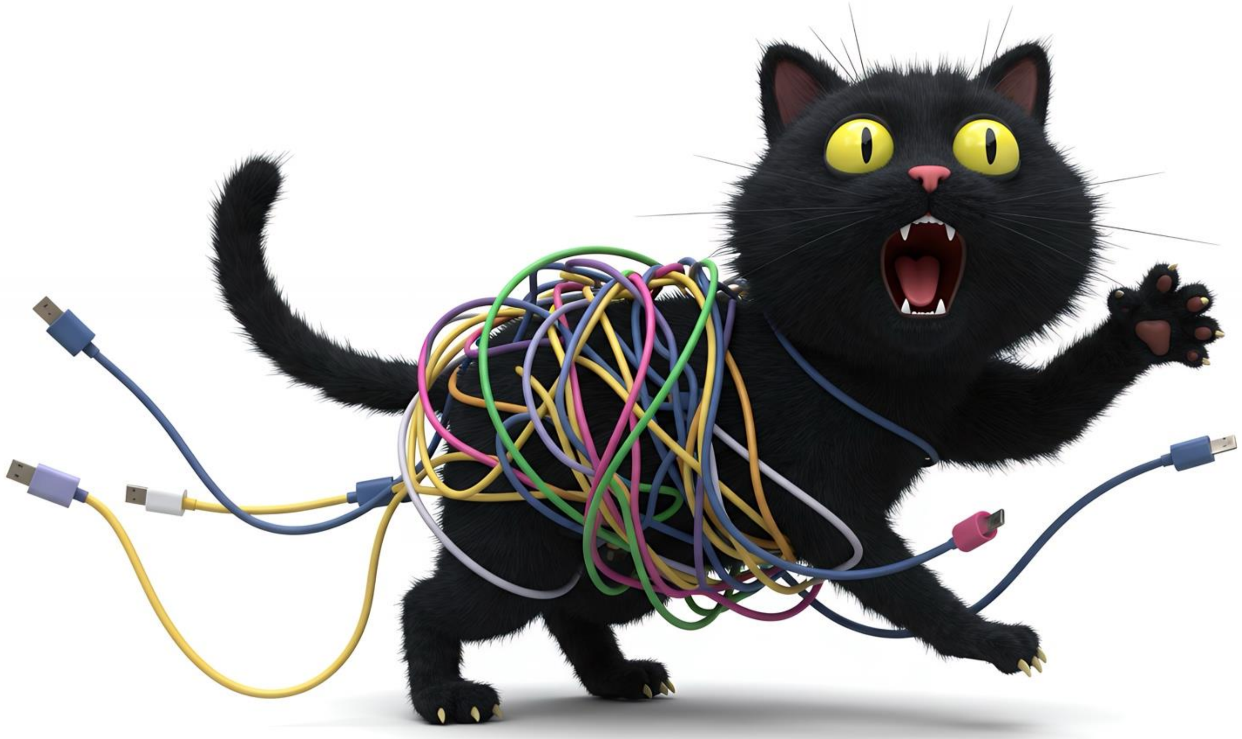


RAGE Against the Machine: Fighting AI Complexity with Kubernetes Simplicity

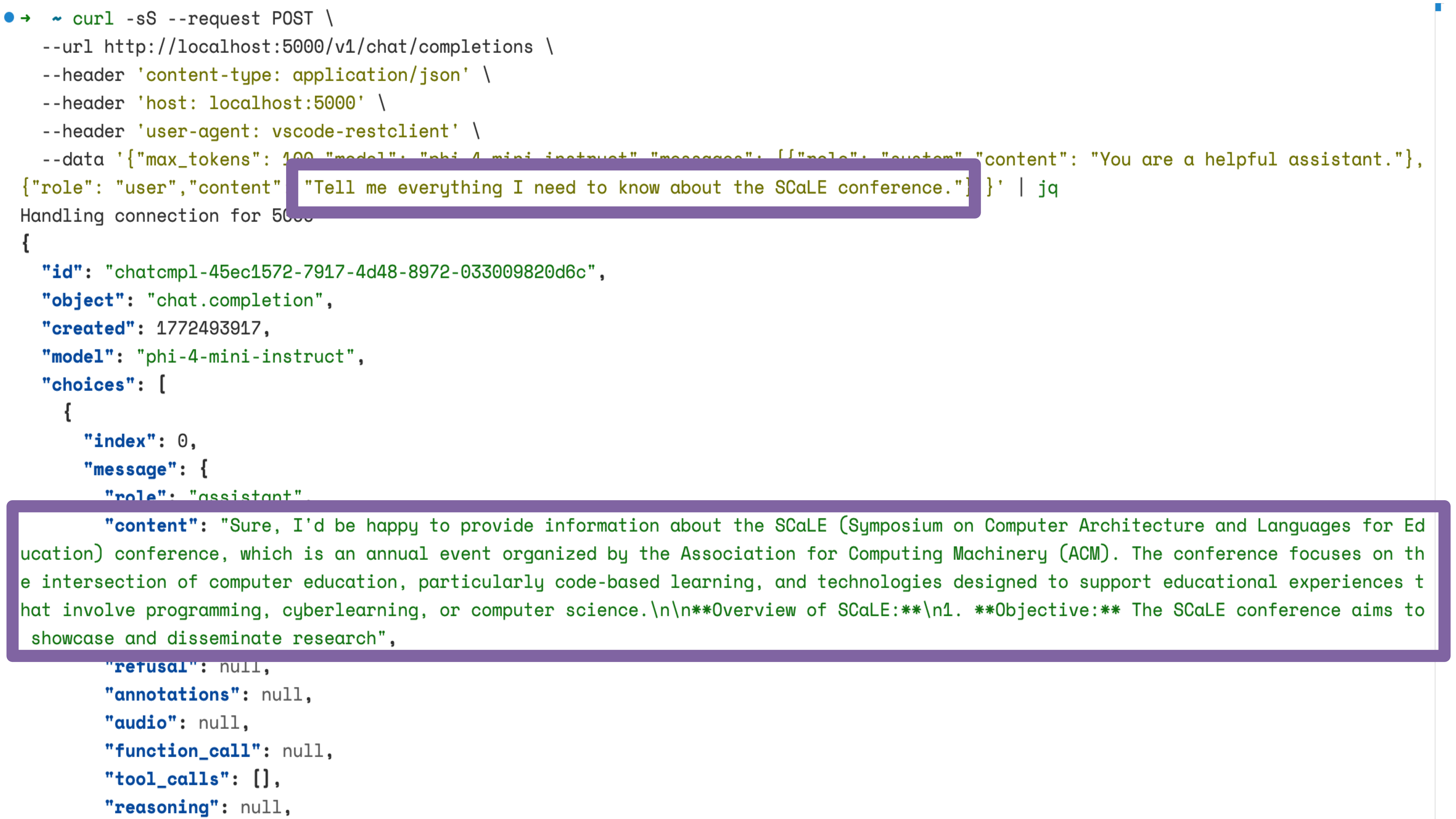
Paul Yu
Microsoft
Developer Advocate

What you’ll learn

- Base LLMs struggle in real systems
- What RAG is and how it add more complexity
- How KAITO simplifies inference + RAG on Kubernetes via CRDs
- How Gateway API Inference Extension helps route inference traffic
- A practical demo you can reproduce from my sample repo

THE CHALLENGE OF MAKING LLMS USEFUL

Great at language… not great at your specific facts



```
● ➜  ~ curl -sS --request POST \
  --url http://localhost:5000/v1/chat/completions \
  --header 'content-type: application/json' \
  --header 'host: localhost:5000' \
  --header 'user-agent: vscode-restclient' \
  --data '{"max_tokens": 100,"model": "phi-4-mini-instruct","messages": [{"role": "system","content": "You are a helpful assistant."},
{"role": "user","content": "Tell me everything I need to know about the SCaLE conference."}]}' | jq
Handling connection for 5000
{
  "id": "chatcmpl-45ec1572-7917-4d48-8972-033009820d6c",
  "object": "chat.completion",
  "created": 1772493917,
  "model": "phi-4-mini-instruct",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "Sure, I'd be happy to provide information about the SCaLE (Symposium on Computer Architecture and Languages for Ed
ucation) conference, which is an annual event organized by the Association for Computing Machinery (ACM). The conference focuses on th
e intersection of computer education, particularly code-based learning, and technologies designed to support educational experiences t
hat involve programming, cyberlearning, or computer science.\n\n**Overview of SCaLE:**\n1. **Objective:** The SCaLE conference aims to
 showcase and disseminate research",
        "refusal": null,
        "annotations": null,
        "audio": null,
        "function_call": null,
        "tool_calls": [],
        "reasoning": null,
```

AI hallucinations 😵‍💫 💫

- LLMs can sound confident even when wrong
- They may invent citations or facts
- In production, wrong answers are worse than ***"I don't know"***

LLMS HAVE A SPECIFIC JOB

They predict tokens - not truth

Prediction machines

- LLMs predict the next token based on patterns
- They optimize plausibility, not correctness
- Without constraints, they’ll fill in the gaps

Base models lack context

- Your runbooks, tickets, wikis, repos, and schedules aren’t in the training set
- Even if they were… they’re outdated the moment they ship
- You need a way to inject current, private context

Fine-tuning can be costly

- Requires curated data, training infra, and iteration
- Can be expensive and time-consuming
- Doesn’t automatically solve freshness for constantly changing data

RETRIEVAL-AUGMENTED GENERATION (RAG)

Ground LLMs with your relevant external knowledge

What is RAG?

- Retrieve relevant documents/snippets for a query
- Augment the prompt with that context
- Generate an answer grounded in retrieved content

How RAG works

Ingest

- Load docs → chunk → embed

Index

- Store vectors in a vector DB

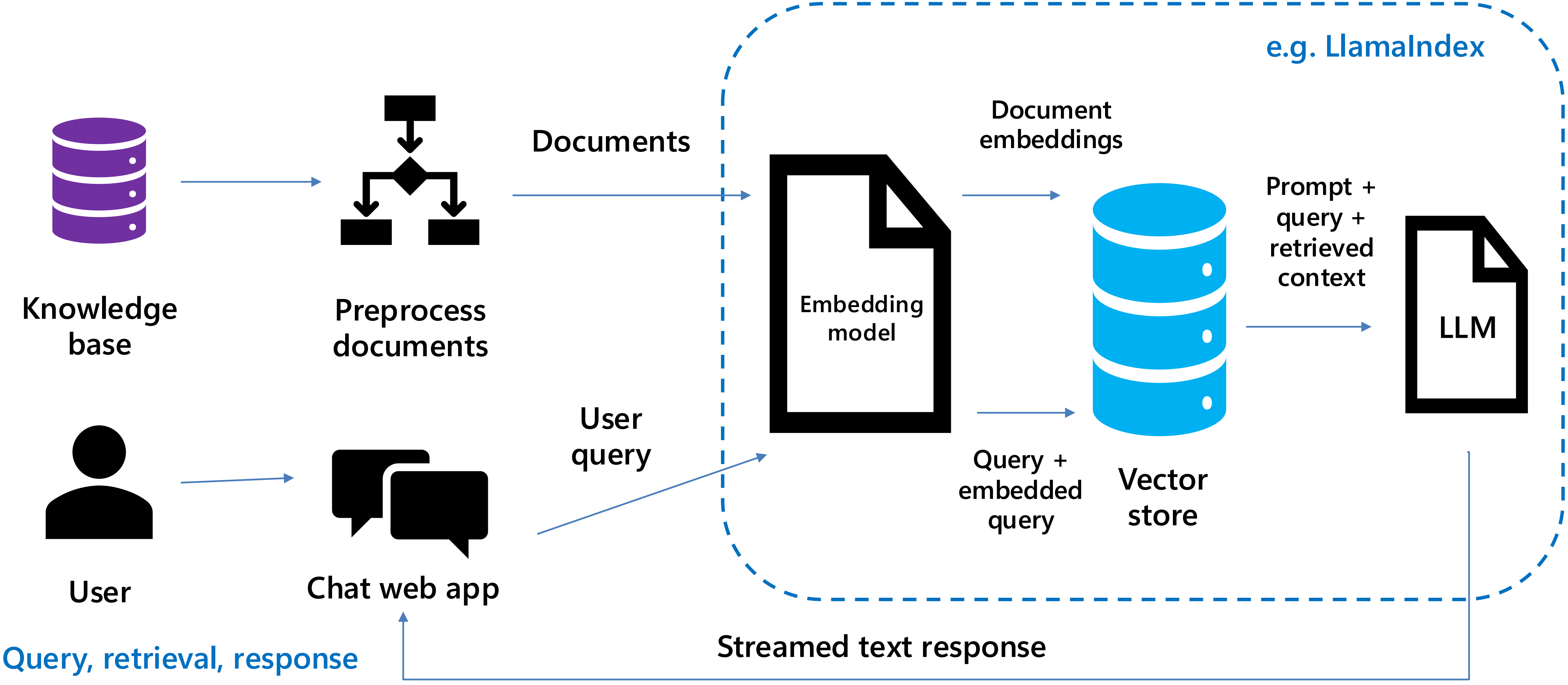
Retrieve

- Embed query → similarity search

Generate

- Send top-k context to LLM → answer

Query, retrieval, response



Knowledge base

Preprocess documents

Documents

User

Chat web app

User query

e.g. LlamaIndex

Embedding model

Document embeddings

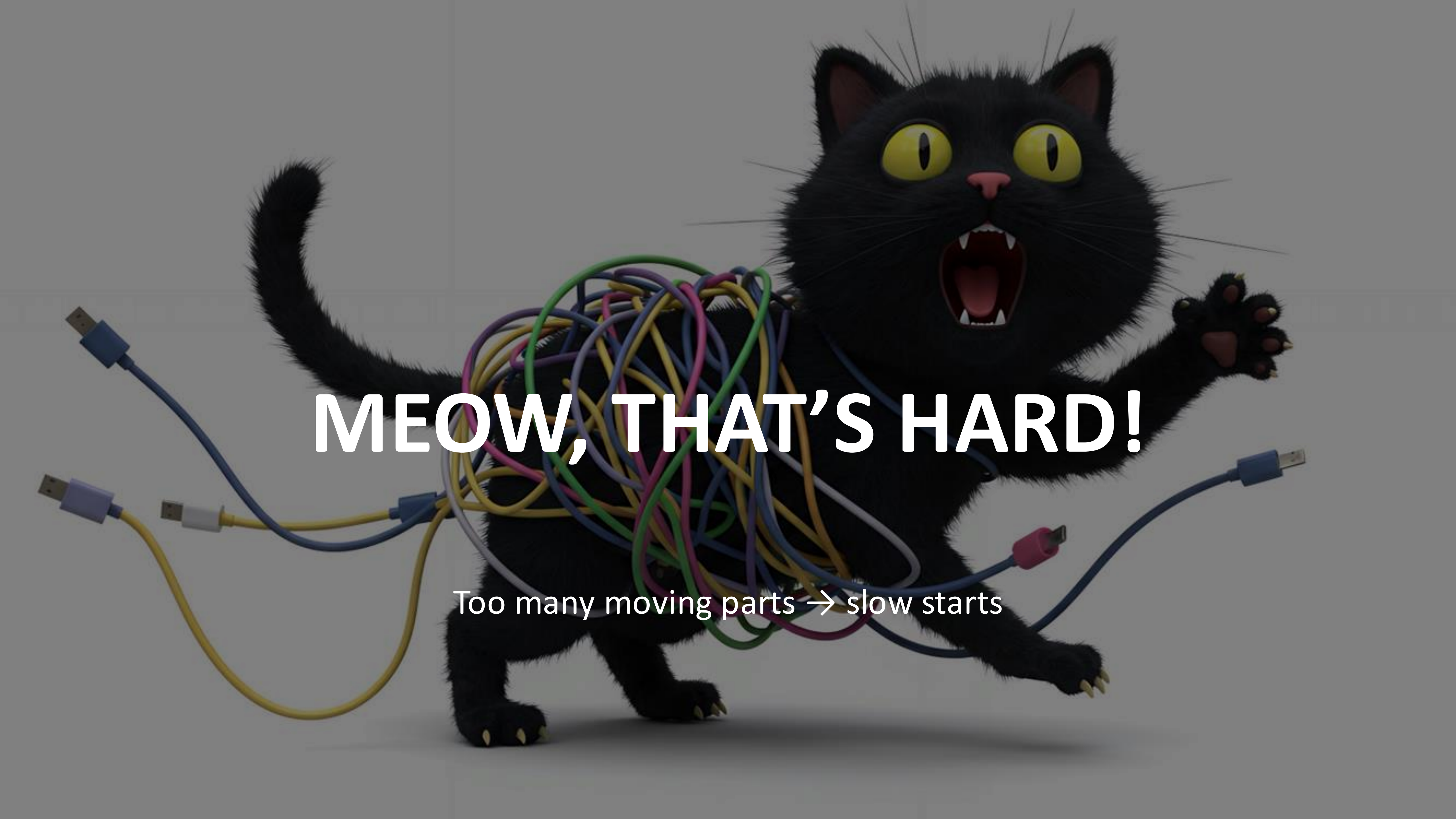
Query + embedded query

Vector store

Prompt + query + retrieved context

LLM

Streamed text response



MEOW, THAT’S HARD!

Too many moving parts → slow starts

The developer’s reality

- You end up building the platform before the product
- Kubernetes makes ops repeatable, but RAG still needs a clean abstraction
- What if RAG were just… a Kubernetes resource?

HELLO,

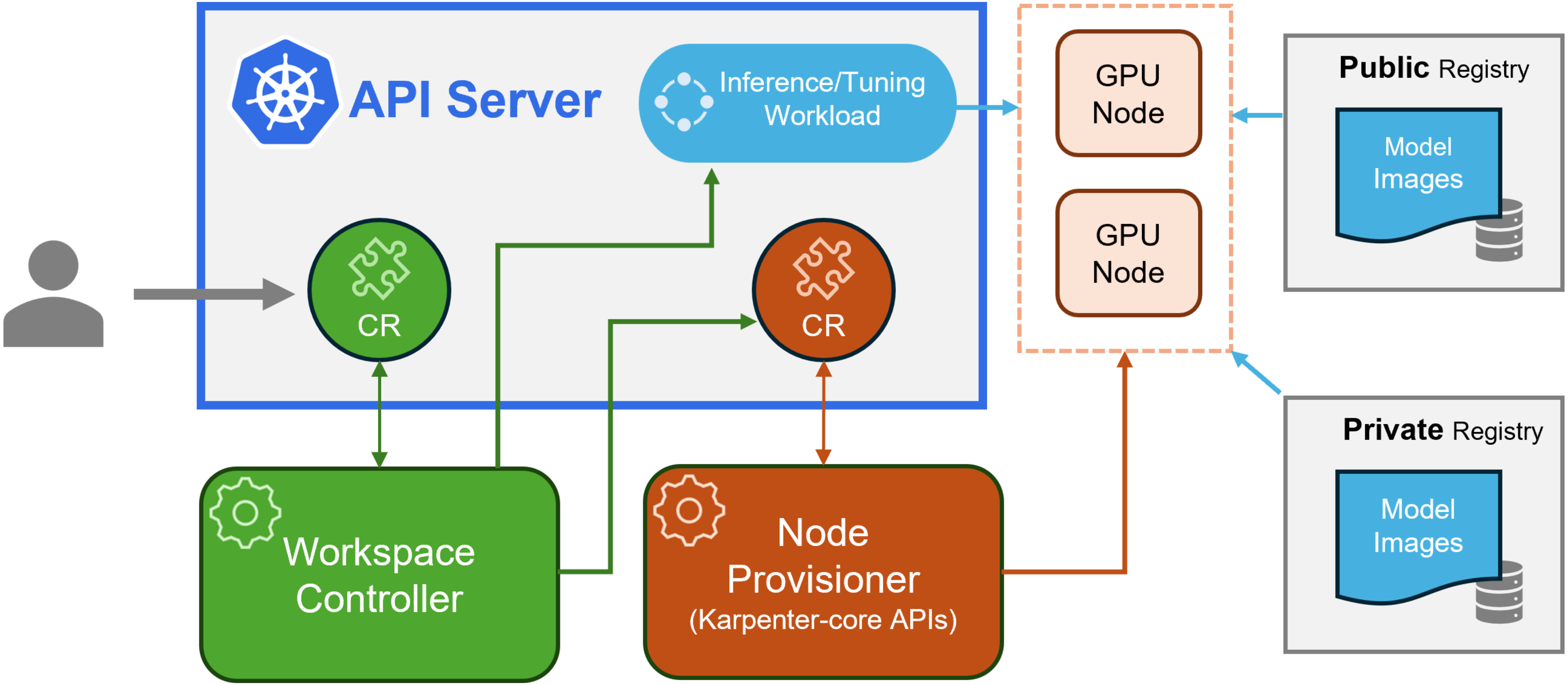
KAITO

Kubernetes AI Toolchain Operator

CLOUD NATIVE
COMPUTING FOUNDATION

Operator design

- Declarative CRDs for inference/tuning/RAG
- Controllers reconcile GPU infra + serving components
- Kubernetes-native UX: YAML + GitOps friendly


API Server
Inference/Tuning Workload
CR
CR
GPU Node
GPU Node
Public Registry
Model Images
Private Registry
Model Images
Workspace Controller
Node Provisioner
(Karpenter-core APIs)

Zero-to-inference in a few minutes

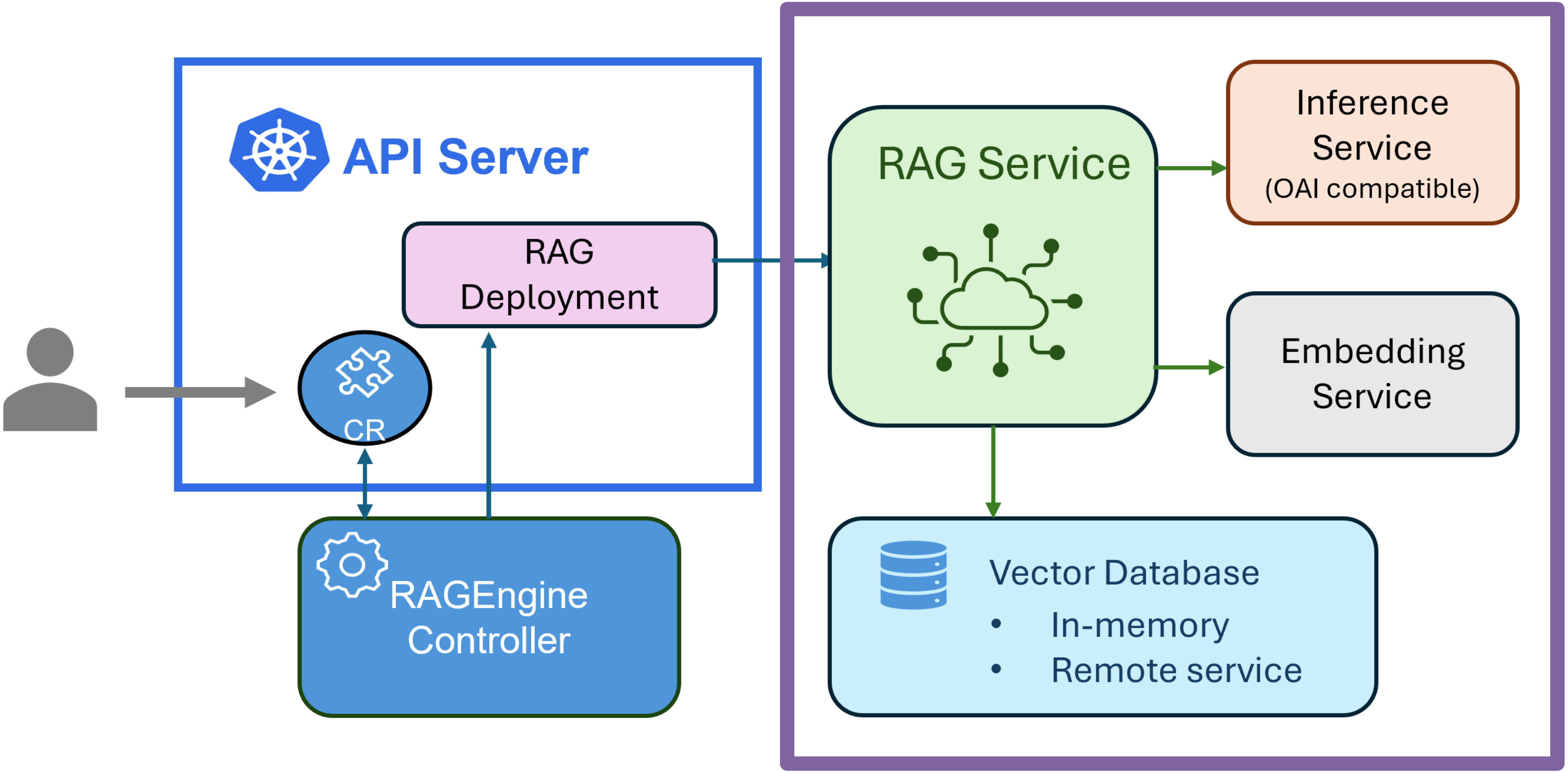
- Pick a supported model preset
- Apply a Workspace CR
- KAITO provisions GPU nodes, pulls model images, exposes an OpenAI-compatible endpoint

Demo

Workspace in action

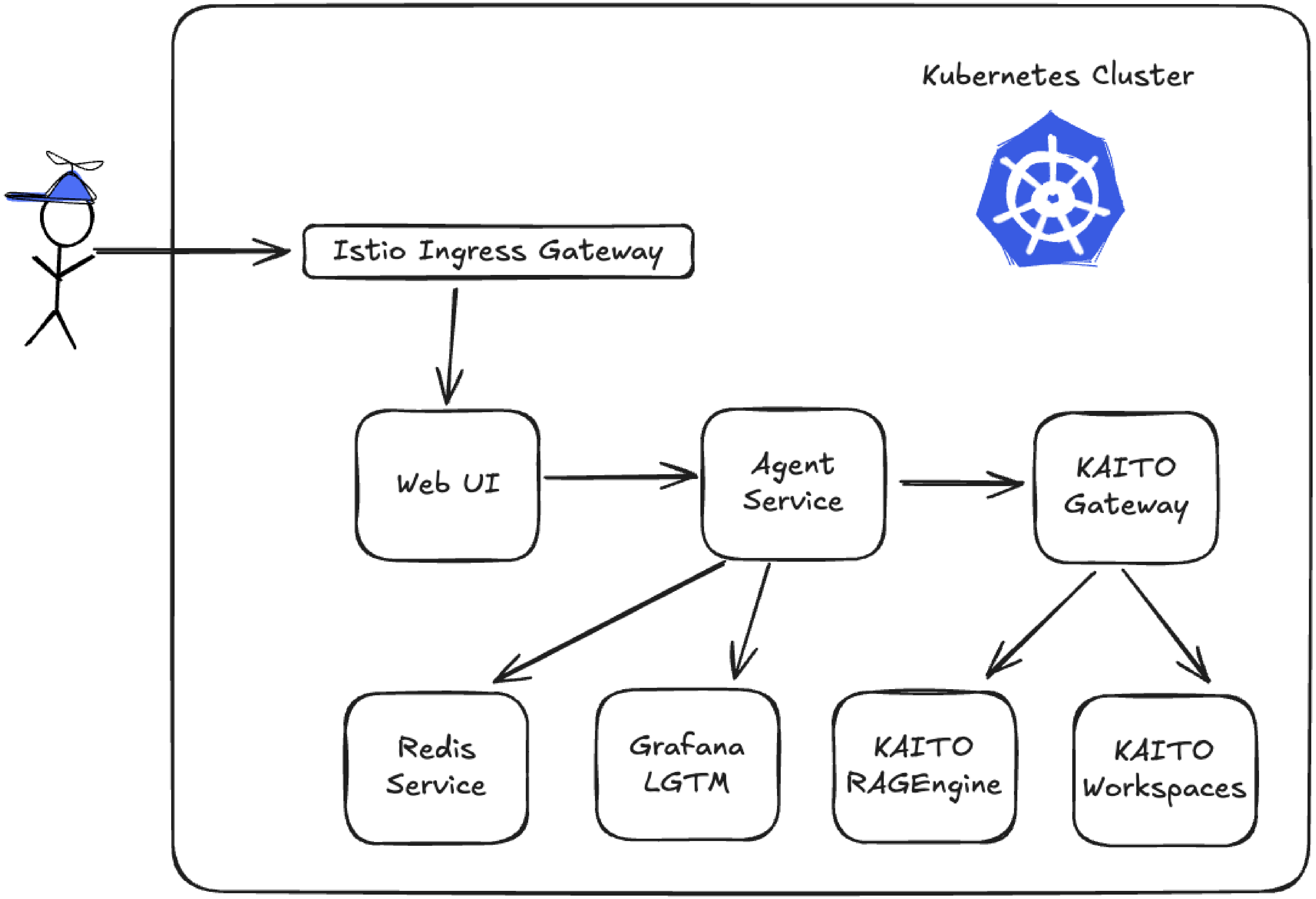
RAGEngine

- A single CR to stand up an end-to-end RAG pipeline
- Encapsulates vector store + embeddings + retrieval + inference wiring
- Runs in-cluster; scales with Kubernetes
- Less glue code, more app logic


API Server
RAG
Deployment
CR
RAGEngine
Controller
RAG Service
Inference
Service
(OAI compatible)
Embedding
Service
Vector Database
• In-memory
• Remote service

DEMO OVERVIEW

High-level architecture

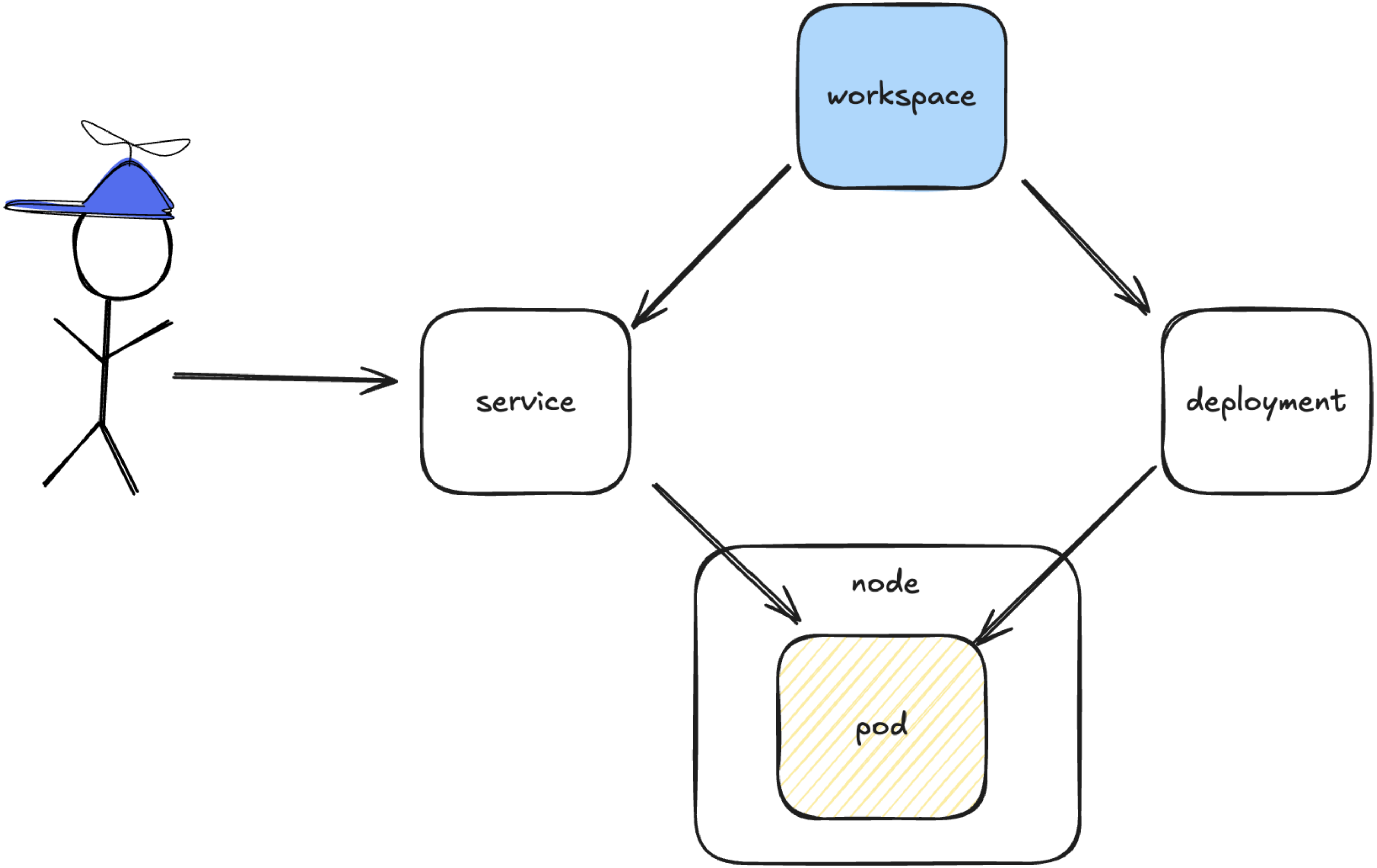

Kubernetes Cluster
Istio Ingress Gateway
Web UI
Agent Service
KAITO Gateway
Redis Service
Grafana LGTM
KAITO RAGEngine
KAITO Workspaces

SCALE23X CHAT

From “no context” → grounded answers

Workspace

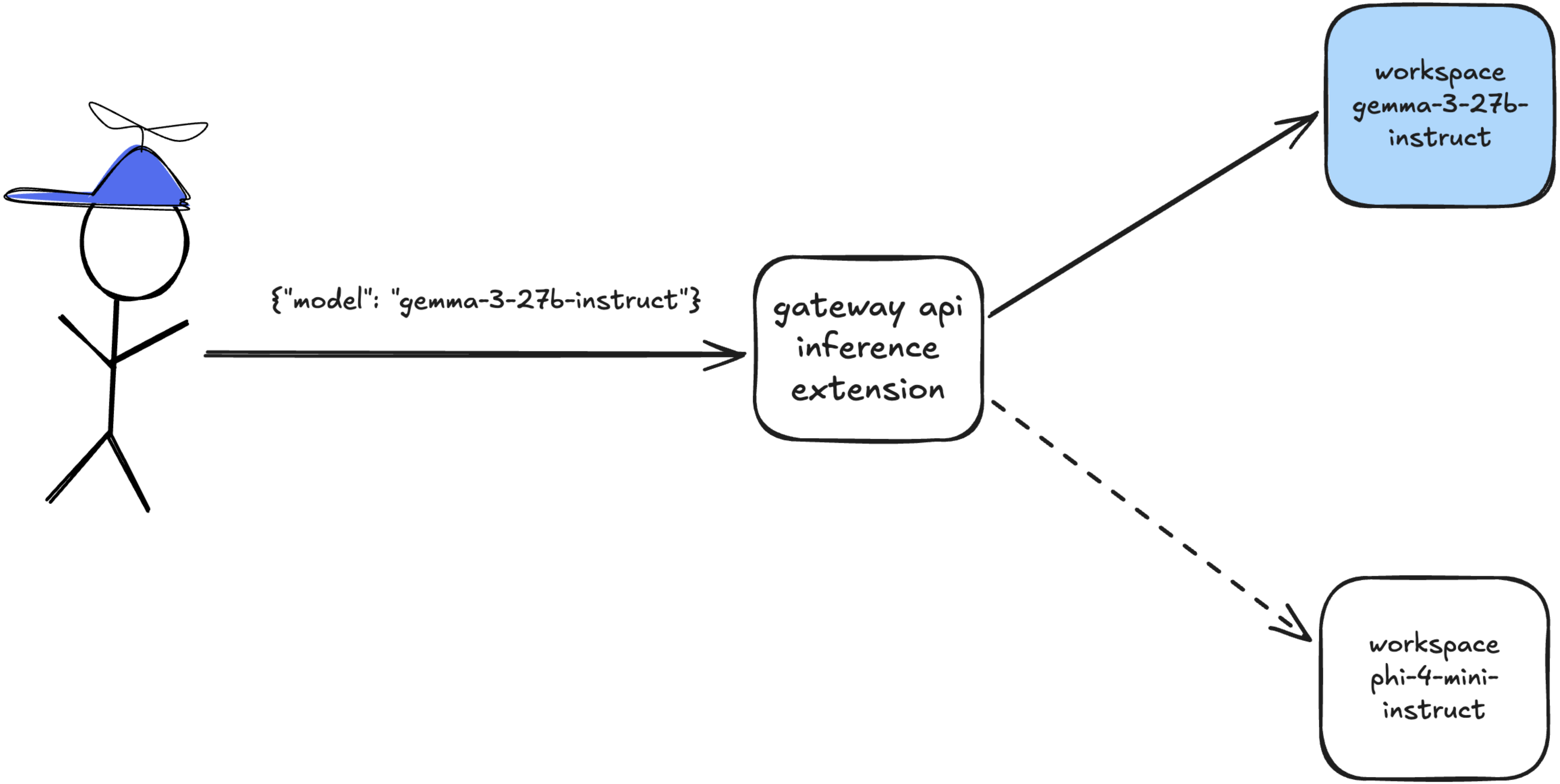
- Smallest unit of inference
  - Node -> Deploy -> Pod -> Svc
  - OpenAI compatible endpoint
- NEW! InferenceSet is a higher-level abstraction for:
  - Dynamic model routing
  - Inference scaling with KEDA



Learn more: https://kaito-project.github.io/kaito/docs/inference

Gateway API Inference Extension (GIE)

- Dynamic routing for inference
- Body-based routing
- Use the right model for the job



{"model": "gemma-3-27b-instruct"}

gateway api inference extension

workspace gemma-3-27b-instruct

workspace phi-4-mini-instruct

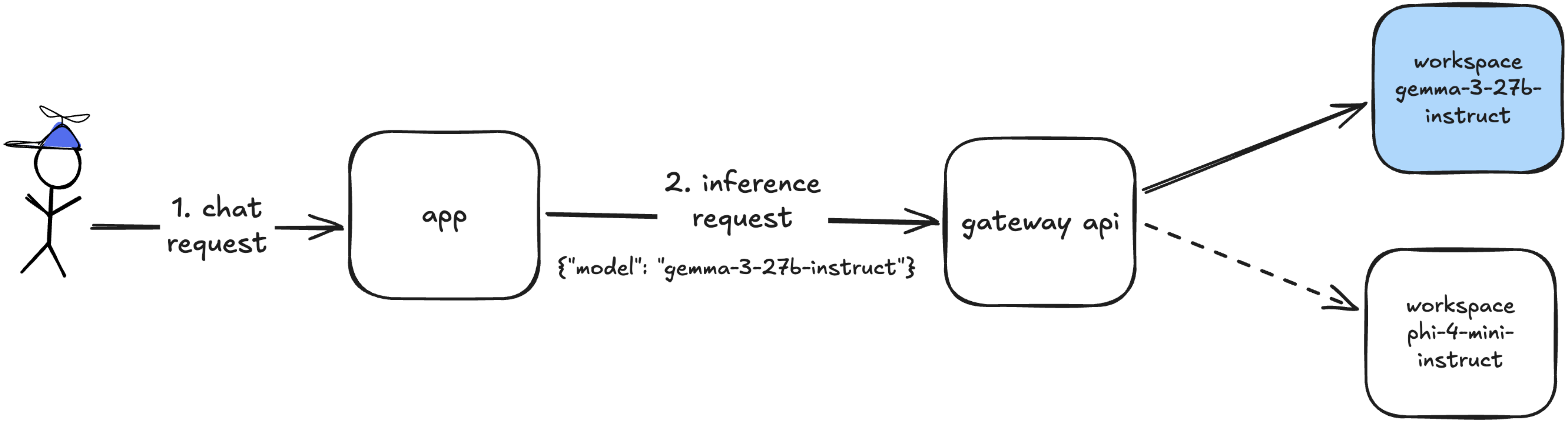
Learn more: https://gateway-api-inference-extension.sigs.k8s.io

Demo

Gateway API Inference Extension

SCALE23X Assistant

- App routing through gateway
- Ask about the conference schedule

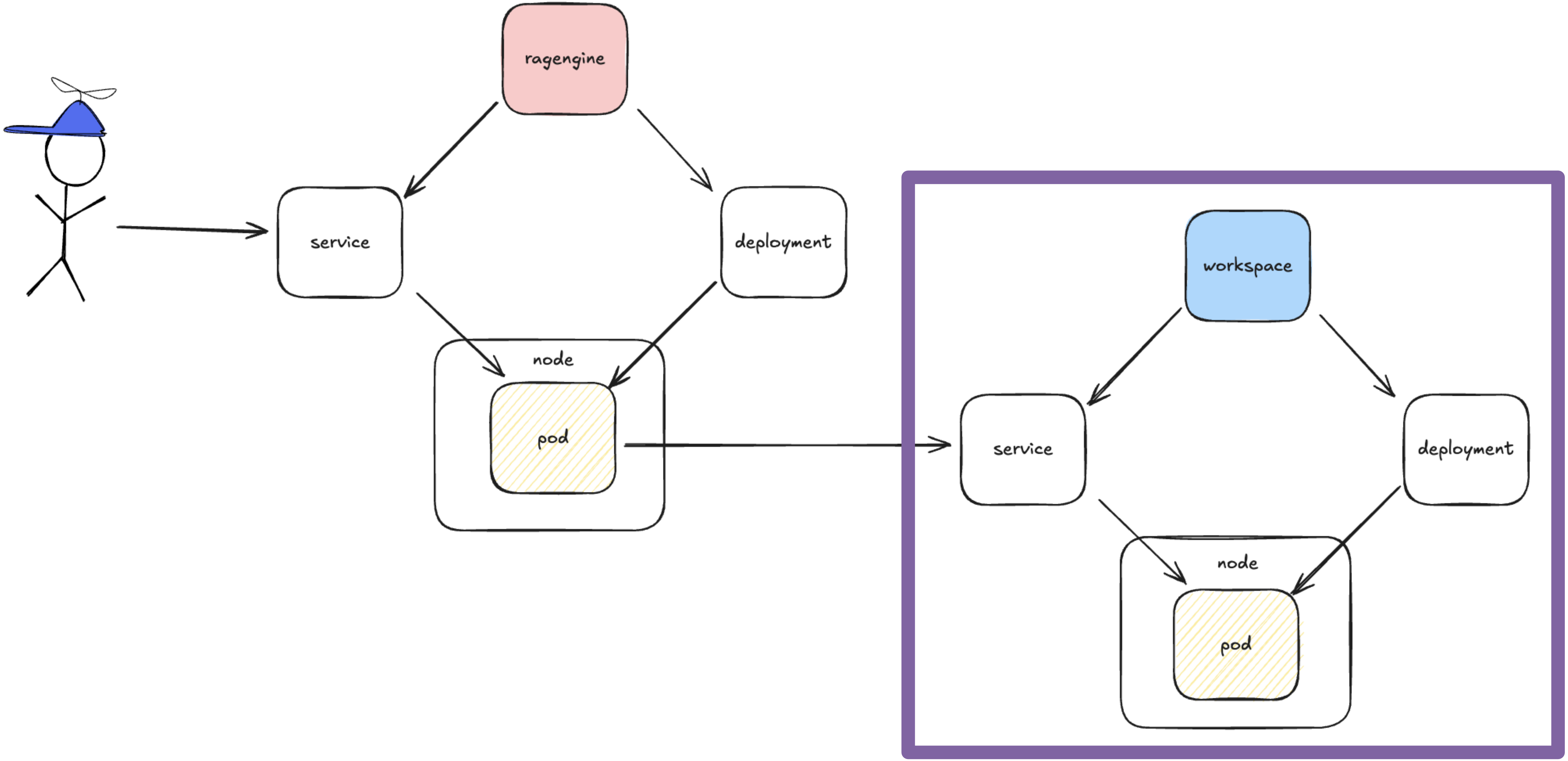

1. chat request
app
2. inference request
{"model": "gemma-3-27b-instruct"}
gateway api
workspace gemma-3-27b-instruct
workspace phi-4-mini-instruct

Demo

Chat with agent

RAGEngine

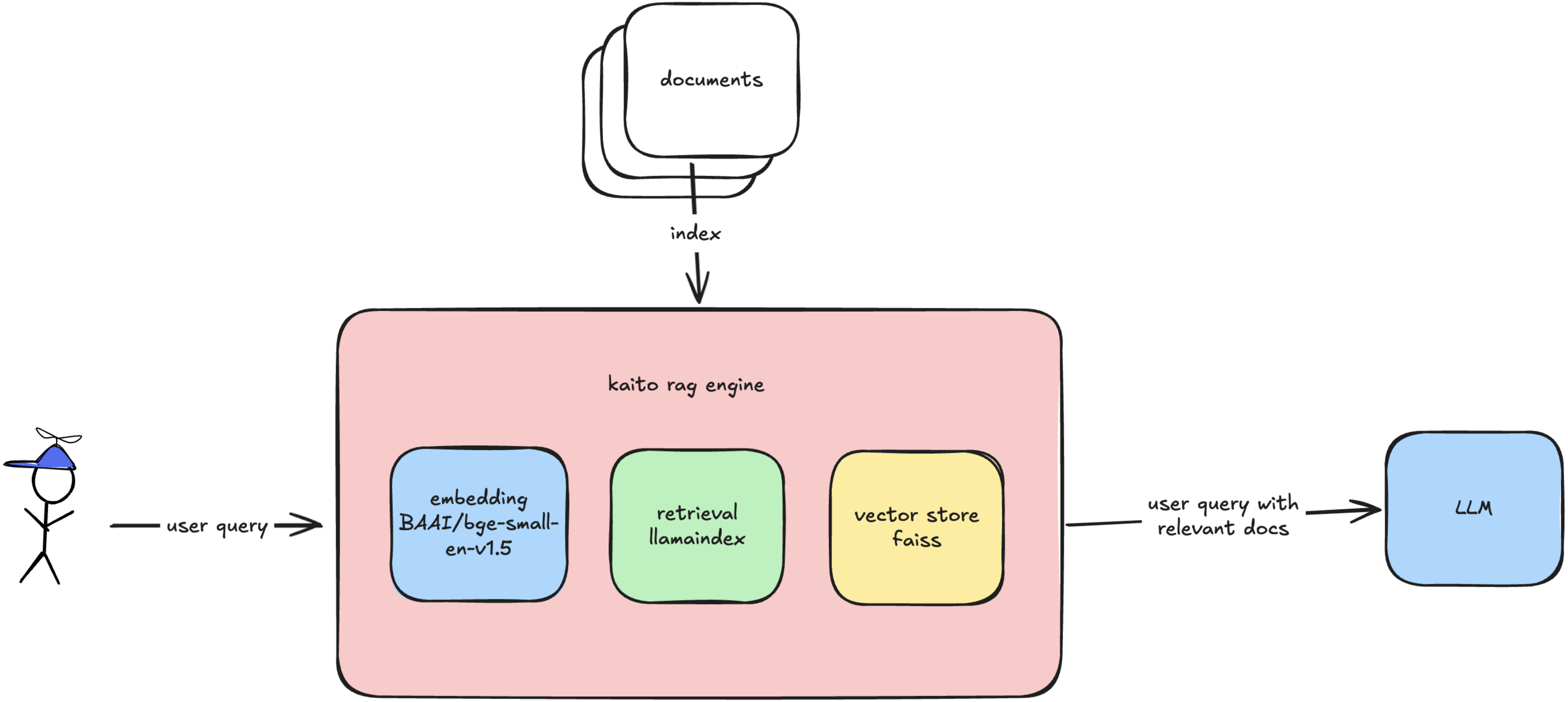
- Yet another custom resource
- Like a Workspace
- Pass-through

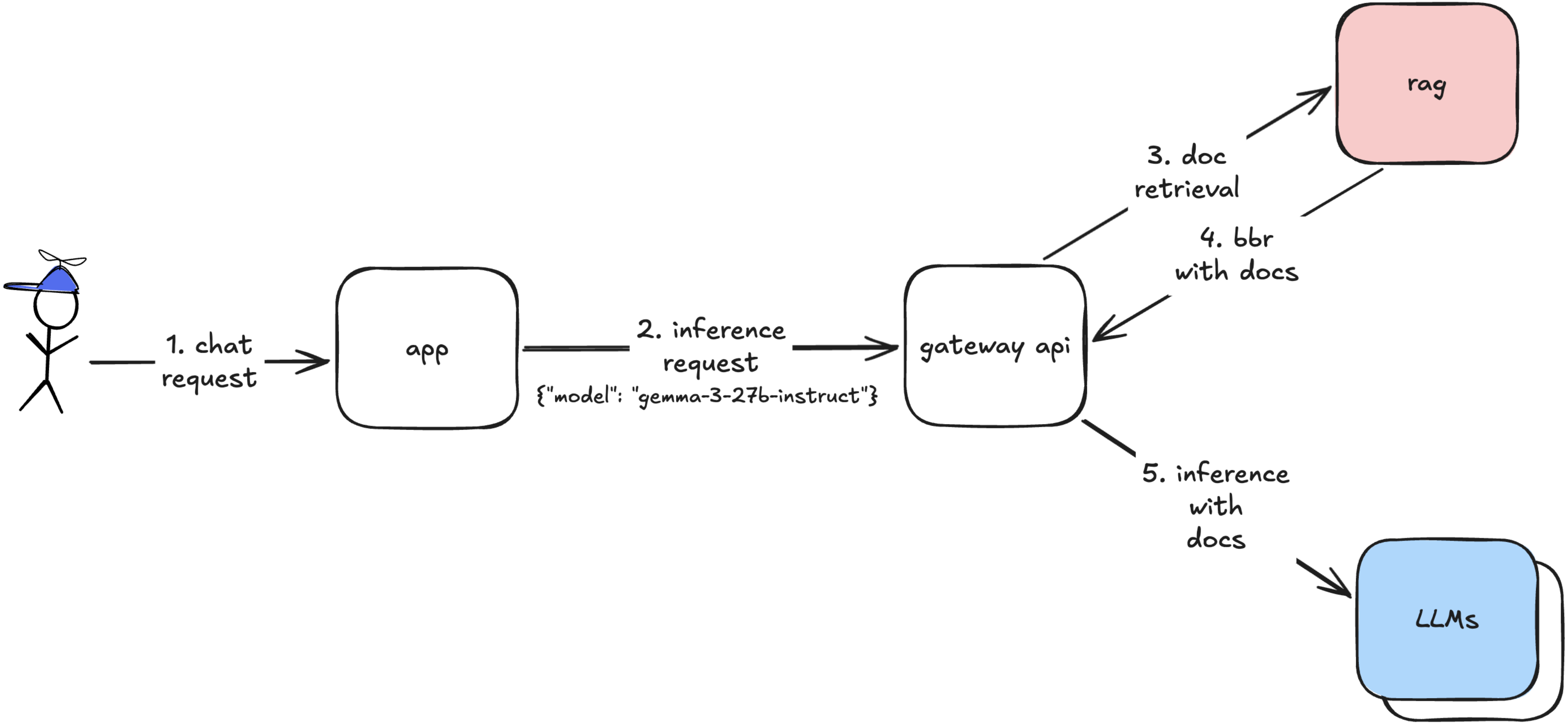


Learn more: https://kaito-project.github.io/kaito/docs/rag

Demo

Context is all you need


documents
index
kaito rag engine
user query
embedding
BAAI/bge-small-en-v1.5
retrieval
llamaindex
vector store
faiss
user query with relevant docs
LLM

rag
3. doc retrieval
4. bbr with docs
1. chat request
app
2. inference request
{"model": "gemma-3-27b-instruct"}
gateway api
5. inference with docs
LLMs

Demo

Routing to RAG endpoint

Demo

Chat with RAG-enabled agent

KEY TAKEAWAYS

Grounding matters: RAG reduces hallucinations by adding relevant context

Classic RAG can be a lot of plumbing

KAITO + RAGEngine makes RAG Kubernetes-native and declarative

Gateway API Inference Extension helps manage inference traffic intelligently

Start small: get a working pipeline first, then iterate

TRY IT YOURSELF

Thank you!

- KAITO docs: kaito-project.github.io/kaito
- RAG docs: kaito-project.github.io/kaito/docs/rag
- GIE docs: kaito-project.github.io/kaito/docs/gateway-api-inference-extension
- Join the community: #kaito channel on CNCF Slack
- https://communityinviter.com/apps/cloud-native/cncf