

[Open in app](#)

≡ Medium

Search

 Write



Level Up Coding

 Member-only story

The guide to MCP I never had



Anmol Baranwal

[Follow](#)

22 min read · Apr 28, 2025

 2.1K

 27



...



AI agents are finally stepping beyond chat. They are solving multi-step problems, coordinating workflows and operating autonomously. And behind many of these breakthroughs is MCP.

MCP is going viral. But if you are overwhelmed by the jargon, you're not alone.

Today, we will explore why existing AI tools fall short and how MCP solves that problem. We will cover the core components, why they matter, 3-layer architecture and its limitations. You will also find practical examples with use cases.

Just the guide I wish I had when I started.

Please note: I've made this story Member-only, but you can still read it for free with this [friend link](#).

What is covered?

In a nutshell, we are covering these topics in detail.

1. The problem of existing AI tools.
2. Introduction to MCP and its core components.
3. How does MCP work under the hood?
4. The problem MCP solves and why it even matters.
5. The 3 Layers of MCP (and how I finally understood them).

6. The easiest way to connect 100+ managed MCP servers with built-in Auth.
7. Six practical examples with demos.
8. Some limitations of MCP.

We will be covering a lot so let's get started.

1. The problem of existing AI tools.

If you've ever tried building an AI agent that actually does stuff like checking emails or sending Slack messages (based on your workflow), you know the pain: the process is messy and most of the time the output is not worth it.

Yes, we have got amazing APIs. Yes, tools exist.

But practical usage and reliability aren't that much.

Even tools like Cursor (which got ultra hyped on Twitter) are getting recent complaints about poor performance.

1. Too many APIs, not nearly enough context

Every tool you want the AI to use is basically a mini API integration. So imagine a user says: "Did Anmol email me about yesterday's meet report?"

For an LLM to answer, it has to:

- Realize this is an email search task, not a Slack or Notion query.
- Pick the correct endpoint let's say `search_email_messages`

- Parse and summarize the results in natural language

All while staying within the context window. That's a lot. Models often forget, guess or hallucinate their way through it.

And if you cannot verify the accuracy, you don't even realize the problem.

2. APIs are step-based but LLMs aren't good at remembering steps.

Let's take a basic example of CRM.

- First, you get the contact ID → `get_contact_id`
- Then, fetch their current data → `read_contact`
- Finally, patch the update → `patch_contact`

In traditional code, you can abstract this into a function and be done. But with LLMs? Each step is a chance to mess up due to a wrong parameter, missed field or broken chain. And suddenly your “AI assistant” is just apologizing in natural language instead of updating anything.

3. Fragile tower of prompt engineering

APIs evolve. Docs change. Auth flows get updated. You might wake up one morning to find that your perfectly working agent now breaks due to third-party changes.

And unlike traditional apps, there's no shared framework or abstraction to fall back on. Every AI tool integration is a fragile tower of prompt engineering, JSON crafting. It risks breaking your AI agent's “muscle memory.”

4. Vendor lock-in.

Built your tools for GPT-4? Cool. But you will need to rewrite all your function descriptions and system prompts from scratch if you ever switch to other tools like Claude or Gemini.

It's not such a big issue but there is no such universal solution.

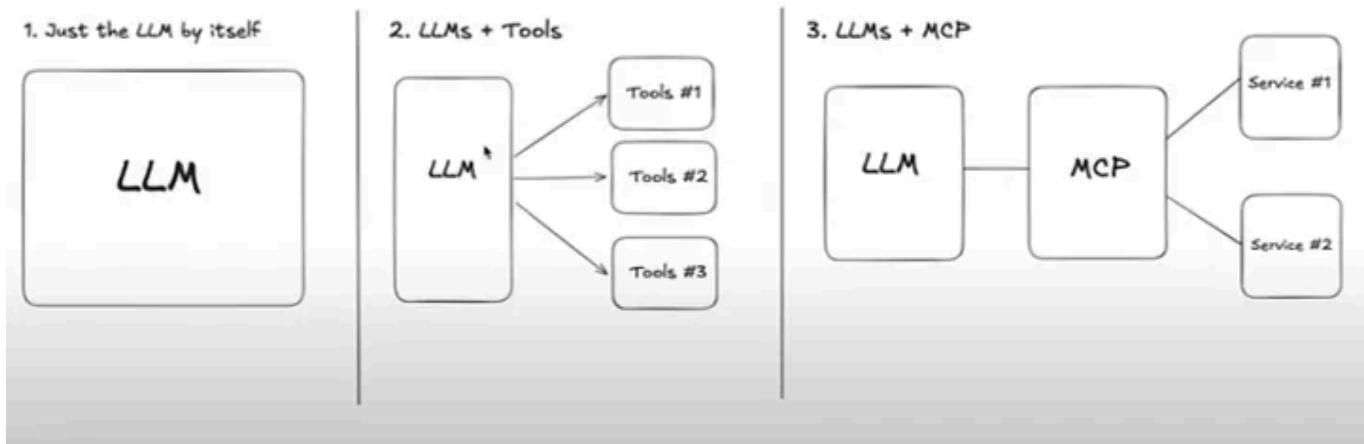
There has to be a way for tools and models to communicate cleanly, without stuffing all the logic into bloated prompts. That's where MCP comes in.

2. Introduction to MCP and its core components.

Model Context Protocol (MCP) is a new open protocol that standardizes how applications provide context and tools to LLMs.

Think of it as a universal connector for AI. MCP works as a plugin system for Cursor which allows you to extend the Agent's capabilities by connecting it to various data sources and tools.

LLMs by themselves are incapable of doing anything meaningful



MCP helps you build agents and complex workflows on top of LLMs.

For example, an MCP server for Obsidian helps AI assistants search and read notes from your Obsidian vault.

Your AI agent can now:

- Send emails through Gmail
- Create tasks in Linear
- Search documents in Notion
- Post messages in Slack
- Update records in Salesforce

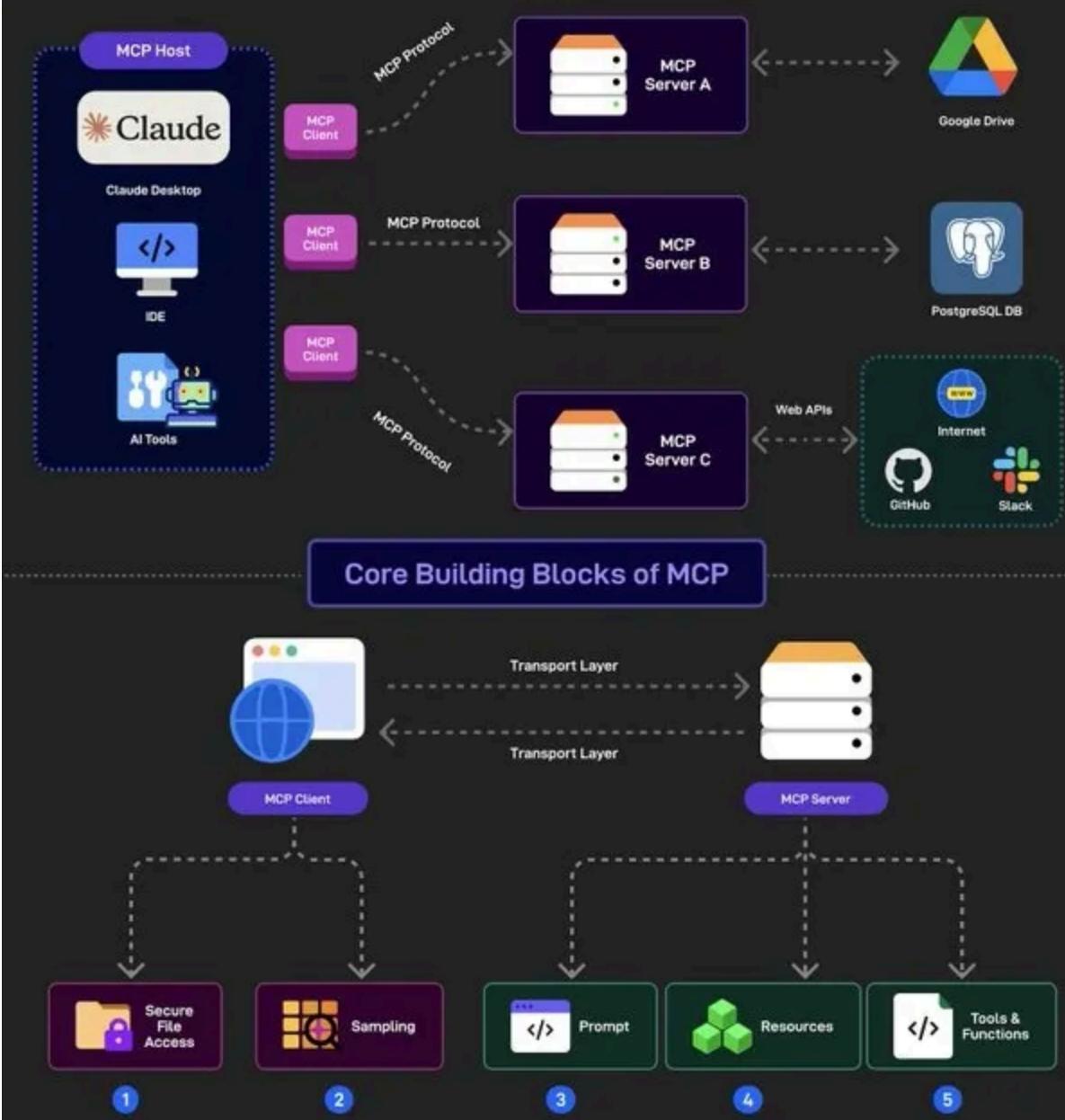
All by sending natural-language instructions through a standardized interface.

Think about what this means for productivity. Tasks that once required switching between 5+ apps can now happen in a single conversation with your agent.

At its core, MCP follows a client-server architecture where a host application can connect to multiple servers.

What is the MCP by Anthropic?

ByteByteGo



Credit goes to ByteByteGo

Core Components.

Here are the core components in any general MCP Server.

- MCP hosts - apps like Claude Desktop, Cursor, Windsurf or AI tools that want to access data via MCP.

- MCP Clients - protocol clients that maintain 1:1 connections with MCP servers, acting as the communication bridge.
- MCP Servers - lightweight programs that each expose specific capabilities (like reading files, query databases...) through the standardized Model Context Protocol.
- Local Data Sources - files, databases and services on your computer that MCP servers can securely access. For instance, a browser automation MCP server needs access to your browser to work.
- Remote Services - External APIs and cloud-based systems that MCP servers can connect to.

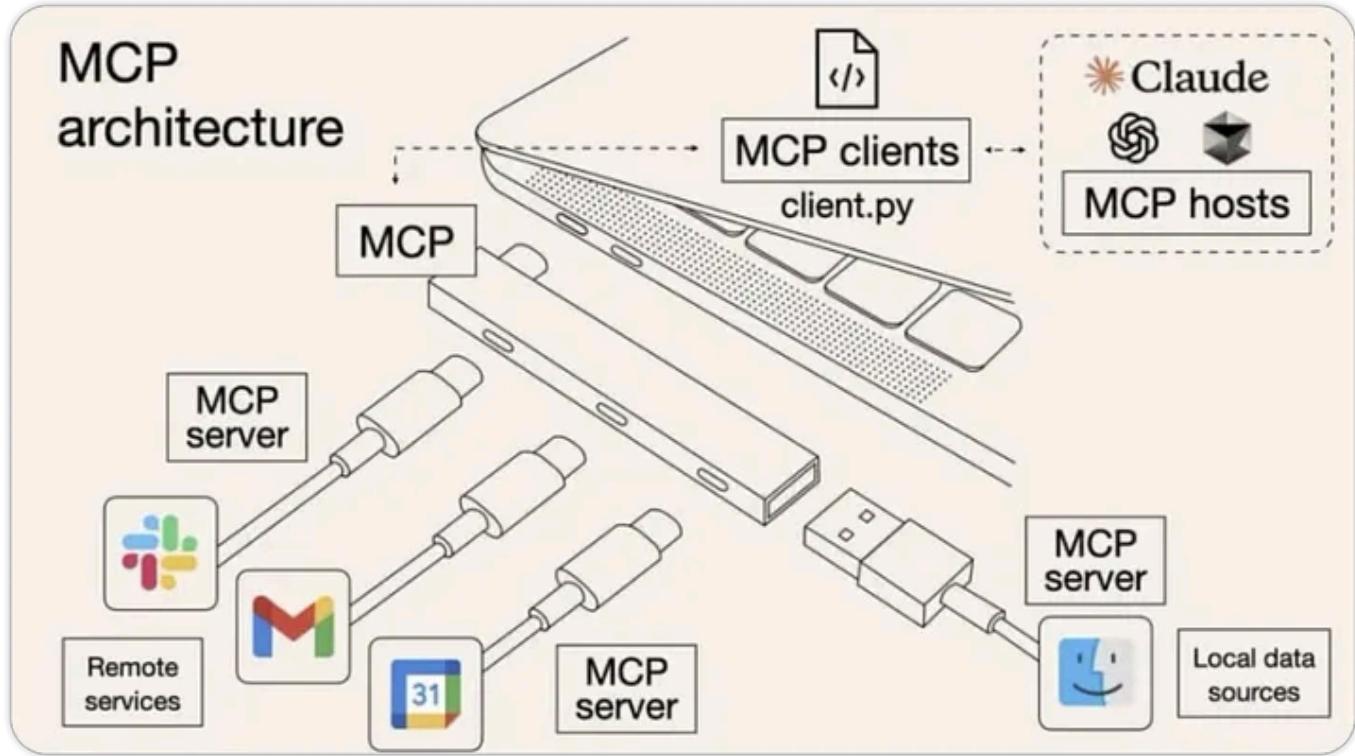
If you're interested in reading the architecture, check out [official docs](#). It covers protocol layers, connection lifecycle and error handling with the overall implementation.

We are going to cover everything but if you're interested in reading more about MCP, check out these two blogs:

- [What is the Model Context Protocol \(MCP\)?](#) by the Builder.io team
- [MCP: What It Is and Why It Matters](#) by Addy Osmani

3. How does MCP work under the hood?

The MCP ecosystem comes down to [several key players](#) that function together. Let's study in brief about them.



Credit goes to Huggingface

- **Clients.**

Clients are the apps you actually use like Cursor, Claude Desktop, and others. Their job is to:

- Request available capabilities from an MCP server
- Present those capabilities (tools, resources, prompts) to the AI model
- Relay the AI's tool usage requests back to the server and return the results
- Provide models with the basic MCP protocol overview for consistent interaction

They handle communication between the system's frontends: the user, the AI model, and the MCP server.

- **Servers.**

MCP servers serve as intermediaries between users/AI and external services.

They:

- Offer a standardized JSON-RPC interface for tool and resource access
- Convert existing APIs into MCP-compatible capabilities without requiring API changes
- Handle authentication, capability definitions and communication standards

They provide context, tools and prompts to clients.

- **Service providers.**

These are external systems or platforms (like Discord, Notion, Figma) that perform actual tasks. They don't change their APIs for MCP.

This whole setup allows developers to plug any compatible API into any MCP-aware client, avoiding dependence on centralized integrations by large AI providers.

MCP Building Blocks: Tools, Resources and Prompts

- Tools.

Tools represent actions an AI can perform such as `search_emails` or `create_issue_linear`. They form the foundation of how models execute real-world functions through MCP.

- Resources.

Resources represent any kind of data that an MCP server wants to make available to clients. This can include:

- File contents
- Database records
- API responses
- Live system data
- Screenshots and images
- Log files
- And more

Each resource is identified by a unique URI (like `file://user/prefs.json`) which can be project notes, coding preferences or anything specific to you. It contains either text or binary data.

Resources are identified using URIs that follow this format.

```
[protocol]://[host]/[path]
```

For example:

- `file:///home/user/documents/report.pdf`
- `postgres://database/customers/schema`
- `screen://localhost/display1`

Servers can also define their own custom URI schemes. You can read more on [official docs](#).

- [Prompts](#).

Tools let the AI do stuff, but prompts guide the AI on how to behave while doing it.

It's like instructions to the model during tool usage. They act like operational guides helping the AI follow specific styles, workflows or safety protocols like if it follows a specific safety checklist before hitting that `delete一切` button.

🎯 Let's explore a practical scenario:

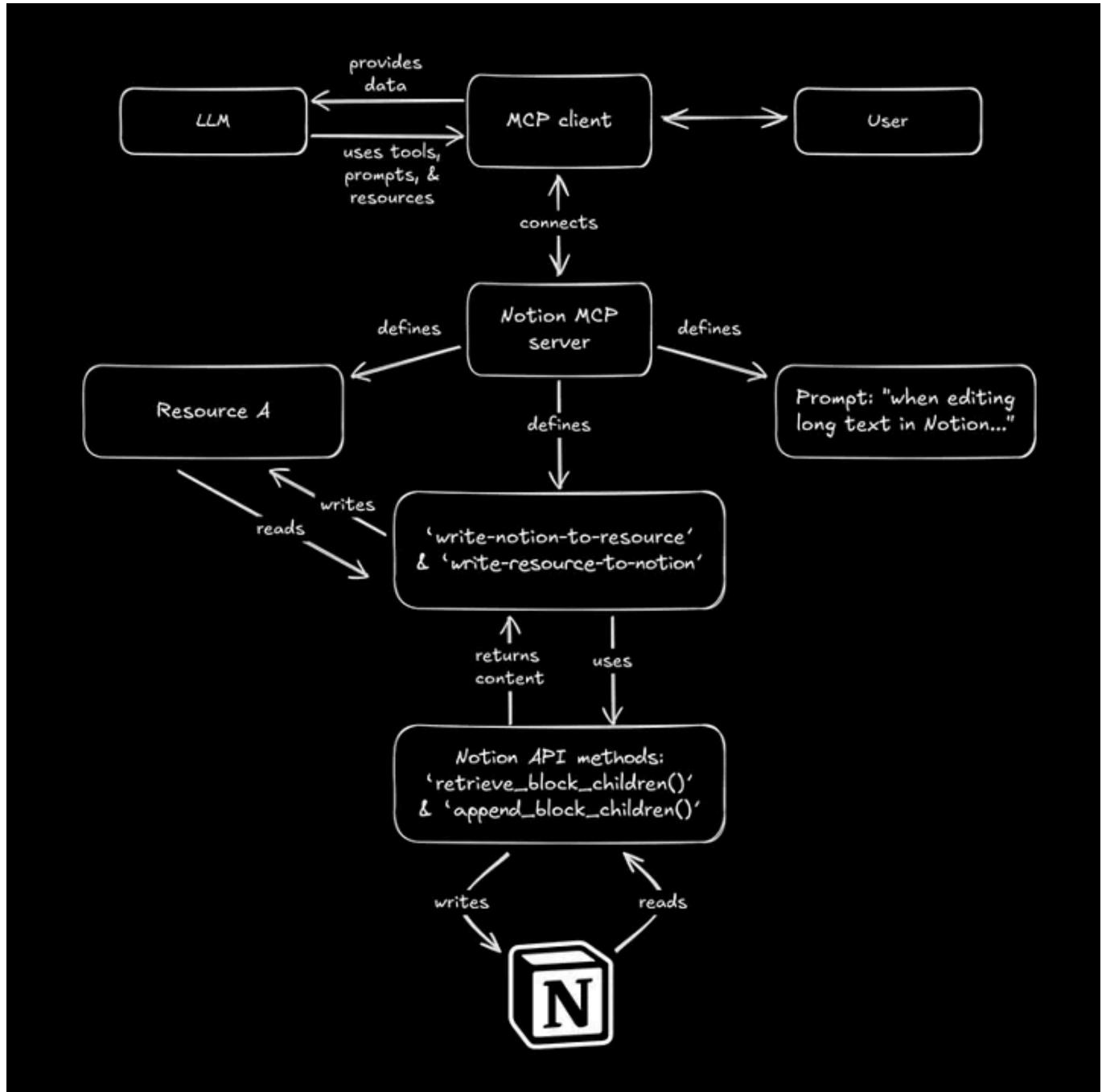
Imagine a Google Calendar MCP server. The Calendar API is powerful but talkative, every event includes fields for guests, time zones, reminders, attachments and more. If you ask an AI model to `reschedule all my meetings with Alice next week`, it may struggle to filter the relevant data from the noise.

This is where prompts and resources come in.

An MCP prompt could instruct the model:

“When working with calendar events, only modify those with title or participant match. Extract relevant events using the `list-events` tool, copy them into a temporary resource (`Resource B`), apply changes there and use `update-events-from-resource` to sync them back.”

This pattern lets the AI focus on clean, editable data in a controlled state (the resource), guided by reusable, standardized instructions (the prompt) with proper actions (tool).



Notion example by Builder.io

You should also read the notion example on [Builder.io](#), they have covered it under MCP prompts.

When a client connects to an MCP server, it asks for the list of available tools, resources and prompts. Based on the user's input, the client selects what to show the model. When the model chooses an action, the client executes it via the server and ensures proper authorization with data flow.

4. The problem MCP solves and why it even matters.

Let's briefly discuss the problems that MCP solves.

- One common protocol = thousands of tools.

This common protocol means one AI can integrate with thousands of tools as long as those tools have an MCP interface, eliminating the need for custom integrations for each new app.

Services describe what they can do ("send Discord message", "create Linear ticket") and how to do it (parameters, auth methods) using a consistent JSON-RPC format.

- Clear separation of roles: model thinks, tools act.

It creates a clear separation between the AI model (the thinker) and the external tools (the doers). Your fancy AI agent doesn't break every time Slack tweaks its API.

- With MCP, you don't have to redo all your tool descriptions when swapping GPT for Claude or Gemini. Your tools and logic stay the same.

- MCP supports memory and multi-step workflows meaning your agent can remember things across tasks and chain actions together intelligently.
- It leads to fewer hallucinations. In a general manner, MCP uses clear, structured tool definitions. That helps AI stay grounded and accurate.

Why does MCP matter?

MCP matters because:

- It turns the dream of a universal AI assistant for developers into a practical reality.
- The potential to compose these actions into sophisticated workflows (with the AI handling the logic) will lead to a new era of intelligent automation .

So MCP makes it way easier for developers to do more with AI.

5. The 3 Layers of MCP (and how I finally understood them).

This is how I've understood the concept in detail. I will attach a common example which will help you understand it very quickly.

Model ↔ Context: “Talk to the LLM in a way it understands”

Imagine the Model as the brain of a robot (LLM). It can process information but needs clear instructions. Context provides those instructions to work correctly.

For example: if you tell a robot, “Make me a sandwich,” that’s too vague. But saying “Use this bread, ham, and cheese to make a sandwich” gives the robot a context to understand and execute the task.

- Model is the robot (LLM).
- Context is the specific instructions you give it (ingredients for the sandwich).

Context ↔ Protocol: “Give the LLM structured memory, tools, state”

Once the robot has instructions (Context), it needs a way to follow them, remember details and use tools. That is done by `Protocol`, it's the system that lets the robot use memory and tools to get the job done.

Let's take the same sandwich example. Giving it a protocol will help it remember the ingredients, know how to handle the knife and more.

- Context tells the robot what to do.
- Protocol gives it the tools and memory to do it.

It's the structure for getting things done.

Protocol ↔ Runtime: “Actually run the AI Agent”

The robot knows what to do (Context) and how to do it (Protocol). Now it needs to actually do it, which is possible using Runtime.

Going back to the sandwich example, the Runtime is the moment it starts executing it. It's like the environment where the task comes to life (like the kitchen).

- Protocol gives the robot the method to follow.
- Runtime is the environment where the robot actually works.

Let's take all three layers together and see what comes using the `restaurant` version .

- The `Model` is the chef. They have the knowledge and skills to make food.
- The `Context` is the menu. It tells the chef what ingredients are needed and how the meal should look and taste.
- The `Protocol` is the waiter. The waiter brings the order to the chef, communicates exactly how the dish should be prepared and even remembers if you're allergic to something.
- The `Runtime` is the kitchen where the chef actually prepares the meal. It's the place where all the tools, heat and preparation happen.

Once you understand the core components like servers and clients (covered in the “How MCP works under the hood” section), it all starts to make sense.

Each layer fits together to make the whole system work.

6. The easiest way to connect 100+ managed MCP servers with built-in Auth.

In this section, we will be exploring the easiest way to connect Cursor with MCP servers.

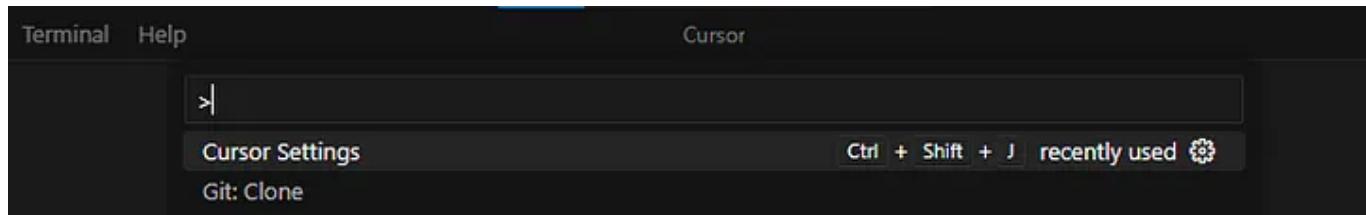
If you want to explore how to add and use custom MCP servers within the Cursor, read the [official docs](#).

Step 1: Prerequisites.

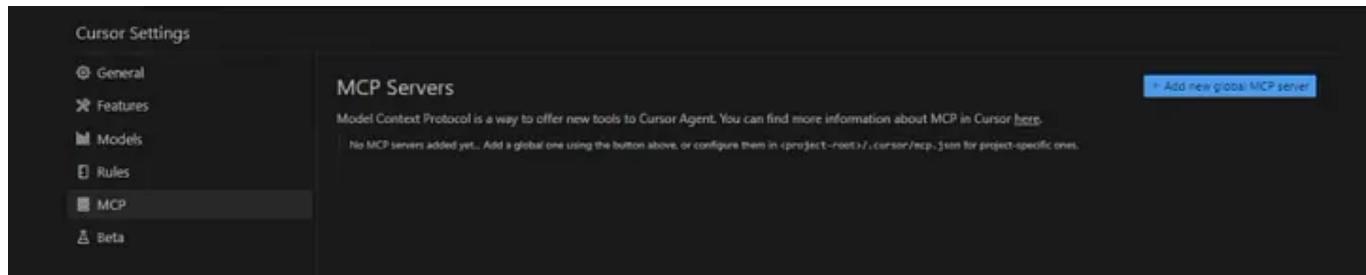
Install Node.js and ensure `npx` is available in your system.

Step 2: Enable MCP server in Cursor.

You can open the command palette in Cursor with `Ctrl + Shift + P` and search for cursor settings.



You will find an MCP option on the sidebar.



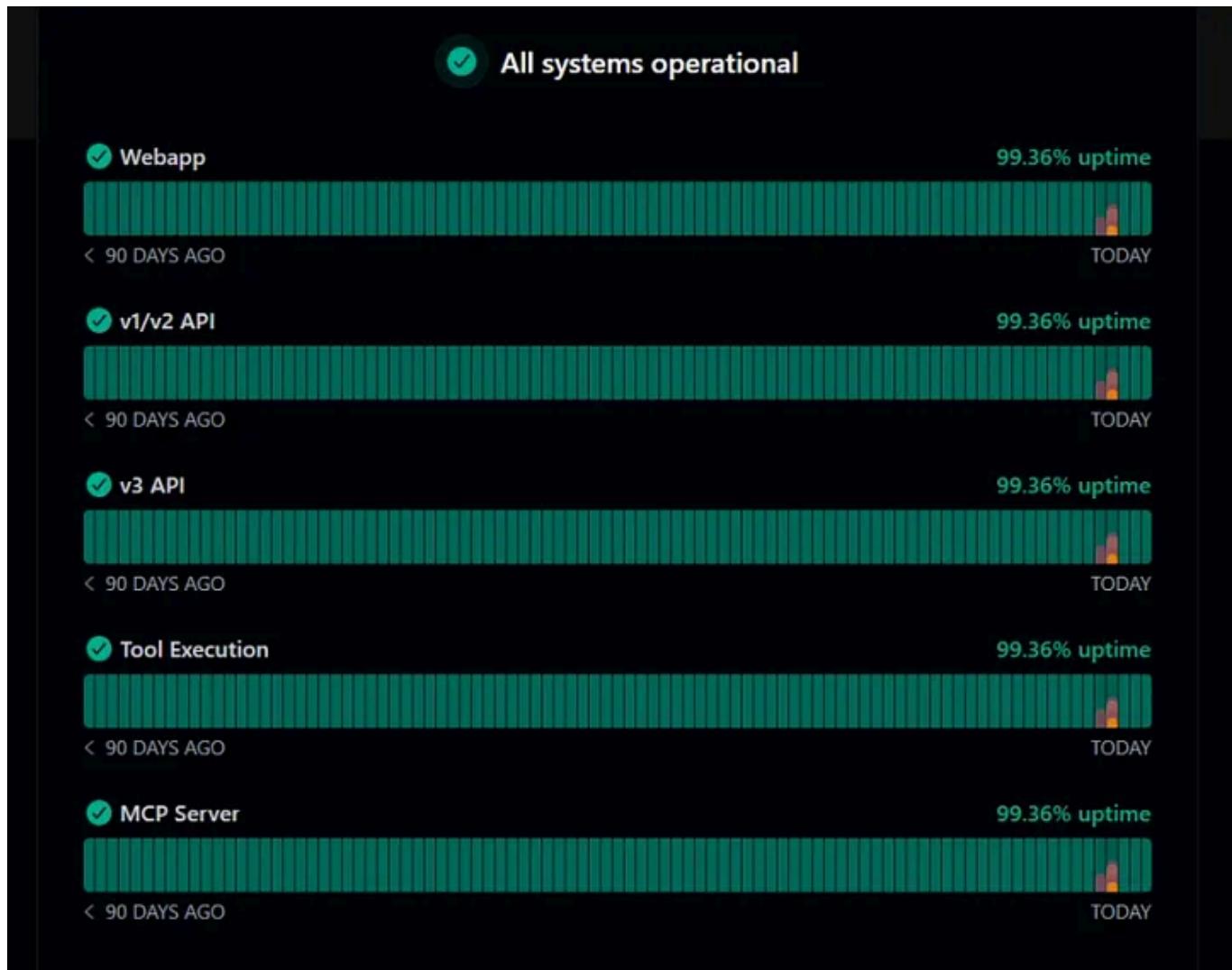
Step 3: Using a pre-defined MCP server.

We can also create one from scratch but let's use predefined-one for the sake of simplicity.

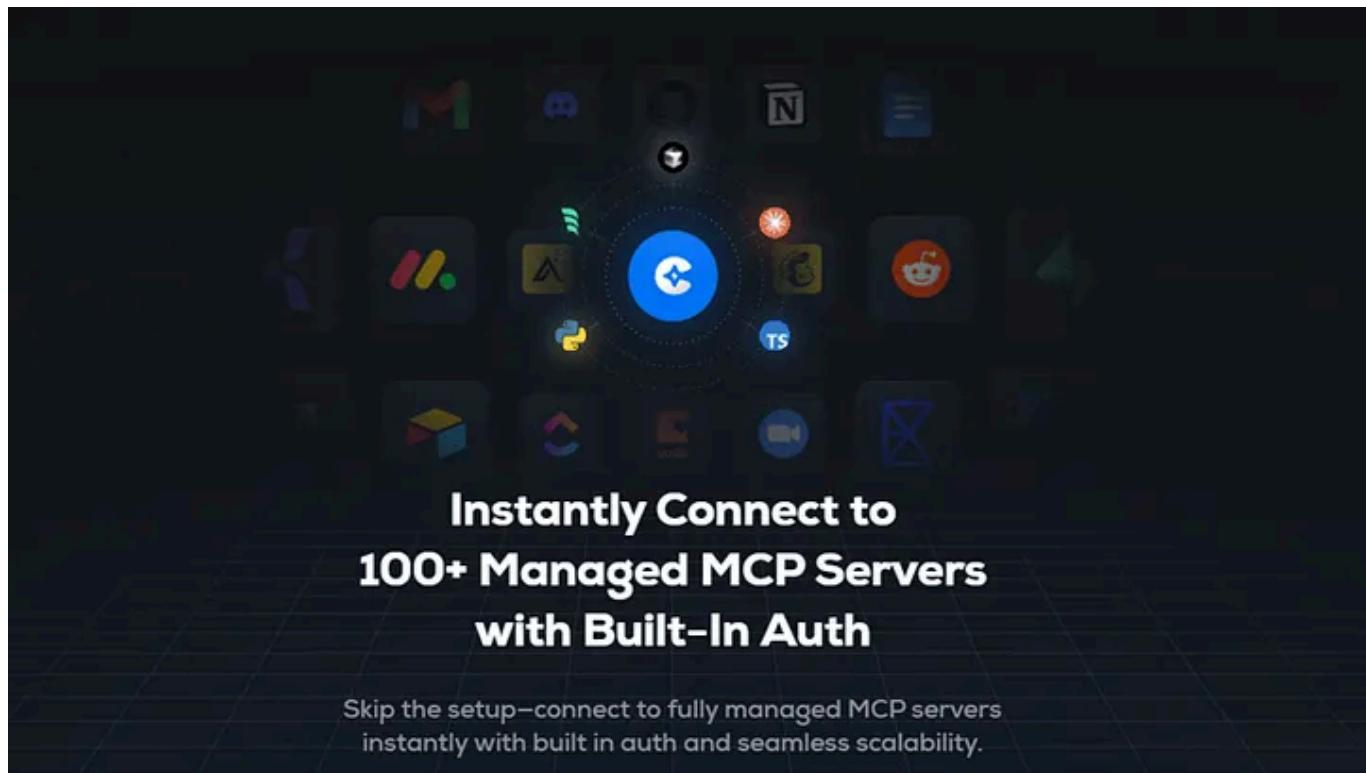
We will use Composio for the servers since they have built-in auth. You can find the list at [mcp.composio.dev](#).

- Built-in Auth comes with support for OAuth, API keys, JWT and Basic Auth. This means you don't have to create your own login system.
- Fully managed servers eliminate the need for complex setups, making it easy to integrate AI agents with 250+ tools like Gmail, Slack, Notion, Linear and more.
- Provides 20,000+ pre-built API actions for quick integration without coding.
- Can operate locally or remotely depending on your configuration needs.
- Better tool-calling accuracy allows AI agents to interact smoothly with integrated apps.
- It's compatible with AI agents which means it can connect AI agents to tools for tasks like sending emails, creating tasks or managing tickets in a single conversation.

It also means less downtime and fewer maintenance problems. You can check the status at status.composio.dev.



You can easily integrate with a bunch of useful MCP servers without writing any code.



With each of the options, you will find the total active users, its current version, how recently it was updated and all the available actions.

You will find instructions to install it using TypeScript, Python and it supports Claude (MacOS), Windsurf (MacOS) and Cursor as MCP hosts.

Gmail

Connect to Gmail to send and manage emails.

COLLABORATION & COMMUNICATION

TOTAL VIEWS	ACTIVE USERS	VERSION	LAST UPDATED
21556	4421	0.1.0	14H AGO

Available Actions
Get all 21 actions for this app by following the instruction guide.

Installation Steps

Run this command in your terminal to set up Cursor with MCP:

```
cursor add gmail
```

Available Actions

Get all 21 actions for this app by following the instruction guide.

Fetch emails
Action to fetch all emails from gmail.

Step 4: Integrating MCP server.

It's time to integrate one with the cursor. For now, we will be using the Gmail MCP server.

Previously it was with SSE but Cursor recently changed this method with the `npx` command. We will need to generate the terminal command. Check [this page](#) to generate yours.

The terminal command will look like this.

```
npx @composio/mcp@latest setup "https://mcp.composio.dev/gmail/xyzxyz..." --cli
```

You can run this command in the terminal and restart Cursor to notice the changes.

If you're using Python, here's how you can Install the `composio-toolset`.

```
pip install composio_openai

from composio_openai import ComposioToolSet, App
from openai import OpenAI

openai_client = OpenAI()
composio_toolset = ComposioToolSet(entity_id="default")
tools = composio_toolset.get_tools(apps=[App.GMAIL])
```

You can place the final configuration in two locations, depending on your use case:

- 1) For tools specific to a project, create a `.cursor/mcp.json` file in your project directory. This allows you to define MCP servers that are only available within that specific project.
- 2) For tools that you want to use across all projects, create a `\~/.cursor/mcp.json` file in your home directory. This makes MCP servers available in all your Cursor workspaces. The terminal would enforce the second option which will make it globally accessible.

```
● PS C:\Users\DELL> npx @composio/mcp@latest setup "https://mcp.composio.dev/gmail/freezing-wrong-dress-7RHvW0" --client cursor
Configuration Details:
URL: https://mcp.composio.dev/gmail/freezing-wrong-dress-7RHvW0
Client: cursor

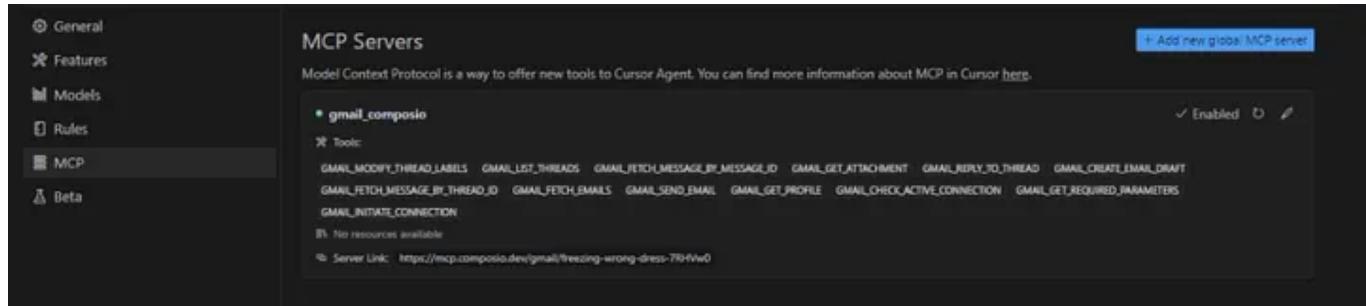
Saving configurations...
Configuration saved to: C:\Users\DELL\.cursor\mcp.json

All done! Please restart cursor for changes to take effect

Saving configurations...
Configuration saved to: C:\Users\DELL\.cursor\mcp.json

All done! Please restart cursor for changes to take effect
```

It will display the necessary actions and status green dot which indicates, that it's successfully integrated.



The `mcp.json` will look like this.

```
{  
  "mcpServers": {  
    "gmail_composio": {  
      "url": "https://mcp.composio.dev/gmail/freezing-wrong-dress-7RHVw0"  
    }  
  }  
}
```

You can check out the list of [sample servers and implementations](#). You can integrate the community servers by following this structure (based on your choice of preference).

SSE Server Configuration.

This configuration is supported in Cursor and you can specify the `url` field to connect to your SSE server.

```
// This example demonstrated an MCP server using the SSE format  
// The user should manually set and run the server  
// This could be networked, to allow others to access it too  
{  
  "mcpServers": {  
    "server-name": {  
      "url": "http://localhost:3000/sse",  
      "env": {  
        "API_KEY": "value"  
      }  
    }  
  }  
}
```

STDIO Server Configuration (Python)

This sets up an MCP server using the standard input/output (STDIO) transport with a Python script. This approach is mainly used for local development.

```
// if you're using CLI server Python
// This example demonstrated an MCP server using the stdio format
// Cursor automatically runs this process for you
// This uses a Python server, run with `python`
{
  "mcpServers": {
    "server-name": {
      "command": "python",
      "args": ["mcp-server.py"],
      "env": {
        "API_KEY": "value"
      }
    }
  }
}
```

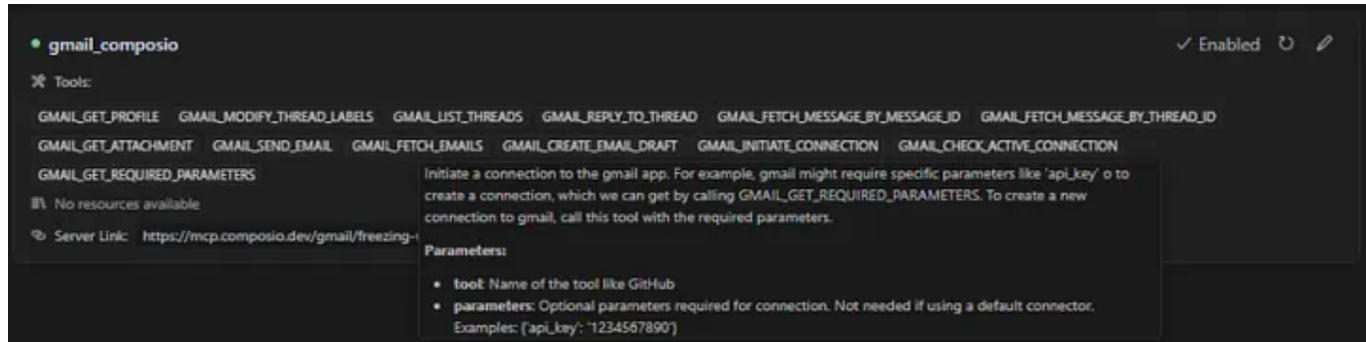
STDIO Server Configuration (Node.js)

```
// if you're using CLI server Node.js
// This example demonstrated an MCP server using the stdio format
// Cursor automatically runs this process for you
// This uses a Node.js server, ran with `npx`
{
  "mcpServers": {
    "server-name": {
      "command": "npx",
      "args": ["-y", "mcp-server"],
      "env": {
        "API_KEY": "value"
      }
    }
  }
}
```

```
    }  
}
```

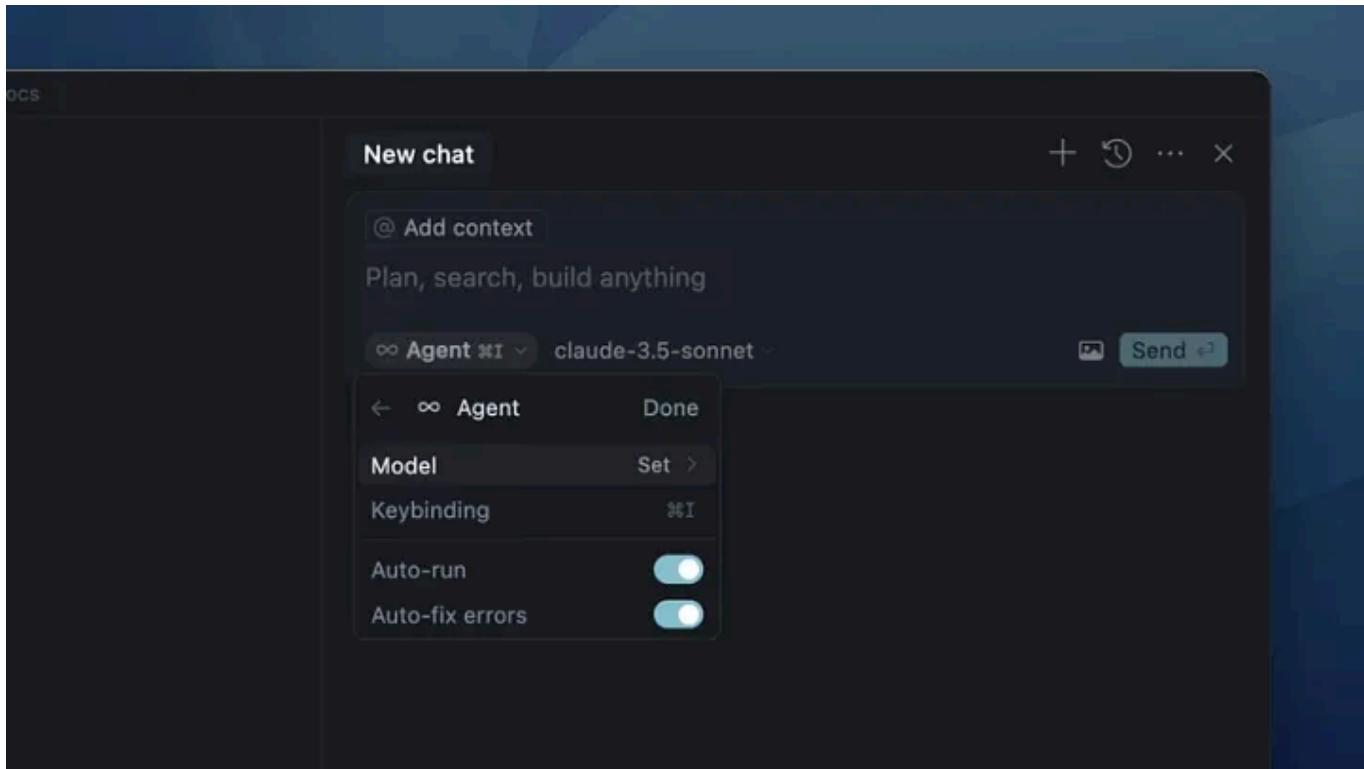
Step 5: Using the server directly within Agent.

Before proceeding, make sure to check available actions on [composio mcp server page](#). You can also find the tools and actions on the dashboard.



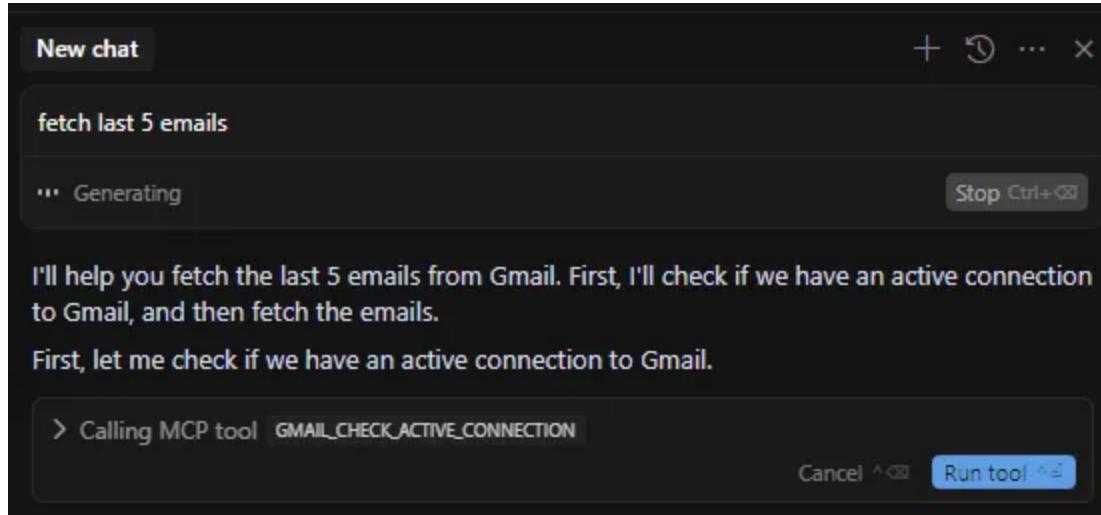
You can open the Chat using the `ctrl + I` command.

You can enable `Agent Mode` which is the most autonomous mode in Cursor, designed to handle complex coding tasks with minimal guidance.



I prefer having some control before executing, so I'm going with the default one. You can type any query. You just need to click on the `run` tool button.

As you can see, it will call the appropriate MCP server (if you have multiple of those) and it will accordingly use the correct action based on your prompt.



Since there is no active connection, it will first establish one. You will need to authorize the process.

I see we don't have an active connection to Gmail yet. Let me help you set that up first.

> Called MCP tool GMAIL_INITIATE_CONNECTION ✓

To proceed, you'll need to:

1. Click on this URL to authorize Gmail access: <https://backend.composio.dev/s/lG1gLEC6>
2. Follow the authentication process
3. Once you've completed the authorization, please let me know and I'll fetch your emails.

After you complete the authentication, I'll fetch the last 5 emails from your inbox. Please let me know when you've completed the authorization process.

Composio wants access to your Google Account

anmolbaranwal09@gmail.com

Select what Composio can access

 Read, compose, and send emails from your Gmail account. [Learn more](#)

Because you're using Sign in with Google, Composio will be able to

 See your personal info, including any personal info you've made publicly available

Make sure you trust Composio

Review Composio's [Privacy Policy](#) and [Terms of Service](#) to understand how Composio will process and protect your data.

To make changes at any time, go to your [Google Account](#).

Learn how Google helps you [share data safely](#).

[Cancel](#) [Continue](#)

I'm using a dummy account (that I created a long time ago) and I recommend doing the same for testing purposes. Once you're satisfied, you can automate things with your primary account.

As you can see, it properly fetched the emails.

The screenshot shows a dark-themed mobile application interface. At the top, there is a 'done' button. Below it, a message says 'Great! Let me verify the connection and then fetch your last 5 emails.' A log entry shows a successful call to 'GMAIL_CHECK_ACTIVE_CONNECTION'. The next message states 'Perfect! The connection is active. Now, I'll fetch your last 5 emails.' Another log entry shows a successful call to 'GMAIL_FETCH_EMAILS'. The application then displays the subject of the last five emails:

- 1. Security alert (Just now)**
 - From: Google
 - Subject: Security alert
 - Content: Notification about Composio being granted access to your Google Account
- 2. AI Hub Update (April 19, 2025)**
 - From: Olha from Visme
 - Subject: 🔥 New: AI Hub, Your One-Stop AI Menu in Visme
 - Content: Update about new AI features in Visme

Let's check by sending an email to `hi@anmolbaranwal.com` with the subject "Demo of Composio" and saying testing MCP server in the body of the email.

send an email to hi@anmolbaranwal.com with subject "Demo of Composio" and saying testing MCP server

I'll help you send an email with those specifications.

> Called MCP tool GMAIL_SEND_EMAIL ✓

I've sent the email successfully! The email has been:

- Sent to: hi@anmolbaranwal.com
- Subject: Demo of Composio
- Message: testing MCP server

The email has been delivered and labeled as "SENT" in your Gmail account. Is there anything else you'd like me to help you with?

↳ ↻ ⌂ ⌂ ...

As you can see I've received that email with the proper subject and body as specified in the prompt.

Demo of Composio

Anmol Baranwal <anmolbaranwal09@gmail.com>
to hi ▾
testing MCP server

4:03 PM (2 minutes ago) ★ ☺ ↵ :

Reply Forward

sent the email

Demo of Composio ▷ [Inbox](#) [x](#) [hi@anmolbaranwal.com](#) [x](#)

Anmol Baranwal
to hi ▾
testing MCP server

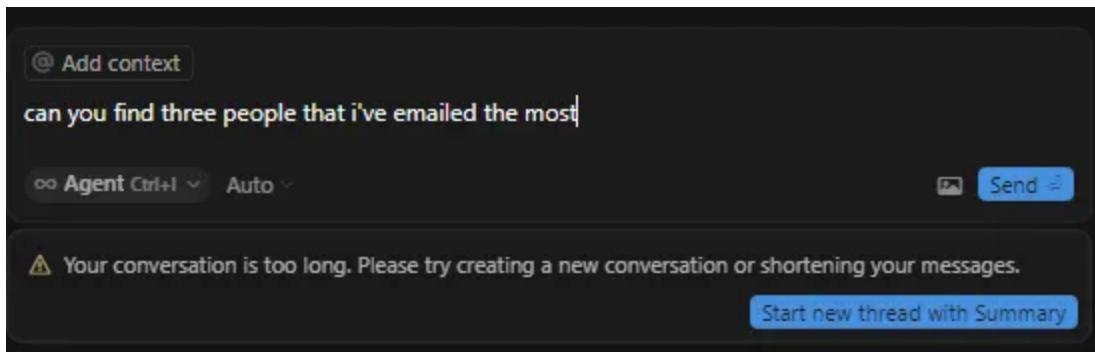
4:03 PM (1 minute ago) ★ ☺ ↵ :

Reply Forward

received the email

With this MCP server, you can do lots of amazing things like Get attachments , Create email draft , Modify thread labels , Reply to a thread , get contacts , delete message , move to trash , search people , send email and much more.

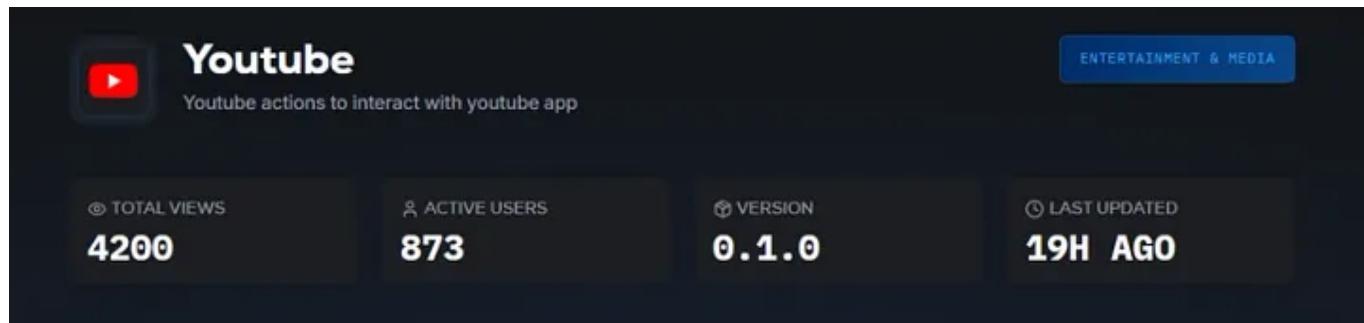
And always remember, there is a limit to what you can do. I've tested with more than 15 prompts to analyze the edge cases.



7. Six practical examples with demos.

Here are five practical examples of MCP servers. Let's discuss the flow and see the application.

✓ YouTube MCP Server



We will follow the same flow as discussed before, you can check the Composio server for YouTube where you can generate the url.

The screenshot shows a user interface for generating a secure MCP URL. At the top left is the text '>_ Installation Steps'. To the right is a 'View Docs' button. Below this, a message says 'Run this command in your terminal to set up Cursor with MCP:'. Underneath is a button labeled 'Click to generate your secure MCP URL' with a lock icon, and a blue 'Generate' button to its right.

The command will be structured like:

```
npx @composio/mcp@latest setup "https://mcp.composio.dev/youtube/freezing-wrong-
```

If you notice in the `mcp.json`, the url for youtube will be added as soon as you run the terminal command. It will look something like this.

```
{
  "mcpServers": {
    "youtube_composio": {
      "url": "https://mcp.composio.dev/youtube/freezing-wrong-dress-7RHVw0"
    }
  }
}
```

Since there is no active connection, it will first establish one. You will need to authenticate by copying the OAuth URL in the browser.

I'll help you search for YouTube videos about the Model Context Protocol (MCP). I'll use the YouTube search functionality and then get detailed information about the videos to sort them by views and likes.

First, let me search for relevant videos:

> Called MCP tool YOUTUBE_SEARCH_YOU_TUBE ✓

I apologize, but it seems we need to establish a connection to YouTube first before we can perform the search. Let me set that up:

> Called MCP tool YOUTUBE_INITIATE_CONNECTION ✓

To proceed with searching YouTube videos, you'll need to authenticate first. Please:

1. Click on this URL to complete the authentication process:
<https://backend.composio.dev/s/SN-sbRus>
2. Follow the instructions to authorize the connection
3. Once you've completed the authorization, let me know and I'll continue with the search for Model Context Protocol videos.

After you've completed the authentication, I'll be able to:

1. Search for relevant videos
2. Get detailed information about each video including views and likes
3. Sort them to show you the top 5 most popular videos about MCP

Please let me know once you've completed the authentication process.

You will have to provide access to the server so it can take action based on your prompt.



Composio wants access to your Google Account

anmolbaranwal09@gmail.com

Select what Composio can access

Select all

See a list of your current active channel members, their current level, and when they became a member. [Learn more](#)

● See, edit, and permanently delete your YouTube videos, ratings, comments and captions. [Learn more](#)

▶ View private information of your YouTube channel relevant during the audit process with a YouTube partner. [Learn more](#)

▶ Manage your YouTube videos. [Learn more](#)

▶ View your YouTube account. [Learn more](#)

▶ View and manage your assets and associated content on YouTube. [Learn more](#)

▶ Manage your YouTube account. [Learn more](#)

Now, you can put any prompt such as Fetch me the top 5 videos about Model Context Protocol based on views and likes.

It will accordingly generate a response.

Top YouTube Videos on Model Context Protocol

+ ⌂ ⋮ ×

done

Great! Let me verify the connection and then proceed with the search:

> Called MCP tool YOUTUBE_CHECK_ACTIVE_CONNECTION ✓

Perfect! Now let me search for videos about Model Context Protocol:

> Called MCP tool YOUTUBE_SEARCH_YOU_TUBE ✓

Now, let me get the detailed statistics for these videos to rank them by views and likes:

> Called MCP tool YOUTUBE_VIDEO_DETAILS ✓

Based on the search results and video details, here are the top 5 YouTube videos about Model Context Protocol (MCP), ranked by views and engagement:

1. "I gave Claude root access to my server... Model Context Protocol explained" by Fireship

- Views: 947,631
- Likes: 37,520
- Channel: Fireship
- Link: <https://www.youtube.com/watch?v=HyzlYwjoXOQ>
- Description: A practical tutorial showing how to build an MCP server in TypeScript that gives Claude additional context and server modification abilities.

2. "Model Context Protocol (MCP): The Key To Agentic AI" by Jack Herrington

- Views: 92,718
- Likes: 3,561

3. "Model Context Protocol Clearly Explained | MCP Beyond the Hype" by codebasics

- Views: 90,652
- Likes: 1,984
- Channel: codebasics
- Link: <https://www.youtube.com/watch?v=tzrwxLNHtRY>
- Description: A technical deep-dive into MCP with code examples for both client and server implementations.

4. "Model Context Protocol (MCP) Explained: How to Give AI Access to Your Files, Web Searches & More" by Dan Vega

- Views: 29,734
- Likes: 924
- Channel: Dan Vega
- Link: <https://www.youtube.com/watch?v=nNLshWCoe0o>
- Description: Focuses on practical applications of MCP for file access and web searches using Claude Desktop.

5. "Model Context Protocol (MCP) Explained: The New Standard for AI Tools?" by Prompt Engineering

- Views: 18,887
- Likes: 505
- Channel: Prompt Engineering
- Link: <https://www.youtube.com/watch?v=rmphqjsc4Po>
- Description: An overview of MCP's role in standardizing AI tool integration and its future implications.

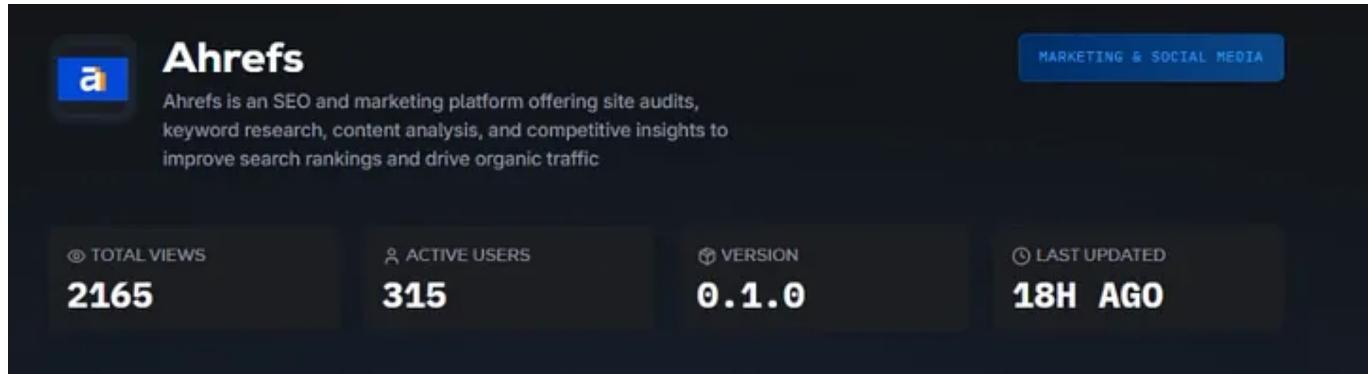
These videos offer a range of perspectives from practical implementations to theoretical

With this MCP server, you can do lots of amazing things like search youtube for videos, channels, playlists, fetch video stats, load captions, subscribe channel, update video's metadata, update thumbnail and much more.

Ahrefs MCP Server

We will follow the same flow as discussed before, you can check [Composio server for Ahrefs](#).

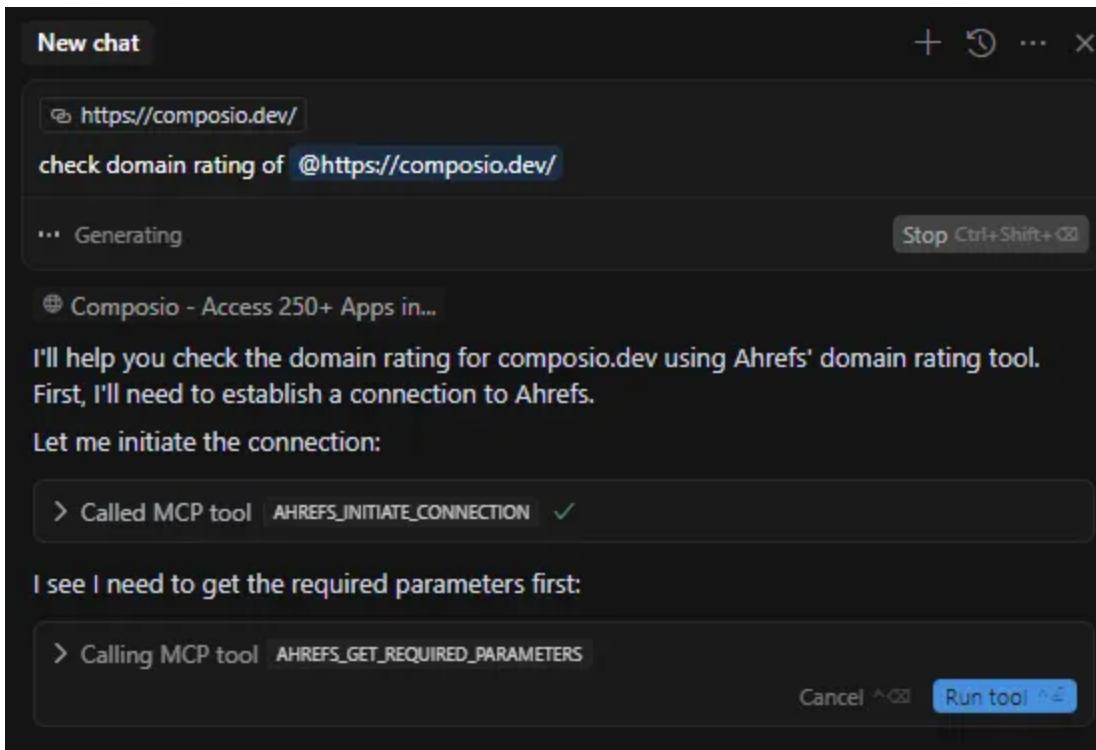
If you're unaware, Ahrefs is an SEO and marketing platform offering site audits, keyword research, content analysis and competitive insights to improve search rankings and drive organic traffic.



If you notice in the `mcp.json`, the ahrefs url will be added.

```
"ahrefs_composio": {  
    "url": "https://mcp.composio.dev/ahrefs/freezing-wrong-xyz"  
}
```

As you can see, it's establishing a connection for the first time.



Once you do that, you will be able to use all the actions like retrieve organic keywords , fetch all backlinks , domain rating history , pages by traffic overview , retrieve public crawler ips , fetch competitors overview , list best by external links , fetch total search volume history and much more.

Please note that you will need an API (which comes under the premium plan for Ahrefs) to complete the integration.

LinkedIn MCP server

You can follow the same process to generate the URL and run it in the terminal. You will then need to establish a connection and authenticate by copying the OAuth URL in the browser.

You will get a confirmation message once it's done.



Authentication successful

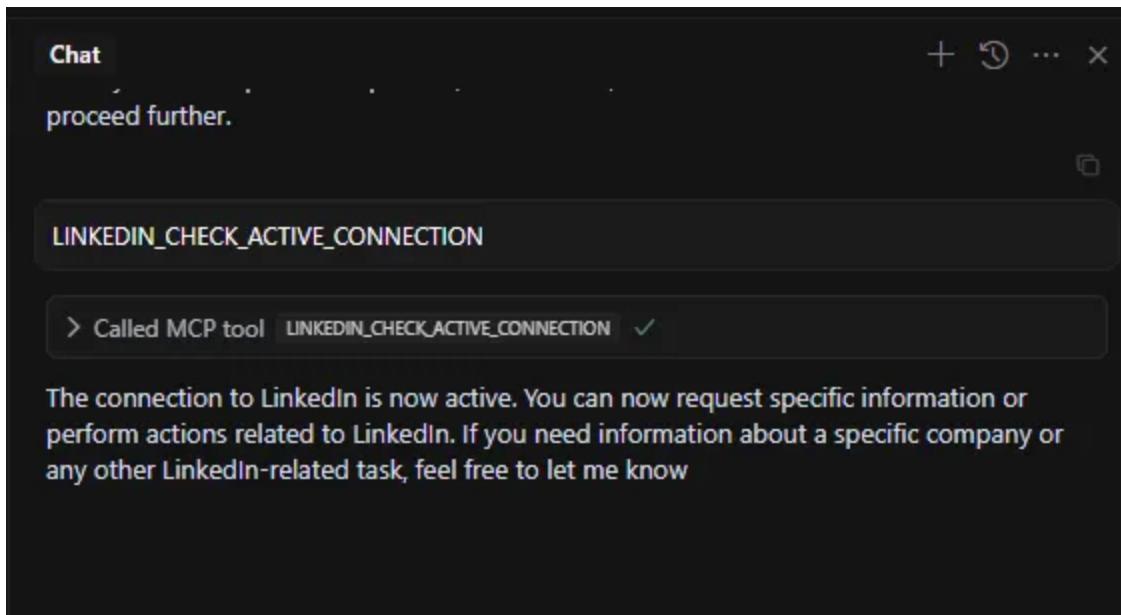


You are now authenticated. You can close
this window and go back.

Give your agents access to real world



You can also check that based on the actions of the server. As you can see,
there is an active connection.



I normally wouldn't recommend using it on your official accounts because you can never be too careful.

With this MCP server, you get the options like `get the info of profile`, `create post`, `get company info` and `delete a post`.

Autonomously reverse engineer apps using Ghidra MCP Server.

This MCP server allows LLMs to autonomously reverse engineer applications. It exposes numerous tools from core Ghidra functionality to MCP clients.

This includes decompiling and analyzing binaries, automatically renaming methods and data, and listing methods, classes, imports and exports.

A couple of use cases:

- Automated vulnerability analysis using LLMs.

- Reverse engineering of malware samples.

Here is the demo.

The [GitHub Repository](#) has 4k stars.

Read and modify Figma designs programmatically

There have been recent developments in generating the Figma board into production-ready apps.

This project implements the same thing using MCP integration between Cursor AI and Figma, allowing Cursor to communicate with Figma to read

designs and modify them programmatically.

It allows for document & selection, annotations, creating elements, styling, layout and much more.

You can simply say: `design a modern-looking signup screen for mobile` and it will create it without you interacting with the Figma file.

Here is the demo.

It has 3.1k stars on GitHub.

You can check the [GitHub Repository](#) and [official Tweet](#).

There is another great [MCP server](#) (with 5k stars on GitHub which provides Figma layout information to AI coding agents.

Create 3D scenes using Blender MCP

Creating 3d stuff has always scared a lot of builders due to the complexities involved.

This connects Blender to Claude AI through the Model Context Protocol (MCP), allowing Claude to directly interact with and control Blender.

This integration enables prompt assisted 3D modeling, scene creation and manipulation. You can watch the [complete tutorial](#) if you're interested in using it.

Prompt examples with demo videos:

⚡ “Create a low poly scene in a dungeon, with a dragon guarding a pot of gold”

⚡ “Get information about the current scene, and make a threejs sketch from it”

⚡ “Create a beach vibe using HDRIs, textures, and models like rocks and vegetation from Poly Haven”

If you’re interested in more demos, the founder created a [thread on X with wild examples](#) of what others have created.

The [GitHub Repository](#) has 10.3k stars on GitHub.

8. Some limitations of MCP.

MCP expectations and reality can be very different. You will understand what I mean as you go through the points.

MCP expectations

Personal agents
in a few clicks

Hot-swappable
model capabilities

Figma, Linear,
Slack = besties

Real security allows
enterprise use

Devs 10x'ed to
build better agents

Jarvis from
Iron man

MCP reality

Tricky to set up
existing servers

Only available in
dev tools

Doesn't work well
with most models

Not many official
integrations

Protocol hard to
learn and integrate

Tinky Winky from
Teletubbies

Credit goes to Builder.io team

Don't get me wrong, MCP is very promising but these are some limitations you should be aware of:

⚡ Not all AI platforms support MCP.

Claude (especially with its desktop app) and tools like Cursor or Windsurf support MCP directly. But if you're using something like ChatGPT or a local LLaMA model, it might not work out of the box.

There are some open source tools trying to solve this, but until MCP becomes more widely adopted, support across all AI assistants is hard.

⚡ Agent autonomy is not perfect.

MCP gives the ability but the judgment by AI is still not perfect.

For example, tool use depends on how well the model understands tool descriptions and usage context. It often needs prompt tuning or agent-side logic to improve reliability.

Performance Overhead.

Using tools through MCP adds overhead. Each call is external and can be much slower than the AI just answering on its own. For example, scraping data from a webpage through an MCP tool might take seconds, while the model could have guessed the answer from training data in milliseconds.

Now, if you are orchestrating multiple tools, the latencies add up, like calling 5 different MCP servers in sequence to:

- Fetch a file from Google Drive
- Summarize the content using an LLM tool
- Translate the summary
- Generate a tweet based on the translation
- Schedule it using a social media tool like Buffer

That chain might take 10–15 seconds, depending on server response times.

Some agents can handle parallel tool use so you can further optimize the process.

The trust issue.

Letting AI perform real actions can feel risky. Even if the AI usually gets it right, users often want to review things before they happen.

Right now, most tools are either fully autonomous or not at all. There's rarely a middle ground where AI can leverage autonomy but still give control to the user when it matters. We all need a `human in the loop`.

✗ **Bad approach:** The AI sends an email instantly without asking.

✓ **Better approach:** The AI says, I'm about to email X with this message, is it okay to send? and only acts after you approve.

⚡ **The problem of scalability.**

Most MCP servers today are built for single users, often just running on a developer's laptop.

One MCP server serving multiple independent agents or users has not been much explored yet. To do that, companies need to handle more complex stuff like concurrent requests, separate data contexts and enforce rate limit usage.

This is an area where the ecosystem still has room to grow, especially with ideas like MCP gateways or enterprise-ready MCP server frameworks.

⚡ **Security standards.**

MCP doesn't come with built-in authentication or authorization.

Authentication & Authorization : MCP doesn't have built-in support for authenticating users or agents. If you expose an MCP server over a network, you have to add your own security.

Some implementations use OAuth 2.1 to add permission scoping (read-only or write-only access), but there's currently no standard approach, so each server handles auth differently.

Correct Permissions : Ideally, agents should only use the tools they need. But if multiple powerful tools are available (like browser access and terminal), nothing stops the AI from using the wrong one, unless you manually disable it.

Prompt Injection : AI can make mistakes if it misunderstands a prompt. Worse, someone could craft a malicious prompt to trick the AI into doing something harmful (prompt injection). The safeguards depend on how each MCP server is built.

If you want to understand how to mitigate security risks in MCP implementations, read this:

- Understanding and mitigating security risks in MCP implementations by Microsoft.
- The Security Risks of MCP by Pillar.

MCP is still new. There will be further developments to resolve more edge cases as needs are discovered.

On the AI model side, we will likely see models that are fine-tuned for tool use and MCP specifically. Anthropic already mentioned future AI models optimized for MCP interaction.

Here are some nice resources if you're planning to build MCP:

- [mcp-chat](#) — is a CLI chat client for MCP servers. Used for testing & evaluating MCP servers and agents
- [mastra registry](#) — a collection of MCP server directories to connect AI to your favorite tools.
- [smithery.ai](#) — extend your agent with 4,630 capabilities via MCP servers. A lot of details include monthly tool calls, local option, tools, API, installation instructions for different clients.
- [Popular MCP Servers directory by official team](#) — 20k stars on GitHub.
- [Cursor directory](#) of 1800+ MCP servers.
- [Those MCP totally 10x my Cursor workflow](#) — YouTube video with practical use cases.

MCP is still evolving but its core ideas are here to stay and I've tried my best explaining the concepts. I hope you found something useful.

One single conversation with an Agent can help you automate complex workflows.

Now go build something wild with MCP and show the world.

Have a great day! Until next time :)

Check out more about me at anmolbaranwal.com.

You can connect me on [GitHub](#), [Twitter](#) & [LinkedIn](#).

If you loved the content, *please clap :)*

Keep learning.

Mcp Server

Ai Agent

AI

Tutorial

Programming



Published in Level Up Coding

274K followers · Last published 2 days ago

Follow

Coding tutorials and news. The developer homepage gitconnected.com && skilled.dev && levelup.dev



Written by Anmol Baranwal

1.3K followers · 60 following

Follow



Just a tech guy who loves writing about technical stuff (1.5M+ reads). Open Source developer. All my blogs are free to read. Work with me:
[hi@anmolbaranwal.com!](mailto:hi@anmolbaranwal.com)

Responses (27)



Bgerby

What are your thoughts?

See all responses

More from Anmol Baranwal and Level Up Coding



The guide to building MCP Agents

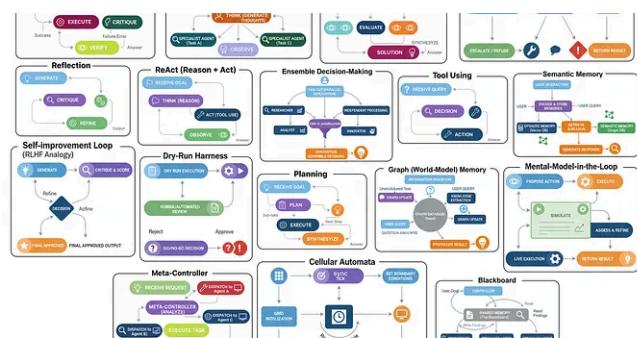
In Level Up Coding by Anmol Baranwal

The complete guide to building MCP Agents

MCP agents can now talk to real apps and actually get stuff done.

Jun 11 932 10

...



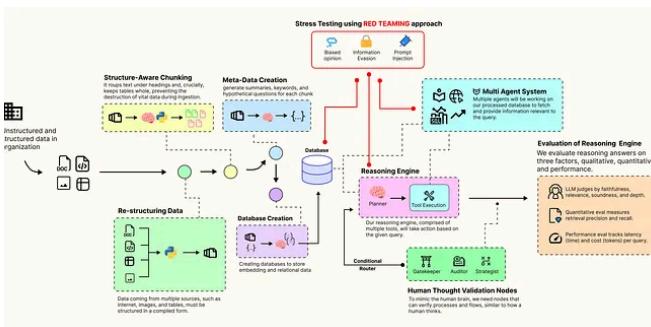
In Level Up Coding by Fareed Khan

Building 17 Agentic AI Patterns and Their Role in Large-Scale AI...

Ensembling, Meta-Control, ToT, Reflexive, PEV and more

Sep 25 1.8K 40

...



In Level Up Coding by Fareed Khan

Building an Advanced Agentic RAG Pipeline that Mimics a Human...

Ambiguity Checks, Multi-Tool Planning, Self-Correction, Causal Inference and more.

Sep 16 1.3K 19

...

In Level Up Coding by Anmol Baranwal

The guide to OpenAI Codex CLI

Hands-on review of the most powerful AI agent for your terminal

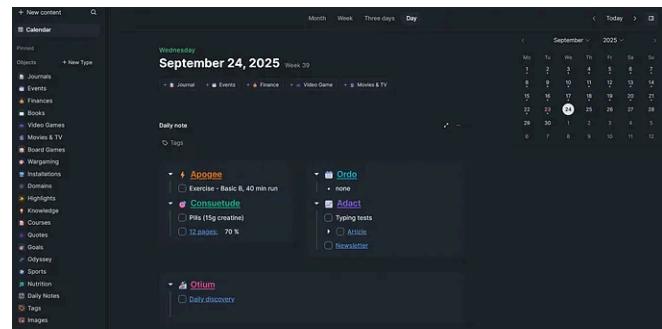
Jul 7 227 4

...

See all from Anmol Baranwal

See all from Level Up Coding

Recommended from Medium





In Stackademy by Blend Visions

5 Best MCP Servers for Effortless Vibe Coding in 2025 🚀

The time has come for you to maximize your coding process through enhanced...



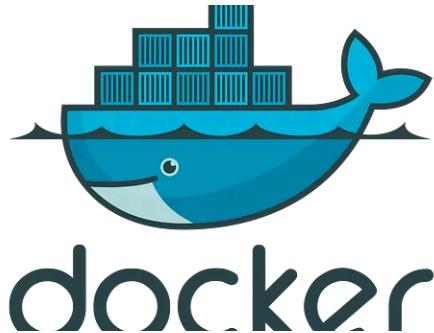
Apr 27

979

12



...



Abhinav

Docker Is Dead—And It's About Time

Docker changed the game when it launched in 2013, making containers accessible and...



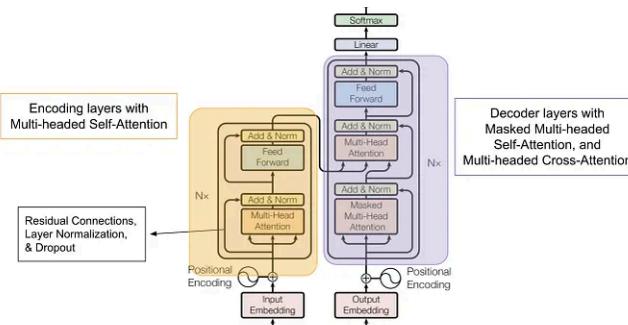
Jun 8

6.7K

187



...



In Towards AI by Ashish Abraham

No Libraries, No Shortcuts: LLM from Scratch with PyTorch

The no BS guide to build, train, and fine-tune a Transformer architecture from scratch



Tosny

7 Websites I Visit Every Day in 2025

If there is one thing I am addicted to, besides coffee, it is the internet.



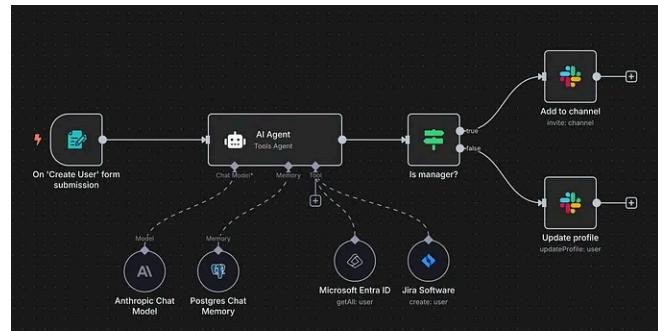
Sep 23

2.4K

84



...



In Coding Nexus by Code Coup

10 n8n Tricks Beginners Wish They Knew Earlier

Everyone loves saying n8n is beginner-friendly. Honestly? Not really.



Sep 30

216

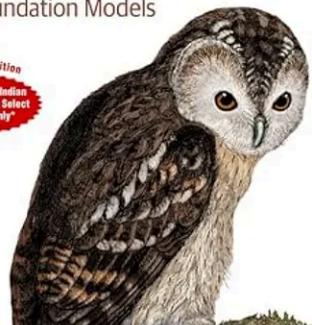
5



...

with Foundation Models

Full Colour Edition
For Sale in the India Subcontinent & Select Countries Only*



In Android Alchemy by Prakash Sharma

Stop watching random AI courses: Complete Guide to AI Engineering

Ultimate guide for AI Engineering preparation , Six months preparation guide —

⭐ 6d ago ⚡ 601 🎙 6



•••

⭐ Jul 30 ⚡ 1.1K 🎙 30



•••

See more recommendations