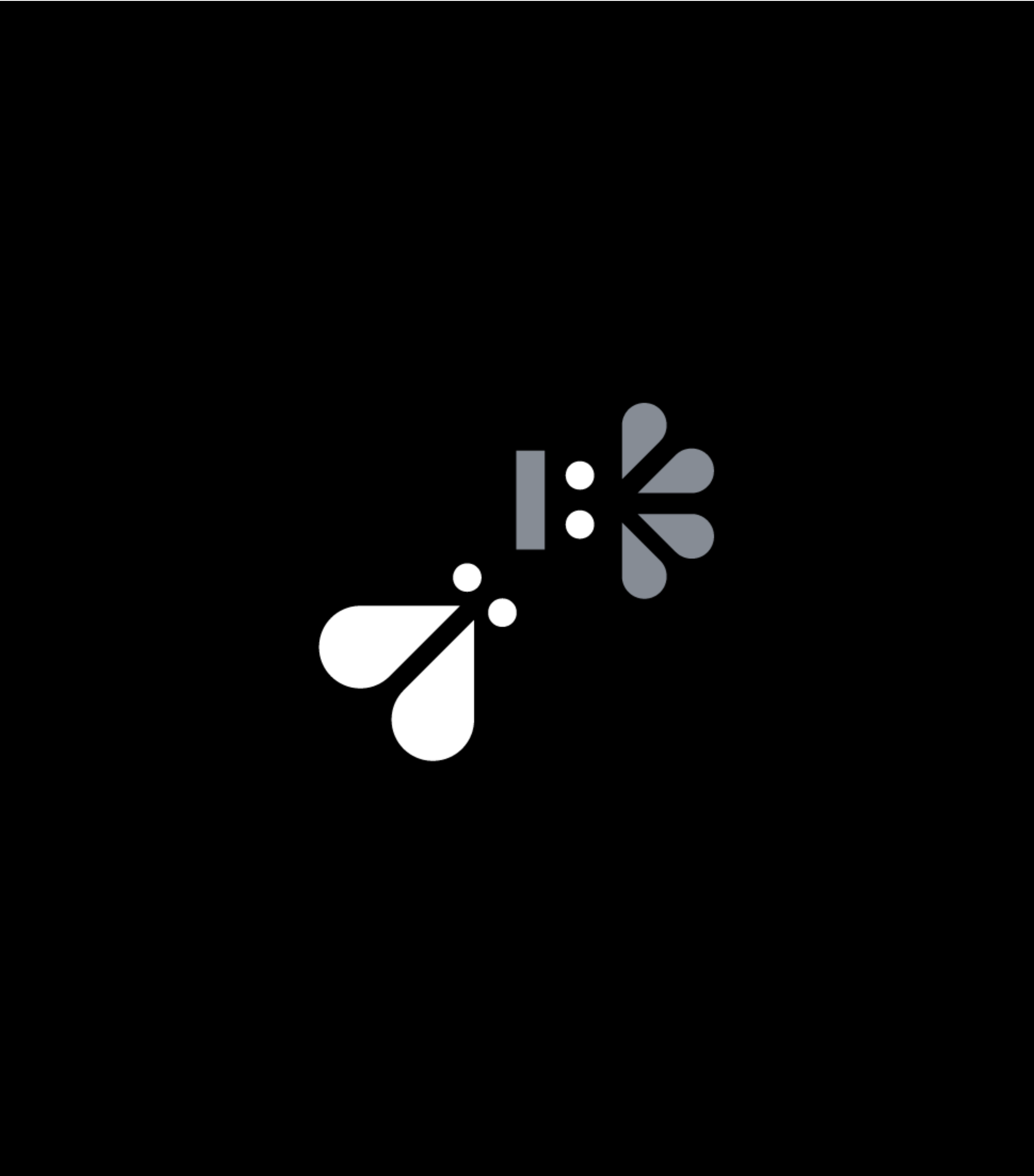


IBM Research

BeeAI

Framework | Platform



Introductions



Ajethaa Pandian
Senior Technical Staff Member
IBM Chief Data Office

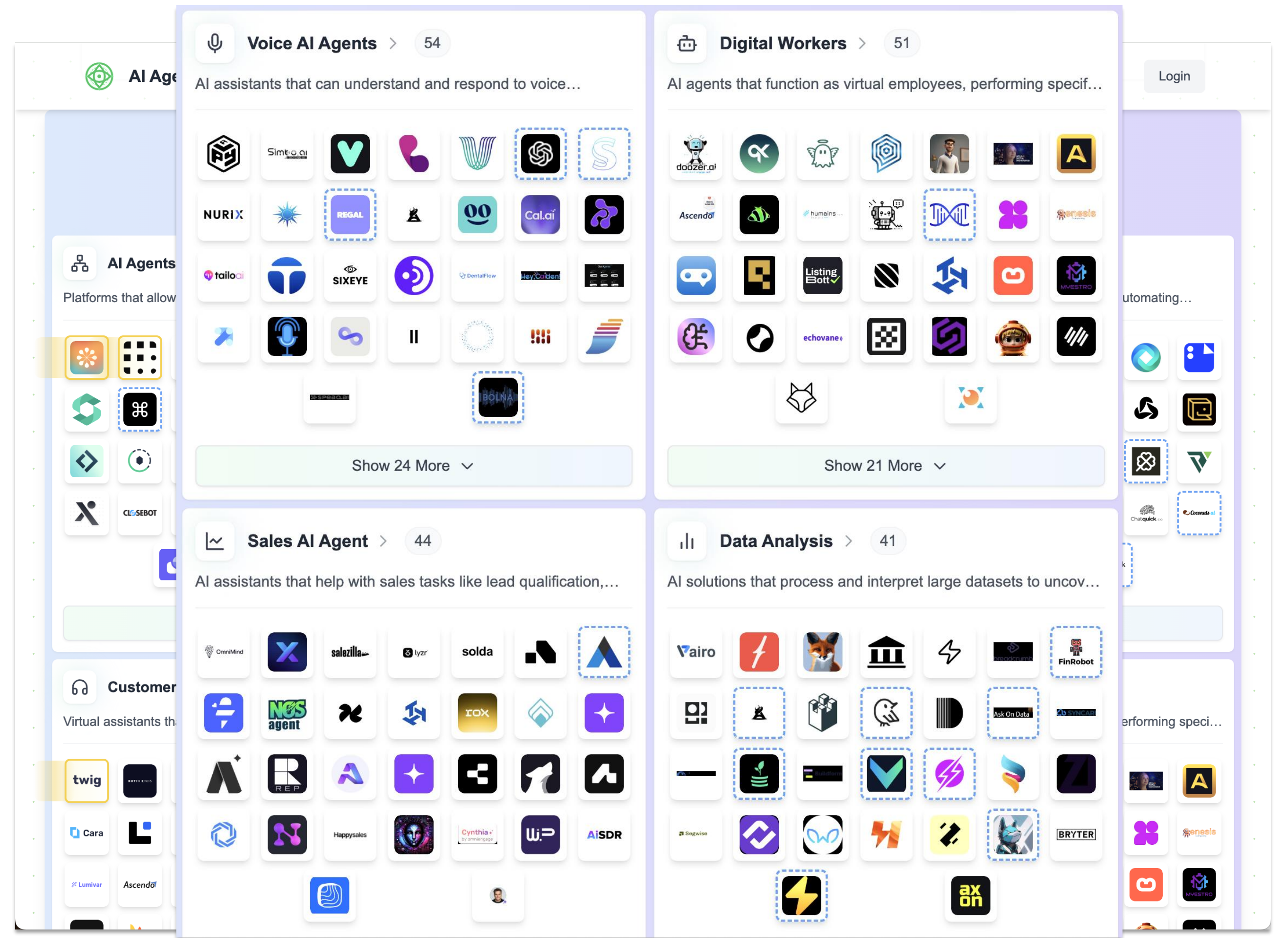


Madeline Lutwyche
Software Developer
IBM Chief Data Office

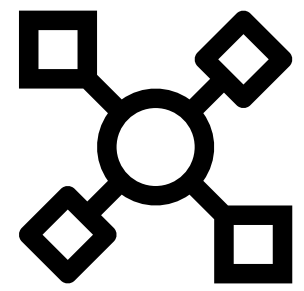
AI Agent Landscape

The AI agent ecosystem is rapidly growing and evolving.

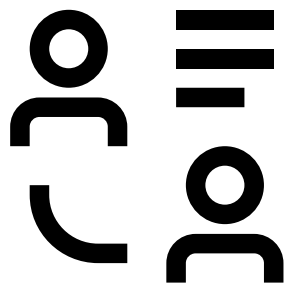
As AI agents become increasingly capable, teams continue to face major challenges.



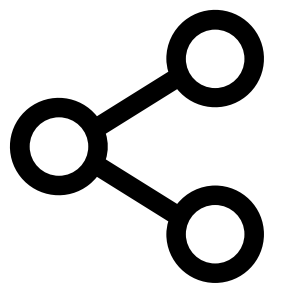
The Problem



Building agents can be **messy** when not made intuitive



Creating **user interfaces** for agents often requires frontend expertise

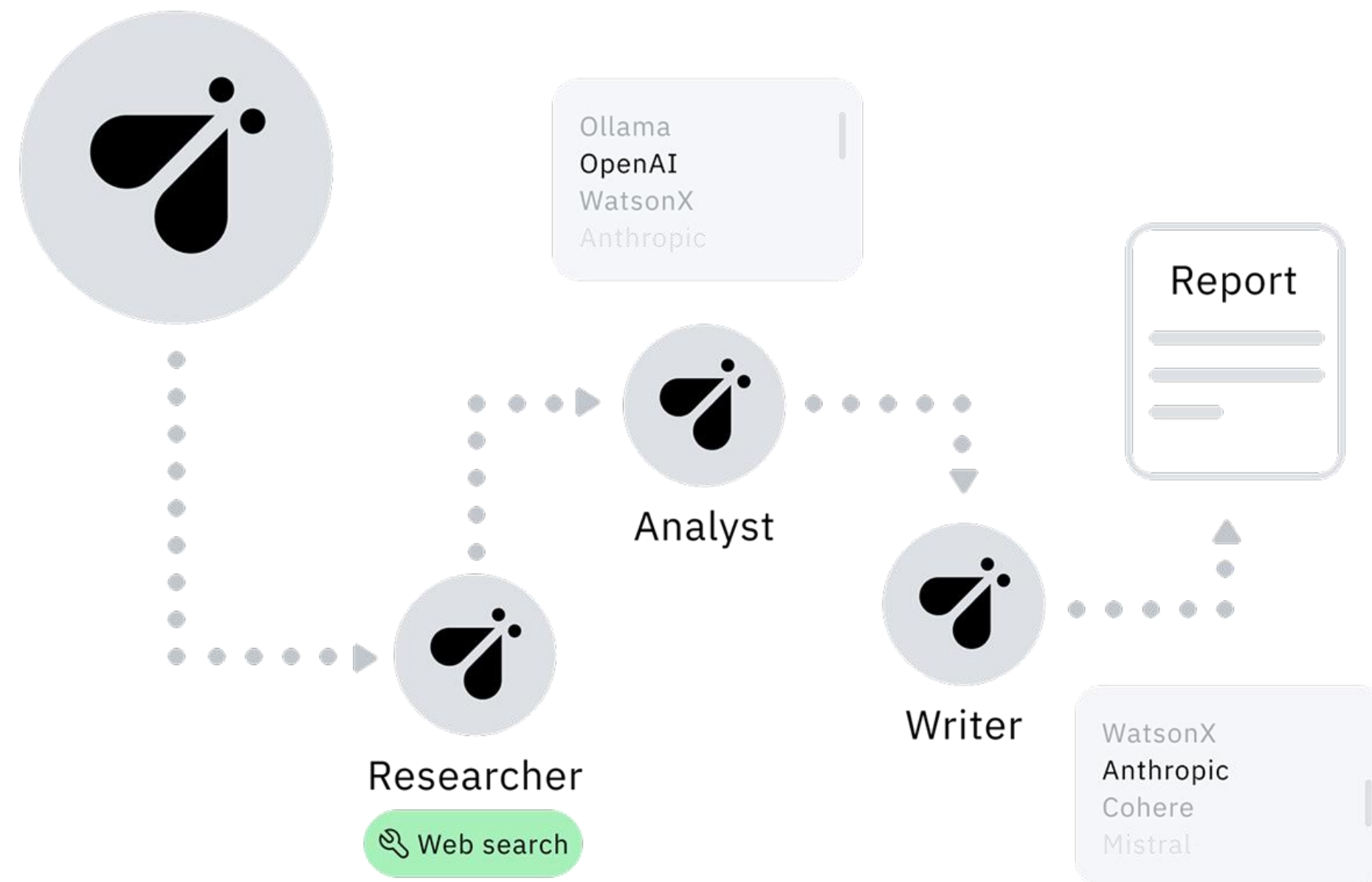


It's hard to **share and run agents** beyond your own development environment

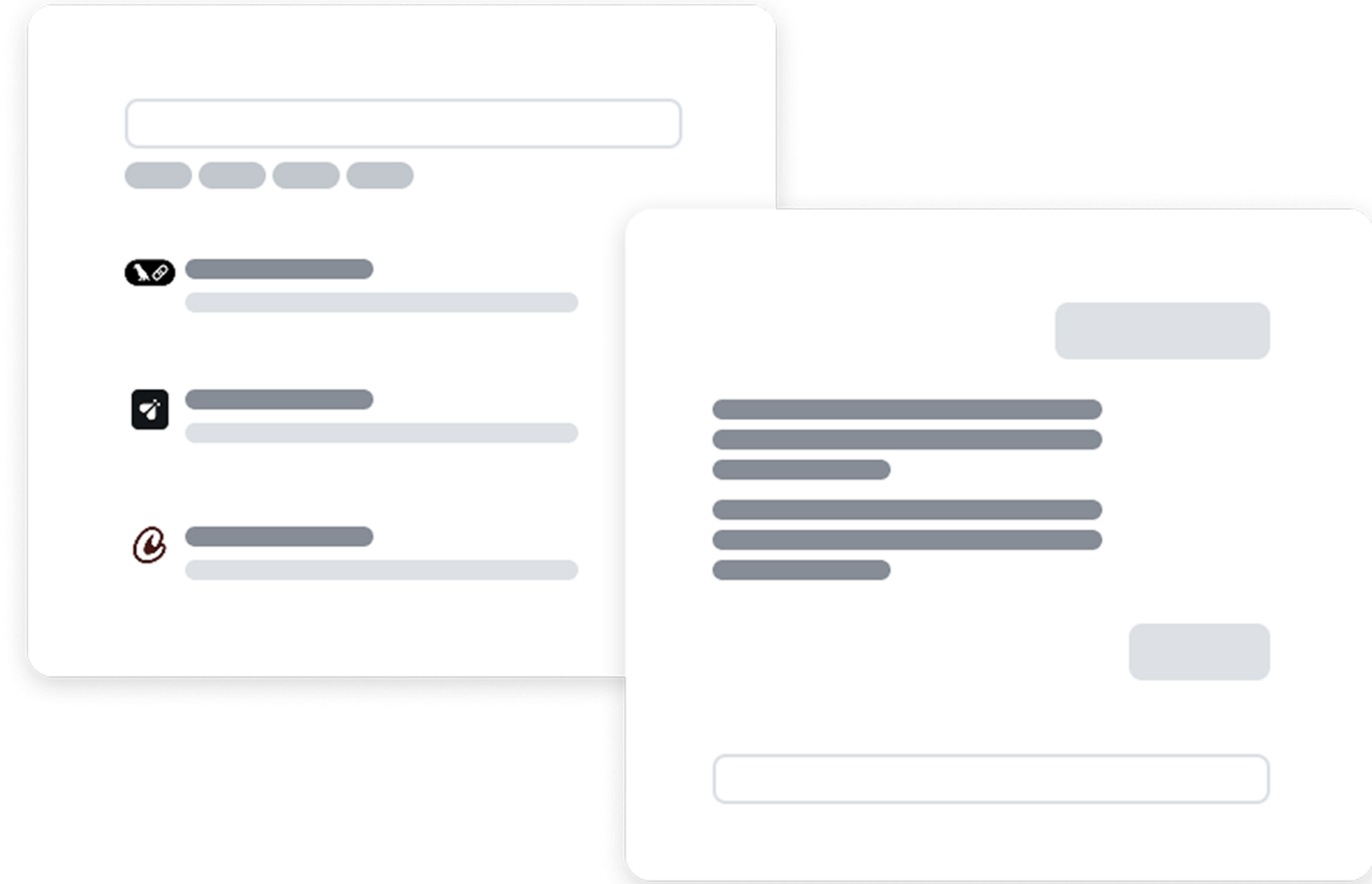
Together, these challenges slow down real-world adoption and collaboration.

BeeAI solves this with two complimentary components

Framework

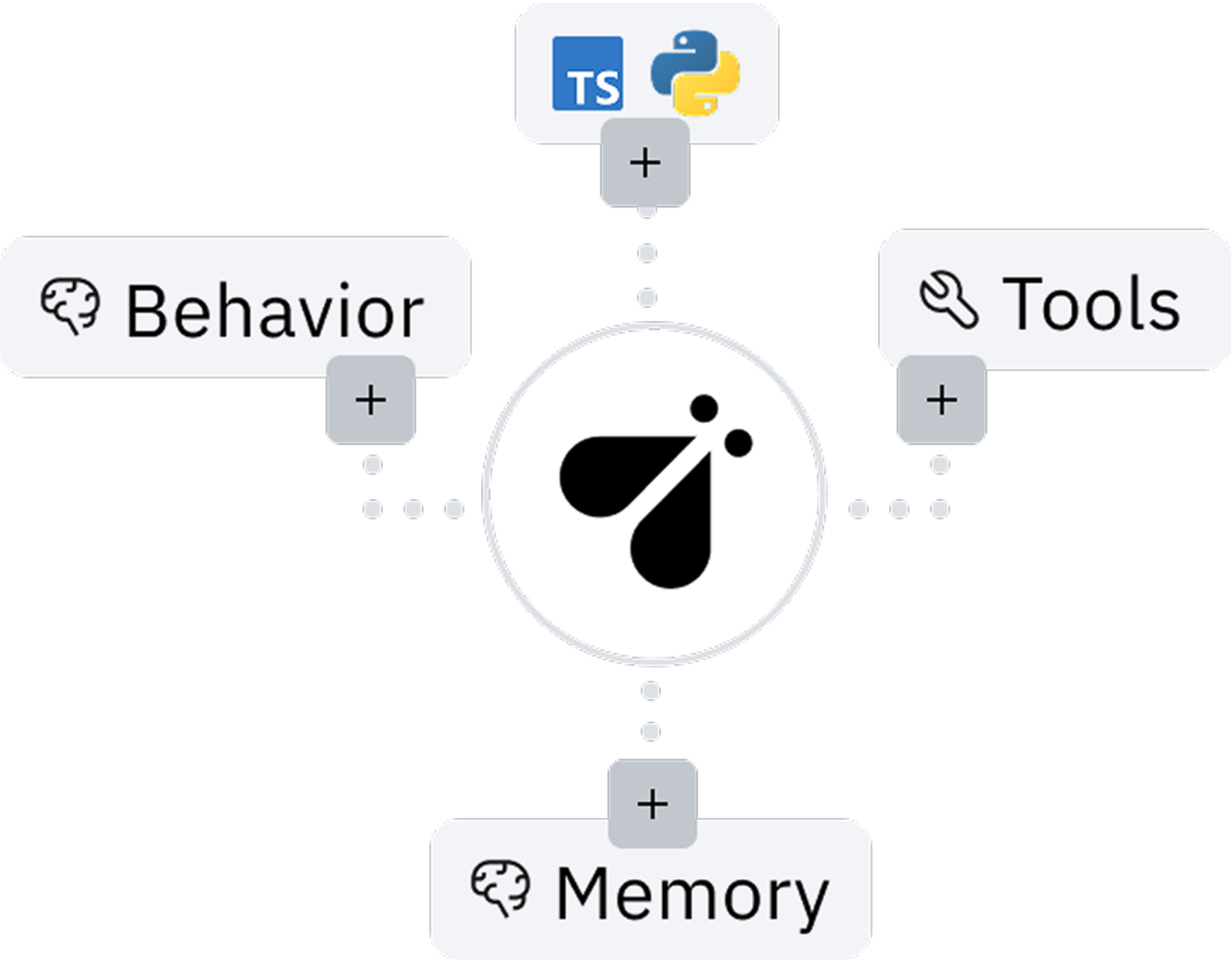


Platform

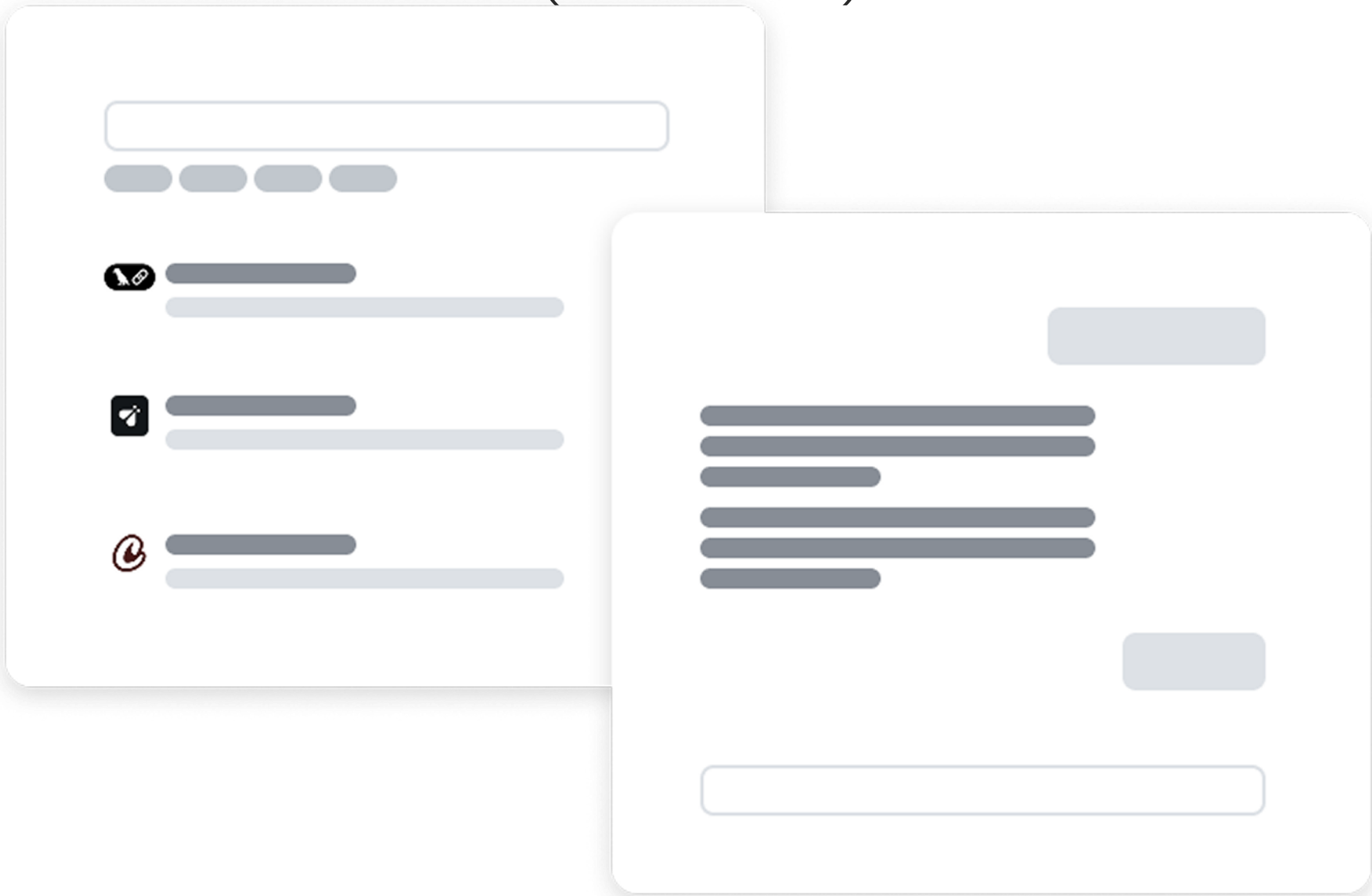


The BeeAI Lifecycle

1. Build (Framework)



2. Compose UI & Deploy Agents (Platform)

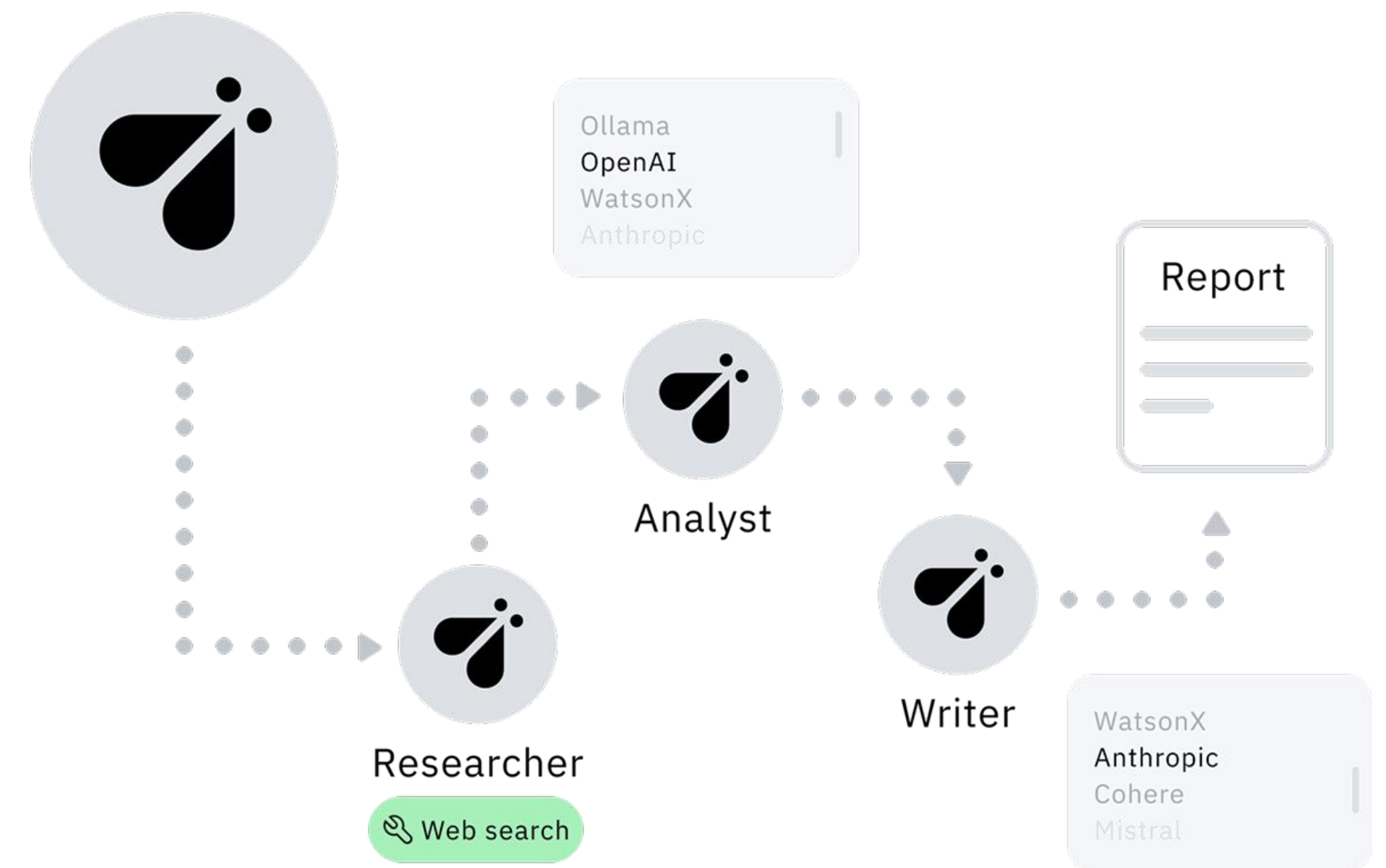


BeeAI Framework

A production-grade framework for building AI agents

Key Features

- Build **complex multi-agent** workflows with state handling, built-in caching, memory optimization, and resource management.
- LLM/Provider Agnostic
- Dual language support with Python & TypeScript
- **Observability** with real-time monitoring, OpenTelemetry, and tracing
- Maintain complete control **over agent behavior, performance optimization, and resource allocation**
- **Integrates into your existing stack** with MCP compatibility, custom tool development support, and seamless tool integration



★ STAR to follow along with the BeeAI Framework

A production-grade framework for building AI agents



BeeAI Platform

Quickly build shareable web interfaces for your agents- no frontend skills needed.

Key Features

- **Create frontend interfaces** for your Agents with ease
- **Run** agents in the CLI or through the UI
- **Share** your agents with friends or teammates
- **Bring** in agents from any frameworks (BeeAI, langchain, crewAI, etc)
- **Connect** to any LLM provider
- **Observe traces** under the hood with build in telemetry
- **Enable agent interoperability** by using A2A and the BeeAI SDK under the hood

The screenshot displays the BeeAI Platform interface. At the top right, an agent named 'ollama_deep_researcher' is shown. Below it, a prompt asks 'What topic do you want to research?'. A text input field contains the placeholder 'Write the agent prompt here', and a 'Run' button is to its right. On the left, a preview of a research report titled 'What is Vibe Coding?' is shown. The report includes an 'Introduction' section and a 'Definition and Principles' section. The 'Introduction' section describes 'Vibe coding' as an emerging approach in software development that emphasizes code readability, maintainability, and developer experience. The 'Definition and Principles' section defines 'Vibe coding' as focusing on creating intuitive, elegant, and maintainable code.

ollama_deep_researcher

What topic do you want to research?

Write the agent prompt here

Run

What is Vibe Coding?

Introduction

Vibe coding is an emerging approach in software development that emphasizes the importance of code readability, maintainability, and overall developer experience. It prioritizes the "feel" or "vibe" of the code, ensuring that it is not only functional but also enjoyable to read and work with. This approach is gaining traction in the developer community as it aligns with the growing recognition of the importance of clean code and its impact on team productivity and software quality.

Definition and Principles

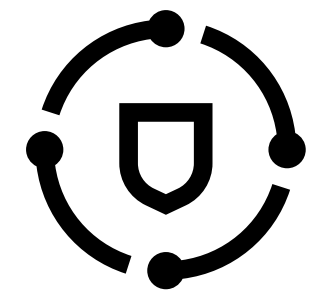
Vibe coding is defined by its focus on creating code that is intuitive, elegant, and maintainable. It goes beyond mere functionality, emphasizing the aesthetic and emotional aspects of coding. The core principles of vibe coding include:

Get Started with the BeeAI Platform

Quickly build shareable web interfaces for your agents- no frontend skills needed.

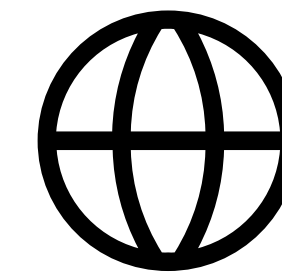


BeeAI is Open by Design



Linux Foundation Governance

Every component of BeeAI is developed under **the Linux Foundation's open governance** model, ensuring long-term sustainability, neutral oversight, and transparent community-led development.



Community- Driven Roadmap

BeeAI evolves through contributions from developers, researchers, and organizations. We prioritize real-world use cases, simplicity, and openness to shape the protocol, platform, and framework.

Learn how to contribute:

<https://github.com/i-am-bee/beeai-platform/blob/main/CONTRIBUTING.md>

Workshop Time!

<https://ibm.github.io/beeai-workshop/>



BeeAI Workshop



Search



beeai-workshop

☆ 8 🍴 8

Workshop Overview: Company Analysis Agent with
the BeeAI Framework

Table of contents

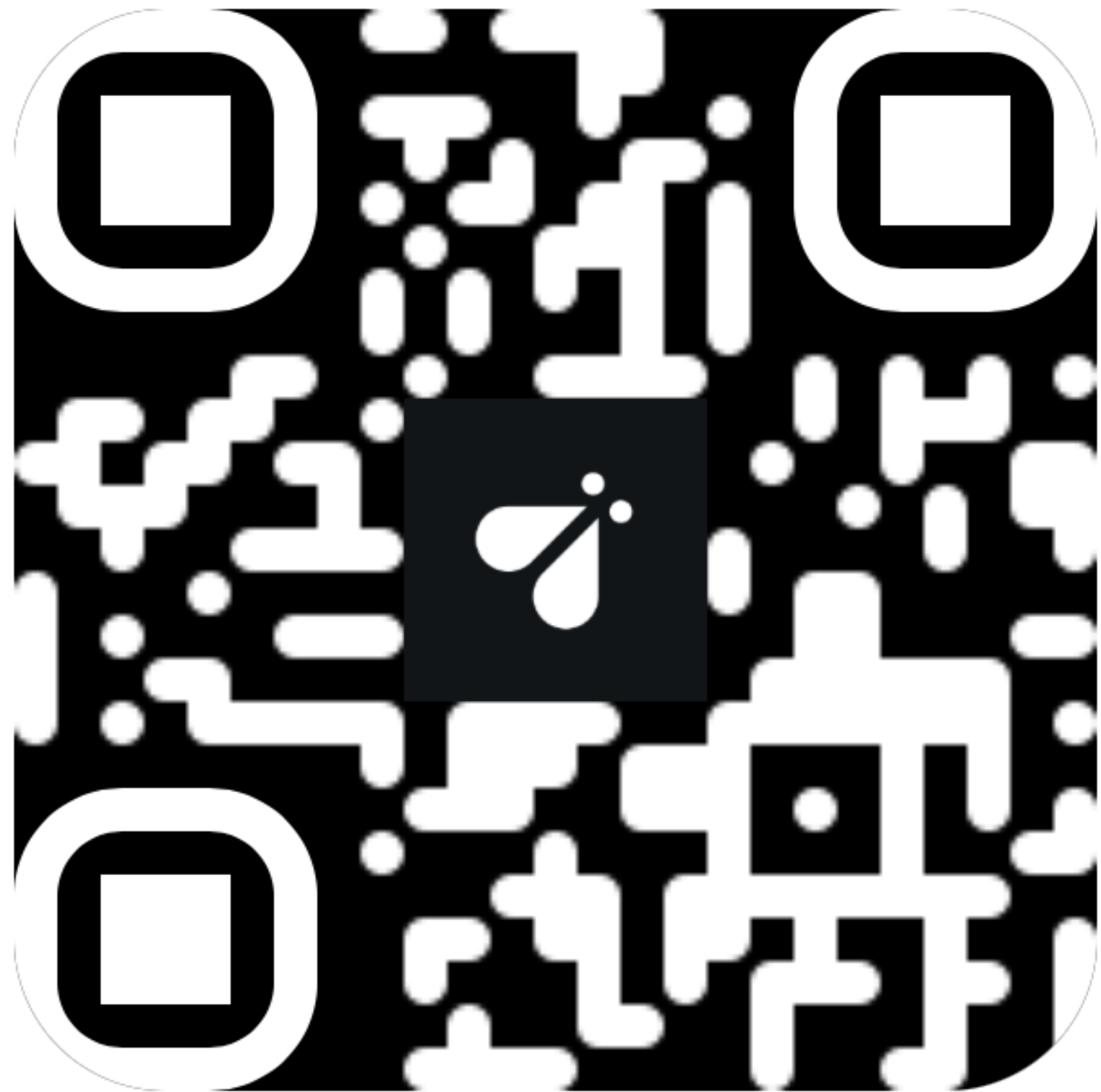
What You'll Build

Workshop Structure

Learn More About BeeAI

What's next

Try BeeAI



Star to follow

 github.com/i-am-bee

BeeAI

Agents

Search the agent catalog

All

BeeAI

AutoGen

CrewAI

Custom

LangGraph

Showing 12 agents

Import agents +

agent-docs-creator

🕒 19s/run (avg) | 5409 tokens/run (avg) | Apache 2.0

The agent analyzes AI source code to generate structured, clear, and complete documentation in a consistent template, supporting multiple languages.

BeeAI

aider

🕒 5s/run (avg) | 5000 tokens/run (avg) | Apache 2.0

An AI pair programmer that edits code in a local Git repository using natural language, executing commands and providing feedback.

Custom

chat

🕒 3s/run (avg) | 89 tokens/run (avg) | Apache 2.0

The agent is an AI-powered conversational system with memory, supporting real-time search, Wikipedia lookups, and weather updates through integrated tools.

BeeAI

content-judge

🕒 22s/run (avg) | 1229 tokens/run (avg) | Apache 2.0

Evaluates multiple documents and agent-generated content based on correctness, depth, clarity, and relevance, selecting the highest-scoring one. It ensures optimal...

BeeAI

gpt-researcher

🕒 2.1s/run (avg) | 111 tokens/run (avg) | Apache 2.0

13

