# Android内存优化的5R法则

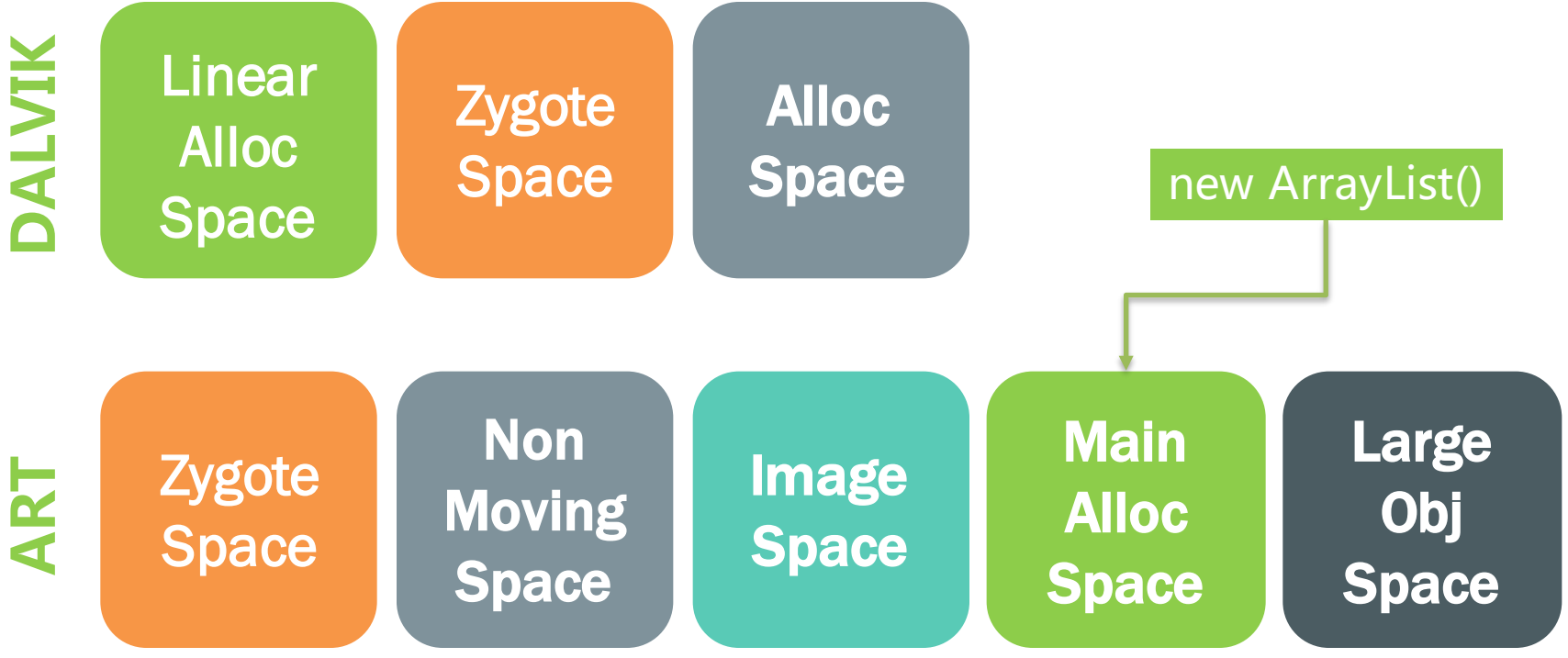
胡凯

Android

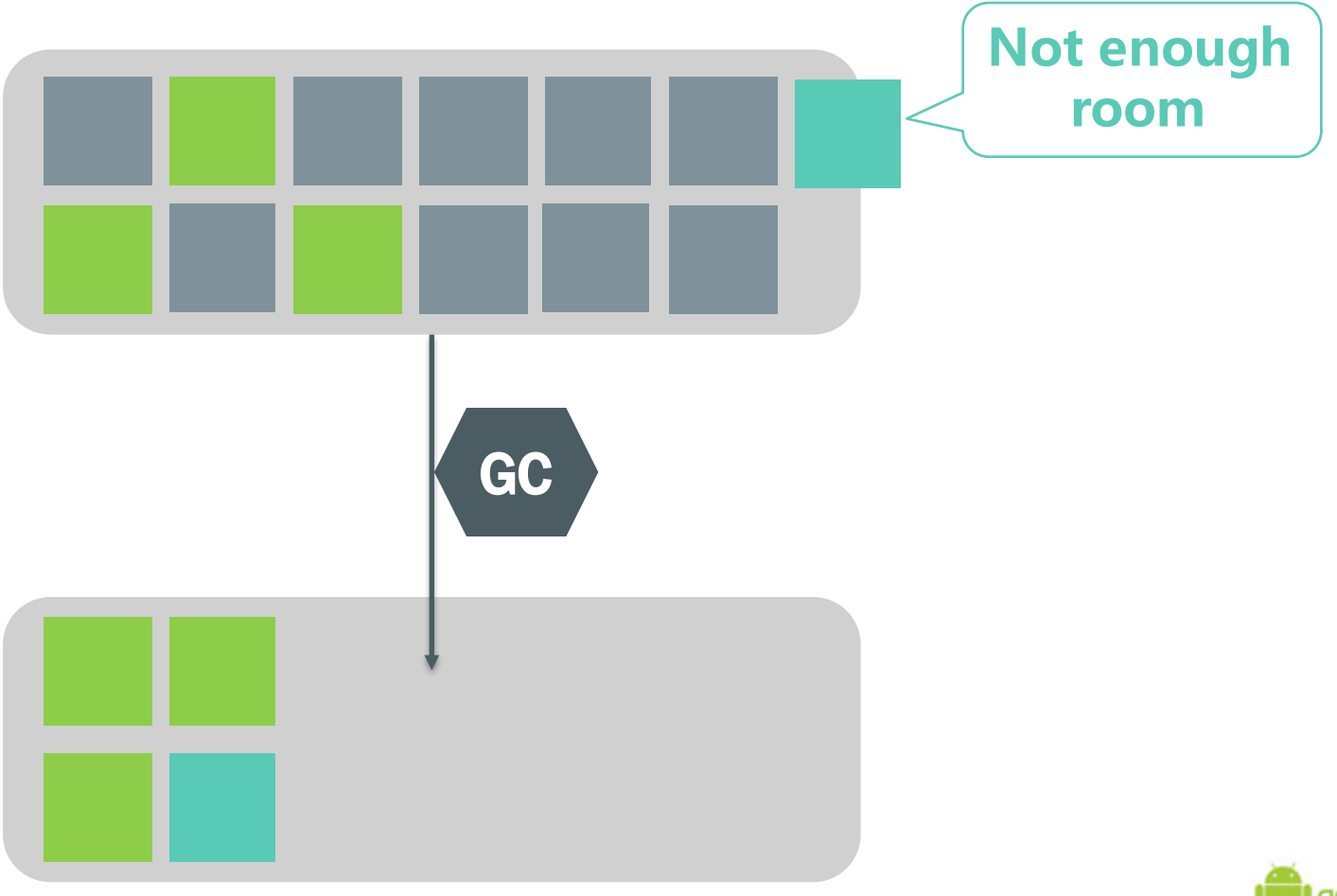
hukai.me

腾讯

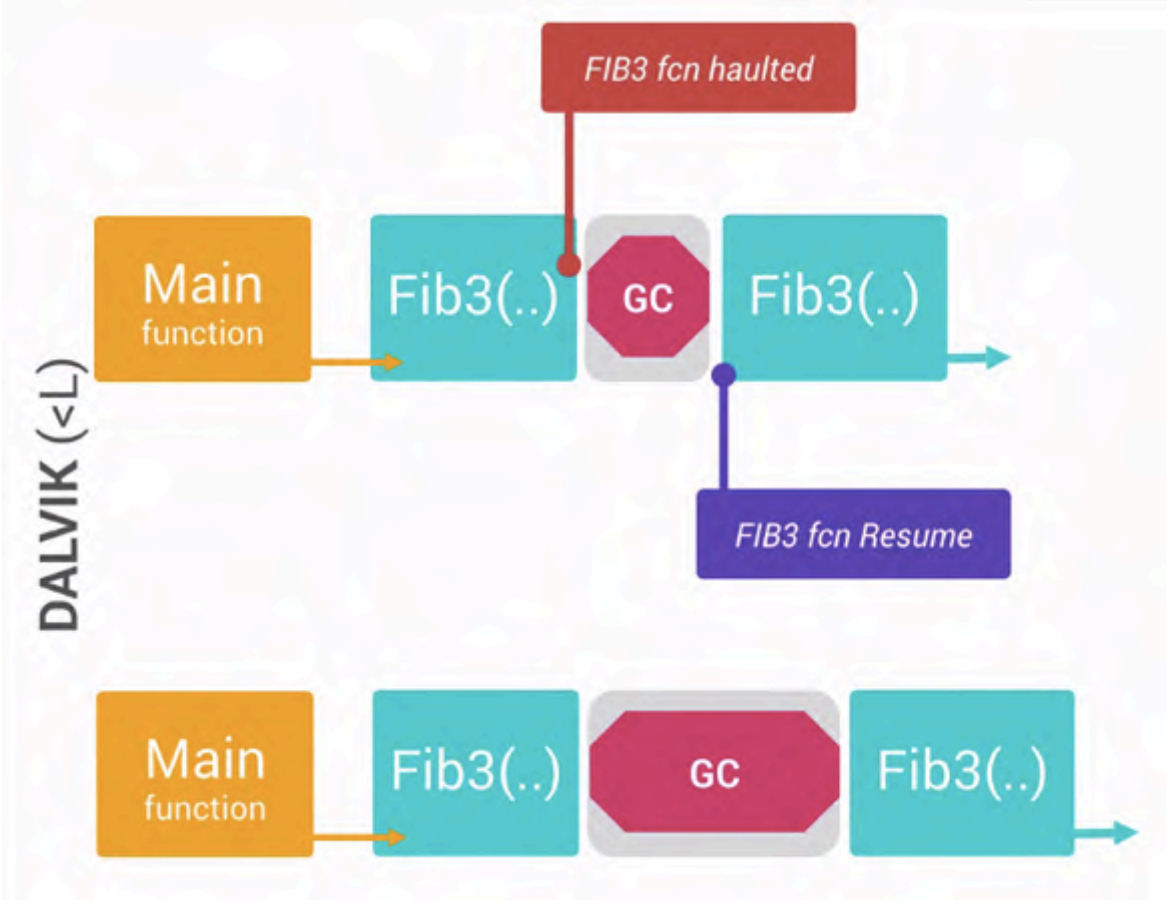
# 内存管理基础 – 共享内存


DALVIK
Linear Alloc Space
Zygote Space
Alloc Space
new ArrayList()
ART
Zygote Space
Non Moving Space
Image Space
Main Alloc Space
Large Obj Space

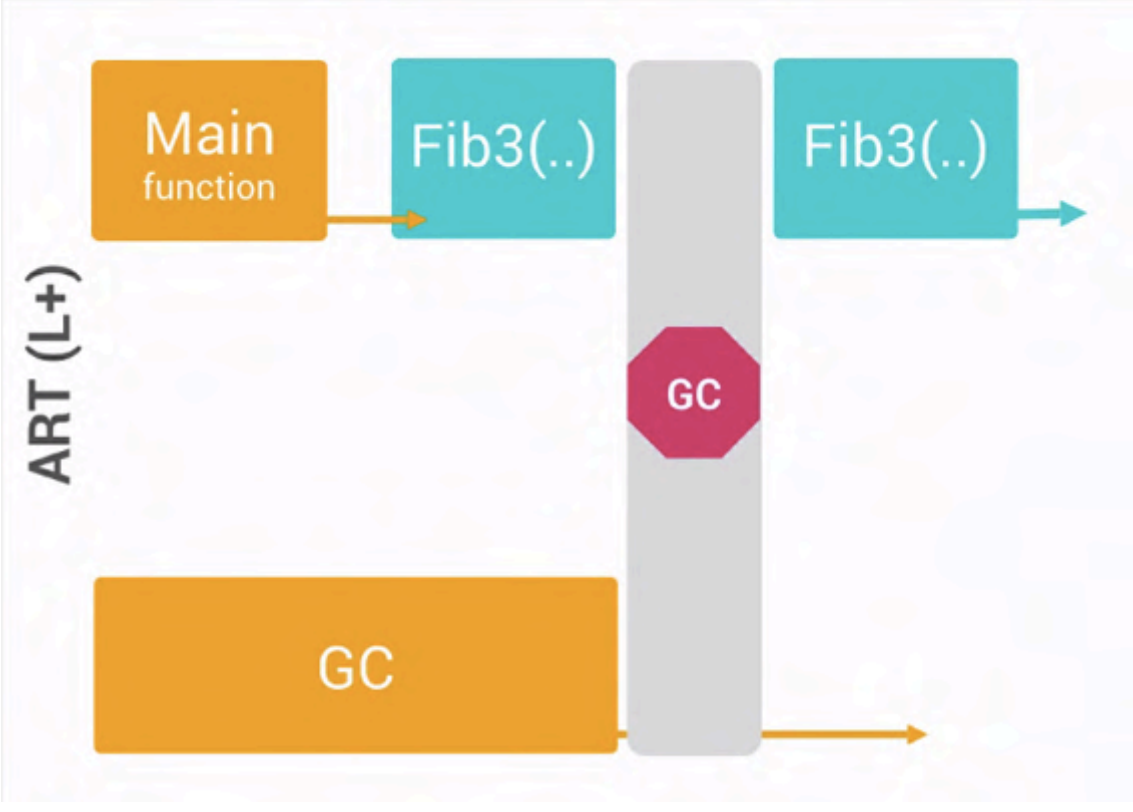
# 内存管理基础 – 内存回收



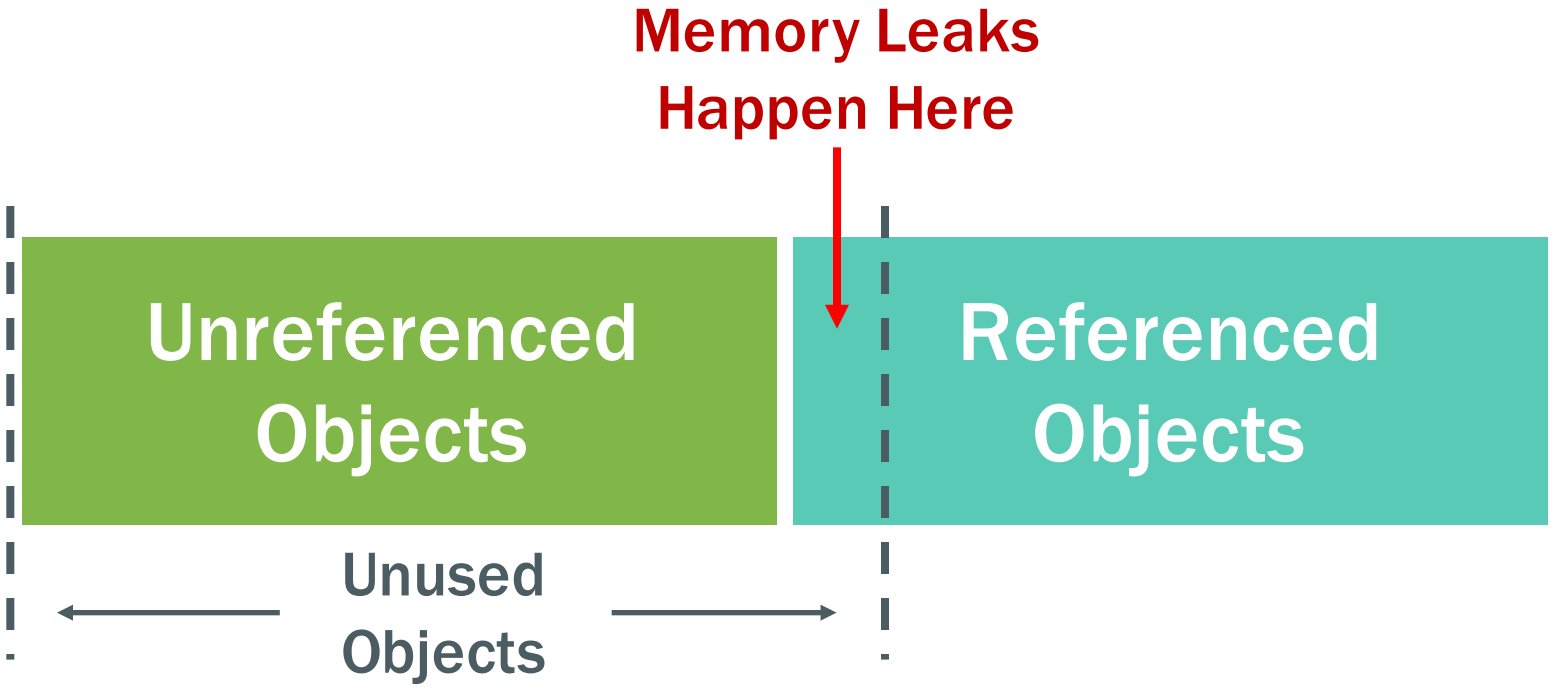
# 内存管理基础 - 内存回收


FIB3 fcn haulted
Main function
Fib3(..)
GC
Fib3(..)
FIB3 fcn Resume
DALVIK (<L)
Main function
Fib3(..)
GC
Fib3(..)

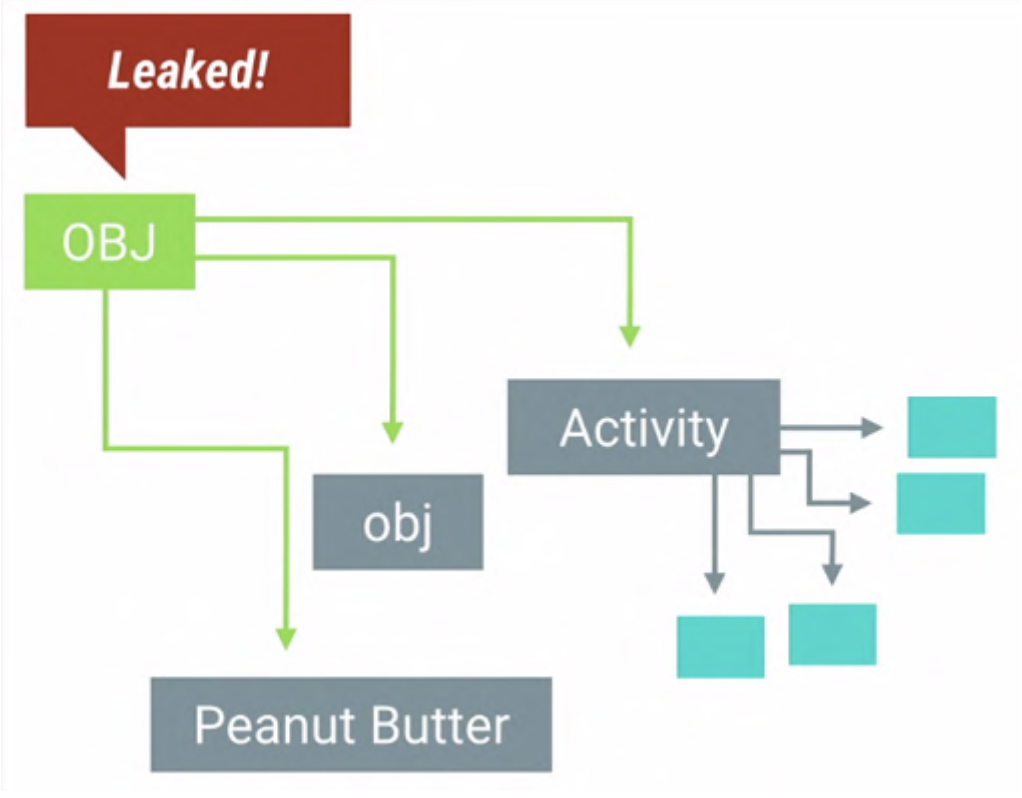
# 内存管理基础 - 内存回收



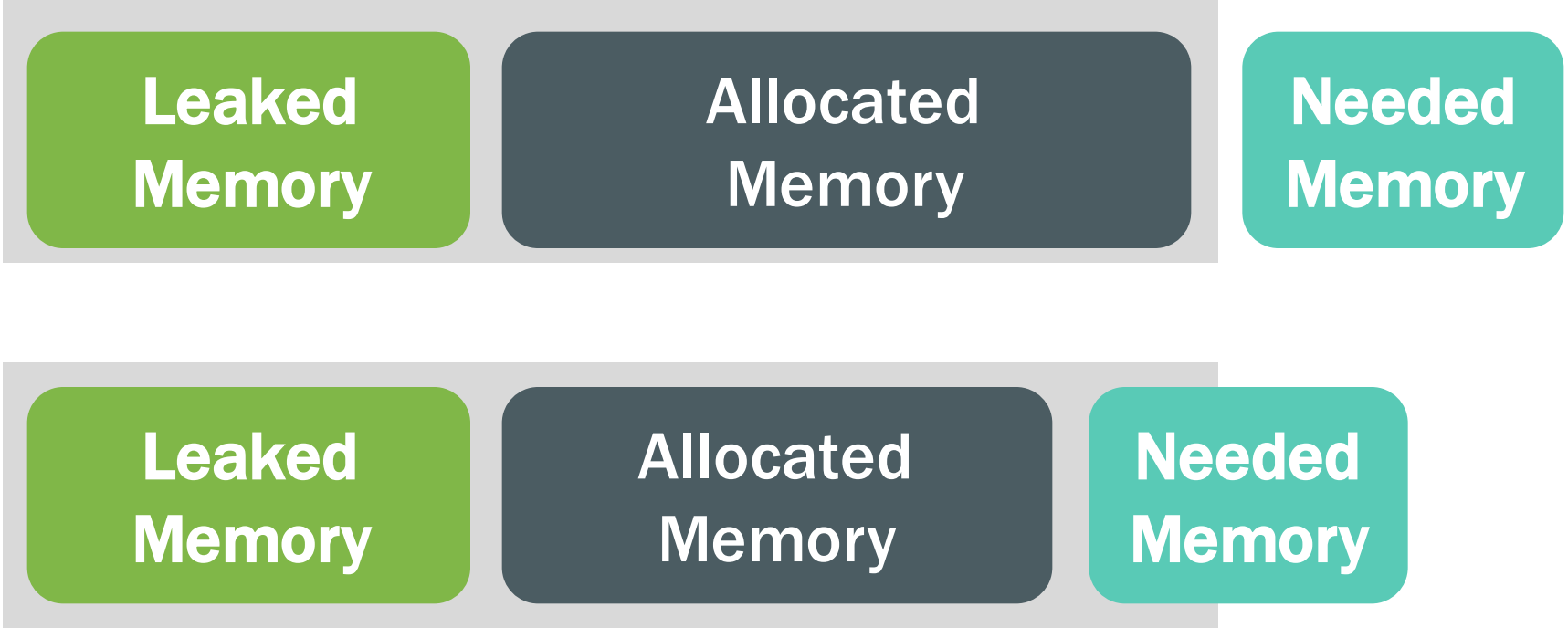
# 内存管理基础 - 内存泄漏


Memory Leaks
Happen Here
Unreferenced
Objects
Referenced
Objects
Unused
Objects

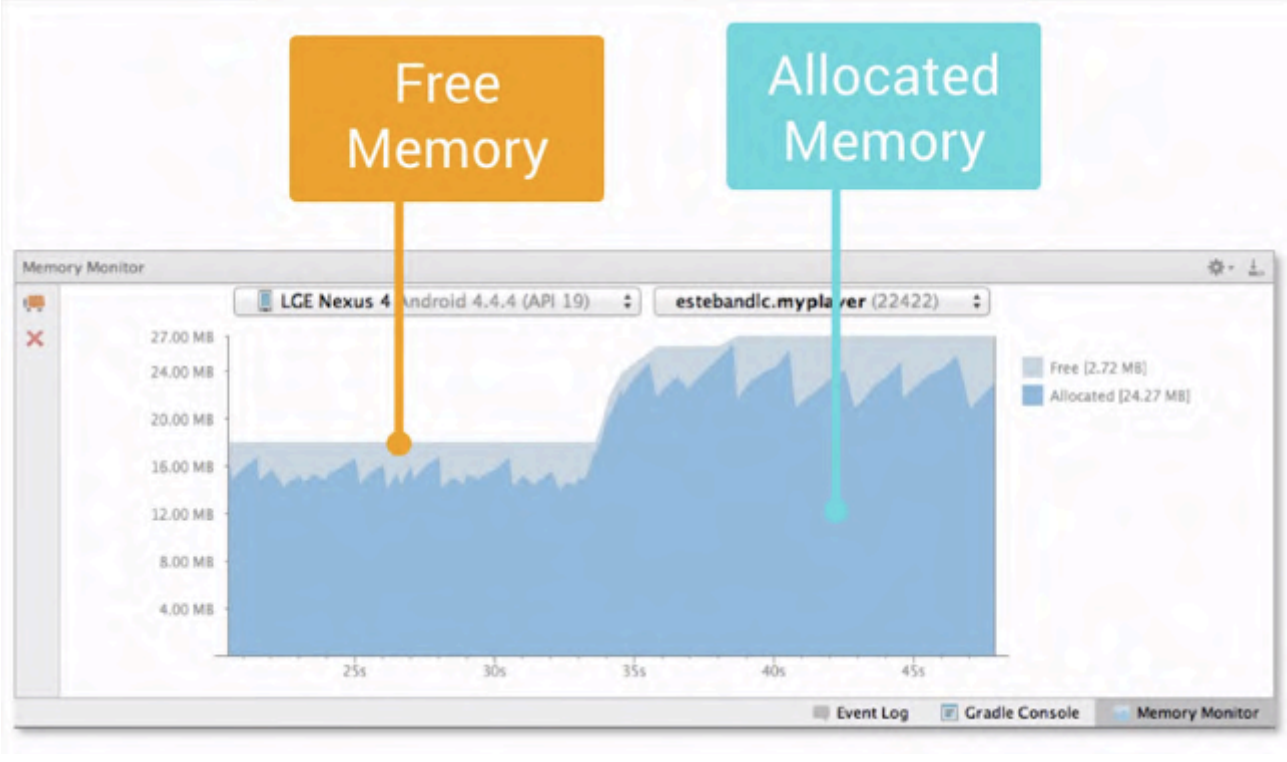
# 内存管理基础 - 内存泄漏



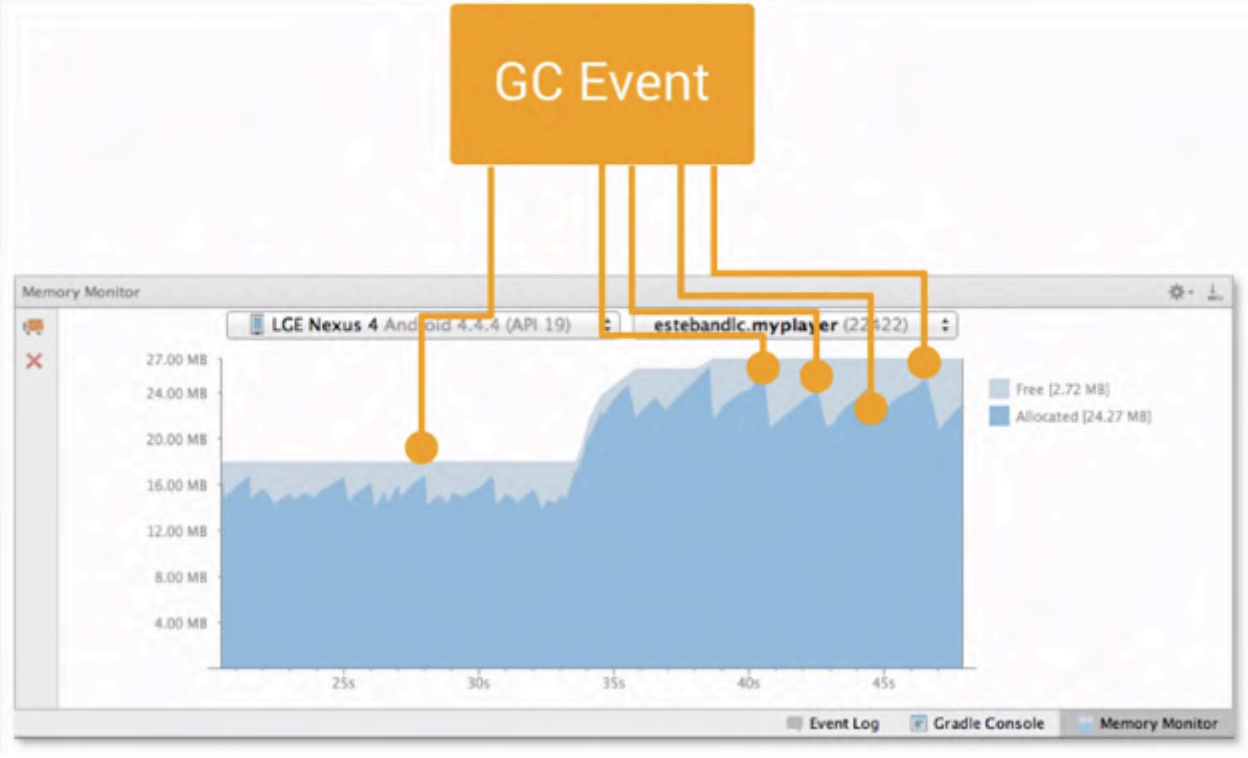
# 内存管理基础 - 内存泄漏


Leaked Memory
Allocated Memory
Needed Memory
Leaked Memory
Allocated Memory
Needed Memory

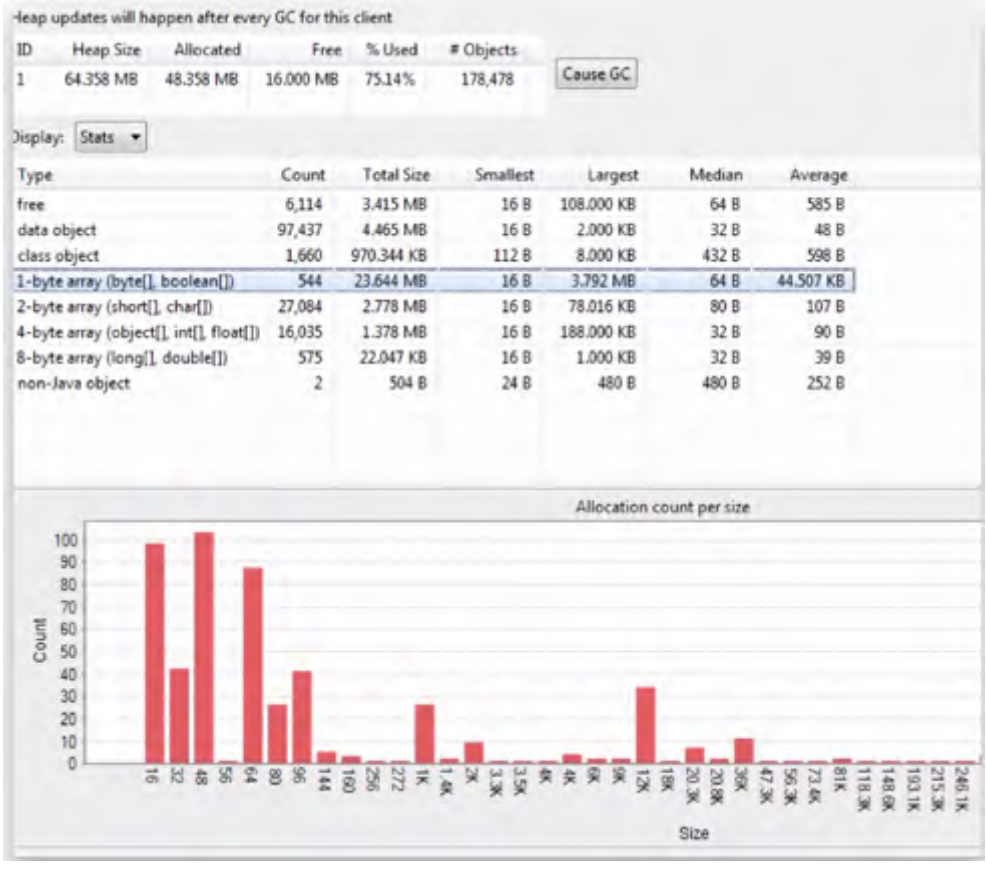
# 内存管理基础 - 内存分析


Free Memory
Allocated Memory
Memory Monitor
LGE Nexus 4
Android 4.4.4 (API 19)
estebandlc.myplayer (22422)
27.00 MB
24.00 MB
20.00 MB
16.00 MB
12.00 MB
8.00 MB
4.00 MB
25s
30s
35s
40s
45s
Free [2.72 MB]
Allocated [24.27 MB]
Event Log
Gradle Console
Memory Monitor

# 内存管理基础 - 内存分析


GC Event
Memory Monitor
LGE Nexus 4 Android 4.4.4 (API 19)
estebandlc.myplayer (22422)
27.00 MB
24.00 MB
20.00 MB
16.00 MB
12.00 MB
8.00 MB
4.00 MB
25s
30s
35s
40s
45s
Free [2.72 MB]
Allocated [24.27 MB]
Event Log
Gradle Console
Memory Monitor

# 内存管理基础 - 内存分析


Heap updates will happen after every GC for this client
ID Heap Size Allocated Free % Used # Objects
1 64.358 MB 48.358 MB 16.000 MB 75.14% 178,478
Cause GC
Display: Stats
Type Count Total Size Smallest Largest Median Average
free 6,114 3.415 MB 16 B 108.000 KB 64 B 585 B
data object 97,437 4.465 MB 16 B 2.000 KB 32 B 48 B
class object 1,660 970.344 KB 112 B 8.000 KB 432 B 598 B
1-byte array (byte[], boolean[]) 544 23.644 MB 16 B 3.792 MB 64 B 44.507 KB
2-byte array (short[], char[]) 27,084 2.778 MB 16 B 78.016 KB 80 B 107 B
4-byte array (object[], int[], float[]) 16,035 1.378 MB 16 B 188.000 KB 32 B 90 B
8-byte array (long[], double[]) 575 22.047 KB 16 B 1.000 KB 32 B 39 B
non-Java object 2 504 B 24 B 480 B 480 B 252 B
Allocation count per size
Count
Size

# 内存管理基础 - 内存分析

```
→  ~  adb shell dumpsys meminfo -a com.example.android.activitylifecycle
Applications Memory Usage (kB):
Uptime: 757671 Realtime: 757671

** MEMINFO in pid 1760 [com.example.android.activitylifecycle] **
```

| | Pss Total | Pss Clean | Shared Dirty | Private | Shared Clean | Private Clean | Swapped Dirty | Heap Size | Heap Alloc | Heap Free |
|---|---|---|---|---|---|---|---|---|---|---|
| Native Heap | 6432 | 0 | 1944 | 6344 | 0 | 0 | 0 | 16721 | 16721 | 15022 |
| Dalvik Heap | 1923 | 0 | 8316 | 1576 | 0 | 0 | 0 | 12152 | 7443 | 4709 |
| Dalvik Other | 448 | 0 | 4 | 448 | 0 | 0 | 0 | | | |
| Stack | 112 | 0 | 0 | 112 | 0 | 0 | 0 | | | |
| Other dev | 4 | 0 | 28 | 0 | 0 | 4 | 0 | | | |
| .so mmap | 1114 | 180 | 2588 | 148 | 6188 | 180 | 0 | | | |
| .apk mmap | 148 | 0 | 0 | 0 | 1064 | 0 | 0 | | | |
| .ttf mmap | 34 | 4 | 0 | 0 | 88 | 4 | 0 | | | |
| .dex mmap | 148 | 144 | 0 | 0 | 8 | 144 | 0 | | | |
| code mmap | 1320 | 204 | 0 | 0 | 9276 | 204 | 0 | | | |
| image mmap | 1953 | 648 | 2560 | 504 | 7824 | 648 | 0 | | | |
| Other mmap | 64 | 0 | 8 | 4 | 140 | 52 | 0 | | | |
| Unknown | 89 | 0 | 100 | 88 | 0 | 0 | 0 | | | |
| TOTAL | 13789 | 1180 | 15548 | 9224 | 24588 | 1236 | 0 | 28873 | 24164 | 19731 |

# 内存管理基础 - 内存分析

## Memory Analyzer (MAT)

The Eclipse Memory Analyzer is a fast and feature-rich **Java heap analyzer** that helps you find memory leaks and reduce memory consumption.

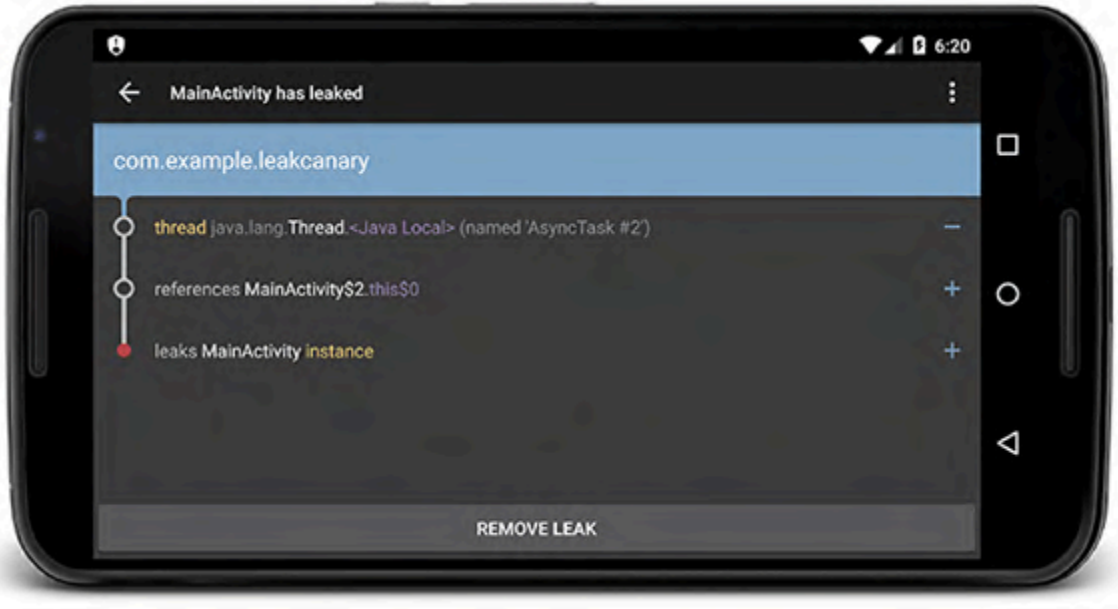
Use the Memory Analyzer to analyze productive heap dumps with hundreds of millions of objects, quickly calculate the retained sizes of objects, see who is preventing the Garbage Collector from collecting objects, run a report to automatically extract leak suspects.

# 内存管理基础 - 内存分析

## LeakCanary

A memory leak detection library for Android and Java.

*"A small leak will sink a great ship."* - Benjamin Franklin



More Free Space

Less Garbage Collection

1. Reduce
2. Reuse
3. Recycle
4. Refactor
5. Revalue


# Reduce

# 1）差异化对待 - 缓存容量

Nexus 4

Nexus 5

Nexus 6

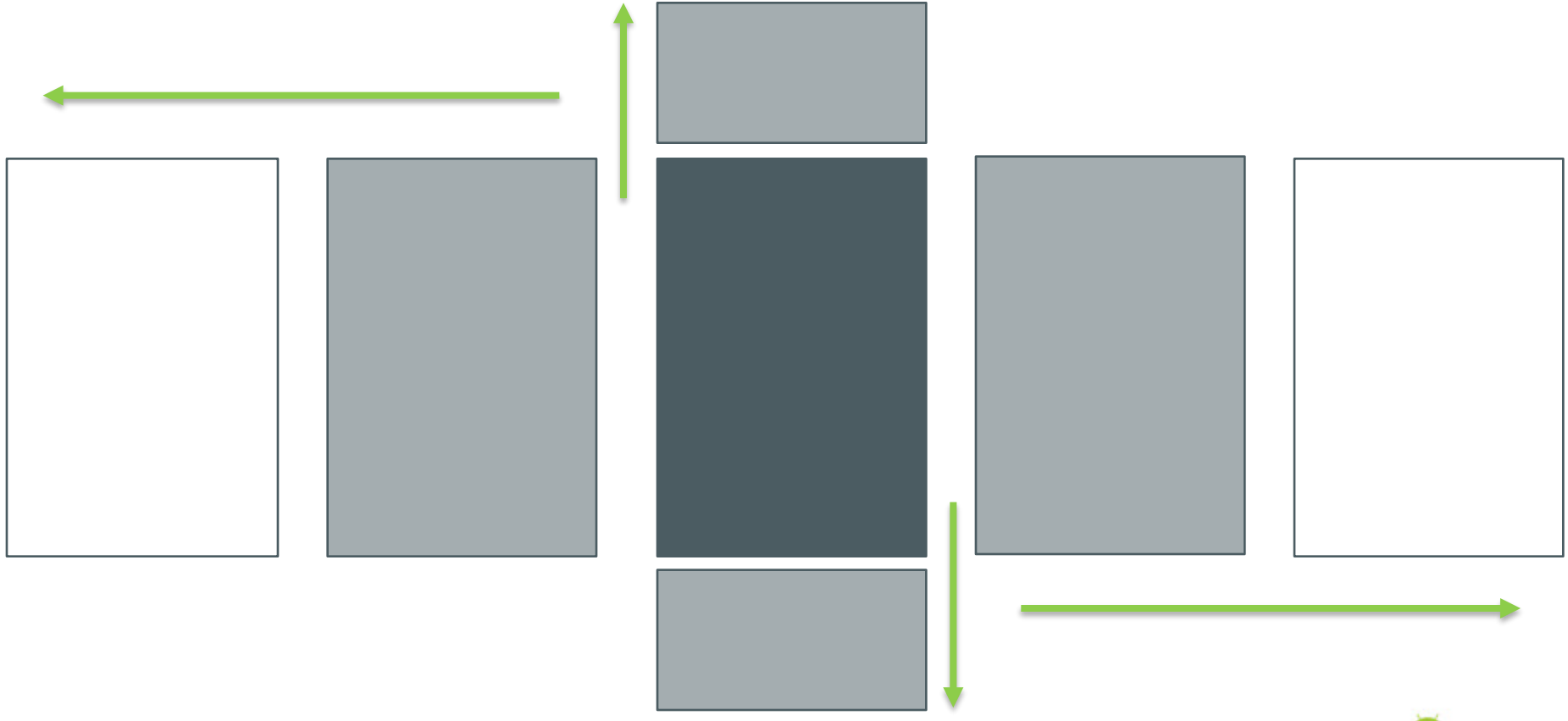
# 1）差异化对待 - 图片资源

xhdpi

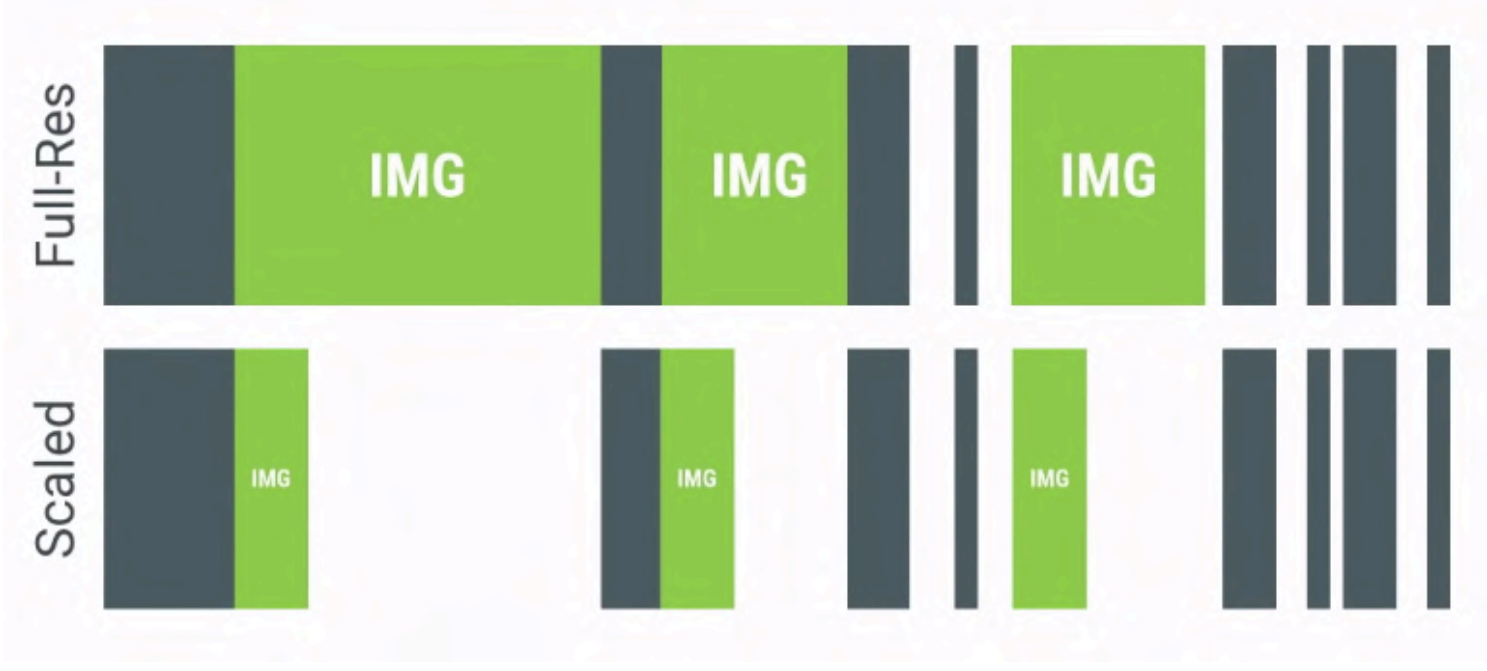
xxhdpi

xxxhdpi

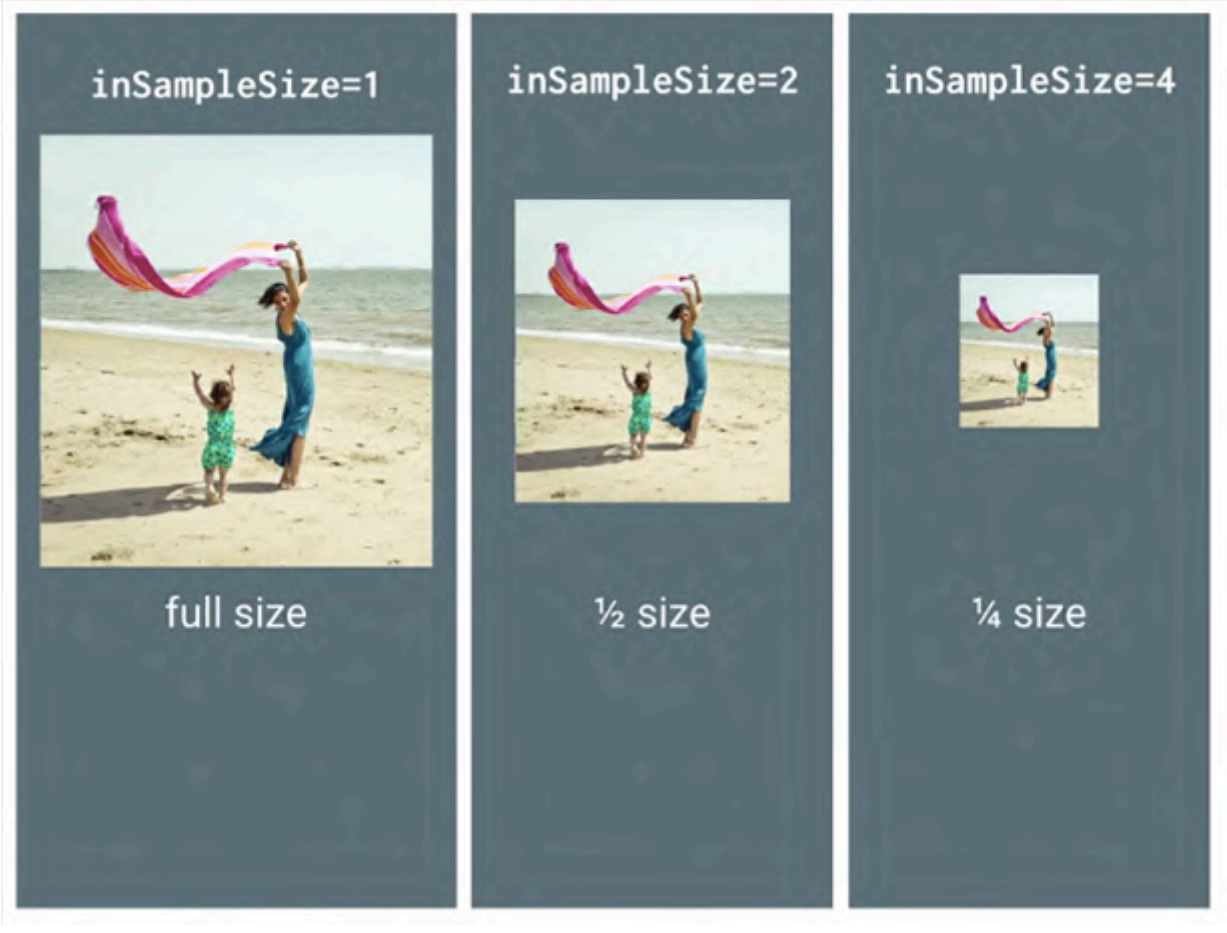
# 2）资源按需加载



# 3）减小Bitmap对象的内存占用



# 3）减小Bitmap对象的内存占用 - inSampleSize



# 3）减小Bitmap对象的内存占用 – decode format

ARGB_8888 :32 Bits Per Pixel

RBG_565 :16 B

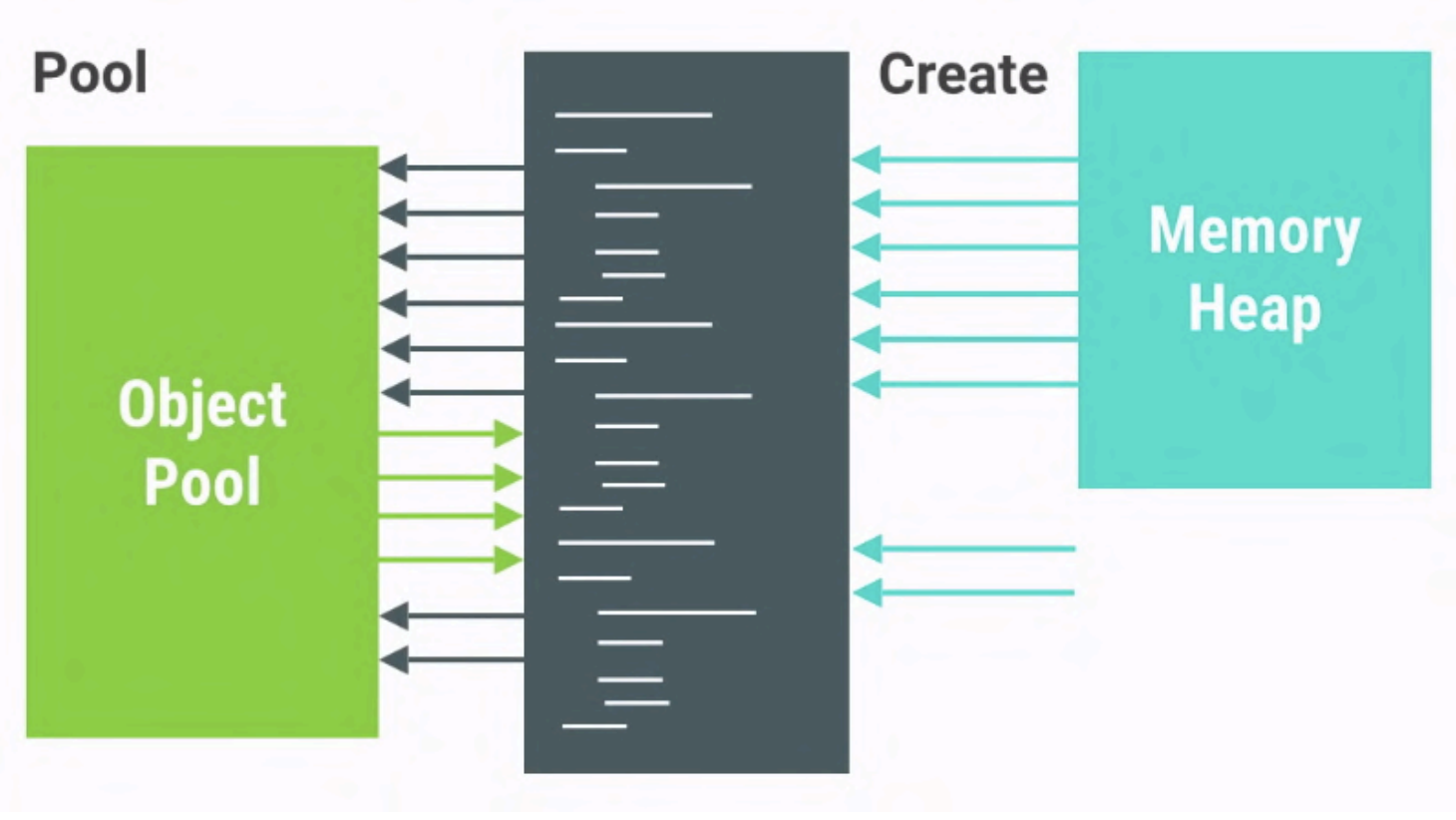
ARGB_4444 :16 B

ALPHA_8 : 8 B

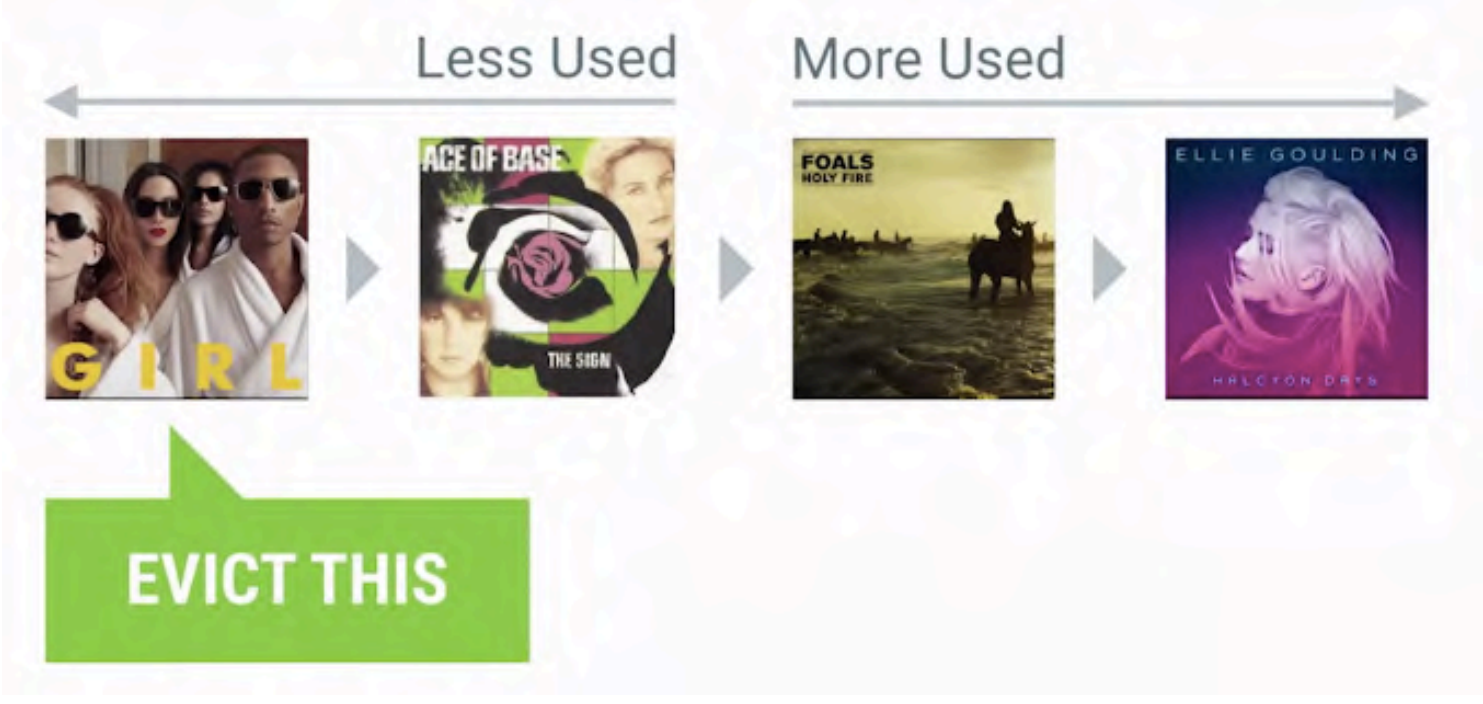
# 2

# Reuse

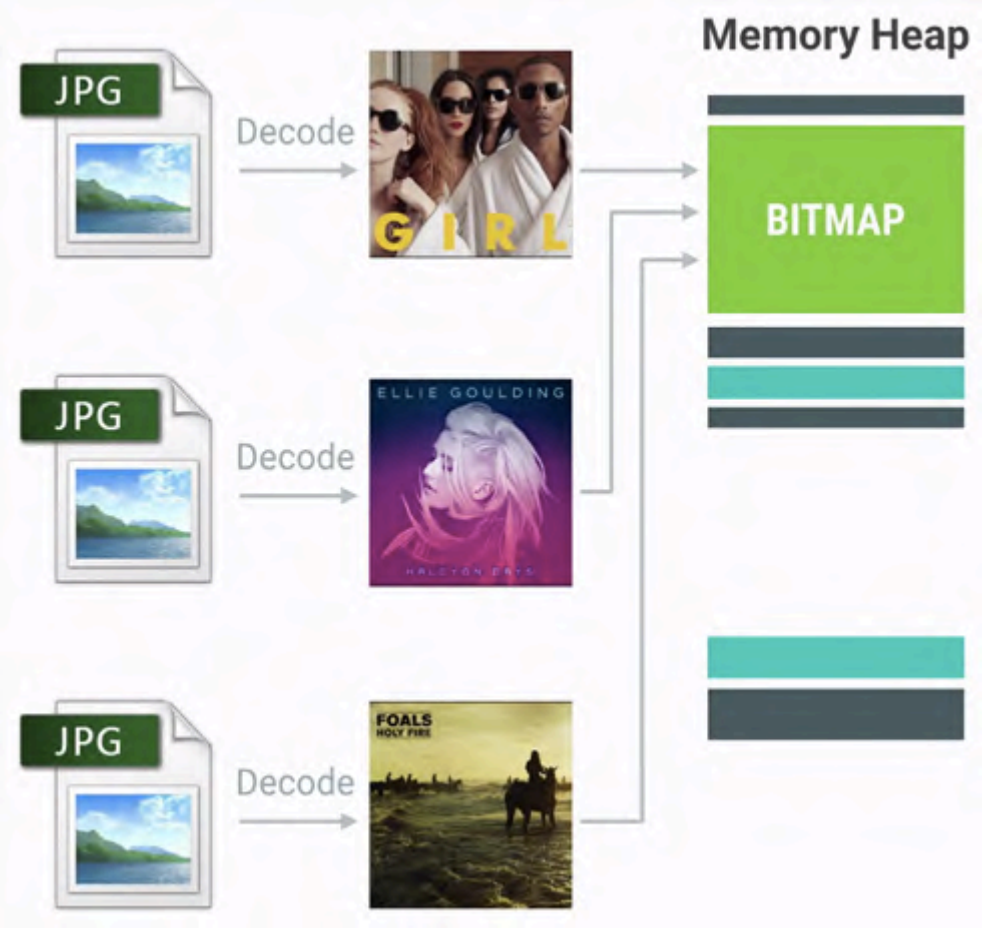
# 1 ) Pools



# 1 ) Pools

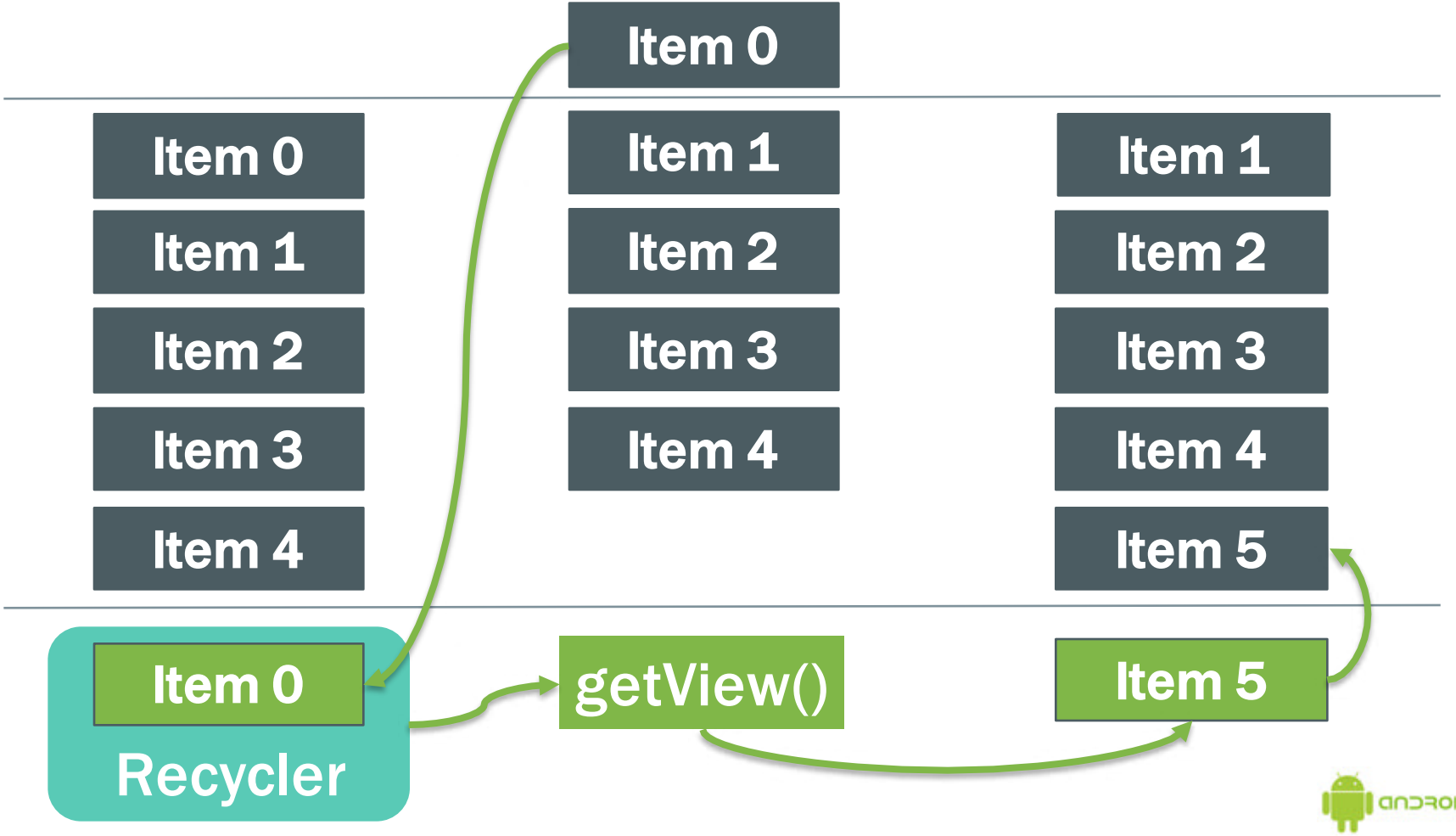

Less Used
More Used
GIRL
ACE OF BASE
THE SIGN
FOALS
HOLY FIRE
ELLIE GOULDING
HALCYON DAYS
EVICT THIS

# 2）Bitmap对象的复用 - inBitmap

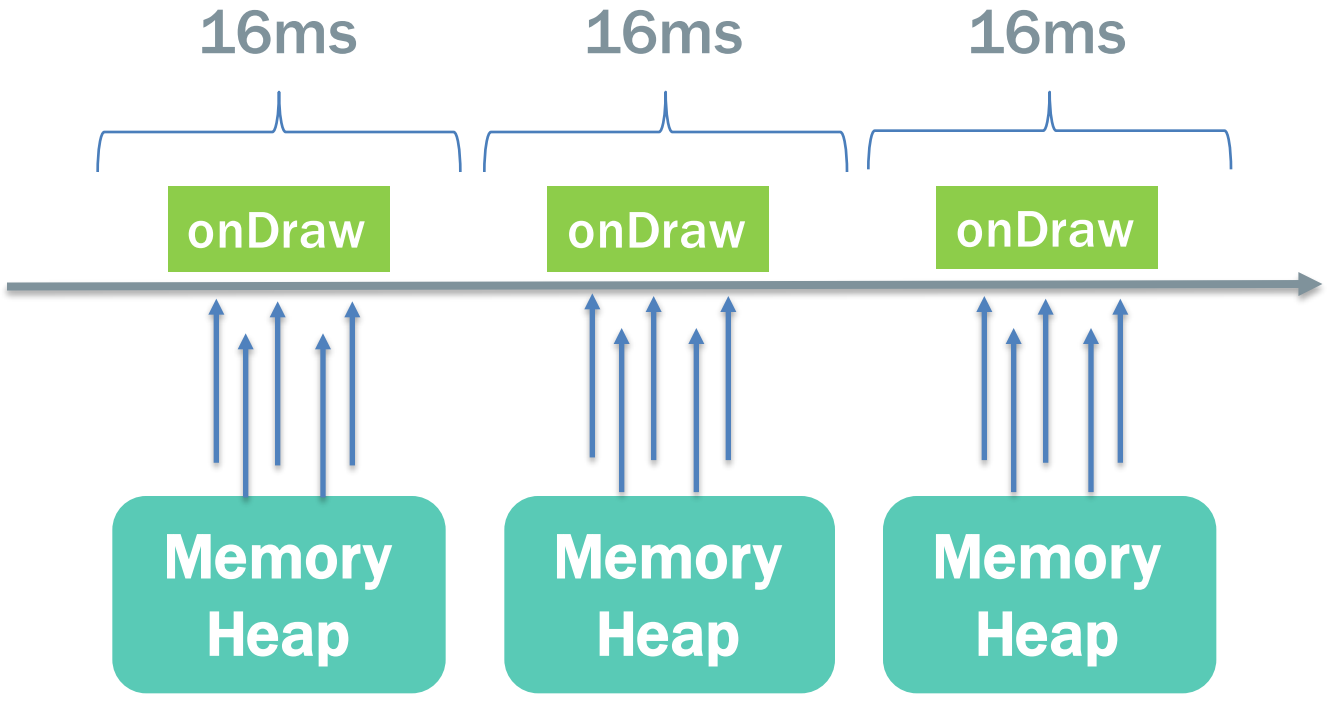


# 2）Bitmap对象的复用 - inBitmap

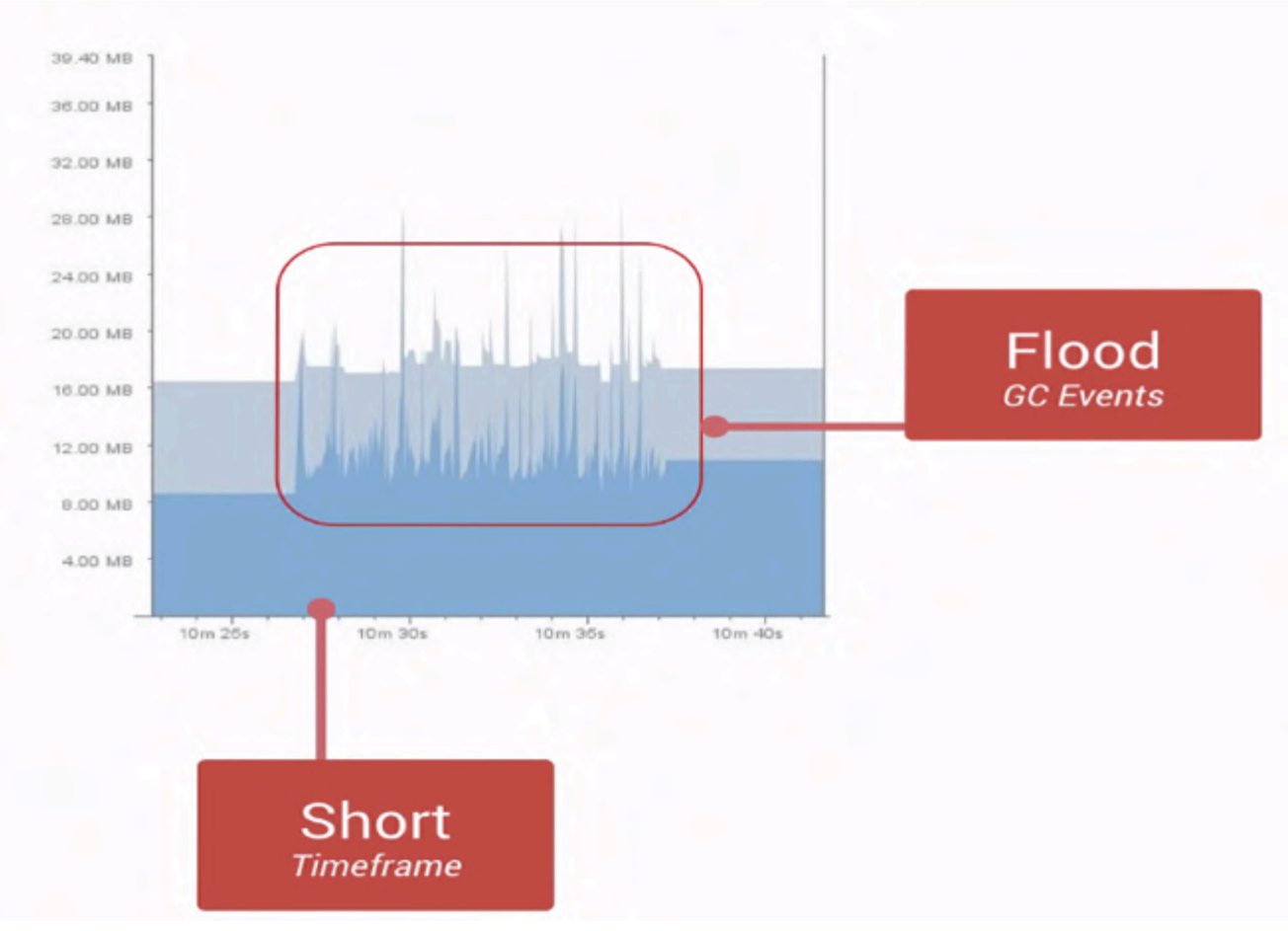
# 3）注意在ListView/GridView中ConvertView的复用



# 4）避免频繁的创建与回收对象



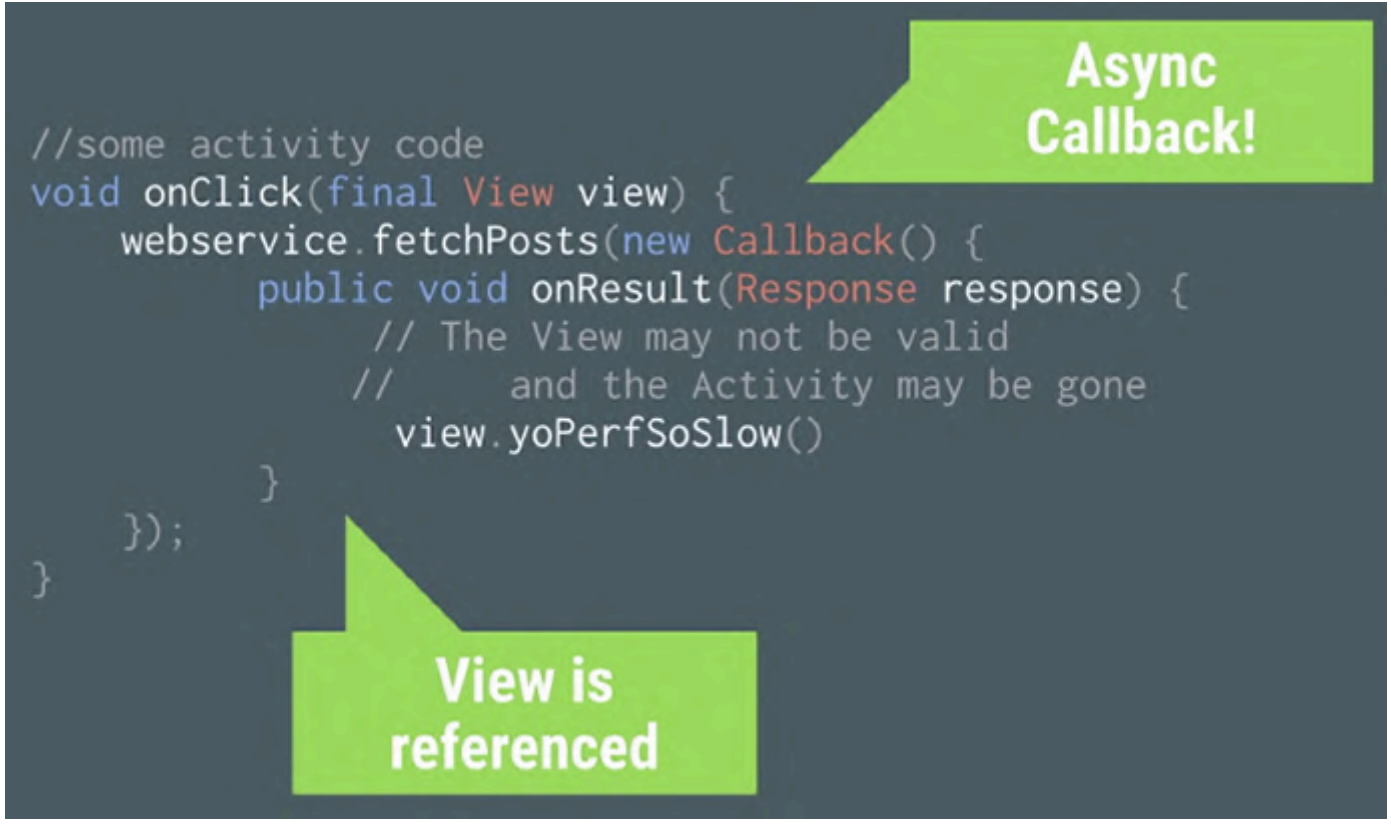
# 4）避免频繁的创建与回收对象



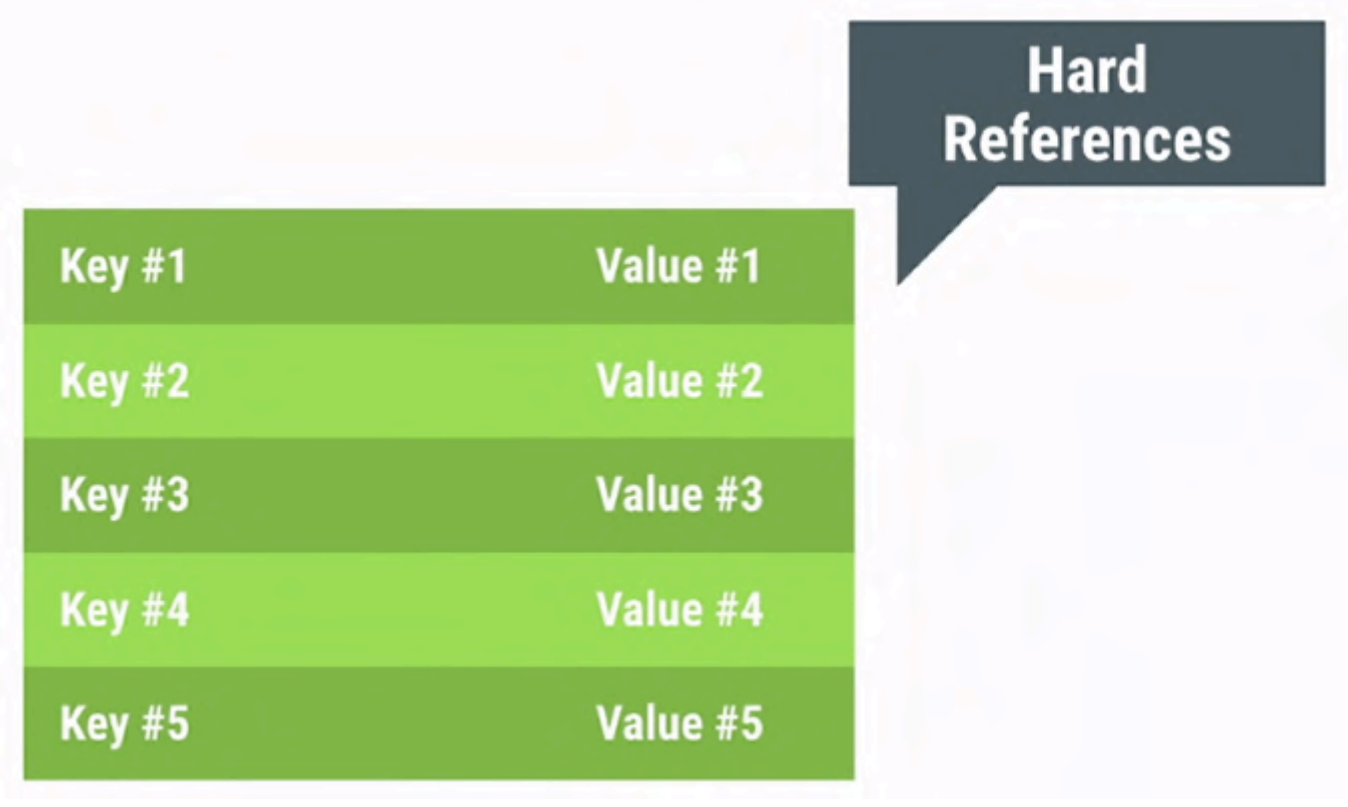
# 3

# Recycle

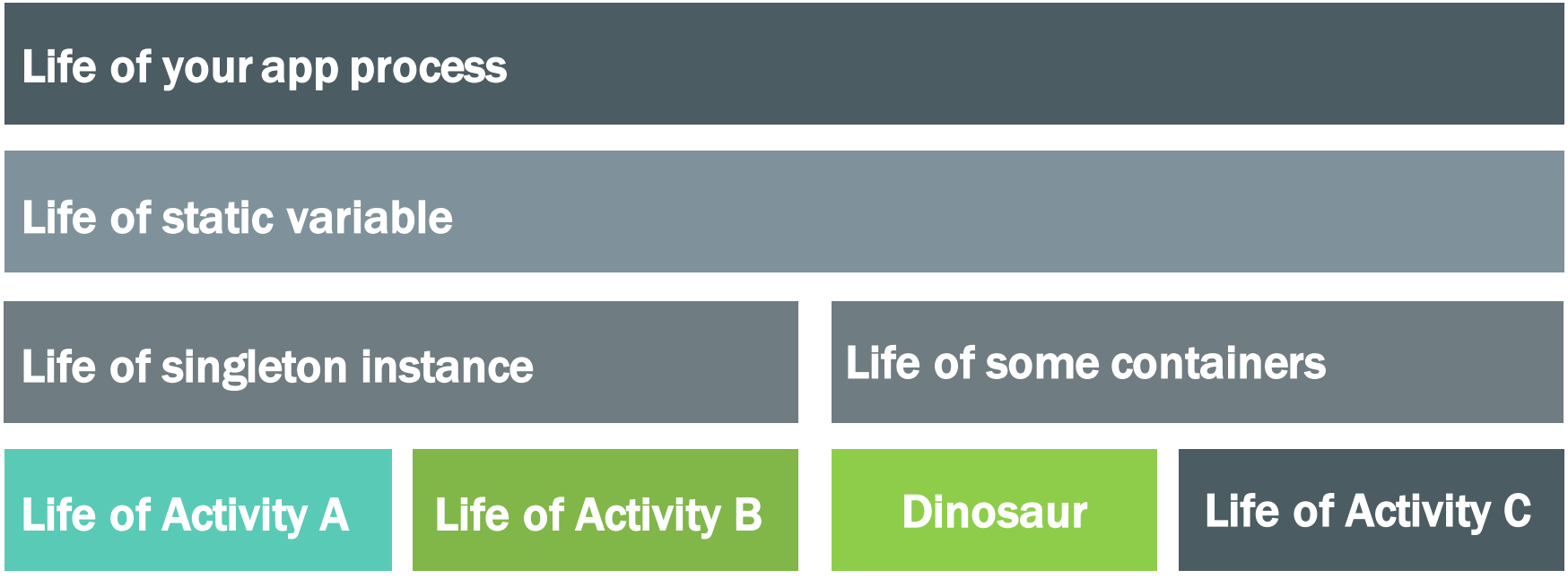
# 1）Activity泄漏 - 内部类



# 1）Activity泄漏 - 容器对象泄漏



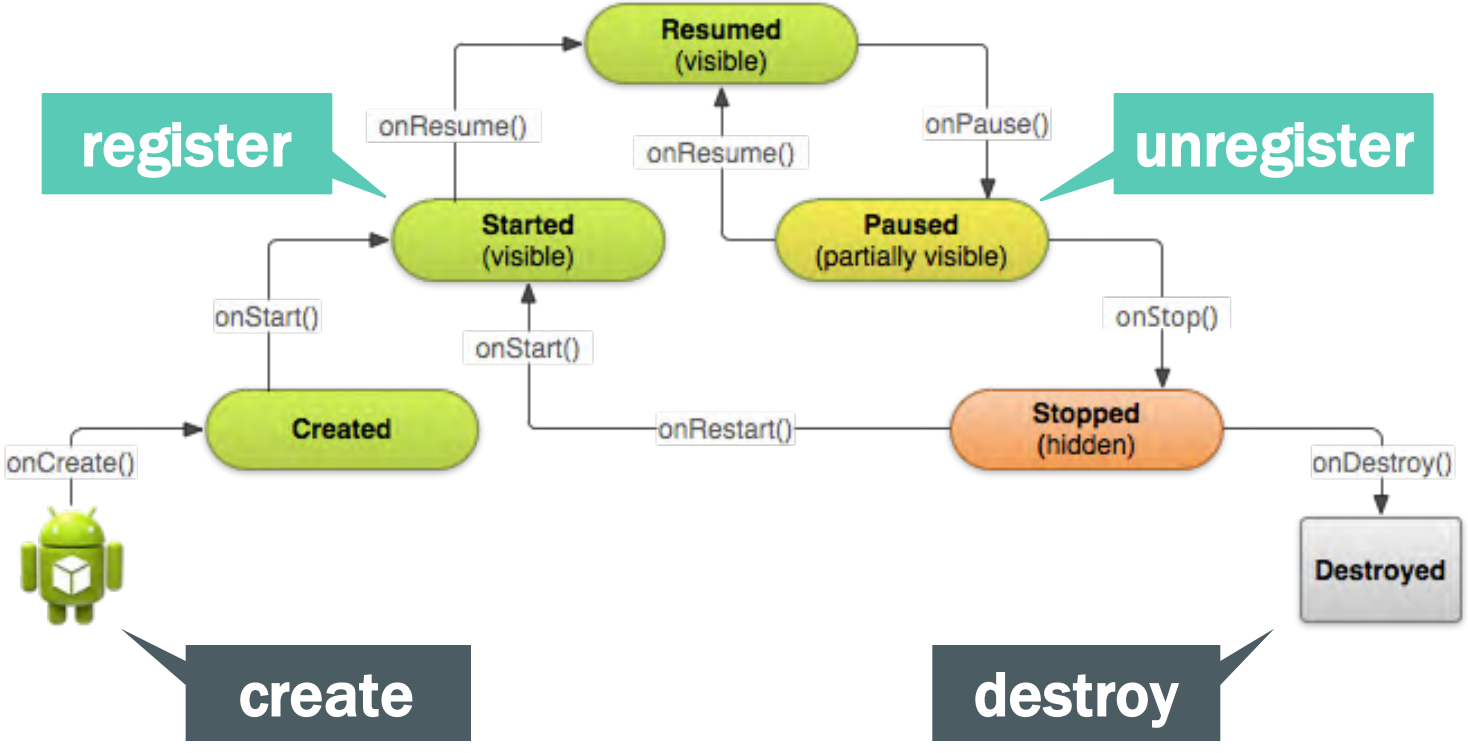
# 1）Activity泄漏 - Static, Singleton



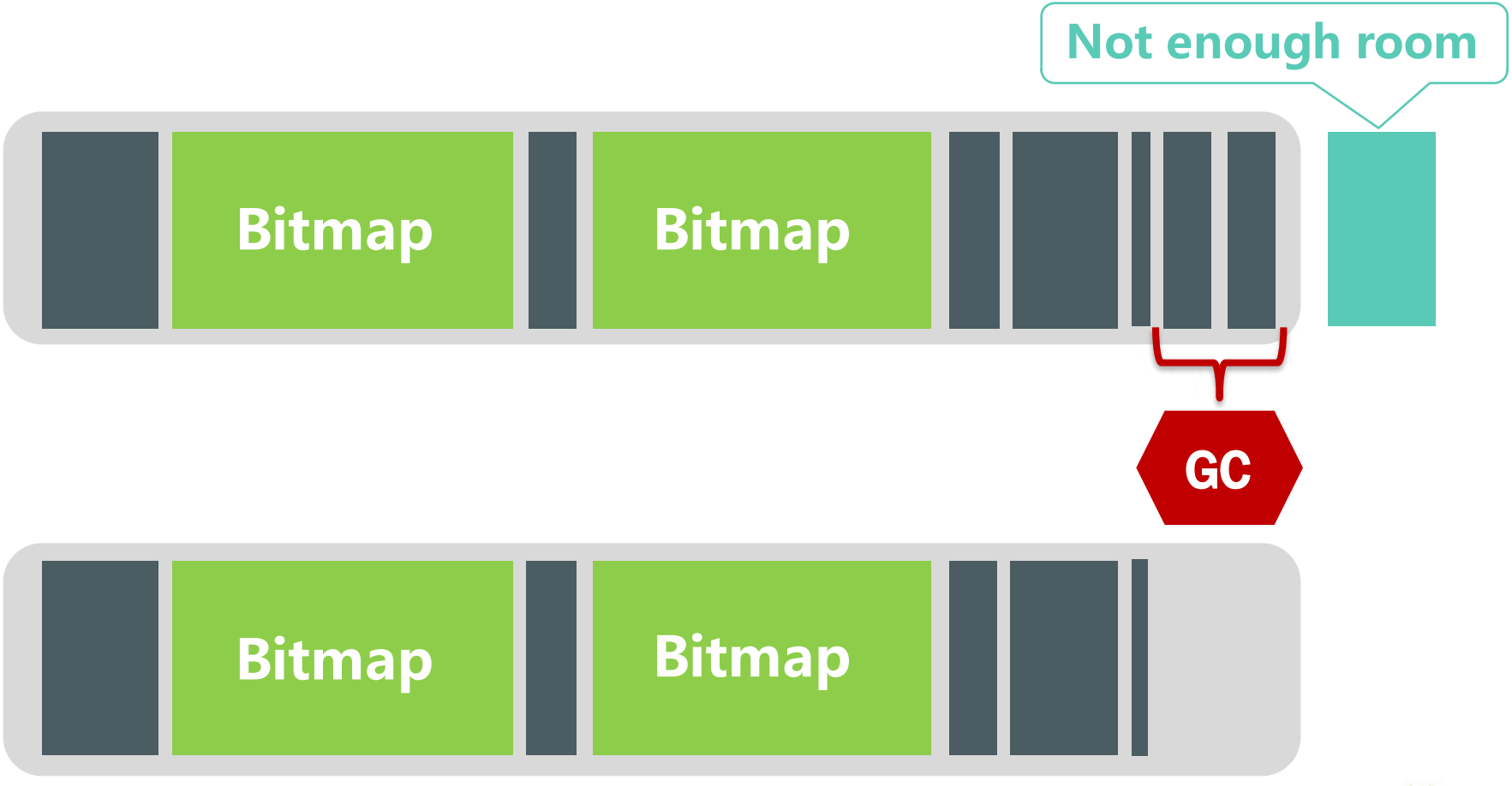
# 2）谨慎选择合适的Context

| | Application | Activity | Service | Content Provider | Broadcast Receiver |
|---|---|---|---|---|---|
| Show a Dialog | NO | YES | NO | NO | NO |
| Start an Activity | NO[1] | YES | NO[1] | NO[1] | NO[1] |
| Layout Inflation | NO[2] | YES | NO[2] | NO[2] | NO[2] |
| Start a Service | YES | YES | YES | YES | YES |
| Bind to a Service | YES | YES | YES | YES | NO |
| Send a Broadcast | YES | YES | YES | YES | YES |
| Register BroadcastReceiver | YES | YES | YES | YES | NO[3] |
| Load Resource Values | YES | YES | YES | YES | YES |

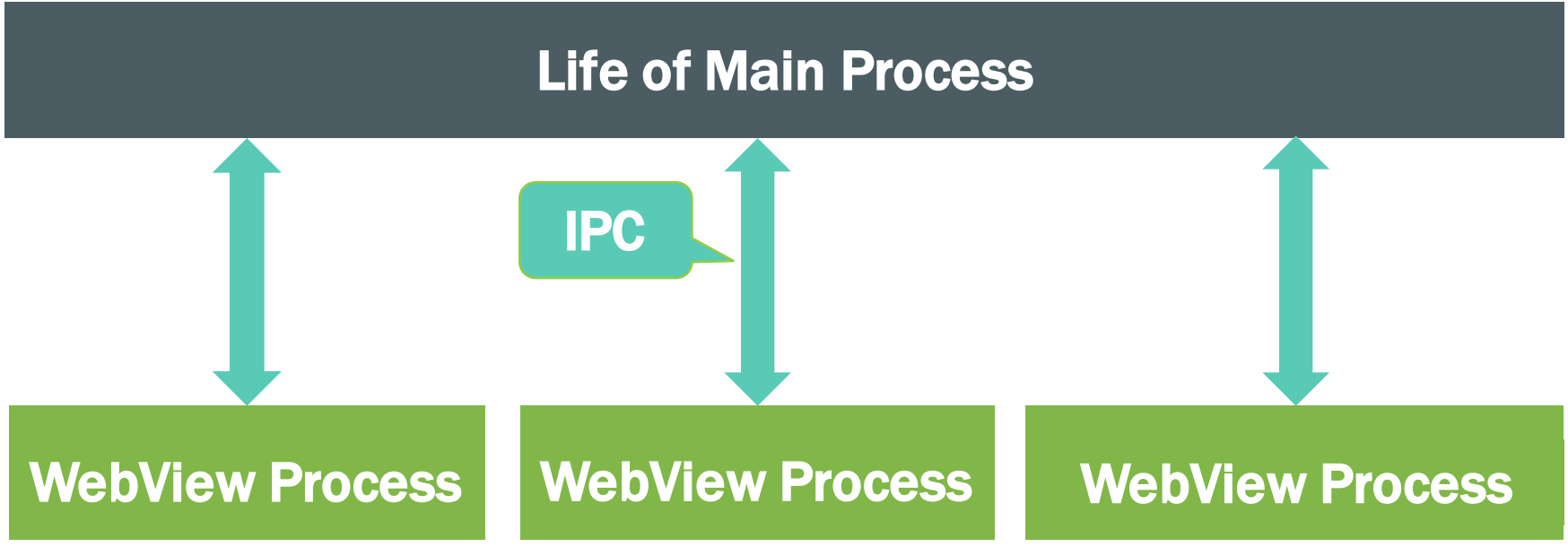
# 3）注意有生命周期对象的注销



# 4）注意大胖子的及时回收 - Bitmap

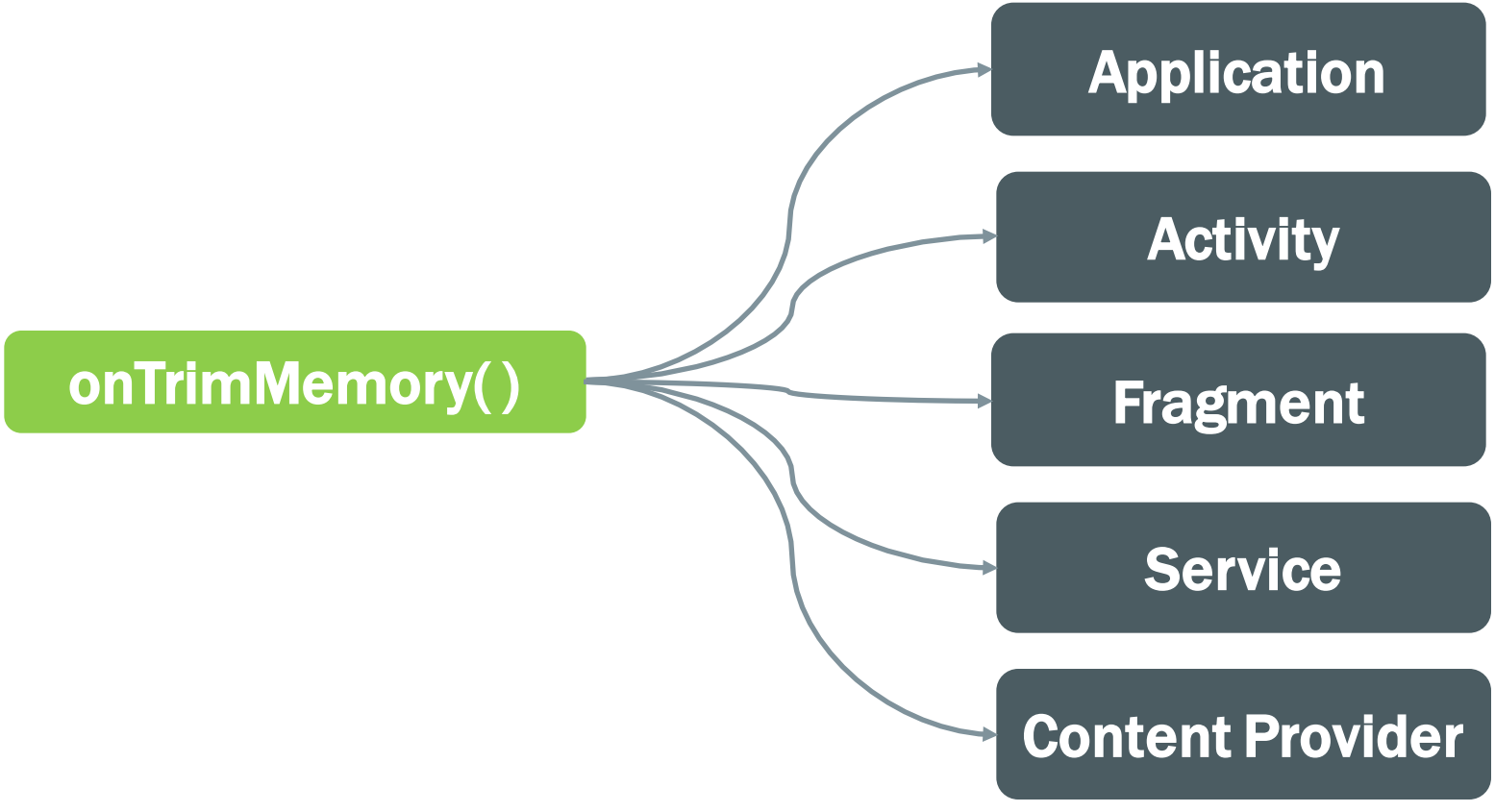


# 4）注意大胖子的及时回收 - WebView



# 4）注意大胖子的及时回收 - Cursor

# 5）onTrimMemory()与onLowMemory()



# 5）onTrimMemory()与onLowMemory()

TRIM_MEMORY_RUNNING_MODERATE
TRIM_MEMORY_RUNNING_LOW
TRIM_MEMORY_RUNNING_CRITICAL

TRIM_MEMORY_UI_HIDDEN

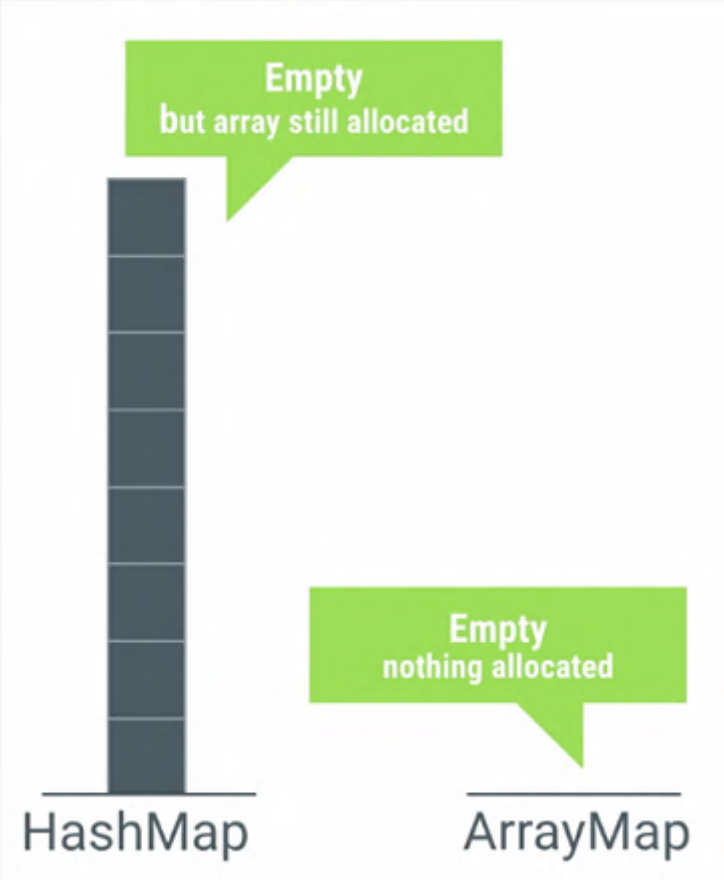
TRIM_MEMORY_BACKGROUND
TRIM_MEMORY_MODERATE
TRIM_MEMORY_COMPLETE

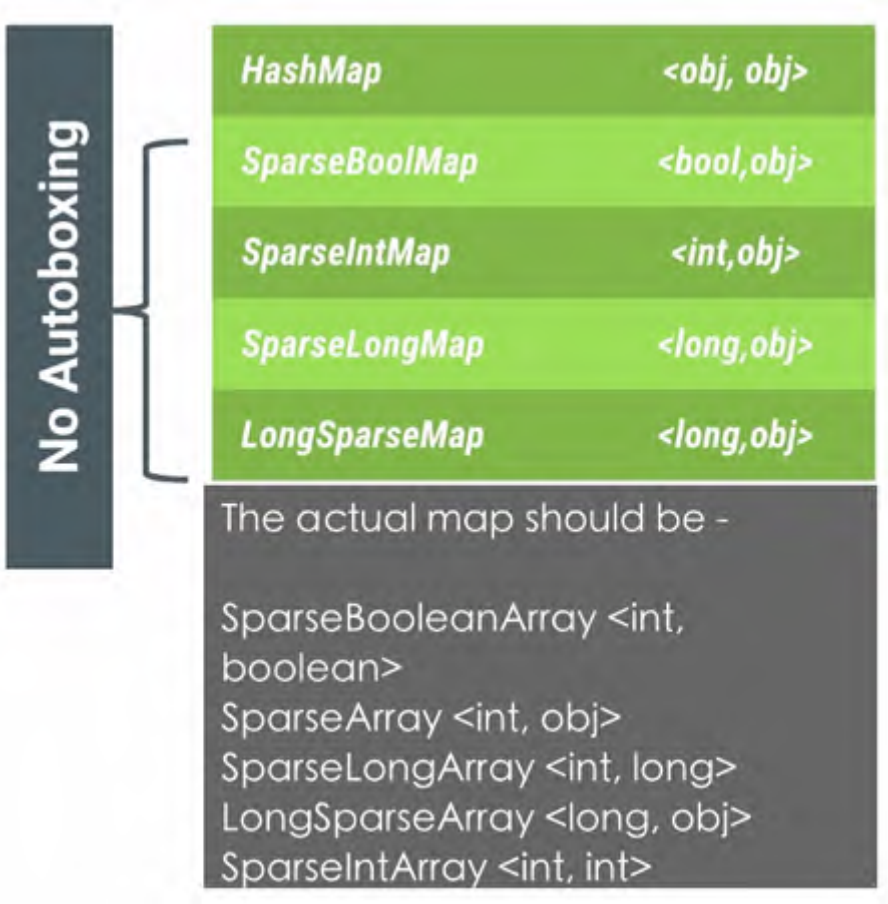
**onLowMemory() for sub-API Level 14**

4

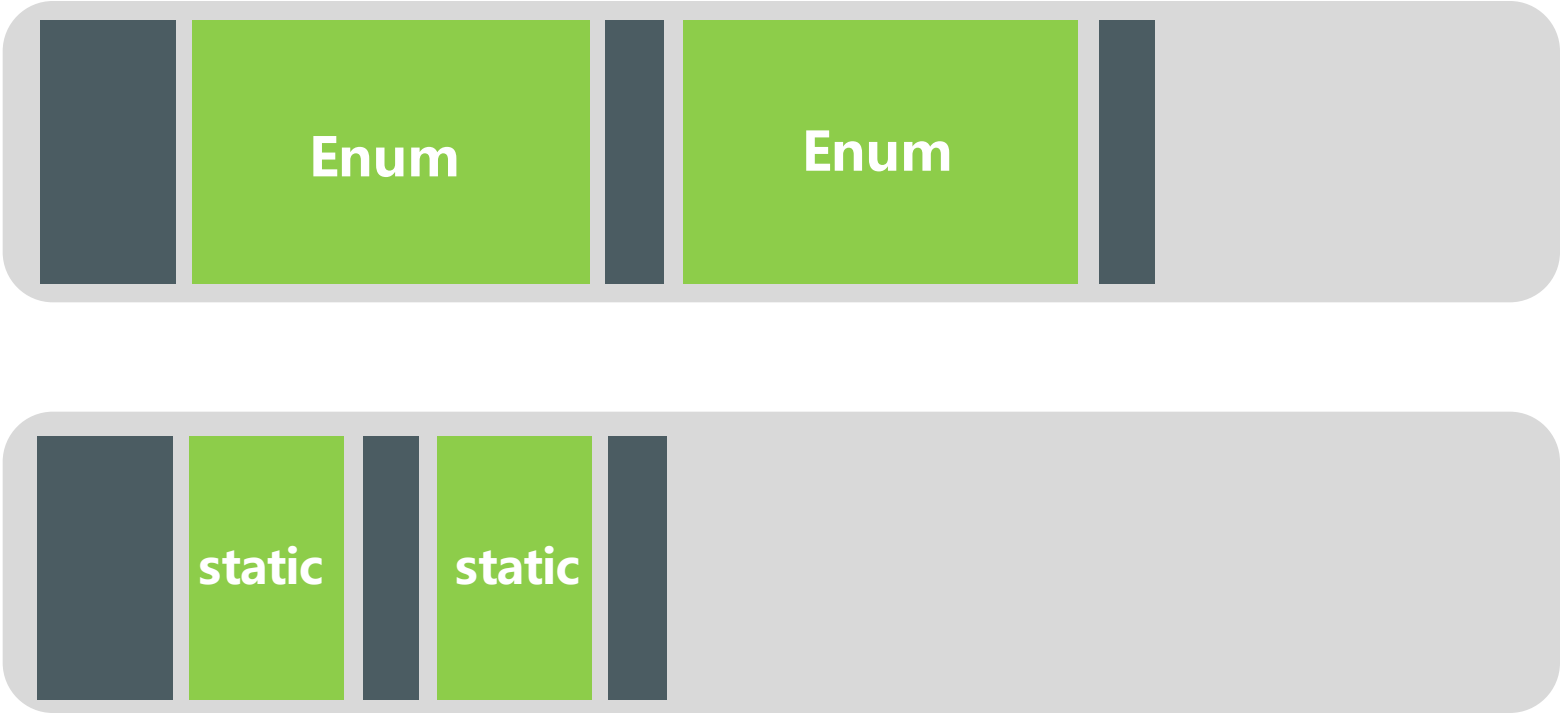
# Refactor

# 1）使用优化过后的数据结构

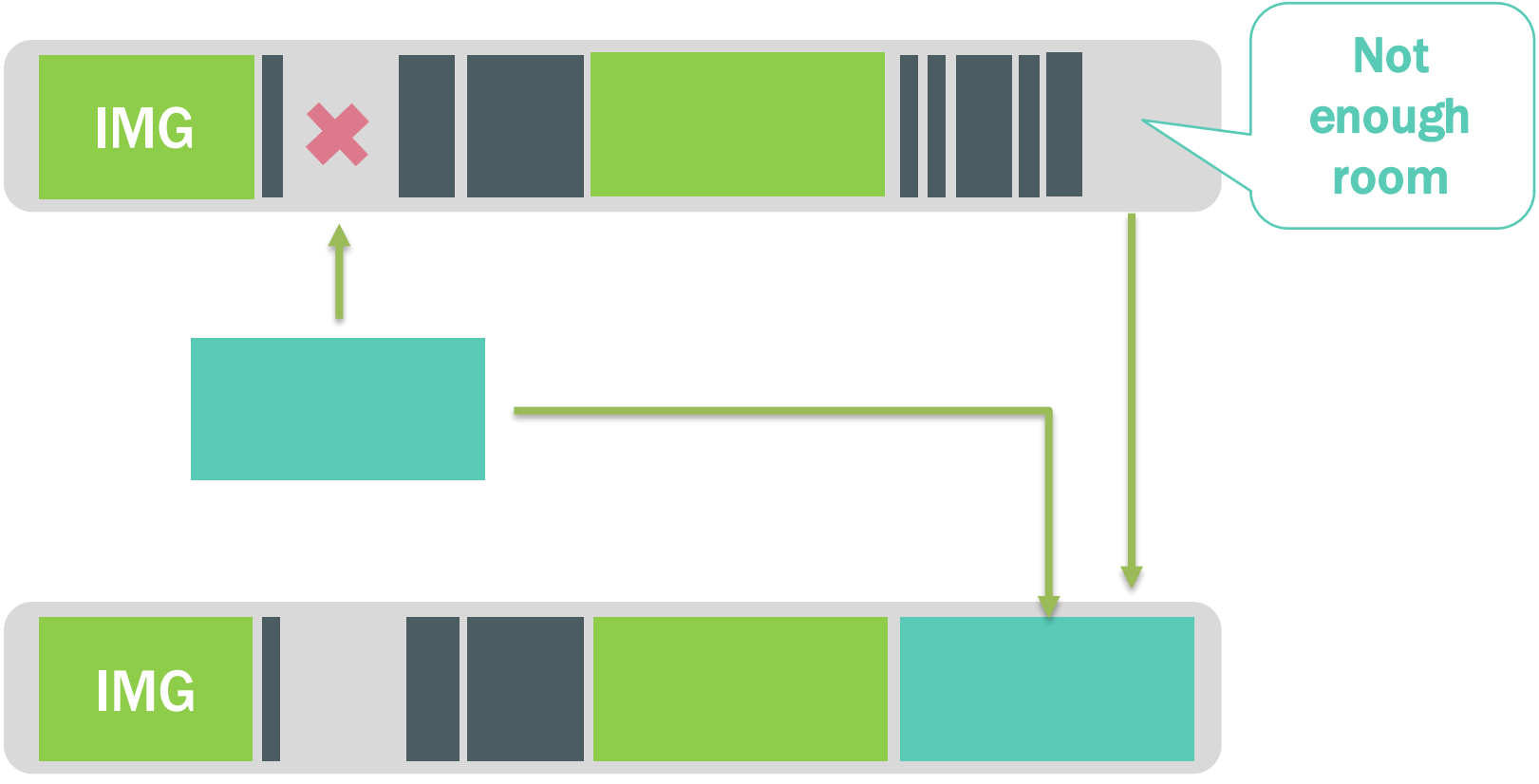


# 1）使用优化过后的数据结构



# 2）减少内存的碎片化



# 3）优化布局，减少内存消耗

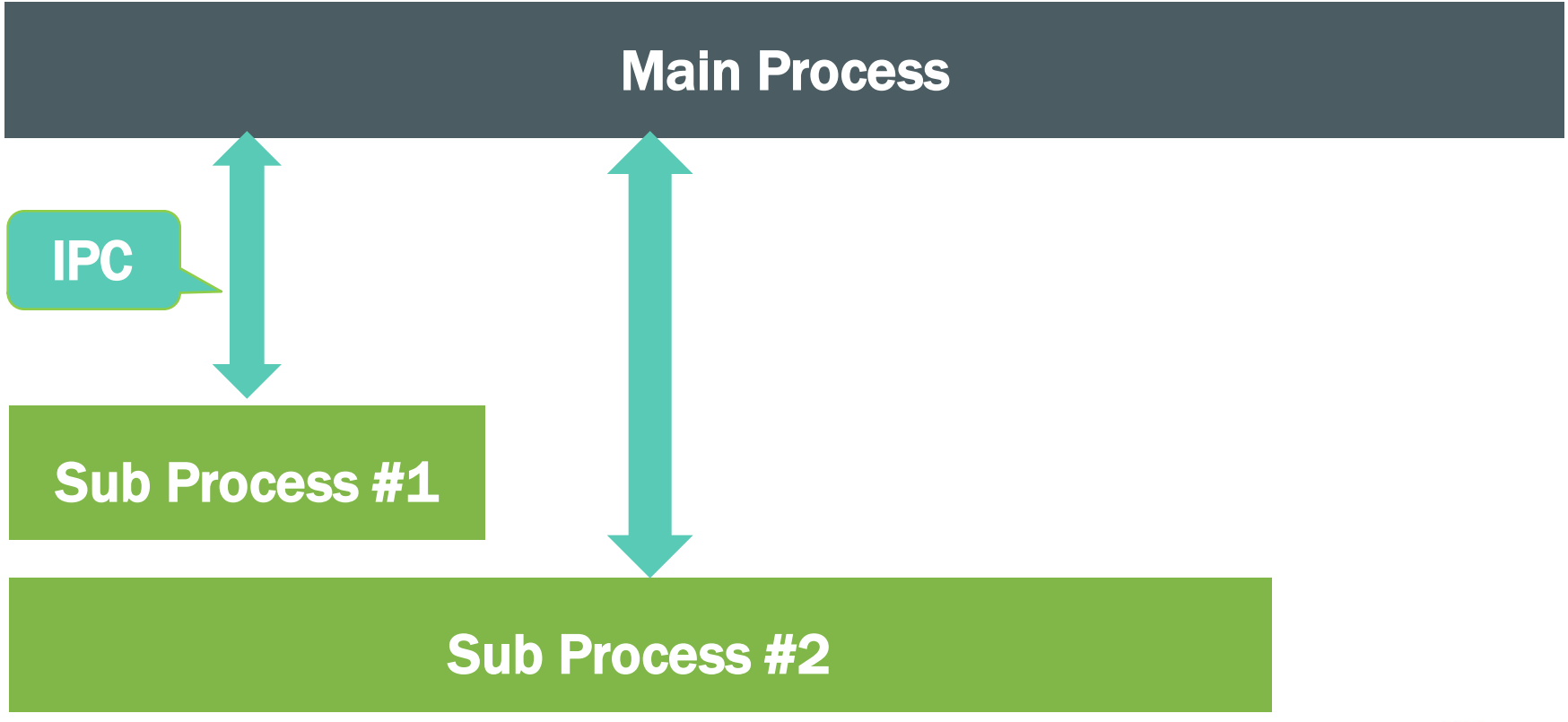
5

# Revalue

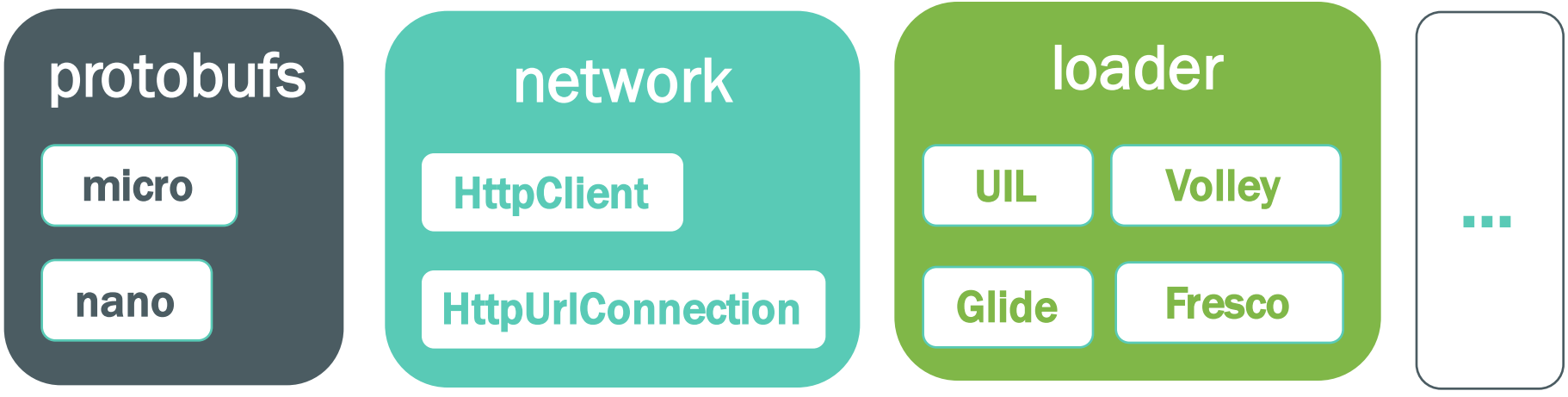
# 1）谨慎使用largeHeap

1）More GC Time
2）More Terminated

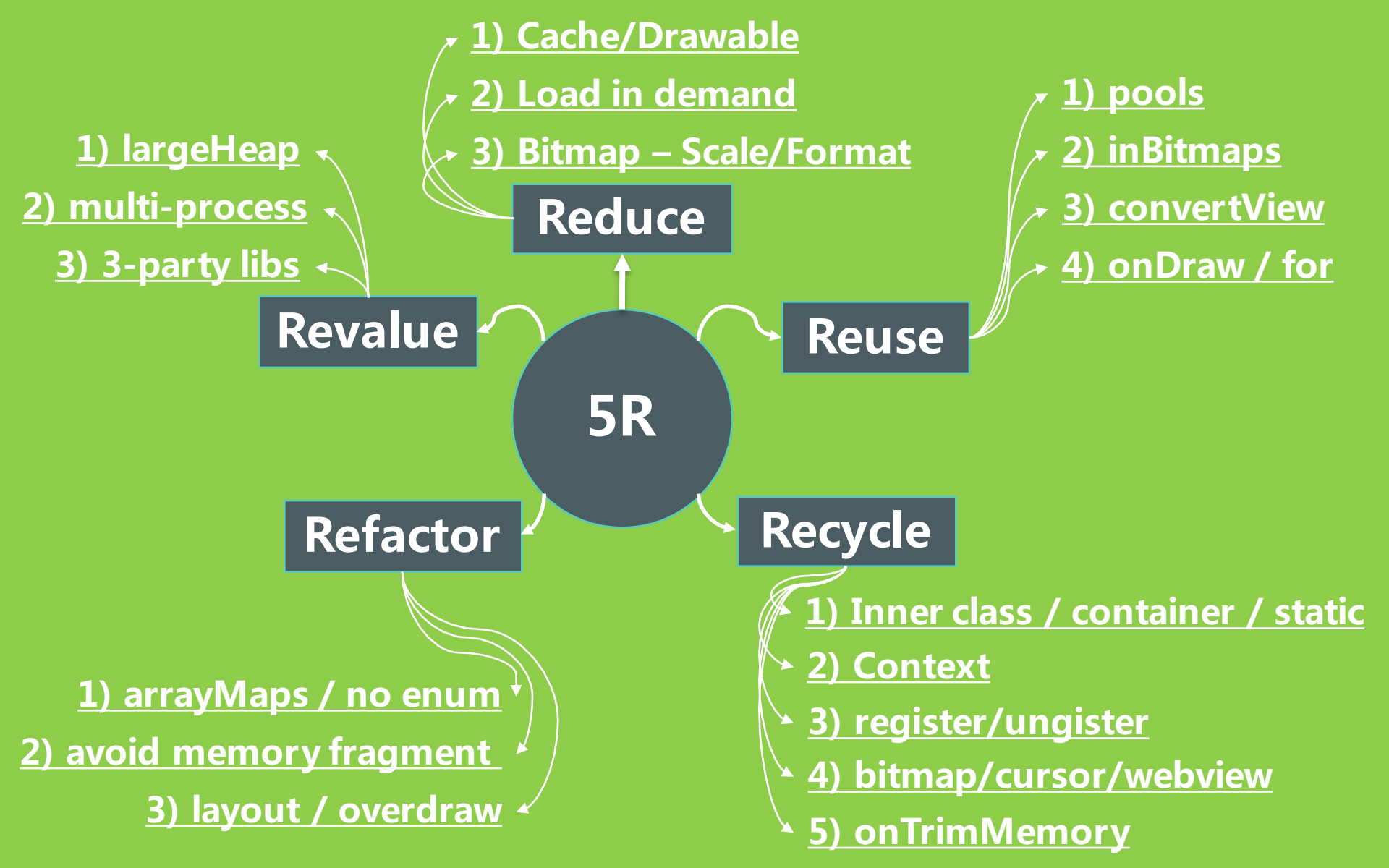
# 2）谨慎使用多进程



# 3）谨慎使用第三方libraries



# Q & A


1) Cache/Drawable
2) Load in demand
3) Bitmap – Scale/Format
Reduce
1) largeHeap
2) multi-process
3) 3-party libs
Revalue
1) pools
2) inBitmaps
3) convertView
4) onDraw / for
Reuse
5R
Refactor
1) arrayMaps / no enum
2) avoid memory fragment
3) layout / overdraw
Recycle
1) Inner class / container / static
2) Context
3) register/ungister
4) bitmap/cursor/webview
5) onTrimMemory

# Reference


- Google I/O 2011: Memory management for Android Apps
- Managing Your App’s Memory
- Avoiding memory leaks
- Wed_230_400_Putting Your App on a Memory Diet, Parts I and II_Murphy
- Context, What Context?
- WebView causes memory leak - leaks the parent Activity
- Android内存优化之OOM
- Android性能优化典范 - 第3季
- Android性能优化典范 - 第2季
- Android性能优化典范
- Android性能优化之内存篇

# Thanks