

# Claude Code 源码架构深度分析

## 整体产品定位

这是 **Claude Code** 的核心源码库 — Anthropic 官方的 AI 编程助手 CLI 工具。共 **1902 个 TypeScript 文件**，是一个高度模块化的大型工程。

Claude Code 是一个**终端原生的 AI 编程环境**，核心能力包括：

- 在终端中与 Claude 模型交互，执行代码编辑、文件操作、Shell 命令等
- 支持多 Agent 协作（团队模式、并行子代理）
- 支持远程会话（claude.ai 网页端控制本地 CLI）
- 内置 Vim 编辑模式、语音输入、MCP 协议集成
- 插件/技能系统实现高度可扩展性

## 核心架构分层

层级	职责
用户交互层	CLI 入口 / 终端 UI (Ink) / Vim 编辑器 / 语音输入
命令与技能层	100+ Slash 命令 / 插件系统 / 技能加载器
核心引擎层	QueryEngine（对话编排） / Tool 系统 / 权限框架
服务层	Claude API / MCP 协议 / OAuth / LSP / Analytics
通信层 (Bridge)	远程会话 / WebSocket/SSE 传输 / JWT 认证 / 断线恢复
基础设施层	状态管理 / 配置系统 / Hook 注册表 / 文件系统 / Git 集成

## 核心模块详解

### 1. QueryEngine — 大脑中枢

 QueryEngine.ts 是整个产品的核心编排引擎，13000+ 行代码，每个对话实例一个 QueryEngine。

- **对话生命周期管理**：消息排序、中止控制、多轮状态维持
- **系统 Prompt 组装**：整合 CLAUDE.md 记忆文件、Git 上下文、项目信息
- **工具执行编排**：调度 45+ 内置工具，处理并发、进度追踪
- **Snip 压缩**：长会话中自动截断历史消息，保持语义连续性
- **推测执行 (Speculation)**：流水线式预生成建议，提升响应速度
- **Token 计数 & 成本追踪**：实时统计 API 用量

## 2. Tool 系统 — 45+ 内置工具


每个工具都是一等公民，拥有完整的权限、渲染、进度追踪机制。

工具类别	示例
文件操作	FileRead, FileEdit, Write, Glob, Grep
执行环境	BashTool (Shell 命令执行, 带 stdio 捕获)
代理系统	AgentTool (并行子代理 / 团队模式)
用户交互	AskUserQuestion (交互式提问)
工作流	EnterPlanMode, EnterWorktree (计划模式 / Git 工作树)
外部集成	MCPTool, SkillTool, LSPTool

工具核心特性：

- **延迟加载**：非核心工具通过 ToolSearch 按需加载，优化启动性能
- **权限网关**：每个工具调用前必须通过多层权限检查
- **进度追踪**：实时 spinner 和活动描述
- **MCP 集成**：支持外部 MCP 服务器的工具发现与调用

## 3. Ink 终端 UI 框架 — 自研渲染引擎

 这不是简单的 Ink 库包装，而是一个完整的 React-based 终端渲染引擎，80+ 文件。

## 渲染管线：

React 组件树 → 自定义 Reconciler → DOM 树 → Yoga Flexbox 布局 → Output Buffer → Screen Buffer (双缓冲) → Diff → ANSI 输出


## 核心能力：

- **Yoga Flexbox 布局：**完整的 flexbox 支持
- **双缓冲渲染：**前后帧原子交换，16ms 节流
- **硬件滚动优化：**DECSTBM 滚动区域指令
- **文本选择：**全屏模式下字符/单词/行级选择
- **事件系统：**捕获/冒泡两阶段传播
- **终端协议探测：**Kitty/xterm/CSI u 自动识别

## 性能优化亮点：

1. 双缓冲帧交换 + 脏标记传播
2. Blit 快速路径 (未变更区域直接复制)
3. 字符池/样式池内存复用
4. Yoga 单槽缓存快速返回
5. 视口裁剪 (ScrollBox 仅渲染可见行)

## 4. Bridge 远程会话系统

 实现本地 CLI 与 claude.ai 网页端的双向通信，支持远程控制本地开发环境。

### 两种架构模式：

#### Environment-based (传统模式)

注册环境 → 轮询工作 → 创建会话 → 消息桥接

适合：多会话管理场景

#### Env-less (简化模式)


直接 OAuth → Session API → SSE/CCR 传输

适合：REPL 交互式会话

### 关键设计：

- **传输抽象：**v1 (WebSocket+POST) 和 v2 (SSE+CCR) 无缝切换
- **UUID 环形缓冲去重：**防止消息回声和重复投递
- **OAuth 自动刷新：**401 时自动换 token 重试
- **崩溃恢复：**bridgePointer.json 持久化 environment\\_id + session\\_id
- **控制协议：**initialize、set\\_model、interrupt、set\\_permission\\_mode

## 5. 权限系统 — 多层决策模型

 安全是 Claude Code 的核心设计原则，权限系统贯穿所有工具调用链路。

权限模式: `default` | `plan` | `acceptEdits` | `bypassPermissions` | `dontAsk` | `auto`

规则来源优先级: `cliArg` > `flagSettings` > `localSettings` > `projectSettings` > `userSettings`

决策类型:

- **allow**: 直接放行，可附加输入修改
- **ask**: 弹出用户确认，支持建议选项
- **deny**: 拒绝访问并给出解释

智能分类器:

- `bashClassifier`: Shell 命令安全评估 (识别 `rm -rf`、`DROP TABLE` 等)
- `yoloClassifier`: LLM 两阶段 XML 分类 (思考 + 决策)
- 拒绝追踪: 记录用户历史拒绝，避免级联误判

## 6. Vim 子系统 — 完整模态编辑器

 实现了完整的 Vim 模态编辑，包括状态机命令解析、Motion、Operator、Text Object。

### 状态机命令解析

10+ 状态流转: `idle` → `count` → `operator` → `find` → `textObj...`

### Motion 系统

- 字符移动: `h/j/k/l`
- 单词移动: `w/b/e/W/B/E`
- 行内定位: `0/^/$, f/F/t/T`

### Operator 系统

- `delete(d)`, `change(c)`, `yank(y)`
- `indent(>/<)`, `paste(p/P)`
- 行级操作: `dd`, `cc`, `yy`

### Text Object

- 单词: `iw/aw`, `iW/aW`
- 引号: `i"/a\"`, `i\x27/a\x27`
- 括号: `i(/a(`, `i{/a{`, `i[/a[`

## 7. 状态管理与 Bootstrap

**AppState** — Deep Immutable 全局状态树:

- 设置与配置、模型选择
- UI 状态 (展开视图、Footer)

- 推测执行状态
- 任务与代理协调
- 消息历史与解析状态
- 权限/拒绝追踪

### Bootstrap State — 会话身份载体:

- `sessionId`: 会话唯一标识
- `sessionIngressToken`: Bridge 传输认证
- API 运行时状态 (用量、成本、耗时)
- 遥测与 OpenTelemetry 集成

## 8. 命令系统 — 100+ Slash 命令

分类	命令示例
Git 工作流	<code>/commit, /diff, /branch, /pr\_comments, /commit-push-pr</code>
项目管理	<code>/tasks, /agents, /memory, /context</code>
配置管理	<code>/config, /keybindings, /theme, /settings</code>
外部集成	<code>/mcp, /skills, /chrome, /ide</code>
开发调试	<code>/doctor, /status, /cost, /debug</code>

## 9. 服务层

### API 服务

- 支持 Anthropic 直连
- AWS Bedrock
- Azure Foundry
- Vertex AI
- 自动重试与指数退避

### MCP 服务

- 多传输协议: `stdio/SSE/WS/HTTP/OAuth`
- 工具发现与权限控制
- 服务器审批 workflow

**其他服务:** OAuth 认证、语音 STT、Growthbook 特性标志、LSP 语言服务器、自动上下文压缩

# 关键架构设计亮点

设计模式	具体实现
懒加载	非核心模块 (snip, coordinator) 通过 feature() 按需加载, 优化启动速度
双缓冲渲染	前后帧原子交换 + 脏标记优化 + Blit 快速路径
传输抽象	Bridge 的 v1/v2 传输层透明切换, 应用层无感知
权限即代码	工具声明自带权限元数据, 运行时自动校验
推测执行	流水线预生成用户可能需要的下一步操作
崩溃恢复	Bridge 指针文件 + 会话重连 + 序列号跟踪
多 Agent 协调	TeamCreate/SendMessage 协议 + 共享任务列表
插件化扩展	技能(Skills) + 钩子(Hooks) + MCP 服务器三位一体

## 总结

★ Claude Code 的代码库展现了极高的工程成熟度, 是一个终端原生的 AI 开发操作系统。

- 全栈自研:** 从终端渲染引擎 (Ink) 到通信协议 (Bridge) 到编辑器 (Vim), 核心链路全部自建
- 深度优化:** 双缓冲、硬件滚动、字符池内联、样式池缓存等性能优化遍布渲染管线
- 安全至上:** 多层权限模型 + 危险命令检测 + LLM 分类器 + 用户拒绝追踪
- 可扩展性:** 45+ 工具 + 100+ 命令 + MCP 协议 + 插件系统, 生态开放
- 多模态交互:** CLI + Web 远程 + IDE 集成 + 语音输入, 全场景覆盖
- 多 Agent 架构:** 原生支持团队协作、并行子代理、协调器模式