



# MCP Agents in PanDA

Paul Nilsson, Tadashi Maeno, Torre Wenaus for the REDWOOD project

October 29, 2025  
**AI4EIC Workshop**

**REDWOOD** (Resilient Federated Workflows in a Heterogeneous Computing Environment), DOE ASCR Award No. DE-SC0012704, 2023–2028



# In this talk

- **Ask PanDA**

An AI-driven virtual assistant that uses language models and retrieval-augmented generation to provide contextual insight into PanDA operations

- **PanDA MCP**

A Model Context Protocol (MCP) interface enabling AI agents to interact with PanDA services and APIs through standardized, discoverable tools

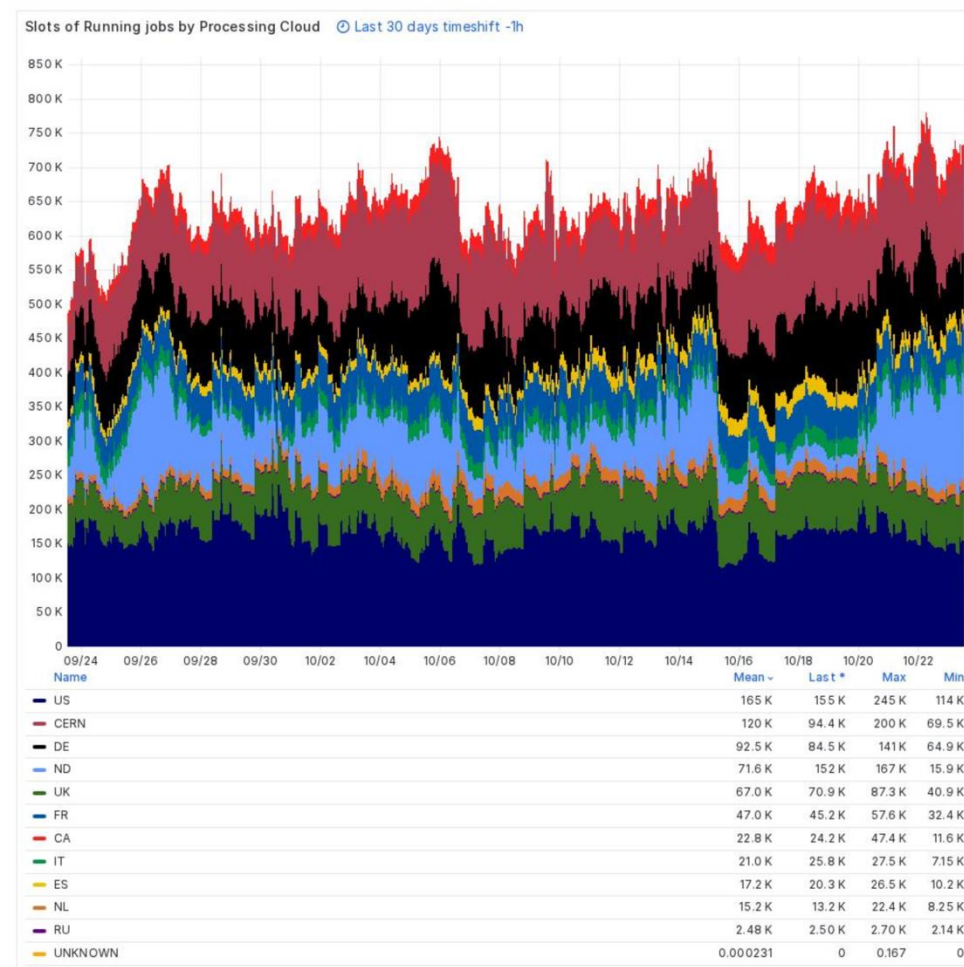
- **ePIC Testbed**

A shared R&D environment at BNL integrating PanDA, Rucio, and MCP to evaluate AI-assisted workflow automation and streaming data processing across experiments



# PanDA: Distributed Computing at Scale

- PanDA was originally developed for the ATLAS experiment at CERN to manage large-scale distributed production and analysis workflows
- The system handles hundreds of thousands of concurrent jobs, across globally distributed computing resources (grids, clouds, high-performance computing) providing a unified interface to the user community
- PanDA embraces heterogeneous resource types
  - Grid sites, commercial clouds, high-performance computing facilities offering resource transparency to users
- It features a real-time monitoring and brokerage layer that maps jobs to resources based on dynamic site/status information enabling optimized scheduling and load-balancing
- PanDA is being adopted beyond ATLAS
  - The Vera C. Rubin Observatory selected PanDA in 2023/2024 for its data-processing workflows (multi-site, DAG support, prompt processing)
  - Under evaluation by the ePIC, SKA and DarkSide collaborations



# Smarter Operations for Distributed Computing

- Scale explosion
  - Soon millions of jobs daily across grids, clouds, and HPC sites
  - Continuous increases in data volume and job diversity (e.g., simulation, ML training, prompt data processing)
- Operational complexity
  - Thousands of users and hundreds of sites → fragmented monitoring, site configurations, and error patterns
  - Many failure modes: transient site errors, data transfer issues, resource contention, etc
- Human bottlenecks
  - Experts manually diagnose failures from logs and dashboards
  - Growing gap between system complexity and available manpower
- Need for smarter tools
  - Automate diagnostics, triage, and recovery
  - Provide natural-language interfaces for querying job and system state
  - Enable proactive monitoring instead of reactive maintenance
- AI as an enabler
  - Large language models can interpret logs, metadata, and documentation in context
  - MCP standardizes these AI interactions safely and modularly

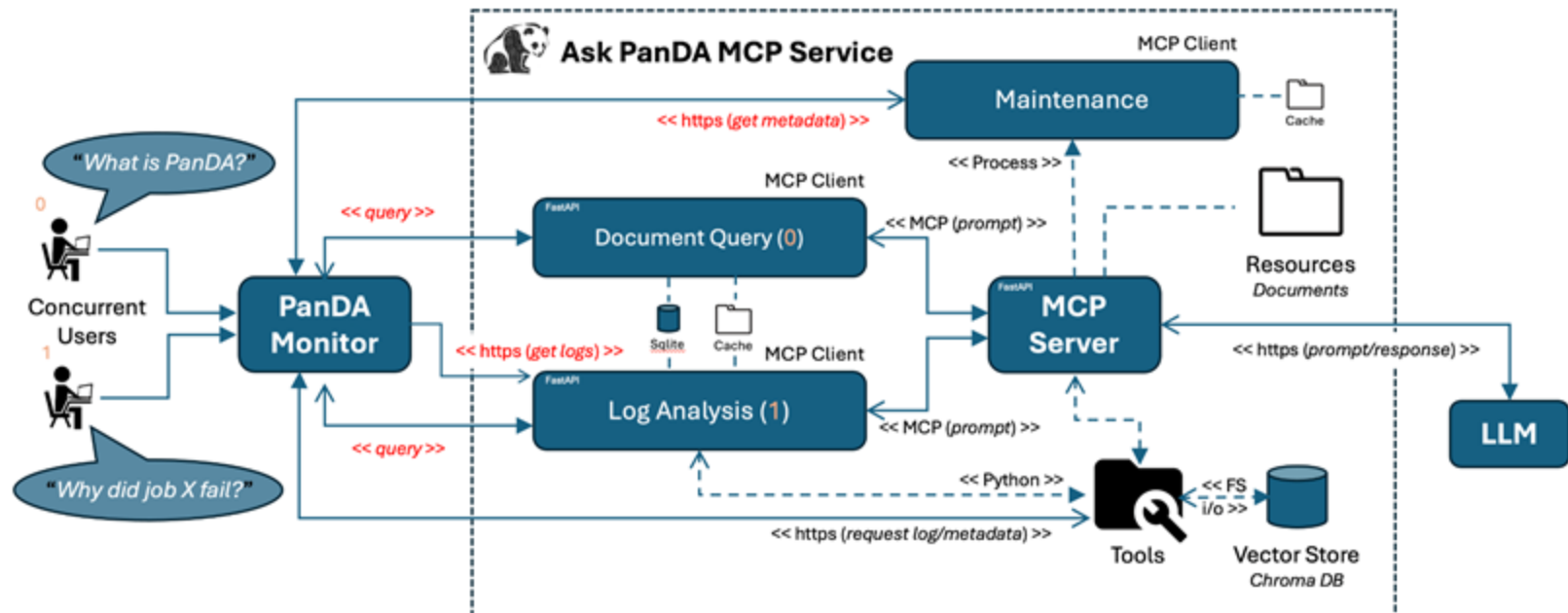
**“Reactive maintenance”** = fix problems **after** they happen.  
**“Proactive monitoring”** = detect and mitigate problems **before** they impact operations.

## Key Message

Modern scientific workflows are too large, dynamic, and heterogeneous for manual monitoring and debugging; intelligent automation is essential

# Ask PanDA: AI-Assisted Operations via MCP

- “Ask PanDA” is an AI agent system built on top of MCP, enabling intelligent interaction with PanDA services
  - It consists of multiple MCP clients communicating with an MCP server exposing PanDA tools
  - Clients exist for static document queries via RAG, log analysis for user and production jobs, metadata queries, maintenance tasks and automatic selection of which client to use for different types of queries
  - Several other clients are in development, incl. database queries for different types of metadata, OpenSearch, ..
- Support for Gemini, Anthropic, Open AI, Llama and Mistral via APIs
- Currently using Open WebUI for user interface
  - To be integrated with next generation PanDA Monitor
- Agent can answer both static and dynamic questions
  - “How do I submit a job to the grid?”
  - “Why did my job XYZ fail?”
- Chat history enabled in all relevant clients



Ask PanDA architecture diagram showing the MCP service with selected clients

# AI Assistance in Ask PanDA

- Document queries using RAG-based documentation
  - ChromaDB vector store index for PanDA and Rucio documentation, Q&A for user analysis problems, frequently asked questions
  - Currently using local vector store, but with ideas for moving everything into OpenSearch
- Dynamic job/task diagnostics
  - Metadata analysis to check on progress with jobs and tasks (group of jobs)
  - Client downloads relevant metadata -> Enhanced prompt sent to LLM for explanations
- Log-based failure analysis
  - Completed (failed) jobs can be analyzed for more detailed explanations and suggestions for how to mitigate problem
  - Client downloads metadata and different types of log files -> Prompt sent to LLM
- Other clients in conceptual design / early development
  - To answer additional types of questions; “What’s the system doing now?”, “Summarize last week’s workflow performance”, “Please retry job XYZ”

## Document queries using RAG-based documentation

What is happening with task 46374283?

## What is happening with task XYZ?

01 Ask PanDA  
Response ready

**Description:**  
The PanDA task with jeditaskId 46374283, named 'user.aspellma.data24.25\_09\_09\_pass4\_eventSel\_config\_CRfailIsoLoose\_do\_master\_nMinus1', is currently in 'running' status. It was created by user Alic Shen Spellman on 2025-09-09 16:48:41. The task is an analysis task ('anal' tasktype) using the 'panda-client-1.5.92-jedi-run' processing type. It involves event selection with specific configurations. The task's input is derived from dataset 'user.gripelli.2025\_08\_01.data24\_13p6TeV.periodAllYear.grp24\_v02\_p6649\_TREE/', and the output is directed to 'user.aspellma.data24.25\_09\_09\_pass4\_eventSel\_config\_CRfailIsoLoose\_do\_master\_nMinus1'. The task's progress is indicated by the file processing status. A significant portion of the input files are marked as 'ready,' some with a completion percentage of 100%, indicating successful processing for those files. A few datasets are finished, but not all. Most input datasets are not finished. Log and output datasets are mostly registered with few finished files. It is using the el9 architecture with x86\_64 CPU.

### Problems:



**Details:**

The task has 4875 total input files, with 1112 files finished and 0 files failed, according to the 'dsinfo' field. This translates to 22% of the input files being finished.



- Input Datasets (IN): A mix of 'ready' statuses with varying 'nfilesfinished' and 'percentfinished'. Many are at 0% finished, while some are at 100%.
- Log Datasets (LOG0): Mostly in 'registered' status with a small number of finished files, and one dataset with a failed file.
- Output Datasets (OUTPUT0): Mostly in 'registered' status with a small number of finished files, and one dataset with a failed file.

**File Processing:**

- The input dataset has a total of 4875 files.
- 1112 input files are marked as 'finished'.
- 0 input files are marked as 'failed'.

Overall, the task is in progress, with a significant number of input files still waiting to be processed. The presence of failed files in the log and output datasets suggests potential issues in specific jobs, which might require further investigation. The input dataset is being processed in chunks as per dataset divisions and associated metadata for

(Continues)



# Ask PanDA Examples (2)

Why did job XYZ fail?

Why did job 6800730400 fail?

What is the status of job XYZ?

What is the status of job 6810532013?

Log-based failure analysis

OI Ask PanDA  
Response ready

**Description:**  
Payload exceeded maximum allowed memory.

**Non-expert guidance - problem:**  
The job used more memory than it was allowed to use.

**Non-expert guidance - possible causes:**

- The job's memory requirements were underestimated.
- A memory leak in the job's code.
- The job processed an unexpectedly large amount of data.
- The site's memory limits are too restrictive for this job.

**Non-expert guidance - recommendations:**

- Request more memory for the job if possible.
- Check the job's code for memory leaks or inefficient memory usage.
- Reduce the amount of data the job processes.
- Contact the site administrators to see if the memory limits can be adjusted.

**Expert guidance - analysis:**  
Raised when the payload consumes more memory than the maximum threshold allowed by site or job configuration. It helps enforce resource constraints and prevent system instability due to overconsumption. The pilot monitors the memory usage of the job and its subprocesses and compares it to the configured limit. The limit is often derived from queuedata parameters such as `pilot_rss_grace` and `maxrss`.

**Expert guidance - investigation steps:**

- Examine the job's memory usage profile using tools like `top`, `ps`, or specialized memory monitoring tools.
- Check the site's queuedata configuration to determine the memory limits for the job.
- Review the job's log files for any indications of memory-related issues.
- Analyze the job's code for potential memory leaks or inefficiencies. Use memory profilers.
- If the job involves external libraries, ensure those libraries are not the source of the memory problem.

**Expert guidance - possible scenarios:**

- Memory Leak: The job's code has a memory leak, causing memory usage to increase over time.
- Unexpected Data Volume: The job is processing a larger-than-expected dataset, leading to increased memory usage.
- Configuration Mismatch: The job's memory requirements are not aligned with the site's configured memory limits.
- Inefficient Algorithms: The job employs algorithms that are not memory-efficient, resulting in excessive memory consumption.

Dynamic job diagnostics

LLM generated follow-up questions

OI Ask PanDA  
Response ready

This PanDA job (ID: 6810532013) is currently in 'running' state. It is an event generation (evgen) job submitted by 'ewelina' within the AP\_PHYS working group, part of the ATLAS VO. The job has been running for approximately 5 hours and 46 minutes after waiting for about 21 hours and 20 minutes.

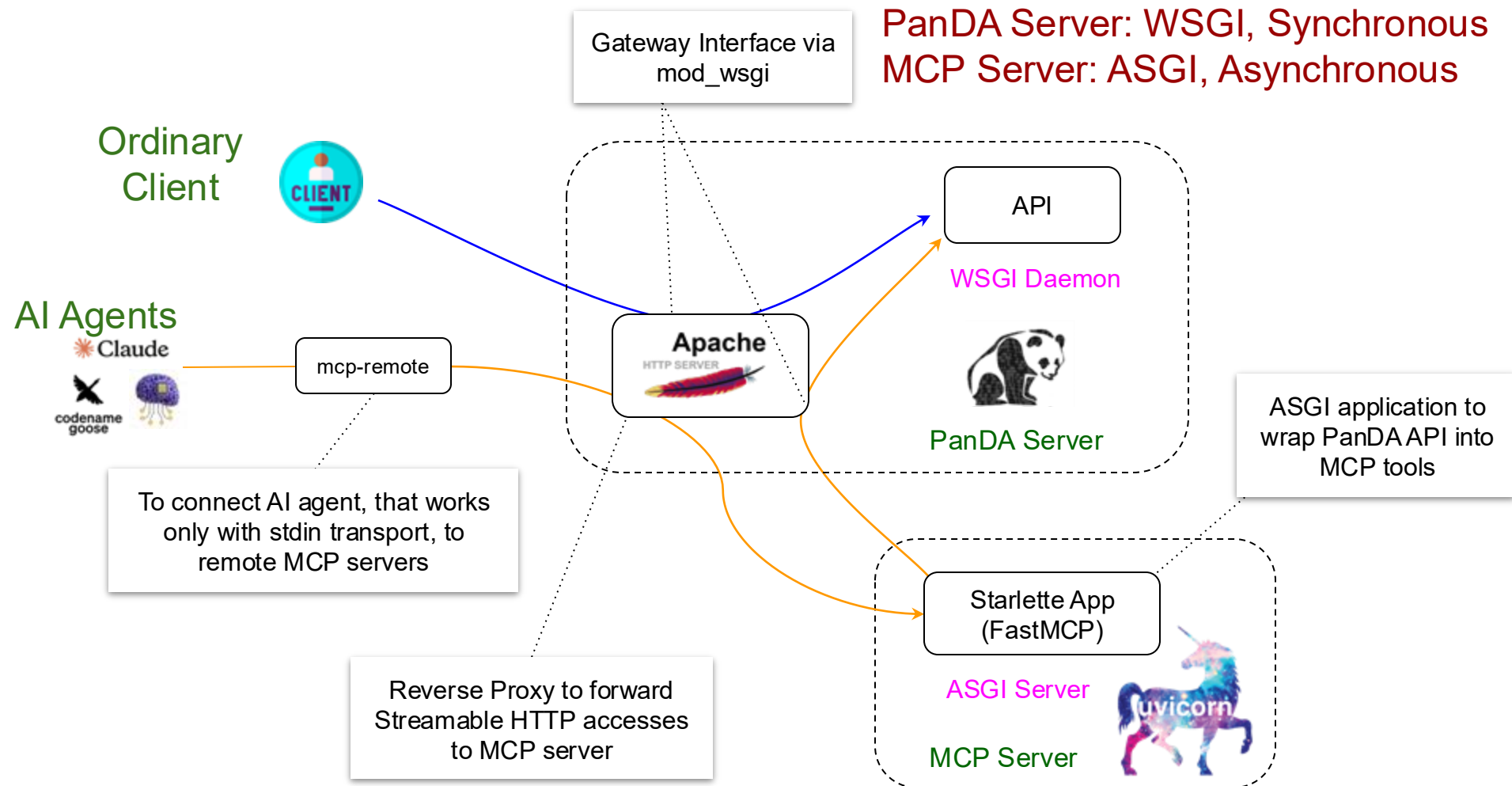
✎ 📄 🔊 👍 💬 🎥 ↺

**Follow up**

- ↳ How can I view the logs or detailed progress for job 6810532013?
- ↳ What is the ID of the task that job 6810532013 belongs to?
- ↳ How can I kill job 6810532013 if it's stuck or I need to stop it?
- ↳ Can I see a list of all jobs within the same task as job 6810532013?



# Bridging PanDA APIs with MCP



# MCP Tools: Exposing PanDA Functionality (1)

```
def create_tool(func):  
  
    mod_path = inspect.getfile(inspect.getmodule(func))  
    mod_name = os.path.basename(mod_path).split('.')[0]  
  
    url = f"{api_url_ssl}/{mod_name}/{func.__name__}"  
  
    sig = inspect.signature(func)  
    params = []  
    for p in sig.parameters.values():  
        if p.annotation != PandaRequest:  
            params.append(p)  
  
    annotations = {}  
    for k, v in func.__annotations__.items():  
        if v != PandaRequest:  
            annotations[k] = v  
  
    def wrapped_func(**kwargs):  
        nonlocal url  
        http_client = HttpClient()  
        status, output = http_client.get(url, kwargs)  
        return output  
  
    wrapped_func.__signature__ = sig.replace(parameters=params)  
    wrapped_func.__annotations__ = annotations  
  
    return wrapped_func, func.__name__, func.__doc__
```

Wrapper function to invoke an arbitrary API call

Modify the function's signature and annotations to align with the original API definition

# MCP Tools: Exposing PanDA Functionality (2)

```
from pandaserver.api.v1.system_api import is_alive
from pandaserver.api.v1.statistics_api import job_stats_by_cloud

mcp = FastMCP("MyServer")

for func in [is_alive, job_stats_by_cloud]:
    f, n, d = create_tool(func)
    mcp.add_tool(f, name=n, description=d)
```

Only API endpoints to expose

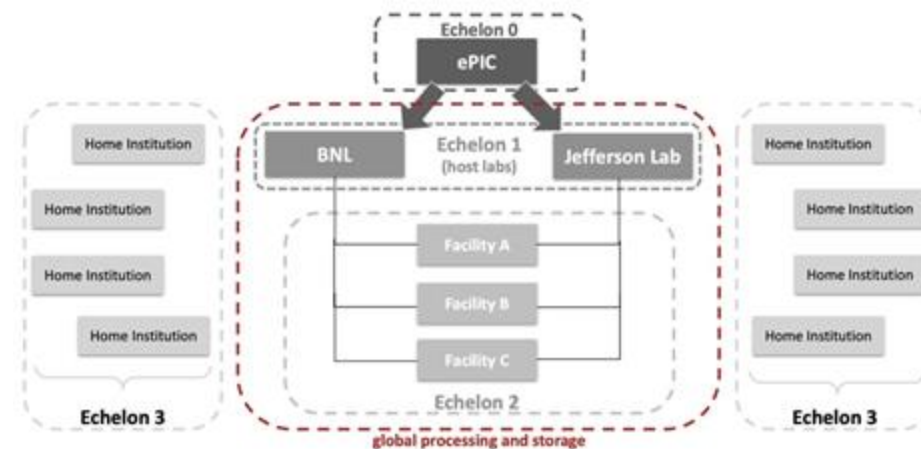
PanDA MCP turns PanDA's REST APIs into self-describing MCP tools that AI agents (such as "Ask PanDA") can query to access and manage jobs, tasks, and system configurations

Add the function as a MCP tool, ensuring its signature, annotations, and docstring are consistent with the original API call

AI agents leverage the signatures, annotations, and docstrings published by MCP server to dynamically generate access parameters and interpret results

# Foundation for Intelligent Workflows: ePIC Streaming Testbed

- ePIC decided in Spring 2025 to initiate testbeds exploring the streaming computing model
  - Streaming workflows, streaming reconstruction and calibration
- Testbed built at Brookhaven National Laboratory to explore *Echelon 0* → *Echelon 1* streaming workflows
- Uses PanDA and Rucio (for data management) as the backbone
- Fully automated, distributed and analytics-rich design
- Agents communicate via ActiveMQ, orchestrating DAQ, data handling and processing tasks
- Testbed mirrors a realistic environment for AI-assisted operations



**Echelon 0:** ePIC experiment, DAQ system

**Echelon 1:** Two host labs, two primary ePIC computing facilities

**Echelon 2:** Global contributions leveraging commitments to ePIC computing from universities and labs domestically and internationally

**Echelon 3:** Supporting the analysis community 'where they are' at their home institutes, primarily via services hosted at E1s/E2s



# Integrating AI via MCP in the Testbed

- MCP enables instrumenting services to expose their APIs to LLMs so they can be live information and context sources for LLMs
- Both Rucio and PanDA teams are implementing MCP services at a fast pace
  - “Why did this job fail?” “What does this error message mean?”, etc
- The testbed monitor’s django implementation also has an asynchronous django service supporting MCP via a (currently trivial) [REST API](#)
- Currently only supports capabilities discovery, agent liveness checking, and heartbeat processing
- Development will continue once the testbed workflow priorities are advanced
- Objective:
  - Make it a source of LLM knowledge of system state, component status, workflow activity, problem diagnostics
  - Use it in concert with PanDA, Rucio MCP services
- How to use it?
  - “what’s the system doing now”
  - “how did the last workflow run perform”
  - “analyze the failures we have seen today”
  - “summarize the last week’s runs [with varying workflow parameters] and what we learned”

# Roadmap: Toward Intelligent Multi-Experiment Operations

## Current Status of Ask PanDA and PanDA MCP

- Development version of Ask PanDA operates in a local test environment
- Ask PanDA validated on ATLAS workloads, PanDA MCP demonstrated successful integration with PanDA APIs
- Architecture and tools are experiment-agnostic, requiring minimal adaptation for new users

## Ask PanDA API and Plugin Framework

- Designing a unified Ask PanDA API to expose core functionalities (RAG queries, diagnostics, log analysis)
- Introducing a modular plugin architecture to support multiple experiments (planned plugins for ATLAS, Vera Rubin, and ePIC)
- Providing reference implementations and developer documentation for community adoption

## Security and Deployment Readiness

- Integrating authentication and authorization aligned with CERN SSO and experiment-specific identity services
- Planning secure connections to external services (e.g., OpenSearch, PanDA Monitor)
- Targeting deployment on PanDA Monitor nodes for initial production testing

## Future Directions

- Shared R&D testbed at Brookhaven National Laboratory for multi-experiment evaluation
- Expansion toward intelligent workflow automation, adaptive monitoring, and failure recovery
- Collaboration across experiments to establish AI-assisted operations as a standard part of distributed computing

# Summary

- **Ask PanDA:** AI-assisted virtual assistant enabling contextual analysis of workflows and logs
- **PanDA MCP:** FastAPI-based interface bridging PanDA APIs to AI agents via the Model Context Protocol
- **ePIC Testbed:** Realistic environment for integrating and validating AI-assisted orchestration
- **Outcome:** Foundation for intelligent, self-analyzing distributed workflows
- **Next:** Live demo of Ask PanDA at Supercomputing 2025 next month (DOE booth)