

个人信息

- 姓名:陈欣聪
- 联系电话:18116302795
- 工作年限:12年(2013-2025)全栈开发经验
- 邮箱:STLCXC@163.com

求职意向

- 目标职位:资深全栈工程师(Web/AI 方向)
- 期望行业:互联网/科技/企业服务
- 核心技术栈:
 - 主力框架:Node.js/React/Angular(精通状态管理与组件化设计)
 - 拓展栈:Golang/Vue
 - 前沿技术:AI集成(RAG/智能搜索)/低代码平台/微服务架构(Nx Monorepo/MFE)
- 期望工作地点:上海(可远程协作)

优势亮点

1. 全栈架构与高性能优化:

主导5+企业级系统设计(智能办公平台/AI低代码平台),核心模块性能优化超80%,接口响应速度从5s优化至1s内(PwC项目).

2. AI技术落地能力:

集成RAG流程/智能搜索系统,推动低代码平台与内部系统融合,数据分析效率提升40%(AI Flow Creator项目).

3. 跨平台工程实践:

精通微服务架构(Nx Monorepo迁移/MFE前端架构),统一多云消息队列(Azure/AWS/GCP),降低部署成本30%.

项目经验

普华永道信息技术(上海)有限公司(2019/03 - 2025/06)

PwC Workforce Orchestrator(2019/03 - 2023/06)

- 项目概述:**该项目旨在通过定制化数字体验提升团队生产力与参与度.Workforce Orchestrator 作为智能数字平台,实时连接人员/信息与应用,打造高效工作环境.借助移动参与平台实现一对一沟通,支持个性化消息推送/任务管理.利用自动化聊天机器人响应咨询,通过脉冲调查收集反馈以调整策略,全方位赋能员工,培育持续改进文化.
- 技术栈:** 全栈: TypeScript/ApolloGraphQL 前端:Angular 后端:Node.js/Express 云服务:GCP Firebase/Pub/Sub/Google Dialogflow 数据库:PostgreSQL/Redis
- 个人贡献:**
 - 设计分布式实时通信系统(Firebase),实现99%+消息推送成功率,支撑高并发场景(峰值2000+TPS).

- 基于Pub/Sub+Express+GraphQL优化异步数据处理,系统响应速度提升60%.
- 主导单点登录系统设计,实现跨平台(PC/移动端)独立登录逻辑,解决同平台互斥登录问题.
- 抽象消息服务层,统一Teams/短信/网页IM的调用接口,支持渠道服务数量动态配置
- 集成Dialogflow实现智能客服,完成自然语言处理与动作绑定,用户满意度提升35%.
- 参与迭代了ui版本使得界面更加清晰流畅,manager和onboarder的权限页面更加明确,使用户页面效率提升了50%.
- 参与设计实现了feature flag功能确保系统的部分可用对于网页,后端以及消息入口的用户拥有相同的权限,已经动态的用户可见

Model Edge(2023/06 - 2024/06)

- **项目概述:**提供更优的模型治理与验证模式,支持端到端的模型生命周期管理.在单一平台中实现全量模型组合(含 AI 模型)的管理/开发/验证及治理,通过展示模型有效性和可解释性,增强对模型项目的信心.
- **技术栈:** 全栈: TypeScript/ApolloGraphql 前端:Angular 后端:Node.js/Express 云服务:GCP Firebase/Pub/Sub/Azure Service Bus/AWS SQS 数据库:PostgreSQL/Redis
- **个人贡献:**
 - 设计ResponseAI系统及风险计算逻辑,通过模型动态选择使风险评估准确率提升25%,支撑千万级交易场景.
 - 集成了基于azure用户认证的登录系统,增加了登录方式的多样性支持
 - 统一多云平台消息队列服务(Azure Service Bus/AWS SQS/GCP Pub/Sub),降低跨云部署成本30%,代码复用率提升至80%.
 - 用JMeter进行性能测试与优化,核心接口响应时间从5s优化至800ms,满足SLA 99.9%要求.

AI Flow Creator(2024/07 - 2025/05)

- **项目概述:**内部 POC 项目,基于 Flowise 的 AI 低代码平台,集成公司内部系统,定制前端 AI 和组件,方便构建 AI 机器人.
- **技术栈:** 全栈: TypeScript 前端:React 后端:Node.js/Express/TypeOrm/Long Chain 云服务:LLM api(chatGpt/ollama) 数据库:PostgreSQL/Redis
- **个人贡献:**
 - 解决低代码平台集成难题:设计用户单点登录与资源分组方案,通过OAuth 2.0+JWT实现跨系统权限管控,保障数据安全性.
 - 开发可复用组件体系:实现后端runtime库与前端组件导出结构,支持AI回答图标插件等扩展功能,推动平台组件复用率达70%.
 - 构建AI应用场景:
 - 开发“问题转SQL+图表可视化”流程,使数据分析效率提升40%.
 - 实现ai搜索功能,搜索效率相比传统浏览器提升了80%.
 - 实现基于RAG的大文档问答系统,通过向量检索使回答相关性提升60%.

Concourse(2024/09 - 2025/05)

- **项目概述:**一个简化/智能且透明的交付平台,让复杂交互变得轻松.
- **技术栈:** 全栈: TypeScript/ApolloGraphql 前端:React/Azure Power App 后端:Node.js/Nest.js 云服务:Azure Service Bus 数据库:PostgreSQL/Redis
- **个人贡献:**
 - 后端负责维护各种数据的 CRUD API,保障了数据的高效管理和操作.
 - 前端参与主页设计和开发,提升了平台的用户界面体验.
 - 编写脚本用于维护和更新客户数据,提高了数据处理的自动化程度和准确性.

- 了解里mfe的前端项目结构使得大项目协作的模块分工更加明确独立
- 参与了Monorepo的迁移,基于nx,解决了微服务架构下多服务版本库不一致已经项目启动复杂臃肿的问题 = 参与了前端部分页面从azure power app向react迁移的实现和设计

上海秀店网络科技有限公司(2018/08 - 2019/02)

某电视台 app(2018/10 - 2019/02)

- **项目概述:**该 app 包含电视节目查看提醒/商城/游戏/抢红包等功能.
- **技术栈:** 全栈: TypeScript 前端:Vue 后端:Node.js/Egg.js/Express/Koa/RabbitMQ 云服务:阿里云 数据库:Mysql/mongodb/Redis
- **个人贡献:**
 - 在 app 对接阶段,参与接口规范的讨论与设计,统一了客户端接口标准,提高了系统的兼容性和可维护性.
 - 开发锦鲤活动的全部逻辑,合理运用消息队列解耦 app 原生逻辑与活动逻辑,提升了系统的灵活性和可扩展性.
 - 设计并实现一套奖品分发逻辑,用于正态分布奖品,优化了抽奖活动的公平性和趣味性.
 - 利用 Redis 实现 session 共享,解决了 JWT 发放时间不可中途把控的问题;通过 SortedSet 统计最常玩的游戏;使用过期 key 实现定时任务,提高了系统的性能和稳定性.
 - 参与阿里云互动直播邀测,设计部分直播逻辑原型,为后续直播功能的开发奠定了基础.

秀店 2.0(2018/08 - 2018/10)

- **项目概述:**电商平台,包含商品/库存等模块,本人参与推广部分/合伙人规则制定以及分润结算.
- **技术栈:** 全栈: TypeScript 前端:Vue 后端:Node.js/Egg.js/Express/Koa/RabbitMQ 云服务:阿里云 数据库:Mysql/mongodb/Redis
- **个人贡献:**
 - 开发推广模块 app 端接口,满足安卓和 iOS 需求.
 - 参与后台接口开发和部分业务逻辑设计,为电商平台的推广和运营提供了技术支持.
 - 参与了合伙人奖金分发结算部分的实现和设计

上海金桥信息股份有限公司(2013/10 - 2018/08)

会议平台系统(2016/10 - 2018/08)

- **项目概述:**基于第三方音视频 SDK 的会议系统,具备建会/约会/控会/直播/录播等功能,衍生出教育行业公开课直播/法院行业执行值守等定制产品.
- **技术栈:** 全栈: Javascript 前端:Backbone 后端:Node.js/Express 云服务:阿里云 数据库:mongodb/Redis
- **个人贡献:**
 - **设计阶段:**
 - 引入认证中心,解决了各项目独立开发用户系统及系统间同步问题,提高了用户管理的效率和一致性.
 - 设计统一权限管理和外部应用介入管理,为后续微服务接入提供了统一的配置和分发方案.
 - 推进使用 Markdown 书写文档,提升了文档的书写/发布/迭代和共享效率.
 - **开发阶段:**
 - 开发第三方 Node.js SDK,为公司其他项目提供标准的 HTTP 接口,方便不同技术栈的接入.
 - 参与钉钉对接通讯录/短信调用/外部录播直播接口调用的设计和开发,实现了服务的单一职责架构.

- 使用 Mocha.js/Nock/SuperTest 实现自动化单元测试,降低了敏捷开发中代码迭代带来的风险.
- 运用新版本 Node.js 的 async/await 改进异步代码的可读性.
- 引入基于 Git - Flow 的开发流程,帮助团队从 SVN 切换到 Git,解决了代码分支混乱和版本发布规范问题.
- 参与 MongoDB 集群的设计和推进,优化数据库设计,将 1000 QPS 的请求时间从 6s 减少到 3s,提高了系统的并发性能和数据库稳定性.
- 引入 Redis 缓存页面,加快了 Jade 模板页面的渲染效率.
- 部署阶段:
 - 使用 Docker/AliNode/NewRelic 管理发布和监控线上及测试环境,提高了团队发现代码问题和性能优化的能力.
 - 配置 Nginx 与各服务集群,实现同一服务下各个实例的负载均衡.

排课系统(2015/05 - 2016/05)

- **项目概述:**基于老师的排课系统.
- **技术栈:** 全栈: Javascript 前端:Backbone 后端:Node.js/Express 云服务:阿里云/本地服务器 数据库:mongodb/Redis
- **个人贡献:**
 - 在预约系统的基础上使用backbone实现了排课组件,使得老师在排课的时候能够更精确顺滑的选择上课时间,此组件性能提升了100%,并且在时间计算上更加精确可以匹配到分钟因为是通过拖拽事件获取了移动距离
 - 到项目上帮助部署和维护项目的升级和运行

会议预约系统(2014/10 - 2015/12)

- **项目概述:**Web 预约系统,配合安卓终端放置在会议室使用.
- **技术栈:** 全栈: Javascript 前端:Backbone 后端:Node.js/Express 云服务:阿里云/本地服务器 数据库:mongodb/Redis
- **个人贡献:**
 - 到不同项目上帮助部署和维护项目的升级和运行
 - 将公司内部框架替换为 Express 框架,解决了原框架数据库和数据持久化的可靠性问题;引入 MongoDB 作为主力数据库.
 - 使用backbone,实现了一个预约组件基于div格子确保每个半小时的时间段

公告系统(2013/10 - 2014/10)

- **项目概述:**Web 界面排版,配合安卓和 Windows 终端展现的公告服务系统.
- **技术栈:** 全栈: Javascript 前端:Backbone 后端:Node.js/ESP(内部框架) 云服务:阿里云/本地服务器 数据库:SQLite
- **个人贡献:**
 - 在 callback 时期,使用 Async.js 管理异步代码的可读性;首次使用 Backbone.js 实现组件化开发,设计文件下载/查看/管理等功能.
 - 参与了公告编辑系统的设计和开发,网页拖拽布局组件,后端存储,客户端渲染不同组件表现形式,天气,时间,ppt,文档,文字等等

客户端项目

微信的门禁(2015/10 - 2016/10)

- **项目概述:**对接企业微信和门禁提供厂商,使用扫一扫打开公司门禁系统.
- **技术栈:**Node.js/Express/微信公众号开发
- **个人贡献:**成功完成微信对接,解决了部分同事忘带门禁卡的问题.

HTML5 扫一扫导航(2016/02 - 2016/06)

- **项目概述:**HTML 制作的简易导航流程,配合门禁内置微信体系中.配合公司导览机项目,在点位稀少的环境中实现导览功能.
- **技术栈:**Node.js/Express/Backbone.js
- **个人贡献:**使用h5实现了基于当前点位二维码扫码后,进行的地图导览

开发基于 **Backbone.js** 的表格以及分页组件

- **项目概述:**总结多个项目经验,开发基于 Backbone.js 的可复用组件,如基于 Collection 自动生成的可定制表格.
- **技术栈:**Node.js/Express/Backbone.js
- **个人贡献:**完成组件开发,提高了代码的复用性和开发效率.

对接钉钉的 **HTML5** 小应用(通讯录同步)

- **项目概述:**类似微信对接,实现钉钉通讯录同步.
- **技术栈:**Node.js/Express/MongoDB
- **个人贡献:**完成钉钉对接工作.

其他项目

- 开发基于阿里云的短信的 npm 模块(当时官方不支持 Node.js)
- 开发梦幻西游加点简易版模拟器,解决当时系统更新后加点模拟器不可用的问题
- 开发第三方的 API 封装模块(与短信系统类似)

专业技能

1. 技术栈体系

(1)服务端开发

- **核心语言:**Node.js(javascript)/Golang(熟练Gin框架)/bash脚本
- **框架与工具:**Express/Nest.js/Apollo(Graphql),掌握微服务架构(Monorepo/Nx)/容器化部署(Docker/Kubernetes)/nginx
- **数据库与缓存:**关系型(PostgreSQL/MySQL)/非关系型(MongoDB/gcp-firebase)/缓存(redis),擅长索引优化与集群设计
- **消息通信:**RabbitMQ/GCP Pub/Sub/Redis/Azure - Service - Bus/aws-sqs/MQTT,实现高并发场景下的异步数据一致性,restful api, GraphQL, Grpc实现同步数据通信,Socket.io/websocket全双工通信

(2)前端开发

- 框架:React/Angular(精通状态管理与组件化设计)/了解Vue/精通Backbone.js
- 组件库:Material UI/Ant Design/Element UI
- 工程化:Webpack/Vite构建优化,TypeScript静态类型检查,MFE架构

2. 工程能力

- 部署与监控:Nginx负载均衡/阿里云/GCP/Azure/Aws云平台管理/NewRelic性能监控
- 测试与CI/CD:Mocha/JMeter/Postman/Jtest自动化测试/GitFlow版本管理/Jenkins持续集成
- AI辅助开发:GitHub Copilot代码生成/低代码平台二次开发

教育经历

- 学校:上海第二工业大学
- 学位:本科(学士)
- 专业:软件工程专业
- 时间:2009.9 - 2013.6

语言能力

- 英语:CET-4,熟练阅读技术文档(如MDN官方文档)/英文技术会议沟通
- 普通话:母语水平,技术方案汇报能力突出