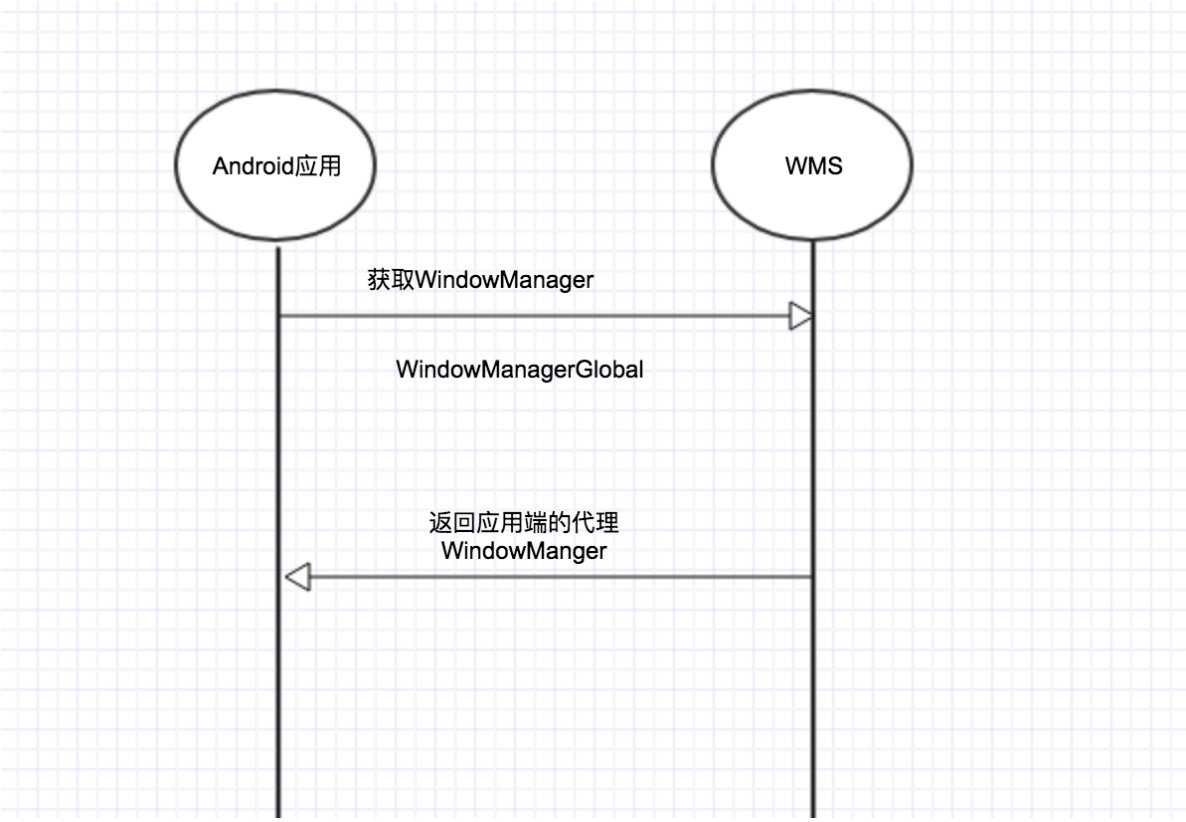
一:WMS是什么

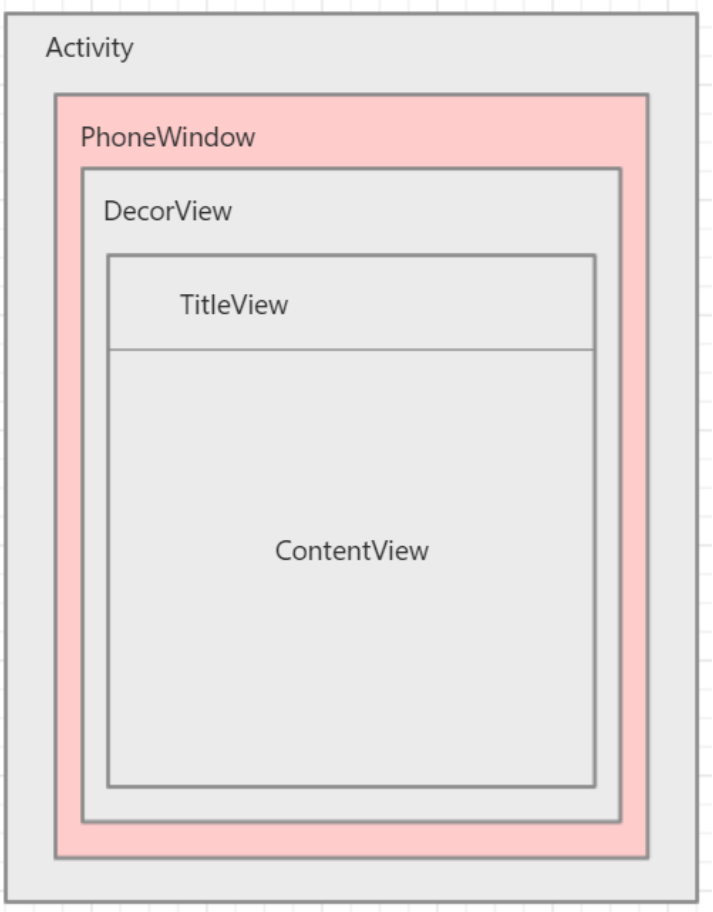
WMS是WindowManagerService的简称，是位于SystemServer的一个进程，负责android中窗口的创建以及管理，在我们应用开发中用到的是WindowManager,它是位于应用端的Client, 通过AIDL实现的跨进城通信。

- 在应用启动之后，会创建一个唯一实例(SystemServiceRegistry:558)
- 在创建实例的时候，会获取一个WindowManagerGlobal单例，这个里面会获取位于SystemServer的本地代理(WindowManagerGlobal:164)



二: Activity里的应用

我们开发中所看到的一个Activity的view层次如下图所示，(比较老的，新的系统会多更多)



- PhoneWindow就是一个window,在activity启动的时候创建(Activity:6620)，如何执行到这个创建的过程涉及到AMS(ActivityThread:2599),这里就不深入，下次再说
- DecorView是 PhoneWindow下的唯一且直接的view(PhoneWindow:311)
- 在DecorView下面再添加很多默认的系统view，比如NavigationBar之类，不是今天的主题，略过
- 在创建完窗口之后，当然是添加到系统的图层上，这时候就会调用activity的显示，来通过今天的主角来添加view了(Activity:5128），这个名字跟ios的显示很像 makeVisible，不知道ios里面是不是也是用这么个类似的系统服务来添加窗口的。
- 然后开始一些参数检测以及调用wms在应用端的代理，来调用一些远程wms的方法来添加窗口了(WindowManagerGlobal: 331),这里可以看到一个ViewRootImpl，这个是所有窗口的根view，可以看到，它是把view加入到一个数组中的，我们应该都知道，ui的显示，是上层layer覆盖下层，这里就保证了你在添加view的过程中，是一层层盖上去的

＊ 到这里,Activity的界面就显示到了手机上了

三: Service里的应用

在我们开发中，经常会碰到会需要你做一个没有依赖于activity/controller的浮窗，这个时候，我们就需要通过wms直接添加窗口到系统层，那这里，我们就需要自己更新它的大小，位置等信息了。

- 首先创建一个view，这个view是什么样的，就根据业务需要
- 获取系统wms应用代理WindowManager，创建WindowManager.LayoutParams，必须是WindowManager的，否则会报错哦,因为会有这段检测

```
if (!(params instanceof WindowManager.LayoutParams)) {
    throw new IllegalArgumentException("Params must be WindowManager.LayoutParams");
}
```

- 后面就是addView的过程，与Activity一样
- 这里不一样的地方是要自己来更新它的大小以及位置，需要另外一个函数updateViewLayout(WindowManagerGlobal:355),通过更新WindowManager.LayoutParams，来实现窗口的更新
- 移动窗口就通过view的onTouch来实现监听位置，然后调用updateViewLayout，那另外一种，就是view的大小会根据不同情况有不同的

大小呢，不管android还是ios，你的view在设置了相对的大小，比如铺满全屏，或者设置为全屏的一半，这个是随便举的例子，那我们必须要在view在屏幕上渲染出来之后，我才能知道宽度，但是，我们更新view是要在它显示之前知道它的宽高的，怎么解决

- ios我不造，但是android这里我找到了一个方法，就是在onLayout的时候，去更新view的大小，这个时候是系统直接告诉你了view的大小，那你在这个view显示不同状态的ui的时候，只需要跟我们平时在activity一样，直接设置就行了,最后再调用一句requestLayout();来请求wms来重新布局

```
// ---------------- State
public void show(@OnlineLiveFloatState int state) {
    if (mState == state) {
        return;
    }
    mState = state;
    mTvMsg.setSelected(mState == STATE_MSG_DETAIL);
    if (mState == STATE_DEFAULT) {
        mRlIcon.setVisibility(GONE);
        mRlContent.setVisibility(VISIBLE);
        mRlMenu.setVisibility(GONE);
        mRlMsg.setVisibility(GONE);
        mViewDiver.setVisibility(GONE);
        mLp.flags = C.FLAG_NOT_FOCUSABLE;
    } else if (mState == STATE_ICON) {
        mRlContent.setVisibility(GONE);
        mRlIcon.setVisibility(VISIBLE);
        mLp.flags = C.FLAG_NOT_FOCUSABLE;
    } else if (mState == STATE_MENU) {
        mRlIcon.setVisibility(GONE);
        mRlContent.setVisibility(VISIBLE);
        mRlMenu.setVisibility(VISIBLE);
        mRlMsg.setVisibility(GONE);
        mViewDiver.setVisibility(VISIBLE);
        mLp.flags = C.FLAG_NOT_FOCUSABLE;
    } else if (mState == STATE_MSG_DETAIL) {
        mRlIcon.setVisibility(GONE);
        mRlContent.setVisibility(VISIBLE);
        mRlMenu.setVisibility(GONE);
        mRlMsg.setVisibility(VISIBLE);
        mViewDiver.setVisibility(VISIBLE);
        mLp.flags = C.FLAG_FOCUSABLE;
    }
```