Red Hat Summit

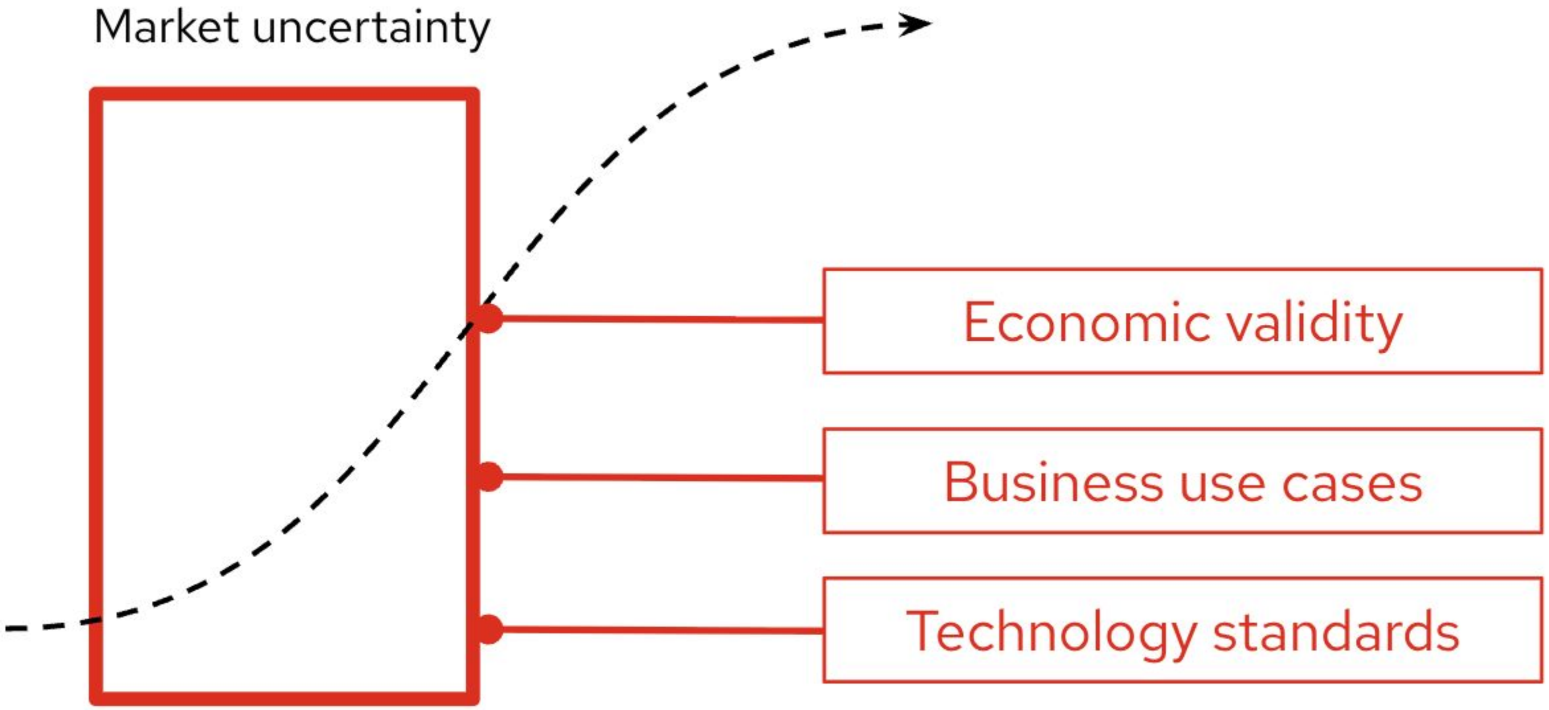
Connect

How Small Models and vLLM Deliver Cost-Effective Scalability

Erica Langhi

Associate Principal Solution Architect - Red Hat

The uncertainty of the AI transition



The challenges of Gen AI adoption

Cost

Generative AI frontier model services are cost prohibitive at scale for most enterprise customer use cases.

Complexity

Tuning models with private enterprise data for customer use cases is too complex for non-data scientists.

Flexibility

Enterprise AI use cases span data center, cloud & edge and can't be constrained to a single public cloud service.

Red Hat AI

Accelerate the development and delivery of AI solutions across hybrid-cloud environments

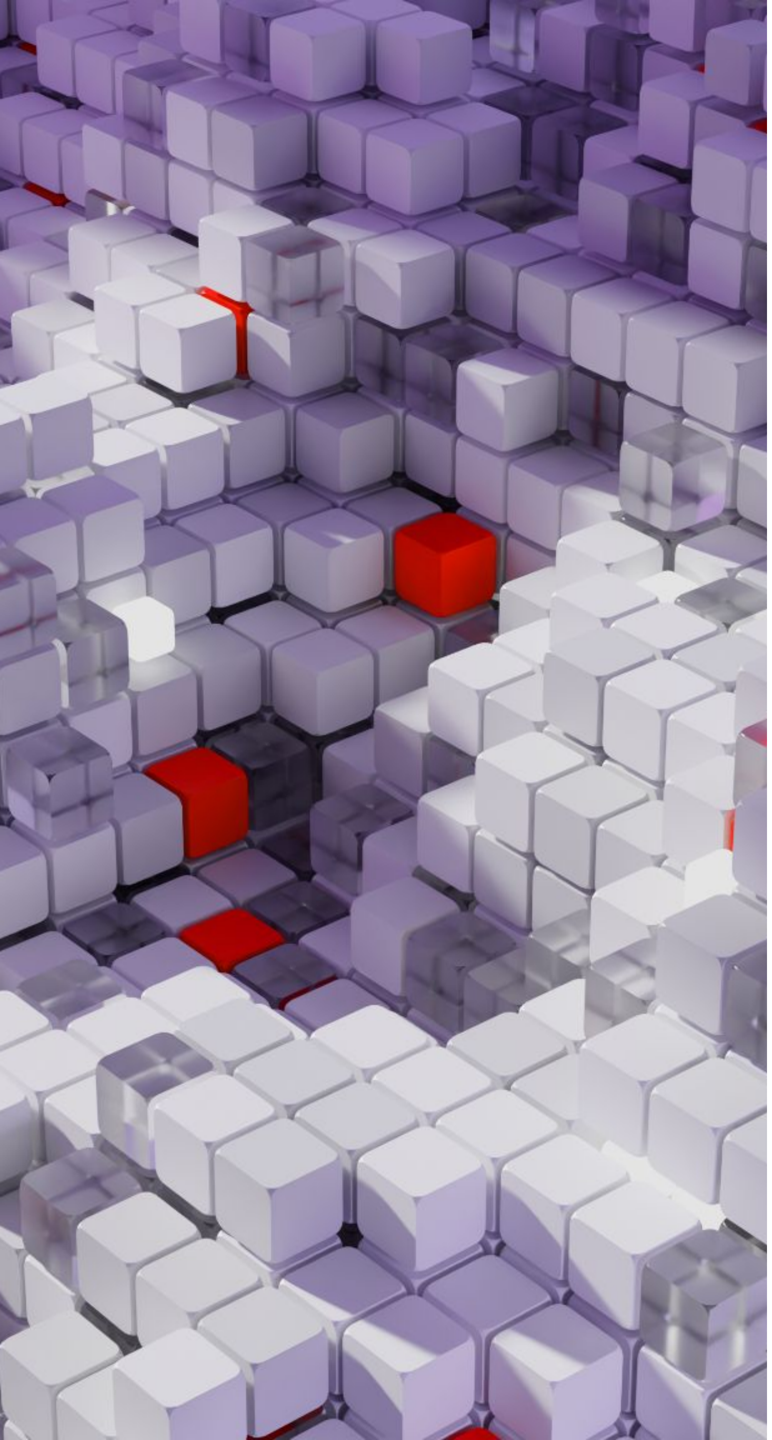
Simplified and consistent experience for **connecting models to data focusing on security**

Increase efficiency with **fast, flexible and efficient inferencing**

Flexibility and consistency when **scaling AI across the hybrid cloud**

Accelerate Agentic AI delivery and stay at the forefront of innovation

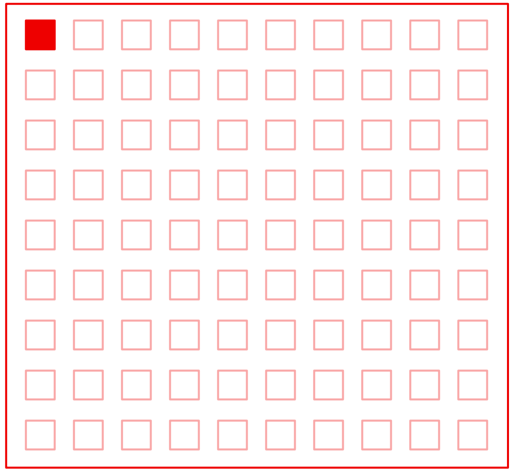
Run any model, on any accelerator, any cloud, on prem and at the edge



Connecting models to data

The value of AI comes from data

LLMs are trained with a range of public data, not enterprise-relevant data

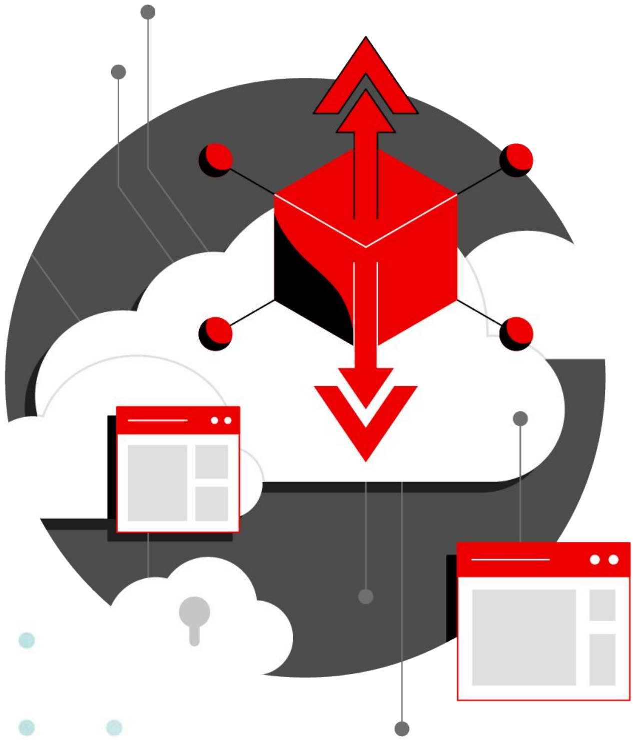


Less than 1% of all enterprise data
is represented in foundation models

Enterprise organizations need to

1. Start from a trusted base model
2. Create a new representation of their data
3. Deploy, scale, and create value with their AI

Choice of Open Source models - large and small



- Red Hat validated models repository on Hugging Face
  - Optimised foundation models including Llama, Mistral, Qwen, Gemma, DeepSeek
  - GuideLLM for benchmarking
- Smaller language models, like IBM Granite, are orders of magnitude smaller than frontier models, **cheaper and faster to run**
- Smaller models can be **tuned and customized with private enterprise data** for domain specific tasks and be used for Agentic AI

Red Hat repository on Hugging Face

A collection of third-party validated and optimized large language models

Broad Collection of models

Llama	Qwen	Gemma
Mistral	DeepSeek	Phi
Molmo	Granite	Nemotron

Validated models

- Tested using realistic scenarios
- Assessed for performance across a range of hardware
- Done using GuideLLM benchmarking and LM Eval Harness

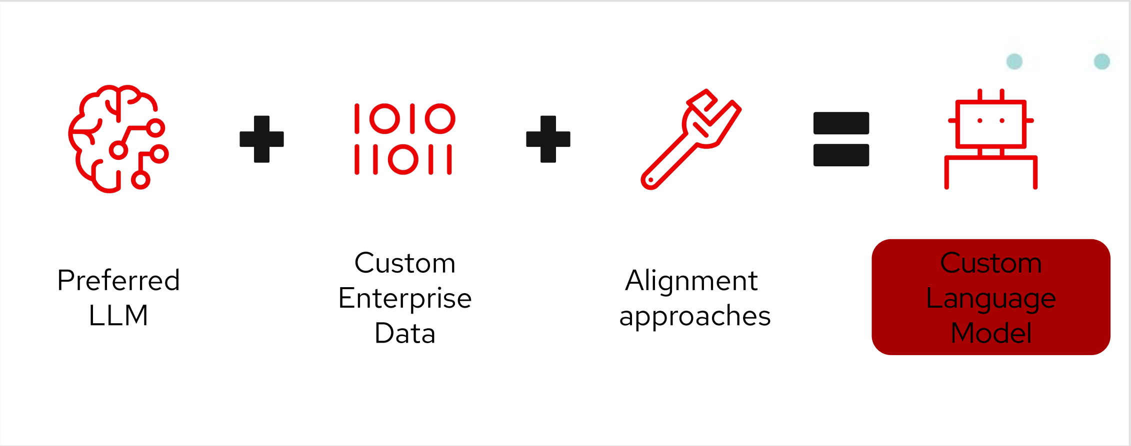
Optimized models

- Compressed for speed and efficiency
- Designed to run faster, use fewer resources, maintain accuracy
- Done using LLM Compressor with latest algorithms

Customize your preferred model using enterprise data to build an efficient, cost-effective solution.

Red Hat AI provides:

- ✓ Validated and optimized models ready-to-use
- ✓ Data ingestion capabilities
- ✓ Synthetic data generation pipelines
- ✓ Multiple alignment techniques



Red Hat AI provides multiple model alignment approaches

Build customized AI solutions that address domain specific business cases

RAG

Retrieval Augmented Generation

Enhance Gen AI model generated text by retrieving relevant information from external sources, improving accuracy and depth of model's responses.

Fine tuning

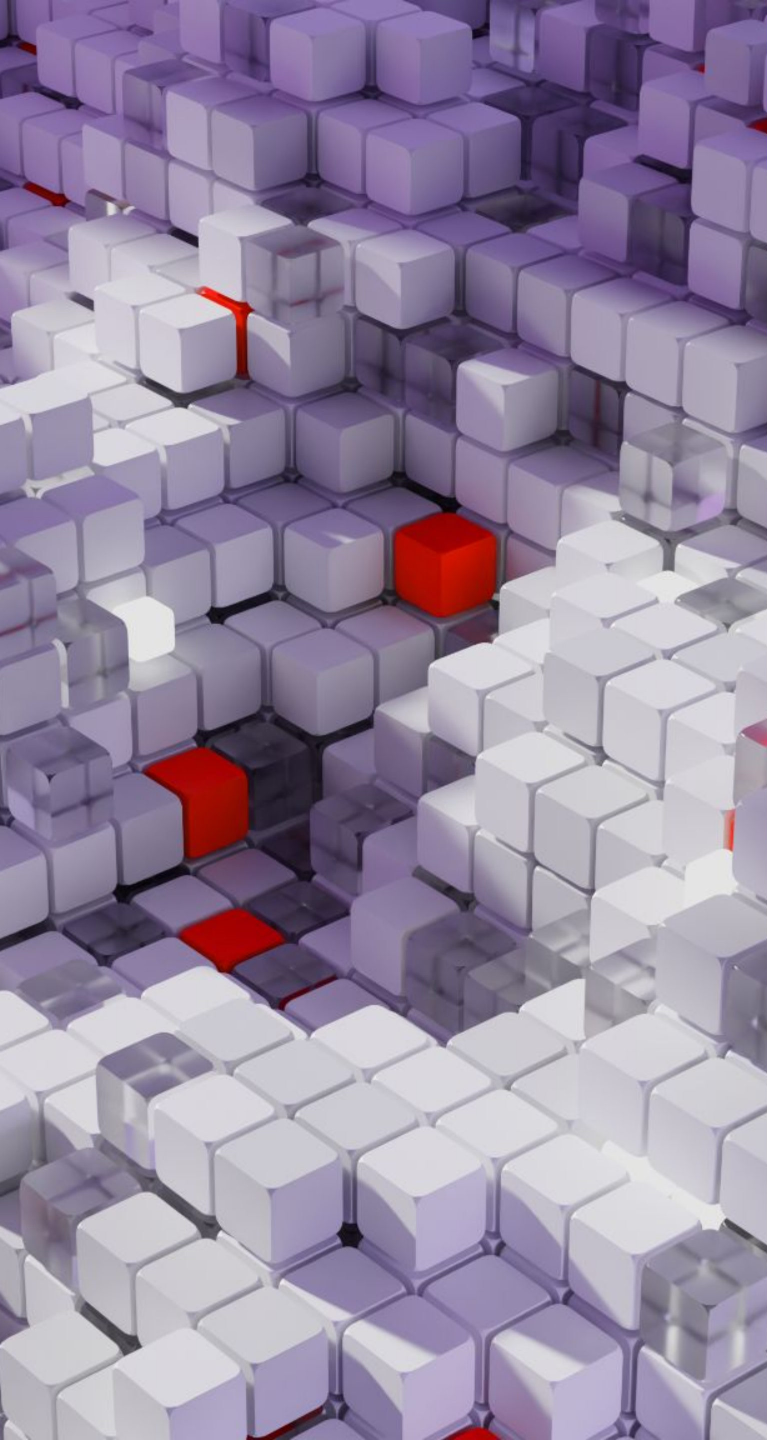
Fine Tuning, LoRa and QLora

Adjust a pre-trained model on specific tasks or data, improving its performance and accuracy for specialized applications without full retraining.

SDG

Synthetic Data Generation

Creates an artificially generated dataset that mimics real data based on provided examples.



Fast, flexible and scalable inference

Inference is where the real world value happens

Need to be fast and accurate in its responses

Manage processing times and token output to control cost

Deliver high throughput and lower latency for best performance

Introducing vLLM : game changer for LLM inference

Open Source Library

Designed for lightning-fast LLM inference and serving.Developed by: UC Berkeley's Large-Scale AI Lab

Optimised usage & Self-service

Maximize throughput and minimize latency for LLM serving.

Key Innovation

Addresses the challenges of inefficient GPU utilization during LLM inference.

Key features for vLLM

- **PagedAttention** breaks the KV cache into "blocks" and allocates them dynamically as need
- **Continuous Batching** dynamically processes requests as they arrive to maximize GPU utilization
- **Significantly higher throughput** up to 24x in some benchmarks
- **Broad Model Compatibility** seamlessly integrates with a wide range of popular open-source LLMs
- **Multi-GPU and Multi-Host Support** distributing workloads across multiple GPUs and across multiple machines in a cluster

Red Hat AI Inference Server

vLLM connects model creators to accelerated hardware providers

Llama | Qwen | DeepSeek | Gemma | Mistral | Molmo | Phi | Nemotron | Granite

Red Hat AI
Inference Server

vLLM

NVIDIA GPU | AMD Instinct | Google TPU | aws Neuron | intel Gaudi | IBM Spyre

Physical | **Virtual** | **Private Cloud** | **Public Cloud** | **Edge**

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

Components of an AI Platform

Model development

Access to frontier models

Tuning capabilities

Serving mechanisms

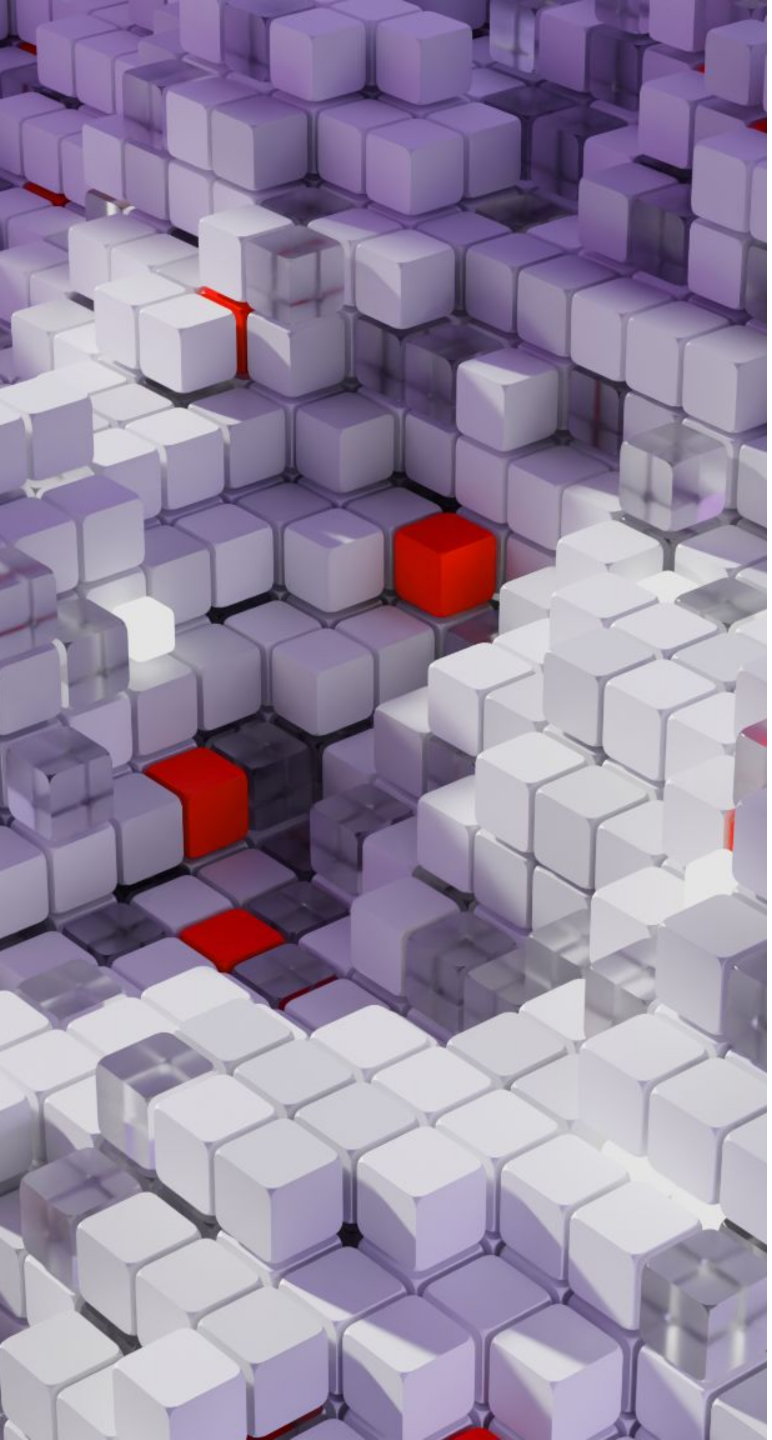
Hybrid-cloud support

Model Lifecycle management

Monitoring

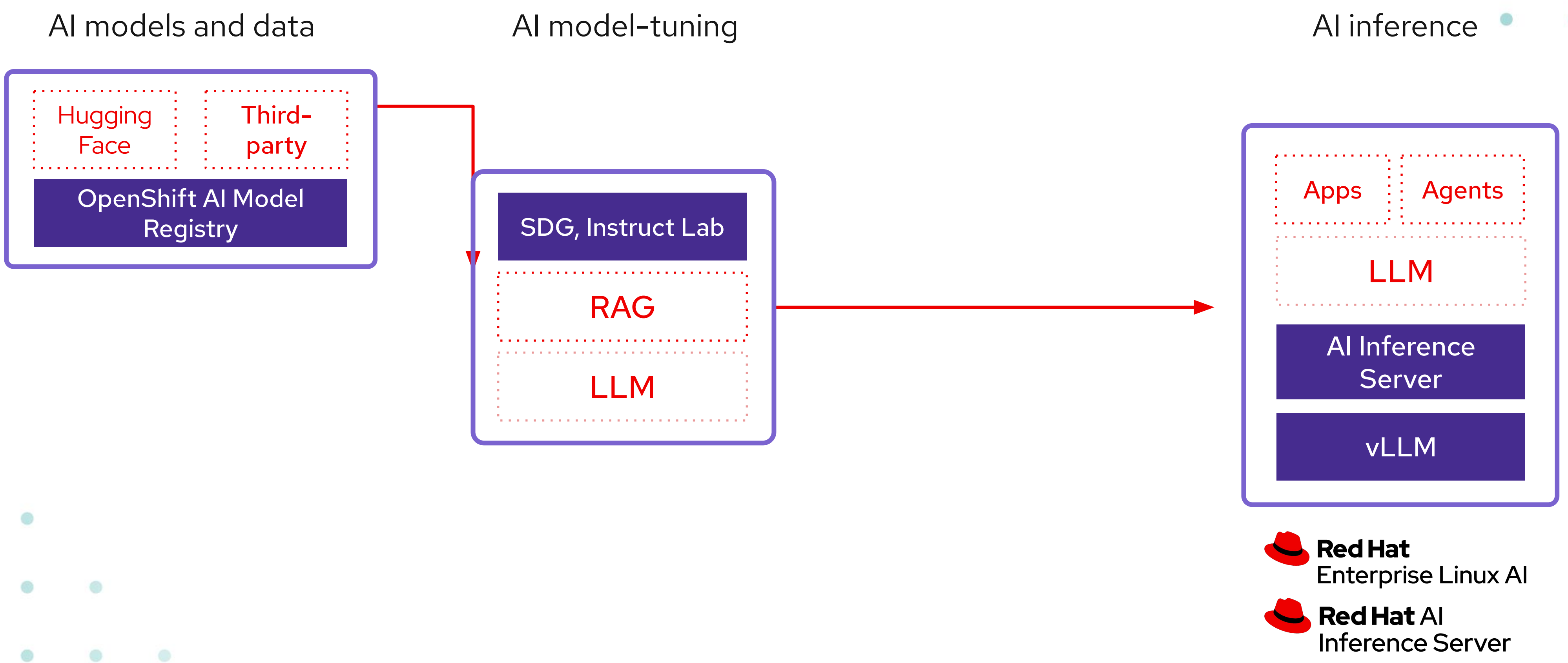
Scalability and automation

Resource optimization and management

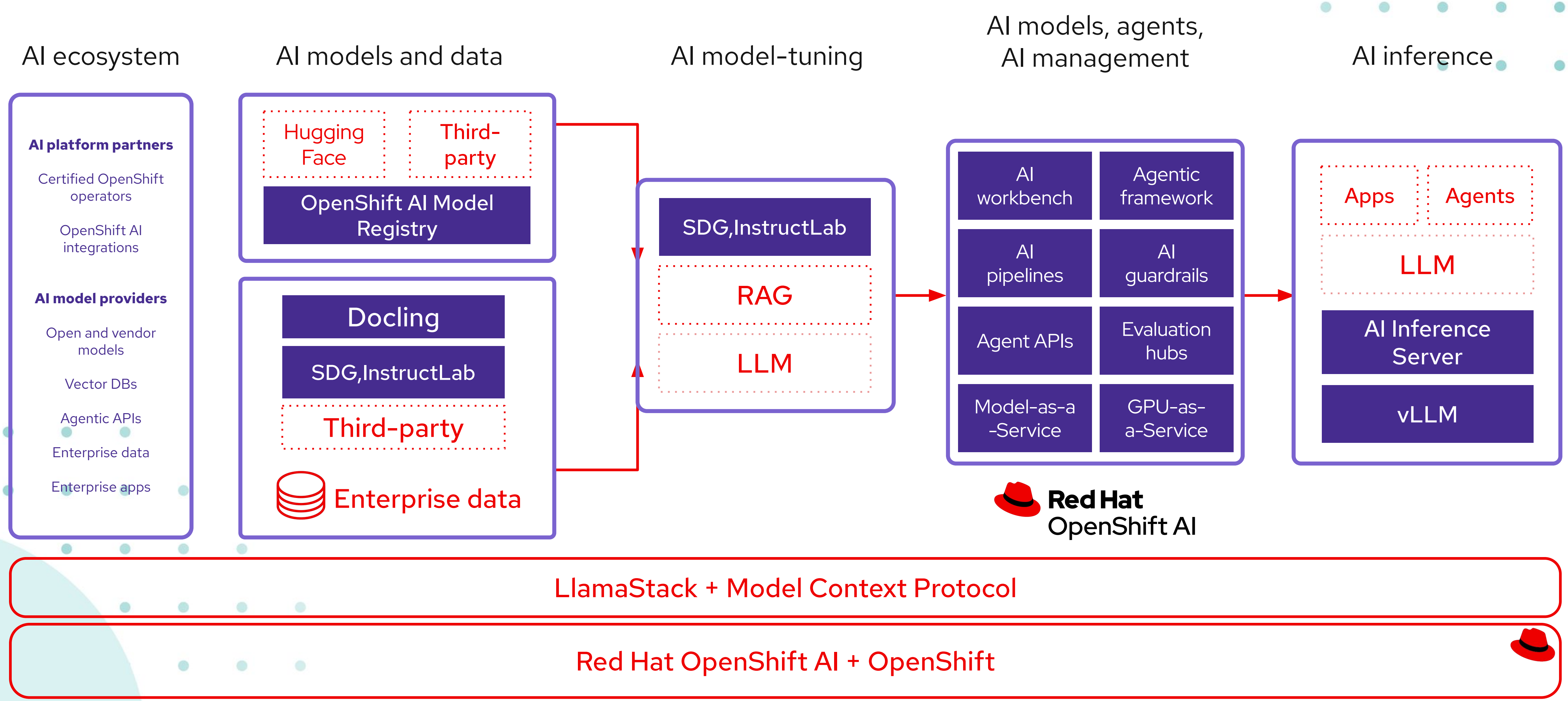


Scaling AI across the hybrid cloud with agentic AI

AI : Get started quickly



AI : Open, Agentic, Enterprise ready



Red Hat Summit

Connect

Thank you

linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Case study : optimisation

We work hand-in-hand with our customers to apply the leading compression research with our llm-compressor framework

DBMS Company: Maximum Compression

- L70B for SQL, deployed to customer with 8 GPUs, with a goal to maximize compression
- Struggled to quantize the model due to poor accuracy and issues with open-source tools
- Customer and engineering team worked together to apply W4A16 quantization their model using llm-compressor
  - Recovered accuracy to >99% of baseline

Reduced GPUs needed for deployment from 8->2

Retail Company: Maximum Throughput

- Fine-tuned Llama-70B models for JSON extract, runs on millions of records per day (H100)
- Saw no benefit from quantization, due to usage of weight-only methods for throughput use case
- Customer and engineering team worked together to apply quantization to their model
  - Select right optimization for their workload
  - Tune hyperparameters for high accuracy

Realized a 40% reduction in GPU hours (data)