



# Guide: 5 key considerations when migrating to a new API management platform

# Guide: 5 key considerations when migrating to a new API management platform

Are you thinking about migrating from an existing API management platform to Tyk? In this guide, we explore 5 key considerations to help you look at your potential migration from every angle and consider whether using an API Management Platform, like Tyk, is the right path for your business.

- ✓ Cost
- ✓ Vendor lock-in
- ✓ Standards and compliance
- ✓ Expertise
- ✓ Support



## 1. Cost: calculate the total cost of migration inclusive of one-time migration cost and on-going costs

Based on Tyk's experience, cost is one of the key drivers for most migrations. Cost tends to be a case of where you are now versus where you see yourself over the next few years, both from a strategic perspective and from a technology perspective.

The different types of products that are available today have different cost implications. If you're looking at an on-premise or self-managed setup, then you're looking at the cost of paying as you go to manage and maintain an infrastructure, as well as purchase the software. At the other end of the spectrum, with SaaS products, the service provider takes care of the infrastructure for you.

Deciding where you land on this spectrum is a question of whether you're looking for more control or more convenience. Do you want the control to extend your feature set? Are you looking to control how exactly deployments work and how the product can be customised? Or are you focused solely on making your APIs available to your users and on the development of your product?

The on-premise versus SaaS consideration is a key driver of migration cost. There's also a middle ground in terms of cost, which is a hybrid of the two.

Also, consider how accurately you are able to forecast demand and identify the expected frequency of calls to your APIs. How will you manage periods of spikes? What does business as usual look like? Understanding the frequency and volume of calls will enable you to choose between different cost plans like a pay per call plan and a fixed cost plan.

Migration on its own is a one-time activity, however, the cost of the project isn't limited to that single activity – it's the cost of ownership on an ongoing basis.

And on that note, remember to include projected costs of making the migration happen. It's not just about the cost of acquiring a new product or even of the migration activity itself, but about what the whole journey, including management and support, looks like. This could go up or down depending on the kind of product you choose, how much expertise you can access in-house versus outsource, and so on.



## 2. Vendor lock-in: convenience of ecosystem buy-in versus the ability to architect a best-of-breed solution

We tend to see very different ideologies when it comes to vendor lock-in. Ultimately, vendor lock-in is neither 'good', nor 'bad', instead it comes down to what your strategic direction is and how you see the next few iterations of your product unfolding.

There are some obvious benefits of vendor lock-in, particularly when it comes to extending services within that ecosystem. AWS, for example, makes it easy for you to manage a cloud native application with an API gateway, hosting (EC2, ECS) and serverless, event-driven interactions (AWS Lambda). You also have the added benefit of managing everything from one single place, and you can easily bolt on additional products and services if you need a 'best of breed' solution (e.g. Tyk API Management available on AWS).

The downside, though, is that naturally you lose some control. If you want to move away from that specific vendor for any reason – perhaps you're no longer happy with their offering or they've discontinued a part of that offering – then you risk being so integrated into that ecosystem that movement away to something new is going to be highly challenging. It's going to take a lot of time, a lot of effort and a lot of money. Sometimes it even means redeveloping the entire product.

Being vendor-agnostic also gives you more control over how you view the different components of your own stack. You can start looking at it from a modular perspective, going after ‘best of breed’ components in order to meet your specific requirements rather than being tied in to specific products.

For example, you can have your gateway hosted anywhere you like. You can opt for the infrastructure offering that you prefer and seamlessly integrate your gateway with it. You can also expand and add new features rapidly, replacing components quickly and painlessly. Tyk is a good example of this – the product works well with things like with external analytics, and CI/CD pipelines, for instance. If you want to have a specific dashboard and customise it in any capacity that you want, you can do that too. This is because all of Tyk’s services are made available through APIs and APIs don’t care which systems you put in front of them!

If you are vendor-agnostic and have a complex product to manage, then you may have to think about a lot of different components. However, the flexibility of knowing that you’re not locked into something for the foreseeable future is a powerful thing to have at your disposal.



### **3. Standards & compliances: understand your industry-specific requirements**

Understanding your own market is key to this: your vertical and your region. Many standards and compliance obligations are specific to a certain region or industry. In Europe, for example, GDPR is a huge consideration in terms of how you handle personal data. So if you’re looking at an API management software that’s not GDPR compliant, in most cases that will mean a big headache when it comes to ensuring your chosen tool follows the required regulations.

Compliance and standards can be fairly clear-cut. So if a product you’re looking at doesn’t meet them, it’s probably a good idea to rule it out. Of course to do this, you need to be aware of what your industry mandates, and any regional variations. The financial industry, for example, has reams of requirements around how to handle the most basic of data transactions. Meanwhile in the US, healthcare organisations have to comply with the requirements of the Health Insurance Portability and Accountability Act.

In terms of your business, you need to be aware of the standards that apply first, then find an API management system that meets those needs. An API management platform like Tyk is an integral part of how your data is handled, managed and stored, and any data that passes through has to do so in a secure way, while any information that is stored has to be kept in such a way as it doesn't violate any regulatory requirements.



#### **4. Expertise: take stock of your team's capabilities and understand what external support you need**

There are two elements to consider when it comes to expertise: your in-house team and external support (we'll take a look at support separately).

First, consider what kind of expertise you have in-house. Do you plan to take on the whole migration process yourself or are you looking to engage a partner? Is the new system something that you're happy to manage, from an infrastructure perspective as well as the APIs themselves?

You need to consider how much responsibility you want to take on and whether you have the skills in-house to do so. If you have the expertise available and want to retain a certain level of control, then a self-managed, on-premise solution could work well for you. You can customise it and extend your offerings, but you'll also be responsible for solving any challenges that arise.

On the other hand, if you don't have the expertise in-house or simply don't want to take on that much of a challenge, then a SaaS product could be the answer. That way, you're not really focusing too much on the specifics of running the product or infrastructure, but on using the product in a way that enables your business to succeed.

It's also worth bearing in mind that when you migrate to a new product, you need to familiarise yourself, not only with the new platform, but also the terminologies associated with it. If you're taking the in-house approach, then you'll need a deep dive into the system's details alongside scenario planning for various situations that may arise. If you're opting for a SaaS approach, it's still a good idea to build up a knowledge base, but a more important question becomes about who you contact should an issue arise. A baseline understanding of what's gone wrong and why, though, will always help.



## 5. Support: understand the different tiers of support and how critical issues are handled

API Management platform providers offer a range of support levels, usually governed by service level agreements with defined response times. So how quickly you want answers and, in the case of mission-critical services, fixes, will affect the kind of support contract that you choose.

Choosing support means greater comfort during, and after, your migration - knowing that there's someone to turn to if something goes wrong. If you have in-house expertise, that's great, but from a product perspective, it's always good to know that you have a product expert on hand.

This means that part of your due diligence should be considering the support that your solution provider delivers. You can assess this through testimonials and case studies, as well as through looking at the company culture and knowledge base. The transparency of the support process here is important, as it will show how the vendor provides support or tackles critical issues as and when they come up.



## How does Tyk tie together these five key considerations when talking to potential users who are considering a migration?

Tyk does this on two levels. Firstly, when someone shows interest in the product initially, we work to understand the client first and foremost, getting an understanding of their business needs and diving into the problem from a technical engineering perspective. We walk the potential user not just through our own process, but also through the migration process more broadly, including the five elements we've discussed today.

The second element relates to the system integration partners with whom we work. These service partners are infrastructure designers and builders who take a similar stance to Tyk in terms of what to consider when migrating from one solution to another. As most migrations do have numerous components, they help work through the whole implementation process and not just the API management component.

Overall, it's about looking at the proposed migration solution as a whole and considering not only each of these 5 elements, but also how they all blend together as part of the process.