

Lightning Agents

Factory-of-Factories for Dynamic AI Agents

Austin AI MUG Lightning Talk

The Problem

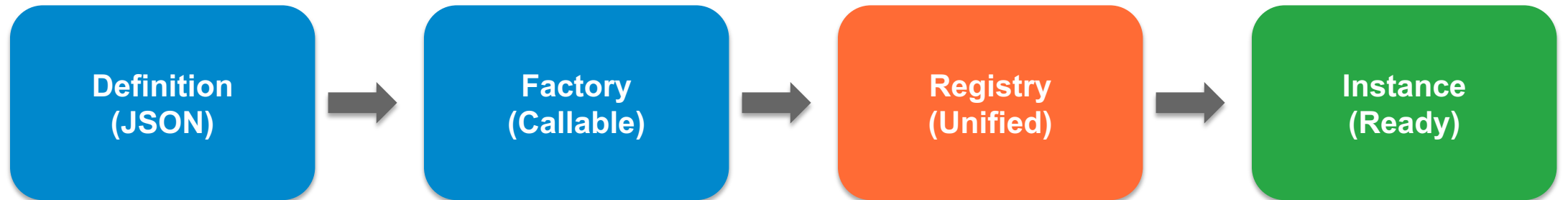
Hardcoded system prompts scattered across files

Tightly coupled agent definitions and execution logic

No standardized way to add new 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
```

Runtime Context Injection

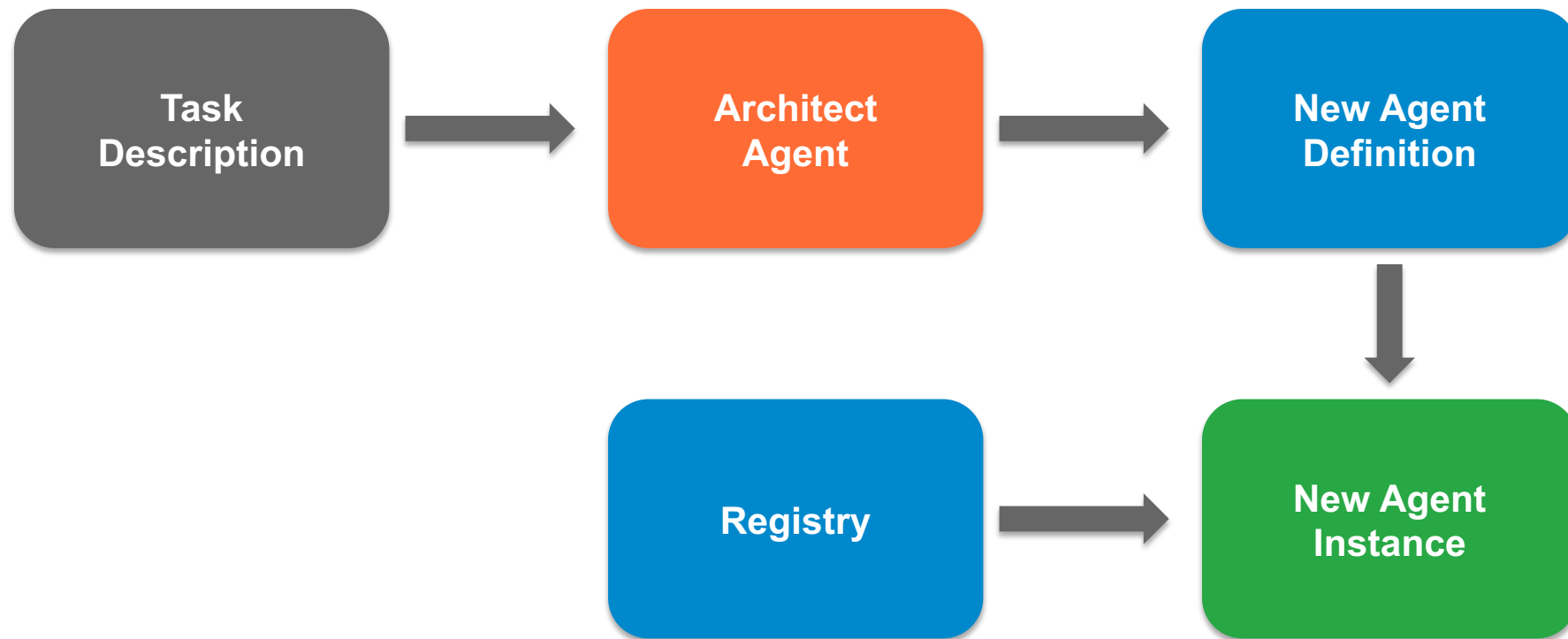
Same definition, different contexts

Inject parameters at instantiation time

agent.prompt merges base + runtime context

Enables agent reusability across use cases

The Architect Agent



Demo Flow

lightning list → Show available agents

lightning run lab_finder "RAG tutorials" → Use existing agent

lightning architect "meeting summarizer" → Create NEW agent

lightning list → New agent now available!

lightning run meeting_summarizer "..." → Use it immediately

Key Hypotheses

H1: Declarative > Imperative for agent configuration

H2: Runtime context injection enables reusability

H3: Architect agents enable organic system growth

Live Demo

```
$ lightning list  
  architect, aimug_researcher, lab_finder...  
  
$ lightning architect "PR reviewer for security"  
  
Created agent: security_reviewer  
Saved to: agents.json  
  
$ lightning run security_reviewer "Review auth.py"
```

Lightning Agents

[github.com/\[your-repo\]/lightning-agents](https://github.com/[your-repo]/lightning-agents)

Built with Claude Agent SDK

Questions?