

Conclusion /

What is good

- On the client, **Apollo** solves a lot of problems out of the box, like pagination, subscriptions, optimistic updates, garbage collection etc. No need to write a lot of boilerplate code
- Better **performance** for clients that need to fetch complex connected data
- Straightforward deprecations and **measuring usage** of certain fields, API evolution without versioning
- It allows to implement clear **domain ownership** between multiple teams while still maintaining a general schema. It unifies them and hides their complexity.

Conclusion /

What is not so good

- Quite steep **learning curve**, it takes some time to get sense of best practices and create a convenient setup. Harder to hire new people and get them onboarded
- Need to be mindful of **performance** for complex queries, on the server need to implement depth limiting etc.
- **Rate limiting** is harder to implement with GraphQL API compared to REST