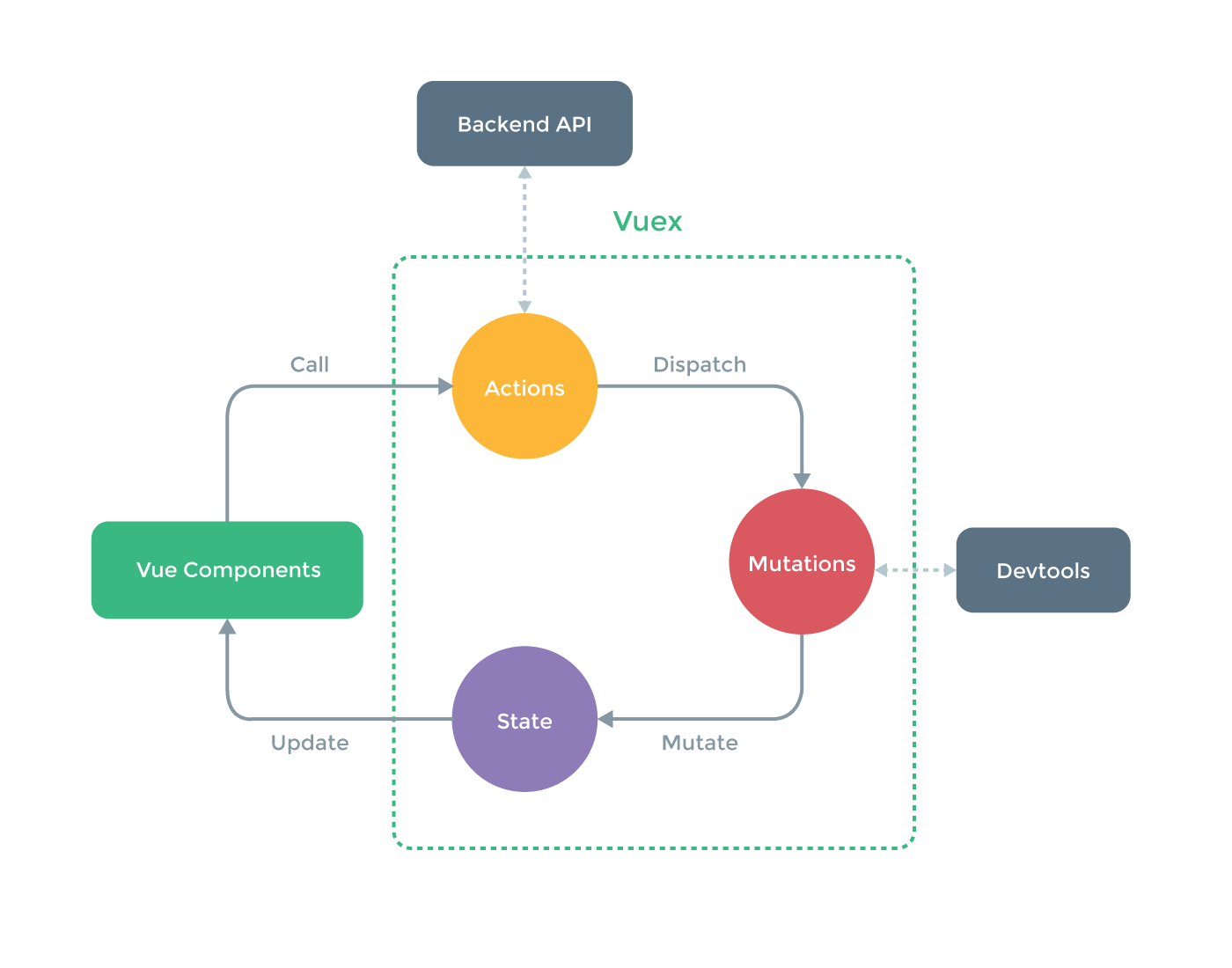
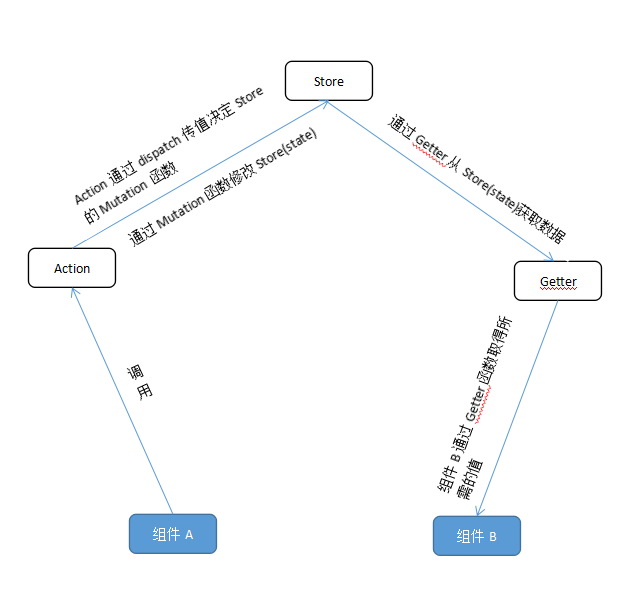
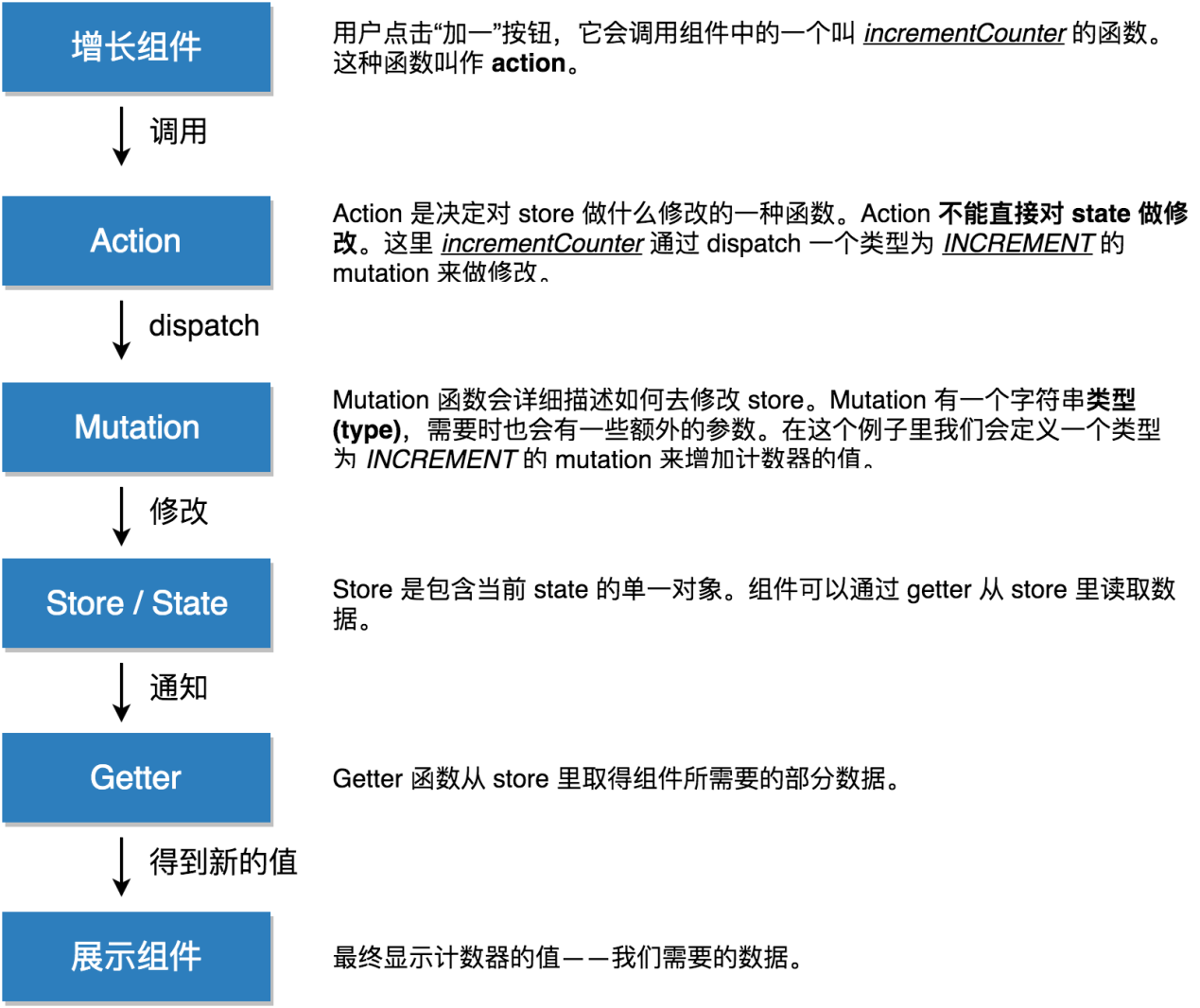
总思路：







从获取消息列表数据来说明一下执行顺序

参考：

<http://www.tuicool.com/articles/vQZbYn7>

<http://zhangdiweb.com/2016/09/08/Vuex%E2%80%94%E2%80%94Vue%E5%8D%95%E9%A1%B5%E5%BA%94%E7%94%A8%E7%8A%B6%E6%80%81%E7%AE%A1%E7%90%86%E6%9E%B6%E6%9E%84/>

<http://www.html-js.com/article/3663>

1. store.js使用Vue.use(Vuex)之后export new Vuex.Store某个module如modules/messageList
2. messageList/messageList.js引入import store from '../../vuex/store'，然后在new Vue载入store，相当于注入根组件，使得子组件通过this.$store实现组件之间通讯
3. messageList.vue中created方法使用this.$store.dispatch('getMessageList')（mutations 也可以通过这个来触发）触发actions中的getMessageList
4. messageList/messageList.js中actions中getMessageList方法使用commit(types.MESSAGE\_GET\_LIST, dataStr)提交mutation
5. mutation中方法[types.MESSAGE\_GET\_LIST] (state, res) {}赋值给state中的banner
6. messageList.vue中使用mapGetters调用方法名为getMessageListGetstate的getters赋值后的新数据
   1. computed: {  
      mapGetters**(**{  
       **banner**: **'getMessageListGet'** }**)**},
7. 最后前端使用渲染返回的数据
   1. <**ul**>  
       <**li v-for='item in banner'**>{{item}}</**li**>  
      </**ul**>

附注computed加上更多方法的话可以这样加上

* + 1. computed: {  
        a **()** {  
        **return 'a'** },  
        ...mapGetters**(**{  
        **banner**: **'getMessageList'** }**)**},