

我把企业级智能体集群，真正部署进了一家连锁品牌

作者：大军（玉衡计划发起人、总架构师） 版本：v1.2 | 2026-08-09（修订：技术选型演进史补全）

先说我做了什么

不是demo，不是PPT，不是"我们正在尝试"。

魔方小镇城市书房——一家书店+咖啡复合业态的连锁品牌，几十城上百家门店——从2026年5月开始，正式运行着一套由20多个AI智能体组成的企业级智能体集群。

财务部有它的智能体，每天自动拉取收银数据、出经营日报、发到部门群。运营部有它的智能体，每天汇总门店数据、标红异常、生成报表。图书部、市场开发部、法务部、工程部.....每个部门都有自己专属的AI助理。而总控智能体在背后统一调度，每晚雷打不动地阅读各部门工作总结，学习、沉淀、汇报。

这套体系，我叫它**玉衡计划**。

玉衡这个名字怎么来的

北斗七星：天枢、天璇、天玑、天权、**玉衡**、开阳、摇光。

玉衡居于正中，是北斗的枢纽——主执掌、统筹、调度、权衡。

企业里最缺的不是"一个AI机器人"，而是一个**能统筹所有AI节点的中枢**。玉衡计划，就是这个中枢。

缘起：从一台电脑开始（2025.11 — 2026.03）

2025年11月，智能体概念开始出圈。我开始关注。

2026年1月，我搭出了第一个智能体。然后因为电脑配置不够，搁置了。

2026年3月，想法逐渐成形，“玉衡计划”这个名字诞生。

2026年3月28日，新设备到位——一台高配苹果工作站。云端+本地双模型部署，玉衡计划正式落地。

说实话，我不是程序员出身。

从3月28日到4月初，智能体频繁崩溃。当时我部署的第一套主力框架是 **OpenClaw**——它有很强的自动化能力和社区生态，但在我的企业级复杂环境里极不稳定：进程守护反复掉线、上下文串群、配置变更后频繁报错。没有团队，没有外援，全靠自己一个一个坑地趟。那段时间的体验就是：报错、搜索、修复、再报错。

好在我坚持下来了。

转折：Hermes入驻（2026.04）

2026年4月4日，Hermes Agent入驻，成了整个体系的主核心。

它有多稳？稳定到此后几乎没让我操过心。对比之下，OpenClaw在企业级环境里的稳定性短板暴露无遗——我逐步褪去了OpenClaw，把主力链路全部迁到Hermes上。

之后我又陆续尝试过国内外多款顶尖智能体——**Claude Code**、**Codex**、**WorkBuddy**等等，各有长处，但面对企业级智能体的复杂环境要求（多进程守护、物理隔离、长期稳定、技能体系沉淀），最终还是主打Hermes的环境。

到这个时候，我的智能体矩阵定型为： – **1个主力框架**：Hermes（企业级主核心） – **编程后端分体**：Codex / OpenCode / Claude Code（写代码、生成文件、做图，各司其职） – **底层多模型路由**：DeepSeek负责日常对话、小米MiMo负责识图OCR、Qwen和Gemma跑本地推理

而整个技能库，从零攒到了**200多个技能**——日报、数据提取、合同审核、图片识别、PDF处理、浏览器自动化……每个技能都是实际需求逼出来的。

关键一步：植入真实企业（2026.05.17）

自己玩明白了，就该上真战场了。

2026年5月17日，玉衡计划正式接入**魔方小镇城市书房**，多部门公测启动。

这是整个项目最关键的转折——从"我自己的智能体玩具"变成"企业的生产工具"。

真实场景里，问题立刻暴露了。

第一个大坑：消息串群。

最初的架构是一个企微机器人接所有部门群，由Gateway在内部做消息路由。听起来很优雅，实际跑起来：

- 运营部要的数据，发到了财务群
- 财务群的报表，回复到了市场部群
- 群消息偶尔还会跑到私聊里

每次修复都是打补丁，修好了过几天又犯。

根因分析：同一个AI智能体同时处理多个群的上下文，上下文窗口会混入不属于当前群的信息。

打个比方：一个服务员同时伺候十桌客人，就算记性再好，上错菜也是迟早的事。

架构升级：物理隔离（2026.06）

找到根因，方案就清晰了——物理隔离。

一个企微Bot只服务一个部门群，各管各的，永不交叉。

就像每桌配一个专属服务员，永不上错菜。

升级后的架构：

维度	1.0（之前）	2.0（现在）
Bot数量	1个	1个总控 + 多个部门独立智能体
隔离方式	逻辑隔离（沙箱）	物理隔离（独立进程）
消息路由	需智能路由，易出错	不需要路由，天然正确
串群风险	存在，反复发生	零，物理不可能
故障影响	一个进程挂了全停	一个挂了不影响其他

每个部门的智能体，知识范围**严格垂直**： – 运营部助理只知道运营的事 – 财务部助理只知道财务的事 – 图书部助理只知道图书的事 – 市场部助理只知道市场的事 – 法务部助理只知道法务的事

绝不混入其他部门的知识。

每日闭环：让AI自己沉淀

这套体系最让我自豪的设计，是**每日闭环**：

23:55 各部门智能体各自生成当天详细工作总结 → 归档 + 发到自己部门的群 **23:57** 总控智能体检查归档、逐字阅读每一份总结、学习各部门业务流程、存入永久记忆 **23:59** 总控向管理层汇报：归档情况 + 各部门今日核心动态

你可能好奇：总控不在任何部门群里，它怎么知道各部门发生了什么？

答案就是这每天雷打不动的工作总结。

所以"详细度"是铁律，不是建议——总结不详细，导致总控无法了解部门情况，视为工作失职。

这套闭环跑下来，管理层每天睡前都能掌握全公司各部门的动态。而总控智能体，每天都在变聪明。

我踩过的坑（真实经验，全是学费换的）

1. **敏感信息写入**：写入工具会自动检测并遮蔽含敏感标识的字符串，得用编码方式绕过
2. **运行环境路径**：系统自带Python缺模块，进程守护配置必须用完整路径
3. **认证失败**：凭据复制有误导致认证失败，得重新从后台复制
4. **API路径问题**：大模型API必须带版本路径，否则返回401
5. **配置缓存**：框架有last-good缓存机制，改配置后必须清理缓存再完整重启
6. **静默吞异常**：`except: pass` 会吞掉所有错误，导致问题完全不可见。必须改成显式日志
7. **克隆复制错误**：从其他实例克隆时容易漏改变量名，需要逐行检查

快速部署一个新部门智能体的流程（9步）： 1. 在渠道后台创建新Bot，获取ID和凭据 2. 写入环境变量文件 3. 从已有实例复制脚本，替换关键字段 4. 替换system prompt（只写本部门知识） 5. 语法检查 6. 启动测试，确认认证OK 7. 创建进程守护配置并加载保活 8. 逐行检查system prompt，确认无其他部门知识混入 9. 拉Bot进群测试

我坚持的几条原则

- 够用就上：不等人、不组建团队，边跑边迭代
 - 先通再扩：先搭通链路，再一个个对接部门
 - 部门自助：各部门用各部门的智能体，不设专人客服
 - 只管链路：总指挥只保证网关/模型/渠道正常
 - 物理隔离：内部智能体和对外智能体分机器部署，对外机不存任何内部数据
 - 代码沙箱：智能体自研技能不能自动生效，须人工Review后才能加载
 - Token配额：各部门月度预算上限，超限自动告警
-

给想落地的人的建议

先选一个业务痛点最重的部门试点。

我当时首选财务日报——数据规则清晰，容易验证，跑通了3个月稳定，再铺开。

不要一上来搞全套，容易散。

还有一句很重要的提醒：**是否让内部员工全权负责智能体系统的部署与运维？务必谨慎。**

- 完整掌握整套部署、排障、迭代运维的培训成本极高
- 掌握核心技术的员工在市场上抢手，跳槽意愿强
- 核心运维人员离职后，系统运维直接断层

建议初期由外部专业团队搭建和运维，内部配合学习。系统稳定后，再评估是否培养内部运维人员。

写在最后

从2025年11月关注智能体，到2026年5月17日魔方小镇城市书房多部门公测，再到6月完成2.0架构升级——半年时间，从零到一套真实运行的企业级智能体集群。

这套体系不是PPT，不是demo，而是每天23:55雷打不动自动跑起来的真实生产系统。

而我现在把这些写出来，是想告诉大家两件事：

第一，企业级智能体集群，不是大厂专属。 一个连锁品牌、一个非程序员出身的人，也能把它真正落地。

第二，架构的每一步演进，都来自真实场景的痛点。 串群、故障、吞异常.....这些坑我都替你趟过了。

如果你也想在自己的企业里做同样的事，欢迎交流。