Powering AI with MCP

https://bit.ly/2025-nov1-mcp

https://bit.ly/2025-nov1-mcp

THOU SHALT NOT MAKE A MACHINE IN THE LIKENESS OF A HUMAN MIND

FRANK HERBERT

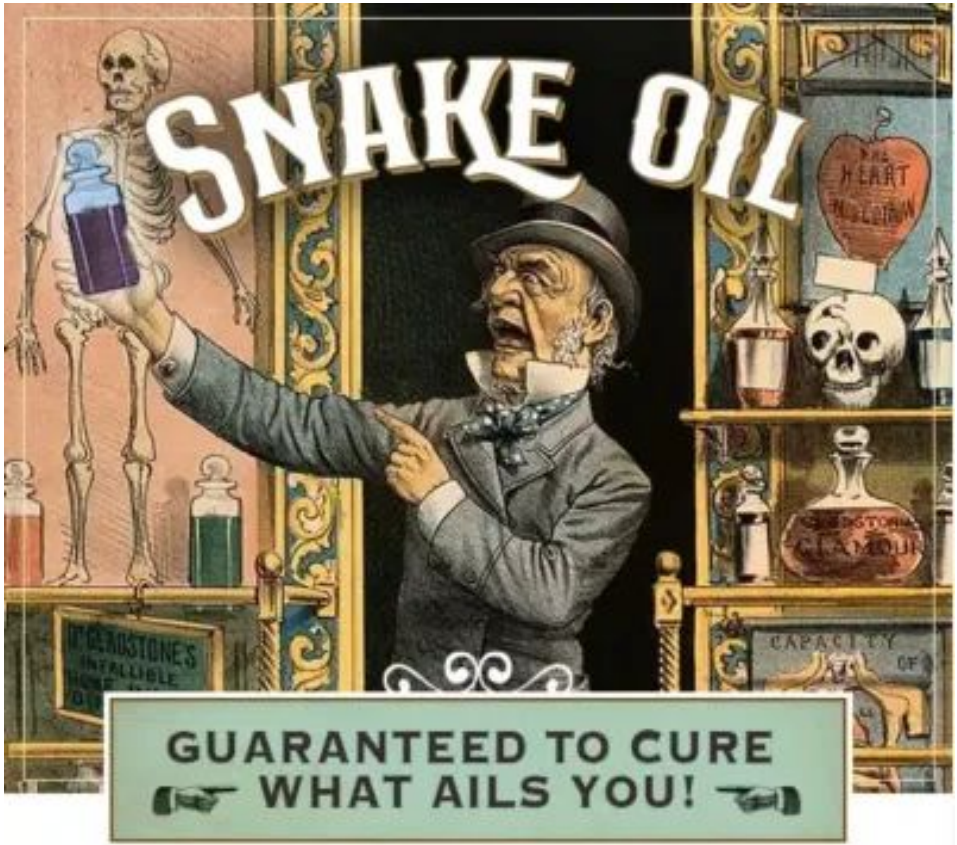
Assume

Comfortable with Python

Have a basic understanding of what is GenAI

- LLM
- Agents
- GenAI != AI

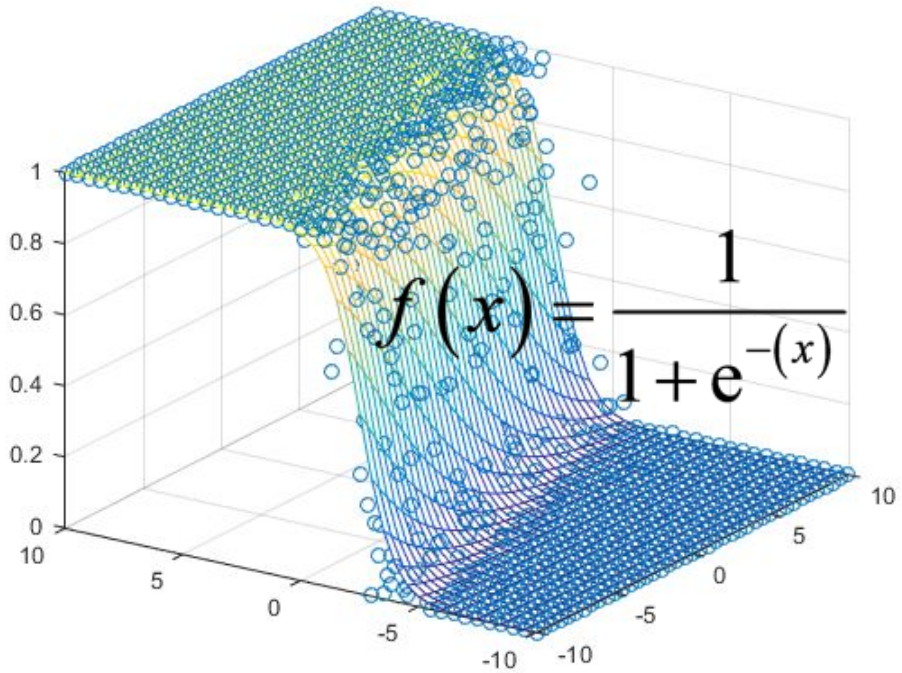
https://bit.ly/2025-nov1-mcp



Ours software is AI enabled!

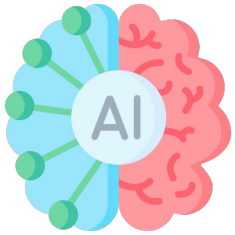
https://bit.ly/2025-nov1-mcp

Programmed vs Learnt



https://bit.ly/2025-nov1-mcp

LLM



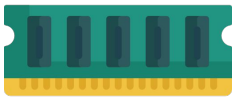
Intelligence

PROMPT

Instructions for the model

Information to augment the model

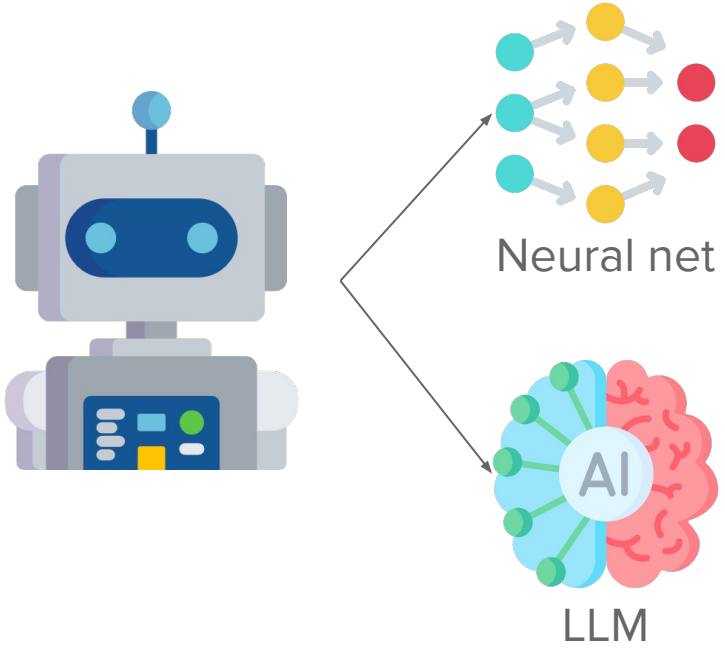
Tools to interact with the outside world



Memory give context to ongoing conversation

https://bit.ly/2025-nov1-mcp

What are Agents?



Autonomous programs

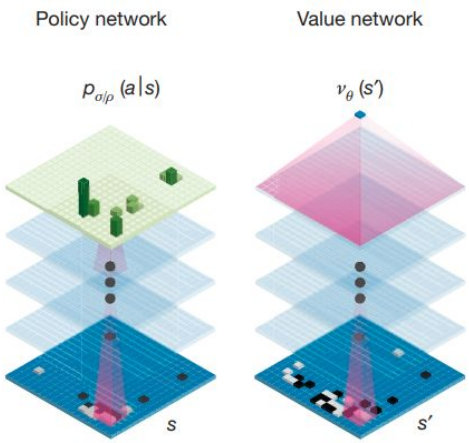
Acts on behalf of another entity to accomplish a given task

Examples of non AI agent
- Log scraper
- Chat bots

AI agents uses 'AI' to make decisions

https://bit.ly/2025-nov1-mcp

Alpha Go



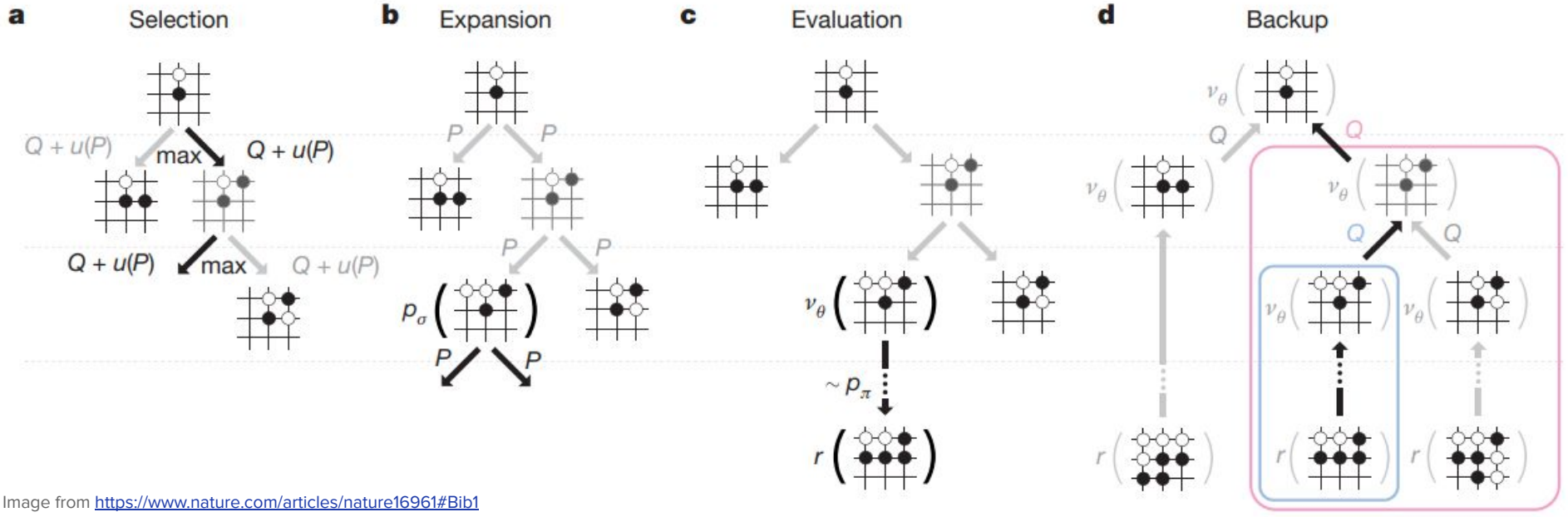


Image from https://www.nature.com/articles/nature16961#Bib1

https://bit.ly/2025-nov1-mcp

Today's AI Agent

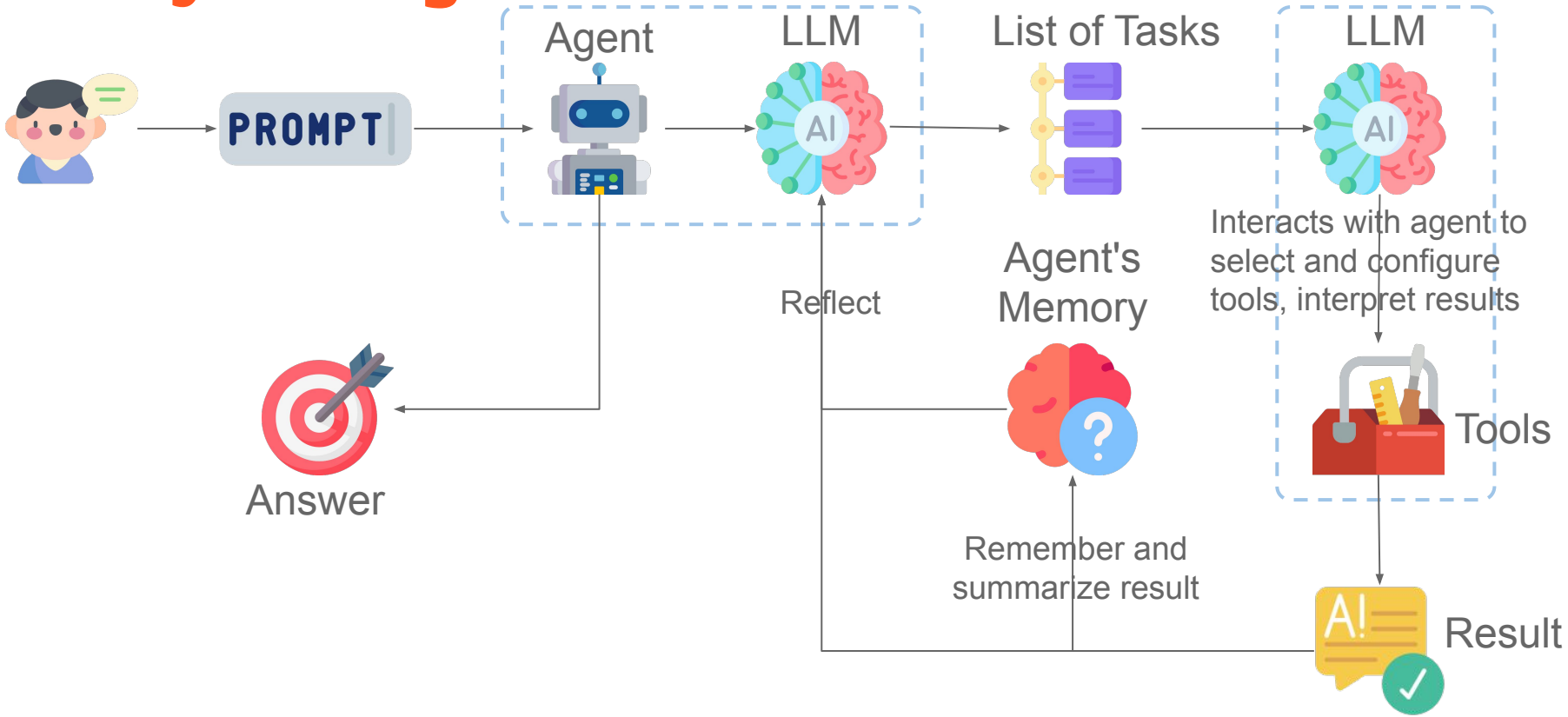
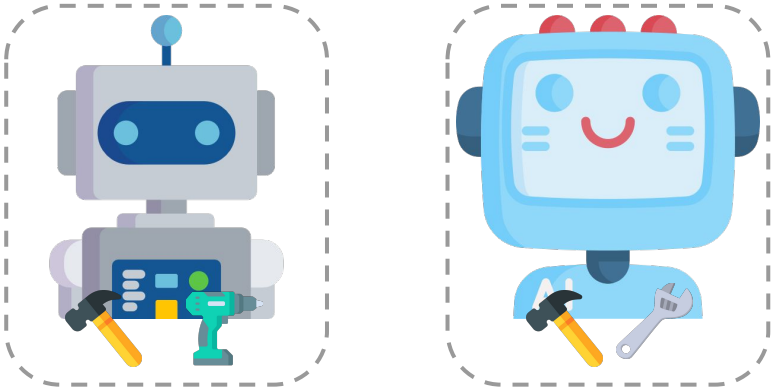


Diagram adapted from https://www.linkedin.com/pulse/building-your-own-autonomous-llm-agents-praphul-singh-uompc/

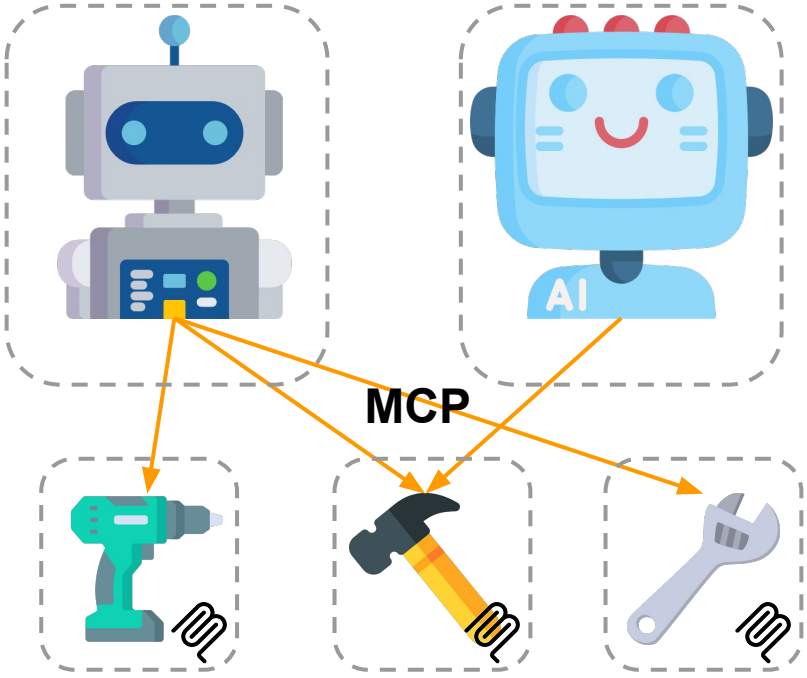
https://bit.ly/2025-nov1-mcp

Tools

Each agent has its own copy of the tool



Tools are externalized. Shared with MCP



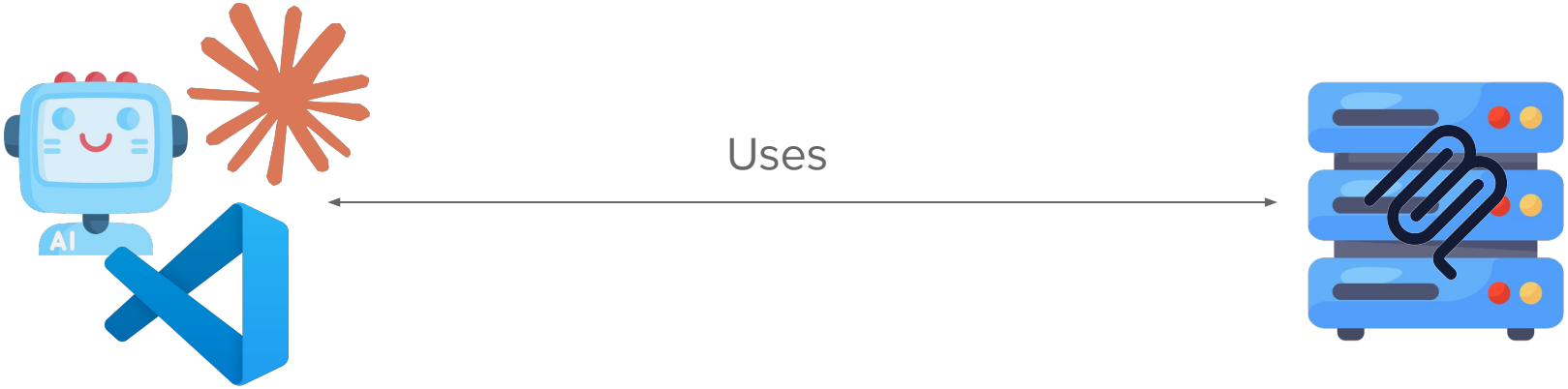
MCP define how AI agents access and utilize external data, tools, and services

https://bit.ly/2025-nov1-mcp

What is MCP?

Standard protocol for a client to connect with external data sources and tools

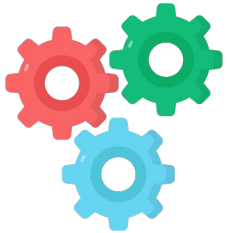
- Client is called the MCP host
- External data source and tools is the MCP server



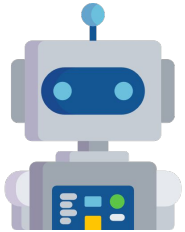
https://bit.ly/2025-nov1-mcp

Yet Another Way of Communication

1995 onwards Human - HTML



2000 onwards Applications - SOAP/REST



2025 onwards Applications - MCP

https://bit.ly/2025-nov1-mcp

Task 1

Project Setup

https://bit.ly/2025-nov1-mcp

Task 1

Initialize with uv - **`uv init`**

Add FastMCP - **`uv add fastmcp`**

- https://gofastmcp.com/getting-started/welcome

Create a file call **`todo-mcp.py`**

Re download the file **`todo_db.py`** ⚠️

The updated version has the timestamp in the first line

https://bit.ly/2025-nov1-mcp

Task 2

Tools

https://bit.ly/2025-nov1-mcp

MCP Server Primitives

Controlled by the AI

Tool (model-controlled) - a function that can be invoked to perform an action and return a result
Eg. making an API call, sending an email, etc

Controlled by the user

Resource (application controlled) - expose data and content that can be read by clients and used as context
Eg. JIRA ticket details, graph, etc

Prompt (user controlled)- reusable prompt templates and workflows
Eg. Chain-of-thought prompt for scaffolding a CRUD application

https://bit.ly/2025-nov1-mcp

MCP Server Primitives

DONT USE THIS

Sampling - server request the client to perform completions on its behalf

Eg. request the client to rate if images is appropriate

Allows long running interactive workflows

Elicitation - request information from the user

Eg. request password to push code to Git

Control what MCP should see or work on

Roots - define the boundaries where servers can operate, provide guidance what category of resources to use

Eg. narrow the JIRA to only fred's open tickets

https://bit.ly/2025-nov1-mcp

MCP Client Support

https://modelcontextprotocol.info/docs/clients/

Feature support matrix

Client	Resources	Prompts	Tools	Sampling	Roots	Notes
5ire	❌	❌	✅	❌	❌	Supports tools
BeeAI Framework	❌	❌	✅	❌	❌	Supports tools in agentic workflows
Claude Desktop App	✅	✅	✅	❌	✅	Full support for all MCP features, including roots
Claude Code	✅	✅	✅	❌	✅	Claude Code programming assistant with roots support
Cline	✅	❌	✅	❌	❌	Supports tools and resources
Continue	✅	✅	✅	❌	❌	Full support for all MCP features

https://bit.ly/2025-nov1-mcp

TODO MCP

MCP to manage `#TODO` comments in the source code

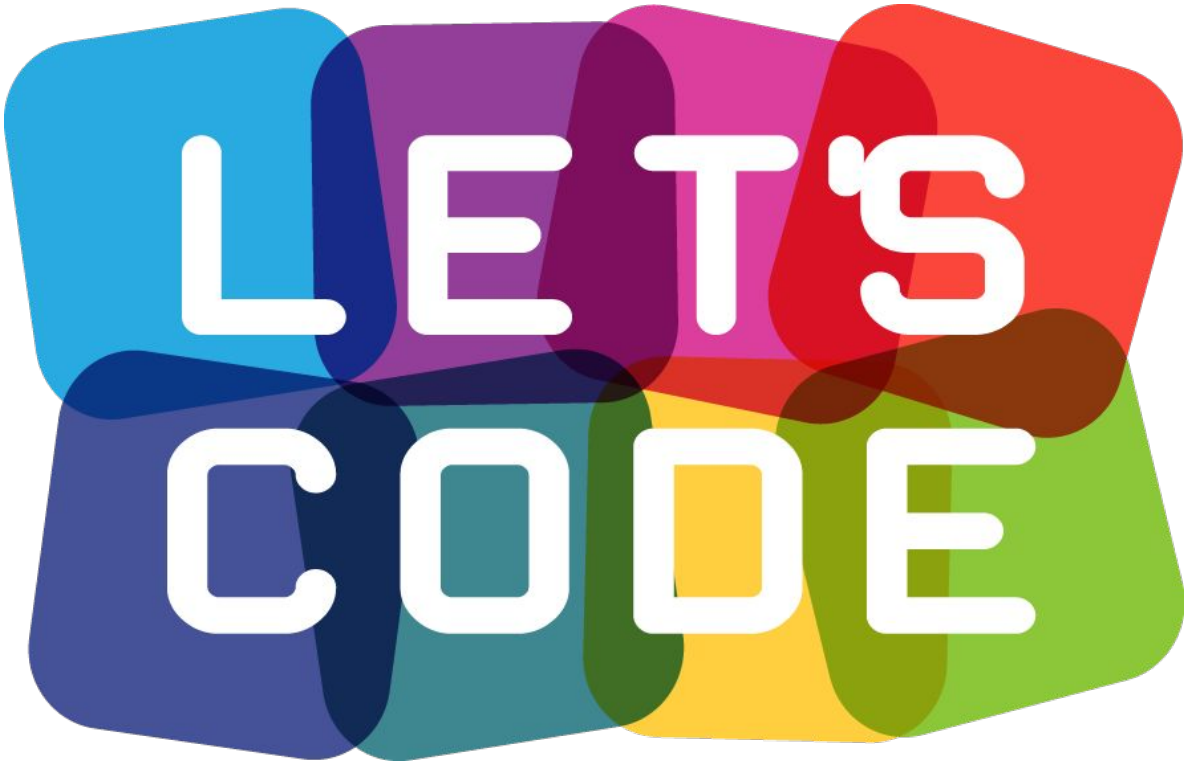
Can be used in IDE that supports MCP

Will implement the following primitives

- Tools
- Resources
- Prompts

https://bit.ly/2025-nov1-mcp

Task 2



https://bit.ly/2025-nov1-mcp

Task 3

Running in MCP Host

Install MCP

fastmcp install mcp-json my-mcp.py

```
{
  "TODO mcp": {
    "command": "/home/fred/.local/bin/uv",
    "args": [
      "run",
      "--with", "fastmcp",
      "fastmcp", "run",
      "/home/fred/src/mymcp/my-mcp.py"
    ]
  }
}
```

Add to MCP configuration file

Better to change to full path

https://bit.ly/2025-nov1-mcp

Task 4

Installing MCP

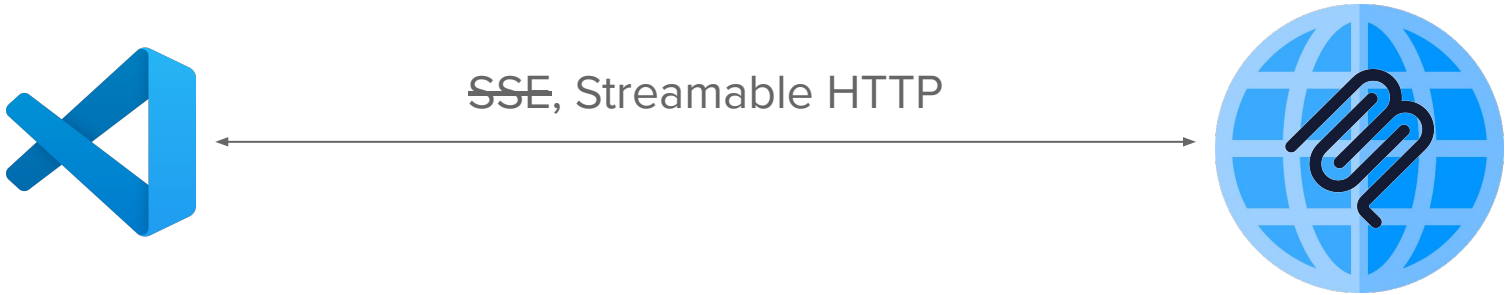
Deployment Model - Local

MCP server runs locally on your OS

- One client per instance
- Uses STDIO protocol for communication

Supports on demand install with npx and vux

Deployment Model - Remote



MCP server runs remotely

- Supports multiple client like a web application
- Uses streamable HTTP for communication

pyproject.toml

```
[project.scripts]
todo-mcp = "todo_mcp:main"

[build-system]
requires = [ "setuptools" ]
build-backend = "setuptools.build_meta"

[tool.setuptools]
package-dir = { "" = "." }

[tool.setuptools.packages.find]
where = [ "." ]
```

1. Test it

   uv run todo-mcp

2. Push code to Git repo

On-Demand Installation and Run

uvx run --from=git+https://github.com/fred/todo-mcp.git **todo-mcp**

```
"todo-mcp": {
  "command": "/home/fred/.local/bin/uvx",
  "args": [
    "--from",
    "git+https://github.com/fred/todo-mcpgit",
    "todo-mcp"
   ]
}
```


Thank You!