APIFORTRESS

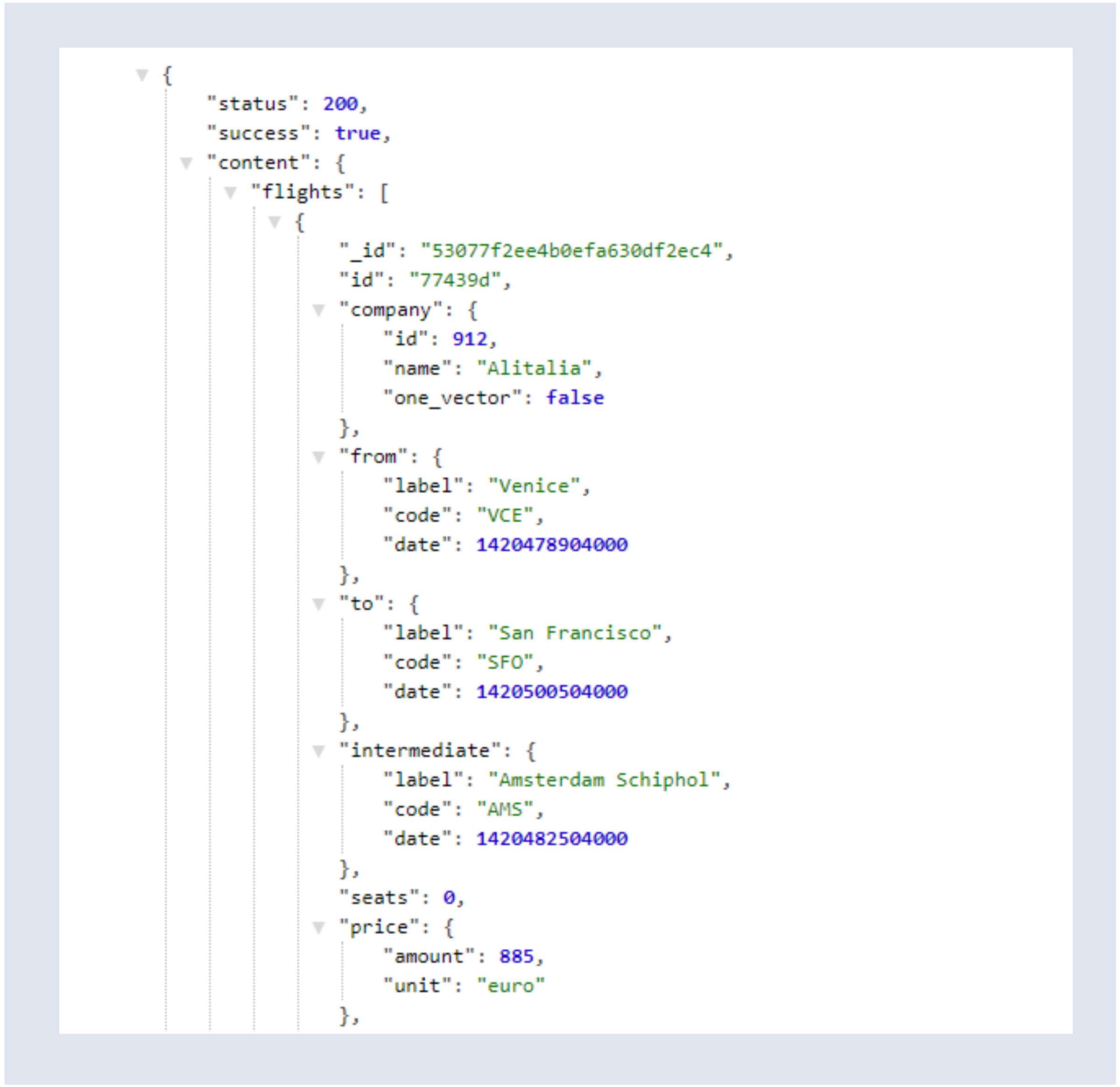
API Monitoring
A False Sense of Security

Patrick Poulin
patrick@apifortress.com

Uptime vs Functional Uptime

One of the larger issues we see is that people use “uptime” but can mean many different things, at different levels. So we prefer to call proper API uptime monitoring “Functional Uptime.”

Many companies we work with set up monitors that simply make a call and look for a Status 200 OK. An API is very verbose and detailed. So a proper monitor should be analyzing the entire payload, every object, and the data associated.



```
{
    "status": 200,
    "success": true,
    "content": {
        "flights": [
            {
                "_id": "53077f2ee4b0efa630df2ec4",
                "id": "77439d",
                "company": {
                    "id": 912,
                    "name": "Alitalia",
                    "one_vector": false
                },
                "from": {
                    "label": "Venice",
                    "code": "VCE",
                    "date": 1420478904000
                },
                "to": {
                    "label": "San Francisco",
                    "code": "SFO",
                    "date": 1420500504000
                },
                "intermediate": {
                    "label": "Amsterdam Schiphol",
                    "code": "AMS",
                    "date": 1420482504000
                },
                "seats": 0,
                "price": {
                    "amount": 885,
                    "unit": "euro"
                },
```

Think of functional API monitoring this way: Imagine you are a parent, and you are concerned about your child’s education. How do you measure their progress? Would you use homeroom attendance as your primary insight?

Or would you use standardized test results? By standardizing testing of your APIs, you ensure coverage extends to every object, every bit of data, positive and negative testing, and data-driven testing. That's a proper monitor. But it's difficult or impossible for many companies to standardize testing for a variety of factors – one of which involves their monitoring tools.

Uncaught Downtime & Security Breaches

An API that is “up” but functionally down is the most expensive problem a company can deal with: a false-positive. It may cause you to lose customers, reputation, and revenues, as well as potentially expose sensitive data for months to cybercriminals.

Security

The Internet runs on APIs, and most API failures are due simply to human error. That's so disappointing because of how easily many of these issues could be solved.

Akamai's State of the Internet report (pdf) found "83% of all web traffic in 2018 was in API traffic."

"Through 2022, at least 95% of cloud security failures will be the customer's fault."

Jay Heiser, VP at Gartner

Security

At APIFortress.com/blog, you can read about three recent API vulnerabilities that went live due to a simple lack of proper testing and monitoring. Failures to take corporate responsibility by the companies and government organizations cited on our blog resulted in huge API breaches that exposed the private data of billions of individuals.

Twitter
17 million accounts

India
1.1 billion identities

USPS
60 million personal account details

Who Owns API Uptime?

One of the key reasons that so many API defects and vulnerabilities slip past today's monitoring platforms is because these platforms are flawed at their very foundation when it comes to monitoring APIs. Most QA, architecture, DevOps, and product leaders, can't escape the assumption that uptime is synonymous with proper monitoring. Consequently, no stakeholder with significant API domain knowledge seems to own an API's uptime.

There are many popular monitoring and analytics platforms today. Most claim to extend monitoring coverage to API monitoring. But these platforms all suffer from critical shortcomings.

APPDYNAMICS part of Cisco

BlazeMeter

DATADOG

dynatrace

New Relic.

pingdom

Monitoring Platform Shortcomings:

1.

"Synthetic Testing"

A euphemism for minimal API testing capabilities

2.

API Privacy

Using 3rd party clouds exposes APIs

3.

Using a Different Platform

Tests are written outside of QA by stakeholders with little/no domain knowledge

4.

Test Intelligence

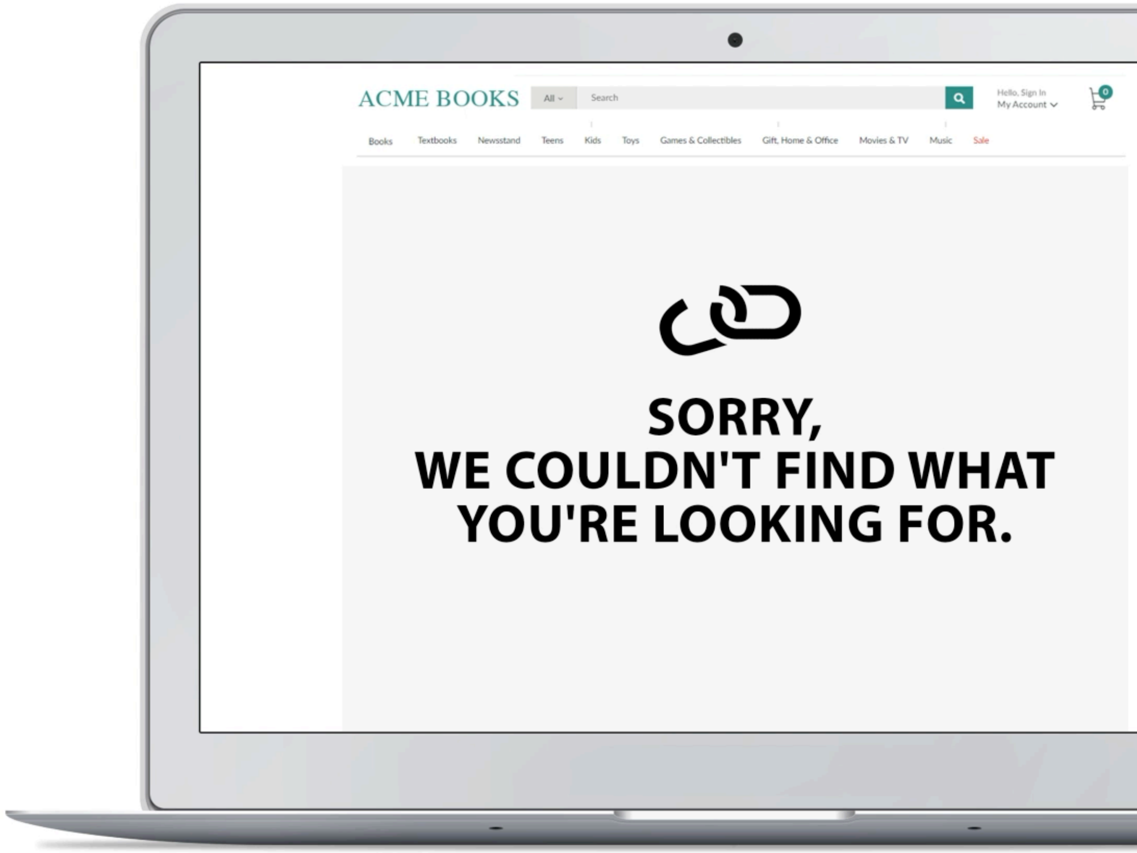
Tests must capture real consumer flows or user behavior

Problem in a Nutshell

To recap: Too many companies today turn QA over to individuals who are not testing experts. As a result, tests are often written to do little more than validate a 200 OK, or to verify that there is a payload. This is simply not enough QA for any organization trying to deliver the digital experiences that customers, employees, and the IoT demand. Next, let's look at a real-world example.

A large book publisher created two partner APIs, one for listing active ISBNs, another to get product details for ISBNs. The publisher's internal API monitoring platform was reporting 100% uptime when they approached API Fortress about getting a second opinion. Within 2 weeks of implementation, API Fortress detected hundreds of errors and 404 soft errors. Rich reporting from API Fortress allowed the publisher to quickly diagnose the errors to a problem with their API gateway's caching of endpoints.

The gateway was only refreshing the cache a few times a day, leaving gaps throughout the day when there would be hundreds of bad listings. Going forward with API Fortress, the publisher gained the ability to use a data-driven test as a monitor, and correctly verify the ISBN and product details endpoints throughout a daily state of rapid, ongoing change.

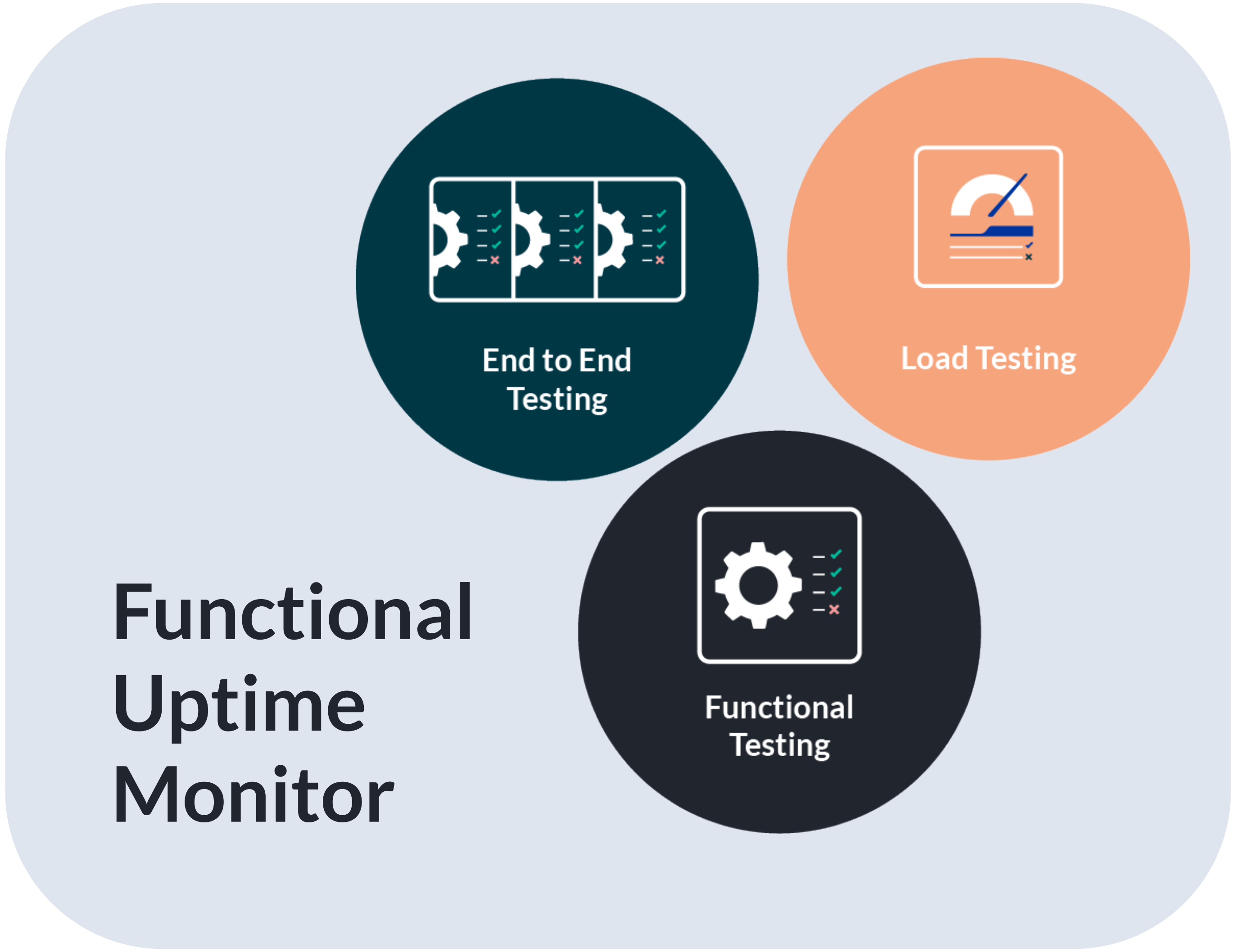


SORRY,
WE COULDN'T FIND WHAT
YOU'RE LOOKING FOR.

Uptime is **NOT** Functional Uptime

Why did the publisher's monitoring fail? It wasn't doing enough. They simply were not monitoring the use case of their average API consumer.

The best solution to close the gaps in monitoring for the publisher involves unifying functional, integration, end-to-end, and performance tests into one holistic test that can then be reused as a *Functional Uptime Monitor*. The tests are likely available for this monitoring transformation – as part of automated testing for a product release.



CI/CD

Every company that is focused on APIs, and has a CI/CD process in place, mostly likely has an automated test suite of functional and end-to-end tests for their APIs.

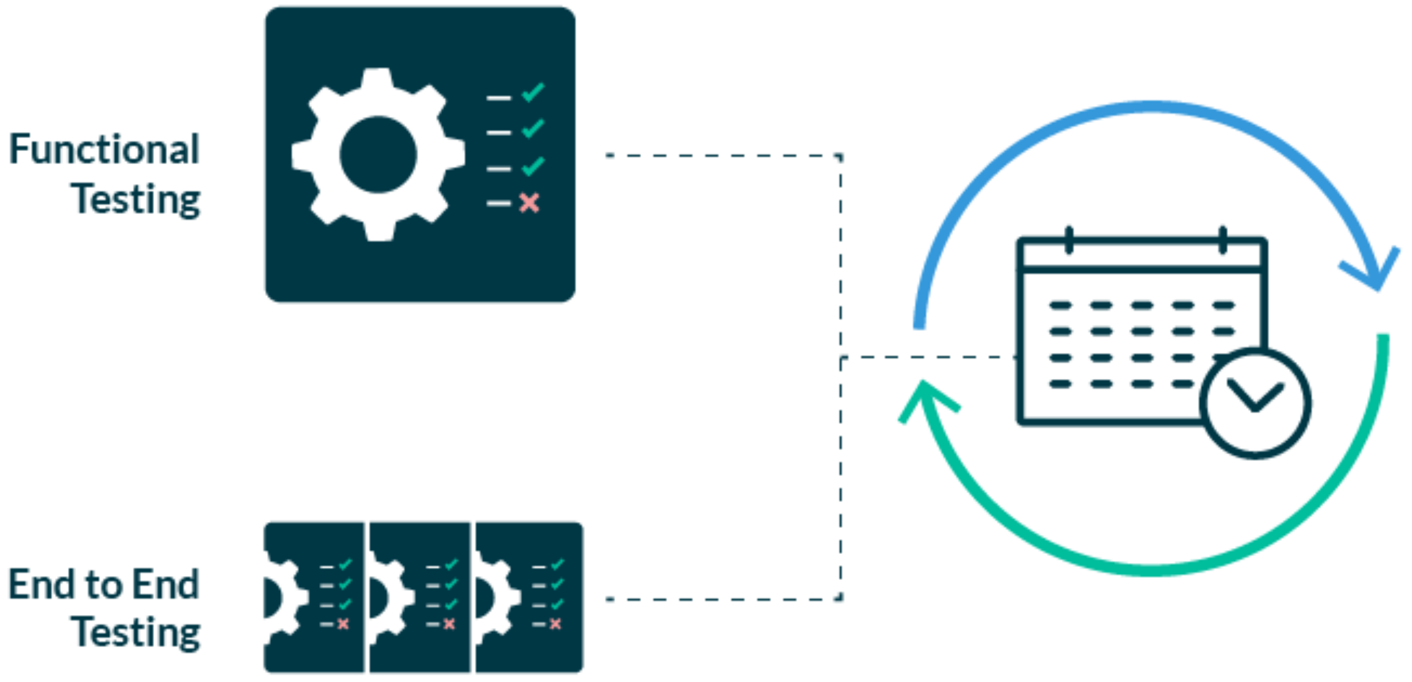
Functional Testing For APIs

If you are using one or a mixture of these tools in addition to open source libraries and languages you are in a good place already. Since you already have great tests written by your testing experts, why not leverage that domain knowledge for monitoring?

APIFORTRESS PARASOFT. SMARTBEAR TRICENTIS Tosca

REST-assured php cucumber

Use your functional and end-to-end tests as your API monitor. Run them on a schedule, and get notified when there is an issue.



Our value prop is fairly simple to understand. Any architect, engineer, developer, tester, or product owner who has been let down (or don't know they're being let down) by their conventional monitoring platform surely understands the ROI of Functional Uptime Monitoring. Visit our website at APIFortress.com to learn more and hear what analysts and customers are saying about Functional Uptime Monitoring.

You Probably Already Have Existing Functional Tests + Real Monitoring Requires Functional Testing + Uptime is not Functional Uptime =

Use your automated testing suite as your API monitor

Thank You!

www.APIFortress.com