

## 1.1 跨平台技术简介

针对原生开发面临问题，人们一直都在努力寻找好的解决方案，而时至今日，已经有很多跨平台框架(注意，本书中所指的“跨平台”若无特殊说明，即特指Android和iOS两个平台)，根据其原理，主要分为三类：

- H5+原生 (Cordova、Ionic、微信小程序)
- JavaScript开发+原生渲染 (React Native、Weex、快应用)
- 自绘UI+原生(QT for mobile、Flutter)

在接下来的章节中我们逐个来看看这三类框架的原理及优缺点。

## 1.2 Hybrid技术简介

### H5+原生混合开发

这类框架主要原理就是将APP的一部分需要动态变动的内容通过H5来实现，通过原生的网页加载控件WebView (Android)或WKWebView (iOS) 来加载（以后若无特殊说明，我们用WebView来统一指代android和iOS中的网页加载控件）。这样以来，H5部分是可以随时改变而不用发版，动态化需求能满足；同时，由于h5代码只需要一次开发，就能同时在Android和iOS两个平台运行，这也可以减小开发成本，也就是说，H5部分功能越多，开发成本就越小。我们称这种h5+原生的开发模式为**混合开发**，采用混合模式开发的APP我们称之为**混合应用**或**Hybrid APP**，如果一个应用的大多数功能都是H5实现的话，我们称其为**Web APP**。

目前混合开发框架的典型代表有：Cordova、Ionic 和微信小程序，值得一提的是微信小程序目前是在webview中渲染的，并非原生渲染，但将来有可能会采用原生渲染。

### 混合开发技术点

如之前所述，原生开发可以访问平台所有功能，而混合开发中，H5代码是运行在WebView中，而WebView实质上就是一个浏览器内核，其JavaScript依然运行在一个权限受限的沙箱中，所以对于大多数系统能力都没有访问权限，如无法访问文件系统、不能使用蓝牙等。所以，对于H5不能实现的功能，都需要原生去做。而混合框架一般都会在原生代码中预先实现一些访问系统能力的API，然后暴露给WebView以供JavaScript调用，这样一来，WebView就成为了JavaScript与原生API之间通信的桥梁，主要负责JavaScript与原生之间传递调用消息，而消息的传递必须遵守一个标准的协议，它规定了消息的格式与含义，我们把依赖于WebView的用于在JavaScript与原生之间通信并实现了某种消息传输协议的工具称之为**WebView JavaScript Bridge**，简称**JsBridge**，它也是混合开发框架的核心。

### 总结

混合应用的优点是动态内容是H5，web技术栈，社区及资源丰富，缺点是性能不好，对于复杂用户界面或动画，WebView不堪重任。

## 1.3 React Native、Weex及快应用

本篇主要介绍一下 **JavaScript开发+原生渲染**的跨平台框架原理。

React Native (简称RN)是Facebook于2015年4月开源的跨平台移动应用开发框架，是Facebook早先开源的JS框架 React 在原生移动应用平台的衍生产物，目前支持iOS和Android两个平台。RN使用 Javascript语言，类似于HTML的JSX，以及CSS来开发移动应用，因此熟悉Web前端开发的技术人员只需很少的学习就可以进入移动应用开发领域。

由于RN和React原理相通，并且Flutter也是受React启发，很多思想也都是相通的，万丈高楼平地起，我们有必要深入了解一下React原理。React是一个响应式的Web框架，我们先了解一下两个重要的概念：DOM树与响应式编程。

## 总结

JavaScript开发+原生渲染的方式主要优点如下：

- 1. 采用Web开发技术栈，社区庞大、上手快、开发成本相对较低。
- 2. 原生渲染，性能相比H5提高很多。
- 3. 动态化较好，支持热更新。

不足：

- 1. 渲染时需要JavaScript和原生之间通信，在有些场景如拖动可能会因为通信频繁导致卡顿。
- 2. JavaScript为脚本语言，执行时需要JIT(Just In Time)，执行效率和AOT(Ahead Of Time)代码仍有差距。
- 3. 由于渲染依赖原生控件，不同平台的控件需要单独维护，并且当系统更新时，社区控件可能会滞后；除此之外，其控件系统也会受到原生UI系统限制，例如，在Android中，手势冲突消歧规则是固定的，这在使用不同人写的控件嵌套时，手势冲突问题将会变得非常棘手。

## 1.4Flutter简介

### 跨平台自绘引擎

Flutter与用于构建移动应用程序的其它大多数框架不同，因为Flutter既不使用WebView，也不使用操作系统的原生控件。相反，Flutter使用自己的高性能渲染引擎来绘制widget。这样不仅可以保证在Android和iOS上UI的一致性，而且也可以避免对原生控件依赖而带来的限制及高昂的维护成本。

Flutter使用Skia作为其2D渲染引擎，Skia是Google的一个2D图形处理函数库，包含字型、坐标转换，以及点阵图都有高效能且简洁的表现，Skia是跨平台的，并提供了非常友好的API，目前Google Chrome浏览器和Android均采用Skia作为其绘图引擎。

目前Flutter默认支持iOS、Android、Fuchsia（Google新的自研操作系统）三个移动平台。但Flutter亦可支持Web开发（Flutter for web）和PC开发，本书的示例和介绍主要是基于iOS和Android平台的，其它平台读者可以自行了解。

## 1.5 小结

本章主要介绍了目前移动开发中三种跨平台技术，现在我们从框架角度对比一下它们，如表1-1所示：

技术类型	UI渲染方式	性能	开发效率	动态化	框架代表
H5+原生	WebView渲染	一般	高	支持	Cordova、Ionic
JavaScript+原生渲染	原生控件渲染	好	中	支持	RN、Weex
自绘UI+原生	调用系统API渲染	好	Flutter高, QT低	默认不支持	QT、Flutter

flutter和其他框架对比详情：[https://book.flutterchina.club/chapter1/mobile\\_development\\_intro.html](https://book.flutterchina.club/chapter1/mobile_development_intro.html)