

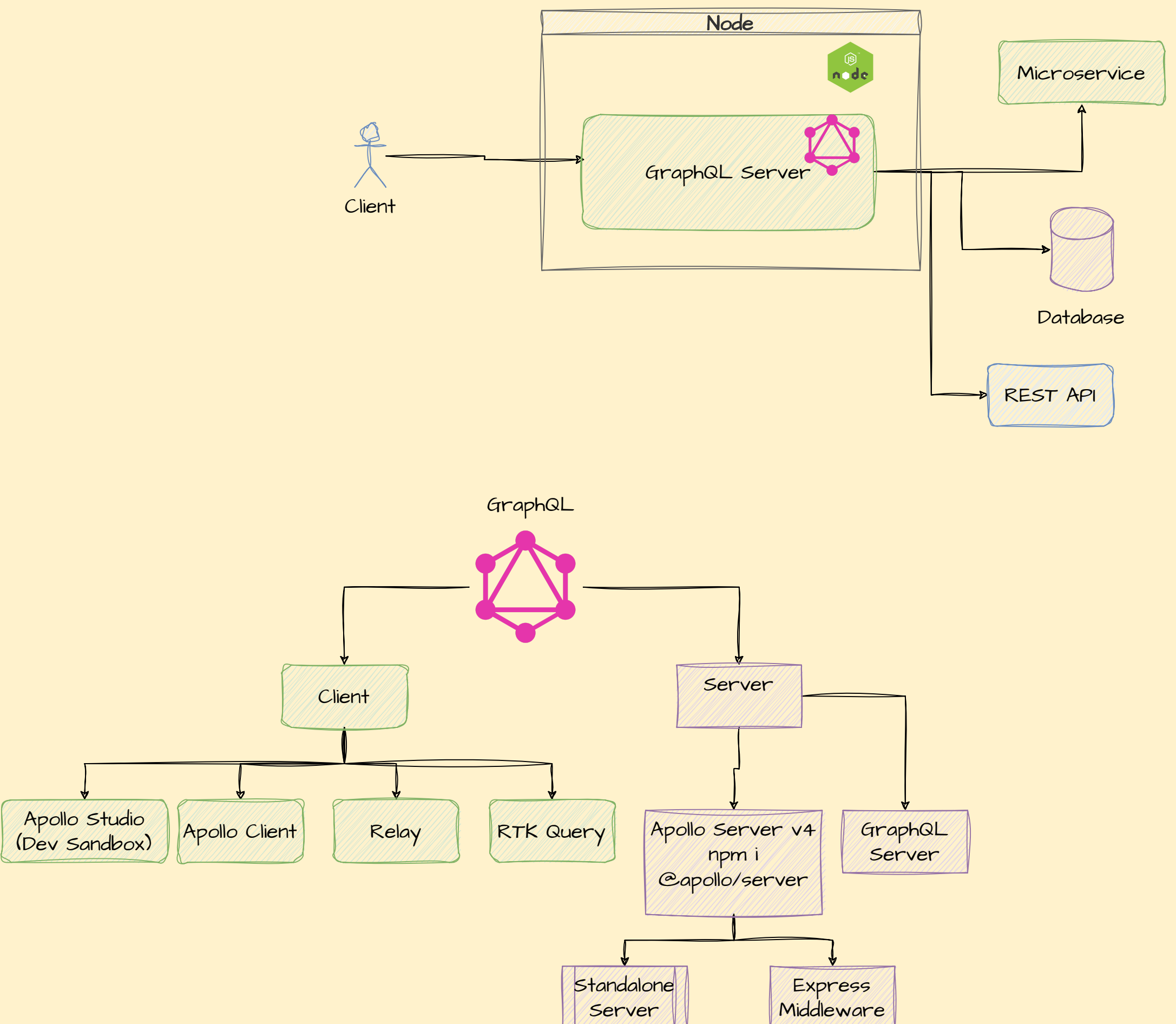
REST API

Have to have the Endpoint defined for different type(structure of data)  
Fixed structure (data)  
Related set of data required addition call (not fetched in the same http call)  
Overfetching / underfetching

VS

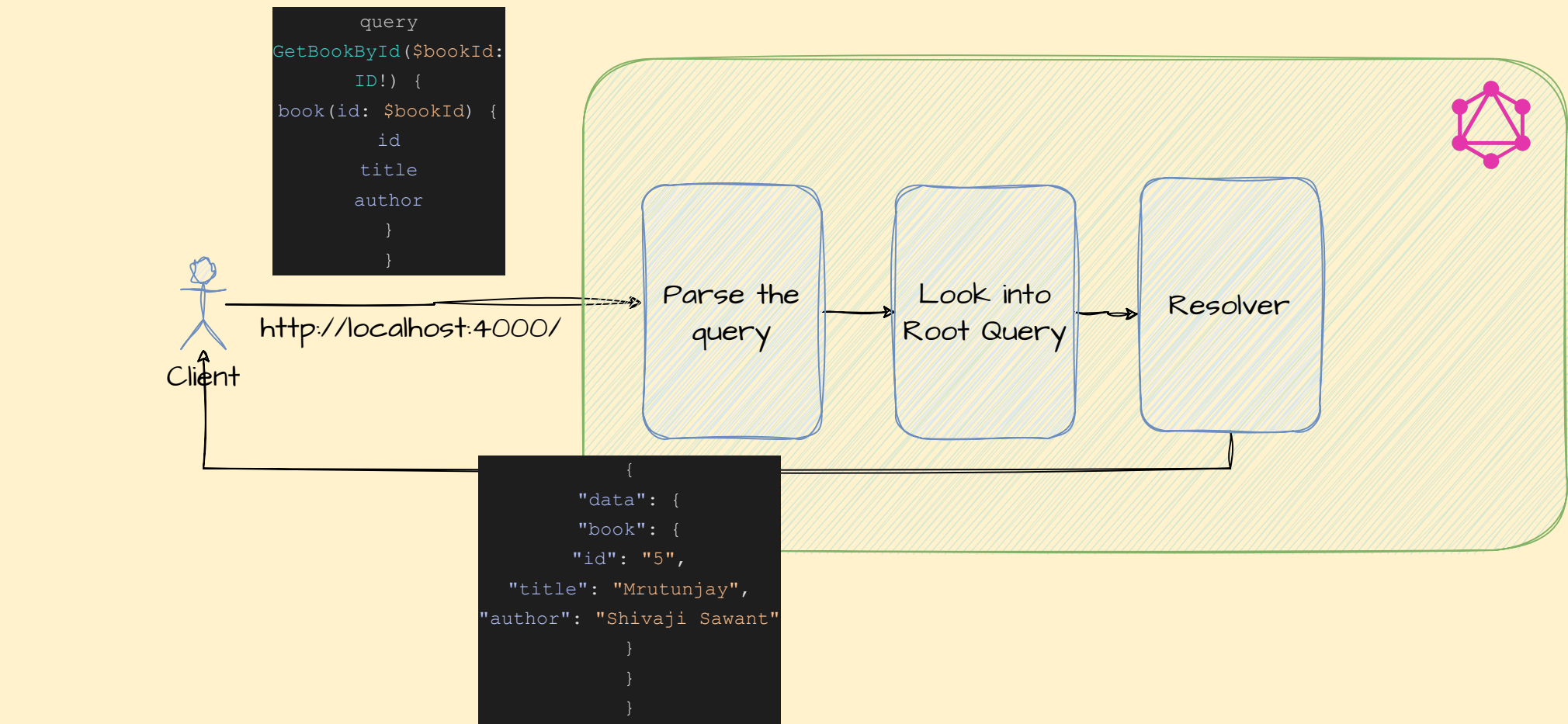
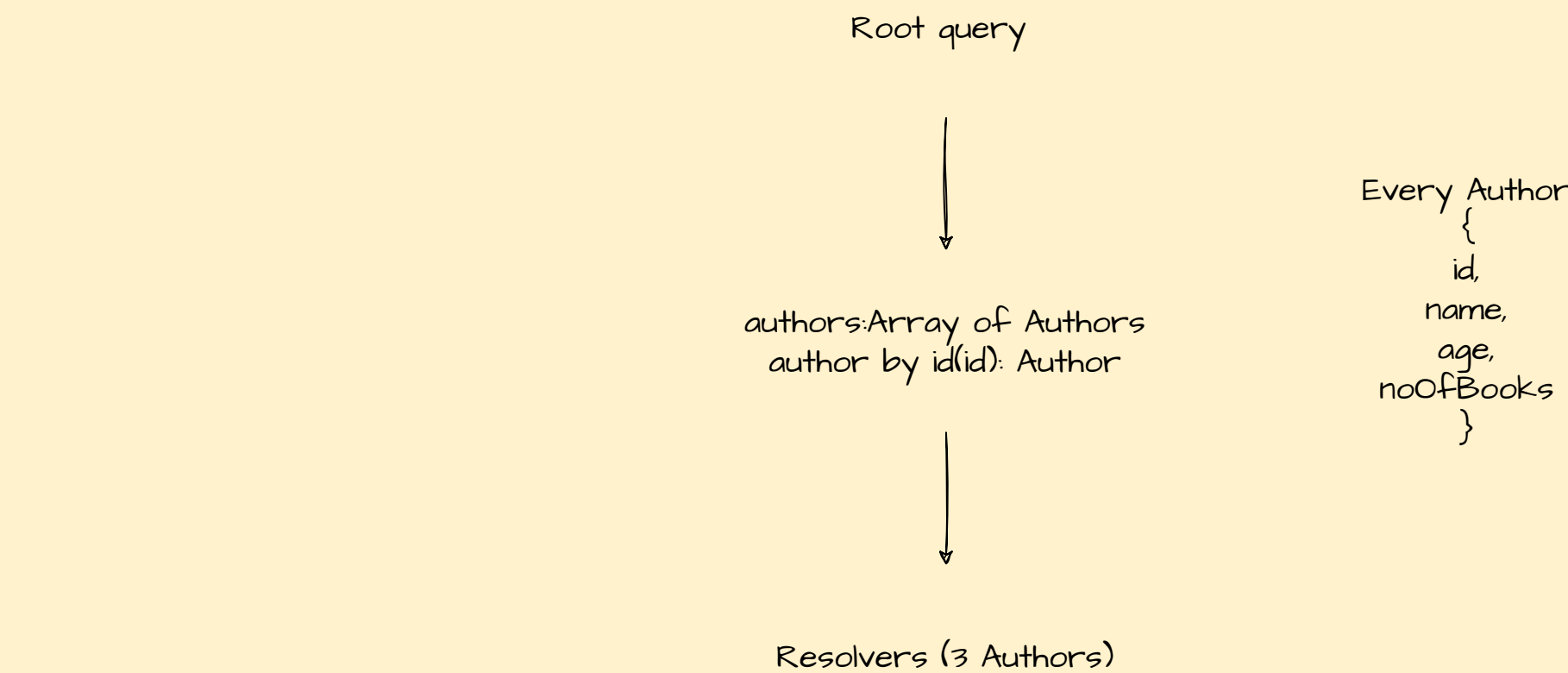
GraphQL

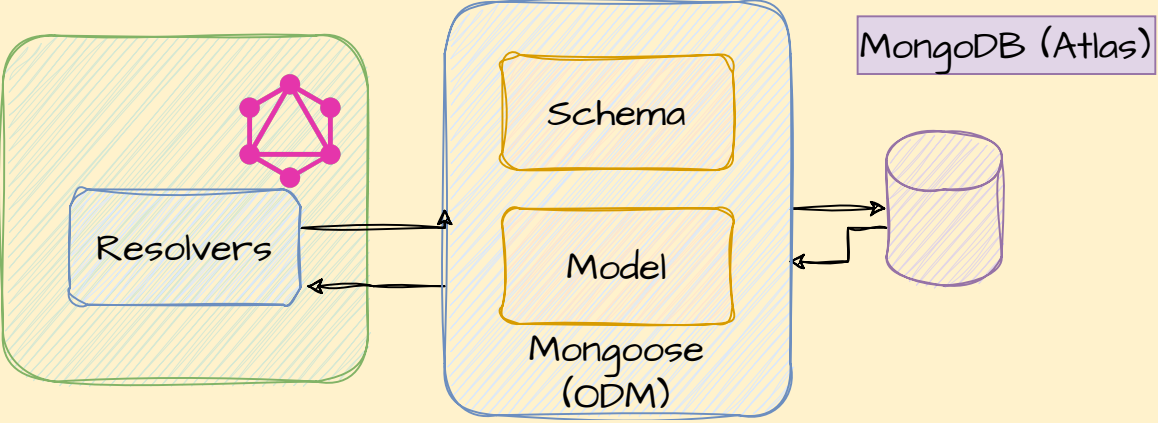
One query can return different structure of data  
Flexible structure of data (returned) - return whatever the client demands  
One query can be sufficing to send related/associated info  
No overfetching / No underfetching



GraphQL Terminologies

1. TypeDefs : Type Definitions [Type/Structure]  
2. Resolvers : Functions that populates data for a query (biz logic)  
3. Root Query : Possible queries that client can make  
4. Fragments : Fragments let you construct sets of fields, and then include them in queries where you need to  
5. Aliases : Giving friendly names for conflict resolving in a query







Client



- Gets accessed with every request
- Allows customization to the object
- Share data between resolvers

