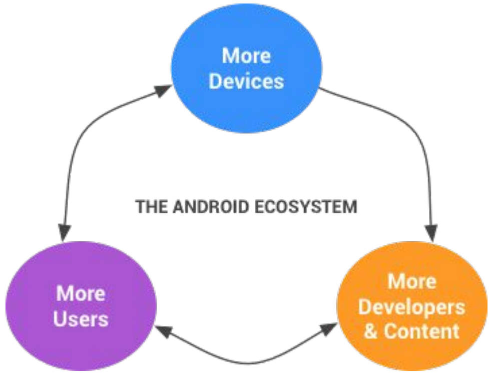
Aiya! My app broke again

Dec, 2016

Why is Compatibility important



Before Android: Many mobile platforms, bad for developers

With Android:

- Openness, Flexibility, and Diversity
- Creating a consistent experience but with choices
- Providing powerful tools to developers

Agenda

General Compatibility

Changes

Emerging Market

General Compatibility

Changes

Emerging Market

General Compatibility

Compatibility

Device compat VS App Compat

What do app developers need to pay attention to?

Device Feature

Platform Version

Screen Config

Different Languages

...

Device Features

If your app requires a device feature

```
<manifest>
    ...
    <uses-feature
    android:name="android.hardware.camera"
       android:required="true" />
    ...
</manifest>
```

Device Features

If your app optionally depend on a device feature

```
<uses-feature android:name="android.hardware.camera"
      android:required="false" />
```

```
PackageManager pm = getPackageManager();
if
(!pm.hasSystemFeature(PackageManager.FEATURE_CAMERA)
{
  disableCameraFeature();
}
```

Google Developer Day

Platform Version

If your app requires an API from specific version

```
<manifest
      ...
      <uses-sdk android:minSdkVersion="18"
      android:targetSdkVersion="24" />
      ...
</manifest>
```

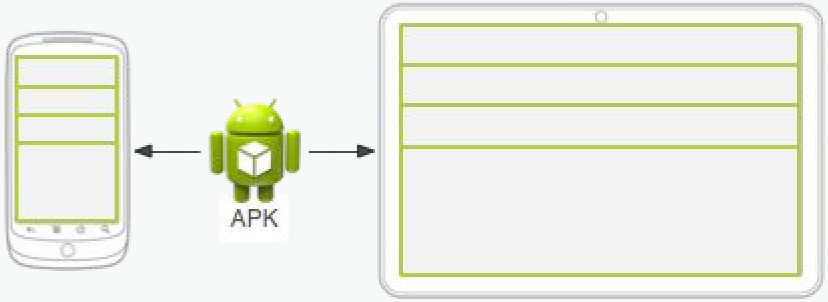
Platform Version

If your app optionally depend on an API

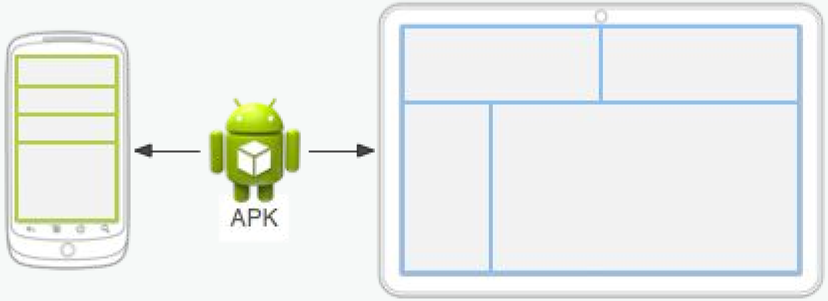
```
if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.LOLLIPOP) {
   doLollipopFeature();
}
```

Google Developer Day

Screen Configuration
Create Different Layouts



Two different devices, each using the default layout (the app provides no alternative layouts).



Two different devices, each using a different layout provided for different screen sizes.

Google Developer Day

Screen Configuration
Create Different Layouts

Specifying default and alternative layout

```
MyProject/
    res/
        layout/                 # default (portrait)
            main.xml
        layout-land/            # landscape
            main.xml
        layout-sw600dp/           # sw600dp (portrait)
            main.xml
        layout-sw600dp-land/    # sw600dp landscape
            main.xml
```

Reference the layout file as usual

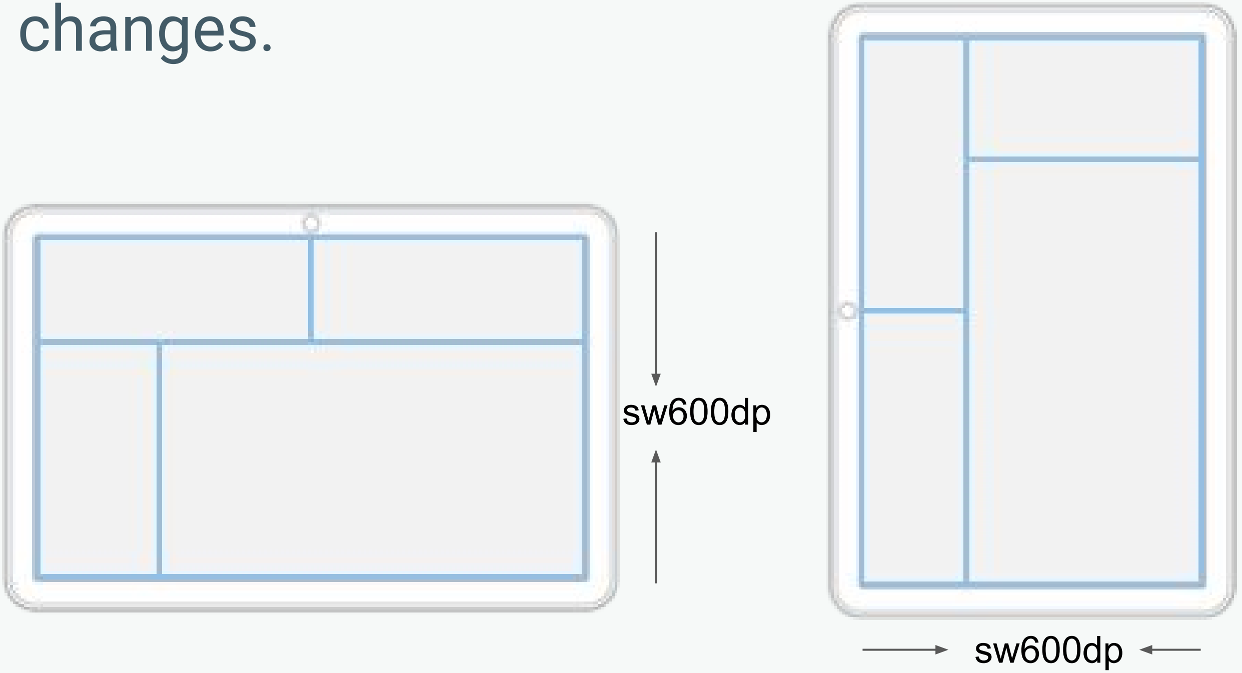
```
     setContentView(R.layout.main);
```

Google Developer Day

Screen Configuration
Smallest Width

sw<N>dp

The device's smallestWidth does not change when the screen's orientation changes.

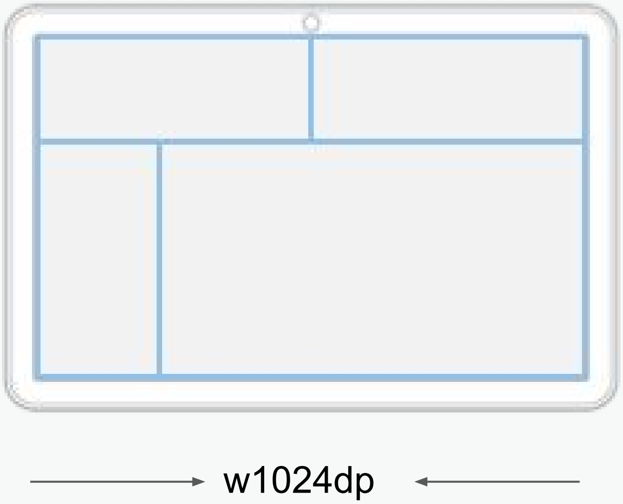


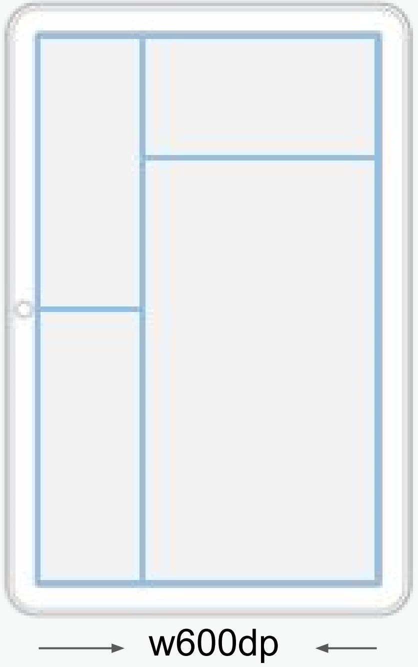
Google Developer Day

Screen Configuration
Available screen width

w<N>dp

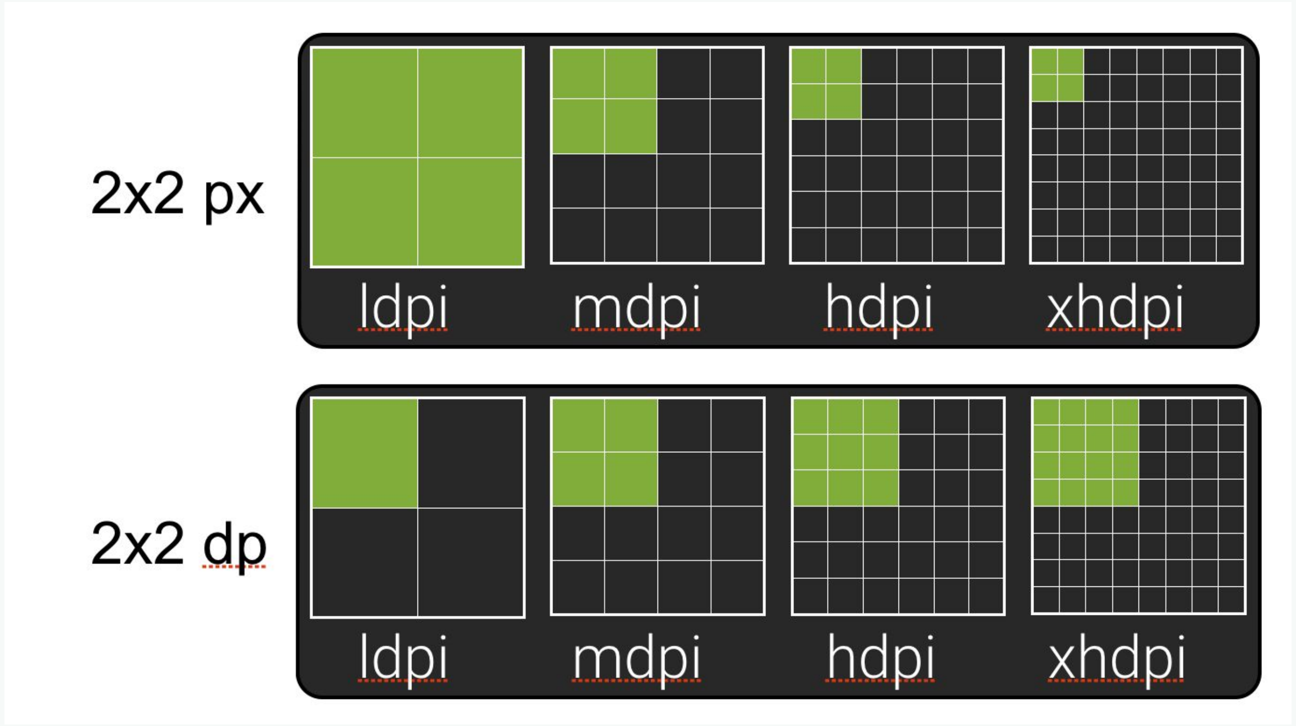
The device's width change when the screen's orientation changes.





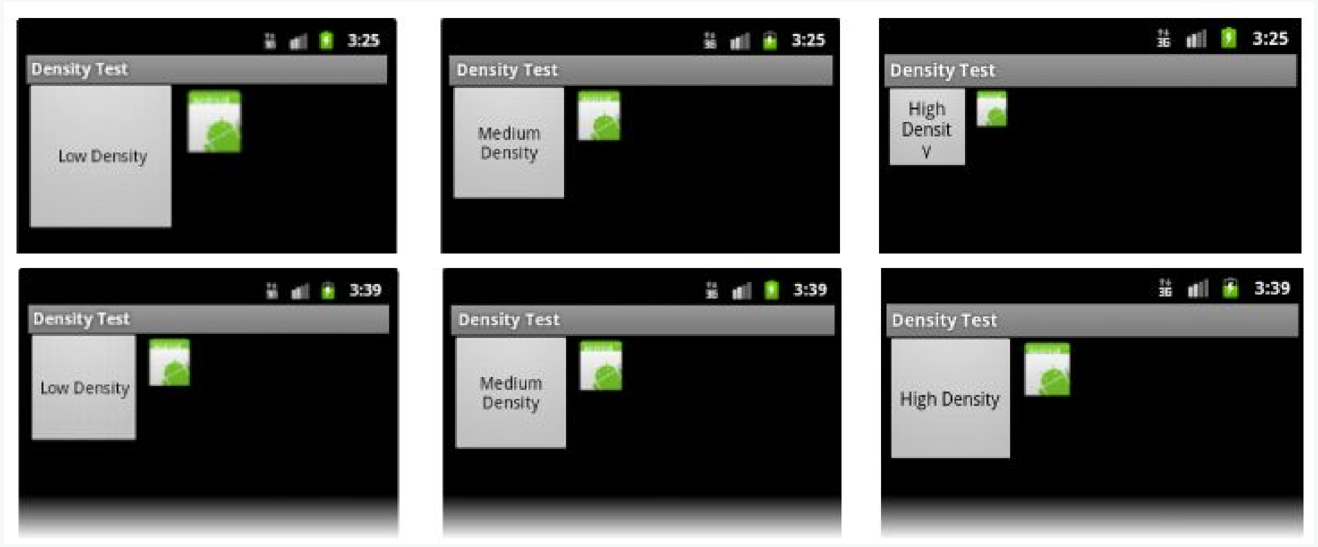
Google Developer Day

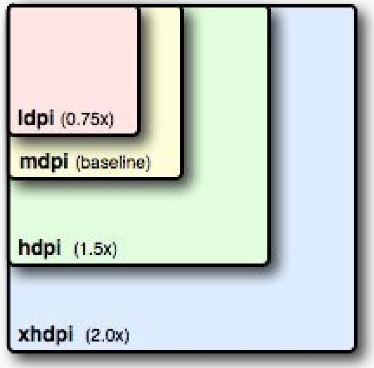
Screen configuration
px & dp



Google Developer Day

Screen configuration
Create Different Bitmaps





ldpi: 0.75
mdpi: 1.0 (baseline)
hdpi: 1.5
xhdpi: 2.0
xxhdpi: 3.0
xxxhdpi:4.0

Different Languages

Defining your resource

```
MyProject/
    res/
       values/
           strings.xml
       values-fr/
           strings.xml
```

```
English (default locale), /values/strings.xml:

    <string name="hello_world">Hello
World!</string>

French, /values-fr/strings.xml:

    <string name="hello_world">Bonjour le monde
!</string>
```

Google Developer Day

Plurals

Defining your resource

XML file saved at res/values/strings.xml:

```
    <plurals name="numberOfSongsAvailable">
        <item quantity="one">%d song found.</item>
        <item quantity="other">%d songs found.</item>
    </plurals>
```

XML file saved at res/values-pl/strings.xml:

```
    <plurals name="numberOfSongsAvailable">
        <item quantity="one">Znaleziono %d
piosenkę.</item>
        <item quantity="few">Znaleziono %d
piosenki.</item>
        <item quantity="other">Znaleziono %d
piosenek.</item>
    </plurals>
```

Google Developer Day

Plurals

Using your resource

```
int count = getNumberOfsongsAvailable();
Resources res = getResources();
String songsFound = res.getQuantityString
(R.plurals.numberOfSongsAvailable, count, 
count);
```

Google Developer Day

Plurals

Using your resource

```
int count = getNumberOfsongsAvailable();
Resources res = getResources();
String songsFound = res.getQuantityString
(R.plurals.numberOfSongsAvailable, 3, 3);
```

```
    <plurals name="numberOfSongsAvailable">
        <item quantity="one">%d song
found.</item>
        <item quantity="other">%d songs
found.</item>
    </plurals>
```

Google Developer Day

General Compatibility

Emerging Market

@hide API

DO NOT USE HIDDEN API

Further enforcement of SELinux policy will cause Apps to break

E.g
```
03-31 15:19:35.187   393   393 E SELinux : avc:  denied  { find } for service=window pid=27811 uid=10095 scontext=u:r:untrusted_app:s0:c512,c768 tcontext=u:object_r:window_service:s0 tclass=service_manager permissive=0
```

Google Developer Day

Battery Enhancement
Doze

What is it

Restricts apps' access to network and CPU-intensive services

Devices periodically resume normal operations for brief periods of time

Force the system to cycle through Doze modes

```
$ adb shell dumpsys battery unplug
$ adb shell dumpsys deviceidle step
```

Battery Enhancement
App Standby

What is it

the system disables network access and suspends syncs and jobs for the apps it deems idle

Force the app into App Standby mode

```
$ adb shell dumpsys battery unplug
$ adb shell am set-inactive <packageName> true
```

Simulate waking your app

```
$ adb shell am set-inactive <packageName> false
$ adb shell am get-inactive <packageName>
```

Battery Enhancement

More details at ...

Android battery and memory optimizations

Google Developer Day

APK Signature Scheme v2

What is it

a new app-signing scheme

offers faster app install times

more protection against unauthorized alterations to APK files.

Proper Sequence

1. Run Zipalign
2. Sign APK

Disable V2 Signing

```
v2SigningEnabled false
```

Changes for NDK developers

Private API

What changed

Native libraries must use only public API

Must not link against non-NDK platform libraries

Enforced since

API 24

Potential Problem

Dynamic linker will not load private libraries, preventing the application from loading

Resolution

Rewrite your native code to rely only on public API

Missing Section Headers

What changed

Each ELF file has additional information contained in the section headers

These headers must be present now

Enforced since

API 24

Potential Problem

Fail dynamic linker sanity checking

Dynamic linker will not load those that failed

Resolution

Remove the extra steps from your build that strip section headers

Text Relocations

What changed

Shared objects must not contain text relocations

The code must be loaded as is and must not be modified

Enforced since

API 23

Potential Problem

Dynamic linker will refuse to load code with text relocations

```
$ readelf --dynamic libTextRel.so | grep TEXTREL
 0x00000016 (TEXTREL)                    0x0
```

Resolution

Rewrite assembler to be position independent to ensure no text relocations are necessary

Google Developer Day

Invalid DT_NEEDED Entries

What changed

The runtime linker will honor the DT_NEEDED exactly as is

The runtime linker will not ignore the full path

Enforced since

API 23

Potential Problem

Dynamic linker won't be able to load the library if it is not present in that exact location on the device

Resolution

Make sure all required libraries are referenced by SONAME only

Missing SONAME

What changed

Each ELF shared object (“native library”) must have a SONAME (Shared Object Name) attribute

Enforced since

API 23

Potential Problem

Namespace conflicts may lead to the wrong library being loaded at runtime

Resolution

Ensure you’re using the current NDK

From Dalvik to ART

Android runtime

What is it

Used by applications and some system services on Android

ART and Dalvik are compatible runtimes running Dex bytecode

Features

Ahead-of-time (AOT) compilation

Improved garbage collection

Improved Development and debugging

Garbage Collection

What changed

Non-compacting garbage collector ->

Compacting garbage collector

What should you do

Avoid using techniques that are incompatible with compacting GC

Use CheckJNI to detect and report errors

Error handling

What changed

ART's JNI throws errors in a number of cases where Dalvik doesn’t.

What should you do

Test with CheckJNI mode.

Handle exceptions

Preventing Stack Size Issues

What changed

Dalvik had separate stacks for native and Java code

ART has a unified stack

What should you do

ART Thread stack size should be approximately the same as for Dalvik

If you explicitly set stack sizes, you may need to revisit those values for apps running in ART.

Java: Thread Constructor

```
C/C++:
    pthread_attr_setstack()
    pthread_attr_setstacksize()
```

Google Developer Day

Object model changes

What changed

Dalvik incorrectly allowed subclasses to override package-private methods.

ART issues a warning in such cases

```
Before Android 4.1, method void com.foo.Bar.quux()
would have incorrectly overridden the package-private method in
com.quux.Quux
```

What should you do

If you intend to override a class's method in a different package, declare the method as public or protected

Use an updated Mockito version when testing with ART.

Fixing AOT Compilation Issues

What changed

ART does tighter bytecode verification at install time than Dalvik does.

Code produced by the Android build tools should be fine.

What should you do

Getting the latest versions of your tools and regenerating the DEX files

General Compatibility

Changes

Emerging Market

Emerging market

A different challenge

Internet is slow and expensive

Higher usage - people use it as primary device

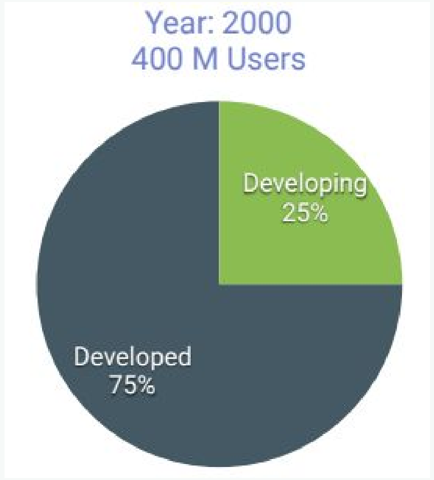
Smaller device storage

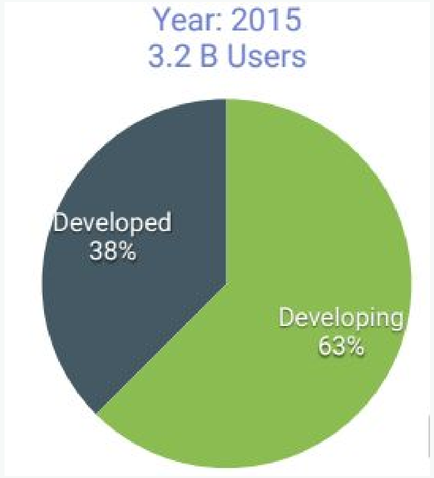
Smaller batteries

Longer app update cycles

Emerging market

Does your app work for Emerging market?

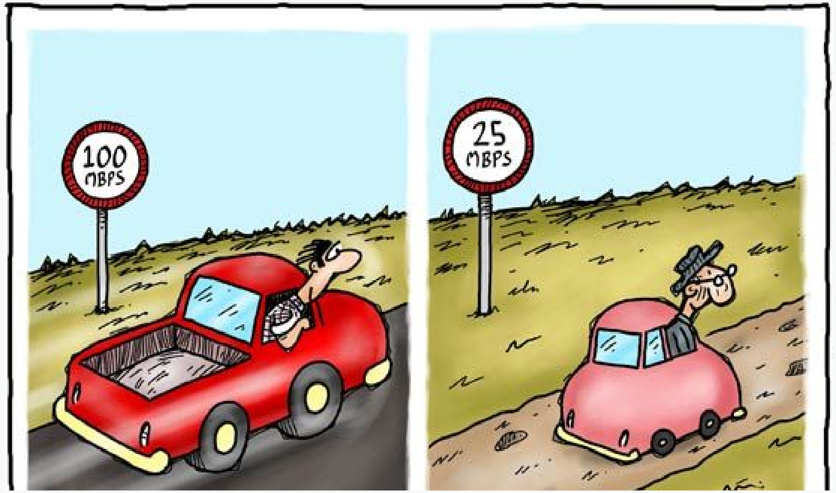




http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf

Mobile Network

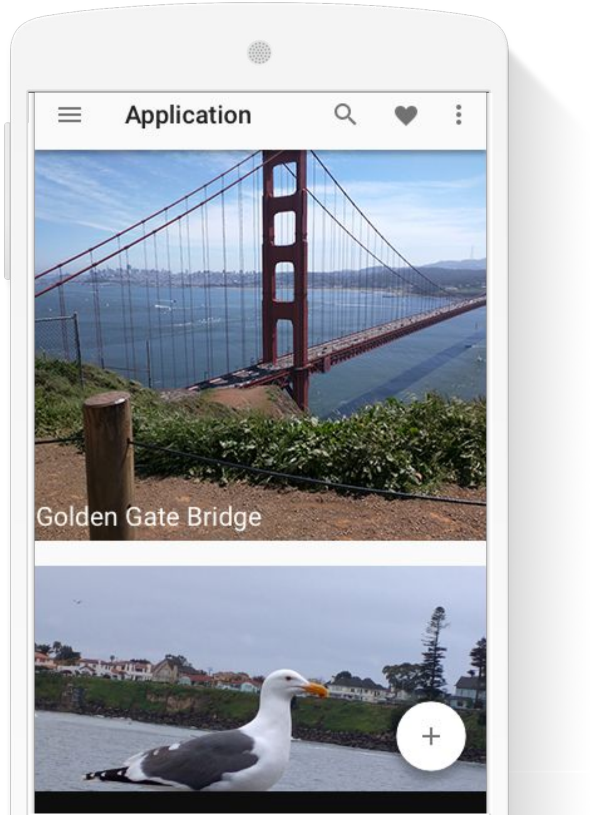
- Networks not as fast as operators advertise
- Dont assume network quality based on network type
- Most users are still on 2G speeds
- Always test in airplane mode



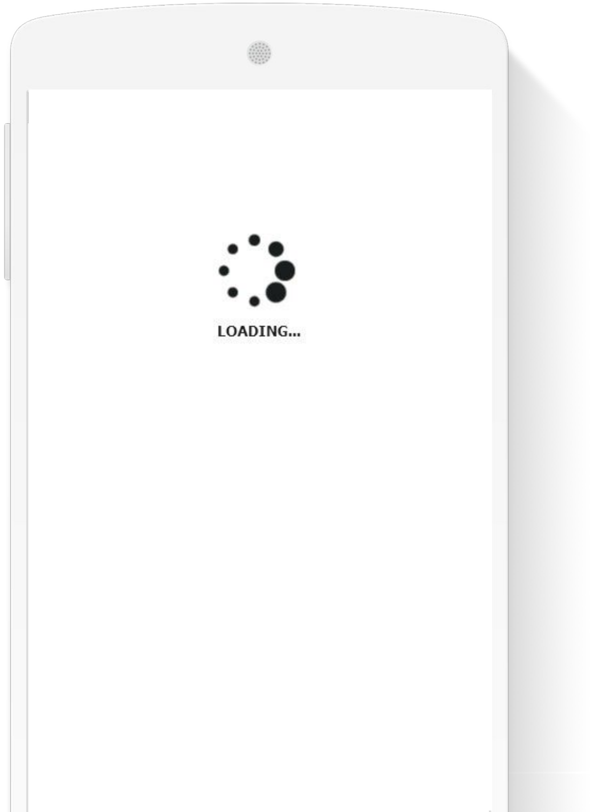
Google Developer Day

Content Fetching

What you see



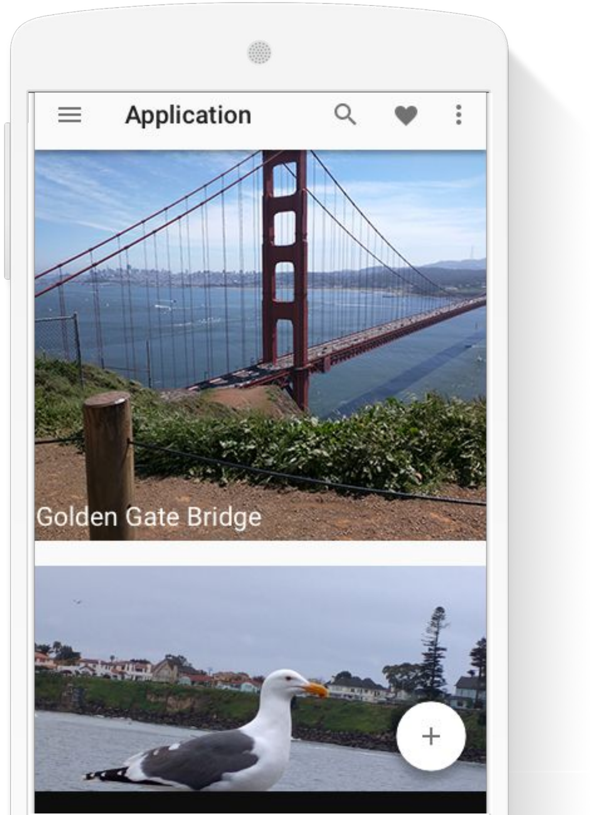
What your users see

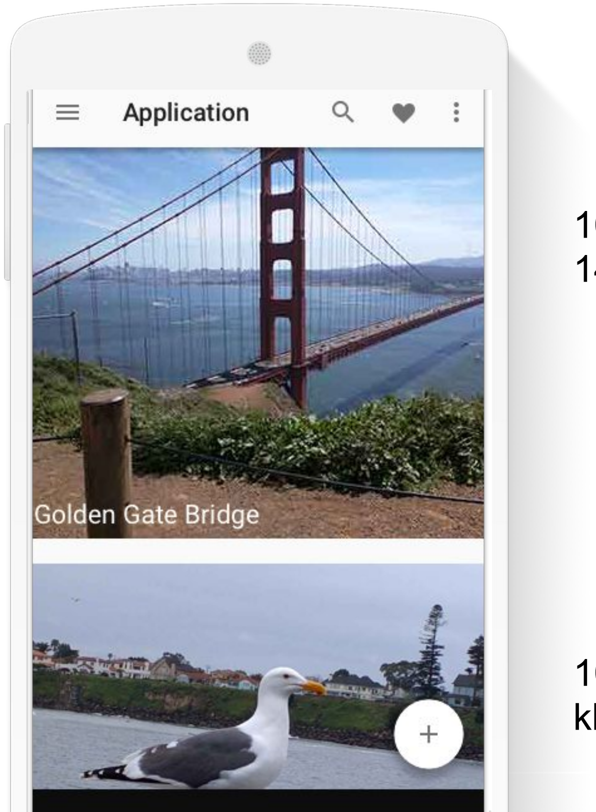


Adaptive Content Fetching

100% jpeg
146kb

100% jpeg
477 kb



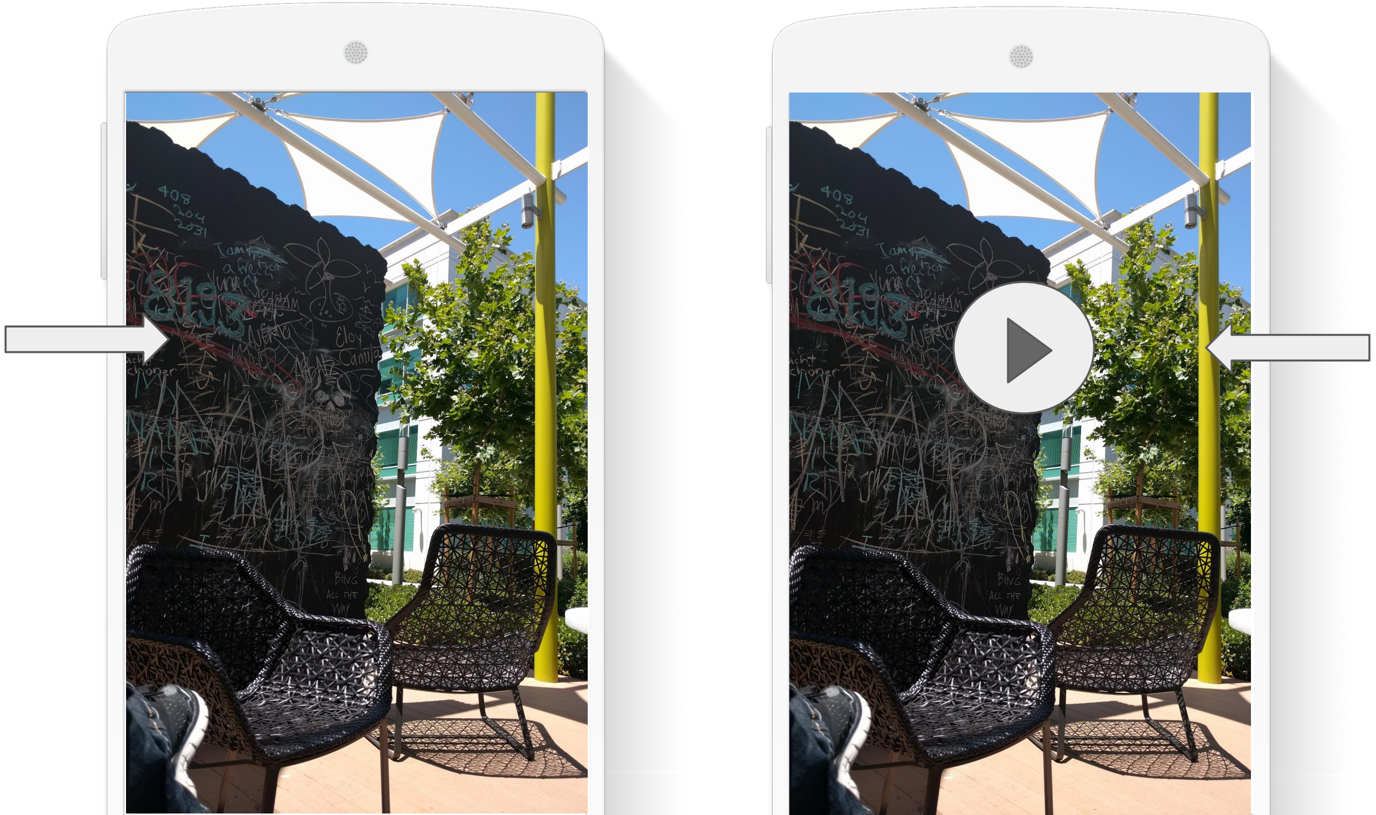


10% jpeg
14kb

10% jpeg 44
kb

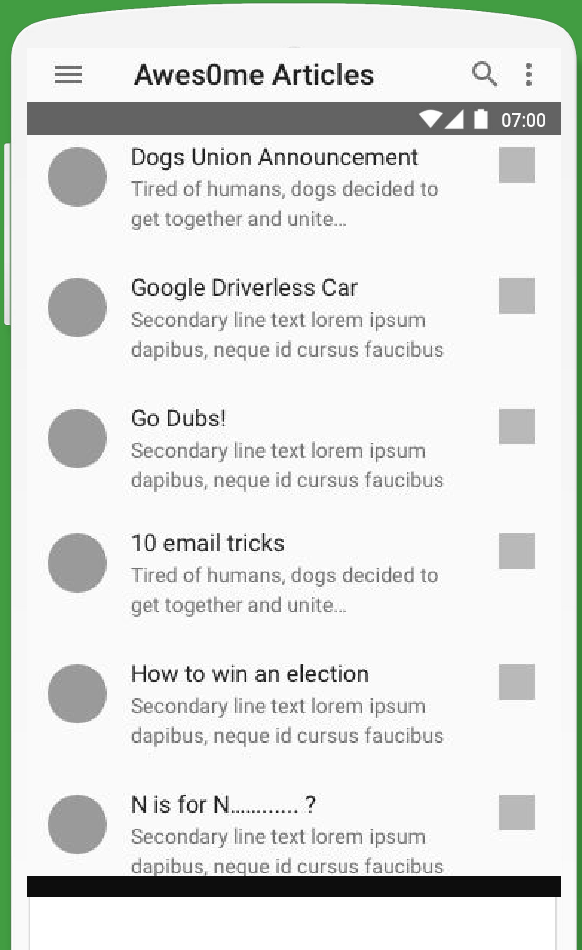
Adaptive Behavior

Network Fast?
Auto play



Network slow?
Tap to play

Prefetch


Awes0me Articles

07:00

Dogs Union Announcement
Tired of humans, dogs decided to get together and unite...

Google Driverless Car
Secondary line text lorem ipsum dapibus, neque id cursus faucibus

Go Dubs!
Secondary line text lorem ipsum dapibus, neque id cursus faucibus

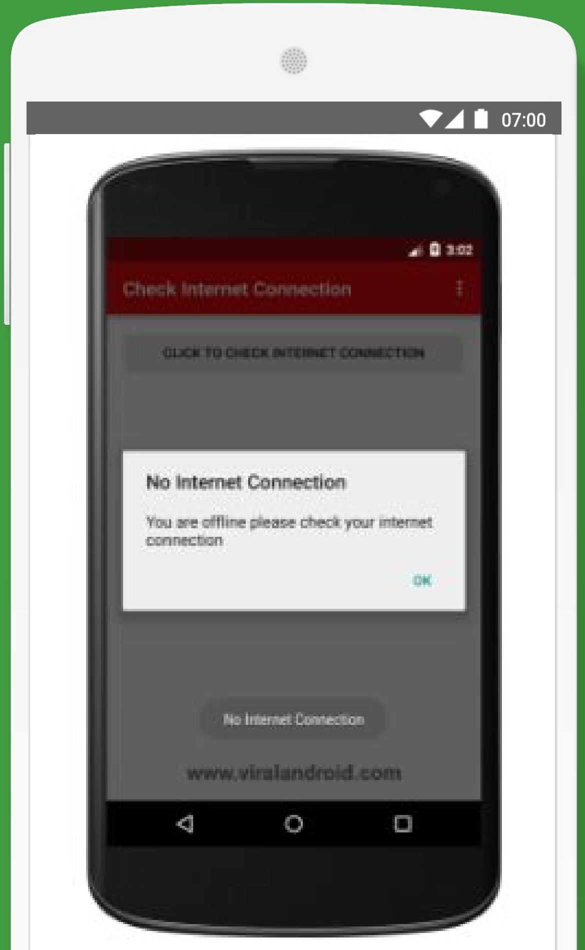
10 email tricks
Tired of humans, dogs decided to get together and unite...

How to win an election
Secondary line text lorem ipsum dapibus, neque id cursus faucibus

N is for N........... ?
Secondary line text lorem ipsum dapibus, neque id cursus faucibus

Google Developer Day

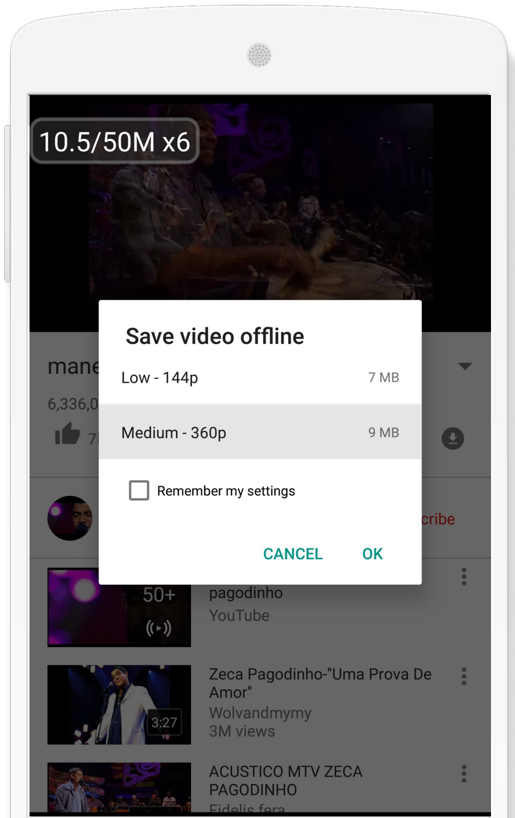
No Internet Connection



No Internet Connection

You are offline please check your internet connection

Offline Mode



10.5/50M x6

Save video offline

Low - 144p 7 MB

Medium - 360p 9 MB

Remember my settings

CANCEL OK

mane

6,336,0

cribe

50+

pagodinho

YouTube

Zeca Pagodinho-"Uma Prova De Amor"

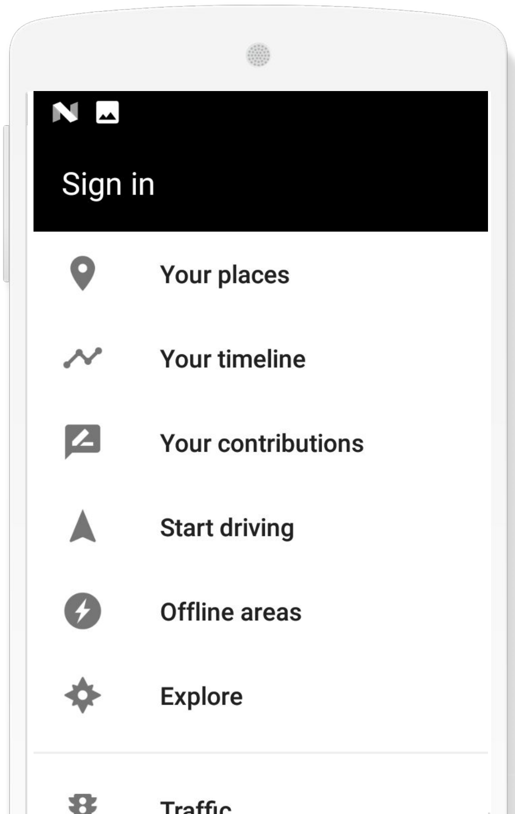
Wolvandmymy

3M views

3:27

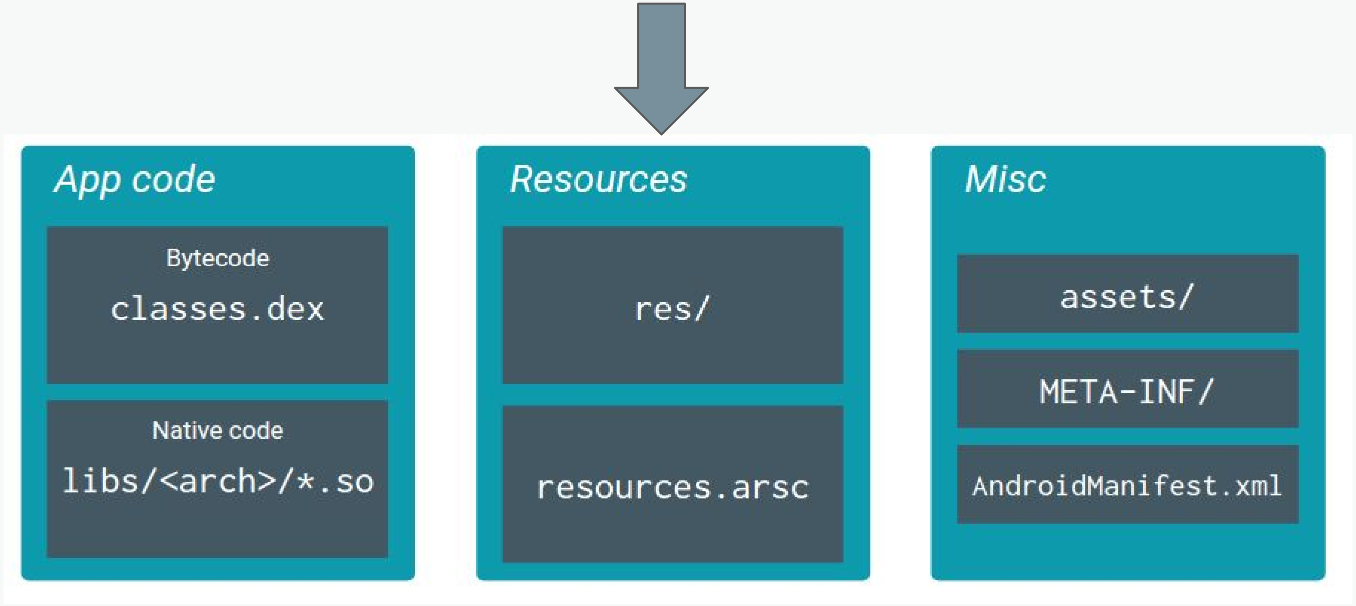
ACUSTICO MTV ZECA PAGODINHO

Fidelis fera



How to keep APK size down

Optimizing resources



Zopflify your APK

DO NOT USE
Zopfli compression on APK

Certain Android 5.0.1 devices might have problem reading Zopfli-compressed APKs and can even crash your apps.

Zopflify your PNGs

Pre-process PNGs in project folder using zopflipng or advpng

It is completely lossless and an easy win for PNG size.

Pre-process images

Manually optimize images in res/drawable/ folders using external tools.

Add following to build.gradle:

```
android {
    ...
    aaptOptions {
        cruncherEnabled = false
    }
}
```

WEBP for images

WebP gives ~30% smaller file sizes

Android 4.0+

Can replace JPEG and non-transparent PNGs with WebP

Android 4.2.1+ (adds lossless & transparency)

Can replace PNG with WebP

Remember to use it server-side!

Google Developer Day

VectorDrawables

Android 5.0+

Replace PNG icons with VectorDrawable.

(follow up
Android Studio Generate PNG)

VectorDrawables

Android < 5.0

```
android {
  defaultConfig {
    ...
    vectorDrawables {
       generatedDensities = ["mdpi", "hdpi"]
    }
  }
}
```

VectorDrawableCompat

Android < 5.0

Using AppCompat from the Support Library:

```
 // Gradle Plugin 2.0+
 android {
   defaultConfig {
     vectorDrawables.useSupportLibrary =
true
    }
 }
```

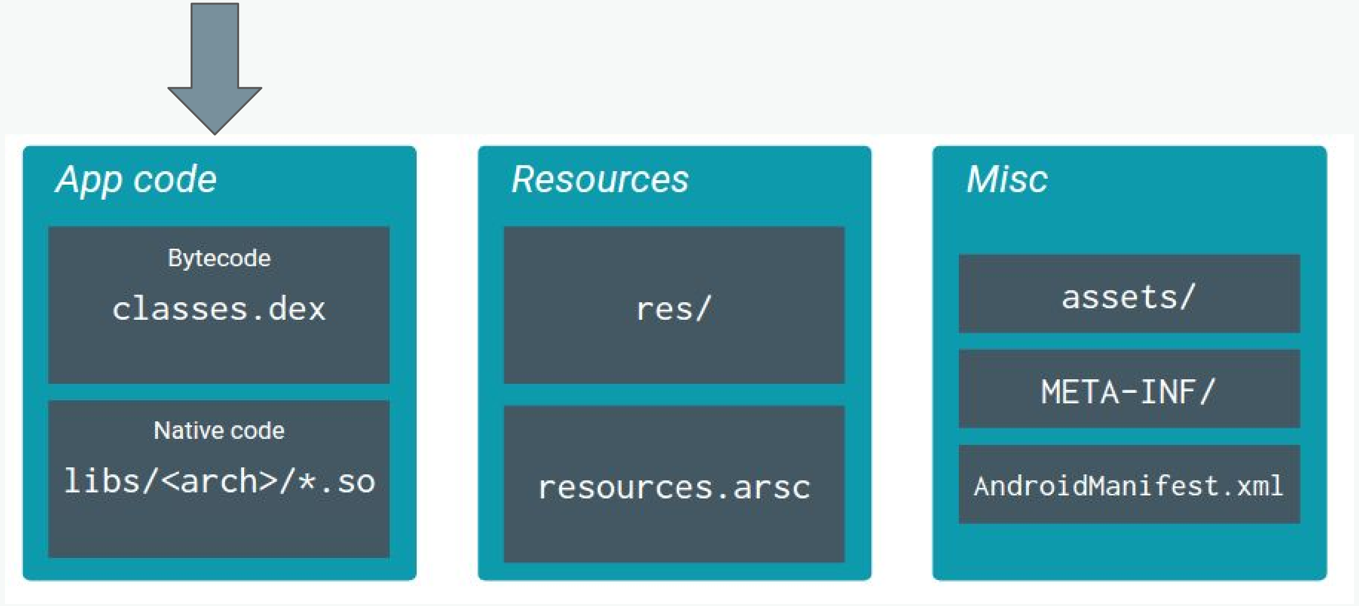
Google Developer Day

ShapeDrawables

Available since Android 1.0!

Can replace images with simple shapes or 9-patches,
e.x. button backgrounds, borders, gradients etc.

Optimizing App code



Google Developer Day

Use ProGuard

build.gradle

only enable minification for your release build

```
android {
    ...
    buildTypes {
        release {
            minifyEnabled true
            proguardFiles
getDefaultProguardFile('proguard-android.txt'),
'proguard-rules.pro'
        }
    }
}
```

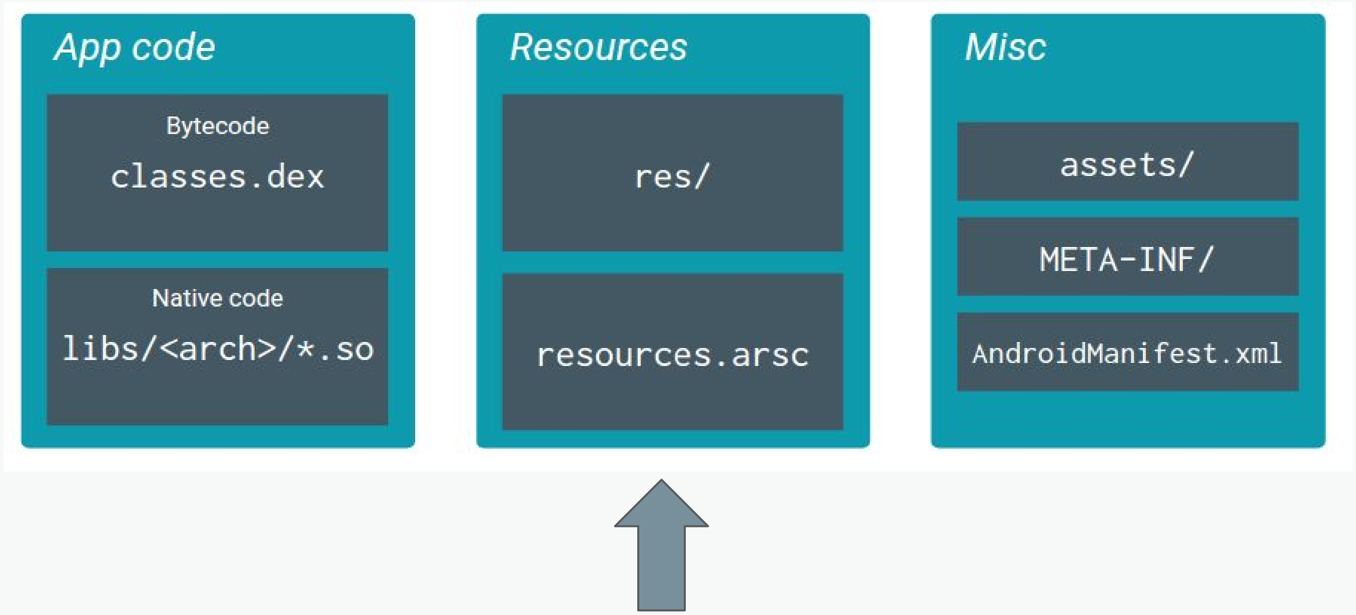
Use ProGuard

proguard-android.txt

```
# keep setters in Views so that animations can
still work.
# see
http://proguard.sourceforge.net/manual/examples.
html#beans
-keepclassmembers public class * extends
android.view.View {
   void set*(***);
   *** get*();
}
```

Google Developer Day

Optimize resConfigs



Google Developer Day

Remove unused resources

```
android {
  ...
  buildTypes {
    release {
      minifyEnabled true
      shrinkResources true
      proguardFiles
getDefaultProguardFile('proguard-android.txt'),
'proguard-rules.pro'
    }
  }
}
```

Remove unused resources

```
<resources
xmlns:tools="http://schemas.android.com/tools"

tools:keep="@layout/used*_c,@layout/used_a"/>



<resources
xmlns:tools="http://schemas.android.com/tools"
    tools:shrinkMode="safe"
    tools:discard="@layout/unused2" />
```

Remove resources using resConfigs

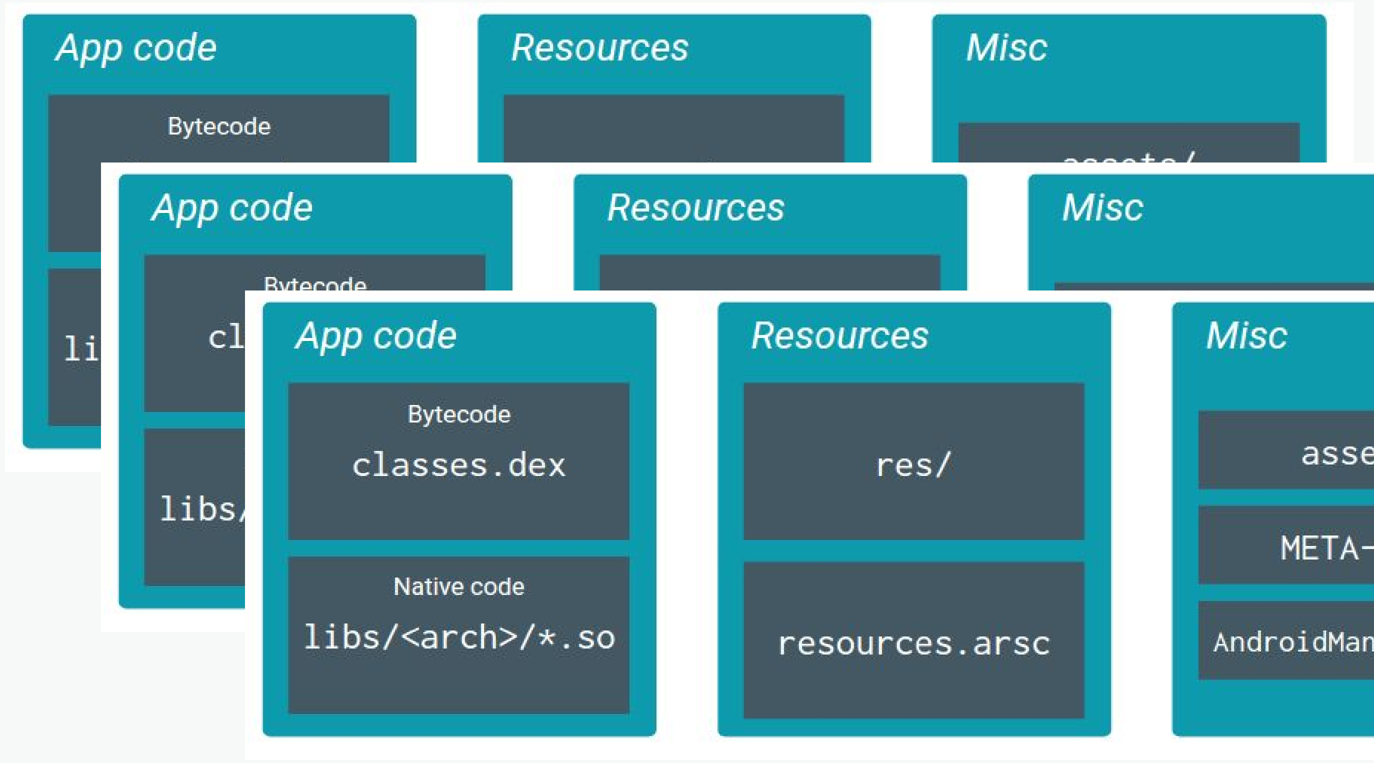
```
Remove resources using resConfigs



android {
    defaultConfig {
        ...
        resConfigs "en", "fr"
    }
}
```

~~`resConfigs "mdpi", "nodpi"`~~

Deprecated!

Google Developer Day

Split APKs



Google Developer Day

Split APKs
Density-based Multi-APK

```
android {
  ...
  splits {
    density {
      enable true
      exclude "ldpi", "tvdpi", "xxxhdpi"
      compatibleScreens 'small', 'normal', 
'large', 'xlarge'
    }
  }
}
```

Google Developer Day

Split APKs
Example density split savings

Topeka app:

Original (universal) APK:	3.5 MB
MDPI devices:	1.1 MB (2.4 MB savings)
HDPI devices:	1.2 MB (2.3 MB savings)
XHDPI devices:	1.3 MB (2.2 MB savings)

Split APKs
ABI-based
Multi-APK

```
android {
  ...
  splits {
    abi {
      enable true
      reset()
      include 'x86', 'armeabi-v7a'
      universalApk true
    }
  }
}
```

Split APKs
through flavors

```
flavorDimensions "density", "version"
productFlavors {
    hdpi {
        dimension "density"
        resConfigs "hdpi"
    }
    prelollipop {
        dimension "version"
        maxSdkVersion 19
        vectorDrawables {
            generatedDensities = ["mdpi", "hdpi"]
        }
    }
    postlollipop {
        dimension "version"
        minSdkVersion 21
```

Google Developer Day

Google Developer Day