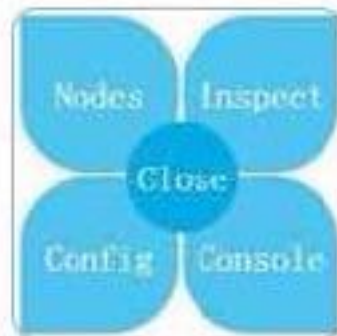


vue/vuex构建大型web应用：VIDE

王骆菲

webOS开发
陆金所前端开发



softwang 

上海 浦东新区



扫一扫上面的二维码图案，加我微信

| 提纲

1. IDE架构演化
2. plugin的新玩法
3. 对vue期待
4. VIDE展望

VIDE是什么

先看看常规设计思路

| IDE架构设计思路

1. 组件模块化
2. 组件之间能通信
3. 能提供事件，第三方可以接受事件

开始设计组件



组件的事件

- 设计需要的事件
- 写相关逻辑
- 集合多个状态值

```
// 事件处理
var bus = new Vue()

bus.$emit('plus', 1)
bus.$emit('minus', 1)

// in component B's created hook
bus.$on('plus', function (n) {
  ...
  this.total += n
})

bus.$on('minus', function (n) {
  ...
  this.total -= n
})
/*
还有很多事件。。。。。。
*/
```

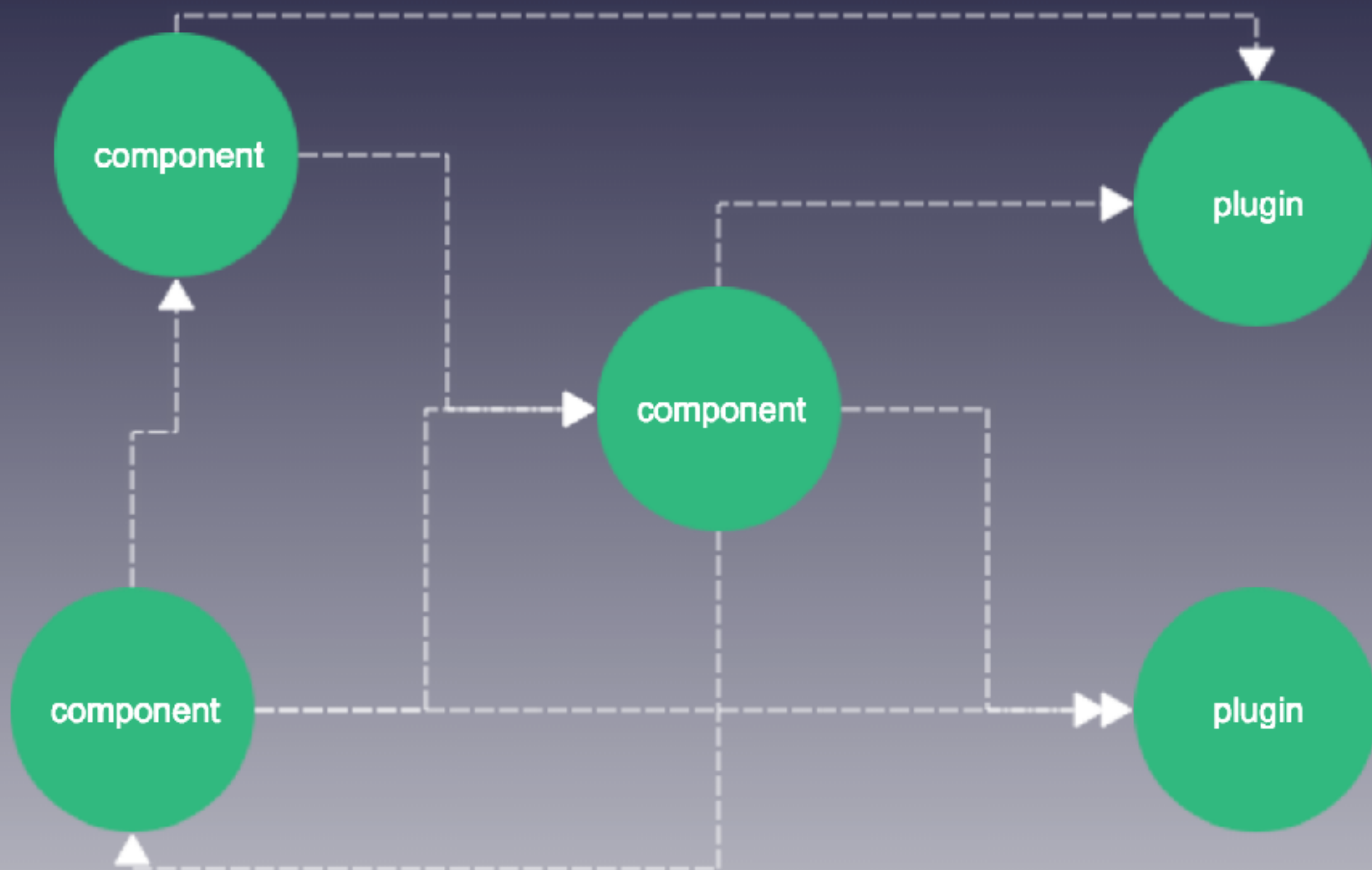

组件的逻辑处理

- 写相应的渲染事件
- 写相应的逻辑事件
- 还要发送事件

```
// 页面处理逻辑
{
  showSearchBox: function () {
    // .....
  },
  hideSearchBox: function () {
    // .....
  },
  filterResult: function () {
    // .....
  },
  clickme: function () {
    // .....
  },
  sendStatus: function () {
    bus.$emit('status', status)
  }
}
```

| IDE的事件流程图

IDE系统



当组件越来越多
功能越来越复杂

问题慢慢开始暴露

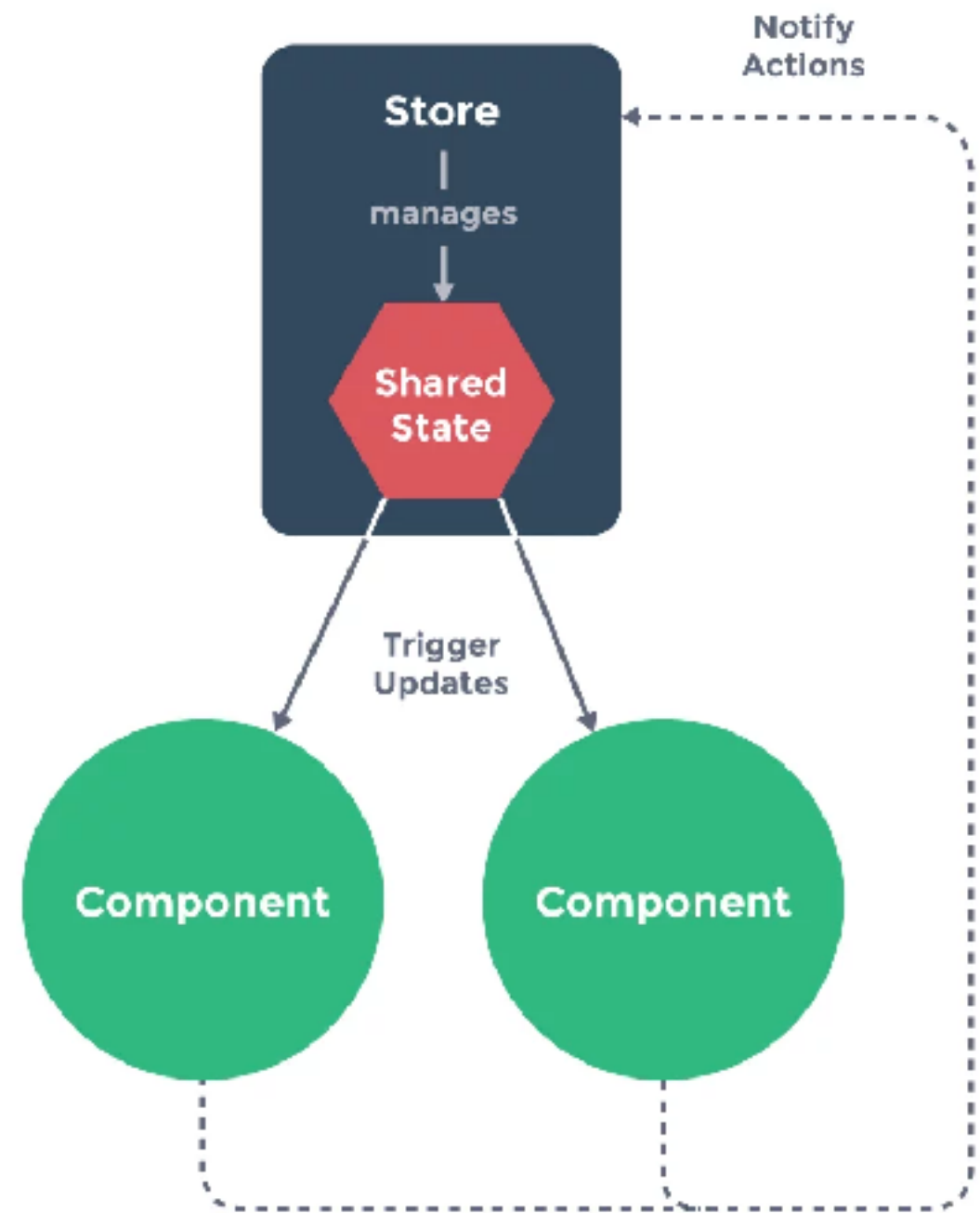
| 设计问题

1. 不能全局查看全部状态
2. 事件消息太多，维护太难
3. 集成多个状态，太麻烦
4. 太多状态事件跟逻辑混合在一起
5. ○ ○ ○ ○ ○

“某种东西的产生是由于社会的进步和发展到达一定的状态,出现了相应的条件,由此而产生”

vuex

集中式存储管理应用的所有
组件的状态



| vuex状态管理

```
const state = {  
  currentFile: '', // 当前打开文件的路径  
  content: '', // 编辑器的内容  
  anchor: {row: 0, column: 0}, // 当前光标位置  
  fileMode: '',  
  fileType: 'text', // 文件类型  
  promptLists: [], // 提示框列表内容  
  promptStr: '',  
  promptName: ''  
}
```

协同开发，只关心有哪些状态可用，不关心具体实现逻辑

| 解决问题

1. 不能全局查看全部状态
2. 事件消息太多，维护太难



vuex的store统一管理

| event bus状态集成

```
// event bus
var bus = new Vue()

bus.on('support', function (n) {
  // 逻辑代码
  showEverything()
})

bus.on('care', function (n) {
  // 逻辑代码
  showEverything()
})

bus.on('love', function (n) {
  // 逻辑代码
  showEverything()
})

function showEverything () {
  // 渲染组件
}
```

| 解决问题

集成多个状态，太麻烦



```
computed: {  
  ...mapState({  
    everything: state => {  
      return state.parents.care + state.wife.love + state.friends.support  
    }  
  })  
},
```

| 解决问题

太多状态事件跟逻辑混合在一起



vue/vuex处理数据+逻辑处理

| 重构后IDE

1. 渲染委托给vue
2. 事件通信委托给vuex
3. 组件简单明了



IDE系统

vuex

auto update

component

component

plugin

plugin

notify action

notify action

组件系统

插件系统

| plugin新玩法

| plugin新玩法

1. 用vue写视图 + 事件处理类



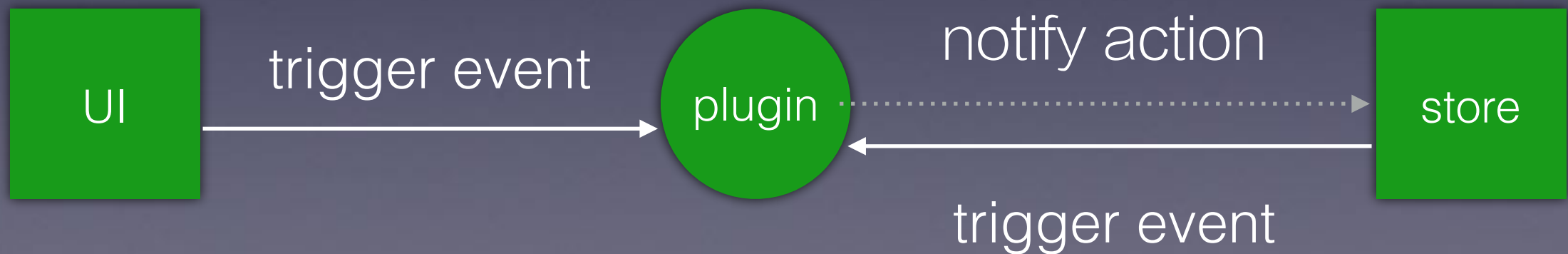
| plugin新玩法

1. 用vue写视图 + 事件处理类
2. vuex管理插件



| plugin新玩法

1. 用vue写视图 + 事件处理类
2. vuex管理插件
3. 监听事件，处理事件



| plugin开发—导入组件

```
import App from './App.vue'  
import path from 'path'  
export default ({editor, store, view, packageInfo, baseClass}) => {  
}
```

| plugin开发—添加视图

```
import App from './App.vue'  
import path from 'path'  
export default ({editor, store, view, packageInfo, baseClass}) => {  
  // 添加工具栏  
  store.dispatch('toolbar/addItem', {  
    name: 'hosts',  
    desc: 'hosts tool',  
    key: 'videPluginToolbarHostsItem',  
    icon: 'data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAADMAAAAsCAYAAAB',  
    func: 'videPluginToolbarHosts:click'  
  })  
}
```

| plugin开发—添加处理类

```
// return execute class
return class videPluginToolbarHosts extends baseClass {
  click () {
    if (isCreated) {
      this.$toggle()
    } else {
      isCreated = true
      let self = this
      let props = {
        propsData: {
          callback: function () {
            self.$toggle()
          }
        }
      }
    }
    let stylePath = path.join(packageInfo.path, './dist/index.css')
    this.$mount({app: App, props, stylePath})
  }
}
$clean () {
  store.dispatch('toolbar/deleteItem', 'videPluginToolbarHostsItem')
}
}
```

```

import App from './App.vue'
import path from 'path'
export default ({editor, store, view, packageInfo, baseClass}) => {
  let isCreated = false
  // add item to toolbar
  store.dispatch('toolbar/addItem', {
    name: 'hosts',
    desc: 'hosts tool',
    key: 'videPluginToolbarHostsItem',
    icon: 'data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAADMAAAAzCAYAA
    func: 'videPluginToolbarHosts:click'
  })
  // return execute class
  return class videPluginToolbarHosts extends baseClass {
    click () {
      if (isCreated) {
        this.$toggle()
      } else {
        isCreated = true
        let self = this
        let props = {
          propsData: {
            callback: function () {
              self.$toggle()
            }
          }
        }
        let stylePath = path.join(packageInfo.path, './dist/index.css')
        this.$mount({app: App, props, stylePath})
      }
    }
    $clean () {
      store.dispatch('toolbar/deleteItem', 'videPluginToolbarHostsItem')
    }
  }
}

```

现场show一下插件

对vue的期待

1. 提供好的代码检测工具?
2. 提供性能分析工具?

ivide的未来规划

1. 未来优先支持vue/weex开发
2. 再支持微信小程序等
3. 基于简单快速的插件开发，服务更多的开发者

520

- 大爱vue/vuex产品
- 大爱element-ui
- 超爱我老婆

Q&A

