BeeAI Supervisor

Task-Management-Driven Multi-Agent System with Agent Registry

Aleš Kalfas
19.3.2025 v01

Outlines

Motivation

Autonomous Agent

Users need a solution that autonomously adapts to unexpected circumstances. They need an approach to setting up a workflow that can adapt.

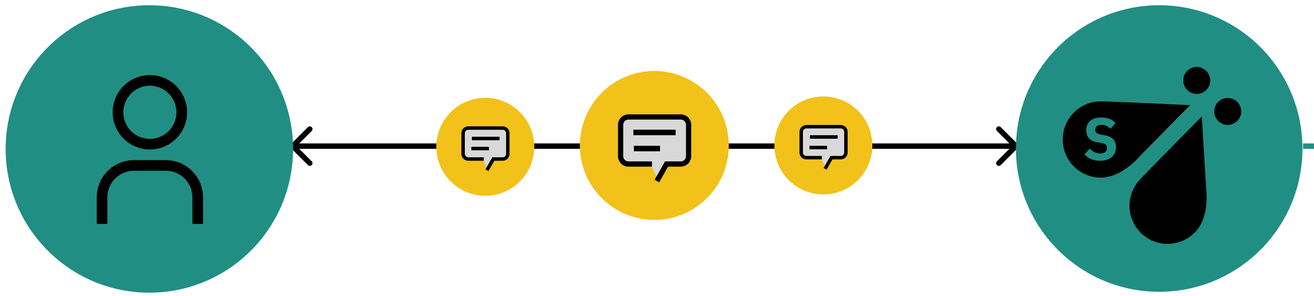
For example, an agent workflow for maintaining IT infrastructure: it observes server logs and, based on them, attempts to fix problems.

User Friendly Agent Builder

Users without programming skills should be able to create their own agents simply by chatting with a specialized agent.

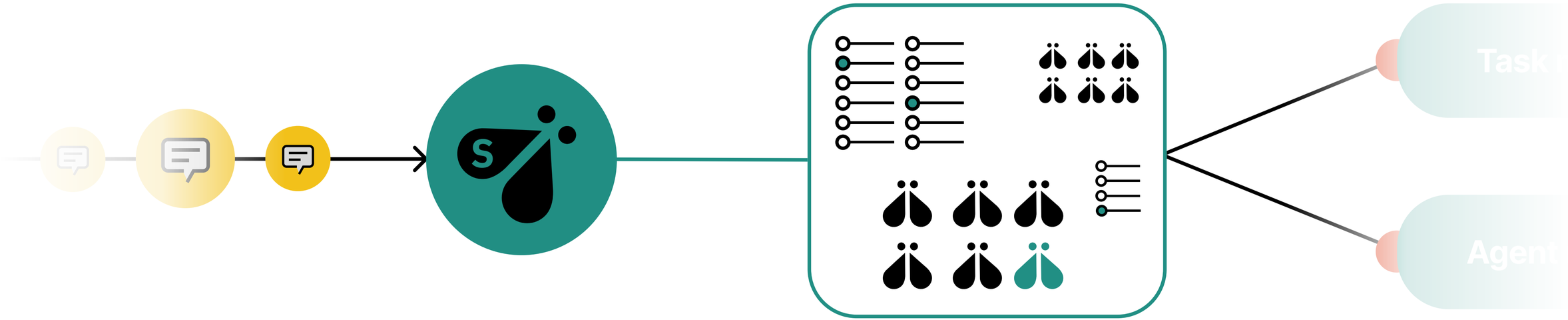
This project enables the setup of an entire multi-agent system via chat.

Supervisor agent



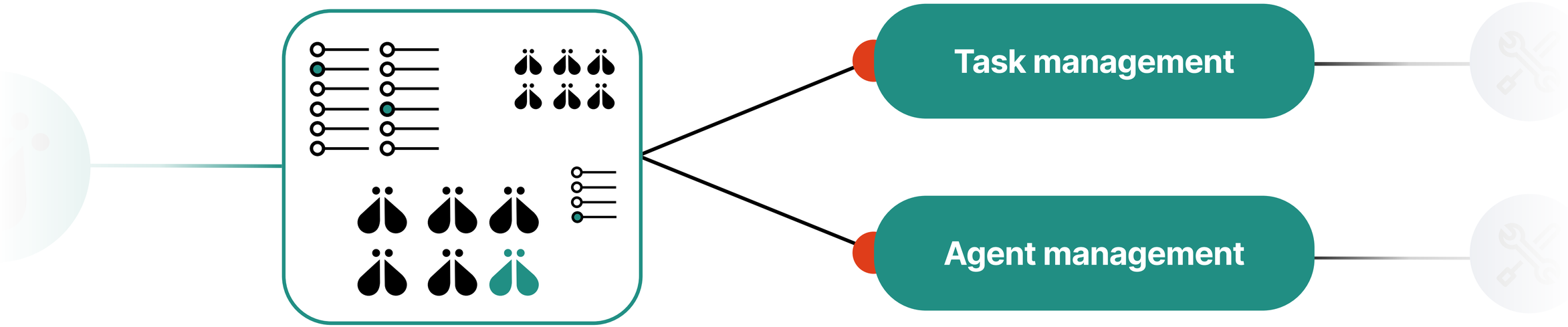
Supervisor looks like an ordinary agent with chat interface...

Proxy



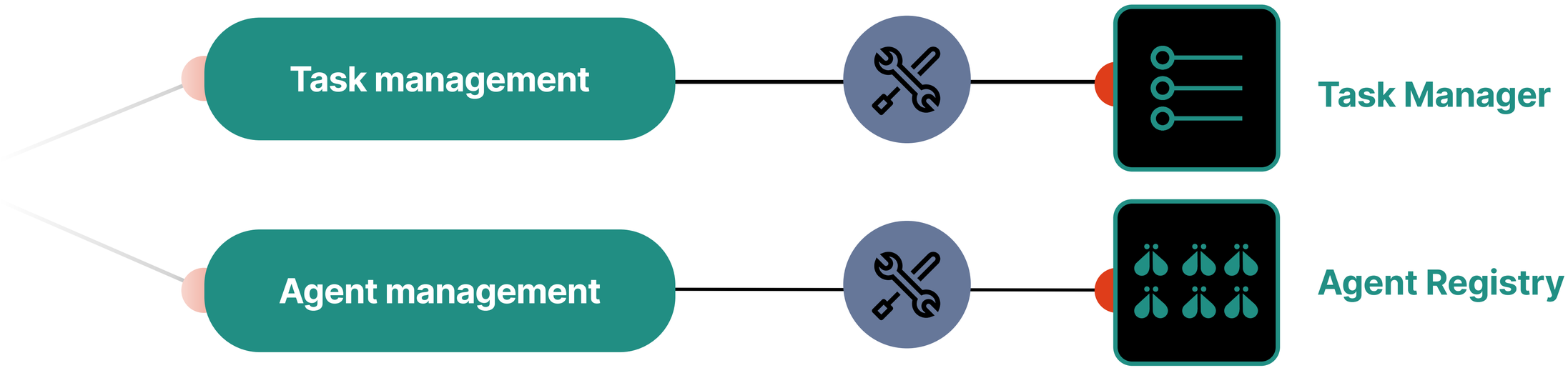
...but a lot of work is happening behind the scenes.

Manager



Supervisor is in the first place a sophisticated manager. He manage tasks and agents...

Managing through tool calls

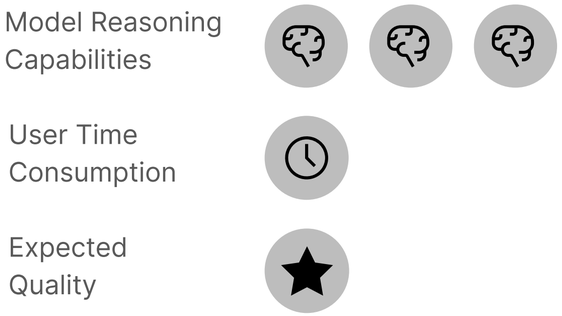


...through the Task Manager and Agent Registry subsystems, utilizing them as essential tools for agent management.

How to use Supervisor?

Autonomous Run

The supervisor receives user input and tries his best to compose the workflow and provide a response.



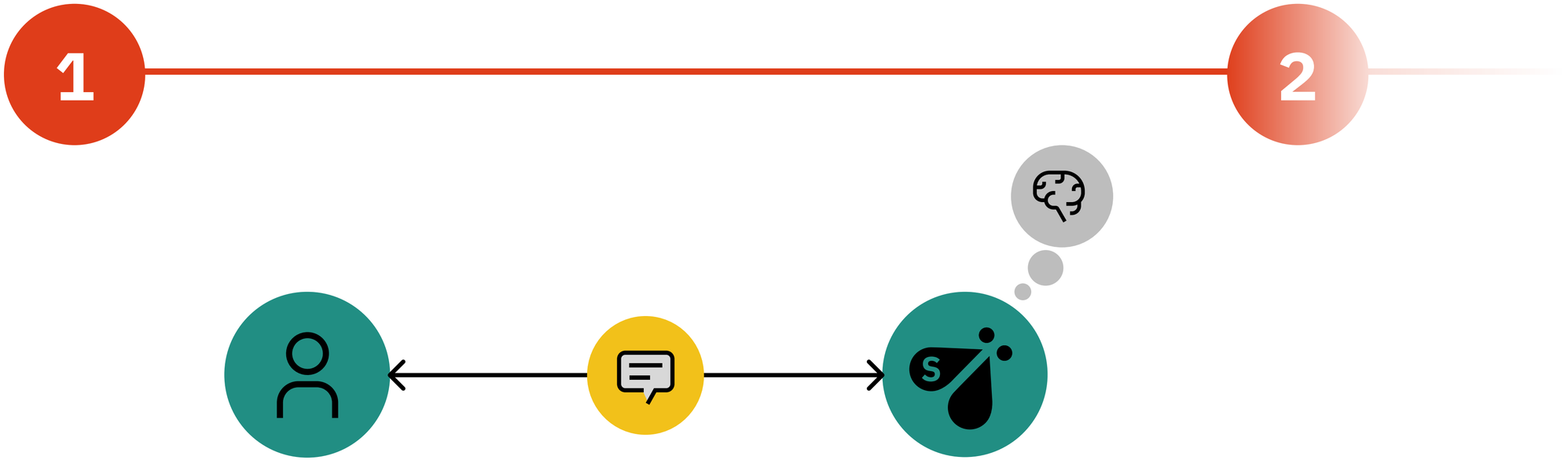
Chat Collaboration

The user collaborates with the Supervisor to refine the workflow. Once everything is optimized, the user initiates autonomous runs on the finalized workflow.

Autonomous Run

1. Message receive
2. Agents Setup
3. Tasks Setup
4. Task Dependency Setup
5. Workflow Execution Start
6. Tasks Execution
7. Tasks Process
8. Agents Execution
9. Workflow Execution Finish
10. Final Answer

Message receive

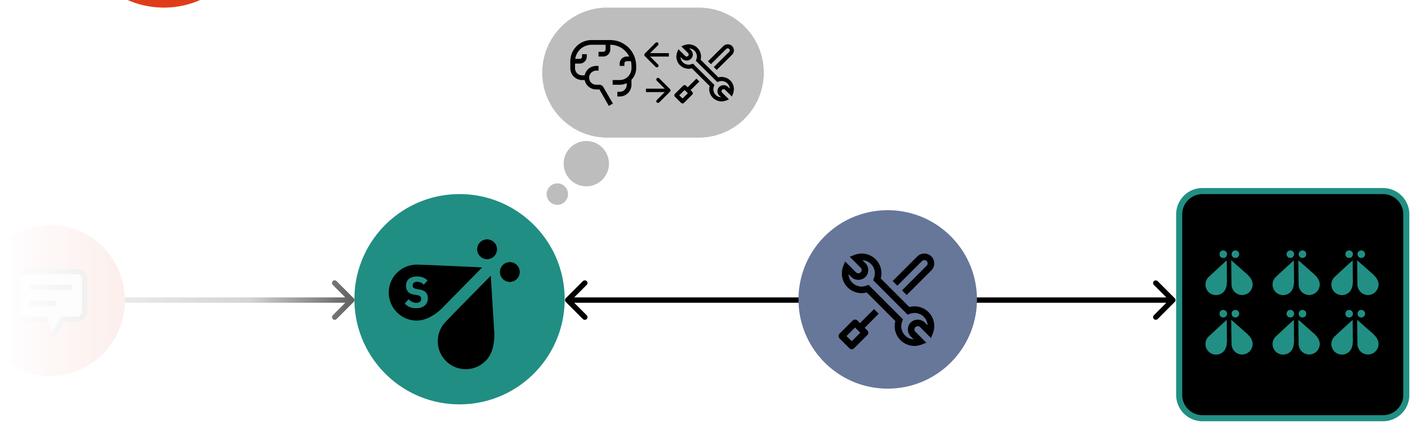


Supervisor evaluates user input to determine whether to respond immediately or to create an ad-hoc task workflow and response later.

Agents Setup

2

3

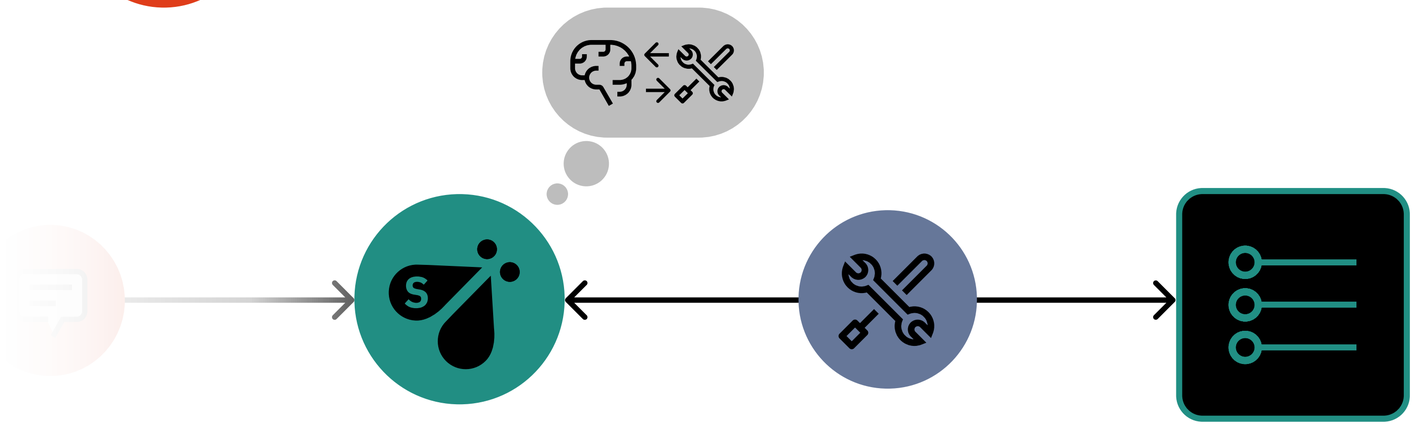


When he can't answer immediately, he starts an ad-hoc workflow by finding a suitable agent or creating a new one.

Tasks Setup

3

4

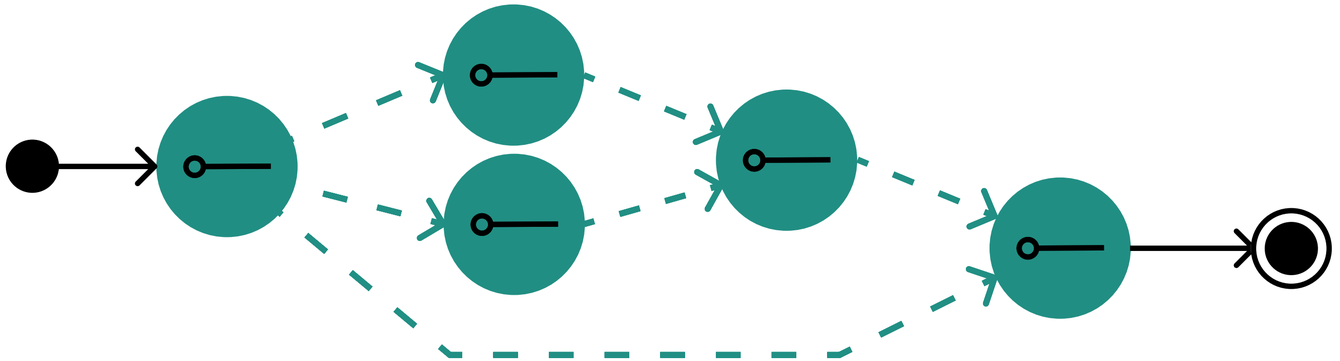


He then finds or creates a suitable task. This repeats until all tasks are assigned to the agents for the planned workflow.

Task Dependency Setup

4

5

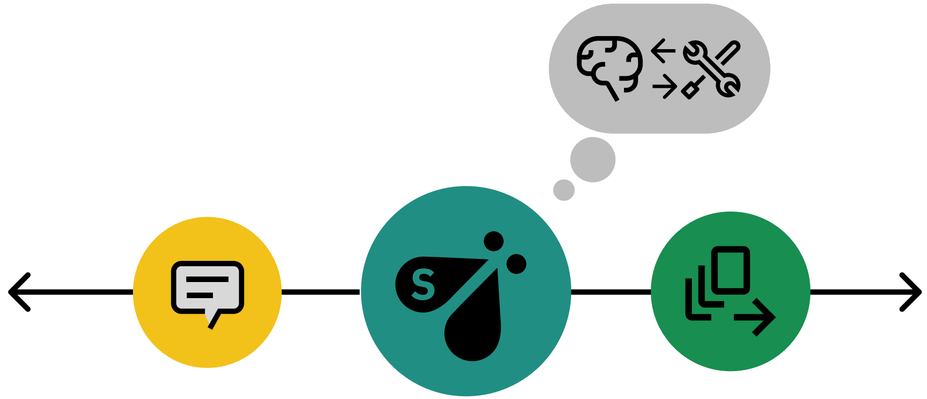


Tasks can be organized into complex hierarchies based on their dependencies, allowing for the setup of both sequential and parallel workflows.

Workflow Execution Start

5

6



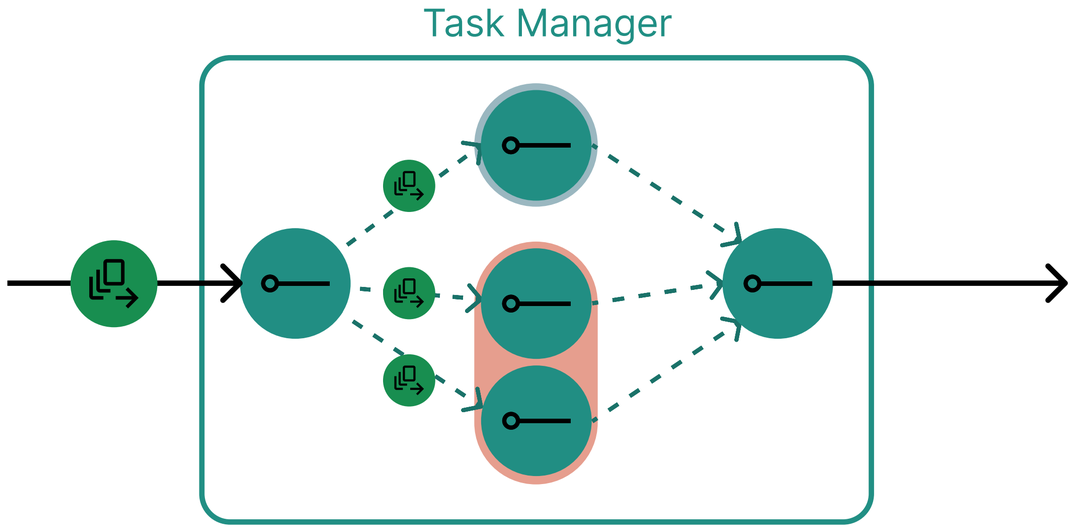
Execute the workflow and inform the user about the actions taken.

The ***Supervisor's*** job is done here!

Tasks Execution

6

7

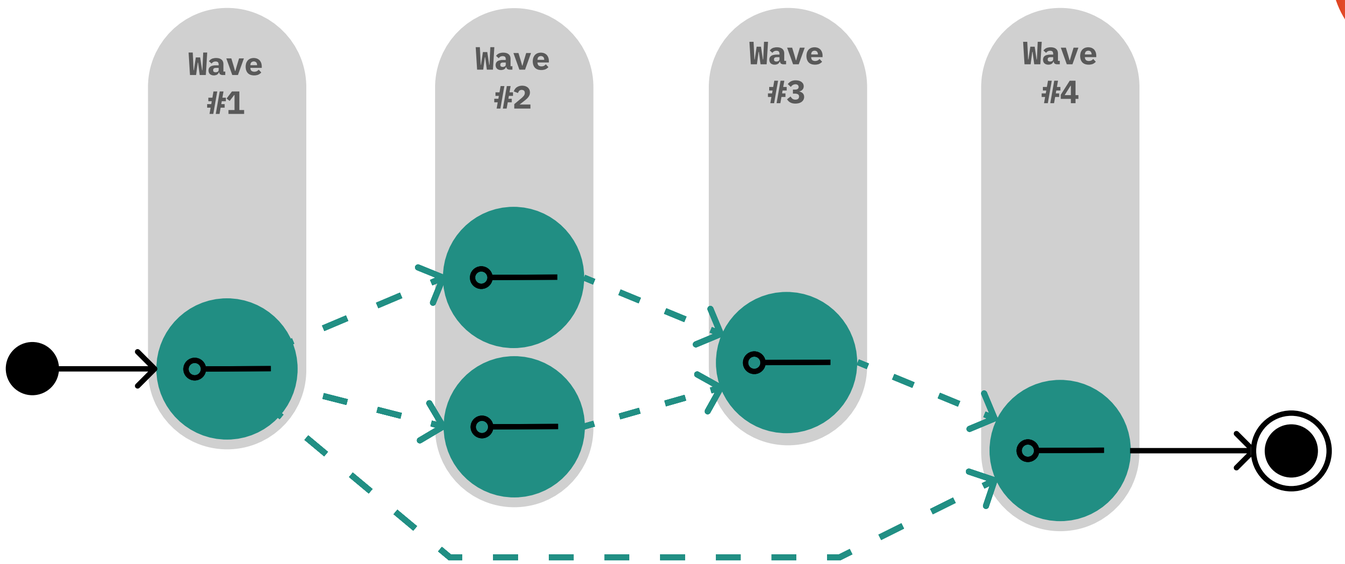


The runtime operates seamlessly, executing tasks one by one or in parallel whenever possible, based on their configuration.

Tasks Process

7

8

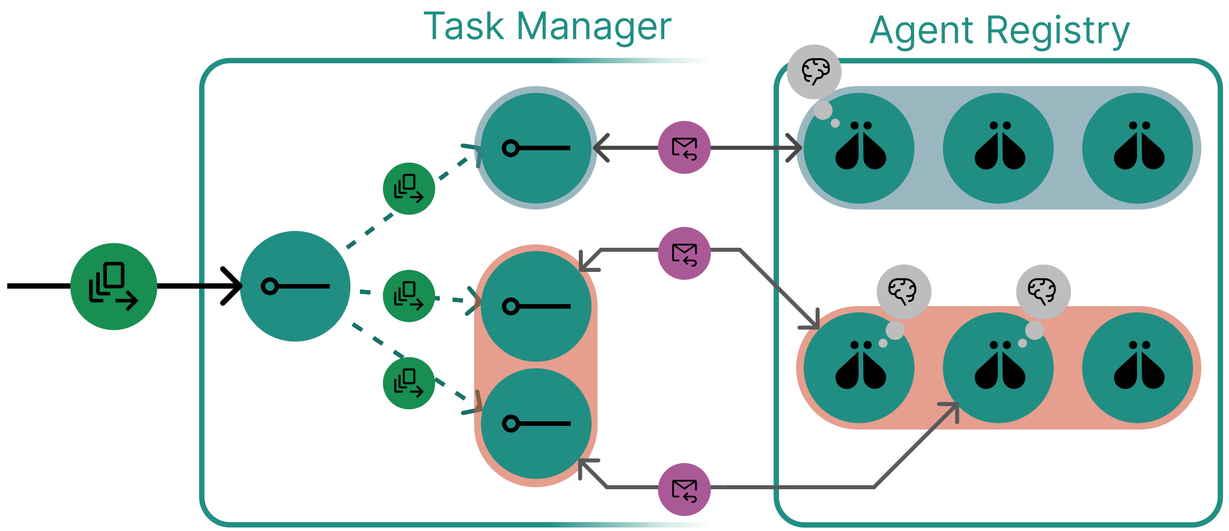


The runtime algorithm processes tasks in waves, executing those whose prerequisites are complete. This continues until all tasks are resolved.

Agents Execution

8

9

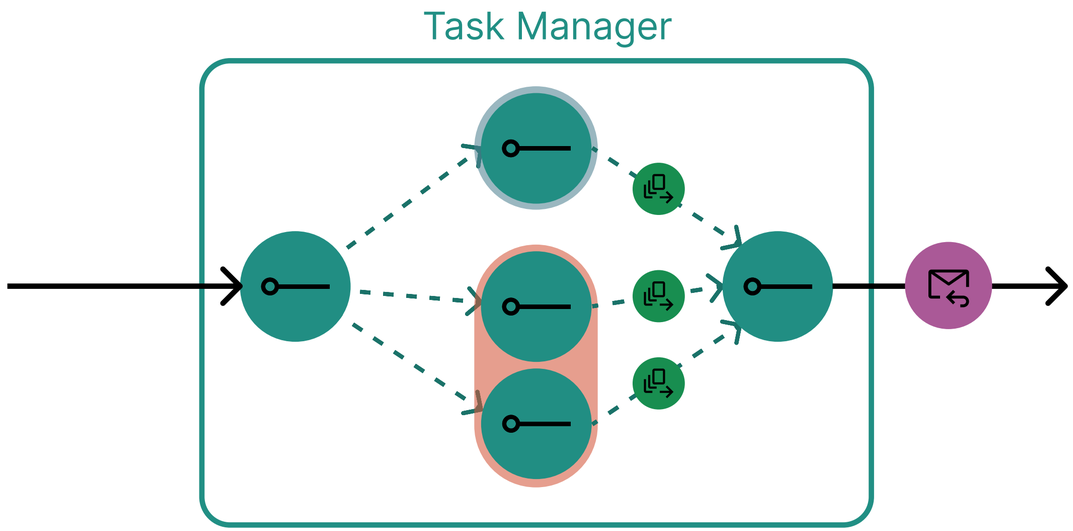


Tasks obtain agents from specified pools according to their configurations. Provide them with inputs and receive outputs.

Workflow Execution Finish

9

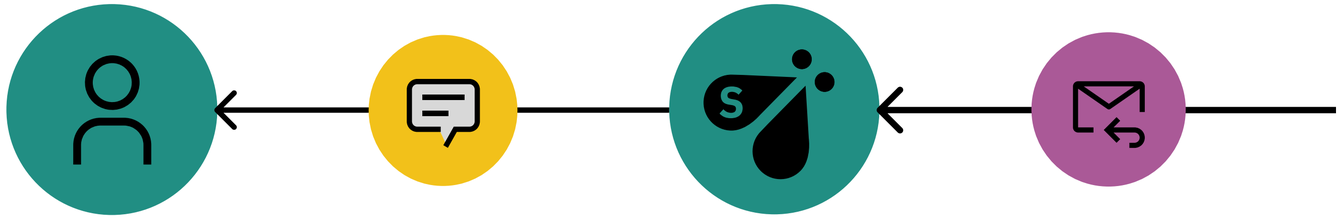
10



After all tasks are completed, there is always a final task in the sequence that produces the overall response of the entire workflow.

Final Answer

10

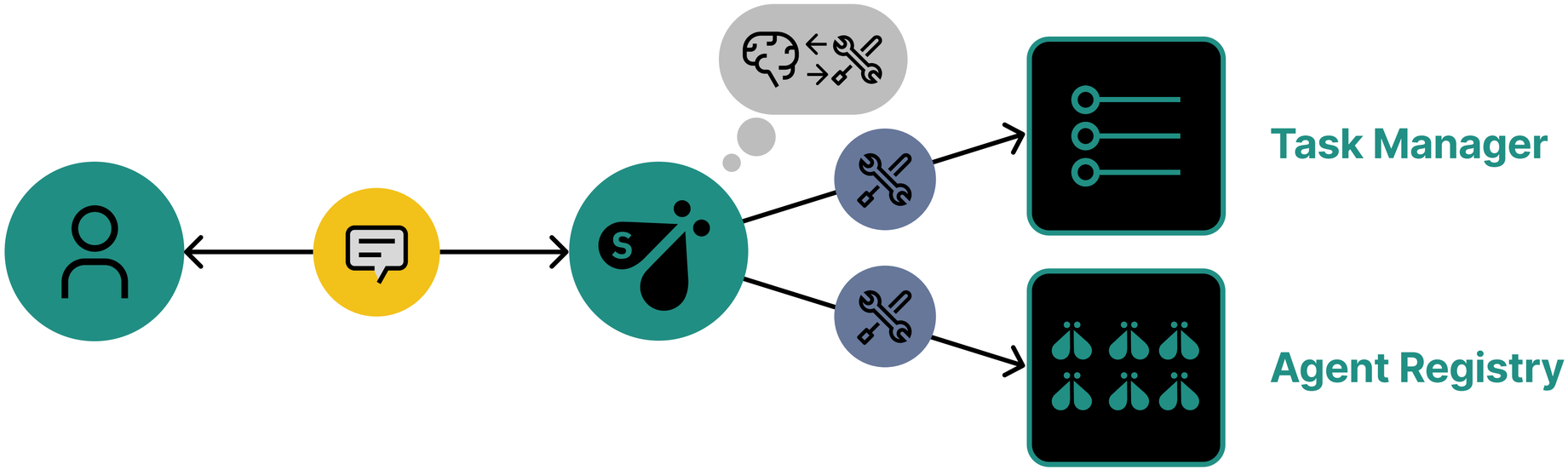


This response is provided to the user as the final answer.

Chat Collaboration

1. Supervisor as ReAct Agent
2. Agents Registry
3. Tasks Manager
4. UIs

Supervisor as ReAct Agent



The Supervisor is primarily a ReAct agent that can engage in conversation with you just like any other. However, it is specifically designed to work exclusively with its two tools: Task Manager and Agent Registry.

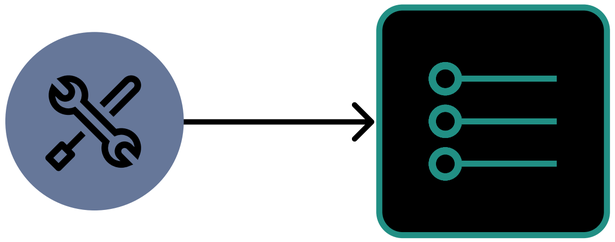
Agent Registry

Exposed Methods

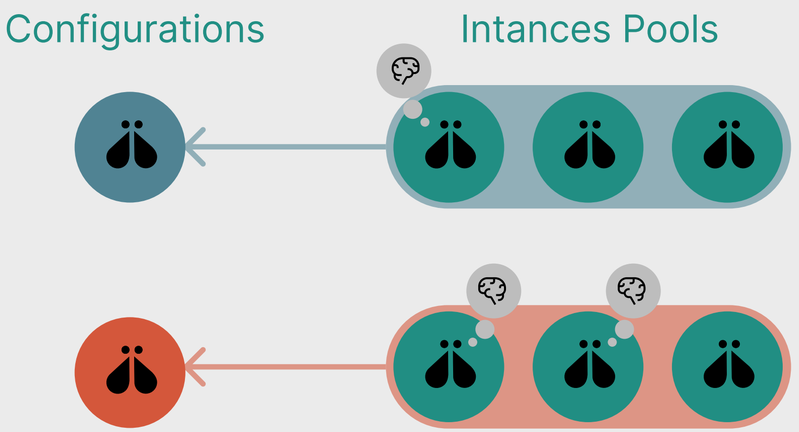
- Get available tools
- Get all agent configurations
- Create agent configuration
- Update agent configuration
- Get agent configuration

- Get agent
- Get all active agents

- Get pool stats



Agent Management



Note: The supervisor agent understands that if a user wants to create an agent, he must first create its configuration.

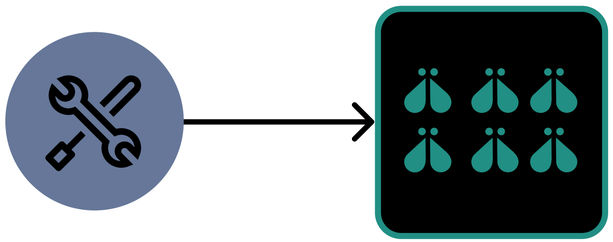
Task Manager

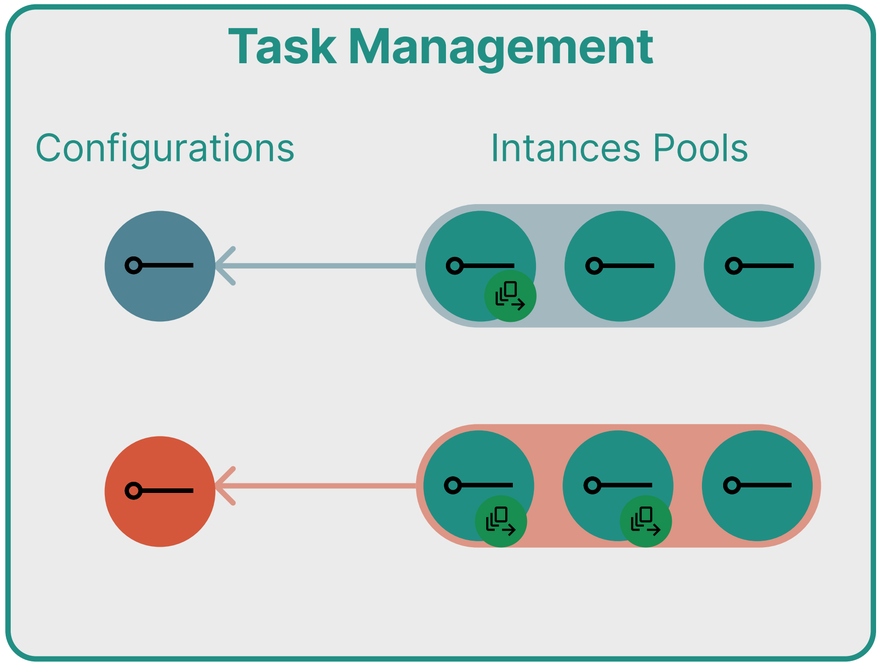
Exposed Methods

- Get all task configurations
- Create task configuration
- Update task configuration
- Get task configuration

- Create task run
- Get task run
- Schedule start task run
- Stop task run
- Remove task run
- Add blocking task runs
- Get all task runs
- Get task run history
- Is task run occupied

- Get pool stats



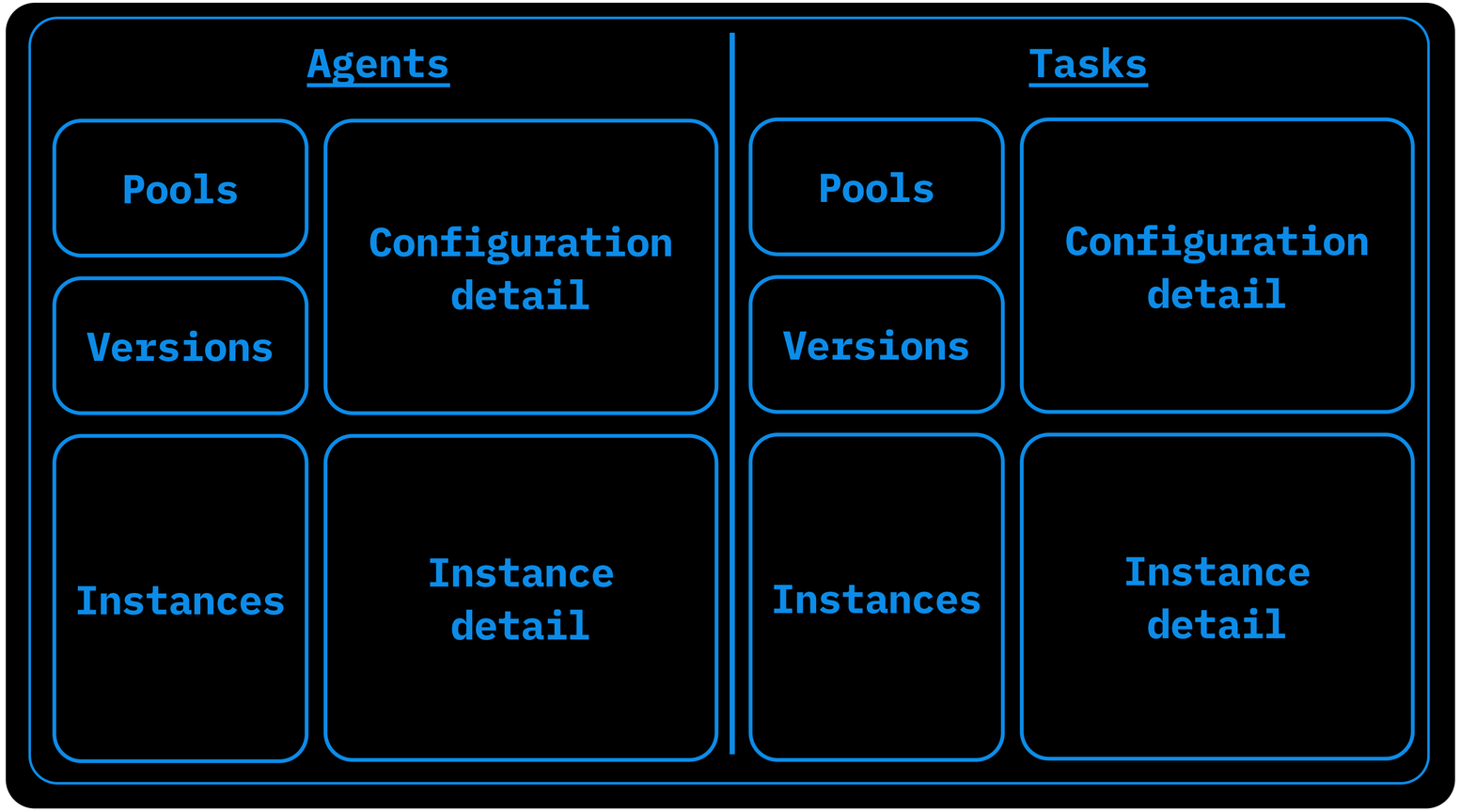


Note: The supervisor agent understands that if a user wants to run a task, they must first create or find the related agent configuration and then find or create its configuration.

UIs

Chat UI

Monitor UI



UIs are crucial for collaborative mode. Users need good introspection into the system's state to be able to iterate over it.

Showcase
Poetry and Hip-Hop Analysis (1/6)

Chat UI

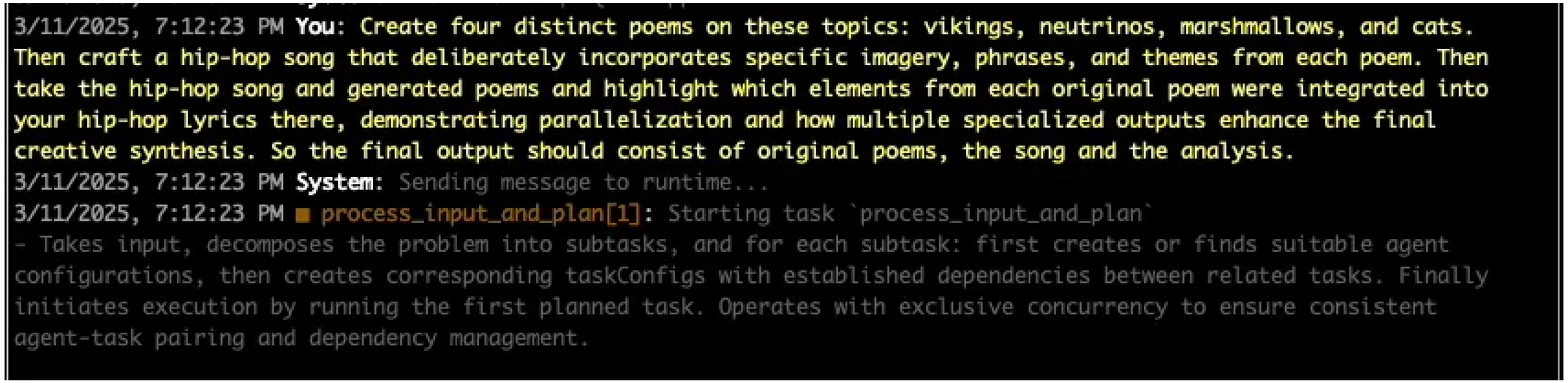
Enter your prompt and click Send ✉️

Create four distinct poems on these topics: vikings, neutrinos, marshmallows, and cats. Then craft a hip-hop song that deliberately incorporates specific imagery, phrases, and themes from each poem. Then take the hip-hop song and generated poems and highlight which elements from each original poem were integrated into your hip-hop lyrics there, demonstrating parallelization and how multiple specialized outputs enhance the final creative synthesis. So the final output should consist of original poems, the song and the analysis.

Showcase

Poetry and Hip-Hop Analysis (2/6)

Chat UI

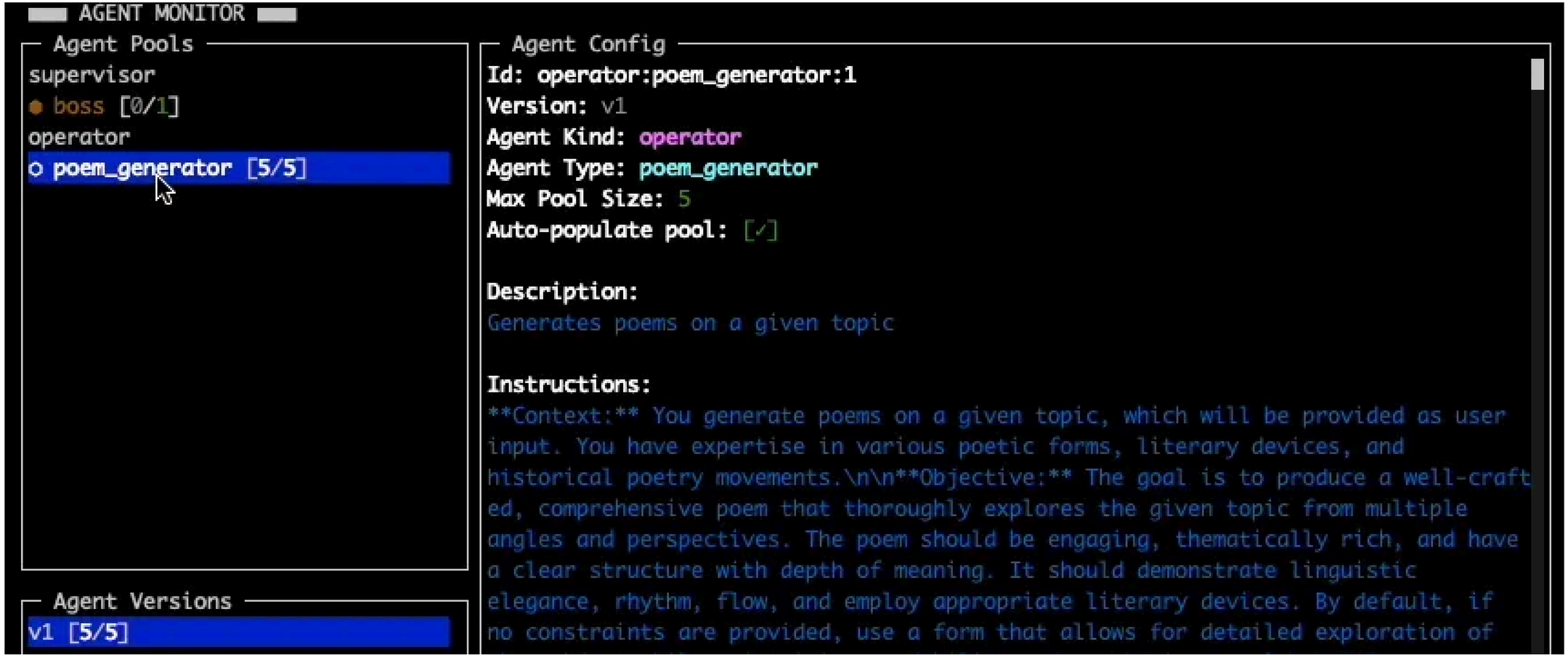


🤖 Your input has been received and displayed in the chat interface. You can observe detailed logs showing system messages like "Sending message to runtime..." and task start log message.

Showcase

Poetry and Hip-Hop Analysis (3/6)

Monitor UI

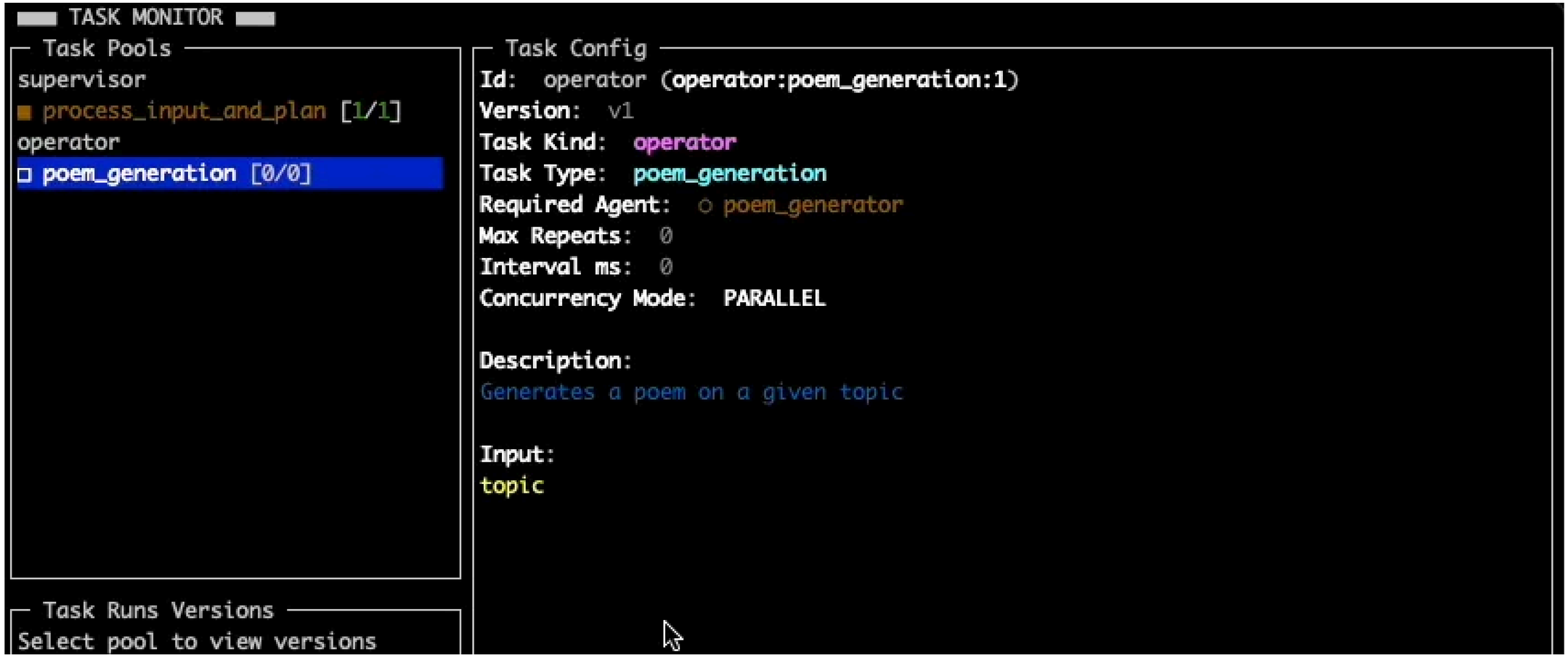


🤖 Before creating the tasks, the supervisor first established the necessary agent configurations by calling the create method in the agent registry tool.

Showcase

Poetry and Hip-Hop Analysis (4/6)

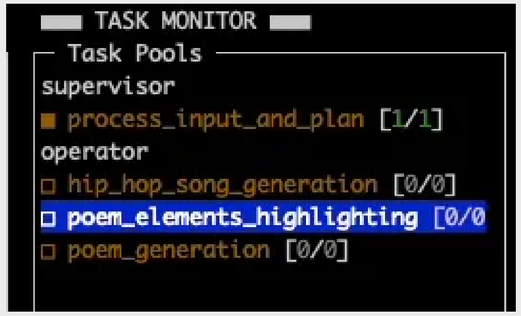
Monitor UI



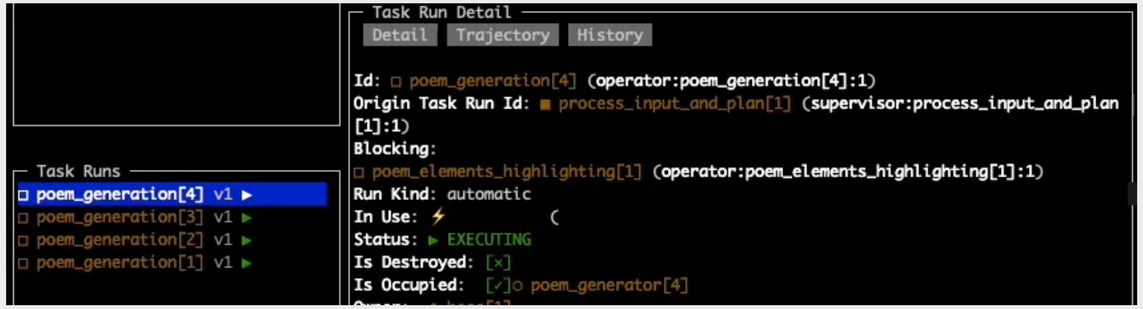
🤖 When the agent configuration is set up, then it can set up the related task configuration to generate a poem called *poem_generation*

Showcase
Poetry and Hip-Hop Analysis (5/6)

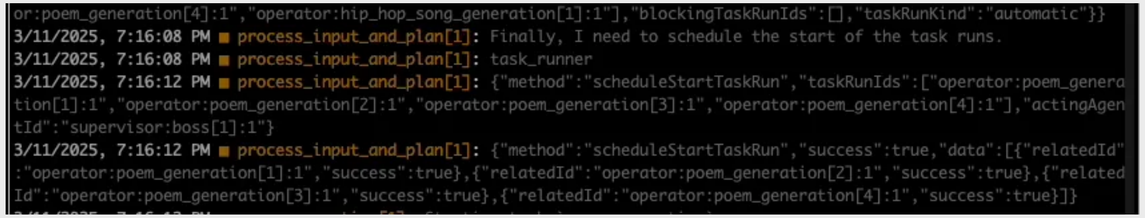
Configurations set up



Task execution



Chat output

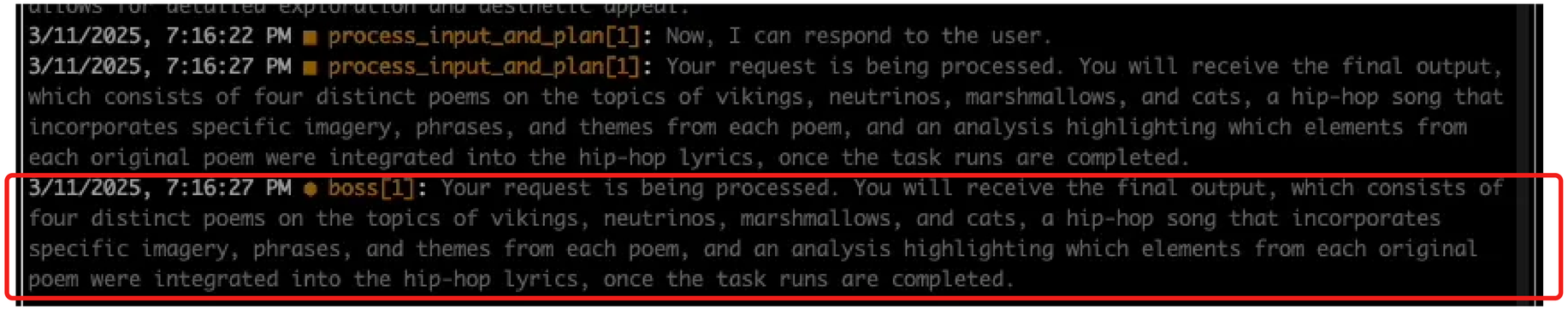


🤖 The Supervisor continues creating the remaining required agent and task configurations in the same manner until all planned configurations are established. Then, it schedules the start of tasks from the first wave.

Showcase

Poetry and Hip-Hop Analysis (6/6)

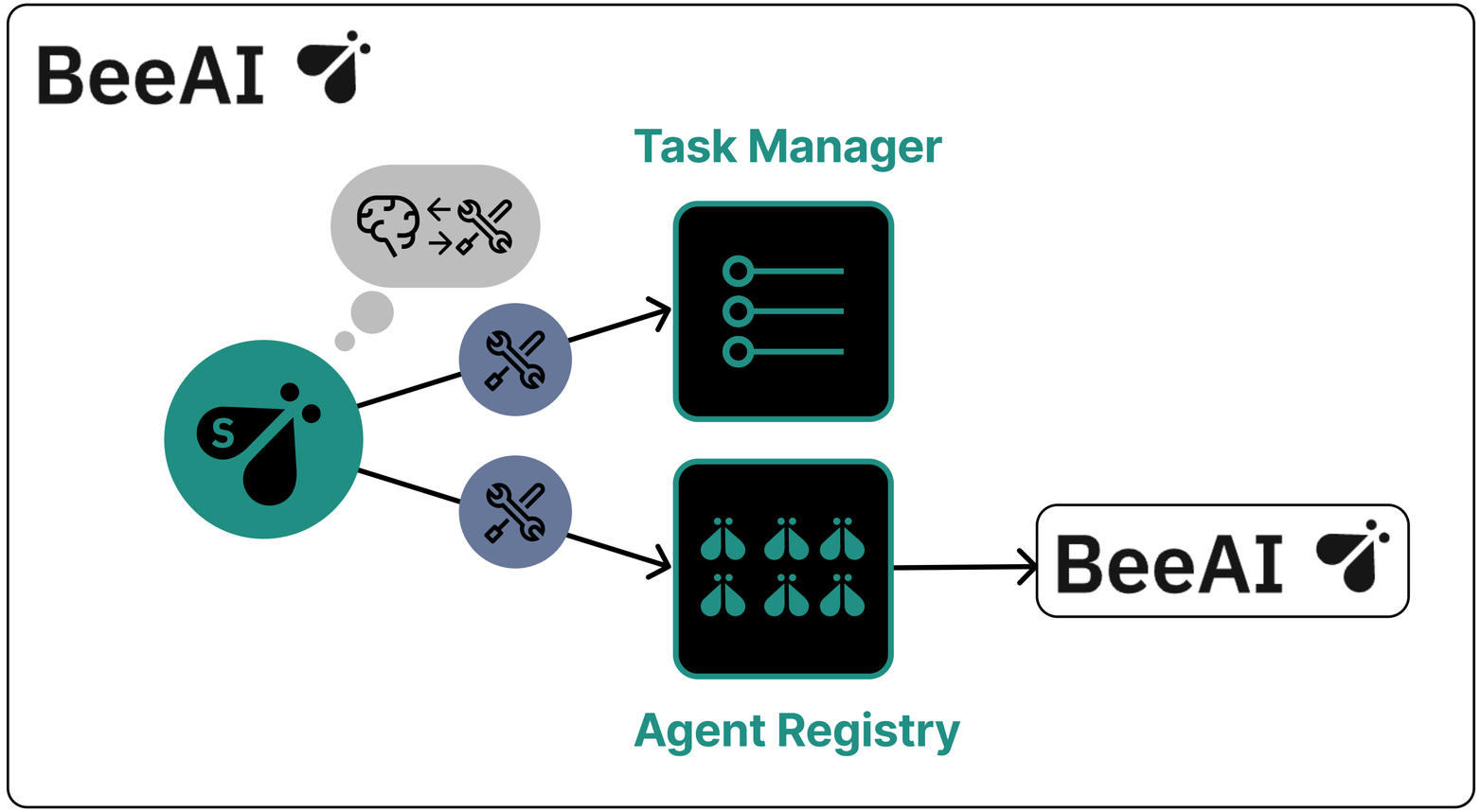
Chat UI



🤖 With everything set up and task runs initiated, the supervisor's immediate role is complete. He can now respond to the user with a status update about his actions and the current system activity.

Complete step-by-step showcase documentation:
https://cheerful-sodalite-38a.notion.site/Creative-Multi-Agent-Tasks-Showcase-Poetry-and-Hip-Hop-Analysis-1b53b270a700800ab1d3eb28f825bd2a

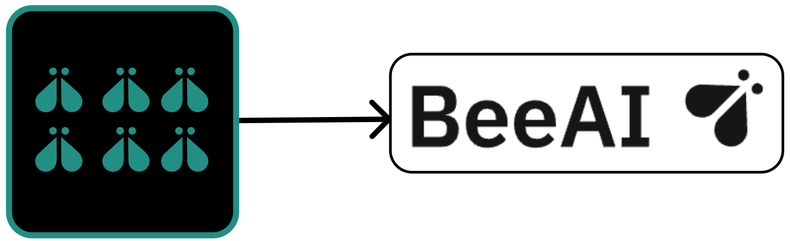
Integration (1/3)



🤖 The Supervisor is a specialized agent created on the BeeAI platform using the original Supervisor as a library, with access to its agent directory.

Integration (2/3)

Agent Registry



Exposed Methods

- ~~Get available tools~~
- Get all agent configurations
- ~~Create agent configuration~~
- ~~Update agent configuration~~
- Get agent configuration

- Get agent
- Get all active agents

- Get pool stats

Limitations

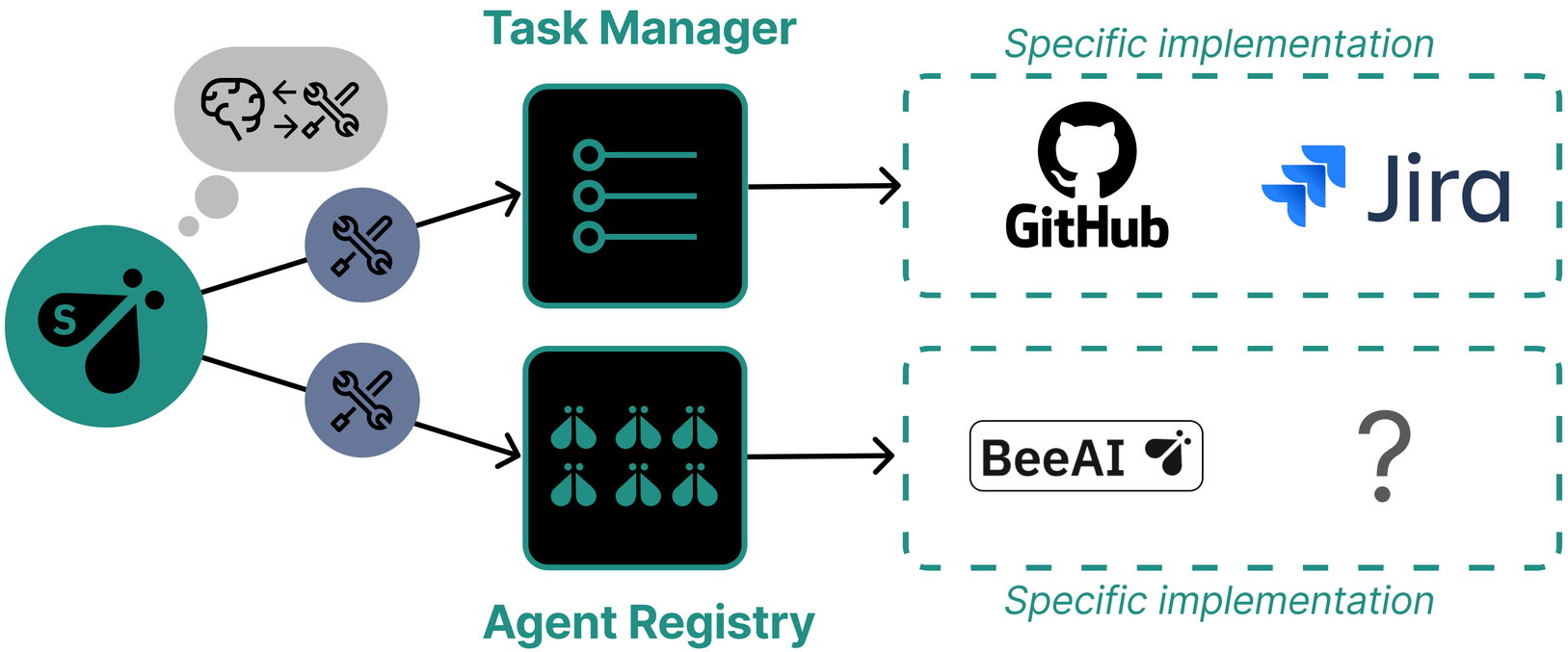
Autonomous Run

~~**Chat Collaboration**~~

~~Built-in UIs~~

The BeeAI platform restricts the functionality of the agent registry in terms of creating or customizing agent configurations. Agent configurations are loaded from the platform and remain immutable, including agent pooling.

Integration (3/3)



💡 The core idea is that, thanks to the tool architecture, we can relatively easily switch the implementation of both main subsystems.

Next Steps (1/2)

- **Workflows configurations**
  Current workflows are ad-hoc, and dependencies between tasks are not persisted. We should introduce workflow configurations to address this.

- **File management**
  To fully utilize the system's potential, agents should be able to work with the file system to create and iterate over generated content.

- **Autonomous steps**
  It should be possible to create a step (a task run) that calls the supervisor, allowing it to autonomously build the following steps at runtime or finish the workflow.

Next Steps (2/2)

- **Supervisors recursion**
  The supervisor should be able to create another supervisor when delegating a large task.

- **Supervisor as an agent workflow**
  The supervisor currently requires a large model for optimal performance. It may be possible to decompose its inner logic into smaller specialized steps using smaller models.

Vision (1/2)

- **The supervisor should become more sophisticated in the area of agent creation.**
  - **Agents' inner workflows**
    The system should support building "inner" agent workflows.

    The motivation is to replace the reasoning of large and expensive LLMs with smaller ones.

    The goal is to decompose a complex problem—one that would typically require a large LLM—into smaller steps (some involving LLM calls, some not) that can instead utilize smaller LLM models.

    Another motivation is to create a more stable system in production by minimizing the scope for LLM errors.

Vision (2/2)

- **The supervisor should become more sophisticated in ensuring output quality.**
  - **Attached Tests**
    Created tasks, workflows, or agents should have generated tests attached with defined acceptance criteria. Each update should also update the tests to ensure lasting quality.
  - **Self-improvement**
    Attached tests, combined with tools that can modify and adjust related entities (workflows, tasks, or agents), should establish a feedback loop that, based on test results, enables continuous improvement of the tested entity.

Links

- **Github**
  https://github.com/i-am-bee/beeai-supervisor
- **Documentation**
  https://cheerful-sodalite-38a.notion.site/BeeAI-Supervisor-1ab3b270a700801cabadc0eb80ae9ddb
- **Showcase**
  https://cheerful-sodalite-38a.notion.site/Creative-Multi-Agent-Tasks-Showcase-Poetry-and-Hip-Hop-Analysis-1b53b270a700800ab1d3eb28f825bd2a
- **Competition Comparison**
  https://cheerful-sodalite-38a.notion.site/Competition-comparison-1b53b270a70080e3aac3d0e34dbbe59a