CHAPTER

APPLICATION PROGRAMMING INTERFACES AND CONNECTED SYSTEMS

11

Today, everything is connected to everything.
IT Manager, Global Bank

INTRODUCTION

In an industry that is hungry for business agility, the growth of the **application programming interface (API)** Economy promises to enable companies to accelerate delivery of new software supporting new lines of business. Today, for example, consumers have unprecedented power in determining the fate of a business. Failure to deliver in terms of application features and quality is no longer simply a matter of losing a single customer. With the rise of social media, one customer's perception can also quickly influence the perceptions of other potential customers and, in doing so, influence their buying habits.

Software-driven businesses, consumer-facing **applications**, **public cloud**, and mobile applications are all driving growth in API usage, as are social media platforms and **Internet of Things (IoT)**. In short, APIs have become a mainstay supporting business agility and high-velocity business growth. APIs are code sequences that expose data [typically via **representational state transfer (REST)**] or functionality [typically via **simple object access protocol (SOAP)**] for use by an internal or external system. APIs enable diverse software systems to interact with one another to share data, **services**, or functionality. Standards-based in the current era, they offer a shortcut methodology for connecting heterogenous systems via a simpler mechanism than that provided by previous **integration** technologies such as electronic data interchange (EDI).

Before APIs, connecting an internal data center–based application to a supplier or partner application required a custom project and weeks or months of effort. Today, the delay introduced by a similarly lengthy development process is not well tolerated by **line of business** leaders seeking rapid transformation of ideas into action. Businesses seek agility, and APIs offer a way to make brittle organizational borders more flexible.

Despite all the current hype around APIs, they are not a freshly minted miracle technology that was purpose-built to support mobile, **cloud**, and IoT. APIs have been around since the 1990s; however, the API technologies of older generations were not necessarily pretty. They consisted of custom-written, code-enabled "plumbing," with each API purpose-built to support a specific application-to-application connection. Each new integration required an arduous analysis, design, and coding effort, which made **interoperability** a luxury versus the commodity it is today.

The growth of standards during the early part of the 21st century provided a foundation for the growth of API usage and resulting API Economy. The interoperability delivered by standards such as

Application Performance Management (APM) in the Digital Enterprise. http://dx.doi.org/10.1016/B978-0-12-804018-8.00011-5

REST and hypertext transfer protocol/secure (HTTP/S) provides a common technology framework for companies that are, or want to become, more flexible and agile in their digital interactions.

This evolution has made API creation and delivery far more efficient. Virtually every company is now playing by the same rules, utilizing REST-based APIs to communicate over standard HTTP/S protocols that use the same HTTP language (GET, POST, PUT, DELETE, etc.) as **web browsers**. Now supported by standards and built over familiar interaction patterns, modern APIs are easier to build and simpler to maintain and run than their predecessors.

THE ROLE OF TOOLS IN THE API ECONOMY

Most companies start out as API consumers. That is, they access APIs provided by other organizations, typically on a very small scale. For example, many **software as a service** and **infrastructure as a service** vendors offer APIs that can be used by customers to access **management** data, application data, or performance/availability statistics. These types of APIs can offer an opportunity for companies to test the waters of API usage before attempting to utilize an unfamiliar technology for delivery of production-grade business applications. Over time, many companies become API providers as well. These companies create APIs that provide access to their own data or functions. Typically serviced and managed by **API gateway** solutions, provider APIs offer easy access to data and/or functionality to internal and external stakeholders alike.

So how and where do tools fit into this picture? Participating in the API Economy does not stop with providing or consuming APIs. **Security**, access, metering, chargeback, and other API-related functions become increasingly relevant as usage increases. As the number of API provider and/or consumer connections grows, as more users and applications connect, and as new API versions are created and deployed, the API Economy begins to look more like a maze to be navigated than a straightforward way to flexibly extend organizational borders.

Tools help mitigate this complexity by addressing key functional questions. Providers, for example, often find themselves asking:

- How can our organization synchronize API development with traditional application development lifecycles since the two are often linked?
- How can we secure API usage to ensure that sensitive data is protected?
- How can we track usage of for-pay services to correctly bill for access?
- How can we track usage growth and the impact of that growth on back-end systems for capacity planning purposes?
- How do we ensure that only authorized users and applications connect to our systems?

Consumers ask:

- How do we find out about new APIs offered by our vendors and partners, and how do we then go about accessing them?
- How do we know when the APIs our systems are accessing are changed or modified by the provider?
- We have hundreds of applications that access APIs—and some of them interact with one another. When one such application fails, how can we determine what changed, what’s wrong, and how to fix it?

- How do we ensure that only authorized users can access for-pay external services so our usage fees don't skyrocket?

In other words, most companies find that API usage requires similar governance and management capabilities as those required for delivering any other type of software application. They also find that active participation in the API Economy, from a production standpoint, eventually requires investments in automation. In short, these types of questions can only be addressed by tools that are purpose-built to support API delivery and consumption.

THE ROLE OF THE APPLICATION PROGRAMMING INTERFACE GATEWAY

As Fig. 11.1 shows, the vast majority of providers and consumers are using commercial, API-specific management solutions such as API gateways to secure and govern API delivery and/or consumption. Such solutions have value for both API providers and consumers, and are essential elements of overall management of API-connected applications.

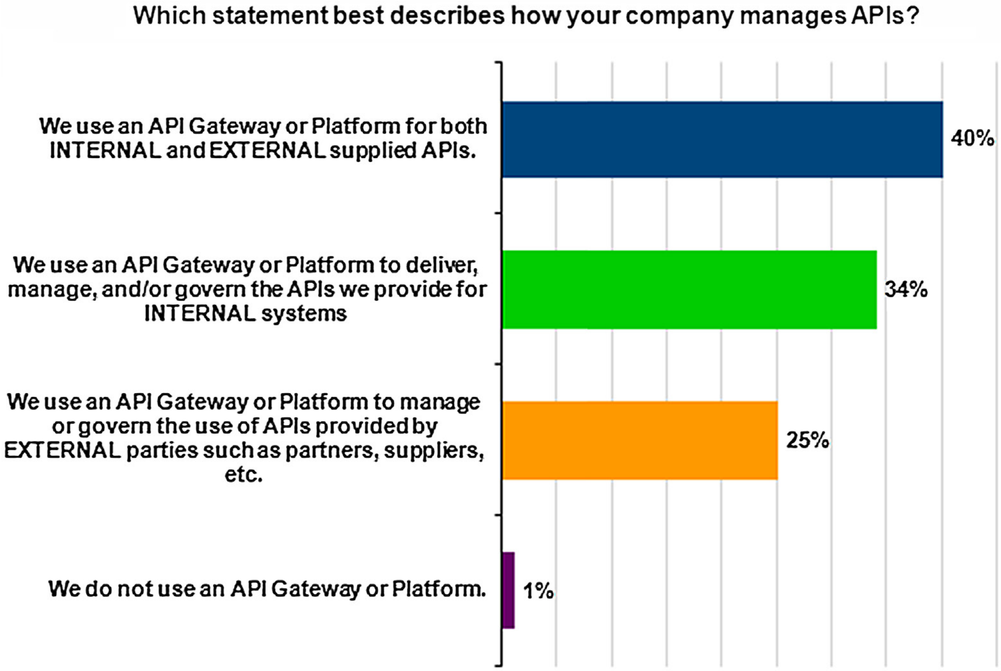

FIGURE 11.1

Application programming interface (API) gateways are used by a majority of both API consumers and providers to manage and govern API connections.

Table 11.1 Examples of Application Programming Interface Gateway Functions[a]

Integrations: Typically integrate with third-party tools supporting multiple additional functions. Integrations with identity management solutions, for example, support authentication and authorization of message traffic. Other examples of integration may include lightweight directory access protocol, Microsoft Active Directory, etc.

Performance optimization: Via offloading traffic from application servers to XML acceleration function on gateway, scaling gateway instances.

REST support: Facilitates exposure of data as provider APIs.

SOAP support: Facilitates exposure of operations and functions as provider APIs.

Reporting: Auditing and reporting on usage to address operational and compliance requirements.

Management: Supports troubleshooting and root-cause analysis by identifying failure points in multiservice transactions and generating alerts.

Traffic monitoring and management: Protects services from traffic spikes via traffic throttling, limits clients to agreed service consumption levels, supports chargeback.

Security-related functions: Identity mediation across multiple identity platforms, governance and metering of API usage, service virtualization designed to shield endpoint services from direct access.

API, *application programming interface;* REST, *representational state transfer;* SOAP, *simple object access protocol.*

[a] *Prospective customers should check specification sheets and vendor descriptions to verify functions supported by specific products.*

As Table 11.1 shows, API gateways[1] have multiple functions including access control, delivering visibility of active users, and mitigation of security risks. Depending on the vendor, API gateways may also provide metering (tracking of API usage supporting billing and/or chargeback), monitoring, and other functions.

In short, API gateways have become the tool of choice for managing all aspects of API delivery and consumption. On the API consumer side, gateways are critical for change notification, usage monitoring, and access control. Considering the importance of change tracking to **performance management** and root-cause analysis, this feature alone could justify the purchase of such a solution. On the API provider side, gateways are critical to tracking users, maintaining security of back-end systems, managing user identity and authentication, and **capacity management**.

Gateways are also important in monitoring and managing applications that access APIs. As standalone solutions, they provide visibility into the health and welfare of APIs being provided and consumed. They also deliver a window into performance and availability of API connections that many companies utilize for monitoring and management of connected applications. However when management data from API gateways can be integrated with application performance management (APM) solutions, IT support teams have an automated basis for managing applications and transactions in context to the APIs they may be accessing.

Increasingly, leading-edge APM solutions[2] are now connecting to gateway solutions for data integration purposes. These connections inject real-time data on API connections into the APM solution for analysis and correlation with other metrics supporting monitoring of end-to-end execution.

[1] More information on how API gateways work is available at http://stackoverflow.com/questions/11331386/how-do-api-gateways-work.

[2] Prospective customers should check specification sheets and vendor descriptions to verify functions supported by specific products.

While most companies today are monitoring API connections from the single point of the gateway, integrations between API gateway monitoring functions and traditional APM platform/suite solutions support true performance management of API-connected applications. These integrations provide a foundation for detecting and resolving issues that may occur within applications/transactions accessing APIs.

EXAMPLE API USE CASE: MICROSERVICES AND APPLICATION PROGRAMMING INTERFACES

Microservices are a good example of state-of-the-art applications with API dependencies. From the business enablement perspective, microservice architectures speed time-to-market for new functions and features, as the componentized form factor enables a building block approach to application/service creation. From the IT perspective, they enable responsive scaling based on load. From the development perspective, they fit well into **agile** development practices and are faster to develop and modify than the monolithic enterprise applications running in many companies.

Often associated with **containerization** using platforms such as **Docker**, **container-based microservices** are portable in the sense that they are capable of running on any server that supports Docker—today usually a Linux-based server. In microservice architectures, monolithic applications are broken into small services (or, alternatively, new code is developed as small services), with each microservice performing a specific set of tasks and running as its own process (see Fig. 11.2).

Often touted as a new architectural model ideally suited for the agile business, software components in the form of microservices can then be strung together (orchestrated) via APIs. Orchestrated services then execute as work streams in much the same way as traditional transactions and applications typically function (see Fig. 11.3).

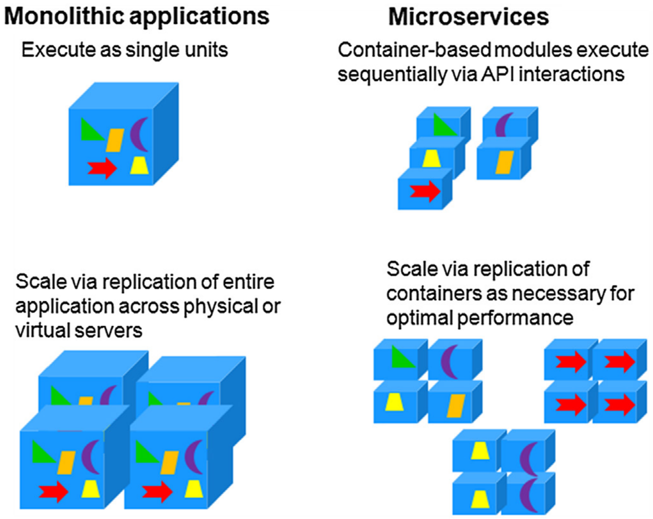
Monolithic applications
Execute as single units
Microservices
Container-based modules execute sequentially via API interactions
Scale via replication of entire application across physical or virtual servers
Scale via replication of containers as necessary for optimal performance

FIGURE 11.2

Microservice architectures enable responsive scaling based on load. They can also speed delivery of new services and expedite time-to-market for new functions and features.

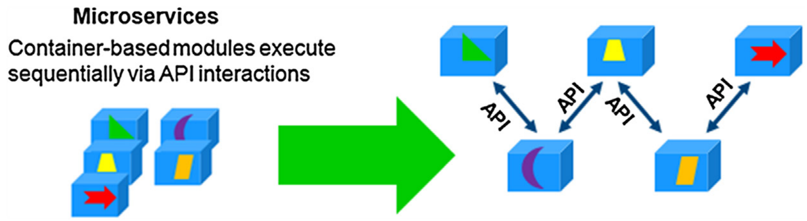
Microservices
Container-based modules execute sequentially via API interactions
API
API
API
API

FIGURE 11.3

Microservices are orchestrated into applications/transactions via application programming interface connections.

With all their benefits, the success of microservice delivery hinges on the performance of the APIs connecting them. The microservices architecture replaces monolithic (or distributed) application delivery with hundreds or potentially thousands of API-based connections. From this perspective, performance and/or availability issues are bound to happen; the only question is when. These challenges highlight the need for integrated, production-grade gateway and APM solutions capable of supporting execution of API-connected services.

APPLICATION PROGRAMMING INTERFACE CONSUMER USAGE IN THE ENTERPRISE

This section and the following section address use cases and value propositions for consumer and provider APIs, respectively. In broad use in today's companies, consumer APIs are essential elements for interacting with partners, suppliers, and customers in the Internet economy. Almost 50% of companies surveyed in a recent Enterprise Management Associates (EMA) study indicated that **consumer APIs** were critical elements supporting revenue generation. These API connections drive inventory and ordering, sales fulfillment, financial reconciliation, and similar functions that are critical to the day-to-day business.

The research also shows that most companies use a substantial number of APIs. Very few API consumer companies are accessing fewer than ten—and as Fig. 11.4 shows, only 10% are in this category. Most are consuming between 11 and 50, and 10% (mostly large companies) are accessing 100 or more.

These numbers are important because they highlight the complexities, from an APM standpoint, of managing applications that connect to APIs. Relatively few companies have full visibility to the dependencies between production applications and the APIs supporting them. And as the number of APIs in use continues to escalate, it becomes increasingly difficult to keep track of API topologies—in essence, which applications are accessing which APIs. In lieu of tools capable of providing visibility into these interrelationships, the business risks increase with each new API rolled into production usage.

In the same way, visibility to change becomes of primary importance. Fig. 11.5 shows the ways in which API users are most often notified of changes to the APIs they are accessing. Since APIs directly connect the user to an API provider's back-end system, it stands to reason that changes to that system (such as database schema changes) may well impact the operation of APIs. And while API providers know when the API or the structure behind it changes, API consumers often have no way of knowing about the change until the application accessing the API stops functioning.

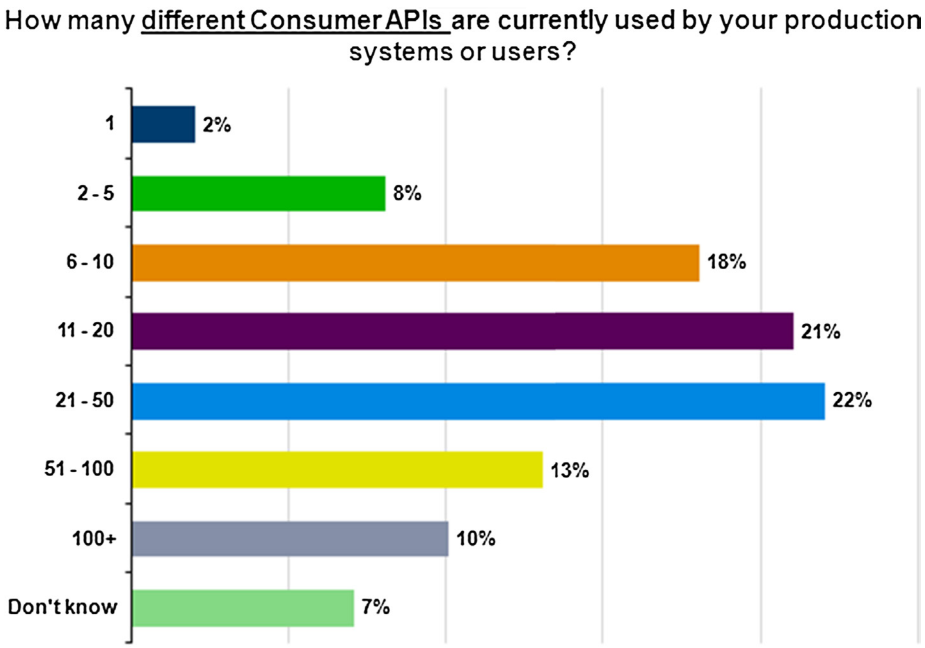

FIGURE 11.4

Most commonly, companies have between 11 and 50 consumer application programming interfaces in use.

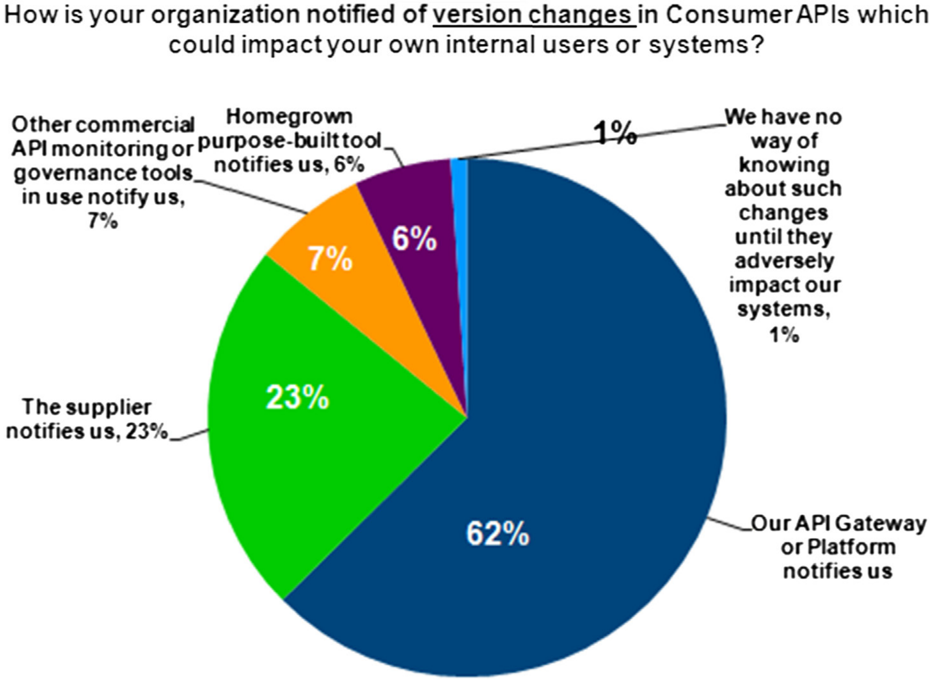

FIGURE 11.5

Application programming interface (API) providers notify consumers of their API of version changes only 23% of the time.

As Fig. 11.5 shows, API consumers are notified of changes by providers only about 25% of the time. Most often, they find out about changes via their own management tools—specifically API gateways—rather than receiving any type of notification from the provider. This is a primary reason why change notification has become an important feature of API gateways. Knowing when APIs change becomes increasingly important as the functionality of the API connection becomes increasingly business critical. If changes may impact transaction viability, IT needs to know if and when a change occurs.

Gateway solutions can also help companies track access to fee-based services provided via API connections. For-pay APIs provide access to paid services, such as financial information (e.g., Dun & Bradstreet), research information, or newspaper and magazine subscriptions. While consumption models can be subscription-based with no limits on access, it is more often the case that consumers access these types of platforms on a per-seat or per-access basis. API gateways/platforms help control access to such platforms on the provider side and can also help track access costs on the consumer side. Per EMA research, more than 60% of API consumer companies surveyed indicated they use their gateways for this purpose.

Fig. 11.6 illustrates the growing volume of API calls made to provider systems by consumer systems. Most companies consuming APIs reported making between 500,000 and 1 million calls to

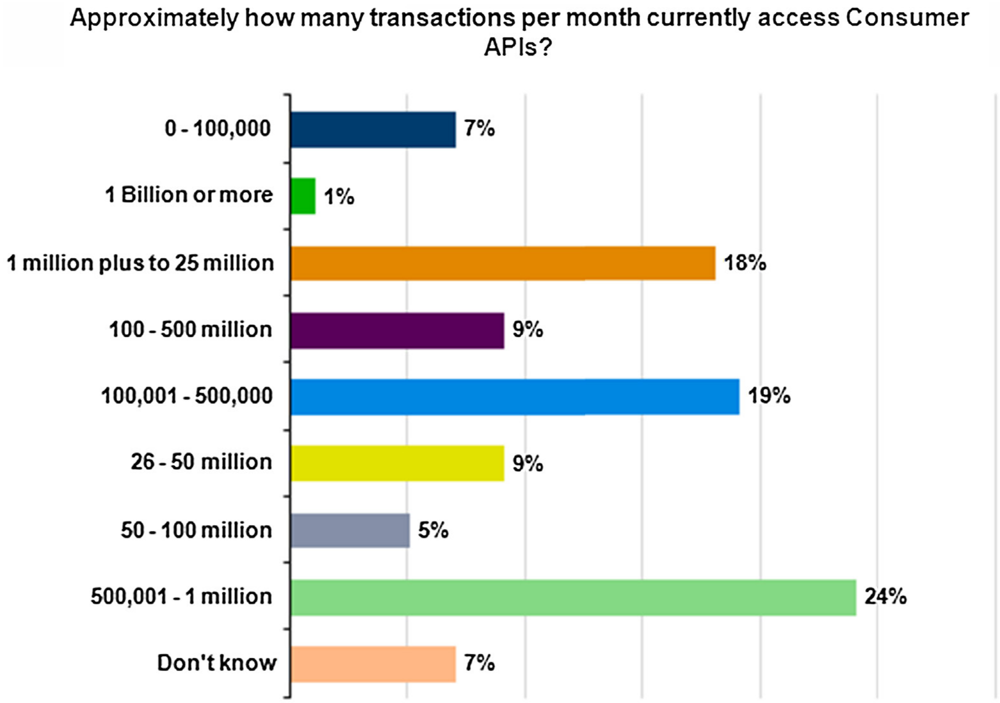

FIGURE 11.6

Most often, between 500,000 and 1 million transactions per month access (external) consumer application programming interfaces.

external APIs every month. However, more than 40% of the companies surveyed by EMA report 1 million or more calls per month. Furthermore, almost 80% report these numbers to be rising, most often by about 20% per month.

These numbers illustrate the facts that API usage is growing and that APIs continue to be increasingly important foundational enablers for eBusiness interactions.

APPLICATION PROGRAMMING INTERFACE PROVIDER USAGE IN THE ENTERPRISE

API providers expose data or functionality for consumption by internal or external entities. Provider APIs are critical or very important to the business at 95% of companies surveyed. Fig. 11.7 shows the types of entities that most often access provider APIs. Customers, suppliers, and partners make up the top three, while internal applications are the primary consumers more than 30% of the time (in this use case, APIs are often used to facilitate internal development by exposing frequently used data or code for easy access by new or existing applications. APIs are also often used to provide mobile access allowing internal users to access internally-delivered applications).

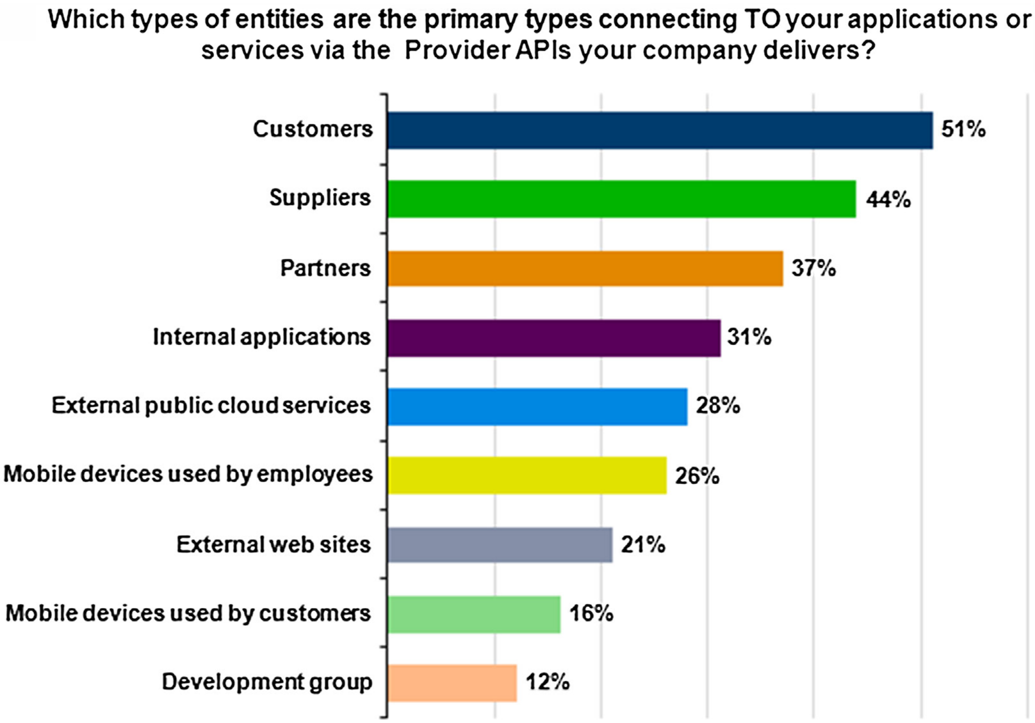

FIGURE 11.7

Customers, suppliers, and partners are the most frequent users of provider application programming interfaces.

Although Fig. 11.7 does not show the targets for these API requests, research points to database connections as the most common points of exposure with mainframes as the second most common targets. This is consistent with reports by IBM and other large vendors, which found that access by **mobile devices**, in particular, is driving increased usage of API connections to back-end mainframe systems. Banks, for example, are refactoring traditional mainframe applications into components and/or microservices capable of handling a single request/response. As consumers increasingly access banking applications from mobile devices, hardware and software that were originally designed for batch processes become far more interactive. This tends to tax computing resources in ways not originally anticipated by designers.

It is also true that the vast majority of providers are not simply supporting a single API. Providers most commonly build and support 21 to 50 APIs; 10% host more than 100. These findings reinforce the need for security and governance support as well as the importance of ongoing capacity management–related measurements and capacity planning.

Fig. 11.8 shows IT professionals' estimates of the volume of transactions accessing the APIs provided by their companies. The numbers are enormous and getting larger by the month. Companies reporting between 100,000 and 1 million transactions per month account for almost 45% of the total. As would be expected, however, this number varies by company size. For example, the 500,000 to 1 million numbers are far more common for small and medium-sized companies. The largest set of

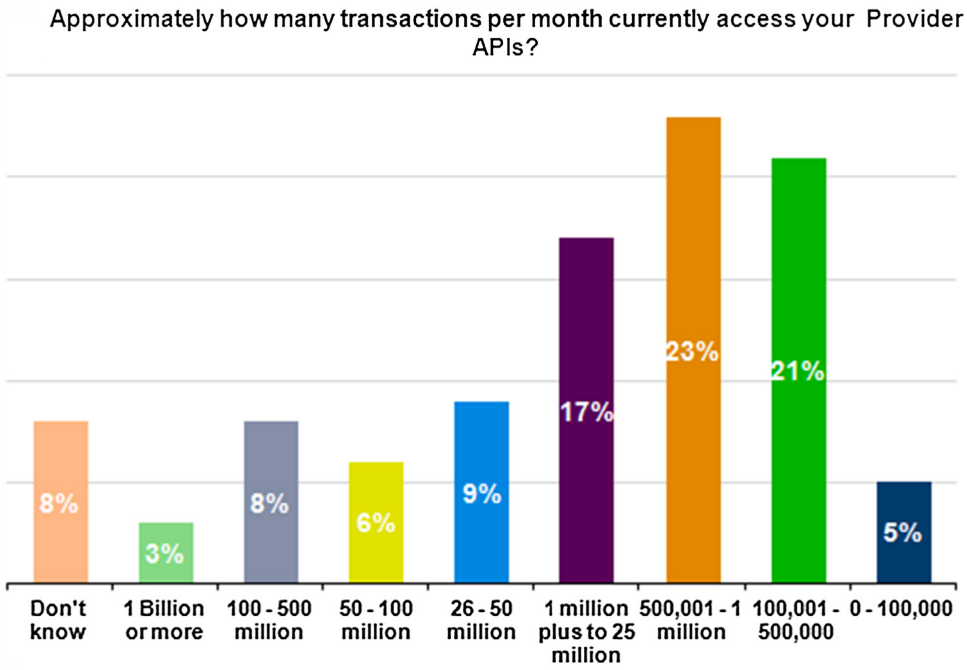

FIGURE 11.8

Most often, a company's provider application programming interfaces are accessed 100,000 to 1 million times per month.

companies, those with 10,000+ employees, most commonly report between 100 and 500 million API accesses monthly; no small companies and very few medium-sized companies experience these volumes.

Providers also often report that once APIs are exposed and become popular, usage growth can mushroom exponentially. Approximately 85% of respondents say traffic volumes are increasing over time. As Fig. 11.9 shows, the most frequently reported growth rate is between 10% and 20% per month.

For capacity planning purposes in the average data center, 20% growth per month is almost unheard of. When API growth percentages are viewed in light of their likely impact on back-end hardware and software infrastructure (including databases, applications, and servers), they reinforce the need for automation supporting performance, availability, and system usage. Visibility to transaction growth is an essential element of capacity planning. And tools capable of monitoring the performance and capacity impacts resulting from such growth are essential to providing a high-quality **user experience** of API-connected applications.

Fig. 11.10 reveals the most common challenges related to API delivery. Traffic volumes top the list by a significant margin, followed by security and identity management concerns. As APIs become increasingly popular with partners, suppliers, and customers, impacts on capacity are apparently as much of a concern as security. In fact, the two appear to be of approximately equal criticality, as there is not enough of a statistical difference between them to categorically declare either as the top issue.

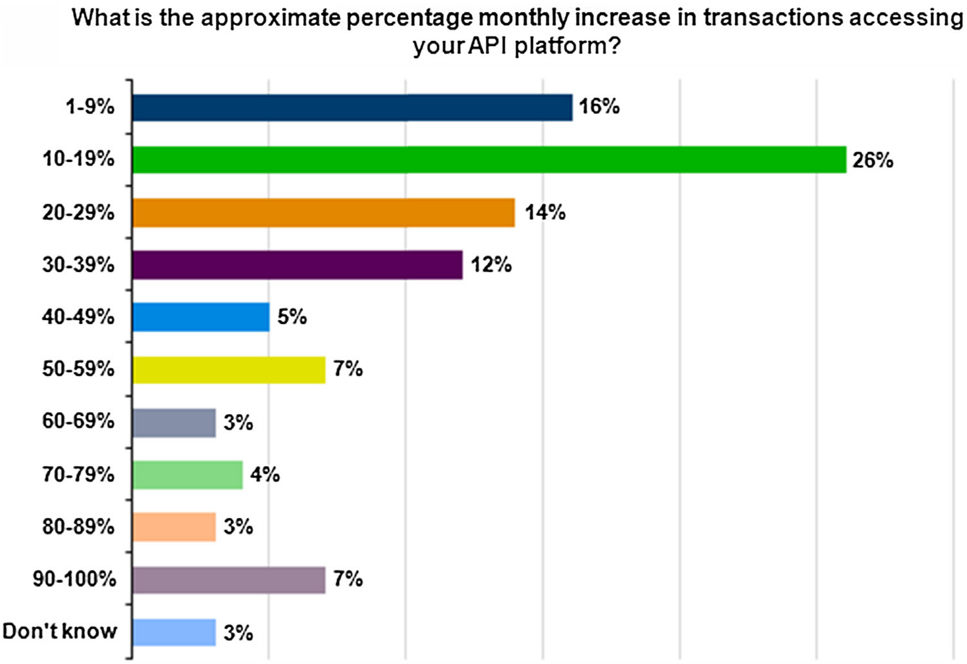

FIGURE 11.9

Monthly transaction growth rates most commonly fall in the 10–19% range.

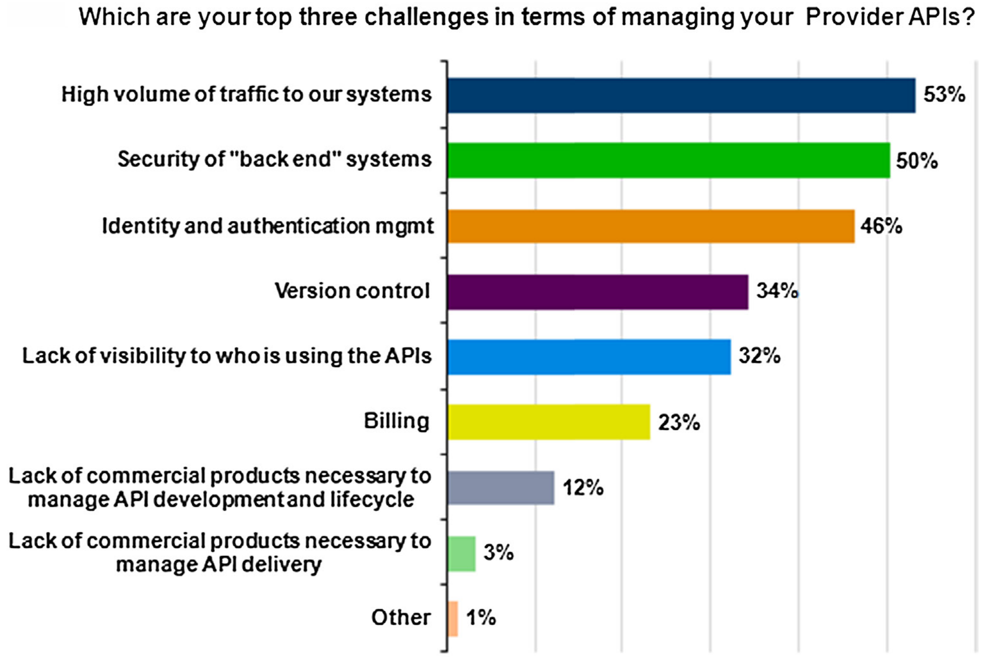

FIGURE 11.10

Traffic volumes, security-related factors are top challenges of application programming interface delivery.

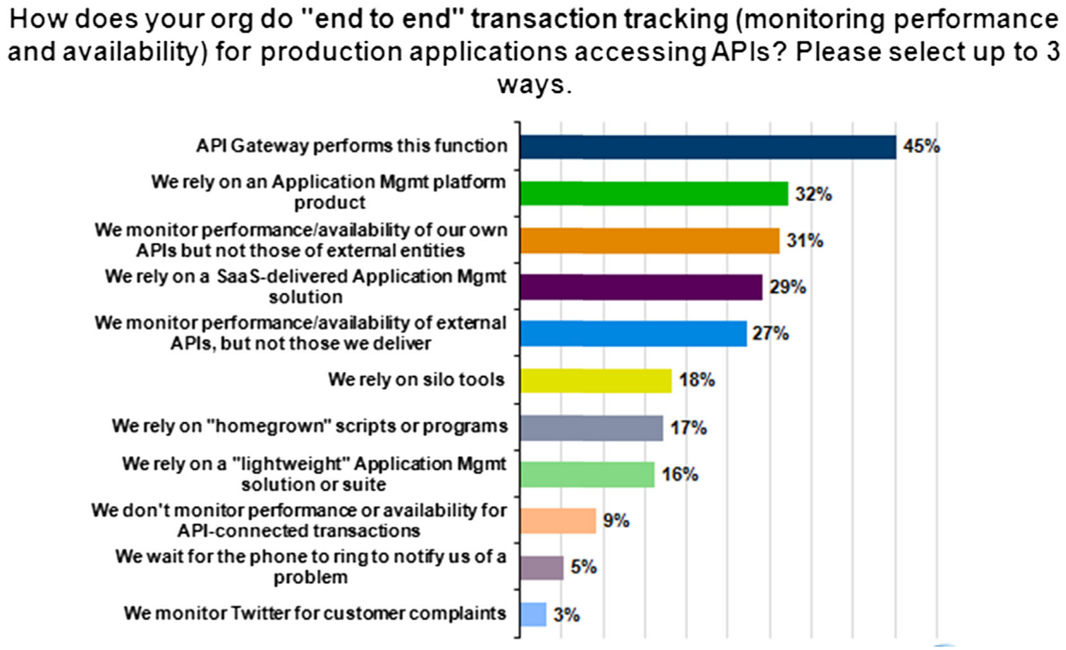

FIGURE 11.11

API gateways are more often used to track performance and availability than application management platforms.

Finally, as Fig. 11.11 shows, a majority of companies monitor performance and availability of applications accessing APIs from the perspective of the gateway. Although this is a good starting point, it is essentially a silo solution to a far broader and more complex problem.

API-connected applications, like any other tiered, distributed applications, have hundreds or thousands of potential failure points. From this perspective, simply monitoring the gateway is akin to monitoring servers, networks, or databases in isolation. In the end, APM platforms should have mechanisms for incorporating gateway data into **analytics**, correlations, and dashboards. Lacking this single point of visibility and control of application execution (versus silo performance), full automation of the end-to-end monitoring/management function remains a fruitless quest.

SUMMARY

APIs can provide a wide variety of business benefits. They can enable a company to become more efficient in delivering new products and services. They allow for a simpler way to integrate compared to the custom integrations and complicated middleware of the past. They enable businesses to become more agile and flexible in their interactions with customers, partners, and suppliers. They support modernization, enabling companies to very quickly adapt to changing technology, application architectures, and business requirements.

However, failure to plan for the realities of API delivery and consumption can be a fatal mistake. Applications relying on APIs for day-to-day execution are obviously business critical and often generate revenue. From this perspective, performance and availability management at multiple levels is an essential element in API delivery as well as API consumption. Gateway solutions support both API providers and consumers.

Chief concerns of providers include traffic volumes, availability, metering (usage of for-pay services), access control, security of back-end systems, and capacity impacts. While most of these functions can be provided by a gateway, the capacity planning function would typically be supported instead by an APM solution in context to data shared by the gateway. This adds visibility to the infrastructure elements impacted by API access and provides a common data and management platform for problem determination and future capacity planning.

Chief concerns of consumers include API change tracking, user authorization to utilize for-pay services, API up-time, and end-to-end performance of applications accessing APIs. Again, the first three are typically provided by the gateway, while end-to-end performance is usually the role of the APM solution. It is also important to recognize the fact that, in many cases, APM solutions will be (or should be) capable of analyzing data provided by external integrations with API gateways. In this case, the analytics supporting the APM solution will correlate both APM-related and gateway-related metrics in context with one another. This is the ideal solution for end-to-end management functionality, as it helps pinpoint the likely source of performance bottlenecks or availability failures.

KEY TAKEAWAYS

- APIs are critical software elements supporting data and/or functionality interchange between diverse entities. They are also utilized for standardizing data access within a software engineering organization to facilitate faster software development.
- The API Economy is well named, since APIs are deemed to be critical or very important to business as usual in the majority of companies surveyed.

- The rise of standards such as REST and HTTP significantly expanded the API Economy because of the ease of use they provide. Today's APIs can be developed and consumed faster and far more easily than was true of the custom-written integrations of past eras.
- API gateways are the universal standard for API delivery and consumption, providing both the real-time link supporting API usage and a host of governance, control, and management functions.
- API gateways are also commonly used for monitoring and managing API performance and availability, for both provider and consumer APIs. However, without information sharing between API gateways and APM platforms, management via the gateway simply creates another functional silo within IT. As with any type of application and **transaction management** function, silo metrics provide the raw material for system management functions (as opposed to **application management** functions). Analytics are the secret sauce that consolidates metrics from multiple sources to provide an end-to-end view of **application performance** and availability.

  Examples of vendors with products in this space:

  3Scale
  Actian
  Adeptia
  Akana
  Apigee
  Attunity
  CA Technologies
  Dell Boomi
  IBM
  Informatica
  Intuit
  Jitterbit
  Liaison Hubspan
  Mashery
  MuleSoft
  Oracle
  Parasoft
  Scribe SW
  SmartBear
  SnapLogic
  Software AG
  Tibco