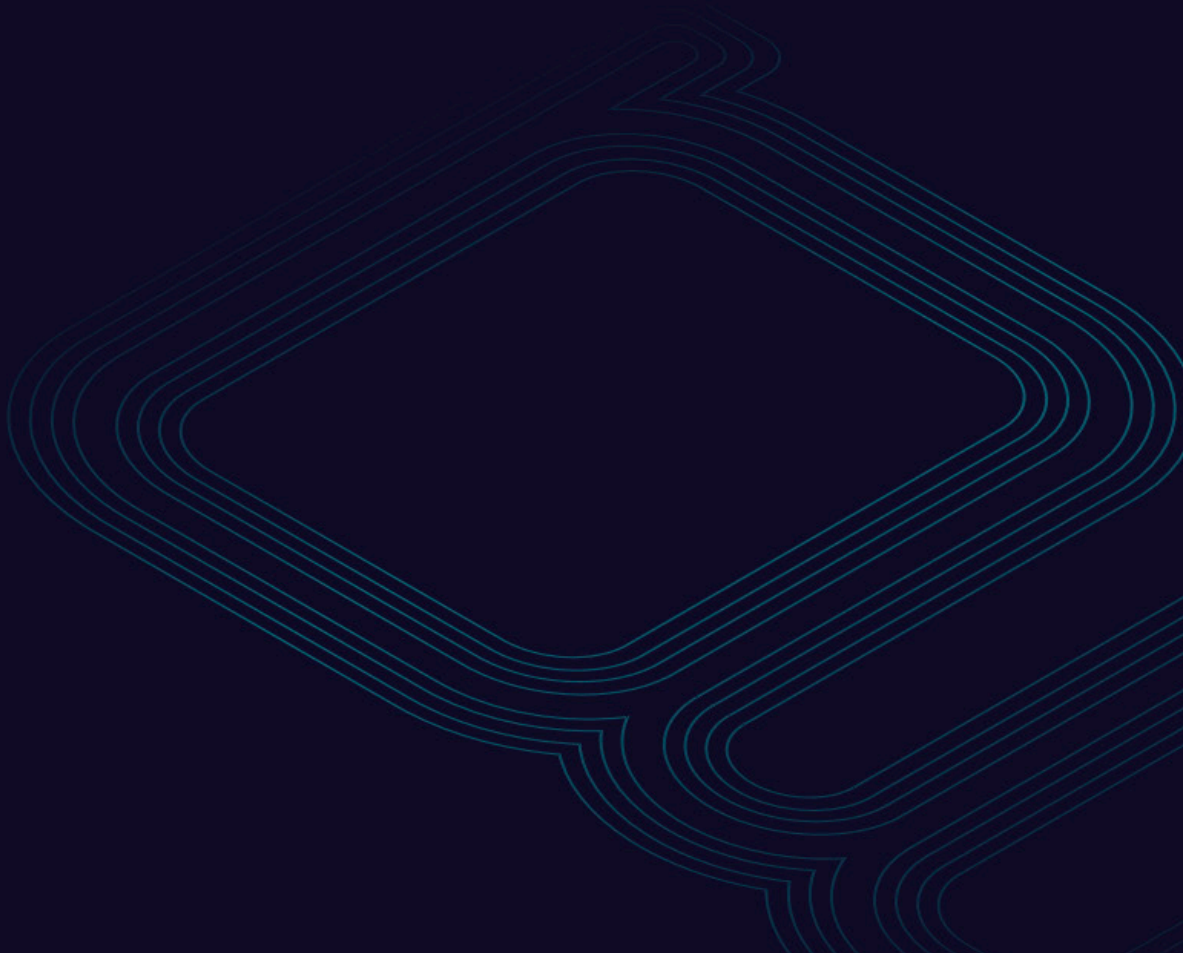
gravitee.io

The Future of API Management:
7 trends to watch



Introduction

As you may have noticed in our 11 API trends you should know white paper, we're pretty trendy here at Gravitee. So in the spirit of staying ahead, we're going to dive into some API Management-specific trends that we've noticed creep up over this past year (and a few that we expect to see more of in coming years). For example, we toot the horn of Event-native API management and talk a lot about event-driven APIs and architectures, but why? What's happening in the tech world that's making us want to talk about these things, and what could it mean for your API strategy moving forward?

Here's a quick TL;DR of the 7 trends we'll be exploring in more detail:

1. **Adoption of Event-native API management:** as event-driven architectures, streaming, and asynchronous communication grow in popularity, organizations will need support for synchronous and asynchronous APIs in one place. How will API management platforms accommodate for the shift?

2. **"Unification" of API Management and Event Management:** organizations will start to bring Event Management and API management under one functional roof, even if different tooling is used. How will events and event APIs be exposed to more consumers in a secure and reliable way? Will they be governed in the same way that synchronous, RESTful APIs are?

3. **Asynchronous API monetization as part of API management strategy:** API monetization is already a widely-known thing, but how will organizations' monetization strategies need to change moving forward given the rise of event APIs and asynchronous APIs?

4. **Rise of hybrid and multi-cloud environments:** managing APIs across multiple environments is challenging, so companies need a way to do it from one place. How will API management manage to provide consistent governance and security across them all?

5. **Improved (and automated) security and compliance:** a rise in APIs means a rise in potential exposure of sensitive data. Organizations will need sophisticated security and compliance measures; what are API management platforms doing to help?

6. **Integration with DevOps workflows:** more and more organizations are adopting DevOps teams and strategies which typically leads to the need for more integrations. API management needs to offer native integrations with DevOps tools to facilitate these teams.

7. **Growing importance of analytics and AI:** everyone's talking about AI these days, but what does that mean for API management? Likely it'll look like platforms being able to identify anomalies in API traffic and take corrective measurements automatically. There are many other examples that we'll get into later in this section.

Before we dive in – a quick note on the "why". Why bother with trends and/or guesses of what's going to be popular in an ever-evolving technological ecosystem? If your organization is one that has a need for an API management platform, choosing is difficult. You want a solution (and even a partner) that can change and grow with you as new business and technological requirements pop up. So keeping a finger on the pulse of rising trends can help to inform your needs and, by extension, your choices for API management investments.

Let's get going.

1. Adoption of Event-Native API Management

With the rise of event-driven architectures and growing demand for real-time, streaming data has come the need for API management platforms to support asynchronous APIs.

For starters, as organizations lean into event-driven architectures – which enables loose coupling between components and increased scalability – they're likely also leveraging asynchronous APIs. Same goes for any applications that require streaming data; they must utilize event APIs.

In order to support these efforts, API management platforms will need to not only be able to handle high volumes of events that are generated by the system, but it also needs to be able to "listen" for events, apply policies to event streams and messages, and support exposing event streams as various kinds of APIs (i.e. expose Kafka topics over REST, WebSocket, Webhook, etc.) In order to do this, the platform would need to natively support event broker and message queue technologies like Kafka, MQTT, and others.

At Gravitee, we call this Event-native API management. Similar to the more familiar term "cloud-native", which refers to a tool that was built for the cloud and to support cloud-based technologies, an event-native tool is built to support an event APIs, asynchronous APIs, and the like. So in order to fully reap the benefits of an event-driven architecture and real-time data streaming, organizations will find that they'll need an event-native API management solution.

2. Unification of API Management and Event Management

To take the above one step further... Not only will organizations find that they require a tool to support their event needs, but they'll also likely need their event-native API management and their event management under one functional roof. Even if they use different tools, they should all work together to allow for secure, reliable transmission of events between services, both north/south and east/west. Let's explore why.

Historically, the worlds of Event Brokers/Event Management and API Management have been operating as distinct silos in the business. This has not been all bad, as both practices have driven real value. API Management has driven value by making it easier to expose (typically synchronous, RESTful) APIs to various API consumers, while keeping access to APIs – and the data they broker – secure, reliable, and efficient. On the other hand, Event Broker and Event Management solutions have helped organizations make the move to event-driven architectures at high confidence – making critical information move faster, more efficiently, and in real-time across (typically backend) systems.

However, innovative organizations have realized that there is even more value to be uncovered by unifying the two practices. These organizations hope to expose both data-at-rest through RESTful APIs and data-in-motion through event-driven APIs. Sometimes there is a need to mix the two, taking the benefits of event brokers and exposing topics and events over more traditional REST APIs.

In fact, there are several opportunities to be had as a result of unifying these two previously distinct tools. Organizations would be able to deliver event APIs as differentiated products to customers; they could create new revenue streams by monetizing real-time data streams as Event APIs; they could make systems run more smoothly and reduce redundant development work by exposing crucial data and functionality to internal developers and stakeholders.

As with most opportunities, though, come some challenges. Making real-time data streams, event brokers, and message queue services easily consumable to a wide variety of different stakeholders is difficult given that not every stakeholder

can “talk” over an event broker’s native protocol. And even if they can, sometimes the APIs are difficult to discover and consume by developers and consumer applications. Plus, exposing real-time data streams come with inherent security risks: what if the wrong consumer gets the data?

It’s never simple to address challenges like these, but unifying your event-native API management platform with your event management would allow you to: mediate between different protocols, provide a centralized developer portal for exposing APIs to consumers, and apply security poli cies to streaming data.

Learn more about how Gravitee and Solace achieve this unification through their partnership here.

3. Asynchronous API monetization as part of API management strategy

API monetization has been a part of many organizations’ revenue strategy, but with this rise in real-time streaming (especially streaming to ad through the edge), will that strategy change? What unique opportunities to make money come with productizing and monetizing access to real-time data streams?

An API monetization strategy in the context of a request/response architecture typically involves charging for access to the API or for the amount of usage. But with asynchronous APIs, things can get a little more complicated (even though you’re still technically charging for usage).

On top of charging for usage, you could charge for access to specific events or streams; you could charge for the amount of data processed or by the number of events triggered; you could also charge a premium for real-time data and analytics and implement a delay on all other data that isn’t charged at a premium. Because the possibilities are more robust, monetizing and productizing real-time data streams comes with unique opportunities to create more revenue streams. Organizations can generate these revenue streams from third-party developers or from other organizations that want access to these data streams and events processed by their applications. Additionally, there are several other benefits of being able to monetize your asynchronous APIs:

- Better resource management: API monetization can help organizations better manage their resources by limiting access to certain types of data or events, or by charging for higher levels of usage.

- Increased adoption: By providing access to valuable real-time data and services through APIs, organizations can attract more developers and partners to their ecosystem. This can lead to increased adoption of their products and services, and potentially even new business opportunities.
- Better customer experience: APIs that enable integrations with other applications and services, organizations can improve the overall customer experience. For example, a travel booking platform could provide APIs that allow customers to book flights, hotels, and rental cars all in one place, rather than having to visit multiple websites. With event-driven APIs, you could show updated travel information and pricing in real-time.
- Competitive differentiation: By providing sophisticated event APIs, organizations can differentiate themselves from competitors. For example, a financial services company could provide (and monetize, of course) APIs that allow developers to build personalized investment portfolios based on a customer's risk tolerance and financial goals.

Where does the API management bit of this fit in? Successfully implementing an API monetization strategy for event APIs would be pretty hard to do without an API gateway.

A gateway allows you to implement certain crucial policies on your APIs, like authentication, access control, rate limiting, routing, filtering, and transformation – all of which would be crucial to allow for API monetization. Many tools also support usage-based billing, tiered pricing, and subscription-based models. An event-native API management platform, specifically, would also allow for management of the flow of events and data streams through applications to ensure that the right events are processed by the right components.

4. Rise of hybrid and multi-cloud environments

This is a popular topic lately, and we've covered it a bit in our latest multi-cloud white paper. So first of all – what is multi-cloud?

Multi-cloud environments refer to the use of multiple cloud computing platforms – such as AWS, Azure, and Google Cloud – to achieve better flexibility, redundancy, and cost optimization in an organization's IT infrastructure. In recent years, organizations have begun to implement multi-cloud environments to avoid vendor lock-in, leverage different cloud providers' strengths, achieve better cost optimization, enhance redundancy and disaster recovery, and improve application performance and availability. Multi-cloud environments can also help organizations meet compliance and regulatory requirements, as well as provide geographic diversity and better latency for users in different regions.

As organizations adopt hybrid and multi-cloud environments to support their operations, the need for an effective API management solution has become more critical than ever. An API management platform can help organizations manage their APIs and services across disparate environments, providing visibility and control over their API usage.

One of the biggest challenges that arises with hybrid and multi-cloud environments is that it's difficult to manage APIs that span across different environments and platforms. An API management platform can help organizations simplify this process by providing a centralized platform for managing their APIs – without the need for separate management tools for each environment.

An API management platform can also help organizations to ensure that different systems and applications are able to communicate with one another by providing tools to facilitate integration with different systems and applications, ensuring that their APIs are compatible with different platforms and environments.

5. Improved (and automated) security and compliance

With more APIs comes more risk to data – plain and simple.

To mitigate these risks, organizations need to adopt effective security and compliance measures. By providing a centralized platform for managing API security and compliance, API management gives organizations the ability to effectively improve their security strategy.

A few key features of API management for security include robust authentication and authorization to ensure that only authorized users and applications are able to access APIs and that APIs are only used for their intended purposes – an imperative piece to help prevent security breaches and compliance violations. A few other features that have become more and more popular with the rise of APIs include:

- Encryption: By encrypting API traffic, organizations can ensure that sensitive data is protected from unauthorized access and that data is transmitted securely across the network. This can help prevent data breaches and other security incidents, and can help organizations meet their compliance obligations.
- Threat protection: with threat protection, organizations can detect and prevent malicious attacks on their APIs, such as denial-of-service (DoS) attacks, SQL injection attacks, and cross-site scripting (XSS) attacks. By protecting against these threats, organizations can minimize the risk of security breaches and downtime, and can ensure that their APIs are available and responsive to users.
- Monitoring and auditing API usage: API monitoring and auditing helps organizations ensure that their APIs are compliant with regulatory frameworks like GDPR, HIPAA, and PCI-DSS.

We've already seen a massive shift in the attention that API security has received in the recent years as more and more organizations are relying on APIs to support their operations, both internally and externally. We expect that many API management vendors will begin to provide ways to automate security measures – maybe via AI; maybe via a management API; maybe both!

For more on API security, we've got a web page full of content here.

6. Integration with DevOps workflows

Ah, yes, another hot topic: DevOps. Until recently, DevOps and API management were rarely used in the same sentence. But with the rising onslaught of APIs in software development, DevOps teams are increasingly responsible for managing and securing APIs both internally and externally.

That means they're often taking over not only the implementation of API management, but in many cases even the vetting and purchasing of the platform too. As a result, API management platforms will need to start giving more attention to their integrations – DevOps teams will want to integrate whatever platform they choose with their existing workflows rather than having an entirely separate, siloed tool.

Besides integration capabilities, DevOps teams are also looking for:

- API gateway configuration: this provides a single point of entry for API requests, allowing a DevOps team to enforce things like security policies and rate limiting.
- API documentation: proper and consistent documentation allows developers to understand how to use the APIs and how to integrate them into applications.
- API testing: DevOps teams must test APIs for functionality and performance to ensure that they meet the organization's standards and requirements; most API management tools have this functionality.
- API monitoring: as we've already addressed, monitoring and alerting is a huge part of DevOps as it allows them to proactively address any problems.
- API security: with an APIM tool, DevOps can implement security measures such as authentication and authorization, encryption, and role-based access controls.
- API versioning: managing API versioning allows for changes and upgrades to the APIs without disrupting existing integrations.
- API analytics: as DevOps teams work to continuously improve processes and tools, they can use an APIM tool to collect and analyze data on API usage in

order to work toward further optimizing their APIs and API strategy.

To learn more about DevOps + API management, check out our white paper on the topic here.

6. Growing importance of Analytics and AI

Everyone's talking about AI these days, but what does that mean for the future of API management? Here's a few capabilities that we could look forward to in coming months/years:

- Intelligent API discovery: With the increasing number of APIs available, it can be challenging for developers to find the right API for their needs. AI could be used to analyze the requirements of a given project and suggest APIs that would be a good fit. This could be done by analyzing the metadata and analytics associated with APIs or by looking at the data flowing through APIs to understand their functionality.
- Smart API monitoring: APIs are critical components of many applications, and downtime or performance issues can have a significant impact on users. AI could be used to monitor APIs in real-time, identifying issues and providing recommendations for remediation. This could include analyzing logs, performance metrics, and user feedback to identify trends and potential problems.
- Automated API testing: Testing APIs can be time-consuming and difficult to do comprehensively. AI could be used to automate the testing of APIs, generating test cases based on the API's functionality and expected behavior. This could include simulating various types of user input and monitoring the API's response to ensure that it meets expectations.
- Intelligent API security: APIs can be vulnerable to security threats, and keeping them secure can be a complex task. AI could be used to analyze API traffic and identify potential security issues, such as suspicious user behavior or unauthorized access attempts. This could be done using machine learning algorithms that can learn from patterns in API traffic and detect anomalies.

- Predictive API scaling: As usage of an API increases, it may become necessary to scale it up to handle the increased load. AI could be used to analyze usage patterns and predict when scaling will be necessary. This could include analyzing historical usage data, monitoring current usage trends, and predicting future demand based on external factors (such as upcoming events or seasonal changes).
- ChatGPT integrations: As ChatGPT and other similar tools continue to become more robust and useful, it's fun to consider the idea of integrating such a tool within your API management platform/strategy. You could input a plain language command to build an API with XYZ requirements, and it could be done in seconds. Just a thought.

Wrapping up

So there you have it. There's some really exciting stuff happening in the world of APIs – there's also a whole new world of challenges. In order to address those challenges to fully take advantage of the ever-growing potential of APIs, it'll be imperative not only for API management platforms to keep up, but also for you as a user to stay apprised of new releases, requirements, and capabilities.

If you want to learn more about what API vendors are available out there and whether they meet your needs (and can keep up with the trends), check out our Event-Native API management Buyer's Guide.