



KubeCon



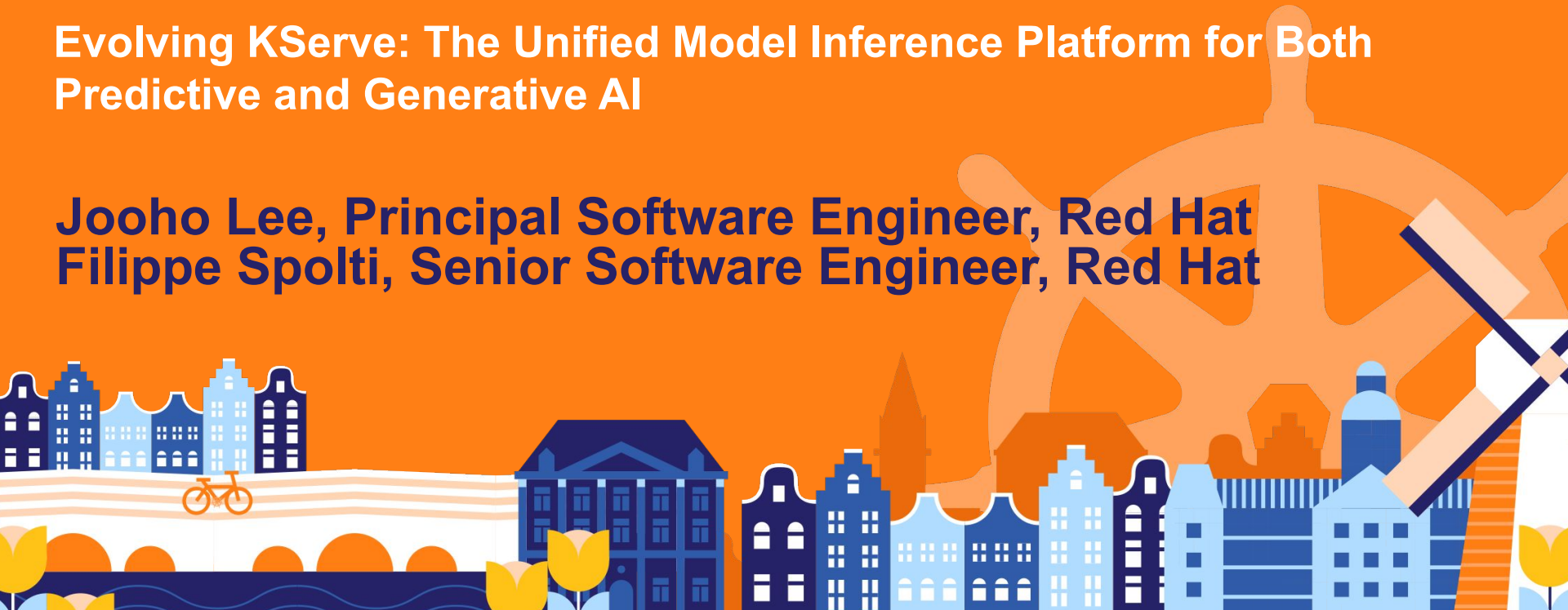
CloudNativeCon

Europe 2026

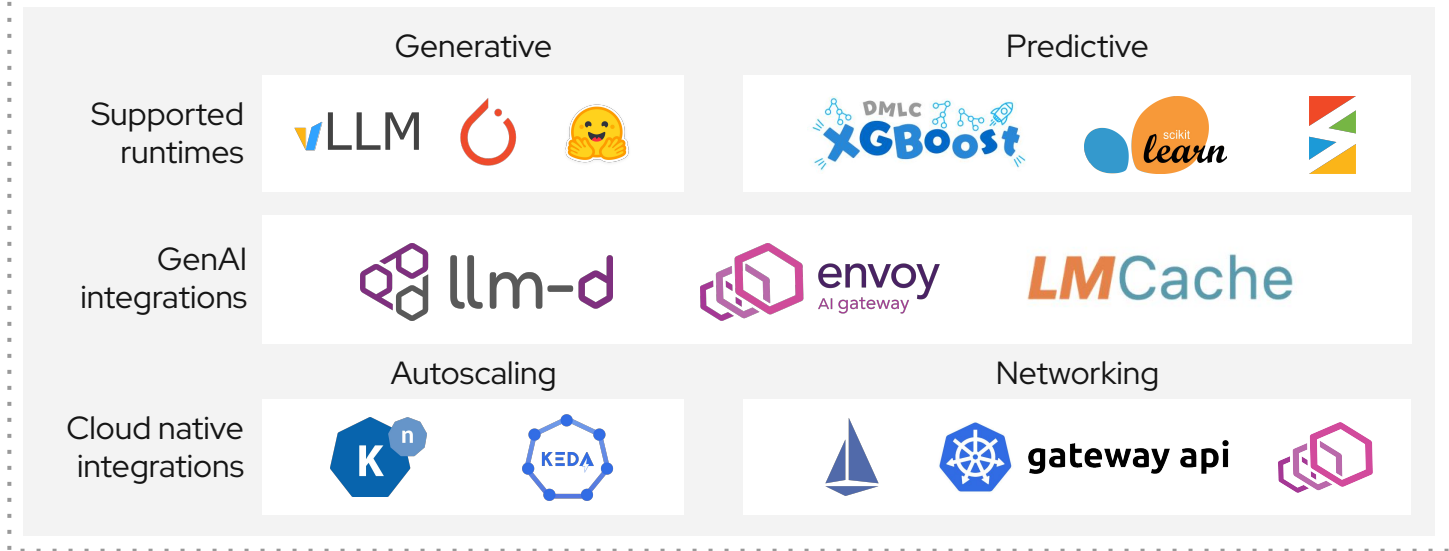
#KubeCon #CloudNativeCon

Evolving KServe: The Unified Model Inference Platform for Both Predictive and Generative AI

Jooho Lee, Principal Software Engineer, Red Hat
Filippe Spolti, Senior Software Engineer, Red Hat



KServe Model inference platform



Orchestration



Hardware accelerators (GPUs, CPUs, etc.)



KubeCon



CloudNativeCon

Europe 2026

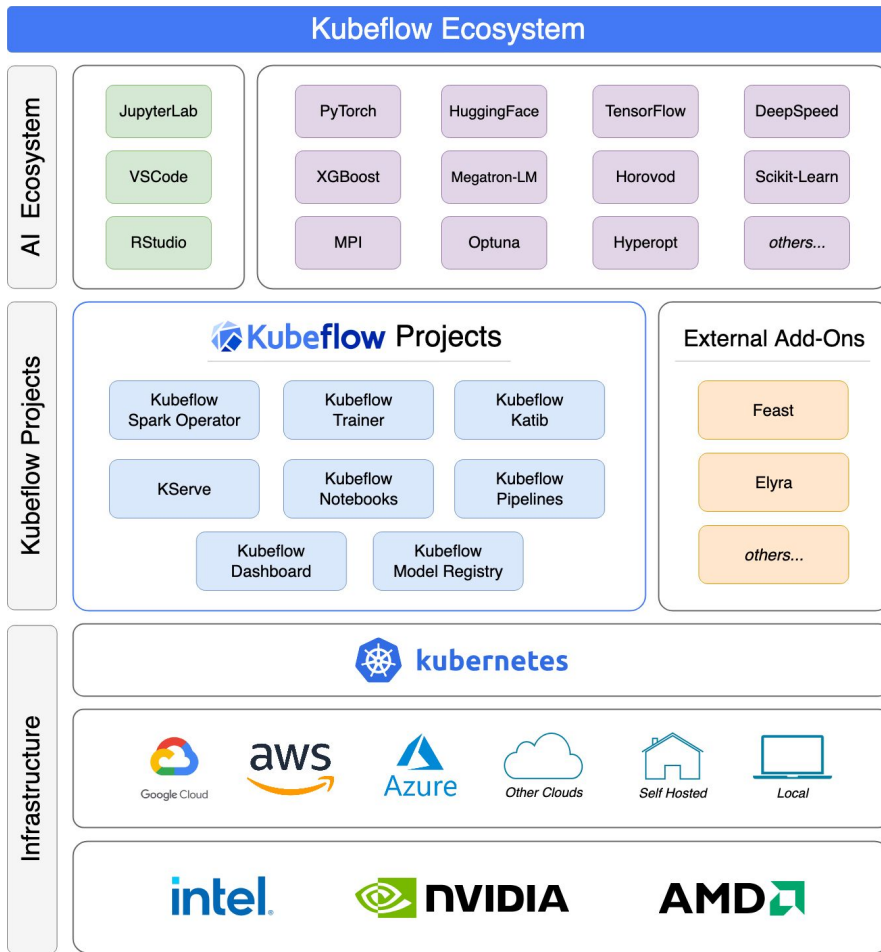
Our Journey



Our Journey

2019:

Developed collaboratively by Google, IBM, Bloomberg, NVIDIA, and Seldon under the [Kubeflow](#) project as [KFServing](#).



Our Journey

2021:

Rebranded from [KFServing](#) to standalone [KServe](#) project.

KServe: The next generation of KFServing

KServe Blog Post

Sep 27, 2021 • 2 min read

 release official

By **Dan Sun** and **Animesh Singh** on behalf of the Kubeflow Serving Working Group

KFServing is now KServe

We are excited to announce the next chapter for KFServing. In coordination with the Kubeflow Project Steering Group, the [KFServing GitHub repository](#) has now been transferred to an independent [KServe GitHub organization](#) under the stewardship of the Kubeflow Serving Working Group leads.

The project has been rebranded from **KFServing** to **KServe**, and we are planning to graduate the project from Kubeflow Project later this year.

<https://blog.kubeflow.org/release/official/2021/09/27/kfserving-transition.html>

Our Journey

2022: KServe joined [LF AI & Data Foundation](#).

LF AI & Data Blog

KServe Joins LF AI & Data as New Incubation Project

By Andrew Bringaze | February 24, 2022 | No Comments

By Animesh Singh, Dan Sun, Alexa Griffith on behalf of KServe community

Today, we are announcing [KServe](#) is joining the LF AI & Data Foundation as its latest Incubation Project. LF AI & Data is the organization building an ecosystem to sustain innovation in artificial intelligence (AI) and data open source projects.

<https://lfaidata.foundation/blog/2022/02/24/kserve-joins-lf-ai-data-as-new-incubation-project/>

Our Journey

Sept 2025: KServe
was accepted as a
CNCF incubating
project!

Our Incubating Projects

[VIEW ON CNCF LANDSCAPE](#)



ArtifactHUB

Application Definition &
Image Build



Backstage

Application Definition &
Image Build



Buildpacks.io

Application Definition &
Image Build



Chaos Mesh

Chaos Engineering



Cloud
Custodian

Automation &
Configuration



CNI

Cloud Native Network



CONTOUR

Service Proxy



cortex

Observability



Crossplane

Scheduling &
Orchestration



Dragonfly

Container Registry



EMISSARY
INGRESS

API Gateway



Certified Kubernetes -
Distribution



Remote Procedure Call



KARMADA

Scheduling &
Orchestration



KEYCLOAK

Security & Compliance



ML Serving



Kubeflow

Scheduling &
Orchestration



Kubescape

Security & Compliance



KubeVela

Application Definition &
Image Build



KubeVirt

Application Definition &
Image Build



KubeCon



CloudNativeCon

Europe 2026



KubeCon



CloudNativeCon

Europe 2026

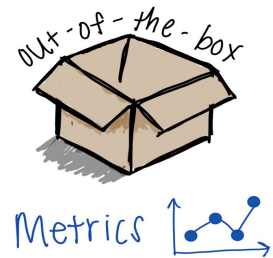
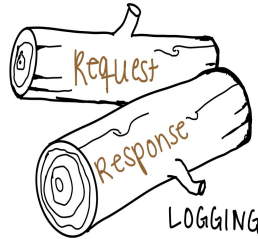
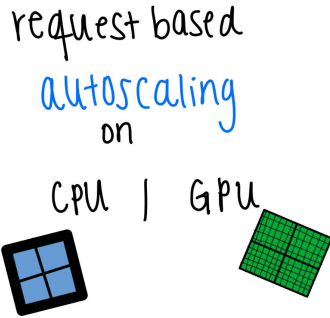
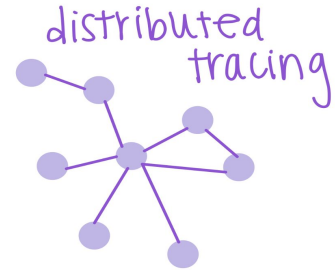
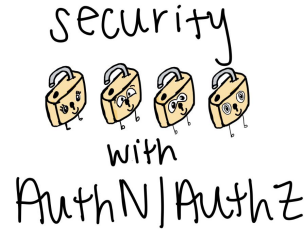
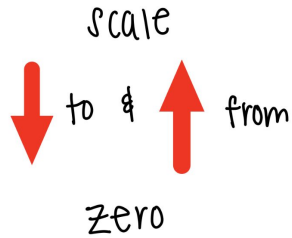
What is KServe?



Assembling AI with KServe

KServe is a platform that unifies **Generative** and **Predictive** AI Inference on Kubernetes. It is simple enough for quick deployments, yet powerful enough to handle enterprise-scale AI workloads with advanced features.

KServe Predictive AI Inference



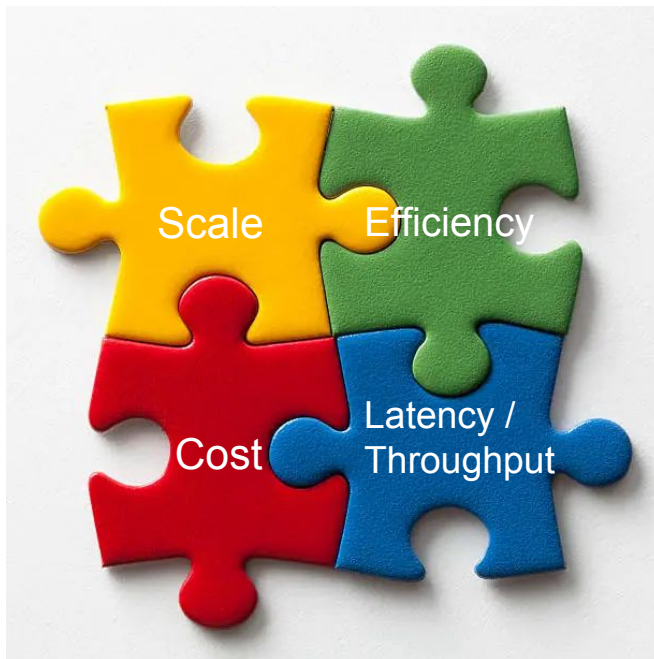
AI Inference Traffic

Feature	Predictive AI Traffic	Generative AI Traffic
Task	Classification, Regression	Generate new data: text, image, code
Input Traffic	Small, structured and high volume	Variable, text/multi-modal Traffic can be bursty
Output Traffic	Small with single label or score	Large, streaming, high bandwidth
Latency	Ultra-low latency	Moderate to high latency
Computation/ Scaling	CPU or GPU on L4 for transformer models Horizontal scaling and maximize QPS	Heavy computation on H100/H200 with high memory bandwidth Efficient GPU/VRAM utilization

KServe GenAI: Advanced Building Blocks for LLMs

LLM Metric-Based
Autoscaling

Prompt Caching

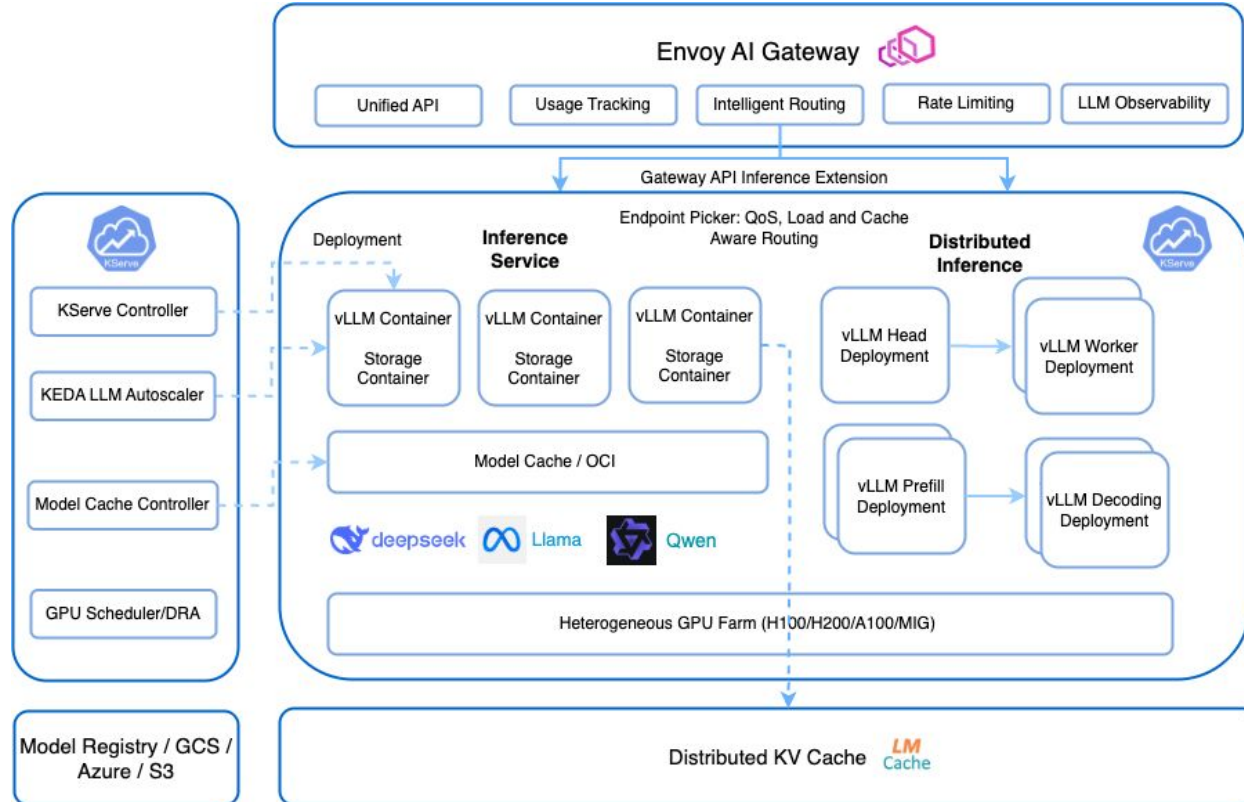


GenAI Runtime

vLLM, TRT-LLM,
llm-d

Intelligent Routing,
Traffic Management

KServe GenAI Architecture



KubeCon



CloudNativeCon

Europe 2026



KubeCon



CloudNativeCon

Europe 2026

Highlighted Features



Why Concurrency Fails for LLMs?

LLMs run on expensive GPUs; idle models are massive wastes of money

Default KServe Autoscaler

In-flight concurrency

A request is concurrent from the moment it is received until the last token is sent

Why Concurrency Fails ?

Bottleneck is the GPU VRAM for KV Cache

A single long prompt can fill VRAM, but KPA only sees **concurrency = 1**

KServe Metric based Autoscaler

KV Cache VRAM %

When VRAM is full, the model can't accept more requests; we must scale before hitting 100%

GPU Utilization %

Measure the compute load, which is a good secondary metric

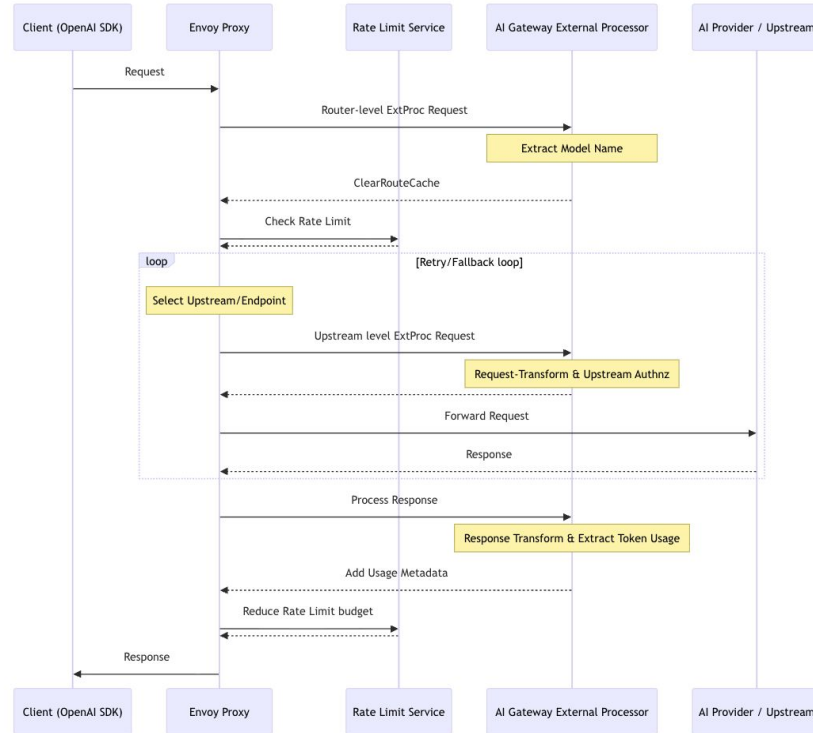
```
predictor:  
  model:  
    modelFormat:  
      name: huggingface  
      storageUri: "hf://Qwen2.5"  
    minReplicas: 1  
    maxReplicas: 5  
    autoScaling:  
      metrics:  
        - type: PodMetric  
          podmetric:  
            metric:  
              query:  
                "vllm:kv_cache_usage_percent"  
            target:  
              type: Value  
              value: "75"
```

Envoy AI Gateway: Token Rate Limiting

Rate Limit for LLMs

Token usage extraction from response body

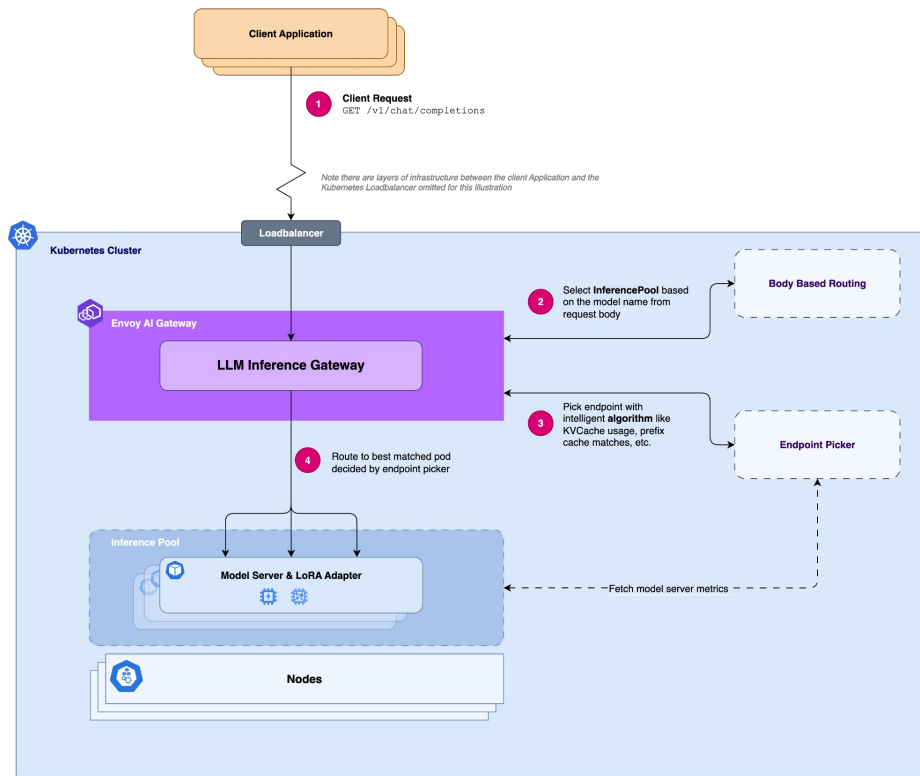
The “cost” is stored in the envoy dynamic metadata



Envoy AI Gateway: Inference Extension

Traditional load balancing falls short for LLM inference workloads

Intelligent endpoint picking with KV cache usage and LLM load information



GenAI Feature: llm-d Integration



An open-source framework for distributed large language model (LLM) inference that runs natively on Kubernetes

<https://github.com/llm-d/llm-d>



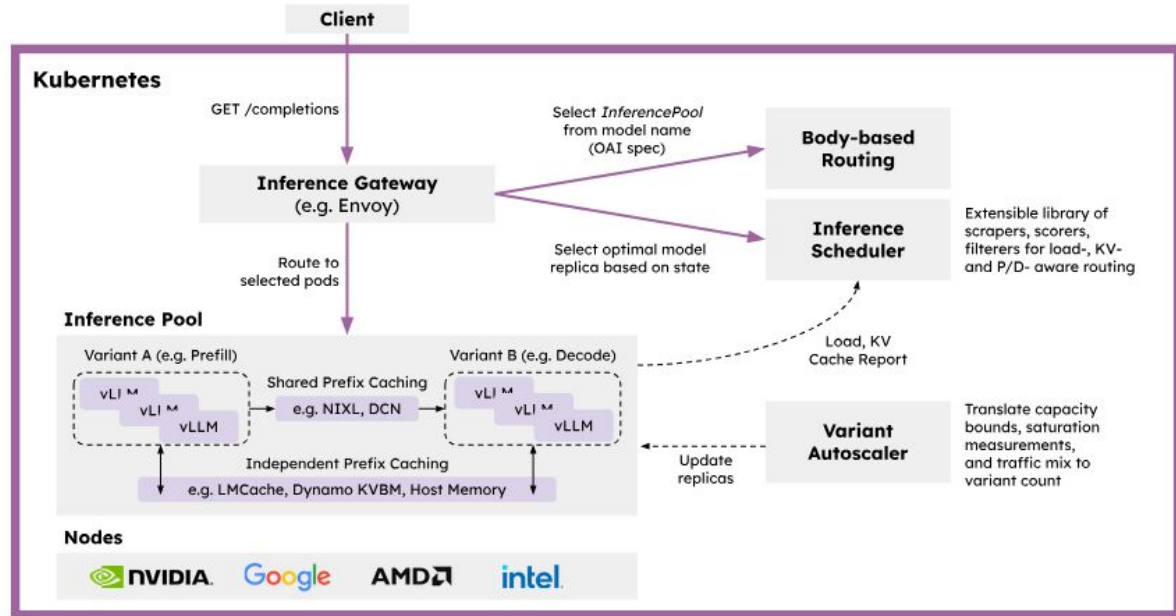
GenAI Feature: Ilm-d Integration

Intelligent Inference Scheduler

Prefix Caching

P/D Disaggregated Serving

Variant Autoscaling



LLM Inference Workload

Single-Node/Multi-GPU:

SLM (less than 70B)

Multi-Node/Multi-GPU: Distributed Inference for large models

Disaggregated Prefill/Decode: High Throughput Requirement

Single-Node Workload

spec.template (Deployment)

- single pod per replica
- No distributed parallelism
- Suitable for small models

Multi-Node Workload

spec.template + spec.worker (LeaderWorkerSet)

- Leader pod (head/master)
- Worker pods (N worker per replicas)
- Tensor/Pipeline/Data parallelism
- Suitable for large models

Disaggregated Workload

spec.prefill
(Prefill pool)

- Leader pod (head/master)
- Process prompts
- Generate KV cache
- High concurrency

spec.template
(Decode pool)

- Generate tokens
- Use KV cache
- Sequential decode

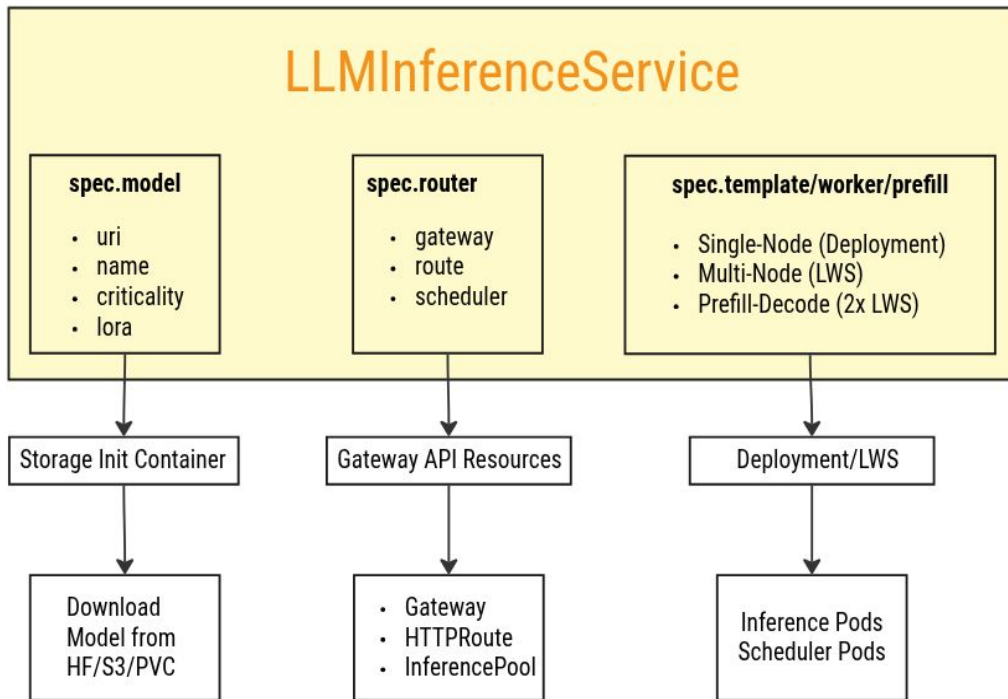
LLM Inference Control Plane API

Model Spec: Model URL,
LORA

Router Spec: Gateway API,
Scheduler

Workload Spec: Single/Multi
Node, Prefill/Decode

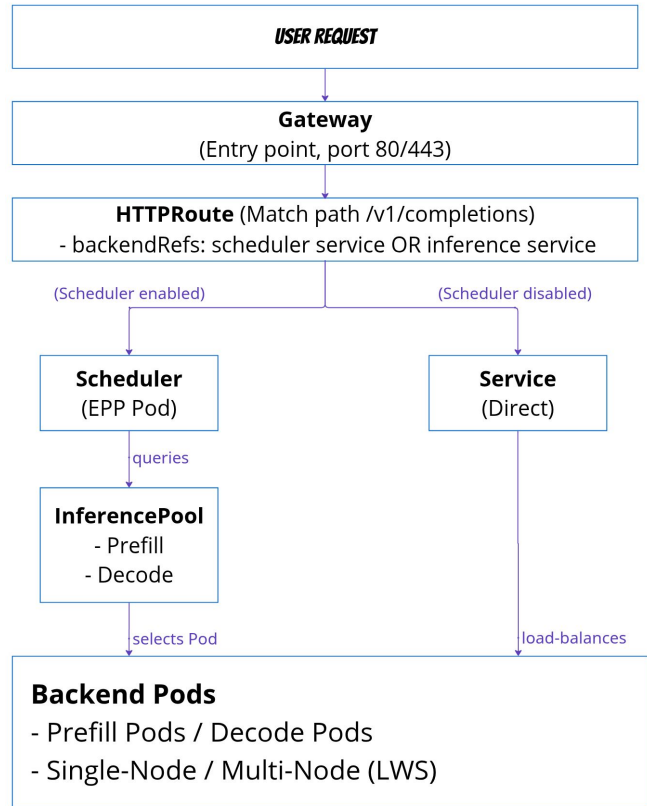
Parallelism Spec: TP/DP/EP



LLM Inference Data Plane

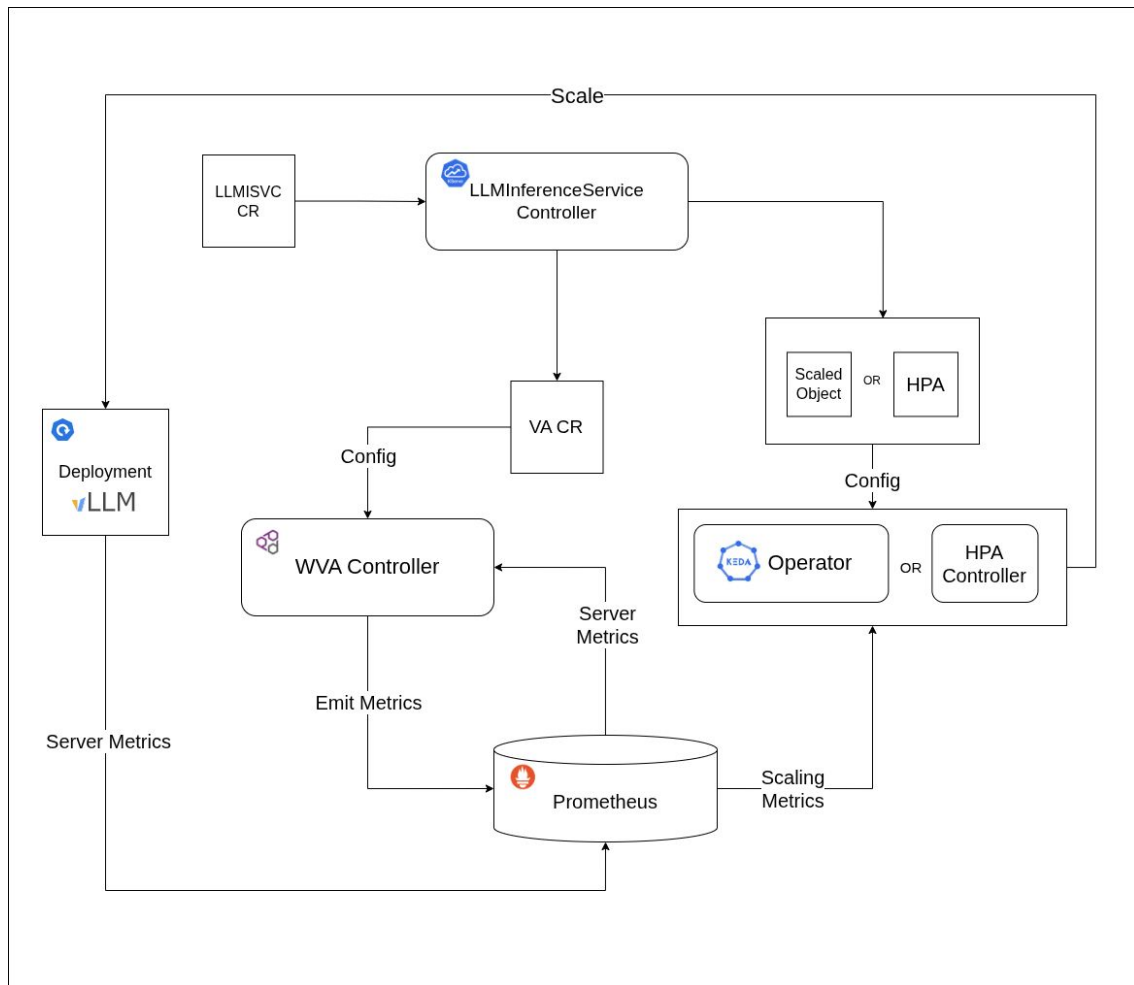
Intelligent Endpoint Picking

- Prefix Cache Routing
- Prefill-Decode Pool Routing
- Load-Aware Routing



WVA

- **WVA-Driven Intelligence**
- **Pluggable Actuators**
- **Single-CR Management**
- **Disaggregated serving support**



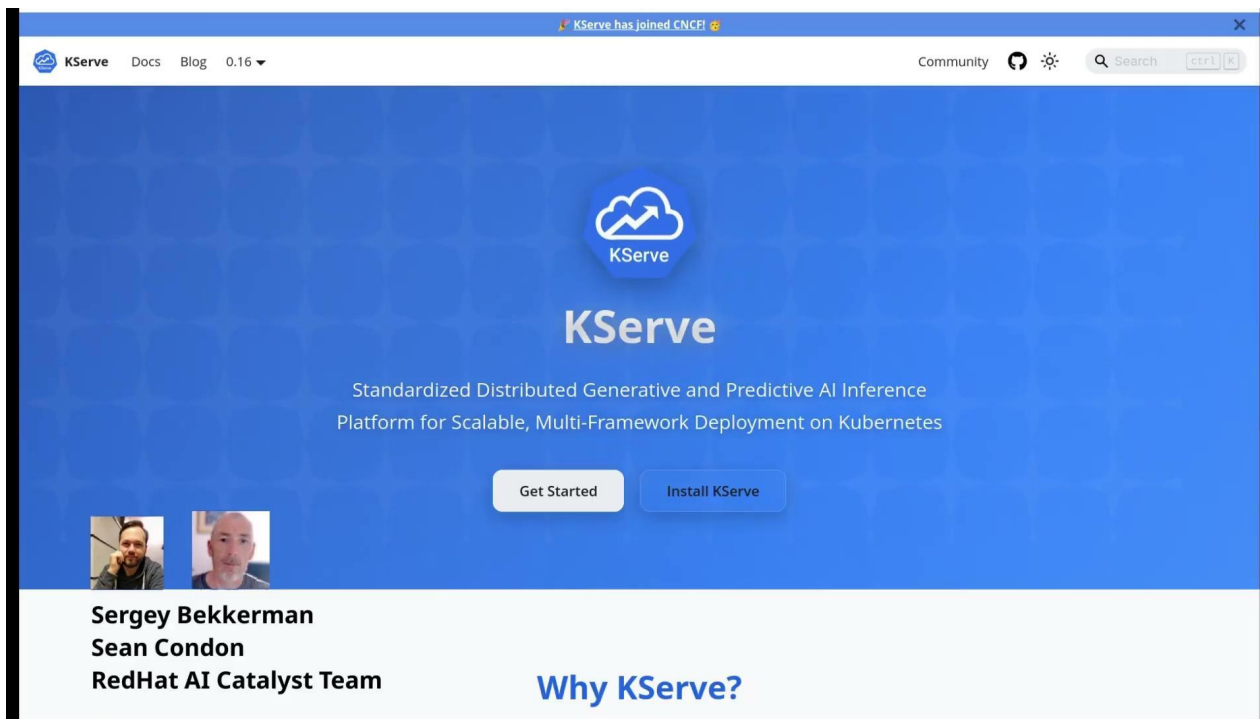
KubeCon



CloudNativeCon

Europe 2026

KServe LLMInferenceService Demo



The screenshot shows the KServe website with a blue background and a grid pattern. At the top, there is a navigation bar with the KServe logo, links to Docs, Blog, and a version dropdown (0.16). On the right, there are links for Community, a settings icon, and a search bar. The main content area features the KServe logo (a cloud with an upward arrow) and the text "KServe". Below this, it says "Standardized Distributed Generative and Predictive AI Inference Platform for Scalable, Multi-Framework Deployment on Kubernetes". There are two buttons: "Get Started" and "Install KServe". At the bottom left, there are two small profile pictures of Sergey Bekkerman and Sean Condon, followed by their names and "RedHat AI Catalyst Team". At the bottom right, there is a link "Why KServe?".

KServe has joined CNCF!

KServe Docs Blog 0.16

Community Search


KServe

KServe

Standardized Distributed Generative and Predictive AI Inference
Platform for Scalable, Multi-Framework Deployment on Kubernetes

[Get Started](#) [Install KServe](#)

 
Sergey Bekkerman
Sean Condon
RedHat AI Catalyst Team

[Why KServe?](#)



KubeCon



CloudNativeCon

Europe 2026

Roadmap

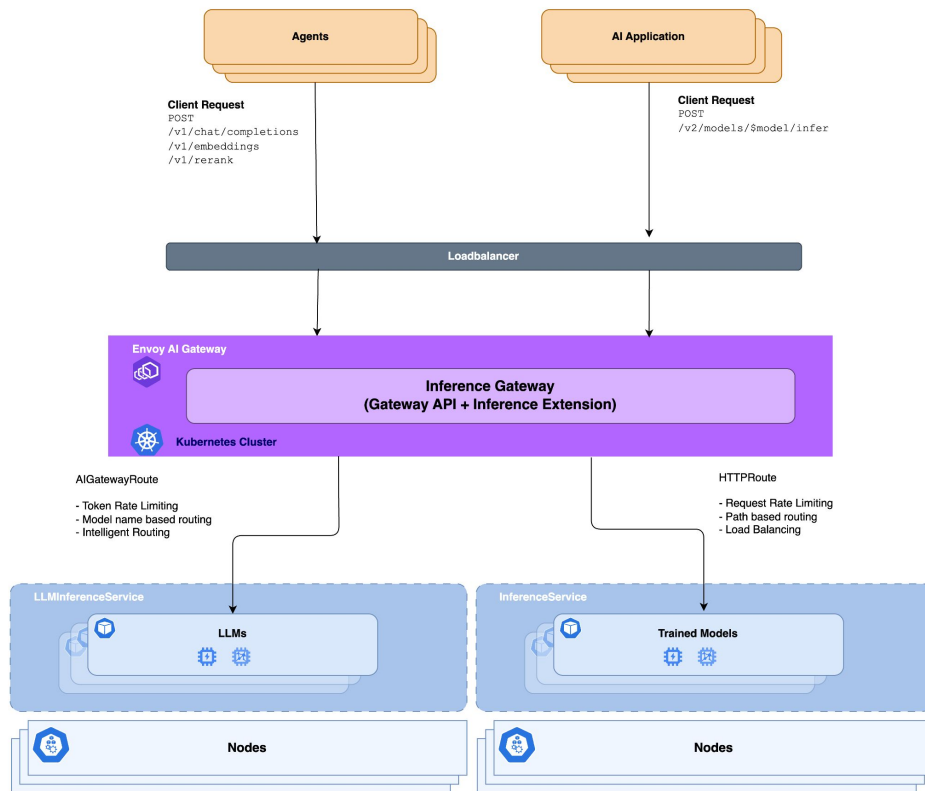


Unified Inference Fabric

Core Objectives: Unified Inference Fabric

- Centralized platform for serving all open-source and fine-tuned, trained models
- Standardized APIs and telemetry for consistency
- Fully managed inference experience with optimized performance and efficient compute resource utilization
- Secured identity for Agent-to-LLM traffic with SPIFFE/SPIRE

KServe: Gateway API to Unify Inference Workload





KubeCon



CloudNativeCon

Europe 2026

Community



19 Maintainers and 300+ Contributors!

Bloomberg

Engineering



Red Hat

CLOUDERA

INTUIT

NUTANIX



<https://github.com/kserve/community/blob/main/MAINTAINERS.md>

Trusted by industry leaders

KServe is used in production by organizations across various industries, providing reliable model inference at scale.

Bloomberg



CLOUDERA



inspur



NAVER



INTUIT

Join Our Community

- Repo: <https://github.com/kserve/kserve>
- Website: <https://kserve.github.io>
- Biweekly community meetings on Thursdays at 9 AM PST
- [#kserve](#) and [#kserve-contributors](#) channels in the CNCF Slack



<https://github.com/kserve/kserve>