

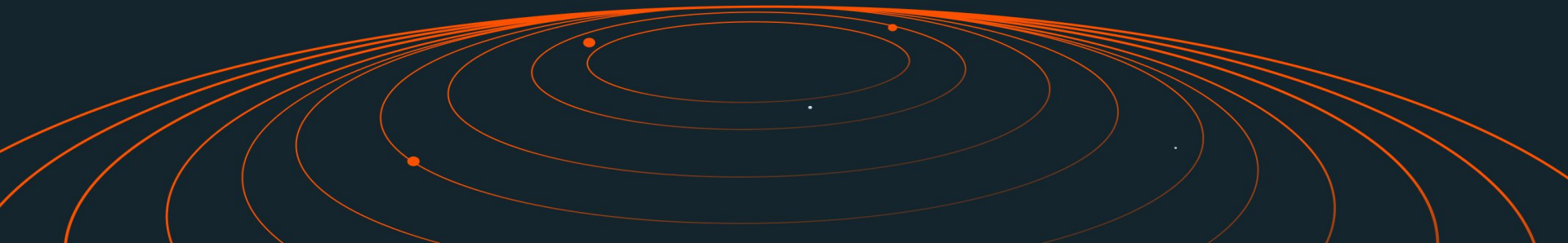


GraphQL Federation

The @key to your platform API strategy

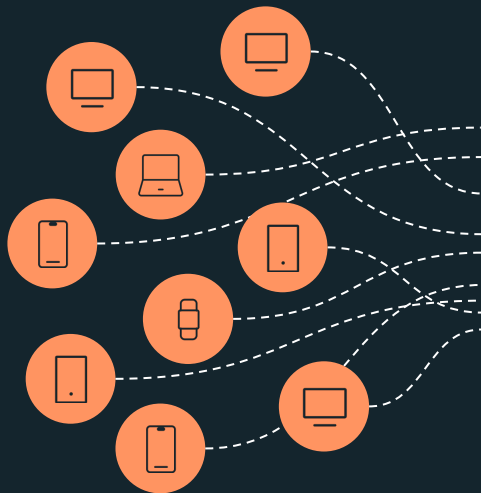
Derek Kuc

APOLLO



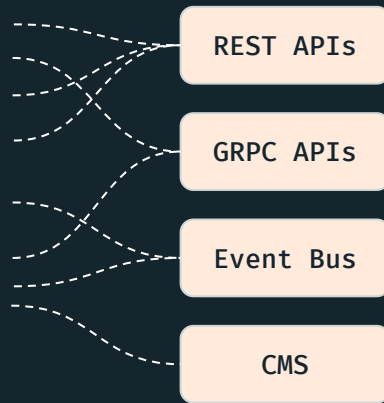
Experience APIs

Clients



- Aggregation?
- Integration?
- Orchestration?
- Augmentation?

Services



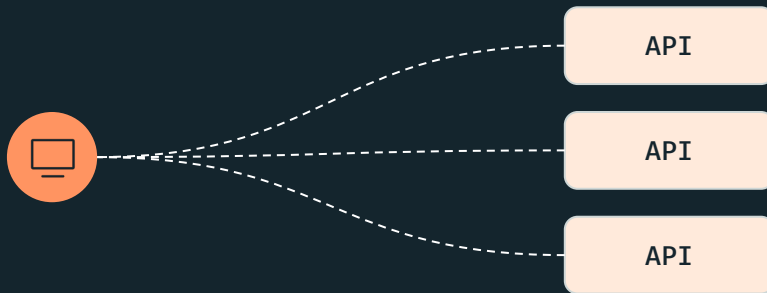
Strategy #1

No layer



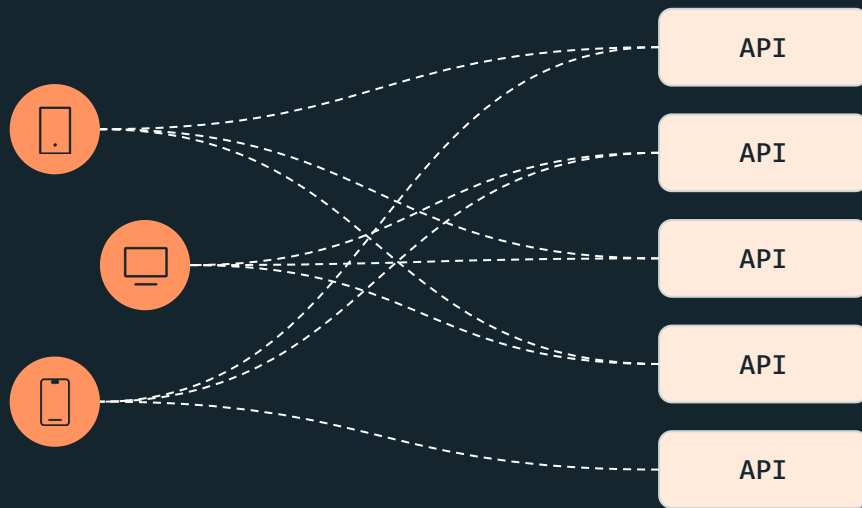
Strategy #1

No layer



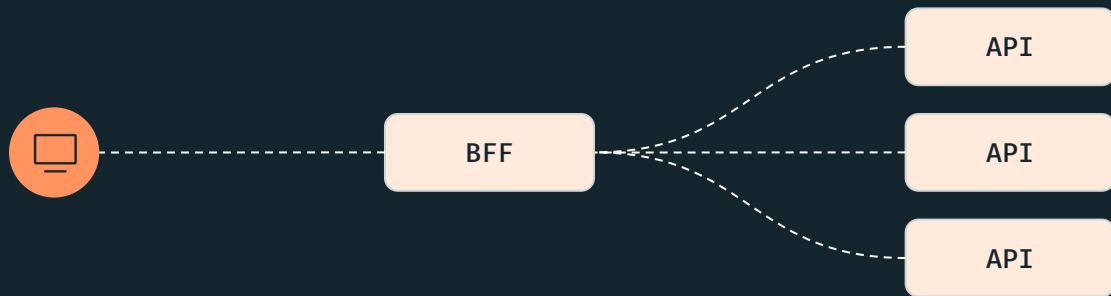
Strategy #1

No layer



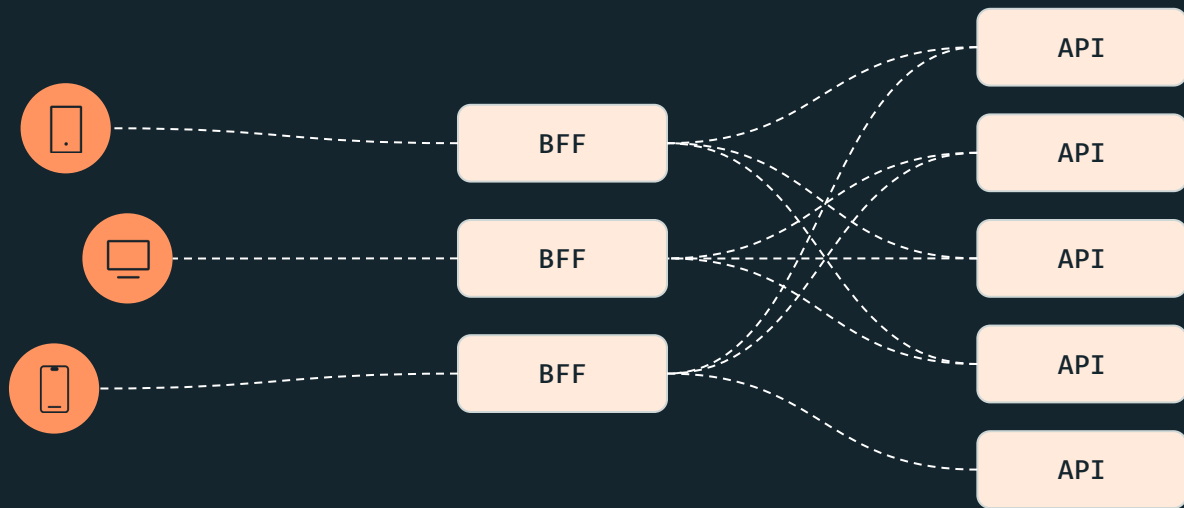
Strategy #2

Backend For Frontend (BFF)



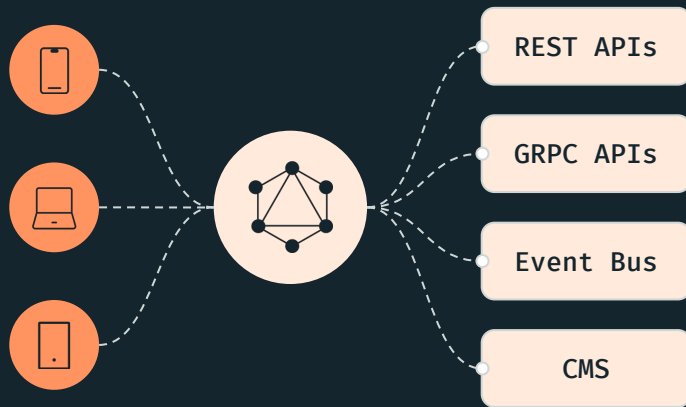
Strategy #2

Backend For Frontend (BFF)



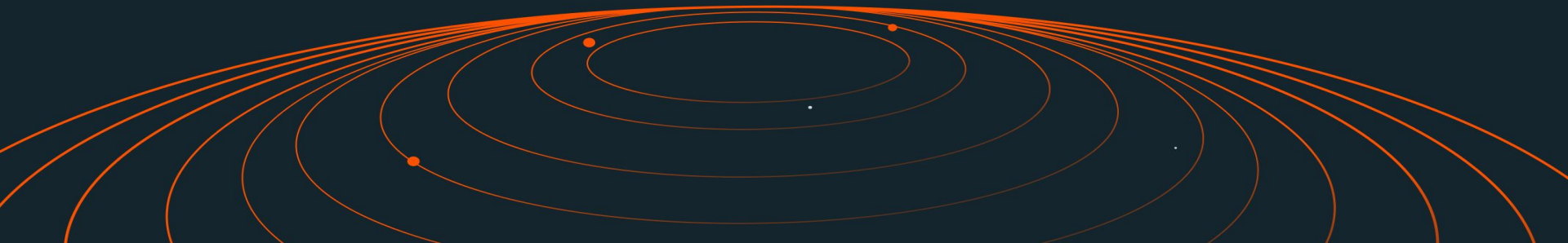
Strategy #3

GraphQL Federation*



What is GraphQL?

A query language for your APIs.



What is GraphQL?

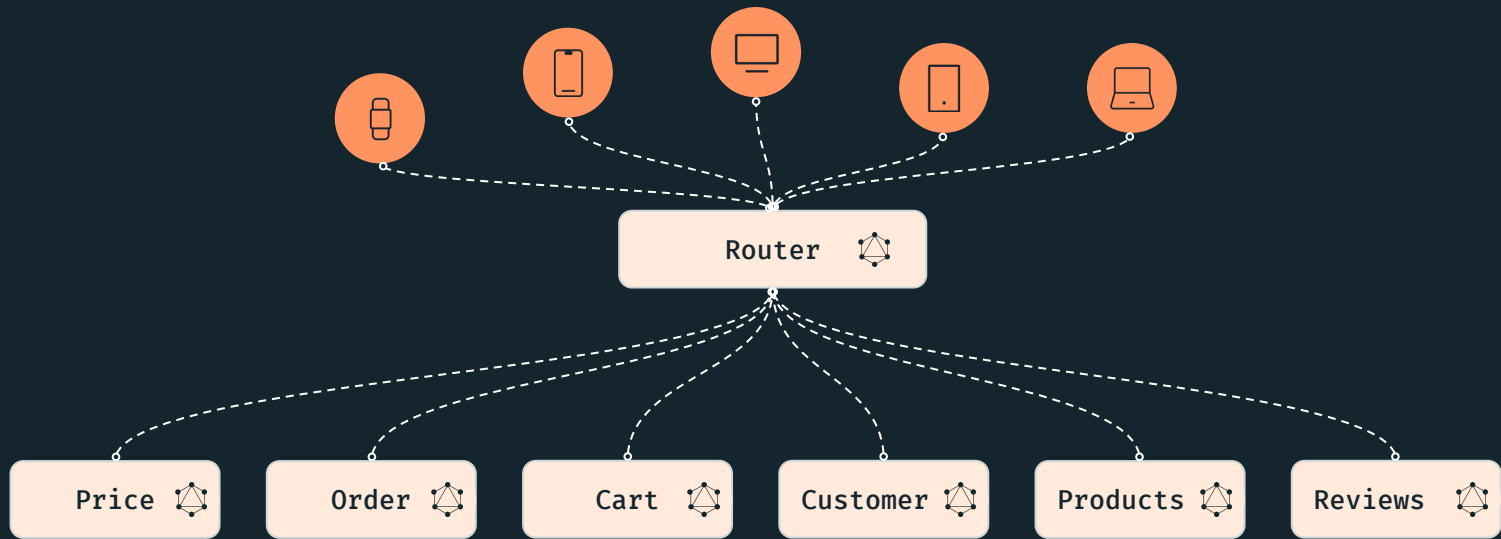
A POST request with a body that specify the shape of the response.

```
query GetProduct($id: ID!) {  
  product(id: $id) {  
    name  
    description  
    reviews {  
      text  
      rating  
    }  
  }  
}
```

```
{  
  "data": {  
    "product": {  
      "name": "GOTO Chicago"  
      "description": "2024"  
      "reviews": [{  
        "text": "Amazing!"  
        "rating": "10/10"  
      }]  
    }  
  }  
}
```

GraphQL Federation

Open Architecture for building **Distributed GraphQL APIs**.



Entities

Extend types

```
# Products subgraph
type Product @key(fields: "id") {
  id: ID!
  name: String!
  description: String
  dimensions: Dimensions!
}
```

```
# Reviews subgraph
type Product @key(fields: "id") {
  id: ID!
  reviews: [Review!]!
}

type Review @key(fields: "id") {
  id: ID!
  text: String
  rating: Int!
}
```

Entities

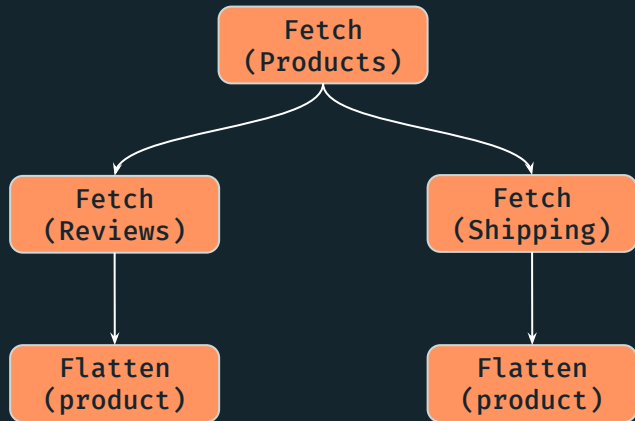
Use data from **other** subgraphs

```
# Products subgraph
type Product @key(fields: "id") {
  id: ID!
  name: String!
  description: String
  dimensions: Dimensions!
}
```

```
# Shipping subgraph
type Product @key(fields: "id") {
  id: ID!
  dimensions: Dimensions! @external
  shippingCost(zipCode: Int!): Price!
  @requires(fields: "dimensions { width
height weight}")
}
```

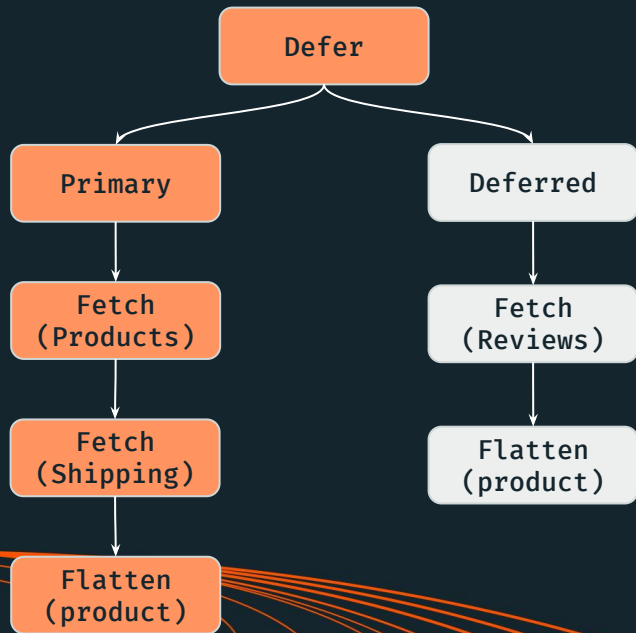
Query Plans

```
query GetProduct($id: ID!, $zipCode: ID!) {  
  product(id: $id) { // products subgraph  
    name  
    description  
    reviews { // reviews subgraph  
      text  
      rating  
    }  
    shippingCost(id: $zipCode) // shipping  
  }  
}
```



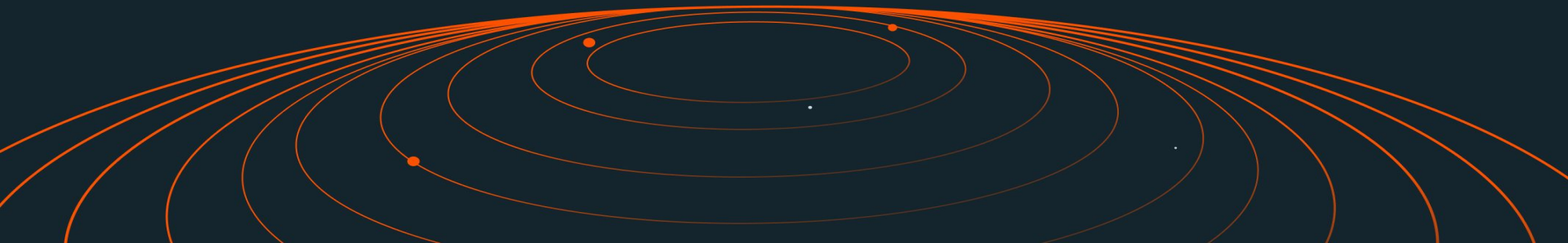
@defer

```
query GetProduct($id: ID!, $zipCode: ID!) {  
  product(id: $id) { // product subgraph  
    name  
    description  
    ... @defer {  
      reviews { // reviews subgraph  
        text  
        rating  
      }  
    }  
    shippingCost(id: $zipCode) // shipping  
  }  
}
```

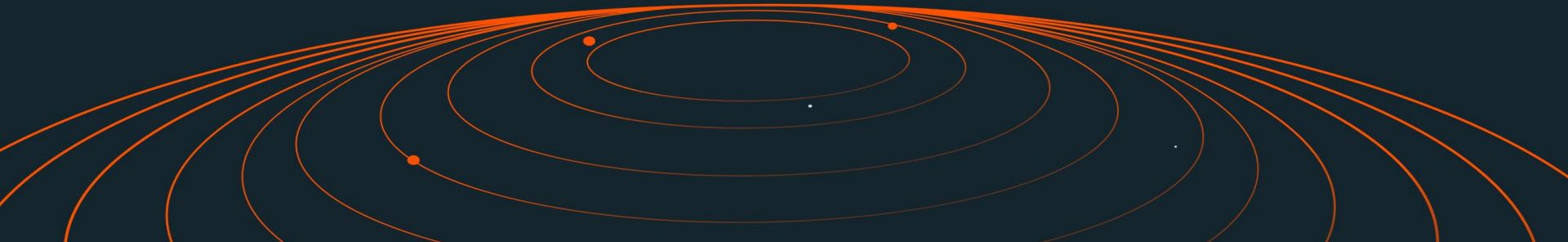


Apollo Federation

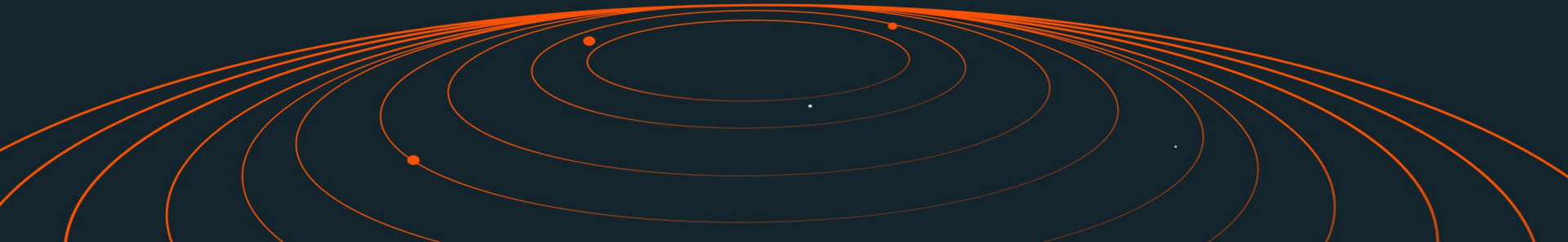
- Entity Caching
- Extend entity interfaces through `@interfaceObject`
- Field level authorization with `@authenticated` / `@requiresScopes` / `@policy`
- Incremental migration with progressive `@override`
- Using `@context` to share data
- ... and more



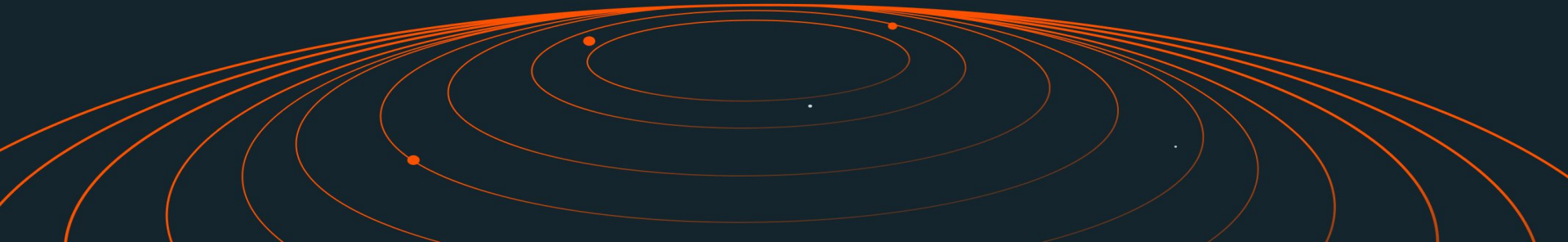
Why?



GraphQL **vs** REST?



~~GraphQL vs REST?~~ Wrong Question!



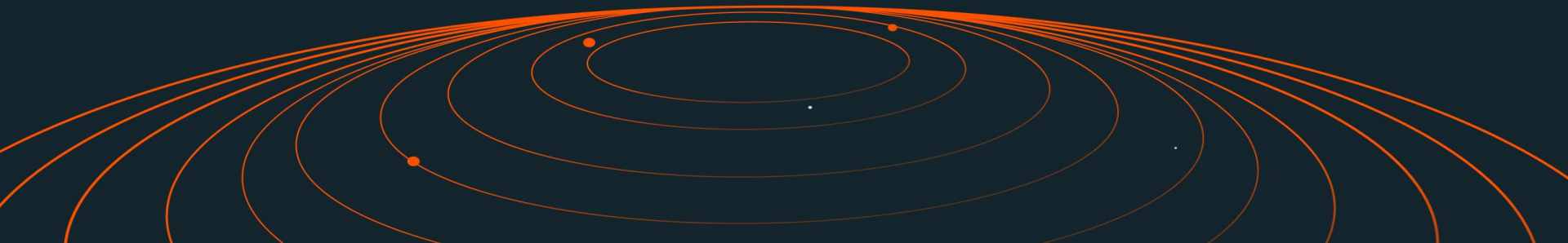
GraphQL Makes your APIs **Better**

Frontend

- No under/over fetching
- Strongly Typed
- Faster Application Delivery
- Extra capabilities - @defer

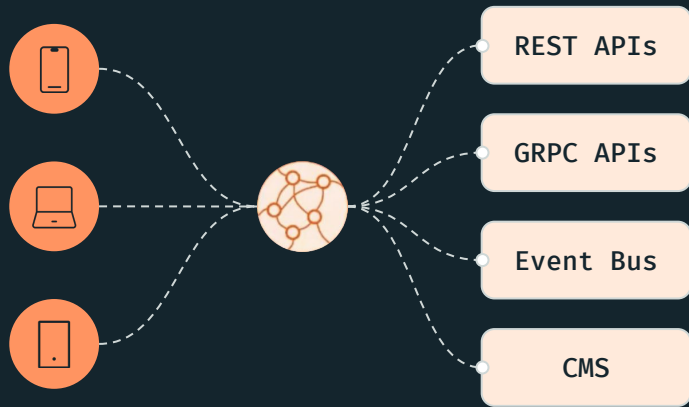
Backend

- "BFF" for everyone
- Observability
- Evolvability
- Extra capabilities - Query Planner



New Experiences?

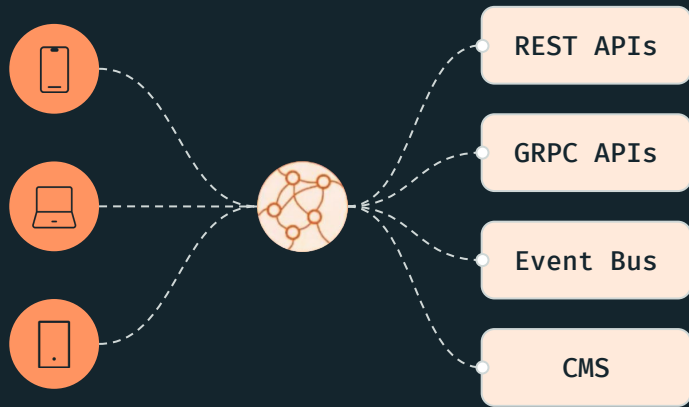
A LLM is just another client



Direct?

New **Experiences?**

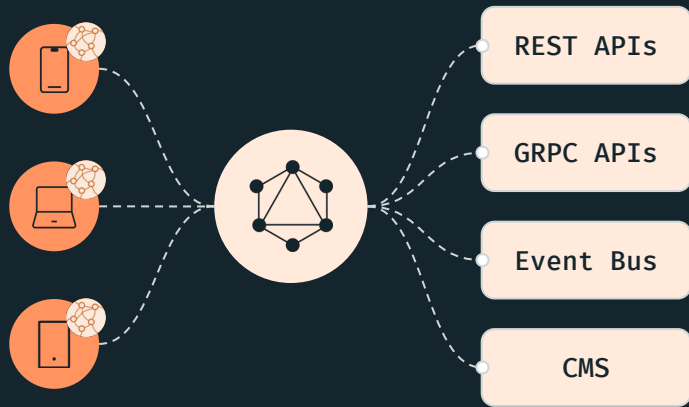
A LLM is just another **client**

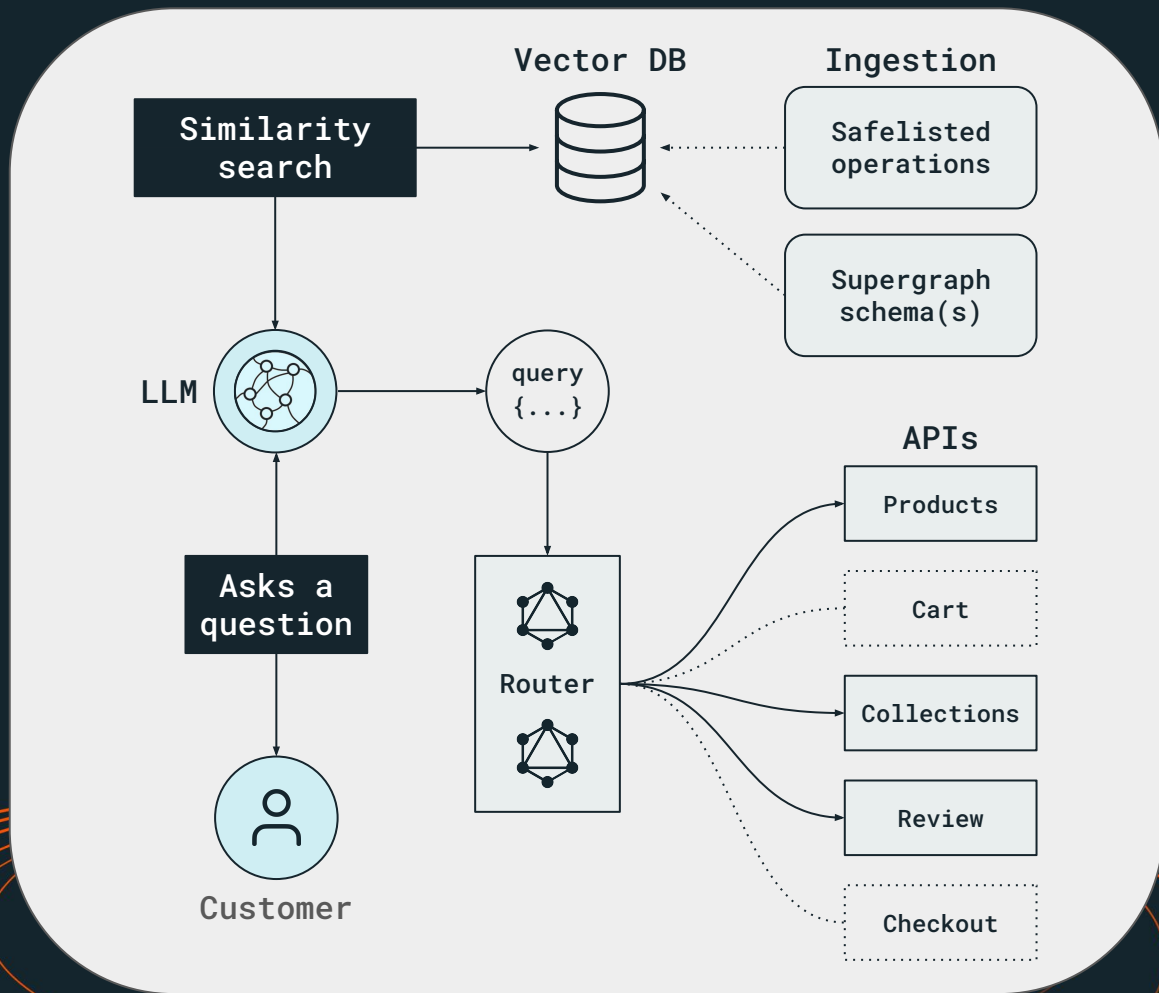


**LLMs aren't
smart enough
to do this**

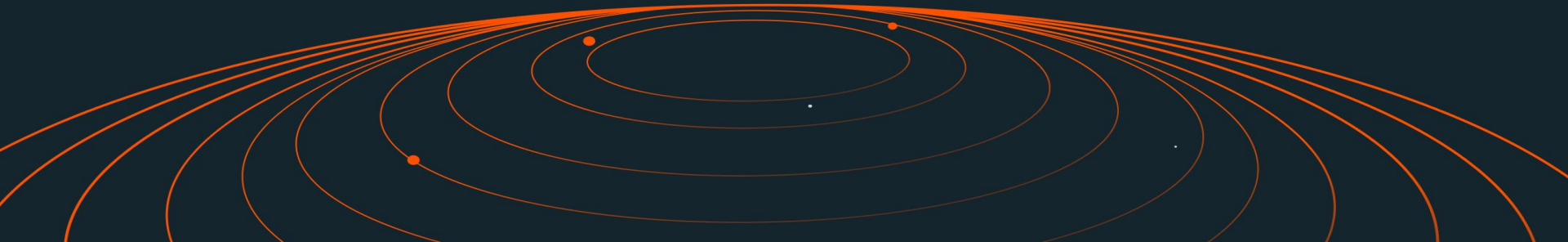
New Experiences?

GraphQL Federation is the ideal translation layer

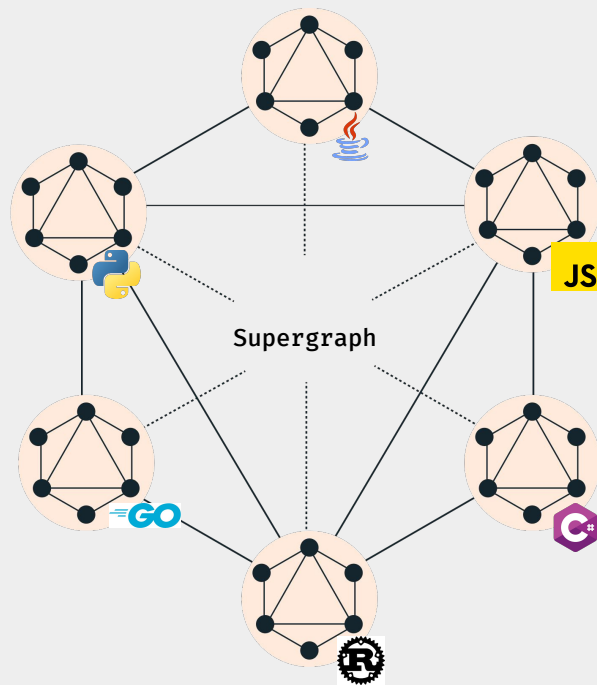




How?



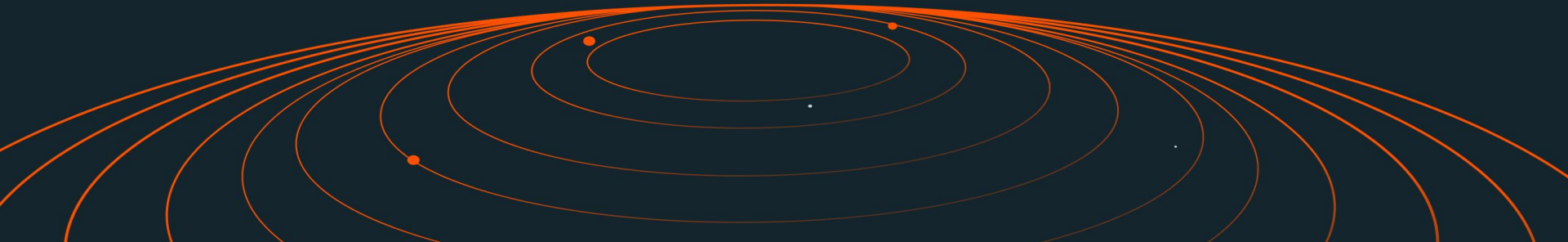
GraphQL Servers



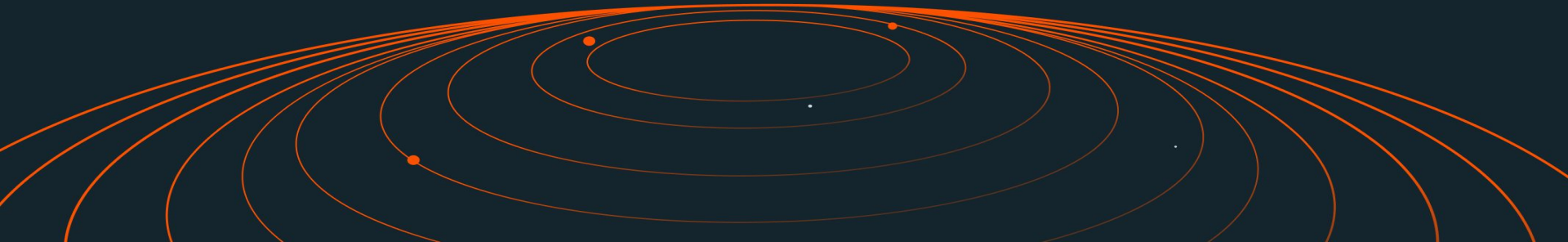
Apollo Connectors

```
type Query {  
  post(id: ID!): Post  
    @connect(  
      source: "rest_api_v1"  
      http: { GET: "/posts/{$args.id}" }  
      selection: ""  
        id  
        title  
        body  
        author: { id: userId }  
      ""  
    )  
}
```

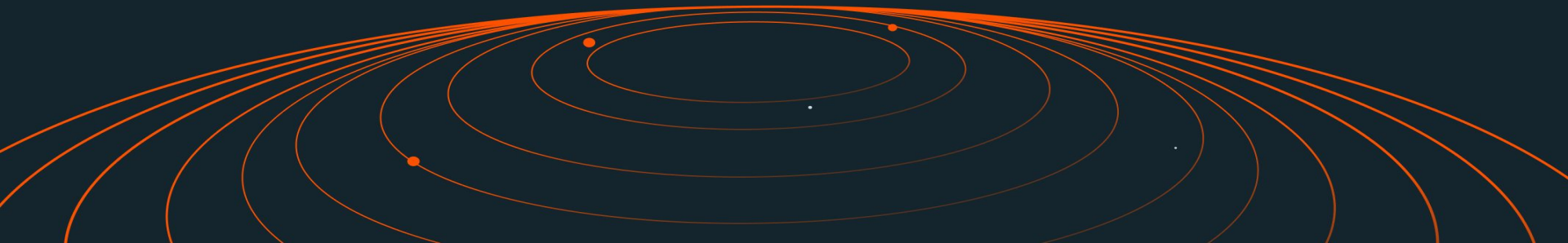
Summary



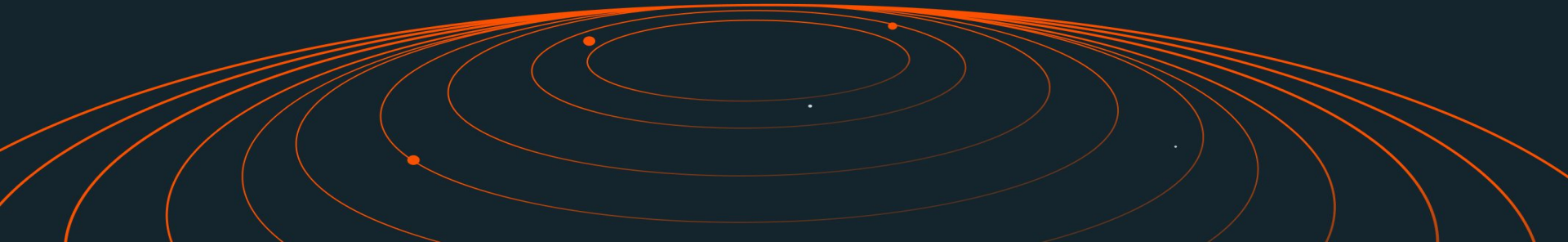
AI needs APIs



GraphQL makes your APIs **better**



**Tomorrow's AI experiences will be
build on GraphQL Federation**



- <https://www.apollographql.com/docs/graphos/schema-design/federated-schemas/federation>
- <https://www.apollographql.com/blog/apollo-connectors-for-rest-apis>
- <https://www.apollographql.com/guides/chatbots>

