Hands-On MCP

(Model Context Protocol)

ABOUT ME

Jonas Bandi
jonas.bandi@ivorycode.com
Twitter: @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

digicomp

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

jonas.bandi@ivorycode.com

About you ...

Imposter Syndrome

... I am not an expert on the topic ...

... the topic is "bleeding edge" ...

... everything seems/is experimental ...

... not deterministic by nature ...

... lets have fun and hopefully learn something on the way.

Material

Git Repository:
`https://github.com/ivorycode/mcp-chopen-2025/`

Initial Clone:
```
git clone https://github.com/ivorycode/mcp-chopen-2025/
```

Update:
```
git pull
```

```
git reset --hard
git clean -dfx
git pull
```

(discard all local changes)

Slides & Exercises: `<checkout>/00-CourseMaterial`

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

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.

Context:

(my interpretation of the AI-Hype)

AI Engineering vs AI Assisted Engineering

AI Engineering vs Machine Learning

Thesis:

AI Engineering will be a needed skill for future Application Development.

AI Assisted Engineering will become a "sandard" feature of dev tools.

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

Agenda

- Agents and Tool Calling
- Local MCP
- Remote MCP
- An MCP to access an application via API
- Advances MCP Features (Demo)

LLMs / Chatbots / Agents / Assistants

Agents:

- Acts autonomously
- Perceives its environment
- Pursues goals
- Makes decisions
- Interacts with other agents or systems

AI Identity Exploration

J What are you?

I'm Claude, an AI assistant created by Anthropic. I'm a large language model designed to be helpful, harmless, and honest in conversations. I can assist with a wide variety of tasks like answering questions, helping with analysis and research, creative writing, coding, math problems, and general conversation.

I process and generate text by recognizing patterns in language, but questions about my inner workings - like whether I have consciousness or subjective experiences - remain philosophically complex and uncertain. What I can say is that I aim to be helpful while being direct about the limits of what I know, including about my own nature.

Is there something specific I can help you with today?

Retry

J Are you an agent?

I can function in ways that might be described as "agentic" - I can use tools, break down complex tasks into steps, search for information, and work toward goals you give me. For instance, I can search the web, analyze data, create documents, and even access some external services through various tools I have available.

However, I don't operate independently or pursue my own autonomous goals. I respond to what you ask me to do within our conversation. I don't take actions outside of our chat or maintain persistent goals across different conversations.

The term "agent" can mean different things in different contexts - from simple automated programs to more sophisticated AI systems that can plan and act independently. I'd say I have some agent-like capabilities when it comes to using tools and working through multi-step problems, but I'm not an autonomous agent operating independently in the world.

What kind of tasks were you thinking about? That might help clarify whether my capabilities align with what you're looking for.

Retry

Claude can make mistakes. Please double-check responses.

Reply to Claude...

Research

Claude Sonnet 4

Standalone "Programs"

- Claude Desktop
  https://claude.ai/download
- Visual Studio Code Copilot
  https://code.visualstudio.com/insiders/
- Goose
  https://github.com/block/goose

"Programmatic Clients"

Vercel AI SDK: The AI SDK is the TypeScript toolkit designed to help developers build AI-powered applications.

https://ai-sdk.dev/docs/introduction

Providers: https://ai-sdk.dev/docs/foundations/providers-and-models

https://ai-sdk.dev/docs/getting-started/nextjs-app-router
https://ai-sdk.dev/docs/getting-started/nodejs
https://vercel.com/templates?type=ai

EXERCISE

Exercise - A Simple AI Chatbot

Zod & JSON Schema

```
import z from 'zod';

const personSchema = z.object({
  name: z.string(),
  films: z.array(z.string())
})
type Person = z.infer<typeof personSchema>;

async function fetchLuke() {
  const lukeResponse = await fetch(`https://swapi.info/api/people/1`);
  const luke = await lukeResponse.json();
  return personSchema.parse(luke);
}

function printMessage(person: Person) {
  console.log(`Luke appears in ${person.films.length} movies`)
}

async function main() {
  const luke = await fetchLuke();
  printMessage(luke)
}
```

runtime check

compile time check

Zod integrates with JSON Schema, to describe an API for the AI consumers.

https://zod.dev/

https://zod-playground.vercel.app/

Zod Demo

"runtime type-checking"

```
import z from 'zod';

const personSchema = z.object({
  name: z.string(),
  films: z.array(z.string())
})
type Person = z.infer<typeof personSchema>;

async function fetchLuke() {
  const lukeResponse = await fetch(`https://swapi.info/api/people/1`);
  const luke = await lukeResponse.json();
  return personSchema.parse(luke);
}

function printMessage(person: Person) {
  console.log(`Luke appears in ${person.films.length} movies.` );
}

async function main() {
  const luke = await fetchLuke();
  printMessage(luke)
}
```

central definition!

runtine check

compile time check

Tool Calling

Programmatic Clients

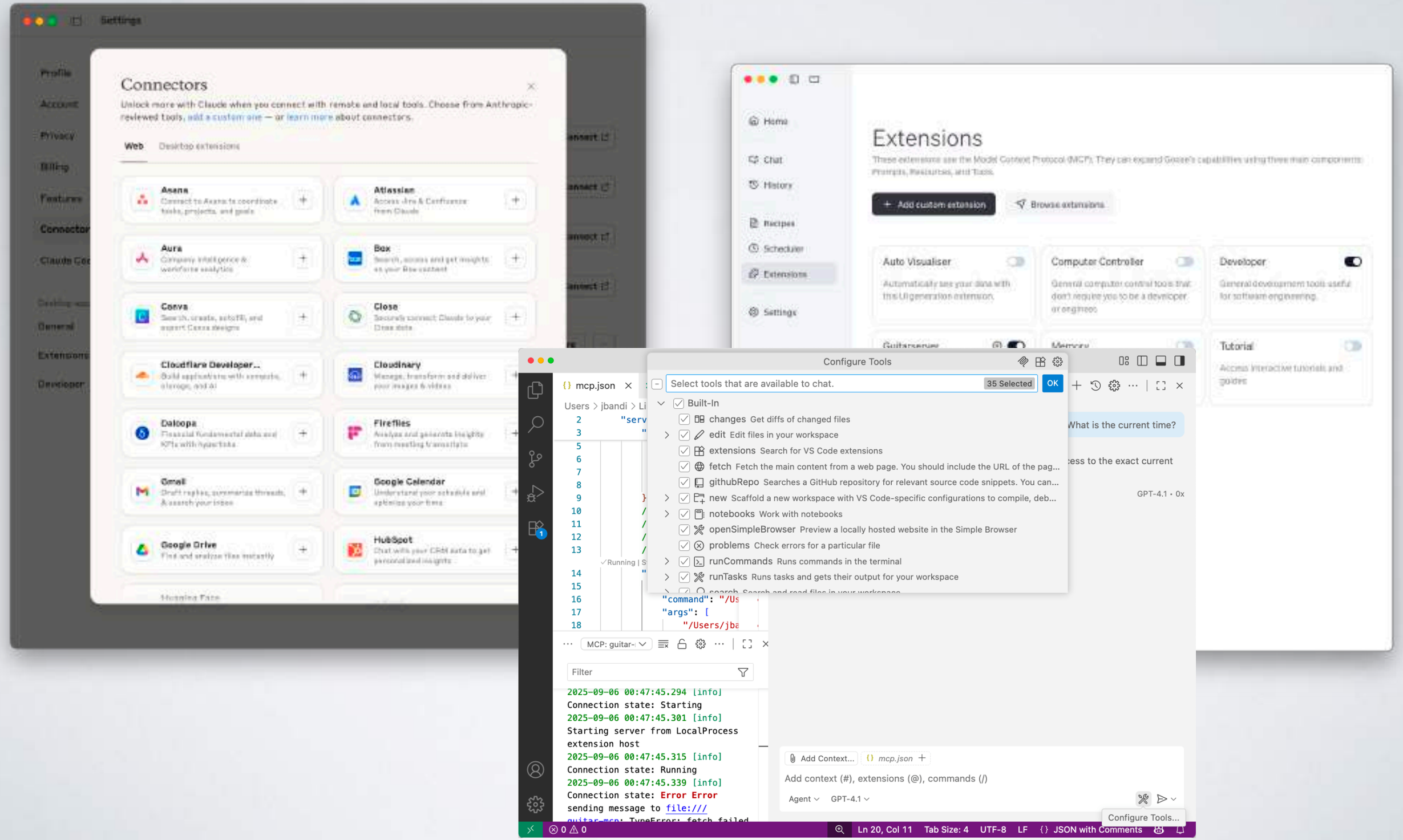
Heavy Coupling

https://ai-sdk.dev/docs/ai-sdk-core/tools-and-tool-calling

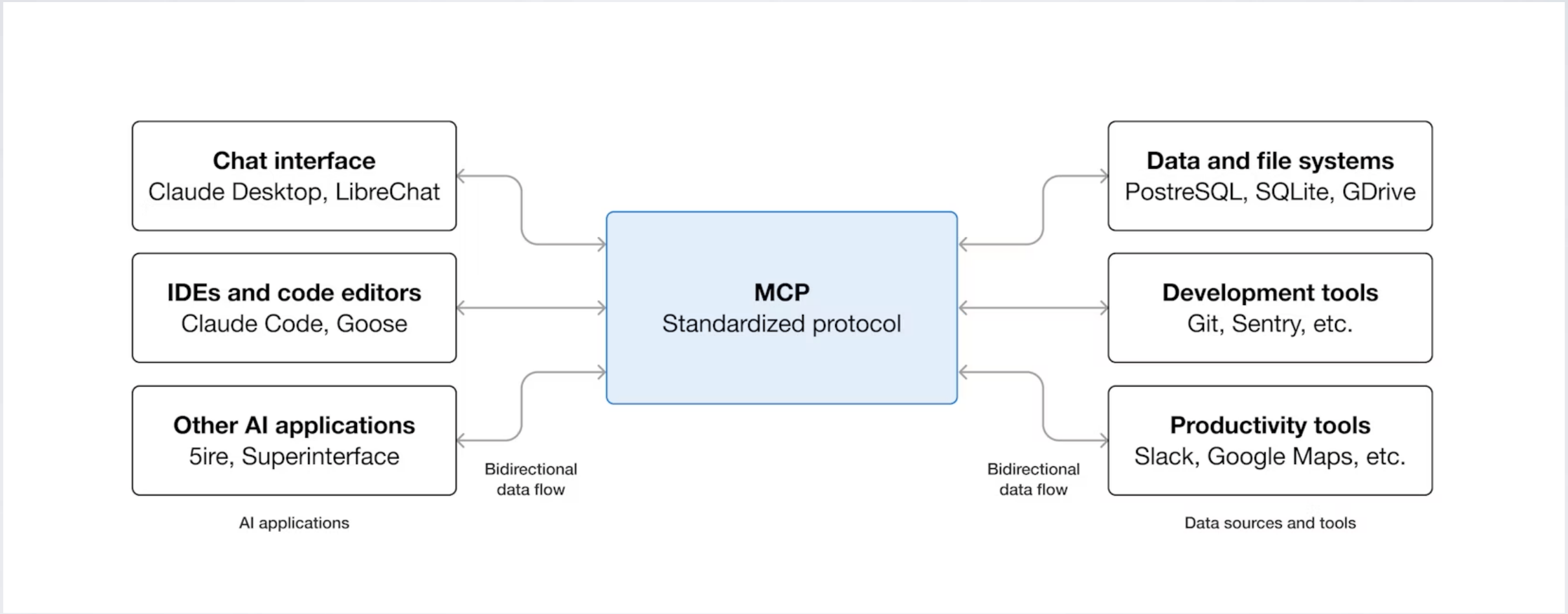
EXERCISE

Exercise - AI Chatbot with Tools

Assistants and Tool Calling

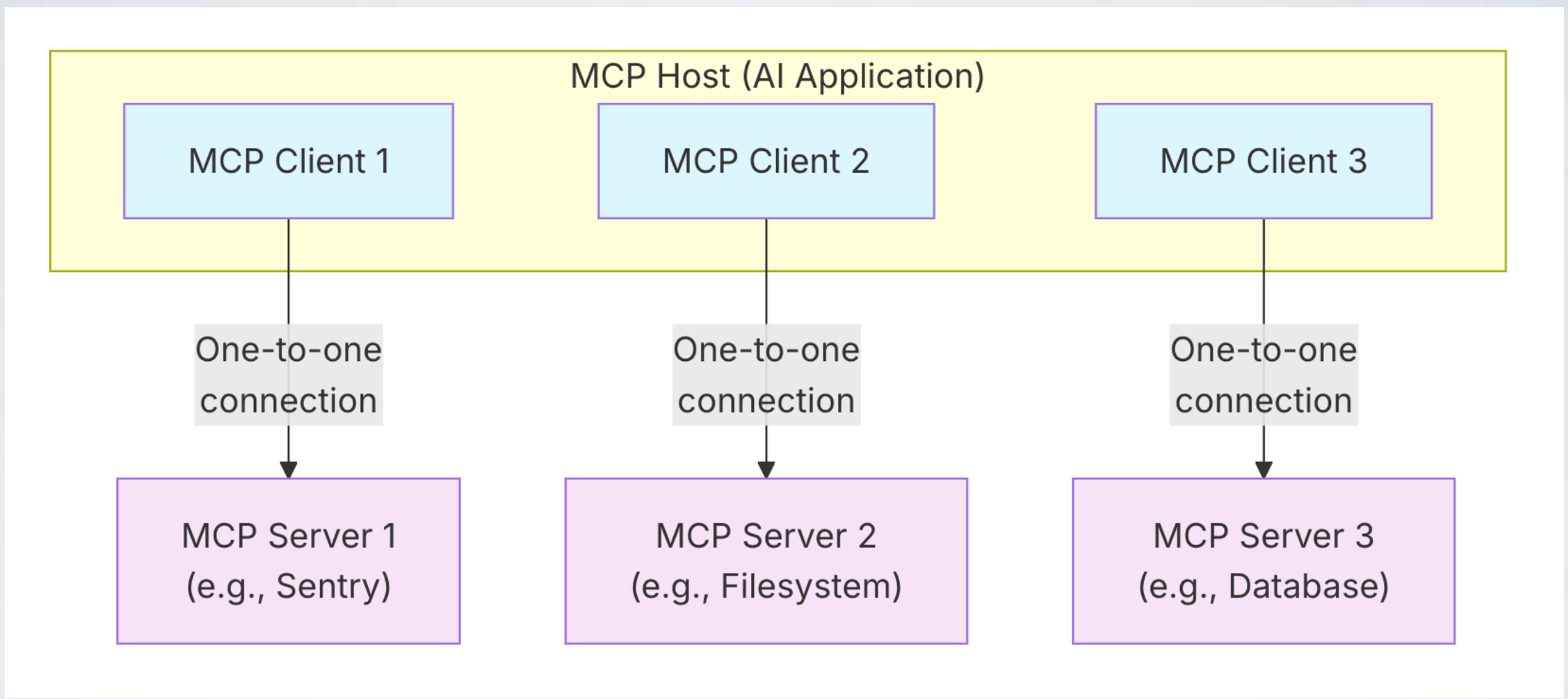


The Role of MCP



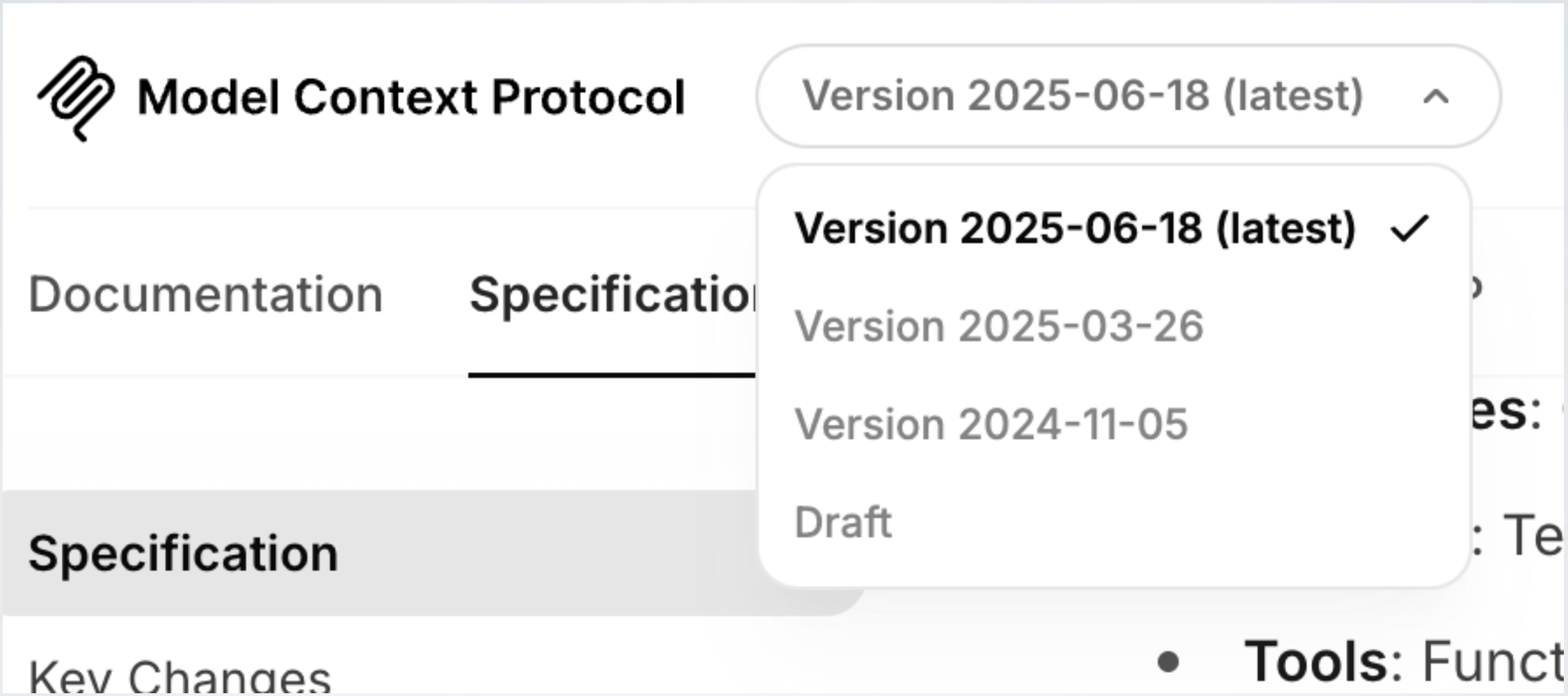
https://modelcontextprotocol.io/docs/getting-started/intro

MCP Architecture



https://modelcontextprotocol.io/docs/learn/architecture

History of MCP



https://modelcontextprotocol.io/specification/2025-06-18

A minimal MCP Server

Server (@modelcontextprotocol/sdk):

```
const server = new McpServer({
  name: "weather",
  version: "1.0.0",
  capabilities: {
    resources: {},
    tools: {},
  },
});
const transport = new StdioServerTransport();
await server.connect(transport);
```

Client (Vercel AI SDK):

```
const mcpClient = await createMCPClient({
  transport: new StdioMCPTransport({
    command: 'node',
    args: ['src/stdio/dist/server.js'],
  }),
});

const tools = await mcpClient.tools();
```

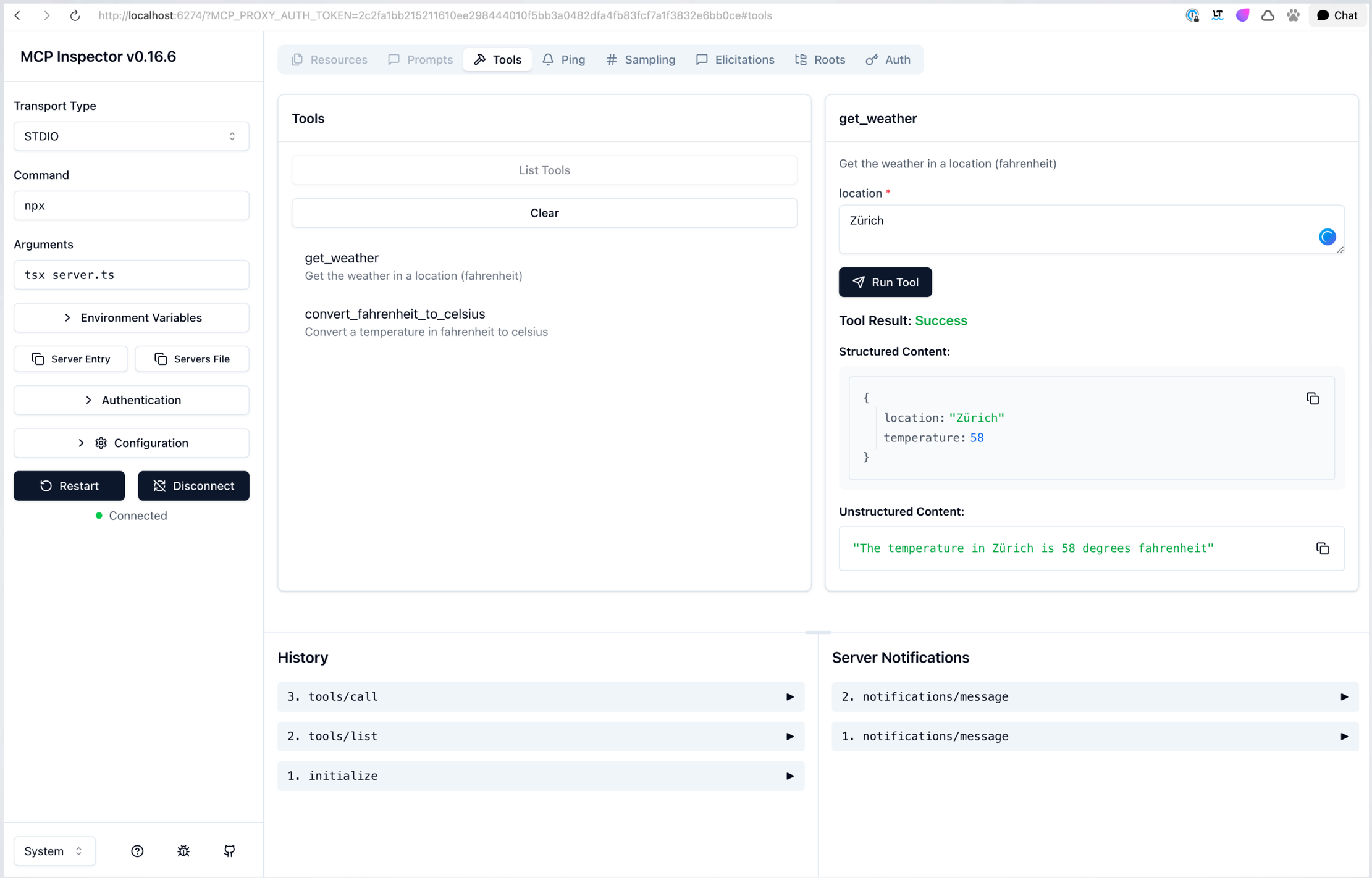
MCP Inspector:

```
npx @modelcontextprotocol/inspector npx tsx server.ts
```

https://modelcontextprotocol.io/quickstart/server#node
https://modelcontextprotocol.io/legacy/tools/inspector
https://modelcontextprotocol.io/specification/2025-06-18/server/tools#structured-content

https://ai-sdk.dev/docs/ai-sdk-core/tools-and-tool-calling#mcp-tools
Schema Discovery / - Definition: https://ai-sdk.dev/docs/ai-sdk-core/tools-and-tool-calling#schema-definition

MCP Inspector



Footgun: logging to stdout

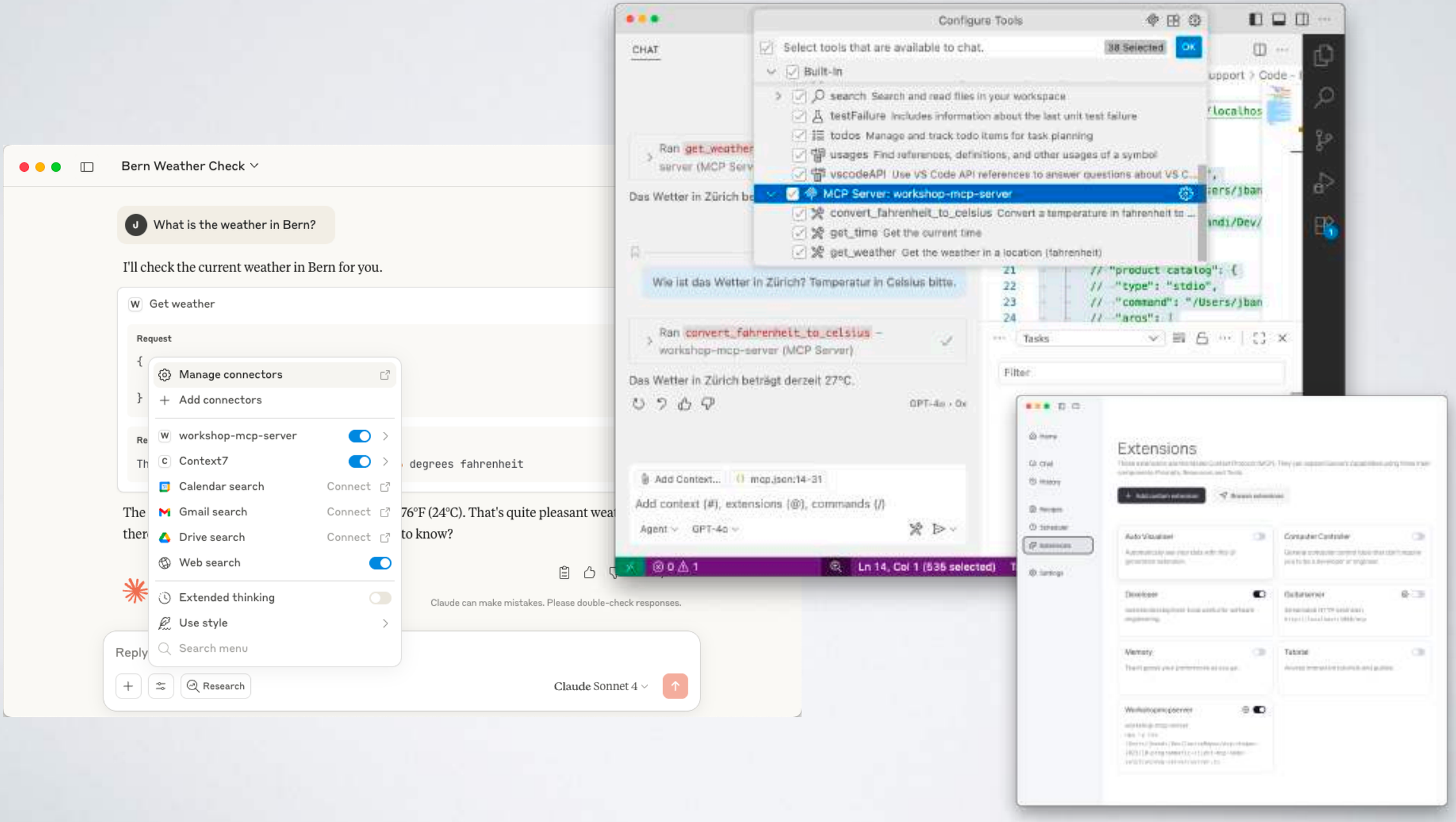
```
// ❌ Bad (STDIO)
console.log("Server started");

// ✅ Good (STDIO)
console.error("Server started"); // stderr is safe
```

Or use a file-based logging solution.

EXERCISE

Exercise - Your First MCP Server

Consuming a MCP Server in Standalone Clients



A remote MCP Server

Basic Express Server:
https://github.com/modelcontextprotocol/typescript-sdk#streamable-http
https://github.com/vercel-labs/express-mcp

MCP Next.js Integration:
https://github.com/vercel/mcp-handler

Cloudflare Template:
https://developers.cloudflare.com/agents/guides/remote-mcp-server/

Consuming a remote MCP Server in Standalone Clients

Claude Desktop:

- via mcp-remote: https://www.npmjs.com/package/mcp-remote
- Connectors? -> require https

Visual Studio Code - Add Server with "type": "http"

Goose - Add Server with "type": "http"

EXERCISE

Exercise - Remote MCP Server

Add Authentication

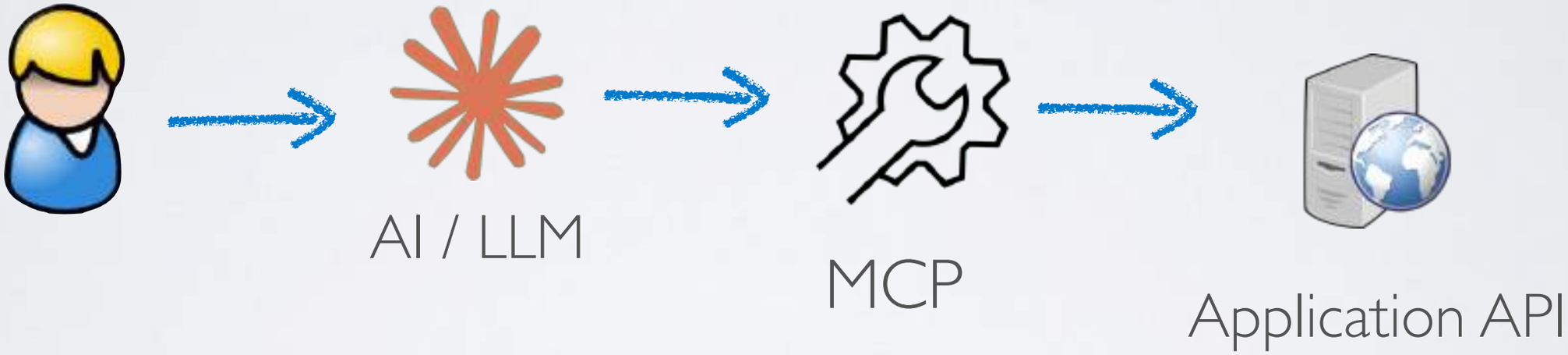
Next.js with Clerk:
https://clerk.com/docs/nextjs/mcp/build-mcp-server

Cloudflare:
https://developers.cloudflare.com/agents/guides/remote-mcp-server/

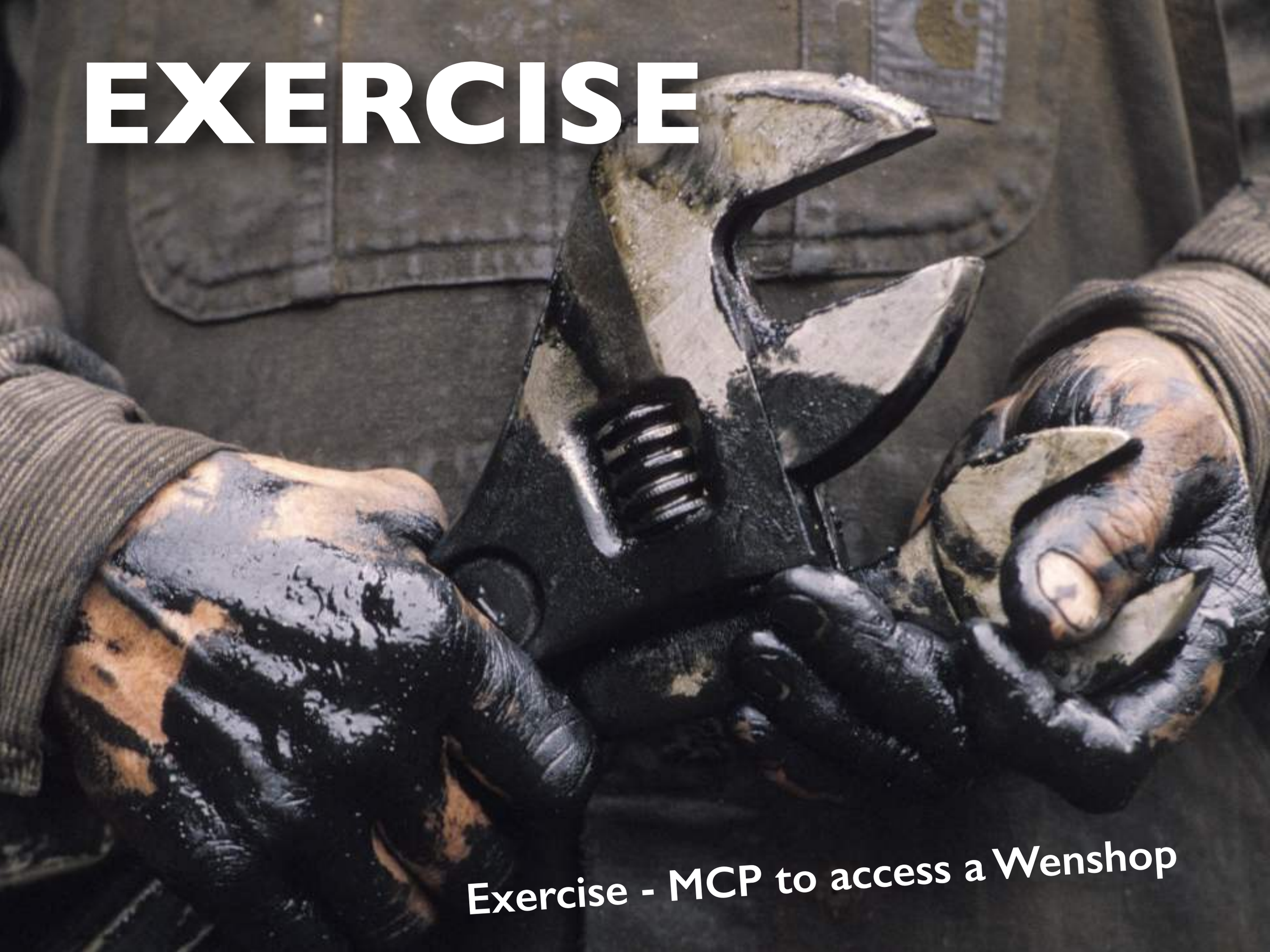
EXERCISE

Exercise - Remote MCP Server with Auth

Accessing an Application



EXERCISE



Exercise - MCP to access a Wenshop

Advanced MCP Features

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

Feature support matrix

Client	Resources	Prompts	Tools	Discovery	Sampling	Roots	Elicitation
5ire	❌	❌	✅	?	❌	❌	?
AgentAI	❌	❌	✅	?	❌	❌	?
AgenticFlow	✅	✅	✅	✅	❌	❌	?
AIQL TUUI	✅	✅	✅	✅	✅	❌	?
Amazon Q CLI	❌	✅	✅	?	❌	❌	?
Amazon Q IDE	❌	❌	✅	❌	❌	❌	?
Amp	✅	❌	✅	❌	✅	❌	?
Apify MCP Tester	❌	❌	✅	✅	❌	❌	?
Augment Code	❌	❌	✅	❌	❌	❌	?
BeeAI Framework	❌	❌	✅	❌	❌	❌	?
BoltAI	❌	❌	✅	?	❌	❌	?
Call Chirp	❌	✅	✅	❌	❌	❌	?

https://modelcontextprotocol.io/clients

MCP Frameworks

- Vercel AI SDK
  https://ai-sdk.dev/docs/ai-sdk-core/tools-and-tool-calling#mcp-tools

- Cloudflare
  https://developers.cloudflare.com/agents/guides/remote-mcp-server/

- xmcp
  https://xmcp.dev/

Have fun with MCP!

Slides & Code: https://github.com/ivorycode/mcp-chopen-2025/

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