

基础篇





**扫码试看/订阅**

**《Vue开发实战》**

内容概述

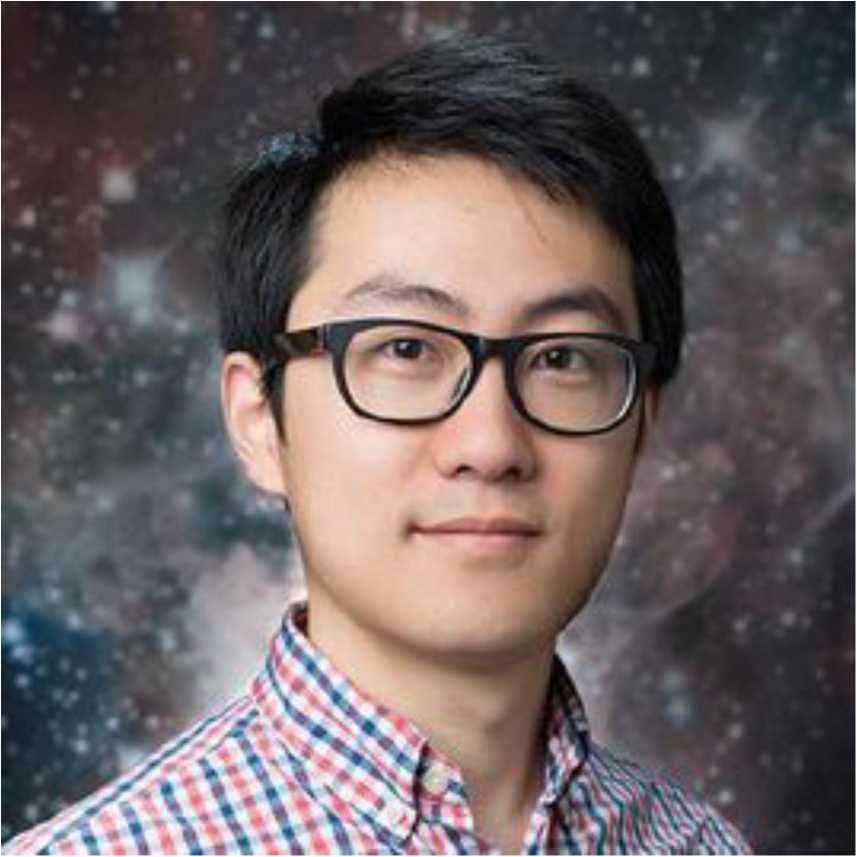


内容概述

•基础篇：Vue核心知识点 •生态篇：大型Vue项目所需的周边技术

•实战篇：开发基于Vue的Ant Design Pro

•福利篇：Vue 3.0相关知识介绍



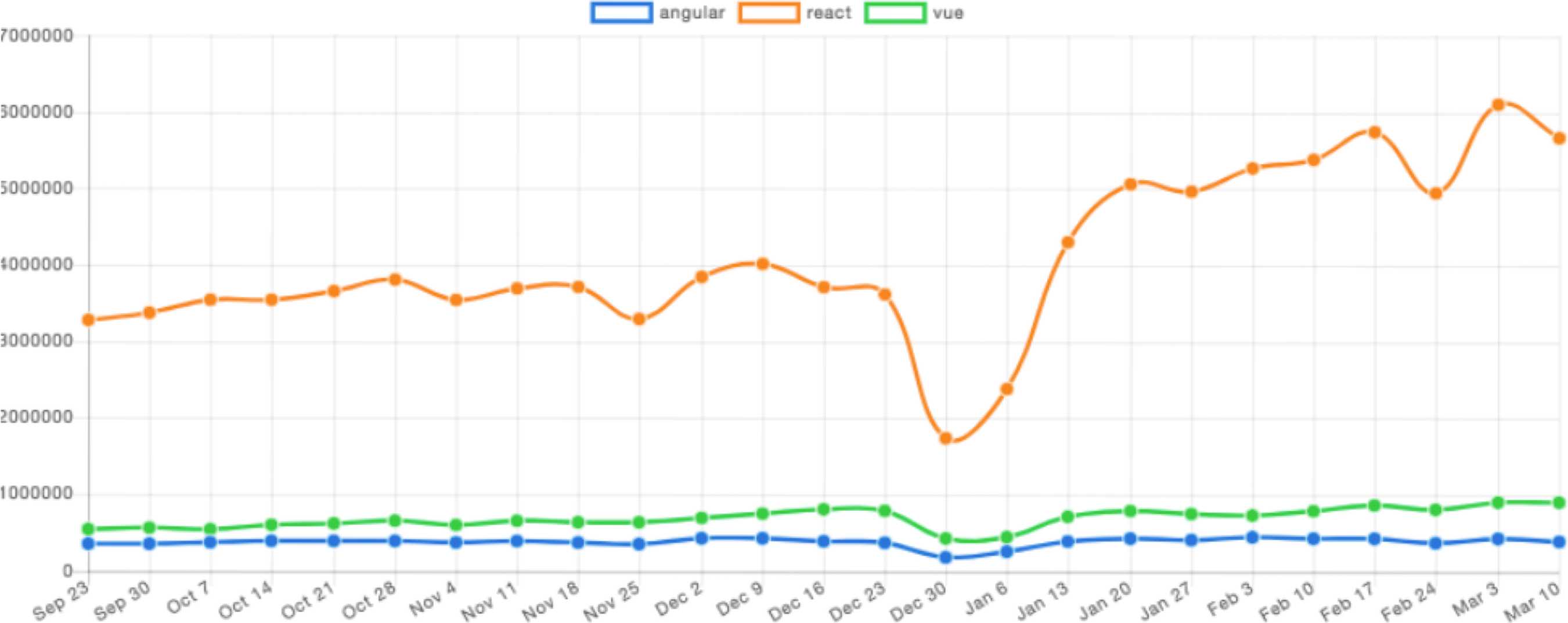
尤雨溪（Evan You

Vue简介



—款渐进式JavaScript框架

Vue vs Angular vs React



Stats

stars

forks

issues !

updated X

created

size W

angular

59413

28923

458

Mar *7,* 2019

Jan 6, 2010

**minified size 179.4 KB**

react

124050

22520

576

Mar 10, 2019

May 25, 2013

**minified size 6.5 KB**

vue

130197

18573

244

Mar 10, 2019

Jul 29, 2013

**minified size 63.3 KB**



衅劉巨甫•

溢醉股•

d；z6 + uiuj q>|0乙喜為叩堇•

草墩

总之，学就对了



编写第一个Vue程序

环境搭建

•浏览器：Chrome

• IDE : Visual Studio Code 或者 WebStorm • Node.js 8.9+ , npm

Vue.component 缺点

**•全局定义：**强制要求每个component中的命名不得重复

**•字符串模板：**缺乏语法高亮，在HTML有多行的时候，需要用到丑陋的\

**•不支持CSS:**意味着当HTML和JavaScript组件化时，CSS明显被遗漏 **•没有构建歩骤：**限制只能使用HTML和ES5 JavaScript,而不能使用预处理器，如Pug

(formerly Jade)和 Babel

Vue CLI

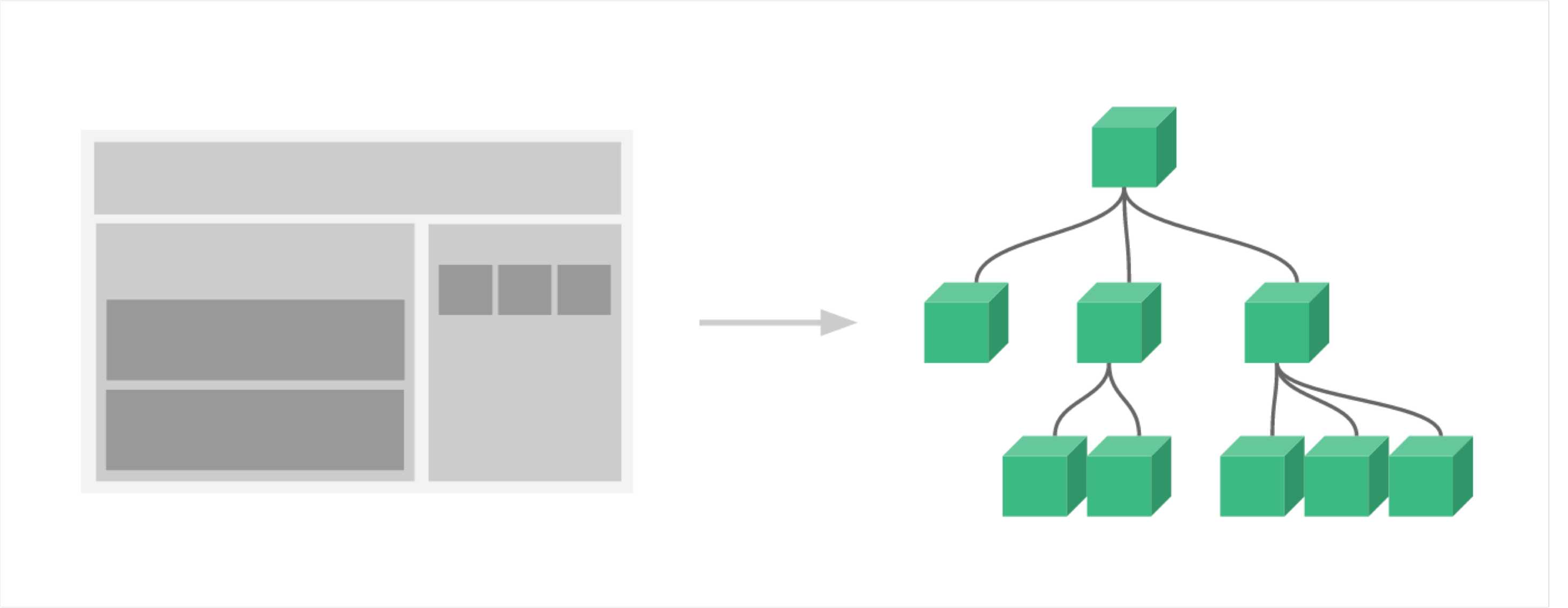
* npm install -g @vue/cli
* vue create my-app
* cd my-app • npm run serve

单文件组件

•完整语法咼亮

• CommonJS 模块 •组件作用域的CSS

组件的三大核心概念



Vue 组件=Vue 实)=new Vue(options)



属性、事件、插槽

自定义属性props

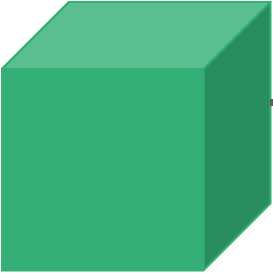
＞原生属性attrs

组件props中声明的属性

没有声明的属性z默认自动挂载到组件根元素上, 设置inheritAttrs为false可以关闭自动挂载

特殊属性class、style

挂载到组件根元素上*，*支持字符串、对象、数组等多种语法



•子组件为何不可以修改父组件传递的Prop，如果修改了，Vue是如何监控到属性的修改 并给出警告的？

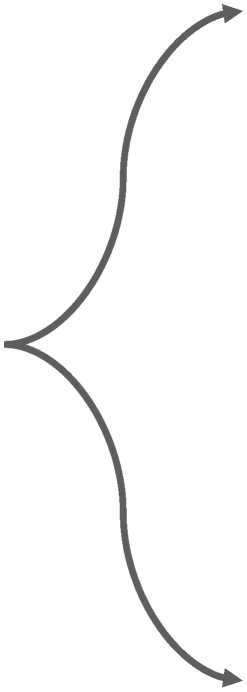
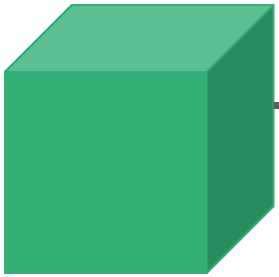
＞事件

普通事件

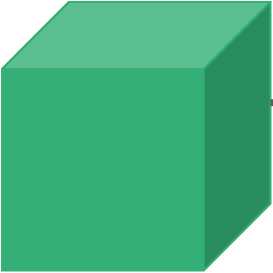
修饰符事件

@clickz @inputz @changez @xxx 等事件 z 通过 this.$emit( zxxxz z …)触发

@input.trimz @click.stopz @submit.prevent 等 z —般用于原生HTML元素z自定义组件需要自行开发支持



• this.$emit的返回值是什么



普通插槽

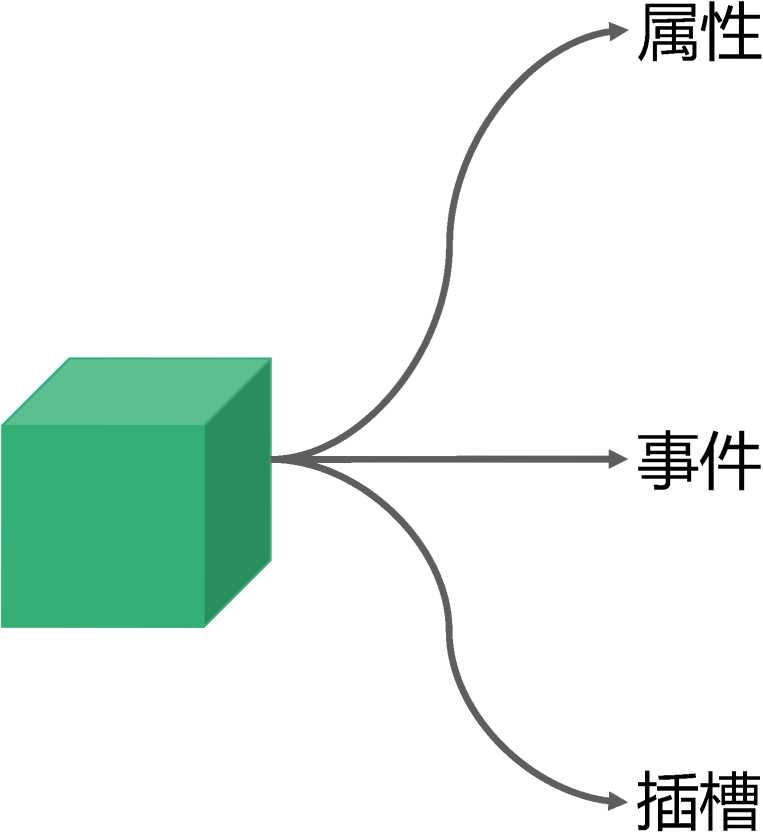
<template slot="xxx">...〈/template〉

<template v-slot:xxx>...〈/template〉

作用域插槽

<template slot="xxx" slot-scope="props">...〈/template〉

<template v-slot:xxx= "props">...〈/template〉



大属性

万物皆"属性"

定义属性props

~~原生属性attrs

特殊属性class、style

**＜**普通事件

修饰符事件

**＜**普通插槽

作用域插槽

•相同名称的插槽是合并还是替换？

双向绑定和单向数据流〃不冲突〃

model 更新 view

model

view

view 更新 model

model

model 更新 view

view

双向绑定or单向数据流

* Vue是单向数据流，不是双向绑定
* Vue的双向绑定不过是语法糖 • Object.defineProperty是用来做响应式更新的，和双向绑定没关系

语法糖

<PersonalInfo

v-model^'phonelnfo"

:zip-code. sync=,,zipCode11

/>

<PersonalInfo

:phone-info^'phonelnfo"

(achange="val => (phoneinfo = vat)"

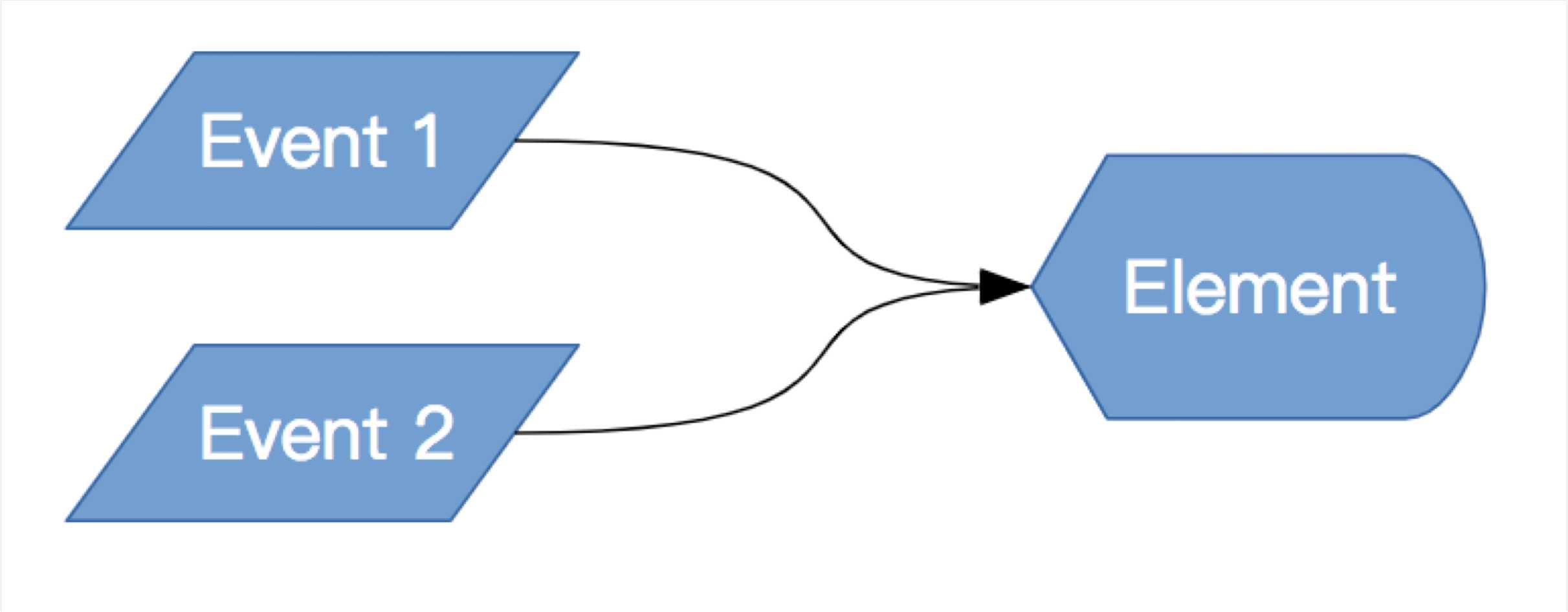
:zip-code=,lzipCode"

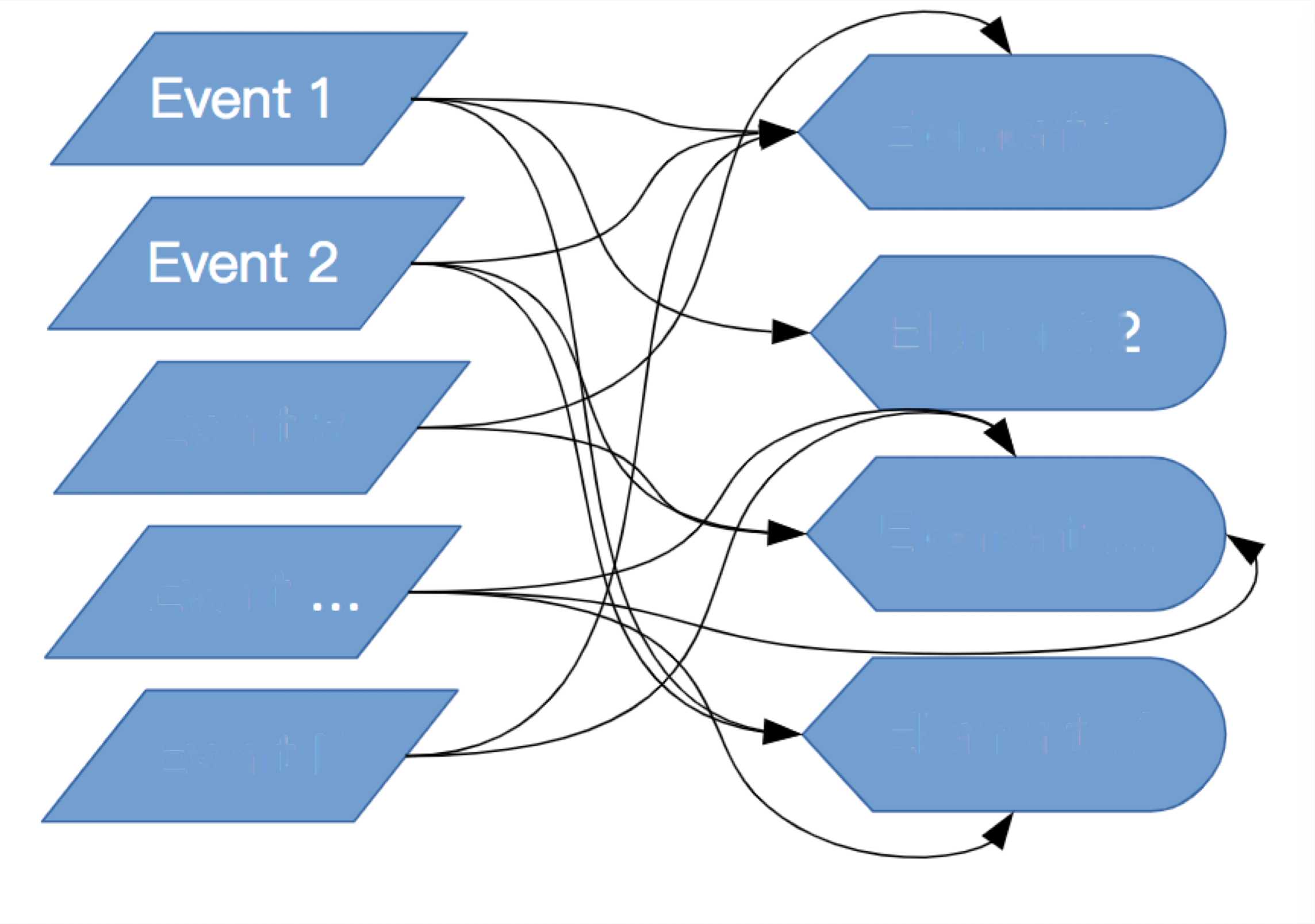
©update:zipCode=”val => (zipCode = vat)" />

•扩展Personalinfo Demo对手机号做非空且合法校验，如不合法则给出错误提示



理解Virtual DOM及key属性的作用





Event

Element 1

Element:

Event 3

Element...

Element M

Event N

再看Vue

Event 1

Event 3

Event

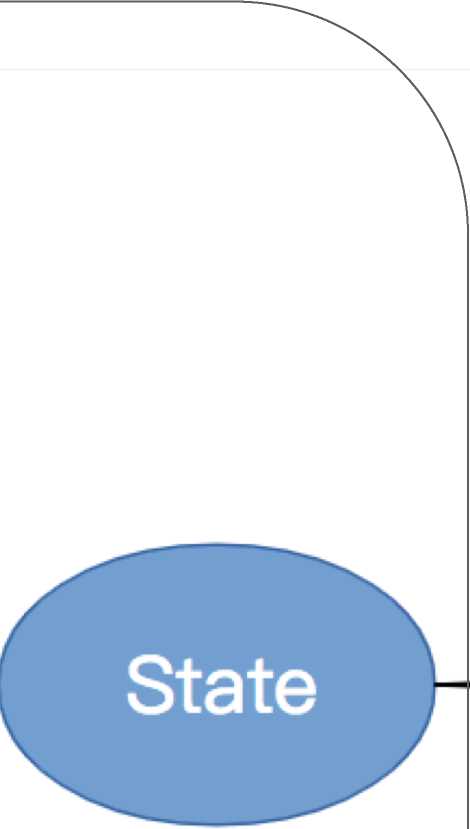
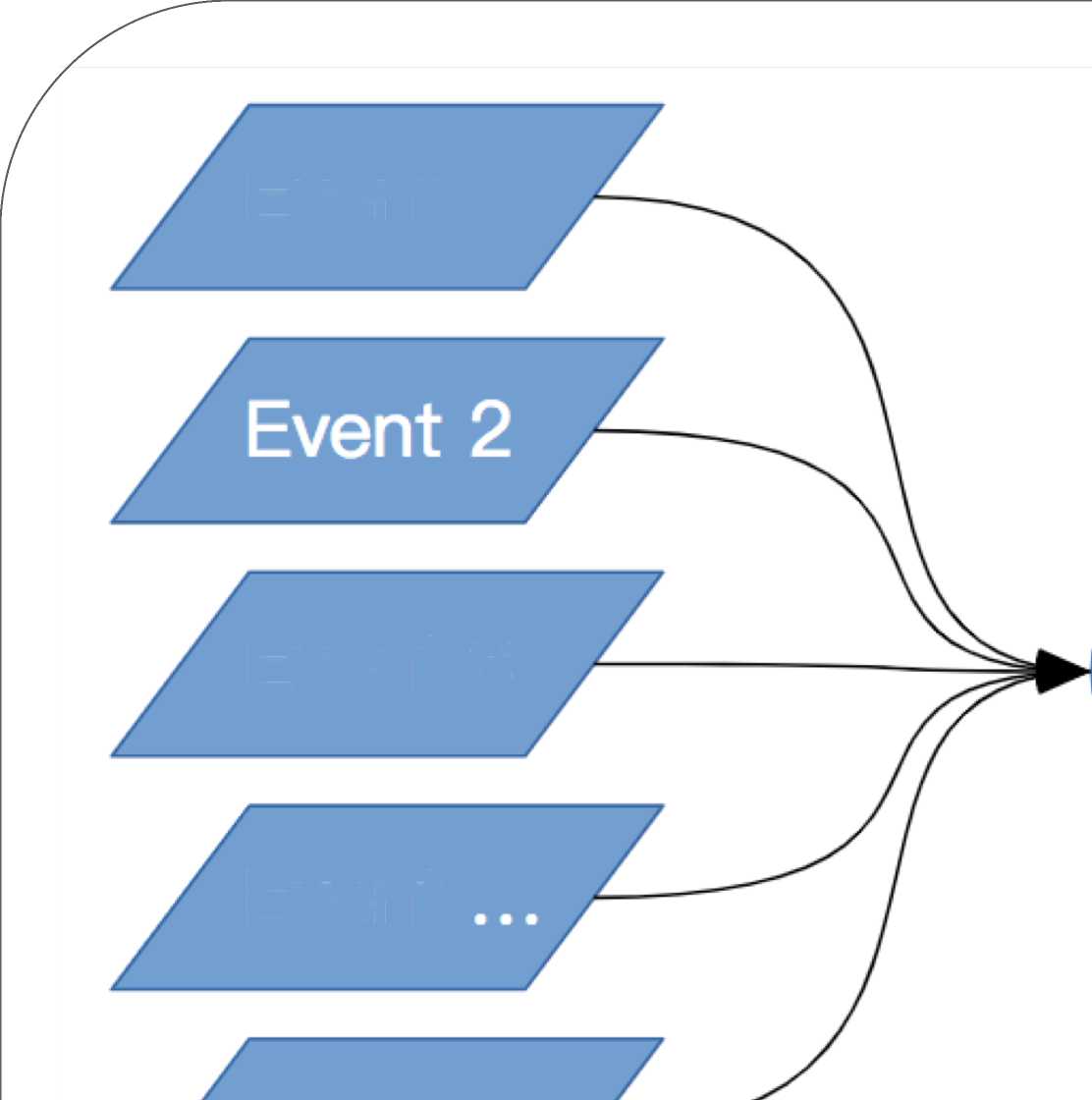
Event N

Element 1

Element:

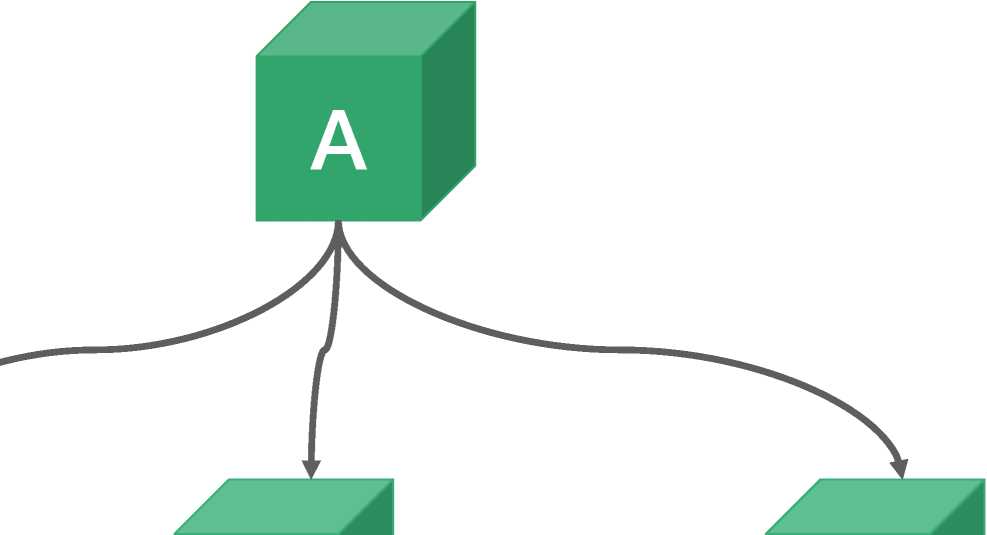
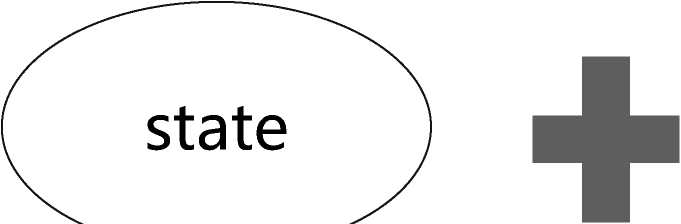
Element ...

Element M

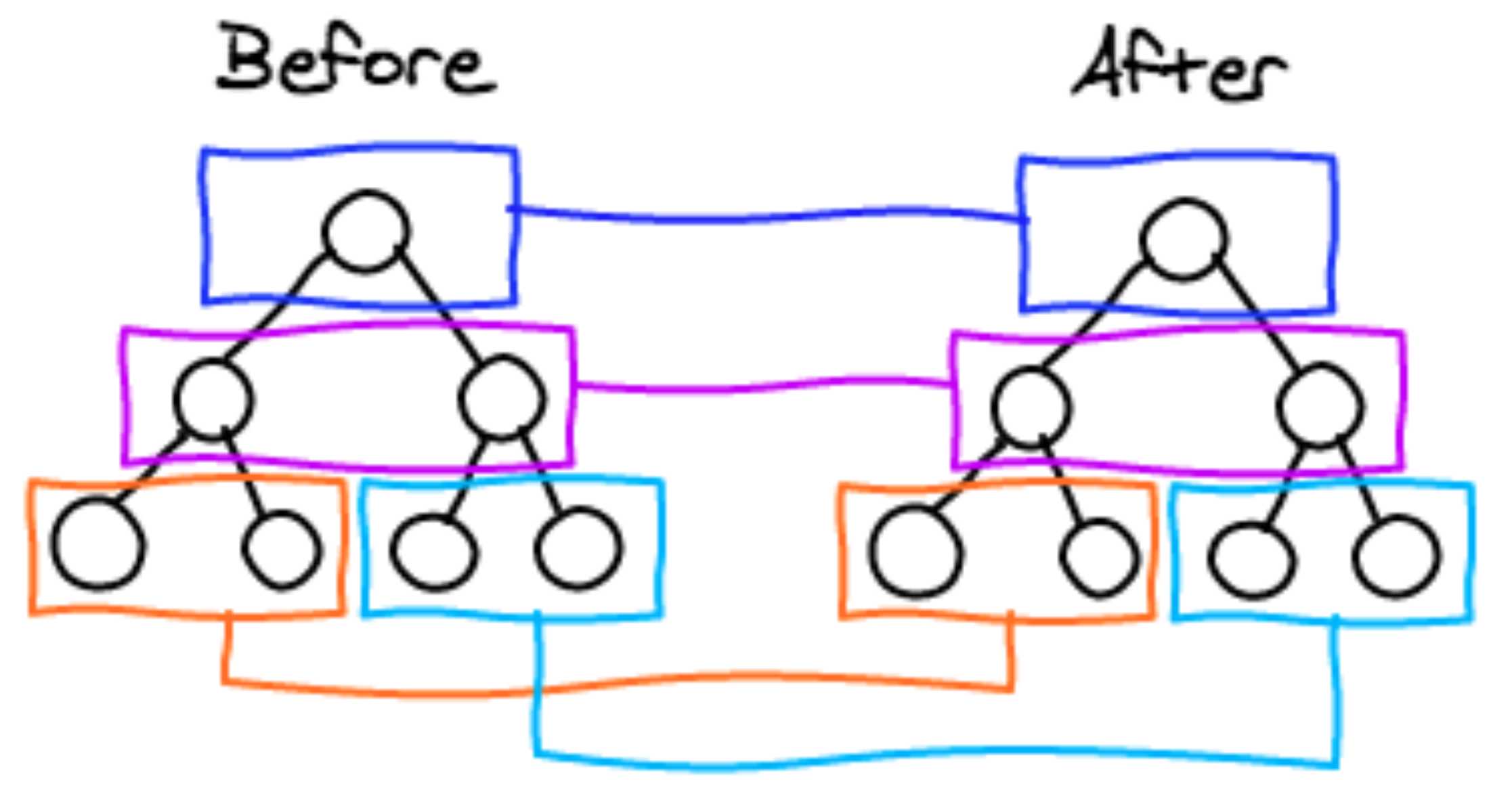


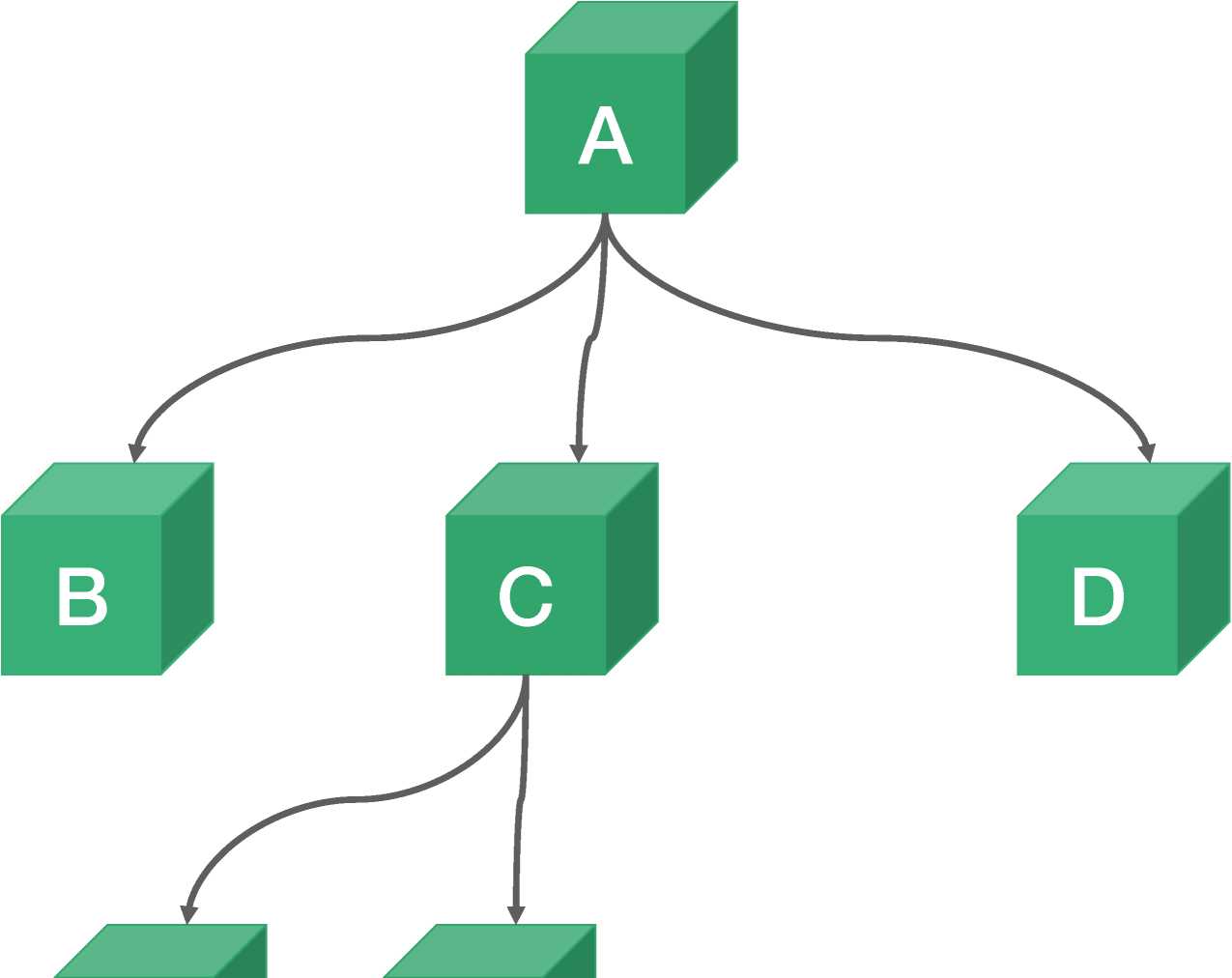
Virtual DOM

template

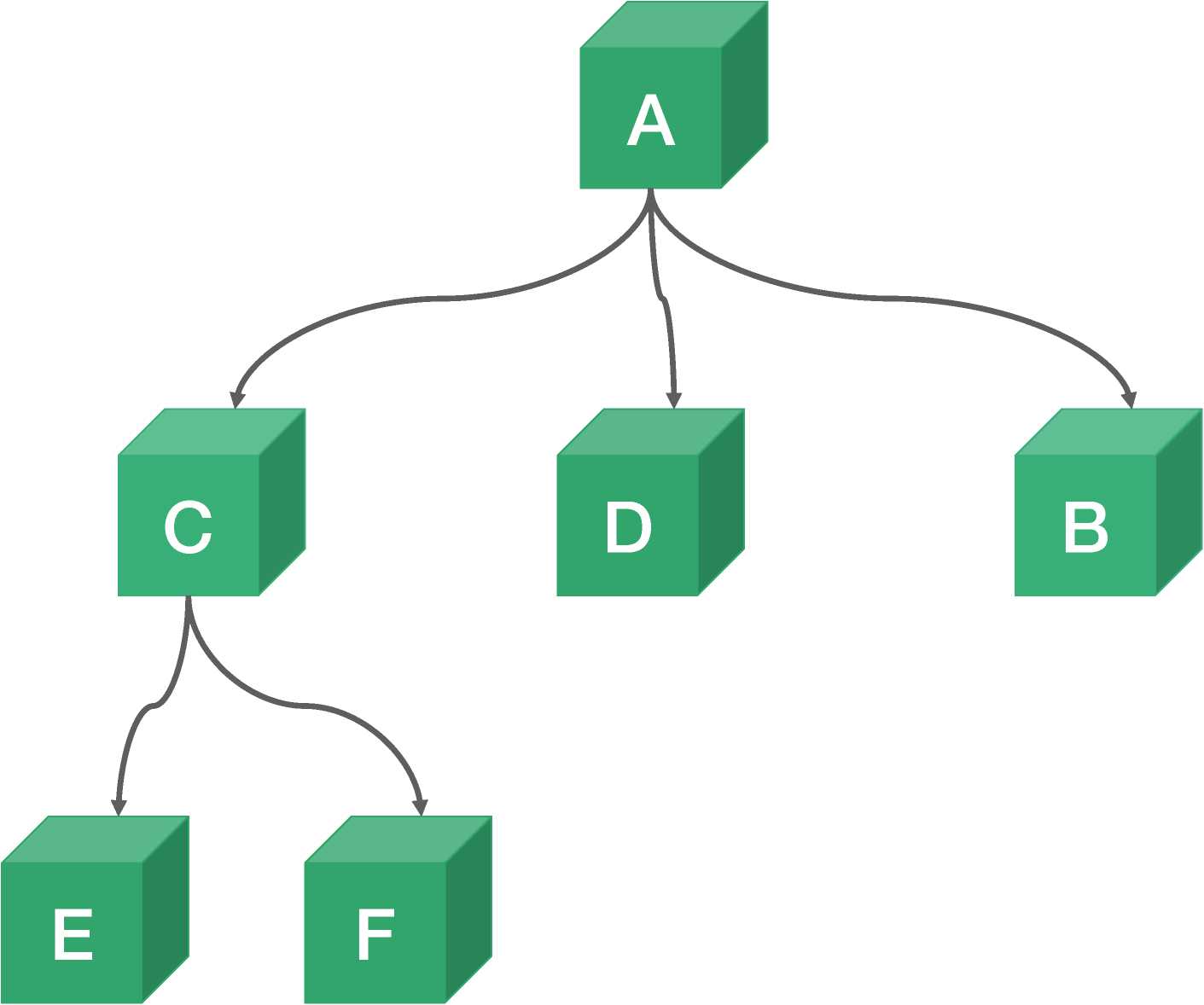


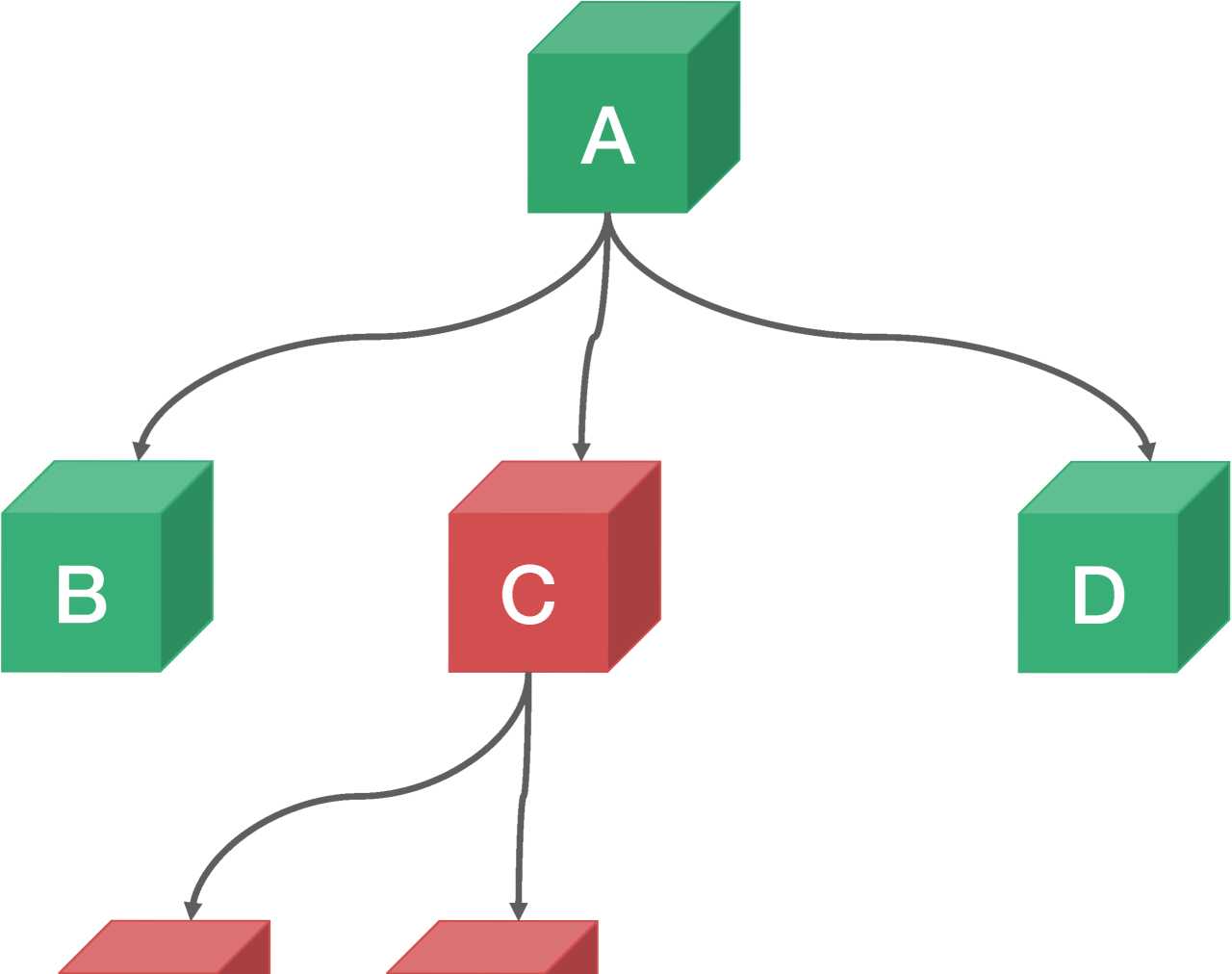
Virtual DO M Diff



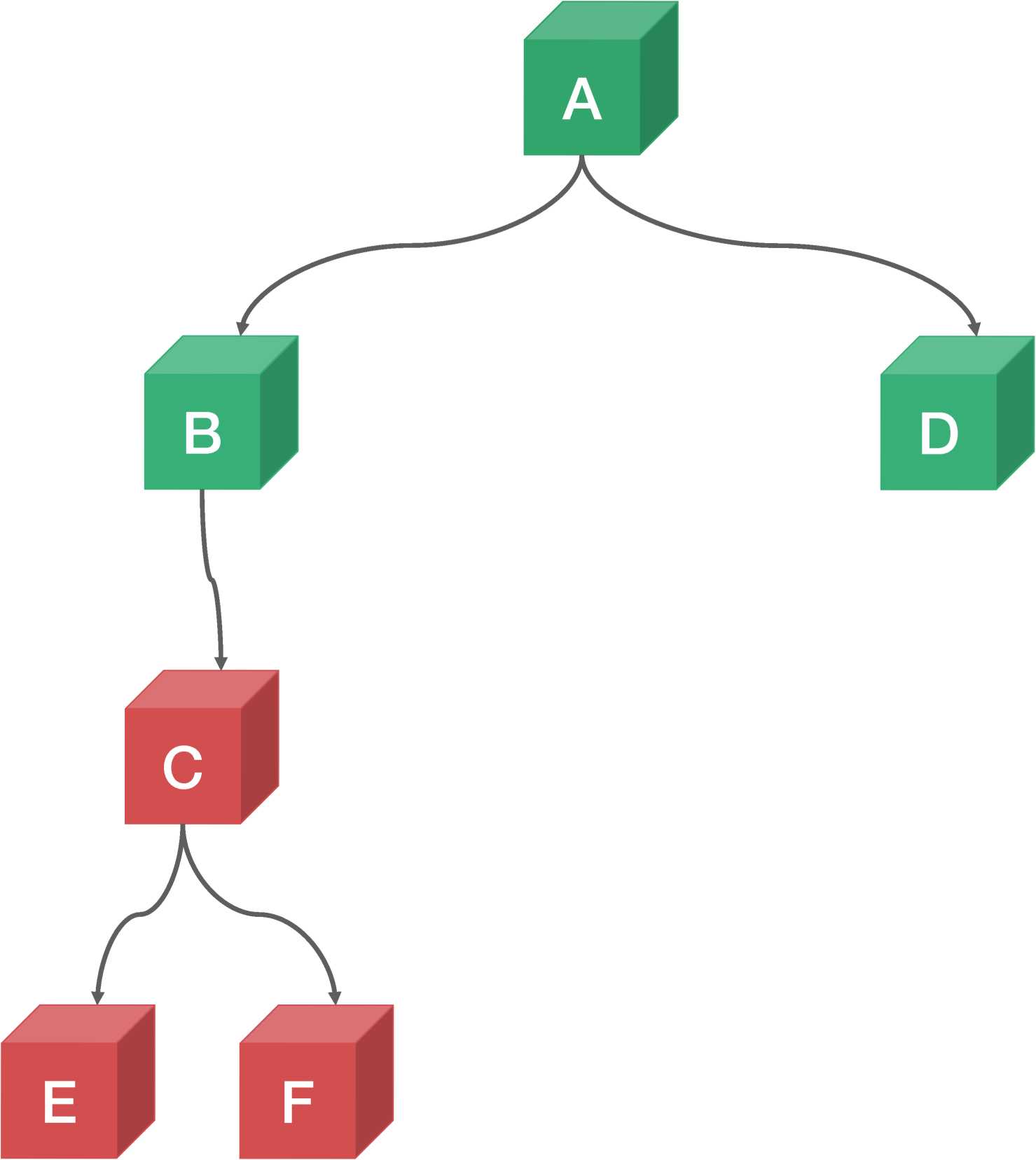


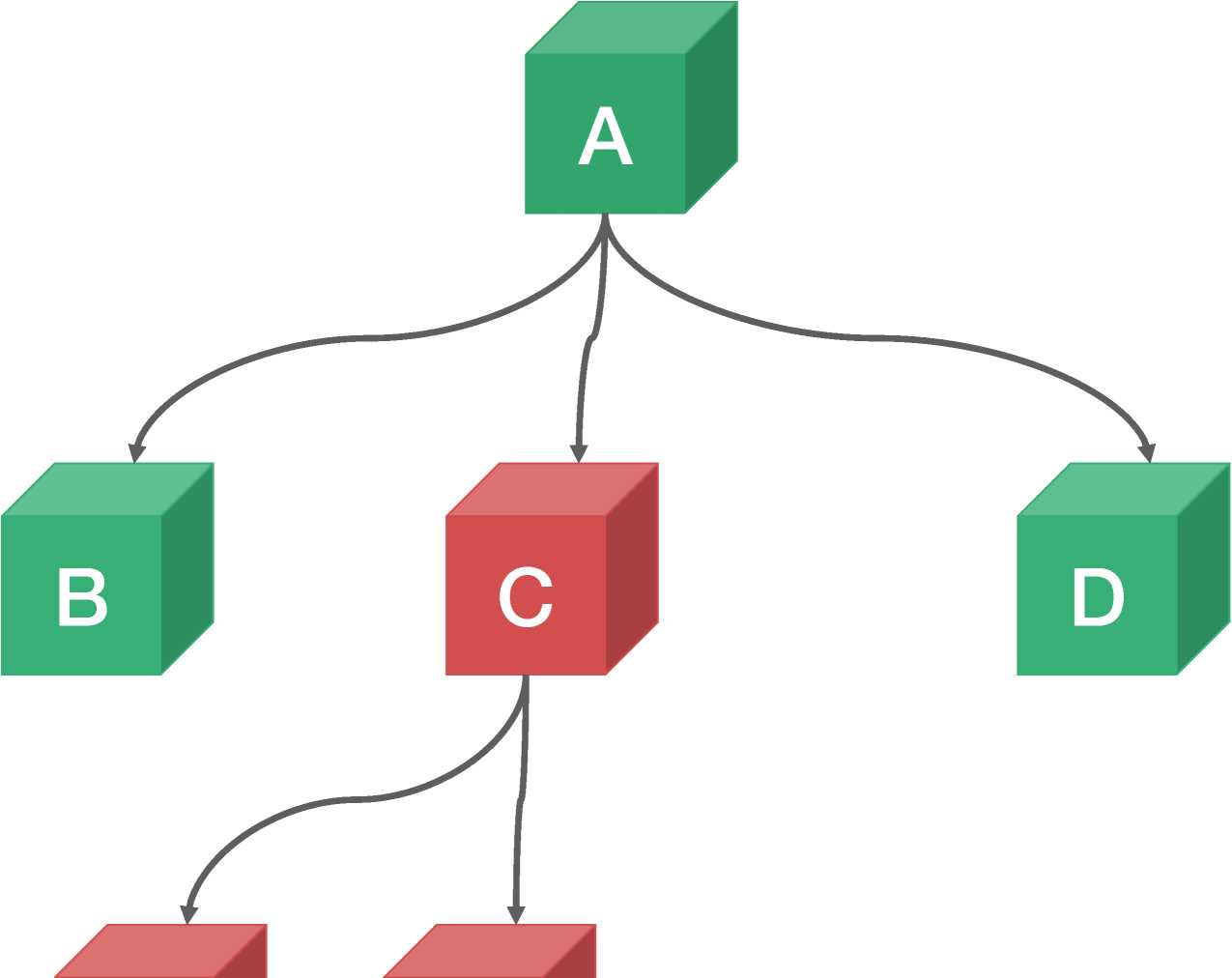
E



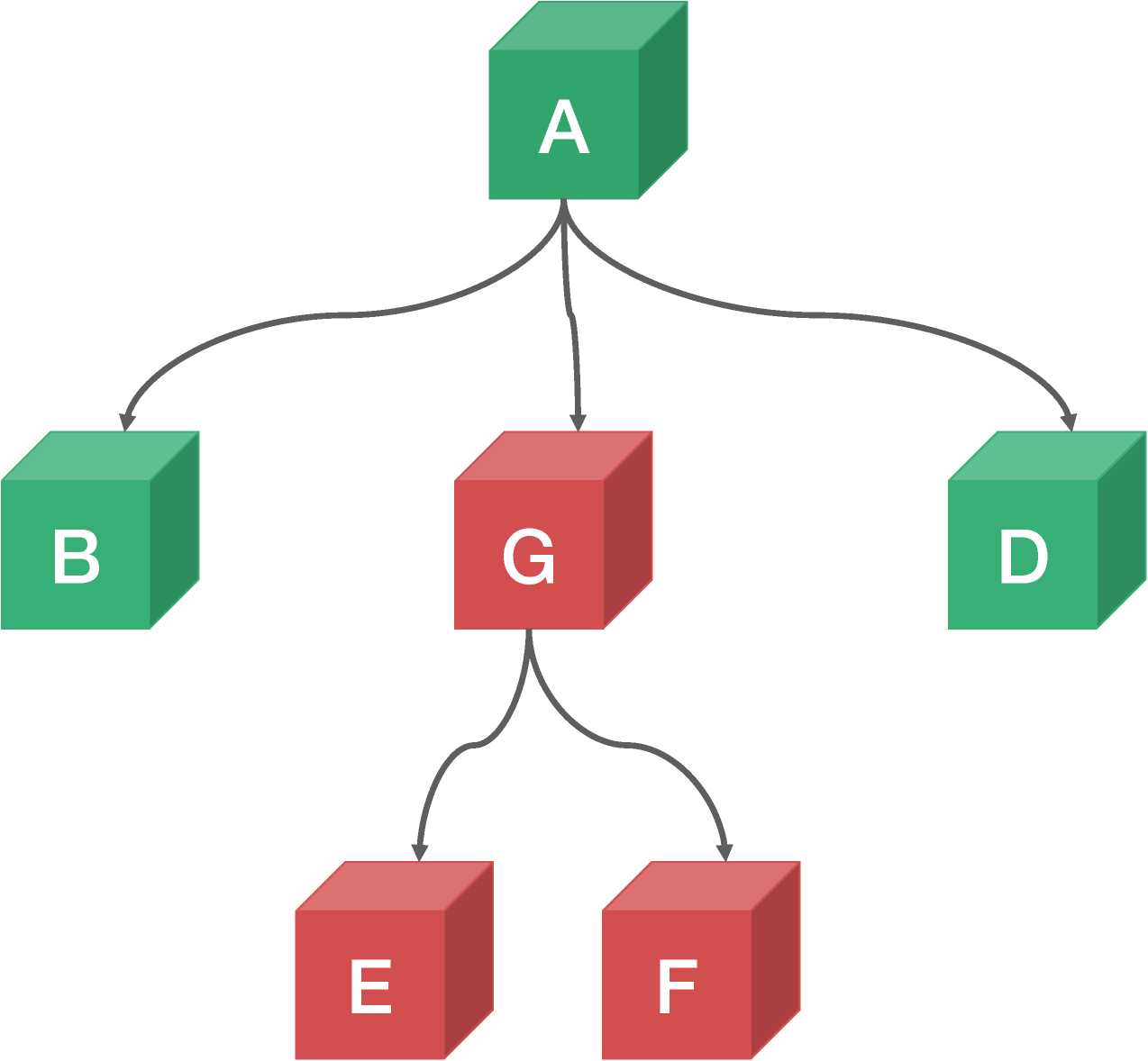


E

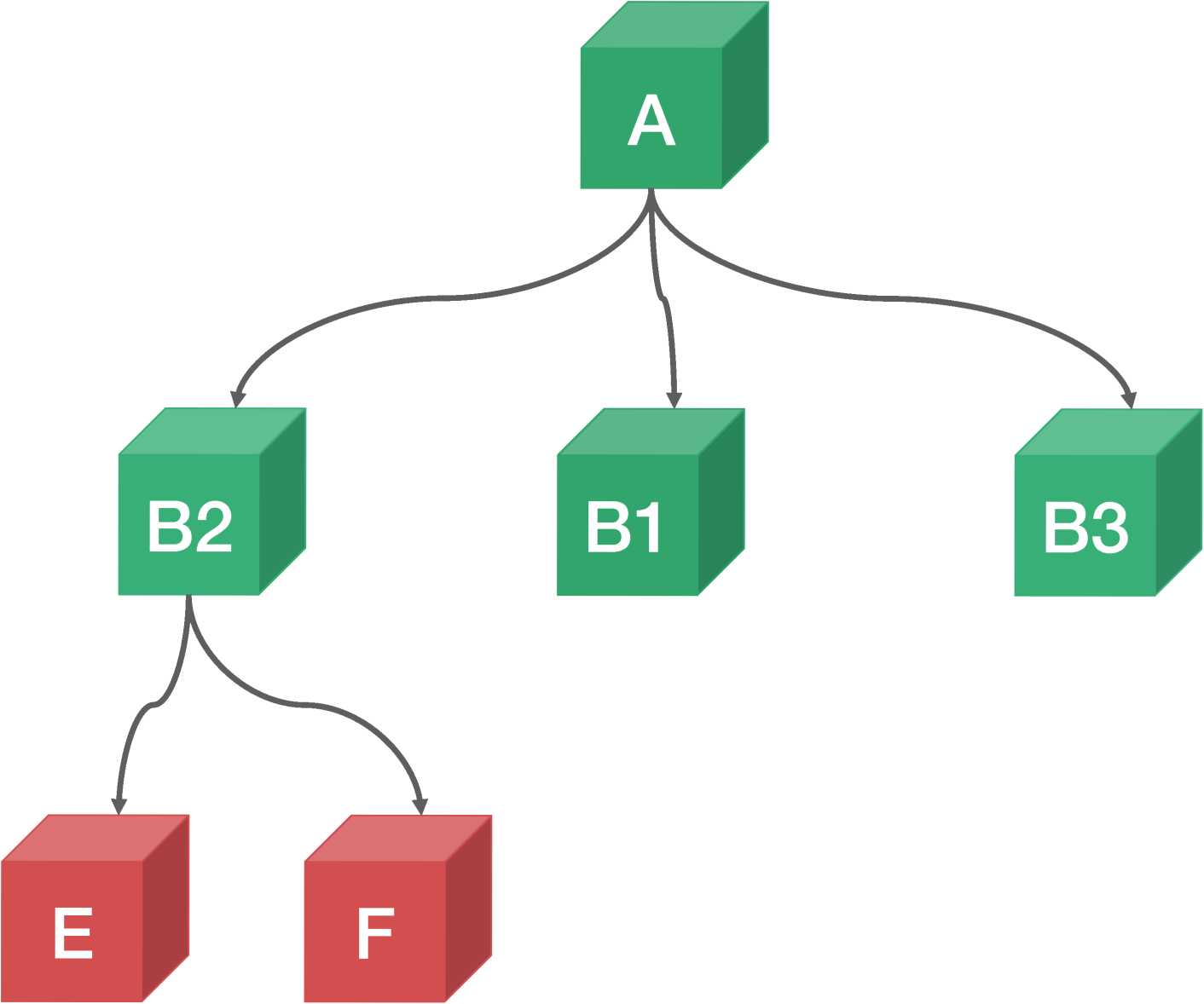
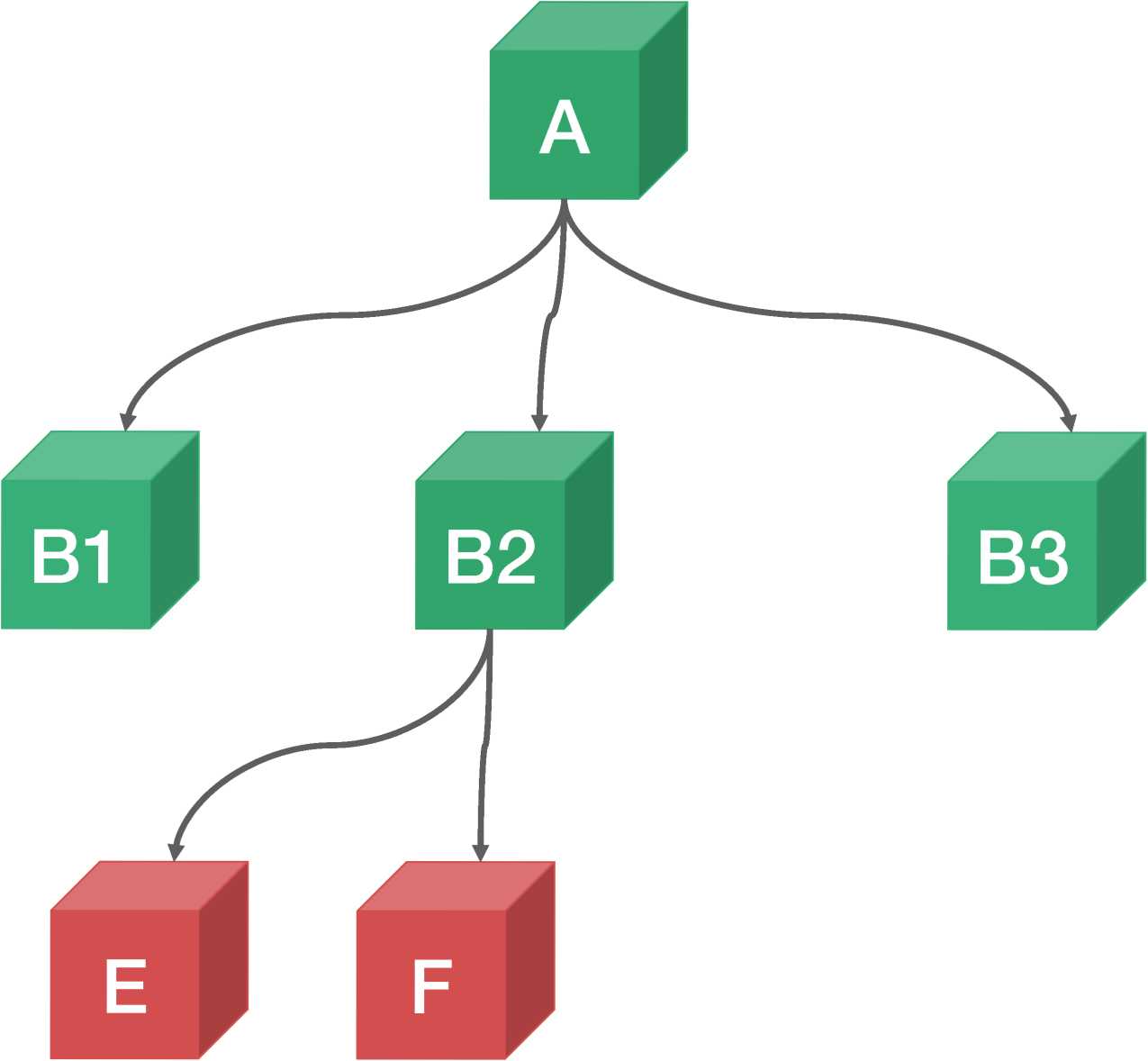




E

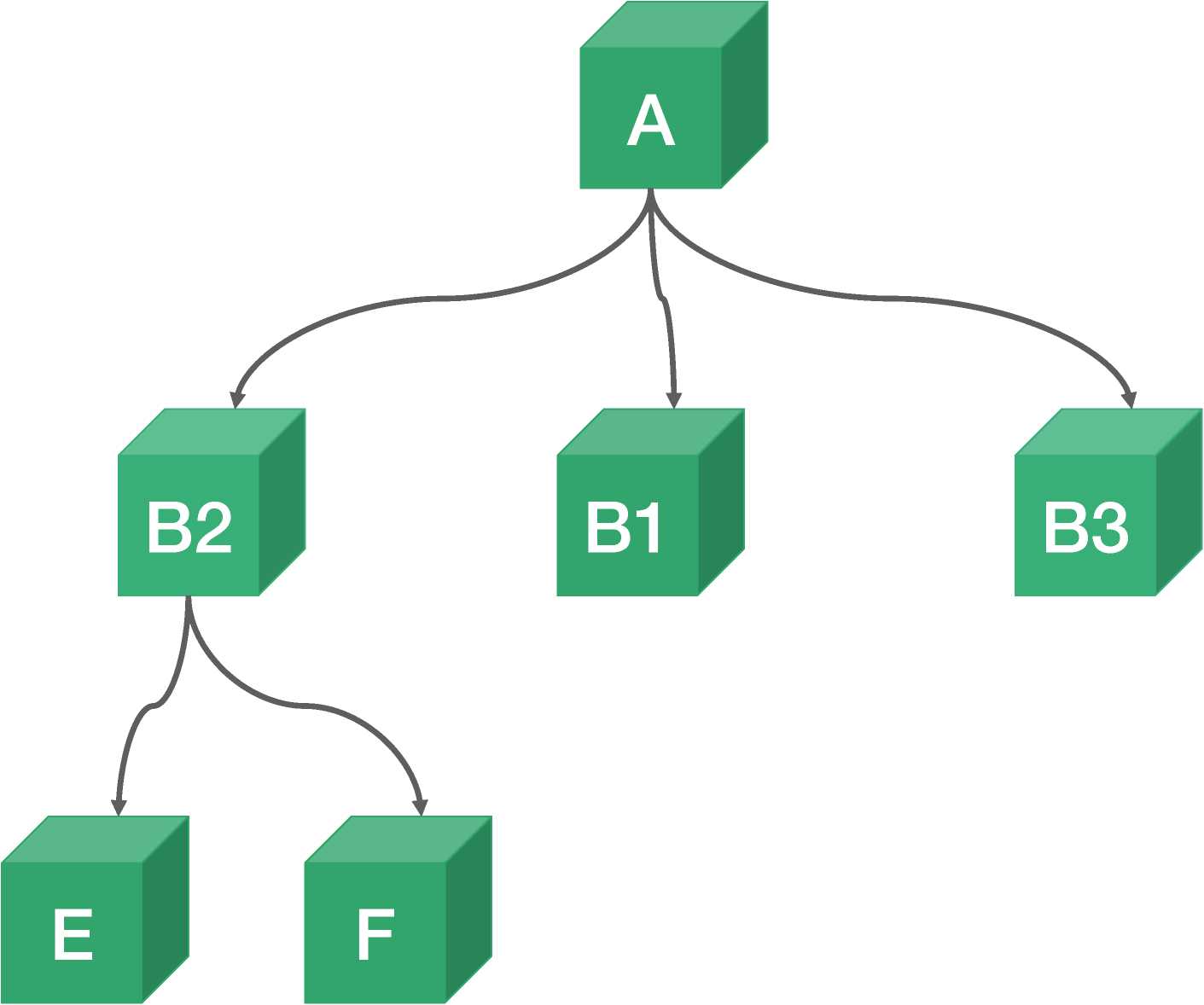
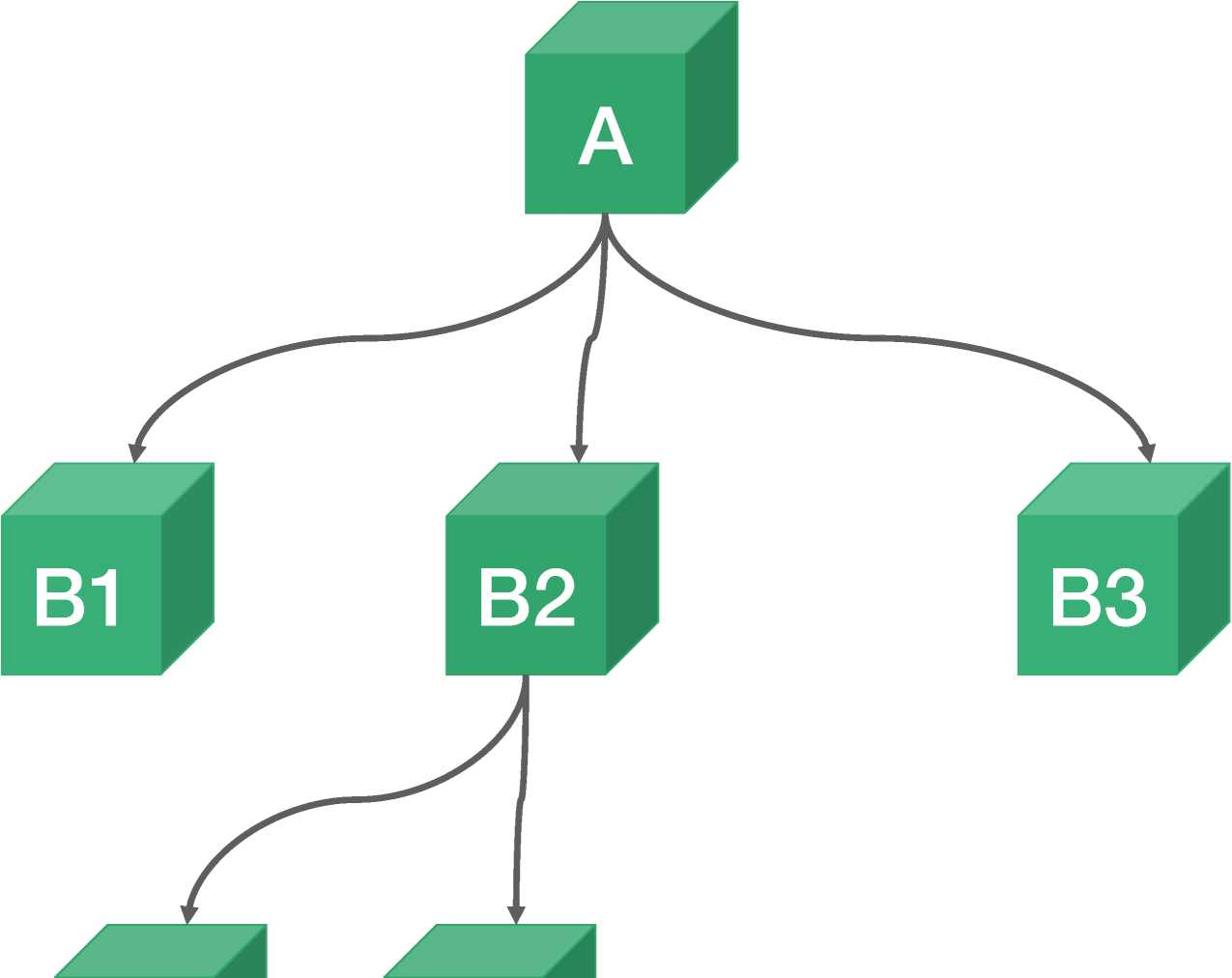


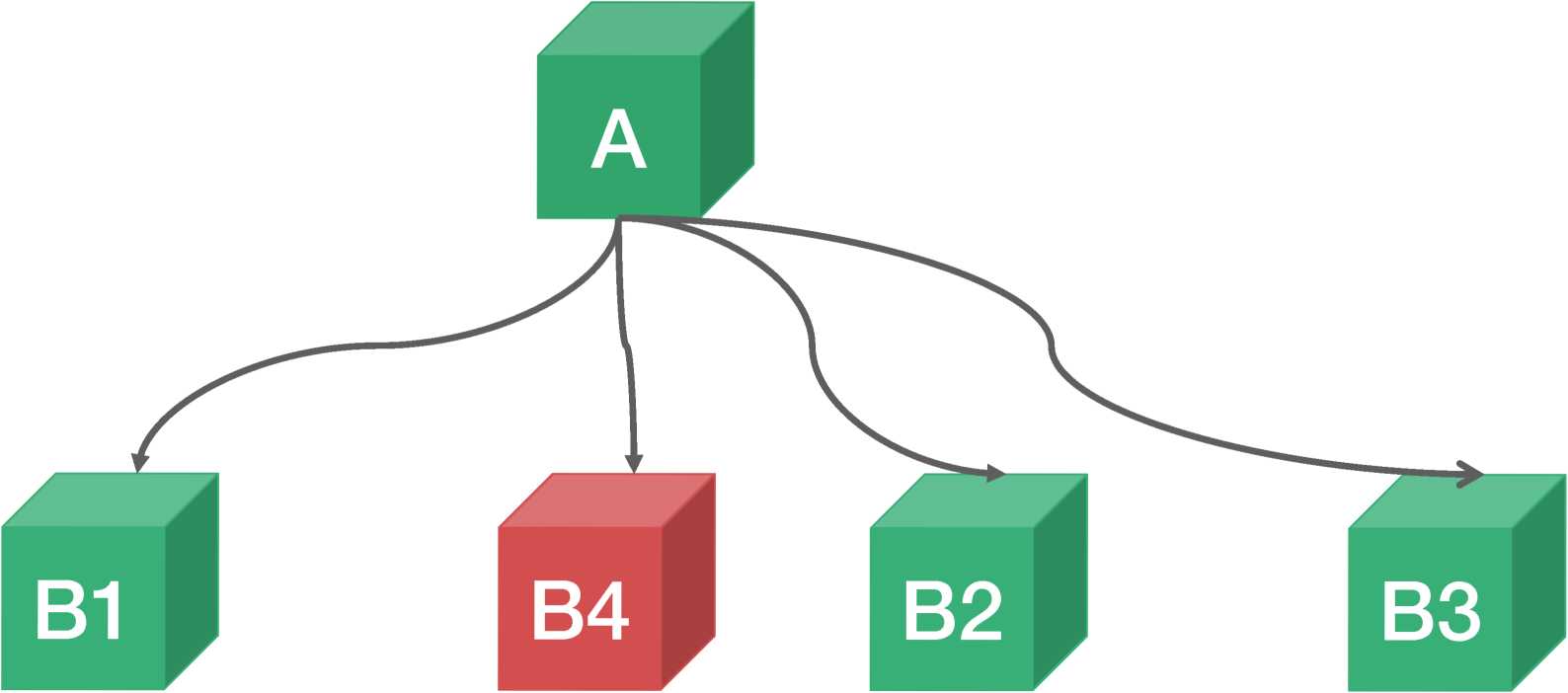
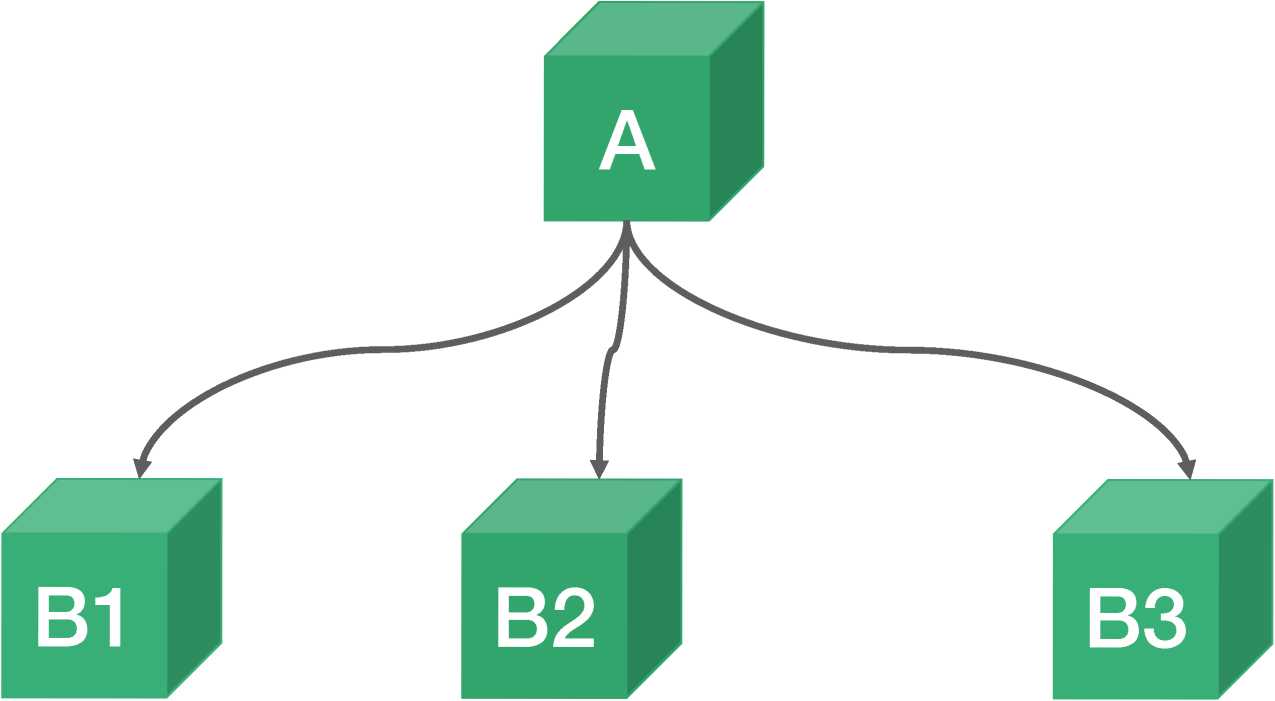
场景4 :更新删除新建（无key ）



场景5 :移动（有key ）

E





课后习题

•为什么不能用index作为key ?切记！切记！切记!

提示：不仅仅是性能问题，有些场景还会引入严重的Bug



如何触发组件的更新

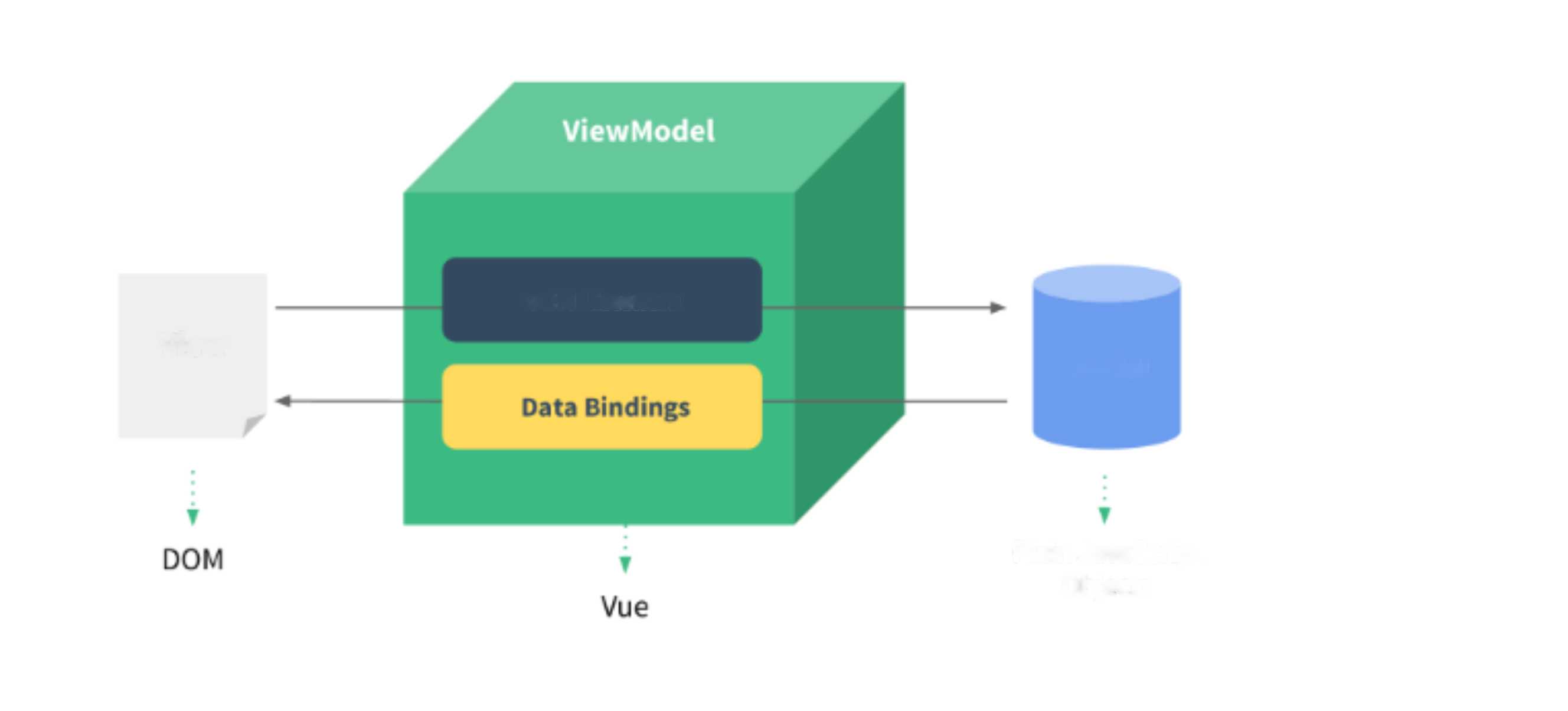
数据驱动

**View**

Plain JavaScript Objects

**Model**

**DOM Listeners**



任何直接更改**DOM**的行为都是在作死！ ！！

**Q**极客时间

数据来源（单向的）

来自父元素的属性

来自组件自身的状态data

来自状态管理器，如vuex, Vue . observable

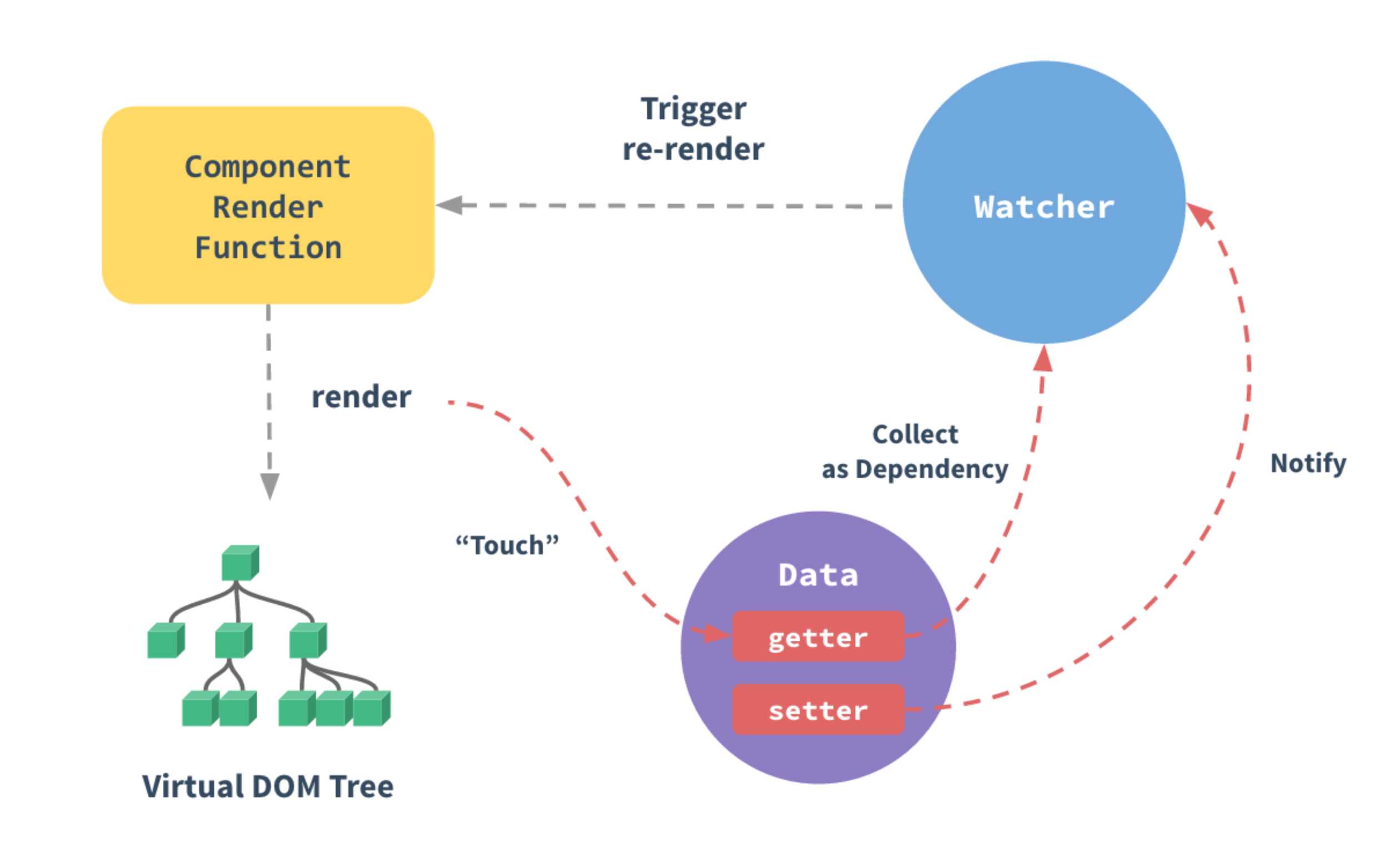
状态data vs属性props

•状态是组件自身的数据

•属性是来自父组件的数据

•状态的改变未必会触发更新 •属性的改变未必会触发更新

响应式更新



课后习题

•数组有哪些方法支持响应式更新，如不支持如何处理，底层原理如何实现的？

合理应用计算属性和侦听器

•减少模板中计算逻辑

•数据缓存

•依赖固定的数据类型（响应式数据）

侦听器watch

•更加灵活、通用

• watch中可以执行任何逻辑，如函数节流，Ajax异步获取数据，甚至操作DOM

• computed能做的，watch都能做，反之则不行

•能用computed的尽量用computed

课后习题

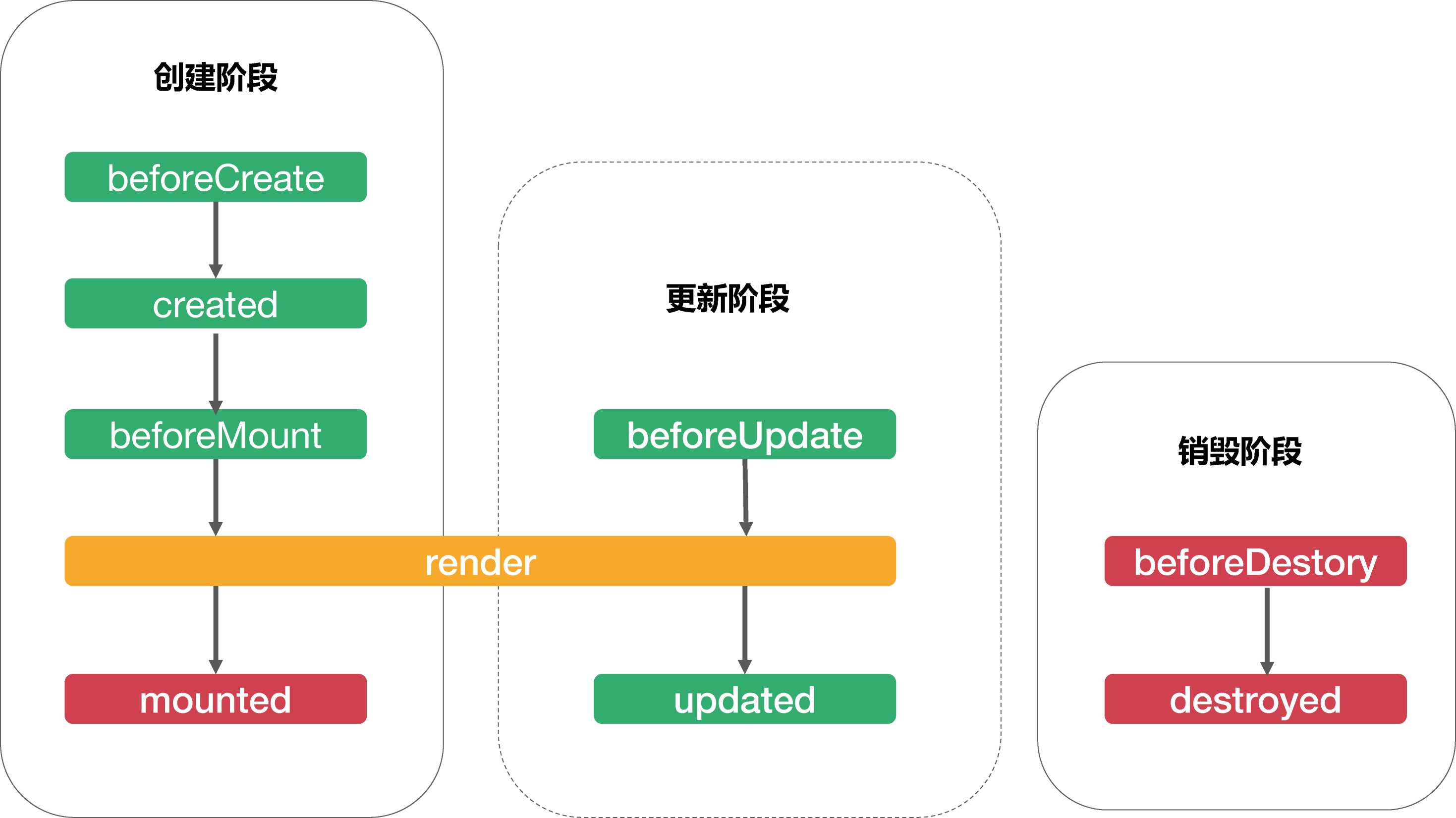
•对Watchl Demo进行防抖改造，即直到用户停止输入超过500毫秒后

才更新fullName

生命周期的应用场景和函数式组件

**. 0**极客肘间

生命周期



创建阶段

**创建阶段**

n 置等

、、 /'

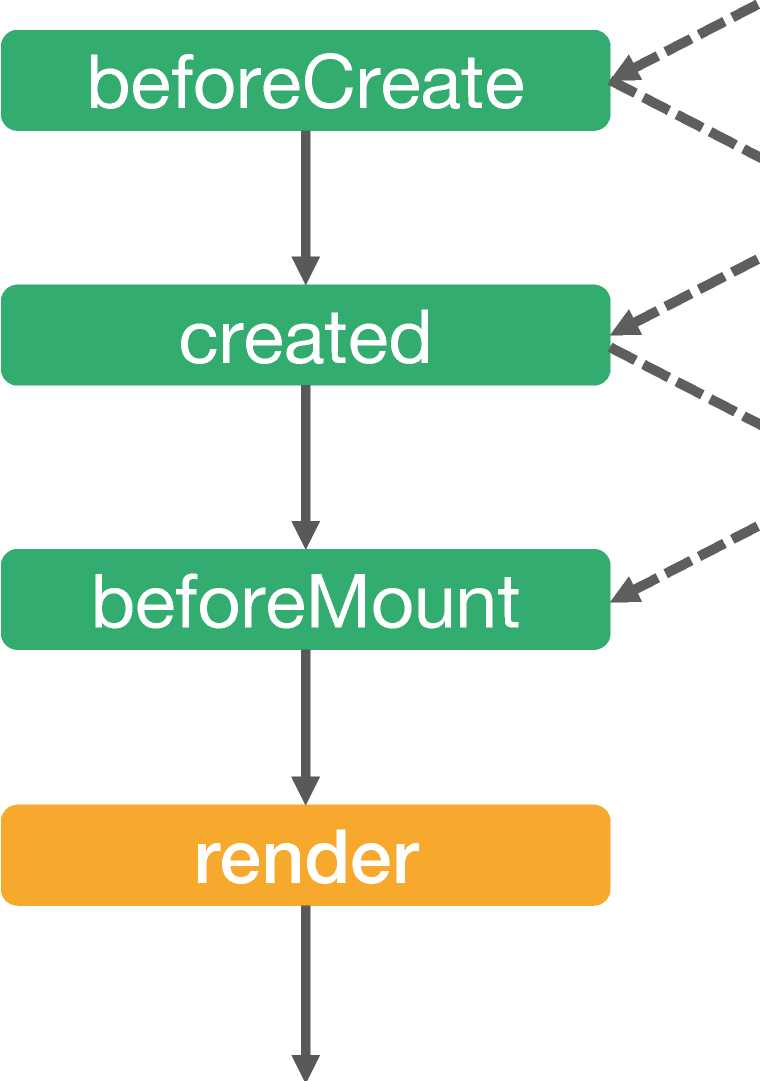
> 模板编译到render

mounted

」异步请亲、操伦DOM、定］

7、 时器等

'、 /



更新阶段

**更新阶段**

」 依赖数据改变或

[$forceUpdate 强制刷新

beforeUpdate

万万不可更改依赖数据

render

updated

」丽 boM詞口

万万不可更改依赖数据

销毁阶段

**销毁阶段**

beforeDestory

此秘昵歸廊薪酬冨祯請〕

destroyed

函数式组件

• functional: true

•无状态、无实例、没有this上下文、无生命周期

•设计一个秒杀倒计时组件

**O**极客时间

指令的本质

内置指令

v-text

v-html

v-show

v-if

v-else

v-else-if

v-for

语法糖、标志位

v-on

v-bind  
v-model

v-slot

v-pre

v-cloak

v-once

自定义指令

bind  
inserted  
update  
componentUpdated

生命周期钩子

unbind

•查看组件生命周期和指令周期钩子的运行顺序

常用高级特性provide/inject

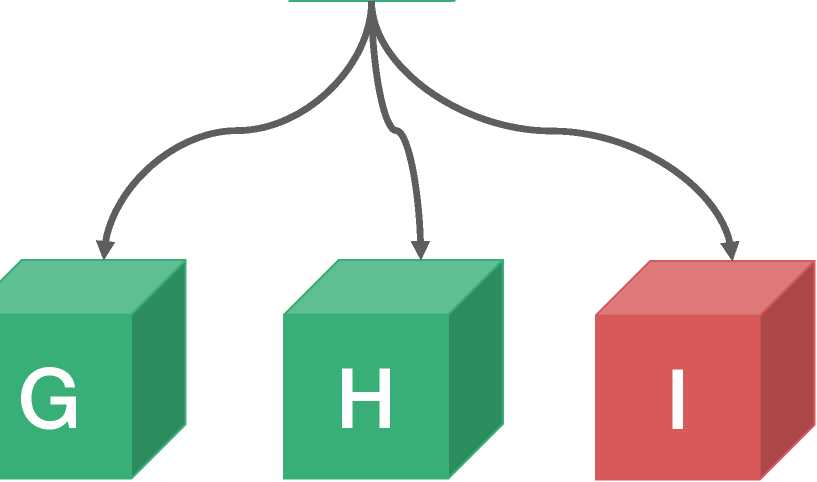
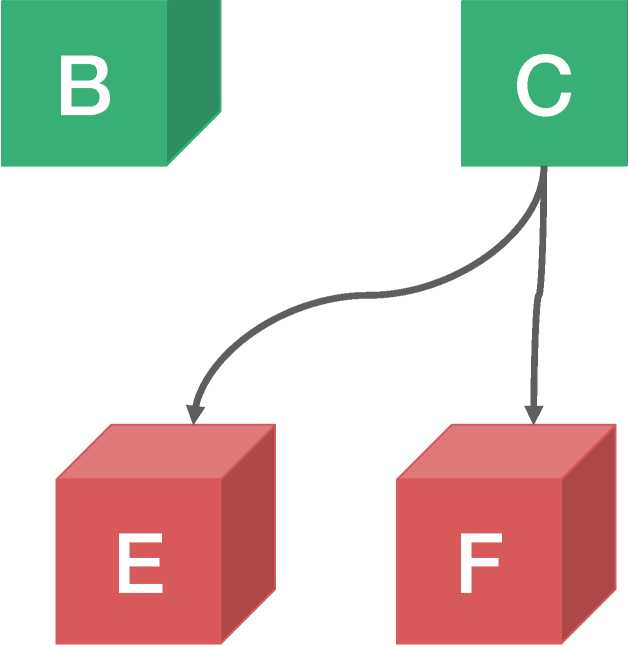
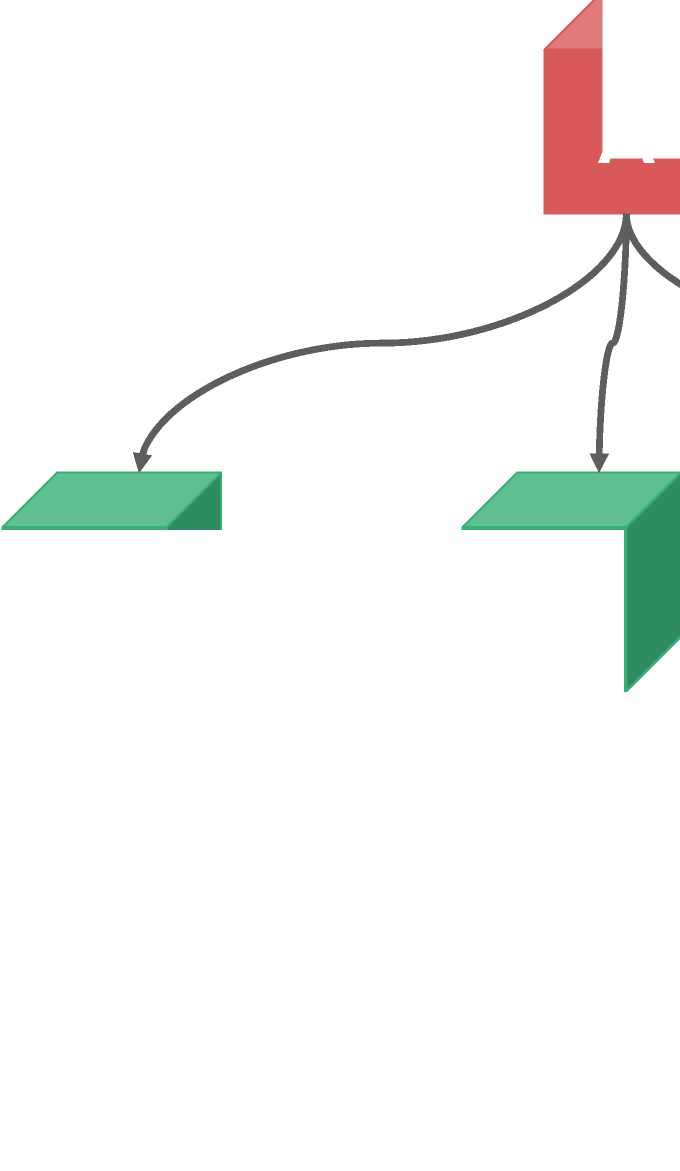
A

inject

inject

provide

inject



•使用 2.6 最新 API Vue.observable 优化响应式 provide 如何优雅的获取跨层级组件实例（拒绝递归）

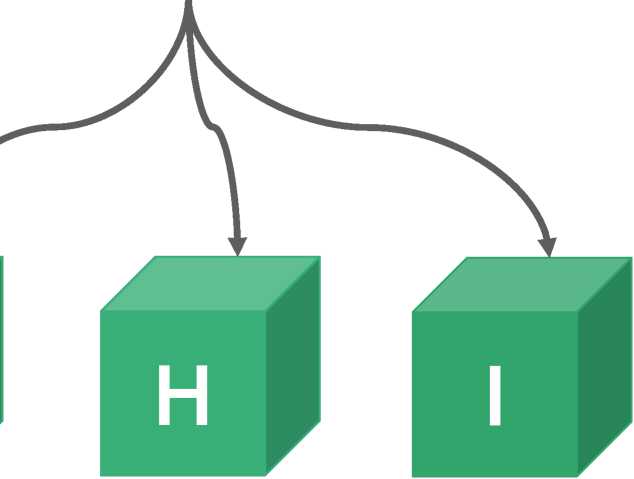
ref引用信息

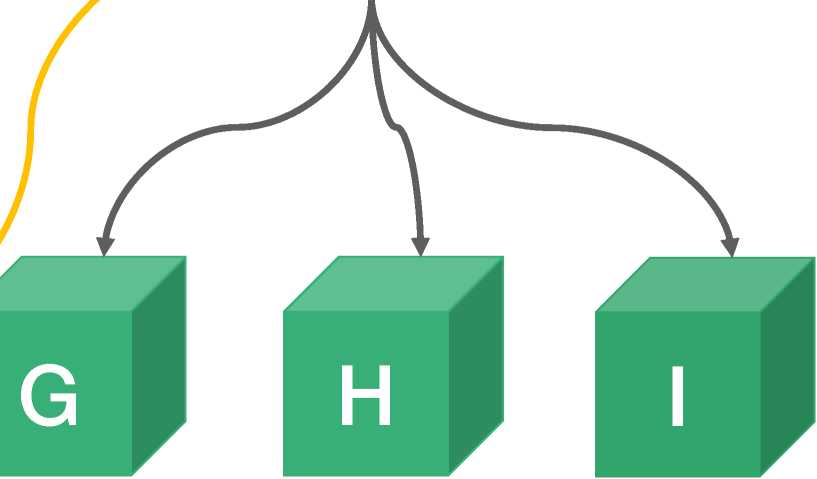
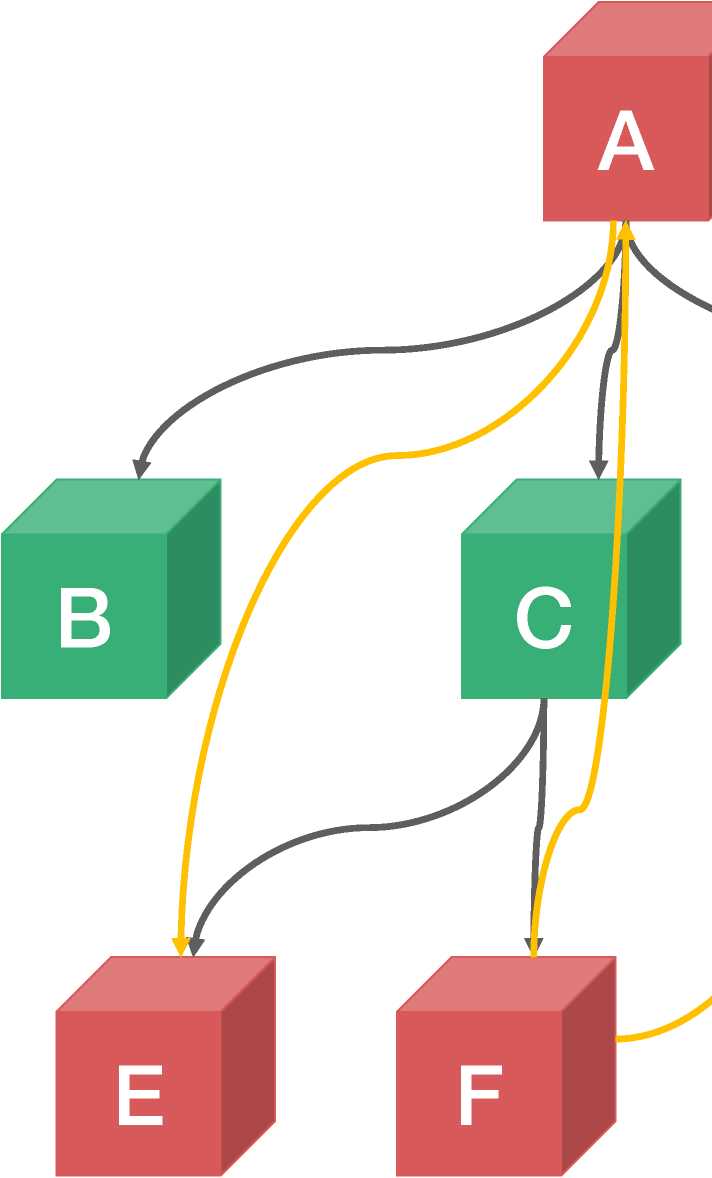
・ vm.$refs.p' will be the DOM node <p ref="pri>hello</p>

・ vm.$refs.child' will be the child component instance --> 〈child-component ref="child'></child-component>

•代码繁琐

•性能低效





callback ref

•主动通知(setXxxRef)

•主动获取(getXxxRef)

• v-ant-ref指令回调中能否对更改响应式数据？为什么？



template和JSX对比及其他们的本质

•模板语法（HTML的扩展）

•数据绑定使用Mustache语法（双大括号）

Message: {{ msg }}

jsx

• JavaScript的语法扩展

•数据绑定使用单引号

Message: {this-msg

template vs JSX

**template**

**JSX**

学习成本低

大量内置指令简化开发

灵活

组件作用域css

灵活

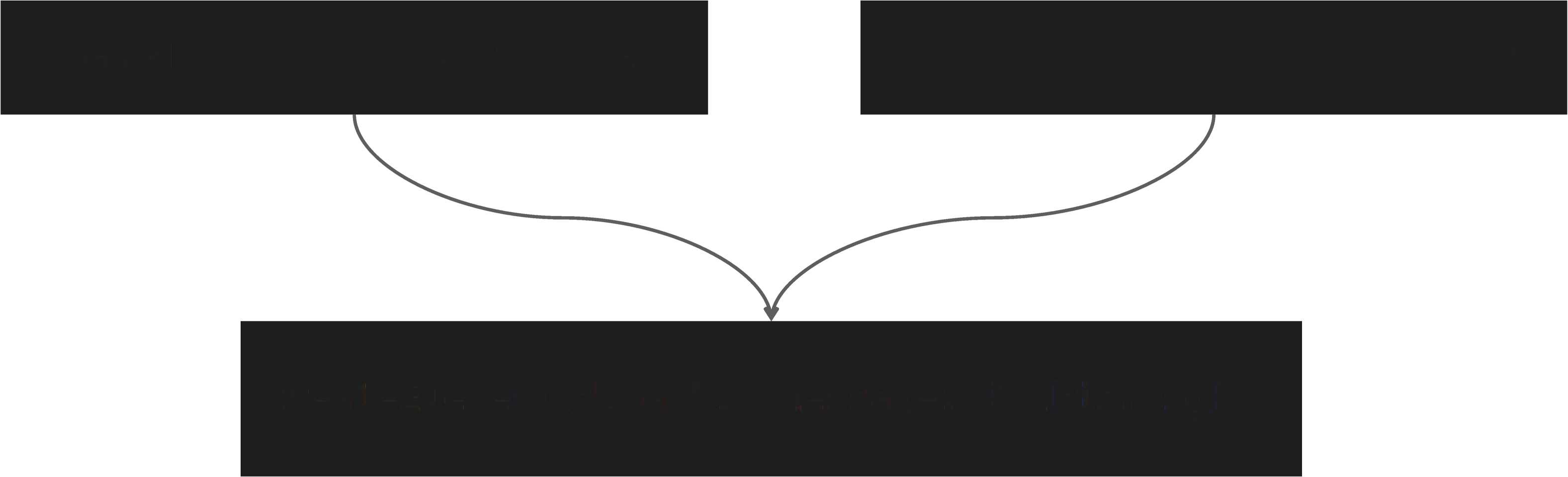
灵活

But灵活性低

“更抽象一点来看，我们可以把组件区分为两类: —类是偏视图表现的(presentational),—类则 是偏逻辑的(logical)。我们推荐在前者中使用模 板，在后者中使用JSX或渲染函数。这两类组件的 比例会根据应用类型的不同有所变化,但整体来说 我们发现表现类的组件远远多于逻辑类组件。“

Vue官方文档

But语法糖



createElement(11 span"f 'Message: ${this.msg}x)

<span>Message: {{ msg }}</span>

<span>Message: {this.msg}</span>

•将组件三大核心概念的的Demo改成JSX形式

扫码试看/订阅

《**Nginx**核心知识**100**讲》



习题解答

L子组件为何不可以修改父组件传递的Prop，如果修改了，Vue是如何 监控到属性的修改并给出警告的？

* Object.definePropertyO
* [https://developer.mozilla.org/en- US/docs/Web/JavaScript/Reference/Global\_Objects/Object/defineProperty](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/defineProperty)

2.this.$emit的返回值是什么？

• this

•如果需要返回值可以使用回调参数

3.相同名称的插槽是合并还是替换？

* Vue 2.5版本，普通插槽合并、作用域插槽替换
* Vue 2.6版本，都是替换

4.扩展Personalinfo Demo对手机号做非空且合法校验，如不合法 则给出错误提示

•看 DEMOL2\_a



5.为什么不能用index作为key ?

•更新DOM性能问题

•会引入状态Bug问题，看Demo 1.3\_a

6.数组有哪些方法支持响应式更新，如不支持如何处理，底层原理 如何实现的？

•支持：push。、pop()、shift。、unshift()、splice。、sort。、reverse。

•不支持：filter。、concat()、slice()

•原理同样是使用Object.defineProperty对数组方法进行改写

7.对Watchl Demo进行防抖改造，即直到用户停止输入超过500 毫秒后才更新fullName

* setTimeout
* lodash debounce

• Demo 1.5/Watchl\_pro

8.设计一个秒杀倒计时组件

• Demo 1.6 a

9.查看组件生命周期和指令周期钩子的运行顺序

• Demo 1.7 a

10.使用2.6最新 API Vue.observable 优化响应式 provide

• Demo 1.8 ChildrenA a

11. v-ant-ref指令回调中能否对更改响应式数据？为什么？

•不能，会死循环

12.将三大核心概念的Demo改成JSX形式

• Demo 1.10 a

扫码试看/订阅

《**Nginx**核心知识**100**讲》

