## 锁屏是什么

锁屏在系统中叫做KeyGuard，在灭屏以及系统准备就绪时由PhoneWindowManager启动。

KeyGuard在原生系统设置中有五种模式，分别为：

* 无
* 滑动(wipe)
* 图案(patten)
* Pin码(Pin)（开机过程中会出现两次，一次是还未开机时，一次是完成开机）
* 密码(password)

需要补充的是，在系统中其实还有几类锁屏也归KeyGuard管控。

* Biometric（指纹、虹膜解锁等生物学解锁）
* Account（忘记密码后，用账号密码）
* SimPin（sim卡pin码，用于验证SD卡，自己设置）
* SimPuk（pin码输入失败三次后出现，密码在卡上有）

5.1版本KeyGuard在目录frameworks/base/packages/Keyguard中编为java lib包，由SystemUI导入。部分关于锁屏的操作是由SystemUI负责。

## 锁屏的结构

锁屏示意图：（选取两张比较有代表性的）



图1

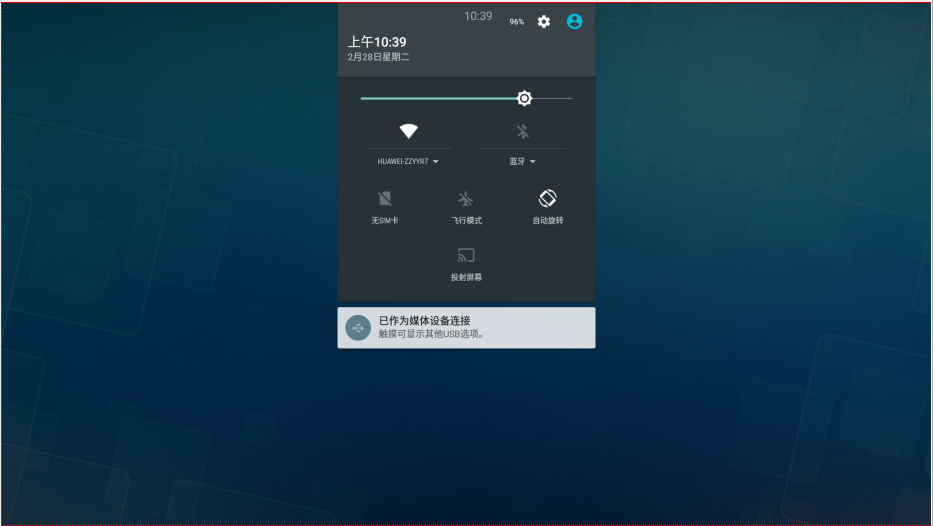
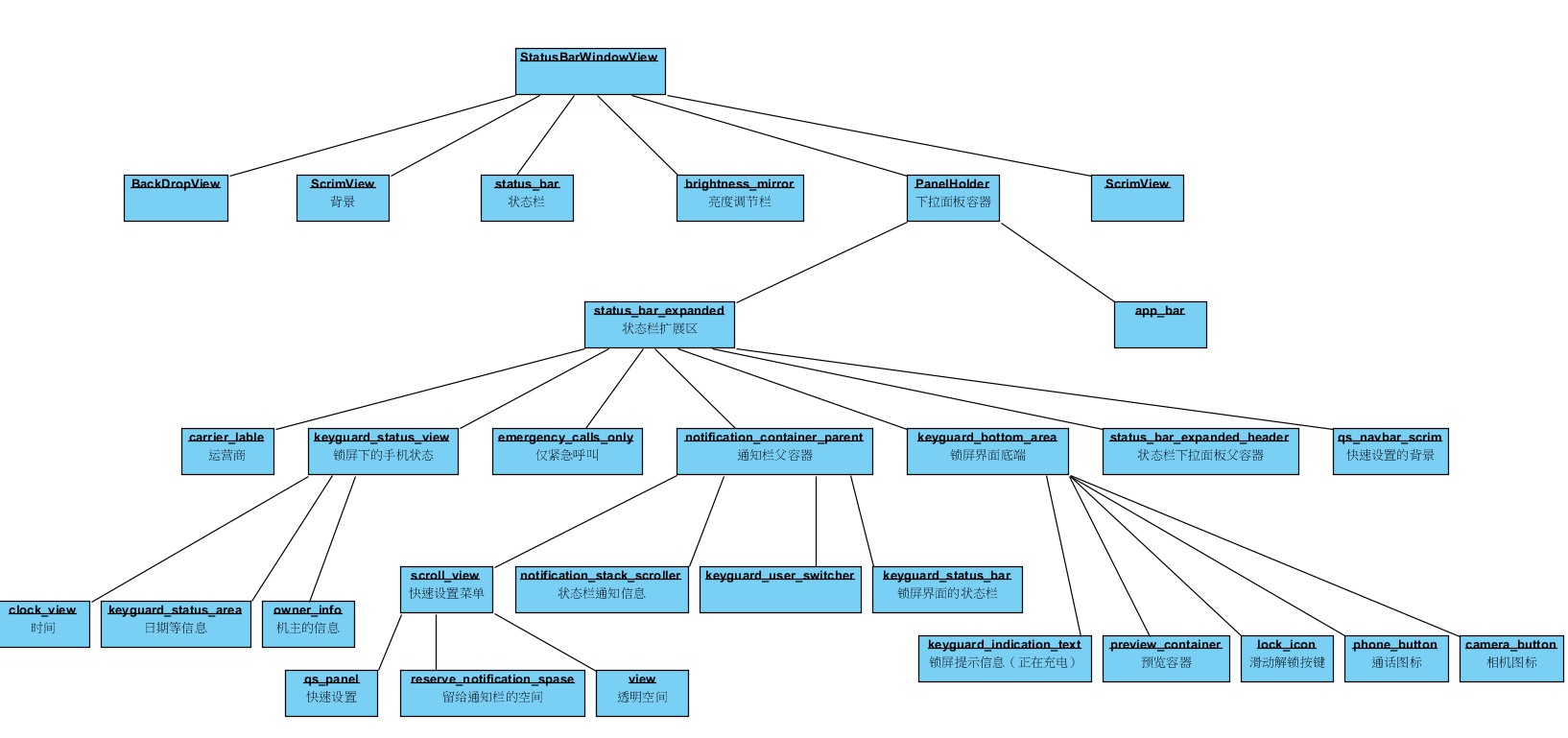


图2

而其布局结构看附图。（图看不清可以放大或另外打开）



可以看到，锁屏时，绝大多数界面布局都在statu\_bar\_expanded中。

图1主要显示的是：

carrier\_lable（运营商）：左上角运营商

keyguard\_status\_view（锁屏下的手机状态）：中间部分的时间、日期

notification\_stack\_scroller（状态栏通知信息）：中间“已作为XX”的通知栏

keyguard\_status\_bar（锁屏界面的状态栏）：顶端状态栏wifi等

keyguard\_bottom\_area（锁屏界面底端）：底部的正在充电以及锁形按钮

图2主要区别有：

scroll\_view（快速设置菜单）：下拉的快速设置菜单出现

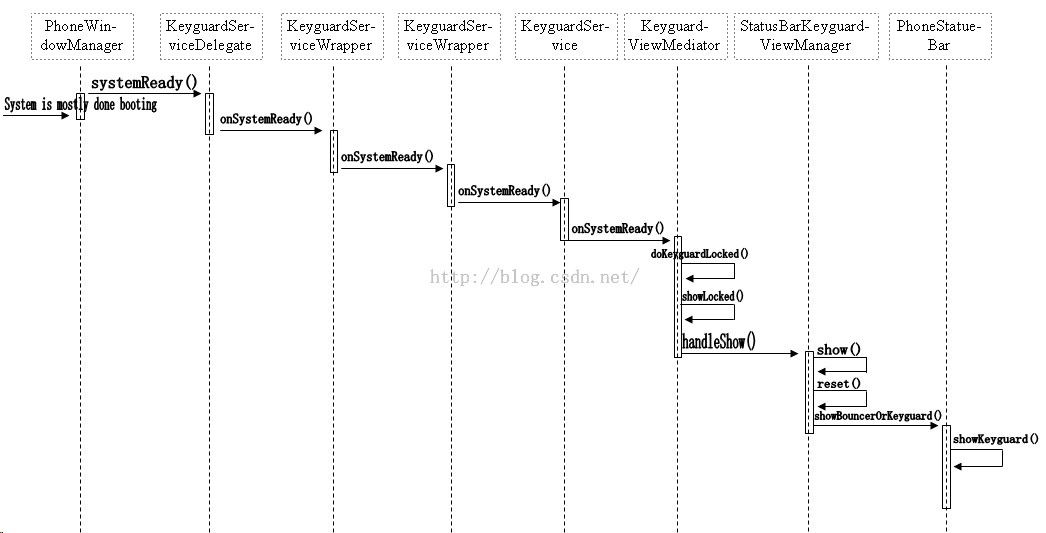
keyguard\_bottom\_area、keyguard\_status\_view等多数界面隐藏。

## 锁屏流程

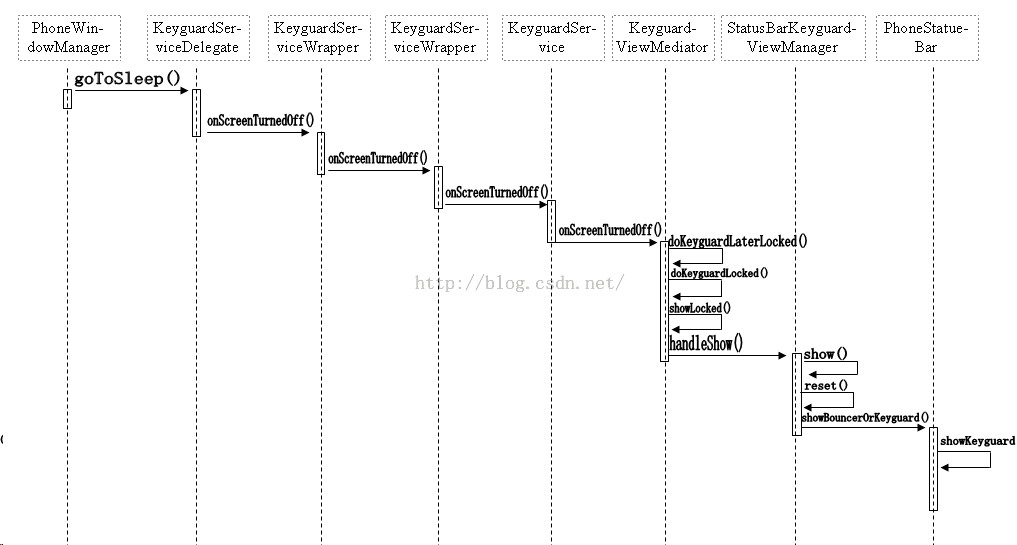
锁屏分作两种方式进行锁屏，分别是系统准备就绪以及灭屏。

代码逻辑时序图：（觉得小的可以放大）

systemReady show keyguard大致流程图：



screen off show keyguard大致流程：



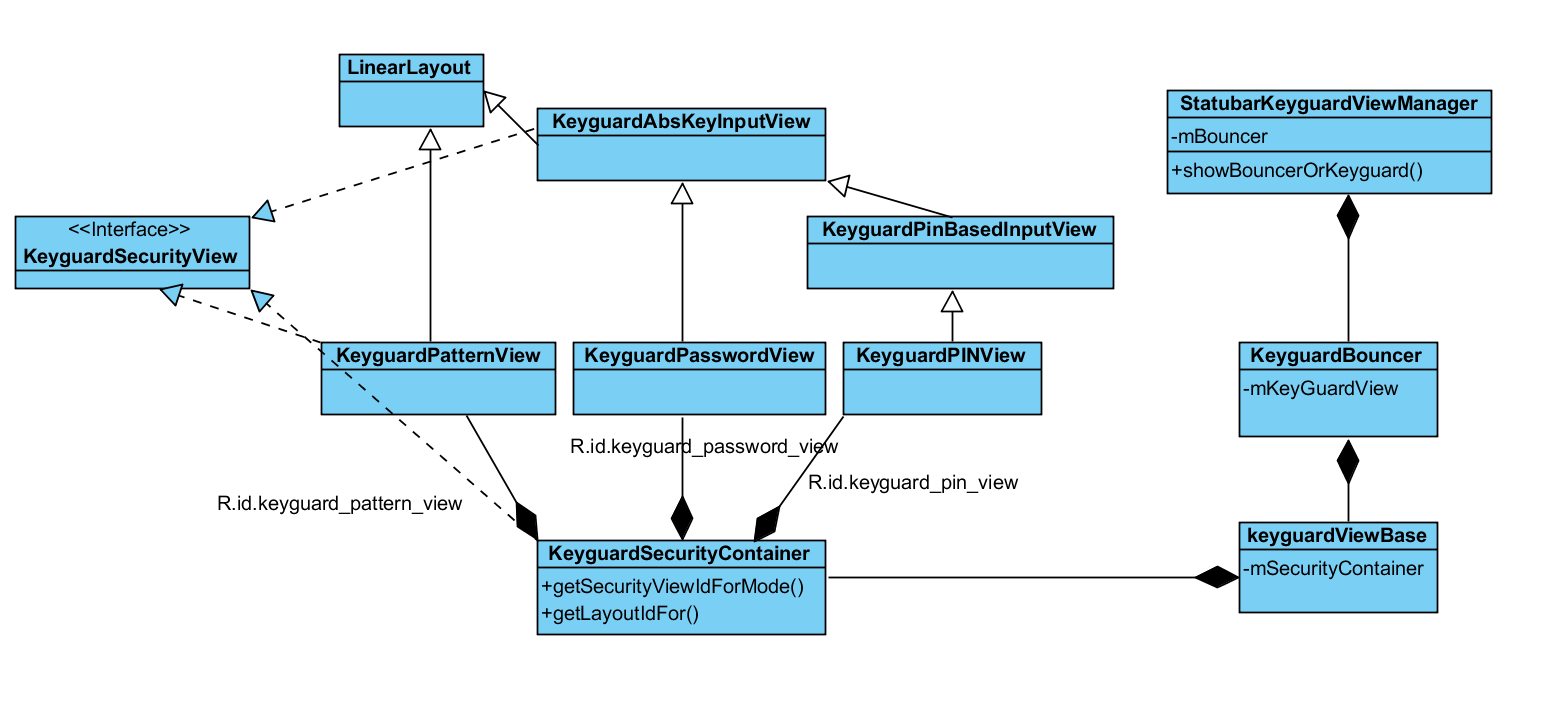
可以看到基本都是一条线的时序，很容易理解。

需要特别说明的是，滑动解锁（wipe）是其他三种解锁方式的前置界面。比如，只有当滑动解锁了之后才能进行密码解锁。

其他三种解锁方式被统称为Bouncer，在StatuBarKeyguardViewManager中的

showBouncerOrKeyguard()会进行判断。

## Bouncer流程



三种解锁方式分别对应类与布局文件：

* 图案(patten)：KeyguardPattenView 以及 keyguard\_patten\_view.xml
* Pin码(Pin)：KeyguardPinView 以及 keyguard\_pin\_view.xml
* 密码(password)：KeyguardPasswordView 以及 keyguard\_password\_view.xml

三者最终都继承到LinearLayout。

这三者在作为layout和id在KeyguardSecurityContainer中取得。

而KeyguardSecurityContainer被父容器keyguardViewBase包含。

keyguardViewBase在KeyguardBouncer作为一个属性存在。

而在StatubarKeyguardViewManager的showBouncerOrKeyguard方法中，决定了是否调用KeyguardBouncer的show方法。

这就是一个大致的Bouncer调用流程。