EBOOK | FEBRUARY 2023

Unlocking the Power of Your API Gateway:

Strategies for Improved Security, Speed, and Cost-Efficiency

solo.io

Contents

INTRODUCTION

API gateways have become a crucial part of modern software architecture with the rise of microservices and cloud native software development. Compared to traditional approaches, applications are now smaller, distributed, and focused on a limited business scope. Therefore, communication between applications and their management becomes more critical.

According to a report from Akamai, a significant majority of web traffic, around 83%, was made up of API calls. Then in 2021, a report from Google on the API economy found that businesses are planning to increase their investments in API programs. The top priorities for 2021 included the following:

- API security & governance (50%)
- Growth & management of API adoption (41%)
- Building a developer community (38%)
- Generating revenue through API monetization (31%)
- Making more services and data accessible (31%)
- Expanding investment in API operations and monitoring (20%)

API gateways work as a middleman between microservices and clients, handling requests and directing them to the microservices. Streamlined communication makes the large and distributed machinery run like clockwork. In addition, API gateways can handle various tasks, such as authentication, encryption, caching, and rate limiting. They can also combine responses from multiple microservices into one answer, improving communication and making the overall system more efficient.

Organizations can use API gateways for both internal and external communication. Different teams and their applications use internal APIs, within an organization, to communicate with each other. External APIs, on the other hand, are exposed to external parties such as customers or partners. An API gateway and its rich configuration options enable a company to manage both private and public APIs efficiently.

BENEFITS OF API GATEWAYS

API gateways are driving the next generation of cloud application productivity, resilience, and security. They can improve developer productivity through API management by providing features such as lifecycle management, discovery, versioning, and deprecation. In addition, they can:

- Increase efficiency through API caching
- Improve service availability through API failover and replication
- Enhance scalability through load balancing and throttling
- Increase data security through API encryption and authentication
- Improve thread protection with firewalls and intrusion detection

The transition to the cloud has increased the use of API gateways in a microservices architecture. The number of requests sent to microservices has exploded with the flexibility of the cloud, making it more important to have a reliable and efficient way to manage these requests.

At the same time, the management and usability of API gateways have increased, with the following highlights:

- Automation and self-service portals streamline API deployment and management.
- Caching, compression, and acceleration enhance the overall speed of API gateways.
- Cost-effectiveness increases as the cost of the infrastructure decreases, with less load balancers and incorporating serverless, edge computing, analytics, and reporting tools

API gateways can improve developer productivity, efficiency, service availability, scalability, data security, and threat protection in cloud native software. This article will start with the business cases for API gateways and then discuss making API gateways easier, faster, and more cost-effective.

BUSINESS CASES FOR API GATEWAYS

As the demand for cloud-based applications has grown, so has the importance of API gateways. These intermediary servers are key to managing communication between applications and microservices, helping streamline requests and ensuring smooth data flow.

The benefits of using API gateways are clear whether you're a large enterprise or a small startup. They can help to improve developer productivity, increase efficiency, enhance resilience, and boost security. This article delves into each of these areas in more detail to show how API gateways can help organizations achieve their goals.

PRODUCTIVITY

One of the primary business cases for API gateways is optimizing productivity. With automated API management, organizations save time and resources and streamline the development process.

Improved Developer Productivity Through API Management

API management is a crucial aspect of API gateways that can help organizations achieve improved developer productivity. It includes lifecycle management, discovery, versioning, and deprecation of APIs:

- **API lifecycle management** involves managing an API's development, deployment, and maintenance.
- **API discovery** can help organizations manage their APIs by providing a centralized repository for documentation, version control, and management.
- **API versioning** is vital for organizations that want to update their APIs without affecting existing clients.
- **API deprecation** allows organizations to retire old APIs that are no longer needed, making it easier to manage their API portfolio.

Increased Efficiency Through API Caching and Acceleration

API caching, compression, and acceleration are three additional features that can help organizations achieve increased efficiency.

API caching strategies can help organizations speed up the API response time by caching the API response in a centralized location. API compression can reduce the size of the API response, making it faster and more efficient to transfer over the network. API acceleration techniques, such as HTTP/3 (QUIC) support to accelerate requests and improve data handling, can further improve the speed of the API response, helping organizations achieve increased efficiency.

RESILIENCE

Another business case for API gateways is resilience. API gateways can help organizations improve service availability through API failover and replication; they also enhance scalability through API load balancing and throttling.

Improved Service Availability Through API Failover

API failover is a strategy to help organizations improve service availability, even in the event of a disaster. API failover involves redirecting API requests to a different instance in case of an issue with the primary instance and ensures high availability and business requirements.

Enhanced Scalability Through API Load Balancing and Throttling

API load balancing and throttling allow organizations to enhance scalability.

API load balancing algorithms let companies distribute API requests across multiple instances, ensuring that no single one becomes overwhelmed. API throttling, also known as rate limiting, can help organizations manage the number of API requests processed at any given time, ensuring that the API remains scalable and performant.

SECURITY

Finally, security is another business case for API gateways since they enable organizations to increase data security through API encryption and authentication; they also improve threat protection through API firewalls and intrusion detection.

Increased Data Security through API Encryption and Authentication

Methods of API encryption, such as SSL/TLS encryption, secure the communication between the client and server over the network, shielding sensitive data from unauthorized access and ensuring confidentiality.

API authentication and authorization tools, such as OAuth and JWT, provide an extra layer of security by confirming the client's identity. In addition, these tools guarantee that the requesters have the appropriate rights to access the desired information.

Improved Threat Protection Through API Firewalls and Intrusion Detection

By implementing threat protection security measures, API gateways add an extra layer to secure the data and systems they serve.

API firewall strategies, including rate limiting and IP filtering, control and keep track of incoming API traffic to block undesirable or malicious traffic from reaching the server. This helps protect from DDoS attacks, bots, and other security risks that could harm the system's security.

Intrusion detection and prevention techniques, such as signature-based detection and behavioral analysis, detect and prevent security threats to the APIs. These methods identify and block malicious traffic before it reaches the server, ensuring the system stays secure and guards against potential threats.

API gateways have solid business cases for productivity, resilience, and security. While the significance of API gateways for businesses continues to grow, so does the demand for more streamlined, efficient, and cost-effective solutions. In the following section, we will discuss how to optimize API gateways to meet current demands.

HOW TO MAKE API GATEWAYS EASIER, FASTER, AND MORE COST-EFFECTIVE

The design and implementation of API gateways can be complex, time-consuming, and expensive. To avoid this, it is essential to get the most value out of your API gateway investment. This section will explore critical considerations and best practices for achieving these goals.

AUTOMATION AND SELF-SERVICE

API gateways create a management layer to operate distributed and microservice APIs. With automation and self-service portals, it is possible to increase the accessibility of APIs.

Streamlining API Deployment and Management

The first step in making API gateways easier to implement and manage is to streamline the deployment process. One way to achieve this is to leverage automation tools that simplify creating, updating, and scaling APIs. Automation reduces the time and effort required to implement and manage complex solutions with API gateways.

Another way to simplify API deployment is to provide self-service portals that allow users to discover, test, and deploy APIs independently without needing manual intervention. In addition to lowering the time and effort required, it also makes the process more accessible to users.

API developer portals are another vital tool for streamlining API deployment and management. These portals provide a central location for developers to access the APIs they need with documentation, examples, and guidance.

Enhancing Collaboration and Governance

Another important consideration for making API gateways easier to manage is to enhance collaboration and governance. API collaboration tools, such as wikis and forums, allow organizations to achieve improved cooperation, as they let developers share information and collaborate on projects in real time.

API governance frameworks are also helpful for ensuring that API gateways are well-designed, secure, and scalable. These frameworks provide guidelines and best practices for designing, building, and managing APIs, helping organizations avoid common pitfalls.

PERFORMANCE OPTIMIZATION

API gateways play a critical role in ensuring the performance and reliability of APIs; thus, optimizing their performance is vital to delivering a better user experience.

Improving API Caching and Acceleration

API caching can be an effective way to improve the performance of API gateways. By caching API responses, gateways can reduce the number of requests to the backend servers and handle requests more quickly. Several widely accepted API caching strategies exist, such as client-side caching, server-side caching, and edge caching.

Another way to improve API performance is through API compression. By compressing the data transmitted between the API gateway and the client, organizations can reduce the amount of data in transit, leading to faster response times and a better user experience.

Finally, companies can also implement API acceleration techniques to improve API performance. These include using content delivery networks (CDNs), edge computing, and other performance optimization strategies.

Enhancing Scalability and Throttling

Scalability is another critical consideration when it comes to API gateways. With increasing API traffic, gateways need to be able to scale up and down dynamically to ensure they can meet user demand.

API load balancing algorithms are essential to ensure scalability; they allow organizations to distribute API traffic across multiple servers to improve performance and reliability. Throttling techniques can also help manage API traffic, ensuring that resources are not overused and that the API gateway remains responsive even under heavy load.

COST MANAGEMENT

Implementing and managing API gateways can be resource-intensive, leading to high cloud resource costs. Fortunately, there are some strategies and techniques organizations can use to reduce the costs associated with API gateways. Specifically, reverse proxy features of an API gateway are efficient in reducing the infrastructure cost since fewer load balancers are required.

Reducing Infrastructure Costs Through Serverless and Edge Computing

Serverless computing is a cost-effective alternative to traditional server-based infrastructure for API gateways. By using serverless computing, organizations can reduce their total expenditure by only paying for the resources they use.

Edge computing is another cost-effective choice for API gateways. By processing API requests and responses at the network's edge, organizations can reduce the amount of data transmitted, leading to lower network costs and better performance.

Tracking Usage and Billing with API Analytics and Reporting

API analytics and reporting tools help organizations understand their API usage and costs. By monitoring API traffic and usage patterns, companies gain insights into how they use their APIs and identify areas where they can optimize their usage to reduce costs.

API billing management also enables organizations to better manage their API spend. These tools track usage and generate bills based on usage patterns, allowing organizations to manage their API costs more effectively.

CONCLUSION

API gateways are essential in maintaining the security and performance of APIs by acting as a safeguard and enhancer for incoming data. API gateways have become increasingly user-friendly, efficient, and cost-effective through automation tools, performance optimization methods, and cost management techniques. Nevertheless, there is still room for further development and advancement.

At Solo.io, we believe in making API gateways easier, faster, and more cost-effective for all. Our platform offers a range of features that streamline API deployment and management, enhance collaboration and governance, optimize performance, and manage usage and billing. From API automation tools and self-service portals to load-balancing algorithms and API analytics, we have everything you need to make your API gateway work for you.

Solo.io's solutions not only address problems related to old infrastructure but are built for the future of cloud native technologies.

Take advantage of our platform today, and experience the benefits of a powerful, efficient, cost-effective API gateway.

About Solo.io

Solo.io, the leading application networking company, delivers a service mesh and API platform for Kubernetes, zero trust, and microservices. The three components of the Gloo Platform – Gloo Gateway, Gloo Mesh and Gloo Network – enable enterprise companies to rapidly adopt microservice applications as part of their cloud journey and digital transformation. Solo delivers open source solutions, and is a community leader in building the technologies of the future.

✉ contact@solo.io

🌐 www.solo.io