



APKBUS

Android最稳定和高效的适配方案

吴钊

Android的适配

- 系统版本适配
- Android UI适配

Android Q 行为变更：所有应用

Android Q 平台做了一些行为变更，这些变更可能会影响您的应用。以下行为变更将影响在 Android Q 上运行的所有应用，无论其采用哪种“targetSdkVersion”都不例外。您应该测试您的应用，然后根据需要进行更改以适当地支持这些变更（如果适用）。

此外，请务必查看[仅影响以 Android Q 为目标平台的应用的行为变更](#)列表。

行为变更：所有应用

Android 9（API 级别 28）向 Android 系统引入了多项变更。当应用在 Android 9 平台上运行时，以下行为变更将影响*所有应用*，无论这些应用以哪个 API 级别为目标。所有开发者都应查看这些变更，并修改其应用以正确支持这些变更（如果适用）。

如需了解仅影响以 API 28 或更高级别为目标的应用的变更，请参阅[行为变更：以 API 级别 28+ 为目标的应用](#)。

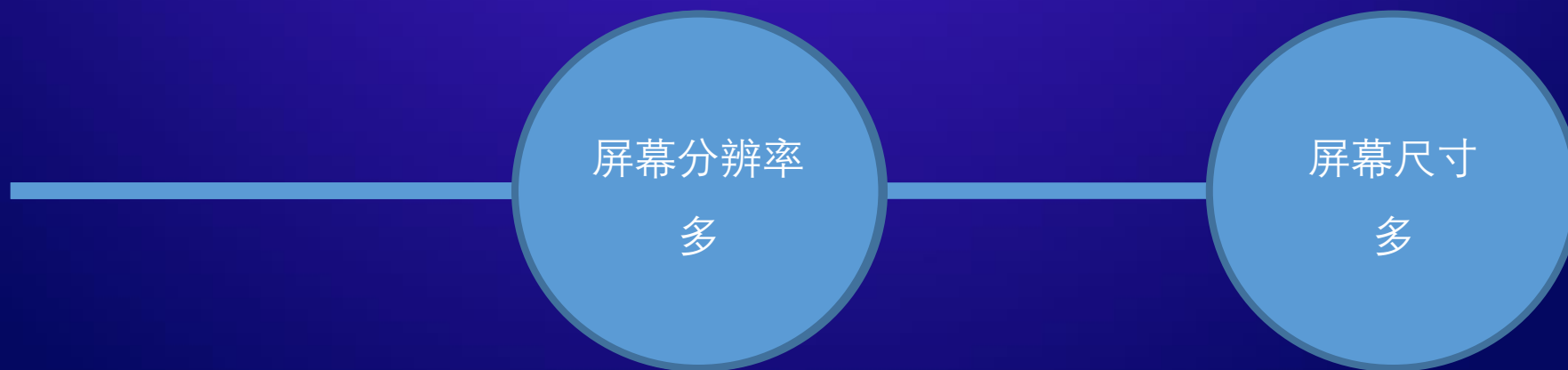
Android 8.0 行为变更

Android 8.0 除了提供诸多新特性和功能外，还对系统和 API 行为做出了各种变更。本文重点介绍您应该了解并在开发应用时加以考虑的一些主要变更。

其中大部分变更会影响所有应用，而不论应用针对的是何种版本的 Android。不过，有几项变更仅影响针对 Android 8.0 的应用。为清楚起见，本页面分为两个部分：[针对所有 API 级别的应用](#)和[针对 Android 8.0 的应用](#)。



</> Android UI适配难点

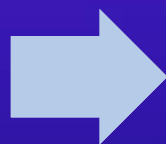


</> 适配方案的评判标准

高效

1

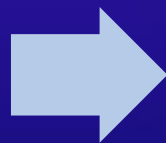
Layout Code



All Screen

2

Layout Code

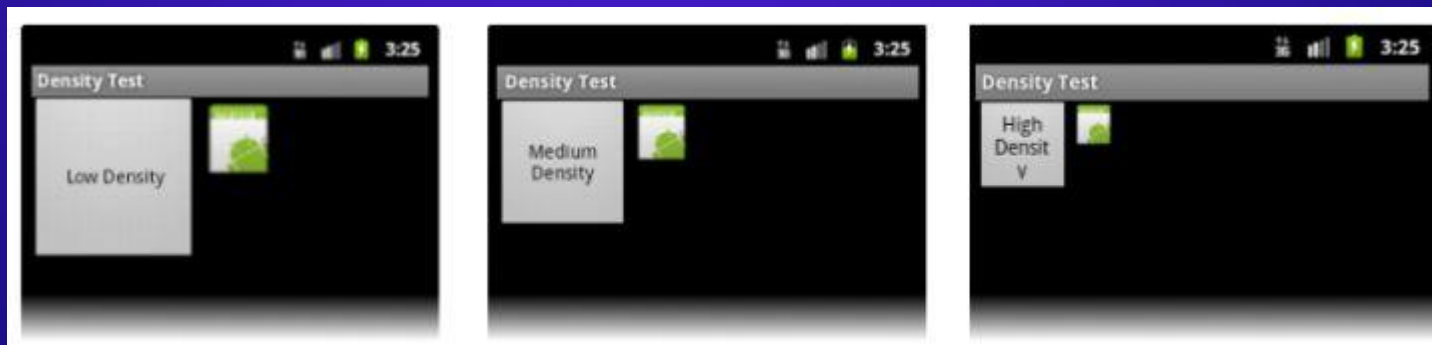


All Screen

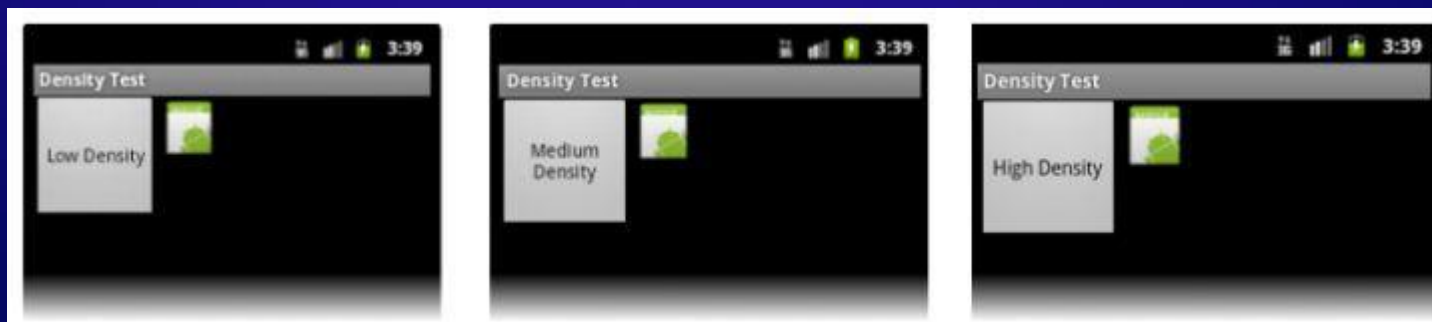
稳定

</> dp直接适配

px :

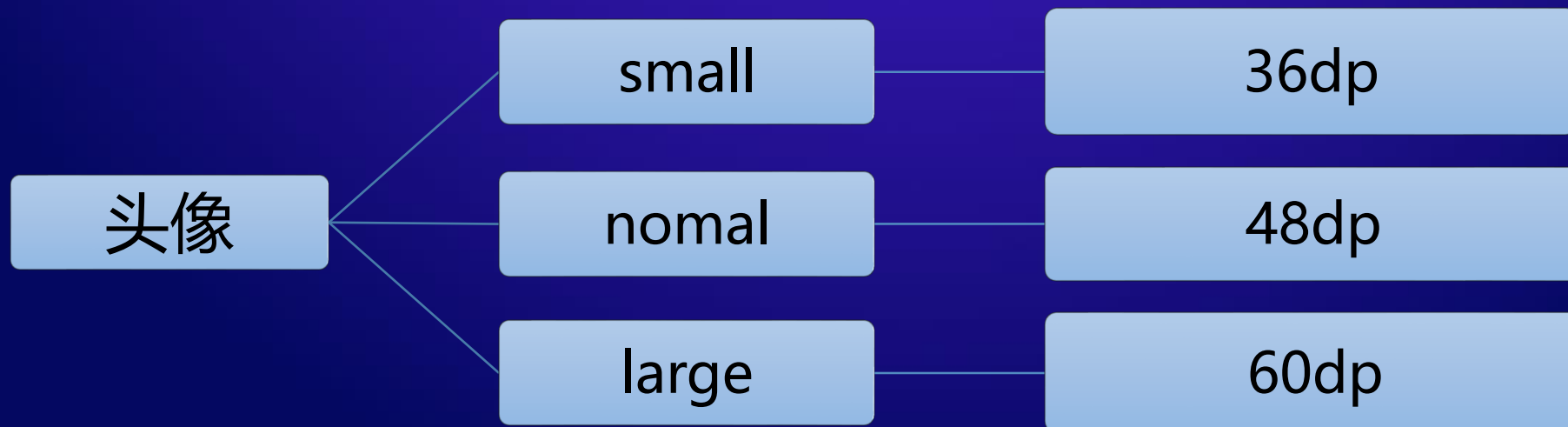


dp :



</> dp直接适配

为不同的控件设定固定的尺寸级别 特殊地方单独适配



</> dp直接适配的问题

Nexus S (480x800)



Nexus 4 (768x1280)



Pixel (1080x1920)



Pixel XL (1440x2560)



咦，你这UI好像在这个手机上不太对啊？

哪里不太对？

头像好像有点大，边距也有点大

又不是不能用！

不是吧 我看差不多啊

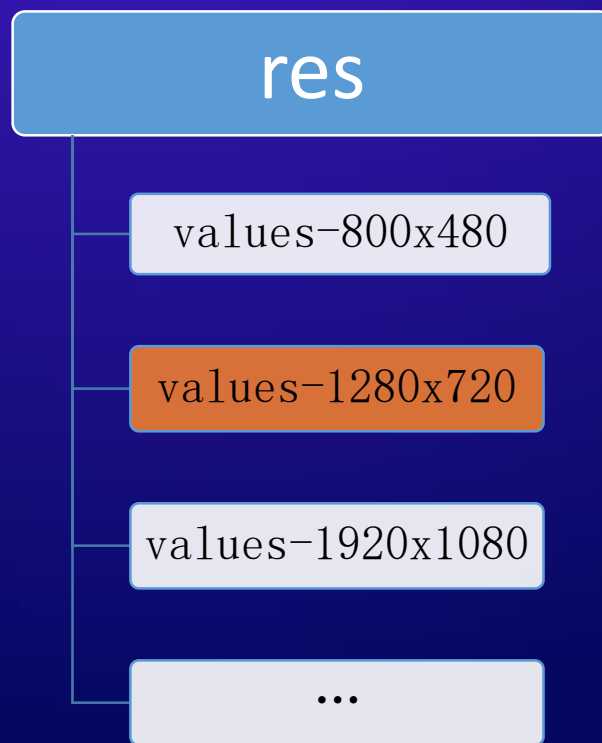
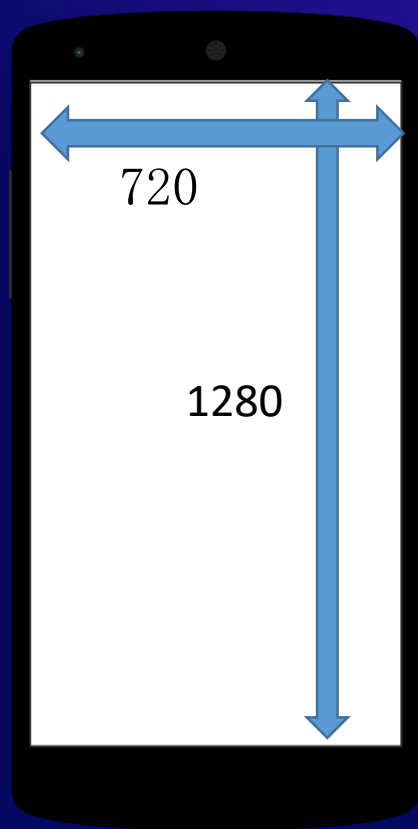
还是有差别的，你对比看，仔细看....

。 。 。 。 。 。

- ◆ 比较稳定，基本能实现适配效果，
- ◆ 开发起来不够高效
- ◆ 还容易引发设计和开发的冲突

</> 宽高限定符适配

通过生成values<h,w>文件夹以及dimens文件，系统根据配置信息加载该文件夹下的dimens尺寸信息





◆ 能实现高效稳定的适配

◆ 屏幕像素尺寸过多

◆ 容错效果不佳

</> 小结

dp直接适配

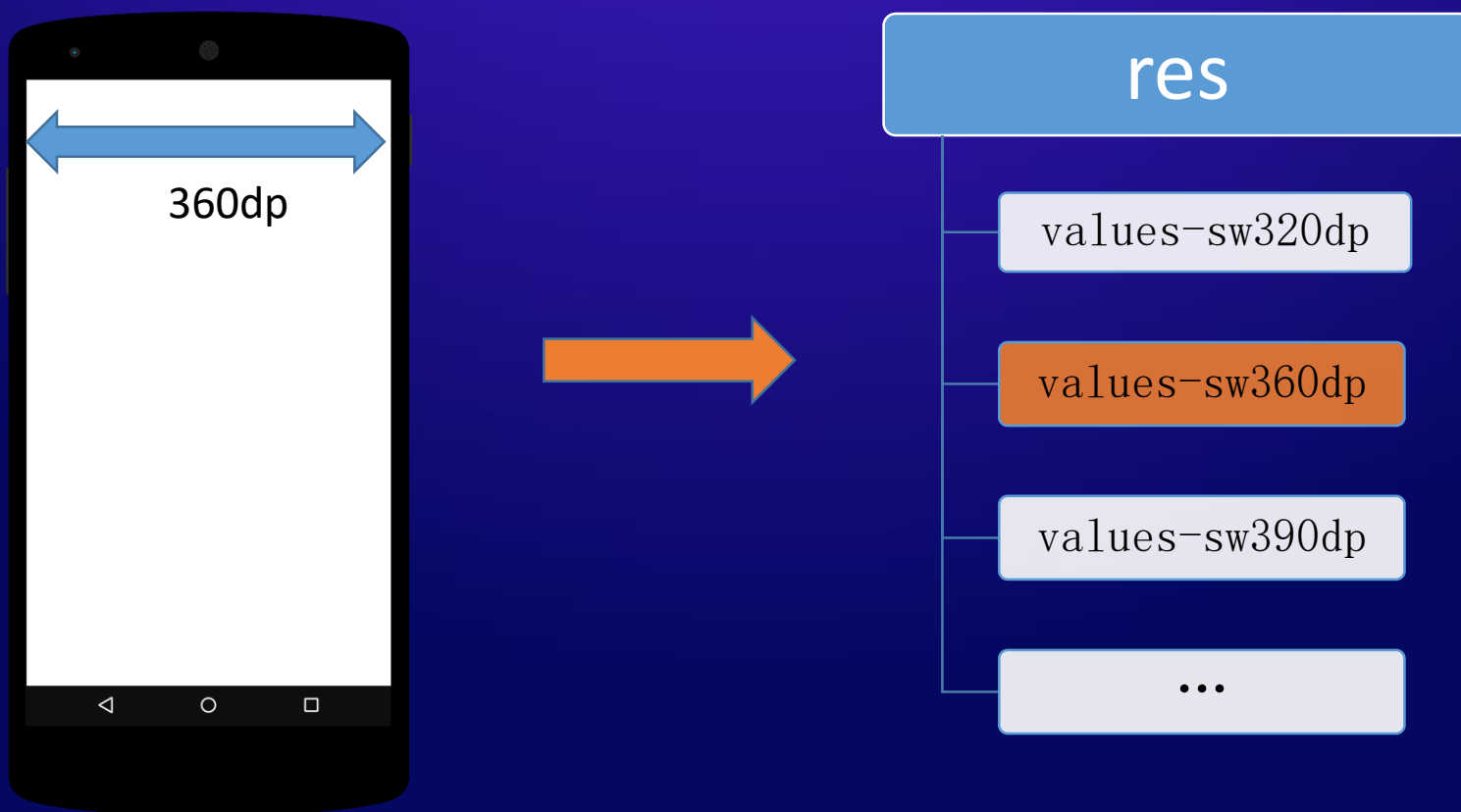
- ◆ 适配效果比较稳定，但是不够高效

宽高限定符适配

- ◆ 适配效果稳定且高效
- ◆ 屏幕尺寸过多，适配覆盖难度大
- ◆ 容错性不佳

</> smallestWidth适配

Android会识别**屏幕最小宽度**的dp值。然后尝试加载对应的文件夹下的资源文件。



</> 手机最小宽度如何计算

	dpi	density	分辨率	SmallestScreenWidth
mdpi	160	1	320x480	320dp
hdpi	240	1.5	480x800	320dp
xxldpi	320	2	720x1280	360dp
dpi_420	420	2.625	1080x1920	411dp
dpi_440	440	2.75	1080x1920	392dp
xxhdpi	480	3	1080x1920	360dp

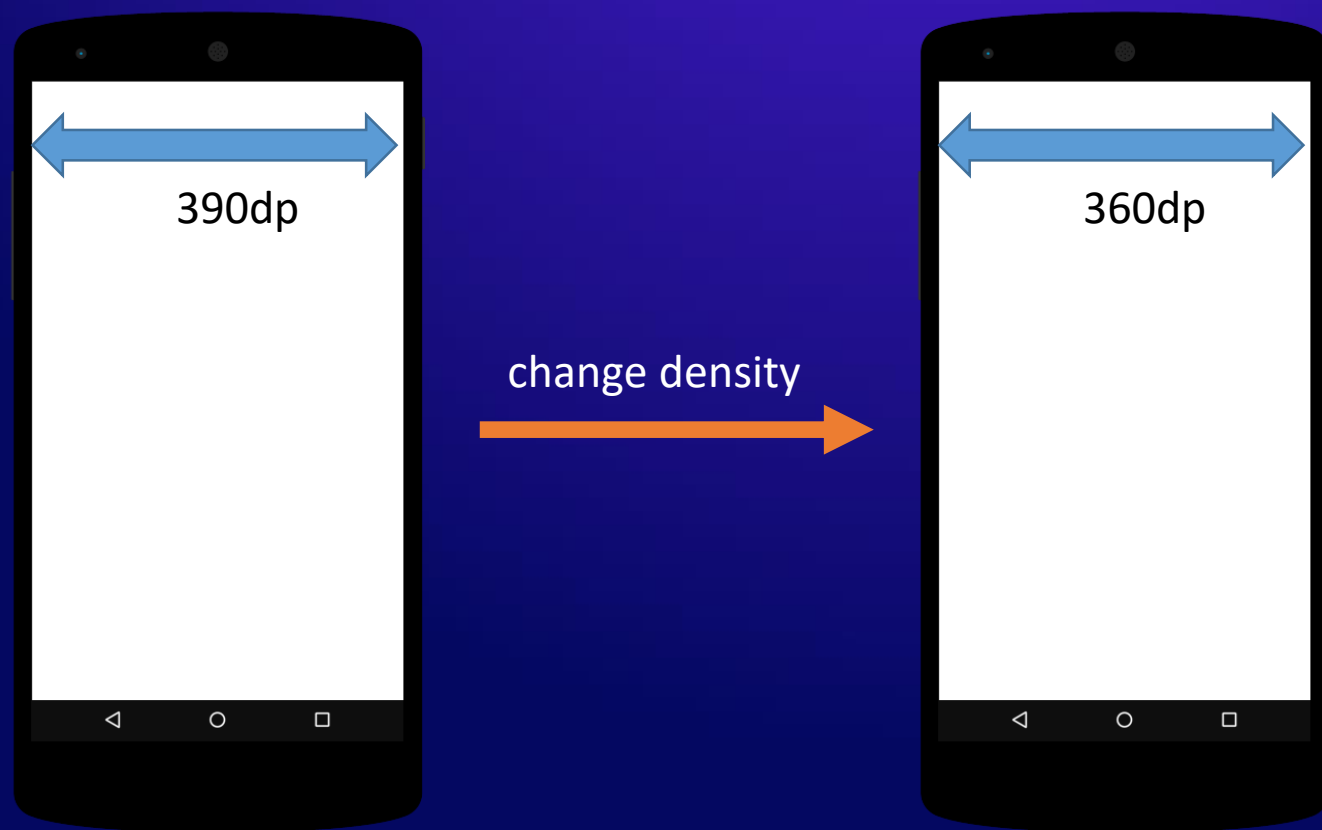
$SW(dp) = smallestScreenWidth_{px} / density$

屏幕最小宽度dp

屏幕最小宽度像素值


</> change density的适配方案

将设计稿宽度和手机屏幕宽度dp值设为一致，然后把所有不符合这个屏幕dp宽度的手机强制改为同一宽度



</> 如何修改手机的dp宽度

~~// 屏幕宽度dp值 = 屏幕宽度像素值 / density~~
int targetDensity = 屏幕宽度像素值 / 设计稿宽度;



~~// 获取APP屏幕配置~~
~~设计稿宽度值 = 屏幕宽度像素值 / density~~
DisplayMetrics dm = activity.getResources().getDisplayMetrics();



// 修改配置
dm.density = targetDensity; *// 把APP的density改为一致*
~~density = 屏幕宽度像素值 / 设计稿宽度值~~
dm.densityDpi = 160 * targetDensity;

</> 两个适配方案的比较

	sw适配	change density适配
是否高效稳定	是	是
实现难度	简单	简单
老项目引入成本	低	较高
是否有隐患	基本没有	可能有
Apk体积	有增加	无影响

怎样看待适配方案？

谢谢

</> 欢迎关注安卓巴士公众号



www.apkbus.com