CrossComm

Empowering Your Product with Agentic AI

What To Know About Model Context Protocol

Quick Background

- Founder and CEO of CrossComm- a Durham-based emerging technology consultancy with a 25+ year history of deploying new technologies- apps, immersive reality, IoT devices, and ML- to solve real-world problems and challenges.

Web 2.0 & Twenty Years Later…

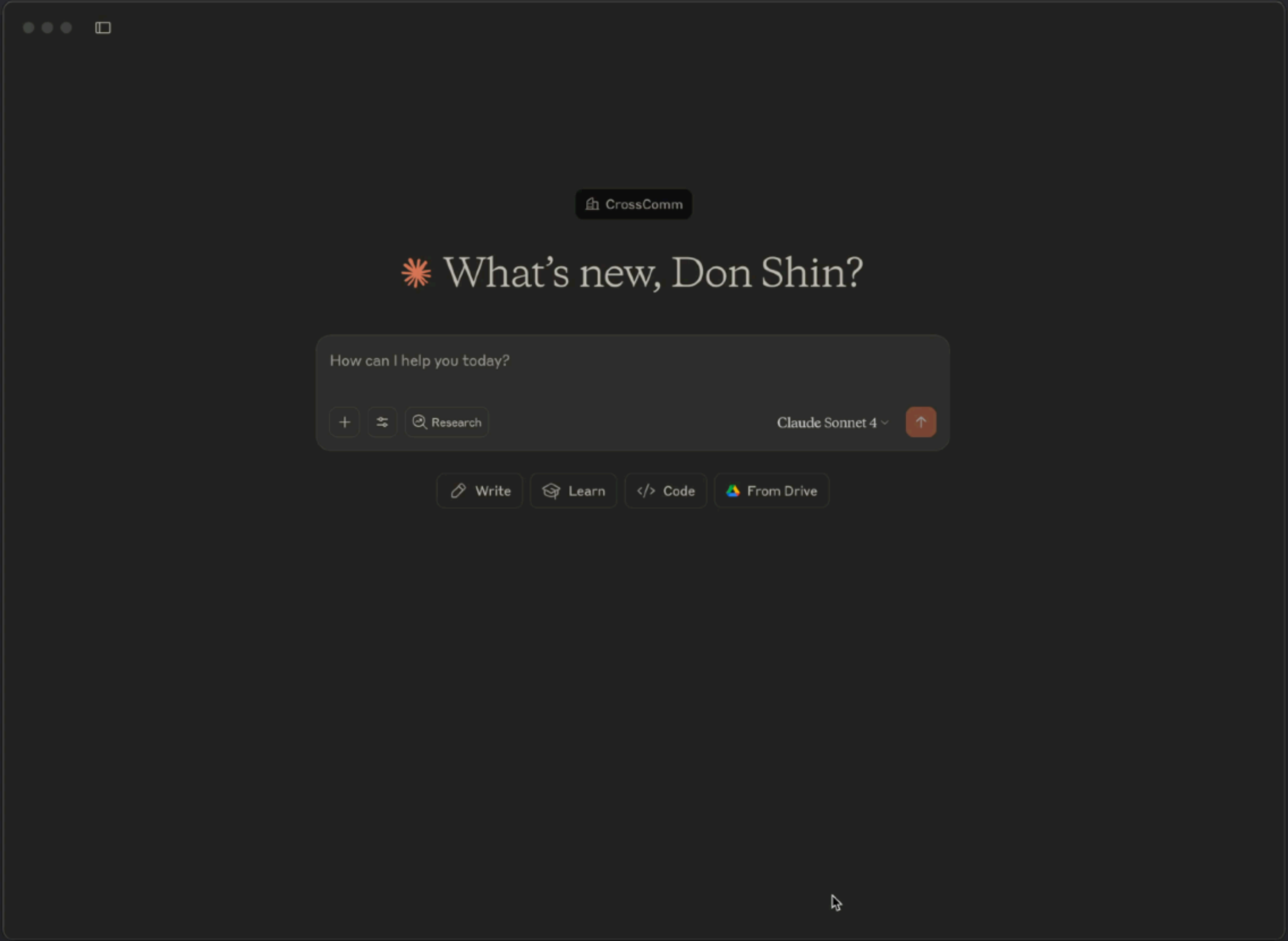
- Building early LLM applications like building pre-Web 2.0 websites (no integration standards)

- Enter Model Context Protocol (MCP) – like an API standard meant for LLMs
  - LLMs can discover and utilize resources and tools
  - LLMs can intelligent decide what to do with those resources and tools

MCP In Action

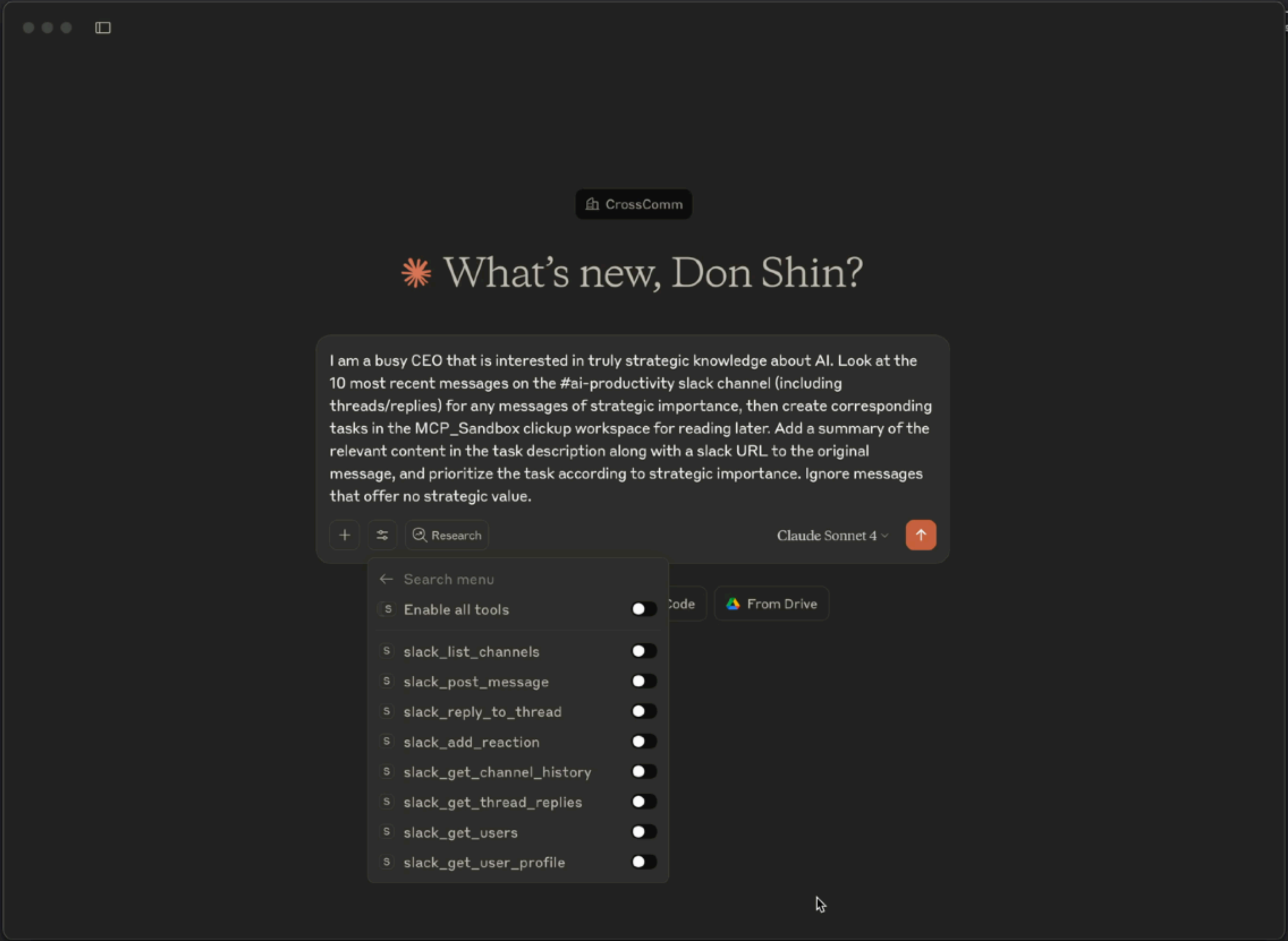
When you see it, you’ll get it

Cros


CrossComm

What's new, Don Shin?

How can I help you today?

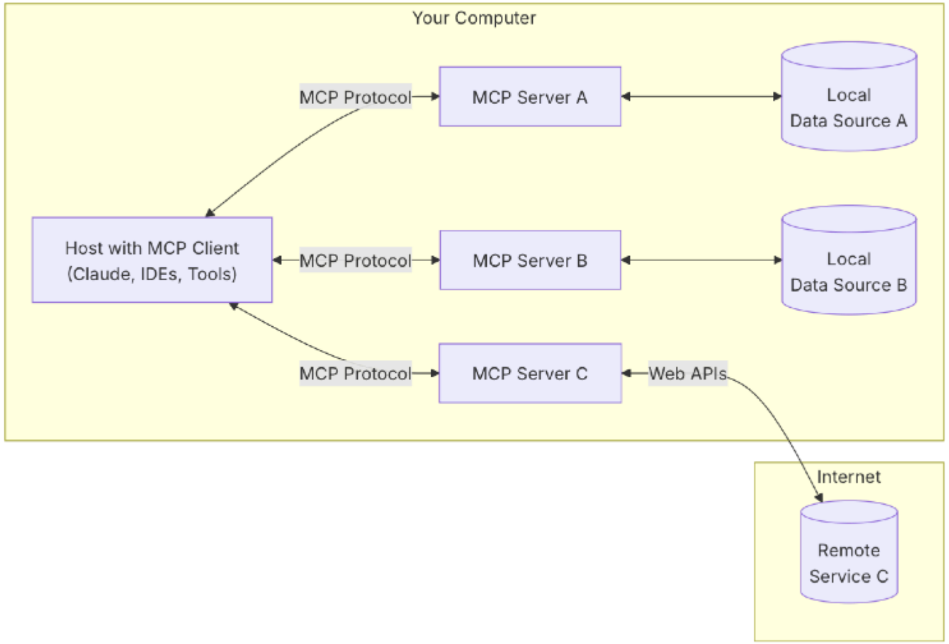


What's new, Don Shin?

I am a busy CEO that is interested in truly strategic knowledge about AI. Look at the 10 most recent messages on the #ai-productivity slack channel (including threads/replies) for any messages of strategic importance, then create corresponding tasks in the MCP_Sandbox clickup workspace for reading later. Add a summary of the relevant content in the task description along with a slack URL to the original message, and prioritize the task according to strategic importance. Ignore messages that offer no strategic value.

So What Just Happened?

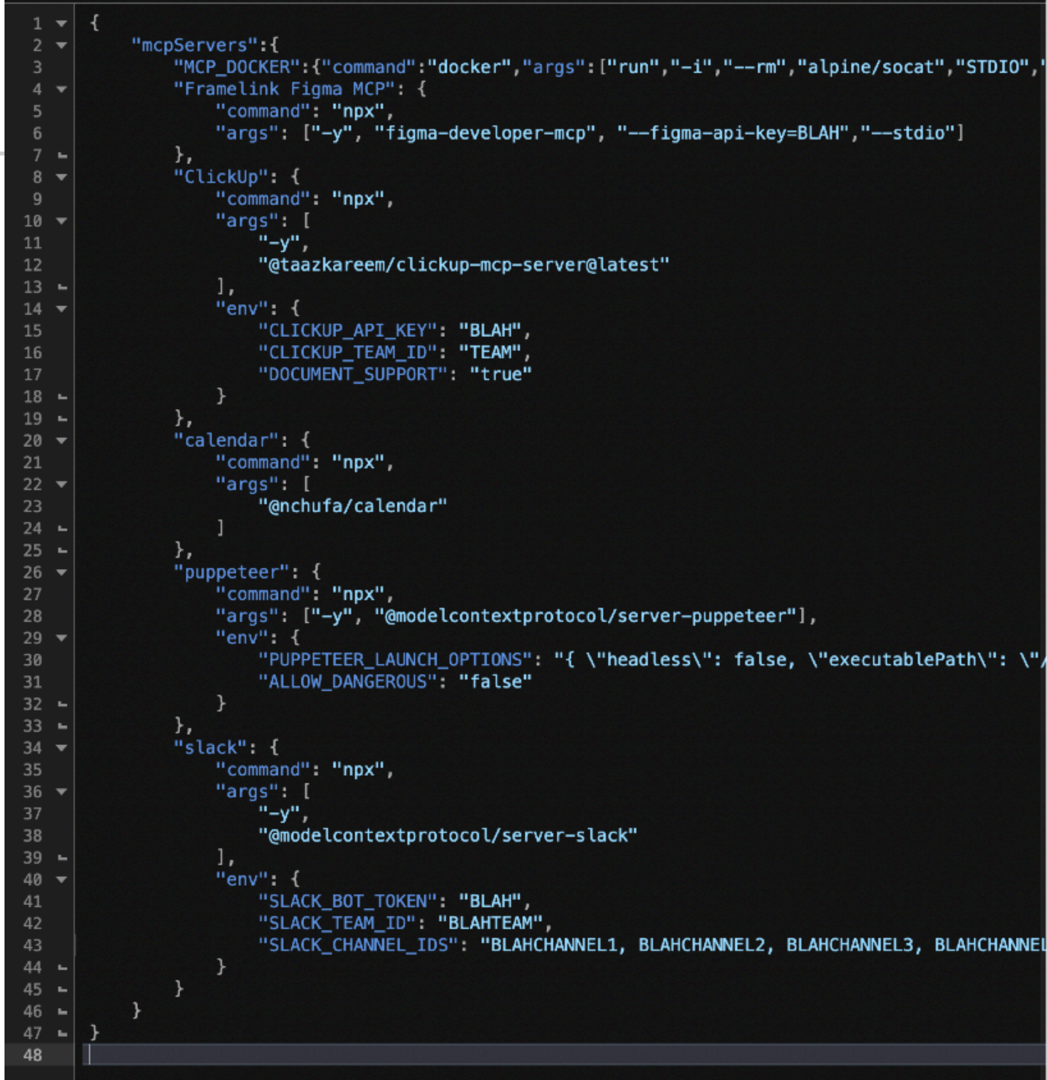
- My MCP Host App (Claude Desktop) containing an MCP Client implementation initializes connections with known MCP Servers (listed in an app configuration file, along with commands/ arguments for starting servers)
- MCP Servers are queried for names and descriptions of available resources, tools, etc.
- Host app includes tool names and descriptions as context for LLM when sending prompt to LLM Service.
- LLM Service provides response- which may involve invocation of tools.
- If tools are invoked, results are provided to LLM Service as additional context
- LLM Service provides natural language response for user



https://modelcontextprotocol.io/introduction

My Config File

- Claude Desktop's MCP Servers are listed in config file, along with commands/arguments for starting the servers



```
{
    "mcpServers":{
        "MCP_DOCKER":{"command":"docker","args":["run","-i","--rm","alpine/socat","STDIO","
        "Framelink Figma MCP": {
            "command": "npx",
            "args": ["-y", "figma-developer-mcp", "--figma-api-key=BLAH","--stdio"]
        },
        "ClickUp": {
            "command": "npx",
            "args": [
                "-y",
                "@taazkareem/clickup-mcp-server@latest"
            ],
            "env": {
                "CLICKUP_API_KEY": "BLAH",
                "CLICKUP_TEAM_ID": "TEAM",
                "DOCUMENT_SUPPORT": "true"
            }
        },
        "calendar": {
            "command": "npx",
            "args": [
                "@nchufa/calendar"
            ]
        },
        "puppeteer": {
            "command": "npx",
            "args": ["-y", "@modelcontextprotocol/server-puppeteer"],
            "env": {
                "PUPPETEER_LAUNCH_OPTIONS": "{ \"headless\": false, \"executablePath\": \"/
                "ALLOW_DANGEROUS": "false"
            }
        },
        "slack": {
            "command": "npx",
            "args": [
                "-y",
                "@modelcontextprotocol/server-slack"
            ],
            "env": {
                "SLACK_BOT_TOKEN": "BLAH",
                "SLACK_TEAM_ID": "BLAHTEAM",
                "SLACK_CHANNEL_IDS": "BLAHCHANNEL1, BLAHCHANNEL2, BLAHCHANNEL3, BLAHCHANNEL
            }
        }
    }
}
```

MCP Servers: adding new capabilities to an LLM application

- MCP Servers can offer:
  - **Resources** – typically read-only data sources (e.g. file system, logs, databases)
  - **Prompts** – prompt templates that can provide context or instruction to the LLM
  - **Tools** – like functions, tools enable MCP Client Hosts to perform actions
- MCP Servers provide endpoints for discovery of above offerings
  - `resources/list`
  - `prompts/list`
  - `tools/list`

MCP Servers: Resources

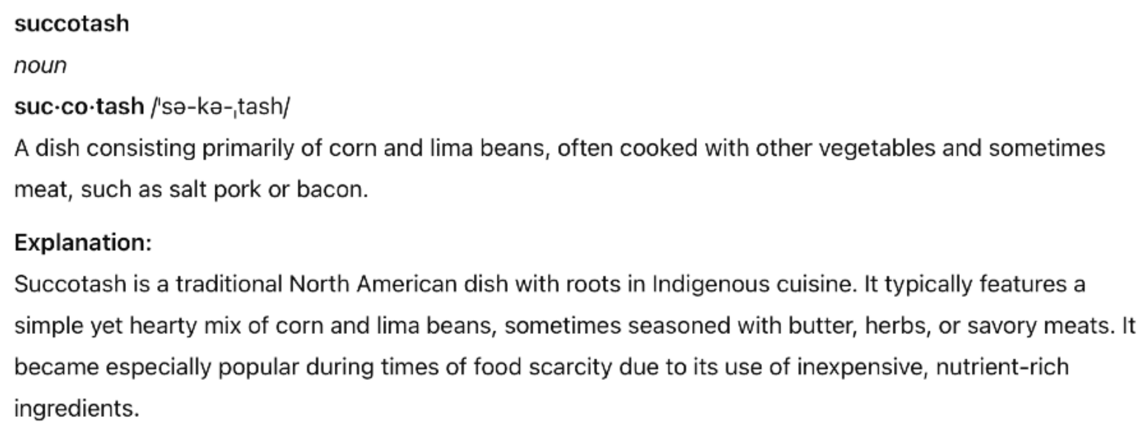
- Fixed URIs, e.g. `file:///home/user/documents/report.pdf`
- URI Templates, e.g. `github://repos/{owner}/{repo}`
- Clients can request a Resource URI from the MCP Server through the endpoint, `resources/read`
- MCP Clients can subscribe to notifications when a resource at a URI is updated, or when the list of resources from the server has changed
- Resources are ”application-controlled” – host application decides how to handle them

MCP Servers: Prompts

- Prompt Templates are like prompt shortcuts.
- Can be invoked by the user (user-controlled) through app UI:
    - Slash commands (e.g. /learnmore dancing cats)
    - Contextual menus (e.g. highlighting word, right-clicking on “Define and Explain”)
    - However the application designer chooses to expose prompt templates
- MCP Host App then leverages LLM to generate response

MCP Servers: Prompts

- "Define and Explain" prompt template example:
  - User highlights the word, "succotash", right-clicks and selects "Define and Explain" in Contextual Menu.
  - LLM is provided with the prompt:
    - Provide the definition for the word, "" as you would find it in a dictionary, and provide a 1-3 sentence explanation for it.



succotash

noun

suc·co·tash /ˈsə-kə-ˌtash/

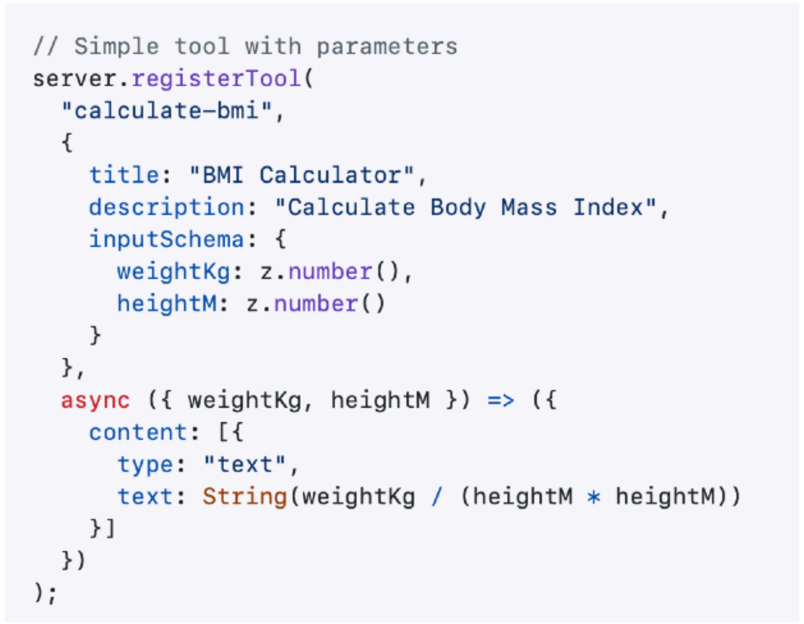
A dish consisting primarily of corn and lima beans, often cooked with other vegetables and sometimes meat, such as salt pork or bacon.

Explanation:

Succotash is a traditional North American dish with roots in Indigenous cuisine. It typically features a simple yet hearty mix of corn and lima beans, sometimes seasoned with butter, herbs, or savory meats. It became especially popular during times of food scarcity due to its use of inexpensive, nutrient-rich ingredients.

MCP Servers: Tools

- Tools enable LLM-powered applications to perform actions
  - Devs: think, function calls/methods - but for LLMs.
- Tools are model-controlled, i.e. an LLM decides to invoke them (potentially requiring end-user permission)
- Tools can be simple and locally executed…

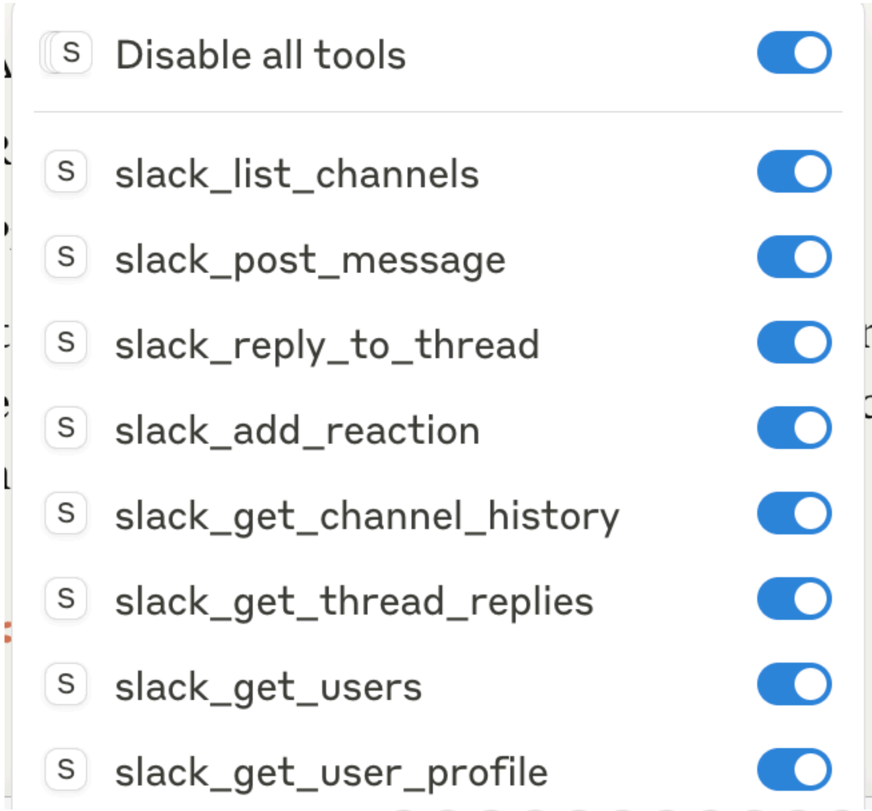


```
// Simple tool with parameters
server.registerTool(
  "calculate-bmi",
  {
    title: "BMI Calculator",
    description: "Calculate Body Mass Index",
    inputSchema: {
      weightKg: z.number(),
      heightM: z.number()
    }
  },
  async ({ weightKg, heightM }) => ({
    content: [{
      type: "text",
      text: String(weightKg / (heightM * heightM))
    }]
  })
);
```

https://github.com/modelcontextprotocol/typescript-sdk

MCP Servers: Tools

- Tools can be complex, operate across network, and change state


Disable all tools
slack_list_channels
slack_post_message
slack_reply_to_thread
slack_add_reaction
slack_get_channel_history
slack_get_thread_replies
slack_get_users
slack_get_user_profile

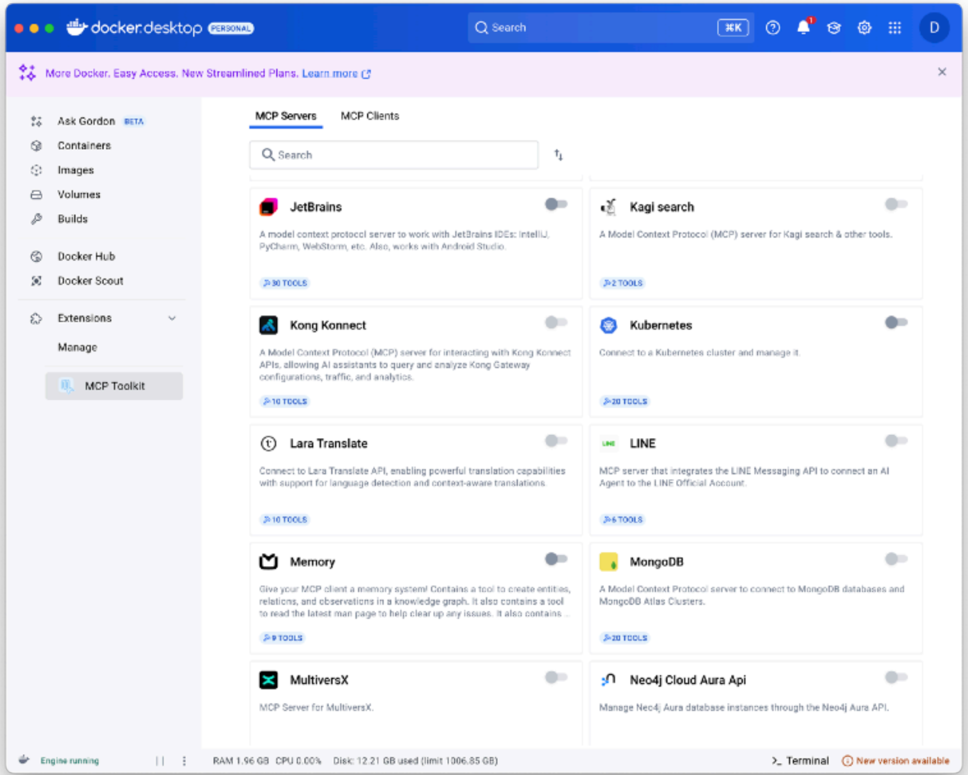
Interoperability

Product Possibilities with MCP

Interoperability

- **If you build MCP client functionality** into your LLM-powered app, you can:
  - Enable your app to leverage your customer's existing data and systems
  - Quickly add capabilities from a rapidly growing list of MCP servers to your app

Windows 11



More Docker. Easy Access. New Streamlined Plans. Learn more

JetBrains
A model context protocol server to work with JetBrains IDEs: IntelliJ, PyCharm, WebStorm, etc. Also, works with Android Studio.
30 TOOLS

Kagi search
A Model Context Protocol (MCP) server for Kagi search & other tools.
2 TOOLS

Kong Konnect
A Model Context Protocol (MCP) server for interacting with Kong Konnect APIs, allowing AI assistants to query and analyze Kong Gateway configurations, traffic, and analytics.
10 TOOLS

Kubernetes
Connect to a Kubernetes cluster and manage it.
20 TOOLS

Lara Translate
Connect to Lara Translate API, enabling powerful translation capabilities with support for language detection and context-aware translations.
10 TOOLS

LINE
MCP server that integrates the LINE Messaging API to connect an AI Agent to the LINE Official Account.
6 TOOLS

Memory
Give your MCP client a memory system! Contains a tool to create entities, relations, and observations in a knowledge graph. It also contains a tool to read the latest man page to help clear up any issues. It also contains ...
9 TOOLS

MongoDB
A Model Context Protocol server to connect to MongoDB databases and MongoDB Atlas Clusters.
20 TOOLS

MultiversX
MCP Server for MultiversX.

Neo4j Cloud Aura Api
Manage Neo4j Aura database instances through the Neo4j Aura API.

RAM 1.96 GB CPU 0.00% Disk: 12.21 GB used (limit 1006.85 GB)

New version available

Interoperability

- **If you build MCP server functionality** into your platform or backend, a growing number of MCP Client Apps will be able to connect with and leverage your platform or backend
    - Make your platform a foundational part of others' tech stack

Product Tips

For Leveraging MCP in Your Product

Product Tips For Leveraging MCP

- Add MCP client support to your LLM-powered products where high context complexity is present
- Use MCP architecture to easily and quickly add agentic tool invocation and new agentic capabilities to your UX
- If your product is a SaaS, provide great MCP server support so that your SaaS becomes an indispensable part of others' agentic workflow
- One place to start: wrap an existing API in an MCP Server to allow LLMs interface with an API
- Use MCP to future-proof and modularize your AI architecture/investment
- Bottom Line: adding agentic behavior to your apps- and encouraging the use of your platform in agentic workflows- **has never been easier with MCP.**

CrossComm

Thank You

For your time and attention!

Don Shin Don.Shin@crosscomm.com

LinkedIn: https://www.linkedin.com/in/donshin1/

BlueSky: @donshin1.bsky.social