Principled GraphQL

Agility Principles

Rapidly rolling out the graph and continuously adapting it to changing needs

4. Abstract, Demand-Oriented Schema

> The schema should act as an **abstraction layer** that provides flexibility to consumers while hiding service implementation details.

A large part of the value of GraphQL lies in providing an abstraction between services and consumers, so the schema should not be tightly coupled either to particular service implementations or to particular consumers as they exist today. By keeping implementation details out of the schema, it should be possible to refactor the services that implement the graph – for example, transitioning from a monolith to microservices, or changing the language in which a service is implemented – without disturbing apps in the field. Likewise, the schema shouldn't be tightly coupled to the way that particular apps fetch data. It should be possible to write new apps with minimal changes to the graph if their functionality is similar to that of existing apps.

To accomplish this, use the standard of a **demand-oriented** schema: a schema focused on providing a great developer experience to an app developer building a new feature against the existing graph. Aiming for this standard will help