𝗥𝗘𝗦𝗧 𝗔𝗣𝗜 𝗩𝘀 𝗚𝗿𝗮𝗽𝗵𝗤𝗟:  
When it comes to API design, REST and GraphQL each have their own strengths and weaknesses.  
𝗥𝗘𝗦𝗧 𝗔𝗣𝗜:  
1) Uses standard HTTP methods (like GET, POST, PUT, DELETE) for common operations (create, read, update, delete).  
2) Works great for creating simple, uniform interfaces between different services or apps.  
3) Easy to implement caching strategies.  
4) However, it may need multiple requests to gather related data from different endpoints.  
𝗚𝗿𝗮𝗽𝗵𝗤𝗟:  
1) Offers a single endpoint for clients to request exactly the data they need.  
2) Clients can specify the precise fields they want, even nested within each other, and the server responds with optimized data containing only those fields.  
3) Supports Mutations for changing data and Subscriptions for real-time updates.  
4) Perfect for pulling data from multiple sources, especially when frontend requirements change frequently.  
5) But, it adds complexity to the client side and may be prone to abuse if not protected properly.  
6) Caching strategies can be more challenging compared to REST.  
  
Choosing between REST and GraphQL depends on your app's needs and your team's preferences. If your frontend is complex or changes often, GraphQL might be the better choice. But if you prefer simplicity and consistency in your API contracts, REST could be the way to go.

Activate to view larger image,

