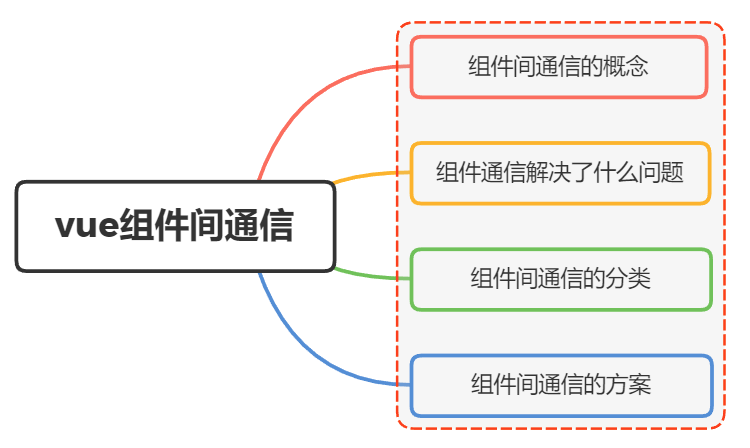
# 面试官：Vue组件之间的通信方式都有哪些？



## 一、组件间通信的概念

开始之前，我们把**组件间通信**这个词进行拆分

* 组件
* 通信

都知道组件是vue最强大的功能之一，vue中每一个.vue我们都可以视之为一个组件通信指的是发送者通过某种媒体以某种格式来传递信息到收信者以达到某个目的。广义上，任何信息的交通都是通信**组件间通信**即指组件(.vue)通过某种方式来传递信息以达到某个目的举个栗子我们在使用UI框架中的table组件，可能会往table组件中传入某些数据，这个本质就形成了组件之间的通信

## 二、组件间通信解决了什么

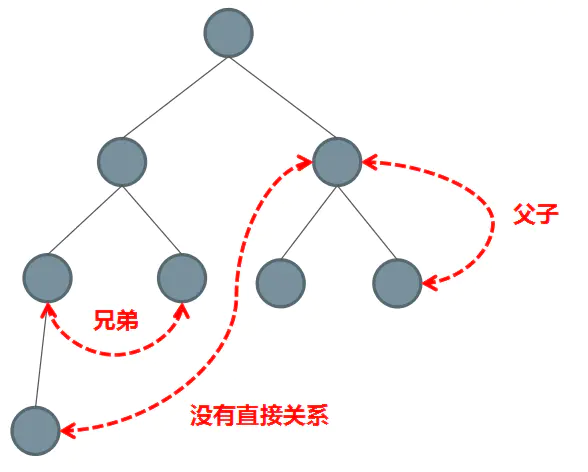
在古代，人们通过驿站、飞鸽传书、烽火报警、符号、语言、眼神、触碰等方式进行信息传递，到了今天，随着科技水平的飞速发展，通信基本完全利用有线或无线电完成，相继出现了有线电话、固定电话、无线电话、手机、互联网甚至视频电话等各种通信方式从上面这段话，我们可以看到通信的本质是信息同步，共享回到vue中，每个组件之间的都有独自的作用域，组件间的数据是无法共享的但实际开发工作中我们常常需要让组件之间共享数据，这也是组件通信的目的要让它们互相之间能进行通讯，这样才能构成一个有机的完整系统

## 二、组件间通信的分类

组件间通信的分类可以分成以下

* 父子组件之间的通信
* 兄弟组件之间的通信
* 祖孙与后代组件之间的通信
* 非关系组件间之间的通信

关系图:

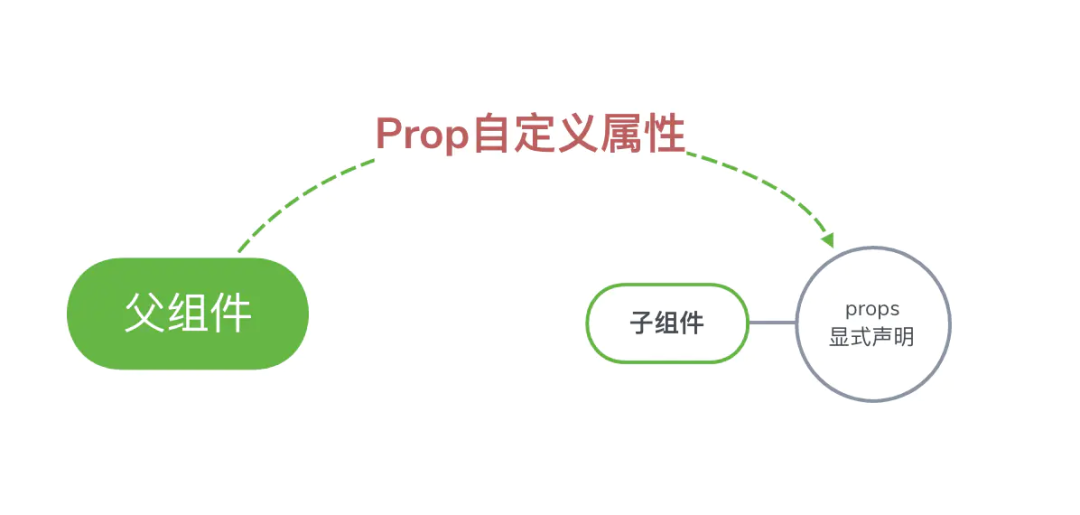


## 三、组件间通信的方案

整理vue中8种常规的通信方案

1. 通过 props 传递
2. 通过 $emit 触发自定义事件
3. 使用 ref
4. EventBus
5. root
6. attrs 与 listeners
7. Provide 与 Inject
8. Vuex

### props传递数据



* 适用场景：父组件传递数据给子组件
* 子组件设置props属性，定义接收父组件传递过来的参数
* 父组件在使用子组件标签中通过字面量来传递值

Children.vue

props:{   
    // 字符串形式   
 name:String // 接收的类型参数   
    // 对象形式   
    age:{     
        type:Number, // 接收的类型为数值   
        defaule:18,  // 默认值为18   
       require:true // age属性必须传递   
    }   
}

Father.vue组件

<Children name="jack" age=18 />

### $emit 触发自定义事件

* 适用场景：子组件传递数据给父组件
* 子组件通过$emit触发自定义事件，$emit第二个参数为传递的数值
* 父组件绑定监听器获取到子组件传递过来的参数

Chilfen.vue

this.$emit('add', good)

Father.vue

<Children @add="cartAdd($event)" />

### ref

* 父组件在使用子组件的时候设置ref
* 父组件通过设置子组件ref来获取数据

父组件

<Children ref="foo" />   
   
this.$refs.foo  // 获取子组件实例，通过子组件实例我们就能拿到对应的数据

### EventBus

* 使用场景：兄弟组件传值
* 创建一个中央事件总线EventBus
* 兄弟组件通过$emit触发自定义事件，$emit第二个参数为传递的数值
* 另一个兄弟组件通过$on监听自定义事件

Bus.js

// 创建一个中央时间总线类   
class Bus {   
  constructor() {   
    this.callbacks = {};   // 存放事件的名字   
  }   
  $on(name, fn) {   
    this.callbacks[name] = this.callbacks[name] || [];   
    this.callbacks[name].push(fn);   
  }   
  $emit(name, args) {   
    if (this.callbacks[name]) {   
      this.callbacks[name].forEach((cb) => cb(args));   
    }   
  }   
}   
   
// main.js   
Vue.prototype.$bus = new Bus() // 将$bus挂载到vue实例的原型上   
// 另一种方式   
Vue.prototype.$bus = new Vue() // Vue已经实现了Bus的功能

Children1.vue

this.$bus.$emit('foo')

Children2.vue

this.$bus.$on('foo', this.handle)

### root

* 通过共同祖辈$parent或者$root搭建通信桥连

兄弟组件

this.$parent.on('add',this.add)

另一个兄弟组件

this.$parent.emit('add')

### listeners

* 适用场景：祖先传递数据给子孙
* 设置批量向下传属性$attrs和 $listeners
* 包含了父级作用域中不作为 prop 被识别 (且获取) 的特性绑定 ( class 和 style 除外)。
* 可以通过 v-bind="$attrs" 传⼊内部组件

// child：并未在props中声明foo   
<p>{{$attrs.foo}}</p>   
   
// parent   
<HelloWorld foo="foo"/>

// 给Grandson隔代传值，communication/index.vue   
<Child2 msg="lalala" @some-event="onSomeEvent"></Child2>   
   
// Child2做展开   
<Grandson v-bind="$attrs" v-on="$listeners"></Grandson>   
   
// Grandson使⽤   
<div @click="$emit('some-event', 'msg from grandson')">   
{{msg}}   
</div>

### provide 与 inject

* 在祖先组件定义provide属性，返回传递的值
* 在后代组件通过inject接收组件传递过来的值

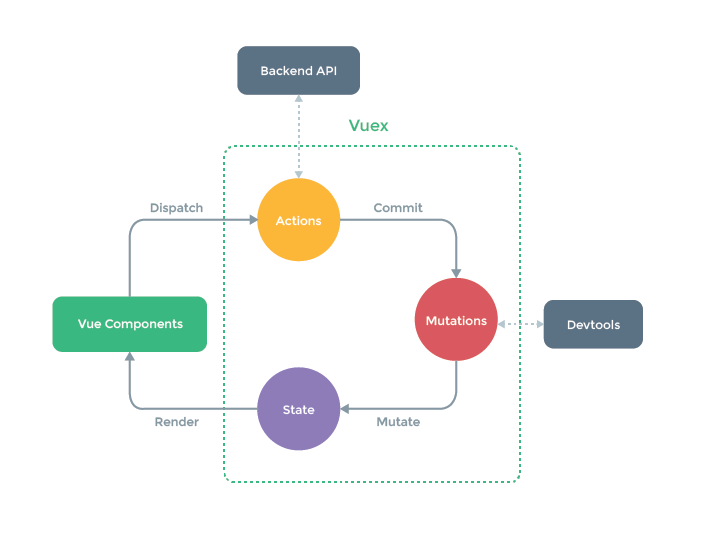
祖先组件

provide(){   
    return {   
        foo:'foo'   
    }   
}

后代组件

inject:['foo'] // 获取到祖先组件传递过来的值

### vuex

* 适用场景: 复杂关系的组件数据传递
* Vuex作用相当于一个用来存储共享变量的容器 
* state用来存放共享变量的地方
* getter，可以增加一个getter派生状态，(相当于store中的计算属性），用来获得共享变量的值
* mutations用来存放修改state的方法。
* actions也是用来存放修改state的方法，不过action是在mutations的基础上进行。常用来做一些异步操作

### 小结

* 父子关系的组件数据传递选择 props  与 $emit进行传递，也可选择ref
* 兄弟关系的组件数据传递可选择$bus，其次可以选择$parent进行传递
* 祖先与后代组件数据传递可选择attrs与listeners或者 Provide与 Inject
* 复杂关系的组件数据传递可以通过vuex存放共享的变量

## 参考文献

* https://juejin.cn/post/6844903990052782094#heading-0
* https://zh.wikipedia.org/wiki/%E9%80%9A%E4%BF%A1
* https://vue3js.cn/docs/zh

面试官VUE系列总进度：5／33

[面试官：说说你对vue的理解?](http://mp.weixin.qq.com/s?__biz=MzU1OTgxNDQ1Nw==&mid=2247484101&idx=1&sn=83b0983f0fca7d7c556e4cb0bff8c9b8&chksm=fc10c093cb674985ef3bd2966f66fc28c5eb70b0037e4be1af4bf54fb6fa9571985abd31d52f&scene=21#wechat_redirect)

[面试官：说说你对SPA（单页应用）的理解?](http://mp.weixin.qq.com/s?__biz=MzU1OTgxNDQ1Nw==&mid=2247484119&idx=1&sn=d171b28a00d42549d279498944a98519&chksm=fc10c081cb6749976814aaeda6a6433db418223cec57edda7e15b9e5a0ca69ad549655639c61&scene=21#wechat_redirect)

[面试官：说说你对双向绑定的理解?](http://mp.weixin.qq.com/s?__biz=MzU1OTgxNDQ1Nw==&mid=2247484167&idx=1&sn=7b00b4333ab2722f25f12586b70667ca&chksm=fc10c151cb6748476008dab2f4e6c6264f5d19678305955c85cec1b619e56e8f7457b7357fb9&scene=21#wechat_redirect)

[面试官：说说你对Vue生命周期的理解?](http://mp.weixin.qq.com/s?__biz=MzU1OTgxNDQ1Nw==&mid=2247484176&idx=1&sn=5623421ed2678046ed9e438aadf6e26f&chksm=fc10c146cb67485015f24f7e9f5862c4c685fc33485fe30e1b375a534b4031978439c554e0c0&scene=21#wechat_redirect)

