



课程二维码

基于 Vue.js 的小程序开发框架 mpvue

美团酒旅 胡成全



胡成全 @huchengquan

专注移动端和小程序方向
致力小程序工程化开发、业务级应用
酒旅终端研发组 - 基础服务平台

初衷 & 愿景

1. 为开发社区提供一个技术方案
2. 为跨端复用提供一个实现思路
3. 提升小程序开发的质量、效率、体验
4. 我们不创造规范，我们是规范的搬运工

小程序开发现状

mpvue 框架介绍

mpvue 实现思路

mpvue 未来计划

小程序技术栈特点

简单的MV框架

- ❖ 推荐简单的逻辑实现
- ❖ 缺乏构建复杂应用能力

1

H5 + Native

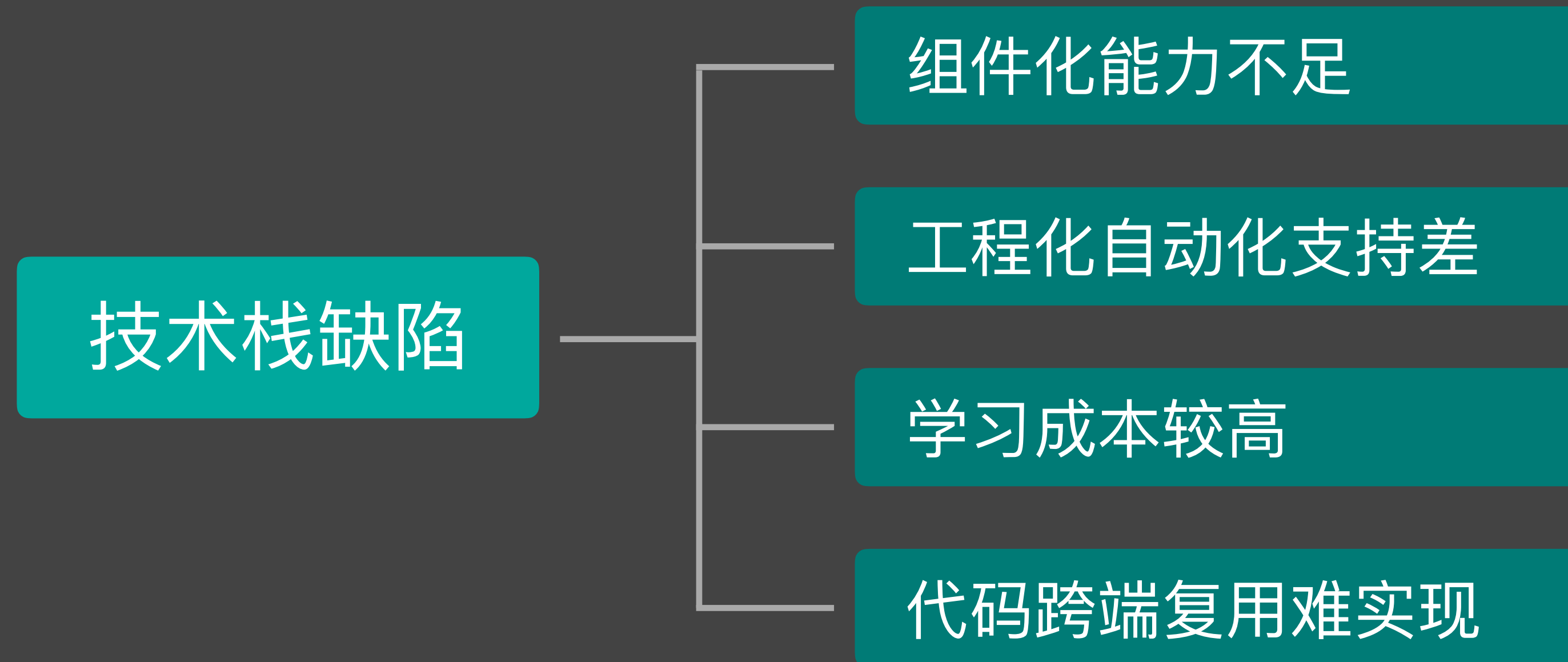
- ❖ 精简的H5能力和封装
- ❖ 扩充的微信平台能力

2

全新技术栈

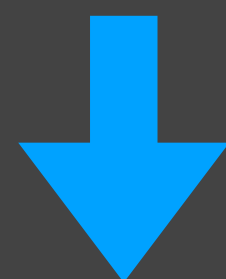
- ❖ 全新的开发范式
- ❖ 全新的技术体系

3

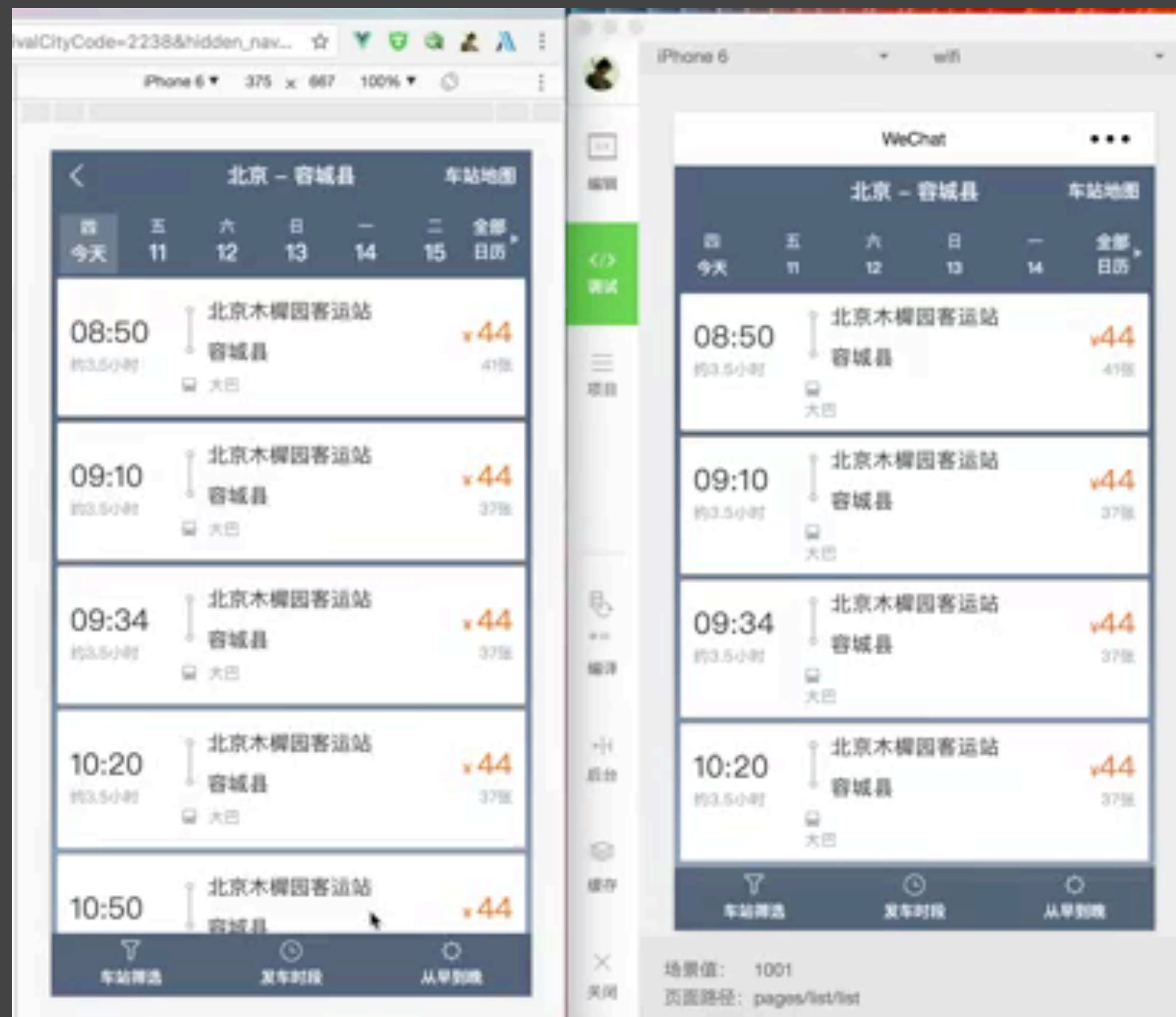


产品有一个大胆的想法

现有H5应用



同等小程序



小程序开发的核心诉求

H5 转化为小程序
代码复用最大化

快速复用已有产品

技术栈体系快速打通
最好能统一到已有技术栈

技术栈无缝迁移

小程序开发踩坑自动规避
自动化工具集成

降低学习成本

mpvue 框架介绍



1. 基于 `Vue.js` 的小程序开发框架
2. 提供完整的 `Vue.js` 开发体验
3. 支持 H5 和小程序代码复用

mpvue 能带来什么

能力扩展

前端赋能

快速上手

节约新人上手
成本

代码复用

技术沉淀收益
规模化

自动化构建

提高开发效率
避免体力劳动

组件化开发

企业级开发的核心能力
对大型项目尤为重要

mpvue 优势

Vue.js支持

Vue.js 编码体验

从底层支持 Vue 语法特性

Vue 组件化支持

内置 Vue 引擎

工程化集成

小程序代码自动编译

npm 依赖支持

开发代码热更新

cli 项目初始化支持

使用场景

- ◆ 新开发小程序但不熟悉技术栈
- ◆ 有 Vue.js 基础，想快速上手
- ◆ 需要组件化开发能力
- ◆ 需要和 H5 复用代码
- ◆ 玩儿票

mpvue 合作案例

美团汽车票

美团充电

美团门票

美团酒店

美团火车票

金融点付

使用方式

```
# install vue-cli
$ npm install --global vue-cli

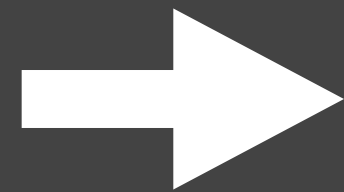
# create new project with template
$ vue init 'bitbucket:git.sankuai.com:hfe/mpvue-quickstart' --clone my-mpvue

# install dependencies and go
$ cd my-mpvue
$ npm install
$ npm run dev
```

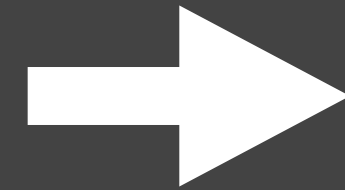
shell



- my-mpvue
 - build
 - config
 - dist
 - node_modules
 - src
 - static
 - .babelrc
 - .editorconfig
 - .eslintignore
 - .eslintrc.js
 - .gitignore
 - .npmrc
 - .postcssrc.js
 - index.html
 - package-lock.json
 - package.json**
 - README.md

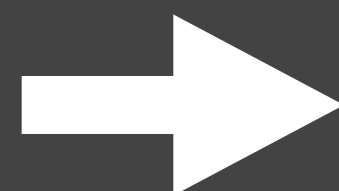


```
1 {
2   "name": "my-mpvue",
3   "version": "1.0.0",
4   "description": "A Vue.js project",
5   "author": "huchengquan <huchengquan@meituan.com>",
6   "private": true,
7   "scripts": {
8     "dev": "node build/dev-server.js",
9     "start": "node build/dev-server.js",
10    "build": "node build/build.js",
11    "lint": "eslint --ext .js,.vue src"
12  },
13  "dependencies": {
14    "@hfe/mpvue": "0.0.11",
15    "vuex": "^2.5.0"
16  },
17  "devDependencies": {
65  },
66  "engines": {
67    "node": ">= 4.0.0",
68    "npm": ">= 3.0.0"
69  },
70  "browserslist": [
71    "> 1%",
72    "last 2 versions",
73    "not ie <= 8"
74  ]
75 }
```

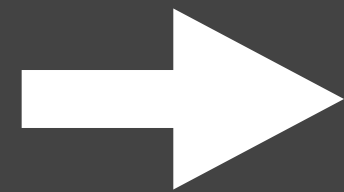
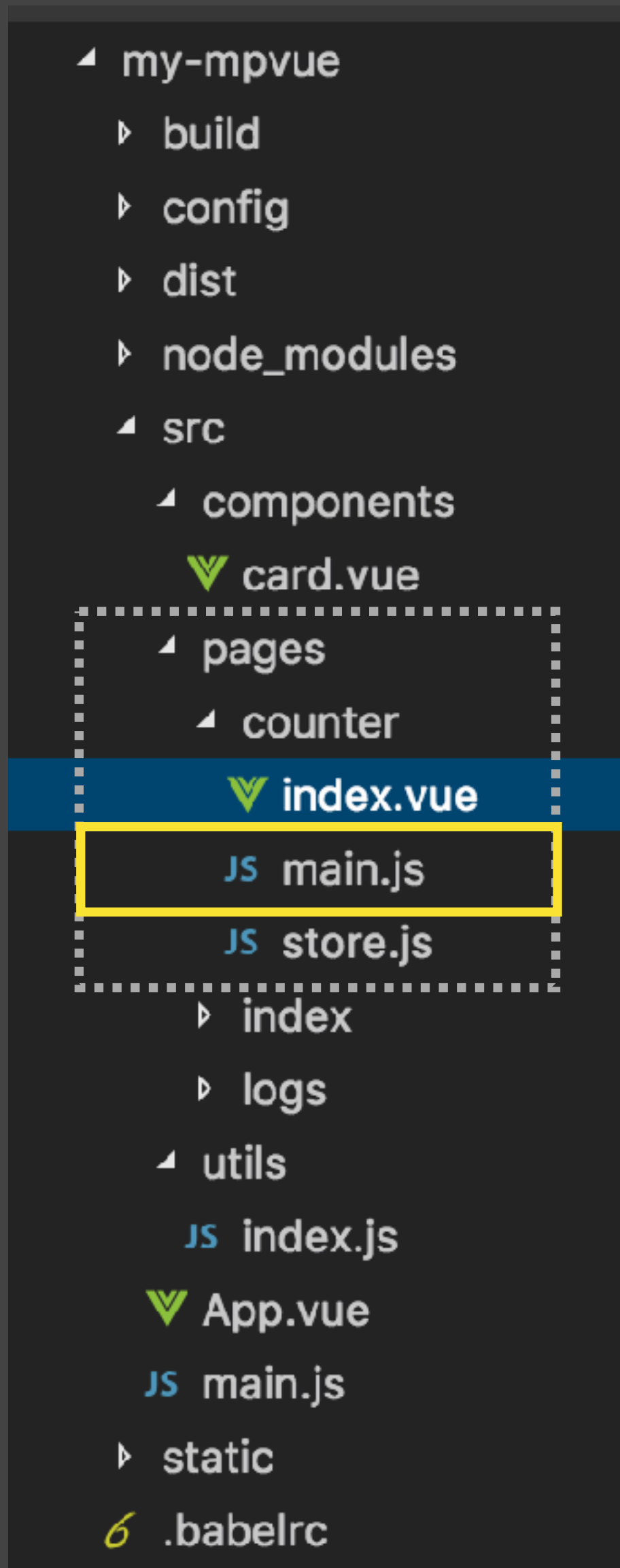


```
JS main.js x
1  import Vue from 'vue'
2  import App from './App'
3
4  Vue.config.productionTip = false
5  App.mpType = 'app'
6
7  const app = new Vue(App)
8  app.$mount()
9
10 export default {
11   // app.json 配置
12   config: {
13     // 如下字段自动填充
14     pages: [],
15     window: {
16       backgroundColor: 'light',
17       navigationBarBackgroundColor: '#fff',
18       navigationBarTitleText: 'WeChat',
19       navigationBarTextStyle: 'black'
20     }
21   }
22 }
23
```

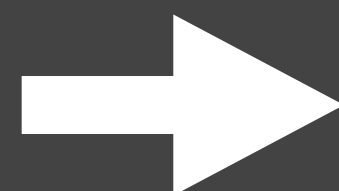
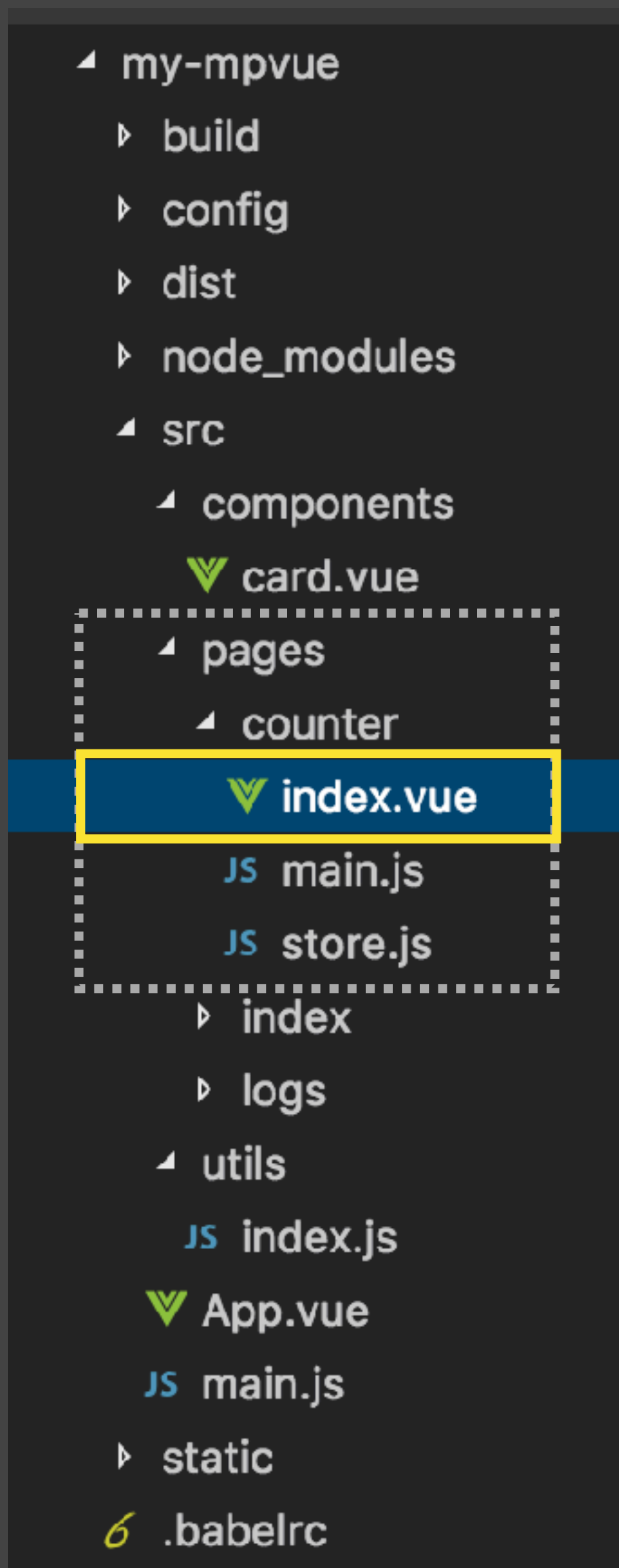
小程序特有配置



```
App.vue x
1  <script>
2  export default {
3    created() {
4      // 调用API从本地缓存中获取数据
5      const logs = wx.getStorageSync('logs') || []
6      logs.unshift(Date.now())
7      wx.setStorageSync('logs', logs)
8
9      console.log('app created and cache logs by setStorageSync')
10 }
11 }
12 </script>
13
14 <style>
15 .container {
16   height: 100%;
17   display: flex;
18   flex-direction: column;
19   align-items: center;
20   justify-content: space-between;
21   padding: 200rpx 0;
22   box-sizing: border-box;
23 }
24 /* this rule will be remove */
25 * {
26   transition: width 2s;
27   -moz-transition: width 2s;
28   -webkit-transition: width 2s;
29   -o-transition: width 2s;
30 }
31 </style>
```

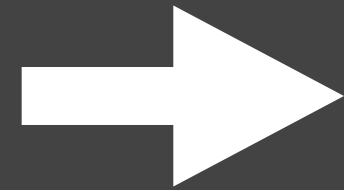



```
JS main.js x
1 import Vue from 'vue'
2 import App from './index'
3
4 const app = new Vue(App)
5 app.$mount()
6
```



```
index.vue x
1  <template>
2    <div class="counter-warp">
3      <p>Vuex counter: {{ count }}</p>
4      <p><button @click="increment">+</button>
5      <button @click="decrement">-</button>
6    </p>
7    <a href="/pages/index/index" class="home">去往首页</a>
8  </div>
9  </template>
10
11 <script>
12   import store from './store'
13   export default {
14     computed: {
15       count () {
16         return store.state.count
17       }
18     },
19     methods: {
20       increment () {
21         store.commit('increment')
22       },
23       decrement () {
24         store.commit('decrement')
25       }
26     }
27   }
28 </script>
```

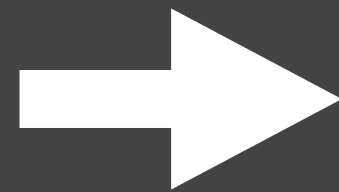
- my-mpvue
 - build
 - config
 - dist
 - node_modules
 - src
 - components
 - card.vue
 - pages
 - counter
 - index.vue**
 - main.js
 - store.js**
 - index
 - logs
 - utils
 - index.js
 - App.vue
 - main.js
 - static
 - .babelrc



```
JS store.js  x
1  // https://vuex.vuejs.org/zh-cn/intro.html
2  // make sure to call Vue.use(Vuex) if using a module system
3  import Vue from 'vue'
4  import Vuex from 'vuex'
5
6  Vue.use(Vuex)
7
8  const store = new Vuex.Store({
9    state: {
10      count: 0
11    },
12    mutations: {
13      increment: (state) => {
14        const obj = state
15        obj.count += 1
16      },
17      decrement: (state) => {
18        const obj = state
19        obj.count -= 1
20      }
21    }
22  })
23
24  export default store
```



```
└─ src
  └─ components
    └─ card.vue
  └─ pages
    └─ counter
      └─ index.vue
      └─ main.js
      └─ store.js
    └─ index
      └─ index.vue
      └─ main.js
    └─ logs
      └─ index.vue
      └─ main.js
  └─ utils
    └─ index.js
  └─ App.vue
  └─ main.js
```



```
└─ dist
  └─ components
    └─ card$4b50dad0.wxml
    └─ counter$71c00f23.wxml
    └─ index$616ea179.wxml
    └─ logs$388d0704.wxml
  └─ pages
    └─ counter
    └─ index
      └─ index.js
      └─ index.wxml
      └─ index.wxss
    └─ logs
  └─ static
    └─ css
      └─ app.wxss
      └─ counter.wxss
      └─ index.wxss
      └─ logs.wxss
    └─ js
      └─ app.js
      └─ app.json
      └─ app.wxss
```

```
└─ js
  └─ app.js
  └─ app.js.map
  └─ counter.js
  └─ counter.js.map
  └─ index.js
  └─ index.js.map
  └─ logs.js
  └─ logs.js.map
  └─ manifest.js
  └─ manifest.js.map
  └─ vendor.js
  └─ vendor.js.map
```

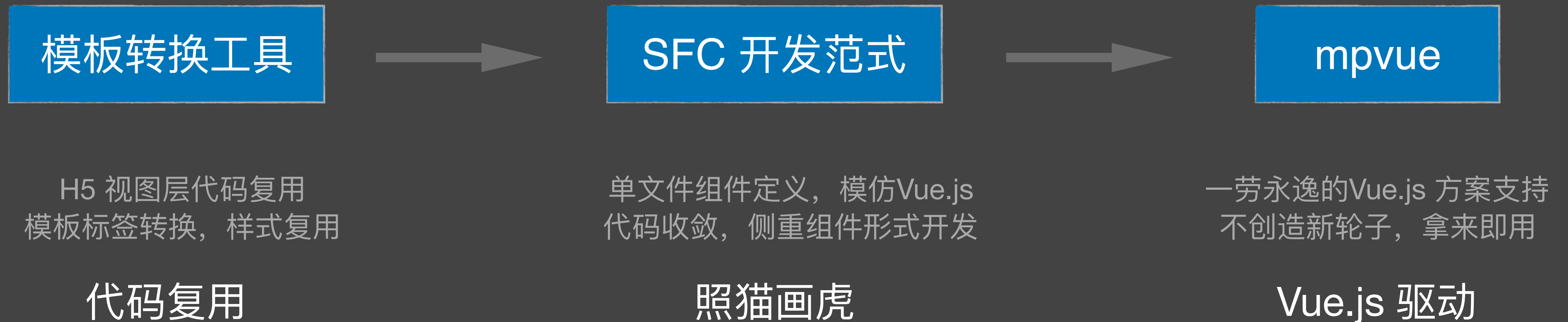
mpvue 实现思路

小程序框架机制



响应式数据绑定，数据驱动视图更新
两个 JS 引擎间通信，字符串数据传输

mpvue Roadmap



Vue.js 引擎引入 - 运行时支持

mpvue runtime

维护 Vue 运行时实例

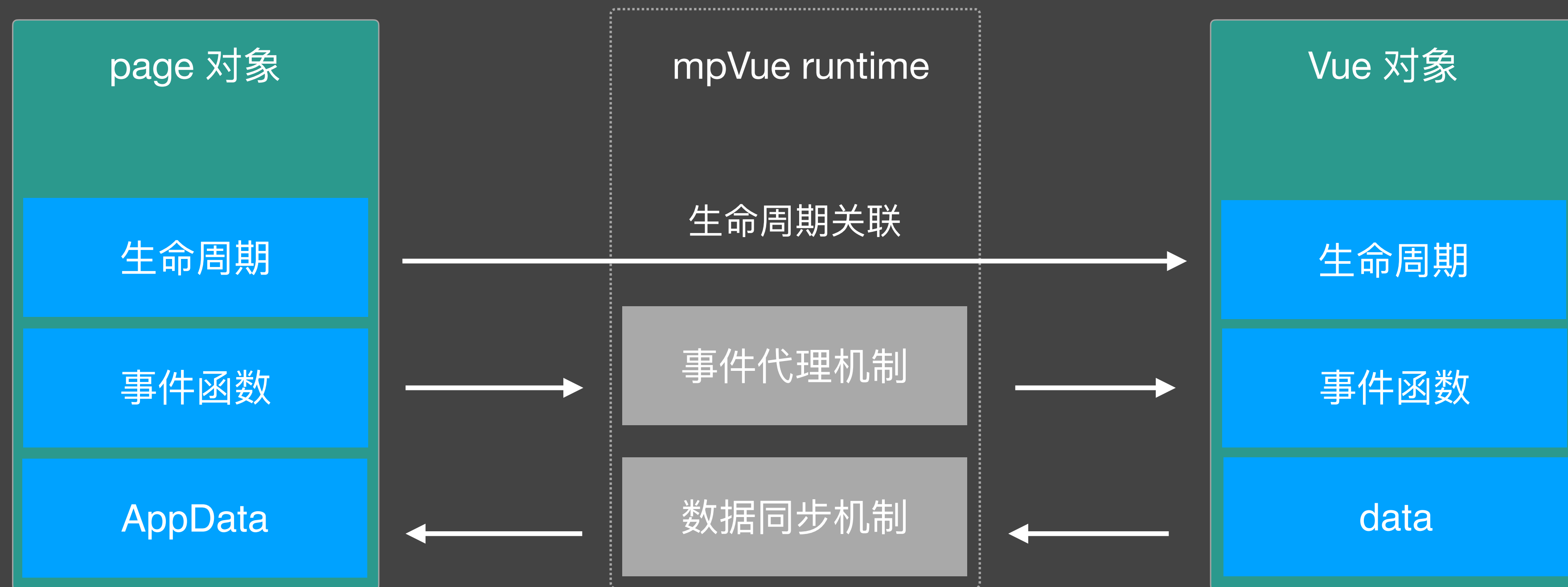
维护小程序运行实例

实例间生命周期驱动

小程序事件触发 Vue 实例方法

Vue 维护数据模型和虚拟 dom
小程序负责视图层渲染

mpvue 实现原理



模板编译器实现 - 构建时支持

mpvue-loader

从 SFC 提取出 AST

AST 数据改造

从 AST 构建模板代码

```
card.vue x
1 <template>
2   <div>
3     <p class="card">
4       {{text}}
5     </p>
6   </div>
7 </template>
8
9 <script>
10 export default {
11   props: ['text']
12 }
13 </script>
14
15 <style>
16 .card {
17   padding: 10px;
18 }
19 </style>
```



```
card$4b50dad0.wxml x
1 <template name="card$4b50dad0">
2   <view class="_div" data-v-cef12b42">
3     <view class="_p" data-v-cef12b42 card">
4       {{text}}
5     </view>
6   </view>
7 </template>
```

mpvue 其它基建

自定义构建

小程序 webpack target

mpvue-loader 自动引入

样式转换

rpx 自动转换工具

postcss 选择器清理

示例项目模板

Vue.js 标准模板

小程序功能定制

同类框架对比

	微信小程序	mpvue	wepy
语法规则	小程序开发规范	Vue.js 开发规范	类 Vue.js 开发规范
标签集合	小程序标签	html 标签 + 小程序标签	小程序标签
样式规范	wxss	sass, less, postcss	sass, less, styus
组件化	弱组件化机制	Vue.js SFC	类 Vue.js SFC
跨端复用	不可复用	支持转换为 H5	支持转换为 H5
自动构建	本身无自动构建	webpack 构建	内置构建
上手成本	全新学习	熟悉 Vue.js 即可	Vue.js 和 wepy
集中数据管理	不支持	Vuex	Redux
引入代码	无增量	模板代码 + 框架SDK	模板代码 + 框架SDK

未来计划





mpvue 大象公开群

感谢

聆听