Model Context Protocol

Local

Real world interaction

A full walkthrough

What you'll learn

Local setup

Ollama & Open WebUI

MCP

Getting to know the protocol.

Own MCP server :)

Creating tools to interact with the world how we want!

Setup!

Wonder how to set the foundation?

We need:

- Tool to run models
- Nice interface

Ollama

CLI Tool

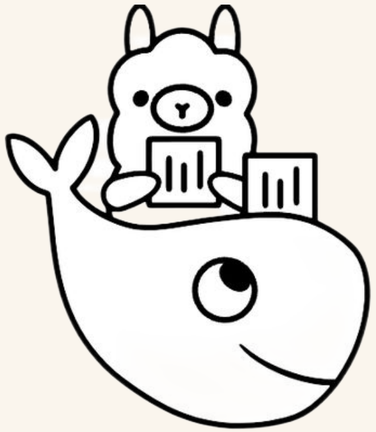
Server for local LLMs

What is Ollama?

- wraps the llama.cpp engine
- Go-based server and CLI
- exposing a simple HTTP API

Why this tool?

- usage
- setup
- community & frequent updates



Easiest way to run it

https://ollama.com/

```
docker run -d --gpus=all -v ollama:/root/.ollama -p
11434:11434 --name ollama ollama/ollama
```

Open WebUI

Web Tool

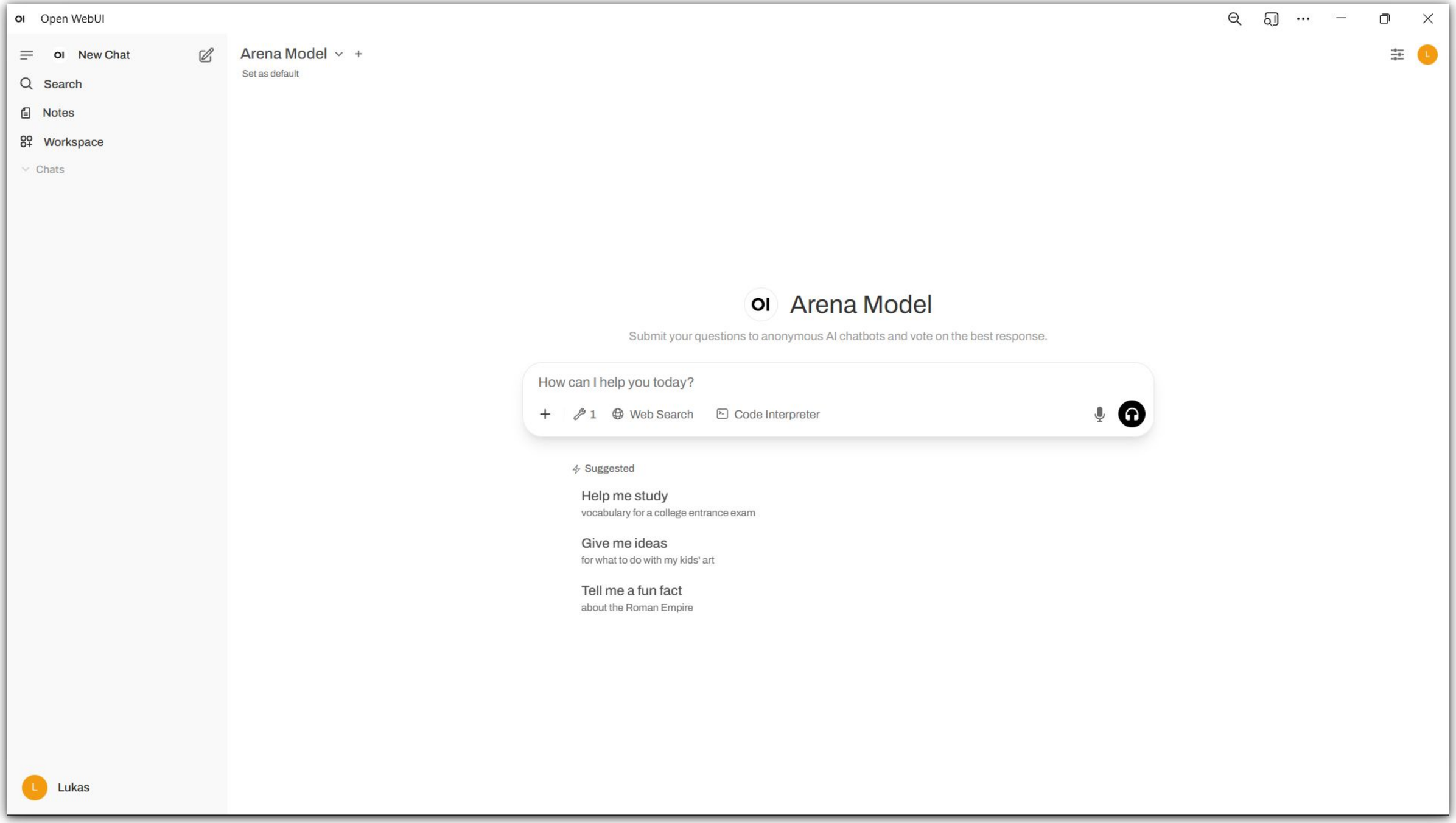
Interface for LLMs

What is Open WebUI?

- Modern Chat tool for interacting with LLMs
- Interests local & cloud LLMs

Why this tool?

- Ease of use
- many great features
- detects Ollama directly
- Supports MCP tols :)


Open WebUI
New Chat
Search
Notes
Workspace
Chats
Lukas
Arena Model
Set as default
Arena Model
Submit your questions to anonymous AI chatbots and vote on the best response.
How can I help you today?
1
Web Search
Code Interpreter
Suggested
Help me study
vocabulary for a college entrance exam
Give me ideas
for what to do with my kids' art
Tell me a fun fact
about the Roman Empire

What Model ?

It depends..
What is the task? The hardware?
speed/accuracy tradeoff?

LMArena

Ollama Search

Models Turbo

Search models

Sign in Download

Embedding Vision Tools Thinking Popular

gpt-oss

OpenAI's open-weight models designed for powerful reasoning, agentic tasks, and versatile developer use cases.

tools thinking 20b 120b

424.2K Pulls 3 Tags Updated yesterday

deepseek-r1

DeepSeek-R1 is a family of open reasoning models with performance approaching that of leading models, such as O3 and Gemini 2.5 Pro.

tools thinking 1.5b 7b 8b 14b 32b 70b 671b

56.3M Pulls 35 Tags Updated 1 month ago

gemma3

The current, most capable model that runs on a single GPU.

vision 1b 4b 12b 27b

11M Pulls 21 Tags Updated 3 months ago

qwen3

Leaderboard Overview

See how leading models stack up across text, image, vision, and beyond. This page gives you a snapshot of each Arena, you can explore deeper insights in their dedicated tabs. Learn more about it here.

Text

3 days ago

Rank (UB)	Model	Score	Votes
1	gemini-2.5-pro	1459	26,137
1	o3-2025-04-16	1452	32,151
2	chatgpt-4o-latest-20250326	1442	30,875
3	gpt-4.5-preview-2025-02-27	1438	15,271
3	qwen3-235b-a22b-instruct-2507	1430	4,264
4	grok-4-0709	1428	13,120
5	claude-opus-4-20250514-think…	1420	18,063
5	kimi-k2-0711-preview	1420	11,809

WebDev

View →

Rank (UB)	Model	Score	Votes
1	Gemini-2.5-Pro	1403	6,813
1	DeepSeek-R1-0528	1389	4,441
2	Claude Opus 4 (20250514)	1380	8,708
2	GLM-4.5	1360	722
3	GLM-4.5-Air	1347	752
4	Qwen3-Coder	1361	6,119
4	Claude Sonnet 4 (20250514)	1359	7,913
4	Claude 3.7 Sonnet (20250219)	1358	7,460

Rules of Thumb

Model-size VRAM

~7 B parameters → ~8 GB VRAM
~13 B parameters → ~16 GB VRAM

Quantization q4

Size: roughly 25 %
Speed: up to 2.4 × faster
Quality loss: typically negligible

Quantization q8

Size: roughly 50 %
Speed: ~1.56 × faster
Quality loss: minimal

Testing!

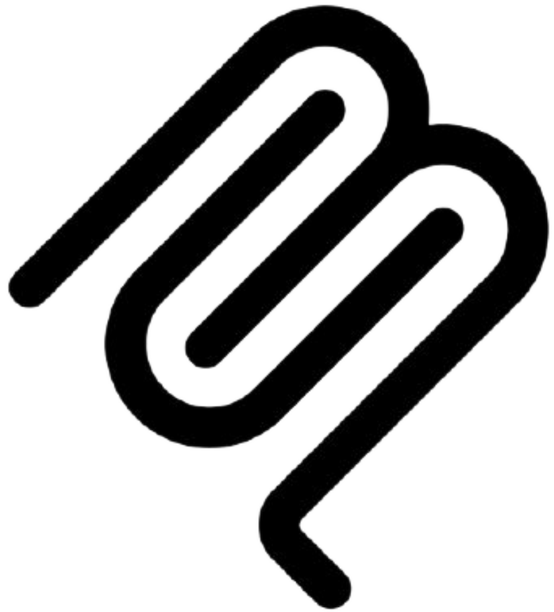
At the end, it really comes down to test out the best options.

Ollama

Open WebUI

LLM

Model Context Protocol

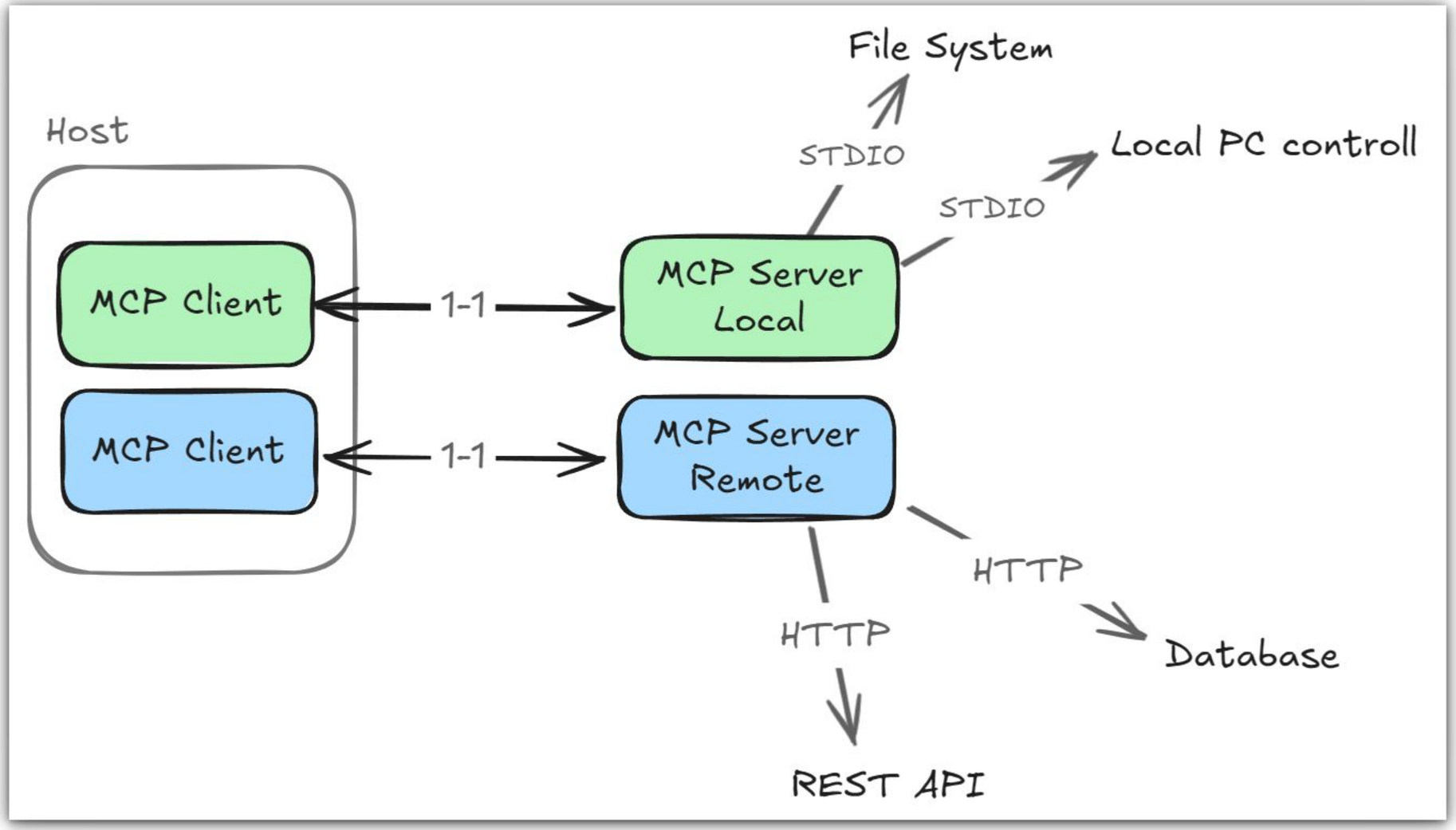


Wonder how why we need it?

- Open interoperability standard
- Client–server architecture
- Rich primitives with human-in-the-loop

Client–server architecture

Simple design where both Server and Client can implement functionalities and communicate bidirectional.


File System
Host
STDIO
Local PC controll
STDIO
MCP Client
1-1
MCP Server
Local
MCP Client
1-1
MCP Server
Remote
HTTP
HTTP
Database
REST API

Primitives

The MCP defines 3 Primitives, with all of them serving different purposes.

Primitives

Tools

- Executable functions
- Provided to the model
- model controlled

Prompts

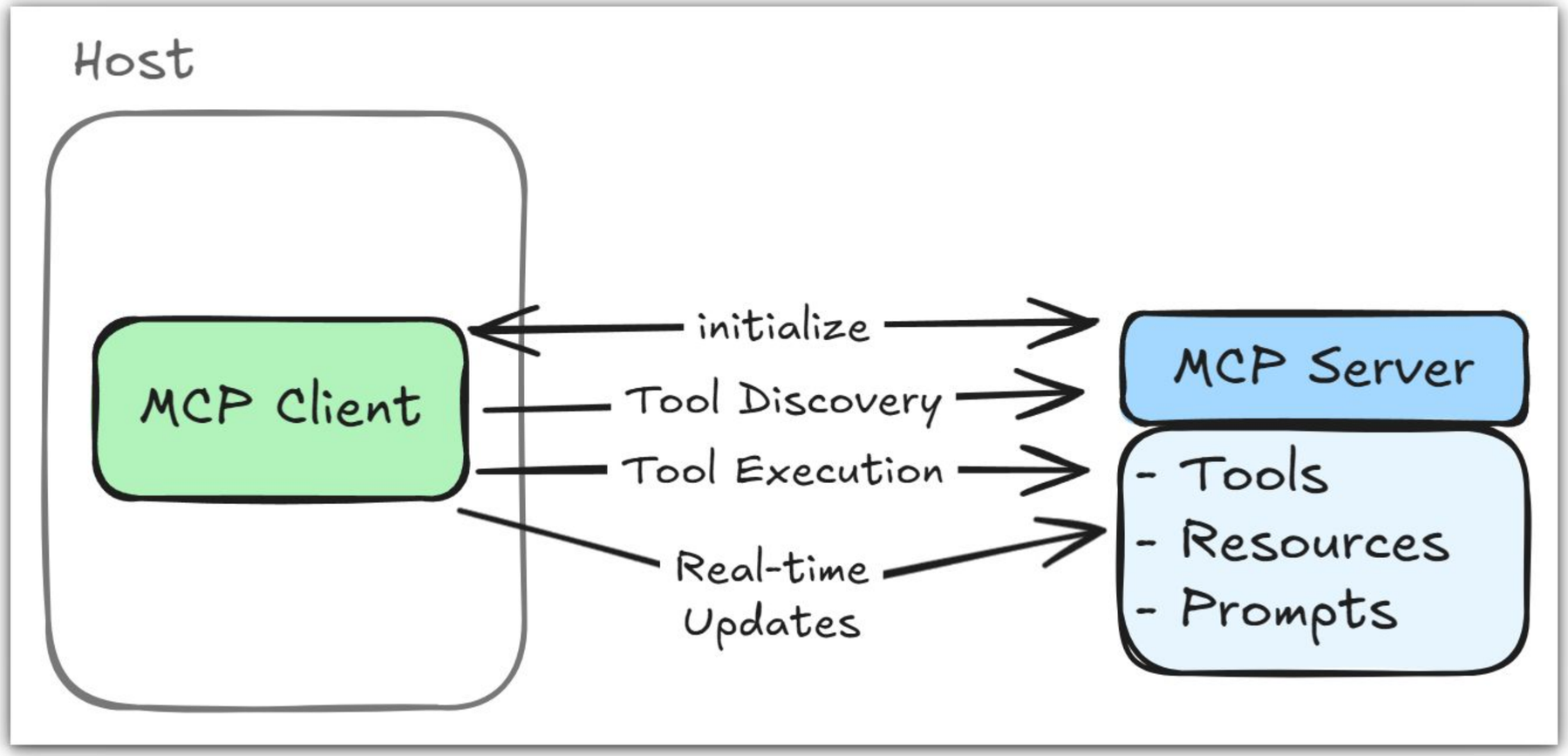
- Reusable templates
- system prompts
- few-shot examples

Resources

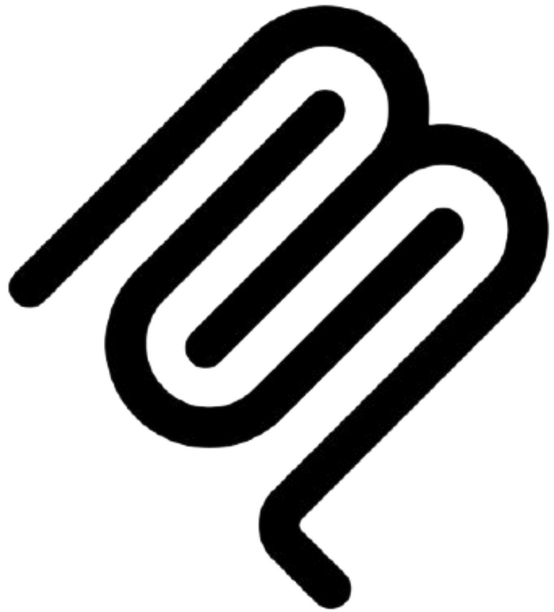
- Data sources that provide contextual information
- Client controlled

Primitives

The MCP defines 3 Primitives, with all of them serving different purposes.


Host
MCP Client
initialize
Tool Discovery
Tool Execution
Real-time
Updates
MCP Server
- Tools
- Resources
- Prompts

Own MCP



Wonder how why we need it?

- Open interoperability standard
- Client–server architecture
- Rich primitives with human-in-the-loop

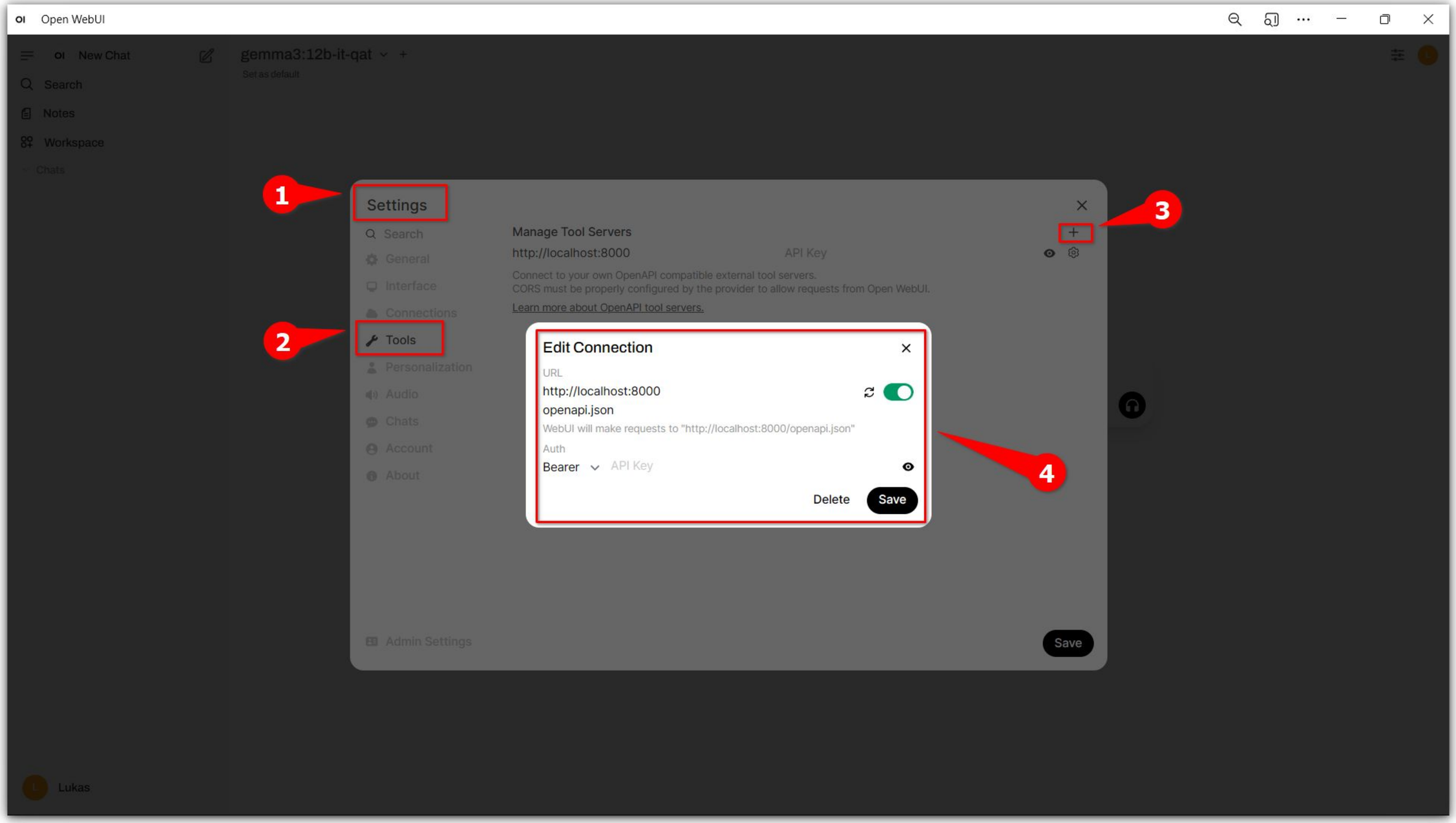
Available SDKs

- TypeScript
- Python
- Go
- Kotlin
- Swift
- Java
- C#
- Ruby
- Rust

```python
from mcp.server.fastmcp import FastMCP
from pynput.keyboard import Key, Controller
import time

# Initialize the FastMCP server
mcp = FastMCP("LazyPresentator")

# Create a keyboard controller instance
keyboard = Controller()

# Placeholder :)


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

```python
@mcp.tool()
def press_n_key(delay_seconds: int = 5) -> str:
    """
    For lazy presentators: Automatically presses the 'n' key to advance to the next slide
    after a specified delay.

    This is perfect for presentators who are too lazy to press the 'n' key themselves.
    The default delay gives you time to finish speaking before advancing.

    Args:
        delay_seconds: Number of seconds to wait before pressing the 'n' key (default: 2)

    Returns:
        A confirmation message about the slide advancement.
    """
    try:
        # Press and release the 'n' key
        time.sleep(delay_seconds)
        keyboard.press('n')
        keyboard.release('n')

        # Add a small delay to ensure the keypress is registered
        time.sleep(0.1)

        return "Successfully pressed the 'n' key. Screen should be blanked/unblanked if in
presentation mode."

    except Exception as e:
        return f"Error pressing 'n' key: {str(e)}"
```

```
mcpo --port 8000 -- python .\lazy_presentator.py
```


Open WebUI
New Chat
gemma3:12b-it-qat
Set as default
Search
Notes
Workspace
Chats
Lukas
1
2
3
4
Settings
Search
General
Interface
Connections
Tools
Personalization
Audio
Chats
Account
About
Admin Settings
Manage Tool Servers
http://localhost:8000
API Key
Connect to your own OpenAPI compatible external tool servers.
CORS must be properly configured by the provider to allow requests from Open WebUI.
Learn more about OpenAPI tool servers.
Save
Edit Connection
URL
http://localhost:8000
openapi.json
WebUI will make requests to "http://localhost:8000/openapi.json"
Auth
Bearer
API Key
Delete
Save

AI took over!

Demo

Now we are ready to get into the demo how easy it is to interact with the **Open Data Hub** via our new MCP Server skills!

OPEN
DATA
HUB

Thank you!

Happy hacking!

Article

Model Context Protocol | Lukas's ὑπόμνημα