

# Accelerators

## 5th ICFA Beam Dynamics Mini-Workshop on Machine Learning for Particle Accelerators

**Raimund Kammering (DESY),**  
Hayden Houscher, Jason St. John (Fermilab)  
Thorsten Hellert, Antonin Sulc (LBNL)



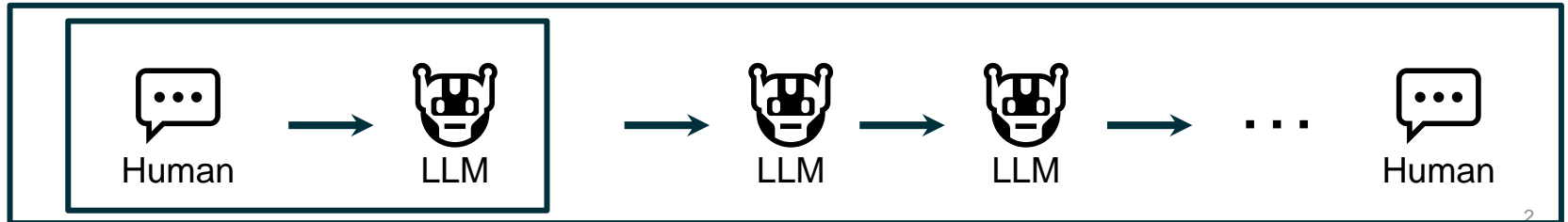
**BERKELEY LAB**

Bringing Science Solutions to the World



# What are Agents

- **What They Are:** AI helpers that don't just respond (LLMs) but actively work to solve your problems
- **How They Work:**
  - *Autonomy* - agents are able to make decisions
  - *State* - they often have memory
  - Use other *tools* and services when needed
  - *Communicate* with other agents to solve problems
- **Why They Matter:** Moving from "AI that answers questions" to "AI that accomplishes goals"



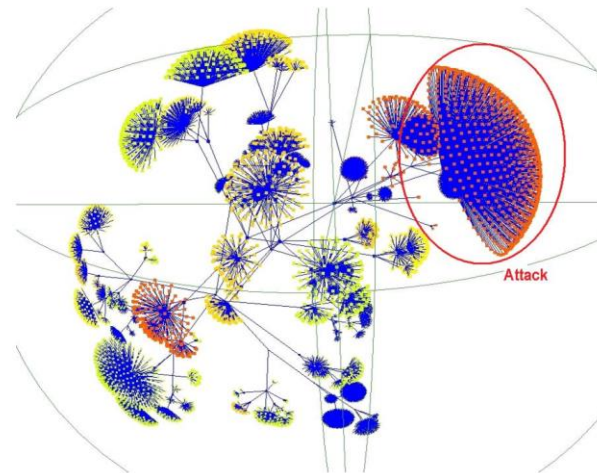
# Agents - What is it about?

- There is almost unlimited list of possible agent combinations
- Multi-agent systems are well known for decades

AgentFly (<https://www.agentfly.com/>)



Network Intrusion Detection



**But we never had such tools available**  
(especially autonomy!)

# Agents - What is it about?

## Multiagent frameworks: A decades-old concept with new capabilities

- Multiagent systems have existed since the 1980s in distributed AI research
- Traditional limitations:
  - Rigid communication protocols
  - Limited reasoning capabilities
  - Constrained to narrow, predefined tasks

## What's genuinely new:

- Large language models enable natural language communication
- Emergent capabilities for complex problem-solving
- Dynamic task decomposition and coordination
- Agents that can learn, adapt, and negotiate in real-time

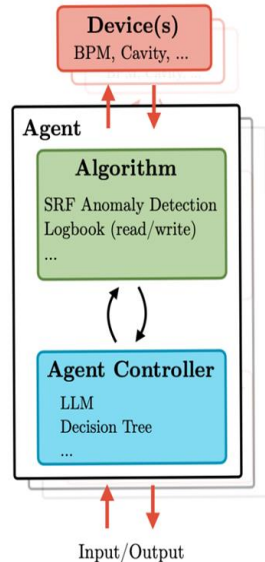
**Autonomy**

**Communication**

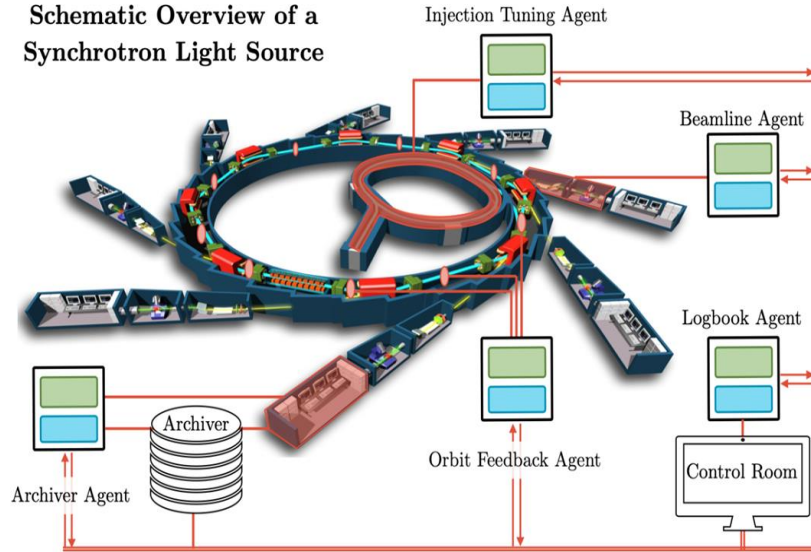
**Distributed**

# Autonomous Accelerators

- Are we getting somewhere close to it?
- There is an impressive work in subparts of accelerators
- How to put these subparts together and run together?



Schematic Overview of a Synchrotron Light Source



# State-of-the-art of agent(ic) frameworks

- **Agent Frameworks**

- Agents to collaborate autonomously on complex problems  
(*AutoGen: framework for building AI agent*)
- Multi-agent debate approaches improve reasoning and factual accuracy for reliable operations

- **Planning & Control**

- Planning complex tasks that require coordination of multiple systems  
(*LLM+P: LLM with planning*)
- Verbal reinforcement learning without traditional weight updates  
(*Reflexion: e.g. self-reflexion, chain-of-thought*)

- **Code Generation**

- LLMs demonstrate capabilities as coding assistants, problem solvers, and data scientists

# Examples of Agents

So we can engineer autonomous agents for instance for

- **Anomaly detection** Uses some standard off-the-shelf anomaly detection algorithms like isolation forests to monitor certain subparts of control system
- **Archiver Agent** The agent that knows how the machine behaved in the past and can quickly retrieve relevant data from the system
- **Coding assistant agent** The agent can put together multi-lingual repositories and can write code or assist creating one
- **Machine state agent** Allows us to query the control system about the current state
- **Logbook agent** Can provide information about the past and current state of the machine and can for instance reason about it

## Example - Orbit Feedback Agent

Imagine following scenario:

An **orbit feedback (FB) agent** observes orbit anomalies

1. FB agent checks the typical orbit by prompting **Archiver Agent**

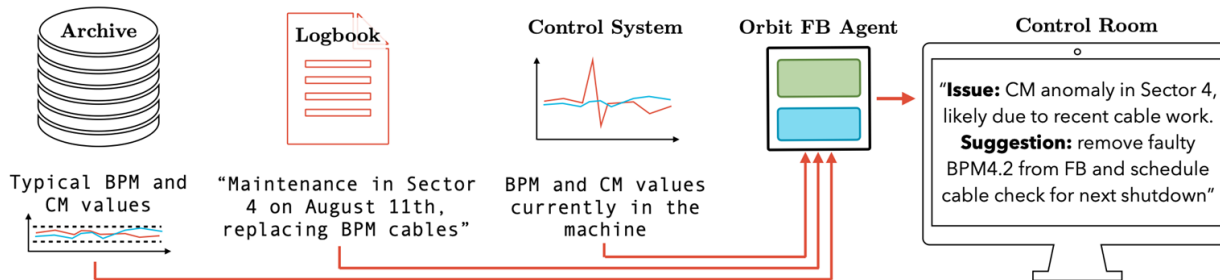
*“Give me typical orbit of BPM4.2”*

2. FB agent checks if there were any known issues on the BPM from **Logbook Agent**

*“Any known issue about BPM4.2”*

3. Control Room Agent gets an update:

*“Anomaly due to cable work that took place on August 11”*



# System Prompt - Key to Control LLMs

You are useful chatbot.

Capital of France is ...

Paris

```
You are a chatbot that
reasons about user
questions
<reasoning>
reasoning about the question
</reasoning>
<answer>
place for answer
</answer>
```

What is capital of France

```
<reasoning>
list of largest French cities is Paris,
Marseille, Lyon by population and
usually it is the most populated city, so
it must be France
</reasoning>
<answer>Paris</answer>
```

## How is it done then?

- You have your knowledge, e.g. about specific orbit behaviour
- If you have e.g. model of requirement
  - e.g. FODO lattice,
  - anomaly detection routines on specific component,
  - search the logbook
- Example:
  - You tell an LLM how to use the model in system prompt  
You are an anomaly detection agent, you continuously monitor BPMs and in case of anomaly, you trigger an alarm and inform OperatorAgent and eLogAgent
  - You develop
    - a separate thread that can communicate with others
    - if anomaly is detected (manually defined by e.g. isolation forests) and decides to perform specific actions

# Conclusion

- You have seen some potential use cases of an multi-agent AI architecture for collaborative particle accelerator control
  - Orbit agent @ ALS
  - Coding agent @ Fermilab
- Agents can observe, recommend actions, and eventually automate operations
- You can come up with large **variety of agents** that can **communicate**
- Represents a massive potential shift in methodology and deployment of known algorithms due to its **modularity**
- There are inherent **risks** (e.g. *alignment*) associated with it's autonomy

# Thank You

And sure we are open for collaborations!

Hayden R. Hoschouer [haydenh@fnal.gov](mailto:haydenh@fnal.gov),

Jason Michael St.John [stjohn@fnal.gov](mailto:stjohn@fnal.gov)

Antonin Sulc [asulc@lbl.gov](mailto:asulc@lbl.gov)

Kammering Raimund [raimund.kammering@desy.de](mailto:raimund.kammering@desy.de)