

# 赵彦兵 | AI Agent / LLM 应用开发实习生

手机：134 6893 4893 | 邮箱：lorasys@outlook.com | 西安 · 2027 届 · 可立即入职 · 每周 5 天 / 连续 3 个月  
个人博客（作品集）：[lora-sys.github.io/loraSys](https://lora-sys.github.io/loraSys) | GitHub：[github.com/lora-sys](https://github.com/lora-sys)

## 教育经历

西安明德理工学院 计算机科学与技术 | 本科 | 2023.07—2027.07  
成都 Monad Blitz 黑客松一等奖（2024）| 蓝桥杯 C++ 算法竞赛省级二等奖（2024）| ACM 算法竞赛铜川邀请赛铜牌（2024）

## 专业技能

### 开发技术

- 使用 TypeScript、React/Next.js 开发实时页面和看板，把服务端事件投影为可查看的运行状态。
- 使用 Go/Hono、Python/FastAPI 和 PostgreSQL/pgvector 编写服务、接口与持久化逻辑，拆分前端、后端和 Agent 的职责。
- 使用 SSE、WebSocket 处理实时推送和连接恢复，遇到重复事件、取消和终态时补充状态保护。
- 使用 Docker Compose、GitHub Actions、Vitest、Playwright、Typecheck、Lint、Build 组织本地启动、持续检查和端到端回归。

### AI 应用开发

- 使用 LangGraph 编排 observe、think、act 等状态；用事件记录保存 Agent 的输入、动作、结果和后续回放所需数据。
- 在多 Agent 场景中用共享事件契约、A2A 消息和阶段状态控制协作顺序，避免模型自行跳过流程或改写结果。
- 将证据、引用、风险审查、人工审批和导出规则放在模型输出之后，区分“模型建议”和“可以对外使用的结论”。
- 对模型超时、证据不足、重复流和运行时阻塞保留降级路径与问题记录，不用固定数据伪装真实执行。

## 工作经历

凌云光技术股份有限公司 | 产品开发部门实习生 | 北京 2025.08—2025.09

负责项目：产品上线前验证与方案材料支持

- 围绕产品上线前的验证任务完成对照实验、参数筛选和可行性分析，整理实验数据与技术判断。
- 输出技术白皮书和 Demo，将验证结论转成验收和方案材料，供项目相关方评估。

ACAM Lab | 软件部门负责人

2024.07—2025.10

负责项目：复合材料质量在线监测系统的软件建设

- 统筹软件模块拆分、开发规范和代码协作，明确数据采集、接口联调和分析模块的责任边界。
- 负责数据采集、跨模块接口联调和初步分析模块开发，处理多人并行开发中的接口衔接问题。

## 项目经历

TrustOps（企业可信 Agent 应用） 2026.08

GitHub 开源：[github.com/lora-sys/trustops](https://github.com/lora-sys/trustops)

项目介绍：面向成长型 B2B SaaS 的安全问卷场景。系统处理政策、当前控制、历史客户承诺和模型建议混在一起后，回答无法解释、也无法安全对外发送的问题。

主要工作：

- 设计 Question → Claim → Evidence → Control Snapshot → Answer Version → Risk Review 的数据链。每个回答都能回到证据、控制快照和审查结果，而不是只保留一段模型文本。
- 实现 Evidence Room、Mission Control、Approval Inbox、Verified Export 四个界面，分别处理证据查看、任务推进、人工审批和最终导出。
- 接入 5 个独立 Agent、版本化证据、持久化任务和 Trace。Agent 可以生成候选结论，但不能直接把结论变成客户承诺。
- 将 Human Approval 和确定性 Export Gate 放在导出前。Reviewer 失败时拒绝导出，Snapshot、Answer、Approval、Manifest 保持不可变，方便复查。
- 后端直接测试为 38 passed。正式五 Agent LIVE\_AI Golden Run 受官方 AgentTeams runtime 阻塞，当前保留待补测状态，没有把 Mock、Test 或 Replay 写成真实运行。

Agent Arena（证据驱动多 Agent 评测） 2026.07

GitHub 开源：[github.com/lora-sys/AgentArena](https://github.com/lora-sys/AgentArena)

项目介绍：面向同一任务下多 Agent 的评测场景。系统要回答的不只是“谁赢了”，还包括为什么得分、经历了什么攻击和修订、刷新后能否复现整个过程。

主要工作：

- 构建 Hono API、Battle Engine、共享事件契约和 React Live Arena。Battle Engine 固定简报→提案→构建→攻击→防守→验证→裁决的顺序，不让模型跳步骤或直接指定冠军。
- 让三支 Agent 团队各自保存提案、修订、补丁、测试和 Mini App。每个 Score 至少绑定一个 evidenceEventId，评分可以追查到具体事件。
- 使用 Zod 校验后再写入 Battle、Artifact、证据和 Passport 事件；Champion 页面和团队护照从持久化事件重新生成，刷新不改写历史事实。
- 处理 SSE 重复分发、取消流和终态保护。外部模型超时或证据不足时进入 Live AI Degraded，不用固定结果冒充真实 Battle。
- 358 项单元/集成测试通过，Playwright 12 passed、2 条按 viewport 跳过；完成 3 场真实 StepFun Battle，均到达 Artifact、Champion 和 Passport。当前为单机黑客松 MVP。

EcoMatrix（A2A 多 Agent 经济沙盒） 2026.07

GitHub 开源：[github.com/lora-sys/ecomatrix](https://github.com/lora-sys/ecomatrix)

项目介绍：面向拥有不同角色、余额和生存目标的 Agent。系统让它们发布需求、协商和交易，同时要处理并发扣款、重复消息、决策过程不可见和前端实时连接不稳定的问题。

主要工作：

- 完成三段交付：Go + PostgreSQL Physical Engine 负责交易状态，Python LangGraph Brain 负责决策，Next.js God's-Eye Dashboard 负责观察，避免把交易、推理和页面状态混在一层。
- 交易接口使用 SELECT ... FOR UPDATE 锁定余额，用 msg\_id 做幂等。重复交易请求不会再次扣款，50 goroutine 并发场景下余额不为负。
- 实现 A2A v1.1 codec 和 EXECUTE\_TRADE、POST\_FEED 两类动作；Python 端与 Go codec 做 parity 测试，状态变更接口使用 per-agent HMAC 和重放窗口。
- 用 LangGraph 的 observe → think → act 状态机驱动不同工作类型；以 world-GOLD conservation invariant 检查交易后总量不变；Dashboard 用 WebSocket backpressure、heartbeat 和重连展示运行过程。
- 39 个 Go-race 测试、23 个 Python 测试、4/4 Playwright E2E 通过。HMAC rotation、按角色过滤 WebSocket 订阅等功能仍列为后续项。

## 其他作品

|                               |                                                                                            |                                                         |
|-------------------------------|--------------------------------------------------------------------------------------------|---------------------------------------------------------|
| 个人博客（作品集）                     | 展示个人项目、GitHub、正在做的方向和可访问作品；文章记录构建过程和技术判断。                                                  | <a href="https://lora-sys.github.io/loraSys">GitHub</a> |
| AI Engineering Harness（Skill） | 用 18 个 Agent 和 9 条工作流把 Issue、Worktree、Plan、Build、Review、Evidence、Merge、Memory 串起来，保留可审查记录。 | <a href="https://lora-sys.github.io/loraSys">GitHub</a> |
| JobLinker                     | 完成岗位检索、匹配解释、PDF 简历解析、求职材料生成和自动申请两阶段流程，面向求职场景整理信息和动作。                                       | <a href="https://lora-sys.github.io/loraSys">GitHub</a> |
| nano-vLLM Interactive Guide   | 以 13 个 HTML 互动实验和 13 章中文教程拆解 LLM 推理引擎，GitHub 记录 23 Stars。                                  | <a href="https://lora-sys.github.io/loraSys">GitHub</a> |
| Trading Pi OS                 | 把市场数据、仓位跟踪、回测和执行收在一个 TypeScript 个人交易系统，便于查看和复盘操作。                                          | <a href="https://lora-sys.github.io/loraSys">GitHub</a> |

## 个人优势

- 持续把想法做成可运行作品。主项目都保留代码、测试、演示或项目文档，遇到没完成的部分会写清原因和下一步。
- 有模块拆分、接口联调和代码协作经历。做多 Agent 系统时会先定状态、消息和责任边界，再处理页面和模型调用。
- 持续维护开源 AI 应用和学习产品。获得黑客松一等奖、蓝桥杯省二、ACM 铜牌；用比赛和公开作品检验工程实现。