



**KubeCon**



**CloudNativeCon**



**OPEN SOURCE SUMMIT**

**China 2023**



KubeCon



CloudNativeCon



OPEN SOURCE SUMMIT

China 2023

# Envoy Gateway: The API Gateway in the Cloud Native Era

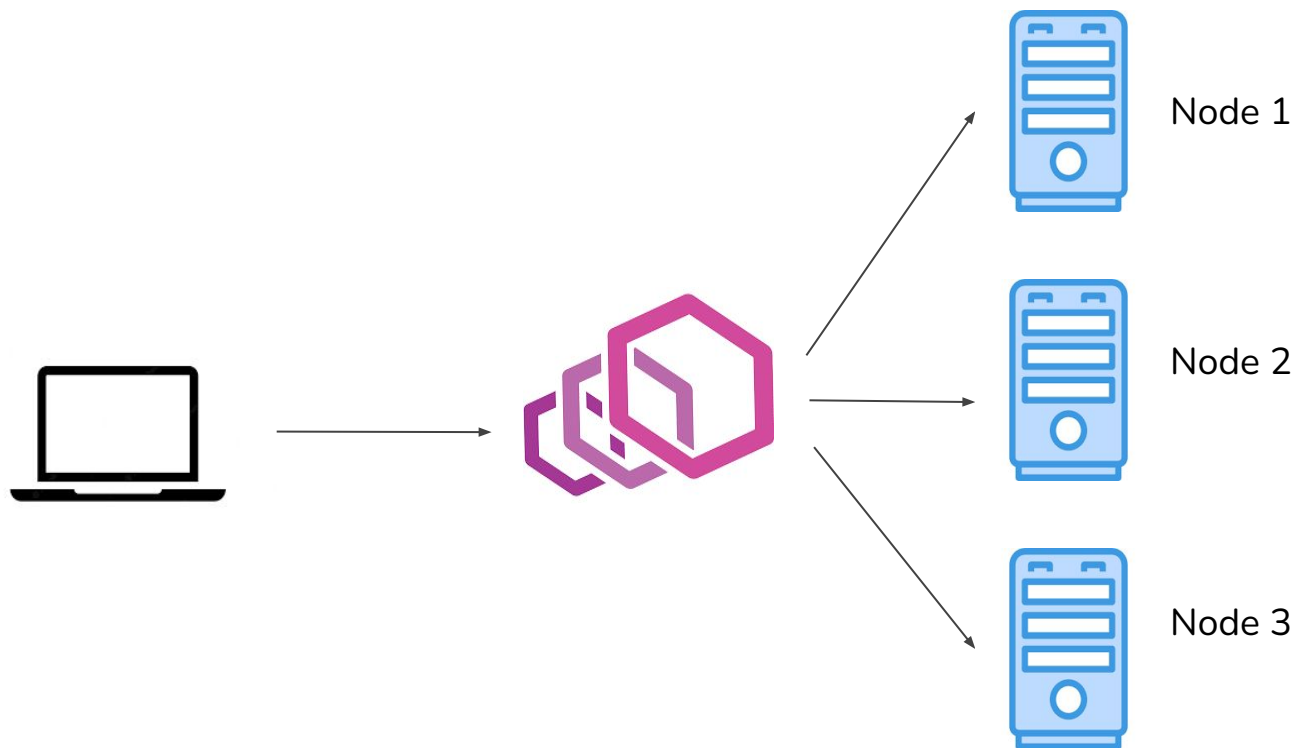
*Huabing Zhao Tetrade.io*  
*Xunzhuo Liu Tencent*

# Envoy as a Layer-7 Proxy

Envoy 是一个为 **云原生应用** 设计的开源 **边缘** 和 **服务代理** - 摘自 Envoy 官方网站。

Envoy 的设计理念: (**七层**) 网络通信应对业务逻辑透明

- 负载均衡
- 超时、重试
- 断路器
- 七层路由
- 调用追踪
- 访问日志
- 请求指标
- 安全通信
- 访问控制





# Created for Cloud-Native

## Envoy 与“传统”网络代理的本质区别:为云原生架构而生!

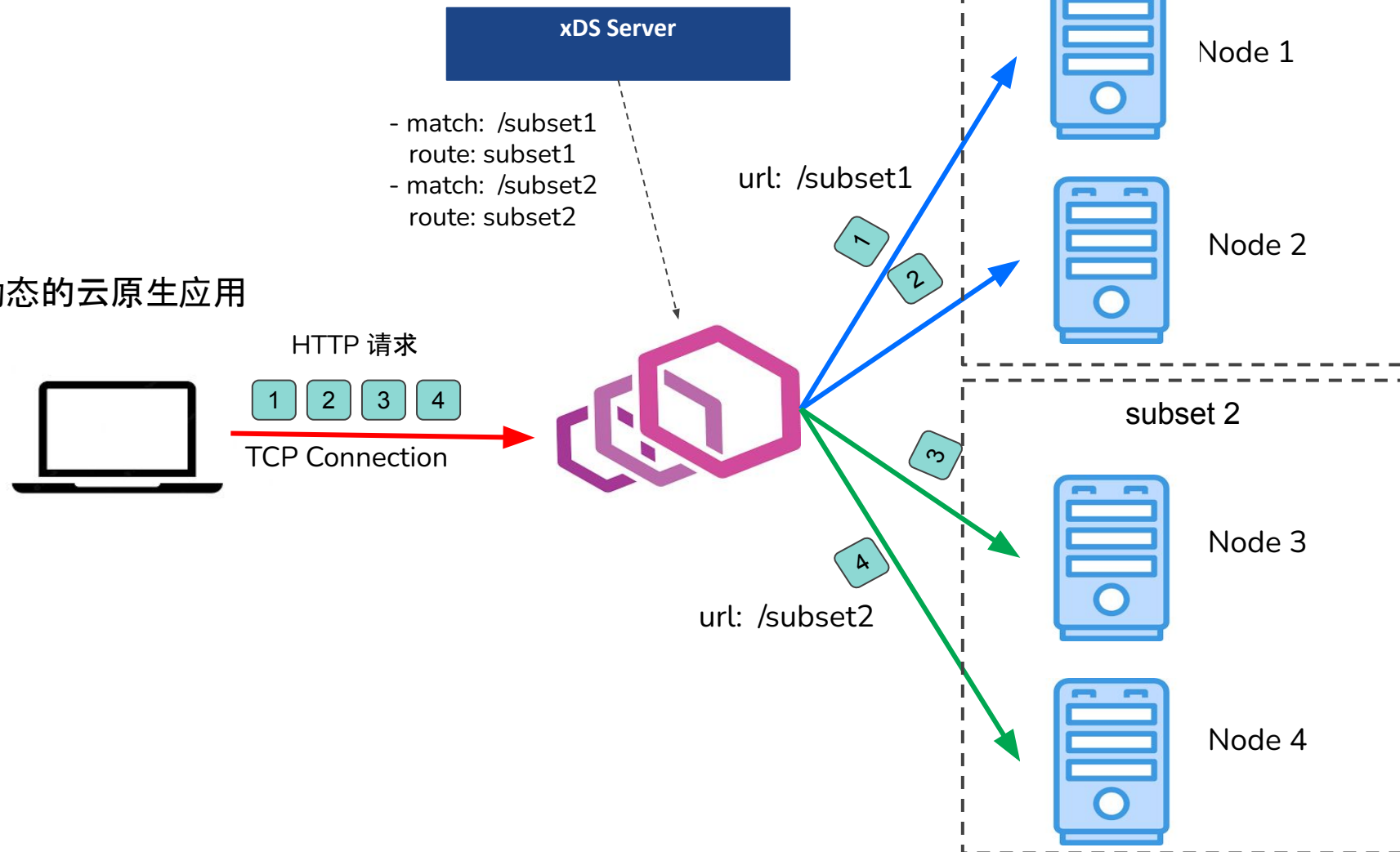
云原生架构的特点:

- 系统由大量微服务构成
- 是“牲口”而不是“宠物”
- 微服务可以动态伸缩

“传统”网络代理的静态配置难以适应高度动态的云原生应用

Envoy 支持动态获取配置

- 服务实例
- 路由配置
- 安全策略
- .....



# Created for Cloud-Native

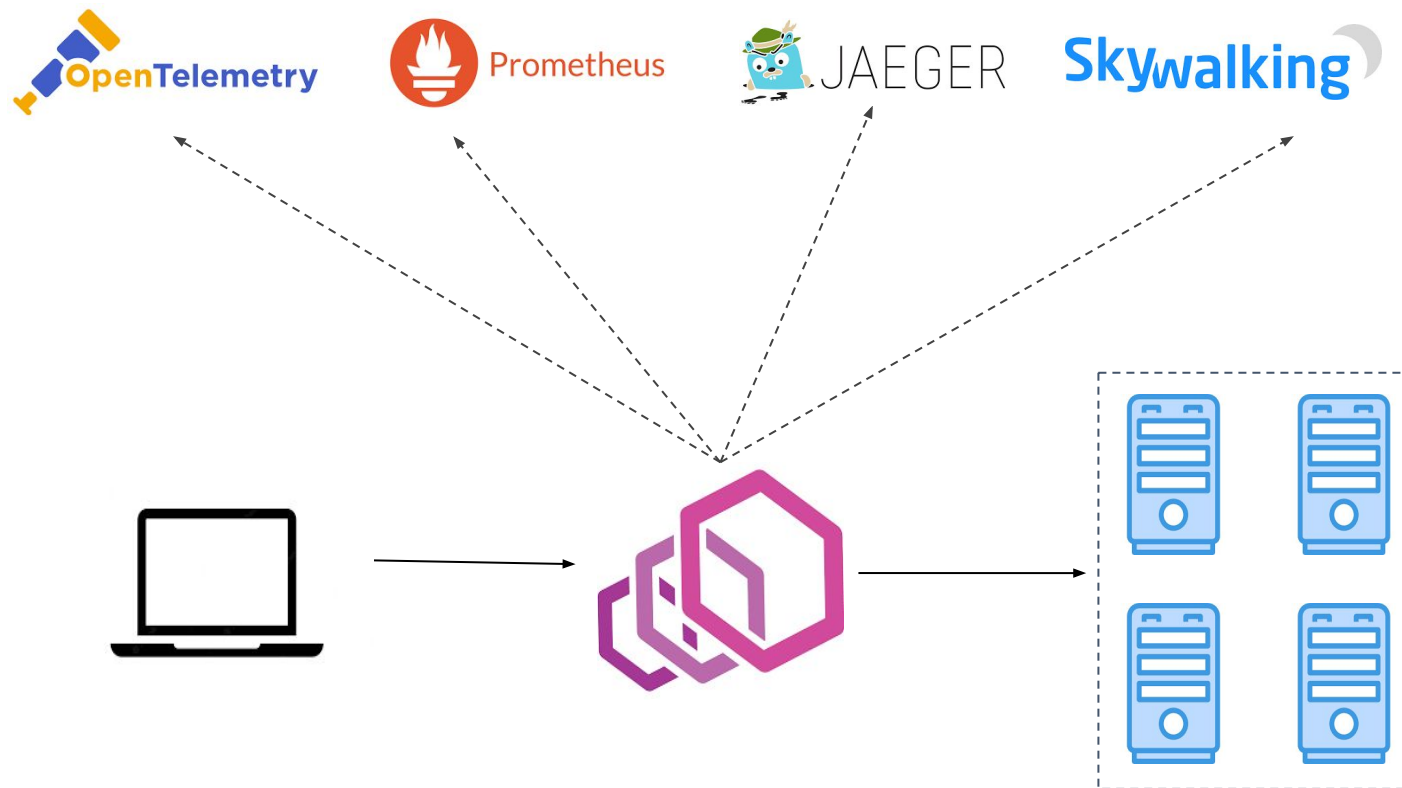
## 内建 Metrics、Logging、Tracing 支持, 快速定位系统故障

云原生应用可观测性挑战:

- 系统包含大量微服务
- 业务调用跨多个 RPC
- 难以查看系统全貌
- 故障定位极其困难

强大的可观测能力:

- OpenTelemetry
- Prometheus
- Jaeger
- GRPC Logging server



# Gateway API as Management API

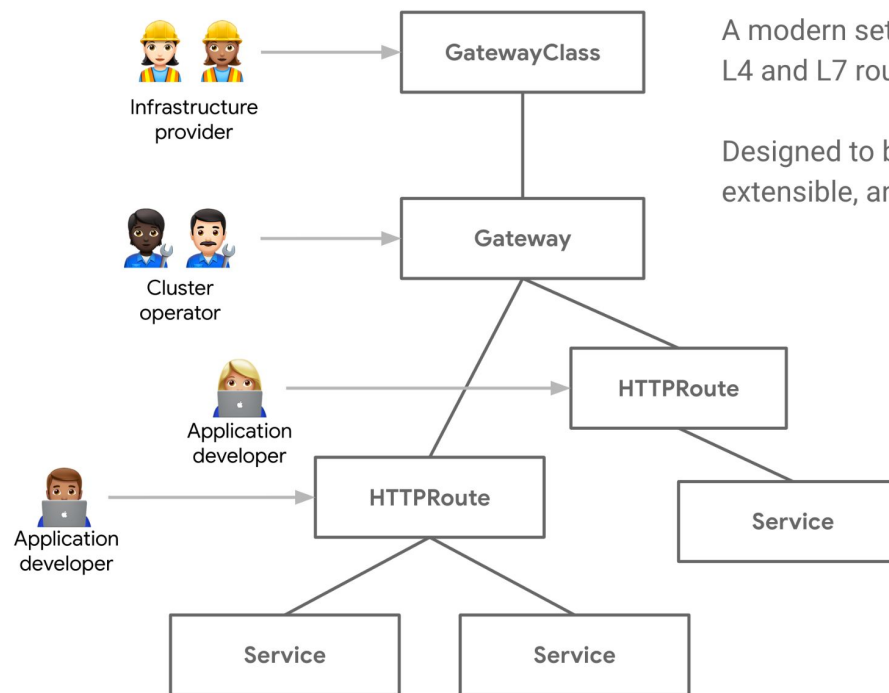
## Ingress

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: test
spec:
  rules:
    - host: foo.bar.com
      http:
        paths:
          - path: /foo
            backend:
              serviceName: s1
              servicePort: 80
          - path: /bar
            backend:
              serviceName: s2
              servicePort: 80
```

- HTTP host matching
- HTTP path matching
- TLS



## Gateway API



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

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

### 更强大的流量管理能力

- HTTP header-based matching
- HTTP header manipulation
- Weighted traffic splitting
- gRPC,UDP,TCP routing
- Role-oriented resource model

### 更灵活的扩展机制

- Arbitrary backend
- Custom filters
- Policy Attachment

# From Gateway to Service Mesh

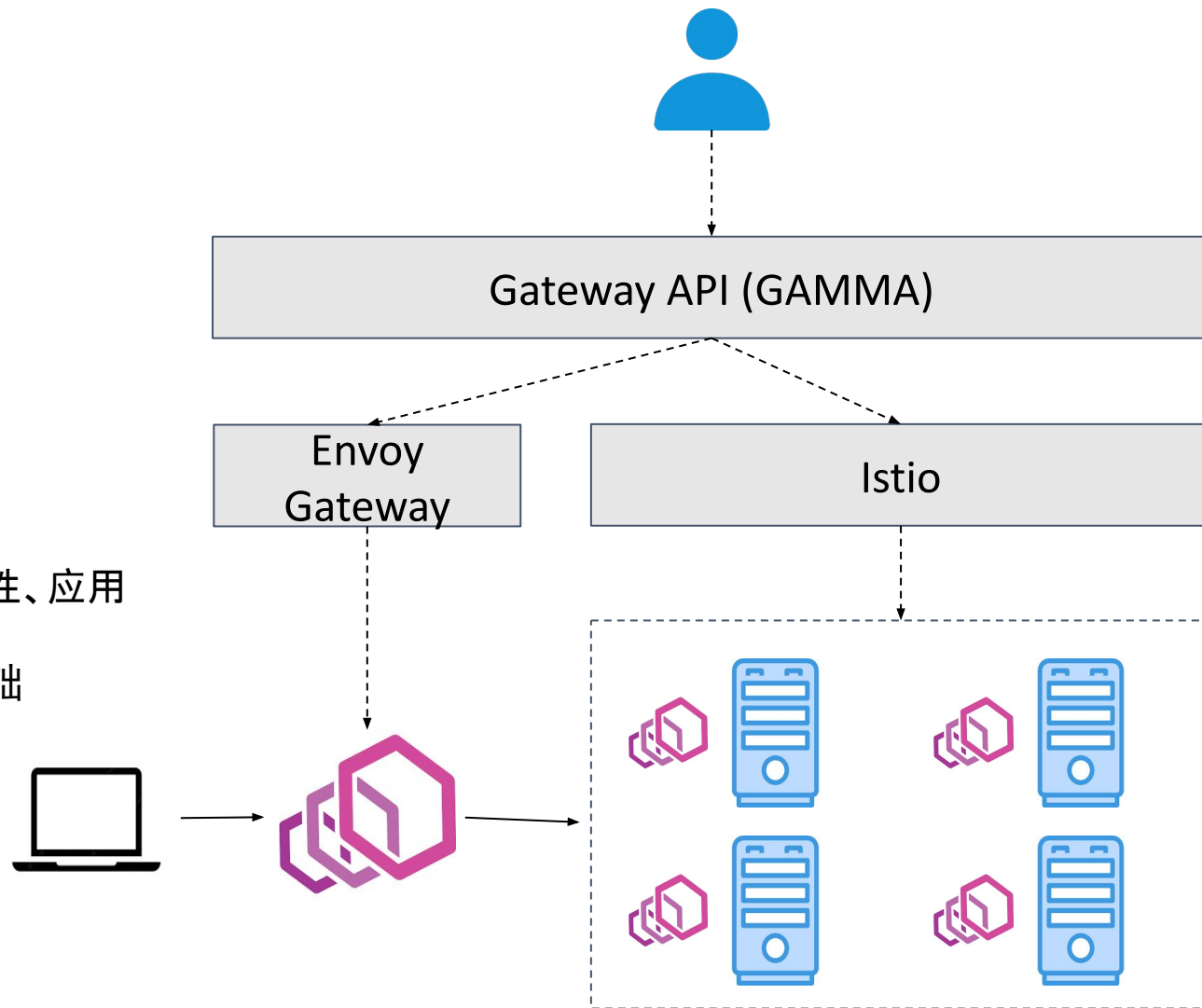
## 采用 Envoy Gateway 平滑过渡到 Service Mesh

### 犹豫是否采用服务网格？

- 额外资源占用？
- 应用迁移风险？
- 缺少运维经验？
- 运维团队驱动？

### 从边缘网关开始！

- 采用 Envoy Gateway 作为边缘网关
- 在边缘网关上获得 Envoy 的所有能力(流量管理、可观测性、应用安全...)
- 熟悉 Envoy 的管理配置, 为过渡到 Service Mesh 打好基础
- 标准 Gateway API/GAMMA 提供统一的管理接口
- 平滑迁移, 最大化减小迁移风险



# Background of Envoy Gateway

## 问题/现状:

- EnvoyProxy 使用门槛高(xDS、控制面、配置、部署)
- 围绕安全、控制平面和其他共同关注点的存在重复工作

2022 年 5 月 EnvoyProxy 创始人 Matt Klein

联合 Ambassador Labs, Fidelity, Tetrade, VMware  
发起 Envoy Gateway 项目

## 这样带来的好处?

- 供应商专注于以扩展、管理平面 UI 等形式上分层增值功能
- 形成 "水涨船高" 的现象, 让全球更多用户, 不管团队规模大小, 技术积累的差异, 都能享受 Envoy 带来的好处
- 更多的用户会带来更多潜在客户、更多对 Envoy 核心项目的支持以及更好的整体体验等良性循环

合并 CNCF 的 API 网关项目: Contour 和 Emissary, 使用 Envoy Gateway 作为其内核, 统一 Envoy-based Kubernetes Ingress

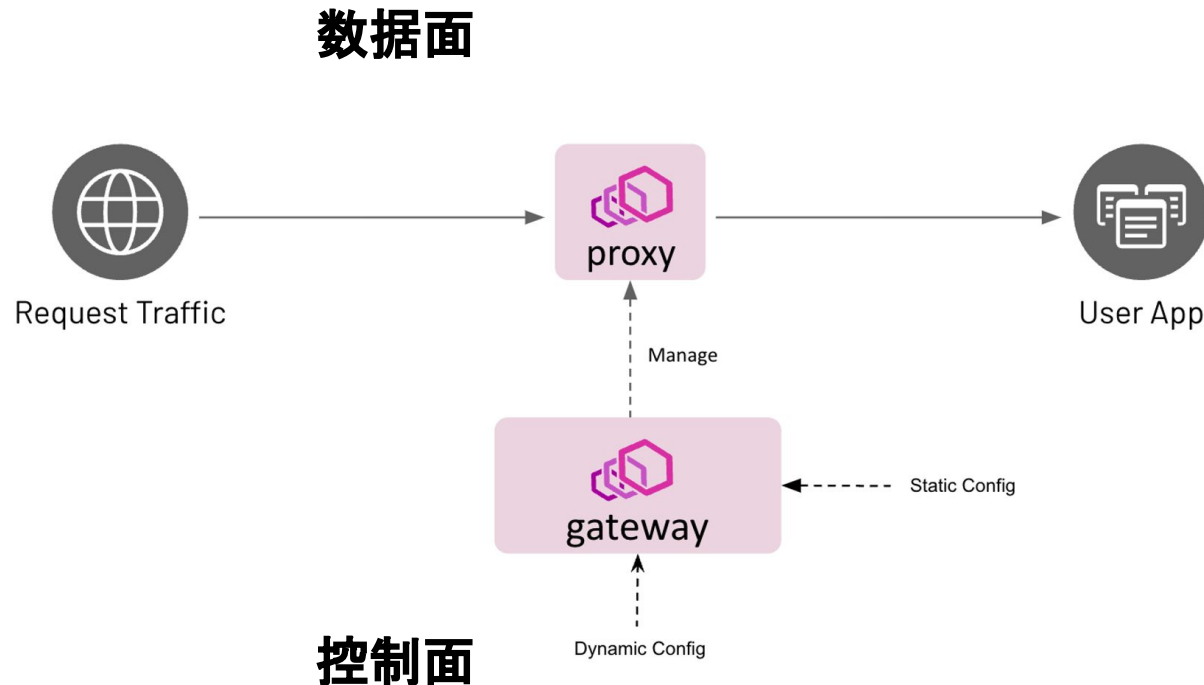




# Goals / Benefits of Envoy Gateway

## Envoy Gateway 的目标

- **Expressive API**: 基于 Gateway API + Envoy-Specific API, 提供简单且富有表现力的 API, **屏蔽 Envoy 底层细节**, 让 Envoy 对开发者**开箱即用**
- **Batteries included**: 简化 Envoy 部署和管理, EG 会自动管理 EnvoyProxy 的资源, **GWAPI 开发/运维/基础设施关注点分离**
- **All environments**: 支持 Kubernetes 以及非 Kubernetes 环境, 额外目标包括支持**多集群和各种运行时环境**。
- **Extensibility**: 为供应商和终端用户, 提供灵活的扩展机制, 如 gRPC Extension Hook、EnvoyPatchPolicy 等, 解决通用 API **无法覆盖所有场景**的问题



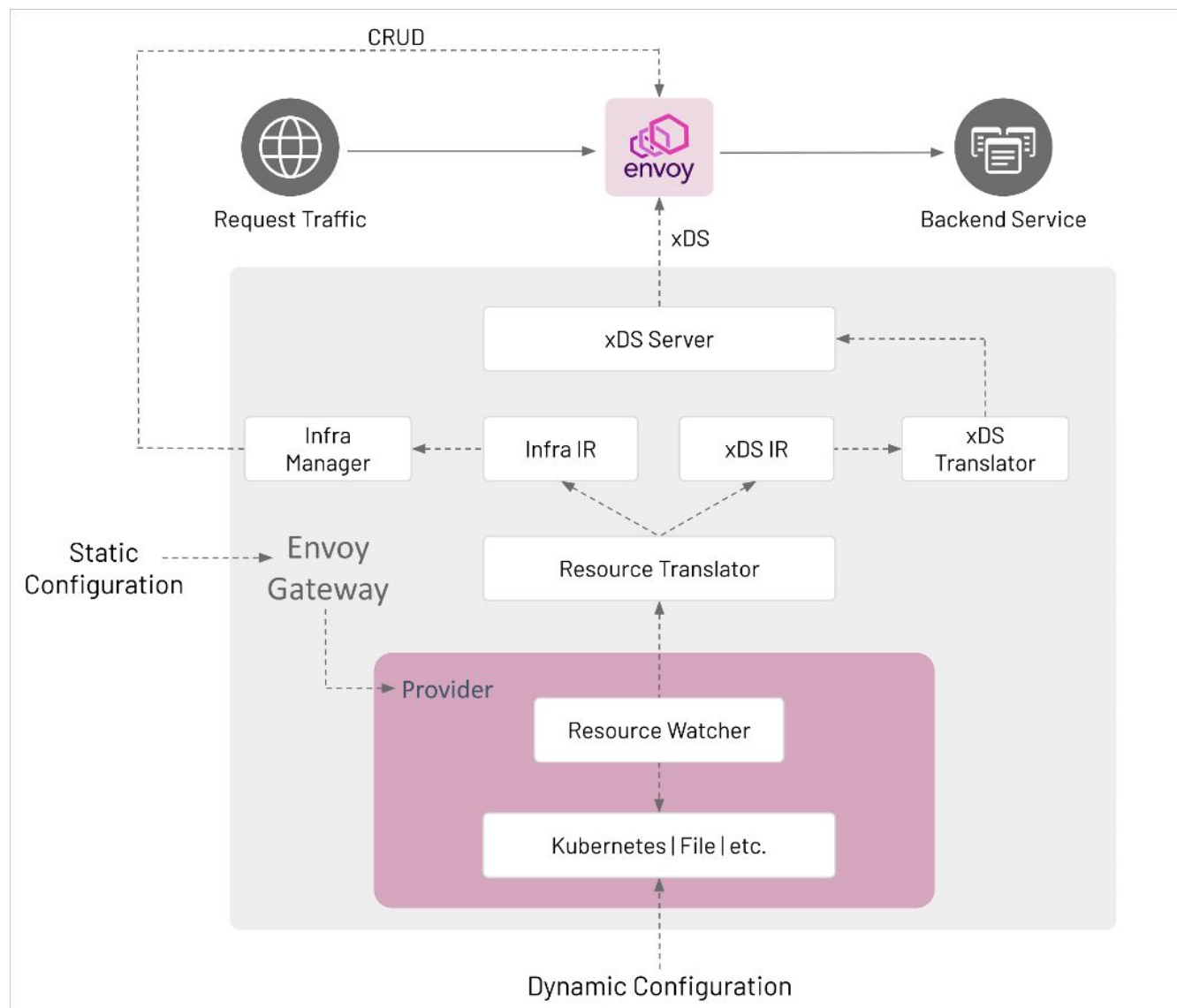
# Architecture of Envoy Gateway

## Configuration:

- **Static:** Bootstrap Configuration
- **Dynamic**
  - **Infra Management:** EnvoyProxy, GatewayClass, Gateway
  - **Traffic Routing:** xRoute (HTTP, gRPC, TCP, UDP)
  - **Extension:** EnvoyPatchPolicy

## Components:

- **Provider** (Service Discovery, Persist Data)
  - Kubernetes
  - File
- **Resource Watcher:** list/watch dynamic configuration
- **Resource Translator:** translate external configuration to infra/xds IR.
- **Intermediate Representation (IR):**
  - **infra IR:** data plane infra
  - **xds IR:** data plane xds configuration
- **xDS Translator:** xds IR to xds Resources
- **xDS Server:** connected to managed data plane envoyproxies.
- **Infra Manager:** infra IR to manage infra resources.



# Configure Envoy Gateway

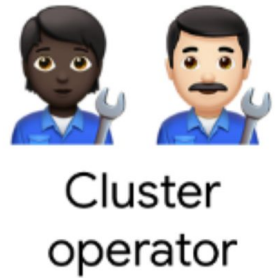


Infrastructure  
provider



```
kind: GatewayClass
metadata:
  name: eg
spec:
  controllerName:
    gateway.envoyproxy.io/gate..
```

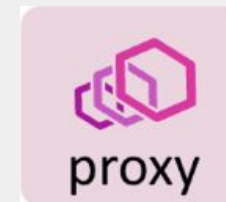
# Configure Envoy Gateway



```
kind: GatewayClass  
metadata:
```

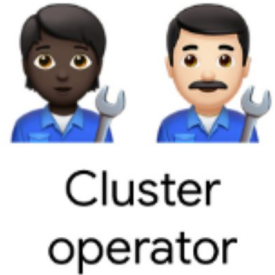
```
  name: eg  
spec: con  
  gateway
```

```
  kind: Gateway  
  metadata:  
    name: eg  
  spec:  
    gatewayClassName: eg  
  listeners:  
  - name: http  
    protocol: HTTP  
    Port: 8080
```





# Configure Envoy Gateway



```
kind: GatewayClass
metadata:
  name:
spec:
  con
gatew
```

```
kind: Gateway
metadata:
  name: eg
spec:
  gateway
  listene
  - name
  pro
  Por
```

```
kind: HTTPRoute
metadata:
  name: backend
spec:
  parentRefs:
    - name: eg
  hostnames:
    - www.example.com
  rules:
    - backendRefs:
      - name: backend
        Port: 3000
```

# Roadmap of Envoy Gateway

- v0.1.0: 设计 Envoy Gateway 2022年 5 月
- v0.2.0: 建立坚实的基础 2022年 10 月
  - 实现核心的 EG 设计
  - 建立测试、e2e、集成等自动化
  - 建立用户与开发者文档
  - 实现自动化 Gateway API Conformance 测试
  - 初步建立完善的 CI/CD 流水线
- v0.3.0: 实现 GWAPI 高级功能 2023年 2 月
  - 实现 URLRewrite / Request Mirror / ResponseHeader Modifier 等 Extended Filters
  - 实现 TCP/UDP/gRPC 等 xRoute
  - 实现 ReferenceGrant 权限控制模型
  - 实现 Global Rate Limiting 与 Request Authentication 等扩展能力

# Roadmap of Envoy Gateway

- v0.4.0: 自定义扩展 2023年 4 月
  - 实现基于 gRPC 的控制面扩展机制
  - 提供 Helm / egctl 等安装/命令行工具, 简化 EG 的使用/安装/运维
  - 实现自定义 EnvoyProxy 的 Kubernetes 资源
  - 实现自定义 EnvoyProxy Bootstrap 配置
- v0.5.0: 可观测性 2023年 8 月
  - 实现 Envoy Gateway 数据面的可观测性
  - 实现 EnvoyPatchPolicy 自定义 xDS 资源
- v0.6.0: 为 GA 做准备 2023年 11 月
  - 实现 Envoy Gateway 控制面的可观测性
  - Envoy Gateway 性能测试
  - 实现 TrafficPolicy(Client/Server)、Security 等 Policy API 提供更多高级功能
- .....

# Thanks To All The Contributors!

- Over 64+ Contributors
- Over 720 + Issues
- Over 1170 + PRs





# Get Involved!

Docs: [gateway.envoyproxy.io](https://gateway.envoyproxy.io)

Project: [github.com/envoyproxy/gateway](https://github.com/envoyproxy/gateway)

微信: Liu-Xunzhuo / zhao\_huabing