

# Instrumentation vs. Observability: Provenance Capture Trade-offs in Agentic Workflows

**Valesca Moura (Presenter)**<sup>1</sup>, Amal Gueroudji<sup>2</sup>, Daniel Rosendo<sup>3</sup>, Ben Clifford<sup>2</sup>, Kyle Chard<sup>2</sup>, Rafael Ferreira da Silva<sup>3</sup>, Daniel de Oliveira<sup>1</sup>, Renan Souza<sup>3</sup>

<sup>1</sup> Universidade Federal Fluminense, Niterói, RJ, Brazil

<sup>2</sup> Argonne National Laboratory, Lemont, IL, USA

<sup>3</sup> Oak Ridge National Laboratory, Oak Ridge, TN, USA

---

**APeX 2026 — Academy, Parsl, and Globus Compute Summit**

September 14–15, 2026



# Adding Provenance to Agentic Workflows



*The question mark: which integration point should carry provenance capture?*

**Initial motivation: add provenance to Academy-based scientific workflows.**

## What can we capture?

- Workflow information
- Agent interactions
- Domain information
- Telemetry and resource usage

## How easy is it to adopt?

- Does the application need to change?
- Does the framework need to change?
- Is additional code or configuration required?
- How tightly coupled are the systems?

**Research question: How should provenance capture be integrated into agentic frameworks?**

# Flowcept: Provenance Capture and Analysis

## Distributed provenance capture

Captures execution data across distributed, HPC, and agentic workflows.

## Unified provenance storage

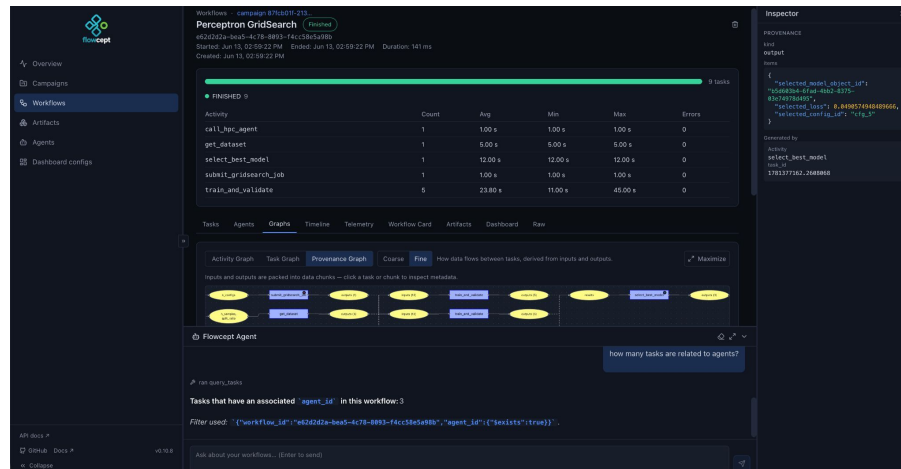
One schema and repository for agentic and non-agentic workflows.

## Provenance analysis and queries

UI visualizations, graphs, and a query/chat agent over captured data.

## Flexible integration with distributed workflows

Low-overhead capture designed for HPC and multi-agent



Flowcept UI: workflow, provenance graph, and chat-based agent

Flowcept — open-source provenance framework developed at ORNL <https://github.com/ORNL/flowcept>



# Six Provenance Capture Strategies

*Generic capture strategies evaluated using Academy + Flowcept*

## Direct Instrumentation

### 1. Direct Application Instrumentation

Instrumentation embedded in user application code

### 2. Direct Framework Instrumentation

Instrumentation embedded in the agentic framework

### 3. Dynamic Framework Instrumentation

Monkey patch-based agentic framework adapter

## Observability Adapters

### 4. Log Handler-based Observability

Log observer customized to the agentic framework logs

### 5. Message Stream Observability

Data streaming observer customized to agentic message streams

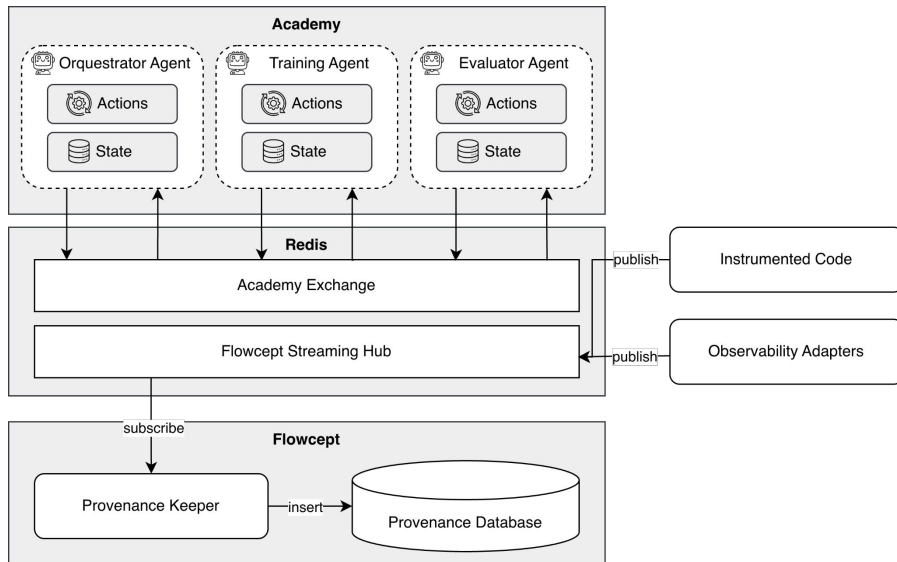
### 6. Third-party Adapters

e.g., OpenTelemetry-specific observability adapters

**What can we capture? × What does it take to adopt it?**

*Detailed characterization: “Instrumentation vs. Observability: Provenance Capture Trade-offs in Agentic Workflows” — WISDOM @ IEEE eScience 2026 (to appear)*

# Academy + Flowcept: Implementation and Evaluation



## Evaluation

- Distributed Perceptron Grid Search workflow
- Same workflow executed with all six capture strategies
- Common evaluation criteria across strategies

Code and experiments [github.com/valescamoura/flowcept-academy-integration-analysis](https://github.com/valescamoura/flowcept-academy-integration-analysis)



# No Single Best Capture Strategy

| Approach                                  | Captured Information                                                | Gaps                                                | Intrusive-ness | Adoption Effort                                      |
|-------------------------------------------|---------------------------------------------------------------------|-----------------------------------------------------|----------------|------------------------------------------------------|
| <b>Direct Application Instrumentation</b> | Domain info, telemetry, workflow provenance                         | Depends on app developer; limited A2A detail        | <b>High</b>    | Requires application code changes (decorators/hooks) |
| <b>Direct Framework Instrumentation</b>   | Workflow provenance, telemetry, agentic prov., possibly domain info | Hard to capture domain-specific semantics           | <b>High</b>    | Requires framework source code changes               |
| <b>Dynamic Framework Instrumentation</b>  | Workflow provenance, telemetry, agentic prov., possibly domain info | Hard to capture domain-specific semantics           | <b>Medium</b>  | No source changes — runtime monkey-patch adapter     |
| <b>Log Handler-based Observability</b>    | Workflow provenance, telemetry (if sent), possibly domain info      | Depends on log richness; limited A2A detail         | <b>Low</b>     | No app changes — attach a logging handler            |
| <b>Message Stream Observability</b>       | Agent-to-agent (A2A) communication details                          | Hard to capture domain semantics; limited telemetry | <b>Low</b>     | No app changes — subscribe to the message bus        |
| <b>Third-party Adapters</b>               | Workflow provenance, telemetry, agentic prov., possibly domain info | Hard to capture domain-specific semantics           | <b>Low</b>     | No app changes — configure an OTel exporter          |

What do I capture? × What do I need to change?

# What Can Each Strategy Capture?

*Coverage of unique mapped fields by analytical capability*

| Approach                           | agentic | dataflow-provenance | execution-traces | platform & repository | resource-usage | timing |
|------------------------------------|---------|---------------------|------------------|-----------------------|----------------|--------|
| Direct Application Instrumentation | 77.8%   | 100%                | 100%             | 100%                  | 100%           | 100%   |
| Direct Framework Instrumentation   | 88.9%   | 100%                | 100%             | 100%                  | 100%           | 100%   |
| Dynamic Framework Instrumentation  | 100%    | 100%                | 100%             | 100%                  | 100%           | 100%   |
| Log Handler-based Observability    | 66.7%   | 100%                | 100%             | 100%                  | 100%           | 100%   |
| Message Stream Observability       | 100%    | 100%                | 100%             | 100%                  | 16.7%          | 100%   |
| Third-party Adapters               | 55.6%   | 100%                | 100%             | 42.9%                 | 33.3%          | 100%   |

Complementary strategies can be combined for broader provenance coverage.

# Takeaways & Resources

---

- No single best capture strategy
- The choice depends on both provenance needs and adoption effort
- Complementary strategies can be combined

Choose based on what you need to capture and what you can change.

## Interested in adding provenance to your Academy application?

Flowcept [github.com/ORNL/flowcept](https://github.com/ORNL/flowcept)

Academy + Flowcept integration study [github.com/valescamoura/flowcept-academy-integration-analysis](https://github.com/valescamoura/flowcept-academy-integration-analysis)\*

*\*User-friendly integration guide and demos coming soon.*



# Thank You!

---

## Valesca Moura

[valescamsousa@gmail.com](mailto:valescamsousa@gmail.com) | [valescamoura@id.uff.br](mailto:valescamoura@id.uff.br) | [valescamoura.github.io](https://valescamoura.github.io)

**Acknowledgements** Amal Gueroudji · Ben Clifford · Daniel Rosendo

---

## Interested in adding provenance to your Academy application?

Flowcept [github.com/ORNL/flowcept](https://github.com/ORNL/flowcept)

Academy + Flowcept integration study

[github.com/valescamoura/flowcept-academy-integration-analysis](https://github.com/valescamoura/flowcept-academy-integration-analysis)\*

*\*User-friendly integration guide and demos coming soon.*

---



Paper “Instrumentation vs. Observability: Provenance Capture Trade-offs in Agentic Workflows”

WISDOM @ IEEE eScience 2026 — to appear