

刘源泉 前端专家

北京 / 男 / 9年前端

522703297@qq.com / 13538356164 / lyq19920919

个人优势

- 1.9年前端开发经验，具备扎实的计算机基础（数据结构与算法、计算机网络、操作系统），熟练掌握 HTML、CSS、JavaScript，精通 Vue/React 框架，具备底层原理理解能力
- 2.具备独立负责中等复杂程度产品的能力，主导过多个从0到1的项目（Trace SDK、Canvas编辑器、UBA平台等），在架构设计、性能优化、工程化实践等方面有丰富经验
- 3.具备团队协作和技术影响力，能够协同设计和产品等团队推动需求落地，推动团队技术基建沉淀，促进团队技术氛围，具备辅导团队工程师成长的能力

工作经历

昆仑万维 | 前端专家 | Skywork-工程六组

2024.11 – 至今

- 1.独立负责 SkyReels 官网从 0 到 1 的架构设计与开发，采用 Nuxt 3 SSR 架构，设计 SSR/CSR 状态同步机制（useState + 环境判断 + 客户端修正）解决移动端检测水合不匹配问题，实现 Ant Design Vue 样式提取与 CSS-in-JS 处理，通过 Manual Chunks 代码分割策略将 vendors 包减少 62%（1,264 KB → 483.42 KB），构建时间减少 65%（~25s → 8.82s）；实施性能优化（图片懒加载、视频封面图方案、资源预加载、动态导入优化），建立完整的 Web Vitals 性能监控系统，实现 LCP < 1s、CLS < 0.1、FCP 168ms、TTFB 3ms 的优秀性能表现，首屏加载时间减少 40%
- 2.主导 Canvas 编辑器核心引擎（Render）的架构设计与重构，采用分层模块化架构，将 882 行单体类重构为职责清晰的模块化系统（Core/Events/History/Settings + Tools/Handlers/Draws/Graphs），实现连接线自动路径规划（基于 A* 算法，支持直线/折线/曲线/贝塞尔等多种连线类型），设计并实现坐标系统转换机制（统一处理屏幕坐标、逻辑坐标、Stage 坐标三种坐标系），代码可维护性和系统性能显著提升

Skyreels: <https://www.skyreels.ai>

小红书 | 前端专家 | 平台产品组

2021.9 – 2024.9

- 1.带领团队完成了 UBA 的从零到一的建设，通过数次关键性升级，该平台现已发展成为集数据看板、自助分析、指标归因、数据门户等功能于一体的 一站式数据分析平台。
- 2.小红书前端工程化实践：主要包括前端规范体系的建立、前端工程化的推进、前端性能优化的逐步落地、通过优化架构提升整体效能、团队技术品牌建设。

好未来 | 前端专家 | 应用产品研发部

2019.5 – 2021.8

- 1.主导 Trace SDK 的整体架构设计与核心功能实现，采用三层架构（入口层/核心层/插件层）和插件化设计，实现基于 Proxy沙箱隔离 的代码执行环境，设计 多级优先级调度系统（集成到核心层，自动优先级判断，实时/批量/空闲三级策略）优化上报性能，支持错误监控（JS/Promise/资源/HTTP错误）、性能监控（Web Vitals/资源/页面性能）、用户行为跟踪等功能；核心技术难点包括：Proxy属性拦截的完整性处理、拒绝名单的多模式匹配、requestIdleCallback兼容性降级、HTTP拦截的XHR/fetch双重覆盖、事件缓冲和批量上报机制等。SDK支持UMD/ESM/CJS多格式构建，已在 PC、H5等业务中落地。
- 2.负责 Trace SDK 的性能优化与工程化实践，通过事件采样率控制、批量上报策略将网络请求减少 60%，SDK体积控制在 50KB 以内（gzip后），对业务性能影响小于 1%；建立完整的构建体系（Rollup多格式构建、TypeScript类型声明、SourceMap支持），设计插件化测试框架支持单元测试和集成测试，编写技术文档和最佳实践指南，推动SDK在团队内的推广使用，提升团队监控能力建设。

滴滴 | 资深前端 | 泛前端技术部

2018.10 – 2019.3

1.负责 滴滴出行 小程序核心功能开发与性能优化，面对业务快速增长导致主包体积超限的挑战，设计并实施小程序分包架构方案，通过代码分析工具识别冗余代码和未使用资源，采用资源压缩、Tree Shaking、按需加载等优化手段，精细化配置分包策略和分包预加载机制，实现主包体积减少 40% 以上，成功将主包控制在平台限制内，保障业务顺利发版

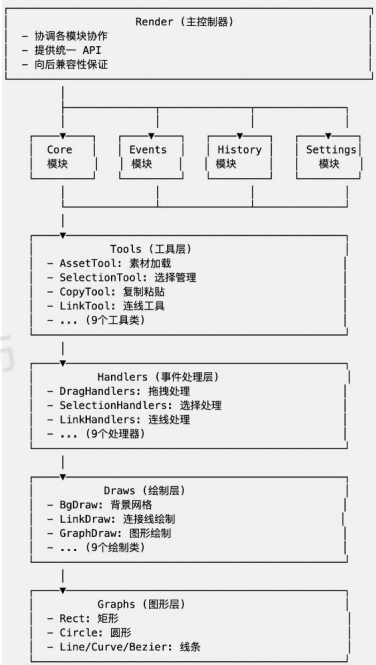
响巢看看 | 前端开发 | 研发部 2016.5 – 2018.3

1.主导 明星空间 小程序从0到1的架构设计与开发，负责技术选型、工程化方案设计和基础架构搭建，实现直播和录播场景的音视频接入与优化，通过CDN加速、码率自适应、预加载策略等技术手段解决延迟问题；封装公共基础组件库，提升团队开发效率 50% 。

项目经历

Canvas 编辑器核心引擎 (Render) 2024.11 – 至今

- 项目简介: 基于 Konva.js 的 Canvas 编辑器核心引擎，支持无限画布、素材编辑、智能连线、撤销重做等功能，采用分层模块化架构设计，提供高性能、可扩展的编辑器能力
- 技术栈: TypeScript Konva.js Vue3 A*算法
- 架构设计:



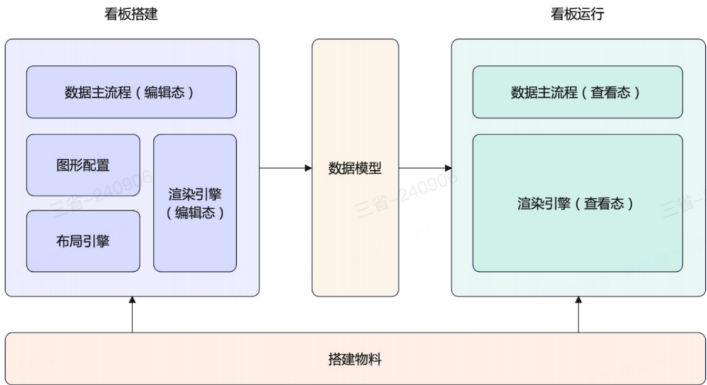
- 工作内容:
- 【整体架构】采用分层模块化架构，将 882 行单体类重构为职责清晰的模块化系统，包含 Core (Stage/Layer 管理)、Events (事件系统)、History (撤销/重做)、Settings (设置管理) 四个核心模块，以及 Tools (9个工具类)、Handlers (9个事件处理器)、Draws (9个绘制类)、Graphs (6个图形类) 四个功能层
- 【连接线路径规划】基于 A* 算法实现智能连线系统，将画布划分为网格，考虑障碍物（其他节点）进行路径查找，支持直线、折线、曲线、贝塞尔曲线等多种连线类型，解决了复杂交互场景下的路径计算问题
- 【坐标系统转换】设计并实现统一的坐标转换机制，处理屏幕坐标、逻辑坐标、Stage 坐标三种坐标系统，确保缩放、拖拽等操作下坐标转换的实时性和准确性，所有模块统一使用转换接口，保证坐标一致性
- 【撤销/重做系统】实现可靠的撤销/重做功能，使用 JSON 序列化保存完整画布状态，通过历史记录数组和索引管理，支持无限撤销（内存允许），确保状态恢复的一致性
- 【多选和变换器集成】实现多选功能，与 Konva.Transformer 深度集成，动态调整 Transformer 配置，处理多选时的层级提升，实现选择框的绘制和交互
- 【磁贴对齐系统】实现智能对齐功能，计算所有节点的对齐参考线，实时检测鼠标位置与参考线的距离，达到阈值时自动吸附，显示对齐线提示，提升用户体验
- 成果:

- 【架构优化】代码从 882 行减少到 560 行，模块职责清晰，代码可维护性显著提升，通过基类复用减少重复代码
- 【算法应用】A* 路径规划算法成功应用于连接线系统，支持多种连线类型，解决了复杂交互场景下的路径计算问题
- 【性能优化】通过按需重绘机制、图层分离优化、事件防抖处理等方式，提升系统性能，确保流畅的用户体验

UBA（一站式数据分析平台）

2022.12 – 2024.9

- 项目简介：UBA 是小红书数据平台部自研的一站式数据分析平台，包含数据看板、自助分析、指标归因、数据门户等功能
- 技术栈：`vue3` `antv` `hulk` `pytest` `playwright` `pixelmatch`
- 工作内容：
 - 【整体架构】整个平台采用分层设计，包含前端项目、业务组件库、BFF层、UI自动化测试四个部分。
 - 【看板引擎】可视化看板系统与低代码在宏观上存在相似性，但是在业务侧重点上又存在不同，最终抽象和设计了一套通用的 `看板引擎`，实现看板的生产和消费。
 - 【性能优化】从页面级指标和图表级指标两个方面对看板进行性能优化。页面级指标主要优化了 LCP、CLS、FCP 等指标；图表级指标主要从 `数据加载性能`、`图表x轴渲染卡顿`、`表格性能优化` 三个方面进行优化。
 - 【技术自驱】做了一些工程治理与维护的工作，保证研发质量和研发效率；从 `webpack` 迁移到 `rsbuild`，打包性能提升了10倍左右，大幅度提高开发效率。

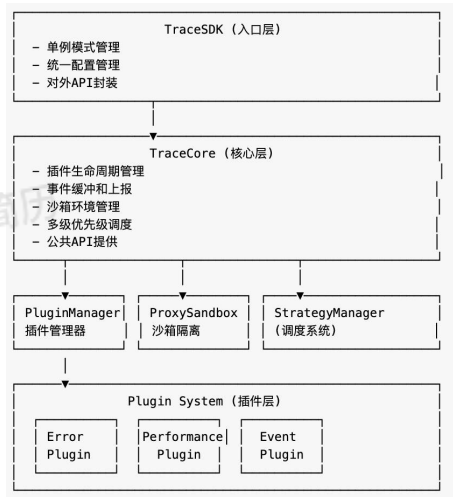


- 成果：
 - 【性能优化】页面级指标：LCP < 1s, CLS < 0.1；图表级指标：表格类呈现时长 P95 `6s -> 2s`，图表类呈现时长 P95 维持在 `50ms` 以内；
 - 【团队基建】推动团队业务组件库、图表库与基础库沉淀，完成了 `30+` 个业务沉淀，提高了团队的协作开发效率

Trace SDK（前端全链路监控 SDK）

2019.5 – 2019.8

- 项目简介：Trace SDK 是一个全面的前端全链路监控SDK，采用三层架构（入口层/核心层/插件层）和插件化设计，基于Proxy沙箱隔离实现安全的代码执行环境，集成多级优先级调度系统（自动优先级判断）优化上报性能，支持错误监控（JS/Promise/资源/HTTP错误）、性能监控（Web Vitals/资源/页面性能）、用户行为跟踪等功能，已在 PC、H5等业务中落地
- 技术栈：`TypeScript` `ES6 Proxy` `Performance API` `Web Vitals`
- 架构设计：



● 工作内容:

【架构设计与沙箱隔离】采用三层架构设计（入口层/核心层/插件层），基于接口的插件化架构，实现插件生命周期管理、依赖管理、错误边界处理；基于ES6 Proxy API实现代码隔离执行环境，通过拦截器实现属性访问控制，支持拒绝名单机制，确保插件代码安全执行且不污染全局环境

【多级优先级调度与事件上报】设计三级优先级调度系统（集成到核心层，自动优先级判断），高优先级数据立即上报，中优先级数据定时批量上报，低优先级数据通过requestIdleCallback利用浏览器空闲时间处理，实现事件缓冲和批量上报策略，提供Safari等浏览器的降级方案，减少网络请求60%

【错误与性能监控系统】实现完整的错误捕获机制，覆盖JS错误、Promise错误、资源加载错误、HTTP请求错误（拦截XHR/fetch），支持错误过滤、采样率控制；基于Performance API和PerformanceObserver实现Web Vitals指标收集（LCP/FID/CLS），支持Navigation Timing API收集页面加载性能，提供旧版浏览器的降级方案

● 成果:

【架构设计】成功设计并实现插件化架构，支持3个核心插件（Error/Performance/Event），将调度系统集成到核心层并实现自动优先级判断机制，代码可维护性和可扩展性显著提升，插件间通过沙箱隔离互不影响

【性能优化】通过事件采样率控制、批量上报策略将网络请求减少60%，SDK体积控制在50KB以内（gzip后），对业务性能影响小于1%，支持UMD/ESM/CJS多格式构建

【技术攻关】攻克Proxy沙箱隔离、多级优先级调度系统、HTTP拦截兼容性、requestIdleCallback降级等核心技术难点，建立完整的构建体系（Rollup多格式构建、TypeScript类型声明、SourceMap支持）和测试框架

教育经历

● 湖北经济学院 – 信息管理学院 – 软件工程 – 本科 2010.9 – 2014.6