**API Gateway**

An API gateway is a software pattern that sits in front of an application programming interface ([API](https://www.techtarget.com/searchapparchitecture/definition/application-program-interface-API)) or group of microservices, to facilitate requests and delivery of data and services. Its primary role is to act as a single entry point and standardized process for interactions between an organization's apps, data and services and internal and external customers. The API gateway can also perform various other functions to support and manage API usage, from authentication to rate limiting to analytics.

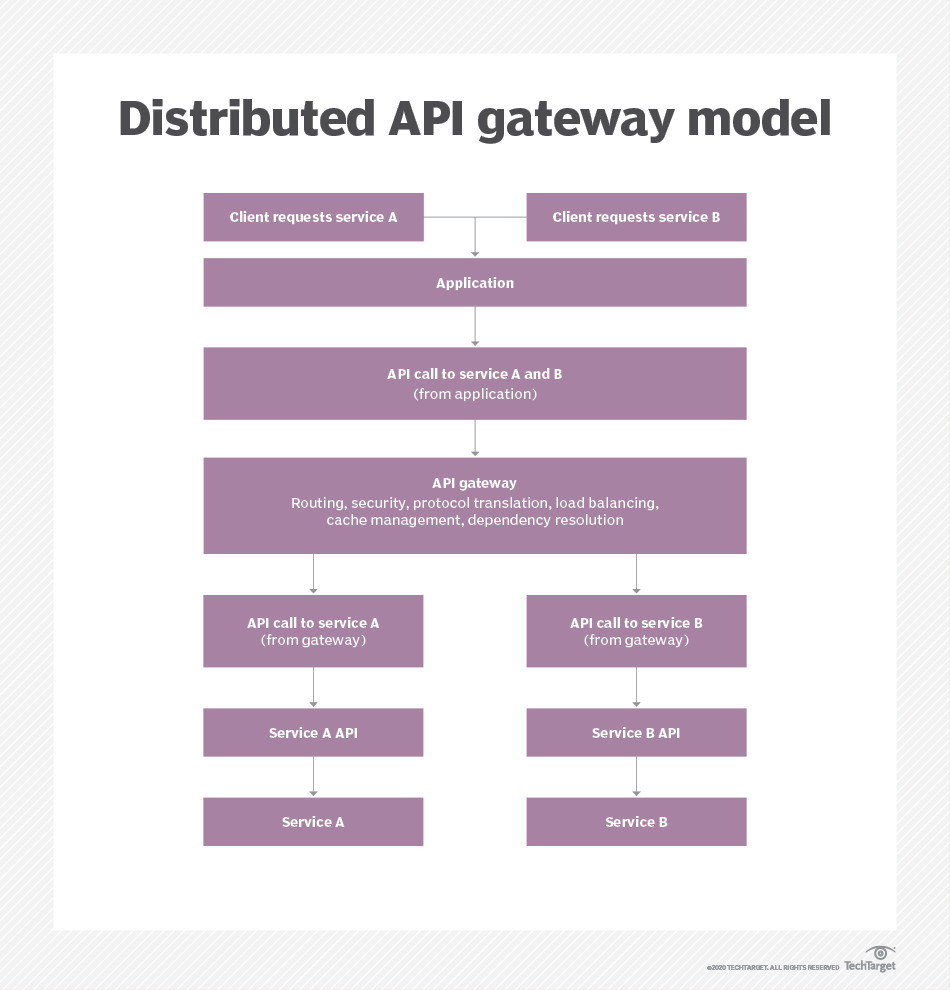
**How does an API gateway work?**

APIs allow separate applications to communicate with each other and exchange data within and outside a business. The API gateway provides a focal point and standard interface to perform these activities. It receives requests from internal and external sources, called "API calls," and packages multiple requests, routes them to the appropriate API or APIs, and receives and delivers the responses to the user or device that made the request.

API gateways also are key to a [microservices](https://www.techtarget.com/searchapparchitecture/definition/microservices)-based architecture, in which data requests invoke numerous applications and services that use multiple, disparate APIs. Here the API gateway's role is similar: Provide a single point of entry for a defined group of microservices, and apply policies to determine their availability and behaviour.

API gateways often handle other functions involved with APIs and microservices:

* protocol translation
* service discovery
* basic business logic
* [authentication](https://www.techtarget.com/searchsecurity/definition/authentication) and security policy enforcements
* stabilization and [load balancing](https://www.techtarget.com/searchnetworking/definition/load-balancing)
* cache management
* monitoring, logging and analytics



### Who uses API gateways and why?

The API gateway is the focal point for API messaging, to organize and streamline API activity and exchanges with internal and external customers. That management and oversight also enables a business to see and control a broad scope of APIs and integrations centrally, rather than attempt to [track and manage APIs](https://www.techtarget.com/searchapparchitecture/definition/API-management) individually. API gateways typically include monitoring and logging capabilities to record and analyze calls and responses in order to ensure security and evaluate errors.

API gateways can also support other functionality that governs APIs. For example, policy managers use logical statements operated through an API gateway to determine the API's availability and behavior, such as how it controls the flow of data, or throttles calls and throughput of API calls.

Organizations that have adopted a microservices-based architecture similarly rely on API gateways to facilitate communications among those services.

API gateways also play a role to help streamline B2B integration as an alternative to legacy approaches such as electronic data interchange services.

**Benefits and challenges of API gateways**

An API gateway's primary benefit is that it standardizes and centralizes delivery of services through APIs or microservices. Beyond this, API gateways also help secure and organize an organization's API-based integrations in a number of ways.

* **Simplify services delivery.** API gateways can combine multiple API calls to request and retrieve data and services, which reduces the volume of requests and traffic. This streamlines the API process and can improve the user experience, particularly for mobile applications.
* **Provide flexibility.** API gateways are highly configurable. Developers can encapsulate the internal structure of an application in multiple ways, to invoke multiple back-end services and aggregate the results.
* **Extend legacy applications.** Enterprises that rely on legacy applications can use API gateways to work with those apps and even extend their functionality, as an alternative to a broader and more complicated (and expensive) migration.
* **Contribute to monitoring and observability.** Most organizations rely on specific tools for monitoring activity through APIs, but [an API gateway can help assist these efforts](https://www.techtarget.com/searchapparchitecture/feature/Three-keys-to-cloud-API-management-success). API gateway logs can help pinpoint an issue during a monitoring failure event.

The API gateway is the gatekeeper between API consumers and providers, and that broad role presents unique challenges.

* **Reliability and resilience.** Any impairment or hindrance to the API gateway's functionality may cause the failure of associated services. Enterprises must be wary of adding features that affect performance, especially since the API gateway represents an extra process step between customers and applications or data.
* **Security.** The API gateway is a trusted source that touches many corners of an enterprise's business. If it is compromised, this is potentially a serious and far-reaching security problem. Businesses should carefully separate external-facing interfaces from internal APIs and systems, and define authentication and authorization parameters.
* **Complexity and dependencies.** Developers must update the API gateway whenever an API or microservice is added, changed or deleted. This is particularly challenging in a model where a few applications could become dozens or hundreds of microservices. Creation and adherence of API and microservices design rules can help mitigate these problems.

### What are the different types of API gateway products?

Given the central importance of the API gateway in today's [API economy](https://www.techtarget.com/searchapparchitecture/definition/API-economy), many providers offer API gateways either as standalone tools or functionality bundled into broader API management platforms.

Examples of API management platform vendors that incorporate some kind of API gateway functionality include Akana, Mulesoft, Postman, Tibco, Workato and others.

Organizations also can separately obtain and use certain API gateway tools. Examples include Apigee (now part of Google Cloud), Express Gateway, Kong Gateway, Oracle API Gateway and Tyk API Gateway.

The major public cloud providers offer API management platforms: AWS API Gateway, Microsoft Azure API Management and Google Cloud Endpoints. They also offer API gateway tools specific to their platforms: Amazon API Gateway, Azure API Gateway and Google Cloud API Gateway (in beta as of January 2021).

**Factors to consider when evaluating an API gateway**

Businesses should weigh several criteria as they choose an API gateway, including the following:

* *Proprietary versus open source.* A business evaluating a vendor's API gateway may already use other products from that vendor. Examples include Oracle, or the major public cloud providers (AWS, Microsoft Azure, Google Cloud). On the other hand, many enterprises are comfortable with open source tools and in-house support.
* *Architecture.* Some API gateway tools emphasize simplicity, while others emphasize extensibility. They also often support different database systems, such as PostgreSQL, Cassandra, Redis or MongoDB. As stated above, businesses that already rely on a particular cloud provider may prefer that provider's API gateway.
* *Customization.* Some API gateways, particularly open source options, may offer more abilities for customization, which requires in-house expertise. However, some API gateways rely on different programming languages, such as Golang or Lua. Ensure that IT staff skills line up with any such requirement. Use of plugins can partly mitigate this issue.