

徐远军

求职方向: Agent 开发工程师 / LLM 应用开发工程师 / Agent 平台工程师

- 手机: 17786397797
- 邮箱: alex_xuyuanjun@163.com
- 所在地: 武汉市
- GitHub: <https://github.com/YuanjunXu>

个人简介

- 具备 Agent 系统开发能力, 熟悉从需求分析、Agent 编排、Tool Calling、业务工具接入到部署治理的完整开发链路。
- 熟悉任务规划、上下文管理、长期记忆、RAG、Workflow 和多 Agent 协作等核心机制, 能够独立完成复杂 Agent 应用的设计与落地。
- 具备 Agent 平台化设计能力, 掌握模型协议抽象、工具注册与调度、扩展机制、权限审批、状态持久化、多租户隔离和可观测性建设。
- 具备扎实的 .NET 后端开发能力, 熟悉 ASP.NET Core Web API、RESTful API、认证授权、关系型数据库、缓存及任务调度, 能够独立完成后端服务的设计、开发与交付。
- 具备后端架构设计能力, 能够围绕模块化、插件化、事件驱动、分布式部署和可插拔基础设施设计系统边界, 兼顾扩展性、稳定性、安全性与国产化适配需求。
- 注重系统稳定性与工程质量, 能够通过类型约束、自动化测试、行为 Eval、日志审计和发布门禁保障 Agent 系统的可维护性。

专业技能

Agent 与大模型开发

- 熟悉 OpenAI 兼容 Chat Completions API、Tool Calling、多轮工具调用及模型请求/响应协议。
- 熟悉 Prompt 管理、上下文预算、历史压缩、长期记忆、任务规划、后台任务和模型异常恢复。
- 熟悉 Skills、Pack、Workflow DAG、MCP、多 Agent 协作及 Agent Tool 设计。
- 掌握 RAG 基本链路, 具备 Embedding、向量存储、Top-K 检索、相关度阈值、索引同步及检索降级的实现经验。

.NET 后端与架构开发

- 熟练使用 C#、.NET 8、ASP.NET Core Web API 和 .NET Framework, 具备 DDD、Middleware、IoC/DI、JWT、RBAC 等后端开发经验。
- 熟悉基于反射与动态 DLL 加载的插件化架构, 以及微服务、Nginx 负载均衡、发布/订阅事件驱动等系统设计方案。
- 熟悉 PostgreSQL/PostGIS、Kingbase、GBase、MongoDB、Redis, 具备国产数据库及麒麟 OS、鲲鹏 ARM 环境适配部署经验。

- 熟悉 SqlSugar 多数据库适配、Evolve 数据库迁移、Redis/内存缓存、MongoDB/MinIO 对象存储及 Quartz.NET 任务调度。
- 熟悉 Serilog、HealthChecks、Swagger/ApiFox、Docker 等开发与运维工具，具备 Vue 2/Vue 3、Element Plus 前端协作经验。

Python 与平台开发

- 熟练使用 Python 3.10+，能够运用类型标注、Protocol、dataclass 和异步任务完成模块化系统设计。
- 熟悉 FastAPI、OpenAPI、JSON Schema、SQLite、PostgreSQL 及 RESTful API 开发。
- 能够设计 CLI、HTTP API、MCP Server、Worker 等多种系统接入方式。
- 熟悉插件化架构、配置管理、可插拔存储、身份认证、RBAC、限流和配额设计。

工程质量与可观测性

- 熟悉 pytest、coverage、Ruff、mypy，能够编写单元测试、集成测试和契约测试。
- 熟悉结构化日志、JSONL 审计、Metrics、OpenTelemetry Trace 等可观测性方案。
- 具备 Agent Eval、Benchmark、性能基线和发布质量门禁建设经验。

项目经历

Agent Core——通用智能体开发与运行平台

项目角色：核心开发 / 架构设计

项目时间：[2026.4]—[至今]

技术栈：Python、OpenAI SDK、FastAPI、MCP、JSON Schema、SQLite、PostgreSQL、OpenTelemetry、pytest

项目简介

面向 Agent 应用开发和平台化建设，设计并实现通用智能体框架。系统覆盖模型调用、任务规划、Tool Calling、上下文与记忆、业务工具接入、多 Agent 协作、安全治理、状态持久化和行为评估，可通过 CLI、HTTP API、MCP Server 等方式对外提供能力。

主要工作与成果

- 设计并实现 Agent Runtime 与多轮 Tool Calling 执行循环，统一模型请求、响应、Token Usage 和 Tool Call 数据结构，支持连续工具调用、执行结果回填、协议校验和最终答案生成。
- 实现任务规划、上下文压缩、持久化 Session 和长期记忆机制，根据上下文预算裁剪大型工具结果、摘要历史消息，并按相关性加载跨会话记忆。
- 建设业务工具接入体系，提供 Python `@tool` 装饰器、OpenAPI 导入和 MCP 三种接入方式，通过统一 ToolSpec 与 JSON Schema 屏蔽不同业务系统的调用差异。
- 设计 AgentExtension、Skills、Pack Manifest 和 Workflow DAG 扩展机制，将提示词、工具、知识和业务流程打包管理，支持不同业务场景按需装配。
- 实现多 Agent 协作与后台任务调度，支持 Lead Agent 拆分任务、启动持久化队友、异步通信、后台执行和结果通知注入。
- 构建统一工具执行平台，对工具进行自动发现、参数校验、权限过滤和执行分发；只读工具支持同批次并发，高风险及存在副作用的工具采用串行处理。
- 设计 `read/write/execute/network/dangerous` 五级工具权限模型，实现工具启停、执行前确认、外部审批、路径与域名策略、命令白名单及危险操作默认拒绝。

- 建设模型调用容错链路，支持输出截断续写、上下文超限压缩重试、429/529 指数退避、Retry-After 处理和 fallback model。
- 实现 Principal 身份映射、role/scope 授权、session/user/tenant 隔离，以及 Local、SQLite、PostgreSQL 可插拔状态后端。
- 接入 Prompt Injection 检测、敏感信息脱敏、JSONL 事件审计、Metrics 和 OpenTelemetry Trace，对模型请求、工具调用、 workflow 节点和异常链路进行追踪。
- 建立自动化测试、行为 Eval、Benchmark 和发布门禁；项目包含约 995 个测试函数，pytest 分支覆盖率门禁设置为 85%，覆盖 Tool Calling、权限、Session、Pack、Workflow、Connector 和存储后端等核心契约。

RAG 与长期记忆检索子系统

项目角色：核心开发

项目时间：[2026.4]—[至今]

项目关系：由 Agent Core 的知识与记忆模块抽取并独立演进，通过 Provider/Tool 接口接入 Agent Runtime

技术栈：Python、ChromaDB、向量检索、Embedding、pytest

项目简介

为 Agent 提供外部知识检索和跨会话记忆召回能力，抽象知识检索、文本向量化及向量存储边界，同时兼顾本地无依赖运行、持久化索引、结果预算和故障降级。

主要工作与成果

- 抽象 KnowledgeProvider、Embedder 和 VectorStore 协议，实现检索逻辑、向量生成与存储后端解耦，为后续接入不同 Embedding 模型和向量数据库预留稳定边界。
- 实现本地 Hashing Embedding，以及 InMemory、Chroma 两种向量存储后端，支持无外部模型服务运行和本地持久化索引。
- 基于内容签名同步长期记忆索引，数据未发生变化时复用已有索引，并处理记忆新增、更新和删除后的索引一致性。
- 实现 Top-K、最小相关度阈值、向量预检索、时间衰减和上下文字符预算，降低无关知识进入模型上下文造成的 Token 浪费与回答干扰。
- 设计向量检索失败降级机制，在向量后端异常时回退到模型选择或启发式检索，避免检索组件故障中断 Agent 主链路。
- 为知识检索结果提供来源、相关度和低信任标记，并与 Prompt Injection 检测链路配合，降低外部知识污染模型上下文的风险。
- 编写向量召回、索引替换、Chroma 持久化、配置解析、数据同步和异常降级测试，验证检索链路的正确性与可恢复性。

Web 应用服务器——插件化 .NET 后端框架

项目角色：核心开发 / 架构设计

项目时间：[2021]—[2025]

技术栈：C#、.NET 8、ASP.NET Core Web API、SqlSugar、PostgreSQL、Kingbase、GBase、Redis、MongoDB、MinIO、Quartz.NET、Nginx

项目简介

面向 Web 项目交付构建自托管式应用服务器框架，通过插件化配置、多数据库适配、缓存与存储抽象及国产化环境支持，为不同项目提供统一后端基础能力，已应用于 100+ 项目。

主要工作与成果

- 主导构建基于 ASP.NET Core Web API 的自托管应用服务器，通过反射和动态 DLL 加载实现功能模块插件化接入，使项目交付周期缩短 20%，系统扩展性提升 80%。
- 建设统一中间件体系，使用 Serilog 记录框架与业务日志，并通过 HealthChecks 监测数据库、Redis 和内存状态；同时支持 Web 站点、图片等静态资源托管。
- 设计 JWT、RBAC 和细粒度权限控制，采用 Access Token 与 Refresh Token 双 Token 机制，在满足等保相关安全要求的同时兼顾用户体验。
- 基于 SqlSugar 适配 PostgreSQL、Kingbase V8、GBase 8s 等数据库，并使用 Evolve 实现数据库自动迁移，将单次迁移由约 1 小时缩短至 3 分钟以内，错误率降低至 5% 以下。
- 通过发布/订阅机制解耦业务模块，结合 Quartz.NET 支持 HTTP、后台和定时任务调度；支持内存与 Redis 缓存切换，使平均响应时间缩短约 60%。
- 抽象本地文件、MongoDB 和 MinIO 资源存储后端，满足不同项目的文件与对象存储需求，并支持按配置切换实现。
- 使用 Nginx 进行网关路由和负载均衡，通过插件配置拆分微服务职责，将新服务扩展时间缩短至 0.5 天以内。
- 集成基于 Flowable 的审批流程服务，并结合二次开发平台复用通用能力，使中小型项目可在 1 个月内完成交付；某省级自然资源项目交付周期缩短 40%。

数据管理系统——插件化 WinForm 桌面应用

项目角色：核心开发

项目时间：[2019]—[2022]

技术栈：C#、.NET Framework、WinForm、DevExpress、反射、插件化开发

项目简介

面向数据管理及项目二次开发场景构建 WinForm 桌面应用，通过统一插件协议、可复用界面组件和动态布局能力支持不同项目快速定制。

主要工作与成果

- 设计包含名称、标识、权限和执行方法的统一插件接口规范，通过反射动态加载外部 DLL，实现业务功能按项目扩展。
- 使用 `AssemblyResolve` 处理插件依赖解析，降低不同插件及第三方程序集版本带来的加载问题。
- 封装可复用 TreeView 目录树控件，支持图标、层级、拖拽、勾选和点击事件自定义，减少项目定制开发中的重复工作。
- 实现 Dock 停靠窗口和用户自定义布局，并集成 DevExpress 组件库，提升复杂数据管理界面的可操作性和一致性。

工作经历

中地软件集团有限公司 | .NET 工程师

2018.07—至今 | 武汉

- 主导通用 Agent 框架研发，完成 Tool Calling、上下文与记忆、业务工具接入、Workflow、多 Agent 协作和安全治理等能力建设，支持 GIS 空间查询，系统问答 类业务场景接入，Agent 任务成功率达到 80%。
- 搭建团队内部 RAG 知识库和产品 API 文档库，完成知识整理、检索接入及使用流程建设，累计管理 200+ 份文档、服务 30 名研发人员，知识检索准确率提升 80%，跨角色沟通效率提升 70%。
- 建设标准化 Prompt 仓库并通过 Git 进行版本管理，将代项目部署、码解释、代码优化和接口文档等常用任务模板化；封装 10+ 个文档与研发类 Skills，使重复任务处理时间缩短 90%。
- 负责公司 Web 项目的后端基础架构和核心模块研发，围绕插件化、多数据库适配、缓存与对象存储、任务调度和认证授权沉淀通用能力，支撑 50+ 个项目交付，平均交付周期缩短 50%。
- 主导基于 .NET 8 和 ASP.NET Core Web API 的自托管应用服务器建设，通过反射与动态 DLL 加载实现模块插件化，系统扩展效率提升 90%。
- 设计 PostgreSQL、Kingbase、GBase 等多数据库适配方案，并通过 Evolve 实现数据库自动迁移，将单次迁移时间由 2小时 降低至 0.5小时，人工操作错误率降低 99%。
- 建设 JWT/RBAC 权限、双 Token 认证、结构化日志和健康检查体系，覆盖数据库、Redis、内存及关键服务状态，线上故障发现时间缩短 80%。
- 推进内存/Redis 缓存及本地文件、MongoDB、MinIO 存储后端统一抽象，使接口平均响应时间缩短至 3 秒，不同部署环境的适配周期缩短 50%。
- 参与国产化项目适配和交付，完成业务系统与麒麟 OS、鲲鹏 ARM、Kingbase/GBase 等软硬件环境兼容适配，支撑 3 个国产化项目落地。
- 负责插件化 WinForm 数据管理系统研发，设计统一插件接口、动态依赖解析、TreeView 组件和 Dock 布局能力，复用于 20+ 个桌面项目，定制功能开发工作量降低 60%。
- 与产品、前端、测试、实施及售前等 5 个角色/团队协作，参与工作量评估、需求评审、技术方案设计、核心功能开发和故障排查，推动 20+ 个项目按期交付。

教育背景

[湖北汽车工业学院] | [电子信息工程] | [本科]

2014.9—2018.6

- CET-4 证书