



KubeCon



CloudNativeCon

Europe 2026

#KubeCon #CloudNativeCon

AI'm At The Gate!

Introducing the AI Gateway Working Group in Kubernetes

Nir Rozenbaum, Morgan Foster, Shachar Tal



Agenda

- Speakers Intro
- What is WG AI Gateway?
- Payload Processing
- Egress Gateway (including live demo!)
- Future Plans
- Get Involved



KubeCon



CloudNativeCon

Europe 2026

Speakers Intro

Speakers Intro



Nir Rozenbaum

Sr Principal Software Engineer
Red Hat



Morgan Foster

Sr Principal Software Engineer
Red Hat



Shachar Tal

Distinguished Engineer
Palo Alto Networks



KubeCon



CloudNativeCon

Europe 2026

What is WG AI Gateway?

What is an AI Gateway?

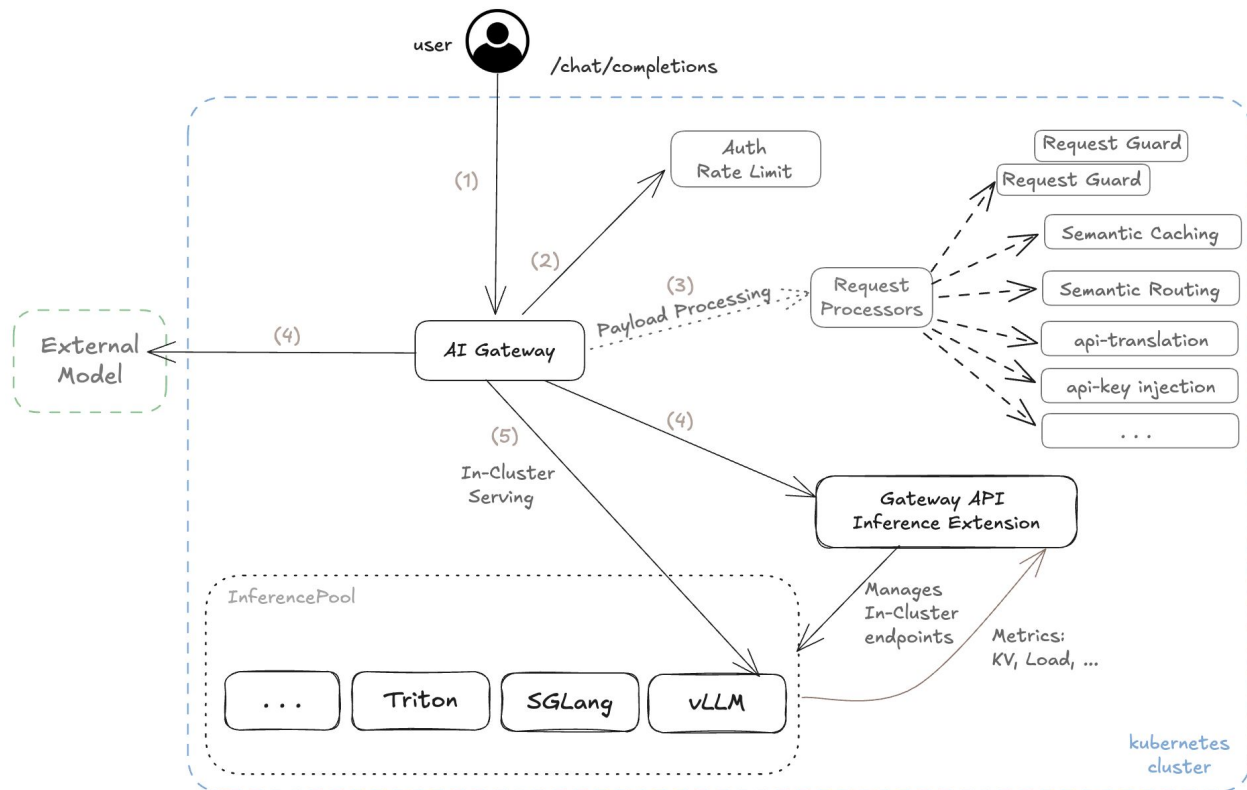
- An **AI Gateway** is a K8s Gateway ([Gateway API](#) compliant) that applies policy and routing decisions on AI inference traffic, based on the request/response payload, and might include:
 - Token-based rate limiting for AI APIs.
 - Fine-grained access controls for inference APIs.
 - Payload inspection enabling intelligent routing, caching, and guardrails.
 - Support for AI-specific protocols and routing patterns.

What is the WG's mission?

- **Standardize AI traffic management at the Gateway layer.**
 - **Standards Development**: Declarative APIs and standards for AI workload networking in K8s.
 - **Extensible Architecture**: Composable, pluggable, and ordered processing for AI-specific gateway extensions.
 - **Leveraging Existing Standards**: Build on Gateway API and established networking foundations.
 - **Community Collaboration**: Foster discussions and build consensus around best practices for AI infrastructure.

A Typical AI Gateway Flow

```
curl ${GATEWAY_IP}:${PORT}/chat/completions /
-H "Content-Type: application/json"
-H "Authorization: Bearer ${API_KEY}"
-d '{
  "model": "gpt-3.5-turbo",
  "messages": [
    {
      "role": "system",
      "content": "You are a helpful assistant."
    },
    {
      "role": "user",
      "content": "Hello, how are you?"
    }
  ]
}'
```



KubeCon



CloudNativeCon

Europe 2026



KubeCon



CloudNativeCon

Europe 2026

Payload Processing

Requirements

- Declarative
- Ordered
- Scalable
- Routing-capable
- Failure Modes
- Bidirectional
- Rejection

Declarative Approach

- Inline: at the HTTPRoute- or Gateway- level
- Reusable: as a CRD, referenceable by multiple resources
- Processing locally or remotely

```
apiVersion: gateway.networking.k8s.io/v1
kind: HTTPRoute
metadata:
  name: example-httproute
spec:
  parentRefs:
  - name: httpbin-gateway
  hostnames:
  - "www.example.com"
  rules:
  - matches:
    - path:
        type: PathPrefix
        value: /query
    backendRefs:
    - name: example-svc
      port: 8080
    filters:
    - type: PayloadProcessing
      payloadProcessing:
        name: guardrails
        backendRefs:
        - kind: Service
          name: sql-injection-scanner
        config:
          timeout: "500ms"
          failureMode: open | closed
          # Context to send in gRPC(?) Metadata
      context:
        tenant_id: "customer-a"
```

Open Questions

- Protocol
- Interactions with existing filters
- Interactions between different levels
- Processing loops



KubeCon



CloudNativeCon

Europe 2026

Egress Gateway

Why is our 'AI Gateway' also an Egress Gateway?

- Greater emphasis on application layer policy breaks today's de facto egress standards.
- Gateway API is producer focused.

“Leading” Question:

- What does egress policy look like when the payload matters?
 - Per route seems too fine grained.
 - Per gateway seems too coarse grained.

Vendor fragmentation lights the way

Addressing Vendor Fragmentation

Fragmentation in ingress implementation led to Gateway API.

Fragmentation in egress implementation reveals a similar need.

Ingress

host: foo.com
path: /api
tls: ...
everything else = annotations

API has trouble handling novel use cases, results in vendor specific annotations.

Gateway API

Gateway (Infra)
HTTPRoute (routing)
*Policy (TLS, etc...)

Separated concerns creates more room in the API for novel use cases.

Similarly..

ExternalName

HTTPRoute
BackendTLSPolicy
- Difficult to express per client TLS, protocol and policy

Lack of API supported client connection semantics = vendor fragmentation

Egress Gateway

Backend (DNS, protocol, TLS)
HTTPRoute
- Backend bundles connection semantics from consumer POV; makes per endpoint policy easy.

More room in API for client side connection semantics.

And..

Backend gives us a place to attach L7 processing...

Vendor fragmentation lights the way

The **Backend** represents the client POV

```
apiVersion: gateway.networking.k8s.io/v1alpha1
kind: Backend
metadata:
  name: openai-backend
spec:
  destination:
    type: FQDN
    fqdn:
      hostname: api.openai.com
  ports:
    - number: 443
      protocol: TLS
      tls:
        mode: None | Simple | Mutual
        sni: api.openai.com
        caBundleRef:
          name: vendor-ca
  extensions:
    - name: inject-credentials
      type: gateway.networking.k8s.io/CredentialInjector:v1
      phase: request-headers
      priority: 10
      failOpen: false
      config:
        secretRef:
          name: openai-api-key
          namespace: platform-secrets
```

The Backend Resource and the Agent Subproject

- **Backend** originated in the agentic subproject
- Agentic Networking also wanted a way to express a destination whether:
 - It was an internal **Service**
 - An external endpoint

Open Questions

Do we:

- **BackendTLSPolicy** versus inline TLS
 - Leaning toward: inline.
- **Backend** scoping
 - Leaning toward: namespace
- Represent **Service** or **InferencePool** via **Backend**
- Reuse **Gateway** or add **EgressGateway**



KubeCon



CloudNativeCon

Europe 2026

Live Demo



KubeCon



CloudNativeCon

Europe 2026

Future Plans

Future Plans

- Exploration of payload processing pipelines for explicit ordering and efficiency.
- Support for internal/external backend representation.
- Token Rate Limits.
- Enabling the reuse of the WG outputs in Batch inference scenarios (Collaboration with WG Batch).



KubeCon



CloudNativeCon

Europe 2026

Get Involved

Get Involved

- Visit the working group [GitHub repository](#).
- Join the [weekly meeting](#) on Thursdays at 2PM EST
- Connect with us on [Slack \(#wg-ai-gateway\)](#).
(visit <https://slack.k8s.io/> for an invitation).
- Join the AI Gateway [mailing list](#).



KubeCon



CloudNativeCon

Europe 2026

Thank You!

