6 Essential Capabilities of a Modern API Gateway

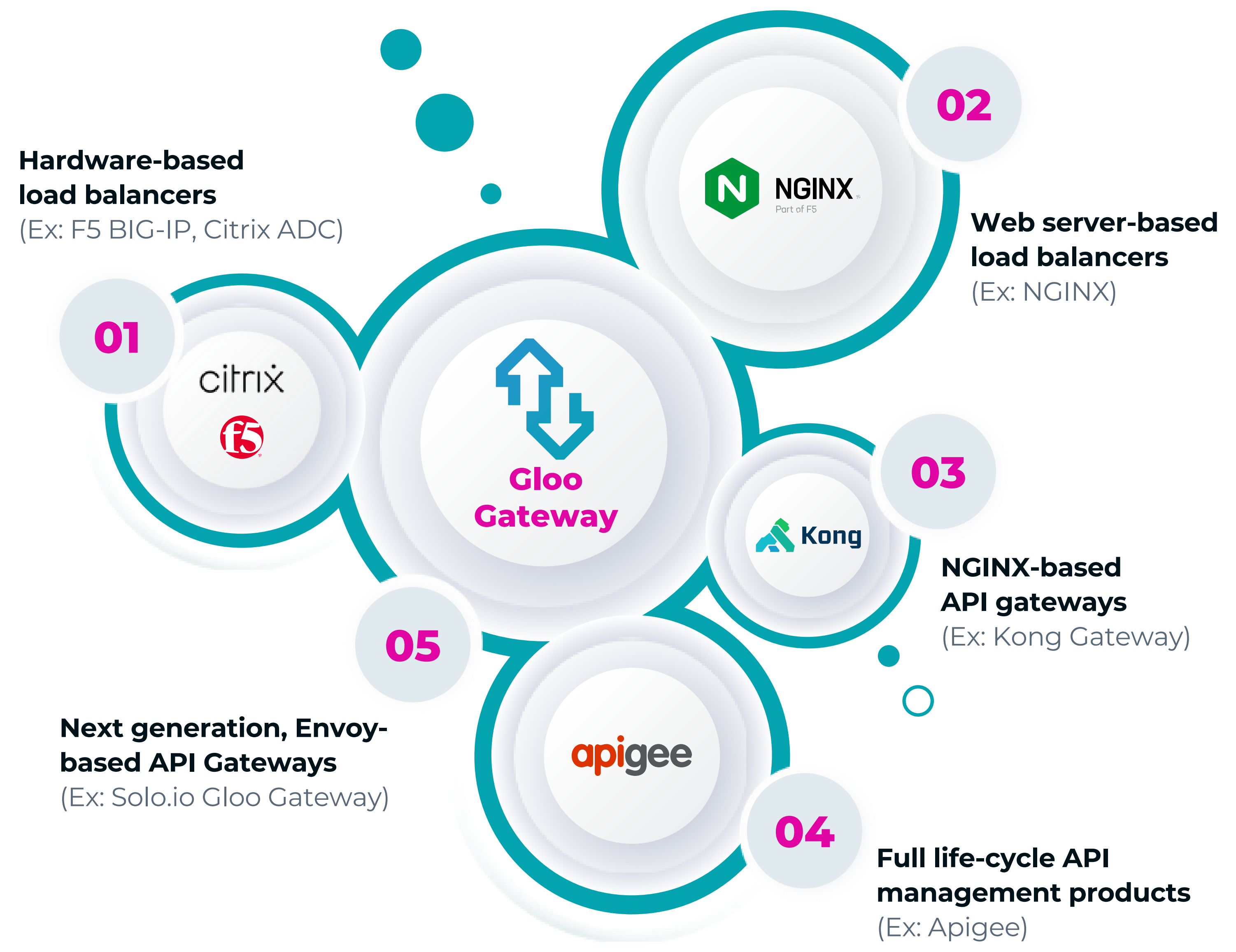
API gateways have been critical components of applications for years, but legacy solutions are showing their age.

As organizations increase their reliance on microservices and container-based architectures, they're finding that legacy API gateways fall short in regard to scalability, traffic management, security, and observability.

So, what are the defining features of a modern API gateway built for the rigors of today's microservices application architectures?

- Supports today's architectures, future-proofed for innovation
- Built on Envoy Proxy
- Extensible across operating environments, customizable with any language
- Capable of zero-trust security and advanced threat prevention
- Built-in Internet scalability and high-availability, with lower resource use
- Integrate seamlessly into DevOps/GitOps workflows

Let's compare today's most prevalent API gateway solutions to see how they stack up against these key criteria.



01 **Hardware-based load balancers**
(Ex: F5 BIG-IP, Citrix ADC)

02 **Web server-based load balancers**
(Ex: NGINX)

03 **NGINX-based API gateways**
(Ex: Kong Gateway)

04 **Full life-cycle API management products**
(Ex: Apigee)

05 **Next generation, Envoy-based API Gateways**
(Ex: Solo.io Gloo Gateway)

		Cloud-native, Envoy-based API Gateways	Full Lifecycle API Management Products	NGINX-based API Gateways	Web Server-based Load Balancers	Hardware-based Load Balancers
	Representative Products	Gloo Gateway	apigee	Kong	NGINX	f5
Modern	Built on Envoy Proxy and other current technologies	Envoy-based NOT Envoy-based	✕ NOT Envoy-based			It's just time to move from this rigid, legacy technology
Architecture	Built to support today's architectures	Supports RESTful APIs, gRPC, and GraphQL	Limited to no support for RESTful APIs, gRPC, and GraphQL			
Flexibility	Extensible across different architectures	Built in support for DLP and WAF. Extend capabilities in language-independent manner via WebAssembly	✕ Limited DLP and WAF. Can't extend capabilities in language-independent manner via WebAssembly			
Security	Capable of zero-trust security	Designd for zero-trust architectures	✕ Not designed for zero-trust	✕ Limited	✕ Not designed for zero-trust	
Scalability	Built-In Internet scalability and high-availability	Highly scalable	✕ Performance and latency issues due to the use of outdated technologies	✕ Does not scale ingress controller separately from the data plane leading to resource issues	Highly scalable	
Cloud-native Ops	DevOps and GitOps ready	Designed for DevOps and GitOps. Configured via CRDs	✕ Not designed for DevOps/GitOps. Requires a persistent data store for fully functional product		✕ Not designed for DevOps/ GitOps	

Not all API gateways are created equal.

Next-generation API gateways are purpose-built for highly dynamic, ephemeral environments such as Kubernetes and incorporate the design principles of declarative configuration, decentralized ownership, and self- service collaboration. In addition, next-gen gateways use declarative CRDs, enabling seamless integration into GitOps workflow.

Ready to learn more about Solo Gloo Gateway and why your next API Gateway needs to be Envoy-based?

solo.io

Find more information and further analysis on the different API gateways at www.solo.io.