

---

# What is all the fuss about GraphQL



# Code PaLOUsa 2019 Sponsors

Humana®

**XC**  
Experience  
Center



<code>  
**LOUISVILLE**



Humana®



modis



PROSOFT **rh** Robert Half®



# My Intro

- Baskar Rao

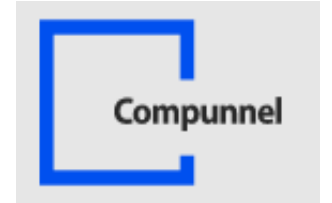
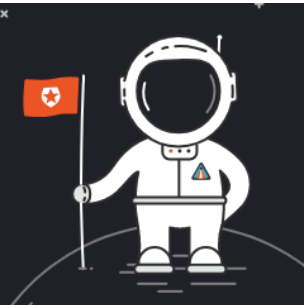
Twitter – baskarmib ,

<https://www.linkedin.com/in/baskarrao-dandlamudi>

[baskarrao.dandlamudi@outlook.com](mailto:baskarrao.dandlamudi@outlook.com)

<https://github.com/baskarmib>

Auth0 Ambassador , JavaScript and Friends Team.



# Agenda

---

- History of API Architectures and Clients
- Overview of Graph QL
- GraphQL Supported Platforms
- Overview of GridSome
- Demo of GraphQL API and Blog to GridSome
- Further Resources for Learning

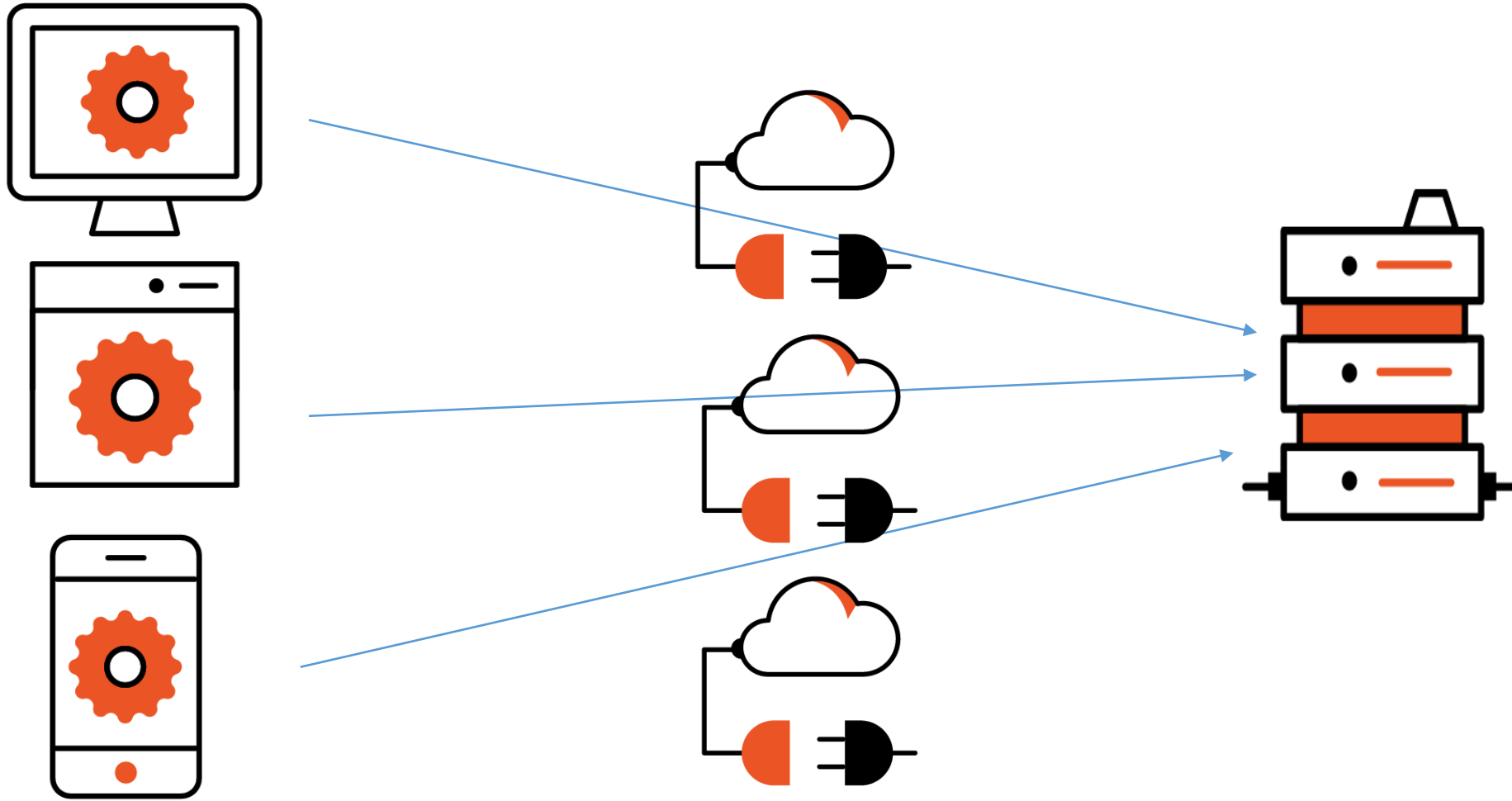
---

Graph Database



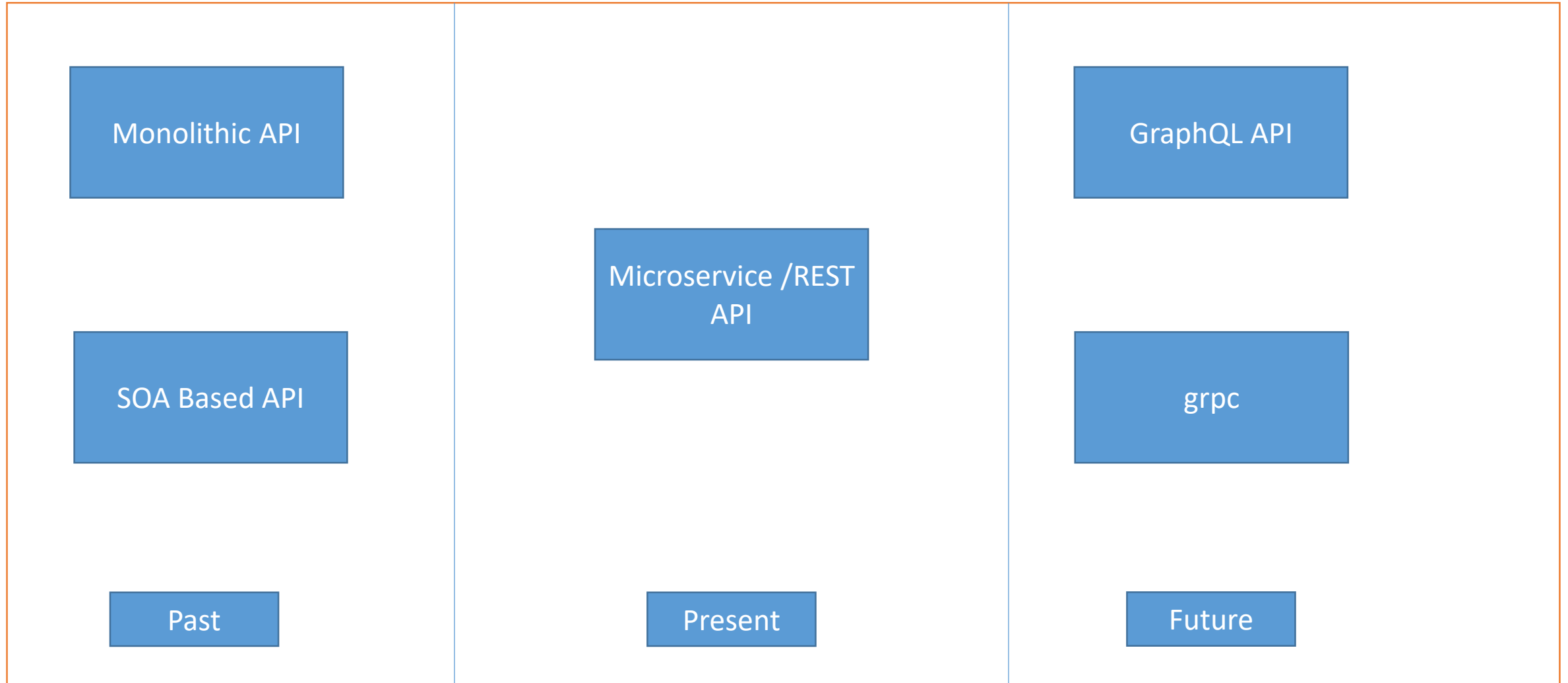
Graph QL

# Evolution of Clients



# History of Architectures

---



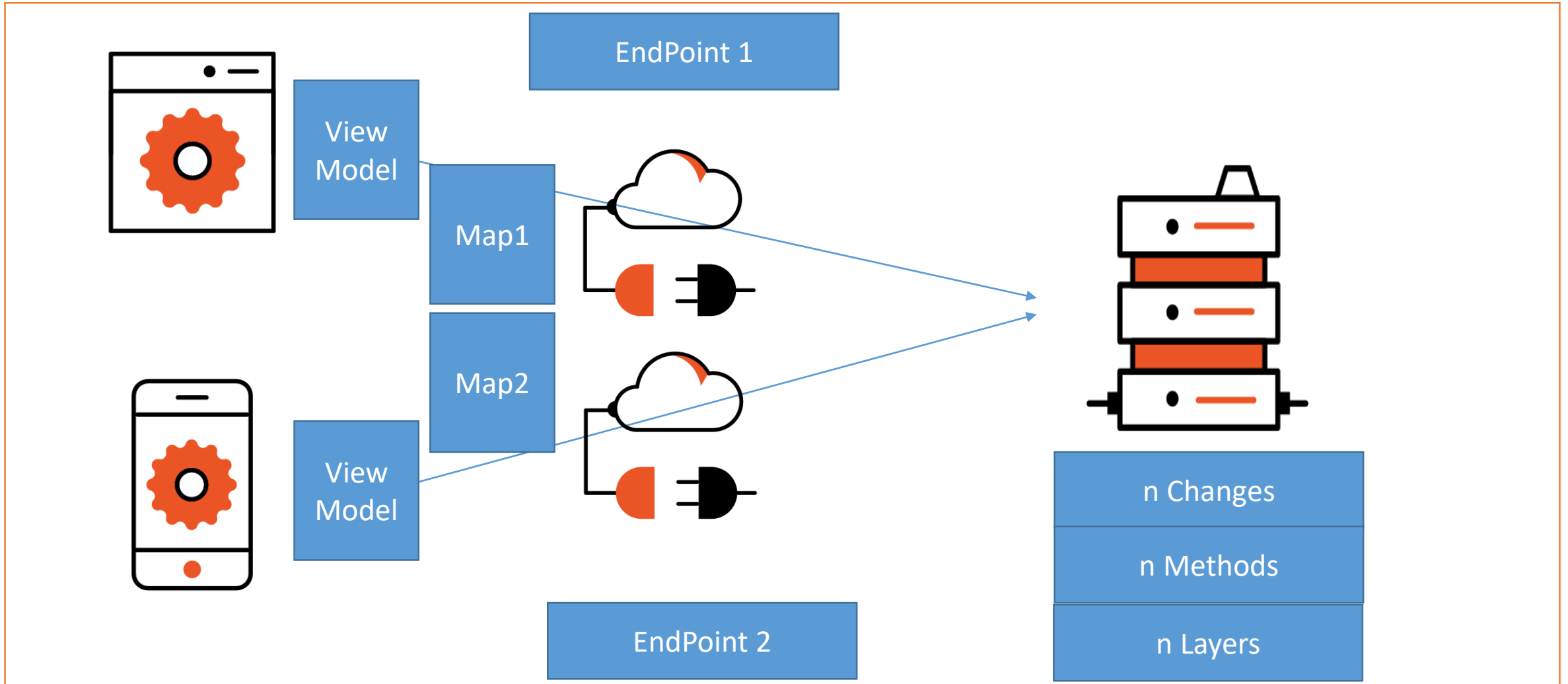
# Common Problems in REST API

---

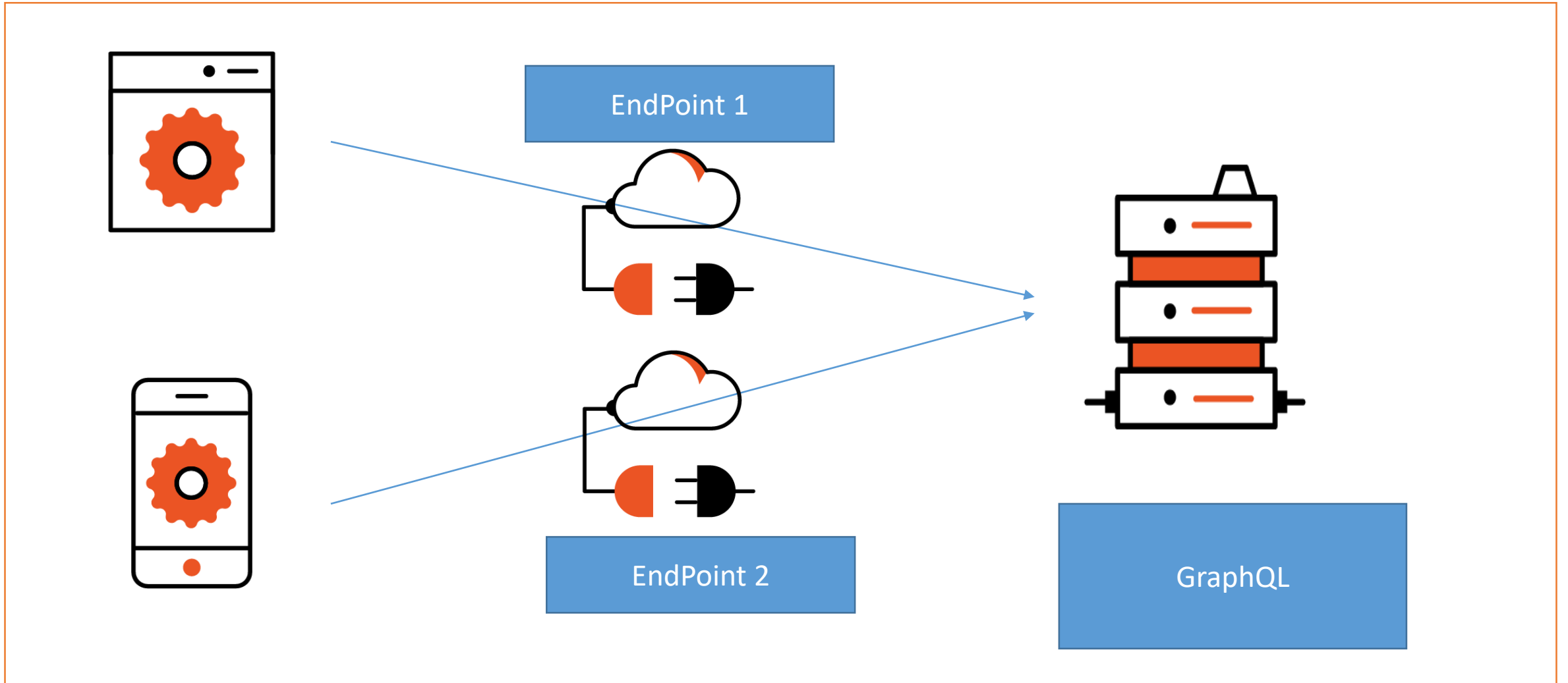
- As the functionalities change, the endpoints change.
- Different clients expecting different results would result in endpoint change.
- N number of clients would lead to n Number of methods or layers to be changed.
- API might be returning major chunk of whole table, and this comes with a cost to network.
- Documentation would need to be implemented separately using swagger.



# Common Problems in REST API



# Common Problems in REST API



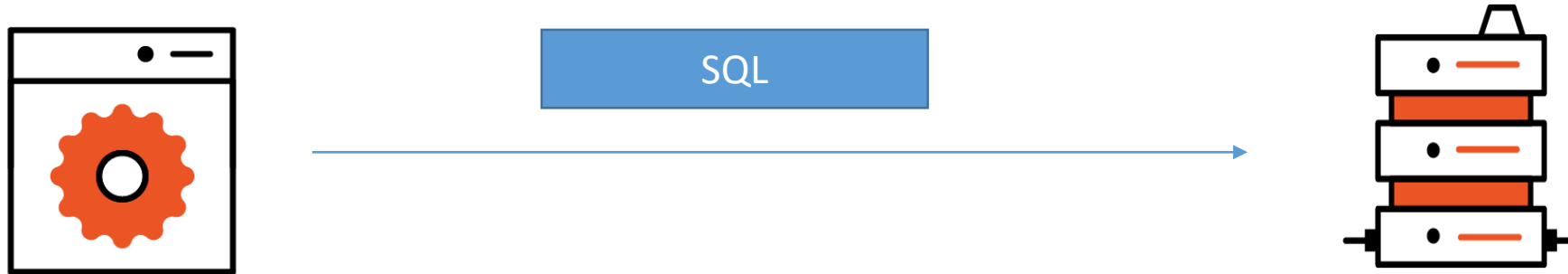
# Overview of Graph QL

---

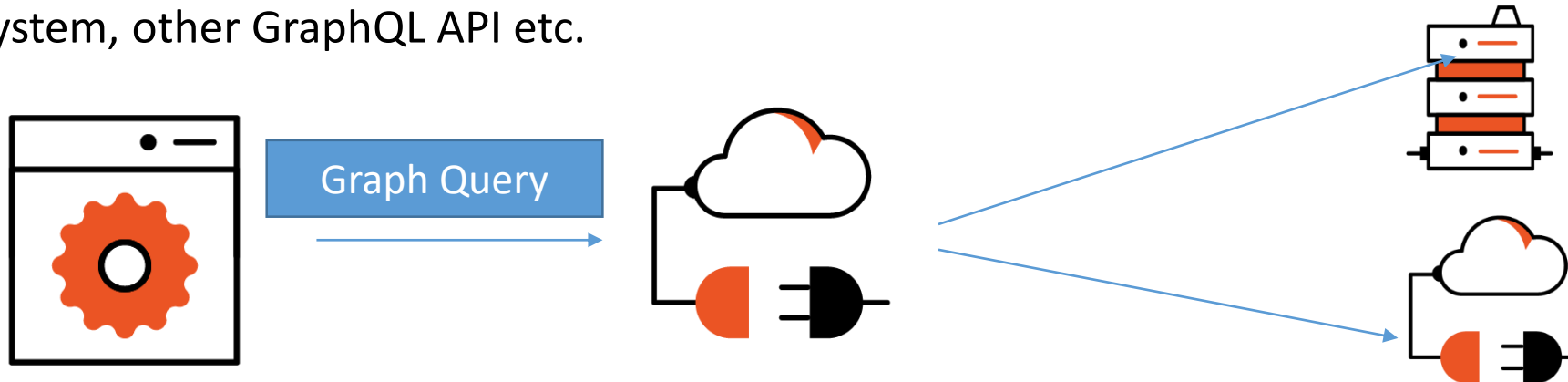
- Graph QL is maintained and open sourced by Facebook Engineers.
- Graph QL is declarative data query language
- Specify what you want and get the same
- Reduce Round Trips between client and server
- Makes development easy and respond to changes soon
- Graph QL can be implemented in any language of choice and architecture of choice.
- C# / .NET, Clojure ,Elixir, Erlang , Go, Groovy, Java, JavaScript, PHP, Python, Scala, Ruby
- Prioritize to use Schema First Development

# Overview of Graph QL

- Traditional SQL is used to query database.



- Graph QL can be used to query anything. Client sends query to API and API can query database, filesystem, other GraphQL API etc.



# GraphQL vs SQL

---

GraphQL	SQL
Query	Select
Mutation	Insert, Update, Delete
Subscription	

# GraphQL vs SQL

- Query in GraphQL is set of string sent in the json body.

The screenshot shows a REST client interface with a POST request to `http://localhost:4000/graphql`. The request body is a GraphQL query: `{ hello }`. The response is a JSON object: `{ "data": { "hello": "Hello CodePalOUsa 2019" } }`.

**Request Details:**

- Method: POST
- URL: `http://localhost:4000/graphql`
- Body Type: GraphQL <sup>BETA</sup>
- Schema: No Schema

**Query:**

```
1 {  
2   hello  
3 }
```

**GraphQL Variables:**

```
1
```

**Response Details:**

- Status: 200 OK
- Time: 21ms
- Size: 288 B

**Response Body (JSON):**

```
1 {  
2   "data": {  
3     "hello": "Hello CodePalOUsa 2019"  
4   }  
5 }
```

# GraphQL vs SQL

- Fragments can be used to define reusable columns

The screenshot shows a GraphQL client interface with a POST request to `http://localhost:4000/graphql`. The 'Body' tab is selected, and the 'GraphQL BETA' radio button is chosen. The query is as follows:

```
1 {
2   project(projectId: "3434343")
3   {
4     ...projectdetails
5     employees{
6       firstName
7       lastName
8     }
9   }
10 }
11
12 fragment projectdetails on Project{
13   projectId
14   projectName
15   projectDescription
16   projectStatus
17 }
```

The query demonstrates the use of a reusable fragment named `projectdetails` to define common fields for a `Project` type, such as `projectId`, `projectName`, `projectDescription`, and `projectStatus`. This fragment is then used within the `project` query to retrieve specific details and related data like `employees`.

# GraphQL vs SQL

---

- SQL

```
select empId, firstName from Employee where employeeId = 5
```

- Graph QL

```
{
```

```
  employee(empId : "5") {empId
```

```
    firstName}
```

```
}
```



# GraphQL vs SQL

Mutation in Graph QL is used to perform data changes.

The screenshot displays a GraphQL client interface. At the top, a dropdown menu is set to 'POST' and the URL is 'http://localhost:4000/graphql'. Below this, a row of radio buttons shows 'GraphQL BETA' as the selected content type. The 'QUERY' section contains a mutation query:

```
1 mutation
2 {
3   updateProjectStatus(projectId:"3434343", projectStatus:"InProgress")
4   {
5     projectId
6     projectStatus
7     projectDescription
8   }
9 }
```

Below the query, the 'Body' tab is active, showing the JSON response in 'Pretty' format:

```
1 {
2   "data": {
3     "updateProjectStatus": {
4       "projectId": "3434343",
5       "projectStatus": "InProgress",
6       "projectDescription": "This is a new project"
7     }
8   }
9 }
```

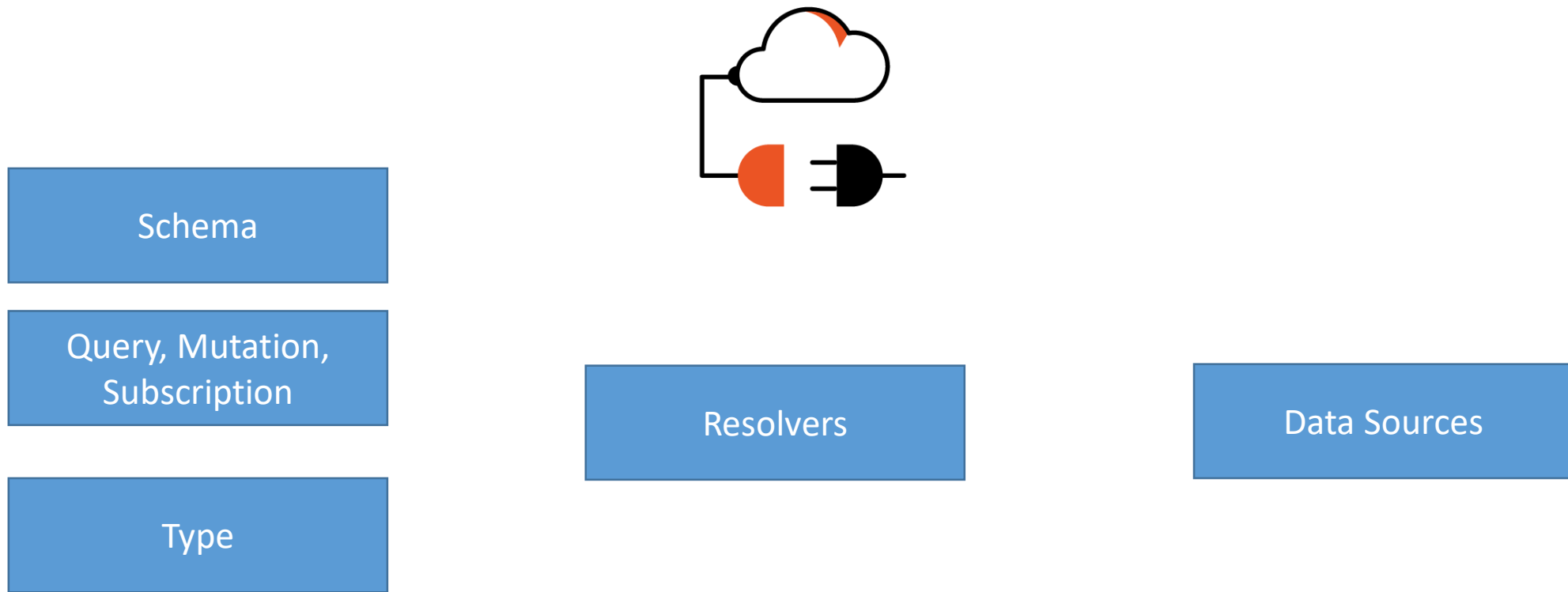
# GraphQL vs SQL

---

- Subscriptions is used for detecting changes to data
- These provide a way to get real time updates using web sockets
- In SQL we will not be able to get the change notifications directly to client, it would involve multiple hacks using Signal R and other libraries.

# GraphQL API

---



# Graph QL Platforms

---

- There are different Graph QL clients like Relay developed by Facebook which couples well with React and React Native
- Apollo Client is another popular client which is framework agnostic.
- Apollo Graph QL Server makes it easy to develop query with nested types using resolvers
- AWS Amplify also supports Graph QL API development
- Hasura provides Graph QL on top of Postgres and there are certain limitations compared to Apollo.

# What is Gridsome

---

- Gridsome provides GraphQL support to Vue Js. It has been inspired from Gatsby which is for React World
- Gridsome can be used to develop PWA applications and static sites using markdown
- Gridsome provides JAMstack support to Vue Js world



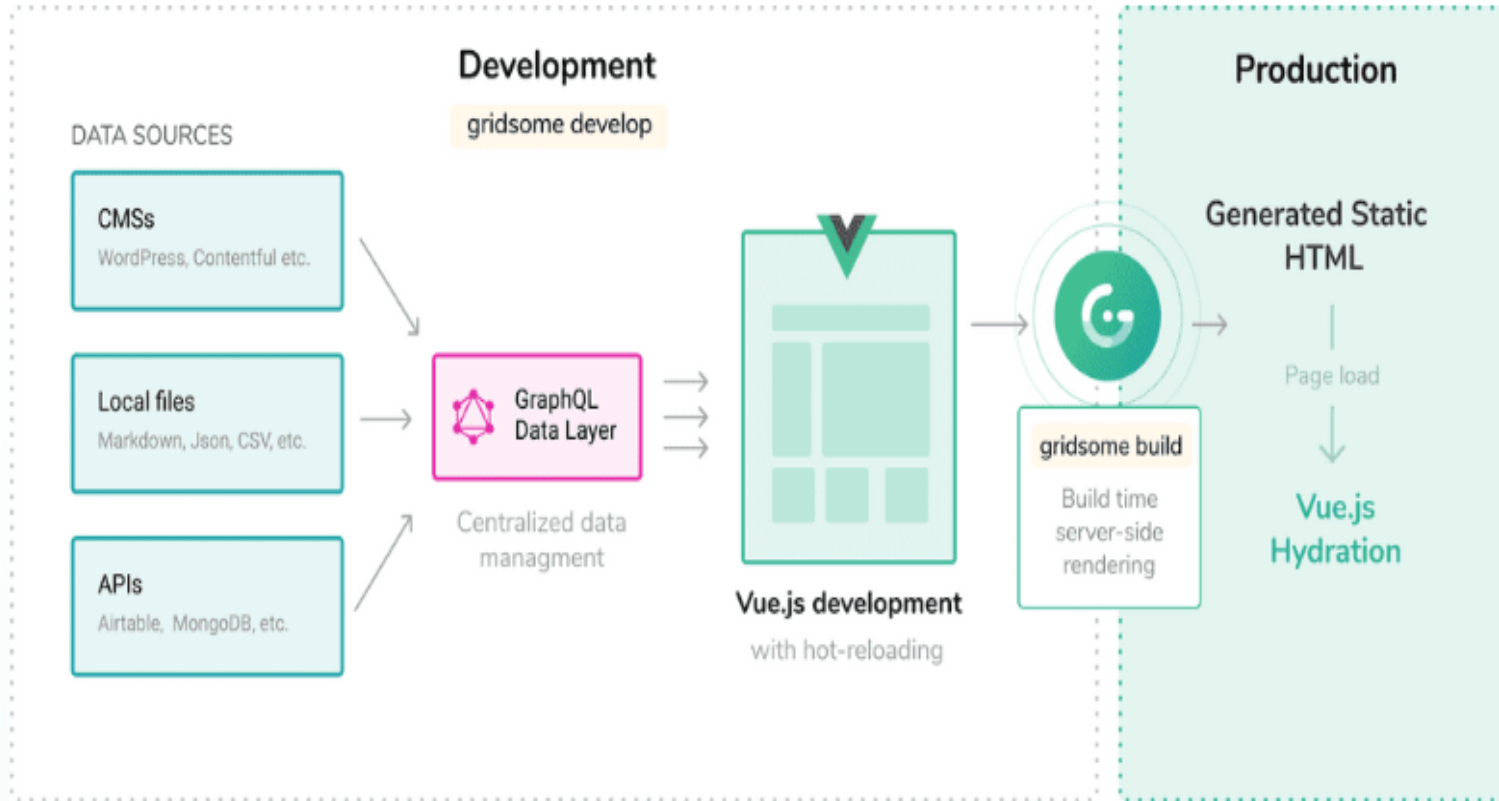
# What is Gridsome

---

- Easy to get started and learn
- Knowledge of HTML, Vue and JavaScript and Node Js
- In Gridsome pages are created using Vue Components
- Gridsome supports routing by page name.



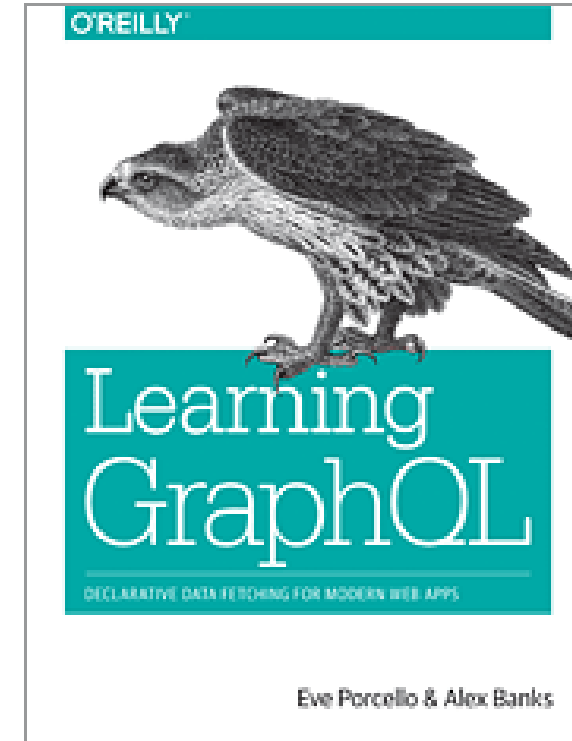
# What is Gridsome



# Reference

---

- <https://gridsome.org/>
- <https://graphql.org/>
- <https://www.graphqlbin.com/v2/new>
- <https://jamstack.org/>
- <https://www.apollographql.com/docs/tutorial/introduction/>
- <https://identicons.dev/> Server, API and Other Images by Auth0





# Thank you CodePaLOUsa 2019

---

## Questions



<https://www.linkedin.com/in/baskarrao-dandlamudi>

[baskarrao.dandlamudi@outlook.com](mailto:baskarrao.dandlamudi@outlook.com)

<https://baskarrao.wordpress.com/>

<https://github.com/baskarmib>