Hype Cycle for APIs and Business Ecosystems, 2021

Published 27 July 2021 - ID G00747571 - 84 min read

By Analyst(s): Mark O'Neill, John Santoro

Initiatives: Software Engineering Technologies

APIs enable applications to connect to other applications and to data, as well as engage developers and provide the foundations of business ecosystems. Software engineering leaders can leverage technologies in this Hype Cycle to navigate the postpandemic world by creating new value chains.

Additional Perspectives

- Summary Translation + Localization: Hype Cycle for APIs and Business Ecosystems, 2021
  (08 September2021)

Analysis

What You Need to Know

APIs form the foundation of application connectivity as well as new business models, by exposing business data and capabilities in a consistent, easy-to-consume fashion. While many APIs are implemented to meet internal application needs, such as opening access to legacy systems, APIs are now expected to be provided as part of SaaS offerings and software products in general. These are used by customers and partners for integration — to create composable applications — and by API providers to enable business ecosystems.

Business ecosystems populated with partner solutions allow companies to better meet their customers' needs and to improve customer engagement. They also provide opportunities for expansion and indirect revenue, as well as to potentially offer a new source of direct revenue by monetizing the ecosystem. Partner ecosystem management platforms, although early in adoption, enable this activity. A better connected ecosystem can operate more efficiently by eliminating legacy methods of communicating and trading with partners, and instead move to the use of APIs. Furthermore, since APIs provide the interface for ecosystem content, API marketplaces simplify their discovery and adoption. API management provides controls and tracks usage to enable monetization, while API mediation and security also become critical factors for successful ecosystem implementation.

The Hype Cycle

This year's APIs and Business Ecosystems Hype Cycle highlights a number of important trends:

- **The importance of securing APIs.** Security is a growing concern for API programs. This is reflected in this Hype Cycle in the maturing of the innovation profiles for API protection, API access control, and API security testing. API security incidents continue to mount (to read about examples, see the Notes within API Security: What You Need to Do to Protect Your APIs).

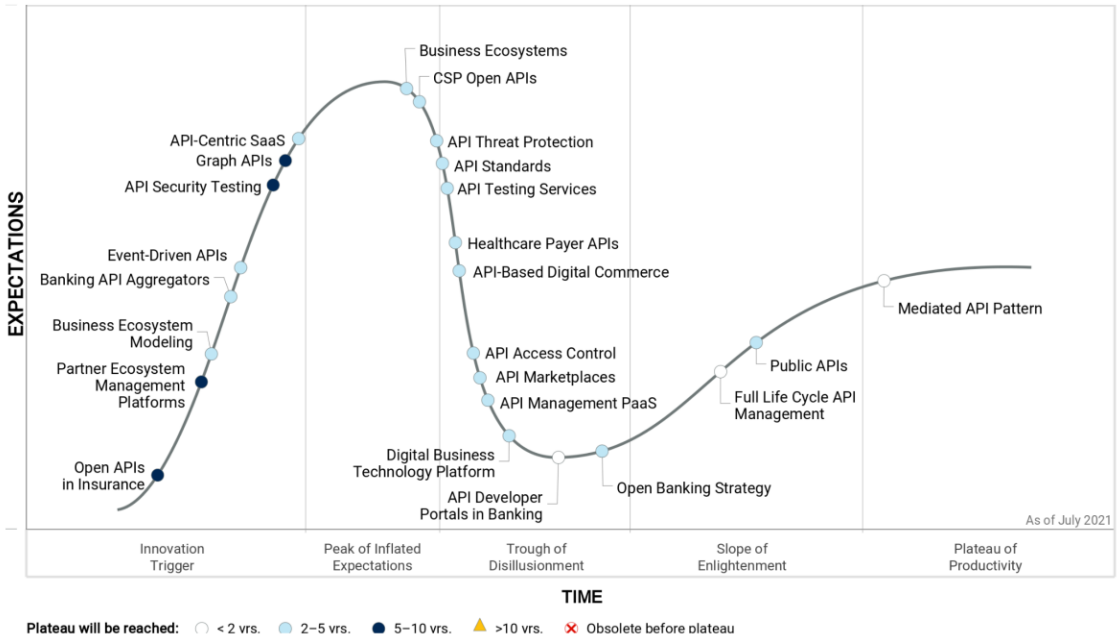
- **Open banking is finally beginning to show success, as other industries follow its lead.** Open banking is now beginning to leave the Trough of Disillusionment and move toward the Slope of Enlightenment, as banks move beyond simple compliance with open banking regulations to create new API products (see Beyond Compliance: Using Banking API Standards for Competitive Advantage). The Hype Cycle shows that other industries including: healthcare, insurance and telecom communication service providers (CSPs) are behind open banking in the progress of their API strategy, but are nonetheless moving forward.

- **API technology is moving beyond REST.** Event-driven APIs are beginning to climb up the Hype Cycle, reflecting the growing awareness and usage of AsyncAPI and other technologies for asynchronous APIs. Graph APIs are also in the Innovation Trigger stage, as organizations follow the lead of early adopters to deliver graph APIs, including using GraphQL.

- **API marketplaces struggle to show business ecosystem value.** Both API marketplaces and API marketplace platforms as a service (PaaS) are descending toward the Trough of Disillusionment. This reflects the difficulties in defining API marketplace business models and generating value (see How to Derive Value From APIs Using API Marketplaces). However, as noted in the Innovation Profiles, internal API marketplaces are showing significant value to developers.

While the future is uncertain, one thing is clear — APIs and business ecosystems will continue to provide both risk and opportunity.

Figure 1. Hype Cycle for APIs and Business Ecosystems, 2021



Source: Gartner (July 2021)

Downloadable graphic: Hype Cycle for APIs and Business Ecosystems, 2021

The Priority Matrix

The majority of the transformational entries in this year's APIs and Business Ecosystems Hype Cycle are related to particular industries. This reflects the impact of APIs and ecosystems on digital commerce, banking, healthcare and telecom CSPs.

Full life cycle API management and API mediation are showing their maturity with widespread adoption. However, API security testing, API marketplaces, graph APIs (including the use of GraphQL) and APIs in insurance are currently less widely adopted.

Table 1: Priority Matrix for APIs and Business Ecosystems, 2021

(Enlarged table in Appendix)

Benefit	Years to Mainstream Adoption			
↓	Less Than 2 Years ↓	2 - 5 Years ↓	5 - 10 Years ↓	More Than 10 Years ↓
Transformational		API-Based Digital Commerce CSP Open APIs Digital Business Technology Platform Healthcare Payer APIs Open Banking Strategy		
High	Full Life Cycle API Management Mediated API Pattern	API-Centric SaaS API Testing Services API Threat Protection Business Ecosystem Modeling Business Ecosystems Public APIs	API Security Testing Open APIs in Insurance Partner Ecosystem Management Platforms	
Moderate	API Developer Portals in Banking	API Access Control API Management PaaS API Marketplaces API Standards Banking API Aggregators Event-Driven APIs	Graph APIs	
Low				

Source: Gartner (July 2021)

On the Rise

Open APIs in Insurance

Analysis By: Sham Gill

Benefit Rating: High

Market Penetration: Less than 1% of target audience

Maturity: Embryonic

Definition:

Open APIs (aka public or external APIs) are application programming interfaces (APIs) that are published for consumption by third-party users and applications. Open APIs typically take the form of lightweight REST APIs, suited to mobile and web consumption. API providers often enable consumers (developers) to use a self-service portal to register and gain access to public APIs. In some cases, open APIs may be monetized, for example through subscription or pay-per-use.

Why This Is Important

APIs are fast moving from a technical must-have for the IT department to a board-level agenda item, and insurance companies must follow suit. It's important that insurance CIOs recognize that open APIs aren't just a technology issue. They enable connectivity between the insurance company and its external ecosystem partners. As such, they need to be considered as reusable products in their own right, with their own intrinsic business value.

Business Impact

- Publication of their own open APIs and leveraging of partner-open APIs can be powerful allies for digital transformation in insurance.
- Open APIs in insurance hold the potential to expand an insurer's reach to a broader audience through connectivity with business ecosystem partners.
- They also enable faster delivery of new products, services and business models that enable the direct and indirect monetization of APIs.

Drivers

- As the number of partners engaged in business ecosystems and the complexity of those interactions increases, the need for APIs will intensify.
- Open APIs give insurers an opportunity to create new revenue, open up a new business channel, add customer reach, enhance the customer experience, increase product "stickiness," and embrace new ecosystems.
- New business models will demand greater connectivity as innovative insurers move from digital platforms to platform businesses; see Case Study: Insurance API-Driven Digital Ecosystem Transformation (Wakam). Open APIs will enable them to move faster with new products, services and business models.
- Increasing availability of partner and third-party open APIs will help insurance CIOs to accelerate their digital platforms and deliver business value (see Top Cross-Industry Open API Trends for Insurance CIOs to Track in 2021).
- Open APIs can also play a vital role in digital optimization efforts. For example, a European life insurer has exposed a limited set of open APIs to enable group life member data to be updated directly from the employer's HR systems in order to drive cost optimization.
- Centralized control of APIs and the creation of APIs as products will enable CIOs to minimize duplication and take a strategic approach to the development and deployment of APIs.

Obstacles

- Most insurance CIOs don't fully understand the strategic and business value of open APIs and, therefore, still build integrations using traditional approaches, like custom-built point-to-point product extensions.
- The lack of standards and regulatory or industry bodies for approving and registering third parties and monitoring usage of open APIs across life and P&C insurance will add additional burden on insurance companies (see 5 Best Practices Insurance CIOs Can Adopt From Open Banking Initiatives to Shape Their API Strategy).
- Publishing APIs alone is not enough. An effective API-based ecosystem requires careful consideration of target markets and close collaboration with partners who will design effective products on the API platform.

- The few, limited open insurance API use cases focus on specific parts of the insurance value chain, like data aggregation, quotes and new business, and predominantly on simpler P&C products, like travel and gadget insurance.

User Recommendations

- Justify the additional design and governance required to create open APIs for in-progress projects by leveraging evidence of future business value, and shift the IT focus from project-specific APIs to strategic-open APIs.
- Track API transactions by using API management tools to monitor message volumes and trends, and provide business stakeholders with clear evidence of the need for open APIs.
- Link API usage to tangible business value by creating an internal monetary representation of transactions to support the case for further investment.
- Convince partners to experiment with creating new open API-based services by creating sandbox environments that require less rigorous scrutiny and registration than production versions to reduce the barriers for adoption.
- Prepare for future API deployments through API marketplaces or API aggregators by using full life cycle API management to create internal API marketplaces and test adoption.

Gartner Recommended Reading

Top Cross-Industry Open API Trends for Insurance CIOs to Track in 2021

5 Best Practices Insurance CIOs Can Adopt From Open Banking Initiatives to Shape Their API Strategy

Case Study: Insurance API-Driven Digital Ecosystem Transformation (Wakam)

Insurance CIOs Must Create a Strategic API Pathway to Succeed With Their Future Digital Ambitions

Partner Ecosystem Management Platforms

Analysis By: Ilona Hansen

Benefit Rating: High

Market Penetration: 1% to 5% of target audience

Maturity: Emerging

Definition:

Partner ecosystem management platforms support an open network of partners, peers and manufacturers to drive revenue with vendors and with other partners. These virtual platforms integrate existing partner relationship management technologies, supporting distribution and sharing of data and content among partners. They ease collaboration in co-selling, marketing and reselling for vendors and partners in the indirect sales channel.

Why This Is Important

Partner ecosystem management platforms change the traditional indirect sales market, which commonly relies on its partner portal functionalities. They enable all stakeholders to open up data sources for collaboration and exchange with multiple parties within the ecosystem — across one virtual environment. Technologies are seeing higher demand as many producers/vendors are exploring alternatives to increase access to end customers and improve exchange with their resell partners.

Business Impact

Partner ecosystem management platforms allow vendors and partners to be more effective and make decisions faster. Users can interact with any type of content and present it on one application across all stakeholders. These platforms enable working with each party's data in one core application, based on the common virtual platform. They also support indirect selling, co-selling models and alliance management strategies.

Drivers

- During the next three to five years, new capabilities being added to the partner data exchange platform will allow additional business activities to be performed, such as real-time insights and comprehensive business analytics.
- Some vendors in this space provide additional capabilities such as business planning support, main distribution frame (MDF) management, marketing functionalities and partner learning features.

Obstacles

- Partner ecosystem management platforms only provide the infrastructure for collaboration, while partner relationship management (PRM) apps are managing tasks.
- Capabilities for performing business activities like real-time insights and comprehensive business analytics are not currently available. These will be added as partner ecosystem management platforms develop further during the next three to five years.

User Recommendations

- Assess partner ecosystem management platforms for sales activities between vendors and sales partners, especially for supporting collaboration on the wealth of unstructured and structured data available, without custom coding and complex data models.
- Compose indirect technology stack by integrating earlier PRM apps investments into the partner ecosystem management platform.

Sample Vendors

Crossbeam; Impartner; Mindmatrix; Vartopia; Webinfinity; WorkSpan

Gartner Recommended Reading

Market Guide for Partner Relationship Management Applications

Toolkit: Choosing the Right Partner Relationship Management Vendor

Business Ecosystem Modeling

Analysis By: Marcus Blosch

Benefit Rating: High

Market Penetration: 1% to 5% of target audience

Maturity: Emerging

Definition:

A business ecosystem model is a dynamic network of entities (people, businesses and things) that interact to create and exchange sustainable value for participants. Business ecosystem modeling focuses on extending the scope of business architecture to the business ecosystem of customers, suppliers, devices, partners and organizations that make up an organization's ecosystem.

Why This Is Important

All organizations exist in business ecosystems that include customers, partners, competitors, regulators, suppliers and other entities. The business ecosystem is made up of a complex set of relationships, roles and dynamics. To gain insight into the business ecosystem and its dynamics, and develop effective business strategies, models of the business ecosystem are needed.

Business Impact

Although organizations have always existed in business ecosystems, the digital age has driven up the complexity and the number of the networks of relationships connecting participants. Business ecosystems now extend around the globe, mediated by technology, and many business models are based on business ecosystems. Models enable organizations to understand an ecosystem and its dynamics and shape effective strategies.

Drivers

- Business strategy is about taking action in a business ecosystem, whether it's entering a new market or reaching new customers. To be effective, a model of the business ecosystem, its participants and dynamics is needed.
- Many of the most successful business models — from Amazon and Alibaba to WeChat, Uber and Airbnb — are based on the business ecosystem. These business models specifically leverage the business ecosystem; however, doing this effectively is complex.
- Business ecosystems are, by nature, complex adaptive systems, so models and, ideally, simulations are of great value to decision makers. With a well-designed model, scenarios can be played out, and more sophisticated strategies shaped.
- Business ecosystems also facilitate open innovation, using the resources of competencies of partner organizations and building them into the organization's business and operating model. The organization becomes effectively fragmented across the business ecosystem, and ecosystem models are needed to design and manage these relationships.
- Technology is driving the interconnection of organizations and society. Ecosystem modeling helps organizations understand the nature and dynamics of these interconnections.

Obstacles

- Business ecosystems represent a change in perspective, away from an internal input/output (I/O) perspective, to an external ecosystem (or complex adaptive system) perspective.
- New skills and competencies, along with tools are needed to model and understand the dynamics of the business ecosystem.
- Data science, simulation, statistical analysis are all complementary skills for more-advanced ecosystem modeling, but are in short supply.
- Some modeling tools are available, but they are not fully mature and lack widespread adoption and understanding.

User Recommendations

- Begin by understanding the concept of business models, and learning how organizations have leveraged them to optimize and transform their operations.
- Develop simple business ecosystem models for your organization, and identify the participants, their roles, their relationships and interrelationships.
- Work with business executives to build and refine business ecosystem models to highlight monetization opportunities, threats and challenges that are external to the organization.

Sample Vendors

Avolution; BiZZdesign; Inlecom; Kumu; Tr3Dent; WorkSpan

Gartner Recommended Reading

Expanding Your Business Ecosystem

8 Ways Ecosystems Supercharge Digital Business Models

Platform Business Models That Adapt and Disrupt

Model Your Ecosystem to Identify the Partners Needed for Digital Business

Banking API Aggregators

Analysis By: Don Free, Mark O'Neill

Benefit Rating: Moderate

Market Penetration: 1% to 5% of target audience

Maturity: Adolescent

Definition:

Banking API aggregators provide APIs that connect to multiple banks, thus simplifying banking integration for application developers. APIs provided by banking API aggregators typically include balance verification and funds availability APIs. Customers of banking API aggregators include fintechs, lenders and banks themselves.

Why This Is Important

Fintechs, lenders and app providers increasingly require access to bank accounts. Although open banking regulations exist in many countries that require banks to provide APIs, APIs remain unstandardized, and developers often prefer to use an intermediary for simplicity. Banking API aggregators provide this capability, using banking APIs or in a customer-permissioned "screenscraping" approach. Increasingly, banking API aggregators provide emergent capabilities, such as risk analysis and payment.

Business Impact

- For application developers, banking API aggregators simplify access to banks' data and services.
- Larger banks generally prefer to directly control their API access and not delegate access to an aggregator. Midtier or smaller banks may benefit from not having to invest in an expensive API platform to create or publish their own APIs.
- Banking API aggregators benefit fintechs by allowing them to work with a broad set of banks without the cost and expense of integrating with each bank individually.

Drivers

- A wide variety of organizations, including lenders, e-commerce sites and fintechs providing services, require access to banking data, which drives the need for banking API aggregators to provide this data.
- APIs are now the preferred mechanism to integrate with banks due to widespread developer skills and tooling support. Banks also favor APIs due to the ability to apply security and traffic throttling.
- Many banks do not currently provide APIs, and where they do, the APIs typically differ from each other. This drives the need for banking API aggregators to provide single APIs in front of multiple banks that may or may not have APIs of their own.

Obstacles

- Some larger banks have reacted to banking API aggregators by blocking their connections. This is because they prefer partners to use the bank's own APIs directly.
- Privacy is a concern when banking customers share their online banking credentials with banking API aggregators as part of "account linking." In jurisdictions with open banking regulations, this concern is addressed by permission-based account linking without password sharing, rather than the use of "screen scraping."
- Banking API aggregators typically operate in just one country or one region. Some banking API aggregators, such as Sweden-based Tink, with its acquisition of Eurobits in Spain and FinTecSystems in Germany, are adding geographical reach through acquisition.

User Recommendations

- Evaluate banking API providers if your organization requires access to banking services, such as account verification, across different banks.
- Question banking API aggregators on their security and privacy controls.
- Compare the pricing models of banking API aggregators before choosing one provider, since they vary on their pricing models.

Sample Vendors

Belvo; Brankas; Finicity; MX; Plaid; Salt Edge; Tink; TrueLayer; Yodlee (Envestnet)

Gartner Recommended Reading

Beyond Compliance: Using Banking API Standards for Competitive Advantage

Event-Driven APIs

Analysis By: Mark O'Neill

Benefit Rating: Moderate

Market Penetration: 1% to 5% of target audience

Maturity: Emerging

Definition:

Event-driven APIs describe the interface to enable a listener to subscribe to notifications of state changes managed by an application or service. Event-driven APIs differ from traditional request/response REST APIs because they enable asynchronous communication. Popular standards include AsyncAPI.

Why This Is Important

- Event-driven APIs enable real-time notification of changes to data and application state.
- Industries such as financial services require real-time updates of data and application state, which has driven the need for event-driven APIs.
- Event-driven APIs enable modern architecture patterns including microservices architecture and mesh app and service architecture (MASA).

Business Impact

- Event-driven APIs open up business opportunities related to the provision of real-time data to enable faster response to change and streaming analytics.
- They can also enable application functionality such as notifications, which can be the basis of new employee-facing or customer-facing functionality.
- Event-driven APIs bring advantages such as enabling push notifications, which are much more efficient and less expensive (from both time and networking perspective) than polling.

Drivers

- Mobile and multiexperience applications involve the widespread use of events, such as Webhooks and mobile notifications using server-sent events (SSEs) delivered over HTTP.
- Data analytics and artificial intelligence drive new use cases related to events and notifications, which in turn drive the need for event-based APIs.
- Open-source and cloud event-broker technologies, especially Apache Kafka, have raised awareness and accessibility to publish-subscribe infrastructure and encouraged adoption of EDA.
- The Open API Specification (OAS) version 3, introduced in 2017 and now reaching widespread adoption as a standard for publishing APIs, includes provision for callbacks as a means to describe event-driven APIs. OAS version 3.1 (February 2021) adds support for Webhooks, the most common event-driven API approach for internet-facing APIs.
- AsyncAPI, which defines a standard for how event-driven APIs are published, has been submitted to the Linux Foundation, which is the same standards body which houses OAS.

Obstacles

- Event-driven APIs require API design and development skills which differ from the more common REST API design and development skills.
- API mediation products, such as API gateways, are often built or configured only for request-response APIs. In some cases, the patterns of event-driven APIs (including fire-and-forget, or publish-and-subscribe methods) are not supported in the API mediation product.
- Protocol support for event-driven APIs is diverse, including Webhook (HTTP/S), WebSockets, MQTT, advanced message queuing protocol (AMQP), and Apache Kafka. This reduces focus and makes decision making more difficult for end-users and vendors.
- Although API portals are widely used for publishing request/response APIs, their usage for event-driven APIs is nascent.
- Event-driven APIs, like all event-driven architecture, introduce challenges for debugging and testing.

User Recommendations

- Evaluate whether your chosen API mediation products, including API gateways as part of API management, support event-driven APIs (see The Impact of Event-Driven IT on API Management).
- Adopt a community-of-practice approach to raise your organization's overall skills on event-driven APIs and their relationship to event-driven architecture and real-time stream processing and analytics.
- Identify opportunities to replace request/response APIs with event-driven APIs within your overall API portfolio to enable both internal and external integration scenarios.

Sample Vendors

Ably; Axual; Axway (Streamdata); Kafkawize; Postman; Software AG; Solace; TIBCO Software

Gartner Recommended Reading

The Impact of Event-Driven IT on API Management

Proven Practices to Enhance Skills in Application Architecture, Development and Software Engineering

Innovation Insight for Event Brokers

Maturity Model for Event Driven Architecture

API Security Testing

Analysis By: Dale Gardner

Benefit Rating: High

Market Penetration: 1% to 5% of target audience

Maturity: Emerging

Definition:

API security testing is a specialized type of application security testing (AST) aimed at identifying vulnerabilities in application programming interfaces (APIs). Checks should include both traditional application vulnerabilities (e.g., injection attacks) and API-specific issues (e.g., broken-object-level authorization). Availability of an API definition (e.g., OpenAPI) is often, but not always, a prerequisite for effective testing; discovery helps ensure that unknown APIs are identified.

Why This Is Important

Attacks on applications are shifting to APIs, and their pace is increasing. By 2022, API abuses will be the most-frequent attack vector, and will result in data breaches. DevSecOps teams are focusing attention on the need for improved API testing in development, looking to a mix of traditional tools — e.g., static AST (SAST) and dynamic AST (DAST) — and emerging solutions focused on the requirements of API testing to identify the optimal approach to API testing.

Business Impact

APIs are a foundational element of many organizations' digital transformation efforts. Hence, securing APIs from attack and misuse is a growing concern for many security and risk management (SRM) professionals. API-specific testing, pre- and post-deployment, builds a solid foundation for an overall API security strategy. API discovery is needed, because many organizations struggle to maintain an inventory of APIs and need help locating them and ensuring they are tested and managed.

Drivers

- APIs enable information flow and support transactions between processes, programs and systems, so APIs have become common in application architectures, especially in support of digital transformation efforts. However, that growth has brought increased attention from attackers, and APIs have become the primary attack surface for many systems. Gartner has long noted that attacks and abuse against APIs are a common attack vector.
- These attacks have resulted in an endless stream of data breaches and other security incidents, yielding significant damage to organizations and individuals. As a consequence, DevSecOps teams — along with the business leaders whose applications are supported by APIs — are increasingly interested in API testing and security.
- Because the creation, development and deployment of APIs may be loosely managed, security teams often have to contend with unknown APIs, which exist outside normal processes and controls. Such APIs may be especially deficient in secure design and testing, posing an even greater risk. Hence, the discovery of APIs deployed to production is another important need.
- Given their increasing importance in application architecture and their sensitivity, identifying and remediating exploitable vulnerabilities in APIs has become an obvious goal. API security testing helps avert the tangible and intangible costs associated with breaches and other types of security incidents.
- Traditional AST tools — SAST, DAST and interactive AST(IAST) — were not originally designed to test for vulnerabilities associated with typical attacks against APIs, or for newer types of APIs (e.g., GraphQL). Although vendors have added support, these gaps have created an opportunity for specialist API security vendors, focused on testing, discovery of running APIs and protection from threats — or some combination of the three.

Obstacles

- Existing tools must be extended to address API-related vulnerabilities, and incorporate new testing approaches. APIs are susceptible to most, if not all, traditional attacks against applications; however, newer attack types have also emerged. For example, improper authorization checks around object identifiers (such as a user identifier) have been cited in a number of breaches.
- APIs in a number of forms are using various protocols. Simple Object Access Protocol (SOAP) remains in widespread use, although has been supplanted by Representational State Transfer (REST) APIs. GraphQL (a combination of query language and runtime) is now challenging REST. This requires additional support from tool vendors for effective tests.
- There is confusion in the marketplace, with various types of companies (traditional AST vendors, along with API testing and protection vendors, etc.) offering products. This complicates evaluation and selection efforts.

User Recommendations

- Do not invest in tools without a solid inventory of APIs — unless the tool is specifically intended to help address shortcomings in this area.
- Assess the overall role APIs play in your application portfolio —their criticality to the organization, their security and business risk, and the technical requirements they pose — to begin evaluation and selection efforts.
- Examine the testing capabilities provided by existing tools in your application security portfolio. Many have added support for APIs, although actual tests may still focus primarily on traditional application vulnerabilities.
- Where gaps are found, evaluate newer alternatives. They may also offer additional options, including audit of design-stage API specifications, as well as vulnerability scans and threat protection.
- API security tool capabilities — including discovery, testing and threat protection — overlap, so evaluate the ability of a given tool to address multiple requirements.

Sample Vendors

42Crunch; APIsec; Checkmarx; Contrast Security; Data Theorem; Imvision; NTT Application Security; Salt; Synopsys; Veracode

Gartner Recommended Reading

API Security: What You Need to Do to Protect Your APIs

Solution Path for Forming an API Security Strategy

Magic Quadrant for Application Security Testing

Critical Capabilities for Application Security Testing

Graph APIs

Analysis By: Mark O'Neill

Benefit Rating: Moderate

Market Penetration: 1% to 5% of target audience

Maturity: Emerging

Definition:

A graph API models its data as an object graph (a set of objects and relationships) described using nodes and edges. It enables users to query and manipulate the data by following and filtering on the relationships, in addition to the objects. This contrasts with a traditional REST API, which models data as resources without relationships. Many graph APIs, such as the Facebook Graph, provide an abstraction over multiple distinct data sources and APIs. GraphQL may be used to define graph APIs.

Why This Is Important

Graph APIs are gaining in importance because developers benefit from flexible self-service access to data. User and consumer needs drive API usage in multiple unpredictable ways, so dedicated APIs become harder to define, compared to graph APIs. Finally, the value of data and the need to exploit information held in data relationships drives the importance of graph APIs.

Business Impact

For API providers, graph APIs unlock business value in data, and in the relationships between data in particular. Graph APIs also provide more efficient use of resources, compared to the overhead of individual REST API calls. For API consumers, graph APIs can simplify integration because developers do not need to consume multiple different APIs from the same API provider in order to build an application or an integration.

Drivers

- The need for developer self-service access to data is a key driver for graph APIs.
- API marketplaces such as RapidAPI have begun to include graph APIs, which offers another point for the publication and discovery of graph APIs.
- Vendors including Microsoft, with Microsoft Graph API accessing multiple products including Microsoft Azure AD and Exchange Online, and SAP, with SAP Graph API accessing multiple SAP applications, including SuccessFactors and S4/HANA, now provide graph APIs across multiple products.

Obstacles

- Graph APIs place significantly more burden on teams supporting APIs and require investment in new design, security, monitoring and governance skills. In addition, vendor support for GraphQL is still not widespread. Security, governance and performance optimization practices are still being developed.
- Privacy concerns have arisen for graph APIs, in particular related to Facebook's original Graph API and its usage by Cambridge Analytica, leading Facebook to address these concerns in later versions.
- Performance is a consideration due to the amount of processing required to traverse a graph.
- Design skills for graph APIs are currently behind the skills for REST APIs.

User Recommendations

- Evaluate Graph APIs when they are available from their platform providers, and adopt them to provide flexible self-service access to data, including for applications used by their users such as dashboards and mobile apps.
- Replace dedicated “experience APIs” with graph APIs that tailor API experiences to different API consumers. This is because graph APIs offer the consumer more flexibility and control over how related data is accessed than traditional APIs.
- Adopt a community of practice approach to spread both Graph API and GraphQL skills in your organization as for both API consumers and API providers, graph APIs introduce new skills.

Sample Vendors

Apollo; Facebook; Google; Hasura; IBM; Microsoft; RapidAPI; SAP; StepZen; Tyk; WSO2

Gartner Recommended Reading

Assessing GraphQL for Modern API Delivery

Ensure Your API Management Solution Supports Modern API Trends Such as Microservices and Multicloud

API-Centric SaaS

Analysis By: Yefim Natis, Mark O'Neill, Anne Thomas

Benefit Rating: High

Market Penetration: 1% to 5% of target audience

Maturity: Emerging

Definition:

API-centric (“headless”) SaaS is a cloud application service that is offered primarily or entirely for programmatic access via APIs or event subscriptions. Some user experience may be provided, but the strategic intent for API-centric SaaS is to be used as a business capability, programmatically packaged for application composition, development and integration.

Why This Is Important

Increasingly, businesses seek a greater role in defining their digital experiences. API-centric SaaS enables customers to define their differentiated application experiences instead of being bound to the user experience packaged by the vendor. API-centric SaaS allows capabilities such as payments or mapping to be used within larger applications. Although as an application (SaaS), the API-centric SaaS is packaged for use in development platforms (PaaS), blurring the boundaries of SaaS and PaaS.

Business Impact

Business organizations that are equipped to capitalize on API-centric SaaS can create new application experiences for their employees and customers through composition of those prepackaged business capabilities, giving them an opportunity for faster, safer and more efficient business innovation. API-centric SaaS is also a base for fast-paced innovation by startups and other software vendors, contributing to the increasing pace of digital business change.

Drivers

- The patterns of software innovation have embraced integration and composition to a degree that application vendors that want to be included must offer their services, at least optionally, as libraries of API-enabled functional capabilities.
- The technology and skills for integration, including management of APIs and event streams, are widespread, allowing some leading businesses to package their business capabilities behind APIs for customers' and partners' customized use. API products serve that purpose.
- The demands for the depth of customization of applications in business organizations have evolved to the point that SaaS vendors must allow rearrangement of their business functionality by the customers. API SaaS serves that purpose.
- Increasing use of digital twin architecture in IoT application design serves as the early experience of utilizing API-packaged business capabilities, preparing organizations' skills and technologies for the use of the similarly designed and encapsulated API-centric SaaS.
- Many older applications are used as if "headless" by businesses accessing them via API and foregoing the vendor-defined user experiences. This prepares organizations' skills and technologies to include the API-centric SaaS capabilities into their application development experience.
- Business application design has become significantly partitioned into the back-end functionality with its APIs and the front-end multiexperience user interface, each side created using different tools and design expertise. Some business-oriented application vendors find it convenient to concentrate on the back-end data and business logic and leave the finalized user experience to separate teams, including the customer's own developers.

Obstacles

- API-centric SaaS is a relatively new phenomenon, when compared to the conventional SaaS offerings. There are not enough dedicated development tools or trained software engineering teams to assure the right balance of cost and value of adopting it.
- API-centric SaaS offers business capabilities, typically delivered as a collection of multiple APIs, but most leading marketplace technologies are designed to manage only the individual APIs, making the discovery and governance of such packaged business capabilities more difficult.
- Minimal or fully absent user interfaces packaged with an API-centric SaaS assume and require that the customer implement their own differentiated application and user experience. What is a welcome opportunity for innovation for some can be a burden to others, inhibiting the wide-spread adoption and API-centric SaaS.

User Recommendations

- Give preference to SaaS offerings that expose more of their business capabilities as API and/or event streams.
- Plan for gradual shift of development to composition and integration of API-centric packaged business capabilities.
- Ensure clean separation of the back-end business logic and the front-end user experience in all applications, to maximize future benefits of the composable application experiences.
- Avoid vendor applications that lock your organization into their user experience technology.
- Give preference to low-code and pro-code PaaS offerings that are well-equipped for access to external API and event marketplaces.
- Practice use of API marketplaces in preparation to greater adoption of API-centric SaaS and API products
- Watch for opportunities to experiment by offering some of your business capabilities as priced API products.

Sample Vendors

Algolia; Alloy; Clearbit; Cloudinary; Lob; MessageBird; Plaid; Strapi; Stripe; Twilio

Gartner Recommended Reading

Create API Portals That Drive API Adoption Among Internal and External Developer Communities

Kick-Start Your Composable Business Journey With 2 Key Strategies

Gartner's API Strategy Maturity Model

Top Trends in Cross-Industry Open APIs for Product Leaders at Banking TSPs for 2021

Choose the Right API Monetization and Pricing Model

At the Peak

Business Ecosystems

Analysis By: Marcus Blosch

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

A business ecosystem is a dynamic network of entities (people, businesses and things) interacting with each other to create and exchange sustainable value for participants. A business ecosystem allows participants to work cooperatively and competitively to support new products, satisfy customer needs and innovate.

Why This Is Important

All organizations exist in a business ecosystem that includes customers, partners, competitors, regulators, suppliers and many other entities. The business ecosystem is a source of opportunity, and many of today's most successful business models are based on taking advantage of the full ecosystem. The business ecosystem is also a source of risk, new competitors, supplier failures, changes in regulations or even global pandemics.

Business Impact

Business ecosystems define successful businesses:

- From Alibaba to WeChat and Amazon to Uber, many of the most successful business models of the last decade are business ecosystems.
- Increasingly, business strategy is turning outwards into the business ecosystem.
- The business ecosystem is also the source of solutions to complex problems, from climate change to smart cities, bringing together a diverse range of stakeholders to collaborate on innovative solutions.

Drivers

- All business strategy takes place in the business ecosystem, from introducing a new product to entering a new market or acquiring another company. Successful strategy depends on understanding the business ecosystems.
- The business ecosystem is a key source of opportunity. Many of the most successful business models, from Alibaba to WeChat and Airbnb to Uber, are based on leveraging the business ecosystem. Business models explicitly based on the business ecosystem are becoming increasingly popular.
- Complex problems often can only be solved by using a business ecosystem that brings together different stakeholders, perspectives and resources. Climate change, urban development and creating smart cities, for example, all required an ecosystem-based strategy.
- Organizations can become more agile and adaptive by using open architectures and leveraging the competencies and capabilities available to them in the business ecosystem. Developments in technology such as cloud, analytics, modular, service-oriented, API-based architectures make this straightforward.
- Information is becoming one of an organization's most important assets, and much of this information is to be found external to the organization and within the broader ecosystem. Sentiment analysis, supplier backlogs, weather patterns and much more can now be used as input to complex algorithms.

Obstacles

- Business ecosystems represent a change in perspective, away from an internal input-output perspective to an external ecosystem (or complex adaptive system) perspective.
- New skills and competencies are needed to model and understand the dynamics of the business ecosystem. Most organizations lack this expertise today.
- Creating new ecosystem-based business and operating models is challenging, particularly where there is an entrenched traditional business model.
- Some modeling tools are available for organizations to visualize the ecosystem, but they are not widespread.

User Recommendations

- Understand the concept of business models and learn how organizations have leverage them to transform their operations.

- Develop simple business ecosystem models for your organization; identify the participants, their roles, and relationships and interrelationships.
- Work with business executives to build and refine business ecosystem models to highlight opportunities and challenges that are external to the organization.

Gartner Recommended Reading

Expanding Your Business Ecosystem

8 Ways Ecosystems Supercharge Digital Business Models

Platform Business Models That Adapt and Disrupt

Model Your Ecosystem to Identify the Partners Needed for Digital Business

CSP Open APIs

Analysis By: Amresh Nandan

Benefit Rating: Transformational

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Open APIs refer to application programming interfaces (APIs) that are publicly available, and developers can use them for programmatic access to proprietary software. Such APIs are often developed by a commercial entity, industry body or industry standardization/specification organizations for the purpose of ease of integration, interoperability and utilization of data and functionalities by the wider community.

Why This Is Important

Telcos/communications service providers (CSPs) are increasingly focusing on open APIs for better modularity, faster integration and reduced vendor lock-in. Integration requirements among several solutions to move toward platforms and ecosystems have forced detailed specification and standardization by industry bodies, such as 3GPP, ETSI, TM Forum and MEF. Open API adoption in the telecom industry is on the rise among leading CSPs and vendors, and 2020 witnessed further increased interest.

Business Impact

- There is strong value in adopting open APIs, for faster, easier and cost-effective integration and interoperability.
- Open APIs can have a far-reaching, positive business impact; however, success depends on the level of adoption.
- TM Forum reports greater adoption of their Open APIs by CSPs as compared to vendors and highlights it as an opportunity for vendors. With increase in adoption, value will increase. However, CSPs must push for it instead of waiting for vendors to adopt such APIs.

Drivers

- One of the key reasons for end-user (CSPs) adoption of open APIs is the desire to achieve greater freedom in sourcing and integrating technologies.
- In addition, such APIs by industry bodies are often specified/developed in a collaborative manner, leading to early identification of nuanced requirements and integration challenges.
- CSPs source technologies from multiple vendors and integration, through open APIs, is the most appropriate way for faster and cost-effective interoperability.
- As CSPs look toward developing ecosystems, open APIs are a necessity to expose data and functionalities, as well as to consumer partner capabilities.
- Many CSPs have started tracking the level of TM Forum Open APIs adoption by the vendor community with a view to evaluate vendors and to formalize their own approach.

Obstacles

- Continuation of old solutions and architecture in many CSPs is an obstacle for open APIs implementation, though some CSPs have been working on overlay mechanisms.
- Many vendors have also shown reluctance in implementing open APIs to address the threat of substitution and to continue to offer complex integration services.
- Some vendors have developed their own set of APIs and project them as analogous to open APIs. Such an approach is detrimental to open APIs adoption.

User Recommendations

- Consider using open APIs during platforms/system design for better technology change, to gain greater freedom in sourcing and integration.
- Encourage technology vendors to adopt open APIs by including the topic in RFIs/RFPs and the vendor evaluation process.
- Dedicate resources for participation in API development initiatives, trials and pilot programs as effectiveness of such APIs comes through active participation in the collaborative development process.
- Examples of such initiatives include TM Forum's Open APIs Catalyst programs (associated with telecom BSS, OSS and related areas), where many CSPs and vendors participate to address specific challenges/problems. This allows faster and collaborative development and integration while maintaining a degree of independence in technology change.

Gartner Recommended Reading

Market Guide for CSP Business Support System Solutions

Market Guide for CSP Operations Support System Solutions

Market Trends: Fundamental Changes Await the Telecom BSS Market

Market Trends: Telecom BSS Evolving as a Set of Federated Capabilities

API Threat Protection

Analysis By: Mark O'Neill, Jeremy D'Hoinne

Benefit Rating: High

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

API threat protection is the defense of web APIs from exploits, abuse, access violations and denial of service (DoS). It is required both for external and internal APIs. API gateways, web application and API protection (WAAP), and specialist API security tools provide API threat protection through a combination of content inspection of API parameters and payloads, traffic management, and traffic analysis for anomaly detection.

Why This Is Important

Every connected mobile, modern web or cloud-hosted application uses and exposes APIs. These APIs are used to access data and to call application functionality. APIs are easy to expose but difficult to defend. This creates a large and growing attack surface, leading to a growing number of publicized API attacks and breaches. Traditional network and web protection tools do not protect against all the security threats facing APIs, including many of those described in the OWASP API Security Top 10.

Business Impact

Because APIs are typically used for access to data or application functionality, often linked to systems of record, the impact of an API breach can be substantial. Privacy regulations typically require reporting if private data is breached through an insecure API. APIs are easily and intentionally programmable, so a vulnerability can leak large volumes of data. That it can be challenging to separate valid API use from nefarious access raises the risk of blocking valid use.

Drivers

- APIs account for increasing amounts of Internet traffic. Analysis by Akamai, a content delivery network provider, indicates that API requests make up 83% of all hits, with API hits growing by 30% year-over-year and expected to reach 42 trillion hits by 2024 (see The State of the Internet).
- The OWASP API Security Top Ten has driven awareness of the requirement for API threat protection.
- A growing number of API breaches are occurring. For a nonexhaustive list, see API Security: What You Need to Do to Protect Your APIs.
- Transactional APIs, such as those used in financial services or ad tracking, drive the need for API threat protection because they tie directly into business capabilities.
- Vendors focused on API threat protection are beginning to gain market traction.
- Some established vendors use machine learning to detect potentially harmful API usage patterns and thus protect APIs from threats. They include Akamai, Ping Identity (with the PingIntelligence for APIs security product), as well as bot mitigation vendors such as PerimeterX and Distil Networks (acquired by Imperva). Other vendors, such as CriticalBlue and Data Theorem, have linked mobile security with API threat protection.

Obstacles

- Many organizations lack visibility of their APIs, as many APIs are used as part of web or mobile applications and not published directly. This means that a key requirement of API threat protection is API discovery, since, as every security professional knows, you can't secure what you don't know.
- Organizational ownership of API threat protection is a challenge. Whereas the security team, under a CISO, typically manages a web application firewall (WAF), API gateways are managed by API platform teams. This can lead to API threat protection being neglected.
- To date, most API threat protection has focused on traffic throttling, payload risk analysis (e.g., looking for private data in requests and responses), and/or validation of API traffic against Swagger/OpenAPI Specification (OAS) API definitions. This basic approach is taken by many API mediation products, particularly API gateways, but also web application and API protection (WAAP) products.

User Recommendations

- Discover and categorize your APIs as the first step in API threat protection.
- Implement a layered API security model. At the outer (typically cloud-based) layer, use volumetric distributed denial of service (DDoS) protection and bot detection. Behind this layer, apply API-specific authentication, authorization and traffic management.
- Assess the API protection provided by your current WAF or API gateway using Critical Capabilities for Cloud Web Application and API Protection and Critical Capabilities for Full Life Cycle API Management. If they provide immature or insufficient API protection, investigate API threat protection specialists such as 42Crunch, Imvision, Cequence Security, Neosec and Salt.
- API protection rules should be adaptable, based on the nature of the API itself. Static rate limits or IP allow/block lists are rarely useful in production environments or at scale.

Sample Vendors

42Crunch; Akamai; APIsec; Cequence Security; CriticalBlue; Imperva (CloudVector); Imvision; Neosec; Noname Security; Salt

Gartner Recommended Reading

API Security: What You Need to Do to Protect Your APIs

How to Implement API Discovery and Detection to Improve API Management and Security

Solution Path for Forming an API Security Strategy

Gartner Peer Connect Perspectives: How to Address API Security?

Sliding into the Trough

API Standards

Analysis By: Mark O'Neill

Benefit Rating: Moderate

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

Application programming interface (API) standards enable common understanding or interoperability between API providers and consumers. There are multiple types of API standards. They include guidelines for how APIs, including event-driven APIs, are described and published. In certain industries such as healthcare and banking, industry vertical API standards are driven, typically indirectly, by government mandates. In many industries, however, API standards are currently absent or nascent.

Why This Is Important

API standards allow API providers and API consumers to communicate more effectively. Within an organization, the use of API standards may take the form of using OpenAPI Specification (OAS) combined with an API style guide for consistency. Within industries such as banking in particular, de facto API standards such as Banking Industry Architecture Network (BIAN) and Financial Data Exchange (FDX) provide API definitions covering common use cases.

Business Impact

API standards have a particular benefit in removing the development and support burden for businesses associated with proprietary API approaches. It also enables an ecosystem of tools implementing API standards, including API design and API testing.

Drivers

- Internal (or private) standards inform API design to create consistency and increase overall API design quality.

- Government mandates are driving the use of API standards in healthcare, e.g., U.S. Office of the National Coordinator for Health Information Technology (ONC) regulations, and banking, e.g., the revised Payment Services Directive (PSD 2) in the European Union and other open banking regulations worldwide.
- API management vendors are adding support for API standards such as AsyncAPI for event-driven APIs.
- Identity standards such as OAuth 2.0 and OpenID Connect, as well as FAPI (Financial-grade API) provide standards used in identity and access control for APIs.

Obstacles

- Many organizations naturally define APIs for their own specific scenarios. This means that, even within the same industry and for the same broad use cases, APIs differ from each other. Even in industries where certain API standards are mandated, the definitions of actual published APIs may differ. This presents an obstacle to industry API standards, as well as to interoperability. However, it is not an obstacle to innovation in the API economy, since organizations will create APIs that map to their business requirements. Indeed, an enterprise may opt to create its own internal API standard and check conformance to it using a style guide.
- Vendor support for industry vertical API standards is still growing.
- Use of identity standards such as OAuth 2.0 and OpenID Connect is inherently complex.

User Recommendations

- Standardize on Open API Specification (OAS) for REST APIs, for GraphQL use schema definitions and for event-based APIs use AsyncAPI.
- Use an API style guide for API quality and consistency.
- Check the progress of industry API standards, but don't let the lack of standards limit your public API initiatives.
- Ensure that your API program uses identity and security standards, including OAuth 2.0 in particular, as they aid developer adoption and eliminate the need to reinvent existing technologies.
- Ensure that your chosen API platforms support API definition standards such as OAS, API security standards such as OAuth 2.0 and OpenID Connect.

Sample Vendors

Postman; SmartBear; Stoplight

Gartner Recommended Reading

Prepare for CMS Interoperability and Patient Access API Compliance for U.S. Healthcare Payers

Beyond Compliance: Using Banking API Standards for Competitive Advantage

Modern Identity and APIs: OpenID Connect, OAuth 2.0 and SCIM 2.0

API Testing Services

Analysis By: Jaideep Thyagarajan

Benefit Rating: High

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Definition:

API testing services refers to a set of testing activities that is performed directly and as part of integration testing to validate if the programming interfaces meet expectations for quality, functionality, reliability, performance and security. Focusing mainly on the business logic layer of the software architecture, API testing involves using software to send calls to the API, receive output and validate the system's response.

Why This Is Important

Defective APIs directly impact digital experiences, such as mobile apps and reactive web applications, meaning that API quality is critical. Poor-quality APIs impact developer experience, which is increasingly important as organizations use APIs internally and build external developer communities. Therefore, API testing services are an important consideration to mitigate business risk and to ensure seamless integration of applications with other services.

Business Impact

- Organizationwide API strategies benefit from API testing services to ensure quality and consistency across the entire API portfolio.
- APIs that fail to deliver the expected level of quality, security, reliability and performance can thus have tremendous business impacts, both to the organization producing it and to those consuming it.
- Risks associated with application failure have broader business impacts; hence, the quality of the APIs produced and consumed is now more important than ever.

Drivers

- Many organizations have created API platform teams composed of API center of excellence (COE) team members with a mandate to deliver high-quality, reliable APIs
- Organizations increasingly rely on APIs as part of their daily operations, such as logistics APIs for retail deliveries, and APIs into systems of record. These APIs must be reliable and well-designed.
- APIs are treated more like products than code. They are designed for consumption for specific audiences (e.g., mobile developers); they are documented; and they are versioned in a way that users can have certain expectations of their maintenance and life cycle. Because they are much more standardized, they have a much stronger discipline for security and governance, as well as being monitored and managed for performance and scale, thereby amplifying the importance of testing them.
- API testing services offer an opportunity to test the core functionality of the app without having to interact with a potentially disparate system. This helps in early bug detection, instead of them becoming larger issues during GUI testing, and thereby protects the application from malicious code and breakage.
- APIs have become a primary attack surface for many systems. These attacks have resulted in an endless stream of data breaches and other security incidents, yielding significant damage to organizations and individuals. As a consequence, SPVM teams — along with the business leaders whose applications APIs support — express significantly increased interest in API testing and security.
- Traditional testing skills and team culture do not apply naturally to APIs, because APIs are used by other systems — not directly by end users. API testing calls for a different set of skills that involves the ability to understand the business logic of the service in addition to scripting and coding skills. Therefore, seeking specialized API testing services is becoming a matter or priority for many SPVM teams.

Obstacles

- APIs present challenges like broader attack surface area, higher potential for unexpected misuse and unpredictable demand, among others. These challenges translate to specific testing ramifications that require strong consulting skills to factor in all the API test scenarios, which some providers may struggle with.
- API testing includes tasks like preparing the environment on which the API will exist, updating the API testing requests scheme, deciding on the sequence of API calls and validating parameters, among others. These activities often call for a strong involvement of client-side resources as well.
- Incumbent vendors offering traditional testing services may not have strong API testing skills. So the selection criteria needs to factor in API-testing-specific key capabilities, like the ability to understand basic rules of creating good APIs, to design tests that are relevant for APIs, to use API testing tools, to understand business logic of service and to locate real user scenarios, among others.

User Recommendations

- Begin evaluation and selection efforts by assessing the overall role APIs play in your application portfolio, their criticality to the organization and the security and business risk — and technical requirements — they pose.
- Give preference to API testing services that are underpinned with intelligent test creation and automated validation, as testing a broad range of conditions and corner cases is critical with APIs, so automation clearly comes to the forefront.
- Ramp up the scope for extensive performance testing as part of services. Considering the highly exposed nature of APIs, there is a strong potential for unpredictable and volatile traffic volumes. So it is critical to determine whether your APIs satisfy SLAs in the event of surging demand.
- Examine the testing capabilities provided by existing tools in your application testing portfolio, including full-life-cycle API management platforms, which may include API testing. Tools are a key component of API testing services.

Sample Vendors

Accenture; APImetrics; Applause; Cigniti; IBM; Postman; SmartBear; Stoplight; TCS; Tricentis

Gartner Recommended Reading

The Evolving Role of the API Product Manager in Digital Product Management

Healthcare Payer APIs

Analysis By: Mandi Bishop

Benefit Rating: Transformational

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Definition:

APIs enable real-time interoperability among internal and external ecosystems, and are a hallmark of composable architecture. APIs like Fast Healthcare Interoperability Resources (FHIR) provide timeliness and agility in clinical and administrative systems, and process integration.

Why This Is Important

APIs are on the leading edge of healthcare's digital transformation. Recent interoperability rules from the U.S. Office of the National Coordinator for Health IT (ONC) and the U.S. Centers for Medicare and Medicaid Services (CMS) make APIs critical to the payer industry. Mobile apps, modern web architectures, digital strategies, IoT and the ubiquity of web APIs provided by cloud service providers and leading enterprises have made APIs a must-have component of any integration architecture.

Business Impact

APIs are extending the value and reach of expensive and critical business systems beyond the limits of their individual capabilities into integrated, comprehensive solutions for more complex payer workflows and problems. They provide the timeliness and agility required for organizations to transform into a composable enterprise supported by a real-time health ecosystem, and usher in a new era of consumer-mediated data exchange.

Drivers

- Purchasers, providers and partners alike now assume that API management, and complementary consent, identity and access management capabilities are core organizational competencies driven largely by new interoperability-focused legislation and compliance requirements.
- Effective APIs are consumed widely across the health ecosystem and IT will need to support, secure, monitor and manage these APIs through their life cycles. API management features are very closely related to integration activities — the act of consuming an API is an integration, and the act of exposing an API often involves integration with a back-end resource of some kind. They have become the integral connecting fabric among digital platforms and their associated ecosystems.
- While APIs and service-oriented principles have been around for some time, they are rapidly becoming pervasive and are thereby central to payers' consumer value proposition. As a key example, the CMS Interoperability and Patient Access Final Rule requires that payers share members' health data via standard FHIR APIs and allow members to grant permission to third-party software applications to access their data as well.
- Payers are beginning to demand APIs from their vendor community that can be safely consumed and orchestrated, to form new apps, workflows, business and operating models.
- Gartner expects a dramatic acceleration in payer adoption of this capability in 2021 as the pace of new business requirements increases. Significant challenges in effectively implementing and adopting API management are also anticipated. Thus, this technology has been advanced to the midpoint of the Peak and Trough of Disillusionment and it is anticipated that it will achieve mainstream adoption within five years.

Obstacles

- Although regulations are driving API adoption acceleration, there is no centralized regulatory or industry body for approving and registering third parties or monitoring open API use. This places the full burden of API policy management on payers. It will result in fragmented experience for partners seeking connection, which will slow ecosystemwide data and workflow integration efforts.
- Data integrity issues and high-profile security breaches will exacerbate existing consumer trust challenges, limiting in the short term the expansion of APIs beyond compliance use cases.
- Legacy architecture does not natively support APIs, so payers will have to modernize in order to optimize the business value.

User Recommendations

- Prioritize your acquisition of API management capabilities based on your organization's maturity with API technologies, use-case scenarios, organizational requirements and business value implications unique to your project or initiative.
- Implement an API program to streamline the delivery of new business capabilities, expand business channels, extend existing applications, enable multichannel clients and facilitate integration.
- Leverage API management technologies to help your teams design, build, consume, operate, secure and manage your APIs.
- Source your API management capabilities from your interfacing and integration platform vendor.
- Use APIs when conventional industry interoperability standards fall short of health information and workflow needs.
- Use API management platforms in conjunction with consent, identity and access management technologies to centralize authentication and authorization for your APIs.

Sample Vendors

Axway; Boomi; Google (Apigee); IBM; Microsoft; MuleSoft; PilotFish; TIBCO Software; WSO2

Gartner Recommended Reading

U.S. Healthcare Administration's Future Requires a Real-Time Payment Ecosystem Powering Value-Based Care

Prepare for CMS Interoperability and Patient Access API Compliance for U.S. Healthcare Payers

6 Critical Technologies to Advance Healthcare Ecosystem Orchestration Ability

API-Based Digital Commerce

Analysis By: Mike Lowndes

Benefit Rating: Transformational

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

API-based digital commerce (also commonly known in the market as "headless" digital commerce) is the use of APIs to decouple front ends from core commerce services, and to integrate commerce capabilities within any touchpoint where selling is required.

Why This Is Important

The proliferation of touchpoints requires a multichannel, multiexperience approach to applications. This requires the decoupling of the presentation from commerce services that an API-based approach offers. Some vendors provide pure-play, API-based commerce platforms, while others retain a native storefront but also provide full APIs for headless operation, known as "head optional." It is the enabling step toward composable commerce.

Business Impact

Decoupling the front end is a step toward a modern, modular commerce architecture that provides business flexibility and IT agility. As commerce journeys become multiexperience, this is an enabler for delivering consistent experiences across touchpoints. The storefront and other touchpoints must be delivered independently of the commerce application (via a DXP or a custom front end). This provides ultimate flexibility for delivering unique experiences, but requires digital maturity to succeed.

Drivers

API-based digital commerce is being rapidly adopted by midsize to large, digitally mature organizations, and is becoming the standard approach for delivery of experiences by the enterprise, even if the underlying application remains monolithic.

Such adoption is driven by:

- Recognition of the quality of digital experience as a key differentiator across multiple touchpoints (for example, native mobile apps, marketplaces, social platforms, in-store experiences, the Internet of Things, wearables, smart homes and vehicles).
- Maturity of front-end frameworks, and the emergence of the single-page application (SPA) and progressive web application (PWA) as the dominant “next generation” of client-side, JavaScript-based presentation.
- Growth in digital experience platforms (DXPs) in supporting “experience-driven commerce.”
- Commerce as an enabling part of a wider digital business technology platform.
- Pace of innovation in digital commerce driving more flexible, modular architectures.
- Expense and complexity of some leading “monolithic” commerce platforms, when a more agile, flexible subset of capability is desired.

Obstacles

API-based digital commerce improves business agility; however:

- The “headless” buzz has caused some disillusionment among those expecting it to be easy to achieve. Despite this, it is rapidly moving toward mainstream acceptance.
- Commerce experiences can be more complex to implement than single-vendor “full stack” solutions due to the increased emphasis on integration efforts and governance of the decoupled front-end technology.
- A key challenge is ensuring business users retain no-code control over the storefront(s). This adds complexity to implementations and the business UIs required to support presentation and commerce processes.

- Most vendors' native commerce platform storefronts are shifting from server-side "themes" or template engines toward client-side SPA/PWA. Thus, some of the benefits of API-based are becoming available from almost all vendors. However, having an API is not the same as being built "API first," and not all the benefits of this new approach are realized.

User Recommendations

API-based commerce may fit your requirements if you:

- Want to retain granular control over multiexperiences, including by deploying a DXP, building via an MXDP, or developing an SPA/PWA presentation tier.
- Are looking to support multiple digital and physical channels equally from the same business logic, and support cross-channel continuity of experience.
- Already have (or are looking to implement) a DXP to provide a more consistent customer experience across commerce, brand and other digital properties.
- Have a large, inflexible, legacy, monolithic, full-stack commerce application that cannot be replaced in a single step, and want to migrate to a modular architecture.
- Have a unique commerce business model that full-stack vendors cannot support without considerable front-end customization.
- Need commerce integrations to support wider digital business strategies.

Sample Vendors

commercetools; Elastic Path; Fabric; Kibo; Spryker; VTEX

Gartner Recommended Reading

Magic Quadrant for Digital Commerce

Choose the Right Digital Commerce Platform Architecture

Harness the Core Capabilities of a Digital Commerce Platform

Industry Vision: Commerce to You

API Access Control

Analysis By: Henrique Teixeira, Michael Kelley

Benefit Rating: Moderate

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

API access control is a big part of the API security capabilities that enable authentication and authorization of API targets. At minimum, it involves an OAuth 2.0 authorization server, which supports and implements consent, handles scope to claim mappings, and is capable of issuing customizable and self-contained JSON Web Tokens (JWTs) to web servers, mobile apps, modern web apps and services used for accessing API targets.

Why This Is Important

API access control has grown in importance. Over the past year, there was a 300% increase in IAM discussions related to APIs. Organizations have evolved their application architectures toward mobile apps, API clients and back-end services, all of which require more sophisticated authentication and authorization controls to APIs and microservices.

Business Impact

API access control capabilities are important in the following scenarios:

- Developer use cases, offering libraries, out-of-the-box integrations, as well as best practices examples for access controls in APIs.
- Enabling user-present and machine-to-machine access to API targets.
- Workforce and customer IAM use cases, including customer data protection, consent management and data privacy strategies in modern applications.
- Architecture standardization with central access controls for APIs.

Drivers

The market has been maturing and access management and API security vendors have added more robust capabilities for API access controls. The key trends and developments driving hype around this innovation are:

- Beyond OAuth 2.0 authorization servers, other functionalities are starting to become more popular in API scenarios, like libraries, sidecars or out-of-the-box integrations with APIs and API mediators to centralize authentication and authorization decisions to the authorization server.
- Developer self-service capabilities are also increasing in demand, for managing apps and services and support for flows such as: OAuth mTLS, JWT and SAML grant types, the device flow, token exchange, introspection and revocation. They also provide strong and agile cryptographic mechanisms for signing tokens.
- Modern applications and service patterns (like service mesh) are extensively adopting JWTs as an authentication mechanism. JWTs are becoming the de facto standard to convey authorization information and claims about users and services.
- The lack of advanced IAM capabilities of API gateways (like session management, advanced adaptive access and support for evolving identity standards) has led to an increased demand for specialized access control features for APIs, either delivered by AM and other IAM and API security specialist tools.
- API access control adoption has benefited from the popularity of OAuth 2.0 as an industry standard as it is widely adopted by the vast majority of AM and API security vendors.
- Customer IAM is another big driver for more recent API access control demand when developing more sophisticated privacy and consent management controls based on those standards.
- API security now benefits from increased awareness and product feature coverage.

Obstacles

API access control is increasing in importance and should be a mandatory piece of API security strategies. However there are still obstacles to mainstream adoption, including:

- Lack of application development alignment with IAM strategies. Time to market is seen as more important than basic API protection in those situations.
- Lack of developer and security staff training in IAM best practices.
- Proprietary methods for API access control are still in use (like custom code using shared databases).

- Belief that API gateways are enough for enabling and securely running API-based applications.
- Lack of tooling that enables scaling access controls to thousands of APIs and services.

User Recommendations

- Use API access control (to control which applications and people can access APIs) alongside API threat protection (to detect and block attacks on APIs).
- Approach API-based app development with both API threat protection services (such as WAFs) and DDOS protection as well as advanced API access control capabilities.
- InvestigateAM tools that provide converged IAM features for APIs, and at least some "lightweight" API gateway capabilities such as token validation services and centralized authorization, for example.
- Phase out proprietary methods for API access control.
- Create and implement an effective API security strategy that looks beyond just API threat protection and that also includes API access controls.
- Provide IAM training for developer and security staff, highlighting API access control.
- Deploy API access control enforcement points as close as possible to the service being protected. This will enable defense in depth.

Sample Vendors

Amazon Web Services; Auth0; Authlete; Cloudentity; Curity; ForgeRock; IBM; Microsoft; Okta; Ping Identity

Gartner Recommended Reading

Critical Capabilities for Access Management

API Security: What You Need to Do to Protect Your APIs

Critical Capabilities for Full Life Cycle API Management

Modern Identity and APIs: OpenID Connect, OAuth 2.0 and SCIM 2.0

Building Authentication, Authorization and SSO Into API-Driven Apps G00390369

API Marketplaces

Analysis By: Mark O'Neill

Benefit Rating: Moderate

Market Penetration: 5% to 20% of target audience

Maturity: Emerging

Definition:

An API marketplace is a platform for API providers to publish and market APIs. They range from API directories to API portals provided by a single API provider to commercial marketplaces. Consumers, mainly developers, use API marketplaces to discover APIs and (in some cases) purchase access to APIs. Although public API marketplaces are better-known, a growing number of organizations are deploying internal or private API marketplaces.

Why This Is Important

API marketplaces empower organizations to share APIs. External API marketplaces allow organizations to share APIs with a community of developers, including facilitating an ecosystem by enabling partners to implement solutions using their APIs. Internal API marketplaces address a different use case, which is to help developers discover and share APIs between teams.

Business Impact

For API providers, registering APIs in API marketplaces can increase developer visibility and consumer mind share, to drive API usage and, by extension, business impact. The API marketplace provider may take a share of the revenue for APIs sourced through the marketplace, but this can be considered a cost of sale. An API marketplace can also facilitate ecosystem creation and is a critical enabler of composable business.

Drivers

- Use of APIs is growing; according to the infrastructure and security vendor CloudFlare, in 2020, API traffic grew 300% faster than web traffic, reaching 50% of HTTP traffic. This demonstrates the need for API marketplaces to discover APIs from the large amount of APIs available.
- The number of APIs within an organization is also climbing, driving the need for developers to find which APIs and services are available.
- Composable business, including composable commerce, relies on the use of API marketplaces to share APIs and packaged business capabilities.
- Increased use of low-code platforms, iPaaS, RPA and analytics tooling enables more citizen development, using APIs which may be sourced from API marketplaces.
- New open-source platforms such as Backstage, from Spotify, are driving the creation of internal API marketplaces as part of larger developer hubs.

Obstacles

- Public API marketplaces which provide a public directory of APIs have generally had disappointing results, because developers are more likely to go to API providers directly to sign up for APIs. This has resulted in API marketplaces approaching the Trough of Disillusionment. However, internal API marketplaces have had more success, since they enable developers to share APIs across multiple teams.
- API portals provided as part of API management platforms are typically basic in nature, resulting in significant customization work to create an API marketplaces based on such an API portal.

User Recommendations

API providers:

- Manage senior business stakeholders' expectations by ensuring they are aware that outcomes from placing APIs in public API marketplaces are often disappointing.
- Examine billing terms to understand what goes to the marketplace provider when considering commercial API marketplaces. Since your APIs may be side-by-side with competing APIs, think carefully about differentiation.

- Establish a commercial model upfront (e.g., through registration fees and/or revenue share) and a clear governance process for onboarding third-party APIs if you plan to build your own API marketplace.

API consumers:

- Ensure that you use APIs from trusted marketplaces and trusted API providers, examining usage agreements, licensing and billing terms carefully. In general, ensure that you are governing your organization's usage of third-party APIs.
- Investigate if subscribing to an API directly from the API provider offers better pricing or usage terms than consuming the API through a marketplace.

Sample Vendors

Achieve Internet; Backstage; Constellant; Cortex; Effx; ProgrammableWeb (Salesforce); Pronovix; RapidAPI; Roadie

Gartner Recommended Reading

How to Derive Value From APIs Using API Marketplaces

Create API Portals That Drive API Adoption Among Internal and External Developer Communities

Choose the Right API Monetization and Pricing Model

To Create a Successful API-Based Ecosystem, Look Before You Leap

API Management PaaS

Analysis By: Mark O'Neill

Benefit Rating: Moderate

Market Penetration: 5% to 20% of target audience

Maturity: Adolescent

Definition:

API management PaaS (APIM PaaS) takes an on-demand approach to the delivery of API management. It provides an alternative to the purchase and installation of stand-alone, full life cycle API management software. APIM PaaS manages API access via provider-hosted API gateway services, with the option of on-premises API gateways, as well as providing an API developer portal. It is typically designed to be used with other PaaS services such as function PaaS (fPaaS) and integration PaaS (iPaaS).

Why This Is Important

APIs are increasingly built on cloud platforms and using platform as a service (PaaS), so it is natural for API management to also be delivered as-a-service. APIM PaaS takes full advantage of cloud benefits, such as autoscaling, resiliency and robust security. It also allows some vendors to offer per-API-call pricing. APIM PaaS may include the ability to deploy on-premises API gateways, to enable hybrid API management architecture with APIs on-premises and cloud-based API management.

Business Impact

APIM PaaS allows costs to scale with the business value of APIs, reducing the impact of a large outlay as an API program scales up. It enables APIs to be managed effectively when API traffic is unpredictable and potentially very large. APIM PaaS also brings business benefits when an APIM PaaS offering is provided as part of the PaaS platforms already in use by an organization, through unified procurement and billing.

Drivers

- APIM PaaS is driven by migration to and adoption of cloud platforms.
- Function PaaS (fPaaS) can act as a major driver for APIM PaaS. This is because fPaaS offerings can make use of API management on their associated cloud platforms. In some cases, they can automatically populate API gateways with endpoints so that fPaaS functions can be called via REST APIs.
- iPaaS and aPaaS are also drivers toward the need for API management provided by PaaS platforms.
- Since many organizations are building APIs in the cloud, APIM PaaS is also increasingly used in hybrid scenarios and multicloud scenarios.
- Automation is also a driver for APIM PaaS. This is because APIM PaaS also includes APIs into the API management platform itself. These are used to automate the creation and management of APIs, often as part of a DevOps pipeline, as well as for customizing the developer experience (DX) provided by an API developer portal.

Obstacles

- Perceptions of network latency can impact on the uptake of APIM PaaS for managing on-premises APIs.
- Data residency concerns, such storage of API payloads that may contain private information, are also an obstacle to the uptake of APIM PaaS for managing on-premises APIs.
- APIM PaaS can result in higher-than-expected pricing as API traffic grows.
- Architecting a hybrid or multicloud API PaaS architecture is nontrivial (see Comparing Architectures for Hybrid and Multicloud API Management).
- APIM PaaS solutions from cloud hyperscalers are generally tied to their larger PaaS platforms, and are not portable for use on other PaaS platforms.

User Recommendations

- Apply API mediation and prioritize the use of APIM PaaS to provide a cost-effective means of providing API management, even when your APIs are on-premises.
- Compare the pricing of APIM PaaS vendors, since not all provide consumption-based pricing (see How Are API Management Platforms Priced?).
- Include API PaaS as part of your API strategy, since it can accelerate time to market for mission-critical digital initiatives.

Sample Vendors

Alibaba Cloud; Amazon Web Services; Google (Apigee); IBM; Microsoft Azure; Oracle; VMware

Gartner Recommended Reading

Magic Quadrant for Full Life Cycle API Management

Critical Capabilities for Full Life Cycle API Management

Ensure Your API Management Solution Supports Modern API Trends Such as Microservices and Multicloud

Toolkit: RFP Template for API Management Platforms

Comparing Architectures for Hybrid and Multicloud API Management

Digital Business Technology Platform

Analysis By: Bill Swanton

Benefit Rating: Transformational

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Definition:

A digital business technology platform (DBTP) integrates and orchestrates new and existing platforms for IT, customer engagement, data and analytics, ecosystem partners and Internet of Things. It senses business events, decides what to do and implements a response that creates value for those involved. Platforms share data, algorithms and transactions with business ecosystems to match, create and exchange services.

Why This Is Important

A DBTP enables enterprises to build a digital business and deliver digital products and services to customers and partners. Without it, enterprises will be unable to gain the business benefits of digital business. DBTPs empower people, businesses and things to give, take or multiply value creation for the enterprise.

Business Impact

DBTPs make it easier for new market entrants, startups, competitors and smart machines to create and pursue new business opportunities. Leveraging DBTPs, organizations can rapidly respond to core business disruptions, such as revamping supply chains disrupted by the COVID-19 pandemic. DBTPs also enable platform business models, which can create rapid market growth and, potentially, dominate industries.

Drivers

- Competition is shifting to digital delivery of value added services, in addition to the traditional products and services. So, traditional businesses need to build a DBTP to compete and/or participate in new digital markets.
- Regulatory requirements in some regions require organizations to share business services through digital platforms. For example, PSD2 requires banks in the EU to provide mandatory access to customer accounts for regulated third parties, a trend now being followed in many other regions.
- Service providers help almost all initial DBTP developments by providing skills, training and reusable assets inevitably sold in conjunction with significant services.

Obstacles

- Managing an inherently hybrid IT infrastructure for the platform and existing applications is a major challenge.
- There is currently no specific market or vendor for a base platform suitable for building digital use cases and data assets. Companies need to assemble components and tools from generally available cloud frameworks, a cluttered market of Internet of Things vendors, public and private APIs and other IT assets.
- While digital-native organizations are adept at these technologies, traditional companies often struggle with new architectural approaches required for large-scale implementations, such as microservices architecture, event-driven architecture and programmable infrastructure.
- A skills learning program is critical as most organizations do not yet have the skills to implement and manage this technology. So skills transfer and culture change need to be a part of any service provider contract.

User Recommendations

- Work with business leaders to identify use cases for your digital business.
- Build out the DBTP as needed to implement the initial digital use cases. The process will take years and may require refactoring as the business scales and the technologies mature. Treat the platform as a continuously evolving product guided through its long life cycle by a product manager.
- Work with technology and service providers to determine what technologies are needed to implement the use case. Most organizations do not yet have the skills to implement this technology so skills transfer needs to be included.
- Ascertain what APIs you might need to consume or provide to interact with customers and/or ecosystem partners inside or outside of the enterprise.
- Keep existing platforms loosely coupled by using techniques such as API mediation so you can modernize those platforms without disrupting your digital business build-out.

Sample Vendors

Amazon Web Services (AWS); Google; Microsoft; NXN; Red Hat; Vantiq; VMware

Gartner Recommended Reading

Use Gartner's Digital Business Layers to Communicate Your Digital Intent

How to Build a Digital Business Technology Platform

Building a Digital Business Technology Platform Requires Clear Goals and a New Team With Cloud Skills

Building a Digital Business Technology Platform Requires New Technology and Service Provider Support

API Developer Portals in Banking

Analysis By: Don Free

Benefit Rating: Moderate

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Definition:

API developer portals in banking provide the capacity to document, configure and manage usage of APIs for application-to-application communications.

Why This Is Important

- API developer portals provide an advanced approach to connect and manage APIs of partner solutions embedded within bank business processes. They're also used to maintain internal APIs.
- Accelerating internal developer portal usage and adoption is crucial to banks. These portals promote awareness and centralize documentation, which increases ease of use for internal developers and third parties. API developer portals are foundational for banks with open banking aspirations.

Business Impact

APIs play a key role in boosting banks' digital business initiatives, and API developer portals promote and optimize API consumption and performance through both functional and technical documentation and tools.

Drivers

- Transparency and self-service access are increasingly important characteristics for successful API development and usability.
- API resources are broadening within banks, and the distributed management of these software assets will intensify complexity, drive duplication and increase TCO; API developer portals help address these issues.
- The API developer portal is a foundational technology and a prerequisite for API marketplaces that support the pursuit of higher-order business models such as banking as a service.
- Internal communication and collaboration are fundamental attributes that promote successful API programs.
- API developer portals may also be adapted for usage by low-code platforms.

Obstacles

- Low internal developer portal adoption rates and decreased trust will promote siloed behaviors and erode innovation.
- Complex navigation and various versions of APIs diminish usage rates among developers.
- Assumptions that internal developer portals are intended only for developers will diminish value.

User Recommendations

- Increase developer portal adoption and credibility by making developers successful such as by building skill sets and providing tools to drive efficiency.
- Drive user adoption through marketing efforts that recognize developers and showcase use cases that contribute to business value.

Sample Vendors

Achieve Internet; Apigee (Google); Axway; IBM (API Connect); Pronovix; Salesforce (MuleSoft)

Gartner Recommended Reading

Create API Portals That Drive API Adoption Among Internal and External Developer Communities

Open Banking Strategy

Analysis By: Alistair Newton

Benefit Rating: Transformational

Market Penetration: 5% to 20% of target audience

Maturity: Early mainstream

Definition:

An open banking strategy is a formal embodiment and integration of open banking principles and technologies into a digital banking and technology strategy. The strategy itself may embrace transformation or simple regulatory compliance, but the formal process and recognition within the bank will enable CIOs to proactively address the technology aspects of open banking initiatives.

Why This Is Important

Many in the banking industry have been primarily focused tactically on open banking technologies — reacting after the event to regulatory requirements or technology advances, rather than proactively integrating technology into the bank's business strategy. Open banking may represent a simple regulatory compliance project or a fundamentally transformative series of initiatives. Without a formal strategy, regulatory compliance will be entirely reactive, while transformation will be unachievable.

Business Impact

An open banking strategy will enable banks to position their ecosystem and API-focused technology investments in the context of the overarching business strategy, which:

- Will help CIOs and senior business executives align open banking strategies with digital banking initiatives, enterprise architectures, infrastructures and operations innovation
- Is essential to create the platforms, marketplaces, app stores, and ecosystems necessary to create new value and revenue for the bank

Drivers

Open banking initiatives have taken two distinct paths: (1) those led by new regulatory regimes that require banks to open their customer data to third-party applications, and (2) those that are more market-led initiatives, with individual or groups of banks focusing on the potential of open banking to drive digital transformation. Thus:

- As governments and national regulators observe new regulations being introduced in other regions, they are following suit and defining new frameworks and standards that promote open banking (such as PSD2 in Europe, the U.K. Open Banking Standard, Society 5.0 in Japan).
- Regulatory-driven open banking is seen by many banks as reducing their power and enabling fintech challengers, so many banks will fail to identify the opportunities and choose to simply comply with regulations.
- Transformative open banking strategies enable banks to consume and deliver functionality from and to third parties in order to expand digital ecosystems. Strategies include business context, business objectives, goals and strategies, strategic choices (business capabilities, people and culture, information and technology, ecosystem), principles, and measures of success, risks and issues.
- Transformative open banking strategies can enable banks to offer developer portals to access APIs to integrate banking products into third-party applications or services. Developer portals are now extremely common among those banks driving innovation through their open banking strategy.
- Transformative open banking can allow banks to create a marketplace that supports both consumer and SMB customers.
- The creation of a well-defined and well-articulated open banking strategy will take two to five years to reach the Plateau of Productivity.

Obstacles

CIOs will struggle to support digital transformation without an open banking strategy:

- Recognizing the impact of disintermediation and regulatory changes, executives are looking to their CIOs for guidance on how to leverage technology to address these challenges.

- In response, IT leaders have begun to explore open banking strategies in partnership with the business, and while progress has accelerated, in many institutions a more expansive view on the innovation opportunities would be beneficial.
- Transformational open banking strategies will challenge many deeply held beliefs and approaches in a bank. These include business capabilities (such as identifying and recruiting business ecosystem partners), people and culture (such as hiring talent with experience building and supporting APIs), information and technology (such as building a digital platform), and business ecosystems (such as turning competitors into customers).

User Recommendations

CIOs should create an executive strategy team composed of risk, compliance, marketing, IT and business leaders responsible to develop an open banking strategy. They should:

- Identify the strategic context: Platform business will reduce banks to invisible, back-end utility service providers unless they change their business strategies. A do-nothing strategy is not an option.
- Define future business objectives, goals and strategies, such as “we will compete by providing higher-quality tools and by sharing data, algorithms and transactions.”
- Create strategic principles, such as “we will empower business ecosystems to create new value for our customers” and “we will expose business services that can be shared internally and externally.”
- Identify measures of success and KPIs such as the conversion rate of customers to business ecosystem partners or percent of SLAs met.
- Identify risks and issues, such as culture risk, and ensure that the strategy is subject to constant review and sense-checking.

Climbing the Slope

Full Life Cycle API Management

Analysis By: Shameen Pillai

Benefit Rating: High

Market Penetration: More than 50% of target audience

Maturity: Early mainstream

Definition:

Full life cycle API management involves the planning, design, implementation, testing, publication, operation, consumption, versioning and retirement of APIs. API management tools enable API ecosystems and publishing APIs that securely operate and collect analytics for monitoring and business value reporting. These capabilities are typically packaged as a combination of developer portal, API gateway, API design, development and testing tools as well as policy management and analytics.

Why This Is Important

APIs are widely used and accepted as the primary choice to connect systems, applications and things to build modern composable software architectures. The use of APIs as digital products monetized directly or indirectly is also on the rise. Advancing digital transformation initiatives across the world have emphasized the need for creation, management, operations and security of APIs and made full life cycle API management an essential foundational capability every organization must have.

Business Impact

Full life cycle API management provides the framework and tools necessary to manage and govern APIs that are foundational elements of multiexperience applications, composable architectures and key enablers of digital transformations. It enables the creation of API products, which may be directly or indirectly monetized, while its security features serve to protect organizations from the business impact of API breaches.

Drivers

- Organizations are facing an explosion of APIs, stemming from the need to connect systems, devices and other businesses. Use of APIs in internal, external, B2B, private and public sharing of data is driving up the need to manage and govern APIs using full life cycle API management.

- APIs that package data, services and insights are increasingly being treated as products that are monetized (directly or indirectly) and enable platform business models. Full life cycle API management provides the tooling to treat APIs as products.
- Digital transformation drives increased use of APIs, which in turn increases the demand for API management.
- APIs provide the foundational elements required for growth acceleration and business resilience.
- Developer mind share for APIs is growing. Newer approaches to event-based APIs, design innovations and modeling approaches such as GraphQL, are driving interest, experimentation and growth in full life cycle API management.
- Cloud adoption and cloud-native architectural approaches to computing (including serverless computing) are increasing the use of APIs in software engineering architectures, especially in the context of microservices, service mesh and serverless.
- Regulated, industry-specific initiatives such as open banking and connected healthcare, along with nonregulated, opportunistic approaches in other industries are increasing the demand for full life cycle API management.

Obstacles

- Lack of commitment to adequate organizational governance processes hinders adoption of full life cycle API management. This can be due to lack of skills or know-how, or due to too much focus on bureaucratic approaches rather than federated and automated governance approaches.
- Lack of strategic focus on business value (quantifiable business growth or operational efficiencies) and too much focus on technical use cases can disengage business users and sponsors. This is particularly apparent in cases where API programs fail to deliver promised return on investment.
- Traditional, single-gateway approaches to API management do not fit well to a modern, distributed application environment.
- Partial or full set of API management capabilities provided by vendors in other markets such as application development, integration platforms, security solutions, B2B offerings, etc., can create confusion and potentially shrink the market opportunities.

User Recommendations

- Use full life cycle API management to power your API strategy that addresses both technical and business requirements for APIs. Select offerings that have the ability to address needs well beyond the first year.
- Treat APIs as products managed by API product managers in a federated API platform team.
- Choose a functionally broad API management solution that supports modern API trends, including microservices, multigateway and multicloud architectures. Ensure that the chosen solution covers the entire API life cycle, not just the runtime or operational aspects.
- Use full life cycle API management to enable governance of all APIs (not just APIs you produce), including third-party (private or public) APIs you consume.
- Question full life cycle API management vendors on their support for automation of API validation and other capabilities, as well as their support for a modern, low-footprint API gateway.

Sample Vendors

Axway; Google; IBM; Kong; Microsoft; MuleSoft; Software AG

Gartner Recommended Reading

Magic Quadrant for Full Life Cycle API Management

The Evolving Role of the API Product Manager in Digital Product Management

How to Use KPIs to Measure the Business Value of APIs

API Security: What You Need to Do to Protect Your APIs

Top 10 Things Software Engineering Leaders Need to Know About APIs

Public APIs

Analysis By: Mark O'Neill

Benefit Rating: High

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Definition:

A public application programming interface (API) is an API that is published for consumption by third-party users and applications. Public APIs typically take the form of lightweight REST APIs, suited to mobile and web consumption. API providers often enable consumers (developers) to use a self-service portal to register and gain access to public APIs. In some cases, a public API may be monetized, for example, through subscription or pay-per-use.

Why This Is Important

Applications are increasingly developed to use public APIs. Many public APIs are available, delivering an extremely wide variety of capabilities, including address verification, mapping, and artificial intelligence (AI). Public APIs are also the foundation of ecosystems, because the delivery of public APIs give organizations an opportunity to create new revenue, open up a new business channel, add customer reach, enhance customer experience, and increase product "stickiness."

Business Impact

The use of public APIs speeds up the software development life cycle by using prebuilt third-party capabilities, to bring applications to market faster. Many SaaS providers offer public APIs for integration and customization, and to establish a platform business model. "API-first" vendors, such as Impala (travel), Plaid (banking), Twilio (communication), and Stripe (payments), have public APIs as their primary products.

Drivers

- Software engineers are increasingly creating applications by composing capabilities sourced as public APIs from outside the organization.
- Organizations are increasingly exposing public APIs as part of creating ecosystems.
- Regulations in industries such as banking (e.g., Directive on Payment Services [PSD 2] in the European Union) and healthcare (e.g., U.S. Office of the National Coordinator for Health Information Technology [ONC] regulations) are driving organizations to provide public APIs.

Obstacles

- Many established businesses have built public APIs with the perspective of "if we build it, they will come." These public APIs without a clear business model have seen poor uptake. This has led some high-profile organizations to abandon their public APIs to concentrate instead of more successful private/B2B APIs within an existing partner ecosystem.
- API security incidents have affected a number of public APIs.

User Recommendations

- Treat public APIs as products. Create the role of API product manager to understand user requirements and communicate a clear roadmap.
- Focus on developer experience (DX) to ensure that your APIs are clearly documented in a developers' portal.
- Ensure that an API has an effective security strategy. Public APIs must be protected from misuse that can lead to data loss, fraud and skewed analytics.
- Use full life cycle API management to manage the API through the entire process of planning, design, runtime deployment and versioning, through to retirement.
- Establish upfront if API pricing will be predictable (e.g., a fixed subscription cost per month) or unpredictable (e.g., directly based on API usage). Use this information to budget the cost of using the public API.
- Ensure that privacy controls are in place when data is being sent to public APIs.
- Examine the API usage agreement of the public API, including versioning policy.

Sample Vendors

api.video; Duffel; Dun & Bradstreet; Google Maps; Impala; Lob; Mux; Plaid; Stripe; Twilio

Gartner Recommended Reading

The Evolving Role of the API Product Manager in Digital Product ManagementChoose the Right API Monetization and Pricing ModelCost-Effectively Scale Your Digital Business by Negotiating API Pricing in Software and SaaS ContractsAPI Security: What You Need to Do to Protect Your APIs

Entering the Plateau

Mediated API Pattern

Analysis By: Abhishek Singh

Benefit Rating: High

Market Penetration: More than 50% of target audience

Maturity: Early mainstream

Definition:

The mediated API pattern is an architectural pattern in which a mediation layer monitors, manages, protects and enriches an API. This layer is typically implemented using one or more proxies or brokers that intercept API traffic and enforce policies and inject capabilities. The layer enables a service to expose one or more abstracted “outer APIs” that support unique user requirements and that map to its native “inner API.”

Why This Is Important

Mediated APIs address the need for more advanced API functionality such as analytics, performance, security and meeting business requirements around customer experience, monetization and support. Mediated API patterns are important in delivering these business requirements by innovating the core applications without disrupting the current APIs.

Business Impact

Mediated APIs deliver many business benefits: improved observability, agility, scalability, performance, resilience, security, privacy and regulatory compliance. They enable organizations to treat APIs as products and to accommodate diverse API consumer requirements. API mediation is a fundamental enabler of the API economy. With mediated APIs, businesses can expect higher rates of innovation, easier scaling and advanced cooperation between the business, IT, customers and partners.

Drivers

- Flexibility — API mediation abstracts an API and ensures loose coupling between consumer and service. This abstraction makes it easier to change or version a service without unduly impacting all consumers.

- Multiple experiences — API mediation enables an API team to expose multiple "experience APIs" that are tailored to the needs of different API consumers, enabling a better developer experience (DX).
- Security — API teams use API mediation to protect and govern access to their APIs — public and private. API mediators enforce authentication and authorization policies, and secure APIs against malware and other intrusions. Mediators can also filter, obfuscate or encrypt message data.
- Traffic management — API mediators can enforce traffic management policies, such as routing, load balancing and throttling.
- Resilience — API mediators often implement resiliency patterns, such as automatic retries, circuit breakers and bulkheads.
- Monitoring — API mediators provide automatic instrumentation of APIs. They can collect operational and business analytics.
- Transformations — API mediators can transform API message content, including adjustments to protocols (such as SOAP to REST), data formats (XML to JSON) and data models. These transformations facilitate integrations between legacy and packaged application APIs.

Obstacles

- Multiple technologies — API mediators come in many forms, such as API gateways, ingress controllers, service meshes, application delivery controllers (ADCs), integration brokers and handcrafted proxies. Each type of mediator can deliver different benefits, and no one solution does everything. But diverse mediators make it more difficult to configure and manage the mediation layer.
- Inadequate governance — All APIs should be monitored and secured via API mediation, but many API teams struggle to track all APIs and ensure that each API is properly mediated.
- Turf battles — Some organizations experience challenges in determining which teams have ownership and responsibility for API mediation. Security teams, network operations and application development all want a piece of the pie.

User Recommendations

- Encourage API teams to use mediated APIs to increase flexibility and agility of all applications that rely on APIs.

- Use mediated APIs to add a host of benefits to both external and internal, and inbound and outbound API interactions: monitoring, scalability, resiliency, security, privacy, compliance, metering, light mediation and orchestration, data filtering, data cleansing, context injection, personalization, and more.
- Use mediated APIs as separate API design from service implementation to enable developers to respond to specific user requests without changing the back-end service.
- Evaluate multiple technologies for implementing an API mediation layer, such as API gateways, service meshes, ingress controllers, ADCs, integration platform as a service (iPaaS), and enterprise service buses.
- Ensure strong governance of the API mediation layer, and that all APIs are mediated. Define processes to discover unmanaged APIs.
- Determine responsibility and authority for configuring API mediators.

Sample Vendors

Axway; F5 (NGINX); Google (Apigee); HashiCorp; IBM; Kong; MuleSoft; Software AG; Solo.io; WSO2

Gartner Recommended Reading

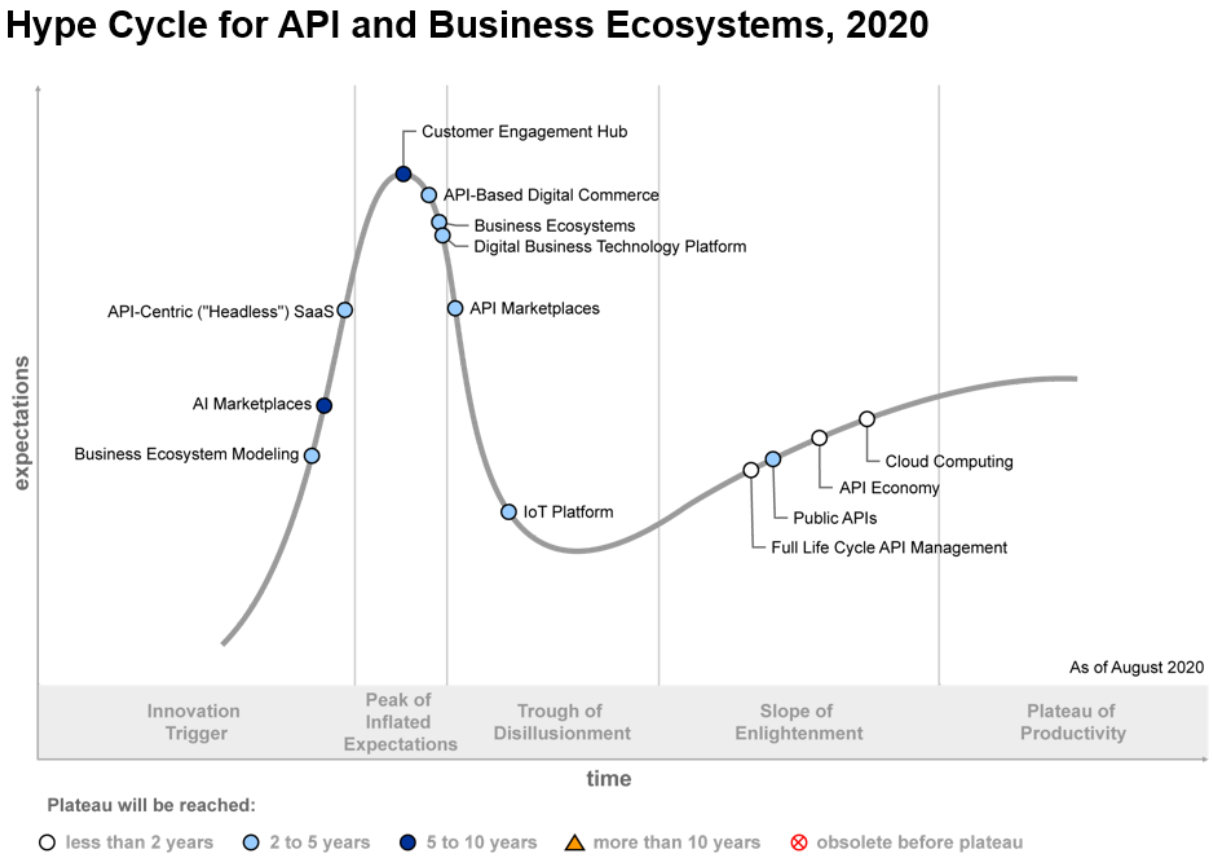
Mediated APIs: An Essential Application Architecture for Digital Business

API Security: What You Need to Do to Protect Your APIs

How a Service Mesh Fits Into Your API Mediation Strategy

Appendixes

Figure 2. Hype Cycle for API and Business Ecosystems, 2020



Source: Gartner
ID: 451430

Gartner

Hype Cycle Phases, Benefit Ratings and Maturity Levels

Table 2: Hype Cycle Phases

(Enlarged table in Appendix)

Phase ↓	Definition ↓
Innovation Trigger	A breakthrough, public demonstration, product launch or other event generates significant media and industry interest.
Peak of Inflated Expectations	During this phase of overenthusiasm and unrealistic projections, a flurry of well-publicized activity by technology leaders results in some successes, but more failures, as the innovation is pushed to its limits. The only enterprises making money are conference organizers and content publishers.
Trough of Disillusionment	Because the innovation does not live up to its overinflated expectations, it rapidly becomes unfashionable. Media interest wanes, except for a few cautionary tales.
Slope of Enlightenment	Focused experimentation and solid hard work by an increasingly diverse range of organizations lead to a true understanding of the innovation's applicability, risks and benefits. Commercial off-the-shelf methodologies and tools ease the development process.
Plateau of Productivity	The real-world benefits of the innovation are demonstrated and accepted. Tools and methodologies are increasingly stable as they enter their second and third generations. Growing numbers of organizations feel comfortable with the reduced level of risk; the rapid growth phase of adoption begins. Approximately 20% of the technology's target audience has adopted or is adopting the technology as it enters this phase.
Years to Mainstream Adoption	The time required for the innovation to reach the Plateau of Productivity.

Source: Gartner (July 2021)

Table 3: Benefit Ratings

Benefit Rating ↓	*Definition* ↓
Transformational	Enables new ways of doing business across industries that will result in major shifts in industry dynamics
High	Enables new ways of performing horizontal or vertical processes that will result in significantly increased revenue or cost savings for an enterprise
Moderate	Provides incremental improvements to established processes that will result in increased revenue or cost savings for an enterprise
Low	Slightly improves processes (for example, improved user experience) that will be difficult to translate into increased revenue or cost savings

Source: Gartner (July 2021)

Table 4: Maturity Levels

(Enlarged table in Appendix)

Maturity Levels	Status	Products/Vendors
Embryonic	In labs	None
Emerging	Commercialization by vendors Pilots and deployments by industry leaders	First generation High price Much customization
Adolescent	Maturing technology capabilities and process understanding Uptake beyond early adopters	Second generation Less customization
Early mainstream	Proven technology Vendors, technology and adoption rapidly evolving	Third generation More out-of-box methodologies
Mature mainstream	Robust technology Not much evolution in vendors or technology	Several dominant vendors
Legacy	Not appropriate for new developments Cost of migration constraints replacement	Maintenance revenue focus
Obsolete	Rarely used	Used/resale market only

Source: Gartner (July 2021)

Document Revision History

Hype Cycle for API and Business Ecosystems, 2020 - 7 August 2020

Hype Cycle for Business Ecosystems, 2019 - 31 July 2019

Hype Cycle for Business Ecosystems, 2018 - 6 August 2018

Hype Cycle for Business Ecosystems, 2017 - 21 July 2017

Recommended by the Authors

Some documents may not be available as part of your current Gartner subscription.

Understanding Gartner's Hype Cycles

Create Your Own Hype Cycle With Gartner's Hype Cycle Builder

Top 10 Things Software Engineering Leaders Need to Know About APIs

How to Successfully Implement API-First Integration

How to Design Great APIs

How to Use KPIs to Measure the Business Value of APIs

How Are API Management Platforms Priced?

Federate, Rebrand and Recharter Your API Center of Excellence to Enable an API Platform Team

---

Table 1: Priority Matrix for APIs and Business Ecosystems, 2021

Benefit ↓	*Years to Mainstream Adoption*			
	Less Than 2 Years ↓	***2 - 5 Years*** ↓	***5 - 10 Years*** ↓	***More Than 10 Years*** ↓
Transformational		API-Based Digital Commerce CSP Open APIs Digital Business Technology Platform Healthcare Payer APIs Open Banking Strategy		
High	Full Life Cycle API Management Mediated API Pattern	API-Centric SaaS API Testing Services API Threat Protection Business Ecosystem Modeling Business Ecosystems Public APIs	API Security Testing Open APIs in Insurance Partner Ecosystem Management Platforms	
Moderate	API Developer Portals in Banking	API Access Control API Management PaaS API Marketplaces API Standards Banking API Aggregators Event-Driven APIs	Graph APIs	

Benefit	Years to Mainstream Adoption			
↓	Less Than 2 Years ↓	2 - 5 Years ↓	5 - 10 Years ↓	More Than 10 Years ↓
Low				

Source: Gartner (July 2021)

Table 2: Hype Cycle Phases

Phase	*Definition*
Innovation Trigger	A breakthrough, public demonstration, product launch or other event generates significant media and industry interest.
Peak of Inflated Expectations	During this phase of overenthusiasm and unrealistic projections, a flurry of well-publicized activity by technology leaders results in some successes, but more failures, as the innovation is pushed to its limits. The only enterprises making money are conference organizers and content publishers.
Trough of Disillusionment	Because the innovation does not live up to its overinflated expectations, it rapidly becomes unfashionable. Media interest wanes, except for a few cautionary tales.
Slope of Enlightenment	Focused experimentation and solid hard work by an increasingly diverse range of organizations lead to a true understanding of the innovation's applicability, risks and benefits. Commercial off-the-shelf methodologies and tools ease the development process.
Plateau of Productivity	The real-world benefits of the innovation are demonstrated and accepted. Tools and methodologies are increasingly stable as they enter their second and third generations. Growing numbers of organizations feel comfortable with the reduced level of risk; the rapid growth phase of adoption begins. Approximately 20% of the technology's target audience has adopted or is adopting the technology as it enters this phase.
Years to Mainstream Adoption	The time required for the innovation to reach the Plateau of Productivity.

Phase ↓	Definition ↓

Source: Gartner (July 2021)

Table 3: Benefit Ratings

Benefit Rating ↓	Definition ↓
Transformational	Enables new ways of doing business across industries that will result in major shifts in industry dynamics
High	Enables new ways of performing horizontal or vertical processes that will result in significantly increased revenue or cost savings for an enterprise
Moderate	Provides incremental improvements to established processes that will result in increased revenue or cost savings for an enterprise
Low	Slightly improves processes (for example, improved user experience) that will be difficult to translate into increased revenue or cost savings

Source: Gartner (July 2021)

Table 4: Maturity Levels

Maturity Levels ↓	*Status* ↓	*Products/Vendors* ↓
Embryonic	In labs	None
Emerging	Commercialization by vendors Pilots and deployments by industry leaders	First generation High price Much customization
Adolescent	Maturing technology capabilities and process understanding Uptake beyond early adopters	Second generation Less customization
Early mainstream	Proven technology Vendors, technology and adoption rapidly evolving	Third generation More out-of-box methodologies
Mature mainstream	Robust technology Not much evolution in vendors or technology	Several dominant vendors
Legacy	Not appropriate for new developments Cost of migration constraints replacement	Maintenance revenue focus
Obsolete	Rarely used	Used/resale market only

Source: Gartner (July 2021)