KPMG

Powering the future

Keys to a successful API management strategy

kpmg.com/us

Contents

1 Introduction

Application programming interfaces (APIs) and microservices have become integral components of modern architecture and software development. APIs allow different internal and external software systems to communicate and exchange data. Microservices is a software architecture pattern that decomposes large monolithic applications into smaller, independent, and loosely coupled services. Microservices communicates with each other via APIs, with scalability, elasticity, and reliability.

The global integration software market revenue is expected to reach US$15.9 billion by 2026, with an annual growth rate of 15.8 percent.[1] A recent enterprise API adoption survey showed that 98 percent of organizations use or are planning to use internal APIs,[2] up from 88 percent in 2019, while 90 percent of organizations use or are planning to use private APIs provided by partners, up from 68 percent in 2019. The high demand for cloud-native applications to support scalability and resiliency drives the increase. There is continuous growth in the use of APIs to support technologies such as Internet of Things (IoT), blockchain, intelligent automation, and artificial intelligence.

Managing APIs and microservices is a complex undertaking. With the constant change in business needs required to respond quickly to evolving digital markets and on-demand customer needs, the task becomes even more challenging. That is where API management comes in to help the process of creating, publishing, and monitoring APIs in a manner that ensures they meet the needs of developers and end users. A good API management structure can be likened to an experienced Formula 1 pit stop crew.

In an environment where a split-second decision can determine the final placement of a racing team, more than 20 team members work together to get a race car back on the track in under 2.5 seconds. As impressive as that may seem, the team relies on a defined and synergetic approach to ensure team goals and stakeholder needs are met or exceeded where possible.

This point of view takes a closer look into what API management involves, the value of strong API management, challenges to a successful approach, and leading practices for implementing an API management strategy. In the next point of view, a similar lens will be applied to microservices.

1. Source: KPMG US Market Intelligence 2022
2. Source: Gartner's 2022 March Hot Topics Survey

2 What does API management involve?

API management involves much more than securing APIs. It includes activities such as designing, publishing, documenting, testing, securing, monitoring, and monetizing APIs. API management is an important part of any API strategy. By using API management, organizations can improve the security, performance, scalability, and developer experience of their APIs.

API management architecture

What does API management involve? (continued)

API consumers

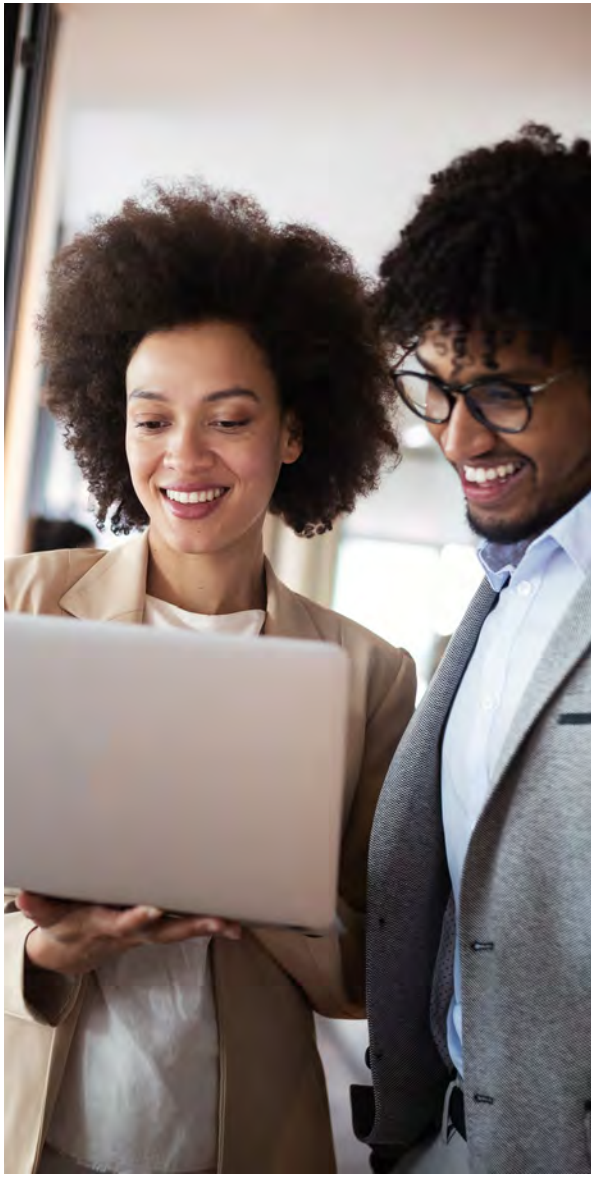
Systems, software, devices, websites, and developers interacting with the developer portal are all considered consumers of the API that performs various tasks, such as accessing data from a remote server, executing business logic, integrating with other applications, automating tasks, and providing a user interface.

API management components

API gateway – A unified entryway to connect services and systems within the API architecture. Responsible for API traffic routing, translating protocols between internal and external API clients and services. An API gateway is a critical component in a microservices architecture, as it helps to simplify API management, improve security and reliability, and promote scalability and strong application performance.

API manager – A tool enabling administrators to centralize control over the creation, distribution, and monitoring of APIs across different teams, departments, and partner organizations. API managers typically offer a range of features, including API design and documentation, security policies such as authentication, authorization, and API analytics. With an API manager, administrators can ensure consistency in APIs and governance practices, monitor APIs to optimize their use and quality, and enable feedback and collaboration with developers to improve APIs. API managers can help organizations to streamline the API management process and promote cross-team and third-party integration. This is done through usage tracking, subscription management, etc.

Developer portal – Serves as a self-service hub for developers to create, access, and share documentation. It also enables streamlined communication of enterprise standards across development teams. It is a self-service platform that enables developers to easily discover and learn about consumer APIs through features such as API documentation, tutorials, code samples, testing environments, API keys, forums, and support. API developer portals act as a bridge between API providers and API consumers, making it easier for developers to build applications that utilize the organization's APIs. A well-designed developer portal can empower developers to build quickly and consistently.



What does API management involve? (continued)

Logging, reporting, and analytics – Enables organizations to better understand how their APIs are being used and how to improve their performance.

Logging involves recording every request that an API receives along with information about the source of the request, the destination, and other relevant information. This information can help organizations monitor API usage, identify errors, and detect potentially malicious or fraudulent requests.

Reporting capability enables logged data to be automatically aggregated, visualized, and analyzed in a structured way. API-related metrics can be packaged into reports and shared with several stakeholders.

Analytics involve using data collected through logging and reporting to gain insights into API usage trends, understand customer behavior, and inform business decisions. Analytics can help organizations to identify which APIs are more popular and understand how they are being used, enabling them to make data-driven decisions about API development, deployment, and optimization.

API providers/back ends

Internal and external/third-party systems, databases, and software exposing data to be used by API consumers are the services that provide the logic, data storage, and computing resources needed to fulfill API requests made by API consumers (such as mobile apps, web applications, or other systems). API providers can be developed in a wide range of programming languages, and they can be deployed on various platforms, such as public or private clouds, data centers, or hybrid environments. In addition to processing API requests, API providers handle tasks such as data processing, storage, and access control, and they can have mechanisms for monitoring, logging, or scaling their computing resources.

3 API management challenges

Key challenges facing API management include:

People

Lack of skills and training: API management is a specialized skill set, and it can be challenging for organizations to find people with the right skills and experience. Furthermore, given the fast-paced nature of the industry, it can be difficult to maintain a deep level of expertise, especially if the organization does not provide adequate ongoing training or learning opportunities.

Collaboration and alignment: APIs need to be developed and managed collaboratively across different business functions and teams. This requires effective communication, alignment on goals, and adherence to standards and governance practices. However, conflicting priorities and siloed operations can make it difficult to achieve this kind of collaboration and alignment, leading to slow adoption, inefficiencies, and lower return on investment.

Cultural change and buy-in: Implementing an API management strategy often requires significant cultural change within an organization. This means getting buy-in from senior management, as well as different stakeholders across the organization. Without a shared vision and appreciation of the value of API management, it can be difficult to drive adoption and momentum.

API management challenges (continued)

Process

Governance and standards: One of the main challenges related to processes is establishing governance and standards around API development, management, and use. This includes developing procedures for API documentation, design, version control, testing, and deployment, among others.

Risk management: Many teams embarking on the API transformation journey often apply risk management approaches as an afterthought, usually only after an incident has occurred. Risks to the success of the overall program may not be identified, assessed, and mitigated continuously.

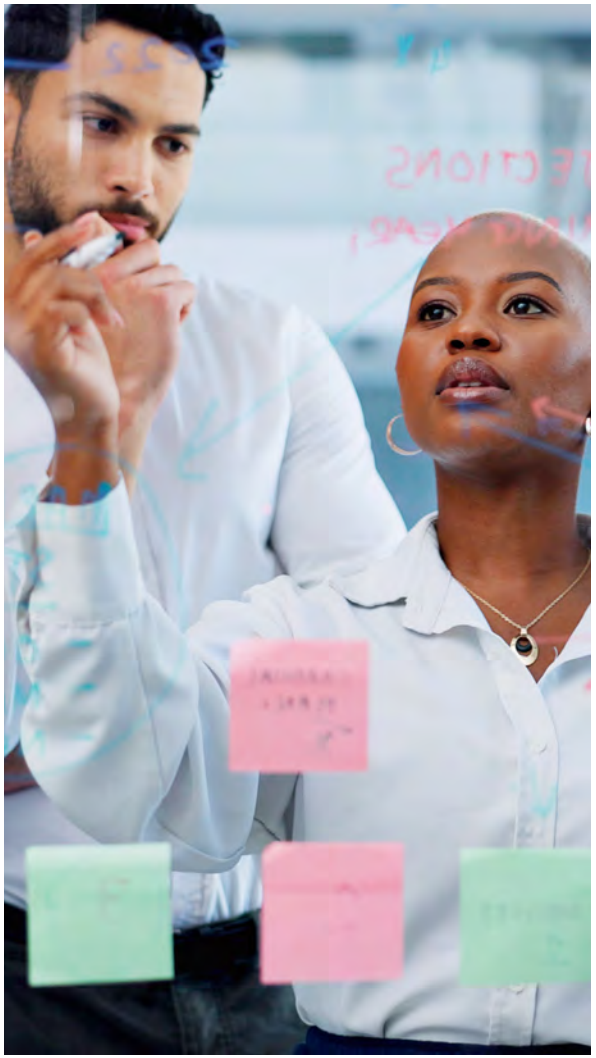
Versioning: APIs evolve and change over time, and this can lead to version control challenges. Organizations need to establish standards around versioning to ensure backward compatibility, manage the lifecycle of different versions, and provide clear communication to API consumers when changes are made.

Scaling and performance: As APIs become well adopted, traffic can increase rapidly, leading to performance issues and the potential to impact other services. Organizations need to establish processes to monitor and scale their API infrastructure as needed to ensure they can perform consistently under varying loads.

Adoption and change management: Another challenge with API management processes is driving adoption and change management within organizations. This means ensuring that teams understand the value of APIs and are willing to use them effectively in their daily work, as well as supporting training and education around their usage.

Third-party integration: API management often requires integrating disparate systems both within and outside the organization. Effective processes for onboarding and managing third-party integration can be challenging, especially when dealing with different technical standards, compliance requirements, and legal responsibilities.

Monitoring and analytics: Continuous monitoring and analysis of API usage are critical to ensuring that APIs continue to perform well, remain secure, and meet business objectives. This requires having robust processes in place to collect data, analyze it, and use the insights gained to inform continuous improvement.



API management challenges (continued)

Technology

Complexities of integration: Integration technologies and standards can be complex and require significant expertise. Organizations may need to invest time and resources in developing their integration capabilities or in purchasing and adopting appropriate API management software.

API security: APIs can present significant security risks, such as data leakage, denial-of-service attacks, or unauthorized access. To address these risks, organizations must implement robust security measures, such as encryption, firewalling, and authentication, and may need to understand the unique security requirements of different systems.

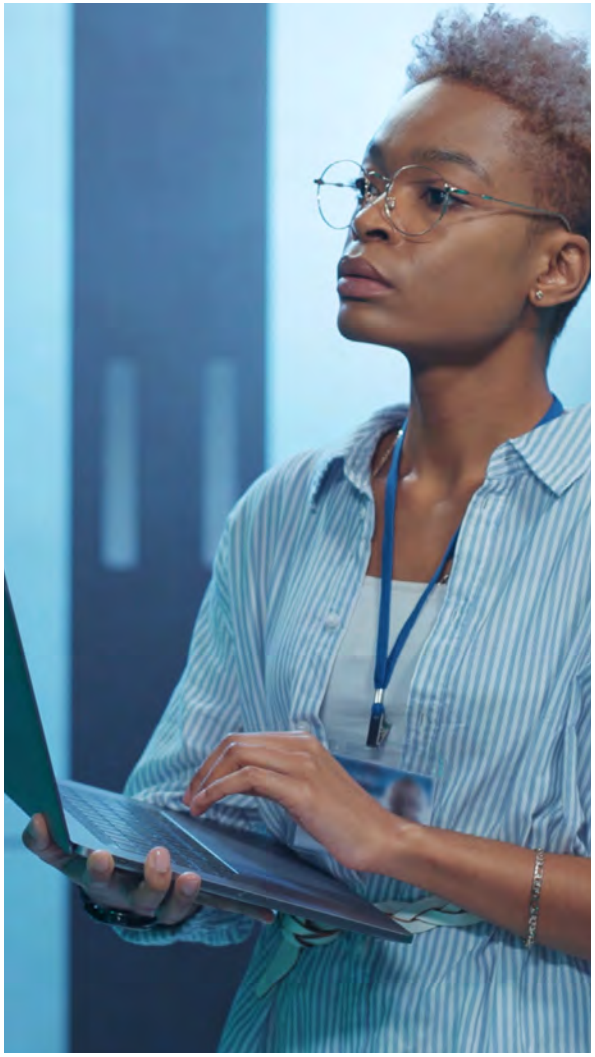
Scalability: Scaling APIs to accommodate large numbers of users, increased traffic, and new features can be challenging. Organizations need robust infrastructure, enabling scaling from the lowest possible volume to handling many requests.

Interoperability and data standards: APIs can have different data formats, protocols, and standards, which can create compatibility and consistency issues when integrating systems or exchanging data. Organizations need to standardize data formats or have translation tables to promote interoperability and adhere to industry standards when available.

Technical debt: APIs can become complex and highly customized over time, leading to technical debt if the appropriate time and investment are not made to keep them up to date or appropriately documented.

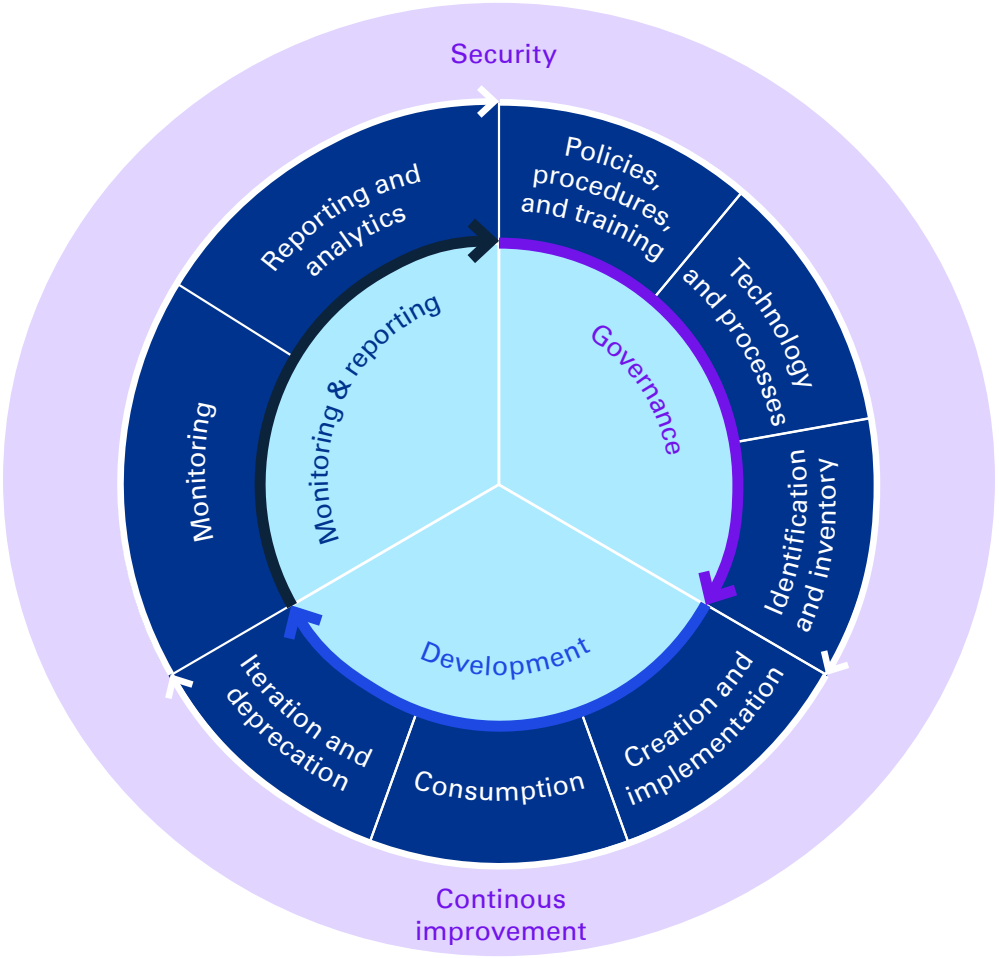
Legacy systems: Finally, many organizations operate in hybrid, fragmented environments with a mix of legacy and modern technology. This can make it challenging to integrate new APIs with existing systems and can slow down innovation and improvement.

Compliance: APIs, like any other technology, need to be managed with compliance in mind. This means ensuring that APIs and relevant processes are compliant with relevant regulations and legal requirements, such as General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act (HIPAA), or other industry-specific standards.



4 API management framework

Organizations are required to take a comprehensive approach to get the best out of their API management strategies. The entire lifecycle of the API environment needs to be broken down and individually addressed to enable organizations to position themselves to meet their strategic goals.



API management framework (continued)

Governance

Governance policies, procedures, and training

API policies, procedures, and training are essential for any organization that wants to establish a secure, reliable, and efficient API management program. To create effective policies and procedures, businesses should begin by identifying their API management goals and risks. Policies should be tailored to address these goals and risks while ensuring compliance with regulatory requirements. Procedures should be designed to provide a clear and consistent approach to managing APIs across the organization, including steps for creating, testing, documenting, and managing APIs. Training is critical, as it ensures that all staff involved in developing, managing, and using APIs understand the policies and procedures governing their use. By implementing the above practices, for API policies, procedures, and training, businesses can help ensure that their APIs are developed, deployed, and managed securely and effectively. This can improve the reliability and performance of APIs, which benefits the organization, its partners, and customers.

Technology and processes

Selecting the right technology and processes for APIs is critical for ensuring the success of an API management program. Choosing the right technology can help increase reliability, scalability, and performance, while selecting the right processes can help reduce development time, increase efficiency, and improve overall security. When selecting technologies, consider factors such as usability, security, flexibility, and speed. Ensure that the technology integrates well with existing infrastructure and meets specific business requirements. With processes, it is important to focus on leading practices like agile development, continuous integration, and continuous delivery to allow for rapid iteration and testing of APIs. By selecting the right technology and processes, organizations can create a robust API management program that is more likely to meet their business objectives while also providing a seamless integration experience for developers and end users alike. The steps above should be considered when selecting and aligning with external partners (vendors and third parties).

Identification and inventory

When managing APIs, it is crucial to have an accurate identification and inventory process in place. Without a clear understanding of which APIs are in use, it can be difficult to ensure their security and maintenance. To begin, it is essential to define a clear naming convention that will allow for easy tracking and identification. This naming and versioning convention should be documented and consistently applied across all APIs. Additionally, it is important to use API management tools to automate the identification and inventory process where possible. There are several tools on the market that can be used to create and manage documentation as well as allow for the automatic discovery of APIs. Regularly reviewing and updating the API inventory will ensure that nothing is missed and that all APIs are up to date and secure. By implementing these leading practices, organizations can streamline their API management and keep their systems secure and efficient.

API management framework (continued)

Development

Creation and implementation

Creating and implementing APIs can be a complex process that requires careful planning and attention to detail.

- First, it is crucial to have a clear understanding of why the API is needed and how it will be used.
- It is also essential to design the API documentation clearly and provide sufficient examples and sample code.
- Proper testing should be conducted to ensure that the API is functioning as intended and can handle various use cases.
- Implementation should be done gradually to allow for adequate monitoring and feedback from users. Additionally,
- APIs should be designed with security considerations in mind, such as enabling authentication and encryption.
- Regular maintenance and updates are necessary to keep the API functioning properly and to address any security concerns.

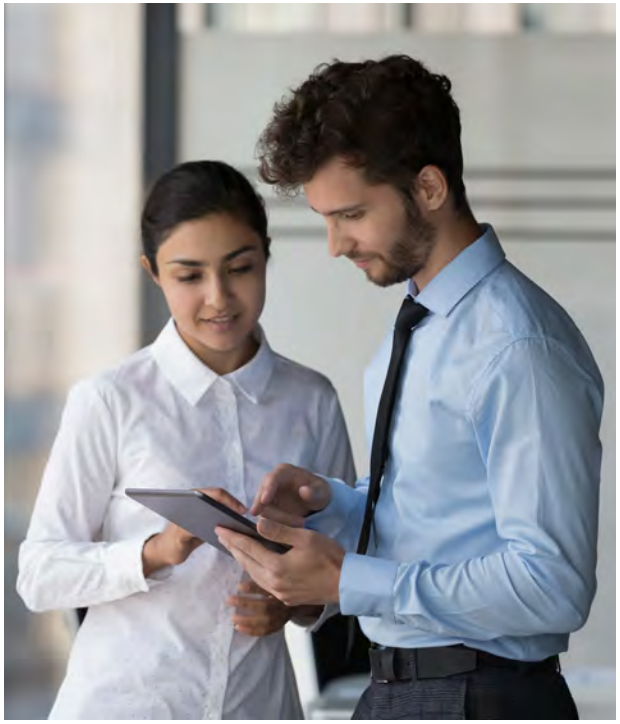
By following these leading practices for API creation and implementation, organizations can ensure that their APIs are useful, reliable, and secure.

Consumption

There are several leading practice that may help in effectively consuming APIs. One key practice is to always use the latest version of the API and keep up to date with any changes or updates. Additionally, understanding API terms of use and guidelines is crucial to avoid using the API in unintended or inappropriate ways. It is also essential to build error handling and resiliency when working with internal and external services to handle issues that arise. Understanding the limitations and capabilities of an API is important when designing an application, as this can impact the proper use and performance of the API. By following these leading practices for API consumption, developers can ensure that their applications are effective, reliable, and secure.

Iteration and deprecation

As new functionalities are added or older functionalities become obsolete over time, APIs need to be iterated and/or deprecated. A documented versioning strategy should be established to allow API consumers to understand how each version is supported and when older versions will be deprecated. Proper communication channels, such as a mailing list or RSS feed, should be established to notify API consumers of changes and upcoming deprecation. It is also important to provide adequate migration guides and documentation to API consumers to help them transition to the latest version. The older versions should be supported for a period to allow for a smooth transition. Proper monitoring and logging should be performed to identify consumers who are still using the deprecated APIs, and need to be transitioned to the latest version. By following these leading practices for API iteration and deprecation, organizations can ensure a smooth transition for API consumers, and minimize the impact on existing applications.



API management framework (continued)

Monitoring & reporting

Monitoring, reporting, and analytics

API monitoring is essential for maintaining the performance, availability, and reliability of any application. The first step toward achieving this is to define appropriate metrics and set up alerts for deviations from acceptable norms. This could include response times, error rates, and throughput, among others. It is also important to monitor API dependencies to ensure that any changes or failures are detected and addressed promptly. Another leading practice is to use automated testing to simulate user behavior and identify potential bottlenecks or performance issues before they impact end-users. Finally, implementing a centralized monitoring and reporting strategy that consolidates data from different monitoring tools can provide actionable insights and help identify trends that can lead to proactive maintenance of APIs. Following these leading practices can help improve the overall performance of applications and provide a better experience for end users.

Security

API security is crucial to protect application and sensitive data from malicious attacks. Leading practices for API security include using authentication mechanisms such as OAuth2, JWT, or API keys to control access to APIs, and ensuring that data is transmitted over a secure HTTPs/SSL protocol. It is also important to implement rate limiting to prevent denial-of-service attacks and to monitor API traffic for suspicious activity. Regular updates to API security protocols are necessary to identify and address vulnerabilities. Additionally, encrypting sensitive data at rest and in motion can further enhance security measures. It is also essential to have proper access control to APIs and to ensure that sensitive data is not exposed through debug endpoints. Following these leading practices for API security can help organizations keep data and users safe from threats, reduce the risk of attacks, and maintain the trust of customers.

Continuous improvement

APIs are never done. Continuously improving an API management strategy involves an iterative process of refining, optimizing, and adapting your approach to better meet the needs of developers, stakeholders, and the evolving technological landscape. To achieve continuous improvement in your API management strategy, organizations need to set clear objectives, regularly assess and evaluate the strategy, gather feedback, and keep up to date with technology upgrades, security, and regulatory requirements.

Setting clear expectations requires organizations to define specific goals and objectives. These objectives could include improving developer experience, increasing API adoption, enhancing security, or streamlining processes.

Regularly assess the current state of the API management strategy. Evaluate key performance indicators (KPIs) such as API usage, response times, error rates, and developer feedback.

Establish a feedback loop to close the loop with developers and stakeholders. Share updates on implemented improvements and gather feedback on the effectiveness of those changes. Keep track of evolving technologies and industry leading practices.

Regularly assess whether the technology stack is up to date and aligned with the latest trends. Upgrading components can lead to performance gains, enhanced security, and improved developer experience.

Continuously monitor and update security measures to address emerging threats. Implement security leading practices, perform regular security audits, and stay compliant with relevant regulations (e.g., GDPR, HIPAA).

5 Potential benefits of a well-executed API management program

The following highlights some of the potential benefits of a well-executed API management program.

Governance and security

API management requires establishment of baselines and/or standards to ensure compliance with corporate policies and external regulations. Governance rules around data, access, integrations, and security can be easily enforced through API management solutions.

Documentation and API reusability

This enables development of new APIs or changes to existing APIs to adhere to standars and ensure consistency. This allows developers to use parts of the existing API code or decrease repetitive APIs.

Agility

API management program enables rapid creation, updating, sharing, and monitoring of APIs without unnecessary costs or undue loss of productivity. This enables organizations to be nimble while addressing strategic demands.

Data-driven approach

API analytics provides insights to developers and organizations on API usage, performance, and security threats through real-time monitoring and dashboards. Resource allocations can be optimized to address issues while enabling opportunities to be harnessed.

Automation

Noncritical business and repetitive tasks such as onboarding and incident management can be streamlined and automated because of a centralized and integrated API management solution. Developers and business teams can spend more time on critical matters requiring attention.

Increased scalability

API management can help organizations to improve the performance and scalability of their APIs. This can be achieved by caching responses to frequently requested API calls, load balancing traffic across multiple servers, versioning APIs and allowing organizations to deploy new versions of APIs without impacting existing users, rate limiting API calls to prevent a single user or application from overloading an API, and monitoring to identify and address scalability issues before they impact users.

6 Getting started

Getting started with API management can seem overwhelming, but there are a few steps organizations can take to begin the process.

Step 1: Establish business goals and objectives for using APIs. This can include defining the use cases for APIs and identifying the internal and external stakeholders who will use them. The organization should also determine the potential benefits of API usage, such as improving customer experience, reducing costs, or increasing revenue. This information can help them prioritize and plan their API management approach. It is also important to create a clear

API governance framework, which includes defining API standards, security policies, and performance metrics. A dedicated API management team or leader should be appointed to own and oversee the API management program.

Step 2: Organizations should identify the technology and infrastructure requirements and establish an API development and deployment process. They should use API design leading practices for long-term scalability and reusability of their APIs. An API management platform can support the API management process.

Step 3: Organizations should establish a continuous feedback loop, soliciting input from both API consumers and developers to ensure that APIs are meeting business objectives and user needs, and continue to make incremental improvements to the program.

By following these steps, organizations can get started with API management and begin realizing the benefits of a well-managed and successful API program.

As discussed earlier, APIs are essential in microservices architecture as they facilitate communication, loose coupling, language independence, scalability, flexibility, encapsulation, and security between the individual services. They act as the glue that enables the composition and coordination of microservices into a cohesive and functional application. In our next publication, we will be taking a deeper dive into how to get the best out of your microservices deployment.

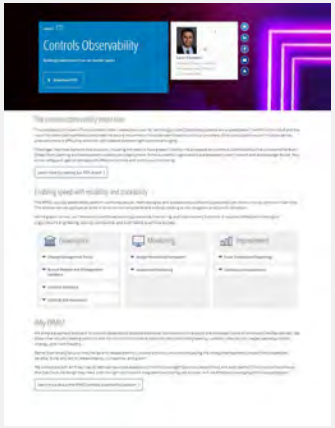
KPMG Technology Risk Centers of Excellence

Learn more at:
visit.kpmg.us/TRMCOE

7 Read more technology risk insights

Building trust in cloud environments

The lowdown on low code



The controls observability imperative

Learn more by visiting our Technology Risk Modernization Centers of Excellence webpage.

Contact us

Kevin Coleman
Partner, Technology Risk
T: 415-963-7209
E: kmcoleman@kpmg.com

Lavin Chainani
Managing Director, Technology Risk
T: 410-949-8834
E: lchainani@kpmg.com

Nana Amonoo-Neizer
Director, Technology Risk
T: 402-999-1816
E: namonooneizer@kpmg.com

Werner Vanzyl
Director, Lighthouse
T: 719-493 0948
E: wvanzyl@kpmg.com

Some or all of the services described herein may not be permissible for KPMG audit clients and their affiliates or related entities.

kpmg.com/socialmedia