

工业级App的技术选型 和标准

字节跳动-Helo项目： 易华琦/黄晓宇

目标

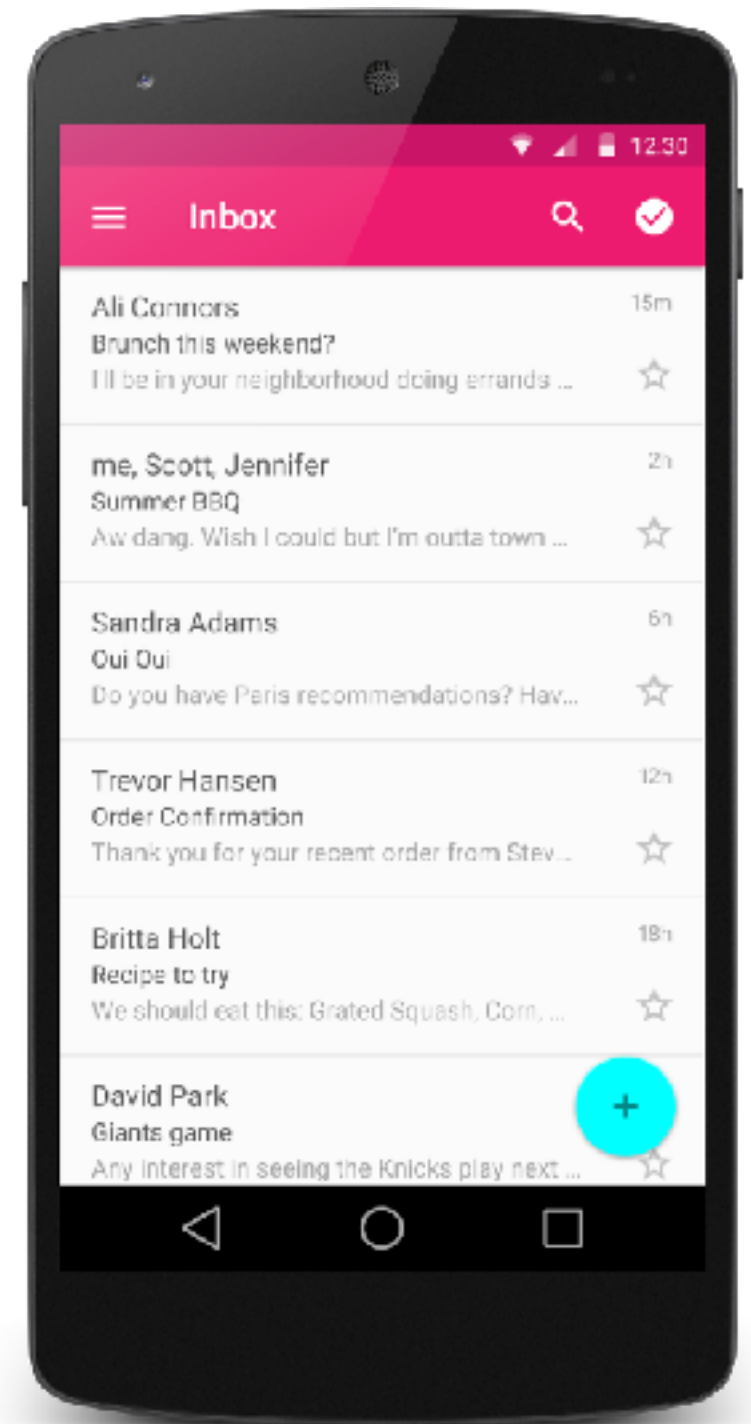
- 了解工业级App的技术选型以及质量标准。
- 即使对于简单的项目，也能有针对性的进行技术上的深入研究。
- 将来面试的时候，能够有干货可以讲。

Layout

- 不要再使用LinearLayout、FrameLayout、RelativeLayout了。
- ConstraintLayout: 一层完成布局，不需要嵌套。
- CoordinatorLayout: 酷炫的动画效果。
- merge, include, viewstub标签。

列表

- 列表组件： ListView, RecyclerView
- 深入RecyclerView：
 - DiffUtil 工具
 - ViewHolder生命周期
 - ViewHolder缓存原理
 - 为什么优于ListView?

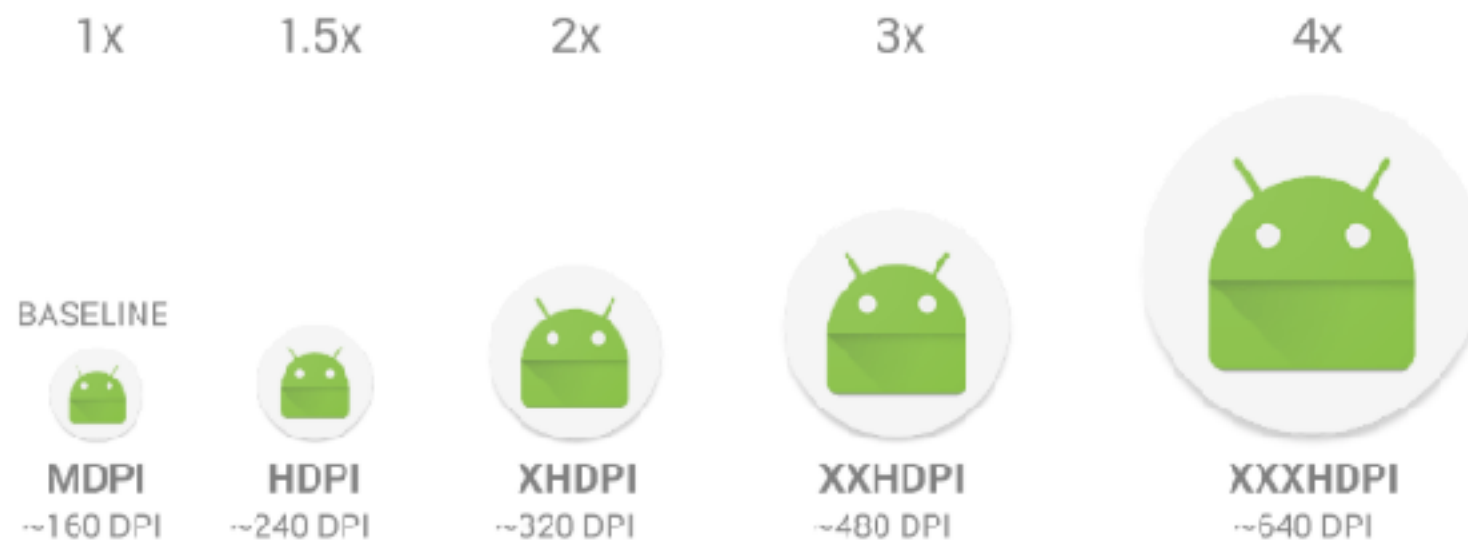


存储

- DataBase:
 - SQLite
 - 数据库migration
 - 如何设计一个具有扩展性的表
- Room: 更好用的数据库组件
- SharedPreferences: 轻量级数据持久化组件
 - 适用场景
 - 实现原理
 - 性能

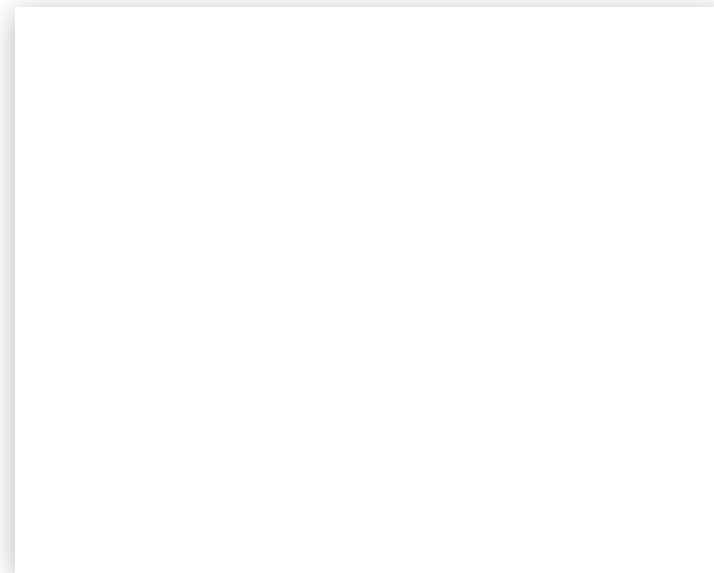
图片

- 图片库： Glide、 Fresco
- 图片库多级缓存设计
- 分辨率适配： xhdpi, xxhdpi, xxxhdpi



图片

- 常见图片格式：PNG, Baseline vs Progressive JPEG, Webp, HEIF, RGB565 vs 8888
- IconFont和Vector [Vector资源](#)

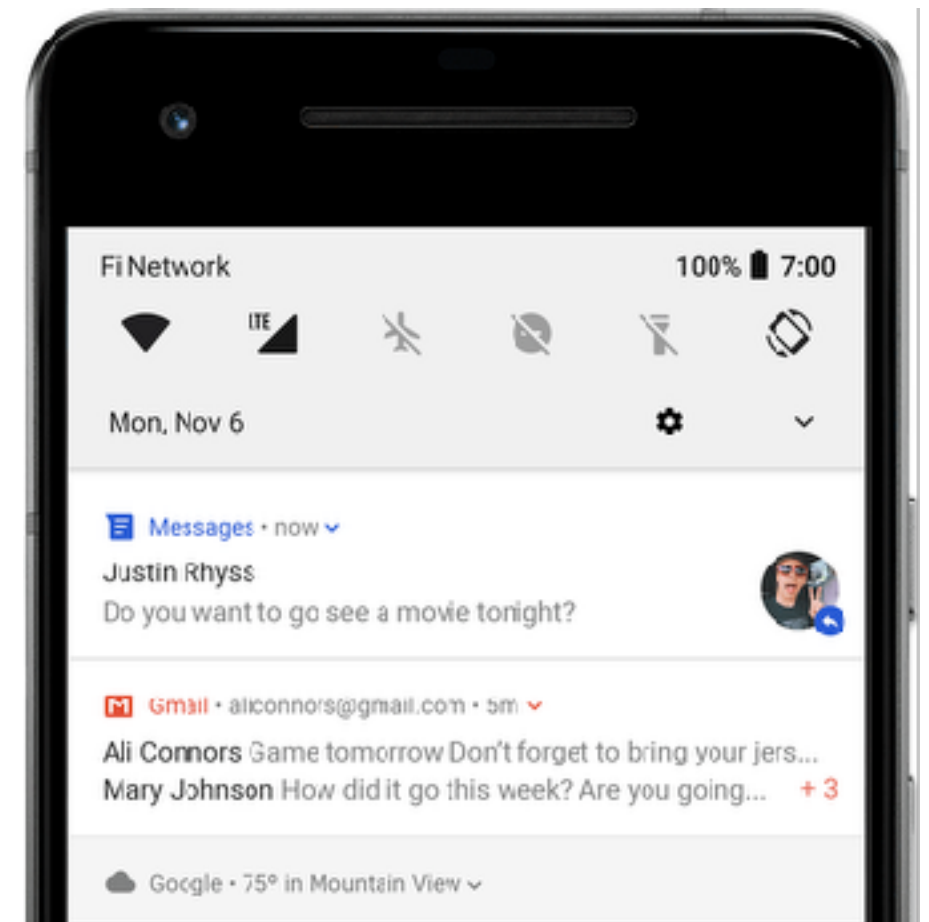


Push

- 什么是推送 (Push)
- 一个Push模块的实现: Notification + Service
- 耗时操作: 网络请求, 单独起一个service
- Push与多进程: 把Push放到别的进程的好处

Notification/Service

- Service种类： 前台 vs 后台
- Notification:
 - 自定义样式： RemoteView
 - 状态更新， 通知跳转
- 常驻Notification



网络

- Retrofit/OkHttp



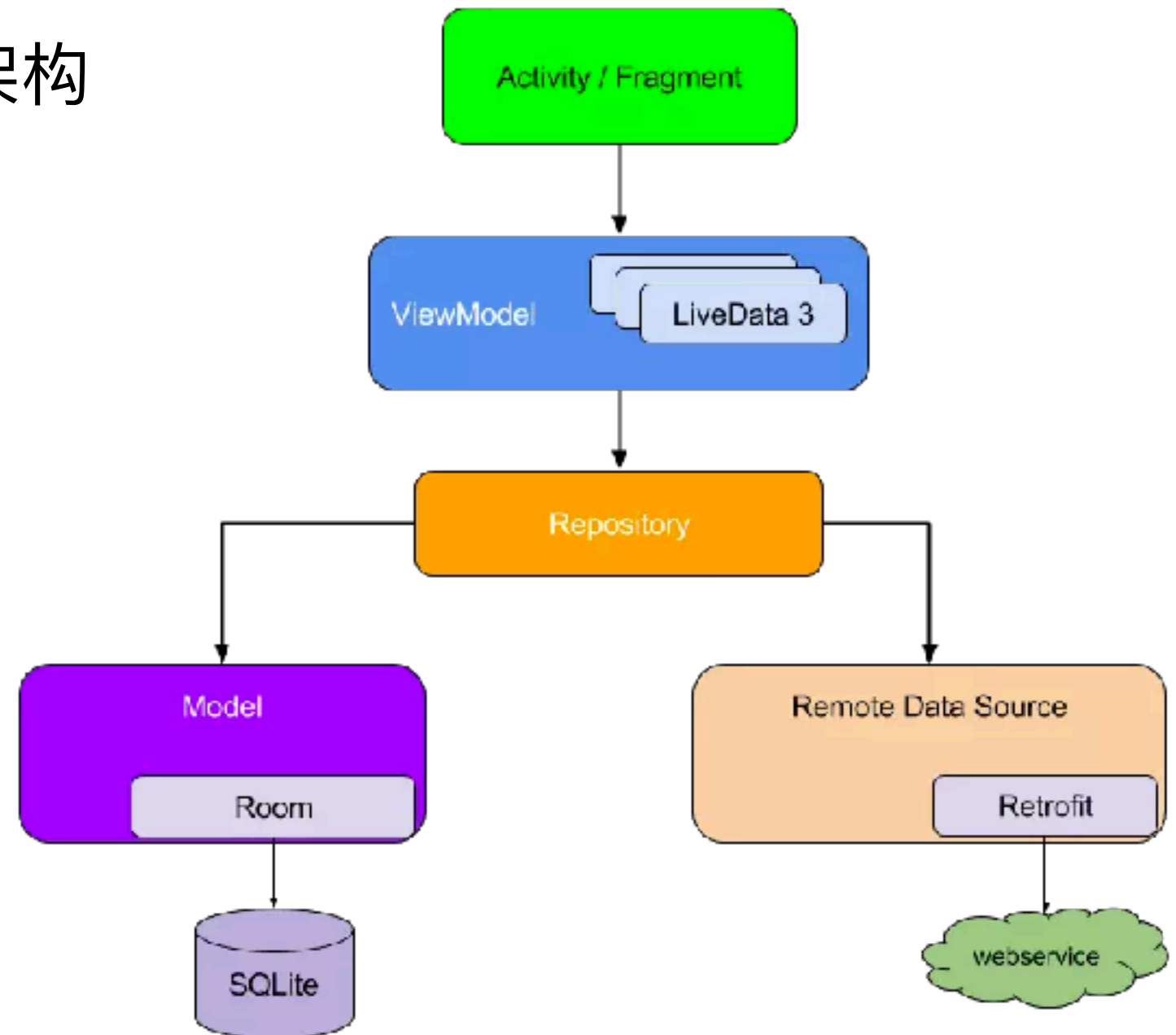
- OkHttp设计: interceptor责任链模式, 网络监控, 通用参数

- 提升网络性能:

- HTTP1.1, HTTP2.0, QUIC
- Keep-alive/gzip
- HTTPS, TLS1.2/TLS1.3
- 请求合并
- JSON vs PB

架构

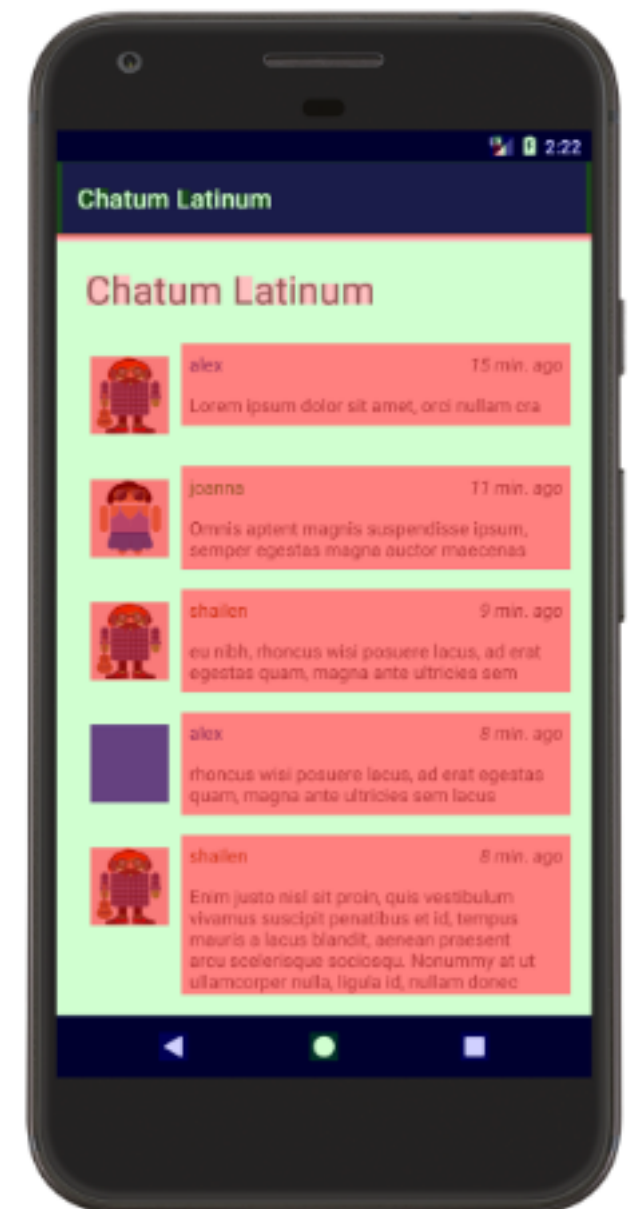
- MVC：基本等于没有架构
- MVP：复杂
- MVVM：现代架构
 - LiveData
 - ViewModel



帧率

- Layout层级。onLayout, onMeasure的开销
- Overdraw。透明背景，降低层级。
- onBindViewHolder做轻量级的事情。
- 开发者选项帮你Debug。
- LayoutInspector。

GPU Overdraw enabled

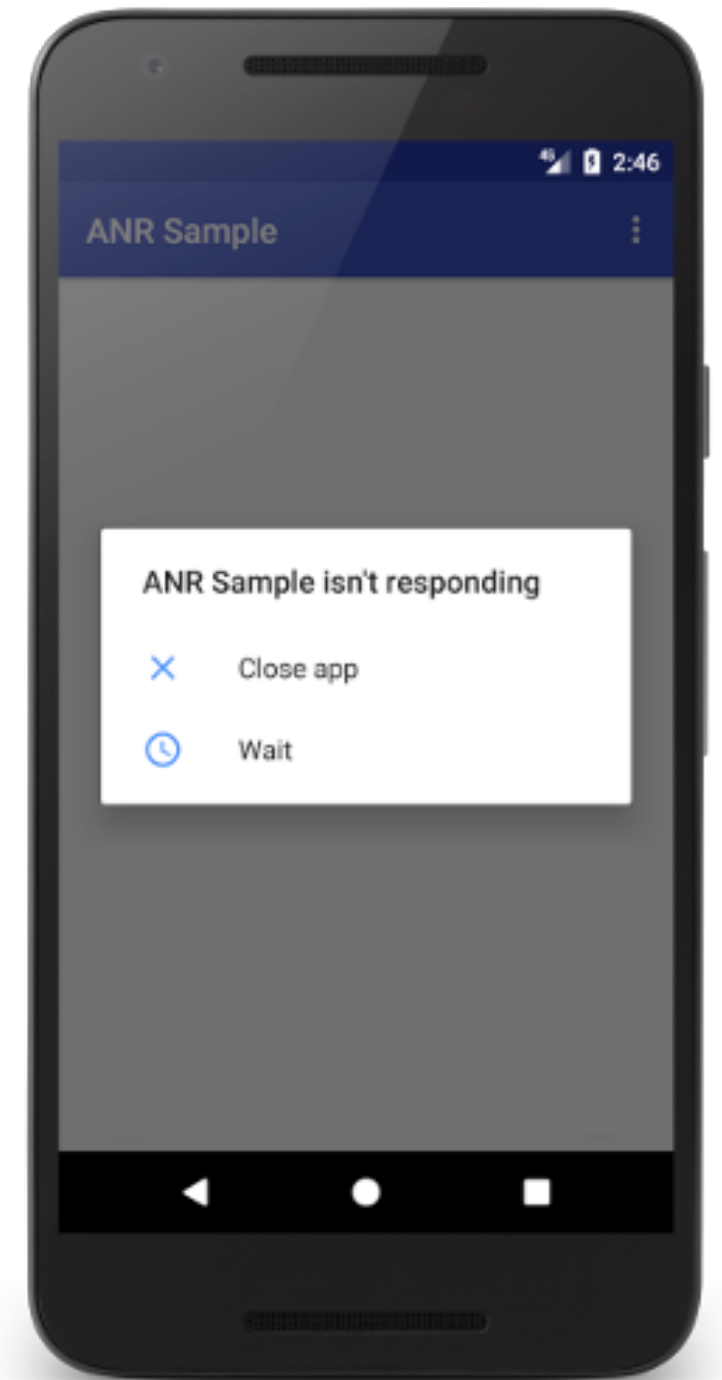


内存

- 内存泄漏：
 - 定义
 - 常见内存泄漏场景
- LeakCanary：工作原理？ WeakReference。ReferenceQueue。
- Profiler查看大内存？
- 图片分辨率。

启动速度/卡顿/ANR

- ANR的原理?
- 如何检测ANR?
- 不要在主线程做太多事。
- IO放在IO线程。
- 能懒加载的懒加载。
- HandlerThread, RxJava、Coroutine



无网/弱网体验

- 无网：本地功能需要尽量可用。在线功能提供缓存。操作迅速提示用户无网，不要一直loading。
- 弱网：酷炫的loading？非主要请求不阻塞界面。按块加载。

优雅的状态恢复

- Don't keep activity: 测试状态恢复的最好debug 选项。
- onSaveInstanceState的作用。
- fragment在activity重建的时候需要检测重复创建。
- fragmentManager和childFragmentManager的区别。

Q & A