

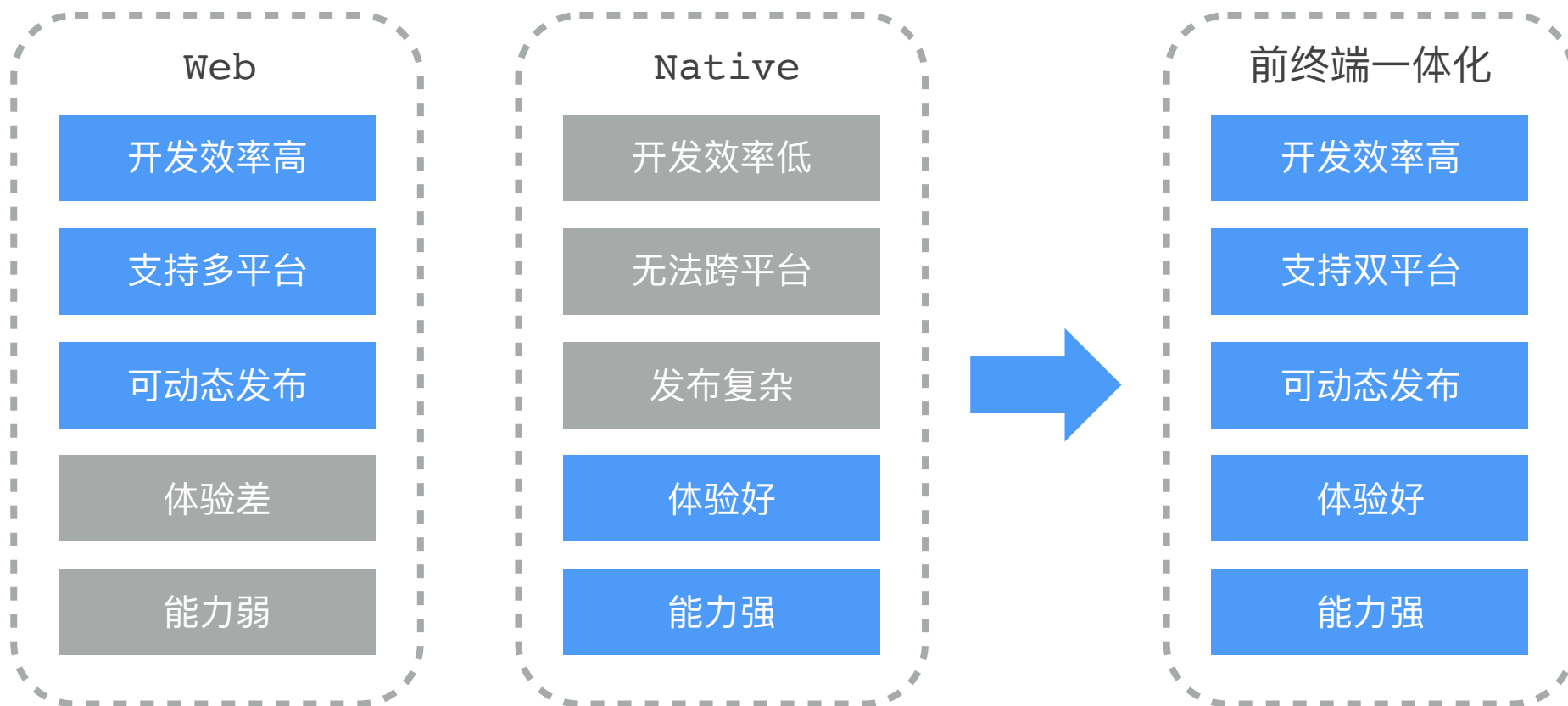


前终端统一方案

Hippy-Vue 设计和实现



传统移动端开发模式，无法满足业务精细化运营需求，急需寻找一种高性能、跨平台、动态发布的开发解决方案。








Hippy 简介

腾讯 Hippy 动态运营解决方案由 **Hippy SDK** 与 **Hippy 管理平台** 组成：

- **Hippy SDK** 是一套多端统一的开发框架，它能够通过 JavaScript **高效率**地开发出：**跨平台**（安卓/iOS/Web）、**高性能**的原生应用，尤其适合需要快速迭代又追求高性能的产品。
- **Hippy 管理平台** 涵盖编译、发布、监控等动态运营所需的各个环节，并支持灰度比例、ABTest、增量包等高级特性。



 t.js
 Library


 User Code
 JavaScript


 Vue.js
 Library

前端主导


支持最流行的框架





HIPPY SDK

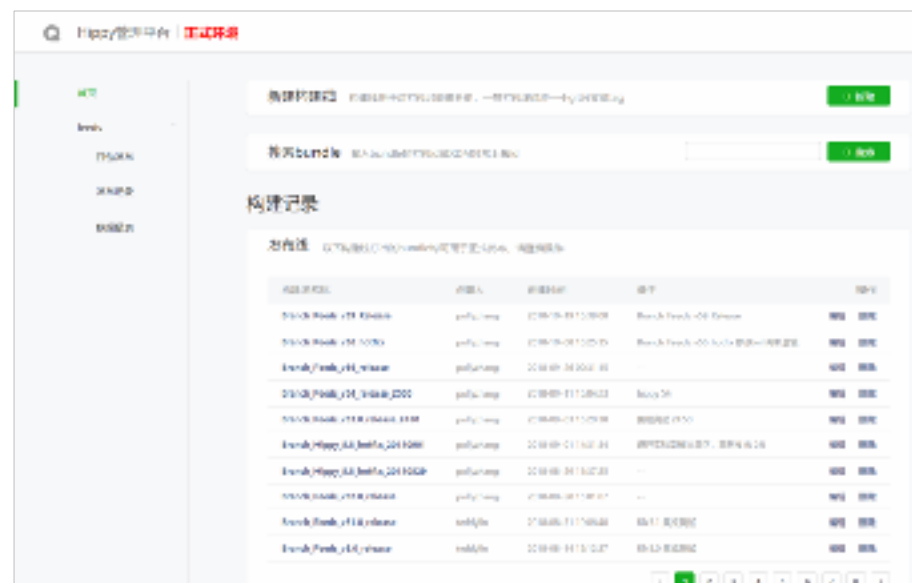
向阳而生

代码即将对外开源


 Android
 App


 iOS
 App


 Flutter
 App





高性能可复用滚动列表

动画和手势实现

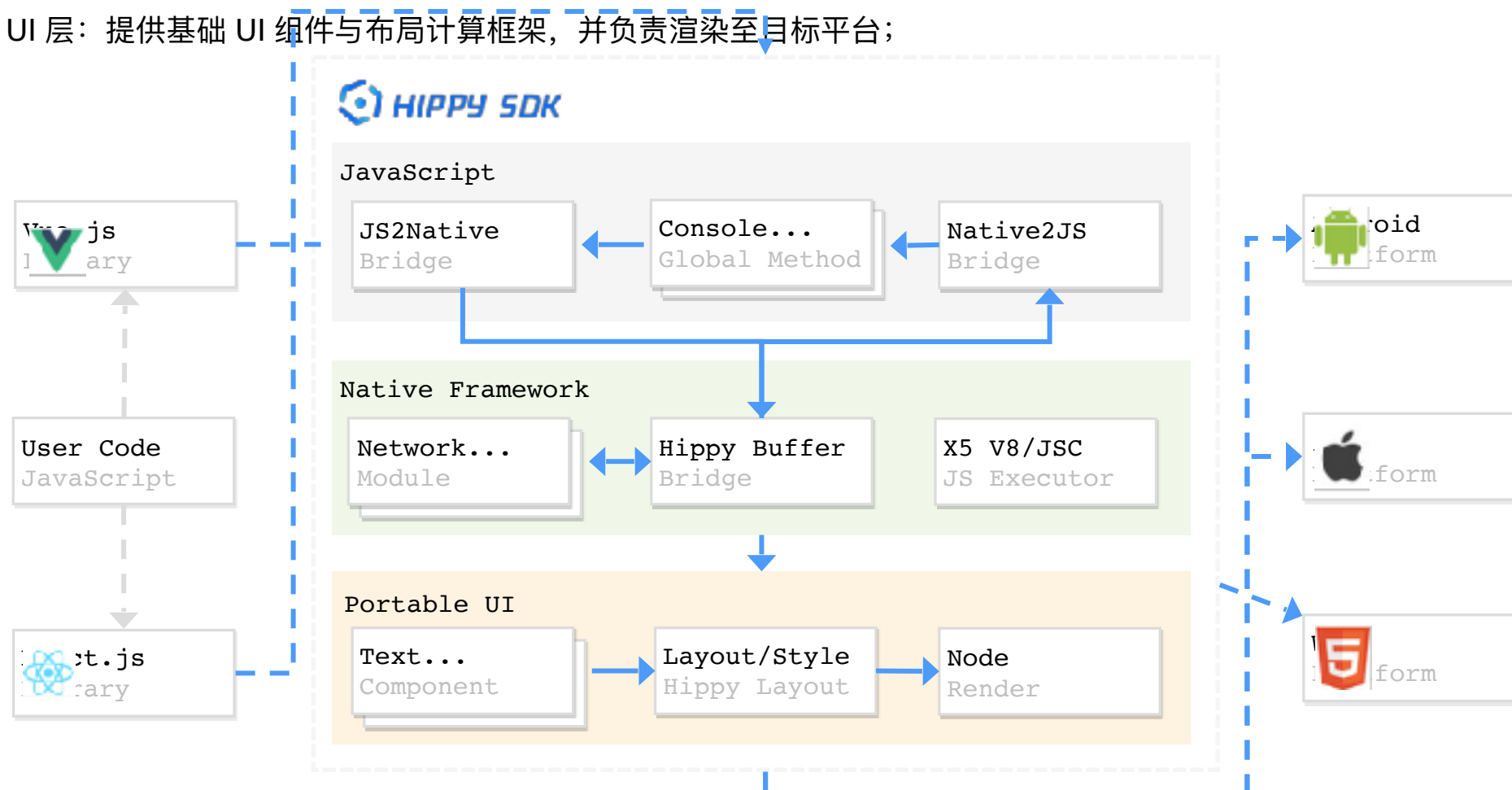
x5 内核团队提供支持

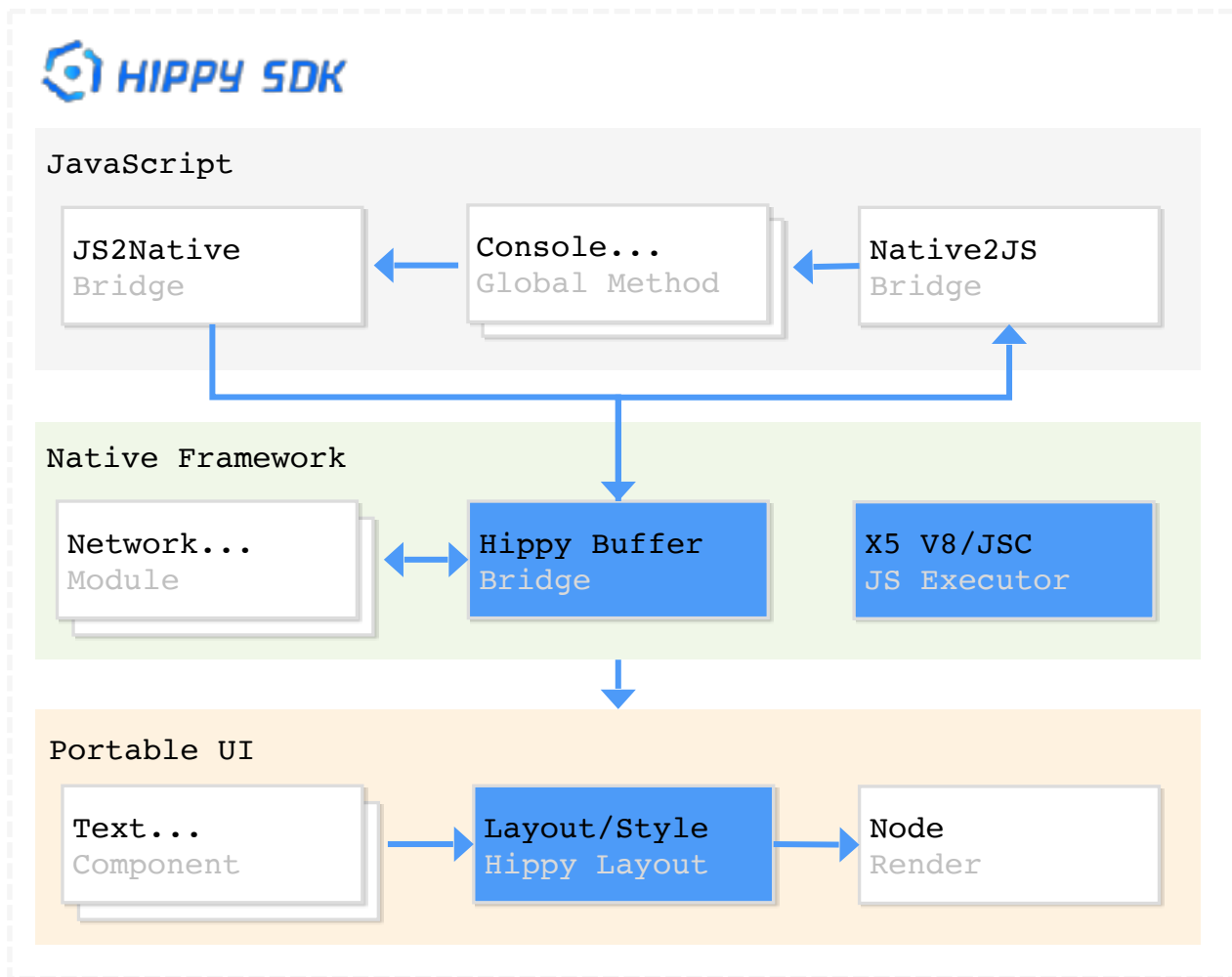


项目架构

Hippy SDK 采用三层设计，其中：

- JavaScript 层：提供业务代码运行时的前端上下文环境；
- Native Framework 层：负责前终端通讯与 JavaScript VM，并提供 Native 相关模块；
- Portable UI 层：提供基础 UI 组件与布局计算框架，并负责渲染至目标平台；





Hippy Buffer

二进制传输协议，编解码性能更好。

x5 v8

x5 团队特供 v8 引擎。

Hippy Layout

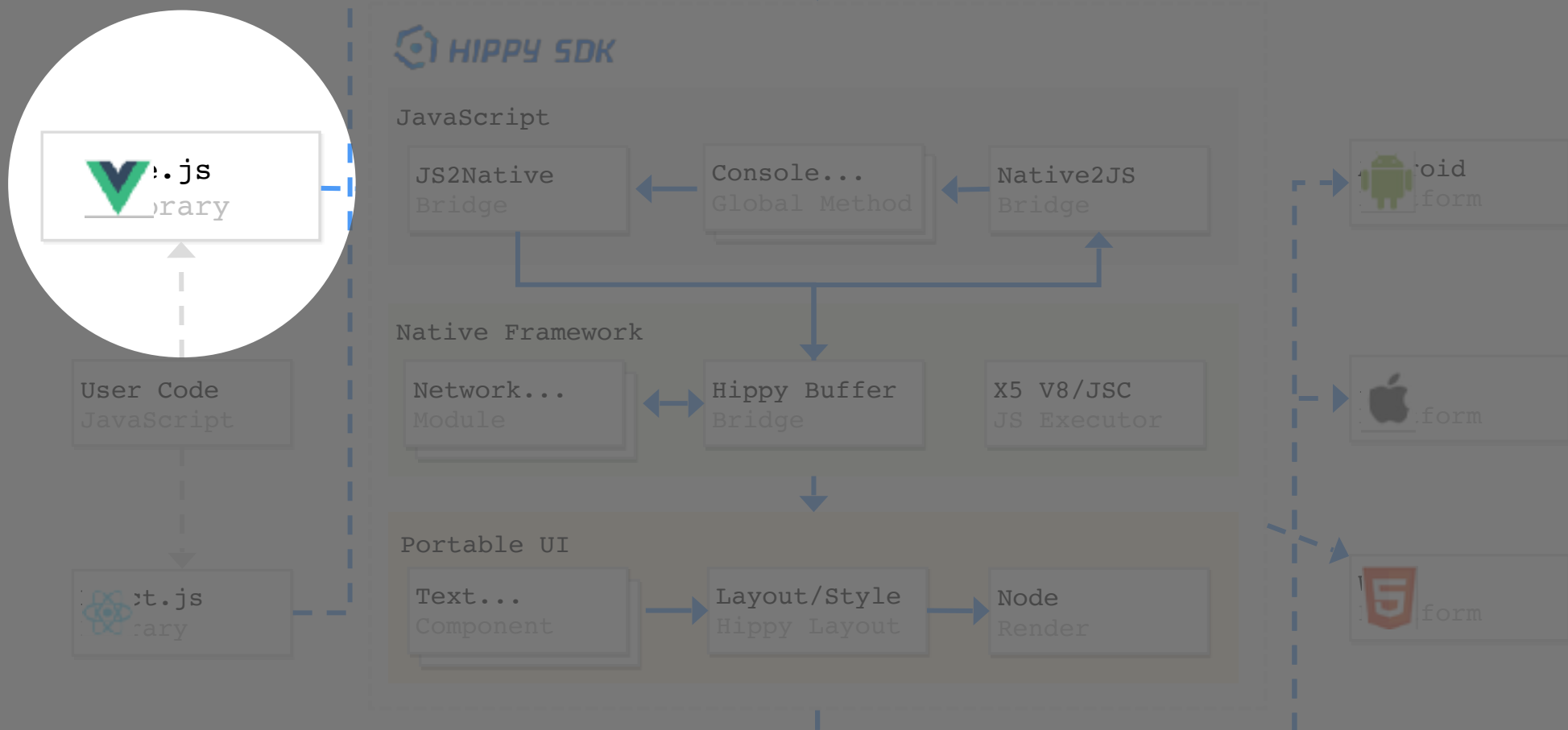
iOS、Android 共享布局引擎，纯 C 开发，只有 50kb。



项目架构

Hippy SDK 采用三层设计，其中：

- JavaScript 层：提供业务代码运行时的前端上下文环境；
- Native Framework 层：负责前终端通讯与 Javascript VM，并提供 Native 相关模块；
- Portable UI 层：提供基础 UI 组件与布局计算框架，并负责渲染至目标平台；





容易

复用

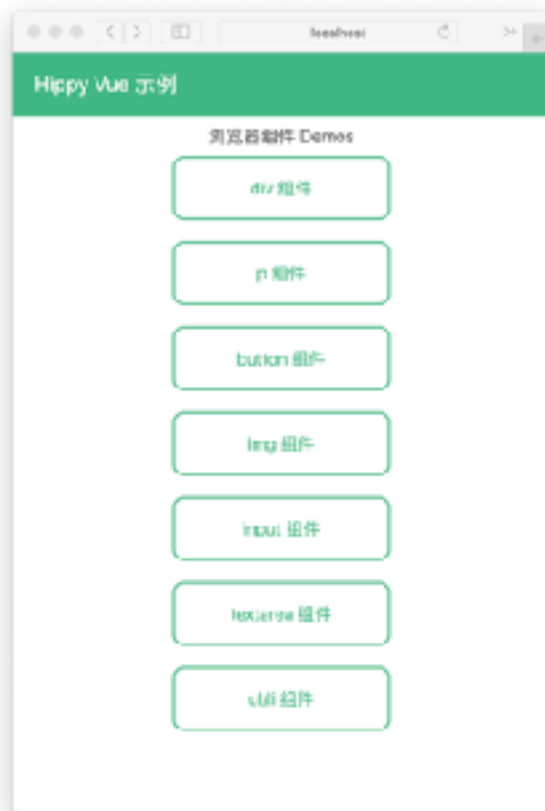
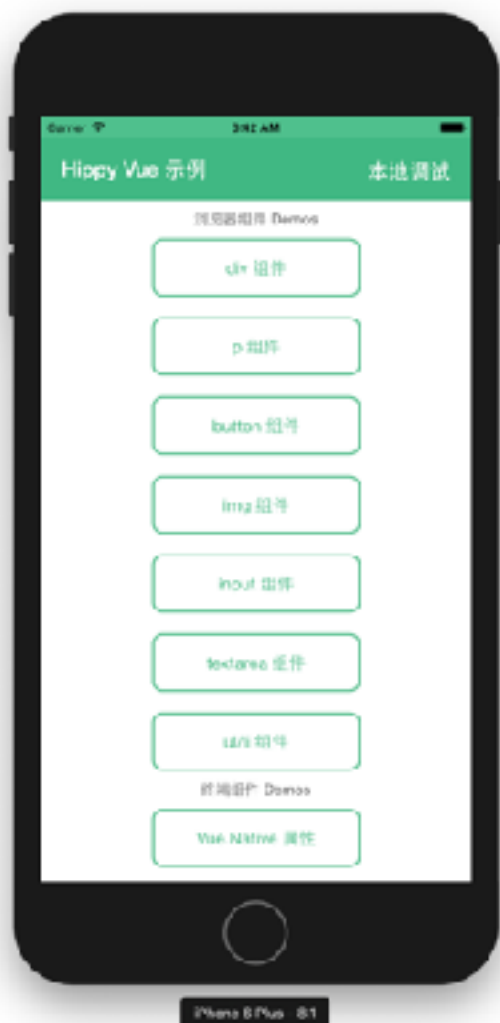
纯粹



容易 前端开发所熟悉的

复用 拥抱标准

纯粹 只做移动端的实现



同一套代码，三端运行：

移动端用 Hippy-Vue 渲染原生代码
浏览器里用原版 Vue 渲染网页代码

终端扩展 Vue.Native，可以通过它
判断是否在终端运行，访问原生能力。



同样的 div

浏览器里支持的背景图样式
Hippy-Vue 同样支持

跟浏览器一样通过 overflow 样式
确定滚动方向



CSS display 参数支持 none 值
v-show 照样使用

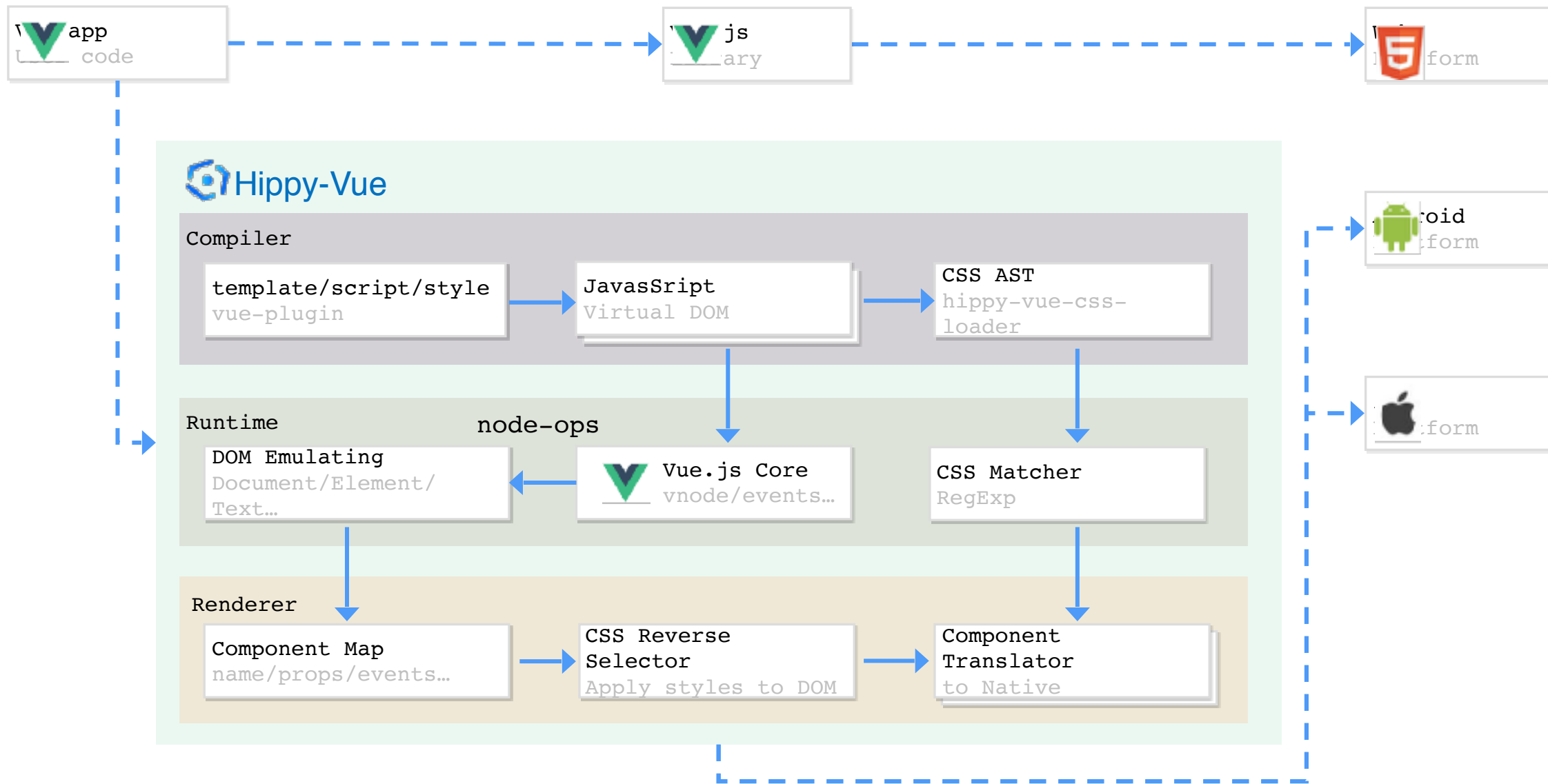


textarea 跟浏览器一模一样

v-model 直接用来做数据绑定

Emoji 支持没问题 😊

Hippy-Vue 架构





HTML

```
<a @click="clickHandler">Hello</a>
```



Hippy Buffer

```
{  
  name: "Text",      // 组件名称转换  
  props: {  
    value: "Hello", // 参数转换  
    onClick: true,  // 事件绑定  
  },  
}
```

```

```



```
{  
  name: "Image",  
  props: {  
    source: {  
      uri: "http://www.qq.com/logo.png",  
    },  
  },  
}
```



CSS AST
hippy-vue-css-loader

```
#id { color: red }  
#id .class { background-color: blue }
```

```
[  
  {  
    selector: [['#id']],  
    style: [{ color: 12345 }],  
  },  
  {  
    selector: [['#id', '.class']],  
    style: [{ backgroundColor: 56789 }]}  
]
```

CSS Matcher
RegExp

```
[  
  Matcher({  
    selector: [[IdSelector('id')]],  
    style: [ { color: 12345 } ],  
  }),  
  Matcher({  
    selector: [[IdSelector('id'), ClassSelector('class')]],  
    style: [ { backgroundColor: 56789 } ]  
  })  
]
```

```
<p>Hello</p>  
<p id="id">  
  VueConf  
  <p class="class">Hangzhou</p>  
</p>
```

CSS Reverse
Selector
Apply styles to DOM

- 0: [false, false] = Mismatch
- 1: [true, false] = Part Matched
- 2: [true, true] = Full Matched;

Hello
VueConf
Hangzhou



 vue 升级到 3.0 后?



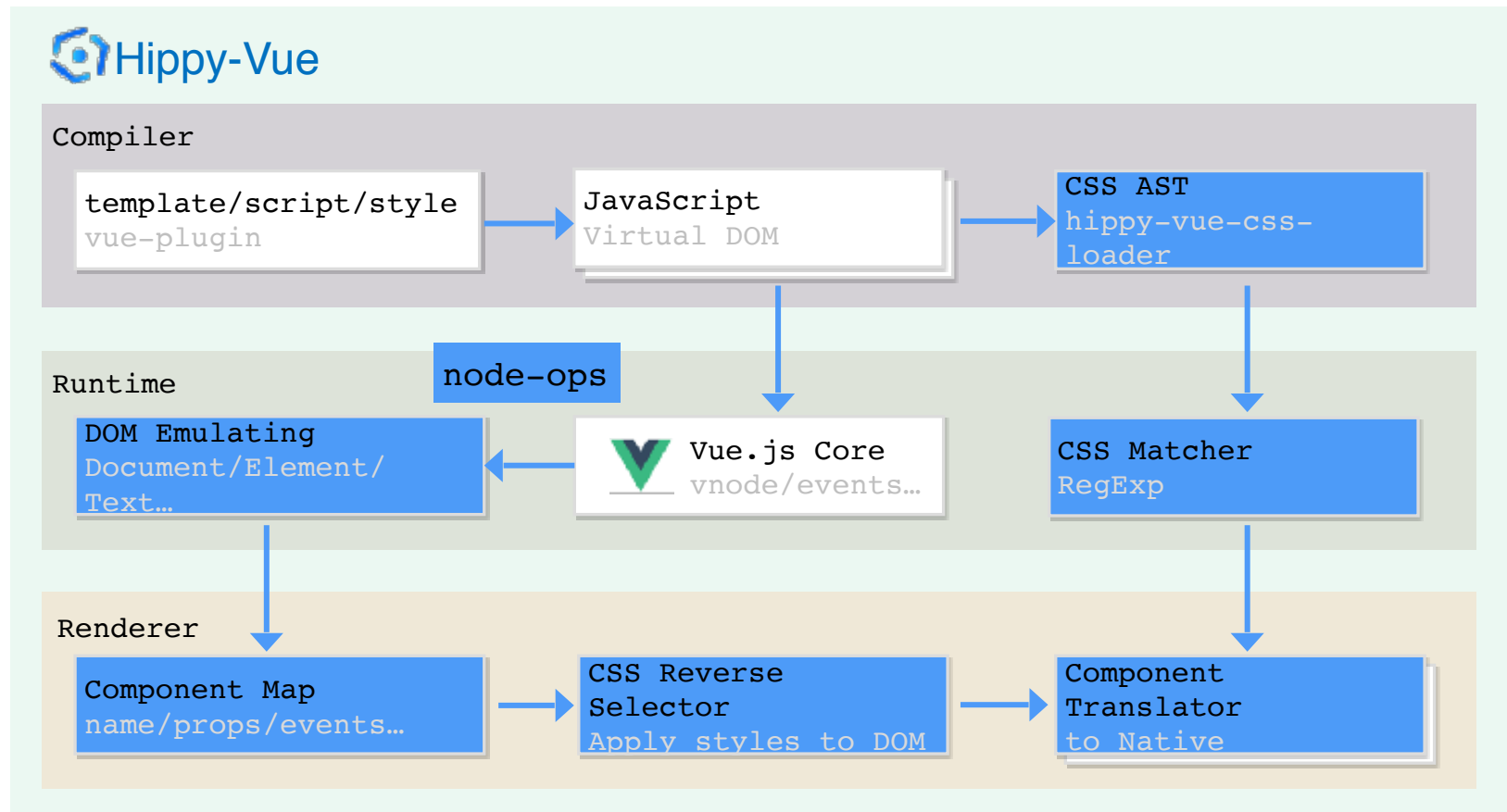
核心升级不影响 Hippy-Vue

Hippy-Vue 只更改 Vue 的输入和输出，核心逻辑并未侵入：

- 通过 node-ops 接口对接自定义 DOM
- 转换 DOM 到终端。
- 单独实现 CSS 编译和解析

Vue 3.0 升级后如果 node-ops 接口不变，则可以直接升级适配。

更新：有了 createRender() 更方便了。



Hippy 将在近期开源

Thx a lot

Keep in touch

