GraphQL and Falcor

New ways for client server communication

Andi Marek

@andimarek

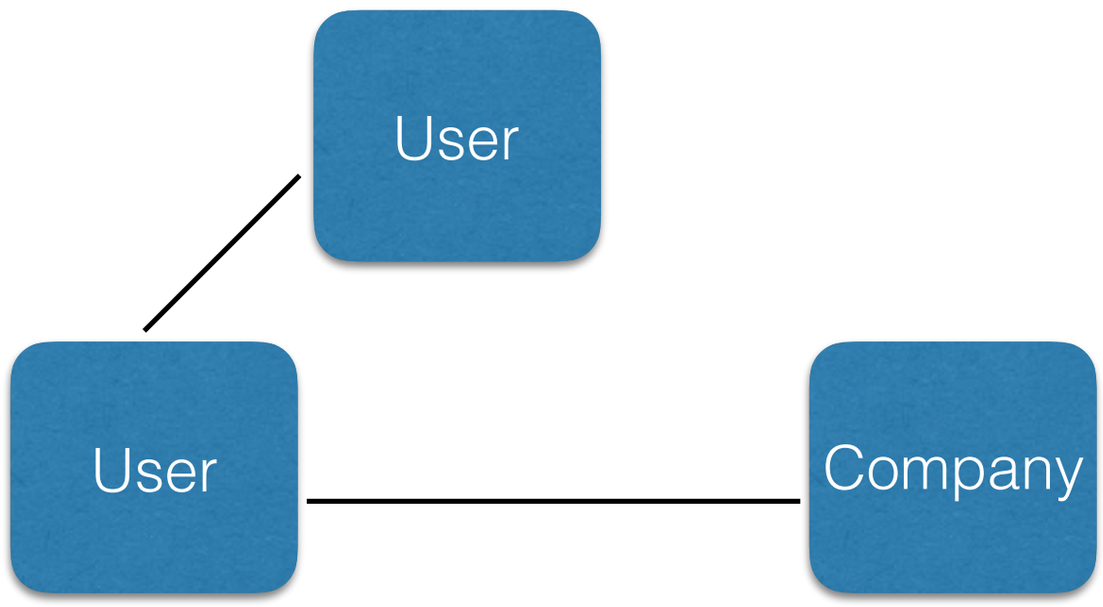
/users?fields=id,name,loginname,\
manager(loginname),\
company(domain,userCount)


```
{
   "id": 1,
   "name": "John",
   "loginname": "john@example.com",
   "manager" : {
      "loginname": "dave@example.com"
   },
   "company" : {
      "domain": "example.com",
      "userCount": "42"
   }
}
```


GraphQL


FALCOR

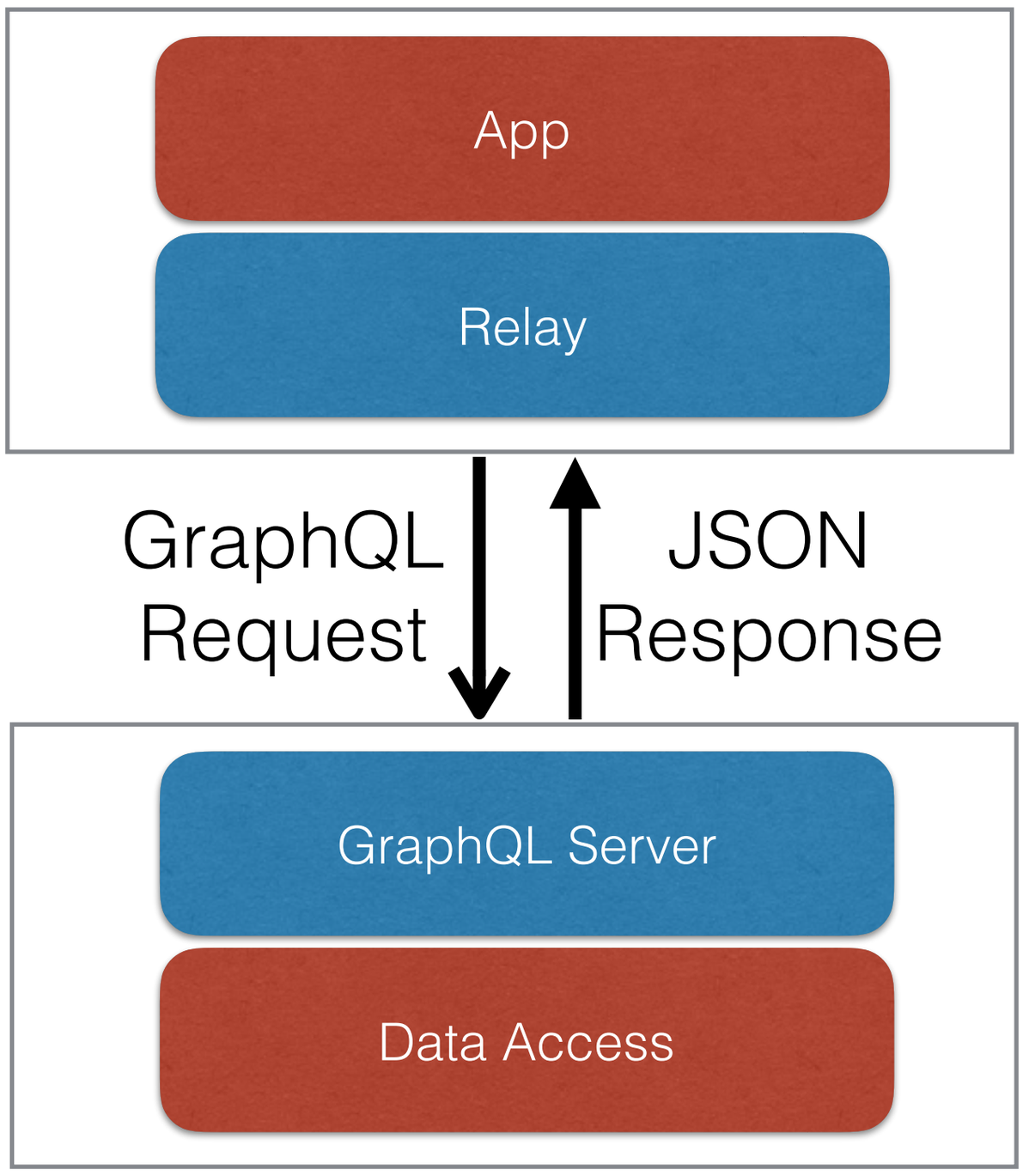
The domain model is a graph



What about REST?

- Invented by Facebook
- Internally used since 2012
- Open Source version published Mid 2015
- Relay released August 2015

GraphQL/Relay Architecture



GraphQL Example

Request

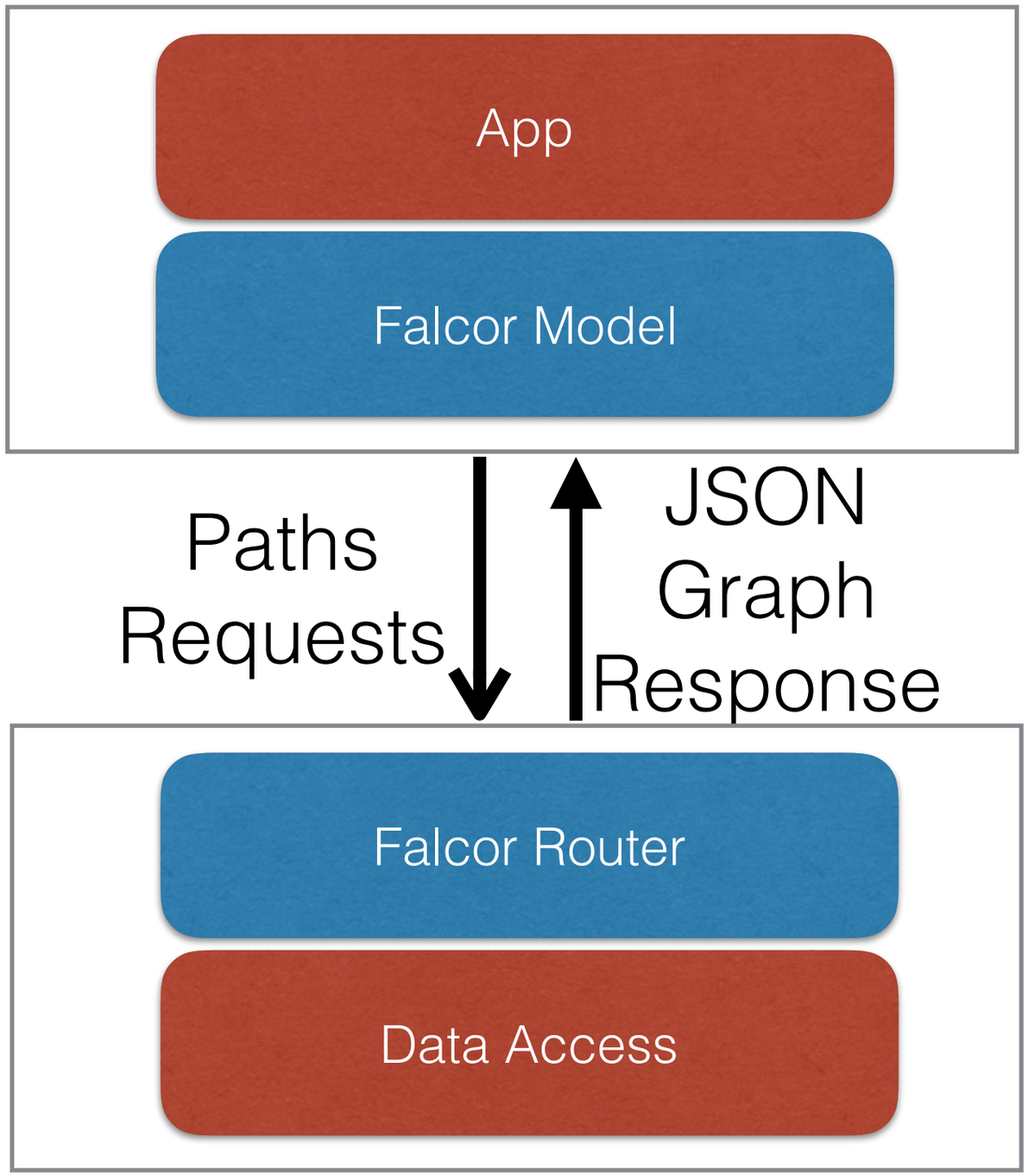
```
{
  user(id: 1) {
    id,
    name,
    loginname,
    company{
     domain,
     userCount
    }
  }
}
```

Response

```
{
  "user" : {
    "id": 1,
    "name": "John",
    "loginname": "john@example.com",
    "company" : {
      "domain": "example.com",
      "userCount": "42"
    }
}
```

- Invented by Netflix
- Internally developed since 2012
- Released August 2015

Falcor Architecture



Falcor Example

Request

```
user[1]['name','loginname']

user[1]['company']
  ['domain','userCount']
```

Response

```
{
  "user": {
    "1": {
      "name": "John",
      "loginname": "john@example.com",
      "company": {
        "domain": "example.com",
        "userCount": 42
      }
    }
  }
}
```

Similarities and differences

Caching the graph

In-browser cache is a central element of Relay and Falcor.

Both support optimised queries for minimal data flow.

Client is king

Client can choose very flexible which data it requests.

The Graph should be/needs to be designed for the client requirements.

Explicit data

GraphQL and Falcor require both to declare all properties that are needed.

You never get more.

Explicit dependencies.

Statically Typed vs Untyped

GraphQL requires a static schema to describe the data.

Falcor doesn’t have a schema.

DSL vs “Data is the API”

GraphQL is a DSL for querying graph data.

With Falcor the graph is directly modified with paths.

Trade Off: Flexible Language vs simple API

GraphQL: Request specific user by loginname

```
{
  user(loginname: 'john@example.com') {
    id,
    name,
    loginname,
    company{
      domain,
      userCount
   }
  }
}
```

```
{
  "user" : {
    "id": 1,
    "name": "John",
    "loginname": "john@example.com",
    "company" : {
      "domain": "example.com",
      "userCount": "42"
    }
}
```

Falcor: Request specific user by loginname

```
userByLoginname['john@example.com']['id','name']

userByLoginname['john@example.com']['company']
['domain','userCount']
```


```
{
  "userByLoginname": {
    "john@example.com": {
      "id": "1",
      "name": "John",
      "company": {
        "domain": "example.com",
        "userCount": 42
      }
    }
  }
}
```

Reading is the strong side

Both solutions are mainly designed for read purposes.

Write examples are rare and not very well documented.

Both support optimistic updates and RPC Calls.

Falcor’s support of simple updates is very elegant.

Ecosystem

Both have active ecosystems. GraphQL/Relay seems bigger.

GraphQL Server alternatives are available (Ruby, Java etc).

Alternative Falcor Servers (Routers) are rare.

Integration

Relay currently only for React

Falcor doesn’t have any UI Dependency

No persistence dependency

Thanks

http://graphql.org/

http://netflix.github.io/falcor/

twitter.com/andimarek