5 KEY CAPABILITIES of API Gateways

An API gateway works by receiving API requests from a client, authorizing their use, and connecting them to the application services. The gateway enforces security policies to protect against threats, and efficiently routes traffic between API producers and consumers. For cloud-native applications, the API gateway also needs to function well with Kubneretes. Here are 5 key capabilities you should look for in an API gateway:

1. API Security

Access control authorizes permitted users and controls user access to functions, data, or operations via mechanisms like role-based access control (RBAC). This needs to scale fluidly to meet the needs of modern applications.

2. Rate-Limiting

Rate-limiting reduces backend API load, prevents abuse, and enables safe exposure to third-party consumers. When using containers, the limits need to adjust to follow the scale and location in the cloud.

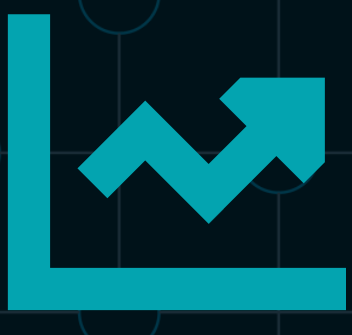
3. API Monitoring and Logging

Monitoring capabilities enable users to track requests, response times, SLAs, and unified logging for all APIs, including request IDs for end-to-end debugging. Cloud-native applications will need support for tools like Grafana and Prometheus.

4. API Transformation

Transforming request and response payloads aids the transition from traditional SOAP-based to modern REST-based architectures and speeds up time-to-market. Containerized applications also require requests and responses to traverse multiple clusters.

5. API Scalability



Scalability, high availability, load balancing, and shared state support must not compromise performance. Integration with Kubernetes is critical for cloud-native applications, especially when traversing multiple cloud-native clusters

Find out how Gloo Gateway can work for you:

https://www.solo.io/products/gloo-gateway/

solo.io