

# Megalo

## — 考拉的小程序解决方案



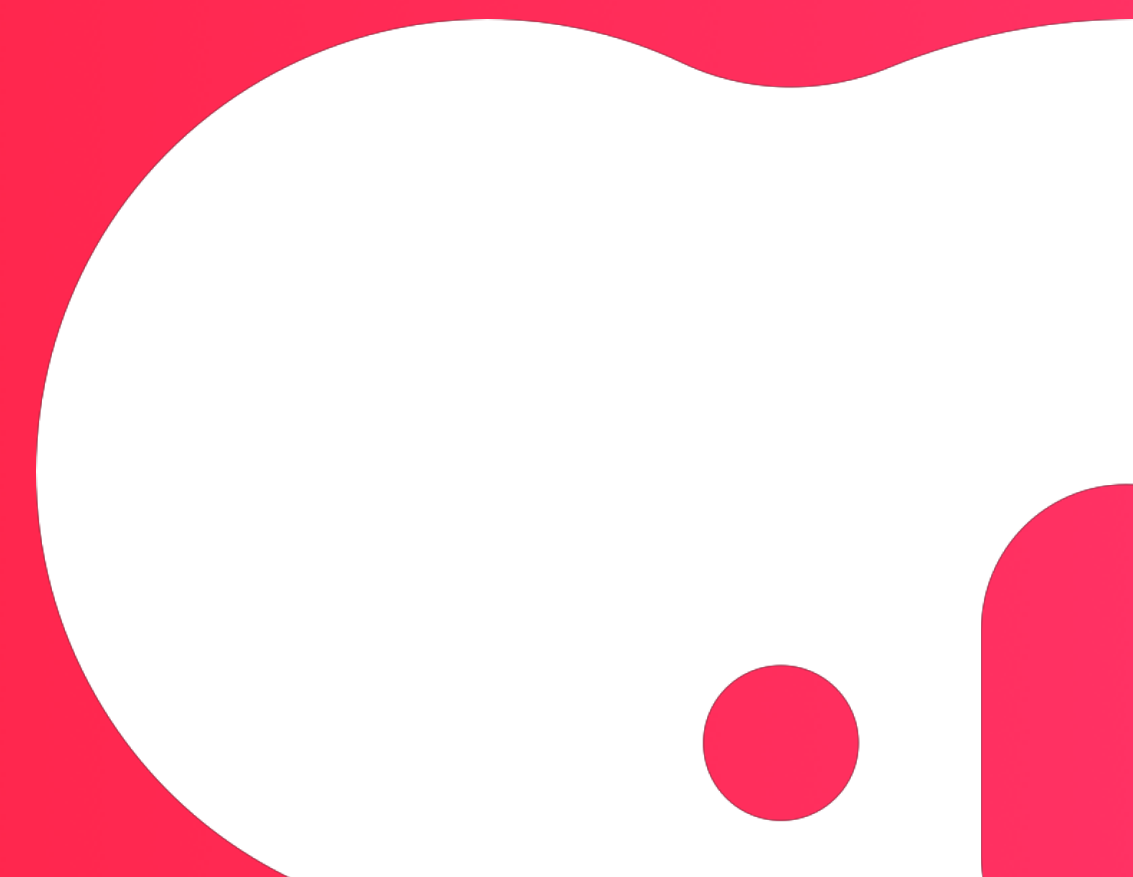
吴子然

github: [elcarim5efil](#)

网易考拉 技术部 前端组

Megalo、Mpregular 核心开发之一

# 小程序开发



# 原生小程序 开发体验差



# WePY

## 增强型框架



# MPVue

## Vue、跨端

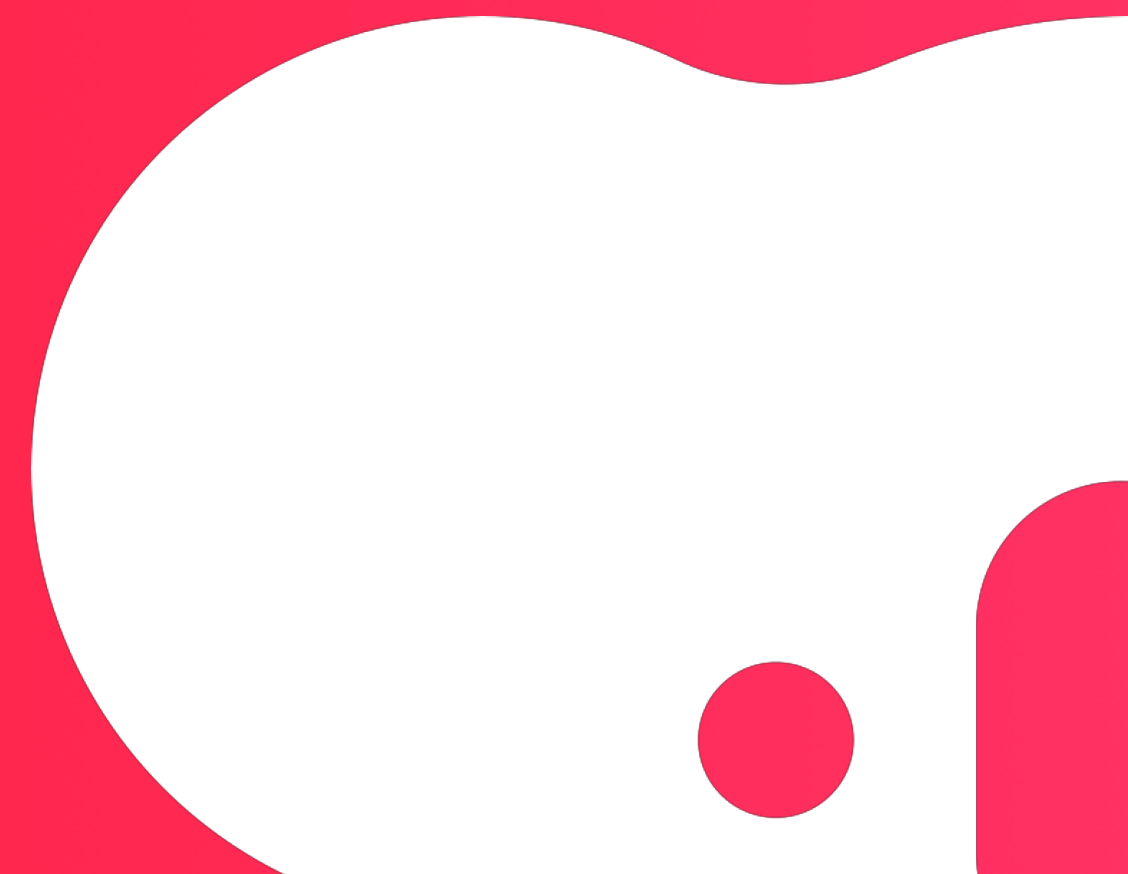


# Taro

## React、跨端



可是  
我们用不了...





**RegularJS**  
社区生态较弱



**MPRegular**  
更多特性，更少差异

**MPREGULAR**

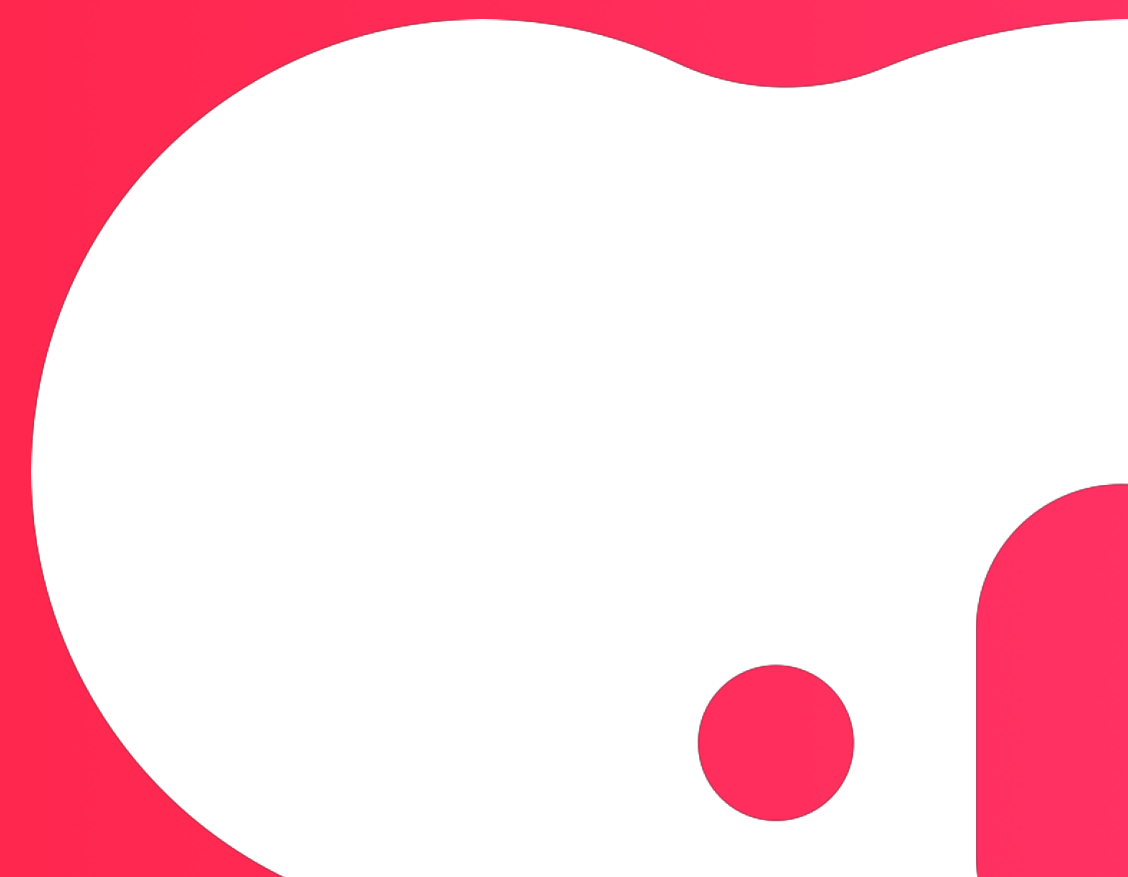


旧（原生）

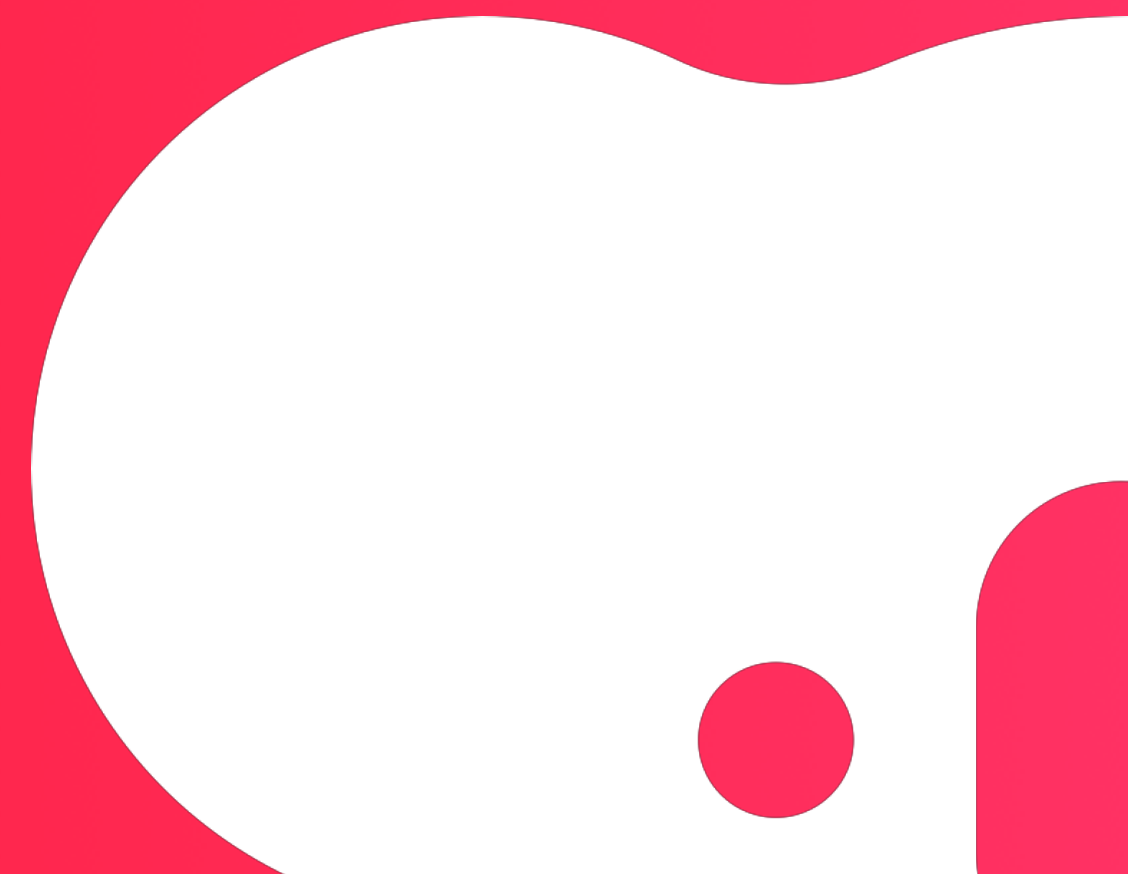


新（mpregular）

然而  
技术栈在更新



技术栈在更新  
Vue、WEEX...



# Megalo

基于 Vue

支持更多特性

让开发更简单



**@megalo/target**      **webpack 构建插件**

**@megalo/cli**          **脚手架工具**

**@megalo/ui**          **组件库**

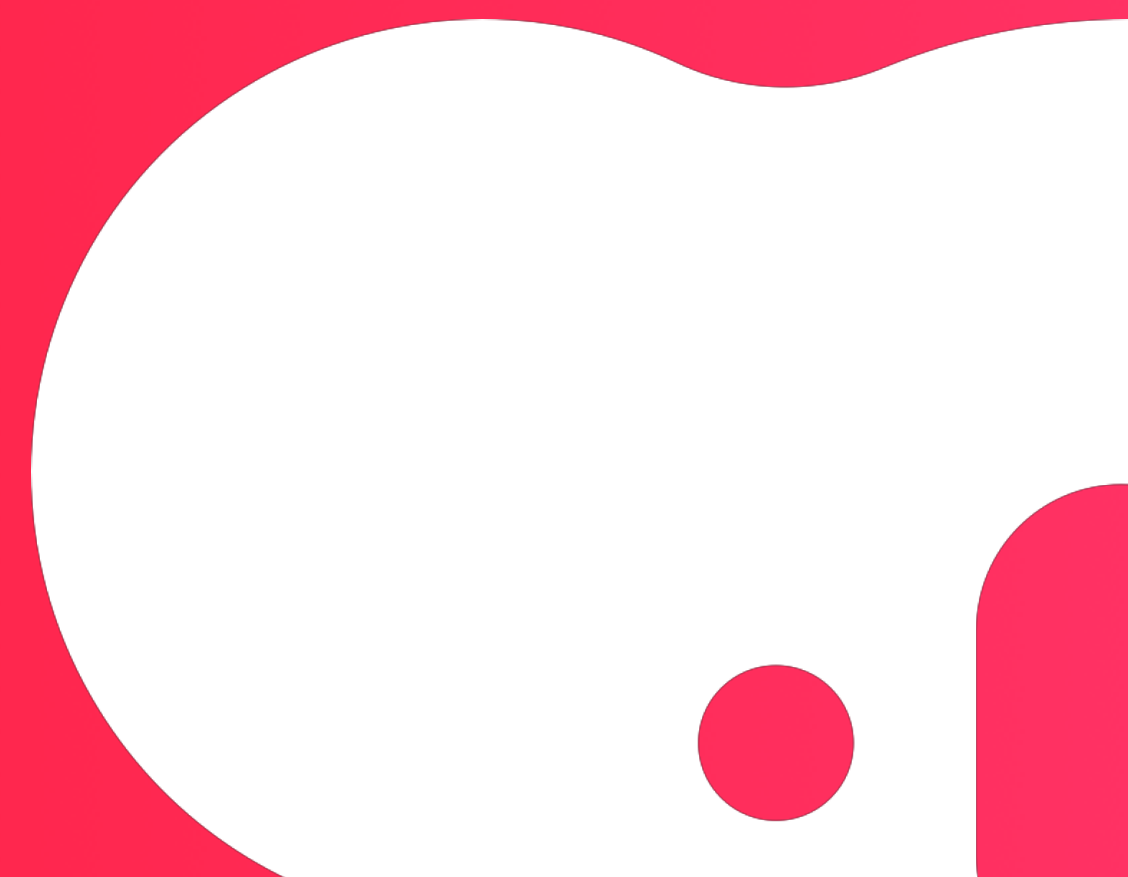
**@megalo/api**        **api 封装**

**@megalo/entry**      **页面入口插件**

...



怎么做？





index.js

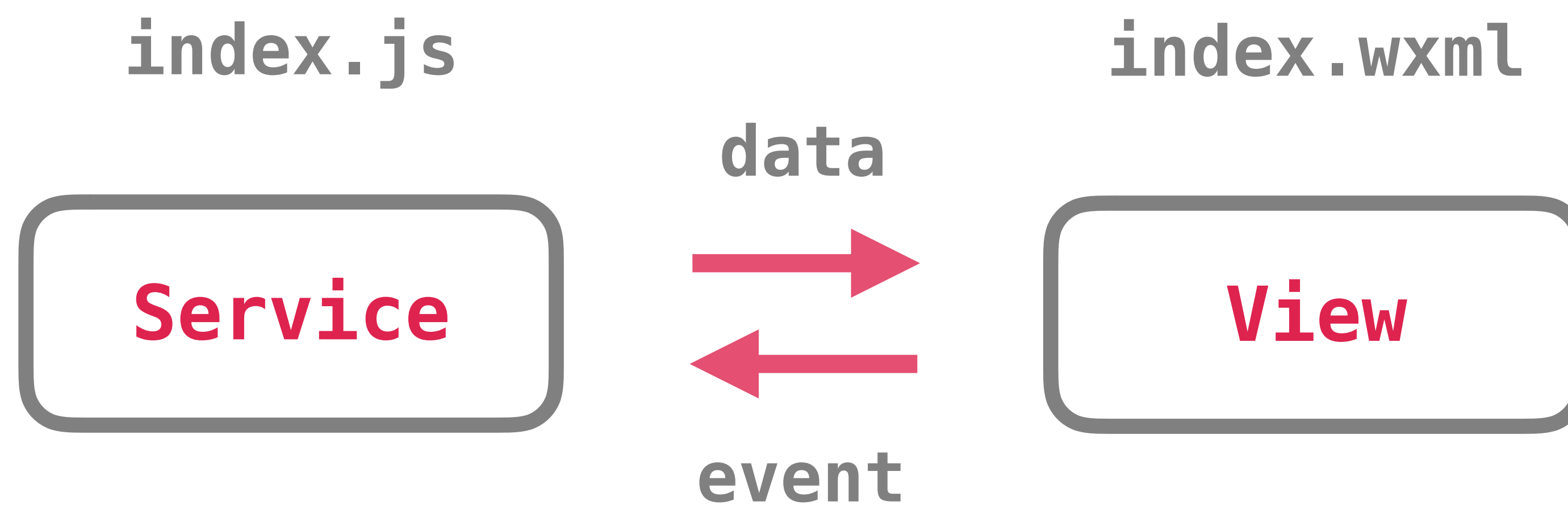
```
Page({  
  data: {  
    title: 'kaola'  
  }  
})
```

## index.js

```
Page({  
  data: {  
    title: 'kaola'  
  }  
})
```

## index.wxml

```
<view>{{ title }}</view>
```

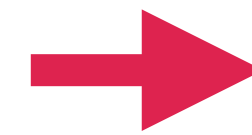


index.vue

```
<template>
  <div>{{ title }}</div>
</template>
<script>
export default {
  data() {
    return {
      title: 'kaola'
    }
  }
}
</script>
```

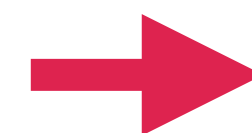
index.vue

```
<template>
  <div>{{ title }}</div>
</template>
<script>
export default {
  data() {
    return {
      title: 'kaola'
    }
  }
}
</script>
```



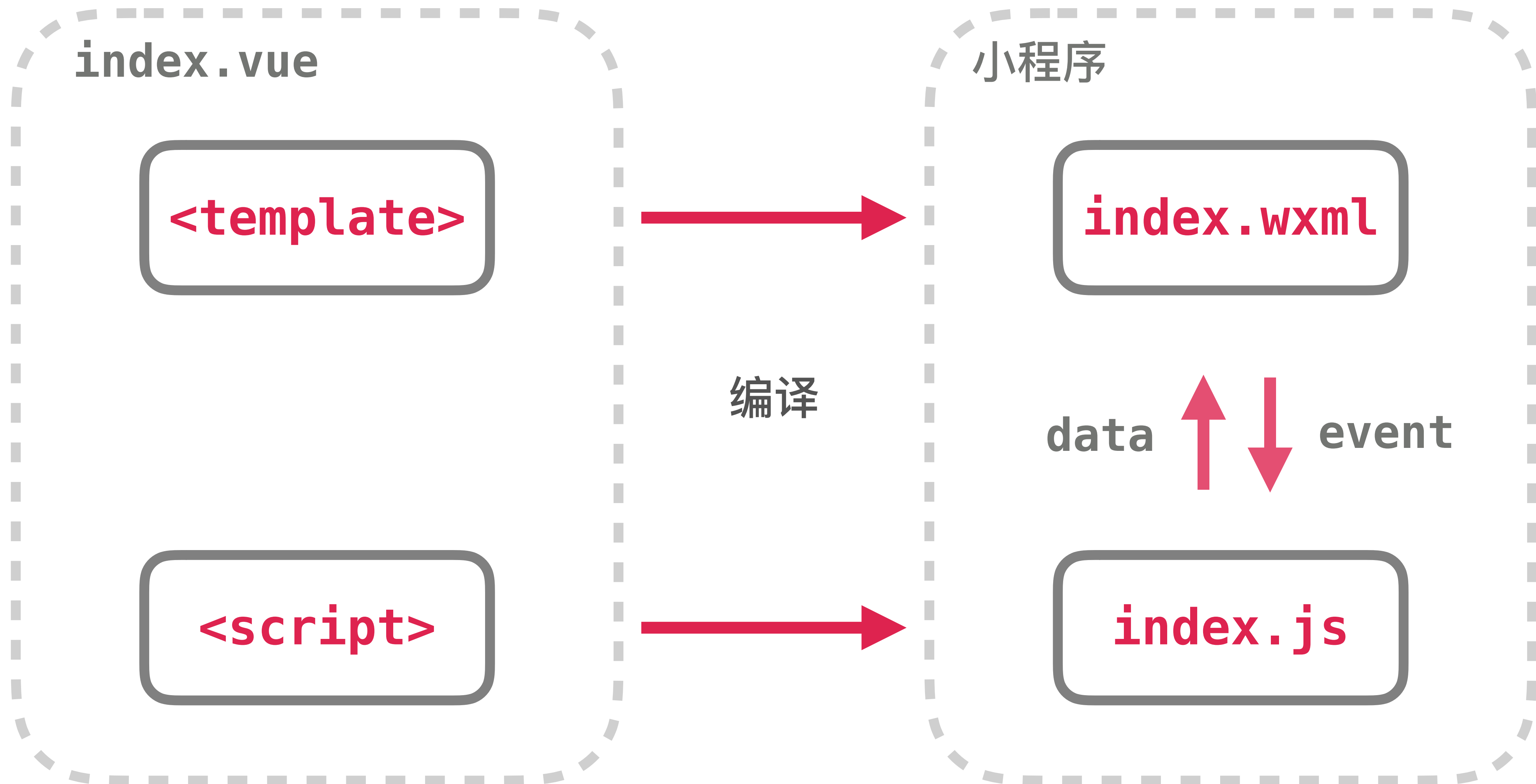
index.wxml

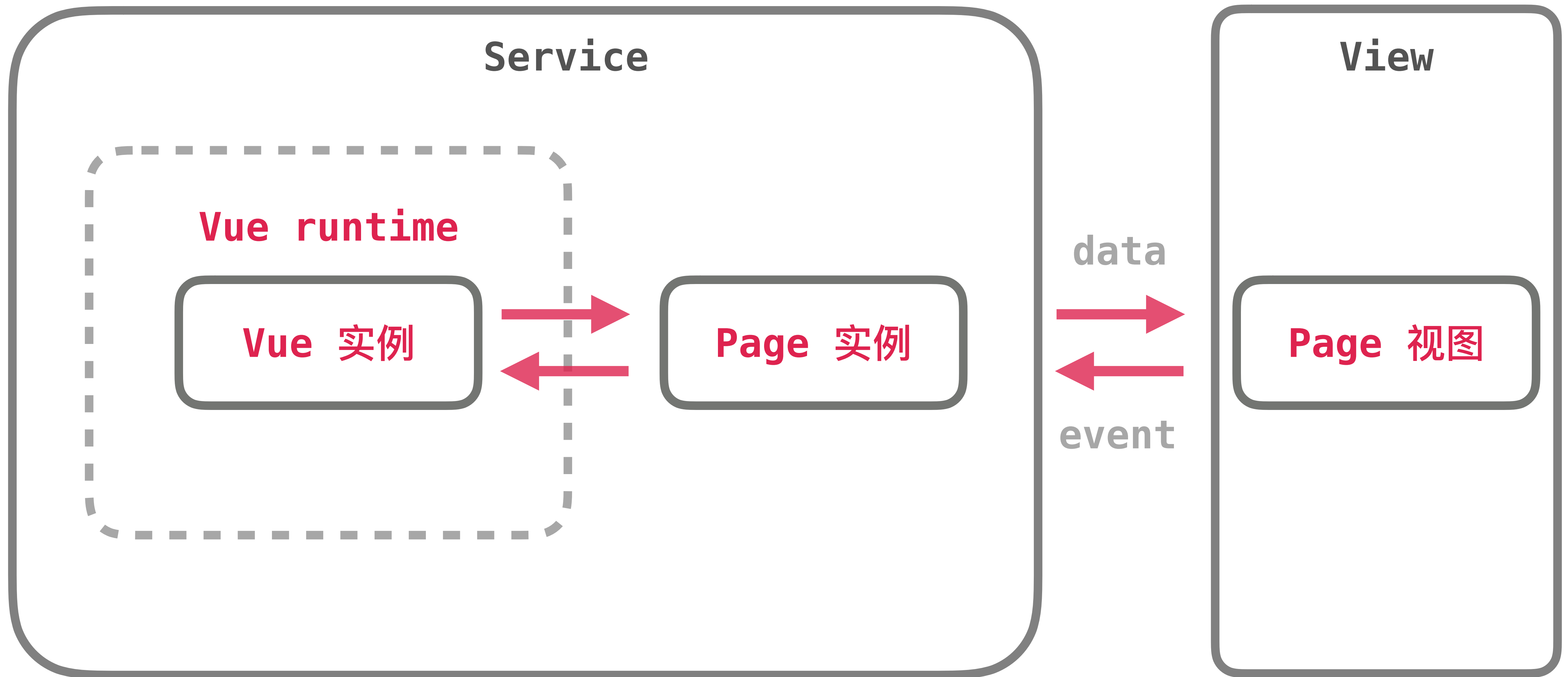
```
<view>{{ title }}</view>
```



index.js

```
Page({
  data: {
    title: 'kaola'
  }
})
```





## Loader

构建流程

## Compiler

做标记  
生成文件

## Runtime

绑定生命周期  
事件代理  
数据映射



<div>

`<div>`



`<view>`

```
<div v-if="a">
```

`<div v-if="a">`



`<view wx:if="a">`

```
<div @click="b">
```

`<div @click="b">`



`<view bindtap="b">`

```
<div>{{ title }}</div>
```

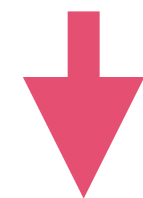
`<div>{{ title }}</div>`



`<view>{{ title }}</view>`

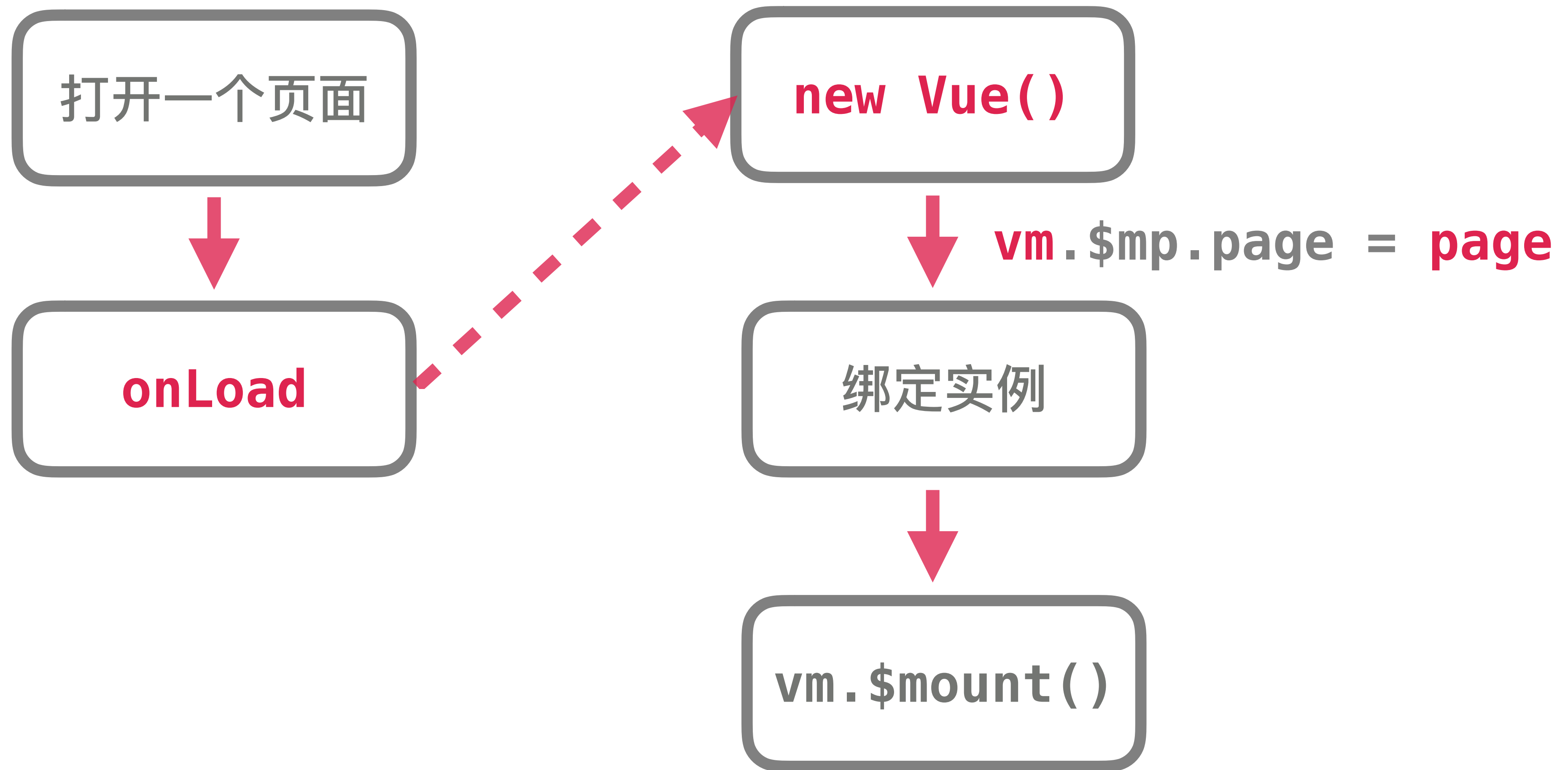


打开一个页面



onLoad

页面加载，创建 **page** 实例



```
<div @click="onClick">
```

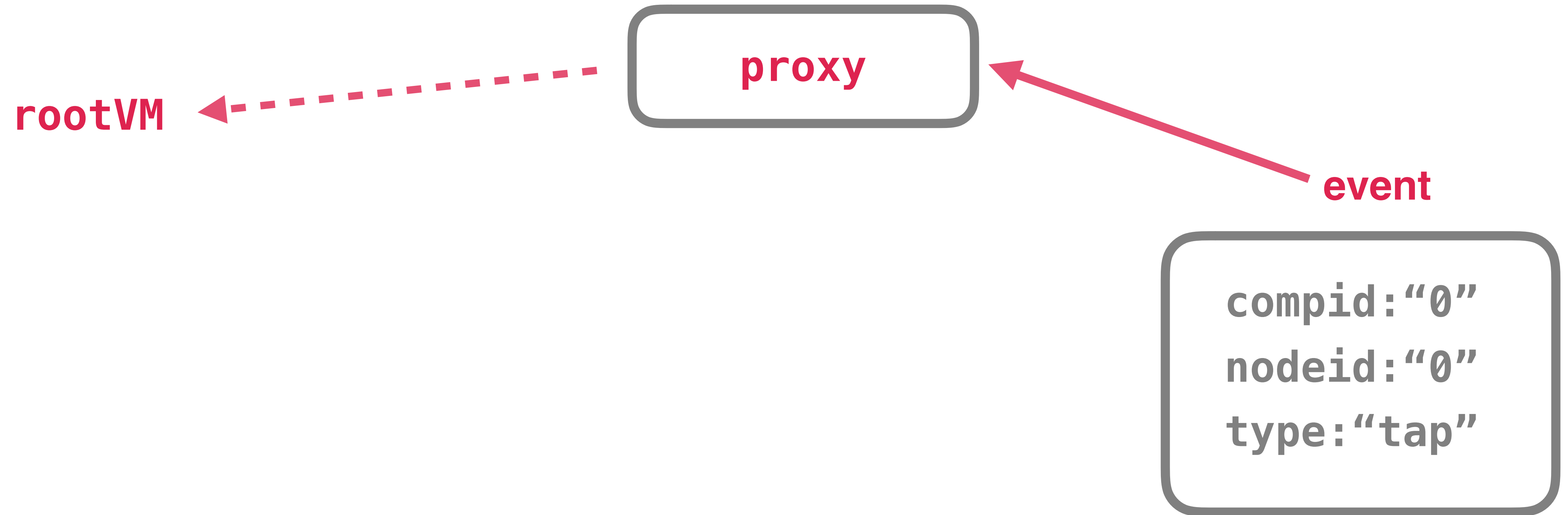


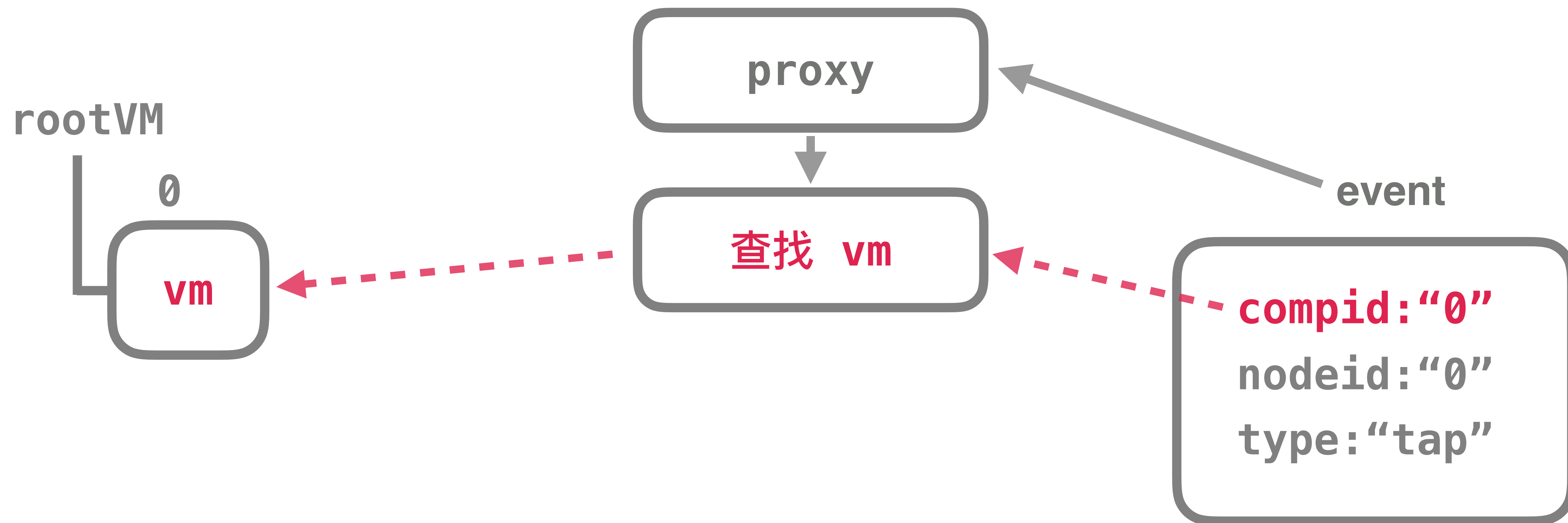
```
<view bindtap="onClick">
```

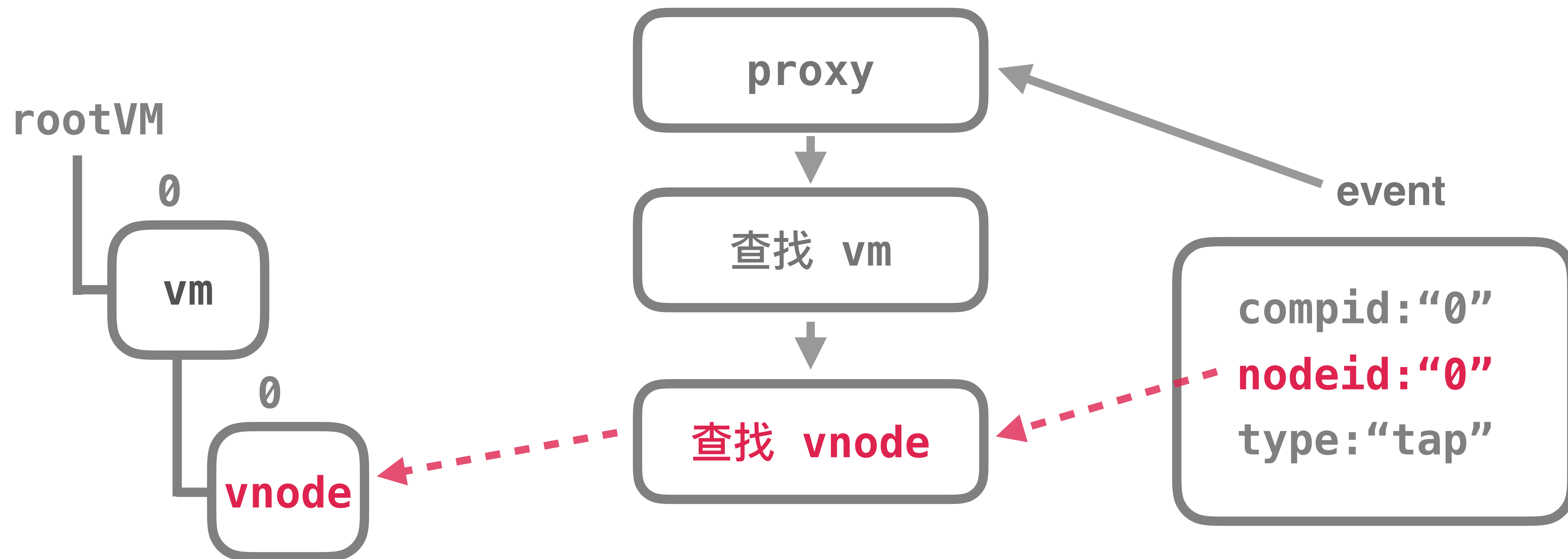
```
<div @click="onClick">
```

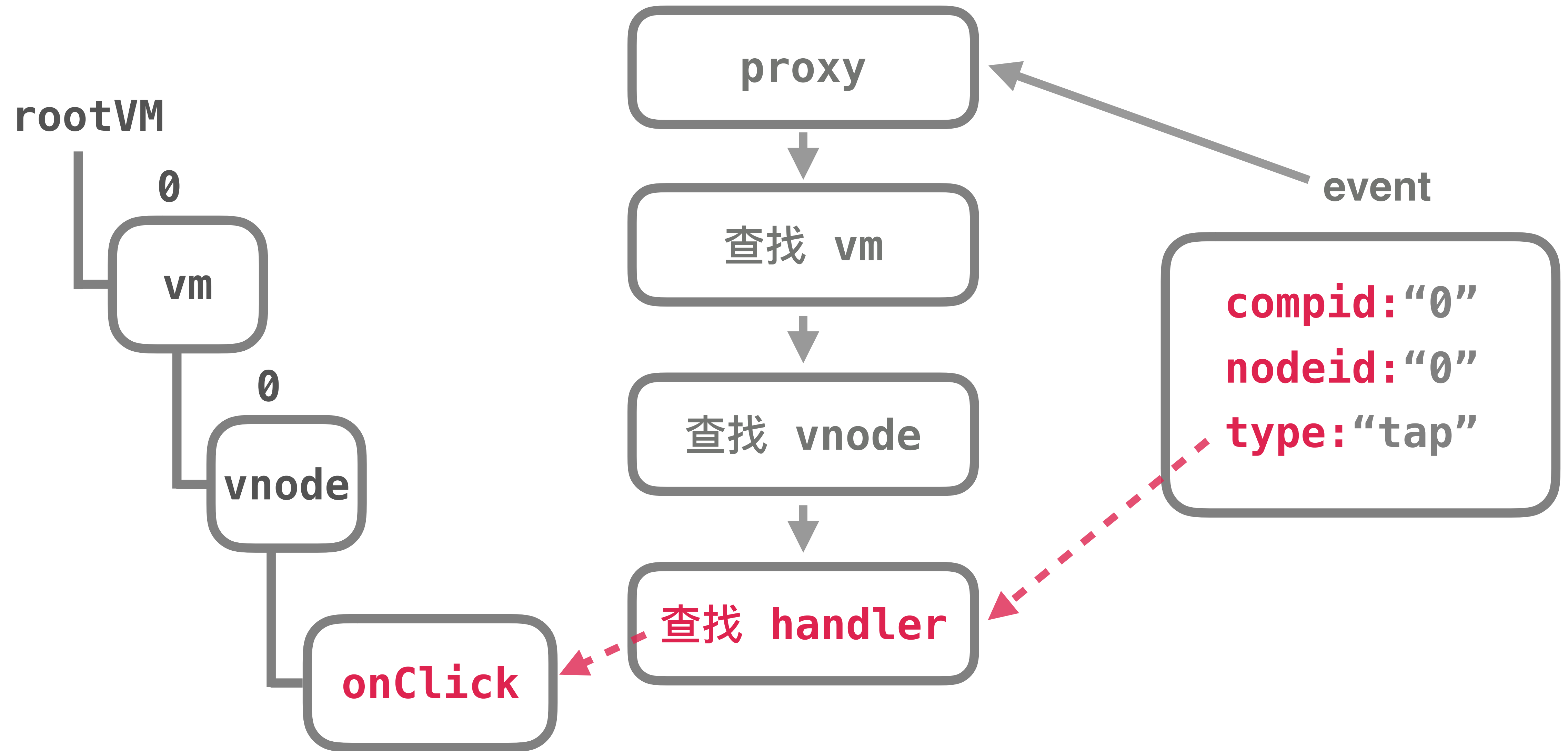


```
<view  
  bindtap="proxy"  
  data-compid="0"  
  data-nodeid="0"  
>
```

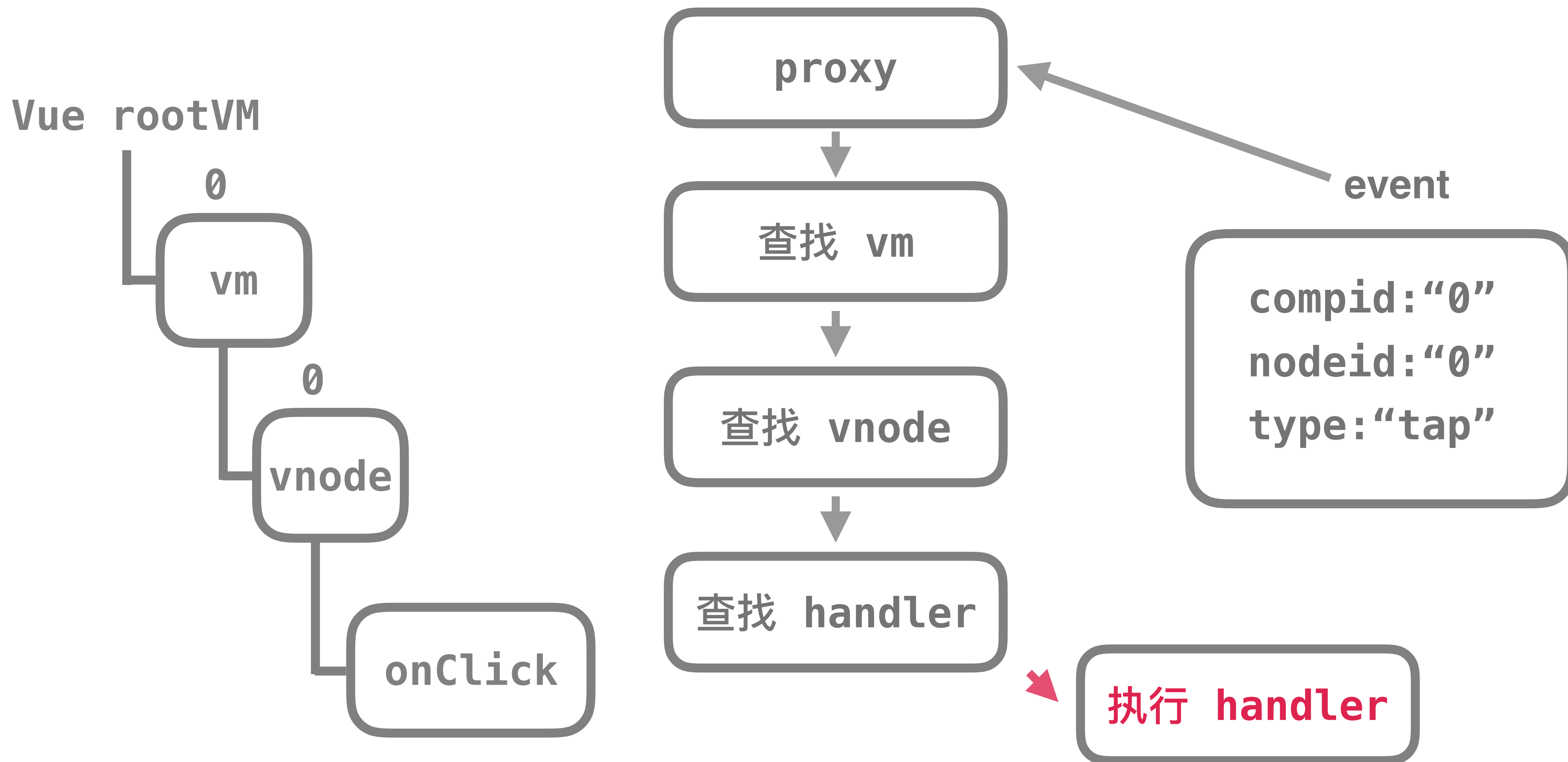












index.js

```
Page({  
  data: {  
    title: "kaola"  
  }  
})
```

data



index.wxml

```
<view>  
  {{ title }}  
</view>
```

index.js

```
setData({title: "2019"})
```



```
data.title === "2019"
```

data



index.wxml

```
<view>
```

```
{{ title }}
```

```
</view>
```

page.data

title: "kaola"

data



index.wxml

<view>

{{ title }}

</view>

script

```
export default {  
  data() {  
    return {  
      title: "kaola"  
    }  
  }  
}
```

template

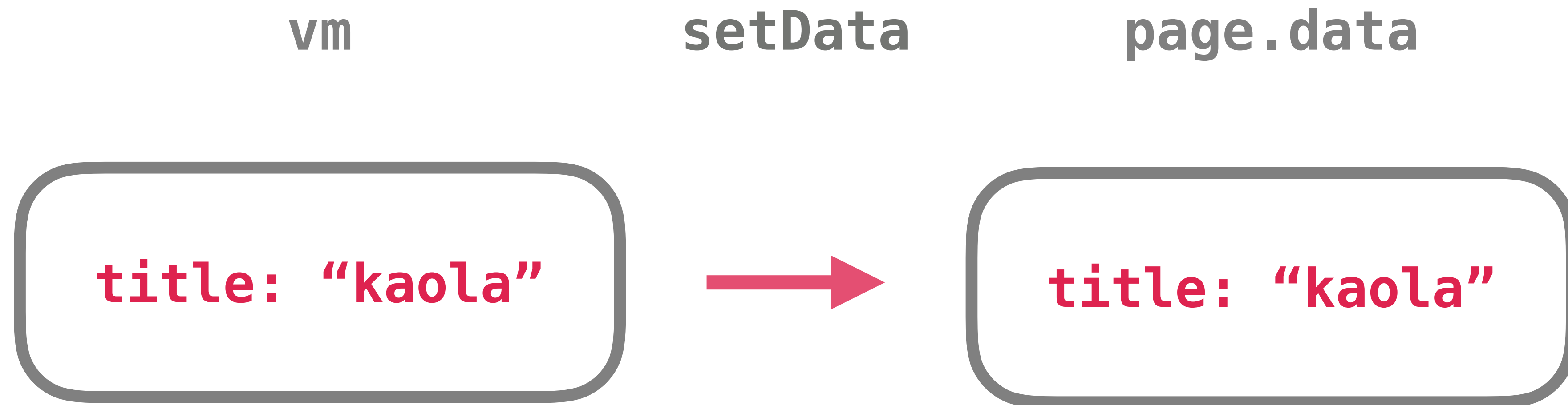
```
<div>  
  {{ title }}  
</div>
```

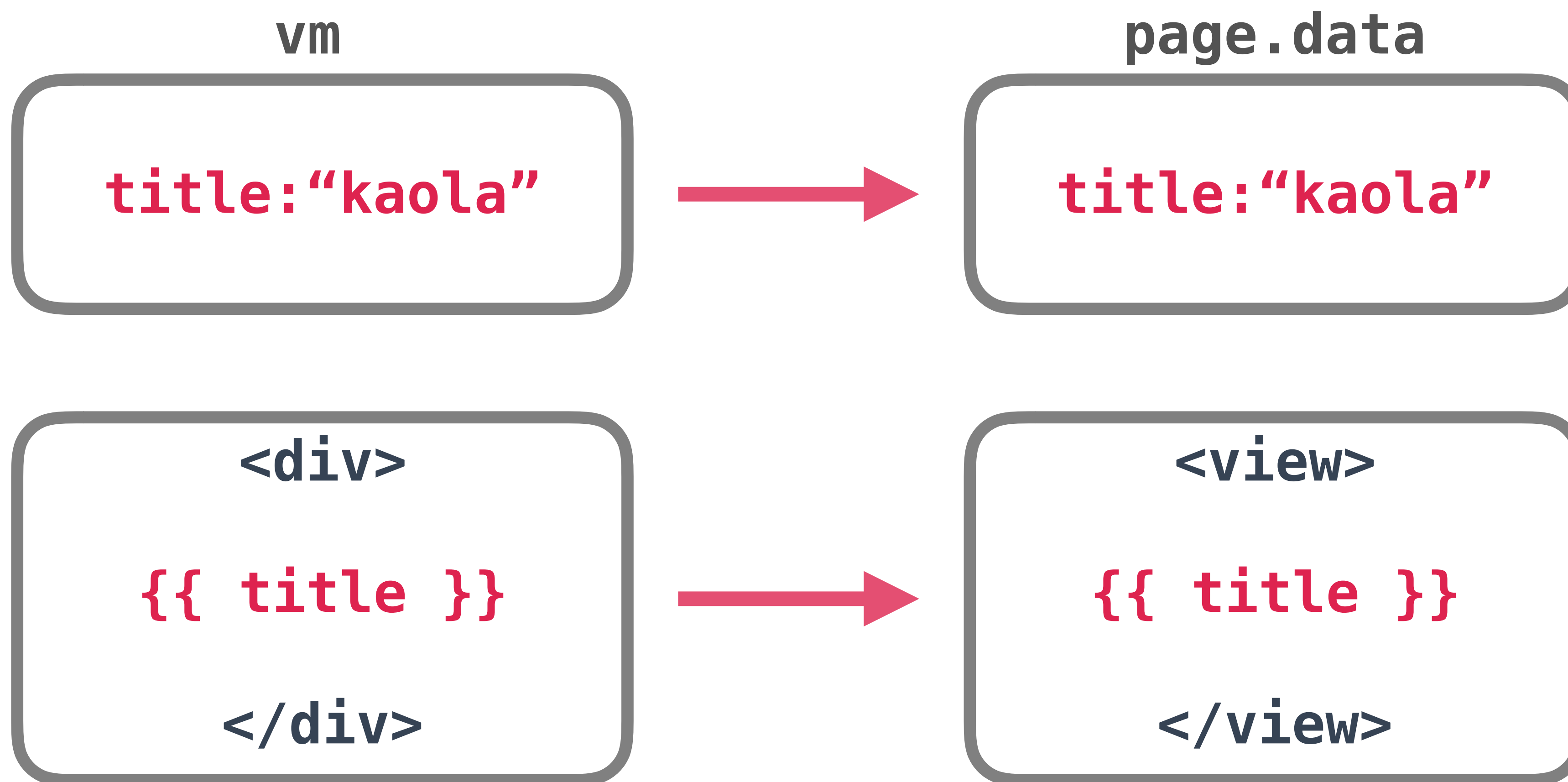
script

```
export default {  
  data() {  
    return {  
      title: "kaola"  
    }  
  }  
}
```

vm

title: "kaola"









频繁调用  
一次更新大量数据

`this.title = "2019"`

vm

```
title: "2019",  
user: {  
  name: "abc"  
}
```



page.data

```
title: "kaola",  
user: {  
  name: "abc"  
}
```

```
this.title = "2019"
```

```
setData({title: "2019"})
```

vm



page.data



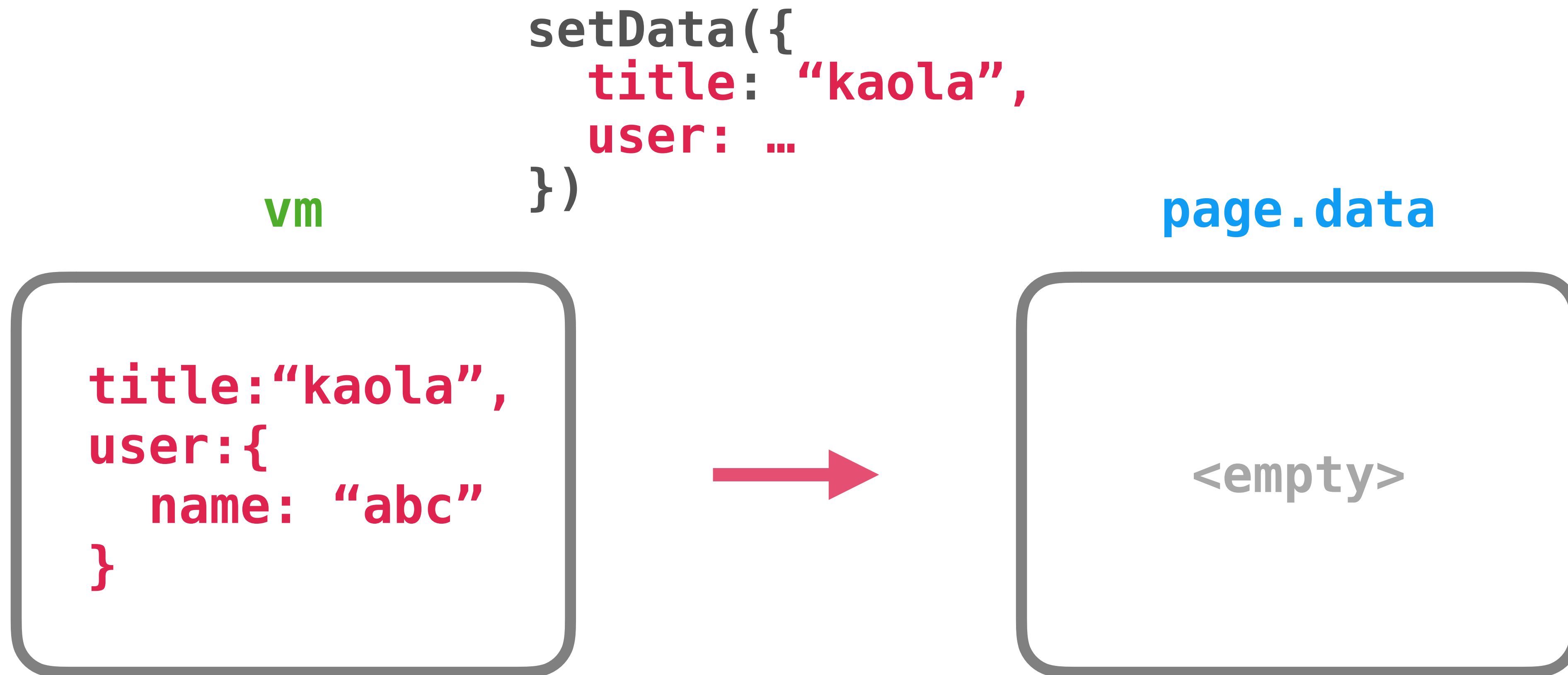
vm

```
title:"kaola",  
user:{  
  name: "abc"  
}
```



page.data

<empty>



page.data

```
title: "kaola",  
user: {  
  name: "abc"  
  // ...  
}
```

```
<view>{{ title }}</view>  
<view>{{ user.name }}</view>
```

page.data

```
title: "kaola",  
user: {  
  name: "abc",  
  cartList: [  
    // ...  
  ],  
  favorList: [  
    // ...  
  ]  
}
```

```
<view>{{ title }}</view>  
<view>{{ user.name }}</view>
```

← 可能是个大对象



```
<div :class="classObj">
```

```
  {{ date|format }}
```

```
</div>
```

```
<view class="{{classObj}}">  
    {{ date|format }}  
</view>
```

更新冗余的数据  
受小程序模版能力限制

```
<div :class="classObj">  
  {{ date | format }}  
</div>
```

tag: "div"  
class: "abc"

tag: "text"  
text: "2019"

```
<div :class="classObj"> ← 0
```

```
  {{ date | format }}
```

```
</div>
```

```
  ↑ 1
```

```
tag: "div"  
class: "abc"  
id: 0
```

```
tag: "text"  
text: "2019"  
id: 1
```

```
tag: "div"  
class: "abc"  
id: 0
```

```
tag: "text"  
text: "2019"  
id: 1
```

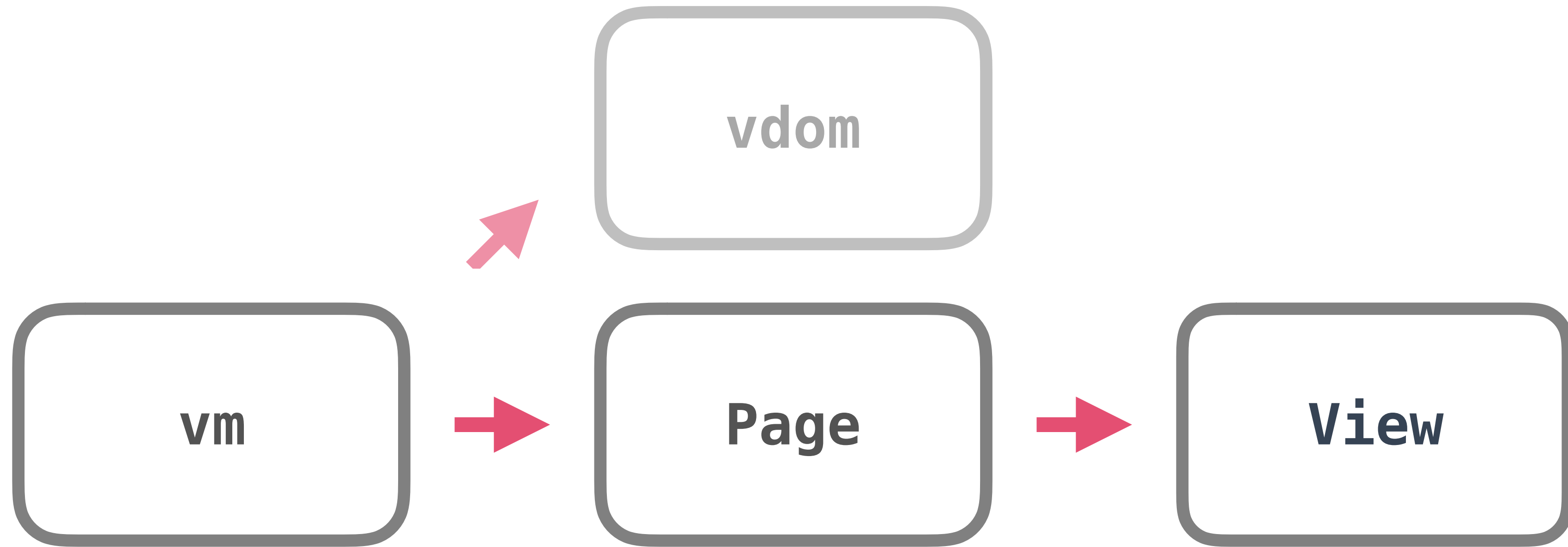
page.data

```
nodes: {  
  0: {  
    class: "abc"  
  }  
  1: {  
    text: "2018"  
  }  
}
```

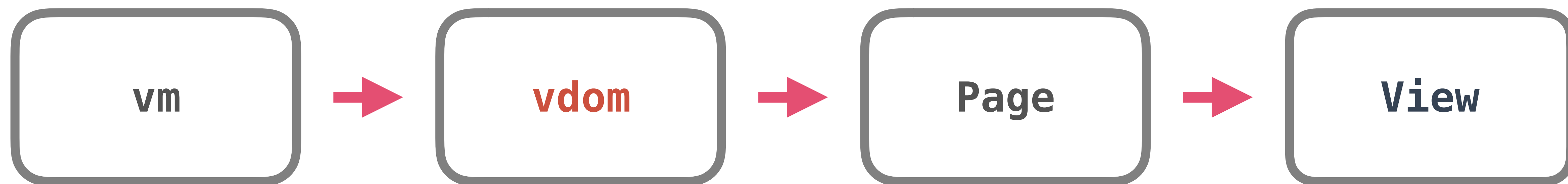
page.data

```
nodes: {  
  0: {  
    class: "abc"  
  }  
  1: {  
    text: "2018"  
  }  
}
```

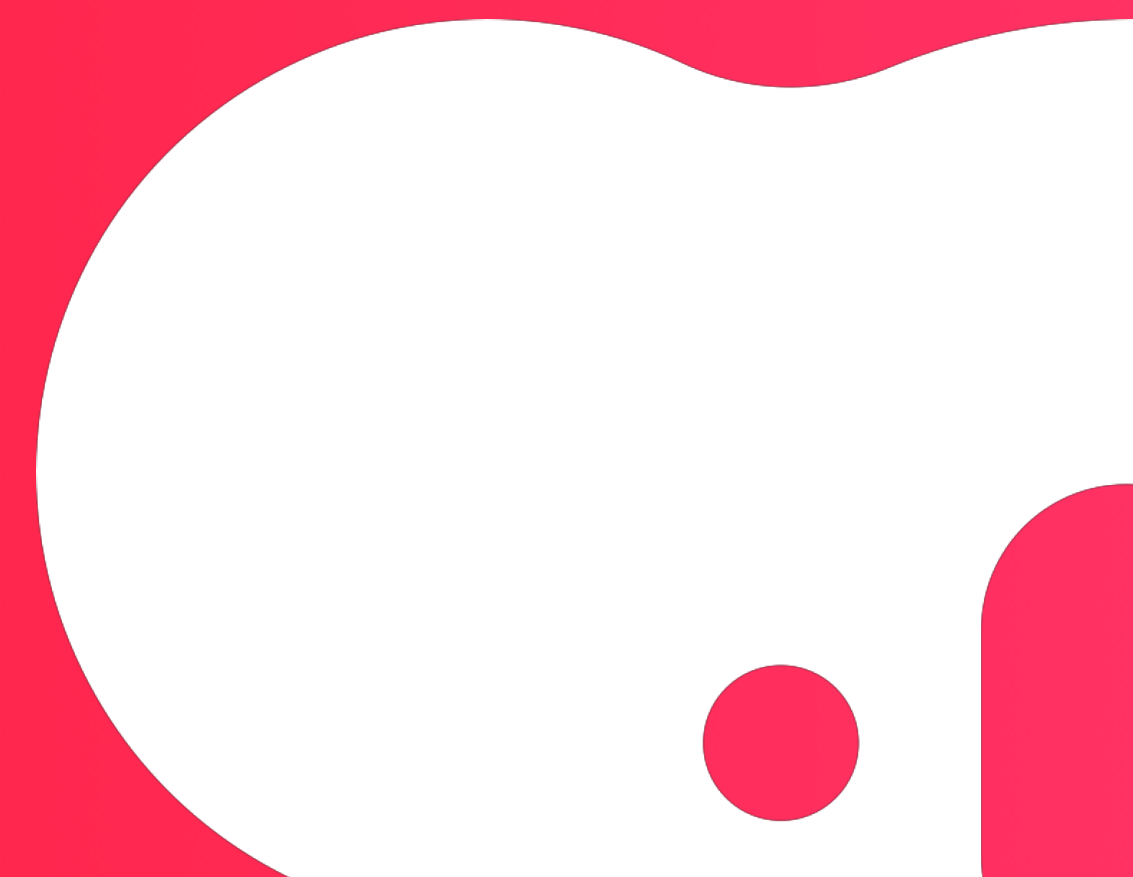
```
<view class="{{nodes[0].class}}">  
  {{nodes[1].text}}  
</view>
```







# Megalo



特性支持	小程序	mpvue	megalo
computed	×	○	○
v-model	×	○	○
slot	○	○	○
scoped slot	×	×	○
filter	×	×	○
v-html	×	×	○
复杂表达式	×	×	○
小程序分包	○	○	○

现在

支付宝小程序

百度小程序

API 库

UI 库

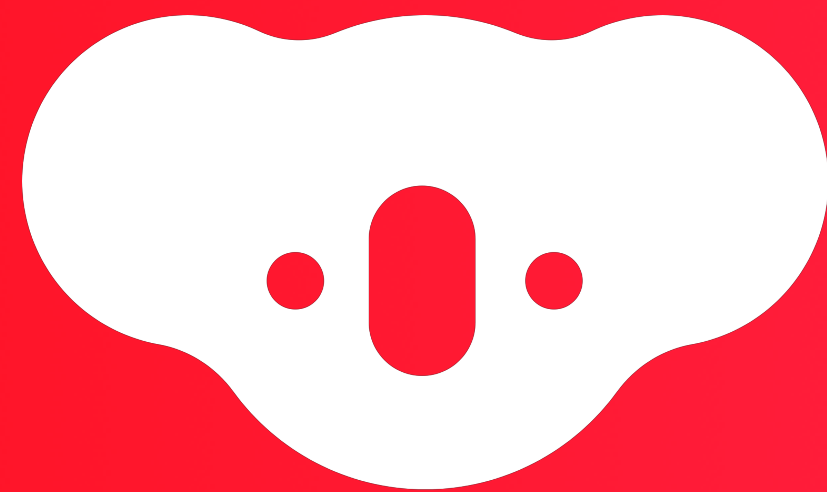
脚手架



头条小程序  
完善跨 H5 端方案  
完善配套设施







THANK YOU !