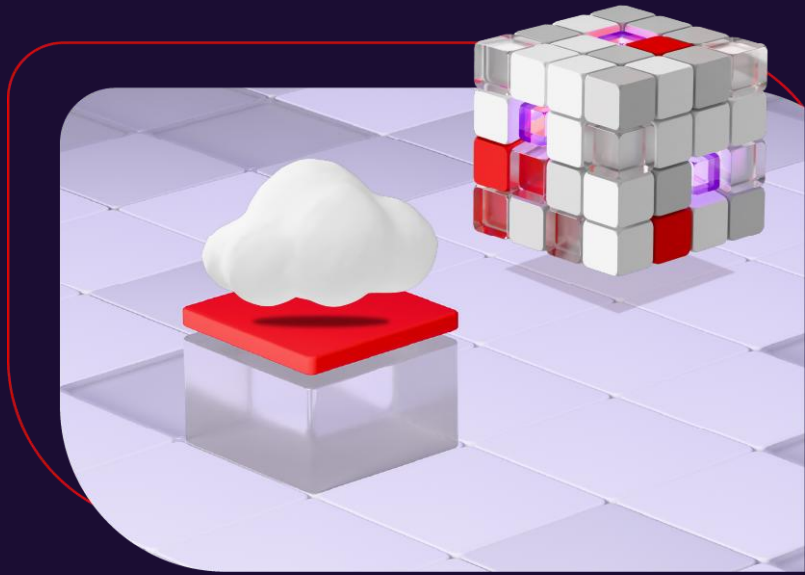
Scaling Distributed Inference with llm-d

vLLM Meetup Bangkok, 6 Sep 2026

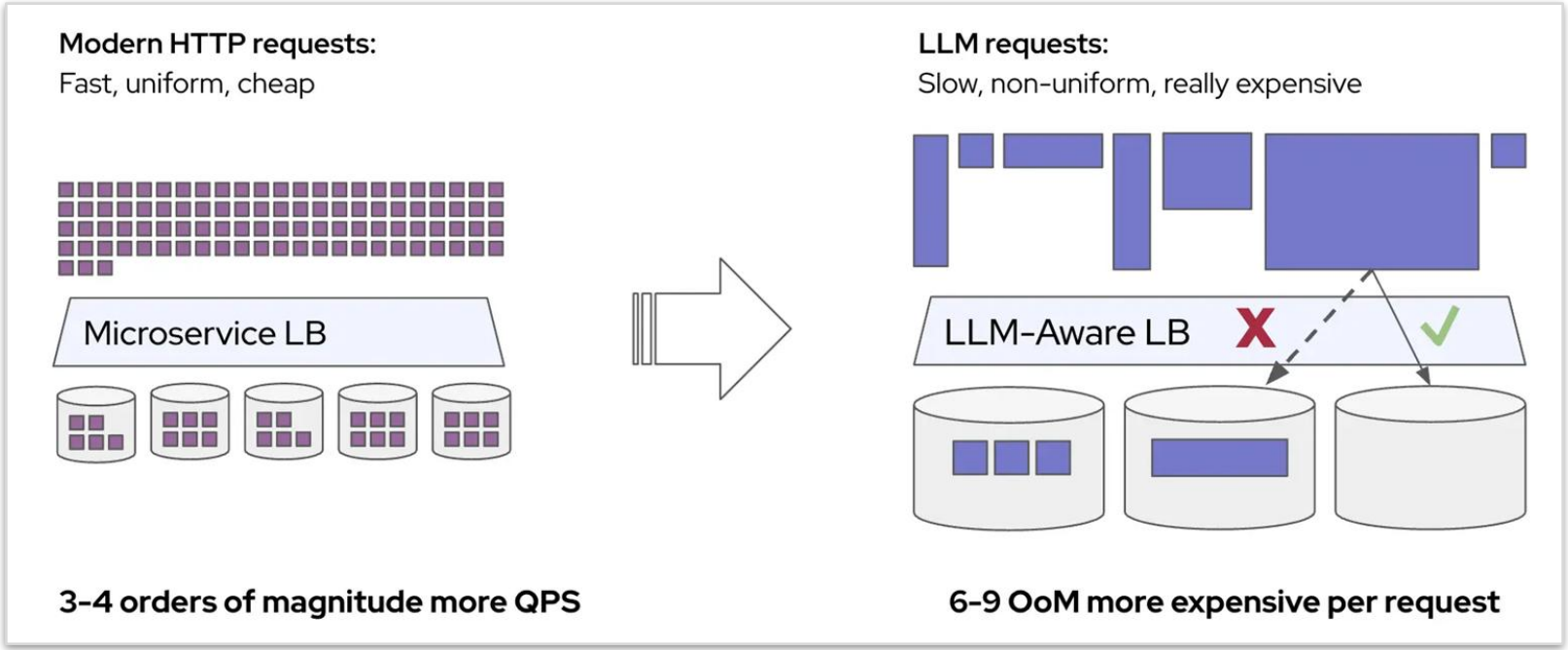
Jing Wen Ng

Specialist Solution Architect, AI, Asia Pacific @ Red Hat

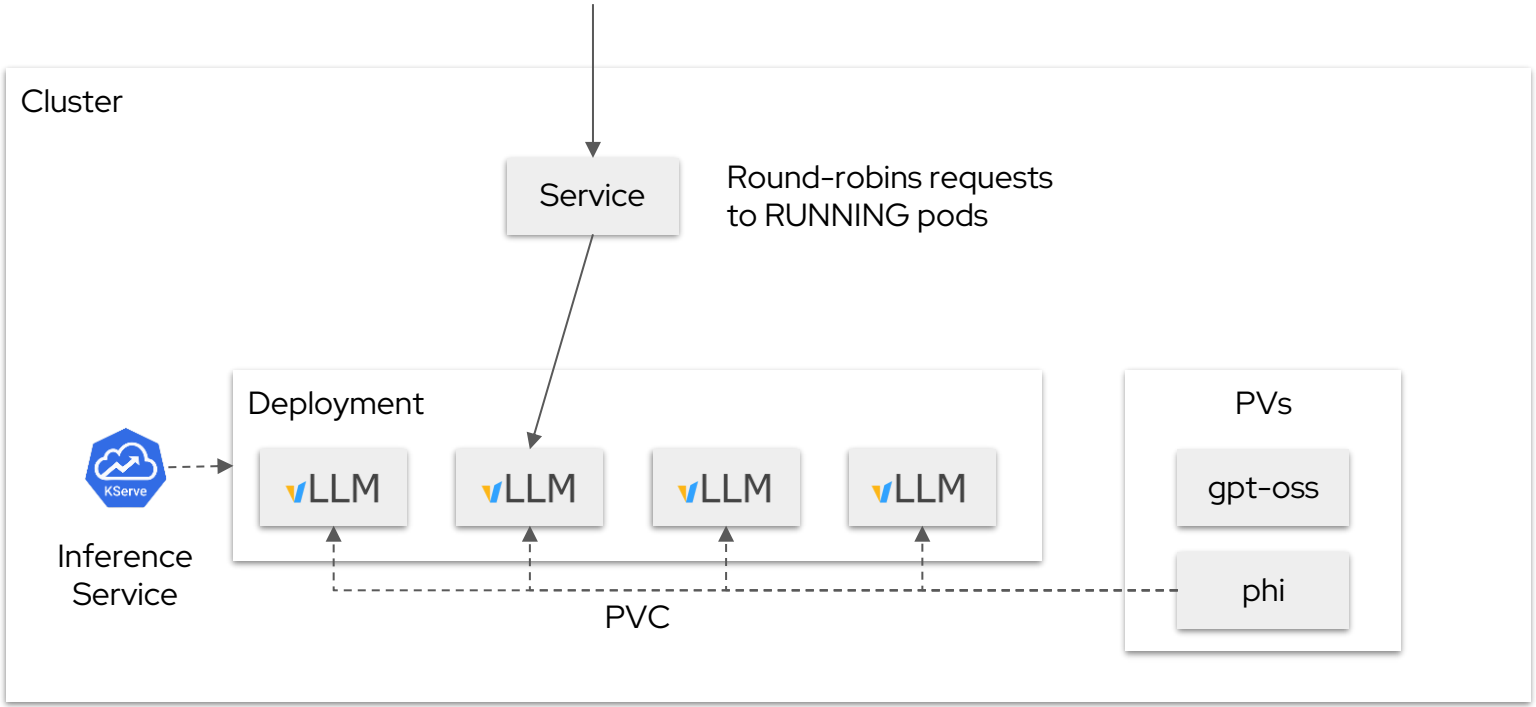
The Problem



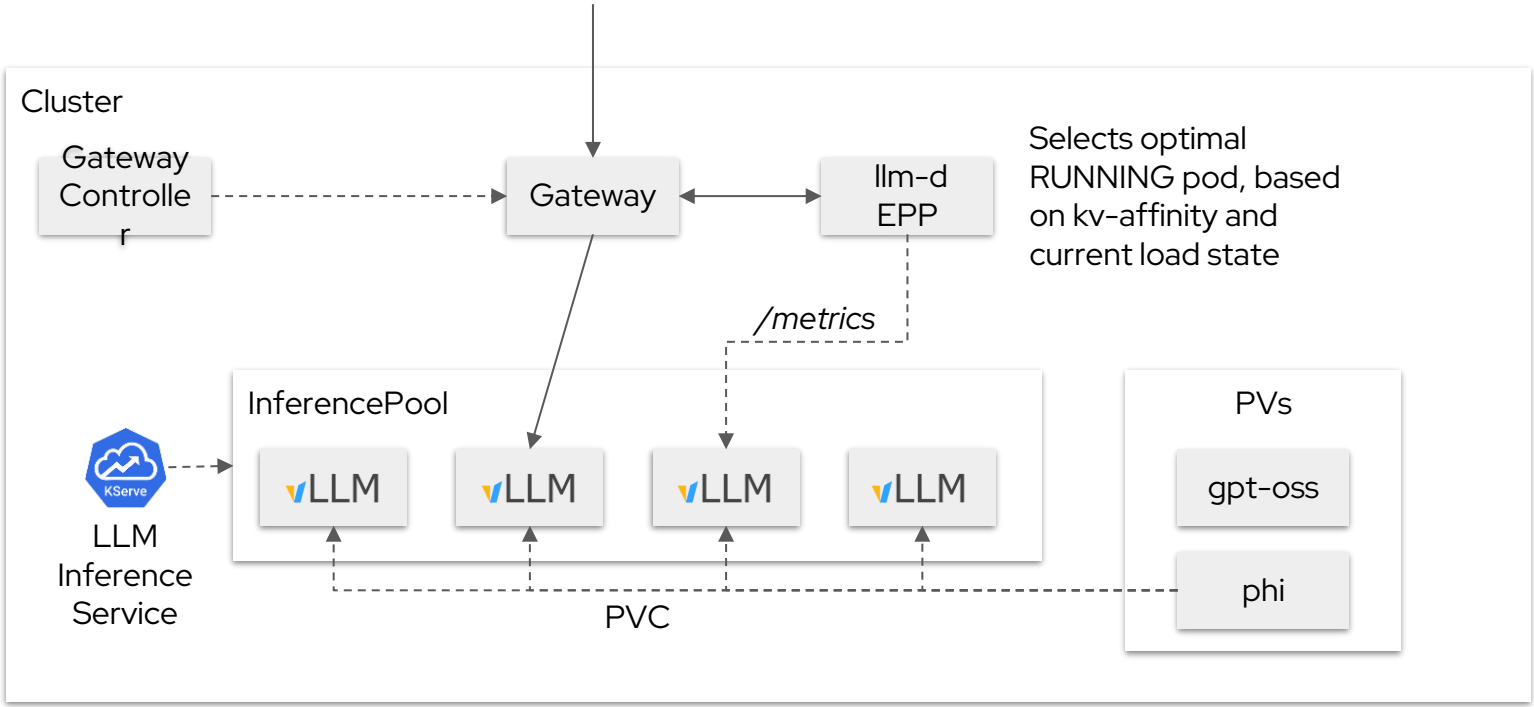
Why Distributed Inference?



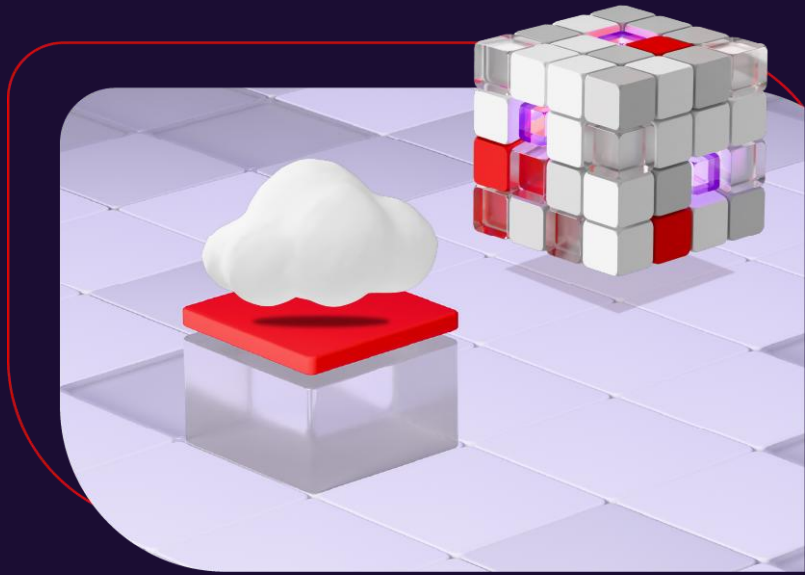
Baseline Deployment Architecture



RHOAI llm-d Deployment Architecture



What is llm-d?



What is llm-d?

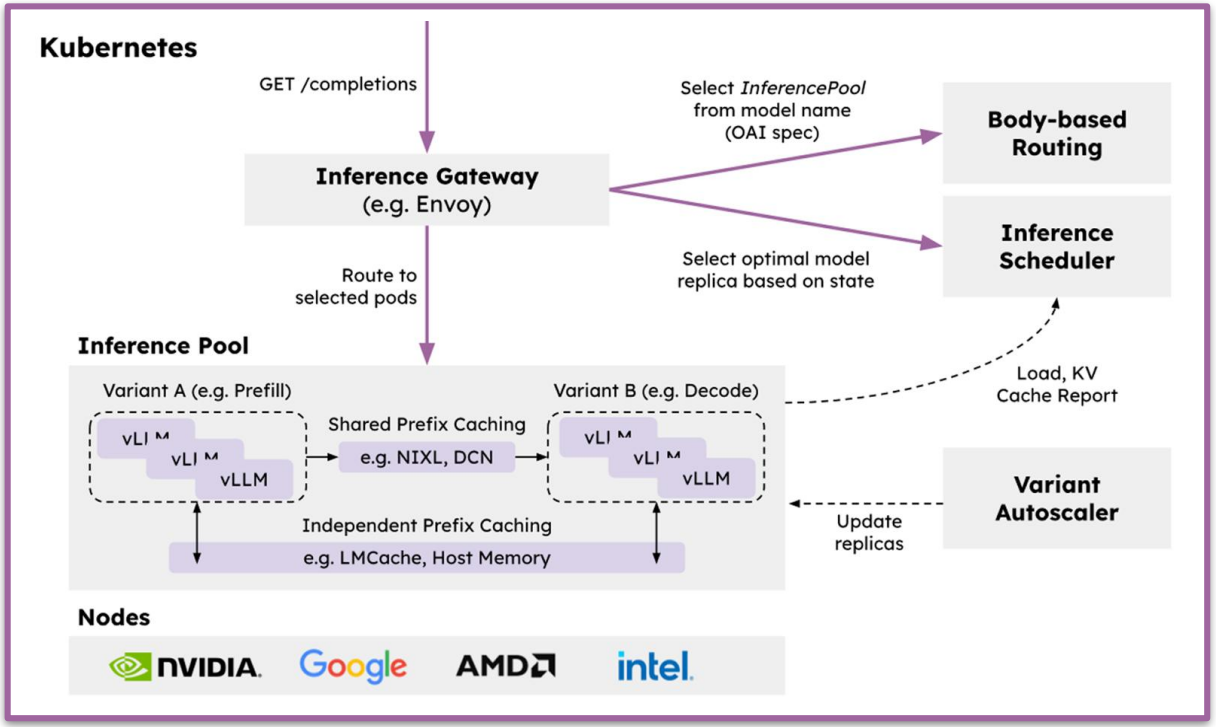
An open-source, Kubernetes-native framework for distributed LLM inference

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

- Joint open source initiative by Red Hat, Google, NVIDIA, Hugging Face, and many more organizations
- Optimized for Kubernetes based distributed platforms (OpenShift)
- Designed to optimize scale, latency, and flexibility for AI workloads
- Built to extend and interoperate with vLLM
- Maximizes performance across multi-tenant and multi-model workloads with intelligent scheduling and resource management

Red Hat

llm-d Architecture



- **Operationalizability:** modular and resilient architecture with native integration into Kubernetes via Inference Gateway API
- **Flexibility:** cross-platform (active work to support NVIDIA, Google TPU, AMD, and Intel), with extensible implementations of key composable layers of the stack
- **Performance**: leverage distributed optimizations like disaggregation and prefix-aware routing to achieve the highest efficiency for available infra

Red Hat

Key Features (“Well-Lit” Paths)

setting_suggestions

Intelligent Inference Scheduling
KV-cache aware, load-aware routing

memory

KV Cache Management
CPU-offloading & Storage-offloading

call_split

Prefill-Decode Disaggregation
Decoupled execution for higher throughput

lan

Multi-Node Serving
“Wide Expert Parallel” serving architecture

traffic

Flow Control
Multi-tenancy & fairness-scheduling

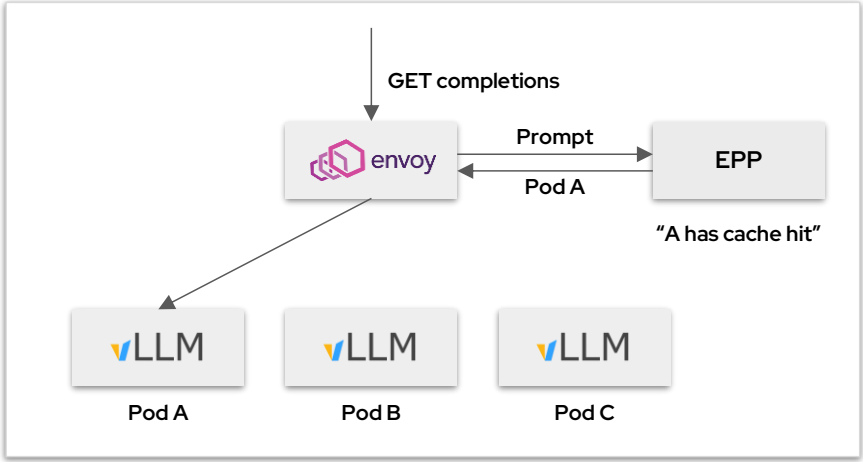
trending_up

Workload Variant Autoscaling
Dynamic and highly responsive auto-scaling

Intelligent Inference Scheduling

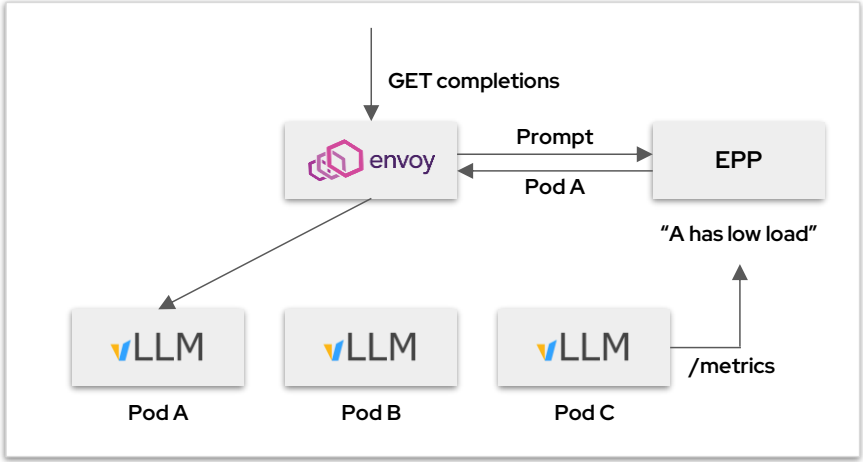
vLLM-aware load-balancing enables smarter request routing that improve SLOs

Prefix-Aware Routing



Dramatically increase prefix-cache hit rate

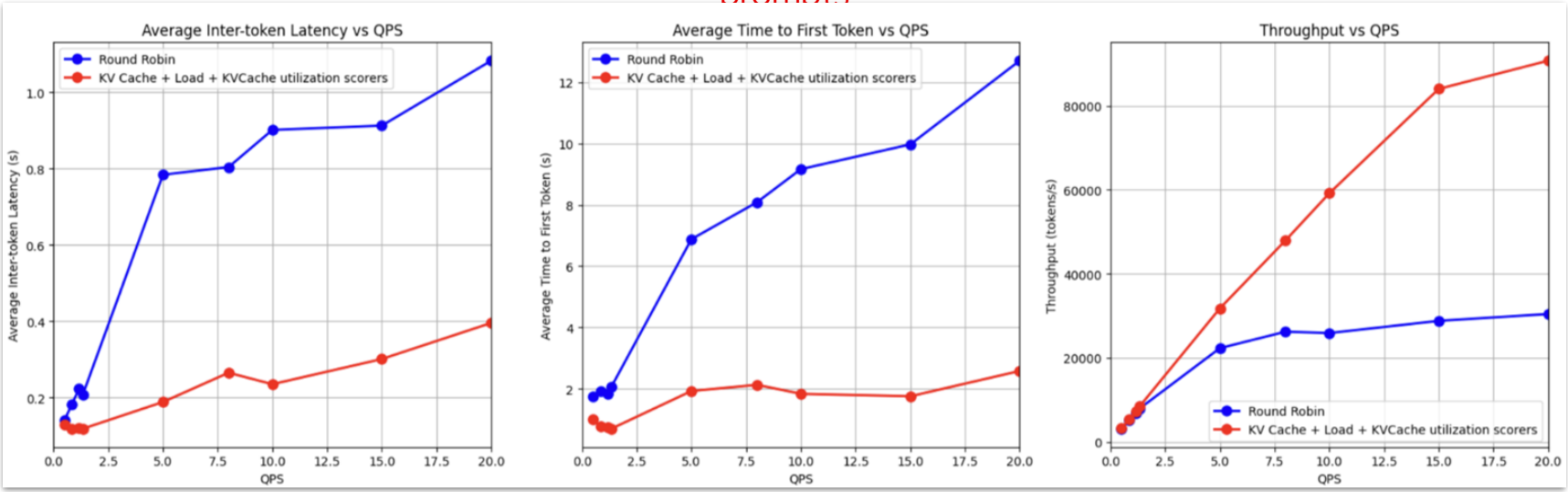
Load-Aware Routing



Load-balancing based on actual replica state

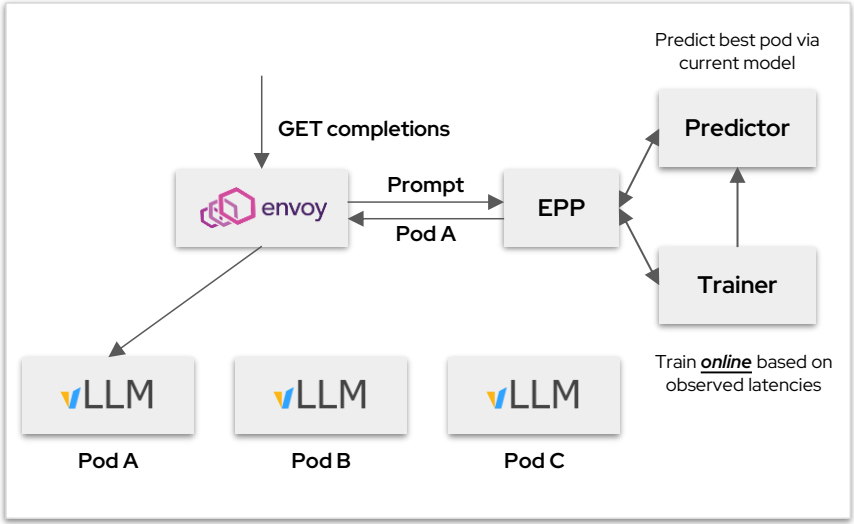
Intelligent Inference Scheduling

Inference scheduling is a no-brainer optimization which can have huge impacts on repeated prompts



Predicted Latency Scheduling

We are currently improving the Intelligent Inference Scheduling with predicted latency



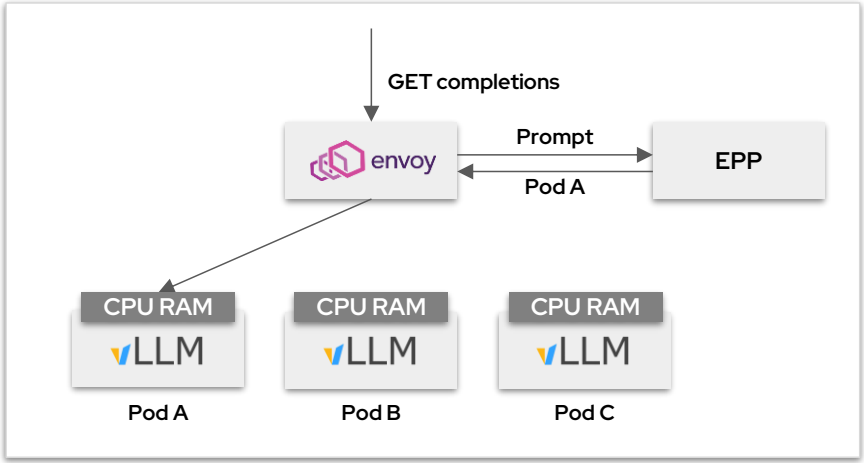
Scenario	Success Rate	E2E p50	E2E p95	TTFT p50	TTFT p95	TPOT p50	TPOT p99
No Gateway	~99.8%	15.98s	38.85s	4.47s	24.04s	35ms	93ms
Load + Prefix (111)	~99.9%	16.42s	35.06s	2.86s	18.06s	39ms	103ms
Load + Prefix (322)	100%	13.42s	26.55s	3.38s	16.78s	28ms	63ms
Latency Prediction	**~99.9%**	**9.06s**	**22.57s**	**0.97s**	**11.34s**	**22ms**	**53ms**

https://llm-d.ai/blog/predicted-latency-based-scheduling-for-llms

KV Cache Management

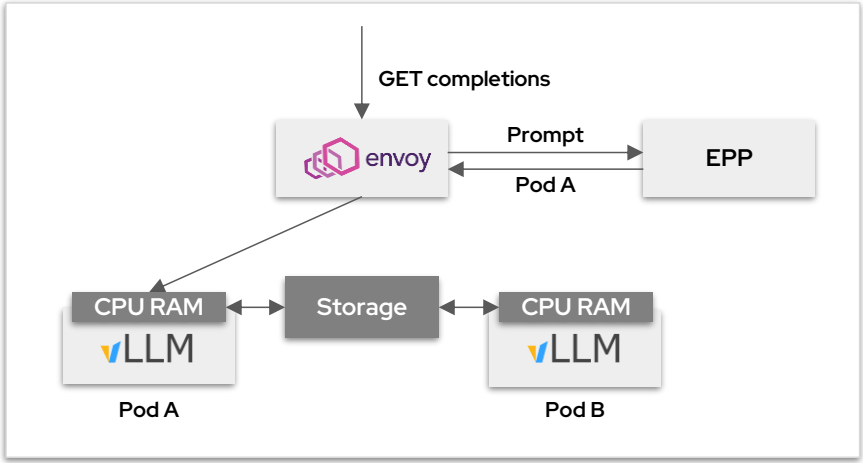
Leverage all system resources to maximize prefix cache hit rate within the cluster

CPU KV Cache Offloading



Offload KV caches to local memory with global index

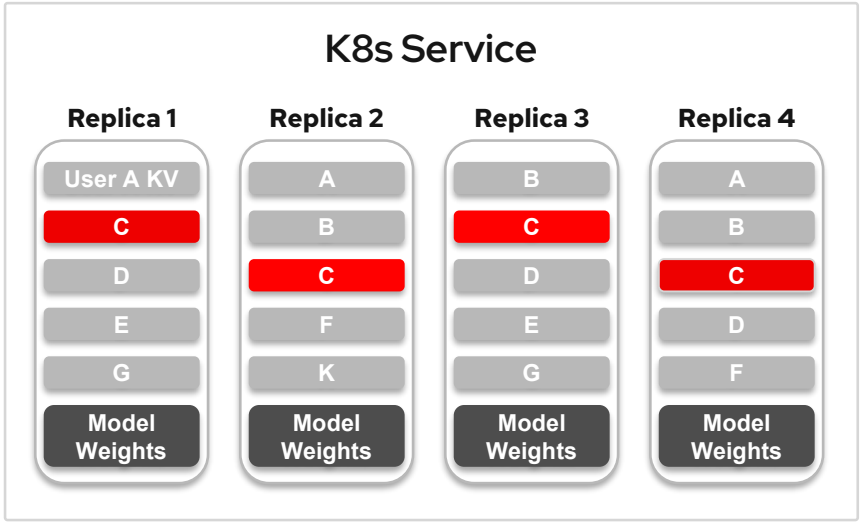
Storage KV Cache Offloading



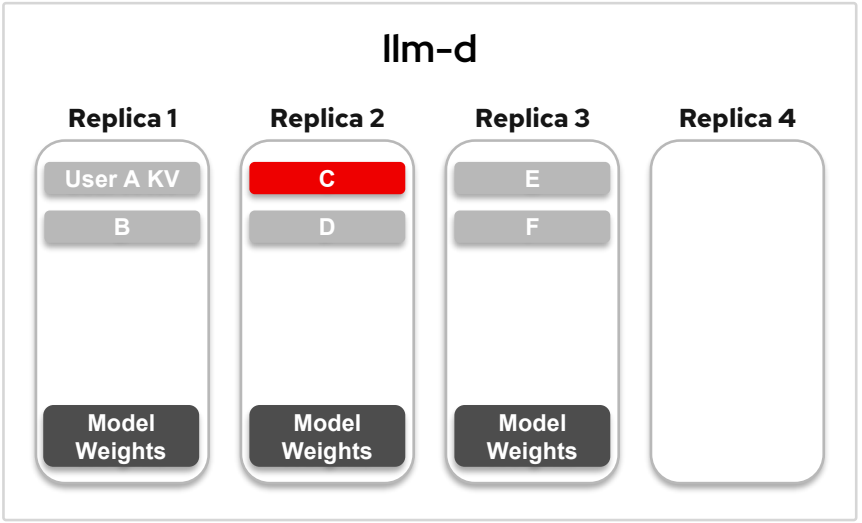
Offload KV caches to global shared remote storage

Example: Impact on KV Cache Re-Use

Leverage all system resources to maximize prefix cache hit rate within the cluster



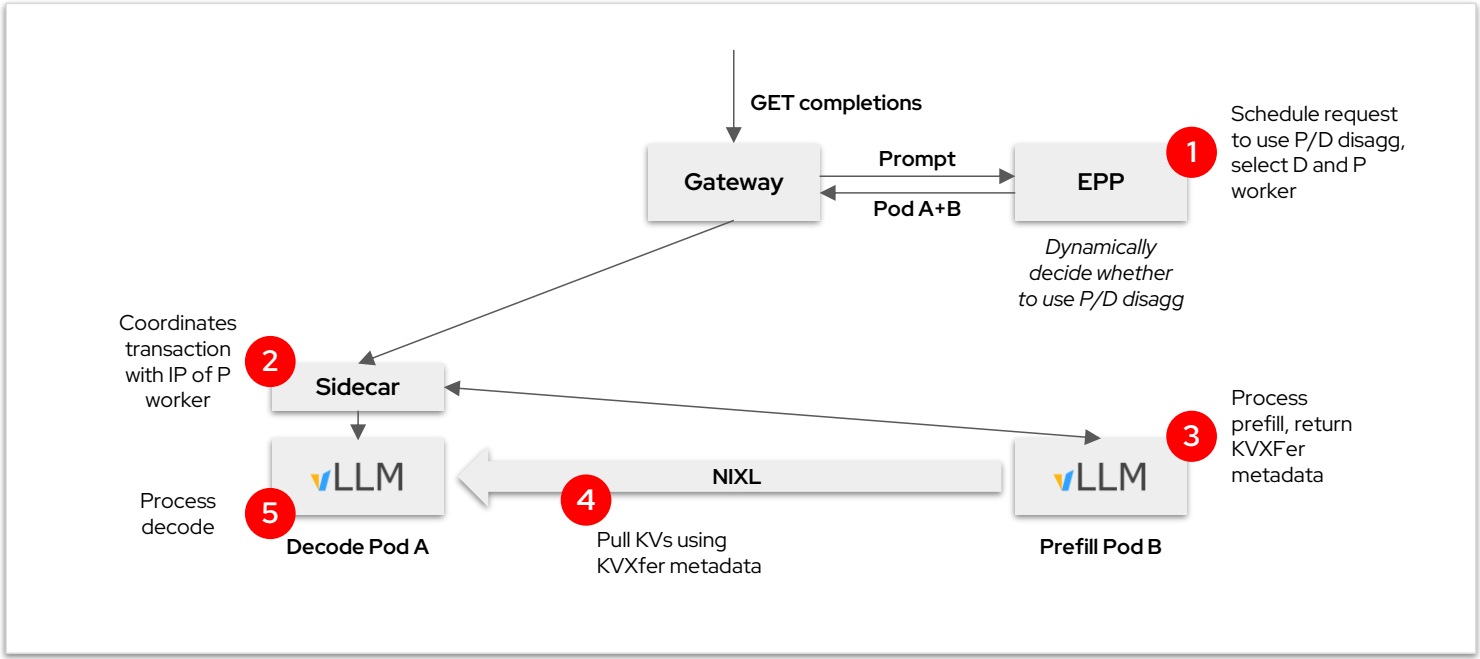
User C's requests are round-robined across all replicas



User C's requests routed to pods with cached tokens, **skipping redundant prefill**

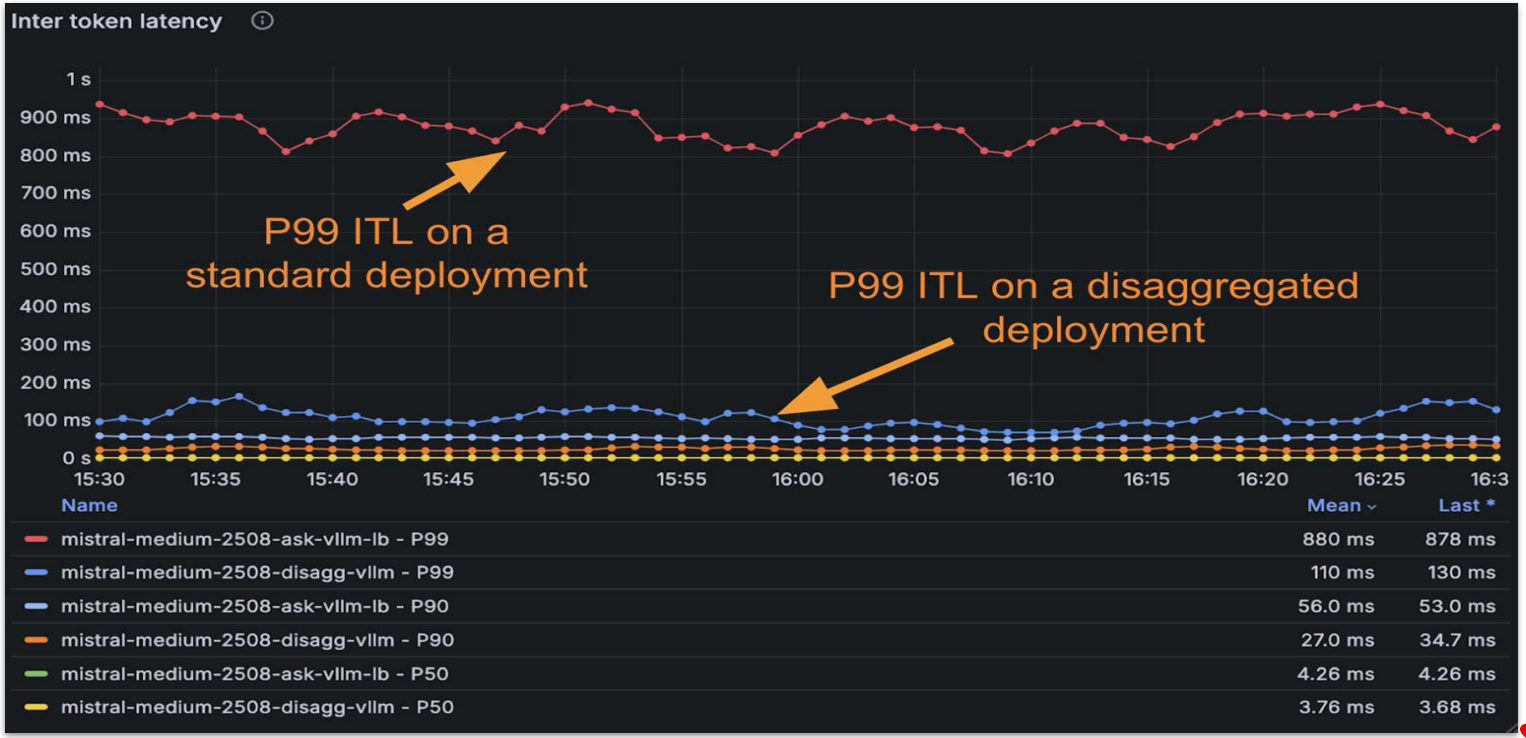
P/D Disaggregation

llm-d composes vLLM and Gateway to enable P/D disaggregation



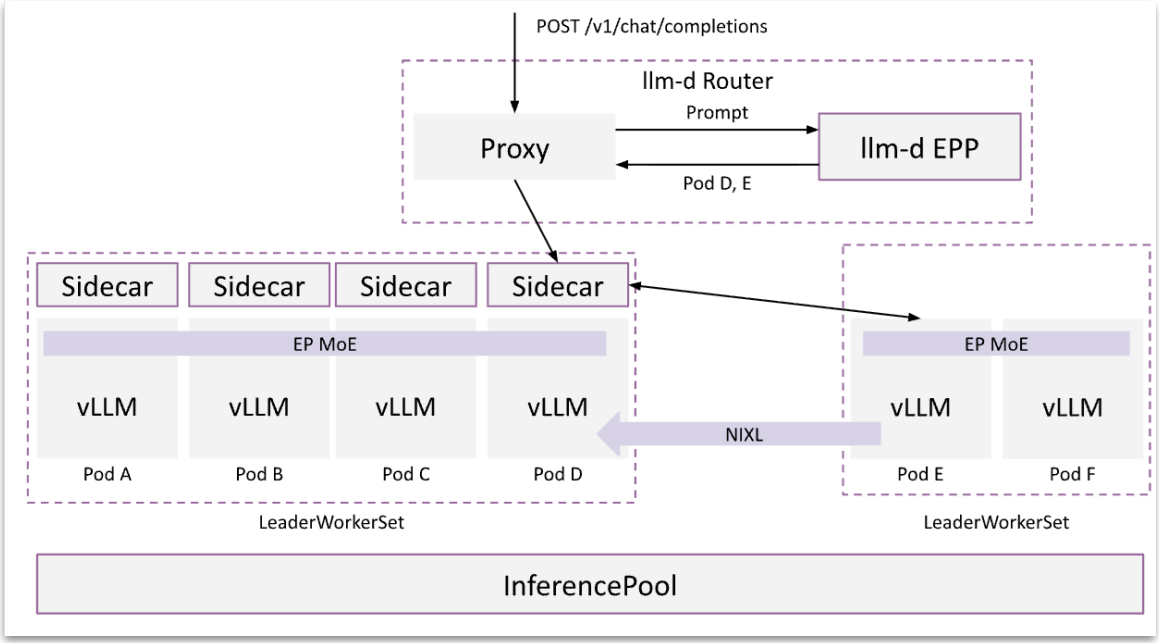
Red Hat

P/D Results - Inter-Token Latency (ITL) smoothing



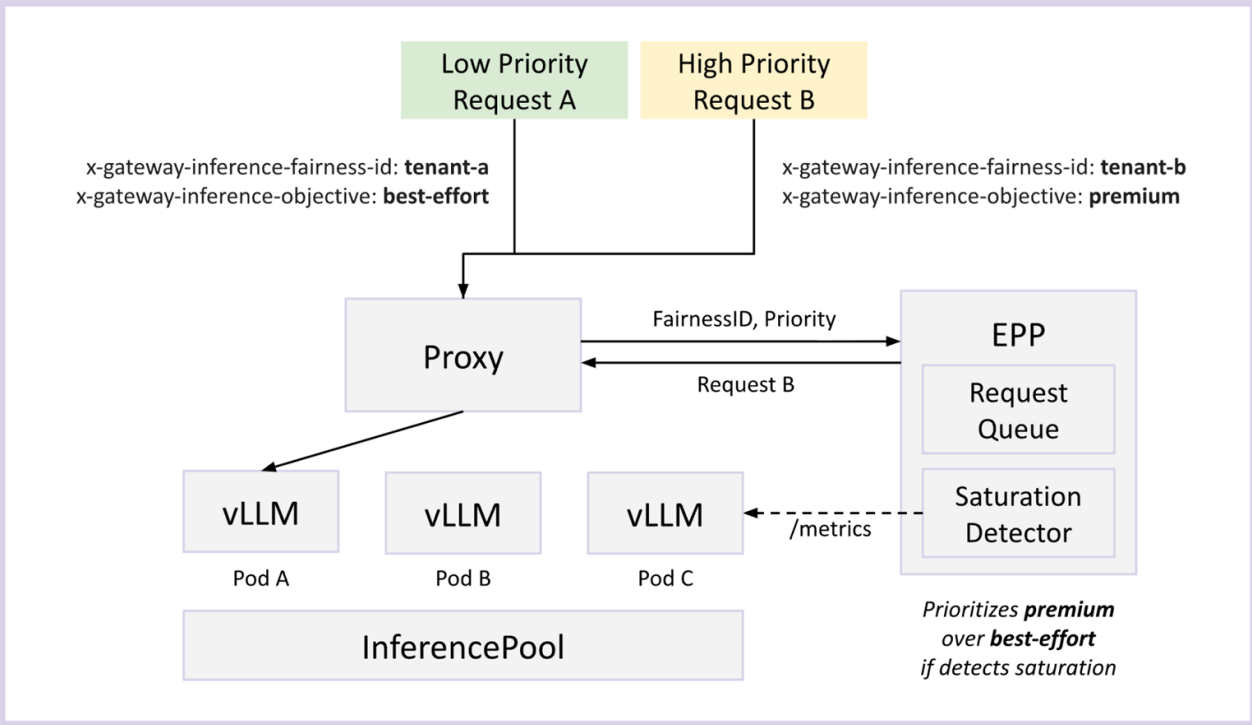
Multi-Node Wide Expert Parallelism

llm-d's K8s-native design composes the DP/EP implementation with the rest of the llm-d system



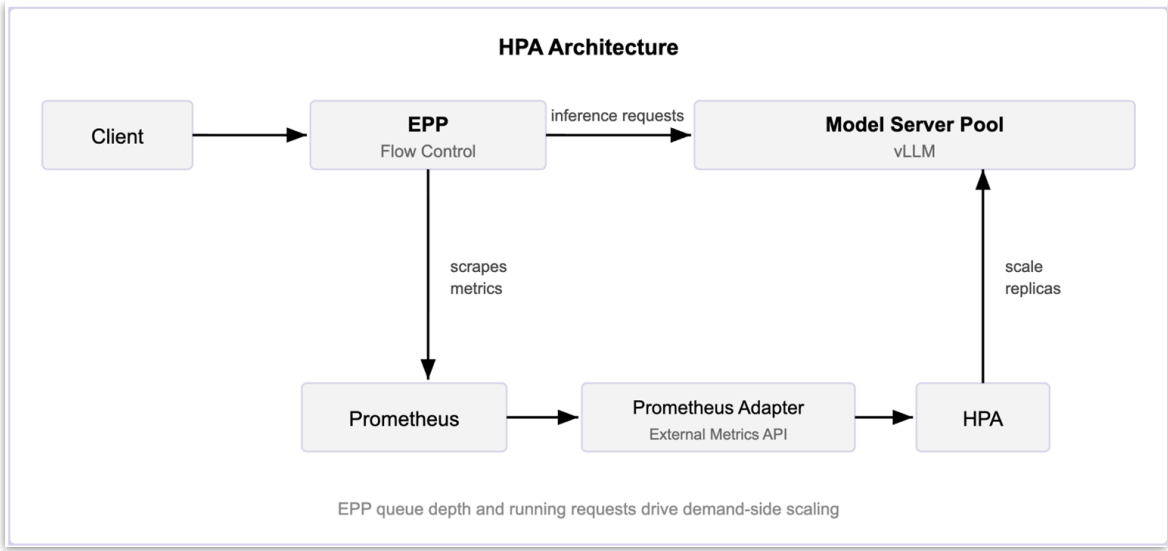
Red Hat

Flow Control for Multi-Tenant Use Cases



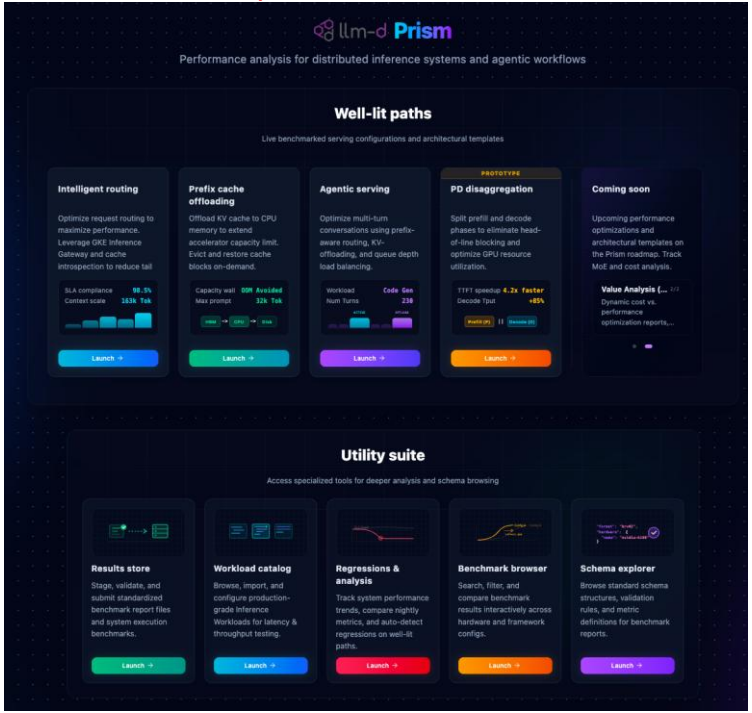
Autoscaling

Flow control multiplexing of different request classes onto the same model deployment



Well Lit Paths on Prism.llm-d.ai

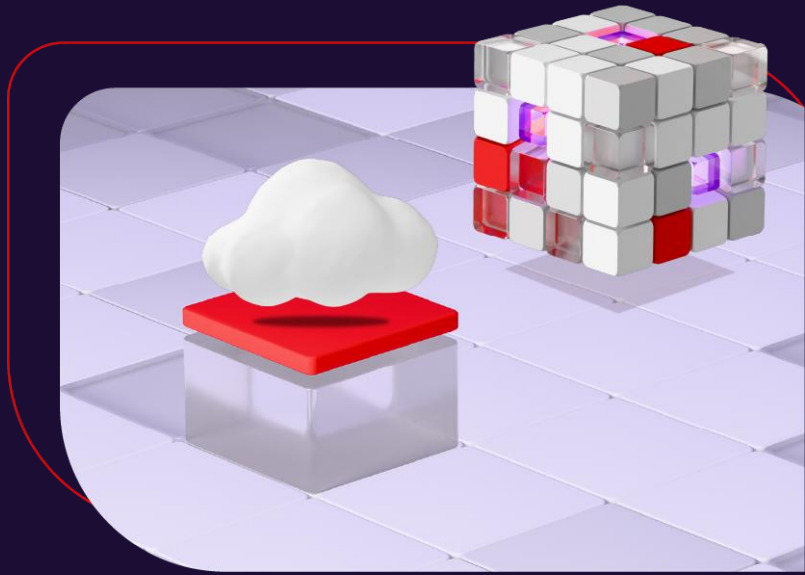
Visualize and compare metrics across benchmarks



https://prism.llm-d.ai/

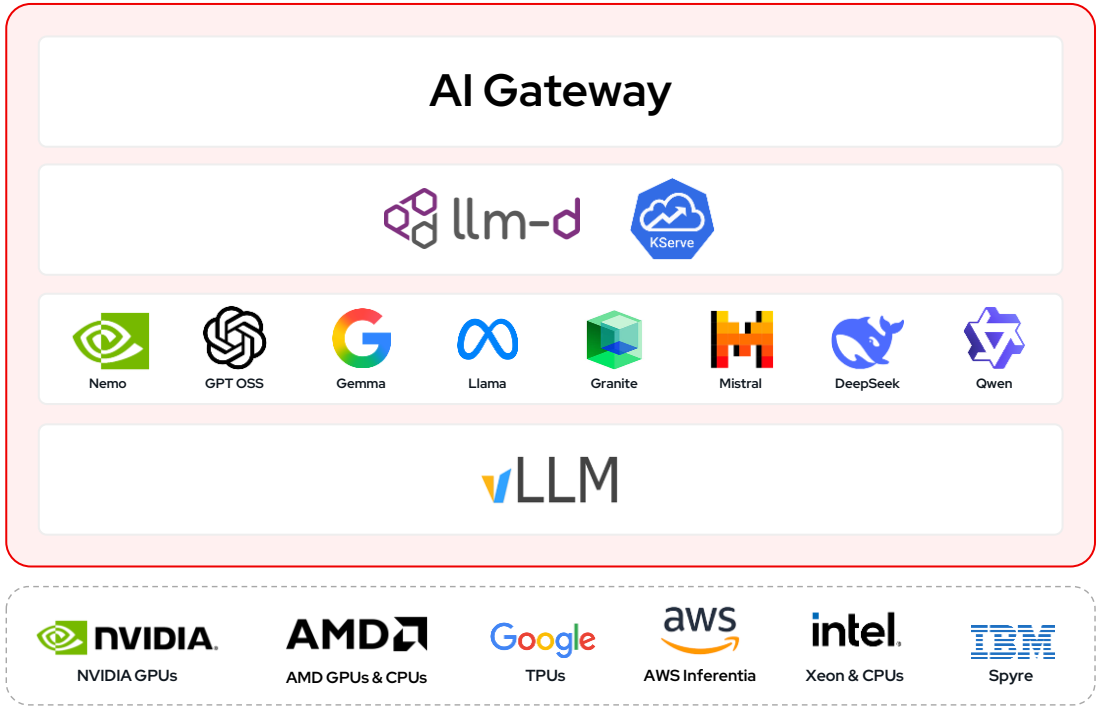
Red Hat

Essential Elements of Enterprise GenAI Stack



Enterprise GenAI Inference Platform

Operationalize Model as a Service across the hybrid cloud



API Management, Token Tracking, Authentication for any model

Distributed Inference

Validated, Optimized Models & Techniques

Inference Server

Flexibility from Red Hat AI Enterprise

Llama · Qwen 通義千問 · DeepSeek 深度求索 · Kimi 月之暗面 · MiniMax 稀宇科技 · Gemma · Mistral · GPT-OSS · Nemotron · Granite

MaaS
Models as a Service
Pre-Optimized Models
AI Gateway

Red Hat AI Enterprise
vLLM · llm-d

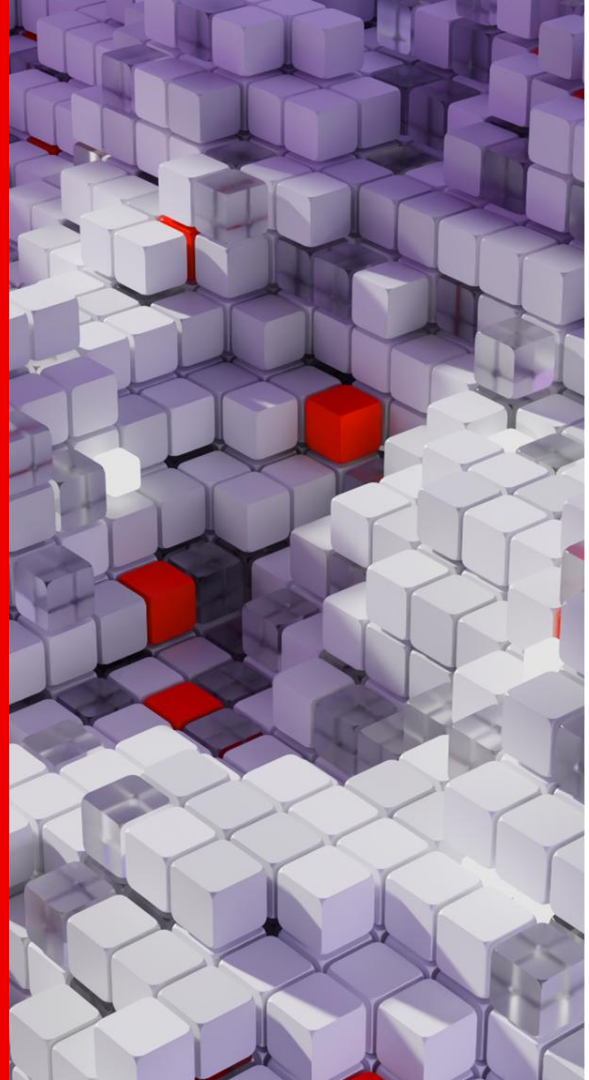
GPUaaS
GPU-as-a-Service
DRA

GPU · Instinct · TPU · Neuron · Gaudi · Spyre · 沐曦卡 (中國) · NPU (Korea)

Physical · Virtual · Private Cloud · Public Cloud · Edge

Single platform to run any model, on any accelerator, on any cloud

Red Hat AI is 100% open source



Thank you!

linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat