



赵兴鹏

27 应届
13540663554

2638265504@qq.com
https://github.com/ZhaoXingPeng

教育经历

太原理工大学	光电信息工程（电子信息类）· 硕士	2024/09 - 至今
- 主修课程：数字图像处理、机器学习与深度学习、光电传感原理、医学影像技术。 - 科研方向：聚焦深度学习在人体感知与医学影像场景的落地应用。		
华侨大学	计算机科学与技术· 本科	2020/09 - 2024/07
主修课程：数据结构与算法、计算机网络、操作系统、软件工程、数据库系统等。		

专业技能

- AI Agent 架构**：理解 Agent/ReAct/Multi-Agent 范式，掌握 RAG、Tool Calling 与上下文、记忆管理。
- AI 应用框架**：熟悉 LangChain/LangGraph、MCP 协议，能编排工具落地可审计、可回归的多轮交互任务。
- 嵌入式 / 语音**：熟悉 C++20、ESP-IDF 与 ESP32-S3，能打通 I2S 音频链路 with 唤醒交互，设备端存储与通信。
- Java 核心**：掌握集合、反射、多线程、JUC、线程池、JVM 类加载与 GC，了解 Java 17/21 与虚拟线程。
- Spring 生态**：熟悉 Spring Boot/MyBatis/Dubbo，理解 IOC、AOP、事务与 Bean 生命周期，能搭建分层服务。
- MySQL / SQL 优化**：熟悉 InnoDB、事务、MVCC、锁与索引，能结合执行计划做表设计与慢 SQL 优化。
- Redis / 缓存**：熟悉核心数据结构、淘汰与持久化主从，能治理穿透、击穿、雪崩问题。
- MQ / ES**：了解 RocketMQ 与 Elasticsearch，掌握消息确认、重试、幂等、顺序与死信队列。
- 微服务 / 分布式**：熟悉 Nacos、OpenFeign、Gateway、Sentinel，了解分布式锁、事务与链路追踪。
- 工程质量**：掌握策略、责任链、模板方法、代理等常用模式，具备分层设计、统一异常、日志定位与接口抽象能力。
- 前端开发**：熟悉 HTML5/CSS/JavaScript 与 Vue 3/Pinia/Vite，能组件化开发并完成接口联调。
- AI 编码提效**：熟练用 Claude Code、Codex、Cursor 完成需求拆解、代码生成、重构与测试补全。

工作经历

上海七牛信息技术有限公司	产品架构实习生	2026/07 - 至今
VoiceLife（声活）智能语音设备端系统 https://github.com/1024XEngineer/VoiceLife C++20 / ESP-IDF / ESP32-S3 / SQLite / I2S / WebSocket / MCP 背景：参与 VoiceLife（声活）智能语音日程设备的产品架构与可靠性建设，覆盖 ESP32-S3 端侧语音链路、日程服务与 MCP/IM 消息渠道的端到端闭环。		

- 状态机建模与可观测性**：将设备运行时、语音 Provider、日程与消息渠道抽象为可组合状态机与事件流，收敛异步边界；以队列水位、I2S 统计与结构化日志建时序化遥测基线，用数据定位连续对话中的跨层竞态与状态不一致。
- 实时音频背压与抖动预算**：从 I2S 消费与网络突发的背压传导入手，确认长回复 PCM 缺口源于有界队列准入策略而非业务逻辑；扩容抖动预算（12→16 帧、240→320 ms）后，8 轮实板帧拒绝 2→0、STT 匹配 8/8。
- 构建可复现性与验收门禁**：定位干净构建失败是 SQLite 组件为生成物却不入库、入口缺幂等兜底，改为缺失自动生成、存在即复用；分层验收门禁支撑 12 轮实板可复现，末轮 ASR 偏差如实计入边界、不标通过，作为回归基线。
- 跨入口结果语义归一**：将日程提醒接入统一交互队列，在 MCP 协议桥接层补齐标量/空返回的成功判定与幂等处理，修复“动作已提交、上层判定失败”的语义不一致，使语音/日程/IM 多入口状态一致且可观测。
- KWS 误触发归因与调优**：对“告别后难以再唤醒”先证伪“检测器未重启”假设，再按串口事件与状态机时序定位为 TTS 播放窗口内回声保护窗的误拦截；结论沉淀为实验记录，以唤醒词命中率与负样本误触发率驱动阈值调优。

项目经历

BabelFlux – AI 实时同声传译系统	独立设计与开发（全栈 / 后端为主）	2026/06 - 至今
https://github.com/ZhaoXingPeng/BabelFLUX-java 背景：面向演讲、技术分享和网课的实时双语传译系统（ www.babelflux.icu ）；负责把多源音频、实时字幕、在线纠偏、会后校正和报告导出收敛为可演示的 Web、桌面端与 Java 服务端闭环。 Java 21 / Spring Boot 3.4 / WebSocket / MySQL / Redis / RabbitMQ / Elasticsearch / Vue 3 / Tauri		

- 流式链路**：Java 21 + Spring Boot 承载 16 kHz PCM，以有界队列和背压承接突发；接入 LiveTranslate 流式同传，按段下发 partial/final 双语字幕与 TTS 音频，新句到达即中断旧播放、防串播。
- 双轨纠偏**：在线以低延迟模型跨句复核，即时修正前句术语/数字/否定错误；会后由强模型通读全场统一术语并生成摘要，按六大领域注入差异化提示词与术语表，超时慢批不丢弃已完成的纠偏、杜绝语义删减。
- 持久化边界**：MySQL 存会话事实与 outbox，Redis 管带 TTL 的 handoff，RabbitMQ 投递可重试事件，ES 提供可重建索引；设置幂等更新、调度时间戳取整归一，跨 JVM 报告最终化以行锁加幂等复用保证只生成一次。
- 多端闭环**：Web 工作台与 Tauri 悬浮窗经 deep link 接管同一字幕流；悬浮窗可图钉固定、二级确认关闭，字幕窗关闭即提示报告去向，报告统一沉淀 Web「报告历史」，避免跨端重复下载与旧会话串用。
- 端到端可用性**：三端单机可部署，模型未配置或中间件不可用即降级为实时译文与可下载报告，高精度处理不阻塞交付；纠偏状态与报告多端沉淀一致，可演示、可回看、可导出，演示站点可复现主流程。

DevBrief Agent – 可恢复的研发决策 Agent	独立设计与开发（全栈 / Agent 工程）	2026/09 - 至今
https://github.com/ZhaoXingPeng/DevBrief-Agent 背景：把研发评审、Bug Triage 与复盘会议的语音输入转化为可验证、可审批、可执行的研发任务，语音与转写是其差异化输入；以可恢复 Agent Harness 为主线，闭环证据检索、工具编排、审批写入与 trace 回放，而非通用会议纪要。 Python 3.11 / LangGraph / FastAPI / SQLite / PostgreSQL / OpenTelemetry / Vue 3		

- 执行控制**：以类型化状态机显式建模会话生命周期与可恢复失败路径；每会话绑定模型不可放宽的执行预算（步数/工具数/截止时间/token/成本），预算耗尽即停并持久化 checkpoint，停止原因与恢复点可审计。
- 恢复与幂等**：checkpoint 最小快照含计划哈希、审批 ID、幂等键与已完成工具；恢复前先核对回执或幂等键，外部写入绝不盲重放，未知写入结果先查询再决策，避免中断后重复建单。
- 工具治理与审批**：canonical 工具注册表声明 Schema、读写分级与审计要求；Policy+Approval 门禁绑定精确计划哈希，未批准的外部写不可执行，修改草稿即令旧批准失效，外部副作用全程可审计。
- 上下文与证据边界**：Context Builder 按来源分区系统规则、转写、证据与工具结果并做预算裁剪；GitHub 仓库/ADR/Issue 作为不可信数据仅作参考，不能改变策略与预算，从机制上隔离提示注入。
- 评测与可观测**：以版本化 fixture 与标注集跑离线 Eval，Replayer 重建会话并比对状态与计划哈希；trace 记录状态迁移、证据引用与审批回执，输出字段准确率、证据覆盖与失败分类，支撑回归与效果归因。

DBJavaGenix – 面向 LLM 的数据库逆向与 Java 工程生成 MCP 服务	独立设计与开发	2025/09 - 至今
https://github.com/ZhaoXingPeng/DBJavaGenix 背景：传统代码生成器难以让开发者在生成中检查和修改上下文，而一次性向 LLM 暴露全部工具会抬高启动成本；因此将数据库分析、语义命名、Java 代码生成和预览组织为可审计、可部署的 MCP 服务。 Python 3.11 / MCP / Skills / MySQL / PostgreSQL / SQLite / Docker / GitHub Actions		

- 三层架构**：Skill 显式编排五阶段工作流，MCP 拆出连接/元数据/图算法/渲染/可观测等 30+ 原子工具：Apps 以 ER 图、仪表盘、code-diff、结构树反馈中间结果，让 LLM 能在生成过程中检查并修改上下文。
- 渐进式发现**：将启动暴露由全量约 4,954 tokens 收敛为 6 个核心工具约 1,090 tokens（约 -78%），其余经 search_tools 按需加载，兼顾完整工作流与受限客户端一键生成。
- 可复现与一致性**：为 MySQL/PostgreSQL/SQLite 建统一元数据契约与方言映射，收口 MCP 响应 JSON 序列化、统一真实生成的 DTO 包路径，配合数据库集成与模板渲染门禁消除跨库、跨版本生成漂移。
- 性能与安全**：隔离同步 DB 调用与事件循环，缓存连接级表元数据以降低重复 schema 采集；拒绝带锁 SELECT、收紧 code-diff 路径边界并脱敏凭据，关键路径通过第三方安全评估。
- 工程交付**：以分层 CI 门禁支撑每次提交可复现，15 个 ADR 沉淀架构决策，提交规范与文档同步入库；仓库获 35 Stars 并合入外部贡献者 PR，体现可审计交付与公开协作。