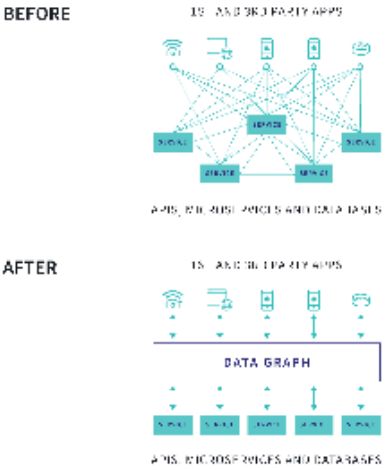
https://medium.com/codingthesmartway-com-blog/rest-vs-graphql-418eac2e3083#targetText=Both%2C%20REST%20and%20GraphQL%2C%20are,to%20deal%20with%20single%20resources.

	Description	REST	GraphQL
microservice architecture	**how well does this fit into micro-service architecture?**  The point-to-point nature of REST, a procedural API technology, forces the authors of services and clients to coordinate each use case ahead of time. When frontend teams must constantly ask backend teams for new endpoints, often with each new screen in an app, development is dramatically slowed down. Both teams need to move fast independently. Instead of a custom API for each screen, app developers describe the data they need, service developers describe what they can supply, and GraphQL automatically matches the two together. Teams ship faster across more platforms, with new levels of visibility and control over how their data is used.		

fit for purpose for DLT reporting requirements	• rapid changes in field names?
	  • Same. Breaks UI.
	• rapid changes in API output?
	  • GraphQL is more flexible
	• rapid changes in relationships?
	  • GraphQL is more flexible		

security at depth	how to ensure permission is checked for each table? Same behavior. Information about caller is passed to each micro-service and each service decided permissions about the caller.		
development phase efficiency	changing query output quickly		
ability to generate specification			
code generation support	can client side generate code from API specification		https://graphql-code-generator.com/#live-demo
server side development effort	custom code vs annotation/configuration based changes		
level of abstraction	changes required on tables or views cater for queries		
data transformation	if data transformation or data formatting required where do we do this? custom code		
asx/industry wide adaptability	if this technology that is something widely used Github now only has GraphQL.		
ability to mock during development	API based mocking for both client & server side development Should be easier with GraphQL as we do not need to create end-points.		

© 2020 ASX Limited ABN 98 008 624 691

ability to interface to other services	Can this technology be integrated with third-party tools like DV (Tibco DV)	
Performance consideration	If caching is required where do we handle this?	https://github.com/graphql-java-kickstart/graphql-spring-boot/tree/master/example-request-scoped-dataloader

http://youtube.com/watch?v=zVNrqo9XGOs

Naming matters

© 2020 ASX Limited ABN 98 008 624 691