mcp-agent

lastmile AI

mcp-agent

Anatomy of an MCP Agent

AGENT TECH STACK IN 2025

Recent developments make Agent design simpler and more robust

AGENTS AS MCP SERVERS

MCP servers can be a lot more than tool call wrappers – they can be Agents!

AGENTS AS ASYNC WORKFLOWS

Rethinking agent architecture

Agent tech stack in 2025

3 big changes are converging to enable effective agents that work well in production

Better Models

With test-time compute & reasoning models, a lot of complexity is **shifting-left** into inference layer.

i.e. less complexity for app developers!

Model Context Protocol

MCP is a standardized interface for connecting LLMs to tools & data.

Proliferation of AI-native services (MCP servers) and a **single way** to give context to LLMs.

Simplified Architecture

Agents = apps that orchestrate LLMs with MCP-enabled tools and resources.

No need for monolithic AI frameworks anymore.

Simple agent patterns is all you need.

mcp-agent

Agent tech stack in 2025

3 big changes are converging to enable effective agents that work well in production

Better Models

With test-time compute & reasoning models, a lot of complexity is **shifting-left** into inference layer.

i.e. less complexity for app developers!

Model Context Protocol

MCP is a standardized interface for connecting LLMs to tools & data.

Proliferation of AI-native services (MCP servers) and a **single way** to give context to LLMs.

Simplified Architecture

Agents = apps that orchestrate LLMs with MCP-enabled tools and resources.

No need for monolithic AI frameworks anymore.

Simple agent patterns is all you need.

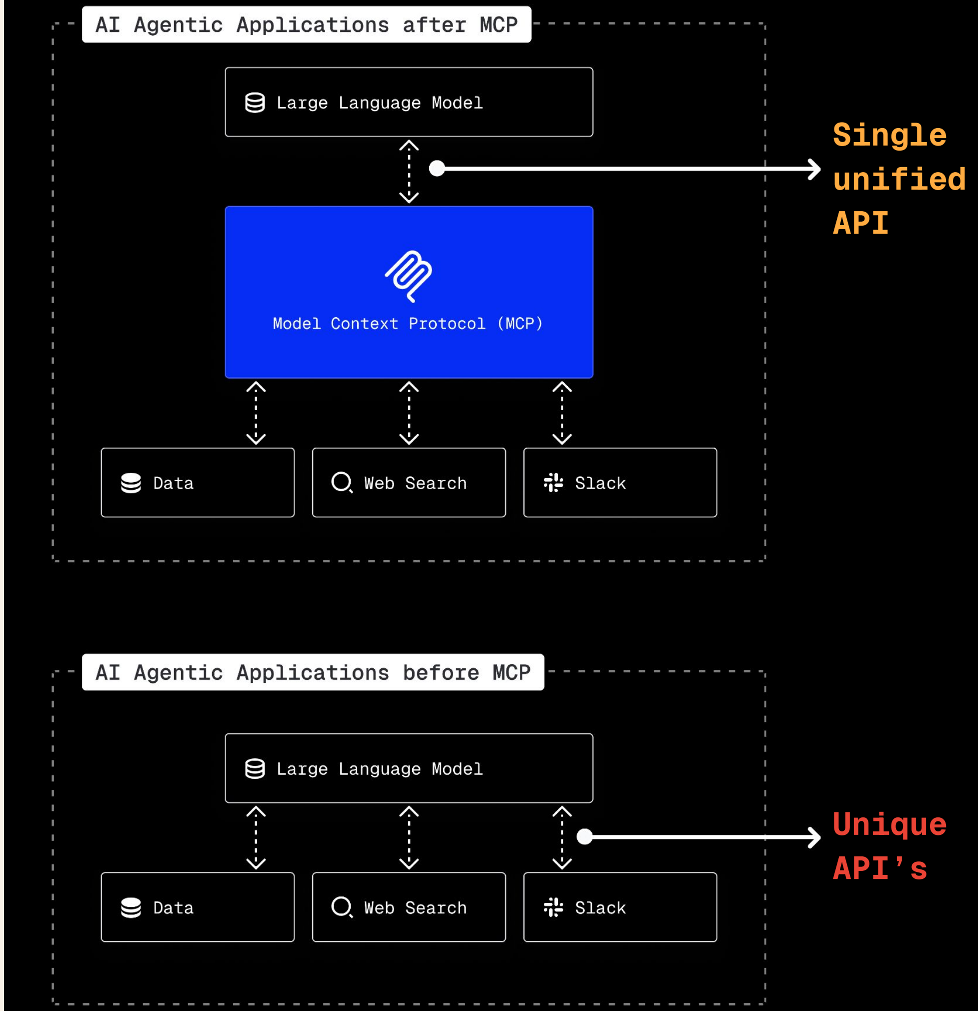
mcp-agent

Agent tech stack in 2025

Model Context Protocol

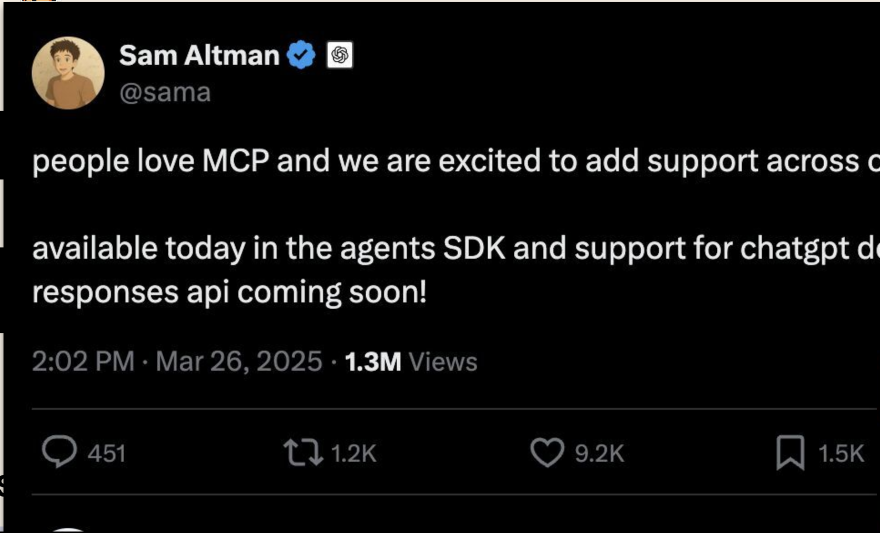
MCP is a standardized interface for connecting LLMs to tools & data.

Proliferation of AI-native services (MCP servers) and a **single way** to give context to LLMs.



mcp-agent 🐕

Agen
stac



Sam Altman @sama

people love MCP and we are excited to add support across our products.

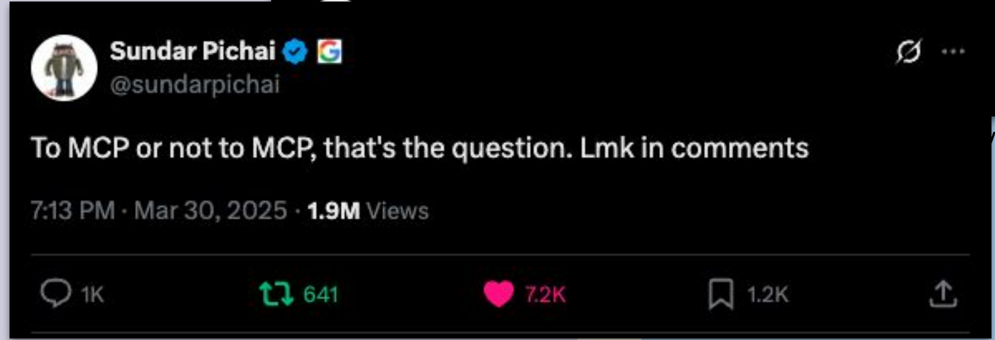
available today in the agents SDK and support for chatgpt desktop app + responses api coming soon!

2:02 PM · Mar 26, 2025 · **1.3M** Views

451 1.2K 9.2K 1.5K

ges are converging to enable
fective agents that work well
in production

Better Models



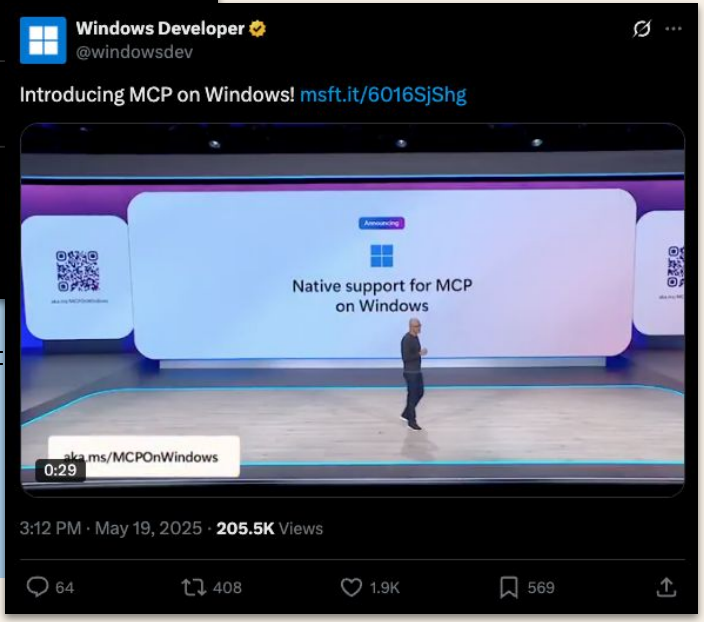
Sundar Pichai @sundarpichai

To MCP or not to MCP, that's the question. Lmk in comments

7:13 PM · Mar 30, 2025 · **1.9M** Views

1K 641 7.2K 1.2K

ices (MCP
give context



Windows Developer @windowsdev

Introducing MCP on Windows! msft.it/6016SjShg

3:12 PM · May 19, 2025 · **205.5K** Views

64 408 1.9K 569

Agent tech stack in 2025

3 big changes are converging to enable effective agents that work well in production

Better Models

With test-time compute & reasoning models, a lot of complexity is **shifting-left** into inference layer.

i.e. less complexity for app developers!

Model Context Protocol

MCP is a standardized interface for connecting LLMs to tools & data.

Proliferation of AI-native services (MCP servers) and a **single way** to give context to LLMs.

Simplified Architecture

Agents = apps that orchestrate LLMs with MCP-enabled tools and resources.

No need for monolithic AI frameworks anymore.

Simple agent patterns is all you need.

mcp-agent

Agent tech stack in 2025

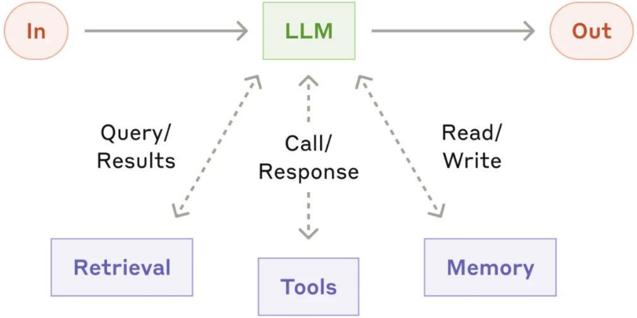
Simplified Architecture

Agents = apps that orchestrate LLMs with MCP-enabled tools and resources.

No need for monolithic AI frameworks anymore.

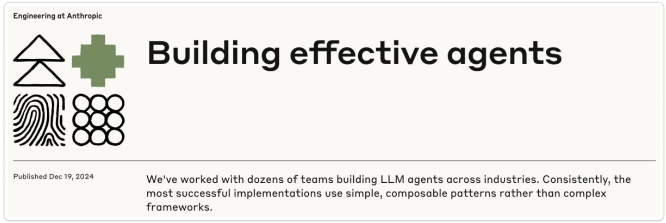
Simple agent patterns is all you need.

Augmented LLM



LLM with access to **tools** and **resources** (data).

Base building block of agent workflows: run LLM in a loop.



Engineering at Anthropic

Building effective agents

Published Dec 19, 2024

We've worked with dozens of teams building LLM agents across industries. Consistently, the most successful implementations use simple, composable patterns rather than complex frameworks.

mcp-agent

Agent tech stack in 2025

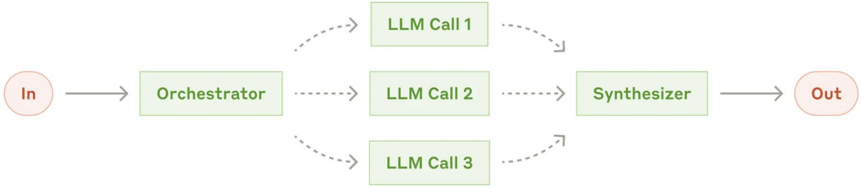
Simplified Architecture

Agents = apps that orchestrate LLMs with MCP-enabled tools and resources.

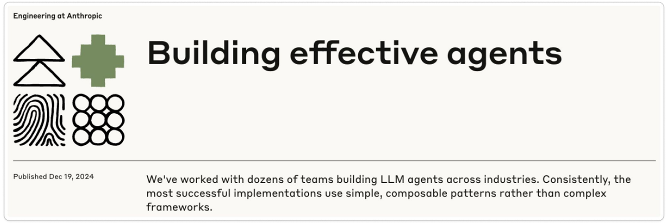
No need for monolithic AI frameworks anymore.

Simple agent patterns is all you need.

Orchestrator Workflow



A higher-level LLM generates a **plan**, then assigns them to **sub-agents**, and **synthesizes** the results.



Engineering at Anthropic

Building effective agents

Published Dec 19, 2024

We've worked with dozens of teams building LLM agents across industries. Consistently, the most successful implementations use simple, composable patterns rather than complex frameworks.

mcp-agent

Agent tech stack in 2025

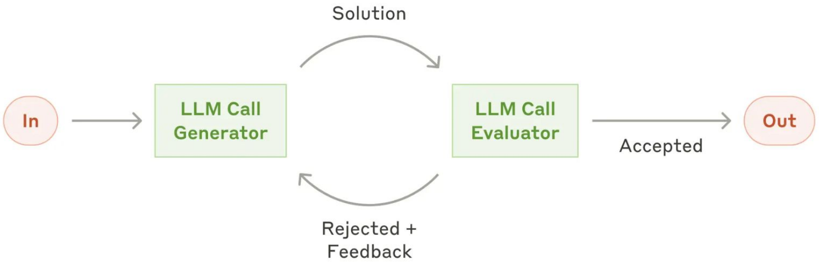
Simplified Architecture

Agents = apps that orchestrate LLMs with MCP-enabled tools and resources.

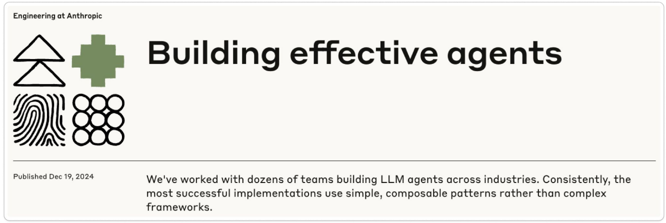
No need for monolithic AI frameworks anymore.

Simple agent patterns is all you need.

Evaluator-Optimizer



One LLM (the “**optimizer**”) refines a response.
Other LLM (ie “**evaluator**”) critiques it until a response exceeds a quality criteria.



Engineering at Anthropic

Building effective agents

Published Dec 19, 2024

We've worked with dozens of teams building LLM agents across industries. Consistently, the most successful implementations use simple, composable patterns rather than complex frameworks.

mcp-agent

Agent tech stack in 2025

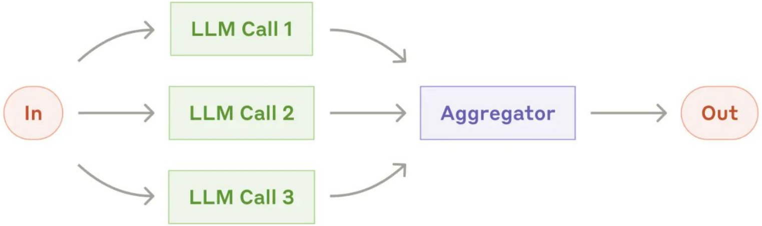
Simplified Architecture

Agents = apps that orchestrate LLMs with MCP-enabled tools and resources.

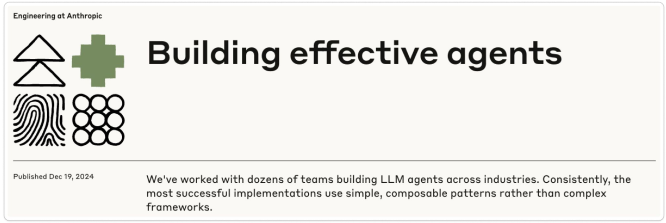
No need for monolithic AI frameworks anymore.

Simple agent patterns is all you need.

Parallel Workflow



Fan-out tasks to multiple sub-agents and **fan-in** the results.


Engineering at Anthropic

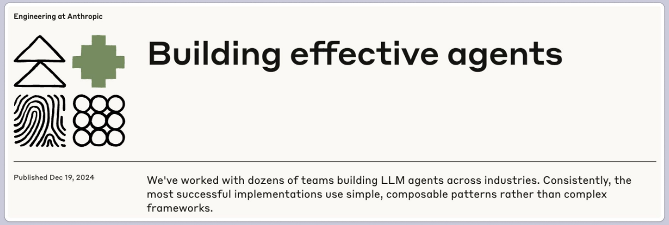
Building effective agents

Published Dec 19, 2024

We've worked with dozens of teams building LLM agents across industries. Consistently, the most successful implementations use simple, composable patterns rather than complex frameworks.

mcp-agent = Model Context Protocol

🤝



Engineering at Anthropic

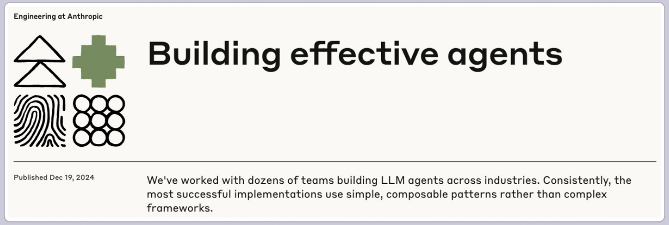
Building effective agents

Published Dec 19, 2024

We've worked with dozens of teams building LLM agents across industries. Consistently, the most successful implementations use simple, composable patterns rather than complex frameworks.

mcp-agent = Model Context Protocol

🤝



Engineering at Anthropic

Building effective agents

Published Dec 19, 2024

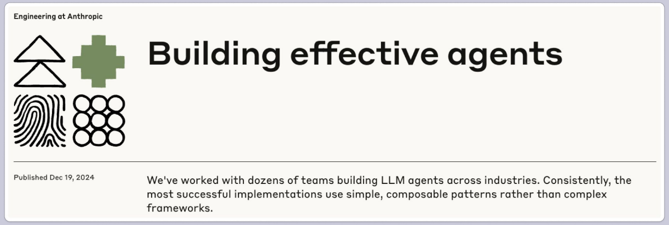
We've worked with dozens of teams building LLM agents across industries. Consistently, the most successful implementations use simple, composable patterns rather than complex frameworks.

Key observations:

1. **MCP everywhere**:
   a. Every line-of-business application will soon be an **MCP client.**
   b. Every service will have an **MCP server.**

mcp-agent = Model Context Protocol

🤝



Engineering at Anthropic

Building effective agents

Published Dec 19, 2024

We've worked with dozens of teams building LLM agents across industries. Consistently, the most successful implementations use simple, composable patterns rather than complex frameworks.

Key observations:

1. **MCP everywhere**:
   a. Every line-of-business application will soon be an **MCP client.**
   b. Every service will have an **MCP server.**

2. **Agents as MCP servers:** Agents are microservices that should be deployed as MCP servers themselves.

mcp-agent = Model Context Protocol

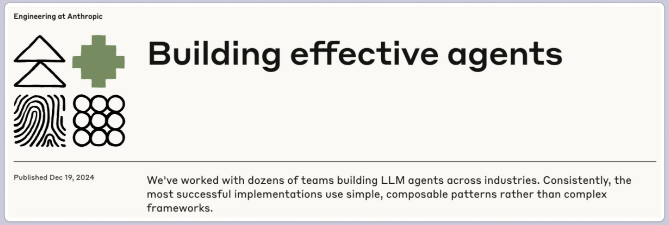
🤝

Key observations:

1. **MCP everywhere**:
   a. Every line-of-business application will soon be an **MCP client.**
   b. Every service will have an **MCP server.**

2. **Agents as MCP servers:** Agents are microservices that should be deployed as MCP servers themselves.

3. **Agents are async workflows**: Agents should be modeled like workflows (think Airflow, Temporal, etc.)



Engineering at Anthropic

Building effective agents

Published Dec 19, 2024

We've worked with dozens of teams building LLM agents across industries. Consistently, the most successful implementations use simple, composable patterns rather than complex frameworks.

Agents as MCP servers

Currently, all “agentic” behavior in MCP is on the MCP client:

Clients like Claude or Cursor use MCP servers to solve tasks.

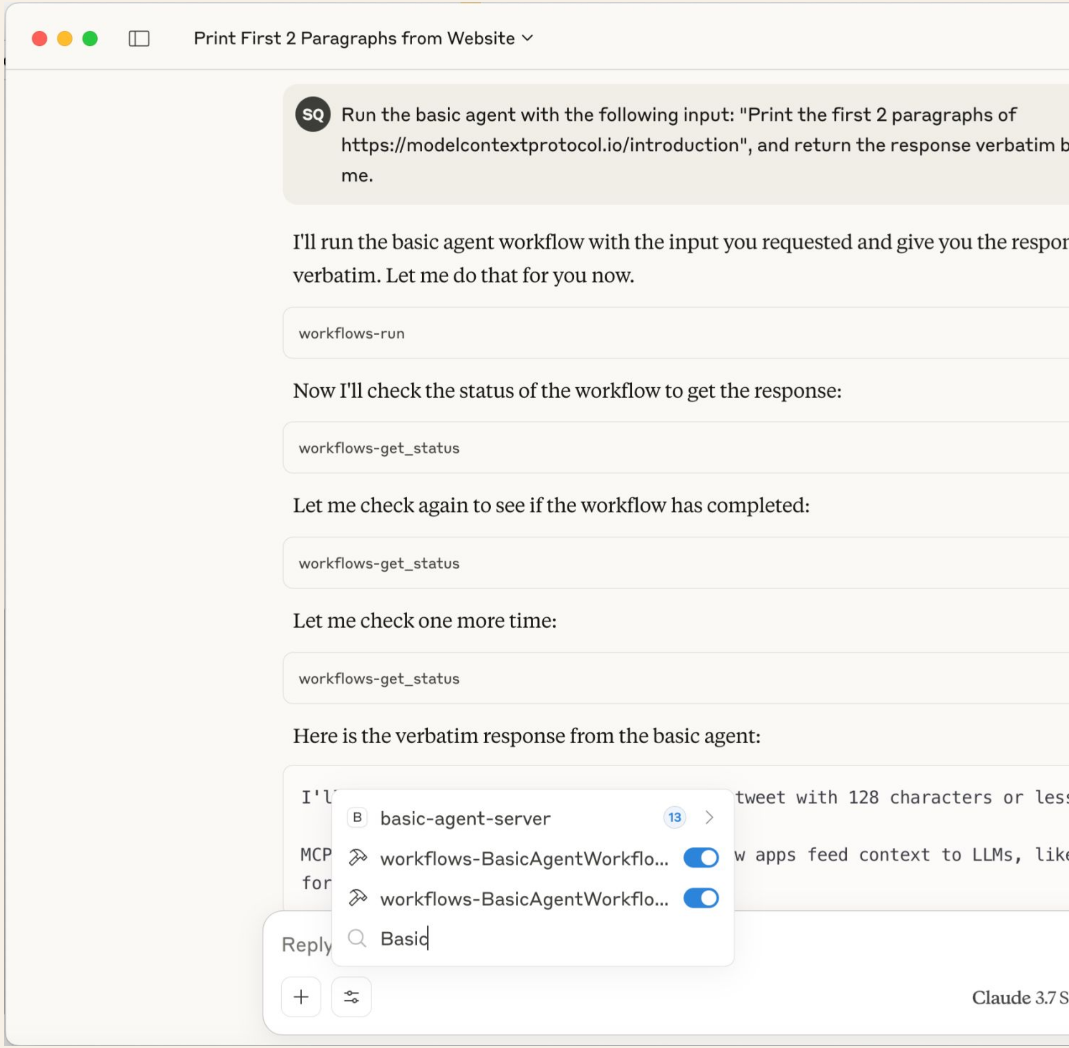
Agents as MCP servers

Currently, all “agentic” behavior in MCP is on the MCP client:

Clients like Claude or Cursor use MCP servers to solve tasks.

What if agents are exposed as MCP servers?

Then any MCP client can
invoke, coordinate and orchestrate agents
the same as any other MCP server.



Run the basic agent with the following input: "Print the first 2 paragraphs of https://modelcontextprotocol.io/introduction", and return the response verbatim b
me.

I'll run the basic agent workflow with the input you requested and give you the respor
verbatim. Let me do that for you now.

workflows-run

Now I'll check the status of the workflow to get the response:

workflows-get_status

Let me check again to see if the workflow has completed:

workflows-get_status

Let me check one more time:

workflows-get_status

Here is the verbatim response from the basic agent:

```
I'l                                   tweet with 128 characters or less

MCP                                   w apps feed context to LLMs, like
for
```

B basic-agent-server 13
workflows-BasicAgentWorkflo...
workflows-BasicAgentWorkflo...
Basic
Reply
+
Claude 3.7 S

Agents as MCP servers

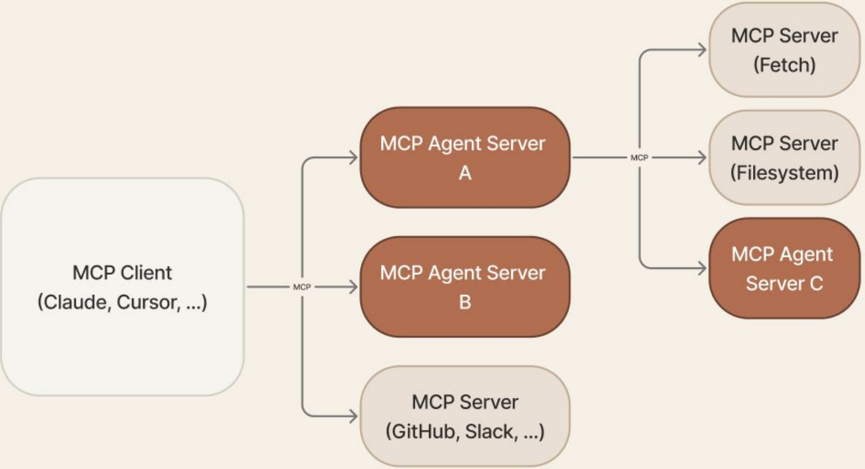
Currently, all “agentic” behavior in MCP is on the MCP client:

Clients like Claude or Cursor use MCP servers to solve tasks.

What if agents are exposed as MCP servers?

Then any MCP client can
invoke, coordinate and orchestrate agents
the same as any other MCP server.

Multi-agents: Agents can then invoke other agents over MCP!



Agents as MCP servers

MCP Agent Servers unlock
big benefits

Composable Agents

Build complex multi-agent systems using the same base protocol (MCP).

Platform-agnostic Agents

Build agents once, reuse anywhere.

Use your agents from any MCP client.

Scalable Agents

Run agent workflows on dedicated infrastructure.

Agent execution is decoupled from the client invoking the agent.

Agents as async workflows

Agents can be **paused** & **resumed**.

They need to await on **human feedback.**

Agent tasks may **fail** and need to be **retried**.

Agents can be **triggered** or **scheduled**
(chat message, or webhook, or cron job).

Agents as async workflows

Agents can be **paused** & **resumed**.

They need to await on **human feedback.**

Agent tasks may **fail** and need to be **retried**.

Agents can be **triggered** or **scheduled**
(chat message, or webhook, or cron job).

Therefore, Agents should be represented as
async workflows.

Agents as async workflows

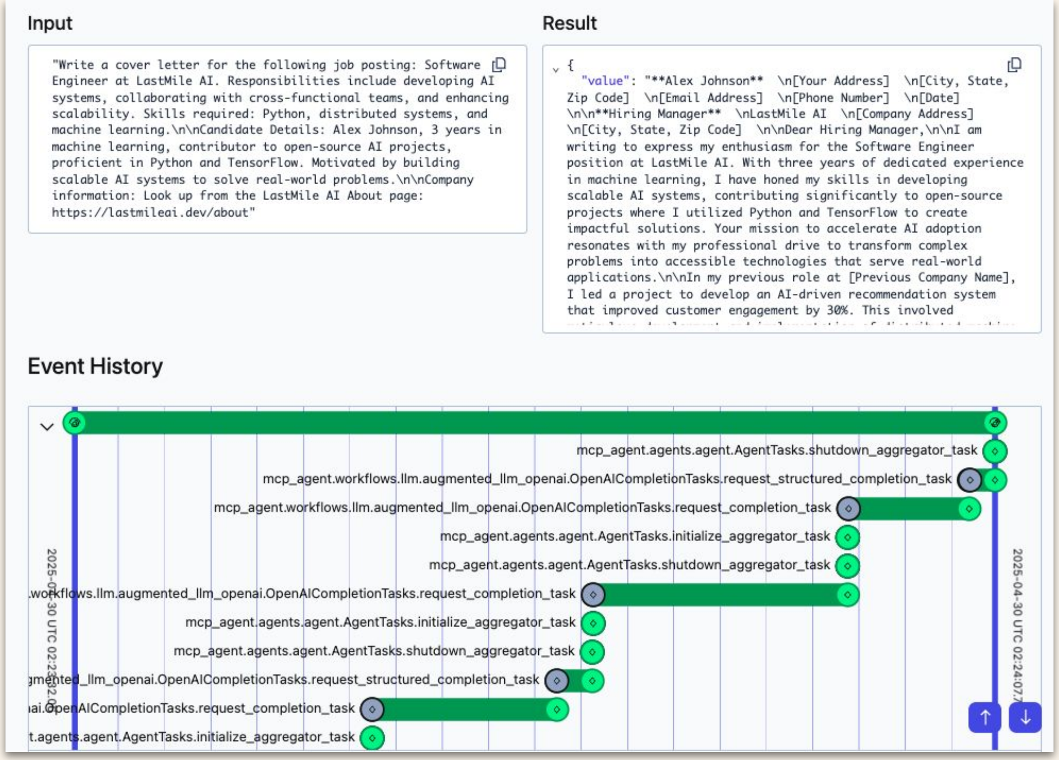
Agents can be **paused** & **resumed**.

They need to await on **human feedback.**

Agent tasks may **fail** and need to be **retried**.

Agents can be **triggered** or **scheduled** (chat message, or webhook, or cron job).

Therefore, Agents should be represented as **async workflows**.



Input

```
"Write a cover letter for the following job posting: Software Engineer at LastMile AI. Responsibilities include developing AI systems, collaborating with cross-functional teams, and enhancing scalability. Skills required: Python, distributed systems, and machine learning.\n\nCandidate Details: Alex Johnson, 3 years in machine learning, contributor to open-source AI projects, proficient in Python and TensorFlow. Motivated by building scalable AI systems to solve real-world problems.\n\nCompany information: Look up from the LastMile AI About page: https://lastmileai.dev/about"
```

Result

```
{
  "value": "**Alex Johnson**  \n[Your Address]  \n[City, State, Zip Code]  \n[Email Address]  \n[Phone Number]  \n[Date]  \n\n**Hiring Manager**  \nLastMile AI  \n[Company Address]  \n[City, State, Zip Code]  \n\nDear Hiring Manager,\n\nI am writing to express my enthusiasm for the Software Engineer position at LastMile AI. With three years of dedicated experience in machine learning, I have honed my skills in developing scalable AI systems, contributing significantly to open-source projects where I utilized Python and TensorFlow to create impactful solutions. Your mission to accelerate AI adoption resonates with my professional drive to transform complex problems into accessible technologies that serve real-world applications.\n\nIn my previous role at [Previous Company Name], I led a project to develop an AI-driven recommendation system that improved customer engagement by 30%. This involved
```

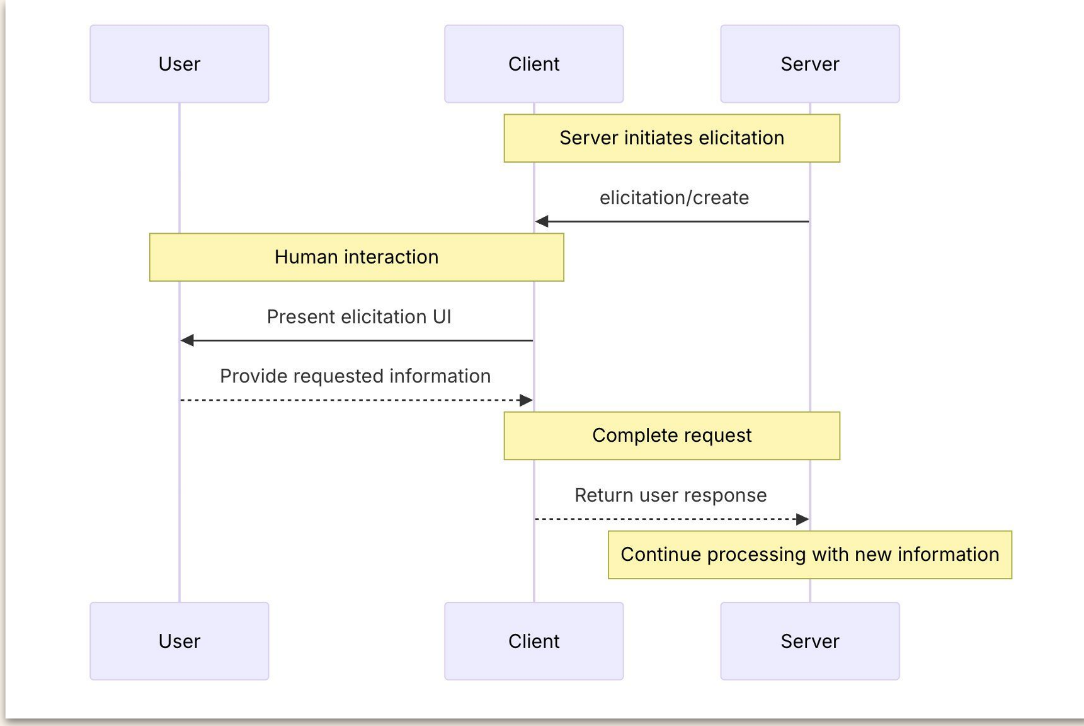
Event History

mcp_agent.agents.agent.AgentTasks.shutdown_aggregator_task
mcp_agent.workflows.llm.augmented_llm_openai.OpenAICompletionTasks.request_structured_completion_task
mcp_agent.workflows.llm.augmented_llm_openai.OpenAICompletionTasks.request_completion_task
mcp_agent.agents.agent.AgentTasks.initialize_aggregator_task
mcp_agent.agents.agent.AgentTasks.shutdown_aggregator_task
.workflows.llm.augmented_llm_openai.OpenAICompletionTasks.request_completion_task
mcp_agent.agents.agent.AgentTasks.initialize_aggregator_task
mcp_agent.agents.agent.AgentTasks.shutdown_aggregator_task
gmented_llm_openai.OpenAICompletionTasks.request_structured_completion_task
ai.OpenAICompletionTasks.request_completion_task
t.agents.agent.AgentTasks.initialize_aggregator_task
2025-04-30 UTC 02:2
2025-04-30 UTC 02:24:07.

mcp-agent uses Temporal to execute agent tasks.

Elicitation

***New*:** MCP servers can request human feedback from clients (and therefore users). This is called “elicitation”.



Emerging agent architecture

1. **Orchestrator:** Planner/task manager agent (main point of contact with human).

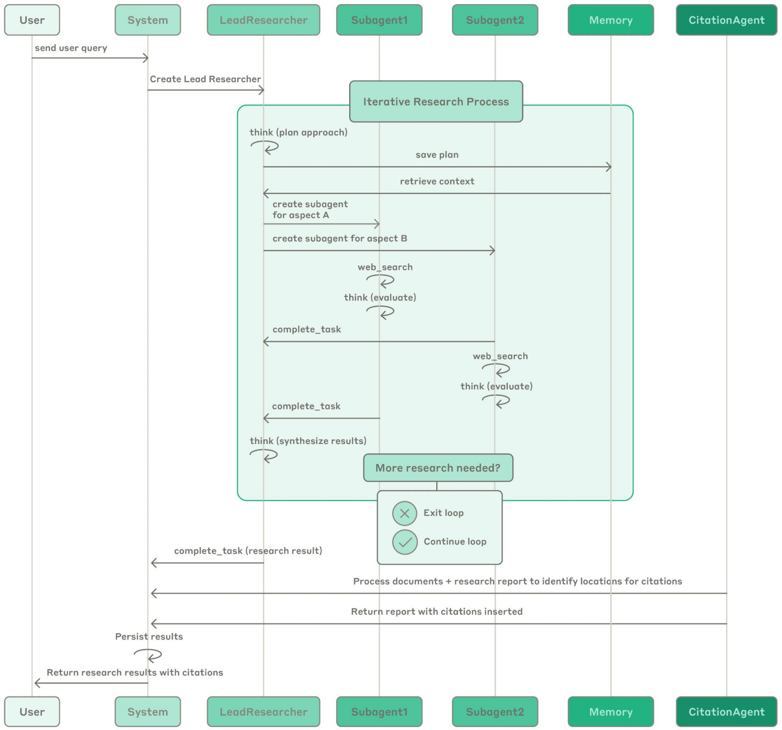
2. **Multiple sub-agents** with specialized tasks

3. **MCP access:** Agents connect to tools & data via MCP servers

4. **Durable workflows**: Long-running async tasks (jobs).

5. **Human-in-the-loop** (not full autonomy)

Multi-agent System Process Diagram



Problems yet to be solved

1. Agents as MCP servers
   a. Long-running tools
   b. Agent authentication / authorization
2. Context management
3. Tool management
4. Human in the loop for long-running workflows

We are working on solving these!

THANK YOU!

github.com/lastmile-ai/mcp-agent

@qadri_sarmad

sarmad@lastmileai.dev