Accelerate AI agents with Multimodal RAG using Friendli Endpoints and Milvus

Zilliz Webinar

Feb 2025

Soomin Chun

About me

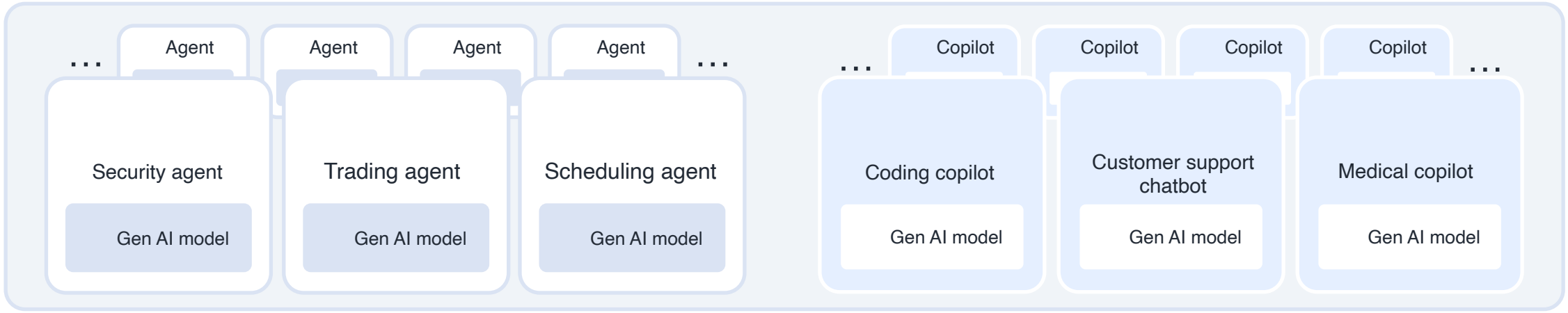
- Software Engineer at FriendliAI, leading provider of generative AI infrastructure
- CS at MIT
- Previously at ed-tech start-up, Meta, quantitative finance

Our mission is to empower organizations to harness the power of generative AI with ease and cost-efficiency.

Agenda

1. Background: LLMs, LLM inference serving, and AI agents
2. Challenges in building AI agents
3. Advantages of FriendliAI and Milvus
4. Live demo – Colab notebook walkthrough

Increasing use of open-source generative AI models



Many companies are embracing (custom) open-source generative AI models.

High quality open-source gen AI models

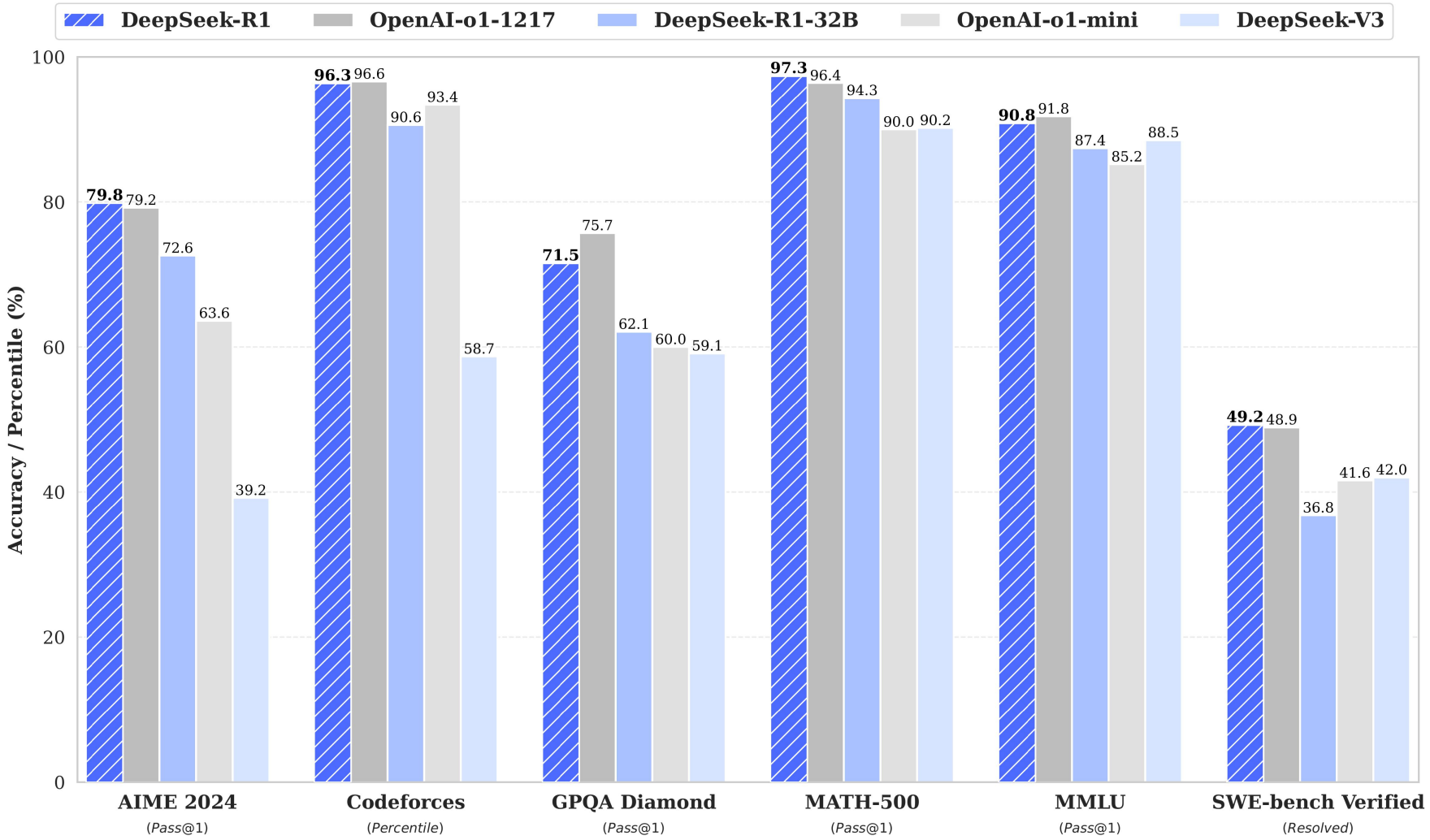
(Meta Llama models get 350 million downloads (2024.8)[1])

Cohere Command R+ · Google Gemma · MistralAI Mixtral · Snowflake Arctic · Meta Llama · Microsoft Phi · Databricks DBRX · deepseek

1. https://www.computerworld.com/article/3499062/metas-llama-models-get-350-million-downloads.html

DeepSeek-R1

Test-time compute



Ref. https://github.com/deepseek-ai/DeepSeek-R1

What is a Large Language Model (LLM)?

Hugging Face | Search models, datasets, users...

meta-llama / **Llama-3.1-70B-Instruct** | like 781 | Follow

Text Generation | Transformers | Safetensors | PyTorch | 8 languages

Inference Endpoints | arxiv:2204.05149 | License: llama3.1

Model card | Files and versions | Community 43

A newer version of this model is available: `meta-llama/Llama-3.3-70B-Instruct`

Gated model You have been granted access to this model

Model Information

The Meta Llama 3.1 collection of multilingual large language models (LLMs) is a collection of pretrained and instruction tuned generative models in 8B, 70B and 405B sizes (text in/text out). The Llama 3.1 instruction tuned text only models (8B, 70B, 405B) are optimized for multilingual dialogue use cases and outperform many of the available open source and closed chat models on common industry benchmarks.

Model developer: Meta

Model Architecture: Llama 3.1 is an auto-regressive language model that uses an optimized transformer architecture. The tuned versions use supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF) to align with human preferences for helpfulness and safety.

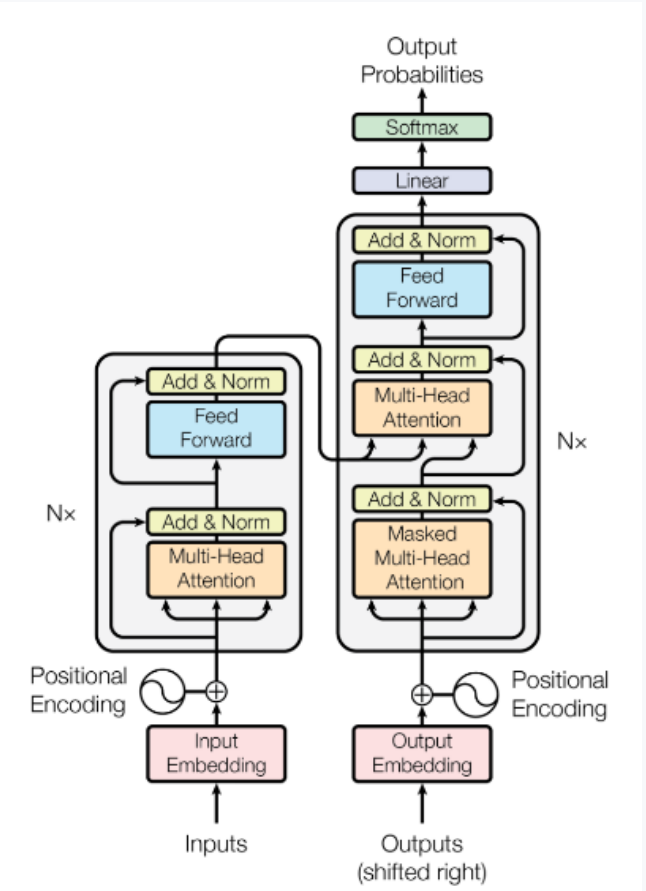
Training Data	Params	Input modalities	Output modalities	Context length	GQA	Token count	Knowled cutoff

File

model.safetensors.index.json 59.6 kB

1/31 | Download | View all files

Tensors	Shape	Precision
model		
model.embed_tokens.weight	[128 256, 8 192]	BF16
model.layers.0		
model.layers.0.input_layernorm.weight	[8 192]	BF16
model.layers.0.mlp.down_proj.weight	[8 192, 28 672]	BF16
model.layers.0.mlp.gate_proj.weight	[28 672, 8 192]	BF16
model.layers.0.mlp.up_proj.weight	[28 672, 8 192]	BF16
model.layers.0.post_attention_layernorm.weight	[8 192]	BF16
model.layers.0.self_attn.k_proj.weight	[1 024, 8 192]	BF16
model.layers.0.self_attn.o_proj.weight	[8 192, 8 192]	BF16
model.layers.0.self_attn.q_proj.weight	[8 192, 8 192]	BF16
model.layers.0.self_attn.v_proj.weight	[1 024, 8 192]	BF16
model.layers.1		
model.layers.1.input_layernorm.weight	[8 192]	BF16
model.layers.1.mlp.down_proj.weight	[8 192, 28 672]	BF16
model.layers.1.mlp.gate_proj.weight	[28 672, 8 192]	BF16
model.layers.1.mlp.up_proj.weight	[28 672, 8 192]	BF16
model.layers.1.post_attention_layernorm.weight	[8 192]	BF16
model.layers.1.self_attn.k_proj.weight	[1 024, 8 192]	BF16
model.layers.1.self_attn.o_proj.weight	[8 192, 8 192]	BF16
model.layers.1.self_attn.q_proj.weight	[8 192, 8 192]	BF16
model.layers.1.self_attn.v_proj.weight	[1 024, 8 192]	BF16
model.layers.2		
model.layers.2.input_layernorm.weight	[8 192]	BF16
model.layers.2.mlp.down_proj.weight	[8 192, 28 672]	BF16
model.layers.2.mlp.gate_proj.weight	[28 672, 8 192]	BF16
model.layers.2.mlp.up_proj.weight	[28 672, 8 192]	BF16
model.layers.2.post_attention_layernorm.weight	[8 192]	BF16
model.layers.2.self_attn.k_proj.weight	[1 024, 8 192]	BF16
model.layers.2.self_attn.o_proj.weight	[8 192, 8 192]	BF16
model.layers.2.self_attn.q_proj.weight	[8 192, 8 192]	BF16
model.layers.2.self_attn.v_proj.weight	[1 024, 8 192]	BF16



Ref. Attention is All you Need

Evolution of generative AI

Vision language models

Describe this image.

This image shows a blue llama with the words “Meta Llama 3.2 Vision”.

Ref. Medium article: Llama 3.2 Vision: Mutimodal AI with Image Reasoning

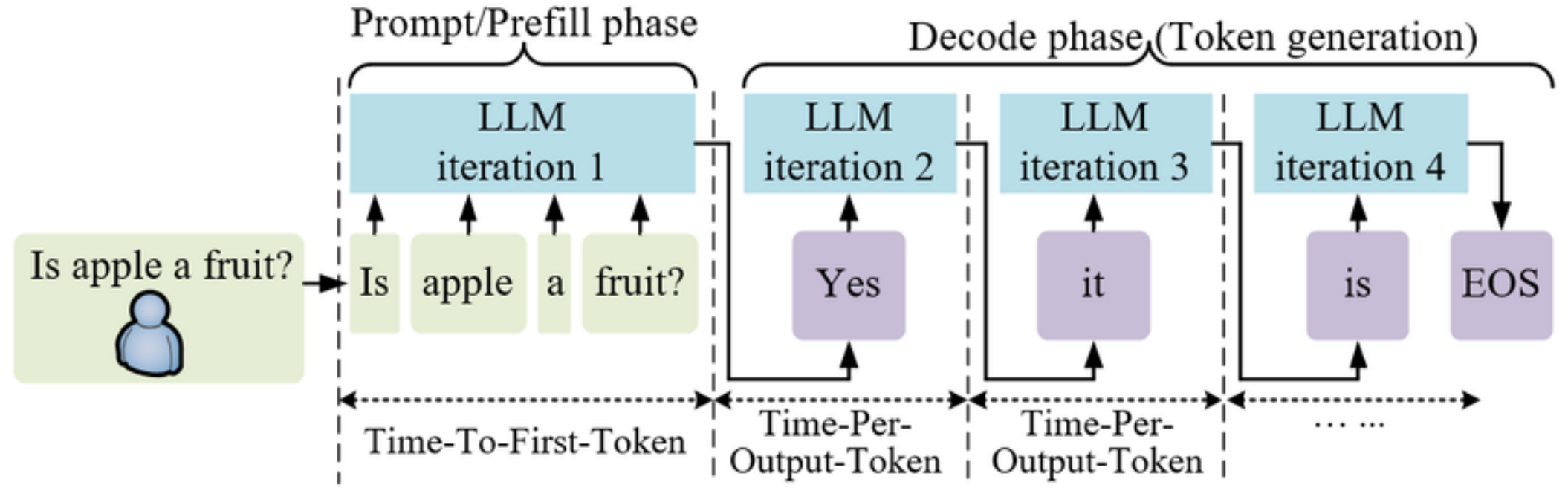
Image generation models

Generate an image of a purple cow in a digital art style.

Ref. Zapier blog: How to use Stable Diffusion

What is generative AI inference serving?

LLM inference

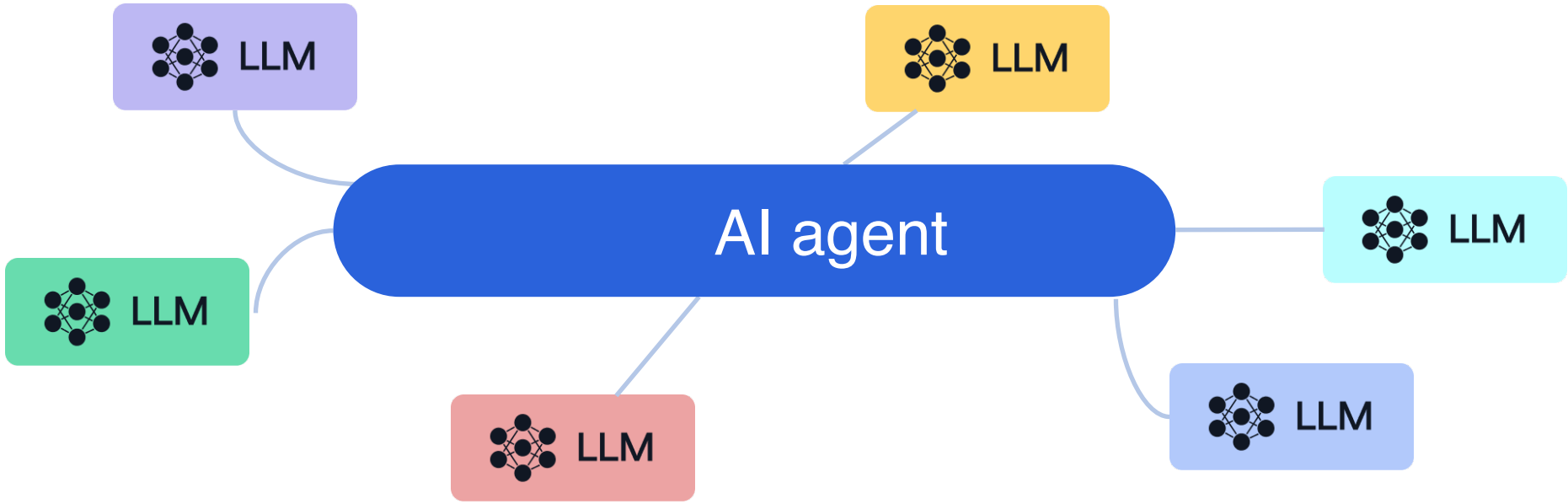


This process is very computationally intensive. For efficient inference, we need AI Hardware (GPUs), which can be very expensive.

Ref. Generative AI as a Service in 6G Edge-Cloud

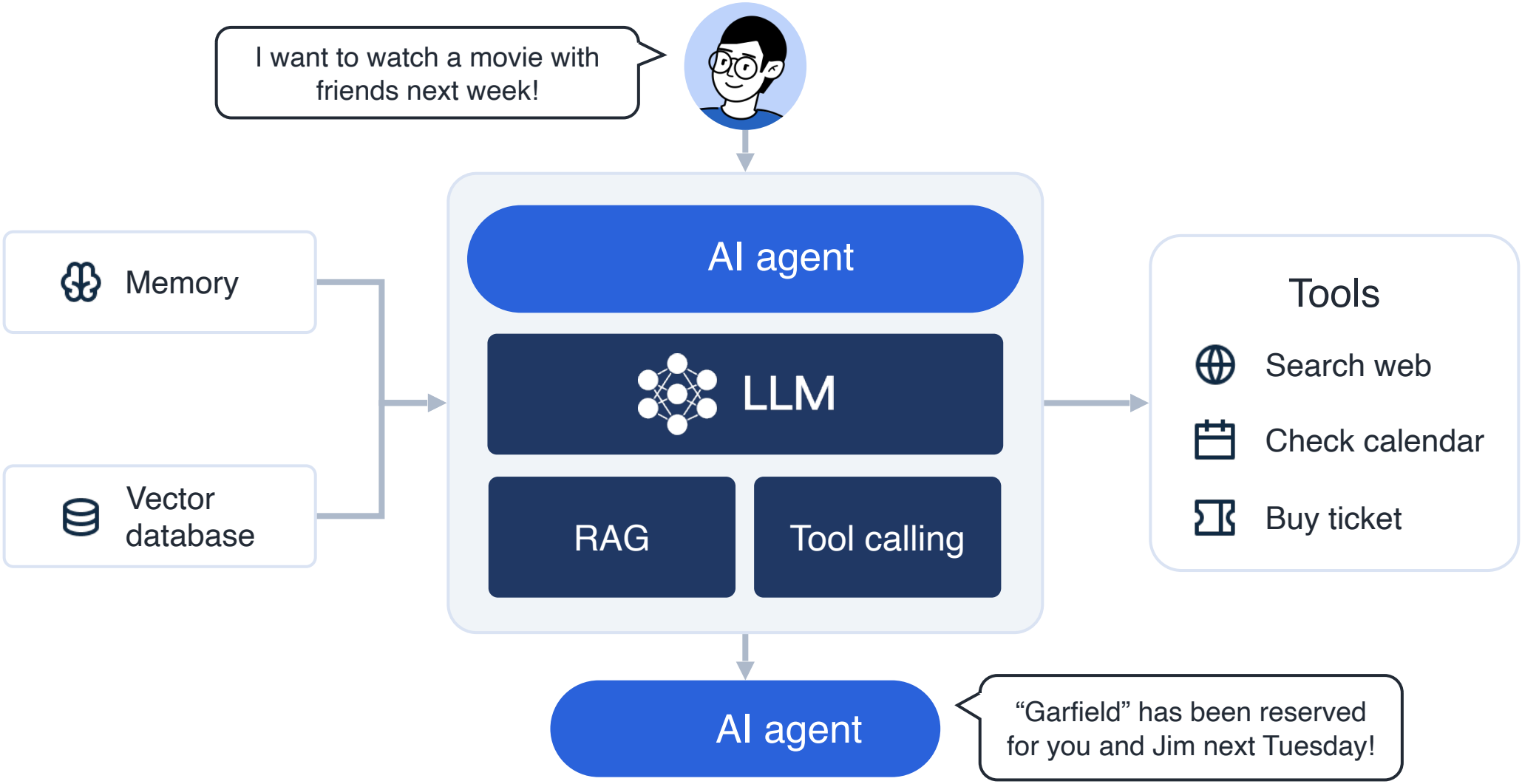
What is an AI agent?

An AI agent is a software system that autonomously interacts with its environment to achieve goals, often powered by large language models (LLMs).



Agentic AI

Many more LLM calls



Challenges in building/running AI agents

AI agent

Goals

Responses should be:

- High quality
- Fast
- Affordable

Hurdles

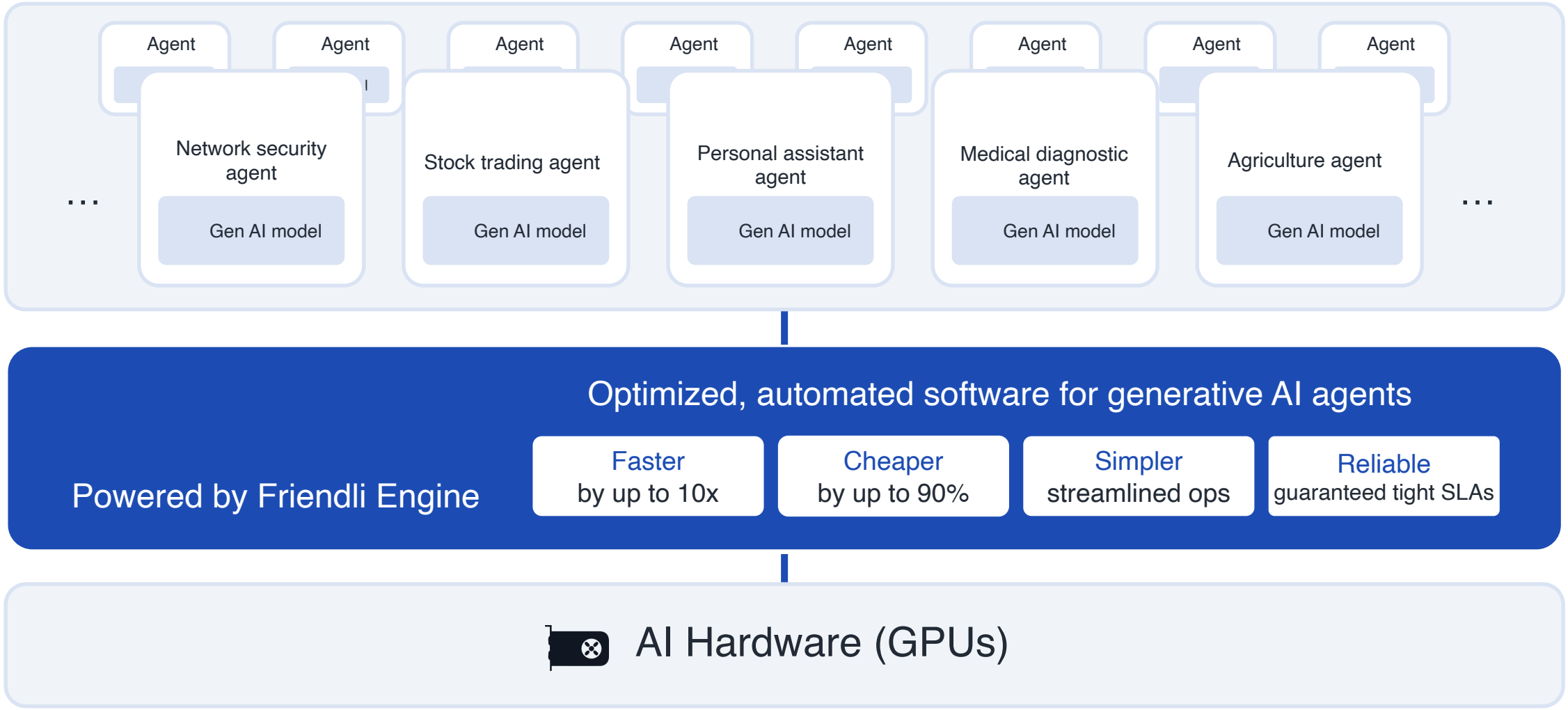
1. Building

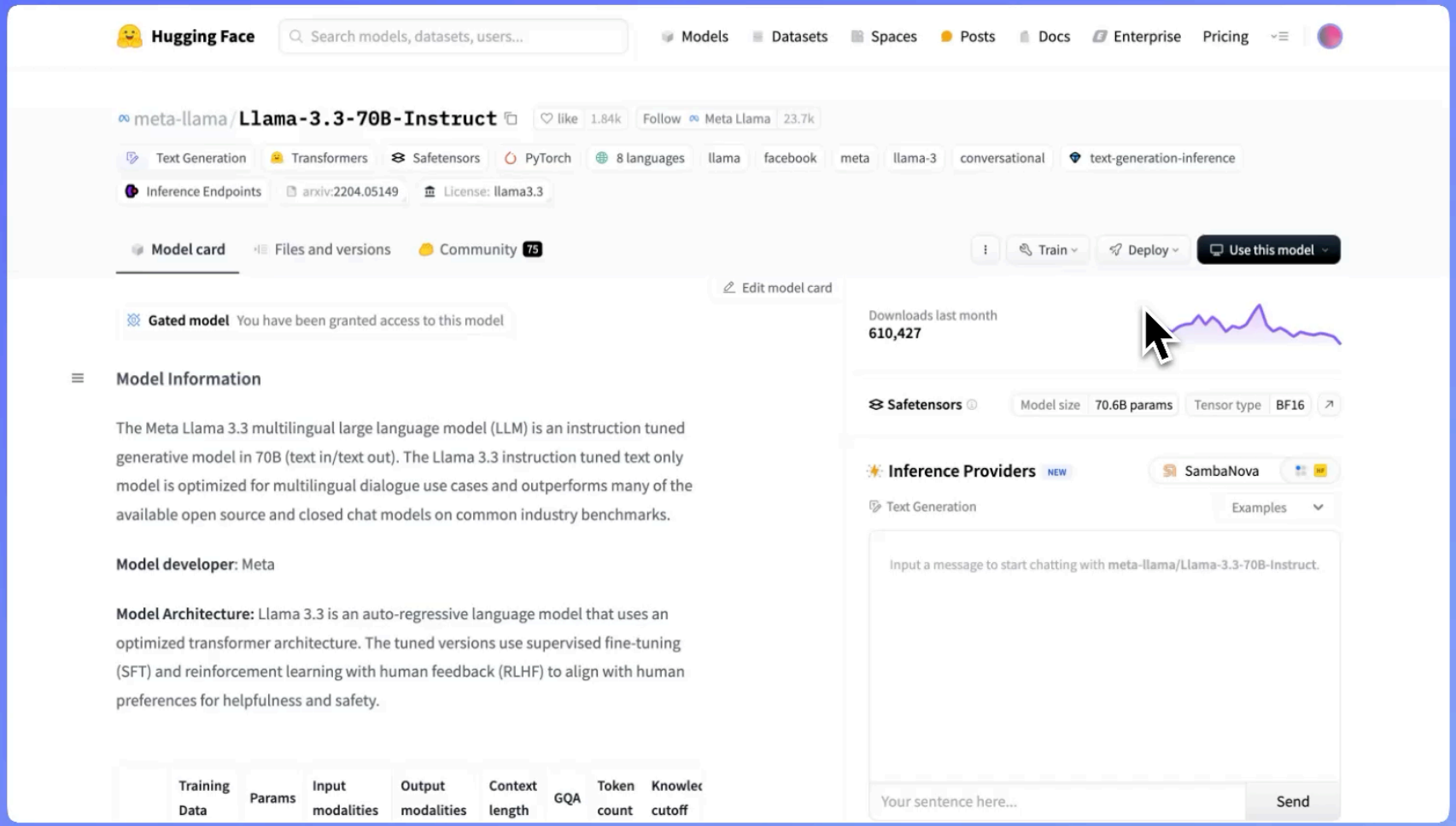
Complicated, with many parts

2. Running

LLMs can be very costly

FriendliAI: Infrastructure for running generative AI

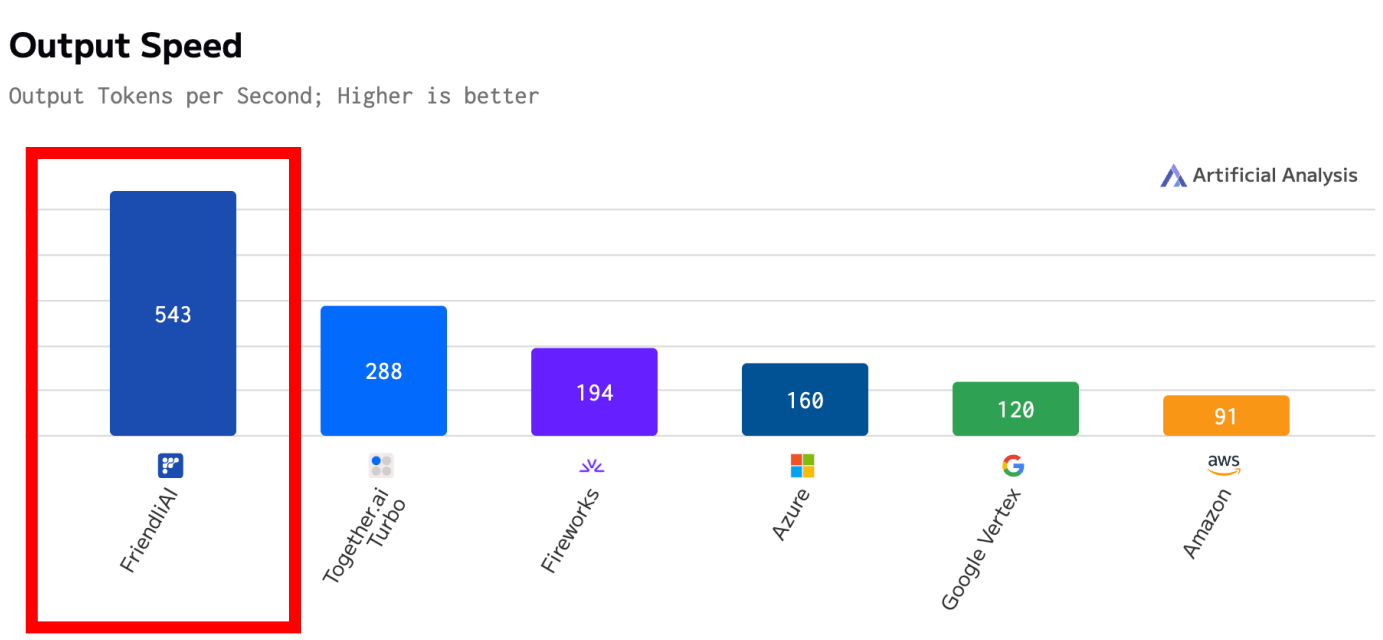




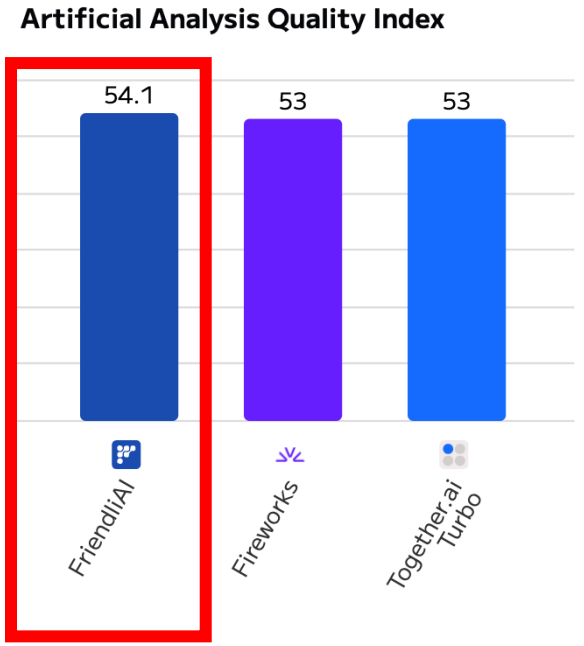
Why use FriendliAI for inference?
The world's No.1 GPU-based API provider

Llama 3.1 8B **(Artificial Analysis https://artificialanalysis.ai/)**

A single request



High speed



High quality

High throughput of Friendli: reduction in required GPUs

Model

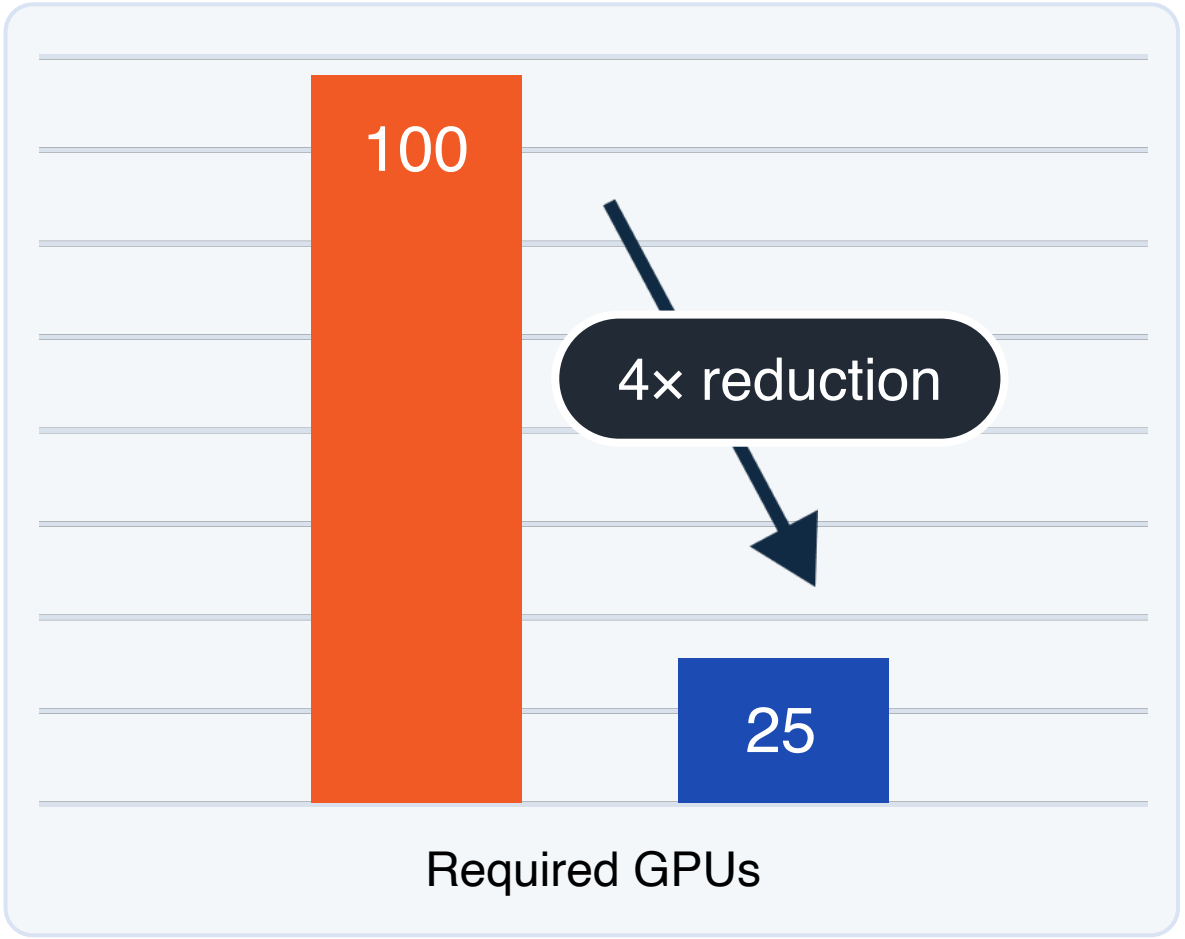
Llama 3.1 8B

GPU

NVIDIA H100 80GB GPU

Concurrent Users (* 1 rpm per user)

75,000



vLLM FriendliAI

Why use Milvus?

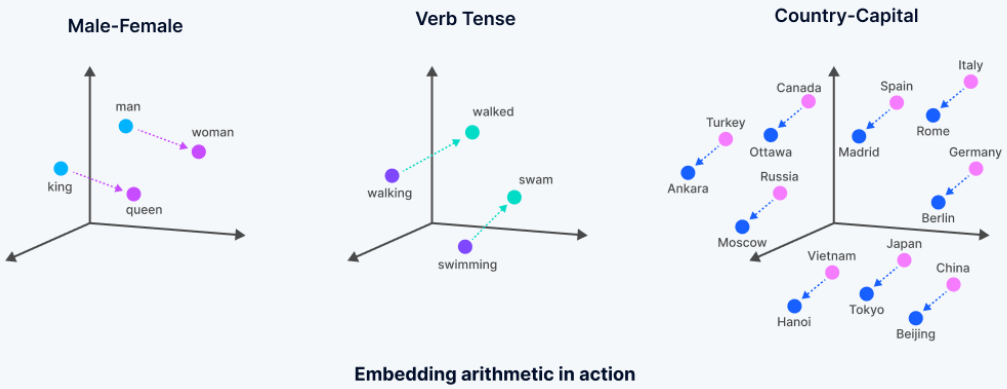
Vector database for generative AI

```
Query: "What is the capital of France?"
```

Milvus

Vector Database

Vector similarity search

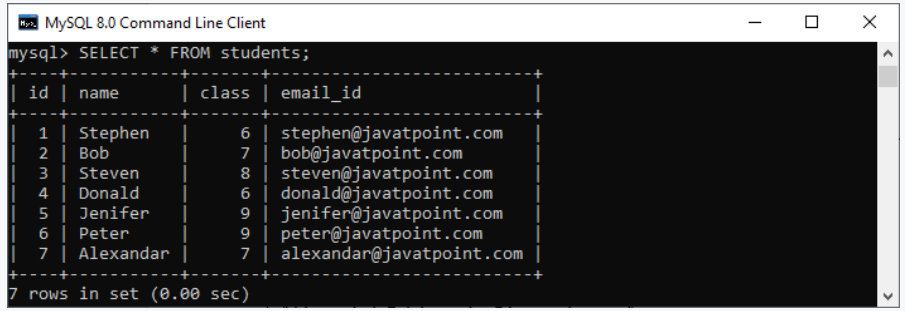


Embedding arithmetic in action

Ref. Milvus

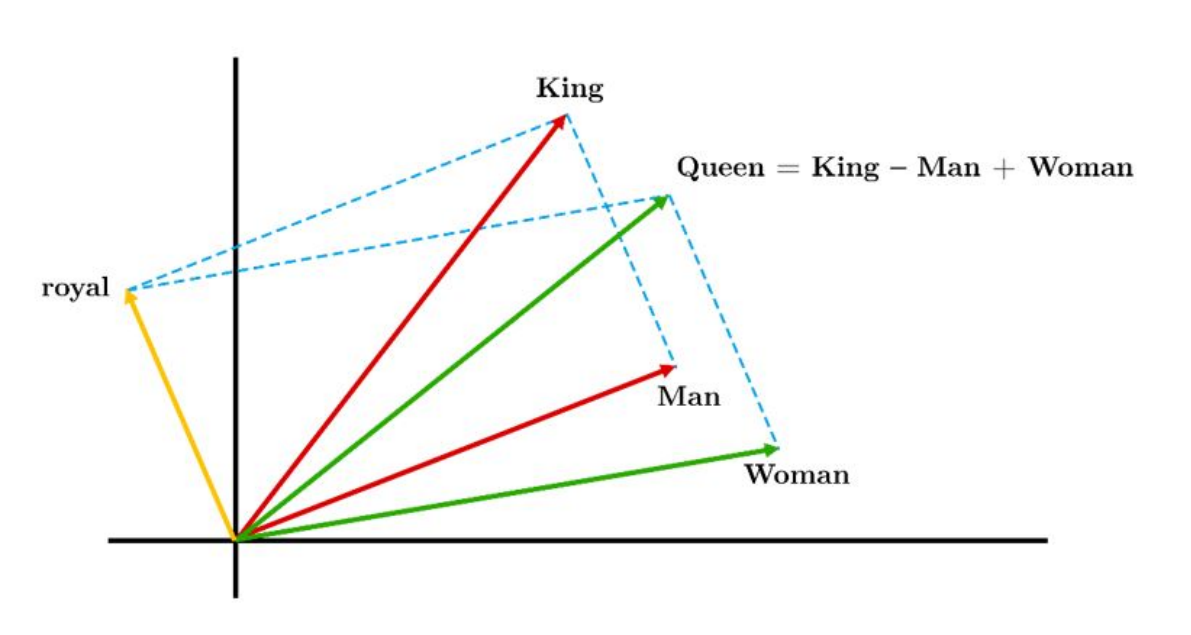
MySQL

Looks for exact matches of “capital” and “France” in a structured table.



Quick overview of Retrieval Augmented Generation (RAG)

Vector representations capture semantic similarity



https://lamarr-institute.org/blog/dedicom-matrix-factorization/

RA

G

Milvus

Company knowledge base

Query

+

Retrieved information

Demo

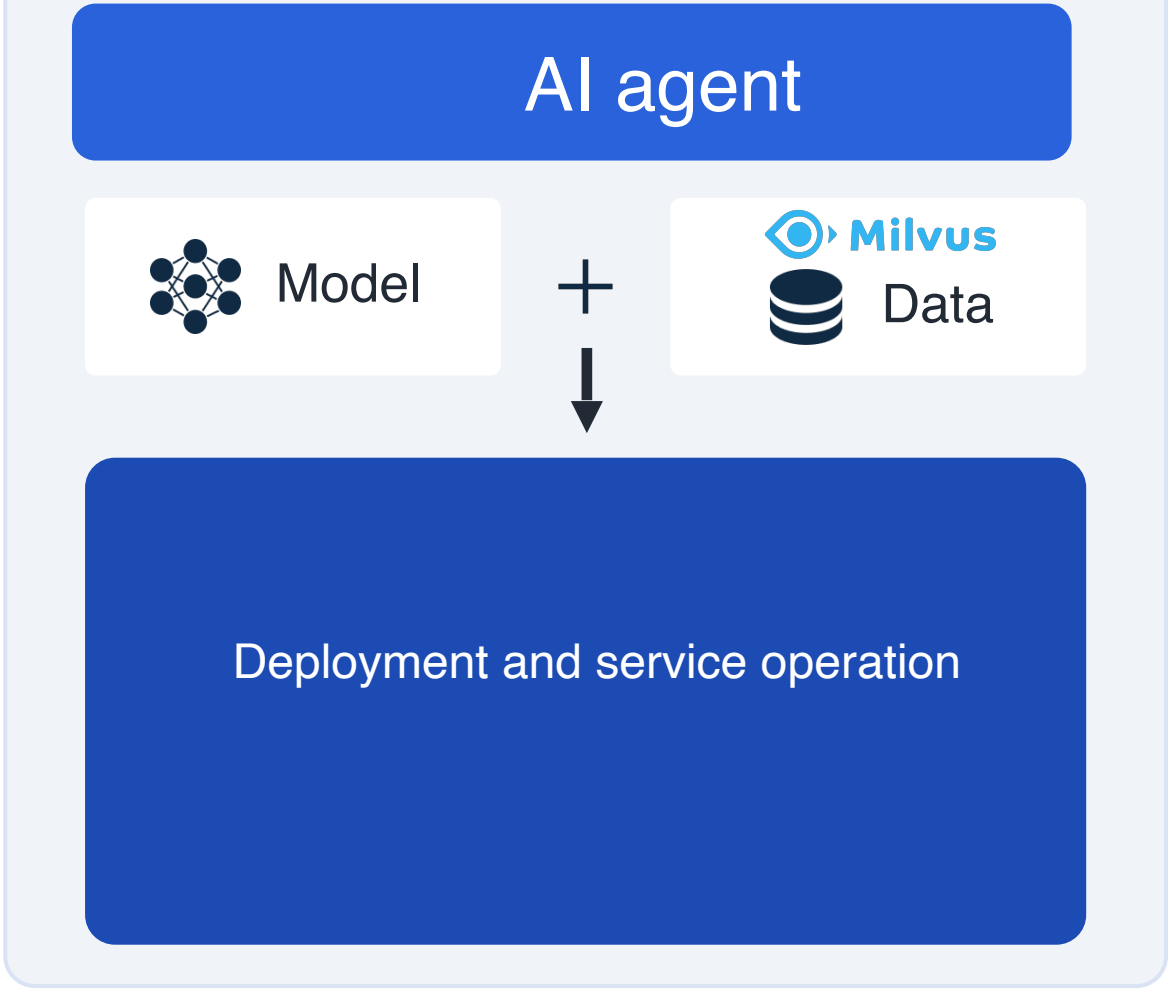
- Milvus & Friendli Serverless Endpoints
- Tool Calling (pre-defined search tool & custom tool)
- Retrieval-Augmented Generation

← Google Colab Link

Operating a large-scale AI agent service with Friendli

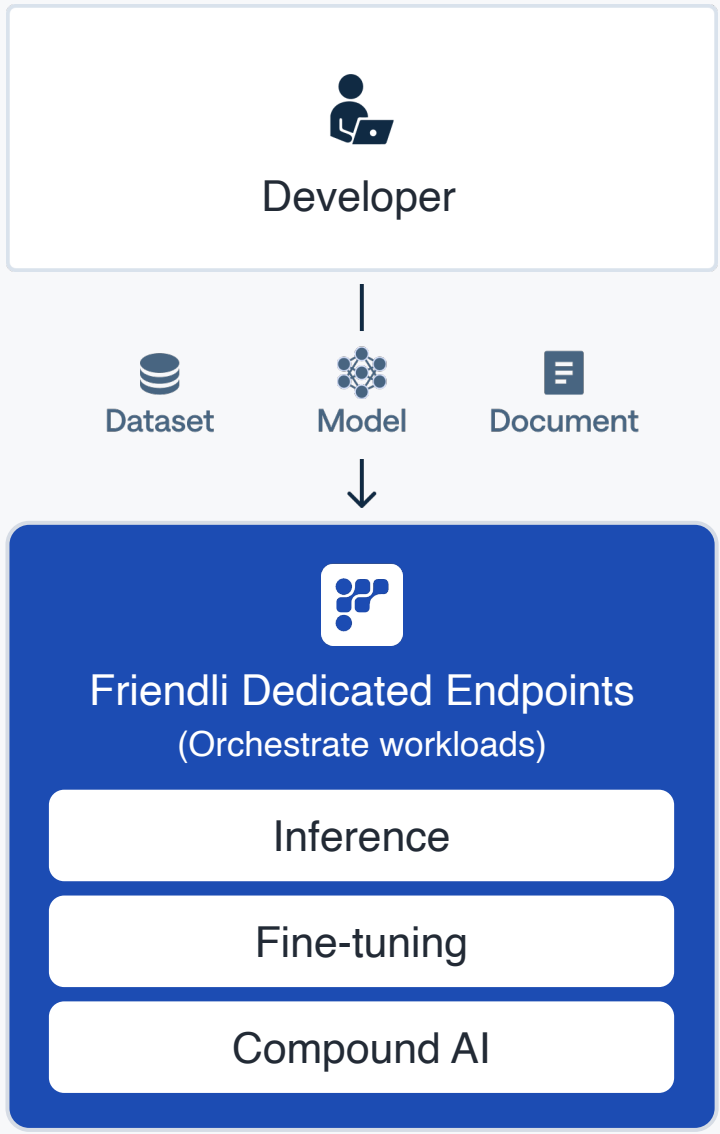
AI agent service operator

Scale up,
Custom models



PRODUCTS

Friendli Dedicated Endpoints



Friendli Dedicated Endpoints

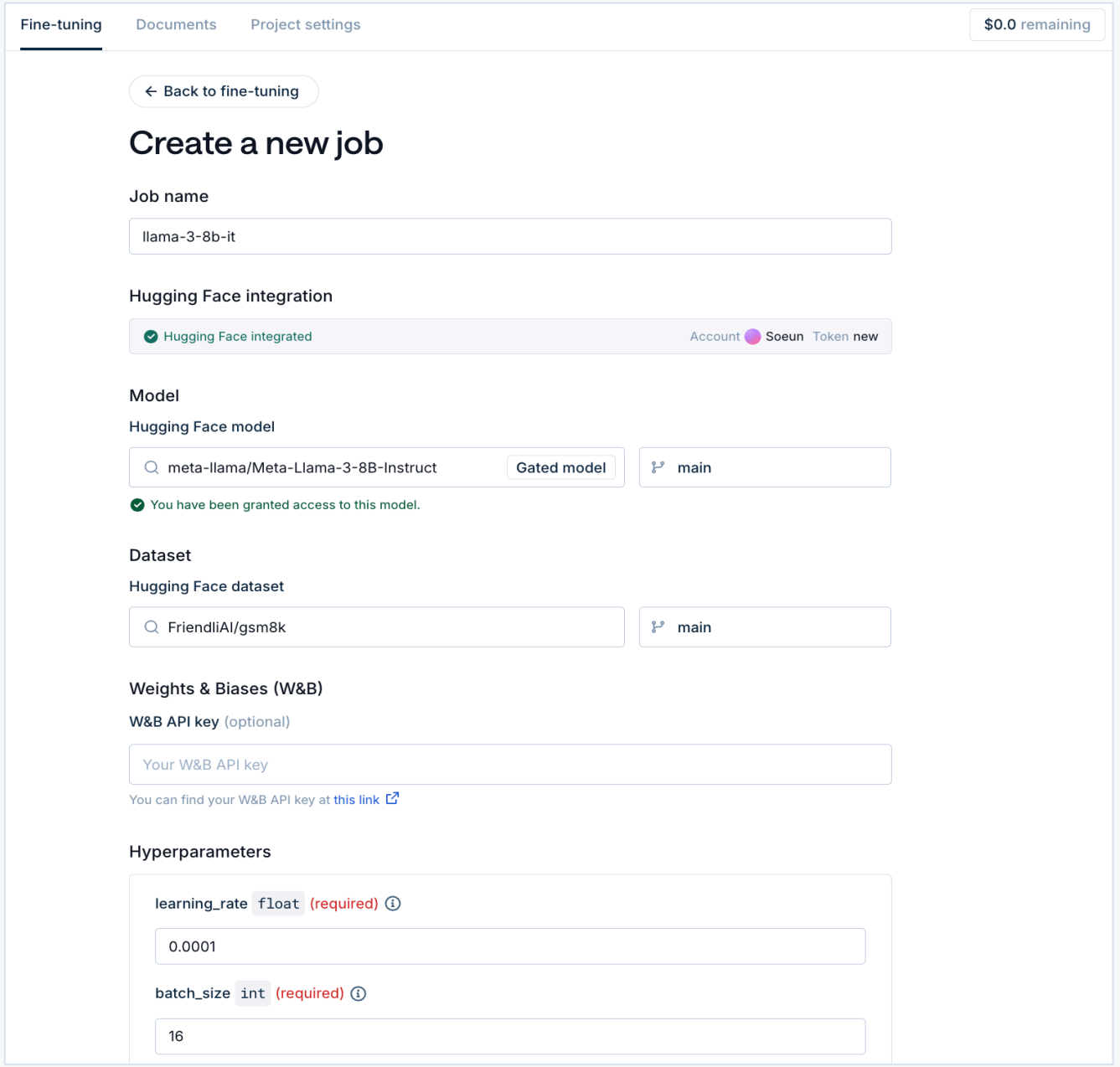
Deploy and create your custom model on the GPUs of your choice with just a few simple clicks.

Automated, cost-efficient serving and fine-tuning

- Superior cost-efficiency and performance
- Custom model support
- Dedicated GPU instances
- Automated and intelligent operations
- Highly reliable service with strict SLAs
- Integration with the gen AI ecosystem

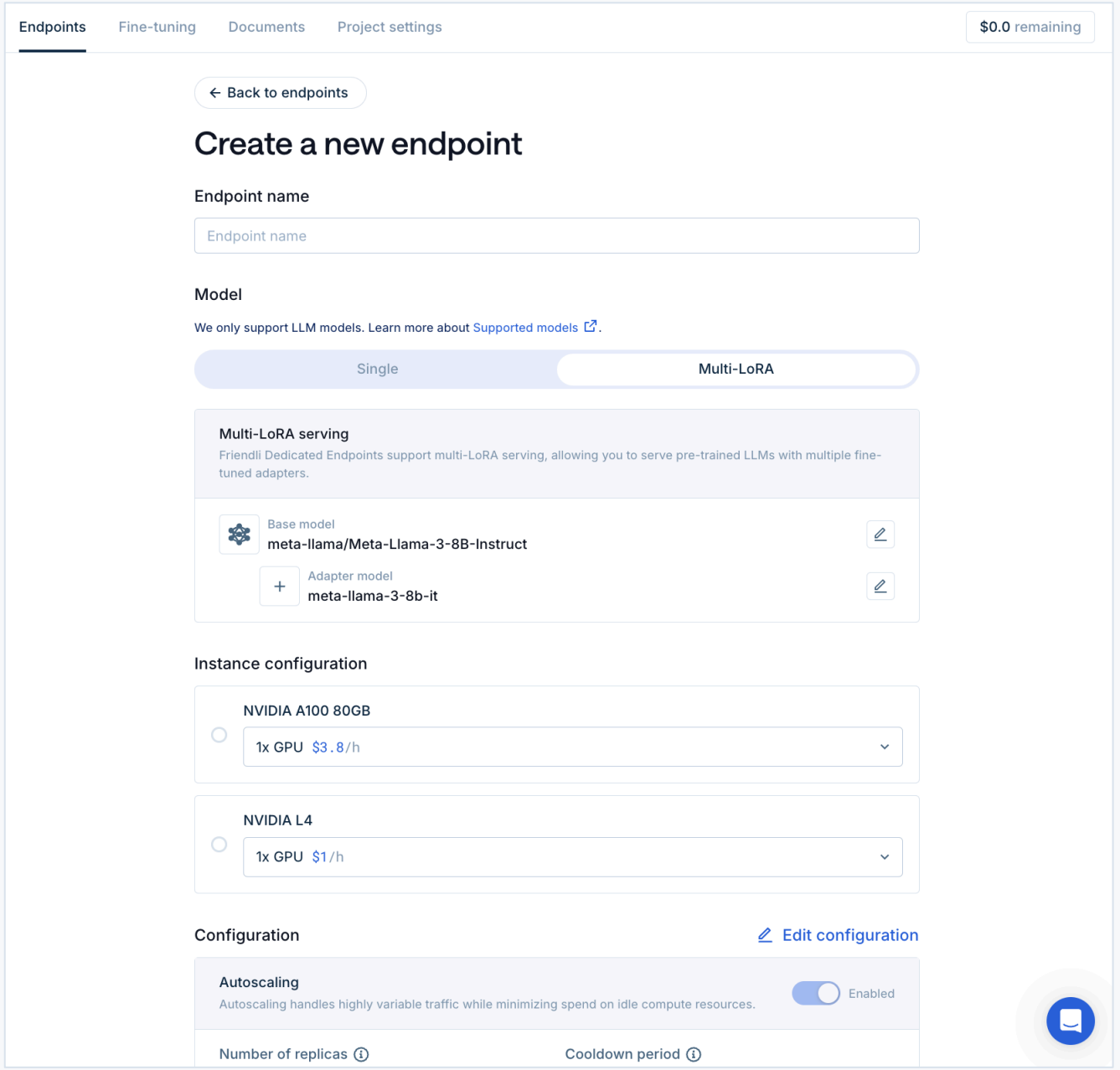
1. Fine-tuning

Create your custom model by fine-tuning

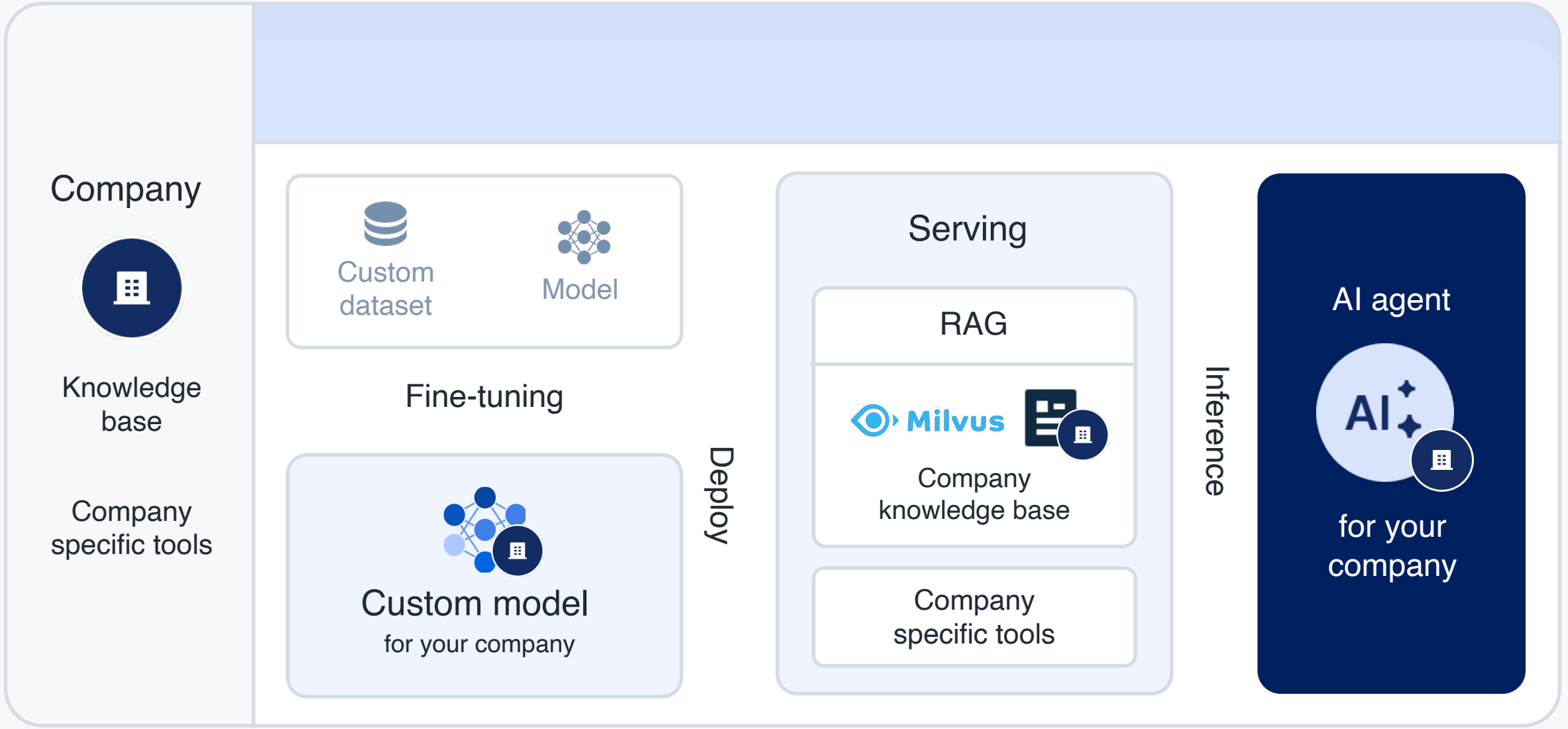


2. Inference

Deploy your fine-tuned model



FriendliAI & Milvus: Build AI agents



Google Colab

FriendliAI

Let’s start building!

LinkedIn

friendli.ai

Friendli Suite

https://suite.friendli.ai