

短视频段子 -- 竞品分析

选取最右、白色不得姐和糗事百科作为竞品分析。技术上分析，主要集中在竞品用到的相关框架和其优缺点

竞品 APP 基本情况

功能名称	最右	百思不得姐	糗事百科
version	4.1.8.9	6.9.2	10.19.2
minSdk	16	14	14
targetSdk	27	20	21
methods	10W	9W	10W
多 dex	MultiDex	MultiDex	MultiDex
热修复	支持 Tinker	不支持	不支持

竞品技术选型情况

功能名称	最右	百思不得姐	糗事百科
图片加载	Fresco	Fresco & Glide & UIL	Fresco
图片格式	Webp	jpeg	jpeg & webp
gif	支持 Fresco	支持 Fresco	支持 Android-gif-drawable & Fresco
视频播放	ijkplayer	自行实现	自行实现
图片裁剪	支持 uCrop CropImageView	支持 cropper	支持 android-crop
网络框架	Okhttp & Retrofit	Okhttp & Retrofit	Okhttp & Retrofit & Volley
多线程	RxJava	RxJava	RxJava
事件总线	EventBus	EventBus	EventBus
滤镜美肤	商汤 SDK	GPUImage	不支持
下拉刷新	SmartRefresh	Android-PullToRefresh & 自行封装实现	android-Ultra-Pull-To-Refresh
推送	集成多家推送 小米魅族等	集成 TrustData ， 多家厂商推送	集成 TrustData ， 多家厂商推送
下载库	英语流利说 filedownloader	自行实现	英语流利说 filedownloader
序列化	fastjson	gson	gson & jackson

各框架的优缺点

gif 方面

- [Android-gif-drawable](#)

- 优点：占用内存少，CPU 占用低。
- 缺点：仅仅支持加载本地 gif 文件。如需加载网络 gif，需要自行拓展。

- Fresco

- 优点：支持网络和本地 gif。
- 缺点：占用内存和CPU，相对高。

- Glide

- 优点：支持网络和本地 gif。
- 缺点：占用内存和CPU，相对高，比 Fresco，CPU 还要略高。缓存模式上，需要缓存原图，加载速度才能快点。

图片格式

- webp

- 优点：大部分情况下，压缩后体积小。
- 缺点：
 - 4.2 才支持完整的 webp 特性。4.0 ~ 4.2 不支持透明度。
 - 图片解码时间，相对长点。
 - 部分图片压缩后体积，不减反增。

- jpg

- 优点：体积相对小，颜色变化不是十分大的情况下，并且无透明度的情况下，相对 png，体积小点。
- 缺点：不支持透明度，有损压缩。

- png

- 优点：支持透明度，支持无损压缩
- 缺点：体积相对大点。

图片加载库

- Glide

- 优点
 - 带生命周期回调
 - 缓存实际显示尺寸
 - 支持 Gif
 - 库体积，相对小

- Fresco

- 优点
 - 在 5.0 机器，内存缓存更加出色。

- 对 Webp 解码支持，即使在旧版本Android上，也可以使用。
- 支持渐进式 JPEG 图片格式。
- 支持 Gif
- 缺点
 - 库整体偏大，偏重。
 - API，不是很友好。
 - 相对 Glide，没有生命周期。

Lottie 动画库

大致原理是，AE 制作动画，并且用Bodymovin 导出的规则json，客户端则解析 json，并可以生成相关的动画。

- 优点
 - 跨平台，支持 Android, iOS 和 React Native
 - 原生实现，仅仅是 json 表达动画，由原生解析，并且生成动画规则并执行。
 - 支持动态下发，json，来实现服务端控制动画。
- 缺点
 - 解析 json，并且生成相关的对象，这里，相对耗时间。如果在列表当中做复杂的动画，有一定的性能损耗需要注意。
 - 部分动画，如 Masks 和 mattes，在 Android 平台上，有一定的性能损耗。

多线程

RxJava

- 优点
 - 链式API，切换线程方便。
 - 把一个完整的任务，切分成不同的小模块，各自模块有对应的上下文。
 - 响应式编程，实现的业务逻辑，简洁，阅读性高。
- 缺点
 - 上手有一定的门槛。
 - 有内存泄露风险。

序列化

- Gson
 - 优点: 无依赖，可以直接跑在 JDK 上，Google 维护。支持复杂的类型。功能齐全。
 - 缺点: 需要有 get set 方法。性能比 FastJson 差点。
- FastJson

- 优点：性能好，第三方测试数据是目前最快的，[第三方测试数据](#)
- 缺点：在复杂类型的 Bean 上，可能会出现转换错误。Issues 处理略慢。
- Jackson
 - 优点：开源，社区维护积极
 - 缺点：复杂类型，转换会异常。体积偏大 1000k+

下拉刷新框架

- [Android-PullToRefresh](#)
 - 优点 谷歌 Chris Banes 开发维护，早期流行下拉刷新库之一。
 - 缺点 受限之前 jar 包，不可以带资源等因素影响，功能相对有限，并且已经 DEPRECATED 多年
- [android-Ultra-Pull-To-Refresh](#)
 - 优点 代码可配置相对高，模块职责明确
 - 缺点 项目3年没维护了，遗留了一些 bug 。也缺少了支持嵌套滚动等新的 Android feature
- [SmartRefreshLayout](#)
 - 优点 可配置高，代码模块职责清晰，支持 NestedScroll 等新 feature. 社区维护积极

热修复

Tinker，腾讯微信开源，并且微信在用。其实算是冷修复，需要重启进程之后，才生效。

- 优点
 - 自研 diff 算法，差异宝很小。
 - 数亿用户验证，相对可靠。
- 缺点
 - 合成的时候，内存占用偏大。
 - 嵌入成本，略高，需要修改相关项目代码，如 Application。

图片裁剪

- [uCrop](#)
 - 优点:
 - 支持功能齐全,支持缩放、旋转、支持手势旋转等
 - 性能相对好，裁剪部分C层实现。
 - 社区维护积极，迭代快。
 - 缺点:

- [cropper](#)
 - 优点：纯 Java 层实现，兼容性好。
 - 缺点：多年没更新维护了，积累 Issue 多。
- [android-crop](#)
 - 优点：基于 AOSP 代码。
 - 缺点：3年多不维护了，Deprecated 状态。

事件总线

EventBus 的使用：

- 2.X 版本，是基于运行时，通过反射收集相关注册信息，发布事件的时间，也是通过反射调用相关的注册方法。因此，性能相对有所影响。
- 2.X 版本，支持编译期间，通过 APT 收集相关的索引信息，因此性能相对较高。
- 优点：
 - API 友好，比广播使用方便。
 - 对解耦，有较大帮助作用。
- 缺点：
 - 不支持跨进程
 - 滥用的情况下，代码逻辑跳跃大，调用链不清晰，因而项目代码阅读性会急速下降。
 - 也容易形成死循环。
- 使用规范建议：
 - 在基础的业务场景下使用，例如登陆登出、网络连接变化等，
 - 页面间，传递数据，一般不建议使用。
 - 没有直接互相引用的对象，谨慎考虑使用，团队内 review 代码。

ixintui 推送

- 优点：
 - 除自有推送通道外，还集成了小米、华为、魅族厂商推送，可自主选择是否开启使用厂商推送
 - 支持 Android 和 iOS
 - 糗事百科 BOSS直聘 蜻蜓FM 图吧导航 百思不得姐 等都在使用
 - 厂商推送采用动态下发的模式，只有对应品牌设备才会下发和加载，真正做到了SDK“瘦身”。

图片滤镜

百思不得姐 采用了 [GPUImage](#) 来实现 图片滤镜功能，基于 OpenGL ES 2.0 实现。我们如果有类似需要，也可以采用该库， 而最右，则采用了商汤科技SDK来实现的相关的滤镜美肤需求。