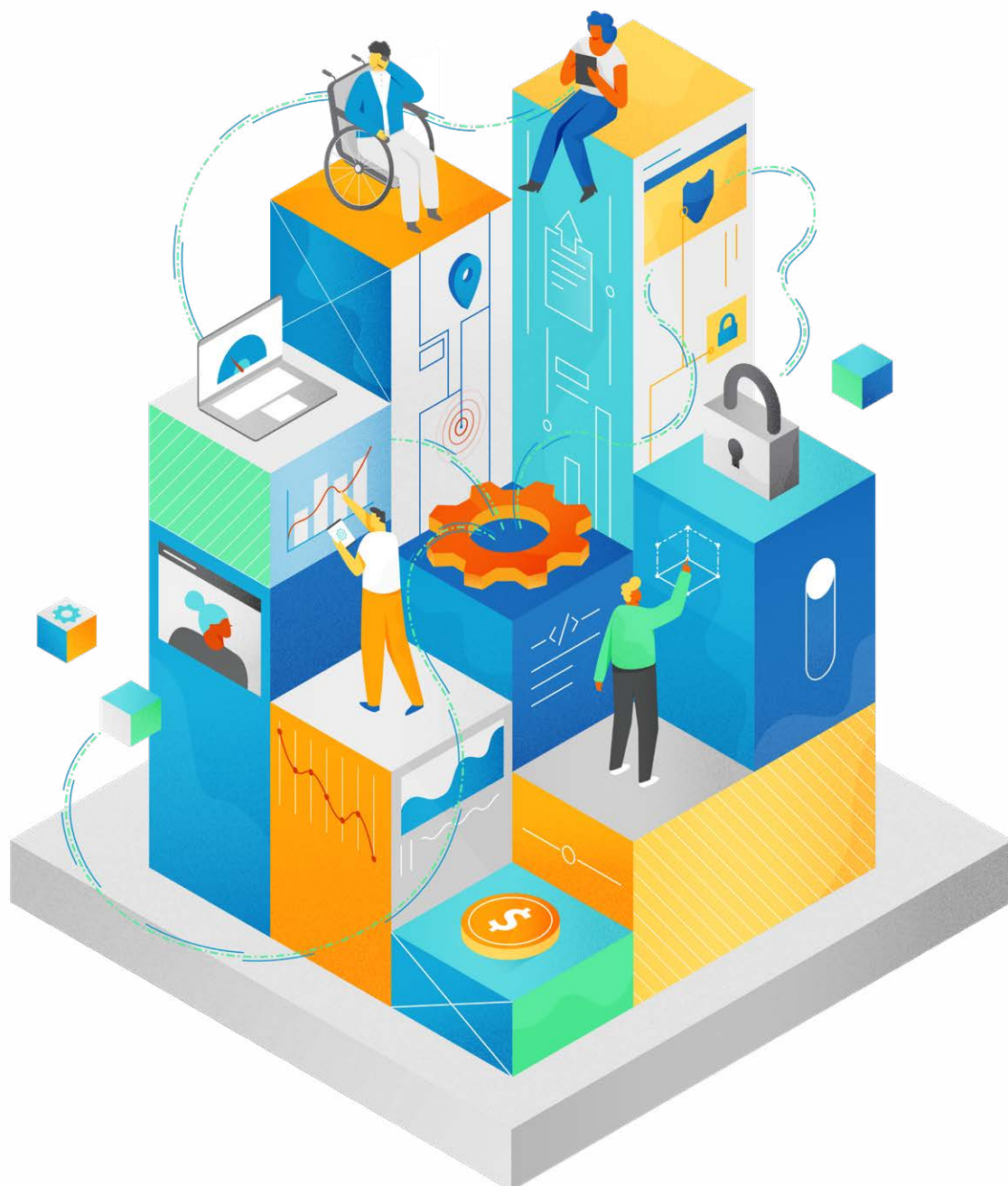# Mastering Full Lifecycle API Management with Analytics

Table of Contents

Introduction: Fueling Digital Transformation with APIs and Analytics

APIs enable software to talk to other software and let developers assemble disparate systems and applications into seamless experiences for an end user. As such, they play a foundational role in virtually every digital transformation effort, large or small.

Consider a modern connected experience, such as someone opening a mapping app on her phone, searching for nearby restaurants, finding one she likes, tapping into its profile, and booking a reservation. For this fluid and friction-free experience to occur, scores of APIs have to fire behind the scenes: APIs that look up restaurant locations, APIs that look up restaurant menus, APIs that look up reservation systems, APIs for customer data, and so on.

Likewise, when an enterprise launches a new mobile app for employees that draws from a legacy database or when an enterprise shares data or functionality with a partner, the connections are powered by APIs.

Given the vital role APIs play in digital transformation, many leading enterprises manage APIs as full lifecycle products, not as traditional IT projects that teams build and leave behind.

Full lifecycle API management is an intensely data-driven process. It entails not only technology solutions but also the data-driven processes and mindsets that those solutions enable.

Enterprises need visibility into whether developers are having good experiences working with APIs and whether the APIs can be improved, for example. They need insight into which APIs are being adopted and what that adoption may portend about emerging business opportunities or investment priorities. They need to understand how legitimate API traffic differs from traffic from bad actors—and how to thwart the latter without disrupting the former. They need data to align business and IT leaders around the ways developer activity contributes to revenue—and much more.

This ebook explores why strong analytics capabilities are crucial to full lifecycle API management and should permeate virtually every aspect of the management process.

Full Lifecycle API Management: More Than a Gateway and Data-Driven at Every Step

API gateways are often a starting point when enterprises begin investing in APIs and API management. To an extent, this focus makes sense: API gateways secure and mediate traffic between clients and backends, and between a company's APIs and the developers, customers, and partners who use them—so they are a crucial piece of the API management puzzle.

But enterprises are learning that it's important for API programs to consider more than gateways; analysts Kevin Matheny and Matt Braiser note in the Gartner report "How to Successfully Implement API Management,"[1] "API management requires an API gateway for endpoint protection, but a gateway alone is insufficient. Gateways do not offer analytics, monitoring, developer support or governance capabilities."

[1] Gartner How to Successfully Implement API Management, Kevin Matheny, Matt Brasier, 17 May 2019.

Full lifecycle API management isn't confined to gateway functions such as reliable processing of API calls or controlling who can use APIs. It also includes things like aligning various internal stakeholders, serving different API user constituencies, continually iterating and improving APIs, recognizing opportunities for innovation, and finding new markets and revenue channels.

As Google's Apigee team has explored in previous ebooks, fulfilling all of these mandates requires a lot of parts and pieces. API programs need a self-service developer portal where developers can discover, learn about, access, and test APIs. They require design, development, publishing, deployment, and versioning capabilities. They may need ways of monetizing their APIs.

Perhaps more than anything, they need advanced monitoring, metrics, and analytics to inform all of the aforementioned processes—capabilities that not only capture raw traffic data but also make that data actionable to help prevent abuse, provide insight into developers' experiences, shape product iteration, align internal stakeholders, and shine lights on untapped opportunities. The following API management use cases illustrate the variety of ways that sophisticated analytics may manifest in an API program's day-to-day operations.

⚙ Serving developers with an "outside-in" perspective

The "outside-in" perspective is a design and development philosophy that focuses on customer experiences, then flows toward the organization's inside view to shape how the enterprise defines products, prioritizes goals, and creates digital experiences. Because developers are an API program's primary user or customer, API team members should strive for an "outside-in" view that helps them understand whether their APIs provide a good experience and whether they can be improved. The API program will need metrics to answer key business and operational questions including:

- How are developers using the APIs?

- Where is the traffic coming from?

- When is API response time fastest or slowest?

⚙ Pursuing innovation and discovering business opportunities

Many APIs are produced with specific, defined business goals in mind—but APIs can also organically unlock unforeseen opportunities as developers begin to leverage them for new digital experiences. Navigational and mapping APIs weren't necessarily created with ridesharing in mind, for example—but these APIs helped enable this new industry nonetheless. When developers can mix and match data and functionality from different sources, new revenue opportunities and perhaps even new verticals can emerge. To make the right decisions in this landscape, enterprises need to be able to combine API traffic data with other data sources to analyze user behavior and consumption patterns, and to relate these patterns to business outcomes or opportunities. To bolster innovation and unlock business opportunities, an API program's analytics functionality should be able to answer questions such as the following:

- What are the most popular APIs?

- How do end users interact with the apps and digital experiences that the APIs power, and are any of these apps and digital experiences generating revenue?

- Will developers pay for API access? Is the API a candidate for API marketplaces?

- How should developers be categorized, and are there geographic, market segment or other patterns emerging organically?

⚙ **Achieving operational excellence**

Developers and the applications and connected customer experiences that developers build rely on APIs that are available, fast, and efficient. To meet this mandate, API teams need metrics and analytics that let them answer questions including:

- Are APIs available and how are they performing?

- What thresholds should be established to trigger alerts so that problems can be detected before they affect customers?

- When an availability issue arises, what needs to be done to fix it?

- When an availability issue arises, is it an isolated incident or a recurring issue, and is it related to a systemic failure or an attack by bad actors?

⚙ Preventing abuse

Dealing with hackers, denial-of-service attacks, and other online assaults is the cost of doing business in today's digital economies. If an enterprise is successful in the digital arena, its APIs are going to come under attack, sooner or later—so it's important to have the necessary insight to be prepared. Questions include:

- Have any developers attempted to violate an API's user policy?

- Are any behaviors suggestive of bots, SQL injections or other attacks? Are automated countermeasures configured such that they appropriately deploy when needed?

- Which incidents involve external threats and which involve employee error and other internal risks?

⚙ Flying Blind

While the preceding provides an overview of the benefits and insights that API analytics can provide, the unfortunate flip side is that the inability to answer the above questions can both limit an enterprise's strategic optionality and leave its data vulnerable:

- If an API program has no insight into how developers are using its APIs or if those developers are having good experiences, it will struggle to meaningfully improve its products, prioritize features, and achieve an "outside-in" perspective.

- If an enterprise can't recognize emerging opportunities or changes in user behavior, it can't agilely fine-tune its approach to capitalize on narrow opportunity windows and will likely struggle to align stakeholders, measure and express the API program's progress, and incentivize desired behaviors among program participants.

- Without analytics that both detect and help remediate uptime issues and other operational problems, the API program will lack the ability to meet service-level agreements and troubleshoot issues before they impact users.

- Without analytics that both detect suspicious behavior and automatically deploy the appropriate defense, the API program will not be equipped to reliably distinguish operational issues from security issues or to stop an attack before it impacts customers.

All of these use cases and risks speak to the importance of comprehensive visibility throughout the API lifecycle. Piecemeal approaches to achieving this visibility can be limiting. Third-party monitoring and analytics solutions bolted onto an API gateway may provide some data and periodic insights but often lack the fine-grained visibility natively built into leading API management platforms. Likewise, many tools lack the ability to define attributes for custom metrics or to automate actions (such as throttling traffic when a potential bad actor or rogue application is detected). Enterprises need the ability to act on API data—not just to detect a problem, but to quickly diagnose and remediate it before it affects customers.

API Analytics in Action: How Businesses are Making Use of API Data

Serving nearly two billion people, AccuWeather is the world's leading weather media and data company, and it has a decorated history of partnering with other enterprises to make weather information available across a wide range of services and devices. But as independent developers and app stores began to increase in relevance, AccuWeather leaders realized they needed to open up their partner programs to other players.

"A single developer always has the potential to be working on the next big thing, and become our next big enterprise partner," said AccuWeather senior technical account manager Mark Iannelli. "We needed a way to reach them."

The company's solution was to monetize their APIs, with different products tailored to different types of developer needs and to make the APIs available via a user-friendly developer portal that contains documentation, testing tools, and other resources that developers need to get up and running quickly. For example, AccuWeather offers APIs with near real time weather data, but it also offers options with more periodic updates for developers who require lighter data overhead.

Since then, tens of thousands of developers have registered accounts and signed up for API keys with the company, and AccuWeather has found that though a portal was crucial to get its efforts off the ground, its analytics capabilities, powered by Google Cloud's Apigee API management platform, have been key to keeping the momentum going.

For instance, analytics enable AccuWeather's API team to observe traffic patterns and understand how users view weather data over the course of a day. This helps AccuWeather improve the services it provides, making it easier for developers to quickly and efficiently deliver information to end users when and how they desire it.

Analytics "have helped us customize our API products to the needs of developers by revealing traffic patterns and making sure users get weather data when and how they want it to best achieve their desired outcomes," said AccuWeather CTO Chris Patti in a blog post. "Using these traffic patterns, we can see which developers are most active, which APIs are most heavily used, what time of day people look at the weather, what clients are growing fast, and which ones may need more support. This lets us be proactive to continue building useful products."

Like AccuWeather, data center service provider Equinix uses information it gleans from its API analytics to shape strategies. In an interview, Yun Freund, the company's vice president of engineering, said that this view into how Equinix APIs are being used improves Equinix's ability to build out the necessary capacity for customers that use its cloud services.

"With the insights coming from API usage, you can actually anticipate certain patterns from certain customers," Freund said. "You don't want to know about these things [only] when you hit a bottleneck and run out of memory."

Freund has the data generated via API analytics in hand during her quarterly business reviews, she added: "Last quarter, we had one company calling a million transactions on an API, which is a great trend. We get a clear picture of how customers are using our products through the APIs."

KPIs for APIs: Measurement Best Practices

Analytics are only useful if they align collaborators and improve decision-making and optionality. To achieve these outcomes, enterprises need to measure and analyze the right things. To position an API program for success, business and technology leaders should define metrics and quantifiable goals that are worth tracking and from which actionable insights can be derived.
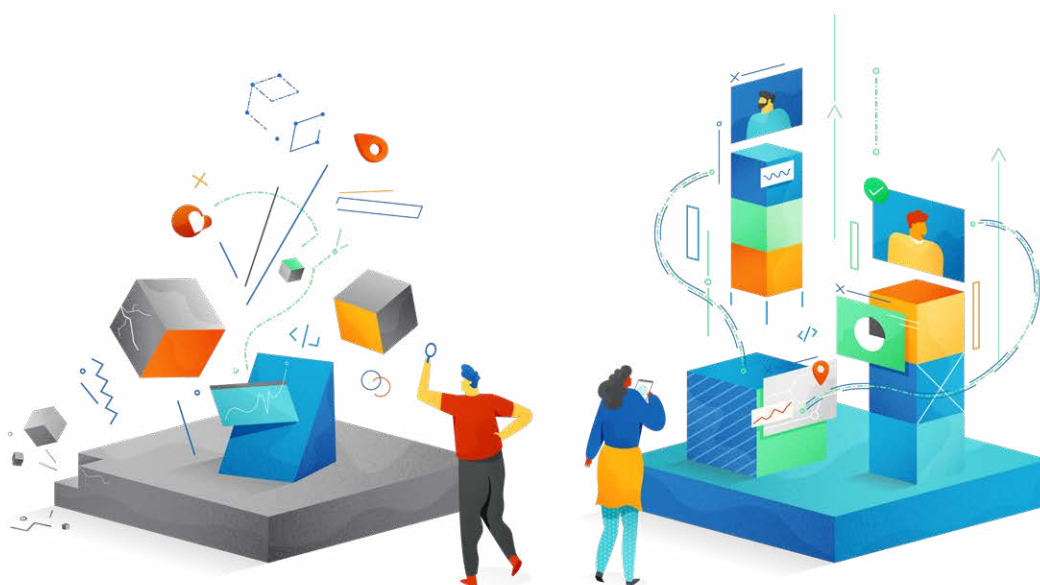
Good key performance indicators (KPIs) are a cornerstone of an effective API analytics effort, but they can be difficult to define. In addition, API-centric KPIs can be misleading in isolation. If such KPIs are not combined with business-level KPIs, the API program may gravitate toward simple IT-level metrics, such as the number of APIs produced, the number of developers using APIs, or the number of apps using APIs. Without any clear relationship to business outcomes, these measurements can lead a team to emphasize the wrong incentives and overlook compelling opportunities.

For example, if an API program creates as many APIs as possible, it may help the enterprise's leaders to feel like they're jumpstarting their progress, but a KPI oriented around API volume generally won't encourage API teams to focus on developer experiences or business value—just on passing existing web services through the API platform layer, typically without simplifying or optimizing them for consumption.
To avoid such pitfalls, it may be useful to follow these guidelines:

- Focus on driving growth, breadth, and speed of API adoption.

- Generally, avoid using "number of APIs produced" as a top-line target. Enterprises that focus too narrowly on this metric often produce APIs of low value, low quality, and low adoption.

- Accelerate the velocity of iterations in not only API and application development, but also the creation of user-facing digital experiences.

- Align the metrics of the API program with the metrics of developers downstream in the digital value chain, such as channel partners using the APIs, and seek to align these downstream metrics with goals generated upstream by business leaders.

⚙ Common KPIs used in API programs

Here are some effective KPIs used in many API programs, along with common pitfalls API teams should look out for. The value of a given KPI is highly contextual, and all of the following targets can be either valuable or deceiving, depending on how each is used.

- **Number of APIs:** As mentioned, this can be a misleading KPI that shifts the focus from developer experiences and business value to output. But if leaders are aware of its limitations, this target can also be useful for boosting short-term productivity. This metric can be used tactically, and teams should take pains to ensure API production targets do not compromise good design and governance.
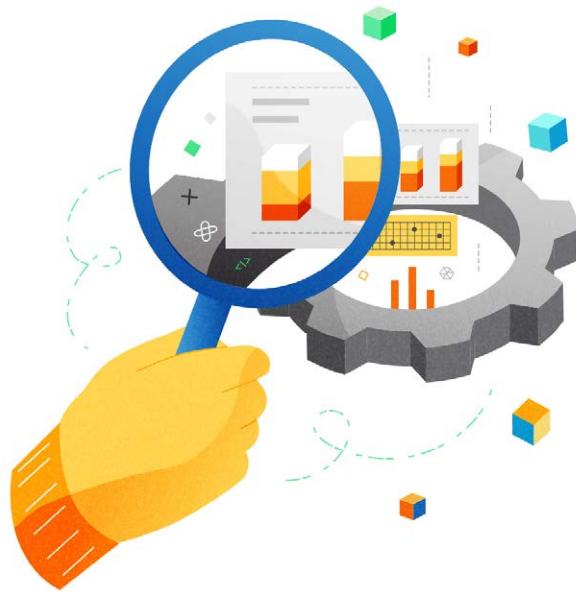
- **Number of developers:** This target is commonly intended to improve adoption. Even so, if an API program focuses singularly on marketing and onboarding metrics, it may produce APIs that are easy for developers to access but that provide unclear value. Enterprises should consider using this target in combination with other metrics that confirm a given API's business utility. API teams should distinguish between overall developer adoption and adoption among specific developers who are using APIs in a known business context, such as integrating the applications of existing ecosystem partners.

- **Number of partners:** This target can be used to accelerate partner outreach, drive adoption, and demonstrate success to existing business units.

- **Number of applications:** This target may be useful to drive reach, but if used in isolation, it can ignore that APIs must be relevant to the business. If an API program leads to the creation of lots of applications that are only used internally and not by customers, it can sometimes feed internal criticism and abandonment of the program.

- **Speed to API:** In order to enable application developers to rapidly create new customer experiences, API teams should consider becoming adept at balancing business goals and "outside-in" design considerations against the speed with which new APIs are launched. When this target also segments for APIs that are requested by the business, it becomes a useful measure of time-to-market for needed functionality.

- **Speed to onboard:** The portal that application developers use to access APIs should ideally feature an automated approval process, including self-service onboarding capabilities that let users register their apps, obtain keys, access dashboards, discover APIs, and so on. The ease and speed with which developers can navigate this process can significantly impact the adoption of an enterprise's API program. This initial automated signup should give access to low-risk APIs and sandbox environments that allow developers to be productive right away. Once developers are onboarded, the portal can provide upgrade options through which they can request access to more sensitive data and business functions. These upgrades often require increased diligence and background checks, some of which may take some time to clear. To distinguish between these processes, API programs can measure the "speed to onboard" for the initial signup duration separately from the "speed to upgrade."

- **Growth of traffic:** This target can help API programs develop a strong DevOps culture by continuously monitoring, improving, and driving value through APIs. Enterprises should consider coupling this target with related metrics up and down the value chain, including reliability and scalability of back-ends.

- **Breadth of business:** This target enables the API program to build relevance across the business, and to drive reuse of the APIs and thus of the back-end assets to which they connect. Business units accustomed to legacy integrations or old systems may resist adopting an API program, and by prioritizing this target, the program can more quickly escalate such pushback to the proper executive level for resolution. As the breadth of business related to APIs grows, API programs may find that particularly valuable APIs, such as those that provide access to unique data or proprietary functionality, may offer new revenue opportunities via monetization.

- **Cost reduction:** Significant cost reductions are often realized as reuse of APIs increases and needless duplication of existing but poorly-documented APIs decreases.

- **Direct revenue:** This target captures the revenue for the sales of core products that are enabled by APIs. With some exceptions, such as APIs that involve high bandwidth overhead or APIs that give developers access to a particularly valuable resource, API program leaders may want to be cautious about collecting an up-charge specifically for the use of the API; these charges may discourage the use of APIs, limiting the business's ability to open new markets and reach new customers.

apigee

Google Cloud

Ecosystem interactions, partner engagement, and other advanced measurement challenges

The preceding KPIs are typical of most API programs, whether they involve mostly an enterprise's internal developers, a limited set of partners, or vast networks of external developers. As partners and digital ecosystems become more important to an API program, however, the program may need to go beyond these typical measures, adopting more complicated metrics and KPIs to reflect more complicated business relationships.

In digital ecosystems, for example, one company may serve as an API provider to a partner that owns the relationship with the end customers. For the API provider, this dynamic can obscure the value the partnership produces. API teams may need to adopt a range of new approaches to address these sorts of challenges:
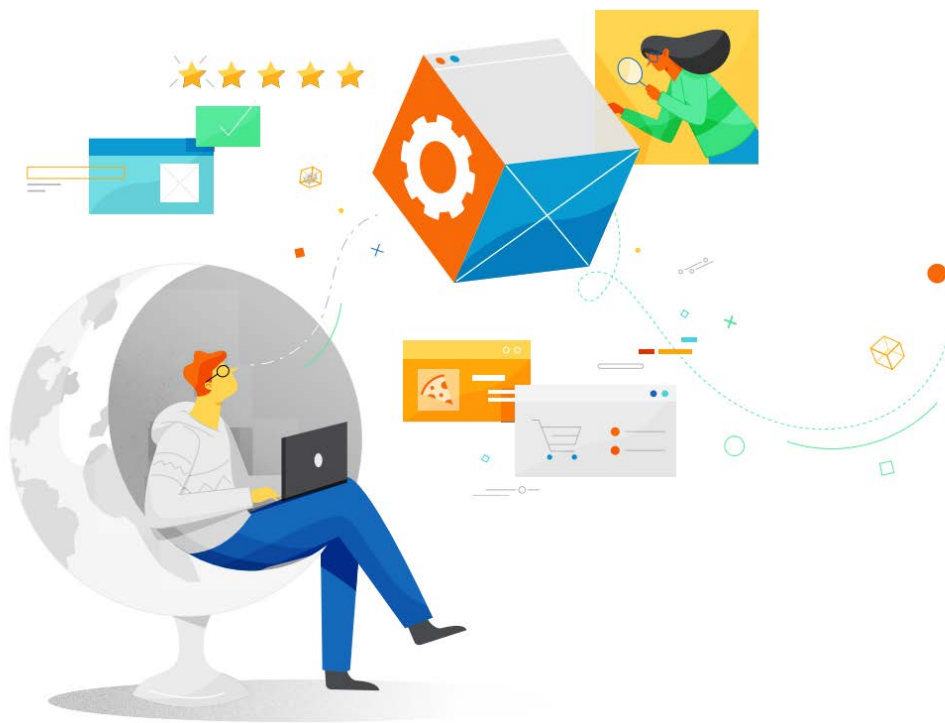
- If the relationship with the partner — or the nature of the API — allows tracking of specific end users, a business can correlate value created via the API directly to an end user. This can help enterprises match activities to specific user demographics.

- Different API calls have different value to both the API provider and its partner. A call to the provider's API to retrieve product information, for example, is somewhat less valuable than one that actually places an order for goods or services.

- Patterns in partner requests may reflect the value partners perceive from the API provider. For example, if an API opened to a partner produces many information requests but few orders, it may indicate the partner likes the provider's data but not its prices.

- Request patterns can also help an enterprise rationalize the cost of supporting a partner. High-touch partners that do not generate a correspondingly high level of value may not be worth the focus placed on them. A business may gain insight by comparing the value and costs of partners that use its API to those of partners that don't.

A business can derive additional insights and improve partner engagement by tracking partner interactions with its developer portal—a capability that reinforces why full lifecycle API management, and the metrics and analytics that fuel it, involves more than an API gateway. Questions that can be investigated via portal-based analytics include:

- How long does it take to onboard a new partner? Streamlining the sign-up and legal/business vetting process may accelerate onboarding and improve engagement.

- Are developers continuing to engage with the portal through documentation, community activities, blog posts or other channels? Ongoing engagement with API documentation, for example, may indicate a continued — or expanded — commitment.

- Is an API reducing partner interaction time and effort? If an organization's API provides good documentation, a platform for experimentation, and tiers of service that allow growth, the business may greatly reduce the manual effort it expends managing partners.

- Where are partners calling APIs from and how are they leveraging APIs for new products and digital experiences? Tracking the nature and location of interactions may provide insight into the expansion of a partner's business, which can in turn support the API provider's own planning process.

Checklists for Success

The preceding sections explore many of the ways enterprises need to be able to capture and leverage API and related data. Below are checklists to help enterprises evaluate whether their API management capabilities are up to the task.

- **Traffic analysis**

  Robust API monitoring and analytics capabilities should enable an enterprise to

  - ☐ analyze overall traffic across various geographies, the success rate of responses, the most common error codes, and transactions per second.

  - ☐ identify the source of traffic: determining which application is generating traffic; discovering what percentage of traffic is coming from a browser, a robot, a library, or other agents; confirming whether traffic originates from a PC or a mobile device; etc.

  - ☐ analyze traffic from not only production APIs, but also the API onboarding and access process that developers undergo via a developer portal.

  - ☐ measure the relative contribution of top APIs, applications, developers, and products to the overall health of the API program.

- **Performance analysis**

  To ensure APIs are supporting cohesive and compelling end user experiences, enterprises should invest in the ability to

  - ☐ measure API response time, target response time, and error count across geographies.

  - ☐ determine latencies of API proxies and targets.

  - ☐ analyze error codes and error composition across proxies and targets.

- **Developer engagement**

  To give developers a world-class experience and ensure that business and IT leaders understand how developer engagement is affecting their API programs and the bottom line, enterprises need the ability to

  - ☐ identify which application developers are generating the most API traffic.

  - ☐ measure how developers interact with the developer portal, such as whether documentation, sample code and other portal resources have helped to reduce the time developers spend troubleshooting and whether developers who test APIs via the portal end up adopting the APIs for products.

  - ☐ learn how developers are consuming APIs.

  - ☐ analyze which API products are most popular.

- **Availability and performance monitoring**

  To increase the likelihood that any API availability issues can be resolved before impacting customers, enterprises need the ability to

  - ☐ monitor across the entire value chain, with the ability to drill down to granular levels of detail.

  - ☐ both generate alerts when errors occur and reduce resolution times by precisely identifying the source of errors, whether in the developer application, proxy layer, or backend target.

- **Security health**

  To keep APIs—and the valuable data and functionality they expose—safe from bad actors, enterprises should prioritize tools to

  - ☐ review API configurations to ensure APIs adhere to the business's security policies and compliance requirements.

  - ☐ learn who is accessing and exporting sensitive data such as developer information, trace sessions, and API keys.

  - ☐ identify anomalies in traffic patterns, distinguish secure traffic versus potential threats, and analyze which applications and targets are affected by a given issue.

Charting a Path Forward with API Analytics

API monitoring, metrics, and analytics play a crucial role at every stage of full lifecycle API management. Understanding API traffic patterns is essential to detecting and blocking bad actors. Visibility into every API call is critical to maintaining uptime and to giving partners and customers a good experience. Relating API traffic and adoption metrics to business KPIs is an important part of aligning internal teams and understanding where APIs are or could be contributing value. And using KPIs to understand the needs of developers is a prerequisite to "outside-in" thinking and the effective product iterations and developer engagement efforts this thinking can promote.

Regardless of where an enterprise is in its API journey, comprehensive monitoring capabilities and robust analytics efforts are among the most surefire ways to accelerate momentum. For an enterprise unsure how to scale its API program or uncertain about which next steps to take, analytics may literally be the difference-maker, providing insights that illuminate previously hidden opportunities, remove ambiguity, drive consensus, and help the business grow.

About Apigee API Management

Google Cloud's Apigee API management platform delivers full lifecycle API management to help businesses unlock the value of data and securely deliver modern applications and digital experiences. 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.

Want to
learn more?

Optimizing API Programs with Monitoring and Analytics

As APIs become the de facto standard of building and connecting modern applications, the experience of several stakeholders is heavily dependent on their performance. Learn how to optimize your API programs with monitoring and analytics.

[Get the ebook]

AccuWeather: Delivering the Weather with API Management

AccuWeather has transformed their business to partner with other enterprises and developers across a wide range of services and devices through the power of APIs. Hear from AccuWeather on how analytics helped them drive this transformation forward.

[Watch the video]

Apigee Edge free trial

Explore Apigee Edge, a full lifecycle API management platform that helps you manage the entire API lifecycle from design through iteration and helps you control the complexity of microservices.

[Try it free]