



solo.io

Solo.io presents

A Buyer's Guide to API Gateways

A comprehensive analysis of modern API gateway solutions

Contents

1. Executive Summary.....	2
2. Market Alternatives.....	7
2.1. Gloo Gateway.....	7
2.2. Apigee.....	7
2.3. Kong Gateway.....	8
2.4. Amazon API Gateway.....	9
2.5. Tyk Gateway.....	9
3. Capabilities Overview.....	11
3.1. Capabilities.....	12
4. Feature Analysis.....	16
4.1. Architecture.....	16
4.2. Scalability and Performance.....	27
4.3. Developer Experience and Documentation.....	34
4.4. Security and Governance.....	39
4.5. Flexibility and Customization.....	47
4.6. Observability and Monitoring.....	50
4.7. Supportability.....	53
5. About the Author.....	59
6. About This Guide.....	60
7. Legal Disclaimer.....	61
8. Glossary.....	63

IMPORTANT NOTICE: This competitive analysis has been carried out on a best efforts basis. Please read our [legal disclaimer](#). If you wish to provide feedback on the document, please complete the contact form at <https://www.solo.io/company/contact/>

1. Executive Summary

In today's development landscape, organizations are realizing the value of an API management system that works across various languages, platforms, and cloud environments. Traditional API management solutions now fall short in meeting the requirements modern workloads demand due to their outdated features, complexity, and inability to keep up with contemporary development practices.

Modern development relies on cloud-native approaches, ephemeral container deployments, GitOps, and continuous integration/deployment (CI/CD) workflows, making it essential for API management systems to adapt to these evolving demands. As organizations increasingly prioritize delivering efficiency, effectiveness, and reliability across their applications, they must concurrently ensure that their supporting infrastructure is scalable and can evolve alongside application requirements.

To address these changes, organizations are now transitioning their stacks to the 'new world' of cloud-native infrastructure. This transition involves embracing lightweight infrastructure, including microservices-based architecture, built on predominantly open source technology to facilitate the hosting and scaling of services while maintaining control of costs.

According to Gartner, by 2027 an estimated 90% of global organizations will have adopted cloud-native container workloads – a significant increase from the predicted 40% in 2021. This global trend underpins the ongoing push for cloud-native digital transformation, emphasizing the crucial need for API management systems to adapt to these evolving demands and changing infrastructure foundation.

In this Buyer's Guide, our evaluation criteria considers what we believe to be vital to a modern development environment. We've identified seven key aspects that we consider essential for a modern application's requirements including;

- **Architectural Design**
Covering concepts including cloud-native design, ARM processor support, serverless support, packaging and deployment, and more.
- **Scalability and Performance**
Comparing concepts including auto scaling functionality, resiliency and load-balancing, and more.
- **Developer Experience**
Exploring ecosystem catalogs alongside GitOps capabilities.

- **Security and Governance**

Comparing the accessibility, authentication, and traffic management features and each solution's capabilities in building the foundation for zero trust security.

- **Flexibility and Customization**

Analyzing the ease of use of each solution and comparing the available integrations and customization options.

- **Observability and Monitoring**

Benchmarking logging, metrics, and analytic functions of each gateway provider.

- **Supportability**

Evaluation of documentation, long-term enterprise support, and community backing of each solution.

It's important to note that the API management and gateway sector has evolved alongside changes in the infrastructure sector over the past two decades. As organizations swiftly move away from their legacy monolithic workloads to modern microservices-based applications, the API sector has continued to adapt accordingly. Today, there are hundreds of API service providers on the market, spanning from those designed for legacy systems to those built for new cloud-native workloads.

Readers of this Buyer's Guide will get a comprehensive comparison of five leading API gateway solutions on the market including: Gloo Gateway by Solo.io, Apigee from Google Cloud, Kong Gateway from Kong, Amazon API Gateway from Amazon Web Services, and Tyk Gateway from Tyk.

Solo.io was founded in 2017 and is a new yet prominent leader in the cloud-native application networking space. Its portfolio of solutions is designed to help users connect and secure their cloud-native applications and APIs at scale and across any environment, whether it be on-premises, hybrid, or multi-cloud.

Solo.io's portfolio encompasses Gloo Gateway, Gloo Mesh, and Gloo Network for Cilium, all leveraging popular open source technologies such as Kubernetes, Istio, Envoy, Cilium, eBPF, GraphQL, WebAssembly, Linux, and more. The product portfolio of Solo.io is strategically built upon these technologies due to their vibrant, engaged communities and widespread adoption by industry leaders. Notably, Envoy, a cornerstone of Solo.io's offerings, has been endorsed by major cloud service providers for its reliability and versatility as a proxy solution. By aligning with these open source ecosystems, Solo.io empowers both customers and community users to scale their modern infrastructure networking effectively, facilitating the easy operation of cloud-native applications.

Solo.io has strong ties across the open source community, holds influential positions in the Cloud Native Computing Foundation (CNCF), and is a major contributor to CNCF open source projects worldwide including Istio, Envoy, and Cilium. Solo.io simultaneously supports their growing enterprise customers with their application networking demands alongside their commitment to the open source community.

Apigee was founded in 2004 as Sonoa Systems and rebranded to Apigee in 2010. In 2016, Apigee was acquired by Google. Prior to its acquisition, Apigee was an established player in the traditional API management space and has maintained a steady customer base as part of Google Cloud Platform.

Kong was originally a product developed by Mashape in 2009, comprising of a mixed platform used to aggregate functions and UI elements from various third party products and services. In 2015 Mashape launched the open source project 'Kong' and in 2017 Mashape rebranded to Kong to capitalize on its open source project. Kong is an open core provider of API management tooling with an open source version. The enterprise version of Kong extends the basic functionality of its open source version with a focus on API management and operations, developer experience, access, security, and performance across hybrid and multi-cloud environments. The enterprise Kong portfolio consists of Kong Gateway, Kong Connect, Kong Insomnia, and Kong Mesh.

Kong utilizes a combination of Nginx, OpenResty, and Lua across its portfolio's foundation. Though technically open source, users may experience limitations in the supportability of these technologies because of the small community base behind these projects. In 2016 Nginx was acquired by F5, and in February 2024, a fork of Nginx 'freenginx' was announced by a core maintainer and original developer of Nginx, who questioned F5's commitment to the openness of the original Nginx project. The uncertainty surrounding Nginx's future may have future downstream implications on Kong.

Amazon Web Services (AWS), founded in 2002, is a leader in the cloud and infrastructure space, providing its customers a vast portfolio of different on-demand cloud computing services including application, infrastructure, and network platforms. Their products are primarily designed for AWS users already integrated into their existing ecosystem looking for a simple way to scale their computing capacity. Their services are primarily provided on a subscription and consumption pay-as-you-go model.

Tyk was founded in 2016 and positions itself as an open source API gateway and management platform. Tyk is not built on any existing open source community-based project and owns its portfolio's entire stack. Tyk does make some

limited parts of its portfolio open source, however, a majority of its portfolio is closed source, which may be a limitation for users looking for solutions built on a foundation of open source projects.

This Buyer's Guide aims to provide a general understanding of the landscape around leading API gateway solutions, to help organizations make informed decisions that align with their specific use cases and their own development practices and operational needs. Readers can learn more about each vendor's API gateway solution in the 'Market Alternatives' section of the document.

2. Market Alternatives

In this guide, we will be evaluating five of the leading API gateway solutions and scoring their functionality using the seven categories mentioned previously. Benchmarked solutions will be compared based on their enterprise offering.

2.1. Gloo Gateway

Built by Solo.io, Gloo Gateway is a fast and lightweight Envoy-based Kubernetes-native API gateway. It is part of Solo's wider application networking platform Gloo – which also includes additional application networking tools including Gloo Mesh and Gloo Network for Cilium.

Gloo Gateway is built for modern development teams and brings security capabilities alongside API-management, including traffic management, insights, multi-cloud, and performance for applications built on any cloud-native environments.

Gloo Gateway is a dynamic and reliable solution for modern organizations that run cloud-native applications built on Kubernetes, VMs, serverless/FaaS, or other container environments. It is innately designed to work on cloud-native workloads hosted across hybrid, multi-cloud environments and does not require any centralized databases. In fact, Gloo Gateway's configuration is built from the ground up on declarative configuration (or Custom Resources in Kubernetes).

Gloo Gateway is built upon and tightly integrated with a collection of open source projects that Solo maintains and contributes to as part of its commitment to the Cloud Native Community Foundation (CNCf).

The 'CAKES' stack is a framework coined by Solo.io to help users in the community choose the best open source projects for their cloud-native application networking requirements:

- Cilium - Container Networking Interface (CNI)
- Ambient/Istio - Service Mesh Requirements
- Kubernetes - Container Orchestration System
- Envoy - Application Proxy
- Spiffe / SPIRE - Identity framework securing communication between workloads

2.2. Apigee

Since the acquisition, Apigee has commanded a strong market share for traditional API management use cases. Apigee leads with a business narrative focused around the API economy, monetization, external innovation, ecosystems, and the enterprise digital value chain. Its core features center around API management, runtime, monitoring, analytics, and developer services to help organizations manage the governance of their API ecosystem.

Apigee offers different options across its API management and gateways:

- 'Apigee X' is the managed service option hosted on Google Cloud, where Google also manages the API infrastructure.
- 'Apigee Hybrid' is a service provided by Google that enables customer-managed runtime on private, AWS, or Azure cloud environments via a Apigee-managed control plane. Organizations have control and full responsibility of the ingress and egress of data paths.
- 'Apigee Edge' is a platform for development and managing APIs with a proxy layer that provides an abstraction for your APIs.
- 'Apigee Adapter for Envoy' is shim tied using Envoy proxy for a simple subset of API gateway capability.

In this guide we will refer, compare, and analyze all service and product offerings collectively as 'Apigee.'

2.3. Kong Gateway

Kong Gateway is part of Kong's portfolio of API management solutions. Kong Gateway is the portfolio's API gateway solution designed for hybrid and multi-cloud environments that is optimized for microservices and distributed architectures. It is positioned as a modern cloud-native API gateway that is lightweight by design. The functions of Kong Gateway center around users being able to automate GitOps workflows, decentralizing applications and services, and enriching the developer platform, security management, and governance of APIs.

There are two ways to deploy Kong Gateway:

- 'Managed by Kong Konnect,' which provides a global control plane hosted in the cloud by Kong, where users can manage the individual data plane nodes within their preferred network environment.
- 'Self Managed via the Open Source (OSS) or Enterprise package'. The open source package contains the basic API gateway functionality and open source plugins and the enterprise package includes the API gateway with additional

features including RBAC and other plug-ins.

For consistency throughout the guide, we will reference all of Kong's enterprise API Gateway components as 'Kong'.

2.4. Amazon API Gateway

Market leading cloud service provider Amazon Web Services (AWS) provides the Amazon API Gateway, which is a fully managed service available for AWS users with integration into the existing AWS suite of cloud services. Its capabilities center around traffic management, developer integrations, security and governance, and simplicity across AWS services. The Amazon API Gateway has been exclusively designed for existing AWS customers and their development teams building applications and services in AWS based environments, and has an extensive ecosystem of integrations built across AWS products.

There are three ways to use the Amazon API Gateway including;

- 'HTTP APIs' are a part of the RESTful API products, and are designed with minimal features and generally used to send requests to AWS Lambda functions or any routable HTTP endpoint.
- 'REST APIs' are a part of the RESTful API products and support more functionality than HTTP APIs and are a good fit for serverless workloads.
- 'WebSocket APIs' enable users to build real-time two-way communication applications with the Amazon API gateway, maintaining a persistent connection between your frontend and backend services.

The AWS API Gateway is only sold to existing AWS customers already with workloads running in the AWS ecosystem.

2.5. Tyk Gateway

Tyk was founded in 2014 and has a diverse portfolio of API-based open and closed source solutions. Tyk Gateway is part of their open source portfolio, which also includes:

- Tyk Pump - Analytics purger tool that sends API analytics data to a chosen backend
- Tyk Operator - Agent deployed to your Kubernetes cluster to manage API configurations
- Tyk Identity Broker - Component that connects third-party identity management systems
- Tyk Sync - Command-line tool to help control API configurations

Tyk's proprietary source portfolio extends the open source portfolio and provides enterprise API management features to cloud-native microservices with a focus on application scalability through insights, developer integration, security, and governance. The portfolio includes:

- Tyk Developer Portal - Exposes a facade of APIs to allow third-party developers to register and use APIs.
- Tyk Dashboard - The GUI and analytical platform for managing Tyk installs.
- Tyk Multi Data Centre Bridge - Extension to Tyk control plane that synchronizes and manages distributed clusters of Tyk API Gateway.
- Universal Data Graph - Combines multiple APIs into one universal interface to help users access multiple APIs with one query.

To maintain consistency throughout the guide, we will compare both the open and closed source capabilities of Tyk Gateway and relevant auxiliaries and collectively refer to them as 'Tyk.'

3. Capabilities Overview

A Buyer's Guide to API Gateways provides a thorough examination of the strengths and weaknesses of each vendor, delivering a detailed comparative analysis. To facilitate a quick overview in this executive summary, we have visually depicted the relative performance of each vendor below.

In this overview, and throughout this document, we will summarize the scoring using the following ranking in each category:

- The full ball (●): best-in-class platform
- The three-quarters ball (◐): runner-up
- The half ball (◑): acceptable capability
- The quarter ball (◒): weak capability
- The empty ball (○): no capability

	Gloo Gateway	Apigee	Kong Gateway	Amazon API Gateway	Tyk Gateway
Architecture	●	◑	◐	◒	◐
Scalability & Performance	●	◒	◐	◒	◐
Developer Experience & Documentation	●	◒	◐	◒	●
Security & Governance	●	◐	◐	◒	◐
Flexibility & Customization	●	◐	◐	◒	◐
Observability & Monitoring	●	◒	◐	◐	◐
Supportability	●	◒	◐	◒	◐

3.1. Capabilities

When evaluating API gateway solutions, organizations often consider various criteria to ensure that the chosen solution aligns with their specific requirements and objectives. These categories have been selected to focus this competitive evaluation with a handful of criteria. These are the seven categories we will use to evaluate API gateway solutions: Architecture, Scalability and Performance, Developer Experience and Documentation, Security and Governance, Flexibility and Customization, Observability and Monitoring, and Supportability.

3.1.1. Architecture

When evaluating API gateway solutions, understanding the ability for the architecture to support modern applications is critical. As many new applications are developed for the cloud or are part of an ongoing lift-and-shift exercise, it is important to recognize the difference between the existing monolithic application and a cloud-native application. To adapt to the highly distributed and constantly changing nature of modern cloud environments, an API gateway should be dynamic and able to run independently in a container to allow for better distribution and scale along with the underlying application.

Architecturally, this flexibility is best achieved with a lightweight API gateway solution that can be utilized to help scale microservices and cloud-native-based applications. To achieve this, it is also important to consider solutions without a static database tier that requires an independent management effort. And finally, to optimize deployment control, infrastructure operations should be completely API-driven (GitOps).

3.1.2. Scalability and Performance

Assessing the performance and scalability of an API gateway is crucial to ensure that it can handle the anticipated volume of incoming requests. Factors such as response times, throughput, and the ability to scale horizontally to accommodate increased traffic are essential considerations. A performant API gateway ensures efficient communication between clients and services, contributing to a responsive and reliable system.

The cloud-native API gateway also needs to be lightweight and be able to easily scale along with the microservices it supports. Using virtual machines (VMs) can hinder the ability of the API gateway to scale with the application it supports.

3.1.3. Developer Experience and Documentation

The developer experience is critical for effective API gateway adoption, emphasizing user-friendly interfaces, comprehensive documentation, and clearly defined APIs. The evaluation process focuses on factors like the simplicity of API configuration, the availability of developer tools, and the thoroughness of documentation and integration with best-of-breed tools. A well-documented API gateway not only streamlines onboarding, minimizing the learning curve and expediting development, but also contributes to a positive experience and workflow for developers interacting with the system.

3.1.4. Security and Governance

Security is a top concern when evaluating API gateway solutions. The best API security solution will make it easier to adhere to security policies while allowing for some flexibility in how components are integrated together. The API gateway should provide robust security features to protect against common threats such as unauthorized access, data breaches, and denial-of-service attacks. Key security features include: OPA, authentication mechanisms, authorization controls, encryption, and support for industry-standard security protocols like OAuth and OpenID Connect. It should also integrate with data loss prevention (DLP) and web application firewall (WAF) components as part of its security policy framework. The ability to enforce security policies consistently across APIs is also critical.

Most security implementations will include a requirement that solutions integrate well with existing platforms to maximize value and mitigate tool sprawl. A good API security offering will integrate through declarative configuration and automation with existing secrets management and internal certificate signing systems. Most leading API security solutions will automatically discover existing secrets management and certificate services to leverage them as needed to accelerate an application's time to market in a safe and scalable manner.

Security and governance also forms a core part of our review, focusing on capabilities around role-based access control (RBAC), external authentication, traffic management, encryption, rate limiting, and advanced security measures, all integral to maintaining robust API security and governance. Modern API gateway providers should also provide users with the capability to build the foundations of a zero trust security framework on their applications and network with control, access, and permissions.

3.1.5. Flexibility and Customization

API gateways must provide flexibility to accommodate multiple use cases and meet specific business requirements. Organizations often have unique needs concerning data transformation, routing logic, and integration with various backend services. An effective API gateway enables customization through plugins, scripting, or configuration options, allowing organizations to tailor the gateway to their specific requirements without compromising performance or security.

As the solution coverage expands to encompass various use cases, it's also important to consider the potential additional overhead required for separate scenarios. These use cases may span from individual, small applications to expansive, multi-cluster, and multi-tenant architectures, covering both internal and external API scenarios.

Cloud-native applications require a robust solution that can facilitate secure access through multiple paths, covering legacy RESTful APIs, gRPC, and more targeted methods like GraphQL. These solutions also need to be able to support multiple cloud platforms to support current and future development directions. Alternatively, users may choose a single API gateway that will work on-prem, in any cloud vendor, or a hybrid of both as a more suitable pathway for their application.

3.1.6. Observability and Monitoring

Comprehensive monitoring and analytics capabilities are vital for gaining insights into API usage, identifying potential issues, and optimizing performance. API gateways should provide real-time visibility into metrics such as request/response times, error rates, and traffic patterns. Integration with monitoring tools and support for logging and auditing are essential components. An effective API gateway should empower organizations to proactively manage and troubleshoot their APIs, ensuring optimal performance and reliability.

Observability is essential as part of application security. The added visibility provides important insights across an application's performance stats and also collects crucial metrics and data on actions that may be blocked or at risk. Teams should be able to understand the insights and metrics collected across their APIs to ensure that they build applications with the confidence and transparency to mitigate any roadblocks or challenges.

3.1.7. Supportability

The role of supportability lies in facilitating smooth operations, timely responses to challenges, and sustained development, ensuring the ongoing success of these software types in meeting diverse user needs. Supportability is a critical factor for

both enterprise and open source software, shaping their effectiveness and user satisfaction. In the enterprise context, the swift resolution of issues is paramount to minimize downtime and maintain smooth operations. This includes timely bug fixes, efficient user training, and support for customization needs. Security is another shared concern requiring vigilance to address vulnerabilities promptly and maintain user trust.

For open source solutions, consider the size and activity of the community around the API gateway, the availability of commercial support, and the frequency of updates and patches. A robust support system is essential for users encountering bugs or seeking guidance. The community's responsiveness is crucial for troubleshooting, providing workarounds, and ensuring the software's continued improvement.

Customization and integration support are common requirements for software that can be tailored to specific needs, necessitating guidance during the customization process. With open source software, a supportive community aids developers in understanding the codebase and ensuring seamless integration with other components.

Throughout this guide, these seven categories will form the framework for comparison across the different API gateway solutions. This guide will delve deeper into each of the seven categories with comparisons to help organizations determine which solutions fit their specific needs best.

4. Feature Analysis

4.1. Architecture

	Gloo Gateway	Apigee	Kong Gateway	Amazon API Gateway	Tyk Gateway
Cloud-native design	●	◐	◑	◑	●
Virtual machine (VM) connectivity	◑	●	●	◐	◑
ARM processor support	●	○	●	◑	●
Serverless function support	●	◐	●	◐	◑
Packaging	●	◑	●	◑	◑
Event-driven architecture	◑	◐	◐	◐	◑
Gateway-to-gateway communications	●	◑	◑	◑	●
Multi-tenancy and isolation	●	◑	●	◑	●

4.1.1. Cloud-Native Design

Solution vendors provide comprehensive microservices support, container orchestration, and/or Kubernetes support.

- Gloo Gateway: ●
- Apigee: ◐
- Kong Gateway: ◑
- Amazon API Gateway: ◑
- Tyk Gateway: ●

Gloo Gateway: Gloo Gateway is designed for scalability and high availability using a control plane/data plane-based architecture to support cloud-native applications.

Gloo Gateway is built on the highly performant and de-facto community standard Envoy proxy. Configuration is built around the Kubernetes Gateway API.

Using a declarative configuration approach, developed using Kubernetes Custom Resource Definitions (CRDs) and built on the Kubernetes Gateway API, Gloo Gateway follows a GitOps approach to automation and continuous delivery. This declarative API can be aligned to specific personas within the organization and leverages delegation where appropriate. For example, infrastructure, SRE, and developer teams can each configure the parts of the API management that makes most sense to them. This reduces the need for centralized teams to own all configurations and reduces time to change. Gloo Gateway is easily installed onto Kubernetes clusters via `glooctl` or via Helm Support for cloud-managed data planes, which is in progress and should be available soon.

Apigee: Apigee users looking to host and manage containerized runtime services in their own Kubernetes cluster can do so with Apigee Hybrid. Users may need to use Apigee Adapter for Envoy alongside Apigee Hybrid to deploy a HTTP service in the same Kubernetes cluster to manage the API calls with Apigee. Since Apigee is tied into Google Cloud, users must ensure their GCP project is associated with their hybrid organization.

Kong Gateway:

For Kubernetes, Kong Gateway repackages its stand-alone, DB-based gateway into a container and leverages the Kubernetes Gateway API. In this mode, DB mode can be turned off and leverage CRDs for configuration. Additionally, users have the option of using Kong Konnect to manage the Kong Gateway. Kong Gateway can also be deployed outside of Kubernetes, but requires a highly-available backend database.

Kong Gateway operates on a ‘freemium’ model with limited features in the free version with an enterprise subscription required to access the full feature-rich components, including the developer portal and enterprise plugins to help with operational and development workflows.

Amazon API Gateway: AWS API Gateway remains a single ecosystem solution. Recently, in April 2023, AWS announced the general availability of Amazon Virtual Private Cloud (VPC) ‘Lattice’. Part of this release is the introduction of the ‘AWS Gateway API controller,’ an implementation of the Kubernetes Gateway API designed to give AWS users with cloud-native workloads better Kubernetes application networking with access to custom resources and Kubernetes APIs.

Tyk Gateway: Tyk Gateway provides two self-managed pathways to install into Kubernetes clusters. The preferred installation method is via Helm charts. The second option is through the 'Tyk Operator' and Kubernetes ingress controller.

By default, the Helm chart will install Tyk Gateway, Tyk Dashboard, Tyk Pump, and Tyk Enterprise Developer Portal (license required to access). Tyk does provide a good number of integrations including frameworks like OpenTelemetry and 3rd Party Identity Providers in its open source version. Alternatively, users can utilize Tyk Operator to manage their APIs in Kubernetes, and Tyk can be installed across different environments directly on the clusters, across hybrid or multi-cloud workloads.

4.1.2. Virtual Machine Environments

- Gloo Gateway: •
- Apigee: •
- Kong Gateway: •
- Amazon API Gateway: •
- Tyk Gateway: •

Gloo Gateway: Adopts an automated, self-service approach to API Gateways in line with cloud-native platforms like Kubernetes, and although it is designed for decentralized ownership and dynamic ephemeral environments, it is still a powerful API Gateway tool for existing on-premise environments including VM deployments. Configuring routing, policy and API management, rate limiting, etc. to backend VM deployments follows exactly the same approach for configuring cloud-native workloads. Gloo Gateway sees all backend endpoints (containers, VMs, lambdas, etc) exactly the same.

Apigee: Users can deploy Apigee onto VMs on-premise via Apigee Edge, but there are strict minimum requirements around hardware, hard disk space, and installation, which may impact cost and resourcing. A typical Apigee Edge installation will consist of Apigee Edge components distributed across multiple nodes. Each install of Apigee Edge enables users to install and configure one or more Edge components to the node. A full feature deployment of Apigee Edge on nodes hosted in a private cloud instance can be data-heavy, complex, and cumbersome. Members of the community recommend splitting production environments into separate nodes in order for Apigee Edge to reliably manage the workloads. Each install of Apigee Edge on any cluster will require a unique license file.

Users can also deploy Apigee Edge Microgateway, a secure HTTP-based message processor for APIs with Apigee Edge. It allows users to extend Apigee Edge's functionality with greater insights and security with a smaller footprint by processing requests and responses to and from backend services securely.

Kong Gateway: Has a flexible deployment model that allows it to be deployed across virtual machines, cloud, and Kubernetes environments. The preferred ways to manage Kong Gateway is through Kong's RESTful API using Kong Konnect or declaratively using Kong deck. Kong encourages the integration and utilization of Kong Mesh and Kong Gateway to expose advanced API capabilities across workloads.

Amazon API Gateway: Users can connect to virtualized workloads from an on-premise network via private endpoints as part of the AWS ecosystem. Users must have an active AWS account in order to use all services. The suggested method to connect APIs to private workloads requires the use of private endpoints, an Application Load Balancer, and other AWS services including Amazon Route 53 and AWS PrivateLink.

Tyk Gateway: Tyk has developed Tyk Gateway to be flexible and agnostic . It can be deployed on any infrastructure and platform including VMs via Tyk Self-Managed system. This is a licensed solution and allows users to set up and use the same configurations and binaries regardless of install type across their environments.

4.1.3. ARM Processor Support

- Gloo Gateway: ●
- Apigee: ○
- Kong Gateway: ●
- Amazon API Gateway: ○
- Tyk Gateway: ●

Gloo Gateway: To support containerized environments, Gloo Gateway supports ARM processors for management plane and control plane applications. Gloo Gateway also offers FIPS-certified builds for ARM processors designed for applications with high security requirements. Support for your ARM images varies with the Istio version and distribution.

Apigee: There is no documentation or guidance from Apigee to run API Gateway on ARM architecture, which implies ARM-based instances are not supported by Apigee.

Kong Gateway: As part of the Kong Gateway Enterprise version, ARM support is available across multiple operating systems including AWS Linux, Debian, and Ubuntu and also includes support for AWS Graviton.

Amazon API Gateway: The AWS API Gateway does not explicitly support ARM architecture outright. In the AWS ecosystem, ARM-based AWS Graviton 2 is used as the main processor for workloads. Recently these same functionalities were made available for AWS Lambda functions, enabling users to configure functions to run ARM/Graviton2 processors.

The Amazon API Gateway can work alongside Lambda functions via HTTP APIs, which can send requests to AWS Lambda functions or to any routable HTTP endpoint.

Tyk Gateway: You can spin up Tyk Gateway via the Tyk Self Management Platform, which minimizes the effort required for users to spin up a Tyk-based infrastructure on Kubernetes. Tyk has clear instructions and requirements around deployment of the Tyk Self Management Platform on Kubernetes and the deployment has also been tested on Linux/Unix-based systems on AMD64 and ARM architectures. Licensing is required.

4.1.4. Serverless Function Support

- Gloo Gateway: ●
- Apigee: ▸
- Kong Gateway: ●
- Amazon API Gateway: ▸
- Tyk Gateway: ○

Gloo Gateway: Gloo Gateway provides comprehensive support for applications built in hybrid-cloud serverless environments. Gloo Gateway supports functions as a service including AWS Lambda. Users can easily migrate functions between clouds through configuration via Gloo Gateway and route requests directly to serverless functions.

For example, Gloo Gateway can route traffic directly to AWS Lambda *without* having to go through an AWS ALB or AWS API Gateway (which are required to communicate with Lambda otherwise). Gloo Gateway integrates with role-delegation and IAM security mechanisms in AWS to provide decentralized and delegated API management for managing calls to Lambda. This typically reduces AWS spend on networking infrastructure not necessary for running Lambda.

Apigee: Cloud Run is a fully-managed compute environment for deploying and scaling serverless containerized microservices built by Google. Users can consume Cloud Run applications from Apigee X. Apigee X works alongside Cloud Run and manages the APIs and provides a facade for backend service APIs.

There are multiple methodologies to connect Cloud Run apps from Apigee X with two primary methods based around Virtual Private Cloud (VPC) peering and Private Service Connect (PSC), both of which consume Internal Cloud Run Apps from Apigee X, which supports both VPC peering and PSC and provides API exposure and target endpoint connectivity.

Kong Gateway: Supports multiple serverless functions including AWS Lambda, Azure Functions, and Apache OpenWhisk, which users can invoke directly from Kong Gateway. Users can extend their serverless functions by combining with other request plugins and integrations to help with security, traffic, and management.

Amazon API Gateway: As part of the AWS ecosystem, Amazon API gateway is integrated with AWS Lambda to offer a consistent familiar experience for developers building AWS serverless based applications. For developers setting up an AWS Lambda proxy, integration is simple, as the API gateway configures the integration request and response for developers and is easily integrated into backend services without required modification.

Scalability of an all-inclusive AWS serverless solution could get costly as application scales and requires multi-region calls.

Tyk Gateway: Tyk supports integration with AWS Lambda – Tyk’s licensed solution can provide API management functionality to AWS Lambda deployments. Tyk users need to modify the AWS Lambda deployment code source to allow traffic through the Tyk Gateway. Support for other serverless functions is limited.

4.1.5. Packaging and Deployment

- Gloo Gateway: ●
- Apigee: ○
- Kong Gateway: ●
- Amazon API Gateway: ○
- Tyk Gateway: ●

Gloo Gateway: Provides simple installation on any Kubernetes cluster as it can be deployed in seconds directly from a Helm chart or gloctl, the Gloo Gateway command line tool. Gloo Gateway is a container-based product with verified images

available for users to install. Users have the option to consume the open source version or utilize the enterprise deployment to access additional UI, security features, and plugins. Leveraging Kubernetes means Gloo Gateway does not need additional databases and becomes much easier to health check and scale. The data plane, based on Envoy proxy, can run on Kubernetes or on VMs.

Apigee: Is part of the Google Cloud ecosystem and users must have a managed service agreement with Google Cloud to use Apigee. Deployment of Apigee is via the Google Cloud console only and incurs additional billing costs with Apigee's 'Pay-as-you-go' model on top of Google Cloud pricing.

Kong Gateway: Kong provides multiple deployment methods for Kong Gateway. Users can choose to run a quickstart script on Docker to run Kong Gateway, or alternatively connect to Kong Konnect, where a global control plane is hosted in the cloud by Kong and users manage the individual data plane nodes within their preferred network.

Kong Gateway has two variations, including an open source package, Kong Gateway (OSS), and Kong Gateway for Enterprise. Kong Gateway OSS contains the basic API gateway functionality and open source plugins where users can manage the open source Gateway with Kong's Admin API, Kong Manager Open Source, or with declarative configuration.

Kong Gateway for Enterprise adds features on top of the OSS version, including federated API management, audit logging, advanced OPA, and more. Users looking to deploy from the Kong Konnect platform will require an enterprise licensing subscription.

Amazon API Gateway: The standard deployment of Amazon API Gateway requires you to create a serverless API via the Lambda function from the AWS Lambda console and then create an HTTP API from the Amazon API Gateway console. Pricing for Amazon API Gateway is transparent and only based on when APIs are in use and for the API calls you receive and the amount of data transferred out.

Users can use Amazon API Gateway on the free-tier of AWS with a cap on the number of API calls and connections per month for up to twelve months. Additional charges can apply based on other AWS services users may consume.

Tyk Gateway: Tyk Gateway is open source and can be easily deployed to different platforms including Kubernetes (Helm Chart), Docker, Ansible, RHEL, and others. The Tyk Gateway does integrate with other open source projects from Tyk.

For enterprise deployments there are two options: Tyk Cloud and Tyk Self Management Platform. Tyk Cloud is a fully managed service that allows API teams to create, secure, publish, and maintain APIs. A free trial is available and pricing is consumption-based on API traffic with four tiers to select from. Tyk Self-Managed allows customers to install the full lifecycle API management solution using on-prem or cloud infrastructure. It offers a two week trial and is licensed based on the number of environments and gateways required.

4.1.6. Event-Driven Architecture

- Gloo Gateway: •
- Apigee: •
- Kong Gateway: •
- Amazon API Gateway: •
- Tyk Gateway: •

Gloo Gateway: Can be combined with Apache Kafka using one of the Kafka HTTP bridges or via the Confluent Cloud REST API to integrate HTTP clients with event-driven architectures and systems. Gloo Gateway can expose Apache Kafka systems over HTTP to external consumers securely, which users can then expose their organizations events stream across to other platforms and applications.

Gloo Gateway provides native integration with serverless technologies like AWS Lambda, where Gloo Gateway can serve as the entrypoint into event-driven architectures based on serverless and event-driven compute.

Apigee: You can manage REST APIs and Async API (event-driven APIs) with Apigee and Google Cloud. Apigee can expose and manage the REST APIs for external consumption.

Apigee also works with Google's Eventarc available on the Google Cloud console. Eventarc lets users build event-driven architectures without needing to customize or operate the underlying infrastructure. Users can build event workflows between the two solutions simultaneously in the Google Cloud console to manage large event-driven architectures.

Kong Gateway: Users can expose REST APIs in front of an event-driven integration and create an asynchronous event flow with an API. It performs as an event-driven, single threaded proxy server architecture where a single thread handles all incoming requests, and operations are handled asynchronously using an event loop. Users with rapidly scaling workloads who use this setup may experience processing limitations, which may impact reliability of their applications.

Kong provides two basic plugins for Apache Kafka based on a forked version of the (experimental) lua-resty-client. The first plugin allows users to transform http requests to Kafka messages and send them to a Kafka broker. The other plugin provides the ability to send request and response logs to Kafka.

Amazon API Gateway: You can create a REST API through a Lambda function and use the Amazon API Gateway to configure and invoke any Lambda functions, making the Amazon API Gateway the entrypoint into event-driven systems based on Lambda functions.

Using API Gateway provides users with a secure HTTP endpoint to invoke your Lambda function and can help manage large volumes of calls to your function by throttling traffic and automatically validating and authorizing API calls. The Amazon API Gateway can also invoke your function synchronously with an event that contains a JSON representation of the HTTP request.

Tyk Gateway: Tyk suggests using technologies like WebSockets to facilitate event-driven communication. Tyk Gateway also has integrations with solutions like RedPanda to help users build their API-based architectures. You must use Tyk's Universal Data Graph (UDG) to read the Kafka data ingested from RedPanda. You can also configure and expose REST APIs via Tyk's Universal Data Graph alongside other APIs into one universal interface.

Tyk's Universal Data Graph is not part of their open source portfolio.

4.1.7. Gateway-to-Gateway Communication

- Gloo Gateway: ●
- Apigee: ○
- Kong Gateway: ●
- Amazon API Gateway: ○
- Tyk Gateway: ●

Gloo Gateway: Gloo Gateway helps users achieve scalability by supporting multiple gateways per control plane and facilitating the creation of new gateways. Some users deploy Gloo Gateway alongside other generic gateways (e.g. AWS API Gateway) and delegating Gloo Gateway to specifically handle Kubernetes traffic.

Gloo Gateway also can be partnered with Gloo Mesh, Solo's service mesh solution to support users, as it can be enabled for each service to allow multi-cluster connections to available resources dynamically.

Apigee: Google recommends utilizing Anthos Service Mesh gateways to Apigee Hybrid installations to manage multiple ingress gateways.

Kong Gateway: Users can run multiple API gateways on Kong Konnect using the Gateway Manager capability. Gateway Manager lets you catalog, connect to, and monitor the status of all control and data plane nodes in a single place. With Kong Konnect, Kong hosts the control plane and users can also use Kong Mesh to enforce API security and governance and improve connectivity between application backend connections.

Amazon API Gateway: Amazon recommends utilizing an API gateway pattern to build complex microservices-based applications with multiple client applications. Amazon API Gateway can expose the backend API endpoints and connect to AWS Lambda functions and other AWS services. To connect with Kubernetes workloads, Amazon API Gateway connects with AWS services including AWS Controller for Kubernetes to create and manage API Gateway resources.

Tyk Gateway: Tyk provides Tyk Multi Data Centre Bridge (MDCB) as a separate licensed extension to the Tyk Control Plane that manages and synchronizes distributed clusters of Tyk API Gateways. You can combine Tyk Dashboard with MDCB to create a single paned control plane that allows you to centrally manage multiple Tyk Gateway clusters whilst keeping Tyk API Gateways highly available. Tyk Control Plane maintains synchronization of configurations across your Tyk deployment via MDCB whilst ‘Worker’ gateways remain running locally reducing latency for API requests.

4.1.8. Multi-Tenancy and Isolation

- Gloo Gateway: ●
- Apigee: ●
- Kong Gateway: ●
- Amazon API Gateway: ●
- Tyk Gateway: ●

Gloo Gateway: Natively integrates with Kubernetes multi-tenancy support, using Kubernetes concepts like namespaces and RBAC. Gloo Gateway can be deployed in a namespace-scoped fashion, allowing users to run many gateway control planes and data planes in a single Kubernetes cluster. This enables “gateway-per-tenant” or even “micro-gateway” deployment topologies.

Users can use Gloo to organize their Kubernetes and Gloo Gateway resources across teams. Gloo allows for multi-tenancy and serves as a boundary for team resources across Kubernetes namespaces and clusters. Gloo Gateway provides comprehensive documentation and examples around tenancy and architecture best practices. Gloo provides boundaries for how Gloo and Kubernetes resources access each other across other namespaces. These boundaries help you manage services, sharing, and security settings across all the teams, or “tenants,” in your organization.

Apigee: Allows users to create multi-tenancy like grouping to manage and build API proxies. Apigee defines an environment as a software environment within an organization for creating and deploying API proxies, whereas an environment group is the basic mechanism for defining the way requests are routed to individual environments. Hostnames are defined for the environment groups and Apigee routes requests to the environments within a group using those hostname definitions. An environment can belong to multiple environment groups before proxies can be deployed within it.

Kong Gateway: Users can configure workspaces in Kong Manager, the graphical user interface for Kong Gateway. Workspaces provide a way to segment Kong entities that are isolated from entities in other workspaces. Kong Manager utilizes the Kong Admin API to administer and control Kong Gateway to enable users to manage all workspaces in one place, operate routes, plugins and services, manage certificates, and perform role-based access control, and more.

Kong Gateway OSS is limited to one workspace whilst Kong Gateway Enterprise allows users to leverage multiple workspaces across different projects to segment services and routes belonging to different upstreams.

Amazon API Gateway: Offers usage plans as part of Amazon API Gateway to enable users to configure who can access deployed API stages and methods or optionally sets the target request rate to start throttling requests. The usage plan uses API keys to identify API clients and who can access the associated API stages for each key.

API keys are alphanumeric string values that you distribute to application developer customers to grant access to API. These API Keys can be used together with Lambda authorizers, IAM roles, or Amazon Cognito to control access to APIs.

Amazon API Gateway can generate API keys. Users can also use Amazon Cognito user pools to secure small multi-tenant applications. They can achieve multi-tenancy design and protect their multi-tenant APIs with a combination of AWS services including Amazon API Gateway, Amazon Cognito, and AWS Lambda.

Tyk Gateway: Users can deploy a multi-tenant approach with Tyk in various ways. With Tyk Self Managed, you can create multiple 'organizations' in the Tyk Dashboard to set boundaries, gatekeep resources, and split gateways across different organizations defined by users.

Users can also utilize a multi-tenant approach built on MDCB, which requires setting up a local gateway cluster per team based on their geographic location. In this setup, every worker cluster belongs to a single Tyk organization and the same organization can have multiple worker data centers. Both organizations will be defined and have data in the control plane, meaning they will share the same infrastructure in terms of configuration and management, but traffic will be processed by different worker data centers, with other infrastructure and specifications and different teams using each organization.

4.2. Scalability and Performance

	Gloo Gateway	Apigee	Kong Gateway	Amazon API Gateway	Tyk Gateway
Auto-scaling	●	●	◐	◑	◐
Load balancing	●	◐	◑	◐	◑
Latency and throughput	●	◐	◑	◑	●
Resource overhead	●	◐	◑	◑	◐
Resiliency	●	◐	◑	◐	●
High performance API					

4.2.1. Auto-Scaling

- Gloo Gateway: ●
- Apigee: ●
- Kong Gateway: ◐
- Amazon API Gateway: ◑
- Tyk Gateway: ◐

Gloo Gateway: As a container-based solution working with Kubernetes, the Gloo Gateway data plane will scale like other containers that contain the application services. Gloo Gateway gives you a set of reusable external authentication resources that use Kubernetes selectors to automatically scale as your policies and workloads grow. It can also manage regular and unexpected variations and spikes in workloads up to thousands of nodes.

Apigee: Apigee users can scale most services running in Kubernetes from the command line or in a configuration override. They can set scaling parameters for Apigee hybrid runtime services in the `overrides.yaml` file. By default, scaling is described at the organization level. You can override the default settings by specifying environment-specific scaling in the `overrides.yaml` file.

Apigee also enables metrics-based scaling, where the runtime can use CPU and application metrics to scale the apigee-runtime pods. The Kubernetes Horizontal Pod Autoscaler (HPA) API uses the `hpaBehavior` field to configure the scale-up and scale-down behaviors of the target service. Metrics-based scaling is not available for any other components in a hybrid deployment.

Kong Gateway: When scaling on Kubernetes, Kong suggests running multiple instances of Kong as Kubernetes supports horizontal scaling. Each instance is configurable from the controller in its pod. These instances are then exposed as a Kubernetes Service, typically of type `LoadBalancer` and instances will have identical configurations, which are updated with new configurations as they arrive via each Kubernetes resource.

Amazon API Gateway: Acts as a proxy to the backend operations that users have configured. Amazon API Gateway will automatically scale to handle the amount of traffic your API receives and does not arbitrarily limit or throttle requests to your backend operations. All requests that are not intercepted by throttling and caching settings in the Amazon API Gateway console are instead sent to your backend operations.

The Amazon API Gateway integrates easily with other AWS services including Amazon EC2, Amazon EKS, and Amazon DynamoDB to build custom auto scaling systems dependent on use cases.

Tyk Gateway: Tyk Gateway instances can horizontally scale according to user requirements. Using a Helm Chart, Tyk can autoscale for up to 3 instances across

Kubernetes installation options with built-in rules for CPU utilization and memory. Users can also define rules for custom metrics inside the Tyk console.

For Tyk Enterprise you can scale the same as the Tyk OSS version with extra functionality available including sharding, deployment of additional gateway groups, and integration with OpenTelemetry.

4.2.2. Load Balancing

- Gloo Gateway: ●
- Apigee: ○
- Kong Gateway: ▸
- Amazon API Gateway: ○
- Tyk Gateway: ●

Gloo Gateway: Is built on top of Envoy and provides several different load-balancing strategies, allowing users to make optimal use of available resources as well as providing high availability and failover semantics. Load balancing in Gloo Gateway can be divided into two main categories: global load balancing and distributed load balancing.

In distributed load balancing, the Gloo Gateway proxy itself determines how loads should be distributed to the endpoints based on knowing the location of the upstream hosts. This includes load balancing based on upstream health, upstream location (e.g. zoneaware routing) using different load-balancing strategies (e.g. round-robin, random, maglev, etc.). In global load balancing, the control plane is responsible for adjusting the load by configuring different parameters such as priority, locality, etc. Weighted load balancing can, for example, be used to support concepts like canary deployments.

All Gloo Gateway load-balancing policies and configurations are deployed as Kubernetes resources, providing out-of-the-box support for infrastructure as code, CI/CD, and DevOps deployment models.

Apigee: Apigee X and Edge has built-in support for load balancing, and failover across multiple backend server instances is provided. For Apigee Hybrid, an Istio Ingress controller hands requests to the Router/Message Processor (RMP) containerized app in the runtime plane.

Kong Gateway: Kong provides multiple ways of load balancing requests to multiple backend services, including the default DNS-based method, and an advanced set of load-balancing algorithms using the Upstream entity.

The DNS load balancer is enabled by default and is limited to round-robin load balancing. The upstream entity has health-check and circuit-breaker functionalities alongside more advanced algorithms like least-connections, consistent-hashing, and lowest-latency.

Amazon API Gateway: Setting up load balancing requests with Amazon API Gateway requires users to configure different AWS services alongside Amazon API Gateway. When integrated with Lambda, AWS API Gateway can manage network scaling easily as Lambda matches the demand. Users can also pair AWS Lambda with AWS Application Load Balancer (ALB) and use the AWS API Gateway to patch any missing features not provided by ALB.

Different AWS instances may require different load balancing set ups. The AWS API Gateway works with multiple AWS services to provide users with the tools and solutions to customize an application network design that fits their environment. The different AWS services integrated may become costly depending on the size and scale of deployments.

Tyk Gateway: Tyk supports native round-robin load-balancing in its proxy. Tyk will rotate requests through a list of target hosts as requests come in. Users can set up load balancing on Tyk Self Managed Platform directly from the Dashboard and customize the Upstream targets and weighting desired. Tyk can also perform load balancing on gPRC traffic using a similar approach.

4.2.3. Latency and Throughput

- Gloo Gateway: ●
- Apigee: ○
- Kong Gateway: ▸
- Amazon API Gateway: ▸
- Tyk Gateway: ●

Gloo Gateway: Gloo Gateway minimizes latency with incremental scaling of resources when deployed in standard Kubernetes environments. Users can easily predict the performance based on the number of CPUs allocated to each environment. Gloo Gateway is based on Envoy, implemented with C++ providing a highly performant, low-latency runtime. Gloo Gateway also gives more predictable latency results at P99 and tail latency.

Performance and request latency can also be optimized through solution features including custom protocols, request bundling, monitoring, and response caching.

Apigee: Apigee provides its users with a ‘Latency Analysis Dashboard’ and ‘Proxy Performance Dashboard’ that alerts users based on any issues your API proxies may be experiencing and helps users see their API Proxy patterns and processing times. There is no benchmark for standardized Apigee performance and Apigee encourages a customized approach to performance testing. Apigee’s performance degrades at P99 and tail latency.

The dashboards are included as part of the Apigee subscription entitlements or as a paid add-on to pay-as-you-go customers of Apigee.

Kong Gateway: Kong Gateway provides a general recommendation and guide on performance characteristics based on typical and expected configuration and traffic plans. Kong states that the design of Kong Gateway is to operate across a variety of deployment environments with no minimum system requirements to operate. Kong Gateway’s maximum throughput is a CPU-bound dimension, and minimum latency is memory-bound. Kong Gateway is also designed to handle large volumes of request traffic and proxying requests with minimal latency with relation to each unique configuration.

Kong Gateway is built on a Lua-based engine. Lua is a scripting language that is slow to run and debug.

Amazon API Gateway: Sends metric data to Amazon Cloud Watch to measure latency. Amazon API Gateway sends data every minute, which is kept for up to 15 months to enable users to analyze and determine their benchmarks. AWS API Gateway lives within its own VPC, which may add additional latency.

Tyk Gateway: Tyk is transparent about Tyk Gateway’s performance and publishes its performance benchmark across different environments. Tyk Gateway is multi-threaded, and as such, will, by default, under load, consume all the CPU made available to it by the host operating system.

4.2.4. Resource Overhead

- Gloo Gateway: ●
- Apigee: ○
- Kong Gateway: ●
- Amazon API Gateway: ▸
- Tyk Gateway: ○

Gloo Gateway: Requires very lightweight resource usage and can be set up in a single or multi cluster configuration. For minimum setup, Gloo Gateway recommends for management cluster nodes - 2vCPU and 8GB memory, and for workload cluster nodes (for a multi cluster setup) - 2vCPU and 8GB memory.

For more robust setups, Gloo Gateway recommends that for management cluster nodes - 2vCPU and 8GB memory and workload cluster nodes (for a multi cluster setup) - 4vCPU and 16GB memory.

Apigee: There are two minimum configurations to set up Apigee Hybrid within clusters either in a stateful node pool or a stateless node pool. Both require 4CPU and 15GB of RAM with variations in storage and disk IOPS requirements.

Kong Gateway: No minimum system requirements to operate. Kong Gateway recommends allowing 500MB of memory allocated per worker process. To access configuration data, Kong Gateway executes a spiky access pattern to its backing database.

Amazon API Gateway: Is a managed service that will scale to meet operational demands of a user's AWS environment and integrations required.

Tyk Gateway: Requires persistent datastore for its operations. MongoDB or PostgreSQL can be used.

4.2.5. Resiliency

- Gloo Gateway: ●
- Apigee: ○
- Kong Gateway: ●
- Amazon API Gateway: ○
- Tyk Gateway: ●

Gloo Gateway: Operators can implement upstream timeouts and retries when experiencing transient network errors. They can determine the maximum duration to handle requests and specify the retry policy for the route where users can say, for a specific error condition, how many times to retry and for how long to try.

Circuit breaker functionality is part of the underlying Envoy proxy. In Gloo Gateway the timeout setting can be useful to avoid your applications from hanging or failing if no response is returned in a specific timeframe.

Apigee: It is possible to retry to a different target server if the response from the original target server is 5XX. However, users cannot configure Apigee to automatically retry in the case of a single endpoint. The I/O timeout for the Message Processor applies to backend servers configured in the target endpoint configuration and is customizable.

Kong Gateway: Whenever an error occurs during proxying, Kong Gateway uses the underlying Nginx retries mechanism to pass the request on to the next upstream. Passive health checks (circuit breakers) look at requests being proxied by Kong, and when a target becomes unresponsive, the passive health checker will detect that and mark the target as unhealthy and the ring-balancer will start skipping this target. Timeouts can be set for multiple variables.

Amazon API Gateway: A retry operation should be implemented on the client side. Implementing the Circuit Breaker Pattern with AWS API Gateway can temporarily stop requests when errors occur. Timeouts of 50 milliseconds - 29 seconds for all integration types, including Lambda, Lambda proxy, HTTP, HTTP proxy, and AWS integrations.

Tyk Gateway: RetryAttempts defines the number of retries that Tyk Gateway should perform during a resource sync (APIs or policies). Tyk has a built-in circuit breaker pattern as a path-based option. The circuit breaker is rate-based and triggers an event. Enables enforced timeouts in API Definitions.

4.2.6. High Performance APIs - Streaming & gRPC

- Gloo Gateway: ●
- Apigee: ○
- Kong Gateway: ●
- Amazon API Gateway: ○
- Tyk Gateway: ●

Gloo Gateway: Gloo supports the full upstream and downstream capabilities of the HTTP/2 & QUIC protocols necessary to enable high-performance end-to-end bidirectional streaming and efficient connection reuse. These features accelerate WebSockets and the responsive web user experiences consumers now expect. Queueing system APIs such as MQTT and libraries like gRPC are also dependent on these protocols for effective deployment within data centers and at the edge. Gloo also supports automatic translation of gRPC to REST and forms suitable for consumption in web browsers. Built on the Envoy proxy, authored in C++.

Apigee: Users of all Apigee product variations can use Websockets for streaming APIs and apply API key and OAuth authentication policies to them; other policy types are not supported. Apigee Hybrid & X products support HTTP/2 allowing for gRPC, though translation to REST is not built in. QUIC support is delivered via integration with GCP's load-balancer infrastructure. Apigee Edge is built on Java, the Hybrid/X products are built on Envoy.

Kong Gateway: Users can leverage support for WebSockets to support streaming, though there is no support for WebSocket over HTTP2 or QUIC to backends. Support for schema-verification of streamed JSON messages is provided by a plugin. gRPC support is also provided, including REST translation and browser consumable forms. Built on the nginx proxy, authored in C with Lua plugins.

Amazon API Gateway: Native support for WebSocket is provided, including the ability to route individual messages based on their content to different backends. HTTP/2 and QUIC frontend support is provided via AWS load-balancer infrastructure. There is no support for HTTP/2 or QUIC to backends, and this limitation also prevents end-to-end gRPC support.

Tyk Gateway: WebSocket and end-to-end gRPC support is built-in through these features. Users can apply a wide range of policies such as authentication and rate-limiting prior to WebSocket stream establishment. Similarly, the set of built-in policies can be applied to gRPC methods. Tyk is authored in Golang.

4.3. Developer Experience and Documentation

	Gloo Gateway	Apigee	Kong Gateway	Amazon API Gateway	Tyk Gateway
API catalog/developer portal	●	●	●	◐	●
DevOps/GitOps friendly	●	●	●	◐	◐
GraphQL	●	◐	◐	◐	●
Service discovery	●	◐	●	◐	●

4.3.1. API Catalog/Developer Portal

- Gloo Gateway: •
- Apigee: •
- Kong Gateway: •
- Amazon API Gateway: •
- Tyk Gateway: •

Gloo Gateway: Uses Kubernetes native resources to discover, document, catalog, and publish APIs. API specifications can be discovered from both Kubernetes services and external locations. These APIs can be combined into API Products that can be exposed via one or multiple Portals. Gloo Gateway supports running multiple Portal (or Portal Servers) on a single Kubernetes cluster, providing native support for multi-tenant deployment models. This can be useful when API Products are exposed to both internal and external consumers.

Gloo Gateway exposes its Portal functionality via a RESTful API. This allows for easy integration with various UI technologies, developer platforms, and third party tools. Gloo Gateway has an out-of-the-box, fully customizable Developer Portal UI, which provides functionalities like an API Catalog, key and credential management, and “try-it-out” functionality to try and test APIs directly from the UI. Integration with the Backstage Internal Developer Platform (IDP) is provided via the Gloo Backstage Plugins. Other platforms and tools can be easily integrated using the aforementioned Gloo Gateway Portal REST APIs.

Cataloging features are available as an API for integrations with third party tools, including a plug-in for Backstage, or for customers to build their own frontend. Gloo Gateway enables publishing, sharing, GitOps calling, and monetization of defined APIs.

Apigee: Apigee offers API producers a developer-focused portal where client application developers can discover APIs, find the documentation required to build applications using them, and register as an app developer to stay in sync with any updates or changes.

Kong Gateway: Kong Developer Portal is packaged into Kong Konnect and provides a single source of truth for developers to locate, access, and consume services and documentation. Users can browse and search API documentation, test API endpoints, and manage their own credentials. There are flexible deployment options to support both internal and external APIs managed from Kong Konnect.

As part of Konnect, API products can be easily published to the Dev Portal and immediately become available to users. When an API product is published, the API specifications and any service documentation become discoverable.

Amazon API Gateway: The Serverless Developer Portal is a developer portal application that allows users to register, discover, and subscribe to API Products via API Gateway Usage Plans, manage their API Keys, and view their usage metrics for your APIs.

After developers build, test, and deploy their APIs, they can package them in an API Gateway usage plan and sell the plan as a Software as a Service (SaaS) product through AWS Marketplace.

Tyk Gateway: The Tyk Enterprise Developer Portal is a straightforward way for API providers to publish, monetize, and drive the adoption of APIs. It provides a full-fledged CMS-like system that enables you to serve all stages of API adoption: from the look and feel customization to exposing APIs and enabling third-party developers to register and use your APIs.

4.3.2. DevOps/GitOps Friendly

- Gloo Gateway: •
- Apigee: •
- Kong Gateway: •
- Amazon API Gateway: •
- Tyk Gateway: •

Gloo Gateway: The API for Gloo Gateway is built with Kubernetes Custom Resource Definitions (CRDs), enabling easy integration with GitOps processes. Configuration artifacts are expressed as YAML in Kubernetes CRDs so they can be stored as artifacts in a git repo. They fit hand-in-glove with GitOps platforms like Flux and Argo Continuous Delivery (ArgoCD) and they can be stored at runtime in native Kubernetes etcd storage. As a result, there are no external databases to configure with Gloo Gateway.

Apigee: Users can build and deploy Apigee proxies using DevOps methods by integrating additional Google services and tools into existing development processes. However, synchronizing all these Google-based services may be difficult and hard to automate across different architectures. Alternatively, users can build GitOps-like deployments on Apigee Hybrid with integrations with ArgoCD and Helmfile depending on your environment.

Kong Gateway: Users can drive a GitOps flow of API design and execution with Kong's deck, which provides a command line interface (CLI) to manage Kong in a declarative way. This allows teams to manage their configuration in Git repositories, manage version control, and leverage GitOps to automate the application of the configuration.

Amazon API Gateway: Uses multiple AWS services to evoke the Amazon API Gateway. The CI/CD pipeline is built using AWS CloudFormation, AWS CodePipeline, AWS CodeBuild, and AWS CodeCommit. The pipeline starts automatically every time you check in your changes into your CodeCommit repository.

Tyk Gateway: Tyk Gateway can be used to deploy applications and associated API updates using a single, streamlined process within a DevOps or GitOps flow. Users can deploy Tyk Operator for Kubernetes environments or Tyk Sync for other environments to integrate with CI/CD processes depending on their architecture.

Tyk Operator is an open source agent deployed on Kubernetes clusters to detect configuration drift between API configurations on Tyk Gateway and the manifest to reconcile. Tyk Sync is a command-line tool for synchronizing API definitions and policies from a Git repository to Tyk.

4.3.3. GraphQL

- Gloo Gateway: •
- Apigee: •
- Kong Gateway: •
- Amazon API Gateway: •
- Tyk Gateway: •

Gloo Gateway: In Gloo Gateway, GraphQL is built in with no external GraphQL servers required. It is also accessed through Gloo Gateway, and natively integrated with other capabilities including ExtAuth and RateLimiting functionality.

The implementation approach allows you to create GraphQL APIs without having to implement a single line of code. Gloo Gateway also supports schema stitching to access multiple schemas together in a single request.

Apigee: Natively supported GraphQL APIs enable users to productize and manage their API lifecycles in Apigee. The GraphQL policy can parse GraphQL payloads into message flow variables, verify GraphQL requests against a schema, or both.

Kong Gateway: Kong integrates with existing GraphQL infrastructure out of the box (Apollo GraphQL server), but only in the licensed enterprise version. Kong provides enterprise-grade proxy caching and rate limiting specifically tailored for GraphQL.

Amazon API Gateway: AWS AppSync offers simplified data access and querying, powered by GraphQL, including serverless WebSockets for GraphQL subscriptions. Amazon API Gateway can be used as a proxy for the GraphQL requests to AppSync and it supports rate limiting and installation of SSL certificates. This is a two-step approach to adding GraphQL to the Amazon API Gateway.

Tyk Gateway: Tyk supports GraphQL natively and does not need to use any external service or process for any GraphQL middleware. You can securely expose existing GraphQL APIs using the GraphQL core functionality. In addition to this, you can also use Tyk's integrated GraphQL engine to build a Universal Data Graph(UDG) which exposes existing services as one single combined GraphQL API.

4.3.4. Service Discovery

- Gloo Gateway: •
- Apigee: ◉
- Kong Gateway: •
- Amazon API Gateway: ◉
- Tyk Gateway: •

Gloo Gateway: A route destination is a single Gloo Gateway Upstream. It's also possible to route to multiple Upstreams, by either specifying a multi destination, or by configuring an Upstream Group. It is also possible to route directly to Kubernetes or Consul services, without needing to use Gloo Edge Upstreams or discovery.

When routing to an Upstream, you can take advantage of Gloo Edge's endpoint discovery system, and configure routes to specific functions, such as a REST endpoint, a gRPC service, or a cloud function like AWS Lambda

Apigee: There is limited information available on Apigee and service discovery capabilities. Requests for a service discovery feature for Apigee from the community are common, with some solutions proposed from the community including calling Eureka and/or integrating Istio as a Service Mesh with Apigee.

Kong Gateway: With the Kong API gateway, client-side discovery is achieved using a ring balancer. Server-side service discovery is performed by implementing a DNS server.

Amazon API Gateway: Limited information around service discovery for Amazon API Gateway. Available materials online discuss customizing multiple AWS services together, including Amazon ECS service discovery, AWS Cloud Map, and Amazon Route 53 to implement service discovery on your architecture.

Tyk Gateway: Provides a service discovery feature, but requires Tyk Gateway to be manually reconfigured or detect the failure and reconfigure itself. Tyk recommends using the service discovery module in conjunction with the circuit breaker features, as this makes the detection and discovery of failures at the gateway level much more dynamic and responsive.

4.4. Security and Governance

	Gloo Gateway	Apigee	Kong Gateway	Amazon API Gateway	Tyk Gateway
Role-based access control (RBAC)	●	◐	●	◐	●
External authentication and authorization	●	●	●	●	●
Traffic management	●	◐	◐	◐	◐
Encryption	●	◐	●	◐	◐
Rate limiting	●	◐	●	◐	◐
Advanced security	●	◐	◐	◐	◐
Secrets integration	◐	◐	●	◐	◐

4.4.1. Role-Based Access Control (RBAC)

- Gloo Gateway: •
- Apigee: •
- Kong Gateway: •
- Amazon API Gateway: •
- Tyk Gateway: •

Gloo Gateway: To control user access to resources, Gloo Gateway utilizes the RBAC capabilities of Kubernetes to ensure a granular and secure approach to access management. This will allow administrators to define roles, role bindings, and cluster roles to specify which users or entities have permission to perform specific actions in the cluster. This control enhances Gloo Gateway security by limiting access to only the necessary resources and actions within the Kubernetes environment.

Apigee: Managing user access on Apigee can be done multiple ways, including using APIs, through the Apigee UI, from the Google Cloud console's IAM service. Apigee provides pre-defined curated roles across organizations, list environments, and projects.

Kong Gateway: Kong Gateway lets you manage and configure user authorization using workspaces and teams in Kong Gateway with Kong Manager. Securing Kong Gateway requires users to turn on RBAC and create a workspace and an admin for segregated administration.

Every administrator using Kong Manager needs an assigned role based on the resources they have permission to access. As the super admin (or any role with read and write access to the /admins and /rbac endpoints), it is possible to create new roles and customize permissions. There are three default roles read-only, admin, and super-admin in Kong.

Amazon API Gateway: Must use the AWS Identity and Access Management (IAM) service to securely control access to all AWS resources, including Amazon API Gateway. IAM administrators control who can be authenticated (signed in) and authorized (have permissions) to use the gateway resources.

Control access in AWS is managed by creating policies and attaching them to AWS identities or resources. Most policies are stored in AWS as JSON documents. By default, users and roles have no permissions in AWS.

Tyk Gateway: Tyk can enable RBAC with User Groups via the Tyk Dashboard which, is multi-tenant capable and allows granular, role-based user access.

Users can be assigned specific permissions to ensure that they only have very specific access to the Dashboard pages, and to the underlying API. Users groups can also be established and assigned to one or more users. This also works for Single Sign On (SSO) by specifying the group ID when setting up SSO. The availability of these features varies depending on the license or subscription.

4.4.2. External Authentication and Authorization

- Gloo Gateway: •
- Apigee: •
- Kong Gateway: •
- Amazon API Gateway: •
- Tyk Gateway: •

Gloo Gateway: Gloo Gateway provides an ext-auth server implementation that provides an out-of-the-box authentication mechanism, like OIDC, API, along with basic authentication capabilities and is able to integrate with legacy systems providing older cryptography mechanisms.

Gloo Gateway can also use a custom authorization mechanism if needed. Gloo Gateway supports multiple authorization mechanisms out of the box to integrate with API keys, JSON web tokens (JWT), lightweight directory access protocol (LDAP), OAuth, OpenID Connect (OIDC), Open Policy Agent (OPA), and custom services.

Apigee: Apigee Integration supports the following authentication types: Auth token, Google OIDC ID Token, JSON Web Token (JWT), OAuth, Google Service Account, and SSL/TLS client certificates.

Kong Gateway: Kong Key Auth plugin lets you add API key authentication to a service or a route. Consumers then add their API key either in a query string parameter, a header, or a request body to authenticate their requests.

Support for JWT, KeyAuth, HMAC, LDAP, OpenID Connect, and SAML. The plugin implementation of OAuth2 in Kong is for experimental or testing purposes only and is not recommended for production environments.

Amazon API Gateway: Authorize access to your APIs with AWS Identity and Access Management (IAM) and Amazon Cognito. If you use OAuth tokens, API Gateway offers native OIDC and OAuth2 support. To support custom authorization requirements, you can execute a Lambda authorizer from AWS Lambda.

Tyk Gateway: Tyk Identity Broker (TIB) is a component providing a bridge between various Identity Management Systems such as LDAP, Social OAuth (e.g. GPlus, Twitter, GitHub), or Basic Authentication providers, to your Tyk installation. Industry Standard Authentication: OIDC, JWT, bearer Tokens, Basic Auth, Client Certificates, and more. Open API Standards: Import your Swagger and OpenAPI Documents (OAS 2.X and OAS 3.0.1) to scaffold APIs in Tyk.

4.4.3. Traffic Management

- Gloo Gateway: •
- Apigee: •
- Kong Gateway: •
- Amazon API Gateway: •
- Tyk Gateway: •

Gloo Gateway: There are four main areas of traffic management in Gloo Gateway: gateways, virtual services, routes, and upstreams. Gloo Gateway listens for incoming traffic on Gateways where the definition includes the protocols and ports on which it listens for traffic.

Virtual Services are bound to a gateway and configured to respond for specific domains containing a set of route rules, security configuration, rate limiting, transformations, and other core routing capabilities.

Routes are associated with Virtual Services and direct traffic based on characteristics of the request and the upstream destination. Routes then send traffic to destinations, called Upstreams.

Upstreams take many forms, including Kubernetes services, AWS Lambda functions, or Consul services. Traffic management features include basic routing (HTTP, TLS, TCP, UDP, gRPC), Global service routing, HTTP Redirects and rewrites, HTTP Traffic Splitting, and priority failover routing.

Apigee: Apigee policies control the flow of request and response messages to provide features like transformation and mediation capabilities. Support for custom scripts and code to extend API proxy functionality.

Kong Gateway: Kong Gateway listens for HTTP traffic on its configured proxy port(s) and L4 traffic on explicitly configured stream_listen ports. Kong Gateway will evaluate any incoming HTTP request or L4 connection against the routes you have configured and try to find a matching one.

If a given request matches the rules of a specific route, Kong Gateway will process proxying the request. Because each route may be linked to a service, Kong Gateway will run the plugins you have configured on your route and its associated service, and then proxy the request upstream. You can manage routes via Kong Gateway's Admin API. Kong Gateway supports routing by arbitrary HTTP headers.

Amazon API Gateway: API Gateway manages traffic with throttling so that backend operations can withstand traffic spikes. Improves the performance of APIs and the latency by caching the output of API calls to avoid calling the backend every time.

Tyk Gateway: Tyk has a built in quota and rate limiting mechanism to ensure that your APIs are secure and so that you can manage and monetize traffic to and from your APIs.

It is possible to modify inbound and outbound body data and header information on the fly using Tyk. This can either be done using the scriptable middleware, or can be achieved using dedicated middleware.

Tyk Dashboard will show you an overview of the aggregate usage of your APIs; this view includes the number of hits, the number of errors and the average latency over time for all of your APIs as an average

4.4.4. Encryption

- Gloo Gateway: ●
- Apigee: ◉
- Kong Gateway: ●
- Amazon API Gateway: ●
- Tyk Gateway: ▶

Gloo Gateway: FIPS 140-2 certified by a NIST laboratory to meet strict security standards, and is in process of getting FIPS 140-3 certification. It also supports mTLS connections between services.

Apigee: By default, the following data is stored encrypted in the Apigee hybrid runtime plane: Key management system (KMS) data, key-value map (KVM) data, and cache data. Data encryption does not require any special configuration.

Uses one-way TLS to secure API proxy endpoints on the ingress gateway, or configure mTLS on the ingress gateway. However, upstream calls from the gateway to services are NOT using mTLS.

Kong Gateway: Kong Gateway Enterprise features a self-managed FIPS 140-2 gateway package. This provides a FIPS mode, which at its core uses the FIPS 140-2 compliant BoringCrypto for cryptographic operations. It also supports mTLS connections.

Amazon API Gateway: API Gateway doesn't support unencrypted (HTTP) endpoints. API Gateway manages the certificates for default execute-api endpoints. For greater security, choose a minimum Transport Layer Security (TLS) protocol version.

Tyk Gateway: Tyk Gateway can use mTLS for gateway-upstream and client-gateway connections. Cloud users can secure their upstream services with mTLS but mTLS between the client (caller of the API) and Tyk's gateway cannot be done.

4.4.5. Rate Limiting

- Gloo Gateway: •
- Apigee: •
- Kong Gateway: •
- Amazon API Gateway: •
- Tyk Gateway: •

Gloo Gateway: Gloo Gateway offers a comprehensive rate-limiting feature and built-in quota management. When utilizing a rate limit server that connects to Memcache, Redis, or Elastic Cache, Gloo Gateway enforces global API limits, supporting advanced rate limiting for custom policies to address complex scenarios.

Rate limit reusability extends to both cluster ingress and service mesh traffic, enabling the application of policies to inbound cluster traffic ("north-south") and services within the mesh ("east-west").

Apigee: Apigee provides two policies that enable you to optimize traffic management to minimize latency for apps. The SpikeArrest policy protects against traffic surges by limiting the number of requests processed by an API proxy and sent to a backend.

Quota policy enforces consumption limits on client apps by maintaining a distributed 'counter' that tallies incoming requests. Rate plans can be monetized and managed with Apigee.

Kong Gateway: Kong Rate Limiting plugin allows users to rate limit how many HTTP requests can be made in a given period of seconds, minutes, hours, days, months, or years. If the underlying service or route has no authentication layer, the Client IP address is used. Otherwise, the consumer is used if an authentication plugin has been configured.

The advanced version of this plugin, Rate Limiting Advanced, provides the ability to apply multiple limits in sliding or fixed windows.

Amazon API Gateway: Throttling (rate limiting) is done on the per second level via usage plans and API keys. It has no concurrency limit on requests, meaning no limits for existing or open requests. Developers can set the target limits for individual API stages or methods to improve overall performance across all APIs.

Tyk Gateway: Distributed Rate Limiter (DRL) is the default rate limiter in Tyk. It is the most performant, and the trade-off is that the limit is approximate, not exact. The Redis rate limiter is the less performant, exact rate limiter for Tyk. The DRL rate limiter will be used automatically unless one of the other rate limit algorithms are explicitly enabled via configuration.

4.4.6. Advanced Security

- Gloo Gateway: •
- Apigee: ▶
- Kong Gateway: •
- Amazon API Gateway: ◉
- Tyk Gateway: ▶

Gloo Gateway: Gloo Gateway has a built-in WAF solution, providing IP restriction, request filtering, and many other WAF capabilities. Gloo Gateway uses TLS/mTLS encryption and integration with OPA. Gloo Gateway also has a built-in data loss prevention filter, masking any sensitive data from being returned and to obfuscate user data in logs. Gloo Gateway also monitors for data breaches or exfiltration to prevent data loss and data leaks.

Member of the vendor consortium, which provides early access to Envoy CVEs before they are made public. Every critical CVE is proactively fixed, each release is scanned with Snyk before release and results are published. Adheres to a process for

maintaining the compliance requirements around FedRAMP where customers need to have CVEs triaged every month.

Apigee: Designed to work with any directory service that supports LDAP, such as Active Directory, OpenLDAP, and others. Apigee has a key manager to store secret key encryption IDs.

Kong Gateway: Kong Gateway provides a mechanism to store sensitive data fields, such as consumer secrets, in an encrypted format within the database. This functionality provides transparent, symmetric encryption of sensitive data fields at rest. Transparency refers to the fact that, when enabled, encryption/decryption of data is done on-the-fly by Kong immediately before writing/immediately after reading from the database.

Amazon API Gateway: Requires use of AWS Key Management Service, which never interacts directly with your external key manager, and cannot create, view, manage, or delete your keys. Instead, AWS KMS interacts only with external key store proxy (XKS proxy) software that you provide.

Tyk Gateway: Tyk initially does not insist on signing any cluster messages or middleware bundles. If you are moving to production, Tyk strongly recommends enabling payload signatures. Payload signatures can be enabled in `tyk.conf` by setting `allow_insecure_configs` to `false` and then setting up a public / private keypair.

4.4.7. Secrets Integration

- Gloo Gateway: •
- Apigee: •
- Kong Gateway: •
- Amazon API Gateway: •
- Tyk Gateway: •

Gloo Gateway: Can manage sensitive credentials like passwords, tokens, and keys with Kubernetes & Hashicorp Vault. Also integrates with AWS Lambda and Azure, utilizing secrets for authentication, configuration of SSL certificates.

Apigee: Apigee integrates with GCP Secret Manager through the External Secrets Operator, facilitating secret management with features like authentication and workload identity. Kubernetes secrets can be provisioned from Azure KV or Hashicorp Vault.

Kong Gateway: Kong Gateway Enterprise offers out-of-the-box secrets management with the following backends: Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), and Hashicorp Vault.

Amazon API Gateway: Amazon has its own AWS Secrets Manager, but no direct integration with the API Gateway is documented.

Tyk Gateway: Supports storing secrets in key value (KV) systems such as Vault and Consul. Tyk also enables the local secrets section inside config store systems.

4.5. Flexibility and Customization

	Gloo Gateway	Apigee	Kong Gateway	Amazon API Gateway	Tyk Gateway
Ease of use	●	◐	●	◐	◑
Out-of-the-box integrations	●	◑	●	◑	●
Customization	◑	◐	◐	◑	◐
Service mesh support	●	◑	●	◑	◐

4.5.1 Ease of Use

- Gloo Gateway: ●
- Apigee: ◐
- Kong Gateway: ●
- Amazon API Gateway: ◐
- Tyk Gateway: ◑

Gloo Gateway: Installation is simple via a Helm chart. Operations are managed through Kubernetes and to some degree automated (if pod fails, etc). Discovery of Kubernetes and Lambda services.

Apigee: The Apigee UI is a containerized app hosted on the management plane. Self-service is done through the Apigee UI and APIs. The different options between Apigee, Apigee Hybrid, and Apigee X can make configuration and deployment of Apigee difficult.

Kong Gateway: A quickstart script is provided to quickly run Kong Gateway and its supporting database. Kong Manager is the graphical user interface (GUI) for Kong Gateway, including a dashboard for all the workspaces in the cluster. It uses the Kong Admin API under the hood to administer and control Kong Gateway.

Amazon API Gateway: Fully managed service with no minimum fees or startup costs. You pay for the API calls you receive and the amount of data transferred out. A tiered pricing model is also available. Full API management experience requires additional integration with other AWS services.

Tyk Gateway: Installation is easy with Tyk offering 10-minute installation scripts for multiple platforms including Docker, Kubernetes, Red Hat, Ubuntu, and more. Tyk Dashboard can be integrated with Tyk Gateway as GUI and analytics platform. Feature restrictions between Tyk open source portfolio and Tyk closed source portfolio may be a roadblock for users.

4.5.2 Out-of-the-Box Integrations

- Gloo Gateway: •
- Apigee: •
- Kong Gateway: •
- Amazon API Gateway: •
- Tyk Gateway: •

Gloo Gateway: Enables customization through plugins and extensions to implement new configurations and policies. Supports integration with HashiCorp Vault. Leverages WebAssembly (Wasm) for customizing Envoy. Utilizes OpenTelemetry (Otel) to seamlessly integrate with leading backend and telemetry providers.

Apigee: Apigee Integration extends the Apigee API Management platform to include core integration features – connectors, an integration engine, and data transformation tools. Apigee Integration offers the ability to connect with various applications so that they collaborate efficiently and function as one unit. Use out-of-the-box triggers, configurable tasks, and the user interface to create enterprise-level integrations.

Kong Gateway: Kong Plugin Hub catalogs plugins and integrations to extend Kong Gateway, supporting hundreds of out-of-the-box plugins to address key use cases.

Amazon API Gateway: API Gateway acts as a "front door" for applications to access data, business logic, or functionality from backend services, such as workloads running on Amazon EC2, code running on AWS Lambda, any web application, or real-time communication applications.

Tyk Gateway: Tyk has multiple integration options with third parties, from plugins, externally to the gateway using a broker, or using built-in federation support via JSON Web Tokens or Open ID Connect.

4.5.3 Customization

- Gloo Gateway: ◉
- Apigee: ▶
- Kong Gateway: ▶
- Amazon API Gateway: ◉
- Tyk Gateway: ▶

Gloo Gateway: Kubernetes-native CRDs allow for flexibility with orchestration. Webassembly allows for customization of Envoy and the proxy layer.

Apigee: Create API specifications by describing your requirements in natural language in Apigee API Design Assistant, a plugin integrated into Cloud Code.

Kong Gateway: Kong Gateway is a Lua application running in Nginx. Kong Gateway is distributed along with OpenResty, which is a bundle of modules that extend the lua-nginx-module.

Amazon API Gateway: Create access policies for resources, actions, and effects by principal users.

Tyk Gateway: Tyk is written in pure Go. Open Policy Agent (OPA) enables users to add custom permissions.

4.5.4 Service Mesh Support

- Gloo Gateway: ●
- Apigee: ◉
- Kong Gateway: ●
- Amazon API Gateway: ◉
- Tyk Gateway: ▶

Gloo Gateway: Has been primarily created for ingress use cases, the Kubernetes Gateway API is flexible enough to also adapt to intra-cluster traffic (service mesh). Gloo Gateway cleanly integrates with the service mesh functionality of Gloo Mesh, which is built using Istio.

Apigee: Works with Anthos Service Mesh. Google has been quick to support the Istio open source service mesh, which Apigee has announced as a key part of its roadmap.

Kong Gateway: Kong Ingress Controller has been tested to support Kong Mesh and Istio.

Amazon API Gateway: Limited integration with AWS AppMesh. No direct documentation found on linking the two AWS application networking services.

Tyk Gateway: Tyk works by running inside a service mesh as the sole ingress. Tyk puts API management as a higher-level abstraction in front of all the network-level activities that the service mesh handles. Tyk can be configured to run ingress use cases with examples on GitHub for an Istio ingress gateway.

4.6. Observability and Monitoring

	Gloo Gateway	Apigee	Kong Gateway	Amazon API Gateway	Tyk Gateway
Logging	●	◐	●	●	●
Tracing	●	◐	◐	◐	●
Metrics and analytics	●	◐	◐	◐	●

4.6.1 Logging

- Gloo Gateway: ●
- Apigee: ▶
- Kong Gateway: ●
- Amazon API Gateway: ●
- Tyk Gateway: ●

Gloo Gateway: You can enable logging during the Helm installation. The access log entries can be customized to include data from the request, the routing destination, and the response. The proxy can be configured to output multiple access logs with different configurations. Logging data is available using OpenTelemetry.

Apigee: A data collector resource enables you to collect a wide variety of custom data from API traffic. Once you have created a data collector, you specify the data you want to gather using the DataCapture policy.

Kong Gateway: Available log levels include: debug, info, notice, warn, error, and crit. Notice is the default and recommended log level. However, if the logs turn out to be too chatty, they can be bumped up to a higher level like warn. Kong Gateway audit logging also provides granular logging of the Admin API to keep detailed track of changes made to the cluster configuration throughout its lifetime.

Amazon API Gateway: Amazon CloudWatch Logs enables execution logging and access logging. The logged data includes errors or execution traces (such as request or response parameter values or payloads), data used by Lambda authorizers (formerly known as custom authorizers), whether API keys are required, whether usage plans are enabled, and more.

Tyk Gateway: Tyk will record detailed usage data on who is using your APIs (raw data only) and try to output structured logs and will include context data around request errors where possible. Log data is usually of the Error level and higher. Logging verbosity via an Environment Variable to affect all Tyk components or just for a specific gateway. The logger supports multiple back-ends, including Sentry, Logstash, Graylog, and Syslog.

4.6.2 Tracing

- Gloo Gateway: ●
- Apigee: ▶
- Kong Gateway: ▶
- Amazon API Gateway: ▶
- Tyk Gateway: ●

Gloo Gateway: By enabling OpenTelemetry (OTel) tracing capabilities, users can obtain visibility and track requests as they pass through the API gateway to distributed backends. OTel provides a standardized protocol for reporting traces, and a standardized collector through which to receive metrics. Additionally, OTel supports exporting metrics to several types of distributed tracing platforms. Zipkin or Jaeger can be used to facilitate root cause analysis of issues across the system made available using OTel.

Apigee: Trace is a tool for troubleshooting and monitoring API proxies running on Apigee Edge. Trace lets you probe the details of each step through an API proxy flow.

Kong Gateway: Kong provides a set of core instrumentations for tracing. The tracing API will detect the propagation format from the headers, and will use the appropriate format to propagate the span context. If no appropriate format is found, then will fallback to the default format, which can be specified. The propagation API works for both the OpenTelemetry plugin and the Zipkin plugin.

Amazon API Gateway: Amazon API Gateway provides active tracing support for AWS X-Ray. Enable active tracing on your API stages to sample incoming requests and send traces to X-Ray.

Tyk Gateway: Tyk currently supports OpenTelemetry and OpenTracing for distributed tracing.

4.6.3 Metrics and Analytics

- Gloo Gateway: •
- Apigee: •
- Kong Gateway: •
- Amazon API Gateway: •
- Tyk Gateway: •

Gloo Gateway: A unified telemetry pipeline based on OpenTelemetry collects telemetry data including metrics, logs, and tracing across a multi-cluster environment. Envoy-based metrics for data plane performance allow for deep instrumentation to performance and scale characteristics of API traffic, including golden signals and vital SLI/SLO metrics for platform teams. API metadata is included in metrics capture to provide rich insights for API analytics teams to determine API adoption and use trends. API analytics are also used for usage reporting and as a data source for API monetization. Metrics are collected by default in Prometheus with visual reporting available through out-of-the-box Grafana dashboards with the ability to customize as needed. The telemetry pipeline can be

easily connected to third-party observability services (Splunk, Datadog, etc.) with existing OpenTelemetry exporters.

Apigee: Metrics are managed by a single Prometheus server per cluster for all services. A data collection pod in the runtime plane uses fluentd and UDCA (Universal Data Collection Agent) to gather analytics and feed the data to the UAP (Unified Analytics Platform) in the management plane.

Kong Gateway: Enterprise users can utilize Kong Konnect, which has built-in analytics features helping you manage and track API product, route, or applications performance managed by Kong. Provides plugins for AppDynamics, Datadog, OpenTelemetry, Prometheus, and Zipkin.

Amazon API Gateway: Monitor performance metrics and information on API calls, data latency, and error rates from the API Gateway dashboard, which allows you to visually monitor calls to your services using Amazon CloudWatch. Additional product integrations include: Amazon CloudWatch Logs, Amazon CloudWatch Alarms, Access Logging to Kinesis Data Firehose, AWS CloudTrail, and AWS X-Ray.

Tyk Gateway: Tyk offers built-in metrics and analytics in Tyk Dashboard through Tyk API Gateway and Tyk Pump. The built-in metrics allow you to track overall API traffic, detailed API analytics including: request count, response time distribution, and error rates. Users can export those metrics to different back-ends including Prometheus and Grafana.

4.7. Supportability

	Gloo Gateway	Apigee	Kong Gateway	Amazon API Gateway	Tyk Gateway
Product documentation	●	◐	●	◐	●
Enterprise support	●	◐	●	◐	●
Community support	●	◐	◐	◐	◐
Long-term version support	●	◐	◐	○	●

Health checks	●	◐	◑	◒	◓
---------------	---	---	---	---	---

4.7.1 Product Documentation

- Gloo Gateway: ◐
- Apigee: ◒
- Kong Gateway: ◐
- Amazon API Gateway: ◓
- Tyk Gateway: ●

Gloo Gateway: Clear documentation includes examples and videos. Some minor product alignment across the documentation required. Good additional resources including Solo Academy and other learning resources are also available online. Slack offers a direct communication channel if users have questions.

Apigee: Complex product differentiation that is compounded by difficult to reference documentation. Challenging to search and navigate, includes some examples and videos.

Kong Gateway: Clear online documentation that guides users between the open and enterprise add-ons of Kong. Good examples and deeper technical examples from blogs linked from documentation.

Amazon API Gateway: Clear online documentation. Limited use case and configuration examples. Documentation primarily at the fundamental level.

Tyk Gateway: Clear online documentation between open source and closed sourced portfolios. Easy to follow examples and instructions across the entire portfolio suite. Additional technical examples available through videos and blog resources. Good community online forum for users to support each other.

4.7.2 Enterprise Support

- Gloo Gateway: ◐
- Apigee: ◓
- Kong Gateway: ●
- Amazon API Gateway: ◓
- Tyk Gateway: ●

Gloo Gateway: Published SLAs providing assurances that issues are responded to in a timely manner. Expert and community support available on Slack. Does not offer 24x7 worldwide coverage for enterprise support.

Apigee: Three tiers of technical support available: standard, enhanced premium, and comprehensive GCP support. Target response times vary depending on support level – there are not defined SLAs. Community forum board is active with support and responses available from Google technical staff. Additional value add services like Google assured support are not available for Apigee.

Kong Gateway: Kong will aim to produce patches for all applicable Kong Gateway versions currently under support within the SLA. Kong Konnect has two subscription levels: Plus and Enterprise. Enterprise provides 24x7x365 technical support and access to professional services. Kong Nation is Kong's community discussion board with free access to community support.

Amazon API Gateway: Standard AWS support options: Basic, Developer, Business, Enterprise On-Ramp, and Enterprise. Strong forum and discussion board for community based support.

Tyk Gateway: Tyk Cloud has different pricing tiers depending on user requirements. Technical support is included across all tiers. Standard does not have designated support hours whereas Premier and Signature Gold (both add-ons) have 24x7x365 SLA support hours. An online community forum is available to everyone, offering multiple support channels.

4.7.3 Community Support

- Gloo Gateway: ●
- Apigee: ◉
- Kong Gateway: ▶
- Amazon API Gateway: ◉
- Tyk Gateway: ▶

Gloo Gateway: Developed following open source standards and utilizing open source components such as Kubernetes and the Kubernetes Gateway API, Envoy, and OpenTelemetry. Solo.io is also a major contributor to popular open source tools like Istio and Cilium. All open source components have strong CNCF community support.

Gloo Gateway features an open source version, and community support is accessible through Slack. Solo also provides free technical training for the open source community on application networking topics.

Apigee: Apigee is not open source software, but integrates with some open source projects like Istio. Users utilize the Google Cloud forum for community support, which is occasionally answered by Google Technical Staff. Limited articles are available on the Google Cloud community website posted by Google staff.

Kong Gateway: Based on Nginx and the lua-nginx-module (specifically OpenResty). Kong Gateway is also available as an open source package containing the basic API gateway functionality and open source plugins. The recent forking of the Nginx and creation of the 'freenginx' project may have downstream implications for Kong in the future.

Kong Nation is their community discussion board and brings together developers, contributors, and technology leaders to share resources, knowledge, and best practices for building and deploying APIs.

Amazon API Gateway: Amazon API Gateway is a fully managed service. There are community Q&A forums available. AWS blogs have a good library of technical resources for users to research and apply to their AWS environments.

Tyk Gateway: Tyk Gateway is an open source API Gateway. It is part of their open source portfolio of services. Tyk has an active community forum for customers to use and get community support.

4.7.4 Long-Term Version Support

- Gloo Gateway: ●
- Apigee: ▶
- Kong Gateway: ●
- Amazon API Gateway: ○
- Tyk Gateway: ●

Gloo Gateway: Solo supports n-3 versions for Gloo Gateway. Within each version, different open source project versions are supported, including n-4 version support for the Solo distribution of Istio – not community Istio. The supported version of Istio, and Kubernetes or OpenShift are dependent on each other.

Apigee: A maximum of three versions of a given Kubernetes platform are supported. At least one version of the Kubernetes platform overlaps with the previous minor release of Apigee hybrid.

Platform versions are supported for at most three hybrid minor releases (for example, v1.9, v1.10, and v1.11) or until that version is no longer supported (EOL) by the vendor, whichever comes first.

Kong Gateway: Kong introduces major functionality and breaking changes by releasing a new major version. There is no regular release cadence of major versions.

Kong aims to release a new minor version every 10 weeks. Minor versions contain features and bug fixes. Minor versions are backwards compatible within that major version sequence. Every minor version is supported for a period of 1 year from date of release.

Kong may designate a specific minor version as a Long-Term Support (LTS) version. Kong provides technical support for the LTS version on a given distribution for the duration of the distribution's lifecycle, or for 3 years from LTS version release, whichever comes sooner.

Amazon API Gateway: There is no documentation found with regards to Amazon API Gateway's long term support or version support.

Tyk Gateway: Long term service releases are scheduled for Q1 release annually. An LTS release at Tyk is left on production for 3 months where an elevated period of support is available, patches are run based on need and criticality, and single fix patching is done whenever needed.

This is followed by full support for 12 months. This means that the release will be on full support for 15 months. With a new LTS release, the previous version remains in full support for a further 3 months before it moves into extended support for 12 months after the full support end date.

4.7.5 Health Checks

- Gloo Gateway: ●
- Apigee: ●
- Kong Gateway: ●
- Amazon API Gateway: ○
- Tyk Gateway: ▶

Gloo Gateway: Gloo Gateway includes an HTTP health checking plug-in that you can enable in a Gateway (which becomes an Envoy Listener). This plug-in responds to health check requests directly with either a 200 OK or 503 Service Unavailable message, depending on the current draining state of Envoy. Users can also add health checks that periodically assess the readiness of Upstream to receive requests.

Apigee: Apigee exposes health checks at different levels, which you can leverage depending upon the use case. These include:

- *Regional-level / Apigee instance health check:* Apigee returns the health of overall Apigee instance in a region.
- *Environment-level health checks:* Returns the health of a particular environment in the Apigee instance.
- *Custom health check through an API proxy:* For complex use cases, you can configure a dedicated API proxy as a custom health check endpoint.

Kong Gateway: Kong supports two kinds of health checks, which can be used separately or in conjunction: active checks, where a specific HTTP or HTTPS endpoint in the target is periodically requested and the health of the target is determined based on its response, and passive checks (also known as circuit breakers), where Kong analyzes the ongoing traffic being proxied and determines the health of targets based on their behavior responding to requests.

Amazon API Gateway: Users can use Amazon Route 53 health checks to control DNS failover from an API Gateway API in a primary AWS Region to one in a secondary Region.

Tyk Gateway: Liveness health check is implemented as per the Health Check Response Format for HTTP APIs. The Health check endpoint will refresh every 10 seconds.

5. About the Author

Established in 2017 in Cambridge, MA, Solo.io is a prominent leader in cloud-native application networking. Our expertise lies in seamlessly connecting, securing, and monitoring Kubernetes applications and APIs on a global scale. Our renowned products – Gloo Gateway, Gloo Mesh, and Gloo Network for Cilium – are the preferred choice for Fortune 2000 industry leaders and cloud-native innovators alike, representing market leadership across industries and geographies.

In the rapidly evolving cloud-native landscape, Solo.io leads with solutions that redefine application development's speed and agility. We are dedicated to streamlining and securing cloud-native services and empowering next-generation digital experiences, particularly in generative AI (genAI). With Solo.io, you're equipped to seamlessly and securely expand in a multi-cloud world, ensuring your applications not only keep pace with technological advancements, but lead the way in innovation.

Gloo Gateway is a cloud-native API gateway designed to simplify and optimize microservices traffic management, ensuring efficient communication and security. Gloo Gateway can handle external traffic seamlessly while leveraging a versatile hybrid gateway and service mesh solution for effective north-south and east-west traffic management.

6. About This Guide

This detailed buyer's guide includes research into Gloo Gateway, Apigee, Kong Gateway, Amazon API Gateway, and Tyk Gateway. The guide was created by a team at Solo.io who reviewed publicly available documentation, vendor information, blogs, videos, and technology details. This comparative analysis dives into the strengths and weaknesses of each platform using selected categories and features to offer valuable insights to potential buyers seeking a detailed understanding of the API gateway alternatives.

7. Legal Disclaimer

This Buyer's Guide ("Guide") has been created to provide enterprise buyers with valuable information when considering the purchase of products or services from various vendors.

Before utilizing the information contained within this Guide, it is important to read and understand the following disclaimer:

1. **General Information:** This Guide is intended for general informational purposes only. It does not constitute legal, financial, or professional advice. Users are advised to seek professional advice and conduct their own research before making any purchasing decisions.
2. **Accuracy of Information:** While we have made every effort to ensure the accuracy and reliability of the information provided in this Guide, we do not guarantee its completeness, accuracy, or timeliness. Information may become outdated or inaccurate over time, and providers' offerings may change without notice.
3. **Third-Party Providers:** This Guide may contain information about products or services offered by third-party providers. We do not endorse or recommend any specific provider, product, or service mentioned in this Guide. Users are encouraged to independently verify the information provided and make their own assessments regarding suitability and compatibility with their needs.
4. **Product and Service Comparisons:** The comparisons made in this Guide are based on the information available at the time of publication and may not reflect the current market conditions or the most recent updates from providers. Users are responsible for verifying the accuracy of any product or service comparisons and should consider their individual preferences and requirements.
5. **No Warranty or Guarantee:** We make no warranties or representations regarding the products or services mentioned in this Guide, including but not limited to their quality, performance, suitability, or fitness for a particular purpose. Users are advised to review the terms and conditions, warranties, and guarantees provided by the relevant providers before making any purchases.
6. **Legal Compliance:** It is the responsibility of users to ensure that any products or services they purchase comply with local, state, and federal laws and

regulations. This Guide does not constitute legal advice or a guarantee of legal compliance.

7. No Liability: We shall not be liable for any loss, damage, or inconvenience arising from the use of the information provided in this Guide. Users use this Guide at their own risk and agree to release us from any liability or claims.
8. Updates and Revisions: We may update or revise this Guide at any time without prior notice. Users should check for the most recent version of this Guide before relying on its information.

By accessing and using this Guide, you acknowledge that you have read and understood this disclaimer and agree to its terms and conditions. If you do not agree with any part of this disclaimer, please refrain from using this Guide.

This disclaimer is subject to change without notice. Please check for updates periodically.

8. Glossary

API (Application Programming Interface): A set of rules and protocols that allows different software applications to communicate with each other.

API Gateway: A server that acts as an API front-end, receiving API requests, enforcing throttling and security policies, passing requests to the back-end service, and then passing the response back to the requester.

Authentication: The process of verifying the identity of a user, system, or application to ensure that they have the necessary permissions to access an API.

Authorization: The process of granting or denying access to specific resources or actions based on the authenticated user's permissions.

CRD (Custom Resource Definition): In Kubernetes, an extension mechanism that allows users to define custom resources and controllers.

DevOps: A set of practices that combines software development (Dev) and IT operations (Ops) to improve collaboration and productivity by automating infrastructure, workflows, and continuously measuring application performance.

Endpoint: A specific URL or URI where an API can be accessed.

GraphQL: A query language for APIs and a runtime for executing those queries with existing data. It allows clients to request only the data they need.

JWT (JSON Web Token): A compact, URL-safe means of representing claims to be transferred between two parties, commonly used for authentication and authorization.

Middleware: Software that acts as a bridge between different applications, allowing them to communicate or share data.

Microservices: An architectural style that structures an application as a collection of small, independent services that communicate through APIs.

Observability: The ability to measure the internal state of a system and infer its behavior based on external outputs.

OAuth (Open Authorization): An open standard for access delegation commonly used for enabling secure authorization to APIs.

Proxy: An intermediary server that forwards requests from clients to other servers, often used in API gateways for routing and load balancing.

RBAC (Role-Based Access Control): A security paradigm that restricts system access to authorized users based on their roles and permissions.

Rate Limiting: Restricting the number of API requests a user or client can make within a specified time period.

REST (Representational State Transfer): A set of architectural principles for designing networked applications, often used for building APIs.

Service Mesh: A dedicated infrastructure layer for handling service-to-service communication, providing features like load balancing, service discovery, and security.

Webhook: An HTTP callback that allows third-party applications to receive real-time information from another application when a specific event occurs.

WebSockets: A communication protocol that provides full-duplex communication channels over a single, long-lived connection.