

Examples

# Example Clients

 Copy page

▾

A list of applications that support MCP integrations

This page provides an overview of applications that support the Model Context Protocol (MCP). Each client may support different MCP features, allowing for varying levels of integration with MCP servers.

## Feature support matrix


Client	<u>Resources</u>	<u>Prompts</u>	<u>Tools</u>	<u>Discovery</u>	<u>Sampling</u>	<u>Roots</u>	<u>Elicitation</u>	Notes
<u>5ire</u>	✗	✗	✓	?	✗	✗	?	Supports tools.
<u>AgentAI</u>	✗	✗	✓	?	✗	✗	?	Agent Library written in Rust w tools support
<u>AgenticFlow</u>	✓	✓	✓	✓	✗	✗	?	Supports tools, prompts, and resources for no-code AI agents a multi-agent workflows.
<u>AIQL TUUI</u>	✓	✓	✓	✓	✓	✗	?	Supports tools, prompts, resource and sampling for MCP servers via Multi-LLM APIs.
<u>Amazon Q CLI</u>	✗	✓	✓	?	✗	✗	?	Supports prompt and tools.
<u>Amazon Q IDE</u>	✗	✗	✓	✗	✗	✗	?	Supports tools.
<u>Apify MCP Tester</u>	✗	✗	✓	✓	✗	✗	?	Supports remote MCP servers and tool discovery.

Client	Resources	Prompts	Tools	Discovery	Sampling	Roots	Elicitation	Notes
 <b>Model Context Protocol</b> <u>Augment Code</u>	✗	✗	✓	✗	✗	✗	?	Supports tools in local and remote agents.
>								
<u>BeeAI Framework</u>	✗	✗	✓	✗	✗	✗	?	Supports tools in agentic workflow
<u>BoltAI</u>	✗	✗	✓	?	✗	✗	?	Supports tools.
<u>ChatWise</u>	✗	✗	✓	✗	✗	✗	?	Supports tools.
<u>Claude.ai</u>	✓	✓	✓	✗	✗	✗	?	Supports tools, prompts, and resources for remote MCP servers.
<u>Claude Code</u>	✓	✓	✓	✗	✗	✓	?	Supports resource prompts, tools, a roots
<u>Claude Desktop App</u>	✓	✓	✓	✗	✗	✗	?	Supports tools, prompts, and resources for loc and remote MCP servers.
<u>Chorus</u>	✗	✗	✓	?	✗	✗	?	Supports tools.
<u>Cline</u>	✓	✗	✓	✓	✗	✗	?	Supports tools at resources.
<u>CodeGPT</u>	✗	✗	✓	?	✗	✗	?	Supports tools in VSCode and JetBrains
<u>Continue</u>	✓	✓	✓	?	✗	✗	?	Supports tools, prompts, and resources.
<u>Copilot-MCP</u>	✓	✗	✓	?	✗	✗	?	Supports tools at resources.
<u>Cursor</u>	✗	✗	✓	✗	✗	✗	?	Supports tools.
<u>Daydreams Agents</u>	✓	✓	✓	✗	✗	✗	?	Support for drop Servers to Daydreams agen
<u>Emacs Mcp</u>	✗	✗	✓	✗	✗	✗	?	Supports tools in Emacs.

Client	Resources	Prompts	Tools	Discovery	Sampling	Roots	Elicitation	Notes
 <b>Model Context Protocol</b> <u>fast-agent</u>	✓	✓	✓	✓	✓	✓	✓	Full multimodal MCP support, with end-to-end tests
>								
<u>FlowDown</u>	✗	✗	✓	?	✗	✗	✗	Supports tools.
<u>FLUJO</u>	✗	✗	✓	?	✗	✗	?	Support for resources, Prompts and Roots are coming soon
<u>Genkit</u>	⚠	✓	✓	?	✗	✗	?	Supports resource list and lookup through tools.
<u>Glama</u>	✓	✓	✓	?	✗	✗	?	Supports tools.
<u>GenAIScript</u>	✗	✗	✓	?	✗	✗	?	Supports tools.
<u>Goose</u>	✗	✗	✓	?	✗	✗	?	Supports tools.
<u>gptme</u>	✗	✗	✓	?	✗	✗	?	Supports tools.
<u>HyperAgent</u>	✗	✗	✓	?	✗	✗	?	Supports tools.
<u>JetBrains AI Assistant</u>	✗	✗	✓	✗	✗	✗	?	Supports tools for all JetBrains IDEs
<u>Kilo Code</u>	✓	✗	✓	✓	✗	✗	?	Supports tools and resources.
<u>Klavis AI</u> <u>Slack/Discord/Web</u>	✓	✗	✓	?	✗	✗	?	Supports tools and resources.
<u>LibreChat</u>	✗	✗	✓	?	✗	✗	?	Supports tools for Agents
<u>Lutra</u>	✓	✓	✓	?	✗	✗	?	Supports any MCP server for reusable playbook creation
<u>mcp-agent</u>	✓	✓	✓	?	⚠	✓	✓	Supports tools, prompts, resources, roots, server connection management, and agent workflows
<u>mcp-client-chatbot</u>	✗	✗	✓	✗	✗	✗	?	Supports tools.
<u>mcp-use</u>	✓	✓	✓	?	✗	✗	?	Support tools, resources, studio

Client	Resources	Prompts	Tools	Discovery	Sampling	Roots	Elicitation	Notes	
 Model Context Protocol								http connection, local llms-agents	
<u>MCPHub</u>	>	✓	✓	✓	?	✗	✗	?	Supports tools, resources, and prompts in Neovim
<u>MCPOmni-Connect</u>		✓	✓	✓	?	✓	✗	?	Supports tools w agentic mode, ReAct, and orchestrator capabilities.
<u>Memex</u>		✓	✓	✓	?	✗	✗	?	Support tools. AI support <i>building and testing</i> MCP server, all-in-one desktop app.
<u>Microsoft Copilot Studio</u>		✗	✗	✓	?	✗	✗	?	Supports tools
<u>MindPal</u>		✗	✗	✓	?	✗	✗	?	Supports tools for no-code AI agent and multi-agent workflows.
<u>MooPoint</u>		✗	✗	✓	?	✓	✗	?	Web-Hosted client with tool calling support
<u>Msty Studio</u>		✗	✗	✓	?	✗	✗	?	Supports tools
<u>NVIDIA Agent Intelligence toolkit</u>		✗	✗	✓	?	✗	✗	?	Supports tools in agentic workflow
<u>OpenSumi</u>		✗	✗	✓	?	✗	✗	?	Supports tools in OpenSumi
<u>oterm</u>		✗	✓	✓	?	✓	✗	?	Supports tools, prompts and sampling for Ollama.
<u>Postman</u>		✓	✓	✓	?	✗	✗	?	Supports tools, resources, prompts and sampling
<u>RecurseChat</u>		✗	✗	✓	?	✗	✗	?	Supports tools.
<u>Roo Code</u>		✓	✗	✓	?	✗	✗	?	Supports tools at resources.

Client	Resources	Prompts	Tools	Discovery	Sampling	Roots	Elicitation	Notes
 <u>Model Context Protocol Shortwave</u>	✗	✗	✓	?	✗	✗	?	Supports tools.
<u>Slack MCP Client</u>	✗	✗	✓	?	✗	✗	?	Supports tools at multiple servers.
<u>Sourcegraph Cody</u>	✓	✗	✗	?	✗	✗	?	Supports resource discovery through OpenCTI
<u>SpinAI</u>	✗	✗	✓	?	✗	✗	?	Supports tools for Typescript AI Agents
<u>Superinterface</u>	✗	✗	✓	?	✗	✗	?	Supports tools
<u>Superjoin</u>	✗	✗	✓	?	✗	✗	?	Supports tools at multiple servers.
<u>systemprompt</u>	✓	✓	✓	?	✓	✗	?	Supports tools, resources, prompts and sampling.
<u>Tambo</u>	✗	✗	✓	?	✗	✗	?	Supports tools at multiple servers, with authentication
<u>Tencent CloudBase AI DevKit</u>	✗	✗	✓	?	✗	✗	?	Supports tools
<u>TheiaAI/TheiaIDE</u>	✗	✗	✓	?	✗	✗	?	Supports tools for Agents in Theia / and the AI-powered Theia IDE
<u>Tome</u>	✗	✗	✓	?	✗	✗	?	Supports tools, manages MCP servers.
<u>TypingMind App</u>	✗	✗	✓	?	✗	✗	?	Supports tools at app-level (appears as plugins) or when assigned to Agent
<u>VS Code GitHub Copilot</u>	✓	✓	✓	✓	✓	✓	✓	Supports dynamic tool/roots discovery, secure secret configuration, an explicit tool prompting
<u>Warp</u>	✓	✗	✓	✓	✗	✗	?	Supports tools, resources, and

Client	Resources	Prompts	Tools	Discovery	Sampling	Roots	Elicitation	Notes
 Model Context Protocol								most of the discovery criteria
<u>WhatsMCP</u> >	✗	✗	✓	✗	✗	✗	?	Supports tools for Remote MCP Servers in WhatsApp
<u>Windsurf Editor</u>	✗	✗	✓	✓	✗	✗	?	Supports tools with AI Flow for collaborative development.
<u>Witsy</u>	✗	✗	✓	?	✗	✗	?	Supports tools in Witsy.
<u>Zed</u>	✗	✓	✗	✗	✗	✗	?	Prompts appear as slash commands
<u>Zencoder</u>	✗	✗	✓	✗	✗	✗	?	Supports tools

## Client details

### 5ire

5ire is an open source cross-platform desktop AI assistant that supports tools through MCP servers.

**Key features:**

- Built-in MCP servers can be quickly enabled and disabled.
- Users can add more servers by modifying the configuration file.
- It is open-source and user-friendly, suitable for beginners.
- Future support for MCP will be continuously improved.

### AgentAI

AgentAI is a Rust library designed to simplify the creation of AI agents. The library includes seamless integration with MCP Servers.

Example of MCP Server integration

## Key features:

### Model Context Protocol

Multi-LLM – We support most LLM APIs (OpenAI, Anthropic, Gemini, Ollama, and all OpenAI API Compatible).

Built-in support for MCP Servers.

Create agentic flows in a type- and memory-safe language like Rust.

## AgenticFlow

**AgenticFlow** is a no-code AI platform that helps you build agents that handle sales, marketing, and creative tasks around the clock. Connect 2,500+ APIs and 10,000+ tools securely via MCP.

### Key features:

No-code AI agent creation and workflow building.

Access a vast library of 10,000+ tools and 2,500+ APIs through MCP.

Simple 3-step process to connect MCP servers.

Securely manage connections and revoke access anytime.

### Learn more:

#### **AgenticFlow MCP Integration**

## AIQL TUUI

**AIQL TUUI** is a native, cross-platform desktop AI chat application with MCP support. It supports multiple AI providers (e.g., Anthropic, Cloudflare, Deepseek, OpenAI, Qwen), local AI models (via vLLM, Ray, etc.), and aggregated API platforms (such as Deepinfra, Openrouter, and more).

### Key features:

**Dynamic LLM API & Agent Switching:** Seamlessly toggle between different LLM APIs and agents on the fly.

**Comprehensive Capabilities Support:** Built-in support for tools, prompts, resources, and sampling methods.

**Configurable Agents:** Enhanced flexibility with selectable and customizable tools via agent settings.



**Advanced Sampling Control:** Modify sampling parameters and leverage multi-round sampling for Model Context Protocol optimal results.

**Cross-Platform Compatibility:** Fully compatible with macOS, Windows, and Linux.

**Free & Open-Source (FOSS):** Permissive licensing allows modifications and custom app bundling.

Learn more:

[TUUI document](#)

[AIQL GitHub repository](#)

## Amazon Q CLI

[Amazon Q CLI](#) is an open-source, agentic coding assistant for terminals.

**Key features:**

Full support for MCP servers.

Edit prompts using your preferred text editor.

Access saved prompts instantly with `@`.

Control and organize AWS resources directly from your terminal.

Tools, profiles, context management, auto-compact, and so much more!

**Get Started**

```
brew install amazon-q
```



## Amazon Q IDE

[Amazon Q IDE](#) is an open-source, agentic coding assistant for IDEs.

**Key features:**

Support for the VSCode, JetBrains, Visual Studio, and Eclipse IDEs.

Control and organize AWS resources directly from your IDE.

Manage permissions for each MCP tool via the IDE user interface.



**Apify MCP Tester** is an open-source client that connects to any MCP server using Server-Sent Events (SSE). It is a standalone Apify Actor designed for testing MCP servers over SSE, with support for Authorization headers. It uses plain JavaScript (old-school style) and is hosted on Apify, allowing you to run it without any setup.

**Key features:**

Connects to any MCP server via SSE.

Works with the **Apify MCP Server** to interact with one or more Apify **Actors**.

Dynamically utilizes tools based on context and user queries (if supported by the server).

## Augment Code

**Augment Code** is an AI-powered coding platform for VS Code and JetBrains with autonomous agents, chat, and completions. Both local and remote agents are backed by full codebase awareness and native support for MCP, enabling enhanced context through external sources and tools.

**Key features:**

Full MCP support in local and remote agents.

Add additional context through MCP servers.

Automate your development workflows with MCP tools.

Works in VS Code and JetBrains IDEs.

## BeeAI Framework

**BeeAI Framework** is an open-source framework for building, deploying, and serving powerful agentic workflows at scale. The framework includes the **MCP Tool**, a native feature that simplifies the integration of MCP servers into agentic workflows.

**Key features:**

Seamlessly incorporate MCP tools into agentic workflows.

Quickly instantiate framework-native tools from connected MCP client(s).

Planned future support for agentic MCP capabilities.

Learn more:  
 Model Context Protocol

## Example of using MCP tools in agentic workflow

>

## BoltAI

**BoltAI** is a native, all-in-one AI chat client with MCP support. BoltAI supports multiple AI providers (OpenAI, Anthropic, Google AI...), including local AI models (via Ollama, LM Studio or LMX)

### Key features:

MCP Tool integrations: once configured, user can enable individual MCP server in each chat

MCP quick setup: import configuration from Claude Desktop app or Cursor editor

Invoke MCP tools inside any app with AI Command feature

Integrate with remote MCP servers in the mobile app

### Learn more:

[BoltAI docs](#)

[BoltAI website](#)

## ChatWise

ChatWise is a desktop-optimized, high-performance chat application that lets you bring your own API keys. It supports a wide range of LLMs and integrates with MCP to enable tool workflows.

### Key features:

Tools support for MCP servers

Offer built-in tools like web search, artifacts and image generation.

## Claude Code

Claude Code is an interactive agentic coding tool from Anthropic that helps you code faster through natural language commands. It supports MCP integration for resources, prompts, tools, and roots, and also functions as an MCP server to integrate with other clients.

### Key features:



Full support for resources, prompts, tools, and roots from MCP servers

### **Model Context Protocol**

Offers its own tools through an MCP server for integrating with other MCP clients

>

## **Claude.ai**

**Claude.ai** is Anthropic's web-based AI assistant that provides MCP support for remote servers.

### **Key features:**

- Support for remote MCP servers via integrations UI in settings
- Access to tools, prompts, and resources from configured MCP servers
- Seamless integration with Claude's conversational interface
- Enterprise-grade security and compliance features

## **Claude Desktop App**

The Claude desktop application provides comprehensive support for MCP, enabling deep integration with local tools and data sources.

### **Key features:**


- Full support for resources, allowing attachment of local files and data
- Support for prompt templates
- Tool integration for executing commands and scripts
- Local server connections for enhanced privacy and security

## **Chorus**

**Chorus** is a native Mac app for chatting with AIs. Chat with multiple models at once, run tools and MCPs, create projects, quick chat, bring your own key, all in a blazing fast, keyboard shortcut friendly app.

### **Key features:**

- MCP support with one-click install
- Built in tools, like web search, terminal, and image generation

 Chat with multiple models at once (cloud or local)  
**Model Context Protocol**  
Create projects with scoped memory

Quick chat with an AI that can see your screen

## Cline

**Cline** is an autonomous coding agent in VS Code that edits files, runs commands, uses a browser, and more—with your permission at each step.

### Key features:

Create and add tools through natural language (e.g. “add a tool that searches the web”)

Share custom MCP servers Cline creates with others via the `~/Documents/Cline/MCP` directory

Displays configured MCP servers along with their tools, resources, and any error logs

## CodeGPT

**CodeGPT** is a popular VS Code and JetBrains extension that brings AI-powered coding assistance to your editor. It supports integration with MCP servers for tools, allowing users to leverage external AI capabilities directly within their development workflow.

### Key features:

Use MCP tools from any configured MCP server

Seamless integration with VS Code and JetBrains UI

Supports multiple LLM providers and custom endpoints

### Learn more:

[CodeGPT Documentation](#)

## Continue

**Continue** is an open-source AI code assistant, with built-in support for all MCP features.

### Key features:



Type "@" to mention MCP resources

### Model Context Protocol

Prompt templates surface as slash commands

Use both built-in and MCP tools directly in chat

Supports VS Code and JetBrains IDEs, with any LLM

## Copilot-MCP

Copilot-MCP enables AI coding assistance via MCP.

### Key features:

Support for MCP tools and resources

Integration with development workflows

Extensible AI capabilities

## Cursor

Cursor is an AI code editor.

### Key features:

Support for MCP tools in Cursor Composer

Support for both STDIO and SSE

## Daydreams


Daydreams is a generative agent framework for executing anything onchain

### Key features:

Supports MCP Servers in config

Exposes MCP Client

## Emacs Mcp

**Emacs Mcp** is an Emacs client designed to interface with MCP servers, enabling seamless  **Model Context Protocol** connections and interactions. It provides MCP tool invocation support for AI plugins like [gptel](#) and [llm](#), adhering to Emacs' standard tool invocation format. This integration enhances the functionality of AI tools within the Emacs ecosystem.

**Key features:**

Provides MCP tool support for Emacs.

## fast-agent

**fast-agent** is a Python Agent framework, with simple declarative support for creating Agents and Workflows, with full multi-modal support for Anthropic and OpenAI models.

**Key features:**

PDF and Image support, based on MCP Native types

Interactive front-end to develop and diagnose Agent applications, including passthrough and playback simulators

Built in support for "Building Effective Agents" workflows.

Deploy Agents as MCP Servers

## FlowDown

**FlowDown** is a blazing fast and smooth client app for using AI/LLM, with a strong emphasis on privacy and user experience. It supports MCP servers to extend its capabilities with external tools, allowing users to build powerful, customized workflows.

**Key features:**

**Seamless MCP Integration:** Easily connect to MCP servers to utilize a wide range of external tools.

**Privacy-First Design:** Your data stays on your device. We don't collect any user data, ensuring complete privacy.

**Lightweight & Efficient:** A compact and optimized design ensures a smooth and responsive experience with any AI model.



**Broad Compatibility:** Works with all OpenAI-compatible service providers and supports local Model Context Protocol offline models through MLX.

**Rich User Experience:** Features beautifully formatted Markdown, blazing-fast text rendering, and intelligent, automated chat titling.

Learn more:

[FlowDown website](#)

[FlowDown documentation](#)

## FLUJO

Think n8n + ChatGPT. FLUJO is an desktop application that integrates with MCP to provide a workflow-builder interface for AI interactions. Built with Next.js and React, it supports both online and offline (ollama) models, it manages API Keys and environment variables centrally and can install MCP Servers from GitHub. FLUJO has an ChatCompletions endpoint and flows can be executed from other AI applications like Cline, Roo or Claude.

**Key features:**

Environment & API Key Management

Model Management

MCP Server Integration

Workflow Orchestration

Chat Interface

## Genkit

**Genkit** is a cross-language SDK for building and integrating GenAI features into applications. The **genkitx-mcp** plugin enables consuming MCP servers as a client or creating MCP servers from Genkit tools and prompts.

**Key features:**

Client support for tools and prompts (resources partially supported)

Rich discovery with support in Genkit's Dev UI playground



Seamless interoperability with Genkit's existing tools and prompts

### **Model Context Protocol**

Works across a wide variety of GenAI models from top providers

>

## **Glama**

**Glama** is a comprehensive AI workspace and integration platform that offers a unified interface to leading LLM providers, including OpenAI, Anthropic, and others. It supports the Model Context Protocol (MCP) ecosystem, enabling developers and enterprises to easily discover, build, and manage MCP servers.

### **Key features:**

Integrated **MCP Server Directory**

Integrated **MCP Tool Directory**

Host MCP servers and access them via the Chat or SSE endpoints – Ability to chat with multiple LLMs and MCP servers at once

Upload and analyze local files and data

Full-text search across all your chats and data

## **GenAIScript**

Programmatically assemble prompts for LLMs using **GenAIScript** (in JavaScript). Orchestrate LLMs, tools, and data in JavaScript.

### **Key features:**

JavaScript toolbox to work with prompts

Abstraction to make it easy and productive

Seamless Visual Studio Code integration

## **Goose**

**Goose** is an open source AI agent that supercharges your software development by automating coding tasks.

### **Key features:**



Expose MCP functionality to Goose through tools.



### Model Context Protocol

MCPs can be installed directly via the [extensions directory](#), CLI, or UI.

Goose allows you to extend its functionality by [building your own MCP servers](#).

Includes built-in tools for development, web scraping, automation, memory, and integrations with JetBrains and Google Drive.

## gptme

[gptme](#) is a open-source terminal-based personal AI assistant/agent, designed to assist with programming tasks and general knowledge work.

### Key features:

CLI-first design with a focus on simplicity and ease of use

Rich set of built-in tools for shell commands, Python execution, file operations, and web browsing

Local-first approach with support for multiple LLM providers

Open-source, built to be extensible and easy to modify

## HyperAgent

[HyperAgent](#) is Playwright supercharged with AI. With HyperAgent, you no longer need brittle scripts, just powerful natural language commands. Using MCP servers, you can extend the capability of HyperAgent, without having to write any code.

### Key features:

AI Commands: Simple APIs like `page.ai()`, `page.extract()` and `executeTask()` for any AI automation

Fallback to Regular Playwright: Use regular Playwright when AI isn't needed

Stealth Mode – Avoid detection with built-in anti-bot patches

Cloud Ready – Instantly scale to hundreds of sessions via [Hyperbrowser](#)

MCP Client – Connect to tools like Composio for full workflows (e.g. writing web data to Google Sheets)

## JetBrains AI Assistant

## Key features: ›

- Unlimited code completion powered by Mellum, JetBrains' proprietary AI model.
- Context-aware AI chat that understands your code and helps you in real time.
- Access to top-tier models from OpenAI, Anthropic, and Google.
- Offline mode with connected local LLMs via Ollama or LM Studio.
- Deep integration into IDE workflows, including code suggestions in the editor, VCS assistance, runtime error explanation, and more.

## Kilo Code

**Kilo Code** is an autonomous coding AI dev team in VS Code that edits files, runs commands, uses a browser, and more.

## Key features:

- Create and add tools through natural language (e.g. "add a tool that searches the web")
- Discover MCP servers via the MCP Marketplace
- One click MCP server installs via MCP Marketplace
- Displays configured MCP servers along with their tools, resources, and any error logs

## Klavis AI Slack/Discord/Web

**Klavis AI** is an Open-Source Infra to Use, Build & Scale MCPs with ease.

## Key features:

- Slack/Discord/Web MCP clients for using MCPs directly
- Simple web UI dashboard for easy MCP configuration
- Direct OAuth integration with Slack & Discord Clients and MCP Servers for secure user authentication
- SSE transport support
- Open-source infrastructure ([GitHub repository](#))

Learn more:  
 Model Context Protocol

[Demo video showing MCP usage in Slack/Discord](#)

>

## LibreChat

**LibreChat** is an open-source, customizable AI chat UI that supports multiple AI providers, now including MCP integration.

### Key features:

Extend current tool ecosystem, including **Code Interpreter** and Image generation tools, through MCP servers

Add tools to customizable **Agents**, using a variety of LLMs from top providers

Open-source and self-hostable, with secure multi-user support

Future roadmap includes expanded MCP feature support

## Lutra

**Lutra** is an AI agent that transforms conversations into actionable, automated workflows.

### Key features:

Easy MCP Integration: Connecting Lutra to MCP servers is as simple as providing the server URL; Lutra handles the rest behind the scenes.

Chat to Take Action: Lutra understands your conversational context and goals, automatically integrating with your existing apps to perform tasks.

Reusable Playbooks: After completing a task, save the steps as reusable, automated workflows—simplifying repeatable processes and reducing manual effort.

Shareable Automations: Easily share your saved playbooks with teammates to standardize best practices and accelerate collaborative workflows.

### Learn more:

[Lutra AI agent explained](#)

**mcp-agent** is a simple, composable framework to build agents using Model Context Protocol.



### Key features:

Automatic connection management of MCP servers.

Expose tools from multiple servers to an LLM.

Implements every pattern defined in [\*\*Building Effective Agents\*\*](#).

Supports workflow pause/resume signals, such as waiting for human feedback.

## mcp-client-chatbot

**mcp-client-chatbot** is a local-first chatbot built with Vercel's Next.js, AI SDK, and Shadcn UI.

### Key features:

It supports standard MCP tool calling and includes both a custom MCP server and a standalone UI for testing MCP tools outside the chat flow.

All MCP tools are provided to the LLM by default, but the project also includes an optional `@toolname` mention feature to make tool invocation more explicit—particularly useful when connecting to multiple MCP servers with many tools.

Visual workflow builder that lets you create custom tools by chaining LLM nodes and MCP tools together. Published workflows become callable as `@workflow_name` tools in chat, enabling complex multi-step automation sequences.

## mcp-use

**mcp-use** is an open source python library to very easily connect any LLM to any MCP server both locally and remotely.

### Key features:

Very simple interface to connect any LLM to any MCP.

Support the creation of custom agents, workflows.

Supports connection to multiple MCP servers simultaneously.

Supports all langchain supported models, also locally.

Offers efficient tool orchestration and search functionalities.

**MCPHub** is a powerful Neovim plugin that integrates MCP (Model Context Protocol) servers into your workflow.

### Key features:

Install, configure and manage MCP servers with an intuitive UI.

Built-in Neovim MCP server with support for file operations (read, write, search, replace), command execution, terminal integration, LSP integration, buffers, and diagnostics.

Create Lua-based MCP servers directly in Neovim.

Integrates with popular Neovim chat plugins Avante.nvim and CodeCompanion.nvim

## MCP Omni-Connect

**MCP Omni-Connect** is a versatile command-line interface (CLI) client designed to connect to various Model Context Protocol (MCP) servers using both stdio and SSE transport.

### Key features:

Support for resources, prompts, tools, and sampling

Agentic mode with ReAct and orchestrator capabilities

Seamless integration with OpenAI models and other LLMs

Dynamic tool and resource management across multiple servers

Support for both stdio and SSE transport protocols

Comprehensive tool orchestration and resource analysis capabilities

## Memex

**Memex** is the first MCP client and MCP server builder - all-in-one desktop app. Unlike traditional MCP clients that only consume existing servers, Memex can create custom MCP servers from natural language prompts, immediately integrate them into its toolkit, and use them to solve problems—all within a single conversation.

### Key features:

**Prompt-to-MCP Server:** Generate fully functional MCP servers from natural language descriptions



**Self-Testing & Debugging:** Autonomously test, debug, and improve created MCP servers

**Model Context Protocol**

**Universal MCP Client:** Works with any MCP server through intuitive, natural language integration

**Curated MCP Directory:** Access to tested, one-click installable MCP servers (Neon, Netlify, GitHub, Context7, and more)

**Multi-Server Orchestration:** Leverage multiple MCP servers simultaneously for complex workflows

**Learn more:**

**[Memex Launch 2: MCP Teams and Agent API](#)**

## Microsoft Copilot Studio

**Microsoft Copilot Studio** is a robust SaaS platform designed for building custom AI-driven applications and intelligent agents, empowering developers to create, deploy, and manage sophisticated AI solutions.

**Key features:**

Support for MCP tools

Extend Copilot Studio agents with MCP servers

Leveraging Microsoft unified, governed, and secure API management solutions

## MindPal

**MindPal** is a no-code platform for building and running AI agents and multi-agent workflows for business processes.

**Key features:**

Build custom AI agents with no-code

Connect any SSE MCP server to extend agent tools

Create multi-agent workflows for complex business processes

User-friendly for both technical and non-technical professionals

Ongoing development with continuous improvement of MCP support

## MooPoint

### MooPoint

MooPoint is a web-based AI chat platform built for developers and advanced users, letting you interact with multiple large language models (LLMs) through a single, unified interface. Connect your own API keys (OpenAI, Anthropic, and more) and securely manage custom MCP server integrations.

#### Key features:

Accessible from any PC or smartphone—no installation required

Choose your preferred LLM provider

Supports `SSE` , `Streamable HTTP` , `npx` , and `uvx` MCP servers

OAuth and sampling support

New features added daily

## Msty Studio

**Msty Studio** is a privacy-first AI productivity platform that seamlessly integrates local and online language models (LLMs) into customizable workflows. Designed for both technical and non-technical users, Msty Studio offers a suite of tools to enhance AI interactions, automate tasks, and maintain full control over data and model behavior.

#### Key features:

**Toolbox & Toolsets:** Connect AI models to local tools and scripts using MCP-compliant configurations. Group tools into Toolsets to enable dynamic, multi-step workflows within conversations.

**Turnstiles:** Create automated, multi-step AI interactions, allowing for complex data processing and decision-making flows.

**Real-Time Data Integration:** Enhance AI responses with up-to-date information by integrating real-time web search capabilities.



**Split Chats & Branching:** Engage in parallel conversations with multiple models simultaneously, enabling comparative analysis and diverse perspectives.

Learn more: [>](#)

## Msty Studio Documentation

## NVIDIA Agent Intelligence (AIQ) toolkit

**NVIDIA Agent Intelligence (AIQ) toolkit** is a flexible, lightweight, and unifying library that allows you to easily connect existing enterprise agents to data sources and tools across any framework.

### Key features:

Acts as an MCP **client** to consume remote tools

Acts as an MCP **server** to expose tools

Framework agnostic and compatible with LangChain, CrewAI, Semantic Kernel, and custom agents

Includes built-in observability and evaluation tools

Learn more:

[AIQ toolkit GitHub repository](#)

[AIQ toolkit MCP documentation](#)

## OpenSumi

**OpenSumi** is a framework helps you quickly build AI Native IDE products.

### Key features:

Supports MCP tools in OpenSumi

Supports built-in IDE MCP servers and custom MCP servers

## oterm

**oterm** is a terminal client for Ollama allowing users to create chats/agents.



## Key features:

### Model Context Protocol

Support for multiple fully customizable chat sessions with Ollama connected with tools.

Support for MCP tools.

## Roo Code

Roo Code enables AI coding assistance via MCP.

### Key features:

Support for MCP tools and resources

Integration with development workflows

Extensible AI capabilities

## Postman

Postman is the most popular API client and now supports MCP server testing and debugging.

### Key features:

Full support of all major MCP features (tools, prompts, resources, and subscriptions)

Fast, seamless UI for debugging MCP capabilities

MCP config integration (Claude, VSCode, etc.) for fast first-time experience in testing MCPs

Integration with history, variables, and collections for reuse and collaboration

## RecurseChat

RecurseChat is a powerful, fast, local-first chat client with MCP support. RecurseChat supports multiple AI providers including LLaMA.cpp, Ollama, and OpenAI, Anthropic.

### Key features:

Local AI: Support MCP with Ollama models.

MCP Tools: Individual MCP server management. Easily visualize the connection states of MCP servers.

MCP Import: Import configuration from Claude Desktop app or JSON

## Shortwave

**Shortwave** is an AI-powered email client that supports MCP tools to enhance email productivity and workflow automation.

### Key features:

MCP tool integration for enhanced email workflows

Rich UI for adding, managing and interacting with a wide range of MCP servers

Support for both remote (Streamable HTTP and SSE) and local (Stdio) MCP servers

AI assistance for managing your emails, calendar, tasks and other third-party services

## Slack MCP Client

**Slack MCP Client** acts as a bridge between Slack and Model Context Protocol (MCP) servers. Using Slack as the interface, it enables large language models (LLMs) to connect and interact with various MCP servers through standardized MCP tools.

### Key features:

**Supports Popular LLM Providers:** Integrates seamlessly with leading large language model providers such as OpenAI, Anthropic, and Ollama, allowing users to leverage advanced conversational AI and orchestration capabilities within Slack.

**Dynamic and Secure Integration:** Supports dynamic registration of MCP tools, works in both channels and direct messages and manages credentials securely via environment variables or Kubernetes secrets.

**Easy Deployment and Extensibility:** Offers official Docker images, a Helm chart for Kubernetes, and Docker Compose for local development, making it simple to deploy, configure, and extend with additional MCP servers or tools.

## Sourcegraph Cody

**Cody** is Sourcegraph's AI coding assistant, which implements MCP through OpenCTX.

## Key features:

### Model Context Protocol

- Support for MCP resources
- Integration with Sourcegraph's code intelligence
- Uses OpenCTX as an abstraction layer
- Future support planned for additional MCP features

## SpinAI

**SpinAI** is an open-source TypeScript framework for building observable AI agents. The framework provides native MCP compatibility, allowing agents to seamlessly integrate with MCP servers and tools.

### Key features:

- Built-in MCP compatibility for AI agents
- Open-source TypeScript framework
- Observable agent architecture
- Native support for MCP tools integration

## Superinterface

**Superinterface** is AI infrastructure and a developer platform to build in-app AI assistants with support for MCP, interactive components, client-side function calling and more.

### Key features:

- Use tools from MCP servers in assistants embedded via React components or script tags
- SSE transport support
- Use any AI model from any AI provider (OpenAI, Anthropic, Ollama, others)

## Superjoin

**Superjoin** brings the power of MCP directly into Google Sheets extension. With Superjoin, users can access and invoke MCP tools and agents without leaving their spreadsheets, enabling powerful AI workflows and automation right where their data lives.

## Key features:

### Model Context Protocol

Native Google Sheets add-on providing effortless access to MCP capabilities

Supports OAuth 2.1 and header-based authentication for secure and flexible connections

Compatible with both SSE and Streamable HTTP transport for efficient, real-time streaming communication

Fully web-based, cross-platform client requiring no additional software installation

## systemprompt

**systemprompt** is a voice-controlled mobile app that manages your MCP servers. Securely leverage MCP agents from your pocket. Available on iOS and Android.

### Key features:

**Native Mobile Experience:** Access and manage your MCP servers anytime, anywhere on both Android and iOS devices

**Advanced AI-Powered Voice Recognition:** Sophisticated voice recognition engine enhanced with cutting-edge AI and Natural Language Processing (NLP), specifically tuned to understand complex developer terminology and command structures

**Unified Multi-MCP Server Management:** Effortlessly manage and interact with multiple Model Context Protocol (MCP) servers from a single, centralized mobile application

## Tambo

**Tambo** is a platform for building custom chat experiences in React, with integrated custom user interface components.

### Key features:

Hosted platform with React SDK for integrating chat or other LLM-based experiences into your own app.

Support for selection of arbitrary React components in the chat experience, with state management and tool calling.

Support for MCP servers, from Tambo's servers or directly from the browser.

Supports OAuth 2.1 and custom header-based authentication.



## Tencent CloudBase AI DevKit

**Tencent CloudBase AI DevKit** is a tool for building AI agents in minutes, featuring zero-code tools, secure data integration, and extensible plugins via MCP.

### Key features:

Support for MCP tools

Extend agents with MCP servers

MCP servers hosting: serverless hosting and authentication support

## TheiaAI/TheiaIDE

**Theia AI** is a framework for building AI-enhanced tools and IDEs. The **AI-powered Theia IDE** is an open and flexible development environment built on Theia AI.

### Key features:

**Tool Integration:** Theia AI enables AI agents, including those in the Theia IDE, to utilize MCP servers for seamless tool interaction.

**Customizable Prompts:** The Theia IDE allows users to define and adapt prompts, dynamically integrating MCP servers for tailored workflows.

**Custom agents:** The Theia IDE supports creating custom agents that leverage MCP capabilities, enabling users to design dedicated workflows on the fly.

Theia AI and Theia IDE's MCP integration provide users with flexibility, making them powerful platforms for exploring and adapting MCP.

### Learn more:

**[Theia IDE and Theia AI MCP Announcement](#)**

**[Download the AI-powered Theia IDE](#)**

**Tome** is an open source cross-platform desktop app designed for working with local LLMs and MCP servers. It is designed to be beginner friendly and abstract away the nitty gritty of configuration for people getting started with MCP.

### Key features:

- MCP servers are managed by Tome so there is no need to install uv or npm or configure JSON
- Users can quickly add or remove MCP servers via UI
- Any tool-supported local model on Ollama is compatible

## TypingMind App

**TypingMind** is an advanced frontend for LLMs with MCP support. TypingMind supports all popular LLM providers like OpenAI, Gemini, Claude, and users can use with their own API keys.

### Key features:

- **MCP Tool Integration:** Once MCP is configured, MCP tools will show up as plugins that can be enabled/disabled easily via the main app interface.
- **Assign MCP Tools to Agents:** TypingMind allows users to create AI agents that have a set of MCP servers assigned.
- **Remote MCP servers:** Allows users to customize where to run the MCP servers via its MCP Connector configuration, allowing the use of MCP tools across multiple devices (laptop, mobile devices, etc.) or control MCP servers from a remote private server.

### Learn more:

[TypingMind MCP Document](#)

[Download TypingMind \(PWA\)](#)

## VS Code GitHub Copilot

**VS Code** integrates MCP with GitHub Copilot through **agent mode**, allowing direct interaction with MCP-provided tools within your agentic coding workflow. Configure servers in Claude Desktop, workspace or user settings, with guided MCP installation and secure handling of keys in input variables to avoid leaking hard-coded keys.

### Key features:



Support for stdio and server-sent events (SSE) transport

**Model Context Protocol**

Per-session selection of tools per agent session for optimal performance

Easy server debugging with restart commands and output logging

Tool calls with editable inputs and always-allow toggle

Integration with existing VS Code extension system to register MCP servers from extensions

## Warp

**Warp** is the intelligent terminal with AI and your dev team's knowledge built-in. With natural language capabilities integrated directly into an agentic command line, Warp enables developers to code, automate, and collaborate more efficiently — all within a terminal that features a modern UX.

### Key features:

**Agent Mode with MCP support:** invoke tools and access data from MCP servers using natural language prompts

**Flexible server management:** add and manage CLI or SSE-based MCP servers via Warp's built-in UI

**Live tool/resource discovery:** view tools and resources from each running MCP server

**Configurable startup:** set MCP servers to start automatically with Warp or launch them manually as needed

## WhatsMCP

**WhatsMCP** is an MCP client for WhatsApp. WhatsMCP lets you interact with your AI stack from the comfort of a WhatsApp chat.

### Key features:

Supports MCP tools

SSE transport, full OAuth2 support

Chat flow management for WhatsApp messages

One click setup for connecting to your MCP servers

In chat management of MCP servers

OAuth flow natively supported in WhatsApp

## Windsurf Editor

 Model Context Protocol

**Windsurf Editor** is an agentic IDE that combines AI assistance with developer workflows. It features an innovative AI Flow system that enables both collaborative and independent AI interactions while maintaining developer control.

### Key features:

- Revolutionary AI Flow paradigm for human-AI collaboration
- Intelligent code generation and understanding
- Rich development tools with multi-model support

## Witsy

**Witsy** is an AI desktop assistant, supporting Anthropic models and MCP servers as LLM tools.

### Key features:

- Multiple MCP servers support
- Tool integration for executing commands and scripts
- Local server connections for enhanced privacy and security
- Easy-install from Smithery.ai
- Open-source, available for macOS, Windows and Linux

## Zed

**Zed** is a high-performance code editor with built-in MCP support, focusing on prompt templates and tool integration.

### Key features:

- Prompt templates surface as slash commands in the editor
- Tool integration for enhanced coding workflows
- Tight integration with editor features and workspace context
- Does not support MCP resources



**Zencoder** is a coding agent that's available as an extension for VS Code and JetBrains family of IDEs, meeting developers where they already work. It comes with RepoGrokking (deep contextual codebase understanding), agentic pipeline, and the ability to create and share custom agents.

**Key features:**

RepoGrokking - deep contextual understanding of codebases

Agentic pipeline - runs, tests, and executes code before outputting it

Zen Agents platform - ability to build and create custom agents and share with the team

Integrated MCP tool library with one-click installations

Specialized agents for Unit and E2E Testing

**Learn more:**

[\*\*Zencoder Documentation\*\*](#)

## **Adding MCP support to your application**

If you've added MCP support to your application, we encourage you to submit a pull request to add it to this list. MCP integration can provide your users with powerful contextual AI capabilities and make your application part of the growing MCP ecosystem.

**Benefits of adding MCP support:**

Enable users to bring their own context and tools

Join a growing ecosystem of interoperable AI applications

Provide users with flexible integration options

Support local-first AI workflows

To get started with implementing MCP in your application, check out our [\*\*Python\*\*](#) or [\*\*TypeScript SDK Documentation\*\*](#)

## **Updates and corrections**

This list is maintained by the community. If you notice any inaccuracies or would like to update information about MCP support in your application, please submit a pull request or [open an issue in our documentation repository](#).

Was this page helpful?



Yes



No

[◀ Example Servers](#)

[Building MCP with LLMs ▶](#)