Tyk

Service Proxy, API Gateway, Service Mesh

What’s the difference? What do I need?

Ahmet Soormally, Carol Cheung

KubeCon | CloudNativeCon

North America 2023

WHAT'S TO COME

Intro

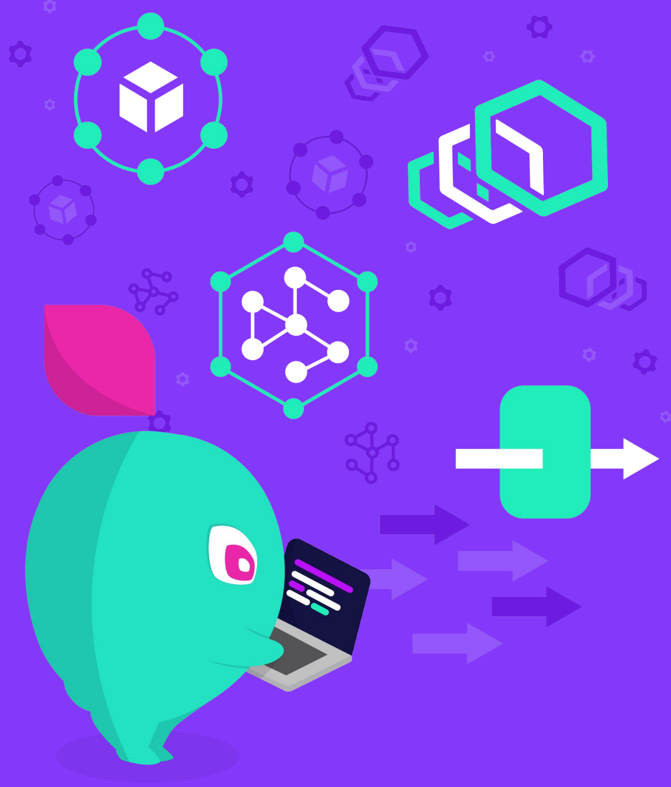
01 **The rise of microservices**
and the challenges they bring

02 **Service proxy**
What is it? How is it used? Deployment topologies

03 **Service mesh**
What is it and how is it used? Deployment topologies

04 **API gateway**
What is it and how is it used? Deployment topologies

05 **Summary**
What’s the difference? What do I need?



WELCOME

Speakers

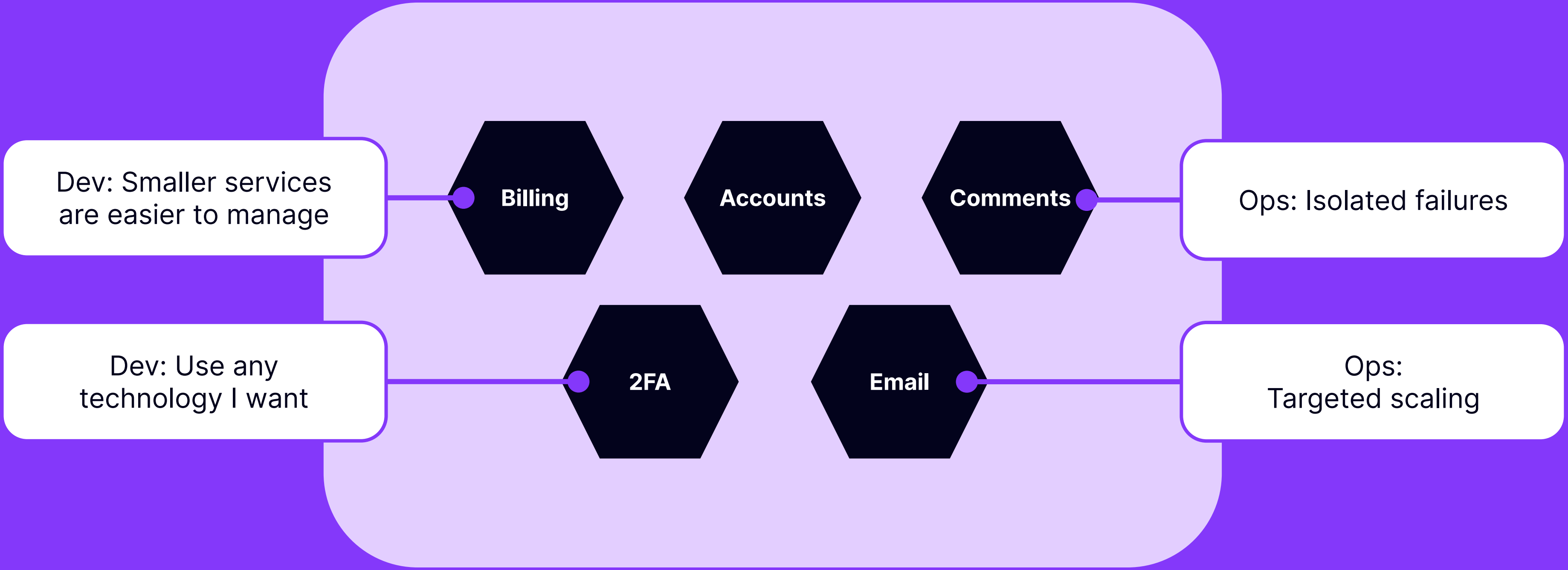
Ahmet Soormally

Head of R&D,
Tyk Technologies

Carol Cheung

Senior Product
Manager -
Platform & Ops,
Tyk Technologies

The rise of microservices



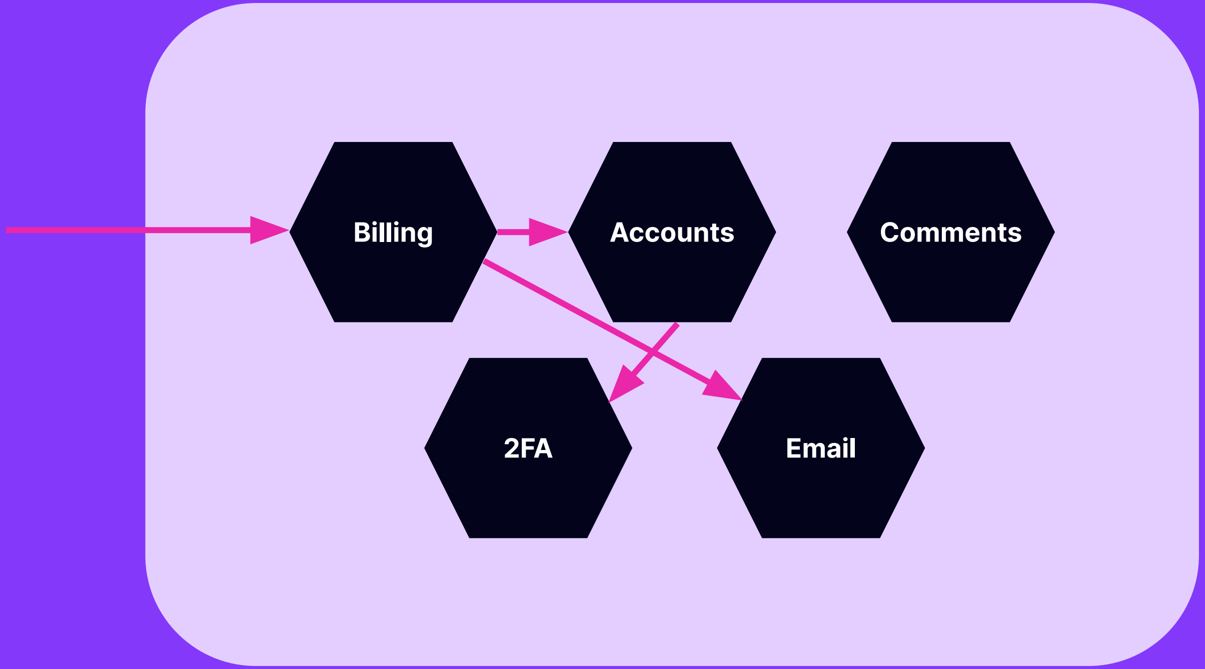
Dev: Smaller services are easier to manage

Dev: Use any technology I want

Ops: Isolated failures

Ops: Targeted scaling

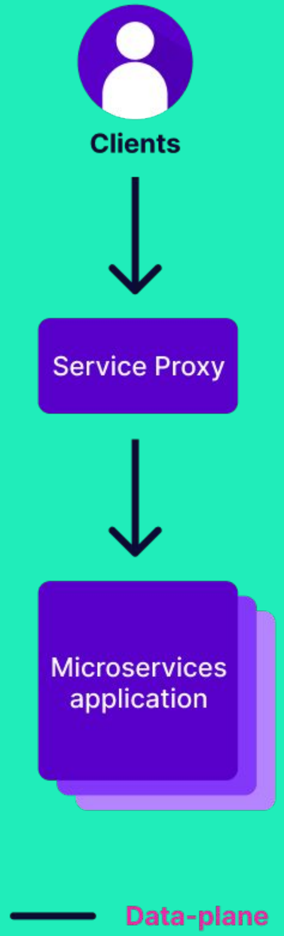
The rise of microservices



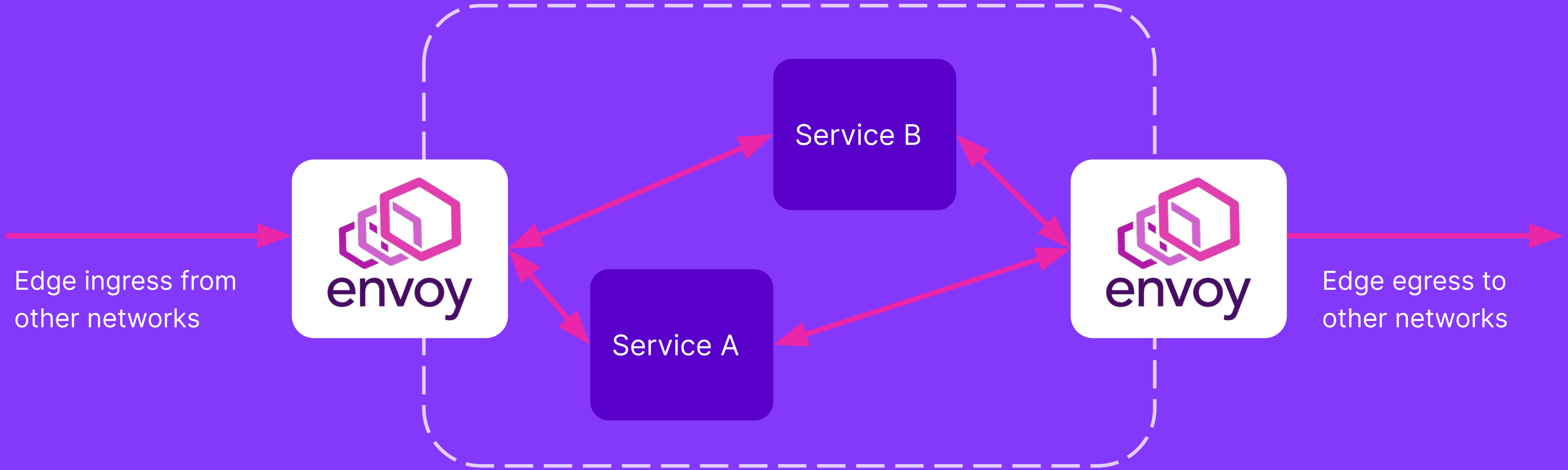
- Increased number of microservices
- Increased number of service to service communications
- Network communication is unreliable, takes time, and susceptible to attack

Service proxy

A **service proxy** intercepts traffic to or from a given service, applies some logic to it, then forwards that traffic to another service. It essentially acts as a **“go-between”** that collects information about network traffic and/or applies rules to it.



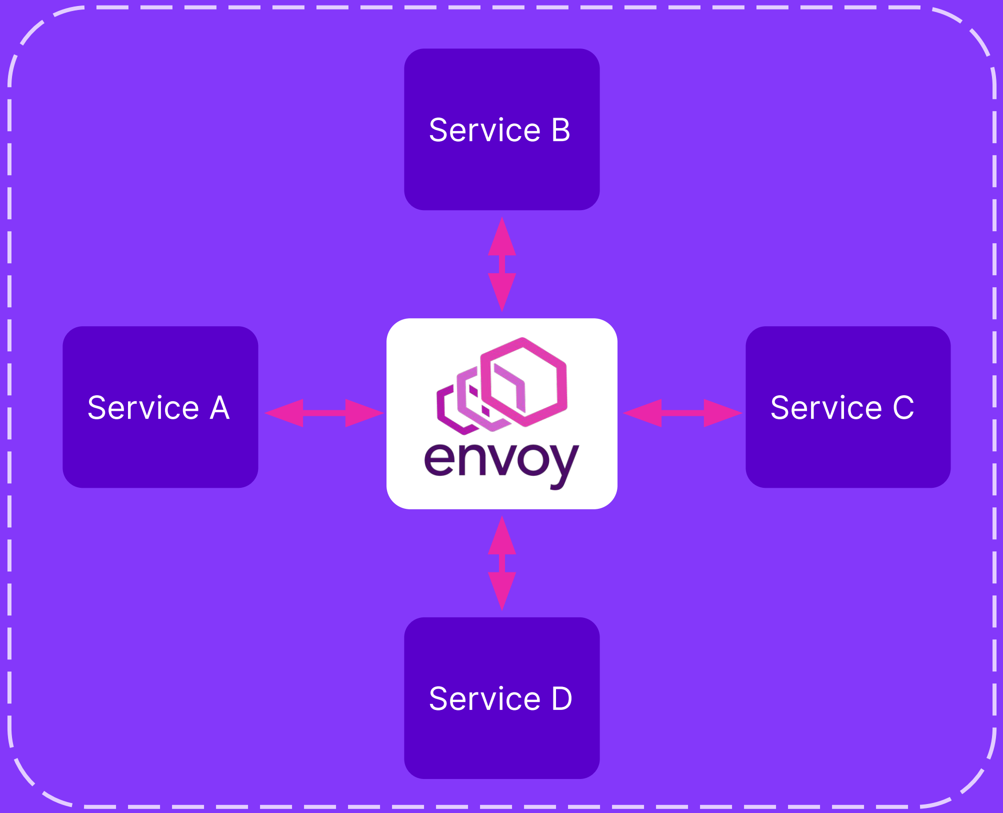
Ingress/egress reverse proxy



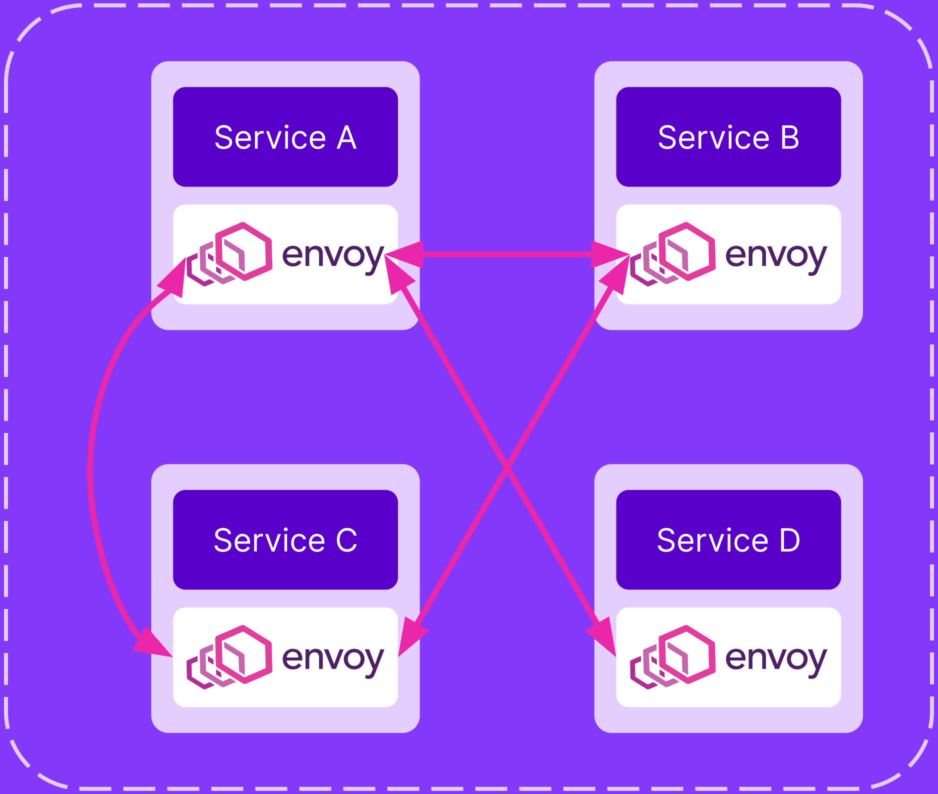
Edge ingress from other networks

Edge egress to other networks

Internal load balancer

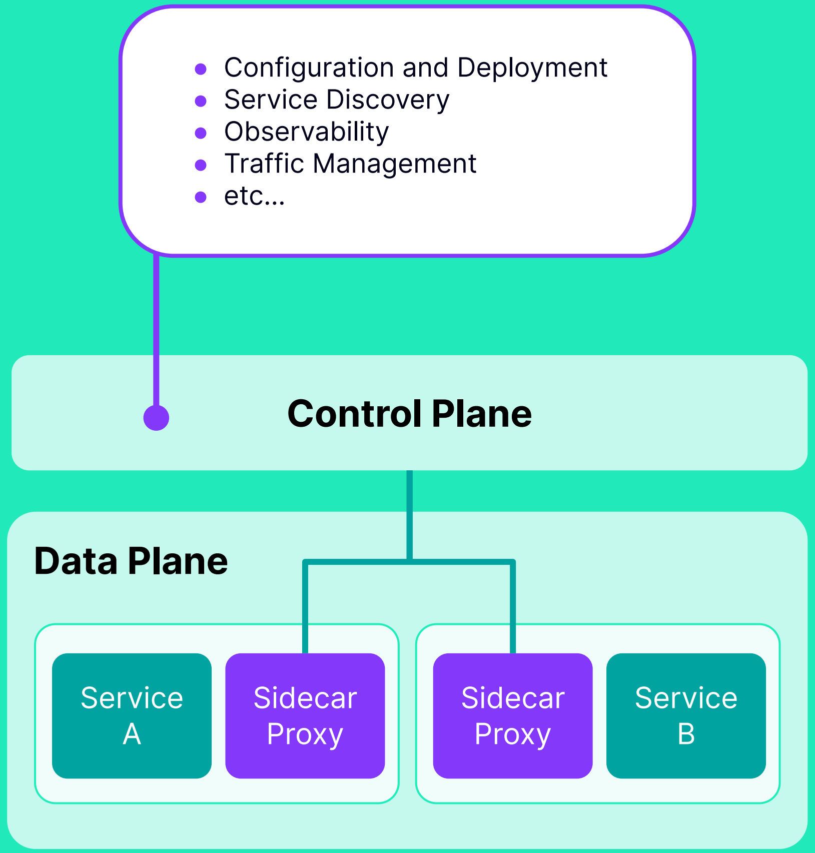


Sidecar proxy

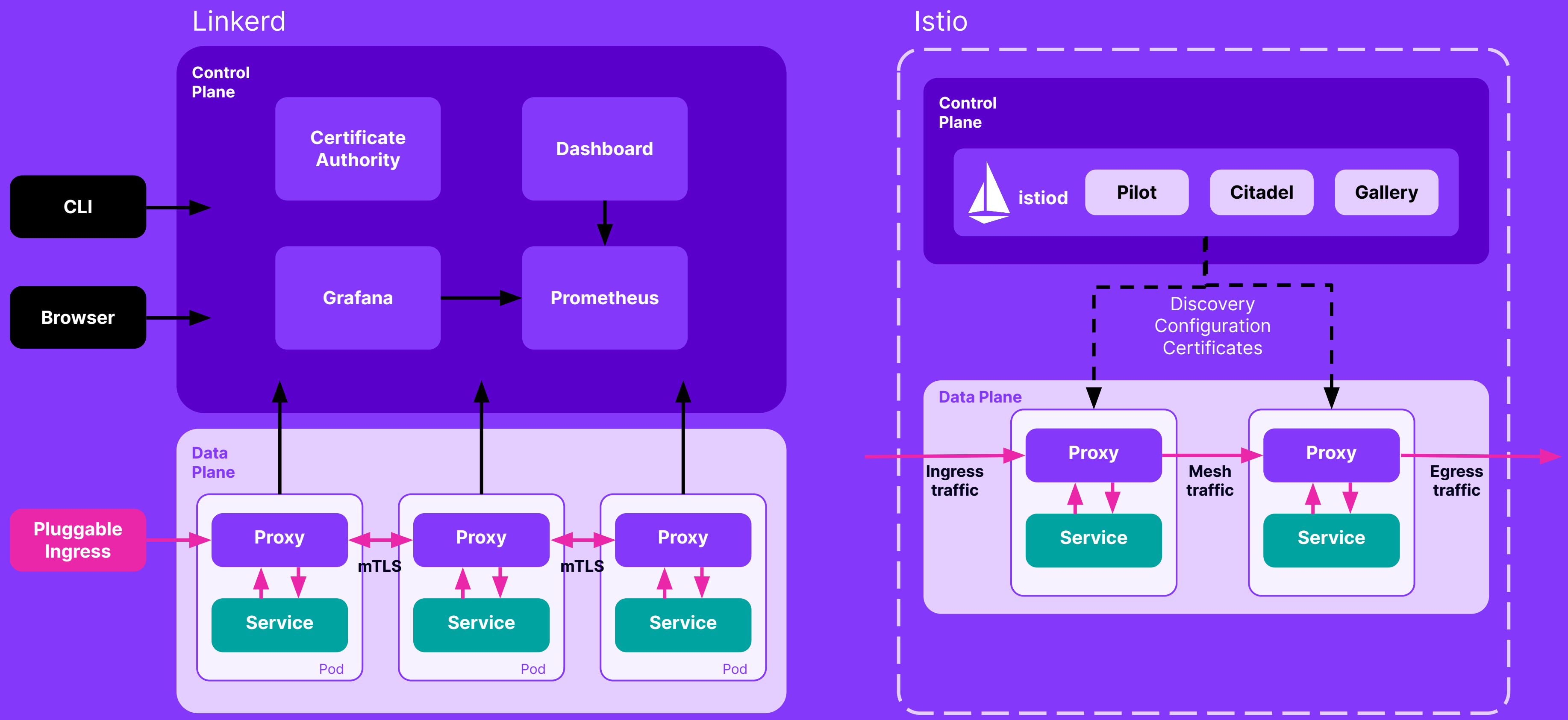


Service mesh

Service meshes manage traffic (i.e. communication) between services. They enable platform teams to add reliability, observability, and security features **uniformly** across all services running within a cluster without requiring any code changes.



Service mesh



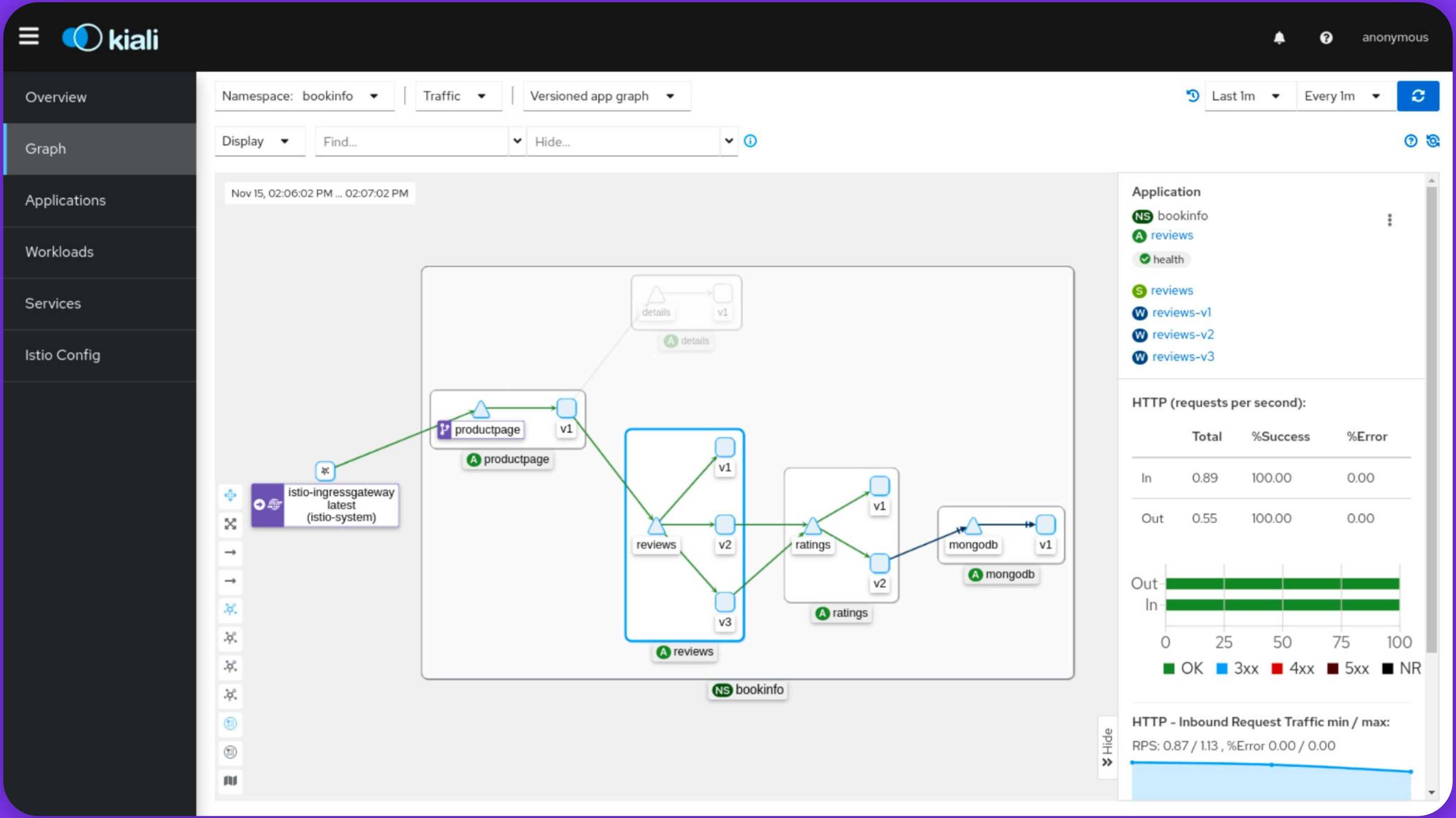
MAIN FEATURES

Service mesh

Dynamic service discovery

Service to service communication security

Traffic management



Namespace: bookinfo

Traffic

Versioned app graph

Last 1m

Every 1m

Nov 15, 02:06:02 PM ... 02:07:02 PM

Application

NS bookinfo

A reviews

health

S reviews

W reviews-v1

W reviews-v2

W reviews-v3

HTTP (requests per second):

	Total	%Success	%Error
In	0.89	100.00	0.00
Out	0.55	100.00	0.00

HTTP - Inbound Request Traffic min / max:

RPS: 0.87 / 1.13 , %Error 0.00 / 0.00

MAIN BENEFITS

Service mesh

Automate DevSecOps needs like observability, security, and traffic management.

Reduce Troubleshooting Time

Provides observability data about how microservices communicate, including metrics, logs and traces.

Zero Trust Security

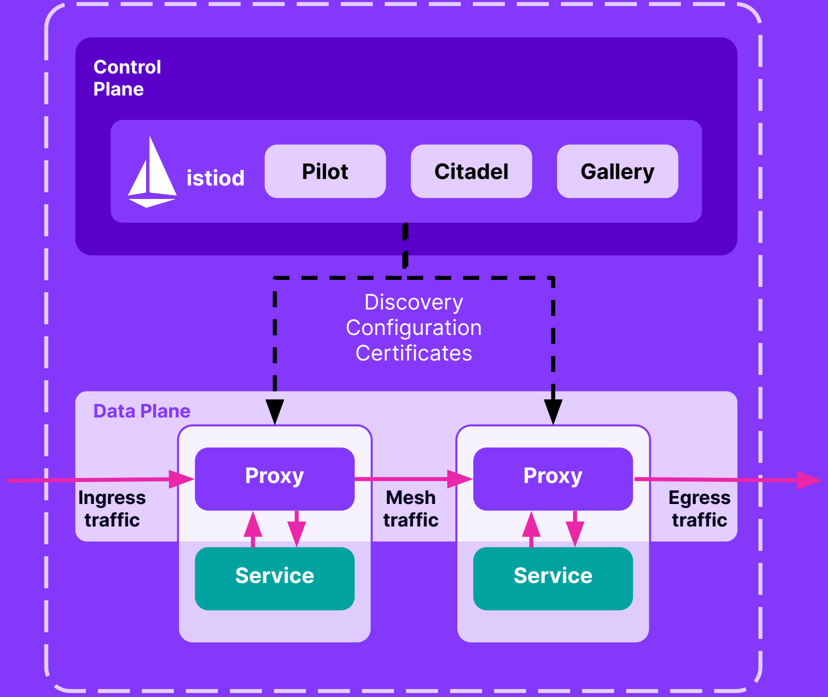
Provides end-to-end encryption and mutual authentication between microservices, helping to secure communication channels. RBAC rules.

Resilience

Provides advanced traffic management like load balancing, retries, timeouts, which can help optimize traffic flow between microservices, and ensure better resource utilization and fault tolerance.

OPERATIONAL COMPLEXITY

Service mesh



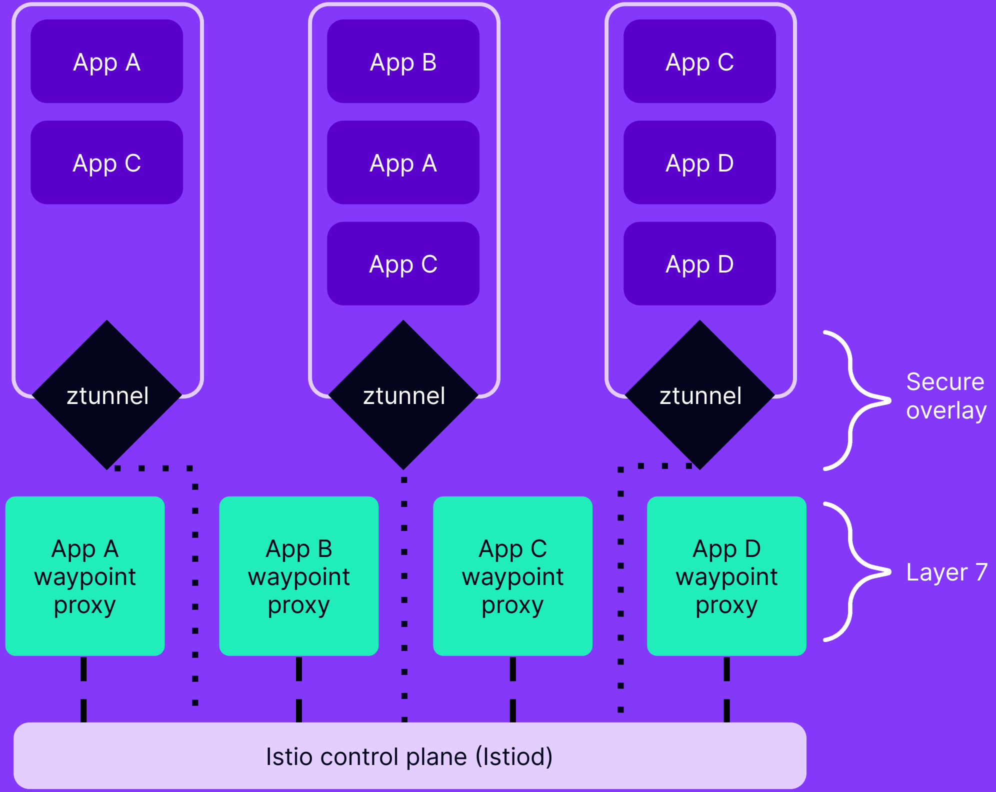
- Container race conditions during startup
- Performance impact
- Resource and Cost impact

Sidecarless architecture

(Istio Ambient Mesh)

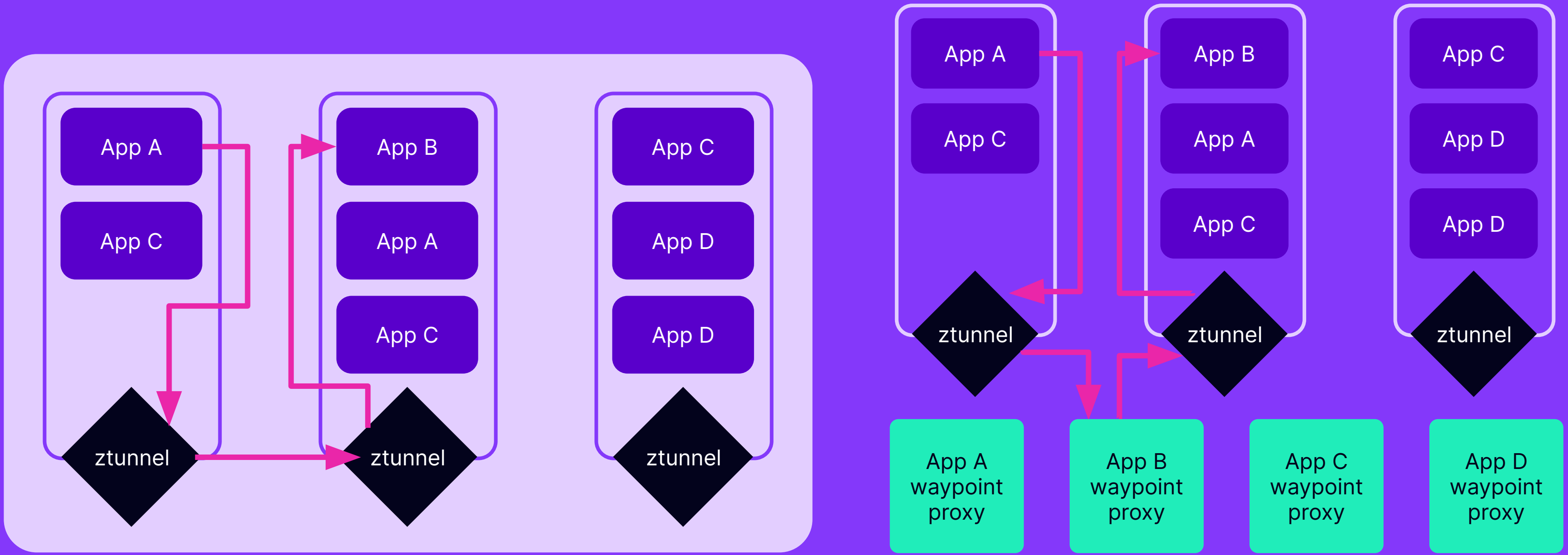
- **Secure overlay layer (L4)**
  TCP Routing, mTLS tunneling, simple authorization policies, TCP metrics & logic

- **Waypoint proxy (L7)**
  HTTP routing & load balancing, circuit breaking, rate limiting, rich authorization policies, HTTP metrics, access logging, tracing



Sidecarless architecture

(Istio Ambient Mesh)



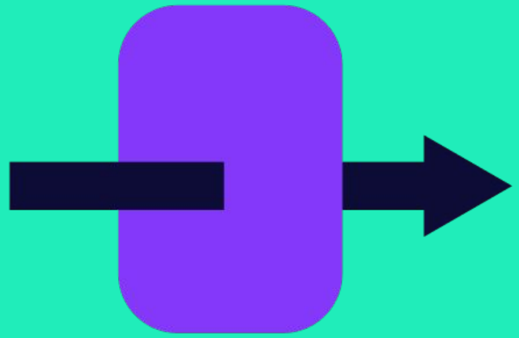
WHAT'S TO COME

API Gateways

01 Key capabilities and use-cases

02 Deployment architectures & topologies

03 API management



01 **Key capabilities**

02 Deployment architectures

03 API management

L4 security

- TLS
- mTLS
- IP allow / deny
- WAF
- TCP routing
- IP Rate limiting

01 **Key capabilities**

02 Deployment architectures

03 API management

L7 security

- External authentication & authorization
- REST
  - Method / path protection
  - Request / response validation
- GraphQL
  - Operation protection
  - Field-based permissions
  - Query depth limiting
  - Query complexity limiting
  - Enable / disable introspection
- gRPC
  - Service & service method authorization
- Asynchronous APIs
  - Rate limiting based on number of events or data
  - Pub/sub authorization
  - Webhooks

01 **Key capabilities**

02 Deployment architectures

03 API management

Endpoint optimization

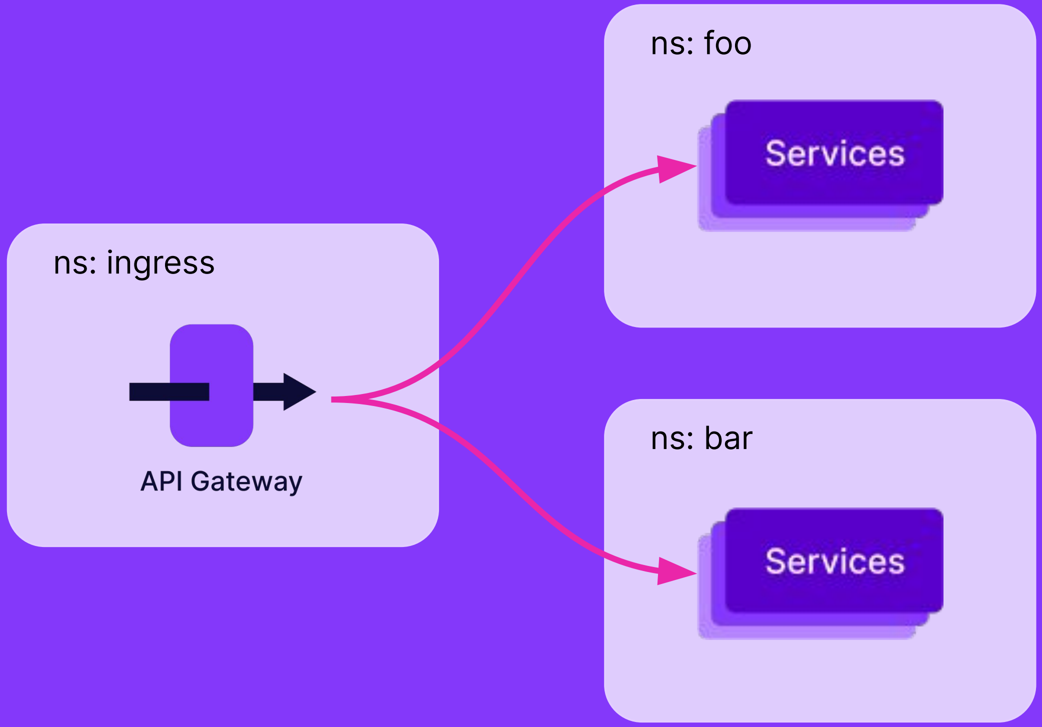
- BFF Pattern / Optimizing endpoints
- Request & Response composition
- API Transformation
- Migrating from monolithic to microservice architecture

01 Key capabilities

02 Deployment architectures

03 API management

Shared gateway

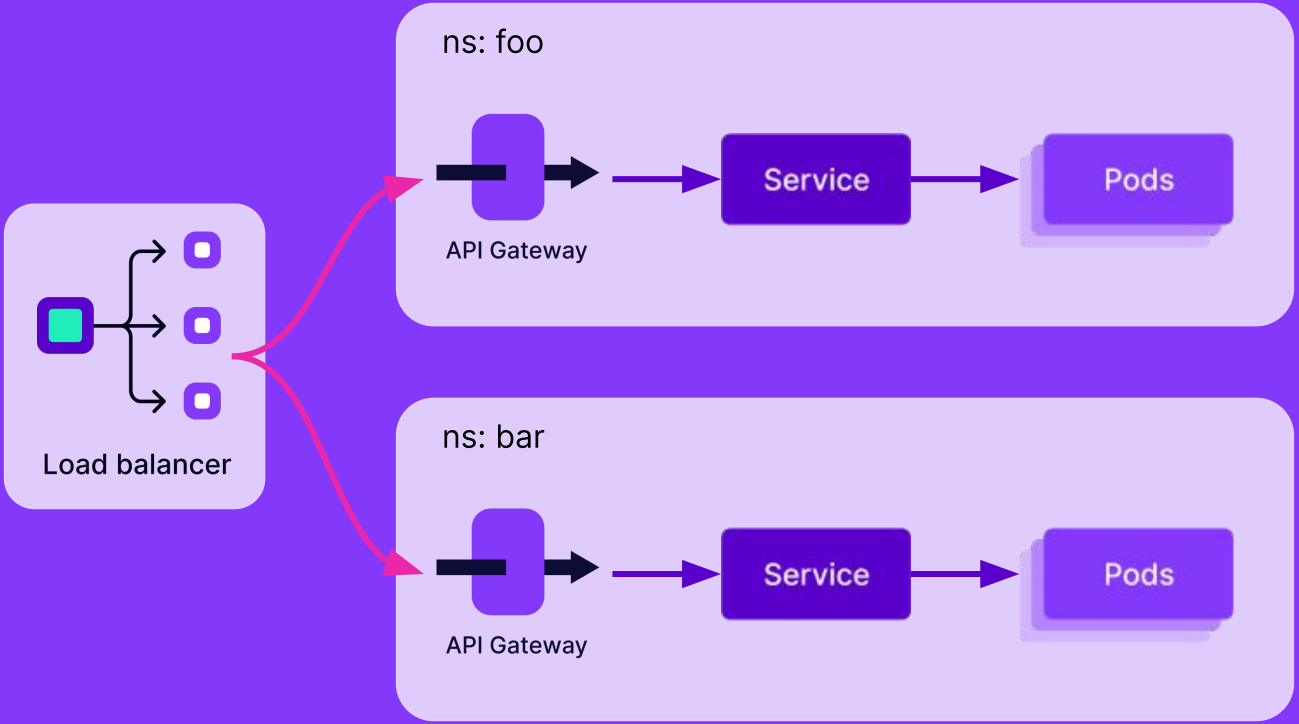


01 Key capabilities

02 Deployment architectures

03 API management

Gateway per service

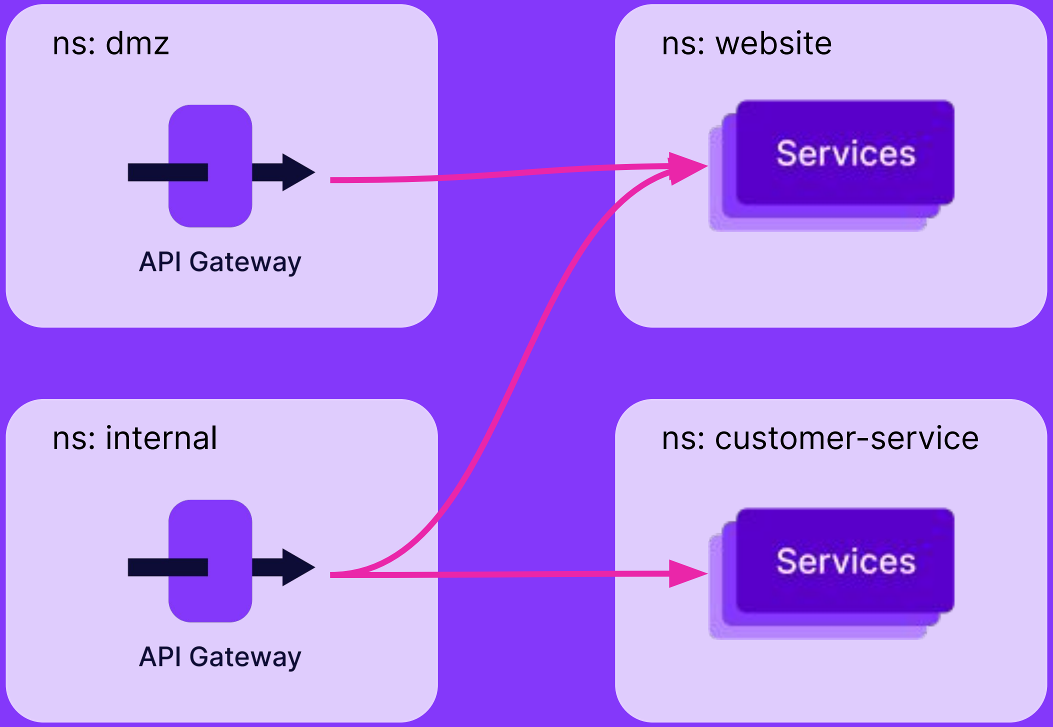


01 Key capabilities

02 Deployment architectures

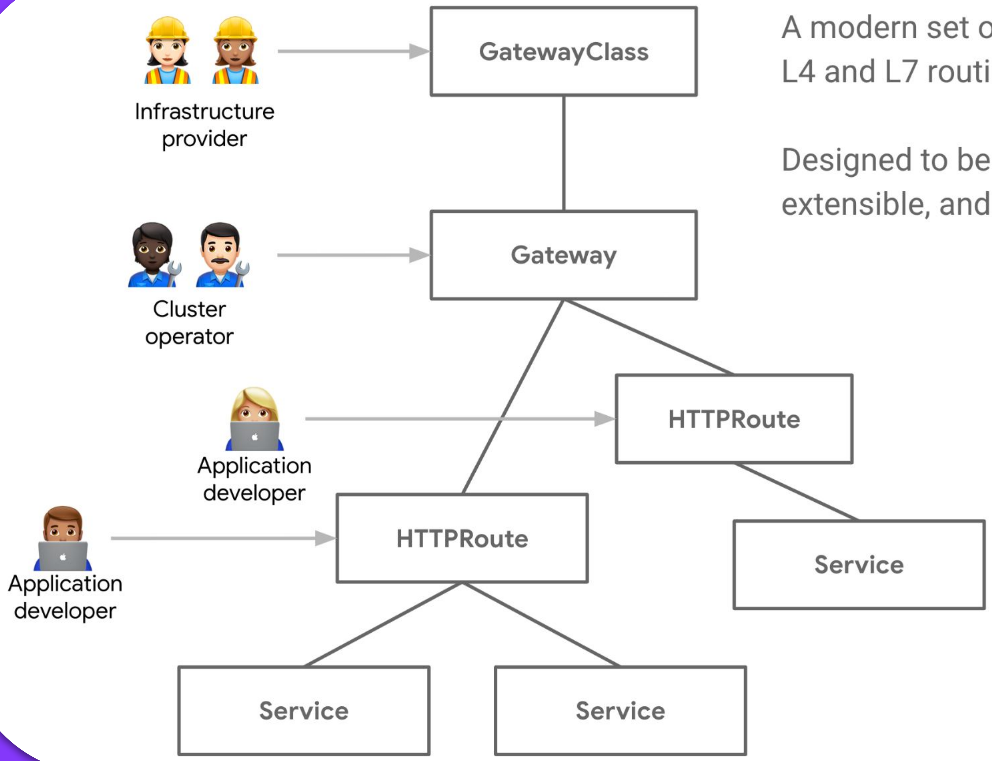
03 API management

Gateway sharding



Gateway API

https://sched.co/1R2qM



A modern set of APIs for deploying L4 and L7 routing in Kubernetes

Designed to be generic, expressive, extensible, and role-oriented

01 Key capabilities

02 Deployment architectures

03 API management

- Development workflows and testing
- Discovery, subscriptions, documentation, specifications
- Version Management & routing
- API analytics

TO SUM IT UP

What’s the difference?
What do I need?

Service Proxy

General Service Proxy

Service Mesh

For Platform Engineers / DevOps / SREs

- Uniformly apply policies for security and governance needs
- Gain detailed insights into application performance for better troubleshooting and optimization

API Gateway

For API Developers and API Product Owners

- Leverage APIs to drive innovations
- APIs as strategic business asset

Thank you!

Come talk to us to continue the discussion or reach out:

Ahmet Soormally

https://www.linkedin.com/in/ahmet-soormally/

Carol Cheung

https://www.linkedin.com/in/carol-c-01111324