

## **Topic: Tools, Interfaces, and Integration Surfaces**

### **0. Introductions (10 min)**

- *Including quick perspective/experience/struggles with Agentic tools and Interfaces*

### **1. Framing & Alignment (10 min)**

- *Background including “Intelligent Interfaces” activities in American Science Cloud*
- *Set scope: categories of tools and interfaces*
- *Clarify outputs and expectations*

### **2. Parallel Group Discussion (60 min)**

- *1 big group*
- *All groups discuss **all charge questions**:*
  - *Tools available today; What tools are needed*
  - *Gaps and development needs; Emerging design directions*
  - *How support genesis mission - near term needs and demonstrations*
- *Focus on concrete examples and transferable insights*

### **3. Report-Outs (10 min)**

- *Each group highlights key takeaways*
- *Emphasis on commonalities and differences*

### **4. Full-Room Synthesis (20 min)**

- *Identify recurring patterns and gaps*
- *Converge on a shared architectural picture - **most promising tools and directions of deployment or development activity needed***

### **5. Roadmap & Alignment (20 min)**

- *Define 6–18 month milestone(s)*
- *Connect outcomes to Genesis / cross-facility goals*

### **6. Wrap-Up (10 min)**

- *Lock in breakout outputs*
- *Confirm next steps and follow-up*

---

**Expected Outputs**

- *Near-term agentic tools and interface capabilities that work today*
- *Requirements from tools*
- *Gaps in tooling and areas of development needed by DoE*
- *Reusable tools and interface and design direction*
- *One concrete milestone for the next 6–18 months to work together on.*

## Notes

### Introductions

#### Struggles / Perspectives

- Shreyas: Keeping up with stack when it is changing so fast
- Steve: Building tools / MCP services;
  - 3rd party tools may not do quite what you want.
  - Looking at smolagents vs langchain
  - Skills.md
- Tazai (Diamond UK)
  - Just started playing with Tools
- Qing Ji LBL Accelerator
  - Interface between agent and machine for low latency operations
- Gopika Software Engineer at SLAC
  - Playing with Osprey
  - Claude / ChatGPT for debugging
- Sara, SLAC
  - Ops, surrogate modeling / DTs
  - Played with Osprey, using agents in workflows
  - Interested in agentic optimization, need better tools to set up
- Dyland McReynolds, ALS
  - Data platform, UIs
  - 2 projects with Osprey
    - Beamline in a box, bluesky test platform
    - Demo beamline for playing, shared
  - Everyone in community wants to do agentic AI
  - Hooking agents into bluesky api
- Erik Wallen accelerators, ALS-U
  - Uses gemini for analysis
  - Auto-repo generation
- Andrew Naylor, NERSC
  - Standards evolving. Once standards develop, should be able to coalesce.
- Alok, UChicago phd student

- Agents writing code can be frustrating when you need to edit; they rewrite a lot from scratch; leads to lots of bespoke frameworks for doing things that we already can do well.
- Agents build code / tools have a lot of things that may have better external libraries.
- Fragile, patching mistakes indefinitely, particularly when you change model.
- Matt, STFC Oxford,
  - Early optimistic adopter of copilot, Q&A and debugging,
  - created simple UI, biggest regret; caused more problems downstream
  - Now use agents to extend functionality, not from scratch.
  - Dabbled with self-hosted models, llama, gpt-oss, mcp tools
  - Anomaly detection/ranking report generation
  - Field changing so rapidly, e.g. learned about Skills today.

## Open discussion

### What do we mean by Tools and Services

- AMSC Stack
- Using full featured Agentic CLI tools
  - Claude, Goose etc.
- Coding agents are more flexible and can be used for a lot
- Model DBs
  - MLFlow
  - ClearML
- What is the boundary between autonomous lab interface and the agent?

### Gaps, where agents fail, tools needed

- Need approaches to capture institutional knowledge, documentation about what previous scientists have done, logging everything, capturing what was done in jupyter notebooks
- Agents should capture their own provenance, reasoning traces etc.
  - Save data from optimization
- Provenance for what the humans are doing as well. Agent that can automatically log everything, possibly via voice conversation.
  - AmSC should help standardize the way this data is collected, stored, retrieved.
  - Question - what is the best way to provide agent memory / context for larger volumes of data for retrieval
  - Need AI-ready metadata, standards for provenance capture
- Need standards for metadata
- What's the best way to retrieve facility info/data for an agent?
  - Context engineering
  - RAG

- At what point should you use fine-tuning?
- Eg, Memory augmented retrieval - <https://arxiv.org/abs/2508.16153>
- FNAL ACL legacy code translation
- Need for low-latency solutions, e.g. in accelerators, fusion experiments

Use of agentic CLI tools that act as coordinators (vs. custom Langchain etc.)

- Osprey does this (similar to Claude CLI etc)
- ALS users write a lot of MCP servers for specific functionality

MCP

- Need for an agent registry / MCP Tool registry
- Is MCP is overkill? Are people just using @tools?
- Alternative to MCP - point to openapi.json+fastapi (w skills) instead of MCP

Genesis Needs

- Can there be canned examples or documentation for how to build an agentic workflow that integrates into various models?
  - Tutorials, Deep Dives for how we do beamline X with an agent and how you can get started building something similar
  - Lessons learned, Best practices etc.
- Having one place to go for central information

-----

From the AI Assistant

From the discussion what are gaps in tooling and areas of development needed by DoE

04:06 PM

**AI Companion Said:**

Based on the discussion, several key gaps in tooling and areas of development were identified for DOE facilities:

**Documentation and Knowledge Capture:**

- Lack of organized code bases and documentation for experimental workflows
- Institutional knowledge exists in people's brains or scattered Jupyter notebooks
- Need for automated provenance tracking and logging of what scientists do during experiments (especially non-routine tasks)
- Better capture of optimization strategies, debugging steps, and reasoning processes

**Metadata and Standardization:**

- Need for standardized metadata capture across facilities (currently in paper notebooks)
- Electronic lab notebook (ELN) systems that are actually used and enforce standards
- ML-ready data with proper metadata for training models
- Standardization of data formats across different DOE facilities

#### **Agent Memory and Learning:**

- Context engineering challenges - how to provide agents with relevant past experiences without overwhelming context windows
- Need for scalable solutions (RAG systems, fine-tuning) to help agents learn from mistakes
- Continuous learning capabilities so agents don't repeatedly make the same errors

#### **Low-latency Requirements:**

- Need for different latency levels - from microsecond responses (plasma control, accelerator stability) to longer decision-making
- Edge computing capabilities for real-time control and anomaly detection
- Integration between high-level AI agents and low-latency local control systems

#### **Interface Development:**

- Tools to bridge lab control systems (EPICS, Blue Sky) with agentic frameworks
- Better integration between existing APIs and agent systems
- Human-in-the-loop approval mechanisms for critical operations

What is worth mentioning in terms of what people are doing today

04:12 PM

#### **AI Companion Said:**

Based on the discussion, here are notable current practices across DOE facilities:

#### **Agentic Tools and Frameworks:**

- **Osprey** is widely adopted at accelerator facilities and beamlines, using LangChain for autonomous optimization and control
- People are building **MCP (Model Context Protocol) servers** extensively at ALS beamlines for everything from controls to analysis
- Use of **coding agents** like Claude CLI, Cursor, and Copilot for development assistance
- **Open Web UI** being integrated with control systems for user interfaces

#### **Practical Applications:**

- **NERSC** developing MCP tools for HPC operations (ServiceNow integration, job submission APIs)
- **SLAC** deploying surrogate models and using XOPT for online optimization with agents
- **ALS** created "Beamline in a Box" - a safe training environment with EPICS and Blue Sky for testing agentic AI

- Automated report generation and routine task automation using ChatGPT/Gemini

**Control Systems Integration:**

- Using **EPICS** with Blue Sky for instrument control
- Developing FPGA-based ultra-precision control systems for lasers and accelerators
- Real-time anomaly detection and machine learning for stability improvements (e.g., laser stabilization)

**Challenges Being Addressed:**

- Rapid tool evolution making it difficult to standardize (frameworks change frequently)
- Agents writing bespoke code that duplicates existing infrastructure
- Fragile context engineering requiring constant corrections
- Legacy software compatibility issues with LLMs

From me: What could be a Genesis-Relevant Demonstration (6–18 Months) that can be demoed at one or more (user/science/compute) facilities

What could be a Genesis-Relevant Demonstration (6–18 Months) that can be demoed at one or more (user/science/compute) facilities

04:14 PM

AI Companion Said:

**Genesis-Relevant Demonstration (6–18 Months)**

**Autonomous Beamline Optimization Agent with Cross-Facility Learning**

**Demonstration Scope:**

- Deploy an AI agent that autonomously optimizes experimental parameters at a beamline (ALS or SLAC)
- Agent learns from past experiments across multiple sessions and users
- Integrates with existing control systems (EPICS, Blue Sky) via MCP servers
- Includes human-in-the-loop approval for critical changes

**Key Components:**

- **Memory system** using RAG to recall successful optimization strategies from previous experiments
- **Standardized metadata capture** during runs for building institutional knowledge
- **Safe testing environment** using "Beamline in a Box" concept before live deployment
- **Real-time performance:** Achieves optimization in minutes vs. hours of manual tuning
- **Documentation:** Automatically generates experiment reports with reasoning traces

**Success Metrics:**

- Reduces experiment setup time by 50%

- Successfully transfers learned strategies between similar beamlines
- Demonstrates provenance tracking and reproducibility
- Shows measurable improvement in data quality or throughput

**Facilities Involved:**

- Primary: ALS or SLAC beamline
- Compute: NERSC for training and RAG infrastructure
- Potential expansion to other DOE light sources

This addresses multiple gaps identified: documentation, metadata standardization, agent memory, and practical control system integration.

From me: Can you give me just two of the top choices

Can you give me just two of the top choices

04:16 PM

AI Companion Said:

## **Top 2 Genesis-Relevant Demonstrations (6–18 Months)**

### **1. Autonomous Beamline Optimization Agent**

- Deploy at ALS or SLAC beamline with EPICS/Blue Sky integration
- Agent learns from past experiments using RAG-based memory
- Includes human-in-the-loop approval and automatic documentation
- Target: 50% reduction in experiment setup time

### **2. Cross-Facility Metadata and Knowledge Capture System**

- Standardized electronic lab notebook deployed at 2-3 DOE facilities
- Automatic provenance tracking of experimental workflows
- ML-ready data with consistent metadata standards
- Creates searchable institutional knowledge base from scattered notebooks and tribal knowledge