The Rise of Model Context Protocol (MCP) in Docker Desktop

Raveendiran RR & Ajeet Singh Raina

Meet Raveendiran RR

- **18 + years of IT experience**
- **Roles :**
  - **Docker Community Speaker | Generative AI | LLM Ops**
  - App Development| CoE | Low-code No-code| SAP BTP | SAP Build Apps| Chatbots | SAP ERP/SuccessFactors support +Implementation
- **Passionate about technology and innovation**

https://www.linkedin.com/in/raveendiranrr/
https://dev.to/raveendiran

Blueprint Technologies
Our Capabilities Our People

Meet Ajeet

- DevRel at Docker
- Former Docker Captain
- Docker Community Leader
- Distinguished Arm Ambassador
- Worked at Dell EMC, VMware, Redis

@ajeetsraina

"The future isn't just about AI,

it's about AI that acts."

Over to you, Raveendran !!

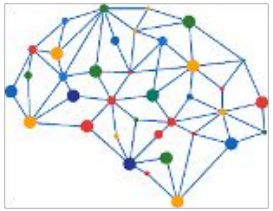
Generative AI Evolution

https://sebastian-petrus.medium.com/top-10-rag-frameworks-github-repos-2024-12b2a81f4a49

Advent of generative AI

Retrieval Augmented Generation (RAG)

AI Agents + Model Context protocol

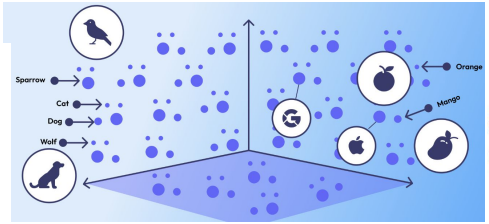


LangChain

Vector |Graph Databases
AI Frameworks :
Langchain

AI Integrations
AI tools

n8n



neo4j

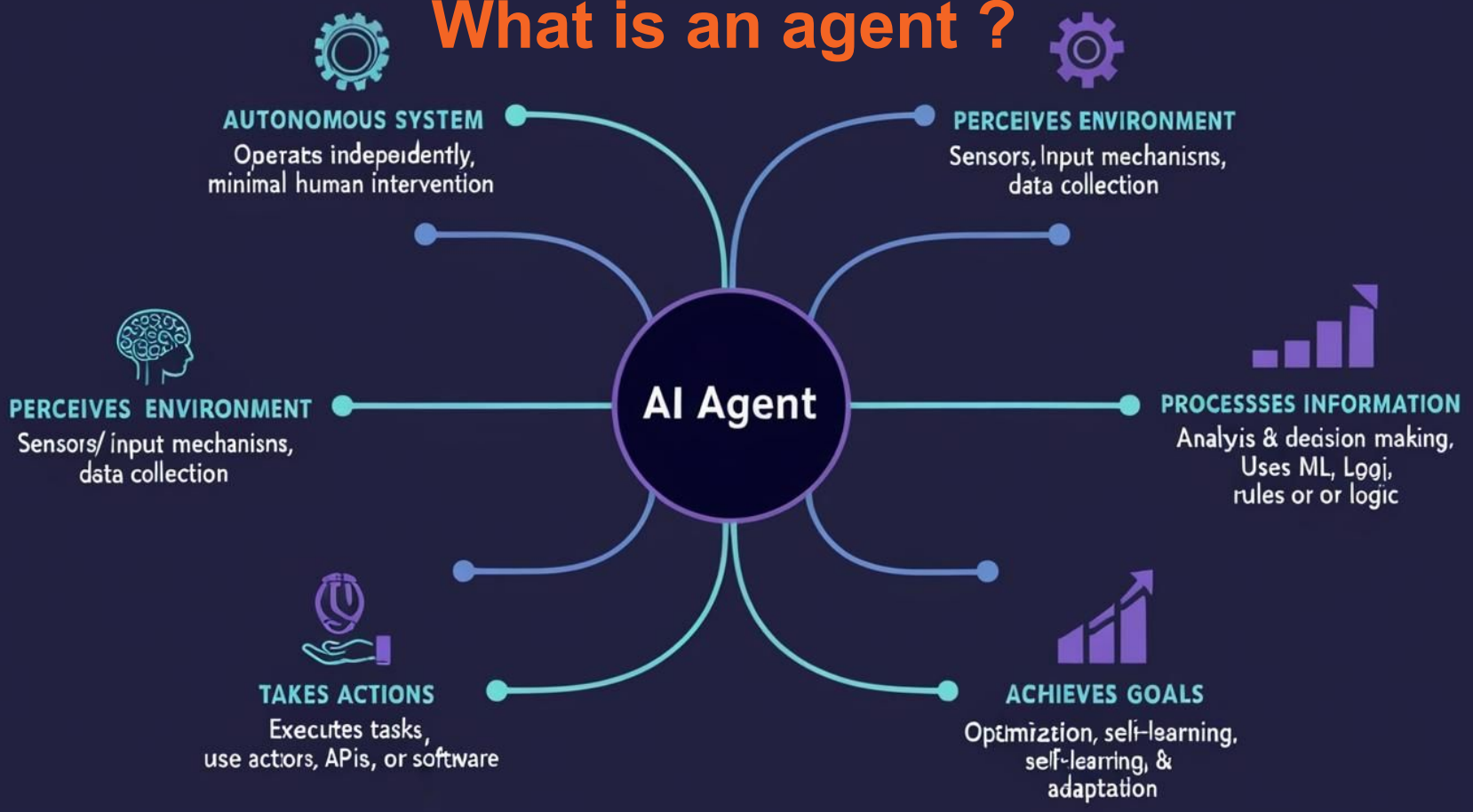
FlowiseAI

Langflow

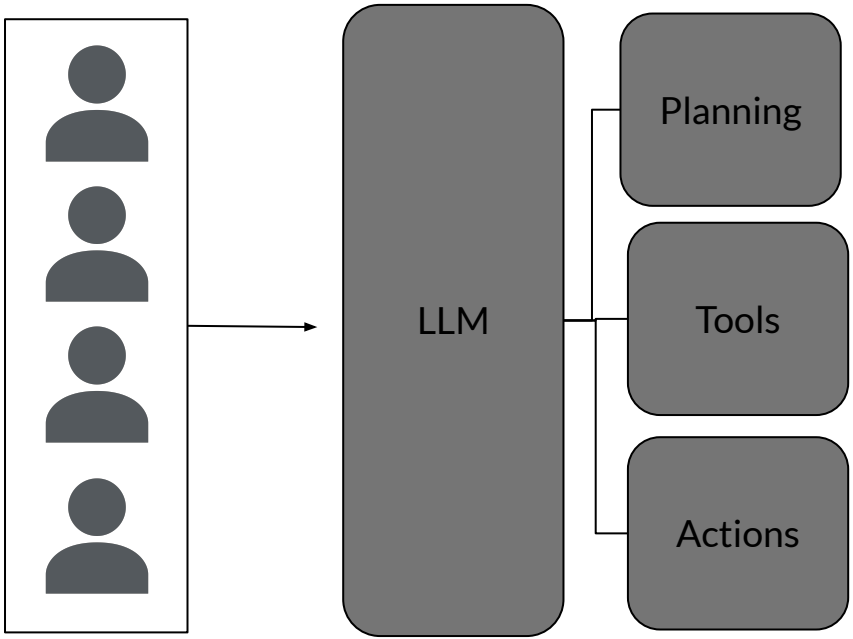
MS Power Automate

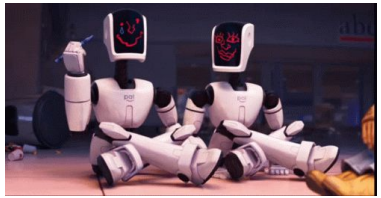
outsystems

What is an agent ?



Agent working

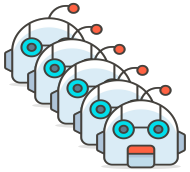




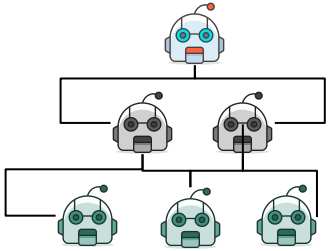
1. Custom Code
2. Apps
3. DB's
4. API

Agent Design Patterns

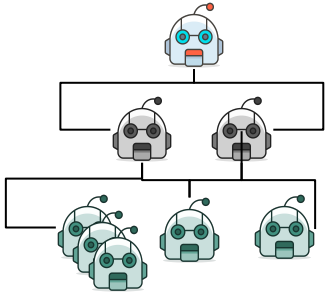
Agent types



Sequential

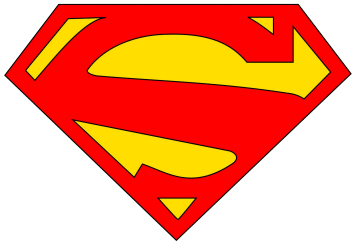


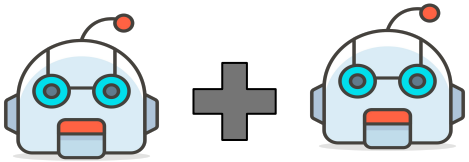
Hierarchical



Hybrid

Agents are more effective with tools





Function and tool Calling – Need for a Standard

OpenAI

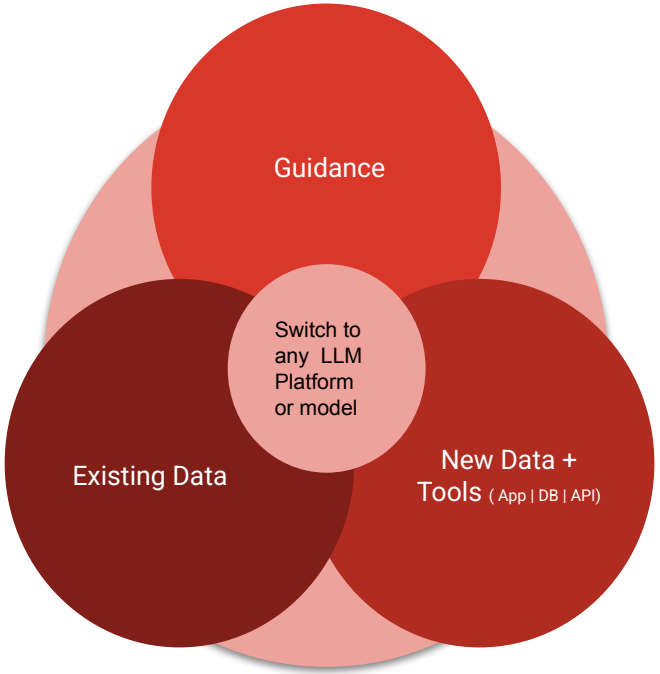
Gemini

MCP

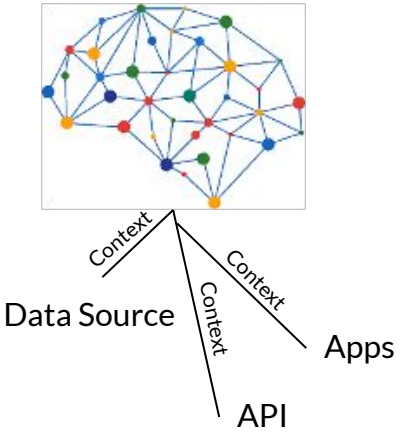
Standard Protocol on how to use tools

What’s unique about Model Context Protocol?

Why MCP?

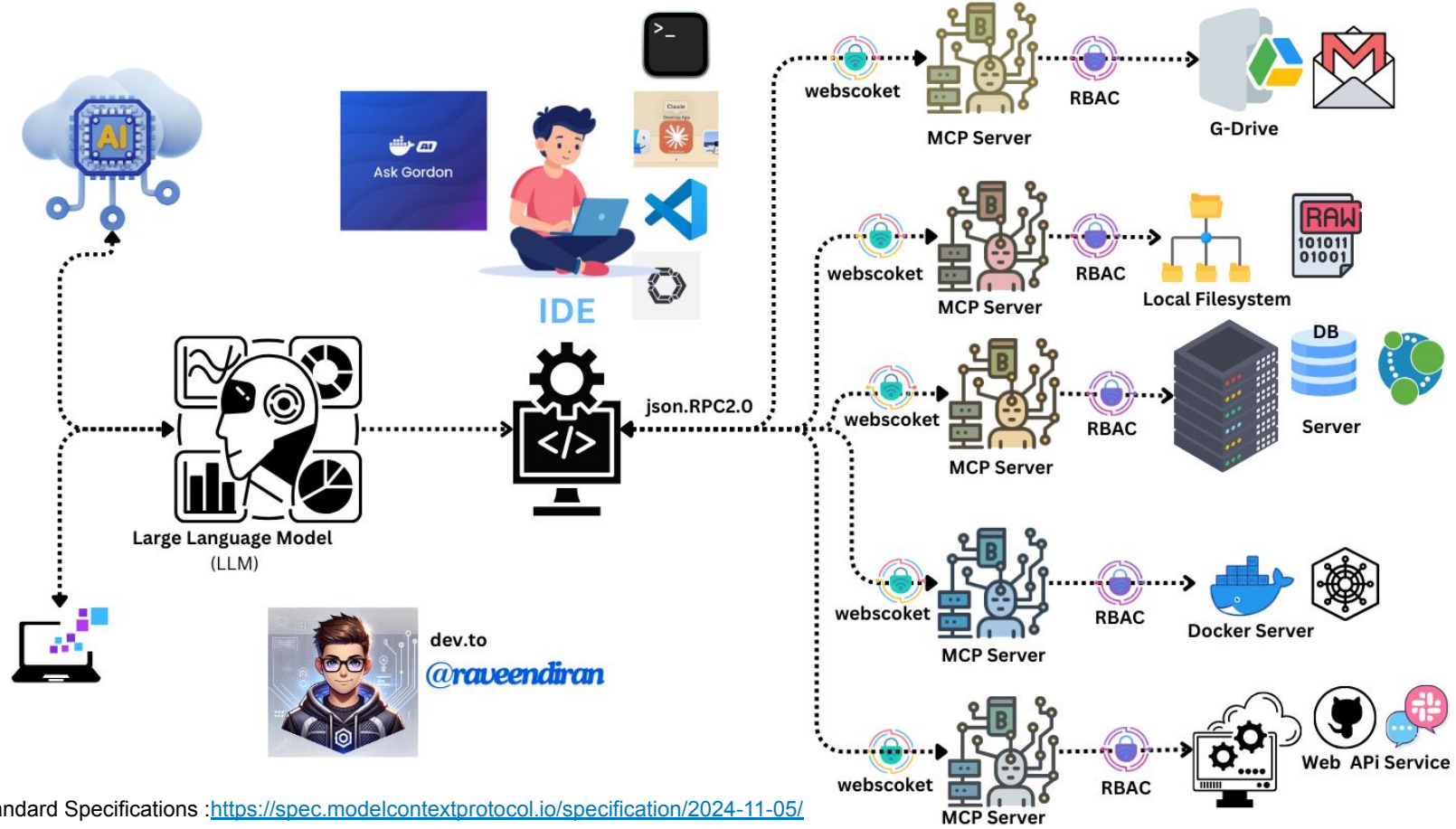


LLM Model



- It provides instant and easy integration between LLM and external tool.
- Freedom to switch between LLM providers.
- Secure data handling within infrastructure.

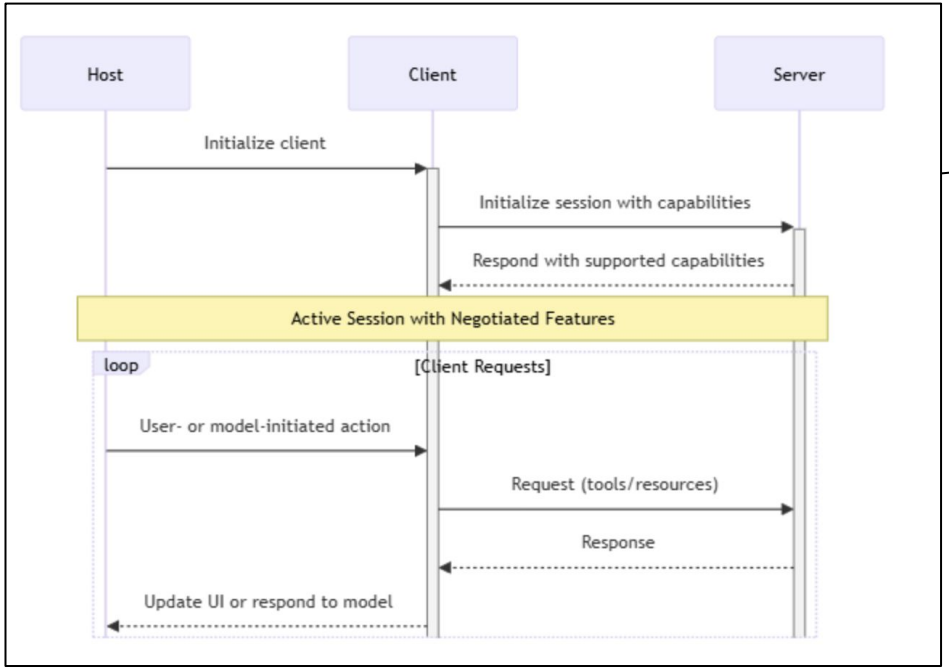
MODEL CONTEXT PROTOCOL (MCP) Architecture

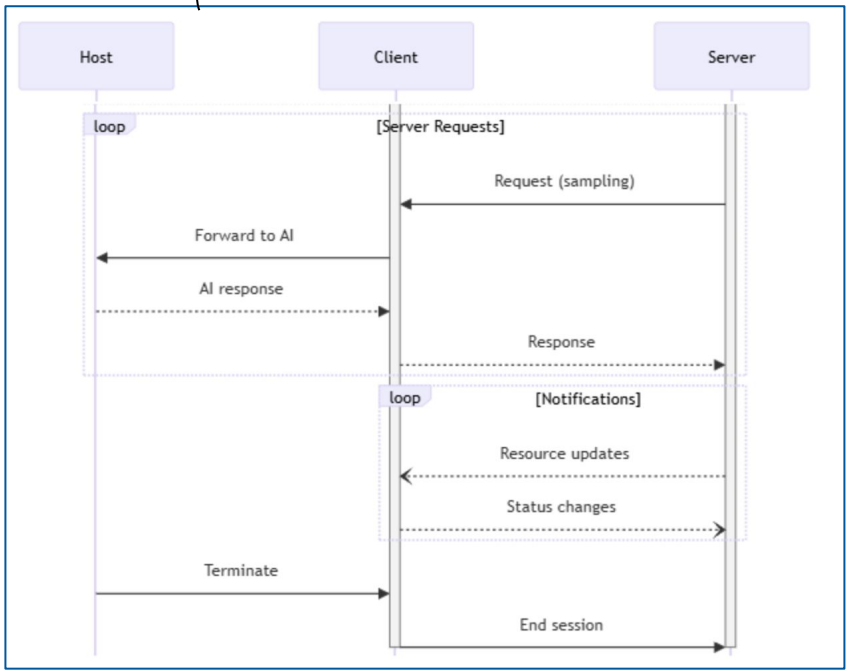


MCP Standard Specifications :https://spec.modelcontextprotocol.io/specification/2024-11-05/

MCP Message Types

Requests | Responses | Notifications

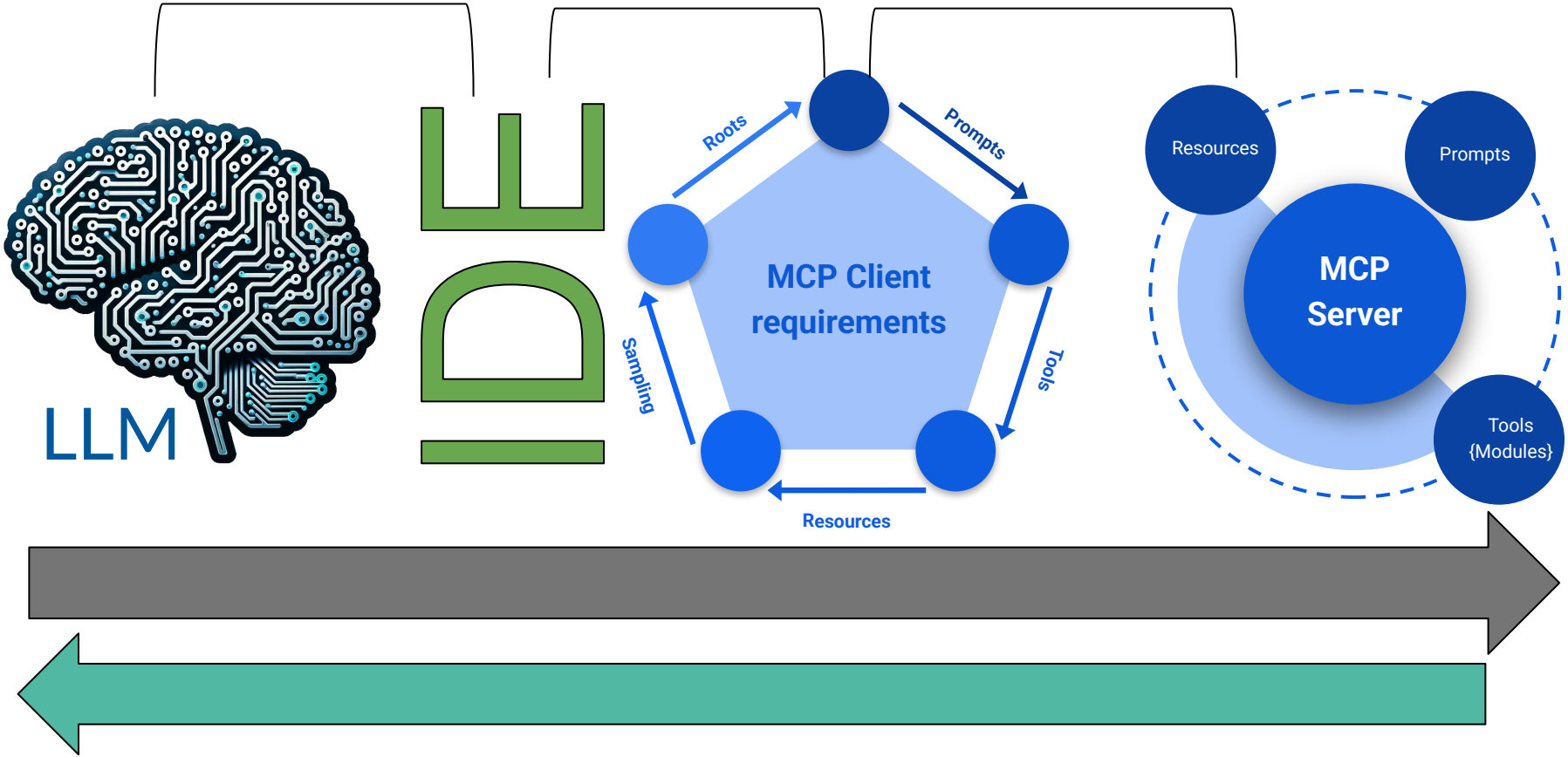




Host with MCP Client: These function as connectors, establishing one-to-one communication links between the host and MCP servers. A single MCP host can manage multiple client instances.

MCP Servers: The core components responsible for executing specific tasks or functions. They utilize the Model Context Protocol (MCP) to expose defined features or capabilities.

MCP Clients and Servers



Getting Started

1. Install Claude Desktop | IDE
2. Connect to the required GenAI Model
3. Install SDK - Python | Type Script
4. Install and create custom MCP Module
5. Update the IDE with the MCP Server Configuration

https://hub.docker.com/u/mcp

Install docker and use the docker container for quick prototyping of MCP

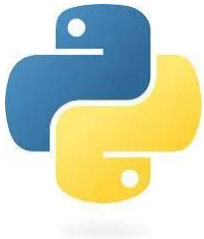
Tool Integration with MCP

- Data and filesystems
- Development tools
- Web Browser automation
- Productivity and communication
- Other AI tools

https://mcp.so/category/developer-tools

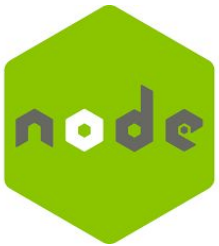
https://hub.docker.com/u/mcp

Install Server Modules



```
pip install mcp
```

```
npm install @modelcontextprotocol/sdk
```



Demo

What are we going to do?

Install Docker Desktop

Local Git Setup

Setup Git MCP Server

Using Claude Desktop

Installing Docker Desktop

Setup Git MCP Server

▼ Using uvx

```
"mcpServers": {
  "git": {
    "command": "uvx",
    "args": ["mcp-server-git", "--repository", "path/to/git/repo"]
  }
}
```

▼ Using docker

- Note: replace '/Users/username' with the a path that you want to be accessible by this tool

```
"mcpServers": {
  "git": {
    "command": "docker",
    "args": ["run", "--rm", "-i", "--mount", "type=bind,src=/Users/username,dst=/Users/username", "mcp/git"]
  }
}
```

▼ Using pip installation

```
"mcpServers": {
  "git": {
    "command": "python",
    "args": ["-m", "mcp_server_git", "--repository", "path/to/git/repo"]
  }
}
```

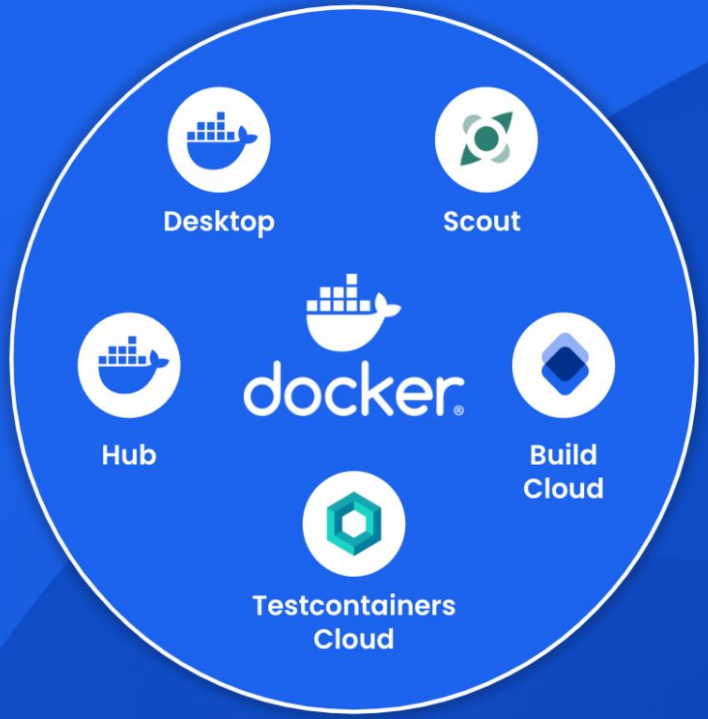
Usage with Zed

Over to you, Ajeet !!

Current Challenges with MCP servers

1. **Environment conflicts**: Installing MCP servers often requires specific versions of Node.js, Python, and other dependencies, which may conflict with existing installations on a user's machine
2. **Lack of host isolation**: MCP servers currently run on the host, granting access to all host files and resources
3. **Complex setup**: MCP servers currently require users to download and configure all of the code and configure the environment, making adoption difficult
4. **Cross-platform challenges**: Running the servers consistently across different architectures (e.g., x86 vs. ARM, Windows vs Mac) or operating systems introduces additional complexity
5. **Dependencies**: Ensuring that server-specific runtime dependencies are encapsulated and distributed safely.

How does Docker Help?



Docker solves these challenges by providing a standardized method and tooling to develop, package, and distribute applications, including Model Context Protocol servers.

- **Docker Desktop** provides a development platform to build, test, and run these MCP servers
- **Docker Hub** is the world's largest repository of container images, making it the ideal choice to distribute containerized MCP servers
- **Docker Scout** helps ensure images are kept secure and free of vulnerabilities.
- **Docker Build Cloud** helps you build images more quickly and reliably, especially when cross-platform builds are required.

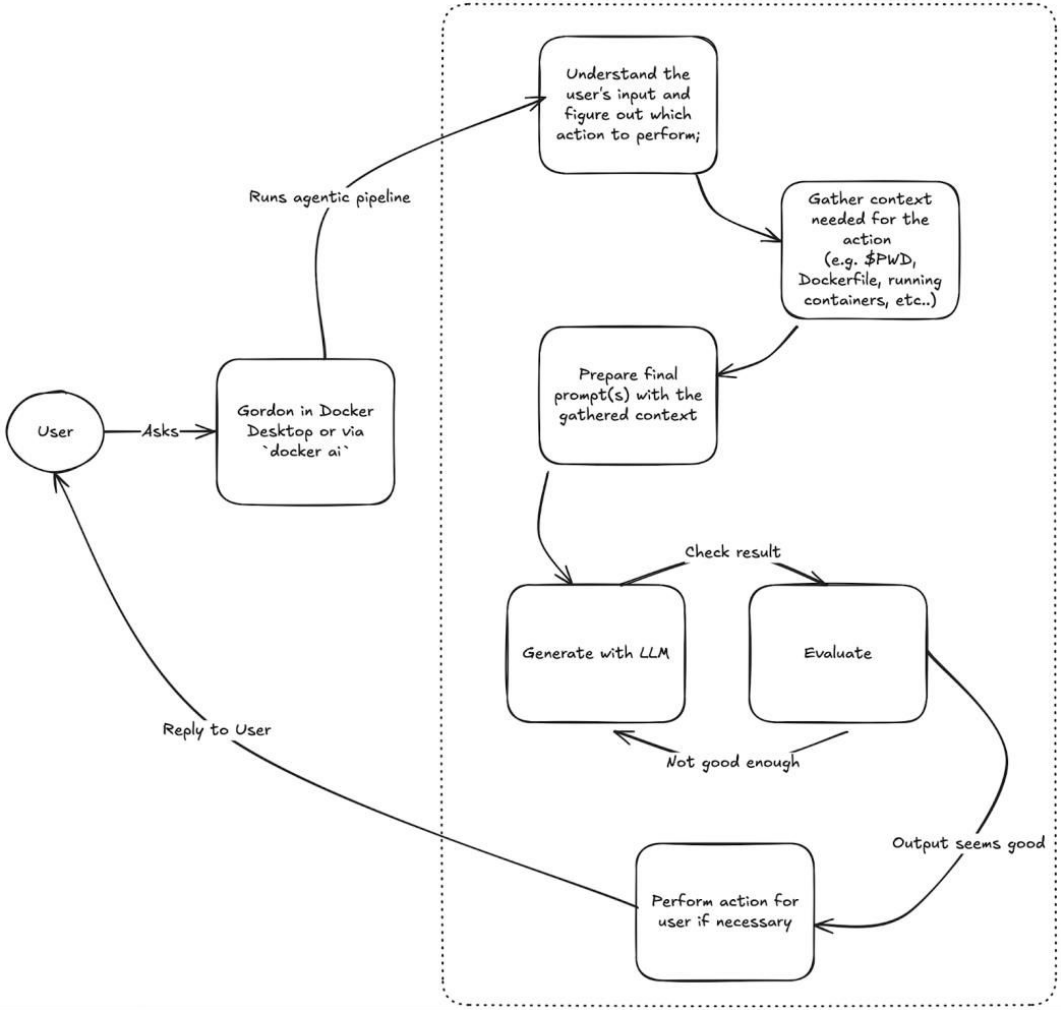

AI
Ask Gordon

Docker AI Agent (Project: Gordon)

1. Docker's new context-aware assistant
2. Integrated into Docker Desktop and CLI
3. Provides real-time guidance for container operations
4. Eliminates context-switching in development workflow

Key Features:

1. Dockerfile optimization and rating
2. Smart container running assistance
3. Context-aware troubleshooting
4. Project containerization help
5. GitHub Actions integration
6. Contextual container management

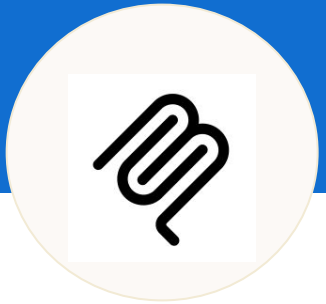

Understand the user's input and figure out which action to perform;
Runs agentic pipeline
Gather context needed for the action (e.g. $PWD, Dockerfile, running containers, etc..)
Prepare final prompt(s) with the gathered context
User
Asks
Gordon in Docker Desktop or via `docker ai`
Check result
Generate with LLM
Evaluate
Reply to User
Not good enough
Output seems good
Perform action for user if necessary

It’s Demo Time

What are we going to do?

Install Docker Desktop

Configure Docker AI Agent



Add MCP Servers

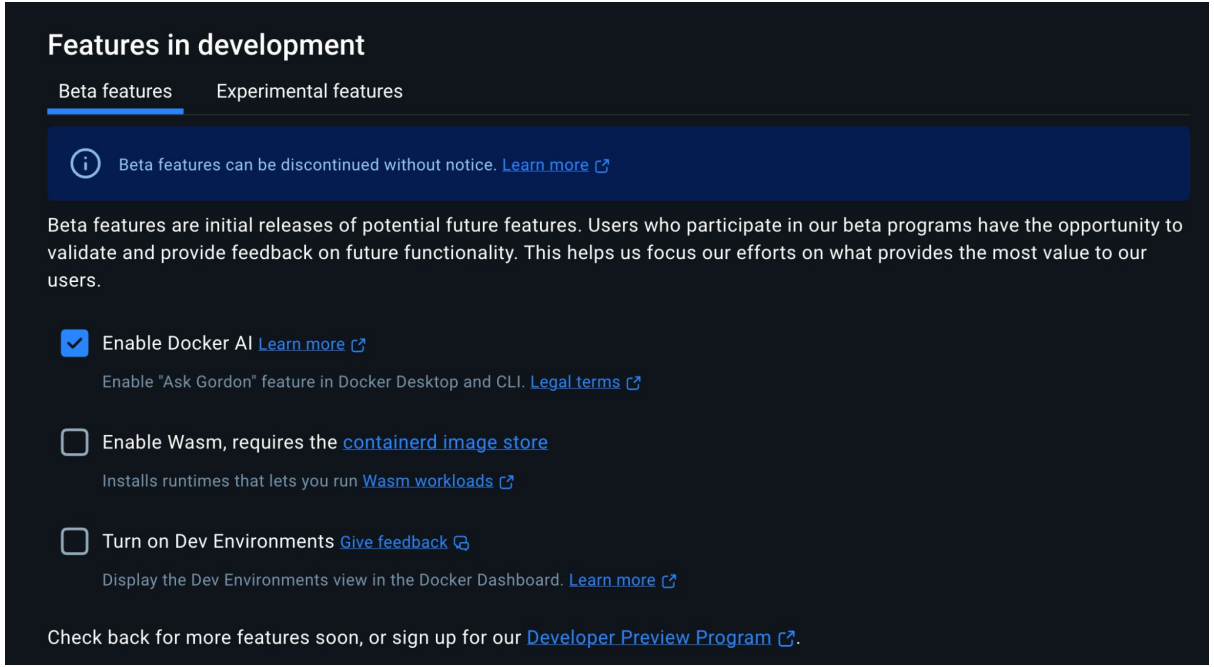
- mcp/time
- mcp/postgres
- mcp/github
- mc/git

$ gordon-mcp.yml

Using Docker AI Agent

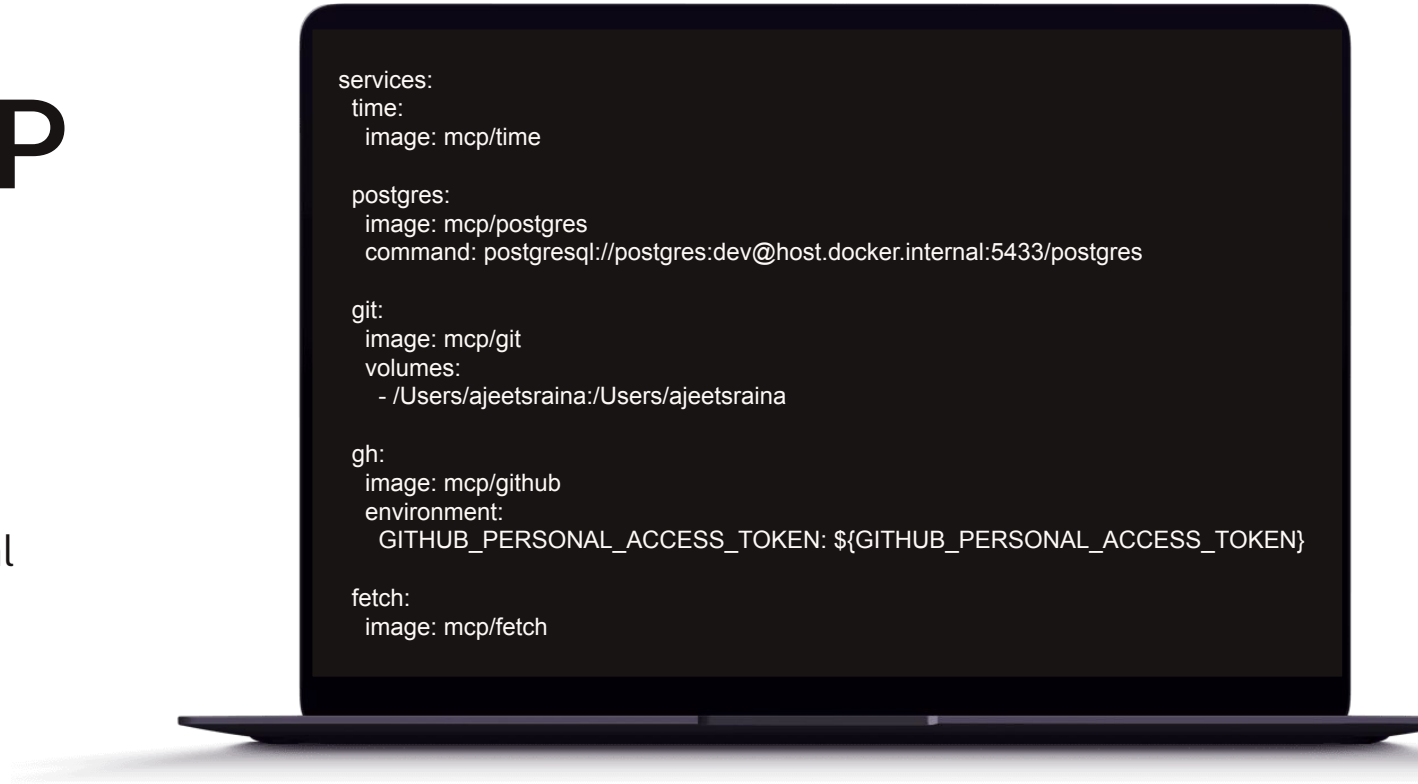
Installing Docker Desktop

Enable Docker AI Agent

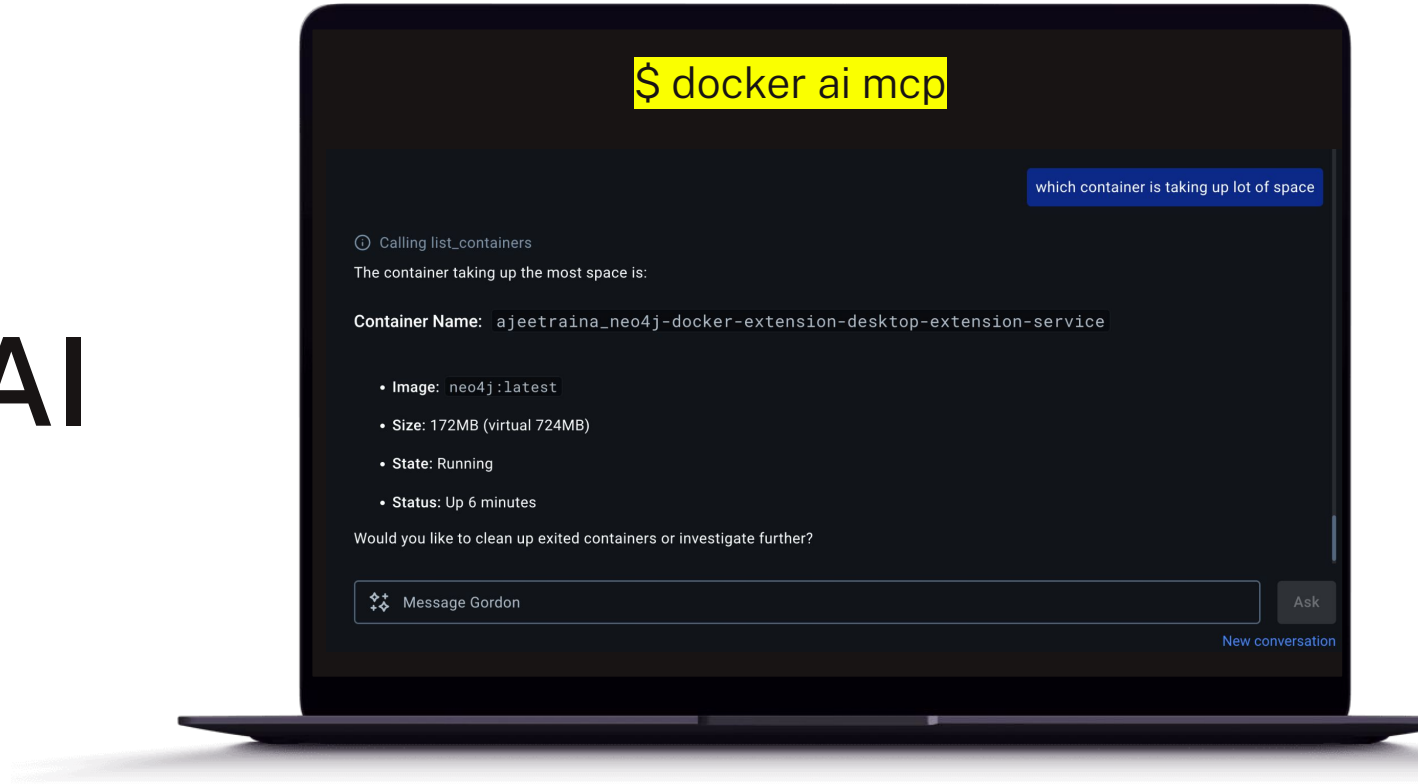


Add MCP Servers

$ cat gordon-mcp.yml



Using Docker AI Agent



References

- https://dev.to/ajeetraina/docker-ai-agent-and-model-context-protocol-mcp-server-working-together-4c4l
- https://collabnix.com/postgres-and-model-context-protocol/
- https://www.docker.com/blog/the-model-context-protocol-simplifying-building-ai-apps-with-anthropic-claude-desktop-and-docker/
- https://github.com/Flux159/mcp-server-kubernetes

Join our Slack Community