Osprey Agent For RHIC/EIC and Injector Control Systems

Wenge Fu, Kevin Brown, Meifeng Lin, John Morris (Brookhaven National Lab)
Ningdong Wang (Cornell University)

- About Osprey-Framework and Agentic AI
- RHIC/EIC and their injection control systems
- Tools for bridging the gap between Osprey and local control systems
- The agent system and workflows
- Testing cases, issues found and upgrade plans
- Summary

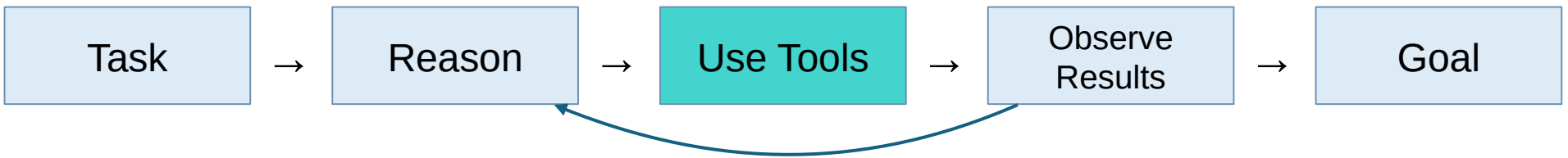
4th EIC-BeamAI Collaboration Meeting, Cornell University, 09/01/2026-09/03/2026

What is Agentic AI

- Agentic AI is an AI system that can independently plan, make decisions, and take multiple actions to reach a goal with very little human help.

- Agentic AI adds infrastructure around LLM so that it can pursue a goal autonomously.

- The agent system often includes memory and access to tools.



Brookhaven National Laboratory Cornell University

Motivation for Agentic AI in Accelerator Operations

Operational Complexity

- Accelerator operations involve many sub-systems, software tools, databases, and procedures.
- A single operational question can require live control data, archived data, elog records, configuration information, and expert knowledge.
- Agentic AI can help connect these information sources, use existing tools, and organize work into operator-supervised workflows.

Benefit of AI assistance

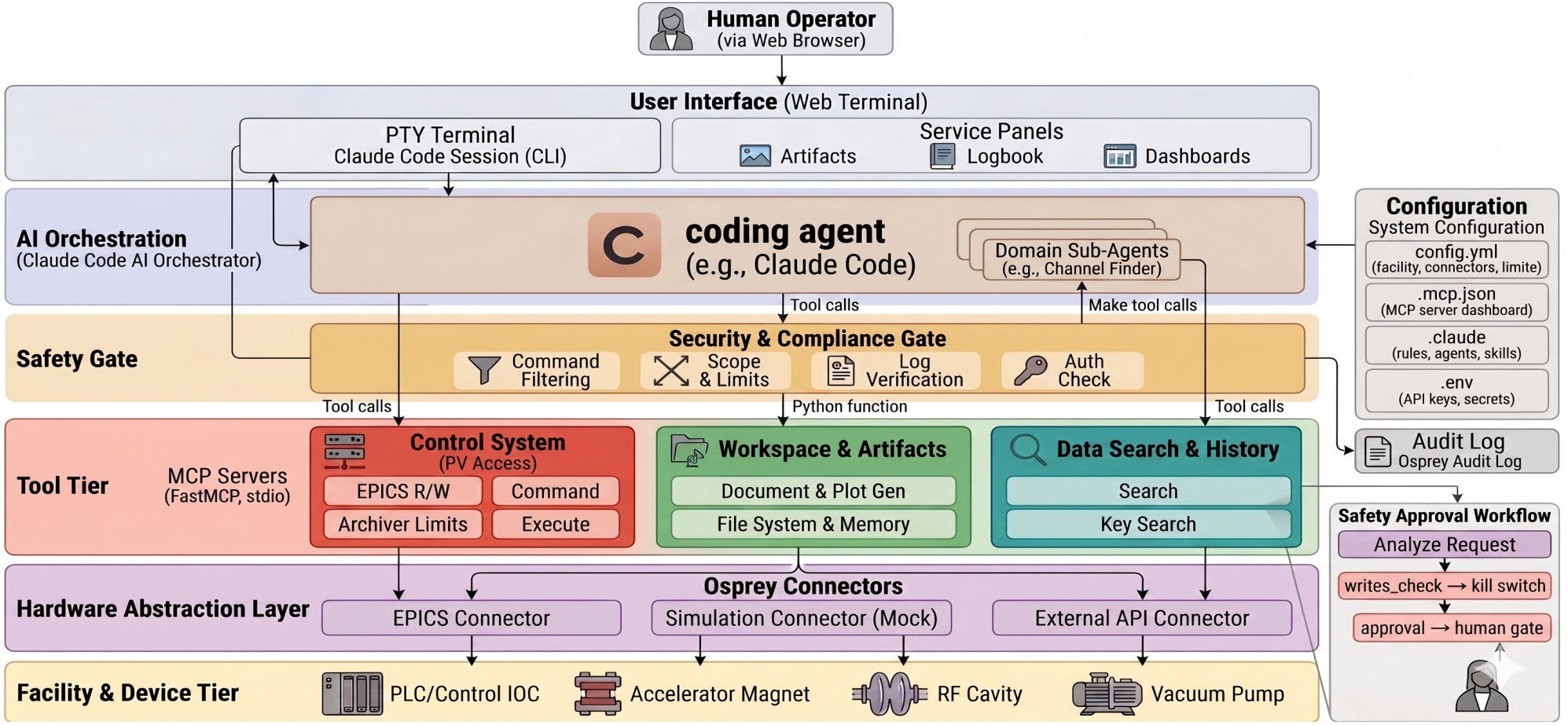
- Natural language interface
- Reduce operator cognitive load
- Accelerate non-routine tasks
- Enable reasoning across subsystems

Areas for Agentic AI in Accelerator Operations

- **Machine status and shift preparation:** summarize machine state and recent changes.
- **Control point discovery:** translate operator intent into facility-specific ADOs or channels.
- **Data retrieval and visualization:** combine live readings, archives, and generated plots.
- **Troubleshooting and fault analysis:** compare data, alarms, configuration, procedures, and elog history.
- **Procedure and experiment support:** prepare scans, analysis, tuning, and operator-approved workflows.
- **Knowledge transfer:** search documentation and operational history and prepare reports.

About Osprey-Framework

- It is an agentic interface and safety harness for safety-critical control systems.
  - Developed by Thorsten Hellert and a team at Lawrence Berkeley National Laboratory
  - Powered by sophisticated LLMs, the framework is capable of task retrieval, orchestrating tasks, understanding human logic, planning workflows, and performing human-controlled task execution.
  - It uses the Osprey agent as the orchestrator, MCP servers as the tool interface, and pluggable connectors for hardware access.
  - It features a broad set of internal capabilities (tool sets) that combine with user-defined capabilities (Tools) to chain elements into an effective and flexible workflow for desired tasks.

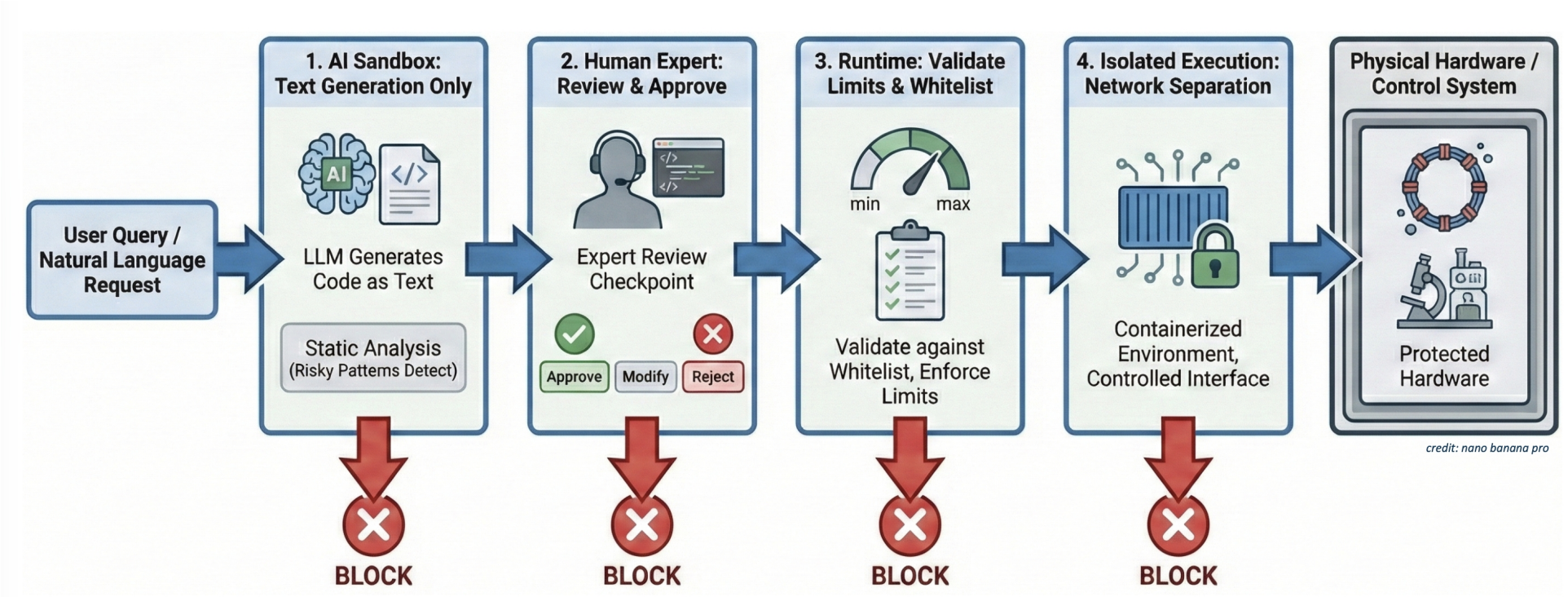


(Diagram from Osprey-framework web site: https://als-apg.github.io/osprey)

The Osprey-framework is fast-evolving software. In this experimental work, we used version v0.10.2 (released in February 2026). The latest version is V2026.6.2 (rewritten version).

Brookhaven National Laboratory Cornell University

Safety and Human Control



Brookhaven National Laboratory

Cornell University

RHIC/EIC and Injection Control Systems

- The RHIC complex consists of the RHIC (completed) and the upcoming EIC. The injector chain (Linac, EBIS, Booster, and AGS) will be continuously utilized by the EIC.
- Both RHIC and its injectors run on an **Accelerator Device Object (ADO)** based control system. This system functions as a massive machine tree encompassing all sub-systems and end control points.
- The system handles ~2.7 million control points (set points and measurement points). Approximately 50% of these points—primarily within the injector chain—will transition to the EIC.

To deploy an Osprey framework-based AI agent for monitoring and controlling these systems, the following technical prerequisites must be met:

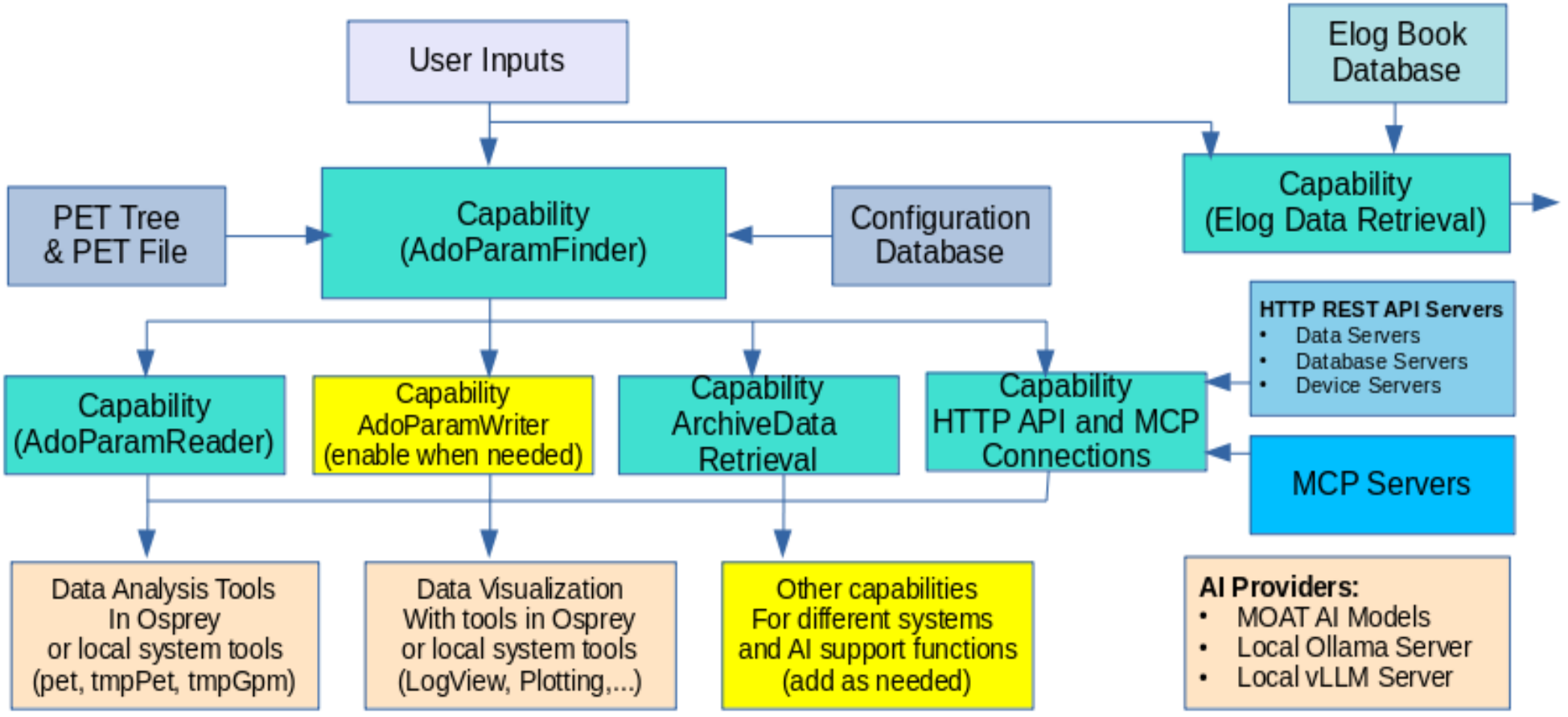
- **Point Discovery**: Navigate the giant machine tree to locate and identify specific target control points (ADO parameters).
- **Data Access:** Read live control point telemetry and write/update control point parameters.
- **Data Retrieval:** Access historical logs and time-series data for specific target control points.
- **Diagnostics**: Query and filter relevant event logs and system messages.
- **Workflow Automation:** Chain these functions using the Osprey agent into a highly flexible workflow to execute complex control tasks.

Capabilities (Tools) For Bridging the Gap Between Osprey-Framework and ADO Based Control Systems

Because the Osprey framework focuses primarily on EPICS-based control systems, its native capabilities (In Tool Tier) — such as PV channel finder, live data reader/writer, and archive data retrieval—cannot be applied directly to the Accelerator Device Object (ADO)-based control system at RHIC. To bridge this gap, we developed the following new capabilities (Tools) for the RHIC/EIC and injector control systems:

- ADO and parameter finder
- Live ADO data access (reading only for now)
- ADO archive data retrieval
- Integration of local Control system programs and APIs
- Elog data retrieval and analysis

ADO Control Assistant (with Osprey Conversational UI)



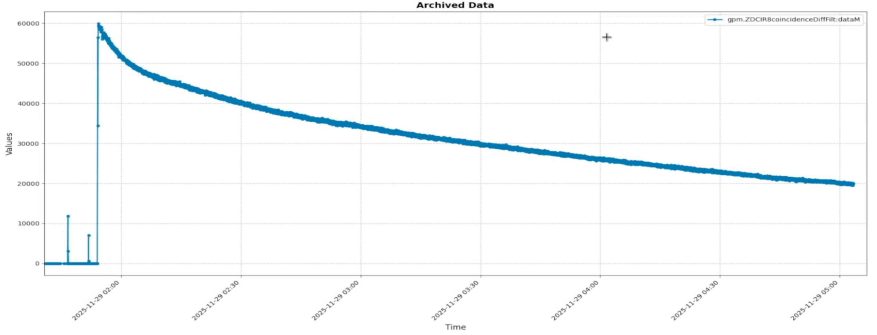
(All capabilities (Tools) here are corresponding to the tools in the Tool Tier in the new version of Osprey)

(Latest version of Osprey support web terminal UI which is much better)

Brookhaven National Laboratory Cornell University

```
**How can I assist you with RHIC control system information or operations today?**
👤 You: what't the current beam luminosity in RHIC ZDC?
```

Find the ADO and parameters and get and display or plot data...



Give user more options to use system programs:

```
Found 1 ADO Parameters.
You can open pet or Gpm for your ADO parameters:
   1. Open pet page for these ADOs
   2. Open Gpm for these ADOs
   3. Skip
Select an option if you want:  (1-3 or the option name): 
```

```
The full dataset contains all luminosity measurements for November 29, 2025. For detailed analysis (averages, peaks, trends), please specify
what statistical analysis you'd like performed on this data.
👤 You: get elog data from elog book RHIC on the same day
```

```
--------------------Start Ollama AI data analysis ...
-------- Gemma3 AI Model response: Here's a summary of the data, highlighting issues and key points, within approximately 300 words:

The provided data represents a log of activities during the execution of RHIC experiments, specifically focusing on Fill 36414 and the subsequ
ent rebucketing process. Initial ramp operations for both Blue and Yellow beams exhibited high efficiencies, with Blue achieving 98.3% and Yel
low 95.4% ramp efficiency during the first ramp. Ramp 2 saw both beams reach 100% efficiency.

Several vacuum issues were noted, primarily through the use of RatView and RatVacuum systems. The data indicates continuous vacuum maintenance
, highlighting the importance of IR6, IR2, and GammaJump RF systems for maintaining beam quality. A notable issue arose with wfgman testS repo
rting no problems related to power supplies.

A key activity was the rebucketing process, which was performed using both the Yellow and Blue beams, indicated by multiple entries mentioning
"Rebucketing" and related processes. The MCR team was actively involved in correlating the data and implementing these changes as evidenced b
y the repeated mentions of "FillPattern" and "Correlation"

There were instances where YS2 events for rebucketing were re-triggered, suggesting ongoing troubleshooting or adjustments to the rebucketing
strategy. Additionally, multiple 'RatView' and 'RatVacuum' events indicated ongoing operations and diagnostics relating to beam quality and va
cuum levels.

A significant point is the consistent monitoring of beam efficiencies, demonstrating meticulous control and adjustments throughout the experim
ents. The final log entry indicates a shift to physics operations, signifying the completion of the rebucketing and associated procedures.
```

Options for other actions: such as email data out, send notification, change data with system tools ...

Brookhaven National Laboratory

Cornell University

The AI Model Selection Matters

We tested several AI models with osprey-framework agents:

- Cloud-Based Model (Provided by the MOAT seed project): Anthropic Claude Haiku 4.5
- Locally Installed Open-Source Models:
  - Gemma3 (via Ollama server or vLLM server)
  - Llama3 (via Ollama server)
  - GPT-OSS-20B (via vLLM server)

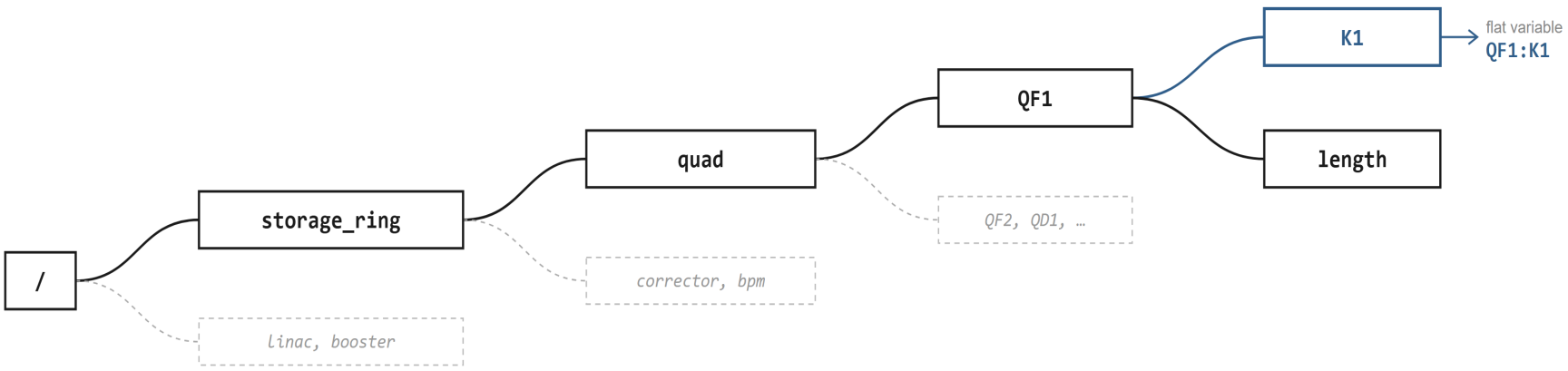
Although the Osprey framework supports a wide variety of AI models, cloud-based models such as the Anthropic Claude models are preferred to leverage the framework's full features and capabilities.
To utilize locally hosted open-source models effectively, we need hardware configurations with higher computational power and VRAM capable of running high-end, high-reasoning models.
Lightweight open source models like Gemma 3 perform well for text generation but lack the processing power required to drive the Osprey framework's complete features and capabilities.

Things need more work ...

- **Integrate internal domain RAG knowledge bases through an MCP server** so the Osprey-driven AI model and agent can answer user queries based on model's pre-trained knowledge and internal domain knowledge.
- **Upgrading the Agent for Osprey Framework V2026.6.2+**
  - Setup a dedicated Anthropic Claude account and safe model API licenses (e.g. AmSC).
  - Convert capabilities/tools to MCP server tools or other harnesses.
  - Develop a more robust and web supported UI for the end users.
- **Advanced Methods for better ADO and Parameter Discovery:**
  - We utilize the RHIC machine tree data alongside our system configuration database, applying both keyword and semantic search methodologies to locate target ADOs and parameters.
  - Due to the massive scale of the control system (>2.7M points), natural language queries alone often lack the resolution required for precise identification. Consequently, an interactive user workflow is frequently used to pinpoint exact ADO and parameters from the big machine tree.
  - New searching method has been developed and is slated for testing.

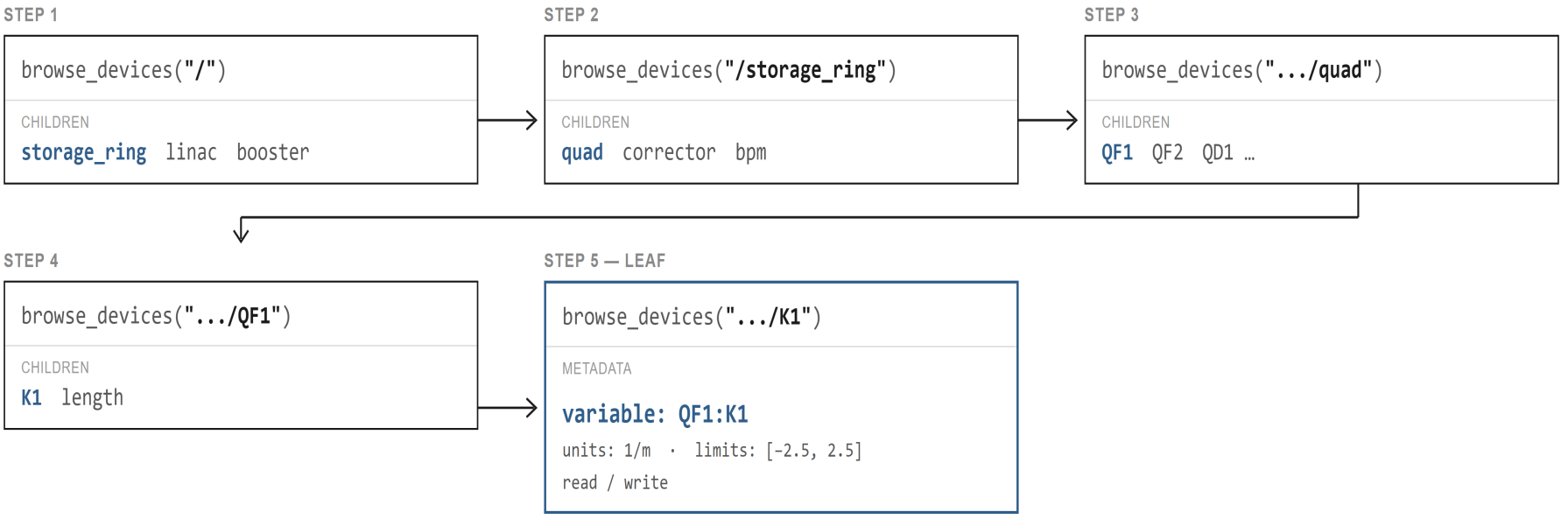
Agentic search ADO finder

- Agents can autonomously explore a hierarchical database
  - Navigate tree branches



Agentic search ADO finder

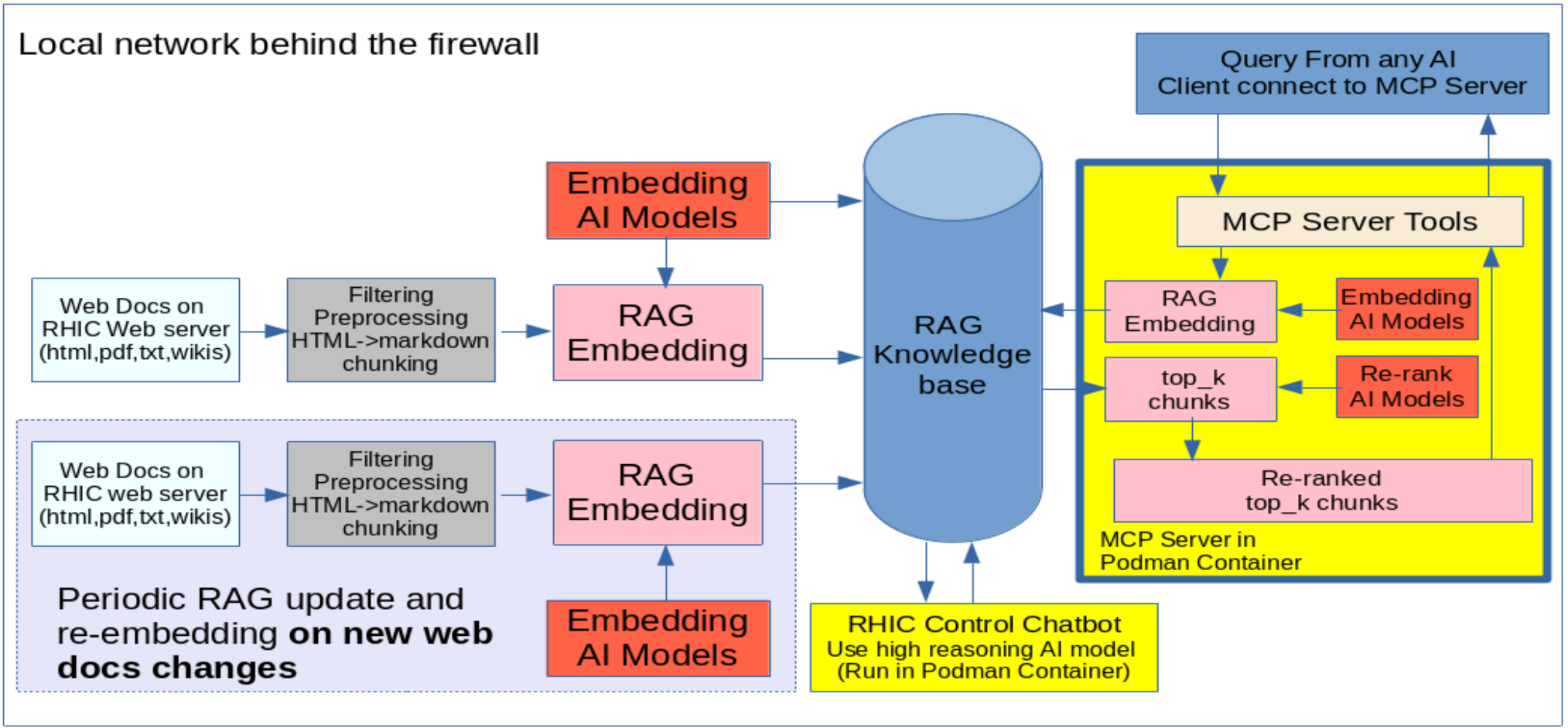
- Agents can autonomously explore a hierarchical database
  - Navigate tree branches
  - Search for target ADOs iteratively with a single tool



RAG Knowledge Base and MCP Servers

- Osprey agent depends on domain based tools for local control systems:
  - AI models must understand local control system domain knowledge to answer questions or execute instructions accurately. Therefore, we need to build RAG-based domain knowledge bases.
  - Executing control tasks requires tools that can interact with local control systems. An MCP (Model Context Protocol) server is an excellent option for this.
- An MCP server can operate as a sub-agent with its own dedicated AI model connection, allowing it to utilize a model tailored specifically to the task. In many scenarios, deploying a locally hosted, open-source AI model works well and significantly reduces overall token costs for the primary AI agent.

RAG Knowledge base and MCP Server



Summary

We have developed an experimental Control Assistant AI agent using the Osprey framework. We implemented and validated new capabilities for ADO and parameter discovery, real-time data access, historical data retrieval, local control system integration, elog data access and built domain RAG base and MCP server.

For ADO-based control systems, this experimental work walks us through the front-end stages of a typical AI control workflow: parsing user queries, planning tasks, connecting to the control system, identifying ADO and parameters, fetching telemetry, and performing control tasks. This brings us to a starting point where we can begin addressing real-world accelerator control problems.

As an early-stage experimental project, further upgrades and refinements are planned. Given the continuous fast evolution of LLM capabilities, the Osprey framework, and our internal codebase, we aim to enhance agent performance and deploy it to solve core system problems in the next phase.