Thank you arsalan.

1. Now we look at which case graphQL have better performance
   1. Data fetching control, for amount of needed data, graphQL is better.
   2. Multiple data source, as Arsalan mentioned, graphQL is the better choice.
   3. Alleviating(减轻) bandwidth, as you know, small devices like mobile phones, smartwatches etc. cannot handle amount of data. GraphQL help minimize the issue.
   4. Rapid prototyping(原型设计), GraphQL exposes a single endpoint that allows you to access multiple resources. That if your UI changes, it will not change the server.
2. Following case rest api is better,
   1. For complex queries, rest api is easier to design because you can establish multiple endpoints for specific needs.
   2. Simple application, GraphQL still needs to add complexity such as types, queries, mutators, resolvers etc to increase respond time.
3. After we compare both api, we find our objectives,
   1. first is identify use cases, we will choose each user cases, evaluate the performance for them to tell which api is better in specific user cases, then give advice to customers,
   2. second is make a metric analysis, we analyze the key parameters like response times, rates, throughput, and system scalability with different user numbers.
4. To meet different use case need, which means for specific categories, we will provide different recommendations of Apis. We can consider different categories like different types of cars, pickup trucks and race car have different performance request, so the inside engine (as API) we recommend will be also different.

For the following part dan will take over.