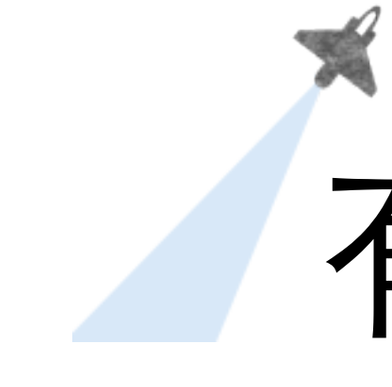
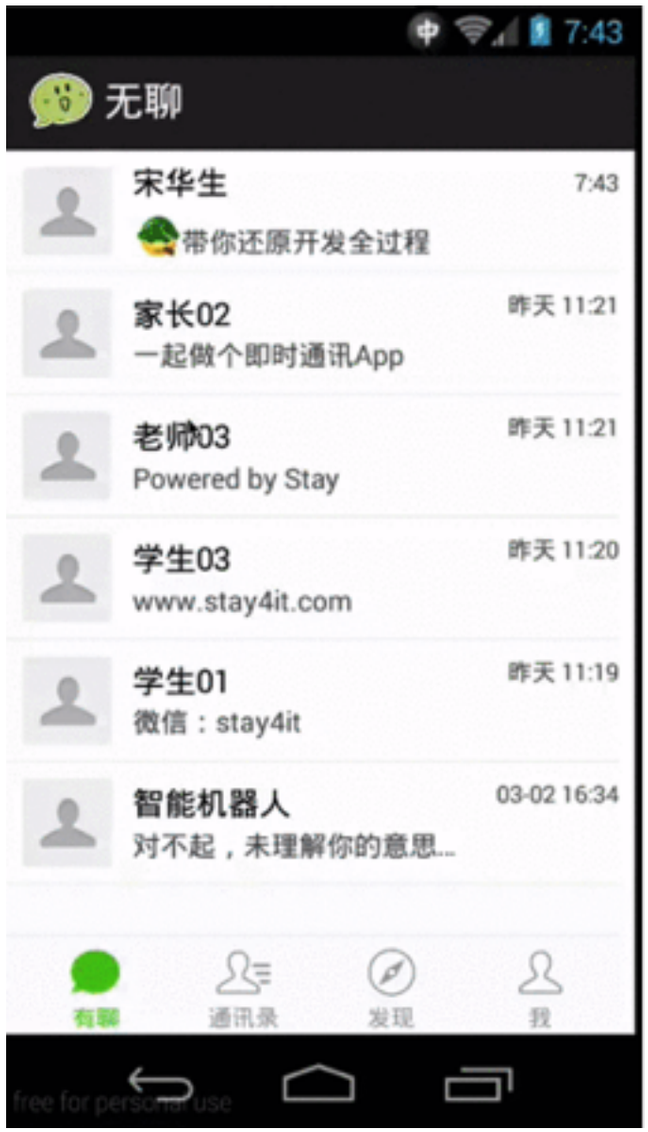


有心创客

传递给你的不仅仅是技术

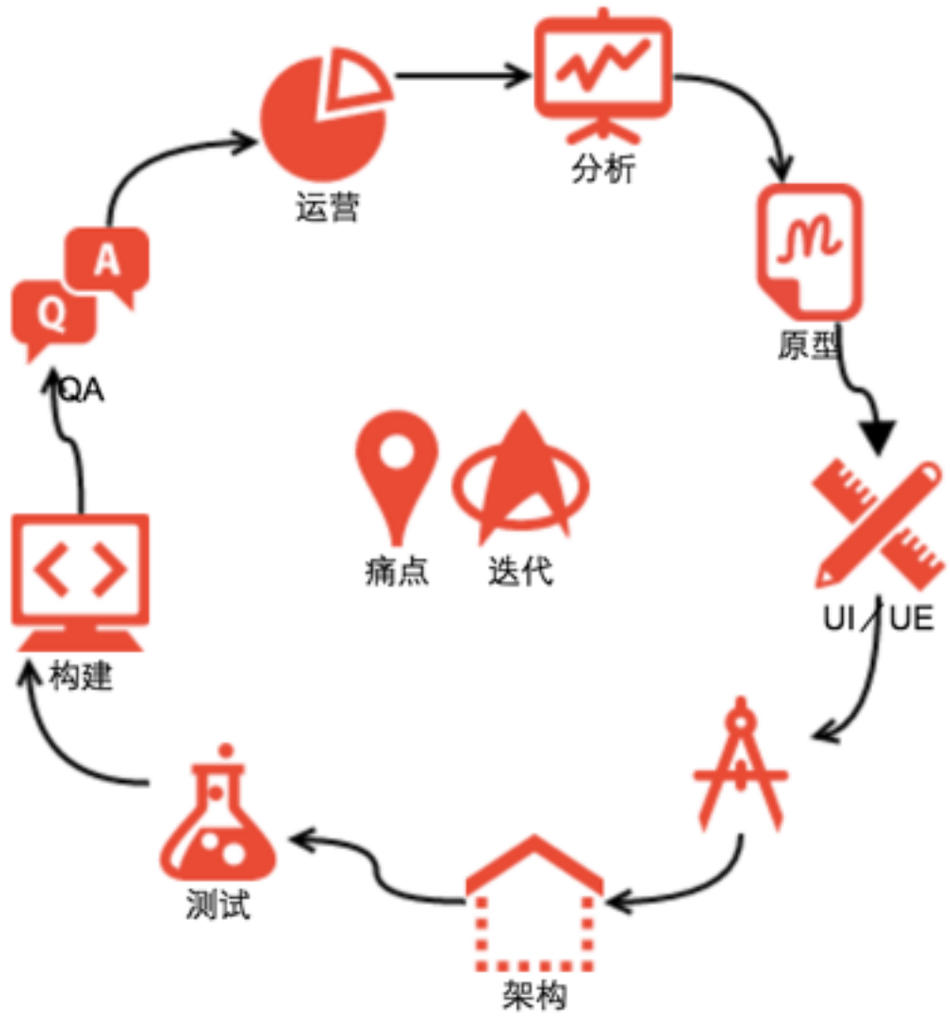
一起做个即时通讯App



Powered by Stay

App生命周期

- What role do we play?



Stay会讲些什么？

- 还原开发过程（分析 – 设计 – 架构 – 测试 – 实现）。比编码更重要的是思维养成。

- App从0到1的构建过程，通过IM课程来体现。

- 优秀攻城师是怎样炼成的。

只听不想不实践
什么都学不会

分析丨设计

- 即时通讯App概要
- IM方案科普
- 消息同步方案
- 数据库设计

架构 | 测试

消息收发模型

编码实现

- 业务逻辑
- 代码风格
- 编码规范
- UI自定义

简历丨面试

- 职责
- 难点

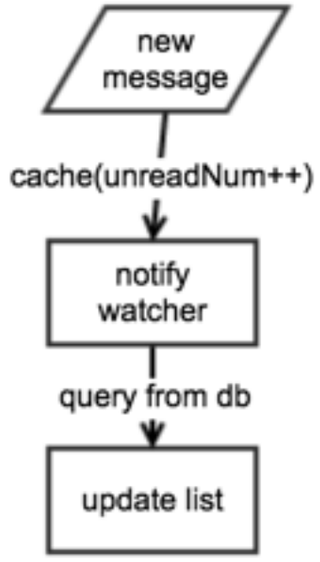
- 即时通讯APP
  - 模块划分
    - 登陆／注册
    - 会话列表
    - 通讯录
    - 聊天detail
    - 用户profile
  - 业务逻辑
    - 推送方案
    - 消息收取－传递
    - 发送消息状态更新
    - 消息同步机制
    - 数据库设计
    - 群组聊天方案
    - 服务器整体方案
    - 方案选择：多设备单账号同时在线／单账号唯一在线
  - 知识点
    - 主架构（聊天IM框架课程）
    - http请求／上传（自己动手写http框架）
    - 数据缓存（ormlite）
    - 表情商店（自己动手写多任务下载框架）
    - 消息类型
      - 文本
      - 图片
      - 表情
        - emoji表情
      - 语音
        - NDK speex/ilbc编解码
      - location
    - IM协议
      - 第三方云推送sdk
      - 更换方案不影响整体架构
    - profile复用（edit/detail）
    - 零散知识点，lib使用不再赘述
  - 耳濡目染
    - 高效编码
    - 代码规范
    - 需求分析
    - 团队协作
    - 任务分配

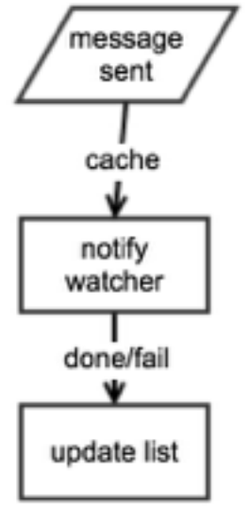
IM方案

- Socket （自建协议）
- XMPP （openfire + mina）
- 第三方云推送

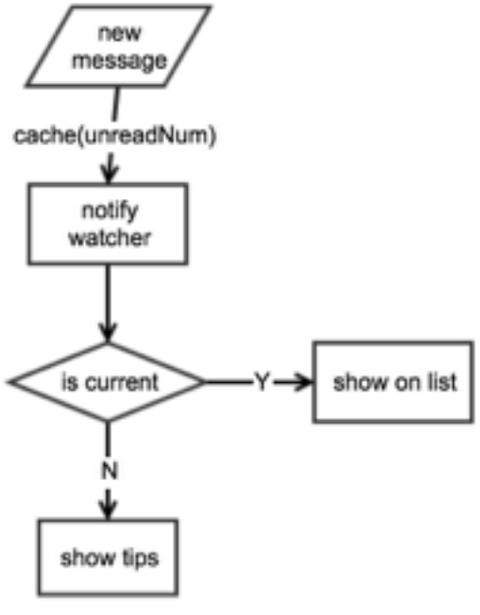
消息收发

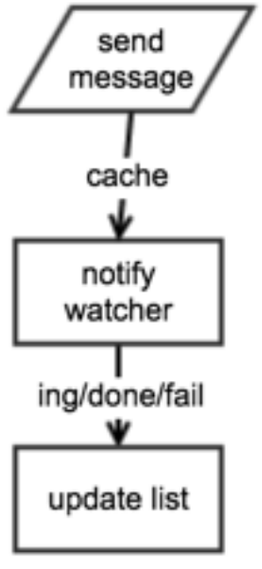
会话列表(Conversation)





聊天详情(Chat)

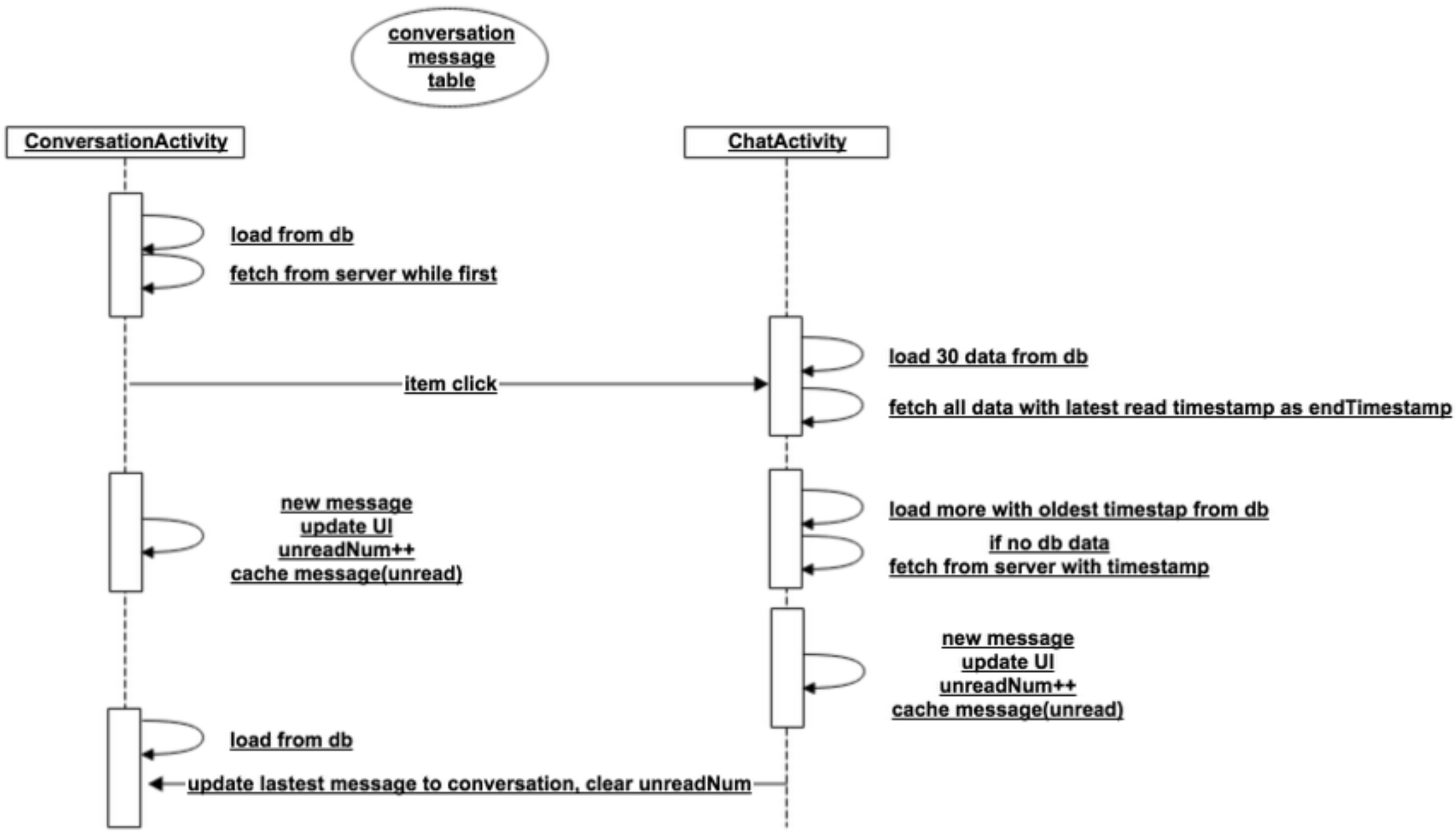




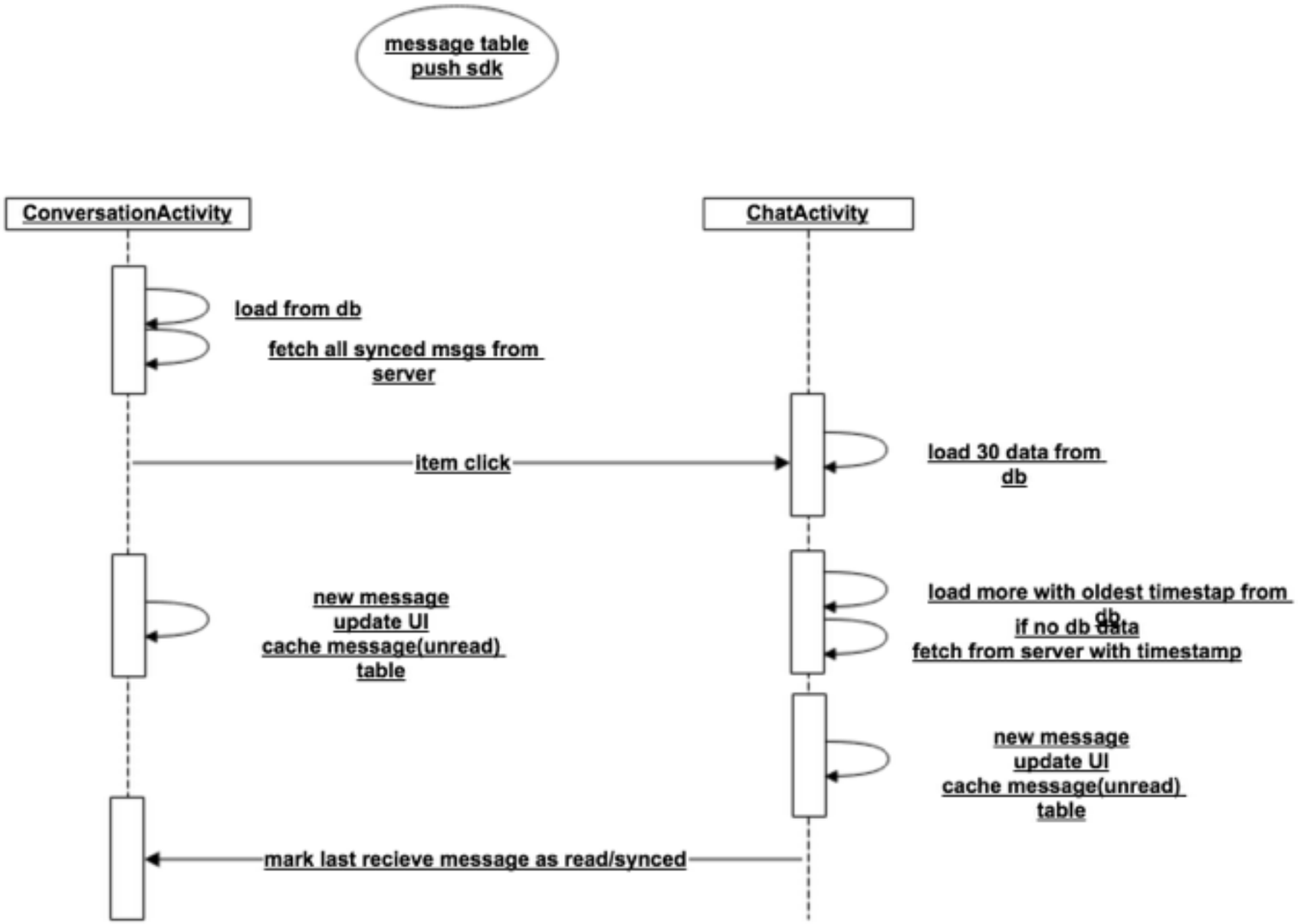
消息同步方案

- 成因：local db && server db – – 网络异常
- 不同IM方案下的不同取舍
  - 管收不管发:
    - 点对点同步
    - 整体同步
  - 管收管发：整体同步

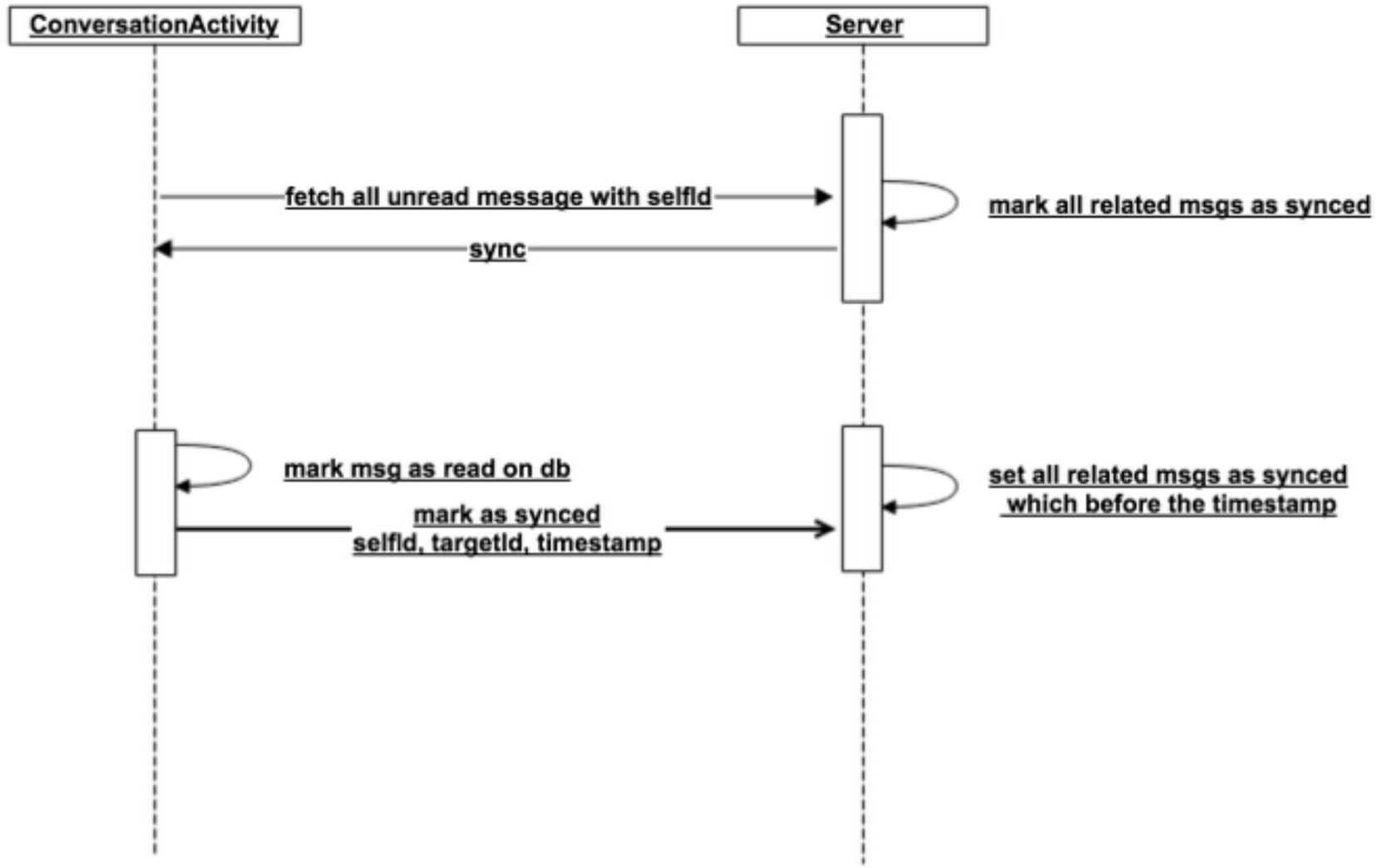
方案1


conversation
message
table
ConversationActivity
ChatActivity
load from db
fetch from server while first
item click
load 30 data from db
fetch all data with latest read timestamp as endTimestamp
new message
update UI
unreadNum++
cache message(unread)
load more with oldest timestap from db
if no db data
fetch from server with timestamp
new message
update UI
unreadNum++
cache message(unread)
load from db
update lastest message to conversation, clear unreadNum

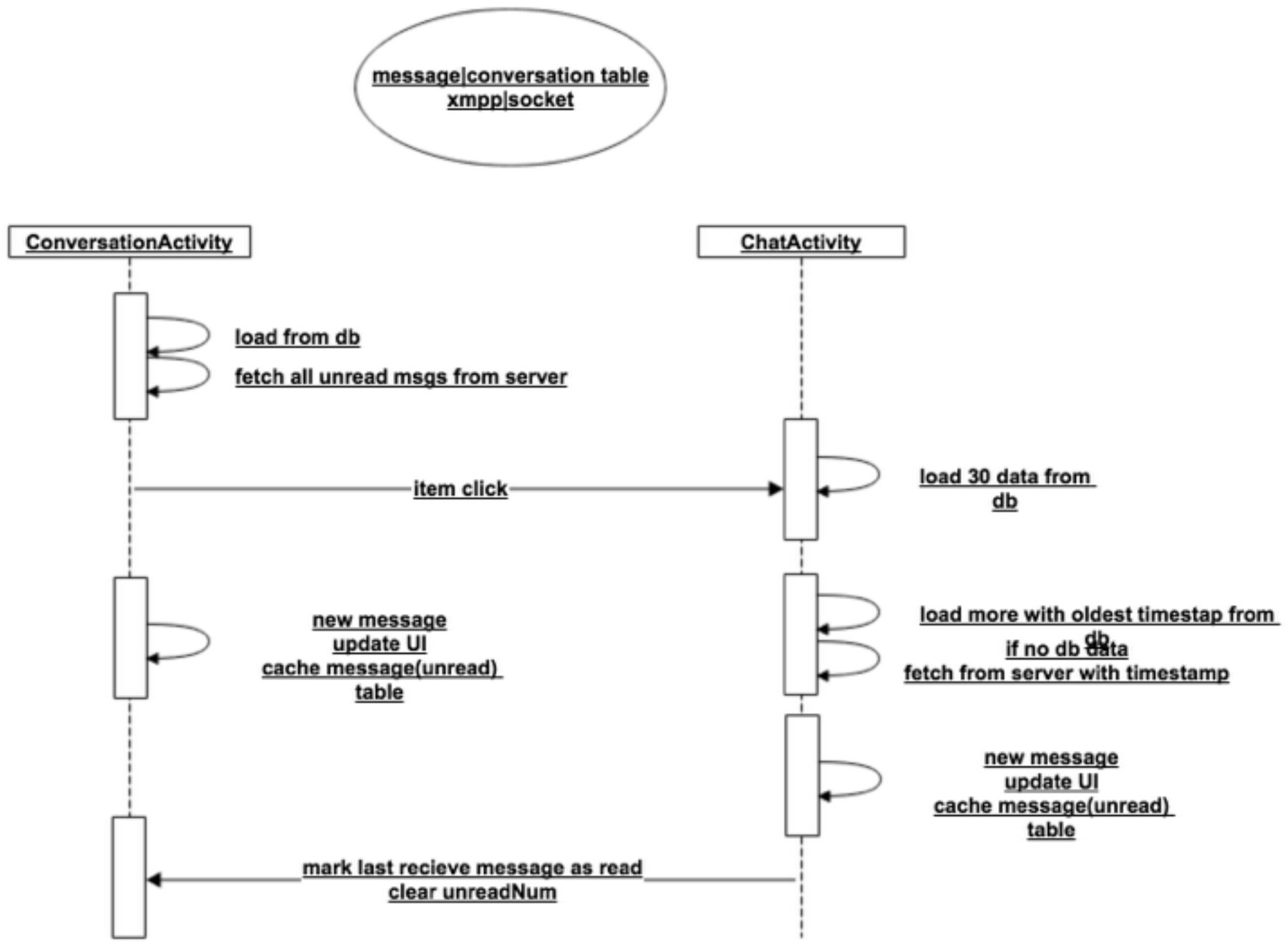
方案2


message table
push sdk
ConversationActivity
ChatActivity
load from db
fetch all synced msgs from server
item click
load 30 data from db
new message
update UI
cache message(unread) table
load more with oldest timestap from db
if no db data
fetch from server with timestamp
new message
update UI
cache message(unread) table
mark last recieve message as read/synced

方案2


ConversationActivity
Server
fetch all unread message with selfId
mark all related msgs as synced
sync
mark msg as read on db
mark as synced
selfId, targetId, timestamp
set all related msgs as synced
which before the timestamp

方案3

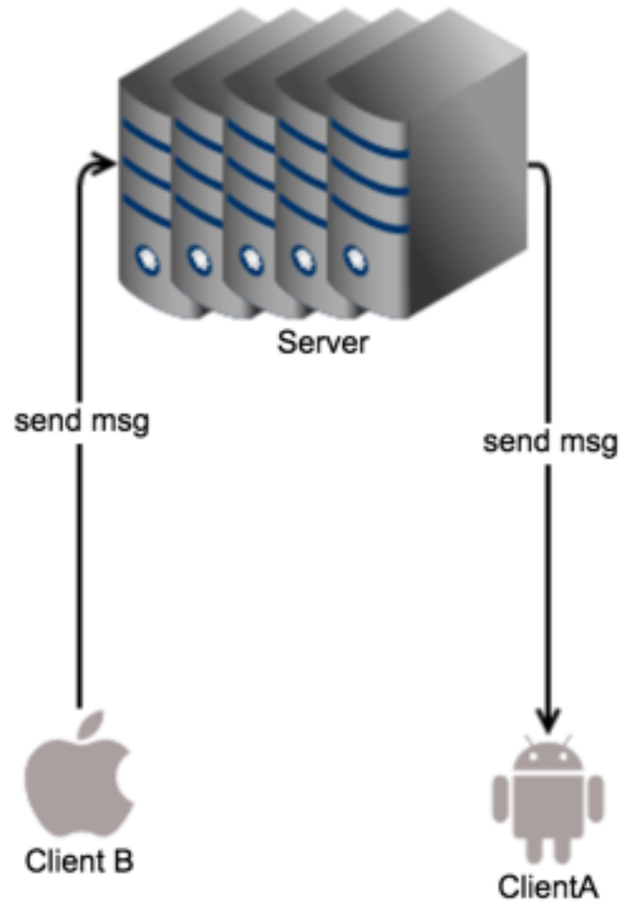

message|conversation table
xmpp|socket
ConversationActivity
ChatActivity
load from db
fetch all unread msgs from server
item click
load 30 data from db
new message
update UI
cache message(unread) table
load more with oldest timestap from db
if no db data
fetch from server with timestamp
new message
update UI
cache message(unread) table
mark last recieve message as read
clear unreadNum

数据库设计

- 是否存在回话列表这张表?
- 关于好友昵称，头像的更新

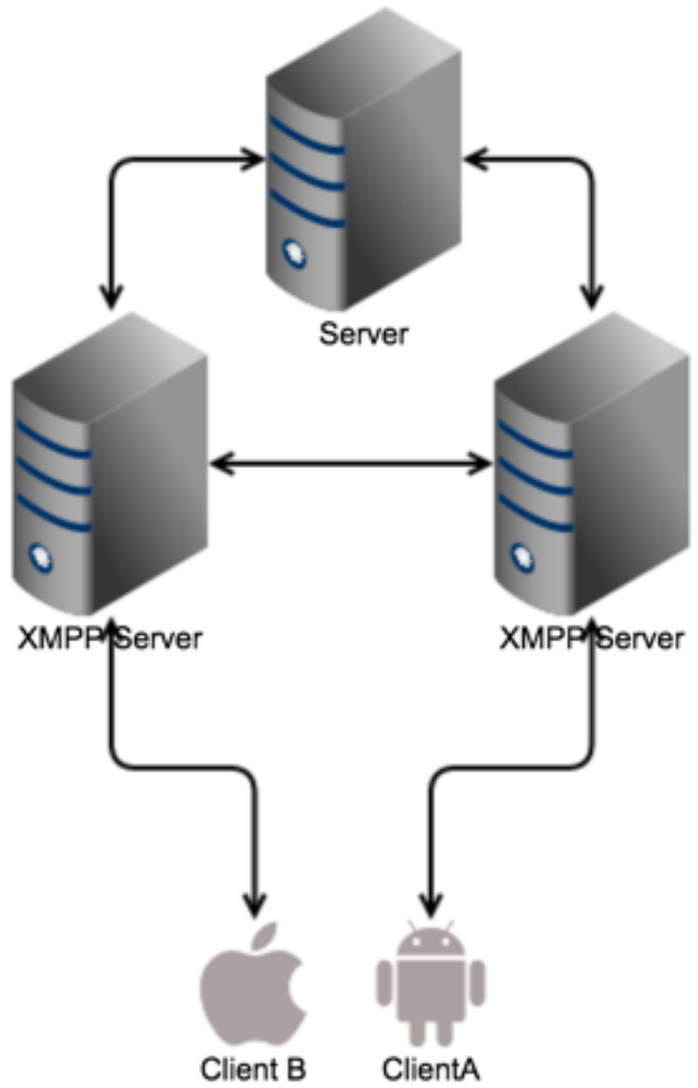
Socket

- 自建协议 （头信息丨长度丨消息）
- 维护长链接的代价
- 管收管发，业务逻辑最简单



XMPP

- openfire + mina （XML）
- 文本是个什么鬼



第三方云推送

- 配置简单
- 服务器无压力
- 省电

VS

- 即时率，到达率，准确率
- 时间限制
- 安全
- 数据限制