PERI / ENGINEERING RESEARCH

Coding Agent 设计图谱

提示词 · 多智能体 · 协作 · 运行架构

指令 规则与提示词	**上下文** 检索与压缩	**执行** 工具与权限
协作 任务与消息	**状态** 恢复与历史	**验证** 评测与交付

从公开源码和官方资料出发，拆开机制、代价与适用边界。

首版：2026-10-18 / 续查：2026-10-19 R7

136 个专题页 · 425 项证据记录 · 203 页

阅读导航

先读决策摘要，再按目标选择产品与机制。正文中的来源编号可点击。

00	导读与研究边界	003
A	官方 CLI：Claude Code、Codex 与 Gemini 的提示词、协作和运行时	007
B	可组合核心：OpenCode、Crush、Pi 与 Goose	023
C	编辑器生态：Cline、Roo、Kilo、Continue 与 Tabby	039
D	软件工程代理：从编辑协议到可恢复运行时	055
E	商业 Coding Agent：上下文、协作与可信交付	071
F	从团队协议到执行框架：Qwen、Trae、Open SWE、AutoGen、MetaGPT 与 ChatDev	087
G	扩展生态：上下文、环境与产品协作	103
H	横向综合：可借鉴机制与 Peri 实验路线	119
I	覆盖补证：产品入口与框架边界	135
R	证据索引与来源	143

证据标签：源码核实 = 已读取实现；官方说明 = 文档可验证；研究综合 = 跨来源分析；设计建议 = 本报告提出，尚需验证。

先看结论：值得学习的是执行合同

导读 / 面向架构师的决策摘要

> 产品名变化很快，但值得长期学习的机制相对稳定：可解释的输入、受控执行、独立上下文、可靠状态与可验证交付。

关注点	应学习的对象	最常见误读
提示词	装配与触发机制	复制长模板即可变强
多智能体	合同与生命周期	角色越多效果越好
协作	工作区与消息语义	worktree 就完全隔离
架构	持久状态与验证	命令成功就任务完成

1. 两项核心研究结论

1.1 提示词不是一段固定文字：规则发现、模型差异、工具可见性、按需技能和瞬时反馈共同构成装配系统。阅读源码时应沿调用路径追踪它真正送给模型的内容。

1.2 多智能体首先是状态与边界问题：上下文隔离、任务生命周期、消息交付、取消、文件所有权和合并验证，决定了并行是否可靠。

2. 如何把报告用于 Peri

2.1 先读 H 章的运行边界和实验路线，再到各产品章节寻找源码入口与反例。把“机制已存在”“稳定版本默认启用”“当前任务适用”分别确认。

2.2 每个借鉴项应形成小型实验：描述收益假设、最强反例、可回滚改动和验收指标。报告中的设计建议没有替代 Peri 源码审计，也没有宣称某个架构天然胜出。

阅读路径

快速决策：H 章；源码：A-D、F；商业机制：E、G。关键反例：目录不证明权限、工作树不隔离全部元数据、检查点不等于逐语句恢复、多数赞同不等于独立证据。各项仍需条件化实验。

来源 [H01] [H02] [H03] [H06] [H09]

覆盖范围：分层研究，避免伪造穷尽

导读 / 产品与样本选择

> 本报告覆盖可找到且有足够一手证据的代表性产品与框架；“热门”按生态代表性选样，不宣称穷尽所有项目或给出流量排名。

章节	对象类别	代表对象
A-B	CLI 与内核	Claude Code / Codex / Gemini / OpenCode / pi / Goose / Crush
C-D	IDE 与工程 Agent	Cline / Roo / Kilo / Continue / Tabby / Aider / OpenHands / SWE
E-F	商业与多智能体	Cursor / Devin / Copilot / Kiro / Qwen / Trae / Open SWE / 研究框架
G	扩展生态	Zed / Replit / Antigravity / Warp / Refact / bolt.diy / Kimi / Augment / Droid / Amp / Junie
I	补覆盖与后继项目	smolagents / CAMEL / Tabnine / Qoder / CodeBuddy / iFlow / Zoo / MAF

1. 主要研究对象

1.1 A-D 深入官方 CLI、轻量内核、IDE 插件与软件工程 Agent；E 分析商业产品公开合同；F 补充区域开源项目与多智能体研究框架；G 扩展环境、检索和更多产品。 I 章补充框架、社区延续与产品入口迁移。

1.2 模型、harness、IDE、云平台与研究框架属于不同层级。比较时先注明对象：同一个模型可运行在不同 harness，同一个 IDE 也可连接不同外部代理。

2. 覆盖的边界

2.1 未逐一实测所有安装版本、付费功能和企业部署；没有对所有市场产品进行统计抽样。某产品缺少完整公开实现时，报告只解释可验证合同与推断边界。

2.2 I 章新增 Tabnine、Qoder、CodeBuddy、iFlow、Zoo、smolagents、CAMEL 与 Microsoft Agent Framework 的窄专题；深度不等同既有多页研究，也不代表完整源码审计或所有市场产品已覆盖。

2.3 活跃分支、稳定发行、项目停维声明和平台归档分别核实。Crush 当前为带用途限制的 FSL，不与 MIT 等同；源码可读、代码许可和独立部署能力是三个维度。[B59][F51] iFlow 按停维历史样本解读，仍保留自定义 API 继续使用的例外；Tabnine 新旧入口分开核实。[I20][I13]

样本原则

优先选择能提供公开源码、官方接口或原始工程说明的对象；不使用传闻系统提示词填补证据缺口。

来源 [B59] [F51] [I20] [I13]

证据规则：实现、说明与建议分开

导读 / 可复核性与时效

> 同一功能在开发分支、稳定版和文档中可能不同。每条结论需要定位到来源与时点，而非仅附一个项目首页。

证据层	可支持	不可直接支持
源码	具体路径/状态/默认值	所有版本均如此
官方文档	公开能力与限制	内部算法细节
工程文章	设计动机与实验	跨产品公平排名
本报告建议	实施与验证思路	已实现或收益保证

1. 四类证据标签

1.1 源码核实：已读取相应实现，优先固定提交；官方说明：厂商文档或工程文章可验证，但未见内部实现；研究综合：跨来源提出解释；设计建议：本报告原创且待验证。

1.2 产品章节可能同时包含事实与建议，页签表示主要依据，条目中的来源编号和"建议/推演"进一步划定边界。闭源产品不使用泄漏提示词，示例模板均明确标为原创。

2. 时间与验证限制

2.1 首版基准为北京时间 2026-10-18；2026-10-19 续查生命周期、权限、协作、长会话恢复、反例与覆盖缺口。固定分支代码不等于稳定发布保证，当前网页合同与闭源引擎实现分别标记。

2.2 本报告未对全部项目开展统一运行 benchmark；源码推导的风险不冒充实际事故。官方性能结果只可理解为其公布配置下的结果，不能外推为 Peri 的确定收益。

事实优先级

保留版本差异和证据缺口。发现官方页面与源码冲突时，写出冲突并缩窄结论，避免强行拼成一个故事。

如何验收这份报告

导读 / 内容、可视化与研究边界

> 验收看覆盖、机制、证据和能否转成实验；页数只是基本格式要求。图表服务于关系与取舍，不代替实现依据。

维度	检查项	失败时处理
内容	机制、代价、场景齐全	补证或降级结论
证据	编号、URL、日期可查	移除无法支撑的断言
图表	箭头和边界语义清楚	重画而非堆装饰
文件	页数/字体/链接/无裁切	重新生成并检查

1. 内容检查

1.1 产品章节应能回答：解决什么问题、实现机制是什么、关键接口在哪里、哪些状态被隔离、如何失败、适合什么任务。只列特性而没有这些关系，不算完成机制研究。

1.2 横向章节应给出可验证的借鉴路径，而不是笼统“增加多智能体”。每个建议应有输入输出合同、可观察信号、失败语义或成对实验。

2. 阅读与维护

2.1 目录可跳转，各页包含页码和来源编号，附录给出复核入口。架构图是根据公开资料绘制的语义示意，不是厂商内部架构原图；没有通过图形暗示未知实现。

2.2 夜间续查围绕版本漂移、权限、协作、恢复和引用缺口展开，并保留变更记录。首版存在范围限制；后续新增事实必须继续遵守同一证据规则，不能仅靠更多页面提升完整感。

备注

详细研究和生成材料另附可继续修订的数据包；报告本身是可独立阅读的最终文档，不要求读者使用生成脚本。

Claude Code：运行时骨架

Claude Code / 模型、提示词与执行宿主

> Claude Code 的可借鉴处是把模型推理、工具执行和上下文装配组合成可持续闭环；公开文档能确认接口契约，不能据此还原闭源核心。



1. 闭环由运行时承接

1.1 官方把工作分为获取上下文、采取行动、验证结果；工具结果重新进入模型，用户可在中途转向。这个 harness 负责模型看见什么、能调用什么，而模型负责选择下一步。[A01]

1.2 Agent SDK 在 Python/TypeScript 中运行 Claude Code binary，复用工具、会话、权限和 hooks。它是可编程宿主入口，不能把 SDK 的公开包装层当作完整引擎源码。[A13]

2. 提示词不是单个固定字符串

2.1 SDK 的 minimal default、claude_code preset、preset+append 和自定义字符串语义不同；最小默认不等同 CLI 默认。迁移时必须明确选哪一个，否则角色与安全指导可能一起丢失。[A02]

2.2 CLAUDE.md 作为项目上下文注入，不直接改写 system prompt。分析上应把角色规则、项目知识、能力目录和本轮任务分开记录，才能追踪是哪一层导致错误行为。

2.3 分析判断：提示词负责表达目标与选择原则，宿主负责真实工具定义、错误返回和停止条件。若两者对能力或权限的描述不同，模型再强也会反复撞上执行边界。

Peri 启发

建议把模型调用的装配清单作为可观测数据：每块提示词记录来源、版本、加载原因与生效范围。排查行为冲突时先检查装配结果，再调整文字。此为设计建议，未对当前 Peri 作实现判断。

来源 [A01] [A02] [A13]

按需提示词与技能

Claude Code / 常驻知识、程序性知识与触发

> 将经常有效的项目事实常驻，把低频流程放入按需技能，再用明确触发条件选择；节省上下文的关键是少装无关内容，同时保留完整执行契约。

载体	适合内容	需要的治理
CLAUDE.md	常驻项目事实	来源与冲突审计
Skill	按需多步流程	触发、版本、输出格式
Subagent	独立探索任务	上下文和工具范围
Hook	固定时机的检查	失败策略与超时

1. 何时加载，决定长期成本

1.1 CLAUDE.md 适合构建命令、约束和项目结构；目录下规则可随涉及路径加载。auto memory 保存修正与经验，属于上下文而非强制访问控制，不能用记忆替代权限规则。[A03]

1.2 技能先暴露目录信息，调用后正文进入会话并可跨轮保留；压缩会按重附预算裁剪，不能假定正文永久完整。disable-model-invocation 限人工触发，context: fork 则改变执行上下文。[A04]

2. 描述是路由接口，正文是执行协议

2.1 分析建议：description 写适用对象与可识别情境，正文写输入、步骤、产物和停止条件。把大量背景塞进描述，会让每一轮为低概率能力付费；只写“资深专家”又难以匹配任务。

2.2 本报告示例（非官方提示词）：审查任务先定位变更边界；每个发现附路径、触发条件、影响及可证伪办法；无法验证者标为假设。它比角色形容词更能约束可验收结果。

2.3 allowed-tools 是调用轮的临时免批，不是能力白名单；disallowed-tools 才缩减工具池，二者下一条用户消息时清除。正文仍在不代表免批仍在，deny/ask 规则另行生效。[A04]

信任边界

官方说明项目技能免批字段不受 workspace trust 阻挡；v2.1.282+ 的托管 allowManagedPermissionRulesOnly 可忽略相关来源的授权。应同时审技能正文、元数据与有效组织策略。[A04]

来源 [A03] [A04]

子代理：隔离探索上下文

Claude Code / 分派输入与运行边界

> 子代理首先是一种上下文和职责划分方式；把冗长探索移出主对话有效，但委派消息不足、摘要失真和共享文件冲突会把收益抵消。

结构示意 · 关系与数据流

节点与职责

1. 主代理
2. 任务说明 · 目标、证据、交付
3. 子代理上下文
4. 限定工具集
5. 结论与证据
6. 共享工作区

有向关系 · 后半句为条件或动作

- 主代理 → 任务说明 · 委派
- 任务说明 → 子代理上下文 · 初始化
- 子代理上下文 → 限定工具集 · 调用
- 限定工具集 → 共享工作区 · 读写
- 子代理上下文 → 结论与证据 · 汇总
- 结论与证据 → 主代理 · 返回

1. 一份定义同时控制路由与能力

1.1 自定义代理由 Markdown 正文和 YAML frontmatter 组成：description 帮助主代理决定何时委派，正文成为该代理自己的系统提示词；工具名单、模型与执行限额另行配置。[A05]

1.2 普通子代理从新上下文开始，由委派消息交代任务；fork 继承已有对话。新鲜上下文更轻，继承上下文更省重新解释，但也继承大量无关历史，选择应依据任务所需证据。

2. 限制必须由运行时实现

2.1 当前文档允许子代理继续派发；自 v2.1.219 起默认深度为主会话以下三层。达到上限时撤去 Agent 能力，fork 保留工具但调用报错；不能沿用早期“永不嵌套”的结论。[A05]

2.2 官方 sandbox 文档说明子代理与父会话同进程并共享 sandbox 配置。独立上下文不产生独立文件系统；worktree 仅分开工作目录和分支，不自动隔离进程、网络和宿主资源。[A10]

2.3 SendMessage 可续跑完成或模型 TaskStop 停止且已退出的普通子代理；用户 x/SDK stop_task 取消后，须用户恢复才允许代理消息续跑。同一 agent ID 开新 run；交互会话中兄弟代理恢复时，结果回给恢复者。[A05]

Peri 启发

建议派发明确输入证据、改动路径和验证标准；另记 cancel_actor、run/attempt 与 reply_to，防止普通代理消息绕过用户停止意图。上下文隔离仍不等于文件隔离。

来源 [A05] [A10]

团队协作：邮箱与任务依赖

Claude Code / Teams 的价值与真实门禁

> Teams 把独立会话、共享任务表和邮箱组合起来，适合交叉验证与分块开发；计划批准、消息送达和任务完成是不同状态，不能混为验收通过。

协作元素	提供什么	没有自动提供什么
任务依赖与锁	可领取顺序与互斥领取	文件写入事务
邮箱写成功	消息已写入接收邮箱	已理解、已执行
计划批准	允许退出规划阶段	独立架构审查
完成 hook	拦截或反馈入口	业务正确性证明
关闭请求	协作接受或拒绝	工具已停止、完整恢复

1. 协作协议比角色数量重要

1.1 Agent teams 仍为实验且默认关闭；leader 派发 teammate，任务可依赖其他任务，领取时使用文件锁。邮箱落在本地 JSON 文件，写成功才报告发送成功，磁盘失败应回传错误。[A06]

1.2 任务锁只防重复领取，不防两人改同一文件。分析建议先分配文件所有权，跨边界改动用消息提交请求；最终由集成人处理依赖和冲突，避免把共享任务表误当事务系统。

2. 计划与完成都需要证据

2.1 当前官方说明：teammate 的计划请求抵达 lead 后自动批准，不经过 lead 审查；执行权限仍独立检查。若要真正质量门禁，应使用 TaskCompleted、TeammateIdle 等 hooks 或外部验证器。[A06][A07]

2.2 teammate 要求交互会话；-p/Agent SDK 的命名委派仍是普通子代理。shutdown request 可接受或拒绝，可能等待当前工具；/resume、/rewind 不恢复 in-process teammates，任务文件留存不等于执行恢复。[A06]

Peri 启发

建议 AOS 的协作面用明确的 task_id、attempt、依赖与验收状态；代理消息只传证据和请求，不可被解释为用户授权。验收关卡应消费测试与审查产物。

来源 [A06] [A07]

长任务：压缩、恢复与事件钩子

Claude Code / 保存记录与继续执行

> 会话记录、模型活动上下文和仍在运行的工具是三种状态；恢复 transcript 不会自动恢复被中断的副作用，压缩成功也不等于完成任务。



1. 历史连续不代表进程连续

1.1 默认保存本地 transcript；从记录恢复默认不重跑未完成工具。原 agent 定义找不到时，会告警并回到默认工具；--mcp-config、--settings、--add-dir 等需按文档重新提供，权限模式也依恢复入口而异。[A12]

1.2 先清理旧工具输出，再摘要。当前文档说明压缩后根规则、auto memory 与 plan 从盘注入；最多回读五个文件，超 5000 token 的文件只给路径引用。规则复载不等于全文或摘要确定性保真。[A01][A59]

2. Hook 是生命周期扩展，不是自由调度器

2.1 PreCompact 可阻止压缩，但若 API 已报上下文超限，阻止会使当前请求失败；PostCompact 可观察摘要。不能把“阻止压缩”当作安全保留全部信息的无成本选择。[A08]

2.2 Stop 在响应结束时触发，不等于业务完成；PostToolUse 也不能撤销已发生副作用。多个 PreToolUse 同时改写参数还存在完成顺序竞争，应让参数变换只有一个明确责任者。[A07]

2.3 分析建议：恢复后先展示未完成动作与当前工作区差异，再选择继续点；把历史事实当作实时状态会导致重复执行、遗漏外部变化或错误宣布完成。 当前 -p 的 defer 会以 tool_deferred 退出并保存待调用，resume 后重跑同一 Hook；这不是任务已成功，宿主仍要保存会话与审批身份。[A08]

Peri 启发

建议恢复前比对代理定义、最终工具集、MCP、目录、策略和工作区版本；会话 ID 不是完整启动快照。缺少受限定义时应显式确认降级，再测试上下文、文件和在途操作。

来源 [A01] [A07] [A08] [A12] [A59]

权限、沙箱与回滚的三条边界

Claude Code / 安全能力不能互相替代

> 权限决定一次调用能否执行，沙箱限制执行中的访问，检查点用于有限回退；三者叠加才形成可控环境，任何一层都不是完整事务保障。

机制	约束对象	典型盲点
提示词	模型的行为选择	不能强制拒绝系统调用
权限规则	即将调用的工具	不等同进程运行时边界
Bash sandbox	命令及子进程访问	不覆盖所有扩展进程
Checkpoint	受跟踪的文件编辑	Bash、远程副作用
外部事务/幂等	业务状态变化	需应用自行实现

1. 声明规则与操作系统隔离

1.1 官方权限文档明确由 Claude Code 执行规则，而非依赖模型服从；禁用整个工具可从可见工具集中移除它，按参数匹配则保留工具并在调用时拦截。[A09]

1.2 Bash sandbox 约束命令及其子进程，网络通过代理检查目标，文件规则控制读写。它不是给所有 hooks、MCP 和扩展代码套上统一容器；项目部署应列清实际覆盖面。[A10]

2. 可撤回的范围必须公开

2.1 Checkpoint 主要覆盖文件编辑工具触及的内容，Bash 命令产生的文件修改不能靠 rewind 撤销，子代理修改也有恢复限制。远程 API、数据库和部署副作用更不属于本地文件快照。[A11]

2.2 普通 PreToolUse 的 updatedInput 替换整个输入对象，再按新参数验权；其 allow 不能覆盖权限 deny/ask。SDK 宿主仍需定义无人审批时的失败与恢复，不能以聊天转述授权。[A08][A13]

验证建议

给测试环境设置独立用例：被禁工具是否不可见、参数重写后是否重验、子代理能否越界、Bash 修改能否撤回。分别判定，避免用一个成功演示替代边界验证。

来源 [A09] [A10] [A11] [A13] [A08]

Codex：核心、会话与执行边界

Codex CLI / Rust 核心与客户端协议

> Codex 把会话控制、持久记录和进程执行拆成显式接口；客户端可读取历史而不启动代理，这对超长对话、多个界面和资源释放尤其重要。

职责与接入示意 · 非必经独立进程链

节点与职责

1. 客户端示意 · CLI / IDE等
2. App Server · 可嵌入或远端接入
3. Rust 核心
4. ThreadStore · 历史与元数据
5. Exec Server · 按执行环境选用
6. 执行环境

有向关系 · 后半句为条件或动作

- 客户端示意 → App Server · 请求与事件
- App Server → Rust 核心 · 会话操作
- Rust 核心 → ThreadStore · 显式持久化
- Rust 核心 → Exec Server · 可选执行后端
- Exec Server → 执行环境 · 受控执行

1. UI 不应拥有唯一的业务状态

1.1 codex-core 承担代理逻辑；app-server 以 thread、turn、item 表达对话、请求和输入输出。初始化后开始或恢复会话，用 turn/steer 追加输入并订阅增量事件。[A14][A16] 固定 TUI 支持嵌入、守护或远端接入，不等于一律独立进程。[A62]

1.2 exec-server 单独处理文件与进程 RPC，提供会话和进程控制。threadId/toolCallId 仅用于关联追踪，源码文档明确其不是授权凭证；远端执行应保留独立鉴权边界。[A31]

2. 读取、加载与运行分别建模

2.1 当前官方 API：thread/read 不加载运行时；resume 可 excludeTurns 后分页。turns/list 默认最新优先与摘要项，反向 cursor 重含锚点；items/list 默认最旧优先且 item 锚点排除自身，底层须支持分页，不能通用一套去重假设。[A14]

2.2 ThreadStore 分开 append_items 与 update_thread_metadata；LiveThread 汇总活跃写入，本地实现用 JSONL 保存规范历史、SQLite 保存可查询元数据。核心依赖存储接口，避免每层自行维护影子状态。[A17]

2.3 当前文档的 unsubscribe 仅移除本连接订阅；最后订阅消失后，还须无活动 30 分钟才卸载并发 closed。关闭视图、取消运行和立即释放内存不是同一事件；未测该宽限期的真实 RSS。[A14]

Peri 启发

建议将 ACP 的运行恢复和历史展示分离：加载核心活动窗口后，前端用稳定游标补旧 turn/item；关闭视图不应等同杀死任务，历史分页也不应重新持有完整 transcript。

来源 [A14] [A16] [A17] [A31] [A62]

提示词装配：来源、更新与能力

Codex CLI / 从模板文本到状态转换

> 值得学习的不只是公开提示词措辞，而是指令如何发现、限量、附带来源并在改变时撤销旧版本；这使长任务中的规则变化可以被检查。

装配对象	工程机制	防止的问题
项目指令	根边界、预算、来源	跨项目污染与无限加载
指令更新	状态 diff 与显式撤销	新旧规则长期并存
模型模板	按模型/版本选择	误把旧措辞当全局契约
技能目录	元数据与正文分离	能力目录吞噬窗口

1. 发现规则有明确路径与预算

1.1 agents_md.rs 按项目根标记定位范围，从根至工作目录收集指令，支持 AGENTS.override.md 与配置回退名；读取经过执行环境的文件接口，并受总字节预算约束，不无限向父目录搜寻。[A18]

1.2 world_state/agents_md.rs 用 render_diff 比较已知状态：不变不重复发送，替换时说明旧指令被替代，移除时显式撤销。仅删掉宿主文件却不告诉模型，历史中的旧命令仍可能持续影响行为。[A19]

2. 可复用提示词要与模型和工具版本绑定

2.1 仓库公开的 gpt-5.2-codex_prompt.md 包含快速检索、保护脏工作区、审查输出和编辑约束；它只是一个模型专用模板，不能当作今天所有 Codex 模型的统一系统提示词。[A21]

2.2 skills/catalog_prompt.rs 生成带来源定位符或别名的能力目录，再指导读取被选中的完整技能。分析建议把能力可用性作为装配输入，避免提示模型调用本轮未暴露的工具。[A20]

Peri 启发

建议给 prompt block 设置稳定 ID、版本与 replace/remove 语义，并在 trace 中保存装配摘要。工具冲突排查可比较“实际暴露能力”与“指令要求能力”，形成可复现诊断。

来源 [A18] [A19] [A20] [A21]

多代理：派生、邮箱与唤醒

Codex CLI / 子线程并发的生命周期

> 多代理的难点是保证派发、消息接收、结束和再次唤醒相互一致；一条消息被排队不等于接收方一定继续执行，必须有独立状态机。

结构示意 · 关系与数据流

节点与职责

1. 发送消息
2. 待收消息队列 · 卸载等情形
3. 运行中接收方
4. 空闲接收方
5. 唤醒条件
6. 新执行轮次

有向关系 · 后半句为条件或动作

- 发送消息 → 运行中接收方 · 活跃直接投递
- 发送消息 → 待收消息队列 · 需要暂存
- 待收消息队列 → 运行中接收方 · 加载后取出
- 待收消息队列 → 空闲接收方 · 队列通知
- 空闲接收方 → 唤醒条件 · 检查
- 唤醒条件 → 新执行轮次 · 符合规则
- 新执行轮次 → 待收消息队列 · 消费

1. 上下文派生不等同资源复制

1.1 v2 spawn handler 解析任务名、任务消息与 fork_turns，区分 all 和 none，再进入 ThreadSpawnRequest；fork 路径会刷出父历史，并重写角色边界，历史工具记录不代表子代理继承活的进程句柄。[A22][A23]

1.2 官方文档说明子代理继承父会话权限；角色写“只读”不产生独立只读 sandbox。当前本地版本还以直接请求或项目/技能指令作为派发依据，工具存在不意味着一定主动派发。[A15]

2. 空闲状态必须有投递契约

2.1 agent/control/mailbox.rs 保留卸载代理的待收消息，watch 通知队列变化，取走消息后更新状态。它解决内存驻留与收件的衔接，但这段代码本身不能证明跨进程崩溃后恰好一次投递。[A24]

2.2 2026-10-18 提交新增默认关闭的 wake_on_agent_messages，让普通迟到消息也能唤醒已完成 root；新增路径排除中断态、Plan 与 child。显式 trigger_turn 及既有 durable-sleep 另有唤醒路径，不能把此限制推成全部消息的统一规则。[A25]

2.3 分析建议：在“刚判断空闲”和“刚有新消息”之间制造竞争，让同一输入经过直接投递与邮箱两条路径；既要检查不漏唤醒，也要检查不会启动两次。

Peri 启发

建议把“消息已接收”“任务已激活”“消息已消费”分开观测；用迟到完成消息、停止竞态、重复通知、计划模式和卸载恢复构造回归场景。不要只加一条提示词要求代理等待。

来源 [A15] [A22] [A23] [A24] [A25]

压缩：候选结果需要安全安装

Codex CLI / 前台继续与后台摘要

> 异步压缩的核心不是并发调用摘要模型，而是验证候选仍适用于当前历史，保留实时后缀、上下文基线和计费后，再安装到活动窗口。

结构示意 · 关系与数据流

节点与职责

1. 历史版本 N
2. 后台压缩
3. 前台新增后缀
4. 候选校验 · 版本、模型、授权
5. 安装摘要与后缀
6. 丢弃或重试

有向关系 · 后半句为条件或动作

- 历史版本 N → 后台压缩 · 快照
- 历史版本 N → 前台新增后缀 · 继续执行
- 后台压缩 → 候选校验 · 候选
- 前台新增后缀 → 候选校验 · 当前状态
- 候选校验 → 安装摘要与后缀 · 仍兼容
- 候选校验 → 丢弃或重试 · 已失效

1. 摘要请求绑定输入身份

1.1 async_compaction.rs 用 history_version、model_slug、compaction_hash 和 auth generation 标识尝试；结果通过 oneshot 返回，在检查点尝试采用。功能依赖开关、远端 V2 能力等条件，不是无条件路径。[A26]

1.2 启动时在锁内捕获输入和注册任务，避免 shutdown 或历史替换抢先；会话关闭会取消并等待后台任务，已完成用量仍计入。放弃候选结果并不代表请求从未消耗资源。

2. 安装摘要是状态变更

2.1 compact_remote_v2_installation.rs 先校验 prepared prefix，再合并尚未被摘要的实时后缀；world-state baseline 与上下文一起安装，并重新计算 token。后缀过大仍可能要求继续回退处理。[A27]

2.2 恢复路径在 rollout_reconstruction.rs 重建压缩点及上下文基线；持久历史和模型活动窗口分属不同职责。分析上必须额外测量旧 Arc、工具输出缓存、观测队列和 UI 引用，才能判断内存是否释放。[A28]

2.3 另一个层次是当前 Responses API：自动压缩的输入数组可裁去最新 compaction item 之前的项；独立 compact 返回的规范窗口不可再裁剪，也不只是加密项；previous_response_id 链不手裁。不能把这些规则当作固定 CLI 分支的同版证明。[A58]

Peri 启发

建议先定义压缩替换的不变量：消息不丢不重、工具调用配对完整、指令来源可恢复、授权不扩张。验收分别看活动 token、堆中可达字节、RSS 与恢复后首轮正确性。

来源 [A26] [A27] [A28] [A58]

执行安全：审批与隔离正交

Codex CLI / 工具契约和操作系统策略

> 同意运行某条命令与允许它访问哪些资源是两项决策；将审批、沙箱选择、执行重试和规则测试分开，有助于避免误放行和无限重试。

检查层	回答的问题	需要保留的证据
权限/审批	本次动作是否允许	批准主体、参数、范围
规则匹配	是否命中可复用规则	规范化命令与命中项
沙箱策略	进程实际能访问什么	解析后的资源边界
执行结果	动作是否真正完成	退出状态、输出、标识
重试判断	能否再次执行	失败位置与幂等性

1. 工具执行有类型化接口

1.1 tools/sandboxing.rs 定义审批需求、沙箱行为与执行尝试相关接口，使各工具接入统一控制流程。审批缓存或已允许前缀只处理决策复用，不能解释为任意后续命令的总授权。[A29]

1.2 execpolicy 使用 prefix_rule 表达 allow/prompt/forbidden，并支持 match/not_match 示例做规则校验。规则覆盖命令前缀，复杂 shell 的解析与实际执行仍需专门处理，不能靠字符串“看起来安全”。[A30]

2. 无法表达的隔离要失败关闭

2.1 core README 说明不同平台用 Seatbelt、Linux 后端或 Windows sandbox 落实策略；复杂读写例外不能由弱后端无损表示时，应失败而不是悄悄放宽。平台适配是语义适配，不只是换启动命令。[A16]

2.2 exec-server 远端转发不拥有业务重放权；重连依赖目标保留的 session 与输出。分析建议让网络重试、审批重试、命令重试分别拥有幂等标识，避免连接恢复意外重做副作用。[A31]

Peri 启发

建议由工具宿主持有最终执行边界，核心接收结构化 allow/deny/error；审批对象应绑定最终参数。被拒绝与执行失败分开返回，让模型知道应改计划还是核查已有副作用。

来源 [A16] [A29] [A30] [A31]

Gemini CLI：能力驱动提示词

Gemini CLI / 可检查的装配器

> Gemini CLI 将提示词拆成可条件化片段，由模型、工具和运行模式共同决定装配结果；这让能力变化能同步反映到指导文字，而非长期堆叠例外。

结构示意 · 关系与数据流

节点与职责

1. 运行配置
2. 工具与代理目录
3. 技能元数据
4. PromptProvider
5. 条件化片段
6. 本轮模型输入

有向关系 · 后半句为条件或动作

- 运行配置 → PromptProvider · 模式与模型
- 工具与代理目录 → PromptProvider · 能力
- 技能元数据 → PromptProvider · 可发现知识
- PromptProvider → 条件化片段 · 选择
- 条件化片段 → 本轮模型输入 · 装配

1. 结构边界足够可读

1.1 packages/cli 提供终端交互，packages/core 管模型通信、工具与状态；PromptProvider 读取实际工具注册表、技能目录、代理目录和审批模式，再构造 SystemPromptOptions。[A32][A33]

1.2 同一装配器在现代与 legacy snippets 间选择，Plan 模式和实施工作流互斥装配，工具不存在时不加入对应工作流提示。GEMINI_SYSTEM_MD 可覆盖模板，另有导出当前提示词的入口。[A33]

2. 公开提示词体现的操作策略

2.1 snippets.ts 把研究、策略、执行和验证写成工作流程，并要求独立任务才并行派发；同时提示小而直接的工作可由主代理完成。其价值在明确委派条件，避免“凡事并发”或“永不委派”。[A34]

2.2 Skills 的启动目录只提供名称和描述；activate_skill 后再加载正文和资源目录，文档还定义了用户同意与允许路径扩展。由此可见“加载知识”和“扩大资源可读范围”需要联动审计。[A45]

2.3 分析建议：为每个片段记录启用条件，并在能力变化时生成差异；这样能说明一次异常行为是来自模型选择、工具可见性还是规则装配。

Peri 启发

建议把工具暴露和提示词装配放在同一能力快照上；做删工具、换模型、切 Plan 的快照测试，检查提示词是否仍要求不存在的调用。避免只靠人工翻长 prompt 找冲突。

来源 [A32] [A33] [A34] [A45]

本地与远程代理的共同接口

Gemini CLI / 委派工具、局部注册表与 A2A

> Gemini CLI 用统一调用入口选择本地或远程代理，但上下文隔离、工具状态隔离和远端任务协议属于不同层；统一外观不代表相同失败语义。

层次	本地代理	远程代理
入口	统一工具与本地定义	同入口与远程定义
执行	本地独立循环	A2A 远端服务
能力	局部注册表与策略	远端声明及认证
身份	本地流与代理编号	上下文及任务编号
取消	本地中止信号	所读链无远端取消确认
终态	按本地终止原因映射	流结束不等于业务成功

1. 本地代理是受限工具运行时

1.1 AgentTool 接收代理名和完整任务，查询定义后分发；本地执行器建立独立工具、提示词和资源注册表，克隆工具，并派生带代理身份的消息总线。[A35][A36]

1.2 构造能力集时过滤代理类工具，阻止递归；用完成工具、时间及轮次限额结束。局部注册表减少工具实例串扰，但共享部分配置与执行环境，不能当作容器隔离。[A36]

2. 远端协议保留任务身份

2.1 远程定义通过 A2A 能力卡与认证接入；远程适配保留 contextId/taskId 并转换统一事件，本地适配包装执行器回调。源码分别位于 local/remote-subagent-protocol.ts。[A37][A38][A39]

2.2 固定快照 remote abort 只传本地 signal，所读链未显式调用另有的 cancelTask；不能据此断定远端必继续。正常 EOF 可映射 COMPLETED，未保留全部 Task 终态；合法失败流正常结束时存在状态失真可能，未实测。[A38][A53][A54][A55][A56]

Peri 启发

建议将 transport_end、remote_task_state 与业务验收分列；远程取消请求、确认及产物版本各有身份。AgentSession 的事件回放也不能替代远端状态对账。[A47]

来源 [A35] [A36] [A37] [A38] [A39] [A47] [A48] [A53] [A54] [A55] [A56]

工具调度：先校验，再并发

Gemini CLI / Scheduler 与 PolicyEngine

> 并发工具需要明确的状态转换和可并行判定；审批与 hook 必须作用于最终参数，而取消要覆盖排队和执行中的两类工作，不能只停止显示。

结构示意 · 关系与数据流

节点与职责

1. 工具请求
2. 构建 invocation
3. BeforeTool hook
4. 策略与决定合并 · 存在 DENY→ASK 条件路径
5. 待执行队列
6. 执行
7. 结构化终态

有向关系 · 后半句为条件或动作

- 工具请求 → 构建 invocation · 验证参数
- 构建 invocation → BeforeTool hook · 检查
- BeforeTool hook → 构建 invocation · 参数变化
- BeforeTool hook → 策略与决定合并 · 最终参数
- 策略与决定合并 → 待执行队列 · 允许
- 策略与决定合并 → 结构化终态 · 拒绝
- 待执行队列 → 执行 · 合适批次
- 执行 → 结构化终态 · 完成或取消

1. Scheduler 组织一次调用的全过程

1.1 scheduler.ts 将调用推进为校验、待审批、待执行、执行中和终态；连续可并行请求组成批次，待活跃调用均就绪后执行。并行集合由专门函数判断，不是模型同批输出就全部并发。[A40]

1.2 BeforeTool 可返回改写参数；实现更新请求并重建调用对象，再进入策略与确认。取消可级联到队列，尾调用重新校验。分析建议将排队耗时、审批耗时和执行耗时分开，以免把等待都归因于模型。

2. 策略层决定动作，不要求模型自觉

2.1 PolicyEngine 局部返回 allow/deny/ask，并在非交互模式把 ask 转 deny；但这不等于后续组合链永不放宽。固定 scheduler 随后可将 Hook ask 覆写为 ASK_USER。[A40][A41][A42]

2.2 所读 main 文档仍警告 workspace policy tier 未生效；不能因项目里有策略文件就推断它正在保护环境。分析建议启动时输出有效规则、来源和被忽略项，并用真实拒绝案例验证配置。

2.3 需有已配置且命中的 ask Hook、工具仍可达；原 policy DENY 可进入强制确认，用户同意后置为待执行。项目 Hook 受信任检查；无相应 Hook 不走此路径。仅静态追踪，未端到端复现。[A49][A50][A51][A52]

Peri 验收建议

将“拒绝不能被后续阶段放宽”写成组合测试，而非只测 PolicyEngine。还要绑定最终参数；非交互 Hook ask 的宿主表现本轮未运行，不把局部 ASK→DENY 当作整链已证保证。

来源 [A40] [A41] [A42] [A49] [A50] [A51] [A52]

超长会话：减载、摘要与回放

Gemini CLI / 内存压力与历史可恢复性

> Gemini 的实现把旧输出折叠、超预算内容外置、模型摘要与持久日志分开处理；这比只提高 context window 更接近长期可控的资源模型。

处理	减少什么	仍需验证什么
旧输出折叠	长期重复 payload	重要证据是否可找回
大输出外置	后续模型输入体积	首次峰值、文件寿命与授权
摘要 + probe	模型活动窗口	遗漏率与新增成本
JSONL 追加	减少整份重写	逻辑删除、多记录提交、停记告警
回放与有限缓存	常驻部分消息	全量重放峰值、队列与背压

1. 压缩前先控制输出体积

1.1 压缩服务会折叠旧工具输出，保护近期轮次；超预算输出可存入临时文件并保留短引用。摘要生成 state_snapshot 后，第二次检查是否遗漏技术细节与约束。[A43][A34]

1.2 空摘要或 token 反增会拒绝安装；主客户端仅在自动压缩 token 反增后置失败标志，后续自动调用才可只截断。强制调用、新 chat 另算；probe 是模型自检，不保证无遗漏。[A43][A60]

2. 记录与事件消费分别优化

2.1 JSONL 的 patch/rewind 形成逻辑恢复视图；旧字节尚在不等于普通恢复可取全原文。有限缓存仍可触发同步全量重放，成功压缩重启前也会调用；ENOSPC 告警停记而会话继续。未证多记录安装原子性。[A44][A60][A61]

2.2 AgentSession 先订阅实时事件再建立回放，按 streamId/eventId 连接历史与新增流，减少回放窗口中的遗漏风险。ACP 模式则用 stdio JSON-RPC 面向客户端，不能从协议支持推断历史恢复已无限可扩展。[A47][A46]

Peri 启发

建议压测同时覆盖高速流、大量稳定历史、压缩后继续读旧证据三类负载；分别报告模型 token、运行堆、事件队列和前端恢复峰值。摘要质量与内存释放不能合并成一个指标。

来源 [A34] [A43] [A44] [A46] [A47] [A60] [A61]

三种路线，可迁移的工程原则

Claude Code × Codex × Gemini CLI / 比较与取舍

> 三者都在缩小模型需要记住的内容，并把关键动作交给运行时约束；对 Peri 更有价值的是可组合边界、可解释状态与验证方法，而非照搬长提示词。

维度	Claude Code	Codex CLI	Gemini CLI
提示词扩展	常驻指令 + 按需技能	来源发现 + 状态替换	能力条件化 snippets
多代理形态	子代理 / 实验团队	子线程与显式控制工具	本地受限 / A2A 远端
长历史路径	摘要 + transcript 恢复	活动窗口 + 分页存储	折叠 + 外置 + 摘要
强制边界	权限 / hooks / shell sandbox	审批 / execpolicy / OS 隔离	PolicyEngine / scheduler
应避免的误读	计划批准即质量通过	main 特性均已发布	工具 registry 即进程隔离

1. 比较要选同一层次

1.1 Claude 公开了技能、子代理、团队和 hooks 的不同扩展粒度；Codex 源码呈现线程、存储与执行边界；Gemini 的提示词和调度模块便于直接追踪装配。三者各自解决的是不同层次的复杂度。

1.2 源码可见不代表默认开启，文档不等于闭源审计。开源部分固定到本章来源中的提交；Claude 按检索日官方文档，未引用泄露提示词。

2. 面向 Peri/AOS 的分析建议

2.1 先保证状态正确：区分运行、空闲、停止、待审批与结束；定义消息唤醒条件，为持久历史、活动窗口、前端历史和观测输出分别设置预算。

2.2 再优化能力契约：按工具集装配指令，委派携带目标和证据，宿主强制权限。最后才增加代理数量，以独立发现数、验证通过率和总成本衡量收益。

可执行验收

选同一仓库与固定任务集，先单代理基线，再只改变一个机制；记录正确性、重复探索、上下文峰值、恢复时延与权限拒绝案例。不能先假定多代理更快。

来源 [A02] [A04] [A06] [A15] [A17] [A25] [A33] [A36] [A40] [A43]

OpenCode：让会话拥有执行权

OpenCode V2 / 架构与消息入口

> 当前 OpenCode 把界面连接、持久输入与运行循环分开；这有利于续跑，但实时事件流本身并不提供历史重放保证。

结构示意 · 关系与数据流

节点与职责

1. 界面 · 连接可消失
2. HTTP 契约 · 请求与订阅
3. 持久输入 · steer / queue
4. 会话运行 · 安全边界交付
5. 数据库状态 · 输入与消息
6. 实时事件 · 无历史重放

有向关系 · 后半句为条件或动作

- 界面 → HTTP 契约 · 提交
- HTTP 契约 → 持久输入 · 接纳
- 持久输入 → 数据库状态 · 落盘
- 持久输入 → 会话运行 · 推进
- 会话运行 → 数据库状态 · 更新
- 会话运行 → 实时事件 · 发布
- 实时事件 → 界面 · 观察

1. 先固定研究对象

1.1 本节研究 anomalyco/opencode 的 2026-10-18 dev 快照，核心路径已是 packages/core，V1/V2 不可混用。[B01] 另一个 opencode-ai/opencode 已于 2025-09-18 归档并指向 Crush，不是本节对象。[B58]

1.2 网络客户端由同一 HTTP 契约生成，TUI、桌面及自建前端可消费服务能力；把前端视为会话观察者，便能减少连接寿命与任务寿命的耦合。[B14]

2. 输入先成为可解释的状态

2.1 session/inbox.ts 将 user、synthetic、compaction、move 作为持久输入，按 session 串行 admission；已见 ID 会对账，类型或生命周期不一致会报冲突。[B10]

2.2 steer 在步骤边界交付，queue 在空闲边界进入下一轮。研究判断：这种区分可以准确表达“纠正当前任务”和“排下一件事”，避免一个模糊 busy 标记承担全部调度语义。

3. 流连接有明确短板

3.1 官方客户端多个订阅者共用惰性连接；事件源会等待订阅者接受事件。慢消费者若直接做重渲染，可能拖慢消费链，因此文档要求先缓冲。[B14]

3.2 订阅只含实时事件，没有重放或自动重连。断线恢复仍需查询权威会话状态；不能由客户端收到最后一个 token 推断服务端已经稳定空闲。研究判断：恢复查询与实时观察须分别定义一致性保证。重连时应对照会话版本，避免把旧状态覆盖新事件。

Peri 启发

设计建议：ACP 的输入确认、run 终止事件与 cursor 历史恢复使用不同契约；先明确哪一层是权威，再讨论缓存和 UI 优化。

来源 [B01] [B10] [B14] [B58]

OpenCode：提示词也是运行状态

OpenCode V2 / 指令发现与冲突

> 提示词不是启动时拼接一次的大字符串；全局、项目、嵌套规则和运行更新具有不同寿命，必须说明来源与生效边界。

指令层	装载方式	容易误判的边界
基础能力	按 profile 与可用工具生成	文本指导不能执行权限检查
常驻规则	起始装载，变更可形成更新	合并不等于冲突已解决
嵌套规则	探索触发，历史内去重	已加载后不会自动刷新
skill 正文	按需载入	发现条目不等于已读全文

1. 装配顺序具有语义

1.1 V2 的顺序是 agent/provider 基础提示、Code Mode 指导、MCP/引用/skill 指导、项目及配置指令、环境信息、会话专属指令。各层合并，系统不会自动解决文本矛盾。[B02]

1.2 system-prompt.ts 根据实际可用工具添加 shell、write、edit 指导。价值在于工具撤销后不再教模型调用不存在的能力；提示与工具目录来自同一个能力事实。[B03]

2. 常驻与探索规则分开

2.1 当前工作区向上的 AGENTS.md 构成初始来源；向下读取文件或列目录时再发现嵌套规则。嵌套指令在可见历史内去重，压缩或回退移除后可再次加载。[B02]

2.2 常驻指令改动会在下一次模型请求前注入更新；已加载的嵌套文件不会自动监视修改。这意味着同一名称的“项目规则”也可能有不同刷新保证。

3. “只读”必须拆成两种证据

3.1 explore 的提示词禁止写入；其 profile 禁止普通编辑工具，但允许 shell:*。源码事实是工具目录受到限制，不能据此宣称任意 shell 都被系统强制为只读。[B04]

3.2 研究判断：最值得学习的是指令来源与版本化，而非增加更多强硬句子。自动生成约束覆盖表，并对已撤销工具、过期嵌套规则和冲突段落做离线检查，更易解释失败。

Peri 启发

设计建议：给每段 prompt 标记 source、scope、version、loaded_at 与失效原因；工具冲突审查应同时检查 description、schema 和执行门禁。

来源 [B02] [B03] [B04]

OpenCode：委派就是子会话

OpenCode V2 / 多智能体协作

> OpenCode 的委派是有身份、父子关系和恢复路径的子会话；新子会话默认新上下文，父模型必须负责提供完整任务契约。

结构示意 · 关系与数据流

节点与职责

1. 父会话 · 分解与上下文交接
2. 启动检查 · 模式 / 权限 / 深度
3. 子会话 · 持久 sessionID
4. 运行与通知 · 内存 job + KV标记
5. 结果与状态 · 成功 / 失败 / 取消
6. 继续同一孩子 · 保留研究上下文

有向关系 · 后半句为条件或动作

- 父会话 → 启动检查 · 委派
- 启动检查 → 子会话 · 建立关系
- 子会话 → 运行与通知 · 执行
- 运行与通知 → 结果与状态 · 完成
- 结果与状态 → 父会话 · 回传
- 父会话 → 继续同一孩子 · 补充要求
- 继续同一孩子 → 子会话 · sessionID

1. 委派工具如何启动工作

1.1 subagent 工具接受 agent、description、prompt，可选 sessionID 与 background。新建时明确告知上下文是新的；继续已有 session 时验证它确属当前父会话。[B05]

1.2 启动先检查 profile 模式、递归深度与 subagent 权限，再创建 child，提交 prompt，并登记 job。默认深度为一；general profile 还显式禁止继续派发。[B04][B05]

2. 前后台共用可追踪身份

2.1 前台等待 job 完成并回传最终输出；父调用被中断时取消子会话及 job。后台立即返回 sessionID，完成后通知父会话，工具说明要求不要轮询或重复做同一份工作。[B05]

2.2 失败保留 sessionID 可续接。job 注册表在进程内，KV 另存后台通知；结果进入父 synthetic 输入后清标记，不等于父模型已读/验收。observer 的进程内去重也不是跨重启恰好一次保证。[B64][B65][B66]

3. 能力边界要单独设计

3.1 自定义子 agent 使用自身 profile 权限，并非自动取父 profile 的权限子集；session 级配置继承也不能替代 profile 权限交集。[B06]

3.2 子会话复用父 Location，不自动创建独立工作树。并行实现必须提前划分文件所有权；即使手动换 worktree，也不能当成进程、文件系统或凭据隔离。[B57] 设计建议：派发前明确共享目录与结果归并责任。

Peri 启发

建议分别保留运行终态、父输入接纳凭据和产物验收记录，另存产物引用；不能把结果进入会话升级为代码已应用。本轮未审完整启动恢复及跨组件原子性。

来源 [B04] [B05] [B06] [B57] [B64] [B65] [B66]

OpenCode：压缩边界进入查询层

OpenCode V2 / 长历史与恢复

> 摘要必须成为下一次历史读取的边界；只减少发给模型的文本，无法自动减少历史反序列化、驻留内存和客户端重放成本。

对象	作用	独立验收指标
完整历史	审计、回看、再恢复	分页读取量与序列正确性
结构化摘要	保留任务语义	约束、决策、未完成项召回
近期对话	保留精确工具与用户细节	tool call/result 不断对
原生 checkpoint	供应商专属上下文表示	切模型及换端点的回退行为

1. 摘要与最近上下文分工

1.1 V2 官方说明默认保留约 15000 token 的近期对话；摘要描述目标、约束、决策、完成与在途工作、阻塞、下一步和相关文件，后续压缩更新同一摘要。[B07]

1.2 源码要求摘要排除 AGENTS.md 等环境指令，因为下一轮会单独得到当前版本。若摘要再次固化旧规则，指令更新就可能被旧记忆反向覆盖。[B08]

2. 真正的读取窗口

2.1 history.ts 先找最近成功 compaction 的 seq，再查询该下界之后的消息；这在存储读取阶段限制模型上下文，不只是取出全历史后切片。[B09]

2.2 旧消息仍存储。研究判断：历史审计、UI 分页和运行上下文可以用不同视图；但是仍需单独测量缓存、观察器及客户端是否保留旧对象，不能凭该查询保证 RSS 下降。

3. 原生 checkpoint 带来兼容约束

3.1 provider 原生压缩可返回不可读 checkpoint；重放依赖 provider、model、endpoint provenance。兼容性不满足时，读取层不能把 opaque 数据当通用摘要。[B07][B09]

3.2 Ended 的 completed 摘要投影与指令 epoch 推进位于同一 Bus 事务，提交后才通知。完整事件 payload 留存另受 persist 控制：Bus 构造默认 false，部署入口可覆盖；durable 类型名不保证全日志恢复。[B68][B69][B70]

Peri 启发

设计建议：full compact 成功后原子发布新的读取下界；运行、ACP 初载、上滚分页分别验证，不用一个“压缩完成”指标覆盖所有路径。

来源 [B07] [B08] [B09] [B68] [B69] [B70]

OpenCode：插件不是权限替身

OpenCode V2 / 扩展与工具规模

> 扩展点可以统一模型、工具和会话行为，但权限必须在实际副作用入口执行；Code Mode 减少上下文负担，并不天然缩小宿主权限。

能力路径	公开控制合同	不能外推
文件/MCP	各自 action/resource 检查	所有能力统一审批
Session/Worktree	部分操作无独立 action	目录工具名即授权边界
Browser	browser * deny 移除目录	每次页面动作都会问
插件	执行主动声明的检查	注册即自动得到隔离

1. 注册状态与运行资源分离

1.1 V2 插件以 setup 注册能力并可返回 cleanup。transform 按注册顺序重放到新状态；已读取快照不会因后续重建被偷偷修改，注册定义也不等于连接已经就绪。[B11]

1.2 研究判断：这种声明式能力目录可降低热更新时的半状态；代价是 transform 要便宜、可重复，外部配置变化必须明确触发 reload，不能在回调里隐藏不可重放副作用。

2. Code Mode 控制模型可见成本

2.1 execute 让模型编写 JavaScript 调用工具并先过滤中间结果；MCP 默认可经目录间接暴露。其运行时无直接文件系统，但提供 HTTP fetch，不能误写为完全无网络。[B13]

2.2 嵌套能力保留各自权限合同，不是统一逐操作审批：当前官网列出 session 改名/移动与 worktree 创建/列表无独立 action；browser 可从目录移除，却不逐次询问。插件只执行其主动接入的检查。[B13]

3. 组织约束在更高层收紧

3.1 permission 决定 allow/ask/deny；policy 可在其后硬拒绝，不能额外授权。组织与全局规则可覆盖项目的较宽设置，这对共享代码库中的可信边界更清楚。[B12]

3.2 policy 非法条目可能被丢弃并只写告警。研究判断：生产平台应把“规则解析有效”列为发布门禁，并对缺少检查的插件工具做清单审计，而非只检查配置文件存在。

版本与审计

新增例外来自 2026-10-19 官网，不冒充旧固定提交同版实现。建议逐项记录能否发现、间接可达、检查点及副作用；没有逐操作 prompt 不等于完全无控制。[B13]

来源 [B11] [B12] [B13]

Crush：把编辑方法写进提示词

Crush / 指令与工具的紧耦合

> Crush 的特点是把文件编辑方法和终端沟通约束写得非常具体；其启发在可操作性，其风险也来自规则重复和局部矛盾。

提示规则	对应机制	迁移时的改造点
先读再改	view + 精确匹配编辑	定义“相关范围已读”，避免整文件冲突
符号操作优先	LSP rename/replace/references	明确语言服务不可用时的回退
匹配失败补上下文	工具错误驱动再读取	限制无信息增量的重复尝试
简短回复且完成任务	行动深度与文本长度分开	交付仍需保留测试与剩余风险

1. 模板不是孤立文本

1.1 Go prompt.Build 通过 text/template 注入 provider、model、工作目录、Git 状态、项目及全局上下文。skills 先发现元数据再按名称去重，用户包可覆盖内置同名条目。[B17]

1.2 coder 模板将阅读、精确匹配、编辑、测试、继续推进串成工作流；还明确区分 edit/multiedit 与 LSP 的符号替换、重命名、引用及调用关系能力。[B16]

2. 为什么对编码任务有效

2.1 提示词告诉模型如何从工具错误恢复：匹配失败先补读精确空白和上下文，再重试；修改共享代码前检查引用，避免靠文本相似性推断影响范围。[B16]

2.2 研究判断：这种指导比抽象的“认真编码”更能改变行动轨迹。LSP 把符号边界交给语言服务，精确文本编辑作为补充，能够减少一类定位错误，但不保证语义正确。

3. 好规则也可能互相干扰

3.1 同一模板同时强调限量读取与编辑前读整文件，也同时出现强硬的不要提问及允许高风险歧义提问的例外。规则优先级对模型仍有解释成本。[B16]

3.2 研究判断：机械地要求每次修改后测试，会放大微小编辑批次成本；过度压低回复行数又可能削弱交付证据。应按任务阶段和风险限定触发条件，并用真实轨迹验证。

复用边界

当前 Crush 许可为 FSL-1.1-MIT，限制竞争用途；特定版本在其提供之日两周年后可用 MIT。源码可读不代表可按 MIT 直接复用到同类产品，本页机制分析与代码许可分开。[B59]

来源 [B15] [B16] [B17] [B59]

Crush：终止事件必须可关联

Crush / 会话、权限与中断恢复

> Crush 的源码展示了长任务最容易被忽略的部分：每个请求的终止关联、未完成工具调用的收尾，以及并发会话对本地数据库的压力。

标识 / 状态	解决的问题	不能据此推导
sessionID	归属同一会话	某一次请求已结束
RunID	关联排队请求的终止	跨进程恰好一次
toolCallID	恢复调用 / 结果配对	副作用一定未发生
permission key	复用明确授权范围	已形成操作系统沙箱

1. Session ID 不足以表示一次请求

1.1 runid.go 把调用方 RunID 经 context 传给实际运行，再由 RunComplete 回显。忙会话内的排队请求可由此区分，避免订阅者误把另一轮结束当自己的结束。[B23]

1.2 run_marker.go 共享原子标记，记录深层已发布的权威终止事件；外围只在缺失时补发。研究判断：它解决的是归属与去重，不应夸大为跨进程 exactly-once。 [B22]

2. 中断也要形成合法历史

2.1 settle.go 为永远不会返回的工具调用写入合成 error result，并根据取消、max tokens、provider拒绝等原因给模型不同解释。恢复时还修复死进程遗留孤儿调用。[B21]

2.2 修复读取持有 session dispatch lock，先排除仍有运行者补结果的可能。研究判断：工具配对修复必须与所有权协调，否则恢复器本身会制造重复结果。

3. 本地并发需要显式资源政策

3.1 SQLite 连接按路径共享并引用计数，WAL 开启，但最大连接数设为一；数据目录进程锁是可选而非全局默认。这是本地一致性取舍，不是高并发数据库方案。[B20]

3.2 权限服务以 session/tool/action/path 记录复用授权，交互请求串行等待且响应取消。[B26] 短时审批状态、请求终态和持久消息应分别观察，避免所有 UI 状态都依赖一个布尔值。

Peri 启发

设计建议：围绕“发送时正在停机”“取消后结果晚到”“旧进程遗留调用”“重复终止事件”建立最小状态转移验收集。

来源 [B20] [B21] [B22] [B23] [B26]

Crush：窄委派与摘要尾部

Crush / 子 agent 与上下文控制

> Crush 的默认子 agent 更像专职检索助手；独立子会话和摘要后尾部查询共同减少主任务负担，但两者解决的是不同层次的问题。

结构示意 · 边界比较

节点与职责

1. 主任务 · 保留目标与决策
2. 检索子会话 · 搜索与筛选轨迹
3. 最终文本 · 关键路径与结论
4. 完整历史 · 持久记录
5. 摘要边界 · SummaryMessageID
6. 模型上下文 · 摘要 + 后续尾部

有向关系 · 后半句为条件或动作

- 主任务 → 检索子会话 · 窄任务
- 检索子会话 → 最终文本 · 压缩输出
- 最终文本 → 主任务 · 回传
- 完整历史 → 摘要边界 · 总结
- 摘要边界 → 模型上下文 · SQL下界

1. 子 agent 的触发条件很窄

1.1 agent_tool 的描述直接把“不确定一次能找到正确匹配的搜索”交给子 agent，并列出 glob、grep、ls、view。执行采用 ParallelAgentTool，实际工具集合仍由 task profile 允许清单筛选。[B18][B19]

1.2 coordinator 用父 messageID 与 toolCallID 关联子 session，提交明确 prompt，不默认复制父全部历史。运行结束只回传最终文本；空文本、执行失败都有显式错误。[B19]

2. 协作的价值来自收口

2.1 研究判断：把大规模搜索轨迹放到子会话，父模型得到文件、片段与结论，能降低上下文噪声。它不是团队黑板、动态 DAG 或多人修改冲突协调器。

2.2 子任务费用尽力累计到父会话，累计失败不会丢弃已有结果。[B19] 这一取舍强调交付可用性；账务精确性需另设对账，而非让观测失败吞掉业务结果。

3. 压缩也进入数据库查询

3.1 摘要模板明确交接当前状态、文件变化、技术背景、策略和精确下一步；没有硬长度上限，细节保真与摘要膨胀之间仍需评估。[B24]

3.2 摘要成功后设置 SummaryMessageID；getSessionMessages 使用 ListFromSummary，只读取后续尾部，并把摘要转成下一轮用户消息。[B25] 这能减少读全历史的成本，但不代表旧历史删除或所有 UI 缓存释放。

Peri 启发

设计建议：让子 agent description 直接说明何时委派、回传什么；把检索委派的成功率、父上下文节省量与总成本同时评估。

来源 [B18] [B19] [B24] [B25]

Pi：极简是可重组的能力边界

Pi / 提示词与工具目录

> Pi 的核心设计是让提示词随真实能力装配；本次 main 快照已包含 MCP 与 Code Mode，不能再用早期“只有四个工具、没有 MCP”的描述代替源码。

层次	模型得到什么	工程意义
默认编码工具	少量稳定操作	减少起始策略分叉
skill	描述及按需正文	知识按任务加载
deferred / Code Mode	目录与被选中能力	控制 schema 和中间输出成本
扩展	新工具及生命周期逻辑	核心保持可组合，责任边界更重要

1. 明确当前身份与范围

1.1 badlogic/pi-mono 已重定向 earendil-works/pi；本节固定 2026-10-18 main 快照。官网仍把 subagents 和 plan mode 留给扩展，而不是默认产品流程。[B27][B39]

1.2 system-prompt.ts 的默认 selectedTools 为 read、bash、edit、write，但这是默认装载集，不是能力上限。其他工具、扩展和 MCP 可加入实际运行目录。[B29][B35]

2. 提示词从结构生成

2.1 构建器把 preamble、tools、rules、docs、project_context、skills、cwd 分成可独立更新的 section；按可见工具合成行为规则，再去除重复条目。[B29]

2.2 构建器的 hiddenTools 指省略直接声明、仍可间接调用的工具；它不同于 MCP exposure=hidden 的不可达语义。提示段差异可持久记录为 patch，解释能力何时发生变化。[B29][B35]

3. 工具规模改用暴露策略控制

3.1 MCP 支持 direct、deferred、codemode、hidden；SDK 须装载扩展。active 是声明集，deferred/codemode 不必 active 也可间接调用；--tools 只列普通工具时不排除 MCP。[B35][B60]

3.2 Code Mode 用 QuickJS 运行组合脚本，无直接文件系统、网络或计时器；通过工具取数据，在脚本内过滤后才让输出进入模型。[B38] 研究判断：这降低上下文成本，但工具本身仍承载真实副作用。

3.3 已安装的 tool_call handlers 顺序执行，block 短路，异常使该调用失败；参数可被后续 handler 改写。门禁是否存在、顺序及最终参数都须验收，不能把可扩展等同默认有审批。[B30][B31][B61]

Peri 启发

设计建议：把最小工具集作为可测量的默认体验；不要把最小化误解为禁止能力增长，而要设计发现、装载、撤销与审计。

来源 [B27] [B29] [B35] [B38] [B39] [B60] [B61] [B30] [B31]

Pi：真正空闲晚于 agent_end

Pi / 可嵌入循环与扩展契约

> Pi 把低层运行结束与全部自动工作停止明确区分；这是长任务系统避免“发送后停住”与错误完成通知的重要参考。

结构示意 · 反馈循环

节点与职责

1. 输入 · 用户 / steering
2. 模型与工具 · 一个 turn
3. agent_end · 一次 run 结束
4. 自动后处理 · 恢复 / 压缩 / 队列
5. agent_before_settle · 允许最后续跑
6. agent_settled · 最终通知边界

有向关系 · 后半句为条件或动作

- 输入 → 模型与工具 · 启动
- 模型与工具 → agent_end · 无更多工具
- agent_end → 自动后处理 · 后处理
- 自动后处理 → 模型与工具 · 仍有工作
- 自动后处理 → agent_before_settle · 已处理完
- agent_before_settle → 模型与工具 · 显式续跑
- agent_before_settle → agent_settled · 无续跑

1. 一个请求不一定只有一次 run

1.1 agent-loop.ts 的内层循环处理工具调用和 steering，外层在将停止时接收 follow-up。prepareNextTurn 可做压缩，之后再次检查期间到达的 steering。[B30]

1.2 agent_end 只表示一次低层 run 结束；自动重试、恢复、压缩重试、队列工作仍可能触发下一轮。扩展文档把最终通知放在 agent_settled。[B31]

2. 扩展的事件语义很具体

2.1 before_agent_start 改结构化提示和工具，context 变换本次模型消息，tool_call 能改参数或拒绝，tool_result 可修改返回内容；不同事件不能混用返回值语义。[B31]

2.2 agent_before_settle 是可追加条目并申请最后续跑的边界；无条件 continue 会制造循环。通知型事件返回值不起控制作用，因此“在结束回调返回任务”并不天然有效。

3. 运行时扩展同样需要治理

3.1 扩展在 Pi 进程内执行；handler 按装载与注册顺序组合，流式 handler 被顺序等待。慢日志、网络调用或图像处理会把扩展延迟传回流消费。[B31][B34]

3.2 CLI、JSONL、RPC 与进程内 SDK 共用会话机制。[B28] 研究判断：适合把 RSI 验证接在公共事件和状态边界；同时必须区分仅改本次请求与真实写入 session 的变更，保证可复现。设计建议：结束通知绑定具体运行标识和队列代次。重试前后使用同一任务身份，保留独立运行状态。

Peri 启发

设计建议：将 run_end、turn_end、session_idle 分为不同事件；对每个终态写出是否允许再入、是否持锁、队列由谁负责唤醒。

来源 [B28] [B30] [B31] [B34]

Pi：历史树与模型投影分离

Pi / 分支、压缩与可审计恢复

> Pi 保存的是可分支的原始历史，模型使用的是当前分支的投影；压缩、上下文编辑和分支摘要因此可以共存而不抹掉审计记录。

条目	影响模型上下文	原始历史
message / system delta	追加内容或能力变化	保留可重放事实
compaction	摘要替代前缀	旧条目继续存在
context_edit	隐藏或替换目标贡献	不覆写原条目
branch_summary	引入另一分支的经验	原分支仍可访问

1. JSONL 存储的不是线性聊天

1.1 session v3 的条目通过 id/parentId 形成树；活动叶子确定下一次模型请求的历史路径。切回旧点继续会在同一文件分支，fork/clone 才复制到新文件。[B28][B32]

1.2 system message 记录提示段和工具声明的初始值，后续用 sections、toolsAdded、toolsRemoved 追加差异。重放可以解释当时有哪些能力，而非只看到当前配置。[B32]

2. 压缩记录可恢复的边界

2.1 CompactionEntry 保存 summary、firstKeptEntryId 与 system checkpoint；后续上下文由系统状态、摘要、保留范围组成。切点不能落在 tool result，使调用与结果分离。[B32][B33]

2.2 reserveTokens 决定何时压缩，keepRecentTokens 决定保留多少近期内容；两者解决不同预算问题。一个超长用户任务可在 assistant 边界拆分，而不是强保留整个任务到再次溢出。[B33]

3. 隐藏内容与删除历史不同

3.1 context_edit 是追加式的投影编辑：可隐藏或替换某条对模型的贡献，原始消息、导出和费用记录仍在。分支摘要则把离开的路径经验带到新的路径。[B32]

3.2 SessionManager 虽分块解析 JSONL，却把全部条目存入 fileEntries/byId；压缩只追加记录并刷新模型投影，不释放原历史对象。这条恢复路径不是窗口懒加载；真实 RSS、UI 与导出峰值未测。[B71][B60]

Peri 启发

建议分别验收历史保真、活动上下文和客户端初载。Pi 压缩先追加会话条目再刷新投影；同步文件追加不自动提供跨内存/磁盘回滚事务。[B71][B60]

来源 [B28] [B32] [B33] [B71] [B60]

Pi：多智能体是一份可替换扩展

Pi / 官方子 agent 样例的真实边界

> Pi 的官方样例展示了怎样用很少的框架原语做委派；它有真实独立上下文和进程，但并非默认团队系统，也没有自动文件隔离。

组合模式	适合任务	额外需要的治理
单次委派	检索、局部诊断	输出契约与预算
并行委派	互不依赖的只读调查	取消、限流、证据归并
顺序链	分析→规划→执行	失败停止、每步验收
并行写入	样例本身不足以保证安全	隔离工作区、冲突检测与合并

1. 样例如何产生一个孩子

1.1 examples/extensions/subagent 通过 spawn 启动独立 pi 子进程，用 JSON 模式收集流事件；Markdown frontmatter 描述 agent 名称、工具、模型，正文成为角色提示。[B36][B37]

1.2 未指定模型时继承派发会话的模型与思考级别；项目 agent 默认不加载，显式启用 project/both 后才进入发现范围，非可信项目还可要求额外确认。[B36]

2. 三种组合不等于同一种协作

2.1 single 是单次任务；parallel 最多八项、四项同时运行；chain 将前一步结果通过 previous 占位符传给下一步，遇到失败即停止。[B36][B37]

2.2 示例流水线不提供动态任务图或持续团队通信。parallel 给父模型的每项最终输出有 50 KiB 上限，完整内容留在工具详情；输出限额与运行隔离是不同合同。[B36][B37]

3. 退出和输出也有契约

3.1 孩子以 --no-session 启动，不提供持久子会话续接。abort 向直接孩子发 TERM 并等 close；五秒后备以！killed 判断，Node 中 killed 只证明信号已发送，不能保证五秒退出或后代树已回收；未实测。[B37][B67]

3.2 独立进程仍使用同一用户权限和工作目录资源；项目 trust 也不是沙箱。[B34] 研究判断：若让多个 worker 写代码，需在上层加工作树分配、文件范围所有权、验收与合并策略。

Peri 启发

建议核心分别报告 cancel_requested、process_exit 与 pipes_closed；验收时测试忽略 TERM 的孩子和持有管道的后代。以上是建议实验，不是本轮产品运行结果。

来源 [B34] [B36] [B37] [B39] [B67]

Goose：薄主提示与厚运行管线

Goose / Rust 核心与扩展装配

> Goose 的基础系统提示相当薄，复杂能力主要来自扩展、配方与有序运行操作；理解它必须同时看提示装配和执行状态机。

结构示意 · 职责分层

节点与职责

1. 客户端 · 桌面 / CLI / 服务
2. 提示构造 · 基础 + hints + 配方
3. 持久会话 · 消息与配置
4. 有序操作 · 审批 / 工具 / 恢复
5. 模型推理 · 供应商适配
6. 扩展能力 · MCP与平台工具

有向关系 · 后半句为条件或动作

- 客户端 → 持久会话 · 提交
- 持久会话 → 提示构造 · 装配
- 持久会话 → 有序操作 · 判断
- 提示构造 → 模型推理 · 上下文
- 有序操作 → 模型推理 · 允许推理
- 模型推理 → 有序操作 · 候选工具调用
- 有序操作 → 扩展能力 · 审批后执行
- 扩展能力 → 持久会话 · 经核心保存结果

1. 产品边界与当前快照

1.1 block/goose 已重定向 aaif-goose/goose，官网确认归属 AAIF。本节固定 2026-10-18 main，桌面、CLI及服务接入共享 Rust 核心，不把界面功能当独立 agent 引擎。[B40][B41]

1.2 system.md 主要定义身份、扩展上下文和 Markdown 响应；专门任务通过额外指令、hints、recipe 及工具说明形成行为，不靠一个巨大通用人格提示包办。[B43]

2. 提示构造关注稳定性

2.1 PromptManager 支持 session override 与有序 extras，加载工作目录 hints；使用扩展列表的构建路径按名称排序，并清理扩展指令中的特定 Unicode 标签。[B42]

2.2 日期上下文按小时固定，源码注释说明用于缓存稳定性。研究判断：动态内容的排序与时间粒度会影响前缀复用；但清理标签不能消除工具返回内容中的语义注入。

3. 控制逻辑成为有序 Operation

3.1 state_machine_builder 按顺序放置审批、技能/配方、工具执行、子 agent、steer、压缩、重试、停止 hook、最大轮次，再进入推理。[B47]

3.2 Hook 名称不代表否决权：PreToolUse 走阻止型入口；BeforeShellExecution、BeforeReadFile 和 PreToolUseResult 走观察路径。Hook 故障默认允许，关键 PreToolUse 需显式 on_failure:block；显式 deny 与脚本失败不是一回事。[B62][B63]

Peri 启发

设计建议：比较"核心有多少功能"时，分别统计机制与策略。小 system prompt 不等于弱能力，复杂状态机也不应迫使提示词解释全部实现。

来源 [B40] [B41] [B42] [B43] [B47] [B62] [B63]

Goose：委派配置先保存再运行

Goose / 显式交接与父子生命周期

> Goose 当前委派以持久化子 session 为中心，先确定模型、工具与交付格式，再执行任务；逻辑上下文隔离不能被误写为操作系统隔离。

阶段	持久事实	失败处理价值
创建	父子ID、recipe、模型	可确定是哪一个任务
配置	扩展选择、目录、轮次	重建不依赖父瞬时变量
执行	子 conversation 与工具结果	可检查中途停在哪里
完成	成功 final_output 记录	避免把任意最后一句当交付

1. load 与 delegate 是不同动作

1.1 summon.load 把来源知识读入当前上下文；delegate 将临时 instructions 或 recipe/agent source 转成子任务。delegate 说明明确孩子只知道任务及来源内容，不自动拥有父全部历史。[B45]

1.2 创建前解析有效模型、max_turns、工作目录和扩展集合；参数可缩窄扩展列表，工作目录需在父范围内。子 session 保存 recipe、模型、extension selection 后再写入启动消息。[B45]

2. 终态是可验证输出

2.1 SubAgent 保存父 ID；Operation 验归属、复用成功 final_output，缺省 schema 为 summary。正常结束读取持久成果时，成功 final_output 优先于晚到 cancel；实时主要上浮审批/询问而非全部 token。[B45][B46][B50][B56]

2.2 子 agent 模板写有范围与轮次要求，代码也拒绝它继续 delegate，限制并不只存在于提示词。[B44][B45] 研究判断：预算、角色和递归限制必须分别在运行时落地。

3. 不把“独立”写成不存在的隔离

3.1 当前 handler 使用共享 Arc<Agent> 与 tokio::spawn，AbortOnDropHandle 负责取消；这证明独立执行任务，并不能证明一个孩子对应独立 OS 进程。[B46]

3.2 源码记录旧后台机制已移除；父取消触发尽力异步清理，不证明外部副作用回滚或进程退出后清理完成。[B46][B56] 官网仍有 process isolation 表述；本节仅描述固定快照，不外推全部稳定版。[B55]

Peri 启发

建议明确成功写入与取消的优先级，分开待决定请求、成果交接和代码验收。上述优先级限正常收尾路径，不能推广到 task drop；取消点击不证明成果或副作用均未发生。

来源 [B44] [B45] [B46] [B50] [B55] [B56]

Goose：配方把交付变成契约

Goose / 可复用工作流与结构化结果

> Goose Recipe 把指令、参数、工具、模型和结果结构打包；最值得学习的是可检查交付，而不是把 YAML 文件当作确定性流程引擎。



1. 配方声明一份可复用任务

1.1 Recipe 可声明 instructions、初始 prompt、parameters、extensions、settings、response、retry 和 sub_recipes；参数模板将同一方案复用于不同目录、语言或审查目标。[B48]

1.2 官方仍将子配方标为实验功能。子配方通过名称与参数接入；当前 summon 把来源统一转成 recipe 再创建 child。研究判断：统一表示减少“临时任务”和“保存模板”两套执行逻辑的漂移。[B45][B49]

2. 输出由工具收集与验证

2.1 response.json_schema 生成 final_output 工具；代码验证 schema，并从成功的工具响应查找实际输出，而非仅相信模型发出一段像 JSON 的文字。[B50]

2.2 执行收集路径在 RecipeOperation 判定：同一 assistant 连续块有未标为外部执行的其他工具请求时，不收集 final_output；schema 成功仅证明结构合格。[B50][B74]

3. 把确定性留给可验证环节

3.1 retry checks 仅在配置存在且进入对应结束分支时运行；goal/grind 分支可能先处理，不能等同每份结果都经业务验收。[B75] 研究判断：检查脚本要独立；只查文件存在可能形式达标而目标未完成。

3.2 模型仍决定何时调用子配方和如何解释说明，不能声称天然拥有任务 DAG 的事务、自动回滚或跨 agent 文件冲突协调。设计迁移时应明确哪些边由模型选择，哪些必须由调度器硬执行。验证器失败应保留具体证据，并消耗明确的重试预算。

Peri 启发

设计建议：AOS 工作流的每个节点声明输入、允许能力、结果 schema、验证器和失败路由；模型总结只是一项证据，不是验收权威。

来源 [B45] [B48] [B49] [B50] [B74] [B75]

Goose：持久保真与运行限额并行

Goose / 压缩、大输出与审批恢复

> Goose 把历史可见性、费用账本和大工具输出分别处理；这提供了可复盘基础，但压缩和文件落盘都不自动消除峰值内存。

机制	直接收益	必须额外验证
agent_visible	裁剪模型投影而保留历史	该主路径全读；另测客户端窗口
usage ledger	摘要后仍能累计成本	父子归集与重复写入
大输出临时文件	减少后续模型输入	首次峰值、文件寿命、落盘失败
持久审批状态	跨界面重建待决请求	取消、重复批准和工具有效性

1. 压缩前先补齐预算信息

1.1 最新工具结果通常尚未出现在上一轮 provider usage 中；CompactionOperation 单独估算这些结果，再判断是否达到阈值。自管上下文的 provider 可跳过本地自动压缩。[B51]

1.2 context overflow 有受限恢复路径，压缩失败会 yield 并告知错误。研究判断：阈值必须覆盖“上轮计费 + 尚未上报工具数据”，否则大型工具批次会绕过正常预算判断。

2. 历史与费用不随摘要消失

2.1 session_manager 使用 SQLite；save_compacted_conversation 在 BEGIN IMMEDIATE 事务里更新原消息 agent_visible，并插入新增摘要，而不是覆写所有历史内容。[B52]

2.2 主 SessionLoader 调 get_session(true)，恢复 SQL 无 agent_visible 过滤并 fetch_all；压缩后 Conversation 仍留不可见旧对象。保真不是窗口懒加载或内存释放，实际 RSS 未测；不外推所有客户端。[B52][B72][B73]

3. 大输出与审批各有失败边界

3.1 文本超过默认 200000 字符时写临时文件并回传路径；落盘失败会回传原全文。[B53] 这是已经收齐输出之后的转存，不能宣称流式背压或固定内存上限。

3.2 审批 Operation 从持久请求及响应重建状态，只处理当时被广告的工具调用，并设置 executable 标记。[B54] 研究判断：审批状态可恢复比仅 UI 弹框可靠，但真实文件、网络和进程隔离仍需独立部署边界。

Peri 启发

摘要/可见性在一个 SQLite 事务内；但 usage 记录、该事务、usage 更新分步 await，之后发 HistoryReplaced，不能称全链原子。本轮未测步骤间崩溃与幂等恢复。[B52][B72]

来源 [B51] [B52] [B53] [B54] [B72] [B73]

Cline：共享核心的边界

Cline / 2026-10-17 固定提交

> 当前 Cline 已从单一编辑器插件演进为共享智能体核心；最有价值的结构是把推理循环、持久化会话和宿主界面分开拥有。

结构示意 · 职责分层

节点与职责

1. IDE / CLI / 桌面
2. Core 会话权威
3. Agent 迭代循环
4. 模型适配
5. 会话与团队存储
6. Hub 多客户端

有向关系 · 后半句为条件或动作

- IDE / CLI / 桌面 → Core 会话权威 · 本地
- IDE / CLI / 桌面 → Hub 多客户端 · 远程
- Hub 多客户端 → Core 会话权威 · 会话
- Core 会话权威 → Agent 迭代循环 · 组合
- Agent 迭代循环 → 模型适配 · 请求
- Core 会话权威 → 会话与团队存储 · 持久化

1. 先纠正研究基线

1.1 当前主仓库同时提供 SDK、命令行、桌面与编辑器入口；README 明确 JetBrains 插件暂不开源。因此“Cline 开源”不意味着每个宿主实现均能审阅。[C01]

1.2 本章固定到 a42df8c8 的源码。旧版插件中的路径、工具名和提示词不能直接当作现状；跨版本对比应先记录宿主、核心提交和启用的配置。

2. 控制权如何分层

2.1 shared 放类型与契约，llms 集中供应商差异；agents 承担无持久化的迭代、工具和事件循环；core 再组合会话生命周期、存储、压缩与扩展。上层应用消费核心，避免供应商逻辑散落在界面层。[C02]

2.2 Hub 路径把会话权威留在可独立运行的服务，客户端连接和断开不等同停止执行；多个界面可以继续观察同一会话。审批及事件需要经过核心转发，不能依赖某个标签页一直存活。[C03]

3. 收益与工程代价

3.1 共享核心减少各宿主语义漂移，却要求显式处理运行 ID、流式终态、恢复和配置归属。界面看到“已连接”，只说明传输状态，不代表某个任务正在运行。

3.2 本报告建议用事件契约而非聊天文本推导业务状态：完成、取消、等待审批应有独立语义，恢复后由权威快照校准，再接增量事件。该建议不是对 Peri 当前实现的结论。

Peri 启发

设计建议：Peri 与 AOS 可以分别拥有执行语义和工作空间编排，但应共享明确的运行状态契约；界面生命周期不应隐式支配后台任务。

来源 [C01] [C02] [C03]

Cline：提示词是装配产物

Cline / 模式、宿主与工具语义

> 提示词的亮点不是一段万能长文，而是把模式、当前环境、工作区规则及宿主能力装配成与真实工具集合一致的运行契约。

装配输入	作用	失败边界
模式标签	识别当前 Plan / Act	旧历史许可不应继续生效
工作区规则	带入项目约定	越多不一定越有效
宿主工具能力	决定如何切换模式	不存在的工具会造成空转
并行指令	减少独立请求往返	并行写入需额外约束

1. 从源码看装配顺序

1.1 Act 模板要求先获得代码约定和验证方法，随后执行并验证；它显式鼓励把独立读取、搜索和命令放在同一轮，避免不必要的往返。这是行为引导，实际并发仍由工具执行器决定。[C04]

1.2 buildClineSystemPrompt 选择 Act 或 YOLO 基础模板，再填环境、规则和模式说明；Plan 复用基础结构并追加约束。自定义 overridePrompt 则走覆盖路径，不能假设仍自动保留全部默认规则。[C05]

2. 防止历史模式污染当前模式

2.1 用户输入携带 mode 标签，模式变化还有切换提示；当前最新输入的模式决定本轮语义。这样旧消息曾允许写文件，不会仅因留在历史里就被解释为现在仍可修改。[C05]

2.2 宿主是否提供 switch_to_act_mode 会改变提示词：支持工具切换的宿主与要求用户操作开关的宿主使用不同说明。复制提示词却不复制工具可用性，会制造无法执行的指令。

3. 运行时守卫有明确覆盖范围

3.1 Plan 预设保留 run_commands、关闭编辑工具；builder 实际装载 beforeTool 守卫，先于策略和审批检查命令。任一条命中黑名单则整批 skip，返回错误但不终止整轮。[C58][C59][C61][C62]

3.2 守卫识别常见写命令、部分参数及重定向，允许特定临时目录输出；它不解释任意 shell 或解释器程序。可学习“提示 + 执行前拦截”，不可直接称为完备只读沙箱。[C60]

文档与实现

官方 Plan 页面写不能执行命令；本页固定 SDK 快照保留只读调查命令并加有限守卫。须按宿主、提交和实际工具面核实，不将差异写成已复现生产漏洞。[C63]

来源 [C04] [C05] [C58] [C59] [C60] [C61] [C62] [C63]

Cline：两层委派机制

Cline / 单次子任务与持续团队

> Cline 把一次性子智能体与可恢复团队分为两层；前者缩小上下文，后者增加任务、消息、异步运行和汇总结果的显式状态。

结构示意 · 关系与数据流

节点与职责

1. 协调者
2. 任务板与邮箱
3. 专用子运行
4. 同步 / 异步执行
5. 结果与状态
6. 会话归属存储

有向关系 · 后半句为条件或动作

- 协调者 → 专用子运行 · spawn_agent
- 专用子运行 → 结果与状态 · 等待后返回
- 协调者 → 任务板与邮箱 · 团队领取与分配
- 任务板与邮箱 → 同步 / 异步执行 · team_run_task
- 同步 / 异步执行 → 结果与状态 · 状态摘要
- 结果与状态 → 协调者 · 回流
- 任务板与邮箱 → 会话归属存储 · 团队记录
- 结果与状态 → 会话归属存储 · 仅团队结果保存

1. 单次委派如何执行

1.1 spawn_agent 接受专用 systemPrompt 和 task，创建子运行后等待 run 完成，向父任务返回文本、迭代数、结束原因和用量。工具声明可并行，表示多个调用具备并行执行资格，并不表示父流程无需等待。[C06]

1.2 委派配置显式携带父身份、供应商连接、工具集合、审批回调和取消信号。独立上下文是模型输入边界；工具若仍访问同一目录，文件副作用不会自动隔离。[C07]

2. 团队增加哪些协议

2.1 启用 enableAgentTeams 后，Teams 提供成员、任务板、邮箱、任务运行和结果片段；team_run_task 区分同步等待与异步运行，claim 只领取任务，不等于工作已经开始或成功。[C08][C09]

2.2 终态进入 pendingTeamRunUpdates，回流的是运行 ID、状态、错误和轮次等摘要；这不等于自动注入完整子任务正文。未取消且父 run 以 completed 或 max_iterations 结束时，核心才考虑自动续接，不能承诺任意终态都会唤醒父任务。[C09][C10]

3. 协作不只靠角色名

3.1 team_cancel_run 关闭 run 记录并使等待返回，内建路径不直接 abort 活跃成员；晚到结果也不覆盖 cancelled。父取消、成员 shutdown 另有 abort 链，宿主回调可能再加行为；不能把逻辑取消当副作用已停止。[C09][C65]

3.2 团队记录按会话保存，恢复记录不等于执行栈恢复。建议子任务报告范围、变更、验证与未决问题；父任务承担集成验收，并计入协调和失败恢复开销。[C08]

Peri 启发

建议一次性委派与长期团队共享运行内核，但分别定义取消请求、run 终态与成员实际停止。邮箱标已读也不代表行动完成；不要由一个 spawn 返回值承诺全部协作生命周期。

来源 [C06] [C07] [C08] [C09] [C10] [C65]

Cline：压缩与恢复是两件事

Cline / 长任务的状态预算

> 上下文压缩决定下一次模型看到什么，检查点决定代码如何恢复；两者可以关联，但不能把一次摘要当成完整任务状态的可靠快照。

机制	主要保存内容	不能据此推导
上下文压缩	摘要与保留尾部	所有运行内存已经下降
会话历史	对话与任务元数据	外部系统已经回滚
Git 检查点	被覆盖的文件状态	进程和数据库事务隔离
恢复基线校验	目标提交与当前 HEAD	任意时间点都能安全恢复

1. 摘要也需要预算

1.1 agentic-compaction 在调用摘要模型前计算其输入上限，扣除已有摘要与请求开销，再投影待折叠消息。摘要模型也可能超限，因此“主模型满了就完整交给摘要模型”不是可靠路径。[C11]

1.2 空摘要或预算构建失败会跳过此次摘要；成功后保留近期尾部，并补充文件操作章节。基础压缩另有预算处理逻辑。源码存在消息压缩，不足以证明 UI、观察系统或其他引用已经释放内存。[C11][C12]

2. 文件恢复有自己的事务

2.1 检查点由 Git 相关钩子记录到会话元数据。恢复前通过私有引用保存现场，失败时具备回滚路径；未跟踪文件也要按检查点是否捕获它们分别处理。[C13][C14]

2.2 恢复比较当前 HEAD 提交与目标基线；哈希不同时拒绝破坏性恢复。新增提交、重写历史或切换到不同提交的分支可能触发；仅切换到同一提交的另一分支，并不必然被此检查拒绝。[C13]

3. 面向长程运行的验收

3.1 本报告建议分开测试上下文连续性、文件一致性和外部副作用。数据库、远程部署、后台进程不在普通 Git 检查点的回滚覆盖内，恢复对话也不会自动撤销这些动作。

3.2 摘要保存 userRunSpan 对齐运行编号，却不保留内部各对话切点。消息恢复目标落在摘要跨度内部会明确拒绝；仅恢复工作区可另选。有文件 checkpoint 不等于任意历史对话都能精确回退。[C11][C13]

Peri 启发

设计建议：长任务验收应同时展示模型上下文大小、持久化历史大小、进程常驻内存和恢复成功率；四者不能用同一个“已压缩”指标替代。

来源 [C11] [C12] [C13] [C14]

Roo：模式是能力包

Roo Code / 已停运项目的设计遗产

> Roo 的主要价值是把角色提示、使用时机、工具组和文件范围放进同一个模式定义；它值得研究，但必须按已停运的代码基线解读。

模式元素	实现目的	设计提醒
roleDefinition	形成角色与目标	角色文本不是权限
whenToUse	帮助模型选择模式	要写清触发条件
groups / fileRegex	缩减工具和编辑范围	仍需执行时校验
customInstructions	定义阶段性工作法	避免强制步骤无差别触发

1. 生命周期必须先说清

1.1 官方文档写明扩展于 2026 年 5 月 15 日关闭，GitHub 同日归档只读。本章分析 b867ec91 的实现，不把旧文档仍能打开理解为产品继续维护。[C15]

1.2 停运不抹去架构价值，却改变采用决策：可以学习接口与演进教训，不能把安全修复、依赖升级或未来模型兼容当作现成承诺。

2. 一份模式承担四种职责

2.1 模式结构包含身份说明、适用情景、补充流程和工具 groups；编辑组还能附 fileRegex。Architect 允许受限 Markdown 编辑，Ask 保留读取与 MCP，Orchestrator 把直接工作交给委派机制。[C16]

2.2 Debug 的提示重点是先展开原因、收敛假设、补观测再修复；Architect 则强调可执行待办。亮点是为任务阶段规定行为，而不只是把“你是专家”换成不同职业名称。[C16]

3. 角色边界与授权边界

3.1 模式切换仍是同一流程中行为和工具可用性的变化，并不自动创建一个并行执行者。只有真正创建独立任务，才出现新的上下文和父子关系。

3.2 工具组限制也不等于所有副作用被隔离：例如 MCP 可能连接外部写操作，必须另查工具语义和审批。本报告建议为每个角色同时写明能做什么、输出什么、何时移交和失败怎么办。

Peri 启发

设计建议：Peri 可以把任务类型、工具可见性和交付契约一起评测；仅修改系统提示，容易让模型既负责规划又无边界地直接动手。

来源 [C15] [C16]

Roo：提示与工具分轨生成

Roo Code / 模型适配与能力交集

> 归档版本的 Roo 使用原生工具调用，提示词段落与工具 schema 分开生成；模型适配可以换工具形态，但不应扩张模式本身的能力。

处理阶段	输入	输出约束
模式筛选	角色工具组	限定能力上界
模型适配	排除 / 加入 / 别名	不突破允许组
运行条件	索引与服务器状态	只暴露实际可用能力
提示装配	规则与环境	描述须与真实工具一致

1. 不要沿用过时描述

1.1 system.ts 先取得模式角色，再组合格式、工具使用原则、能力、模式、技能、规则、系统信息与自定义指令；MCP 能力说明还取决于模式是否允许及是否存在服务器。[C17]

1.2 该提交明确采用 native-only，toolsCatalog 在系统提示中为空。旧版文档将详细工具说明列入提示词，并不能证明最后版本仍采用相同方式；研究必须以相同提交的代码为准。

2. 工具选择是多条件交集

2.1 filterNativeToolsForMode 从模式工具组出发，再经过允许性检查、模型工具适配与运行条件过滤。模型可排除不擅长的工具，也可指定别名或更合适的编辑工具接口。[C18]

2.2 includedTools 只有属于该模式允许组时才会加入，因此“某模型喜欢某工具”不能自然升级为额外权限。这比为每个模型复制整套权限表更易保持一致。[C18]

3. 提示词维护的可迁移方法

3.1 本报告建议把角色、工作法、环境事实和工具协议四部分分别管理。角色适配通常是语义调整，工具适配可能改变参数和行为，二者需要不同的回归用例。

3.2 应记录最终发给模型的工具名称与 schema 版本；别名映射错误会让模型看见一个名字而执行器识别另一个名字。工具不可用时还应移除对应提示，避免反复尝试造成空转。

Peri 启发

设计建议：针对 Peri 的工具冲突审查，可优先检查“名字、schema、可见性、执行授权”是否来自同一能力注册表，避免多处手写清单长期漂移。

来源 [C17] [C18]

Roo：回旋任务的真实语义

Roo Code / 序列式层级委派

> Boomerang 的强项是用独立子上下文承接阶段工作，再把摘要送回父任务；它是一种层级委派，不能直接称为多智能体并行团队。

结构示意 · 反馈循环



1. 从工具调用走到新任务

1.1 new_task 先验证模式、消息及可选待办，再请求审批，随后调用 delegateParentAndOpenChild。源码注释明确把子任务设为唯一活跃任务，父流程转入委派状态。[C19]

1.2 宿主维护父子关系及恢复路径；子任务结束后，父任务得到摘要继续决策。这里关键对象是 Task 与任务栈，而不是让一个模型在同一历史里连续扮演多种角色。[C24]

2. 上下文隔离的收益与损失

2.1 官方说明子任务不会自动继承完整父历史：必要背景随 message 下传，结果通过完成摘要上返。大量读文件、编辑和试错细节留在子历史里，父上下文保持可管理。[C20]

2.2 同时，摘要成为信息瓶颈。父任务若没收到未解决风险、测试条件或改动位置，就可能过早推进。上下文隔离也没有自动提供独立 worktree、进程或外部服务环境。

3. 为何模型愿意委派

3.1 Orchestrator 默认不给直接读写和命令工具，把详细实现留给子任务；这既保护协调上下文，也改变模型最容易选择的路径。单纯鼓励“多用子智能体”未必有同样效果。[C16][C20]

3.2 本报告建议按依赖关系选择串行或并行：架构结论尚未确定时先串行；互不修改重叠资源的调查可并行。委派数量不是质量目标，可靠的结果交接才是。

Peri 启发

设计建议：若 Peri 默认不愿委派，先审查协调角色是否仍拥有全部实现工具，以及子任务结果能否稳定回流；提示词只负责其中一部分。

来源 [C16] [C19] [C20] [C24]

Roo：状态与检查点的边界

Roo Code / 扩展宿主驱动的执行

> Roo 把任务状态投影给 CLI，并用影子 Git 管理文件检查点；这两项提高可操作性，也暴露了从消息推导状态和文件回滚覆盖面的限制。

状态或机制	语义	容易混淆之处
交互 ask	审批或问题待回答	不是任务成功结束
command_output	命令仍在运行	不是所有 ask 都阻塞
文件检查点	恢复被跟踪文件	不是进程与数据库快照
文件加任务恢复	同时裁剪后续历史	需要重新核对现实状态

1. 状态如何抵达用户界面

1.1 CLI 复用扩展宿主的消息：ExtensionClient 接收 ClineMessage，StateStore 保存消息，MessageProcessor 更新后推导状态；ask 被区分为交互等待、空闲、可恢复或非阻塞输出。[C21]

1.2 因而“最后出现 ask”并不足以判断结束，partial 和 ask 子类仍影响语义。该 CLI 文档还明确默认自动批准工具与命令，只有 require-approval 模式才转为手动，不能用 IDE 默认行为推导所有宿主。

2. 文件恢复与会话恢复分开

2.1 检查点使用独立影子 Git；用户可只恢复文件，也可连同后续任务消息一起回退。服务负责初始化、提交和差异等操作，项目原有提交历史与检查点存储并非同一个概念。[C22][C23]

2.2 文档指出检查点在任务开始和文件修改前触发，不保证每次命令前自动建立；被排除文件、外部数据库和进程也不能简单纳入回滚承诺。因此“能撤销 Agent 工作”必须说明覆盖范围。

3. 对长程运行的反向启发

3.1 本报告建议将等待用户、等待工具、任务完成、执行失败和传输断开分别建模；聊天消息用于展示，权威状态用于驱动调度。这样停止或续跑不会依赖最后一条展示消息的偶然形态。

3.2 恢复检查点后要重新观察工作区，而不是只重放旧界面。文件状态和对话状态可以独立恢复，恢复结果也必须明确告诉后续智能体，避免按过期假设继续修改。

Peri 启发

设计建议：对 Peri 的“发送后仍停住”类问题，可以用明确的状态转换日志做回放；不要仅凭 UI 是否还显示最后一条消息判断主循环状态。

来源 [C21] [C22] [C23]

Kilo：一个服务承载多目录

Kilo Code / 新架构与编辑器边界

> 当前 Kilo 以 OpenCode 派生的 CLI 作为本地智能体引擎，编辑器是客户端；多目录实例共享服务进程，隔离粒度必须逐层说明。

结构示意 · 关系与数据流

节点与职责

1. 多视图客户端
2. SDK / SSE 桥
3. 共享 serve 进程
4. 目录实例路由
5. 工作区实例
6. Worktree 实例

有向关系 · 后半句为条件或动作

- 多视图客户端 → SDK / SSE 桥 · 请求与事件
- SDK / SSE 桥 → 共享 serve 进程 · directory
- 共享 serve 进程 → 目录实例路由 · 规范化
- 目录实例路由 → 工作区实例 · 选择
- 目录实例路由 → Worktree 实例 · 选择
- 工作区实例 → SDK / SSE 桥 · 目录事件
- Worktree 实例 → SDK / SSE 桥 · 目录事件

1. 基线已发生变化

1.1 当前 README 明确 CLI 来自 OpenCode fork；本章固定 2a1903f2，不能把早期 Roo 衍生扩展的内部结构直接套到现状。专用 Orchestrator 也已经标记弃用，委派成为主智能体可用能力。[C25][C32]

1.2 VS Code 扩展按需启动编辑器拥有的 kilo serve 子进程，经生成式 SDK 请求与全局 SSE 接收事件。它与可复用的独立 daemon 是不同的生命周期所有者。[C26][C27]

2. 目录上下文如何路由

2.1 同一个服务可处理工作区和多个 worktree；请求带 directory，由 InstanceStore 规范化后选取目录实例。事件再携带目录信息，客户端继续按目录与会话过滤。[C26]

2.2 扩展宿主还对高频流更新做合并后发给 Webview。界面订阅范围与执行实例范围因此是两次边界判断，不能只在渲染端隐藏不相关消息。[C27]

3. 共享带来的真实代价

3.1 目录状态分离不意味着服务内部所有状态都分离；官方架构特别指出进程级服务状态仍共享。相同进程的崩溃、资源压力和某些全局策略会影响多个工作区。

3.2 本报告建议为 AOS 工作空间分别定义目录、会话、进程和沙箱四种身份；协议中携带并校验作用域。重连先重建权威快照，再消费增量，以免事件丢失或串到其他会话。

Peri 启发

设计建议：若复用一个宿主连接多个 Peri 会话，应把隔离承诺精确到具体状态所有者；“每个会话一个目录”并不覆盖进程共享资源。

来源 [C25] [C26] [C27] [C32]

Kilo：模型适配和委派约束

Kilo Code / 提示词、记忆与权限

> Kilo 将模型专用模板、项目记忆、技能目录和委派权限分别处理；值得学习的是装配依据和约束传播，而不是某个模型的一段口令。

输入	归属	应验证的问题
模型模板	模型适配层	是否选中预期版本
项目记忆	上下文层	是否过时或只是一份索引
智能体描述	委派选择层	是否明确适用范围
工具权限	执行策略层	恢复后是否仍正确收紧

1. 模型差异不只在 API

1.1 SystemPrompt.provider 优先读取模型的显式 prompt 类型，再按模型标识回退到不同模板；技能说明和 MCP 指令另行生成，并根据智能体权限过滤。这是源码可见的模板选择，不是逆向闭源提示词。[C28]

1.2 项目记忆通过独立块注入，提示要求按需查询详细记录，并明确记忆属于上下文，当前用户、仓库和工具事实优先。记忆索引不是完整历史，不能凭摘要假装精确恢复旧结论。[C29]

2. 明确哪些限制会传给子任务

2.1 task 校验恢复对象的 parentID，并合并当前限制；但继承是选择性的：原生角色的 edit 拒绝、部分 notebook/MCP 拒绝及 session 拒绝继续约束孩子，自定义父 profile 的 edit 和广泛 bash 拒绝不会原样投射。不能声称子权限必为父权限子集。[C31][C35]

2.2 子任务不能直接向最终用户提问；需要通过结果、父任务或允许的协作渠道沟通。自定义描述用于帮助模型选择专长，权限字段则负责工具允许、询问或拒绝，两者职责不同。[C30][C34]

3. 可迁移的评测方法

3.1 本报告建议用同一组任务分别评估模板适配与权限执行：模型是否选对工具是一项，越权调用是否被阻断是另一项。不要因提示词说“不修改”就跳过执行边界测试。

3.2 对旧子任务恢复应重验父子身份、当前策略、模型与工作区；长时间任务里配置可能已经改变。能继续历史并不代表可以继续持有过去全部能力。

Peri 启发

设计建议：Peri 的技能、模型提示和工具授权可以关联，但应分别版本化。任何“恢复会话”路径都应重新计算当前有效能力，避免历史配置成为隐式授权。

来源 [C28] [C29] [C30] [C31] [C34] [C35]

Kilo：并行、工作树与消息板

Kilo Code / 三种协作对象

> Kilo 明确区分子智能体、顶层工作树会话和 Swarm 消息板；执行、隔离和通信是三个问题，拥有其中一种不代表另外两种自动成立。

结构示意 · 边界比较

节点与职责

1. 主会话
2. Task 子会话
3. Swarm 消息板
4. Agent Manager
5. 顶层会话 A · 可选worktree
6. 顶层会话 B · 可选worktree

有向关系 · 后半句为条件或动作

- 主会话 → Task 子会话 · 创建
- Task 子会话 → 主会话 · 结果回传
- 主会话 → Swarm 消息板 · 发帖
- Task 子会话 → Swarm 消息板 · 发帖
- Swarm 消息板 → 主会话 · 显式读取
- Swarm 消息板 → Task 子会话 · 显式读取
- Agent Manager → 顶层会话 A · 管理
- Agent Manager → 顶层会话 B · 管理

1. 子任务有前台和后台

1.1 task 默认前台等待，background 模式立即返回；完成或失败后由运行时把结果送回父会话。源码还处理取消、恢复与增量费用传播，说明后台任务需要生命周期协议。[C31]

1.2 子智能体有独立历史，但共享父目录或 worktree；多个子任务可并发，仍可能改同一文件。后台只是调度语义，不能据此宣布文件隔离。[C30][C32]

2. 独立工作与协作分别建模

2.1 Agent Manager 提供独立顶层会话，可选创建新 worktree，也可复用本地目录；同一 worktree 内多个会话仍共享检出文件和分支。Git 工作树分隔代码现场，并不自动分隔端口、数据库或机器凭据。[C33]

2.2 Swarm 面向一个主会话及其 task 后代。board_post 和 board_read 交换信息，但消息保存不等于接收、阅读或执行；它不会创建、唤醒、暂停或取消智能体。[C33]

3. 避免把图画成假调度器

3.1 本报告建议在协作图上区分实线控制流与消息流。任务完成由运行时确认，消息仅传递证据和建议；这样“收到同事留言”不会被误判为重新启动任务的命令。

3.2 AOS 若采用工作空间内多智能体，应同时规定文件所有者、共享服务端口、结果验收和清理范围。分支合并只是交付步骤之一，还需要整体验证来识别跨任务冲突。

Peri 启发

设计建议：消息 API 应返回“已存储”“已投递”“已读取”“已采取行动”等可区分证据；不要让一个笼统的发送成功掩盖执行者已经结束。

来源 [C30] [C31] [C32] [C33]

Continue：可组合配置的遗产

Continue / 最终 2.0.0 与模块拼装

> Continue 的可借鉴点是把模型、规则和工具作为可组合配置管理；当前研究应先承认官方已宣布停维，避免将历史云服务能力视为现有承诺。

资源	适合共享的部分	应独立管理的部分
模型	角色与调用参数	凭据与实际可用性
规则	工程约定与来源	路径范围和覆盖顺序
MCP 工具	接口与服务定义	连接、授权及副作用
项目配置	项目默认组合	用户临时选择与运行状态

1. 最终版本不等于活跃服务

1.1 仓库 README 明确说明不再积极维护、对所有用户只读，并发布 VS Code、CLI 和 JetBrains 最终 2.0.0，移除了认证和匿名遥测。本章固定 5522c6f4，不将其混称 GitHub 归档状态。[C36]

1.2 文档仍保留共享配置块等内容；可据此分析配置设计，不能据页面可访问就承诺依赖的远程目录或登录服务仍持续运营。生命周期信息应随技术结论一起交付。

2. 配置如何减少重复

2.1 配置按模型、规则与 MCP 工具分块；全局配置表达个人通用偏好，工作区配置表达项目约束。历史设计还支持引用块并覆盖属性，复用与本地定制因此不必复制整份助手配置。[C37]

2.2 规则加载保留 source 和 sourceFile，普通规则与可主动调用的提示分开处理；源码也维护就地 rules.md 的来源与缓存。这为解释“某条指令为什么出现”提供基础。[C38][C46]

3. 组合能力需要显式优先级

3.1 本报告建议对组织共享资源采用身份、版本、适用范围和覆盖来源四项元数据。若模型、技能和 MCP 都能全局注入，却没有可见优先级，组合越灵活，冲突排查越困难。

3.2 配置加载成功只是结构合法；仍需检查模型是否支持工具、工具是否连接、规则是否真正命中。针对 Peri，应把配置诊断与执行诊断分开，避免把加载失败误当成模型不配合。

Peri 启发

设计建议：AOS 的组织资源管理可以借鉴配置块思想，但共享不应抹去凭据边界、资源版本和项目自治；用户需要看到最终解析出来的有效配置。

来源 [C36] [C37] [C38] [C46]

Continue：规则激活与模式语义

Continue / 上下文不是全量堆叠

> Continue 通过文件范围和内容条件激活规则，并为 Chat、Plan、Agent 使用不同的提示与工具集合；提示约束和执行保证仍须逐项核对。

层级	表达什么	可观察证据
规则匹配	何时加载某条规范	来源、路径、触发条件
模式提示	当前应采用的工作方式	最终系统消息
工具可见性	模型能请求哪些动作	实际请求的 schema
执行策略	动作是否获准执行	批准 / 拒绝及原因

1. 规则按上下文装配

1.1 规则支持常驻、路径 glob、内容正则等条件。源码从上下文文件路径和内容计算匹配，再把相关规则加入系统消息；就地规则带来源路径，而不只是收集所有 Markdown 一次性拼接。[C39][C45][C46]

1.2 好处是减少无关规范占据上下文；代价是匹配逻辑成为正确性边界。路径规范化、负向 glob、多根工作区和未读取文件都应进入回归，避免规则存在却从未触发。

2. 模式改变真实请求形态

2.1 官方工具握手为模型返回调用、用户批准、内置或 MCP 执行、结果反馈；Chat 不带工具，Plan 选择只读工具，Agent 开放修改能力。Agent 提示同时允许多个只读工具并行。[C40][C41]

2.2 最终源码的 Plan 筛选器仅对内置工具组要求 readonly；其他已启用工具仍可进入请求，先前设置的禁用或排除策略继续生效。提示旁也注明让模型判断 MCP 只读性，因此不能把 Plan 理解为所有外部副作用已被封锁。[C40][C57]

3. 模型上下文与策略各守边界

3.1 本报告建议规则只承担操作指导；会写文件或修改远程数据的能力须由结构化策略识别。一个名叫查询的工具也可能有副作用，不能仅通过自然语言名称自动授权。

3.2 对 Peri 的规则系统，可记录命中原因和未命中原因，在执行审计中保留真正的授权结果。这样既能解释模型为何忽略规范，也能确认没有把“请勿修改”误认为可靠隔离。

证据边界

当前文档同时写“只读”和“支持全部 MCP”；最终源码仅对内置组筛 readonly。禁用策略与审批仍另行生效，不能推导外部 MCP 无副作用或无需确认。[C57][C64]

来源 [C39] [C40] [C41] [C45] [C46] [C57] [C64]

Continue：委派实现的反例价值

Continue / 共享可变服务的风险

> Continue 最终 CLI 的子智能体实现展示了轻量委派的便利，也展示了危险捷径：独立聊天数组不等于独立权限、服务或并发运行域。

实现选择	轻量收益	迁移风险
独立聊天数组	减少父上下文污染	不隔离其他服务
动态专长描述	帮助选择委派目标	过长提示会扩大工具描述
临时通配允许	避免交互卡住	改变原有权限语义
共享服务恢复	减少重构成本	异常和并发需独立验证

1. 工具如何找到专用智能体

1.1 委派工具描述动态列出子模型名称和专用系统提示；调用前校验名称，再查找父会话，等待 executeSubAgent 返回。这里可核实的是一次子调用流程，而非成熟的并行团队系统。[C43][C44]

1.2 执行器创建只有当前任务的内存聊天数组，把基础系统消息与子模型 baseSystemMessage 组合，然后复用 streamChatResponse。它返回末条消息作为结果，父任务不接收全部内部执行历史。[C42]

2. 共享服务为何值得警惕

2.1 源码保存主权限后，把共享容器策略改成通配允许，再构建提示并替换 getSystemMessage 与 ChatHistoryService.isReady；运行流的内层 finally 负责恢复这些共享状态。[C42]

2.2 控制流推导：权限放开后，buildAgentSystemMessage 若在进入内层 try 前抛错，外层 catch 没有对应恢复；并行移植还可能交错覆盖。两者都是待验证的静态风险，本报告未运行复现，也不称为已证实攻击。

3. 可以借鉴与必须重做的部分

3.1 可借鉴的是动态专长目录、明确输入和精简结果；应重做的是每次委派独立的运行依赖、权限上界和取消域。父会话标识只是关联信息，不会自动提供资源隔离。

3.2 对 Peri 的设计建议是把权限和上下文作为子运行构造参数，而不是临时改全局单例。验收要包含子任务失败、初始化异常、取消和多路交错，确认主任务状态始终不被污染。

证据边界

以上共享状态和通配允许来自固定提交源码；尚未执行该版本。此页用于识别设计风险，不构成已复现漏洞公告，也不推断其他版本行为。

来源 [C42] [C43] [C44]

Tabby：检索先于宏大提示

Tabby / 自托管代码上下文

> Tabby 的长处在代码补全与知识上下文基础设施；它提醒我们，很多编码质量问题来自检索、预算和当前代码事实，而非缺少更长的角色提示。

结构示意 · 关系与数据流

节点与职责

1. 编辑器代码现场
2. LSP 客户端代理
3. 声明与近邻片段
4. 可选仓库检索 · 先验允许仓库匹配
5. 预算与模板输入
6. 模型补全

有向关系 · 后半句为条件或动作

- 编辑器代码现场 → LSP 客户端代理 · 位置与文档
- LSP 客户端代理 → 声明与近邻片段 · 采集
- 声明与近邻片段 → 预算与模板输入 · 优先
- 可选仓库检索 → 预算与模板输入 · 预算允许时补充
- 预算与模板输入 → 模型补全 · 模板输入

1. 定位与名称边界

1.1 官方仓库仍名为 Tabby，定位自托管编码助手；README 中出现 Pochi 任务集成，不能据此断言 Tabby 已更名为 Pochi。本章采用 21b29048 快照，不凭提交间隔断言项目停止维护。[C47]

1.2 tabby-agent 是编辑器与服务端间的 Node 语言服务器，支持标准补全并扩展 tabby/* 方法；这个 agent 名称并不自动代表会自主循环调用工具的多智能体执行器。[C48]

2. 让上下文与代码结构对齐

2.1 历史设计用 Tree-sitter 提取符号片段，通过 BM25 检索相关声明和使用方式；随后增加向量相关性与 RRF 融合。最新检索源码仍保留三类分数，避免把历史博客当成未经复核的现状。[C50][C51][C56]

2.2 服务端 PromptBuilder 先消费客户端提供的声明和近邻片段，预算剩余足够才查仓库；还要按允许仓库匹配 git_url。检索结果以目标语言的注释格式注入前缀，再套模型模板。[C52]

3. 学习的是信息分配

3.1 本报告建议先分析失败是否源于缺少正确函数签名、调用约定或最近改动，再决定扩大模型上下文。多智能体重复搜索同一仓库，也需要统一的检索证据和新鲜度。

3.2 检索融合不保证语义正确；过时索引、错误仓库映射和截断声明都会影响结果。面向 Peri，应让命中片段携带路径和版本，并允许智能体回到原文验证。

Peri 启发

设计建议：将 Tabby 作为上下文与交互延迟的参照，而非并行自治团队的直接替代；按实际执行能力分类，才能避免产品名称误导架构判断。

来源 [C47] [C48] [C50] [C51] [C52] [C56]

Tabby：实时交互的调度细节

Tabby / 预算、取消与缓存

> Tabby 把低延迟拆成上下文裁剪、取消过期请求、自适应防抖和可复用补全；这些机制比单纯缩短提示词更接近真实交互瓶颈。

机制	减少的浪费	迁移限制
上下文去重	重复代码与无关输入	不能截断关键语义
请求取消	过期补全的计算与展示	已发生副作用不能撤销
自适应防抖	过密请求及界面抖动	需要真实延迟反馈
前向补全缓存	连续输入重复生成	文档版本必须参与失效

1. 请求前先控制输入量

1.1 buildRequest 裁剪前后缀行数，对声明及最近修改、最近打开片段设数量和长度限制；利用文件路径与范围交叠去重，避免同一段代码通过多个来源重复占据预算。[C49]

1.2 这些限制保留了来源优先级：类型声明、刚改过的内容和当前用户浏览现场各有作用。对于大量工具结果，同样应保留来源与摘要，不能把所有文本无差别串起来。

2. 新输入使旧工作失效

2.1 补全入口会取消上一次活动请求，后续采集和返回步骤继续检查取消信号；上下文采集有超时，结果还经过缓存前后两阶段处理。这避免用户已继续输入时仍展示过期建议。[C53]

2.2 防抖根据近期输入间隔、光标位置和预估响应时间调节等待，手动请求可绕过等待；缓存键包含当前前后缀及其他已打开文档版本，并为连续输入构造可复用剩余补全。[C54][C55]

3. 对流式 Agent 的适用范围

3.1 本报告建议借鉴“明确过期、协同取消、分层预算”的原则。补全可以丢弃旧请求，而写文件或数据库的 Agent 工具不能仅因新输入到来就假定副作用不存在，必须区分可取消阶段。

3.2 Peri 的流式界面与历史恢复也应采用不同调度：活动文本可合并刷新，稳定历史对象应按游标分页；缓存和重放保持对象身份，避免高频输出淹没控制消息。此处是迁移建议，并非 Tabby 已实现 ACP。

Peri 启发

设计建议：面向高频交互分别量测等待、采集、模型、传输与渲染耗时；只有把瓶颈拆开，才能判断应改提示、缓存还是调度。

来源 [C49] [C53] [C54] [C55]

仓库地图是检索预算器

Aider / 结构上下文

> Aider 把代码库压成可排名的定义地图，再让模型索取工作文件；它解决的是有限上下文中的导航，不等于完整理解仓库。

结构示意 · 关系与数据流



1. 从源码生成导航骨架

1.1 tree-sitter 提取定义与引用，记录符号、文件和位置；相较只保留名称，源码树上下文能给出更有用的函数签名。其设计演进有作者文章可追溯。[D03]

1.2 repomap.py 以引用和定义建立文件关系图；同名符号可能连接多个候选文件，因此这是语法线索形成的近似图，不能当成编译器精确解析的调用图。[D02]

2. 预算驱动的相关性选择

2.1 PageRank 个性化权重偏向对话中的文件、提及路径与标识符；引用次数经过弱化，排名再分配给定义。随后搜索能装进 token 目标的标签数量，渲染精选代码树。[D02]

2.2 地图提供文件名和重要定义，模型再请求完整文件进入编辑上下文。官方说明预算会随剩余上下文动态调整；地图并非每轮无差别注入所有文件。[D01]

3. 失效条件与工程取舍

3.1 分析：反射、动态导入、生成代码和同名接口会削弱引用图；排名靠前代表更值得查看，不能证明某处就是缺陷根因。缓存失效策略也应与文件版本同步。

3.2 对 Peri 的建议：将结构地图、全文工作集和原始档案分开计量，记录“地图建议后实际打开”的命中率；命中差时改检索策略，而非直接扩大所有提示词。

设计启发

把地图当作可解释的检索候选生成器；最终修改仍需读到实际文件，并确认符号所在版本。

来源 [D01] [D02] [D03]

编辑协议本身是产品能力

Aider / 提示词与可修复编辑

> Aider 把输出格式、文件可见性和失败反馈写成一套编辑协议；关键不只是生成正确补丁，还要知道哪些修改已经落盘。

协议	收益	常见失效
整文件替换	结构简单、无需定位旧片段	输出长，易覆盖并行改动
SEARCH/REPLACE	输出紧凑、位置可核对	旧内容不匹配或片段重复
差异格式	改动范围便于审阅	格式细节仍会生成错误
逐块反馈	修复目标明确	批次已部分成功时不可盲重放

1. 提示词约束编辑边界

1.1 开源模板先要求识别需改文件；已有文件若未进入对话，应请求路径并取得内容。随后用精确 SEARCH 与替换段表达局部修改，配合少量格式示例约束输出。[D05]

1.2 whole、diff、diff-fenced 和简化统一差异适配不同模型；编辑器专用格式进一步缩小职责。选择格式是“模型生成稳定性 × 输出成本 × 应用可靠性”的联合问题。[D04]

2. 恢复是逐块状态机

2.1 get_edits 先解析本轮编辑列表，apply_edits 再逐块替换；成功就写入，失败记入集合。批次末尾有失败时，成功块仍保留，因此一次响应不是多文件原子事务。[D06]

2.2 实现还容忍部分前导空白差异，并在原文件匹配失败后尝试其他已入聊文件；精确匹配是提示要求，不是唯一执行分支。错误反馈指出相似内容及已成功块，要求仅修复失败项，不能盲目重放整批。[D06]

3. 值得迁移的实现原则

3.1 分析：文本匹配适合局部代码编辑，遇到重复片段、格式器改写或并发写入时会变脆。重试应重新读取相关区间，不能仅让模型换一种措辞再试。

3.2 对 Peri 的建议：每个编辑记录预期文件版本、匹配结果、应用后摘要与失败原因；若需要整批原子性，应增加事务层和回滚记录，不能从提示词中推导。

边界

编辑约束能提高可执行率，但文件访问权限、并发写锁与事务提交仍应由运行时落实。

来源 [D04] [D05] [D06]

Architect 是串行职责分离

Aider / 思考与编辑的交接

> Architect 让方案生成与代码落盘由两次调用承担，降低格式负担；它是串行角色分工，不能据此声称拥有并行多代理协作。

结构示意 · 关系与数据流



1. 一个明确的交接点

1.1 architect 模式先讨论解决方案，再把方案交给编辑模型。作者提出的动机是：推理能力与稳定输出编辑语法未必集中在同一模型，也可使用同一模型承担两种角色。[D07][D09]

1.2 reply_completed 收到方案后进入可配置的确认分支，选择 editor_model 或主模型；创建编辑 coder，使用编辑格式，将方案作为新输入执行。[D08]

2. 源码揭示了上下文取舍

2.1 编辑阶段关闭仓库地图、清空新的当前及历史消息，并继承原 coder 的工作环境。它并不是把两个完整会话无限叠加，而是围绕交接文本重建编辑任务。[D08]

2.2 分析：窄上下文减少了无关讨论与格式冲突，但未写入交接的约束也可能丢失；如果方案遗漏隐含不变量，编辑器没有天然保证会主动恢复全部推理。

3. 怎样判断分工是否划算

3.1 分析：应分别记录方案错误、编辑格式失败和测试失败。第二次调用增加时延与费用；只在格式失败确实拖累任务时，角色分离才可能带来净收益，历史模型成绩不是普遍定律。

3.2 对 Peri 的建议：交接协议至少包含目标文件、必须保留的行为、允许改动范围、验收命令与未决问题。把“实现者”与“独立验证者”分开建模，避免把第二次调用自动算作评审。

分类结论

这是方案→编辑的串行流水线；共享同一任务目标不等于多个自治代理同时协商或分配资源。

来源 [D07] [D08] [D09]

Git 检查点不代表验收

Aider / 反馈、提交与回滚

> Aider 把编辑、lint、测试和 Git 操作连成反馈回路，便于人类审阅与撤销；每个环节的成功信号仍需要单独验证和解释。

信号	能够说明	不能推出
编辑成功	补丁已写入目标文件	行为正确
lint 成功	覆盖规则未报错	测试通过
测试成功	执行的用例通过	未运行场景安全
Git 提交	形成可引用快照	hooks 或 CI 已通过
Git 撤销	代码回到某个状态	外部副作用已恢复

1. 自动化反馈来自命令结果

1.1 编辑后可自动 lint；测试可由 /test 或配置命令触发，并将失败信息交给模型修复。非零退出码与输出是接口核心，命令若错误编码成功状态，后续判断也会偏离。[D11]

1.2 例如格式器“发现并修正格式”也可能返回非零；若直接当作语义失败，代理会陷入重复修复。应根据工具真实契约包装返回码、修改列表与剩余错误。[D11]

2. Git 保存的是代码检查点

2.1 Git 集成支持自动提交、查看差异和撤销；处理已有脏文件时，可先分离用户原有改动。这样能提高归因清晰度，但不能证明提交的代码通过了业务验收。[D10]

2.2 官方文档明确默认提交会跳过 Git hooks，另有开启验证的选项。因此“已经 commit”不能被 UI 或调度器翻译成“通过 pre-commit / CI”。[D10]

3. 失败恢复必须分清对象

3.1 分析：撤销仓库差异无法撤销测试访问的外部服务、数据库写入或发布行为。命令执行前后的副作用与代码文件应使用不同日志和补偿策略。

3.2 对 Peri 的建议：为 AOS 长任务分别维护修改检查点、测试证据与外部动作收据；恢复时询问“回到哪个代码状态、哪些验收仍有效”，而不是只显示一个成功标志。

实现建议

文件回滚、进程终止、测试复现与外部补偿属于四种不同能力；统一按钮可以统一入口，但不应抹平语义。

来源 [D10] [D11]

把上下文与评测分别记账

Aider / 摘要与双轮基准

> Aider 的地图、对话摘要和编辑评测解决不同问题；把 token 降低、任务通过和恢复能力混成一个指标，会掩盖真正瓶颈。

层次	保留什么	应测量什么
仓库地图	重要定义与关系	相关文件命中率
当前工作集	实际文件内容	版本一致性
对话摘要	任务状态与决策	约束遗漏率
原始轨迹	可追查的完整记录	归档与内存成本
编辑评测	固定题目与反馈机会	首轮/累计通过率

1. 摘要降低模型输入体积

1.1 history.py 按消息 token 计数；超出预算时偏向保留近期尾部，将较早内容摘要，并处理对话边界。预算仍不满足时继续压缩，深度受限后转向整体摘要。[D12]

1.2 分析：摘要是有损表示，不能确保文件版本、异常细节或被否决方案全部保留；减少发送给模型的 token，也不能自动证明 UI 或内存已释放原始记录。

2. 读懂基准的分母与机会数

2.1 Aider 基准先让模型实施，再提供失败测试进行第二次修复；两次之后的累计通过率并非首轮通过率。适合检验“代码编辑+测试反馈”的闭环，但不是无人值守长任务的完整替代。[D13]

2.2 Polyglot 页面披露任务集合、编辑格式、模型配置、提交/日期及格式错误等字段；这些控制变量比脱离配置的单一排行榜位置更适合复现实验。[D14]

3. 面向长任务的最小仪表盘

3.1 对 Peri 的建议：分别度量结构检索命中、每轮输入、摘要后约束遗漏、编辑可应用率与测试通过率；同时标记首轮结果和重试结果，才能定位收益来自模型还是运行时。

3.2 进一步建议：为长会话记录活跃视图大小、归档字节数和进程常驻内存；用同一真实任务回放检查摘要质量，而不是只用更短的字符串证明压缩成功。

证据边界

本报告读取实现和评测方法，没有重跑这些基准；不据此提供当前模型排行或跨产品统一胜率。

来源 [D12] [D13] [D14]

运行时围绕事件组织

OpenHands / 当前 SDK 架构

> OpenHands 当前 SDK 将推理控制器、会话状态、工具与运行环境拆开；可恢复性的关键是先处理已有动作，再继续生成下一步。

结构示意 · 反馈循环

节点与职责

1. 会话状态 · 事件与待办
2. 处理待办 · 确认与执行决策
3. 模型视图 · 可产生压缩事件
4. 模型调用 · 生成工具请求
5. 动作事件 · 先提交会话回调
6. 工具执行 · 产生观察
7. 追加观察 / 压缩 · 更新事件状态

有向关系 · 后半句为条件或动作

- 会话状态 → 处理待办 · 有待办
- 会话状态 → 模型视图 · 无待办
- 模型视图 → 模型调用 · 视图可用
- 模型视图 → 追加观察 / 压缩 · 需压缩
- 模型调用 → 动作事件 · 记录动作
- 动作事件 → 处理待办 · 确认策略
- 处理待办 → 工具执行 · 允许执行
- 工具执行 → 追加观察 / 压缩 · 观察
- 追加观察 / 压缩 → 会话状态 · 投影

1. 分层让同一控制逻辑迁移

1.1 官方把 core SDK、tools、workspace、agent-server 分为四层包；本地可在同一进程运行，远程通过 Agent Server 的接口控制。工具与代理在运行环境中执行，并非每次都经 workspace 转发。[D15]

1.2 Agent 是推理与动作控制器，可变状态属于 Conversation；EventLog 留下追加式记录，Local/RemoteConversation 提供相近调用入口。接口统一降低迁移成本，但并不统一底层隔离强度。[D18][D48]

2. 一步不总对应一次模型调用

2.1 当前 agent.py 先检查活跃分支中待完成动作；存在时执行并返回。否则准备模型消息；若得到压缩事件，同样先写事件并返回，下一步才继续采样。[D16]

2.2 _get_action_event 先将动作提交会话回调，再进入工具执行；Observation 通过标识配对，同轮动作按 llm_response_id 组合为模型消息。追加事件、模型视图与界面展示是不同层次，不能把投影重放当成副作用重执行。[D16][D17]

3. 给 Peri 的控制流建议

3.1 分析：恢复后若跳过未完成动作直接再次采样，模型可能重复请求同一操作；若盲目重放所有历史事件，也可能重新执行已经发生的副作用。

3.2 建议把待确认、已批准未执行、执行中、已观察明确编码，恢复时依据动作收据与状态转换决定下一步。ACP 客户端只订阅投影，不应成为执行事实的唯一存储。

边界

事件可重放不代表动作可以重执行；安全恢复依赖副作用收据、幂等键或显式人工处理。

来源 [D15] [D16] [D17] [D18] [D48]

提示词成为可组合构件

OpenHands / Prompt、Skill 与工具契约

> OpenHands 把角色、工具指导、环境与动态技能拆成提示章节，并用类型化工具贯穿执行；提示结构化与权限强制仍是两件事。

构件	主要职责	强制边界在哪里
静态章节	稳定角色与行为要求	仍是模型输入
动态技能	按任务提供知识或步骤	加载器及执行策略
Action schema	校验工具参数结构	类型/字段验证
工具注解	描述预期副作用	不自动授权
执行器与策略	执行并检查能力范围	运行时权限与环境

1. 从模板文件到章节注册表

1.1 当前 AgentBase 对旧 system_prompt.j2 等名称保留兼容映射，再通过 registry 构建 static/dynamic 两部分；自定义模板仍可使用，因此只研究旧模板文件会错过当前默认路径。[D43]

1.2 presets.py 显式注册角色、效率、文件、质量、版本控制、工具和环境章节；动态部分承载仓库、记忆、技能等，时间放在末尾，帮助稳定前缀。规划预设是不同组合，不天然拥有只读权限。[D21]

2. 工具与技能进入不同接口

2.1 工具由 Action schema 验证输入，Executor 执行业务逻辑，Observation 转为模型可读结果，ToolDefinition 连接三者。结构错误能在执行边界暴露，错误输出不必伪装成普通文本。[D19]

2.2 技能可常驻、按关键词或路径注入；动态技能还可执行内联 shell。ACP 场景的规则归属有单独约定，因此技能加载器既涉及上下文，也可能涉及代码执行和重复注入控制。[D20]

3. 组合之后还要检查冲突

3.1 分析：模块化让章节可测试，却不会自动解决互相矛盾的要求。需要明确优先级、来源、适用范围和最终渲染结果；工具的只读/破坏性注解属于提示，不能替代运行时授权。

3.2 对 Peri 的建议：调试界面提供“本轮生效提示清单”，记录来源与版本；让权限策略独立验证动作，避免代理因为某段技能文本而扩大能力。

实现启发

提示词应像配置组合一样可追踪：能看到哪一章被选中、什么触发了注入，以及最终哪些工具真正可执行。

来源 [D19] [D20] [D21] [D43]

压缩视图不等于删除历史

OpenHands / Condenser 与长会话

> OpenHands 让压缩成为可记录的状态转换，原始事件与模型视图各司其职；这为恢复提供基础，也提醒我们单独治理内存引用。

结构示意 · 职责分层

节点与职责

1. 原始事件日志 · 文件 + 物化对象缓存
2. 压缩事件 · 隐藏集合与摘要
3. 缓存投影 · 按规则维护
4. 模型视图 · 供下一轮使用

有向关系 · 后半句为条件或动作

- 原始事件日志 → 缓存投影 · 读取事件
- 压缩事件 → 缓存投影 · 改变投影
- 缓存投影 → 模型视图 · 提供输入
- 模型视图 → 压缩事件 · 需要压缩

1. 压缩是一种事件

1.1 Condenser 可保留初始约束与近期事件，把中段换为摘要；Condensation 指出被隐藏的事件 ID、摘要及插入位置。后续模型看到压缩后的视图，原始事件仍可用于审计。[D22]

1.2 当前 agent.py 使用 state.view 的缓存投影，而非每一步无条件重建全部历史；同时先解析模型容量信息，避免用错误预算触发压缩。文档示意流程不能替代这条实际代码路径。[D16]

2. 失败语义决定能否继续

2.1 base.py 要求输入 View 只读，因为直接改动会污染会话缓存投影；输出应是新的视图或压缩事件。软需求允许未压缩继续，硬需求无法满足时应显式失败。[D23]

2.2 agent-server 恢复 stale RUNNING 时转 ERROR，扫全日志并在必要时给首个未匹配动作补挂到原动作的中断错误，处理事件已写而 HEAD 未存的窗口。它与通用 Agent 的 pending 执行不同，不是全部工具无条件重放；外部效果仍待对账。[D49][D52]

3. 对大 transcript 的工程启示

3.1 EventLog 用普通字典缓存物化 Event，无容量驱逐；底层文本 LRU 不限制该层。冷恢复仍遍历活动分支重建视图，压缩投影较小不证明只读窗口或内存有界；未测 RSS。[D49][D50][D51]

3.2 对 ACP/cursor 场景的建议：增量游标应绑定投影版本；压缩导致视图变化时发出可识别的重置或变更事件。另行测量 Rust 共享引用、缓存、前端副本和后台任务持有的内存。

边界

设计建议：把事件文件、活动 HEAD、模型投影与缓存驻留分别观测；单文件原子写不等于跨文件事务。恢复需对账在途动作，不能根据缺少结果便断定外部操作未发生。

来源 [D16] [D22] [D23] [D49] [D50] [D51] [D52]

运行位置与安全策略分离

OpenHands / 环境、确认与副作用

> OpenHands 把运行环境和确认策略设计为可组合能力；远程调用、工作目录和风险评分都不能单独证明一个任务已被安全隔离。

机制	回答的问题	不足以说明
工作目录	从哪里读写相对路径	无法访问目录外
远程服务	请求由哪里执行	环境已隔离
容器配置	进程/挂载等边界	网络和凭据自动受限
确认策略	动作是否需用户确认	动作本身无风险
风险分析	如何估计危险程度	能可靠解析所有命令

1. 三条边界回答三类问题

1.1 workspace 描述代理在哪运行，Agent Server 提供远程控制入口；工具通常在代理所在环境执行。是否使用容器、挂载什么目录、能否联网，仍取决于实际部署。[D15]

1.2 会话可配置确认策略与风险分析器，决定哪些动作等待确认。官方说明模式分析并非完整 shell 解析，也不能替代注入防护或沙箱；评分只是策略输入之一。[D24]

2. 结构化观察不消除外部风险

2.1 事件与工具 schema 可以准确记录“请求了什么、返回了什么”，但无法仅靠这些数据强迫外部服务幂等。模型生成命令时，权限检查仍需作用于真实执行参数。[D17][D19]

2.2 分析：只读意图可能藏在可写工具内，命令串也可能包含多步操作。生产系统应优先提供范围明确的能力接口；若必须开放 shell，要以部署策略约束文件、网络与凭据。

3. 可复用的权限模型

3.1 对 Peri 的建议：将工作区标识、可写路径、网络目的地、凭据和动作风险分别建模。子代理得到的是能力集合，不是因为“角色可信”而自动继承所有能力。

3.2 分析：验证隔离必须检查操作系统和部署配置，不能只检查 UI 标签。审计事件还应关联实际执行环境、策略版本与批准者，否则恢复后无法解释旧决定。

边界

本页是部署与权限的设计分析；未对 OpenHands 各部署方式做渗透测试，也不把默认配置宣称为统一安全保证。

来源 [D15] [D17] [D19] [D24]

子代理工具带来新的资源账本

OpenHands / TaskTool 的真实边界

> 当前 TaskTool 支持创建与恢复子会话，并管理预算和释放对象；子代理默认共享父工作目录，协作正确性必须另有写入协议。

管理对象	源码可见能力	建议补充的协作约束
子会话	独立消息与任务 ID	输入/输出契约
文件系统	继承父 working_dir	排他写入或独立工作区
运行预算	按定义或父会话设置	全局预算与公平调度
恢复与内存	子会话落盘、关闭、同管理器续接	跨进程任务映射重建
成本指标	按任务覆盖累计值	统一总账与重试标识

1. 委派规则写进工具入口

1.1 TaskAction 接收任务、类型及恢复 ID；工具描述鼓励把代码理解或测试等完整任务交给子代理，同时反对为单次 grep 付出调度开销。文档说明调用同步等待子任务结果。[D25][D44]

1.2 当前 schema 的 max_turns 已弃用且被忽略；新建任务的迭代上限取子代理定义或父会话。接入应读实际 schema 与创建路径，不能只照搬旧字段名。[D44][D45]

2. 上下文分离不等于文件隔离

2.1 manager.py 创建与续接 LocalConversation 均使用父 working_dir。子会话数据落盘后可关闭并释放对象，累计指标按任务覆盖更新；但 task_id 到 conversation_id 的 _tasks 映射仍保存在该管理器内存。[D45]

2.2 分析：这支持同一管理器内的低驻留续接，不能仅凭子会话落盘推出跨进程任务注册表已自动恢复。多个任务写同一目录仍会冲突，子会话名称也不能证明进程、环境变量或外部状态被隔离。

3. 怎样向长期任务系统迁移

3.1 对 Peri 的建议：委派请求明确输入文件、输出产物、验收标准和写入范围；使用独立工作区或排他写入权，再让父代理合并证据。仅增加“请多派代理”的提示不足以解决冲突。

3.2 版本辨析：SDK 论文描述过以单代理为主的局限；当前 TaskTool 已提供实质子任务能力。论文适合解释历史设计，当前功能应以文档与固定源码交叉确认。[D26]

协作边界

子代理返回结论仍应附可检查证据。委派、执行、合并和验收是四个阶段，不能把“任务已返回”当作整体完成。

来源 [D25] [D26] [D44] [D45]

ACI 让工具适应模型

SWE-agent / 接口设计的实证价值

> SWE-agent 的经典贡献是把搜索、查看和编辑重新设计为模型易用的计算机接口；历史消融说明接口质量可以显著影响任务完成。

设计杠杆	作用机制	需要观测的代价
搜索摘要	先定位文件再读内容	相关结果是否被截断
代码窗口	减少无关上下文	跨区间信息是否遗漏
编辑即反馈	立即确认作用位置	观察文本与 token 成本
语法保护	提前阻断明显坏编辑	中间状态是否被误拒绝

1. 缩小每次观察和动作

1.1 经典 ACI 用文件名摘要限制搜索膨胀，用固定窗口查看代码，并在编辑后立即展示相关内容；没有输出的成功命令也给出明确反馈，减少模型猜测“是否执行”。[D27]

1.2 原论文限制过宽搜索并要求收窄；编辑配合语法检查，失败修改可被拒绝。这让工具承担部分状态校验，而非把所有恢复负担交给模型自行发现。[D28]

2. 消融比产品口号更有解释力

2.1 原论文在 GPT-4 Turbo 与 SWE-bench Lite 的指定实验中，完整 ACI 为 18.0%，移除专用编辑为 10.3%，移除 lint 为 15.0%；这些是历史实验条件下的解决率，不是当前排名。[D28]

2.2 分析：结果支持“接口会影响效果”，却不能推出每个新模型都需要相同窗口大小。搜索截断能节省注意力，也可能隐藏真正目标；编辑保护也可能拒绝有意进行中的中间状态。

3. 从经典设计到当前配置

3.1 当前官网标注 SWE-agent 进入维护状态并指向 mini-SWE-agent；当前 default.yaml 使用函数调用和另一套工具组合。不能把论文中的每个 ACI 细节都说成现行默认行为。[D27][D29]

3.2 对 Peri 的建议：围绕真实失败建立接口实验，分别测试返回窗口、输出截断、唯一匹配和即时反馈；固定模型及任务，避免把提示词变化与工具变化同时混在一次比较中。

证据边界

这里引用历史受控实验说明机制；本报告没有复现实验，也没有把旧数据折算为当下产品强弱。

来源 [D27] [D28] [D29]

环境状态与上下文状态分开

SWE-agent / 工具、历史与执行层

> SWE-agent 将环境执行、提示模板与历史处理分层配置；同一条轨迹同时包含 shell 状态、文件状态和模型看到的历史，三者不能混淆。

结构示意 · 职责分层

节点与职责

1. 任务模板 · 目标与约束
2. 历史处理 · 模型可见消息
3. 模型与动作 · 结构化工具请求
4. 工具定义 · 输入和编辑规则
5. SWE-ReX 环境 · 进程与持久 shell
6. 文件与观察 · 实际执行结果

有向关系 · 后半句为条件或动作

- 任务模板 → 模型与动作 · 输入
- 历史处理 → 模型与动作 · 转换
- 模型与动作 → 工具定义 · 调用
- 工具定义 → SWE-ReX 环境 · 执行
- SWE-ReX 环境 → 文件与观察 · 更新
- 文件与观察 → 历史处理 · 反馈

1. 工具接口限制编辑歧义

1.1 SWEEnv 通过 SWE-ReX 管理执行环境、工具安装和持久 shell；与每次新 shell 的设计相比，目录或环境状态可跨命令存在，恢复与调试需要理解这些隐藏状态。[D30]

1.2 当前编辑工具 schema 提供查看、新建、替换、插入和撤销；替换要求旧文本唯一匹配，避免猜测目标位置。undo_edit 针对单文件最近编辑，不是整个环境事务回滚。[D34]

2. 历史处理是显式策略

2.1 history_processors.py 提供保留近期观察、标签控制和缓存控制等处理器；压缩旧观察能减少输入，但逐轮改写消息前缀也可能影响缓存。具备一个处理器不代表默认就启用它。[D31]

2.2 当前 default.yaml 配置的是 cache_control 历史处理，而非直接启用经典论文的旧观察折叠方案。配置还关闭分页器等交互行为，减少命令等待和输出不可控。[D29]

3. 模板必须匹配产品任务

3.1 当前默认任务模板带有基准特有的“不改测试”和清理复现脚本等约束。分析：将它原样移植到真实开发助手，会妨碍补回归测试；提示词必须按验收目标重新审查。[D29]

3.2 对 Peri 的建议：为提示模板、工具 schema、历史处理和环境配置分别留版本；失败回放固定这四项。把环境状态摘要与模型消息摘要分开，避免恢复后目录或依赖状态悄悄漂移。

实现启发

“同一个模型”不是可复现实验的充分条件；配置、环境镜像、工具定义和历史转换共同决定实际接口。

来源 [D29] [D30] [D31] [D34]

多次尝试不是多代理协商

SWE-agent / 轨迹、候选与评测

> SWE-agent 的多尝试配置通过生成候选再选择提高成功机会；它与分工协作不同，评测必须把尝试预算和最终验收一起公开。

阶段	产物	不能替代
单次执行	轨迹与候选补丁	完整任务验收
重复尝试	多个候选及成本	代理间分工协商
选择器	选中的候选与理由	测试正确性
独立评测	固定环境下的结果	部署后的全部行为
清理回收	环境终止记录	代码回滚

1. 轨迹让故障可以定位

1.1 轨迹文件可保存响应、动作、观察、状态和实际模型查询，旁边保留配置、补丁及预测输出。分析：这些信息能区分“模型没想到”“工具执行失败”和“评测环境不一致”。[D32]

1.2 仅保存最终 diff 无法解释被截断的搜索、失败的编辑或错误重试；只保存自然语言总结也难以重放。长任务档案应关联任务 ID、代码版本与环境标识。

2. 候选选择改变了统计对象

2.1 官方 competitive runs 描述多个尝试与选择器；固定 heavy 配置使用不同代理配置、retry/chooser，最多十次尝试并设预算限制。这是特定实验配置，不能当作所有运行的默认值。[D33][D46]

2.2 分析：独立候选增加覆盖面，但选择器可能偏好表面合理的补丁；它不等价于多个代理共享证据并分工解决子问题，更不代表通过了独立隐藏测试。

3. 面向真实交付的验收链

3.1 对 Peri 的建议：候选各自保存工作区和轨迹，用统一测试门禁比较；选择器说明选择理由，最终交付同时记录总调用、总时长和全部尝试成本。失败候选也要可追查。

3.2 分析：实验结束还包括容器、进程和临时工作区清理；官方提醒异常中止可能留下环境。调度器应将资源回收设为持久任务，不依赖代理临时记住执行清理。[D33]

评测建议

报告首轮通过、候选覆盖、选择成功和最终验收四个指标；把它们合成一个成功率会隐藏选择器和预算的影响。

来源 [D32] [D33] [D46]

极简循环也需要版本辨析

mini-SWE-agent / v2 的真实结构

> mini-SWE-agent 把代理缩为查询、执行与记账的循环，并让模型适配器处理协议；当前 v2 已采用原生工具调用，不能沿用旧宣传描述。

层次	v2 职责	接口风险
代理	循环、预算、保存	结束条件与异常处理
模型适配器	调用、动作解析、观察格式	供应商消息差异
环境	执行命令并返回结果	状态和隔离语义
配置模板	任务与工具使用指导	旧约束与任务不匹配

1. 先消除文档冲突

1.1 FAQ 仍出现“不使用工具调用”的旧描述，而 v2 迁移文档明确默认采用原生 Bash 工具调用；固定源码也提供 BASH_TOOL。功能判断应以新版迁移说明和实际入口为准。[D35][D37][D42]

1.2 v2 将动作解析与观察格式化交给模型适配器，适应不同 API 的消息形状。代理不需要把所有供应商协议硬编码进循环，代价是模型适配器承担更多正确性责任。[D35]

2. 循环简单，但责任清楚

2.1 DefaultAgent 用扁平列表保存消息：查询模型、记录响应、执行动作、追加观察。执行同一响应的动作使用顺序列表遍历；此处没有由基础循环提供的并行子代理调度。[D36]

2.2 LiteLLM 适配器保留原生消息，并在 extra 中附加动作、原始响应和成本等元数据；发给模型前剥离非协议字段。上下文溢出属于需要中止处理的错误，而非默认自动摘要。[D37]

3. 极简架构适合做什么

3.1 分析：少层次让研究者更容易控制变量，替换模型或环境后能直接观察轨迹变化。但基础路径没有复杂记忆与调度，不应由“小代码量”推导长任务可靠性天然更高。

3.2 对 Peri 的建议：保留一个足够简单的参考执行器作为回归基线；新增协作、记忆或缓存后，用相同任务确认收益和额外失败模式。极简实现适合做对照组，不必成为所有产品的唯一形态。

版本结论

本页结论绑定 2026-10-18 读取的固定提交；“无工具调用”只作为旧文档冲突记录，不作为当前特征。

来源 [D35] [D36] [D37] [D42]

独立 Shell 不是沙箱

mini-SWE-agent / 执行与完成协议

> mini-SWE-agent 每次使用独立 shell，减少会话隐含状态；文件系统和外部副作用仍会持续，完成标记也需要与普通输出仔细区分。

状态	独立 shell 后是否保留	应由谁治理
当前目录/局部变量	通常不跨命令保留	命令显式参数
文件修改	继续保留	工作区与版本控制
子进程	取决于进程生命周期	超时与任务管理
网络/外部写入	效果可能持续	权限与补偿流程
完成意图	从输出协议解释	结构化控制接口

1. 消除一类状态，保留另一类

1.1 提示模板明确每次命令在独立 shell 执行，目录切换应写入同一命令；环境还可替换为本地或容器等实现。统一接口不意味着这些环境具备同样的权限边界。[D38][D40]

1.2 当前 LocalEnvironment 用 Popen 执行，合并标准输出与错误；继承系统环境并叠加配置。POSIX 上超时终止进程组并收集输出，返回可识别的异常信息，便于模型判断重试。[D39]

2. 输出既是证据，也承担控制

2.1 默认模板对过长观察保留头尾并提示省略字符数；这减少上下文占用，却可能丢掉中段关键错误。分析：生产环境最好保留完整输出引用，允许按需读取具体区间。[D38]

2.2 本地实现用成功退出及首个有效输出行的特定完成标记识别结束。分析：控制与普通 stdout 共用通道，存在误判碰撞的可能；不能假定任何出现标记的文本都是可靠意图。[D39]

3. 给运行时的设计选择

3.1 对 Peri 的建议：明确区分命令进程状态、工作区文件状态和外部系统状态；独立 shell 只减少目录/局部变量沿用，并不提供文件快照、网络隔离或撤销。

3.2 进一步建议：用结构化 finish 动作表达完成，要求验收证据字段；观察输出作为不可信数据保存。超时策略还需定义后台任务归属，避免任务结束后仍有无人管理的进程。

边界

本报告指出的是接口层面的可预见失效模式，没有声称已复现完成标记攻击或各环境的隔离漏洞。

来源 [D38] [D39] [D40]

研究基线也要可恢复与可评测

mini-SWE-agent / 预算、轨迹和批量运行

> mini-SWE-agent 用明确预算和轨迹支持实验，批量 runner 并行的是独立题目；生成预测文件只是评测输入，不代表问题已解决。

记录	最小字段	解释的风险
模型调用	配置、请求、费用、错误	预算与格式恢复
动作执行	命令、环境、返回码	环境或工具失败
候选提交	补丁、退出原因、实例 ID	生成与验收混淆
独立评测	测试结果、版本、环境	评分条件漂移
批量调度	并发数、资源、清理状态	吞吐与可靠性混淆

1. 把失败当成可记录结果

1.1 DefaultAgent 在下一次查询前检查成本、步骤与时间限额；格式错误也可能已经产生模型费用，因此会记账并限制连续错误。配置输出路径时，循环通过 finally 保存状态。[D36]

1.2 分析：调用前检查不能保证一次昂贵调用绝不越过预算，且保存轨迹不等于恢复所有进程。成本上限、模型超时和环境期限应分别处理，不能只依赖总轮数。

2. 并行单位是基准实例

2.1 runner 使用线程池分发不同实例，每题独立输出；重跑前移除旧预测，保存轨迹后更新预测文件，并依据已有预测决定跳过已完成实例。这是任务并行，不是同题内部的代理团队。[D47]

2.2 官方流程将补丁预测生成与独立 SWE-bench 评测命令分开。代理退出、预测文件存在和隐藏测试通过是三种状态；不能把“跑完全部实例”描述为“解决全部问题”。[D41]

3. 构建可信的比较基线

3.1 对 Peri 的建议：为每次运行固定代码版本、模型配置、任务集合和环境，保留完整失败原因；与极简基线比较时纳入总耗时、总费用及首次成功时间，不只比较最终补丁。

3.2 分析：多实例并行可提高吞吐，却增加宿主资源竞争；隔离粒度、并发数与环境准备时间都影响结果。测试重跑应由具体失败假设驱动，避免无限加尝试直到得到好看的数字。

综合结论

Aider 强在编辑协议，OpenHands 强在运行时组合，SWE-agent 强在 ACI 实证，mini 强在可控基线；迁移价值取决于你的故障类型。

来源 [D36] [D41] [D47]

规则是有作用域的上下文

Cursor | 提示词装配与可观测性

> Cursor 值得借鉴的是规则的选择、来源和上下文成本可以被解释；持续指令、单次技能与历史摘要各有明确生命周期。

载体	激活方式	主要代价
常驻规则	每轮提供	稳定成本持续累积
路径规则	命中目标文件	错误 glob 会静默漏用
描述规则	模型判断相关	描述质量影响召回
模式技能	进入后持续提供	须有明确退出语义

1. 规则先被选择，再进入模型

1.1 项目规则放在 .cursor/rules/*.mdc，以 alwaysApply、globs、description 决定常驻、文件匹配、模型选择或手动激活；裸 .md 不是该目录的有效规则格式。[E01]

1.2 团队、项目、用户规则存在优先顺序；嵌套 AGENTS.md 提供更细目录约束。规则中的 @文件只是定位线索，内容不会自动展开，模型仍需工具读取。[E01]

2. 长任务的指令生命周期

2.1 单次 Skill 调用会随对话推进淡出；Custom Mode 则让相应 Skill 在退出模式前持续留在上下文。界面还分别展示系统、工具、规则、技能、MCP、子智能体及摘要的 token 占用。[E02]

2.2 研究判断：这使“规则未触发”“规则触发但引用未读”“工具目录过大”成为不同故障。只追加一条更强硬的提示，无法定位这些问题。

3. 对 Peri 的可执行启发

3.1 建议保存每轮规则装配清单：来源版本、触发理由、作用路径、载入方式、token 成本。客户端应能解释某条规则为何缺席，而不必让用户猜测模型意图。

3.2 建议把长期原则压缩成短约束，详细范例按需读取；若每条规则都常驻，工具与领域信息会竞争同一上下文预算。

边界

此处描述公开配置与可见行为；Cursor 私有系统提示、内部提示拼接和检索算法并未因此被公开。规则也不构成执行权限。

来源 [E01] [E02]

子智能体首先隔离上下文

Cursor | 委派接口与工作副本

> 把搜索和浏览器噪声留在子上下文，可以保护主任务；但上下文隔离、工作树隔离和虚拟机隔离必须分别表达。

结构示意 · 关系与数据流

节点与职责

1. 父任务 · 目标与验收
2. 探索子任务 · 独立上下文
3. 验证子任务 · 独立证据
4. 工作副本 · 请求隔离后分配
5. 结果汇合 · 摘要与证据

有向关系 · 后半句为条件或动作

- 父任务 → 探索子任务 · 只读委派
- 父任务 → 验证子任务 · 独立复核
- 父任务 → 工作副本 · 可选隔离
- 探索子任务 → 结果汇合 · 结论
- 验证子任务 → 结果汇合 · 验证
- 工作副本 → 结果汇合 · 变更

1. 委派消息就是任务接口

1.1 子智能体从干净上下文启动，父智能体负责提供必要背景，结束后返回结果；前台等待结果，后台立即返回。Explore 和 Browser 针对高噪声中间输出提供专门执行链。[E03]

1.2 自定义配置位于 .cursor/agents/ 等目录，描述用于委派选择，提示用于执行；可恢复的 agent ID 则让后续调用继续先前上下文。云子智能体使用团队云 MCP 配置，不能假定继承本地连接。[E03]

2. 并行写入需要另一个边界

2.1 子代理默认共享父 checkout，并发编辑可能相互覆盖；请求隔离才各有分支与 worktree 或云环境，由父任务合并。工作树仍不隔离同机端口和外部状态，不能将可选隔离写成默认能力。[E03]

2.2 研究判断：委派协议应同时携带目标、允许修改范围、基线提交、预期证据与停止条件。只有角色名而没有输出契约，会让父任务花更多上下文重新解释子任务。

3. 对 Peri 的验证设计

3.1 建议对长搜索、浏览器调试、独立复核分别测量父上下文增长、结果遗漏、总 token 与端到端耗时；不能只比较一次子调用的延迟。

3.2 /best-of-n 只比较候选，不自动合并；选中后另行提交或 /apply-worktree。setup 也影响隔离，官方不推荐依赖软链接回主工作树。建议分记候选、应用结果和验证版本。[E43]

本报告示例

委派提示应写成：“只审计恢复链路；基于提交 X；输出可复现路径、证据、严重度和反证；不要改动代码。”这是设计示例，并非 Cursor 私有提示词。

来源 [E03] [E43]

可运行环境决定闭环能力

Cursor | 云环境准备、启动与交付

> 云智能体的关键工程不是把聊天搬到服务器，而是提前准备可验证环境，并把磁盘状态、运行进程和网络权限分开管理。

结构示意 · 关系与数据流

节点与职责

1. 环境定义 · 安装脚本与基础镜像
2. 成功 Build · 保存磁盘
3. 新实例 · 恢复副本
4. 服务启动 · start / terminals
5. 任务验证 · 测试与浏览器
6. 失败 Build · 保留旧版本

有向关系 · 后半句为条件或动作

- 环境定义 → 成功 Build · 构建成功
- 环境定义 → 失败 Build · 构建失败
- 成功 Build → 新实例 · 按需恢复
- 新实例 → 服务启动 · 启动进程
- 服务启动 → 任务验证 · 启动后执行
- 失败 Build → 成功 Build · 旧版本继续可用

1. 把冷启动准备移出任务路径

1.1 Cloud Agents 在独立 VM 和分支工作，能够测试、操作浏览器并交付变更；当前产品也支持多仓环境。并行会话的执行能力来自完整开发环境，而不只是多个模型请求。[E04]

1.2 Build 提前执行 install 并保存磁盘状态；运行中的进程不会进入 Build。启动后另跑 start、terminals。install 必须幂等，失败 Build 不替换最近成功环境。[E05]

2. “快照就绪”与“服务可用”分离

2.1 研究判断：依赖已安装不等于数据库、隧道和开发服务器已启动。环境健康检查应该分别验证工具链、依赖版本、服务端口和最短测试路径，不能把安装成功直接标成任务可执行。

2.2 托管云环境默认能访问互联网，可配置出站限制；自托管机器使用操作者自己的网络控制。迁移执行位置时，网络能力不会自动获得相同保护。[E07]

3. 对 AOS 工作空间的启发

3.1 建议把环境定义版本、成功构建标识、启动检查和任务分支记录到交付证据中。同一任务失败时，才能区分环境漂移与代码回归。

3.2 建议按“准备可复用磁盘 / 恢复实例 / 启动服务 / 执行验收”计时，分别优化。只测模型生成速度，会掩盖大量环境准备损耗。

边界

本地 Agent 的命令允许列表与 Auto-review 被官方描述为尽力防护；它们与云 VM、操作系统沙箱、出站防火墙属于不同层次。[E08]

来源 [E04] [E05] [E07] [E08]

长程协作需要明确责任

Cursor | Planner / Worker 研究经验

> Cursor 的长期实验说明，增加并发数量之前应先消除共享协调热点，并让有人持续承担全局目标与验收责任。

角色	本报告建议的职责	不应替代
规划者	明确任务与验收边界	不能靠任务数量证明进展
执行者	完成工作包并给出证据	不能自行放宽验收
裁决者	决定继续、返工或终止	不能只复述执行者总结
运行时	锁、预算、超时与恢复	不能交给提示词自觉遵守

1. 官方实验真正报告了什么

1.1 早期平级智能体通过共享文件抢任务，锁被长期持有或忘记释放；改为乐观并发后，仍出现只做低风险小改动、无人承担端到端难题的问题。[E06]

1.2 后续实验分为持续探索与派单的 Planner、专注完成分配任务的 Worker，以及判断下一周期是否继续的 Judge；周期性重开上下文缓解漂移。官方同时说明额外集成角色曾成为瓶颈。[E06]

2. 提示词要编码责任，而非人格

2.1 研究判断：Planner 的输出应是可独立验收的工作包，包含依赖、涉及文件、证据要求；Worker 不应随意重写总目标；Judge 应对照原始目标与运行证据，而非投票相信工作总结。

2.2 一个模块尚未闭环时，继续生成更多周边任务会制造虚假进展。应追踪阻塞时间、重复修改、回退率和真正满足的验收条目，避免以提交数或代码行数度量协作成功。

3. 对 Peri / AOS 的实验建议

3.1 用相同任务集比较单智能体、平级共享任务表、规划执行分层三组；固定总预算，检查完成质量、冲突成本、主上下文增长和人工介入次数。

3.2 重启前应持久化目标、已验收结果、未解决风险与下一步，而不是保存一份包含全部噪声的长摘要。对照测试应包含任务中断和工作包重复投递。

证据边界

这是 2026 年 1 月的官方研究实验，不代表当前 Cursor 产品内默认运行同一组织结构；本页角色契约是研究综合，未复制任何未公开系统提示。

来源 [E06]

记忆不能代替团队规则

Windsurf / Cascade | 当前由 Devin Desktop 文档承接

> Cascade 的记忆、规则和技能解决不同问题；当前产品迁移进一步说明，稳定知识必须拥有显式载体、作用域和迁移方式。

内容	推荐载体	主要边界
会话学到的线索	记忆候选	局部事实可能过期
团队工程约束	Git 中的规则	需审阅与版本
可重复执行程序	Skill	触发与工具应明确
本次目标与进度	任务状态	不宜升级成永久知识

1. 先校正产品生命周期

1.1 研究当日，Windsurf 的 Cascade 文档重定向到 Devin Desktop。未选偏好且 Local 可用时，新会话默认 Devin Local；企业需先启用。旧 Cascade 会话保留，不能直接把旧特性算作新引擎能力。[E14]

1.2 Cascade 自动记忆保存在本机并按工作区关联，不随 Git 分享，也不跨工作区直接复用。官方建议重要知识写入规则或 AGENTS.md；Devin Local 不持久化该旧记忆机制。[E09]

2. 提示装配的四种触发

2.1 规则可常驻、按文件 glob、按描述由模型选择或手动引用。描述模式先展示摘要，完整内容按需读取。当前优先 .devin/rules，兼容旧 .windsurf/rules。[E09]

2.2 研究判断：自动记忆适合低成本候选事实，正式约束应经审阅进入可版本化文档。否则个人试验中的偶然结论可能被误认为整个项目的永久规则。

3. 对 Peri 的启发

3.1 建议把“自动捕获事实”“团队批准约束”“临时任务计划”分成不同类型，每条都携带来源、版本、作用域和失效条件，避免同一个 memory 文本承担所有职责。

3.2 迁移配置时应列出保留、转换和丢弃的能力，并给用户可审阅差异；静默把 Workflow 改成 Skill，可能改变触发时机及行为范围。

边界

这些是公开机制和迁移说明，不能据此推断 Cascade 内部规划模型、检索排序算法或 Devin Local 与云 Devin 完全同构。

来源 [E09] [E14]

流程提示与模式不是调度器

Cascade | Workflow、Plan 与交互式澄清

> 把重复工作写成步骤能减少遗忘，但顺序提示、计划模式和真正的多智能体调度仍是三种不同机制，需要独立验收。



1. Workflow 的确定部分与不确定部分

1.1 Cascade Workflow 是 Markdown 步骤模板，用户以 slash 命令手动启动；模型不会自行挑选 Workflow，自动选择程序应使用 Skill。流程可以引用其他流程，但执行仍由智能体顺序处理。[E10]

1.2 研究判断：这种结构适合 PR 反馈处理与发布清单，不能自然推出持久化任务图、幂等重试或 exactly-once 副作用。检查表完成标记仍要关联真实命令和产物。

2. Plan 文件成为可交接产物

2.1 Plan 会探索代码、提出澄清选项，并把方案放进会话外的 Markdown；后续会话可引用该计划重新实现。官方能力表中 Plan 仍列全部工具，Ask 才明确只有搜索工具。[E11]

2.2 因而“先规划再实现”首先是流程约定，不能仅凭模式名称宣称强只读。若业务要求禁止写入，必须检查实际工具能力与权限层，不能只看界面标签。

3. 对长程任务的改进建议

3.1 计划文档应包含已决策事项、未决问题、验收标准与依赖；进度状态另存，避免每次压缩或重启重新解释整份长对话。计划变化要能追溯谁改了哪些假设。

3.2 对有副作用步骤增加前置状态检查和结果证据；重跑时先读取实际状态，再决定执行或跳过。本建议属于运行时设计，不是 Cascade Workflow 的已公开保证。

生命周期

图为本报告建议的交接闭环，不是产品强制状态机。 Devin Local 不支持旧 Cascade Workflows，官方引导迁移到 Skills。继承相同编辑器界面，不代表继承相同提示加载与执行语义。[E10]

来源 [E10] [E11]

并发工作空间要补齐环境

Windsurf / Devin Desktop | 工作树、Hooks 与 ACP

> 工作树解决代码改动分离；Hook 提供机器可判断的检查点；ACP 扩展可插接引擎，三者合起来仍不自动形成协作权限模型。

层次	解决的问题	尚需补足
工作树	源码分支分离	端口、外部数据与密钥
前置 Hook	动作前校验	错误路径与策略保护
ACP	客户端接入协议	服务端权限与身份
会话看板	人类可见的并发	依赖、仲裁和资源配额

1. 隔离源码不等于复制开发机

1.1 Cascade 工作树在会话开始时选定，缺少未跟踪的环境文件和依赖；相对路径引用也可能失效。post_setup_worktree 可补齐环境，原工作区路径通过环境变量提供。[E12]

1.2 研究判断：并行开发服务还需要独立端口、数据库命名空间与测试数据。仅换工作目录，仍可能让两个 agent 操作同一外部状态。

2. 阻断语义要精确到返回值

2.1 Cascade Hook 通过 stdin 接收 JSON。前置 Hook 以退出码 2 阻断；其他非零退出码不阻断，后置 Hook 也无法撤销已发生动作。全量 transcript 通过 JSONL 文件路径交付。[E13]

2.2 研究判断：策略脚本异常与明确拒绝是不同状态。高风险门禁必须验证错误路径，审计消费者则应按文件游标读取，避免每次把完整历史复制进内存。

3. 界面统一不等于权限统一

3.1 Agent Command Center 将本地和云会话按状态呈现，Space 聚合会话、PR 与文件。[E15] ACP 接入第三方引擎时，操作交由该引擎；registry 当前只启动预装二进制，不自动下载。[E16]

3.2 对 AOS 的建议：界面统一状态，执行端保留权限和环境身份；事件携带 agent、session、workspace 标识，避免同屏会话混用授权。

边界

Cascade 与 Devin Local 的 Hook 格式不同，迁移需核对事件及阻断语义。[E13]

来源 [E12] [E13] [E15] [E16]

把专业角色编译成工具面

GitHub Copilot | 仓库规则与自定义 Agent

> Copilot 把角色提示、工具集合、来源版本和仓库指令组织成可复用配置；真正的专业化同时减少能力噪声与行为歧义。

配置部分	负责什么	需要验证
仓库指令	工程上下文	命令是否实际有效
角色提示	目标与方法	输出契约是否清楚
工具列表	可调用能力	缺省与空集合区别
版本快照	重现一次执行	后续交互版本一致性

1. 仓库入职信息比泛化口号重要

1.1 .github/copilot-instructions.md 提供仓库规则；.github/instructions/*.instructions.md 用 applyTo 选择路径，AGENTS.md 就近提供目录约束。官方生成指令的示例要求实际运行构建和测试命令，记录前后置条件。[E17]

1.2 研究判断：提示词最有价值的内容是项目中的不明显事实，例如启动顺序、真实测试入口、必须同步修改的文件，而不是重复模型已经知道的通用风格规范。

2. 角色配置必须与能力配置一致

2.1 自定义 Agent profile 用 Markdown 和 YAML 描述提示、工具及 MCP；项目、组织和企业可分别管理。[E24] tools 缺省开放全部，空列表关闭全部，显式列表过滤；agent 工具允许调用另一个自定义角色。[E18]

2.2 profile 基于 Git 提交版本实例化，同一个 PR 后续交互保持该版本；但部分 IDE 字段如 handoffs 在云端被忽略。跨端复制配置必须验证真实支持项。[E18]

3. 对 Peri 工具冲突的启发

3.1 建议先按任务能力收缩工具面，再优化描述。一个只读审计者若仍持有 shell 与写接口，提示中的“只读”不能替代权限限制。

3.2 建议记录有效 profile 哈希、工具名单及覆盖来源；项目模板覆盖组织同名模板时必须可见，便于重现行为。

边界

自定义角色可作为主执行者或被委派；多个 profile 不证明实际并行，须查看调用链。

来源 [E17] [E18] [E24]

云交付嵌入仓库治理

GitHub Copilot | 临时环境、分支与审批

> Copilot 的工程亮点是把后台编码接入已有 GitHub 权限、检查与审阅流程，使产物和发起者都有可追溯归属。

结构示意 · 关系与数据流

节点与职责

1. 授权请求 · 明确仓库
2. 环境准备 · 安装与服务
3. 云任务 · 有时限
4. 变更分支 · 受限推送
5. 独立检查 · 默认可配置检查
6. 人工验收 · 仓库合并权

有向关系 · 后半句为条件或动作

- 授权请求 → 环境准备 · 创建会话
- 环境准备 → 云任务 · 含准备结果
- 云任务 → 变更分支 · 产生变更
- 变更分支 → 独立检查 · 校验
- 独立检查 → 人工验收 · 证据交付
- 人工验收 → 云任务 · 反馈返工

1. 执行单元有明确上界

1.1 Cloud agent 在临时开发环境中改代码、运行测试；当前文档限制每次任务为单仓、单分支、单 PR，最长 59 分钟。复杂任务因此应拆分，而不能假定它是无限续跑的跨仓项目经理。[E19]

1.2 环境由 GitHub Actions 提供，固定名称的 copilot-setup-steps job 在 agent 前运行。值得注意：一个 setup 步骤失败会跳过后续准备，agent 仍从当前环境继续。[E21]

2. 交付能力与合并权分离

2.1 官方限制推送分支，agent 不能自行批准或合并 PR；默认 Actions 在人工批准后运行。生成代码默认经过安全和第二意见检查，日志记录相关分析；这些默认检查也可以被配置关闭。[E20]

2.2 研究判断：系统保证“必须经过某门禁”与“门禁一定能发现错误”是两回事。模型复核、扫描器与 CI 可以互补，但都应留下独立结果，而不是只有一段完成宣言。

3. 对 AOS Factory 的启发

3.1 建议发起身份、工作分支、任务会话和验收记录形成一条可追溯链。让 Dev 负责提交证据，Preview 和人工验收负责各自结论，最终合并权由仓库策略控制。

3.2 建议环境失败进入显式降级状态：可继续做静态修改，但不能把未运行测试记成通过。用户应在产物中看到缺失能力及其影响。

信任边界

自托管 runner 的网络控制由部署方负责；官方要求针对该执行位置自行配置，而非照搬托管云防火墙假设。[E21]

来源 [E19] [E20] [E21]

记忆引用应在使用前复核

GitHub Copilot | 跨功能记忆与 MCP 边界

> Copilot Memory 最值得学习的是把仓库事实绑定代码证据，并在当前分支复核；共享经验的前提是防止过期事实跨作用域传播。

对象	使用前的检查	失败后的动作
仓库事实	当前分支能否支持结论	失效或重新研究
用户偏好	当前用户与归属是否匹配	不跨主体套用
MCP 工具	最小权限和副作用	缩小工具集
审阅会话	实际有效工具配置	不给写权限作默认假设

1. 事实与个人偏好分开

1.1 当前公开预览把仓库事实与用户偏好分离。仓库事实带代码引用，使用前对照当前分支验证；未使用的记录 28 天后清理，成功验证使用可重置计时。代码审阅只使用仓库事实。[E22]

1.2 研究判断：这比直接复用上一轮摘要更接近可失效缓存。仍应区分“引用存在”与“结论仍成立”：架构语义可能已改变，因此验证器不能只检查文件路径存活。

2. 工具接入也有产品边界

2.1 云 Agent 的 MCP 目前只接 tools，不接 MCP resources/prompts，远程 OAuth 服务也不支持；默认 GitHub 工具凭据只读当前仓库，默认 Playwright 只访问本环境 localhost。[E23]

2.2 配置后的工具会自主使用，不逐次请求批准；因此新增写工具就是扩大执行权限。仓库 MCP 配置也作用于 code review，不能把审阅者想当然地当作永久只读。[E23]

3. 对 Peri 记忆与能力治理的启发

3.1 建议记忆项携带仓库身份、来源提交、引用范围、推导过程和最后验证时间。恢复历史时先检查作用域，再按需载入事实，避免把另一个分支或租户的结论引入当前运行。

3.2 建议给每类 MCP 标出读写副作用和授权主体，按实际任务建立能力清单；协议支持一个接口，不代表每个产品实现都支持所有子能力。

本报告建议

对记忆增加“反证入口”：新测试、重构或用户纠正能撤销旧结论。被反复召回不应自动等同于越来越正确。

来源 [E22] [E23]

经验沉淀为可版本化程序

Devin | 从 Knowledge 到 Skills 与 Playbooks

> Devin 的经验管理正在从单独知识条目转向技能与插件；有价值的资产是可触发、可审阅、可复现的工作方法，而非无限积累记忆。

载体	表达重点	工程代价
Skill	描述触发与操作程序	需防止描述召回冲突
Playbook	完整任务与后置条件	要定期用真实任务回归
Plugin	分发与组织作用域	需要版本及权限治理
候选经验	成功/失败来源	未经验证不能自动升格

1. 当前生命周期与载入路径

1.1 Knowledge 已被官方标记弃用，正自动迁入托管 Plugins 的 Skills；触发描述映射到技能描述，正文和作用域保留，迁移不会自动向用户仓库提交文件。[E25]

1.2 仓库 Skill 的后台索引让会话启动时可发现程序；仓库克隆后磁盘扫描覆盖同仓库的索引版本。模型先看名称与描述，调用后才载入正文；当前只允许一个 active skill。[E26]

2. Playbook 表达完整工作契约

2.1 Playbook 作为人工附加到会话的模板，可以分开写 Procedure、Specifications、Advice、禁止动作和用户所需输入；其中 Specifications 描述结束后必须成立的状态，版本历史支持回退。[E27]

2.2 研究判断：方法步骤与验收后置条件分开，能防止“按步骤做完但目标没达到”。不过 Forbidden Actions 仍是语言约束，需要另用权限系统承接硬限制。

3. 对 Peri / RSI 的启发

3.1 建议把成功与失败 trace 聚合成候选技能变更，以 diff 说明修改触发、流程还是验收标准；经验证库验证后再升级版本，而不是直接修改所有会话的永久提示。

3.2 需要测试索引与分支文件的覆盖规则：刚提交技能、仓库未克隆、旧索引仍存在时，应能解释所用版本。

边界

Skills、云 Memory、旧 Knowledge 与 Cascade 本机 Memories 须按实际表面和版本区分。

来源 [E25] [E26] [E27]

把多智能体编排变成可恢复程序

Devin | Managed Devins 与 Dynamic Workflows

> Devin 把会话式协调和代码式编排分成两条路径；后者通过结构化结果与运行记录，让大规模并行任务能够观察、停止并恢复。

结构示意 · 关系与数据流

节点与职责

1. 工作流脚本 · 依赖与预算
2. 审计任务 A · 独立 VM
3. 审计任务 B · 独立 VM
4. 修复任务 A · 消费结构化结果
5. 汇总验证 · 等待必要结果
6. 运行记录 · 记录结果；幂等键为建议

有向关系 · 后半句为条件或动作

- 工作流脚本 → 审计任务 A · 派发
- 工作流脚本 → 审计任务 B · 派发
- 审计任务 A → 修复任务 A · 无需等 B
- 修复任务 A → 汇总验证 · 修复证据
- 审计任务 B → 汇总验证 · 审计证据
- 审计任务 A → 运行记录 · 记录
- 修复任务 A → 运行记录 · 记录
- 运行记录 → 工作流脚本 · 复用已完成结果

1. 会话式协调适合动态探索

1.1 Managed Devins 各自运行在独立 VM；协调者可派单、发消息、看预算、休眠或终止子任务。子任务完成或需要输入会唤醒协调者，减少依赖持续轮询。[E28]

1.2 研究判断：事件唤醒改变的是调度效率，不能保证工作包本身互不冲突。协调者仍需定义提交基线、写入所有者和最后集成策略。

2. 代码式编排适合稳定结构

2.1 Dynamic Workflows 使用 Python 控制派单顺序与提示构造；agent 调用以 JSON Schema 返回结构化结果。pipeline 允许单个条目完成一阶段就进入下一阶段，parallel 用在确需等待全部结果的位置。[E29]

2.2 恢复以 run ID 取记录，重跑脚本并按提示、schema、执行配置哈希复用结果；脚本须保持确定性，不得直接依赖时间、随机或外部状态。未完成调用重开会话；改提示会重跑该项及下游。[E29]

3. 对 AOS 编排核心的启发

3.1 建议将模型负责的任务判断与运行时负责的依赖、预算、取消、恢复分开。异步消息需区分进度、求助、最终结果，并携带任务版本，避免迟到结果覆盖新决策。

3.2 建议外部写操作保存业务幂等键，恢复前先核实副作用。默认独立 VM，也可选共享 VM；后者共享未提交文件，不能假定写入隔离。停止会让剩余子会话休眠，非事务回滚。[E29]

证据边界

公开文档说明外部契约和编排原语；底层调度器、持久化存储与隔离实现细节未公开，本报告不把恢复契约扩张为分布式事务保证。

来源 [E28] [E29]

交付证据与权限一起继承

Devin | 环境快照、验收录像与安全配置

> 可靠自治需要既能复现执行环境，也能证明结果；多智能体权限必须沿继承链收紧，不能通过启动子任务绕过。

证据或控制	证明什么	不能证明什么
环境快照	同一启动基线	外部服务完全相同
测试与录像	选定路径运行结果	所有边界条件正确
强制 Profile	可声明的权限上界	自托管网络已执行策略
子任务继承	委派不扩权	存量会话已即时撤权

1. 环境定义是可复现资产

1.1 Devin 的 blueprint 构建可启动 snapshot，每个会话从新副本开始，会话改动不回写基础快照。准备工具链和依赖能缩短开始工作的时间，也减少“无法验证”的环境失败。[E30]

1.2 测试录像流程从 diff 和代码定位关键路径，再准备环境、执行浏览器操作并记录结果；官方明确这是短而聚焦的主路径验证，不能替代完整测试套件。[E32]

2. 限制沿层级取交集

2.1 Security Profile 可约束网络、MCP、Git 与 GitHub CLI。强制配置作为上界，低层只可收紧；子会话继承同一约束链。配置在创建或唤醒时解析，已运行会话不会即时追随改动。[E31]

2.2 自托管 Outposts 的网络策略由运营者执行，Devin 发布有效策略却不会替客户机器安装防火墙。研究判断：声明了策略与实际强制执行必须有不同状态与验证证据。[E31]

3. 对 AOS 可信交付的启发

3.1 建议交付包包含基础环境版本、代码提交、测试命令与结果、录像对应场景，以及未验证项。录像表现成功只覆盖观察到的流程，不能作为全部功能正确的替代指标。

3.2 建议权限快照带 generation：派生会话不应扩大父任务能力；紧急撤权需要主动终止或更新运行实例，不能只修改下一次启动才读取的配置。

本报告建议

可信交付的最小单位应是“版本 + 环境 + 验收证据”。把模型完成状态直接映射为可合并状态，会丢失这三者之间的关系。

来源 [E30] [E31] [E32]

规范成为任务依赖的载体

Kiro | Spec、并行 Wave 与性质测试

> Kiro 的突出设计是把需求、设计和执行任务变成可审阅文件，并将依赖与验证追溯到需求；文档本身仍不等于正确性证明。

结构示意 · 关系与数据流

节点与职责

1. 需求与不变量 · 明确保留行为
2. 设计 · 接口与数据流
3. 任务依赖图 · 可验收工作包
4. 独立任务 A · 当前 wave
5. 独立任务 B · 当前 wave
6. 性质与验收 · IDE可选性质测试

有向关系 · 后半句为条件或动作

- 需求与不变量 → 设计 · 设计输入
- 设计 → 任务依赖图 · 拆解
- 任务依赖图 → 独立任务 A · 并行
- 任务依赖图 → 独立任务 B · 并行
- 需求与不变量 → 性质与验收 · 验证依据
- 独立任务 A → 性质与验收 · 实现结果
- 独立任务 B → 性质与验收 · 实现结果
- 性质与验收 → 需求与不变量 · 反例促成修订

1. 从自然语言到可跟踪任务

1.1 Spec 生成 requirements.md 或 bugfix.md、design.md、tasks.md。当前文档支持 Feature、Bugfix 与简化 Quick Spec；Bugfix 显式记录当前、期望和应保持不变的行为。[E33]

1.2 执行全部任务时分析依赖，按 wave 调度：同一 wave 内独立任务并行，后续 wave 等前置依赖完成。2026-10-15 主文档已列 IDE、CLI、Web 支持该能力。[E33]

2. 从验收条件到可运行反例

2.1 Correctness 将 EARS 需求转成可测试性质，生成随机输入并缩减反例；失败后可能改实现、测试或需求。官方明确 PBT 并非形式证明且默认可选；当前能力表仅列 IDE 支持，不可因 CLI 支持 Spec 就一并外推。[E33][E34]

2.2 研究判断：自动提炼性质也可能把错误需求固化。关键不变量应由领域负责人确认；仅测试生成器与实现互相一致，可能产生系统性的共同盲点。

3. 对 Peri 优化任务的启发

3.1 建议每个性能或稳定性目标拥有量化指标、测试条件、允许变化和保持不变行为，再链接到实现任务。若只写“降低内存”，agent 可能通过丢弃必要历史得到虚假收益。

3.2 并行关系应由读写集与依赖验证，而非只由任务标题推断。共享协议或数据结构改动先确定接口，再放开独立实现；wave 过粗也会让慢任务拖住整个阶段。

来源冲突

较旧 Q Developer 迁移表仍写 CLI 无 Spec；本页采用更新时间更近的 Specs 主文档，并保留该差异，避免把旧兼容表当成当前能力清单。[E41]

来源 [E33] [E34] [E41]

持续指导与事件检查分开

Kiro | Steering、Hooks 与权限

> Steering 描述如何工作，Hook 在事件点执行检查，权限决定动作是否允许；三层分离可以减少长提示承担不可能的安全责任。

层次	输入与输出	失败处理建议
Steering	领域规则 → 模型上下文	显示未加载或冲突
Agent Hook	事件 → 新提示	不能当作硬拒绝
命令 Hook	事件 JSON → 校验结果	明确超时与错误语义
授权策略	动作身份 → 许可	不允许模型自行扩大

1. 项目知识按用途装配

1.1 Steering 的 product.md、tech.md、structure.md 分别描述产品、技术与结构；其余文档可常驻、文件匹配、手动或按描述自动包含。工作区规则优先于全局；自定义 Agent 必须用 resources 显式带入 Steering。[E35]

1.2 研究判断：不同 harness 应解释各自加载方式；切换角色后展示有效知识清单，检查是否遗漏项目约束。

2. 从事件进入确定检查

2.1 当前 Hook 配置位于 .kiro/hooks/*.json，以 trigger、matcher、action 描述触发点。命令动作接收 stdin JSON，Agent 动作则向当前会话注入提示；两者不是同一种保证。[E36]

2.2 文件事件适合格式化，工具前置事件适合检查条件。后置检测只能报告或补救，不能取消外部副作用；脚本超时与错误策略须明确。

3. 对 Peri 工具治理的启发

3.1 建议规则负责工作方法，工具注册负责能力收缩，Hook 负责检查，授权层负责许可。全部塞进系统提示，会使冲突难解释。

3.2 高频流式任务宜按工具提交、文件写完、阶段结束等语义事件检查，避免每个 token 触发昂贵脚本；事件序号要关联结果。

生命周期

IDE 1.0 / CLI V3 使用独立 JSON Hook；旧 profile 与 Hook 需转换。配置版本应显式适配。[E36]

来源 [E35] [E36]

统一产品不等于统一执行边界

Kiro / Amazon Q Developer | Agent Profile 与自治运行

> Kiro 承接 Q CLI 的技术与配置，同时扩展本地和云执行；评估时要按具体版本和表面区分 profile、沙箱与自主编排能力。

对象	本地表面	云端表面
自定义角色	可作为主会话角色	可供子任务委派
隔离对象	命令及可选 stdio 服务	托管云执行环境
上下文来源	本地文件与全局配置	仓库与显式同步配置
续跑责任	本地进程与会话	自治状态与用户干预

1. 先识别继承与变化

1.1 Kiro CLI 承接 Q CLI，但许可并不继承：旧 Q CLI 为 Apache，当前 Kiro IDE 和 CLI 的官方许可均为 AWS Intellectual Property License。其开源组件归属不证明整个引擎已开源。[E40][E42]

1.2 当前自定义 profile 分开声明 tools、permissions、resources，可用 JSON 或 Markdown。IDE / CLI 可选主 Agent；Web 的项目自定义 Agent 用于子任务委派，不能直接选为主会话角色。[E37]

2. 本地沙箱和云自治是不同维度

2.1 本地沙箱启用且后端可用时约束 shell / 子进程，可包裹 stdio MCP；harness、内建文件工具与远程 MCP 在外。后端缺失时可无 OS 隔离执行，要求保护的模式例外。[E38]

2.2 Web 自治模式默认关闭，启用后规划、委派和验收，必要时等待用户；它与显式定义步骤和循环的 Workflow 是不同编排方式。[E39]

3. 对 Peri / AOS 的架构建议

3.1 建议 profile 声明行为、能力、上下文，执行实例另声明隔离位置和资源身份；从本地迁云时必须重新确认权限及数据可见性。

3.2 自治任务应区分运行、等待用户、预算耗尽、环境失败和完成；停止输出不代表结束。恢复必须识别未满足的等待条件。

证据边界

本页是公开行为契约。Q CLI 的历史开源许可不能证明 Kiro 当前引擎或云编排实现已开源。

来源 [E35] [E37] [E38] [E39] [E40] [E42]

Qwen：提示词与工具同步

Qwen Code / 提示词工程

> 值得学习的是按运行环境和真实工具集合生成提示词，让模型收到的工作方法与运行时能力相符，减少不存在的调用和自相矛盾的指令。

组件	给谁阅读	设计收益
角色 description	父代理路由	说清何时委派与何时不用
角色正文	被委派代理	限定任务步骤与结果格式
工具能力快照	提示词生成器	删去不可执行的指导
宿主交互模式	问题与停止策略	避免无人可答仍反复提问

1. 提示词是一段可执行配置的投影

1.1 prompts.ts 用 resolveInteractionMode 区分交互终端、无交互运行和 ACP。澄清问题是否可返回答案，取决于宿主交互通道；同一任务原则因入口不同而生成不同指导。[F03]

1.2 PromptToolSurface 把已声明工具与 Agent 的延迟可达性传入提示生成；gateToolGuidance 删除不可调用工具的用法，filterToolCallExamples 同步裁剪示例，避免正文删除后示例仍诱导错误调用。[F03]

2. 工程职责清楚，扩展才不会串线

2.1 架构文档把模型循环、上下文、权限和工具注册置于 packages/core；CLI 与 ACP 宿主承接输入、取消和结构化事件。提示词可复用，用户界面的交互假设仍须由宿主明确提供。[F02]

2.2 子代理由 Markdown 元信息描述用途、模型和工具；description 面向路由选择，正文面向执行。这两段各有消费者，不能用一段冗长人格说明兼任任务触发条件。[F08]

3. 分析：提示词质量应成为接口测试

3.1 建议为 Peri 建立“工具开关→最终提示”的契约检查：禁用工具后，正文、示例、技能目录和委派描述都不再承诺该能力；延迟工具则明确发现方式。

3.2 只检查文本出现与否还不够：要覆盖工具别名、代码模式和子代理可见范围。真正收益是降低模型遇到矛盾时自行发明替代路径的概率，而非单纯缩短提示词。

Peri 启发

以上是设计建议，不代表已经审计 Peri 当前实现。优先让工具注册表成为提示词能力声明的唯一事实来源，再评估增加更多指令的收益。

来源 [F02] [F03] [F08]

Qwen：子代理与团队不同

Qwen Code / 角色与协作边界

> 一次委派、可持续协作的队员、带历史的 fork 和独立 worktree 是不同维度；把它们混成一种“多智能体”，会隐藏权限和生命周期问题。

维度	回答的问题	不能自动推导
命名子代理	谁适合这类任务	已获得全部父历史
fork	从哪份上下文开始	可以管理父会话
团队队员	如何持续领取与沟通	彼此共享全部输出
worktree	代码改动放在哪里	拥有完整运行环境隔离

1. 普通委派重在缩小任务上下文

1.1 官方文档分别描述命名子代理与 fork：前者按定义接收任务，后者用于需要父会话背景的场景。工具白名单与黑名单共同收窄能力，maxTurns 用于限制运行预算。[F08]

1.2 工具策略源码进一步排除普通子代理的会话控制面能力：定时任务、父侧停止、团队管理、共享待办和工作树管理等不能仅靠提示词说“不要碰”。团队队员则保留协作所需的消息和任务工具。[F06]

2. 团队提示词描述的是操作协议

2.1 promptAddendum.ts 要求队员先查任务、认领、执行、报告、完成，再取下一项；队员最终文字由运行时转给 leader，同行不会自动看到这些文字，需要显式发消息。[F04]

2.2 readOnly 分支强调调查和证据；planModeRequired 分支要求通过 exit_plan_mode 发起计划批准、拒绝后修订。角色文字表达行为，实际允许的工具还必须结合运行时策略核验。[F04][F06]

3. 分析：别把分支当成权限

3.1 工作树隔离解决文件修改冲突的一部分，不自动隔离进程、网络、凭证和仓库外路径。任务定义应分别声明历史来源、写入目录、工具能力、审批继承与结果归属。

3.2 对长任务更有价值的配置是可观测的委派契约：为何委派、交付什么、如何验收、何时结束。默认一直自己动手的问题，应先查路由描述和成本暗示，再增加角色数量。

边界

权限模式存在继承和覆盖规则，应按当前版本的工具策略核实。配置写了“只读”不能单独证明所有间接调用链都只读。

来源 [F04] [F06] [F08]

Qwen：协作状态要能竞争

Qwen Code / 任务归属与消息存储

> 团队真正困难的是两个代理同时认领、消息消费与任务更新的竞态；Qwen 把这些问题落到锁和状态检查，而不是只要求模型互相礼让。

结构示意 · 关系与数据流

节点与职责

1. 读取任务快照
2. 请求更新
3. 任务锁
4. 检查归属与状态
5. 原子写入
6. 冲突后重读

有向关系 · 后半句为条件或动作

- 读取任务快照 → 请求更新 · 携带预期状态
- 请求更新 → 任务锁 · 串行化
- 任务锁 → 检查归属与状态 · 读取最新状态
- 检查归属与状态 → 原子写入 · 匹配
- 检查归属与状态 → 冲突后重读 · 不匹配
- 冲突后重读 → 读取任务快照 · 重新决策

1. 任务认领在写入点决定

1.1 team/tasks.ts 的 updateTask 在任务锁内读取最新记录，再按 callerName 保护归属、状态、描述和依赖等关键字段；metadata 与 activeForm 注解允许队友更新，不能说整条记录均禁止改写。[F07]

1.2 leader 的分派还可携带 expectedOwner 与 expectedStatus；旧快照在提交时发生变化就抛出冲突，让调用者重读。任务依赖、拥有者和状态因此有明确的并发更新语义。[F07]

2. 邮箱是独立的数据结构

2.1 mailbox.ts 以每个 inbox 路径的进程内互斥锁串行化本地访问，再用跨进程文件锁保护其他进程；读改写最终通过 atomicWriteJSON 落盘。不同收件箱不必抢同一把全局锁。[F05]

2.2 consumeUnread 在锁内选出未读项并写入已读标记；发送时保留未读消息、清理过期已读消息。团队销毁还清除锁对象缓存，说明生命周期本身也是内存管理问题。[F05]

3. 分析：本地正确性不等于集群一致性

3.1 文件锁和原子替换适合共享本机目录下的协作，不能据此宣称跨主机一致性、消息恰好处理一次或崩溃后业务效果可撤销。消费标记与外部副作用仍可能发生在不同事务。

3.2 若把类似协议引入 AOS 工作空间，建议测试双认领、过期分派、发送与销毁竞争、接收后崩溃四种场景；记录任务版本和结果引用，使失败能明确归因到运行时或业务验收。

Peri 启发

建议把“任务已认领”“结果已送达”“验收已通过”分成三个事件。代理说完成可以产生候选结果，验收事件应另有权威来源。

来源 [F05] [F07] [F45]

Trae：把修复流程写具体

Trae Agent / 单代理提示与完成信号

> Trae Agent 的可取之处是将修复任务拆成可观察的工程动作；同时源码说明，调用完成工具与证明修复正确仍然是两件事。

信号	源码中能说明	不能说明
task_done	模型请求收束	需求已正确实现
过滤后 diff 非空	存在未被路径规则排除的改动	属于业务代码或修复正确
max_steps	运行预算达到上限	已经获得最优方案
验证报告	可供后续检查的证据	所有测试真实执行过

1. 提示词要求形成证据链

1.1 trae_agent/prompt/agent_prompt.py 依次组织问题理解、代码定位、先复现、诊断、最小修复、回归验证和总结。绝对路径规则单独前置，针对工具接口常见误用作明确约束。[F10]

1.2 sequential_thinking 提供结构化推演的用法提示，但它仍是同一代理的工具，不应计为第二位独立审查者。多次反思也不能替代外部测试或不同假设的验证。

2. 收束由结构化信号驱动

2.1 TraeAgent 覆盖基础类的完成检测：只有响应包含 task_done 调用才视为主动完成，避免把普通回复中的“done”当作终态。基础循环同时以 max_steps 限制推进。[F11][F12]

2.2 must_patch 为 true 时，完成检查按路径模式过滤 git diff 中的测试相关改动，再要求剩余非空；这只是启发式检查，可能留下文档等改动，既不能识别全部测试，也不要求测试通过。[F12]

3. 分析：应把测试承诺变成结果类型

3.1 适合借鉴的提示结构是“输入事实 / 定位 / 可复现证据 / 最小改动 / 验证结论”，每段都有产物。若任务不是修复，应切换流程，避免为解释问题强制制造补丁。

3.2 对 Peri 可定义完成候选携带补丁引用、验证命令、退出码、失败项和未验证项；运行时只检查证据完整，项目门禁再决定是否验收。这样不会把诚实报告失败的代理误标为系统故障。

边界

本页研究开源 bytedance/trae-agent 的 CLI 与源码；不能把这些机制直接归属到闭源 TRAE IDE 或其全部产品模式。

来源 [F09] [F10] [F11] [F12]

Trae：轨迹可读也有成本

Trae Agent / 观测与运行资源

> Trae 的轨迹格式把模型输入、工具调用与步骤状态放在一起，适合研究复盘；将它直接用于超长会话，需要重新设计增量存储和恢复语义。

结构示意 · 职责分层

节点与职责

1. LLM 交互记录 · 输入、响应、用量
2. 步骤记录 · 调用、结果、错误
3. 内存轨迹对象
4. 完整 JSON 文件
5. 离线复盘

有向关系 · 后半句为条件或动作

- LLM 交互记录 → 内存轨迹对象 · 追加
- 步骤记录 → 内存轨迹对象 · 追加
- 内存轨迹对象 → 完整 JSON 文件 · 整体序列化
- 完整 JSON 文件 → 离线复盘 · 读取

1. 一次运行有两种观察尺度

1.1 TrajectoryRecorder 同时保存 llm_interactions 与 agent_steps：前者记录输入消息、响应、模型、使用量和可用工具；后者记录每步状态、工具结果及错误，能回答“模型看见了什么”和“执行了什么”。[F13]

1.2 每次记录会把对象追加到内存中的 trajectory_data，随后以写模式序列化整份 JSON。实现直白、易于离线分析；在长轨迹下，重复输入消息和全量重写都值得单独测量。[F13]

2. 执行环境是配置分支

2.1 BaseAgent 在有 docker_config 时建立 DockerManager，并把 bash 与编辑类工具交给 DockerToolExecutor；否则使用原始工具执行器。因此不能把“框架支持 Docker”表述成所有调用默认隔离。[F11]

2.2 主循环对步骤异常设置错误状态并结束，预算耗尽返回未完成；结束阶段关闭工具并清理 MCP 客户端。此处可见运行资源释放路径，未由此证明持久任务可任意断点续跑。[F11]

3. 分析：把轨迹拆为事实日志与查看视图

3.1 对千轮会话，建议把不可变事件、消息正文和索引分离；大正文按内容标识去重，查看器按游标恢复。模型输入可以由引用重建，不必在每轮重新持有同一大对象的副本。

3.2 验证应覆盖峰值常驻内存、每轮写入字节、恢复首屏延迟和异常尾部可读性。不要只比较压缩后文件大小：全量 JSON 的构建和编码峰值可能早于磁盘节省发生。

Peri 启发

这是从 Trae 实现推导的存储建议，不是对 Peri 当前内存根因的结论。观测日志、运行上下文和前端历史应分别测量，避免把文件落盘误认为对象已经释放。

来源 [F11] [F13]

Trae：多代理用于补丁搜索

Trae Agent / 生成、剪枝、选择

> Trae 的研究路线把多代理用于探索候选补丁，再筛选与投票；价值取决于候选差异和选择证据，不能简单等同于“多个人聊天更可靠”。

阶段	真正产物	常见盲区
生成	不同假设下的补丁	同模型输出高度同质
剪枝	失败证据与去重映射	错误回归集淘汰好方案
选择	候选比较和测试证据	被迫从全错候选中选一个
聚合	票数及支持理由	票数掩盖共同假设

1. 三个阶段具有不同职责

1.1 原始论文让 coder 独立产生多个候选，经规范化去重与回归测试剪枝，再让 selector 读取仓库并按需运行测试；多个选择结果可投票聚合。增加候选数也增加筛选成本。[F14]

1.2 仓库把 selector 放在 evaluation/patch_selection 的独立评测入口；输入是候选 JSONL，可带回归结果，可分组、可开启 majority_voting。默认 CLI 工作流不应被写成始终自动执行完整集成搜索。[F49]

2. 提示假设会决定失效方式

2.1 selector_agent.py 要求只选已有补丁，不提出新补丁，并声明候选中至少一个正确；结果解析无效时可回退到第一个候选，运行预算耗尽也保留默认候选。这个决策契约适合受控实验，生产环境可能不成立。[F50]

2.2 分析：候选都错、回归集漏测或同一模型共享盲点时，多数票仍会一致地错。生成器数量只是探索预算；是否覆盖不同根因、是否使用独立证据，才决定信息增量。

3. 对抗审查应允许无人胜出

3.1 建议把最终选择扩展为“接受一个、拒绝全部、补充验证、请求决策”四态。选择器必须列出能区分候选的测试或反例，不能仅以文字解释更流畅作为依据。

3.2 对高危 Peri 变更，可让独立任务分别攻击状态机、持久化、取消/恢复，再由一个拥有测试证据的整合者处理冲突；用缺陷发现增量和误报率评估，而不是以代理数量验收。

边界

论文的实验轮数、脚本示例预算与 SelectorAgent 默认值并不相同。本报告不混用它们，也不把历史榜单结果外推为当前生产修复成功率。

来源 [F14] [F49] [F50]

Open SWE：模型循环与后端分开

Open SWE / 当前 MDA 架构

> 当前 Open SWE 已把模型循环和检查点放到 Managed Deep Agents，后端继续掌握沙盒、凭证和工具门禁；理解这个边界比照搬旧架构图更重要。

结构示意 · 关系与数据流

节点与职责

1. 用户与集成入口
2. Open SWE 后端 · 身份、沙盒、门禁
3. MDA 运行时 · 模型循环、检查点
4. 模型提供者
5. 工具与线程沙盒
6. 转录与状态

有向关系 · 后半句为条件或动作

- 用户与集成入口 → Open SWE 后端 · 触发线程
- Open SWE 后端 → MDA 运行时 · 签名运行上下文
- MDA 运行时 → 模型提供者 · 推理
- MDA 运行时 → Open SWE 后端 · hooks / 工具调用
- Open SWE 后端 → 工具与线程沙盒 · 执行
- Open SWE 后端 → 转录与状态 · 记账与事件

1. 新线程的真实执行入口

1.1 固定快照的 mda/README.md 明确：新 agent、reviewer 与 review-scout 线程创建于 MDA；已有后端线程保留兼容路径。README 同时提示项目仍在快速开发，原样部署依赖 MDA 私有测试资格。[F15][F17]

1.2 mda/agent.py 从后端 catalog 获取模型、工具与子代理定义，再调用 define_deep_agent。运行上下文携带签名令牌；MDA 不持有 GitHub、Slack 或数据库凭证，工具实际调用回后端。[F16][F17]

2. 一轮推理跨过可审计边界

2.1 BackendRunMiddleware 将 prepare、before_model、after_model 等阶段交给后端 hooks；模型回复和运行时工具调用另行报告。后端因此保留转录、权限门禁和使用量视图，而模型循环可以独立托管。[F16][F17]

2.2 本轮目录区分直接向模型暴露的工具与只能从代码调用的工具。子代理获得独立工具集合和结果报告中间件；不能假定父代理的全部过滤器天然作用于另一个子代理图。[F16]

3. 开源部署仍有信任边界

3.1 当前安装指南明确：普通成员编码沙盒通常使用安装级 GitHub App 凭据；工作空间绑定和私有线程可见性不收窄仓库访问，预载源码也不会因令牌缩权消失。部分审查/公开事件线程使用较窄令牌。[F51]

3.2 指南要求不同信任组采用不同部署、App 安装及受限镜像。对 AOS 的设计启发：工作空间路由、界面可见性、凭据和预载数据分别验收；不能把目录分组当作组织授权。此处是官方部署合同，未做越权复现实验。

版本边界

模型循环和安装信任模型均固定于 aa00dc8；本轮重新读取部署文档。旧 openwiki 仍写旧路径，仅作漂移证据。源码与文档阅读不等于部署实测。[F48][F51]

来源 [F15] [F16] [F17] [F48] [F51]

Open SWE：提示词定义交付协议

Open SWE / 场景分流与权限

> Open SWE 的提示词把工程任务嵌入来源、仓库、协作与交付场景；最有价值的部分是防止同一请求在不同通道触发错误工作流。

层次	解决的问题	检验方式
来源上下文	给谁、在哪里回复	多通道结果回传测试
任务意图	解释、实现或审查	近义请求分流用例
共享工程规则	怎样验证与报告	失败结果不能被美化
运行时门禁	哪些动作可执行	直接与间接调用路径

1. 用片段而非一篇固定长文

1.1 主 Jinja 模板依次组合工作环境、来源上下文、仓库范围、协作、任务执行、依赖、提交、仓库指令与工作区信息。静态 Markdown 与带变量模板由 prompts.py 分开处理。[F18][F46]

1.2 这种装配使 Slack、GitHub 和面板入口可以共享工程原则，又保持回复地点、参与者和仓库约束不同。分析上，片段之间仍应有优先级与冲突检查，否则只是把一份长提示拆成许多文件。

2. 让相近意图区分到不同动作

2.1 task-execution.md 将信息查询、代码变更、自动 PR review 与人工 review 分流；准备另启实现任务前，要求查找已有 PR 工作线程并复用，减少两个代理同时修复同一变更。[F19]

2.2 shared-base.md 要求核验当前提交的完整 CI 状态；未知、取消、待运行不能被说成通过。后台命令共享工作树，也需要避免与编辑和提交竞争。这里主要是行为指导，不能直接当作强制隔离。[F47]

3. 提示与门禁承担不同责任

3.1 pr_creation_guard.py 在工具包装层识别若干通过 gh 或 API 直接创建 PR 的回退路径，限制绕过既定交付入口。源码检查覆盖的命令形式有边界，并不是所有 shell 副作用的完备证明。[F20]

3.2 建议 Peri 将“什么请求需要什么交付”放入提示规则，将“谁能向哪里写”放入权限层；测试使用同一任务从多个入口发起，检查结果路由、身份和去重是否一致。

Peri 启发

建议建立输入意图到交付结果的结构化契约。提示词应能解释选择，工具层应能拒绝违规动作，两者共享同一任务身份与授权上下文。

来源 [F18] [F19] [F20] [F46] [F47]

Open SWE：长任务需要连续性

Open SWE / 跟进、压缩与恢复

> 长任务不是把上下文无限增长：新指令要有注入时机，压缩要留下恢复线索，沙盒失联则必须保留未提交工作的存在性。

结构示意 · 关系与数据流

节点与职责

1. 线程新消息
2. 模型前检查
3. 摘要与历史引用
4. 下一轮推理
5. 当前工作沙盒
6. 失联待触发

有向关系 · 后半句为条件或动作

- 线程新消息 → 模型前检查 · 排队
- 模型前检查 → 摘要与历史引用 · 必要时压缩
- 摘要与历史引用 → 下一轮推理 · 精简上下文
- 下一轮推理 → 当前工作沙盒 · 执行工具
- 当前工作沙盒 → 失联待触发 · 无法访问
- 失联待触发 → 当前工作沙盒 · 重新触发同线程
- 模型前检查 → 下一轮推理 · 无需压缩

1. 在模型边界接收新信息

1.1 check_message_queue_before_model 在每次模型调用前读取线程消息队列快照，按先后顺序注入；结构化 envelope 各自成为消息，避免多个来源混成一块不可解析文本。[F24]

1.2 TaskCoordinationMiddleware 另处理协调者与 worker 的任务上下文、欠送消息和取消状态；上下文 hash 用于避免仍可见的信息重复注入。它与普通对话跟进不是同一层概念。[F22]

2. 把压缩过程作为状态事件

2.1 ConversationOffloadingMiddleware 包装摘要能力，隐藏摘要模型的内部流式输出，向界面发出开始、失败和完成事件；完成记录包含截断位置与历史文件路径，手动压缩结束后可直接结束本次运行。[F21]

2.2 MDA 负责模型循环/checkpoint 与原生摘要，后端索引和工具状态另管；本地每次新 mda dev 启动会重建 .mda/build、清空检查点。开发重启不能当生产恢复演练，线程索引存在也不证明执行检查点尚在。[F17]

3. 恢复不能伪造成功

3.1 sandbox_circuit_breaker 报告“沙盒不可达”，不武断宣称已经永久丢失：编码线程失联时不静默创建空沙盒替换，防止把尚未提交的文件丢失包装成恢复成功。只读审查允许不同策略。[F23]

3.2 建议 Peri 验证压缩后继续输入、取消时收到队员结果、历史游标回读和沙盒短暂掉线的组合场景。恢复事件应携带同一任务与工作目录身份，UI 不能仅因新进程启动就显示已恢复。

边界

代码中存在事件和文件引用，不代表已验证千轮吞吐或压缩后的内存下降。本报告未运行这些产品的长程基准，建议以峰值内存和恢复正确性共同验收。

来源 [F17] [F21] [F22] [F23] [F24]

AutoGen：路由与执行分开

AutoGen / 发言者选择与消息契约

> AutoGen 的研究价值在于将代理、消息和团队路由分成可组合部件；SelectorGroupChat 轮流选择一个发言者，并不天然并行执行整个开发团队。

能力	实际语义	工程代价
模型选择发言者	一轮选一个候选	多一次路由推理
候选函数	代码缩小允许集合	状态逻辑需要维护
工具后反思	同一代理再次推理	增加 token 和延迟
共享会话	消息分发给成员	上下文复制与噪声

1. 描述服务于路由，提示服务于角色

1.1 SelectorGroupChat 根据候选的名字、description 与共享对话选择下一位发言者；被选代理执行后把回复广播给成员，再判断是否终止。共享的是消息历史，而非自动共享同一个可写工作目录。[F26]

1.2 源码允许 selector_func 确定性指定人选，candidate_func 收窄候选；只有仍需选择时才调用模型。模型名称输出会被解析、重试；失败后回到上一发言者或第一个候选。[F27]

2. 代理对象拥有自己的会话状态

2.1 AssistantAgent 明确要求调用者仅传入新消息，禁止每轮再传整份历史；同一实例不能被并发任务或协程同时使用。把实例随手放进共享单例，会破坏这一状态契约。[F30]

2.2 工具执行后的 reflect_on_tool_use 再次调用模型整理结果，属于同一代理的反思；多个工具调用可并发，但发言选择仍是一位。工具并行、代理并行和多角色轮转应分别标注。[F30]

3. 分析：把容易出错的调度写进代码

3.1 如果“先规划、再执行、最后审查”是必需顺序，建议用状态或候选过滤固定它；模型只负责在允许集合中作有意义选择。长提示词反复强调顺序不能消除路由不确定性。

3.2 对 Peri，可把角色 description 做成短小的适用/不适用判据，把工具结果整理与独立审查分开计费。评估时记录实际并发数、重复查询量和路由回退次数，避免角色越多看起来越先进。

状态提醒

当前 README 宣布 AutoGen 项目归档且微软停止维护，并指向 Agent Framework；这是项目声明，本轮未见平台归档横幅。此页学习历史接口，不视为活跃产品承诺。[F52]

来源 [F25] [F26] [F27] [F30] [F52]

AutoGen：停止必须是接口

AutoGen / 终止、消息与维护状态

> “对话停止”“代理空闲”“工具完成”“业务验收通过”应分别表示；AutoGen 的终止接口提供了组合方式，但业务成功仍由应用定义。

状态	可采用的证据	不能替代
停止	StopMessage 与原因	任务验收
空闲	无待处理调度项	所有外部副作用完成
消息发布	发布调用返回	接收方处理成功
验收	测试、审查、产物版本	代理自我声明

1. 独立而可组合的终止条件

1.1 TerminationCondition 是有状态的调用对象，消费新增消息，返回 StopMessage 或空值；触发后需要 reset，AND 与 OR 条件可组合。团队条件和单代理工具迭代上限属于不同控制层。[F28][F29]

1.2 分析：实际开发流水线应把预算耗尽、用户取消、等待人类和成功完成分成不同原因；仅以文本包含 TERMINATE 停止，无法证明补丁测试通过，也可能被引用文本意外触发。

2. 等待消息不等于等待业务完成

2.1 Core 官方指南区分直接消息与发布订阅。publish_message 虽需 await，以便调度投递，但返回值是空；接收方处理异常会记录，却不作为发布者的调用失败返回。[F31]

2.2 因此“消息已发布”不能当作“审查已完成”。建议应用显式定义请求编号、完成事件、失败事件与超时，并在聚合器中等待所需任务的结果，而不是等待队列暂时空闲。

3. 选型要看当前维护责任

3.1 2026-10-16 的官方 README 已将 AutoGen 标为归档，明确不再提供错误修复、安全修复、新功能或支持，建议迁移 Microsoft Agent Framework。早先“维护模式”的文章不能覆盖这次更新。[F25]

3.2 对 Peri 的合理借鉴是提取稳定思想：类型化消息、可组合终止、状态保存接口与可替换路由。是否引入框架依赖，应另行评估当前维护者、升级路径和自身运行时边界。

Peri 启发

建议将停止原因与业务结论分开存储。恢复时可以重新调度未完任务，却不应把“已取消”无意转成“成功”，也不应因一次空闲事件丢掉晚到结果。

来源 [F25] [F28] [F29] [F31]

MetaGPT：用产物约束角色

MetaGPT / SOP 与文档交接

> MetaGPT 把团队协作从自由聊天转成角色之间交换需求、设计和代码等产物；收益来自交接结构，风险来自上游错误被格式化后继续放大。

结构示意 · 关系与数据流

节点与职责

1. 需求产物
2. 设计接口
3. 文件任务
4. 代码实现
5. 测试反馈
6. 版本与未决项

有向关系 · 后半句为条件或动作

- 需求产物 → 设计接口 · 约束
- 设计接口 → 文件任务 · 分解
- 文件任务 → 代码实现 · 限定范围
- 代码实现 → 测试反馈 · 执行验证
- 测试反馈 → 代码实现 · 具体错误
- 设计接口 → 版本与未决项 · 记录决策
- 测试反馈 → 版本与未决项 · 定位冲突

1. 角色通过标准产物协作

1.1 原始论文将产品、架构、项目管理和工程职责组织为 SOP；每阶段交付为下游可消费的文档，减少所有角色同时讨论全部问题的开销。这是受控软件生成研究的组织方式。[F32]

1.2 不能只学习“有产品经理、架构师”的命名。真正可复用的是每个角色的输入、输出、依赖和回流条件，否则多角色只会生成彼此引用的长文。

2. 写代码动作展示了交接细节

2.1 metagpt/actions/write_code.py 把 Design、Task、Legacy Code、调试日志和缺陷反馈分别填入模板，限定实现当前文件并遵循已有接口。生成后由 CodeParser 提取代码，返回 CodingContext。[F36]

2.2 该动作会读取测试输出和代码摘要作为下一次生成的上下文；解析失败可重试。但模板要求“完整代码”和“遵循设计”仍属提示约束，不能据此推导接口正确性已被程序证明。[F36]

3. 分析：格式权威与技术权威要区分

3.1 上游设计可能过时，写代码代理却被强烈要求不得改变设计。建议把“遵循已批准接口”与“发现矛盾必须上报”同时编码；为每个产物附版本、来源、未决问题和审批状态。

3.2 对 AOS 软件工厂，可以借鉴产物契约，让实现者、审查者和验证者消费同一需求版本；验收失败应能定位要修改需求、设计还是实现，避免简单回到上一角色重写整篇文档。

边界

图中“版本与未决项”是本报告提出的改进方向；MetaGPT 论文的研究效果不等于真实企业项目能无人值守交付，也不构成生产支持承诺。

来源 [F32] [F36]

MetaGPT：消息订阅驱动行动

MetaGPT / 共享环境与私有状态

> MetaGPT 并不是所有代理每轮读取一个无限大聊天框：环境负责投递，角色有私有接收缓冲和记忆，再依据关注的动作决定是否反应。

层次	保存什么	决定什么
Environment	角色与调试历史	消息投递和本轮调度
RoleContext	缓冲、记忆、关注集合	哪些事件值得处理
Action	当前动作及输入产物	输出消息和结果
Team	预算与轮数	何时停止和归档

1. 事件如何进入角色

1.1 Environment.publish_message 按收件地址调用角色 put_message，并把消息加入调试历史；环境一次 run 并发执行非 idle 角色。这里的环境是应用内协作容器，不等同于系统沙盒。[F34]

1.2 Role._observe 先消费私有 msg_buffer，再依据 cause_by 的订阅或 send_to 指定收件人过滤；已有记忆用于避免重复处理。是否保留所有收到的消息与是否触发行动可以分开配置。[F33]

2. 角色内部还有独立执行策略

2.1 Role.react 支持按序、ReAct 和 plan-and-act；ReAct 受 max_react_loop 限制，plan-and-act 先生成计划，再执行当前任务并处理结果。不同策略共享角色接口，却有不同的结束语义。[F33]

2.2 Team.run 在轮数、预算和全员空闲条件下停止，结束后归档；invest 设置成本预算，循环检查余额。预算检查发生在调度轮次边界，不应被解读成每个并发模型请求都有精确硬额度。[F35]

3. 分析：角色关注范围是一种降噪机制

3.1 对 Peri 的建议是让消息带明确事件类型、任务编号、产物版本与收件人；代理订阅真正影响它的事件。不是把全部转录广播出去，再要求每个模型自行忽略无关内容。

3.2 同时需要独立处理消息积压、失效订阅和循环触发：A 的产物触发 B，B 的评论再触发 A，可能形成没有新证据的往返。预算上限能止损，内容版本与进展检测才能解释为何停。

边界

本页所见接口主要说明应用内编排。没有据此证明持久队列、跨主机容错或分布式消息恰好处理一次；这些需要额外实现与故障注入测试。

来源 [F33] [F34] [F35]

ChatDev：从角色剧本到执行图

ChatDev 1.0 与 DevAll 2.0 / 范式演化

> ChatDev 的名字如今包含两代不同系统：旧版研究软件公司式对话，新版 DevAll 研究可配置执行图；比较前必须先固定版本和目标。

结构示意 · 关系与数据流

节点与职责

1. 任务归一化
2. ReAct 子图
3. 质量编辑
4. 最终结果

有向关系 · 后半句为条件或动作

- 任务归一化 → ReAct 子图 · 执行
- 任务归一化 → 质量编辑 · 只传上下文
- ReAct 子图 → 质量编辑 · 执行
- 质量编辑 → ReAct 子图 · 出现 TODO
- 质量编辑 → 最终结果 · 满足要求

1. 旧范式解决“怎么沟通”

1.1 ChatDev 1.0 论文用 chat chain 决定沟通主题，以 communicative dehallucination 组织协作，让不同角色在设计、编码、测试阶段通过对话产出软件。它是多代理软件生成研究，不能当作当前主分支架构。[F38]

1.2 当前 README 明确主分支为 ChatDev 2.0 DevAll；1.0 位于 chatdev1.0 分支。DevAll 面向通用多代理工作流，除软件外也覆盖研究和视觉产物，编码只是可配置场景之一。[F37]

2. 新范式把信息流和调度分开

2.1 DevAll 原始论文将边定义为激活条件、数据传播策略与调度策略的组合；一条边可以把信息交给下游，但不一定触发它执行。这样才能表达旁路上下文、条件回流和循环。[F39]

2.2 仓库 react.yaml 就给出实例：任务归一化输出交给 ReAct 子图，再到质量编辑者；归一化到编辑者的边设 trigger:false，只提供上下文。编辑者含 TODO 时才触发回到 ReAct 的边。[F43]

3. 分析：流程可视化必须显示语义

3.1 如果图上所有箭头长得一样，使用者会误以为每次传数据都会启动任务。建议 AOS 至少区分依赖、上下文传递、条件触发和人工验收，并显示本轮哪条边真的被激活。

3.2 示例里的关键词条件容易受正文措辞影响；生产任务应尽量返回结构化状态和缺陷清单。图能执行不代表流程正确，仍需验证无条件环、缺输入节点和输出未被消费的分支。

版本边界

该图重绘公开 react.yaml 的主要逻辑，最后的交付节点用于解释最终输出。它展示通用编排表达力，不证明某个软件任务的验证质量。

来源 [F37] [F38] [F39] [F43]

DevAll：循环必须有执行语义

ChatDev 2.0 / 环、记忆与产物

> 让图允许回边只是第一步；DevAll 用强连通分量识别循环，再在限定范围内迭代，提供了比“遇到失败重新来”更具体的实现样本。

机制	源码承担的职责	应用仍需定义
强连通分量	识别有环区域	是否允许这类业务回路
迭代上限	防止无限执行	耗尽后的人工或重试策略
记忆 attachment	限定读写与检索阶段	信息可信度及过期处理
运行产物文件	保留输出供检查	原子恢复与副作用去重

1. 环检测与环执行是两个模块

1.1 workflow/cycle_manager.py 用 Tarjan 算法发现强连通分量，自环也计入；CycleInfo 保存入口、出口、迭代计数与配置上限，未配置时使用默认上限。循环首先成为可检查的数据结构。[F40]

1.2 cycle_executor.py 在环作用域内构建拓扑层并处理嵌套环；只有出口边实际触发外部节点才返回下游，入口未再触发则结束，否则继续迭代，达上限记警告。这是调度语义，不是任务成功证明。[F41]

2. 记忆和运行产物另有接口

2.1 记忆文档把 store 与节点 attachment 分离：节点声明在哪个阶段检索、是否可读写以及返回数量。文件检索、语义记忆和只按时间取最近项的 blackboard 各有用途，不能一律称为长期推理记忆。[F44]

2.2 GraphContext 保存节点输出与子图状态，并写 node_outputs.yaml 和 workflow_summary.yaml。能查看运行产物，有助于复盘；这些文件本身不证明执行到中途崩溃后可无重复恢复。[F42]

3. 分析：编排引擎应报告为什么结束

3.1 建议区分条件满足、无可运行节点、迭代耗尽、工具失败和人工终止；将图级停止与节点级验收分开。否则用户只看到“工作流完成”，却不知道某条修复回路只是用完预算。

3.2 对 Peri/AOS，优先验证回边跨越人工步骤、并行分支共享文件、循环内更换产物版本等组合。图调度开销可以测毫秒级，但总体成本往往仍取决于模型调用次数与共享状态冲突。

研究边界

DevAll 论文重点是用同一平台复现不同工作流，并非证明全面超过专用 Coding Agent；框架表达力、生成质量和生产运维能力应分别验收。[F39]

来源 [F39] [F40] [F41] [F42] [F44]

框架能力不等于产品能力

跨产品比较 / 面向 Peri 与 AOS 的选择

> 比较多智能体系统时，应先看它交付的是可直接使用的编码产品、研究执行器还是通用编排框架，再判断哪些机制能进入自己的运行时。

对象	核心学习单元	最容易误判
Qwen Code	工具一致提示与团队协议	有 worktree 即全环境隔离
Trae Agent	证据式修复与补丁选择	默认 CLI 已自动多代理投票
Open SWE	线程交付与执行面分离	旧架构文档就是当前实现
AutoGen	消息、选择器、终止接口	多角色轮转即并行工程团队
MetaGPT	SOP 与文档交接	角色名本身带来正确性
DevAll	语义边与有环图调度	通用框架等于成品编码产品

1. 按责任边界比较，而不是数角色

1.1 Qwen Code 提供终端与多入口工程代理；Trae CLI 提供可检查的修复循环，另有实验性候选选择管线；Open SWE 增加线程、集成和交付后端。它们覆盖的产品责任并不相同。[F01][F09][F15]

1.2 AutoGen 提供通用代理与团队原语，且 README 宣布归档停维；MetaGPT 强调角色与 SOP；DevAll 强调执行图表达。能搭建编码团队，不能直接推导已有仓库授权、CI 门禁、可靠恢复和生产支持。[F25][F32][F37]

2. 值得分层吸收的机制

2.1 面向 Peri 核心，建议优先吸收提示词与工具一致性、类型化终止、子代理能力收窄和消息身份；面向 AOS，优先吸收任务归属、产物版本、工作空间和验证结果回流。

2.2 不宜把全部厂商的长提示简单拼接：研究场景的“候选必有正确答案”、本地目录假设、强制提交与通用回复协议都可能互相冲突。先写自身不可变契约，再选择适合的局部机制。

3. 用同一验收问题测试差异

3.1 建议准备长历史恢复、并行修改同一模块、用户中途纠偏、测试失败、工具结果丢失五类任务；固定模型、预算、仓库快照和授权，分别记录正确性与恢复成本。

3.2 本章给出源码与官方机制对照，没有运行跨产品基准，不能据此排序总体效果。真正可执行的下一步，是把候选机制各做成最小实验，并保留失败轨迹与反例。

Peri 启发

建议采纳顺序：先定义完成与权限契约，再修正工具/提示冲突，随后增强任务协作，最后引入复杂编排。排序是本报告分析，不是对 Peri 当前缺陷的事实断言。

来源 [F01] [F09] [F15] [F25] [F32] [F37]

Zed：把编辑器与代理运行时分开

Zed / ACP 与宿主边界

> 统一交互界面不意味着统一运行内核。外部代理保留自己的认证、模型和工具配置，宿主承担展示与协议适配。

结构示意 · 关系与数据流

节点与职责

1. 编辑器
2. ACP 适配
3. 代理运行时
4. 模型服务
5. 工具环境

有向关系 · 后半句为条件或动作

- 编辑器 → ACP 适配 · 请求/事件
- ACP 适配 → 代理运行时 · 能力协商
- 代理运行时 → 模型服务 · 推理
- 代理运行时 → 工具环境 · 执行

1. 已公开的设计

1.1 ACP 定义编辑器与代理间通信，本地形态为子进程上的 JSON-RPC/stdio。[G30] Zed External Agent 沿此边界连接；认证、模型和原生配置通常由代理负责，Zed 原生 profiles 与 skills 不自动控制它。[G01]

1.2 Agent Panel 展示流式工具状态，但历史恢复、检查点、token 显示是否可用取决于具体集成，统一 UI 不能抹去能力差异。[G02]

2. 对接层怎样才清楚

2.1 建议为每个代理保存能力矩阵：是否能恢复、是否能取消、是否转发 MCP、谁审批。建立连接时校验矩阵，界面只暴露双方共同支持的操作。

2.2 将用户身份、模型账单、文件权限分别标在所属边界。代理换供应商时，不应顺带继承上一代理的凭证或推定它读取相同的规则文件。

2.3 对接验收应覆盖连接中断：界面断线不等于代理退出。重连先查询原会话，再决定是否新建任务；若盲目重发写入请求，旧进程与新进程可能同时修改同一工作区。还需记录各自的最终结果。

Peri 启发

把 ACP 作为明确边界，并单独验收能力降级。协议接通只是起点，语义一致性需要合同测试。

来源 [G01] [G02] [G30]

Zed：并行线程与指令加载是两件事

Zed / 工作树与规则

> 独立会话解决上下文相互污染，worktree 解决文件版本分叉；项目规则的发现顺序则决定每条会话究竟收到什么。

对象	独立什么	仍需验证
线程	会话与模型上下文	相同文件是否并发写
worktree	分支与文件副本	端口/缓存/外部服务
规则加载器	指令优先级	原生与外部是否一致

1. 官方行为

1.1 Parallel Agents 将多个独立线程按项目组织，必要时使用 Git worktree。原生 Agent 的 steer 可在步骤边界介入，外部代理是否支持这类边界不同。[G02][G03]

1.2 Instructions 文档列出多个兼容文件，并使用首个匹配文件；项目规则与个人规则冲突时优先项目。外部代理仍可能使用自己的加载器。[G04]

2. 工程上应避免的混淆

2.1 同目录多线程不等于隔离执行。即使使用 worktree，端口、数据库、构建缓存、环境变量仍可能共享，必须由环境配置说明真实隔离范围。

2.2 兼容多个规则文件时，要展示加载来源与被遮蔽文件。否则用户以为 AGENTS.md 生效，实际命中了更早的兼容文件，调试会错误归因于模型。

2.3 建议为每条线程标出工作区身份、基线提交与当前写入者。跨线程复用测试结果时验证代码和环境相同，不能仅凭任务名称一致就把旧通过记录附到新分支。

验收问题

在同一项目同时启动原生和外部代理，记录各自实际加载的规则文件、有效权限和工作目录。

来源 [G02] [G03] [G04]

Replit：把提醒放到决策时刻

Replit / Decision-time guidance

> 稳定核心提示词配合短暂、按场景触发的提醒，可降低长任务中的无关规则负担；提醒本身仍是模型建议。



1. 工程说明中的机制

1.1 Replit 描述用轻量分类器观察近期轨迹，只注入当下相关的小段指令；诊断信号提示代理主动取日志，循环失败时提示咨询另一个代理。提醒不永久写入历史。[G05]

1.2 该文是厂商经验说明。触发准确率、跨模型迁移效果和运行成本没有在本报告环境复现，不能把文中的收益百分比直接用于 Peri 预算。

2. 可落地的实验设计

2.1 先选三个可检测失败：重复相同工具错误、连续补丁回退、任务已完成却继续扩张。每个信号只绑定一个短指令，记录触发理由与过期时间。

2.2 与静态提示词对照时固定模型、任务和总预算，测成功率、重复动作率及用户指令被忽略次数。若提醒增加调用却没有提高通过率，就收缩触发范围。

2.3 分类器也应有误报与漏报账本。重复报错可能来自外部服务故障，提醒“换一种方案”未必有效；先定位失败类别，再限制提醒次数，避免注入器自身制造新的循环。

边界

图的结果记录与反馈属于本报告评估设计，不是已公开的完整内部循环。“提示尽量使用子代理”可以作为实验；权限、预算、终止条件仍必须由运行时执行。

来源 [G05]

Replit：委派能力拆成可组合选择

Replit / 可复用子代理与能力档位

> 委派是否有价值，取决于任务切分、上下文回报和重新唤醒的成本。把选择做成少量工具参数，便于模型按任务取舍。

选择	调度价值	失败风险
专长	限制工具和输出噪声	角色说明相互重叠
档位/强度	按难度分配成本	错误低配导致返工
复用	节约重新解释成本	旧状态与新代码错位
不委派	避免小任务调度开销	漏掉独立验证机会

1. 当前公开设计

1.1 2026 年 9 月工程文描述了专长子代理、能力档位、推理强度以及返回既有子代理的机制。主循环决定何时委派、交给谁，小任务可以完全不委派。[G06]

1.2 这证明了一种公开设计方向，不能证明任意模型都有相同调度能力。文中比较的是特定模型、任务集与运行配置，本报告不据此给出跨产品性能排名。

2. 如何借鉴而保持可解释

2.1 将 role、task、context_refs、effort、resume_id 做成清晰参数，输出同时给出 task_id、成本归属、状态和产物引用。不要用一个自然语言参数暗含所有调度行为。

2.2 复用子代理前先验证它的文件基线与权限是否仍有效；上下文保留不能成为过期假设的缓存。任务失败时保留失败证据，但下一次咨询可以重新构建干净上下文。

Peri 启发

先把委派的上下文包与返回格式做稳，再调优“何时主动派发”。仅改鼓励性文字无法解决接口歧义。

来源 [G06]

Replit：完成与应用必须分开

Replit / Task board

> 后台任务完成的是一个候选结果。把 Ready 与真正应用到主版本分开，有助于并行实现保持可审查性。

建议交付阶段	需要保存的证据	进入下一阶段条件
计划/运行	目标、依赖、副本与日志	执行终止
Ready 候选	diff、自测与风险	请求应用/单次自动应用
Applying 应用	目标版本、冲突处理记录	形成集成结果
验收（另设）	实际受测代码树与结论	接受或返工

1. 产品契约

1.1 任务在项目副本中执行；Ready 是候选完成，Applying 才向主版本应用。官方称自动处理多任务冲突；task board 可单次 auto-apply，跳过等待人工审核，不是持续开关。[G07][G37]

1.2 Done 同时容纳已应用、归档或取消的任务，所以它是产品视图中的归档集合，不能直接解释为所有任务验收通过。

2. 编排设计推演

2.1 建议运行状态与交付状态分列。运行成功、补丁可合并、测试通过、人工已接受分别记录，避免把模型说“完成”升级为生产就绪。

2.2 合并时必须重新绑定当前主分支，验证依赖提交与产物摘要。两个独立通过的补丁可能在组合后失败，最终验收针对合并结果而非两个历史测试报告。

2.3 任务板卡片建议显示结果依据：完成原因、应用提交、验证版本和遗留问题。取消任务可正常归档，但不能累加进验收成功数；人工撤销已应用变更也应形成独立事件。

AOS 启发

官方未在所读页面公开合并算法、原子性或最终集成版本测试绑定；表中独立验收为设计建议，不写成产品已实现保证。Done 也不等于验收成功。

来源 [G07] [G37]

Antigravity：用产物承接异步协作

Antigravity / Artifacts

> 当任务跨越许多工具步骤，用户最需要审阅计划、diff 和验证证据。产物应成为明确的交接对象。



1. 文档可验证能力

1.1 官方 artifacts 包括结构化计划、代码差异、架构图、图片和浏览器录屏；审阅面向里程碑，用户可对计划给出内联反馈。[G08]

1.2 子代理文档说明独立上下文、继承父级安全设置和将审批请求上浮到主界面。实际隔离与授权能力仍取决于所选运行方式。[G09]

2. 对产物模型的启发

2.1 建议每个产物附上任务 ID、版本、生成依据与待决策项。用户批注应引用精确版本及范围，避免计划已重写而旧批注仍影响新执行。

2.2 录屏能证明某次操作发生，但不能独立证明所有验收条件成立。将浏览器观察、自动测试和人工结论关联到相同构建版本，才能支持回退与前进判断。

2.3 产物需要草稿、待审阅、已接受和已过期等版本语义。后续实现偏离已审计划时，比较具体差异并重新关联验收条目；不能把用户曾确认旧计划解释为接受未来任意实现。

Peri/AOS 启发

图的版本绑定与同一构建验收是本报告设计建议，不是产品强制流程。 把聊天流与产物版本分开存储。会话可以压缩，验收依据仍然能够独立查回。

来源 [G08] [G09]

Antigravity：钩子与沙箱各守边界

Antigravity / Hooks 与执行控制

> 钩子能按语义批准、拒绝或补充上下文；沙箱限制命令实际触及的资源。两者不能互相替代。

控制层	解决什么	不能推定什么
提示词	解释操作意图	保证模型必定遵守
执行前钩子	语义检查与拦截	自动形成 OS 隔离
沙箱	路径/网络/进程边界	判断业务操作是否正确

1. 公开接口

1.1 Hooks 文档定义模型调用与工具调用前后的生命周期点；PreToolUse 可给出 allow、deny、ask 等决策。它属于运行时契约，强于仅在提示词里写禁止事项。[G10]

1.2 终端沙箱在 Linux 使用 namespaces、在 macOS 使用 Seatbelt，Windows 的设置行为另有说明。平台差异意味着同名安全模式需要逐系统验证。[G11]

2. 可复制的工程原则

2.1 工具授权记录应包含最终参数，而不是只保存模型最初提出的参数；钩子若允许重写参数，必须在实际执行前对修改后的请求重新确认有效策略。

2.2 钩子失败、超时与权限服务不可用需要明确结果；不能把“未返回拒绝”自动当成允许。异步日志钩子可以降级，决定写入许可的钩子则应遵循更严格失败语义。

2.3 多个钩子共同参与时，应固定执行顺序、决策优先级与超时预算，并记录最终依据。钩子自身调用其他工具也要受同一策略约束，否则审计只能看到表层工具。

验证

分别测正常、拒绝、超时、参数重写和取消五条路径，并核对日志中最终执行的命令。

来源 [G10] [G11]

Warp：开源客户端与云端编排

Warp / 客户端与云端分层

> 值得学习的是把环境、会话可观察性和任务入口放到平台层；不能从客户端开源推导云端调度全部透明。

结构示意 · 关系与数据流

节点与职责

1. 触发入口
2. 任务平台
3. 环境分配
4. 代理内核
5. 会话审阅

有向关系 · 后半句为条件或动作

- 触发入口 → 任务平台 · 任务
- 任务平台 → 环境分配 · 配置
- 环境分配 → 代理内核 · 运行
- 代理内核 → 会话审阅 · 事件/产物
- 会话审阅 → 任务平台 · 跟进

1. 当前状态与能力范围

1.1 当前 Warp 仓库 README 已明确客户端开源：UI 框架采用 MIT，其余仓库代码为 AGPL v3。旧资料中“Warp 全闭源”的结论需要更新。[G13]

1.2 当前 /oz 官方页面以 Warp Factories 展示云端编排，列出 Warp、Claude Code、Codex 等 harness、团队会话观察与自托管选项；记忆和自改进入口仍带等待名单，不视作全量可用能力。[G12][G14]

2. 平台化的代价

2.1 跨内核调度需要统一最小任务契约，而非统一所有提示词。任务状态、文件副本、日志、取消结果和产物引用应保持一致，模型专有功能通过能力协商开放。

2.2 允许本地接管时，必须说明接管的是终端连接、同一会话还是复制后的新会话。若不能确认所有权交接，就可能出现两个控制者同时发送修改命令。

2.3 跨内核平台应保留实际模型、工具版本与运行环境摘要。统一显示“成功”会隐藏内核不同的停止语义，接入测试需要覆盖部分结果、预算耗尽、连接丢失与明确失败。

研究边界

本页核实客户端许可与官方功能契约；/oz 页面已使用 Factories 品牌，未审计完整客户端或云端调度实现。图中统一任务平台为分析抽象。

来源 [G12] [G13] [G14]

Refact：常驻控制面与热项目工作进程

Refact / 本地引擎的资源分层

> 低启动开销可以来自复用项目服务，同时也带来生命周期、回收和权限隔离问题。常驻不等于免费。

建议生命周期	适合持有	建议回收信号
会话	任务、上下文、授权	关闭或归档
项目 worker	索引、解析、项目缓存	空闲/版本变化
daemon	注册表、监管、路由	显式关闭/监管退出

1. 版本定位

1.1 smallcloudai/refact README 已标为旧归档，开发转向 JegernOUTT/refact，并说明 Refact Cloud 退休。当前仓库声明本地 daemon 监管项目 worker，服务终端、浏览器与 IDE 客户端。[G15][G16]

1.2 当前维护仓库 README 将 daemon 与 TUI 标为 early preview；其 task planner 和 fleet 描述了工作树及依赖任务卡。这里核实的是维护者声明，尚未独立验证调度器的崩溃恢复。[G16]

2. 对 Rust Agent 内核的启发

2.1 可将模型会话的短生命周期与索引、语言服务等较重项目资源分离。资源键应包含项目身份与配置代际，避免相同路径被不同权限会话错误复用。

2.2 需要测热 worker 的空闲内存、最大驻留数、最后引用释放时间和异常重建。若用户切换项目后仍保留所有历史工具输出，热启动收益会被常驻内存吞没。

2.3 建议让客户端退出、会话关闭、工作进程回收和 daemon 停止成为独立动作。一个编辑器窗口关闭不应无条件终止其他客户端的任务，仍有任务时的回收策略须可观察。

建议实验

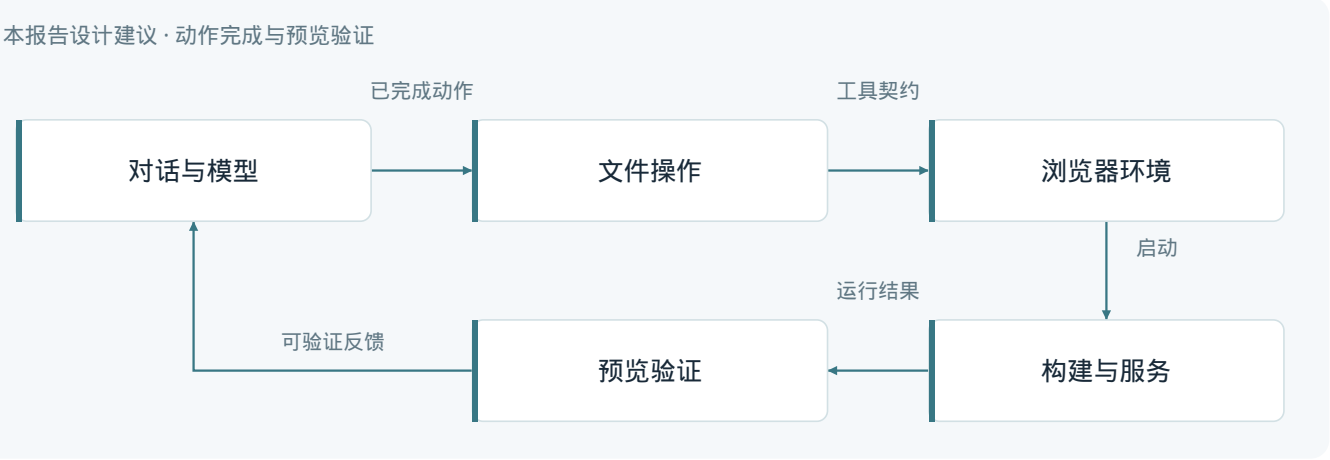
记录冷启动、热启动、100 次会话后常驻内存与回收延迟；不要只看单次首 token 延迟。

来源 [G15] [G16]

bolt.diy：浏览器中的执行与预览闭环

bolt.diy / WebContainer

> 浏览器可以同时承载文件编辑、终端执行和实时预览。设计重点是运行环境能力边界和输出验证。



1. 官方能力

1.1 官方文档将多模型接入、MCP、WebContainer 终端与实时预览结合。执行环境面向 Node.js 及常见前端框架，文档级“全栈”不能理解为任意宿主程序都能执行。[G17]

1.2 本页没有对其完整提示词解析器和流式 action runner 做固定提交审计，因此不把前端产品宣传当作协议正确性结论。

2. 浏览器 Agent 需要额外回答的问题

2.1 文件系统属于运行环境，Agent 通过工具操作它。应分别定义标签页关闭后的保存、构建进程恢复、后台资源限制和不同项目之间的文件隔离。

2.2 流式生成中的代码块不能在语法尚未闭合时随意执行。将可执行操作转成有 ID、明确完成信号的动作，先落盘再触发构建，错误消息回到同一任务版本。

2.3 预览页应绑定构建序号与输入文件版本，避免旧服务仍可访问时被误判为新改动已生效。刷新、构建失败与进程退出分别展示，并把验证记录关联到实际打开的预览版本。

Peri 浏览器启发

图为本报告设计建议；未审计流式 action runner 的完成协议。 把 Agent 核心和持有文件系统的工具环境分离；需要实测环境支持的语言、进程和持久化语义。

来源 [G17]

Kimi：迁移时保留任务，重建授权

Kimi Code CLI / 新旧实现边界

> 项目名称相似并不代表实现连续。迁移时哪些内容继承、哪些授权重做，往往比兼容一个命令更重要。

迁移对象	应怎样处理	原因
任务目标/历史	可转换并保留来源	维持工作连续性
工具与插件	重新解析/验证	接口和行为会变化
凭证与授权	重新建立有效授权	账号、范围与有效期需核验

1. 当前官方状态

1.1 Python Kimi CLI 仓库已宣布归档，由 Kimi Code CLI 接替；旧仓库不再维护。新项目 README 描述单二进制分发、coder/explore/plan 子代理、hooks 和 ACP。[G18][G19]

1.2 迁移说明允许导入配置、MCP 服务定义、输入历史和会话，凭证、MCP 授权及旧插件不迁移。本报告将其解读为工作连续性与授权连续性的分离；实际需重新登录和授权。[G18]

1.3 包名也会误导：PyPI 的 kimi-code 仍是旧 CLI 别名；kosong、pykaos、kimi-sdk 一并归档。新 CLI 的 npm 包为 @moonshot-ai/kimi-code，不能只凭相同命令名判断运行实现。[G36]

2. 对协议与会话升级的启发

2.1 会话恢复不应只反序列化文本。恢复时要重建工具注册、验证路径、核对模型适配与有效授权；旧能力已经消失时，要显式说明降级。

2.2 格式可以不兼容，但历史应能导出；迁移器报告对象映射和跳过原因，重复迁移不重复建对象。迁移完成与新环境可执行分别验收，后者仍取决于模型、项目路径和重新建立的授权。

研究边界

本页核实新旧 README 与迁移边界，不把旧 Python Soul/Wire 架构误写为新 CLI 内部实现。

来源 [G18] [G19] [G36]

Augment：让上下文成为可替换服务

Augment / Context Engine 与 Cosmos

> 检索服务与任务执行器可以分开演进。关键不只是找得到代码，还要知道检索结果属于哪个版本与权限范围。



1. 官方公开契约

1.1 Context Engine MCP 的本地模式索引当前工作目录，远端模式索引选定仓库的默认分支。对尚未合并的改动，两者天然可能返回不同事实。[G20]

1.2 Cosmos 的产品说明把专家、执行环境、触发器与审核配置分开；公开 MCP 插件不等于检索服务内部算法或云端编排器完全开源。[G21][G22]

2. 设计价值与验证方法

2.1 建议检索响应带 repository、revision、path、range、index_time，并在读取最新文件后才生成补丁。用默认分支的语义结果修改工作树前，必须处理版本偏差。

2.2 评估应包含跨仓库调用链、未提交文件、刚重命名符号和权限撤销四类任务。只统计召回的文本数量，无法判断模型是否拿到了可执行的正确证据。

2.3 检索不命中与没有权限应保留可区分的诊断。授权撤销或仓库删除后，旧索引及模型缓存的使用规则也要更新，不能把曾经获得的片段视作永久有效的代码事实。

Peri 启发

revision/index_time 等响应字段为设计建议，不是已核实的接口保证。 把检索做成可测量、可替换的工具服务；Agent 核心只依赖证据契约，避免绑定某种向量索引。

来源 [G20] [G21] [G22]

Factory Droid：让角色成为工具边界

Factory / Custom droids

> 角色定义同时约束提示词、模型和可调用工具，才能使“只读审查员”等名称对应真实执行能力。

配置维度	作用	独立验收
提示词	说明任务和输出	冲突/冗余规则
工具集合	限制可调用能力	禁止工具实际不可用
自治级别	决定是否需确认	不能突破组织上限
生命周期	后台/续跑/取消	状态与结果可追踪

1. 官方接口

1.1 自定义 droid 使用 Markdown 定义，Task 支持前台、后台和 resume。子代理非交互，不能使用 AskUser 或继续调用 Task；阻塞时向父级报告。[G24]

1.2 工具策略与自治级别是不同维度。自治级别受组织上限约束，命令规则、工具可见性和沙箱仍独立发挥作用。[G25]

1.3 后台任务返回 task_id，以 TaskOutput 收集结果；TaskStop 文档说明先发 SIGTERM，必要时再发 SIGKILL。这描述进程停止方式，不证明外部服务已回滚。resume 会对齐父级当前自治级别。[G24]

2. 为什么这种分层值得学

2.1 定义“审查”角色时，禁止修改不能只写在正文里。工具集合应在加载时校验，未知工具与拼写错误应明确报错，避免默默回退到全部工具。

2.2 父任务改变权限后，续跑子任务应重新计算有效权限。任务先前获得过写权限，不代表它在新的只读阶段仍可继续沿用。

边界

Fresh context 是首次调用的上下文特性；resume 明确保留既有子任务上下文，不能一概描述为每次都全新。

来源 [G24] [G25]

Factory：按锚点增量压缩

Factory / 任务级上下文成本

> 压缩应保留目标、变更轨迹和可重新取回证据的线索；节省单轮 token 不一定节省整个任务成本。

本报告设计推演 · 检查点与视图切换

节点与职责

1. 原始事件
2. 待压缩区间
3. 摘要候选
4. 提交检查点
5. 活动窗口

有向关系 · 后半句为条件或动作

- 原始事件 → 待压缩区间 · 按游标
- 待压缩区间 → 摘要候选 · 增量摘要
- 摘要候选 → 提交检查点 · 验证/提交
- 提交检查点 → 活动窗口 · 切换视图
- 原始事件 → 活动窗口 · 最近尾部

1. 官方工程设计

1.1 Factory 的压缩文章描述持久化锚定摘要：只处理新移出的历史区间，再更新已有摘要；使用触发与保留两个阈值，避免每轮重新总结整个前缀。[G23]

1.2 这篇 2025 年文章表达的是当时设计，不能证明当前全部实现细节不变；它仍可作为明确的算法模式研究。

2. 对长会话的改进推演

2.1 建议原始事件存储、模型活动窗口和界面历史分页各自独立。摘要提交成功才移动活动窗口；摘要失败或进程崩溃时，旧窗口仍然可恢复。

2.2 摘要应携带目标、未完成任务、关键决定、产物引用与压缩覆盖游标。恢复时验证游标连续性，避免把同一段历史重复追加或跳过工具结果。

2.3 评估加入重新读文件次数、错误结论率与任务总耗时。若更激进压缩引起大量回读，单次请求更短并不意味着总体更优。

Peri 启发

图的检查点验证、提交与切换顺序为本报告推演，不是已公开的厂商事务协议。 内存回收发生在引用释放和视图切换之后；仅把新的摘要写入 transcript 并不能证明旧对象已释放。

来源 [G23]

Amp：专长子代理与外部工作空间

Amp / 模型分工与 Orbs

> 把搜索、复杂推理和资料研究分开，可以让主上下文更干净；不同协作机制必须分别描述能力。

对象	适用任务	必须补充的契约
搜索子代理	读大量文件后定位	路径、范围、检索限制
推理顾问	挑战方案与定位难点	前提、反例、验证方法
云端线程	长程实现和运行	环境、恢复、接管与合并

1. 当前文档界定

1.1 Amp 专长子代理包括 Search、Oracle、Librarian、Task 与 Read Thread；各有上下文，主代理只收最终总结。该文明确它们不能彼此通信，也不能中途被用户引导。[G26]

1.2 Orbs 是按线程创建的机器；另有 Agent-to-Agent 机制，可启动独立线程并回报结果、显式传文件。消息不会自动同步文件或提交，因此不能把专长子代理的限制推广到全部协作。[G27][G34]

2. 可借鉴的接口取舍

2.1 只返回总结可以降低噪声，但总结必须带证据定位和未解决问题。否则父代理获得一个看似确定的结论，却无法检查它是否读了正确文件。

2.2 不能中途引导的子任务应尽量短、边界明确。较长实现任务适合具备可观察状态、可追加要求与可取消能力的独立工作空间，二者可以共存。

Peri 启发

把“咨询”与“执行任务”分成不同调用契约，不必要求所有委派都拥有同样的生命周期。

来源 [G26] [G27] [G34]

Junie：把规则发现过程做透明

JetBrains Junie / Guidelines 与 Skills

> 常驻规则和按需技能分开加载，可以减少提示词膨胀；优先级、去重和来源可见性决定规则是否可调试。

结构示意 · 关系与数据流

节点与职责

1. 全局规则
2. 项目规则
3. 任务语义
4. 技能目录
5. 有效上下文

有向关系 · 后半句为条件或动作

- 全局规则 → 有效上下文 · 默认约束
- 项目规则 → 有效上下文 · 优先/去重
- 任务语义 → 技能目录 · 相关性
- 技能目录 → 有效上下文 · 按需材料

1. 官方加载规则

1.1 Junie 优先查找项目 .junie/AGENTS.md；其余回退路径可合并根 AGENTS.md、playbook 与 rules。全局与项目规则有明确冲突优先级，相同内容可去重。[G28]

1.2 Skills 以任务相关性按需加载，与每次任务都携带的 guidelines 分开。description 是发现技能的重要依据，而不是无意义的标签。[G29]

2. 提示词工程不只发生在正文

2.1 建议显示实际命中规则、被覆盖规则、token 预算与触发原因。规则文件越多，越需要解释装配结果；只提供“支持 AGENTS.md”不足以帮助用户定位冲突。

2.2 将高频稳定约束保留在核心层，把流程细节和示例放入按需材料。用成对任务测试误触发与漏触发：同关键词但不需该技能、无关键词但确需该技能。

2.3 规则发现可用目录树样例验收：同时存在多种文件、子目录覆盖、文件重命名与任务切换。记录预期命中清单和真实装配结果，再判断错误来自路径查找、优先级还是模型遵循。

验收

规则装配器需要可观测输出和冲突诊断。把所有规则无差别拼接，会将工程问题转交给模型猜测。

来源 [G28] [G29]

统一架构：把概率决策包在确定边界内

横向综合 / 六层职责

> 模型负责提出下一步，运行时负责确认这一步能否发生、如何记录以及失败后如何恢复。两者的职责要能单独测试。

结构示意 · 关系与数据流

节点与职责

1. 指令与上下文
2. 模型决策
3. 执行与授权 · 启动期 + 调用期
4. 事件与状态
5. 协作与交付

有向关系 · 后半句为条件或动作

- 指令与上下文 → 模型决策 · 有效输入
- 模型决策 → 执行与授权 · 工具意图
- 执行与授权 → 事件与状态 · 提交结果
- 事件与状态 → 指令与上下文 · 投影
- 事件与状态 → 协作与交付 · 任务/产物

1. 建议的职责拆分

1.1 指令装配层产生明确的模型输入；上下文层选择当前可见证据；代理循环只负责模型调用与意图解释。权限、工具执行、任务状态和持久化不应藏在提示词文本中。

1.2 协作层持有任务图、父子关系、消息与预算；交付层持有补丁、测试记录和验收结论。一个会话可以结束，但它生成的任务与产物不应随 UI 消失。

2. 借鉴方式

2.1 先为现有系统画出每个状态的唯一写入者，再决定是否抽象接口。相同字段被代理、前端、工具服务同时修改时，增加角色数量会放大竞争问题。

2.2 本图是本报告提出的参考分层，不是任何一家产品的完整内部结构，也不代表 Peri 当前已经如此实现。可优先围绕停止、权限和恢复边界作最小验证。

设计原则

提示词可以改变策略偏好；确定性的安全与一致性不变量应由运行时保障。

来源 [H02]

提示词装配器：把加载过程变成可审计结果

横向综合 / Prompt composition

> 比一段更长的系统提示词更重要的是明确加载顺序、作用域、有效工具和失效时机。每次调用都应能解释输入来自哪里。

输入层	保留位置	失效条件
稳定核心	缓存前缀	版本变更
项目/角色	作用域配置	项目或任务切换
技能/证据	按需上下文	任务无关/证据过期
临时提示	当前调用覆盖层	下一决策或明确 TTL

1. 六种输入分别管理

1.1 稳定核心规定目标、工作原则与输出标准；项目规则带路径作用域；技能按相关性加载；角色约束只作用于对应子任务；工具说明来自真实注册表；临时提示有触发条件与过期时间。

1.2 每段记录来源、作用域、优先级、版本与 token。工具还应绑定可信服务身份、定义摘要、授权作用域和读取时点；跨服务同名或自报 serverInfo.name 不是身份，旧批准不自动覆盖新定义。[H01][H02]

2. 先评估装配，再评估模型

2.1 固定输入集合，验证装配顺序和去重是否符合契约；再测模型是否遵循。若不存在的工具名称仍出现在提示中，先修装配器而不是要求模型更聪明。 声明目录与间接可调用集合需分别快照；审批身份、授权过期和提示正文保留也不能共用一个“已加载”标记。[A04][B60]

2.2 临时提醒可借鉴 Replit 的按需机制，规则发现可借鉴 Junie/Zed 的明确路径优先级；具体优先级应由 Peri 自己定义并稳定下来。[G04][G05][G28]

交付件

有效提示词清单、冲突诊断和同一输入可重放的装配测试，比人工维护一个超长模板更可持续。

来源 [G04] [G05] [G28] [A04] [B60] [H01] [H02]

工具设计：名称、说明、策略必须一致

横向综合 / Tool interface

> 模型选择工具时看到的是名称与描述，运行时执行的是 schema 与策略。两边表达不同，就会制造委派失败和权限误解。

不清楚的接口	建议明确的语义	验收观察
执行子任务	创建新的任务实例	返回稳定 task_id
通知代理	只投递 / 或投递并唤醒	状态变化和消息回执
停止任务	取消请求与完成确认分开	已停止/失败/远端未确认
等待	观察状态与消费结果分开	重复等待不重复应用

1. 工具注册表应表达的内容

1.1 注册表应声明前提、类型、副作用、幂等、取消与权限，但声明不等于已验证行为。MCP 不可信服务的 annotations 不能直接作为只读、免批或安全重试证明；可信且受限部署可将其作为策略输入。[H01]

1.2 spawn、send、followup、wait、cancel 不应互相暗含。发送消息是否启动空闲代理、wait 是否消费消息、取消能否中止远端操作，均须显式定义；取消请求已受理不能等同于所有资源已退出。

2. 对工具冲突的实验

2.1 制作任务对照集：只需并行读取、需要独立上下文、需要改文件、需要恢复旧子任务。记录模型选择、无效调用、重复调用与额外成本。

2.2 工具缩减要同时测直接、Code Mode、MCP 与子任务入口；Hook 还应声明 can_block、can_mutate、on_error。用“策略拒绝 + Hook ask”“先批后改参”“Hook 故障”验证最终决定，不将这些建议写成已运行的 Peri 测试。[A40][B62][C60]

Peri 启发

建议把目录文本视为外部输入：同名跨服务、定义变更和凭据切换后重验批准。工具不被调用也可能通过目录影响别的调用；这是官方威胁模型，未对产品攻击复现。[H02]

来源 [G24] [G25] [A40] [B62] [C60] [H01] [H02]

为什么 Agent 有工具却不主动委派

横向综合 / 选择机制与干扰

> “不喜欢派 subagent”不应先归因为人格。委派工具可见性、相互冲突的指令、上下文成本与回报格式都可能改变选择。

观测	可能原因	下一步排查
从不派发	工具不可见/规则压制	查看有效提示词
派发后返工	任务范围或证据不足	检查输入输出合同
频繁重复开工	无法续跑/找回结果	增加状态检索
成本高但无增益	拆分不独立	衡量合并与协调开销

1. 待验证的五类原因

1.1 策略层：存在“自己完成”“不要等待”等上位规则；发现层：角色描述过泛、加载时机错误；接口层：无法知道子任务能否编辑或恢复；经济层：派发成本大于直接完成；能力层：当前模型对任务切分较弱。

1.2 Replit 的公开经验表明不同模型会采用不同的委派模式；这支持按模型测量，而不支持预设“任何任务都应并行”。[G06]

2. 最小对照方案

2.1 使用相同任务、模型与预算，比较无提示、短委派说明、结构化任务工具、执行反馈四组。记录是否派发只是中间指标，最终比较验收通过率和总成本。

2.2 对明显可拆任务检查是否提供独立输入与返回合同；对小修复允许不派发。若强制派发只增加串行等待，就应修工具契约或缩小适用范围。

成功标准

在值得拆分的任务上更常委派，同时保持小任务速度。委派率本身不应成为绩效目标。

来源 [G06]

委派提示词：交付合同先于角色扮演

本报告原创模板 / 可直接改写使用

> 一句“你是资深工程师”提供的约束很少。子任务要清楚知道目标、范围、输入、权限、返回内容与停止条件。

合同字段	实现任务	研究任务
完成定义	行为+验证	结论+反例+来源
写入范围	指定工作树与文件	默认只读
返回材料	补丁/测试记录	证据索引/不确定性
终止	成功/失败/取消/预算	足够证据或明确缺口

1. 实现型子任务模板

1.1 目标：在给定基线实现指定行为；范围：只改列出的模块，遇到跨边界依赖先报告；输入：任务说明、基线提交、相关文件与已确认决定。不要把未见过的历史当成事实。

1.2 权限：角色文字不授予能力；工具目录仅表示可请求入口，实际权限还由执行门禁、服务凭据及进程/网络边界限定。[H02] 验证须记录实际结果，交付变更/补丁引用和剩余风险。

2. 研究型子任务模板

2.1 独立阅读给定问题涉及的证据，不接受父代理结论作为前提。每个结论给出来源、成立条件、最强反例及可复现检查；无法确认时写明未知。

2.2 返回结论与证据定位，不回传整段原始日志；发现范围外重大问题先说明影响，再等待父级决定。任务无法继续时报告阻塞，不伪装完成。

使用说明

以上为本报告原创示例，既不是任何厂商完整系统提示词，也不是已经写入 Peri 的默认规则。

来源 [H02]

多智能体拓扑：按依赖选择协调方式

横向综合 / 拓扑与成本

> 多智能体并不只有“组建团队”一种形态。真正应该选择的是信息怎样流动、谁能写入、何时汇总和谁确认完成。

形式	最适合	关键代价
主从委派	可独立的子问题	分包与汇总损耗
阶段流水线	明确产物依赖	阶段等待与交接
共享任务板	多个长程任务	状态一致性与合并
独立对抗	关键假设审查	额外成本与裁决

1. 四种常用形式

1.1 主从委派适合独立检索或局部实现；计划-执行流水线适合前置产物稳定后进入下一阶段；共享任务板适合多个任务长期推进；独立对抗适合验证关键设计假设。

1.2 这里的分类是本报告分析框架，不是产品成熟度排名。一个产品可以同时采用多种形式，工具模式切换也不自动构成多代理。

2. 选择标准

2.1 把依赖写成数据：等待同一 API、编辑同文件、依赖同服务，都限制并行。Anthropic 2025 年 Research 经验也指出编码较少真正可并行部分；其研究评测收益不能当编码提速证明。[H06]

2.2 让每种拓扑拥有专属停止条件：主从等待必需结果，流水线检查产物版本，任务板按依赖完成，对抗评审达到证据门槛。不要一律以“所有代理说完成”为结束。

决策

先画依赖再选代理数。独立复核描述的是证据与判定边界，不是人数保证；多个裁判共享错误前提仍可一致。此为待测试反例，不是本轮测得的误判率。

来源 [H06]

消息系统：送达、读取与应用分开

横向综合 / 持久收件箱

> 协作可靠性不应依赖某条实时通知恰好被收到。事件落盘、唤醒和消费需要分别可验证。

结构示意 · 关系与数据流

节点与职责

1. 发送者
2. 持久收件箱
3. 唤醒调度
4. 接收者
5. 状态提交

有向关系 · 后半句为条件或动作

- 发送者 → 持久收件箱 · 先保存
- 持久收件箱 → 唤醒调度 · 再通知
- 唤醒调度 → 接收者 · 恢复执行
- 持久收件箱 → 接收者 · 按游标读
- 接收者 → 状态提交 · 本地事务应用
- 状态提交 → 持久收件箱 · 消费回执

1. 推荐消息合同

1.1 消息包含 message_id、task_id、sender、recipient、epoch、sequence、payload_ref 与创建时间。发送成功只说明系统接受了消息，不自动说明接收者已理解或应用。

1.2 建议分列 accepted、delivered、consumed、applied。去重键绑定租户/收件人、任务代际及规范化参数摘要；同键不同语义应报冲突，不能静默当成功。operation_id 另管业务副作用，大产物只传引用。

2. 崩溃与重试

2.1 消息落盘后再唤醒，恢复扫描补偿通知丢失。本地状态更新与消费游标在同一事务提交，可防止本地重复应用；这不能原子覆盖远端 API、shell 或另一个数据库的副作用。

2.2 外部操作先持久化意图，优先携带服务端幂等键，再记录结果；“执行成功但回执丢失”进入待对账状态。无查询或去重能力时保留未知，不能凭消费游标或模型总结声称恰好执行一次。幂等键的保存期限也应覆盖允许重试的时间窗，避免键过期后重放旧命令。

故障注入

域外对照：SQS FIFO 去重窗内同 ID 可接收成功却不投递，内容去重哈希不含 attributes；Stripe v1 同 key 不同参数报错。[H07][H08] OpenCode 父输入接纳也不是验收。[B66] 建议测试同键不同 payload/epoch、回执丢失及未知结果对账。

来源 [B66] [H07] [H08]

停止后的唤醒：任务状态必须可区分

横向综合 / 长程执行与收尾竞态

> 一轮模型停止、任务结局与是否允许唤醒是不同维度。分别持久化后，才能解释续跑、取消和迟到结果。

运行/任务条件	到达事件	建议处理
等待依赖	同代际结果	更新依赖并继续
空闲或本轮结束	获准跟进	新轮次；保持任务身份
已请求取消	迟到结果	归档；记录在途操作结果
新任务代际	旧任务结果	拒绝应用或人工关联

1. 推荐状态语义

1.1 建议分列运行态、任务结局、cancel_requested 与续开策略，并记录 cancel_actor、run/attempt、reply_to。Claude 用户取消与模型停止的再唤醒不同；Cline run 取消也不等于成员已停止。[A05][C65]

1.2 子结果到达时检查父任务 epoch、依赖和取消策略，决定是否新开轮次；不能由过期结果复活未经授权的工作。已持久成功与晚到取消的优先级另行定义，不能抹掉既成成果。[B46]

2. 收尾窗口的检查

2.1 在“检查队列为空”和“标记停止”之间可能有新消息到达。用同一状态提交或代际重检消除丢唤醒窗口，并保留可再次触发的持久事件。

2.2 ACP 通用 $/cancel_request 的级联为可选；session/cancel 则要求相关操作中止后以 cancelled 结束原 prompt。UI 预标记、通知发出与最终回执要分开，远端既成副作用仍需对账。[G32][G35]

验收

将消息注入收尾窗口并重复投递：不丢唤醒，不因旧结果启动新修改；在途操作单独记录已停止、已完成或尚未确认。

来源 [G32] [G35] [A05] [C65] [B46]

工作空间：隔离要有对象清单

横向综合 / Context、worktree、process、sandbox

> “隔离”必须说明隔离了什么。会话上下文、代码副本、进程、权限和外部服务是不同的对象。

隔离层	解决的问题	仍共享的可能对象
上下文	推理历史污染	文件/权限/进程
工作树	代码副本与分支	共同Git配置/refs、缓存/服务
独立进程	故障与运行状态	用户身份/宿主资源
沙箱/VM	资源访问限制	显式挂载与外部服务

1. 按资源逐项定义

1.1 新上下文可共享进程与目录；linked worktree 仍共享多数 refs 和默认仓库 config，启用 worktreeConfig 只增加专属层。若任务能写共同元数据，源码文件从不冲突也不足以隔离 Git 控制面。[H03]

1.2 OS 沙箱提供另一层资源限制，但允许共享的目录与网络服务仍然构成共享面。需要把运行策略和业务写入范围一起描述。[G03][G11][G25]

2. 可执行的所有权策略

2.1 文件集设当前所有者或独立分支；同时保护共同 Git 元数据。强隔离可选择独立仓库和运行身份，但独立 clone 仍不自动隔离全局配置、凭据与外部服务。生成物、锁文件和迁移由集成任务治理。

2.2 为每个任务分配独立端口范围、临时目录和服务命名空间；测试必须显示使用的环境身份，否则一个任务可能把另一个任务启动的服务误当作自己的成果。

Peri/AOS 边界

建议分工：Peri 表达任务所需资源与能力，工作空间平台落实隔离并回传有效环境；是否适合现有代码仍需审计。

来源 [G03] [G11] [G25] [H03]

可信交付：验证应绑定合并后的版本

横向综合 / 审查与验收

> 并行产出增加的是候选结果。只有对集成后的同一版本形成测试和审查证据，才能判断任务是否真正完成。

结构示意 · 关系与数据流

节点与职责

1. 候选补丁
2. 受测输入清单
3. 自动测试
4. 独立审查 · 受保护检查配置
5. 验收记录

有向关系 · 后半句为条件或动作

- 候选补丁 → 受测输入清单 · 集成并登记
- 受测输入清单 → 自动测试 · 同一输入
- 受测输入清单 → 独立审查 · 同一输入
- 自动测试 → 验收记录 · 测试证据
- 独立审查 → 验收记录 · 风险结论
- 验收记录 → 候选补丁 · 失败返工

1. 推荐交付流程

1.1 子任务报告基线、补丁身份与自测；集成者记录目标版本及冲突处理；验证者对实际集成代码树验收，再报告失败与未覆盖范围。候选比较、应用和验收不是一个动作。[E43][G37]

1.2 评审者应先得到需求、补丁与证据，然后再看实现者解释。这样能降低被自述成功锚定的风险；对抗评审要提出可验证反例，而不是为了反对而反对。

2. 证据不能错配

2.1 记录 commit、实际受测 tree/diff 的覆盖范围，并补实际使用的 ignored/untracked 产物、子模块版本/脏状态与有效配置；常规 status 无输出不等于输入一致。敏感配置只留受控版本引用及脱敏参数，不公开密钥或其简单摘要。[H04]

2.2 工具成功只支持该工具明示的完成语义，不能直接推出用户目标达成；模型自述也不是独立验证。执行完成、测试通过、审查接受和用户验收应分别记录。[G07][G08]

验收记录

受测输入身份不证明检查器可信：若执行者可改 core.hooksPath，本地 Git Hook 不是独立门禁。[H05] 建议记录检查器版本/受保护配置，在独立运行域验收；图中“同一输入”均指已声明覆盖范围。

来源 [G07] [G08] [E43] [G37] [H04] [H05]

长会话：三份视图，各有自己的生命周期

横向综合 / 事件日志、模型上下文与 UI 历史

> 完整历史、模型本轮所需内容和前端正在展示的内容不必常驻为同一份大对象。分开管理才能独立优化。

结构示意 · 关系与数据流

节点与职责

1. 持久事件
2. 摘要检查点
3. 模型活动窗
4. 界面历史页
5. 大产物存储

有向关系 · 后半句为条件或动作

- 持久事件 → 摘要检查点 · 压缩投影
- 摘要检查点 → 模型活动窗 · 恢复
- 持久事件 → 界面历史页 · 游标分页
- 大产物存储 → 模型活动窗 · 按需片段
- 大产物存储 → 界面历史页 · 按需展示
- 持久事件 → 模型活动窗 · 近期尾部

1. 推荐的数据关系

1.1 原始事件日志保留可追溯证据；模型视图由摘要、近期消息和按需材料构成；界面视图根据游标加载历史页。三者通过稳定 ID 关联，而不是各自复制完整文本。

1.2 ACP 区分 load 的历史回放与可选 resume 的不回放恢复；可选 compaction 更新不要求客户端丢弃旧历史。私有分页游标仍需协商扩展与客户端支持。[G31][G35]

2. 资源管理要求

2.1 实现反例：Pi 恢复建全历史索引，Goose 主路径全读且保留不可见对象，OpenHands 还有独立 Event 缓存。三者缩短模型投影，不由此保证历史窗口驻留。[B71][B52][B72][B73][D50] 建议分别设置读取、缓存和引用归还预算。

2.2 当前 v2 导航仍标 Draft，不能因读到草案就迁移全部客户端。协议版本、扩展命名和存储格式版本分别管理，兼容性测试也分开执行。[G33]

Peri 启发

建议共用稳定事件 ID 与代际，各投影保留自己的游标及覆盖区间映射；模型压缩、内存回收和界面分页不必使用同一个游标。

来源 [G31] [G33] [G35] [B71] [B73] [D50] [B52] [B72]

压缩提交：先形成可恢复状态，再释放旧引用

横向综合 / Compaction 的一致性

> 压缩失败只是一次派生计算失败，不应破坏已有可恢复历史。内存优化要落到引用生命周期和提交顺序。

阶段	可见状态	故障后的选择
生成候选	旧视图仍有效	失败丢弃候选
提交版本	覆盖范围与源版本匹配	冲突重算/按已提交头恢复
重建尾部	摘要+连续事件	重检水位与重复事件
释放引用	所有读者退出旧视图	检查缓存持有者

1. 建议的提交步骤

1.1 固定源版本与覆盖区间，边界保持工具调用/结果配对，或保留明确的在途记录。生成并验证摘要后，先保存候选，再以版本比较提交检查点及活动头；旧候选不能覆盖较新的摘要。

1.2 新消息始终追加持久事件。切换时按摘要覆盖终点重建尾部，并在提交与订阅衔接处重检水位，避免快照期间的消息落空；最后才释放无读者持有的旧视图。

2. 失败与观测

2.1 建议恢复只用已提交头；但“有 checkpointer”不等于每步同步落盘：LangGraph exit 仅图执行退出时保存，async 与下步并行，sync 等检查点写入再继续。[H10] 必须声明持久化模式、失败窗口及旧候选处置。

2.2 Factory 公开锚定增量摘要与双阈值；本页并发提交协议为设计建议。测摘要、活动窗、缓存及存活对象字节；RSS 还受分配器保留影响，token 减少不等于进程内存立即下降。[G23]

实验

建议分开注入：日志 ENOSPC 停记、事件已写而 HEAD 未存、usage/摘要跨步中断，再检查恢复及RSS。对照 Gemini 停记继续、OpenHands 修复窗口与 Goose 分步持久化，不能只测试模型摘要成功。[A44][D52][B72]

来源 [G23] [A44] [D52] [B72] [H10]

性能：衡量整个任务，而非一轮响应

横向综合 / 成本、延迟与并发

> 优化提示词、委派或压缩都可能把开销移到别处。只有任务级账本才能识别速度提升是否来自更多成本或更少验证。

指标	需要固定	发现什么
验收成功率	任务与规则	质量是否退化
任务总成本	模型和预算	返工/重复读取
端到端延迟	并发与环境	真实关键路径
资源增长	会话长度/输出大小	泄漏与回收缺口

1. 分解任务总成本

1.1 把输入/输出 token、模型等待、工具执行、排队、持久化、恢复、审阅和返工分开计量。并行工具节约的是关键路径时间，未必减少总计算量。

1.2 关注达到同一验收标准的耗时与费用。一个只生成补丁却未运行测试的系统，不能与完成验证的系统直接比较首轮时间。

2. 对资源约束的测量

2.1 长程场景测内存随轮次增长、历史恢复峰值和取消后残留任务；高并发场景测限流公平性、队列等待与共享服务热点；低资源场景测背压和可降级能力。

2.2 优化前先做成对基线，再改变单一机制。成本同时报告总量和成功任务摊销；把失败任务排除会人为美化效率。厂商 benchmark 只能作为外部参考。[G06][G23]

禁止虚构收益

本报告未在统一硬件上运行所有产品，不提供未经实测的“提升倍数”或跨产品绝对性能名次。

来源 [G06] [G23]

评估矩阵：把漂亮设计变成可否证实验

横向综合 / 配对任务与故障注入

> 研究结论只是候选方案。验收集应同时覆盖普通成功路径和决定架构可靠性的边界条件。

实验轴	对照组	处理组
委派	单代理相同预算	可选委派
提醒	稳定提示词	按事件短提醒
压缩	现有恢复逻辑	摘要+游标检查点
协作	共享目录	任务工作树+集成门禁

1. 建议任务集合

1.1 小修复考察不必要委派；大仓库定位考察检索与证据；跨模块改动考察任务拆分和合并；长会话恢复考察压缩；服务启动任务考察异步工具和取消。

1.2 加入同 key 不同 payload/epoch、定义变更后沿用旧批准、ignored 测试输入差异、父结束与子结果迟到、审批取消、写入后未回执等对抗场景。每个场景明确反例前提，不用不存在的能力制造假故障。[H02][H04][H07]

2. 实验纪律

2.1 固定模型版本、任务基线、权限、网络和预算；保存轨迹、产物和退出原因。更换模型时重新比较，不把旧 harness 优势视作永久规律。

2.2 一次改变一个变量，并记录重复差异、漏报与误报。Anthropic 的历史 Research 评估曾以单 rubric 调用最贴近人工，说明多 judge 并非必胜；建议与工具测试和人工抽检对照，不外推为代码审查定律。[H06]

交付件

实验配置、任务基线、轨迹索引、实际产物和失败分类。没有这些，评估数字难以指导工程优先级。

来源 [H02] [H04] [H07] [H06]

Peri：先建立可证明的运行边界

研究建议 / 实施顺序

> 先审查可能造成越权、任务丢失或虚假完成的边界，再评估委派与角色策略。实施阶段与漏洞严重程度分别管理。

阶段	可交付成果	通过条件
正确性	状态/权限合同	边界场景可重放
长程稳定	事件+摘要+游标	本地不重复应用；外部未知可对账
协作质量	明确任务合同	同标准比较质量、成本与延迟
策略演进	评估与回滚机制	新模型/规则可对照

1. 按风险组织工作

1.1 审计越权写入、取消后新启动的未授权操作、不可恢复的消息丢失；按可复现影响、范围和补救能力定级，不预设均为 P0。在途远端副作用可能不可撤销，需标明已发生和未知状态。

1.2 建议阶段一建立任务/消息/产物身份与完成契约；阶段二验证恢复检查点、提示词装配及上下文分层；阶段三评估委派和模型路由。这是研究路线，未确认 Peri 存在上述缺陷。

2. 一个阶段一个可验收结果

2.1 每阶段验收分别检查服务启动期与工具调用期的可达权限、停止/唤醒/取消，以及恢复与引用释放。stdio 本地服务启动已经执行代码，tools/call 审批本身不约束该进程；纯远端 HTTP 不直接适用这一本机启动反例。[H02]

2.2 可供讨论的分工是：AOS 管跨工作空间调度、环境和交付，Peri 管代理执行语义与可观察状态。此为参考边界，需对照项目代码确认；本报告没有审计或改变现有实现。

决策建议

先选三个高价值机制做实现审计：可靠唤醒、分层上下文、明确委派合同。其余亮点进入实验池。

来源 [H02]

长程任务推演：从接单到次日续跑

研究推演 / 多轮用户输入与迟到结果

> 把长任务拆成可持久化阶段，既能保持用户目标连续，也能允许模型、进程和界面在中间重启。

结构示意 · 关系与数据流

节点与职责

1. 接单与范围
2. 独立调查
3. 并行实现
4. 检查点/恢复
5. 集成验收
6. 交付证据

有向关系 · 后半句为条件或动作

- 接单与范围 → 独立调查 · 证据优先
- 独立调查 → 并行实现 · 有界任务
- 并行实现 → 检查点/恢复 · 持久状态
- 检查点/恢复 → 集成验收 · 继续依赖
- 集成验收 → 交付证据 · 同一版本
- 集成验收 → 并行实现 · 失败返工

1. 示例轨迹

1.1 用户提交跨模块需求；主代理生成验收条目，启动两个只读调查任务。每个任务返回代码定位与风险，主代理据此分配不同工作树中的实现任务。

1.2 用户追加兼容性约束，系统记录新的目标版本并发给受影响任务。已完成的旧任务结果仍保留，但必须重新检查是否符合新约束，不能直接合并。

2. 跨夜边界

2.1 恢复粒度须显式：LangGraph interrupt 恢复重跑所在节点，节点内多中断按序号匹配，前置副作用应幂等或拆开。[H09] 本报告建议稳定动作身份、先存结果再唤醒；不能把图状态恢复等同逐语句继续。

2.2 集成后验证同一版本，并报告未覆盖项。用户取消时禁止启动新动作，传播停止并记录回执；不能撤销的在途副作用单列，晚到结果归档，不自动触发新一轮修改。

验收重点

参考官方 Goals：目标持久在 thread，续跑仅在空闲安全边界且受预算约束；中断会暂停，预算耗尽不是目标完成。[A57] 本报告建议将这些规则与新约束、检查点、迟到结果及集成验收共同测试。

来源 [A57] [H09]

smolagents：代码是动作语言

通用代理框架 / 提示装配与执行拓扑

> smolagents 将工具组合写成 Python 片段，把委派也暴露为可调用接口；关键取舍不是代码比 JSON 更强，而是解释器、凭据和子代理分别在哪里执行。

层次	实际机制	不应推导
动作与记忆	日志投影为模型消息，执行后记录 ActionStep [I04]	日志存在即崩溃检查点
主从委派	子代理对象被当作可调用接口 [I01]	自动建立分布式团队
停止接口	interrupt 置标记，主循环在步边界检查 [I01]	在途外部工具立即终止
沙盒拓扑	片段远端执行与整套代理入沙盒分开 [I03]	凭据、工具均已同等隔离

1. 把提示与工具接口一起生成

1.1 CodeAgent 将工具、托管代理、授权导入、定制指令和代码块边界送入 Jinja 模板；托管代理有 task/report 包装，调用后取结果返回父代理。[I01]

1.2 公开模板用代码动作、观察和最终答案组织过程；委派要求同时返回简要结论、详细结果和相关上下文。这里是开源模板机制，不是闭源提示词重建。[I02]

2. 组合能力需要明确执行位置

2.1 官方本地执行器按 AST 逐操作解释，限制导入并计数操作；但授权库仍可能带来资源耗尽等风险，不能当作操作系统隔离。[I03]

2.2 当前 create_python_executor 对远端执行器与 managed_agents 的组合直接报错。官方另给整套代理进入沙盒的方案，意味着模型调用和凭据边界也随之迁移，不能只切换一个参数便声称完成多代理隔离。[I01][I03]

Peri 启发 / 未实测

设计建议：将动作语言与执行位置分别建模，委派报告和最终校验分开。此为通用代理框架的静态快照，不是开箱编码产品；未运行逃逸、取消或性能测试。

来源 [I01] [I02] [I03] [I04]

CAMEL：协作落实为任务包

通用多代理框架 / Workforce 的通信与恢复边界

> CAMEL Workforce 将任务拆解、分配和执行分开，再以任务包状态协调工作者；它提供应用内协作原语，但内存快照与停止标记不能替代持久恢复和外部副作用收尾。

结构示意 · 关系与数据流

节点与职责

1. 任务待投递 · 发布者与收件人
2. SENT
3. PROCESSING · 受锁保护认领
4. RETURNED · 发布者取回结果
5. 依赖归档 · 独立发布依赖路径

有向关系 · 后半句为条件或动作

- 任务待投递 → SENT · post_task
- SENT → PROCESSING · 认领
- PROCESSING → RETURNED · return_task
- 任务待投递 → 依赖归档 · post_dependency

1. 模型提议，运行时验证与投递

1.1 官方区分 task_agent 拆解与 coordinator_agent 分配；单代理 Worker 和双角色 RolePlayingWorker 是不同组织方式，不是所有成员都在自由群聊。[I08]

1.2 分配提示要求任务 ID、工作者 ID 和依赖；执行提示带父任务、依赖结果，输出内容与失败标记。附加信息被标成资料而非指令，这仍是提示约束。[I07]

1.3 运行时另校验 assignee_id 是否属于有效工作者。TaskChannel 通过 Condition 保护认领，SENT 转 PROCESSING，再按发布者取回 RETURNED 包；结果归档后可供后续依赖使用。[I05][I06]

2. 停止与恢复各有作用范围

2.1 优雅停止设置停止请求；立即停止清理待办并通知子节点、取消监听，但不等待孩子完全清理。状态显示 STOPPED 不能作为外部副作用已结束的证明。[I05]

2.2 save_snapshot 把任务、待办、完成项、依赖和分配记录追加到对象内列表；恢复从该列表取回字段。该路径未持久化通道或外部环境，不能据此承诺进程崩溃续跑。[I05]

Peri 启发 / 未实测

图示当前进程内通道，不是持久消息队列。设计建议：为 AOS 另定义耐久任务记录、迟到结果和收尾确认；分配给合法工作者不证明依赖完整，模型返回未失败也不证明补丁通过验证，均需独立验收。不要把 role playing 当独立审查保证，也不把框架能力当编码产品交付能力。

来源 [I05] [I06] [I07] [I08]

Tabnine：治理层与入口迁移

IDE Agent / 新 OpenCode 分发 / 旧 CLI 分开核实

> 可借鉴的是把组织模型、工具和检索治理附着到既有执行器；同一品牌下的新旧入口不能共享未经核实的恢复或权限保证。

入口	已核实机制	不推定
新 CLI 分发	OpenCode + Tabnine 治理	同 B 章源码版本
IDE Agent	逐工具审批、远端检索	检索即写权限
旧 CLI 6.6后	维护期与恢复命令	新入口恢复兼容

1. 当前 CLI：复用执行器，叠加组织政策

1.1 新入口由 Tabnine 管理独立 OpenCode 副本并预装插件，启动器不回退到 PATH 中另一份 OpenCode；模型请求使用 Tabnine 凭据，供应商锁定为 tabnine，内置上下文检索与 Coaching MCP 自动注册。[I10]

1.2 官方说明组织工具/MCP政策、模型可用性和配额自动适用。应分别记录 Tabnine 包版本与内层 OpenCode 版本；本报告未核实其与 B 章固定提交同版，也不把治理声明当成完整引擎源码事实。[I10]

2. IDE：把检索能力与执行审批分开

2.1 IDE 设置将原生读文件、改文件、执行命令等分别设为自动、先问或禁用；上下文引擎暴露远端符号、文件及语义检索接口。检索结果进入上下文，不等于获得修改远端仓库的权限。[I11]

2.2 IDE MCP 文档要求打开项目目录后才读取配置，并按 command、url 或显式 SSE 识别传输。本页只将此记为 IDE 合同，不移植为新 OpenCode 分发的配置规则。[I12]

3. 生命周期决定建议适用范围

3.1 旧 Tabnine CLI 自6.6停止主动开发，仅关键更新至2026-12-31，之后弃用并引导迁移新插件。旧文档的会话恢复、文件恢复和摘要命令，是不同恢复粒度，不能据此保证新入口等价迁移。[I13]

Peri 启发 / 设计建议

AOS可把组织政策做为适配层，但应显式标记执行器版本、政策来源和入口能力矩阵。未审计完整实现，也未找到足以在本页承诺原生子代理通信或崩溃恢复的合同。

来源 [I10] [I11] [I12] [I13]

Qoder：选择提示与执行提示分离

CLI 当前公开合同；不混入 Cloud Agents 机制

> description 决定何时委派，正文决定受委派后如何工作；独立上下文、可选 worktree 和审批通道是三条不同边界。

层次	公开机制	验收问题
选择	description	是否委派给正确专长
执行	正文 + 工具集合	是否允许再次派发
工作副本	可选 worktree	是否实际返回独立路径
审批消费	TUI / Headless / 宿主	ask 是等待还是拒绝

1. 把委派配置变为可审查合同

1.1 Markdown 定义的 description 用于选择子代理，主体是其系统提示词；各子代理有自己的上下文、工具注册表、记录及压缩流程，结果回主会话。这是公开配置合同，不是闭源默认系统提示词的复原。[I14]

1.2 未设 isolation 使用默认工作区；需要独立检出才显式设 worktree。tools 可允许 Agent(指定名称)，或禁用 Agent 以禁止再次委派。父会话若已为 acceptEdits、bypassPermissions 或 auto，子代理不能靠 permissionMode 自行收紧；上下文独立不等于权限独立。[I14]

2. 工具调用与审批结果不能只看提示词

2.1 主循环将工具结果回填上下文，背景来自会话历史、项目指令、长期记忆；长上下文可压缩。非交互 Headless 将 ask 转为拒绝，SDK交给宿主回调，ACP交给 IDE RPC，故同一任务可因承载端不同而停在不同边界。[I15][I16]

2.2 Agent 启动规则按 deny、ask、allow 处理，没有匹配规则时无需确认。前置 PreToolUse 的 allow/deny 可短路常规权限流水线，因此只阅读静态 allow/deny 列表不足以还原有效决策，还需审查 Hook 及运行模式。[I16]

Peri 启发 / 设计建议

为不爱委派的问题分别评估选择描述、任务粒度和工具可达性；记录每次有效审批来源。上述为当前 CLI 动态文档说明，未运行二进制、未验证取消或持久恢复；不能外推到云端多代理。

来源 [I14] [I15] [I16]

CodeBuddy：团队数据不等于团队恢复

CodeBuddy Code / 官方文档镜像 2026-09-30

> 子代理返还结果，团队增加成员信箱与任务表；本地保存协作数据仍不等于恢复成员，工具配置也可能被记忆能力扩展。

对象	已说明	未由此保证
普通子代理	独立上下文、作用域 MCP	独立进程或沙盒
Teams	直连消息、共享任务	崩溃后成员恢复
工具白名单	memory 可自动扩展	声明值等于最终值

1. 两类协作拓扑，两个生命周期

1.1 普通子代理拥有独立上下文和自定义提示词；省略 tools 会继承主线程工具。作用域内联 MCP 只对子代理可见并随其结束关闭；借用全局 MCP 不随子代理关闭。可见性与进程权限仍须另审。[I17]

1.2 Teams 成员在进程内运行，加载项目上下文与领导初始任务，但不继承领导对话历史；成员可互发消息，任务表含依赖。关闭请求允许成员接受或拒绝，完成成员收到新消息可再启动。[I18]

2. 核对配置的实际展开值

2.1 memory 可在生成时注入限量 MEMORY.md，但启用后会向显式 tools 白名单自动加入 Read/Write/Edit。因此只看声明中的读工具，不能认定最终能力只读；该点是官方配置说明，未运行越权实验。[I17]

2.2 Settings 说明子代理权限可被覆盖，而父会话 auto/dontAsk 仍形成上限；autoMode 的自然语言分类规则只从用户、本地项目或CLI读入，不从共享项目配置读入，避免仓库静默改写本地信任边界。[I19]

3. 存盘、停止与恢复分别验收

3.1 团队配置、信箱、任务表有本地存放路径；但 Teams 仍属实验功能，/resume 和 /rewind 不恢复成员，关闭也可能等当前请求或工具结束。不能将现存 JSON 文件视为可重建执行状态的保证。[I18]

Peri 启发 / 设计建议

编译后输出有效工具清单和权限来源，再做子任务验收；AOS应把信箱持久化、成员重建、唤醒和资源释放列为不同状态。这里仅核实官方合同，未审计完整实现；权限采用Settings细则，不泛化Teams末尾的简单继承描述。

来源 [I17] [I18] [I19]

iFlow：历史配置的继承陷阱

iFlow CLI / 停维基线与子代理合同

> iFlow 的学习价值在可配置代理与权限继承的组合；但它已停止维护，公开说明与可审查的执行内核必须分别计入证据。

配置单元	公开用途	不能推定
whenToUse	帮助选择代理	一定主动委派
systemPrompt	规定角色行为	运行时阻断副作用
继承开关 + 列表	组合工具和 MCP 范围	默认最小权限
Task 与结果展示	承接专项工作	并行及持久恢复保证

1. 先限定产品与时间

1.1 团队公告确认 2026-03-20 停维、4 月 17 日关闭 CLI 产品及配套 API/模型库；FAQ 又说明已安装 CLI 可接自定义 API 继续使用。因此不是所有本地执行都被禁用，也不是心流 AI 助手整体停运。[I20]

1.2 本次固定 4642808a 的公开仓库以文档、安装脚本和资源为主，未取得 CLI 内核源码。SDK 文档、可安装包与远程模型服务是不同边界，本页不把它们合称完整开源运行时。[I22]

2. 选择说明与权限不是同一个字段

2.1 官方子代理文档将 agentType、systemPrompt、whenToUse 放入项目 .iflow/agents 配置；Task 工具负责调用，用户也可用 $ 类型名指定。触发条件、角色提示和调用入口因而可以独立检查。[I21]

2.2 文档默认 isInheritTools/isInheritMcps 为 true：继承父级再附加列表；置 false 写为采用显式列表，但 allowedMcps 空数组又被解释为不限制/继承，组合仍有歧义。仅写 allowedTools 不代表收窄；这是公开合同，非执行器或恶意配置测试。[I21]

3. 分析：可迁移的是审查方法

3.1 文档示例按审查、优化、写文档顺序交接，不能据此认定已实现并行调度、可靠取消或事务合并。[I21] 建议 Peri/AOS 将继承策略与最终有效能力一并展示，配置审查不能只看角色名称。

3.2 建议用“父级宽权限 + 子级窄列表”作回归用例，明确测试继承开关、MCP 服务范围与第三方代理信任；这些是本报告建议，没有运行 iFlow 或取得其权限执行路径。

历史样本

按停维时公开合同学习；不采用旧 README 的竞品勾选表、免费服务承诺或未验证效率描述。2026-10-19 复核。[I20][I22]

来源 [I20] [I21] [I22]

Zoo：延续身份不替代机制核查

Zoo Code / 社区 fork 与挂起委派动作

> Zoo 是 Roo 的社区延续而非原项目复活；当前源码提供了审批动作身份的具体样本，也提醒我们不能将团队宣传直接换成并行保证。

已观察层	具体对象	仍需补证
项目声明	社区延续、许可、协调能力	发行版可用性及全部并行路径
原生工具协议	严格参数与单独调用提示	运行时强制互斥
待批准动作	actionId + 参数 + 审批文本	落盘与重放事务性
委派接口调用	parentTaskId / pendingActionId	故障后创建去重与回流

1. 保留分叉后的独立身份

1.1 当前 Zoo-Code-Org/Zoo-Code README 自述由曾为 Roo 贡献代码的开发者延续，许可为 Apache 2.0；原 Roo 的停运结论不应套成 Zoo 同样停止维护。两者仍是不同项目和代码快照。[I23]

1.2 本页固定 b69989b8。README 声称加强并行任务协调，但本次核查的是 new_task 的具体路径，不将某个函数的串行特征扩大为全产品无并行，也不反向用宣传保证任意调用可并发。[I23][I24]

2. 审批前登记可识别的动作

2.1 NewTaskTool 校验参数、待办和目标模式后，若有 toolCallId，清洗为 actionId，将 create_subtask 的参数及审批文本写入 provider 和当前 Task 的 pending action，然后请求批准。批准后传 pendingActionId 调用父子委派接口。[I24]

2.2 这让“等待批准的哪一个创建动作”具有显式身份。它仍不是持久化写入原子性、重复审批去重或崩溃后恰好一次创建的完整证明；本次未追完 provider 存储与恢复实现，更未做故障注入。

3. 工具协议与调度约束分别验收

3.1 原生 schema 要求 mode、message、todos 字段，todos 可为空值；工具说明要求单独调用 new_task。执行文件注释称新子任务为唯一活跃任务，但提示里的“单独调用”不能自行证明执行时互斥。[I24][I25]

3.2 对 Peri/AOS 的设计建议：把审批记录绑定动作身份与标准化参数，再独立检查父任务状态、子任务所有者和结果回流。应测试批准后响应丢失与重复续跑，而不是只观察界面成功打开子任务。

证据分层

身份与能力宣传属官方说明；动作登记和调用顺序属固定源码观察；可靠性结论仍需完整路径及实验。本页不重复 Roo 的旧模式页。

来源 [I23] [I24] [I25]

MAF：会话与图恢复分层

Microsoft Agent Framework / Python 与 C#

> 汇集 AutoGen 与 Semantic Kernel 经验的 MAF 值得补入，但跨语言统一概念不等于同名接口，更不等于统一的自动持久恢复。

层次	Python 文档接口	C# 文档接口
会话容器	AgentSession / state	AgentSession / StateBag
会话还原	AgentSession.from_dict	agent.DeserializeSessionAsync
图状态保存	on_checkpoint_save	OnCheckpointingAsync
图状态恢复	on_checkpoint_restore	OnCheckpointRestoredAsync

1. 核心稳定与扩展成熟度分开

1.1 2026-04-03 官博宣布 Python 与 .NET 1.0，核心代理、工作流稳定；同篇另列预览能力。不能沿用早期资料称整体仍预览，也不能把核心 GA 扩大成所有扩展稳定。[I27]

1.2 当前 ebcf4db7 README 仍将 AF Labs 标为实验包，Go 指向另一仓库；官方概览将 Go 列为公测。各语言版本号/API 不因此一致，通用框架也不是可直接交付的编码产品。[I26][I30]

2. AgentSession 保存对话，不代表保存整张图

2.1 C# AgentSession 是含 StateBag 的抽象基类；Python 则有 session_id、service_session_id 与 state，采用 to_dict/from_dict 恢复。会话带有代理/服务适配条件，不能任意换提供方后沿用。[I28]

2.2 服务端会话 ID 不是最终用户授权边界：多租户共享后端凭据时，应用仍需映射并核验归属。序列化对话状态和授权检查是两种责任，不因可以续聊便自动解决。[I28]

3. 图检查点有另外的状态与恢复入口

3.1 通常在超步完成后记录执行器、待投递消息等图状态；Python 1.13.0 增加入口检查点。两种语言须接入各自保存/恢复钩子。重建 .NET 图须保持拓扑与执行器身份；本地 ChatClientAgent 的 Id 及已设置的 Name 应稳定。[I29]

3.2 建议 Peri/AOS 分开验收会话恢复、图调度恢复和外部副作用对账；内存检查点不跨进程持久。稳定逻辑角色身份应与租户/请求身份分开，数据库、文件或进程也不会仅靠会话 JSON 自动复原。

验证边界

表中是 2026-10-19 读取的官方语言分支，不是兼容性测试；图恢复须配置存储并接入执行器状态，不能与 AgentSession 序列化合并承诺。[I28][I29]

来源 [I26] [I27] [I28] [I29] [I30]

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[A01] Claude Code: How Claude Code works

https://code.claude.com/docs/en/how-claude-code-works

official · 2026-10-18 · 核实 agentic harness、上下文来源、JSONL 会话、压缩与工具定义延迟加载。

[A02] Claude Agent SDK: Modifying system prompts

https://code.claude.com/docs/en/agent-sdk/modifying-system-prompts

official · 2026-10-18 · 核实 minimal default、claude_code preset、append/custom string、项目指令与 system prompt 的独立加载路径。

[A03] Claude Code: Memory

https://code.claude.com/docs/en/memory

official · 2026-10-18 · 核实 CLAUDE.md、路径规则、auto memory、说明性上下文与强制权限的区别。

[A04] Claude Code: Skills

https://code.claude.com/docs/en/skills

official · 2026-10-19 · 核实元数据发现、按需加载、disable-model-invocation、context: fork、动态上下文和调用生命周期。 本轮补核实正文跨轮/压缩预算、allowed-tools 临时免批而非 allowlist、disallowed-tools 移除工具池、workspace trust 与托管锁例外。

[A05] Claude Code: Subagents

https://code.claude.com/docs/en/sub-agents

official · 2026-10-19 · 2026-10-19 复读当前官方文档：普通代理完成/模型 TaskStop 与用户 x/SDK stop_task 的再唤醒差异；同 ID 新 run、兄弟恢复结果路由及当前对话内同名身份检查。非闭源实现审计。

[A06] Claude Code: Agent teams

https://code.claude.com/docs/en/agent-teams

official · 2026-10-19 · 2026-10-19 复读当前文档：实验默认关闭、JSON 邮箱/任务锁、自动计划批准；交互会话 teammate 与 -p/SDK 普通代理区别，协作关闭及 in-process 恢复限制。

[A07] Claude Code: Hooks guide

https://code.claude.com/docs/en/hooks-guide

official · 2026-10-18 · 核实确定性生命周期 hook、Stop 不等于任务完成、并发 updatedInput 非确定顺序、HTTP hook 失败策略。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[A08] Claude Code: Hooks reference

https://code.claude.com/docs/en/hooks

official · 2026-10-19 · 核实 PreCompact/PostCompact 的输入与阻止压缩时不同情境的结果。 本轮核实普通 PreToolUse 整对象 updatedInput、重验新参数、deny/defer/ask/allow 合并及 -p defer/resume 合同；非引擎源码实测。

[A09] Claude Code: Permissions

https://code.claude.com/docs/en/permissions

official · 2026-10-18 · 核实工具移除与按参数拦截、权限由 harness 执行、运行中更新规则的生效边界。

[A10] Claude Code: Sandboxing

https://code.claude.com/docs/en/sandboxing

official · 2026-10-18 · 核实 Bash/子进程隔离、文件与网络两层、代理及子代理共享配置；不是全进程容器。

[A11] Claude Code: Checkpointing

https://code.claude.com/docs/en/checkpointing

official · 2026-10-18 · 核实 Bash 修改不回滚、subagent 编辑恢复限制、文件回滚与会话回退不同。

[A12] Claude Code: Sessions

https://code.claude.com/docs/en/sessions

official · 2026-10-19 · 2026-10-19 复读当前 sessions：默认记录可禁用；从盘恢复不默认重跑中断工具，依代理定义重新加载，缺失则默认工具并告警；启动flags与permission mode依入口恢复，不等于完整环境/权限快照。

[A13] Claude Agent SDK overview

https://code.claude.com/docs/en/agent-sdk/overview

official · 2026-10-18 · 核实 Python/TypeScript SDK 运行 Claude Code binary，提供工具、权限、会话和 hooks。

[A14] Codex App Server

https://learn.chatgpt.com/docs/app-server

official · 2026-10-19 · 当前官方文档复读：thread/read 不装载；resume excludeTurns 与分页；turn 反向游标重含锚点、item anchor 排除自身及底层能力要求；最后 unsubscribe 后无订阅无活动30分钟才卸载。公开接口说明，不强推固定CLI快照或实测RSS。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[A15] Codex Subagents

https://learn.chatgpt.com/docs/agent-configuration/subagents

official · 2026-10-18 · 核实当前本地版本依直接请求/项目或 skill 指令派发；子代理继承父会话权限，角色文字不能制造只读 sandbox。

[A16] Codex core architecture and OS enforcement

https://github.com/openai/codex/blob/8c076942cb70fba8e5c4732c23f1faabc4aed484/codex-rs/core/README.md

official · 2026-10-18 · 固定快照 8c076942cb70fba8e5c4732c23f1faabc4aed484；读取业务核心边界及 Seatbelt/bubblewrap/Windows 策略适配和 fail-closed 说明。

[A17] Codex ThreadStore boundary

https://github.com/openai/codex/blob/8c076942cb70fba8e5c4732c23f1faabc4aed484/codex-rs/thread-store/README.md

official · 2026-10-18 · 固定快照 8c076942cb70fba8e5c4732c23f1faabc4aed484；读取 append_items、update_thread_metadata、LiveThread、JSONL+SQLite 分工。

[A18] Codex AGENTS.md discovery

https://github.com/openai/codex/blob/8c076942cb70fba8e5c4732c23f1faabc4aed484/codex-rs/core/src/agents_md.rs

code · 2026-10-18 · 固定快照 8c076942cb70fba8e5c4732c23f1faabc4aed484；读取 load_project_instructions、项目根标记、预算、权限约束 filesystem 与来源信息。

[A19] Codex instruction state transitions

https://github.com/openai/codex/blob/8c076942cb70fba8e5c4732c23f1faabc4aed484/codex-rs/core/src/context/world_state/agents_md.rs

code · 2026-10-18 · 固定快照 8c076942cb70fba8e5c4732c23f1faabc4aed484；读取 AgentsMdState::render_diff；不变不注入、替换/删除显式提示。

[A20] Codex skill catalog prompt

https://github.com/openai/codex/blob/8c076942cb70fba8e5c4732c23f1faabc4aed484/codex-rs/ext/skills/src/catalog_prompt.rs

code · 2026-10-18 · 固定快照 8c076942cb70fba8e5c4732c23f1faabc4aed484；读取 skill source locator/alias、目录呈现及按选择读取指令；不将全部参考文件塞入上下文。

[A21] Codex public model-specific prompt example

https://github.com/openai/codex/blob/8c076942cb70fba8e5c4732c23f1faabc4aed484/codex-rs/core/gpt-5.2-codex_prompt.md

code · 2026-10-18 · 固定快照 8c076942cb70fba8e5c4732c23f1faabc4aed484；公开模型专用提示词实例：搜索/编辑/脏工作区/审查/输出契约；不声称当前所有模型都采用此模板。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[A22] Codex multi-agent v2 spawn handler

https://github.com/openai/codex/blob/8c076942cb70fba8e5c4732c23f1faabc4aed484/codex-rs/core/src/tools/handlers/multi_agents_v2/spawn.rs

code · 2026-10-18 · 固定快照 8c076942cb70fba8e5c4732c23f1faabc4aed484；读取参数校验、fork_turns all/none、ThreadSpawnRequest、活动事件及 trace。

[A23] Codex agent spawn and resume control

https://github.com/openai/codex/blob/8c076942cb70fba8e5c4732c23f1faabc4aed484/codex-rs/core/src/agent/control/spawn.rs

code · 2026-10-18 · 固定快照 8c076942cb70fba8e5c4732c23f1faabc4aed484；读取 fork 历史刷盘、父角色转换、live process handles 不继承、权限快照和恢复路径。

[A24] Codex unloaded-agent mailboxes

https://github.com/openai/codex/blob/8c076942cb70fba8e5c4732c23f1faabc4aed484/codex-rs/core/src/agent/control/mailbox.rs

code · 2026-10-18 · 固定快照 8c076942cb70fba8e5c4732c23f1faabc4aed484；读取队列转交、watch 通知、清理空邮箱；不是自行推断持久化投递保证。

[A25] Codex: wake idle root on agent messages (feature-gated)

https://github.com/openai/codex/commit/8c076942cb70fba8e5c4732c23f1faabc4aed484

code · 2026-10-18 · 已读 commit 元数据与 patch；wake_on_agent_messages 默认关闭且 UnderDevelopment；拒绝中断态/Plan/child 自动唤醒，测试覆盖 direct 与 host mailbox。

[A26] Codex asynchronous compaction

https://github.com/openai/codex/blob/8c076942cb70fba8e5c4732c23f1faabc4aed484/codex-rs/core/src/session/async_compaction.rs

code · 2026-10-18 · 固定快照 8c076942cb70fba8e5c4732c23f1faabc4aed484；读取 history/model/auth generation 绑定、oneshot、任务跟踪与关闭计费；依 Feature::AsyncCompaction 和 remote V2。

[A27] Codex compaction installation

https://github.com/openai/codex/blob/8c076942cb70fba8e5c4732c23f1faabc4aed484/codex-rs/core/src/compact_remote_v2_installation.rs

code · 2026-10-18 · 固定快照 8c076942cb70fba8e5c4732c23f1faabc4aed484；读取 prepared prefix validation、live suffix、world-state baseline、替换历史及 token 重算。

[A28] Codex rollout reconstruction

https://github.com/openai/codex/blob/8c076942cb70fba8e5c4732c23f1faabc4aed484/codex-rs/core/src/session/rollout_reconstruction.rs

code · 2026-10-18 · 固定快照 8c076942cb70fba8e5c4732c23f1faabc4aed484；读取恢复 compaction、baseline、turn items 路径；用于区别持久记录与模型活动历史。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[A29] Codex approval and sandbox tool interfaces

https://github.com/openai/codex/blob/8c076942cb70fba8e5c4732c23f1faabc4aed484/codex-rs/core/src/tools/sandboxing.rs

code · 2026-10-18 · 固定快照 8c076942cb70fba8e5c4732c23f1faabc4aed484；读取 Sandboxable、审批缓存/请求、执行尝试与重试契约。

[A30] Codex execpolicy

https://github.com/openai/codex/blob/8c076942cb70fba8e5c4732c23f1faabc4aed484/codex-rs/execpolicy/README.md

official · 2026-10-18 · 固定快照 8c076942cb70fba8e5c4732c23f1faabc4aed484；读取 prefix_rule、host_executable、allow/prompt/forbidden 以及规则的可执行测试示例。

[A31] Codex execution server

https://github.com/openai/codex/blob/8c076942cb70fba8e5c4732c23f1faabc4aed484/codex-rs/exec-server/README.md

official · 2026-10-18 · 固定快照 8c076942cb70fba8e5c4732c23f1faabc4aed484；读取 JSON-RPC 执行服务、session/process 生命周期、远端转发、元数据只用于关联而非授权。

[A32] Gemini CLI core

https://geminicli.com/docs/core/

official · 2026-10-18 · 核实 packages/cli 与 packages/core 的 UI/模型/工具/状态分工。

[A33] Gemini PromptProvider

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/prompts/promptProvider.ts

code · 2026-10-18 · 固定快照 ed96faedfd5571f055c44f2c9509d62321d1138d；读取模型现代/legacy 选择、工具能力条件、plan 模式、skills/agents 目录、memory 与 prompt 覆盖/导出。

[A34] Gemini prompt snippets

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/prompts/snippets.ts

code · 2026-10-18 · 固定快照 ed96faedfd5571f055c44f2c9509d62321d1138d；读取职责分片、独立任务派发条件、研究/策略/执行/验证工作流、state_snapshot 压缩输出要求。

[A35] Gemini unified AgentTool

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/agents/agent-tool.ts

code · 2026-10-19 · 固定 ed96faed；2026-10-19 重读 AgentTool，新旧远程 invocation 由 isAgentSessionSubagentEnabled 分支选择；未运行。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[A36] Gemini LocalAgentExecutor

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/agents/local-executor.ts

code · 2026-10-18 · 固定快照 ed96faedfd5571f055c44f2c9509d62321d1138d；读取局部 registry/message bus、tool clone、Kind.Agent 过滤阻止递归、complete_task、时间/轮次/压缩。

[A37] Gemini local subagent protocol

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/agents/local-subagent-protocol.ts

code · 2026-10-18 · 固定快照 ed96faedfd5571f055c44f2c9509d62321d1138d；读取 send 后延迟启动、streamId、abort、生命周期事件与终止原因映射。

[A38] Gemini remote subagent protocol

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/agents/remote-subagent-protocol.ts

code · 2026-10-19 · 固定 ed96faed；2026-10-19 重读 blob 379d2948b9ce114da7bfc21b96944d843ff3c11f：abort 仅本地 signal；正常 EOF 直接 COMPLETED，未按远端 Task 终态细分；静态条件路径，非事故复现。

[A39] Gemini Remote Subagents documentation

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/docs/core/remote-agents.md

official · 2026-10-18 · 固定快照 ed96faedfd5571f055c44f2c9509d62321d1138d；读取 agent card、local/user definitions、认证与代理配置；A2A 与 MCP 不是同一语义。

[A40] Gemini tool scheduler

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/scheduler/scheduler.ts

code · 2026-10-19 · 固定快照 ed96faedfd5571f055c44f2c9509d62321d1138d；读取 validating/policy/confirmation/scheduled/executing/terminal、连续可并行批次、取消、hook 改参后重建 invocation。 本轮追读：policyDecision 后 hook ask 可覆写为 ASK_USER 并传 forcedDecision；用户确认后置为 Scheduled，静态条件性路径非生产复现。

[A41] Gemini PolicyEngine implementation

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/policy/policy-engine.ts

code · 2026-10-18 · 固定快照 ed96faedfd5571f055c44f2c9509d62321d1138d；读取排序规则、shell parsing/matching 和安全检查接口；不推断所有部署配置已开启。

[A42] Gemini PolicyEngine documentation

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/docs/reference/policy-engine.md

official · 2026-10-18 · 固定快照 ed96faedfd5571f055c44f2c9509d62321d1138d；读取 allow/deny/ask、tier priority、non-interactive deny；该快照仍告警 workspace policy tier 禁用。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[A43] Gemini compression service

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/context/chatCompressionService.ts

code · 2026-10-19 · 2026-10-19 重取固定 blob d5ff083aeda3ebfc59a780c6c6d30ccc442b95da：折叠/外置、state_snapshot与模型probe；空摘要/token反增拒绝安装，only-truncate依失败标志且非force；标志由client特定条件设置。

[A44] Gemini chat recording

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/services/chatRecordingService.ts

code · 2026-10-19 · 2026-10-19 重取固定 blob fe7a86762fa799599c050ca06ae4f53ccc77b87f：JSONL append+$patch/$rewindTo形成逻辑恢复视图，缓存驱逐后getConversation同步全量重放；ENOSPC告警停记而会话继续。临时文件rename仅特定回退分支，非普遍压缩事务。

[A45] Gemini Skills documentation

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/docs/cli/skills.md

official · 2026-10-18 · 固定快照 ed96faedfd5571f055c44f2c9509d62321d1138d；读取 discovery→activate_skill→consent→注入正文/目录访问，及技能来源优先级。

[A46] Gemini ACP mode documentation

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/docs/cli/acp-mode.md

official · 2026-10-18 · 固定快照 ed96faedfd5571f055c44f2c9509d62321d1138d；读取 stdio JSON-RPC、session/prompt/cancel 和 client filesystem proxy；未运行互操作测试。

[A47] Gemini AgentSession event wrapper

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/agent/agent-session.ts

code · 2026-10-18 · 固定快照 ed96faedfd5571f055c44f2c9509d62321d1138d；读取 AsyncIterable、先订阅再 replay、eventId/streamId 与 agent_start/end 范围。

[A48] Gemini Subagents documentation

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/docs/core/subagents.md

official · 2026-10-18 · 固定快照 ed96faedfd5571f055c44f2c9509d62321d1138d；读取独立 context/tool registry、前置元数据、subagent policy 匹配；文档的每 agent 同名工具描述与 unified AgentTool 实现存在层次差异，正文按源码路径叙述。

[A49] Gemini BeforeTool 结果解释

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/scheduler/hook-utils.ts

code · 2026-10-19 · ask 非阻止输出可到 scheduler；无 HookSystem 则 continue 固定快照重新读取，未运行产品；完整 blob 与路径链见本轮研究笔记。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[A50] Gemini 强制确认传递

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/scheduler/confirmation.ts

code · 2026-10-19 · 向 invocation.shouldConfirmExecute 传入 forcedDecision，确认后返回 outcome 固定快照重新读取，未运行产品；完整 blob 与路径链见本轮研究笔记。

[A51] Gemini 基础工具确认逻辑

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/tools/tools.ts

code · 2026-10-19 · forcedDecision 优先于再次消息总线策略查询；forced ask 进入确认详情 固定快照重新读取，未运行产品；完整 blob 与路径链见本轮研究笔记。

[A52] Gemini 项目 Hook 信任入口

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/hooks/hookRegistry.ts

code · 2026-10-19 · 项目合并 Hook 受 trusted-folder 与禁用名单约束；不能假定不受信任仓库默认注入 ask 固定快照重新读取，未运行产品；完整 blob 与路径链见本轮研究笔记。

[A53] Gemini A2A stream 与独立取消接口

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/agents/a2a-client-manager.ts

code · 2026-10-19 · 固定开发快照重新读取，未运行产品；stream 传 signal；另有 cancelTask，所读 remote 调用链未显式使用；blob a40e39f2f4ec2b756deed0b778876d8625a5146e。

[A54] Gemini A2A 终态与结果重组

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/agents/a2aUtils.ts

code · 2026-10-19 · 固定开发快照重新读取，未运行产品；extractIdsFromResponse 对 completed/failed/canceled/rejected 清 taskId；业务失败不统一抛异常；blob 876f6239114fe55806bcdef956592cdb6c122170。

[A55] Gemini remote session invocation

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/agents/remote-session-invocation.ts

code · 2026-10-19 · 固定开发快照重新读取，未运行产品；父 abort 接 session.abort 与本地 CANCELLED 显示，未单独确认远端取消；blob 6ff32ba41a701634486f7f5a94b92e4d90b11221。

[A56] Gemini legacy remote invocation

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/agents/remote-invocation.ts

code · 2026-10-19 · 固定开发快照重新读取，未运行产品；旧入口正常最终响应后标 COMPLETED；与新 protocol 对照，不把单一 feature 分支当全部实现；blob 1510849683b5696edd4bf1ec1672b38b5638dd46。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[A57] Codex Goals：持久目标与有界续跑

https://developers.openai.com/cookbook/examples/codex/using_goals_in_codex

official · 2026-10-19 · 官方 2026-05-09 工程示例；thread-scoped 持久目标、生命周期/预算、空闲安全边界续跑、中断暂停与无工具续跑抑制。官方说明，非所有构建默认可用或运行验证。

[A58] Responses Compaction：两类输出合同

https://developers.openai.com/api/docs/guides/compaction

official · 2026-10-19 · 当前官方 API：服务端压缩支持按最新compaction item裁前缀（previous_response_id链不手裁）；独立compact返回含保留项的规范窗口，不能只取加密项或再裁剪。非固定Codex CLI分支对应证明；未调用API。

[A59] Claude Code：压缩后重新注入的材料

https://code.claude.com/docs/en/context-window

official · 2026-10-19 · 当前官方说明：根规则、auto memory、plan 重新注入；最多回读五个文件，超5000token只返路径；路径规则按需加载。公开合同非闭源持久化/事务审计。

[A60] Gemini 压缩客户端：失败标志与安装

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/core/client.ts

code · 2026-10-19 · 固定 ed96faed；blob 01bcc1e19388c14ae3b310b50daa58d4791b28dc：仅自动token反增置失败标志，force及new chat不同；成功安装前getConversation可全量重放。静态未运行。

[A61] Gemini chat：新历史与记录协调

https://github.com/google-gemini/gemini-cli/blob/ed96faedfd5571f055c44f2c9509d62321d1138d/packages/core/src/core/geminiChat.ts

code · 2026-10-19 · 固定 ed96faed；blob 095a5292c4820c8158bbb73c5af97d729e69bd87：显式newHistory优先、补消息ID、初始化recorder后updateMessagesFromHistory；不能据此宣称跨多条JSONL原子提交。

[A62] Codex TUI：嵌入、守护与远端入口

https://github.com/openai/codex/blob/8c076942cb70fba8e5c4732c23f1faabc4aed484/codex-rs/tui/src/lib.rs

code · 2026-10-19 · 固定 8c076942；blob 8dd93b2cda5cf11f21255c5ea2fa7000c01dc509：start_app_server 与 app_server_target_for_launch 支持 embedded、daemon、remote；仅核查入口选择，不推定全部工具必经独立执行服务器。

[B01] OpenCode 官方仓库及当前开发快照

https://github.com/anomalyco/opencode

official · 2026-10-18 · 浏览官方仓库，并克隆 dev HEAD 802f2d8e75098ae0b45ad839ce98a31589ea8102；提交时间 2026-10-18。当前源码为 V2 模块布局，不能视为全部稳定安装版本。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[B02] OpenCode V2 指令装配与热更新

https://opencode.ai/v2/docs/instructions/

official · 2026-10-18 · 已读取 Scope、Discovery、Ordering、Updates：AGENTS.md 发现、嵌套去重及运行中更新。

[B03] OpenCode 工具条件化系统提示词

https://github.com/anomalyco/opencode/blob/802f2d8e75098ae0b45ad839ce98a31589ea8102/packages/core/src/session/system-prompt.ts

code · 2026-10-18 · 本地读取固定提交源码：render 根据 shell/write/edit 可用性插入相应工具指导。

[B04] OpenCode 内置 Agent profile

https://github.com/anomalyco/opencode/blob/802f2d8e75098ae0b45ad839ce98a31589ea8102/packages/core/src/plugin/agent.ts

code · 2026-10-18 · 本地读取固定提交源码：general/explore/title/summary/compaction 的模式、提示词、权限；explore 允许 shell:*，只读命令要求存在于提示词。

[B05] OpenCode subagent 工具实现

https://github.com/anomalyco/opencode/blob/802f2d8e75098ae0b45ad839ce98a31589ea8102/packages/core/src/tool/plugin/subagent.ts

code · 2026-10-18 · 本地完整读取固定提交源码：新上下文、父子校验、默认深度 1、前后台 job、取消、继续 session、动态 description。

[B06] OpenCode V2 权限

https://opencode.ai/v2/docs/permissions/

official · 2026-10-18 · 已读取：有序最后匹配、默认规则、自定义子 agent 非父 agent 权限子集、宿主 shell 边界。

[B07] OpenCode V2 压缩

https://opencode.ai/v2/docs/compaction/

official · 2026-10-18 · 已读取：默认保留约 15000 token、请求合并、幂等 ID、native checkpoint 可移植性限制、历史保留。

[B08] OpenCode 压缩源码

https://github.com/anomalyco/opencode/blob/802f2d8e75098ae0b45ad839ce98a31589ea8102/packages/core/src/session/compaction.ts

code · 2026-10-19 · 2026-10-19 重取固定 blob 3cb52790295e7057d06573598a2928f3ea27b4cf：成功发布Ended、失败Failed；摘要/原生provenance既有描述保持。结合Bus/projector查完成提交边界，未执行测试。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[B09] OpenCode 历史窗口查询

https://github.com/anomalyco/opencode/blob/802f2d8e75098ae0b45ad839ce98a31589ea8102/packages/core/src/session/history.ts

code · 2026-10-19 · 2026-10-19 重取固定 blob aeb7cfc9f81b2d9b27d3e43ff656ed7eb4352a16：最近成功compaction seq下界在查询中使用，native provider/model/endpoint兼容性限制保持。

[B10] OpenCode 持久输入队列

https://github.com/anomalyco/opencode/blob/802f2d8e75098ae0b45ad839ce98a31589ea8102/packages/core/src/session/inbox.ts

code · 2026-10-18 · 本地读取固定提交源码：SQL inbox、按 session 加锁、steer/queue 边界、同 ID admission 对账与生命周期冲突。

[B11] OpenCode V2 插件接口

https://opencode.ai/v2/docs/build/plugins/

official · 2026-10-18 · 已读取 setup/cleanup、同步 transform 重放、不可变已读快照及 resource 注册与连接状态的区分。

[B12] OpenCode V2 组织策略

https://opencode.ai/v2/docs/policies/

official · 2026-10-18 · 已读取 policy 与 permission 区分、tool.use 只收紧、上层配置优先、非法规则仅日志告警。

[B13] OpenCode V2 工具与 Code Mode

https://opencode.ai/v2/docs/tools/

official · 2026-10-19 · 当前官方 V2 tools 页：嵌套调用各守自身合同；session rename/move、worktree create/list 无独立 permission action，browser 目录级 deny 而非逐操作提示。Code Mode 无直接 FS、有 fetch。新版例外未声称与旧源码同版。

[B14] OpenCode V2 网络客户端

https://opencode.ai/v2/docs/build/client

official · 2026-10-18 · 已读取客户端与 HTTP API 同契约、共享惰性事件连接、等待订阅者接受事件、仅实时且无自动重连/重放。

[B15] Crush 官方仓库

https://github.com/charmbracelet/crush

official · 2026-10-18 · 已浏览 README；本地 main 快照 14ac9bc49eb9702c7618b9e199fae05f23b788f9，提交时间 2026-10-17（美国东部）。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[B16] Crush coder 提示词

https://github.com/charmbracelet/crush/blob/14ac9bc49eb9702c7618b9e199fae05f23b788f9/internal/agent/templates/coder.md.tpl

code · 2026-10-18 · 本地读取核心规则、编辑工作流、LSP 工具优先及输出规则；核实限量读与整文件读、不要提问与可提问例外之间的张力。

[B17] Crush 提示词构建器

https://github.com/charmbracelet/crush/blob/14ac9bc49eb9702c7618b9e199fae05f23b788f9/internal/agent/prompt/prompt.go

code · 2026-10-18 · 本地读取 Go template 构建、模型/环境/Git/上下文路径、skills 元数据发现与同名覆盖去重。

[B18] Crush 搜索子 agent 工具

https://github.com/charmbracelet/crush/blob/14ac9bc49eb9702c7618b9e199fae05f23b788f9/internal/agent/agent_tool.go

code · 2026-10-18 · 本地完整读取：fantasy.NewParallelAgentTool、task profile、子 session 的父 message/toolcall 关联。

[B19] Crush 协调器

https://github.com/charmbracelet/crush/blob/14ac9bc49eb9702c7618b9e199fae05f23b788f9/internal/agent/coordinator.go

code · 2026-10-18 · 本地读取 buildAgent/buildTools/runSubAgent：初始化 context.WithoutCancel、配置允许工具过滤、子会话、费用尽力汇总、空结果报错。

[B20] Crush SQLite 连接治理

https://github.com/charmbracelet/crush/blob/14ac9bc49eb9702c7618b9e199fae05f23b788f9/internal/db/connect.go

code · 2026-10-18 · 本地完整读取：crush.db、WAL、共享连接池引用计数、SetMaxOpenConns(1)、可选数据目录进程锁；未自行复现注释提及的历史损坏。

[B21] Crush 中断工具调用收尾

https://github.com/charmbracelet/crush/blob/14ac9bc49eb9702c7618b9e199fae05f23b788f9/internal/agent/settle.go

code · 2026-10-18 · 本地完整读取：按结束原因写合成 error tool result，启动读取修复孤儿调用，dispatch lock 防止与仍在运行的调用竞争。

[B22] Crush RunComplete 去重

https://github.com/charmbracelet/crush/blob/14ac9bc49eb9702c7618b9e199fae05f23b788f9/internal/agent/run_marker.go

code · 2026-10-18 · 本地完整读取：通过 context 共享 atomic.Bool 标记权威 RunComplete，dispatch 仅缺失时补发。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[B23] Crush RunID 关联

https://github.com/charmbracelet/crush/blob/14ac9bc49eb9702c7618b9e199fae05f23b788f9/internal/agent/runid.go

code · 2026-10-18 · 本地完整读取：调用方 RunID 通过 context 传至 SessionAgentCall 并在终止事件回显，空 ID 回退 session 关联。

[B24] Crush 摘要提示词

https://github.com/charmbracelet/crush/blob/14ac9bc49eb9702c7618b9e199fae05f23b788f9/internal/agent/templates/summary.md

code · 2026-10-18 · 本地完整读取：Current State/Files & Changes/Technical Context/Strategy/Exact Next Steps 五类交接信息；无长度上限。

[B25] Crush 会话运行及摘要窗口

https://github.com/charmbracelet/crush/blob/14ac9bc49eb9702c7618b9e199fae05f23b788f9/internal/agent/agent.go

code · 2026-10-18 · 本地读取 Summarize、getSessionMessages、阈值与队列收尾，另浏览官方 GitHub 页面。摘要成功后更新 SummaryMessageID；只查询摘要之后尾部。

[B26] Crush 权限服务

https://github.com/charmbracelet/crush/blob/14ac9bc49eb9702c7618b9e199fae05f23b788f9/internal/permission/permission.go

code · 2026-10-18 · 本地读取 Request：skip/allowlist/hook 预批准、session/action/path 复用、请求串行与取消。

[B27] Pi 官方仓库与更名

https://github.com/badlogic/pi-mono

official · 2026-10-18 · 浏览确认重定向至 earendil-works/pi；main 固定快照 482c1381a00d8d00bc81697e4a980c593db6803e（2026-10-18）。

[B28] Pi 运行机制说明

https://github.com/earendil-works/pi/blob/482c1381a00d8d00bc81697e4a980c593db6803e/packages/coding-agent/docs/how-pi-works.md

official · 2026-10-18 · 已读取 agent loop、接口、树形 session、扩展以及操作系统权限边界。网页原 main；正文按本地固定快照复核。

[B29] Pi 系统提示词装配

https://github.com/earendil-works/pi/blob/482c1381a00d8d00bc81697e4a980c593db6803e/packages/coding-agent/src/core/system-prompt.ts

code · 2026-10-18 · 本地完整读取默认四工具、可见工具过滤、按工具合成规则并去重、XML 分段、section delta、forceSystemPrompt。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[B30] Pi Agent 核心循环

https://github.com/earendil-works/pi/blob/482c1381a00d8d00bc81697e4a980c593db6803e/packages/agent/src/agent-loop.ts

code · 2026-10-19 · 本地读取内外循环、prepareNextTurn、steering catch-up、follow-up、turn/agent 事件及并行/串行工具入口。 本轮重读工具 gate、顺序、改参和 callable/declared 筛选相关段落，细节见 night-02-kernel-policy-notes.md。

[B31] Pi 扩展生命周期契约

https://github.com/earendil-works/pi/blob/482c1381a00d8d00bc81697e4a980c593db6803e/packages/coding-agent/docs/extensions.md

official · 2026-10-19 · 已读取 hook 顺序、agent_end 与 agent_settled 的区别、handler 合成、慢流事件处理器造成背压、工具调用门禁。 本轮重读工具 gate、顺序、改参和 callable/declared 筛选相关段落，细节见 night-02-kernel-policy-notes.md。

[B32] Pi session v3 格式

https://github.com/earendil-works/pi/blob/482c1381a00d8d00bc81697e4a980c593db6803e/packages/coding-agent/docs/session-format.md

official · 2026-10-18 · 已读取 id/parentId、system prompt/tool delta、compaction firstKeptEntryId、context_edit、branch_summary；未将逻辑压缩等同内存释放。

[B33] Pi compaction 与分支摘要

https://github.com/earendil-works/pi/blob/482c1381a00d8d00bc81697e4a980c593db6803e/packages/coding-agent/docs/compaction.md

official · 2026-10-18 · 已读取触发条件、reserve 与 keep 分离、模型调用边界、tool pair 切点约束、overflow 恢复与 context_edit。

[B34] Pi 安全与项目信任

https://github.com/earendil-works/pi/blob/482c1381a00d8d00bc81697e4a980c593db6803e/packages/coding-agent/docs/security.md

official · 2026-10-18 · 已完整读取：同进程扩展继承 OS 权限；项目信任只控制资源装载，不形成工具权限或沙箱；AGENTS 等不受该门禁阻止。

[B35] Pi 内置 MCP

https://github.com/earendil-works/pi/blob/482c1381a00d8d00bc81697e4a980c593db6803e/packages/coding-agent/docs/mcp.md

official · 2026-10-19 · 已完整读取：stdio/streamable HTTP，direct/deferred/codemode/hidden 暴露，动态连接，嵌套调用走 tool pipeline。 本轮重读工具 gate、顺序、改参和 callable/declared 筛选相关段落，细节见 night-02-kernel-policy-notes.md。

[B36] Pi 官方 subagent 示例

https://github.com/earendil-works/pi/blob/482c1381a00d8d00bc81697e4a980c593db6803e/packages/coding-agent/examples/extensions/subagent/README.md

official · 2026-10-18 · 已完整读取：示例扩展，独立 pi 子进程，single/parallel/chain，8 个任务/4 并发，项目 agent 信任、失败传播与回传上限。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[B37] Pi subagent 示例源码

https://github.com/earendil-works/pi/blob/482c1381a00d8d00bc81697e4a980c593db6803e/packages/coding-agent/examples/extensions/subagent/index.ts

code · 2026-10-19 · 2026-10-19 重取固定 blob 843297454c6a46d5d66a06a31d1a8da17d48bf63：--no-session、直接子进程 SIGTERM、五秒 !proc.killed 检查及 close 等待；保留并发与50KiB输出合同。未运行取消实验。

[B38] Pi Code Mode

https://github.com/earendil-works/pi/blob/482c1381a00d8d00bc81697e4a980c593db6803e/packages/coding-agent/docs/codemode.md

official · 2026-10-18 · 已读取 QuickJS 无 FS/network/timer、脚本输出才入模型、工具发现、未完成调用取消、已完成副作用不回滚。

[B39] Pi 官方定位

https://pi.dev/

official · 2026-10-18 · 已读取官网 minimal harness、自定义扩展/skills/templates/packages、默认不带 subagents 与 plan mode 的定位。

[B40] Goose 官方仓库与归属

https://github.com/block/goose

official · 2026-10-18 · 浏览确认重定向 aaif-goose/goose；main 快照34a222e3f2b95d8b4a43482b53c2cc957ecce26d（2026-10-18）；官网确认 AAIF。

[B41] Goose 架构说明

https://goose-docs.ai/docs/goose-architecture/

official · 2026-10-18 · 已读取共享核心、接口和扩展的官方架构说明；详细执行机制以当前源码为准。

[B42] Goose PromptManager

https://github.com/aaif-goose/goose/blob/34a222e3f2b95d8b4a43482b53c2cc957ecce26d/crates/goose/src/agents/prompt_manager.rs

code · 2026-10-18 · 本地读取模板构造、session override/extras、hints、extension稳定排序、日期小时粒度及unicode标签清理。

[B43] Goose 主系统提示词

https://github.com/aaif-goose/goose/blob/34a222e3f2b95d8b4a43482b53c2cc957ecce26d/crates/goose/src/prompts/system.md

code · 2026-10-18 · 本地完整读取：基础身份、扩展条件段、动态扩展与响应格式；不夸大为复杂规划器。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[B44] Goose 子 agent 提示词

https://github.com/aaif-goose/goose/blob/34a222e3f2b95d8b4a43482b53c2cc957ecce26d/crates/goose/src/prompts/subagent_system.md

code · 2026-10-18 · 本地完整读取：scope、max_turns、工具清单、效率和最终交付要求；拒绝继续委派不只依赖该提示词。

[B45] Goose summon 委派机制

https://github.com/aaif-goose/goose/blob/34a222e3f2b95d8b4a43482b53c2cc957ecce26d/crates/goose/src/agents/platform_extensions/summon.rs

code · 2026-10-18 · 本地读取 load/delegate、explicit context、source转recipe、session持久化、extension过滤、工作目录检查及禁止递归。

[B46] Goose 子 session 执行

https://github.com/aaif-goose/goose/blob/34a222e3f2b95d8b4a43482b53c2cc957ecce26d/crates/goose/src/agents/subagent_handler.rs

code · 2026-10-19 · 2026-10-19 重取固定 blob 1e5002e336b53667318355774b08d47086f49b33：只转发 ToolConfirmation/Elicitation 决策；正常结束后读持久 final_output，成功优先于取消/运行错误；task drop 不保证走此路径。

[B47] Goose 状态机组装

https://github.com/aaif-goose/goose/blob/34a222e3f2b95d8b4a43482b53c2cc957ecce26d/crates/goose/src/agents/state_machine_builder.rs

code · 2026-10-18 · 本地读取有序 operation pipeline：审批、skills、recipe、执行、subagent、steer、compact、retry、stop、max turns 后推理。

[B48] Goose Recipe Schema

https://goose-docs.ai/docs/guides/recipes/recipe-reference/

official · 2026-10-18 · 已读取 instructions/prompt/parameters/extensions/response/retry/settings/sub_recipes；配方不是强确定性 DAG 调度器。

[B49] Goose Subrecipes

https://goose-docs.ai/docs/guides/recipes/subrecipes/

official · 2026-10-18 · 已读取子配方绑定与委派；官方标注实验性，参数固定值及源码优先级另由 summon 核实。

[B50] Goose 最终结构化输出工具

https://github.com/aaif-goose/goose/blob/34a222e3f2b95d8b4a43482b53c2cc957ecce26d/crates/goose/src/agents/final_output_tool.rs

code · 2026-10-18 · 本地读取 JSON schema 校验、与其他工具同批时拒绝、成功 tool response 才被视为最终输出。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[B51] Goose 压缩 Operation

https://github.com/aaif-goose/goose/blob/34a222e3f2b95d8b4a43482b53c2cc957ecce26d/crates/goose/src/agents/state_machine/ops_compaction.rs

code · 2026-10-18 · 本地读取未计入tool结果补估、模型自管context时跳过、reactive overflow、失败yield及最大重试限制。

[B52] Goose SQLite session 与压缩落盘

https://github.com/aaif-goose/goose/blob/34a222e3f2b95d8b4a43482b53c2cc957ecce26d/crates/goose/src/session/session_manager.rs

code · 2026-10-19 · 2026-10-19 重取固定 blob 1d9a8da79e4901b0d74499abb34c221c2bc08c1f：save_compacted_conversation自身BEGIN IMMEDIATE保历史改visible；主SessionLoader对应get_conversation按session_id fetch_all无visible过滤。

[B53] Goose 大输出落盘

https://github.com/aaif-goose/goose/blob/34a222e3f2b95d8b4a43482b53c2cc957ecce26d/crates/goose/src/agents/large_response_handler.rs

code · 2026-10-18 · 本地完整读取：超过200000字符落临时文件，返回路径，落盘失败返回原全文；不是预先流式限额。

[B54] Goose 工具审批 Operation

https://github.com/aaif-goose/goose/blob/34a222e3f2b95d8b4a43482b53c2cc957ecce26d/crates/goose/src/agents/state_machine/ops_tool_approval.rs

code · 2026-10-18 · 本地读取持久化审批请求/响应重建、tool advertised 校验、executable标记、inspection与人工许可流程。

[B55] Goose 公开 subagents 说明

https://goose-docs.ai/docs/guides/context-engineering/subagents/

official · 2026-10-18 · 已全文读取；其中“process isolation”、默认模式限制及上下文措辞与当前源码细节有差异，因此不作为OS隔离或默认继承全历史的证据。

[B56] Goose 当前前台子 agent Operation

https://github.com/aaif-goose/goose/blob/34a222e3f2b95d8b4a43482b53c2cc957ecce26d/crates/goose/src/agents/state_machine/ops_subagent.rs

code · 2026-10-19 · 2026-10-19 重取固定 blob 7d16d97cbdaac724f2dd26fb721948e5e073368a：校验父子身份、成功 final_output 复用及 stop 相关静态测试；测试仅阅读未执行。

[B57] OpenCode 会话父子继承

https://github.com/anomalyco/opencode/blob/802f2d8e75098ae0b45ad839ce98a31589ea8102/packages/core/src/session.ts

code · 2026-10-18 · 本地读取 create：子会话通过 parentID 复用父 Location，默认继承 metadata 与 session permissions；不自动创建独立 worktree。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[B58] 旧 OpenCode：归档与迁移身份

https://github.com/opencode-ai/opencode

official · 2026-10-19 · 2026-10-19 重新打开仓库；平台显示 2025-09-18 归档，README 指向 Crush，不能与 anomalyco/opencode 混淆。当前页面观察，未取得新的固定提交。

[B59] Crush 当前 FSL-1.1-MIT 许可

https://github.com/charmbracelet/crush/blob/14ac9bc49eb9702c7618b9e199fae05f23b788f9/LICENSE.md

official · 2026-10-19 · 2026-10-19 复核正文同一固定提交的许可证（只读文件接口，blob 950d0d85cee17e533b53572b19ba34d689d42680），并对照 current main；FSL 限制竞争用途，版本提供两周年后可使用 MIT；历史 MIT 归属段不代表全部当前代码。

[B60] Pi 声明与间接可调用集合

https://github.com/earendil-works/pi/blob/482c1381a00d8d00bc81697e4a980c593db6803e/packages/coding-agent/src/core/agent-session.ts

code · 2026-10-19 · 固定 482c1381；2026-10-19 重取 blob 8b330d555d0f45469fb291a2206217671d3977d5：沿用工具gate复核；另确认手动/自动compact先appendCompaction再刷新agent.state.messages，不清SessionManager原历史。

[B61] Pi tool_call Handler 顺序

https://github.com/earendil-works/pi/blob/482c1381a00d8d00bc81697e4a980c593db6803e/packages/coding-agent/src/core/extensions/runner.ts

code · 2026-10-19 · emitToolCall 顺序执行、block 短路；异常传到 agent-loop 形成该调用 error，不是必然整个会话崩溃 固定快照重新读取，未运行产品；完整 blob 与路径链见本轮研究笔记。

[B62] Goose 工具 Hook 调用路径

https://github.com/aaif-goose/goose/blob/34a222e3f2b95d8b4a43482b53c2cc957ecce26d/crates/goose/src/agents/state_machine/ops_toolcalling.rs

code · 2026-10-19 · PreToolUse 使用 blocking API，BeforeShellExecution/BeforeReadFile 和 PreToolUseResult 走观察型 emit 固定快照重新读取，未运行产品；完整 blob 与路径链见本轮研究笔记。

[B63] Goose Hook 故障策略

https://github.com/aaif-goose/goose/blob/34a222e3f2b95d8b4a43482b53c2cc957ecce26d/crates/goose/src/hooks/mod.rs

code · 2026-10-19 · OnFailure 默认 Allow；on_failure:block 的故障关闭仅在 PreToolUse 阻止路径生效 固定快照重新读取，未运行产品；完整 blob 与路径链见本轮研究笔记。

[B64] OpenCode job 与后台通知标记

https://github.com/anomalyco/opencode/blob/802f2d8e75098ae0b45ad839ce98a31589ea8102/packages/core/src/job.ts

code · 2026-10-19 · 固定开发快照只读重取，未运行；进程内 SynchronizedRef registry；独立 KV job.background 通知恢复标记；blob 771c5a79722effe0ae48d072b95c68f5f03be0ee。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[B65] OpenCode 子 job 观察与结果选择

https://github.com/anomalyco/opencode/blob/802f2d8e75098ae0b45ad839ce98a31589ea8102/packages/core/src/session/subagent-job.ts

code · 2026-10-19 · 固定开发快照只读重取，未运行；进程内 observer Set 不等于跨重启去重；resume 后最近20消息选完成非错误 assistant 文本；blob c64cb04732471657e2550b25d60d7a2f7d08a56d。

[B66] OpenCode 子结果向父输入交接

https://github.com/anomalyco/opencode/blob/802f2d8e75098ae0b45ad839ce98a31589ea8102/packages/core/src/session/subagent-completion.ts

code · 2026-10-19 · 固定开发快照只读重取，未运行；synthetic 父输入接纳后才清 KV 后台标记；沿用 notificationID，非父模型验收；blob 79a00e112aa2f7b6141cef08efd83e3d14fb69d1。

[B67] Node child_process：killed 不代表退出

https://nodejs.org/api/child_process.html#subprocesskilled

official · 2026-10-19 · 当前 Node v26.12.0 API 文档：killed 表示成功发送信号，不表示进程终止。用于解释 Pi 样例 Node 路径，不冒充其锁定运行时或 Bun/Windows 实测。

[B68] OpenCode Bus：投影提交与事件留存

https://github.com/anomalyco/opencode/blob/802f2d8e75098ae0b45ad839ce98a31589ea8102/packages/core/src/bus.ts

code · 2026-10-19 · 固定开发快照2026-10-19重取；blob 30bf10a31363954c800f352ad844416aa9e0dfd1：immediate事务内projector及seq，提交后route/wake；persist默认false仅控制完整event payload留存。未运行崩溃恢复测试。

[B69] OpenCode compaction projector 与 epoch

https://github.com/anomalyco/opencode/blob/802f2d8e75098ae0b45ad839ce98a31589ea8102/packages/core/src/session/projector.ts

code · 2026-10-19 · 固定开发快照2026-10-19重取；blob e007d6f4db86b5f8470076d3f7fb3c66588ab95e：Ended先更新completed摘要，再推进InstructionState epoch。未运行崩溃恢复测试。

[B70] OpenCode 压缩完成消息投影

https://github.com/anomalyco/opencode/blob/802f2d8e75098ae0b45ad839ce98a31589ea8102/packages/core/src/session/message-updater.ts

code · 2026-10-19 · 固定开发快照2026-10-19重取；blob 14143bbfa8e103661cdb322c57e92344df277ae6：running→completed或直接追加completed摘要，与Bus事务链合读。未运行崩溃恢复测试。

[B71] Pi SessionManager：全量读取与树索引

https://github.com/earendil-works/pi/blob/482c1381a00d8d00bc81697e4a980c593db6803e/packages/coding-agent/src/core/session-manager.ts

code · 2026-10-19 · 固定 blob df5281a0a4258462b14faa7d055efa09783ea6aa：分块JSONL读取仍累计全部fileEntries/byId；appendCompaction不删除旧条目，内存更新先于同步_persist。未测RSS或写盘失败上层恢复。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[B72] Goose SessionLoader 与效果持久化顺序

https://github.com/aaif-goose/goose/blob/34a222e3f2b95d8b4a43482b53c2cc957ecce26d/crates/goose/src/agents/state_machine/session.rs

code · 2026-10-19 · 固定 blob 37f092c29d68a03e220a669521e8f63ae05c2239：load→get_session(true)；usage、摘要事务、usage更新依次await，再HistoryReplaced。非整个操作单一事务；非所有UI入口审计。

[B73] Goose 压缩结果中的不可见历史对象

https://github.com/aaif-goose/goose/blob/34a222e3f2b95d8b4a43482b53c2cc957ecce26d/crates/goose/src/context_mgmt/mod.rs

code · 2026-10-19 · 固定 blob 63bd3a43ffc9ecb4b0cb7784efc5783a59469ced：clone原消息改agent_visible再构Conversation，仍保留旧对象；模型可见性不是内存释放。未测实际对象字节/RSS。

[B74] Goose：RecipeOperation 收集门槛

https://github.com/aaif-goose/goose/blob/34a222e3f2b95d8b4a43482b53c2cc957ecce26d/crates/goose/src/agents/state_machine/ops_recipe.rs

code · 2026-10-19 · 固定 34a222e3；blob 1fb2c3bb75f9b2d4fd34642200b5a19655ada851：Execute 路径检查同一 assistant 连续块中其他未外部执行的工具请求；成功 schema 只证明结构，Chat 模式先跳过。

[B75] Goose：可选 retry checks 的结束分支

https://github.com/aaif-goose/goose/blob/34a222e3f2b95d8b4a43482b53c2cc957ecce26d/crates/goose/src/agents/state_machine/ops_retry.rs

code · 2026-10-19 · 固定 34a222e3；blob a90c69a724200a9a882169952772l3a10f9f6463：ends_turn 后先处理 goal/grind，再在 retry_config 存在时运行成功检查；不是每份输出的无条件业务验收。

[C01] Cline：产品与 SDK 入口

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/README.md

official · 2026-10-18 · 固定提交 a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0；核实 SDK、CLI、Desktop、IDE 产品边界；JetBrains 插件并未开源。

[C02] Cline SDK：运行时分层

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/sdk/ARCHITECTURE.md

official · 2026-10-18 · 固定提交 a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0；核实 shared / llms / agents / core 的职责；agents 不持久化，core 管生命周期和存储。

[C03] Cline：Hub 与多客户端

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/docs/sdk/architecture/hub-spoke.mdx

official · 2026-10-18 · 固定提交 a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0；核实独立 Hub、客户端连接、共享会话和权限转发。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[C04] Cline：Act 提示词

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/sdk/packages/shared/src/prompt/system/act.ts

code · 2026-10-18 · 固定提交 a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0；核实环境占位符、先收集上下文、独立工具批量执行、验证及完成约定。

[C05] Cline：提示词装配

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/sdk/packages/shared/src/prompt/cline.ts

code · 2026-10-18 · 固定提交 a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0；核实模式标签、宿主是否提供切换工具、规则和元数据注入；不把注释当沙箱保证。

[C06] Cline：单次子智能体工具

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/sdk/packages/core/src/extensions/tools/team/spawn-agent-tool.ts

code · 2026-10-18 · 固定提交 a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0；核实 executionMode=parallel，执行体等待 child.run，返回迭代数、结束原因及用量。

[C07] Cline：委派配置继承

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/sdk/packages/core/src/extensions/tools/team/delegated-agent.ts

code · 2026-10-18 · 固定提交 a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0；核实父身份、连接设置、工具策略、审批回调、AbortSignal 的显式传递。

[C08] Cline：持久化团队

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/docs/sdk/guides/multi-agent-teams.mdx

official · 2026-10-18 · 固定提交 a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0；核实任务板、邮箱、任务运行、结果片段与会话 ID 下的 teams.db。

[C09] Cline：团队工具协议

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/sdk/packages/core/src/extensions/tools/team/team-tools.ts

code · 2026-10-19 · 2026-10-19 复读固定 blob 0335bb89ae121b3f77c7a28dd4206206af2d8eb4：team_cancel_run 直接调用 runtime.cancelRun，无独立成员 abort；sync/async、任务状态与结果片段合同保持。

[C10] Cline：异步结果回流

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/sdk/packages/core/src/session/team/team-session-coordinator.ts

code · 2026-10-19 · 2026-10-19 重取 blob 5bd295fd57abe4c30ad423350412ef38df025c98：RunCancelled 进入终态摘要、移除 active ID、通知等待，不直接 abort 成员；保留 completed/max_iterations 且未取消的父自动续接限定。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[C11] Cline：摘要压缩

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/sdk/packages/core/src/extensions/context/agentic-compaction.ts

code · 2026-10-19 · 2026-10-19 重取固定 blob 17624b2e6b826aeee7aeecd5658036b2874974b8：预算/尾部/文件操作描述保持；摘要userRunSpan记录运行跨度，不保存内部每条原消息。

[C12] Cline：基础压缩

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/sdk/packages/core/src/extensions/context/basic-compaction.ts

code · 2026-10-18 · 固定提交 a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0；核实预算投影与保留消息；仅据源码讨论，不声称内存已释放。

[C13] Cline：检查点恢复事务

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/sdk/packages/core/src/session/checkpoint-restore.ts

code · 2026-10-19 · 2026-10-19 重取固定 blob 574acd5c1025316a1fb6134f3f1089e4b2d99923：摘要跨度内部run消息恢复拒绝，restoreMessages:false可仅恢复文件；原HEAD检查与补偿边界保持，不宣称hard-kill自动事务恢复。

[C14] Cline：检查点生命周期

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/sdk/packages/core/src/hooks/checkpoint-hooks.ts

code · 2026-10-18 · 固定提交 a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0；核实 Git 检查点钩子、会话元数据与引用；不等同外部资源事务。

[C15] Roo Code：官方停运与仓库归档状态

https://github.com/RooCodeInc/Roo-Code

official · 2026-10-18 · 独立重开官方仓库：GitHub 顶部明确 owner 于 2026-05-15 archived/read-only，README 同时写明扩展 5 月 15 日 shutdown；另复核官方文档首页更新日期为 2026-05-15。

[C16] Roo：模式类型与内置角色

https://github.com/RooCodeInc/Roo-Code/blob/b867ec9145750d0ae1ff7f02d35406e9bf2a0b16/packages/types/src/mode.ts

code · 2026-10-18 · 固定提交 b867ec9145750d0ae1ff7f02d35406e9bf2a0b16；核实 roleDefinition / whenToUse / groups / fileRegex，Architect、Debug、Orchestrator 提示词。

[C17] Roo：系统提示词生成

https://github.com/RooCodeInc/Roo-Code/blob/b867ec9145750d0ae1ff7f02d35406e9bf2a0b16/src/core/prompts/system.ts

code · 2026-10-18 · 固定提交 b867ec9145750d0ae1ff7f02d35406e9bf2a0b16；核实 native-only 工具协议，toolsCatalog 为空，角色与规则等分段装配。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[C18] Roo：工具模式与模型适配

https://github.com/RooCodeInc/Roo-Code/blob/b867ec9145750d0ae1ff7f02d35406e9bf2a0b16/src/core/prompts/tools/filter-tools-for-mode.ts

code · 2026-10-18 · 固定提交 b867ec9145750d0ae1ff7f02d35406e9bf2a0b16；核实 excludedTools / includedTools / aliasRenames；模型加入工具仍须属于模式许可组。

[C19] Roo：new_task 实现

https://github.com/RooCodeInc/Roo-Code/blob/b867ec9145750d0ae1ff7f02d35406e9bf2a0b16/src/core/tools/NewTaskTool.ts

code · 2026-10-18 · 固定提交 b867ec9145750d0ae1ff7f02d35406e9bf2a0b16；核实参数、模式、待办校验、审批及 delegateParentAndOpenChild 唯一活跃子任务。

[C20] Roo：Boomerang 委派

https://github.com/RooCodeInc/Roo-Code/blob/b867ec9145750d0ae1ff7f02d35406e9bf2a0b16/apps/docs/docs/features/boomerang-tasks.mdx

official · 2026-10-18 · 固定提交 b867ec9145750d0ae1ff7f02d35406e9bf2a0b16；核实上下文显式下传、仅摘要上返、父任务暂停；这是序列式委派。

[C21] Roo：CLI 代理状态

https://github.com/RooCodeInc/Roo-Code/blob/b867ec9145750d0ae1ff7f02d35406e9bf2a0b16/apps/cli/docs/AGENT_LOOP.md

official · 2026-10-18 · 固定提交 b867ec9145750d0ae1ff7f02d35406e9bf2a0b16；核实 ClineMessage、ask/say、partial、状态推导及 CLI 默认自动批准。

[C22] Roo：检查点边界

https://github.com/RooCodeInc/Roo-Code/blob/b867ec9145750d0ae1ff7f02d35406e9bf2a0b16/apps/docs/docs/features/checkpoints.mdx

official · 2026-10-18 · 固定提交 b867ec9145750d0ae1ff7f02d35406e9bf2a0b16；核实影子 Git、文件或文件加任务恢复、排除规则、命令前不自动保存检查点。

[C23] Roo：影子 Git 服务

https://github.com/RooCodeInc/Roo-Code/blob/b867ec9145750d0ae1ff7f02d35406e9bf2a0b16/src/services/checkpoints/ShadowCheckpointService.ts

code · 2026-10-18 · 固定提交 b867ec9145750d0ae1ff7f02d35406e9bf2a0b16；核实仓库初始化、状态和恢复相关实现；不把所有项目文件概括为均可恢复。

[C24] Roo：任务委派与恢复宿主

https://github.com/RooCodeInc/Roo-Code/blob/b867ec9145750d0ae1ff7f02d35406e9bf2a0b16/src/core/webview/ClineProvider.ts

code · 2026-10-18 · 固定提交 b867ec9145750d0ae1ff7f02d35406e9bf2a0b16；核实父子关系和任务栈、父任务委派状态、重建和恢复子任务摘要路径。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[C25] Kilo：当前代码基线

https://github.com/Kilo-Org/kilocode/blob/2a1903f29186e25adec0bace14f6ee589095f7eb/README.md

official · 2026-10-18 · 固定提交 2a1903f29186e25adec0bace14f6ee589095f7eb；核实 CLI 为 OpenCode fork、产品入口及自主模式权限语义。

[C26] Kilo：CLI 服务架构

https://github.com/Kilo-Org/kilocode/blob/2a1903f29186e25adec0bace14f6ee589095f7eb/packages/kilo-docs/pages/contributing/architecture/cli-runtime.md

official · 2026-10-18 · 固定提交 2a1903f29186e25adec0bace14f6ee589095f7eb；核实服务复用、directory 路由、InstanceStore、SQLite 与进程共享状态。

[C27] Kilo：扩展宿主

https://github.com/Kilo-Org/kilocode/blob/2a1903f29186e25adec0bace14f6ee589095f7eb/packages/kilo-docs/pages/contributing/architecture/vscode-extension.md

official · 2026-10-18 · 固定提交 2a1903f29186e25adec0bace14f6ee589095f7eb；核实 ServerManager、SDK、SSE、目录和会话过滤、流式合并与断线恢复。

[C28] Kilo：按模型选提示词

https://github.com/Kilo-Org/kilocode/blob/2a1903f29186e25adec0bace14f6ee589095f7eb/packages/opencode/src/session/system.ts

code · 2026-10-18 · 固定提交 2a1903f29186e25adec0bace14f6ee589095f7eb；核实 model.prompt 优先、模型标识回退、skills 与 MCP 指令过滤。

[C29] Kilo：环境与项目记忆

https://github.com/Kilo-Org/kilocode/blob/2a1903f29186e25adec0bace14f6ee589095f7eb/packages/opencode/src/kilocode/system-prompt.ts

code · 2026-10-18 · 固定提交 2a1903f29186e25adec0bace14f6ee589095f7eb；核实记忆块注入、查询指引及当前用户/仓库优先于记忆的明文约定。

[C30] Kilo：自定义子智能体

https://github.com/Kilo-Org/kilocode/blob/2a1903f29186e25adec0bace14f6ee589095f7eb/packages/kilo-docs/pages/customize/custom-subagents.md

official · 2026-10-18 · 固定提交 2a1903f29186e25adec0bace14f6ee589095f7eb；核实 primary/subagent/all、描述字段、模型覆盖和 non-interactive 子会话。

[C31] Kilo：task 生命周期

https://github.com/Kilo-Org/kilocode/blob/2a1903f29186e25adec0bace14f6ee589095f7eb/packages/opencode/src/tool/task.ts

code · 2026-10-18 · 固定提交 2a1903f29186e25adec0bace14f6ee589095f7eb；核实父子身份校验、权限重算、background 完成注入、取消与费用传播。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[C32] Kilo：Orchestrator 弃用

https://github.com/Kilo-Org/kilocode/blob/2a1903f29186e25adec0bace14f6ee589095f7eb/packages/kilo-docs/pages/code-with-ai/agents/orchestrator-mode.md

official · 2026-10-18 · 固定提交 2a1903f29186e25adec0bace14f6ee589095f7eb；核实 dedicated orchestrator 已 deprecated；能力移入可委派的主智能体。

[C33] Kilo：双层编排和 Swarm

https://github.com/Kilo-Org/kilocode/blob/2a1903f29186e25adec0bace14f6ee589095f7eb/packages/kilo-docs/pages/automate/agent-manager.md

official · 2026-10-18 · 固定提交 2a1903f29186e25adec0bace14f6ee589095f7eb；核实 task 子会话、顶层 worktree 会话与共享板三者不同；板消息不唤醒智能体。

[C34] Kilo：权限策略

https://github.com/Kilo-Org/kilocode/blob/2a1903f29186e25adec0bace14f6ee589095f7eb/packages/kilo-docs/pages/customize/agent-permissions.md

official · 2026-10-18 · 固定提交 2a1903f29186e25adec0bace14f6ee589095f7eb；核实 allow/ask/deny、最后匹配优先及不同动作的权限模式。

[C35] Kilo：委派权限约束

https://github.com/Kilo-Org/kilocode/blob/2a1903f29186e25adec0bace14f6ee589095f7eb/packages/opencode/src/kilocode/tool/task.ts

code · 2026-10-18 · 固定提交 2a1903f29186e25adec0bace14f6ee589095f7eb；独立重取 inherited 实现：native-agent edit 与 notebook/MCP 等拒绝上界、session deny 传播；custom delegator edit 与广泛 caller bash deny 不自动继承，父 ask/allow 不覆盖子 profile。

[C36] Continue：最终发布说明

https://github.com/continuedev/continue/blob/5522c6f44ca0ac3528b37244818fbfa39b5af470/README.md

official · 2026-10-18 · 固定提交 5522c6f44ca0ac3528b37244818fbfa39b5af470；明确仓库不再积极维护且只读，最终 2.0.0，移除认证和匿名遥测；不混称 GitHub archive。

[C37] Continue：可组合配置

https://github.com/continuedev/continue/blob/5522c6f44ca0ac3528b37244818fbfa39b5af470/docs/guides/configuring-models-rules-tools.mdx

official · 2026-10-18 · 固定提交 5522c6f44ca0ac3528b37244818fbfa39b5af470；核实模型、规则、MCP 配置块与全局/工作区范围；共享云入口不视为仍受维护承诺。

[C38] Continue：规则载入

https://github.com/continuedev/continue/blob/5522c6f44ca0ac3528b37244818fbfa39b5af470/core/config/markdown/loadMarkdownRules.ts

code · 2026-10-18 · 固定提交 5522c6f44ca0ac3528b37244818fbfa39b5af470；核实规则来源元数据、alwaysApply、invokable 过滤；不泛化为所有兼容文件都正确回退。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[C39] Continue：规则匹配装配

https://github.com/continuedev/continue/blob/5522c6f44ca0ac3528b37244818fbfa39b5af470/core/llm/rules/getSystemMessageWithRules.ts

code · 2026-10-18 · 固定提交 5522c6f44ca0ac3528b37244818fbfa39b5af470；核实 glob、内容正则、全局规则、文件上下文选择与提示词拼装。

[C40] Continue：模式提示词

https://github.com/continuedev/continue/blob/5522c6f44ca0ac3528b37244818fbfa39b5af470/core/llm/defaultSystemMessages.ts

code · 2026-10-18 · 固定提交 5522c6f44ca0ac3528b37244818fbfa39b5af470；核实 Chat / Agent / Plan 行为、并行只读工具指令及 MCP 只读依赖模型判断的注释。

[C41] Continue：工具握手

https://github.com/continuedev/continue/blob/5522c6f44ca0ac3528b37244818fbfa39b5af470/docs/ide-extensions/agent/how-it-works.mdx

official · 2026-10-18 · 固定提交 5522c6f44ca0ac3528b37244818fbfa39b5af470；核实模式工具集、用户审批、内置工具或 MCP 调用、结果反馈。

[C42] Continue：子智能体执行器

https://github.com/continuedev/continue/blob/5522c6f44ca0ac3528b37244818fbfa39b5af470/extensions/cli/src/subagent/executor.ts

code · 2026-10-18 · 固定提交 5522c6f44ca0ac3528b37244818fbfa39b5af470；独立重取：共享策略先置 wildcard allow，提示构建在内层 try/finally 前；运行流 finally 恢复服务。初始化异常恢复缺口与并发覆盖为静态控制流推导，未执行复现。

[C43] Continue：子智能体工具

https://github.com/continuedev/continue/blob/5522c6f44ca0ac3528b37244818fbfa39b5af470/extensions/cli/src/tools/subagent.ts

code · 2026-10-18 · 固定提交 5522c6f44ca0ac3528b37244818fbfa39b5af470；核实动态工具名校验、父会话查找、等待执行与状态回传。

[C44] Continue：委派描述

https://github.com/continuedev/continue/blob/5522c6f44ca0ac3528b37244818fbfa39b5af470/extensions/cli/src/subagent/get-agents.ts

code · 2026-10-18 · 固定提交 5522c6f44ca0ac3528b37244818fbfa39b5af470；核实将子模型名称和 baseSystemMessage 嵌入委派工具描述。

[C45] Continue：规则触发文档

https://github.com/continuedev/continue/blob/5522c6f44ca0ac3528b37244818fbfa39b5af470/docs/customize/deep-dives/rules.mdx

official · 2026-10-18 · 固定提交 5522c6f44ca0ac3528b37244818fbfa39b5af470；核实规则前置元数据及上下文触发设计。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[C46] Continue：就地规则缓存

https://github.com/continuedev/continue/blob/5522c6f44ca0ac3528b37244818fbfa39b5af470/core/config/markdown/loadCodebaseRules.ts

code · 2026-10-18 · 固定提交 5522c6f44ca0ac3528b37244818fbfa39b5af470；核实工作区扫描、colocated rules.md、来源路径和缓存更新。

[C47] Tabby：产品范围

https://github.com/TabbyML/tabby/blob/21b29048d7bcf6b94f9f482f2d0fd05efadfd19f/README.md

official · 2026-10-18 · 固定提交 21b29048d7bcf6b94f9f482f2d0fd05efadfd19f；核实自托管补全/问答定位及 Pochi 集成条目；未见官方改名为 Pochi 的说明。

[C48] Tabby：LSP Agent

https://github.com/TabbyML/tabby/blob/21b29048d7bcf6b94f9f482f2d0fd05efadfd19f/clients/tabby-agent/README.md

official · 2026-10-18 · 固定提交 21b29048d7bcf6b94f9f482f2d0fd05efadfd19f；核实 tabby-agent 是 Node 语言服务器；标准 completion 与 tabby/* 扩展。

[C49] Tabby：客户端上下文预算

https://github.com/TabbyML/tabby/blob/21b29048d7bcf6b94f9f482f2d0fd05efadfd19f/clients/tabby-agent/src/codeCompletion/buildRequest.ts

code · 2026-10-18 · 固定提交 21b29048d7bcf6b94f9f482f2d0fd05efadfd19f；核实 prefix/suffix、声明、最近修改/打开片段、路径和范围去重。

[C50] Tabby：仓库级检索设计

https://github.com/TabbyML/tabby/blob/21b29048d7bcf6b94f9f482f2d0fd05efadfd19f/website/blog/2023-10-16-repository-context-for-code-completion/index.md

official · 2026-10-18 · 固定提交 21b29048d7bcf6b94f9f482f2d0fd05efadfd19f；历史设计：Tree-sitter 符号片段、BM25、代码注释注入；不当作最新全栈架构。

[C51] Tabby：排序融合设计

https://github.com/TabbyML/tabby/blob/21b29048d7bcf6b94f9f482f2d0fd05efadfd19f/website/blog/2024-06-11-rank-fusion-in-tabby-code-completion/index.md

official · 2026-10-18 · 固定提交 21b29048d7bcf6b94f9f482f2d0fd05efadfd19f；历史设计：BM25 与向量相似度通过 RRF 融合；不直接沿用文章性能宣传。

[C52] Tabby：服务端提示词预算

https://github.com/TabbyML/tabby/blob/21b29048d7bcf6b94f9f482f2d0fd05efadfd19f/crates/tabby/src/services/completion/completion_prompt.rs

code · 2026-10-18 · 固定提交 21b29048d7bcf6b94f9f482f2d0fd05efadfd19f；核实客户端片段优先、剩余预算后检索、允许仓库匹配及语言注释封装。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[C53] Tabby：补全调度

https://github.com/TabbyML/tabby/blob/21b29048d7bcf6b94f9f482f2d0fd05efadfd19f/clients/tabby-agent/src/codeCompletion/index.ts

code · 2026-10-18 · 固定提交 21b29048d7bcf6b94f9f482f2d0fd05efadfd19f；核实新请求取消旧请求、缓存/上下文采集/模型调用/后处理和延迟统计。

[C54] Tabby：补全缓存

https://github.com/TabbyML/tabby/blob/21b29048d7bcf6b94f9f482f2d0fd05efadfd19f/clients/tabby-agent/src/codeCompletion/cache.ts

code · 2026-10-18 · 固定提交 21b29048d7bcf6b94f9f482f2d0fd05efadfd19f；核实 LRU、上下文哈希含其他文档版本，以及连续输入的 forwarding contexts。

[C55] Tabby：自适应防抖

https://github.com/TabbyML/tabby/blob/21b29048d7bcf6b94f9f482f2d0fd05efadfd19f/clients/tabby-agent/src/codeCompletion/debouncer.ts

code · 2026-10-18 · 固定提交 21b29048d7bcf6b94f9f482f2d0fd05efadfd19f；核实输入间隔滑窗、上下文评分、预计响应时间、可取消等待及手动绕过。

[C56] Tabby：当前检索服务

https://github.com/TabbyML/tabby/blob/21b29048d7bcf6b94f9f482f2d0fd05efadfd19f/ee/tabby-webserver/src/service/retrieval.rs

code · 2026-10-18 · 固定提交 21b29048d7bcf6b94f9f482f2d0fd05efadfd19f；核实当前命中具有 bm25 / embedding / rrf 分数，结果使用 rrf 排序；企业目录代码需另审许可。

[C57] Continue：当前模式工具筛选器

https://github.com/continuedev/continue/blob/5522c6f44ca0ac3528b37244818fbfa39b5af470/gui/src/redux/selectors/selectActiveTools.ts

code · 2026-10-18 · 固定提交 5522c6f44ca0ac3528b37244818fbfa39b5af470；Plan 对内置组检查 readonly，其他已启用工具不经该 readonly 过滤；仍先遵守 disabled/exclude。

[C58] Cline：Plan 工具预设

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/sdk/packages/core/src/extensions/tools/presets.ts

code · 2026-10-19 · 本轮重新读取固定提交；plan 保留 Bash、禁用编辑工具。未运行产品；读取 blob 见 night-02-editor-notes.md。

[C59] Cline：Plan 命令守卫

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/sdk/packages/core/src/extensions/tools/command-guard-extension.ts

code · 2026-10-19 · 本轮重新读取固定提交；beforeTool 检查 run_commands，整批 skip 而非 stop。未运行产品；读取 blob 见 night-02-editor-notes.md。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[C60] Cline：Plan 黑名单语义

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/sdk/packages/core/src/extensions/tools/command-guard.ts

code · 2026-10-19 · 本轮重新读取固定提交；常见命令、参数和重定向黑名单；允许部分 tmp 输出；不解释任意 shell/解释器程序。未运行产品；读取 blob 见 night-02-editor-notes.md。

[C61] Cline：Plan 守卫运行注册

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/sdk/packages/core/src/runtime/orchestration/runtime-builder.ts

code · 2026-10-19 · 本轮重新读取固定提交；plan 且 enableTools 时实际注册守卫扩展。未运行产品；读取 blob 见 night-02-editor-notes.md。

[C62] Cline：Hook、策略与审批顺序

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/sdk/packages/agents/src/agent-runtime.ts

code · 2026-10-19 · 本轮重新读取固定提交；beforeTool→skip→policy→approval；需审批却无回调时拒绝。未运行产品；读取 blob 见 night-02-editor-notes.md。

[C63] Cline 当前 Plan 说明

https://docs.cline.bot/core-workflows/plan-and-act

official · 2026-10-19 · 页面仍写 Plan 不执行命令；与固定 SDK 快照有范围差异，不冒充运行验证。

[C64] Continue Plan 文档范围

https://docs.continue.dev/ide-extensions/agent/plan-mode

official · 2026-10-19 · 页面同时写只读与全部 MCP；与最终代码仅过滤内置组的边界一起理解，不冒充运行验证。

[C65] Cline team runtime：取消与成员中断

https://github.com/cline/cline/blob/a42df8c8d5fd9d9f9a9073589baffe32ea9bb2f0/sdk/packages/core/src/extensions/tools/team/multi-agent.ts

code · 2026-10-19 · 重取 blob 4c5c6685de77a35cd12a60c81064526fe18412aa：cancelRun 关闭记录/等待但不直接 abort；晚到成功/异常不覆盖 cancelled；父 cancelOutstandingWork 与 shutdownTeammate 另有 abort。交叉追踪内建监听，宿主自定义回调不在此保证内；未运行。

[D01] Aider：Repository map

https://aider.chat/docs/repomap.html

official · 2026-10-18 · 已打开正文；核实每次请求包含精选定义、签名和文件关系，地图预算随上下文调整。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[D02] Aider：仓库地图提取、排序与预算

https://github.com/Aider-AI/aider/blob/5dc9490bb35f9729ef2c95d00a19ccd30c26339c/aider/repomap.py

code · 2026-10-18 · 固定提交 5dc9490bb35f9729ef2c95d00a19ccd30c26339c；已读取 aider/repomap.py。核实 tree-sitter 标签、MultiDiGraph、个性化 PageRank、定义分数分配和按 token 预算搜索地图大小。

[D03] Aider：Tree-sitter repository maps

https://aider.chat/2023/10/22/repomap.html

official · 2026-10-18 · 已打开作者技术文章；用于解释由 ctags 转向 AST 提取的历史动机，不当作当前性能比较。

[D04] Aider：Edit formats

https://aider.chat/docs/more/edit-formats.html

official · 2026-10-18 · 已打开正文；核实 whole、diff、diff-fenced、udiff 与 editor 专用格式，格式选择与模型表现相关。

[D05] Aider：SEARCH/REPLACE 提示模板

https://github.com/Aider-AI/aider/blob/5dc9490bb35f9729ef2c95d00a19ccd30c26339c/aider/coders/editblock_prompts.py

code · 2026-10-18 · 固定提交 5dc9490bb35f9729ef2c95d00a19ccd30c26339c；已读取 aider/coders/editblock_prompts.py。核实编辑文件范围、请求尚未加入对话的文件、精确匹配及示例约束；正文仅概述，没有整段复制提示词。

[D06] Aider：编辑块解析与失败恢复

https://github.com/Aider-AI/aider/blob/5dc9490bb35f9729ef2c95d00a19ccd30c26339c/aider/coders/editblock_coder.py

code · 2026-10-18 · 固定提交 5dc9490bb35f9729ef2c95d00a19ccd30c26339c；独立重取：get_edits 先物化解析列表，apply_edits 逐块写入且部分失败不撤销成功块；存在前导空白容错、其他已入聊文件回退与相似行反馈。未做运行复现。

[D07] Aider：Chat modes

https://aider.chat/docs/usage/modes.html

official · 2026-10-18 · 已打开正文；核实 ask/code/architect 的交互目的和 architect 向编辑模型移交。

[D08] Aider：ArchitectCoder

https://github.com/Aider-AI/aider/blob/5dc9490bb35f9729ef2c95d00a19ccd30c26339c/aider/coders/architect_coder.py

code · 2026-10-18 · 固定提交 5dc9490bb35f9729ef2c95d00a19ccd30c26339c；已读取 aider/coders/architect_coder.py。核实 reply_completed、确认分支、editor_model、清空 editor 消息历史、map_tokens=0 和成本/提交信息回传。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[D09] Aider：Separating reasoning and editing

https://aider.chat/2024/09/26/architect.html

official · 2026-10-18 · 已打开作者文章；核实思考方案与产生编辑语法分离的设计动机，历史基准不视为当前通用性能。

[D10] Aider：Git integration

https://aider.chat/docs/git.html

official · 2026-10-18 · 已打开正文；核实自动提交、脏文件改动处理、undo/diff，以及默认跳过 Git hooks、可显式开启验证。

[D11] Aider：Linting and testing

https://aider.chat/docs/usage/lint-test.html

official · 2026-10-18 · 已打开正文；核实自动 lint、可配置测试命令、失败退出码和错误输出反馈、自动修复流程。

[D12] Aider：ChatSummary

https://github.com/Aider-AI/aider/blob/5dc9490bb35f9729ef2c95d00a19ccd30c26339c/aider/history.py

code · 2026-10-18 · 固定提交 5dc9490bb35f9729ef2c95d00a19ccd30c26339c；已读取 aider/history.py。核实按 token 预算切分近期尾部与较早消息、递归摘要、边界处理、摘要模型回退及摘要消息封装。

[D13] Aider：Benchmark methodology

https://aider.chat/docs/benchmarks.html

official · 2026-10-18 · 已打开正文；核实先实现，再把失败测试反馈给模型进行第二次尝试，累计通过率不可直接视为 pass@1。

[D14] Aider：Polyglot leaderboard methodology

https://aider.chat/docs/leaderboards/

official · 2026-10-18 · 已打开正文；仅使用任务数、模型/编辑格式/提交日期/语法失败等披露字段，不引用动态排名。

[D15] OpenHands SDK：Architecture overview

https://docs.openhands.dev/sdk/arch/overview

official · 2026-10-18 · 已打开正文；核实 sdk/tools/workspace/agent-server 的包职责，本地与远程运行模式及 Agent Server 接口。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[D16] OpenHands SDK：Agent step 实现

https://github.com/OpenHands/software-agent-sdk/blob/6ae64079cfd30cfe2f01a060c59168a4d650d883/openhands-sdk/openhands/sdk/agent/agent.py

code · 2026-10-18 · 固定提交 6ae64079cfd30cfe2f01a060c59168a4d650d883；已读取 openhands-sdk/openhands/sdk/agent/agent.py。核实 pending actions、state.view、模型元数据解析、condensation 事件返回路径，以及异步工具调度。

[D17] OpenHands SDK：Events

https://docs.openhands.dev/sdk/arch/events

official · 2026-10-18 · 已打开正文；核实不可变追加事件、动作/观察配对、LLM 可见事件转换、并行工具响应重组。

[D18] OpenHands SDK：Conversation

https://docs.openhands.dev/sdk/arch/conversation

official · 2026-10-18 · 已打开正文；核实本地/远程会话、可持久化 EventLog、ConversationState 的状态职责与投影。

[D19] OpenHands SDK：Tool system

https://docs.openhands.dev/sdk/arch/tool-system

official · 2026-10-18 · 已打开正文；核实 Action/Observation/Executor/ToolDefinition 四层以及工具注解并非强制授权。

[D20] OpenHands SDK：Skills

https://docs.openhands.dev/sdk/arch/skill

official · 2026-10-18 · 已打开正文；核实常驻、触发、路径规则、ACP 规则归属、动态技能可执行内联 shell 命令。

[D21] OpenHands SDK：提示词预设组成

https://github.com/OpenHands/software-agent-sdk/blob/6ae64079cfd30cfe2f01a060c59168a4d650d883/openhands-sdk/openhands/sdk/context/prompts/presets.py

code · 2026-10-18 · 固定提交 6ae64079cfd30cfe2f01a060c59168a4d650d883；已读取 openhands-sdk/openhands/sdk/context/prompts/presets.py。核实 DEFAULT/PLANNING 的 section 注册、人格替换范围、动态技能等章节，以及 DateTime 位于动态部分末尾。

[D22] OpenHands SDK：Condenser

https://docs.openhands.dev/sdk/arch/condenser

official · 2026-10-18 · 已打开正文；核实保留头尾、摘要中段、通过 Condensation 事件重建模型视图，支持组合策略。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[D23] OpenHands SDK：Condenser 接口与故障语义

https://github.com/OpenHands/software-agent-sdk/blob/6ae64079cfd30cfe2f01a060c59168a4d650d883/openhands-sdk/openhands/sdk/context/condenser/base.py

code · 2026-10-18 · 固定提交 6ae64079cfd30cfe2f01a060c59168a4d650d883；已读取 openhands-sdk/openhands/sdk/context/condenser/base.py。核实只读 View 约定、View/Condensation 返回类型、软/硬压缩需求和无法满足硬需求时的异常。

[D24] OpenHands SDK：Security guide

https://docs.openhands.dev/sdk/guides/security

official · 2026-10-18 · 已打开正文；核实会话确认策略、风险分析器、策略可配置；模式分析器不能替代 shell 解析、注入防护或沙箱。

[D25] OpenHands SDK：Task Tool Set

https://docs.openhands.dev/sdk/guides/task-tool-set

official · 2026-10-18 · 已打开正文；核实同步阻塞子任务、任务 ID 恢复和子代理注册；当前默认值以固定源码为准。

[D26] OpenHands Software Agent SDK paper

https://arxiv.org/html/2511.03690v2

paper · 2026-10-18 · 已打开原论文 HTML；用于解释 SDK 事件化与可组合的历史设计。论文单代理局限已被后续 TaskTool 源码部分超越。

[D27] SWE-agent：Agent-computer interface

https://swe-agent.com/latest/background/aci/

official · 2026-10-18 · 已打开正文；页面标注 maintenance-only，并介绍受约束搜索、窗口查看、编辑反馈等经典 ACI。

[D28] SWE-agent：Agent-Computer Interfaces Enable Automated Software Engineering

https://arxiv.org/html/2405.15793v3

paper · 2026-10-18 · 已打开原论文 HTML；核实历史 ACI 设计和 GPT-4 Turbo / SWE-bench Lite 消融，不将结果外推至当前模型。

[D29] SWE-agent：当前 default 配置

https://github.com/SWE-agent/SWE-agent/blob/3ea751c087f32b16e039a2233dd6eefecef325d5/config/default.yaml

code · 2026-10-18 · 固定提交 3ea751c087f32b16e039a2233dd6eefecef325d5；已读取 config/default.yaml。核实任务模板、函数调用、bash、工具包、分页器环境配置和 cache_control 历史处理器；模板含基准专用不改测试指令。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[D30] SWE-agent：Architecture

https://swe-agent.com/latest/background/architecture

official · 2026-10-18 · 已打开正文；核实 SWEEnv/SWE-ReX 环境层、工具安装、持久 shell 与 history processor。未把文档方法名当作当前源码方法名。

[D31] SWE-agent：History processors

https://github.com/SWE-agent/SWE-agent/blob/3ea751c087f32b16e039a2233dd6eefecef325d5/sweagent/agent/history_processors.py

code · 2026-10-18 · 固定提交 3ea751c087f32b16e039a2233dd6eefecef325d5；已读取 sweagent/agent/history_processors.py。核实 LastNObservations、省略旧观察、标签控制、缓存控制与每轮改写历史可能破坏前缀缓存的权衡。

[D32] SWE-agent：Trajectories

https://swe-agent.com/latest/usage/trajectories

official · 2026-10-18 · 已打开正文；核实轨迹中的模型响应、动作、观察、查询和状态，以及配置、补丁与提交记录。

[D33] SWE-agent：Competitive runs

https://swe-agent.com/latest/usage/competitive_runs

official · 2026-10-18 · 已打开正文；核实多次独立尝试、选择器、成本控制和资源清理注意点；示例尝试数以所引用固定配置为准。

[D34] SWE-agent：Anthropic 编辑工具 schema

https://github.com/SWE-agent/SWE-agent/blob/3ea751c087f32b16e039a2233dd6eefecef325d5/tools/edit_anthropic/config.yaml

code · 2026-10-18 · 固定提交 3ea751c087f32b16e039a2233dd6eefecef325d5；已读取 tools/edit_anthropic/config.yaml。核实 view/create/str_replace/insert/undo_edit、唯一匹配要求和单文件最近编辑撤销。

[D35] mini-SWE-agent：v2 migration

https://mini-swe-agent.com/latest/advanced/v2_migration

official · 2026-10-18 · 已打开正文；核实默认原生工具调用、模型适配器负责动作/观察转换、成本归属代理、返回数据和 v1 的变化。

[D36] mini-SWE-agent：DefaultAgent

https://github.com/SWE-agent/mini-swe-agent/blob/04d809ceab9df28f9adaed044884180159172930/src/minisweagent/agents/default.py

code · 2026-10-18 · 固定提交 04d809ceab9df28f9adaed044884180159172930；已读取 src/minisweagent/agents/default.py。核实扁平消息列表、查询/执行循环、成本/步骤/时间限额、格式错误计费与恢复、finally 保存。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[D37] mini-SWE-agent：LiteLLM 模型适配器

https://github.com/SWE-agent/mini-swe-agent/blob/04d809ceab9df28f9adaed044884180159172930/src/minisweagent/models/litellm_model.py

code · 2026-10-18 · 固定提交 04d809ceab9df28f9adaed044884180159172930；已读取 src/minisweagent/models/litellm_model.py。核实 BASH_TOOL、原生消息及 extra 元数据、工具动作解析、观察格式化和上下文溢出异常策略。

[D38] mini-SWE-agent：默认交互提示模板

https://github.com/SWE-agent/mini-swe-agent/blob/04d809ceab9df28f9adaed044884180159172930/src/minisweagent/config/mini.yaml

code · 2026-10-18 · 固定提交 04d809ceab9df28f9adaed044884180159172930；已读取 src/minisweagent/config/mini.yaml。核实 Bash 工具调用要求、独立 shell 语义、任务完成标记、按头尾截断观察和格式修复提示。

[D39] mini-SWE-agent：LocalEnvironment

https://github.com/SWE-agent/mini-swe-agent/blob/04d809ceab9df28f9adaed044884180159172930/src/minisweagent/environments/local.py

code · 2026-10-18 · 固定提交 04d809ceab9df28f9adaed044884180159172930；已读取 src/minisweagent/environments/local.py。核实 Popen、独立 shell、环境变量继承、超时终止进程组、结构化异常与完成标记判定。

[D40] mini-SWE-agent：Environments

https://mini-swe-agent.com/latest/advanced/environments

official · 2026-10-18 · 已打开正文；核实本地和容器等可互换环境，不能由统一接口推导相同隔离能力。

[D41] mini-SWE-agent：SWE-bench runs

https://mini-swe-agent.com/latest/usage/swebench

official · 2026-10-18 · 已打开正文；核实多个实例并行、预测文件生成与外部评测命令，单例调试和批量输出用途不同。

[D42] mini-SWE-agent：FAQ（存在过时描述）

https://mini-swe-agent.com/latest/faq

official · 2026-10-18 · 已打开正文；FAQ 的“不使用工具调用”等描述与 v2 文档/固定源码冲突，仅用于版本证据辨析。

[D43] OpenHands SDK：AgentBase prompt 构建

https://github.com/OpenHands/software-agent-sdk/blob/6ae64079cfd30cfe2f01a060c59168a4d650d883/openhands-sdk/openhands/sdk/agent/base.py

code · 2026-10-18 · 固定提交 6ae64079cfd30cfe2f01a060c59168a4d650d883；已读取 openhands-sdk/openhands/sdk/agent/base.py。核实旧 Jinja 模板名作为预设兼容标记、registry 构建 static/dynamic 部分、工具指导去重及自定义模板回退。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[D44] OpenHands SDK：TaskAction 与委派描述

https://github.com/OpenHands/software-agent-sdk/blob/6ae64079cfd30cfe2f01a060c59168a4d650d883/openhands-tools/openhands/tools/task/definition.py

code · 2026-10-18 · 固定提交 6ae64079cfd30cfe2f01a060c59168a4d650d883；已读取 openhands-tools/openhands/tools/task/definition.py。核实子代理类型、恢复 ID、任务提示要求、委派粒度建议、工具注解与已忽略的 max_turns 字段。

[D45] OpenHands SDK：TaskManager

https://github.com/OpenHands/software-agent-sdk/blob/6ae64079cfd30cfe2f01a060c59168a4d650d883/openhands-tools/openhands/tools/task/manager.py

code · 2026-10-18 · 固定提交 6ae64079cfd30cfe2f01a060c59168a4d650d883；独立重取：子会话创建/续接共享父 working_dir；暂停关闭后清对象引用；任务指标覆盖累计。_tasks 是内存字典，_resume_task 首先依赖其 task_id 映射，不能仅据子会话落盘宣称跨进程注册表恢复。

[D46] SWE-agent：heavy 运行配置快照

https://github.com/SWE-agent/SWE-agent/blob/3ea751c087f32b16e039a2233dd6eefecef325d5/config/benchmarks/250212_sweagent_heavy_sbl.yaml

code · 2026-10-18 · 固定提交 3ea751c087f32b16e039a2233dd6eefecef325d5；已读取 config/benchmarks/250212_sweagent_heavy_sbl.yaml。核实 retry/chooser、多配置候选、max_attempts=10 和静态预算参数；这是历史命名实验配置，不是当前价格。

[D47] mini-SWE-agent：SWE-bench runner

https://github.com/SWE-agent/mini-swe-agent/blob/04d809ceab9df28f9adaed044884180159172930/src/minisweagent/run/benchmarks/swebench.py

code · 2026-10-18 · 固定提交 04d809ceab9df28f9adaed044884180159172930；已读取 src/minisweagent/run/benchmarks/swebench.py。核实 ThreadPoolExecutor 按实例并行、每实例文件、重跑先删旧预测、保存轨迹再更新预测与依据预测跳过完成实例。

[D48] OpenHands SDK：Agent architecture

https://docs.openhands.dev/sdk/arch/agent

official · 2026-10-18 · 已打开正文；核实推理控制器与可变会话状态的分离、先处理待执行动作、再调用模型的执行次序。

[D49] OpenHands State：分支投影与恢复

https://github.com/OpenHands/software-agent-sdk/blob/6ae64079cfd30cfe2f01a060c59168a4d650d883/openhands-sdk/openhands/sdk/conversation/state.py

code · 2026-10-19 · 固定快照2026-10-19只读重取；blob ed774175926db0f574d920f59fb096594b09d3dd：冷load后rebuild_view→path_to_root，逐事件物化活动分支；event append与HEAD autosave不是跨文件原子提交。未运行/测量。

[D50] OpenHands EventLog：物化事件缓存

https://github.com/OpenHands/software-agent-sdk/blob/6ae64079cfd30cfe2f01a060c59168a4d650d883/openhands-sdk/openhands/sdk/conversation/event_store.py

code · 2026-10-19 · 固定快照2026-10-19只读重取；blob ddfd38f13662ddb7c28e4d5142862e91d05716ca：普通dict _event_cache，无容量驱逐；重扫索引可clear，不写永久不可清空。未运行/测量。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[D51] OpenHands LocalFileStore：独立文本LRU

https://github.com/OpenHands/software-agent-sdk/blob/6ae64079cfd30cfe2f01a060c59168a4d650d883/openhands-sdk/openhands/sdk/io/local.py

code · 2026-10-19 · 固定快照2026-10-19只读重取；blob b31d270068557f125b45e48095c7ebbc3c0825fc：MemoryLRUCache文本层独立于EventLog对象字典，不能约束后者驻留量。未运行/测量。

[D52] OpenHands agent-server：中断恢复对账

https://github.com/OpenHands/software-agent-sdk/blob/6ae64079cfd30cfe2f01a060c59168a4d650d883/openhands-agent-server/openhands/agent_server/event_service.py

code · 2026-10-19 · 固定快照2026-10-19只读重取；blob ef67d132ad708e67e01a4582e4fab3ab9a54f9b4：stale RUNNING→ERROR，扫完整日志，对首个unmatched action补非retryable错误，显式parent纠正HEAD滞后窗口；不是无条件重放全部动作。未运行/测量。

[E01] Cursor Rules

https://cursor.com/docs/rules

official · 2026-10-18 · 核实 MDC 激活字段、团队/项目/用户优先级、AGENTS.md 层级以及引用文件不自动内联。

[E02] Cursor Prompting agents

https://cursor.com/docs/agent/prompting

official · 2026-10-18 · 核实上下文分类展示、压缩摘要、Skill 单次调用与持续 Custom Mode 的区别。

[E03] Cursor Subagents

https://cursor.com/docs/subagents

official · 2026-10-19 · 2026-10-19 复读当前官方文档：默认共享父 checkout，并发可覆盖；请求 isolation 才分配各自 worktree/VM。上下文隔离、前后台、恢复 ID 与云 MCP 来源分别说明。

[E04] Cursor Cloud Agents

https://cursor.com/docs/cloud-agent

official · 2026-10-18 · 核实独立 VM、分支交接、多仓环境、团队 MCP、Hooks 与远程执行入口。

[E05] Cursor Cloud Environment Setup

https://cursor.com/docs/cloud-agent/setup

official · 2026-10-18 · 核实 Build 提前运行 install、磁盘快照不含进程、start/terminals 及失败 Build 保留旧环境。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[E06] Scaling long-running autonomous coding

https://cursor.com/blog/scaling-agents

official · 2026-10-18 · 2026-01-14 官方研究博客；核实共享锁失败、planner/worker/judge 分工与周期性新上下文，属于实验报告。

[E07] Cursor Secrets & Network

https://cursor.com/docs/cloud-agent/security-network

official · 2026-10-18 · 核实托管云 VM 默认有互联网、可设出站域名控制；自托管机器采用自身网络控制。

[E08] Cursor Agent Security

https://cursor.com/docs/agent/security

official · 2026-10-18 · 核实默认敏感动作审批；Run Modes 与 Auto-review 为尽力防护，并非硬隔离边界。

[E09] Cascade Memories & Rules

https://docs.devin.ai/desktop/cascade/memories

official · 2026-10-18 · 旧 Windsurf URL 重定向至此；核实 Cascade 的本地记忆、规则触发、.devin 与 .windsurf 兼容。

[E10] Cascade workflows

https://docs.devin.ai/desktop/cascade/workflows

official · 2026-10-18 · 核实手动 slash 调用、顺序任务、规则与流程区别；Devin Local 不支持旧 Workflows。

[E11] Cascade Modes

https://docs.devin.ai/desktop/cascade/modes

official · 2026-10-18 · 核实 Code/Plan/Ask 能力表；Plan 外置 Markdown，Ask 仅搜索；Plan 表中仍列全部工具。

[E12] Devin Desktop Worktrees

https://docs.devin.ai/desktop/cascade/worktrees

official · 2026-10-18 · 核实独立工作目录、会话开头选择、未跟踪文件缺失和 post_setup_worktree 环境恢复。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[E13] Cascade Hooks

https://docs.devin.ai/desktop/cascade/hooks

official · 2026-10-18 · 核实 stdin JSON、pre-hook exit 2 阻断、其他非零继续、post-hook 不可回滚、transcript_path。

[E14] Devin Local Agent

https://docs.devin.ai/desktop/devin-local

official · 2026-10-18 · 核实未选偏好且可用时新会话默认 Devin Local；企业需启用；与 CLI 共用 harness、独立 subagent 链、可选 OS sandbox；旧记忆/Workflow 不兼容。

[E15] Devin Desktop Agent Command Center

https://docs.devin.ai/desktop/agent-command-center

official · 2026-10-18 · 核实本地/云会话看板、状态与 Space 聚合；界面聚合不等于执行层多智能体协同。

[E16] Devin Desktop Agent Client Protocol

https://docs.devin.ai/desktop/acp

official · 2026-10-18 · 核实 ACP 插接第三方 agent、本地 registry 只启动已安装 binary、第三方自行负责 agent 操作。

[E17] GitHub repository custom instructions

https://docs.github.com/en/copilot/how-tos/copilot-on-github/customize-copilot/add-custom-instructions/add-repository-instructions

official · 2026-10-18 · 核实仓库规则、applyTo 路径规则、就近 AGENTS.md，以及官方 onboarding 提示要求实际验证命令。

[E18] GitHub custom agents configuration

https://docs.github.com/en/copilot/reference/custom-agents-configuration

official · 2026-10-18 · 核实工具过滤、空列表/缺省区别、agent 工具、跨端字段差异、profile 基于 commit SHA 版本。

[E19] GitHub Copilot on GitHub.com

https://docs.github.com/en/copilot/concepts/copilot-surfaces/copilot-on-github

official · 2026-10-18 · 核实 cloud agent 临时环境、聊天交接、一次单仓单分支单 PR 及 59 分钟会话上限。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[E20] GitHub cloud agent risks and mitigations

https://docs.github.com/en/copilot/concepts/security-governance-and-network-settings/risks-and-mitigations

official · 2026-10-18 · 核实分支凭据、人工合并、默认 Actions 审批、默认 CodeQL/依赖/secret 检查及审计。

[E21] GitHub Copilot development environment

https://docs.github.com/en/copilot/how-tos/copilot-on-github/customize-copilot/customize-cloud-agent/customize-the-agent-environment

official · 2026-10-18 · 核实 Actions ephemeral runner、固定 setup job、失败后剩余 setup 跳过但 agent 继续、自托管网络责任。

[E22] GitHub Copilot Memory

https://docs.github.com/en/copilot/concepts/agents/copilot-memory

official · 2026-10-18 · 核实公开预览、仓库事实与用户偏好分离、代码证据校验、28 天未用清理、跨功能共享限制。

[E23] GitHub cloud agent MCP

https://docs.github.com/en/copilot/concepts/agents/cloud-agent/mcp-and-cloud-agent

official · 2026-10-18 · 核实仅 tools、不支持 resources/prompts 与 OAuth 远程服务；默认 GitHub 只读及 Playwright localhost。

[E24] GitHub About custom agents

https://docs.github.com/en/copilot/concepts/agents/cloud-agent/about-custom-agents

official · 2026-10-18 · 核实 Markdown profile、工具/MCP/提示配置、repo/org/enterprise 等作用域及跨表面兼容限制。

[E25] Devin Knowledge migration

https://docs.devin.ai/product-guides/knowledge

official · 2026-10-18 · 核实 Knowledge 弃用并迁到托管 Plugins 中 Skills；触发/正文/作用域映射而不自动写用户仓库。

[E26] Devin Skills

https://docs.devin.ai/product-guides/skills

official · 2026-10-18 · 核实 Agent Skills 格式、索引和磁盘合并、描述按需载入、允许工具、当前仅一个 active skill。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[E27] Devin Creating Playbooks

https://docs.devin.ai/product-guides/creating-playbooks

official · 2026-10-18 · 核实 Procedure/Specifications/Advice/Forbidden/Required 输入结构、手动附加与版本历史。

[E28] Devin Advanced Capabilities

https://docs.devin.ai/work-with-devin/advanced-capabilities

official · 2026-10-18 · 核实 managed Devin 独立 VM、协调者发消息/限额/睡眠终止以及子任务完成或求助唤醒。

[E29] Devin Dynamic Workflows

https://docs.devin.ai/work-with-devin/dynamic-workflows

official · 2026-10-18 · 核实 Python 编排、JSON Schema、pipeline 无阶段屏障、run ID 与调用哈希重放；恢复要求脚本确定性；默认独立 VM 也可共享 VM。

[E30] Devin environment setup

https://docs.devin.ai/onboard-devin/environment

official · 2026-10-18 · 核实 blueprint 构建 snapshot、每次 session 新副本、会话变更不回写基础快照。

[E31] Devin Security Profiles

https://docs.devin.ai/product-guides/security-profiles

official · 2026-10-18 · 核实限制相交、子会话继承、boot/wake 生效时点以及 Outposts 网络需客户执行。

[E32] Devin Testing & Video Recordings

https://docs.devin.ai/work-with-devin/testing-and-recordings

official · 2026-10-18 · 核实环境准备、diff 驱动单主路径计划、浏览器录像交付及成功经验提出 Skill PR。

[E33] Kiro Specs

https://kiro.dev/docs/specs/

official · 2026-10-18 · 页面更新 2026-10-15；核实 requirements/design/tasks、Feature/Bugfix/Quick Spec 与依赖 wave 并行。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[E34] Kiro Correctness

https://kiro.dev/docs/specs/correctness/

official · 2026-10-18 · 页面更新 2026-08-04；核实由 EARS 提炼性质、生成测试、缩减反例；PBT 可选且非形式证明。

[E35] Kiro Steering

https://kiro.dev/docs/steering/

official · 2026-10-18 · 核实工作区/全局、三份基础文档、四种注入方式、自定义 agent 需显式 resources 和云同步边界。

[E36] Kiro Hooks

https://kiro.dev/docs/hooks/

official · 2026-10-18 · 页面更新 2026-10-10；核实 IDE 1.0/CLI V3 JSON 格式、命令/agent 动作、stdin、事件与超时。

[E37] Kiro Custom agents

https://kiro.dev/docs/custom-agents/

official · 2026-10-18 · 核实 profile 的 tools/permissions/resources、JSON/Markdown、Web 仅自定义子 agent 和 V3 迁移。

[E38] Kiro local command sandbox

https://kiro.dev/docs/sandbox/

official · 2026-10-18 · 核实 shell/子进程与可选 stdio MCP 的 OS 边界、边界外组件；必须启用且后端可用，后端缺失可无隔离运行（要求保护的模式例外）。

[E39] Kiro Autonomous mode

https://kiro.dev/docs/web/autonomous-mode/

official · 2026-10-18 · 页面更新 2026-10-02；核实默认关闭、规划/委派/PR、多仓任务、Needs attention 和显式 workflow 区别。

[E40] Upgrading from Amazon Q Developer CLI

https://kiro.dev/docs/upgrade-guides/migrating-from-q/

official · 2026-10-18 · 页面更新 2026-10-15；核实 Kiro CLI 继承 Q CLI、新特性在 Kiro、许可与配置兼容区别。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[E41] Migrating from Amazon Q Developer IDE

https://kiro.dev/docs/upgrade-guides/migrating-from-q-developer/

official · 2026-10-18 · 页面更新 2026-08-04；核实 Q IDE 与 Q CLI 分开迁移；其 CLI spec 表格已落后于新 Specs 主文档。

[E42] Kiro IDE 与 CLI 当前许可页

https://kiro.dev/license/

official · 2026-10-19 · 2026-10-19 打开官方许可页；IDE 与 CLI 均作为 AWS Content 受 AWS Intellectual Property License 约束；开放组件归属清单不等于整个产品开源。

[E43] Cursor Worktrees：候选比较与应用

https://cursor.com/docs/configuration/worktrees

official · 2026-10-19 · 2026-10-19 当前官方说明：/best-of-n 仅比较独立候选，不自动合并；/apply-worktree 单独应用；不推荐依赖软链接回主工作树。未公开合并算法或 dirty workspace 原子性。

[F01] Qwen Code 当前仓库

https://github.com/QwenLM/qwen-code

official · 2026-10-18 · README 宣告 Agent Teams、多协议与多入口；当前 commit 46bf492d05f4520bfd918c62e104d45f433da493；不采用仓库自报横向优劣或性能数字。

[F02] Qwen Code 架构边界

https://github.com/QwenLM/qwen-code/blob/46bf492d05f4520bfd918c62e104d45f433da493/docs/developers/architecture.md

official · 2026-10-18 · 核实 CLI / core / ACP 桥接职责及独立运行时。 文档固定于 URL 所含 commit，读取于 2026-10-18。

[F03] Qwen Code 提示词装配

https://github.com/QwenLM/qwen-code/blob/46bf492d05f4520bfd918c62e104d45f433da493/packages/core/src/core/prompts.ts

code · 2026-10-18 · 核实 resolveInteractionMode、PromptToolSurface、gateToolGuidance、filterToolCallExamples；工具可见性与提示一致。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F04] Qwen 团队角色提示

https://github.com/QwenLM/qwen-code/blob/46bf492d05f4520bfd918c62e104d45f433da493/packages/core/src/agents/team/promptAddendum.ts

code · 2026-10-18 · 核实 buildTeammatePromptAddendum 的普通、planModeRequired、readOnly 分支及最终回复自动转发说明。 源码固定于 URL 所含 commit，读取于 2026-10-18。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[F05] Qwen 团队邮箱

https://github.com/QwenLM/qwen-code/blob/46bf492d05f4520bfd918c62e104d45f433da493/packages/core/src/agents/team/mailbox.ts

code · 2026-10-18 · 核实路径级 Mutex、proper-lockfile、atomicWriteJSON、已读标记与历史清理。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F06] Qwen 子代理工具权限

https://github.com/QwenLM/qwen-code/blob/46bf492d05f4520bfd918c62e104d45f433da493/packages/core/src/agents/runtime/subagent-plan-tool-policy.ts

code · 2026-10-18 · 核实普通子代理与队员不同的控制面工具排除集，含桥接工具约束。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F07] Qwen 团队任务存储

https://github.com/QwenLM/qwen-code/blob/46bf492d05f4520bfd918c62e104d45f433da493/packages/core/src/agents/team/tasks.ts

code · 2026-10-18 · 核实 updateTask 在锁内检验 callerName、expectedOwner、expectedStatus；归属保护限于协作关键字段，metadata/activeForm 可跨队员注释。源码固定于 URL 所含 commit，读取于 2026-10-18。

[F08] Qwen Code 子代理文档

https://qwenlm.github.io/qwen-code-docs/en/users/features/sub-agents/

official · 2026-10-18 · 2026-10-18 当前文档；核实配置位置、工具 allowlist/blocklist、maxTurns、worktree、普通与 fork 子代理；注意权限继承覆盖规则。

[F09] Trae Agent 仓库

https://github.com/bytedance/trae-agent

official · 2026-10-18 · 当前 commit e839e559ac61bdd0e057c375dd1dee391fee797d，最后提交 2026-02-05；开源 CLI，不等同于闭源 TRAE IDE。

[F10] Trae Agent 系统提示

https://github.com/bytedance/trae-agent/blob/e839e559ac61bdd0e057c375dd1dee391fee797d/trae_agent/prompt/agent_prompt.py

code · 2026-10-18 · 核实七步修复工作流、绝对路径规则、sequential_thinking 建议与 task_done 收束。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F11] Trae Agent 基础循环

https://github.com/bytedance/trae-agent/blob/e839e559ac61bdd0e057c375dd1dee391fee797d/trae_agent/agent/base_agent.py

code · 2026-10-18 · 核实 max_steps、工具执行器、可选 Docker、异常终止、工具关闭与失败反馈。 源码固定于 URL 所含 commit，读取于 2026-10-18。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[F12] Trae Agent 完成判定

https://github.com/bytedance/trae-agent/blob/e839e559ac61bdd0e057c375dd1dee391fee797d/trae_agent/agent/trae_agent.py

code · 2026-10-18 · 核实 task_done；must_patch=true 以路径启发式过滤测试相关 diff 后要求非空，不能证明余项是业务代码，亦非测试通过门禁。源码固定于 URL 所含 commit，读取于 2026-10-18。

[F13] Trae Agent 轨迹记录

https://github.com/bytedance/trae-agent/blob/e839e559ac61bdd0e057c375dd1dee391fee797d/trae_agent/utils/trajectory_recorder.py

code · 2026-10-18 · 核实 llm_interactions、agent_steps、完整 input_messages、每次 append 后整体 JSON 保存。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F14] Trae Agent 原始论文

https://arxiv.org/html/2507.23370v1

paper · 2026-10-18 · 2025 原始论文 v1；核实生成 / 剪枝 / 选择、回归过滤、选择器动态验证与投票；数字不外推到今天产品。

[F15] Open SWE 当前仓库

https://github.com/langchain-ai/open-swe

official · 2026-10-18 · commit aa00dc83871b11e134d30565e7901d4b8ae3ed1f；当前 README 披露开发中、原样部署需要 Managed Deep Agents private beta。

[F16] Open SWE MDA 入口

https://github.com/langchain-ai/open-swe/blob/aa00dc83871b11e134d30565e7901d4b8ae3ed1f/mda/agent.py

code · 2026-10-18 · 核实 define_deep_agent、run token、catalog、代码调用工具、子代理工具白名单、run_limit 与报告中间件。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F17] Open SWE MDA 运行说明

https://github.com/langchain-ai/open-swe/blob/aa00dc83871b11e134d30565e7901d4b8ae3ed1f/mda/README.md

official · 2026-10-19 · 固定 aa00dc8；重取 mda/README.md blob f7f273f0e5f63c3c0d861931b039826a1522c416，核实模型循环 /checkpoint属MDA、后端线程索引及hooks分工；fresh mda dev 重建 .mda/build 并从空 checkpoints 启动。不能外推生产持久化策略。

[F18] Open SWE 主提示模板

https://github.com/langchain-ai/open-swe/blob/aa00dc83871b11e134d30565e7901d4b8ae3ed1f/openswe/resources/prompts/system/main.md.jinja

code · 2026-10-18 · 核实工作环境、来源、repo、协作、执行、提交、用户工作区等片段按 Jinja 条件组装。 源码固定于 URL 所含 commit，读取于 2026-10-18。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[F19] Open SWE 任务执行提示

https://github.com/langchain-ai/open-swe/blob/aa00dc83871b11e134d30565e7901d4b8ae3ed1f/openswe/resources/prompts/system/task-execution.md

code · 2026-10-18 · 核实信息请求/代码变更/自动审查/人工审查分流，PR 线程复用与避免竞争实现线程的说明。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F20] Open SWE PR 创建门禁

https://github.com/langchain-ai/open-swe/blob/aa00dc83871b11e134d30565e7901d4b8ae3ed1f/openswe/middleware/pr_creation_guard.py

code · 2026-10-18 · 核实工具调用包裹层检测直接 gh/API 创建 PR 回退路径；不是完整操作系统安全边界。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F21] Open SWE 对话卸载

https://github.com/langchain-ai/open-swe/blob/aa00dc83871b11e134d30565e7901d4b8ae3ed1f/openswe/middleware/conversation_offloading.py

code · 2026-10-18 · 核实 SummarizationMiddleware 包装；隐藏摘要模型流、started/failed/completed 事件、cutoff/file_path。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F22] Open SWE 任务协作

https://github.com/langchain-ai/open-swe/blob/aa00dc83871b11e134d30565e7901d4b8ae3ed1f/openswe/middleware/task_coordination.py

code · 2026-10-18 · 核实 TaskCoordinationMiddleware；协调者/worker、取消检查、TaskMessage 注入、动态上下文 hash 去重。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F23] Open SWE 沙盒失联语义

https://github.com/langchain-ai/open-swe/blob/aa00dc83871b11e134d30565e7901d4b8ae3ed1f/openswe/middleware/sandbox_circuit_breaker.py

code · 2026-10-18 · 核实 coding sandbox 不可达时不静默替换的告警，read-only reviewer 替换例外。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F24] Open SWE 跟进消息注入

https://github.com/langchain-ai/open-swe/blob/aa00dc83871b11e134d30565e7901d4b8ae3ed1f/openswe/middleware/check_message_queue.py

code · 2026-10-18 · 核实 before_model 检查 thread 消息队列、快照消费、FIFO、分消息 envelope。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F25] AutoGen 归档状态

https://github.com/microsoft/autogen

official · 2026-10-18 · commit 004749e6b7102c8fd317b9b1499ce5807223c278，2026-10-16 README 明确 archived、不再提供修复和支持、指向 Microsoft Agent Framework。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[F26] AutoGen SelectorGroupChat 官方指南

https://microsoft.github.io/autogen/stable/user-guide/agentchat-user-guide/selector-group-chat.html

official · 2026-10-18 · 核实单发言者动态选择、广播上下文、描述用于路由、selector_func/candidate_func、状态跨任务保留。

[F27] AutoGen 选择器源码

https://github.com/microsoft/autogen/blob/004749e6b7102c8fd317b9b1499ce5807223c278/python/packages/autogen-agentchat/src/autogen_agentchat/teams/_group_chat/_selector_group_chat.py

code · 2026-10-18 · 核实 select_speaker 候选校验、模型命名解析、有限重试与回退到上一发言者或第一候选。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F28] AutoGen 终止指南

https://microsoft.github.io/autogen/stable/user-guide/agentchat-user-guide/tutorial/termination.html

official · 2026-10-18 · 核实状态化终止条件、组合表达式与 reset 使用语义。

[F29] AutoGen 终止接口

https://github.com/microsoft/autogen/blob/004749e6b7102c8fd317b9b1499ce5807223c278/python/packages/autogen-agentchat/src/autogen_agentchat/base/_termination.py

code · 2026-10-18 · 核实 StopMessage、And/OrTerminationCondition 与 reset；终止与业务成功是不同概念。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F30] AutoGen AssistantAgent 源码

https://github.com/microsoft/autogen/blob/004749e6b7102c8fd317b9b1499ce5807223c278/python/packages/autogen-agentchat/src/autogen_agentchat/agents/_assistant_agent.py

code · 2026-10-18 · 核实增量消息输入、非线程/协程安全警告、工具迭代上限、reflect_on_tool_use、模型上下文截断选择。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F31] AutoGen 消息通信指南

https://microsoft.github.io/autogen/stable/user-guide/core-user-guide/framework/message-and-communication.html

official · 2026-10-18 · 核实直接消息与发布订阅；publish 等待调度但不返回接收方业务结果，异常记录不反向传播。

[F32] MetaGPT 原始论文

https://arxiv.org/html/2308.00352v7

paper · 2026-10-18 · 2024 修订 v7；核实 SOP 文档交接、角色职责与消息观察；研究评测非生产保证。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[F33] MetaGPT Role

https://github.com/FoundationAgents/MetaGPT/blob/11cdf466d042aece04fc6cfd13b28e1a70341b1f/metagpt/roles/role.py

code · 2026-10-18 · 核实 _observe 订阅过滤与私有 msg_buffer、_act 输出 Message、REACT/BY_ORDER/PLAN_AND_ACT。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F34] MetaGPT Environment

https://github.com/FoundationAgents/MetaGPT/blob/11cdf466d042aece04fc6cfd13b28e1a70341b1f/metagpt/environment/base_env.py

code · 2026-10-18 · 核实 publish_message 按地址投递与 history 记录，run 并发调度非 idle roles。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F35] MetaGPT Team

https://github.com/FoundationAgents/MetaGPT/blob/11cdf466d042aece04fc6cfd13b28e1a70341b1f/metagpt/team.py

code · 2026-10-18 · 核实 invest 成本预算、n_round 限制、idle 停止与 archive/serialize。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F36] MetaGPT 写代码动作

https://github.com/FoundationAgents/MetaGPT/blob/11cdf466d042aece04fc6cfd13b28e1a70341b1f/metagpt/actions/write_code.py

code · 2026-10-18 · 核实 Design/Task/Legacy Code/Debug Logs 输入段、单文件输出、CodeParser 与测试反馈读取。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F37] ChatDev 版本边界

https://github.com/OpenBMB/ChatDev/blob/a2ea4a263cebe9a7abd01ee3b5b0bf95698ed328/README.md

official · 2026-10-18 · 核实当前主分支为 2.0 DevAll；2026-01-07 发布；1.0 virtual software company 移至 chatdev1.0。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F38] ChatDev 1.0 原始论文

https://arxiv.org/abs/2307.07924

paper · 2026-10-18 · 核实 chat chain、communicative dehallucination、设计/编码/测试的语言协作范式。

[F39] ChatDev 2.0 DevAll 论文

https://arxiv.org/html/2609.00714v1

paper · 2026-10-18 · 2026-09 v1；核实 semantic edges 把数据传播和激活分开，CADET 环收缩与分层调度；复现而非全面超越各专用系统。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[F40] DevAll 环检测

https://github.com/OpenBMB/ChatDev/blob/a2ea4a263cebe9a7abd01ee3b5b0bf95698ed328/workflow/cycle_manager.py

code · 2026-10-18 · 核实 Tarjan SCC、CycleInfo、配置上限与默认最大 100 次迭代。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F41] DevAll 环执行

https://github.com/OpenBMB/ChatDev/blob/a2ea4a263cebe9a7abd01ee3b5b0bf95698ed328/workflow/executor/cycle_executor.py

code · 2026-10-18 · 核实作用域内嵌套环和拓扑层；退出要求外部目标被出口边实际触发，另有入口重触发与上限终止。源码固定于 URL 所含 commit，读取于 2026-10-18。

[F42] DevAll 图上下文

https://github.com/OpenBMB/ChatDev/blob/a2ea4a263cebe9a7abd01ee3b5b0bf95698ed328/workflow/graph_context.py

code · 2026-10-18 · 核实节点输出与子图状态、node_outputs.yaml / workflow_summary.yaml 落盘；不据此声称分布式 exactly-once。源码固定于 URL 所含 commit，读取于 2026-10-18。

[F43] DevAll ReAct 模板

https://github.com/OpenBMB/ChatDev/blob/a2ea4a263cebe9a7abd01ee3b5b0bf95698ed328/yaml_instance/react.yaml

code · 2026-10-18 · 核实 Task Normalizer→ReAct subgraph→QA，trigger:false 只传信息边与 TODO 条件回边。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F44] DevAll 记忆模块文档

https://github.com/OpenBMB/ChatDev/blob/a2ea4a263cebe9a7abd01ee3b5b0bf95698ed328/docs/user_guide/en/modules/memory.md

official · 2026-10-18 · 核实 memory store 与节点 attachment 分离、按阶段检索、read/write、simple/file/blackboard/mem0 区别。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F45] Qwen Agent Board 文件锁

https://github.com/QwenLM/qwen-code/blob/46bf492d05f4520bfd918c62e104d45f433da493/packages/core/src/agents/team/board-lock.ts

code · 2026-10-18 · 核实路径级双层锁、私有目录、name validation 与原子 ID 分配；与团队任务表同类但独立实现。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F46] Open SWE 提示资源加载

https://github.com/langchain-ai/open-swe/blob/aa00dc83871b11e134d30565e7901d4b8ae3ed1f/openswe/prompts.py

code · 2026-10-18 · 核实静态 Markdown / Jinja 模板分离、模板路径校验、工具 description 及 schema 缓存。 源码固定于 URL 所含 commit，读取于 2026-10-18。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[F47] Open SWE 共同提示约束

https://github.com/langchain-ai/open-swe/blob/aa00dc83871b11e134d30565e7901d4b8ae3ed1f/openswe/resources/prompts/system/shared-base.md

code · 2026-10-18 · 核实完整 CI 状态验证、未知状态保持未知、后台执行共享 worktree 约束、结束报告语义。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F48] Open SWE 旧架构页

https://github.com/langchain-ai/open-swe/blob/aa00dc83871b11e134d30565e7901d4b8ae3ed1f/openwiki/architecture/agent-graph.md

official · 2026-10-18 · 页面自身 verified 日期 2026-09-08，包含旧 agent/ 路径；只用来展示文档时效风险，不作为当前 MDA 结构证据。源码固定于 URL 所含 commit，读取于 2026-10-18。

[F49] Trae Selector 使用说明

https://github.com/bytedance/trae-agent/blob/e839e559ac61bdd0e057c375dd1dee391fee797d/evaluation/patch_selection/README.md

official · 2026-10-18 · 核实独立 evaluation/patch_selection 入口、候选 JSONL、可选回归结果、分组及可选 majority_voting。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F50] Trae Selector 实现

https://github.com/bytedance/trae-agent/blob/e839e559ac61bdd0e057c375dd1dee391fee797d/evaluation/patch_selection/trae_selector/selector_agent.py

code · 2026-10-18 · 核实 build_system_prompt 强制至少一个正确候选且不许新增补丁；结果解析、默认第一候选与 max_turn。 源码固定于 URL 所含 commit，读取于 2026-10-18。

[F51] Open SWE 安装指南：共享信任与部署边界

https://github.com/langchain-ai/open-swe/blob/aa00dc83871b11e134d30565e7901d4b8ae3ed1f/docs/INSTALLATION.md

official · 2026-10-19 · 2026-10-19 对照 main 安装文档后，通过只读文件接口成功读取正文同一固定提交（blob e0a56b5d1926dabe5d051dc780289c97ae3b0930）；普通成员编码沙盒通常使用安装级 GitHub App 凭据，工作空间/私有线程不收窄仓库访问；reviewer/public-event 部分流程例外。

[F52] AutoGen 当前停维声明与迁移指引

https://github.com/microsoft/autogen

official · 2026-10-19 · 2026-10-19 打开当前 README；项目自述 archived 且微软不再维护，推荐 Agent Framework。本轮未观察到平台 Public archive 横幅，不将 README 声明等同于平台归档标志。

[G01] Zed: External Agents

https://zed.dev/docs/ai/external-agents

official · 2026-10-18 · ACP 外部代理的配置、认证与工具责任边界。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[G02] Zed: Agent Panel

https://zed.dev/docs/ai/agent-panel

official · 2026-10-18 · 独立线程、消息排队/steer、检查点的集成差异。

[G03] Zed: Parallel Agents

https://zed.dev/docs/ai/parallel-agents

official · 2026-10-18 · 多线程、worktree 与历史恢复的产品契约。

[G04] Zed: Instructions

https://zed.dev/docs/ai/instructions

official · 2026-10-18 · 指令文件按首个匹配加载；与外部代理分开。

[G05] Replit: Decision-Time Guidance

https://frontier.replit.com/decision-time-guidance/

official · 2026-10-18 · 2026-01-20；按轨迹信号选择瞬时提示，闭源工程说明。

[G06] Replit: Free the models

https://replit.com/blog/free-the-models

official · 2026-10-18 · 2026-09-29；可复用子代理、能力档位和动态 effort；不采用厂商成绩作独立结论。

[G07] Replit: Task board

https://docs.replit.com/features/agent/task-board

official · 2026-10-19 · 2026-10-19 复读当前官方 task-board：Done 还含归档/取消，自动处理应用冲突；auto-apply 为单次快捷动作，Ready 后可跳过等待人工审核，不是持续开关。

[G08] Antigravity: Artifacts

https://www.antigravity.google/docs/artifacts/

official · 2026-10-18 · 计划、diff、录屏与异步里程碑审阅。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[G09] Antigravity: Custom subagents

https://www.antigravity.google/docs/subagents/

official · 2026-10-18 · 子代理的上下文、权限继承和审批上浮；官方契约。

[G10] Antigravity: Hooks

https://www.antigravity.google/docs/hooks/

official · 2026-10-18 · 模型调用/工具调用钩子及 deny/ask 等结果。

[G11] Antigravity: Terminal sandbox

https://www.antigravity.google/docs/sandbox/

official · 2026-10-18 · Linux namespaces、macOS Seatbelt；Windows 行为不同。

[G12] Warp Factories（/oz 官方入口）

https://www.warp.dev/oz

official · 2026-10-18 · 跨 harness 云端编排的产品能力与等待名单边界。；2026-10-18 重开时页面使用 Warp Factories 品牌，持久记忆与自改进仍带 waitlist。

[G13] Warp repository README

https://github.com/warpdotdev/warp/blob/1e6ef379bc9f51257c4243e8c4396a046cede8af/README.md

official · 2026-10-18 · 已读当前 README；客户端开源，UI MIT、其余 AGPL v3。非完整云端审计。；独立复核：README 固定于 1e6ef379bc9f51257c4243e8c4396a046cede8af（提交日期 2026-10-18），2026-10-18 重新读取。

[G14] Warp: Cloud agents overview

https://docs.warp.dev/platform/

official · 2026-10-18 · 云端 Agent 平台的官方入口与环境配置。

[G15] Refact legacy repository README

https://github.com/smallcloudai/refact/blob/27f86def9423b1153f1ff857a51329a88bbbe292/README.md

official · 2026-10-18 · 旧仓库指向 JegernOUTT/refact；Refact Cloud 已退休。；独立复核：README 固定于 27f86def9423b1153f1ff857a51329a88bbbe292（提交日期 2026-05-30），2026-10-18 重新读取。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[G16] Refact current repository README

https://github.com/JegernOUTT/refact/blob/0d096c958c6d1af9b702e48607adab03cc689c86/README.md

official · 2026-10-18 · 常驻 daemon、项目 worker 与 worktree fleet；daemon/TUI 为 early preview。 ；独立复核：README 固定于 0d096c958c6d1af9b702e48607adab03cc689c86（提交日期 2026-09-08），2026-10-18 重新读取。

[G17] bolt.diy documentation index

https://github.com/stackblitz-labs/bolt.diy/blob/main/docs/docs/index.md

official · 2026-10-18 · WebContainer、终端与实时预览；文档级能力，未深审执行器。

[G18] Kimi CLI archived README

https://github.com/MoonshotAI/kimi-cli/blob/9ab1286b8fe4e6bcd116949a27ce5e0ac3389c82/README.md

official · 2026-10-18 · Python 旧版停维并迁移至 Kimi Code CLI；凭证不迁移。 ；独立复核：README 固定于 9ab1286b8fe4e6bcd116949a27ce5e0ac3389c82（提交日期 2026-09-22），2026-10-18 重新读取。

[G19] Kimi Code CLI README

https://github.com/MoonshotAI/kimi-code/blob/4a82a07f08d0b135c4349c20aae53a2d51e5c3fa/README.md

official · 2026-10-18 · 新 CLI、coder/explore/plan、hooks 与 ACP；MIT。 ；独立复核：README 固定于 4a82a07f08d0b135c4349c20aae53a2d51e5c3fa（提交日期 2026-10-17），2026-10-18 重新读取。

[G20] Augment: Context Engine MCP

https://docs.augmentcode.com/context-services/mcp/overview

official · 2026-10-18 · 本地工作树与远端默认分支的索引范围不同。

[G21] Augment: Cloud Agents

https://www.augmentcode.com/product/cloud-agents

official · 2026-10-18 · Cosmos 专家、环境、触发器与审核配置；仅官方产品契约。

[G22] Augment Context Engine plugin

https://github.com/augmentcode/context-engine-plugin

official · 2026-10-18 · 公开插件入口，不能由此推定云端检索实现开源。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[G23] Factory: Compressing Context

https://factory.com/news/compressing-context

official · 2026-10-18 · 2025-07-21；锚定增量摘要、双阈值与任务级 token 目标。

[G24] Factory: Custom droids

https://docs.factory.com/harness/subagents

official · 2026-10-18 · Task/TaskOutput/TaskStop/resume、工具策略与非交互子代理。

[G25] Factory: Autonomy Level

https://docs.factory.com/autonomy-and-safety/auto-run

official · 2026-10-18 · 自治级别与工具存在性、命令规则、沙箱分层。

[G26] Amp: Modes and Models

https://ampcode.com/docs/models-and-subagents

official · 2026-10-18 · 主代理/专长子代理/系统模型；专长子代理不互聊且只回总结。

[G27] Amp: Introduction

https://ampcode.com/docs

official · 2026-10-18 · 每线程云端 orb 与跨客户端会话概念。

[G28] Junie: Guidelines and memory

https://junie.jetbrains.com/docs/guidelines-and-memory.html

official · 2026-10-18 · 2026-10-15 文档；AGENTS.md 优先级、全局/项目去重。

[G29] Junie: Agent skills

https://junie.jetbrains.com/docs/agent-skills.html

official · 2026-10-18 · 按需技能与持续 guidelines 区分；description 触发。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[G30] ACP: Introduction

https://agentclientprotocol.com/get-started/introduction

official · 2026-10-18 · 编辑器/Agent 协议边界；远程完整支持仍在推进。

[G31] ACP v1: Session Setup

https://agentclientprotocol.com/protocol/v1/session-setup

official · 2026-10-18 · session/load 回放与可选 resume/close 能力。

[G32] ACP v1: Cancellation

https://agentclientprotocol.com/protocol/v1/cancellation

official · 2026-10-18 · 通用请求取消及子活动取消的可选性。

[G33] ACP v2: Overview

https://agentclientprotocol.com/protocol/v2/overview

official · 2026-10-18 · 检索时导航仍标 v2 Draft；不得视作全部客户端已支持。

[G34] Amp Agent to Agent

https://ampcode.com/docs/orbs/agent-to-agent

official · 2026-10-18 · 独立线程拥有独立对话、上下文、工作副本与 Orb；支持回报和显式文件传输，消息本身不传输文件或提交。

[G35] ACP v1 Prompt Turn / cancellation and compaction

https://agentclientprotocol.com/protocol/v1/prompt-turn

official · 2026-10-18 · session/cancel 要求 pending permission 返回 cancelled；prompt 的 cancelled 响应在中止相关操作后给出。compaction 更新是可选能力，不命令客户端删除历史。

[G36] Kimi 旧仓库：归档组件与包名陷阱

https://github.com/MoonshotAI/kimi-cli

official · 2026-10-19 · 2026-10-19 读取当前 README 包清单；PyPI kimi-code 为旧 kimi-cli alias，不是新 CLI；kosong、pykaos、kimi-sdk 同步归档。新 CLI 使用 npm @moonshot-ai/kimi-code / brew kimi-code；只核实发布身份，未执行安装。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[G37] Replit Task lifecycle：应用中间态

https://docs.replit.com/features/agent/task-lifecycle

official · 2026-10-19 · 2026-10-19 当前官方说明区分 Ready、Applying 与 Done，称自动处理多任务冲突；未公开具体合并算法、原子性或集成后测试绑定。独立首读成功；主审直开暂失败后，经 task-board 官方导航重取成功，交叉核实。

[H01] MCP 2026-07-28 Tools：提示字段与身份边界

https://github.com/modelcontextprotocol/modelcontextprotocol/blob/1155ca460e5097019f32769b38dcba194432e2fb/docs/specification/2026-07-28/server/tools.mdx

official · 2026-10-19 · 固定当前提交，blob 449020f54a6582122607b4869129bec5f1035f37：不可信服务annotations不可信；工具名仅server内唯一，serverInfo.name不保证唯一；目录变化受规范条件约束，不外推全部客户端协议版本。

[H02] MCP 本地服务安全：启动期与目录投毒

https://github.com/modelcontextprotocol/modelcontextprotocol/blob/1155ca460e5097019f32769b38dcba194432e2fb/docs/docs/2026-07-28/tutorials/security/local-server-security.mdx

official · 2026-10-19 · 固定当前提交，blob 78e0d09153f5a87f45e3bac5844df1552e7bbe54：stdio不提供沙箱，启动已执行代码；目录投毒可能影响别的工具、批准后定义可变；宿主可部署强隔离。官方威胁模型，非产品攻击复现。

[H03] Git 2.56.0 worktree：共享控制面

https://git-scm.com/docs/git-worktree/2.56.0

official · 2026-10-19 · 官方版本手册（2026-09-28更新）：linked worktree独立HEAD/index但共享多数refs与默认config；worktreeConfig添加专属层而非独立仓库。未运行反例。

[H04] Git status：未跟踪、忽略和子模块显示

https://git-scm.com/docs/git-status

official · 2026-10-19 · 当前官方手册：ignored通常不显示、untracked可隐藏、ignore-submodules影响脏状态显示。只登记当前成功读取URL，不冒充固定版本复核；清单建议只覆盖实际受测输入。

[H05] Git config：core.hooksPath

https://git-scm.com/docs/git-config#Documentation/git-config.txt-corehooksPath

official · 2026-10-19 · 当前官方手册：core.hooksPath改变Git Hook目录或禁用。若执行者可修改该配置，不能把本地Git Hook当独立强制验收；不涉及各产品自定义Hook。

[H06] Anthropic Research：并发适用性与评估反例

https://www.anthropic.com/engineering/multi-agent-research-system

official · 2026-10-19 · 2025-06-13作者工程经验：宽度优先研究适合并行、编码常较少真正并行部分；该评估中单rubric judge优于其多judge尝试。历史Research系统非当前Claude Code内部保证，不外推收益或普适优劣。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[H07] AWS SQS SendMessage：去重不校验全部意图

https://docs.aws.amazon.com/AWSSimpleQueueService/latest/APIReference/API_SendMessage.html

official · 2026-10-19 · 当前FIFO合同：同dedup ID五分钟内可接收成功但不投递；内容去重哈希只含body不含attributes。用于域外反例，不声称Peri或CodingAgent采用SQS；未调用API。

[H08] Stripe API v1：幂等键与参数绑定

https://docs.stripe.com/api/idempotent_requests

official · 2026-10-19 · 当前API v1说明同key参数不同报错，首次执行响应包括500可复用；v2不可一概外推。用于业务幂等对照，不涉及支付操作或产品实测。

[H09] LangGraph Interrupts：节点重跑与顺序匹配

https://docs.langchain.com/oss/python/langgraph/interrupts

official · 2026-10-19 · 当前Python公开合同：resume重跑所在node；节点内多个interrupt按序号匹配，前置副作用应幂等或拆开。保存graph state不等于逐语句执行栈；不外推Open SWE MDA或所有框架部署。

[H10] LangGraph Checkpointers：重放与持久化模式

https://docs.langchain.com/oss/python/langgraph/checkpointers

official · 2026-10-19 · 当前Python文档：exit仅图退出时落盘，async与下步并行，sync等写checkpoint再继续；重放可重执行checkpoint后的节点。durable-execution旧URL已转persistence，沿官方链接找到此页；未运行恢复实验。

[I01] smolagents：CodeAgent 与托管代理实现

https://github.com/huggingface/smolagents/blob/96f33faaf028479119ec8d34507b47694cf14e34/src/smolagents/agents.py

code · 2026-10-19 · 当前 main 提交 96f33fa（2026-10-06）；实读提示装配、__call__、create_python_executor 与 interrupt/_run_stream。blob a8f6e52de0361a6c23c0aa5d5907180969f259b7。

[I02] smolagents：代码代理公开提示模板

https://github.com/huggingface/smolagents/blob/96f33faaf028479119ec8d34507b47694cf14e34/src/smolagents/prompts/code_agent.yaml

code · 2026-10-19 · 实读 system_prompt、planning 与 managed_agent 的 task/report 模板；blob c81073a84a67d7c6f5ed03ded9f525c7944c9c1f；仅概述，未转载全文。

[I03] smolagents：安全代码执行与沙盒边界

https://huggingface.co/docs/smolagents/en/tutorials/secure_code_execution

official · 2026-10-19 · 当前动态文档：AST 本地解释器、导入限制、资源风险，以及片段执行和完整代理入沙盒两种拓扑。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[I04] smolagents：多步代理的消息与记忆循环

https://huggingface.co/docs/smolagents/en/conceptual_guides/react

official · 2026-10-19 · 当前文档说明 SystemPromptStep/TaskStep、write_memory_to_messages、ActionStep 与可选 PlanningStep。

[I05] CAMEL：Workforce 调度、停止与内存快照

https://github.com/camel-ai/camel/blob/e5e55256e9446f744bf7309c478b084587dbf985/camel/societies/workforce/workforce.py

code · 2026-10-19 · 当前 master 提交 e5e5525（2026-10-12）；实读任务分配校验、停止方法、WorkforceSnapshot/save_snapshot/restore_from_snapshot。blob e1fff726c2cbf61f07218f6fe622474c93f29b28。

[I06] CAMEL：进程内 TaskChannel 的认领与返回

https://github.com/camel-ai/camel/blob/e5e55256e9446f744bf7309c478b084587dbf985/camel/societies/workforce/task_channel.py

code · 2026-10-19 · 实读 PacketStatus、Condition、任务字典/索引、认领、返回与依赖归档；blob 39dca42af0edb29874bacf4271d64d97d30a31e2。

[I07] CAMEL：任务分配与工作者提示协议

https://github.com/camel-ai/camel/blob/e5e55256e9446f744bf7309c478b084587dbf985/camel/societies/workforce/prompts.py

code · 2026-10-19 · 实读 ASSIGN_TASK_PROMPT 与 PROCESS_TASK_PROMPT，含任务身份、依赖、结果结构及附加信息提示边界；blob 2411435886e3889f56ab121350396457e14021ac。

[I08] CAMEL：Workforce 当前模块说明

https://docs.camel-ai.org/key_modules/workforce

official · 2026-10-19 · 当前模块文档区分任务拆解、协调分配、两种 Worker 与 share_memory 参数；未据此推导编码交付保证。

[I10] Tabnine Plugin for OpenCode: Overview

https://docs.tabnine.com/main/getting-started/tabnine-plugin-for-opencode

official · 2026-10-19 · 当前动态文档正文：托管 OpenCode 副本、插件、Tabnine 供应商锁定、自动注册上下文/Coaching MCP、组织治理与双版本标识；非引擎源码审计。

[I11] Tabnine: In-IDE Agent Settings

https://docs.tabnine.com/main/getting-started/tabnine-agent/agent-settings

official · 2026-10-19 · 当前 IDE 合同：原生工具逐项 Auto-approve / Ask first / Disable；远端目录/符号/语义检索工具与 MCP 服务器设置。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[I12] Tabnine: Understanding MCP Servers

https://docs.tabnine.com/main/getting-started/tabnine-agent/mcp-intro-and-setup

official · 2026-10-19 · 当前 IDE MCP 合同：需打开项目目录；项目/用户配置；command、url、SSE 的传输识别。未由此推定新 OpenCode 入口的配置加载相同。

[I13] Tabnine CLI: Commands and maintenance notice

https://docs.tabnine.com/main/getting-started/tabnine-cli/features/commands

official · 2026-10-19 · 旧 CLI 自6.6停止主动开发，关键更新至2026-12-31，之后弃用并迁移至 OpenCode 插件；旧命令区分会话恢复、项目文件恢复、上下文摘要。

[I14] Qoder CLI: Subagent

https://docs.qoder.com/cli/subagent

official · 2026-10-19 · 当前 CLI 文档：独立上下文/工具注册表/压缩；description 与主体提示词分工；Markdown worktree 可选；Agent(name) 控制二级委派。未绑定具体二进制版本。

[I15] Qoder CLI: How Task Execution Works

https://docs.qoder.com/cli/how-it-works

official · 2026-10-19 · 当前 CLI 合同：上下文三层来源、工具观测回流；非交互 ask 转 deny，SDK/ACP 向宿主审批。

[I16] Qoder CLI: Permissions

https://docs.qoder.com/cli/permissions

official · 2026-10-19 · 当前权限文档：Agent 规则 deny>ask>allow，无匹配规则时启动无需确认；前置 Hook 直接allow/deny可短路流水线。属于官方承诺而非实际绕过测试。

[I17] Tencent Cloud: CodeBuddy Code Subagent

https://intl.cloud.tencent.com/document/product/1316/84444

official · 2026-10-19 · 官方镜像正文最后更新2026-09-30 18:41:45；明确独立上下文、工具继承、作用域 MCP 生命周期与 memory 自动加入写工具；英文为官方AI翻译。

[I18] Tencent Cloud: CodeBuddy Code Agent Teams

https://intl.cloud.tencent.com/zh/document/product/1316/84445

official · 2026-10-19 · 官方中文正文最后更新2026-09-30 18:41:45：进程内成员、项目上下文但不继承领导对话历史、信箱、任务、关闭握手、完成后消息唤醒，以及resume不恢复成员。正文内权限简述不如Settings精确。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[I19] Tencent Cloud: CodeBuddy Code Settings

https://www.tencentcloud.com/document/product/1316/84425

official · 2026-10-19 · 官方镜像正文最后更新2026-09-30 18:15:48：子代理权限覆盖与auto/dontAsk父权限上限；autoMode只读用户、本地项目和CLI而不读共享项目配置。

[I20] iFlow CLI 团队停维与关闭公告

https://vibex.iflow.cn/t/topic/4819

official · 2026-10-19 · 2026-03-20 团队公告：当日起停维，2026-04-17 关闭 CLI 产品及 iFlow API/模型库；FAQ 说明已安装 CLI 可接自定义 API 继续使用，心流 AI 助手为另一产品。由官方 README 直接链接，正文已读取。

[I21] iFlow CLI 子代理配置与权限继承文档

https://github.com/iflow-ai/iflow-cli/blob/4642808afbc6580ac117d930f6c64ac0d84955c7/docs_cn/examples/subagent.md

official · 2026-10-19 · 实际观察 main commit 4642808afbc6580ac117d930f6c64ac0d84955c7 并读取文件：Task 调用、项目配置字段、默认工具/MCP 继承、顺序协作示例；这是公开文档，不是执行器源码。

[I22] iFlow CLI 停维基线 README 与公开文件树

https://github.com/iflow-ai/iflow-cli/blob/4642808afbc6580ac117d930f6c64ac0d84955c7/README.md

official · 2026-10-19 · README 顶部关闭提示及自定义 OpenAI-compatible API 入口；同提交递归树共 93 项且未截断，主要为文档/资源/安装脚本，未取得 CLI 内核实现。不能据公开仓库认定完整内核开源，未审 SDK 源码。

[I23] Zoo Code 当前社区身份与公开源码入口

https://github.com/Zoo-Code-Org/Zoo-Code/blob/b69989b823422bb3839454a479915d4c2003ba00/README.md

official · 2026-10-19 · 实际观察 main commit b69989b823422bb3839454a479915d4c2003ba00；README 自述原 Roo 贡献者组成社区延续、Apache 2.0，并宣称加强并行协调；本报告不把该宣称直接当作已核实调度保证。官网与该仓库互链。

[I24] Zoo NewTaskTool：审批前挂起动作与父子委派

https://github.com/Zoo-Code-Org/Zoo-Code/blob/b69989b823422bb3839454a479915d4c2003ba00/src/core/tools/NewTaskTool.ts

code · 2026-10-19 · 全文读取：由 toolCallId 清洗 actionId，可用时在审批前写 provider/task pending action；批准后 delegateParentAndOpenChild 并传 pendingActionId。注释称子任务为唯一活跃任务；未核查所有 provider 实现及外部并行路径。

[I25] Zoo new_task 原生工具协议

https://github.com/Zoo-Code-Org/Zoo-Code/blob/b69989b823422bb3839454a479915d4c2003ba00/src/core/prompts/tools/native-tools/new_task.ts

code · 2026-10-19 · 全文读取：提示要求 new_task 单独调用；strict schema，mode/message/todos 为 required，todos 可 null，additionalProperties=false。提示要求不等于执行器强制互斥。

来源与复核入口

访问日期以各记录为准；分支链接会随时间变化，固定提交优先。

[I26] Microsoft Agent Framework 当前 README

https://github.com/microsoft/agent-framework/blob/ebcf4db7f9462ef6c377761e0b3159c4c5a73027/README.md

official · 2026-10-19 · 实际观察 main commit ebcf4db7f9462ef6c377761e0b3159c4c5a73027 并读取全文；Python 与 .NET 分立目录，Go 指向独立仓库；AF Labs 仍标实验包，不能将核心 GA 扩大到全体扩展。

[I27] Microsoft Agent Framework 1.0 发布公告

https://devblogs.microsoft.com/agent-framework/microsoft-agent-framework-version-1-0/

official · 2026-10-19 · 2026-04-03 官博宣布 .NET/Python 1.0，核心代理与工作流稳定；同期明确列出 preview 能力。只证明公告时的成熟度，不声称这些扩展至今都未转正。

[I28] MAF AgentSession：语言对象与服务会话授权

https://learn.microsoft.com/en-us/agent-framework/concepts/agents/conversations/session

official · 2026-10-19 · 当前官方正文可读（页面有通用授权提示，但正文完整），更新日期 2026-08-25；C# 抽象 AgentSession/StateBag 与 Python session_id/service_session_id/state 不同；序列化与恢复、服务 ID 非用户授权边界、会话具有 agent/service 适配条件。Edit 链接失败，采用可读官方页面。

[I29] MAF 工作流检查点：超步、语言恢复钩子与身份

https://learn.microsoft.com/en-us/agent-framework/workflows/checkpoints

official · 2026-10-19 · 当前官方正文更新2026-09-16；Python 1.13.0另有入口检查点；语言保存/恢复钩子不同。.NET重建须保留拓扑/执行器身份，本地ChatClientAgent保持Id及已设Name；存储/执行器接入须配置，未故障实测。

[I30] MAF 当前概览：核心领域与 Go 公测范围

https://learn.microsoft.com/en-us/agent-framework/overview/

official · 2026-10-19 · 当前官方正文更新2026-08-25：代理、Harness、工作流、集成分层；Go仍public preview且缺少部分能力，不由Python/.NET核心GA外推语言特性对等。