REST vs GraphQL  
  
𝗥𝗘𝗦𝗧  
- Relies on standard HTTP methods like GET, POST, PUT, DELETE to perform CRUD operations  
- Excellent for simple, uniform interfaces between separate services or applications  
- Caching strategies are straightforward to implement  
- Can require multiple API calls to fetch related data, leading to chatty interfaces  
  
𝗚𝗿𝗮𝗽𝗵𝗤𝗟  
- Provides a single endpoint and lets consumers precisely query for the data they need  
- Clients specify the exact fields required in nested queries, and the server returns optimized payloads containing just those fields  
- Supports Mutations for modifying data and Subscriptions for real-time notifications  
- Great for aggregating data from multiple sources and works well with complex frontends that have rapidly changing data needs  
- Shifts complexity to clients and can enable abusive queries without proper safeguards  
- Caching strategies are less straightforward compared to REST  
  
The best choice depends on the application architecture and team preferences. GraphQL fits better for fluid or intricate frontend requirements, while REST suits applications needing simple, consistent interfaces.  
  
Neither one is a silver bullet - evaluating tradeoffs and limitations is important. Both REST and GraphQL are solid options for exposing data and powering modern applications. Carefully analyzing requirements rather than defaulting to one or the other is key to picking the right API approach.