

王健俊-AI应用开发工程师

<https://github.com/passionworker> | 17820577270 | 17820577270@163.com

教育背景

深圳技术大学 本科 数据科学与大数据技术 2024.09-2028.07

- 曾获奖项：飞书AI先锋大赛百强 | 抖音黑客松二等奖 | 全国法律AI大赛八强 | 国赛数模省一 | 华数杯数模国一 | 正大杯省一

实习经历

蕉内（三立人科技有限公司） 数字研发部门 AI应用工程师 2026.5-至今

负责模块：围绕企业内部 AIGC 视觉生产与 Agent 平台建设，负责产品设计、权限管理及AI 数据链路的工程落地。

- AIGC 多模态视觉生产**：主导商品生图平台与 PS AI 插件开发，基于 **FastAPI + React + Adobe UXP** 搭建多模型生成/编辑链路，设计 **LookSpec 结构化理解 + DAG 任务编排 + BBox 自动定位/局部修复 + 质量自检重试** 的 Agent Pipeline，并将选区 Mask、多参考图编辑能力原生集成 PS，推动单人日产约 **10→30 张**、**单张成本下降 51%**。
- 企业 Agent Platform**：建设公司级 **AI Gateway + MCP Capability Layer**，统一承接模型、北极星 BI、知识库及业务 API 的接入与路由，设计身份透传、RBAC / Tool Scope 与调用治理；基于 **RAG + MCP + Skill / Agent 编排落地** AI 企划、供应链 QA Agent 等场景，沉淀可复用的企业 Agent Runtime 与能力接入体系。

深圳市麟远科技有限公司 AI应用工程师 2025.10 - 2026.01

负责模块：面向电商视觉供应链构建多模态 AI 素材生产工作台，打通竞品发现、视觉理解、Prompt 沉淀与图片生成链路。

- 基于 **FastAPI + Redis + WebSocket** 搭建异步生成 Pipeline，并通过 Adapter / Gateway 统一多模型，结合 Prompt / Style Library 与 **LLM-as-a-Judge** 评测优化生成链路，实现 **API 成本降低约 20%**，**人工审核 / 二改介入降低约 30%**。

项目经历

腾讯mini项目 | RepoMind Code Knowledge Base & Agent Tuning MCP Agent治理 Tree-sitter AST 2026.03- 至今

项目简介：面向大型代码仓库的 Coding Agent 代码理解与检索系统，参与 Benchmark 建设，并负责 Agent Trace 分析、MCP 调用治理与代码图谱优化。

- 代码图谱与增量索引**：设计基于 Tree-sitter 的本地图谱增量重建与新鲜度保障机制，以内容寻址 Snapshot、不可变 Generation、依赖闭包和 FreshnessGate 解决多工作树及源码变更后的 stale-index 问题；在 310 文件、3,855 符号的真实代码库上，将单文件更新由 **10.78s 降至 0.12s (约 90×)**，并验证增量图与全量构建结果一致。
- Agent 可观测与数据飞轮**：基于 496 个 Trace 重构分析方式，设计 **payload-neutral、可回放的 Observation Kernel**，采集 Tool Call / Evidence / Usage / Latency / Outcome 等结构化事件；结合 **Replay、Grounding Audit、A/B 与 Guardrail** 定位高频失败模式，并将有效策略版本化验证与回滚，形成数据驱动的 Agent 优化闭环。
- Benchmark & Eval**：构建多语言 Benchmark，将端到端任务拆解为 **Evidence Retrieval / Reasoning / Patch Validation** 并建立 Trace-level 归因；设计 **C/E/Q 连续评分 + Regression Penalty**，融合 Rule-based、LLM-as-a-Judge 与 DeepEval，并通过可复现 Runner、回归对比与失败分层定位能力瓶颈，推动跨文件分析 **F1 0.28→0.61**。

agent共享记忆总线 MCP BM25 Node.js 独立开发 2026.01- 至今

项目描述：独立开发面向 Claude Code、Codex 等 Coding Agent 的 **Local-first Shared Memory Infrastructure**。

- Memory Architecture**：设计 **L0 transient + L1-L5 hierarchical memory lifecycle**，以 **append-only JSONL + SQLite WAL** 持久化，并通过 Dream / Compaction、LSN + content_sha256 实现记忆晋升、压缩与跨 Agent 幂等同步。
- Retrieval & Reliability**：实现 **BM25 + Dense → weighted RRF → MMR → optional Cross-Encoder** 混合检索与按需 Context Loading；构建 Worker Pool、Circuit Breaker、Backpressure Queue，并将 MCP 拆分为服务实现故障隔离。

技能

- AI / Agent**：MCP | Skill | LangGraph | Agent Memory | Context Engineering | RAG / Hybrid Retrieval | RRF / Rerank | Tree-sitter / AST
- 模型与评测**：VLM | Structured Output | Image Generation / Editing | LLM-as-a-Judge | Benchmark / Eval
- 工程开发**：Python | TypeScript | FastAPI | React / Node.js | Redis | asyncio | Docker | Git