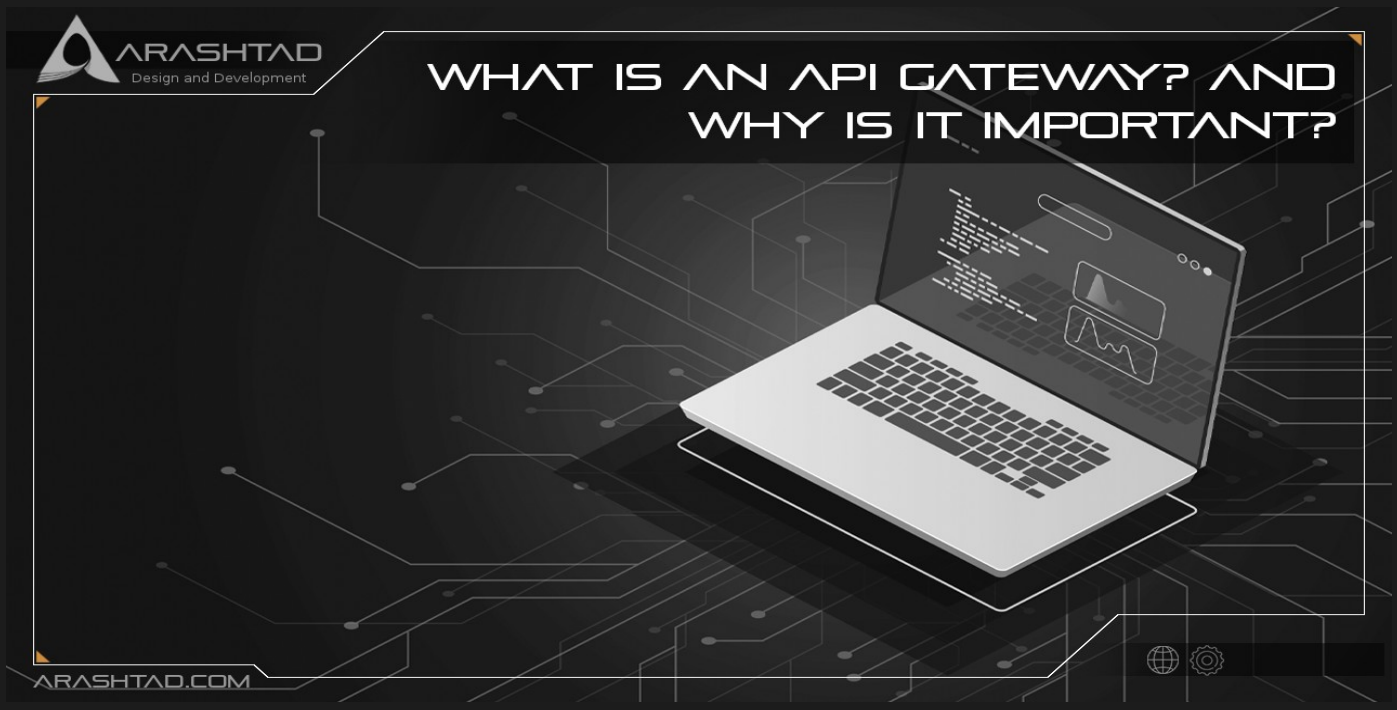
Title	WHAT IS AN API GATEWAY? AND WHY IS IT IMPORTANT?
Description	Intro
Date	2022 15 Augus
Author	Arashtad
Author URI	https://Arashtad.com



API gateways are tools that allow you to manage APIs between a client and a collection of backend services. API gateways act as reverse proxies to collect API calls from clients, aggregate the services required to fulfill them, and return the appropriate results. An API gateway sits in front of a set of APIs or microservices to coordinate data requests and delivery. In addition to acting as a single entry point and standardizing interactions between apps, data, and services, API gateways can also support and manage API usage by performing a variety of other functions, from authentication to rate limits to analytics.

WHAT IS AN API GATEWAY?

Typically, API gateways handle requests by invoking multiple microservices and aggregating the results, so that the best path is determined. they route all API calls from clients to the appropriate microservices through request routing, composition, and protocol translation. The API gateway allows mobile clients to retrieve all product details with a single request using an API gateway. It can translate between web protocols and web-unfriendly protocols used internally. Invoking various services, such as product information and reviews, and combining the results.

One of the components of an API management system is an API gateway. It intercepts all incoming requests and forwards them to the API management system, which handles a variety of necessary tasks. The API gateway differs from implementation to implementation in terms of what it does. An example of a common API gateway function would be authentication, routing, rate limiting, billing, monitoring, analytics, policies, alerts, and security.

WHY DO WE USE AN API GATEWAY?

There are many enterprise APIs deployed via API gateways. these gateways handle common tasks across a system of API services, including user authentication, rate limiting, and statistics. In its simplest form, an API service accepts a remote request and returns a response. But real life is never that simple. When you host large-scale APIs, consider all your concerns. Those concerns are:

1. You use authentication and rate limiting to prevent API abuse.
2. You want to measure how people use your APIs, so you include analytics and monitoring.
3. If you intend to monetize your APIs, you should connect to a billing system.
4. Microservice architectures may involve calling dozens of different applications.
5. Over time, you'll add some API services and retire others, but your clients will still want to find all your services together.

With all of this complexity, your challenge is to provide your clients with a simple and reliable experience. An API gateway lets your client interface be decoupled from your backend implementation. When a client makes a request, the API gateway breaks it into multiple requests, routes them to the right places, produces a response, and keeps track of everything.

HOW DOES AN API GATEWAY WORK?

Using APIs, separate applications can communicate and exchange data inside and outside of a business. API gateways provide a focal point and standard interface for these activities. As well as receiving requests from internal and external sources, it packages multiple requests, routes them to the appropriate API or APIs, and receives and delivers responses to the user or device who made the request.

An API gateway is also essential to a microservices-based architecture, where data requests are invoked by a variety of applications and services using multiple, disparate APIs. API gateways perform similar functions here: Provide a single point of entry for a defined group of microservices, and apply policies to determine their availability and behavior.

Additionally, API gateways handle the tasks that are involved with microservices and APIs. These task are:

1. API Security
2. service discovery
3. basic business logic
4. authentication and security policy enforcements
5. stabilization and load balancing
6. cache management
7. monitoring, logging and analytics
8. API Transformation
9. Rate-Limiting
10. protocol translation

BUILDING MICROSERVICES USING AN API GATEWAY:

Using an API gateway is a good idea for most microservices-based applications since it acts as the single entry point for the system. The API gateway is responsible for request routing, composition, and protocol translation, and can streamline the system. with an API gateway, each client gets a customized API. Depending on the request, the API gateway routes it to the appropriate backend service, while for others it invokes multiple backend services and aggregates the results. if there are any failures in the backend services, the API gateway can return cached or default data to mask them.

PROS AND CONS OF API GATEWAYS:

In addition to standardizing and centralizing the delivery of services via APIs or microservices, API gateways also help secure and organize API-based integrations in a variety of ways. The followings are the benefits of an API gateway:

1. SIMPLIFIES SERVICE DELIVERY:

With API gateways, multiple API calls can be combined to request and retrieve data, which reduces traffic and reduces requests. This benefits mobile applications and streamlines the API process.

2. PROVIDES FLEXIBILITY:

Developers can customize API gateways to encapsulate the internal structure of an application in a variety of ways, to invoke and aggregate multiple back-end services.

3. EXTENDS LEGACY APPLICATIONS:

As an alternative to a broader and more complicated (and expensive) migration, enterprises can leverage API gateways to work with legacy applications and even extend their functionality.

4. CONTRIBUTES TO MONITORING AND OBSERVABILITY:

For monitoring API activity, most organizations use specific tools, but an API gateway can help. Monitoring failure events can be pinpointed using API gateway logs.

In spite of all the benefits mentioned above, there are some challenges that teams might face with an API gateway:

1. Reliability and resilience:
Enterprises must be wary of adding features that adversely affect performance, especially since API gateways represent an extra step between customers and applications or data. Any impairment or hindrance to the API gateway’s functionality may result in the failure of associated services.
2. Security:
If the API gateway is compromised, a serious security problem could occur across a wide range of an enterprise’s business areas. External-facing interfaces and internal APIs and systems should be separated carefully, and authentication and authorization parameters should be defined.

3. Complexity and dependencies:
Every time an API or microservice is added, changed or deleted, developers must update the API gateway. this is especially challenging in an environment where a few applications can become hundreds of microservices. It is possible to mitigate these issues by creating and adhering to API and microservice design rules.

API GATEWAYS VS API PROXY:

There are also API proxies, which are basically subsets of API gateways that provide minimal processing for API requests as an alternative to API gateways. API proxy handles communication, including protocol translation, between specific software platforms, such as proxy endpoints and target APIs. However, API gateways usually provide better performance analysis and monitoring capabilities. in addition, they can control the flow of traffic between sending and receiving points.

HOW AN API GATEWAY SUPPORTS DEVOPS AND SERVERLESS ENVIRONMENTS:

The most common way microservices communicate in organizations that follow a DevOps approach is through APIs. microservices are used to build apps fast and iteratively. Moreover, modern cloud development, including the serverless model, relies on APIs for provisioning infrastructure. you can deploy serverless functions and manage them with an API gateway. In general, APIs become increasingly important as integration and interconnectivity become increasingly important. As API complexity increases and usage increases, API gateways become increasingly valuable.

CONCLUSION

In this article, you have got familiar with API gateways, what they are, their use cases, pros and cons, and their implementation. Essentially, API gateways are API proxies that sit between API providers and API consumers; API gateways are façades that provide API interfaces for complex subsystems. The API gateway acts as a protector, enforcing security and ensuring scalability and high availability of defined back-end APIs and microservices (both internal and external). Typically, The gateway sits in front of an API and serves as its single-entry point. API gateways combine API requests from clients, determine which services are needed, and provide a seamless user experience.