

CrewAgent 产品白皮书

Universal Agent Operating System

1. 产品概述

CrewAgent 是一个面向专业领域的通用 Agent 操作系统（Universal Agent Operating System）。它的核心愿景是让 AI 能够处理复杂的、长流程的专业工作，而不仅仅是简单的对话。

与通用的聊天助手不同，CrewAgent 在架构上彻底解耦了“大脑”（定义与逻辑）与“双手”（执行与工具）：

层级	说明
平台端 (The Platform)	云端可视化环境，专家在这里定义工作流、提示词 (Prompt) 和逻辑，无需编写代码
运行时 (The Runtime)	本地优先 (Local-First) 的桌面应用程序，在用户电脑上安全地执行这些工作流，直接操作本地文件、工具和数据

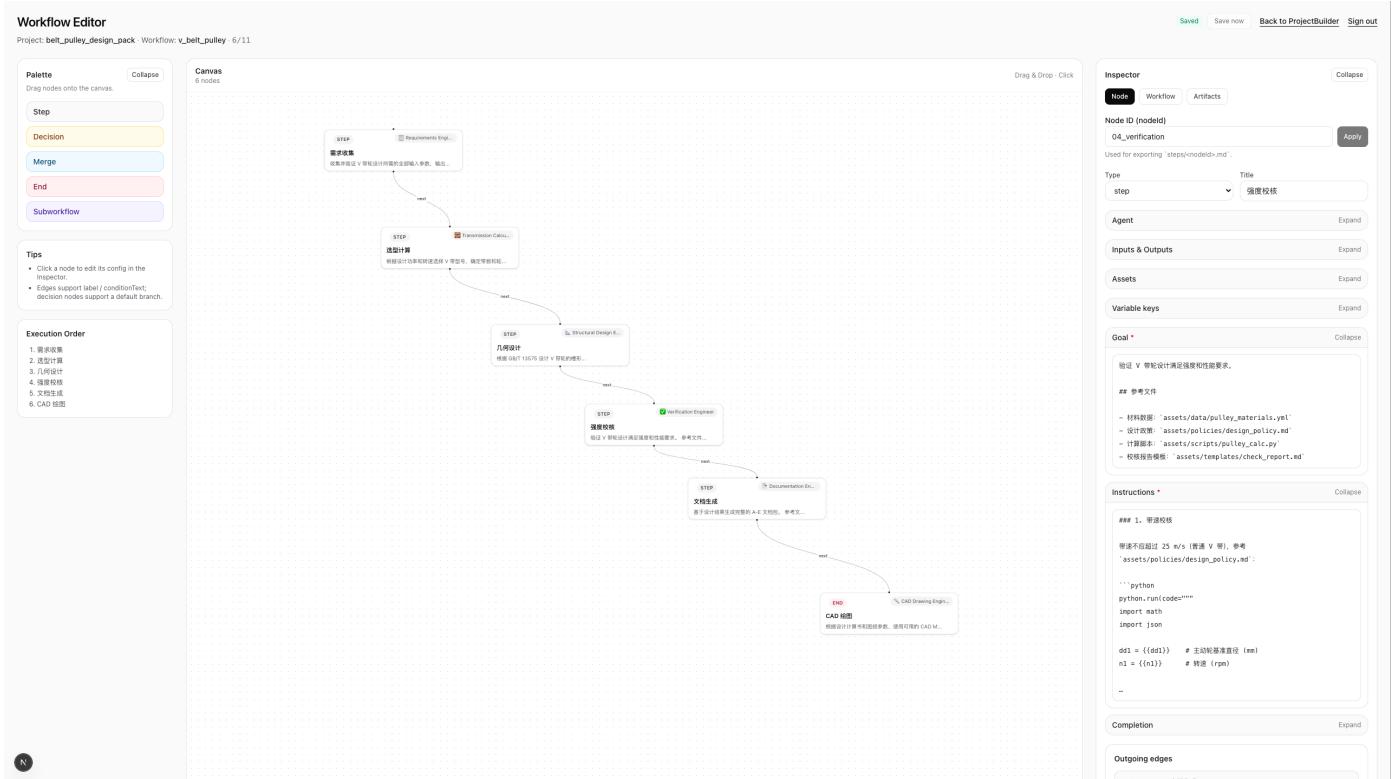
这种架构将用户从“手动搬运数据的操作员”转变为“数字专家的监督者”，实现了“仿生 (Bionic) ”工作流——用云端的智慧驱动本地的工具，完成各种复杂的专业任务。

2. 核心能力

CrewAgent 提供了一套完整的生态系统来创建和运行专业级 Agent：

面向创作者 (Visual Builder)

- 无代码节点图 (No-Code Node Graph): 通过拖拽界面设计复杂的业务逻辑和工作流，而非仅靠文字 Prompt
- 仿生 Agent 定义 (Bionic Agent Definition): 定义专门的角色、原则和工具策略
- 包导出 (Package Export): 一键将整个工作流导出为可移植的标准 `.bmad` 包 (ZIP 格式)
- 模板优先 (Template Library): 提供经过验证的专业模式 (如“参数校验流程”、“自动化报告生成”)，让专家无需从零开始



面向消费者 (Runtime Engine)

- 本地优先执行 (Local-First Execution):** 纯桌面端运行 (macOS/Windows) , 确保数据隐私, 且与本地工具 (IDE、终端、Excel 等) 无缝交互零延迟
- MCP 集成 (MCP Integration):** 原生支持 **Model Context Protocol (MCP)**, 标准化了与本地工具 (文件系统、终端、Python 环境、甚至未来的 CAD/ERP) 的连接
- 沙箱安全 (Sandboxed Security):** 严格限制文件系统的读写权限 (仅限项目目录) , 防止 AI 误操作导致数据丢失或泄露
- 离线/私有化支持 (Offline Trust):** 支持断网运行 (激活后) , 并兼容本地大模型 (如 Ollama) , 完美适配对数据安全极其敏感的场景
- 可验证与可恢复 (Verifiable Output):** "文档即状态 (Document-as-State)" 的架构确保每一步操作都记录在本地 Markdown 文件中, 过程可审计, 崩溃可恢复

The screenshot shows the CrewAgent application's user interface. On the left, there's a sidebar titled 'dl' with sections for 'Files' and 'Works'. Under 'Files', there are several sub-folders and files listed, including 'artifacts' which contains various JSON and MD files. The main workspace is titled 'CONVERSATION' and shows a 'Workflow: V 带轮设计' (V-belt design) task. This task has three input forms:

- 请提供V带轮设计的基本参数**: Submits power P (kW) from 0.1-500 kW and主动轮转速 n1 (rpm) from 100-3000 rpm.
- 请选择工作条件信息以确定工作条件系数 K_a**: Selects drive motor type (electric motor/gearbox), work machine type (crane/transporter), and daily working hours (10-16h).
- 请提供可选参数 (如无特殊要求可使用默认值)**: Submits active run parameters d1 (mm) and d2 (mm).

To the right of the conversation area is a 'WORKFLOW PROGRESS' panel with a table showing tasks like '需求收集' (In Progress), '选型计算' (Pending), etc. At the bottom right are 'Logs' and 'Settings' buttons.

3. 核心优势

CrewAgent 解决了专业级 AI 应用落地的四个核心痛点：

数据主权与绝对隐私

"逻辑在云端，数据在本地"

痛点	CrewAgent 解决方案
企业不敢将核心代码库、财务底稿或机密图纸上传到云端 SaaS AI 平台	Runtime 运行在本地设备上，只下载"如何做任务"的逻辑，所有数据从未离开过您的设备
芯片设计、金融审计等领域数据就是生命	支持在断网（Air-gapped）环境下配合本地大模型运行，实现物理级的隐私安全

真·生产力：执行者而非建议者

"直接交付最终结果"

传统 AI 局限	CrewAgent 能力
ChatBot 只能停留在"给建议"的层面	像一个真正的数字员工，拥有操作文件系统的权限（沙箱内）
无法处理涉及 50 个文件联动的复杂任务	可以打开项目文件夹，批量读取 100 份财报，提取数据生成 Excel，甚至直接修改代码仓库并提交 Git
交付的是"工作的建议"	交付的是"完成的工作"

仿生架构：经验的无损复制

"SOP 的数字化与执行化"

- 可视化的业务流程编排器让业务专家能像画流程图一样，将自己处理逻辑固化
- 定义好的"数字专家包 (.bmad)"可以分发给任何初级员工
- 初级员工一键运行，即可获得资深专家级别的辅助
- 实现了隐性知识的规模化、标准化复制

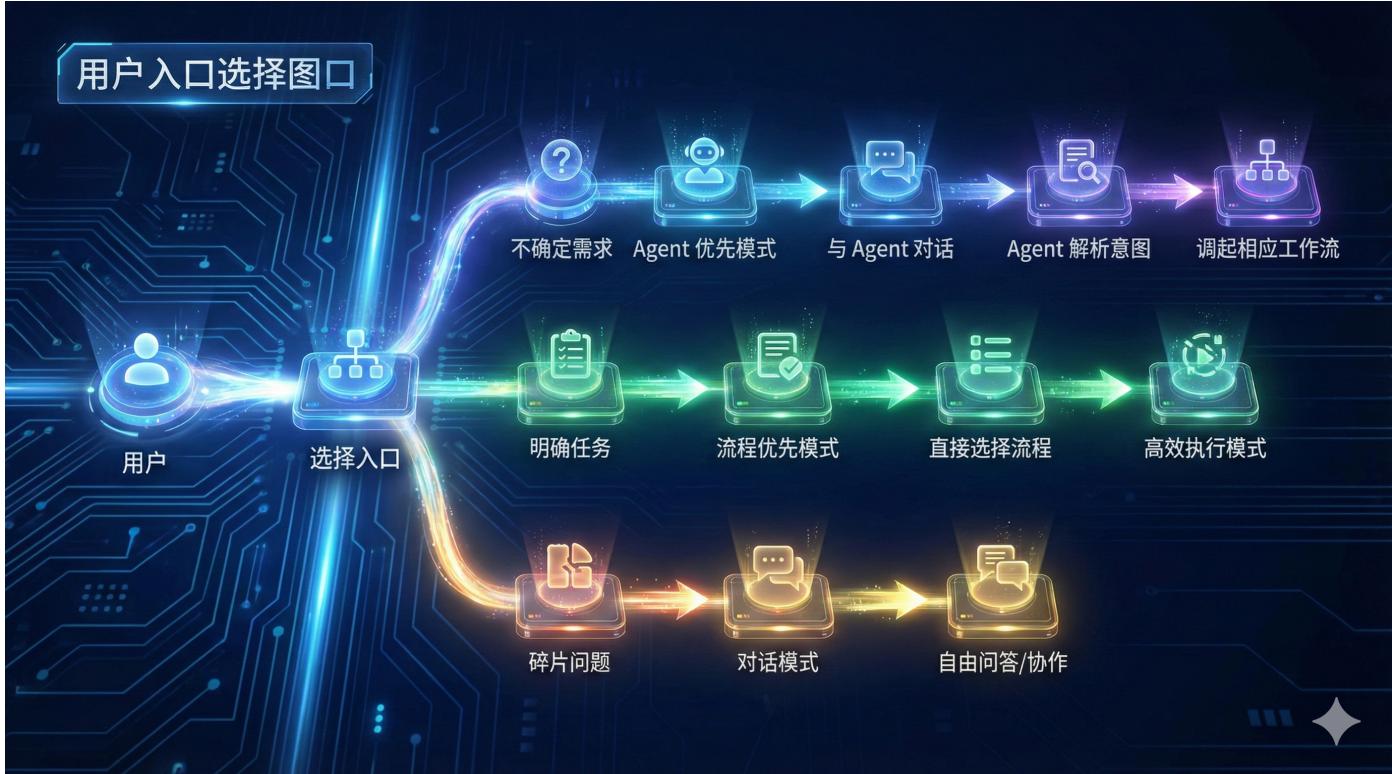
基于 MCP 的无限扩展

"标准协议连接万物"

- 原生支持 Anthropic 提出的 **MCP (Model Context Protocol)** 开放标准
- 不再需要为每个软件单独开发插件，只要该软件提供了 MCP 驱动即可即插即用
- 轻松让 Agent 读取本地的 SQLite 数据库，或调用本地 Python 环境进行科学计算
- 打破了 SaaS AI 无法触及本地工具的壁垒

4. 交互模式

CrewAgent 支持三种入口模式，满足不同场景需求：



模式 A: Agent 优先 (Agent-First)

像咨询顾问一样服务

- 场景: 用户不确定具体要跑哪个流程, 或者只是想找个帮手
- 体验: 用户直接与 Agent (如"资深律师") 对话, Agent 提供服务菜单
- 价值: 降低使用门槛, 用户不需要知道系统里有多少流程

模式 B: 流程优先 (Workflow-First)

像工厂开机一样执行

- 场景: 用户非常明确今天要处理这批数据, 或进行批量作业
- 体验: 用户在项目面板直接选择流程, 点击"运行"进入高效执行模式
- 价值: 最高效的生产力工具, 省去寒暄和对话, 直奔主题

模式 C: 对话模式 (Chat Mode)

像同事一样交流

- 场景: 日常的编程辅助、代码片段解释、临时的头脑风暴或非结构化任务
- 体验: 标准的 Chat 交互界面, 支持上下文多轮对话, 可随时调用工具但无强制流程约束
- 价值: 灵活、即时, 完美解决碎片化、探索性的问题

5. 竞品对比

CrewAgent vs. Dify vs. Coze

特性	CrewAgent	Dify	Coze (扣子)
核心定位	专业桌面 Agent 系统 (本地优先)	LLM 应用开发平台 (BaaS)	社交/C端 Bot 搭建平台
主要用户	工程师、领域专家、专业工作者	开发者、初创团队、IT 部门	内容创作者、普通 C 端用户
执行环境	本地桌面 (Electron)	云端 / 服务器 (Docker)	云端 (字节云)
数据隐私	极高 (数据不离本地)	中 (数据流经其后端)	低 (数据托管在云端)
工具交互	直接操作本地 (文件, CLI, 软件)	API 调用 (HTTP 请求)	API + 社交平台插件
工作流类型	长流程/复杂任务 (文件处理/编码)	请求/响应式 (API/对话为主)	聊天机器人 / 互动流
核心差异	脑手分离架构 (云端定义/本地执行)	开源的 LLMOps 与 RAG 流水线	生态集成 (抖音/微信/飞书)

差异总结

[!IMPORTANT]

对比 Dify: Dify 适合构建 SaaS 应用、企业内部知识库或对外 API 服务。CrewAgent 专为单人/本地的深度工作设计，可以直接操作电脑上的文件（如“帮我重构这个 1GB 的代码仓库”），这是运行在服务器上的 Dify 难以安全做到的。

[!IMPORTANT]

对比 Coze: Coze 是消费级/社交级的“玩乐场”或“轻工具”，运行在云端服务器上。CrewAgent 是一个生产力工具，用于完成复杂的专业任务（写代码、做设计、搞分析），在一个安全、私密的本地环境中运行。

6. 技术架构：流程引擎

核心问题：由谁来驱动流程？

这是与传统平台最根本的架构分歧点。

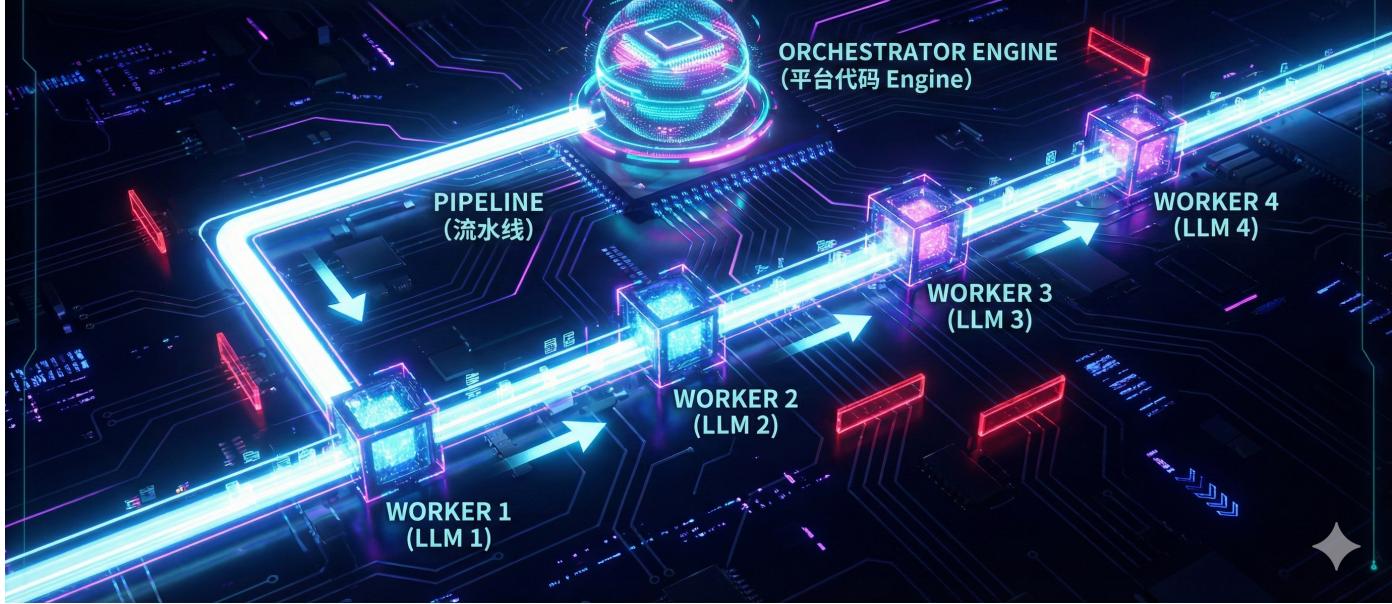
传统模式：编排器驱动 (Orchestrator-Driven)

传统模式：编排器驱动 (Orchestrator-Driven)

机制：代码预设，线性执行。

流程：Engine -> LLM 1 -> LLM 2 -> LLM 3 -> LLM 4。

局限：死板，异常分支需画死，无自主性。



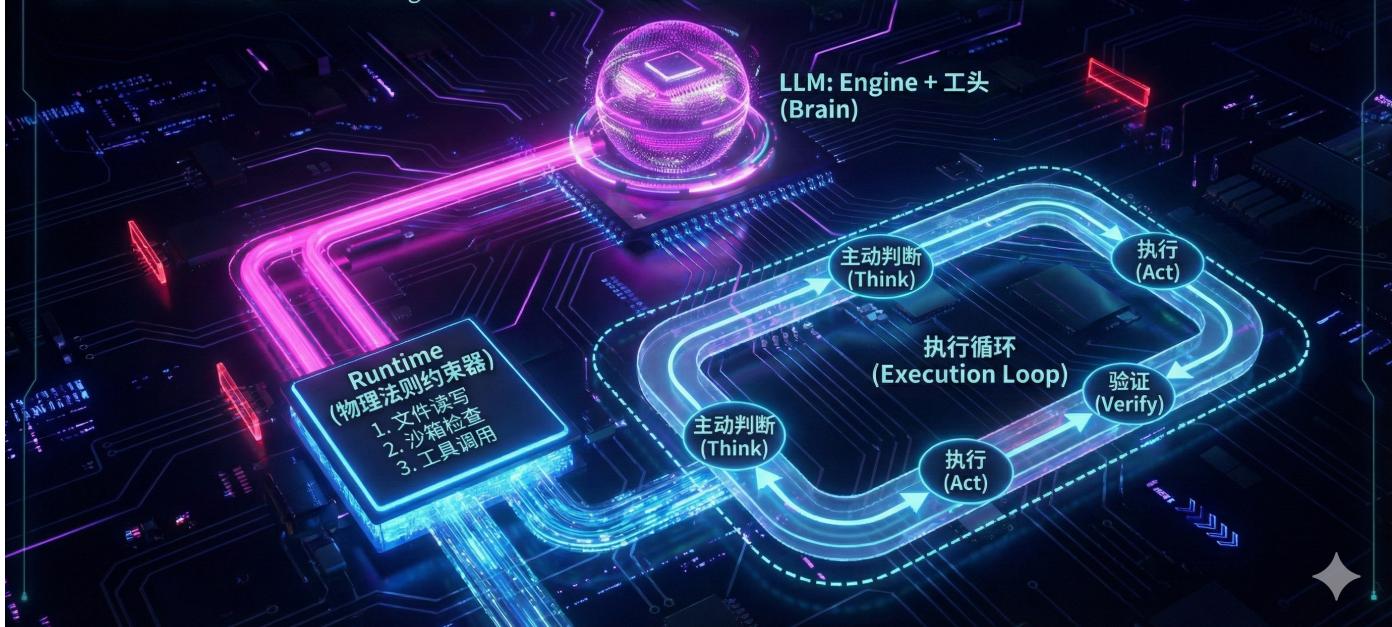
- 机制：类似于传统的工业流水线，平台代码是“工头”，LLM 只是“工人”
- 流程：代码说“执行第 1 步”，把输入喂给 LLM，LLM 吐出输出，代码传给第 2 步
- 局限：非常死板，必须在画图时就把所有异常分支都画死

CrewAgent 模式：Agent 驱动 (Agent-Driven with Guardrails)

CrewAgent 模式：Agent 驱动 (Agent-Driven with Guardrails)

机制：“LLM-as-Engine”（LLM 即引擎），Runtime 只是约束器，真正的“工头”是 LLM 本身。

优势：具备“主观能动性”，允许 Agent 在预定义的 SOP 框架内拥有像人类专家一样的微观自主权。



- 机制：“LLM-as-Engine”（LLM 即引擎），Runtime 只是约束器，真正的“工头”是 LLM 本身
- 流程：LLM 读取当前“案卷”，自己判断当前步骤需要什么，可以在一个节点内自主进行多次工具调用
- 优势：具备“主观能动性”，允许 Agent 在预定义的 SOP 框架内拥有像人类专家一样的微观自主权

7. 应用场景

CrewAgent 适用于各种需要深度本地操作的专业场景：

领域	应用示例
软件开发	代码重构、自动化测试、技术文档生成
数据分析	批量报表处理、数据清洗、Excel 自动化
法律合规	合同审查、法规比对、意见书生成
财务审计	财报分析、底稿整理、数据核对
工业设计	CAD 参数化设计、图纸批量处理 (未来)

8. 总结

CrewAgent 代表了下一代 AI 工具的发展方向：

- 本地优先：数据主权完全掌控
- 专业级执行：不只是建议，而是交付完成的工作
- 知识复制：专家经验可规模化传承
- 开放扩展：基于 MCP 标准，无限扩展可能

[!TIP]

从“手动搬运数据的操作员”转变为“数字专家的监督者”，这就是 CrewAgent 带来的工作方式革命。

CrewAgent - Universal Agent Operating System