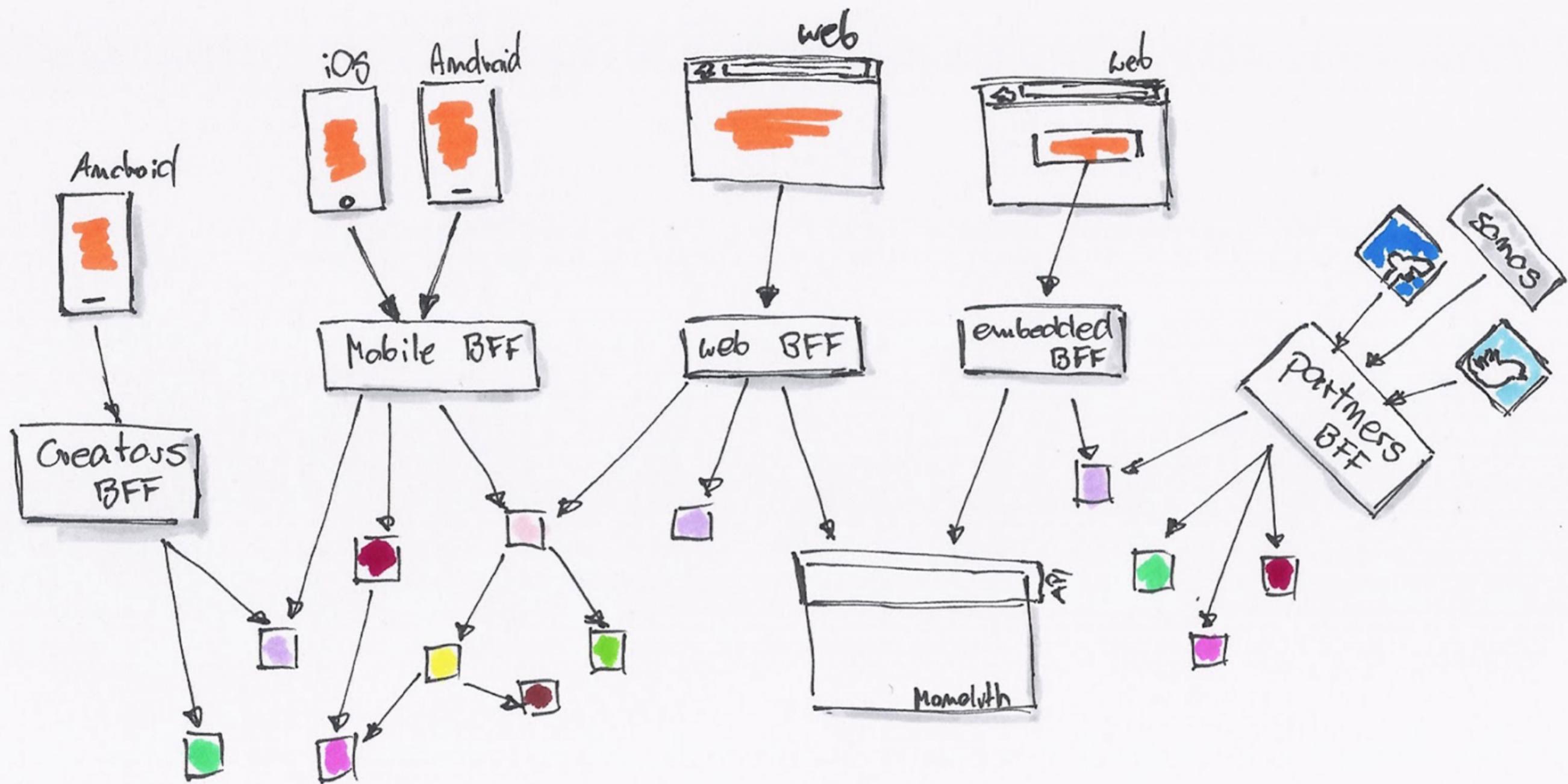






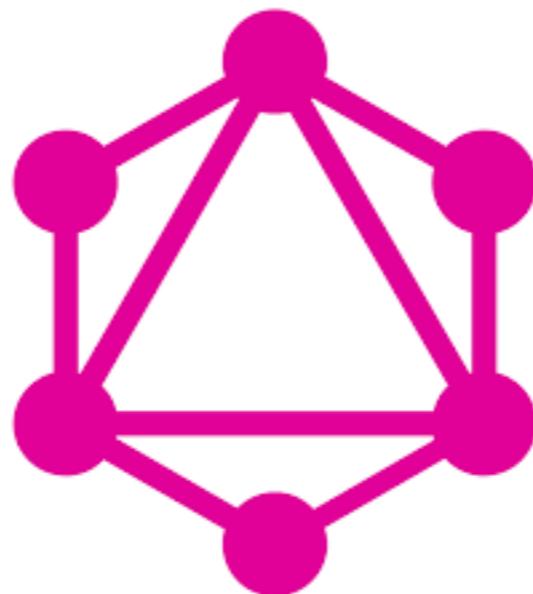
BFF (soundcloud)

Backends for Frontends



GraphQL

A Query Language for API



```
{  
  me {  
    name  
    company  
  }  
}
```

```
{
```

```
me {
```

```
    name: "박형식"
```

```
    company: "텀블벅"
```

```
}
```

```
}
```

```
{  
me {  
    name  
    company  
    hobby  
}  
}
```

```
{
```

```
me {
```

```
    name: "박형식"
```

```
    company: "텀블벅"
```

```
    hobby: "축구"
```

```
}
```

```
}
```

Query

REST GET method 

REST

GET /me

```
{  
    name: "박형식",  
    company: "텀블벅",  
    hobby: "축구",  
    favoriteColor: "파랑색",  
    country: "대한민국",  
    conference: "playnode 2017",  
    cloudProvider: "IBM",  
    age: "?^^^^^",  
    address: "?????????",  
    cardNumber: "0#0#0#0#0%#0FD0#0#0#"  
}
```

GraphQL

```
{  
  user (id: "1") {  
    name  
    company  
  }  
}
```

GraphQL

```
{  
  user {  
    name: "chulsoo"  
    company: "younghee co."  
  }  
}
```

REST

Cart

/carts/1

ProductList

ProductData

/products/1

ProductData

/products/2

ShippingAddress

/shippingAddresses/1

GraphQL

Cart

ProductList

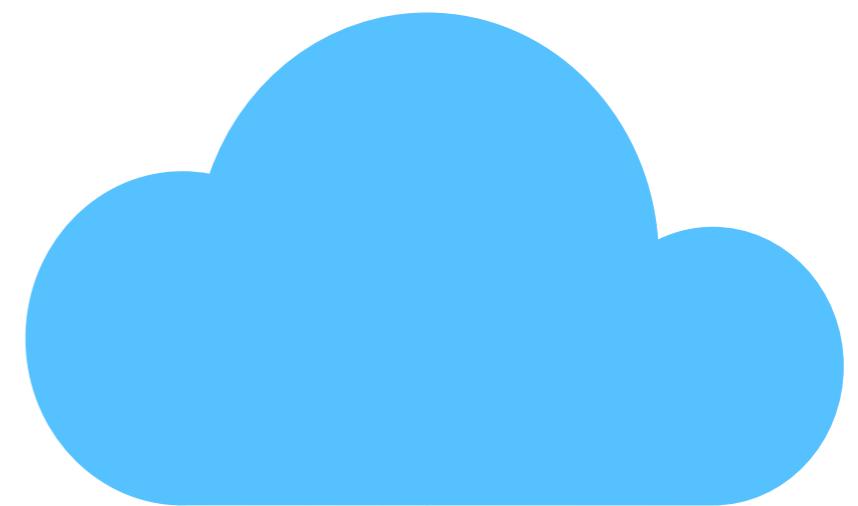
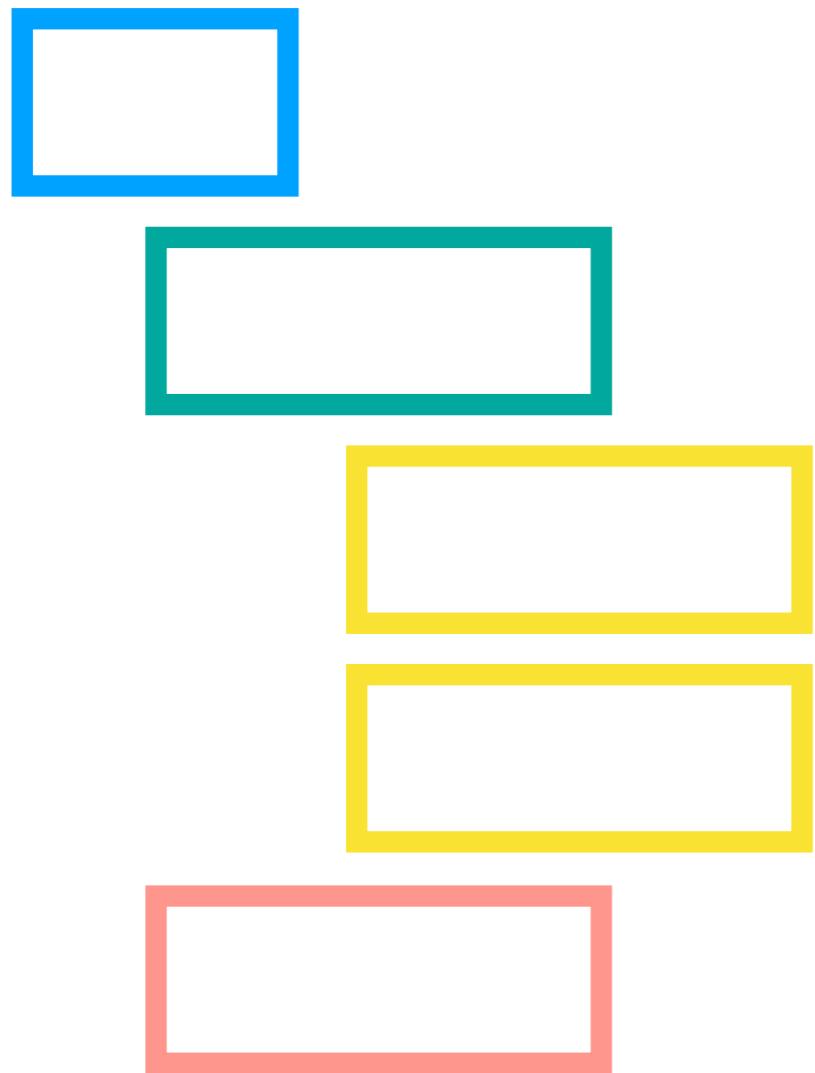
ProductData

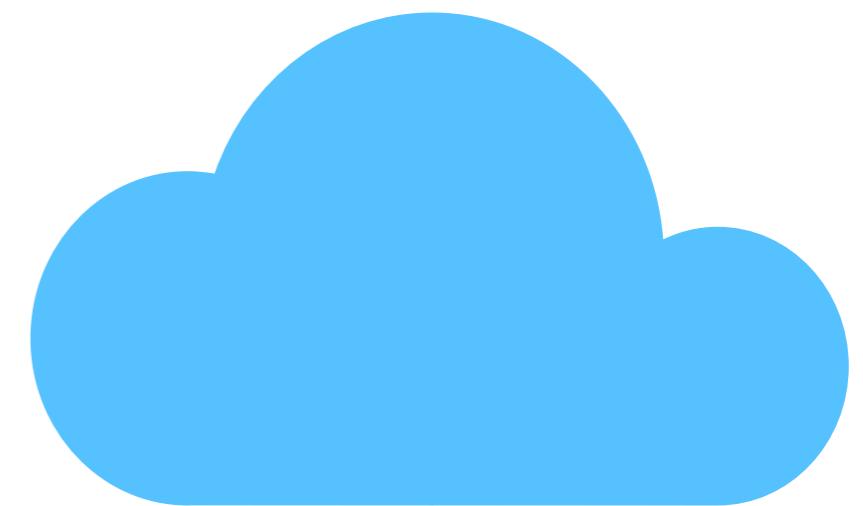
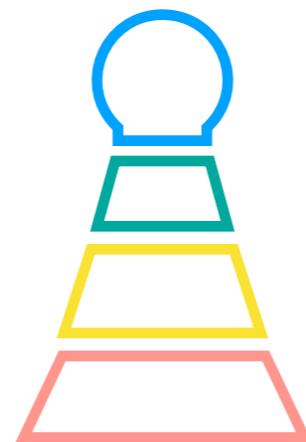
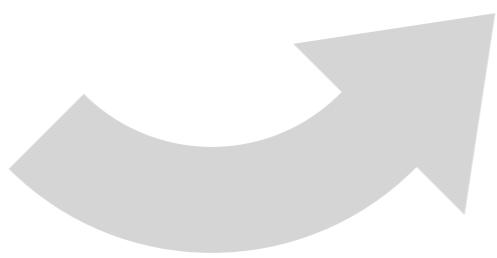
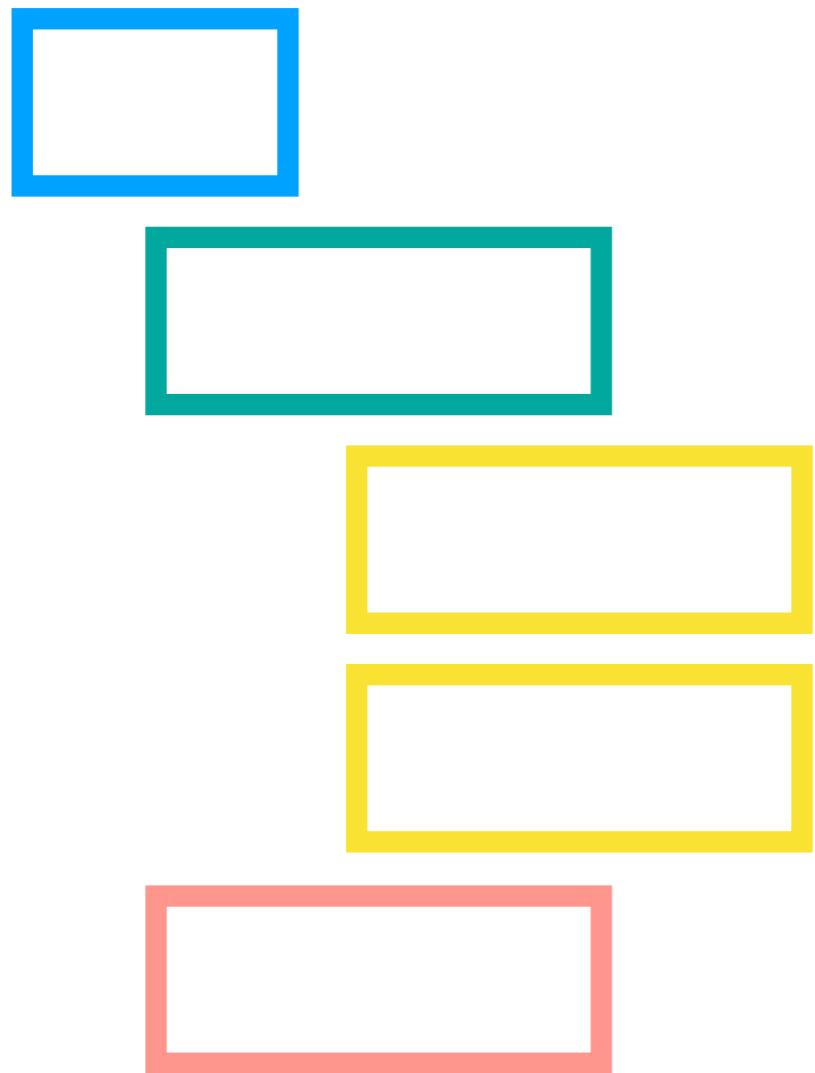
ProductData

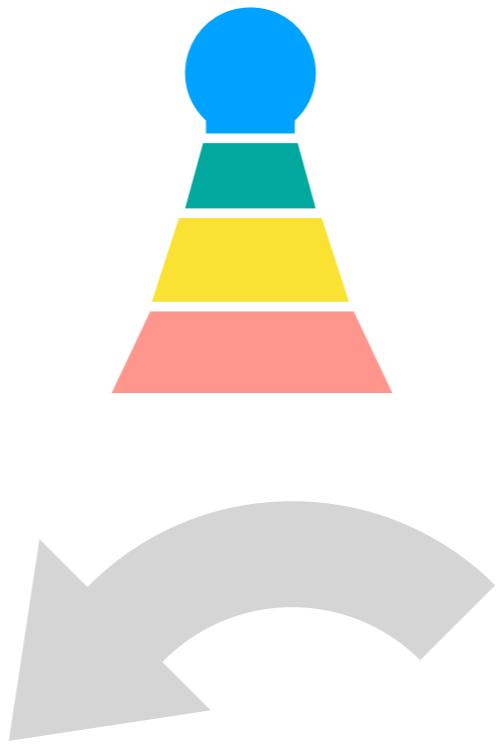
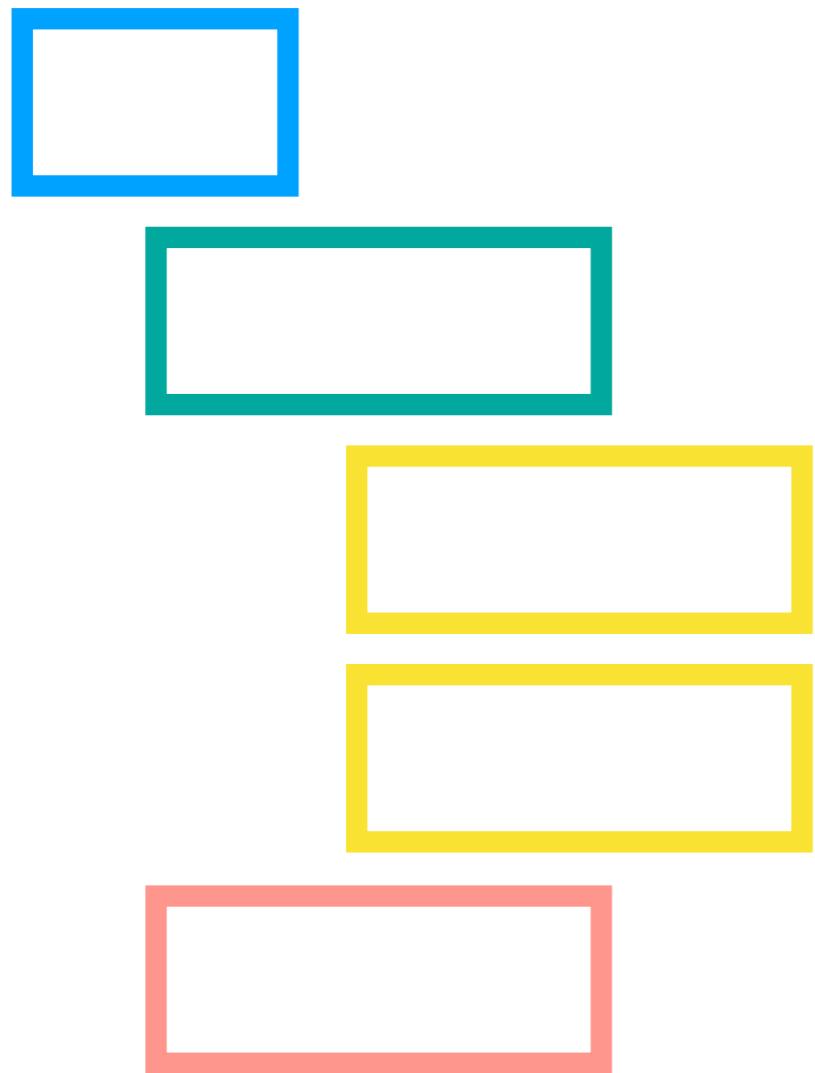
ShippingAddress

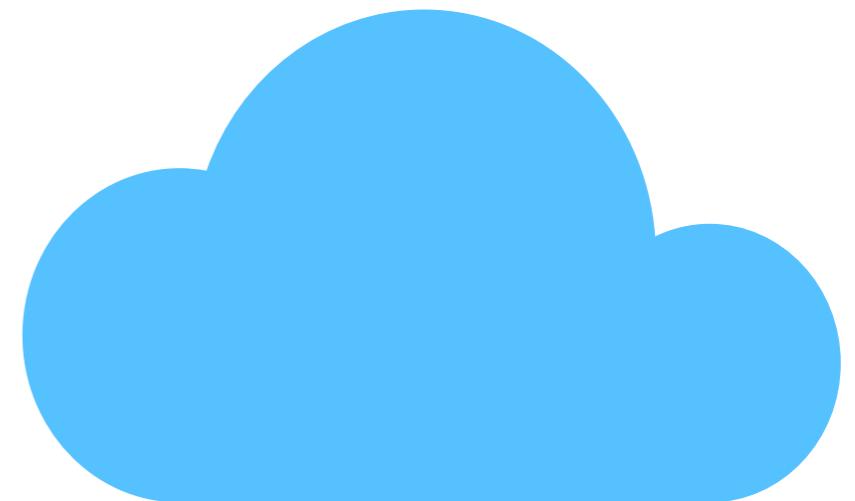
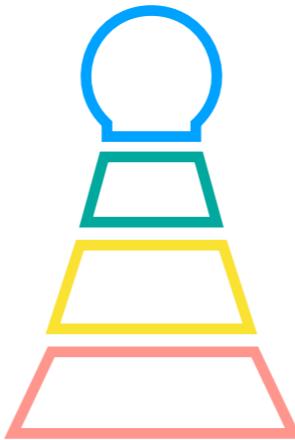
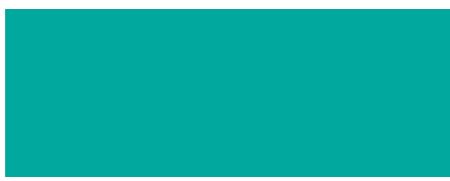
```
{  
  cart {  
    products {  
      imageUrl  
      description  
      price  
      shippingCost  
    }  
    shippingAddress  
  }  
}
```

POST /graphql









Type System



Types

```
type Query {  
    cart: Cart  
}  
  
type Cart {  
    products: [Product]!  
    shippingAddress: String!  
}  
  
type Product {  
    imageUrl: String!  
    description: String!  
    price: Int!  
    shippingCost: Int!  
}  
  
    cart {  
        products {  
            imageUrl  
            description  
            price  
            shippingCost  
        }  
        shippingAddress  
    }
```

Types

```
type Query {  
  cart: Cart  
}  
  
type Cart {  
  products: [Product]!  
  shippingAddress: String!  
}  
  
type Product {  
  imageUrl: String!  
  description: String!  
  price: Int!  
  shippingCost: Int!  
}  
  
cart {  
  products {  
    imageUrl  
    description  
    price  
    shippingCost  
  }  
  shippingAddress  
}
```

Types

```
type Query {  
    cart: Cart  
}  
  
type Cart {  
    products: [Product]!  
    shippingAddress: String!  
}  
  
type Product {  
    imageUrl: String!  
    description: String!  
    price: Int!  
    shippingCost: Int!  
}
```

```
    {  
        cart {  
            products {  
                imageUrl  
                description  
                price  
                shippingCost  
            }  
            shippingAddress  
        }  
    }  
}
```

Types

```
type Query {  
    cart: Cart  
}  
  
type Cart {  
    products: [Product]!  
    shippingAddress: String!  
}  
  
type Product {  
    imageUrl: String!  
    description: String!  
    price: Int!  
    shippingCost: Int!  
}
```

```
{  
    cart {  
        products {  
            imageUrl  
            description  
            price  
            shippingCost  
        }  
        shippingAddress  
    }  
}
```

Types

```
type Query {  
    cart: Cart  
}  
  
type Cart {  
    products: [Product]!  
    shippingAddress: String!  
}  
  
type Product {  
    imageUrl: String!  
    description: String!  
    price: Int!  
    shippingCost: Int!  
}
```

```
{  
    cart {  
        products {  
            imageUrl  
            description  
            price  
            shippingCost  
        }  
        shippingAddress  
    }  
}
```

Mutation

REST POST/PUT/PATCH/DELETE Method



Mutation

```
mutation {  
  createProduct(  
    input: {  
      description: "멋진 옷"  
      price: 50000  
    }  
  )  
} {  
  product {  
    Price  
  }  
}
```



더도 말고 덜도 말고
딱 클라이언트가 원하는 값만

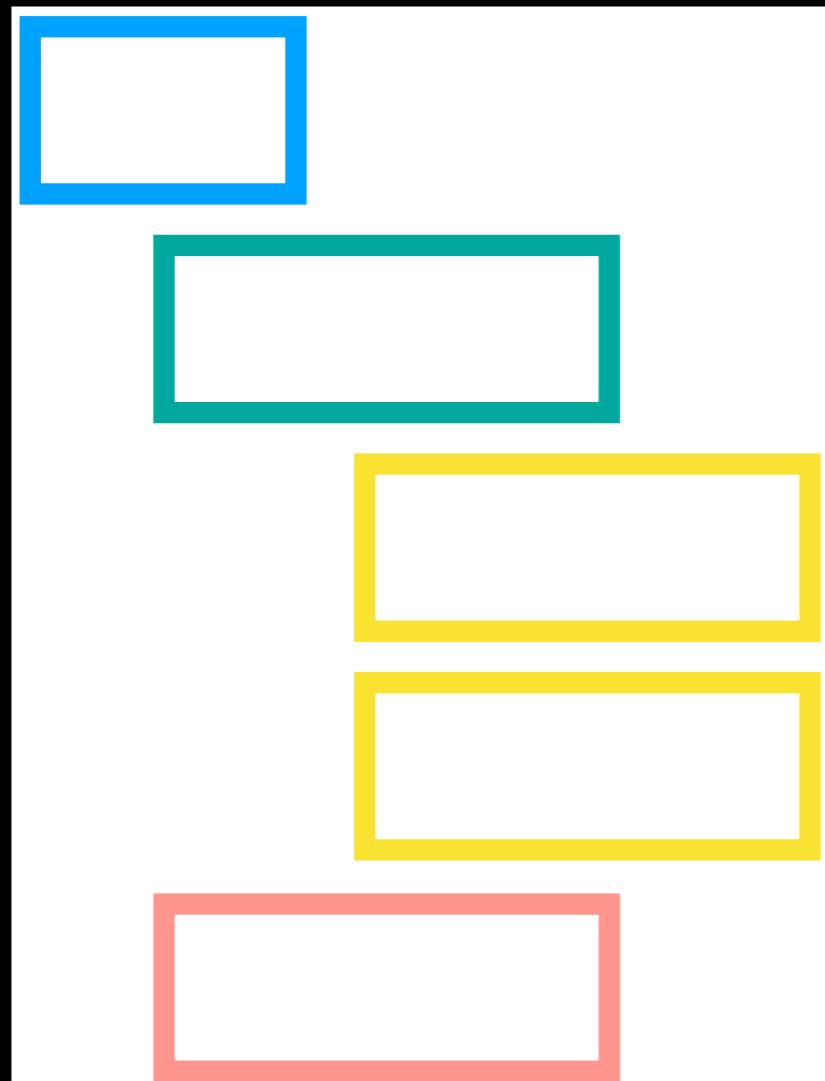


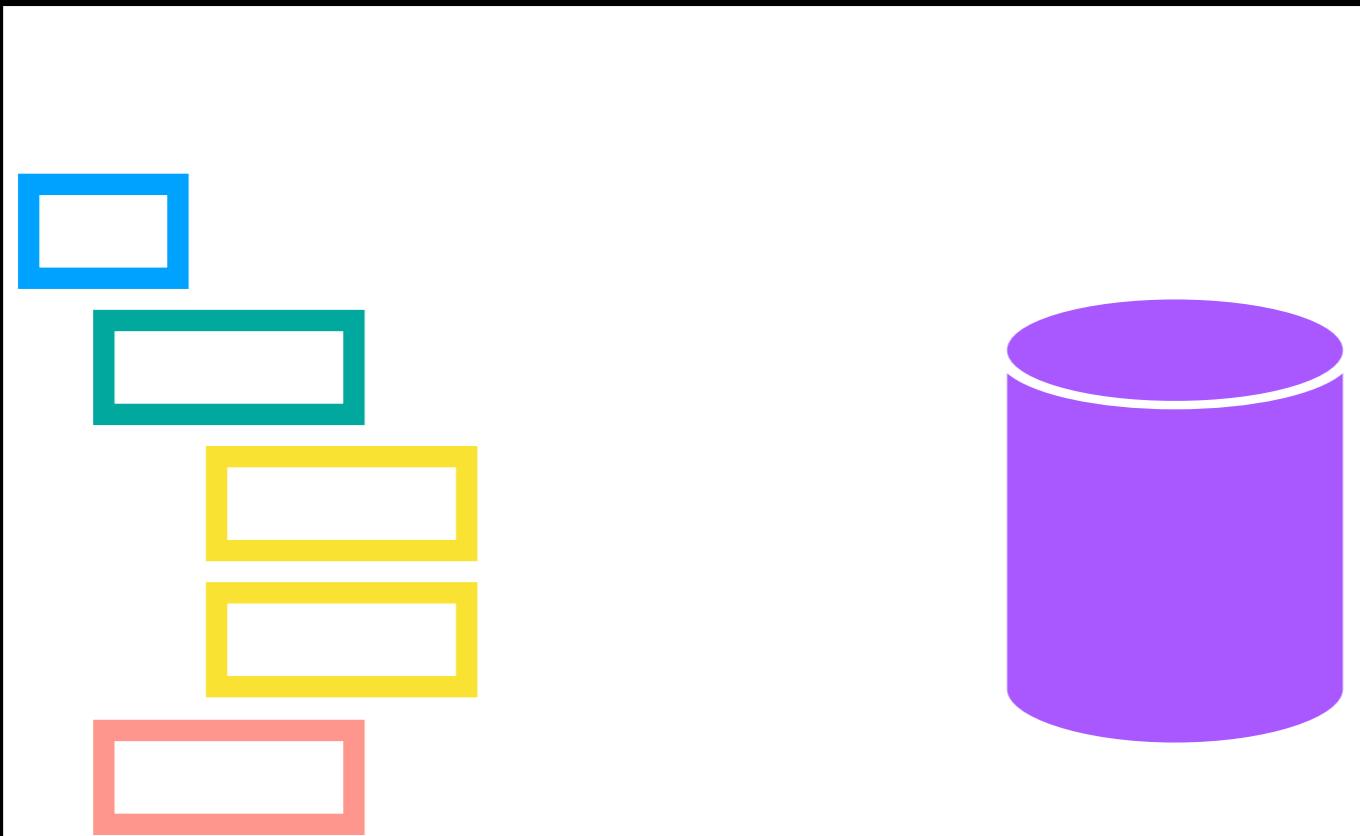
비싼 네트워크 요청

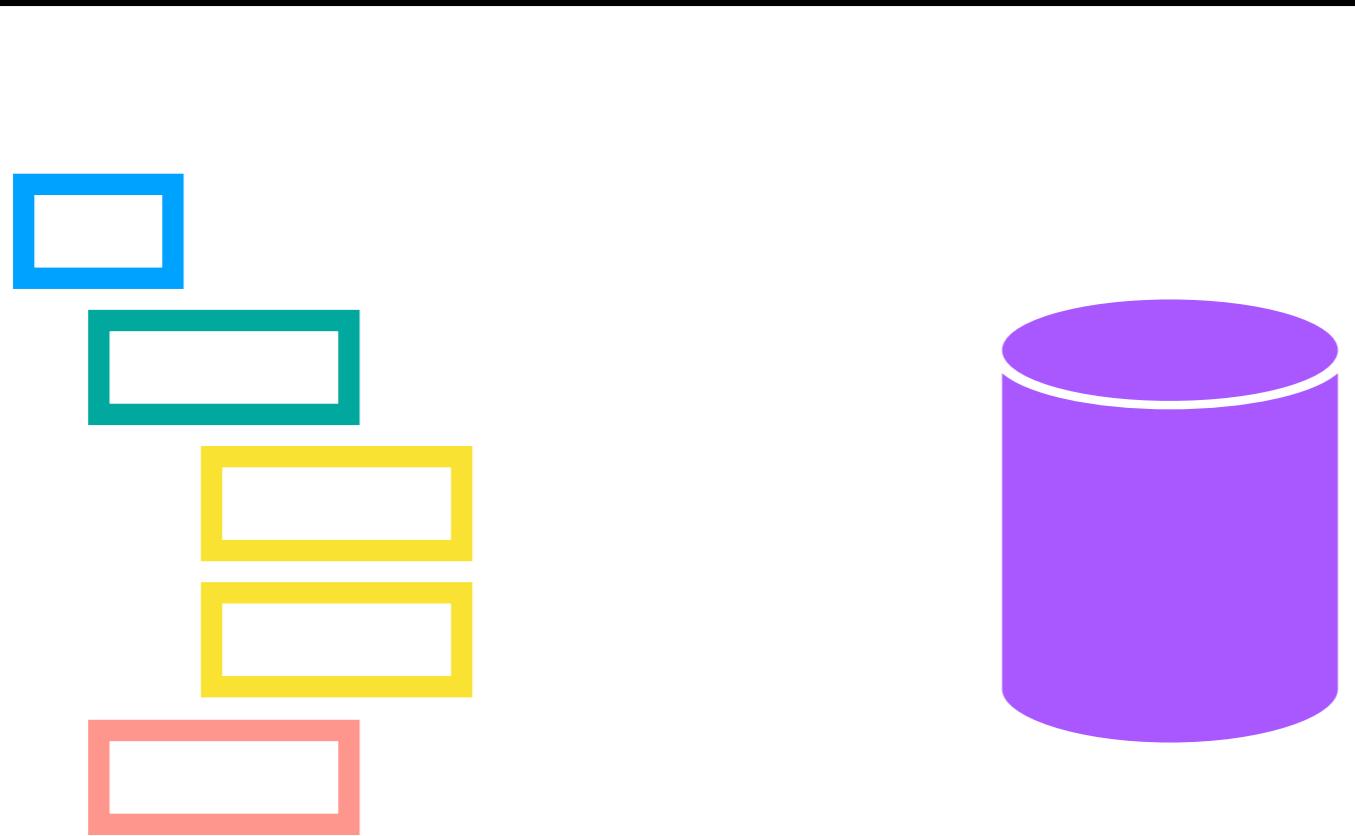


최소화

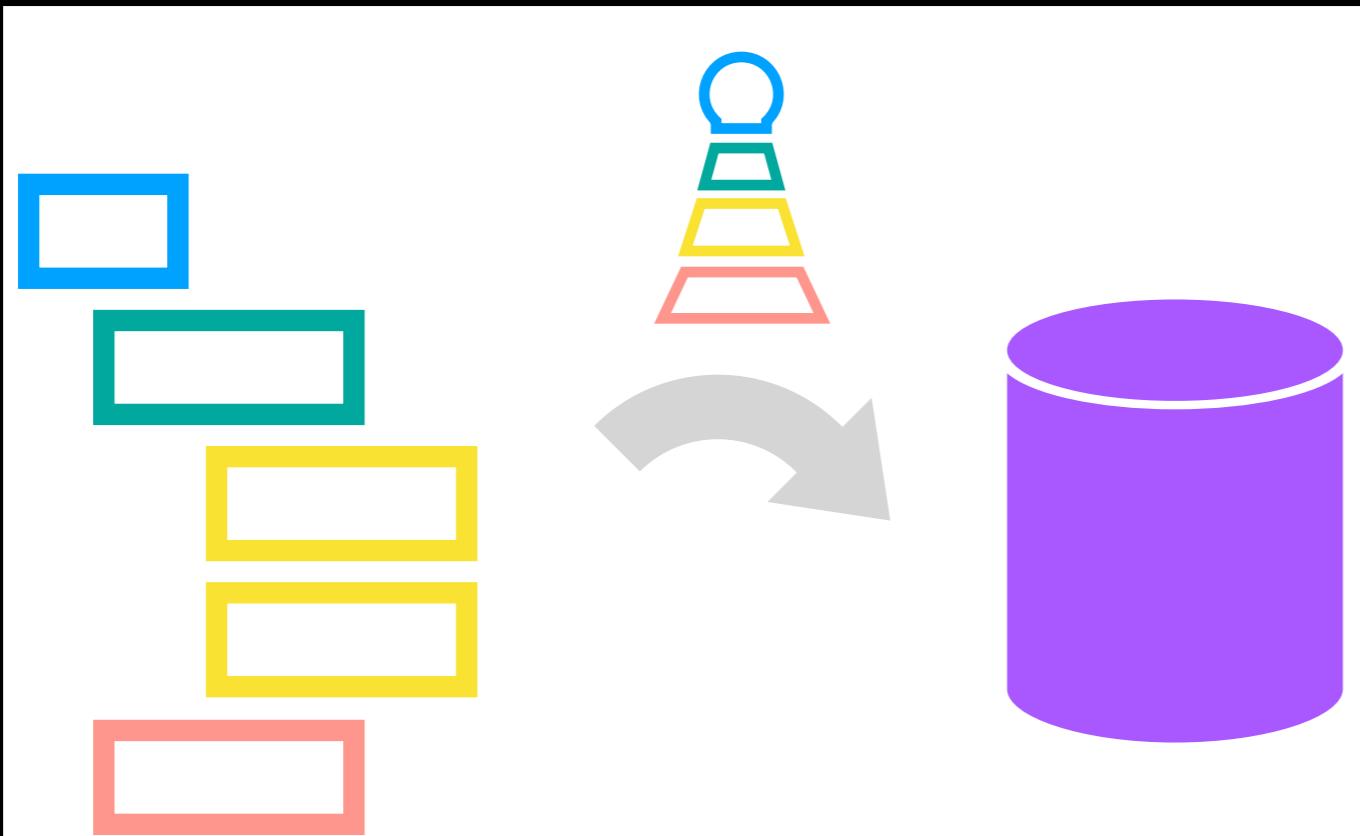




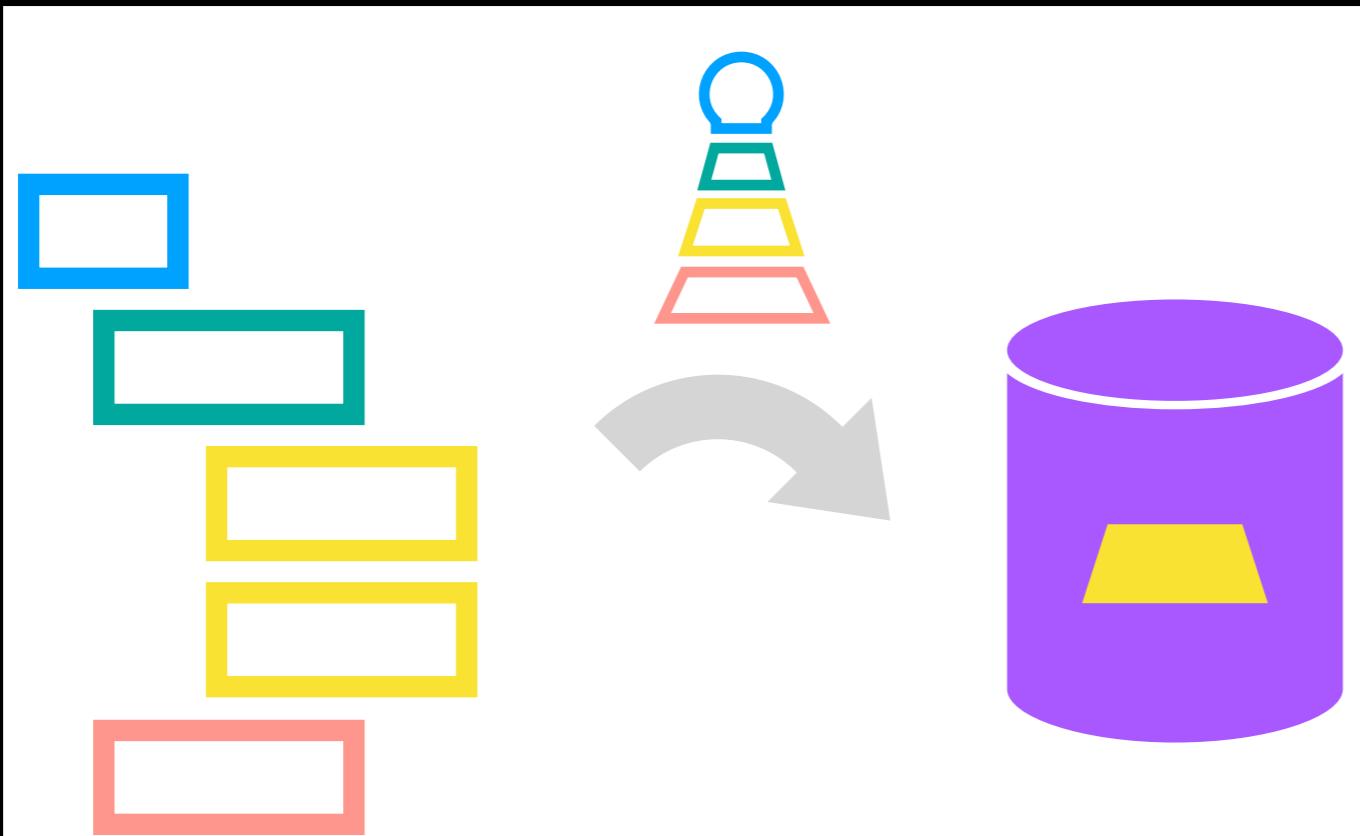




Relay



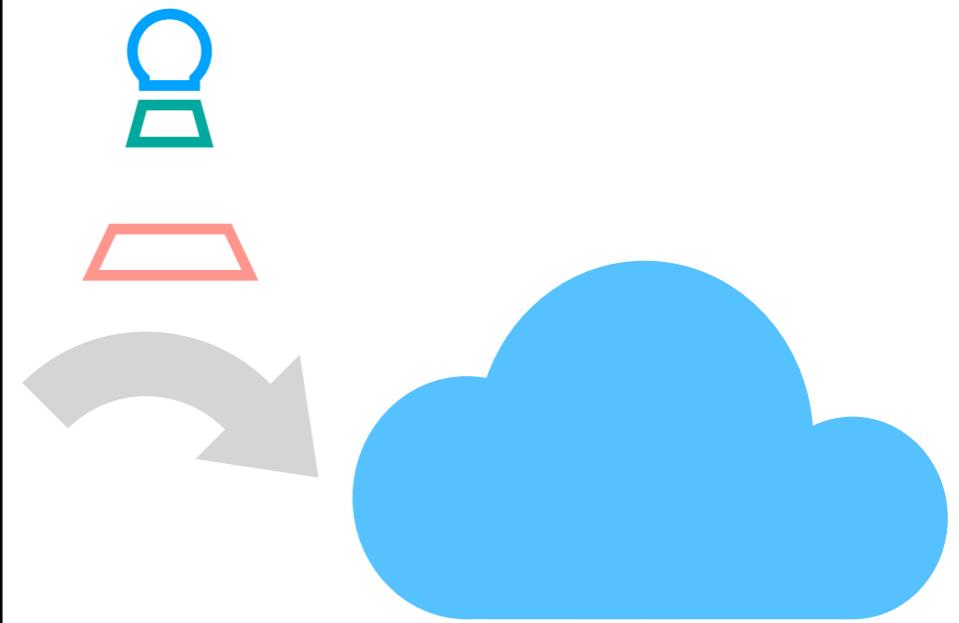
Relay

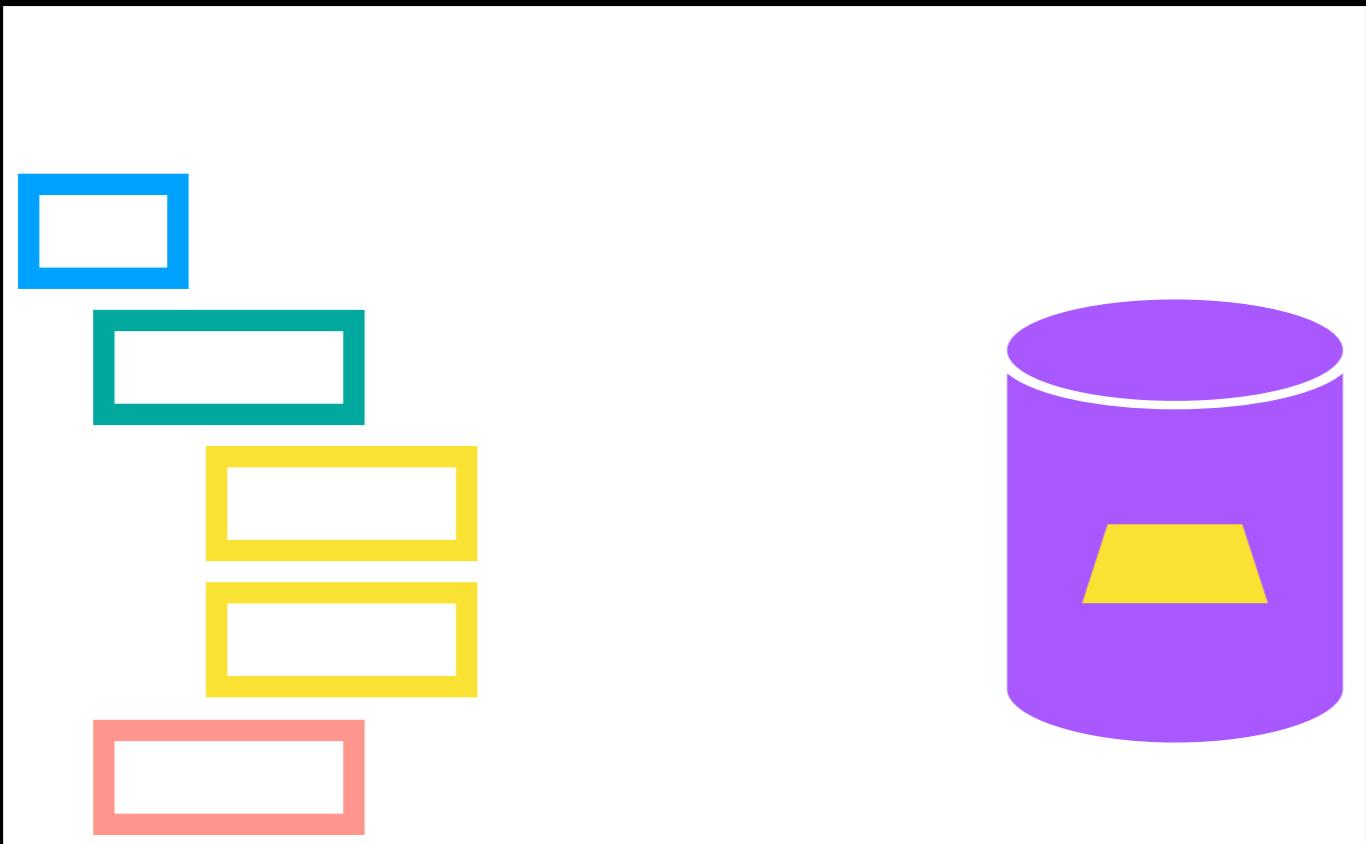


Relay

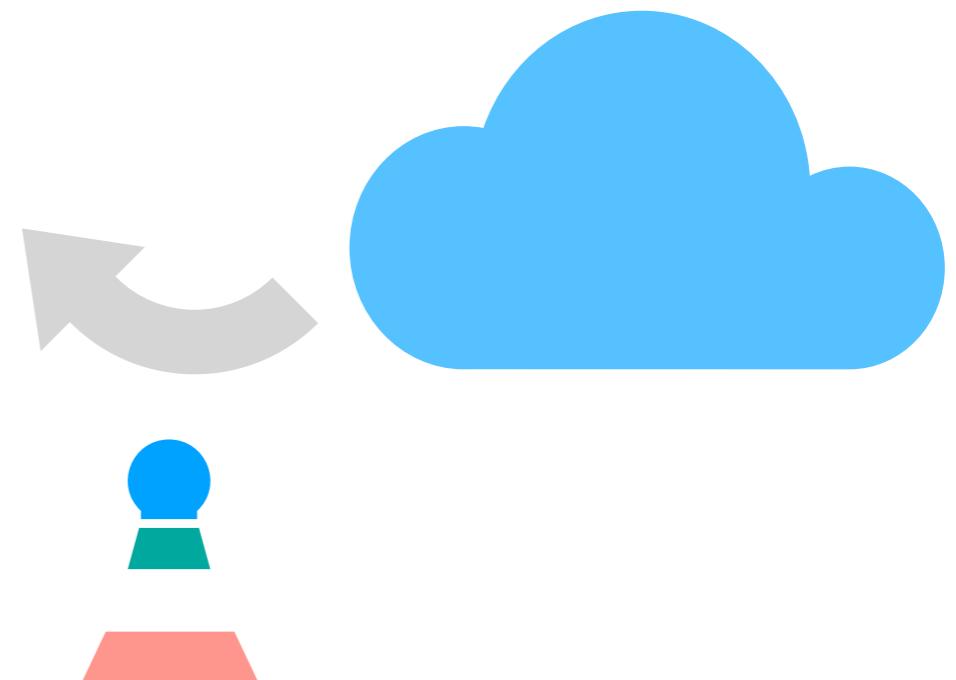


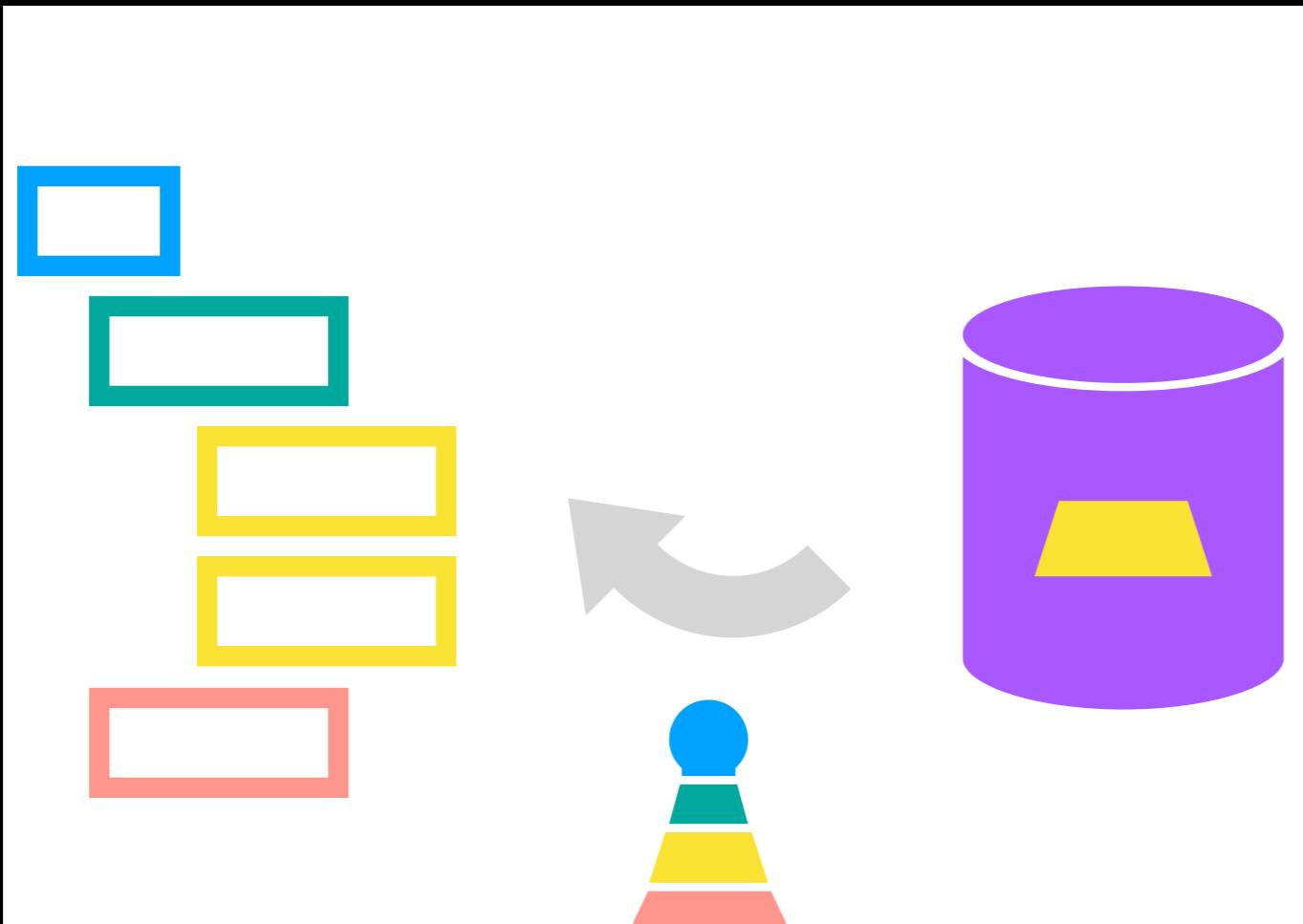
Relay

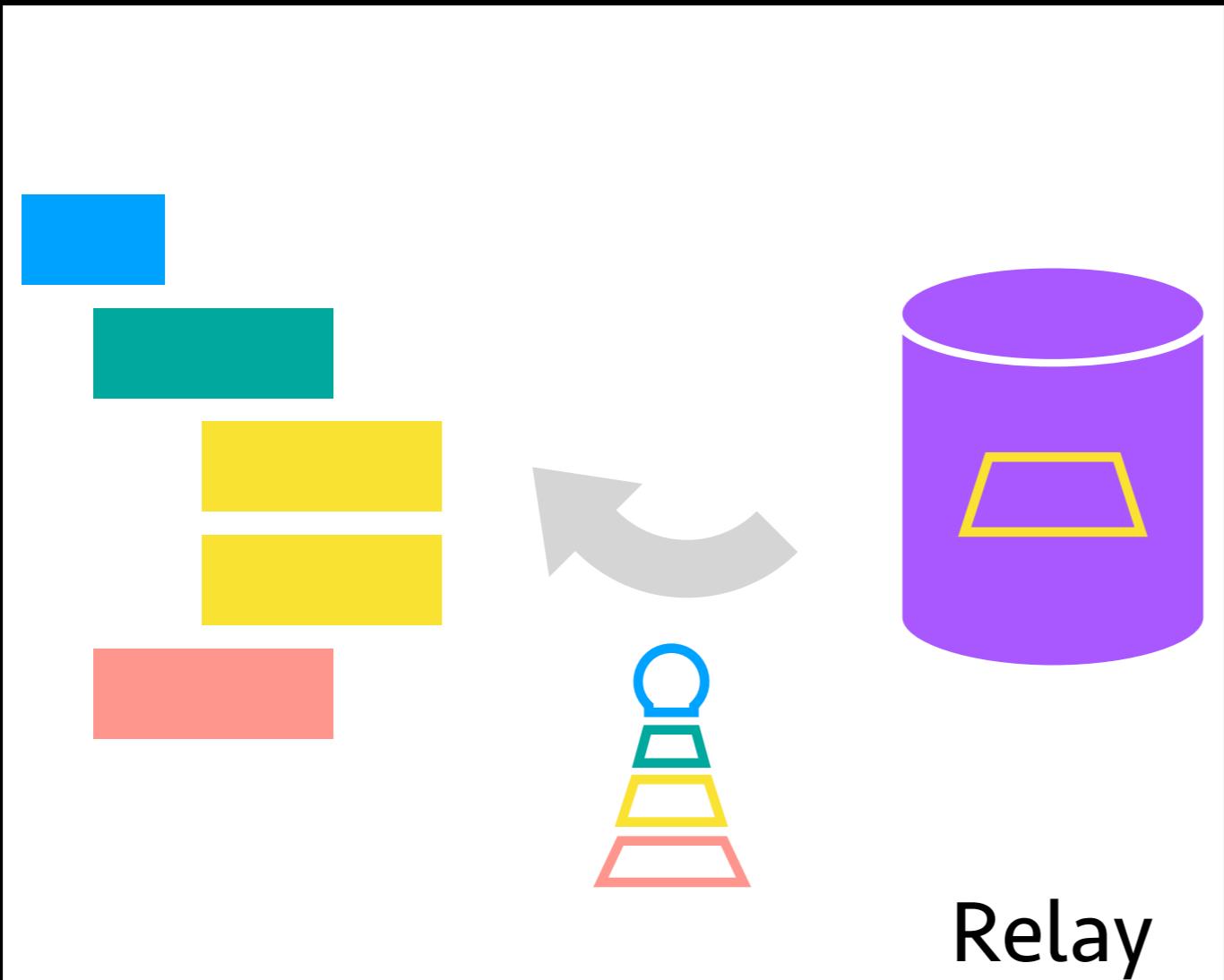




Relay

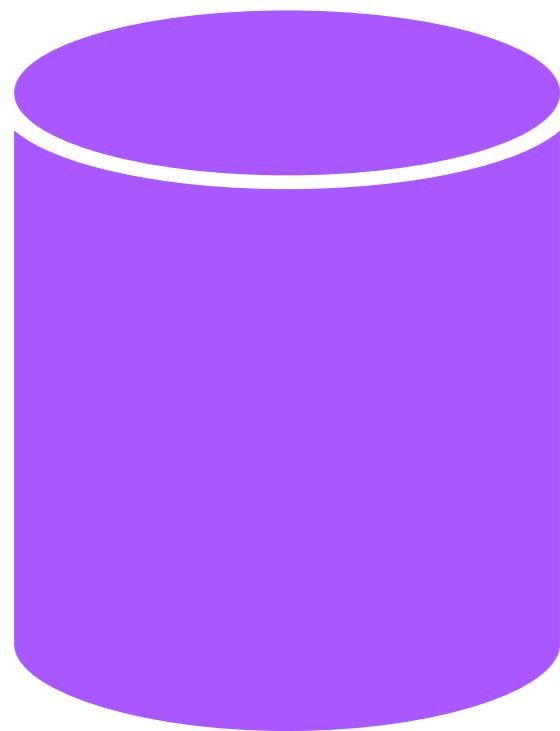




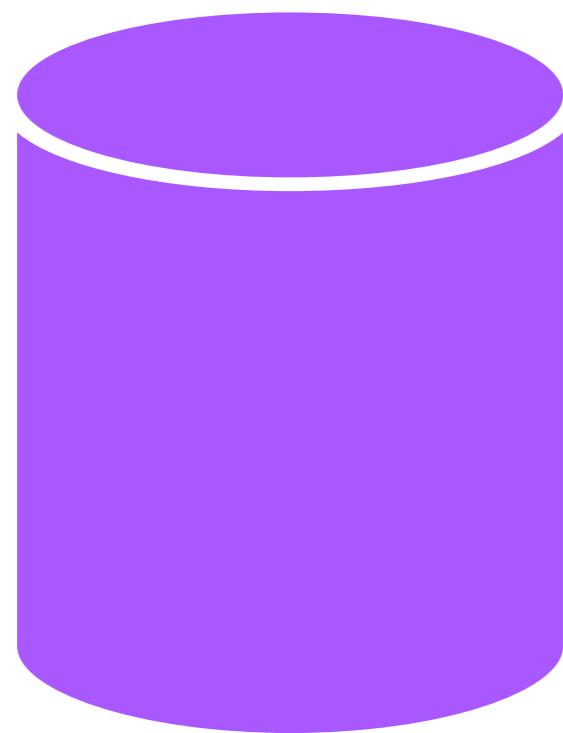


Relay

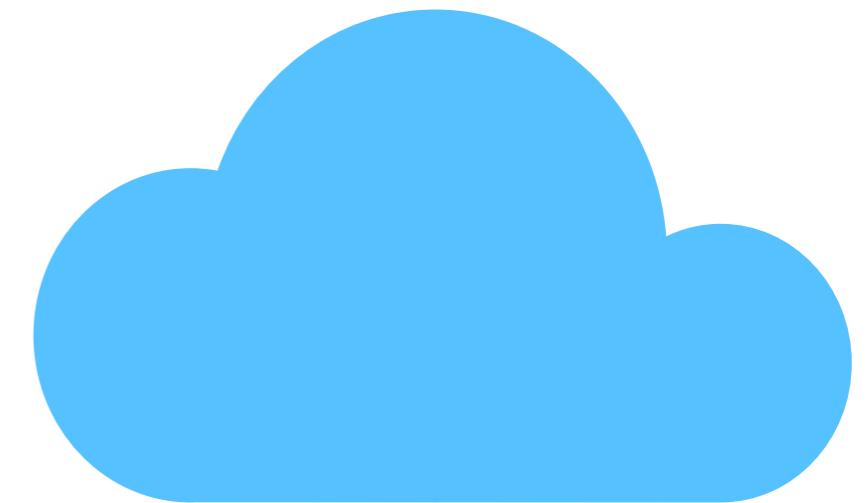
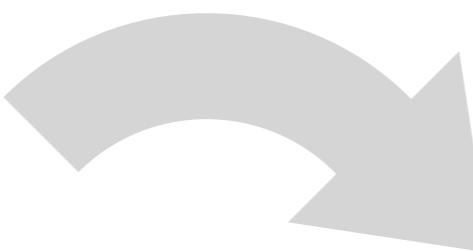
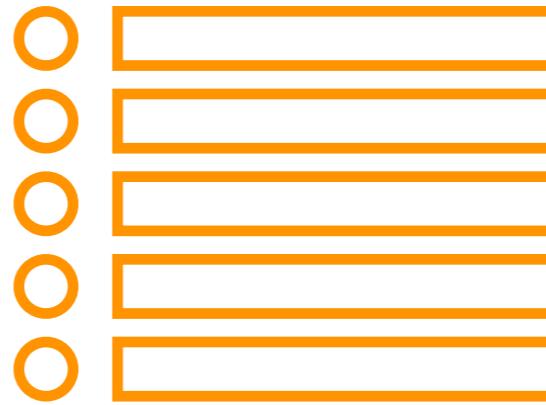
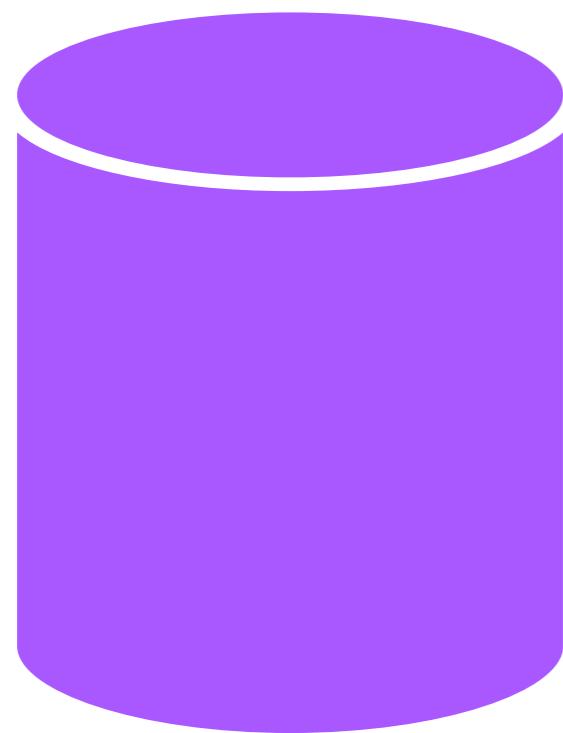
GraphQL Relay



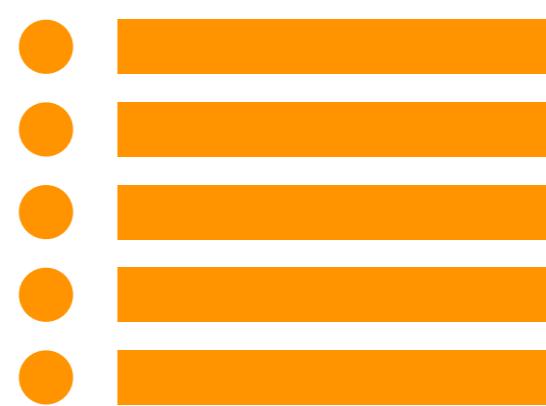
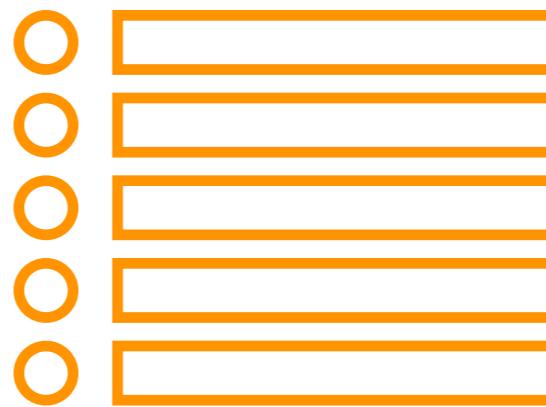
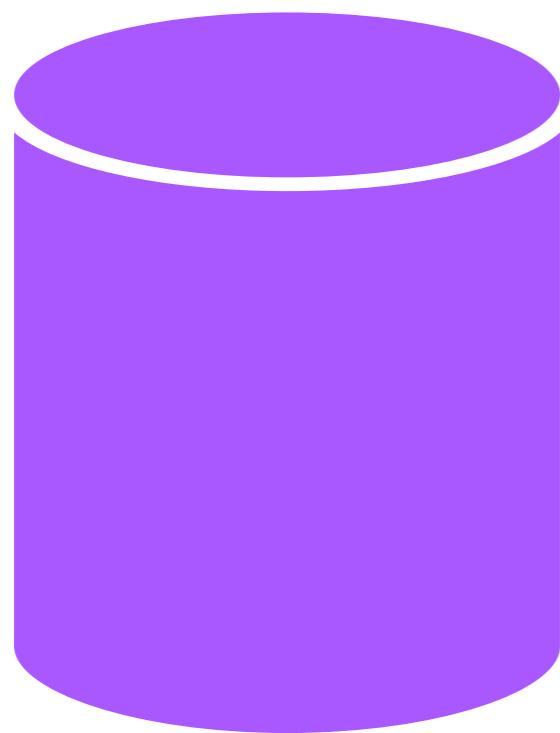
GraphQL Relay



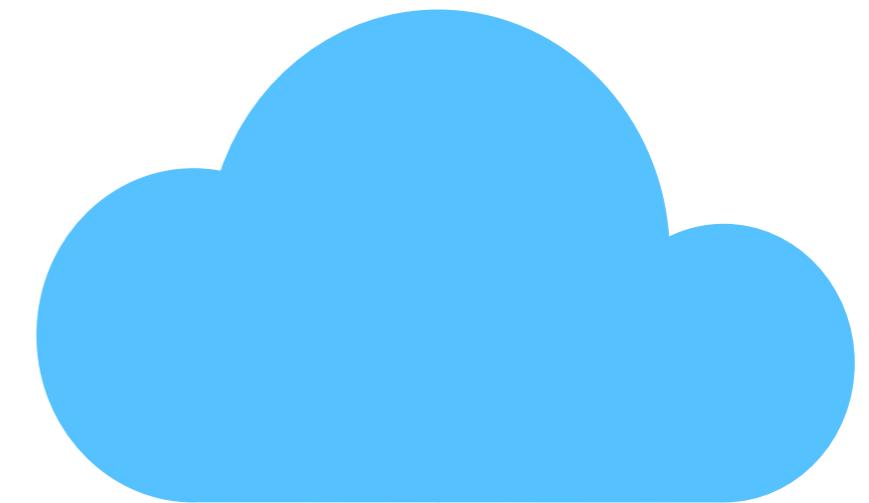
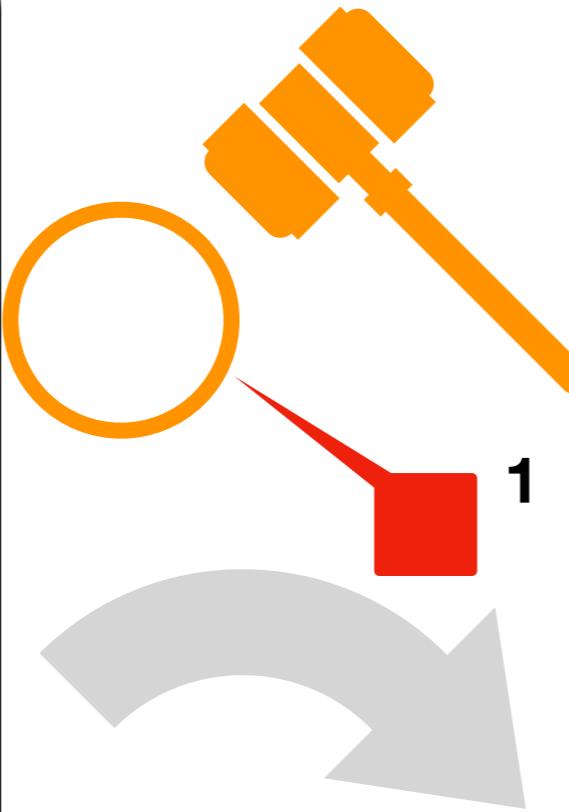
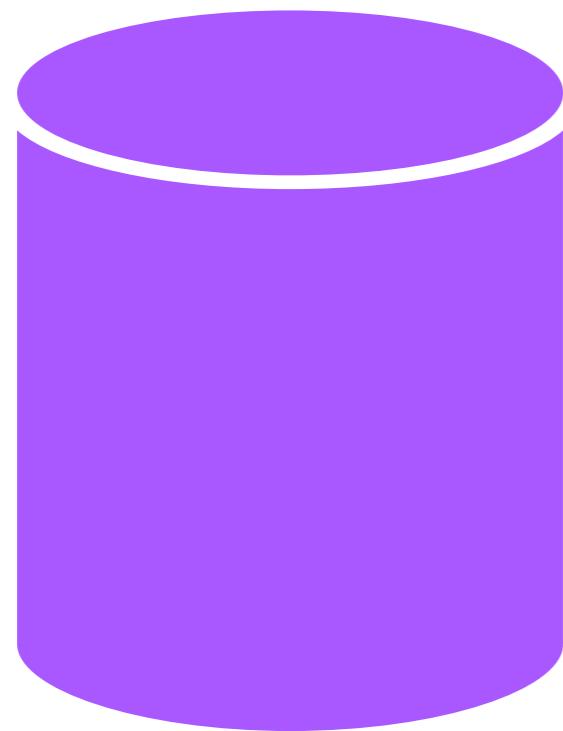
GraphQL Relay



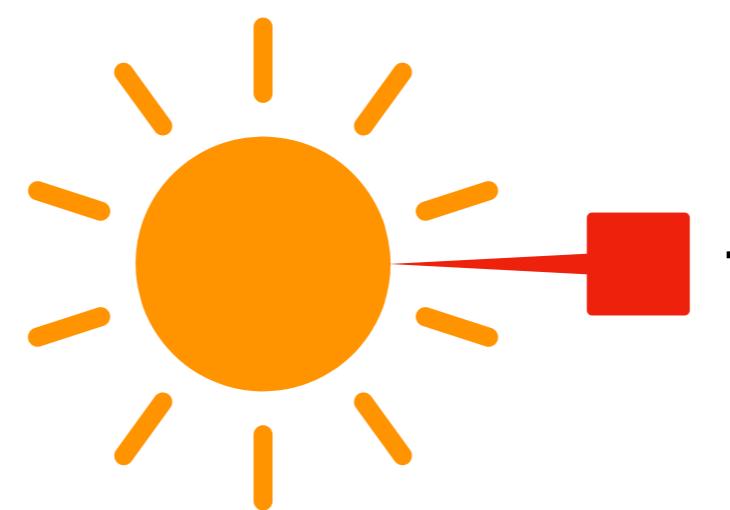
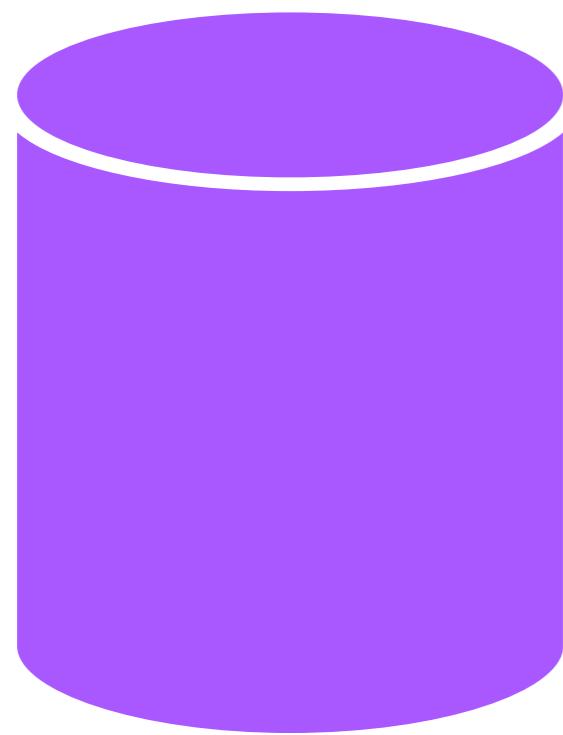
GraphQL Relay

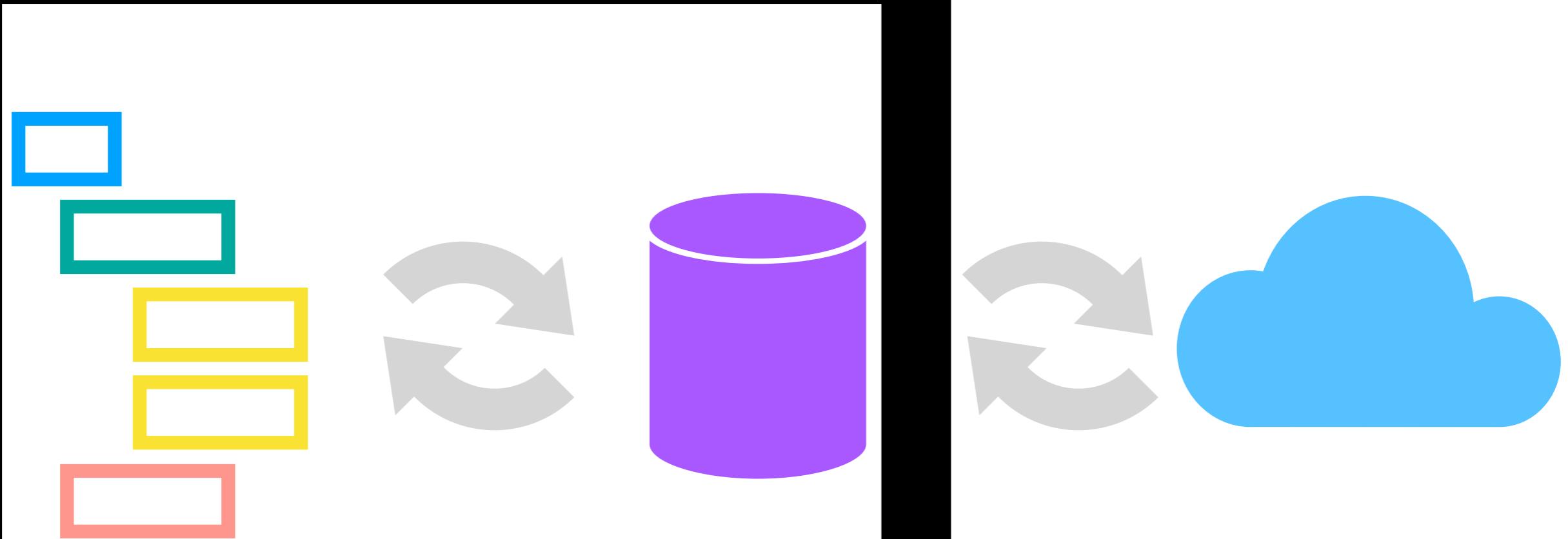


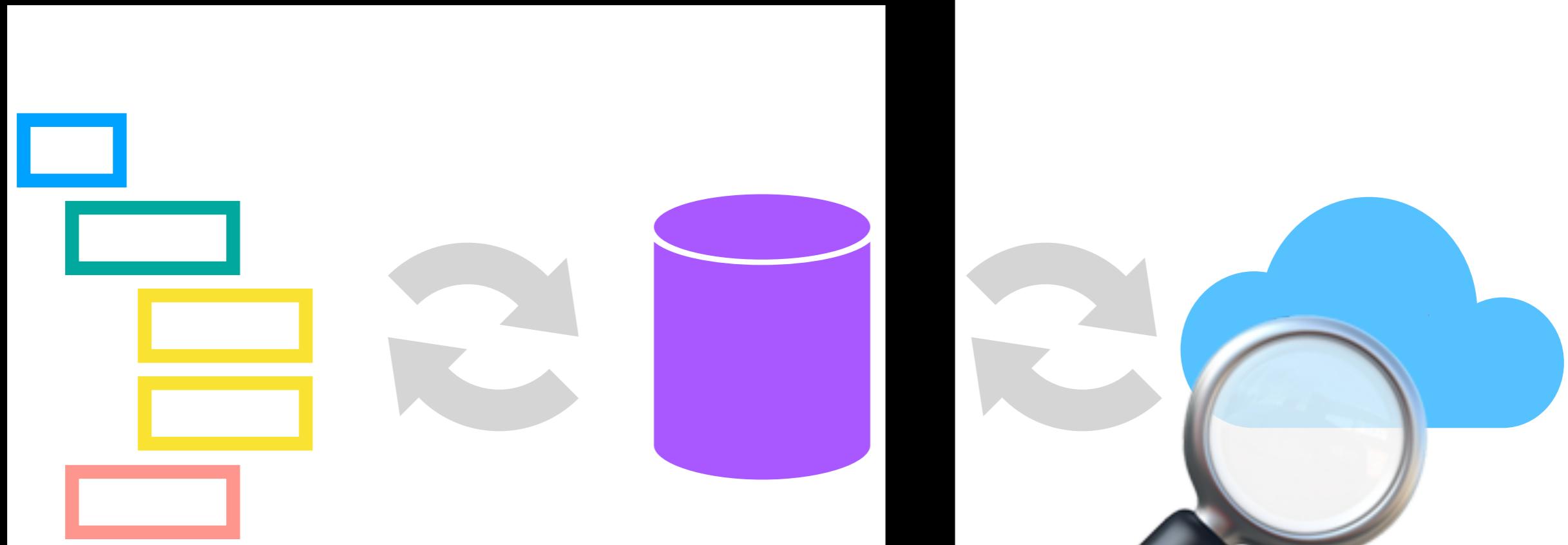
GraphQL Relay



GraphQL Relay

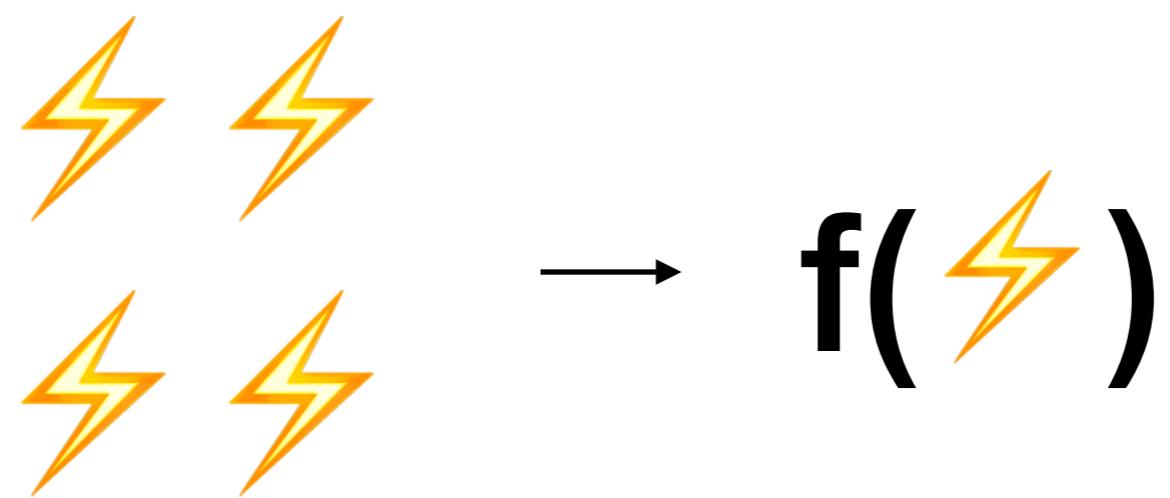






Server

Serverless



Functions

Serverless

= function-as-a-service?

Serverless

Pattern / Architecture

서버 관리의 모든 걸



클라우드에게 맡기자



/provisioning, OS patching/upgrading,
scaling, self-healing, ... /

Pricing



- IBM Cloud functions
 - 500ms 실행
 - 512MB @ 5,000,000 요청 횟수
 - \$14.45

Pricing



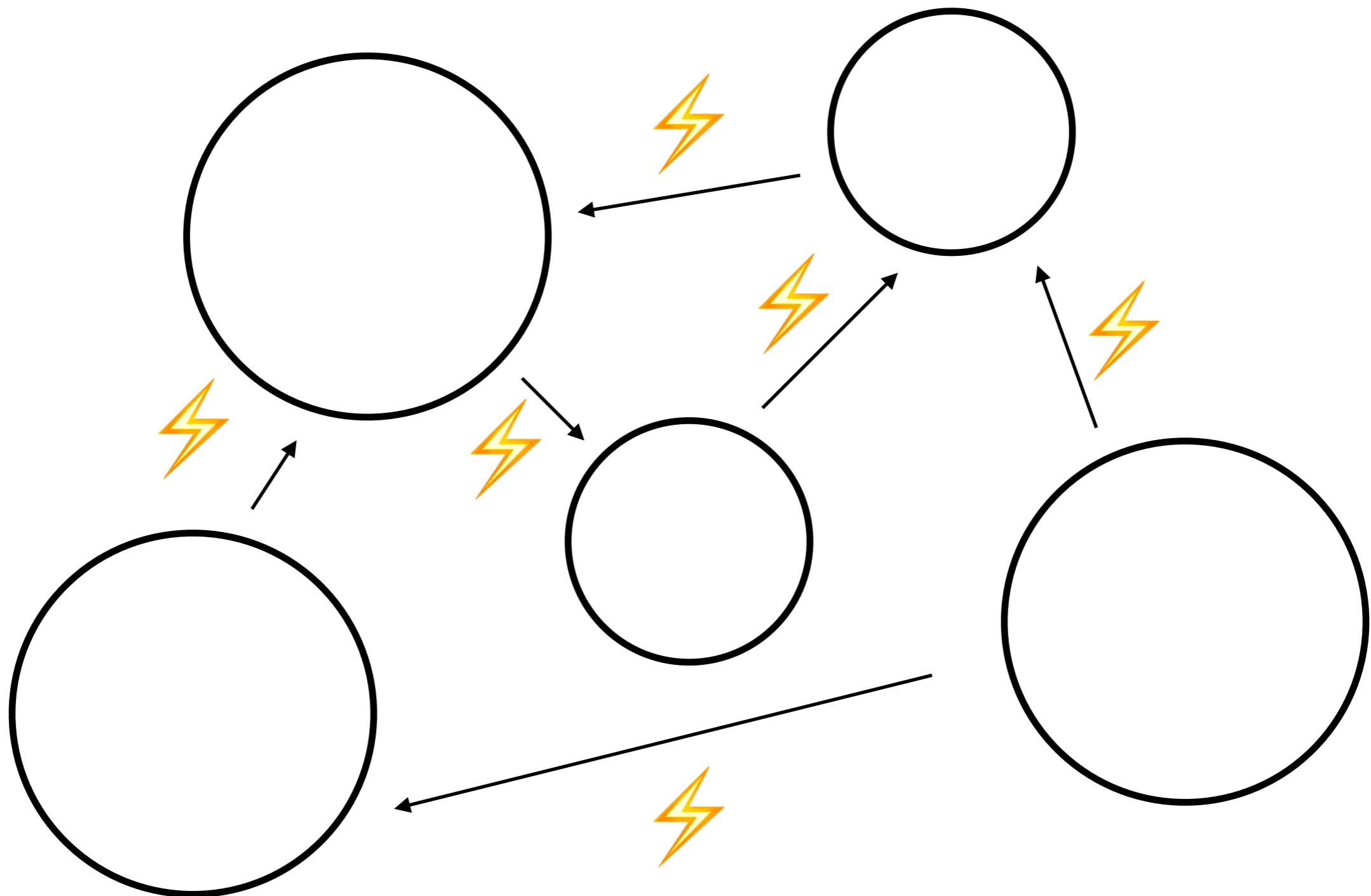
- 실행 시간 ⏳
- 메모리 소비량 🔥
- 요청 횟수 ⚡

“서버” 역할의 모든 걸

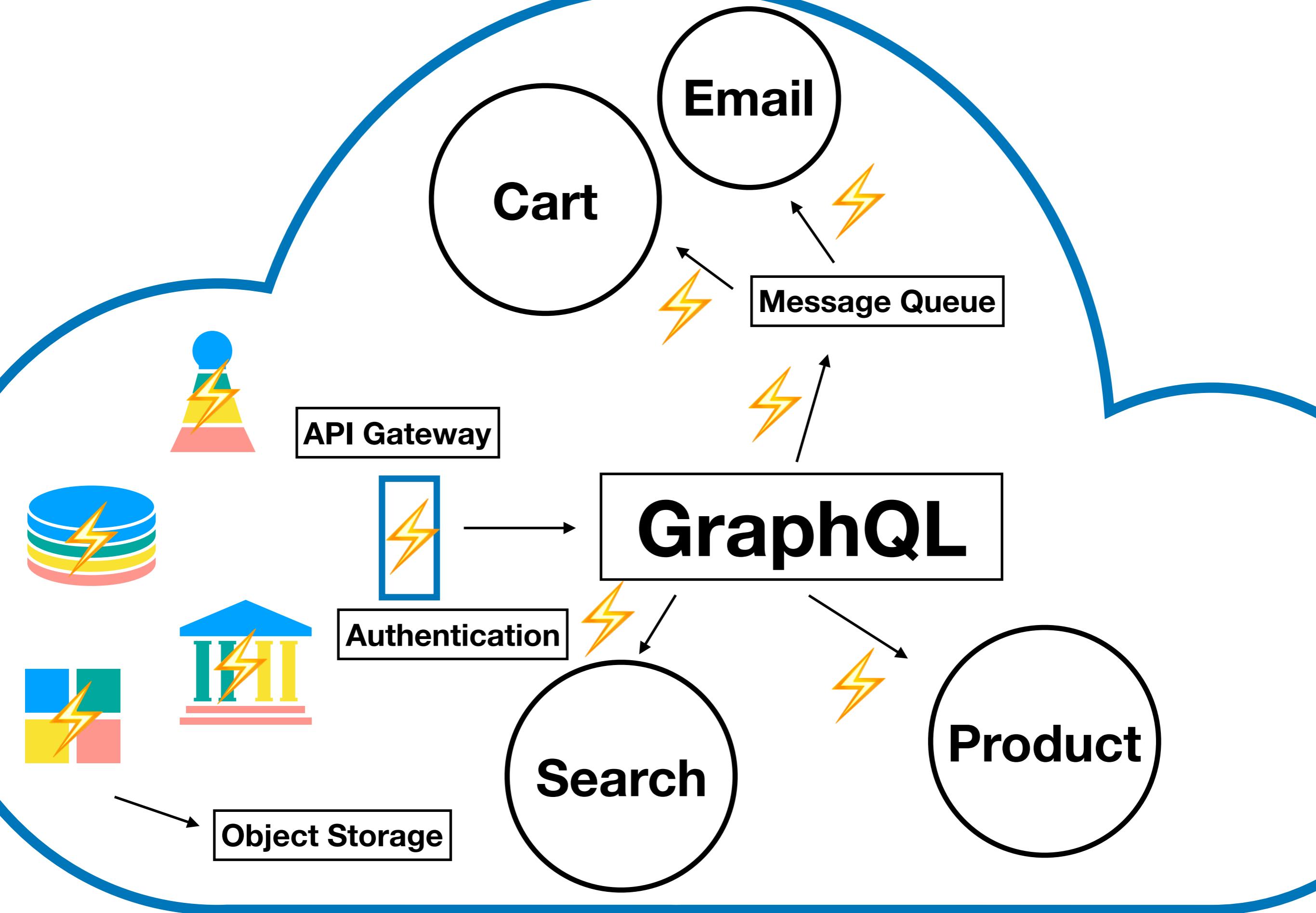
클라우드에게 맡기자 

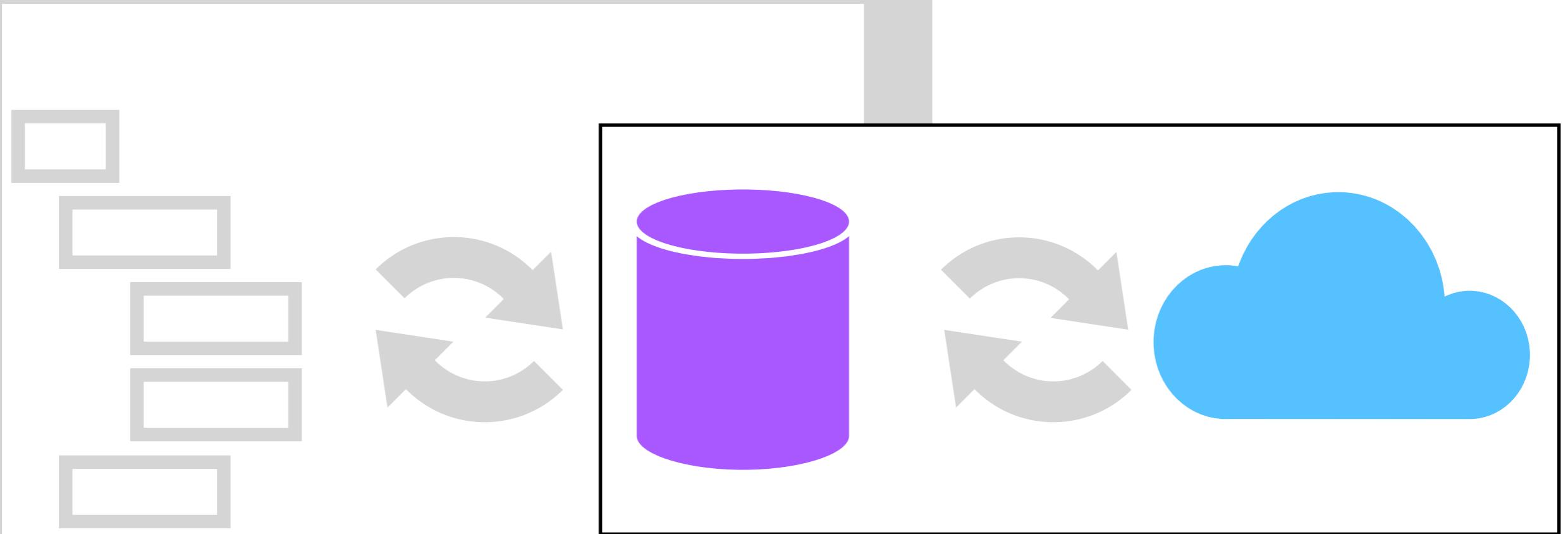
/API Gateway, Message Queue, Search,
Analytics, Authentication, ... /

이 “서버”들을 함수로 연결하자









GraphQL + Relay + Serverless

tumblbug.workable.com

창조적인 시도의 기술적 기반을 함께 만들
소프트웨어 엔지니어를 찾습니다.