

李玉锋

dreamer22qq@gmail.com | 上海

教育经历

杭州电子科技大学 | 计算机科学与技术·本科

2019.09 - 2023.06

工作经历

美团 | 点评技术部·Picasso 动态化容器研发

2023.07 - 至今

负责大众点评 App Picasso 动态化容器研发，覆盖 JavaScript 上层、引擎、三端容器及 CLI 开发工具。

美团 | 点评技术部·研发实习生

2022.06 - 2023.02

建设 Picasso 双端 JavaScript 引擎调试器，统一 iOS、Android 调试体验，提升框架研发与问题定位效率。

项目经历

Picasso CLI 3.0: 面向 GUI Agent 的自动化验证与验收 | AI 研发工具链

2026.03 - 至今

- 将 Picasso CLI 重构为 C/S 架构，面向 iOS、Android 统一封装设备管理、页面操作、截图、视图树、日志及 LiveLoad 等原子能力，并通过 CLI / MCP 提供调用。
- 基于上述能力建设 GUI Agent 自动化验证与验收闭环，融合截图、视图树、运行时上下文和日志，使 Agent 可在真实 App 中完成核心链路操作、多轮验证及运行态证据留存。
- 同时支持 D2C 生成结果的真实设备验收，串联设计稿、源码、Mock 数据与 LiveLoad，基于“生成一对比一纠正”闭环将规范化输入下首轮还原度提升至接近 90%。

PicassoJSI: 三端统一 JS 底座 | 项目主 R

2026.01 - 2026.06

- 针对三端 JS 引擎接入方式与运行时封装不一致，以 QuickJS 为统一执行引擎建设 PicassoJSI 仓库；将运行时与核心逻辑下沉至 C++ 公共层，平台侧仅保留引擎接入胶水，通过统一 JSI 接口向上屏蔽差异，实现存量代码无痛适配与三端能力对齐。
- 基于统一底座，将 iOS 验证的字节码方案工程化为可发布、可灰度、可回归的链路，补齐产物发布、缓存、调试、自查及 QA SOP；支撑多个核心页面上线、覆盖 Picasso 大部分容器流量，取得加载性能与稳定性收益。

Picasso iOS: QuickJS 技术验证与字节码链路 | 项目主 R

2025.07 - 2025.12

- iOS 17 发布后，系统 JavaScriptCore 持续崩溃；推进 QuickJS 替代引擎验证，封装 JSC 兼容层实现上层无感切换，并针对共享引擎的性能负收益与全局对象污染，将多页面共享引擎改为页面独占的 QuickJS 实例。
- QuickJS 字节码需通过独立产物链路与 JS 配套发布，因此改造 Picasso 发布流水线，新增字节码编译、产物上传与下发能力，并扩展 PicassoClient 拉取缓存及容器侧消费执行；上线实验中 Crash 率降低 2bp，典型页面容器准备加载耗时由 42ms 降至 11ms。

Picasso HarmonyOS: 容器从零建设与性能优化 | 核心开发

2024 - 2025

- 在 HarmonyOS 原生生态及 ArkUI / C-API 尚处早期、缺少成熟容器方案时，参与从零建设 Picasso HarmonyOS 端，负责壳工程、运行调试体系与基础视图，支撑大众点评鸿蒙版上线。
- 早期基于 ArkUI 状态驱动适配 Picasso 命令式刷新；验证中发现性能瓶颈后，参与 C++ 化渲染链路改造，基于 C-API 封装视图基类，负责刷新、内嵌组件及 List 优化，使核心页面达到 120 FPS。
- C++ 渲染链路落地后，进一步定位跨语言数据层的内存冗余与调用开销；以 `std::variant` 重构类型存储，单对象内存由 80B 降至 32B，并通过批量序列化减少 JS、ArkTS 与 C++ 间的调用，将核心页面 DSL 视图树解析耗时由 40ms 降至 8ms。

技术能力

- 语言：C/C++、JavaScript / TypeScript。
- 平台与引擎：HarmonyOS ArkUI / C-API、iOS、Android；QuickJS、JavaScriptCore、JSI、字节码。
- 工程实践：动态化容器、跨语言通信、Native 渲染与性能优化；CLI、MCP、CDP、移动设备自动化。

个人总结

- 近三年持续深耕 Picasso 动态化容器，形成从单端容器建设、JavaScript 引擎与字节码链路到三端统一底座的技术积累，覆盖 C++ 底层复用、跨端架构与性能优化。近期以 Picasso CLI 3.0 为入口，建设 GUI Agent 自动化验证与验收能力，并支持 D2C 的真实设备交付闭环。