

Lightning Agents

Factory-of-Factories for Dynamic AI Agents

Austin AI MUG Lightning Talk

The Problem

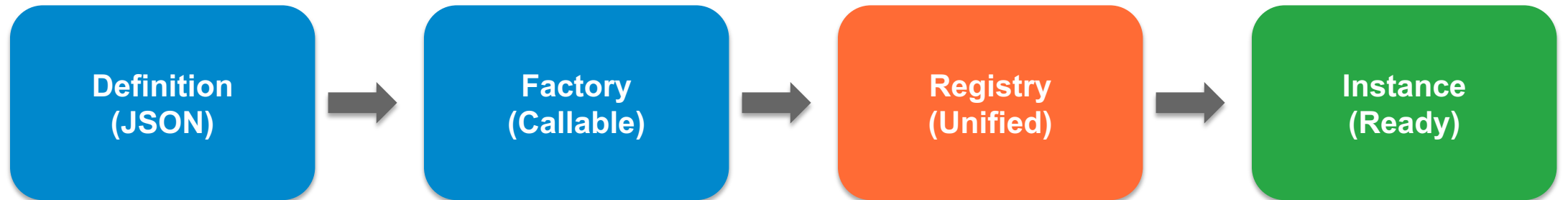
Hardcoded system prompts scattered across files

Tightly coupled agent definitions and execution logic

No standardized way to add agents **without code changes**

Manual **configuration duplication** when agents share patterns

The Solution: Factory-of-Factories



Simple Factory vs Factory-of-Factories

Simple Factory

```
def create_agent(config):  
    return Agent(config)
```

Factory-of-Factories

```
registry = AgentRegistry.from_json(  
    "agents.json"  
)  
agent = registry.create(  
    "researcher", {"topic": "AI"}  
)
```

Declarative Agent Definition

```
{  
  "architect": {  
    "name": "Agent Architect",  
    "description": "Designs new agents",  
    "system_prompt": "You are an Agent Architect...",  
    "model": "sonnet",  
    "tools": []  
  }  
}
```

The Registry Pattern

```
class AgentRegistry:
    def from_json(path) -> "AgentRegistry":
        # Load definitions -> build factories

    def create(id, opts) -> AgentInstance:
        # Factory creates configured instance

    def register(id, defn) -> None:
        # Add new agent to registry
```

The Voyager Insight

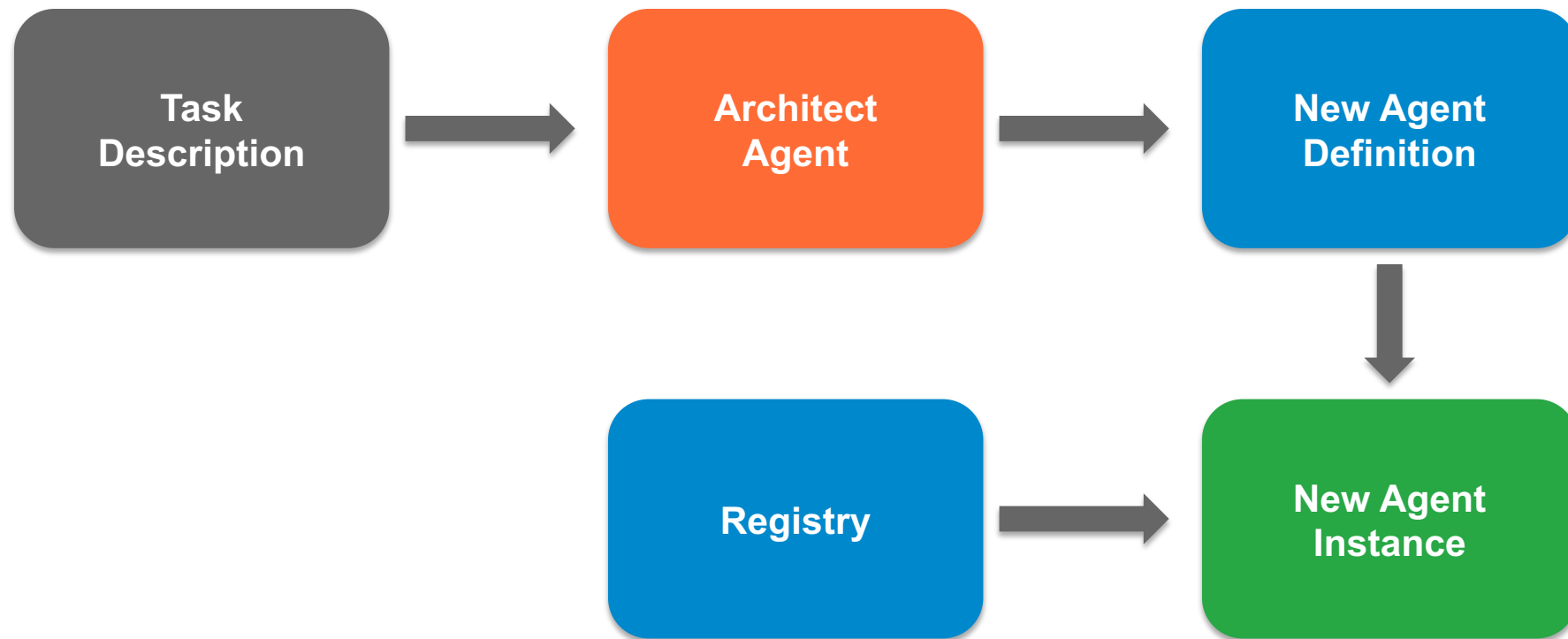
Voyager: Minecraft AI with a growing **skill library**

Learns new skills → **stores them** → reuses them

Lightning Agents: same pattern for AI agents

architect creates → **Registry** stores → **CLI** runs

The Architect Agent



But Wait... Tool Architect Too

Agents can also design **NEW TOOLS**

`tool_architect` registers definitions in `db/tools.json`

`paper_researcher` searches papers **AND** downloads PDFs

`presentation_slide_writer` manipulates slides **directly**

Demo: What We Built

```
$ lightning list  
10 agents: architect, tool_architect, paper_researcher...  
  
$ lightning run paper_researcher "Find the Voyager paper"  
Found: arxiv.org/abs/2305.16291  
Downloaded: voyager_lifelong_learning.pdf  
  
$ lightning run presentation_slide_writer "List slides"  
12 slides in current presentation...
```

The Meta Moment

We used `paper_researcher` to find the **Voyager paper**

Voyager inspired the **architect pattern**

We built `tool_architect` to create **more tools**

`presentation_slide_writer` is building **THIS presentation**

Lightning Agents

Agents creating **agents** creating **tools**

Built with **Claude Agent SDK + MCP**

github.com/aimug-org/lightning-agents