

邓乐涛

☎ (+86) 186-7500-3737 ✉ 584605539@qq.com 🌐 TOETOE55

教育经历

陕西师范大学
软件工程

西安
2017.7 - 2021.6

工作经历

RustSDK 开发工程师（客户端），日历业务
字节跳动，飞书

北京
2021.7 - 2022.5

- 负责日历会议室视图需求的开发与迭代
- 参与日历日程同步流程的重构工作

RustSDK 开发工程师（客户端），平台能力方向
字节跳动，飞书

北京
2022.5 - 2023.6

- 主导 rust-sdk 一些业务基础能力（比如一些网络重试基建、三方加密库等）的重构、设计与开发
- 深度参与 rust-sdk 用户态架构的迁移与设计
- 参与 rust-sdk 测试框架的设计与后续搭建
- 参与 rust-sdk 开发教程的开发

iOS 开发工程师，多轨工具与素材方向
字节跳动，剪映

广州
2023.6 - 2024.4

- 参与基础剪辑与智能剪辑需求的开发迭代
- 完善了一些与 ab 实验相关的基础功能
- 在剪映工作流程引入新的日志分析系统提高问题排查效率
- 一些 oncall 自动化的工作

项目经历

会议室支持多层级

RustSDK RD/TECH OWNER

2021.8 - 2022.6

- 概述：**KA 客户定制化需求，使得会议室可以灵活设置在一个树状的组织结构下，以便支持会议室与组织架构绑定，以及批量预约会议室的能力
- 产出：**
 - 一期定义了多层级会议室的基础结构以及 CRUD 接口，从 0 到 1 搭建起完整的多层级会议室链路
 - 二期定义了旧会议室结构与多层级会议室结构的 mapping 和新旧系统的切换规则，支持新旧会议室系统的兼容，方便客户在新旧会议室系统之间迁移
 - 三期引入缓存、迭代拉取、分页拉取等技术，完善了若干体验问题（如查询速度），对标旧会议室系统

日程同步流程重构

RustSDK RD

2021.11 - 2021.12

- 概述：**日程通过日程同步机制实现客户端 <-> 服务端之间数据的同步，日程同步是日程保存、修改、查看、回复等功能的核数据流。重构日程同步流程以解决若干遗留问题，比如接口对服务端压力大、数据不一致等
- 产出：**
 - 对原先单一接口单一流程，拆解为读与写的两部分，在不同时机选择走不同的流程，减少了功能的耦合，降低了对服务端压力
 - 将日程参与人的部分从原先的日程全量的同步中拆分出来，减少每次同步的数据量，同样减轻了服务端的压力
 - 重新约定客户端、服务端同步的协议，SDK 侧使用同步队列的方式，保证了数据的一致性
 - 整体重构后 p99 首屏同步耗时少了 60%，首次同步耗时减少 52%。尤其在首次订阅日历耗时体感明显降低

用户态架构迁移

RustSDK RD

2022.3 - 2023.6

- 概述：**RustSDK 旧框架中，与用户相关的数据、资源分散在各个全局变量中，以及缺乏收敛的用户数据管理能力，导致旧框架下存在用户串数据的问题，也无法支持多用户同时在线。在新架构中将所有用户相关的逻辑全部收敛到“用户态容器”中管理，解决旧架构所带来的问题

- **角色：** 深度参与用户态架构的设计与开发，主要完成三部分工作

1. 业务迁移
2. 基建改造
3. 框架本身功能的完善

- **产出：**

1. 通过对登录登出流程重构，重新整理并定义好登录登出的接口与流程，引入了“用户状态不变量”检查、并发控制、压力测试、流程埋点等手段，消灭了由登录登出引起的用户串数据问题
2. 将长链接，pipeline，客户端调用与推送，迁移至用户态架构内，添加用户校验以及一些生命周期管理的逻辑，从数据源头上解决用户串数据问题。
3. 将基于线程的 worker 基建，重新设计为基于异步且兼容用户态框架的 worker 基建，复用共有的异步运行时，减少多用户环境下的开销；同时利用异步 rust 的特性，能在用户态下更好地管理 worker 的生命周期
4. 迁移了日历核心业务和 im 部分业务逻辑，并指导 vc 业务、ccm 业务的迁移工作
5. 总共迁移 4 万+行代码
6. 基本解决用户串数据问题

SDK 网络请求中间件

RUSTSDK RD/TECH OWNER

2022.12 - 2023.6

- **概述：** 在客户端环境中容易遇到网络不稳定以及杀 App 的情况，对于一些需要“必达”的网络请求，各个业务经常重新实现一套不完整不鲁棒的重试逻辑，会带很多业务问题。所以统一实现一个支持离线和弱网下发起请求，上线后恢复的请求重试基建

- **产出：**

1. 通过维护事件循环来进行并发控制，控制请求任务执行的时机，可以避开一些资源使用的高峰期（比如说冷启动时），可以控制任务执行的并发数，减少网络资源的拥塞情况
2. 将请求重试的过程描述为一个状态机，可以精确定义每个步骤的前置条件和后置条件，方便测试代码的编写
3. 接口上区分是否幂等请求，一方面提醒业务方考虑当前场景是否幂等，还能不同情况下采取不同的重试策略提高性能

容器反转实验

iOS RD

2023.7

- **概述：** 在剪映中引入容器反转实验，可以观察某个业务线长期迭代，引入多个需求的收益

- **产出：**

1. 添加新的配置，可以直接在配置平台配置容器绑定的需求实验，方便后续新需求接入
2. 实现上 hook 了代码中获取实验值的地方，使得命中容器反转实验时，自动返回容器内配置的实验值，而无需新增额外的反转实验的代码

马赛克特效支持应用于多人脸

iOS RD

2023.11 - 2023.12

- **概述：** 剪映的马赛克特效希望可以应用于多个人脸，且支持让用户选择特效应用于哪个人脸

- **角色：** 主要负责在播放器上人脸编辑的能力

- **产出：**

1. 多人脸马赛克之于旧的单人脸马赛克导出渗透率有明显提升
2. 搭建了多人脸特效的通用链路，支持后续直接添加可编辑的多人脸特效
3. （使用混合模式）额外实现了一个支持挖多个孔蒙层的组件

数字人克隆

iOS RD

2024.1 - 2024.3

- **概述：** 可以通过录制一段视频，训练出一个专属的个人数字人形象，应用于文本朗读中

- **工作：**

1. 主要负责拍摄链路的开发，以及新数字人面板前期的设计工作
2. 拍摄器部分每个组件都是用 MVVM 模式，再通过一个主 VC 和主 VM 将所有组件组合在一起，负责组件间通信以及生命周期管理。比较地将几个拍摄阶段、拍摄状态以及环境检查状态几个相互耦合的状态管理起来。

个人项目

cfg-vis

DESIGNER/DEVELOPER

TOETOE55/cfg-vis

- 使用过程宏模拟 cfg，可以控制 rust items 的模块可见性。可以应用在跨 crate 写内部测试的场景（由飞书 RustSDK 框架引出的需求）

2024-04-12

邓乐涛 · RÉSUMÉ

2

- 在 crates.io 上已有 86k 下载量

dep-inj

TOETOE55/dep-inj

DESIGNER/DEVELOPER

- 一个 rust 静态的依赖注入库，允许使用者通过泛型进行依赖注入，在编译时可以检查依赖项是否注入。这是由飞书用户态框架引出的需求，以解决动态依赖注入带来的各种问题。
- 已经有商业项目使用 <https://github.com/kuintessence/agent>

lens-rs

TOETOE55/lens-rs

DESIGNER/DEVELOPER

- rust 的 lens 库，可以通过组合“lens”，直接访问复杂的深层次结构。
- 已获得 106 个 star

个人规划

技术方面

- 充分掌握大前端（客户端、Web 端以及客户端 SDK）各项基本技术，能从比较全局的视角上看问题；后续也打算尝试接触更多后端相关技术。

作为工程师方面

- 积累更多业务经验，培养项目推进以及沟通能力。

博客

- Is RVO the Basis for Rust's Move trait and Placement New?
- 线程安全、原子变量、内存顺序
- 你懂 Unicode 吗？可以教教我吗？
- 稍微聊聊 Rust 中的 Invariant —— 那些必须保持的性质
- 来谈谈 Rust 的大目标和原则，
- 函数式的动态规划

工作技能

编程语言 **Rust**, Swift, TypeScript, Haskell, Kotlin, Java, C

工具框架 tokio(Rust), UIKit(Swift), Node.js(TypeScript), React(Typescript), sqlite, git

其它知识储备 编程语言理论（类型系统，静态分析），多线程（线程安全，原子变量内存顺序相关），字符串处理（unicode 相关知识），函数式编程