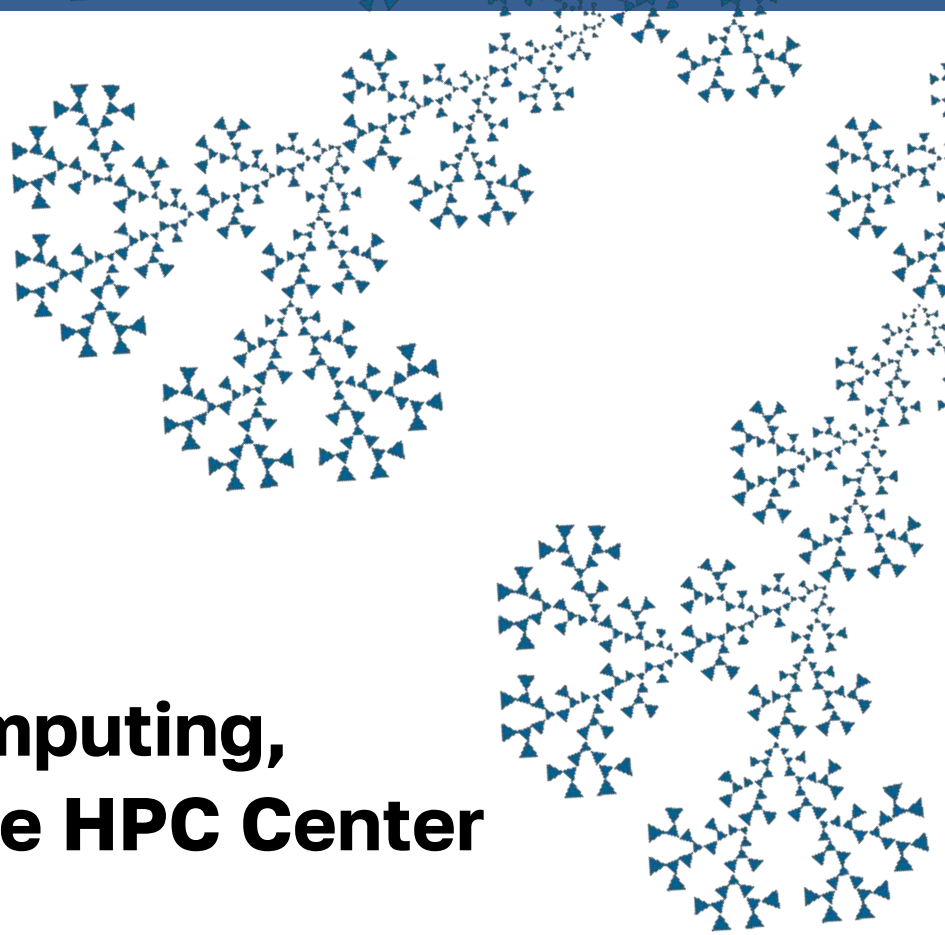


The Future of Discovery: Converged Computing, Agentic Science, and the Evolution of the HPC Center

Daniel Milroy, Vanessa Sochat
Lawrence Livermore National Laboratory

January 14, 2026

LLNL-PRES-2014740
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

The Post-Exascale Crisis of Complexity

Our science demands orchestration, automation, and infrastructure.

	Traditional HPC	Converged HPC Center
Workload	Monolithic batch simulation	Coupled AI/ML Simulation Ensembles
Resource Model	Static Node Allocation	Dynamic, Elastic Resource Graphs
Management	Imperative (Fixed Configurations and Shapes)	Declarative (Desired Status)
Interconnect	Fixed, Administrator Managed	Scoped, Dynamic Environments
Discovery	Environment Modules / Manual	Agentic / Fully Autonomous

The Post-Exascale Crisis of Complexity

Our science demands orchestration, automation, and infrastructure.

	Traditional HPC	Converged HPC Center
Workload	Monolithic batch simulation	Coupled AI/ML Simulation Ensembles
Resource Model	Static Node Allocation	Dynamic, Elastic Resource Graphs
Management	Imperative (Fixed Configurations and Shapes)	Declarative (Desired Status)
Interconnect	Fixed, Administrator Managed	Scoped, Dynamic Environments
Discovery	Environment Modules / Manual	Agentic / Fully Autonomous

The Post-Exascale Crisis of Complexity

Our science demands orchestration, automation, and infrastructure.

	Traditional HPC	Converged HPC Center
Workload	Monolithic batch simulation	Coupled AI/ML Simulation Ensembles
Resource Model	Static Node Allocation	Dynamic, Elastic Resource Graphs
Management	Imperative (Fixed Configurations and Shapes)	Declarative (Desired Status)
Interconnect	Fixed, Administrator Managed	Scoped, Dynamic Environments
Discovery	Environment Modules / Manual	Agentic / Fully Autonomous

Much of the above is driven by the workload manager.

Flux Framework is up to the task.

Flux Framework is an advanced workload manager and resource orchestration suite designed for modern, complex, and heterogeneous high-performance computing (HPC) environments.

Backed by expertise

Flux has 25+ contributors, including principal engineers behind Slurm, and developers with expertise in container technologies, scheduling, workflows, and cloud computing.

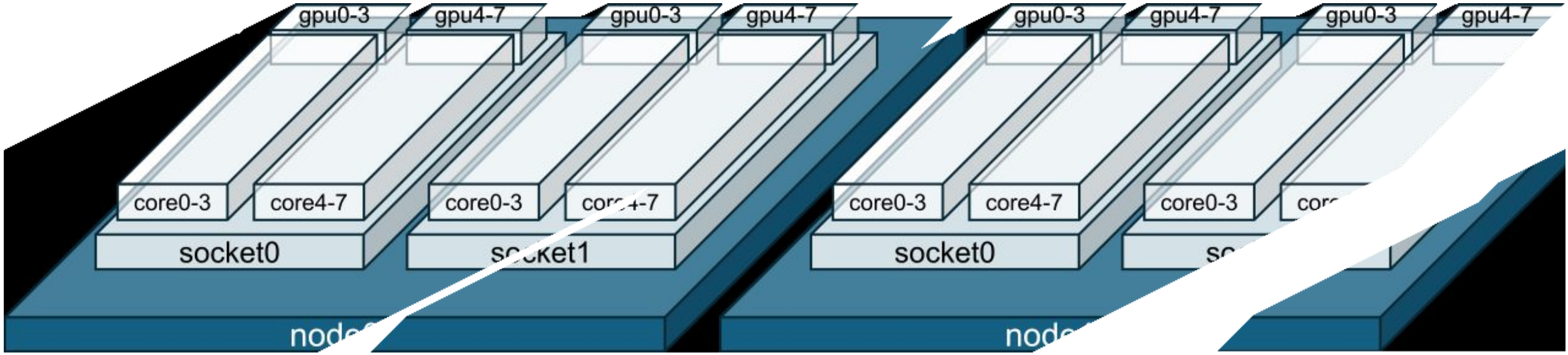
The Flux architecture addresses post-exascale challenges.

Flux architecture is designed to solve resource management challenges of the post-exascale era, bridging the gap between traditional batch processing and modern, dynamic workflows.

- **Exascale & Top 500**: NNSA exascale system El Capitan and other Top 500 supercomputers.
- **Converged Computing (HPC + Kubernetes)**: fully functional batch schedulers inside Kubernetes pods, bypassing etcd bottlenecks (Flux Operator).
- **Complex, heterogeneous workflows**: oriented for workflows where simulation and AI/ML components must coexist and share resources efficiently.
- **Graph-based resource model**: handles complex hardware topologies, ensuring the right job gets the right specific hardware (e.g., specific GPU interconnects) for existing and future hardware.
- **Systems research and innovation**: prototype novel scheduling algorithms.

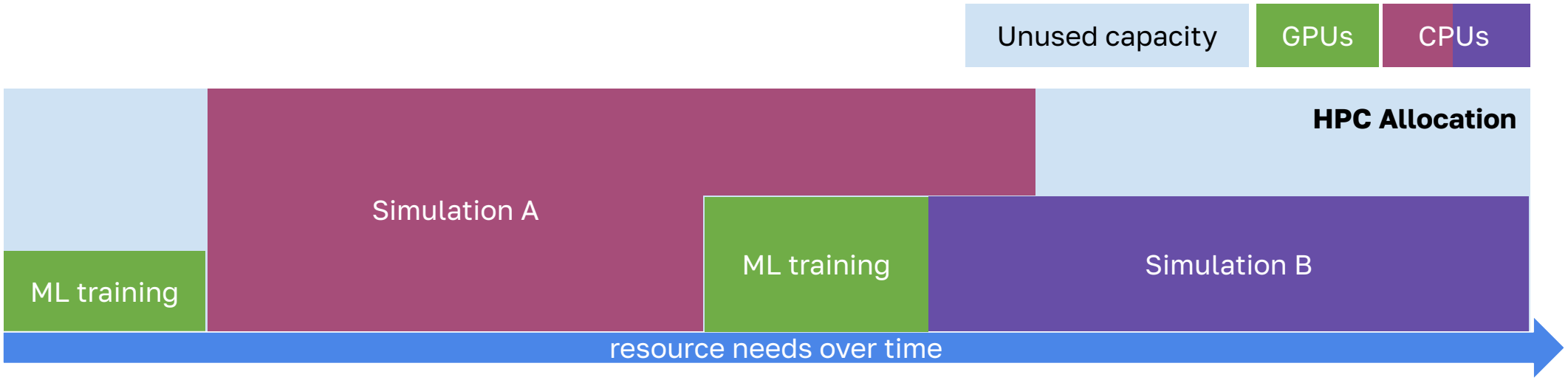
The Flux architecture addresses post-exascale challenges.

- Hierarchical scheduling breaks the traditional monolithic scheduler bottleneck by distributing scheduling decisions and managing resources at multiple levels.



The Flux architecture addresses post-exascale challenges.

- Designed to manage the complex, dynamic workloads required by coupled AI/ML and simulation research.



A traditional approach with a static allocation and underutilization of resources

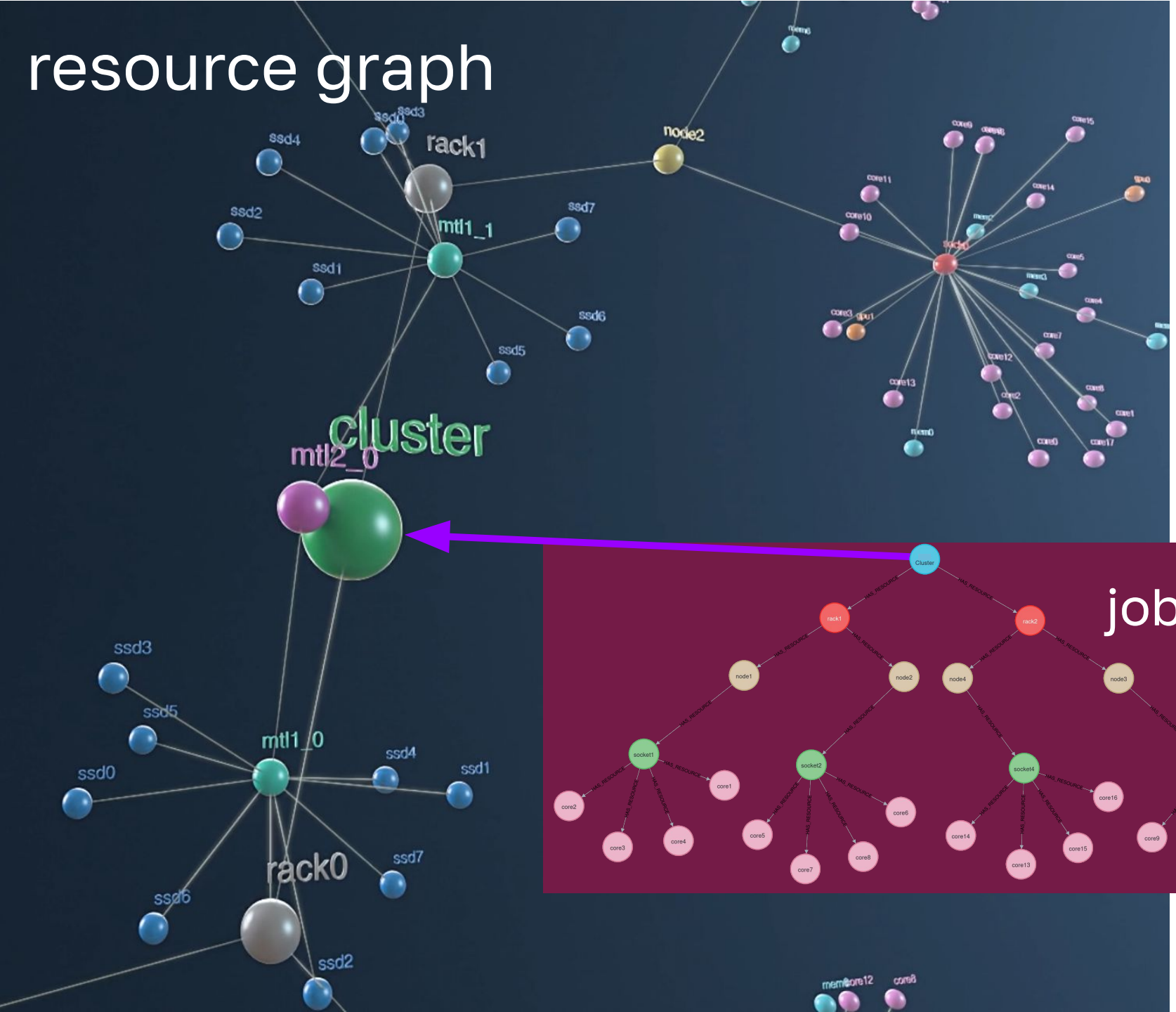
The Flux architecture addresses post-exascale challenges.

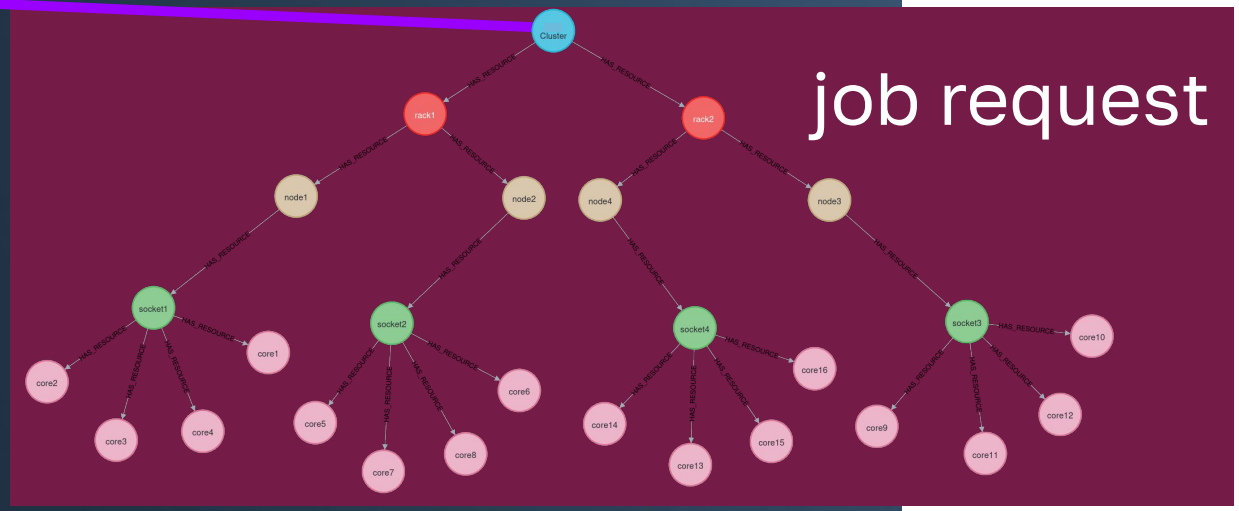
- Graph-based resource model to natively expresses and control complex hardware topologies and devices (like GPUs).

```
flux run -N 6  -n 216 ./app
```

```
version: 1
resources:
  - type: cluster
    count: 1
    with:
      - type: rack
        count: 2
        with:
          - type: slot
            label: myslot
            count: 3
            with:
              - type: node
                count: 1
                with:
                  - type: socket
                    count: 2
                    with:
                      - type: core
                        count: 18

# a comment
attributes:
  system:
    duration: 3600
tasks:
  - command: app
    slot: myslot
    count:
      per_slot: 1
```

resource graph



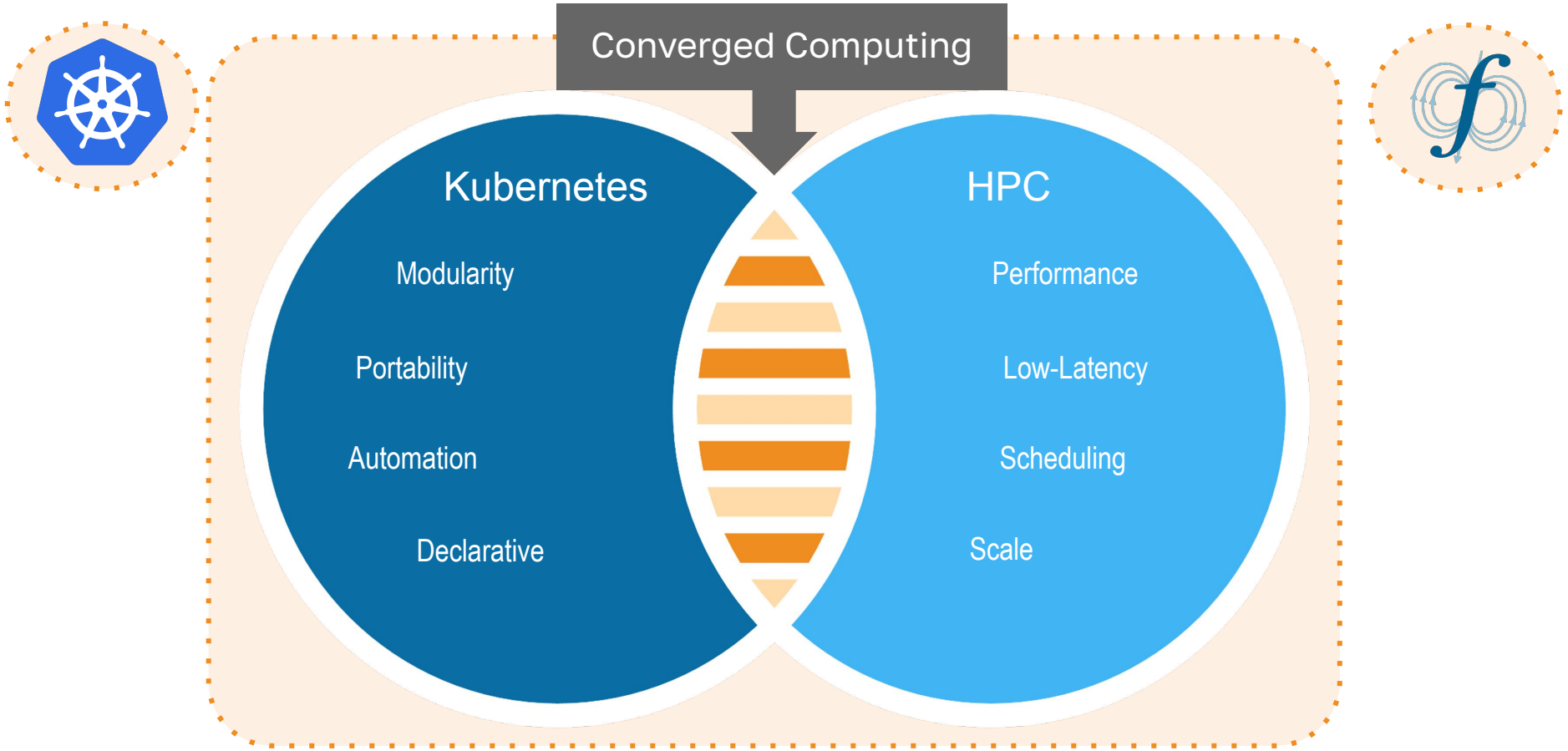


job request

The Flux directed graph model enables sophisticated **resource** expression and **job** specification.

Converged Computing brings together cloud-native and HPC.

It matches the needs of agentic, combined AI/ML and simulation-based workloads

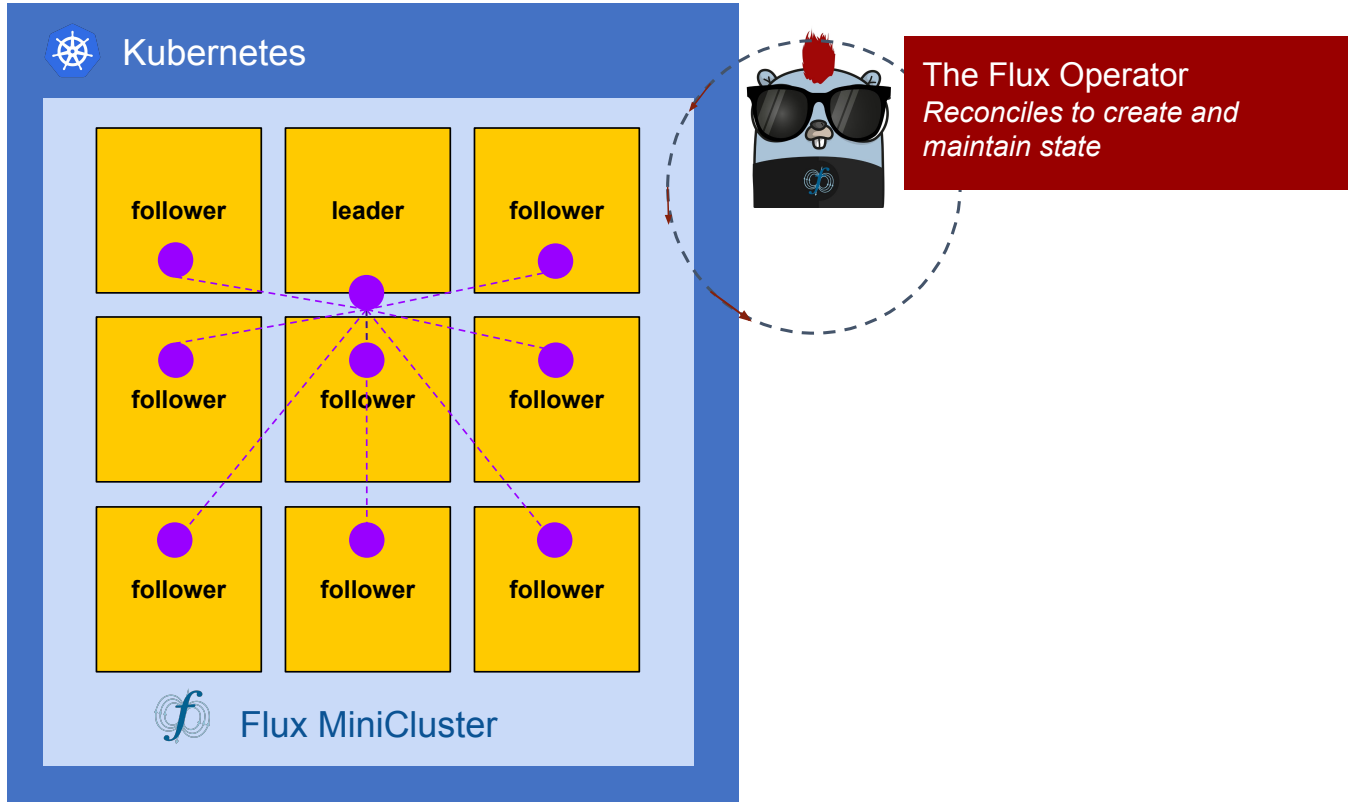


The Flux Operator

- Runs seamlessly on both HPC bare metal and cloud Native/Kubernetes environments.

HPC → Cloud

Deploying HPC technologies (low-latency scheduler) in cloud.



User-space Kubernetes and HPC

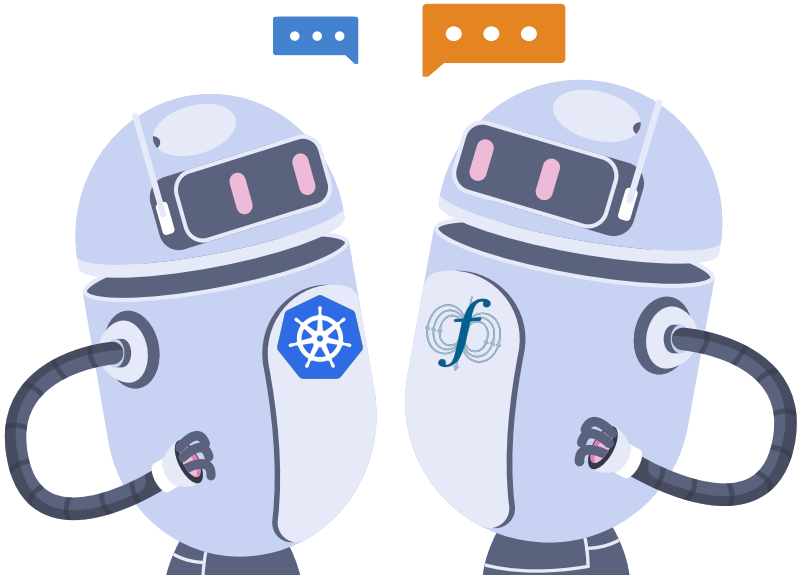
"The Bare Metal Bros"

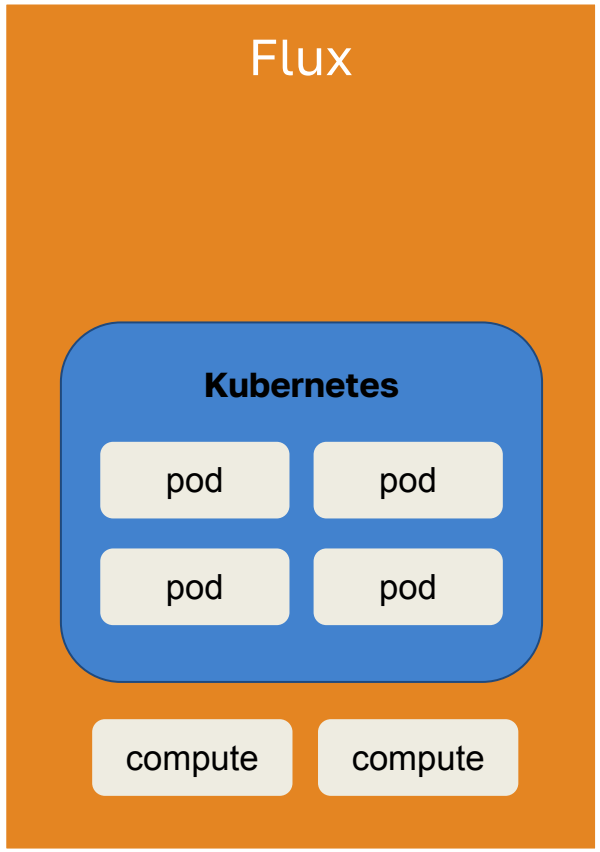
Deploy Kubernetes with Flux to get the best of both worlds - HPC simulations and services. We have achieved near-equal performance between bare metal and Kubernetes for environments, including:

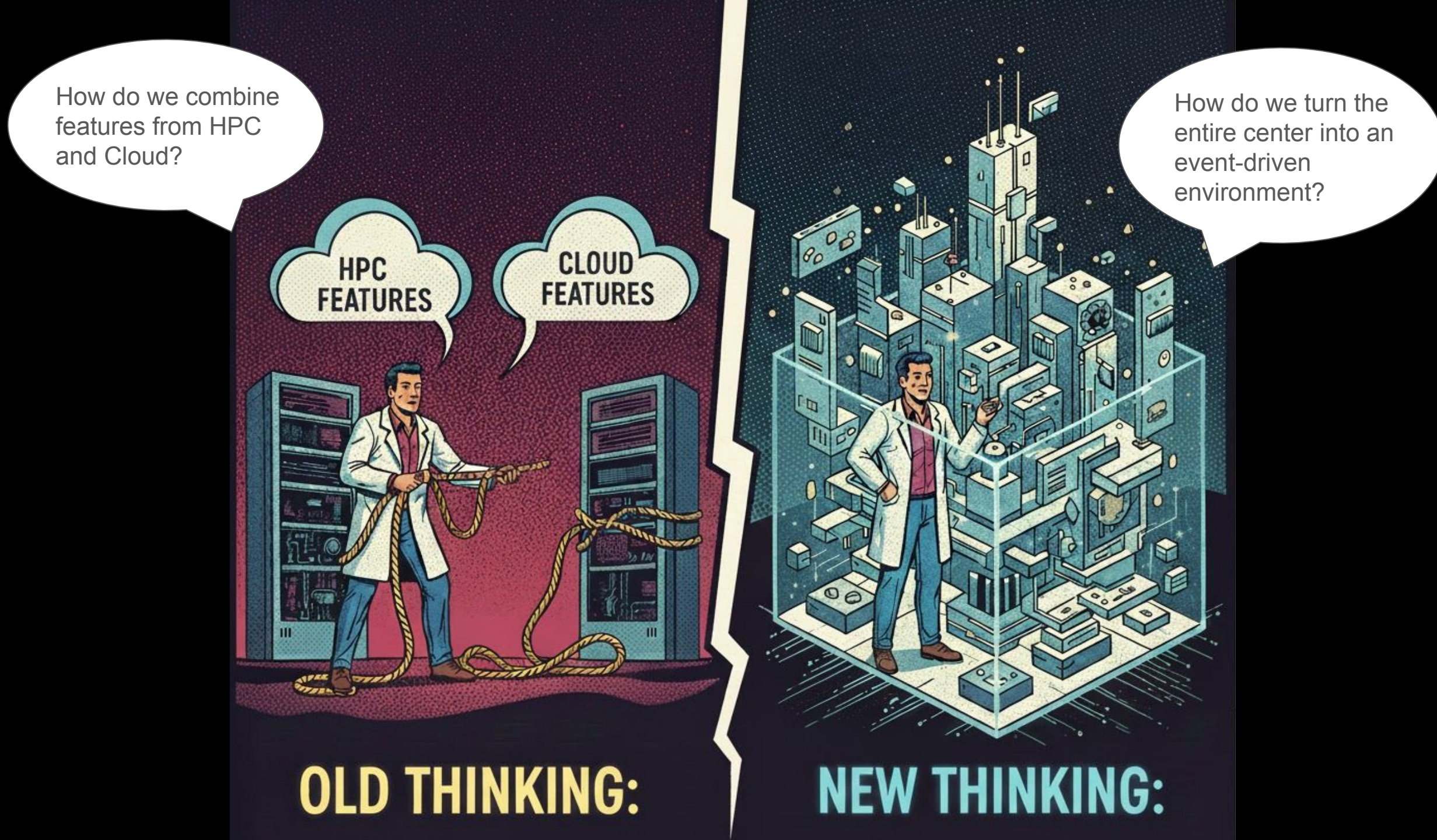
- AWS using the Elastic Fabric Adapter
- Azure using Infiniband
- Google Cloud (ethernet)
- On-premises LLNL (Corona)

Cloud → HPC

Deploying Cloud-native (automation, declarative, modularity) on HPC.






How do we combine features from HPC and Cloud?
HPC FEATURES
CLOUD FEATURES
OLD THINKING:
How do we turn the entire center into an event-driven environment?
NEW THINKING:

US Initiatives Seek to Accelerate Scientific Productivity through AI Innovation

The Future Generation High Performance Computing Center (FG-HPCC)

The Genesis Mission

American Science Cloud (AmSC)
Transformational AI Model Consortium (ModCon)

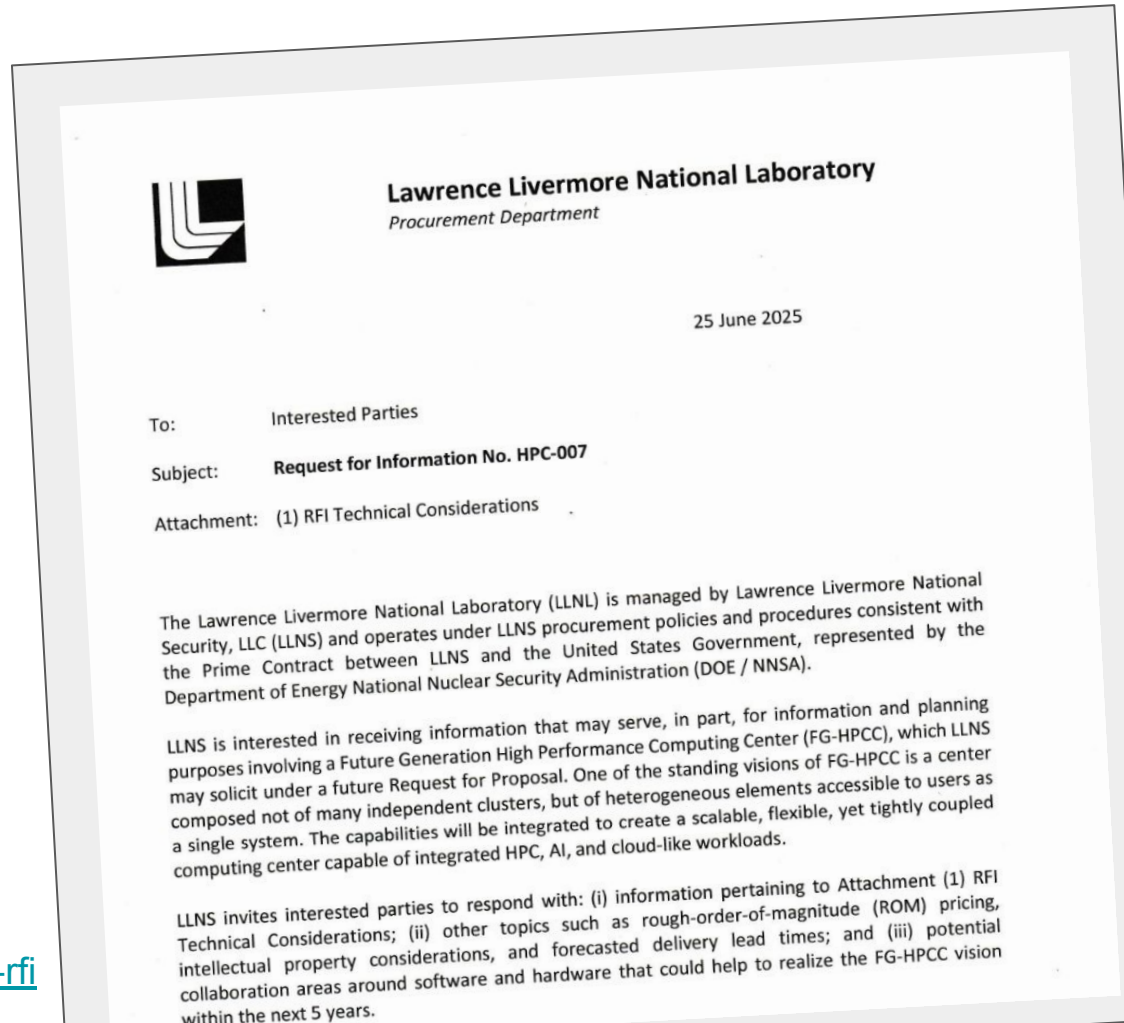
FG-HPCC

Future Generation High Performance Computing Center.

- LLNL put out a request for information for an HPC center of the future
- Timeframe ~2030, developed via a multi-vendor strategy

Requirements

- Single, unified system that includes compute, storage, networking, etc.
  - Performance of virtualization: approaching bare-metal
  - Secure enclaves and isolation
  - Cloud-native management interfaces
  - Open source stack



Lawrence Livermore National Laboratory
Procurement Department

25 June 2025

To: Interested Parties

Subject: **Request for Information No. HPC-007**

Attachment: (1) RFI Technical Considerations

The Lawrence Livermore National Laboratory (LLNL) is managed by Lawrence Livermore National Security, LLC (LLNS) and operates under LLNS procurement policies and procedures consistent with the Prime Contract between LLNS and the United States Government, represented by the Department of Energy National Nuclear Security Administration (DOE / NNSA).

LLNS is interested in receiving information that may serve, in part, for information and planning purposes involving a Future Generation High Performance Computing Center (FG-HPCC), which LLNS may solicit under a future Request for Proposal. One of the standing visions of FG-HPCC is a center composed not of many independent clusters, but of heterogeneous elements accessible to users as a single system. The capabilities will be integrated to create a scalable, flexible, yet tightly coupled computing center capable of integrated HPC, AI, and cloud-like workloads.

LLNS invites interested parties to respond with: (i) information pertaining to Attachment (1) RFI Technical Considerations; (ii) other topics such as rough-order-of-magnitude (ROM) pricing, intellectual property considerations, and forecasted delivery lead times; and (iii) potential collaboration areas around software and hardware that could help to realize the FG-HPCC vision within the next 5 years.

https://hpc.llnl.gov/fg-hpcc-rfi

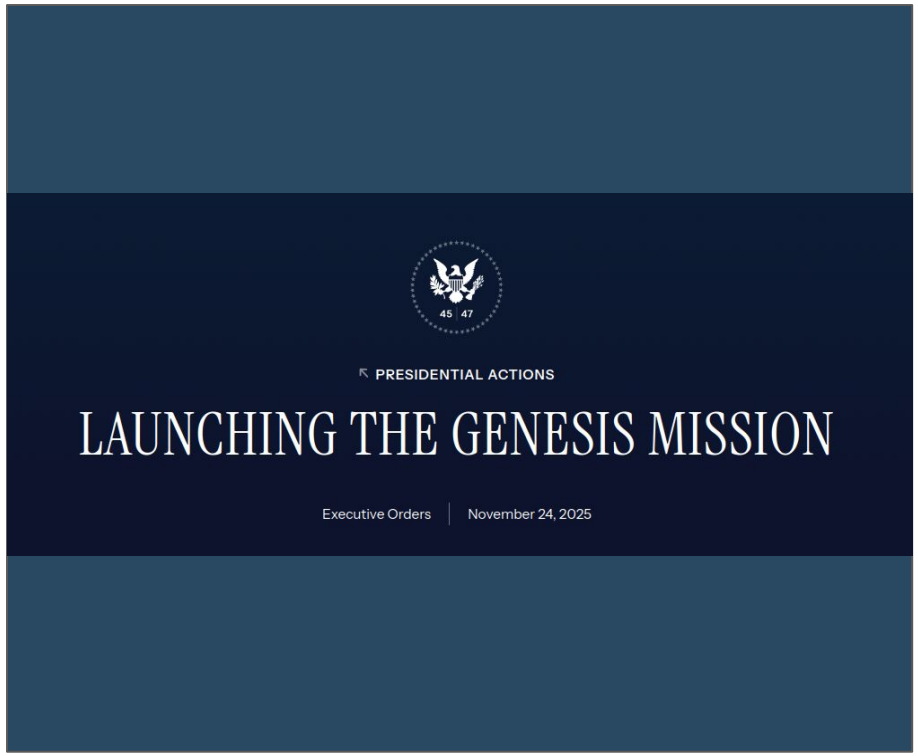
US Government Initiatives

A coordinated national effort to double scientific productivity within a decade through AI-accelerated innovation.

- American Science and Security Platform across 17 national labs
- Workflows include fusion energy, nuclear modernization, national security

- **American Science Cloud (AmSC):**
  - $40M initiative, software and hardware infrastructure AI science cloud

- **ModCon Consortium**:
  - AI Model Consortium ($30M) foundation models for science



Example Projects to Consider

Current projects span the application to infrastructure space

- **Orchestrated Platform for Autonomous Laboratories** (OPAL): https://opal-doe.org/
  - Uses agents and robotic systems to manage AI experiments (e.g., gene discovery)
- **IOWarp**: (https://grc.iit.edu/research/projects/iowarp/) Data management for scientific workflows.
- **Multi-Office particle Accelerator Team:**
  - AI foundation models and intelligent assistants for particle accelerator teams.
- **Agentic Lab**:
  - A multi-agent setup for knowledge retrieval and protocol design, specifically for organoid research.

Rise of Agentic Workflows in Science

Can an LLM convert a job specification?

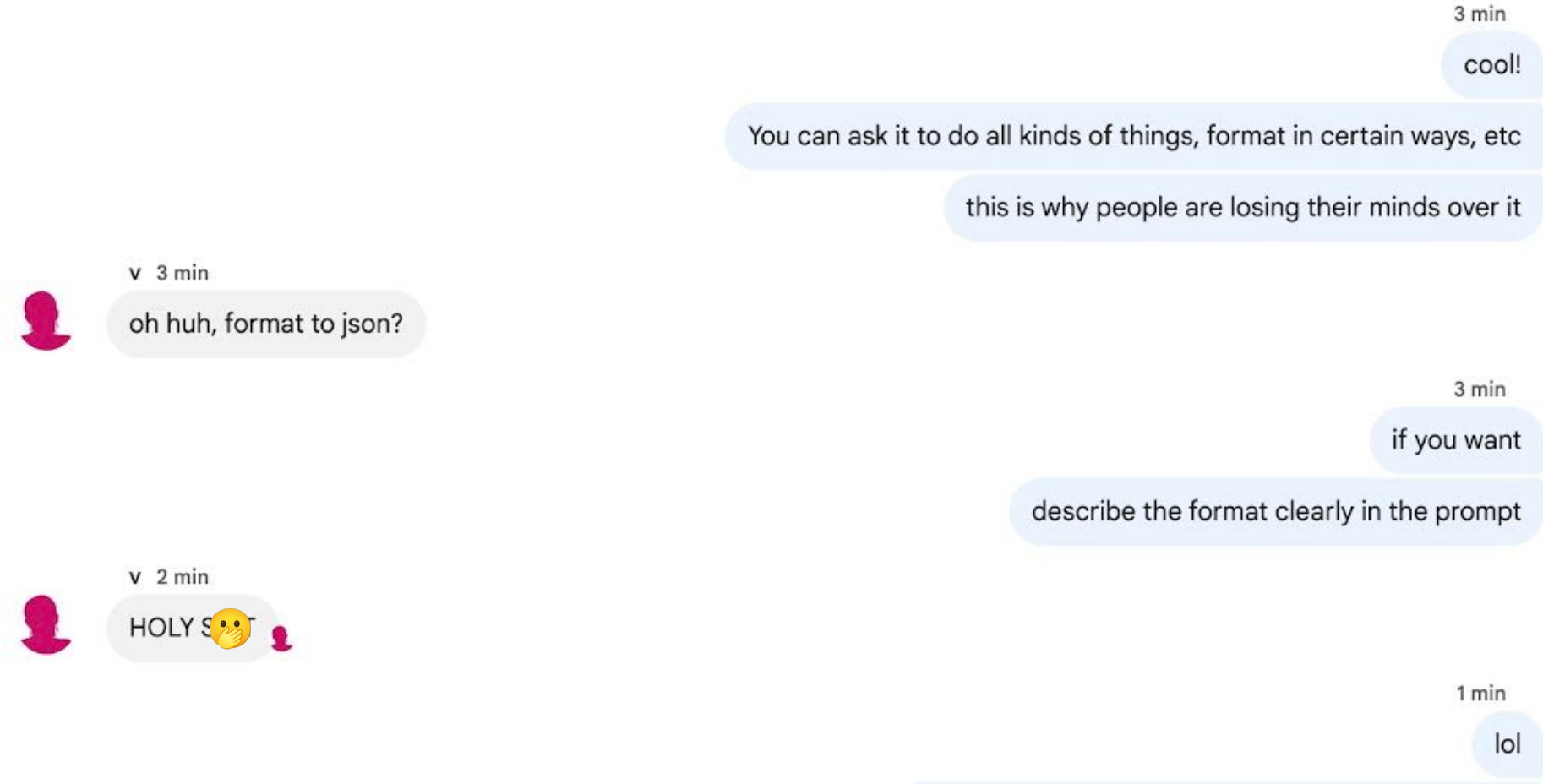
Slurm batch (sbatch job.sh)

```
#!/bin/bash
#SBATCH -N 3
#SBATCH -q regular
#SBATCH -t 05:00:00
#SBATCH -J prob-multiproc
#SBATCH -o logs/%x-%j.out
module load pytorch/v1.6.0
srun -n 96 -c 2 python $HOME/mldas/mldas/assess.py probmap -c $HOME/mldas/configs/assess.yaml -o
$SCRATCH/probmaps --mpi
```

Flux batch (flux batch job.sh)

```
#!/bin/bash
#FLUX: -N 3
#FLUX: -t 05:00:00
#FLUX: --job-name=prob-multiproc
#FLUX: --output=logs/{{name}}-{{id}}.out
module load pytorch/v1.6.0
flux run -n 96 -c 2 python $HOME/mldas/mldas/assess.py probmap -c $HOME/mldas/configs/assess.yaml -o
$SCRATCH/probmaps --mpi
```

"The moment Vanessa lost her mind over LLMs"

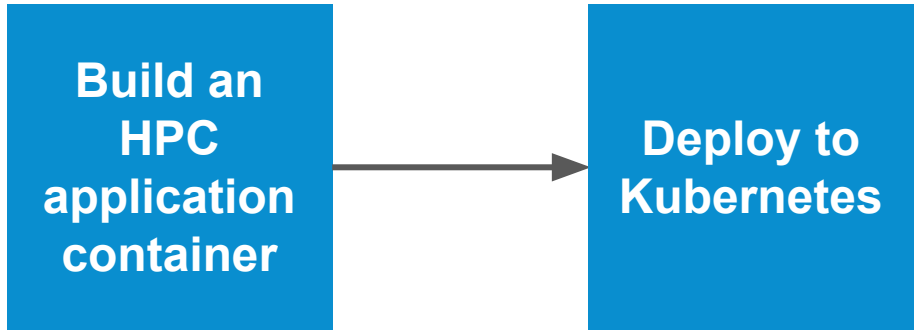


"Do the entire HPC scaling study"

"Do the entire HPC scaling study"

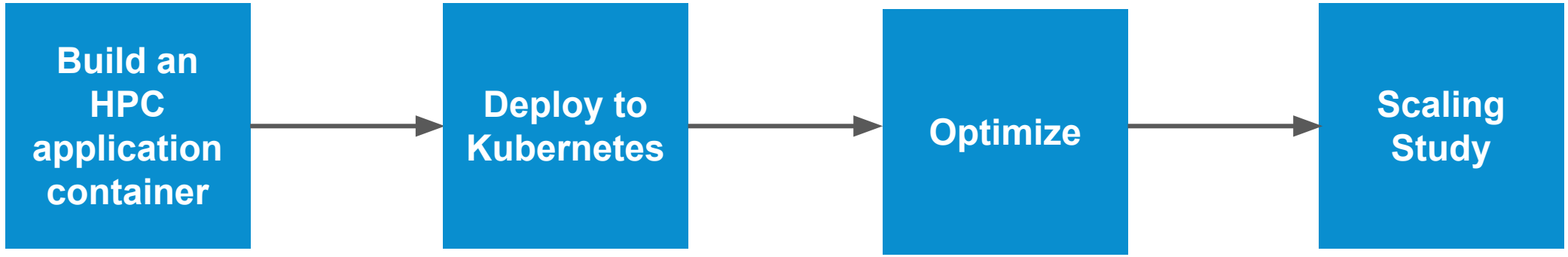
Build an HPC application container

"Do the entire HPC scaling study"



"Do the entire HPC scaling study"

"Do the entire HPC scaling study"



Agentic Orchestration of HPC Applications

A study using Google Gemini in Cloud

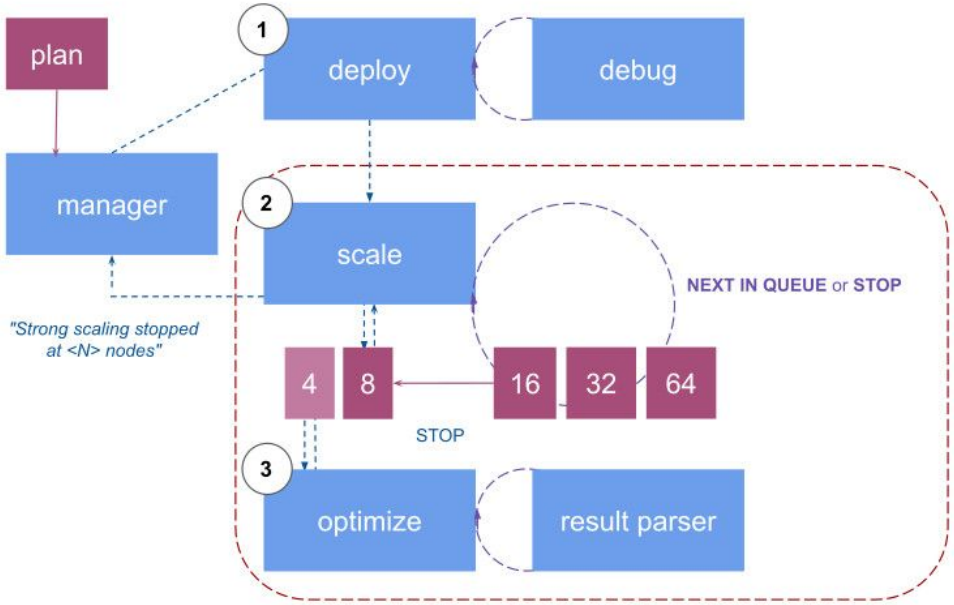
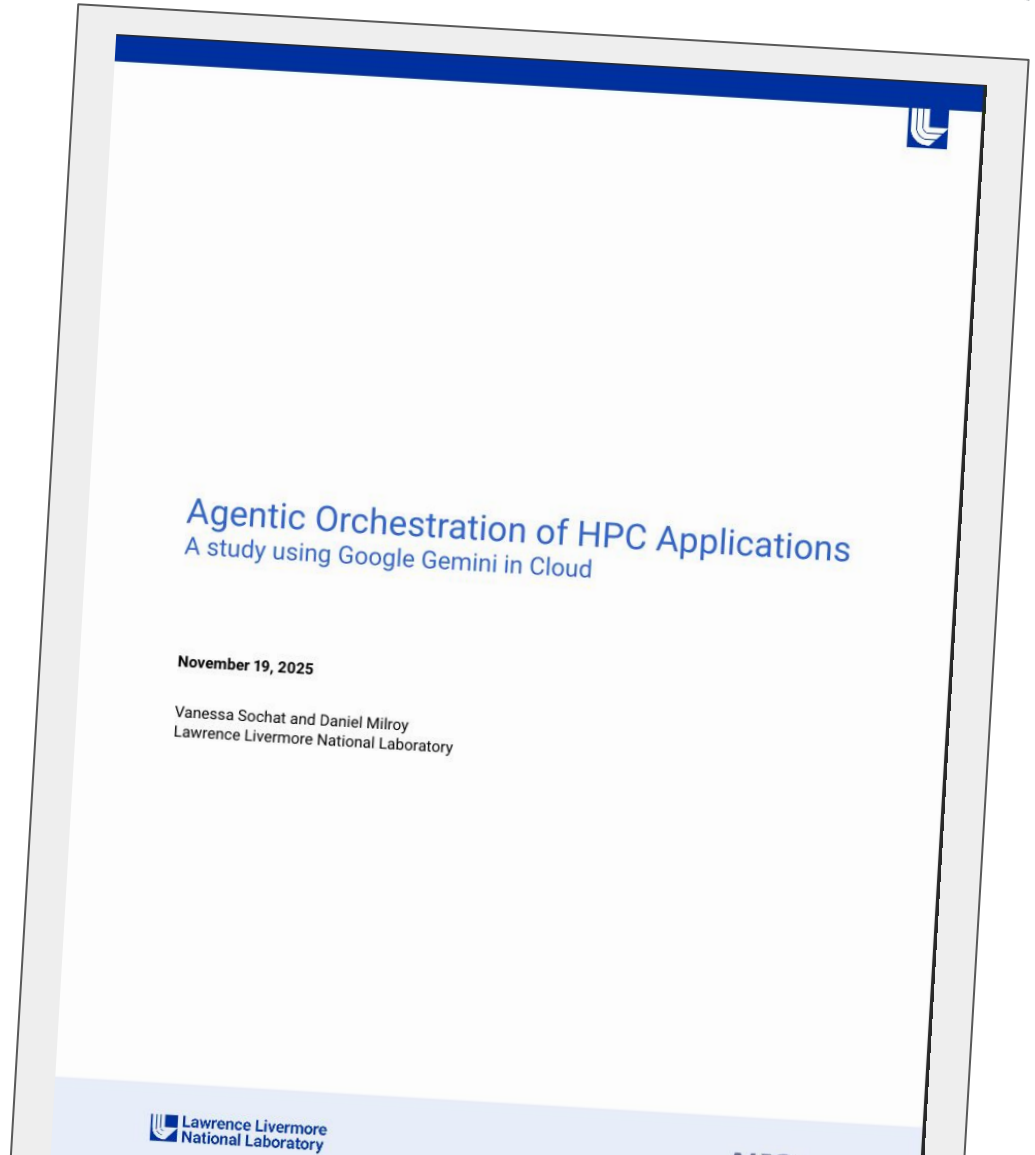


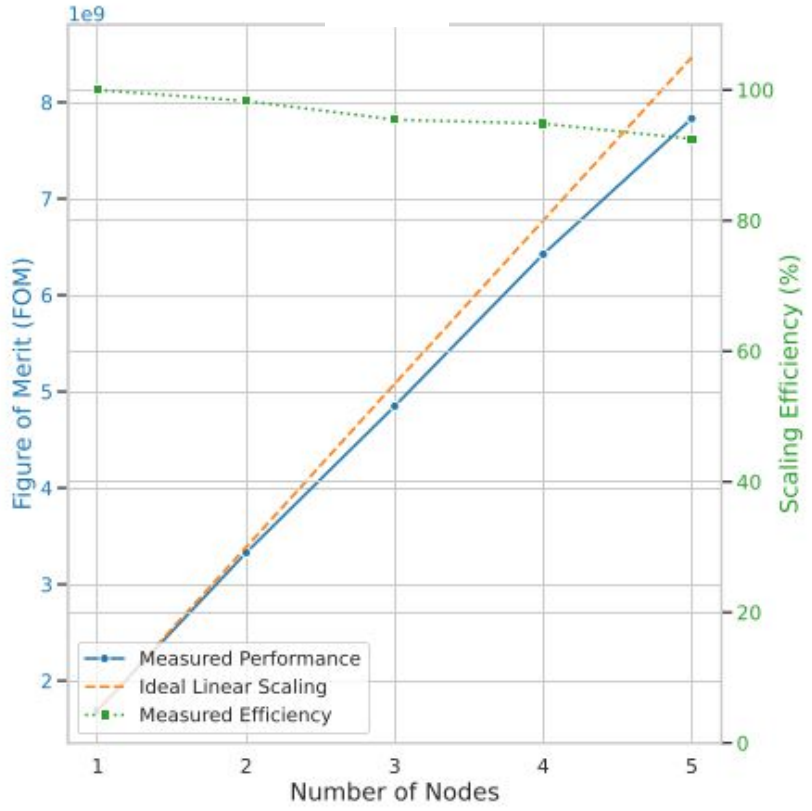
Figure 2: Scaling Study Agentic Team for a hypothetical scaling study for 4 to 64 nodes (maroon boxes). The deploy agent generates a manifest in a working state. The scaling agent receives instruction to start the study, and updates the prompt to deploy at each requested size. At the smallest size, the scale agent directs execution to the optimization agent, which decides on a configuration for subsequent sizes. At subsequent sizes, the scaling agent assesses the previous result, and decides to stop or proceed to the next size. Execution of subsequent sizes is done by the deploy agent. The scaling agent reports evidence and reasons for its decisions. Helper debug and result parsing agents assist primary step agents.



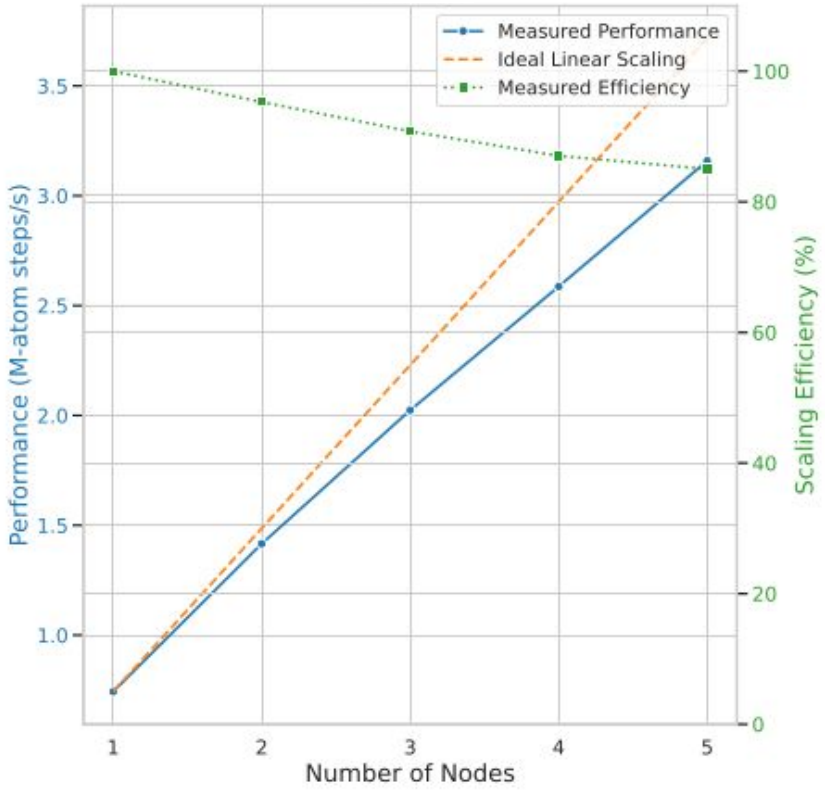
Application Performance and Scaling Efficiency

hpc7g.16xlarge instances on AWS

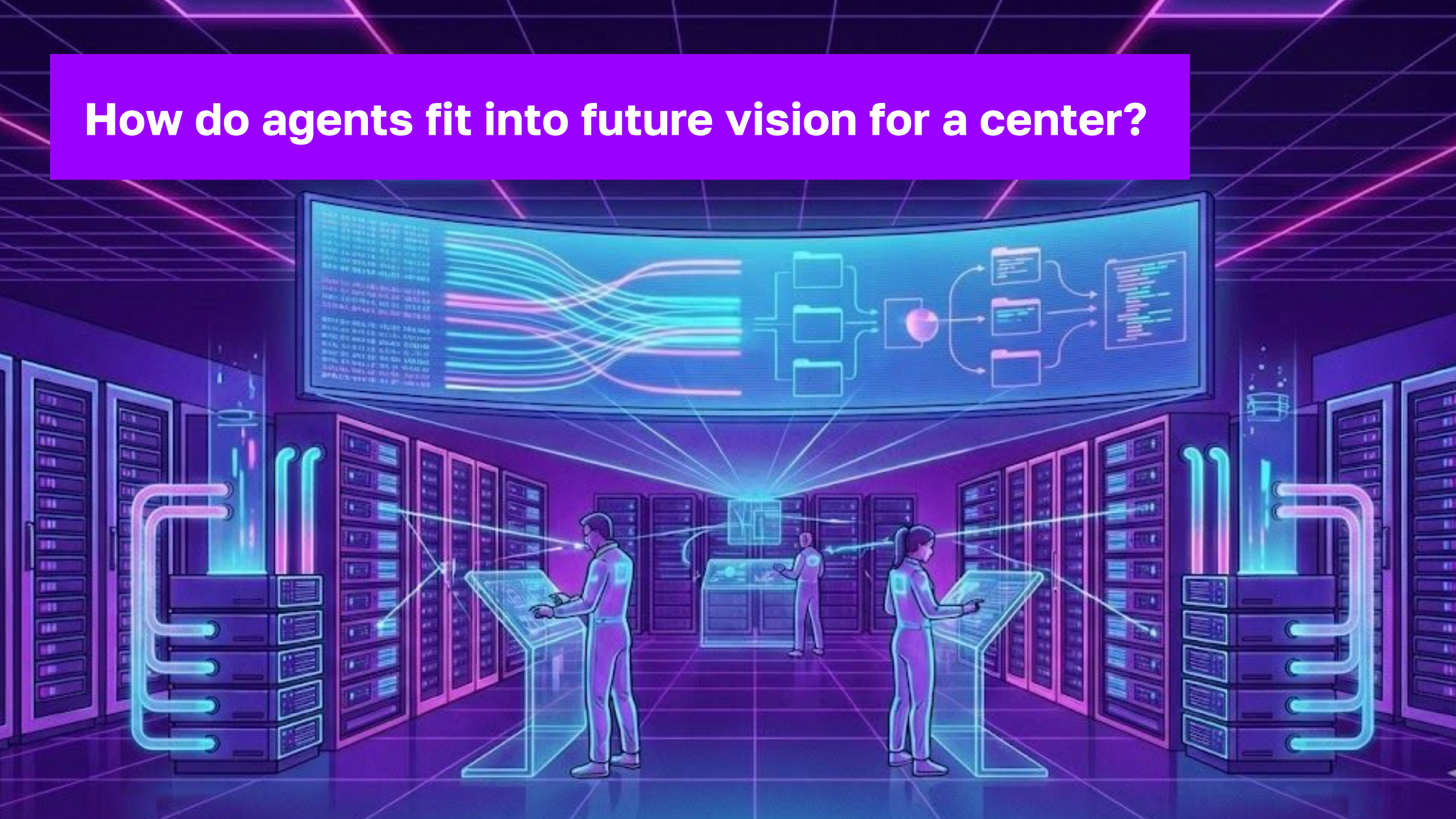
AMG



LAMMPS

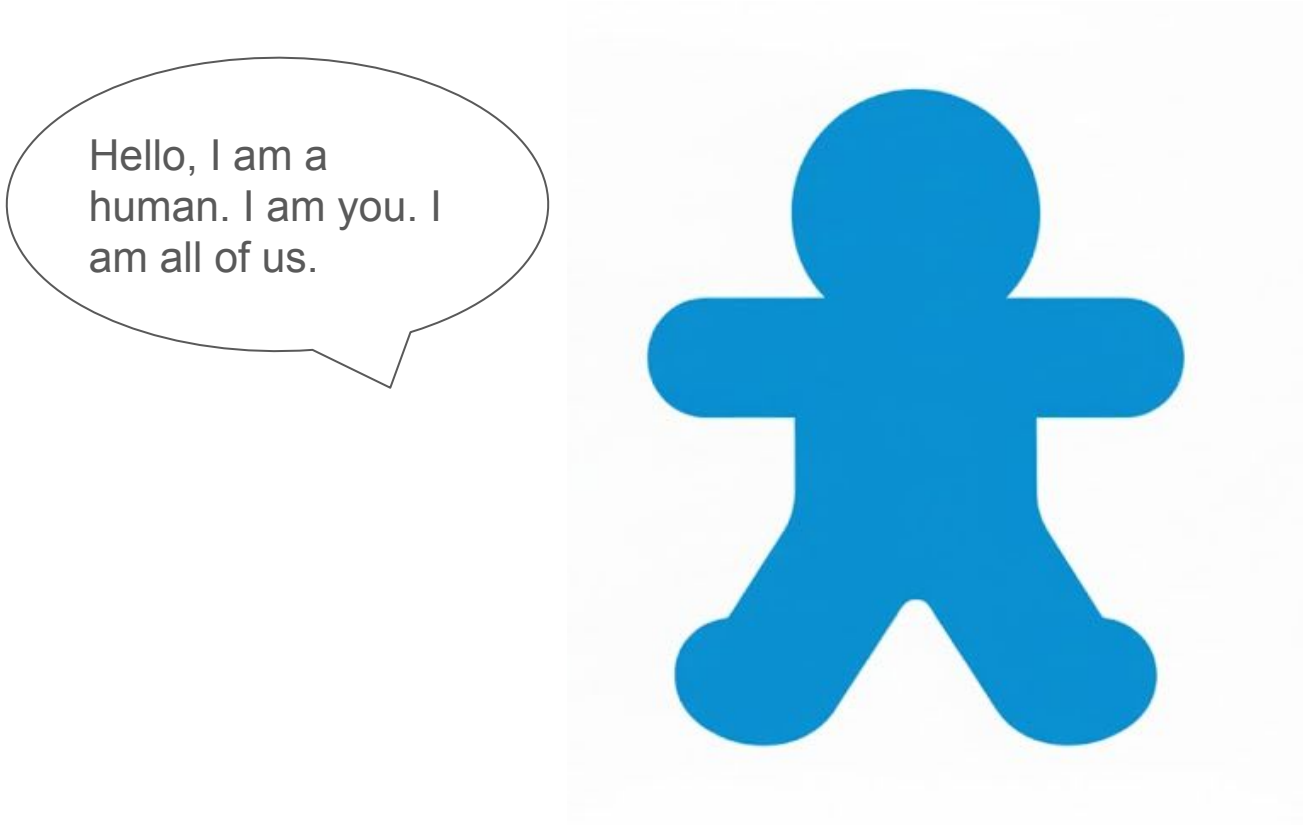


How do agents fit into future vision for a center?



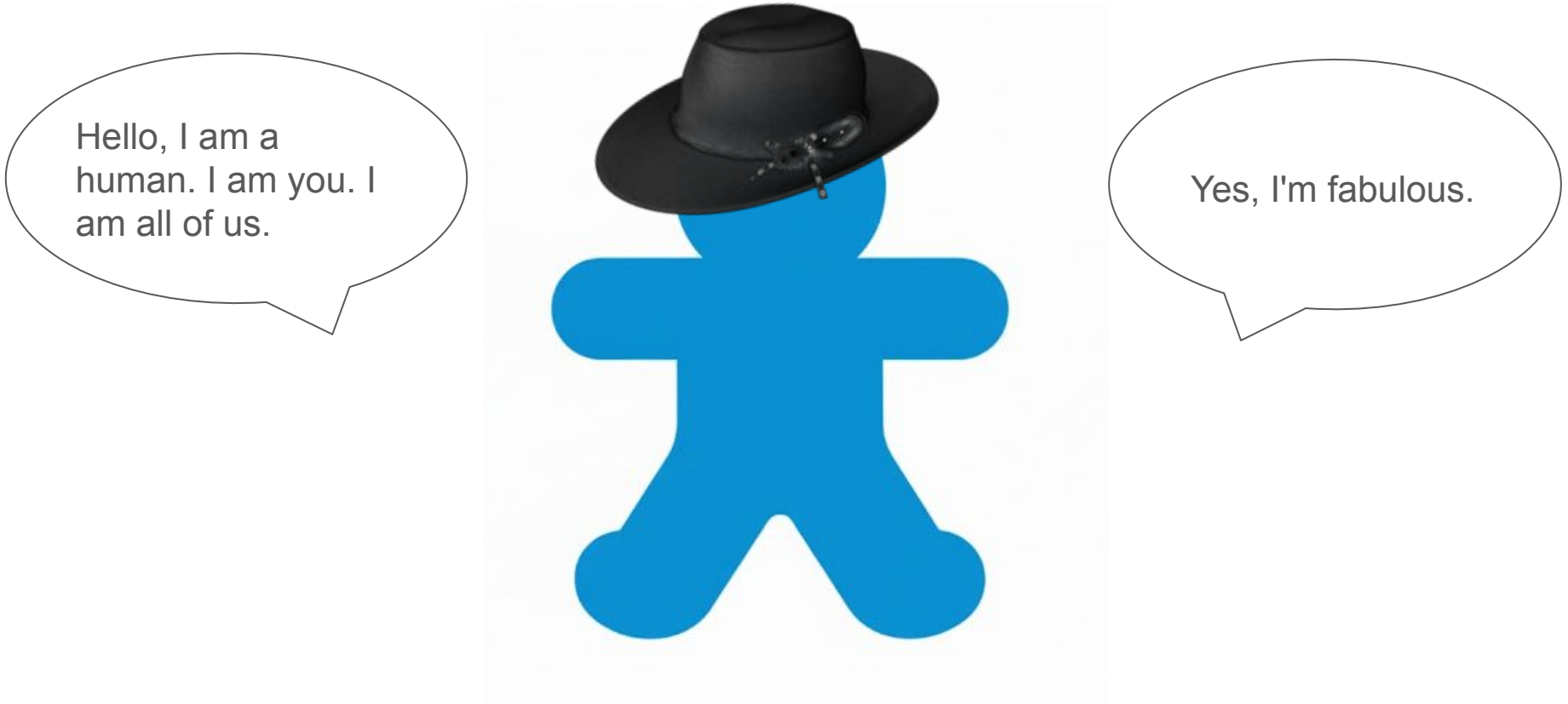
"Run an HPC Application"

Manually, as a human user.



"Run an HPC Application"

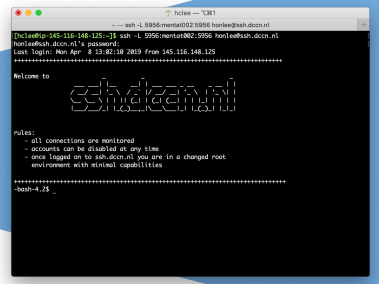
Manually, as a human user.



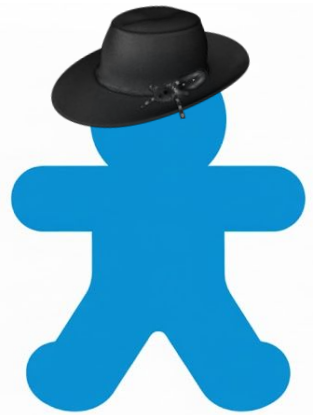
"Run an HPC Application"

1. Prepare application binaries

SSH



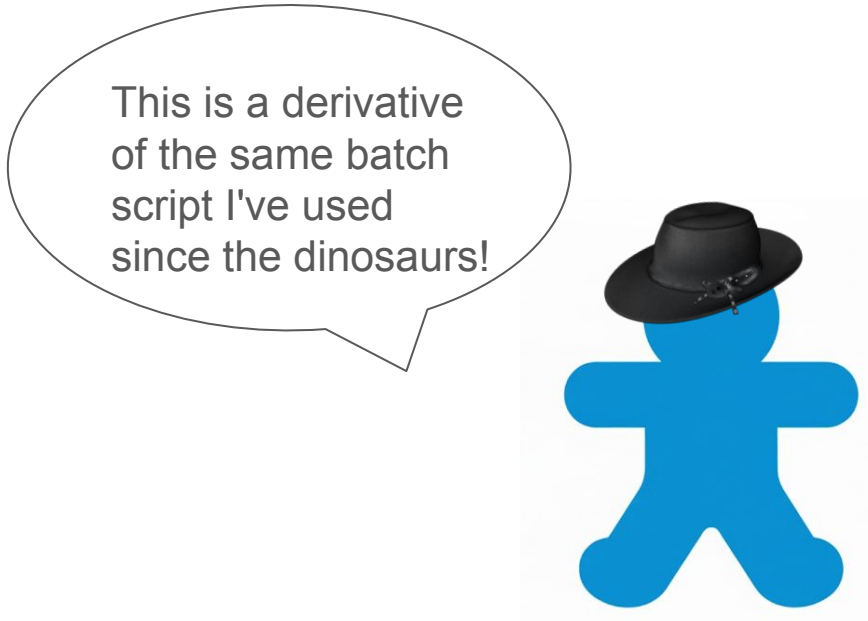
```
# Install with spack or easybuild
spack install gromacs

# Build from source
cmake ../

# Build a container
podman build -t gromacs .
```



"Run an HPC Application"

2. Write batch script

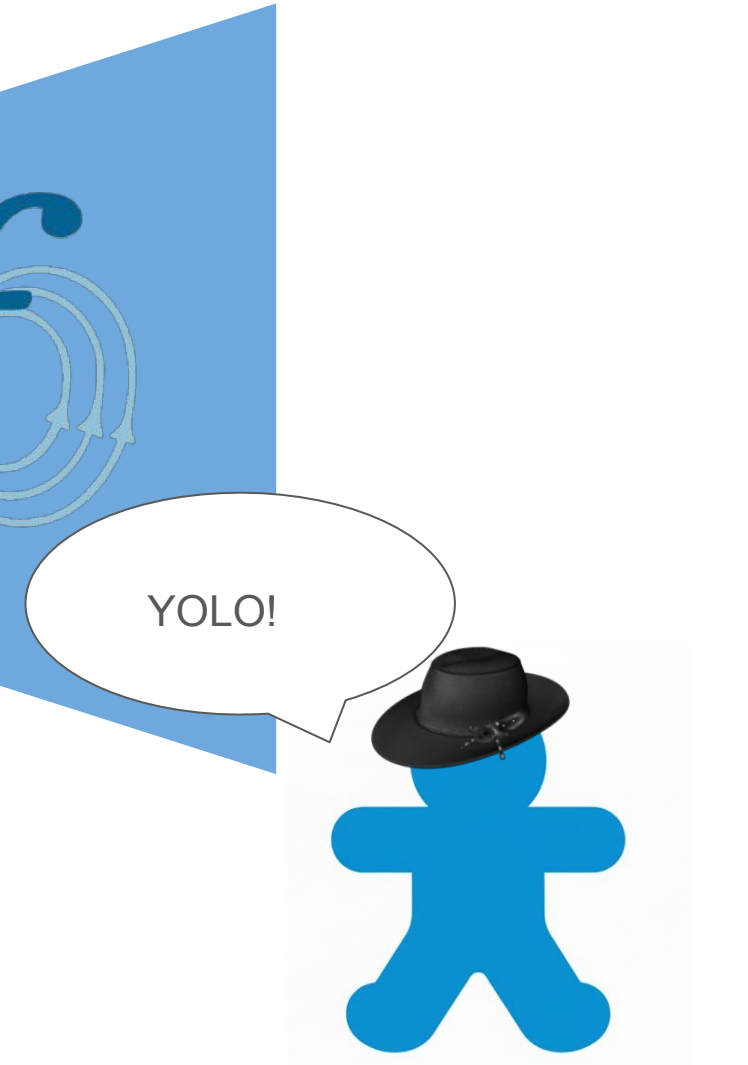
```
#!/bin/sh
# Submit using: flux batch <filename>
#flux: --job-name=lammps
#flux: --output='lammps.{{id}}.out'
#flux: --error='lammps.{{id}}.err'
#flux: -N 32                # 32 nodes
#flux: -n 3072              # tasks
#flux: -t 10m               # Time in minutes
#flux: -q pbatch            # Queue name

# Set environment variables for network
export FI_PROVIDER=efa

# Run the command
lmp -v x 128 -v y 128 -v z 128 -in ./in.reaxff.hns --nocite
```



"Run an HPC Application"

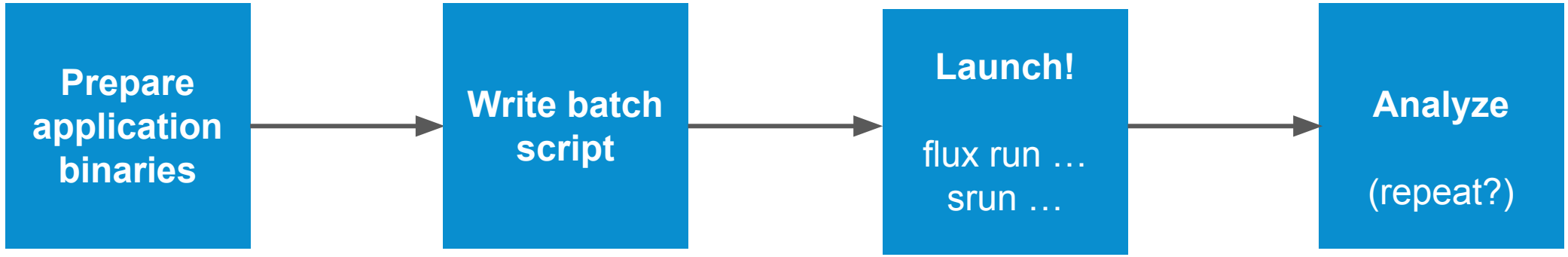
3. Launch the job!

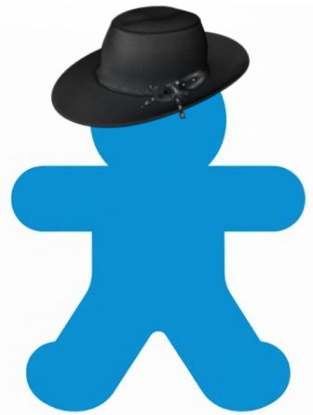
```
flux batch ./run-lammps.sh
```



"Run an HPC Application"

4. Monitor, re-queue and debug, and analyze to finish.





Rise of Agentic Workflows in Science

Agents are central to the HPC center of the future – active participants to almost every part of the scientific method.

- Perceive, plan, and accomplish scientific goals over long time horizons
  - **Gather** and **interpret** data from scientific simulations
  - Execute traditional tasks via **API calls** and learn from results
  - Use **foundational models** to reason about experiments

Rise of Agentic Workflows in Science

Agents are central to the HPC center of the future – active participants to almost every part of the scientific method.

- Perceive, plan, and accomplish scientific goals over long time horizons
  - **Gather** and **interpret** data from scientific simulations
  - Execute traditional tasks via **API calls** and learn from results
  - Use **foundational models** to reason about experiments

Agents use **MCP (Model Context Protocol)** and MCP servers with tools, prompts, and resources to carry out tasks. These are the API calls referenced above!

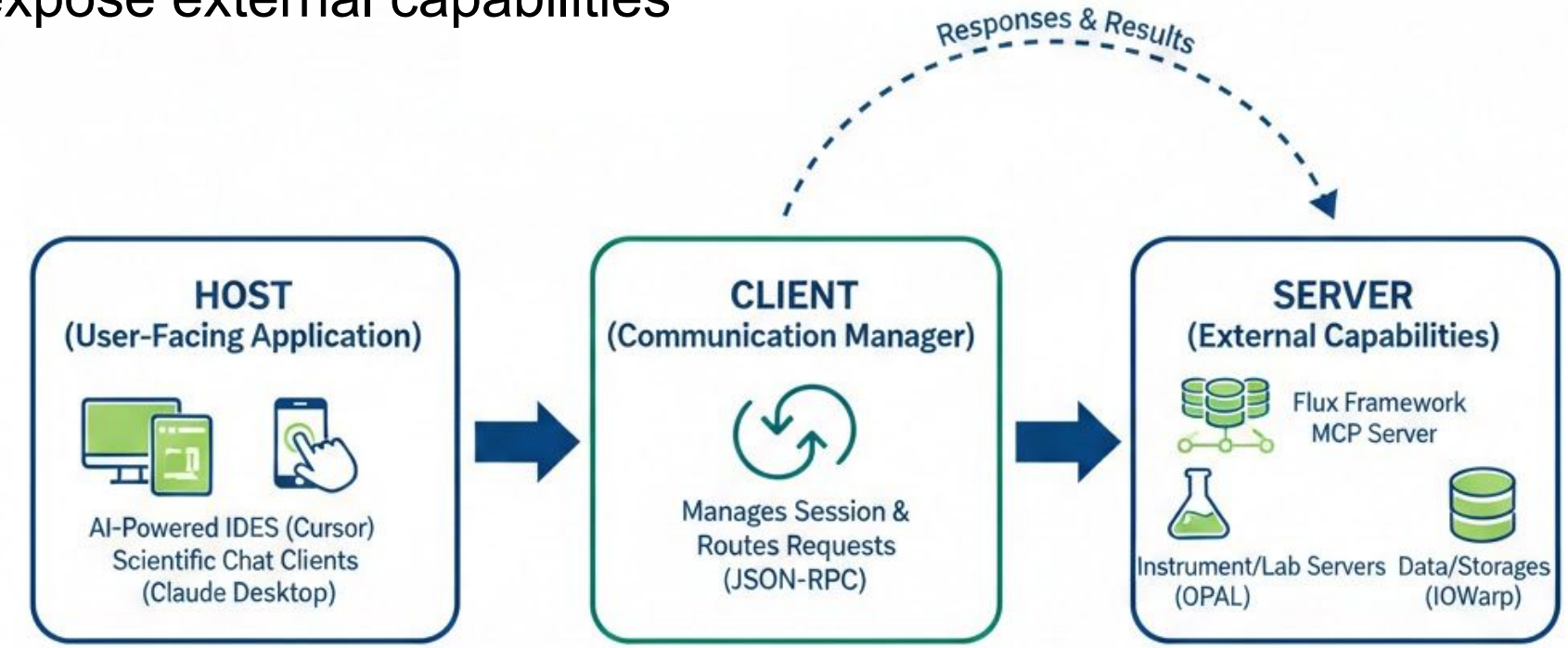
Are we just finally using APIs in HPC?

- Nothing special about MCP servers - they are tool APIs with prompt generation.
- The problem that was solved is not technical but cultural - agreeing to use the same standard.
- The AI/ML explosion has pushed centers to finally think about being programmatic.

Model Context Protocol (MCP)

Standard for models (agents) to connect to data, tools, and APIs securely.

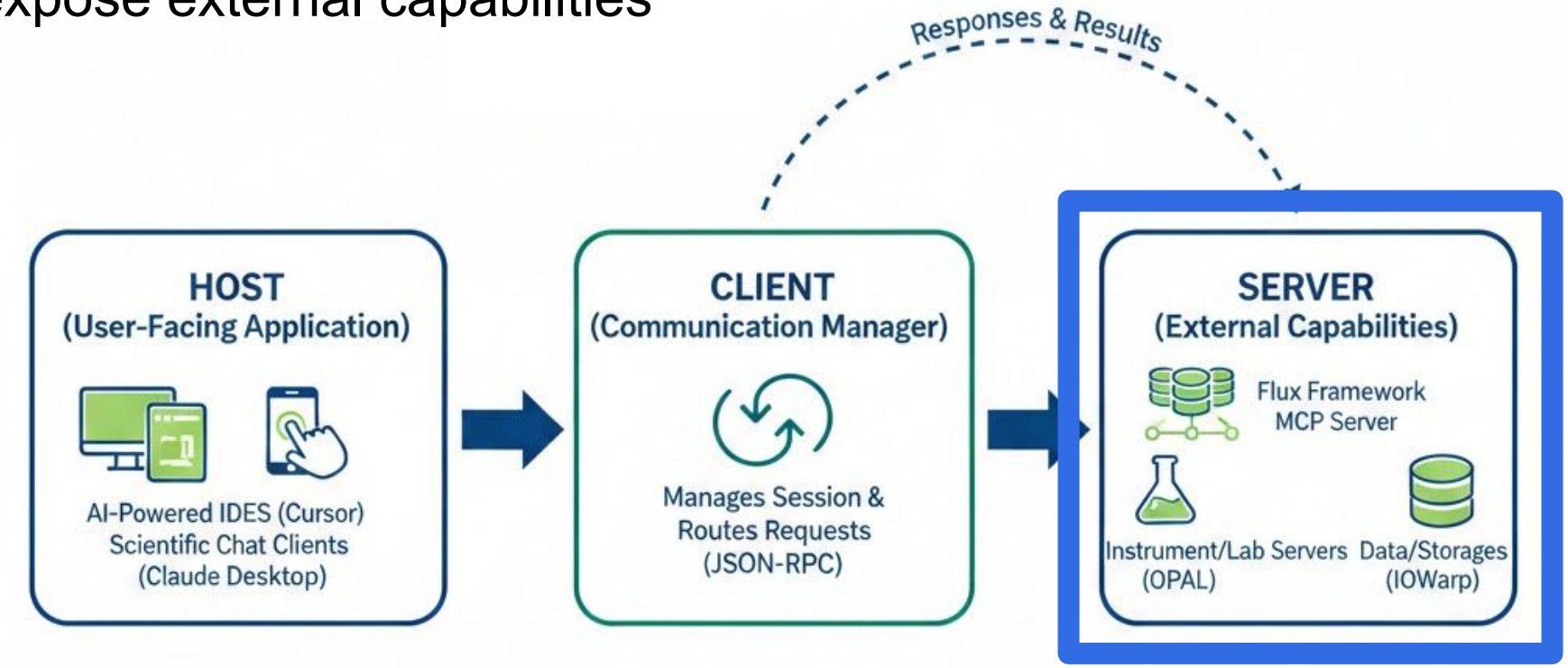
- **Hosts:** User-facing applications.
- **Clients:** Components that manage communication
- **Servers:** expose external capabilities



Model Context Protocol (MCP)

Standard for models (agents) to connect to data, tools, and APIs securely.

- **Hosts:** User-facing applications.
- **Clients:** Components that manage communication
- **Servers:** expose external capabilities



Model Context Protocol (MCP)

Standard for models (agents) to connect to data, tools, and APIs security.

MCP Server

Flux tools
submit job
job info
…
cancel job

Build tools
spack install…
podman build
…
cmake ../

Resources (data)
and

Prompts
I am a debugging agent…

Model Context Protocol (MCP)

Standard for models (agents) to connect to data, tools, and APIs security.

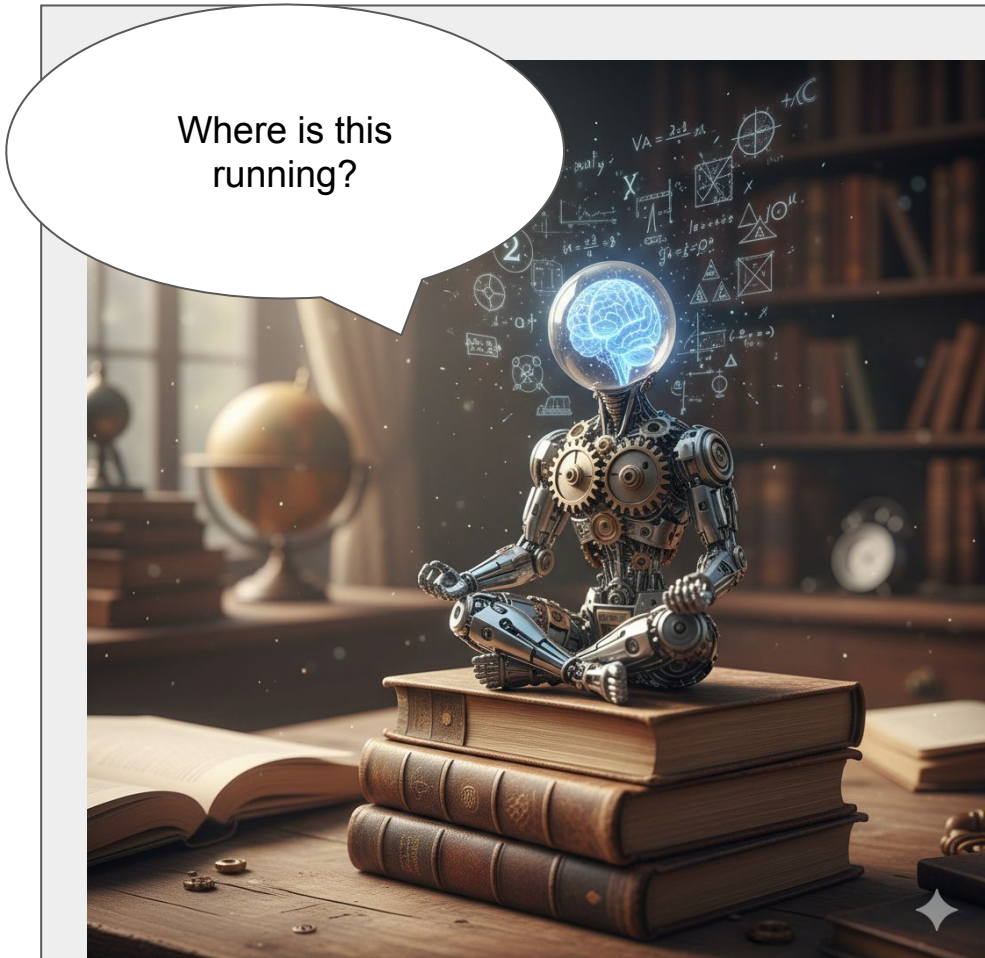
MCP Server

Flux tools
submit job
job info
…
cancel job

Build tools
spack install…
podman build
…
cmake ../

Resources (data)
and

Prompts
I am a debugging agent…



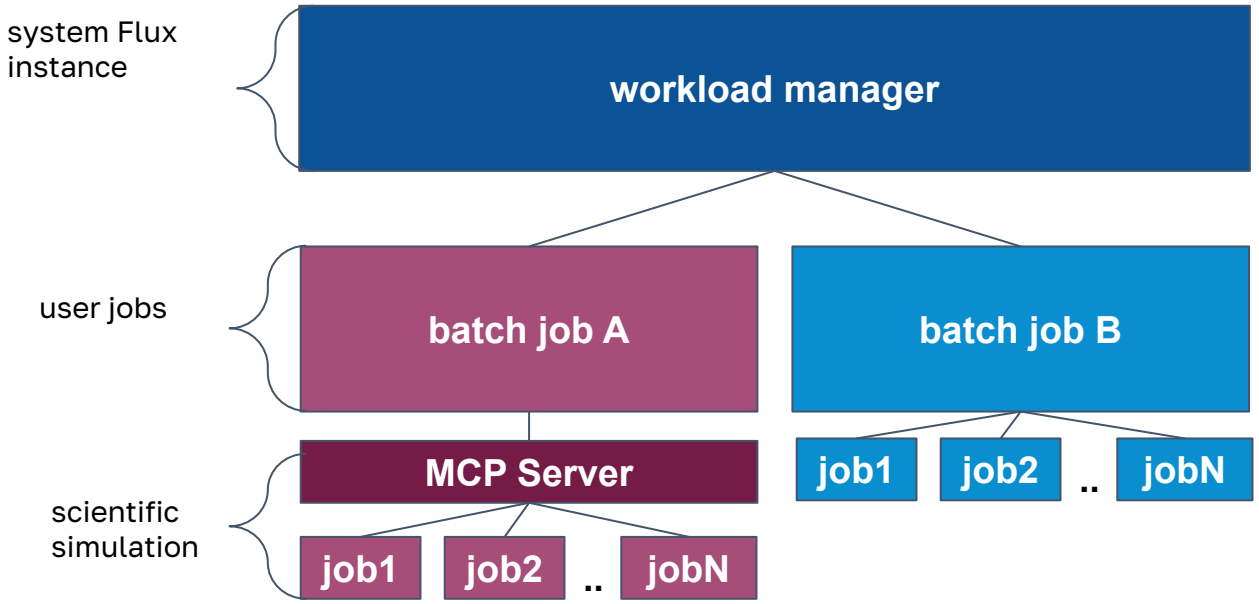
Model Context Protocol (MCP)

Traditional HPC clusters are not oriented to hosting services.

MCP Server

Where do you run an MCP Server on HPC Systems?

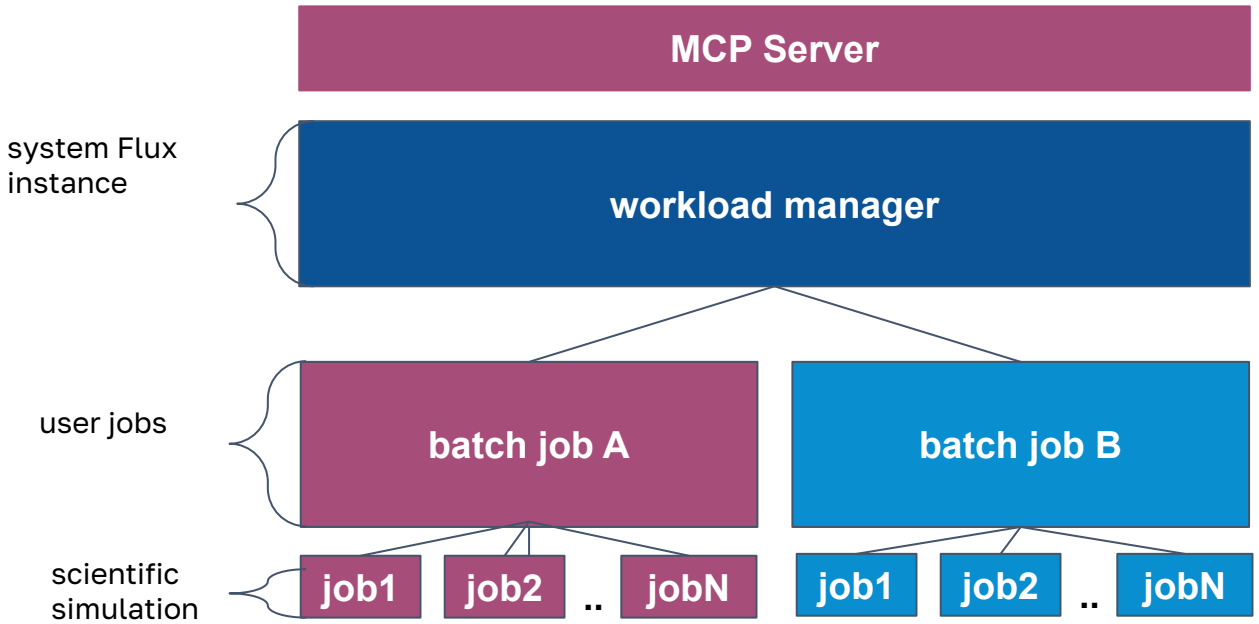
Case 1: In the context of a batch job.



Use case: I want to deploy ephemeral, app-level services. I might run it on bare-metal or in user-space Kubernetes.

Where do you run an MCP Server on HPC Systems?

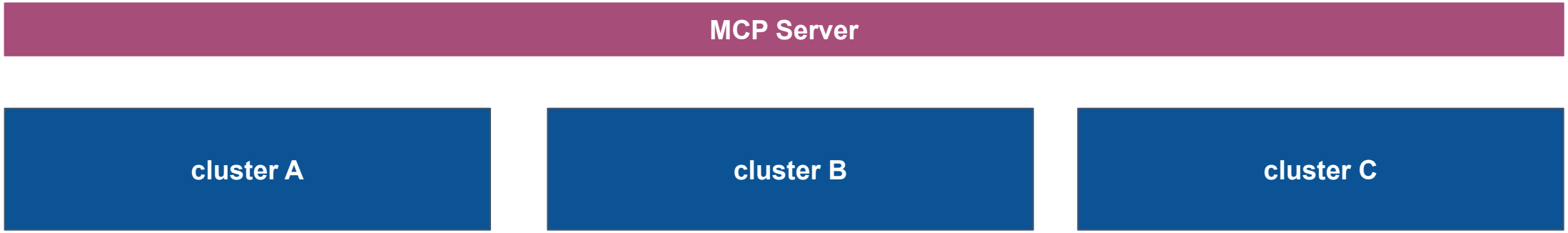
Case 2: As a cluster service



Use case: I want to deploy persistent, cluster-level services. I might use my center's production, integrated service setup.

Where do you run an MCP Server on HPC Systems?

Case 3: As a multi-cluster service

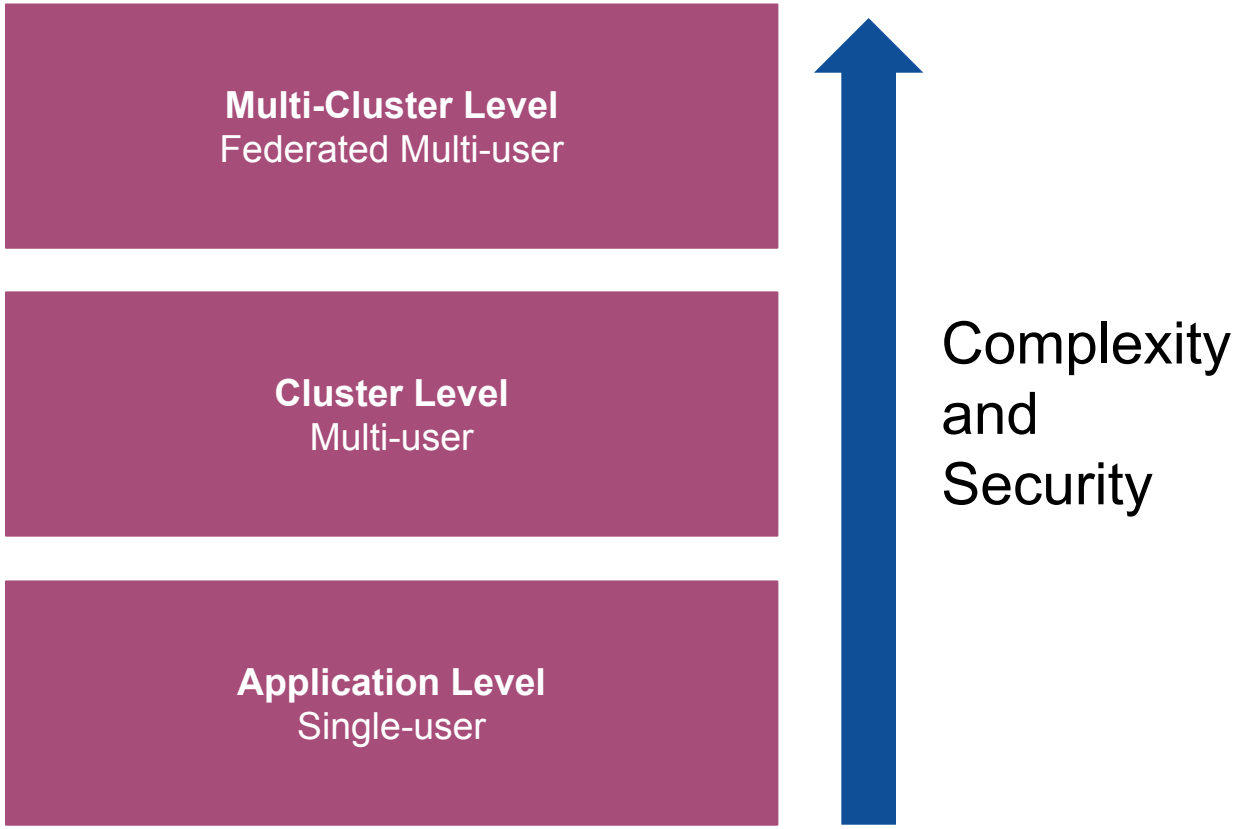


Use case

I want the agent to choose the right cluster to submit my job to.

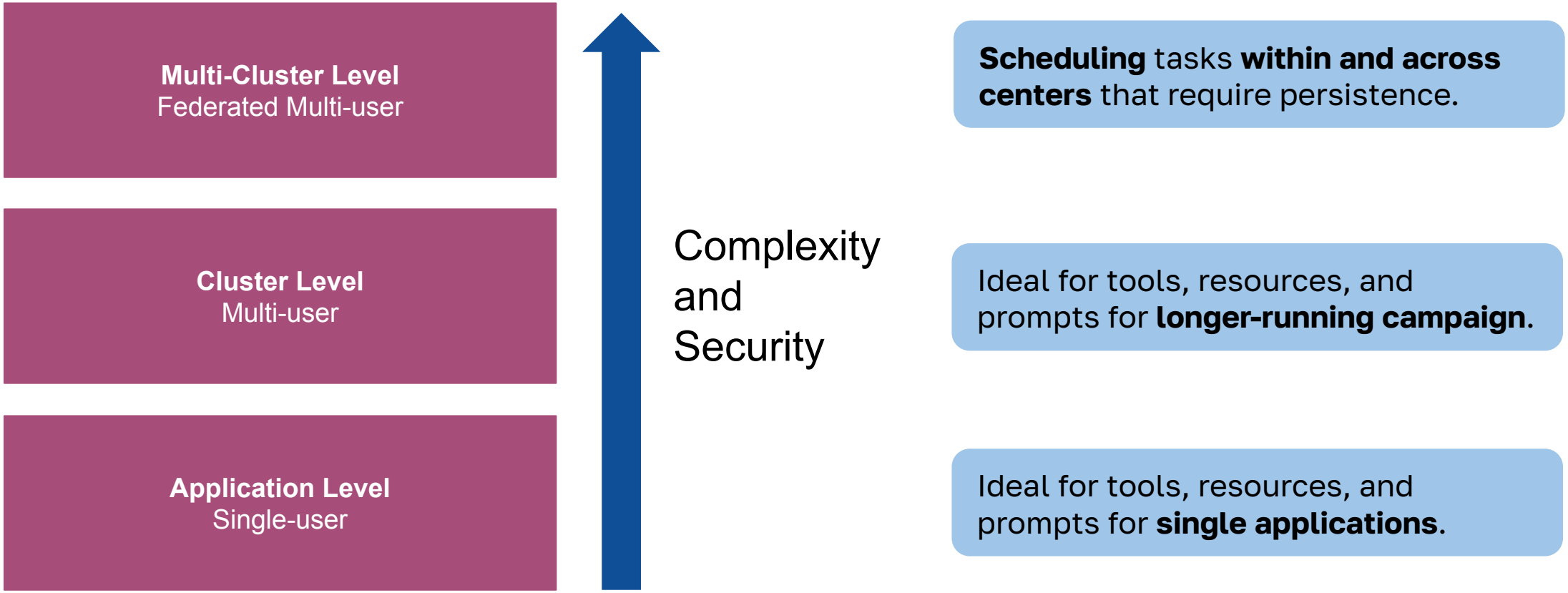
Security is more complex as you move up levels

More users sharing an MCP service means more layers of AuthZ



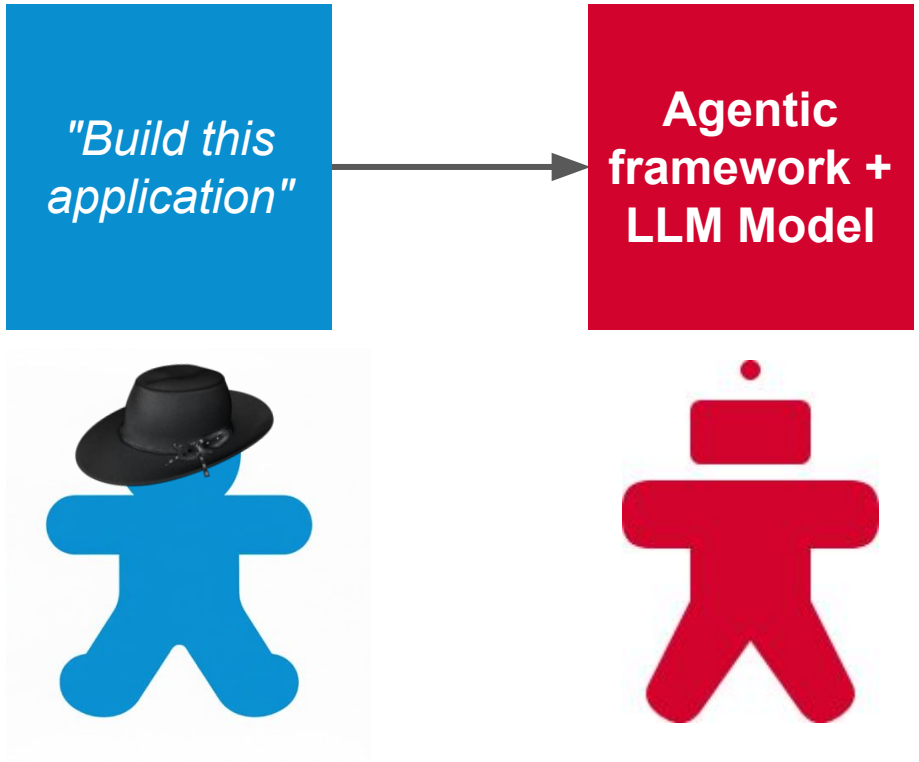
Deployment level choice maps to use case

A choice of where to deploy an MCP server depends on what you need to do.



Model Context Protocol (MCP)

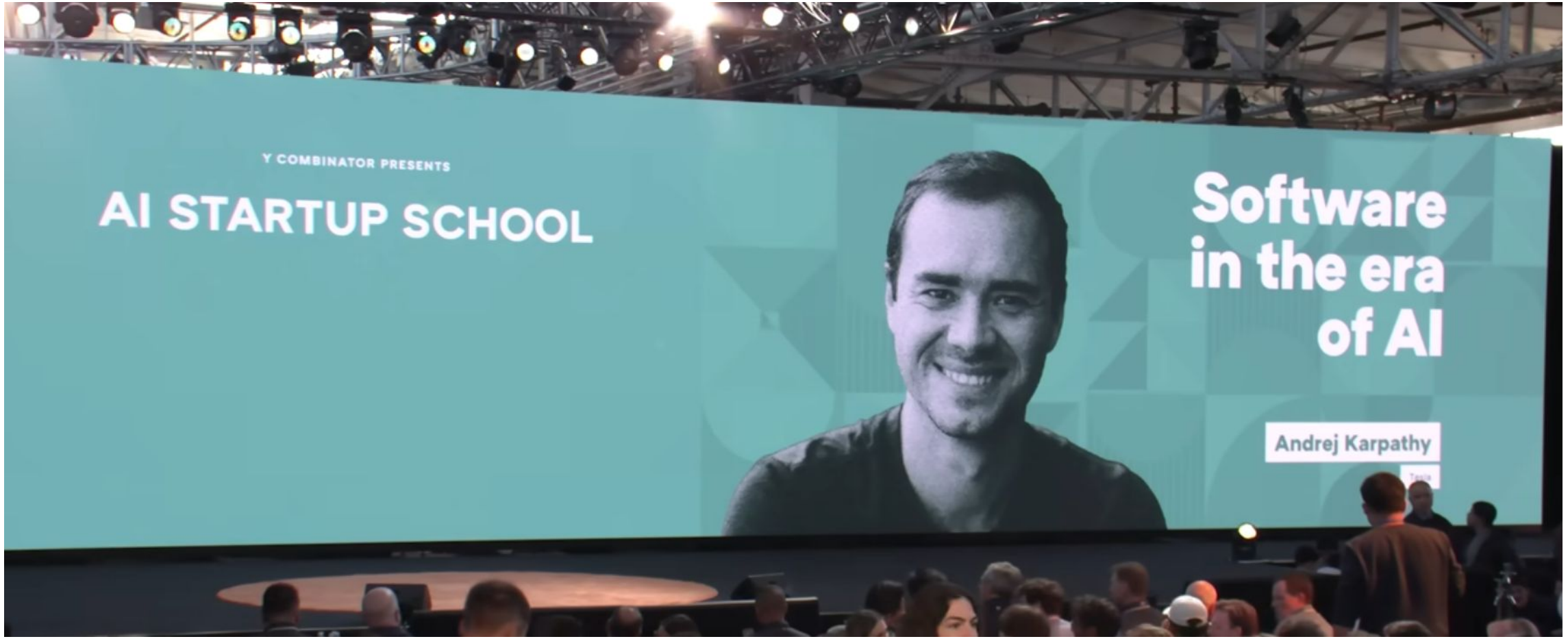
Standard for models (agents) to connect to data, tools, and APIs security.




How is agentic work orchestrated?

The Evolution of Software

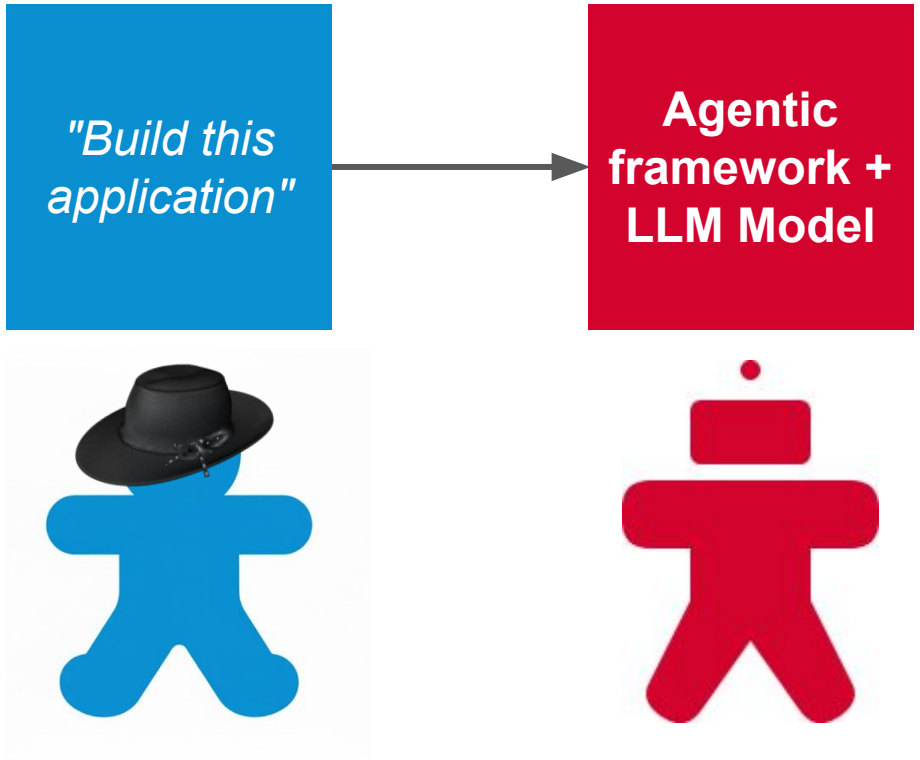
Software 3.0 is conversational



Andrej Karpathy, keynote on June 17, 2025 at AI Startup School in San Francisco.

An Agentic Framework is a design philosophy

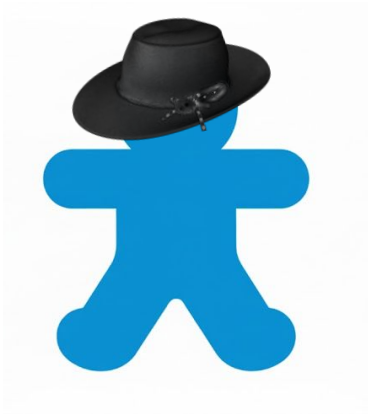
For how steps in a workflow are strung together to call tools, and prompt LLMs



An Agentic Framework is a design philosophy

For how steps in a workflow are strung together to call tools, and prompt LLMs

"Build this application"



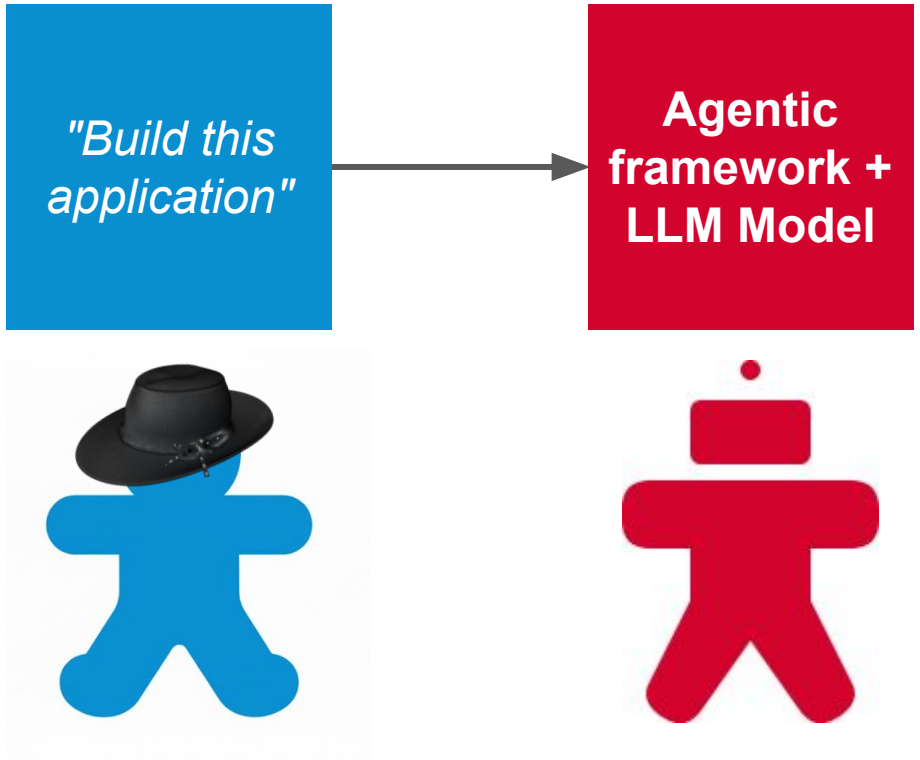
Agentic framework + LLM Model

Agent:

Using an LLM as its "brain" to reason through problems, create plans, and execute multi-step tasks. Agents typically use external tools and resources.

An Agentic Framework is a design philosophy

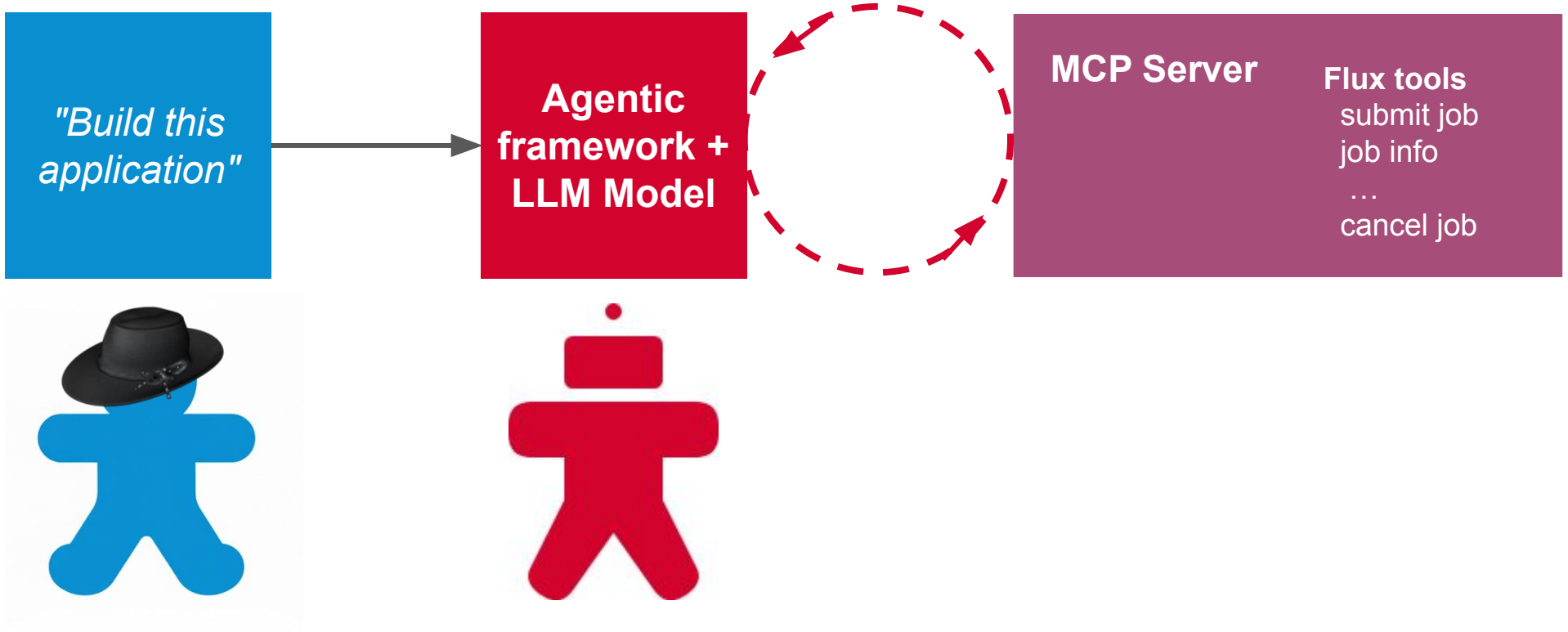
For how steps in a workflow are strung together to call tools, and prompt LLMs



AutoGen
LangChain
State Machine

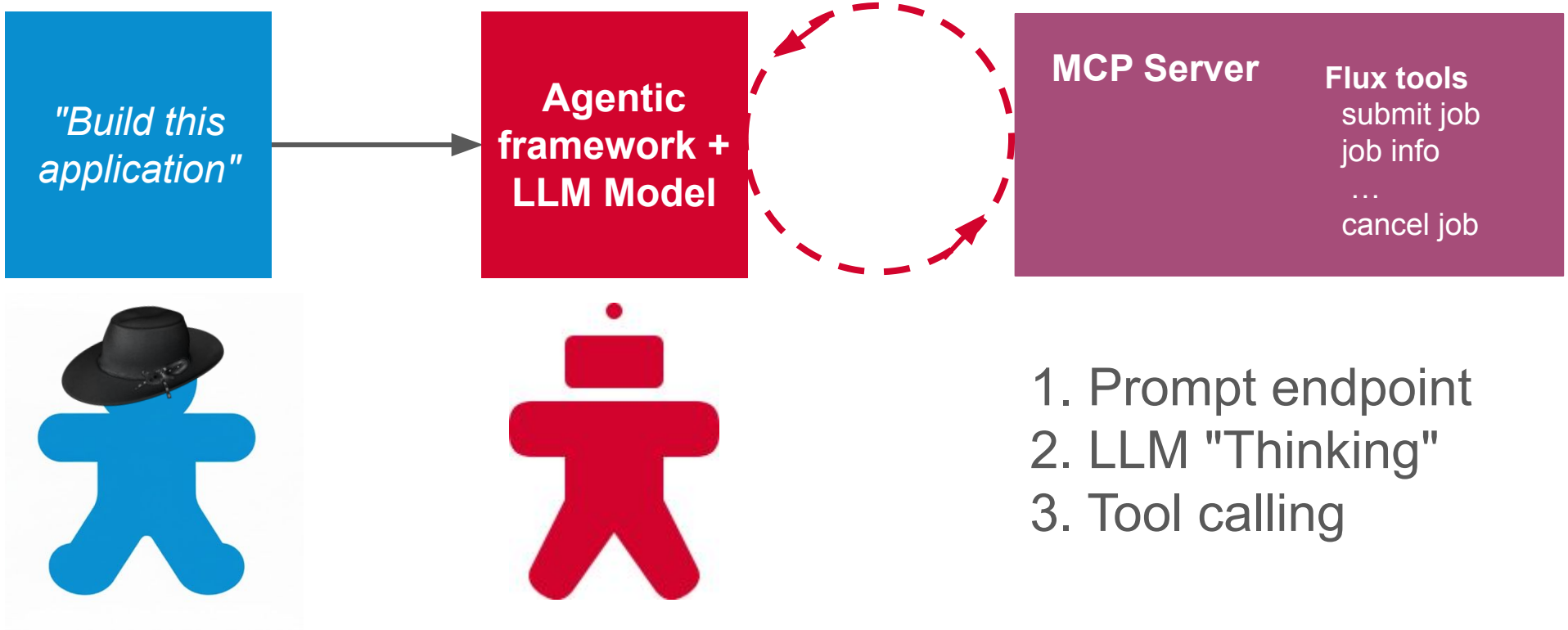
Agentic Frameworks Interact with MCP Servers

The frameworks typically provide hosts and create clients



"Run an HPC Application"

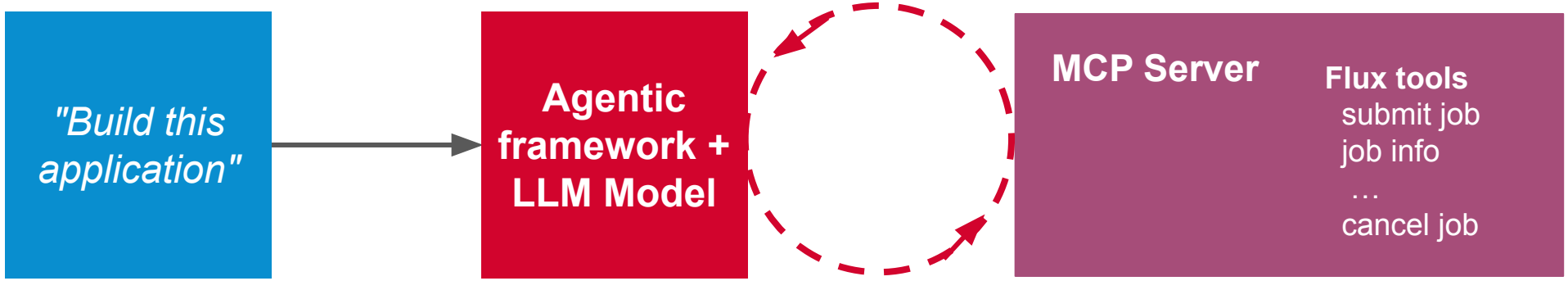
Agentic, and automated.

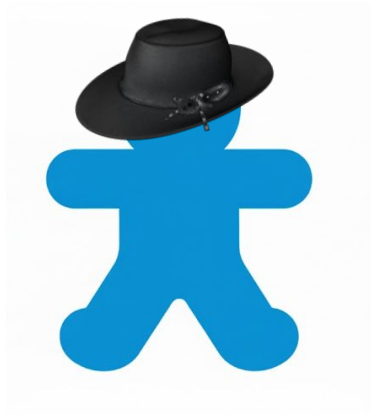


"Run an HPC Application"

Agentic, and automated.

1. Build Prompt endpoint
2. Spack build tool
3. Batch script tool





4. Flux submit tool
5. Flux status tool
6. Analyze results

Are we ready from a staffing perspective?

There is an HPC / MCP skill gap that needs to be bridged.

Are we ready from a staffing perspective?

There is an HPC / MCP skill gap that needs to be bridged.

- Asynchronous programming: async/await patterns to handle non-blocking I/O

Are we ready from a staffing perspective?

There is an HPC / MCP skill gap that needs to be bridged.

- Asynchronous programming: async/await patterns to handle non-blocking I/O
- Event-Driven Architectures: Move away from linear scripts to systems that respond to events.

Are we ready from a staffing perspective?

There is an HPC / MCP skill gap that needs to be bridged.

- Asynchronous programming: async/await patterns to handle non-blocking I/O
- Event-Driven Architectures: Move away from linear scripts to systems that respond to events.
- Model View Control: understanding separation of concerns:
  - Model (data/logic) - data/HPC code
  - View (user interface) - command line or GUI/TUI interaction
  - Control (communication or interaction) (API)

Are we ready from a staffing perspective?

There is an HPC / MCP skill gap that needs to be bridged.

- Asynchronous programming: async/await patterns to handle non-blocking I/O
- Event-Driven Architectures: Move away from linear scripts to systems that respond to events.
- Model View Control: understanding separation of concerns:
  - Model (data/logic) - data/HPC code
  - View (user interface) - command line or GUI/TUI interaction
  - Control (communication or interaction) (API)
- Modern API standards (JSON-RPC, REST, WebSockets)

Are we ready from a staffing perspective?

There is an HPC / MCP skill gap that needs to be bridged.

- Asynchronous programming: async/await patterns to handle non-blocking I/O
- Event-Driven Architectures: Move away from linear scripts to systems that respond to events.
- Model View Control: understanding separation of concerns:
  - Model (data/logic) - data/HPC code
  - View (user interface) - command line or GUI/TUI interaction
  - Control (communication or interaction) (API)
- Modern API standards (JSON-RPC, REST, WebSockets)
- Persistent service management: building long running, or medium-term running services and orchestration.

Are we ready from a staffing perspective?

There is an HPC / MCP skill gap that needs to be bridged.

- Asynchronous programming: async/await patterns to handle non-blocking I/O
- Event-Driven Architectures: Move away from linear scripts to systems that respond to events.
- Model View Control: understanding separation of concerns:
  - Model (data/logic) - data/HPC code
  - View (user interface) - command line or GUI/TUI interaction
  - Control (communication or interaction) (API)
- Modern API standards (JSON-RPC, REST, WebSockets)
- Persistent service management: building long running, or medium-term running services and orchestration.
- Web Security AuthZ or OAuth, JWT tokens, CORS layers above typical SSH/LDAP

Are we ready from a staffing perspective?

There is an HPC / MCP skill gap that needs to be bridged.

- Asynchronous programming: async/await patterns to handle non-blocking I/O
- Event-Driven Architectures: Move away from linear scripts to systems that respond to events.
- Model View Control: understanding separation of concerns:
  - Model (data/logic) - data/HPC code
  - View (user interface) - command line or GUI/TUI interaction
  - Control (communication or interaction) (API)
- Modern API standards (JSON-RPC, REST, WebSockets)
- Persistent service management: building long running, or medium-term running services and orchestration.
- Web Security AuthZ or OAuth, JWT tokens, CORS layers above typical SSH/LDAP

HPC *must* become more cloud-like to use latest technologies and stay relevant.

Strategies to Bridging to Agentic HPC

How do we start from traditional HPC and move to programmatic setups?

Strategies to Bridging to Agentic HPC

How do we start from traditional HPC and move to programmatic setups?

- **Short term portable solutions.**
  - We can deploy MCP servers in the context of a job.
  - If you already have cluster services, deploy them there.
  - User-space Kubernetes

Strategies to Bridging to Agentic HPC

How do we start from traditional HPC and move to programmatic setups?

- **Short term portable solutions.**
  - We can deploy MCP servers in the context of a job.
  - If you already have cluster services, deploy them there.
  - User-space Kubernetes

- **Long term persistent solutions.**
  - Design centers that are programmatic, event-driven.
  - This is the vision of NG-HPCC and Genesis.
  - Opinion: Kubernetes, convergence of community is important for a successful outcome.
  - We do not have an "answer" for this yet beyond "We'll find out."

Strategies to Bridging to Agentic HPC

How do we start from traditional HPC and move to programmatic setups?

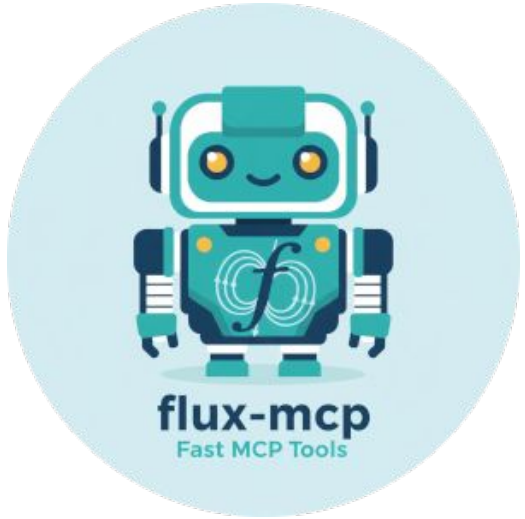
- **Short term portable solutions.**
  - We can deploy MCP servers in the context of a job.
  - If you already have cluster services, deploy them there.
  - User-space Kubernetes

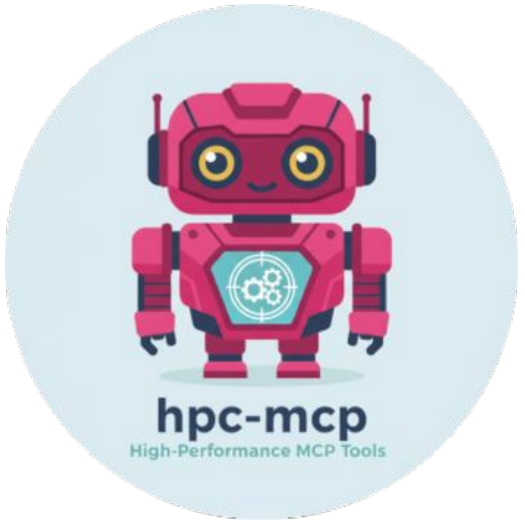
- **Long term persistent solutions.**
  - Design centers that are programmatic, event-driven.
  - This is the vision of NG-HPCC and Genesis.
  - Opinion: Kubernetes, convergence of community is important for a successful outcome.
  - We do not have an "answer" for this yet beyond "We'll find out."

Caveat! I am *not* a system administrator, but I have worked in research computing and done extensive cloud development.

What is our team actively working on?

We believe these to be important components for next stages of work.

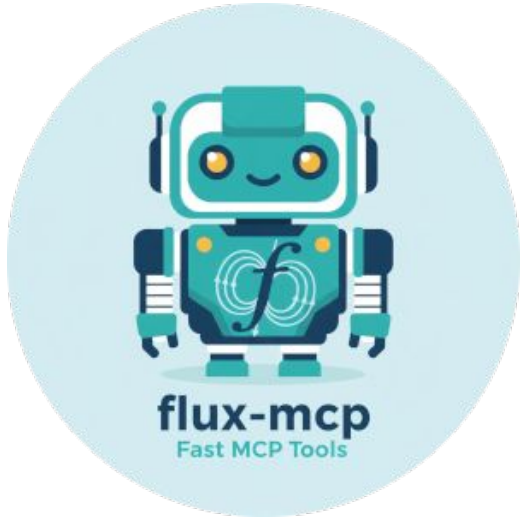


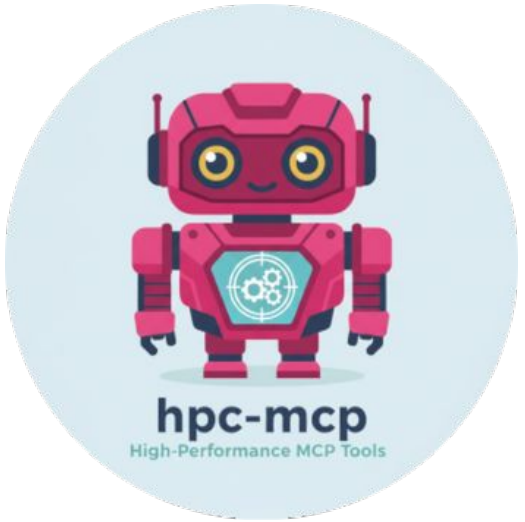


What is our team actively working on?

We believe these to be important components for next stages of work.

- Flux Integration with Kubeflow for AI/ML and HPC Simulation (**KEP 2841**)

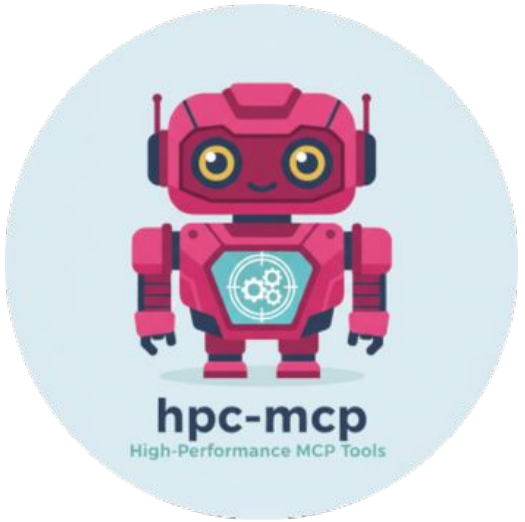




What is our team actively working on?

We believe these to be important components for next stages of work.

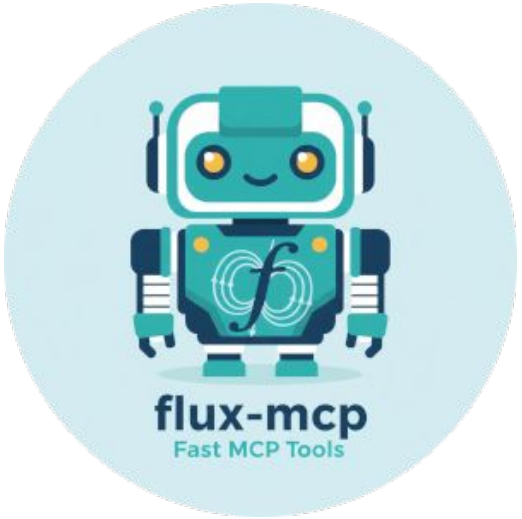
- Flux Integration with Kubeflow for AI/ML and HPC Simulation (**KEP 2841**)
- Kubernetes Operator for agentic workflows

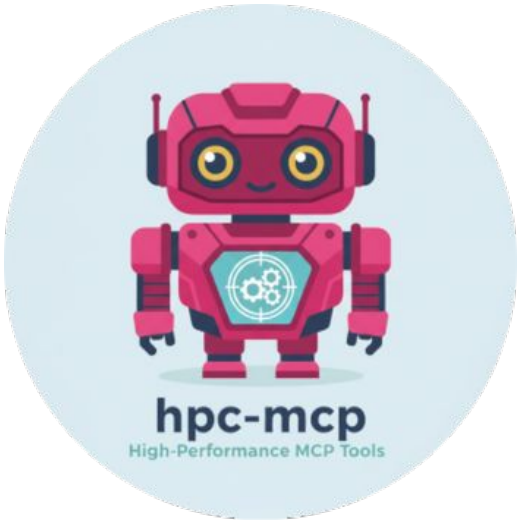


What is our team actively working on?

We believe these to be important components for next stages of work.

- Flux Integration with Kubeflow for AI/ML and HPC Simulation (**KEP 2841**)
- Kubernetes Operator for agentic workflows
- MCP Tools for Flux and High Performance Computing
  - flux-mcp: MCP tools for Flux Framework
  - hpc-mcp: HPC tools for a larger set of HPC and converged computing use cases.
  - fractale-mcp: (fractale) MCP orchestration (agents, databases, ui interfaces).

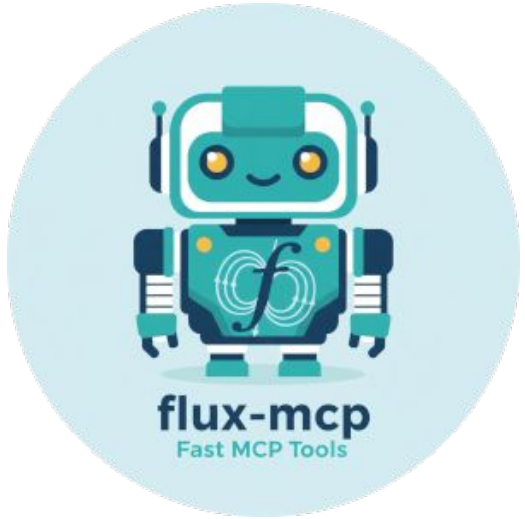


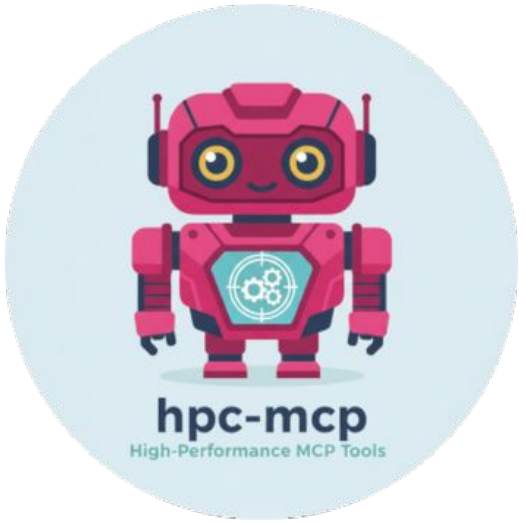


What is our team actively working on?

We believe these to be important components for next stages of work.

- Flux Integration with Kubeflow for AI/ML and HPC Simulation (**KEP 2841**)
- Kubernetes Operator for agentic workflows
- MCP Tools for Flux and High Performance Computing
  - flux-mcp: MCP tools for Flux Framework
  - hpc-mcp: HPC tools for a larger set of HPC and converged computing use cases.
  - fractale-mcp: (fractale) MCP orchestration (agents, databases, ui interfaces).
- Multi-cluster scheduling algorithms and infrastructure.





What is our team actively working on?

We believe these to be important components for next stages of work.

- Flux Integration with Kubeflow for AI/ML and HPC Simulation (**KEP 2841**)
- Kubernetes Operator for agentic workflows
- MCP Tools for Flux and High Performance Computing
  - flux-mcp: MCP tools for Flux Framework
  - hpc-mcp: HPC tools for a larger set of HPC and converged computing use cases.
  - fractale-mcp: (fractale) MCP orchestration (agents, databases, ui interfaces).
- Multi-cluster scheduling algorithms and infrastructure.
- NG-HPCC

Challenges to Work On

What are current challenges, and areas of opportunity?

- **Tool call decision**: We can do better than a single endpoint that lists *all* functions
- **Multi-agentic**: Collaboration between agents that weren't intended to work together
- **Validation**: Always a challenge! Distinguish between truth and hallucination
- **Guardrails**: Safeguards added to infrastructure to protect from erroneous behavior
- **Human interact**: How and when humans should come into the loop
- **Scheduling**: At scale, multi-objective, and choosing between heterogeneous resources
- **Identity**: Especially for multi-cloud setups
- …

What are steps you can take now?

Large change can start with small steps, as it inspires thinking.

- How can you enable users to deploy MCP servers under single jobs?

What are steps you can take now?

Large change can start with small steps, as it inspires thinking.

- How can you enable users to deploy MCP servers under single jobs?
- Think of a workflow *you* have as an administrator. What does it look like, agentic? 👀

What are steps you can take now?

Large change can start with small steps, as it inspires thinking.

- How can you enable users to deploy MCP servers under single jobs?
- Think of a workflow *you* have as an administrator. What does it look like, agentic? 👀
- What larger community projects can be extended to the HPC community?

What are steps you can take now?

Large change can start with small steps, as it inspires thinking.

- How can you enable users to deploy MCP servers under single jobs?
- Think of a workflow *you* have as an administrator. What does it look like, agentic? 👀
- What larger community projects can be extended to the HPC community?
- Think about user security, and permission to access models (e.g., your data may be private)
- …

Conclusion

The HPC center of the future is built upon convergence, modularity, agency

- **Hierarchical resource orchestration**: A move from monolithic to modular frameworks like Flux will support modern, complex, heterogeneous workloads
- **User-space orchestration:** Empower users to run the show, and don't rely on center services unless absolutely necessary.
- **Standard protocols:** Use standard protocols and community libraries like MCP and Kubeflow.
- **Scientific AI Agents:** Foster scientific AI tools. Imagine tasks of today in this setup.
- **Roles are Changing:** Do not be afraid to learn, and work differently than before.
- **Prototyping is Cheap:** Code is not a valuable asset anymore. Make 20 prototypes and iterate on designs quickly. Learn a lot.

Conclusion

The HPC center of the future is built upon convergence, modularity, agency

- **Hierarchical resource orchestration**: A move from monolithic to modular frameworks like Flux will support modern, complex, heterogeneous workloads
- **User-space orchestration:** Empower users to run the show, and don't rely on center services unless absolutely necessary.
- **Standard protocols:** Use standard protocols and community libraries like MCP and Kubeflow.
- **Scientific AI Agents:** Foster scientific AI tools. Imagine tasks of today in this setup.
- **Roles are Changing:** Do not be afraid to learn, and work differently than before.
- **Prototyping is Cheap:** Code is not a valuable asset anymore. Make 20 prototypes and iterate on designs quickly. Learn a lot.

The HPC Center of the future is not just a facility for running jobs. It is a proactive, autonomous platform that accelerates science.

Thank you!
Questions?

milroy1@llnl.gov | sochat1@llnl.gov