



Streamlining API Governance: Kubernetes Native API Management



Pubudu Gunatilaka

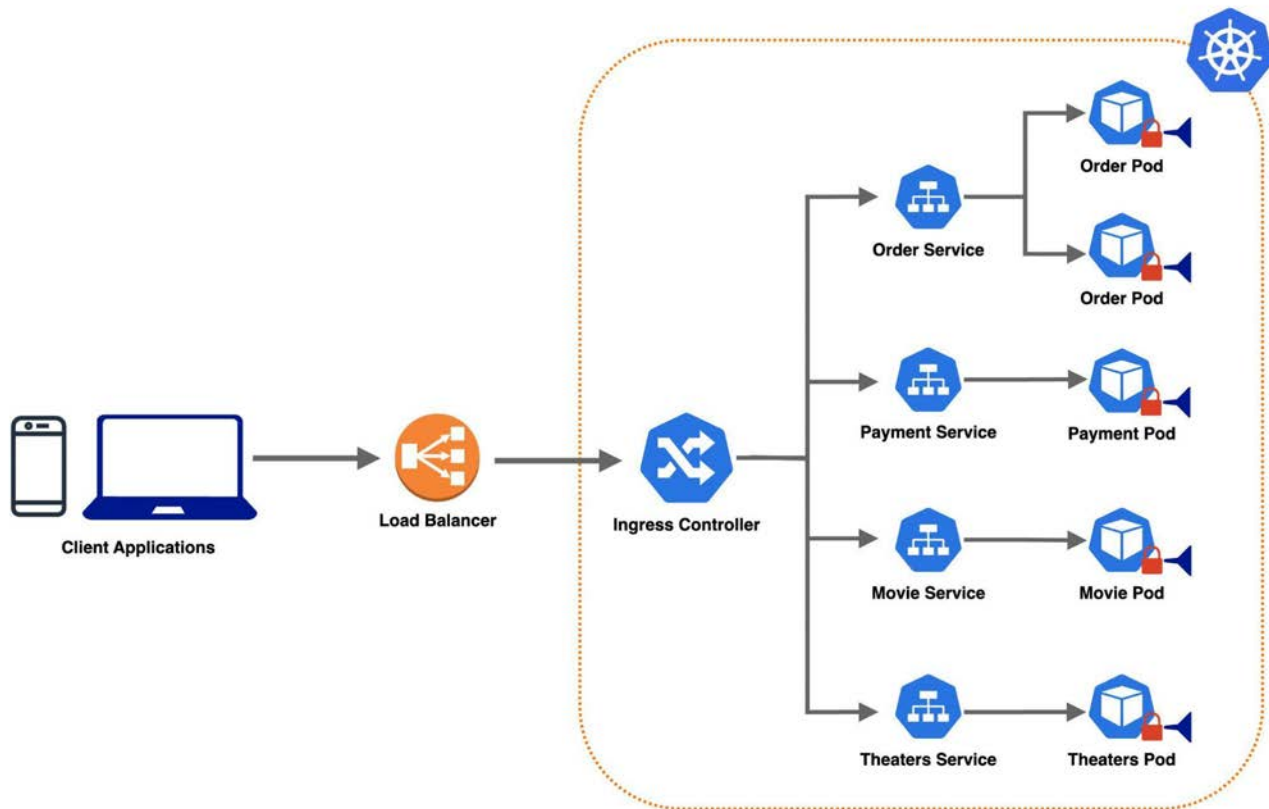
Senior Technical Lead



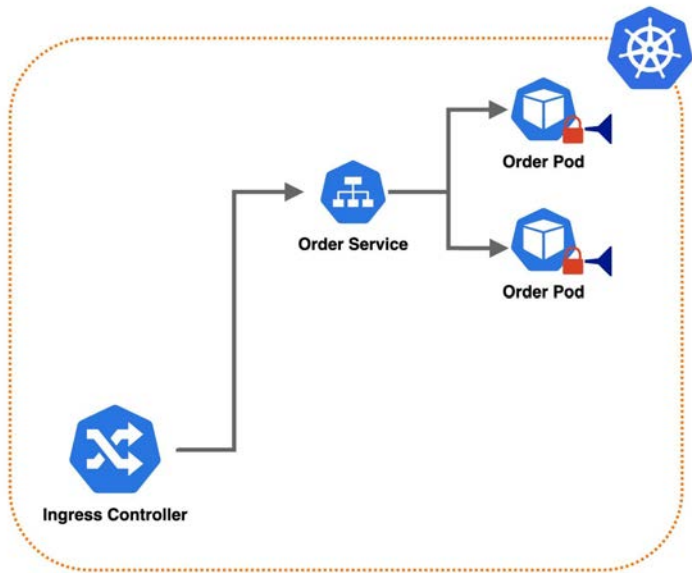
pubudug@wso2.com

September, 2024

Kubernetes Application Architecture



The Ingress Resource in Kubernetes



```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: order-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /orderList
spec:
  ingressClassName: nginx
  rules:
  - host: "tickets.mn.com"
    http:
      paths:
      - path: /orders
        pathType: Prefix
        backend:
          service:
            name: order-service
            port:
              number: 9090
```

Challenges with Kubernetes Ingress

- Limited to HTTP/HTTPS: Lacks native support for other protocols (e.g. TCP, UDP).
- Basic routing: Lacks advanced capabilities like traffic splitting and header-based routing.
- Not easily extensible: Relies on custom controllers for additional features.
- Basic security: Requires extra configurations for advanced features like mutual TLS.
- Limited observability: Lacks built-in metrics and detailed traffic monitoring.
- No granular traffic control for retries, timeouts, or rate limiting.

Kubernetes Gateway API



Introduction to **Kubernetes Gateway API**

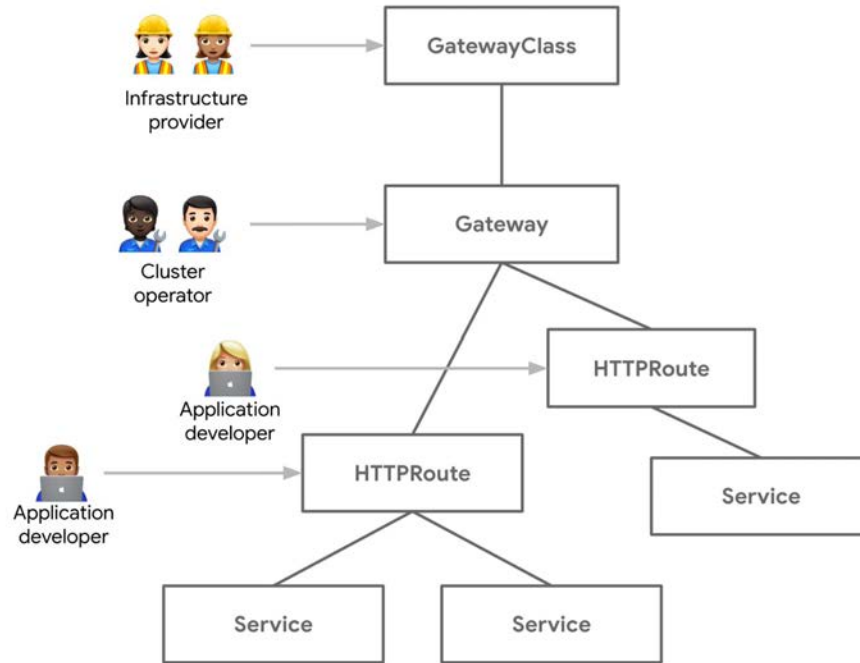


Image Source: <https://gateway-api.sigs.k8s.io/>

Kubernetes Gateway API Standardization

- **API Standardization:** Aims to standardize and streamline API gateway management in Kubernetes.
- **Simplified Configuration:** Common interface for easier setup and maintenance.
- **Enhanced Interoperability:** Smooth integration with different gateway implementations.
- **Dynamic Automation:** Dynamic configuration, scaling, and service discovery.
- **Seamless Kubernetes Integration:** Effortlessly works within the Kubernetes ecosystem.
- **Community Collaboration:** Driven by a diverse developer community for continuous improvement.

Kubernetes Gateway Custom Resources



OpenAPI
Specification

Order Service

```
apiVersion: gateway.networking.k8s.io/v1beta1
kind: HTTPRoute
metadata:
  name: order-route
spec:
  hostnames:
  - "tickets.mn.com"
  rules:
  - matches:
    - method: POST
      path:
        type: RegularExpression
        value: /orders
    filters:
    - type: URLRewrite
      urlRewrite:
        path:
          replaceFullPath: /orderList
          type: ReplaceFullPath
    - type: RequestHeaderModifier
      requestHeaderModifier:
        add:
        - name: x-order-id
          value: "PG-109"
    backendRefs:
    - name: order-service
      port: 9090
```

Kubernetes Custom
Resource - HTTPRoute



Moving from Ingress to K8s Gateway API

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: order-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /orderList
spec:
  ingressClassName: nginx
  Rules:
  - host: "tickets.mn.com"
    http:
      paths:
      - path: /orders
        pathType: Prefix
        backend:
          service:
            name: order-service
            port:
              number: 9090
```



```
apiVersion: gateway.networking.k8s.io/v1beta1
kind: HTTPRoute
metadata:
  name: order-route
spec:
  hostnames:
  - "tickets.mn.com"
  rules:
  - matches:
    - method: POST
      path:
        type: RegularExpression
        value: /orders
    filters:
    - type: URLRewrite
      urlRewrite:
        path:
          replaceFullPath: /orderList
          type: ReplaceFullPath
    - type: RequestHeaderModifier
      requestHeaderModifier:
        add:
        - name: x-order-id
          value: "PG-109"
    backendRefs:
    - name: order-service
      port: 9090
```



OpenAPI
Specification

Order Service

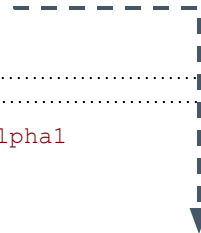
Extensibility of the Kubernetes Gateway API

```
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: order-route
spec:
  hostnames:
  - "tickets.mn.com"
  rules:
  - matches:
    - method: POST
      path:
        type: RegularExpression
        value: /orders
    backendRefs:
    - name: order-service
      port: 9090
```

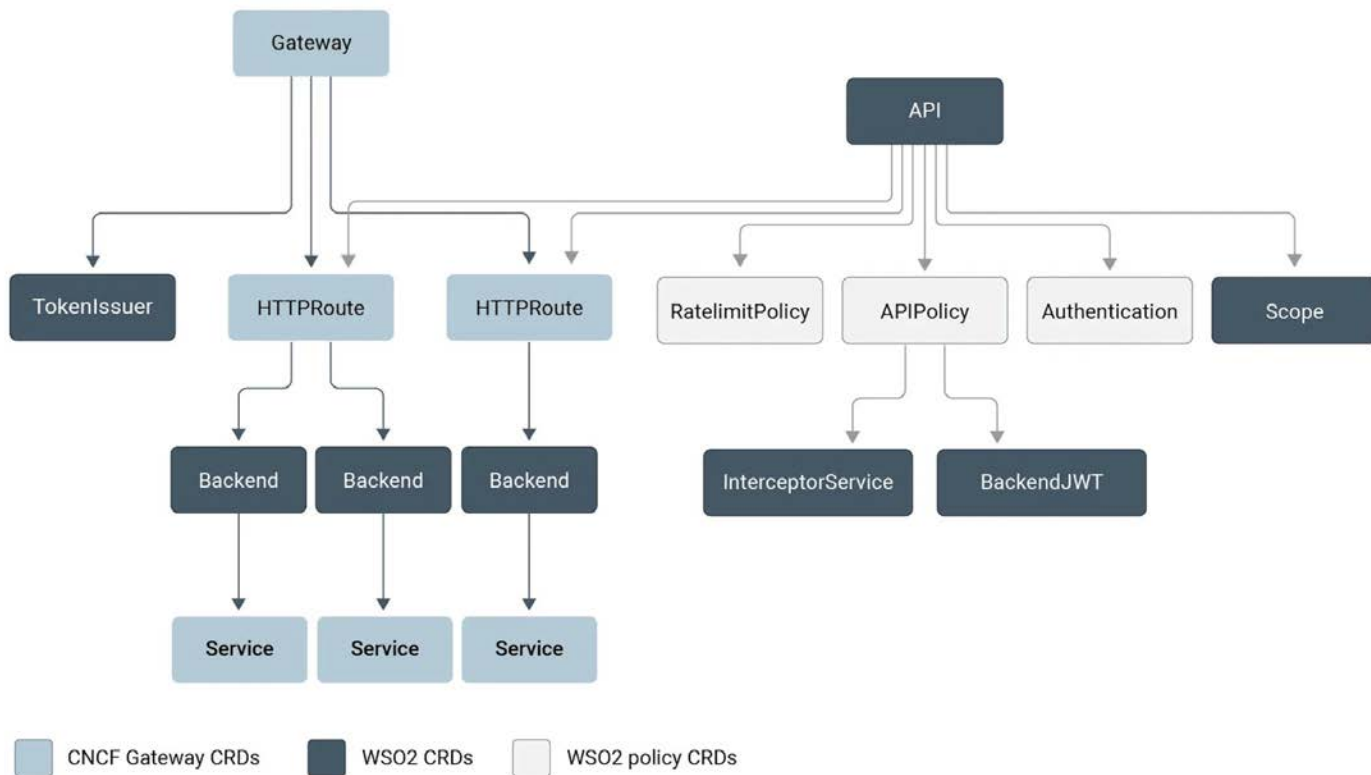


```
apiVersion: gateway.networking.k8s.io/v1beta1
kind: HTTPRoute
metadata:
  name: order-route
spec:
  hostnames:
  - "tickets.mn.com"
  rules:
  - matches:
    - method: POST
      path:
        type: RegularExpression
        value: /orders
    backendRefs:
    - group: dp.xyz.com
      kind: Backend
      name: order-backend
```

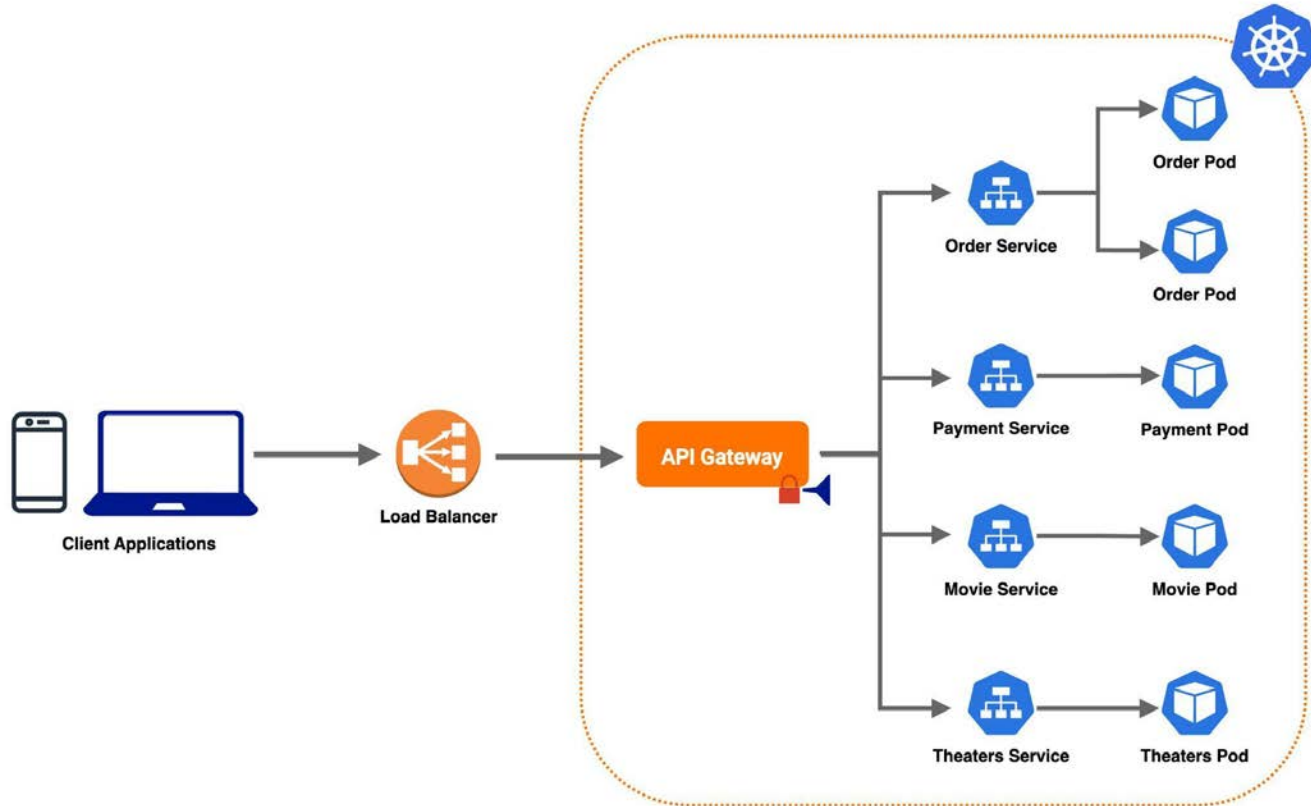
```
apiVersion: dp.xyz.com/v1alpha1
kind: Backend
metadata:
  name: order-backend
spec:
  services:
  - host: order-service
    port: 9090
  basePath: "/"
  protocol: "http"
  timeout: 20s
```



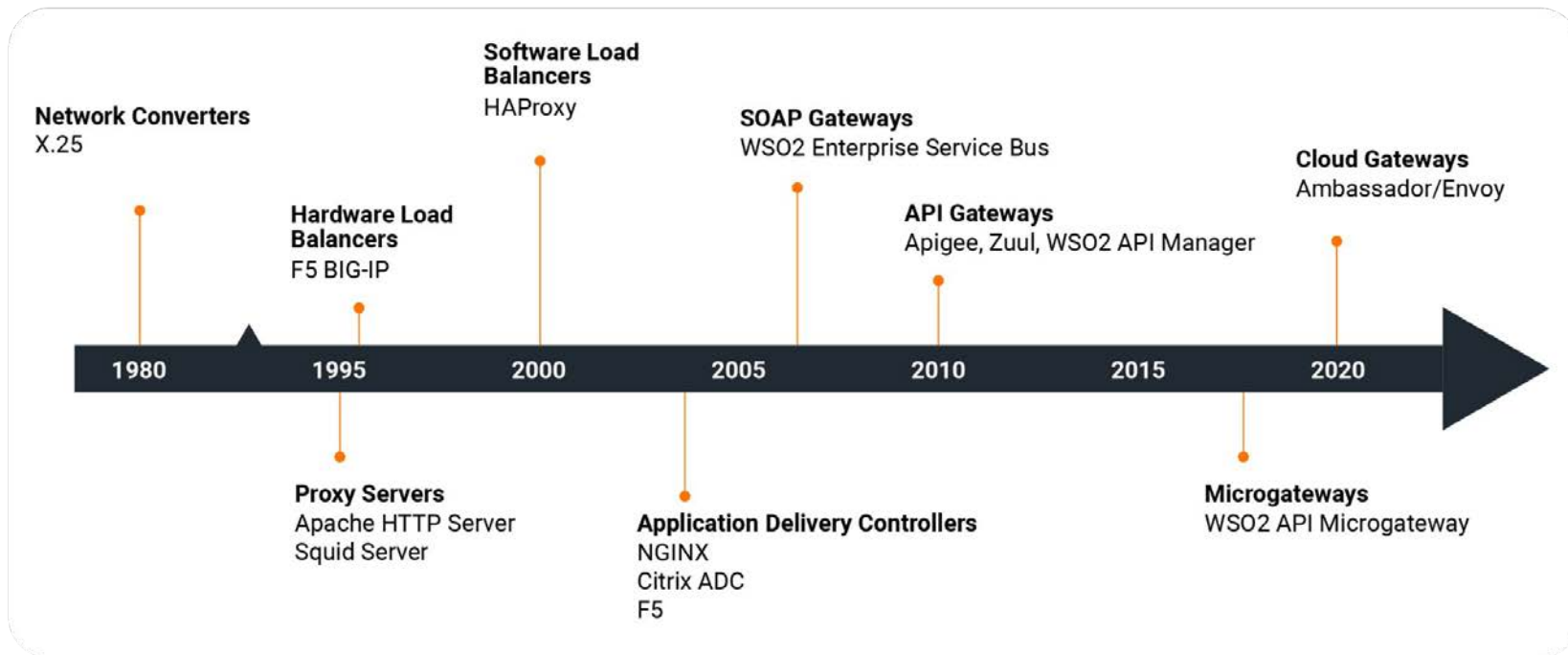
Extensibility of the Kubernetes Gateway API...



Kubernetes Gateway API Based Architecture for Kubernetes Applications



The Evolution of API Gateways

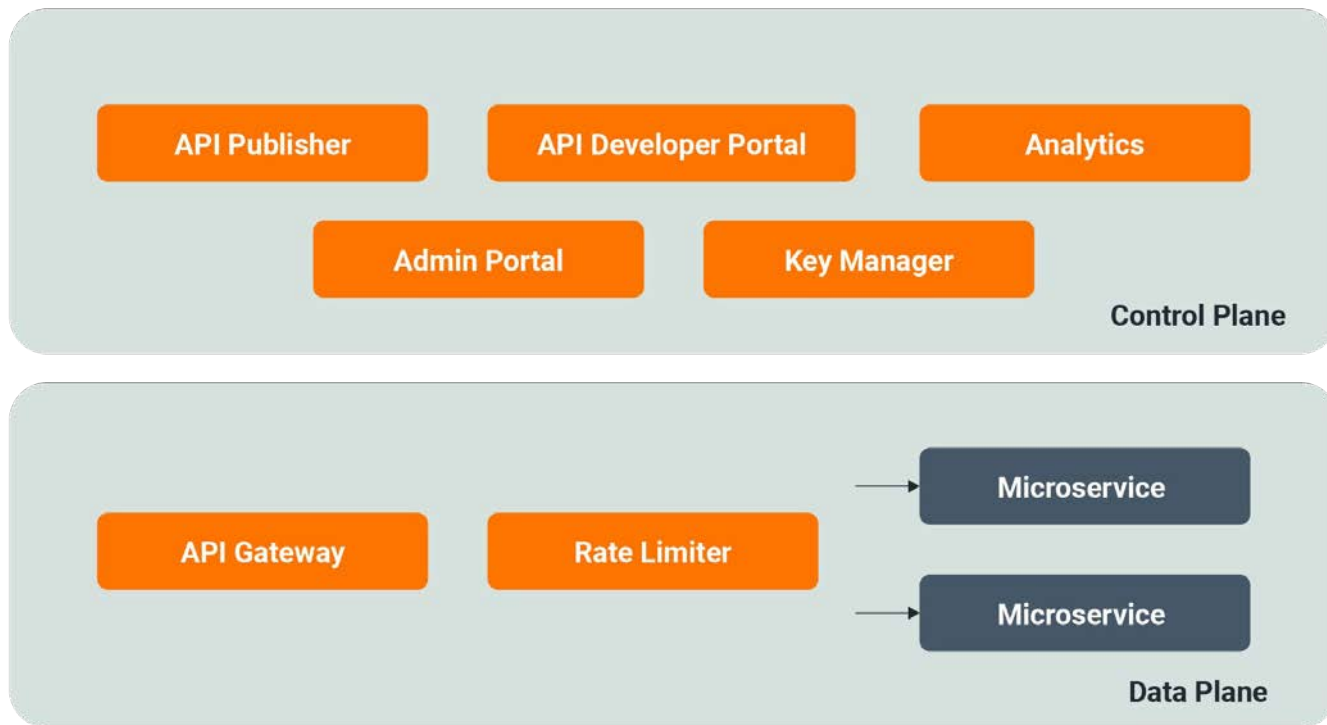


Empowering API Gateway Transformation with Envoy Proxy

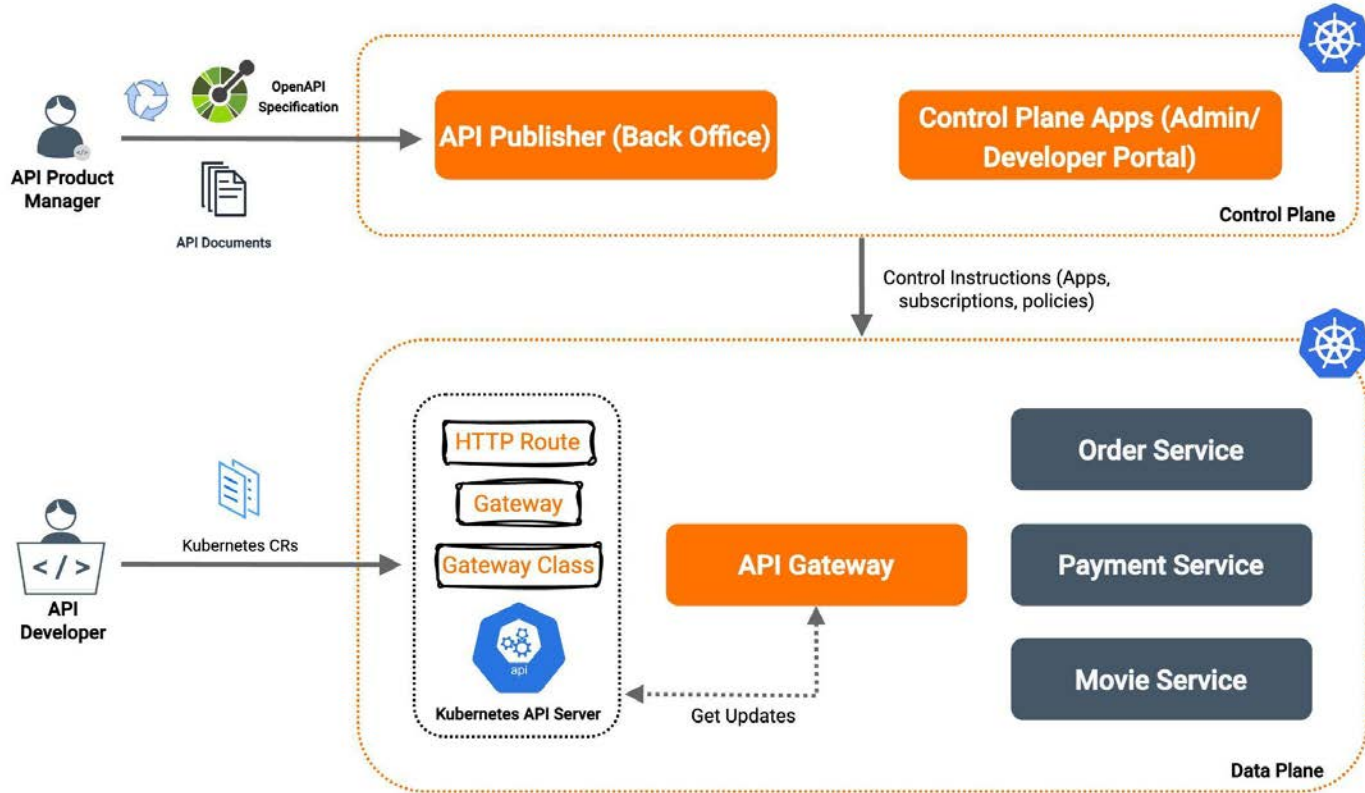
- Continuous Development and Open Source Benefits
- Envoy as an Ingress Controller in Kubernetes
- Essential API Management Features
- Support for REST and gRPC Services
- Extensibility options to enhance and customize its capabilities
- Flexible Deployment Patterns



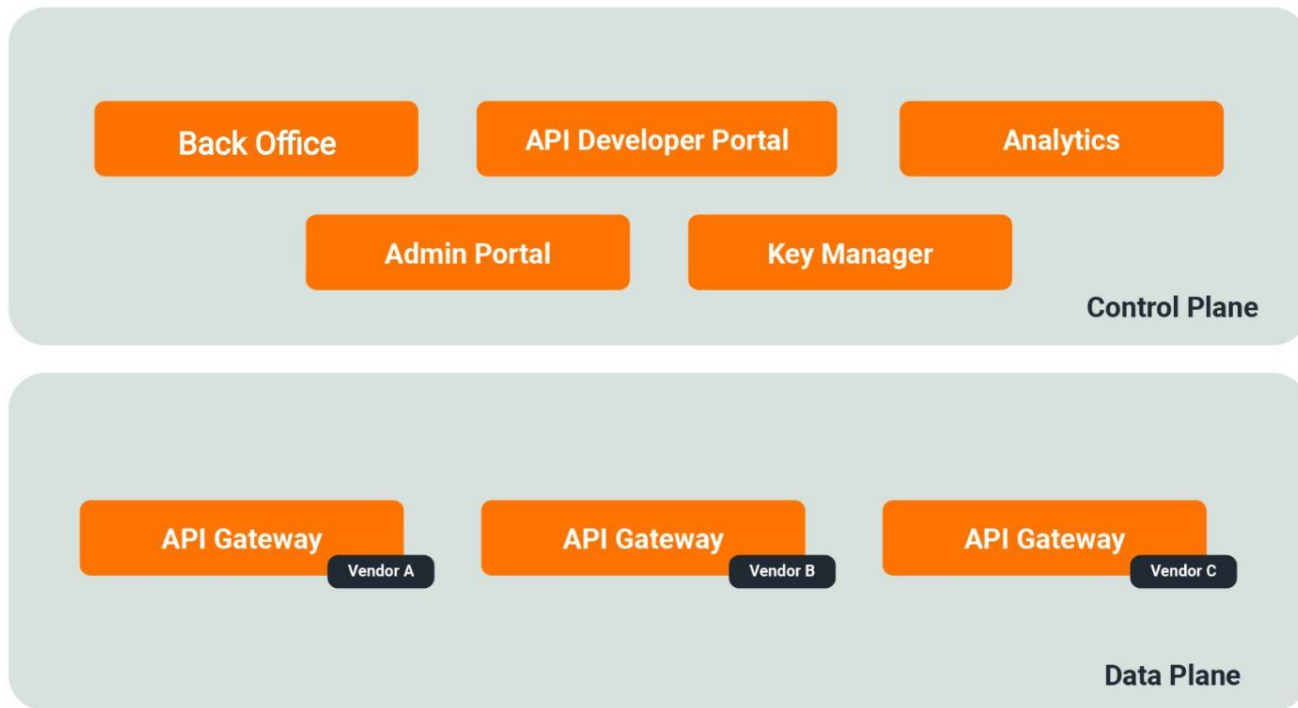
The **Cloud Native** API Management Architecture



API Governance in Kubernetes Native API Management



Unified API Control Plane for Diverse API Gateways





API Gateways evolving into a vital but commoditized part of infrastructure.

Navigating Change: Focusing on API Management

- **Shifting Developer Focus:**
 - ⦿ Developers are freed from the complexities of gateway management.
 - ⦿ Can focus on building and maintaining APIs and core development tasks.

- **Key API Management Features:**
 - ⦿ API Lifecycle Management
 - ⦿ API Governance
 - ⦿ API Marketplace
 - ⦿ API Version Management
 - ⦿ API Insights...

Thanks!



wso2.com

