

# Vue的组件化与单页应用

崔迪@新新农人

# 技术栈 - Vue全家桶



Vue

Vuex

Vue-router

Vue-cli (webpack)

Vux

[m.xinxinnongren.com](http://m.xinxinnongren.com)

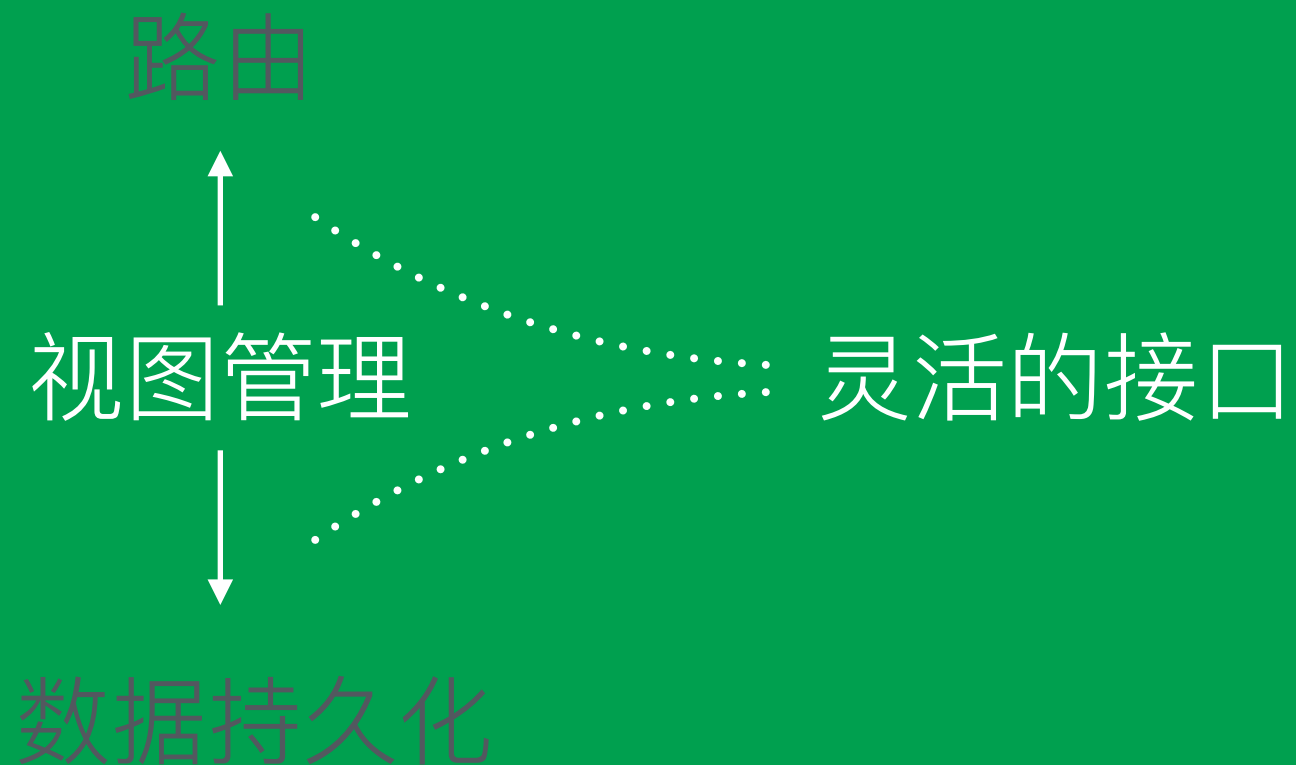
# 目录

- Vue与MVVM
- Vue与Angular简单对比
- Vue组件化
- Flux与Vuex
- Vue-router

# Vue?

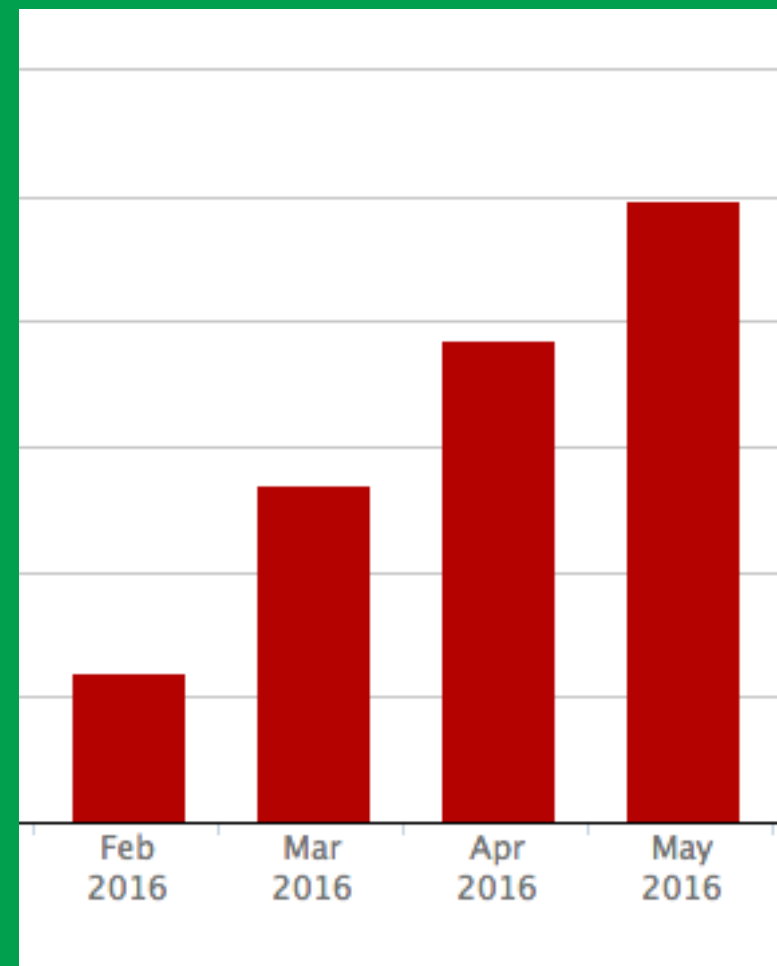
读音 /vjuː/, 类似于 view

数据驱动 + 组件化的前端界面开发的库



# Vue.js发展情况

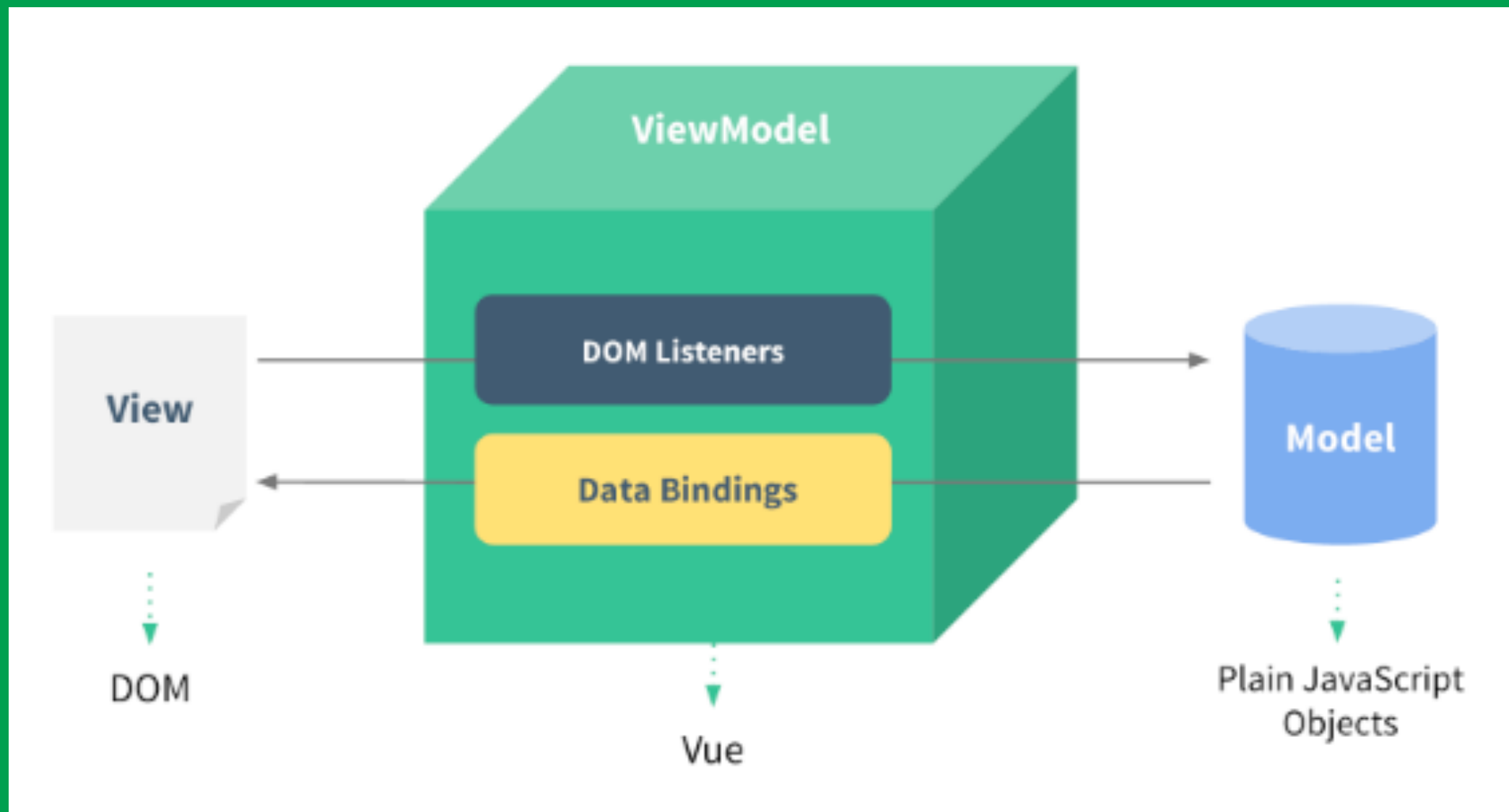
- Vue 2.0
- 阿里Weex
- NPM 总下载 60 万



npm每月下载数

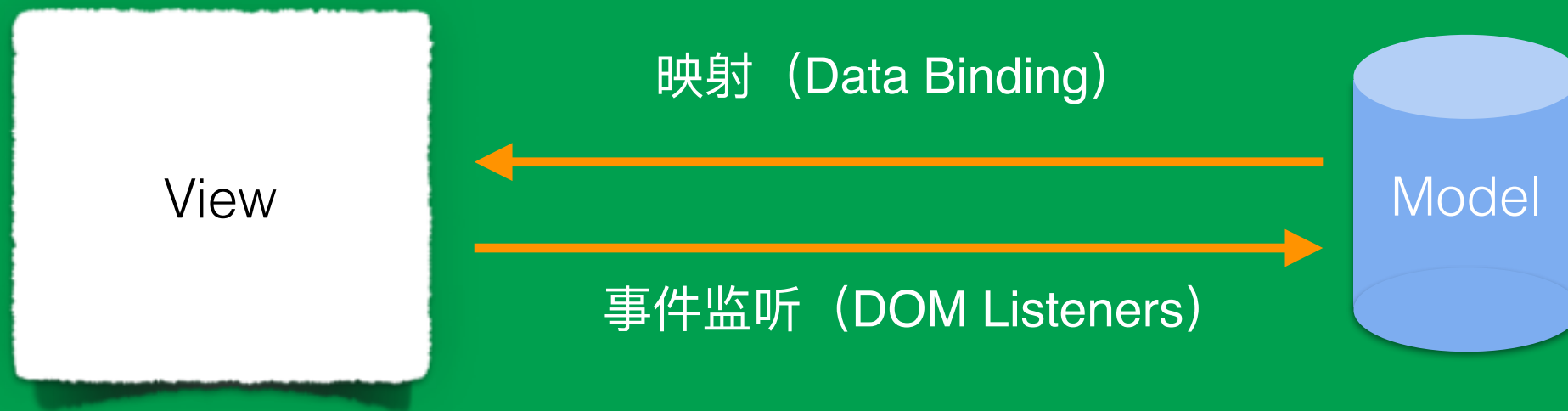
# MVVM

Model - View - ViewModel



$UI = VM(State)$

# 双向绑定



$M \rightarrow V$ : 模版引擎初始化, 监听model属性改变

$V \rightarrow M$ : v-on

# Angular VS Vue

	Angular	Vue
性能	一般（脏检查）	更好的性能(defineProperty)
兼容性	IE 8(1.2)	IE 9
学习曲线	比较陡峭	容易入门
组件化	支持一般	天然属性
与其他前端工具的契合度	自己搭配	Vue-cli脚手架
完整性	大而全	没有Router、Ajax
相关资料	丰富	比较少



# 我们选择Vue原因

强大的脚手架

完整的单页应用解决方案 (vue-router, vuex)

适合移动端：轻 (25.79kb) + 快

组件管理 (Single File Components)

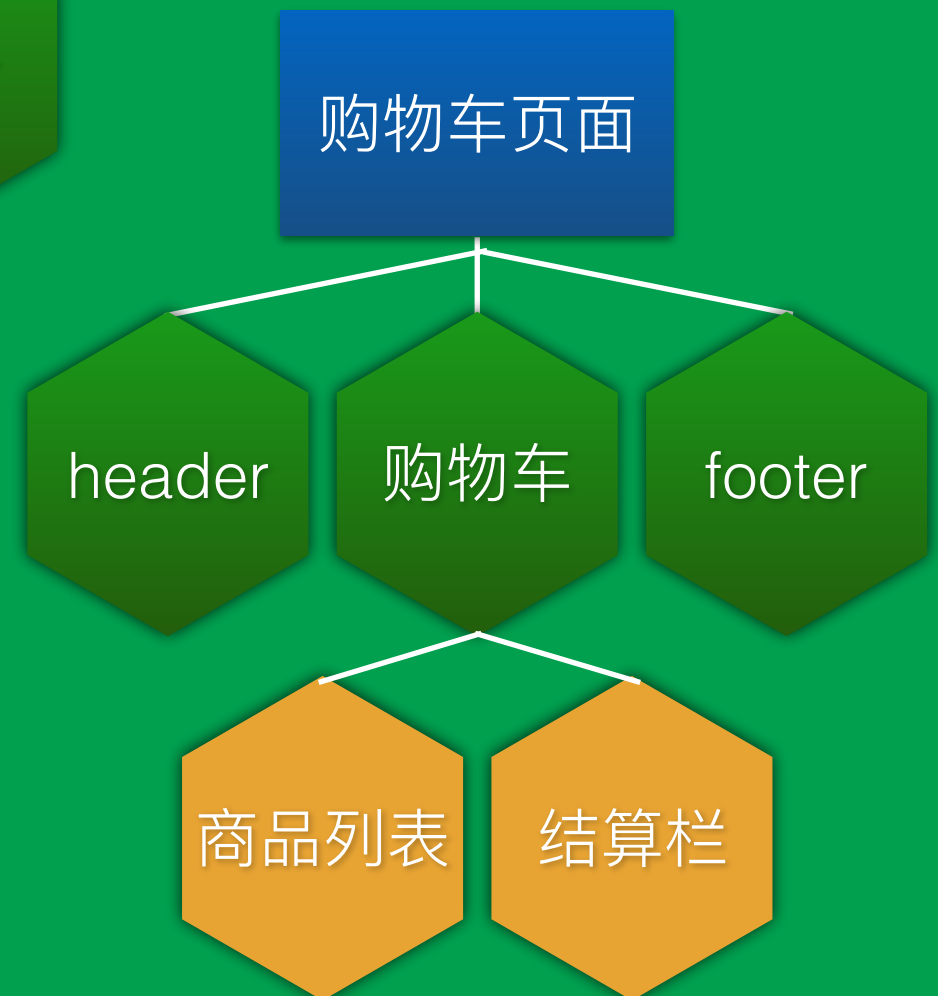
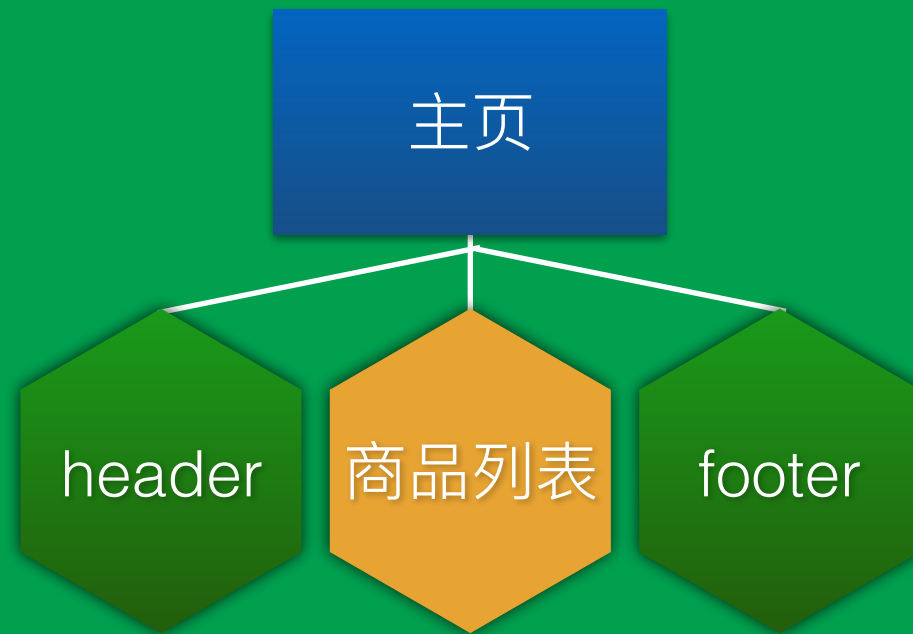
API 与设计简单，清晰简单的中文文档



# 组件化?

—“坨” 代码  
控制所有组件的状态

标签化  
组件树

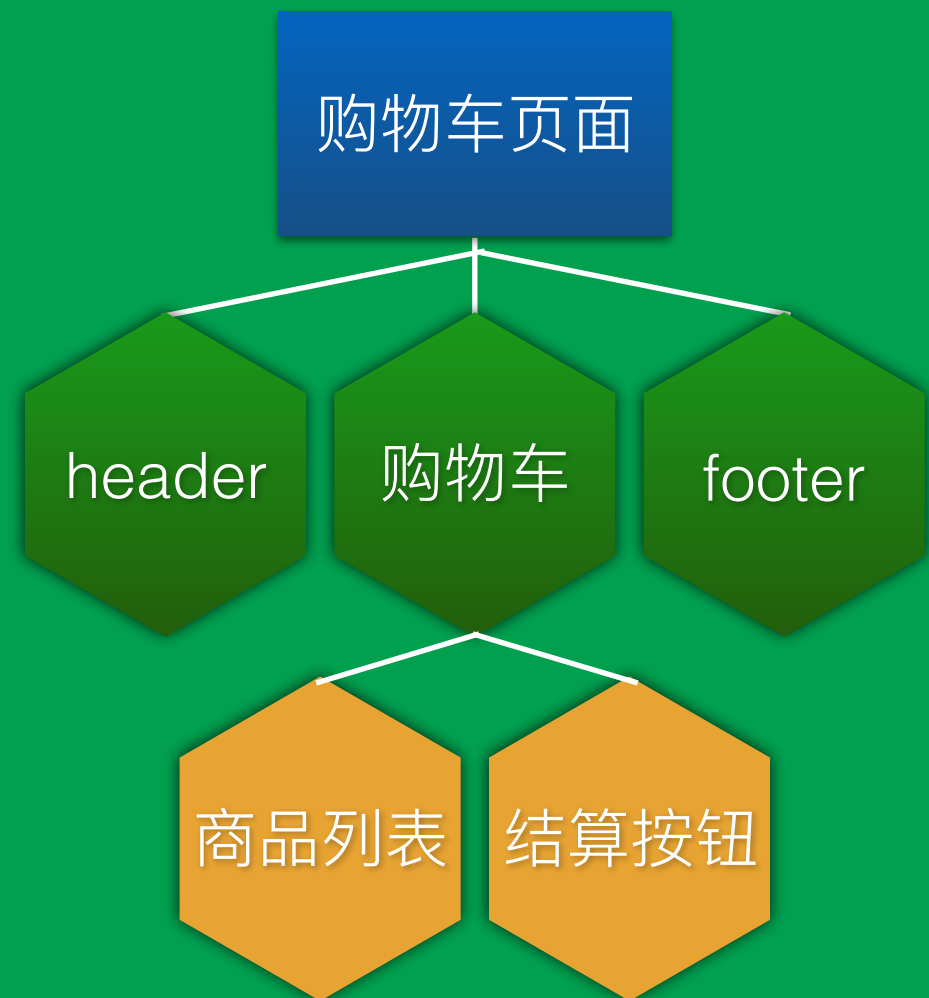
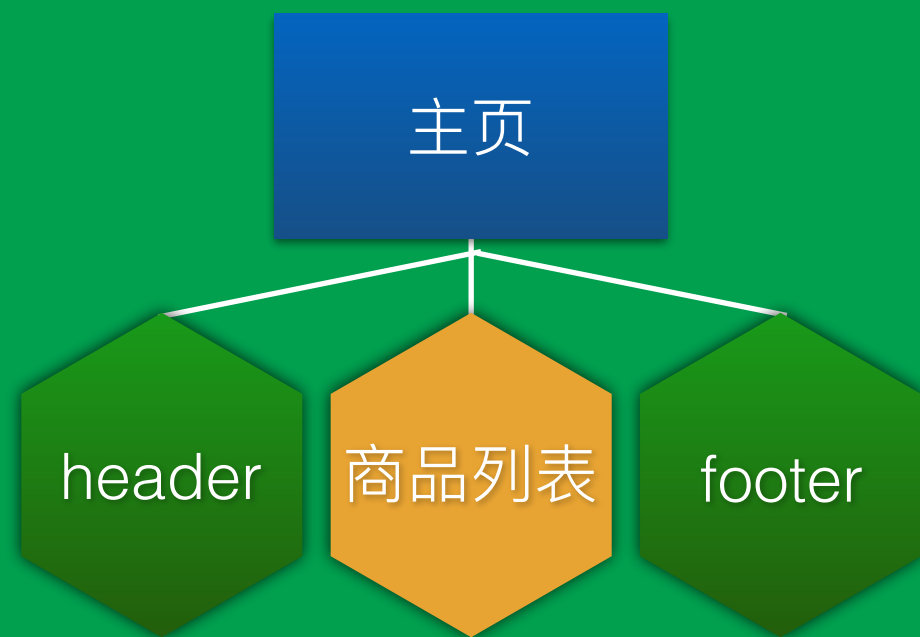


# Why 组件化?

提升开发和维护的效率

分治 > 复用

# 复用 — 分治



# 直接复用难

 <p>抢券立减 <b>5元</b></p> <p>热销推荐 <b>¥39.9</b></p> <p>原价¥69立减30元</p> <p>莫斯利安官方旗舰店</p> <p>¥39.90 包邮 10231人付款</p> <p>光明莫斯利安酸奶 常温酸奶 350g*6盒梦幻旋盖装大容量原味酸牛奶</p> <p>莫斯利安旗舰店 上海</p>	 <p>¥36.90 1786人付款</p> <p>【独立日定制装】蒙牛 精选牧场纯牛奶 250ml*12盒</p> <p>天翼超市华北站 北京</p>	 <p>Flevomel 百年乳企 乳此好荷</p> <p>¥99.00 包邮 112人付款</p> <p>风车牧场 荷兰原装进口部分脱脂新鲜早餐高钙低脂纯牛奶200ml*24</p> <p>风车牧场旗舰店 山东 烟台</p>	 <p>十二盒装</p> <p>¥79.00 10793人付款</p> <p>【天猫超市】澳洲原装进口 鲜语牧场全脂纯牛奶1L*12盒 极速新鲜</p> <p>天猫超市华北站 北京</p>
 <p>十二盒装</p> <p>【天猫超市】澳洲原装进口 鲜语牧场全脂纯牛奶1L*12盒 极速</p> <p>总销量: 59450</p> <p>¥79.00</p>	 <p>【天猫超市】德国进口牛奶 德亚全脂牛奶1Lx12 纯牛奶</p> <p>总销量: 13273</p> <p>¥89.90</p>	 <p>【天猫超市】新西兰进口牛奶Anchor安佳全脂纯牛奶250ml*24</p> <p>总销量: 13125</p> <p>¥89.00</p>	 <p>天猫超市】新西兰原装进口MeadowFresh纽麦福全脂牛奶1L*12</p> <p>总销量: 18283</p> <p>¥79.90</p>

# vue-loader

## Single File Components

vue: 一个文件来管理一个组件

内容: `<template>`

样式: `<style>`

行为: `<script>`

扩展: `lang=" ... "`

合作: 彼此间源引

```
example.vue
<template lang="jade">
  div.my-component
    h1 {{ msg }}
    other-component
</template>

<script>
  import OtherComponent from './other-component.vue'

  export default {
    components: { OtherComponent },
    data() {
      return {
        msg: 'Hello Vue.js!'
      }
    }
  }
</script>

<style lang="sass" scoped>
  $font-stack: Helvetica, sans-serif;
  $primary-color: #333;

  .my-component {
    font: 100% $font-stack;
    color: $primary-color;
  }
</style>
```

Line 1, Column 23      Spaces: 2      Vue Component

# 组件之间的传递数据

## 使用Props传递数据

- 组件实例有独立作用域
- 默认为单向绑定
- props 验证

## 组件间的事件通信

- 使用 `$dispatch()` 派发事件，事件沿着父链冒泡；
- 使用 `$broadcast()` 广播事件，事件向下传导给所有的后代。

```
Vue.component('child', {
  // 声明 props
  props: {
    msg: {
      type : String,
      required : true
    },
    // prop 可以用在模板内
    template: '<span>{{ msg }}</span>'
  })

<!-- 默认为单向绑定 -->
<child :msg="parentMsg"></child>

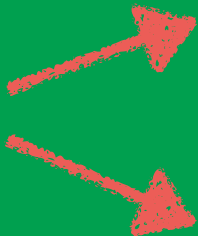
<!-- 双向绑定 -->
<child :msg.sync="parentMsg"></child>

<!-- 单次绑定 -->
<child :msg.once="parentMsg"></child>
```



# Vue组件分类实践

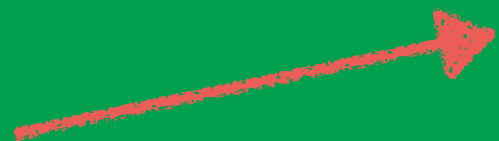
- 页内元素（分治）



- 公用元素（复用）



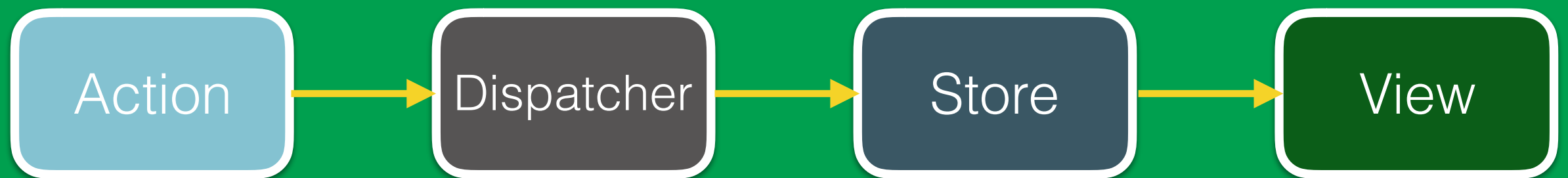
- UI控件



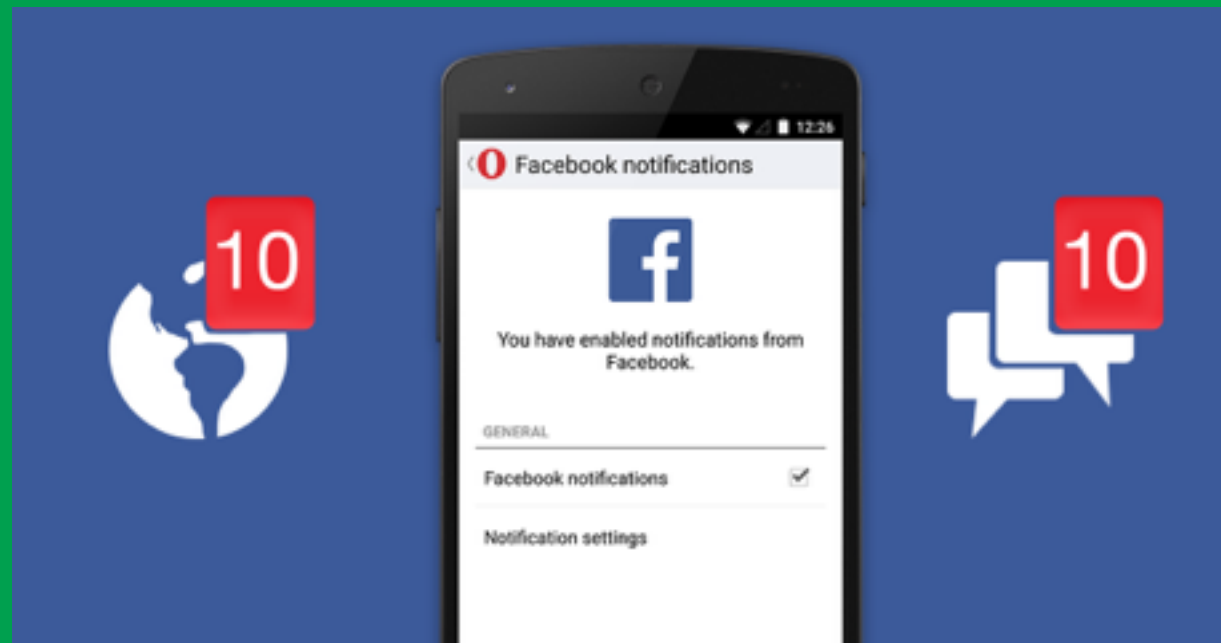


# Flux

Flux是Facebook用来构建用户端的web应用的应用程序体系架构。它通过利用数据的单向流动为React的可复用的视图组件提供了补充。



# Flux解决的问题



多个数据源的数据的汇总同步bug（跨组件）



支付方式

全额支付

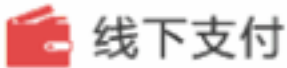
更多支付方式

选择您的付款方式

平台支付



线下支付



立即支付

支付金额: ¥6500.02

## 支付方式

全额支付

分次支付

选择本次支付金额

3000

+

修改

点击+/-调节金额，每次幅度500元，最低调至3000元，不足则按实际付款金额

收起支付方式

选择您的付款方式

平台支付

支付宝

UnionPay

在线支付

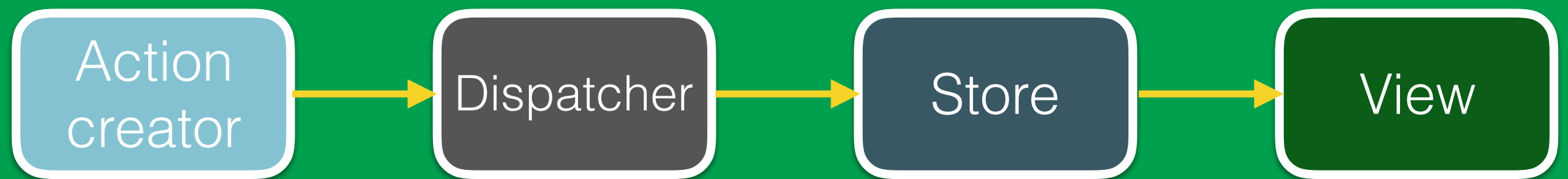
Online Payment

线下支付

线下支付

立即支付

支付金额: ¥6500.02



起草行政令

生成  
格式化  
action

把政令呈给慈禧

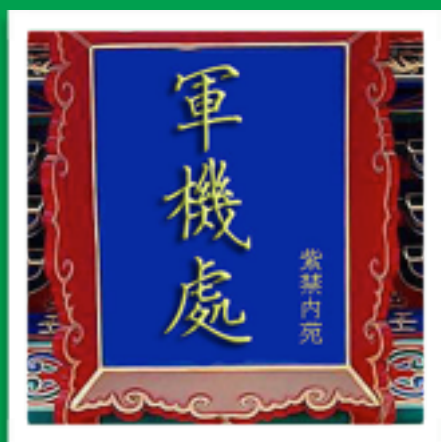
action  
按顺序  
发给  
store

决定执行与否

只执行  
必要的  
action

执行行政令

拿到  
store的  
数据变更  
render



军机处



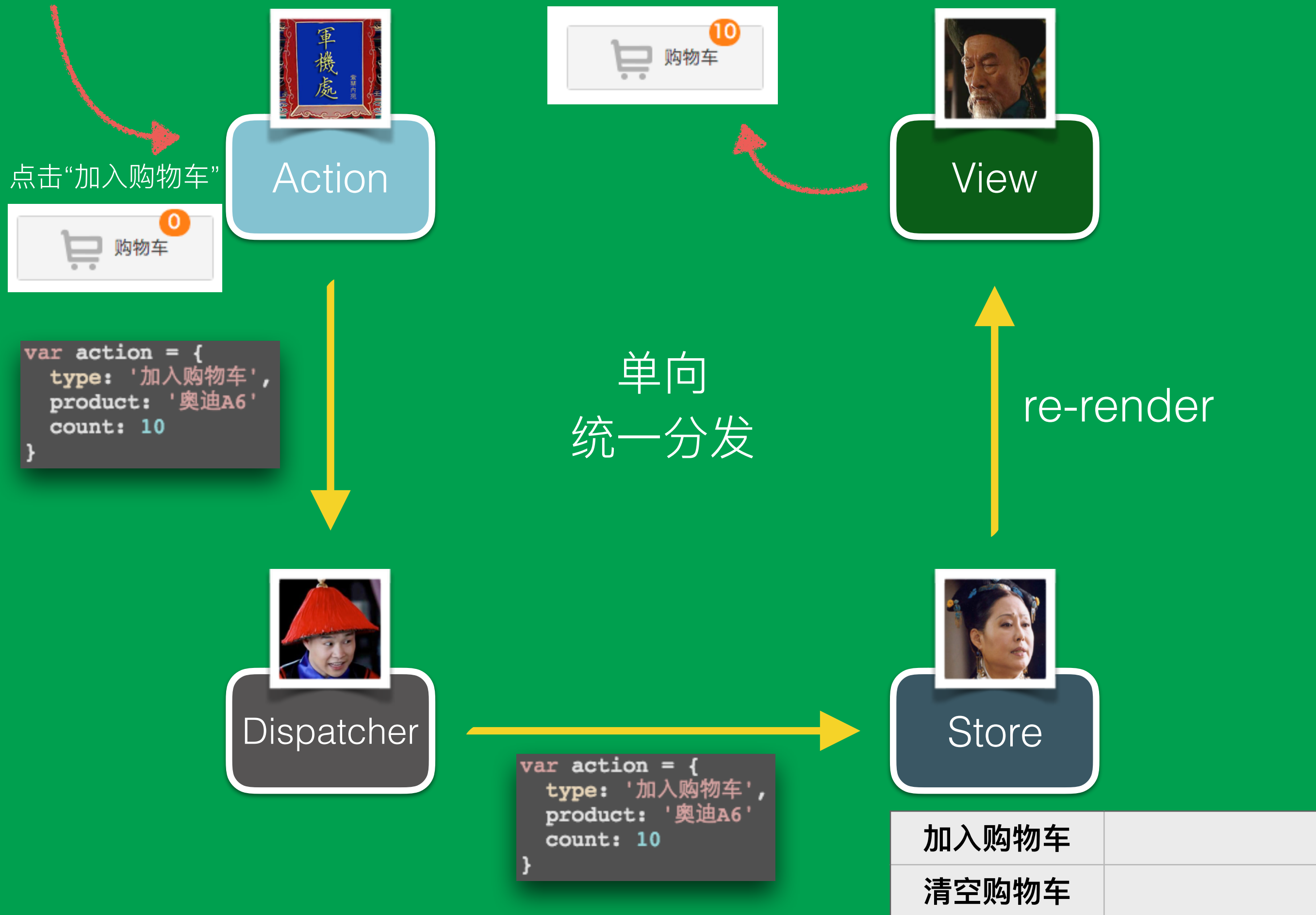
太监



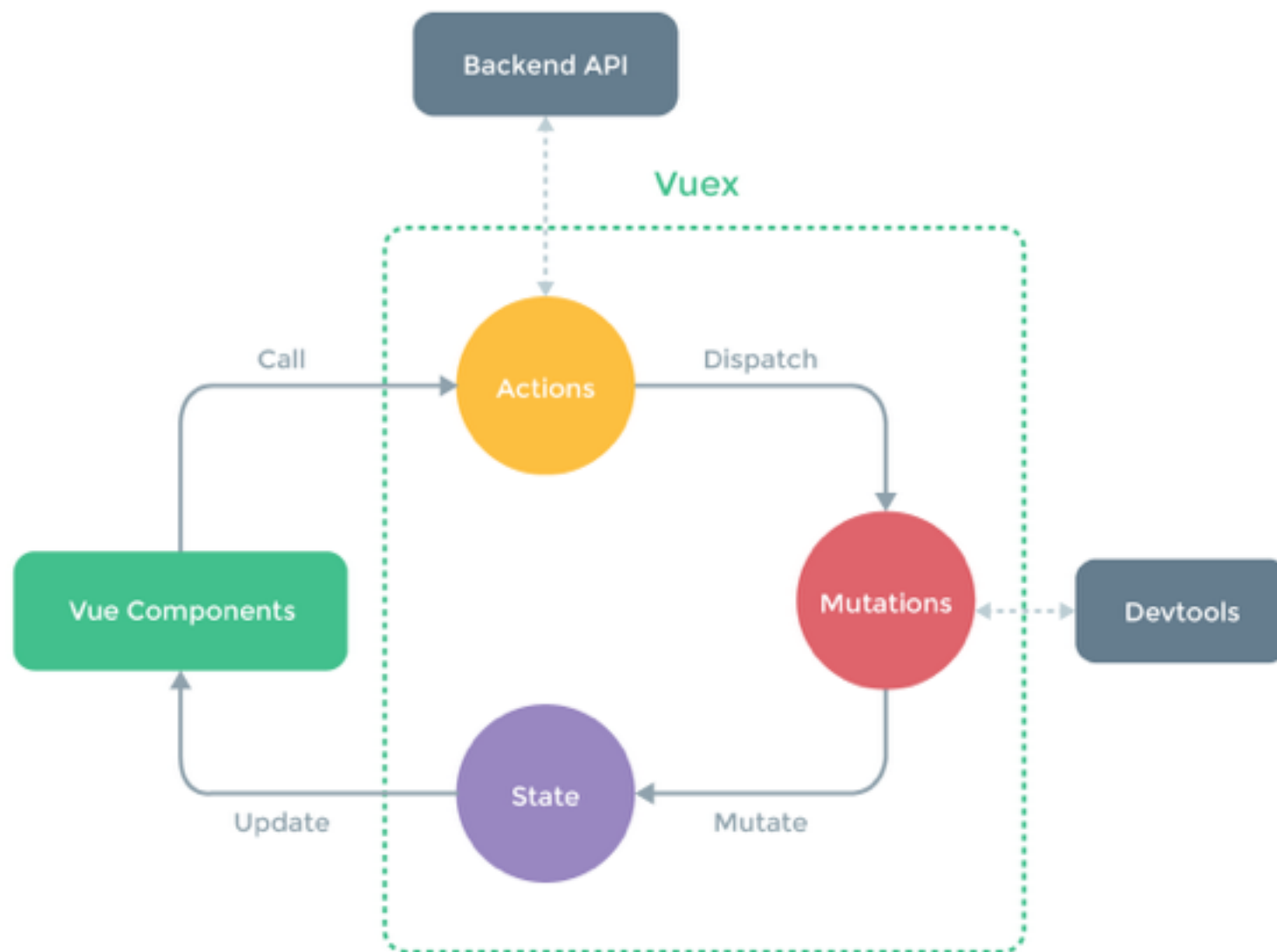
慈禧太后



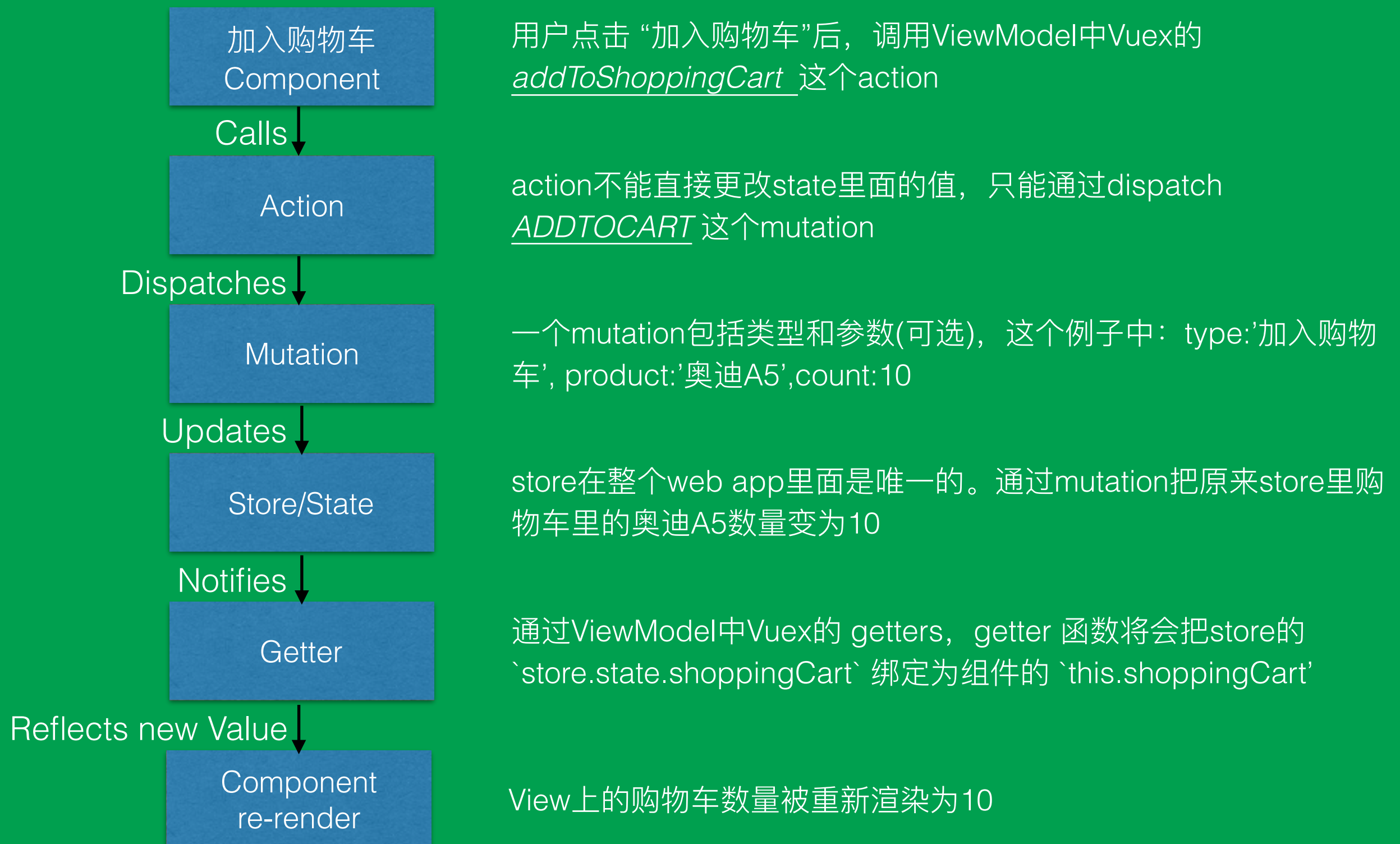
李鸿章



# Vuex



# Vuex Flow with ShoppingCart





# 何时使用Vuex

- 应用层级状态(application level state)
- 多数据源的同步汇总

```
/vuex
├── modules
│   ├── checkout.js
│   └── header.js
├── actions.js
├── mutation-types.js
└── store.js
```



# Vue-router & SPA

SPA（单页应用）是指在浏览器中运行的应用，在使用期间页面不会重新加载。

Vue-router可以和Vue深度结合已达到良好的单页应用开发体验。

# Vue-router



```
<template>
  <xxnr-header></xxnr-header>
  <div class="mainContainer">
    <router-view keep-alive></router-view>
  </div>
</template>

<script>
  import XxnrHeader from './components/XxnrHeader.vue'
  import store from './vuex/store'

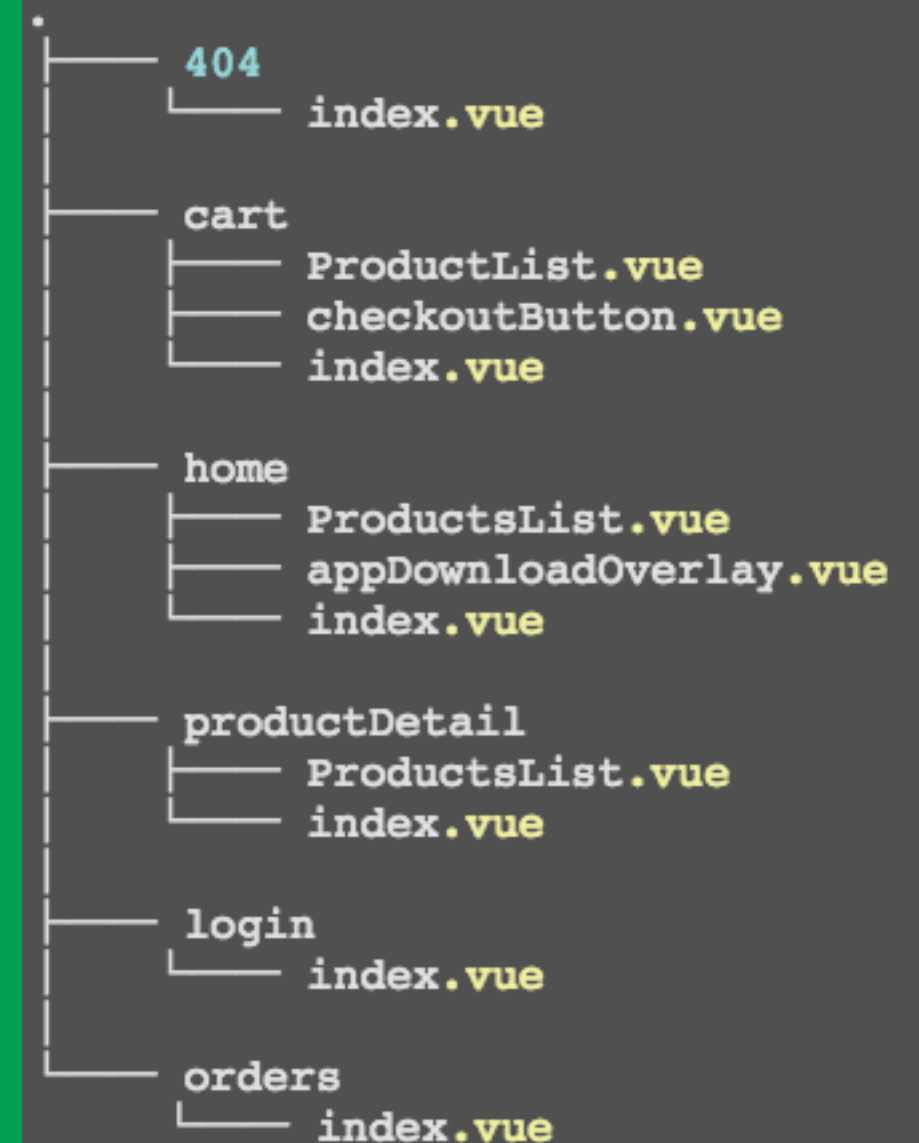
  export default {
    store,
    components: {
      XxnrHeader
    }
  }
</script>

<style>
  /*some common css style */
  body {
    font-family: "Microsoft YaHei", "Lucida Sans Unicode", "Myriad Pro",
    background-color: #fafafa;
    color: #595757;
  }
  input {
    border: none;
    _border: 0;
    outline: none;
    background: none;
    font-family: "Microsoft YaHei";
    border-radius: 0;
    opacity: 1;
  }
</style>
```

app.vue

# router-config

```
export function configRouter (router) {  
  // normal routes  
  router.map({  
    '/home': {  
      component: require('./components/home/index.vue')  
    },  
    '/cart': {  
      component: require('./components/cart/index.vue')  
    },  
    '/productDetail': {  
      component: require('./components/productDetail/index.vue')  
    },  
    '/login': {  
      name: 'login', //具名路由  
      component: require('./components/login/index.vue')  
    },  
    '/orders': {  
      component: require('./components/orders/index.vue'),  
      auth: true  
    },  
    '/404': {  
      component: require('./components/404/index.vue')  
    }  
  });  
}
```



# code example

```
<template>
  <div class="mobile-header">
    {{headerTitle}}
    <a v-link="{path: '/orders'}"></a>
  </div>
</template>
```

header

```
<template>
  <xxnr-header></xxnr-header>
  <div class="mainContainer">
    <router-view keep-alive></router-view>
  </div>
</template>
```

app.vue

components/orders/index.vue

# Authentication

```
'/orders': {  
  component: require('./components/orders/index.vue'),  
  auth: true
```

```
router.beforeEach((transition) => {  
  if (transition.to.auth) {  
    const user = getCookie('__user');  
    if(!user){  
      let redirect = encodeURIComponent(transition.to.path);  
      transition.redirect('/login?redirect=' + redirect);  
      //redirect 作为参数，登录之后跳转回来  
    }else{  
      transition.next();  
    }  
  }else{  
    transition.next();  
  }  
});
```

# 整个目录结构

```
components
├── cart
│   ├── CheckoutButton.vue
│   └── index.vue
├── home
│   ├── appDownloadOverlay.vue
│   └── index.vue
├── publicElements
│   ├── ProductList.vue
│   ├── xxnrFooter.vue
│   └── xxnrHeader.vue
└── xxnr_mobile_ui
    ├── xxnrAlert.vue
    └── xxnrToast.vue

vuex
├── actions.js
├── modules
│   ├── checkout.js
│   └── header.js
├── mutation-types.js
└── store.js

route-config.js

App.vue
main.js

utils
├── authService.js
└── vueFilter.js
```

Vue  
components

Vuex

Vue-router

utils

Vue App

实践



# 总结

- MVVM
- Vue
  - 数据驱动 + 组件化
- 组件化
  - 分治 > 复用
- Flux
  - 单向 + 统一分发
- Vue-router



# Thanks

掘金： 阴明 《Vue.js 的组件化编程》



徐飞： Web应用组件化的权衡

Lin Clark: A cartoon guide to Flux

尤小右： 知乎， 视频