



GraphQL

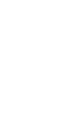
**A query language for
your API.**

**Declarative data
fetching.**

**Single endpoint,
multiple resources.**

Benefits of GraphQL.

- No under-fetching or over-fetching.
- No need to structure endpoint according to client.
- Easier to evolve API, deprecate unneeded features.

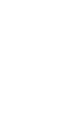


Benefits of GraphQL.

- Schema creates contract between client / server.
- Easy to integrate with existing systems.
- Subscriptions for real-time data updates.
- Client and Server caching.



Example.



App using Rest.

3 API endpoints

`/users/<id>`

`/users/<id>/posts`

`/users/<id>/followers`



① Fetch user data

Mary

Mary's posts:

Last three followers:



HTTP GET

/users/<id>

/users/<id>/posts

/users/<id>/followers



```
{  
  "user": {  
    "id": "er3tg439frjw"  
    "name": "Mary",  
    "address": { ... },  
    "birthday": "July 26, 1982"  
  }  
}
```



② Fetch posts

Mary

Mary's posts:

Learn GraphQL Today

React & GraphQL - A declarative love story

Why GraphQL is better than REST

Relay vs Apollo - GraphQL clients

Last three followers:



HTTP GET

```
{
  "posts": [{
    "id": "ncwon3ce89hs",
    "title": "Learn GraphQL today",
    "content": "Lorem ipsum ... ",
    "comments": [ ... ],
  }, {
    "id": "dsifr3as0vds",
    "title": "React & GraphQL - A declarative love story",
    "content": "Lorem ipsum ... ",
    "comments": [ ... ],
  }, {
    "id": "die5odnvls1o",
    "title": "Why GraphQL is better than REST",
    "content": "Lorem ipsum ... ",
    "comments": [ ... ],
  }, {
    "id": "dovmdr3nvl8f",
    "title": "Relay vs Apollo - GraphQL clients",
    "content": "Lorem ipsum ... ",
    "comments": [ ... ],
  }
]
```

/users/<id>

/users/<id>/posts

/users/<id>/followers



③ Fetch followers

Mary

Mary's posts:

Learn GraphQL Today

React & GraphQL - A declarative love story

Why GraphQL is better than REST

Relay vs Apollo - GraphQL clients

Last three followers:

John, Alice, Sarah



HTTP GET

```
{
  "followers": [{
    "id": "leo83h2dojsu"
    "name": "John",
    "address": { ... },
    "birthday": "January 6, 1970"
  }, {
    "id": "die5odnvls1o"
    "name": "Alice",
    "address": { ... },
    "birthday": "May 1, 1989"
  } {
    "id": "xsifr3as0vds"
    "name": "Sarah",
    "address": { ... },
    "birthday": "November 20, 1986"
  }
  ...
]
```

/users/<id>

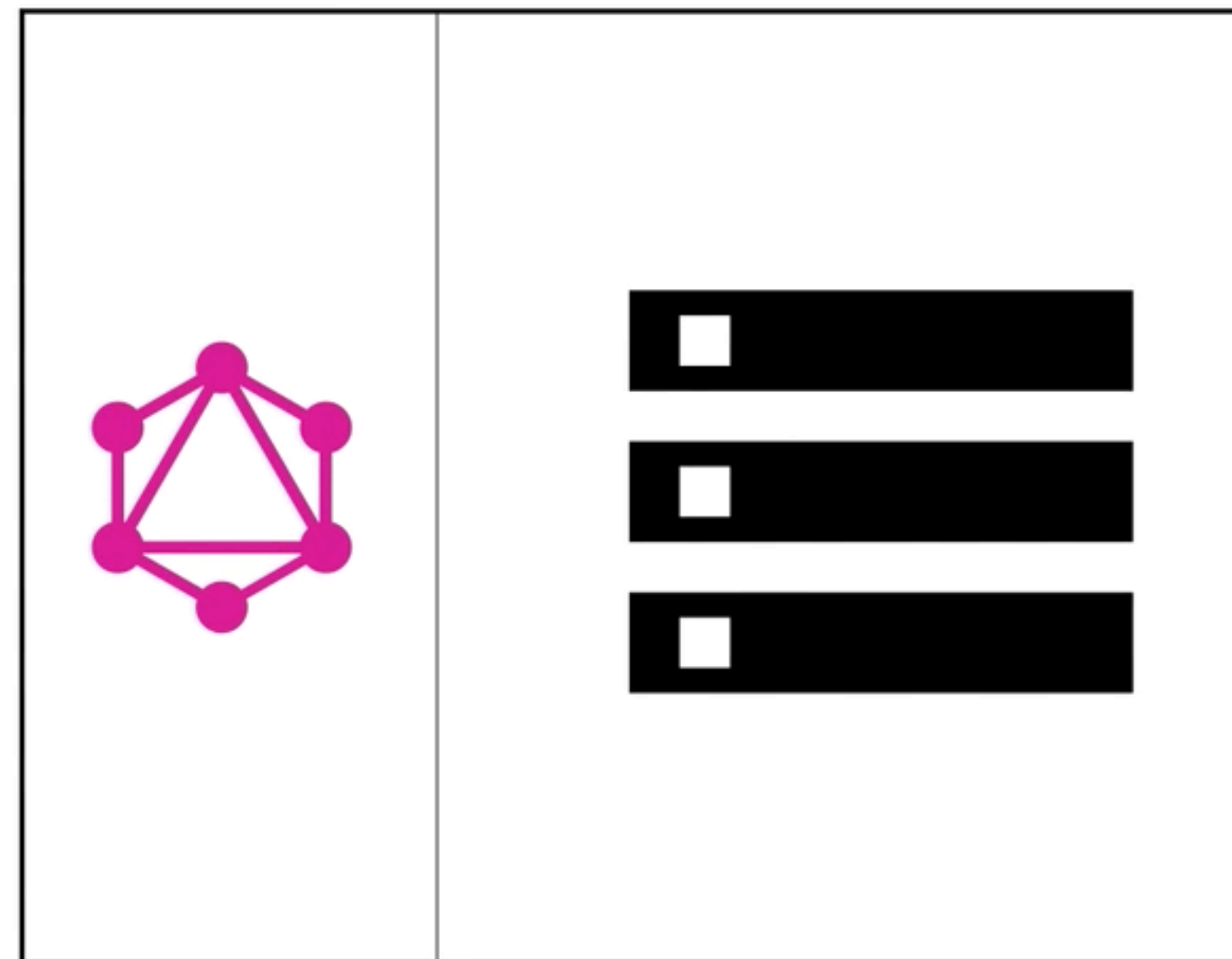
/users/<id>/posts

/users/<id>/followers

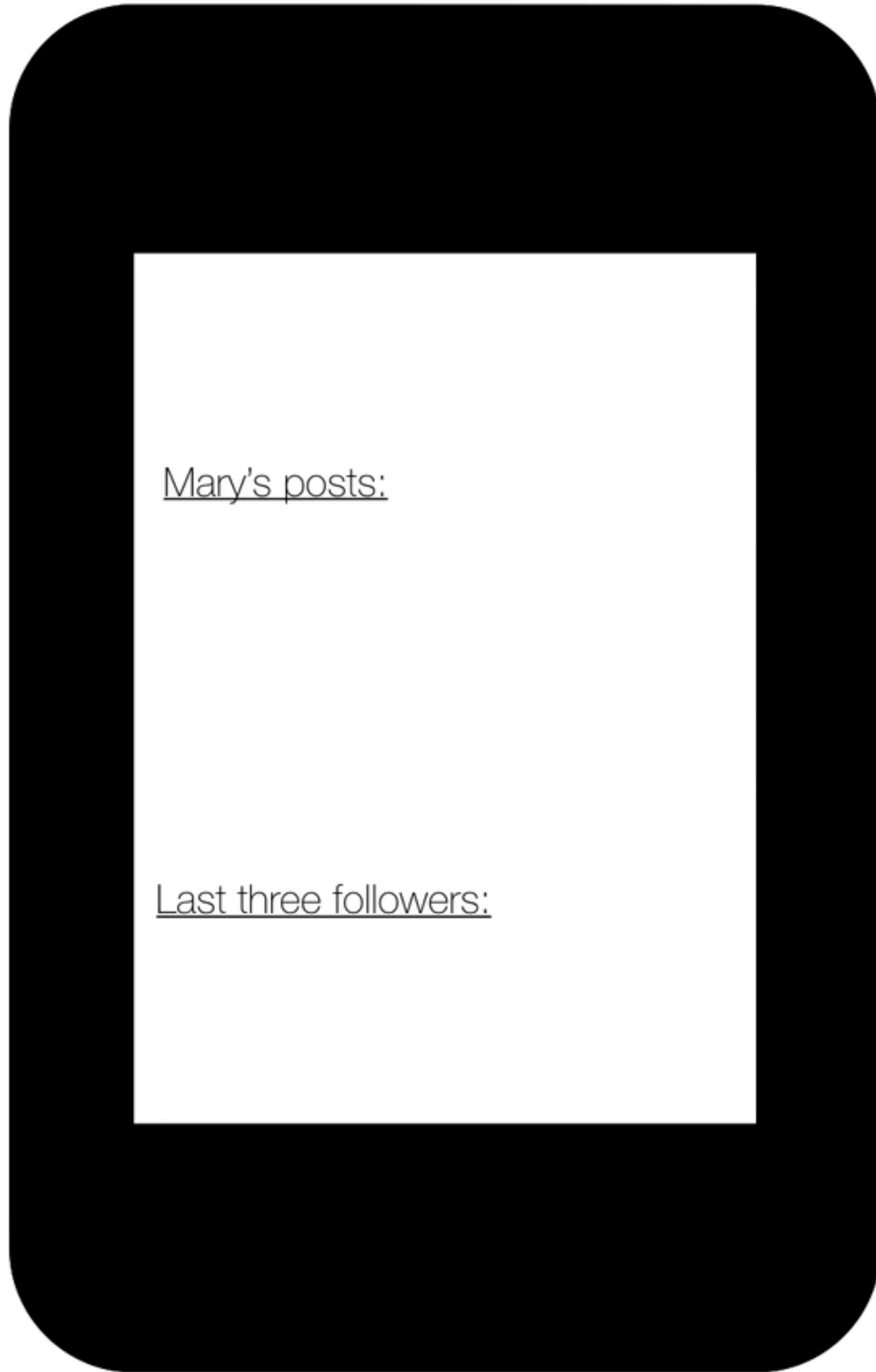


App using GraphQL.

1 API endpoint

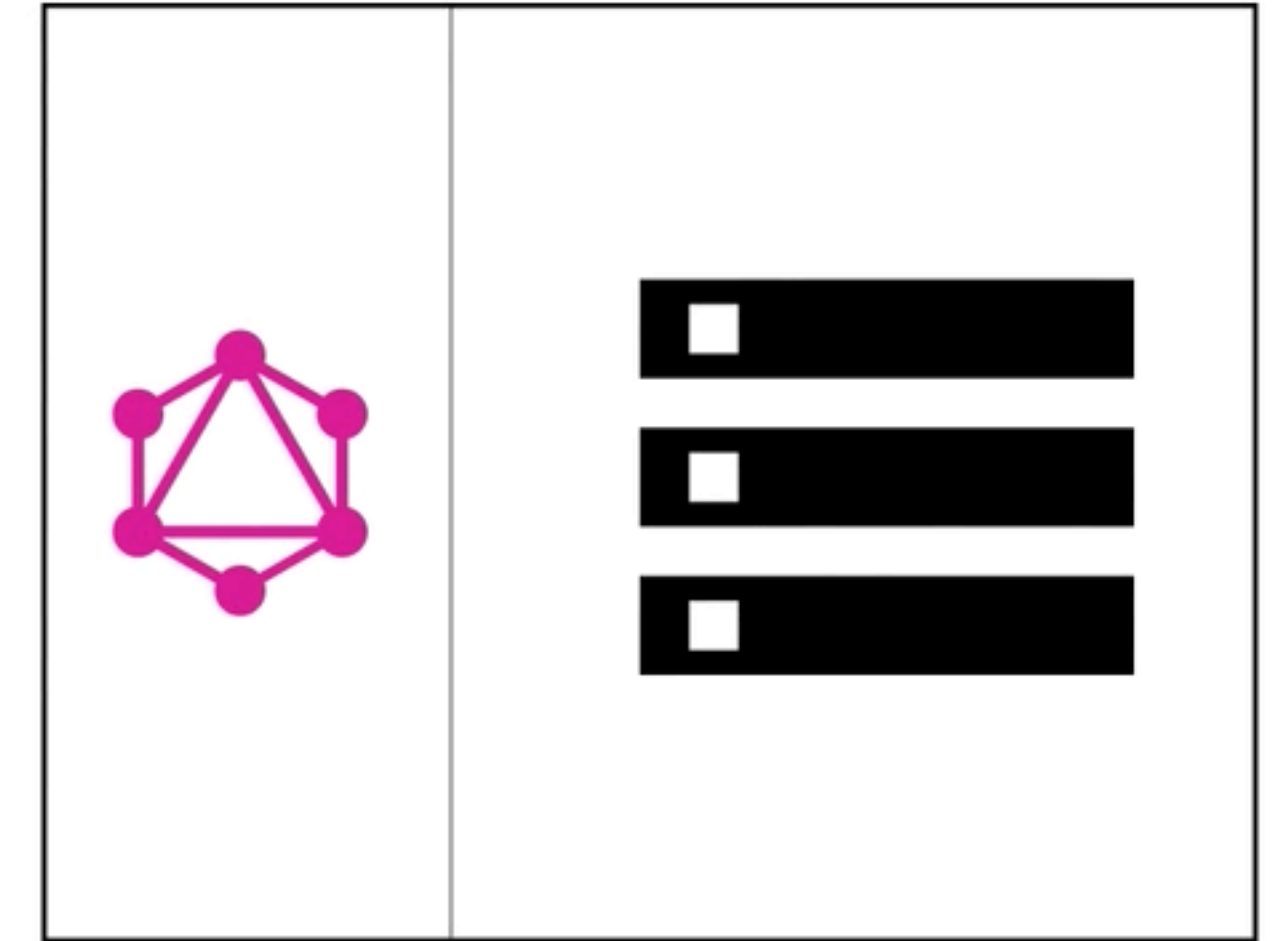


① Fetch everything with a single request



HTTP POST

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



① Fetch everything with a single request

Mary

Mary's posts:

Learn GraphQL Today

React & GraphQL - A declarative love story

Why GraphQL is better than REST

Relay vs Apollo - GraphQL clients

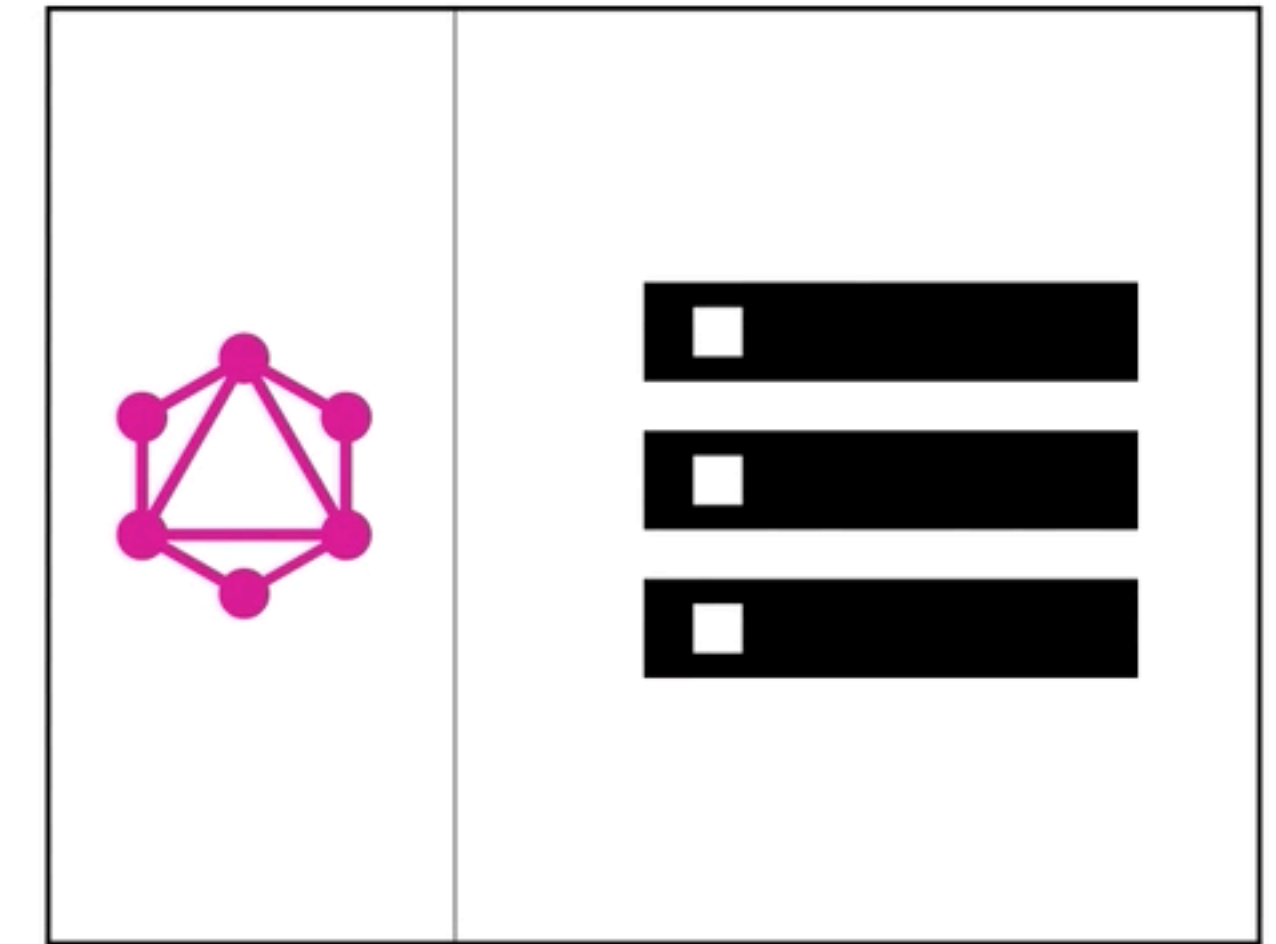
Last three followers:

John, Alice, Sarah



HTTP POST

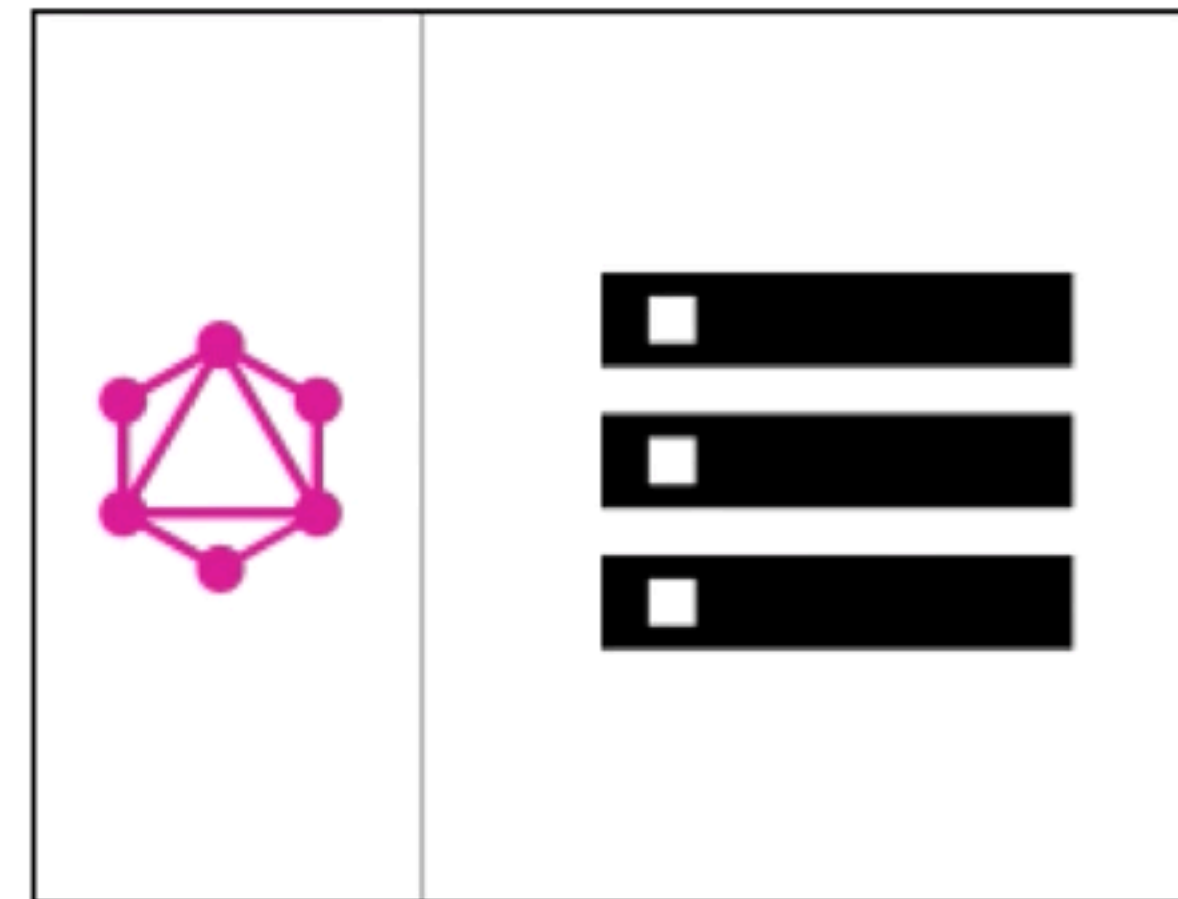
```
{
  "data": {
    "User": {
      "name": "Mary",
      "posts": [
        { title: "Learn GraphQL today" },
        { title: "React & GraphQL - A declarative love story" },
        { title: "Why GraphQL is better than REST" },
        { title: "Relay vs Apollo - GraphQL Clients" }
      ],
      "followers": [
        { name: "John" },
        { name: "Alice" },
        { name: "Sarah" }
      ]
    }
  ]
}
```



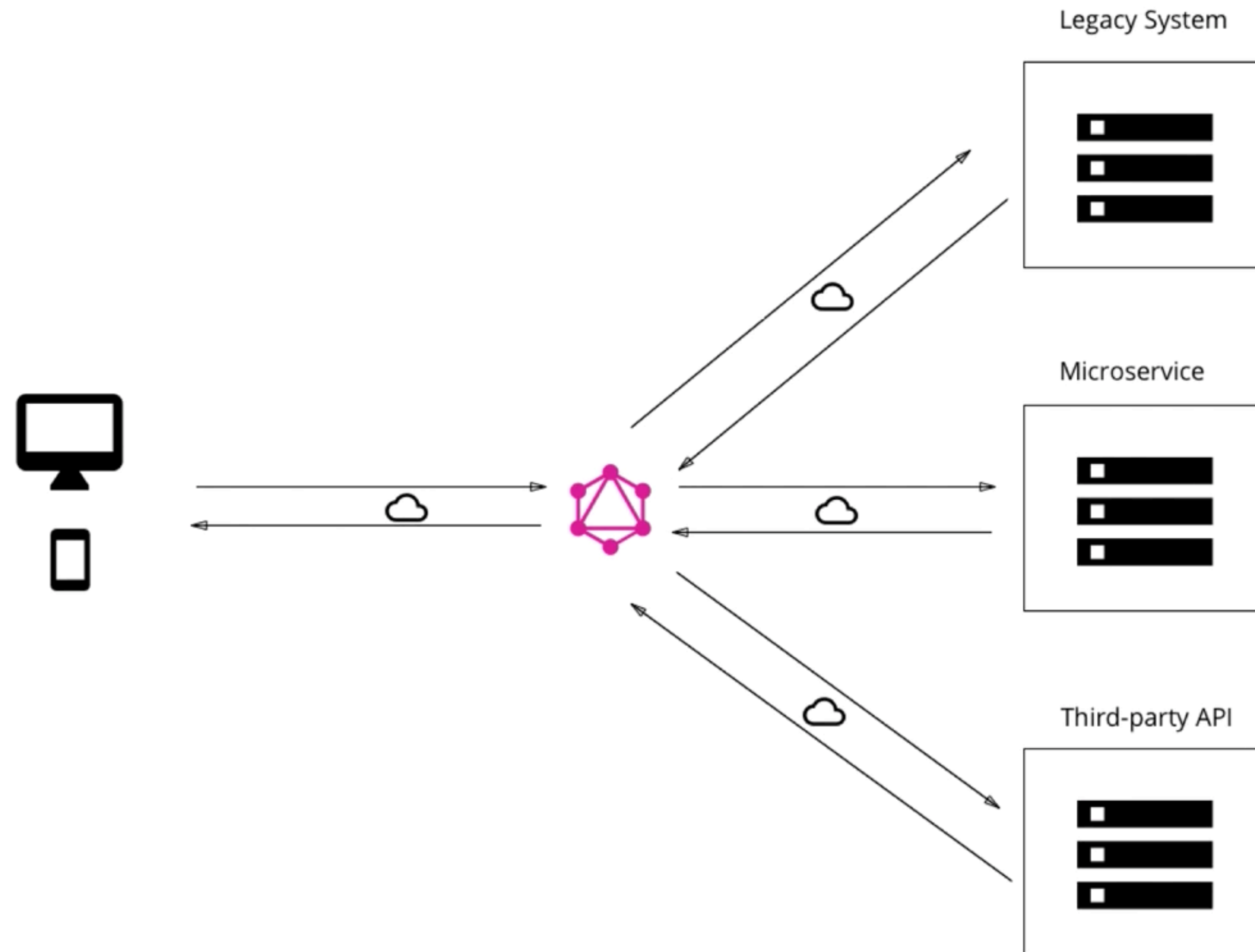
Systems.



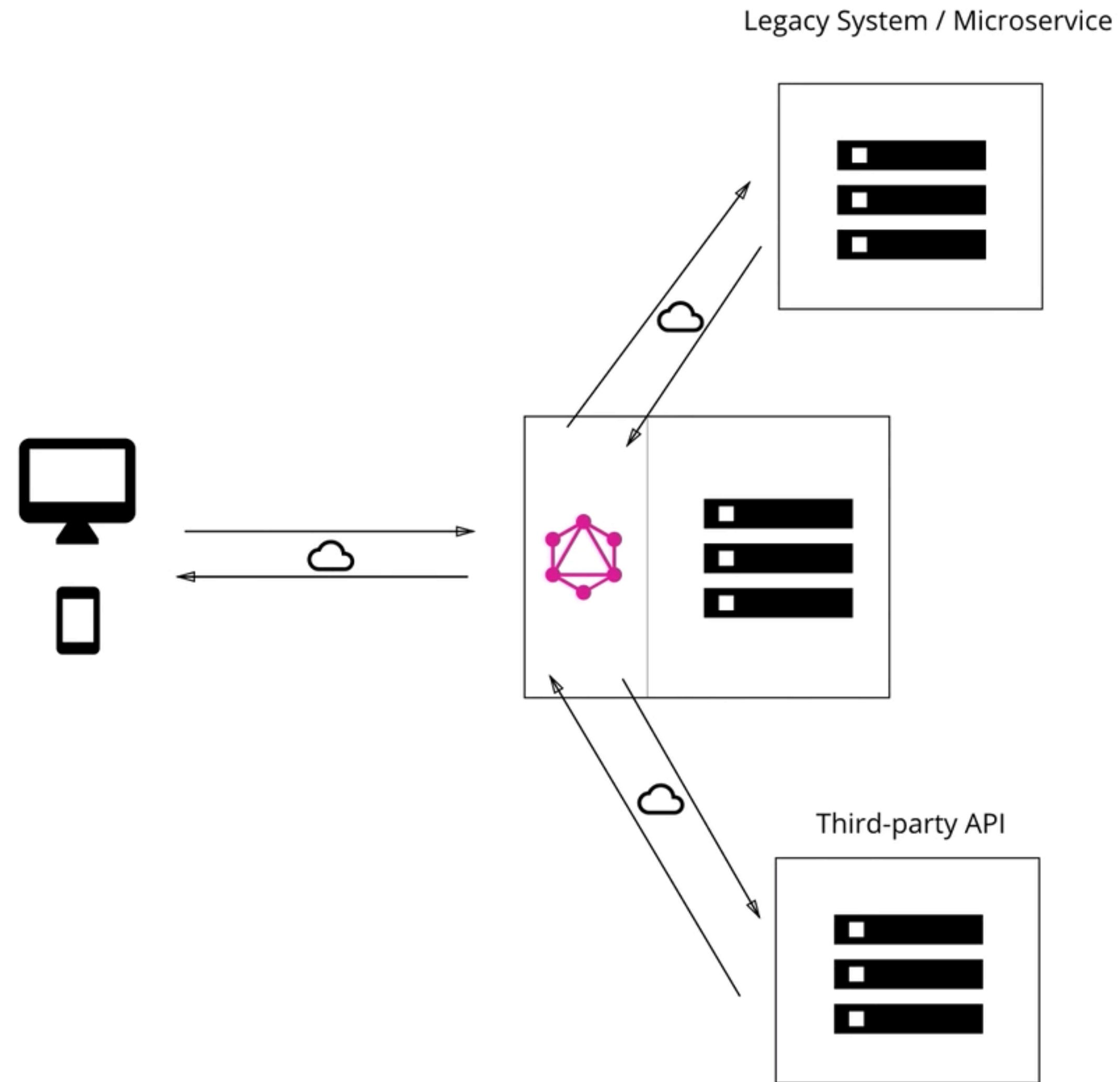
As a connected Database.



Integrating with existing systems.



Hybrid system.



Setting up GraphQL.

- Server using Apollo + Express (or Hapi)
- Client using Apollo + React
- Define data schema
- Define data resolvers

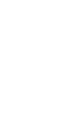


Demo.



Recap.

- Single endpoint for easier data fetching
- Declarative data, request only what you need
- Real-time data feedback
- Caching on client / server



Info.

- howtographql.com

Tutorials

- apollographql.com



- graphql.org

