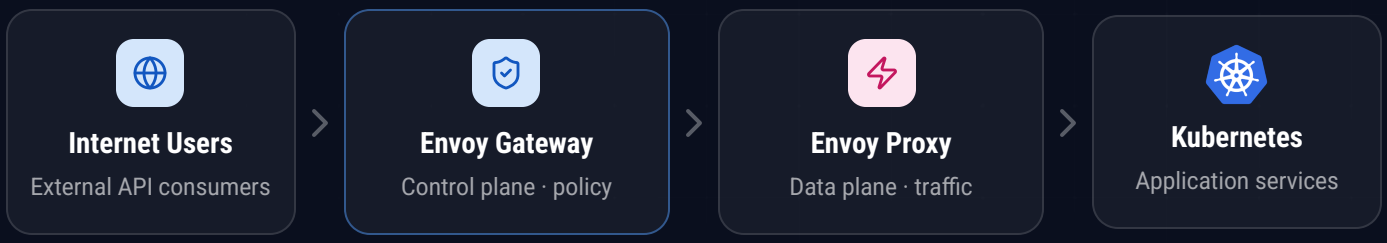
CLOUD-NATIVE NETWORKING

Envoy Gateway

Architecture, Best Practices & Enterprise Deployment Recommendations

Modern organizations need a **scalable, secure, and Kubernetes-native** approach to API traffic management. Envoy Gateway simplifies the deployment and operation of Envoy Proxy while embracing the Kubernetes Gateway API standard — so platform teams can build resilient, future-ready application networking.



10M+
Requests/sec at fleet scale

0.5ms
P99 latency @ 100K RPS

99.99%
Availability with multi-instance HA

10x
Faster ingress provisioning

BUILT ON OPEN STANDARDS

- **Kubernetes** — GATEWAY API
- **Envoy** — DATA PLANE
- **Backstage** — DEV PORTAL
- **CNCF** — ECOSYSTEM

AUDIENCE · PLATFORM ENGINEERS · SOLUTION ARCHITECTS · ENGINEERING MANAGERS · DIRECTORS

EXECUTIVE SUMMARY

Modernize API traffic on open standards

Architectures have shifted to microservices and Kubernetes. Traditional gateways struggle to deliver the flexibility, scale, and operational simplicity these environments demand. **Envoy Gateway closes that gap.**

ENVOY GATEWAY COMBINES

Envoy Proxy Performance

Battle-tested reliability handling millions of requests per second.

Gateway API Standardization

A portable, vendor-neutral Kubernetes standard.

Cloud-Native Operations

GitOps and declarative, reviewable workflows.

Open-Source Flexibility

No lock-in. Community-driven and CNCF-aligned.

AND NOW, OUT OF THE BOX

- Built-in scale
- Sub-millisecond latency
- Developer portal
- API lifecycle & governance

Key Benefits

- Standardize ingress across teams
- Improve application security posture
- Increase operational efficiency
- Reduce vendor lock-in
- Enable future platform growth

10×
Faster provisioning

99.99%
HA availability

Open standards and proven cloud-native technology — one consistent ingress model across every cluster and team.

IMESH · ADOPTION GUIDE

THE PROBLEM SPACE

The API gateway challenge

Many organizations still rely on gateways designed before Kubernetes became the dominant deployment model — creating friction as teams modernize.

COMMON CHALLENGES

Rising licensing costs
Per-instance pricing that scales against you.

Complex operating models
Appliance-style config and ticket-driven change.

Vendor lock-in
Proprietary APIs that resist portability.

Fragmented ingress
Inconsistent architecture across teams.

BUSINESS IMPACT

Slower application delivery
Releases gated on manual gateway changes.

Increased operational burden
More toil, more on-call, more risk.

Inconsistent security controls
Policy that drifts between environments.

Reduced developer productivity
Teams wait instead of self-serve.

TWO OPERATING MODELS

Legacy gateway — Traditional

- ✗ Static, appliance-style configuration
- ✗ Manual, ticket-driven changes
- ✗ Proprietary APIs & licensing
- ✗ Bolted-on to Kubernetes

→

Cloud-native gateway — Modern

- ✓ Declarative, API-driven topology
- ✓ GitOps-managed, self-service
- ✓ Open Gateway API standard
- ✓ Kubernetes-native by design

THE SOLUTION

Why Envoy Gateway

An open-source project that simplifies managing Envoy Proxy through the Kubernetes Gateway API — purpose-built for modern platforms.

Kubernetes Native

Built specifically for Kubernetes environments and their lifecycle.

Open Standards

Based on the vendor-neutral Gateway API specification.

Proven Technology

Powered by Envoy Proxy — a widely adopted cloud-native data plane.

Operational Simplicity

Far less complexity than managing raw Envoy configuration.

Open Source

Community-driven, transparent, and free of vendor lock-in.

Future Ready

Aligned with CNCF direction and platform-engineering practice.

One control plane. One open standard.

One consistent ingress model — across every cluster and every team.

POWERED BY

Envoy PROXY

Kubernetes GATEWAY API

PERFORMANCE

Built for scale & speed

Envoy's data plane was engineered for the most demanding traffic on the planet. The same engine powers your gateway — so capacity and latency are never the bottleneck.

10M+

Scale — millions of RPS

Horizontally scale the Envoy fleet to absorb millions of requests per second without re-architecting.

0.5ms

Speed — 0.5ms P99 latency

Process 100,000 requests per second at just 0.5 ms P99 latency. Speed your users feel.

100%

Secure by default

TLS, mTLS, OIDC/OAuth2, rate limiting and WAF enforced at the edge — out of the box.

Simple to complex — out of the box

From basic L7 routing to high-throughput event-streaming backends, Envoy Gateway handles it without bolt-ons or custom proxies.

- Apache Kafka
- Amazon Kinesis
- Zerobus
- gRPC & HTTP/2

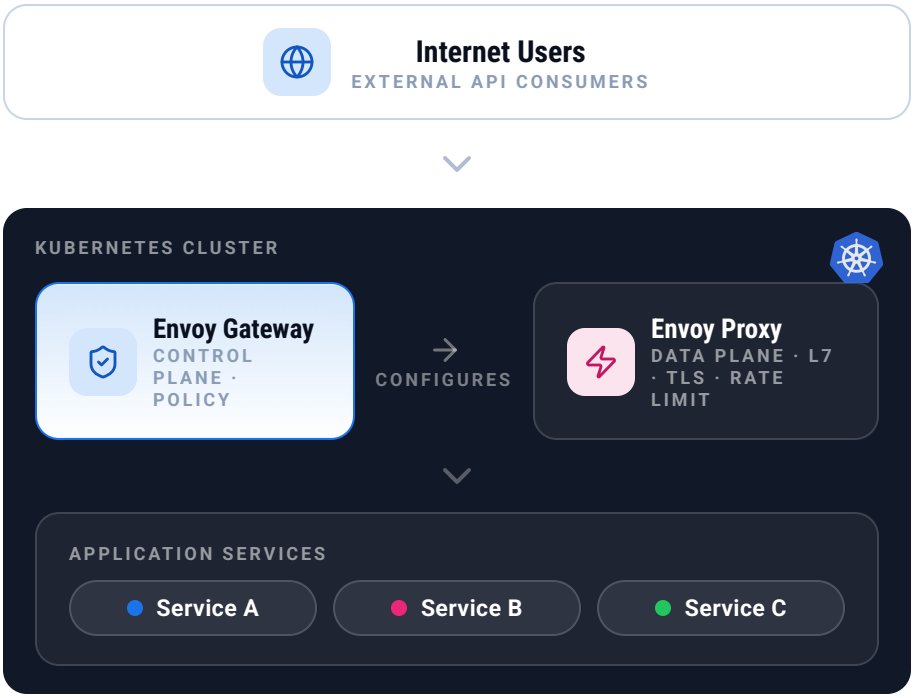
Connection pooling & circuit breaking built in

Adaptive load balancing & retries

Backpressure-aware streaming

REFERENCE ARCHITECTURE

Where Envoy Gateway fits



Key Outcomes

- Centralized ingress management
- Consistent governance
- Scalable traffic processing
- Improved developer experience

CROSS-CUTTING CONCERNS

Security
TLS, authn/authz, policy

Traffic Routing
Host / path / header rules

Observability
Metrics, logs & traces

Policy Enforcement
Rate limits & governance

CAPABILITY MATRIX

Core capabilities

A single gateway layer that unifies routing, security, visibility, performance, and platform operations under one declarative model.

L7	6	100%	0
Full Layer-7 control	Domains, one control plane	Declarative configuration	Vendor lock-in

Traffic Management

- Advanced routing
- Traffic splitting
- Canary
- Blue-green

Security

- TLS / mTLS
- AuthN
- AuthZ
- Rate limiting

Observability

- Metrics
- Logging
- Tracing

Scale & Speed

- Millions RPS
- 0.5ms P99
- Autoscaling

API Lifecycle & Governance

- Versioning
- Catalog
- Policy

Platform Operations

- Multi-tenancy
- GitOps
- Dev portal

Open source, no lock-in. Every capability above ships in the upstream project — backed by IMESH for enterprise support.

UNDER THE HOOD

Architecture deep dive

Envoy Gateway separates a Kubernetes-aware **control plane** from a high-performance **data plane** — keeping configuration declarative and traffic handling fast.

Control Plane
ENVOY GATEWAY · POLICY

Translates Kubernetes Gateway API resources into validated Envoy configuration and implements policy — continuously reconciling desired state. **It never sits in the request path.**

Data Plane
ENVOY PROXY · TRAFFIC

Envoy Proxy receives and routes live application traffic with high performance, resilience, and rich Layer-7 capabilities at scale.

The control plane configures the Envoy fleet via xDS APIs — configuration stays declarative while the data plane handles traffic independently.

REQUEST LIFECYCLE — DECLARATIVE IN, DETERMINISTIC OUT

1. **Request Enters** — Hits the Envoy Proxy
2. **Policy Eval** — Auth & rules applied
3. **Routing** — Destination selected
4. **Backend** — Reaches the service
5. **Telemetry** — Signals emitted

Every Gateway API change reconciles into a consistent Envoy state — with no manual drift.

IN PRODUCTION

Enterprise use cases

From modernizing legacy infrastructure to governing AI traffic and high-volume event streams, Envoy Gateway adapts to the most demanding enterprise scenarios.

API Modernization

Move legacy API gateway infrastructure onto an open, cloud-native foundation.

Ingress Standardization

One consistent Kubernetes ingress model across the organization.

Multi-Tenant Platforms

Secure application delivery across many teams and business units.

AI & LLM Applications

Secure and govern AI traffic entering enterprise environments.

Event Streaming & Data

Front Kafka, Kinesis and Zerobus pipelines with secure, observable ingress.

Multi-Cluster Deployments

Consistent config and policy across distributed, multi-region clusters.

Simple to complex — one gateway

Basic L7 routing to high-throughput streaming backends, all on the same declarative model.

- Apache Kafka
- Amazon Kinesis
- Zerobus
- gRPC streaming

- Enterprise scale
- Governed by default
- Multi-region ready

DEVELOPER PORTAL

Self-service APIs with Backstage

Backstage
OPEN-SOURCE IDP

Envoy Gateway integrates with **Backstage**, the popular open-source Internal Developer Platform — turning API delivery into a governed, self-service experience.

For Developers

Spin up an API directly from the Backstage portal using **reusable template forms** — no tickets, no hand-written gateway config.

- Pick a golden-path template
- Fill a simple form
- API provisioned in minutes

For Platform Engineers

Visualize and **govern every API** from one catalog — ownership, policy, and standards enforced by default.

- Curate reusable templates
- Visualize the API catalog
- Enforce policy & ownership

SELF-SERVICE FLOW

Browse template › **Fill the form** › **Gateway API provisions** › **Catalog & govern**

IMESH delivers the full integration — Backstage plugins, golden-path templates, and the gateway wiring — with enterprise support and services behind all of it.

LIFECYCLE & CONTROL

API lifecycle & governance

Manage every API from design to retirement under one governed model — versioned, secured, observed, and owned, without slowing teams down.

THE API LIFECYCLE

Design
Spec-first, OpenAPI

Publish
Self-service deploy

Secure
Auth, rate limits

Version
Safe rollout

Monitor
SLOs & usage

Retire
Graceful deprecate

GOVERNANCE, BUILT IN

Central Policy
Consistent rules across every API and cluster.

Approval Workflows
Reviewable, GitOps-backed change control.

Catalog & Ownership
Every API discoverable with a clear owner.

Compliance & Audit
Traceable policy and full audit history.

Governance shouldn't slow teams down — encode it once, enforce it everywhere, and let developers ship.

GOVERNANCE PRINCIPLE

DEFENSE IN DEPTH

Security best practices

The gateway is the front door to every application. Treat it as a primary enforcement point and apply security consistently across all traffic.

Encrypt Everything

Use TLS and mTLS wherever possible — in transit and between services for zero-trust east-west traffic.

Strong Authentication

Integrate with OIDC and OAuth2 identity providers; enforce JWT validation at the edge.

Traffic Protection

Apply rate limiting, request validation and WAF rules at the edge to blunt abuse and attacks.

Governance

Centralize access controls and enforce consistent policy across every cluster and team.

LAYERED ENFORCEMENT AT THE EDGE

TLS / mTLS termination	Layer 1 — encryption
Authentication & authorization	Layer 2 — identity
Rate limiting · WAF · validation	Layer 3 — protection
Centralized policy & audit	Layer 4 — governance

> **The gateway should serve as a primary security enforcement point for all application traffic.**
>
> SECURITY PRINCIPLE

RUN IT WELL

Operational best practices

Operational excellence comes from automation, observability, and disciplined lifecycle management — not heroics.

01 **High Availability**
Deploy multiple gateway instances across failure domains and availability zones.

02 **GitOps Automation**
Manage gateway configuration through Git-based, reviewable workflows.

03 **Observability**
Monitor latency, traffic volume, and error rates with SLO-backed alerting.

04 **Capacity Planning**
Validate performance under realistic peak load conditions before go-live.

05 **Upgrade Management**
Regularly align with current Envoy and Gateway API releases.

COMPARISON

Envoy Gateway vs alternatives

How Envoy Gateway compares against common ingress and API gateway options on the criteria that matter most to platform teams.

CRITERIA	ENVOY GATEWAY	NGINX	KONG	CLOUD PROVIDER
Open Source	✓	Partial	Partial	—
Kubernetes Native	✓	Partial	Partial	—
Gateway API Support	Full	Partial	Partial	Partial
Scale (millions RPS)	High	High	High	High
Speed · P99 latency	0.5 ms	Medium	Medium	Medium
Developer Portal	✓	—	Partial	Partial
API Lifecycle & Governance	✓	—	Partial	Partial
Vendor Lock-In	Low	Medium	Medium	High
Cloud Portability	✓	✓	✓	—
Operational Simplicity	High	Medium	Medium	High

Organizations increasingly favor Envoy Gateway for its alignment with Kubernetes standards and a long-term, open cloud-native strategy.

BEFORE YOU GO LIVE

Production readiness checklist

A pragmatic checklist to validate that your gateway deployment is resilient, secure, and operable before it carries production traffic.

Architecture

- ✓ High availability
- ✓ Capacity planning
- ✓ Backup strategy

Security

- ✓ TLS / mTLS enabled
- ✓ Authentication configured
- ✓ Access policies reviewed

Operations

- ✓ Monitoring enabled
- ✓ Logging enabled
- ✓ Alerting configured

Platform

- ✓ GitOps established
- ✓ Developer portal live
- ✓ Disaster recovery tested

Treat this as a release gate — every box should be verifiable in code and observable in your monitoring stack before promotion.

PHASED APPROACH

Adoption roadmap

A staged path that de-risks adoption — proving value early, then standardizing across the platform.

1

PHASE 1

Assessment & Architecture

Evaluate current state and design the target topology.

2

PHASE 2

Pilot Deployment

Stand up a controlled pilot to validate the approach.

3

PHASE 3

Non-Production Rollout

Expand across dev and staging environments.

4

PHASE 4

Production Deployment

Promote to production with HA and full observability.

5

PHASE 5

Platform Standardization

Make it the default ingress for every team.

SUCCESS FACTORS

Executive Sponsorship

Clear mandate and platform ownership.

Incremental Migration

Move workloads in safe, reversible steps.

Enablement & Training

Upskill teams as adoption scales.

PARTNERSHIP

How IMESH helps

IMESH partners with platform teams end-to-end — from architecture and deployment through developer enablement and 24×7 enterprise support.

Advisory Services

Architecture assessments · design reviews · migration planning.

Professional Services

Deployment · configuration · platform integration.

Training

Workshops · platform enablement · operational readiness.

Enterprise Support

24×7 support · incident assistance · upgrade guidance.

Developer Portal

Backstage integration · golden-path templates · self-service.

Performance Tuning

Scale & latency optimization for peak production load.

SPECIALIZED EXPERTISE

Deep, hands-on across the cloud-native networking stack

From ingress to service mesh to eBPF.

- Envoy Gateway
- Istio Service Mesh
- Cilium Networking

Envoy GATEWAY

Kubernetes GATEWAY API

Backstage DEV PORTAL

CONCLUSION

Modernize API infrastructure with confidence

Envoy Gateway gives organizations a scalable, secure, and cloud-native foundation for managing application traffic — and a path that scales with the business.

BY COMBINING

- **Envoy Proxy Performance**
- **Gateway API Standardization**
- **Open-Source Flexibility**
- **Enterprise Scale & Speed**
- **Developer Self-Service**
- **Built-in Governance**

…organizations establish a future-ready application networking platform — one that standardizes ingress, strengthens security, and accelerates delivery across every team.

Ready to evaluate Envoy Gateway?

Start small, standardize early, and build a gateway that scales with your business.

Talk to the IMESH team →