

# What's New in Android

Google I/O 17 redux

吴晶 @Android笔记

# What's New?

- 手机支持画中画
- 颜色管理
- 支持多显示器
- 多媒体API更新
- WebView更新
- AnimatorSet API 更新
- 自动填表
- 字体资源
- 可下载字体
- 自动尺寸的 TextView
- findViewById()
- Java API 更新
- Kotlin
- 自适应图标
- 快捷方式和桌面小工具
- 通知中心
- 严格模式
- 多媒体文件访问
- 缓存数据
- 安全策略
- 辅助功能工具
- Art 更新
- 物理动画
- App架构组件
- 后台APP策略
- Android Studio 3.0
- Android Go
- Support lib v26
- Instant App
- . . .

# What's New?

- I/O 17 Android 视频 54 个
- What's New XXX 主题 13 个

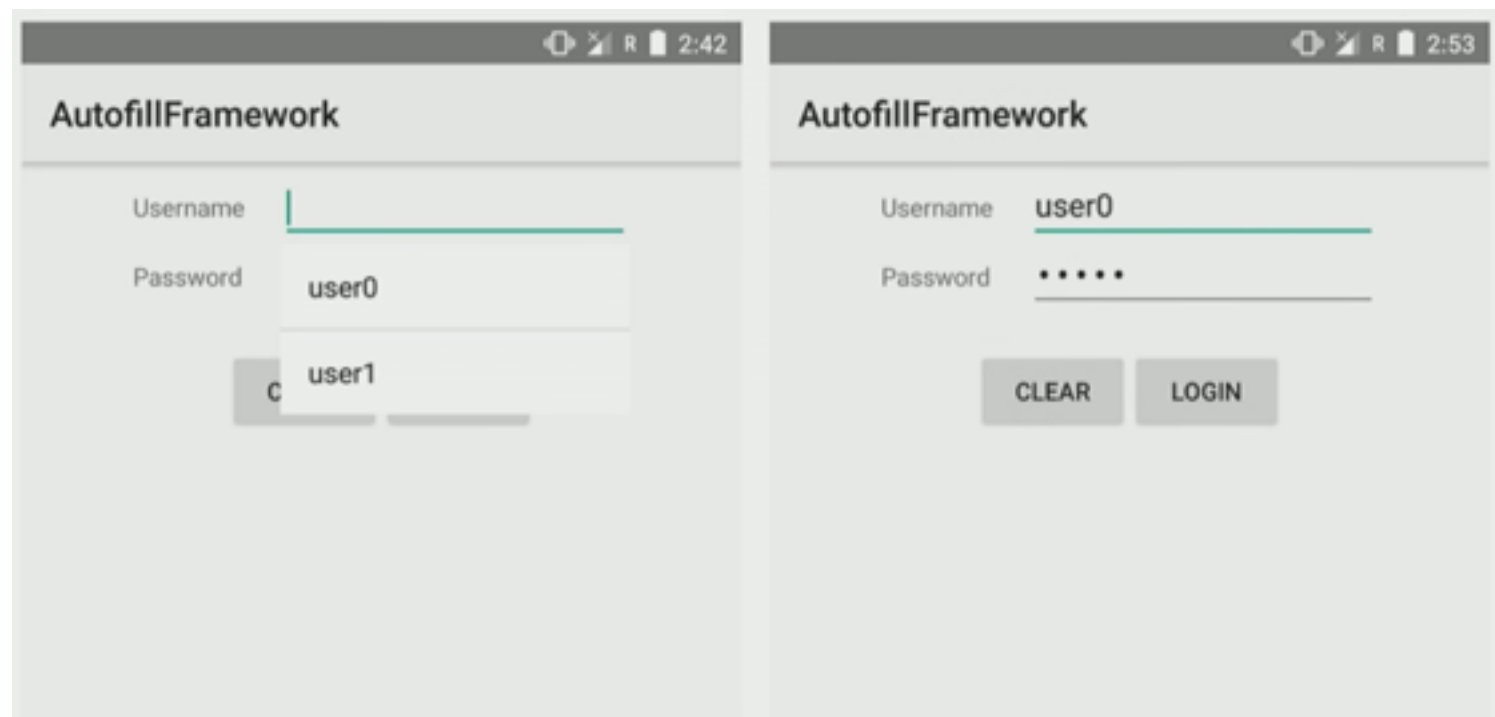
# 目录

- Android O
- Support library
- Android Studio 3.0

Android O

# 自动填表

- 减少输入
- 更加安全



# 自动填表

- 服务端
  - 保存表格数据
  - 密码管理工具
  - Google Play service
  - 与 Chrome 的数据互通
- 客户端
  - 没有或者少量工作

# 通知

- 频道 Channel
  - 解决的问题
    - 更加细粒度，不是一刀切
  - 特性
    - 系统设置统一管理
    - 基于发送通知
    - 每个 APP 有一个或者多个频道
- 角标通知



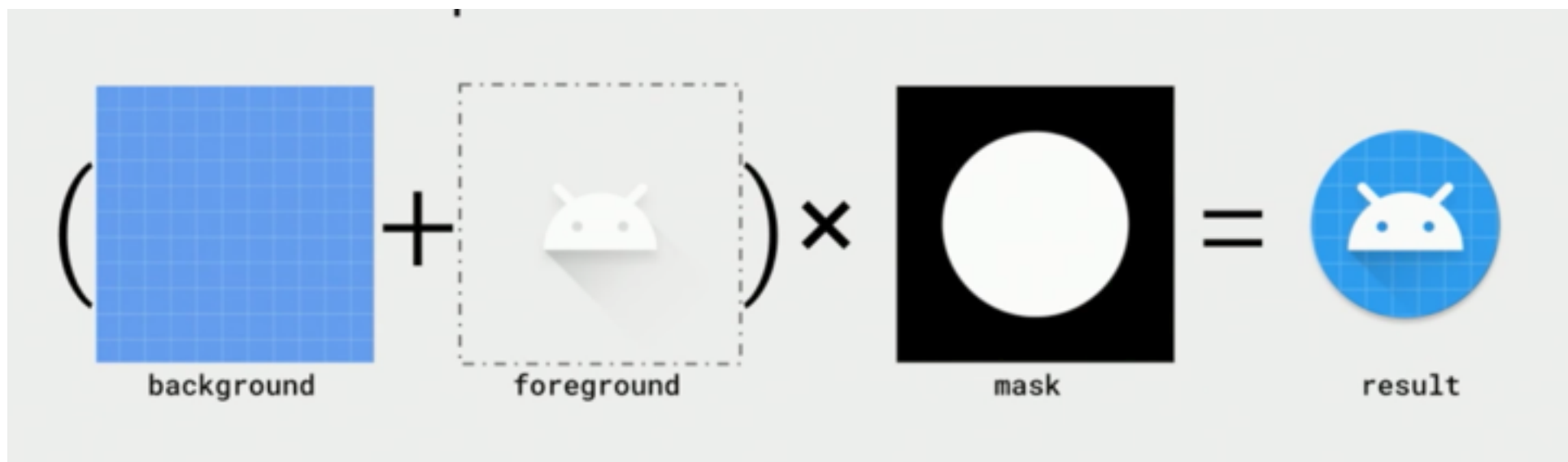


# 通知

- UI
  - 按照优先级顺序：Ongoing—> ‘P2P’ —> General —> BTW
  - 背景颜色
  - 炫酷动画
  - 推迟通知

# 自适应图标

- Adaptive Icon
  - 不同 OME 厂商，对 Icon 的处理不一致
  - 更加灵活



## AndroidManifest.xml

```
<manifest ...  
    <activity or application...  
        android:icon="@mipmap/ic_launcher">
```

No new manifest attribute defined

## Resource directory structure

```
res/  
    mipmap-[m|h|xh|xxh|xxxh]dpi/  
        ic_launcher.png  
        ic_foreground.png  
  
    mipmap-anydpi/  
        ic_launcher.xml
```

Same name xml file as the legacy png asset

```
ic_launcher.xml  
<adaptive-icon>  
    <background android:drawable="@color/ic_background"/>  
    <foreground android:drawable="@mipmap/ic_foreground"/>  
</adaptive-icon>
```

# 快捷方式和桌面小工具

- 快捷方式
  - INSTALL\_SHORTCUT 广播将不工作了
  - ShortcutInfo/ShortcutManager
- 桌面小工具
  - 以前是用户主动到 Launcher 中去找
  - AppWidgetProviderInfo/AppWidgetManager

# 后台应用

- 服务
  - 后台服务会被杀掉
  - `Context.startForegroundService()`
- 定位信息
  - 后台应用一个小时只能收到少量几次位置更新
  - WiFi 扫描也一样
- 广播
  - 在 Manifest 中不能注册接收非指向性广播（有一些例外）

# 隐私

- ANDROID\_ID
  - 每个用户每个 APP 不一样
  - 重新安装不变
- `net.hostname == null`

# findViewById()

```
public View findViewById(int id)
```

```
TextView text = (TextView) findViewById(R.id.text1);
```

# findViewById()

```
public <T extends View> T findViewById(int id)
```

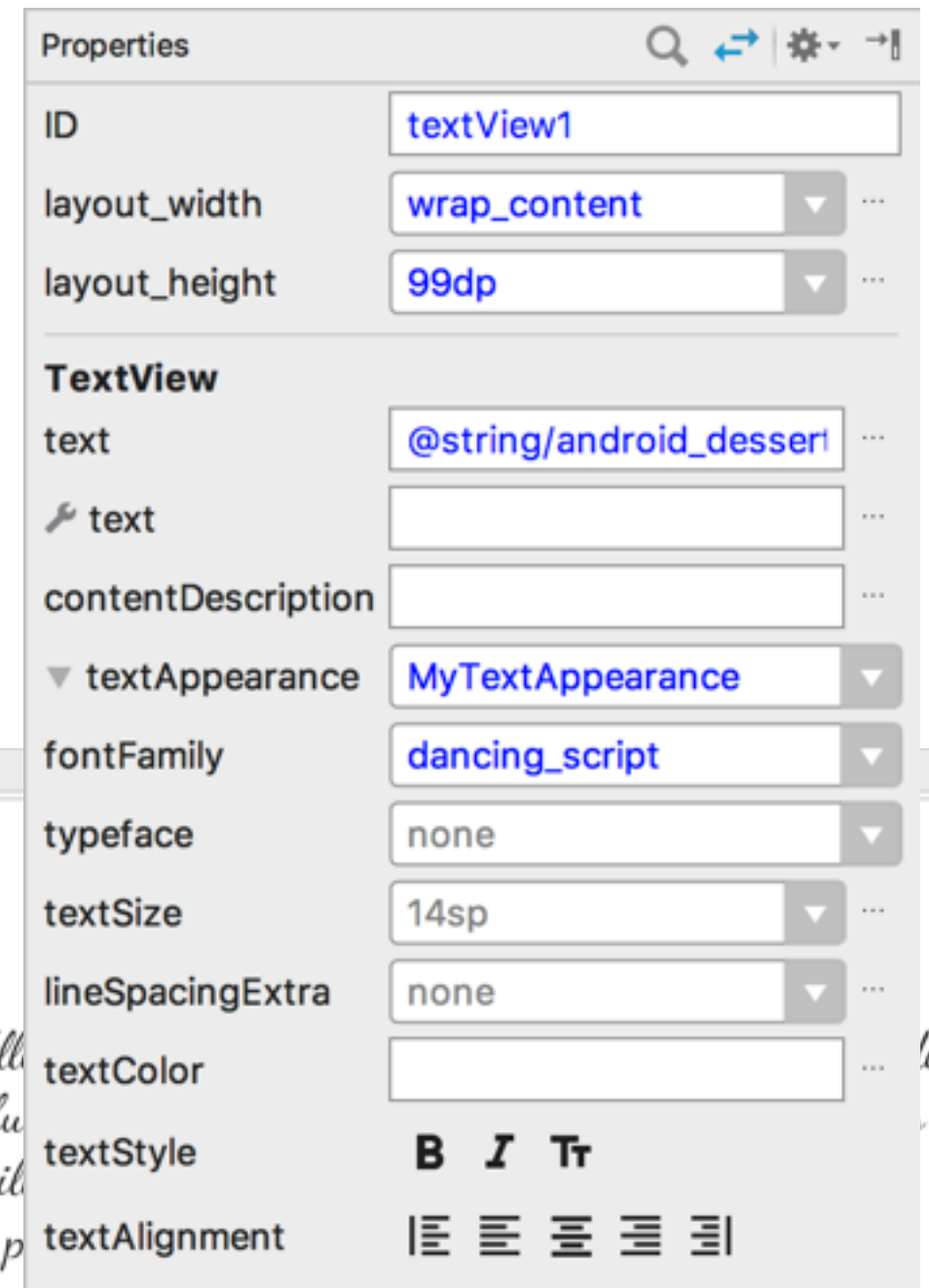
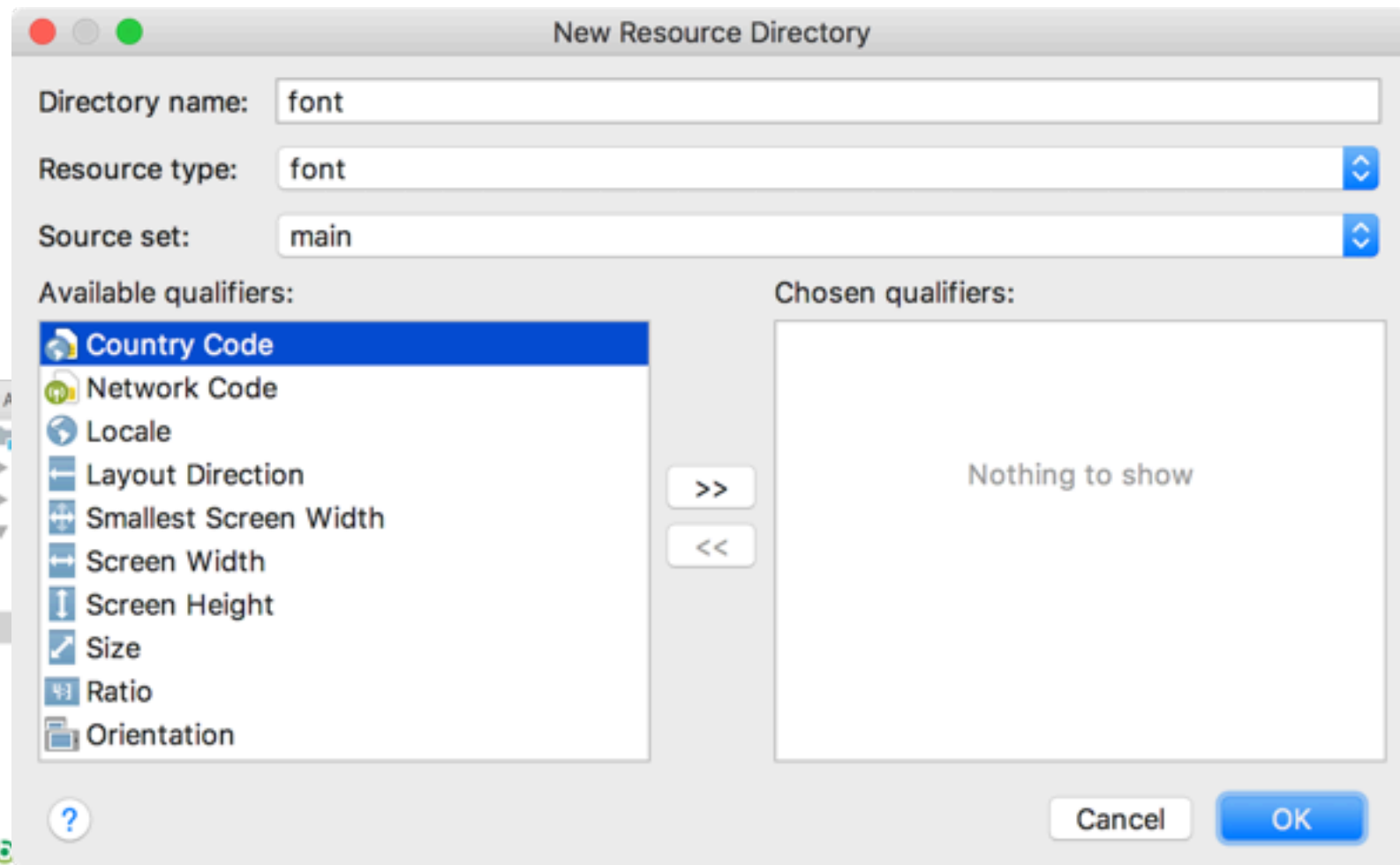
```
TextView text = findViewById(R.id.text1);
```



Support library

# 字体 Font

- 字体资源



# 字体 Font

- Font Family

```
<?xml version="1.0" encoding="utf-8"?>
<font-family xmlns:android="http://schemas.android.com/apk/res/android">
    <font
        android:fontStyle="normal"
        android:fontWeight="400"
        android:font="@font/lobster_regular" />
    <font
        android:fontStyle="italic"
        android:fontWeight="400"
        android:font="@font/lobster_italic" />
</font-family>
```

res/font/myvibur.xml

```
<?xml version="1.0" encoding="utf-8"?>
<font-family xmlns:android="http://schemas.android.com/apk/res/android"
              xmlns:app="http://schemas.android.com/apk/res-auto">
    <font android:font="@font/vibur"
          app:font="@font/vibur"
          android:fontWeight="400"
          android:fontStyle="normal"/>

    <font android:font="@font/vibur"
          app:font="@font/vibur"
          android:fontWeight="800"
          android:fontStyle="normal"/>
</font-family>
```

res/layout/mylayout.xml

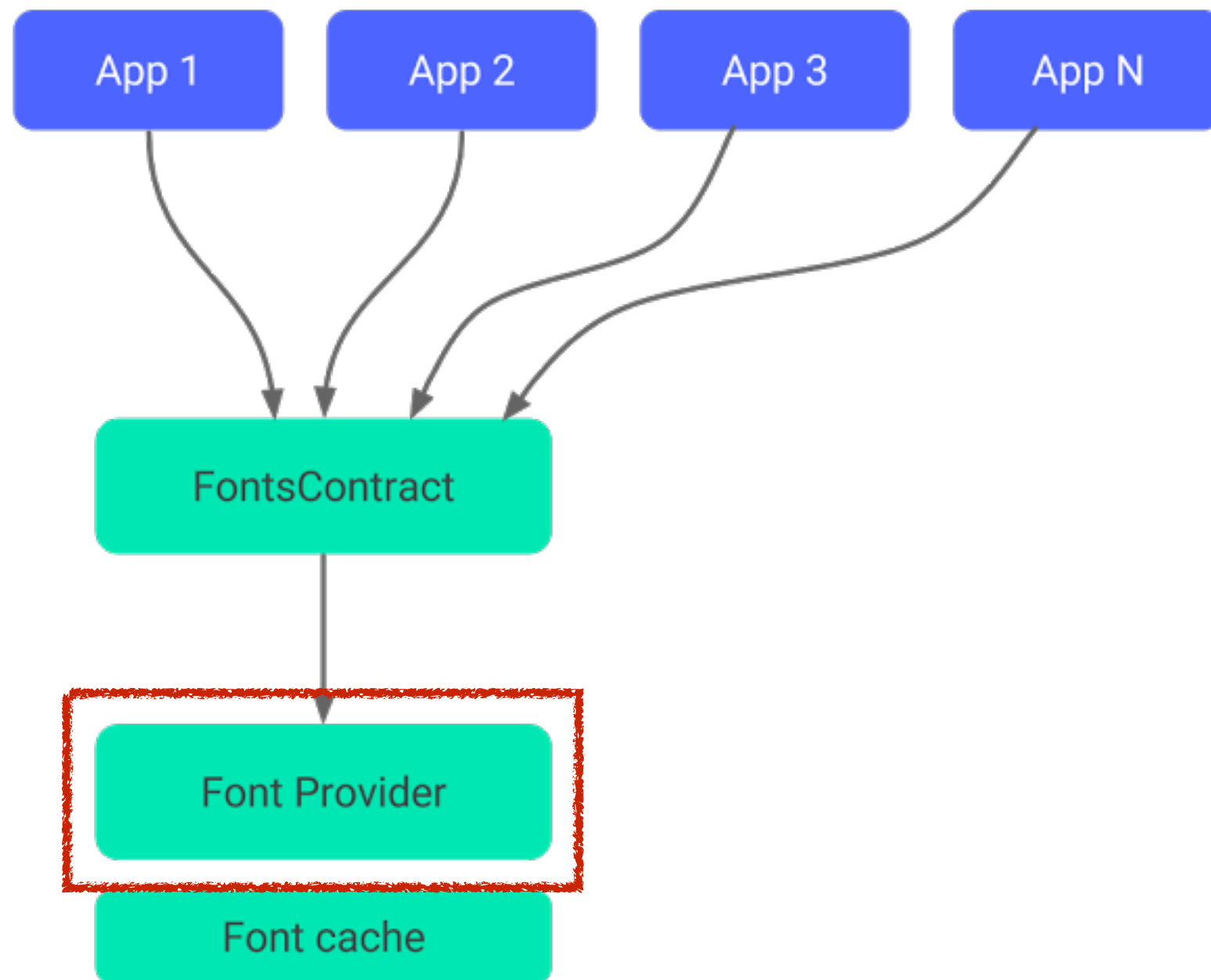
```
<TextView
    .....
    android:fontFamily="@font/myvibur"
    android:textStyle="bold"/>
```

```
res/values/styles.xml
<?xml version="1.0" encoding="utf-8"?>
<style name="customfontstyle">
    <item name="android:fontFamily">@font/vibur</item>
</style>
```

MyClass.java

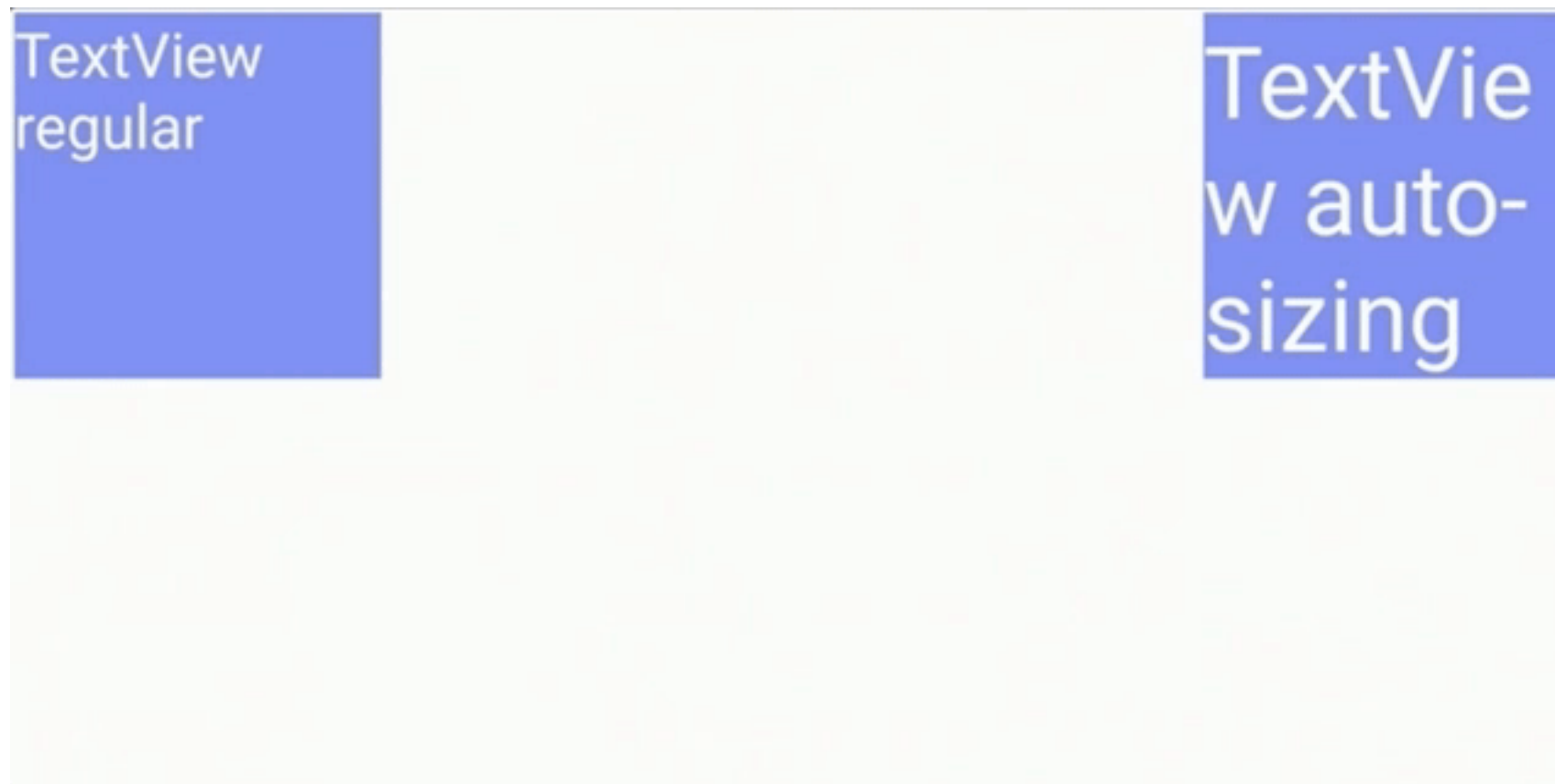
```
Typeface typeface = getResources().getFont(R.font.vibur);
// 或者
Typeface typeface = ResourcesCompat.getFont(context, R.font.vibur);
textView.setTypeface(typeface);
```

# Downloadable Fonts



```
compile "com.android.support:support-compat:26.0.0-beta1"
```

# 自动字体大小



```
<TextView
    android:id="@+id/auto_size_text"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:autoSizeTextType="uniform"
    android:text="Auto sizing TextView"
/>
```

```
TextViewCompat.setAutoSizeTextTypeWithDefaults(textView,
    TextViewCompat.AUTO_SIZE_TEXT_TYPE_UNIFORM);
```



控制粒度

```
<TextView
    android:id="@+id/auto_size_text"
    android:layout_width="match_parent"
    android:layout_height="100dp"
    app:autoSizeTextType="uniform"
    app:autoSizeMinTextSize="12sp"
    app:autoSizeMaxTextSize="100sp"
    app:autoSizeStepGranularity="2sp"
/>
```

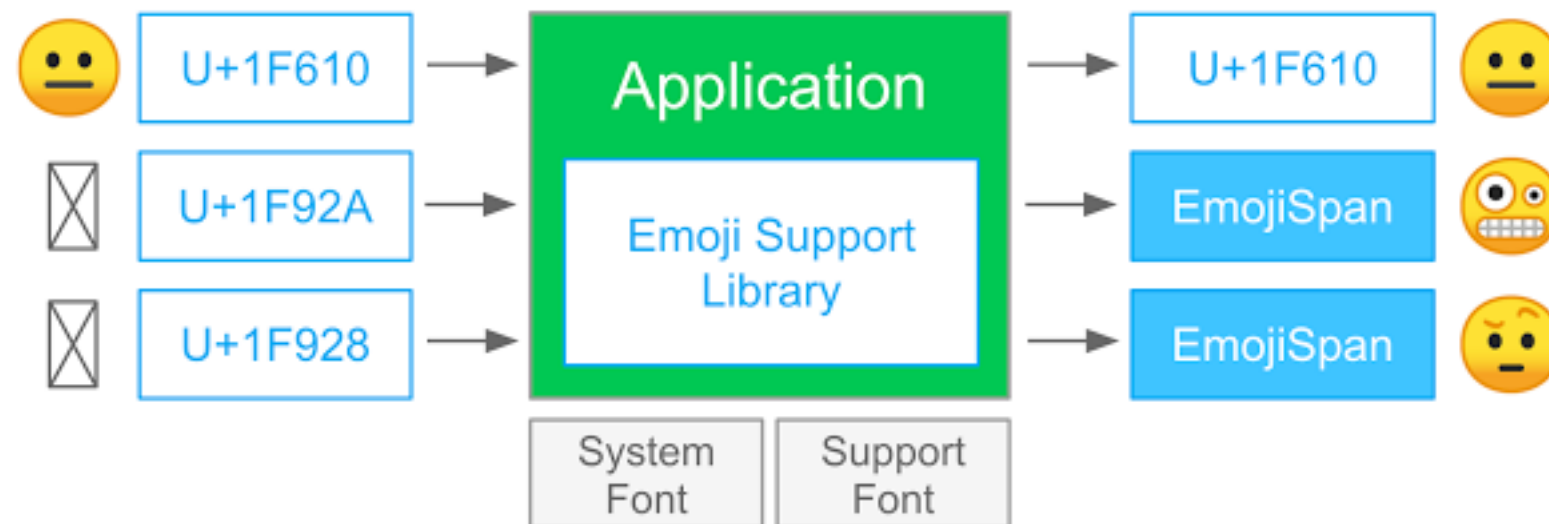
```
TextViewCompat.setAutoSizeTextTypeUniformWithConfiguration(
    textView, minSize, maxSize, granularity,
    TypedValue.COMPLEX_UNIT_SP);
```

预设大小

```
<TextView
    ...
    app:autoSizePresetSizes="@array/autosize_text_sizes"/>
```

# Emoji Compatibility

- 兼容最新的 Emoji 符号
- API 19+
- 可以直接打包进 APK — 7M 🤔



```
compile "com.android.support:support-emoji:26.0.0-beta1"
```

# 物理动画

- 基于速度，而不是时长
- 更接近物理世界
- 视觉更自然
- SpringAnimation/FlingAnimation

```
compile "com.android.support:support-dynamic-animation:26.0.0-beta1"
```

## Non Physics-based Animation



## Physics-based Animation



# 架构组件

- APP 组件生命周期管理
  - LifecycleObserver, LifecycleOwner
  - LiveData, ViewModel — Lifecycle Aware
- 数据库 DAO
  - Room

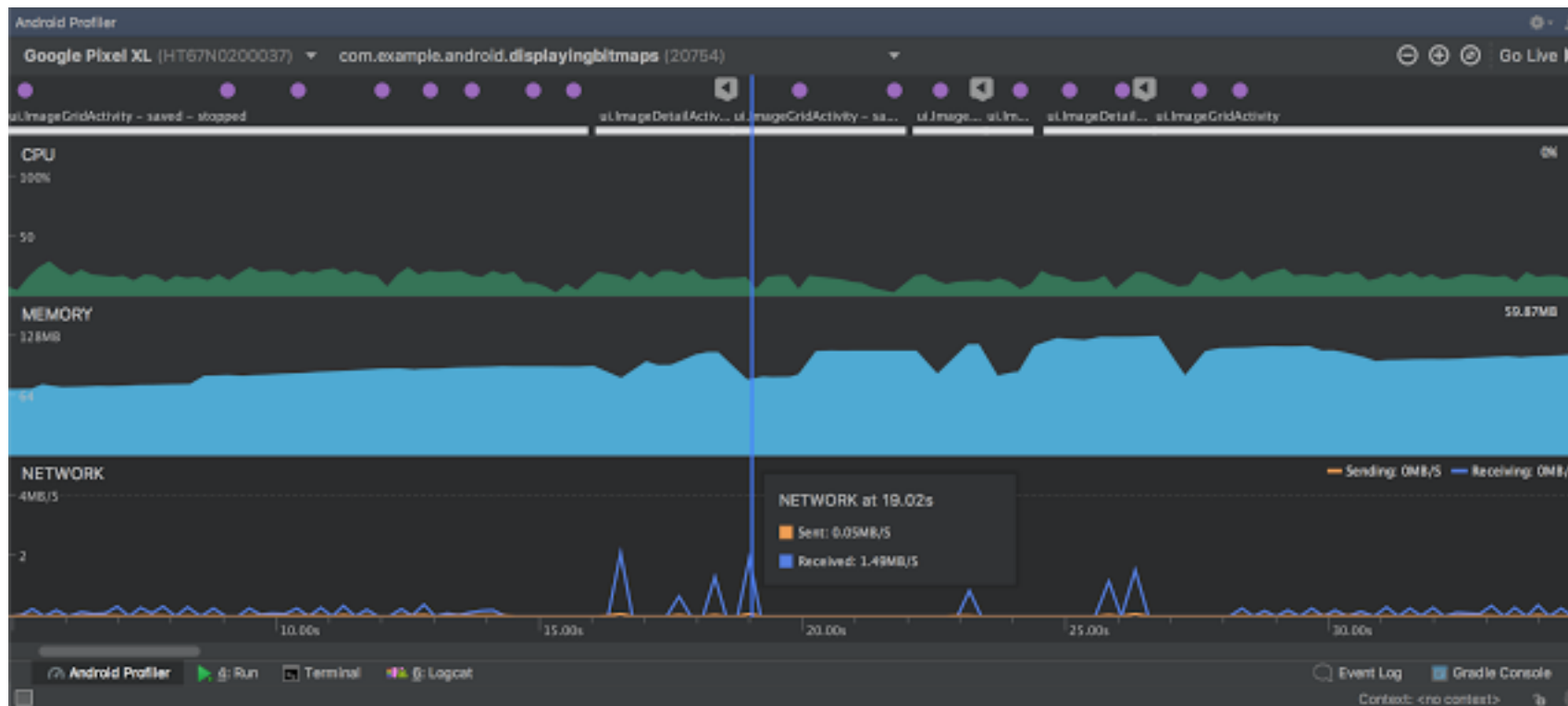
# Android Studio 3.0

# Android Studio 3.0

- 支持前面所说的新特性
  - Kotlin、Font & XML、自适应图标、可下载字体
- 调试 APK
- 设备上文件浏览器
- APK 分析

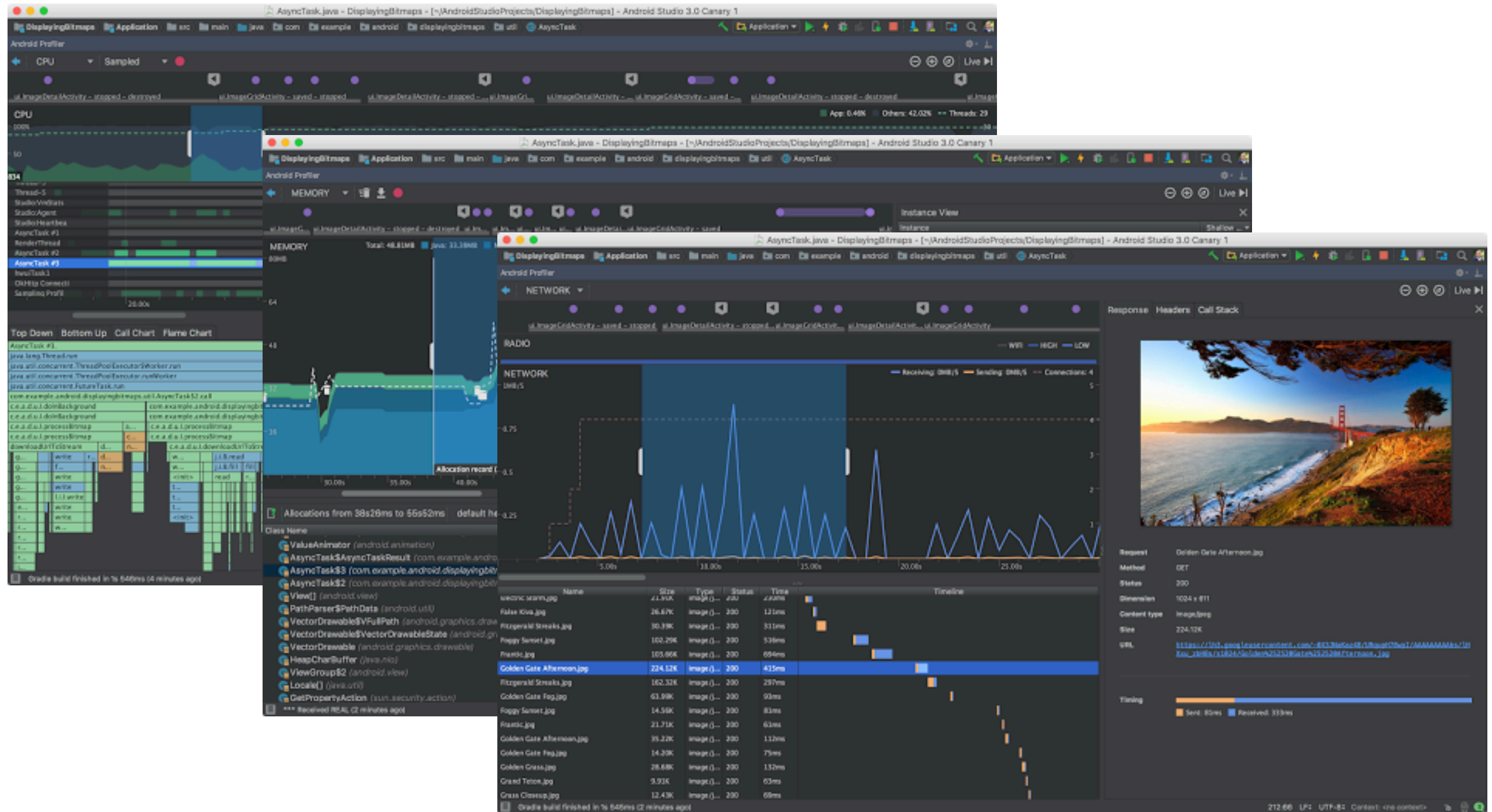
# Android Studio 3.0

- Android Profiler
- CPU、内存、网络分析





# Android Studio 3.0



谢谢！