Manus AI Architecture: A Detailed Workflow

This document explains the complete, step-by-step process of how Manus AI operates, from receiving a user's request to delivering the final result. The architecture is designed for maximum safety, efficiency, and autonomy.

The Core Concept: The Agentic Loop

Think of Manus as a **Digital Assistant** with its own computer. When you give a task, Manus doesn't just talk about it; it creates a plan and uses its tools (like a web browser or code editor) to execute the work in a secure, isolated environment.

The Full Workflow: Step-by-Step

The entire process is broken down into six distinct layers, ensuring that every request is handled safely and accurately.

1. Input & Safety Layer (The Gatekeeper)

The moment your message arrives, it goes through a parallel check system:

Component	Simple Explanation	Purpose
System Gateway	The entry point for your message.	Receives and routes the request.
Safety Guardrails	A parallel security check.	Scans the message for harmful, illegal, or policy-violating content. If a violation is found, the process stops immediately.
Context Retrieval	Pulling up your history.	Gathers all previous messages and your preferences to understand the full context.
Intent Analysis	Figuring out what you want.	Determines the main goal of your request (e.g., “Write a report,” “Build a website,” “Find information”).

2. Planning & Strategy (The To-Do List Maker)

Once the intent is clear, Manus creates a detailed plan:

- **Planning Engine:** This is where the overall strategy is formed.
- **Goal Decomposition:** The main goal is broken down into smaller, manageable steps called “Phases” (e.g., Phase 1: Research, Phase 2: Code, Phase 3: Deliver).

3. Execution Loop (The Decision Maker)

This is the heart of the system, where Manus decides what to do next:

- **Reasoning & Decision:** Manus looks at the current step in the plan and decides the best action to take (e.g., “I need to search the web, so I will use the Browser tool”).
- **Tool Execution:** The decision is converted into a specific command for one of the available tools.

4. Sandbox Environment (The Isolated Workspace)

This is the secure, isolated virtual computer where all the work is done. It prevents any outside interference and ensures safety.

Tool	Function
Browser	Used for web searches, navigating URLs, and interacting with web applications.
Shell	Used for installing software, managing files, and running system commands.
Code Runtime	Used for running Python or Node.js scripts for data analysis, automation, or software development.

5. Feedback & Verification (The Quality Check)

After a tool runs, Manus checks the result:

- **Observation & Logs:** The output from the tool (e.g., website content, terminal output) is collected.
- **Internal Critic:** This is a self-correction mechanism. It checks if the result matches what was expected. If an error occurred (e.g., a website was down), the Critic sends a signal back to the Reasoning stage to try a different approach.

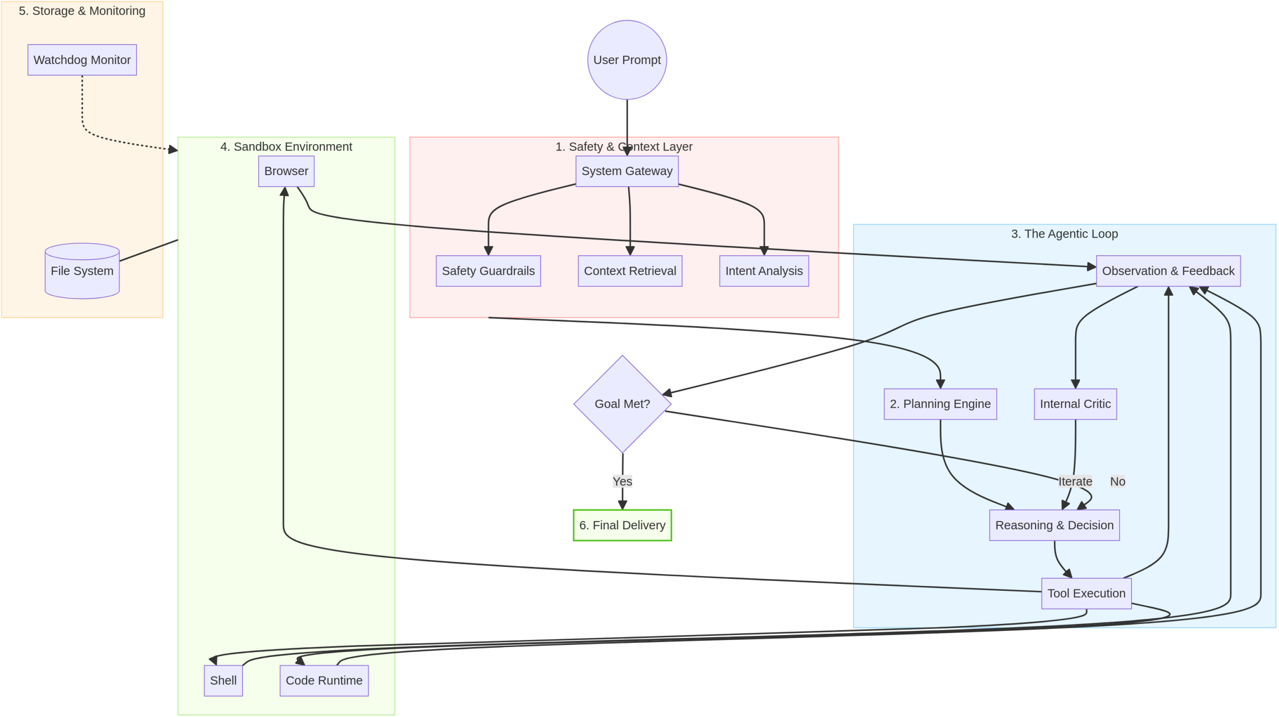
6. Finalization (The Delivery)

The loop continues until the entire plan is complete:

- **Goal Met?** A final check to ensure all requirements are satisfied.
- **Synthesis & Artifacts:** All files, reports, and data are gathered and prepared.
- **Final Result:** The complete answer and all attachments are delivered to you.

Visual Representation of the Architecture

The following diagram illustrates the clean, layered flow of the Manus architecture.


5. Storage & Monitoring
Watchdog Monitor
File System
User Prompt
4. Sandbox Environment
Browser
1. Safety & Context Layer
System Gateway
Safety Guardrails
Context Retrieval
Intent Analysis
3. The Agentic Loop
Observation & Feedback
Goal Met?
2. Planning Engine
Internal Critic
Yes
Iterate
No
6. Final Delivery
Reasoning & Decision
Tool Execution
Shell
Code Runtime