OSC 源创会 Opensource Innovation Meetup | IT大咖说 知识共享平台

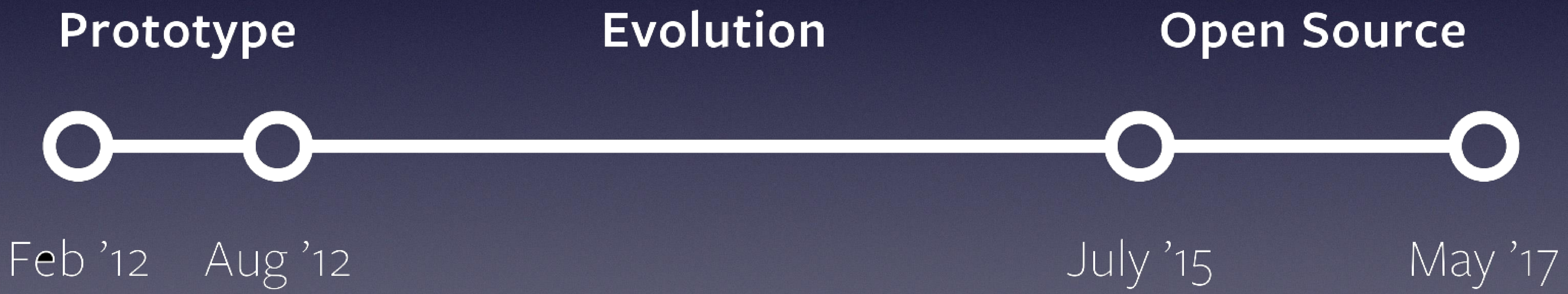
未来已来

GraphQL 在 Apache SkyWalking(Incubating)中的实践

Skywalking

GraphQL一览

History of GraphQL



Prototype

Evolution

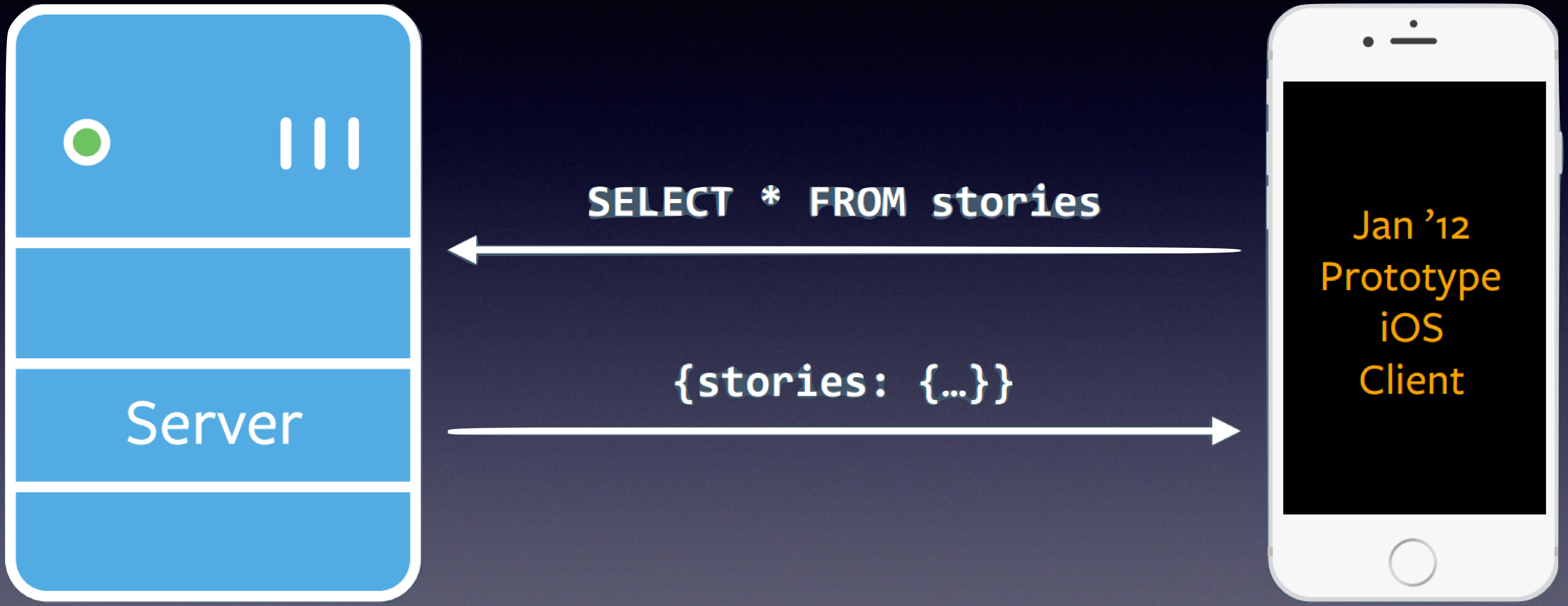
Open Source

Feb ’12

Aug ’12

July ’15

May ’17

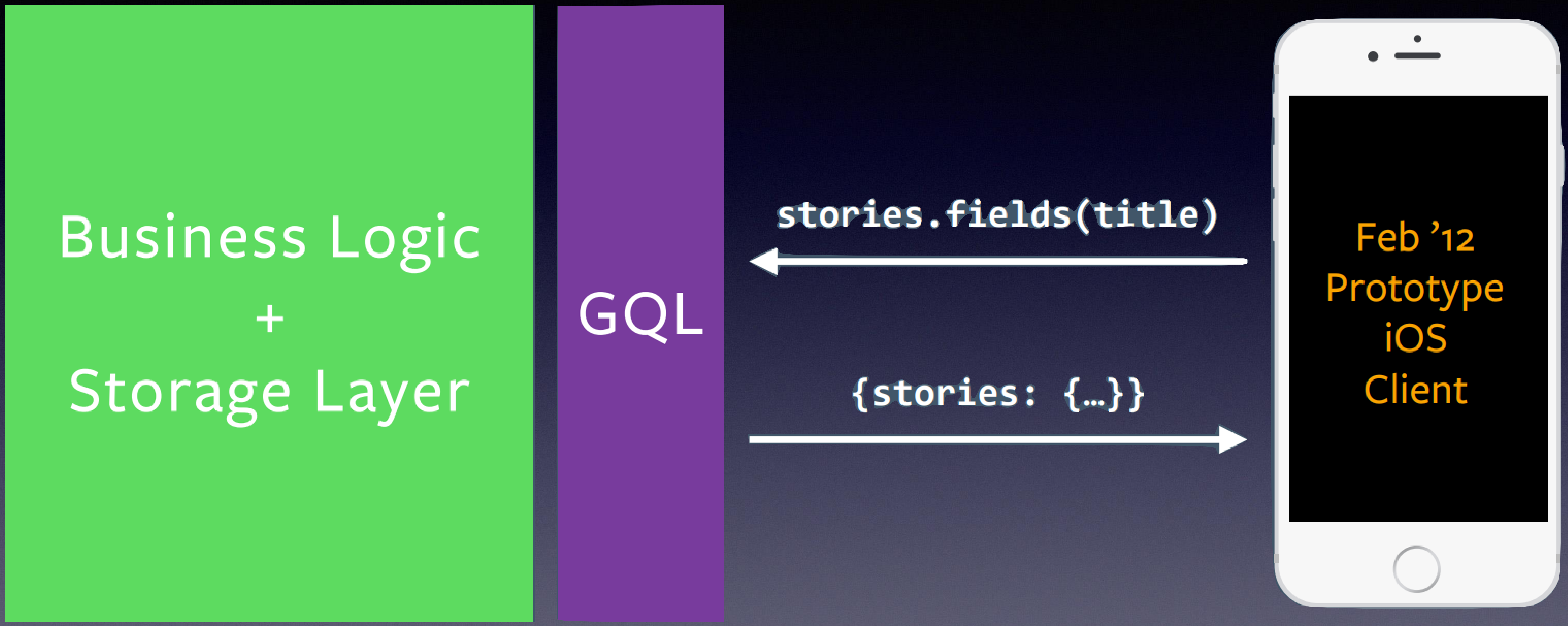


Server

```
SELECT * FROM stories
```

```
{stories: {…}}
```

Jan ’12
Prototype
iOS
Client



Business Logic
+
Storage Layer

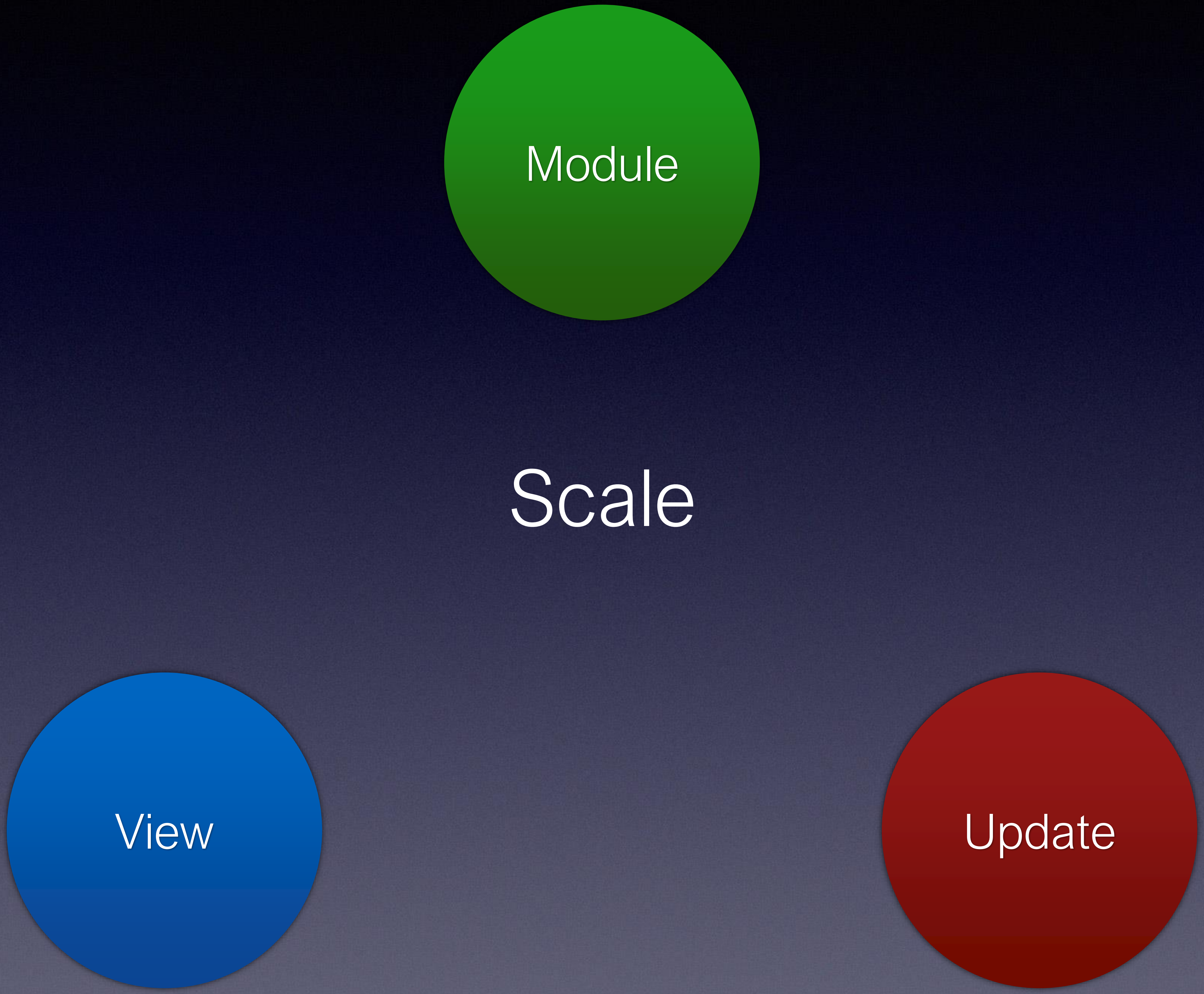
GQL

stories.fields(title)

{stories: {…}}

Feb ’12
Prototype
iOS
Client

Client-Driven



Module

Scale

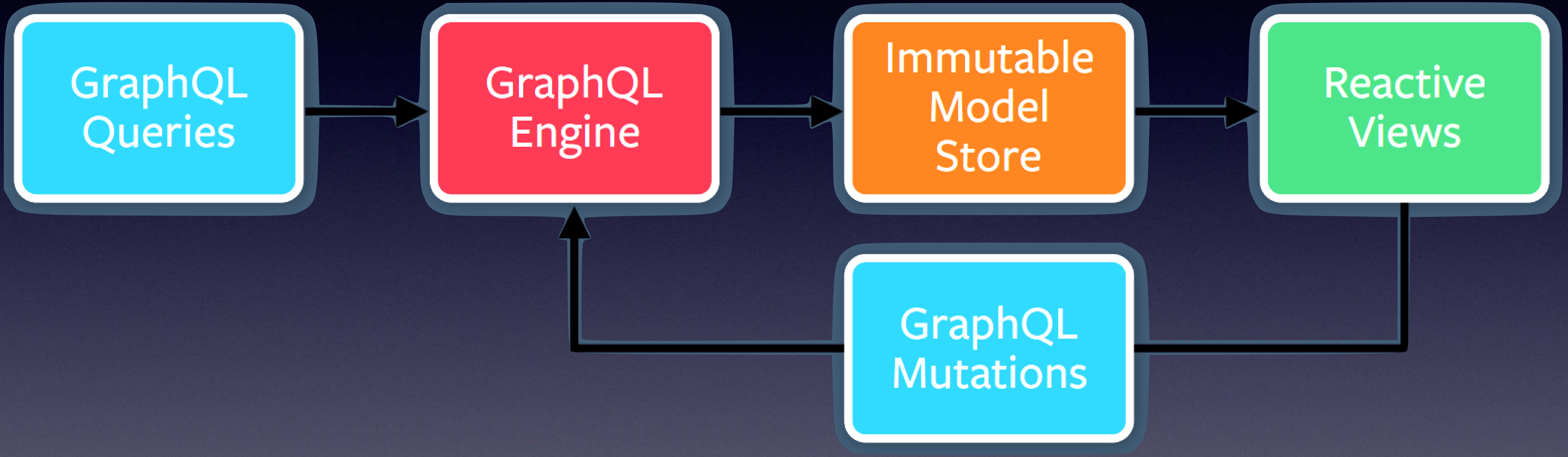
View

Update

```
query {
  me {
    name
    profilePicture {
      width
      height
      url
    }
  }
}
```

→

```
query {
  me {
    name
    ...ProfilePicture
  }
}

fragment ProfilePicture on User {
  profilePicture {
    height
    width
    uri
  }
}
```



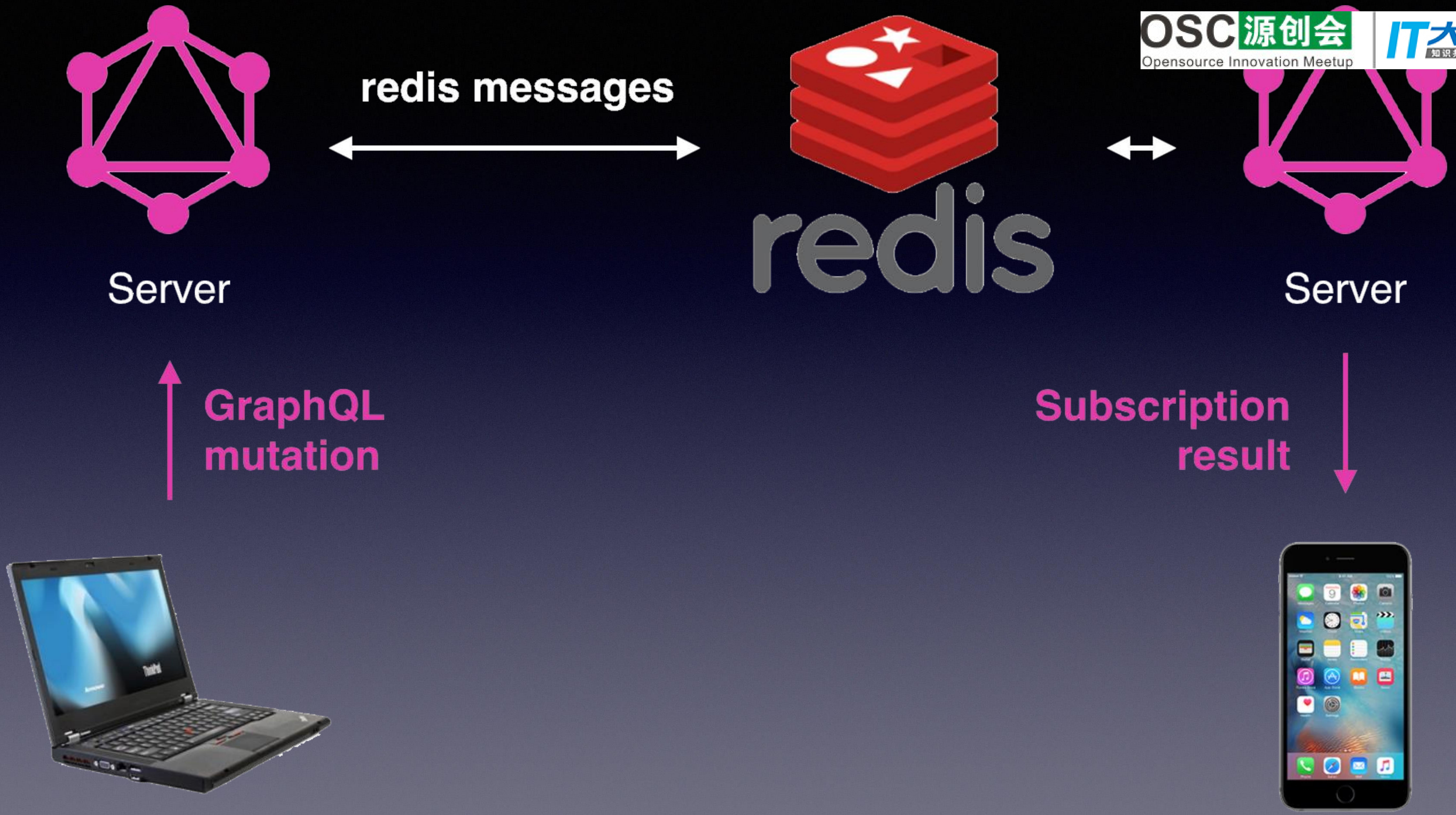
GraphQL Queries

GraphQL Engine

Immutable Model Store

Reactive Views

GraphQL Mutations



redis messages

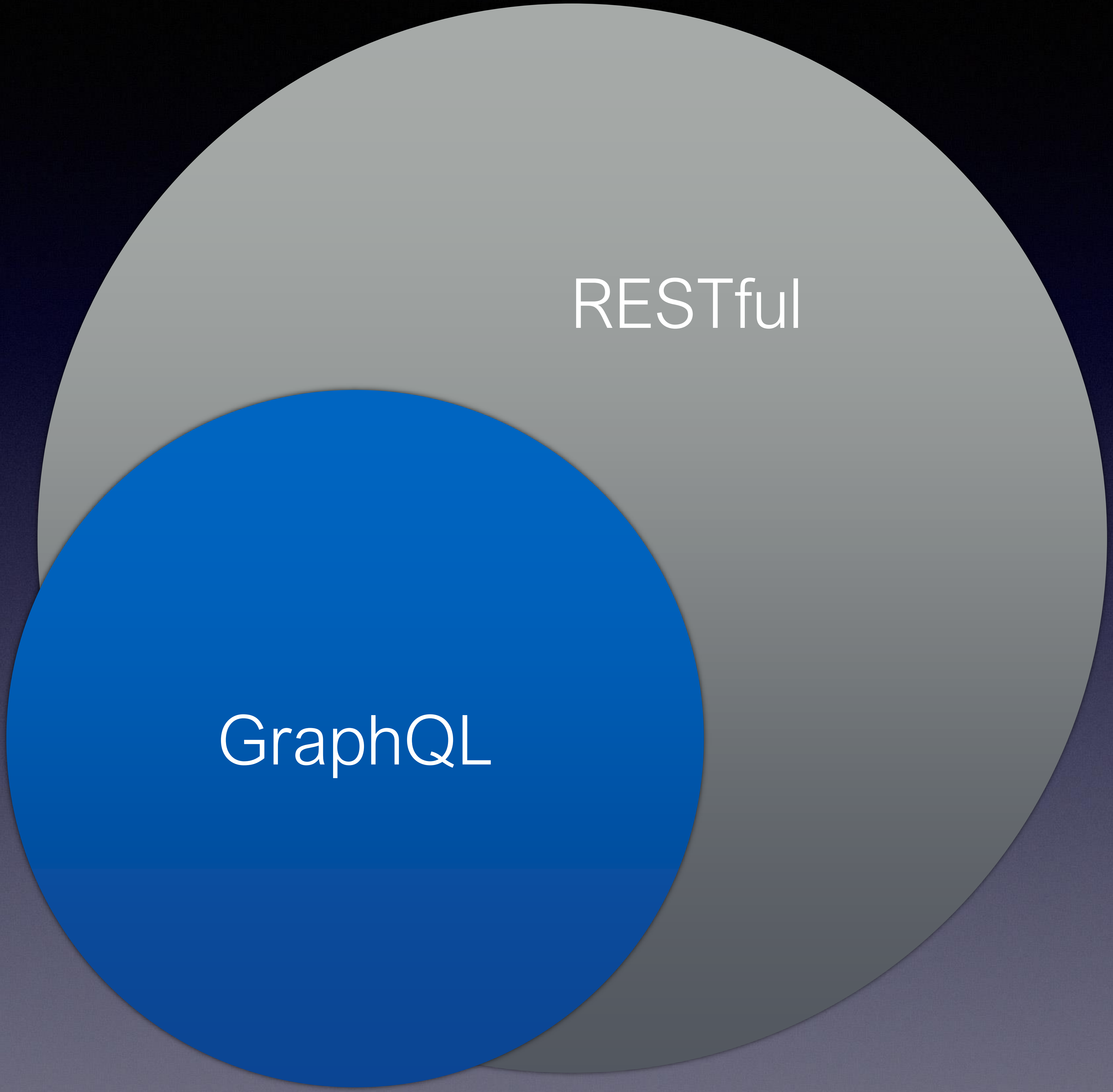
Server

Server

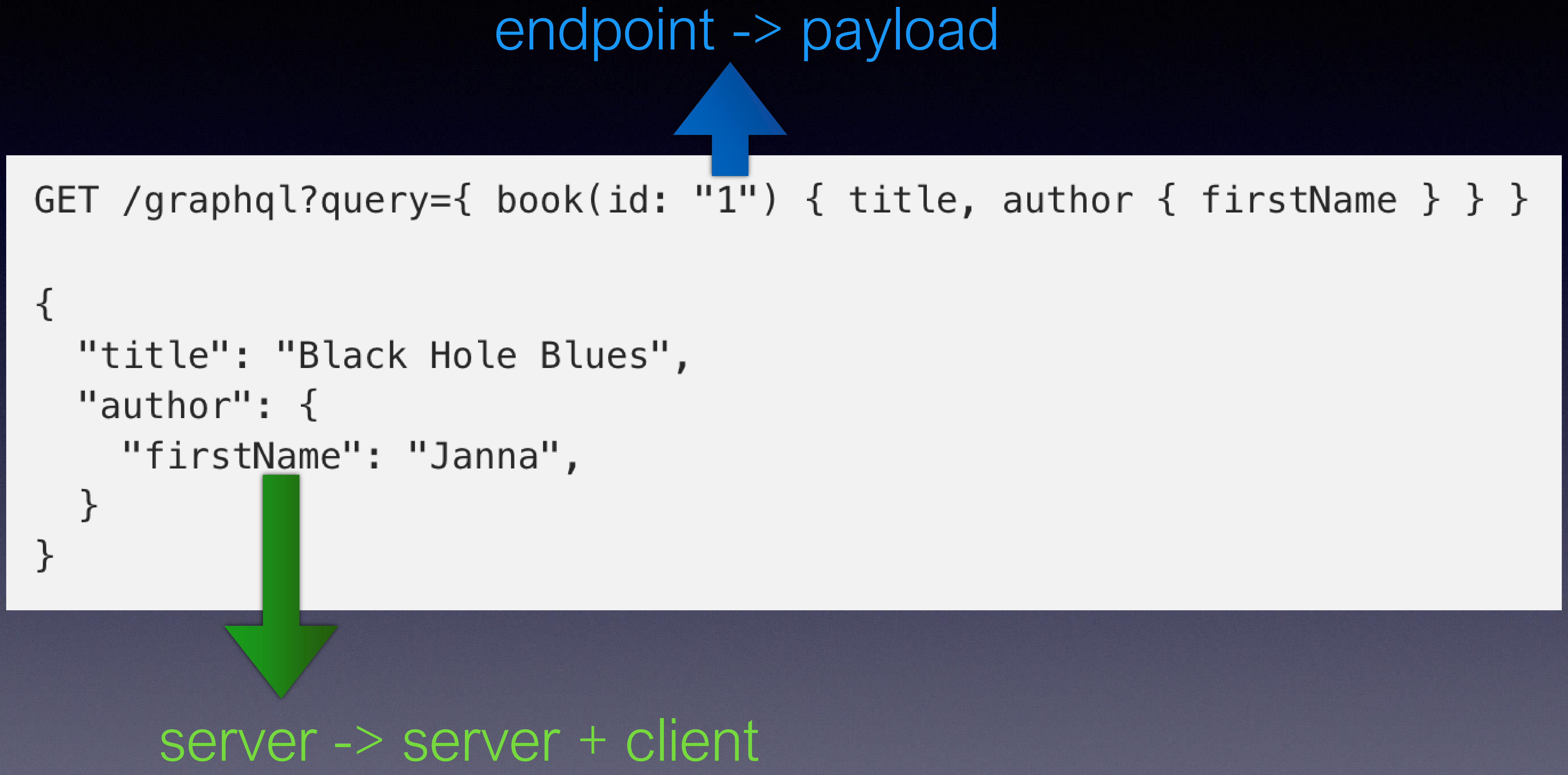
GraphQL
mutation

Subscription
result

RESTFul vs GraphQL


RESTful
GraphQL

```
GET /graphql?query={ book(id: "1") { title, author { firstName } } }

{
  "title": "Black Hole Blues",
  "author": {
    "firstName": "Janna",
  }
}
```

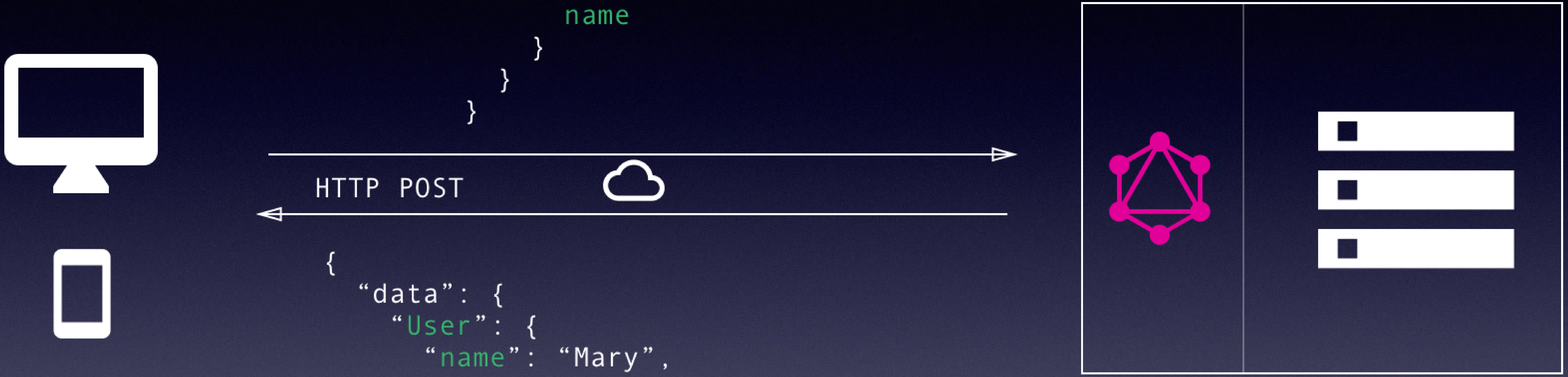


endpoint -> payload

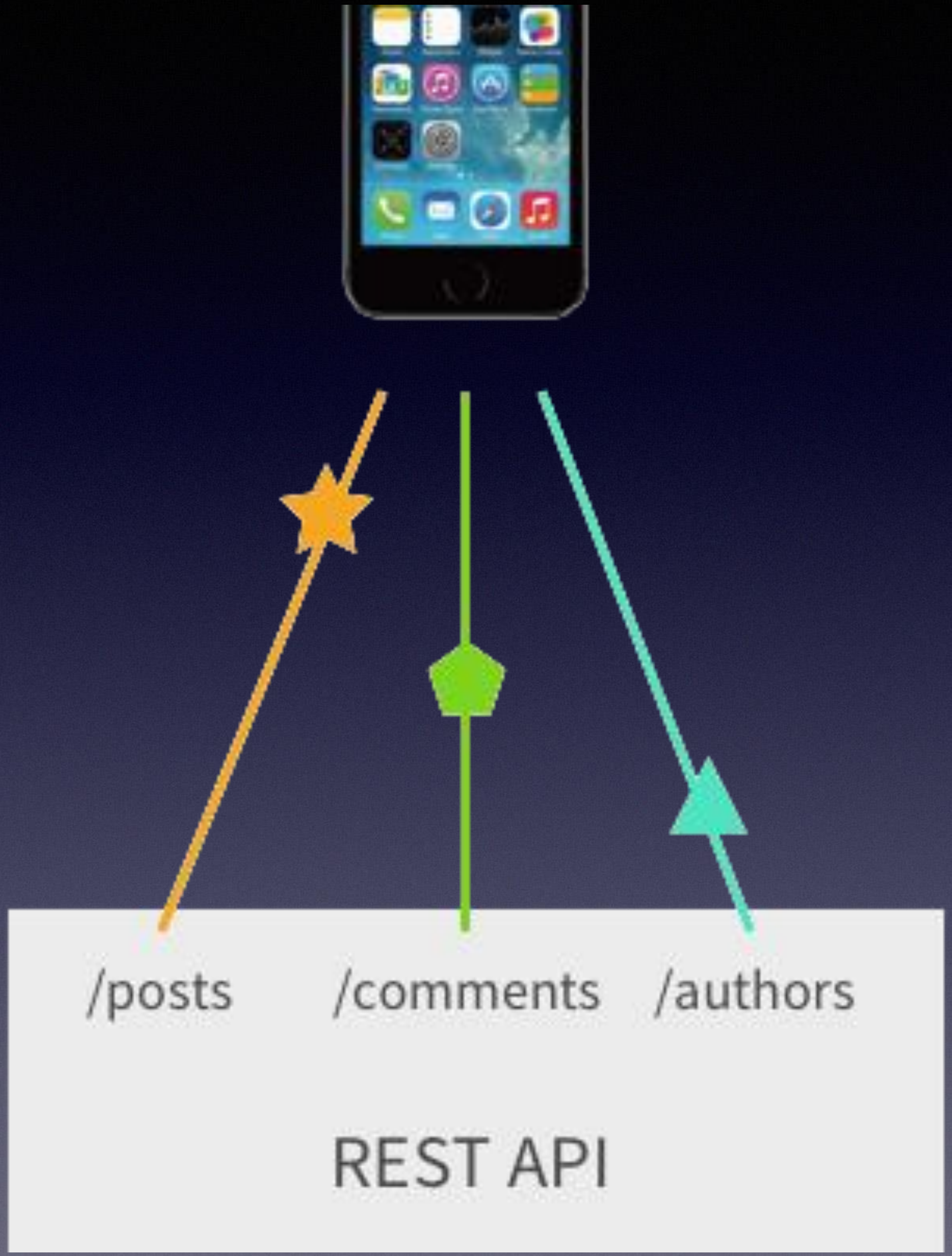
```
GET /graphql?query={ book(id: "1") { title, author { firstName } } }

{
  "title": "Black Hole Blues",
  "author": {
    "firstName": "Janna",
  }
}
```

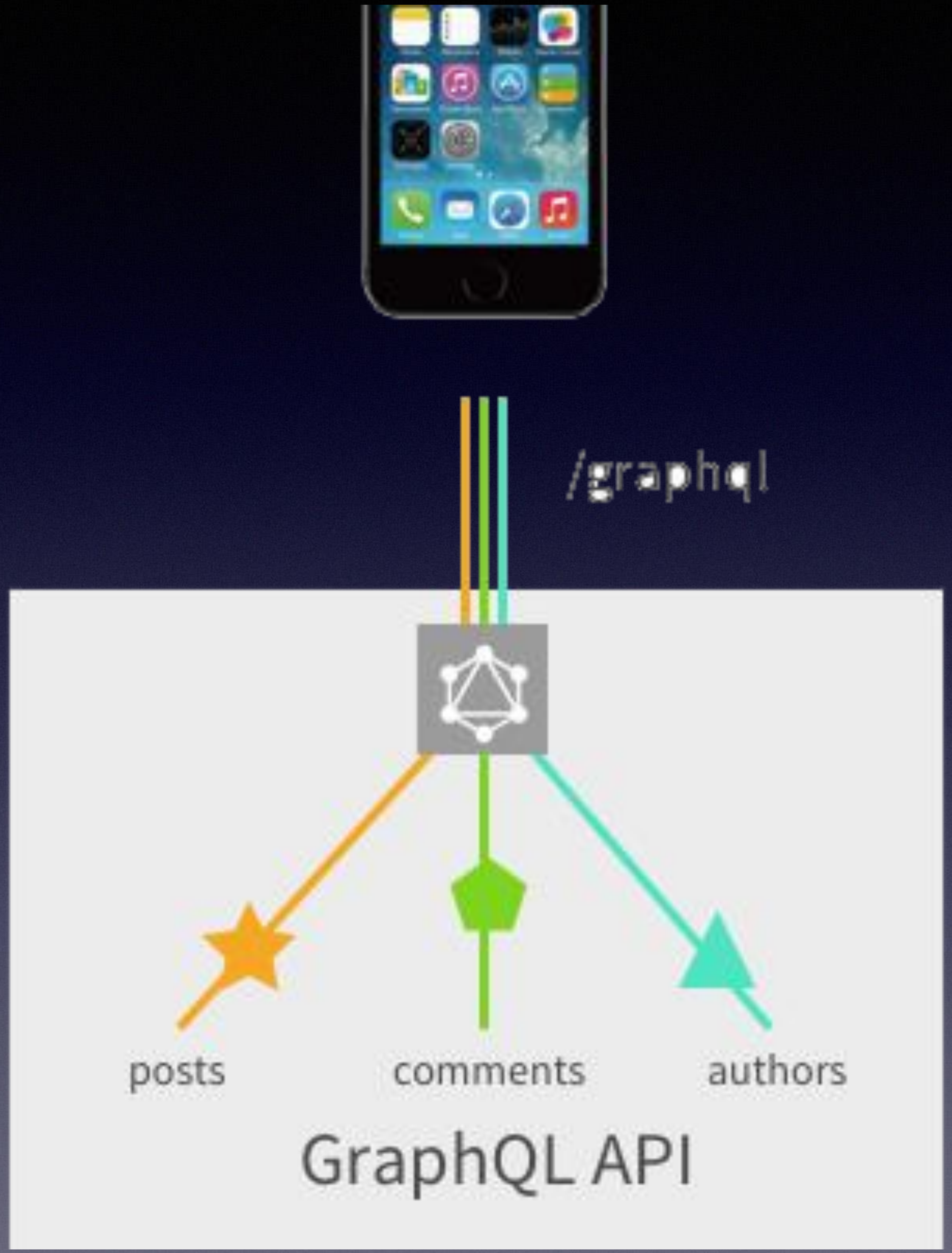
server -> server + client

```
query {
  User(id: “er3tg439frjw”) {
    name
    posts {
      title
    }
    followers(last: 3) {
      name
    }
  }
}
```



```
{
  “data”: {
    “User”: {
      “name”: “Mary”,
      “posts”: [
        { title: “Learn GraphQL today” }
      ],
      “followers”: [
        { name: “John” },
        { name: “Alice” },
        { name: “Sarah” },
      ]
    }
  }
}
```


/posts
/comments
/authors
REST API

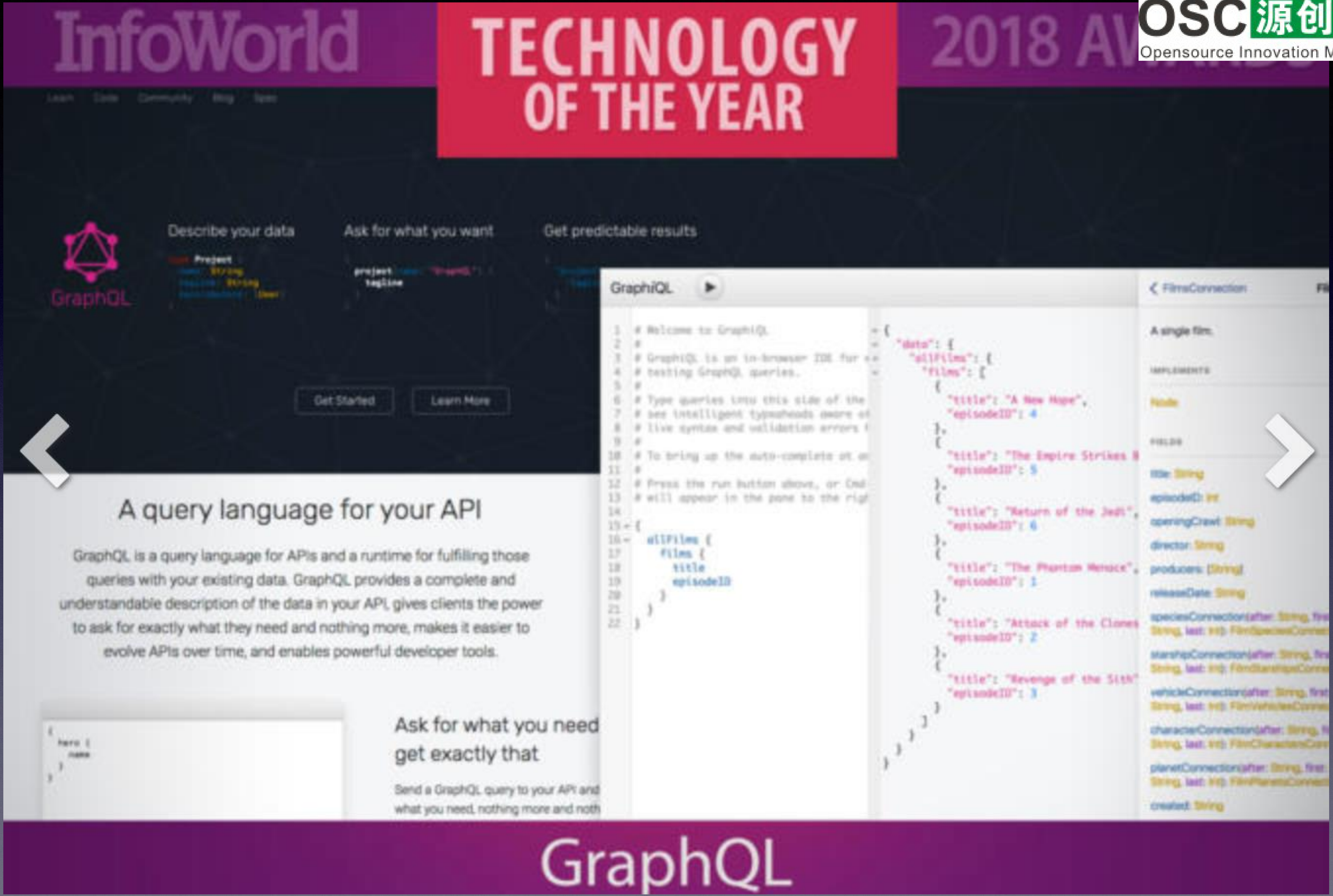

/graphql
posts
comments
authors
GraphQL API

未来已来

Java
Ruby
Microsoft .NET
Clojure
JS
Swift
TypeScript
php
elixir
Scala

aws

GitHub



A query language for your API

GraphQL is a query language for APIs and a runtime for fulfilling those queries with your existing data. GraphQL provides a complete and understandable description of the data in your API, gives clients the power to ask for exactly what they need and nothing more, makes it easier to evolve APIs over time, and enables powerful developer tools.

```
{
  hero {
    name
  }
}
```

Ask for what you need get exactly that

Send a GraphQL query to your API and what you need, nothing more and noth

```
# Welcome to GraphiQL
#
# GraphiQL is an in-browser IDE for w
# testing GraphQL queries.
#
# Type queries into this side of the
# see intelligent typeaheads aware of
# live syntax and validation errors f
#
# To bring up the auto-complete at an
#
# Press the run button above, or Cmd
# will appear in the pane to the righ

{
  allFilms {
    films {
      title
      episodeID
    }
  }
}
```

```
{
  "data": {
    "allFilms": {
      "films": [
        {
          "title": "A New Hope",
          "episodeID": 4
        },
        {
          "title": "The Empire Strikes B
          "episodeID": 5
        },
        {
          "title": "Return of the Jedi",
          "episodeID": 6
        },
        {
          "title": "The Phantom Menace",
          "episodeID": 1
        },
        {
          "title": "Attack of the Clones
          "episodeID": 2
        },
        {
          "title": "Revenge of the Sith"
          "episodeID": 3
        }
      ]
    }
  }
}
```

A single film.

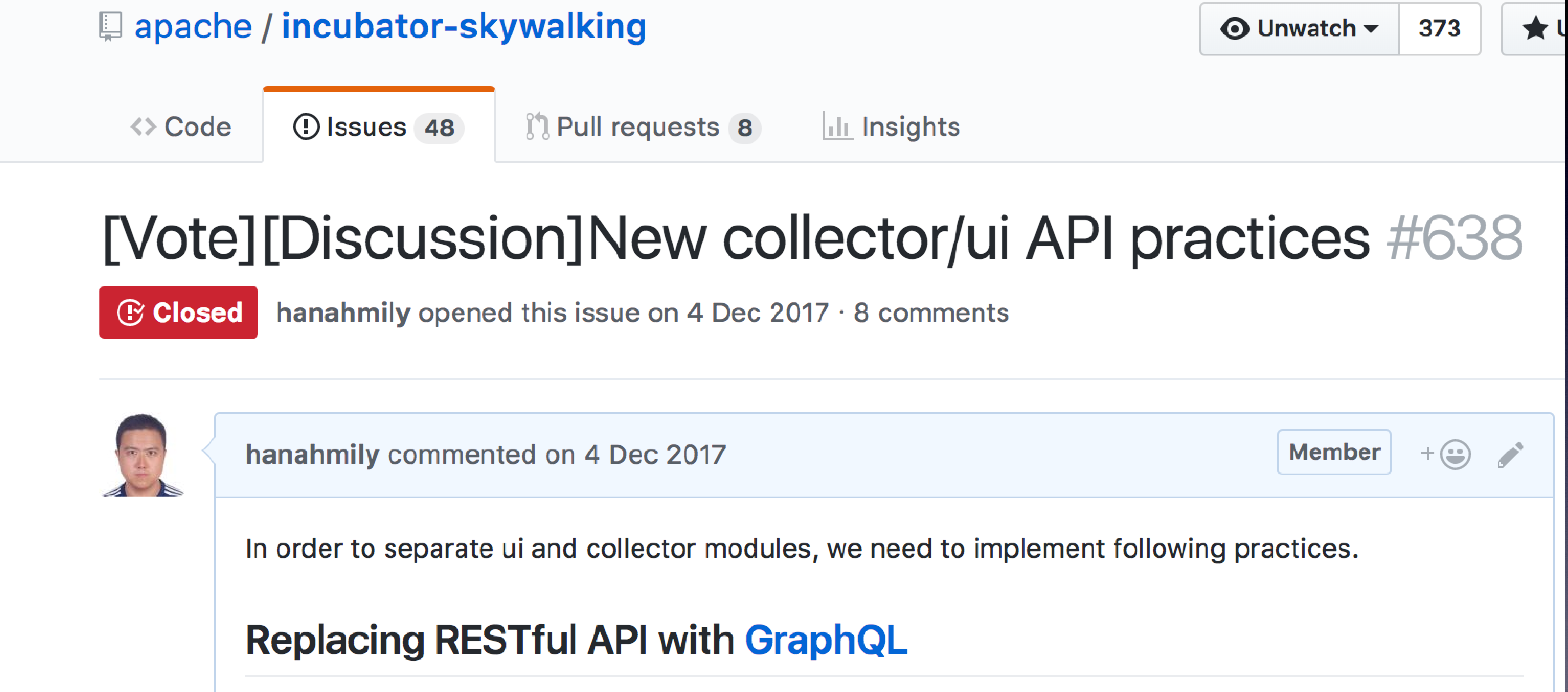
IMPLEMENTS

Node

FIELDS

title: String
episodeID: Int
openingCrawl: String
director: String
producers: [String]
releaseDate: String
speciesConnection(after: String, first
String, last: Int): FilmSpeciesConnec
starshipConnection(after: String, firs
String, last: Int): FilmStarshipsConne
vehicleConnection(after: String, first
String, last: Int): FilmVehiclesConnec
characterConnection(after: String, fi
String, last: Int): FilmCharactersCon
planetConnection(after: String, first:
String, last: Int): FilmPlanetsConnect
created: String

GraphQL in SkyWalking



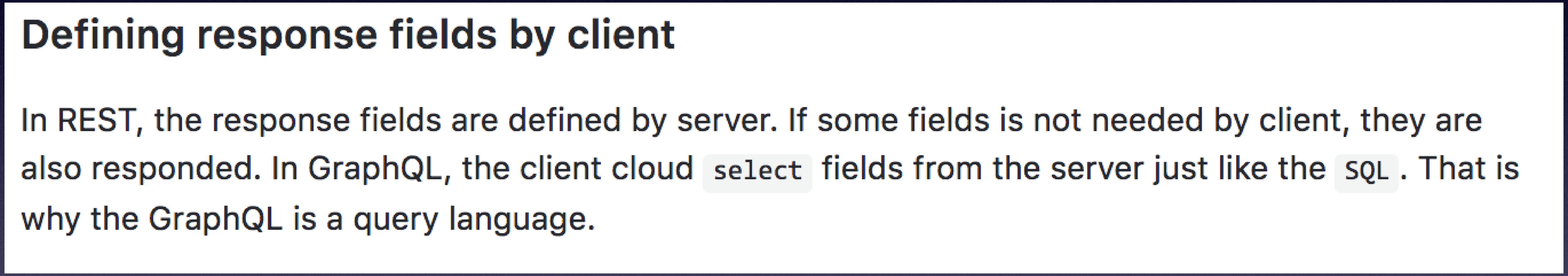
[Vote][Discussion]New collector/ui API practices #638

hanahmily opened this issue on 4 Dec 2017 · 8 comments

hanahmily commented on 4 Dec 2017

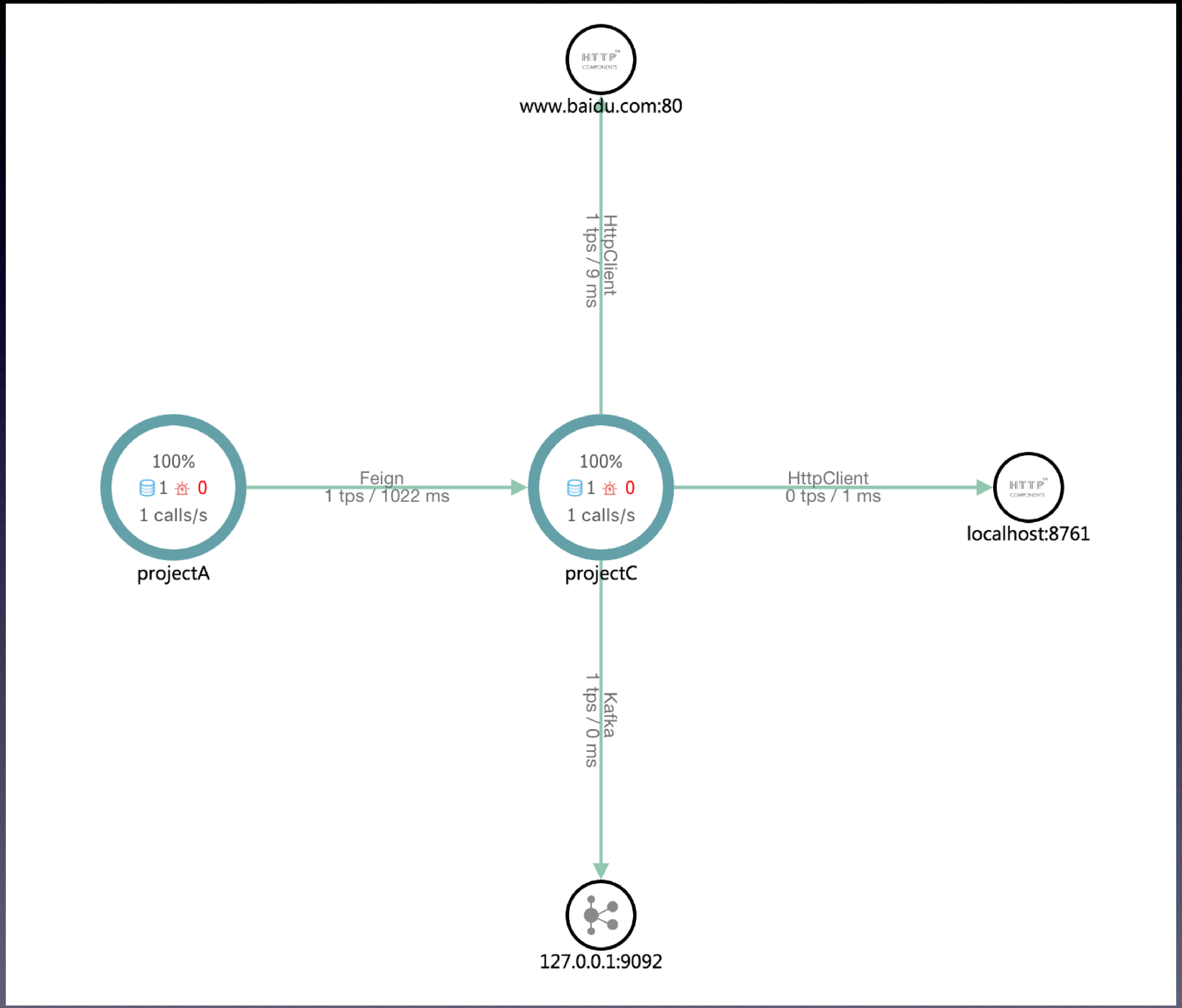
In order to separate ui and collector modules, we need to implement following practices.

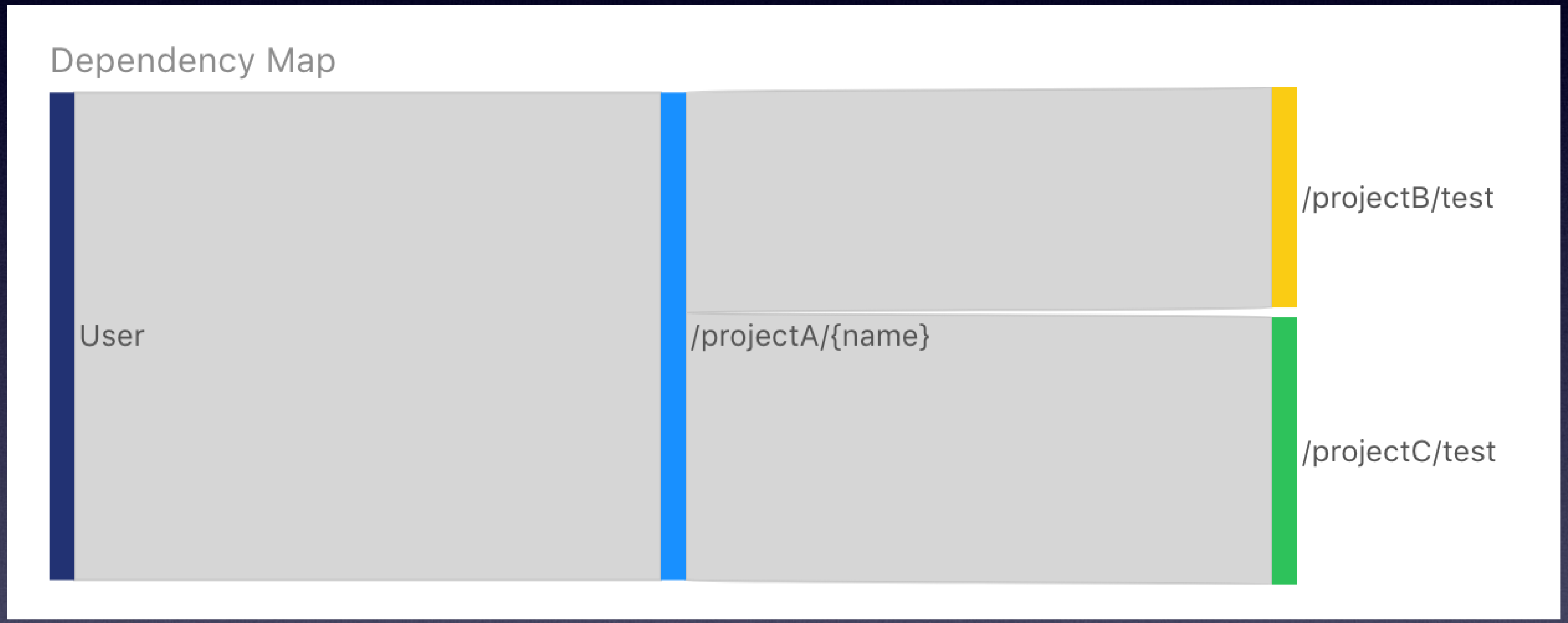
Replacing RESTful API with GraphQL

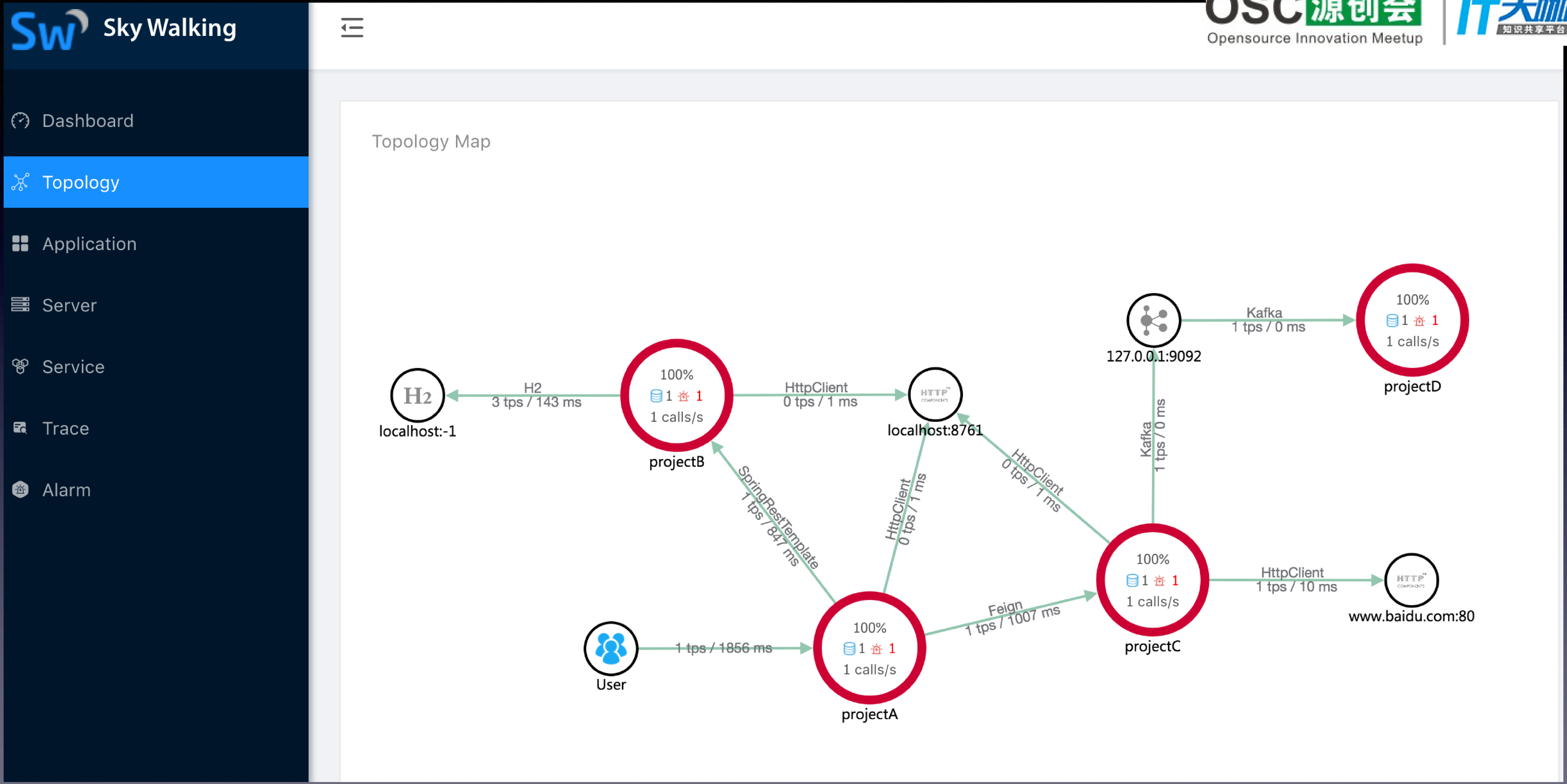


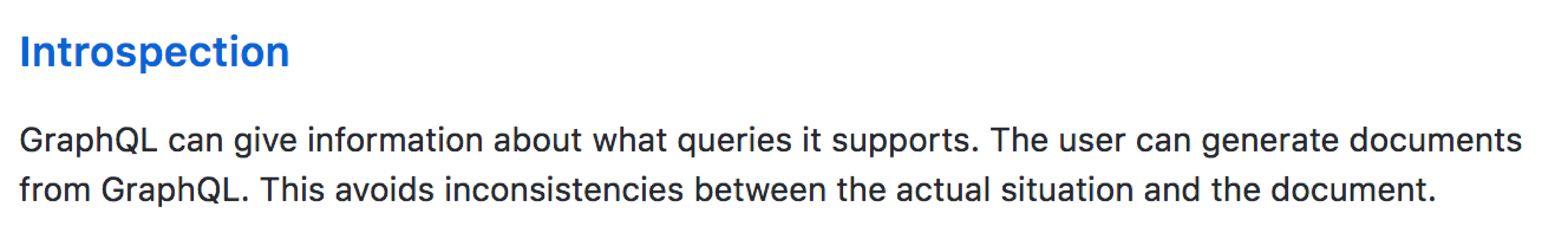
Defining response fields by client

In REST, the response fields are defined by server. If some fields is not needed by client, they are also responded. In GraphQL, the client cloud `select` fields from the server just like the `SQL`. That is why the GraphQL is a query language.


www.baidu.com:80
HttpClient
1 tps / 9 ms
100%
1 0
1 calls/s
projectA
Feign
1 tps / 1022 ms
100%
1 0
1 calls/s
projectC
HttpClient
0 tps / 1 ms
localhost:8761
Kafka
1 tps / 0 ms
127.0.0.1:9092

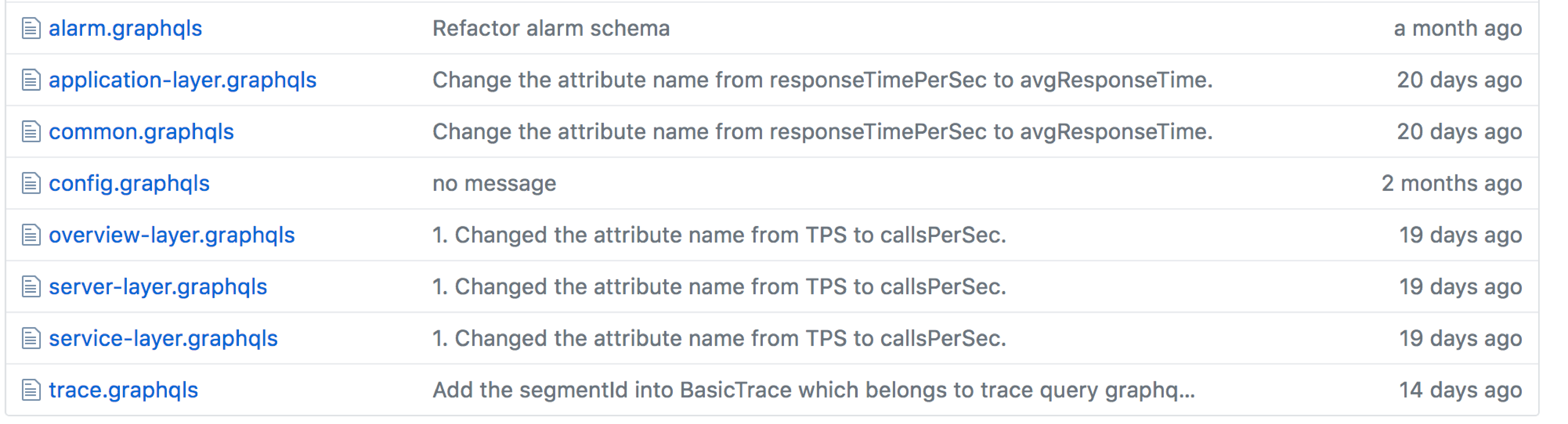

Dependency Map
User
/projectA/{name}
/projectB/test
/projectC/test


Sky Walking
Dashboard
Topology
Application
Server
Service
Trace
Alarm
Topology Map
localhost:-1
H2
3 tps / 143 ms
100%
1 1
1 calls/s
projectB
HttpClient
0 tps / 1 ms
localhost:8761
127.0.0.1:9092
Kafka
1 tps / 0 ms
100%
1 1
1 calls/s
projectD
Kafka
1 tps / 0 ms
SpringRestTemplate
1 tps / 847 ms
HttpClient
0 tps / 1 ms
HttpClient
0 tps / 1 ms
100%
1 1
1 calls/s
projectC
HttpClient
1 tps / 10 ms
www.baidu.com:80
User
1 tps / 1856 ms
100%
1 1
1 calls/s
projectA
Feign
1 tps / 1007 ms

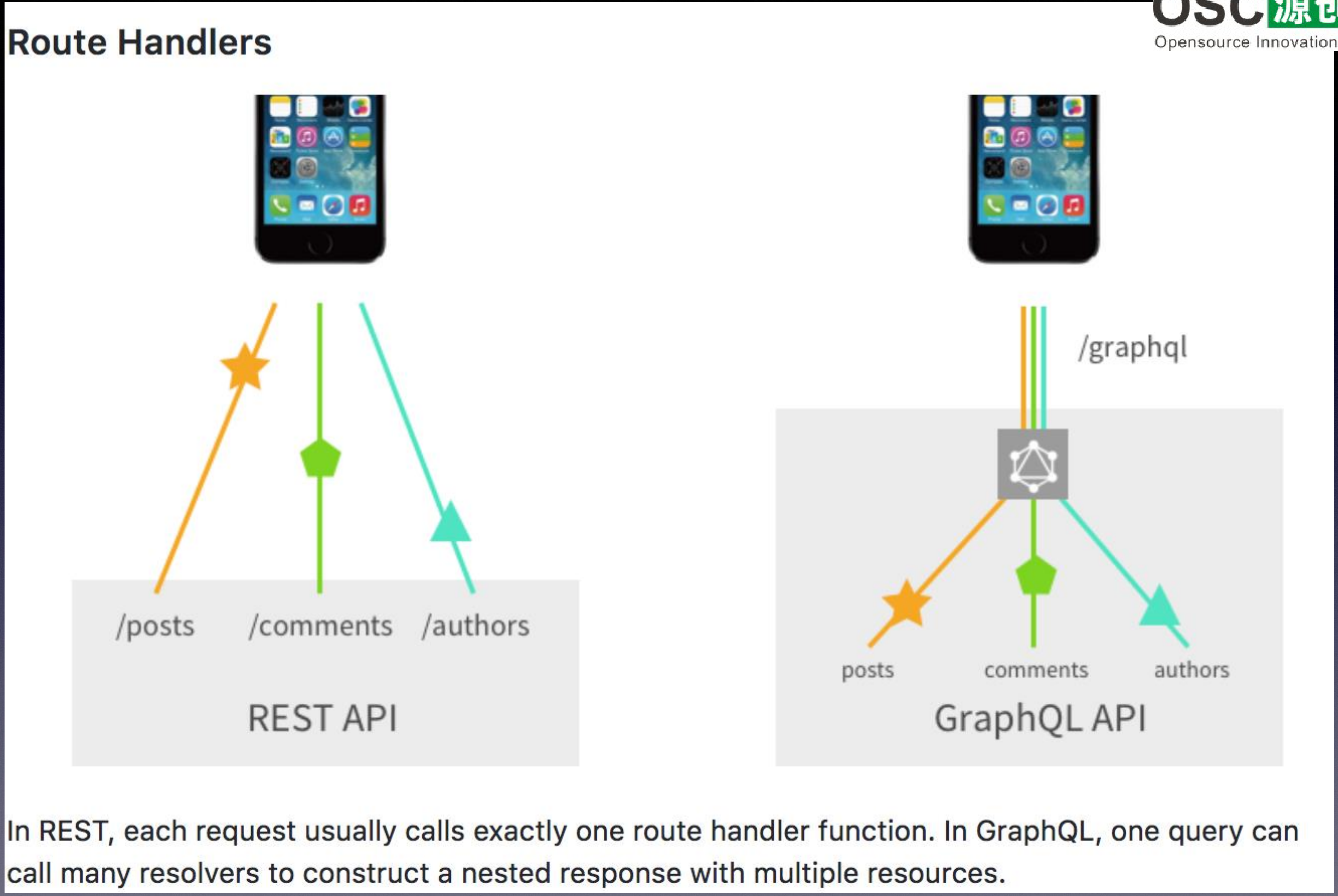


Introspection

GraphQL can give information about what queries it supports. The user can generate documents from GraphQL. This avoids inconsistencies between the actual situation and the document.

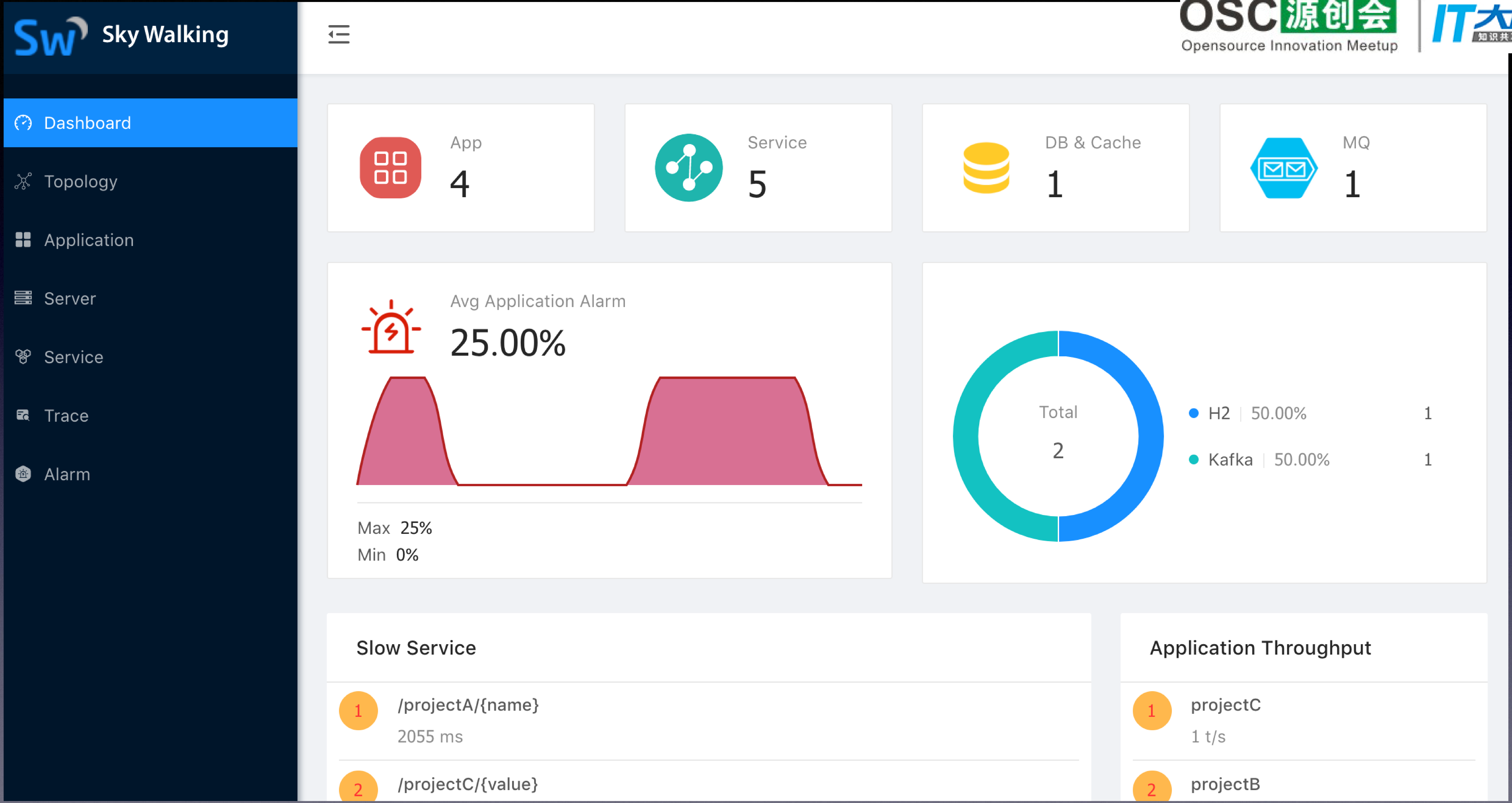


alarm.graphqls	Refactor alarm schema	a month ago
application-layer.graphqls	Change the attribute name from responseTimePerSec to avgResponseTime.	20 days ago
common.graphqls	Change the attribute name from responseTimePerSec to avgResponseTime.	20 days ago
config.graphqls	no message	2 months ago
overview-layer.graphqls	1. Changed the attribute name from TPS to callsPerSec.	19 days ago
server-layer.graphqls	1. Changed the attribute name from TPS to callsPerSec.	19 days ago
service-layer.graphqls	1. Changed the attribute name from TPS to callsPerSec.	19 days ago
trace.graphqls	Add the segmentId into BasicTrace which belongs to trace query graphq...	14 days ago



Route Handlers

In REST, each request usually calls exactly one route handler function. In GraphQL, one query can call many resolvers to construct a nested response with multiple resources.



App 4

Service 5

DB & Cache 1

MQ 1

Avg Application Alarm

25.00%

Max 25%
Min 0%

Total 2

● H2	50.00%	1
● Kafka	50.00%	1

Slow Service

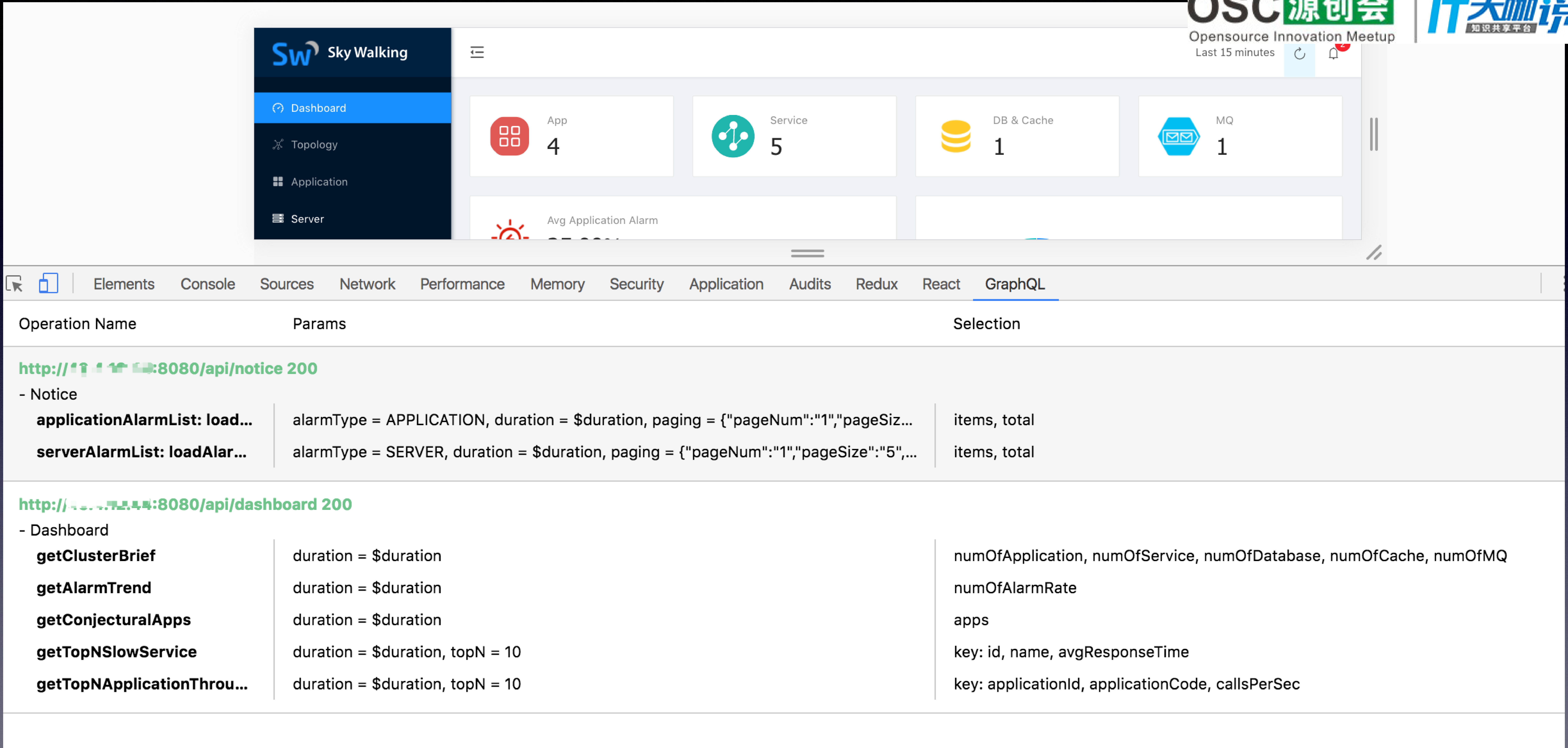
1 /projectA/{name}
2055 ms

2 /projectC/{value}

Application Throughput

1 projectC
1 t/s

2 projectB



App 4

Service 5

DB & Cache 1

MQ 1

Avg Application Alarm

Operation Name	Params	Selection
http://[redacted]:8080/api/notice 200		
- Notice		
applicationAlarmList: load...	alarmType = APPLICATION, duration = $duration, paging = {"pageNum":"1","pageSiz...	items, total
serverAlarmList: loadAlar...	alarmType = SERVER, duration = $duration, paging = {"pageNum":"1","pageSize":"5",...	items, total
http://[redacted]:8080/api/dashboard 200		
- Dashboard		
getClusterBrief	duration = $duration	numOfApplication, numOfService, numOfDatabase, numOfCache, numOfMQ
getAlarmTrend	duration = $duration	numOfAlarmRate
getConjecturalApps	duration = $duration	apps
getTopNSlowService	duration = $duration, topN = 10	key: id, name, avgResponseTime
getTopNApplicationThrou...	duration = $duration, topN = 10	key: applicationId, applicationCode, callsPerSec

最佳实践

1. Single URI
2. Null or Default value
3. Reducing chattiness
4. Minimizing failures