apigee

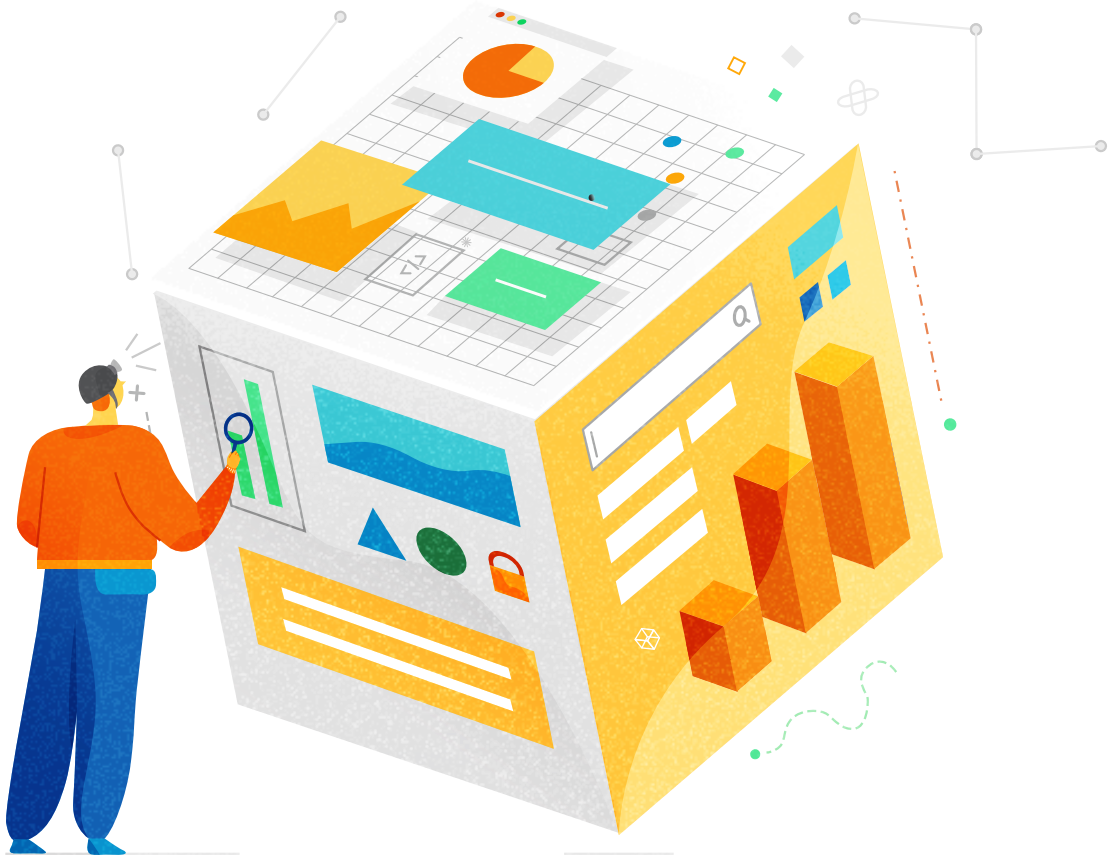
Google Cloud

1
2
3

Inside the API Product Mindset

PART 4

# Optimizing API Programs with Monitoring and Analytics



- Field-tested best practices
- Real-world use cases
- API monitoring and analytics checklist

# Table of contents

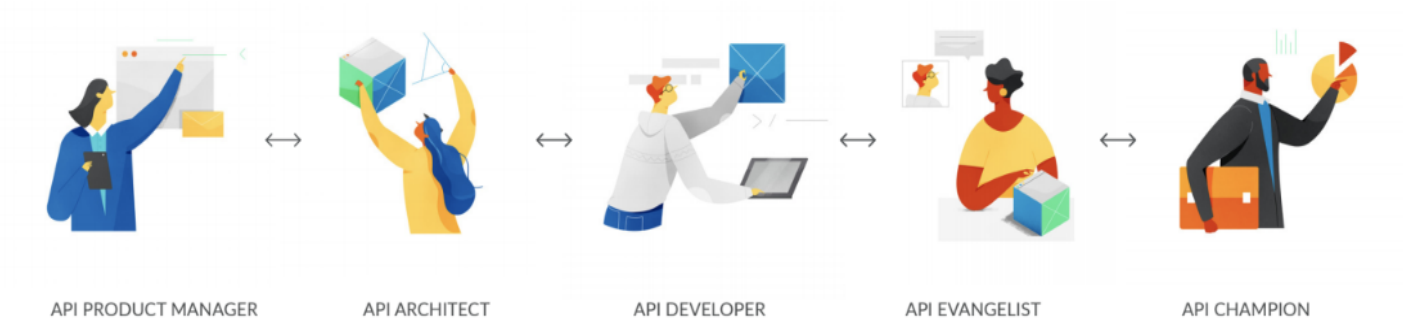
# Inside the API product mindset

APIs (or application programming interfaces) are the de facto standard for building and connecting modern applications. With APIs, a business can securely share its data and services with developers—both inside and outside the enterprise—to foster new operational efficiencies, unlock new business models, and enable business transformation.

APIs are often characterized as products for developers who build the connected experiences that power the digital economy. All businesses have valuable digital assets—functionality, data, etc.—but many of the systems that contain this value were not designed to easily connect. APIs abstract the complexity of making these connections into an interface that allows developers to easily leverage digital assets and combine them in new ways for new services.

In this way, APIs are not only expressions of a business's capabilities and points of differentiation but also the mechanisms that make those capabilities and differentiation leverageable for strategic purposes.

With this in mind, many successful organizations manage APIs like products, with full lifecycles, long-term roadmaps, a customer-centric approach, and constant iteration to meet business needs. In our experience in Google Cloud's Apigee team, organizations that treat APIs as products—as opposed to one-off *technology projects*—are more likely to realize the potential value of APIs as business accelerators.




As we explored in our recent ebook *The API Product Mindset*, an API product team carries several critical responsibilities:

- Design easy-to-use and secure APIs
- Deliver a world-class API developer experience
- Drive ongoing API improvements with monitoring and analytics
- Maximize the business value of APIs

This ebook dives deeper into a particular aspect of an API program: how to use API monitoring and analytics to optimize your API programs and digital strategies.

# Understanding API Monitoring and Analytics Use Cases: *Right Now* vs. *Trending*

Given that APIs are products, it should go without saying that monitoring and analytics are an integral part of the API product lifecycle. Would an API be a useful product if its provider were not proactively watching how it performs in order to help ensure SLAs are met and increase the odds that developers are happy? Of course not. Are users likely to remain happy if the provider does not apply analytics to improve the product and intelligently plan investments and roadmaps? Same answer: no.

Because both monitoring and analytics involve decisions based on data derived from API traffic, the two topics may blend together from certain angles—but in many ways, they are distinct. Some users of an API management platform may access only a finite number of monitoring tools without touching analytics dashboards and vice versa.

Broadly, one can think of monitoring as watching continually updated streams of data that informs real-time action to maintain system health. It is focused on *right now*—the equivalent of a heart rate monitor that lets the user know "your heart is fine" or "seek help—you're having a heart attack."

One can think of analytics, in contrast, as the analysis applied to data to create insights, sometimes in near real-time, but more typically in retrospect. Analytics are less focused on *right now* than on what is *trending*; instead of a heart monitor alerting the user to a heart attack, it's the physician making lifestyle and exercise recommendations based on a patient's activity data, dietary habits, and lab results. One extreme is focused on keeping API performance up to standard in the present tense, the other is focused on learning over time how to improve APIs, how they are being used, and how they might be leveraged for new business opportunities.

Enterprises should consider this full range of API monitoring and analytics use cases—from in-the-moment action to retrospective insight—when assessing API management solutions. It's important to see this as a spectrum, with the data collected during monitoring fueling the insights derived via analytics. It's also important that the monitoring and analytics be integrated into the core management system, rather than added via an outside service, so that monitoring and analytics can more directly lead to management action.

Operations teams, for example, need a robust system for monitoring large volumes of traffic and detecting and fixing API issues in minutes—ideally before they impact customers. They need to be not only alerted when something goes wrong, but also enabled to diagnose quickly and correct any problems. Maintaining the health of APIs is important at all times, as service degradation can lead to customer churn, revenue loss, and reputational damage—and it is particularly important during seasonal traffic spikes. Failure to handle peak traffic during Black Friday or Cyber Monday could literally cost a business millions of dollars.

But operators are only one type of user, and monitoring system health in near-real time is only one use case that an API monitoring and analytics solution should support. Indeed, some use cases do not divide monitoring and analytics discretely. For example, suppose an API features an SLA of 2 seconds, and that the business has set up alerts to trigger whenever latency slips to at least 1.5 seconds. When diagnosing the problem, the operations team might find something immediate, such as a spike in requests, but it might also find something more systemic, such as latency that's slowly been inching up over the last month. In the latter case, the situation rapidly shifts from a monitoring use case to an analytics one that may involve more team members, and the ability for the management solution as a whole to support this shift can be critical.

Possible use cases along this monitoring-analytics spectrum are numerous. API product managers will want to understand how APIs are being adopted and consumed, which specific APIs are popular with which developers, how APIs can be improved, and so on. Developers who leverage APIs will want to know how their apps are performing. Business stakeholders will want analyses of whether API investments are paying off and how future investments can best be allocated. Security professionals may require monitoring and analytics tools to effectively combat threats.

Enterprises that neglect this multiplicity of users and use cases for API monitoring and analytics may find themselves unable to easily recover in the event of a major outage, blindsided by changes in user behavior, susceptible to bad actors, and ill-equipped to measure an API program's progress and plan future investments.

# Field-tested best practices

## Know the needs of different users

API monitoring and analytics are often mentioned in the same breath and may be aspects of the same API management solution—but the term refers to a range of tools and use cases that need to fulfill the requirements of a variety of users. Key user groups include:

- The API team: The API team includes API developers who need visibility into the step-by-step behavior of APIs in order to diagnose latency problems and otherwise improve performance. The team should also include a product manager responsible for the success of the API program who needs analytics related to API adoption and usage, often sliced across dimensions such as channels, developers, or locations. It may also include an API evangelist responsible for communicating with and relaying important information to developer communities, including insights into how developers are leveraging APIs.
- Developers: Analytics can help app developers know how their apps are doing and may provide insight into how apps can be improved—which means that by sharing insights and metrics with developers (e.g., success rate, response times, and response codes), enterprises can increase the likelihood that developers produce quality apps.
- Operations administrators: Operations teams need to understand API patterns and anticipate when to add backend resources or make other critical adjustments. They are responsible for maintaining peak performance and API availability, which means they need to monitor throughput, latency, and errors in near real-time, be alerted when problems arise, and harness tools to combat bots and other security threats.
- Business stakeholders: Business stakeholders need to understand how API investments are impacting and driving digital business strategies, and they need to use insights gleaned from analytics to inform continuous strategy iteration and where to invest API dollars in the future.

SECTION SUMMARY

**Different users, different needs**

- Keep in mind the spectrum of analytics and monitoring use cases when assessing solutions.
- Remember: Monitoring use cases and solutions involve near-real time visibility into API traffic, and analytics use cases and solutions involve insights into API usage and improvements over time. Though some tasks involve aspects of both monitoring and analytics, these terms are not interchangeable and may mean different things to different users.

## Invest in proactive monitoring and analytics tools integrated into the core API management solution

It is important that API monitoring and analytics be part of the core API management system, not a black-box, bolt-on solution. Enterprises should consider white-box, contextual API monitoring and analytics tools that are part of the main management solution. When monitoring and analytics tools are integrated directly, rather than bolted on, the platform managing APIs is the same platform capturing data—which means the data can be acted on more easily. Not only do alerts and the tools that mediate them exist within the same interface, but many tasks can be more easily automated (such as automatically throttling traffic to a struggling backend) with this kind of integration.

Traditional synthetic or black-box monitoring tools that involve periodically probing the system are generally limited to reporting API availability data. These tools, which are separate from and run atop the management platform, run live checks on a predefined schedule, calling the API to make sure it is available, but they do not offer visibility into performance metrics. This means that synthetic monitoring may alert an operator to a problem, but it is unlikely to help them understand the nature of the problem. With such tools, operations teams need to manually investigate multiple systems and correlate debug sessions in order to diagnose API issues.

By contrast, an API-specific management solution that provides near real-time monitoring and enables ops staffers to quickly diagnose and resolve problems can enable teams to more efficiently and effectively keep abreast of the essential aspects of their API-powered digital business.

API monitoring should provide more than just availability metrics and alerts; it should detect anomalies and empower users to dive deeper and find the root cause of issues. Monitoring dashboards should provide at-a-glance visibility into hotspots, latency, and error rates while enabling users to drill down to find policies where faults occur, target problems, and address other specific elements that require remediation. A default view might provide quick looks at the APIs with the most traffic, the highest error rates, or the most latency, for instance. Other views should enable users to dive deeper, such as a timeline view that displays historical trends for a given API.

SECTION SUMMARY

**Real-time problems demand real-time monitoring and insights**

- Invest in real-time, API-specific white-box monitoring. Black box tools that are bolted on to the main management platform typically offer only limited options for automation and periodic system probes that can leave a business in the dark.
- Identify and resolve issues by drilling down from high-level metrics into the specific policies, targets, and code causing faults.

## Define the right alerts and use historical data to set up and tweak alert thresholds

Alerts are obviously an important monitoring feature—but they have to be configured correctly to trigger the intended results. If they fire too often in situations that don't actually require someone's attention or action, for example, members of the operations team may begin to ignore them.

When developing APIs, enterprises should perform tests to set baselines for latency and establish initial values for alerts. These values should be refined over time based on analytics and the enterprise's "signal-to-noise" ratio (the proportion of alerts that require investigation and remediation). This is an interesting case in which analytics insights flow back into monitoring, rather than the more typical case where monitoring data becomes the foundation for analytics. Alerts should always be configured to surface before a customer SLA is violated.

Enterprises should also consider including a short description of recommended actions in alerts, which can help a first responder get up to speed more quickly. Many businesses also connect alerts to their service management system to ensure that when an alert is triggered, it is tracked for follow up.

Full-featured API management solutions may also include "collections" features to reduce the number of alerts an enterprise needs to implement and manage. Rather than allowing alerts to fire for each API and potentially causing chaos, an operator can create alerts of a given collection of proxies, targets, and developer apps. To be useful, collections should include items with common characteristics, such as APIs with similar latency requirements or error rates, or developer apps assigned the same level of business criticality.

SECTION SUMMARY

**Use alerts—but don't overuse them**

- Configure alerts to notify operations administrators in the event of a problem, and to help ensure that SLAs are met.
- Leverage analytics to derive insights, tweak thresholds, and avoid sending too many alerts; if operators are alerted when there is no emergency, they are likely to begin ignoring notifications.
- Use "collections" to group alerts with common characteristics, and to limit the number of alerts and keep them relevant.

## Focus on consumption metrics

Production-oriented measures, such as the number of APIs produced, provide little if any insight into the performance of an API program. Metrics that describe how APIs are being used are more likely to be valuable. Is API traffic trending up over time? Who are the top developers? When is API response time fastest or slowest and which geographies generate the most traffic? On which devices do end users run the apps that leverage the APIs? Which APIs are generating the most revenue or driving some other business KPI? Businesses that can answer these questions are well-positioned to understand their API users, communicate progress to internal stakeholders, align around problems and goals, and ultimately grow their API programs.

Not all API management products are equipped to produce this level of detail. It's one thing for a solution to generate basic traffic metrics about an API—but it's another thing for it to generate metrics about developers and apps that call this API. It is important that any API monitoring and analytics solution be able to produce this more descriptive data, typically by allowing the enterprise to register app and developers and apply policies to proxies.

SECTION SUMMARY

**Learn from how APIs are used**

- Focus on metrics that describe how APIs are being used, such as which APIs are driving the most traffic, trending up or down with developers over time, etc.
- Connect API analytics to KPIs to align teams around internal goals and communicate the API program's progress to business stakeholders.

## Test assumptions and iterate based on insights

API developers apply policies to ensure robust app functionality while protecting backend systems—and API teams must ensure that once these policies are implemented, they function as expected.

If an API producer implements the wrong policy, for example, other developers might not adopt the API. Suppose an API developer applies an OAuth policy to a product catalog API. This policy would force end users to authenticate before getting generic information about a company's products on a mobile app.

Such friction can be a blocker to adoption—which is why it is important to analyze API patterns across a wide population of customers and iterate APIs based on insights. Moreover, just as analytics might reveal blockers, they might show that some APIs are unexpectedly popular or being used in unexpected ways, perhaps even to the extent that they could be monetized or shaped into new lines of business. They might show that an API is being used in an unexpected geography, leading the enterprise to reassess how it promotes APIs to that market and how its business operates in the market as a whole.

Analytics are where assumptions and intentions meet user behavior—and where new, smarter iterations are born.

SECTION SUMMARY

**Always look for ways to make APIs more useful**

- Avoid assuming that users will react to API design and policy decisions as intended; always be prepared to make adjustments based on how APIs are consumed.
- Leverage API analytics to understand and eliminate sources of user friction.

# Real-world use cases

## Driving the Business While Maintaining Visibility and Control

Traditional monitoring tools are limited to reporting availability information by making synthetic calls to live APIs, limiting visibility into API traffic and, because the tools are add-ons to the main management platform, limiting control when problems arise and need to be addressed. APIs are literally expressions of an enterprise's ability to do business—if they go down, so does the opportunity to do business. As Apigee customers attest, these limitations are simply not acceptable for the mission-critical use cases that many organizations demand.

In a case study, AccuWeather senior technical account manager Mark Iannelli explained how Google Cloud's full lifecycle API management platform is helping the company to support and delight developers. "With Apigee Edge, we're able to keep close tabs on who's signing up, what sort of traffic they are producing, from where—and even observe patterns in traffic activity," he said. These proactive monitoring and analytics capabilities built into the core management platform enable AccuWeather not only to ensure developers are enjoying a good experience working with the APIs, but also to identify developers who might be candidates for other API packages or options.

amadeus

> *"Knowing the number of transactions, response times on APIs, or the page travelers are spending the most time on could be invaluable for us to make informed decisions."*
>
> Xavier Gardien, Amadeus

"Understanding how our APIs are consumed is also key for us and our customers. With Apigee we are able to see this and provide them with a detailed view of API analytics," said Olivier Richaud, senior manager, API management & web services, technology platforms & engineering at Amadeus, and Xavier Gardien, head of portfolio and product management, technology platforms & engineering at Amadeus, in a blog post. "In this big data era, knowing the number of transactions, response times on APIs, or the page travellers are spending the most time on with a mobile app could be invaluable to make the informed decisions that help us maintain an edge over competitors. This also serves as a great feedback tool to closely monitor where the industry is heading."

Whether for insights, to meet SLAs, to help ensure APIs are delivering excellent developer experiences, or to help ensure business-critical services are available, enterprises rely on API monitoring and analytics for API transaction loads both small and large—and they can be *very* large.

During 2018's Black Friday Cyber Monday (BFCM) period, the number of API calls for Apigee's retail customers (excluding those who host the platform on-premises) reached a peak of 108,000 transactions per second. This sort of exceptional digital activity—and the millions or even billions of dollars it can generate—is only going to increase: API calls to Apigee's platform, which maintained 99.999% uptime through BFCM, grew 95% compared to the same five-day span in 2017.

Apigee Edge's integrated monitoring capabilities enable users to precisely find the source of an API error—whether it's a developer application, the proxy layer, or a backend target. Navigating from an alert notification, users can diagnose an error in just a few clicks; with traditional tools, this requires toggling between and cross referencing multiple systems.

# API monitoring and analytics checklist

Here are some key monitoring and analytics capabilities that businesses should consider when evaluating API management solutions:

- **Monitoring Across the API Value Chain:**
  - ☐ Provide in-depth insights into API availability and performance metrics
  - ☐ Enable users to drill down into granular details such as latencies and errors caused by proxies and backend targets
- **Precisely Diagnose Issues:**
  - ☐ Investigate API issues quickly without toggling among multiple tools and correlating debug sessions and log sources
  - ☐ Precisely diagnose the source of errors, whether in the developer application, proxy layer, or backend target
  - ☐ Tree-map views for network operations center (NOC) teams to visualize issues
- **Generate Contextual Alerts:**
  - ☐ Provide insights for users to take appropriate actions in the context of the issue being investigated
  - ☐ Facilitate grouping of proxies and targets to monitor business critical APIs
  - ☐ Support alerts by webhooks and other channels such as Slack, PagerDuty, or email
- **Gain API Insights:**
  - ☐ View overall traffic for all of the APIs in an organization, watch how developers engage with your APIs, and see which apps receive the most traffic
  - ☐ Monitor and compare traffic for specific API patterns across multiple API proxies; understand changes in API traffic relative to business, marketing, or partner events
  - ☐ Identify spikes or dips in API traffic and gain insight into what is happening around the time of the anomaly; see API proxy traffic patterns and processing times

# About Apigee API Management

The Apigee API management platform delivers full lifecycle API management to help businesses unlock the value of data and securely deliver modern applications. Apigee offers a rich set of capabilities to enable enterprises to gain control over and visibility into API traffic, including the ability to automate troubleshooting and problem resolution and to derive insights from API usage. Learn more about Apigee's API monitoring and analytics capabilities.


Now that you've finished reading, why stop learning?
Visit the Apigee website for more.

apigee

Share this eBook

on social

with a colleague

Google Cloud