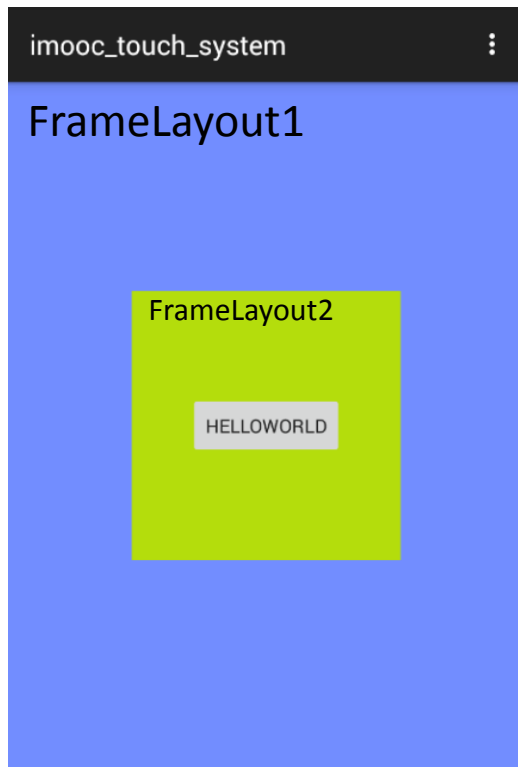


Android事件分发机制

为什么需要事件分发机制？



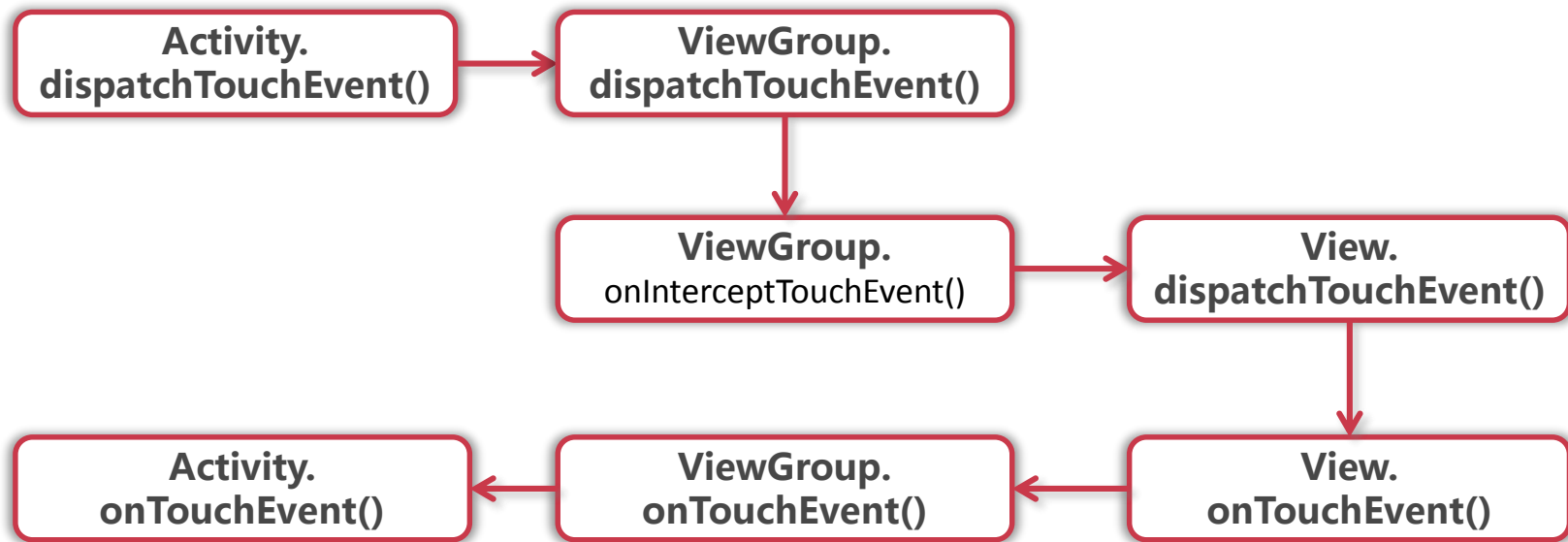
常见的事件分发分为两种

- 冒泡（自下而上的过程）
- 捕获（自上而下的过程）

Android中的事件分发机制是样子呢？



Android中的事件分发机制



注意：严格来说以上流程只是ACTION_DOWN的一种特殊的情况

细节？



对事件感兴趣的View



View自身如何表明对事件感兴趣呢？

最主要是View.dispatchOnTouchEvent()在ACTION_DOWN的时候返回true。

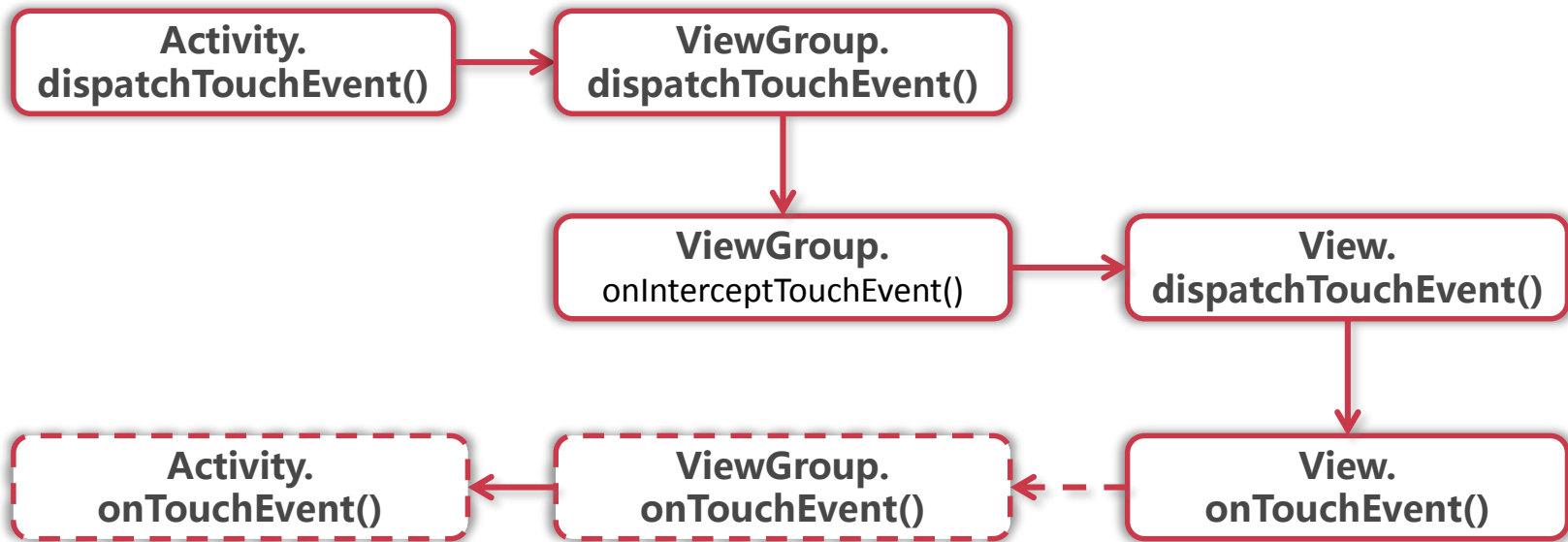
但是一般情况下，我们主要重写的方法是onTouchEvent，所以要保证ACTION_DOWN返回true。

注：凡是clickable = true 或者 longClickable = true 的控件，正常情况下View.onTouchEvent()一定返回true。



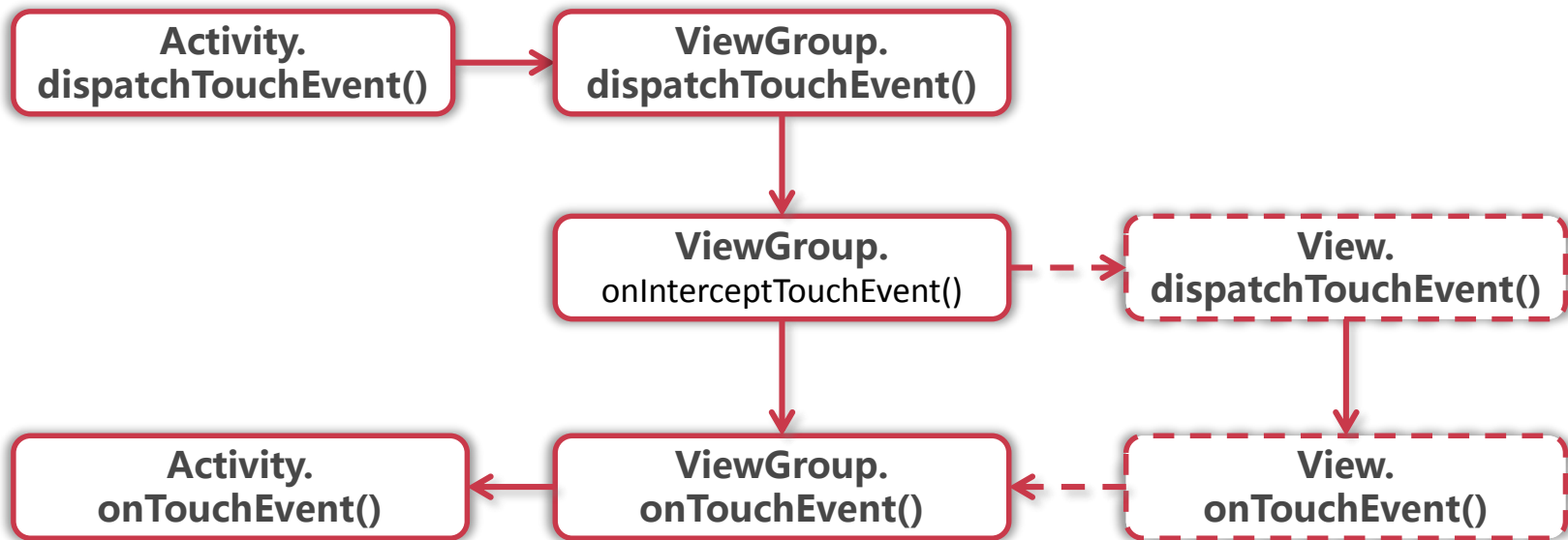
对事件感兴趣的View

DOWN-MOVE*-UP



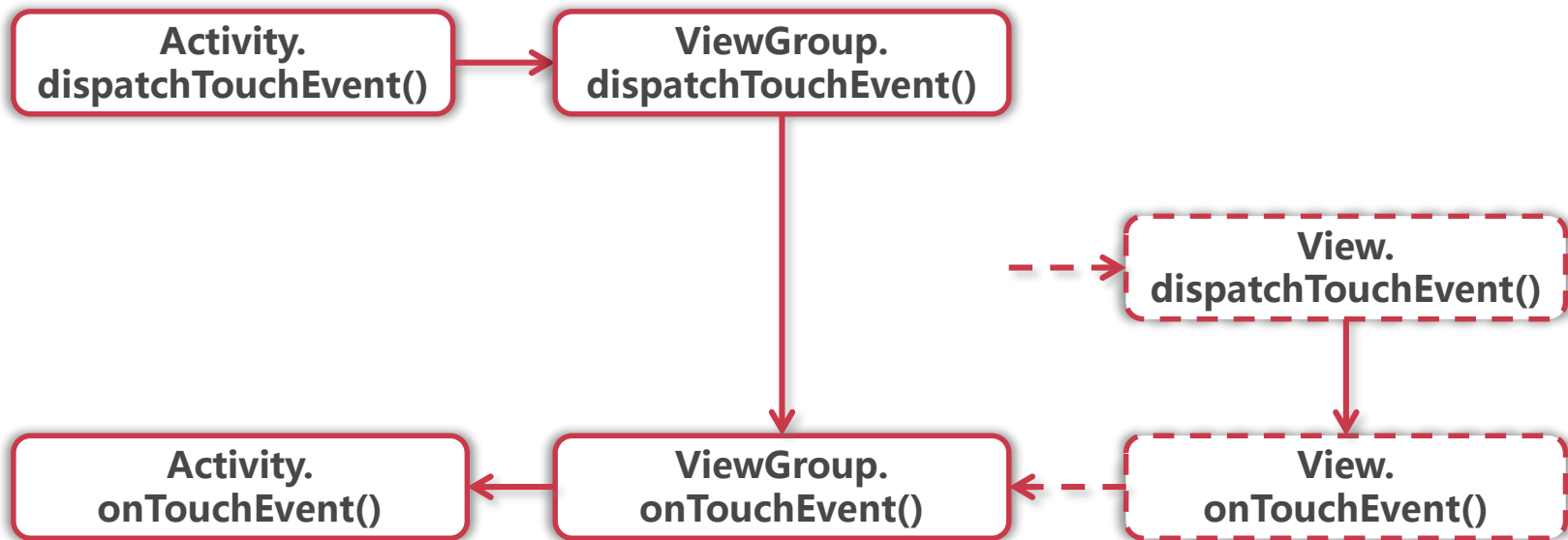
ViewGroup对事件进行拦截

DOWN



ViewGroup对事件进行拦截

MOVE*-UP



View与ViewGroup相关方法职责

➤ dispatchTouchEvent

ACTION_DOWN的时候：逆序遍历子View，找出对该事件感兴趣的View，标记为targetView，然后对于该手势（DOWN-MOVE*-UP）的后续事件都传给targetView。

➤ onInterceptTouchEvent

在ACTION_DOWN，或者存在targetView的情况下，可以随时对该事件进行拦截，交给自己处理。

➤ onTouchEvent

拿到事件后做针对当前View的相关操作。

自定义View中的事件分发

- 处理Touch事件
 - 可以编写setOnTouchListener(无需继承View)
 - 复写onTouchEvent
- 注意标明对事件感兴趣
 - 如果需要自己获取touch事件进行处理，ACTION_DOWN必须返回true，保证整个手势的事件都能够传递到该View。

自定义ViewGroup中的事件分发

- 包含上述View的所有事项
- 拦截子View事件
 - 可以在onInterceptTouchEvent()中子View的事件进行拦截，交给自己的onTouchEvent进行处理。
 - 注意：一旦拦截针对当然的手势所有事件都将由当前的ViewGroup处理。会传递一个ACTION_CANCEL交给当前的子View，让子View明白后续的事件不会到来了。

其他？

- **requestDisallowInterceptTouchEvent**
- **ViewConfiguration**
 - **getScaledTouchSlop()**
 - **getScaledMinimumFlingVelocity()**
 - **getLongPressTimeout()**
- **OnScrollListener / View.onScrollChanged()**
- **GestureDetector**
- **ScaleGestureDetector**

扩展学习

- **Mastering the Android Touch System**

- 视频 (中文字幕)

- http://v.youku.com/v_show/id_XODQ1MjI2MDQ0.html?from=s1.8-1-1.2

- 英文 (Google搜索下)

- 代码

- <https://github.com/devunwired/custom-touc-examples>

Thank You !