

### OPTION 2: JUST HAVE THE FRONT END DO IT

Already have a REST API? The front end time savings in a complex app are often enough to justify having *them* do all of the work in writing a GraphQL server that proxies to existing endpoints. Does this work for you? You'll need:

- ▶ A strong front end team that is comfortable working in Node
- ▶ Complicated enough state management/RESTful logic to justify the extra work
- ▶ An existing server to proxy requests to\*

### OPTION 3: TEAM HUDDLE

Neither of the previous options look good? Maybe you have limited front end development time or the server is designed in such a way as to make the previous options infeasible. This option serves as a blend of the last 2.

Here the front end team is responsible for building and maintaining GraphQL specific logic (schemas, types, subscriptions, etc) while the back end team handles the resolver logic and DB interactions. This might be best if:

- ▶ You have an existing server AND
- ▶ Both front and back end teams are limited on time
- ▶ The front end team is only slightly familiar with tools and technologies used on the server