

Claude Code Source Code Deep Dive Report (Enhanced Version)

Author: Xiao Tan

Date: March 31, 2026

Overview:

This report presents a systematic analysis of the Claude Code source code reconstructed from cli.js.map. It focuses on architecture, system prompts, agent orchestration, tools, permissions, plugins, hooks, MCP integration, and runtime behavior.

Key Conclusion:

Claude Code is not just a prompt or tool wrapper. It is a full "Agent Operating System" that integrates:

- Prompt architecture
- Tool execution pipeline
- Permission model
- Agent orchestration
- Skill and plugin systems
- Hooks and governance
- MCP integration
- Context optimization

Core Insights:

1. Prompts are dynamically assembled at runtime, not static text.
2. Tools are executed through a governed pipeline (permissions, hooks, validation).
3. Agents are specialized (Explore, Plan, Verification, etc.).
4. Skills are reusable workflow packages.

5. Plugins extend both behavior and runtime constraints.
6. MCP injects both tools and behavioral instructions.
7. Context and token efficiency are treated as critical resources.

System Architecture:

- Entry points: CLI, SDK, MCP
- Tools layer: enables real-world actions
- Services: runtime systems (tools, MCP, analytics)
- Commands: user control interface
- Plugins / Hooks: extensibility and governance

Agent System:

- Multi-agent orchestration with fork and specialization
- Background and foreground execution modes
- Lifecycle management with transcripts and cleanup

Final Summary:

Claude Code's strength comes from treating AI as a structured system rather than a single model. It unifies prompts, tools, permissions, agents, and runtime into a coherent, production-grade platform.