

Taro

[《小程序跨框架开发的探索与实践》演讲全文](#)

[选型参考](#)

一、

二、

[架构](#)

[3之前的:](#)

[3之后的](#)

[区别:](#)

[mpvue](#)

[Vue支持](#)

[Taro小程序](#)

[Taro DOM Tree 如何更新到页面](#)

[React](#)

[Vue](#)

[更新](#)

[Taro NPM](#)

希望能够使用 React 语法写小程序的同时，通过「**Write once Run anywhere**」来实现跨端的。

[《小程序跨框架开发的探索与实践》演讲全文](#)

选型参考

一、

	微信小程序	mpvue	wepy	Taro
语法规范	小程序规范	vue.js 规范	类 vue.js 规范	React 规范
模板系统	字符串模板	字符串模板	字符串模板	JSX
类型系统	不支持	业务代码	业务代码	业务代码 + JSX 模板
组件规范	小程序组件	html 标签 + 小程序组件	小程序组件	小程序组件
样式规范	wxss	sass, less, postcss	sass, less, styus	sass, less, postcss
组件化	小程序组件化	vue 组件化规范	自定义组件化	React 组件化规范
多端复用	无	复用为 H5 *	复用为 H5	复用为 H5 *
自动构建	无	Webpack 构建	内建构建系统	内建构建系统 + Webpack
上手成本	全新学习	熟悉 Vue.js	熟悉 Vue.js + wepy	熟悉 React
数据流管理	不支持	Vuex	Redux	Redux

二、

		chameleon	mpvue	Taro	uni-app	WePY
开发工具	DSL	类 Vue	Vue	React	Vue	类 Vue
	IDE/图形化开发工具	无	无	无	有	无
	语法校验工具	自研	无	ESLint 规则	IDE 支持	无
	TypeScript	无	有	有	有	有
	Typing/自动补全	无	API	API + JSX	IDE 支持	无
多端支持	样式	sass, less, stylus	sass, less, stylus	sass, less, stylus, CSS Modules	sass, less, stylus	sass, less, stylus
	小程序支持	微信、百度、支付宝	微信、百度、支付宝、字节跳动	微信、百度、支付宝、字节跳动	微信、百度、支付宝、字节跳动	微信、百度、支付宝
	移动端容器	Weex	无	React Native	Weex	无
	移动端增强	chameleon SDK	无	Expo	nvue	无
	多端编译方式	自研多态协议	环境变量条件编译	环境变量 + 文件条件编译	自研条件编译语法	环境变量条件编译
	H5 兼容 API	自研多态协议	无	有	有	无
组件库/工具库/demo	跨端组件库	有	无	有	有	无
	第三方组件	较少	丰富	较丰富	丰富	丰富
	第三方工具库	较少	较丰富	较丰富	丰富	丰富
	Demo	较少	较丰富	较丰富	较丰富	丰富
	状态管理工具	Vuex	Vuex	Redux/MobX/Dva	Vuex	Redux
	转换微信小程序工具	无	无	有	无	有
流行度	自研组件库	有	无	有	有	无
	GitHub Star	3843	16281	16084	2921	16779
	GitHub Issue/PR	68/8	1369/104	2026/368	215/18	1662/441
	NPM + CNPM 下载量 ¹	241/周	1971/周	4332/周	N/A	976/周
	案例	较少	丰富	丰富	非常丰富	丰富
	开发者人数 ²	~1000 人	~ 5000 人	~ 5000 人	10000+	~ 5000 人
	自建开发者社区	无	无	无	有	无

¹ 数据采集于 2019 年 3 月 11 日，以各框架目前稳定版特性为标准

² 统计上周各框架 CLI 工具下载量，uni-app 可用 IDE 工具直接开发

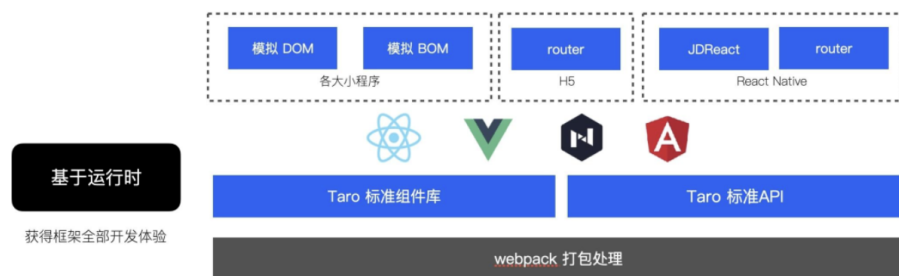
架构

3之前的：

Taro 当前架构



3之后的



区别：

重编译和重运行时，而重运行时带来的好处是，模拟DOM/BOM API：

新的架构基本解决了之前的遗留问题：

- **无 DSL 限制**：无论是你们团队是 React 还是 Vue 技术栈，都能够使用 Taro 开发
- **模版动态构建**：和之前模版通过编译生成的不同，Taro Next 的模版是固定的，然后基于组件的 template，动态“递归”渲染整棵 Taro DOM 树。
- **新特性无缝支持**：由于 Taro Next 本质上是将 React/Vue 运行在小程序上，因此，各种新特性也就无缝支持了。
- **社区贡献更简单**：错误栈将和 React/Vue 一致，团队只需要维护核心的 taro-runtime。
- **基于 Webpack**：Taro Next 基于 Webpack 实现了多端的工程化，提供了插件功能。

mpvue

mpvue 本质上还是将 Vue 运行在了小程序，在platform目录下实现了小程序的转化。

Vue支持

- 从 `v3.0.0` 版本（01 Jul 2020）支持的vue3开发

Taro小程序

1. 先基于cli为react/vue代码用webpack打包
2. 运行时使用适配器进行适配，调用Taro实现的DOM/BOM API，
3. 渲染到小程序页面

Taro DOM Tree 如何更新到页面

React

1. 先将小程序的组件模版化
2. 将Taro Runtime 生成的Taro Dom Tree，去匹配模版
3. 匹配遍历完Taro Tree之后，得到的就是小程序的组件Tree

React 实现流程图



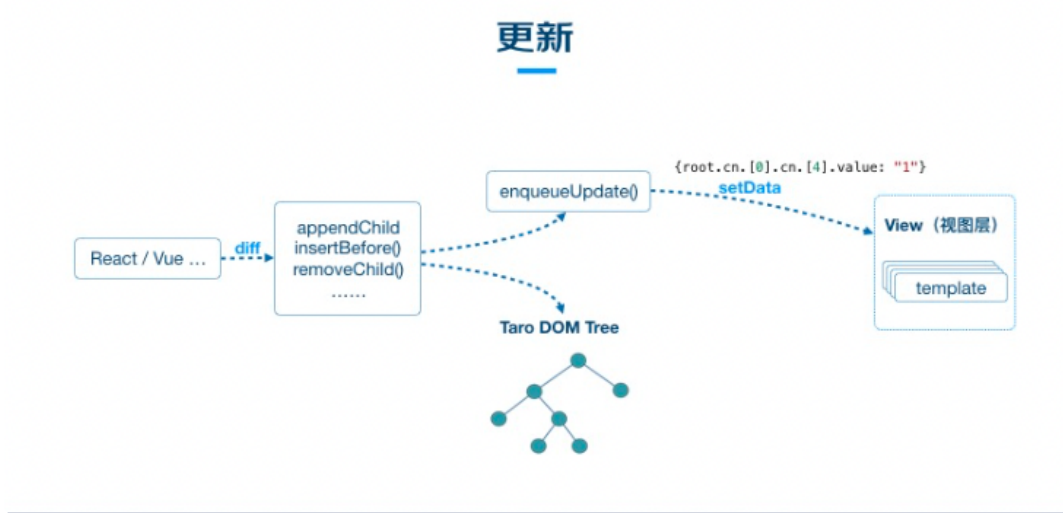
Vue

Vue 和 React 最大的区别就在于运行时的 CreateVuePage 方法，这个方法里进行了一些运行时的处理，比如：生命周期的对齐。其他的部分，如通过 BOM/DOM 方法构建、修改 DOM Tree 及渲染原理，都是和 React 一致的。

更新

无论是 React 还是 Vue，最终都会调用 Taro DOM 方法，如：appendChild、insertChild 等。

这些方法在修改 Taro DOM Tree 的同时，还会调用 enqueueUpdate 方法，这个方法能获取到每一个 DOM 方法最终修改的节点路径和值，如：{root.cn.[0].cn.[4].value: "1"}，并通过 setData 方法更新到视图层。



Taro NPM

NPM 包	描述
babel-preset-taro [3]	给 Taro 项目使用的 babel preset
@tarojs/taro [4]	暴露给应用开发者的 Taro 核心 API
@tarojs/shared [5]	Taro 内部使用的 utils
@tarojs/api [6]	暴露给 @tarojs/taro 的所有端的公有 API
@tarojs/taro-h5 [7]	暴露给 @tarojs/taro 的 H5 端 API
@tarojs/router [8]	Taro H5 路由
@tarojs/react [9]	基于 react-reconciler 的小程序专用 React 渲染器
@tarojs/cli [10]	Taro 开发工具
@tarojs/extend [11]	Taro 扩展, 包含 jQuery API 等
@tarojs/helper [12]	内部给 CLI 和 runner 使用辅助方法集
@tarojs/service [13]	Taro 插件化内核
@tarojs/taro-loader [14]	露给 @tarojs/mini-runner 和 @tarojs/webpack-runner 使用的 Webpack loader
@tarojs/runner-utils [15]	暴露给 @tarojs/mini-runner 和 @tarojs/webpack-runner 的公用工具函数
@tarojs/webpack-runner [16]	Taro H5 端 Webpack 打包编译工具
@tarojs/mini-runner [17]	Taro 小程序 端 Webpack 打包编译工具
@tarojs/components [18]	Taro 标准组件库, H5 版
@tarojs/taroize [19]	Taro 小程序反向编译器
@tarojs/with-weapp [20]	反向转换的运行时适配器
eslint-config-taro [21]	Taro ESLint 规则
eslint-plugin-taro [22]	Taro ESLint 插件

