MDCC 2016
中国移动开发者大会
Mobile Developer Conference China 2016

IM 即时通讯技术在多应用场景下的技术实现，以及性能调优
（iOS视角）

-- 陈宜龙
微博@iOS程序犭袁

mdcc.csdn.net

自我介绍

- 陈宜龙，来自LeanCloud
- 我们的 IM 用户

。 。 。

三个部分

- 应用场景
- 针对移动网络特点的性能调优
- 技术实现细节

第一部分：应用场景

- IM 发展史
- 大家都在使用什么技术
- 社交场景
- 直播场景
- 数据自动更新场景
- 电梯场景（假在线状态处理）

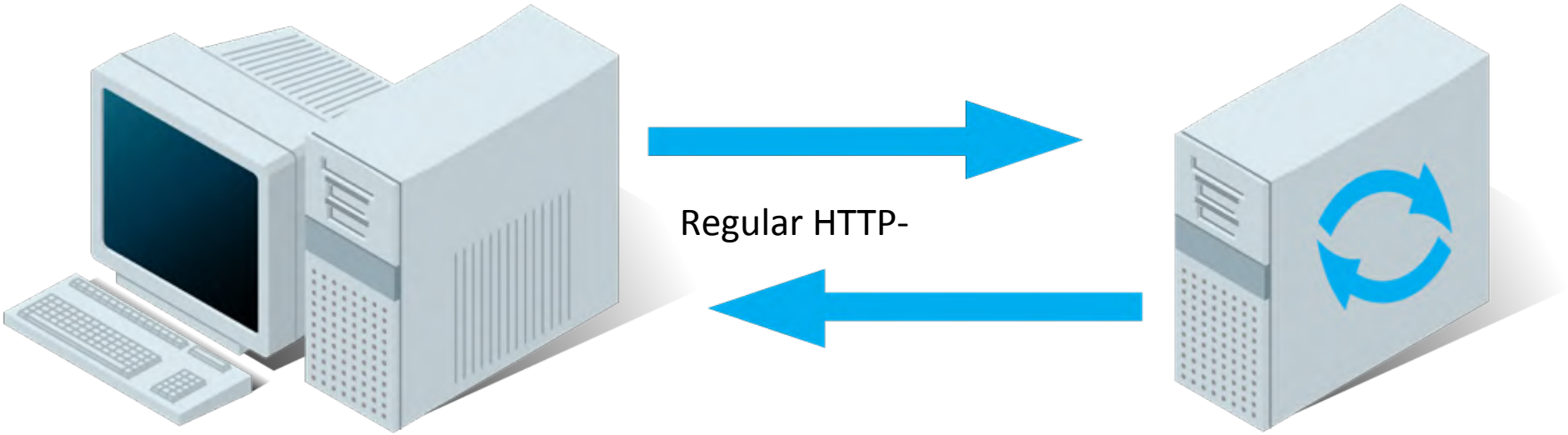
大规模即时通讯技术上的难点

- 电量，流量，及长连接的健壮性
- 保证IM系统的整体安全
- iOS生态下的政策以及结合新技术
- 降低开发者集成门槛

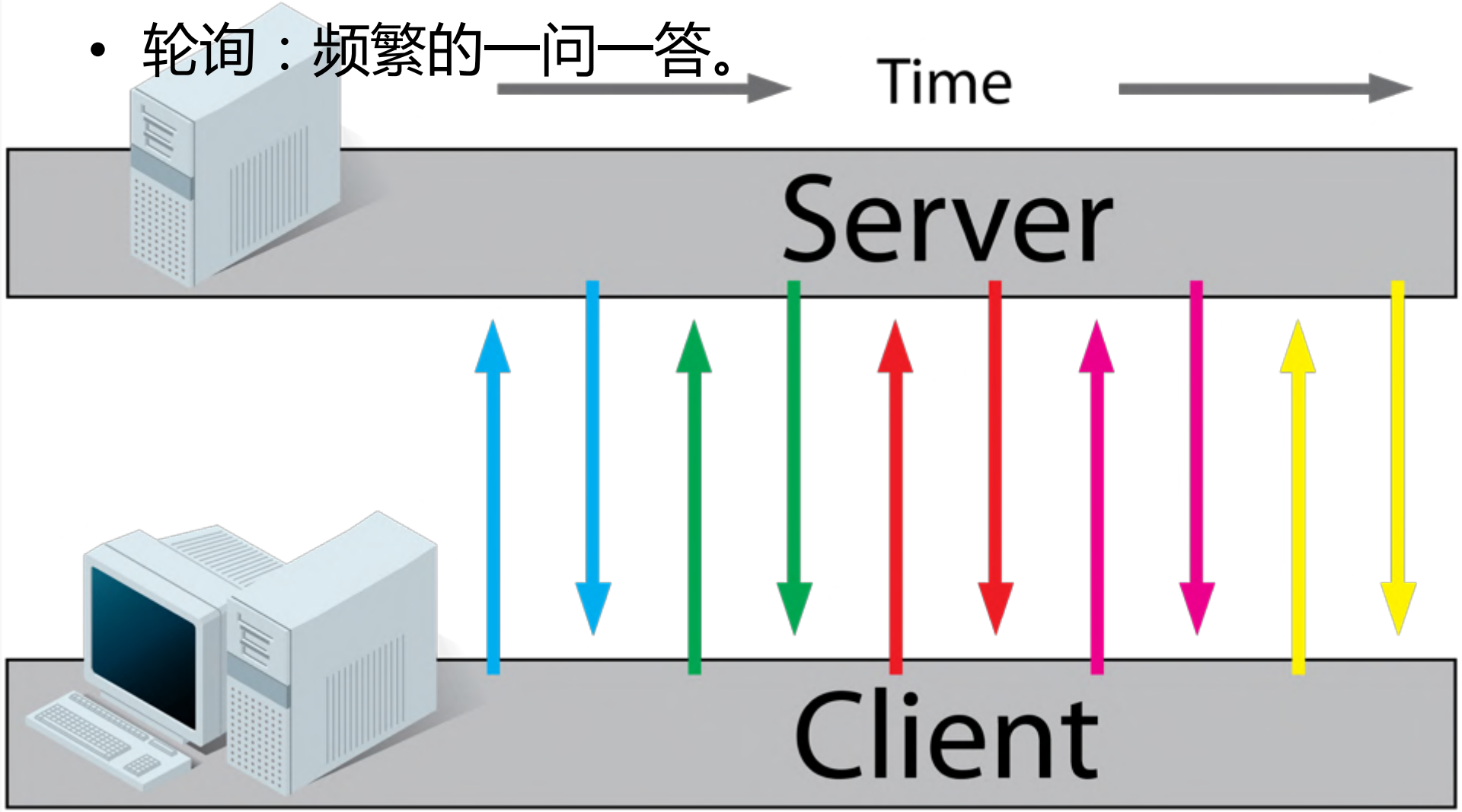
IM 发展史

- 轮询
- 长轮询
- 长连接

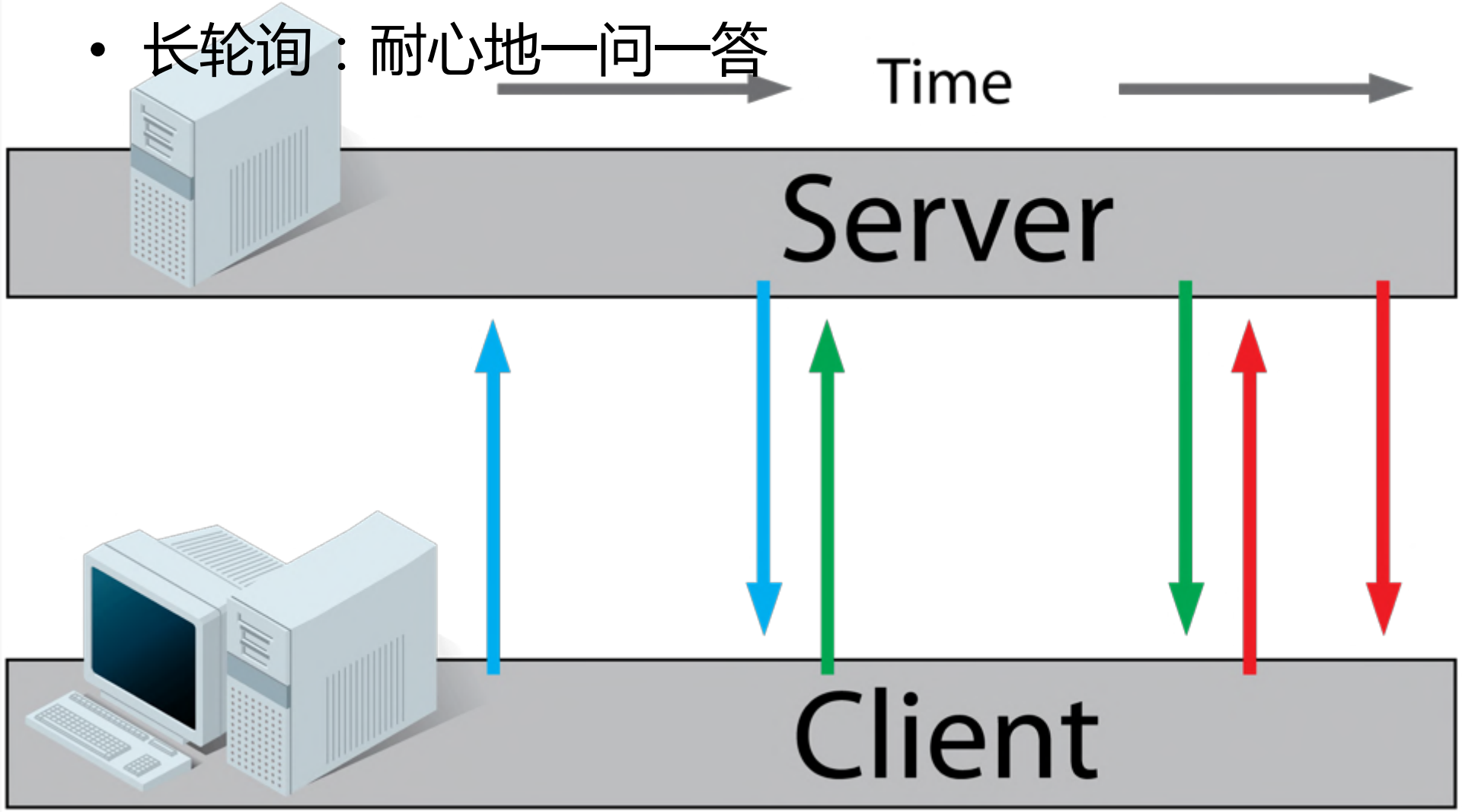
正常请求



- 轮询：频繁的一问一答。



- 长轮询：耐心地一问一答

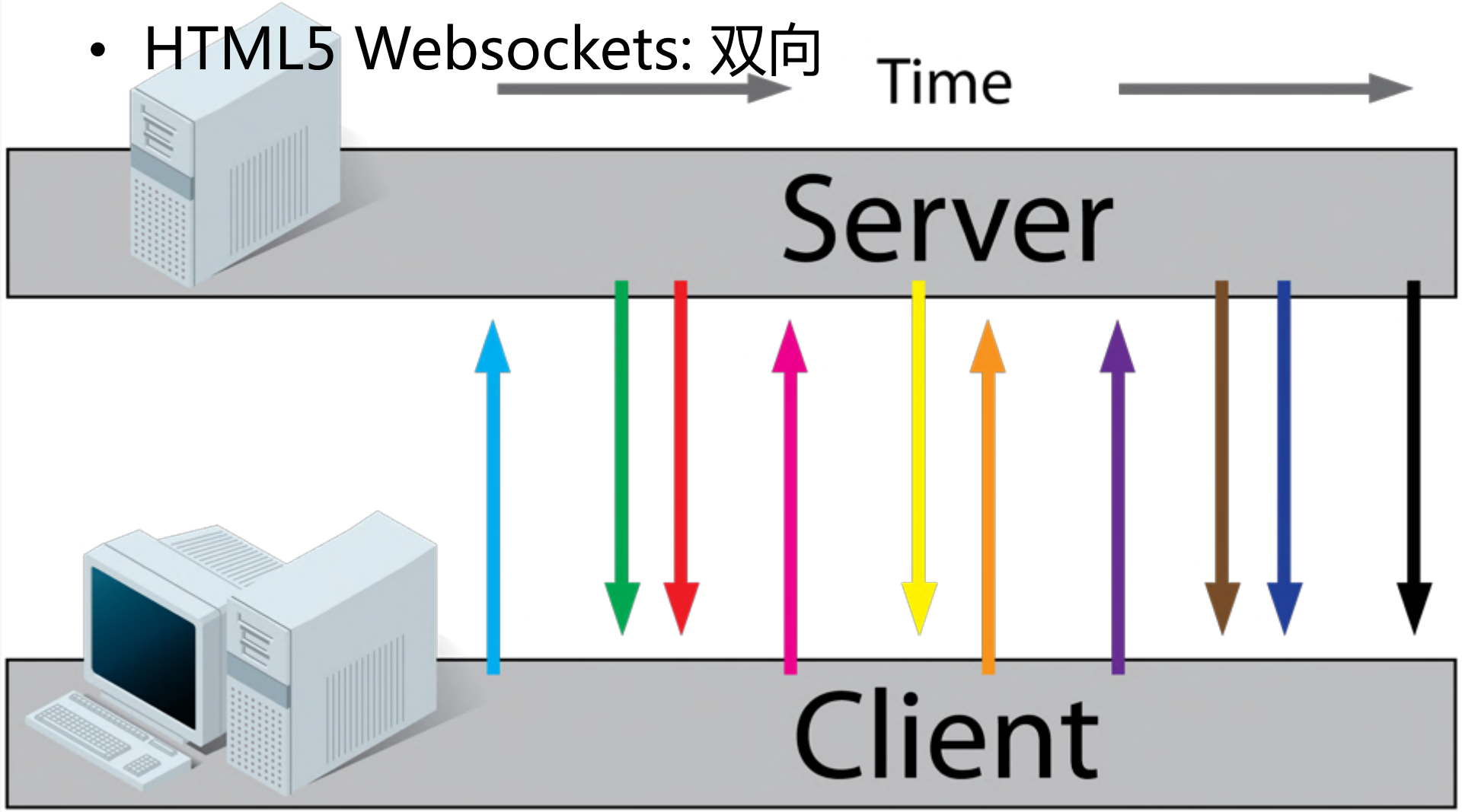


曾被 Facebook 早起版本采纳

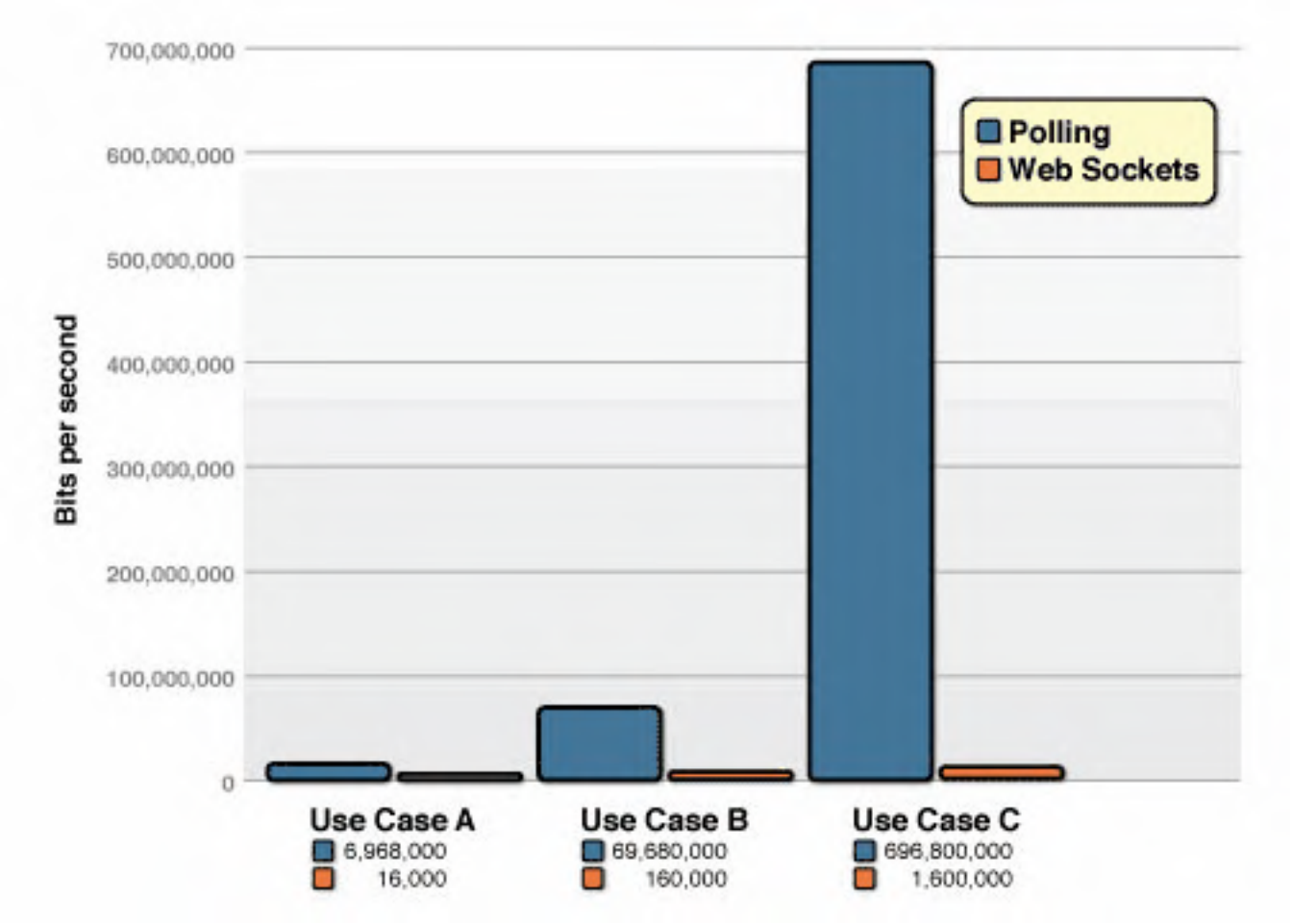
库存：100

秒杀

- HTML5 Websockets: 双向



- 轮询与Websocket的花费的流量对比：435倍



大家都在使用什么IM技术：

- 《你项目中使用什么协议实现了 IM 即时通讯》
- 《IM 即时通讯中你会选用什么数据传输格式？》

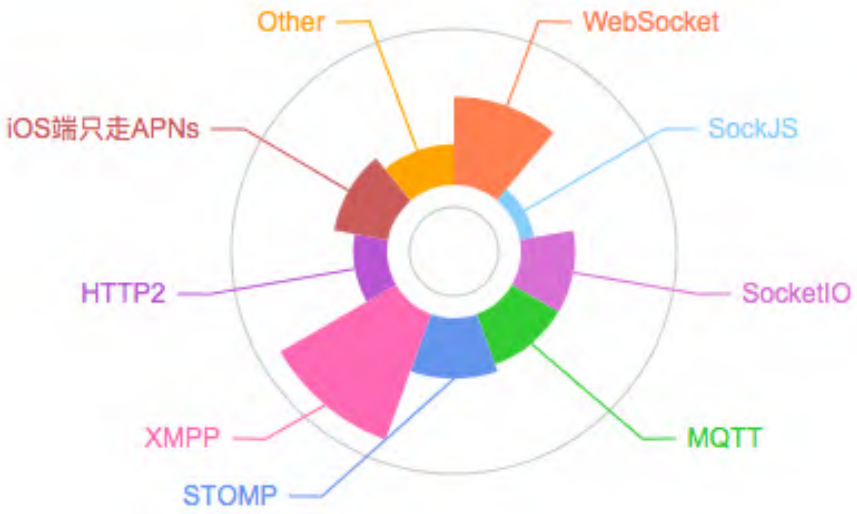
注：以上两次投票是发布在微博@iOS程序犭袁，鉴于微博的粉丝关注机制，本数据只能反映出 IM 技术在移动领域或者说是 iOS 领域的使用情况，可能并不能反映出整个IT行业的情况。


你项目中使用什么协议实现了 IM 即时通讯：
581
参与人数
结束倒计时:355天
如果使用第三方SDK，请自行 Google 对应技术。
投票选项 最多选8项
WebSocket 138(21.5%)
SockJS 3(0.5%)
SocketIO 55(8.6%)
MQTT 48(7.5%)
STOMP 6(0.9%)
XMPP 290(45.2%)
HTTP2 18(2.8%)
iOS端只走APNs 59(9.2%)
其它（请在评论中写出对应协议） 25(3.9%)
已投票

你项目中使用什么协议实现了 IM 即时通讯：

如果使用第三方SDK，请自行 Google 对应技术。

470
参与人数



WebSocket SockJS SocketIO MQTT STOMP XMPP HTTP2
iOS端只走APNs Other

IM协议选择原则一般是：

- 易于拓展
- 节约流量
- 高效，简洁，可读性好
- 能够匹配当前团队的技术堆栈。

名称	优点	缺点
XMPP	优点：协议开源，可拓展性强，在各个端(包括服务器)有各种语言的实现，开发者接入方便；	缺点：缺点也是不少，XML表现力弱、有太多冗余信息、流量大，实际使用时有大量天坑。
MQTT	优点：协议简单，流量少；订阅+推送模式，非常适合Uber、滴滴的小车轨迹的移动。	缺点：它并不是一个专门为 IM 设计的协议，多使用于推送。IM 情景要复杂得多，pub、sub，比如：加入对话、创建对话等等事件。
私有协议	市面上几乎所有主流IM APP都是是使用私有协议，一个被良好设计的私有协议优点非常明显。优点：高效，节约流量(一般使用二进制协议)，安全性高，难以破解；	缺点：在开发初期没有现有样列可以参考，对于设计者的要求比较高。
WebSocket	web原生支持，很多第三方语言实现，可以搭配XMPP、MQTT等多种聊天协议	-

社交场景

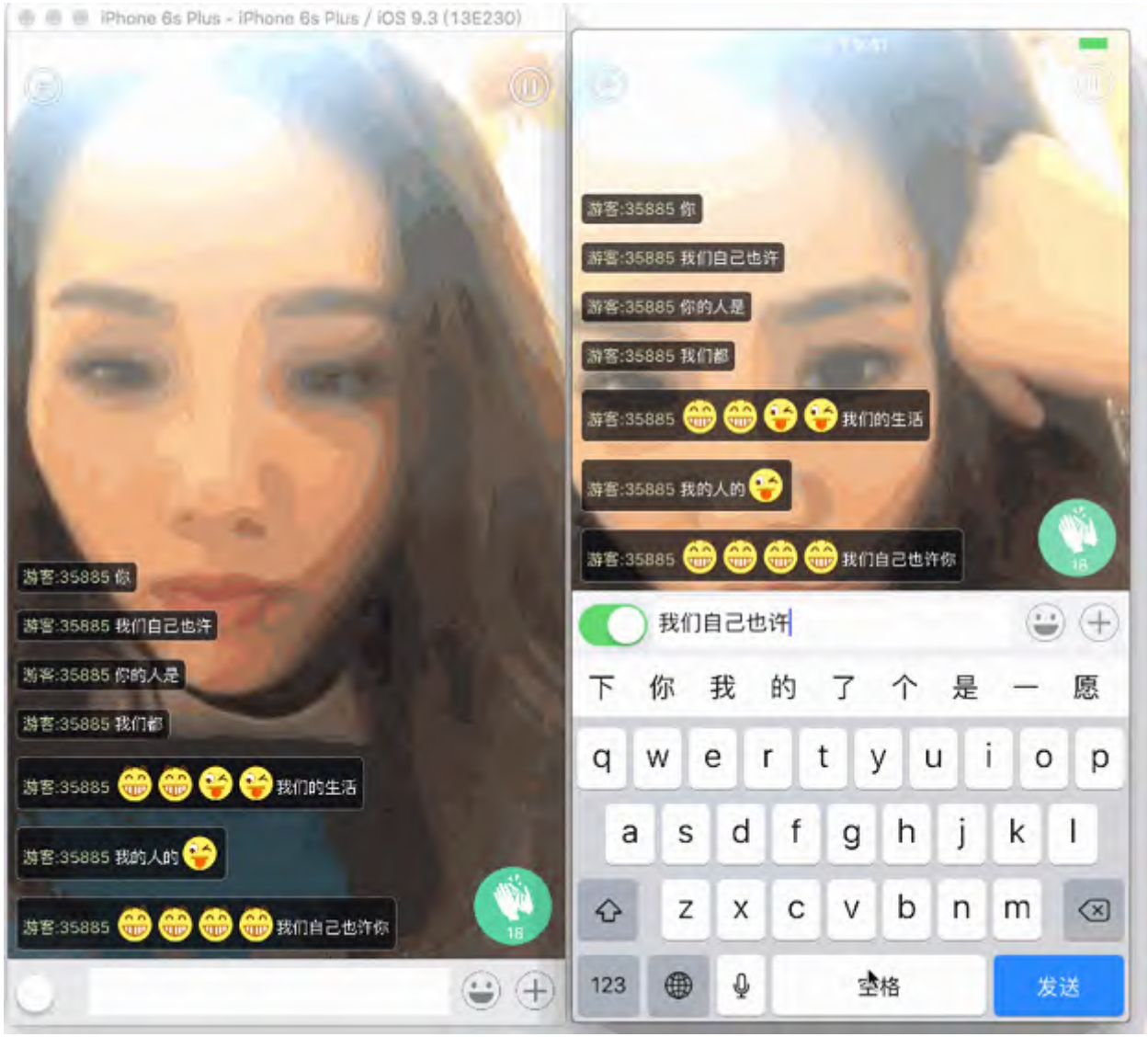
- 模式成熟，界面类似

IM+直播Demo：LiveKit-iOS

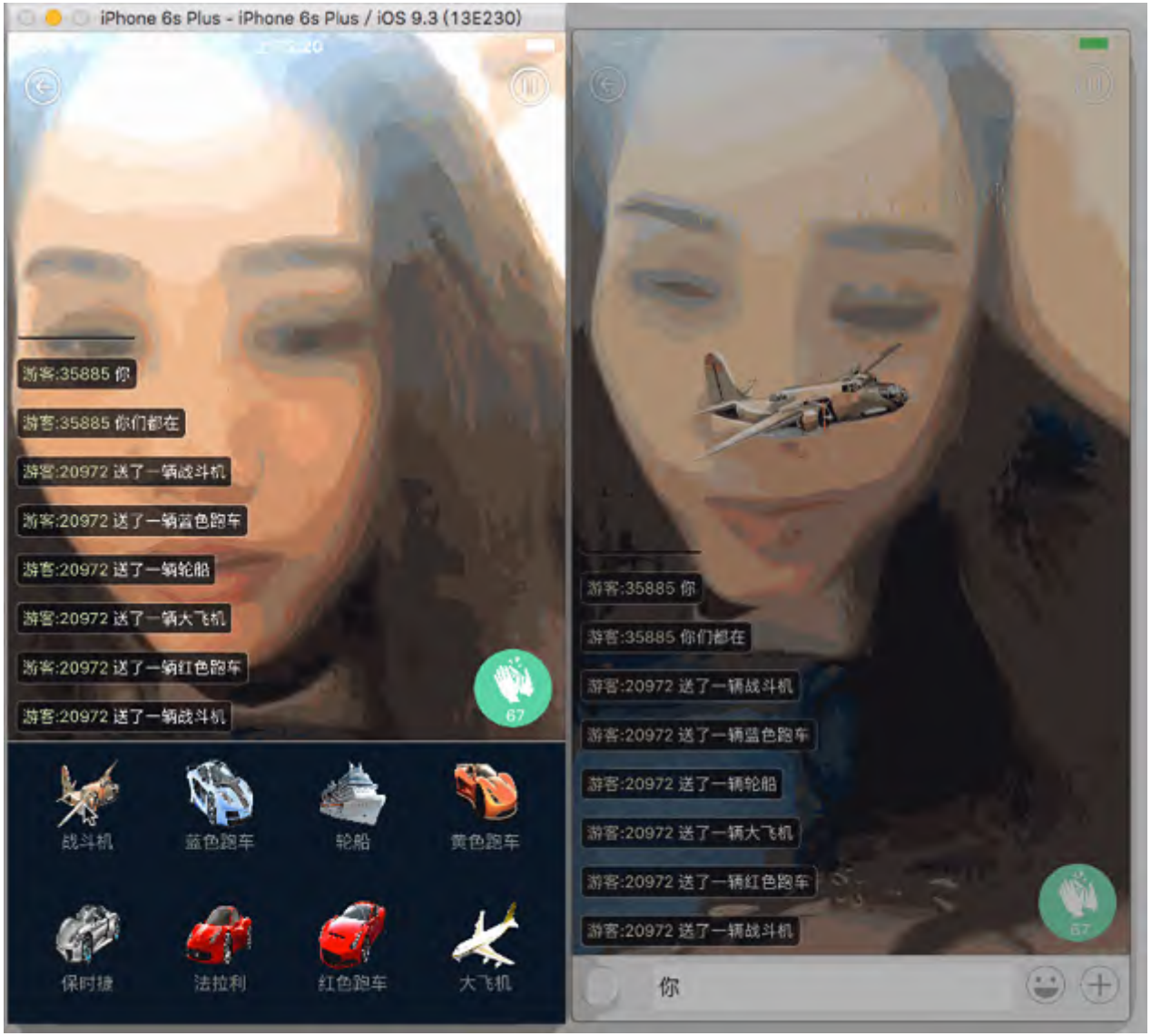
LiveKit 相较社交场景的特点：

- 无人数限制的聊天室
- 自定义消息
- 打赏机制的服务端配合

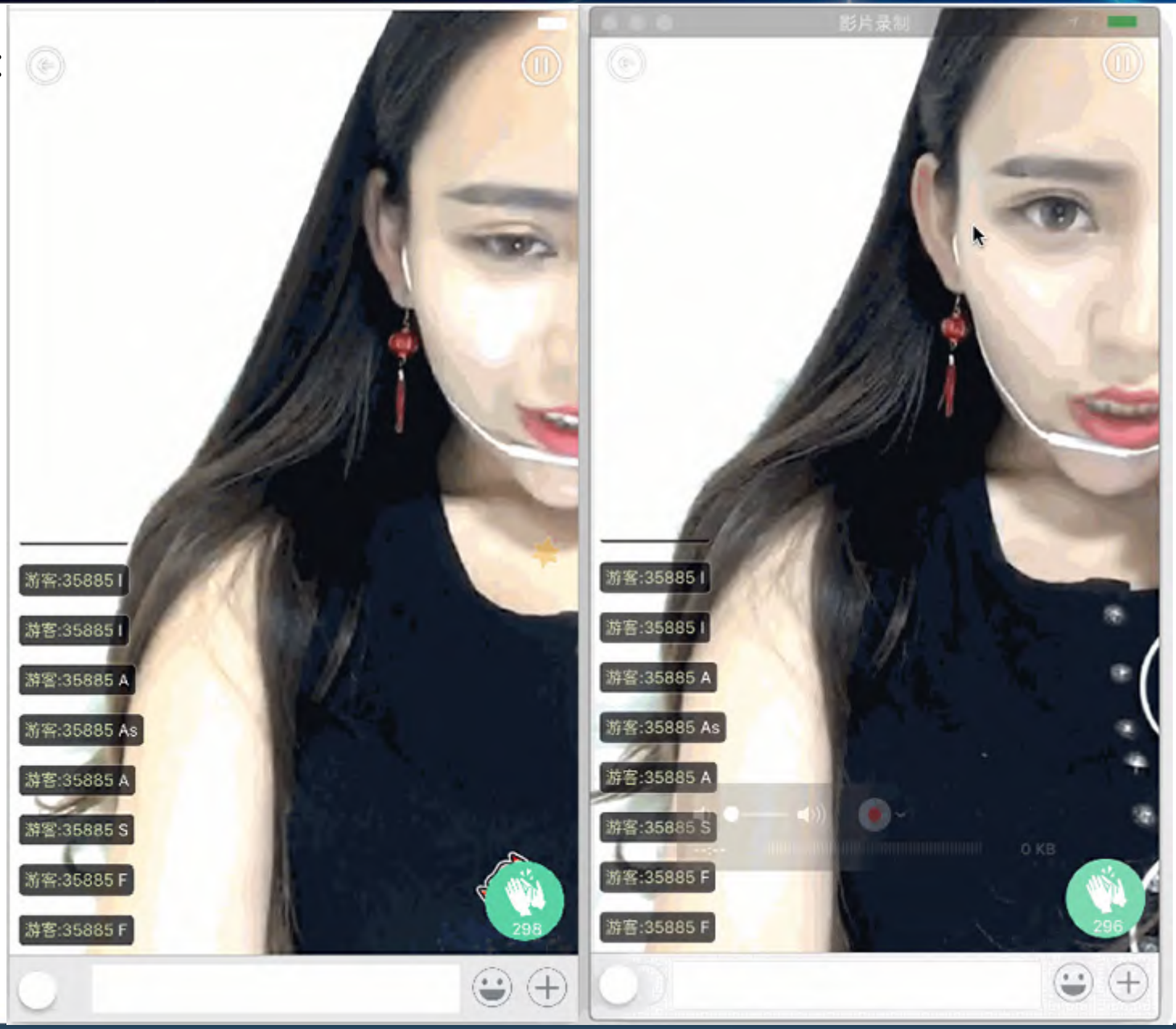
聊天室：


iPhone 6s Plus - iPhone 6s Plus / iOS 9.3 (13E230)
游客:35885 你
游客:35885 我们自己也许
游客:35885 你的人是
游客:35885 我们都
游客:35885 我们的生活
游客:35885 我的人的
游客:35885 我们自己也许你
我们自己也许
下 你 我 的 了 个 是 一 愿
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
123
空格
发送

礼物：


iPhone 6s Plus - iPhone 6s Plus / iOS 9.3 (13E230)
游客:35885 你
游客:35885 你们都在
游客:20972 送了一辆战斗机
游客:20972 送了一辆蓝色跑车
游客:20972 送了一辆轮船
游客:20972 送了一辆大飞机
游客:20972 送了一辆红色跑车
游客:20972 送了一辆战斗机
67
战斗机
蓝色跑车
轮船
黄色跑车
保时捷
法拉利
红色跑车
大飞机
游客:35885 你
游客:35885 你们都在
游客:20972 送了一辆战斗机
游客:20972 送了一辆蓝色跑车
游客:20972 送了一辆轮船
游客:20972 送了一辆大飞机
游客:20972 送了一辆红色跑车
游客:20972 送了一辆战斗机
67
你

点赞出心：


游客:35885 I
游客:35885 I
游客:35885 A
游客:35885 As
游客:35885 A
游客:35885 S
游客:35885 F
游客:35885 F
298
影片录制
游客:35885 I
游客:35885 I
游客:35885 A
游客:35885 As
游客:35885 A
游客:35885 S
游客:35885 F
游客:35885 F
0 KB
296

数据自动更新场景

- 打车应用场景（Uber、滴滴等APP移动小车）
- 朋友圈状态自动更新

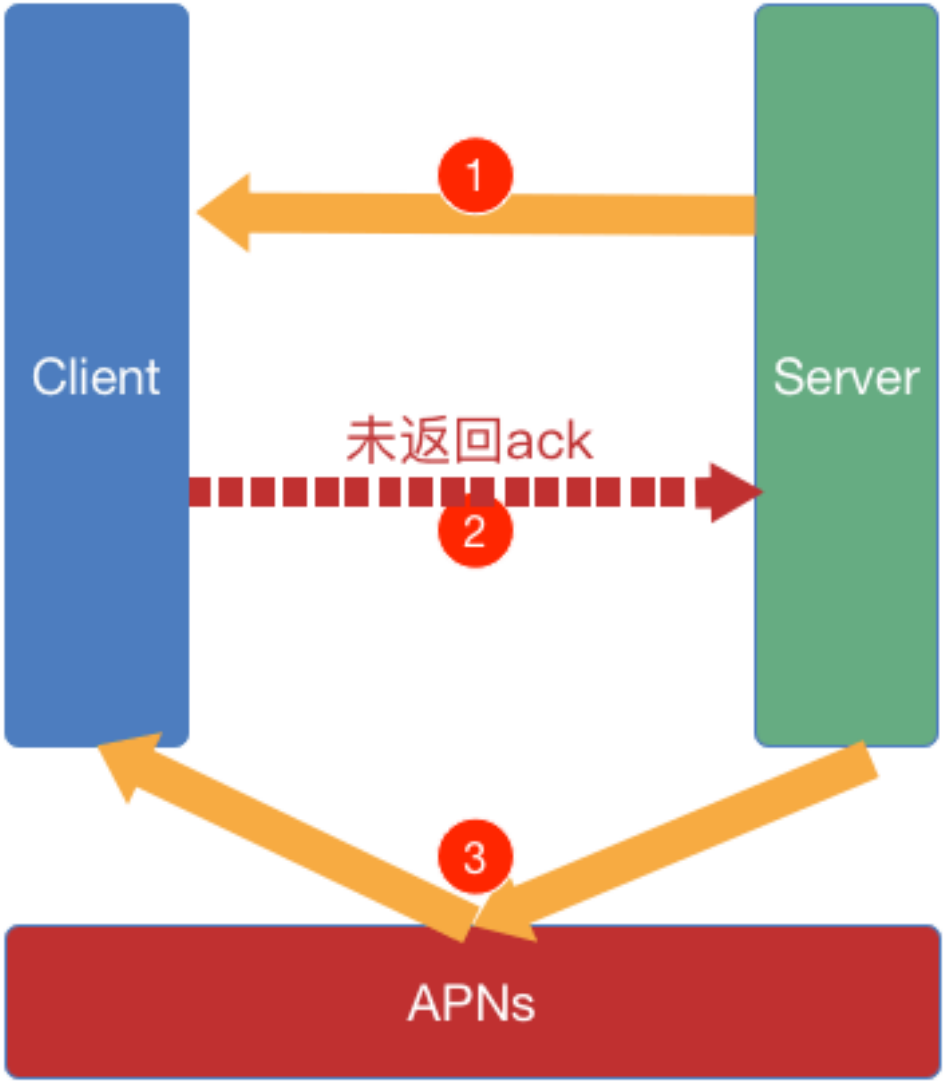
建议：使用 MQTT 实现最为经济

电梯场景（假在线状态处理）

iOS端的假在线的状态，有两种方案：

- 双向ping pong机制
- iOS端只走APNs

双向ping pong机制：



使用 APNs 来作聊天的优缺点

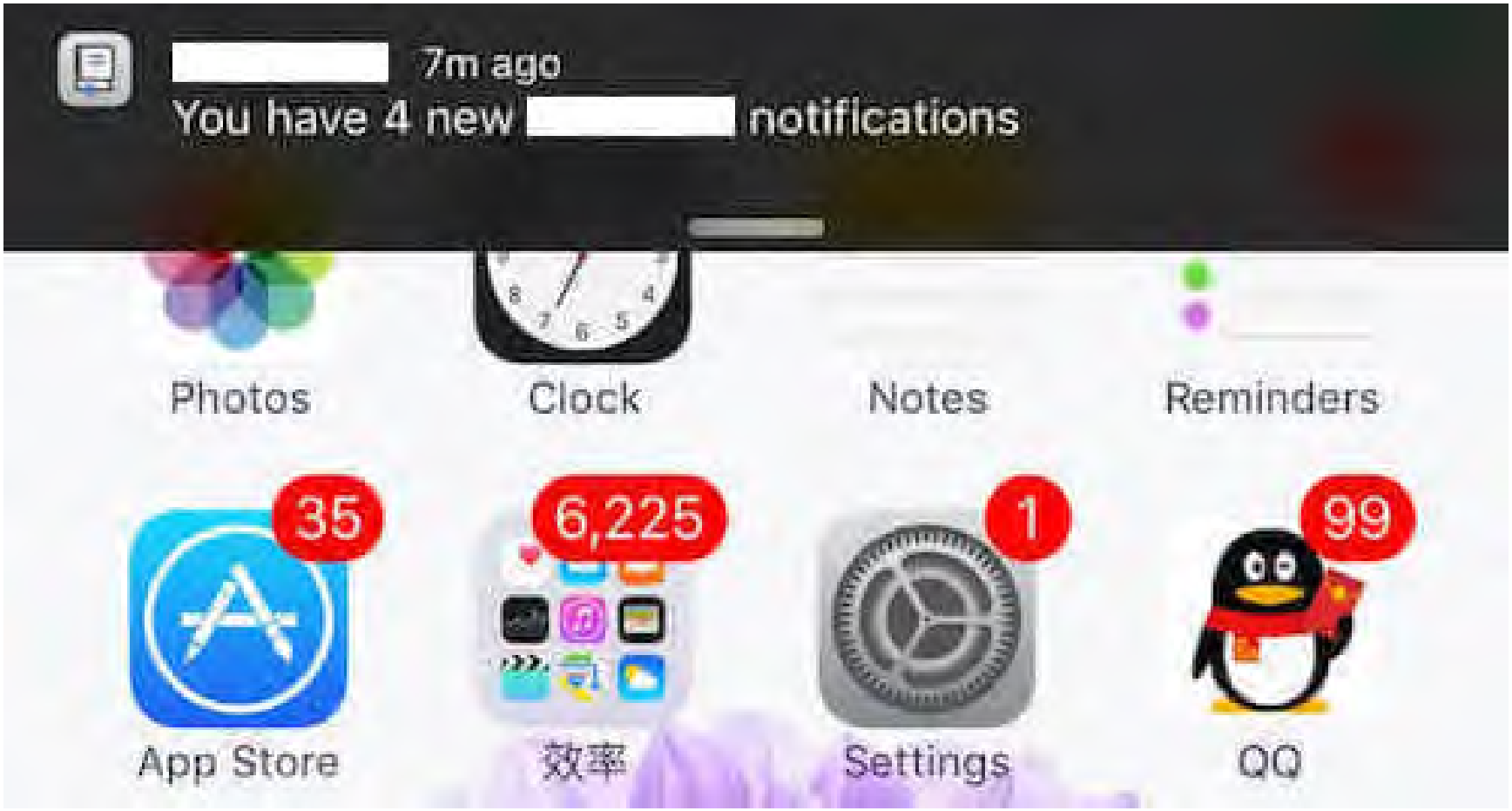
优点：

- 解决了iOS端假在线的问题。

缺点：

- 无法保证消息的及时性。无法保证准确性。
- 服务端压力大

被折叠的推送消息：


7m ago
You have 4 new notifications
Photos
Clock
Notes
Reminders
35
6,225
1
99
App Store
效率
Settings
QQ

适用于：

- 用户对消息的及时性并不敏感。
- 用户量小

第二部分：
针对移动网络特点的性能调优

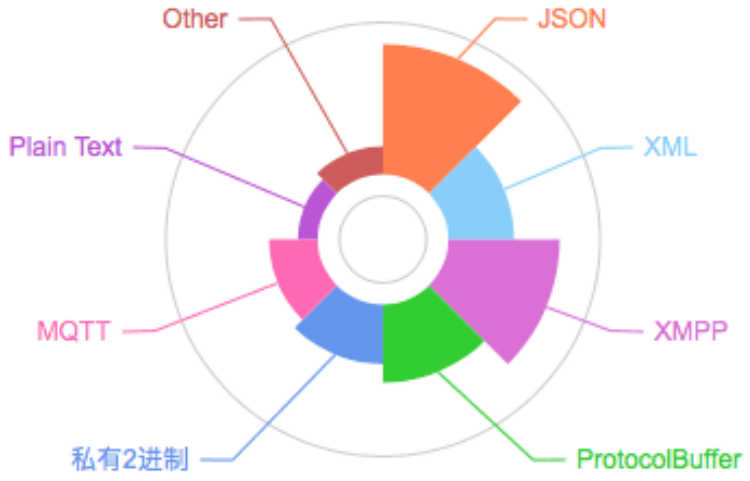
- 极简协议，传输协议 Protobuf
- 在安全上做了哪些事情？
  - 防止 DNS 污染
  - 防止 DDos 攻击
  - 账户安全
- 重连机制
- 使用 HTTP/2 减少不必要的网络连接
- 设置合理的超时时间
- 图片视频等文件上传
- 使用缓存：类似 Hash 的本地缓存校验


IM 即时通讯中你会选用什么数据传输格式？
301
参与人数
结束倒计时:360天
如果使用第三方SDK，请自行 Google 对应技术。
投票选项 最多选7项
JSON 122(35.2%)
XML 29(8.4%)
XMPP 97(28%)
ProtocolBuffer 49(14.1%)
私有2进制 27(7.8%)
MQTT 17(4.9%)
Plain text 2(0.6%)
其它（请在评论中写出对应格式） 4(1.2%)
已投票

IM 即时通讯中你会选用什么数据传输格式？

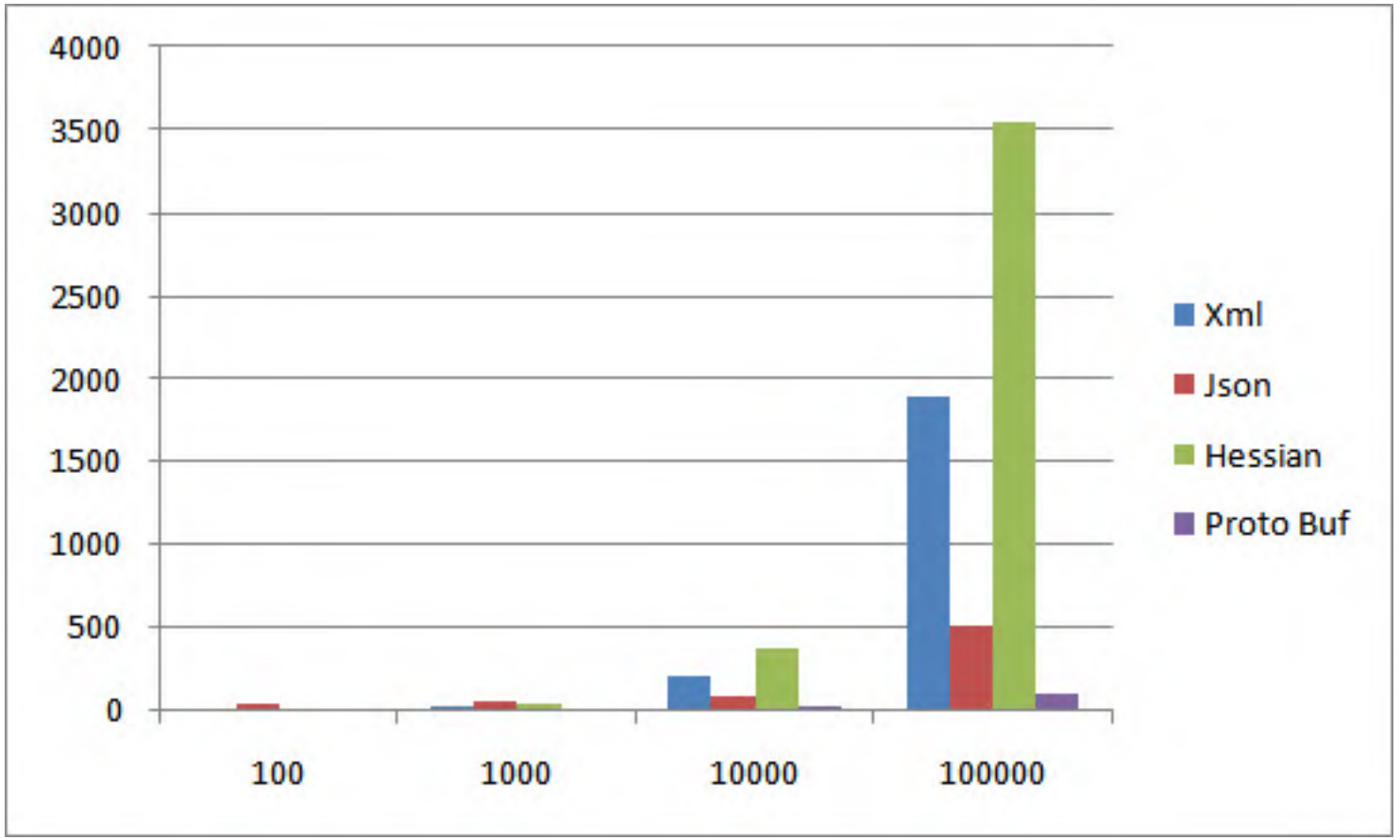
201
参与人数

如果使用第三方SDK，请自行 Google 对应技术。

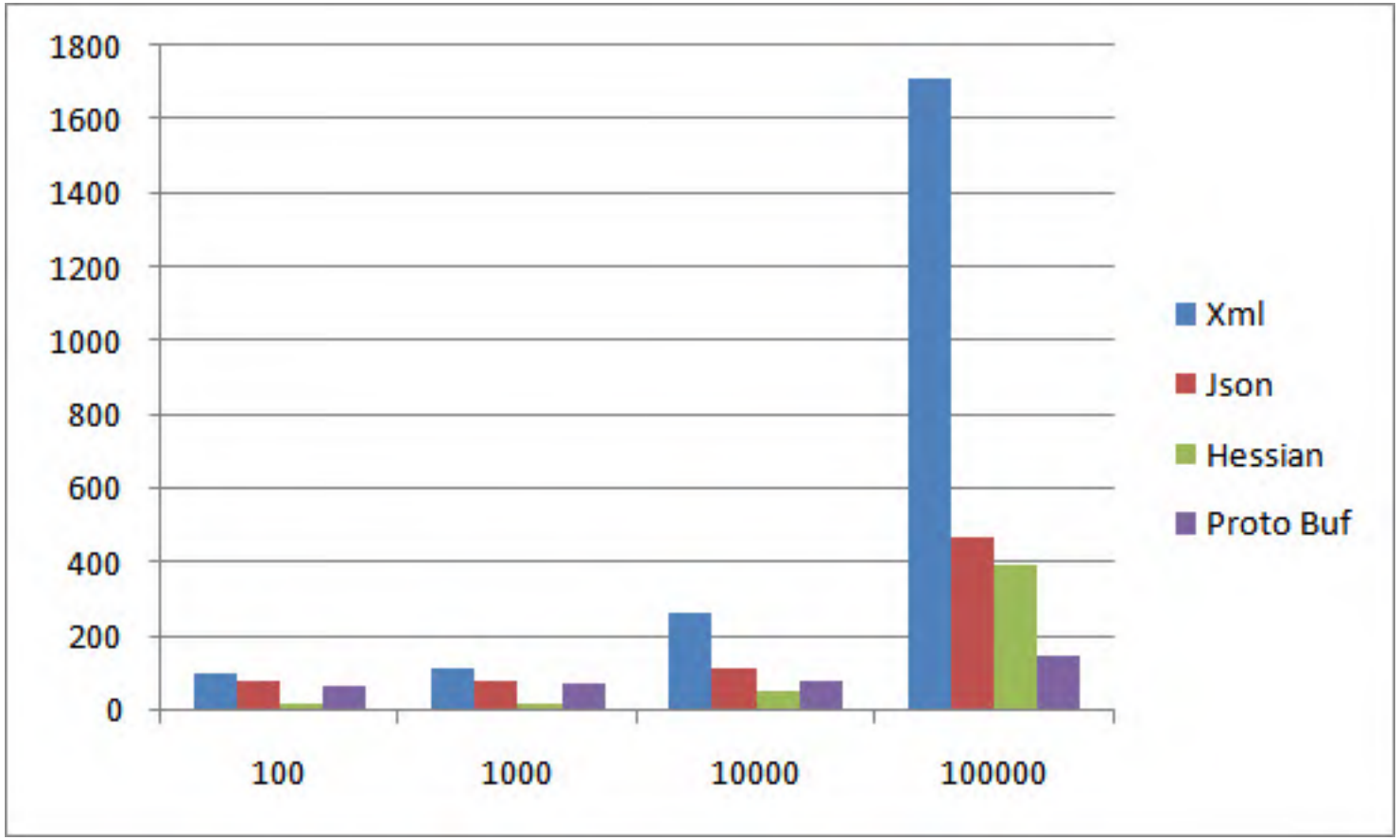


JSON XML XMPP ProtocolBuffer 私有2进制 MQTT
Plain Text Other

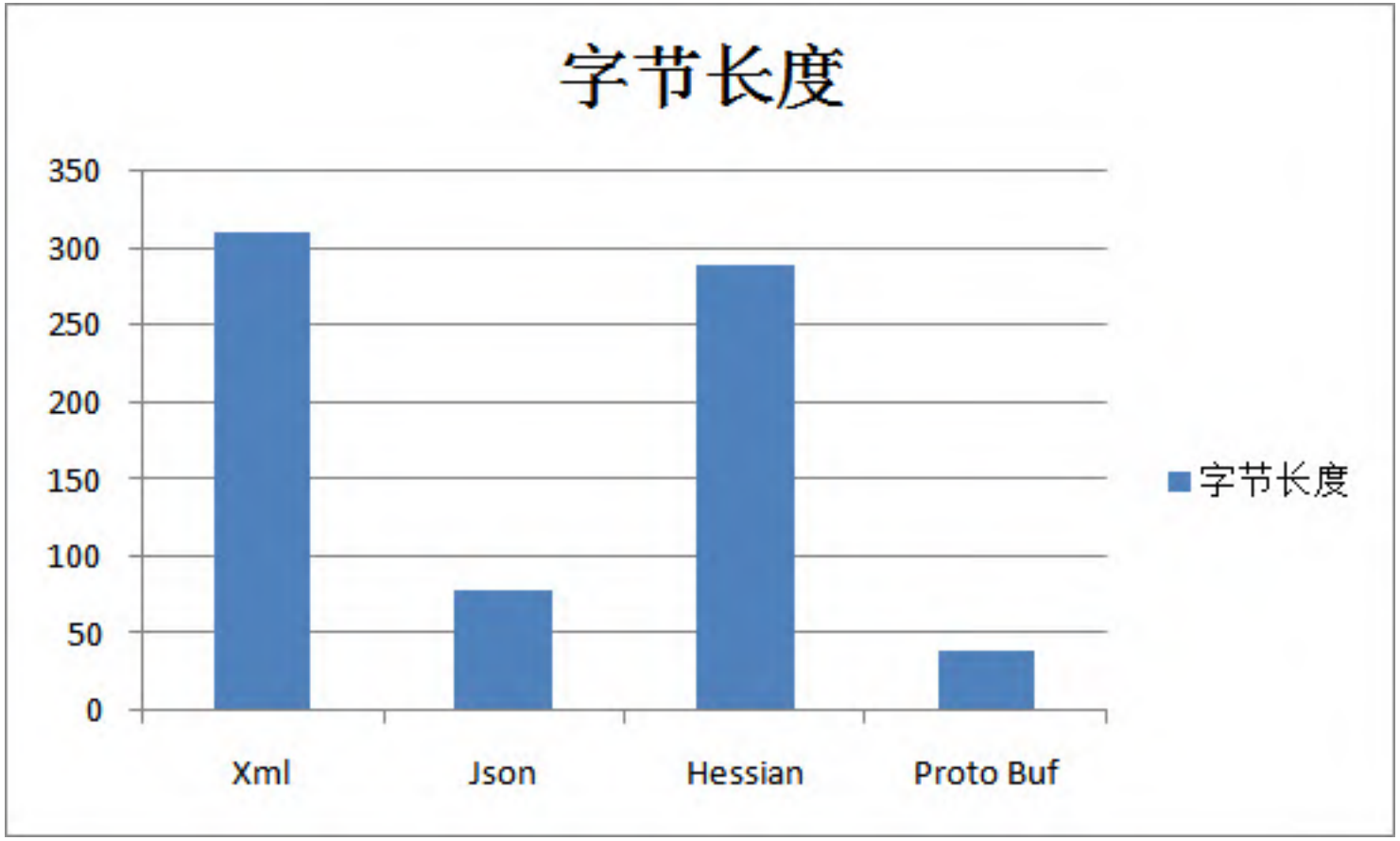
- 主流协议对比：反序列化


4000
3500
3000
2500
2000
1500
1000
500
0
100
1000
10000
100000
Xml
Json
Hessian
Proto Buf

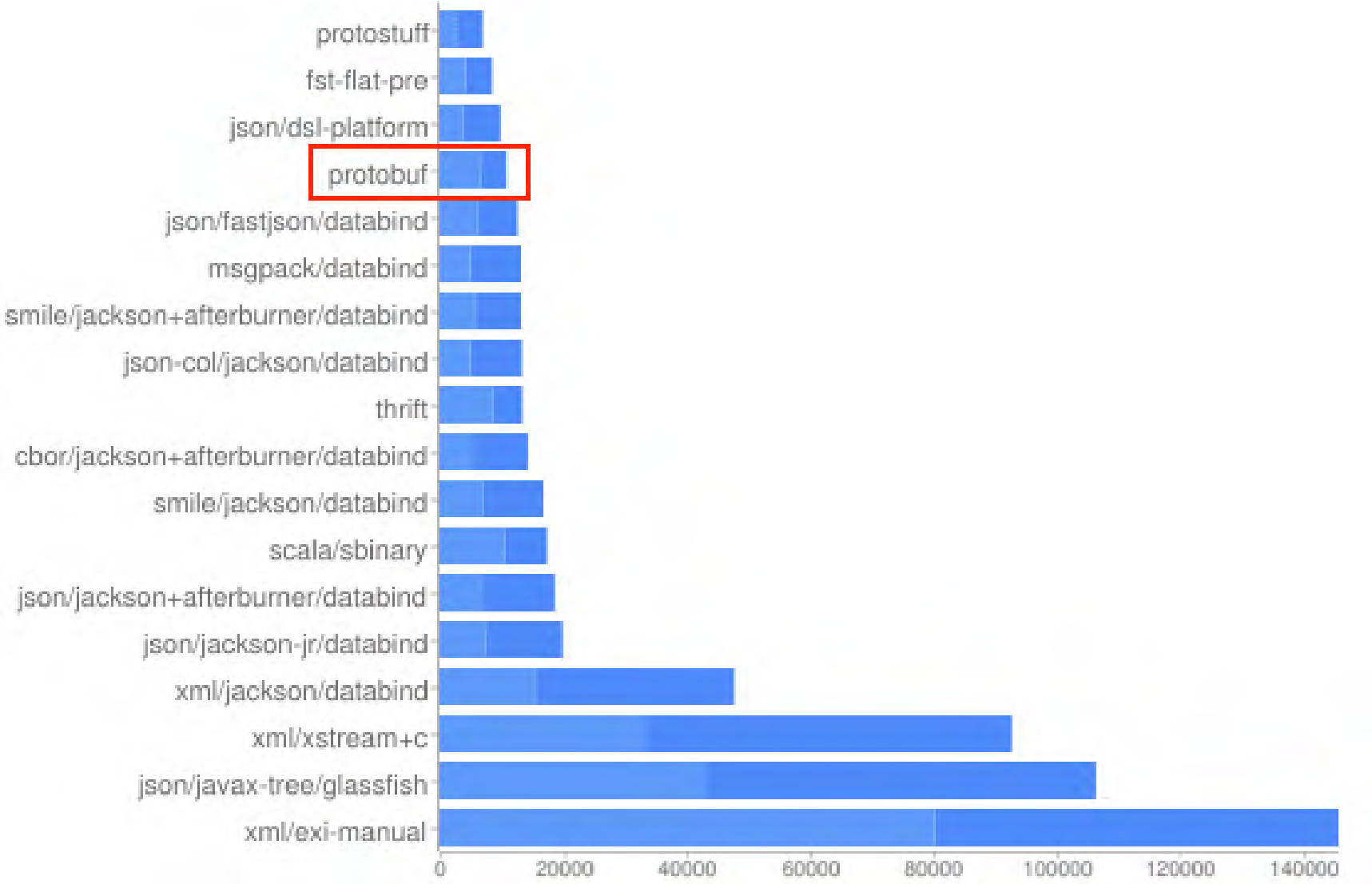
- 主流协议对比：序列化


1800
1600
1400
1200
1000
800
600
400
200
0
100
1000
10000
100000
Xml
Json
Hessian
Proto Buf

- 主流协议对比：字节长度

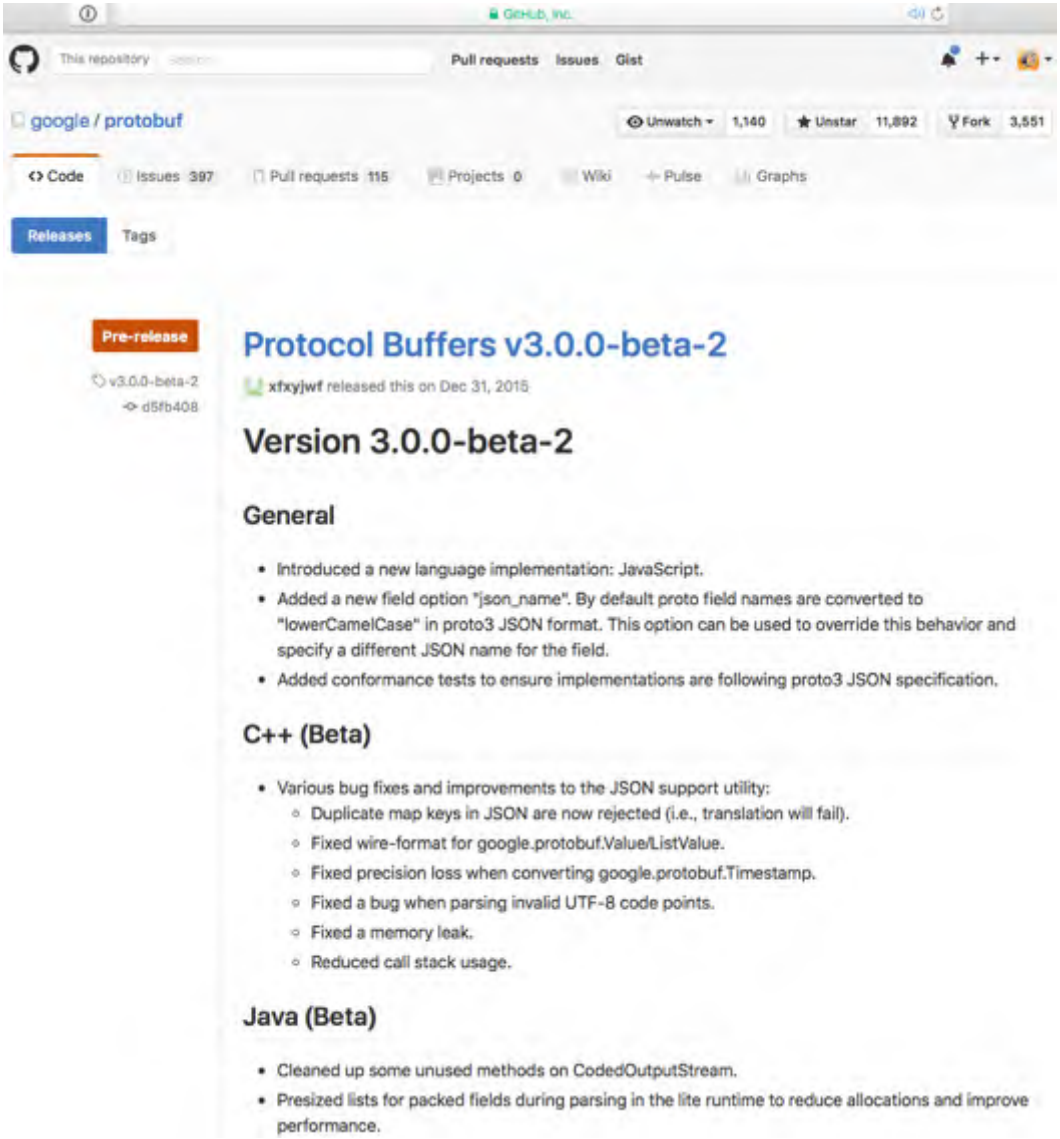

字节长度
350
300
250
200
150
100
50
0
Xml
Json
Hessian
Proto Buf
字节长度

主流协议对比：PB的序列化、反序列化、创建综合性能高。



使用 ProtocolBuffer 原因

- 省流量
- 高效
- 省电
- 成熟可靠
- 易于使用


GitHub, Inc.
This repository
Pull requests
Issues
Gist
google / protobuf
Unwatch 1,140
Unstar 11,892
Fork 3,551
Code
Issues 397
Pull requests 116
Projects 0
Wiki
Pulse
Graphs
Releases
Tags
Pre-release
v3.0.0-beta-2
d5fb408
Protocol Buffers v3.0.0-beta-2
xfxyjwf released this on Dec 31, 2015
Version 3.0.0-beta-2
General
Introduced a new language implementation: JavaScript.
Added a new field option "json_name". By default proto field names are converted to "lowerCamelCase" in proto3 JSON format. This option can be used to override this behavior and specify a different JSON name for the field.
Added conformance tests to ensure implementations are following proto3 JSON specification.
C++ (Beta)
Various bug fixes and improvements to the JSON support utility:
Duplicate map keys in JSON are now rejected (i.e., translation will fail).
Fixed wire-format for google.protobuf.Value/ListValue.
Fixed precision loss when converting google.protobuf.Timestamp.
Fixed a bug when parsing invalid UTF-8 code points.
Fixed a memory leak.
Reduced call stack usage.
Java (Beta)
Cleaned up some unused methods on CodedOutputStream.
Presized lists for packed fields during parsing in the lite runtime to reduce allocations and improve performance.

- 在安全上需要做哪些事情？
  - 防止 DNS 污染
  - 防止 DDos 攻击
  - 账户安全
    - 帐号安全
    - 签名机制
    - 单点登录


迎使用360WiFi网络
web.free.wifi.360.cn/portal/andindex?wifiname=VenusFD&url=http%3A%2F%2F172.26.206.1%3A8087%2Fgoto%3F21EC476DC85045EB77DFDB7E1D3E449
超91%的企业WiF
隐私泄露
存在DNS劫持
建议使用『360免费WiFi』APP安全检测


应用鉴权服务器
1
2
• 用户登录
• 新建对话
• 加入群组
• 邀请他人
• 踢出成员
0
3
4
LeanMessage
Cluster
终端

防DNS污染

- DNS解析时间过长
- DNS劫持
  - 基于 UDP 不可靠
  - 域名中包含敏感词

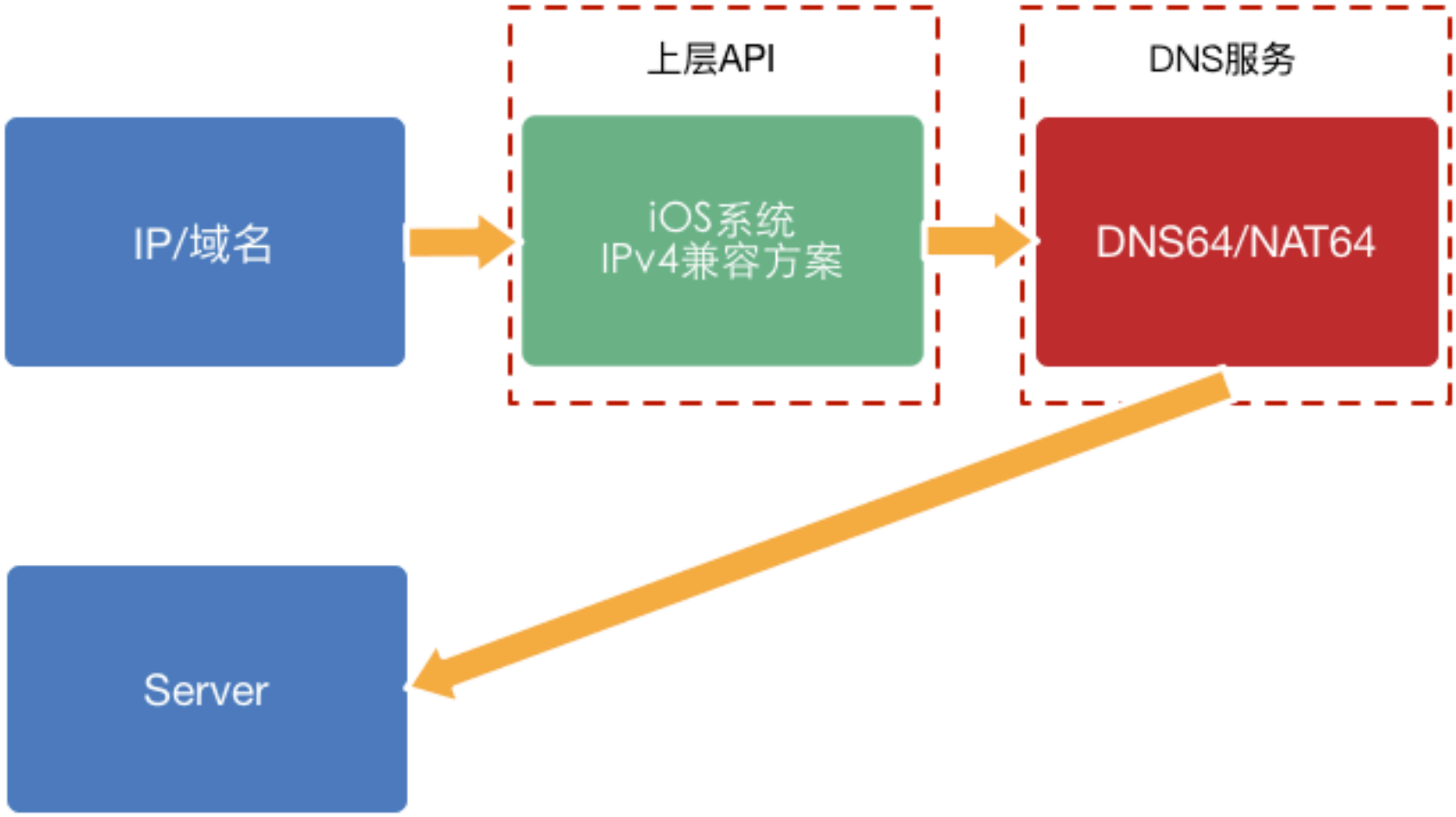
- avoscloud.com
- leancloud.cn

- 传统的解决方法：投诉

防 DNS 污染方案

- HTTP 场景 IP 直连
- 客户端动态部署 IP 列表
- 使用基于 HTTP 的 DNS 解析方案

方案一：HTTP 场景 IP 直连



方案二：客户端动态部署 IP 列表

- 维护IP列表(电信、移动、联通，异步DNS解析)
- 无效映射淘汰机制，请求成功就+1、失败就-1
- 也可以根据网络延迟选择服务端 IP

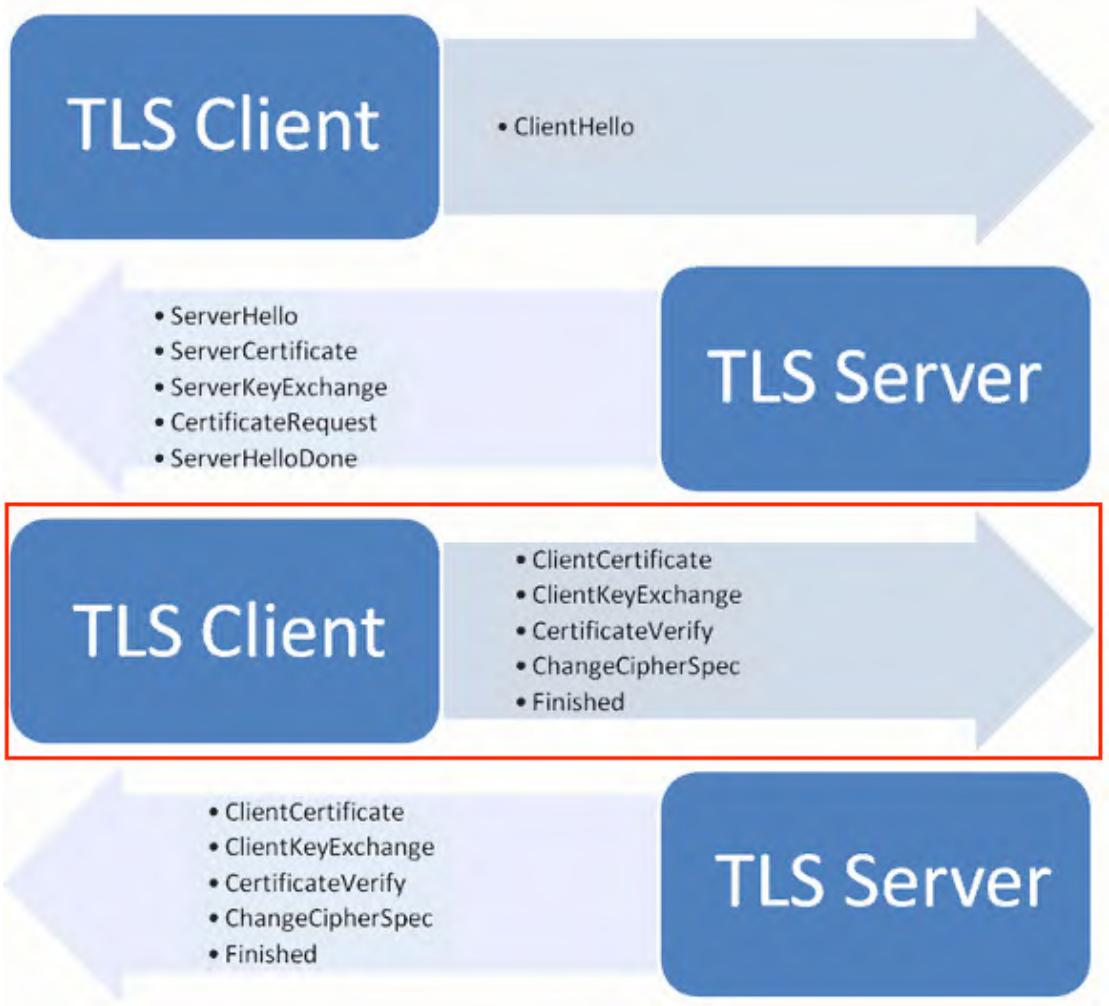
方案三：使用基于 HTTP 的 DNS 解析方案

- 绕过运营商直接连可以信任的第三方服务。

图中首选和备选的优先级可以调整。

- **实现时的问题**

发送HTTPS请求首先要进行SSL/TLS握手，握手过程大致如下：



解决方法：

- 如果使用第三方网络库：curl，原生支持
- NSURLSession、NSURLConnection需要hook住SSL握手方法，AFN需要被改源码重写。

对应于下面的代理方法：

`-connection:**willSendRequestForAuthenticationChallenge**:`
`-URLSession:task:**didReceiveChallenge**:completionHandler:`

优化重连机制

- 精简心跳包
- 减少心跳次数
- 重连冷却

注：这样灵活的策略也同样决定了，只能在 APP 层进行心跳ping。

使用 HTTP/2 减少不必要的网络连接

- 必然要使用 NSURLSession。
- AFN2.x 也需要升级到AFN3.x.

- 设置合理的超时时间

图片视频等文件上传

- 文件分块上传
- 提供文件秒传的方式
- 支持断点续传
- 上传失败，合理的重连，比如3次。

使用缓存：类似 E-Tag 的本地消息缓存校验

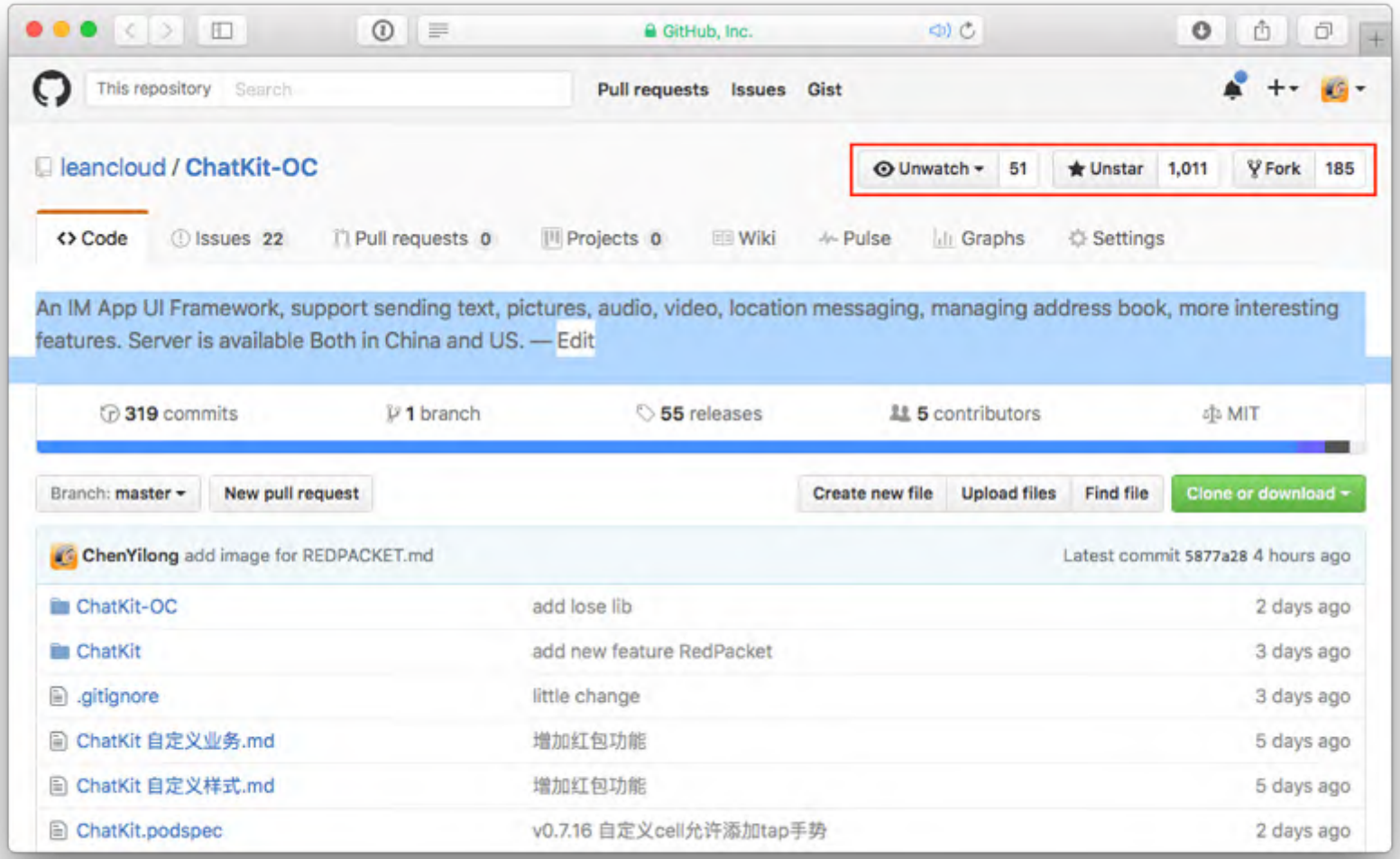
- 普通消息本地缓存
- 使用hash校验，保证消息同步

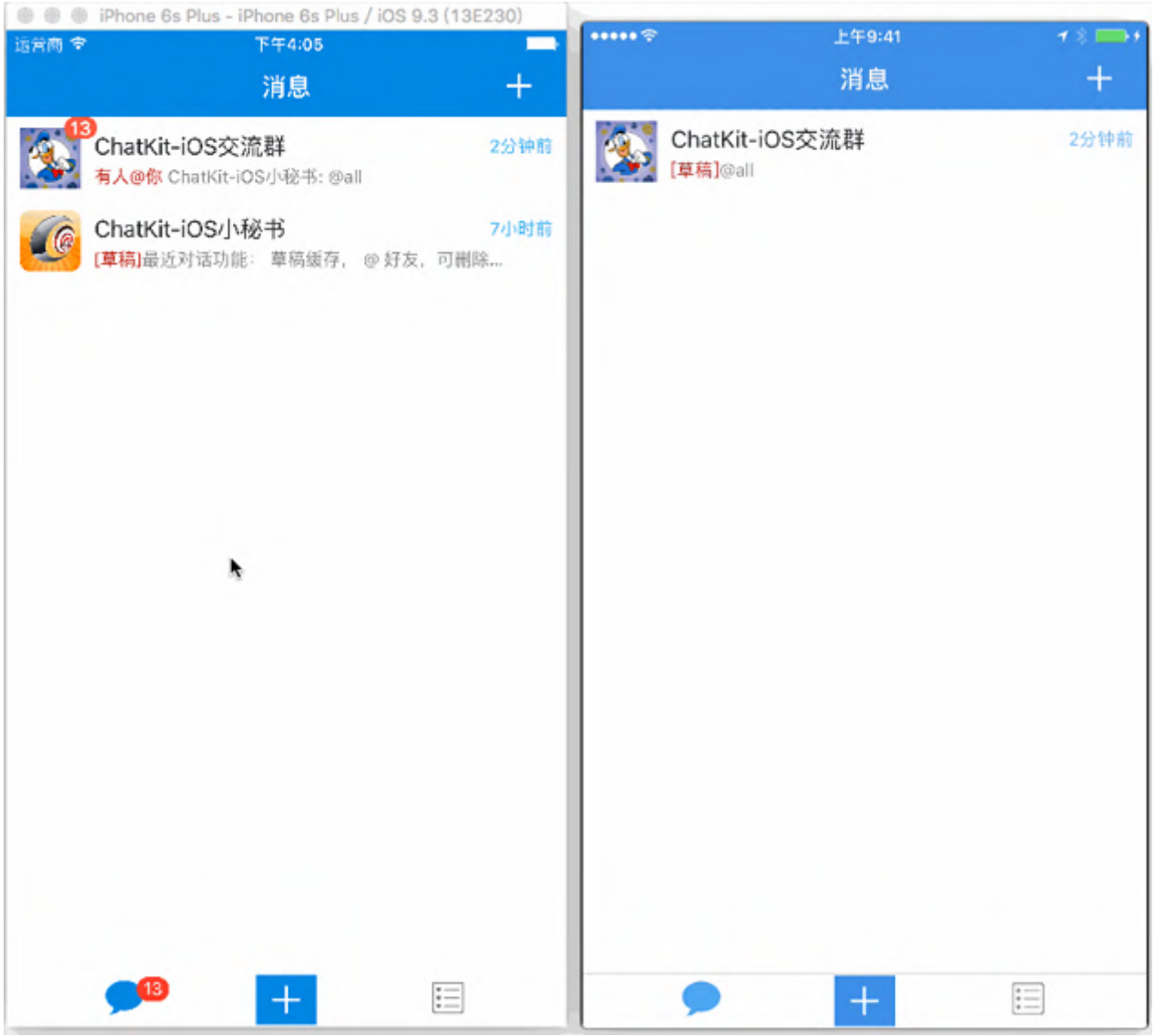
第三部分:
技术实现细节

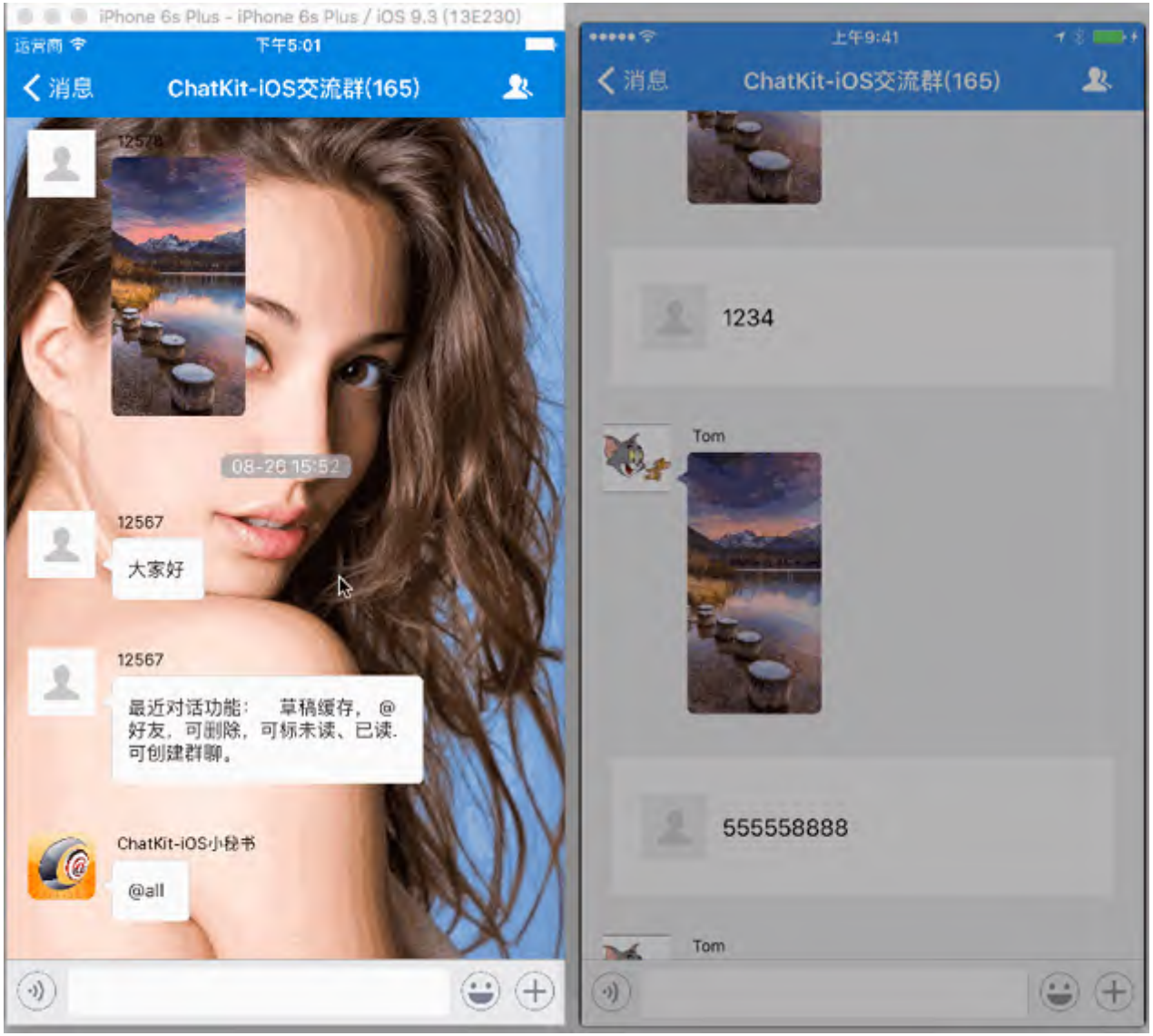
社区现状

- Demo
- 闭源
- 部分开源
- 非原生
- 手撕 Frame
- 自定义能力太弱的太多

- 为社交场景开发的开源组件：[ChatKit-OC]

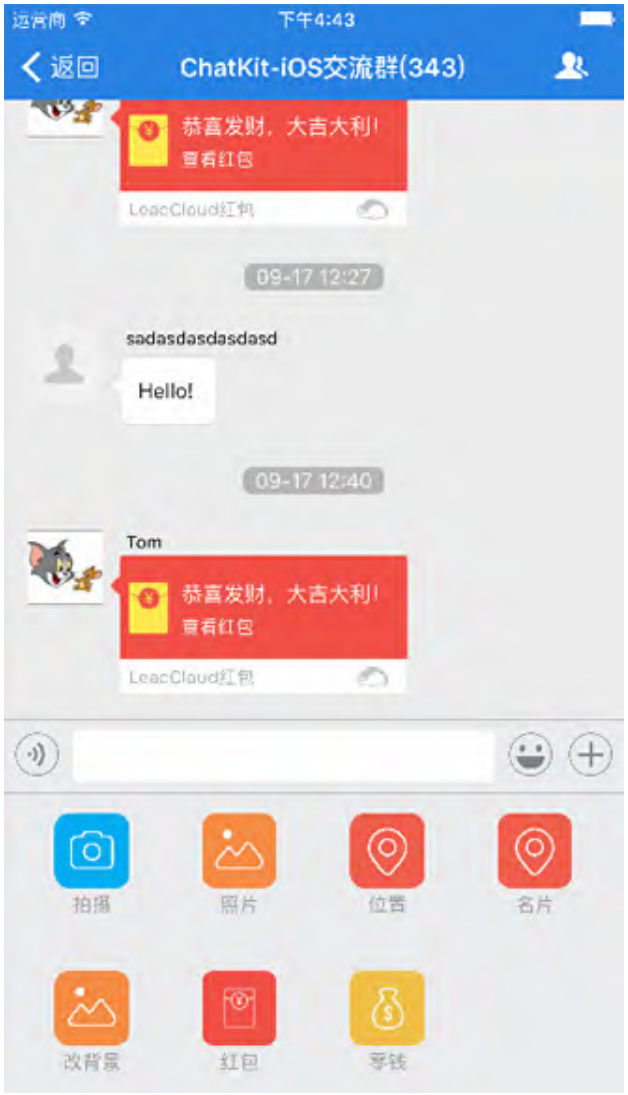



iPhone 6s Plus - iPhone 6s Plus / iOS 9.3 (13E230)
运营商
下午4:05
消息
13
ChatKit-iOS交流群
2分钟前
有人@你 ChatKit-iOS小秘书: @all
ChatKit-iOS小秘书
7小时前
[草稿]最近对话功能： 草稿缓存， @ 好友，可删除...
13
上午9:41
消息
ChatKit-iOS交流群
2分钟前
[草稿]@all


iPhone 6s Plus - iPhone 6s Plus / iOS 9.3 (13E230)
下午5:01
消息
ChatKit-iOS交流群(165)
08-26 15:52
12567
大家好
12567
最近对活功能： 草稿缓存， @好友，可删除，可标未读、已读. 可创建群聊。
ChatKit-iOS小秘书
@all
上午9:41
消息
ChatKit-iOS交流群(165)
1234
Tom
555558888
Tom


影片录制
消息
ChatKit-iOS交流群(165)
1
2
失败消息可缓存重发，飞行模式下不crash，不白屏，表现稳定
1
0 KB
失败消息可缓存重发，飞行模式下不crash，不白屏，表现稳定

使用 CocoaPods 集成，在 Demo 层面能实现红包：




运营商
下午4:16
消息
ChatKit-iOS交流群(343)
Simon
Tom
发了一个红包，金额随机
恭喜发财，大吉大利！
拆红包


运营商
下午4:43
关闭
红包
Tom的红包
恭喜发财，大吉大利！
1个红包，4秒被抢光
Tom
11:55
0.01元
手气最佳
查看我的红包记录

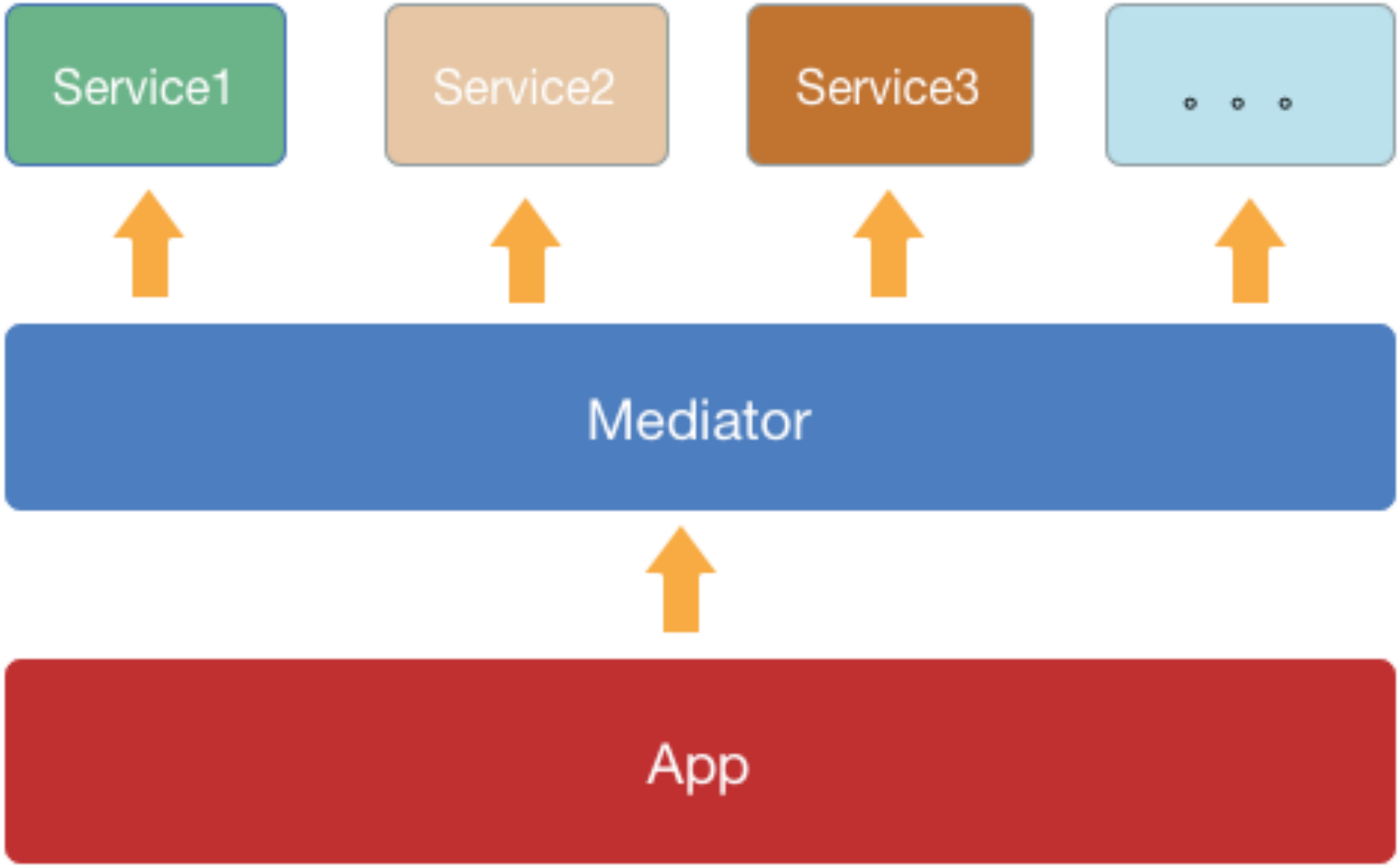
特点：

- 易于集成，扩展性好。
- 100%开源（iOS端）
- 原生语言开发，利于调试
- Masonry 布局
- 友好的 API 设计
- 接地气
- 支持 CocoaPods
- 不需要改源码，不需要设 Delegate
- 不需要在代码里调整聊天气泡位置

API设计：

- 紧凑的
- 可维护性强
- 无侵入的用户系统接入
- 面向 ID 编程
- 可拓展性强
- 封装程度高

紧凑的API设计：门面模式



可维护性强

- **传统做法：**
  - 直接写坐标可维护性太差
  - 定义成宏，无法满足组件化后的自定义需求
  - 失去动态更新能力
- 现在的方案：可自定义的配置文件。

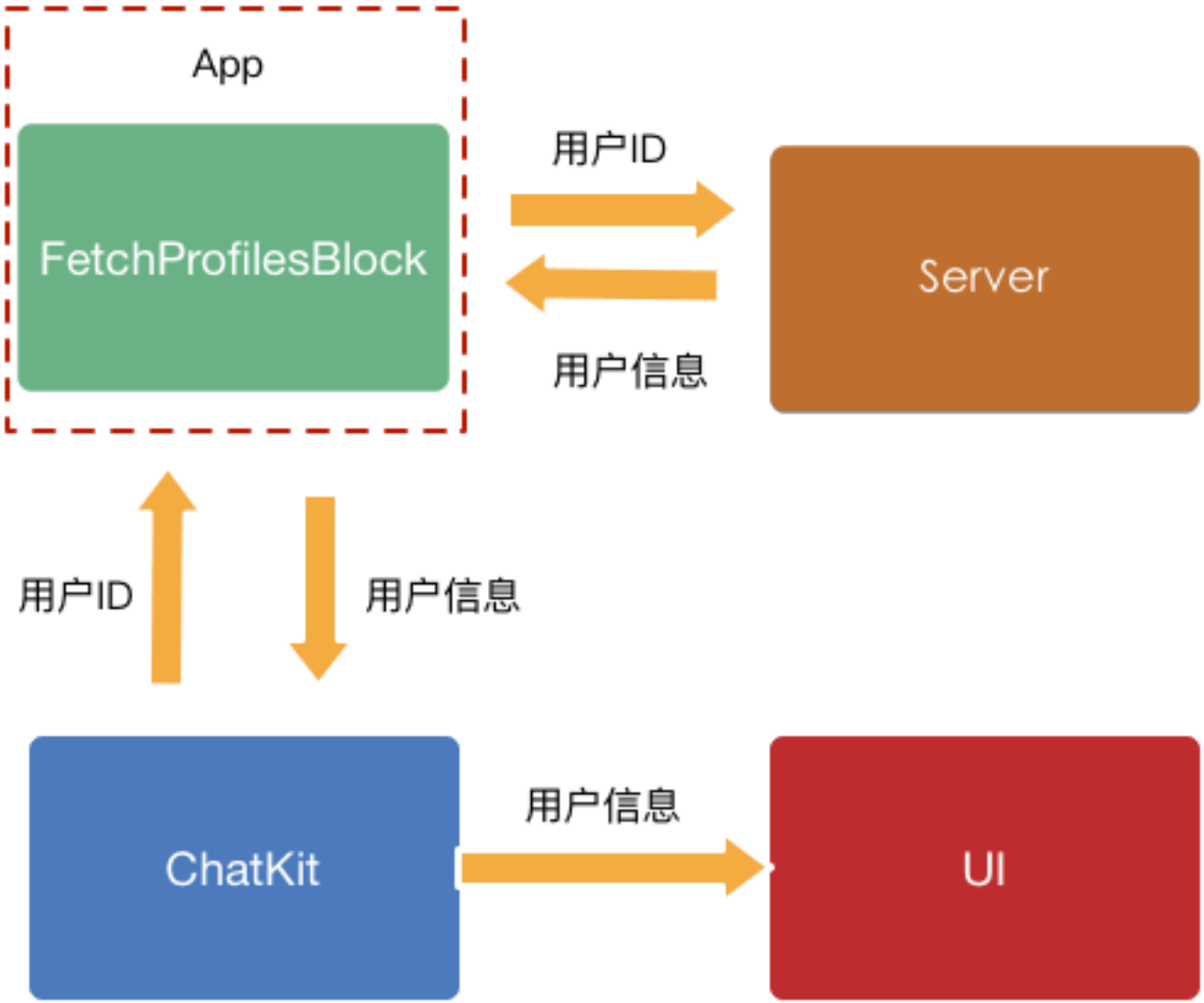
- 无侵入的用户系统接入

俄罗斯即时通讯QIP.ru 3300万明文密码被盗

E安全 / 数据泄露 / 2016-09-09 13:00

E安全9月9日讯 黑客盗取QIP.ru超过3300万用户记录。QIP.ru是俄罗斯一家即时通讯服务公司。

网络安全公司HEROIC将数据样本发送给了媒体Softpedia。HEROIC公司负责保护用户免受黑客攻击和网络威胁。


App
FetchProfilesBlock
用户ID
用户信息
Server
用户ID
用户信息
ChatKit
用户信息
UI

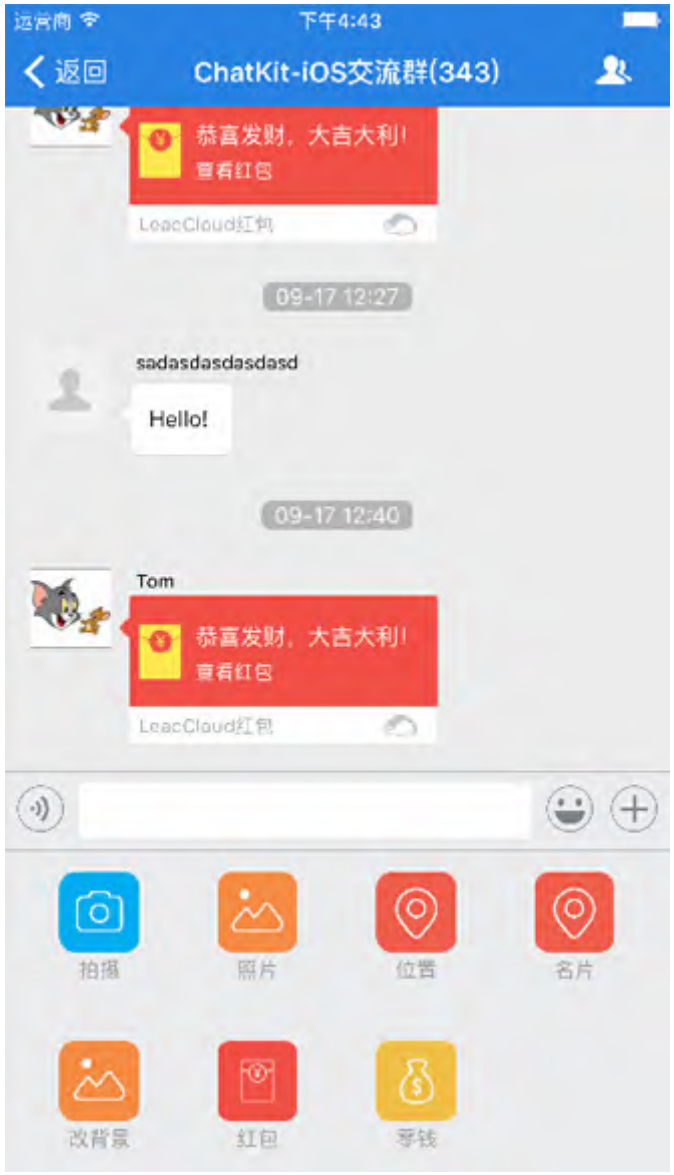
面向 ID 编程

- ClientID
- ConversationID
- PeerID

可拓展性强：

插件映射机制

- 自定义cell类型
- 自定义Message类型
- 自定义输入框底部插件



- **封装程度高**
- 自定义消息
  - 业内：操作字符串
  - ChatKit：Model

结束。
Q-A

https://github.com/ChenYilong

http://weibo.com/1692391497