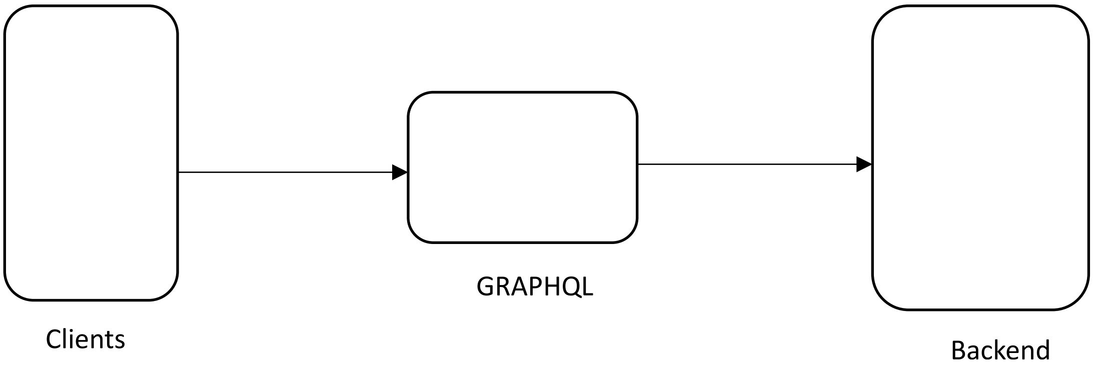
GRAPHQL

GRAPHQL is a query language developed by **Meta**. It provides schema for the data in API.

And gives power to clients what exactly the need. GARPHQL is between Clients and Backend Services



It can also Aggregate multiple resources requests into single query.

It also Supports **Mutations and Subscriptions**. Mutations are GRAPHQL's way of applying data modifications to the resources. Subscriptions are way for receiving notifications on modifications that are made on data.

Feature	**REST**	**GraphQL**
Data fetching	Multiple endpoints	Single endpoint
Over-fetching	Yes (returns fixed structure)	No (client specifies fields)
Versioning	Uses versioned APIs (v1, v2)	No versioning (evolves via schema)
Flexibility	Less flexible	Highly flexible and efficient

Both uses HTTP requests and HTTP responses.

Query example:

RSET API

URL: GET/users/123

Response:

```
{
  "id": 123,
  "name": "Yashwanth",
  "email": "yash@example.com",
  "phone": "9876543210",
  "address": {
    "city": "Bangalore",
    "zip": "560001"
  }
}
```

GRAPHQL

Query:

```
{
  user(id: 123) {
    name
    email
  }
}
```

Response:

```
{
  "data": {
    "user": {
      "name": "Yashwanth",
      "email": "yash@example.com"
    }
  }
}
```

As we can see only name and email are required in REST API it gives extra Phone and Address.This is called **Over Fetching.** In GRAPHQL it inly gives name and email. There is no Over Fetching.

GRAPHQL doesn’t use URL it uses Schema.

Drawbacks of GRAPHQL:

REST API uses GET requests for fetching records, whereas GRAPHQL uses POST reuests making difficult in caching.