

Service Mesh vs API Gateway



Kasun Indrasiri

Follow

3 min read · Oct 9, 2017



1.7K



11



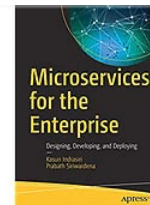
In one of my previous articles on service mesh, there were a couple of questions related to the relationship between **Service Mesh** and API Gateway. So, in this post, I'm planning to discuss the usage of Service Mesh and API Gateway.

Q 1

Microservices for the Enterprise: Designing, Developing, and Deploying

Microservices for the Enterprise: Designing, Developing, and Deploying [Kasun Indrasiri, Prabath Siriwardena] on...

www.amazon.com



In order to differentiate API Gateways and service mesh, let's have a closer look at the key characteristics API Gateways and Service Mesh.

API Gateway: Exposes your services as managed APIs

The key objective of using API Gateway is to expose your (micro) services as managed APIs. So, the API or Edge services that we develop at the API Gateway layer serves a specific business functionality.

- API/Edge services call the downstream (composite and atomic) microservices and contain the business logic that creates compositions/mashups of multiple downstream services.
- API/Edge services also need to call the downstream services on the resilient manner and apply various stability patterns such as Circuit Breakers, Timeouts, Load Balancing/Failover. Therefore most of the API Gateway solutions out there have these features built in.
- API Gateways also come inbuilt support for service discovery, analytics(observability: Metrics, monitoring, distributed logging, distributed tracing.) and security.
- API Gateways closely work with several other components of the API Management ecosystem, such as API marketplace/store, API publishing portal.

Service Mesh

Now let's look at how we can differentiate Service Mesh.

Q 1 · Top highlight

- Service Mesh is a network communication infrastructure which allows you to decouple and offload most of the application network functions from your service code.
- Hence when you do service-to-service communication, you don't need to implement resilient communication patterns such as Circuit breakers, timeouts in your service's code. Similarly, service mesh provides other functionalities such as service discovery, observability etc.

Coexistence of API Gateway and Service Mesh

The key differentiators between API Gateways and service mesh is that API Gateways is a key part of exposing API/Edge services where service mesh is merely an inter-service communication infrastructure which doesn't have any business notion of your solution.

Figure 1 illustrates how API Gateway and service mesh can coexist. As we discussed above, there are also some overlapping features (such as circuit breakers etc.) but it's important to understand these two concepts are serving fundamentally different requirements.

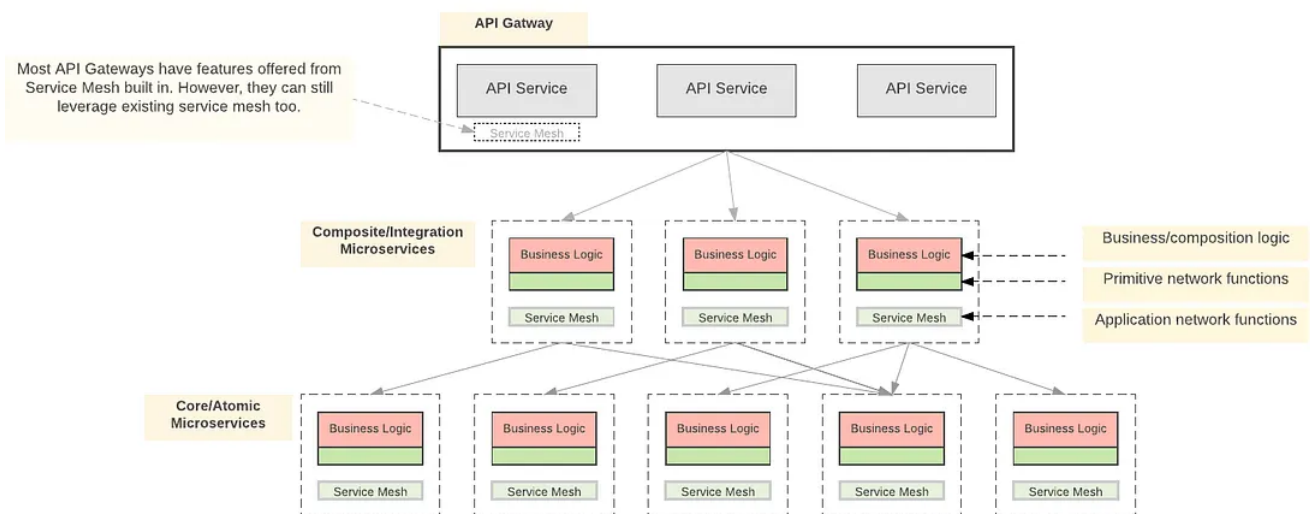


Figure 1: API Gateways and service mesh in action

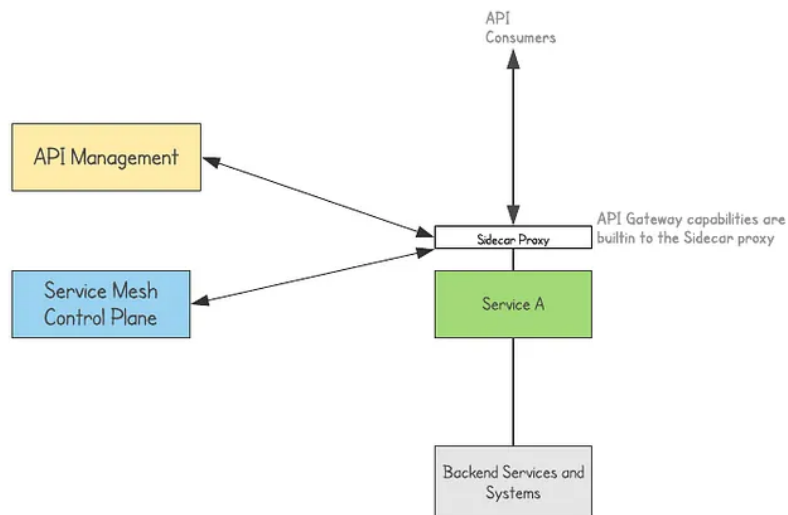
As shown in figure 1, service mesh is used alongside most of the service implementations as a sidecar and it's independent of the business functionality of the services.

On the other hand, API Gateway hosts all the API services (which has a clearly defined business functionality) and it's a part of the business functionality of your solution. API Gateway may have in-built inter-service communication capabilities but that doesn't prevent API Gateway using service mesh to call downstream services(API Gateway->service mesh->microservices).

At API Management level, you can either use in-built inter-service communication capabilities of API Gateway or API Gateway can call downstream services via service mesh by offloading application network functions to service mesh.

Replacing API Gateway with Service Mesh sidecar proxy

In certain use cases, we can eliminate API Gateway altogether and offload the API Gateway capabilities to the Sidecar proxy that is colocated with the service that you want to expose as an API. The API management plane/control plane needs to communicate with the sidecar proxy to apply the API Management capabilities.



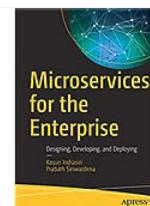
However, this model force us to expose services through the sidecar proxy layer which is colocated with the microservice.

References

Microservices for the Enterprise: Designing, Developing, and Deploying

Microservices for the Enterprise: Designing, Developing, and Deploying [Kasun Indrasiri, Prabath Siriwardena] on...

www.amazon.com





Published in Microservices in Practice

1.7K followers · Last published Oct 9, 2017

Microservices in Practice

Follow



Written by Kasun Indrasiri

1.7K followers · 82 following

Sr. Product Manager at Confluent, Ex-Product Manager of Azure Event Hubs,
O'Reilly Author of gRPC & Cloud Native Patterns, Speaker @KubeCon

Follow

Responses (11)



Ishu

What are your thoughts?



Huabing Zhao

Apr 12, 2018



It's an inspiring post, so I translated this article into Chinese and published it at my blog with the original link.
Please let me know if you're OK with that.

<https://zhaohuabing.com/2018/04/11/service-mesh-vs-api-gateway/>

Thanks,

Huabing



4



1 reply

[Reply](#)



Ilya

Jan 13, 2018 (edited)



Can you list some most popular implementations of API Gateway?



2



3 replies

[Reply](#)



Miraz Al-Mamun

Jun 17, 2019



Service Mesh is a network communication infrastructure which allows your to decouple and offload most of the application network functions from your service code

I guess there is a small typo:

Service Mesh is a network communication infrastructure which allows you to decouple and offload most of the application network functions from your service code

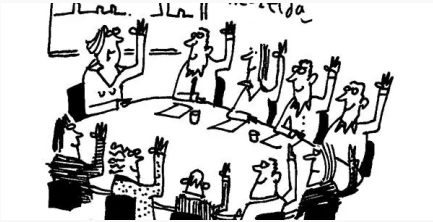



2

[Reply](#)

[See all responses](#)

More from Kasun Indrasiri and Microservices in Practice

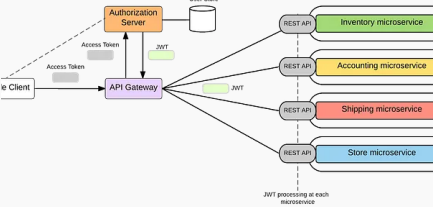



 Kasun Indrasiri

Understanding RAFT Distributed Consensus

Distributed Consensus is one of the hardest problems in the distributed computing...

★ Apr 5, 2020 🖱 421 💬 3 📌 + ...

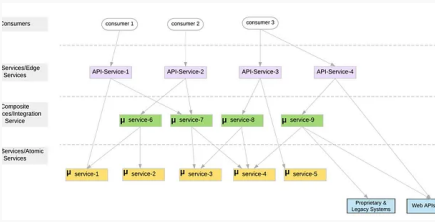


 In Microservices in Practice by Kasun Indrasiri

Pragmatic Microservices

Nowadays, Microservices is one of the most popular buzz-words in the field of software...

Aug 29, 2017 🖱 2.2K 💬 12 📌 + ...

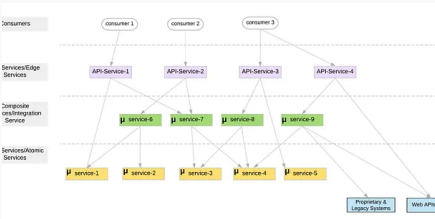



 In Microservices in Practice by Kasun Indrasiri

Microservices Layered Architecture

With Microservices architecture, a single software application/functionality is...

Sep 15, 2017 🖱 676 💬 6 📌 + ...



 Kasun Indrasiri

Microservices, APIs and Integration

I've been writing/speaking on the topics related to Microservices for a quite a...

Aug 22, 2017 🖱 266 💬 3 📌 + ...

See all from Kasun Indrasiri

See all from Microservices in Practice

Recommended from Medium



Rahul Kumar

Idempotency : RESTful API in microservices

Failures are the reality of any application. Single off failures are still bearable but the...

May 15 8



Ujjawal Rohra

What Senior Architects Really Know About the SAGA Pattern...

Not just theory—practical SAGA examples every Java developer should know.

6d ago 6



Eric Anicet

Spring Cloud Gateway OpenID Connect with Keycloak

In this story, we'll explore how to secure microservices architectures with Spring...

Jun 23 7



Matias Salinas

GCP Pub/Sub v/s Amazon SQS v/s Azure Service Bus

As cloud computing continues to gain popularity, more and more businesses are...

May 1 38



In Another Integration Blog by Rajeshkumar (RKD)

GenAI for MuleSoft Architects: From API-Led to Agent-Led...

From API-Led to Agent-Led: A Practitioner's View

Aug 11 1



In Designed to Scale by Ryan Aminollahi

Why We Need Real AI Architects Now

The business case for strategic AI leadership, not just technical skills

May 2



See more recommendations