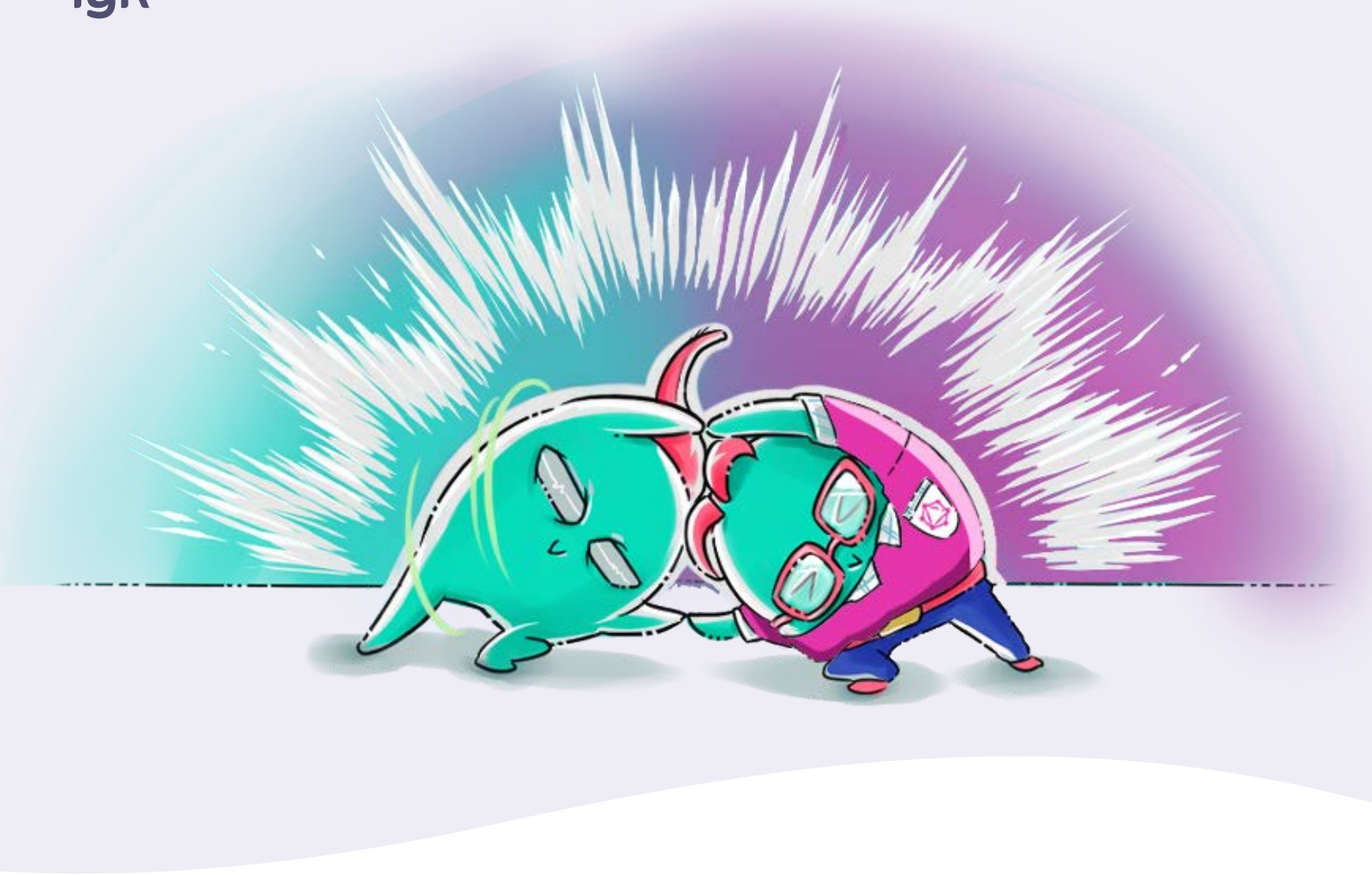

All hail Tyk's Universal Data Graph!

We live in a world where blazing a path with the most cutting-edge technologies, like GraphQL, is essential if you want your business to stay relevant. But you don't have to do it alone. Tyk's Universal Data Graph (UDG) is here to help.

The release of our Universal Data Graph has been one of the highlights of our year. Why? Because the technology has fundamentally changed the way we think. Whether we're implementing GraphQL or exposing a company's data through a single graph, the Tyk UDG provides a slick and easy solution. No new code, no new infrastructure, just bags of new potential.

When does it make sense to use Tyk's UDG?

There's a huge array of scenarios where you may want to leverage our UDG. We'll lay out three of them here to walk you through some of the possibilities, but there are plenty of other use cases once you start tapping into the UDG's potential.

Scenario I: the single schema

You have existing services (RESTful or GraphQL-based) that you want to make available in a single schema through GraphQL. What's the easiest and most painless way to do so? Through our UDG, of course!

Exposing a GraphQL endpoint that contains all of your organization's data can really expedite your front-end web and mobile development. Normally, this involves some pretty extensive work. Do you relish the prospect of creating an entirely new GraphQL backend ecosystem before your frontend team is even able to begin to use it? Do you have the budget to do so?

With our UDG, you can achieve the same outcome but with a vastly simplified process. Your developers will thank you and so will your FD.

Scenario II: balancing old and new APIs

Would you like to expose a new GraphQL API while maintaining your current RESTful APIs for existing applications? How about doing it without added build, support and infrastructure costs?

Let's say you have different apps consuming your RESTful APIs, in an API portfolio where a single RESTful API may be used by many different teams or departments. You could likely achieve multiple benefits from using GraphQL to expose your company's data. However, it's not feasible to rewrite the API in GraphQL and force the frontend consumers to adopt the new API.

Should you decide that GraphQL adoption will have to wait? Or perhaps start creating a legacy system headache by building a new GraphQL API for those ready to consume it while leaving the existing RESTful services running for the consumers that need them?

If you don't fancy scrapping the idea of using GraphQL or of running two parallel services (which would then require double the amount of maintenance and support), it's time for Tyk's UDG.

Scenario III: fear of the unknown

Is your team familiar with building RESTful services but out of the loop when it comes to knowing about GraphQL? It's only natural to fear the unknown. After all, bringing in a new technology comes with two very apparent risks:

1. Your team will take a while to learn the new technology. Learning while implementing may lead to some less than optimal coding choices. These will add to the development cost and possibly the cost of technical debt acquired with your initial GraphQL implementation.

2. It is very tough to fully assess the limitations of new technology. If your company moves full-steam ahead and adopts the technology, they may belatedly discover that it doesn't support some edge cases or the future roadmap for the system. That then means choosing between reverting to the old technology or seeking another new technology that will (hopefully) support it.

Adopting GraphQL does carry these risks, but Tyk's UDG helps to mitigate them. It can make the initial proof-of-concept and even the production implementation of the new technology faster and easier. It can reduce the cost of building with a new technology and of ownership. Not only that, but we provide our own highly skilled and experienced team to reduce the strain on your own employees. You can use our skills, knowledge and infrastructure to build what you need.

In each of these three scenarios, Tyk's Universal Data Graph can really help to accelerate your organization's goals.

The solution!

These scenarios demonstrate how the UDG approach can aid in the rapid development of GraphQL services while minimising both hassle and expense. Let’s take a look at some of the issues that it addresses.

GraphQL skills gap

Are you worried that you'll need new in-house skills to develop GraphQL services? Possibly even new employees?

With Tyk's Universal Data Graph the learning curve required to deploy GraphQL services is extremely small. There are two skills that you'll need in order to create a working endpoint:

1. How to write a GraphQL schema.
2. How to configure your data sources to satisfy those data requirements.

Both of these are extremely easy to do with our UDG. Within a matter of minutes you will understand exactly what is required to begin integrating your existing services into a single unified data graph. No coding is required in order to accomplish this and our intuitive dashboard UI allows you to complete both of these tasks with ease.

Time and budget estimation concerns

It's tough to budget for introducing a new technology when you don't know how long building, integrating, testing, and releasing into production will take.

Thankfully, because the work required to integrate and expose your data through the Universal Data Graph is so minimal, time and budget concerns are a thing of the past. The path from ideation to implementation is a very short one. This allows your team to accurately estimate the amount of time and effort required for your new endpoint to be operational and available for use. Severely cutting down on the time and budget required for your move to GraphQL equates to an increased likelihood of getting funding.

Supporting new and existing services simultaneously

Are you worried about having to double your backend support efforts in order to support both your current infrastructure and services and your new ones?

With our UDG, you don't need to be. Tyk's Universal Data Graph simply calls your existing services to retrieve the data, so you can treat it as just another client that is using them. Your support team won't need to expand its skills or numbers to support the GraphQL service, as there's no new code or service to support. You just need to continue supporting your existing services.

Technical debt from inexperience and sub-optimal implementation

If your developers are having to create new services while still learning the technology, their inexperience may result in poor choices during implementation. This can build up your technical debt.

By removing the need to implement new GraphQL services from the ground up, you can vanquish any worries of suboptimal implementations and resulting technical debt. Your team can rapidly learn to create new GraphQL endpoints within Tyk, without needing to be familiar with the nuances and best practices for creating GraphQL services. This eliminates the possibility of a shoddy implementation.

Long-term feasibility

Are you worried that the new technology might work for current use cases but not support future ones or that you might end up with increasingly complex workarounds? Or perhaps even that you will end up reverting back to the previous technology or having to find a new one?

They're valid points of concern. Sometimes technologies don't pan out. When that happens, depending on the severity and urgency, it can end up costing a lot of money and time to remedy.

With something like GraphQL, you're looking at sweeping backend and frontend changes when it comes to introducing the technology. And, indeed, when it comes to removing it. The benefit of using UDG, in this regard, is that your backend won't change at all. So if you did decide to revert back, those services still exist. All you would need to do is change the frontend code to once again utilize the previous services.

This heavily reduces the risk of adopting the new tech in the case where a fallback to the old technology may be required. It also lessens the path in which the new technology can be validated or discarded for a particular use case.

Discover for yourself

The benefits of Tyk's Universal Data Graph are immense. You can use the UDG to diminish all of the pain points that GraphQL adoption may bring in terms of build, support and risk.

Tyk's UDG can power your organization's journey into adopting and reaping the rewards of this fantastic technology. Want to try it out for yourself?
Sign up and get started today. In just a few clicks you can be GraphQL ready!

TYK LONDON

87a Worship Street, EC2A 2BE, United Kingdom

info@tyk.io | +44 20 3409 1911

TYK SINGAPORE

Found8, High Street Centre, 1 North Bridge Road,
#08-08, Singapore 179094

info@tyk.io | +65 6813 2083

TYK ATLANTA

22 Technology Parkway South, Peachtree Corners, GA30092

info@tyk.io | +1 (404) 4511123

ISOQAR REGISTERED UKAS MANAGEMENT SYSTEMS 0026
Cert.no: 1519
ISO 9001

PCI Security Standards Council

HIPAA COMPLIANT

www.tyk.io