Service Mesh

Computación en la Nube

Oscar H. Mondragón

Contexto

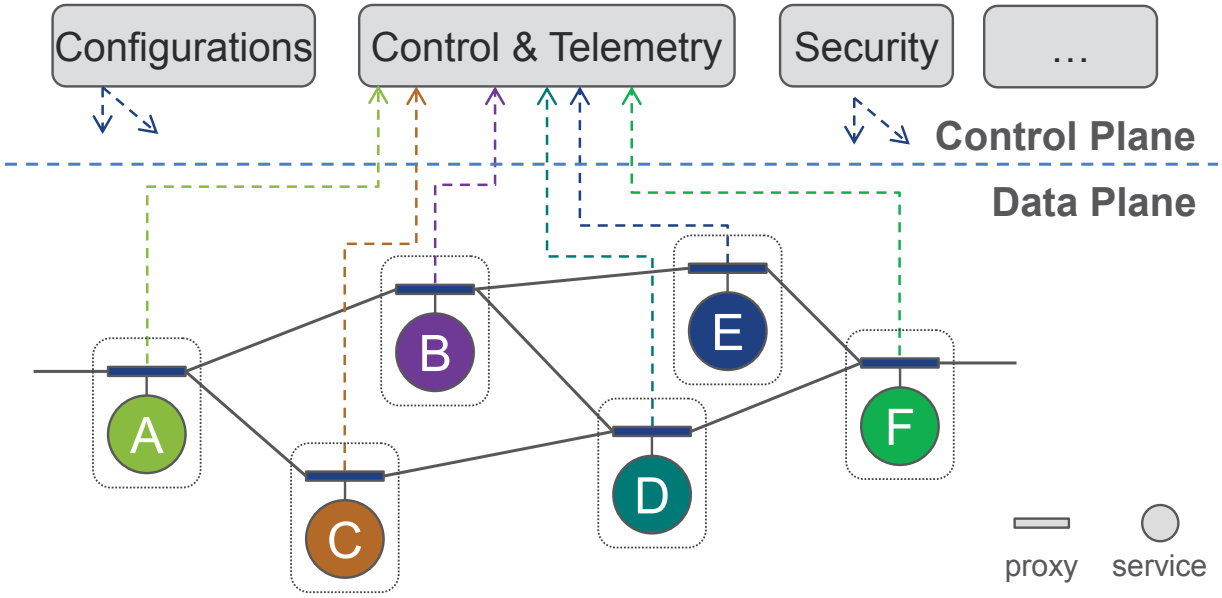
Category	The Past	The Present
Infrastructure & Execution Environment	Data Centers	Orchestrated Environments
	Bare Metal / Virtual Machines	Containers
Network Concerns	IP Address, DNS	Service Discovery
	TCP/IP	gRPC, REST, ...
Software Design	Monolithic Applications	Microservices
	Hardware Redundancy	Design for Failure
	On Premises	Cloud Native
Development Methodology	Isolation between Development and Operations	DevOps

Fuente: Li, W., Lemieux, Y., Gao, J., Zhao, Z., & Han, Y. (2019, April). Service mesh: Challenges, state of the art, and future research opportunities. In *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)* (pp. 122-1225). IEEE.

Que es un Service Mesh?

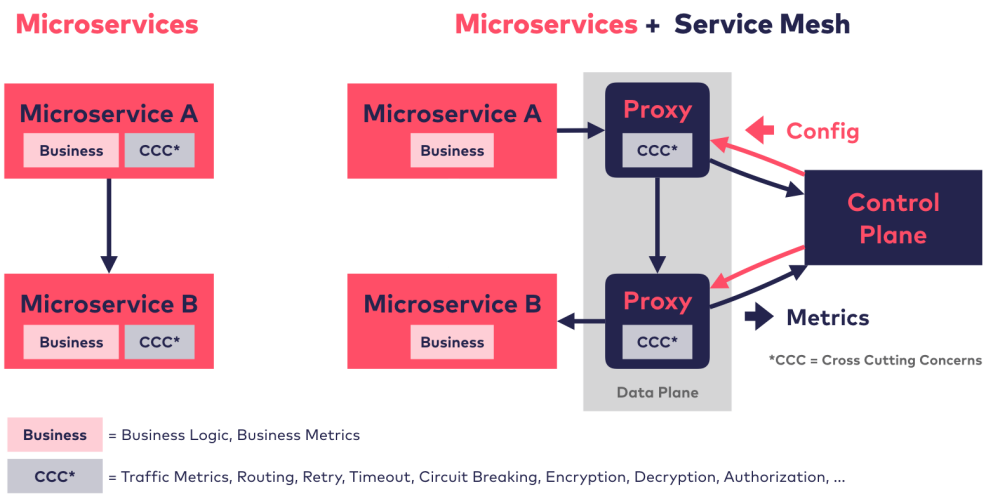
- Capa de infraestructura dedicada para manejar la comunicación de servicio-a-servicio.
- Responsable de la entrega confiable de solicitudes a través de la compleja topología de servicios que comprende una aplicación nativa en la nube.
- Se suele implementar como una matriz de proxies de red ligeros que se implementan junto con el código de la aplicación, sin que la aplicación tenga que darse cuenta.

Ejemplo de Arquitectura Service Mesh



El plano de datos se compone de un conjunto de proxies inteligentes, que normalmente se implementan como sidecars. Estos proxies median y controlan todas las comunicaciones de red entre microservicios.

Características



- Service discovery
- Health checking
- Routing
- Load balancing
- Authentication/authorization
- Observability
- Telemetry
- Fault tolerance
- Traffic Monitoring
- Circuit breaking

Implementaciones de Service Mesh



Istio

If you have heard about service mesh, you have probably heard about Istio too. Istio is by far the most popular service mesh because of its rich feature set and Google's and IBM's support.

Linkerd

Linkerd was the first service mesh. The modern 2.x versions are committed to simplicity, performance, and building on top of Kubernetes as the underlying platform.

Consul

HashiCorp's Consul has been well known as a service discovery solution for a long time. Now that it has adopted the Envoy proxy and Sidecar pattern, Consul can serve as a service mesh for a variety of platforms like Kubernetes and VMs.

AWS App Mesh

Not long after the service mesh hype, AWS added its own service mesh for applications on AWS.

Fuente: https://servicemesh.es/

Implementaciones de Service Mesh

Traefik Mesh

As the name already reveals, Traefik Mesh (formerly Maesh) is the service mesh based the cloud-native API gateway Traefik.

Kuma

Kuma is a service mesh using Envoy and the sidecar pattern made by developers of an API gateway - Kong. It focuses on multi-cloud and can run non Kubernetes workloads.

Open Service Mesh (OSM)

A new implementation by Microsoft, following common service mesh design principles like adopting envoy and implementing SMI spec.

Cilium

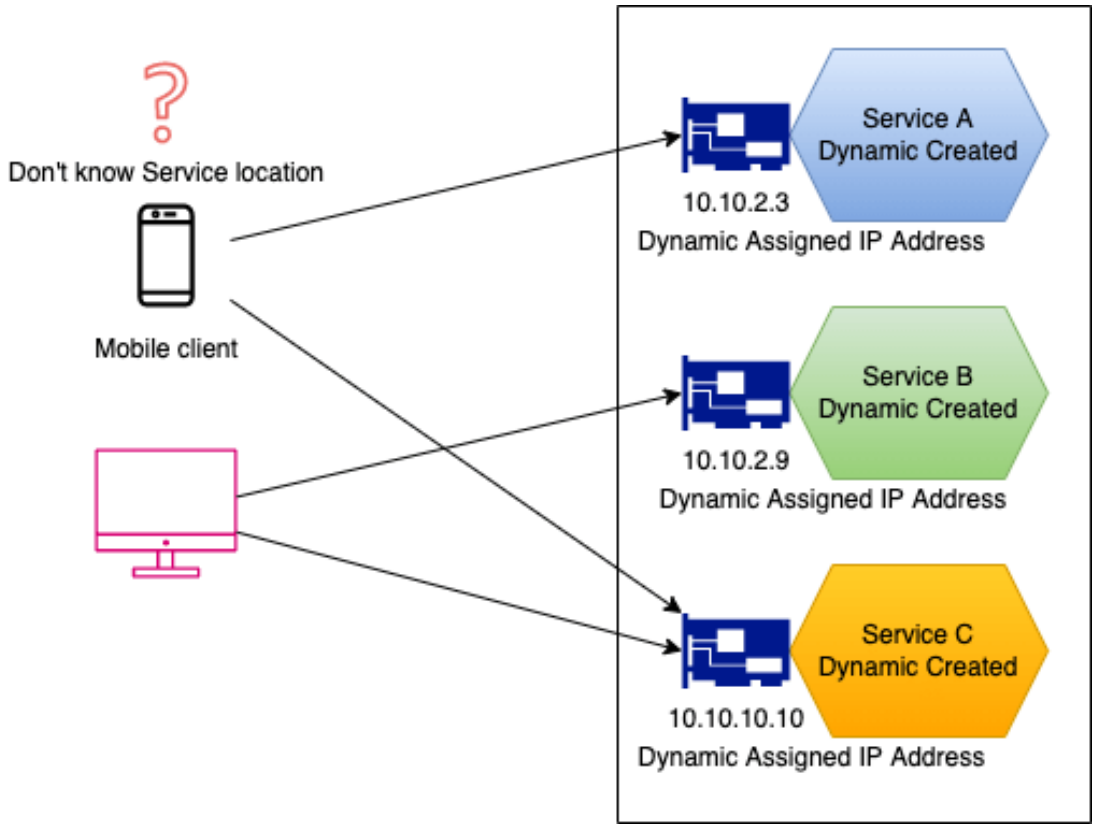
Cilium existed long before the term Service Mesh even came up, providing security, networking and oservability features efficiently with eBPF instead of sidecars.

Fuente: https://servicemesh.es/

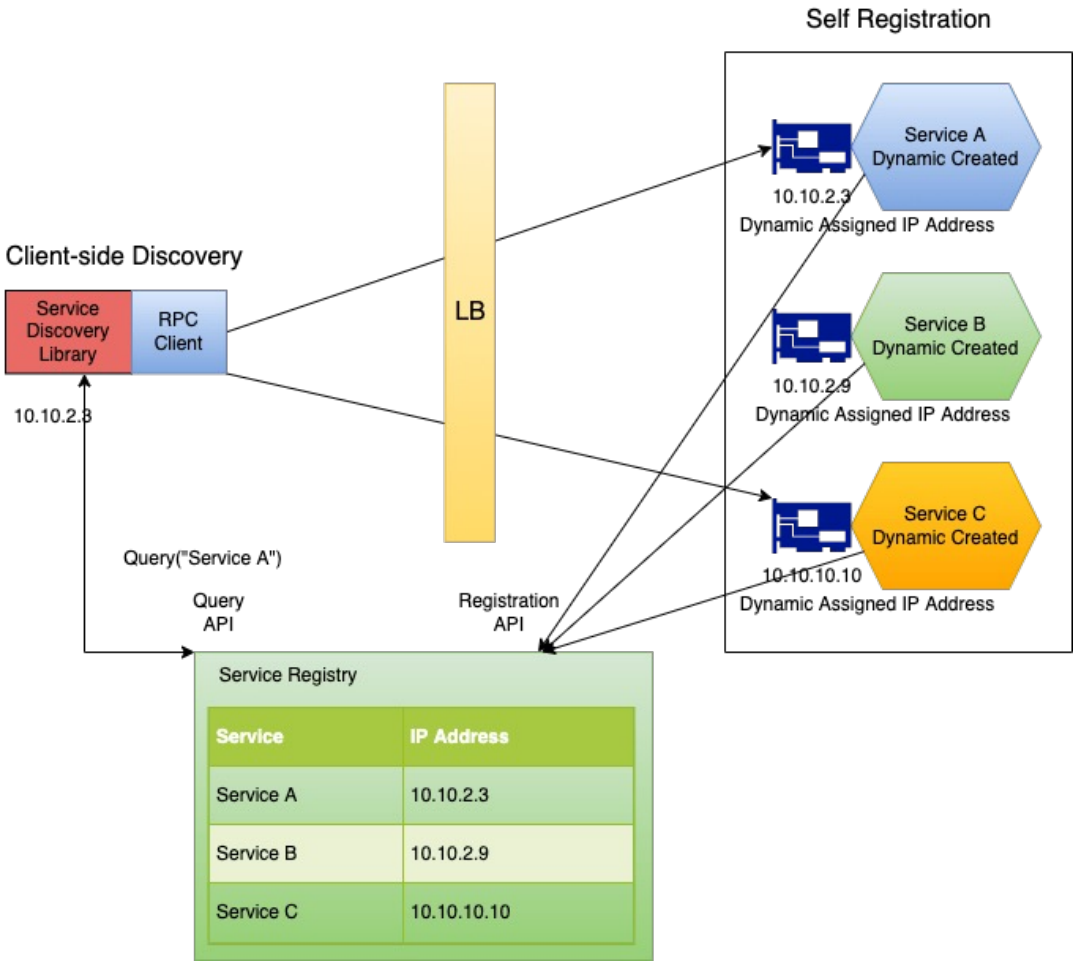
Service Discovery

- En las aplicaciones de microservicios, la cantidad de instancias de servicio, así como los estados y la ubicación de un servicio, cambian dinámicamente con el tiempo.
- Se requiere un mecanismo de descubrimiento de servicios para permitir que los consumidores de servicios descubran la ubicación y realicen solicitudes a un conjunto de instancias de servicios que cambian dinámicamente.
- Por lo general, las instancias de servicio se descubren buscando un registro que mantiene registros de nuevas instancias, así como también de instancias que se eliminan de la red.

Sin Service Discovery



Service Discovery a nivel de aplicación



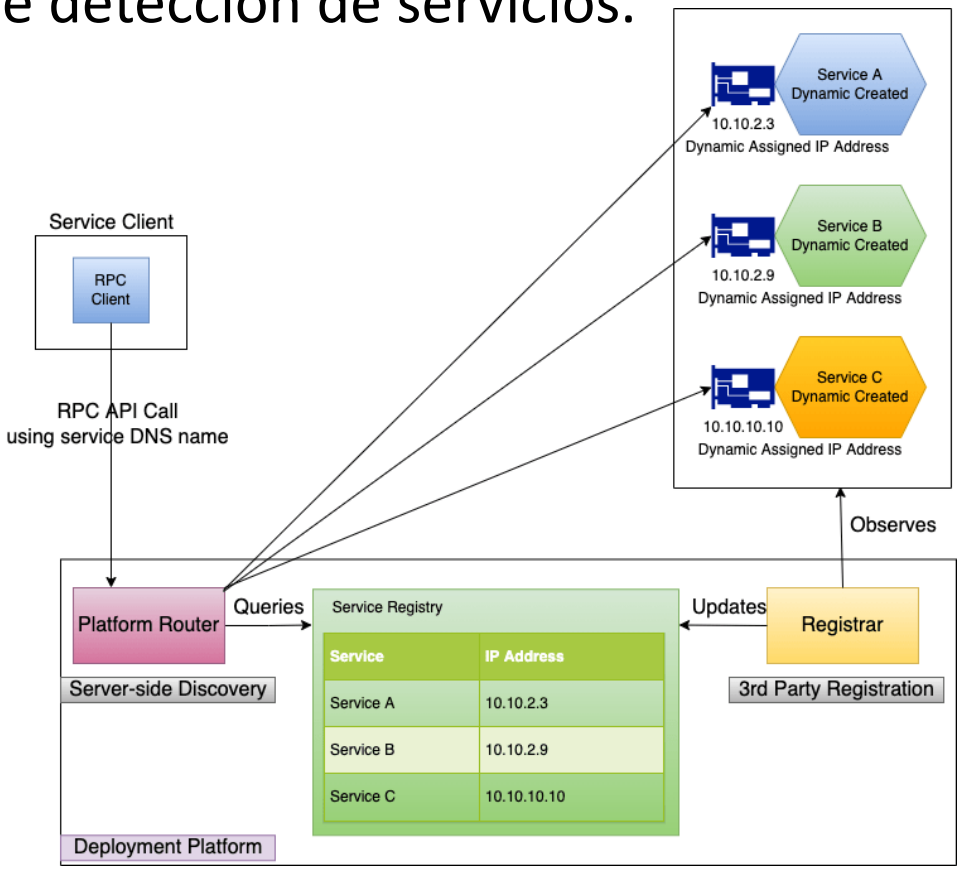
Combinación de dos patrones:

Self-registration pattern

Client-side discovery pattern

Service Discovery a nivel de plataforma

Muchas plataformas de implementación modernas, como Docker y Kubernetes, tienen registros de servicios integrados y mecanismos de detección de servicios.



Combinación de dos patrones:

3rd Party registration pattern

Server-side discovery pattern

Service Discovery en Consul

1. Descargar Consul desde https://www.consul.io/downloads e instalar
2. Iniciar el agente Consul

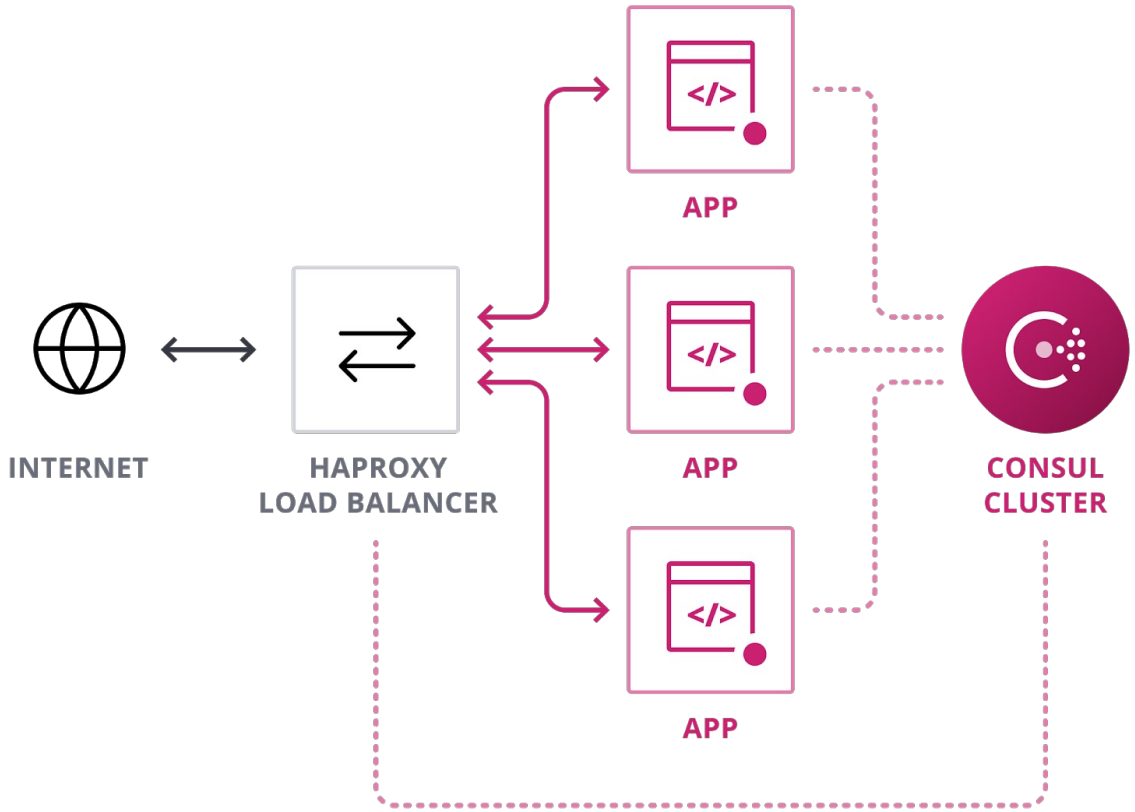
```
consul agent -ui -dev -bind=192.168.100.3 -client=0.0.0.0 -data-dir=.
```

3. Registrar el servicio

```
/* Registro del servicio */
var check = {
  id: SERVICE_ID,
  name: SERVICE_NAME,
  address: HOST,
  port: PORT,
  check: {
       http: SCHEME+'://'+HOST+':'+PORT+'/health',
       ttl: '5s',
       interval: '5s',
     timeout: '5s',
     deregistercriticalserviceafter: '1m'
       }
  };

consul.agent.service.register(check, function(err) {
    if (err) throw err;
    });
```

Proyecto



Referencias

- Li, W., Lemieux, Y., Gao, J., Zhao, Z., & Han, Y. (2019, April). Service mesh: Challenges, state of the art, and future research opportunities. In *2019 IEEE International Conference on Service-Oriented System Engineering* (pp. 122-1225). IEEE.
- Introduction to Consul. https://www.consul.io/docs/intro
- Service Mesh Comparison. https://servicemesh.es/
- Microservices Service Discovery Design Patterns. https://www.learncsdesign.com/microservices-service-discovery-design-patterns/
- Service Discovery, patron arquitectonico. https://www.oscarblancarteblog.com/2018/09/24/service-discovery-pattern-para-microservicios/