

API Gateway vs Service Mesh

Similarities, differences, recommendations and user cases

Vijay Shankar Jha

Similarities

API Gateway and Service Mesh both can handle:

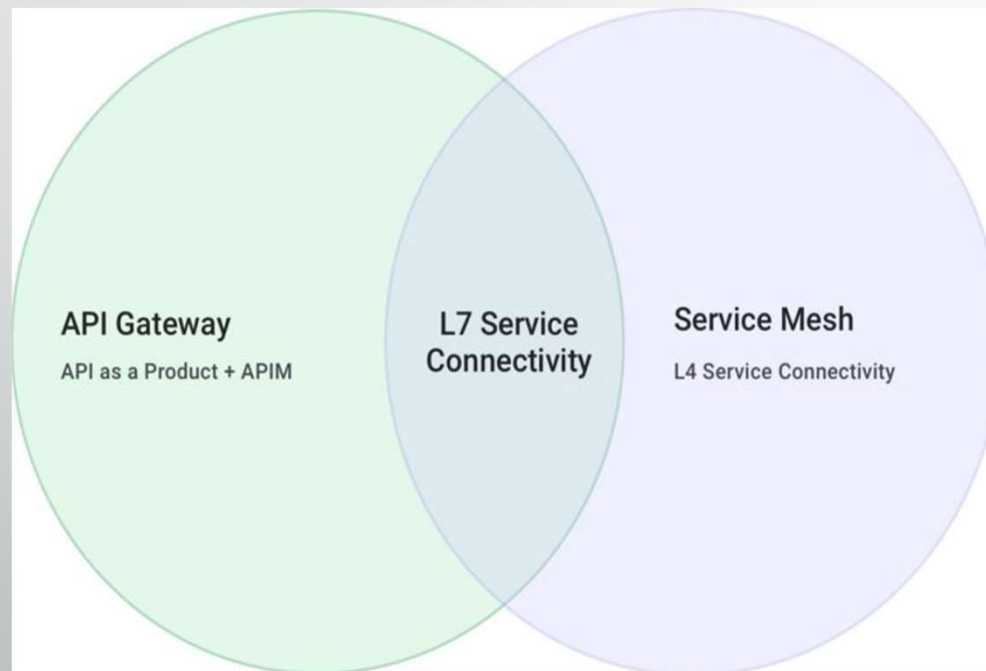
1. Service discovery
2. Request routing
3. Authentication
4. Rate limiting and
5. Monitoring

Differences

API Gateway	Service Mesh
Primarily meant for external client-to-service communication	Primary purpose is to manage internal service-to-service communication
Handles north-south traffic	Handles east-west traffic
Exposes external service to easily consumable	Manage and controls service inside own network
Maps external service to internal resources	Focuses on brokering internal resources
Exposes API to serve a specific business function	Sits between the network and the application, no real notion of business of solution

"east-west" traffic refers to traffic within a data center -- i.e. server to server traffic. "North-south" traffic is client to server traffic, between the data center and the rest of the network

API Gateway VS Service Mesh



1. The service connectivity capabilities that service mesh provides are conflicting with the API connectivity features that an API gateway provides
2. Service mesh are more inclusive (L4 + L7, all TCP traffic, not just HTTP and not just limited to APIs but to every service), they are in a way more complete

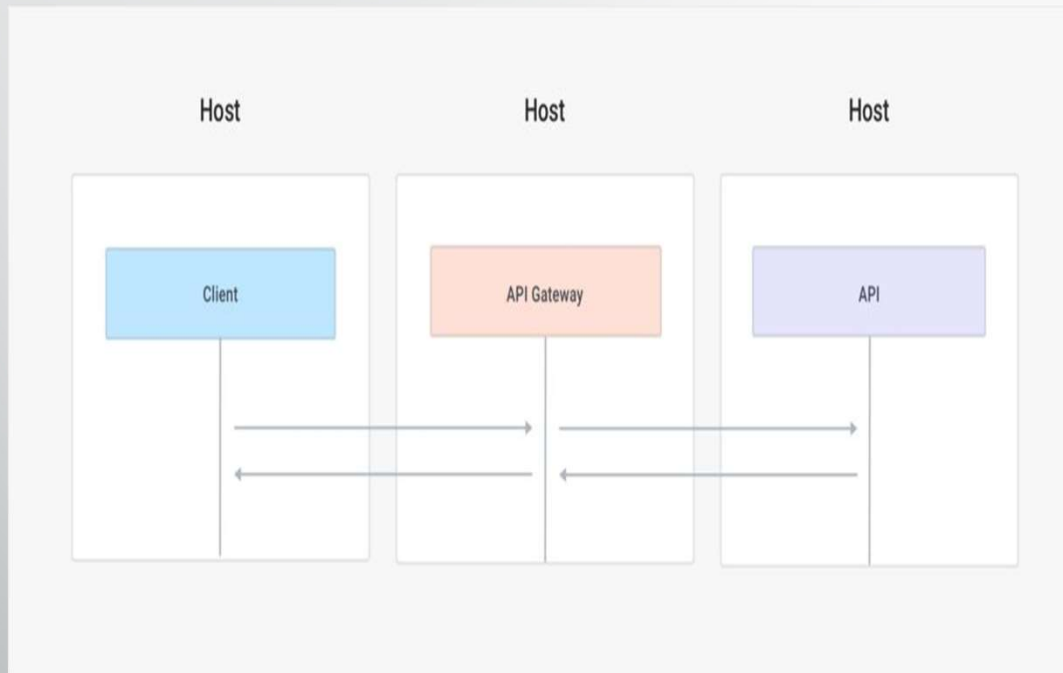
Recommendations

Use cases that service mesh does not provide, and that is the “API as a product” use case as well as the full API management lifecycle, which still belong to the API gateway pattern.

If an organization is not into API business, then Service Mesh is more suited.

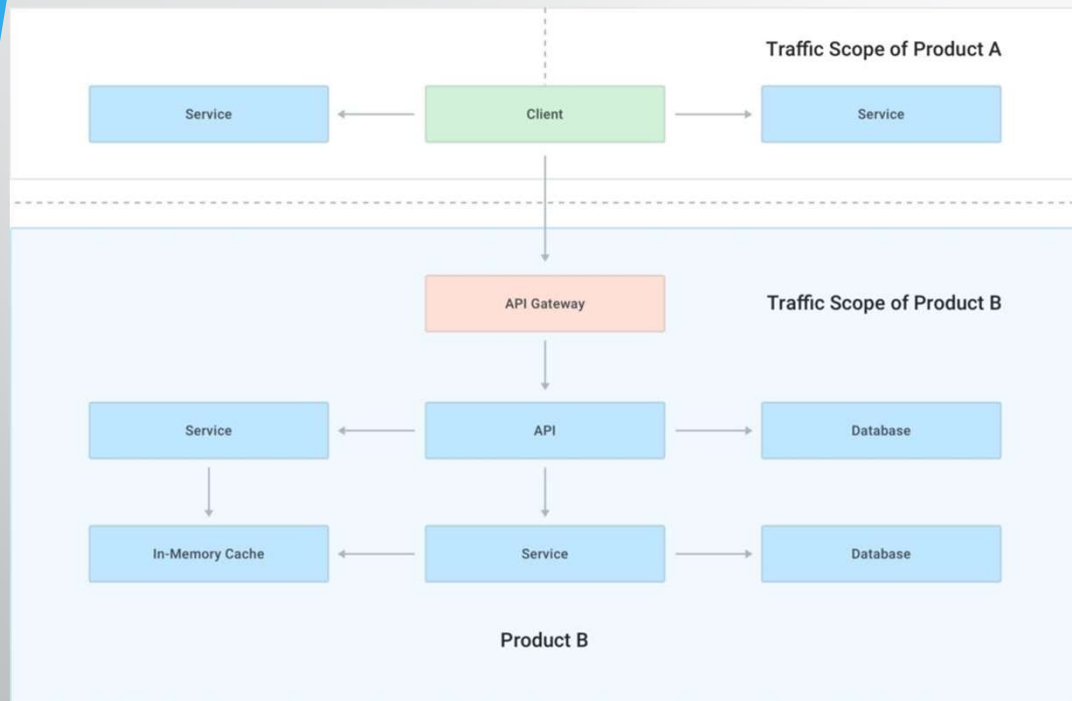
Later on if they require, they can use both API gateway and Service Mesh which complements each other or may be service mesh may come with API management in future

API Gateway Deployment



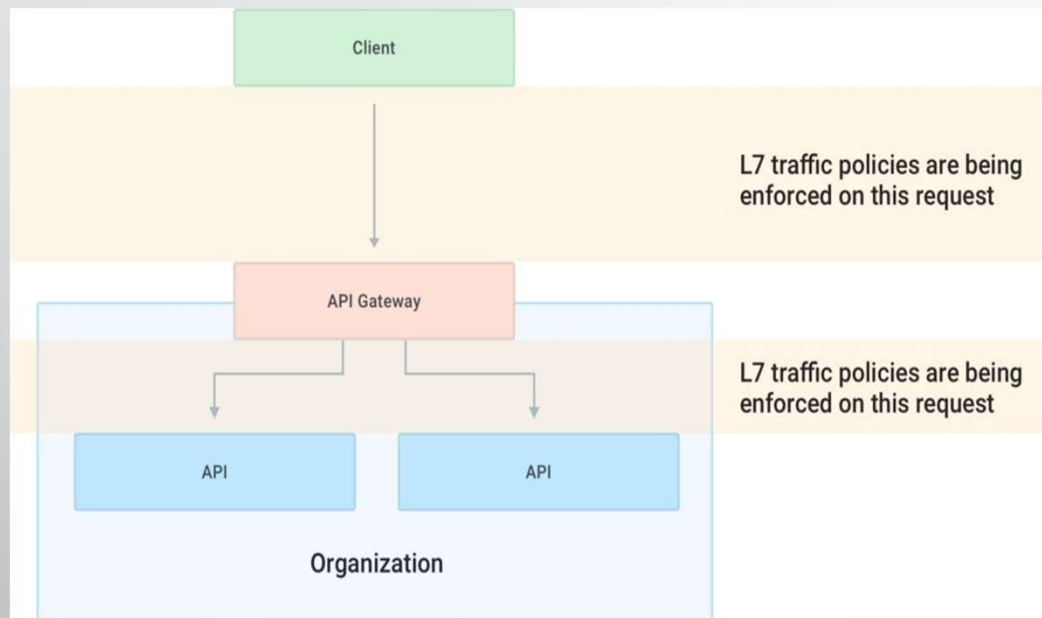
1. Deployed in its own instance (its own VM, host or pod)
2. Fully separated from the rest of our system
3. Fully lives in its own architectural layer
4. Usually cover three primary API use cases for both internal and external service connectivity
5. Covers both north-south (outside the datacenter) and east-west (inside the datacenter) traffic.

API Gateway Use Cases -- API as a Product



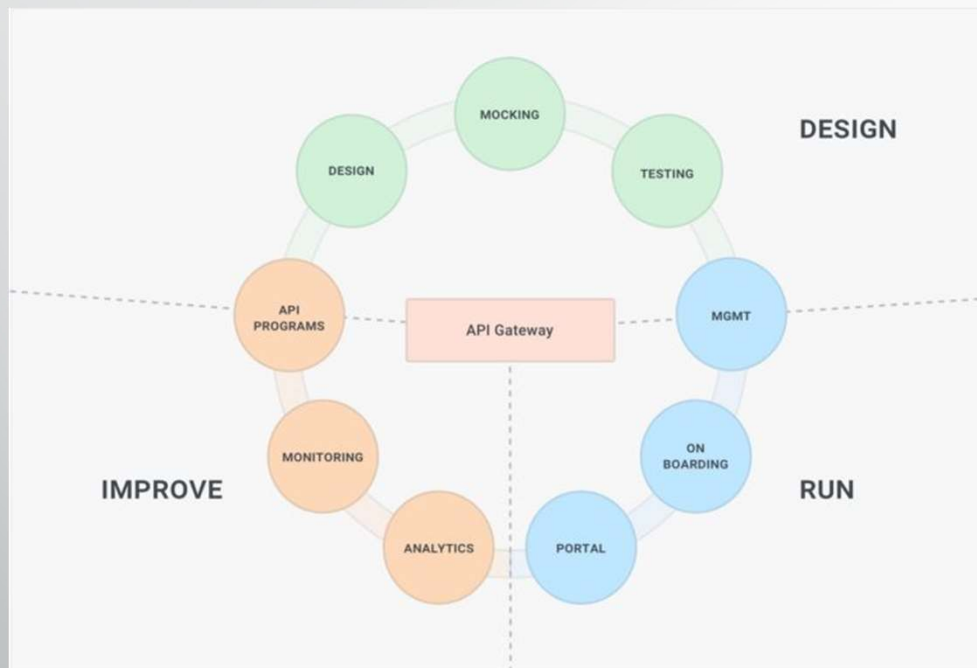
1. Packaging the API as a product that other developers, partners or teams will consume.
2. Will encapsulate services like AuthN/AuthZ use cases, rate-limiting, developer on-boarding, monetization or client application governance.
3. These are higher level use cases implemented by **L7 user policies**
4. Can be used as an abstraction layer to avoid code update for the clients consuming them.

API Gateway Use Cases -- Service Connectivity



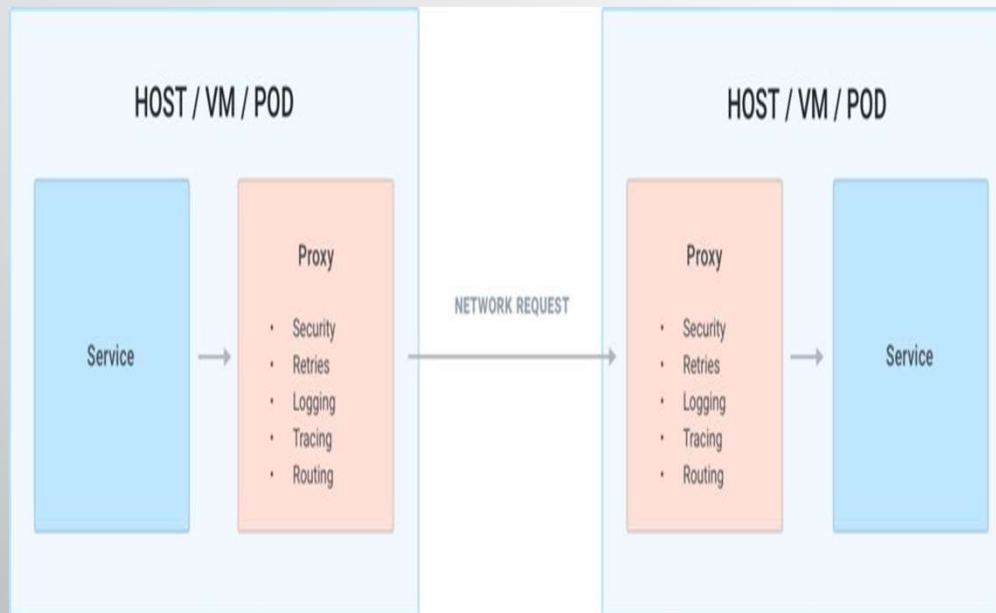
1. Can be used to enforce networking policies to connect, secure, encrypt, protect and observe the network traffic between the client and the API gateway, as well as between the API gateway and the APIs.
2. Can be called L7 traffic policies because it operates on the underlying network traffic as opposed to governing the user experience.

API Gateway Use Cases - API Lifecycle Management



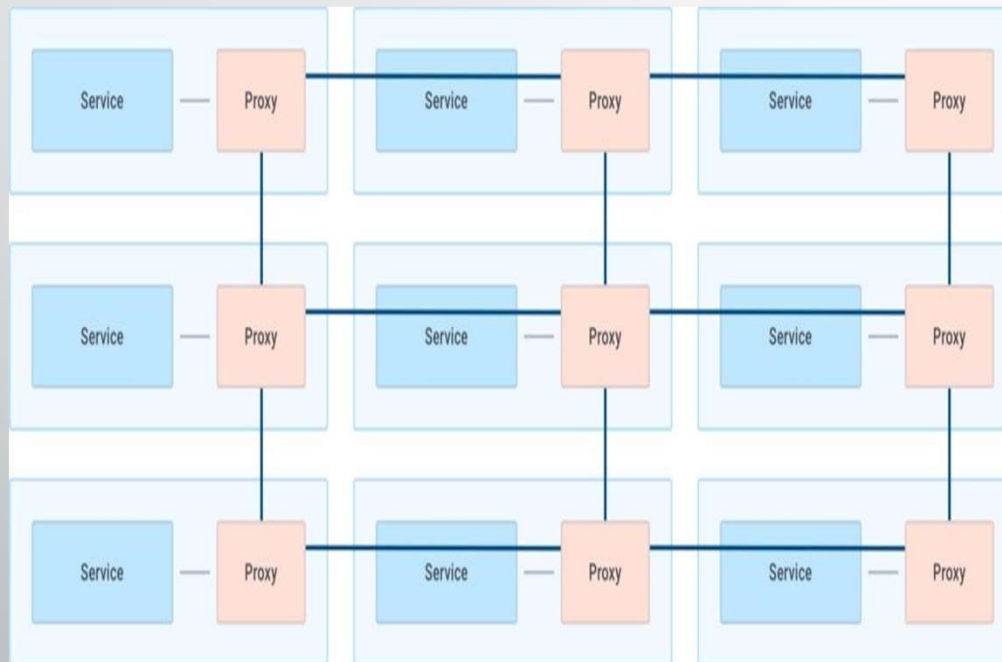
1. When offering APIs as a product, the APIs will have to provide a portal for end users to register their applications, retrieve the credentials and start consuming the APIs.
2. End-to-end and touches various points of the API lifecycle (and most likely different personas will be responsible for different parts of the lifecycle)

Service Mesh Deployment



1. Run one instance of the proxy (in purple) alongside every instance of our services
2. One proxy per service replica is considered to be the industry standard since it guarantees the best, highest availability and allows us to assign a unique identity (via a mTLS certificate) to every replica of every service.
3. A decentralized deployment (as opposed to the API gateway pattern, which is a centralized deployment)

Service Mesh Use Case -- Service Connectivity



1. Teams can avoid implementing network management in their own services
2. Proxy can then implement features like mutual TLS encryption, identity, routing, logging, tracing, load-balancing and so on for every service
3. a complete service mesh implementation will ideally support not just HTTP but also any other TCP traffic
4. Unlike API gateway it can support L4 and L7 both policy

A decorative graphic in the bottom-left corner consisting of several parallel lines. One prominent line is bright blue, and others are dark grey or black. The lines are arranged in a way that suggests a corner or a stylized 'L' shape.

Thank you