

移动端SDK优化的特点与经验分享

极光云服务技术中心 王可为

2017/5/26

目录

Part

01

SDK与APP的差别

- 面向开发者vs面向用户

Part

02

极光SDK的架构优化

- 抽象、分层设计

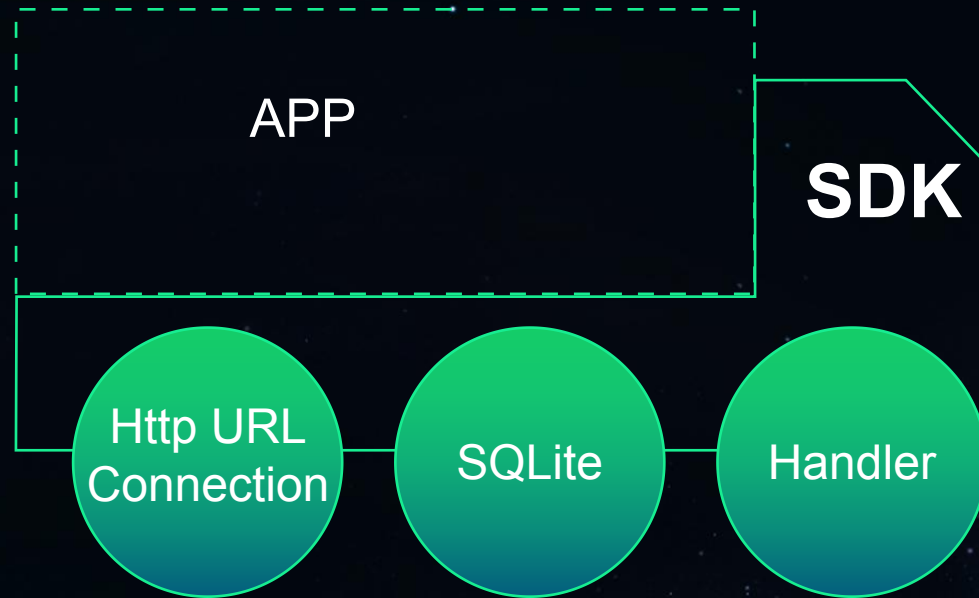
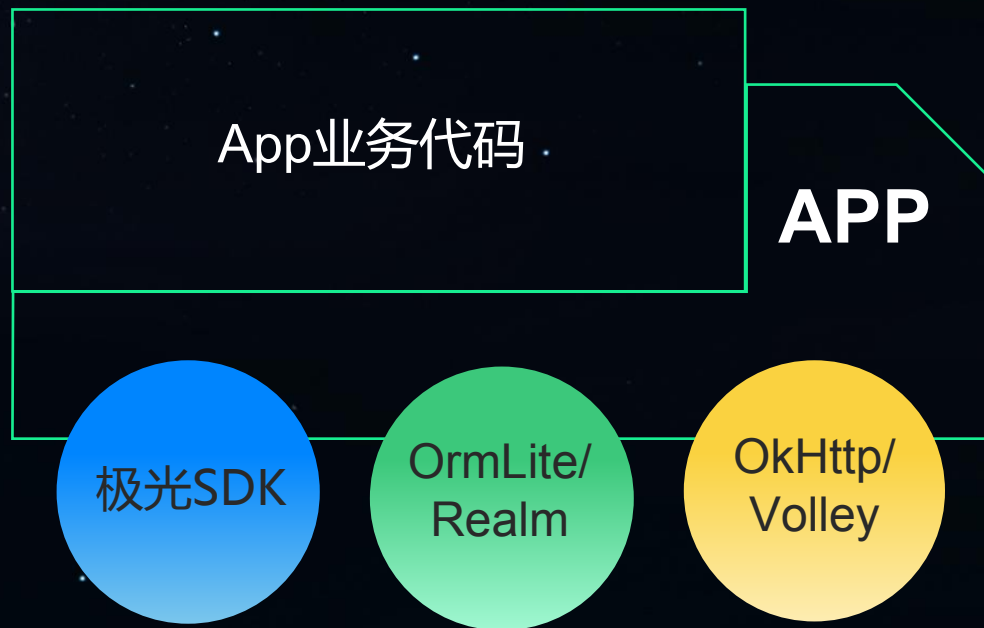
Part

03

极光SDK的性能优化

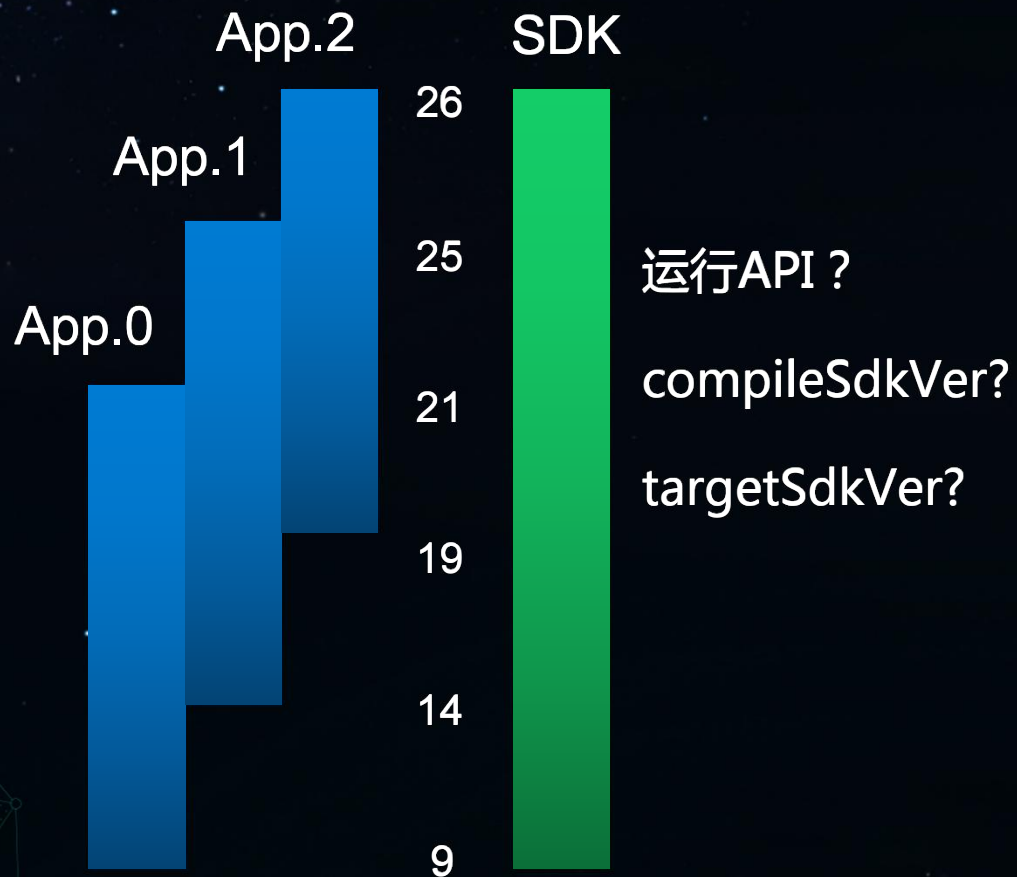
- 多进程、I/O、网络..

SDK与APP的区别——重复造轮子



保持代码精简，并且内部可见，方便调试和修改

SDK与APP的区别——全面兼容



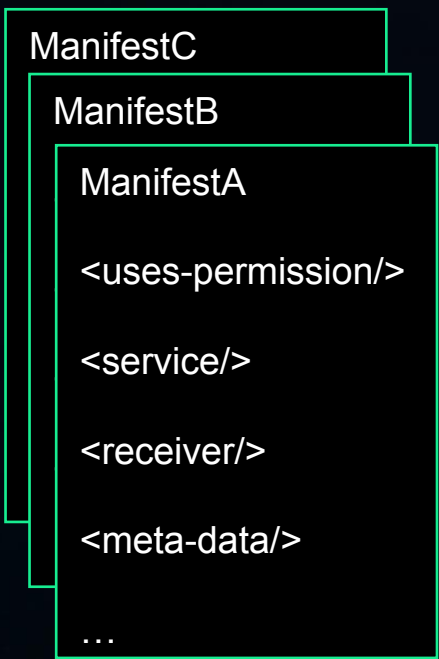
不同Rom、不同平台的兼容措施

SDK与APP的区别——配置

APP开发者根据需要写一份AndroidManifest

```
<uses-permission/>  
<service/>  
<receiver/>  
<meta-data/>  
...  
...
```

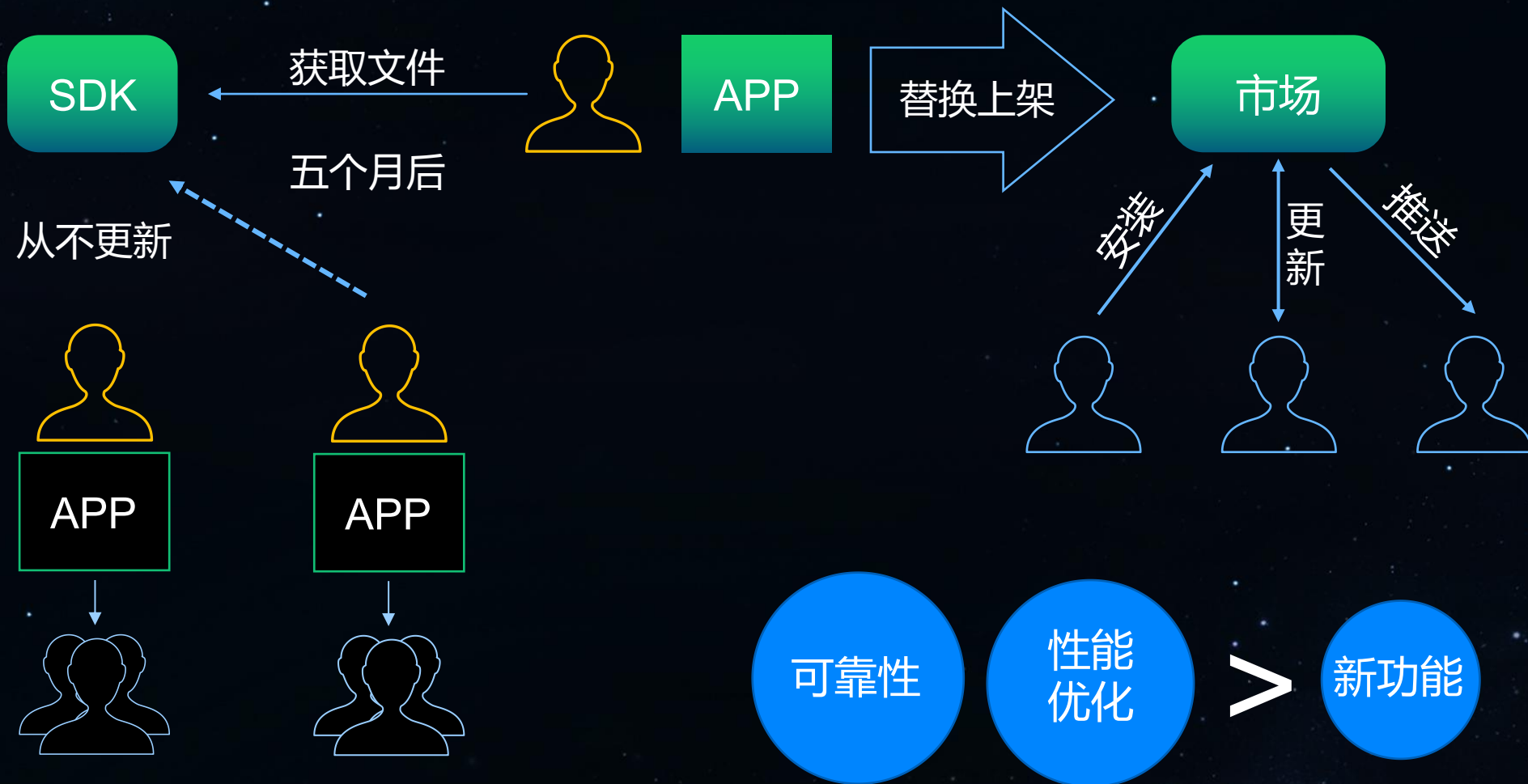
SDK开发者只能指导开发者配置



jcenter

自动Demo

SDK与APP的区别——升级



极光SDK的架构优化——旧架构



...

极光短信

极光统计

极光分享

JPush

JMessage

Network

Network

Protocol

Protocol

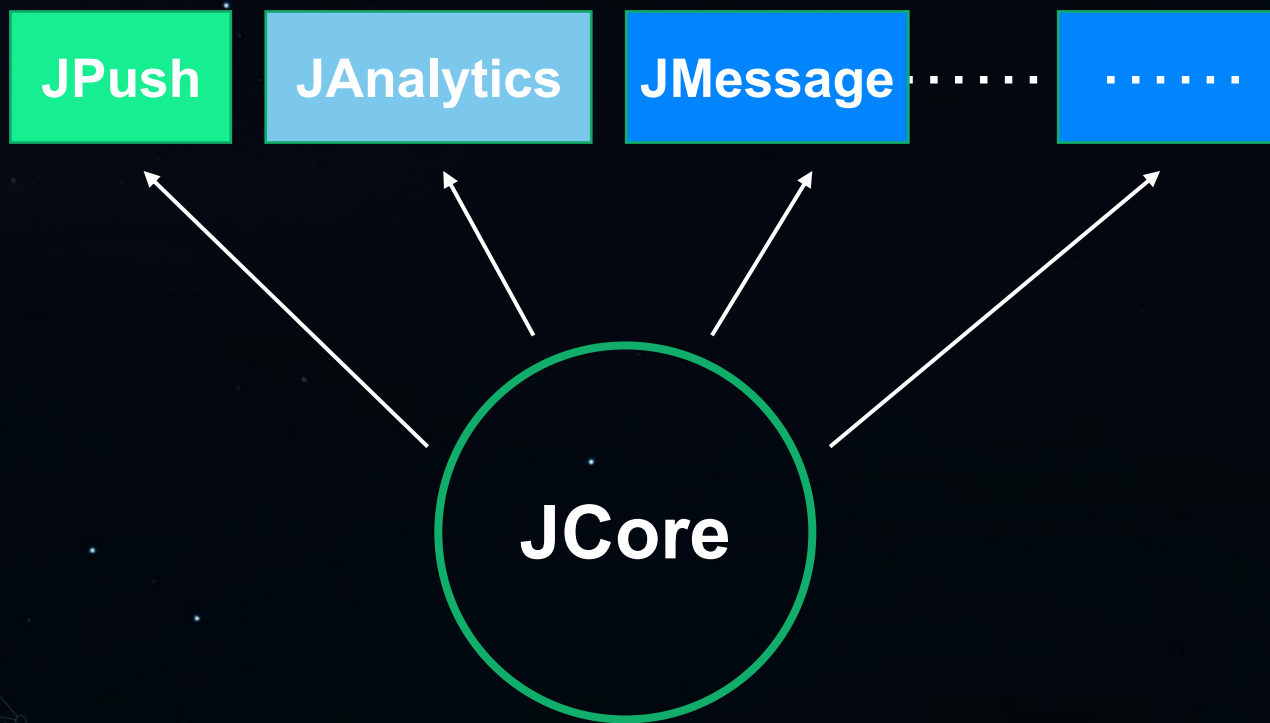
Thread

Thread

推送和IM是独立的SDK，存在很多冗余代码

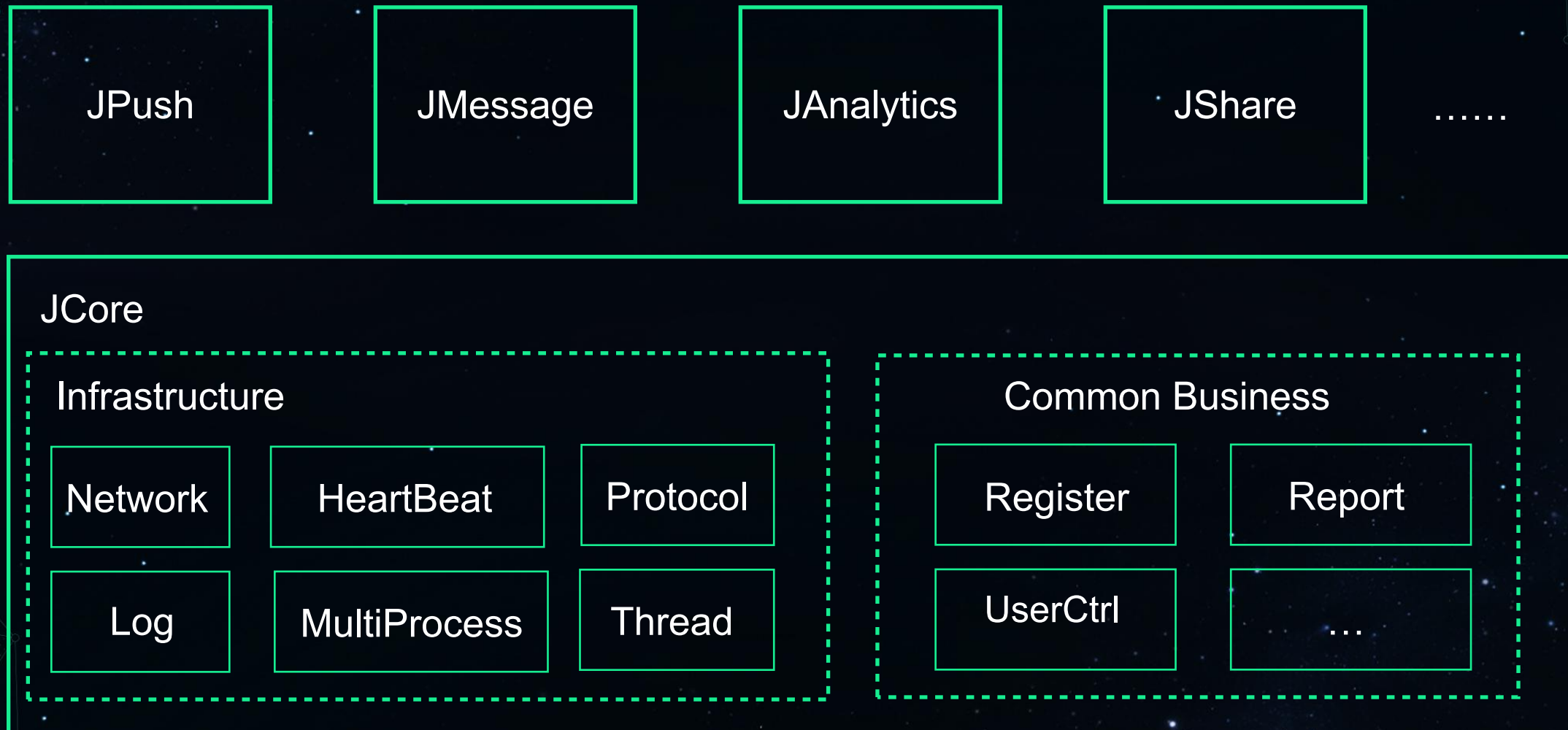
1. 占用空间大
2. 重复的操作
3. 占用通道和线程资源
4. 冗余代码升级管理麻烦

极光SDK的架构优化——新架构

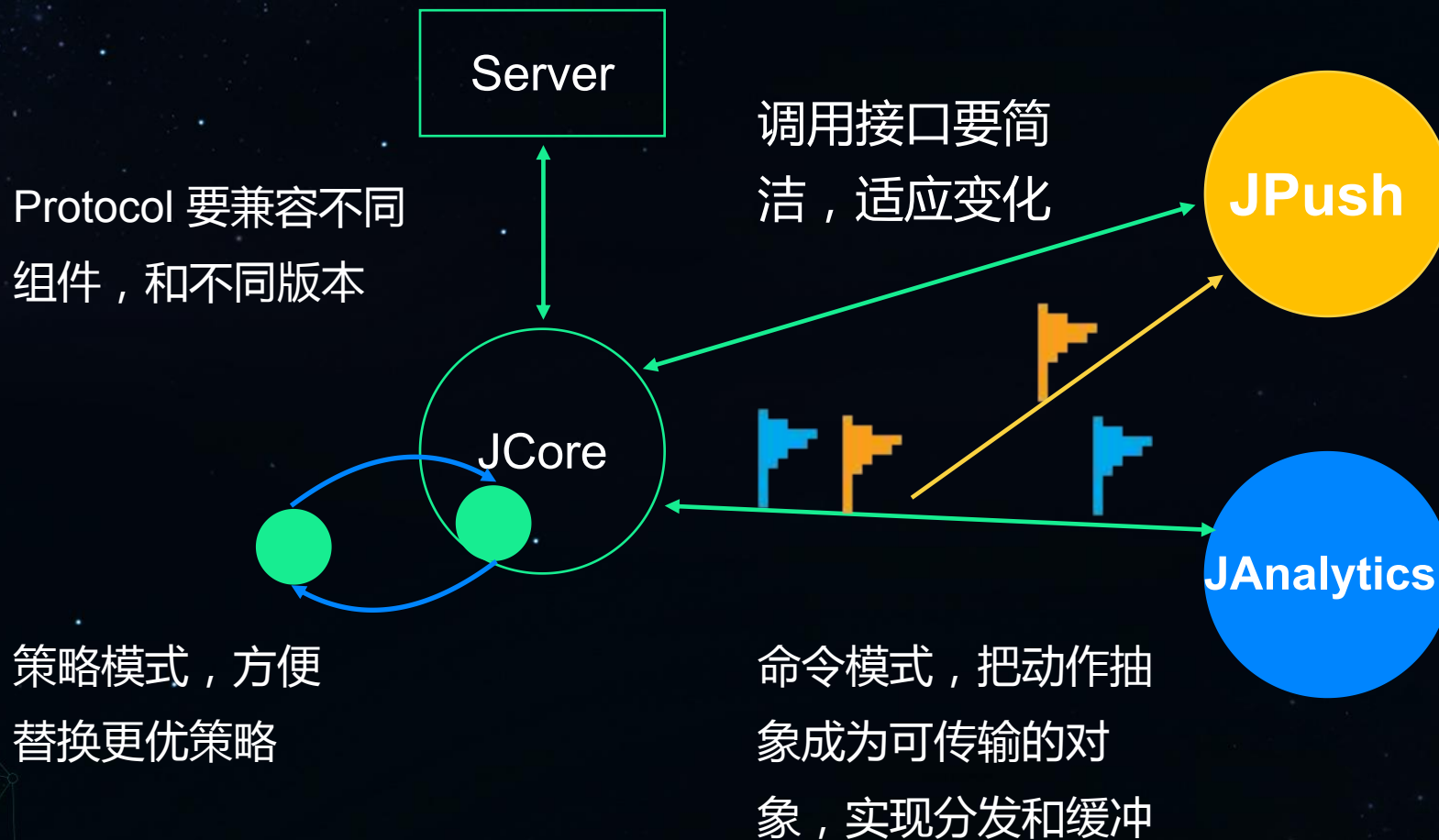


JCore负责核心通用的功能，上层SDK各自在JCore之上运行自有业务
结构更加清晰，利于扩展
共享资源，减少重复动作
针对性做基础优化更加方便

极光SDK的架构优化——新架构



极光SDK的架构优化——实际开发中需要考虑的

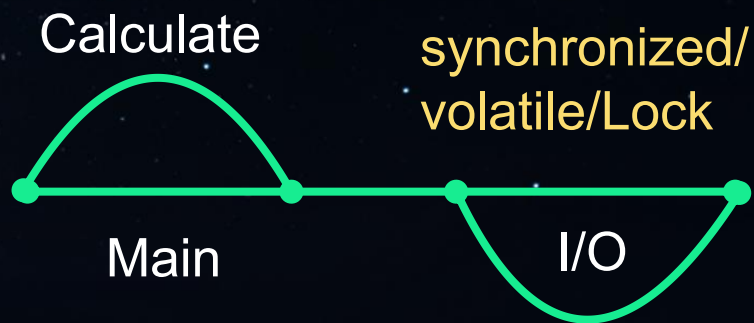


Jenkins+Gradle

构建产生文件

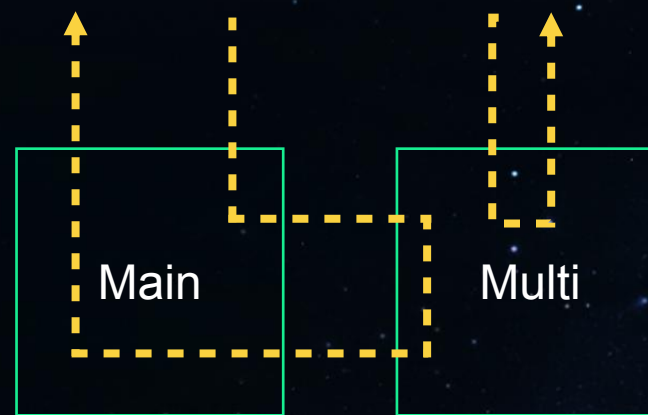
- 📄 jcore-obfuscate.zip
- 📄 iniObj.zip
- 📄 jpush-android-1 min-3.0.0.zip
- 📄 jpush-android-30s-3.0.0.zip
- 📄 jpush-android-debug-3.0.0.zip
- 📄 jpush-android-release-3.0.0.zip
- 📄 jpush-obfuscate.zip

极光SDK的性能优化——多进程与多线程



核心问题：数据不同步（线程/进程安全）

- 单例模式 双重检查锁
- 变量在多进程时是不同内存区的
- 进程间通信 把动作放到指定进程



极光SDK的性能优化——多进程与多线程

存储方式的 响应级别

Memory(ns)
Disk(ms)
IPC(ms+)
Network(s)

SharedPreferences为例

for 100times

getXX
getXX
getXX
getYY
getYY

setXX
setYY

x = getXX
y = getYY

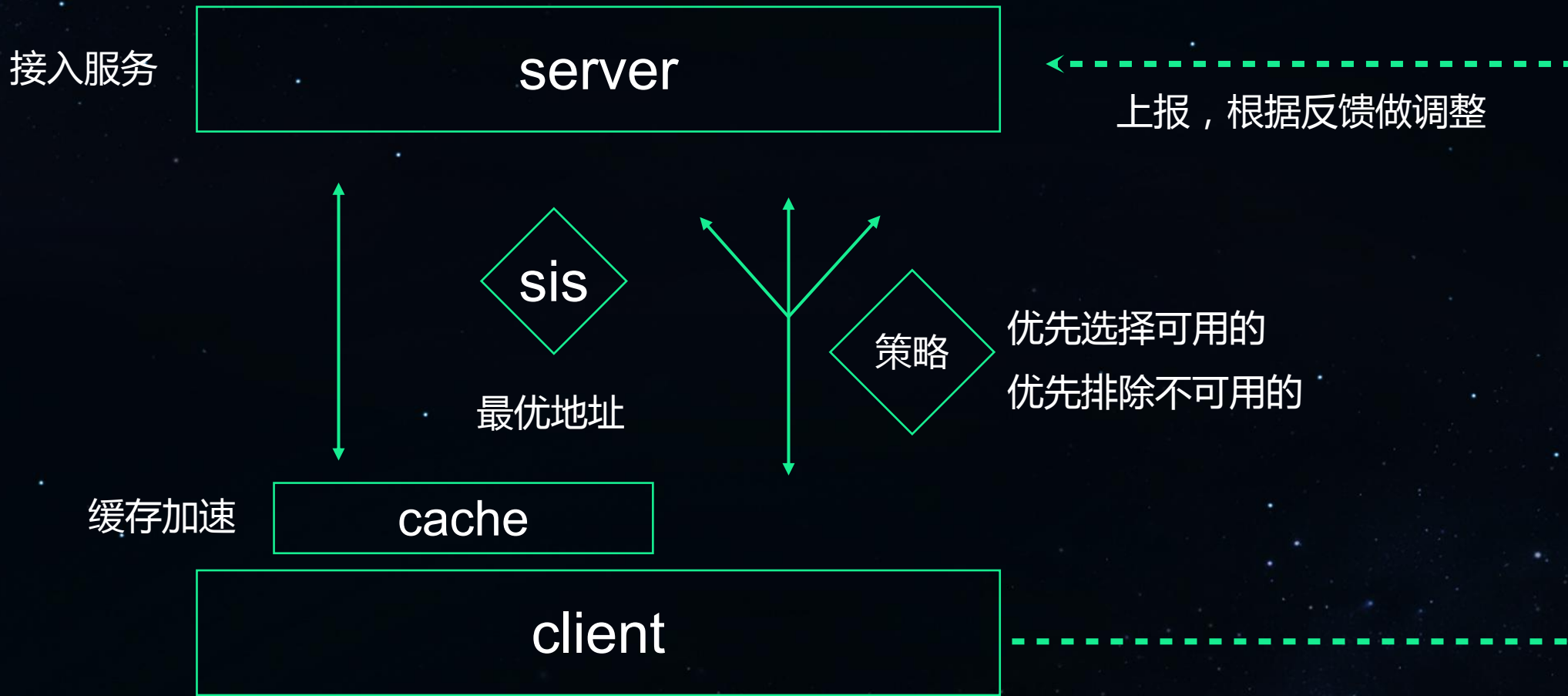
for 100times

x、x、x
y、y

setXY

1. 同线程一次读取多次使用
2. 写操作批量提交
3. 使用内存级别存储
4. 跨进程批量读取和提交
5. 拆分存储区

极光SDK的性能优化——长连接优化



END
T H A N K S