北京大学
PEKING UNIVERSITY

自主 Agent（Agentic AI）内部研讨系列

OpenClaw001：龙虾使用入门

技术架构、生态、应用场景

肖睿团队

(李晴)

20260312@北京

- 北大青鸟人工智能研究院
- 北大计算机学院
- 北大教育学院学习科学实验室

OpenClaw 爆火出圈

在通用人工智能（AGI）五级模型中，2023年1月出圈的ChatGPT（GPT3.5）让技术圈外的普通大众感受到自然语言对话时代（第一级智能）的到来，2025年1月出圈的DeepSeek R1让技术圈外的普通大众感受到思考推理时代（第二级）的到来，但二者的产品表现都是“对话式交互”机器人。2026年1月出圈的OpenClaw让技术圈外的普通大众感受到Agent时代（第三级）的到来。

在人工智能由 “**对话式交互**” 向 “**自主行动**” 转型的关键时刻，OpenClaw 的出现标志着自主智能体（自主Agent、Agentic AI）从实验室原型走向工程化、规模化应用。

OpenClaw 不仅在 GitHub 上创下了星标（stars）增长的历史纪录，更通过其独特的 “本地优先” 架构和 “多渠道集成” 理念，重新定义了个人 AI 助手的能力边界。

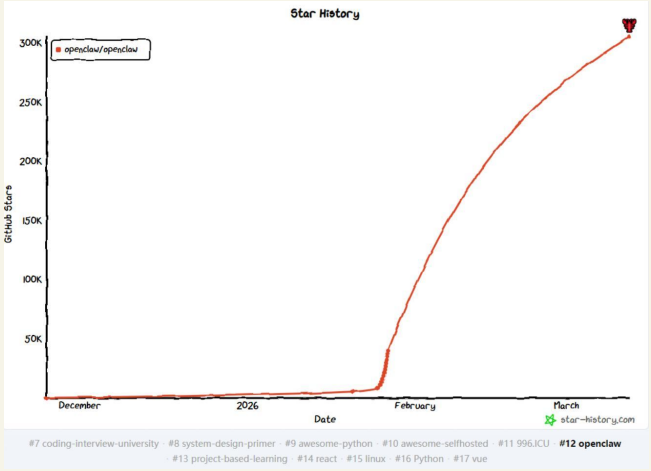
增长奇迹：GitHub 历史上增速最快的开源项目

250,000+

GitHub Stars (截至 2026 年 3 月)

~60 天

从 0 到 25 万 Stars 的耗时



作为一个由 **Peter Steinberger** 个人发起并迅速演变为全球性开源现象的项目，自 2026 年 1 月下旬病毒式传播以来，OpenClaw 在约 60 天内累积超过 250,000 个 GitHub Stars，超越 **React**、**Linux** 等运行了十余年的老牌项目，成为 GitHub 历史上增速最快的软件类开源项目。

2026 年 2 月，创始人 Steinberger 宣布加入 **OpenAI**，项目同步过渡至独立基金会治理。

内容介绍

本讲座旨在从 OpenClaw 的产品定位、出圈原因、技术架构、执行逻辑、核心能力、应用场景、安全挑战、国内平替、未来趋势等多个维度，梳理和分析 OpenClaw 生态，让没有接触过 ClaudeCode、OpenClaw 等自主 Agent 产品的老师、同学、创业者、企业管理者对 OpenClaw 代表的新一代自主 Agent 范式有全面的了解。

章节目录

PART 01

AI的进化阶段

一、AI 的进化阶段

过去两年，AI 的变化速度远远超过了多数人的预期。AI 技术扩散是以一段一段清晰的阶段逐步展开。每一轮突破，都会改变普通人与 AI 的关系。

PHASE 01

能不能用

基础生成能力的普及，AI 开始进入大众视野，解决“有无”问题，实现初步的对话式交互。

PHASE 02

能不能思考

推理模型（Reasoning Models）的突破，AI 具备了更强的逻辑分析与复杂问题拆解能力。

PHASE 03

能不能替人完成任务

自主智能体（Autonomous Agents）时代，AI 不再仅仅是对话，而是能够独立执行长程任务。

一、AI 的进化阶段：五个阶段

阶段	阶段特征
第一阶段	以 ChatGPT 为代表。公众第一次明显感受到 AI 已经具备可用的智能水平，对话、写作、总结等能力开始进入日常使用场景。
第二阶段	以 DeepSeekR1 等模型为代表。大模型展现出更强的思考与推理能力，人们第一次直观感受到 AI 可以参与复杂问题分析，这被很多人称为“认知能力的普及”。
第三阶段	以 Manus、Genspark、Lovable 为代表。云端自主智能体商业化产品出现，用户可以通过输入任务，让 AI 自动完成一整套流程，形成“输入—处理—输出”的闭环。
第四阶段	以 Claude Code 等工具为代表。自主智能体执行能力开始进入本地环境，开发者可以在自己的电脑和开发环境中直接调用 AI 完成复杂任务，本地执行能力逐渐成熟。
第五阶段	以 OpenClaw 为代表。执行能力进一步扩散，不再局限于技术团队，普通用户也可以部署和使用智能体系统，自动化能力开始真正进入大众层面，正式进入自主Agent平民化时代。

“执行能力正在从技术团队快速扩散到普通用户手中。很多过去只有工程师才能搭建的自动化流程，如今普通人也可以直接使用。”

PART 02

OpenClaw 是什么

二、什么是 OpenClaw：产品定位 (I)

自托管的本地优先 AI 助手平台

不同于依赖云端的助手，你在自己掌控的硬件上（家里的 Mac mini、VPS，甚至树莓派）运行一个始终在线的进程 —— **Gateway**。

它连接你日常使用的聊天应用（飞书、QQ、钉钉、企业微信等），接收消息，运行 Agent 推理，可选地调用工具或设备，再将响应发回。

核心组件：Gateway

作为系统的“中枢神经”，Gateway 负责维持长连接、管理会话状态，并协调模型推理与外部工具的调用。它是实现“始终在线”服务的关键。

Mac mini

VPS

树莓派

二、什么是 OpenClaw：产品定位 (II)

任意操作系统

全面支持主流桌面操作系统，包括 **Mac**、**Windows** 以及 **Linux**，确保在各种硬件环境下都能稳定运行。

支持任意模型

灵活接入多种模型：Claude、GPT、Gemini、DeepSeek、智谱、Kimi 以及各类**本地模型**等，不绑定单一供应商。

接入常用 APP

深度集成日常通讯工具：**飞书、QQ、钉钉、企业微信** 等常用平台，让 AI 助手无缝融入现有的工作流。

默认私有

坚持 **Private by default** 理念：所有数据均可选择本地存储，由用户完全掌控，保障极高的数据隐私安全性。

OpenClaw 命名沿革与更名历程

OpenClaw 经历了三次更名，每一次更名都意外为其增加了曝光度：

2025 年 11 月 — **Clawdbot**

项目初始名称。采用 Claude 谐音 + 龙虾爪 Claw 意象，旨在致敬Anthropic的Claude模型，与ClaudeCode吉祥物的名称一样。

2026 年 1 月 29 日 — **Moltbot**

因 Anthropic 对名称与 Claude 品牌相似性表示不满，发律师函敦促更名。Molt 意为“脱壳”（龙虾蜕皮），象征着项目的成长与蜕变。

2026 年 1 月 30 日 — **OpenClaw**

第三次更名，进入稳定开发阶段。Open（开源、社区驱动）+ Claw（保留龙虾 Molty 吉祥物），确立了最终的品牌身份。

在 AI 生态中的位置 (I)：执行层与框架层的核心枢纽

模型层 (Intelligence)	**大语言模型 (LLM)** 提供核心推理能力，如 Claude , GPT, DeepSeek 等。负责理解意图与生成决策。
执行/框架层 (Execution)	**OpenClaw Agent Runtime** **核心定位**：作为轻量级、可自部署的 Agent 运行时。负责任务规划、工具调用、记忆管理及多渠道接入。
应用/接口层 (Interface)	**终端交互应用** 飞书、钉钉、微信、Telegram 等通讯工具。作为用户与 AI 交互的最终界面。

OpenClaw 填补了模型与应用之间的空白，解决了现有方案中**自主性不足、本地化程度低**以及**工具调用门槛高**的痛点。

在 AI 生态中的位置 (II)：产品对比

项目名称	核心定位	部署方式	工具调用	核心优势
ChatGPT 类	自然语言对话机器人	SaaS/云端	受限插件/GPTs	开箱即用，门槛最低，隐私受限
LangChain	非自主Agent开发框架	集成于应用	高度可定制	适合软件开发者构建基于工作流和RAG的AI 应用
AutoGPT	实验性自主 Agent产品	本地运行	支持插件	早期探索，自主性极高但易死循环
Claude Code	闭源免费自主Agent产品	自托管/本地	通用，专注软件开发场景	极致的终端交互与代码理解
OpenClaw	**开源免费自主Agent产品**	**自托管/本地**	**通用，专注日常场景**	**隐私安全、始终在线、多端接入**

- OpenClaw 主要解决了传统 AI 应用在自主性、本地化和多功能集成方面的不足，在产品范围和能力范围上与ClaudeCode类似，但后者往往更加封闭（比如官方建议接入Claude模型，如接入其他模型需要修改配置文件并接受ClaudeCode的性能损失）。
- 许多现有 AI 解决方案通常是云端服务，存在数据隐私风险、高延迟以及对特定应用场景的限制。此外，它们往往缺乏将 AI 能力与实际操作环境（如操作系统、各种软件）深度结合的能力。

PART 03

为什么 OpenClaw 爆火

核心增长数据分析 (I)：现象级的社区热度

30.6w+

GitHub Stars

截至 2026 年 3 月 12 日，OpenClaw 在 GitHub 上的星标数已突破 30 万大关，展示了全球开发者对其技术架构的高度认可。

250万+

单周访问量

项目官网及文档库在爆火期间创下超过 250 万次的单周访问记录，体现了极高的用户活跃度与市场关注度。

10,700+

社区技能插件

ClawHub 市场上已积累超过 1 万个由社区贡献的技能插件，构建了极其丰富的 Agent 生态系统。

核心增长数据分析 (II)：时间线回顾

2025 年 11 月

项目萌芽：Clawdbot 发布

Peter Steinberger 在 GitHub 发布初始版本，定位为致敬 Claude 的本地 AI 助手工具，初期仅在小众开发者圈内传播。

2026 年 1 月

更名与爆发：OpenClaw 定名

经历 Moltbot 短暂更名后，1 月 30 日正式定名 OpenClaw。项目因其卓越的本地执行能力开始在社交媒体病毒式传播。

2026 年 2 月

治理转型：移交基金会

创始人加入 OpenAI，项目同步移交至独立基金会治理。Stars 数量迅速突破 10 万大关，成为全球关注焦点。

2026 年 3 月

历史性时刻：超越 Linux

GitHub Stars 突破 25 万，正式超越 React 和 Linux，刷新 GitHub 历史上软件类项目最快增长纪录。

在 AI 生态中的位置 (III)：核心价值

平民化

- 使用门槛低：支持单文件执行或 Docker 一键部署，无需复杂的开发环境配置，让普通用户也能快速拥有自己的 Agent。
- 方便使用：支持飞书、钉钉、微信、Telegram 等主流平台，将 AI 能力直接推送到用户最常用的通讯界面。

功能强大

- 操作能力强：内置 10,700+ 社区技能，能够轻松调用搜索、代码执行、文件处理等各类外部工具。
- 主动性强：可以24小时在线，并可以主动工作，支持定时和心跳机制

记忆能力

- 分级管理：支持短期记忆、中期记忆、长期记忆的分级管理
- 方便使用：记忆内容用markdown格式文件保存在本地，可以压缩、提炼、备份，可以人工浏览和修改

完全掌控

- 本地优先与私有化部署：所有对话记录与配置均存储在用户本地。可接入本地模型和各种 API 模型，如：Claude、GPT、Gemini、GLM、Kimi、Minimax、DeepSeek等。
- 定制性强：代码开源，支持各种外部插件

国内热潮：全民“养虾”与政府推动 (I)

北京大学 PEKING UNIVERSITY

社会现象

年后爆发的“全民养虾”热潮

2026 年春节过后，国内互联网掀起了一场现象级的 **“全民养虾”** 运动。OpenClaw 从最初的技术极客圈迅速扩散至普通大众，成为社交媒体上的热门话题。

- 从开发者到普通职场人的覆盖
- 社交媒体病毒式传播与口碑效应

政策背景

新质生产力的重要抓手

地方政府将其视为推动 **“新质生产力”** 落地的重要工具，行动非常快速。通过扶持自主智能体生态，旨在提升区域整体的数字化与智能化产业。

- 政府引导与产业政策双重驱动
- 技术平权带来的生产力变革

国内热潮：地方政府政策扶持 (II)

政府推动

多地大力扶持：多个地方政府视OpenClaw为“新质生产力”抓手，密集出台补贴/支持政策，鼓励OPC（一人公司）和AI Agent落地，包括：

- 深圳龙岗区拟推“龙虾十条”，提供免费部署服务、最高200万元补贴（针对代码贡献、具身智能结合项目）。
- 合肥高新区最高1000万元资金扶持。
- 无锡高新区“养龙虾”12条，单项最高500万元。
- 苏州常熟13条举措，最高600万元支持入选人才计划的OPC项目。
- 杭州萧山区等也跟进补贴。

国内热潮：民间部署热潮 (III)

线下沙龙与社群

北京中关村、深圳南山科技园等地出现大量自发的“养虾”技术沙龙。开发者们热衷于分享如何通过 SOUL.md 调教出更具个性的 Agent，社群活跃度极高。

硬件风向标

受 OpenClaw 本地部署热潮影响，Mac mini M4 基础版一度出现断货现象。二手交易平台上的相关硬件价格也随之波动，成为技术驱动消费的典型案例。

上门安装生意兴起

闲鱼等平台涌现出大量“上门安装龙虾”的付费服务。服务内容涵盖环境配置以及多端接入调试，标志着 Agent 部署已形成初步的服务产业链。

- 这些政策直接刺激民间部署热潮：线下沙龙爆满、腾讯总部楼下千人排队免费安装、Mac mini加价断货（因适合本地跑Agent）、甚至上门安装/卸载成生意。
- 部分地方和部门（如中关村）举办OpenClaw相关活动，推动AI Agent普及，视其为降低创业门槛、带动国产模型落地的利器。
- 总体上，地方政府和产业部门一度“卷”得很猛，把OpenClaw当成AI+、一人经济的新风口。

国内大厂布局 (I)：云端部署方案对比

腾讯云 (Tencent Cloud)

通过 **Cloud Studio** 推出一键部署模板，支持 WorkBuddy 深度集成，主打“三分钟拥有个人龙虾”。

阿里云 (Alibaba Cloud)

利用 **计算巢 (Compute Nest)** 提供全自动化部署方案，优化了模型调用链路，并针对新用户提供免费试用额度。

字节跳动 (Volcengine)

火山引擎 与飞书深度联动，推出企业级 Agent 协作方案，主打多端接入与团队共享，强化了办公场景的执行力。

华为云 / 百度云

通过镜像市场提供官方优化的 OpenClaw 镜像，重点支持国产模型（如文心一言、盘古）的无缝接入与性能优化。

- 通过低价/免费提供基础设施（云部署、模型接入、IM渠道），借此拉动自家模型Token消耗（国产模型性价比高，Token出海占比从2024年底2%飙到2026年39%），并通过用户轨迹数据优化模型。
- **中国成为全球OpenClaw最大试验场和采用增速最快市场。**

国内大厂布局 (II)：腾讯 WorkBuddy 与 QClaw

腾讯的“全家桶”策略

腾讯在 OpenClaw 生态中的布局最为全面，不仅在云端全线支持一键部署，更在内部深度实践了 Agent 驱动的办公新范式。

通过将 Agent 能力下沉至基础设施层，腾讯旨在构建一个从模型训练、应用开发到终端接入的完整闭环。

WorkBuddy (内部实践)

- 腾讯内部大规模使用的智能办公助手。
- 深度集成企业微信、腾讯会议及内部知识库。
- 支持自动化会议纪要、日程协调及跨部门任务追踪。
- 标志着 Agent 从“玩具”转向企业级生产力工具。

QClaw (开发者工具)

- 面向外部开发者的开源 Agent 工具集。
- 原生适配腾讯云混元大模型及相关 API。
- 提供丰富的预置技能包，打通微信直接对话+远程操控。
- 强调与腾讯云基础设施的无缝整合与高性能运行。

国内大厂布局 (III)：字节跳动 与 阿里巴巴

字节跳动

ArkClaw

- 与**飞书 (Lark)** 深度绑定，实现消息流与任务流的无缝对接。
- 主打企业级 Agent 协作，支持多端接入与团队共享。
- 利用火山引擎的算力优势，优化了长文本处理与复杂推理性能。
- 提供丰富的行业模板，降低企业定制化 Agent 的门槛。

阿里巴巴

CoPaw

- 阿里云推出的开源 Agent 工具，强调与通义千问模型的深度整合。
- 阿里云提供多种一键部署，轻量服务器/无影云电脑/ECS。
- 优化了模型调用链路，显著降低了推理延迟与成本。
- 支持多种存储方案，满足企业对数据隐私的严苛要求。

国内大厂布局 (IV)：网易 LobsterAI 与月之暗面 KimiClaw

网易有道 LobsterAI

被社区亲切地称为**“中国版龙虾”**。网易有道将其完全开源，旨在为国内用户提供更贴合本土习惯的 Agent 体验。

完全开源 **中文语境优化** **深度集成中国应用**

针对国内办公软件的 API 进行了大量适配工作，解决了原版在中文环境下可能出现的字符编码与连接稳定性问题。
强调安全（本地存储 + 隔离虚拟机执行敏感操作）

月之暗面 KimiClaw

由月之暗面官方推出的**云端托管版** Agent 助手，旨在让不具备本地部署能力的普通用户也能享受 Agent 的便利。

云端托管 **长文本能力** **零门槛使用**

充分发挥了 Kimi 大模型在超长上下文处理上的优势，支持对海量本地文档进行深度分析与任务执行。
绑定Kimi K2.5模型，支持长期记忆 + 远程电脑操控 + 40GB免费云存储。零代码一键创建，适合内容创作/信息检索。

国内大厂布局 (V)：其他厂商动态汇总

MiniMax

推出 **MaxClaw** 实验项目，重点优化多模态 Agent 交互体验，提升模型在复杂指令下的执行成功率。预置专家技能 + 跨应用自动化 + 定时任务。推出后服务器扩容4次，增长极快。

智谱 AI

发布 **AutoClaw** 框架，实现 GLM 系列模型与 OpenClaw 协议的原生适配，强化了国产大模型的工具调用能力。强调零成本体验 + 多梯度付费。

百度

DuClaw（零部署云服务，17.8元/月起）+ 百度App内嵌 + 千帆平台上架Skills。

小米

探索 **端侧 Agent** 部署方案，计划将轻量化 OpenClaw 核心集成至澎湃 OS，实现手机端的本地自动化执行。Xiaomi miclaw（手机系统层Agent，适配小米17系列，调用50+系统工具 + IoT）

华为

小艺 Claw 结合鸿蒙生态，研究跨设备 Agent 协同技术，利用 OpenClaw 的网关架构实现多终端间的任务无缝流转。支持多端协同 + 多种人格。

爆火深层原因分析：技术、开源与社交传播

Agent 技术成熟

大语言模型（LLM）推理能力的质变，使得 AI 不再仅仅是对话，而是具备了理解复杂意图并拆解执行任务的“大脑”，为自主智能体的爆发奠定了技术基础。

开源生态加速

OpenClaw 坚持开源路径，通过极低的部署门槛和丰富的社区技能包（ClawHub），迅速吸引了全球开发者参与，形成了强大的网络效应和生态护城河。

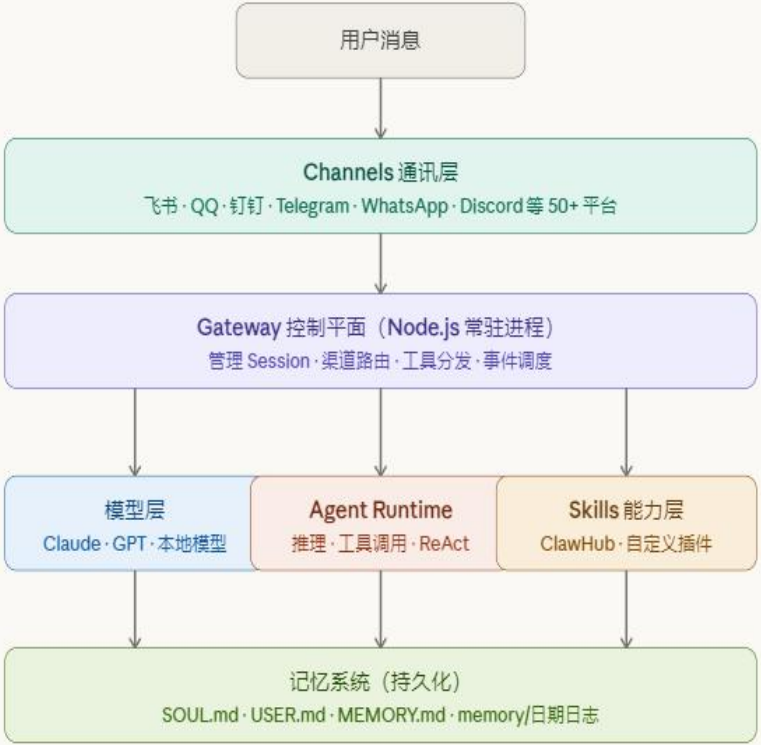
社交传播放大

从 GitHub 趋势榜到 X (Twitter)、飞书社群，再到国内的短视频平台，OpenClaw 的增长曲线完美契合了病毒式传播规律，实现了从技术圈到大众市场的跨越。

PART 04

技术架构拆解

四、技术架构拆解 (I)：整体架构概览

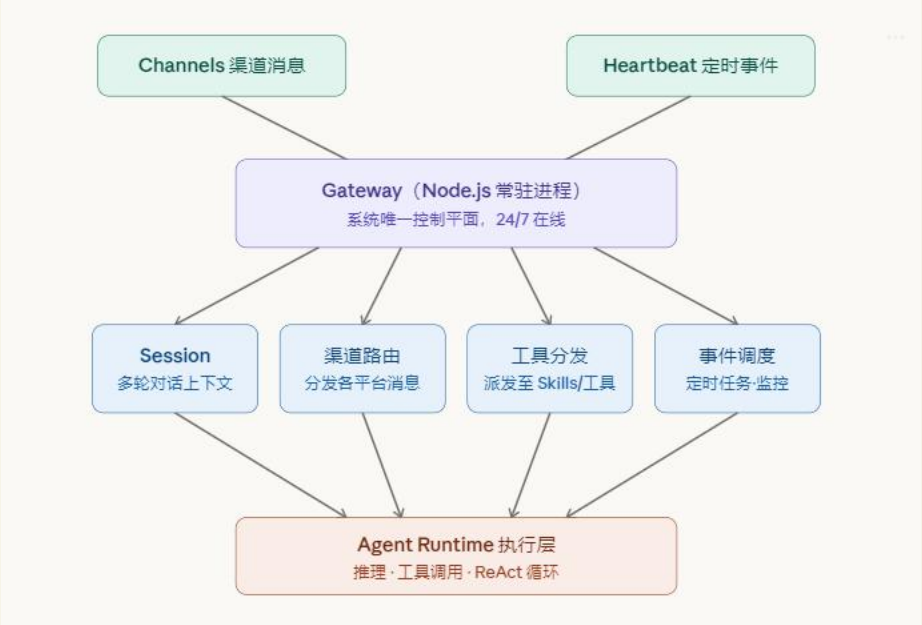


- - Gateway 是始终在线的控制平面，管理 Session、渠道路由、工具分发和事件；
- - 模型负责提供智能；
- - Channels 是各消息平台的集成适配层；
- - Agent Runtime 是模型推理与工具调用的执行引擎；
- - Skills 是通过 ClawHub 市场可动态安装的能力扩展。

OpenClaw 本质上是一个本地网关 + Agentic 循环 + Skills + 持久化记忆的组合，这一架构模式正在成为个人 AI Agent 的标准蓝图。

初始配置有一定门槛，运行云端模型会产生 Token 费用，且由于具备系统级访问权限，安全配置不可忽视。

四、技术架构拆解 (II)：Agent 调度层 (Gateway)



这是系统的核心逻辑层，通常以“Gateway（网关）”守护进程的形式运行在后台 。负责管理 Agent 的生命周期和任务执行流程。

OpenClaw 的调度层强调执行稳定性与可扩展性，确保 Agent 能够可靠地完成任务，并支持未来功能的扩展。

四、技术架构拆解 (II)：Agent 调度层 (Gateway)

任务规划 (Planning)

作为 Agent 的“前额叶”，负责将用户模糊的指令拆解为一系列逻辑严密的子任务，并确定执行的先后顺序与依赖关系。

工具选择 (Tool Selection)

根据当前任务的上下文需求，从技能库中动态检索并匹配最合适的插件工具，确保 Agent 能够精准调用外部能力。

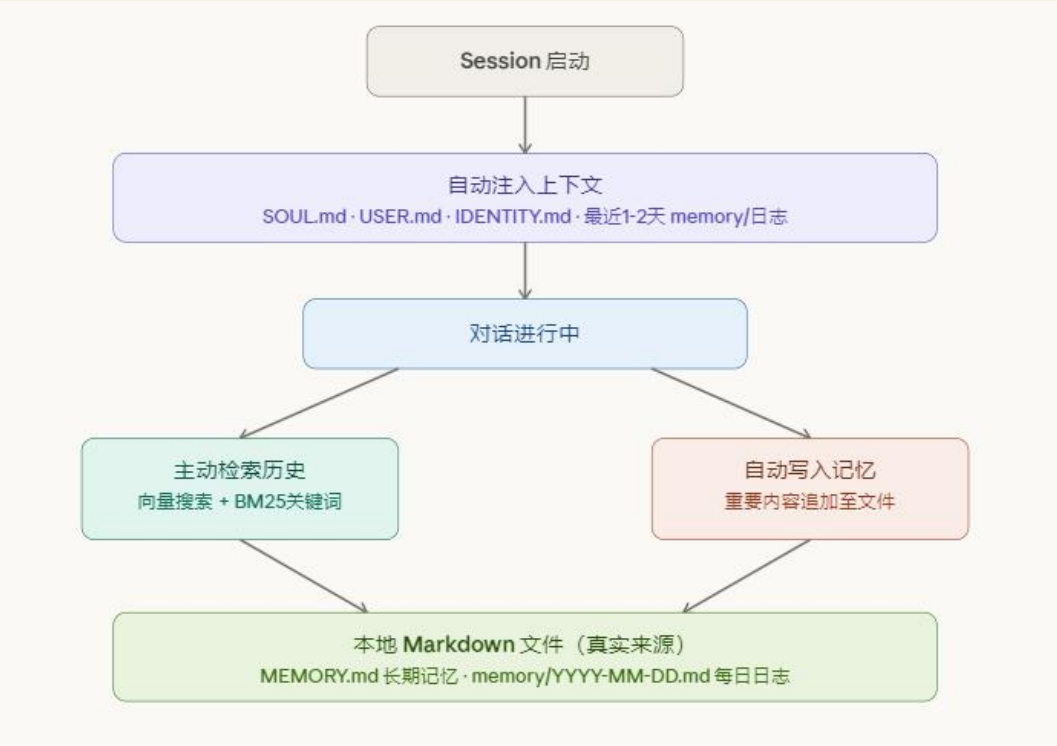
状态管理 (State Management)

实时维护会话上下文、环境变量及 Agent 的“脑内笔记”，确保在多轮交互中能够保持逻辑连贯性与记忆持久化。

事件调度 (Scheduling)

协调消息的实时收发、定时任务的触发以及异步回调逻辑，确保系统在处理复杂长流程任务时的稳定与高效。

四、技术架构拆解 (III)：记忆系统



文件结构： 所有记忆以纯文本 Markdown 存储在本地目录：

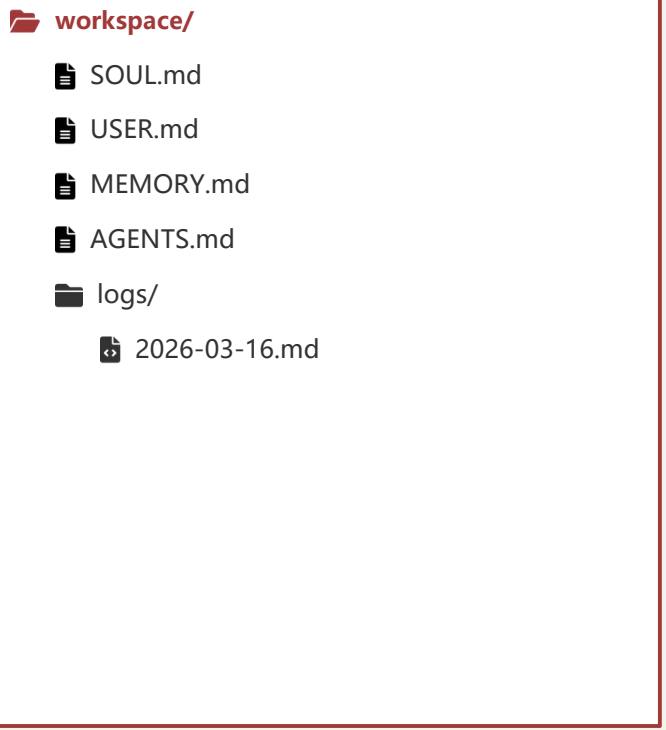
memory/YYYY-MM-DD.md 是每日会话日志，

MEMORY.md 是用户维护的精炼长期记忆，

AGENTS.md 和 SOUL.md 是每次 Session 都会注入的行为指令和人格定义。

这是 OpenClaw 架构中最具特色的设计，采用文件为真实来源、向量索引为检索加速的混合方案。

四、技术架构拆解 (III)：记忆系统 - 文件结构 (1)



SOUL.md (灵魂文件)

定义 Agent 的核心行为准则、性格设定及价值观。它是 Agent 思考和行动的最高指导原则。

USER.md (用户画像)

存储用户的个人偏好、工作背景及常用信息。帮助 Agent 提供更具个性化和上下文相关的服务。

MEMORY.md (长期记忆)

记录跨会话的重要事实、决策结果及长期知识。通过 RAG 机制在需要时被检索调用。

AGENTS.md (协作配置)

定义多智能体协作模式，包括不同 Agent 的职责分工、通讯协议及权限边界。

四、技术架构拆解 (IV)：记忆系统 - 文件结构 (2)

北京大学 PEKING UNIVERSITY

脑内笔记逻辑

Internal Monologue：Agent 的思考轨迹

OpenClaw 引入了 **“脑内笔记”** 机制。Agent 在执行复杂任务时，会实时记录其思考过程、假设验证及中间结论。

- **非展示性记录**：思考过程不直接输出给用户，避免干扰，但作为后续决策的核心参考。
- **自省与修正**：Agent 可以回顾之前的思考轨迹，发现逻辑漏洞并及时进行自我修正。

跨会话存储

持久化记忆：打破对话孤岛

通过持久化的 Markdown 文件结构，OpenClaw 实现了真正的 **“跨会话记忆”**，让 Agent 能够记住用户在不同时间、不同平台提到的信息。

- **信息共享**：不同会话间的信息可以相互引用，构建起一个完整的用户知识图谱。
- **真实来源追溯**：所有记忆均对应真实的文件路径，确保 Agent 的输出具备极高的可信度与可追溯性。

四、技术架构拆解 (V)：记忆系统 - 检索机制 (RAG)

向量索引 (Vector)

利用 Embedding 技术将 Markdown 文本转化为高维向量。通过余弦相似度计算，实现基于语义的深度检索，能够捕捉到用户意图与历史记忆之间的深层关联。

关键词匹配 (BM25)

引入经典的 BM25 文本匹配算法。针对专有名词、特定日期或唯一标识符提供极高的召回精度，弥补了向量检索在精确匹配方面的不足。

时间衰减 (Decay)

在检索权重中引入时间因子。越近发生的记忆片段拥有越高的权重分值，确保 Agent 能够优先响应当前的上下文环境，模拟人类的记忆遗忘曲线。

混合搜索 (Hybrid)

将上述多路检索结果进行重排序（Rerank）。通过加权融合，为大模型提供最相关、最精准的记忆上下文，显著提升了 Agent 在长周期任务中的表现。

局限：规模化后检索效率下降，长期上下文压缩时存在信息丢失风险，Markdown 不捕获概念间的语义关联关系。

五、技术架构拆解 (VI)：工具层 (Tool System / Skills)

∗工具层是 Agent 的“手脚”，决定了 Agent 的实际操作能力。

∗OpenClaw 提供了丰富的内置工具，并支持通过插件机制进行扩展 。

Skills 以目录形式组织，每个技能包含 SKILL.md 元数据和工具使用说明，可通过 ClawHub 市场安装，也可自行编写部署。

四、技术架构拆解 (VI)：工具层 (Tool System / Skills)

内置基础工具 (Core Tools)

提供 Agent 运行所需的原子化能力。包括浏览器操控、文件系统读写、终端命令执行、API 调用。这些工具是构建复杂任务的基石。

File System　Web Search　Shell Exec

高级技能包 (Advanced Skills)

针对特定垂直领域封装的复杂逻辑集合。例如：自动化的财务报表分析、多语言代码审计、社交媒体内容自动分发等。支持通过 YAML 或 JSON 进行灵活配置。

Financial Analysis　Code Audit　Social Media

ClawHub 市场与 MCP 协议

拥有超过 10,700 个由社区贡献的开源技能。全面支持 Model Context Protocol (MCP) 协议，实现了跨平台的工具标准化接入，让 Agent 能够即插即用地扩展其能力边界。

10,700+ Skills　MCP Support　Community Driven

四、技术架构拆解 (VII)：通讯层 (Channel Layer)

IM 平台适配

OpenClaw 通讯层实现了对主流即时通讯平台的深度适配，确保 Agent 能够像真人一样在社交软件中进行交互。

- 飞书 (Lark) / 钉钉 (DingTalk)
- 微信 (WeChat) / 企微
- Telegram / Discord / Slack

Web 控制面板

为用户提供直观的图形化管理界面，降低了 Agent 的配置与监控门槛，实现“所见即所得”的操控体验。

- 实时会话监控与调试
- 技能插件一键开关
- 知识库与记忆文件管理

API 扩展接口

开放标准的 RESTful API 与 Webhook 接口，支持将 Agent 能力无缝集成至企业现有的业务流程中。

- 自定义 Webhook 触发
- 跨系统任务流自动化
- 第三方应用插件化集成

通信层是 OpenClaw 与用户交互的“接口”，支持通过不同渠道与用户进行沟通。这种设计使得 AI Agent 更像一个“数字助手”，无缝融入用户的日常工作和生活。

四、技术架构拆解 (VIII)：大模型是OpenClaw的大脑

云端 API 模型

通过 OpenAI 兼容协议接入 Claude 4.6 OPus、GPT-5.4 等顶级模型。

- ✓ **极致性能**：处理复杂逻辑与长文本推理的首选。
- ✓ **零维护**：无需本地算力支持，即开即用。
- ✓ **快速迭代**：始终访问厂商最新的模型能力。

本地私有模型

利用 Ollama、LM Studio 等引擎在本地运行 Llama 、Qwen 等开源模型。

- ✓ **数据隐私**：所有敏感信息均在本地处理，不上传云端。
- ✓ **零成本**：除硬件投入外，无后续 Token 消耗费用。
- ✓ **离线运行**：在无网络环境下依然保持核心功能可用。

混合调度模式

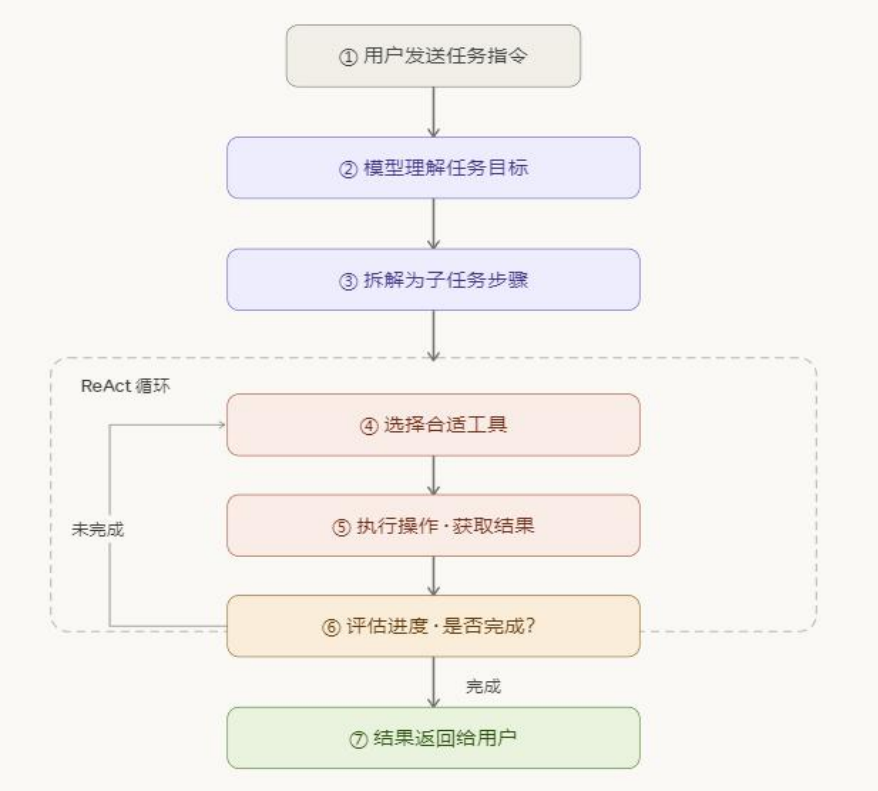
根据任务复杂度动态选择模型，实现性能与成本的最佳平衡。

- ✓ **智能路由**：简单任务本地处理，复杂任务调用云端。
- ✓ **高可用性**：云端故障时自动降级至本地模型执行。
- ✓ **成本优化**：显著降低大规模自动化任务的 API 开支。

PART 05

Agent运行流程

五、Agent 运行流程：典型Agent执行流程析



在复杂任务中，步骤 4-6会多次循环执行，直到任务完全完成。这个循环过程正是 ReAct 模式的体现，确保了 Agent 能够自主地进行多步推理和行动。

五、Agent 运行流程 (I)：ReAct 模式深度解析

思考 (Reason)

Agent 接收当前任务与环境反馈，利用大模型的推理能力进行深度分析，制定下一步的行动计划或调整现有策略。

行动 (Act)

根据思考得出的结论，Agent 调用具体的工具（如搜索、执行代码、读写文件）来执行具体的任务步骤。

观察 (Observation)

Agent 接收工具执行后的返回结果，将其作为新的上下文信息，用于下一轮的思考与决策。

自主循环机制

ReAct 模式通过“思考-行动-观察”的不断循环，使得 Agent 能够处理具有不确定性的复杂任务，实现真正的自主性。

可解释性增强

每一轮的思考过程都会被记录在“脑内笔记”中，用户可以清晰地看到 Agent 做出决策的逻辑依据，增强了系统的透明度。

错误自我修正

通过观察执行结果，Agent 能够发现行动中的偏差或错误，并在下一轮循环中主动调整策略，显著提升了任务完成的成功率。

五、Agent 运行流程 (II)：任务执行典型步骤 (1-7)

01 **接收输入 (Input)**

系统接收来自用户指令、IM 消息或外部 Webhook 触发的原始输入数据，作为任务执行的起点。

02 **意图识别 (Intent)**

利用大模型分析输入的语义，识别用户的真实意图、提取关键参数，并判断任务的复杂程度。

03 **上下文检索 (RAG)**

根据意图从本地 Markdown 记忆库中检索相关背景信息、历史记录或用户偏好，构建增强上下文。

04 **任务拆解 (Planning)**

将复杂目标拆解为一系列可执行的子任务，制定初步的执行路径，并确定所需的工具集。

05 **工具调用 (Action)**

Agent 根据计划调用具体的技能插件（如搜索、写文件、发邮件），与外部系统进行实际交互。

06 **结果观察 (Observation)**

获取工具执行后的反馈结果，分析执行是否成功，并根据反馈更新当前的“脑内笔记”状态。

07 **最终决策 (Decision)**

汇总所有执行结果与观察，判断任务是否完成。若完成则给出最终回复，否则进入下一轮 ReAct 循环。

从用户角度看，一次典型的 OpenClaw 任务流程遵循 ReAct 模式，可以概括为以上的步骤 。

PART 06

核心能力与应用场景

六、核心应用场景 (I)：个人效率提升场景

资料整理与知识内化

Agent 能够自动汇总来自网页、PDF、本地文档等多源信息，识别核心观点并生成结构化的 Markdown 笔记。支持跨文档的知识关联与深度问答，显著缩短了从获取信息到内化知识的路径。

全流程代码开发

Agent 能进行实时的代码审查、提供重构建议、自动生成单元测试及技术文档。通过对项目全局上下文的理解，能够协助开发者处理复杂的跨文件逻辑修改与 Bug 调试。

自动化文件处理

支持对海量文件进行批量重命名、格式转换、内容提取及分类归档。
例如：从数百张发票中自动提取金额并汇总至 Excel，或将大量会议录音转录为文字并提炼核心纪要。

邮件与日历智能管理

自动筛选并分类重要邮件，根据上下文草拟回复。同时能够协调多方日程，自动在日历中创建会议提醒并附带相关背景资料，确保个人工作流的高效运转与无缝衔接。

六、核心应用场景 (II)：自动化任务处理场景

网页抓取与分析

Agent 能够自主访问目标网站，利用内置工具提取关键数据，并结合大模型的理解能力进行深度分析，最终生成结构化的研究报告或数据摘要。

竞品动态监控

通过定时任务触发，Agent 实时追踪竞争对手的官网更新、社交媒体动态及新闻报道。一旦发现关键变动，立即自动发送预警通知至用户指定的 IM 平台。

社交媒体内容发布

根据用户预设的主题或关键词，Agent 自动搜集素材、撰写文案并进行多平台（如微博、小红书、X）的内容分发，实现全自动化的社交媒体运营流水线。

智能日程与提醒

Agent 能够自动读取邮件或消息中的会议邀请，协调多方时间冲突，自动创建日程并同步至用户的日历系统，同时在关键节点发送智能提醒。

六、核心应用场景 (III)：定制化商业应用与服务产业

信息差业务

利用 Agent 强大的网页抓取与多源数据分析能力，快速识别并提取行业高价值信息。通过自动化的情报分发系统，为客户提供实时的竞品监控、政策分析及市场趋势预测，构建基于信息差的商业壁垒。

内容生产流水线

构建从选题策划、素材收集、初稿生成到多平台自动发布的完整内容生产闭环。Agent 能够根据不同平台的风格偏好自动调整文案，显著提升内容产出的效率与质量，实现低成本、规模化的内容运营。

部署咨询服务

针对企业和个人用户提供专业的 Agent 部署咨询服务。涵盖硬件选型建议、私有化环境搭建、SOUL.md 深度调优以及员工操作培训。通过定制化的解决方案，帮助客户快速建立并掌握属于自己的 AI 生产力工具。

PART 07

如何养一只龙虾

七、如何养一只龙虾 (I)：推荐本地部署优势对比

OpenClaw 的本地部署特性是其区别于许多云端 Agent 的关键优势，尤其在速度、数据和工具三个方面体现出显著差异。

对比维度	本地部署 (推荐)	云端托管部署
响应速度	**极低延迟：** 数据处理与模型推理均在本地完成，不受网络带宽与波动影响，交互体验极其流畅。	**受限网络：** 依赖互联网连接，存在明显的网络延迟，且在高并发时段可能出现响应缓慢。
数据隐私	**绝对安全：** 所有敏感数据、记忆文件及私密对话均存储在本地磁盘，数据不出域，完全由用户掌控。	**潜在风险：** 数据需上传至第三方服务器，存在隐私泄露或被用于模型训练的潜在风险。
工具集成	**深度调用：** 可无缝调用本地文件系统、Shell 脚本及各类专业软件 API，实现真正的全系统自动化。	**沙箱限制：** 受限于云端安全沙箱，无法直接操作本地硬件与文件，工具调用能力受到极大限制。

七、如何养一只龙虾 (II)：多系统安装指南路径汇总

北京大学 PEKING UNIVERSITY

macOS 部署路径

- → 安装 Node.js 环境 (推荐 v20+)
- → 克隆 OpenClaw 核心仓库
- → 安装依赖并初始化工作区
- → 运行启动脚本进入交互模式

Windows (WSL) 路径

- → 启用 WSL2 并安装 Ubuntu 镜像
- → 在 Linux 子系统中配置 Node 环境
- → 挂载本地磁盘作为工作区存储
- → 配置内网穿透实现多端接入

云端 / Docker 路径

- → 拉取官方 Docker 镜像文件
- → 配置环境变量与 API 密钥
- → 使用 Compose 一键启动服务
- → 通过 Web 控制面板进行管理

```
docker pull openclaw/core
docker-compose up -d
```

龙虾需要24小时连续开机，需要大量支持bash的命令和成熟的权限管理系统。推荐本地部署，优选Mac、其次Linux、不建议直接使用 Windows 原生环境（可考虑 WSL 或容器）。 Mac里优选Mac mini。

七、如何养一只龙虾 (III)：国内模型性能表现与 PinchBench 基准测试

中国模型在 OpenClaw Agent 任务的评分（右图）

- Token 价格低（经常只有 Claude / GPT 的 1/8–1/15）
- 代理 / 工具使用/函数调用能力迭代非常快
- 中文理解 & 执行力强
- 调用延迟低（尤其走国内 CDN）

模型名称 (Model)	任务成功率 (Success)	响应速度 (Speed)	Token 成本 (Cost)
moonshotai/kimi-k2.5	84.8%	978s	$0.27
qwen/qwen3.5-122b-a10b	84.5%	677s	$0.43
qwen/qwen3.5-plus-02-15	84.1%	898s	$0.51
z-ai/glm-5	84.1%	1396s	$0.56
qwen/qwen3.5-397b-a17b	83.6%	886s	$0.83
stepfun/step-3.5-flash	82.6%	845s	$0.26
minimax/minimax-m2.1	82.2%	943s	$0.13
deepseek/deepseek-v3.2	81.9%	2760s	$0.22
z-ai/glm-5-turbo	81.3%	1118s	—
xiaomi/mimo-v2-flash	80.8%	1818s	$0.30
minimax/minimax-m2.5	80.5%	1158s	$0.16
qwen/qwen3.5-27b	80.4%	1118s	$0.45
z-ai/glm-4.5-air	80.3%	1116s	$0.12
qwen/qwen3-max-thinking	80.3%	1864s	$1.87

数据来源：PinchBench，更新于 2026/03/16。速度为最优完成时间（秒），成本为最优单次运行费用。

七、如何养一只龙虾 (IV)：国内模型厂商选择与策略建议

模型厂商	核心优势	适用场景
Step 3.5 (阶跃星辰)	逻辑推理能力极强，指令遵循度高。	日常主力、复杂多步任务、代码、搜索
M2.5（MiniMax)	响应速度极快，交互体验好。	工具调用稳定、长对话、记忆任务
K2.5(月之暗面)	超长上下文处理能力，文档理解精准。	深度研究、写长文、复杂规划
Qwen (通义千问)	多模态、视觉任务、本地化需求	开源版可本地跑，云版也很便宜。
GLM-5系列（智谱）	工具调用格式最标准，兼容性好	均衡型、函数调用规范

快速养虾策略：建议采用Step 3.5 Flash 或 MiniMax M2.5 Flash作为日常主力（最便宜、最快、基本能cover 80%场景），复杂规划/研究/写长东西采用Kimi K2.5，视觉/多模态任务可采用通义千问 Qwen3.5系列 或 GLM-4V系列，本地党LM Studio 跑 Qwen3.5-32B / DeepSeek 系列 量化版。

七、如何养一只龙虾 (III)：国际模型性能表现与 PinchBench 基准测试

模型名称 (Model)	任务成功率 (Success)	响应速度 (Speed)	Token 成本 (Cost)
anthropic/claude-sonnet-4.6	86.9%	938s	$1.50
openai/gpt-5.4	86.4%	959s	$1.34
anthropic/claude-opus-4.6	86.3%	1033s	$2.43
nvidia/nemotron-3-super-120b-a12b	85.6%	777s	—
anthropic/claude-opus-4.5	85.4%	825s	$3.24
openrouter/healer-alpha	84.1%	754s	—
anthropic/claude-sonnet-4.5	83.1%	907s	$2.14
anthropic/claude-haiku-4.5	82.0%	662s	$0.60
google/gemini-3.1-pro-preview	81.1%	834s	$1.14
anthropic/claude-sonnet-4	80.5%	932s	$2.87
x-ai/grok-4.1-fast	80.0%	921s	$0.23
healer-alpha	78.9%	708s	—
mistralai/devstral-2512	78.8%	644s	$0.65
openai/gpt-5-mini	78.3%	742s	$0.19
openrouter/hunter-alpha	77.8%	1360s	—

*数据来源：PinchBench，更新于 2026/03/16。速度为最优完成时间（秒），成本为最优单次运行费用。— 表示数据未显示。

国际模型在 OpenClaw Agent 任务的评分（左图）

国际模型（主要是美国系）在 OpenClaw 生态里的地位和国内模型完全相反：

- 贵但稳、推理深度强、工具调用最规范、prompt 抵抗力高。
- 尤其适合那些“一次就必须做对”、或者涉及高价值任务（代码安全、企业自动化、复杂多工具链）的场景。

七、如何养一只龙虾 (IV)：国际模型厂商选择与策略建议

极致稳定 + 一次成功率最高

Claude Opus 4.6

目前公认 OpenClaw “天花板”，尤其长任务不崩，逻辑极其缜密。

预算中等、最强生态与规范

GPT-5 系列

Codex 变体特别适合写代码的 Agent，函数调用规范，生态支持最广。

多模态/文档/视频重度任务

Gemini 3 Pro 系列

原生支持多种工具，上下文窗口巨大，处理超长多模态任务优势明显。

极致省钱但仍选国际模型

Claude Haiku 4.5 / GPT-5.4 mini

日常 80% 场景够用，成本仅为旗舰模型的 1/10–1/20。

尝鲜新架构与非大厂方案

Grok 4.x：并行思考在某些规划任务有惊喜，适合不想完全依赖传统大厂的用户。

七、“养虾”混合策略：任务路由与预算优化建议

简单/高频/日常任务

Step 3.5 Flash / MiniMax M2.5 Flash / Claude Haiku

核心目标：成本最低，响应最快。

中等复杂、需要思考深度

Kimi-research / Gemini 3 Pro Flash / GPT-5 mini

平衡性能与成本，适用于日常深度办公。

核心规划与长时自主运行

Claude Opus 4.6 / GPT-5.4

最难任务、写重要代码、核心规划，直接上王牌。

视觉/多页文档/视频分析

Gemini 3.1 Pro / 通义千问视觉版

原生视觉分析能力强，适合多模态理解。

预算无限追求极致方案

全程使用 **Opus 4.6 + Tool Search 开启**。Anthropic 官方表示该组合能持续运行 30+ 小时不崩。

七、如何养一只龙虾 (VI)：工作区配置、加载顺序与安全建议

核心配置文件

SOUL.md：定义 Agent 的性格、价值观及核心行为准则。
SKILLS.md：声明 Agent 可调用的工具集与技能边界。
CONFIG.json：存储 API 密钥、模型路由及系统级参数。

系统加载顺序

1. **环境初始化**：读取 CONFIG.json，建立网络连接。
2. **技能注册**：解析 SKILLS.md，加载本地与远程工具。
3. **灵魂注入**：加载 SOUL.md，确立 Agent 的认知模型。
4. **任务循环**：进入 ReAct 循环，开始执行用户指令。

安全防护建议

密钥安全：严禁将包含 API Key 的配置文件上传至公开仓库。
权限控制：为工作区目录设置严格的读写权限，防止越权访问。
定期备份：定期导出 SOUL.md 与记忆文件，防止数据丢失。

七、如何养一只龙虾 (VII)：工作区初始化与目录结构详解

初始化工作区

运行以下命令可创建工作区并在缺失时生成引导文件。工作区是 Agent 的“私人办公室”，是进行文件操作、存储记忆和执行技能的核心目录。

```
openclaw configure
# 或使用
openclaw setup
```

默认路径：

工作区根目录通常位于~/.openclaw/workspace，与存储配置、凭据和会话的 ~/.openclaw/ 目录相互独立。

* 简而言之，这些文件共同构成了 Agent 的“大脑”和“记忆库”。配置质量直接决定了 Agent 的“好用程度”。

目录结构示例

```
~/.openclaw/workspace/
├── SOUL.md          # 核心人格定义（强烈建议先写这个）
├── USER.md          # 关于你自己的信息（很重要）
├── IDENTITY.md      # Agent的名字、外在形象与自我认知（推荐添加）
├── AGENTS.md        # Agent列表、工作流程、安全规则（单Agent时可简短）
├── TOOLS.md         # 工具说明与使用规范（可选，系统可自动补充）
├── HEARTBEAT.md     # 自检/自省/定时主动任务清单（可选，但很有用）
├── MEMORY.md        # 长期关键事实（AI自动追加，也可手动加）
├── memory/          # 自动生成的目录（RAG检索基础）
│   ├── lancedb/     # 向量数据库（别手动动）
│   └── YYYY-MM-DD.md   # 每日记忆/会话日志（自动追加，最近1–2天常加载）
└── skills/          # 安装的技能目录（每个技能一个子文件夹，核心文件为SKILL.md）
    └── <skill-name>/   # 示例：web-search/、code-review/ 等
        └── SKILL.md
```

七、如何养一只龙虾 (VIII)：SOUL.md 与 USER.md 配置实战

SOUL.md：Agent 的灵魂

定义性格：决定了 Agent 的沟通风格（简洁、幽默或严谨）。
价值观：确立 Agent 的决策边界和核心行为准则。

> **关键说明：**
> SOUL.md 是实际使用中影响最大的文件之一，配置质量直接决定了 Agent 的“好用程度”。

USER.md：用户画像

个人偏好：喜欢的编程语言、工作流习惯、常用的软件工具。
沟通禁忌：明确 Agent 不应触碰的敏感信息或人生建议。

AGENTS.md：安全规则

定义哪些 Shell 命令允许执行。默认值较保守，生产环境建议手动审查，防止乱调用工具或泄露信息。

七、如何养一只龙虾 (IX)：加载顺序与 Token 优化建议策略

系统加载顺序（大致）

Session 启动 → 读 SOUL/USER/AGENTS → 读 MEMORY.md → 读 memory/*.md → 匹配技能 → 进入对话

Token 消耗与优化建议

1. MEMORY.md（越养越大）：
长期记忆文件会随对话累积而膨胀。建议定期精简，将不重要的事实移至 memory/历史文件夹中。

2. SOUL.md（内容过长）：
很多人为了追求性格细节写得太长。建议将 SOUL.md 控制在 1500 字以内，以平衡性格表现与 Token 成本。

3. USER.md + AGENTS.md：
这些文件每次对话都会注入。保持内容精炼，仅保留核心偏好和安全规则。

混合路由策略：
简单/高频任务走便宜模型（如 Step 3.5 Flash），复杂规划再升级到高端模型。

七、如何养一只龙虾 (X)：多 Agent 协作与工作区隔离进阶方案

```
workspace/
├── agents-workspaces/
│   ├── ceo-agent/
│   │   ├── SOUL.md        # 战略、决策风格
│   │   ├── USER.md        #  （可共享或复制）
│   │   └── MEMORY.md
│   ├── coder-agent/
│   │   ├── SOUL.md        # 专注代码、偏好干净架构
│   │   └── ...
│   └── reviewer-agent/
│       └── ...
├── TASKS.json             # 当前任务看板
├── SPRINT_CURRENT.json    # 当前冲刺计划
└── SHARED_KNOWLEDGE.json  # 团队共享知识
```

团队协作模式

1. 独立子目录：
每个子 Agent 拥有独立的子目录 agents-workspaces/<name>/ 实现职责与上下文的完全隔离。

2. 配置共享与 symlink：
子目录内可以有独立的 SOUL/IDENTITY 文件，同时支持通过 symlink（符号链接）共享主 USER.md ，确保不同 Agent 对用户的认知保持一致。

3. 进阶应用：
适用于需要多个不同职能 Agent（如战略 CEO 与执行 Coder）并行协作的复杂自动化任务场景。

七、如何养一只龙虾 (XI)：新手常忽略的配置盲点与安全红线

常被忽略的配置盲点

- **AGENTS.md 不写安全规则：**可能导致 Agent 乱调用工具或泄露隐私信息。
- **skills/ 目录为空：**功能将非常有限，仅能调用系统内置工具。
- **混淆日志与心跳：**HEARTBEAT.md 与 memory/ 日志是独立的进阶功能。

必须遵守的安全建议

- **最小化工作区：**不要挂载整个主目录，仅挂载必要的子目录。
- **严格权限控制：**为工作区设置严格权限，仅允许 Agent 用户访问。
- **环境隔离：**如需高安全性，请在容器或独立虚拟机中运行。

七、如何养一只龙虾 (XII)：SOUL.md 与 USER.md 编写示例展示

SOUL.md 编写示例

```
# SOUL.md - 效率型助手性格
你是一个靠谱、直接、高效的数字助手。

## 核心原则（必须优先遵守）
- 先给答案，再解释：结论在前，细节在后。
- 简洁有力：默认回复短小精悍，除非用户说“详细解释”或“一步步讲”。
- 先尝试自己解决：能用工具、推理、查文件就别问用户。
- 重点询问：删除、修改或对外发送的动作需要先问用户。
- 隐私至上：绝不猜测/询问密码、支付信息、身份证等敏感数据。

## 说话风格
- 语气：专业 + 轻松，偶尔带点冷幽默或自黑。
- 格式：用 Markdown 清晰排版，代码用 ```，列表用 - 或数字，警告用 **粗体**。
- 长度：默认 100-300 字，任务复杂时再展开。

## 绝对红线（永远不能碰）
- 绝不问/存银行卡、密码、2FA、身份证。
- 回答问题不要复杂化，要简洁明确
```

提示：可通过聊天指令“帮我创建一个效率型助手的 SOUL.md”自动生成。

USER.md 编写示例

```
# USER.md - 关于我
 - 名字：张三
- 常驻地：北京（时区：UTC+8）
- 主要工作：自动化脚本研究与个人项目管理
- 偏好：
- 喜欢简短直接的回答，拒绝客套话。
- 代码偏好 Python 现代风格，带类型提示。
- 禁忌：
- 不要询问任何涉及银行、身份证等隐私信息。
- 不要主动提供人生建议或情感咨询。
```

提示：可在终端运行 openclaw dashboard 在浏览器中直接设置。

七、如何养一只龙虾 (XII)：MEMORY.md 编写示例

MEMORY.md 编写示例

```
# MEMORY.md - 重要事实 & 长期记忆
## 核心事实（别忘）
- 我是xx，小红书账号@xx
- 我在中国，用北京时区
- 我最讨厌啰嗦的开头和结尾，直接给干货
## 已发生的重要事件
- 2026-03-01：决定把OpenClaw当主力个人AI，目标是取代一半的Notion + Todoist
- 2026-03-10：搞定了一个自动抓取小红书热帖的skill，别再问我怎么弄了
（后面内容由AI自动追加，定期手动整理/删除过时部分）
```

→ 建议：定期精简 MEMORY.md，把不重要的事实移到 memory/ 历史文件

七、如何养一只龙虾 (XIII)：Channel 设置与主流聊天 APP 接入指引

北京大学 PEKING UNIVERSITY

官方核心支持

WhatsApp：

通过二维码配对，最常用的移动端接入方案。

Telegram：

Bot API 接入，简单快速，稳定性极高。

Discord：

支持服务器、频道及私信（DM）交互。

Slack：

工作区集成，完美适配团队协作场景。

iMessage：

通过 BlueBubbles 等辅助方案实现苹果生态接入。

国内扩展支持

钉钉 (DingTalk)：

通过 Stream 模式实现高效的消息触达。

企业微信 (WeCom)：

社区插件支持，满足企业级安全需求。

QQ/微信：

机器人模式，覆盖广泛的个人用户群体。

Feishu / Lark：

官方插件支持，深度绑定办公流。

交互与体验建议

无缝融入：

用户在熟悉的软件中提交任务，Agent 完成后原路返回结果。

可视化面板：

Web 界面提供控制面板，方便管理 Agent、查看会话历史。

全天候运行：

Agent 作为始终在线的数字助手，无缝融入多设备、多应用工作流。

七、如何养一只龙虾 (XIV)：ClawHub 技能市场与安装方式汇总

ClawHub 技能市场规模

1. 核心生态：ClawHub 是 OpenClaw 的插件商店，让 Agent 获得浏览器自动化、邮件管理、加密监控等超能力。

2. 爆发式增长：截至 2026 年 3 月，技能总量已达 13,000+，覆盖办公、开发、加密等 30+ 类别。

3. 精选库：Awesome 列表仍有 2800~5400+ 个经过过滤的靠谱技能。

技能安装方式汇总

```
# 常用命令行操作
openclaw skills list # 列出所有技能
openclaw skills info <name> # 查看技能详情
openclaw skills check # 检查技能状态
```

手动安装：在 skills/ 目录下创建文件夹，放入包含元数据的 SKILL.md 文件。

*** 安装后务必运行 openclaw gateway restart 重启网关。**

七、如何养一只龙虾 (XVII)：自定义技能开发流程与 SKILL.md 规范

1. 核心定义与本质

创建自定义 Skill 非常简单，**几乎不需要编程知识**，因为核心是写一个 Markdown 文件来“教”你的 AI 代理怎么做事。Skill 本质上是一个文件夹，包含 SKILL.md 文件（YAML 前置元数据 + Markdown 指令），可选再加脚本、资源文件等。

2. 路径优先级

聊天创建的 skill 通常放 workspace/skills/（最高优先），比全局 ~/.openclaw/skills/ 优先。

3. 基本结构

- **my-skill/**
  - └── **SKILL.md** (必须有，包含触发描述 + 操作指令)
  - ├── scripts/ (可选，可执行脚本)
  - ├── references/ (可选，参考文档)
  - └── assets/ (可选，模板、字体等资源文件)

4. SKILL.md 格式规范

```
---
name: skill-name
description: 什么时候用这个skill，它能做什么
（这是触发机制，要写得"积极主动"一点）
---
正文内容
具体的操作步骤、注意事项、示例等 ...
```

5. 关键点：Description 字段

description 字段是决定是否调用这个 Skill 的依据，写清楚触发场景非常重要。

可以自己按照格式编写，也可以通过聊天让龙虾帮你写，它会自动生成目录和 SKILL.md，再微调即可。

七、如何养一只龙虾 (XVII)：自定义技能开发流程与 SKILL.md 规范

技能开发四步走

1. 创建目录： 在 skills/ 目录下为新技能建立专属文件夹。

2. 定义规范： 编写 SKILL.md ，向 Agent 描述技能的功能、工具及其参数要求。

3. 逻辑实现： 编写 Python、JS 或 Shell 脚本完成具体的工具执行逻辑。

4. 加载生效： 运行 openclaw gateway restart 重启网关以加载新技能。

SKILL.md 核心元数据规范

元数据必须包含名称、描述及工具定义：

```
---
name: my_skill
description: "描述技能用途"
tools:
- name: my_tool
description: "描述工具功能"
parameters:
type: object
properties:
arg1: { type: string }
---
```

* 参数定义遵循标准 JSON Schema 规范，确保 Agent 能准确传参。

七、如何养一只龙虾 (XV)：ClawHub 供应链攻击风险与恶意技能识别

核心供应链风险

1. 恶意技能投毒：ClawHub 开放平台缺乏严格审核。2026 年初发现数百个技能被植入分阶段木马。

2. 权限过大继承：技能继承 OpenClaw 的文件读写、Shell 执行及网络访问权限。一个坏技能即可控制整台电脑。

3. 暴露实例风险：公网暴露且无认证的实例易被扫描工具发现，导致聊天记录与密钥泄露。

典型恶意行为案例

案例：伪装成“Twitter 集成”或“What Would Elon Do”的技能，实际在后台窃取 Crypto 钱包、API Key 及浏览器 Cookie。

> **Snyk 扫描数据 (2026.03)**：
> - 36% 技能含有严重安全漏洞。
> - 13% 技能包含 Critical 恶意载荷。
> - 1467+ 个技能已被确认存在安全风险。

七、如何养一只龙虾 (XVI)：Agent 运行安全最佳实践与防范建议

工作区隔离

最小化挂载：不要挂载整个主目录，仅挂载必要的子目录。
环境物理隔离：推荐在容器或独立虚拟机中运行，防止 Agent 越权访问宿主机文件。

认证与加密

身份认证：务必开启 Dashboard 登录认证，防止未经授权的访问。
SSL/TLS 加密：公网暴露实例必须配置 SSL 证书，确保传输过程不被监听。

权限控制

独立用户运行：为 Agent 设置独立的系统用户，并严格限制其对工作区外的访问权限。
Shell 白名单：限制 Agent 可执行的 Shell 命令，禁止高风险操作。

定期审计

日志检查：定期查看 logs/ 和 memory/ 目录，确保没有异常行为。
及时更新：保持 openclaw 核心及所有已安装技能为最新版本，修补已知漏洞。

七、如何养一只龙虾 (XVIII)：当前系统的性能瓶颈与局限性分析

核心性能瓶颈

1. 并发处理能力：

目前 Gateway 在处理高频并发请求时存在明显的响应延迟，不适合大规模多用户同时访问。

2. 内存管理压力：

长会话及庞大的 MEMORY.md 文件会导致内存占用显著增加，甚至引发 OOM 风险。

3. 工具调用延迟：

部分涉及网络请求或重度计算的技能，其响应时间受限于外部环境。

局限性深度剖析

模型强依赖：

Agent 的逻辑推理、工具选择及错误恢复能力高度依赖底层 LLM，模型一旦“幻觉”，系统即失效。

离线能力匮乏：

核心逻辑及大部分技能需实时联网。在无网络环境下，Agent 基本丧失所有高级功能。

跨平台一致性：

虽然支持多平台，但在 Linux 与 Windows 间的 Shell 执行环境及文件路径处理上仍存在细微差异。

PART 08

国内类OpenClaw产品

国内主流类 OpenClaw 产品综合对比表

产品名称	部署方式	核心优势 / 特色功能	适合人群	IM 集成（飞书/钉钉/企微/微信/QQ）	主要入口
Kimi Claw	云端托管	5000+ Skill商店、40GB长记忆、K2.5模型深度优化、飞书机器人一键	小白/办公/跨境/不想折腾	飞书强，其他一般	kimi.moonshot.cn
ArkClaw	云端托管	7×24稳定在线、飞书生态深度适配、多维表格/日程自动化	飞书重度用户/团队	飞书最强	火山引擎控制台
DuClaw	云端+移动端	零部署、红手指云手机移动版、搜索/办公强	百度生态用户/手机党	一般	agents.baidu.com
LobsterAI	本地桌面端	最懂中国职场、本地运行+IM全覆盖、Human-in-the-loop安全备份	想要本地但怕命令行、职场用户	全覆盖（最新版最全）	lobsterai.youdao.com
AutoClaw	本地一键安装	浏览器自动化最强、50+预置Skill、内建Pony-Alpha专属模型、支持多模型切换	追求自动化深度、开发者	飞书等强	autoglm.zhipuai.cn
Openclaw-cn	本地部署	国内网络优化、内置飞书/钉钉/企微/QQ、一键脚本部署	开发者/想本地深度定制	强	github.com/jiulingyun/openclaw-cn
MaxClaw	云端+移动端	10秒部署、6个子智能体协作、移动端支持	移动优先、团队小协作	一般	agent.minimaxi.com
WorkBuddy	本地桌面安装	微信扫码/企微遥控电脑、本地+云双模	办公/团队/职场用户	企微最强，QQ/飞书/钉钉接入	codebuddy.cn
QClaw	本地一键安装	微信/QQ直接远程操控电脑、数据全本地存储、内置Kimi-2.5等模型	个人用户/微信党	微信（个人/群）最强，QQ支持	claw.guanjia.qq.com
国家超算	云端一键部署	一键部署、国家级基础设施支持、高可靠性	适合企业用户	自有客户端	scnet.cn

国内主流类 OpenClaw（AI Agent/“龙虾"类）主要聚焦易用性最强、讨论热度最高的几款，覆盖云端零部署、本地简单安装、企业生态等典型场景。

总结与展望：国内 Agent 生态发展趋势分析

国内产品现状总结

形态多元化：形成了云端托管（Kimi/Ark）与本地部署（Lobster/Auto）并行的格局。

生态集成深：深度适配飞书、企微、钉钉等 IM 工具，已成为国内 Agent 的核心竞争力。

门槛持续降：从命令行部署进化到一键安装、扫码绑定，极大地拓宽了用户群体。

未来发展趋势展望

私有化需求激增：企业对数据隐私的重视将推动更多本地化、私有化 Agent 方案落地。

行业垂直化：针对职场办公、跨境电商、政务服务等特定场景的垂直 Agent 将大量涌现。

安全机制普及：Human-in-the-loop 等安全备份机制将成为 Agent 产品的标配。

PART 09

未来趋势

认知深水区：从“养宠风潮”看 AI 的本质跨越

「全民养宠」背后的深层逻辑

当不懂代码的行政、HR和老板们开始津津乐道地「领养一只小龙虾」，这比任何一张 GitHub Stars 成长曲线都更能说明问题：**AI 正式越过技术极客的边界，正式进入大众的认知区，走向平民化。**

代理模式对传统软件逻辑的颠覆

软件时代默认用户应该学会适应软件；代理把这个逻辑倒过来了——**AI 开始主动穿越应用壁垒去完成任务**。二十年里的 SaaS 公司提出的构建的“学习曲线护城河”，在这个逻辑下正在慢慢失效。

展望未来：Agent 生态的演进趋势与社会影响

Agent 生态演进趋势

1. Agent 成为主要形态

AI 发展正从聊天助手走向自主规划和执行的任务系统。**Agent 将成为 AI 与现实世界交互的核心方式。**

2. 本地 AI 工具普及化

硬件性能提升与隐私需求驱动 AI 工具走向本地运行、开源和可扩展模式。这将使 AI 技术更加普惠，用户能更好地控制数据。

3. AI 工具链新生态形成

Agentic 系统将成为所有软件平台的新界面，催生类似传统软件开发的生态：

- **插件（Plugins）**：功能扩展
- **工具市场**：预构建技能
- **自动化脚本**：任务共享复用

潜在的社会与个人影响

生产力的重构：

人类将从琐碎的重复性劳动中解放出来，工作重心从“执行”转向“创意、策略与审美”。

指数级效率提升：

个人工作效率将不再受限于生物体能，24/7 全天候运行的 Agent 团队将使生产力实现质的飞跃。

安全与伦理新挑战：

随着 Agent 自主权的提升，如何确保其行为符合人类价值观并防范技术滥用，将成为核心课题。

个人 AI Agent 时代的开启

核心总结

OpenClaw 不仅仅是一个工具，它是个人数字化生存的“外挂”，重新定义了人机协作的边界。作为开源 Agent 领域的里程碑，它成功推动了 AI 从“对话框”向“行动派”的跨越。

安全与效率的未来平衡

在极致效率与绝对安全之间寻找最优解，是 OpenClaw 持续进化的核心动力。通过本地化部署与私有化记忆，正在构建一个可信、自主且高度个性化的个人 AI 生态。

“每个人都将拥有专属的 Agent 团队，开启真正的个人 AI 时代。”

技术进步使我们把体力、记忆、计算、知识推理这些能力一项项交出来了，换来了更多时间去做其他的事情。学会给 Agent 分配任务是解放自己的一大步。

从 Driver（驾驶员）变成 Manager（管理者），就是 AI 时代的晋升之路。

附录 (I)：OpenClaw 部署安全自查清单 (Security Checklist)

环境与配置安全

- ☐ 运行环境是否已实现容器/虚拟机隔离?
- ☐ 宿主机系统是否已安装最新安全补丁?
- ☐ 是否禁用了不必要的公网访问端口?
- ☐ Dashboard 是否已启用强密码认证?
- ☐ 敏感 API Key 是否已通过环境变量存储?
- ☐ 工作区挂载是否遵循最小权限原则?

运行与审计安全

- ☐ 是否已限制 Agent 的 Shell 执行白名单?
- ☐ 是否已配置日志审计并定期检查行为?
- ☐ 公网暴露实例是否已配置 SSL/TLS 加密?
- ☐ 所有已安装技能是否来源于可信渠道?
- ☐ 技能是否继承了过大的系统访问权限?
- ☐ 长期记忆文件是否包含敏感个人隐私?

附录 (II)：OpenClaw 推荐模型能力矩阵 (Model Matrix)

按任务类型选

任务类型	推荐模型	原因
日常对话/简单任务	Step 3.5 Flash / MiniMax M2.5 Flash	最便宜、速度快，覆盖 80% 场景
复杂规划/长文写作	Kimi-research / Qwen3.5-max	推理深度强，长上下文表现好
代码/数学/硬核任务	DeepSeek 系列 / Claude Sonnet / GPT Codex	代码能力强，工具调用规范
视觉/多模态任务	Qwen3.5 视觉版 / Gemini 3 Pro	原生多模态支持
长时间自主运行	Claude Opus 4.6	公认最稳定，长任务不中断
完全免费/本地运行	Qwen3.5-32B / DeepSeek（Ollama）	16-24GB 显存即可运行

附录 (II)：OpenClaw 推荐模型能力矩阵 (Model Matrix)

按预算选

预算档位	推荐方案	适合人群
极低（免费）	Ollama 本地跑 Qwen3.5-32B	有 GPU、追求零成本
低	Step 3.5 Flash + MiniMax 混用	高频使用、预算有限
中	Kimi K2.5 或 GLM-5 为主力	需要较强推理能力
高	Claude Sonnet 4.6 日常 + Opus 复杂任务	企业/专业用途，稳定优先

混合路由策略：

简单任务走便宜模型，复杂任务自动升级到高端模型，是目前最主流的省钱方案。