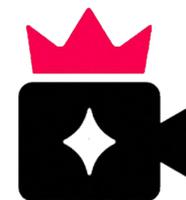


个人介绍

10110110001010111000101110100111000110100

柯灵杰 (lingtonke)

- 腾讯QQ空间视频项目组iOS技术负责人
- 腾讯学院认证讲师，主讲课程《设计模式》
- 参与过QQ、QQ空间、企鹅MV、微云等多个项目开发
- 主要从事视频、图像、特效相关的工作
- 数项国家发明专利第一发明人



Tencent 腾讯

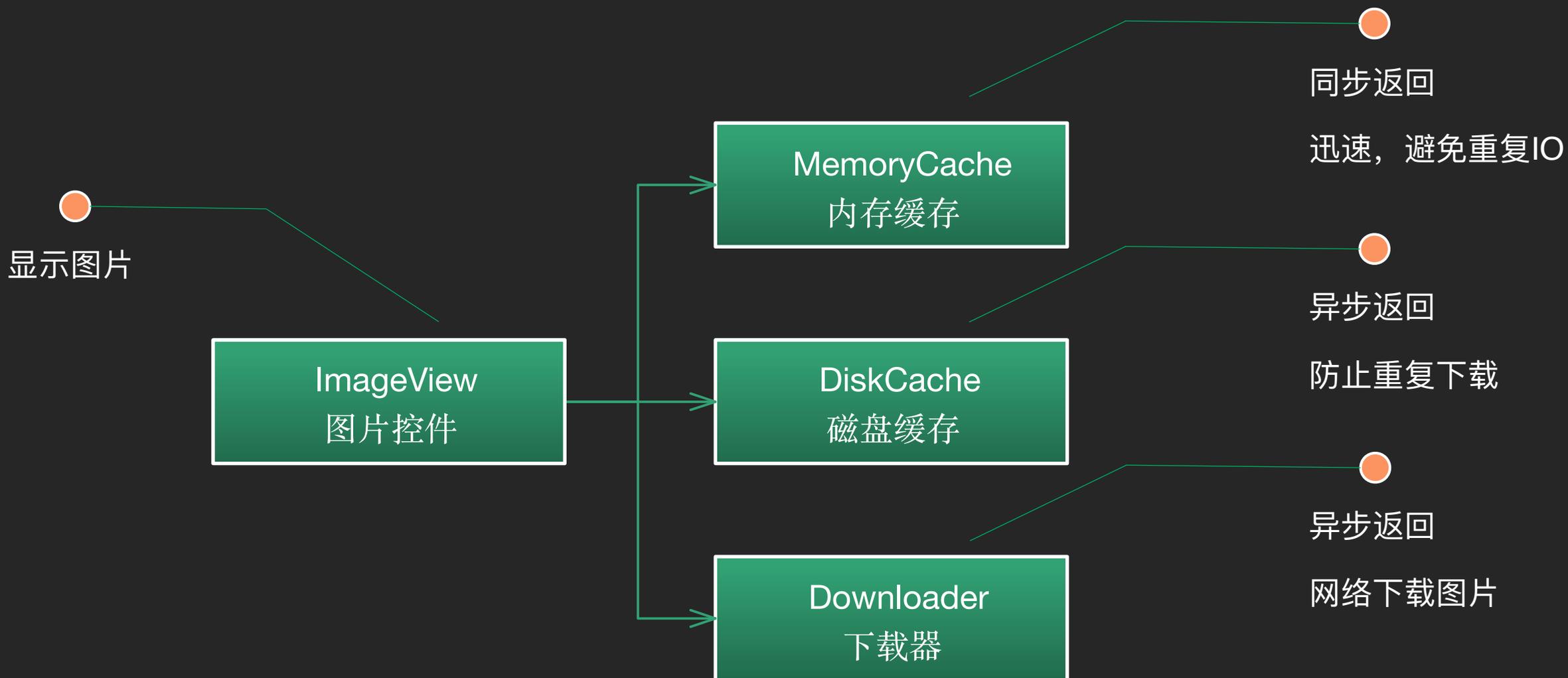


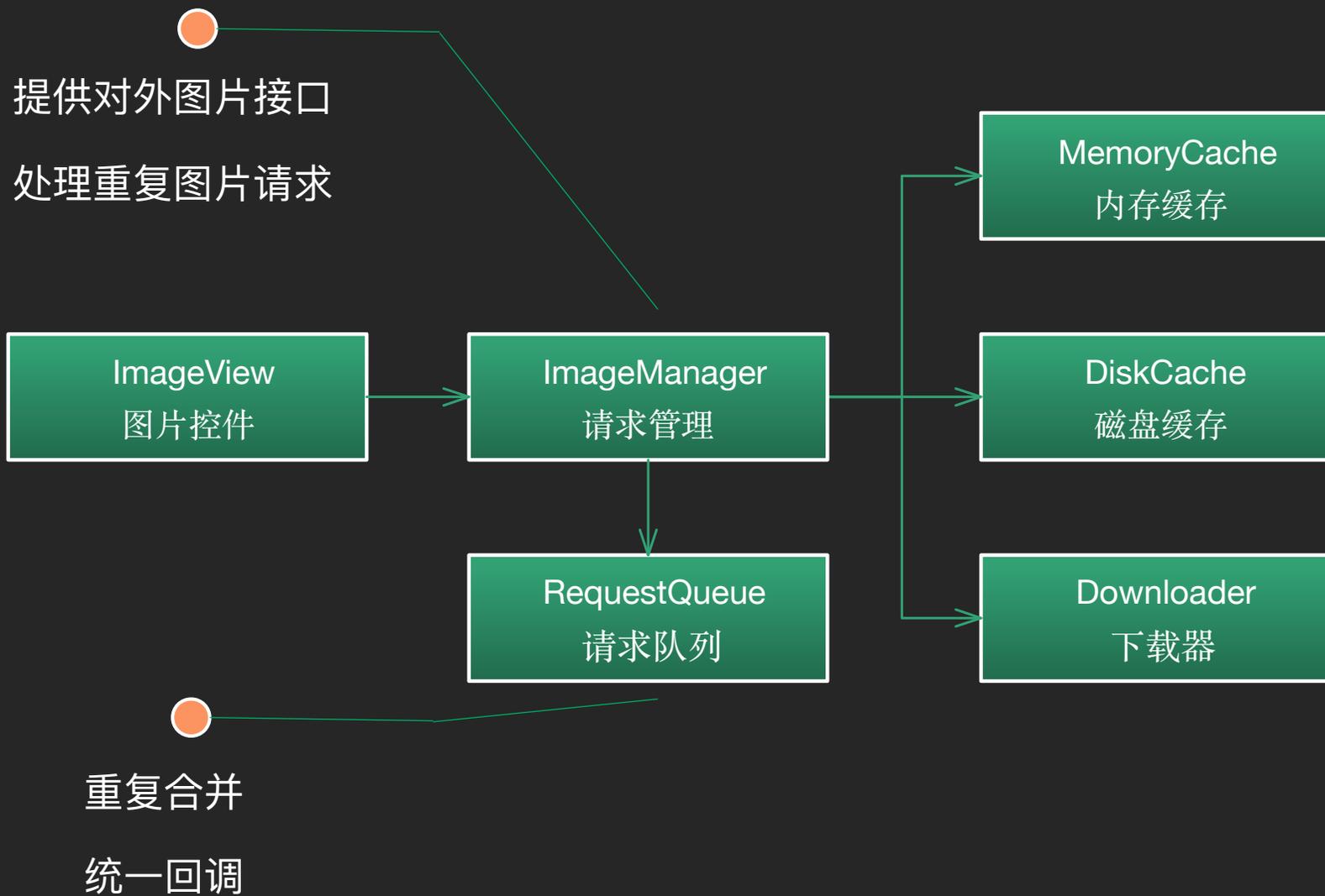
打造易扩展的 高性能图片组件

柯灵杰 (LingtonKe)

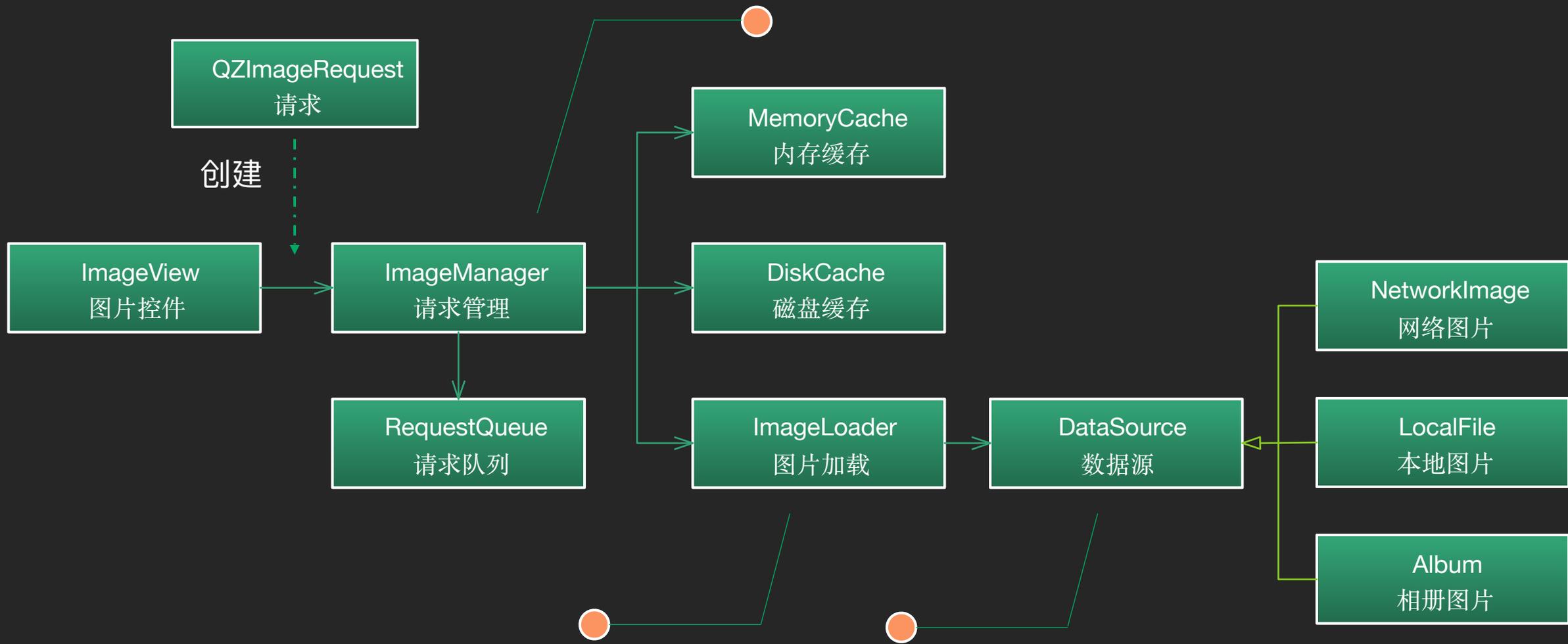
01

从0开始
打造易扩展的架构



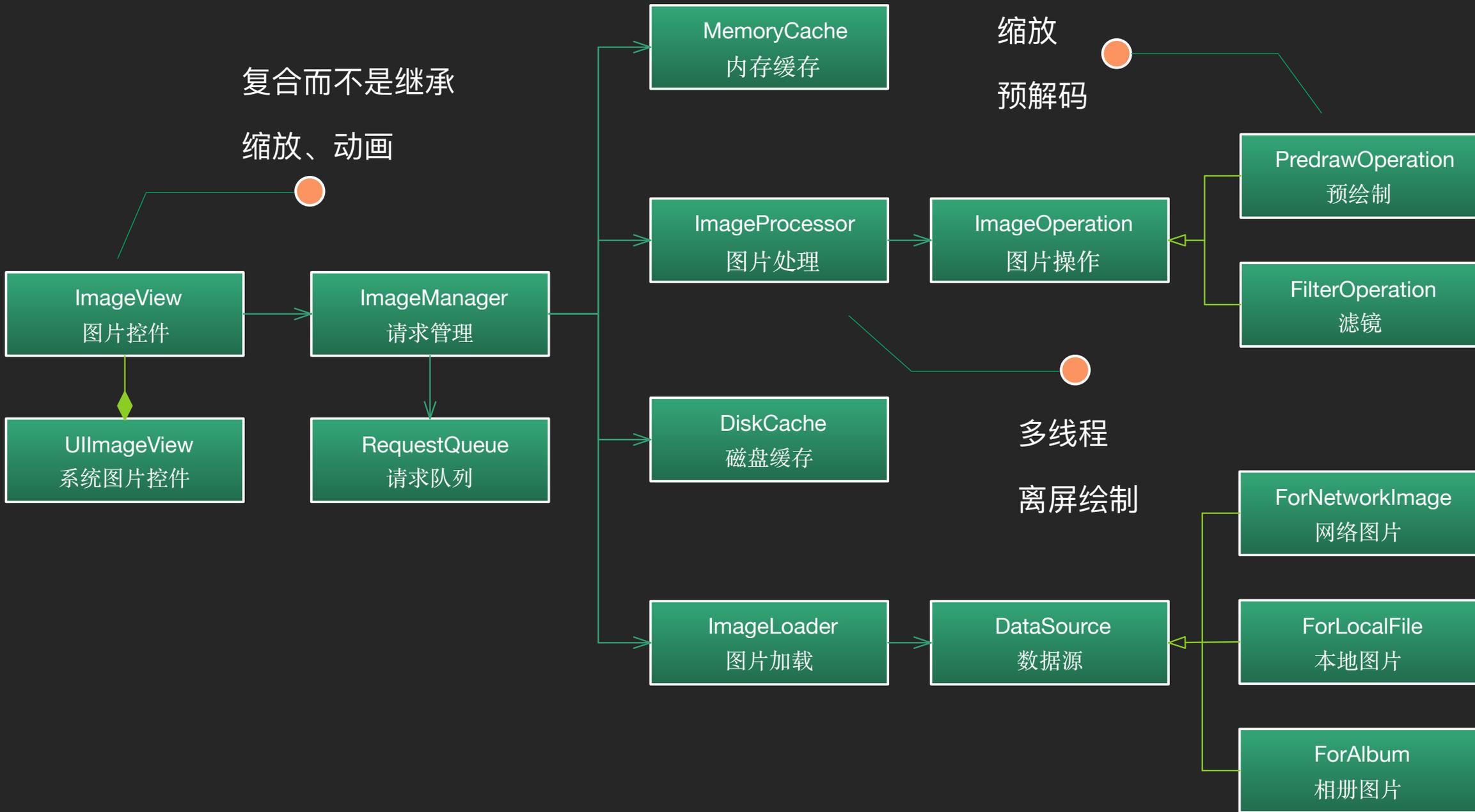


控制不同缓存策略



责任链模式

提供统一接口





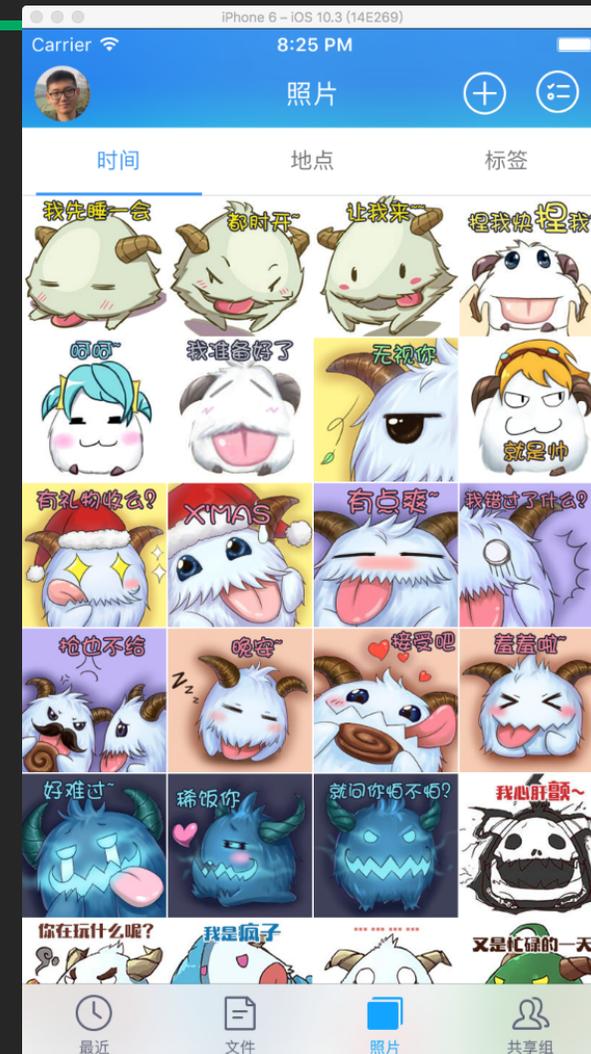
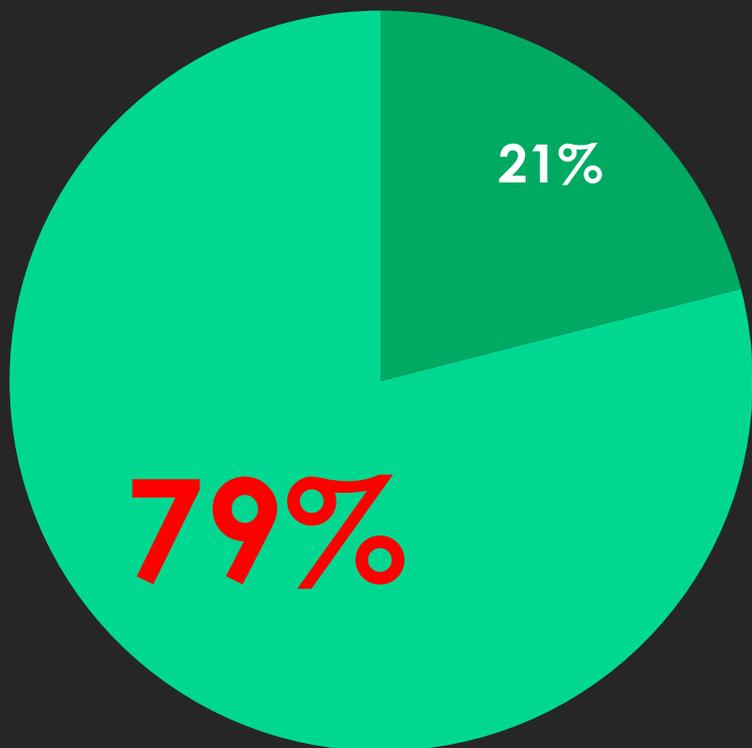
02

向更高的性能前进

渲染性能优化

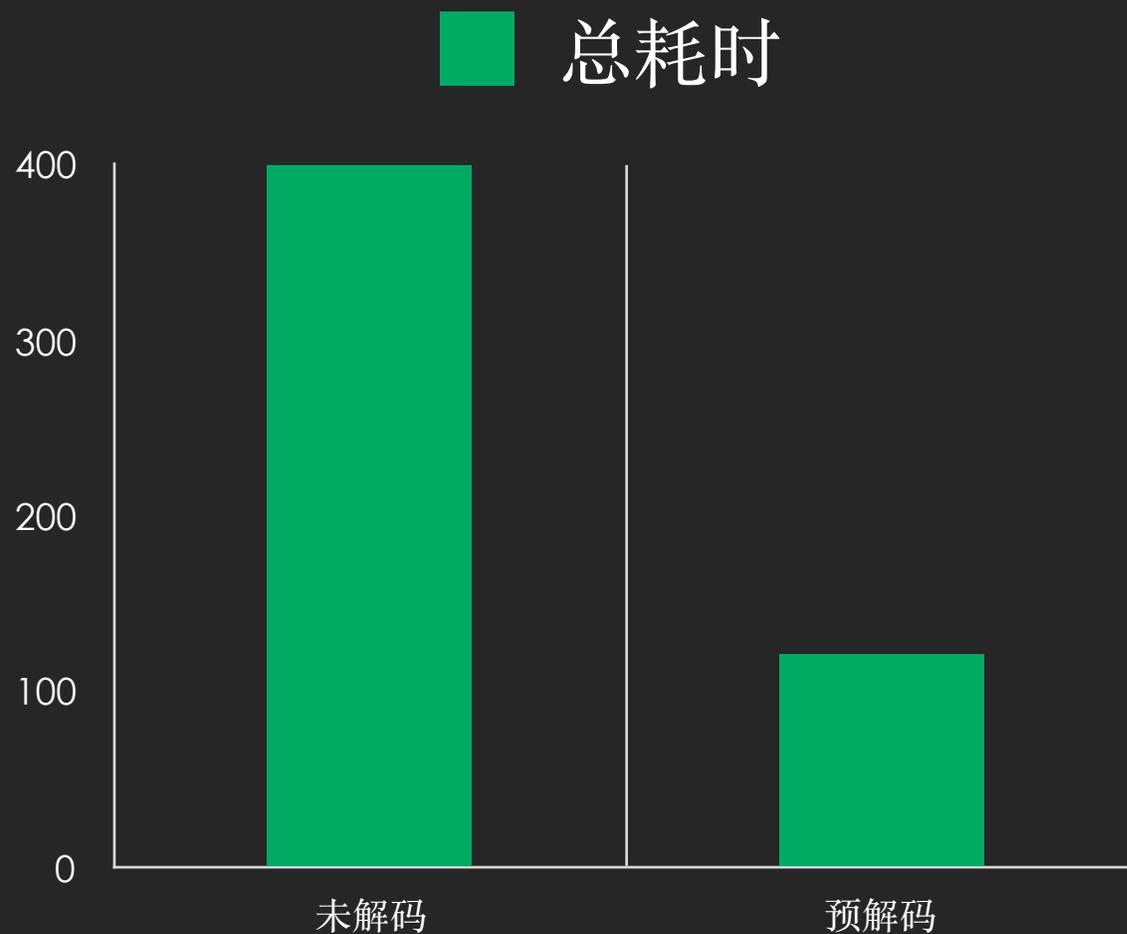
谁吃掉了我们的CPU?

- CA::Render::create_image_from_provider



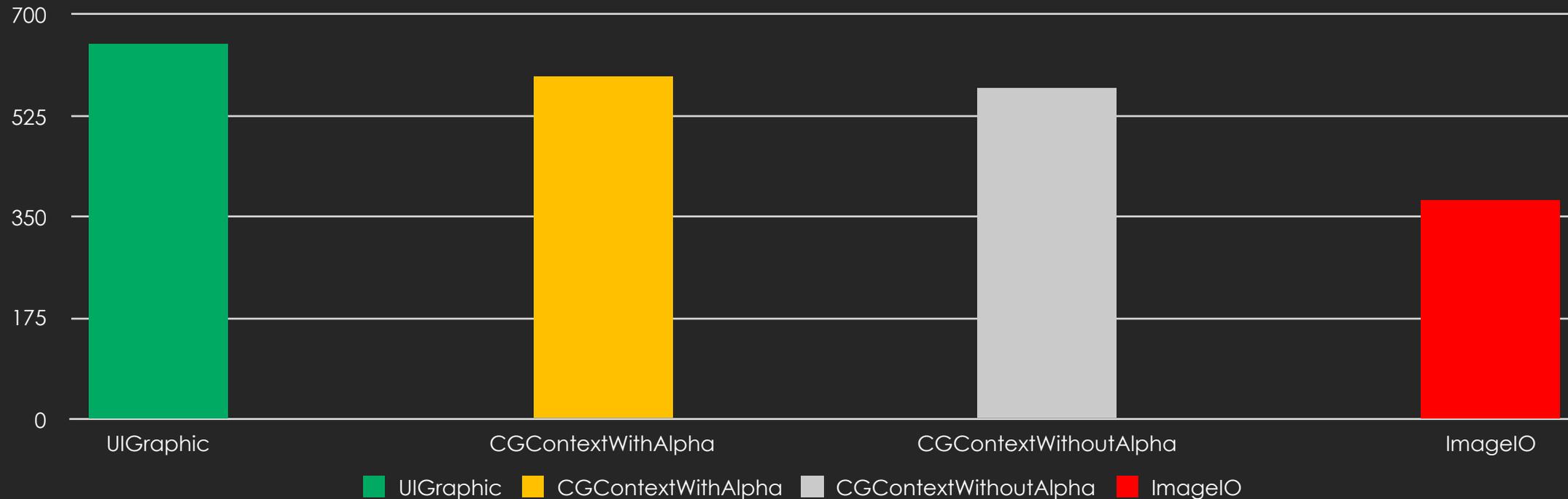
谁吃掉了我们的CPU?

- 效果明显
- 大幅提高ScrollView流畅度



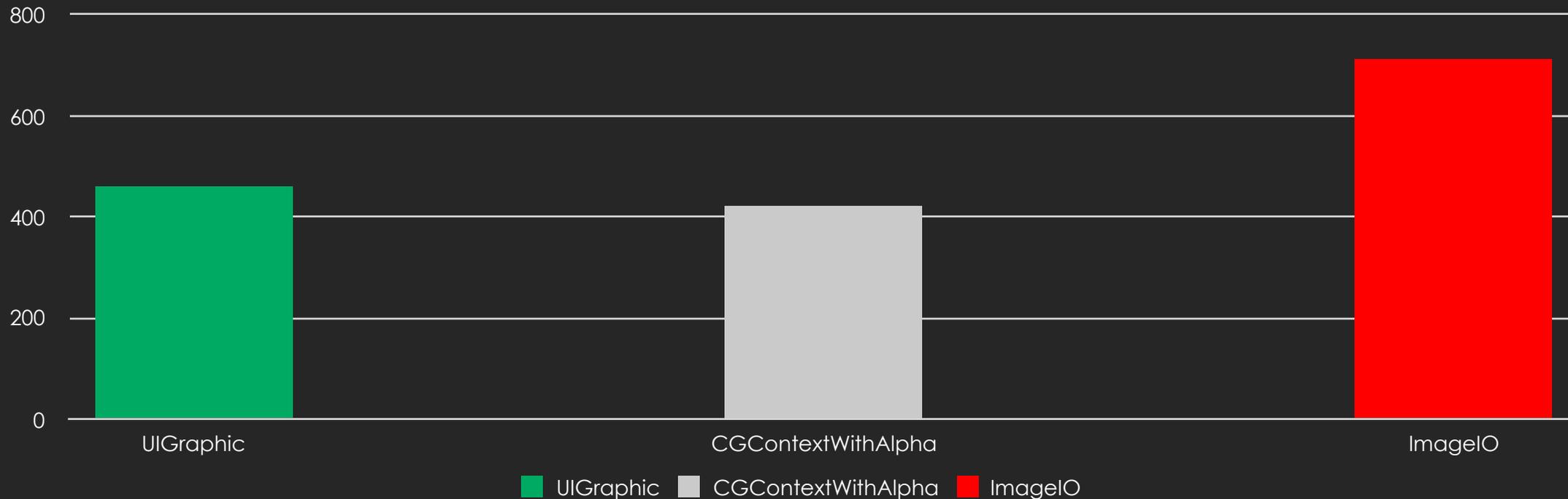
解码性能深入分析

50张1000*1000图片解码耗时



多线程解码

50张1000*1000图片解码耗时



内存占用优化

Physical
Memory



Wired

Active

Inactive

Used

Free

Wired



- 系统使用
- 不能借用
- Allocations 不可见

Active



- 正在运行的程序使用
- 内存不足时页面置换

Inactive



- 不在运行的程序使用
- 加快启动速度
- 内存不足优先回收

Free



- 空闲内存
- 优先被使用

占用内存的3种方式

- ImageIO和其他绘制方法
- UIImageView
- 字节对齐copy
- Active Memory
 - (CG Raster Data)
- Wired Memory
- Wired Memory
 - (动画结束释放)

字节对齐

Active
Memory

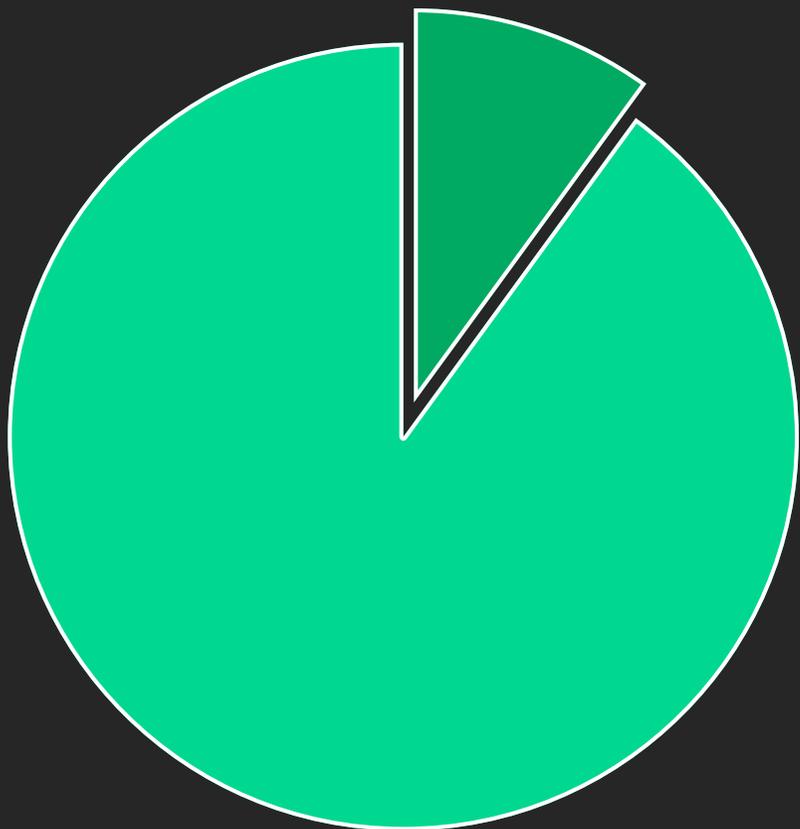


Wired
Memory



CA::Render::copy_image

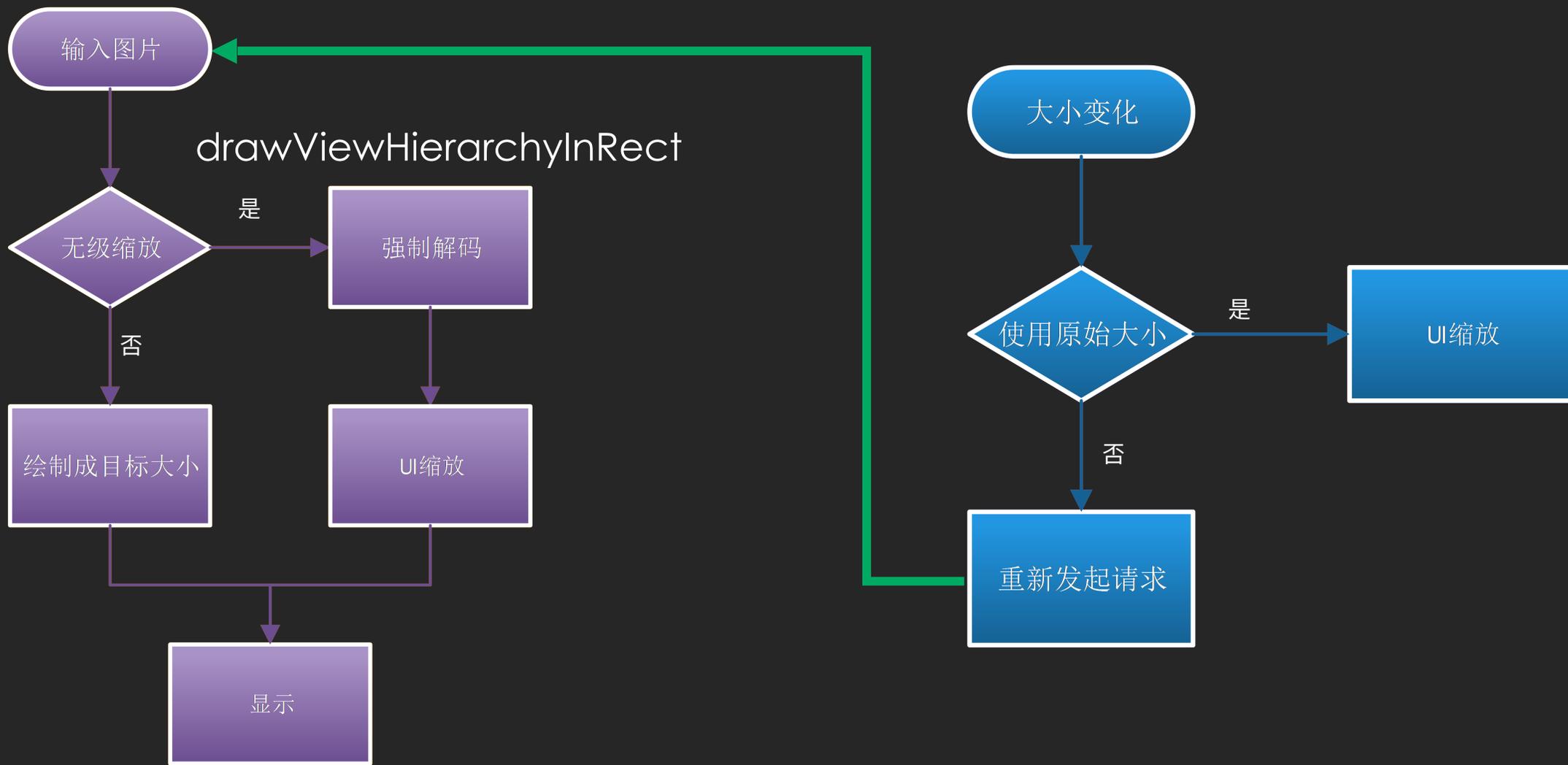
图片显示区域是否变化



● 有变化 ● 无变化

绘制成需要的大小

- ✓ 大部分图片显示区域不会变化
- ✓ 省去了缩放的运算开销
- ✓ 节约了大量内存空间
- ◆ 不支持无级缩放



缓存优化

内存缓存

使用C++提高性能

链表和哈希表的组合

插入、删除、查找 $O(1)$

淘汰策略

FIFO+LRU

内存限制

数量限制

模糊匹配

可以使用比请求图片更大的
缓存图片

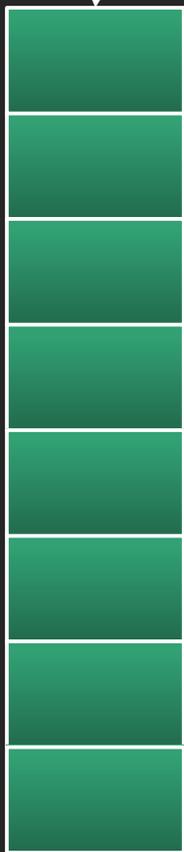


HashTable



新数据

FIFO



被访问
数据

LRU



被访问
数据

100 X100
Gray

100 X100
AspectFit

200 X200
ScaleToFill

末尾淘汰数据

末尾淘汰数据



THANK YOU!

柯灵杰 (LingtonKe)

Tencent 腾讯