

## MODEL CONTEXT PROTOCOL (MCP): CONNECTING AGENTS AND TOOLS

**L**

tools.

LMs, like humans, become much more powerful when given tools. MCP provides a standard way to give models access to

### **What is MCP**

In November 2024, a small team at Anthropic proposed MCP as a protocol to solve a real problem: every AI provider and tool author had their own way of defining and calling tools.

You can think about MCP like a USB-C port for AI applications.

It's an open protocol for connecting AI agents to tools, models, and each other. Think of it as a universal adapter: if your tool or agent "speaks"

MCP, it can plug into any other MCP-compatible *Principles of Building AI Agents*  
43

system—no matter who built it or what language it's written in.

But as any experienced engineer knows, the power of any protocol is in the network of people following it.

While initially well-received, it took until March for MCP hit critical mass in March, after it gaining popularity among prominent, vocal supporters like Shopify's CEO Tobi Lutke.

In April, OpenAI and Google Gemini announced they would support MCP, making it the default.

### **MCP Primitives**

MCP has two basic primitives: *servers* and *clients*.

*Servers* wrap sets of MCP tools. They (and their underlying tools) can be written in any language and communicate with clients over HTTP.

*Clients* such as models or agents can query servers to get the set of tools provided, then request that the server execute a tool and return a response.

As such, MCP is as a standard for remote code execution, like OpenAPI or RPC.

### **The MCP Ecosystem**

As MCP was gaining traction, a bunch of folks joined the fray.<sup>44</sup> SAM BHAGWAT

*Vendors* like Stripe began shipping MCP servers for their API functionality.

*Independent developers* started making

MCP servers for functionality they needed, like browser use or, and publishing them on Github Registries like Smithery, PulseMCP, and mcp.run popped up to catalogue the growing ecosystem of servers (as well as validate the quality and safety of providers).

Frameworks like Mastra started shipping MCP server and client abstractions so that individual developers didn't have to reimplement specs themselves.

### **When to use MCP**

Agents, like SaaS, often need a number of basic integrations with third-party services (calendar, chat, email, web). If your roadmap has a lot of this kind of feature, it's worth looking at building an MCP *client* that could access third-party features.

Conversely, if you're building a tool that you want *other* agents to use, you should consider shipping an MCP *server*. *Principles of Building AI Agents* 45

### **Building an MCP Server and Client**

If you want to create MCP servers and give an agent access to them, here's how you can do that in TypeScript with Mastra:46 SAM BHAGWAT

Conversely, if you want to create a client with access to other MCP servers, here's how you would do that:

### **What's next for MCP**

MCP as a protocol is technically impressive, but the ecosystem is still working to resolve a few challenges:

**First, discovery.** There's no centralized or standardized way to find MCP tools. While various registries have popped up, this has created its own sort of fragmentation.

In April, we somewhat tongue-in-cheek built the *Principles of Building AI Agents* 47

first MCP Registry Registry, but Anthropic is actually working on a meta-registry

**Second, quality.** There's no equivalent (yet) of NPM's package scoring or verification badges. That said, the registries (which have rapidly raised venture funding) are working hard on this.

**Third, configuration.** Each provider has its own configuration schema and APIs. The MCP spec is long, and clients don't always implement them

completely.

### **Conclusion**

You could easily spend a weekend debugging subtle differences between the way that Cursor and Wind-surf implemented their MCP clients (and we did).

There's alpha in playing around with MCP, but you probably don't want to roll your own, at least not right now. Look for a good framework or library in your language.