Top 10 Things Software Engineering Leaders Need to Know About APIs

Shameen Pillai, Senior Director Analyst
Mark O’Neill, VP Analyst

27 May 2021
Top 10 Things Software Engineering Leaders Need to Know About APIs

Published 27 May 2021 - ID G00746066 - 16 min read

By Analysts Shameen Pillai, Mark O'Neill

Initiatives: Software Engineering Technologies

APIs have become highly integral parts of mission-critical business capabilities, differentiating customer experiences and software engineering architectures. It is critical for software engineering leaders to balance the technical and business goals of APIs by incorporating these top 10 aspects of APIs.

Overview

Key Findings

- Widespread use of APIs without an overarching API strategy leads to governance challenges, security and compliance risks, and failure to deliver meaningful business outcomes.

- Software engineering leaders often have too many technical use cases involving APIs to focus on causing them to overlook the latest API trends and best practices or its business potential.

Recommendations

As a software engineering leader responsible for APIs, you should:

- Govern your APIs without introducing bottlenecks by establishing a federated API platform team. An adaptive governance approach that uses automatic validation of APIs against style guides is proven to be effective compared to a bureaucratic process of API approval.

- Treat APIs as products by creating the role of an API product manager to promote a consumer-centric product mindset for APIs. Advance your API strategy by applying the top trends and best practices for APIs.

Strategic Planning Assumptions

By 2025, more than 80% of organizations will identify themselves to have implemented advanced or expert level API strategies.

By 2025, more than 75% of organizations will directly or indirectly monetize APIs.
By 2024, API abuses and related data breaches will nearly double.

Introduction

APIs are widely used, as discovered with Gartner’s 2020 Building Digital Platforms Survey finding that 70% of organizations are using API management and mediation to build their digital platforms. ¹ Software engineers and developers around the world have broadly accepted APIs as the primary choice to connect systems, applications and things to build impactful and composable software architectures.

The use of APIs as digital products is also on the rise. Maturing and advancing digital transformation initiatives across a wide range of industries particularly in retail, banking and healthcare have accelerated the creation of innovative API products that are monetized directly or indirectly in an internal or external marketplace. While direct monetization remains a lower percentage of all API use, indirect monetization, API-based business models and headless B2B commerce are all on the rise. In fact, Gartner’s client inquiries on API strategies in 2020 attest to this growing trend of advancing API strategies to create new business value.

Software engineering leaders responsible for API strategies can no longer remain only focused on the technical benefits of APIs (see Figure 1). They must incorporate the business perspectives to capitalize on the business potential and to avoid risking loss of business stakeholder support. So what aspects of APIs do software engineering leaders need to focus on to be successful? This research lays out the top 10 things they need to know and focus on in developing an effective API strategy, such as governing APIs without introducing bottlenecks, treating APIs as products and securing APIs.
What Software Engineering Leaders Need to Know About APIs

<table>
<thead>
<tr>
<th>Don’t create an API governance bottleneck</th>
<th>Adopt API product management</th>
<th>Discover Your APIs before attackers do</th>
<th>Manage the full API life cycle</th>
<th>APIs are for integration, also for app development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prioritize internal API market places</td>
<td>Your teams rely on third-party APIs</td>
<td>APIs are not all REST anymore</td>
<td>Organizations now have multiple API gateways</td>
<td>API platforms vary widely</td>
</tr>
</tbody>
</table>

Source: Gartner
746066_C

Analysis

Govern Your APIs, but Don’t Let API Governance Introduce Bottlenecks

API programs that are not governed by an effective API strategy often lead to the sprawl of APIs in various parts of an organization. APIs that are created locally but never cataloged or managed or rarely used or reused can introduce security risks, data and privacy concerns and prevent adoption. Software engineering leaders should unlock the value of APIs and effectively govern their use and adoption by developing an API strategy (see Gartner’s API Strategy Maturity Model).

The goal of an API strategy should be to provide a framework for all stakeholders to develop, manage and govern APIs without introducing barriers to engineers, developers or specific business areas. This can be achieved through an “adaptive governance” model using automatic validation of APIs against style guides, rather than introducing a more bureaucratic process of API approval. This should take the form of a federated API platform team, as shown in Figure 2 below, rather than a centralized team which can create a bottleneck.
Figure 2. Avoid Bottlenecks With a Federated API Platform Team

**Federated API Platform Team**

- **API Platform Team**
  - Drive overall API strategy
  - Create a community of practice for APIs
  - Manage the overall API portfolio
  - Operate an API management platform

- **API Product Managers**
  - Identify APIs to be managed as products
  - Promote API quality in their BU API team

Source: Gartner
BU = business unit
746066.C

Supporting localized standards, tools and processes may sometimes be necessary. In doing so, exercise caution to ensure that each API product team does not create disjointed, overlapping standards that undermine the overall API strategy. Encourage each API product team to actively participate and contribute in federated API governance. Ensure there is a regular and active feedback cycle across API product managers and the API platform team to improve API standards, to facilitate sharing of common assets, policies and practices (see [Federate, Rebrand and Recharter Your API Center of Excellence to Enable an API Platform Team](#)).

**Treat APIs as Products, Even If You Don’t Plan to Monetize Them**

APIs are essential to a modern digital business strategy. APIs enable creation of innovative digital products and new business models. Gartner’s thousands of interactions in 2020 with clients regarding the use of APIs indicate a growing trend of using APIs to advance digital business strategies in the post pandemic era. Elevating APIs beyond technical purposes to being products is critical to establish and maintain business alignment; and to ensure that it has the business stakeholder’s backing and delivers the much needed business value.
Treating APIs as products involves adopting a consumer-centric product mindset for creation, organization and management of APIs. Create the role of an API product manager who is responsible for establishing API roadmaps, and measuring business outcomes as part of digital product management. Additionally, an API product manager understands and caters to the needs of the customers of API products — for example, developers that consume API products — to not only promote API products but to improve developer relations (DevRel).

Keep in mind, direct monetization of APIs, or API products, is not the primary reason why they should be treated as products. While many organizations monetize APIs, the majority don’t. In fact, many organizations use API products for internal consumption only. Not all APIs need to be treated as products either. Thinking, organizing and managing APIs as digital products provides the best opportunity to effectively prioritize, build and manage APIs through their life cycle; and help your organization achieve successful business outcomes.

Discover Your APIs Before Hackers Do

When it comes to APIs, security is of paramount importance. APIs are the doors to your systems, applications and services. There is way too much at risk and there are numerous possibilities of security vulnerabilities. Each new API increases the surface area of attack and adds a potentially unique attack vector for exploitation. In fact, each year, exploitation of APIs causes private and sensitive information on millions of users to be stolen (see Note 1 for recent examples of security incidents involving APIs).

A comprehensive security strategy that focuses on threat protection, well-refined access control and data privacy is essential (see API Security: What You Need to Do to Protect Your APIs). It is common for software engineering leaders to focus on securing APIs that are designed and published. Often so-called shadow APIs, which are unpublished but used as part of web or mobile applications, are left unsecured. The most vulnerable APIs are the ones that they don’t know exist. You can’t secure what you don’t know about, therefore API discovery is key to ensure there are no blind spots. Adopt a DevSecOps strategy and institute a compliance and security governance policy across the organizations that produce and consume APIs. Evaluate the need for specialist API security solutions to supplement the capabilities of API management solutions, especially for API discovery and discovering malicious usage.

APIs Have Life Cycle, Beyond Their Development

APIs have a life cycle that stems from initial design through to when the API is no longer used. Gartner defines an API’s life cycle to involve four life stages:

- Planning and initial design
- Implementation and testing
- Deploy and run
- Versioning and retirement
A typical organization will have APIs that are in different stages of this life cycle at all times (see Magic Quadrant for Full Life Cycle API Management).

Software engineering leaders must organize both their own and the organization's thought processes around the various life cycle stages to help address different aspects of developing, running and governing APIs. This must be developed into a comprehensive API life cycle management strategy for their organization. For example, this strategy must consist of a design approach and methodology, design tooling as well as a design governance and iterative process. Likewise, APIs in the “deploy and run” stage (such as in a production environment) require a different set of tools, metrics and processes to operate. This divides into basic functionality, such as traffic throttling, and advanced functionality, such as monetization. The entire life cycle is shown in Figure 3 below.

**Figure 3. API Full Life Cycle**

<table>
<thead>
<tr>
<th>API Full Life Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning and Design</td>
</tr>
<tr>
<td>Implementation and Testing</td>
</tr>
<tr>
<td>Deploy and Run</td>
</tr>
<tr>
<td>Versioning and Retirement</td>
</tr>
</tbody>
</table>

- Plan and design the right APIs to enable integration and development
- “API first” or generated from existing infrastructure, with thorough API testing
- API operational management, security, format translation and usage metrics
- Advanced security, analytics to measure API business value, monetization
- Governed API versioning and support for API retirement

Likewise, development of APIs need to also follow the adopted methodology within each organization's DevOps process for continuous integration and delivery. This may introduce additional considerations on how to build, deploy, test and implement APIs in various environments. For example, specific testing guidelines for APIs may involve functional, performance and security testing requirements. It is important to align the technical capabilities and processes for managing the API’s life cycle to that of DevOps for a cohesive software engineering strategy.

**APIs Are Key to Integration, but Aren't Only for Integration**

A composable architecture is a highly desirable characteristic of a modern business, and APIs are essential to achieving composability. This is because APIs are now fundamental to application integration. Many organizations use APIs for integrating legacy systems, assets and data into
mainstream software engineering architectures. However, APIs are not only an integration technology, but are also central to software engineering as a whole.

APIs play a vital role in software engineering architectures involving mobile and multiexperience applications, microservices as well as in API oriented integration architectures. A modern software engineering strategy must accommodate commercial off-the-shelf applications and services seamlessly integrated with distinctly engineered capabilities that provide an indispensable strategic edge to the business. APIs offer the level of abstraction and mediation to create that seamless integrated experience, while allowing previously glued-together architecture to begin to move independently, driving agility and composability.

Software engineering leaders must utilize APIs and capitalize on their potential in these different architectural patterns to modernize their application landscape, unique engineer and business-critical user experiences and to construct an agile and resilient composable business architecture.

API Marketplaces Have Great Value, but Manage Expectations for Public API Marketplaces

API marketplaces bring benefits through API sharing and reuse, especially internal marketplaces. In Gartner inquiries, software engineering leaders have reported success deploying platforms such as Backstage, an open-source project from Spotify, to share APIs across developer teams. These may be part of a developer self-service platform (see How to Build Agile Infrastructure Platforms That Enable Rapid Product Innovation). An example is adidas’ developer hub which provides access to APIs as well as other developer resources (see Building a Platform for Product Team Productivity [adidas]).

In contrast to internal API marketplaces, public API marketplaces have shown less value. Many organizations are enticed by the prospect of registering their APIs in public API marketplaces, hoping that potential users of their APIs will find their APIs there. Gartner has spoken with organizations that have been disappointed with the return from their investment of time and money in registering their APIs in API marketplaces. In fact, in client inquiries, Gartner has found that traditional search engines and direct business relationships remain the ways in which APIs are typically discovered. This means that it is important to take a pragmatic approach to engaging with public API marketplaces (see How to Derive Value From APIs Using API Marketplaces).

Software engineering leaders should ensure that their approach to public API marketplaces is balanced with other go-to-market approaches for their APIs, including the creation of a dedicated API portal (see Create API Portals That Drive API Adoption Among Internal and External Developer Communities).

Your Software Engineering Teams Rely More and More on Third-Party APIs

Most organizations consume more APIs than they produce. These could be publicly exposed APIs, or internal or private APIs that are exposed by their partners, solution providers or third parties. Examples of commonly used third-party APIs include Dun and Bradstreet’s APIs, Google Maps Platform API and Twilio’s APIs. Software engineering leaders often do not consider the health, stability, quality and security
of APIs they consume which ultimately affects their software architecture and often results in major incidents. Examples include an API outage by Auth0, that impacted applications using Auth0 for login, and a broken API leading To Tesla’s entire network going down (see Note 2).

Discover internal and shadow APIs (APIs that exist, but are not tracked or managed) in your areas of influence and adopt a governance model that supports the registration and management of APIs going forward. API catalogs, internal marketplace solutions and API management solutions can provide the tools necessary. Ensure tactical local decisions are made by developer groups, to create APIs that map to data sources, and that other components are reviewed and the APIs are registered for visibility. Adopt an API mediation strategy that involves external APIs.

Most APIs Are REST, but Don’t Forget the Rest of Them

Most APIs are REST APIs. This is natural given the explosive growth of web applications, mobile apps, user experiences and architectural frameworks that support those. Real-time or near real-time application and data interactions also played a key role in the widespread popularity of REST APIs. However, software engineering leaders must recognize REST APIs are not suitable for all scenarios and there are alternative approaches that may be far more suitable for certain architectural patterns.

Non-REST approaches include asynchronous APIs (including the use of webhooks, and AsyncAPI as a description language), events and streams (including Kafka APIs and WebSockets), gRPC, MQTT and GraphQL. These approaches are growing in popularity as the diversity of engineering scenarios to support diverse business needs continues to grow. Standards, tools and offerings in these patterns are continuing to grow and mature. Encourage engineering teams to utilize these emerging technologies in addition to continuing to use REST APIs where appropriate. Also, ensure that your chosen API management solution supports these new types of APIs, not just REST (see Ensure Your API Management Solution Supports Modern API Trends Such as Microservices and Multicloud).

Your Team Is Likely to Use Multiple API Gateways, Which Could Come From Different Vendors

APIs are typically used by high-performing mobile and multiexperience applications and high-volume device level interactions, such as BFF (backend for frontend) patterns and Internet of Things (IoT) interfaces. This introduces performance concerns, as well as the architectural problem that there is no single place to put an API gateway. In fact, one of the concerns for API gateways among software engineering leaders is their potential to create bottlenecks and become single points of failure. This is especially true when multiple teams are building APIs using different platforms (e.g., one team using AWS with Lambda, another using an integration platform) as well as when APIs are being considered for east-west traffic mediations between granular-, mini- and microservices. All of these factors drive the need for multiple API gateways.

When multiple API gateways are used, it is still important for API portals to not be tied to just one API gateway (for more information about the architecture shown in Figure 4 below, see Comparing Architectures for Hybrid and Multicloud API Management).
**Multiple API Gateways**

Consider API management platforms that support multiple different API gateways, as described in *Ensure Your API Management Solution Supports Modern API Trends Such as Microservices and Multicloud*.

**API Management Solutions Vary Widely, and Must Meet the Needs of Your Team**

The field of technologies and vendor solutions for developing and managing APIs is quite diverse, just as diverse as the uses of APIs. The vendor solutions in this market have also evolved from different backgrounds. While the primary market is composed of more than seventy full life cycle API management solutions, there are numerous solutions that focus on specific aspects of APIs as well, some examples include:

- API design
- API testing
- API monitoring
- API security
- API portals and ecosystem management
As this market continues to expand, several vendors in adjoining markets, such as integration platform as a service (iPaaS), application security etc., are also looking to expand their offerings to this market.

It is important for software engineering leaders to understand their specific needs and choose wisely. Start with developing a clear understanding of what different engineering groups need, then move toward what the broader organization needs. Involve engineers, architects, API product managers, API platform teams and security teams in this process. Vendor's viability, maturity and messaging are often not distinctive enough to understand what capabilities are strong in each offering. Review Critical Capabilities for Full Life Cycle API Management to understand the relative strengths and weaknesses of the vendor products along with Gartner's Magic Quadrant for Full Life Cycle API Management for vendor selection. Consider specialist vendors that offer support for specific areas if needed such as API design, API Security and Marketplaces, if needed.

Evidence

1 Gartner's 2020 Building Digital Platforms Survey was conducted online during May and June 2020 among 206 respondents working for organizations in North America and Western Europe with at least 1B USD in annual revenue. Organizations were from the manufacturing and natural resources, communications, media, services, retail, banking and financial services, insurance, healthcare, transportation and utilities industries.

Organizations also had to be working on digital business efforts or have plans to do so, defined as involving IoT, delivery of public APIs, private/B2B APIs, or a combination thereof. Quotas were set to ensure a majority of organizations have a fully implemented digital business initiative.

Respondents were required to have a job title of director or more senior and to be involved in either digital business, data analytics, IoT- or API-based platforms for partners. In respect to digital business initiatives, they were also required to have a role in either defining technology requirements, investigating or evaluating service providers or making final decisions.

Results of this study do not represent global findings or the market as a whole but reflect sentiment of the respondents and companies surveyed.

Gartner's Inquiry evidence — In the year 2020, Gartner had approximately 5000 inquiries with clients about topics directly covered in this research such as API strategy, API management, API governance and API security. Total inquiries in which APIs were discussed for the same period was approximately 6000. Clients often discuss best practices, strategic approaches and specific measures to improve security posture, governance and business outcomes for APIs.

Note 1: Examples of Recent Security Incidents Involving APIs

- API Data Breaches in 2020, CloudVector
Note 2: Two Notable Health, Stability, Quality and Security Related API Incidents

- Increased Errors In Auth0, Auth0 Status (An API outage by Auth0 impacted on applications which use Auth0 for login)
- How A Broken API Led To Tesla's Entire Network Going Down, Security Boulevard
Become a Client

Get access to this level of insight all year long — plus contextualized support for your strategic priorities — by becoming a client.

gartner.com/en/become-a-client

U.S.: 1 800 213 4848
International: +44 (0) 3331 306 809

About Gartner

Gartner is the world’s leading research and advisory company and a member of the S&P 500. We equip business leaders with indispensable insights, advice and tools to achieve their mission-critical priorities today and build the successful organizations of tomorrow.

Our unmatched combination of expert-led, practitioner-sourced and data-driven research steers clients toward the right decisions on the issues that matter most. We are a trusted advisor and an objective resource for more than 14,000 enterprises in more than 100 countries — across all major functions, in every industry and enterprise size.

To learn more about how we help decision makers fuel the future of business, visit gartner.com.