Grouk

移动时代的多终端同步协议设计与探索

王渊命
Grouk
@jolestar

终端爆发

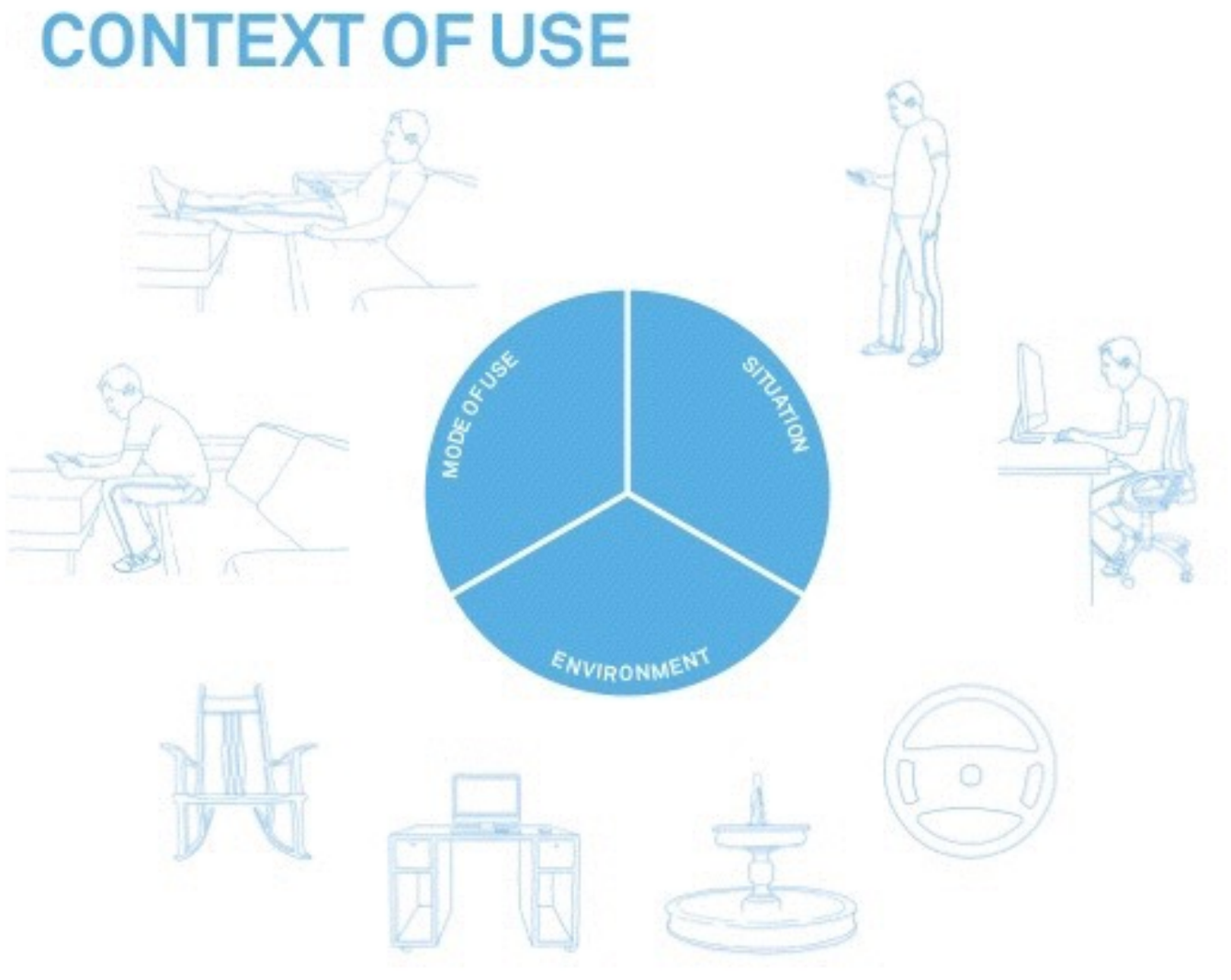
- 手机
- 平板电脑
- PC
- TV
- 智能设备(手表,手环等)

图片来源:http://themultiscreenblog.com/multiscreens-4-biggest-challenges-2015/

跨屏体验

- 终端切换时保持一致性体验，不丢失上下文
- 支持多个终端同时操作，终端之间需要实时通信，同步数据

用户场景



图片来源: http://www.slideshare.net/digiparden/multiscreen-experiencedesign-shortpresentationenjuly2013

典型案例

- Quip 多人协作文档编辑工具
- Trello 任务看板

应用架构模式变化

- BS—->CS转变
- 拉模式(http)到推模式(comet，websocket, 长连接)
- 内置实时通讯机制

Grouk

传统的IM协议

- Instant Message 即时的/瞬间的/不可变的
- 基于到场(Presence)机制
- 在线终端直接投递
- 离线消息一般只能一个设备拉取
- rfc2778/rfc2779
- XMPP(rfc3921)/SIMPLE
- SMC定理，Single-Message Communication(任何端到端的消息传递协议，消息既不丢失，也不重复是不可能的)

多终端同步机制

- 数据是可变对象
- 在线设备实时投递或通知(增强功能)
- 设备可增量同步
- 多终端数据冲突合并机制
- 数据最终一致

业界同步协议

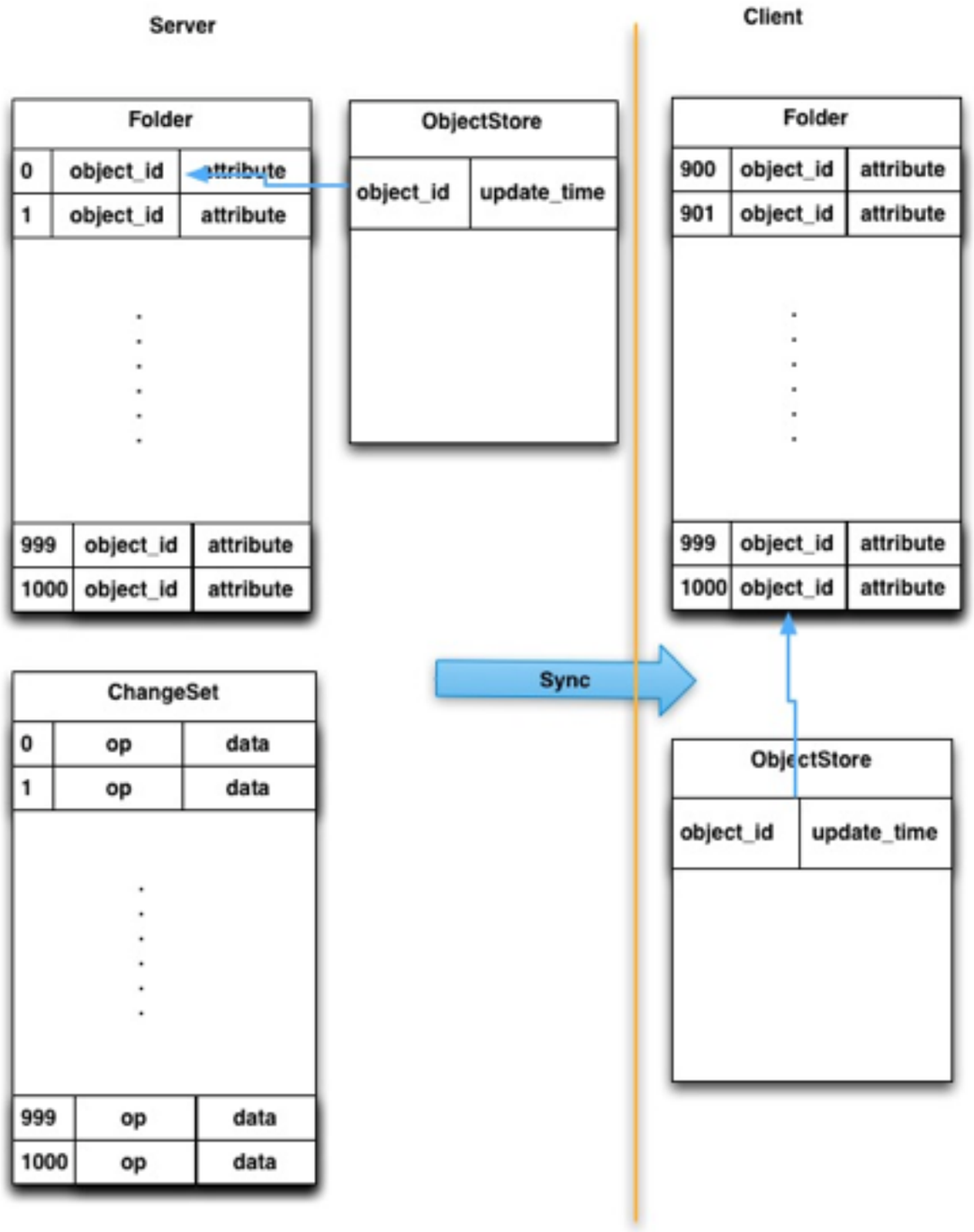
- 版本控制系统(git/svn)
- CalDAV/CardDAV/WebDAV
- SyncML
- Exchange ActiveSync（Mail, Calendar and Address Book)
- WeiSync（微信)

Grouk

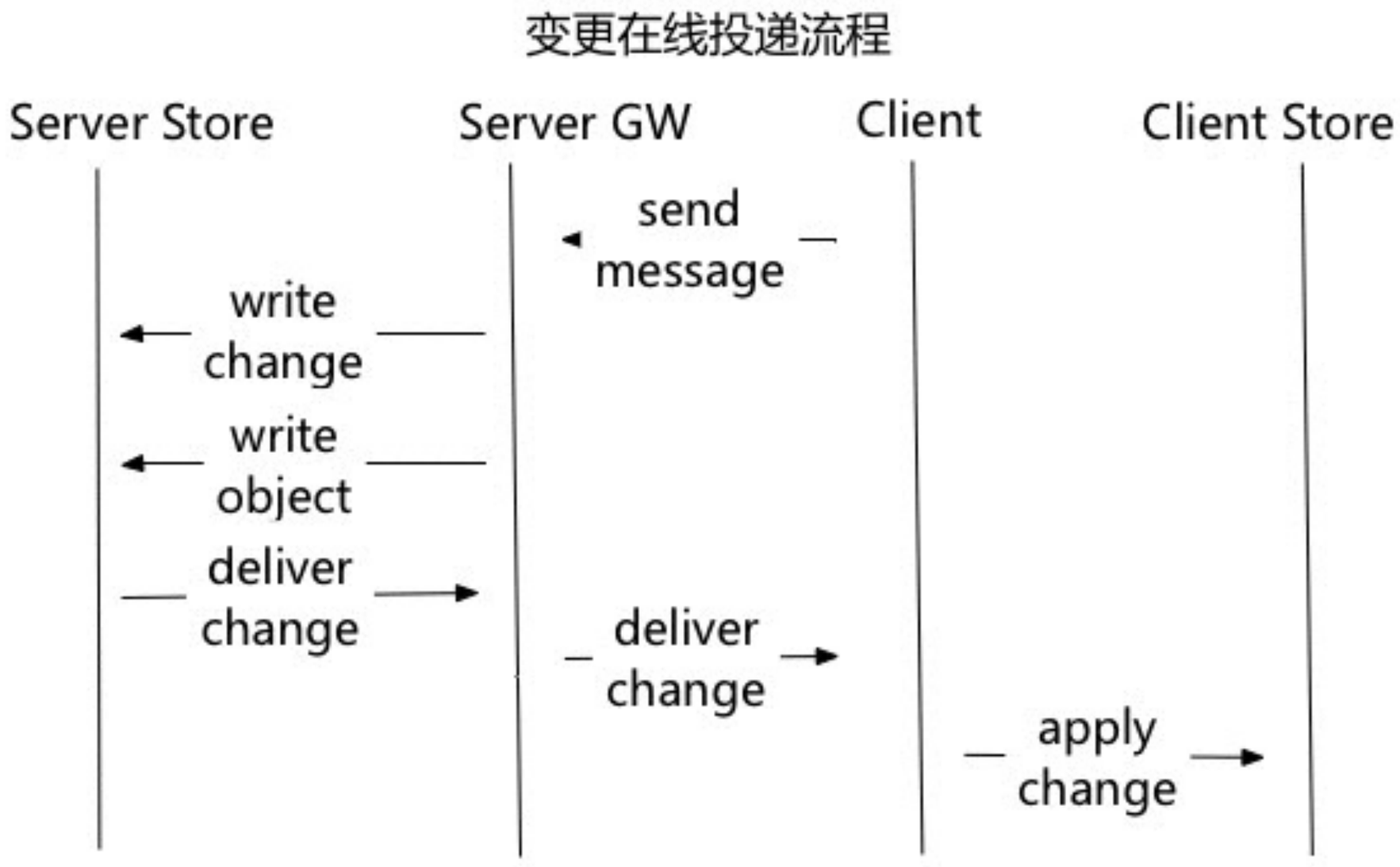
Grouk同步协议实践

- 记录对象变更历史
- 实时投递变更而不是数据对象
- 客户端通过回放变更来更新数据
- 支持多种数据对象(联系人/消息/群组)
- 冲突解决(新数据优先)

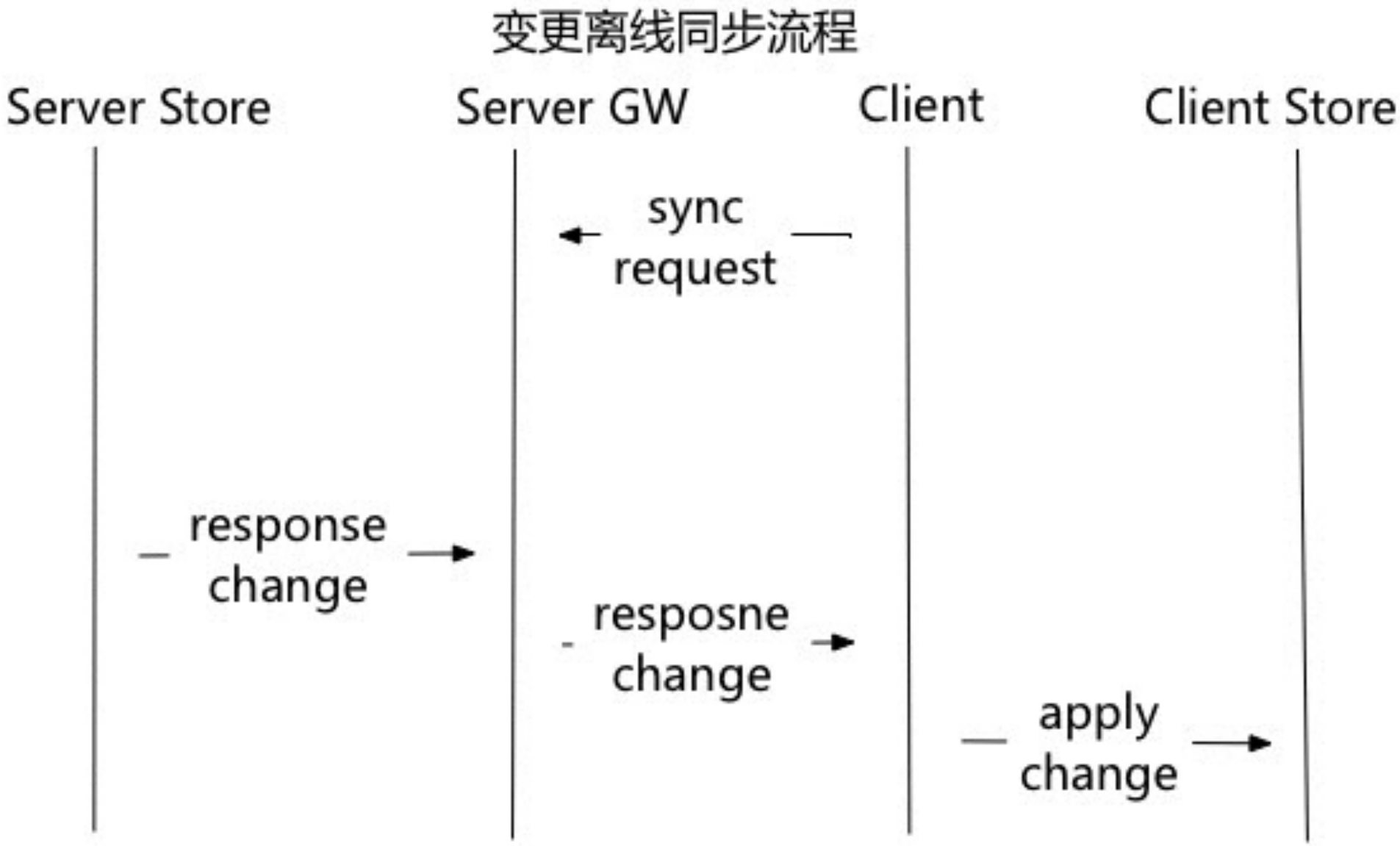
Grouk同步协议-数据结构



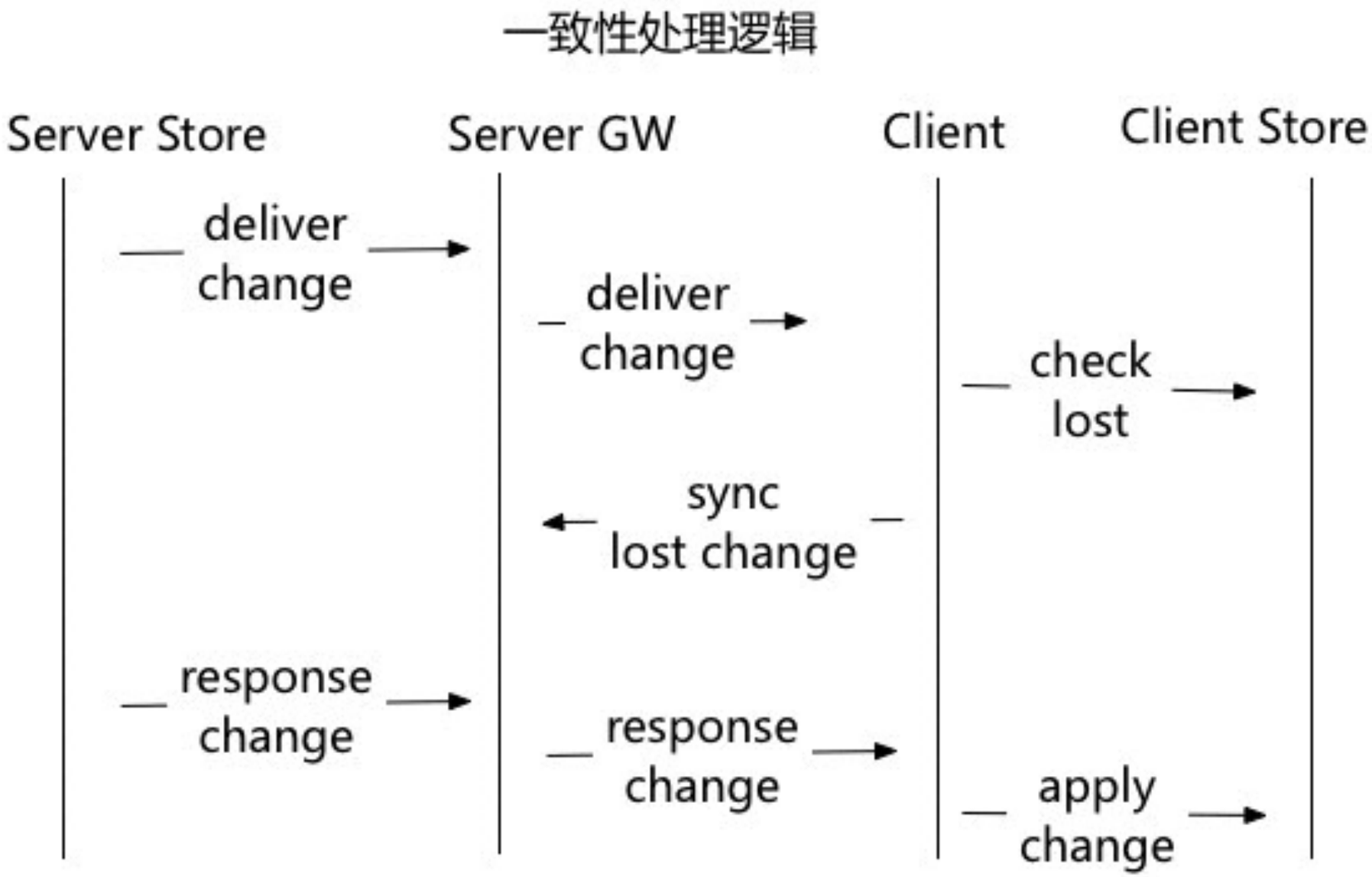
Grouk同步协议-投递流程



Grouk同步协议-同步流程



Grouk同步协议-一致性



Folder索引设计

- 会话
- 用户的会话列表
- 用户加入的群
- 团队联系人
- 收藏列表
- 已读未读状态处理

本地缓存以及省流量

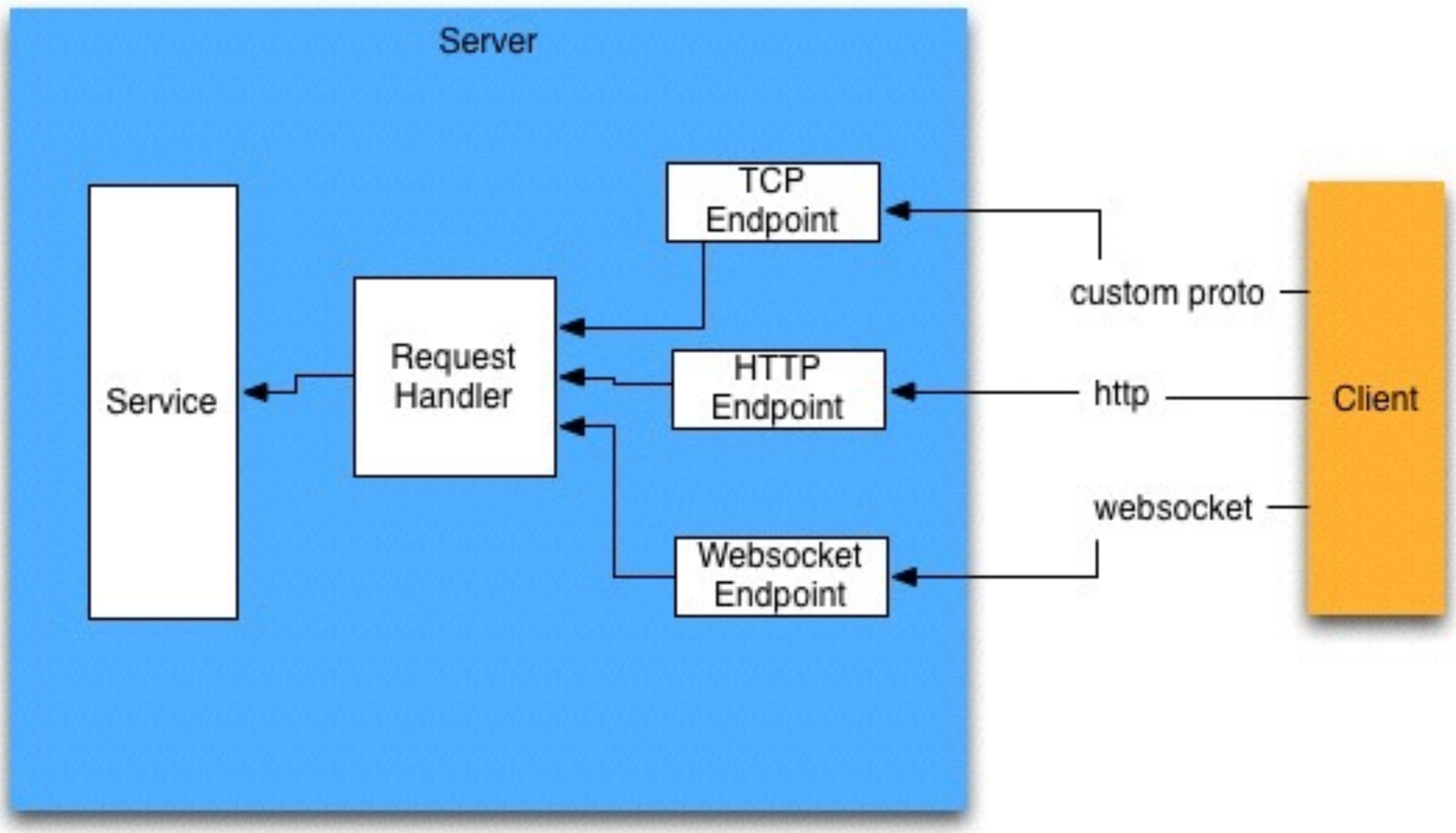
	Web	App
http缓存机制(ETag/Expires)	浏览器内置	无
gzip压缩	浏览器内置	自己实现
缓存对象	页面/图片	数据对象(user/group)
统一的标准	有	无

本地缓存以及省流量

- 统一的数据对象设计
- 数据对象本地缓存过期机制
- 列表接口只返回id
- 接口支持Protobuf和json

Grouk

网关架构


Server
TCP
Endpoint
Request
Handler
Service
HTTP
Endpoint
Websocket
Endpoint
custom proto
http
websocket
Client

Grouk

实现效果

- 多终端数据保持一致，用户切换后不会丢失上下文
- 允许多个终端登录，比如，多个手机、多个Web
- 历史记录可以在任何一个端获取，也可以通过搜索从任何一条历史消息开始上下回溯
- 未读数多终端实时同步
- 收藏列表多终端实时同步
- 联系人信息/群组信息实时同步

业界IM多终端支持

- Skype（支持多终端同步，未读同步）
- QQ （支持多终端 不支持离线消息同步）
- 微信 （PC通过旁路机制实现 未实现跨终端同步）
- Slack (支持多终端 基于拉取历史记录实现 非同步协议)

架构优点

- 用户在线的情况下，大多数情况变更是直接投递下去的。比通知→拉取模式和服务器的交互少，更省资源
- 离线缓存比较容易实现，离线浏览的体验会比较好。
- 能保证终端和服务器的数据一致性
- 相对比较通用，可以适用于多个业务场景

架构缺点

- 本地客户端的实现逻辑比较重。和轻客户端，重服务器的思路有冲突
- 只能保证同一个Folder的最终一致性
- 未实现双向同步

Grouk

Thanks

王渊命
@jolestar