

■基本信息：

姓 名：	周尧云	性 别：	男	婚姻状况：	已婚
年 龄：	29 岁	工作年限：	5 年	户籍：	湖南省邵阳市
第一学历：	硕士	专 业：	软件工程		
毕业学校：	中国科学技术大学				
住址：	深圳市南山区大新地铁站附近				

■期望：

期望职位：	Android 高级工程师或架构师
-------	-------------------

■教育经历：

时 间	学 校	学 历	专 业
2009.09 - 2012.03	中国科学技术大学	研究生	软件工程
2005.09 - 2009.07	哈尔滨工业大学	本 科	软件工程

■工作经历：

时 间	任职公司	职 务
2014.06 - 现在	腾讯	移动终端小组技术负责人
2011.11 - 2014.05	瓶子科技有限公司(刷机精灵)	Android 高级工程师

■技术优势：

- 1) 具有 Android 客户端架构经验，通过模块化、组件化的设计提升了代码的可重用性，提升了新产品开发的效率，并降低了开发成本
- 2) 深刻理解常见设计模式，实践中用过单例模式，工厂模式，模版方法模式，观察者模式，桥接模式，职责链模式，外观模式，访问者模式，命令模式，代理模式，构造者模式
- 3) 多线程程序的设计有丰富的经验，在保证线程安全方面，有深入研究
- 4) 深入研究过 Android 的 Binder 机制，Activity 启动流程，对 ActivityManagerService 也有深入研究，并为这两个主题写了一系列博客，可参考 www.cloudchou.com
- 5) 具有性能优化经验，包括内存的性能优化，启动时间的优化
- 6) 有非常强的问题解决能力，并有解决问题的一套方法论，对 ANR 问题的定位，内存泄漏问题的定位，OOM 问题的定位，以及常见问题的定位都有丰富的经验
- 7) 有团队管理经验，对技术人才有深刻的理解

■项目经历 A:

项目名称	手机版刷机精灵的重构及维护
项目时间	2015年11月 ~ 至今
项目职务	技术负责人
项目简介	<p>手机版刷机精灵是公司的核心产品，有以刷机功能为核心的一系列功能，比如:ROM 下载，Root，资料备份及恢复，系统优化，安全加固，耗电管理，手机降温，应用市场，卸载预装，自启管理，应用升级，百宝箱，推荐信息流。这么多复杂的功能给技术架构带来很大的挑战。</p> <p>技术要点:</p> <ol style="list-style-type: none">1) 模块化的架构设计 以功能为单位划分模块，每个功能模块都在一个单独的模块工程里，只选择需要的模块即可快速开发出一个新的产品。每个模块还可以为不同的产品定制不同的逻辑，资源，配置以实现定制。通过这种架构设计，每个产品都只包含它需要的模块，逻辑，配置，资源，可以让每个产品的体积尽量小。另外，每个模块只需维护一套供各产品共用的逻辑，具有更低的维护成本。所有产品共用一个公共基础模块，在公共基础模块里设计了一个模块管理器，产品用到的模块对象都会在模块管理器里注册，每个模块都可通过模块管理器取得其它模块，然后调用其它模块的功能2) 提取通用组件，放在局域网的组件仓库，供所有产品使用。每个组件都有版本管理，如果需要使用通用组件，只需在 <code>build.gradle</code> 里添加依赖即可使用该组件，如果组件依赖了其它组件，也会自动下载依赖的特定版本的组件。通过这种方式解决了以前通用代码散布在各个产品带来的可维护性非常差的问题。另外因为这些组件通常都是 <code>aar</code> 包，解决了以前使用第三方组件时还需拷贝动态库，资源文件，设置 <code>AndroidManifest</code> 等多种麻烦问题3) 多进程的架构设计 将内存消耗比较大的逻辑放在非主进程，解决主进程因内存回收带来的卡顿问题，将常驻逻辑放在常驻进程，和主进程隔离，解决主进程经常被系统杀掉的问题，还可以通过多进程解决引入外部库导致的主进程卡死问题4) 模块内部设计采用 MVC，隔离业务逻辑和界面逻辑，应用启动时会初始化每个模块的业务逻辑层，提前缓存该模块的数据，当用户进入该功能模块时因为数据已经准备好，所以可以无缝使用该功能5) 通过并行初始化，延迟非关键组件的初始化等手段保证应用启动速度6) Task-Based 的多线程设计，将任务进行分类，不同的任务放到不同类型的线程池里运行，保证任务的高效执行以及多线程逻辑的可调试性，因为在调试器里可以知道执行的任务所在的线程，可以分析线程当前的状态得到当前任务执行的状态，也能明确是否产生死锁7) 支持抢断的优化任务调度器，可以高效响应用户的优化操作，解决优化任务非常耗时还不能打断的问题。清理扫描时，用户进入安全扫描后，会立即启动安全扫描，安全扫描后又会继续先前未完成的清理扫描。此种情况下还能保证全局的扫描进度是累加的，而不是从头开始。
项目职责	搭建框架，实现核心模块的业务逻辑，解决疑难问题，代码 Review，任务分配

■项目经历 B:

项目名称	精灵助手重构
项目时间	2015年8月~2015年9月
项目职务	技术负责人
项目简介	<p>精灵助手是 PC 端刷机精灵的辅助应用，协助 PC 端刷机精灵完成手机信息的获取，并实现联系人，短信，通话记录，wifi，图片，应用数据等资料的备份和还原。</p> <p>重构时实现了 Jar 包形式的精灵助手，在有 Root 权限的情况下，可以绕过第三方授权管理组件备份还原用户资料(通话记录，短信，联系人)。Apk 形式的精灵助手也做了比较大的架构调整，解决了先前通讯，线程管理混乱的问题，提高了备份，还原的成功率。</p> <p>技术要点:</p> <ol style="list-style-type: none"> 1) 实现 Jar 包形式的精灵助手参考了 content 命令的实现，通过 ActivityManager 调用 getContentProviderExternal 函数可得到 ContentProviderHolder，然后通过该对象指定要操作的 uri 即可查询联系人，短信，通话记录等资 2) Task-Based 的多线程设计以前 PC 端每发送一个命令过来，都会创建一个新的线程，执行该命令，这样可能造成多个线程的同步问题，比如出现两个线程同时还还原资料，造成资料重复。另外因为多线程的原因，增加了分析问题的复杂性。现在采用了 Task-based 的多线程设计，将所有 PC 端的命令分成了 4 种类型，SIMPLE 类型的命令放在单线程的线程池里执行，PIM 类型的命令放在另一个单线程的线程池里执行，并且有看门狗线程负责观察命令是否执行超时，FILE 类型的命令放在固定数目线程的线程池里执行，MEDIA_TIME 类型的命令放在 cache 类的线程池里执行。如果某个命令执行出问题，可以在调试窗口迅速找到执行命令的线程，然后查看线程执行的状态，相比 Actor-based 的多线程设计，有更好的可分析性 3) 我们在和 PC 端通信时，最初设计便采用了 NIO 的通信方式，但是虽然采用了该方式，我们还是设计了 3 个线程用于管理和 PC 端的通信，一个监听 PC 端请求的线程，一个读取 PC 端发送过来的数据的线程，一个发送数据至 PC 端的线程，这样的设计非常复杂，不便于分析。重构后，我们取消了读取数据和发送数据的线程，只保留了一个监听 PC 客户端请求的线程，在该线程里检测到 PC 端发送数据过来后，就调用 ClientSession 的 receiveData 接收数据，如果执行的命令准备好发送给 PC 端的数据了，则通过 SelectKey 的 interestOps 将该通道设置有数据需要发送，则在服务线程里即可知道，然后在服务线程里将数据真正地发送给 PC 端
项目职责	搭建框架，实现核心的业务逻辑(任务调度，和 PC 端通信的逻辑，命令执行框架，部分命令逻辑)，任务分配

■项目经历 C:

项目名称	预研进程注入
------	--------

项目时间	2015年5月~2015年8月
项目简介	<p>通过将我们自己编写的动态库注入至 system_server 进程，这个动态库就获得了 system_server 进程的权限，并且可以管理各种 binder service 对象，拦截各种 binder service 提供的服务，比如 ActivityManagerService,PackageManagerService，这样就可以为手机版刷机精灵提供一些普通进程无法获得的功能：卸载手刷时提醒用户确认是否卸载手刷，这样可以挽留用户；清除通知栏所有其他应用的通知，在做系统优化时可以用到；禁止第三方软件开启自启动。</p> <p>技术要点：</p> <ol style="list-style-type: none"> 1) 通过 ptrace 实现动态库的注入 2) 修改 system_server 使用的 libandroid_runtime.so 的 GOT 表，将 GOT 表中填写的 BBinder:transact 地址改成注入的动态库的入口地址，并保存 BBinder:transact 函数地址，需要时再同坐这个地址调用 BBinder:transact 函数

■项目目经历 D:

项目名称	手刷和 PC 刷机精灵的云端刷机方案
项目时间	2014年4月~2015年5月
项目职务	技术负责人
项目简介	<p>刷机精灵的核心功能是刷机，刷机最核心的技术是刷机方案，最核心的指标是刷机成功率和刷机效率。刷机方案放在云端这种架构使得刷机方案更新非常灵活，可随时更新刷机方案，带来了刷机成功率的迅速提升并且能快速满足运营需求。</p> <p>技术要点：</p> <ol style="list-style-type: none"> 1) 刷机方案内部采用了模板方法，桥接模式，职责链模式来保证刷机流程，刷 Recovery 的方案的可扩展性，使得各个机型的刷机流程，刷 Recovery 的流程可灵活定制 2) 快速提升成功率的方法论：上报错误信息，分析各个错误占比，针对主要错误分析错误原因，然后制定解决方案，针对不明原因的错误则收集错误发生时的更多环境信息，发布版本后再进行分析 3) 提升开发效率的诀窍：测试流程一键化，可迅速验证修改的代码，智能代码提示，无须查阅文档，分级日志和分色显示日志工具快速定位错误原因
项目职责	主导云端方案的设计和实现

■项目目经历 E:

项目名称	底层 Recovery 定制及底层技术研究
项目时间	2013年8月~2014年3月
项目职务	Android 底层开发工程师

项目简介	<p>在 CM Recovery 源码基础上定制刷机精灵的 Recovery，定制刷机，系统备份，系统还原，清除数据，清除缓存等界面。优化并精简 CM recovery 的功能，定制刷机精灵需要的功能。改造了 Android 编译系统，支持多分辨率适配，可直接编译生成 MTK 机型的 Recovery。</p> <p>技术要点：</p> <ol style="list-style-type: none"> 1) 使用 nx 连接服务器，在服务器上开发，使用 secure crt 连接到开发服务器，进行编译，本地电脑安装虚拟机，手机连接到本地电脑的虚拟机，每次修改 Recovery 源码后，可一键编译并测试新修改的逻辑是否有问题 2) 改造 CM 的编译系统，使其支持 Recovery 的多分辨率适配，支持生成 MTK 机型的 Recovery，支持小米手机的双系统切换，还添加了华为和 Htc 的充电模式检测。另外还编写了用于提取 Device 配置文件的 Shell 脚本，从手机里提取 Device 后，只需对配置文件做少量修改即可生成 Recovery，提高了适配 Recovery 的效率 3) 还实现了在 Recovery 模式下不重启就能直接刷机，还能汇报刷机状态给 PC 刷机精灵 4) 深入研究了 Recovery 源码，init 源码，adb 源码，SELinux 机制
项目职责	编译系统改造，Recovery 开发，底层源码研究

■项目目经历 F:

项目名称	手机版刷机精灵
项目时间	2012 年 11 月~2013 年 4 月
项目职务	Android 工程师
项目简介	<p>刷机精灵手机端的第 1 个版本，可查看 ROM 列表，ROM 详情及评论，用户还可以下载 ROM 并刷机，但该项目并未上线。</p> <p>技术要点：</p> <ol style="list-style-type: none"> 1) 软件架构采用 MVC 模式，将作为 activity 的 controller 放在同一个包，自定义 view 放在同一个包。业务逻辑按功能分类放在不同的包，所有请求服务器的 json 消息放在同一个包。图片缓存相关类放在同一个包，网络请求放在同一个包，刷机方案放在同一个包，下载引擎放在同一个包。 2) tab 内容 Activity 采用模版方法模式设计，这些 Activity 继承自同一个父 Activity，父 Activity 固定了如何生成整个 tab 页内容的算法，也就是说将标题栏和内容组合起来，子 Activity 类生成标题栏时，只需告知父 Activity 要显示的文字即可。这种设计减少了代码的重复，具有更好的可维护性 3) 刷机方案的设计采用了模版方法模式和观察者模式，因为刷机步骤是固定的，只是某些具体的刷机步骤的实现会所有不同，所以采用模版方法。另外还设计了一个简单工厂，根据服务器下发的刷机方案，选择不同的刷机方案实现进行刷机。刷机方案执行时通过观察者模式通知界面层 4) 图片缓存采用二级缓存，有文件缓存和内存缓存。先从网络把图片下载到本地，再将图片从本地加载到内存，内存对加载的图片所占用的总大小有限制，如果图片大小超过限制会使用 LRU 算法将图片从内存中移除。实现图片的

	<p>内存缓存采用 Map，并将图片名字，尺寸等参数作为 key，而 Bitmap 作为 value，从 Map 里移除某个 Bitmap 后，垃圾回收机制会自动回收 Bitmap 所使用的内存，并不需要调用 system.gc()，如果担心不能及时回收的话，可用 bitmap 的 recycle 方法</p> <p>5) 可局部更新下载列表界面。更新 ListView 时，如果调用 notifydatasetchanged 会使得整个界面重绘，看起来像是滑动了界面一样。局部更新只需要在 getView 方法里使用 map 缓存位置即可，再用 AdapterView 的 getFirstVisiblePosition 得到位置，再计算出最终位置，再用 ViewGroup 的 getChildAt(position)即可得到任务所在的 view，然后操作 view 的数据进行局部更新</p>
项目职责	独立完成软件设计，并编写核心代码

■项目目经历 G:

项目名称	刷机精灵助手
项目时间	2011 年 12 月~2013 年 5 月
项目职务	Android 工程师
项目简介	<p>刷机精灵助手是 PC 端刷机精灵的辅助软件，运行在手机端，协助 PC 端刷机精灵完成联系人，短信，通话记录，应用数据备份以及还原功能，调用手机管家实现手机体检功能，提供 ftp 服务和检测系统更新等功能。</p> <p>技术要点:</p> <ol style="list-style-type: none"> 1) 采用 MVC 架构，只有一个简单的界面作为 View,还有一个 Activity 作为控制层，业务模型比较复杂，既要实现和 PC 端刷机精灵通信，还要实现支持的 PC 端的命令逻辑。如果和 PC 端刷机精灵连接上了，将会让界面显示已连接，如果正在执行 PC 端刷机精灵的命令，则让界面显示正在执行命令 2) 采用 TCP 和 PC 客户端通信，TCP Server 在某个端口监听，PC 端刷机精灵通过 adb 转发端口然后连接该端口，从而建立连接。精灵助手为每个连接建立一个命令队列，每个命令都在不同的线程执行，并有看门狗线程确保命令执行不超时 3) 资料获取辅助类采用单例模式，针对 2.2 以下版本和 2.2 以上版本有不同的子类实现，通过反射实现单例 4) 实现各种命令采用模版方法模式，设计了抽象类 ClientCommand，在该类里固定了命令执行的算法，先调用执行时间统计的初始化操作，启动看门狗线程，最后统计命令执行时间。 5) 添加新的命令，不需要修改现有代码。在解析通信协议生成命令对象时，采用反射机制，生成需要执行的命令的对象，所以不需要修改现有代码。真正实现了对修改封闭，对扩展开放
项目职责	项目初期独立完成软件设计及实现，上线后也负责维护并添加新功能。另外为了绕过授权管理软件，我还写实现了一个 su 程序，协助精灵助手获取 Root 权限