

极光 iOS SDK 架构 及性能优化经验分享

极光研发中心 沈孝星

2018/3/10

目录

Part 01

关于极光

- 规模
- 产品线
- 产品图谱

Part 02

SDK架构优化

- 旧架构
- 新架构

Part 03

性能优化

- 长连接
- 本地缓存

Part 04

APP开发的一些优化建议

- 思路

Part I - 关于极光

- 规模
- 产品线
- 产品图谱

关于极光-规模

25_万

25万开发者

60_万

60万款APP

110_亿

110亿移
动终端

8_亿

月独立活跃
设备

90%

90%覆盖率

关于极光-产品线



极光开发者服务



极光
数据服务



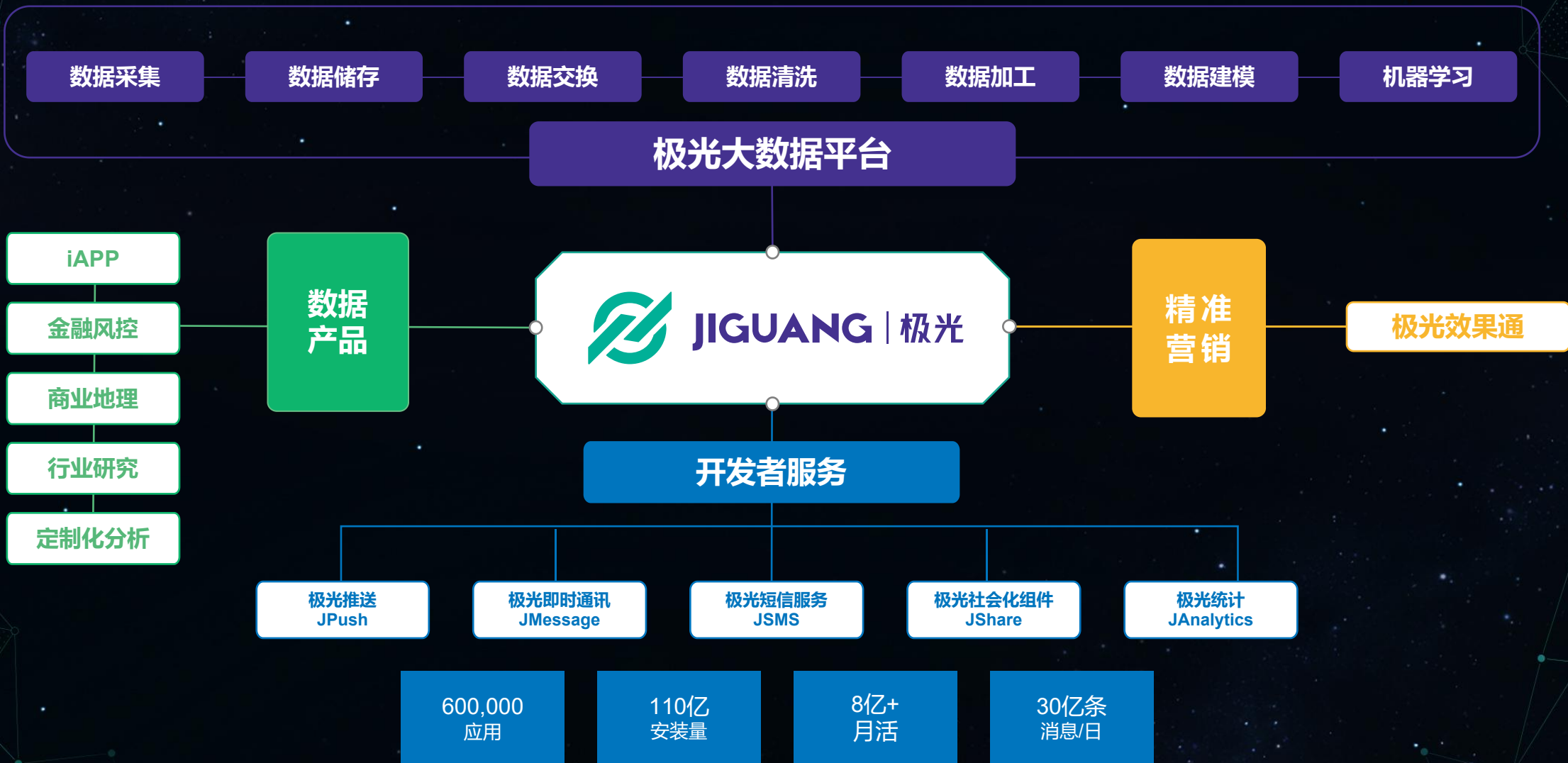
极光
效果通

极光开发者服务： 高效稳定SDK、节约开发资源、增强用户体验、提升业务效率。

极光效果通： 精准锁定目标用户，量身定制广告投放策略，提高运营效率。

极光数据服务： 解析目标客户，洞悉行业趋势，跟踪企业运营情况，帮助企业决策者提升决策效率。

关于极光-产品图谱

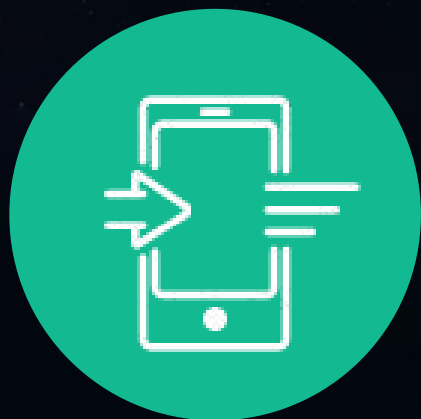


Part II – SDK架构优化

- 旧架构
- 新架构

SDK架构优化-旧架构

2016年以前SDK



极光推送



极光IM

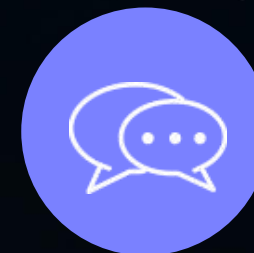


JPush

Network

Model

Cache



JMessage

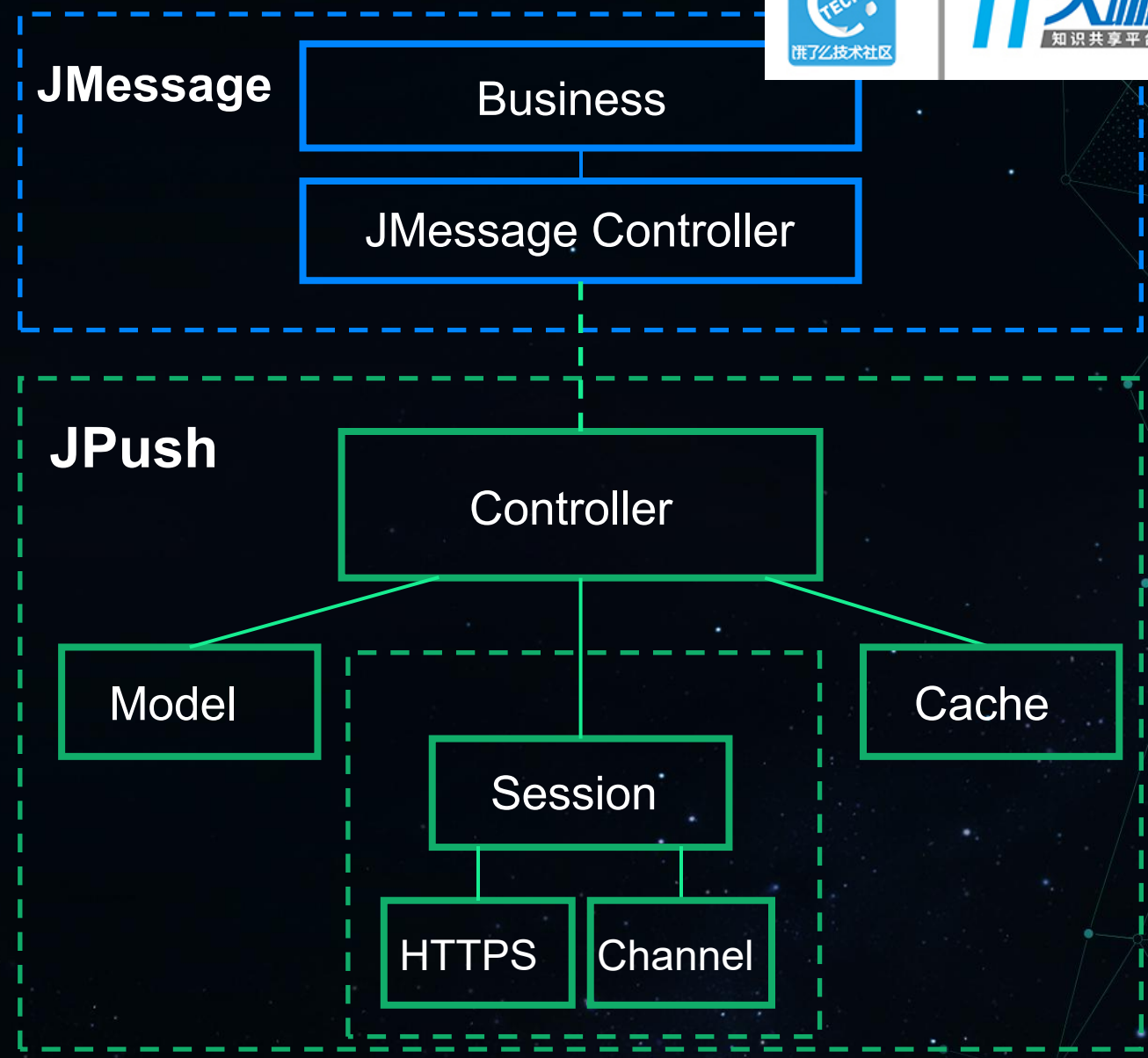
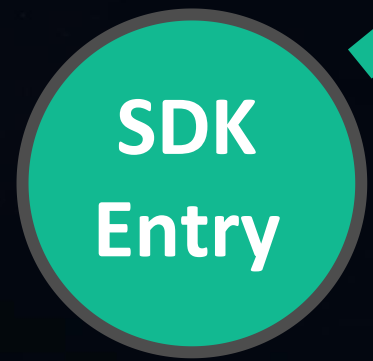
Network

Model

Cache

SDK架构优化-旧架构特点

- JPush SDK由单独的类库承载， JMessage业务依附于JPush的底层实现
- JPush实现了如数据通道， 任务队列， 用户活跃， 用户新增等功能
- JMessage拥有JPush的功能
- JMessage和JPush各自独立
- JMessage和JPush不能同时集成



2016年极光产品结构大变动



极光推送



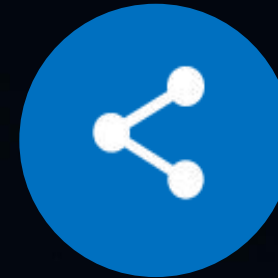
极光IM



极光短信



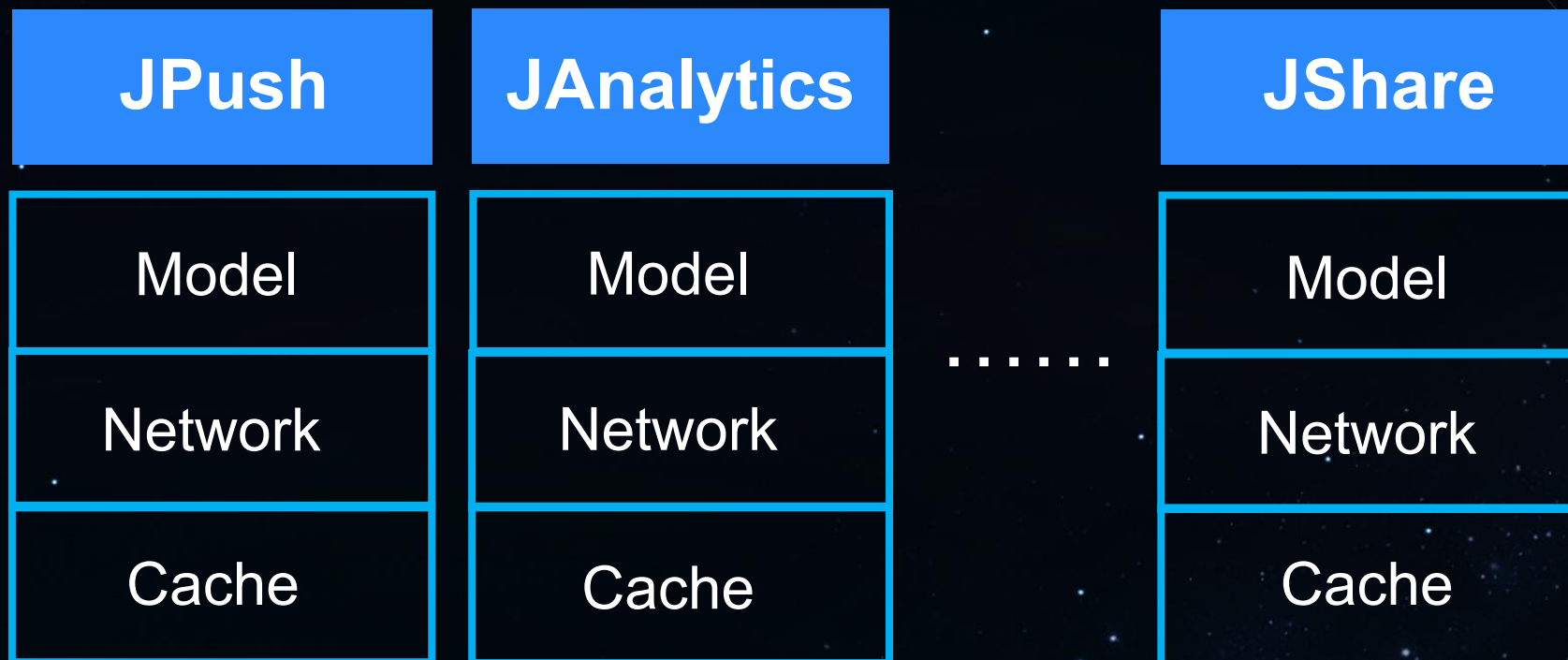
极光统计

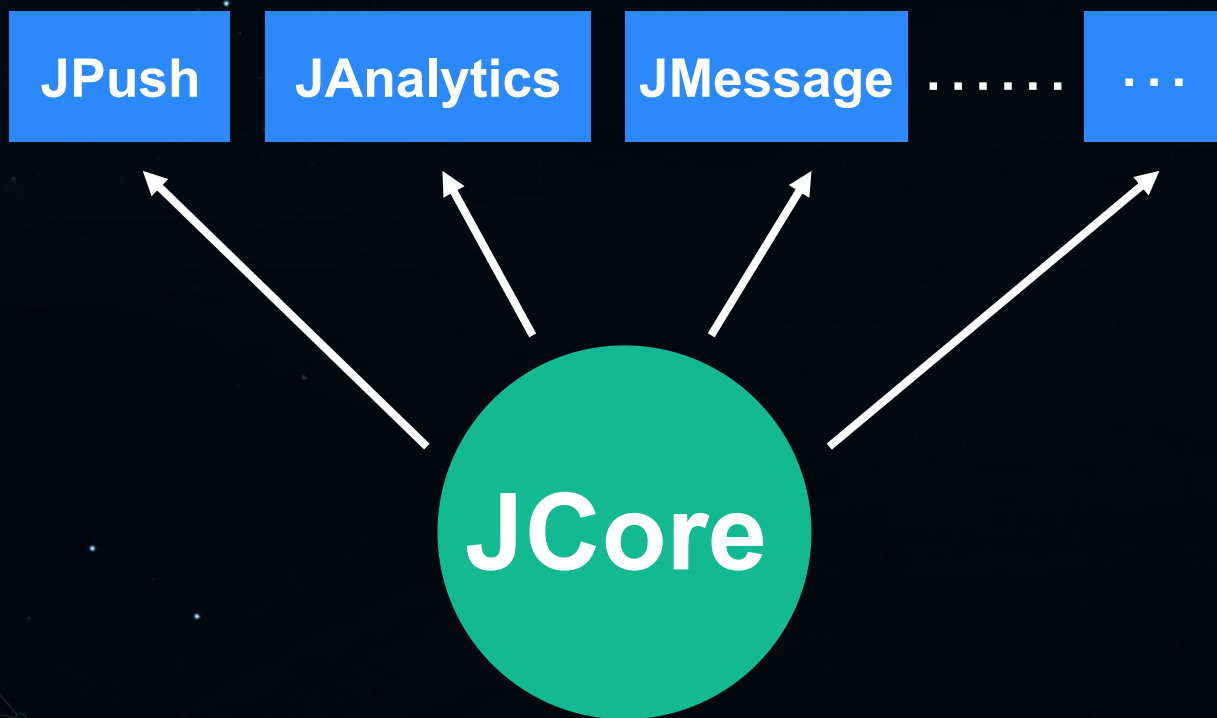


极光分享

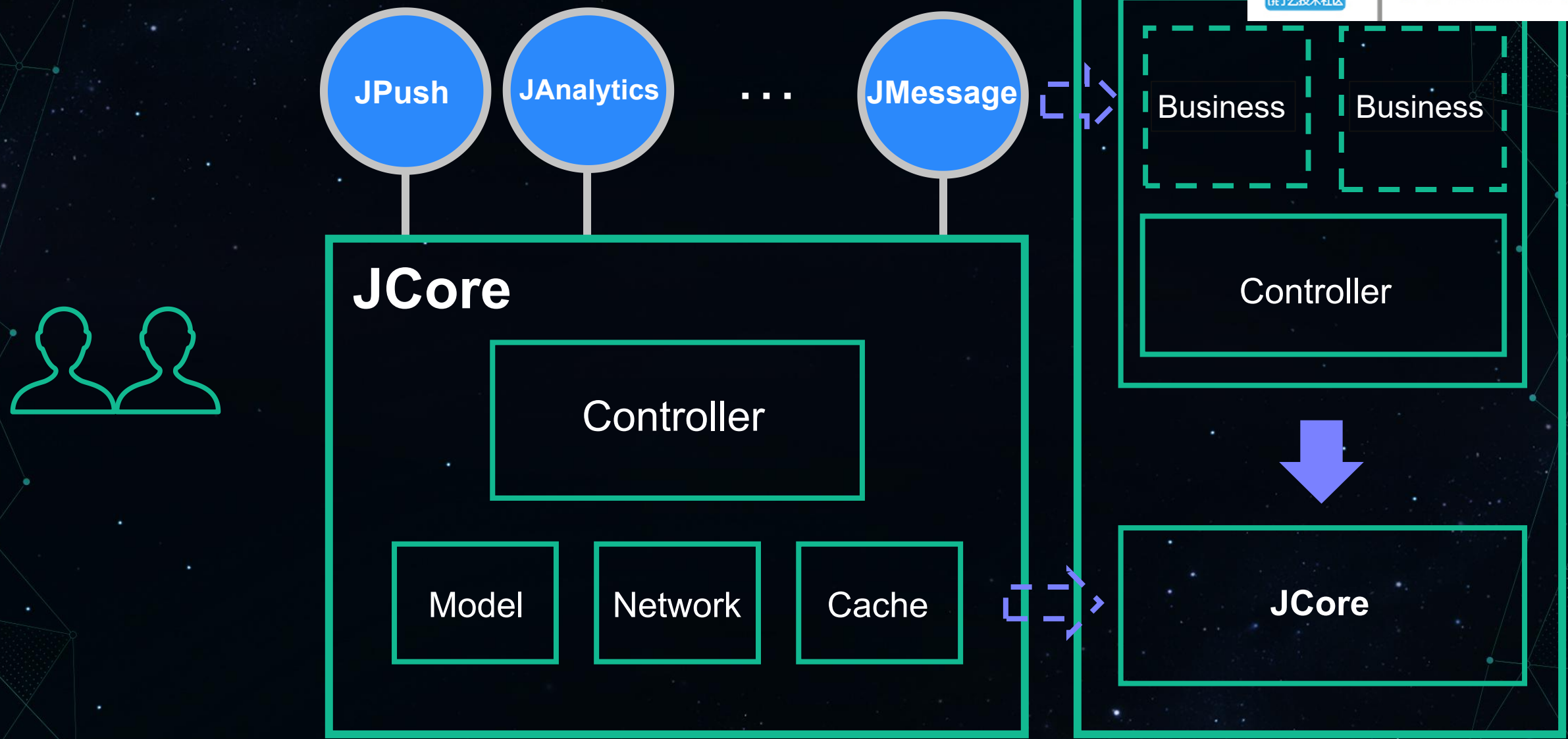
SDK架构优化-旧架构弊端

- SDK各自独立，不易管理
- 空间占用大
- 需要重复集成，重复注册SDK
- 占用没必要的线程
- 开启多个通道更消耗资源





- 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



JXxx

Business

Business

Business

.....

Business

JCore

Infrastructure

Thread

Channel

HTTPS

Log

Cache

Protocol

Common business

Register

Report

UserCtrl

.....

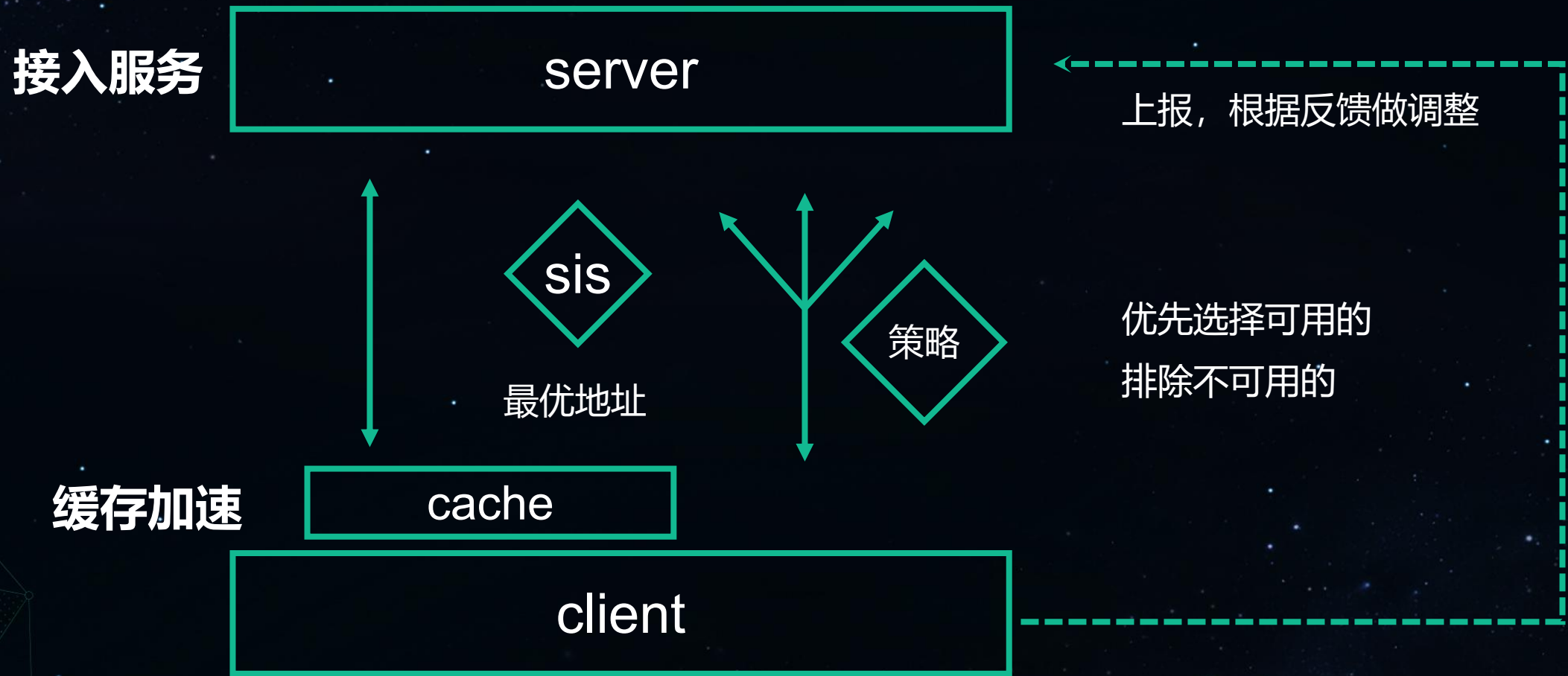
SDK架构优化-新架构劣势及存在的问题

- **核心包更新问题**
 - 测试成本高
 - 发布体验不好
- **相对于旧版集成繁琐**
- **核心包可能还存在一些性能问题**

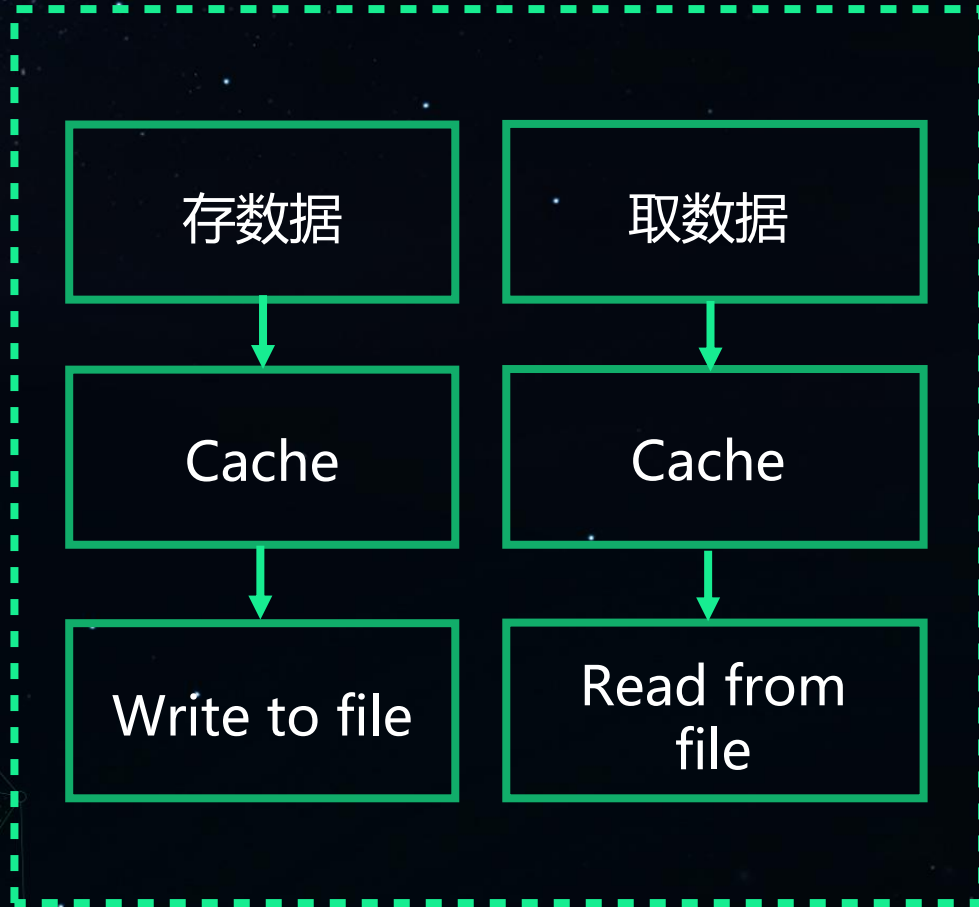
Part III - 性能优化

- 长连接
- 本地缓存

性能优化-长连接



性能优化-本地缓存



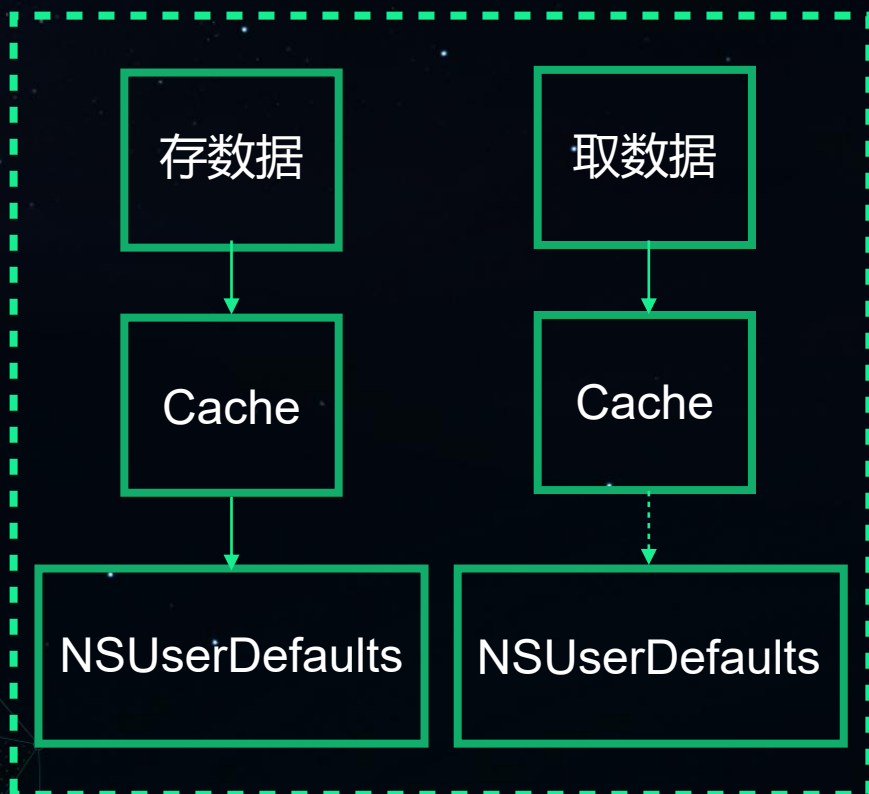
无论数据大小，都这么干，真的好么??

存在的弊端：

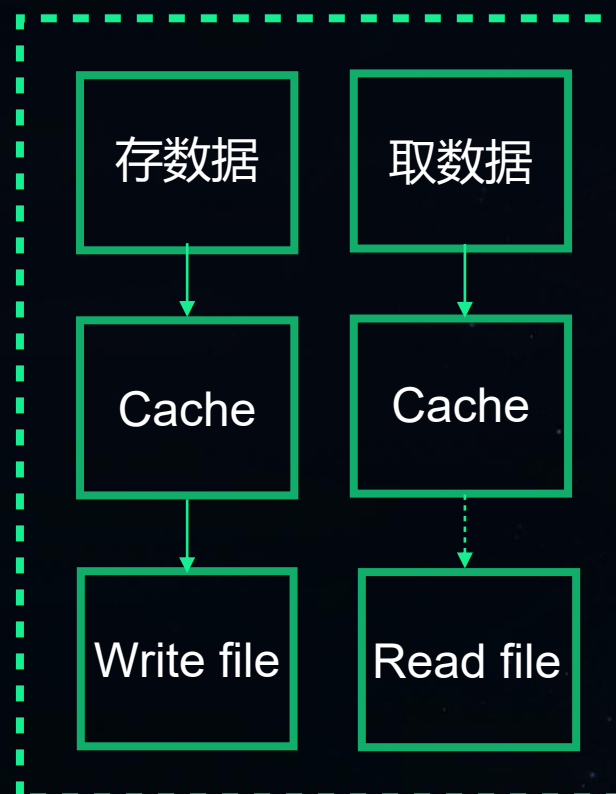
- 单线程操作，容易堵线程
- 达到一定数据量后，读写耗时较长
- 某种特殊情况下，容易丢数据

性能优化-本地缓存

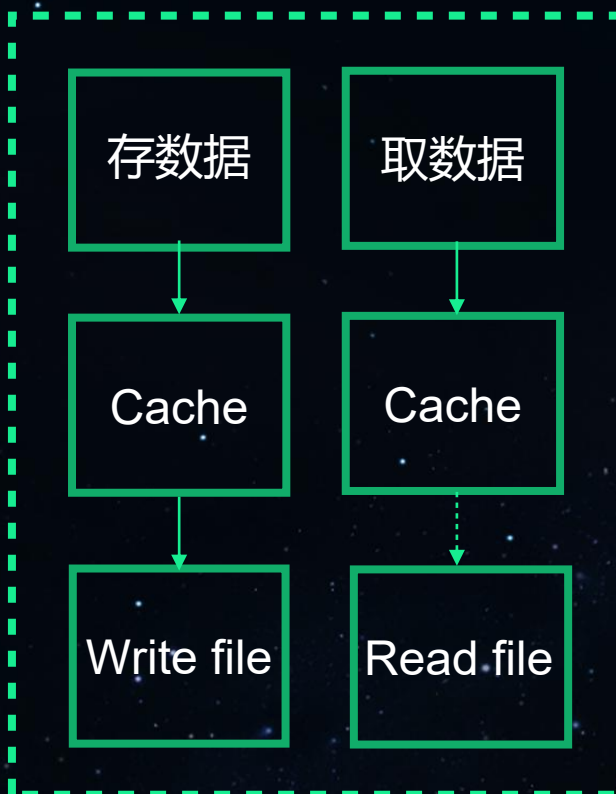
A线程



B线程



C线程



Part IV

APP开发的一些优化建议

- 思路

APP开发的一些优化建议-思路

- 把握架构设计合理性，多做结构优化分析
- 复杂的逻辑简单化，简单的逻辑更容易实现及维护
- 权衡利弊，不同的场合用适当的方法解决
- 多回头看看自己写的代码，或许能发现一些问题



IT大咖说
知识共享平台

END
T H A N K S