N+1?

N+1 queries are a common concern whenever you abstract over the data fetching layer, and GraphQL is no outlier. GraphQL is at high risk for N+1 queries because each object is given a function as a resolver, meaning that it is very easy to naively call that function and make too many round trips. Luckily there is tooling such as Facebook's DataLoader too ease the process.

CACHING

Currently the most popular and well supported libraries all provide built in caching functionality, further reducing client <-> server requests required. This cache is normalized, allowing even deeply nested objects to be identified and retrieved from cache rather than the database.