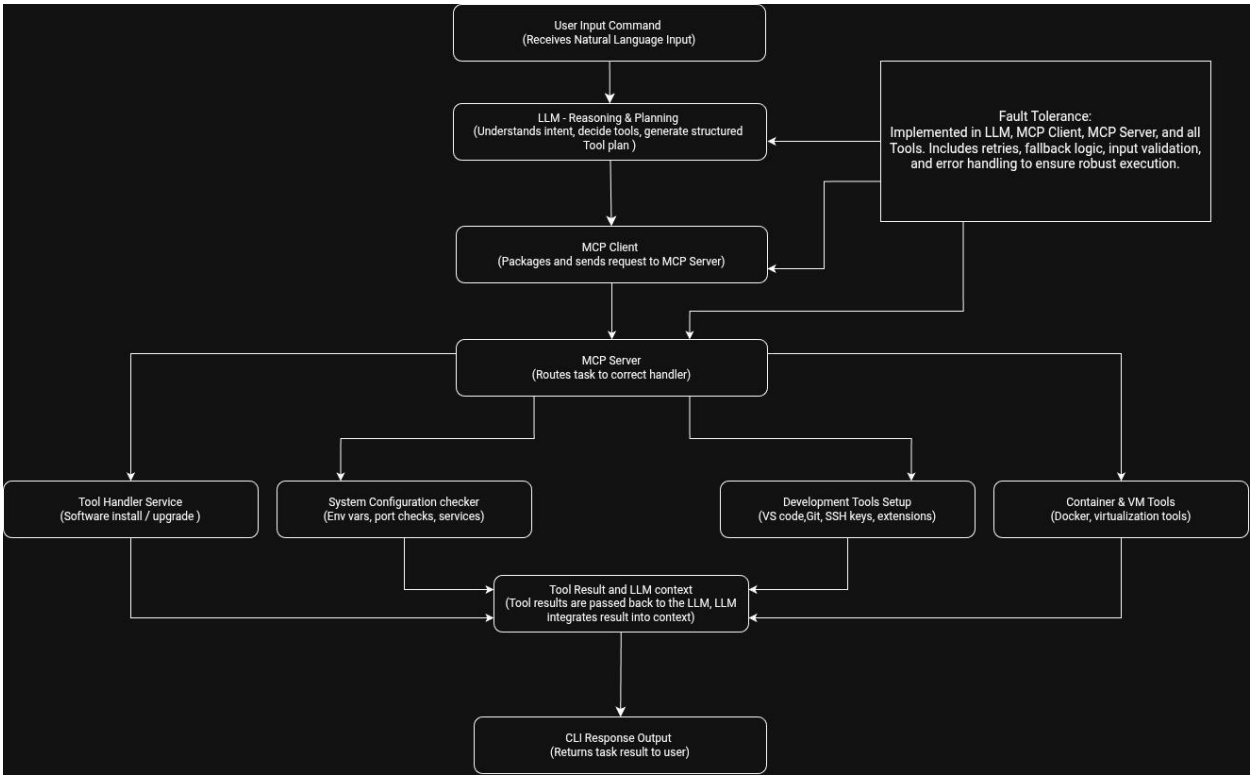
AI CLI Agent – System Architecture Summary

This architecture outlines the workflow of our intelligent CLI agent, which automates development environment setup based on user commands. The system is powered by an LLM for understanding instructions and uses the Model Context Protocol (MCP) to communicate with modular tool servers.



Workflow Breakdown

1. User Input Command

Users begin by entering a natural language instruction into the terminal, such as “Install Docker and set JAVA_HOME.” This input triggers the agent workflow.

2. LLM – Reasoning & Planning

The language model interprets the user’s intent, extracts necessary components, and plans the action. It selects the appropriate tool from the available options and generates a structured function call (MCP-compatible) for execution.

3. MCP Client

The MCP Client receives the tool call from the LLM, packages it into the proper protocol format, and routes it to the appropriate MCP Server. It ensures correct formatting and communication.

4. MCP Server

The MCP Server acts as the central dispatcher. It parses incoming requests and delegates the task to the correct service handler based on the tool name and context.

5. Tool Handler Services

Each tool handler is an independent service responsible for executing specific system-level tasks. The main categories include:

- **Tool Installer:** Installs or upgrades tools like Node.js, Java, Docker, etc.
- **System Config Checker:** Verifies or modifies environment variables, port status, and running services.
- **Dev Tools Setup:** Handles Git configuration, code editor setup, extension installation, and SSH key generation.
- **Container/VM Tools:** Manages Docker, Compose, or virtual machine environments.

6. Tool Result & LLM Context

The result of the tool execution is sent back from the MCP Server through the Client to the LLM. The model integrates this result into its ongoing reasoning process to determine the next step or finalize the output.

7. CLI Output

Finally, the agent prints a human-readable message or result in the terminal, completing the interaction loop with clear feedback to the user.