

# Service Mesh

apidays

Kong

Kuma





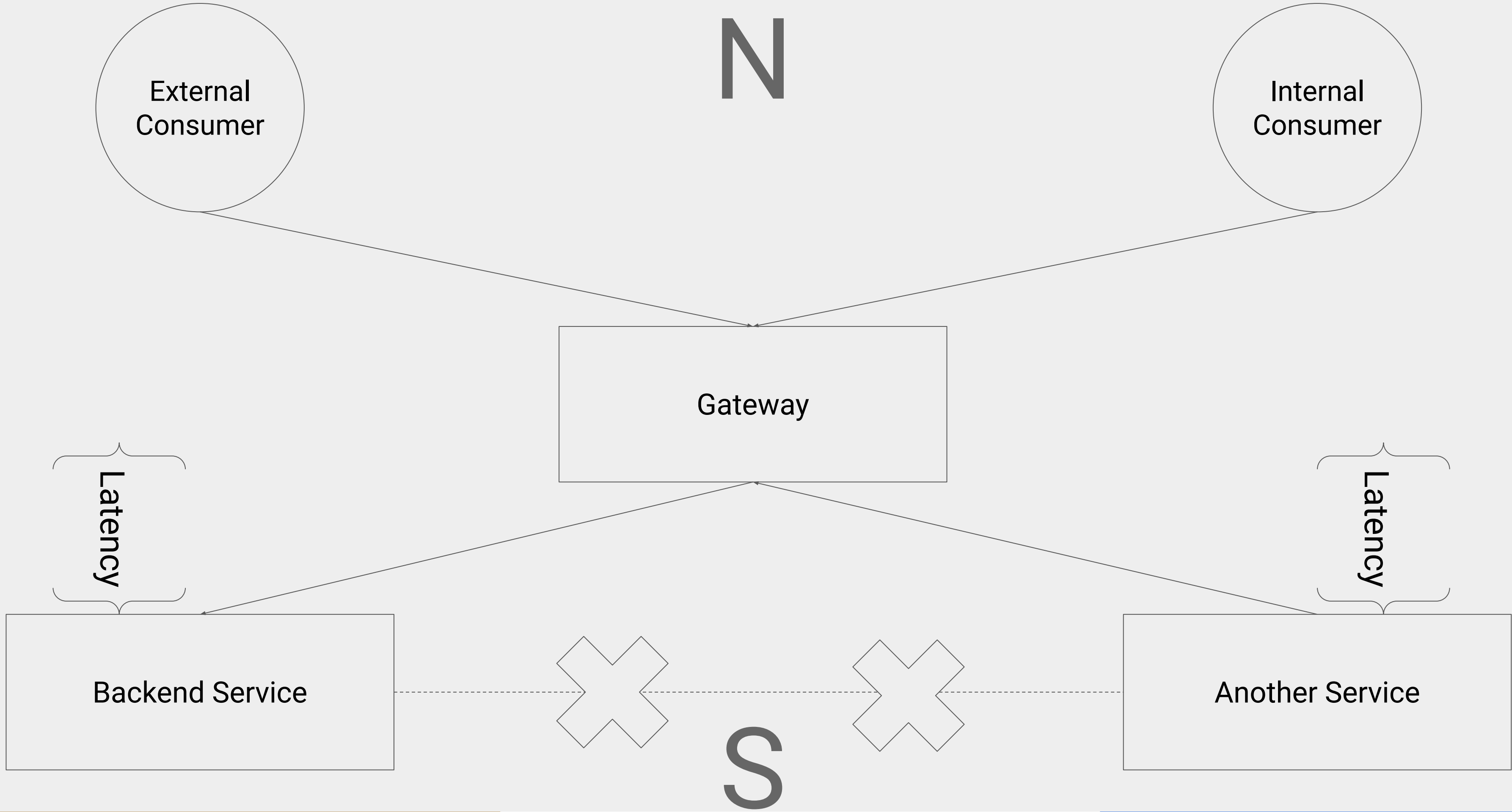
# Service Mesh

## Agenda

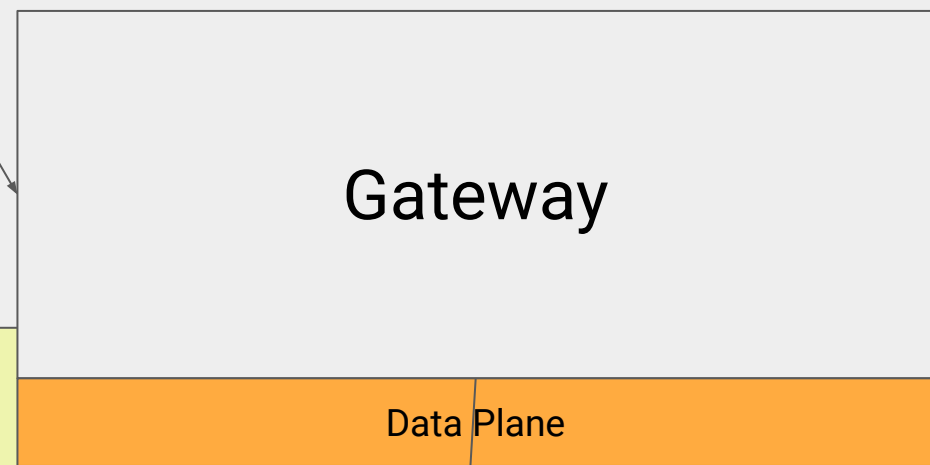
- Service Mesh
- Service Mesh
- Live demo
- Microservices
- Lab environment
- Q&A

why  
how  
see  
where  
try  
!?

# API Gateway - a little bit of history



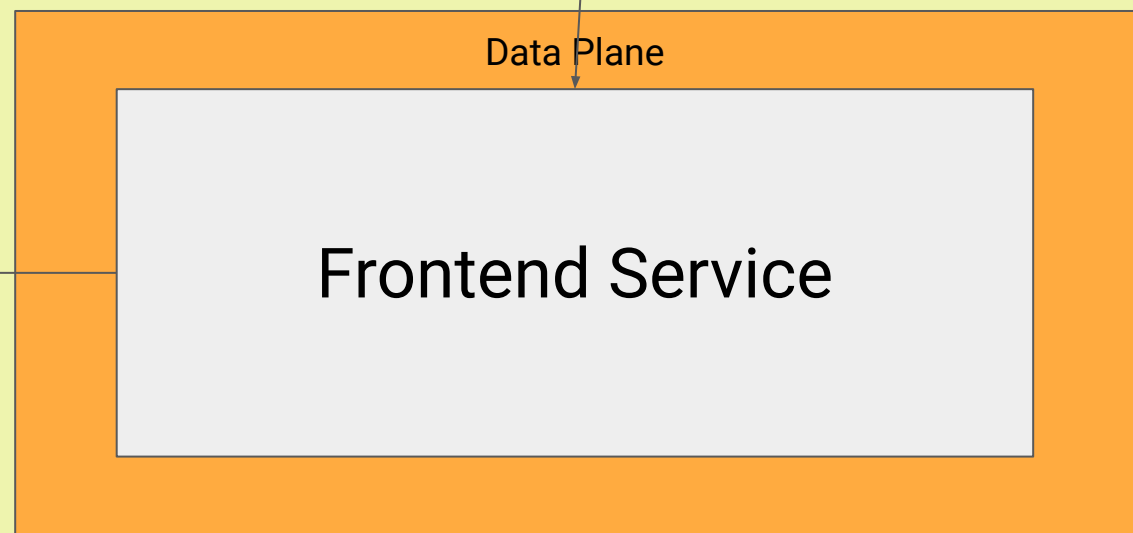
# Service Mesh - One Ring To Rule Them All?



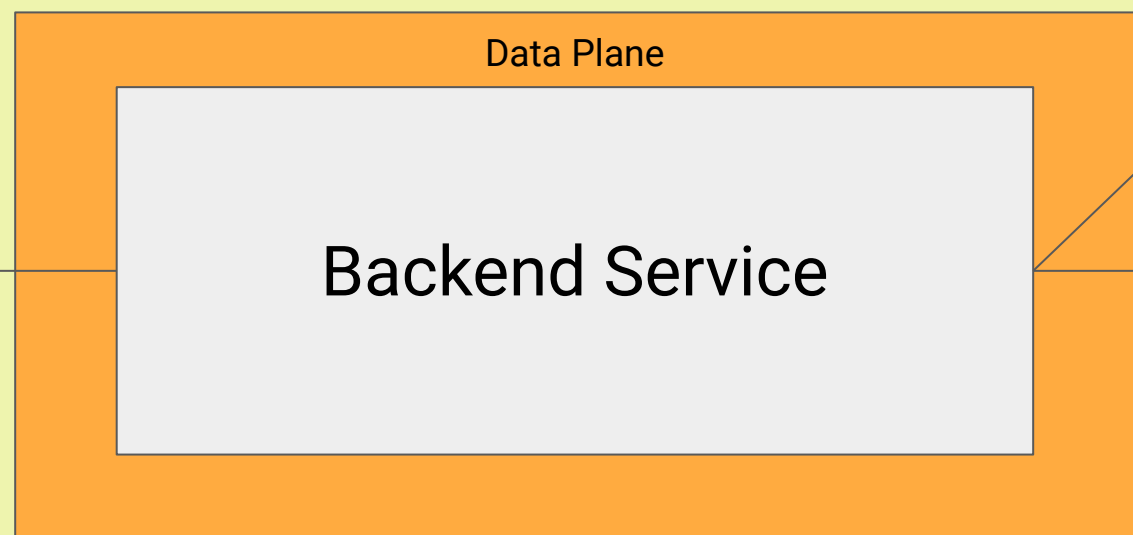
N

Mesh

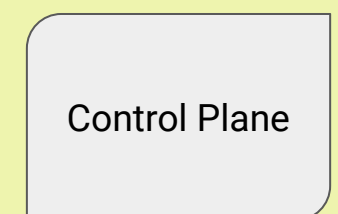
W



E



S



# What is a Service Mesh?

## Definition

- A dedicated infrastructure layer for making service-to-service communication.
- It relies on the separation of the control plane and data plane.
- Data plane is deployed in a sidecar pattern

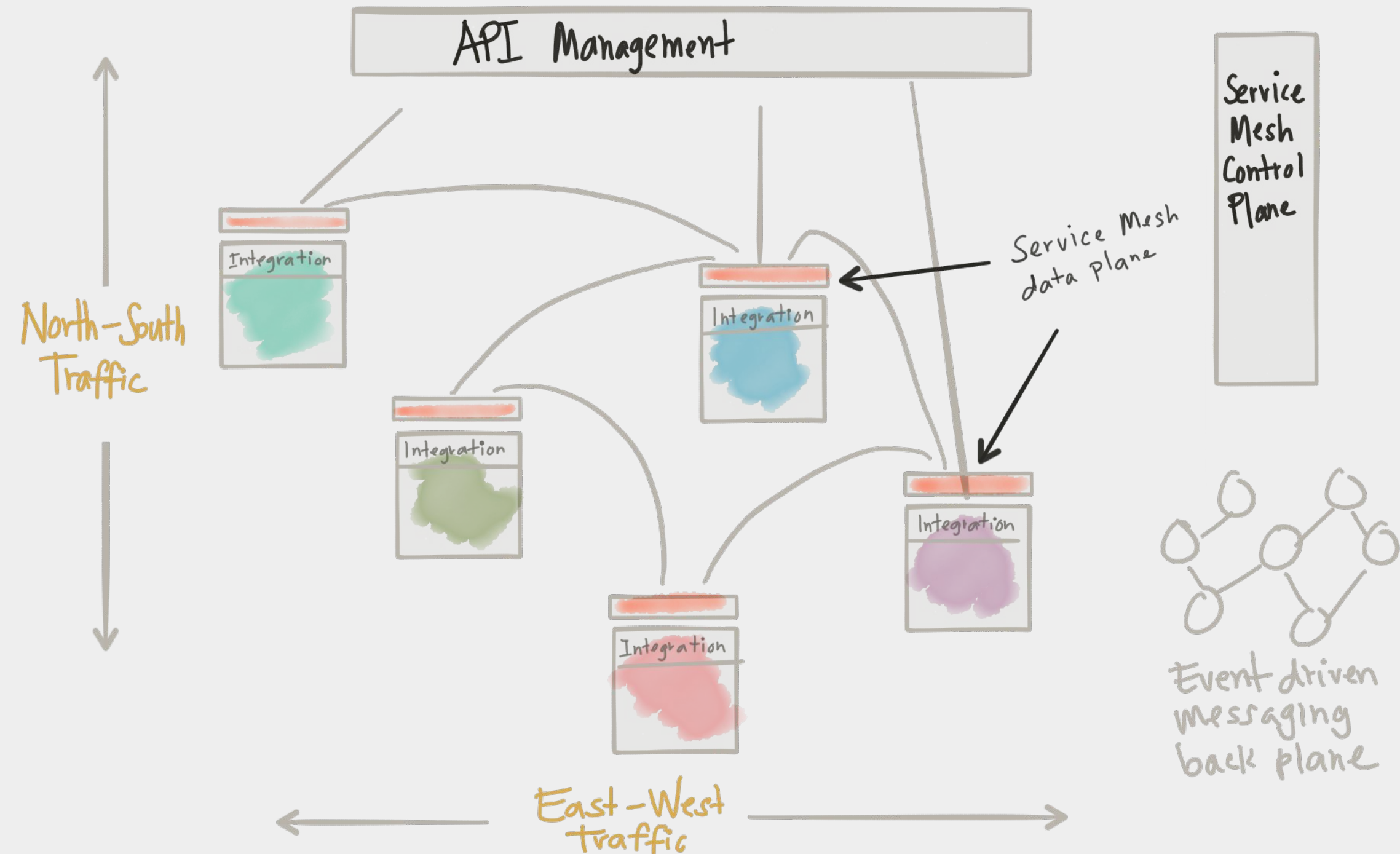


Image source: <http://blog.christianposta.com/microservices/application-network-functions-with-esbs-api-management-and-now-service-mesh/>

# What does a service mesh do?

1

Improve  
Security

2

Externalize  
Policy

3

Reduce  
Complexity

4

Improve  
Observability

Inside Kubernetes

Outside Kubernetes

# Improving Network Security

## Service Mesh

### Control access at the logical level

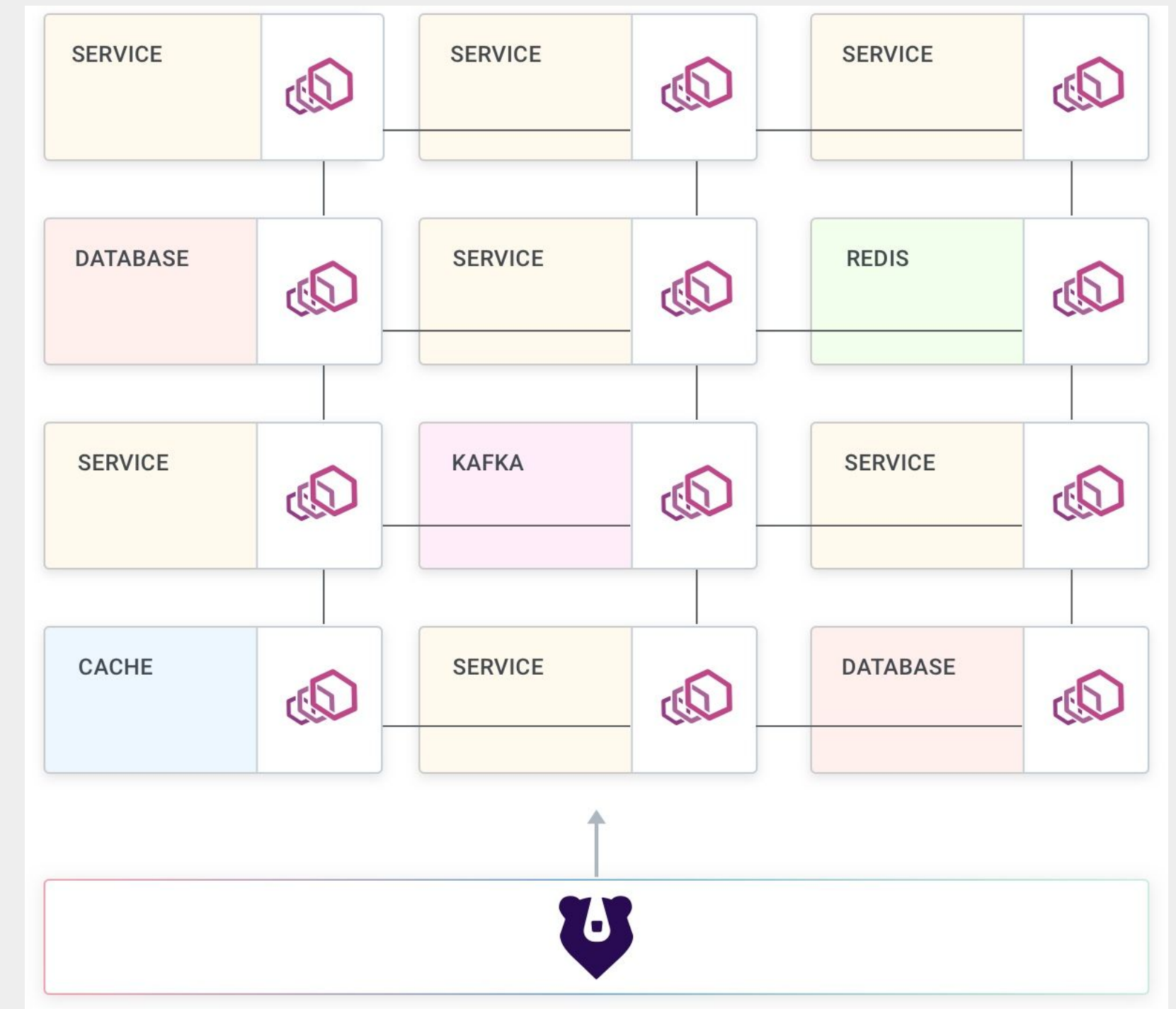
Services, not IP ranges or VPC names

### Principle of least privilege (zero trust)

Apply to network connectivity, not just AuthN/AuthZ

### First line of defense

No exploit without connectivity (e.g. Heartbleed)



# Externalize Network Policy

## Service Mesh

### Decentralize business logic, centralize policy

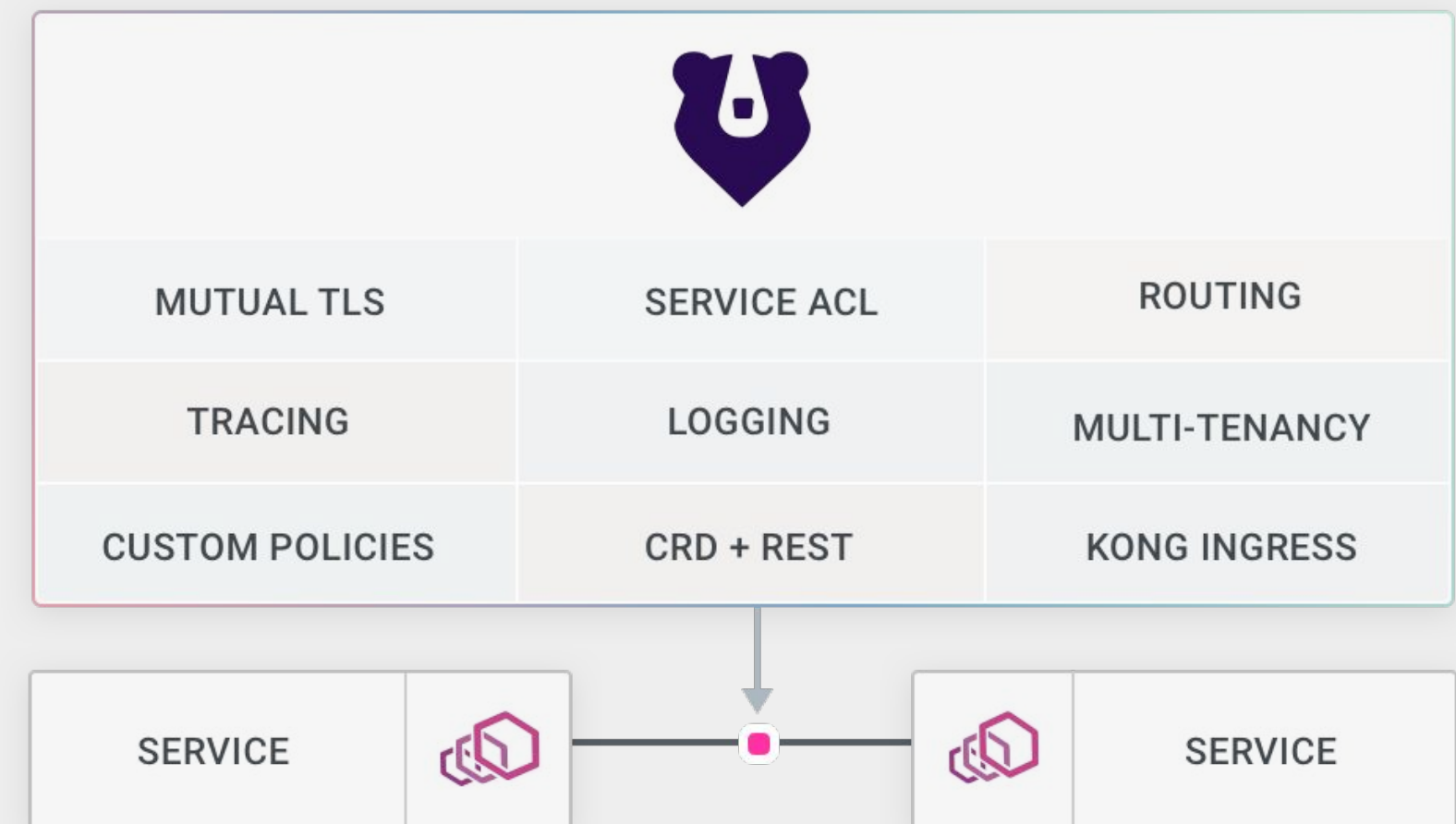
Apply policy out-of-the-box outside application code

### Security, Observability, Performance

Mutual TLS, Logs/Metrics/Traces, Health checks

### No additional latency or code

Out-of-process transparent proxy on same host (sidecar)





# Reduce Network Complexity

## Service Mesh

### Distributed network configuration

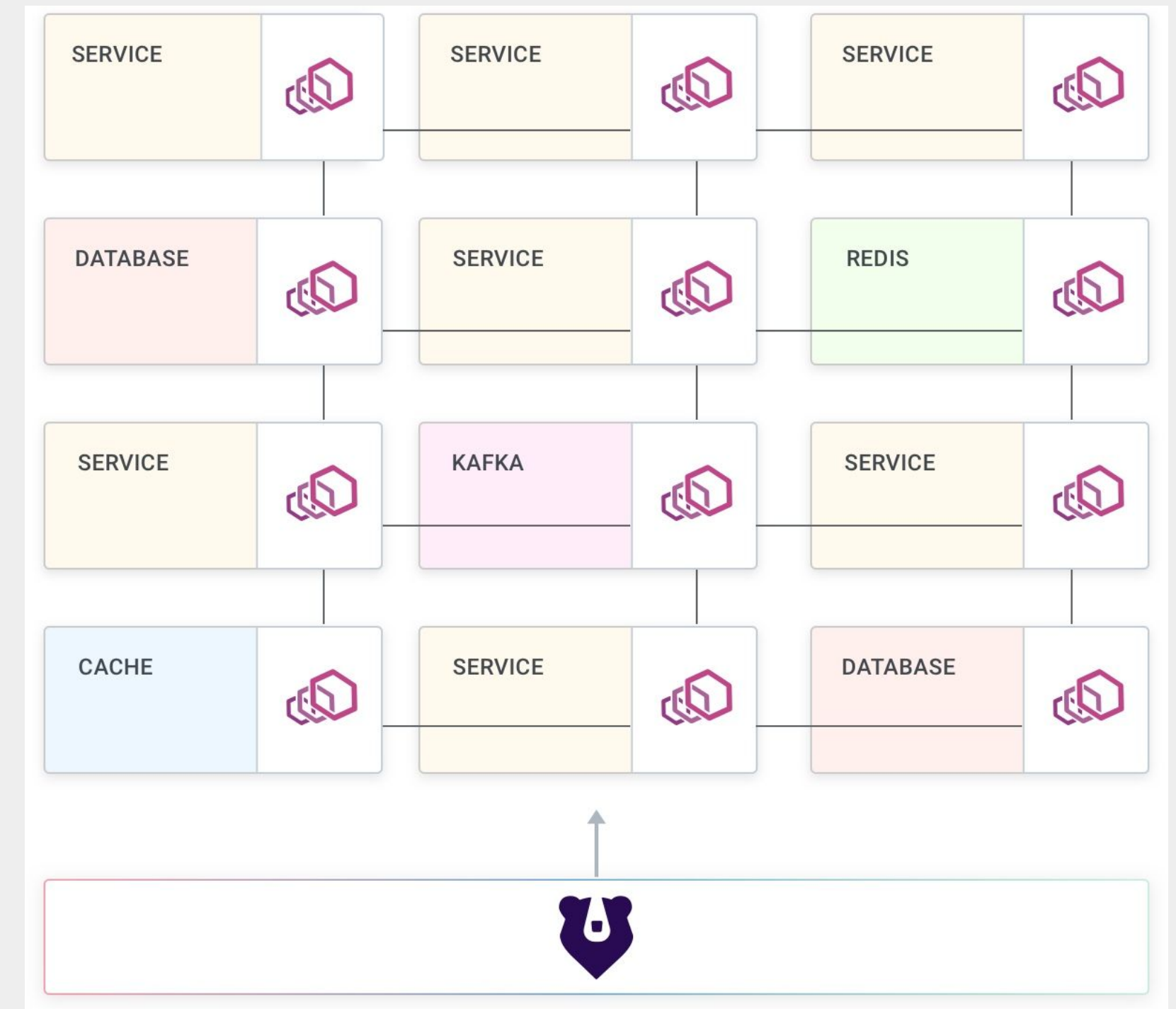
Firewalls, load balancers, VPCs, compute instances

### Hard to know complete configuration

Which IP can actually connect to which other IP

### Service mesh as the central configuration

(Other layers will still exist)



# Improve Network Observability

## Service Mesh

### Distributed network visibility

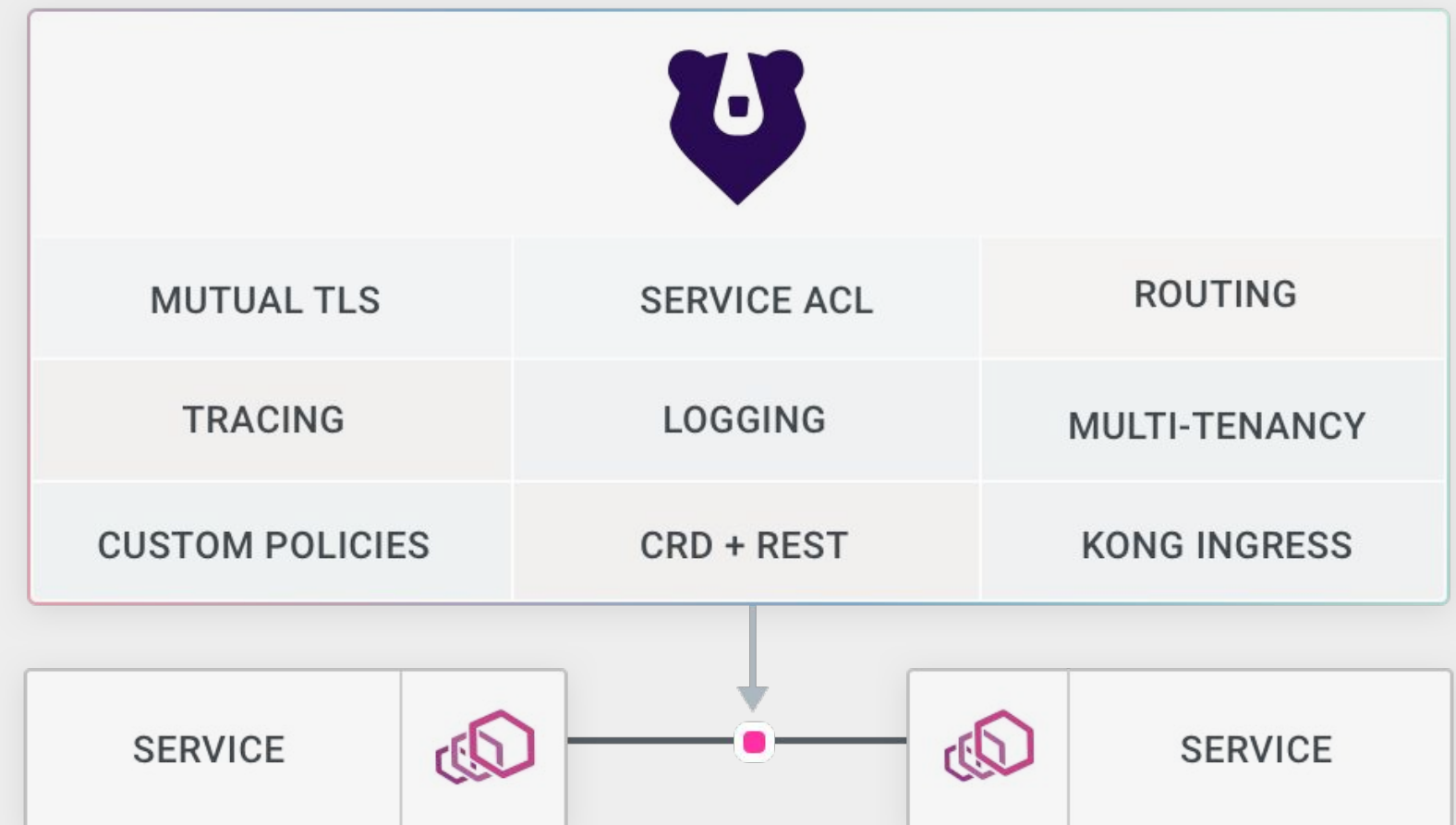
Firewall logs, ELB logs, VPC flow logs

### Logs, Metrics, Traces

Collect automatically and transparently for every service call

### Consistency

Single pane of glass





# API Gateway

When should we use one?

## APIs as a product

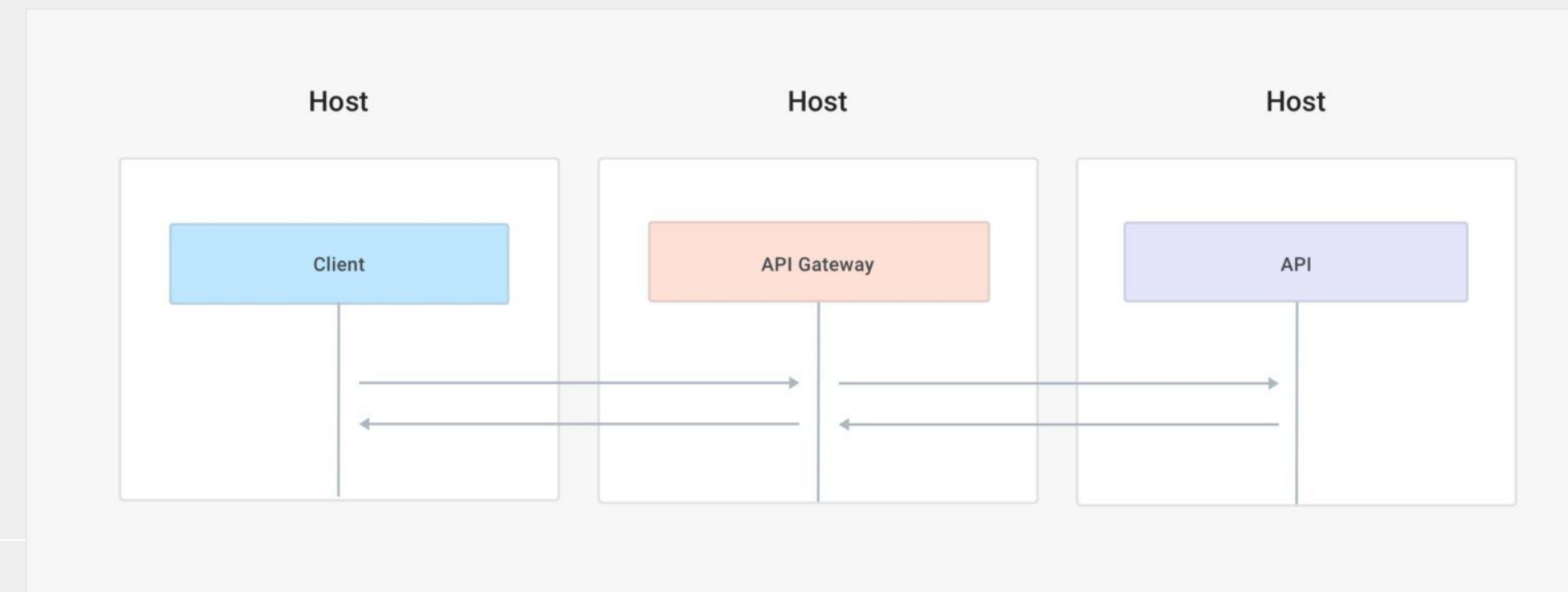
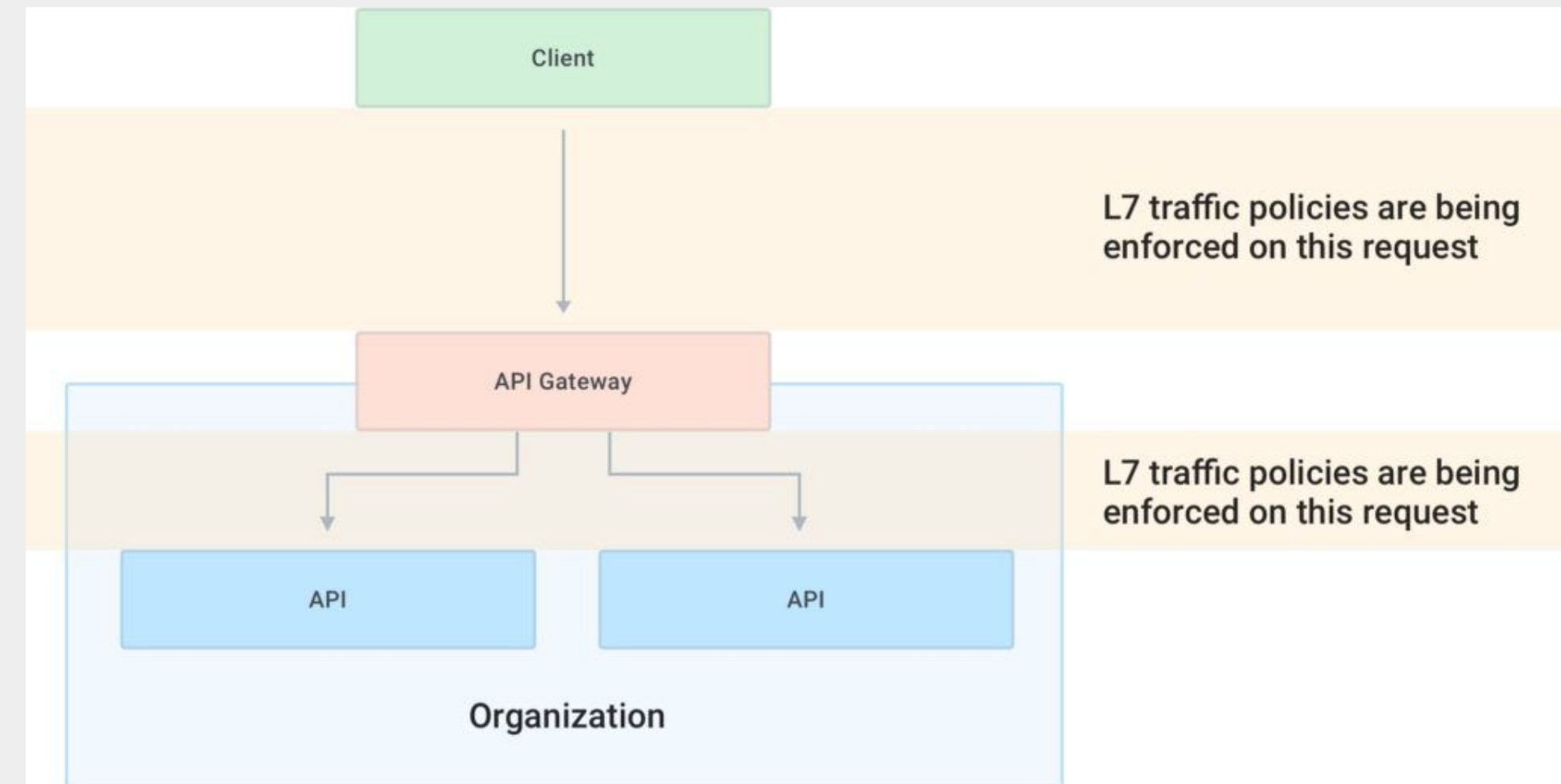
Packaging the API as a product that other developers, partners or teams will consume.

## Service Connectivity (at Layer 7)

Enforcing 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.

## Full Lifecycle API Management

An API gateway is being one piece of a larger puzzle in the broader context of API management.



# Service Mesh

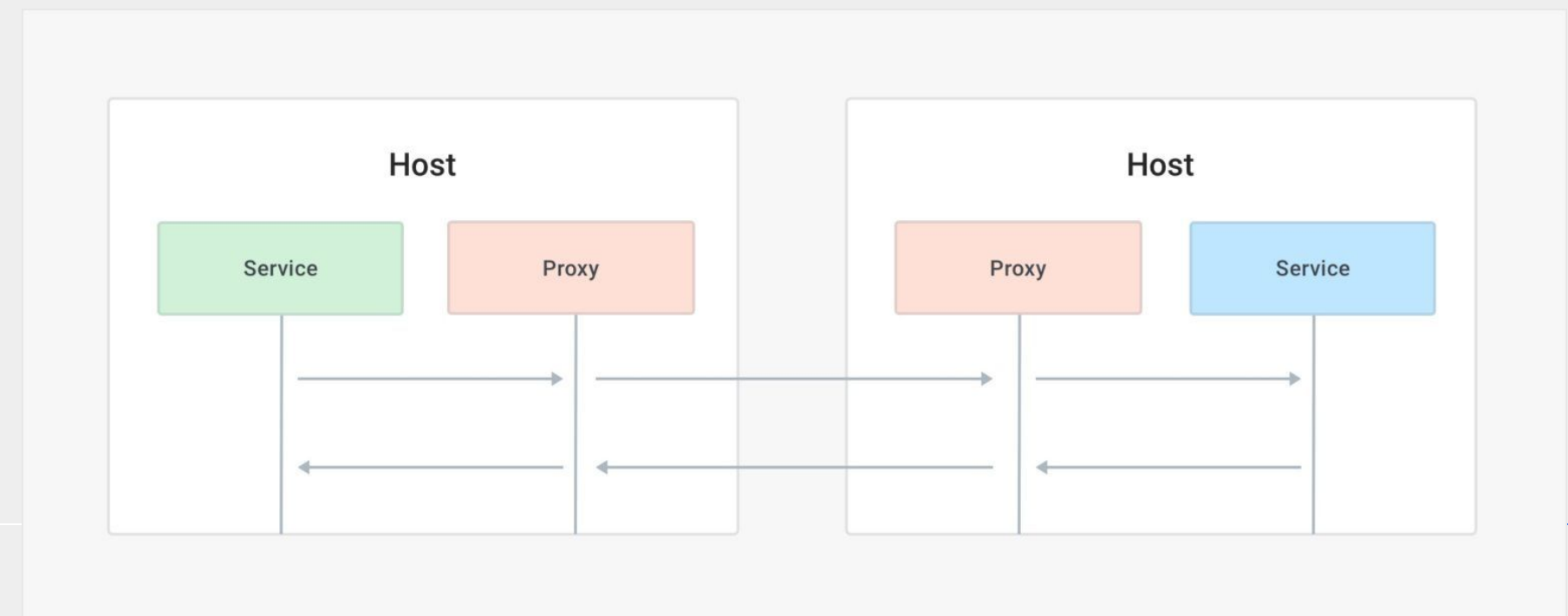
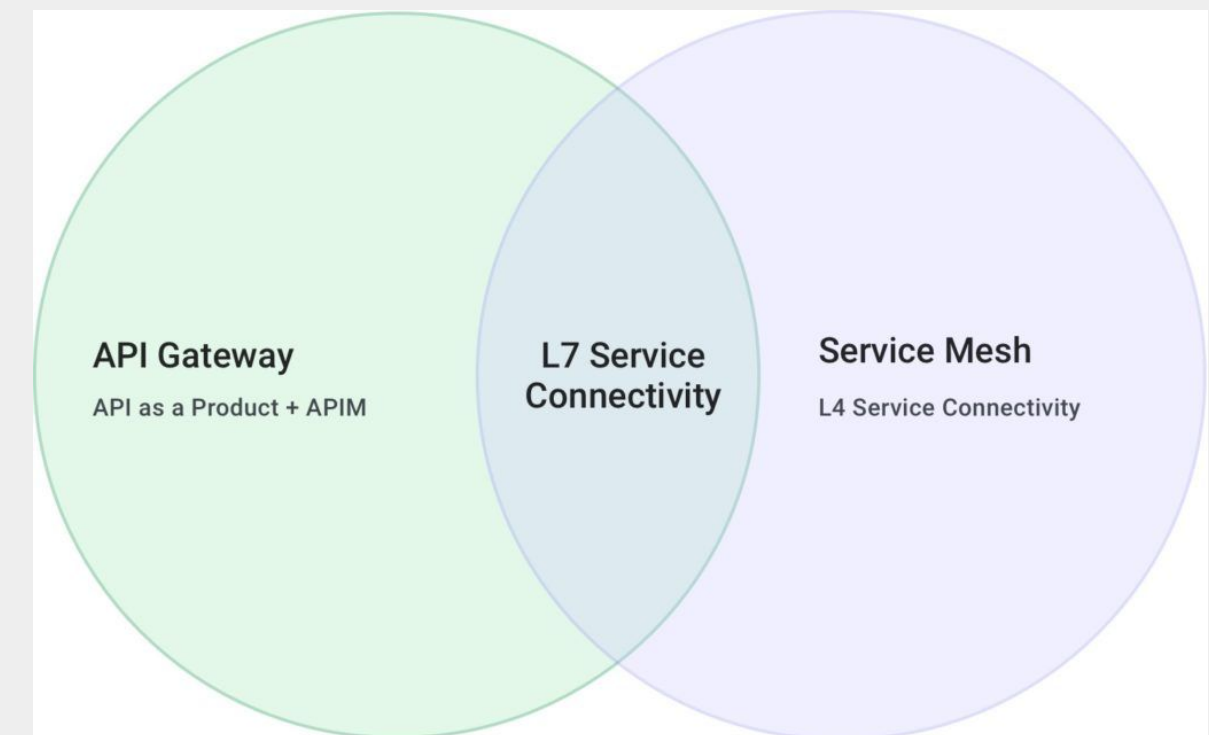
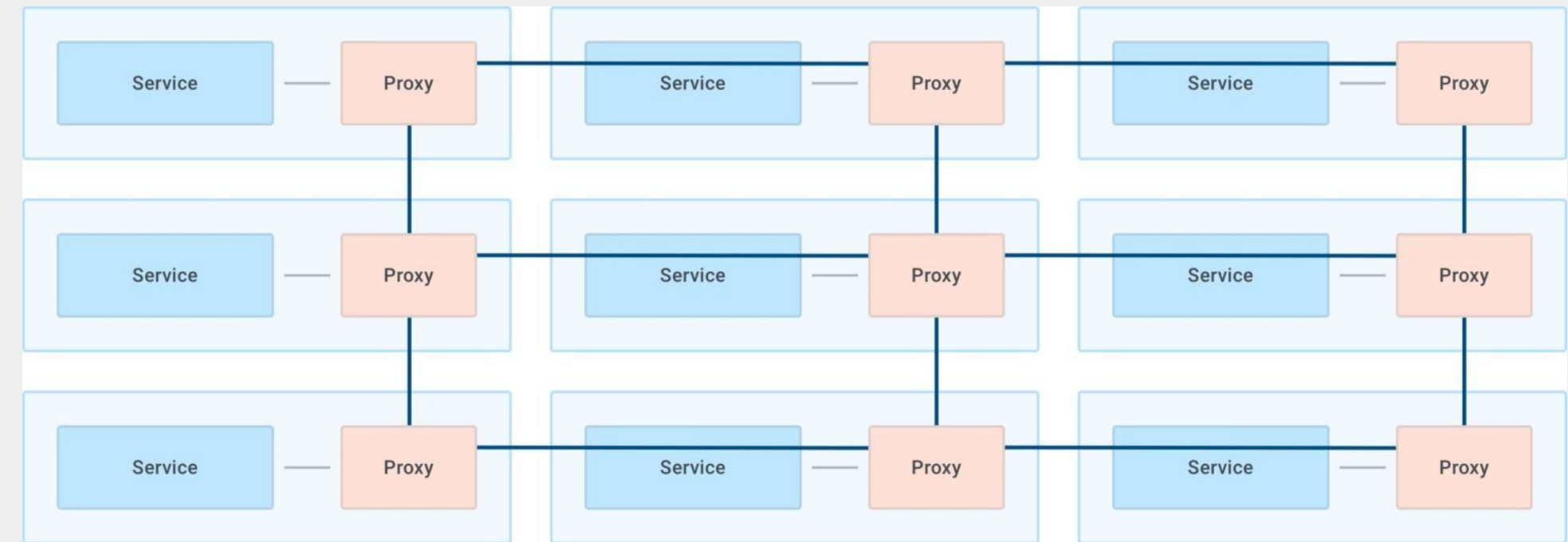
When should we use one?

## Independent of Architecture

With service mesh, we are identifying a pattern that fundamentally improves how we build service-to-service connectivity among two or more services running in our systems

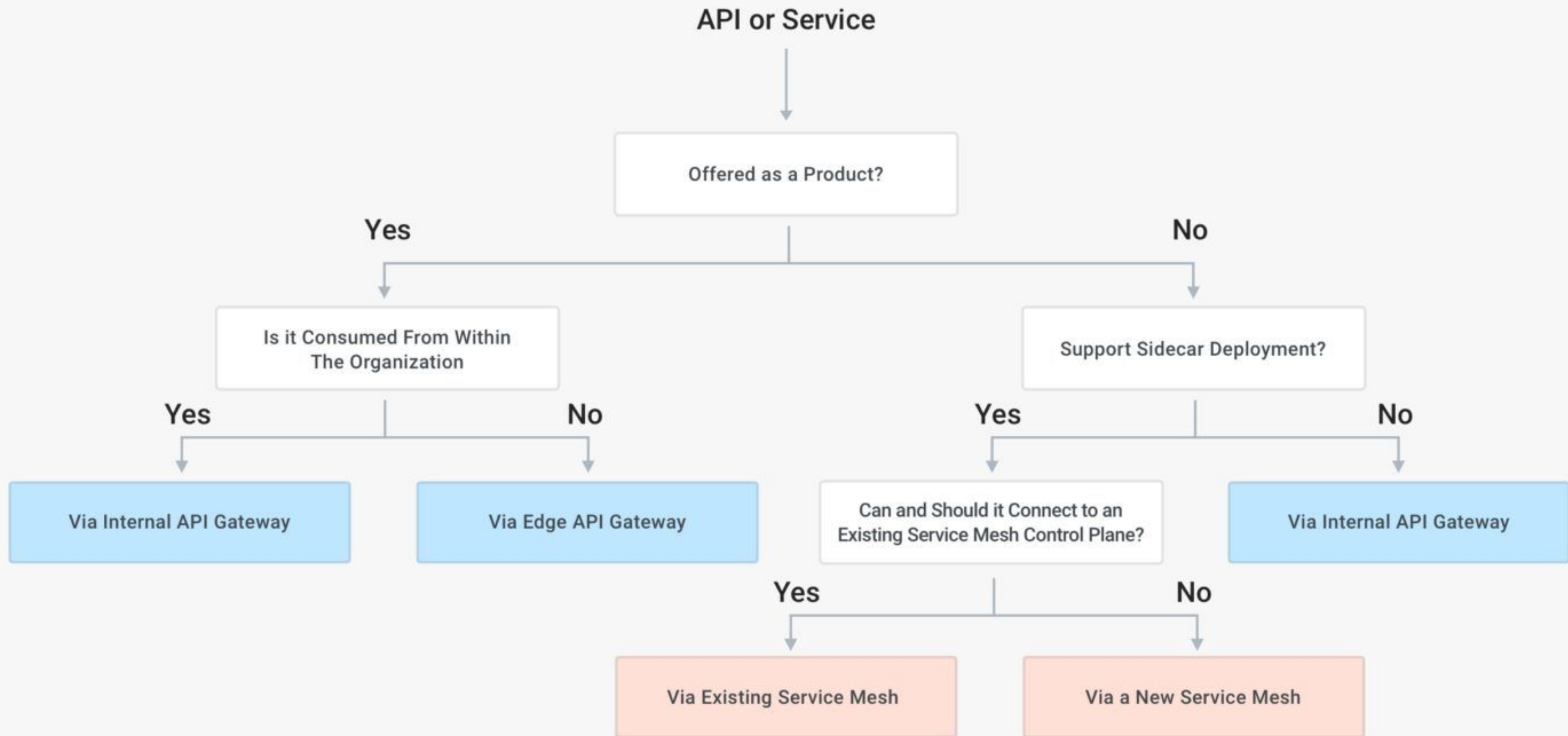
## Service Connectivity (at Layer 4 and 7)

A complete service mesh implementation will ideally support not just HTTP but also any other TCP traffic, regardless if it's north-south or east-west





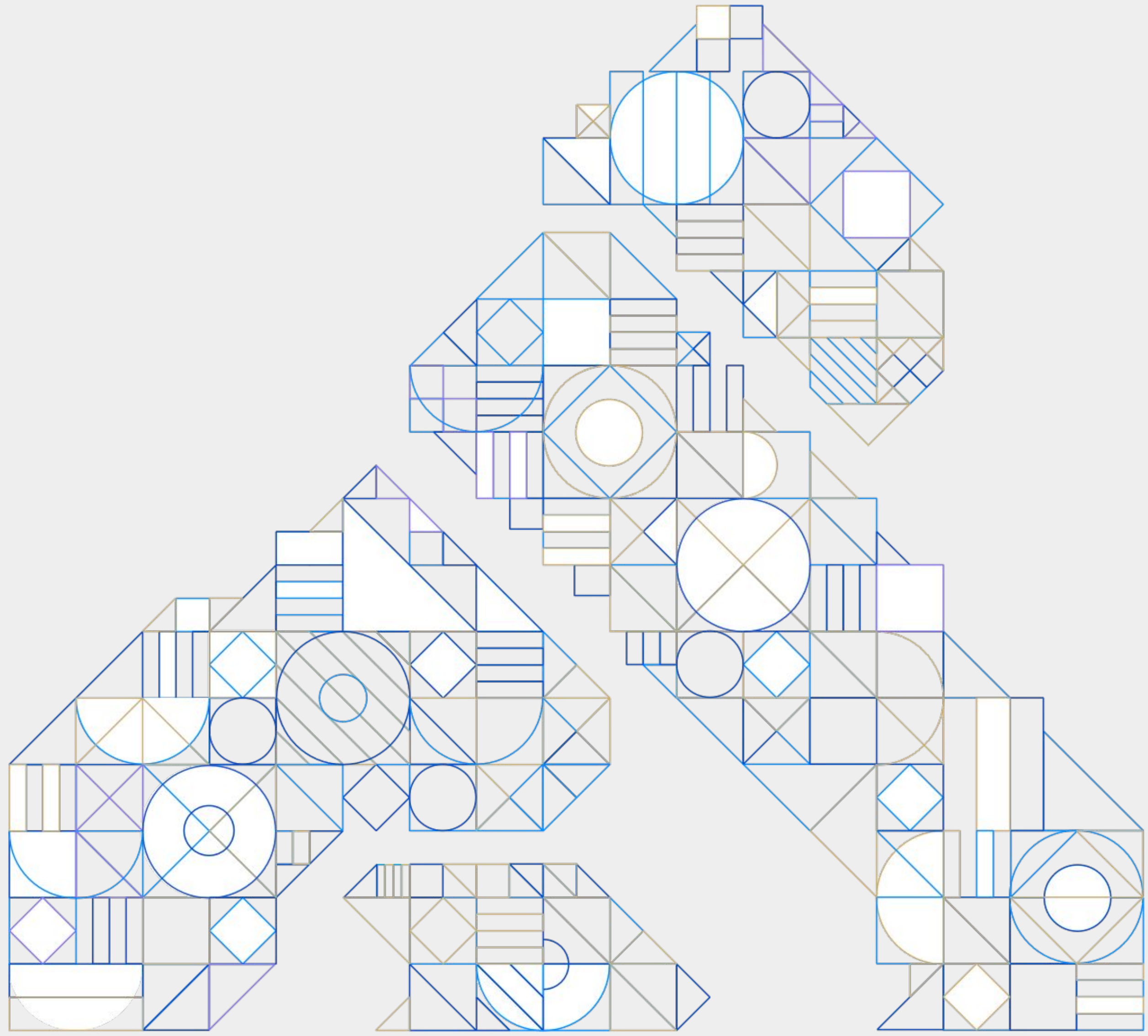
# Decision Chart





THE CLOUD CONNECTIVITY COMPANY

Live





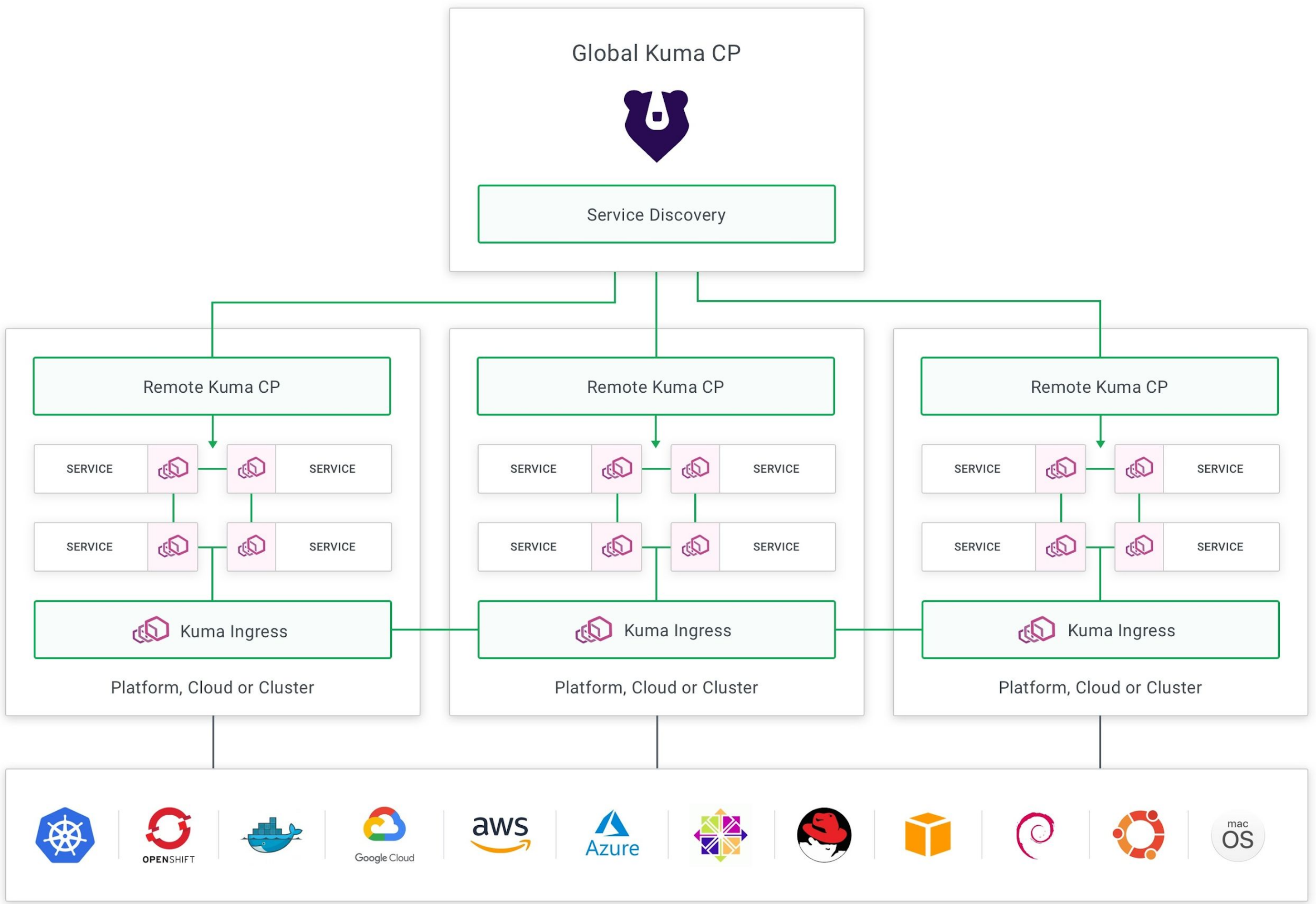
Where do we have service meshes?

Microservices =  **kubernetes** ?

Service Mesh =  **kubernetes** ?

[https://upload.wikimedia.org/wikipedia/commons/thumb/6/67/Kubernetes\\_logo.svg/1000px-Kubernetes\\_logo.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/6/67/Kubernetes_logo.svg/1000px-Kubernetes_logo.svg.png)

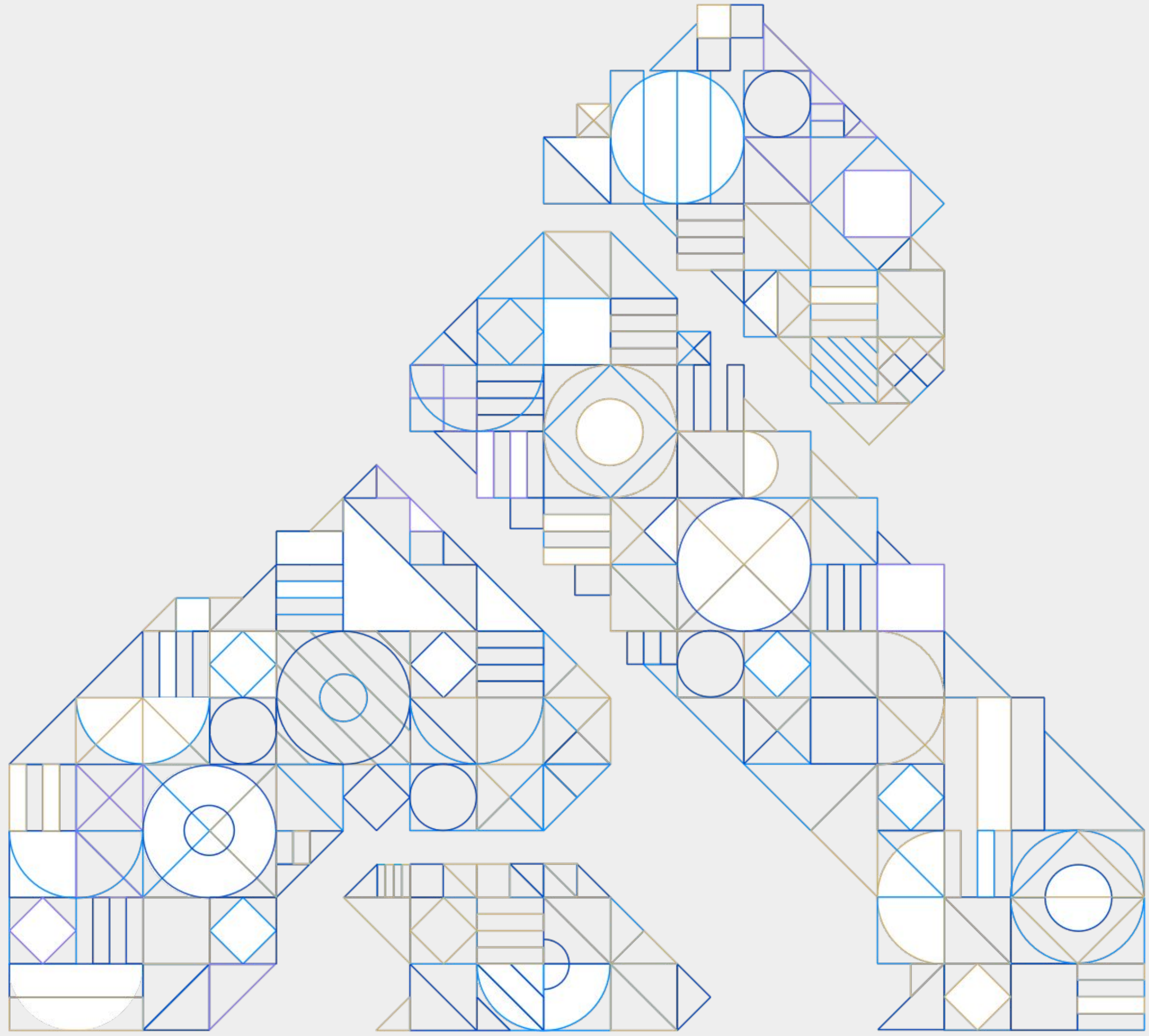
# Distributed setup





THE CLOUD CONNECTIVITY COMPANY

# Try it out





# Try it out - BYOD or online

<http://bit.ly/apidaysMesh1>

## Kubernetes Deployment Guide

### Introductions

In this directory, you will find the necessary files and instructions to get Kuma up and running in Kubernetes mode via Minikube.

When running on Kubernetes, Kuma will store all of its state and configuration on the underlying Kubernetes API Server, therefore requiring no dependency to store the data.

## Universal Deployment Guide

### Introductions

In this repository, you will find the necessary files and instruction to get Kuma up and running in universal mode via Vagrant.

When running in universal mode, there are two ways to store Kuma's state: in-memory or PostgreSQL. The first option stores all the state in-memory. This means that all data will be deleted on restart. This mode is only recommended for use when running locally. The second option is to utilize a PostgreSQL database to store state. The PostgreSQL database and schema will have to be initialized according to the installation instructions.

For the purposes of this demo we will use in-memory.

Note: only for live viewers of the workshop - won't be part of the public slides

The screenshot shows a Coda workspace titled "Kuma in Kubernetes Mode" with a "View only" button and an "Open in Coda" button. The main content area displays an "Overview" page with the following text:

### Overview

#### Introduction

The "service mesh" deployment pattern has become a very popular deployment architecture pattern. There are a number of benefits when utilizing a service mesh deployment pattern including:

- Observability of service communication
- Securing connections between services
- Traffic control, among others.

All of these benefits are realized without making changes to the underlying application code. In a service mesh, network communication from one service to another passes through proxies, known as side cars, which attach to each service. By using side cars, service meshes perform different functions and apply policies unbeknown to the services. Examples of

On the right side of the workspace, there is a terminal window with a black background and white text. It displays a "WELCOME TO" message followed by a grid of hash symbols (#) and a prompt character (\$).



**Kong** SUMMIT  
DIGITAL 2020

# Connecting Beyond the Cloud

October 7 - 9, 2020 | Digital Conference

Tickets are free with \$10 Donation to  
Doctors Without Borders.

[Kongsummit.com](https://kongsummit.com)



# Connecting Beyond the Cloud

October 7 - 9, 2020 | Digital Conference

## DISCOVER



Strategies and tactics for ensuring reliable connectivity across APIs, Kubernetes, microservices, service meshes.

## OPERATE



Hands-on technical workshops, real-world experiences, thought leadership, and exciting Kong product announcements.

## LEARN



Modernize architectures and create business value. Explore how to connect across clouds and beyond!