1. 图片以guid命名并且与用户关联并缓存在client，缓存设定一个过期策略（大小，时间），聊天记录也需要存储，但最多存储3天的（这个可以defer）。
2. 独立出rest api server，content server，p2p server，server 之间的通信通过进程间消息进行（mq），比如注册完后需要在 content server、p2p server 中进行缓存，登出后需要进行缓存的清理，这种msg 必须是request-reply模式。
3. 用户登录后，首先推送一定数量的人员（屏幕滚动1次能放得下的数量）过去，包括其头像（也通过guid缓存），昵称，用户的照片列表，然后后台依次获取用户的信息，并能匹配图片的guid等信息，有缓存则不下载。
4. 用户往下滚动，则继续执行第3点。
5. Jexus作为web server，是否使用 signalR 代替content Server（需要验证mono下是否能使用），因为涉及到session（缓存）的问题，如果使用 content Server，一旦 content Server 分布式部署，则api server 与 content server 使用 msg 的方式进行单机缓存的设置就会无法进行，就得需要采用数据库存储或者分布式的缓存进行处理。
6. 采用signalR ，则后端使用redis进行图片的缓存（暂时不使用自己写的了，怕有错误，后续可试试），但图片的保存到本地是必须的。
7. 关系数据库用mysql