

# **Build Your 1st MCP Server**

AgentCon Workshop

**Shaw Talebi - June 25, 2025**



# Who is this guy?



Physics PhD  
Applied AI Researcher  
**2018**



Data Scientist  
**2022**



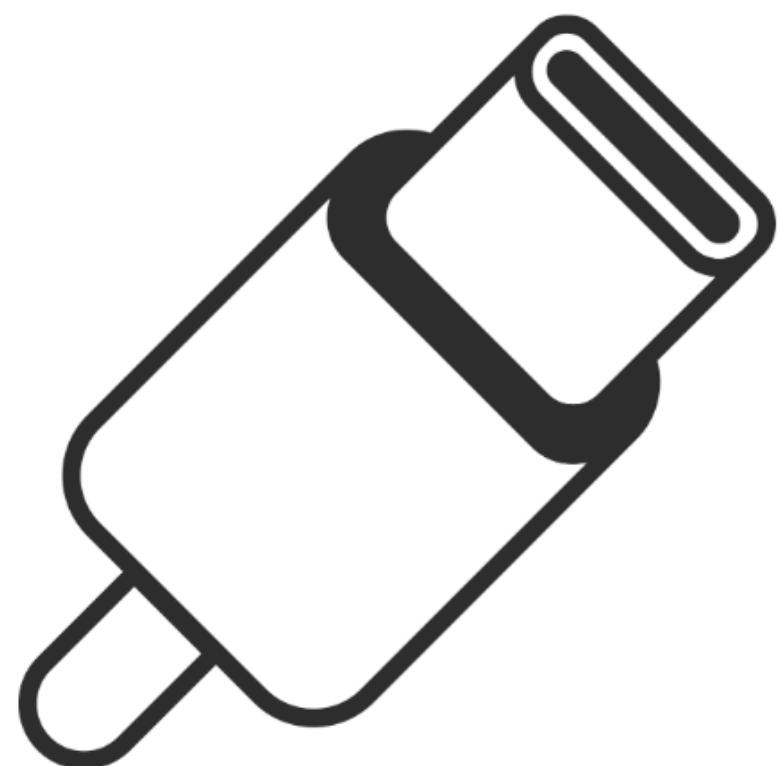
Entrepreneur  
**2023**

- Helped 100+ clients
- 18k followers
- 65k subscribers

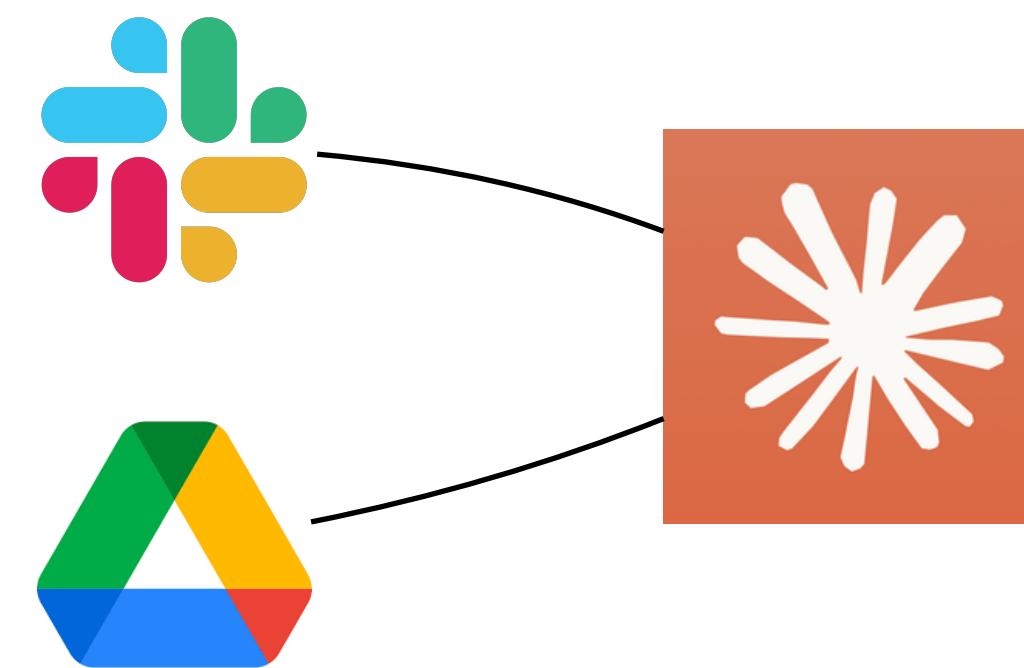


# What is MCP?

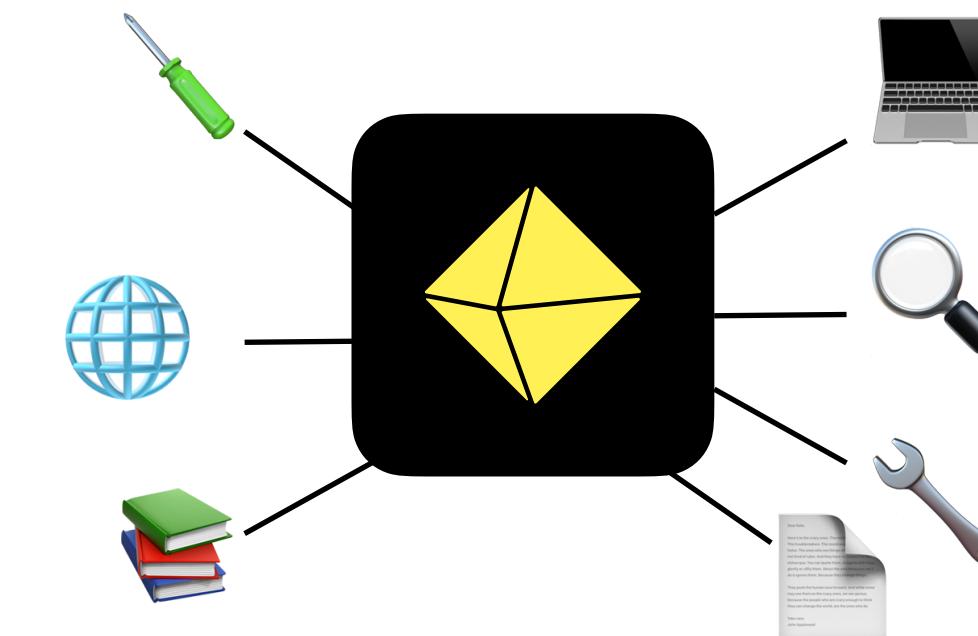
A standard way to connect tools and context to AI apps



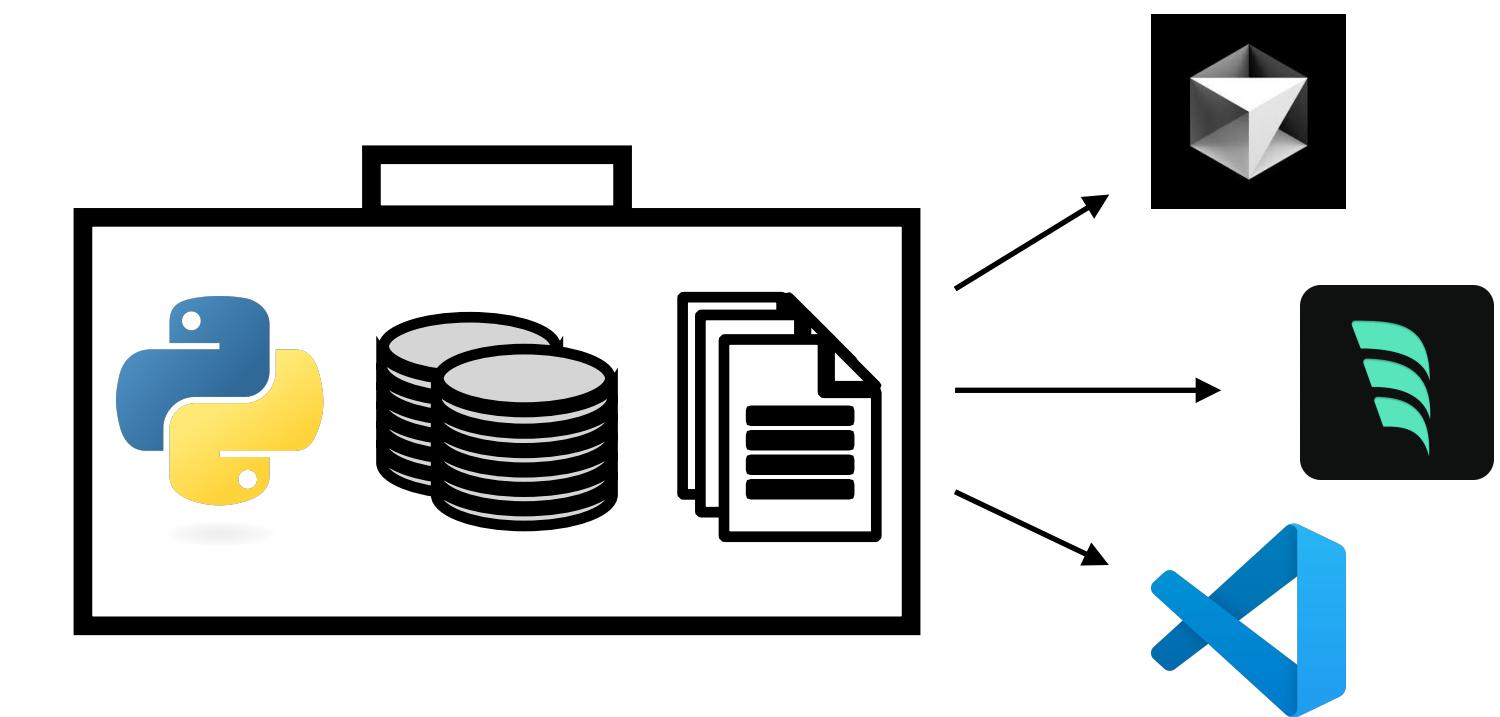
*“USB-C Port of AI Apps”*



**Portable LLM Toolset**



**Custom integrations**



**Increased App Value**

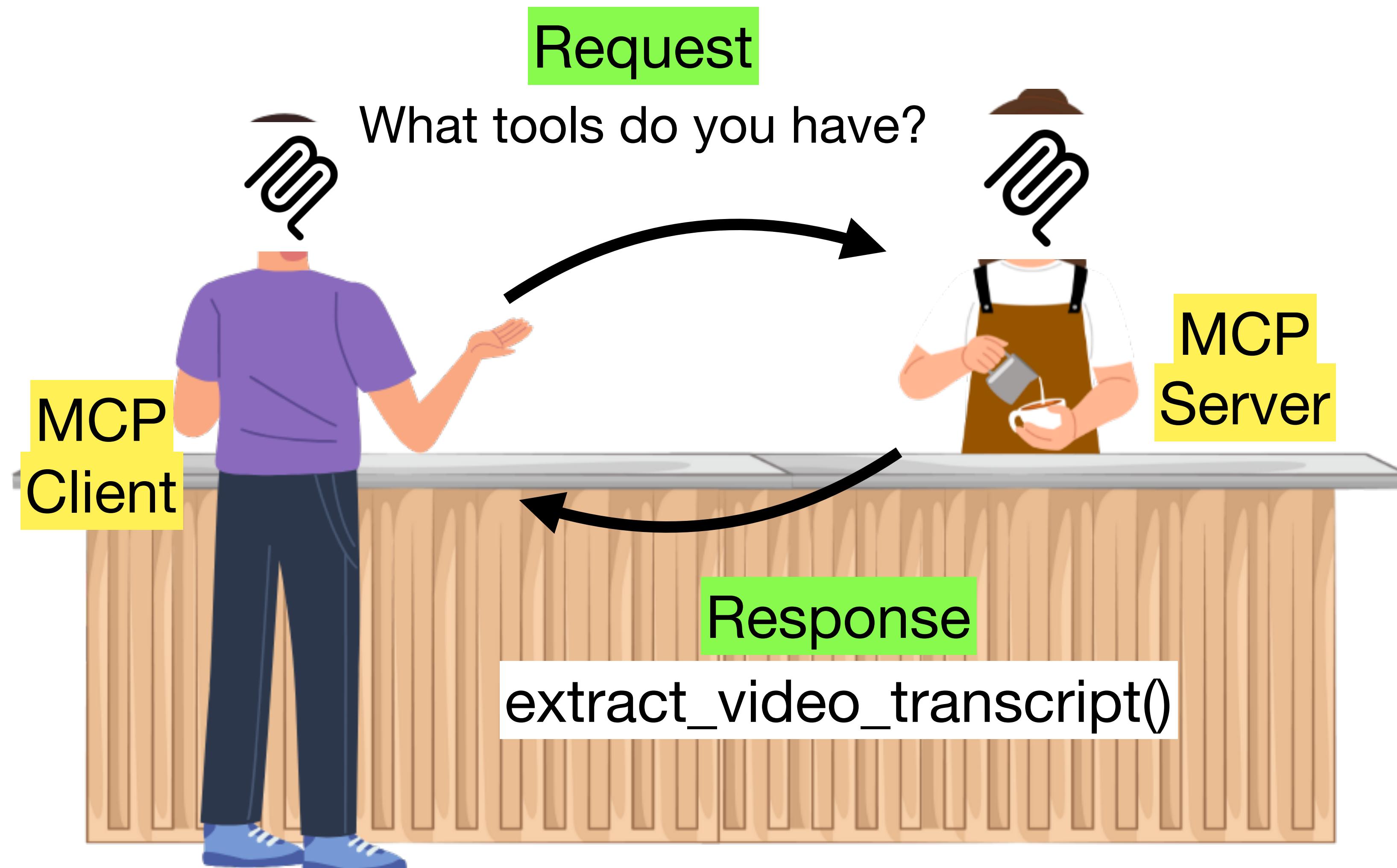
# How MCP Works

Client-server architecture



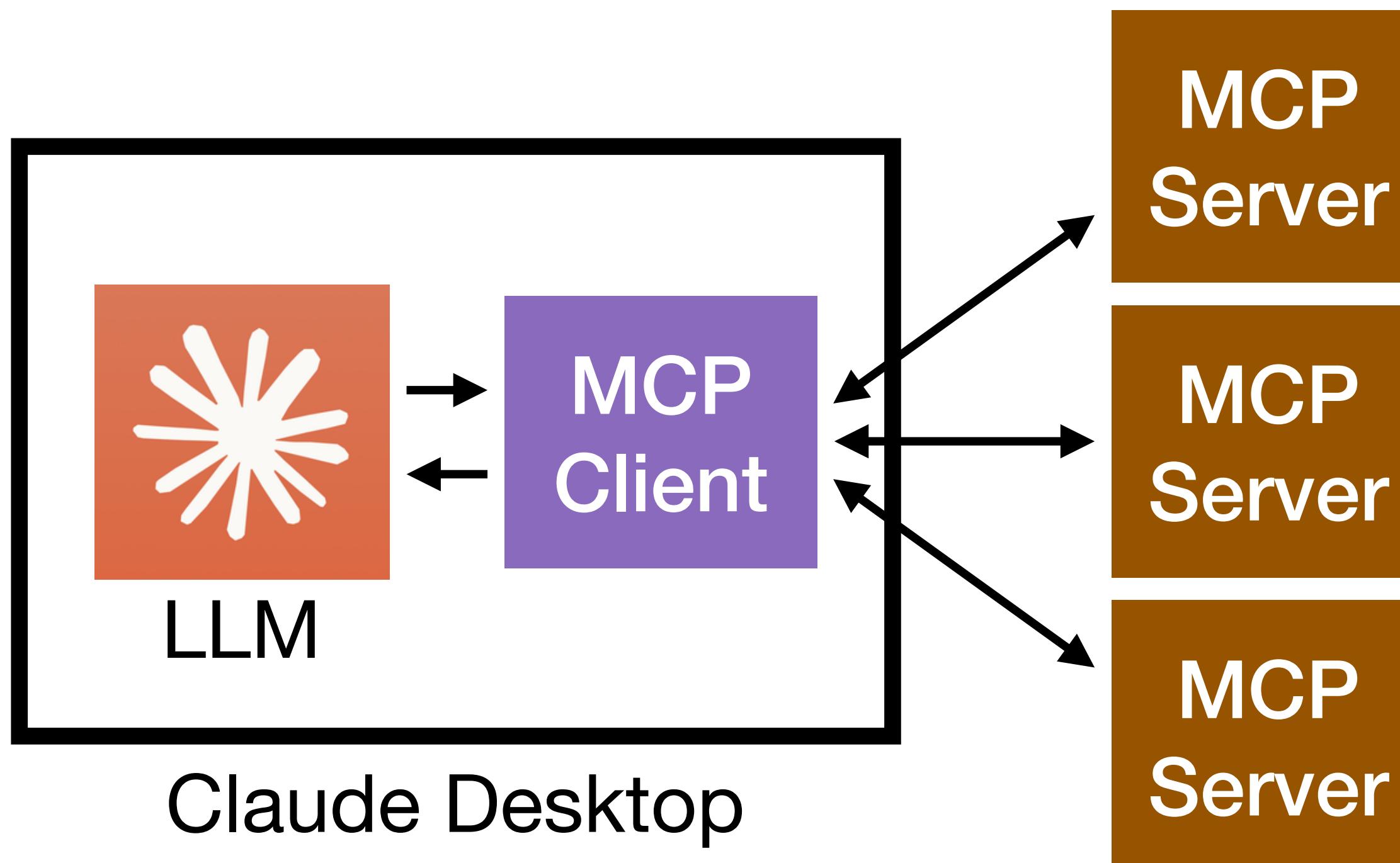
# How MCP Works

Client-server architecture



# MCP Client

Built into AI apps, sends requests to MCP servers

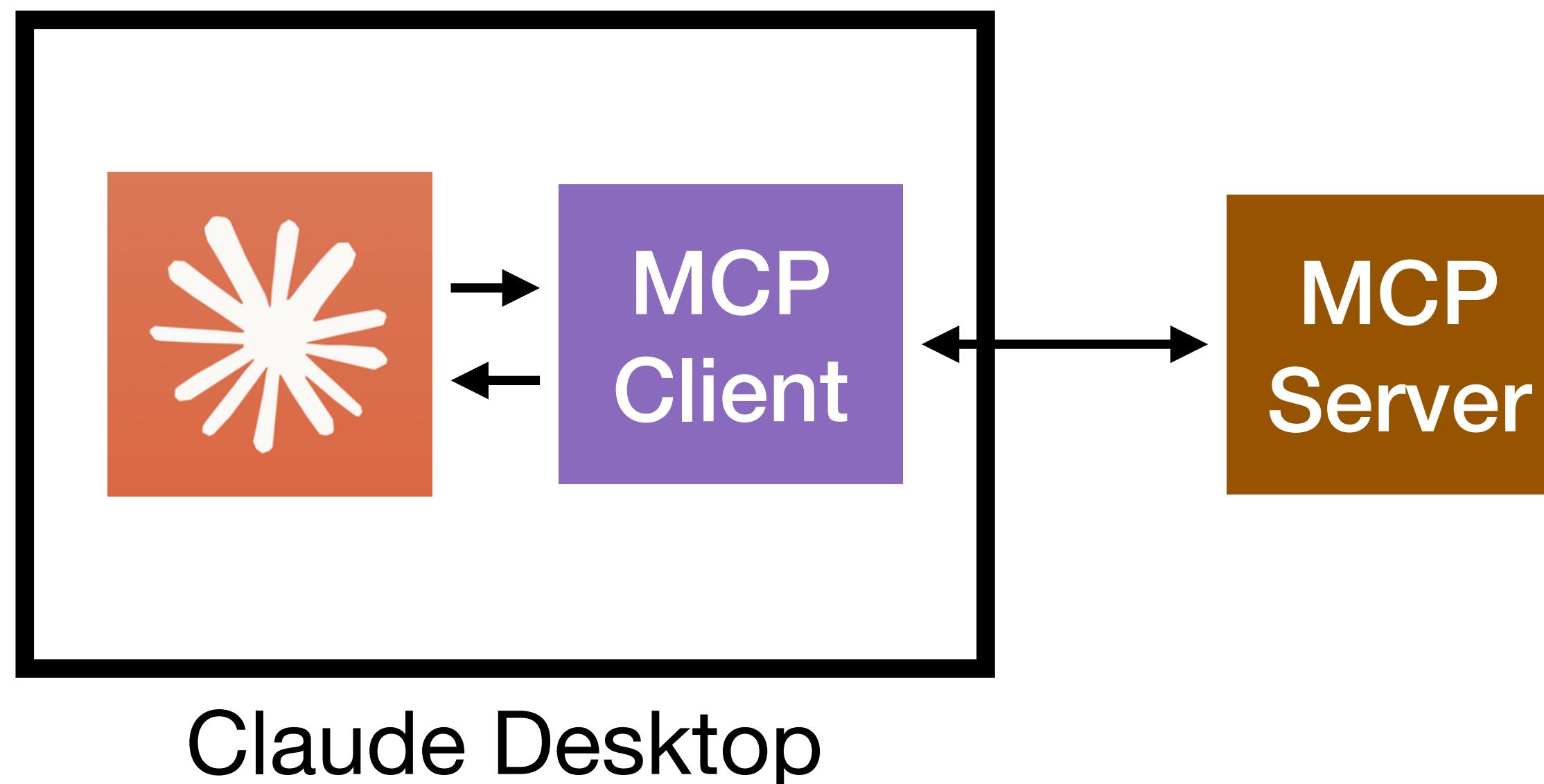


## Client Responsibilities

- 🔍 Discover server capabilities
- 📀 Receive data from servers
- 🔧 Manage LLM tool execution

# MCP Server

Listens for requests from MCP clients and responds accordingly



## 3 Key Services



**Prompt** = prompt templates



**Resource** = data, filesystem, database



**Tool** = function, API, image processing

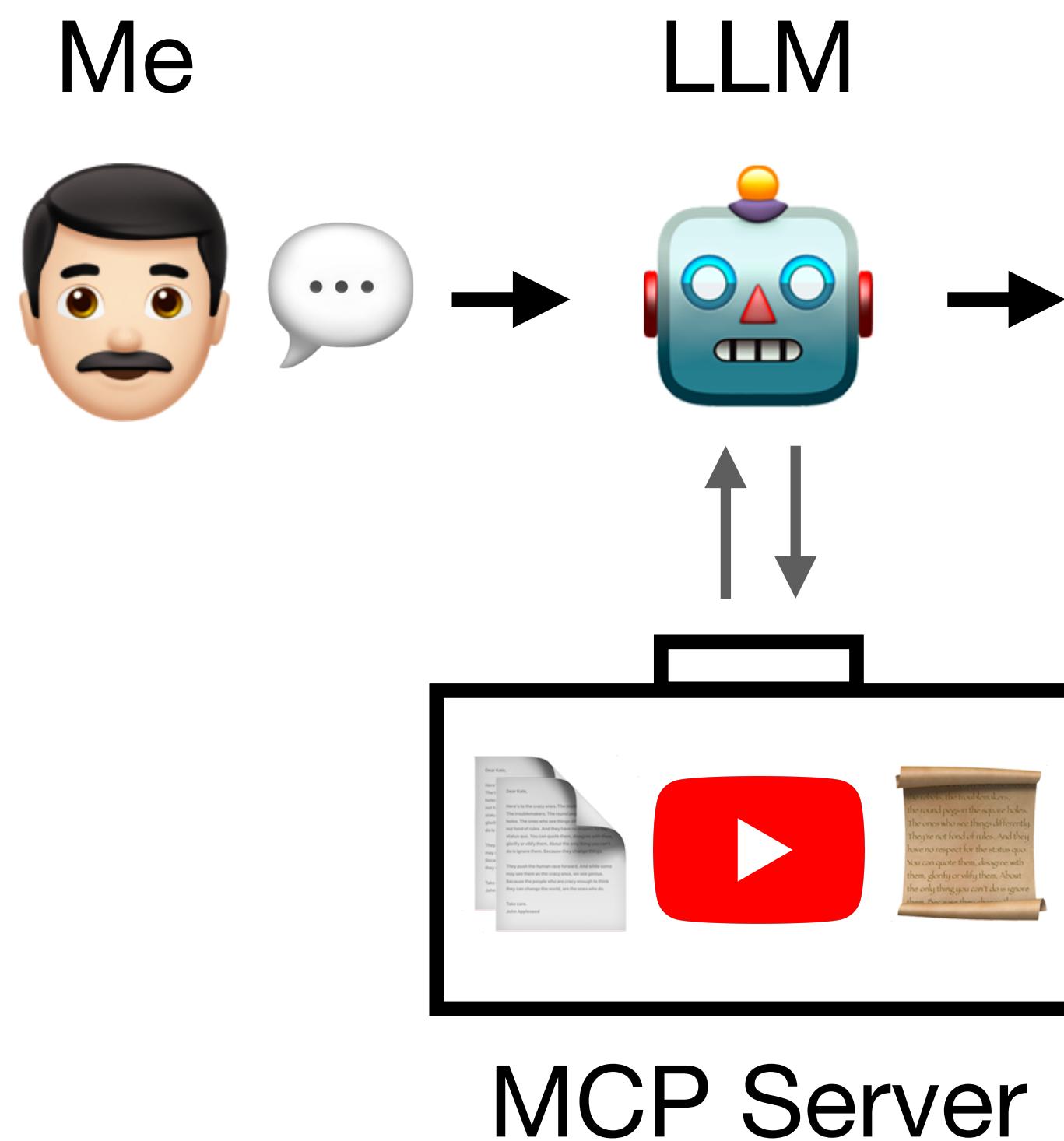
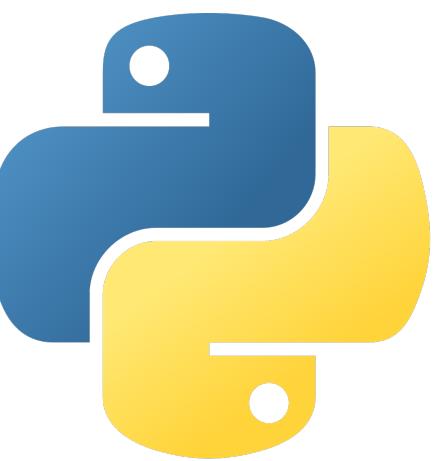
## 2 Default Transports

Stdio (local)

HTTP with Server-Sent Events (SSE) (remote)

# Example: YouTube MCP Server with Python

## Overview



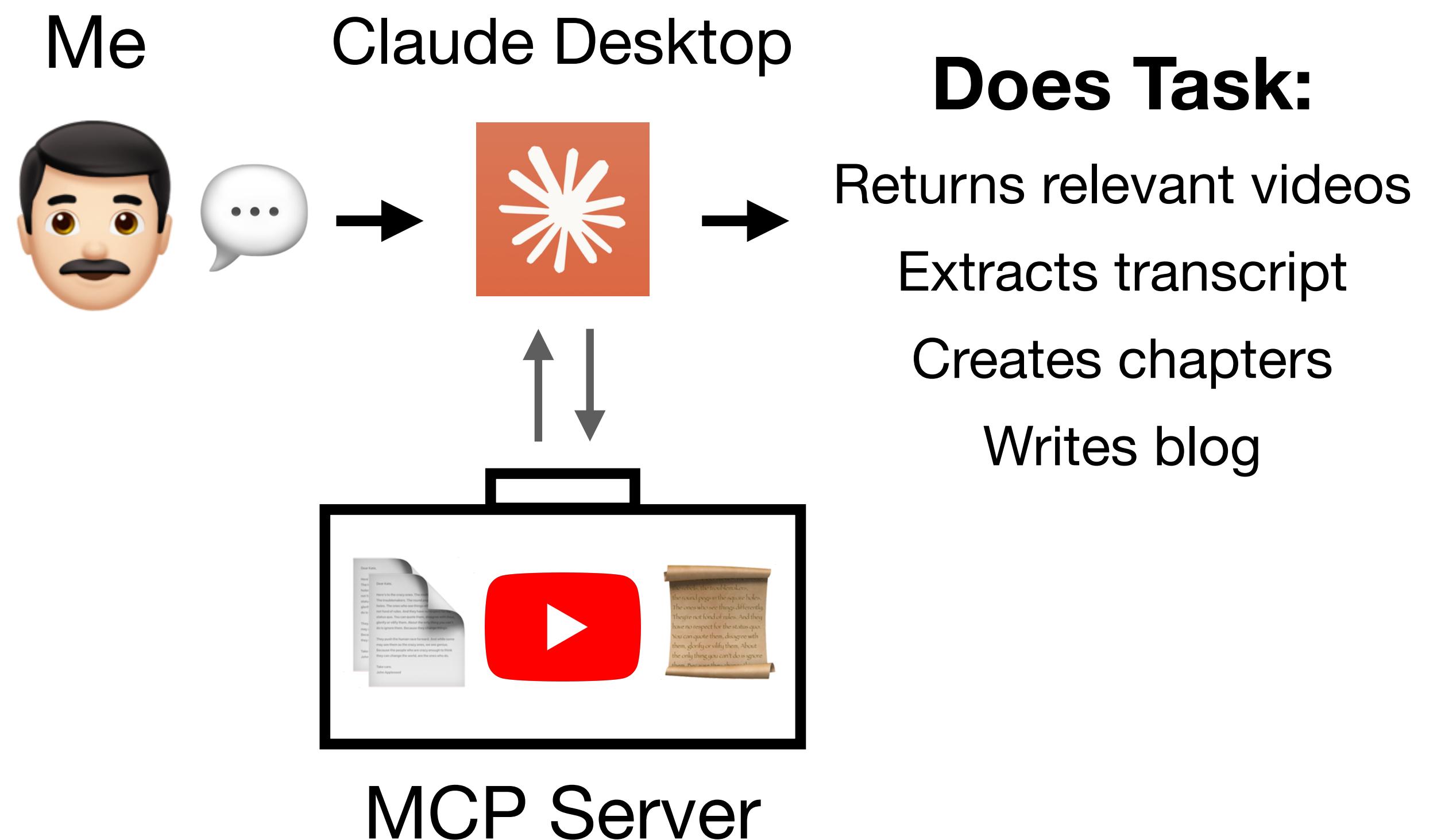
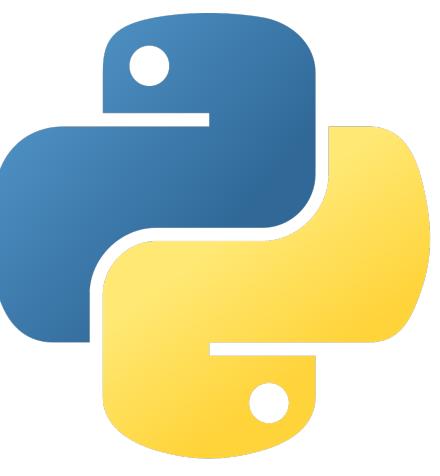
### Does Task:

- Returns relevant videos
- Extracts transcript
- Creates chapters
- Writes blog



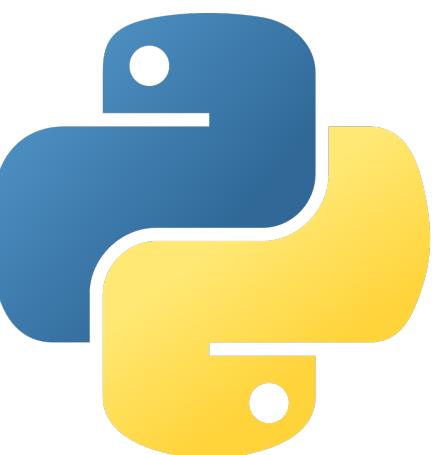
# Example: YouTube MCP Server with Python

## Overview



# Example: YouTube MCP Server with Python

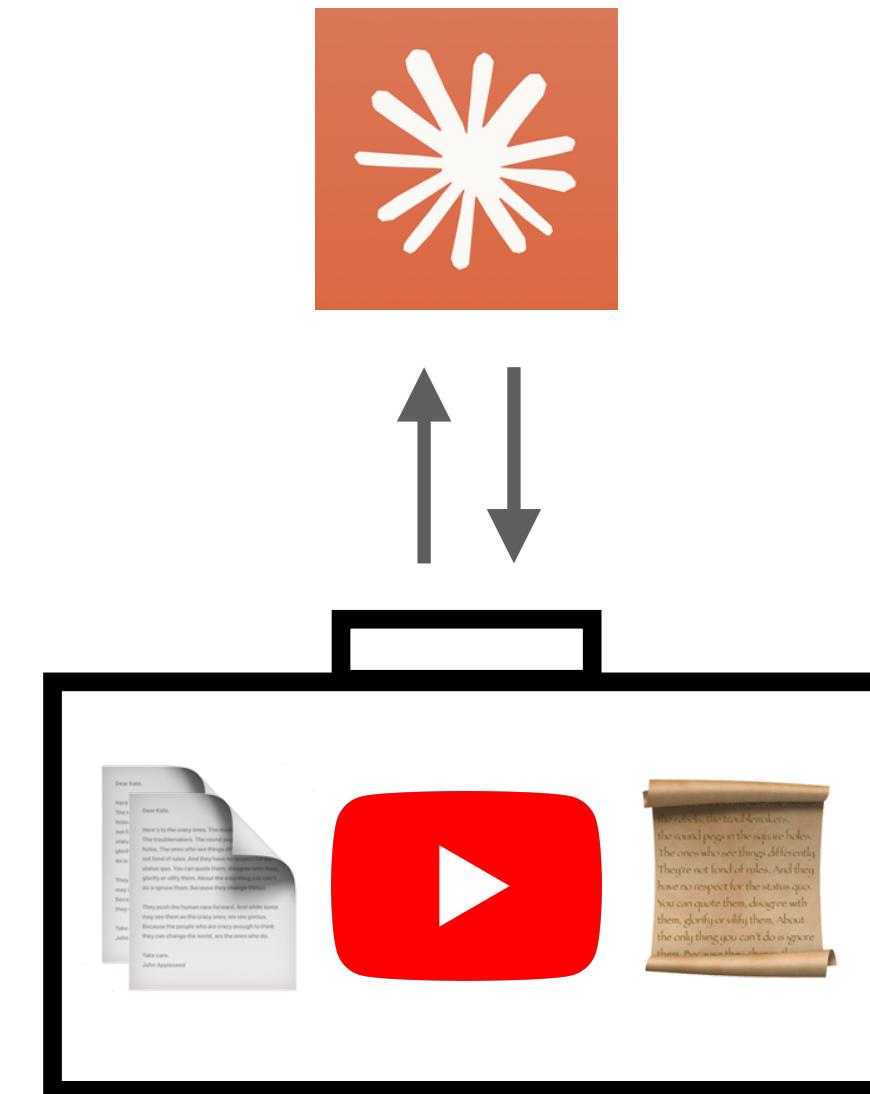
## Overview



### Key Steps:

1. Install uv
2. Create MCP Server
3. Add prompts, resources, and tools
4. Set up local transport
5. Integrate server into Claude Desktop

Claude Desktop



MCP Server



# Example: YouTube MCP Server with Python

## Step 1: Install uv



```
# Mac/Linux
curl -LsSf https://astral.sh/uv/install.sh | sh

# Windows
powershell -ExecutionPolicy ByPass -c "irm https://astral.sh/uv/install.ps1 | iex"
```



# Example: YouTube MCP Server with Python

## Step 2: Create Server

### Imports

```
from mcp.server.fastmcp import FastMCP
from youtube_transcript_api import YouTubeTranscriptApi
import csv
import re
```

### Create Server

```
# Create an MCP server
mcp = FastMCP("yt-shaw")
```

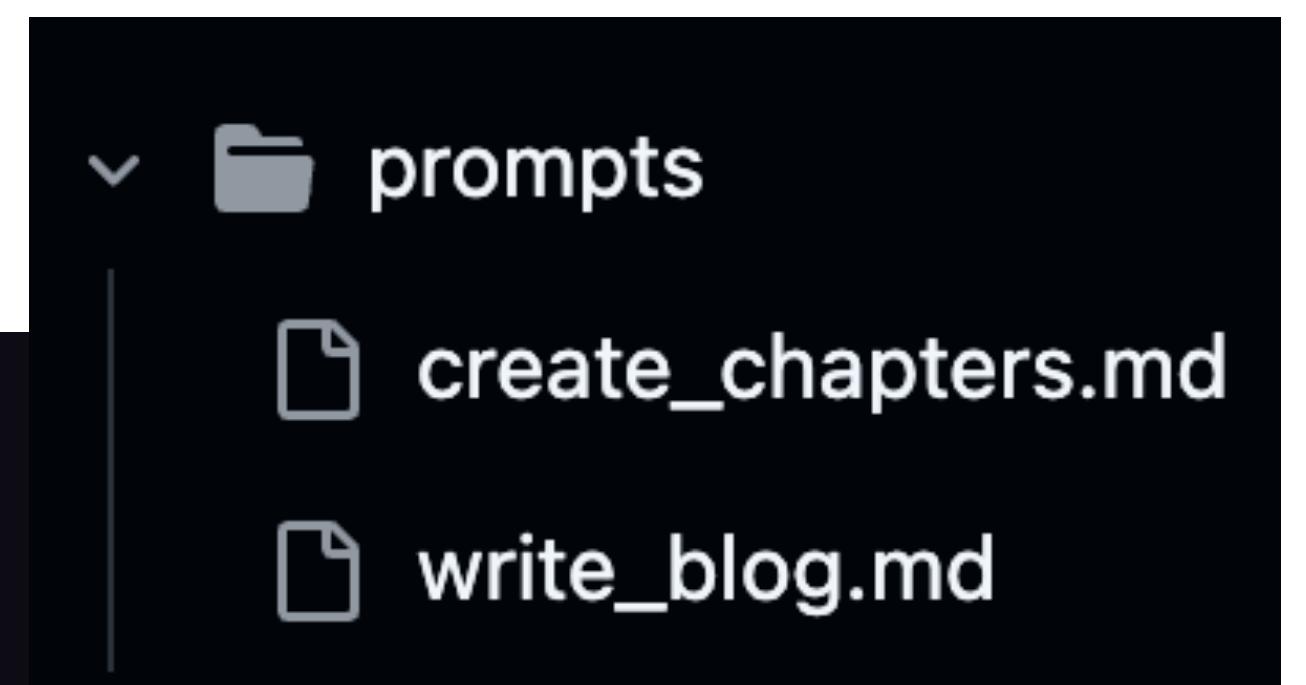


# Example: YouTube MCP Server with Python

## Step 3: Add Prompts

```
# Create prompts
@mcp.prompt()
def create_chapters_instructions() -> str:
    """Instructions for creating YouTube video chapters from a transcript."""
    with open("prompts/create_chapters.md", "r") as file:
        return file.read()

@mcp.prompt()
def write_blog_instructions() -> str:
    """Instructions for writing a blog post based on a YouTube video transcript."""
    with open("prompts/write_blog.md", "r") as file:
        return file.read()
```



# Example: YouTube MCP Server with Python

## Step 3: Add Resource

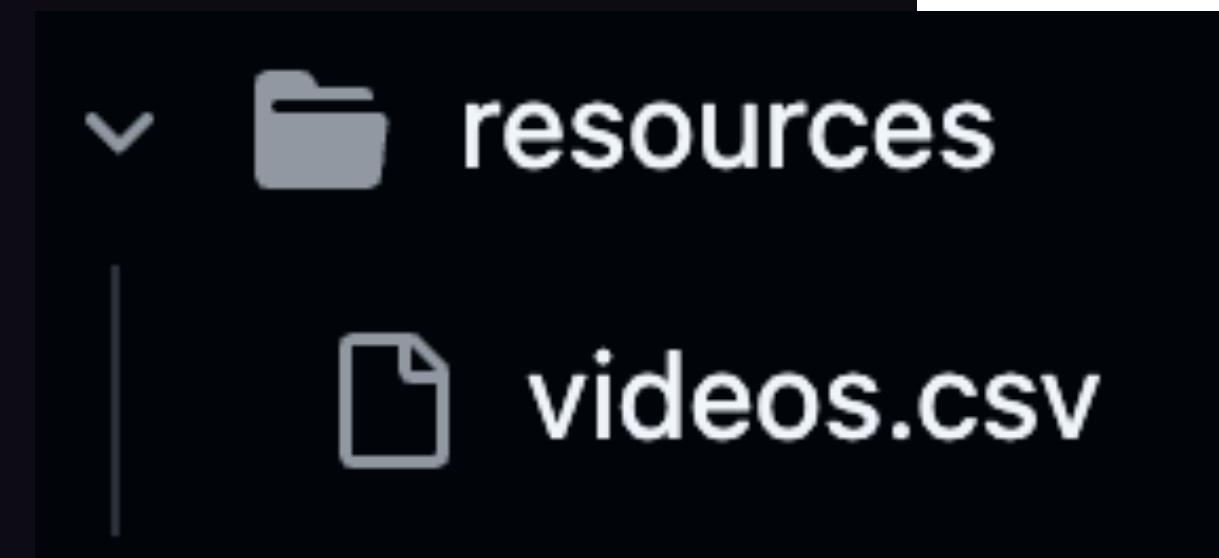
```
# Create resource
@mcp.resource("yt-library://")
def yt_library() -> str:
    """Return all the videos from Shaw Talebi's YouTube channel on AI and entrepreneurship as a Markdown table"""

    with open("resources/videos.csv", "r") as file:
        reader = csv.reader(file)
        rows = list(reader)

    if not rows:
        return "No data available."

    header = "| " + " | ".join(rows[0]) + " |"
    separator = "| " + " | ".join(["---"] * len(rows[0])) + " |"
    body = "\n".join(["| " + " | ".join(row) + " |" for row in rows[1:]])

    markdown_table = "\n".join([header, separator, body])
    return markdown_table
```



video_id	datetime	title
982V2ituTdc	2025-06-15T15:00:13Z	How to Improve LLM Apps with Error Analysis
Q2HxSfS6ADo	2025-06-08T15:00:42Z	Your LLM Prompts Suck... here's how to fix them.
nSH2vZjb2TA	2025-06-01T15:01:13Z	My PhD Defense [Applied Physics and AI Research]
nPQkBGf55YA	2025-05-25T15:01:12Z	30 AI Buzzwords Explained (in 22 Minutes)



# Example: YouTube MCP Server with Python

## Step 3: Add Tool

```
# Create tool
@mcp.tool()
async def fetch_youtube_transcript(url: str) -> str:
    """
    Extract transcript with timestamps from a YouTube video URL and format it for LLM consumption

    Args:
        url (str): YouTube video URL

    Returns:
        str: Formatted transcript with timestamps, where each entry is on a new line
        in the format: "[MM:SS] Text"
    """

```

```
# Extract video ID from URL
video_id_pattern = r'(?>v=|\\>)\\([0-9A-Za-z_-]{11})\\.*'
video_id_match = re.search(video_id_pattern, url)

if not video_id_match:
    raise ValueError("Invalid YouTube URL")

video_id = video_id_match.group(1)
```

```
try:
    transcript = YouTubeTranscriptApi.get_transcript(video_id)

    # Format each entry with timestamp and text
    formatted_entries = []
    for entry in transcript:
        # Convert seconds to MM:SS format
        minutes = int(entry['start'] // 60)
        seconds = int(entry['start'] % 60)
        timestamp = f"[{minutes:02d}:{seconds:02d}]"

        formatted_entry = f"{timestamp} {entry['text']}"
        formatted_entries.append(formatted_entry)

    # Join all entries with newlines
    return "\n".join(formatted_entries)

except Exception as e:
    raise Exception(f"Error fetching transcript: {str(e)}")
```



# Example: YouTube MCP Server with Python

## Step 4: Local Transport

```
if __name__ == "__main__":
    mcp.run(transport='stdio')
```



# Example: YouTube MCP Server with Python

## Step 5: Integrate into Claude Desktop



Settings > Developer > Edit Config

```
{  
  "mcpServers": {  
    "YT-MCP": {  
      "command": "/Users/shawhin/.local/bin/uv",  
      "args": [  
        "--directory",  
        "/Users/shawhin/Documents/repos/yt-mcp/",  
        "run",  
        "main.py"  
      ]  
    }  
  }  
}
```

**Global uv path**

**Path to MCP server source**

Demo!



<https://github.com/ShawhinT/yt-mcp>

AgentCon Dallas - June 2025

# Q&A



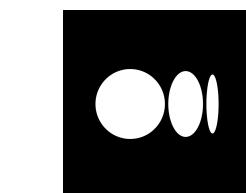
**Shaw Talebi**  
AI Educator & Builder | PhD, Physics



[@shawhintalebi](#)



[@ShawhinTalebi](#)



[@shawhin](#)

# References

- [1] [Official MCP Docs](#)
- [2] [MCP Client Development Guide](#)
- [3] [Official MCP Python SDK](#)
- [4] <https://docs.astral.sh/uv/>

