



Model Context Protocol (MCP):
Die Zukunft von User-Interaktionen?

ABOUT ME

Jonas Bandi

jonas.bandi@ivorycode.com

Twitter: [@jbandi](https://twitter.com/@jbandi)



- Freelancer, in den letzten 12 Jahren vor allem in Projekten im Spannungsfeld zwischen modernen Webentwicklung und traditionellen Geschäftsanwendungen.
- Dozent an der Berner Fachhochschule seit 2007
- In-House Kurse & Beratungen zu Web-Technologien im Enterprise: UBS, Postfinance, Mobiliar, AXA, BIT, SBB, Elca, Adnovum, BSI ...



Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

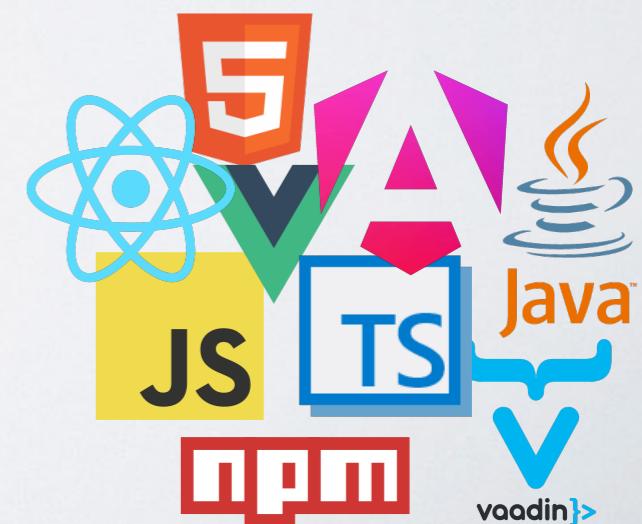


letsboot.ch
swiss dev training



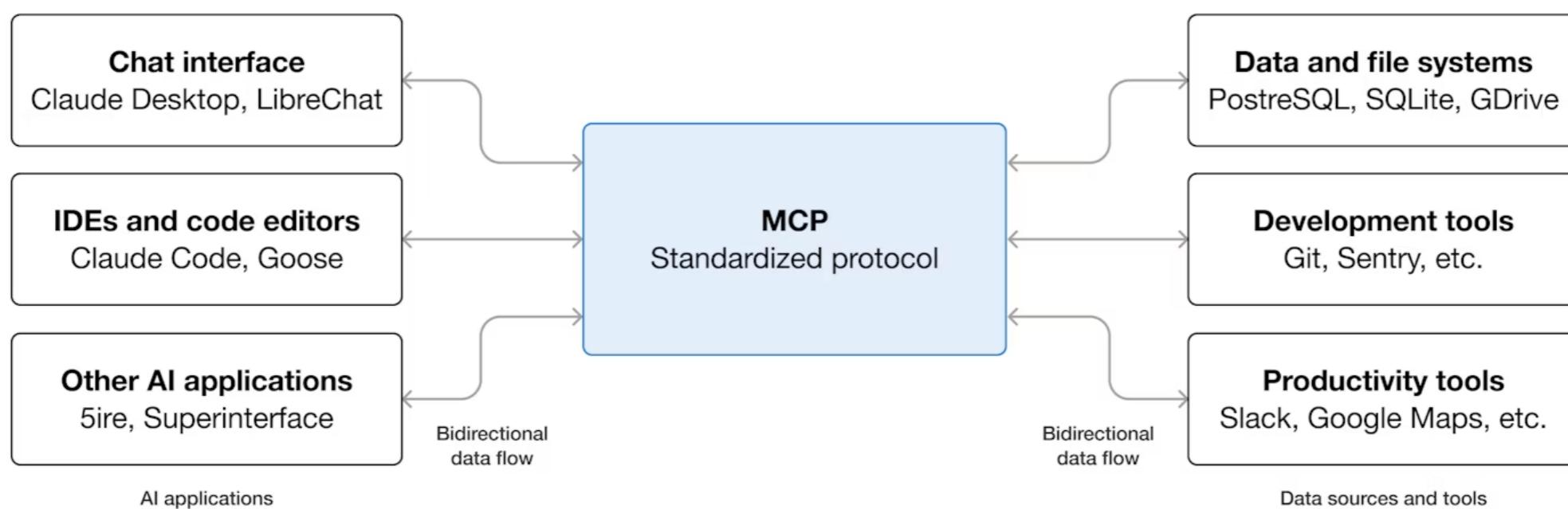
JavaScript / Angular / React / Vue / Vaadin
Schulung / Beratung / Coaching / Reviews

jonas.bandi@ivorycode.com



Who is already using MCP?
(Model Context Protocol Servers)

What is MCP?



<https://modelcontextprotocol.io/docs/getting-started/intro>

Connect your AI applications to the world

AI-enabled tools are powerful, but they're often limited to the information you manually provide or require bespoke integrations.

Whether it's reading files from your computer, searching through an internal or external knowledge base, or updating tasks in an project management tool, MCP provides a secure, standardized, *simple* way to give AI systems the context they need.



How it works

1 Choose MCP servers

Pick from pre-built servers for popular tools like GitHub, Google Drive, Slack and hundreds of others. Combine multiple servers for complete workflows, or easily build your own for custom integrations.

2 Connect your AI application

Configure your AI application (like Claude, VS Code, or ChatGPT) to connect to your MCP servers. The application can now see available tools, resources and prompts from all connected servers.

3 Work with context

Your AI-powered application can now access real data, execute actions, and provide more helpful responses based on your actual context.

MCP in AI-Assisted Coding

Settings

mcp

- Appearance & Behavior
- Editor
- Inlay Hints
- Plugins
- Tools
- AI Assistant
- Model Context Protocol (MCP)
- GitHub Copilot
- Model Context Protocol (MCP)
- Junie

reg@jonasbandi.net
Free Plan

Search settings ⌘F

General

Agents

Tab

Models

Cloud Agents

Tools & MCP

Rules and Commands

Indexing & Docs

Network

Beta

Docs

Tools

Browser

Browser Automation

Connected to Browser Tab

Show Localhost Links in Browser

Automatically open localhost links in the Browser Tab

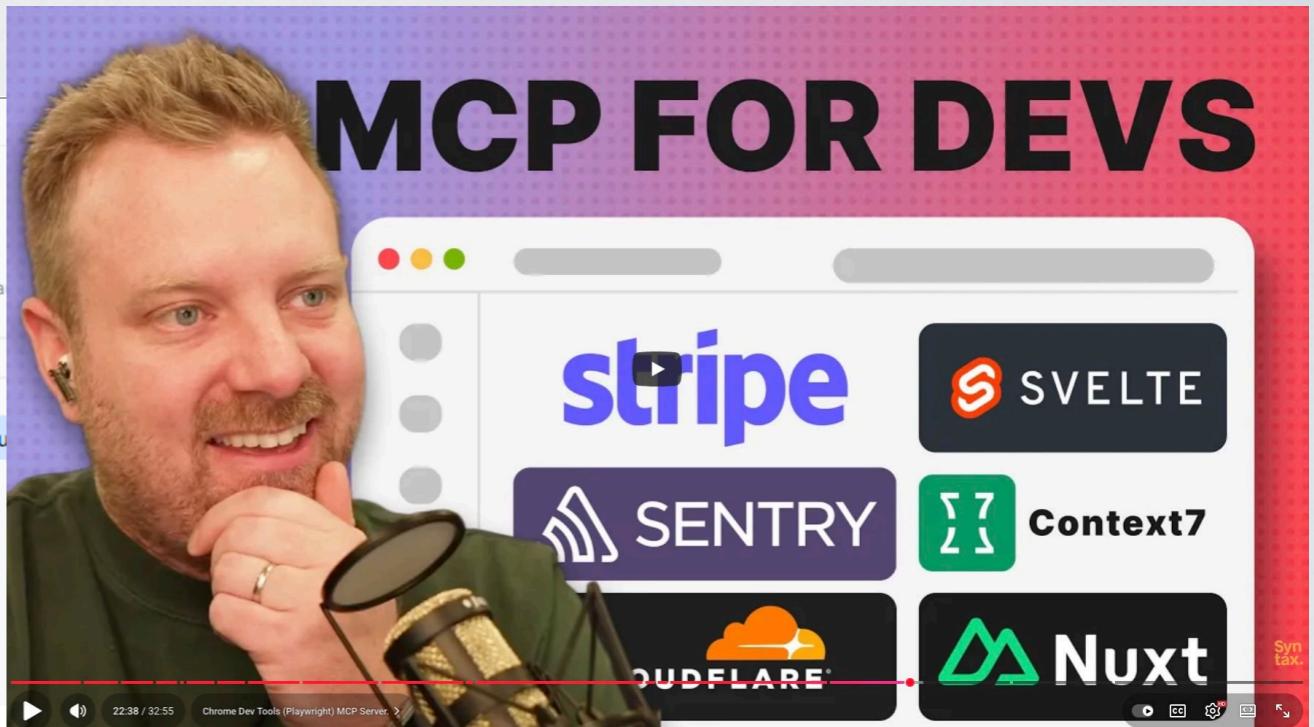
Installed MCP Servers

Context7

2 tools enabled

New MCP Server

Add a Custom MCP Server



Simon Martinelli [in](#) • 1st
Java Champion | Vaadin Champion | Oracle ACE Pro | Speaker | ...
[Book an appointment](#)
1w •

Amazing! Now I have all the MCP servers I need:

- Spring
- Vaadin
- jOOQ
- Karibu Testing

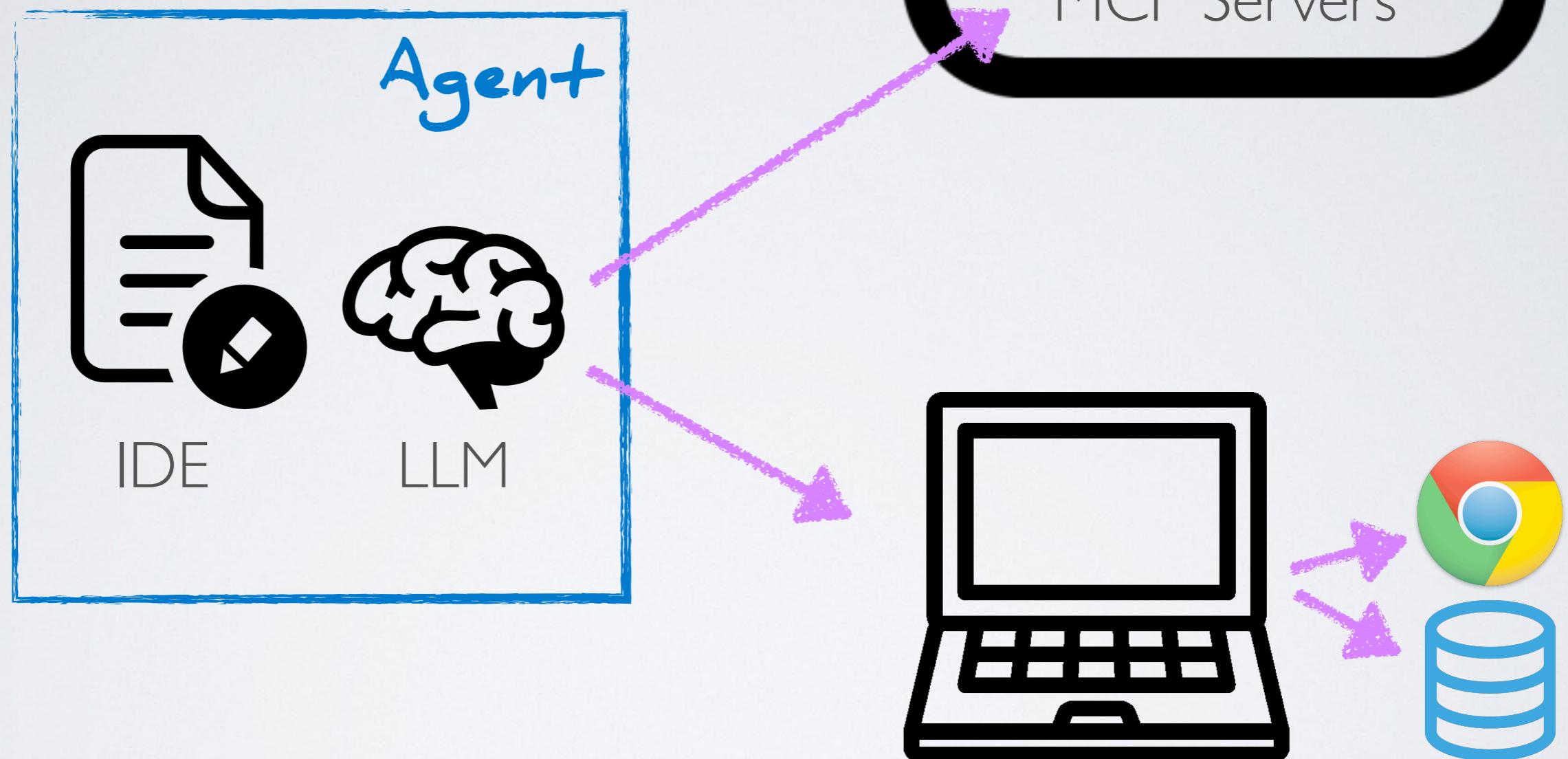
Andreas Lange [in](#) • 2nd
Principal Platform Architect @ VMware Tanzu by Bro...
1w • Edited •

[Introducing a Spring Documentation MCP Server \(not an official Spring Project\) ...more](#)

+ Follow

<https://github.com/punkpeye/awesome-mcp-servers>

Tool Calling



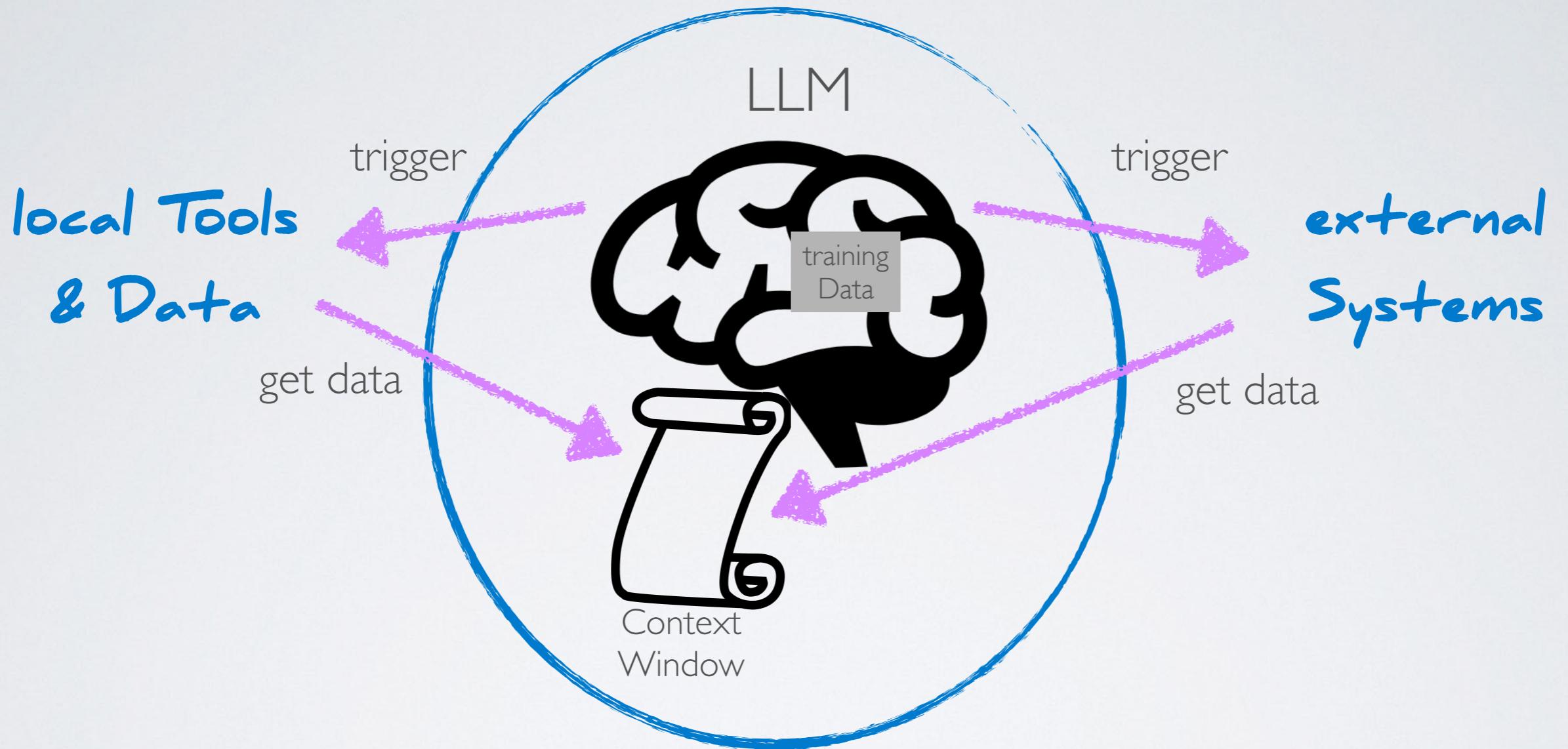
<https://jooq-mcp.fly.dev/>

<https://github.com/upstash/context7>

<https://github.com/microsoft/playwright-mcp>

<https://github.com/oracle/mcp>

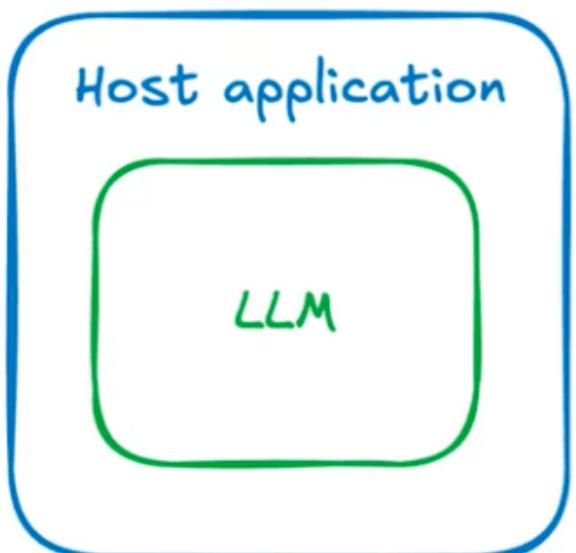
local MCP Servers



MCP standardizes how an LLM interacts with the outside world.

Phase 1

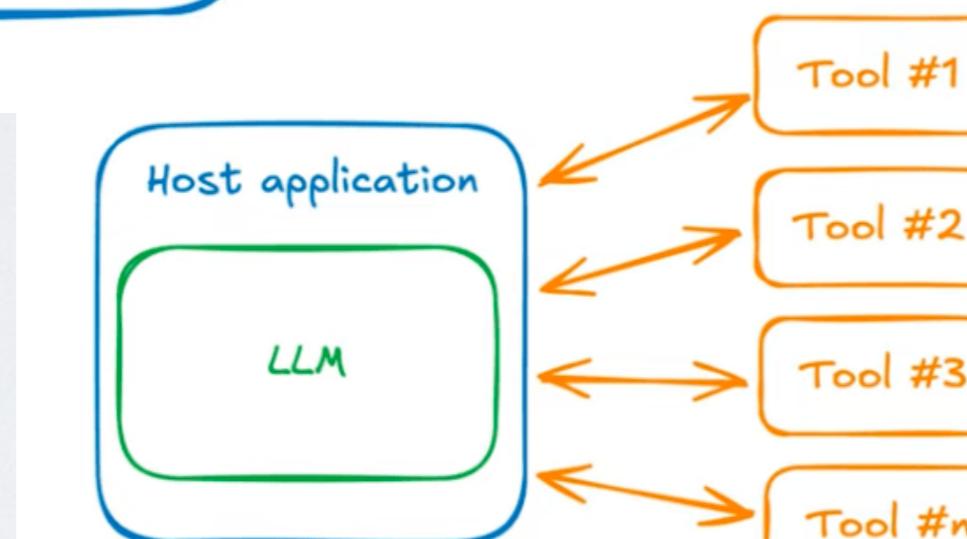
Answers questions 😐



History of Tool Calling

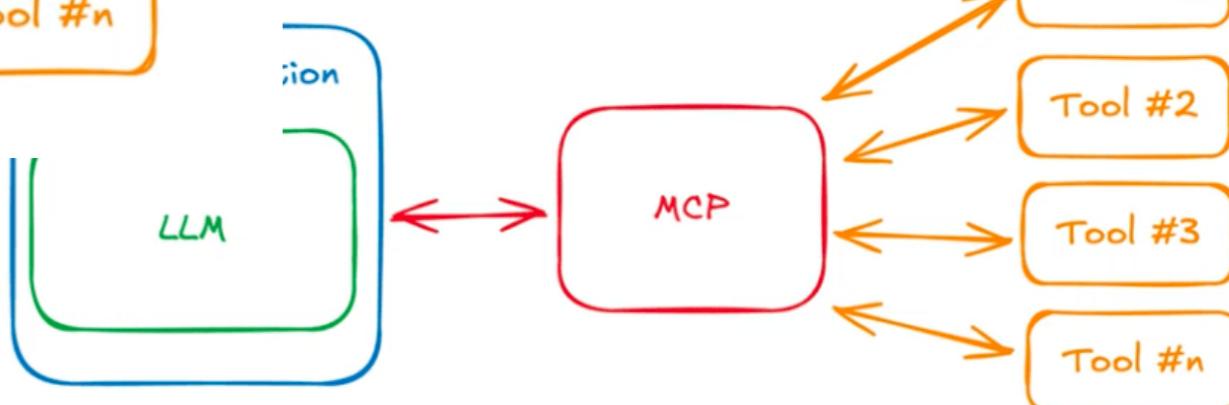
Phase 2

Can do stuff 😐



Phase 3

Can do ANYTHING 😱



MCP is bleeding edge ...



... everything seems/is experimental ...
... everything is not deterministic by nature ...

MCP might already be dead ...



<https://www.youtube.com/watch?v=JEznT-qXiA4>

The screenshot shows a blog post from Anthropic's website. The title is 'Turning off MCP Cut our Costs by 98%'. It features a large image of a man with a surprised expression. Below the title, there is a short video player showing a video about MCP. The video player has a play button, a progress bar at 0:00 / 28:35, and other video controls. The text below the video reads: 'The Model Context Protocol (MCP) is an open standard for connecting AI agents to external systems. Connecting agents to tools and data traditionally requires a custom integration for each pairing, creating fragmentation and duplicated effort that makes it difficult to scale truly connected systems. MCP provides a universal protocol—developers implement MCP once in their codebase and it works with every tool and system that supports MCP.' At the bottom, there is a caption: 'Anthropic admits that MCP sucks'.

<https://www.youtube.com/watch?v=lpiFEKA9XL0>



MCP is Dead, Long Live to ... the Code 🐧

<https://www.linkedin.com/pulse/mcp-dead-long-live-code-fred-cecilia-wgwue/>

No, MCP Isn't Dead

November 16, 2025

Every few months, the AI world goes through a familiar ritual: a new idea appears, Twitter catches fire, and someone declares that something foundational has become obsolete. Anthropic's new code-execution pattern is the latest spark, and the proclamation arrived right on schedule:

"MCP is dead."

It's not. In fact, the opposite is true. The pattern everyone is excited about only works because Model Context Protocol exists—and uses it more deeply than any agent system we've seen so far. The confusion comes from misunderstanding what MCP actually is.

<https://www.nateking.dev/blog/mcp-isnt-dead>

<https://blog.cloudflare.com/code-model/>

<https://www.anthropic.com/engineering/code-execution-with-mcp>



A look beyond AI Assisted Engineering ...

I am a fullstack developer with a focus on modern frontend development.

I work on projects, where the end-user interacts with a browser.

The browser-based user interface is an abstraction on top of the business functionality.

This talk explores a path into the future where the last two points are no longer the de-facto standard ...

Demo

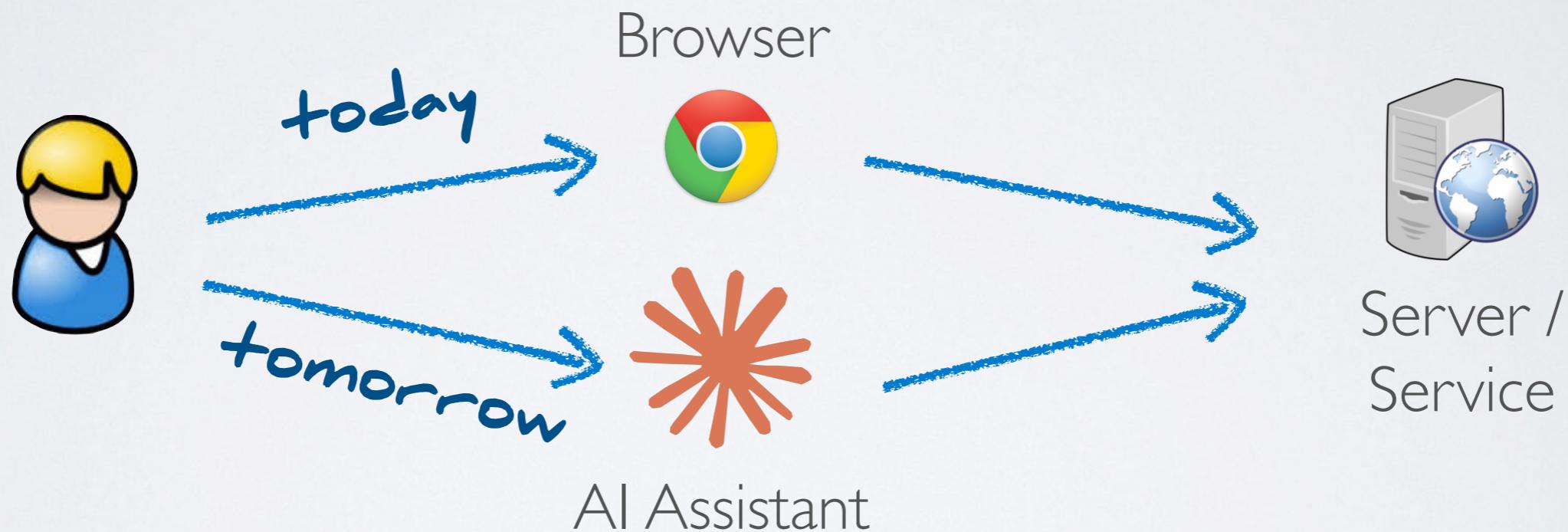
We used to add chatbots to our apps.
Now we add our apps to the chatbots.

- Kent C. Dodds

Why MCP?

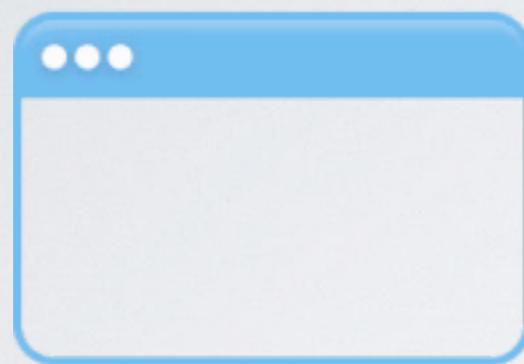
Thesis:

AI-Assistants are the the future of user interactions.

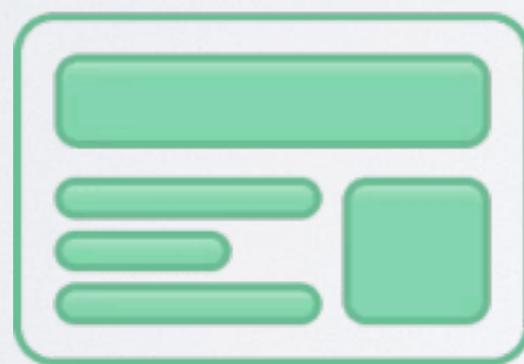


MCP was introduced as an open standard in November 2024:
https://en.wikipedia.org/wiki/Model_Context_Protocol

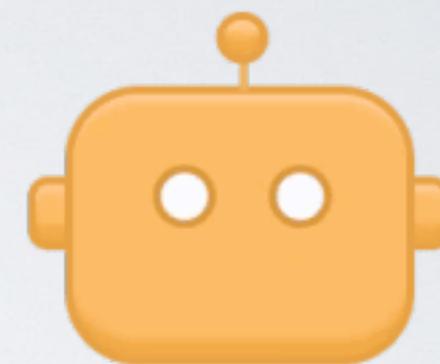
Same relationship, different domain



Browser



Website



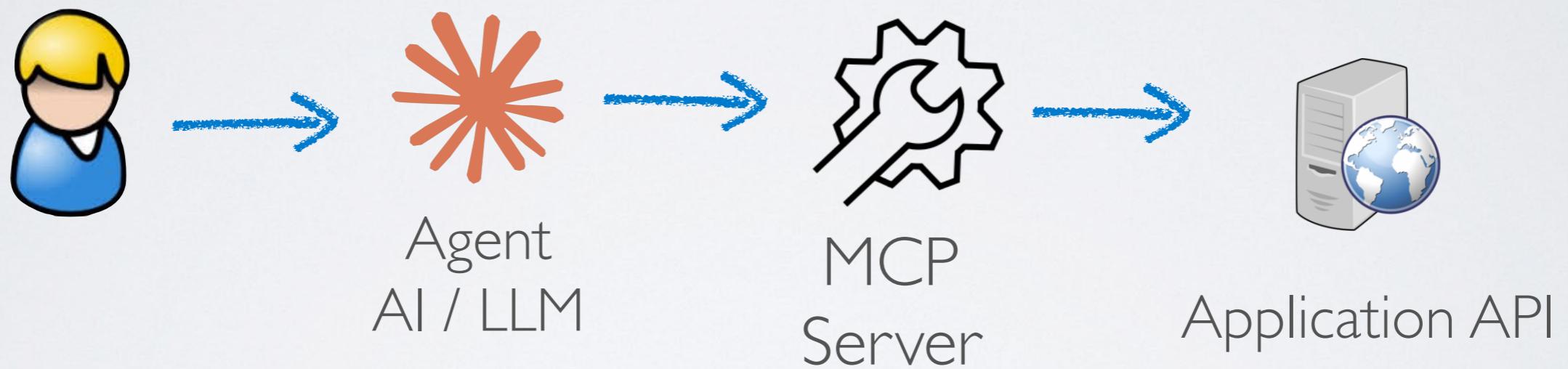
Agent



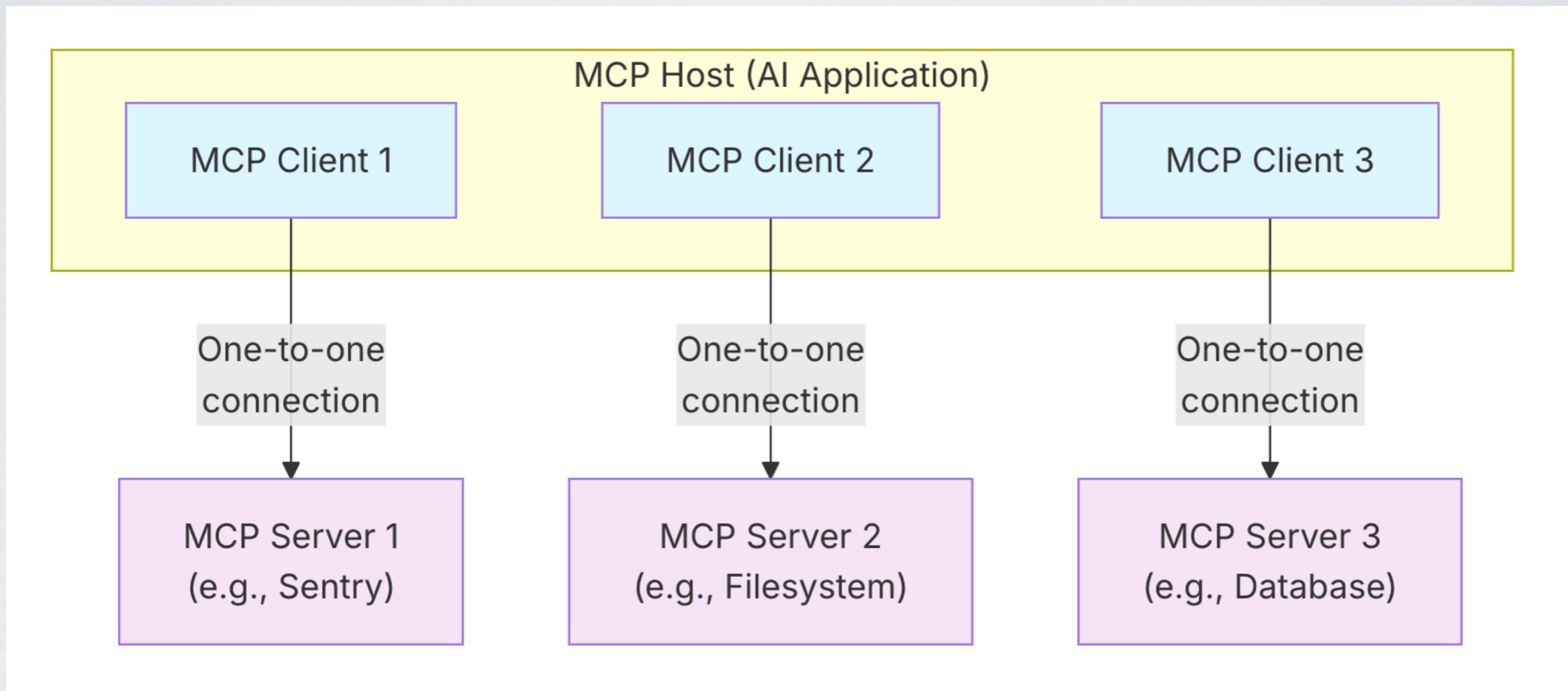
MCP Server

<https://www.epicai.pro/the-website-for-agents-mcp-servers-ee6re>

Accessing an Application



MCP Architecture



A minimal MCP Server

Server (@modelcontextprotocol/sdk):

```
const server = new McpServer({
  name: "demo-mcp-server",
  version: "1.0.0",
});
const transport = new StdioServerTransport();
await server.connect(transport);
```

```
server.registerTool(
  "get_time",
  {
    description: "Get the current time",
  },
  async () => {
    const date = new Date().toISOString();
    return {
      content: [
        type: "text" as const,
        text: `Current Date: ${date}`
      ],
      structuredContent: {
        date
      }
    };
}
```

MCP Inspector

http://localhost:6274/?MCP_PROXY_AUTH_TOKEN=2c2fa1bb215211610ee298444010f5bb3a0482dfa4fb83fcf7a1f3832e6bb0ce#tools

MCP Inspector v0.16.6

Transport Type: STDIO

Command: npx

Arguments: tsx server.ts

> Environment Variables

< Server Entry < Servers File

> Authentication

> Configuration

Restart Disconnect

Connected

Resources Prompts Tools Ping Sampling Elicitations Roots Auth

Tools

List Tools

Clear

get_weather
Get the weather in a location (fahrenheit)

convert_fahrenheit_to_celsius
Convert a temperature in fahrenheit to celsius

get_weather
Get the weather in a location (fahrenheit)
location *

Zürich

Run Tool

Tool Result: Success

Structured Content:

```
{  
  location: "Zürich"  
  temperature: 58  
}
```

Unstructured Content:

```
The temperature in Zürich is 58 degrees fahrenheit
```

History

1. initialize
2. tools/list
3. tools/call

Server Notifications

1. notifications/message
2. notifications/message

System

MCP Features

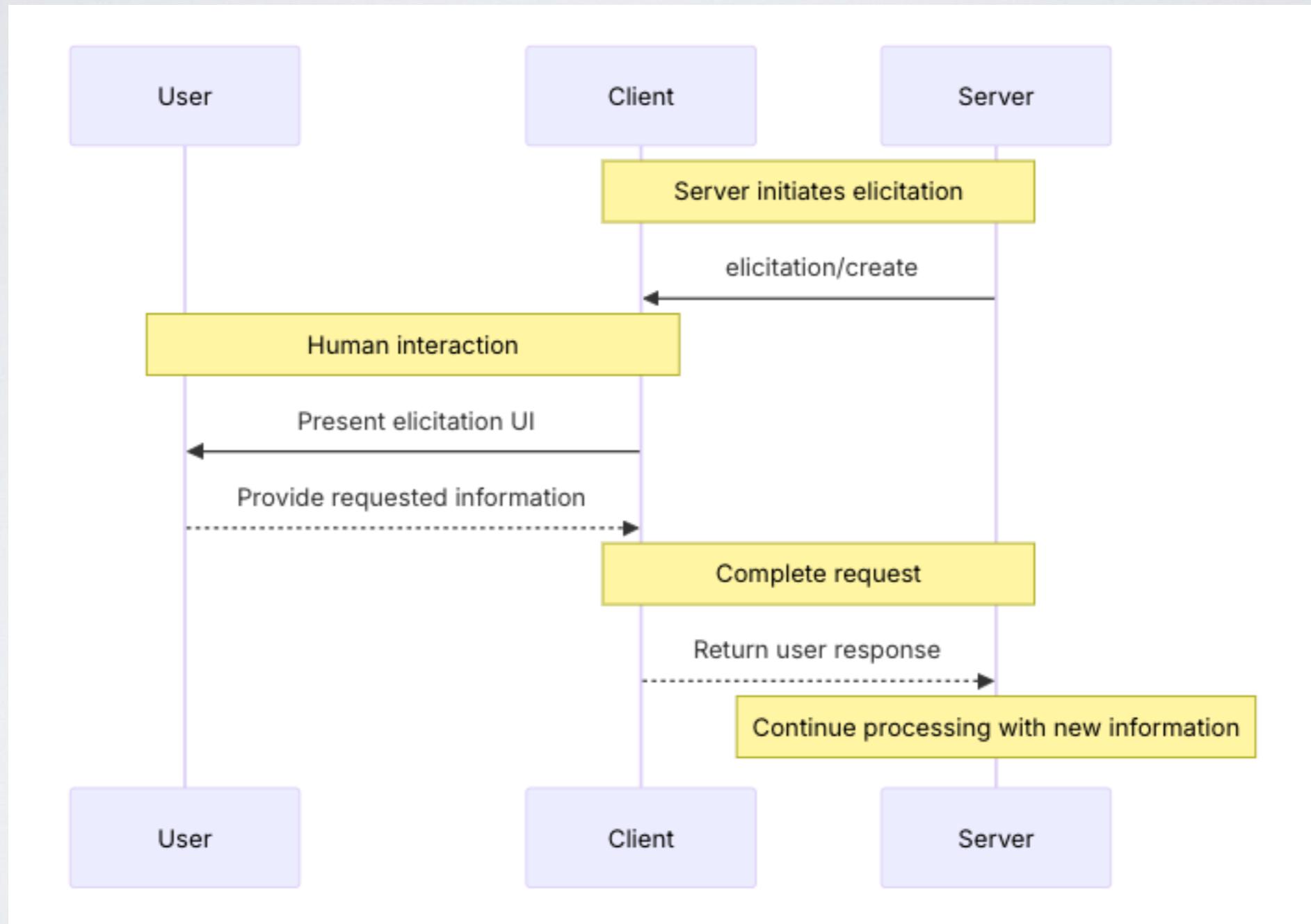
- Resources
- Prompts
- Elicitation
- Sampling
- ...

Feature support matrix

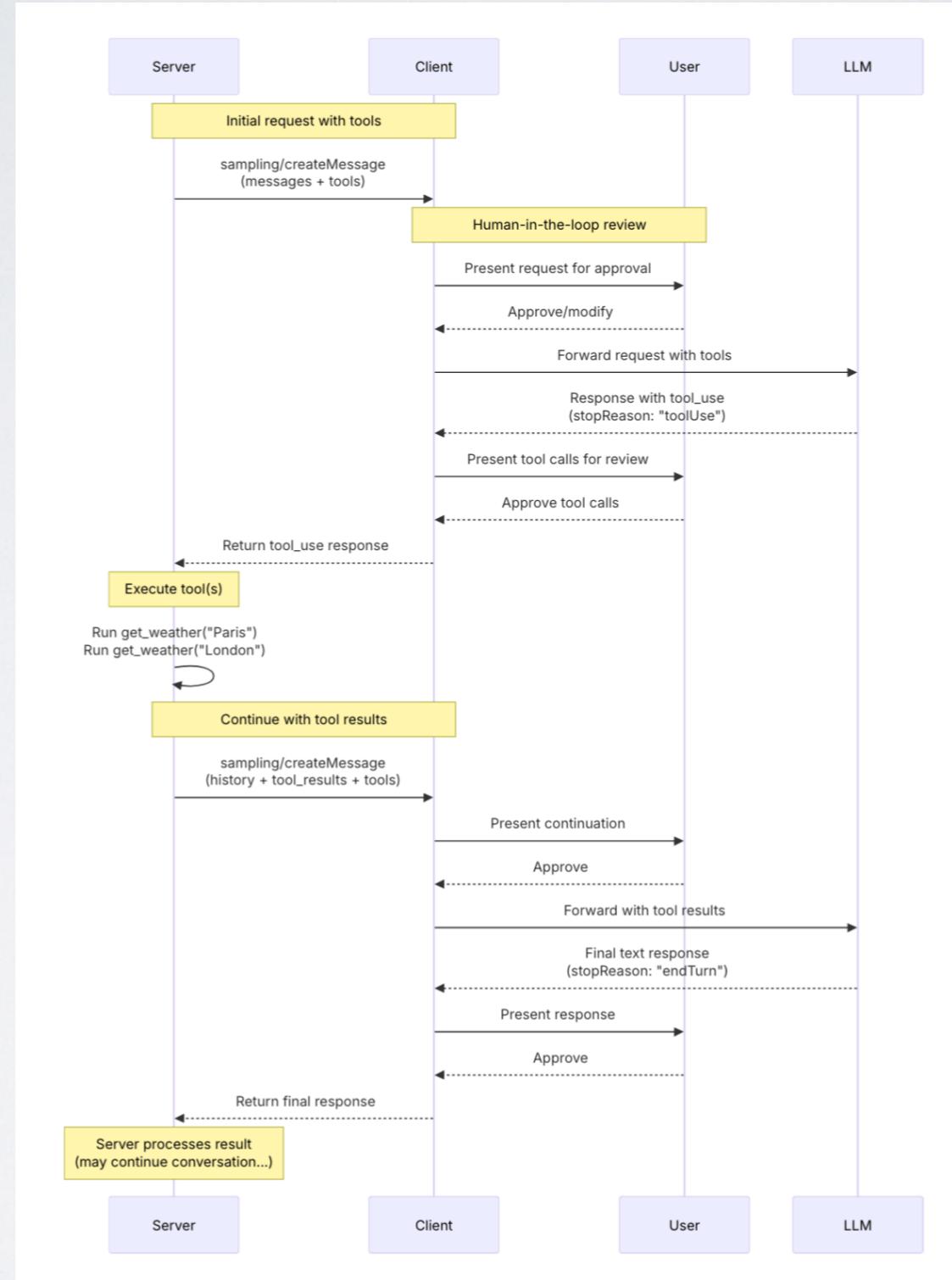
Client	Resources	Prompts	Tools	Discovery	Sampling	Roots	Elicitation
<u>5ire</u>	✗	✗	✓	?	✗	✗	?
<u>AgentAI</u>	✗	✗	✓	?	✗	✗	?
<u>AgenticFlow</u>	✓	✓	✓	✓	✗	✗	?
<u>AIQL TUUI</u>	✓	✓	✓	✓	✓	✗	?
<u>Amazon Q CLI</u>	✗	✓	✓	?	✗	✗	?
<u>Amazon Q IDE</u>	✗	✗	✓	✗	✗	✗	?
<u>Amp</u>	✓	✗	✓	✗	✓	✗	?
<u>Apify MCP Tester</u>	✗	✗	✓	✓	✗	✗	?
<u>Augment Code</u>	✗	✗	✓	✗	✗	✗	?
<u>BeeAI Framework</u>	✗	✗	✓	✗	✗	✗	?
<u>BoltAI</u>	✗	✗	✓	?	✗	✗	?
<u>Call Chirp</u>	✗	✓	✓	✗	✗	✗	?

<https://modelcontextprotocol.io/clients>

Elicitation



Sampling



MCP-UI

<https://mcpui.dev/>



Demo: <https://shopping.nanobot.ai/>
<https://mcp-aharvard.netlify.app/>

MCP-UI is now standardized into MCP Apps:
<https://github.com/modelcontextprotocol/modelcontextprotocol/pull/1865>



Search for a shirt

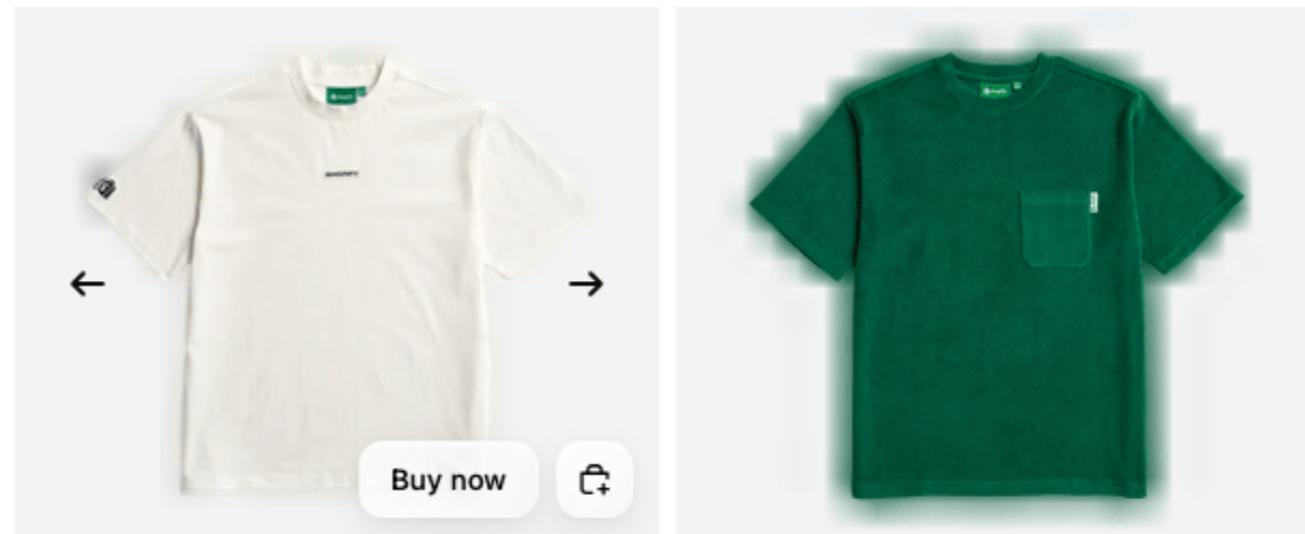
Conversations

Searching for a nice shirt

9d

I'll search for shirts in the store for you. Let me do that right away.

⚙ Tool call: search_shop_catalog



SHOPIFY SUPPLY

Front Runner Tee

\$34.00



SHOPIFY SUPPLY

Power On Tee

\$34.00



<https://shopping.nanobot.ai/>

Rendering a UI-Snippet for embedding:

```
server.registerTool(
  "search_transgourmet",
  {
    title: "Search Transgourmet Catalog",
    description: "Searches the Transgourmet catalog f....",
    inputSchema: searchTransgourmetInputSchema,
  },
  async (args) => {
    const searchTerm = args?.searchTerm?.trim?(). ?? "";

    const products = await searchTransgourmetCatalog(searchTerm);
    const htmlContent = generateProductsHTML(products, searchTerm);

    const uiResource = await createUIResource({
      uri: `ui://transgourmet-search/${searchTerm}`,
      content: {
        type: "rawHtml",
        htmlString: htmlContent,
      },
      encoding: "text",
      uiMetadata: {
        "preferred-frame-size": ["850px", "900px"],
      },
    });
  });

  return {
    content: [
      uiResource,
      {type: "text" as const, text: 'Transgourmet Search Results'}],
  };
}
```

Events calling Back into the Agent

```
<button onclick="window.parent.postMessage({ type: 'tool', payload:  
          { toolName: 'myTool', params: {} } }, '*')">  
  Call Tool  
</button>
```

Supported Actions:

- **✓ Tool calls** - { type: 'tool', payload: { toolName, params } }
- **✓ Prompts** - { type: 'prompt', payload: { prompt } }
- **✓ Intents** - { type: 'intent', payload: { intent, params } } (converted to prompts)
- **✓ Notifications** - { type: 'notify', payload: { message } }
- **✓ Render data** - Access to toolInput, toolOutput, widgetState, theme, locale
- **⚠ Links** - { type: 'link', payload: { url } } (may not be supported, returns error in some environments)

Open AI Apps SDK

announced in October 2025

<https://developers.openai.com/apps-sdk/>

Based on MCP.
UI integration is proprietary.

```
server.registerTool(  
  "get_transgourmet_cart",  
  {  
    title: "Get Transgourmet Cart",  
    description: "Retrieves the current shopping cart ...",  
    inputSchema: getCartInputSchema,  
    _meta: {  
      "openai/outputTemplate": "ui://widget/transgourmet-cart.html",  
      "openai/toolInvocation/invoking": "Fetching cart",  
      "openai/toolInvocation/invoked": "Cart retrieved",  
    },  
  },  
  async (args) => { ... }
```

```
let cart = window.openai?.toolOutput?.cart;  
  
window.addEventListener("openai:set_globals", handleSetGlobals, {  
  passive: true,  
});
```

<https://openai.com/index/introducing-apps-in-chatgpt/>

Is it ready for production?

Like with most things AI, there is a big hype around MCP.

There are some shiny demos.

For me it feels like the early browser days:
features are very basic
duck-tape implementations

But it might worth to keep an eye on:
frameworks and patterns might evolve very quickly.

Personally I think it will not replace browsers anytime soon.
(I am not sure I would even like a future where it replaces browsers)

Have fun with MCP!

Slides & Code: <https://github.com/ivorycode/mcp-sbb-2025>



Jonas Bandi
JavaScript / Angular / React / Vue / Vaadin
Schulung / Beratung / Coaching / Reviews
jonas.bandi@ivorycode.com