



# **BAD** **API** **OBSERVABILITY**

**POCKET GUIDE**



# The bad API observability pocket guide



Copyright © 2024 Tyk Technologies.

All rights reserved. This book or any portion thereof may not be reproduced or used in any manner whatsoever without the express written permission of the publisher except for the use of brief quotations in a book review.

[tyk.io](https://tyk.io)

Bad API observability can lead to inefficiencies, frustrated customers and missed revenue opportunities. Flick through our pocket guide to ensure you don't tumble down the slippery, bad observability slope.

Prefer to watch content rather than read? Then let Tyk's Sonja Chevre [talk you through](#) the pitfalls of bad API observability.

## Why worry about observability?

---



APIs have become the building blocks of the digital world. You never start a new project from scratch; you use APIs to build upon. So when anything fails in the API that you are using, your whole user experience crashes. When it's slow, your whole product is slow.

**Sonja Chevre**, Group Product Manager, Tyk

Well-implemented API observability can help you:

- Surface insights to improve the developer experience
- Deliver consistent performance that keeps end users happy
- Improve overall reliability while working with different styles of APIs
- Identify errors and their causes to aid speedy, effective troubleshooting
- Ensure that information is shared and leveraged across different teams and tools
- Provide a comprehensive view of how APIs behave under different scenarios
- Identify performance bottlenecks and architectural problems
- Give a complete picture of the journey of all your users

Bad observability means you miss out on some or even all of this.

**Bad API observability is bad for business.**

## Bad API observability pitfall #1: You forgot your users

---

Your APIs have two kinds of users:

1. Developers who are integrating your API into their systems
2. End users who interact with apps or services that rely on your API

### The problem

Too much focus on APIs as a technical interface and not enough on user experience.

### The result

You don't understand how users interact with your APIs or where their pain points lie.

### What you gain by fixing it

Avoiding this bad observability pitfall means you can understand:

- How long it takes developers to make their first successful API call
- Which errors are slowing them down
- How your releases are impacting requests from your customers' applications or services
- Which versions of your APIs customers are using
- Which are your most popular endpoints
- Whether all customers are getting the same performance and error rates

You can use this information to enhance adoption, improve *Time to First Hello World* rates and improve the user experience. You'll have happier developers integrating your APIs and more satisfied end customers. Your finance director will likely be pretty pleased with you, too.



Nowadays, companies don't sell a user interface, they don't have an app, they have APIs. APIs are a way to make money... and if you're selling APIs, you need to make sure that they're reliable. If those APIs go down, your company will lose money.

**Sonja Chevre**, Group Product Manager, Tyk



## Bad API observability pitfall #2: You rely on monitoring alone

---

API monitoring uses automated test requests and response verification to check the health of your APIs. Great. Or is it...?

### The problem

API monitoring shows that your APIs receive requests and send responses. If failures occur, it doesn't provide any insights into why. It also only tests some potential use cases, not all.

### The result

You lack insight into what's happening behind the headline uptime and failure rates and miss out on observing real-world use cases and edge cases.

### What you gain by fixing it

When you expand your monitoring by adding observability into the mix, you can:

- See what's causing any failures
- Understand API performance based on all use cases, not just a few possible ones

These insights can help you deliver greater performance reliability that benefits all your consumers.

## Bad API observability pitfall #3: You use API access logs for troubleshooting

---

Access logs are records of requests and responses. They capture the requestor's details – IP address, HTTP method, requested endpoint, status codes, etc.

### The problem

Using access logs instead of distributed tracing means you need a fast, efficient way to identify the root cause of performance issues.

### The result

You lose countless hours – days, even – to troubleshooting API issues every time a performance problem arises.

### What you gain by fixing it

Enabling distributed tracing as part of your observability strategy means you can track a single user request from the gateway to the upstream services and their dependencies, so you can:

- Reduce troubleshooting from hours (or longer) to seconds
- Enjoy a deep understanding of what is happening in your system
- Give your team hours of their time back

Moving from inefficient troubleshooting to an effective, speedy solution means you can free up your team's time to focus on proactive development, not fighting hidden fires.

## Bad API observability pitfall #4: You lack coordinated data and insights

---

Coordinating and integrating your teams' different tools and data sources can be tricky.

### The problem

A lack of coordination means a lack of joined-up insights – but forcing everyone to use the same tool drives inefficiency, so that isn't the answer.

### The result

You open the door to inefficiencies and a lack of clarity, fuelling poor decision-making.

### What you gain by fixing it

Using open standards, such as OpenTelemetry, means you can:

- Share and leverage information and insights across teams and tools
- Make sound, data-driven decisions based on the full picture
- Enjoy efficiencies in gathering insights, saving time and effort

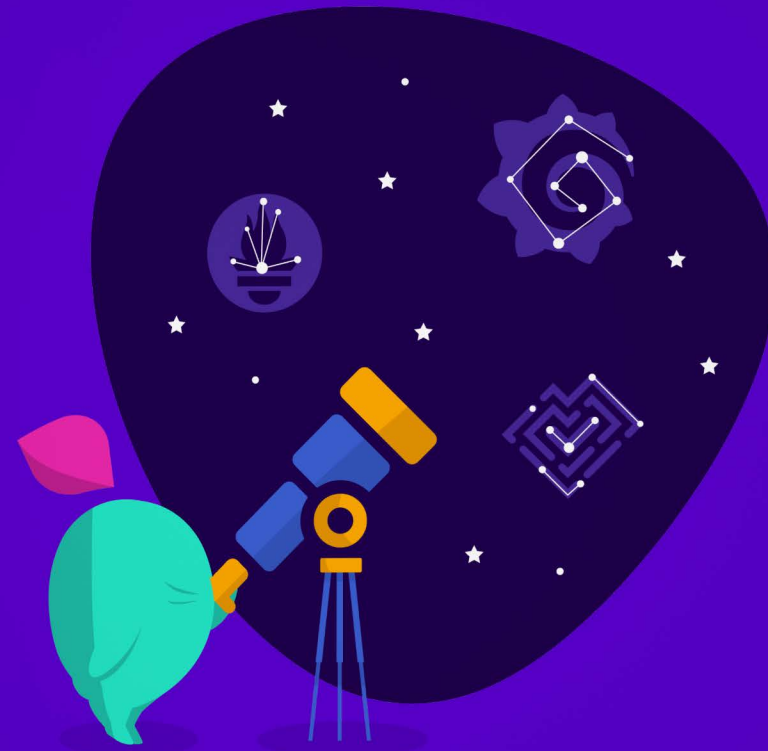
**Remember: your DevOps and product teams should rely on the same error rate when assessing the performance and reliability of your new API products.**





The architects defining the general architecture and guidelines, (using REST, gRPC, GraphQL, and web sockets), the API platform team responsible for managing the gateways, (including the authentication, authorisation and rate limiting), the developer team implementing the API and the product manager responsible for adoption and monetisation all have to work together to create a valuable API experience. Modern observability can help them.

**Sonja Chevre**, Group Product Manager, Tyk



## Bad API observability pitfall #5: You try a 'one-size fits all' approach

---

Whether we're talking t-shirts or API observability, one size does not fit all.

### The problem

Your observability strategy doesn't account for different API architecture styles' different observability needs.

### The result

You end up missing insights or draining time collecting data in different ways and then trying to tie it all together—or both.



### What you gain by fixing it

When you account for the different observability needs of REST, GraphQL, gRPC and other APIs, you gain:

- Full insights into overall reliability and performance
- More efficient observability processes

You can then add different styles of APIs into the mix, however best suits your needs, without worrying you'll be missing any valuable insights.

## Bad API observability pitfall #6: You only use observability for production

Why leave pre-production out in the cold?

### The problem

API observability is only implemented at the production stage.

### The result

You miss valuable insights that could improve performance prior to production.



### What you gain by fixing it

When you use distributed tracing across the API development lifecycle, you can trace the flow of requests and responses across different services, gaining:

- A comprehensive view of how each API behaves under various scenarios
- Easy identification of performance bottlenecks
- The ability to detect architectural issues, such as N+1 problems in GraphQL queries

With observability carrying across the API lifecycle, you can use the data as additional validation before promoting an API to the production stage.

## Bad API observability pitfall #7: You don't start tracing at the API gateway

---

Tutorials about observability usually start the instrumentation process at a microservice level. Of course, it's great to have detailed insights into your microservices. But a lot will happen at the API Gateway level when dealing with APIs.

### The problem

Tracing doesn't start at the gateway level.

### The result

You miss out on a whole heap of data, so you make decisions without seeing the big picture.

### What you gain by fixing it

By starting distributed tracing at the gateway level, you gain a clear entrance point and a complete picture of the journey of all your users, including:

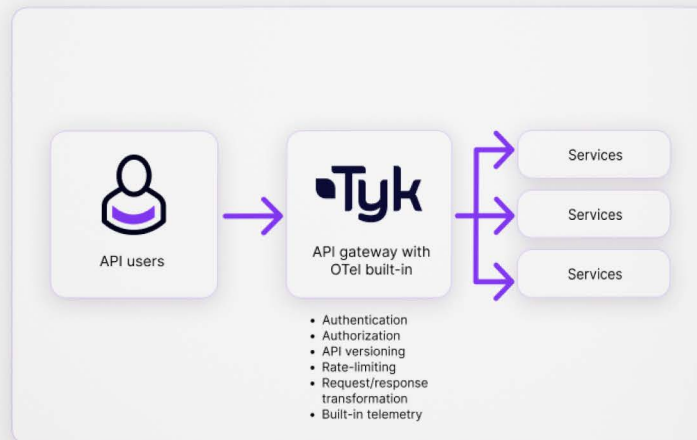
- Capturing data about all user transactions, including those that never reach your microservices because of rate-limiting rules, an authentication problem or a caching mechanism
- The ability to make decisions driven by comprehensive data rather than partial insights

Remember to choose a modern API gateway with built-in support for OpenTelemetry to get the best out of API observability.



## Turn bad API observability...

Tyk can take your API observability to the next level. You can [enable our built-in OpenTelemetry in just two steps](#), and we'll fit flexibly into your current infrastructure and preferred tooling. You can flow everything through Tyk...



## ...into good API observability

...or export observability data to the tools of your choice:



Get ready to see your APIs like never before!

## Elevate your API observability skills

Sign up for our free on-demand API observability fundamentals course, and learn how better API observability practices can:

- Help your teams create more efficient API platforms
- Rapidly reduce your dev team's troubleshooting time on their APIs
- Tackle the true cause of your API issues, not just the symptoms
- Eliminate the 'blame game' between teams and vendors
- Unlock more value from your API products.

The course features 20+ speakers delivering insightful sessions across two parallel tracks – one focused on decision-makers and the other on implementers. And that's not all.



By completing the course, you'll unlock exclusive access to the API observability fundamentals certification. Be among the first globally to earn this credential, validating and elevating your skills in the ever-evolving field of API observability.

[Sign-up for free](#)

