1. **GraphQL vs RestAPI:**

* Multiple end points in Rest vs Only one Endpoint (Post/graphql)
* Both JSON data is exchanged
* Any server side language any front end framework for Both
* Both are stateless(no session based authentication). They have their own authentication mechanism

Diagram

Description automatically generated with low confidence

1. **URL Driven Vs Query Language:**

* Core difference between these two API is: How we send the requests to them
* In rest APi we work with multiple endpoints, get,post etc.It is URL based, so if we send request to wrong URL – we will get error
* In Grapql api, we have one end point only – can have any path we want ; it always takes post request only
* While a REST API is flexible where it supports – GET, POST, PATCH, PUT, DELETE etc, Graphql APIs work with only POST

Diagram

Description automatically generated

* This is because in Graphql api ,we don’t target different endpoints/urls – Instead we state what we want in our request body. Hence HTTP request will all aways be post. Only the request body which includes a query language – a command which the graphql api on the backend is able to parse
* Because the graphql query language which in the end is encoded in the request body is standardized. Hence there is a common understanding on how it should be parsed, regarding which commands we can use.
* Hence the graphql-api server is able to parse the incoming request and understands what we want.
* IT is almost similar like we query a database with sql or with mongodb or whichever database or query language we are using - The same logic applies here.
* Then the Backend takes the parsed command we sent and does whatever needed in the backend

1. **REST API**

* Client server communication

Chart

Description automatically generated with medium confidence

* API endpoints map resources or actions that happen on the server

Diagram, calendar

Description automatically generated

1. **How does GRAPHQL work?**

* Same Client server communication
* No Multiple end points – only post request
* Query expression that is part of request body, which is then parsed by the server – so that server retrieves the required data.

Diagram

Description automatically generated

1. **A GraphQL Query:**

* **Eg:**

Text

Description automatically generated

1. **Operation Type:**

* Three types a) Query - Query or get for some data b) Mutation – Edit some data c) Subscriptions – Informs about changes . Subscription not possible with normal http

1. **Operation Identifier:**

* user is the operation endpoint or identifier. This will inform the server with which **resource**  we want to work.
* We cant send arbitrary queries
* In server while we pass the incoming query, we support only couple of endpoints in server side
* We don’t want everything to be mutatable.
* In server side - When we define the graphql api – Just like that we don’t parse the incoming query . Instead we define which kind of queries or resources we support

1. **Requested Fields;**

* Fields we request as part of the query

Diagram

Description automatically generated**\***

* With rest API we target a specific resource, eg: GET /posts, where we will get all posts with all fields related to posts. We may use query param to limit the number of posts retrieved, but it adds more complexity to the backend .
* Often we will end up fetching more data than required . If client app wants to output only an overview of all the posts, Hence fetching too much is common problem of rest APIs
* The advantage of graphql is that we can be specific about which kind of data you want to retrieve. If we want to get user data here but only age means, we can retrieve that alone
* Hence graphql allows to be more specific, regarding which data you want to fetch