

# Danale新方案

---

## 一、Danale当前存在的问题

当前Danale是采用纯原生开发，项目目前是较稳定的。但是app中可能会存在问题，需要发布紧急版本，然而发布版本有很多不确定的因素，可能会导致版本更新时间较久，不能及时解决问题。

对于上面提出的问题，在android上有热更新的方案，但是随着android版本的不断更新，热更新方案存在诸多不确定性。

iOS版本app在热更新方面纯原生可能没有很好的方案。

同时，开发者对于原型的理解可能有或多或少的差异，导致android和iOS应用在界面上会存在一些差异。

总结：

1. 热更新不易
2. 界面差异明显
3. 重复开发
4. 开发需要维护android和ios两套完全代码
5. 模块耦合大
6. 开发编译时间长，不支持热重载

## 二、方案研究

针对android和iOS在热更新和界面统一上的问题，我们针对现有的一些方案，简单介绍。

### 1. 原生开发

这里列出sophix，代替了一类android原生热更新的方案，分别针对java代码、图片文字资源以及静态库去动态替换的方案。比较成熟，但是也有较多的限制，这里不展开讲述，想了解的可以去看看相关文档资料。

优点：

- a. 可控性大

缺点：

- a. 不能解决一中提到的问题。

### 2. react native

React Native是facebook开源的开发框架，目前存在一些性能上的问题，facebook也在积极去更新。未来应该是一个不错的框架。

React Native可以去解决一些可热更新去处理的bug，或者热更新Rn模块中的功能。

经过一个阶段的react native开发，发现在列表使用时会存在刷新列表空白的问题。并且rn开发界面效率更高。

优点：

- a. 热更新成熟
- b. android和ios公用一套代码，避免重复开发
- c. 界面开发更快速
- d. 维护成本低
- e. 热重载

缺点：

- a. 开发时适配工作量大（android和ios差异）
- b. 渲染性能稍差

### 3. flutter

google开源的移动UI框架，可以快速在iOS和android上构建app。

从长远的角度来看，flutter是更合适的一种方案

优点：

- a. 渲染性能好，直接进行渲染（RN先转为原生，再去渲染）
- b. android和ios的适配性好
- c. 热重载

但是目前来说，有以下的

缺点：

- a. 需要学习成本
- b. 热更新方案不确定
- c. 三方库支持少

## 三、Danale新方案设计

基于上面的优缺点，考虑React Native目前大部分人已经了解了该框架，相对于原生开发带来了诸多的好处，选择使用Rn开发部分功能模块。

由于视频播放模块，工作量比较大，功能复杂，考虑暂时先接入目前原生中的模块，后面时间允许，逐步去替换掉。

所以新方案将采用React Native和原生共同开发的方式。但是该方案初期会导致安装包较大。（原生和rn两套代码）

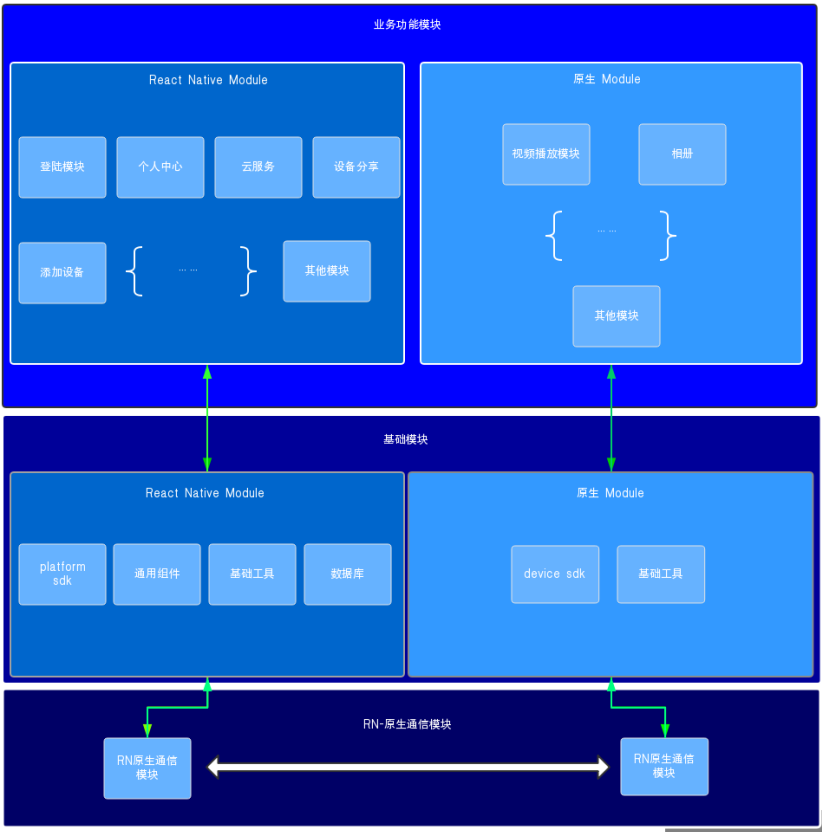
以React Native项目为基础，接入原生视频播放和相册等功能。具体的设计模块如下：

设备库sdk和视频播放模块采用原生开发，

平台sdk和登陆、个人中心、云服务、设备分享、添加设备等模块使用rn开发，以便及时动态更新。

本次开发计划使用组件化开发，每个功能模块是一个单独的组件，各个功能模块相互解耦，仅依赖基础框架。组件之间交互可采用传参方式。

具体方案如下图：



开发计划如下：

1. 新Danale项目基于当前开发的React native项目，进行sdk重构、项目重构、模块整理，完成项目的基础框架
2. 在此基础上接入目前Danale项目，选择性使用Danale项目中的部分功能模块。
3. 一期可以使用Rn替换部分简单功能模块，后面时间允许，逐步去替换掉。

关注：

*rn 和原生隔离开发*

*各组件之间的通信协议指定*