Question 1. Explain some additional concerns related to using REST APIs.

Ans.

The idea of utilizing full range of powers of HTTP in REST API lies on the assumption that the middleware also implements HTTP correctly which is not always true. There are a lot of libraries that support only GET and POST requests but not PUT, PATCH, DELETE. This also means that RESTful services are subject to all the application layer security vulnerabilities that traditional web applications have had to deal with over the years (e.g. OWASP Top 10, etc.).

REST APIs make it difficult to find out the URL spaces since the URL is used for directories as well as input parameters which could be injected with malicious data. For example:

https://www.example.com/api/phonebook/UserID/12345

The parameter name (UserID) and value (12345) are in the URL Path. This is a common practice in REST but this completely throws off automated test tools. They have no way of knowing what to do with such elements – whether to consider them as directories or to fuzz test them as parameters.

Although controversial, the vocabulary of the HTTP status codes used by REST is also ambiguous. There are no clear object demarcations of the sensible reply that can be given in all the cases. For example, the status codes 200 – OK and 201 – Created can both be used in the problem of creating a new entry in the database. Although there are conventions that are followed, it is less than intuitive in many cases.

And finally, REST has an issue of sending multiple requests to fetch data which desirable amount and type of data and the requested format is not controlled by the client.

Question 2. Compare and contrast the benefits and disadvantages of using a RESTful architecture vs. a graph query language. See http://graphql.org/ for details.

Ans.

- GraphQL is better in a way that we can control the data we get since we are querying for specific different resources as compared to REST APIs in which we don't have much control over all the parameters as there might be more than 1 resources in one JSON response.

- GraphQL API helps reduce the number of queries required to be made to the server to get the same amount of data as compared to REST API since it can pack requests for multiples resources in a single request. REST APIs would need unstructured parameters and conditions that are hard to manage.

- Since the number of queries are reduced in GraphQL as compared to REST, it can work better with slow network speed connections

- Since the clients control the request type in GraphQL, they can be decoupled from servers and hence improved on their own

- REST uses the HTTP caches, status codes, etc. while GraphQL has its own conventions. Since HTTP is an established standard, it’s easier to follow

- REST API has a more robust structure making it more long-lasting as compared to GraphQL which is more performance optimizer

- As compared to REST, GraphQL endpoint needs to be on the server which gives rise to the problem of new libraries that we don't know yet

Sources:

https://stackoverflow.com/questions/41141577/graphql-or-rest

https://news.ycombinator.com/item?id=13759520

https://medium.freecodecamp.org/rest-apis-are-rest-in-peace-apis-long-live-graphql-d412e559d8e4

https://blog.barracuda.com/2015/07/09/the-challenges-in-securing-rest-apis/