

The top ten considerations of architecting APIs

By Jorge García de Bustos Technical Presales Consultant
Godel Technologies

Take it from us, APIs aren't straight forward. There are plenty of pitfalls to side-step when considering how best to open up your system to the outside world. Here we look at the top 10 considerations we consider when architecting APIs.

1. Are you built for APIs?

Firstly, does your system allow for an API – is it architected in such a way that an API can be built around it? For APIs to be architected correctly your software platform must be modern enough to cope with the demands APIs will put it under.

In these circumstances those applications that are built in a microservices architecture will be built for best-use and longevity. If you have legacy apps, you might want to future-proof by moving them to the cloud first.

2. Don't forget your SLAs

Next think about your audience and the commercial model you will choose to have with them. Will you be opening the API up to the world at large or just to pre-registered users? Will you monetise it on a transaction basis or make it free to use and rely on an increase in business to fund the work?

It's important to consider the business decisions that the APIs must support so that commercial decisions drive the technology, not the other way around. Whatever you decide, organisations will have expectations of the support you deliver for your API and you will need SLAs to clarify those.

3. Architect your API.

Your API is essentially a contract with the outside world, so you really need to pay attention to the details of that contract. Your API must be designed paying attention to every single clause – the minutiae is important – making sure that the API is 'coarse-grained' so that your system doesn't have to process lots of unnecessary requests.



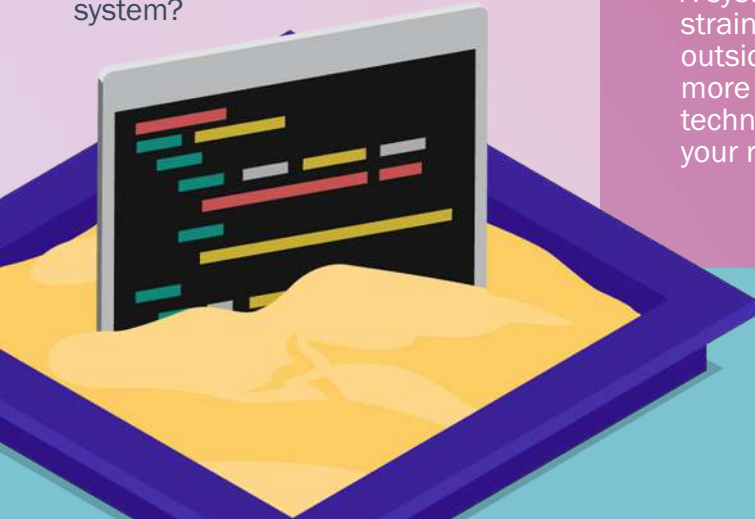
4. Pre-empt misuse

It goes without saying that if you're opening up your system to the outside world, security is an instant consideration. How can you protect your API?

One measure is to consider your rate limitation – how many requests per second are you willing to accept per unit of time from users?

5. Geography

What will be your end-points of distribution? Will you have multiple end-points, one in the US and one in the UK for example? Where are you going to host the system?



6. Play in the sandbox?

Be sure to think about a development sandbox. Engineers like to know that they can test their applications and develop new features in the safety of a sandbox before going into development – you should consider providing one.

7. Take the strain

Probably the most obvious technical consideration is that your API must withstand high-loads – it should still be able to perform with many concurrent requests from many different partners.

A system that falls over under strain is open to criticism by the outside world and can impact more widely than just technologically – it can damage your reputation and your brand.

8. Company culture.

This is not to be underestimated – the culture in your company must be ready to cope with and support an API. Developers have to be open to feedback and ready to learn quickly from any mistakes.

The considerations don't stop once you've implemented your API, so it's worth thinking ahead so as not to fall into the most common post-implementation traps.

“Developers have to be open to feedback and ready to learn quickly from any mistakes.”

9. Beware the ‘Common Law Features’

Once you have a working API, you've made a contract with the outside world and modifying that contract becomes challenging. Versioning your API is the best way to deliver new or significant changes or features so as not make changes that break the code.

Common Law Features are those bugs and defects that have become expected functionality – they've been around so long that partners have found a workable way around them and now expect them to be there. If you fix the bugs, the work-arounds might stop working.

Which leads us to our final point...

10. Software developers have strong opinions

Remember that your users are developers and they are opinionated creatures. Every time you do something that is not standard, they will be sure to let you know their displeasure.

By providing high-performing, agile dedicated teams of software engineering expertise, Godel is helping its clients accelerate delivery of their API development and integration projects. If you have API projects to deliver, get in touch with Godel.

Get in
touch with
Godel

