

AI Agents for Enabling Autonomous Experiments at ORNL's HPC and Manufacturing User Facilities

Daniel Rosendo
Oak Ridge National Lab.
Oak Ridge, TN, USA

Stephen DeWitt
Oak Ridge National Lab.
Oak Ridge, TN, USA

Renan Souza
Oak Ridge National Lab.
Oak Ridge, TN, USA

Phillipe Austria
Oak Ridge National Lab.
Oak Ridge, TN, USA

Tirthankar Ghosal
Oak Ridge National Lab.
Oak Ridge, TN, USA

Marshall McDonnell
Oak Ridge National Lab.
Oak Ridge, TN, USA

Ross Miller
Oak Ridge National Lab.
Oak Ridge, TN, USA

Tyler J. Skluzacek
Oak Ridge National Lab.
Oak Ridge, TN, USA

James Haley
Oak Ridge National Lab.
Oak Ridge, TN, USA

Bruno Turcksin
Oak Ridge National Lab.
Oak Ridge, TN, USA

Jesse McGaha
Oak Ridge National Lab.
Oak Ridge, TN, USA

Benjamin Mintz
Oak Ridge National Lab.
Oak Ridge, TN, USA

Feiyi Wang
Oak Ridge National Lab.
Oak Ridge, TN, USA

Mallikarjun Shankar
Oak Ridge National Lab.
Oak Ridge, TN, USA

Sarp Oral
Oak Ridge National Lab.
Oak Ridge, TN, USA

Rafael Ferreira da Silva
Oak Ridge National Lab.
Oak Ridge, TN, USA

ABSTRACT

This paper presents a modular architecture for enabling autonomous cross-facility scientific experimentation using AI agents at ORNL's HPC and manufacturing user facilities. The proposed system integrates a natural language interface powered by an LLM, a multi-agent framework for decision making, programmable facility APIs, and a provenance-aware infrastructure to support adaptive, explainable, and reproducible workflows. We demonstrate how AI agents can orchestrate and optimize additive manufacturing experiments through near real-time coordination between experimental and HPC resources. The architecture is evaluated through a realistic end-to-end workflow that employs a simulated version of the manufacturing facility, showing that the approach reduces coordination overhead and accelerates the scientific discovery process.

CCS CONCEPTS

• **Computing methodologies** → **Distributed computing methodologies; Multi-agent systems; Distributed computing methodologies; Parallel computing methodologies.**

Notice: This manuscript has been authored in part by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (<http://energy.gov/downloads/doe-public-access-plan>).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

XLOOP 2025, November 16, 2025, St. Louis, MO, USA

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/2018/06
<https://doi.org/XXXXXXX.XXXXXXX>

KEYWORDS

Autonomous Scientific Workflows, Cross-Facility Orchestration, AI Agents, Experiment Steering

ACM Reference Format:

Daniel Rosendo, Stephen DeWitt, Renan Souza, Phillipe Austria, Tirthankar Ghosal, Marshall McDonnell, Ross Miller, Tyler J. Skluzacek, James Haley, Bruno Turcksin, Jesse McGaha, Benjamin Mintz, Feiyi Wang, Mallikarjun Shankar, Sarp Oral, and Rafael Ferreira da Silva. 2025. AI Agents for Enabling Autonomous Experiments at ORNL's HPC and Manufacturing User Facilities. In *Proceedings of 7th Annual Workshop on Extreme-Scale Experiment-in-the-Loop Computing (XLOOP 2025)*. ACM, New York, NY, USA, 8 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

Recent advances in artificial intelligence (AI), robotics, and networking are transforming how science is conducted, enabling a new era of autonomous, cross-facility experimentation [7]. In this paradigm, high-performance computing (HPC) systems are increasingly coupled with experimental infrastructure in near real-time. This integration forms dynamic, agent-driven scientific workflows that span national laboratories, user facilities, and edge instruments [1, 21]. These workflows support the rapid and intelligent coordination of synthesis, characterization, simulation, and analysis across domains, accelerating scientific discovery in materials science, biology, energy, and manufacturing. Agents orchestrate experiments and simulations in a way that significantly reduces the time from hypothesis to validation [11]. Achieving this vision requires rethinking scientific workflows, shifting from static orchestration to adaptive systems driven by AI agents that can reason and act in complex environments.

This shift introduces significant challenges. Distributed infrastructures consist of heterogeneous systems, including experimental laboratories, edge devices, and HPC clusters, each with different interfaces, protocols, and access constraints. Traditional workflow systems are not designed to respond dynamically to near real-time data or evolving goals, and often rely on manual coordination that

is slow and error-prone. The integration of AI into the scientific process also raises concerns about reproducibility, transparency, and trust. Workflows must be able to incorporate domain knowledge, maintain provenance, and support explainable decision making. Moreover, agents operating within these environments must be capable of robust, fault-tolerant coordination while navigating asynchronous operations and intermittent availability of resources.

This paper presents an architecture for enabling autonomous cross-facility experiments using AI agents to support scientific experimentation at Oak Ridge National Laboratory’s (ORNL) Manufacturing Demonstration Facility (MDF) and Oak Ridge Leadership Computing Facility (OLCF). Our approach builds on several key initiatives, including the Interconnected Science Ecosystem (INTERSECT) [19], the Secure Scientific Service Mesh (S3M) facility API [15], Flowcept’s provenance-aware infrastructure [16], and recent work on agentic workflows. We propose a modular design in which agents encapsulate facility capabilities, communicate through interoperable interfaces, and reason about scientific goals using schema- and guideline-driven prompts. We evaluate this architecture through a realistic end-to-end workflow that employs a simulated version of resources at the MDF and demonstrates how agent-based orchestration can reduce coordination overhead, increase adaptability, and accelerate the scientific discovery process. Specifically, this paper makes the following key contributions:

- (1) Describe the challenges of connecting ORNL’s MDF and HPC facilities, and explain how AI agents can help enable autonomous experimentation (Section 2).
- (2) Propose a reference architecture to enable cross-facility experiments with AI agents (Section 3).
- (3) Detail the implementation of the architecture by leveraging ORNL’s initiatives and technologies for future autonomous laboratories, including INTERSECT, the S3M Facility API, agentic workflows, and the Flowcept multi-workflow provenance system (Section 4).
- (4) Validate the architecture through a realistic cross-facility experiment using the HPC facility and a simulated MDF environment (Section 5).

2 BACKGROUND

User Facilities. ORNL’s Manufacturing Demonstration Facility (MDF) is a U.S. Department of Energy (DOE) user facility supported by the Advanced Materials and Manufacturing Technologies Office. The MDF is a 110,000 square-foot facility with over 100 additive manufacturing (AM) systems for metals, composites, and ceramics, plus a variety of characterization capabilities. In-situ monitoring is available for many of these AM systems, including infrared and visible-wavelength cameras and thermocouples. For the directed energy deposition (DED) printers relevant to this work, data streams from in-situ monitoring are coordinated through ORNL’s Stereo Correlated Optical and Pyrometric Sensing (SCOPS) software, which is built upon the open-source Robot Operating System (ROS) framework [8]. To manage and harness the data generated across the facility, the MDF has a framework to integrate process specifications, in-situ characterization, ex-situ characterization, and simulation data into a single digital platform [2].

The Oak Ridge Leadership Computing Facility (OLCF) is a U.S. DOE user facility that provides world-class computational resources for supporting cutting-edge scientific investigations. The Advanced Computing Ecosystem (ACE) [10] is a testbed environment that combines heterogeneous compute and data resources to support scientific workloads outside the constraints of production systems. The Oak Ridge Research Cloud (ORC) is an internal cloud platform designed to support flexible, secure, and scalable computing for research and development. Built on an OpenStack infrastructure, ORC enables users to deploy containerized workflows, virtual machines, and CI/CD pipelines for internal projects. In this work, we explore both ACE and ORC to perform cross-facility experiments.

Challenges in Cross-facility Experiments. Experiments spanning DOE instrument facilities and HPC resources face multifaceted challenges. Workflow integration remains labor-intensive due to gaps in technology, policy, and automation. Scientific teams often need to design data pipelines, manage disparate tools, and handle coordination [4]. Time-sensitive, cross-facility workflows are further complicated by network disruptions, maintenance outages, and dynamic routing, which demand robust automation and reconfiguration capabilities to avoid manual intervention [5]. Security and access control are also challenging; current batch-style HPC security policies are unsuited to interactive or near real-time streaming workflows, requiring token-based authentication, policy adaptation, Science DMZs, and SLA-aware scheduling models [6]. At the core lies the challenge of managing heterogeneous, shared ecosystems. Ensuring reproducible performance, seamless data movement, and transparency across diverse systems remains a critical unmet need.

AI Agents for Autonomous Experimentation. AI integration in scientific research has reached a stage where agents can autonomously conduct experiments and manage complex workflows [11]. Traditional scientific research has long relied on manual processes where researchers design experiments, collect data, analyze results, and make decisions about the next steps in a largely sequential and time-consuming manner. AI agents for autonomous discovery represent a paradigm shift by embedding intelligent decision-making capabilities directly into the experimental process, allowing systems to adapt, learn, and optimize research strategies in real-time [7]. These agents integrate with lab automation, computational resources, and instruments to form closed-loop systems that can hypothesize, test, and refine scientific understanding with minimal human intervention. Recent advances in chat-based large language models (LLMs) further enhance this paradigm by providing researchers with intuitive, natural language interfaces to interact with autonomous systems [13]. LLM-powered assistants can support the entire experimental lifecycle, from planning and resource coordination to real-time monitoring and result interpretation. Their ability to reason over scientific literature, datasets, and provenance data makes them valuable collaborators in guiding decision-making and accelerating iterative experimentation. This new class of intelligent systems tackles key bottlenecks in discovery, including data growth, complex optimization, and the need for reproducible, scalable methods.

3 ARCHITECTURE OVERVIEW

Figure 1 illustrates the architecture, which consists of the following core components: (1) an LLM chat assistant to allow scientists to drive their experiments using natural language; (2) a messaging system to enable communication and data exchange between the various services deployed on the Edge, Cloud, and HPC; (3) Facility APIs to enable resource leasing on instrument and computing facilities; (4) a multi-workflow provenance system to enable cross-facility provenance data capture and to support insightful analysis throughout the experimental process; (5) a multi-agent system for managing a group of domain-expert AI agents responsible for making decisions at runtime; and (6) an orchestrator service for coordinating the experiment execution workflow.

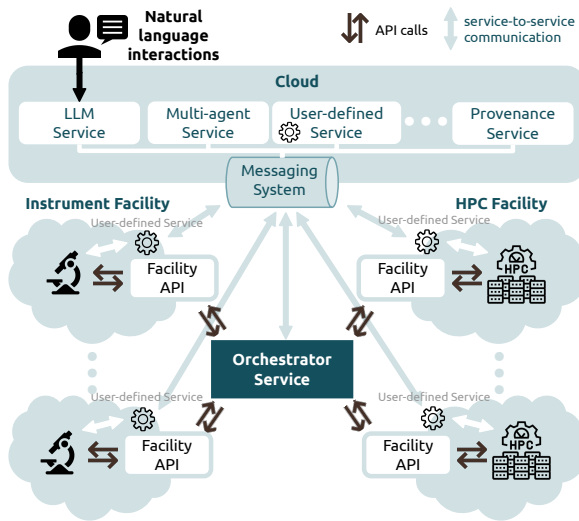


Figure 1: System architecture for autonomous cross-facility experimentation that integrates LLM-driven user interaction, multi-agent decision-making, facility APIs, and provenance capture across instrument, cloud, and HPC environments.

LLM Chat Assistant. In our architecture, the LLM chat assistant serves as an entry point for scientists to interact with complex cross-facility experiments using natural language. Rather than requiring deep familiarity with the underlying infrastructure or scripting, users can define goals, refine experimental parameters, and query results conversationally. The assistant is powered by a domain-adapted large language model and operates as part of the human-machine interface (HMI) agent. The assistant translates high-level scientific intent into structured tasks, enabling seamless coordination with downstream agents and services. It can access metadata, past experiment records, and relevant literature to provide context-aware guidance throughout the experimental lifecycle. Integrated with the provenance system, it also allows users to issue traceability queries, explore agentic decisions, and monitor workflow progress in near real-time. This capability positions the LLM assistant as an intelligent collaborator that enhances transparency, accessibility, and agility in agent-driven autonomous experimentation.

Messaging System. The messaging system is the core component that enables communication among instruments, sensors, and compute resources, while also allowing users to discover these resources and understand their interfaces. The messaging system is the backbone to establish an ecosystem of network-connected yet loosely-coupled systems shown in the lower portion of Figure 1 and allows for other services (i.e., LLM chat, multi-agent, provenance) to interact with this ecosystem. As the user communicates via the natural language interface, the LLM chat agent constructs a workflow that is submitted over the messaging system to the various connected services. During the autonomous experiment, the messaging system transports both results and inputs as service messages between components.

Facility API. Large scientific computing clusters have traditionally required manual, session-based access via SSH, where users authenticate, prepare batch scripts, and submit jobs to a scheduler such as Slurm. These sessions expire, interrupting work and requiring repeated manual logins. The Facility API model addresses these limitations by exposing secure, remotely accessible endpoints for interacting with HPC resources [4, 15, 18]. This approach enables interoperability across facilities, promotes code and workflow reusability, and supports modular, service-oriented architectures. It strengthens security by enforcing consistent access controls and auditability, while also enabling automation for data movement, job management, and monitoring. Facility APIs transform clusters from isolated systems into a seamlessly integrated scientific workflow fabric, thereby allowing researchers to have their work automatically stretch across institutional and geographic boundaries.

Multi-workflow Agentic Provenance System. Provenance data management is essential for understanding experimental data, validating hypotheses, monitoring, steering, and ensuring reproducibility [16]. As workflows span federated environments across the edge–cloud–HPC continuum, we need flexible and adaptable systems capable of capturing fine-grained provenance, including agentic interactions. Agents now play a central role by generating plans, making decisions, and reasoning about execution and dataflow, introducing new challenges around accountability and transparency. Given the non-deterministic nature of these agents, it is critical to track their actions and downstream workflow effects. Agentic provenance data serve as the glue that links diverse execution traces, data products, and decisions across the workflow [17]. An effective provenance system must represent these relationships to build a coherent and contextualized provenance graph, enabling deep, broad insights and supporting trust, reproducibility, and responsible integration of agents in scientific workflows.

Multi-agent System. Multi-agent systems represent a distributed AI paradigm where autonomous software entities collaborate to solve complex problems that exceed the capabilities of individual agents. Each agent possesses specialized knowledge and decision-making abilities, enabling the system to decompose intricate challenges into manageable, domain-specific tasks. Multiple agents are necessary because real-world problems often span multiple disciplines, require parallel processing, and benefit from distributed expertise that no single agent could efficiently encompass. In our AM workflow, this framework is especially valuable due to process

complexity and interdependence. Specialized agents coordinate design optimization, material selection, process planning, real-time quality monitoring, and post-processing operations. During experiment runtime, agents communicate through structured protocols and shared knowledge repositories, enabling dynamic adaptation when conditions change. For example, quality monitoring agents detecting print anomalies immediately trigger process parameter adjustments while updating downstream scheduling agents. This distributed intelligence ensures robust workflow management capable of handling manufacturing variability and complexity. The system’s modular architecture facilitates extension beyond AM to subtractive processes, assembly operations, and broader industrial automation applications.

Orchestrator Service. This service is responsible for enforcing the experiment settings from the LLM chat assistant. The orchestrator uses the various Facility APIs to make scheduling decisions, synchronize resource leasing on the HPC and instrument facilities, deploy supporting services, and trigger the job execution. Supporting services include data streaming, which enables feedback loops between services across facilities, and storage for preserving provenance and experimental data. The orchestrator also communicates with user-defined services to enforce their tasks on the HPC and instrument facilities and to forward data between services placed in different facilities.

4 ARCHITECTURE IMPLEMENTATION

This section details the implementation of the proposed architecture using ORNL technologies and initiatives to support autonomous, cross-facility experimentation (Figure 2).

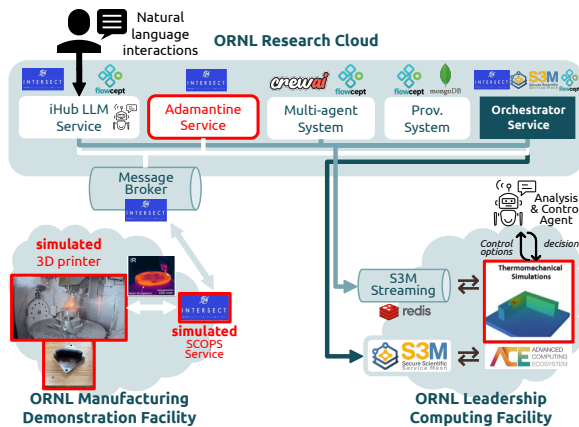


Figure 2: Architecture implementation. Our architecture can be generalized to other use cases. Highlighted in red color are the components specific to the Additive Manufacturing (AM) use case. We simulated the 3D printer and SCOPS service.

iHub HMI Agent. The HMI agent applications are accessible via iHub, a web-based platform that serves as a gateway into INTERSECT. iHub offers a range of services, including the Capability Store, which enables users to search for and share a variety of resources. These resources include INTERSECT capabilities (e.g.,

laboratory tools and computational assets) and standalone applications. Through the Capability Store, users can discover and install HMI agent applications, such as chat assistants, which are then integrated into their personalized dashboards (Figure 3). The HMI agent is comprised of multiple components and libraries. The underlying LLM is hosted by Ollama¹, and currently utilizes the *llama3.2:latest* model to power the LLM Chat Assistant. The chat interface is developed using Next.js² and the Mantine³ component frameworks. The application leverages the LangChain framework to interface with the Ollama API and deliver the initial prompt. Additionally, LangChain enables tool invocation through adapters that support Model Context Protocol (MCP)-compliant functions.

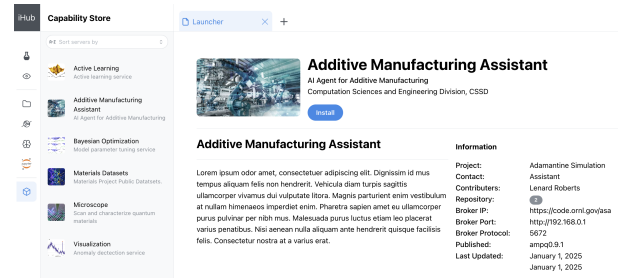


Figure 3: iHub Capability Store showing the AM Assistant (HMI agent) profile.

INTERSECT. The INTERSECT platform allows for the creation of an ecosystem of complex, interacting instruments, sensors, and compute resources by providing tools for science teams to develop small, well-defined systems that connect on a common platform. This provides the foundation for a science ecosystem, a network connected collection of loosely-coupled services that are maintained and operated independently. Figure 2 is an example of an INTERSECT instance with a common platform for communication through the message broker and interactivity via the iHub web platform and then the independent services indicated by the blue INTERSECT logo for the SCOPS service, the adamantine simulation service, and the S3M service. For the systems used in this work, the Python INTERSECT Software Development Kit v0.8 was used [3].

S3M Facility API. The Secure Scientific Service Mesh (S3M) [15] provides the foundational infrastructure for enabling agentic, cross-facility workflows at ORNL by exposing secure, programmable APIs to computational and experimental resources. Designed to support both human and machine clients, S3M abstracts the complexity of facility integration through a service mesh architecture that enforces fine-grained access control, project-scoped authentication, and dynamic resource provisioning. Its support for high-throughput data streaming and workflow orchestration allows intelligent agents to trigger compute jobs in response to live experimental results or predictive insights, enabling near real-time decision-making and closed-loop automation. The S3M Streaming API provisions messaging services (e.g., RabbitMQ, Redis) in secure environments

¹<https://ollama.com>

²<https://nextjs.org>

³<https://mantine.dev>

co-located with HPC resources, eliminating latency bottlenecks and manual configuration overhead. The Workflow API integrates with Argo Workflows to support multistep execution pipelines with parallelism, data dependency resolution, and fault tolerance.

Flowcept Agentic Provenance System. Flowcept is a lightweight, scalable provenance system for capturing and contextualizing fine-grained workflow and agentic data across distributed environments [16]. It uses a broker-based architecture, where components and agents emit provenance messages to a central broker. A centralized Provenance Keeper curates the data and applies a W3C PROV-compliant agentic schema [17], linking data products, execution traces, and agent decisions into a coherent, queryable graph. Its modular design supports integration across the edge–cloud–HPC continuum and enables near real-time or retrospective data analysis. Beyond provenance capture, Flowcept plays a central role in unifying the architecture. Provenance data acts as the glue across distributed components, ensuring all workflow tasks use a consistent schema and enabling inter-component communication through brokered provenance streams. This establishes a lightweight, traceable protocol that supports accountability, transparency, and reproducibility in loosely coupled agentic workflows.

CrewAI Multi-agent System. We present a CrewAI⁴-enabled multi-agent system for intelligent decision-making in AM. This architecture integrates autonomous agents, human expertise, and simulation environments to orchestrate optimal control across complex 3D printing workflows. By leveraging CrewAI's Crews and Flows, we achieve both agentic autonomy and fine-grained execution control within a robust, production-grade environment. At the heart of the framework is a CrewAI "Crew" composed of specialized agents that collaborate through role-based delegation. The Analysis and Control (A&C) Agent acts as the coordinator, integrating inputs from simulations, sensors, and human instructions via the HMI Agent. This natural language interface ensures intuitive interaction and enables near real-time updates from human-in-the-loop operators. The A&C Crew consists of (Figure 4): a (1) **Control Option Agent** that generates parameterized control options; a (2) **Control Decision Agent** to select optimal strategies informed by experimental context; a (3) **Literature Discovery Agent**, which retrieves relevant scientific literature; a (4) **Citation Intent Agent** to assess the supportiveness of references; and a (5) **Verification & Validation Agent** to ensure safety and parameter compliance. These agents communicate via A2A protocol while MCP bridges external systems and data flows. CrewAI *Flows* define the procedural logic, ensuring conditional branching, safety checks, and controlled task execution. This enables dynamic adaptation to real-world events while preserving autonomy and system integrity.

HPC Orchestrator. The HPC orchestrator leverages INTERSECT, the S3M Facility API, and the Flowcept provenance system, communicating with the LLM chat assistant and user-defined services via INTERSECT messages. To allow this, it exposes capabilities (e.g., INTERSECT capability implementations) to trigger tasks like the simulation execution on HPC clusters, to forward data returned

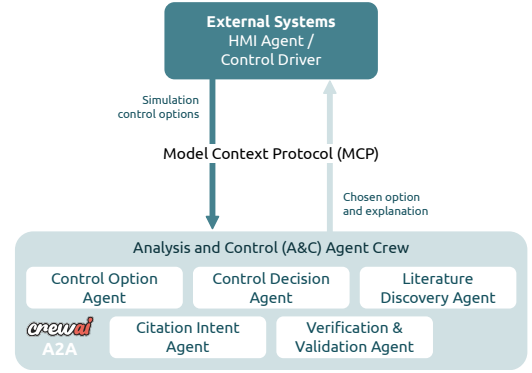


Figure 4: Multi-agent system for analysis and control in the additive manufacturing (AM) workflow, showing the interactions between specialized agents and the HMI.

from simulations to user-defined services running near the instruments, among other tasks. The experiment settings defined by users on the LLM chat assistant reach the orchestrator as INTERSECT messages (e.g., the HPC cluster to be used, the number of computing nodes, the simulation settings, among others). The orchestrator uses the S3M API to lease the HPC nodes and then submit the simulation job. In addition, the S3M API is used to provision the data streaming service to allow data flow between the HPC simulation and the scientific instrument. Provenance data from the message exchanges between the HPC simulation, the user-defined services, and AI agents are captured by the orchestrator via Flowcept.

5 CROSS-FACILITY EXPERIMENT

In this section, we evaluate the architecture in a realistic end-to-end workflow for model predictive control [14] of an AM print that employs a simulated version of resources at the MDF, ORC, and ACE. The setup is shown in Figure 2.

5.1 Experimental Setup: MDF, Cloud, and HPC

As this work is in progress toward a fully connected real-world experiment, the SCOPS service and 3D printer are represented by simulated components. At MDF, a simulated SCOPS service handles interactions with the simulated 3D printer. ORC hosts a virtual machine running the HMI agent, CrewAI multi-agent system, Flowcept provenance system, Adamantine service, and the HPC orchestrator, with the INTERSECT message broker deployed on a separate Kubernetes cluster⁵. For the HPC component, we use the Quokka system from the ACE testbed, comprising 16 general-purpose Intel Xeon compute nodes connected via NDR200 InfiniBand. On Quokka, the Adaptive Control Driver Apptainer container executes Adamantine [20] simulations, while the S3M Redis streaming service is provisioned in the OLCF data stream cluster.

5.2 Experiment Workflow

We next describe the runtime message exchanges in the end-to-end cross-facility experiments, as illustrated by the simplified sequence diagram in Figure 5.

⁴<https://github.com/crewAIInc/crewAI>

⁵<https://kubernetes.io/>

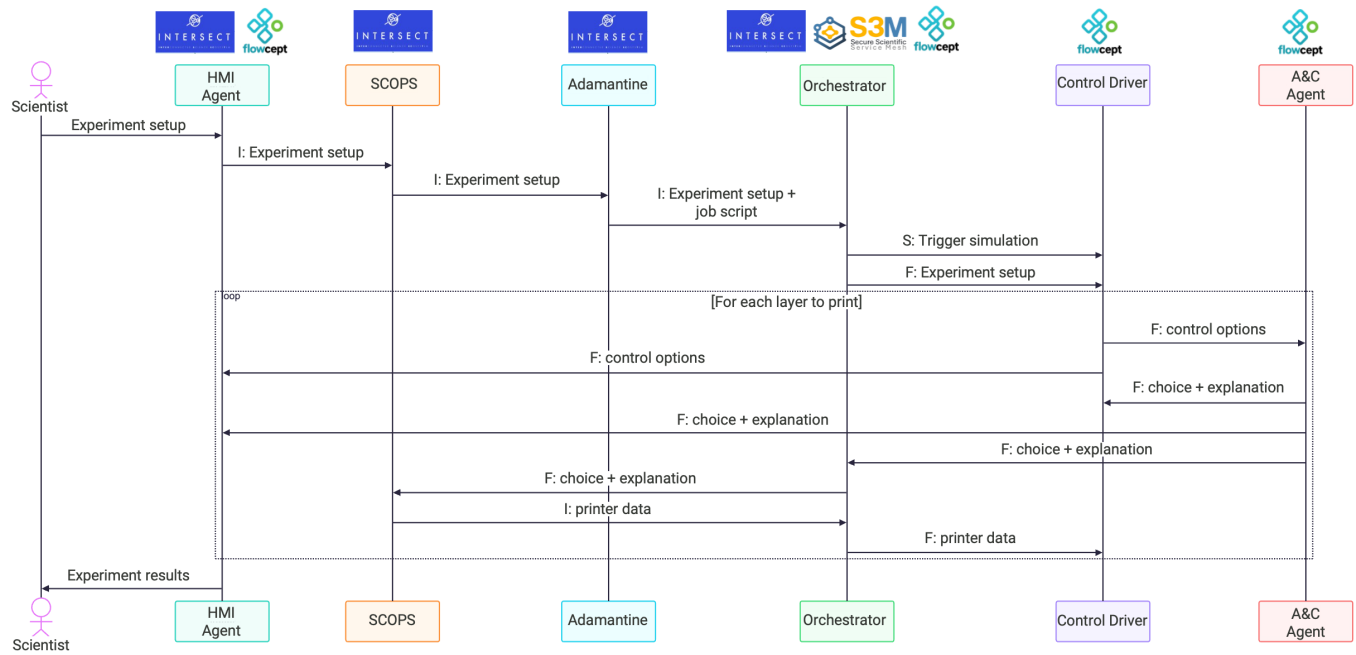


Figure 5: Simplified Sequence Diagram: In the inter-service communications, I: represents INTERSECT messages; F: represents Flowcept messages; S: represents S3M API calls.

Scientist and HMI Agent Interactions. Through the iHub marketplace, the scientist selects the AM Assistant profile, as illustrated in Figure 3. The HMI agent has a knowledge base of pre-optimization steps for metal component parts. The interaction between the scientist and the HMI Agent is illustrated as follows:

HMI Agent: Greeting and Material Selection

Hello Dr. Smith I am your additive manufacturing workflow assistant. According to my knowledge base, I am aware that pre-optimization steps have run for five relevant prints. Here are the available prints:

ID	Material Name
1	Dwell Rook (17-4PH Stainless Steel)
2	Dwell Rook (410 Stainless Steel)
3	Dwell Rook (316L Stainless Steel)
4	Curved Wall (316L Stainless Steel)
5	Sub-scale Runner (316L Stainless Steel)

You can ask for more details about these prints or we can begin to print any of these autonomously. Please select a material to simulate.

Scientist: 2

HMI Agent: Print Selection Confirmation and Simulation

You have selected Dwell Rook (410 Stainless Steel). Sure, the pre-optimization step is related to a print job **consists of 5 layers**. Would you like to start the print? If so, how many control options do you want to simulate for each layer? Please enter a number:

Scientist: 4

HMI Agent: Print Simulation Confirmation and Start

Ok, I can start the control driver for this autonomous print on the Quokka system from OLCF.

Do you have any guidance before we start? For instance, are there constraints we should keep in mind before choosing options for each layer?

Please confirm if you would like to proceed with the print simulation.

Scientist: Yes. If scores are fairly close, choose the one that has lowest dwells.

HMI Agent: Summary of Print Settings

Here is a summary of your print settings:

ID: 2

Material Name: Dwell Rook (410 Stainless Steel)

Number of Control Options: 4

Number of Layers: 5

Please confirm that this is correct. If you are ready to start the print, please respond with "start".

Scientist: This is correct. Start.

Simulation and Printer Loop. Once the scientist confirms the start of the experiments, the HMI Agent sends an INTERSECT message to the SCOPS service containing the experiment setup, including the number of layers to print (five), the component to

be printed (Dwell Rook 316L Stainless Steel), and the number of control options to simulate for each layer (four). The SCOPS service then relays a message to the SCOPS client to start the 3D printer and also sends a message to the Adamantine service. The Adamantine service forwards an INTERSECT message to the HPC Orchestrator service with the experiment setup and a submission job script specifying parameters such as job duration, required compute nodes, and execution commands. The Orchestrator service uses the S3M API to provision the Redis streaming service, submit the simulation job to the Quokka system, and check job status before forwarding the experiment setup.

During execution, the Control Driver simulation connects to the S3M Redis service to consume and publish data. It ingests in-situ characterization data from the 3D printer and outputs simulation results, such as parameters for the next layer to be printed. For each simulation layer, it sends four control options with associated scores to the Analysis and Control Agent, which selects the best option and returns it to the Control Driver for continued simulation. The chosen option and its explanation are also sent to the Orchestrator. The selected simulation output is then sent via the S3M streaming service and INTERSECT messages to the HMI Agent (allowing the scientist to monitor the process layer by layer), and the SCOPS service to configure the 3D printer for the next layer. The 3D printer generates new in-situ characterization data, this data is forwarded by the SCOPS and Adamantine services to the orchestrator, and finally to the control driver to start a new simulation. This loop repeats until all five layers are printed, after which the HMI GUI notifies the scientist that the print job is complete.

The Flowcept Agent. This component captures agent decisions, actions, and interactions as structured provenance messages. Since all architectural components embed provenance capture, these messages are continuously streamed through the broker, enabling agents and components to communicate via a shared, provenance-aware messaging protocol. This protocol ensures that agentic interactions remain traceable, accountable, and transparent across the workflow. Agentic provenance is integrated with data from non-agentic tasks in a unified provenance graph, enabling scientists to issue domain-specific queries. Next, we present query examples related to the printing process, expressed in natural language.

Scientist: What are the activities in this workflow?

Flowcept Agent: The activities in this workflow are:

ID	activity
0	publish_experiment_setup
1	call_generate_options_set
2	llm_interaction
3	generate_options_set
4	simulate_layer
5	call_choose_option
6	choose_option

Scientist: Which activities are associated with agents?

Flowcept Agent: The activities associated with agents are:

ID	activity
2	llm_interaction
3	generate_options_set
6	choose_option

Scientist: for the *choose_option* activity, what are the inputs and outputs?

Flowcept Agent: The are the inputs and outputs:

layer	planned_controls	control_options	scores	option
2	['power': 321, 'dwell_0': 30, 'dwell_1': 25, 'power': 101, 'dwell_0': 35, 'dwell_1': 115]	['power': 110.0, 'dwell_0': 40, 'dwell_1': 110, 'power': 245.0, 'dwell_0': 115, 'dwell_1': 110]	[58, 57, 63, 69]	3
3	['power': 321, 'dwell_0': 30, 'dwell_1': 25, 'power': 101, 'dwell_0': 35, 'dwell_1': 115]	['power': 240.0, 'dwell_0': 110, 'dwell_1': 115, 'power': 150.0, 'dwell_0': 95, 'dwell_1': 100]	[77, 50, 20, 29]	2
4	['power': 321, 'dwell_0': 30, 'dwell_1': 25, 'power': 101, 'dwell_0': 35, 'dwell_1': 115]	['power': 325, 'dwell_0': 115, 'dwell_1': 100, 'power': 135, 'dwell_0': 110, 'dwell_1': 110]	[29, 24, 36, 84]	3
5	['power': 321, 'dwell_0': 30, 'dwell_1': 25, 'power': 101, 'dwell_0': 35, 'dwell_1': 115]	['power': 190.0, 'dwell_0': 80, 'dwell_1': 85, 'power': 180.0, 'dwell_0': 90, 'dwell_1': 90]	[52, 40, 5, 31]	2
6	['power': 321, 'dwell_0': 30, 'dwell_1': 25, 'power': 101, 'dwell_0': 35, 'dwell_1': 115]	['power': 230, 'dwell_0': 25, 'dwell_1': 100, 'power': 240, 'dwell_0': 30, 'dwell_1': 90]	[97, 73, 44, 84]	0

Scientist: For the *choose_option* activity, when the input was layer 2, what were planned controls, control options, scores, option, and explanation?

Flowcept Agent: For layer 2, the planned controls, control options, scores, option, and explanation are:

layer	planned_controls	control_options	scores	option
2	['power': 321, 'dwell_0': 30, 'dwell_1': 25, 'power': 101, 'dwell_0': 35, 'dwell_1': 115]	['power': 110.0, 'dwell_0': 40, 'dwell_1': 110, 'power': 245.0, 'dwell_0': 115, 'dwell_1': 110]	[58, 57, 63, 69]	3

Explanation: The lowest score of 58 indicates the best quality outcome, suggesting optimal setting.

6 RELATED WORK

Recent research has increasingly focused on leveraging AI agents to assist scientists in experiments that integrate both experimental and computational facilities. One example is VISION [9], a modular architecture that assembles LLMs into task-specific cognitive blocks, enabling scientists to interact with instruments using natural language. Designed for synchrotron beamlines, VISION facilitates voice-controlled experimentation and adaptive workflow augmentation, allowing experiments to be dynamically adjusted in response to real-time feedback. Another representative system is Academy [11], a middleware platform that enables the deployment of autonomous agents across federated scientific infrastructure,

including experimental laboratories and HPC facilities. Academy overcomes limitations in existing agent frameworks by providing abstractions for agent state, behavior, asynchronous communication, and dynamic coordination. Its flexibility is demonstrated through case studies in materials discovery, decentralized learning, and large-scale information extraction, showcasing the breadth of domains that can benefit from agent-based coordination.

Other work explores context-aware integration of LLMs with facility-specific tools and data resources to improve the reliability and usability of AI-driven experimentation. CALMS [12] exemplifies this approach by combining LLMs with semantic search and tool execution capabilities to assist with experimental design, instrument operation, and automated execution through direct API and hardware interfaces. Applied to x-ray, nanoscience, and computing facilities, CALMS demonstrates how retrieval-augmented methods can reduce hallucinations, provide accurate context, and improve decision-making in complex scientific workflows. Together, these systems illustrate the growing importance of agentic AI in scientific experimentation and point toward opportunities for convergence with federated, cross-facility workflows such as those targeted by our architecture, where interoperability, provenance tracking, and adaptive orchestration are central to accelerating discovery.

7 CONCLUSION AND FUTURE WORK

We presented a novel architecture for connecting experimental scientific facilities with HPC for autonomous scientific workflows managed by a collection of AI agents. This architecture consists of six core components: a LLM chat assistant, a messaging system, facility APIs, a provenance system, a crew of AI agents, and an orchestrator service. This architecture is designed to permit dynamic near-real-time decision making, based on both human and artificial intelligence. We also presented a preliminary reference implementation of this architecture using iHub, INTERSECT, S3M, Flowcept, and CrewAI. This reference implementation was deployed in the context of a cross-facility experiment between ORNL's Manufacturing Demonstration Facility (MDF) and ORNL's Oak Ridge Leadership Computing Facility (OLCF) for a metal additive manufacturing process steered using an agent-augmented version of model predictive control. This demonstration employs a simulated version of the sensors and printer at the MDF, but otherwise deploys the core components of the architecture in a joint Edge-Cloud-HPC workflow. By tying this demonstration to a real use case, we highlight the applicability of the proposed architecture to the complexities of scientific workflows.

The work presented here provides a concrete framework for building AI-agent-managed workflows that bridge multiple scientific facilities. The natural extension of this work is a demonstration of the full end-to-end workflow connected to the sensors and printer at the MDF. Beyond that demonstration, deployment for other types of autonomous experiments is crucial to validate the generalizability of the architecture and our implementation. Such deployments should also include cross-institutional to evaluate the ability for this architecture to function with even weaker ties between workflow components.

This approach presented here directly addresses the growing complexity of modern scientific challenges that require coordination across disciplines, facilities, and institutions without sacrificing security or performance. The capacity for AI agents to dynamically orchestrate scientific experiments, adapt to real-time data, and infuse human decision-making with tool-based rigor offers the potential to dramatically accelerate scientific progress. As this vision is realized through future deployments and expanded use cases, the true impact of this architecture lies in its ability to broaden access to cutting-edge tools and unlock scientific insights that would have been out of reach without an integrated approach.

Acknowledgment. This research used resources of the Oak Ridge Leadership Compute Facility and ORNL Research Cloud Infrastructure at ORNL, which is supported by DOE's Office of Science under Contract No. DE-AC05-00OR22725.

REFERENCES

- [1] Katerina B Antypas, et al. 2021. Enabling discovery data science through cross-facility workflows. In *IEEE Big Data*.
- [2] Stephanie Cooper et al. 2024. *FY 2024 Multidimensional Data Correlation Platform: Unified Software Architecture For Advanced Materials And Manufacturing Technologies Data Management And Processing*. Technical Report. Oak Ridge National Laboratory (ORNL).
- [3] Lance Drane et al. 2024. INTERSECT-SDK (Python). <https://doi.org/10.11578/dc.20240927.1>.
- [4] Bjoern Enders et al. 2020. Cross-facility science with the Superfacility Project at LBNL. In *IEEE/ACM Workshop on Extreme-scale Experiment-in-the-Loop Computing (XLOOP)*.
- [5] ESnet. 2024. Jefferson Lab, ESnet complete test of cross-country experimental data flow to HPC centers. <https://www.es.net/news-and-publications/esnet-news/2024/ejfat-jefferson-lab-esnet/>.
- [6] Brian D Etz et al. 2025. Enabling Seamless Transitions from Experimental to Production HPC for Interactive Workflows. *arXiv:2506.01744* (2025).
- [7] Rafael Ferreira da Silva et al. 2025. A Grassroots Network and Community Roadmap for Interconnected Autonomous Science Laboratories for Accelerated Discovery. In *International Conference on Parallel Processing Companion*.
- [8] James Haley et al. [n. d.]. In-situ digital image correlation and thermal monitoring in directed energy deposition additive manufacturing. *Opt. Express* ([n. d.]).
- [9] Shray Mathur et al. 2025. VISION: a modular AI assistant for natural human-instrument interaction at scientific user facilities. *Machine Learning: Science and Technology* (2025).
- [10] Sarp Oral et al. 2024. *OLCF's advanced computing ecosystem i(ACE): FY24 Efforts for the DOE integrated Research Infrastructure (IRI) program*. Technical Report. Oak Ridge National Laboratory (ORNL).
- [11] J Gregory Pauloski et al. 2025. Empowering Scientific Workflows with Federated Agents. *arXiv:2505.05428* (2025).
- [12] Michael H Prince et al. 2024. Opportunities for retrieval and tool augmented large language models in scientific facilities. *npj Computational Materials* (2024).
- [13] Chandan K Reddy and Parshin Shojaei. 2025. Towards Scientific Discovery with Generative AI: Progress, Opportunities, and Challenges. *AAAI Conference on Artificial Intelligence* (2025).
- [14] Max Schwenzer, Muzaffer Ay, Thomas Bergs, and Dirk Abel. 2021. Review on model predictive control: an engineering perspective. *The International Journal of Advanced Manufacturing Technology* (2021).
- [15] Tyler J. Skluzacek et al. 2025. Secure API-Driven Research Automation to Accelerate Scientific Discovery. In *Practice and Experience in Advanced Research Computing (PEARC)*.
- [16] Renan Souza et al. 2023. Towards lightweight data integration using multi-workflow provenance and data observability. In *IEEE e-Science*.
- [17] Renan Souza et al. 2025. PROV-AGENT: Unified Provenance for Tracking AI Agent Interactions in Agentic Workflows. In *IEEE e-Science (accepted)*.
- [18] Joe Stubbs et al. 2021. Tapis: An API platform for reproducible, distributed computational research. In *Future of Information and Communication Conference*.
- [19] Addi Malviya Thakur et al. 2023. Towards a Software Development Framework for Interconnected Science Ecosystems. In *Smoky Mountains Computational Sciences and Engineering Conference (SMC)*.
- [20] Bruno Turckin and Stephen DeWitt. 2024. Adamantine 1.0: A Thermomechanical Simulator for Additive Manufacturing. *Journal of Open Source Software* (2024).
- [21] Nicholas Tyler et al. 2022. Cross-facility workflows: Case studies with active experiments. In *IEEE/ACM Workshop on Workflows in Support of Large-Scale Science (WORKS)*.